جامعة الملك سعود
قسم الإحصاء وبحوث العمليات

طرق الحسابات في بحوث العمليات

بإستخدام

EXCEL SOLVER, LINGO, SageMath AND THE MATHEMATICAL MODELING LANGUAGE R

تأليف
د. عدنان ماجد عبد الرحمن بري
استاذ الإحصاء وبحوث العمليات المشارك

بسم الله الرحمن الرحيم

الحمد لله رب العالمين والصلاة والسلام على اشرف خلق الله سيدنا ونبينا محمد وعلى آله وصحبه وسلم.

أما بعد.

هذه هي المسودة الأولى لكتاب طرق الحسابات في بحوث العمليات بإستخدام EXCEL SOLVER, LINGO, AND THE MATHEMATICAL MODELING LANGUAGE R لطلاب مرحلة البكالوريوس.

وكما ذكرت في مقدمة كتبي السابقة " طرق التنبؤ الإحصائي – جزء أول – " و" النمذجة والمحاكاة بإستخدام Excel, SIMAN, Arena and General Purpose Simulation System (GPSS WORLD)" و بناء النماذج بإستخدام Excel and Vensim أن هذا الكتاب كسابقيه سيظل مسودة إلى ماشاء الله لأني وبإذن الله تعالى سوف أقوم بتطويره وتجديده وتحسينه بشكل مستمر وسيظل بشكله الإلكتروني هذا لأني أعتقد ان العلوم والتقنية تتطور يوميا وبشكل متسارع بحيث ان وضعها في كتاب جامد ستاتيكي لايتناسب مع ديناميكية الموضوع وخاصة في عصر ثورة المعلومات والإنترنت.

يغطي هذا الكتاب بعض الطرق المستخدمة في الحسابات للمساعدة في حل النماذج التي تنتج عن التكاوين المختلفة لمشاكل الأمثلية والتي تغطي البرمجة الرياضية والشبكات والتخصيص والجدولة وإتخاذ القرار والنمذجة المالية وغيرها.

لقد إستخدمت برنامج صفحات النشر Excel في حل الكثير من الأمثلة وذلك لمقدرته الهائلة في حل الصيغ العددية للمشتقات والتكاملات والمعادلات التفاضلية، كما استعرضت مقدرته في إيجاد التكاملات بواسطة المحاكاة وتقدير المعالم في النماذج غير خطية. كما استعرضت مقدرة EXCEL SOLVER في حل مشاكل الأمثلية في البرمجة الرياضية.

بقية الكتاب تستعرض بعض الحزم المتخصصة والتي تستخدم لحل مشاكل بحوث العمليات مثل برنامج النمذجة LINGO وبرنامج النمذجة الرياضية R.

هذا وارجوا من الله ان يوفقني في إنجاز هذا العمل لوجهه الكريم ولإثراء المكتبة العربية الفقيرة إلى مثل هذا الكتاب.

سيكون هذا الكتاب مجاني لأي طالب علم وسيكون متواجد على شبكة الإنترنت في الموقع http://www.abarry.ws/books/CalculationMethodsBookWithR.pdf

والله الموفق.

المؤلف
د. عدنان ماجد عبد الرحمن بري

جامعة الملك سعود
ربيع الثاني 1431 هـ
إبريل 2010 م

محتويات الكتاب

المحتوى	الصفحة
مقدمة:	3
محتويات الكتاب:	4
الفصل الأول:	6
بعض الطرق العددية بإستخدام EXCEL	6
تقريب عددي لمشتقات الدوال بإستخدام EXCEL	6
تقريب المشتقة الأولى	6
تقريب المشتقة الثانية	6
تقريب الدوال بكثيرات حدود	7
تقريب عددي لتكاملات الدوال بإستخدام EXCEL	9
تطبيق المنحنيات وتقدير المعالم Curve Fitting and Parameters Estimation	13
حل المعادلات التفاضلية العادية The Solution of Ordinary Differential Equations	22
إيجاد تكامل بالمحكاة Monte Carlo Integration	26
الفصل الثاني:	30
إستخدام EXCEL SOLVER في حل مسائل البرمجة الرياضية	30
مثال 1: مثال على البرمجة الخطية	30
مثال 2: مثال آخر على البرمجة الخطية	36
مثال 3: البرمجة العددية Integer Programming	41
مثال 4: مثال على الشبكات والنقل	44
مثال 5: مثال آخر لمشكلة النقل Transportation Problem.....	50
مثال 6: مثال على التخصيص Assignment Problem	55
مثال 7: مثال آخر على التخصيص	59
مثال 8: مشكلة أقصر طريق Shortes Path Problem.....	64
مثال 9: مشكلة التدفق الأعظم Maximal Flow Problem	75
الفصل الثالث:	82
مقدمة إلى لغة البرمجة الرياضية للأمثلية LINGO	82
البيانات	82
دالة الهدف The Objective Function	83
القيود Constraints	84
تعريف المجموعات	85
إدخال البيانات	86
النموذج بشكله الكامل	86
المخرجات	87
أسماء القيود	89
عنوان النموذج	91
الفصل الرابع:	92
أمثلة وحالات دراسة على لغة النمذجة LINGO	92

مثال 1: البرمجة الخطية 92
مثال 2: برمجة خطية 94
مثال 3: برمجة عددية 96
مثال 4: مشكلة نقل 98
مثال 5: مشكلة تخصيص 100
مثال 6: مشكلة أقصر طريق 103
مثال 7: مشكلة التدفق الأعظم 107
مثال 8: إنحدار بسيط 111
مثال 9: مشكلة الجراب 115
مثال 10: نموذج سلسلة ماركوف 118
الفصل الخامس: 129
حل مشاكل الأمثلية بواسطة R 129
حزمة Rglpk 129
مثال 1 : البرمجة الخطية 130
مثال 2: برمجة خطية 130
حزمة linprog 131
مثال 1: 132
حزمة lpSolve 132
مثال 1: 133
مثال 2: 134
البرمجة العددية لمشكلة تخصيص 135
برمجة مشكلة نقل 136
مثال 3: 137
مثل آخر لمشكلة نقل 139
مثال تخصيص 140
مشكلة الجراب 142
ملحق: مقدمة للغة النمذجة الرياضية R 144

أسئلة وإجابات الإختبارات السابقة 235
المراجع 258

الفصل الأول

بعض الطرق العددية بإستخدام EXCEL:

تقريب عددي لمشتقات الدوال بإستخدام EXCEL

تقريب المشتقة الأولى:

1) تقريب المشتقة الأولى بثلاثة نقاط

$$f'(x)=\frac{f(x+h)-f(x-h)}{2h}+O(h^2)$$

2) تقريب المشتقة الأولى باربعة نقاط

$$f'(x)=\frac{-f(x+2h)+6f(x+h)-3f(x)-2f(x-h)}{6h}+O(h^3)$$

3) تقريب المشتقة الأولى بخمسة نقاط

$$f'(x)=\frac{-2f(x+2h)+16f(x+h)-16f(x-h)+2f(x-2h)}{24h}+O(h^4)$$

تقريب المشتقة الثانية:

1) تقريب المشتقة الثانية بثلاثة نقاط

$$f''(x)=\frac{f(x+h)-2f(x)+f(x-h)}{h^2}+O(h^2)$$

2) تقريب المشتقة الثانية بأربعة نقاط

$$f''(x)=\frac{f(x+h)-2f(x)+f(x-h)}{h^2}+O(h^2)$$

3) تقريب المشتقة الثانية بخمسة نقاط

$$f''(x)=\frac{-f(x+2h)+16f(x+h)-30f(x)+16f(x-h)-f(x-2h)}{12h^2}+O(h^4)$$

حيث لقيمة $q>0$ نقول ان $g(h)$ تكون $O(h^q)$ عند $h=0$ إذا وجد عددين C و D بحيث كلما كان $|h|\le D$ فإن $|g(h)|\le C|h^q|$ (أمثلة $\sin(h)=h+O(h^3)$ و $\cos(h)=1+O(h^2)$)

ملاحظة: الصيغ السابقة مشتقة من الصيغة

$$f(x+jh)=\sum_{n=0}^{k}\frac{f^{(n)}(x)}{n!}(jh)^n+O(h^{k-n+1}),\quad j=0,\pm1,\pm2,...$$

مثال:

أوجد المشتقتين الأولى و الثانية للدالة $f(x)=e^x$ عند $x=0$ و $h=0.1$.

	A	B	C	D	E
1	h=	0.1	f(x)	f(0)=	1
2	x=	0			
3	3 point		=(EXP(B2+B1)-EXP(B2-B1))/(2*B1)	error=	=E1-C3
4	4 point		=(-EXP(B2+2*B1)+6*EXP(B2+B1)-3*EXP(B2)-2*EXP(B2-B1))/(6*B1)		=E1-C4
5	5 point		=(-2*EXP(B2+2*B1)+16*EXP(B2+B1)-16*EXP(B2-B1)+2*EXP(B2-2*B1))/(24*B1)		=E1-C5
6					

	A	B	C	D	E	F
1	h=	0.1	f(x)	f(0)=	1	
2	x=	0				
3	3 point		1.001668	error=	-0.0016675	
4	4 point		0.999913		8.681E-05	
5	5 point		0.999997		3.3373E-06	
6						

تمرين:

جرب قيم $h=0.01,0.001$ ماذا تلاحظ على الخطأ.

تقريب الدوال بكثيرات حدود:

تقرب أي دالة $f(x)$ والتي تكون جميع مشتقاتها موجودة ومعرفة حتى المشتقة النونية (n) بكثيرة حدود $p_{n-1}(x)$ كالتالي:

$$p_{n-1}(x)=\sum_{k=0}^{n-1}\frac{f^{(k)}(x_0)}{k!}(x-x_0)^k$$

والتي تسمى بكثيرات حدود تايلور. العلاقة السابقة يمكن حسابها تكراريا كالتالي:

$$p_j(x)=p_{j-1}(x)+\frac{f^{(j)}}{j!}(x-x_0)^j$$

مثال:

قرب الدالة $f(x)=\sin(x)$ بكثيرة حدود حول النقطة $x_0=1$.

الحل: بإستخدام الصيغة التكرارية

$$p_0(x)=f(x_0)=\sin(1)$$

$$p_1(x)=p_0(x)+f'(x_0)(x-x_0)=\sin(1)+\cos(1)(x-1)$$

$$p_2(x)=p_1(x)+\frac{1}{2}f''(x_0)(x-x_0)^2=\sin(1)+\cos(1)(x-1)-\frac{1}{2}\sin(1)(x-1)^2$$

$$p_3(x)=p_2(x)+\frac{1}{6}f^{(3)}(x_0)(x-x_0)^3=\sin(1)+\cos(1)(x-1)-\frac{1}{2}\sin(1)(x-1)^2-\frac{1}{6}\cos(1)(x-1)^3$$

تمرين:

قرب الدالة $f(x)=e^x$ حول النقطة $x_0=0$.

تقريب عددي لتكاملات الدوال بإستخدام EXCEL

سوف نقوم بإجراء تكاملات عددية من الشكل $\int_a^b f(x)dx$ ، الفترة $[a,b]$ تقسم إلى $N>1$ فترات عند النقاط $a=x_0,x_1,...,x_N=b$.

سوف نرمز لحجم الخطوة بالرمز $h>0$ ويكون $x_{k+1}=x_k+h$ ونرمز للدالة عند x_k بالرمز $f_k=f(x_k)$ و ζ هي نقطة في مجال التكامل.

1) قاعدة الترابزويد البسيطة:

$$\int_{x_0}^{x_1} f(x)dx \approx \frac{h}{2}(f_0+f_1), \quad error=-\frac{h^3}{12}f''(\zeta)$$

2) قاعدة سمبسون البسيطة:

$$\int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3}(f_0+4f_1+f_2), \quad error=-\frac{h^5}{90}f^{(4)}(\zeta)$$

3) قاعدة سمبسون $\frac{3}{8}$

$$\int_{x_0}^{x_3} f(x)dx \approx \frac{3h}{8}(f_0+3f_1+3f_2+f_3), \quad error=-\frac{3h^5}{80}f^{(4)}(\zeta)$$

4) قاعدة بود البسيطة:

$$\int_{x_0}^{x_4} f(x)dx \approx \frac{2h}{45}(7f_0+32f_2+32f_3+7f_4), \quad error=-\frac{8h^7}{945}f^{(6)}(\zeta)$$

قواعد مركبة:

سوف نرمز لحجم الخطوة بالرمز $h=x_{i+1}-x_i=\frac{b-a}{N}$ و $x_{k+1}=x_k+h$ و $f_k=f(x_k)$ و ζ هي نقطة في مجال التكامل.

1) قاعدة الترابزويد المركبة:

$$\int_a^b f(x)\,dx \approx h\left(\frac{f_0}{2}+f_1+\cdots+f_{N-1}+\frac{f_N}{2}\right), \quad error=-\frac{(b-a)h^2}{12}f''(\zeta)$$

2) قاعدة الترابزويد المعدلة:

$$\int_a^b f(x)\,dx \approx h\left(\frac{f_0}{2}+f_1+\cdots+f_{N-1}+\frac{f_N}{2}\right)+\frac{h}{24}(-f_{-1}+f_1+f_{N-1}-f_{N+1}),$$

$$error=-\frac{11(b-a)h^4}{720}f^{(4)}(\zeta)$$

3) قاعدة سمبسون المركبة:

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\left[f_0+4(f_1+f_3+\cdots+f_{M-2})+2(f_2+f_4+\cdots+f_{M-1})+f_M\right],$$

$$error=-\frac{(b-a)h^4}{180}f^{(4)}(\zeta), \quad \left(h=\frac{b-a}{M}\right), \quad M=2N$$

مثال:

أوجد التكامل العددي للتالي:

$$\int_0^{10} 10\sin\left(1-0.1x\right)dx$$

القيمة الحقيقية هي 45.96977

سوف نستخدم صيغة منتصف النقطة:

$$\int_a^b f\left(x\right)dx \approx h\left[f\left(x_1\right)+f\left(x_2\right)+\cdots+f\left(x_N\right)\right], \quad h=\frac{b-a}{N},$$

$$x_j = a+\left(j-\frac{1}{2}\right)h, \quad j=1,2,...,N$$

وقاعدة الترابزويد:

$$\int_a^b f\left(x\right)dx \approx h\left[\frac{1}{2}f\left(x_0\right)+f\left(x_1\right)+\cdots+f\left(x_{N-1}\right)+\frac{1}{2}f\left(x_N\right)\right], \quad h=\frac{b-a}{N},$$

$$x_j = a+jh, \quad j=0,1,2,...,N$$

وقاعدة سمبسون M نقطة:

$$\int_a^b f\left(x\right)dx \approx \frac{h}{3}[f\left(x_0\right)+4\left(f\left(x_1\right)+f\left(x_3\right)+\cdots+f\left(x_{M-1}\right)\right)$$
$$+2\left(f\left(x_2\right)+f\left(x_4\right)+\cdots+f\left(x_{M-2}\right)\right)+f\left(x_M\right)],$$

$$M=2N, \quad h=\frac{b-a}{M}, \quad x_j=a+jh, \quad j=0,1,2,...,M$$

الحل:

أدخل التالي في صفحة من EXCEL

	A	B	C	D	E	F	G	H	I	J
1	Midpoint Formula					a=	0		h=(b-a)/N	1
2	j	x(j)	f(x(j))	Int		b=	10			
3	1	=$G$1+(A3-0.5)*$J$1	=10*SIN(1-0.1*B3)	=J1*SUM(C3:C12)		true=	45.96977		error=	=G3-D3
4	2	=$G$1+(A4-0.5)*$J$1	=10*SIN(1-0.1*B4)							
5	3	=$G$1+(A5-0.5)*$J$1	=10*SIN(1-0.1*B5)							
6	4	=$G$1+(A6-0.5)*$J$1	=10*SIN(1-0.1*B6)							
7	5	=$G$1+(A7-0.5)*$J$1	=10*SIN(1-0.1*B7)							
8	6	=$G$1+(A8-0.5)*$J$1	=10*SIN(1-0.1*B8)							
9	7	=$G$1+(A9-0.5)*$J$1	=10*SIN(1-0.1*B9)							
10	8	=$G$1+(A10-0.5)*$J$1	=10*SIN(1-0.1*B10)							
11	9	=$G$1+(A11-0.5)*$J$1	=10*SIN(1-0.1*B11)							
12	10	=$G$1+(A12-0.5)*$J$1	=10*SIN(1-0.1*B12)							
13										

فينتج:

	A	B	C	D	E	F	G	H	I	J
1	Midpoint Formula					a=	0		h=(b-a)/N	1
2	j	x(j)	f(x(j))	Int		b=	10			
3	1	0.5	8.134155	45.98893		true=	45.97		error=	-0.019
4	2	1.5	7.512804							
5	3	2.5	6.816388							
6	4	3.5	6.051864							
7	5	4.5	5.226872							
8	6	5.5	4.349655							
9	7	6.5	3.428978							
10	8	7.5	2.47404							
11	9	8.5	1.494381							
12	10	9.5	0.499792							
13										

	A	B	C	D	E	F	G
1	Trapezoidal rule					h=	1
2	j	x(j)	f(x(j))			a=	0
3	0	=$G$2+A3*$G$1	=10*SIN(1-0.1*B3)	=0.5*C3		b=	10
4	1	=$G$2+A4*$G$1	=10*SIN(1-0.1*B4)	=C4		Int=	=G1*SUM(D3:D13)
5	2	=$G$2+A5*$G$1	=10*SIN(1-0.1*B5)	=C5		true=	45.96977
6	3	=$G$2+A6*$G$1	=10*SIN(1-0.1*B6)	=C6		error=	=G5-G4
7	4	=$G$2+A7*$G$1	=10*SIN(1-0.1*B7)	=C7			
8	5	=$G$2+A8*$G$1	=10*SIN(1-0.1*B8)	=C8			
9	6	=$G$2+A9*$G$1	=10*SIN(1-0.1*B9)	=C9			
10	7	=$G$2+A10*$G$1	=10*SIN(1-0.1*B10)	=C10			
11	8	=$G$2+A11*$G$1	=10*SIN(1-0.1*B11)	=C11			
12	9	=$G$2+A12*$G$1	=10*SIN(1-0.1*B12)	=C12			
13	10	=$G$2+A13*$G$1	=10*SIN(1-0.1*B13)	=0.5*C13			
14							

	A	B	C	D	E	F	G
1	Trapezoidal rule					h=	1
2	j	x(j)	f(x(j))			a=	0
3	0	0	8.4147	4.2074		b=	10
4	1	1	7.8333	7.8333		Int=	45.931
5	2	2	7.1736	7.1736		true=	45.97
6	3	3	6.4422	6.4422		error=	0.0383
7	4	4	5.6464	5.6464			
8	5	5	4.7943	4.7943			
9	6	6	3.8942	3.8942			
10	7	7	2.9552	2.9552			
11	8	8	1.9867	1.9867			
12	9	9	0.9983	0.9983			
13	10	10	0	0			

	A	B	C	D	F	G
1	M-point	Simpson's	rule		h=	0.5
2	j	x(j)	f(x(j))		a=	0
3	0	=$G$2+A3*$G$1	=10*SIN(1-0.1*B3)	=C3	b=	10
4	1	=$G$2+A4*$G$1	=10*SIN(1-0.1*B4)	=IF(ISODD(A4),4*C4,IF(ISEVEN(A4),2*C4,""))	N=	10
5	2	=$G$2+A5*$G$1	=10*SIN(1-0.1*B5)	=IF(ISODD(A5),4*C5,IF(ISEVEN(A5),2*C5,""))	M=	20
6	3	=$G$2+A6*$G$1	=10*SIN(1-0.1*B6)	=IF(ISODD(A6),4*C6,IF(ISEVEN(A6),2*C6,""))	int=	=(G1/3)*SUM(D3:D23)
7	4	=$G$2+A7*$G$1	=10*SIN(1-0.1*B7)	=IF(ISODD(A7),4*C7,IF(ISEVEN(A7),2*C7,""))	true=	45.96977
8	5	=$G$2+A8*$G$1	=10*SIN(1-0.1*B8)	=IF(ISODD(A8),4*C8,IF(ISEVEN(A8),2*C8,""))	error=	=G7-G6
9	6	=$G$2+A9*$G$1	=10*SIN(1-0.1*B9)	=IF(ISODD(A9),4*C9,IF(ISEVEN(A9),2*C9,""))		
10	7	=$G$2+A10*$G$1	=10*SIN(1-0.1*B10)	=IF(ISODD(A10),4*C10,IF(ISEVEN(A10),2*C10,""))		
11	8	=$G$2+A11*$G$1	=10*SIN(1-0.1*B11)	=IF(ISODD(A11),4*C11,IF(ISEVEN(A11),2*C11,""))		
12	9	=$G$2+A12*$G$1	=10*SIN(1-0.1*B12)	=IF(ISODD(A12),4*C12,IF(ISEVEN(A12),2*C12,""))		
13	10	=$G$2+A13*$G$1	=10*SIN(1-0.1*B13)	=IF(ISODD(A13),4*C13,IF(ISEVEN(A13),2*C13,""))		
14	11	=$G$2+A14*$G$1	=10*SIN(1-0.1*B14)	=IF(ISODD(A14),4*C14,IF(ISEVEN(A14),2*C14,""))		
15	12	=$G$2+A15*$G$1	=10*SIN(1-0.1*B15)	=IF(ISODD(A15),4*C15,IF(ISEVEN(A15),2*C15,""))		
16	13	=$G$2+A16*$G$1	=10*SIN(1-0.1*B16)	=IF(ISODD(A16),4*C16,IF(ISEVEN(A16),2*C16,""))		
17	14	=$G$2+A17*$G$1	=10*SIN(1-0.1*B17)	=IF(ISODD(A17),4*C17,IF(ISEVEN(A17),2*C17,""))		
18	15	=$G$2+A18*$G$1	=10*SIN(1-0.1*B18)	=IF(ISODD(A18),4*C18,IF(ISEVEN(A18),2*C18,""))		
19	16	=$G$2+A19*$G$1	=10*SIN(1-0.1*B19)	=IF(ISODD(A19),4*C19,IF(ISEVEN(A19),2*C19,""))		
20	17	=$G$2+A20*$G$1	=10*SIN(1-0.1*B20)	=IF(ISODD(A20),4*C20,IF(ISEVEN(A20),2*C20,""))		
21	18	=$G$2+A21*$G$1	=10*SIN(1-0.1*B21)	=IF(ISODD(A21),4*C21,IF(ISEVEN(A21),2*C21,""))		
22	19	=$G$2+A22*$G$1	=10*SIN(1-0.1*B22)	=IF(ISODD(A22),4*C22,IF(ISEVEN(A22),2*C22,""))		
23	20	=$G$2+A23*$G$1	=10*SIN(1-0.1*B23)	=C23		
24						

	A	B	C	D	E	F	G
1	M-point Simpson's rule					h=	0.5
2	j	x(j)	f(x(j))			a=	0
3	0	0	8.4147	8.4147		b=	10
4	1	0.5	8.1342	32.537		N=	10
5	2	1	7.8333	15.667		M=	20
6	3	1.5	7.5128	30.051		int=	45.96977101
7	4	2	7.1736	14.347		true=	45.96977
8	5	2.5	6.8164	27.266		error=	-1.00983E-06
9	6	3	6.4422	12.884			
10	7	3.5	6.0519	24.207			
11	8	4	5.6464	11.293			
12	9	4.5	5.2269	20.907			
13	10	5	4.7943	9.5885			
14	11	5.5	4.3497	17.399			
15	12	6	3.8942	7.7884			
16	13	6.5	3.429	13.716			
17	14	7	2.9552	5.9104			
18	15	7.5	2.474	9.8962			
19	16	8	1.9867	3.9734			
20	17	8.5	1.4944	5.9775			
21	18	9	0.9983	1.9967			
22	19	9.5	0.4998	1.9992			
23	20	10	0	0			
24							

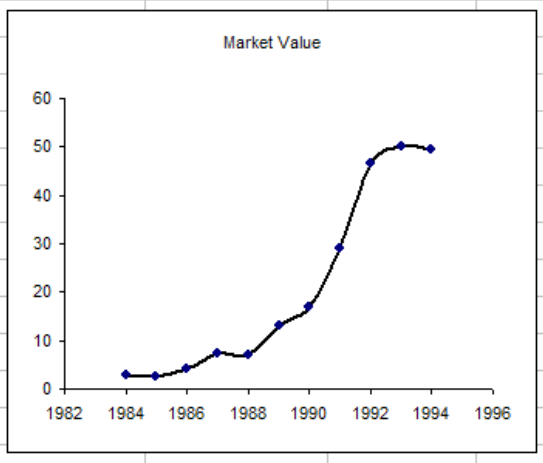
تطبيق المنحنيات وتقدير المعالم Curve Fitting and Parameters Estimation:

البيانات التالية هي نسبة قيمة السوق لمنتجات مكروسوفت و إنتل من سنة 1984حتى سنة 1994

Year	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994
Market Value	3.0	2.5	4.0	7.5	7.0	13.0	17.0	29.0	46.5	50.0	49.5

ندخل هذه البيانات على صفحة نشر ونرسمها.

	A	B	C	D
1	Year	Market Value	Base Year	
2	1984	3	0	
3	1985	2.5	1	
4	1986	4	2	
5	1987	7.5	3	
6	1988	7	4	
7	1989	13	5	
8	1990	17	6	
9	1991	29	7	
10	1992	46.5	8	
11	1993	50	9	
12	1994	49.5	10	
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				



واضح أن المنحني على شكل حرف S وهذه خاصية المنحنى اللوجستي والذي له الشكل الرياضي:

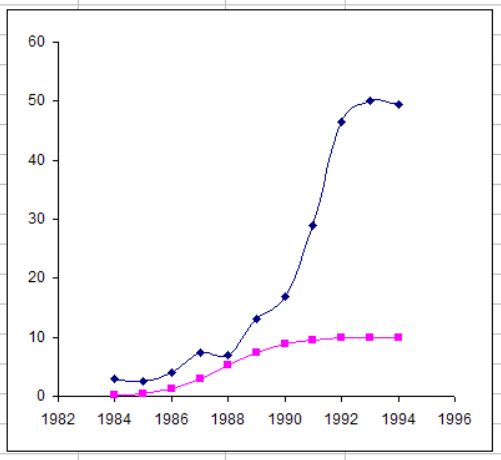
$$x(t)=\frac{a}{1+be^{-ct}},\ t\geq 0$$

من البيانات السابقة نريد تعيين قيم المعالم a و b و c والتي تعطي أفضل تطبيق للبيانات ولذلك ندخل البيانات في صفحة نشر كالتالي:

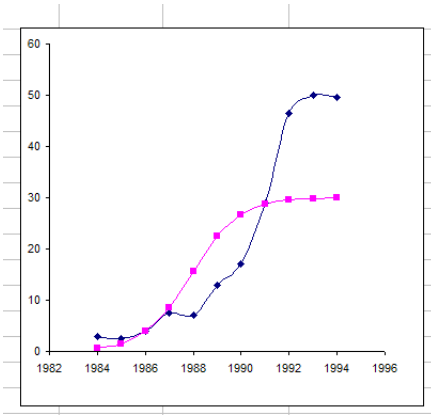
	A	B	C	D	E	F
1	Year	Base Year	Market Va	Fit	Error	
2	1984	0	3	=$F$15/(1+$F$16*EXP(-$F$17*B2))	=(D2-C2)^2	
3	1985	1	2.5	=$F$15/(1+$F$16*EXP(-$F$17*B3))	=(D3-C3)^2	
4	1986	2	4	=$F$15/(1+$F$16*EXP(-$F$17*B4))	=(D4-C4)^2	
5	1987	3	7.5	=$F$15/(1+$F$16*EXP(-$F$17*B5))	=(D5-C5)^2	
6	1988	4	7	=$F$15/(1+$F$16*EXP(-$F$17*B6))	=(D6-C6)^2	
7	1989	5	13	=$F$15/(1+$F$16*EXP(-$F$17*B7))	=(D7-C7)^2	
8	1990	6	17	=$F$15/(1+$F$16*EXP(-$F$17*B8))	=(D8-C8)^2	
9	1991	7	29	=$F$15/(1+$F$16*EXP(-$F$17*B9))	=(D9-C9)^2	
10	1992	8	46.5	=$F$15/(1+$F$16*EXP(-$F$17*B10))	=(D10-C10)^2	
11	1993	9	50	=$F$15/(1+$F$16*EXP(-$F$17*B11))	=(D11-C11)^2	
12	1994	10	49.5	=$F$15/(1+$F$16*EXP(-$F$17*B12))	=(D12-C12)^2	
13				Error sum of squares	=SUM(E2:E12)	
14						
15					a=	10
16					b=	50
17					c=	1

لاحظ أننا أخذنا السنة 1984 كسنة أساس وساويناها بالصفر (المنحنى اللوجستي يجب أن يبدأ من الصفر)، ندخل قيم أولية للمعالم $a=10$ و $b=50$ و $c=1$ ونرسم Market Value و Fit مع الزمن.

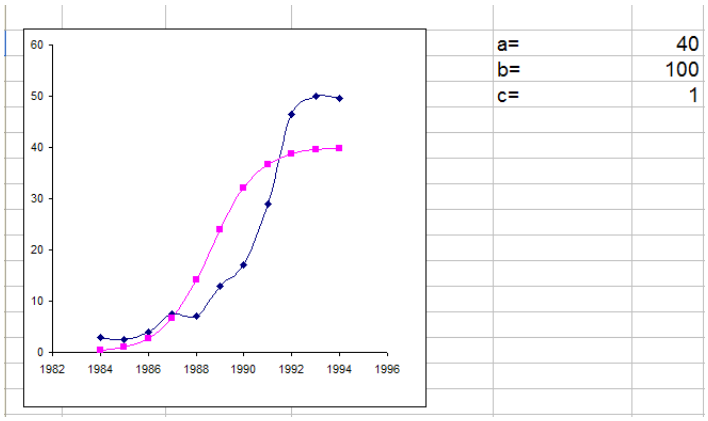
	A	B	C	D	E	F
1	Year	Base Year	Market Val	Fit	Error	
2	1984	0	3	0.196078431	7.861976	
3	1985	1	2.5	0.515624132	3.937748	
4	1986	2	4	1.287537486	7.357453	
5	1987	3	7.5	2.865860462	21.47525	
6	1988	4	7	5.219800734	3.169109	
7	1989	5	13	7.480005851	30.47034	
8	1990	6	17	8.897291025	65.65389	
9	1991	7	29	9.563940746	377.7604	
10	1992	8	46.5	9.835035655	1344.32	
11	1993	9	50	9.938673512	1604.91	
12	1994	10	49.5	9.977351447	1562.04	
13				Error sum of squ	5028.955	
14						
15					a=	10
16					b=	50
17					c=	1
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						



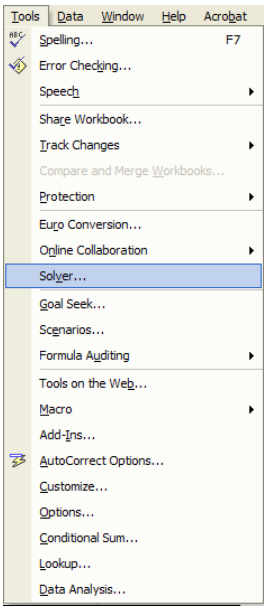
لاحظ أن المنحنى للقيم المطبقة بعيد عن المنحنى الناتج من البيانات لهذا نجرب قيم مختلفة للمعالم كالتالي:



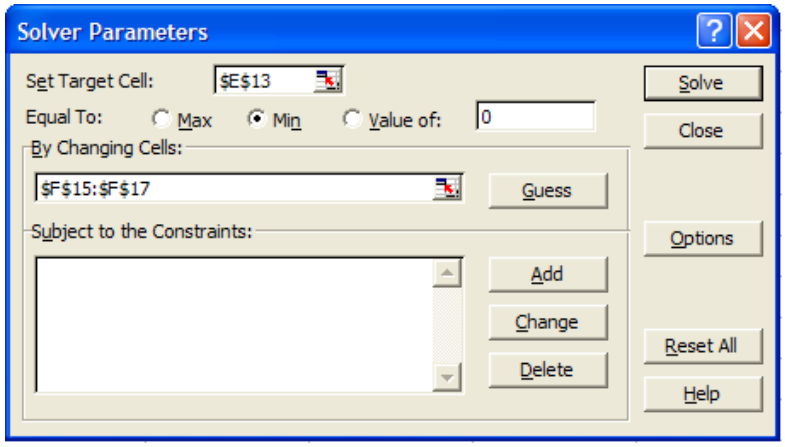
	a=	30
	b=	50
	c=	1



من الشكلين السابقين نجد أن القيم الأولية للمعالم $a=40$ و $b=100$ و $c=1$ مناسبة، نضع المؤشر في الخلية المراد تصغيرقيمتها E13 ونختار Solver من قائمة الأدوات كالتالي:

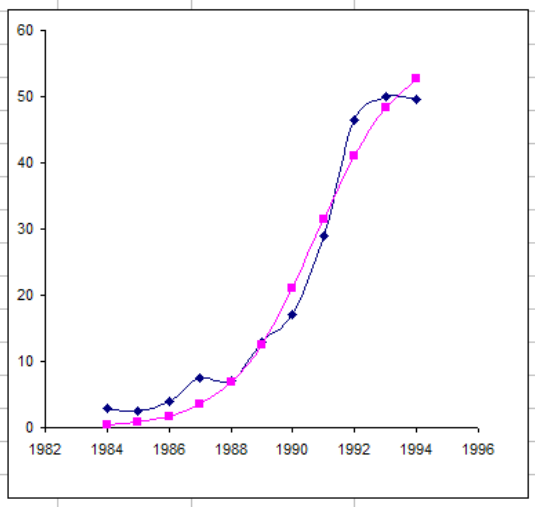


فتظهر النافذة التالية:



لتصغيرقيمته $E$13 (لاحظ العنونة المطلقة) نختار Min ثم في صندوق إختيار القيم التى تغير نختار المجال الموجودة به تقديرات المعالم الأولية أي $F$15:$F$17 ثم Solve فتظهر النتائج التالية:

	A	B	C	D	E	F
1	Year	Base Year	Market Value	Fit	Error	
2	1984	0	3	0.414542265	6.684591701	
3	1985	1	2.5	0.852705443	2.713579359	
4	1986	2	4	1.739946887	5.107840073	
5	1987	3	7.5	3.493689512	16.05052372	
6	1988	4	7	6.800502541	0.039799236	
7	1989	5	13	12.51540951	0.234827947	
8	1990	6	17	21.04921791	16.39616568	
9	1991	7	29	31.36782219	5.606581939	
10	1992	8	46.5	41.08403009	29.3327301	
11	1993	9	50	48.3020243	2.883121473	
12	1994	10	49.5	52.77515711	10.72665407	
13				Error sum of squares	95.7764153	
14						
15					a=	57.76
16					b=	138.3
17					c=	0.729
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						

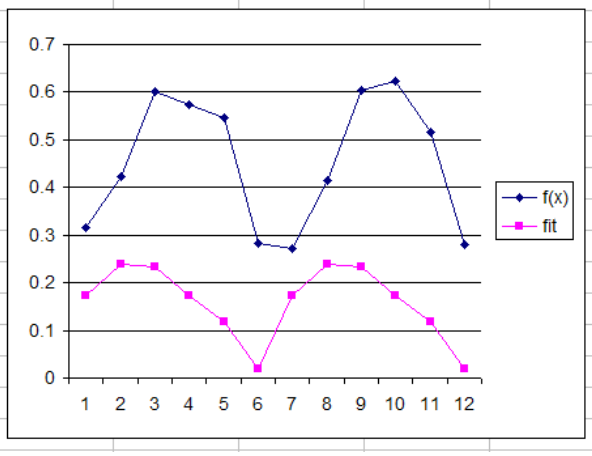


وهذا أفضل تطبيق للمنحنى على البيانات المعطاة. أي أن البيانات تنطبق بشكل جيد على المنحنى

$$x(t)=\frac{57.76}{1+138.3e^{-0.729t}},\ t\geq 0$$

مثال آخر:

	A	B	C	D	E	F
1	x	f(x)	fit	error sq	a=	1
2	0.25	0.316	0.17227	0.020658	b=	2
3	0.5	0.422	0.238651	0.033617		
4	1	0.602	0.232544	0.136498		
5	1.5	0.573	0.173343	0.159726		
6	2	0.546	0.11702	0.184024		
7	4	0.282	0.01798	0.069706		
8	0.25	0.273	0.17227	0.010147		
9	0.5	0.415	0.238651	0.031099		
10	1	0.604	0.232544	0.137979		
11	1.5	0.622	0.173343	0.201293		
12	2	0.516	0.11702	0.159185		
13	4	0.279	0.01798	0.068131		
14			SSE=	1.212063		
15						



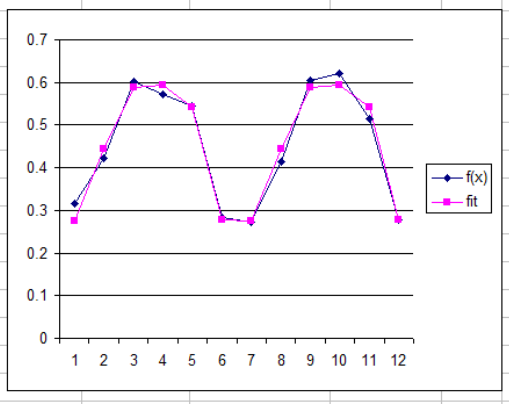
سوف نطبق على البيانات هذه الدالة

$$f(x)=\frac{a\left(e^{-bx}-e^{-ax}\right)}{a-b}$$

	A	B	C	D	E	F
1	x	f(x)	fit	error sq	a=	1
2	0.25	0.316	=$F$1*(EXP(-A2*$F$2)-EXP(-A2*$F$1))/($F$1-$F$2)	=(B2-C2)^2	b=	2
3	0.5	0.422	=$F$1*(EXP(-A3*$F$2)-EXP(-A3*$F$1))/($F$1-$F$2)	=(B3-C3)^2		
4	1	0.602	=$F$1*(EXP(-A4*$F$2)-EXP(-A4*$F$1))/($F$1-$F$2)	=(B4-C4)^2		
5	1.5	0.573	=$F$1*(EXP(-A5*$F$2)-EXP(-A5*$F$1))/($F$1-$F$2)	=(B5-C5)^2		
6	2	0.546	=$F$1*(EXP(-A6*$F$2)-EXP(-A6*$F$1))/($F$1-$F$2)	=(B6-C6)^2		
7	4	0.282	=$F$1*(EXP(-A7*$F$2)-EXP(-A7*$F$1))/($F$1-$F$2)	=(B7-C7)^2		
8	0.25	0.273	=$F$1*(EXP(-A8*$F$2)-EXP(-A8*$F$1))/($F$1-$F$2)	=(B8-C8)^2		
9	0.5	0.415	=$F$1*(EXP(-A9*$F$2)-EXP(-A9*$F$1))/($F$1-$F$2)	=(B9-C9)^2		
10	1	0.604	=$F$1*(EXP(-A10*$F$2)-EXP(-A10*$F$1))/($F$1-$F$2)	=(B10-C10)^2		
11	1.5	0.622	=$F$1*(EXP(-A11*$F$2)-EXP(-A11*$F$1))/($F$1-$F$2)	=(B11-C11)^2		
12	2	0.516	=$F$1*(EXP(-A12*$F$2)-EXP(-A12*$F$1))/($F$1-$F$2)	=(B12-C12)^2		
13	4	0.279	=$F$1*(EXP(-A13*$F$2)-EXP(-A13*$F$1))/($F$1-$F$2)	=(B13-C13)^2		
14			SSE=	=SUM(D2:D13)		

وبإستخدام Solver نجد الحل:

	A	B	C	D	E	F
1	x	f(x)	fit	error sq	a=	1.3739
2	0.25	0.316	0.27576233	0.001619	b=	0.4022
3	0.5	0.422	0.44498469	0.000528		
4	1	0.602	0.58780591	0.000201		
5	1.5	0.573	0.59336895	0.000415		
6	2	0.546	0.54195088	1.64E-05		
7	4	0.282	0.27718074	2.32E-05		
8	0.25	0.273	0.27576233	7.63E-06		
9	0.5	0.415	0.44498469	0.000899		
10	1	0.604	0.58780591	0.000262		
11	1.5	0.622	0.59336895	0.00082		
12	2	0.516	0.54195088	0.000673		
13	4	0.279	0.27718074	3.31E-06		
14			SSE=	0.005469		
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						

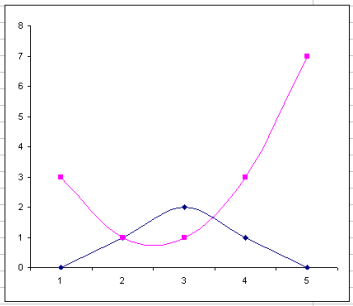


لاحظ جودة التطابق.

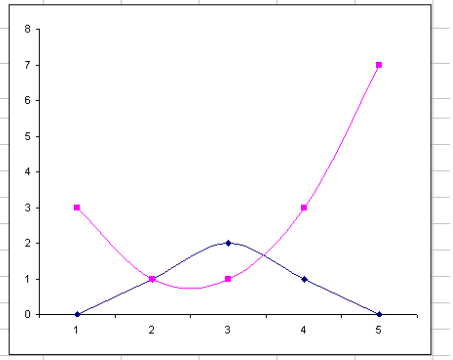
ملاحظة: هذا مثال للإنحدار غير الخطي.

مثال آخر:

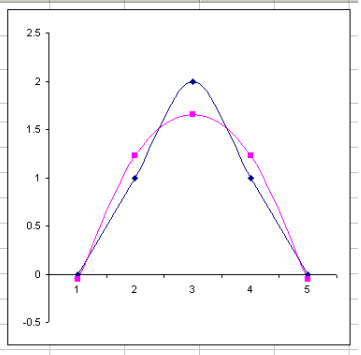
	A	B	C	D	E
1	Fitting Polynomial of degree 2				
2	p(x)=a(0)+a(1)x+a(2)x*x				
3	to the data by Least Squares				
4					
5	x(j)	f(x(j))	poly(j)	error	errorSq
6	-2	0	=$B$12+$B$13*A6+$B$14*A6^2	=B6-C6	=D6*D6
7	-1	1	=$B$12+$B$13*A7+$B$14*A7^2	=B7-C7	=D7*D7
8	0	2	=$B$12+$B$13*A8+$B$14*A8^2	=B8-C8	=D8*D8
9	1	1	=$B$12+$B$13*A9+$B$14*A9^2	=B9-C9	=D9*D9
10	2	0	=$B$12+$B$13*A10+$B$14*A10^2	=B10-C10	=D10*D10
11					
12	a(0)=	1	=58/35	=C12-B12	=SUM(E6:E10)
13	a(1)=	1	0	=C13-B13	
14	a(2)=	1	=-3/7	=C14-B14	
15					



	A	B	C	D	E
1	Fitting Polynomial of degree 2				
2	p(x)=a(0)+a(1)x+a(2)x*x				
3	to the data by Least Squares				
4					
5	x(j)	f(x(j))	poly(j)	error	errorSq
6	-2	0	3	-3	9
7	-1	1	1	0	0
8	0	2	1	1	1
9	1	1	3	-2	4
10	2	0	7	-7	49
11					
12	a(0)=	1	1.6571429	0.65714	63
13	a(1)=	1	0	-1	
14	a(2)=	1	-0.428571	-1.4286	
15					



	A	B	C	D	E
1	Fitting Polynomial of degree 2				
2	p(x)=a(0)+a(1)x+a(2)x*x				
3	to the data by Least Squares				
4					
5	x(j)	f(x(j))	poly(j)	error	errorSq
6	-2	0	-0.057143004	0.057143004	0.0033
7	-1	1	1.228571448	-0.228571448	0.0522
8	0	2	1.657142857	0.342857143	0.1176
9	1	1	1.228571225	-0.228571225	0.0522
10	2	0	-0.05714345	0.05714345	0.0033
11					
12	a(0)=	1.66	1.657142857	-7.2476E-11	0.2286
13	a(1)=	-0	0	1.11512E-07	
14	a(2)=	-0.4	-0.428571429	9.24087E-08	
15					



حل المعادلات التفاضلية العادية: The Solution of Ordinary Differential Equations

حل المعادلة التفاضلية العادية التي على الشكل:

$$y'(x) = f\left(x, y(x)\right), \quad y(x_0) = y_0$$

وتسمى بمشكلة القيمة الأولية Initial Value Problem وسوف نوجد الحل في الفترة المحدودة $[x_0, b]$ مبتدئين بالقيمة الأولية x_0 سوف نكتب $y_j = y(x_j)$ للسهولة.

1) صيغة أويلر Euler's Formula

$$k_1 = f\left(x_j, y_j\right)$$

$$k_2 = f\left(x_j + \frac{h}{2}, y_j + \frac{h}{2}k_1\right)$$

$$y_{j+1} = y_j + h\,k_2$$

2) طريقة هيون Heun Method

$$k_1 = f\left(x_j, y_j\right)$$

$$k_2 = f\left(x_j + \frac{2}{3}h, y_j + \frac{2}{3}h\,k_1\right)$$

$$y_{j+1} = y_j + h\left(\frac{1}{4}k_1 + \frac{3}{4}k_2\right)$$

3) صيغة رونج-كوتا Runge-Kutta Formula

$$k_1 = f\left(x_j, y_j\right)$$

$$k_2 = f\left(x_j + \frac{1}{2}h, y_j + \frac{1}{2}h\,k_1\right)$$

$$k_3 = f\left(x_j + \frac{1}{2}h, y_j + \frac{1}{2}h\,k_2\right)$$

$$k_4 = f\left(x_j + h, y_j + h\,k_3\right)$$

$$y_{j+1} = y_j + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

مثال: حل المعادلة

$$y' = x^2 + y^2, \quad y(0) = 0$$

في الفترة $[0,1]$.

الحل:

	A	B	C	D	E	F	G
1	Euler's Formula					b=	1
2	x(j)	y(j)	k(1,j)	k(2,j)		n=	10
3	0	0	=A3^2+B3^2	=(A3+($G$4/2))^2+(B3+($G$4/2)*C3)^2		x(0)=	0
4	=A3+$G$4	=B3+$G$4*D3	=A4^2+B4^2	=(A4+($G$4/2))^2+(B4+($G$4/2)*C4)^2		h=	=(G1-G3)/G2
5	=A4+$G$4	=B4+$G$4*D4	=A5^2+B5^2	=(A5+($G$4/2))^2+(B5+($G$4/2)*C5)^2			
6	=A5+$G$4	=B5+$G$4*D5	=A6^2+B6^2	=(A6+($G$4/2))^2+(B6+($G$4/2)*C6)^2			
7	=A6+$G$4	=B6+$G$4*D6	=A7^2+B7^2	=(A7+($G$4/2))^2+(B7+($G$4/2)*C7)^2			
8	=A7+$G$4	=B7+$G$4*D7	=A8^2+B8^2	=(A8+($G$4/2))^2+(B8+($G$4/2)*C8)^2			
9	=A8+$G$4	=B8+$G$4*D8	=A9^2+B9^2	=(A9+($G$4/2))^2+(B9+($G$4/2)*C9)^2			
10	=A9+$G$4	=B9+$G$4*D9	=A10^2+B10^2	=(A10+($G$4/2))^2+(B10+($G$4/2)*C10)^2			
11	=A10+$G$4	=B10+$G$4*D10	=A11^2+B11^2	=(A11+($G$4/2))^2+(B11+($G$4/2)*C11)^2			
12	=A11+$G$4	=B11+$G$4*D11	=A12^2+B12^2	=(A12+($G$4/2))^2+(B12+($G$4/2)*C12)^2			
13	=A12+$G$4	=B12+$G$4*D12	=A13^2+B13^2	=(A13+($G$4/2))^2+(B13+($G$4/2)*C13)^2			
14							

	A	B	C	D	E	F	G
1	Euler's Formula					b=	1
2	x(j)	y(j)	k(1,j)	k(2,j)		n=	10
3	0	0	0	0.0025		x(0)=	0
4	0.1	0.0003	0.01	0.0225		h=	0.1
5	0.2	0.0025	0.04	0.0625			
6	0.3	0.0088	0.0901	0.1227			
7	0.4	0.021	0.1604	0.2033			
8	0.5	0.0414	0.2517	0.3054			
9	0.6	0.0719	0.3652	0.4306			
10	0.7	0.115	0.5032	0.5821			
11	0.8	0.1732	0.67	0.7652			
12	0.9	0.2497	0.8723	0.9885			
13	1	0.3485	1.1215	1.2662			
14							

	A	B	C	D	E	F	G
1	Heun Method					b=	1
2	x(j)	y(j)	k(1,j)	k(2,j)		x(0)=	0
3	0	0	=A3^2+B3^2	=(A3+(2*$G$4)/3)^2+(B3+(2*$G$4*C3)/3)^2		n=	10
4	=A3+$G$4	=B3+$G$4*(C3/4+3*D3/4)	=A4^2+B4^2	=(A4+(2*$G$4)/3)^2+(B4+(2*$G$4*C4)/3)^2		h=	=(G1-G2
5	=A4+$G$4	=B4+$G$4*(C4/4+3*D4/4)	=A5^2+B5^2	=(A5+(2*$G$4)/3)^2+(B5+(2*$G$4*C5)/3)^2			
6	=A5+$G$4	=B5+$G$4*(C5/4+3*D5/4)	=A6^2+B6^2	=(A6+(2*$G$4)/3)^2+(B6+(2*$G$4*C6)/3)^2			
7	=A6+$G$4	=B6+$G$4*(C6/4+3*D6/4)	=A7^2+B7^2	=(A7+(2*$G$4)/3)^2+(B7+(2*$G$4*C7)/3)^2			
8	=A7+$G$4	=B7+$G$4*(C7/4+3*D7/4)	=A8^2+B8^2	=(A8+(2*$G$4)/3)^2+(B8+(2*$G$4*C8)/3)^2			
9	=A8+$G$4	=B8+$G$4*(C8/4+3*D8/4)	=A9^2+B9^2	=(A9+(2*$G$4)/3)^2+(B9+(2*$G$4*C9)/3)^2			
10	=A9+$G$4	=B9+$G$4*(C9/4+3*D9/4)	=A10^2+B10^2	=(A10+(2*$G$4)/3)^2+(B10+(2*$G$4*C10)/3)^2			
11	=A10+$G$4	=B10+$G$4*(C10/4+3*D10/4)	=A11^2+B11^2	=(A11+(2*$G$4)/3)^2+(B11+(2*$G$4*C11)/3)^2			
12	=A11+$G$4	=B11+$G$4*(C11/4+3*D11/4)	=A12^2+B12^2	=(A12+(2*$G$4)/3)^2+(B12+(2*$G$4*C12)/3)^2			
13	=A12+$G$4	=B12+$G$4*(C12/4+3*D12/4)	=A13^2+B13^2	=(A13+(2*$G$4)/3)^2+(B13+(2*$G$4*C13)/3)^2			
14							

	A	B	C	D	E	F	G
1	Heun Method					b=	1
2	x(j)	y(j)	k(1,j)	k(2,j)		x(0)=	0
3	0	0	0	0.0044		n=	10
4	0.1	0.0003333	0.01	0.0278		h=	0.1
5	0.2	0.0026667	0.04	0.0711			
6	0.3	0.0090024	0.0901	0.1347			
7	0.4	0.0213546	0.1605	0.2188			
8	0.5	0.0417764	0.2517	0.3245			
9	0.6	0.0724106	0.3652	0.4538			
10	0.7	0.1155772	0.5034	0.61			
11	0.8	0.1739125	0.6702	0.7989			
12	0.9	0.2505858	0.8728	1.0298			
13	1	0.3496395	1.1222	1.3179			
14							

	A	B	C	D
1	Runge-Kutta Formula			
2	x(j)	y(j)	k(1,j)	k(2,j)
3	0	0	=A3^2+B3^2	=(A3+$H$4/2)^2+(B3+$H$4*C3/2)^2
4	=A3+$H$4	=B3+$H$4*(C3+2*D3+2*E3+F3)/6	=A4^2+B4^2	=(A4+$H$4/2)^2+(B4+$H$4*C4/2)^2
5	=A4+$H$4	=B4+$H$4*(C4+2*D4+2*E4+F4)/6	=A5^2+B5^2	=(A5+$H$4/2)^2+(B5+$H$4*C5/2)^2
6	=A5+$H$4	=B5+$H$4*(C5+2*D5+2*E5+F5)/6	=A6^2+B6^2	=(A6+$H$4/2)^2+(B6+$H$4*C6/2)^2
7	=A6+$H$4	=B6+$H$4*(C6+2*D6+2*E6+F6)/6	=A7^2+B7^2	=(A7+$H$4/2)^2+(B7+$H$4*C7/2)^2
8	=A7+$H$4	=B7+$H$4*(C7+2*D7+2*E7+F7)/6	=A8^2+B8^2	=(A8+$H$4/2)^2+(B8+$H$4*C8/2)^2
9	=A8+$H$4	=B8+$H$4*(C8+2*D8+2*E8+F8)/6	=A9^2+B9^2	=(A9+$H$4/2)^2+(B9+$H$4*C9/2)^2
10	=A9+$H$4	=B9+$H$4*(C9+2*D9+2*E9+F9)/6	=A10^2+B10^2	=(A10+$H$4/2)^2+(B10+$H$4*C10/2)^2
11	=A10+$H$4	=B10+$H$4*(C10+2*D10+2*E10+F10)/6	=A11^2+B11^2	=(A11+$H$4/2)^2+(B11+$H$4*C11/2)^2
12	=A11+$H$4	=B11+$H$4*(C11+2*D11+2*E11+F11)/6	=A12^2+B12^2	=(A12+$H$4/2)^2+(B12+$H$4*C12/2)^2
13	=A12+$H$4	=B12+$H$4*(C12+2*D12+2*E12+F12)/6	=A13^2+B13^2	=(A13+$H$4/2)^2+(B13+$H$4*C13/2)^2
14				

	E	F	G	H
1			b=	1
2	k(3,j)	k(4,j)	x(0)=	0
3	=(A3+$H$4/2)^2+(B3+$H$4*D3/2)^2	=(A3+$H$4)^2+(B3+$H$4*E3)^2	n=	10
4	=(A4+$H$4/2)^2+(B4+$H$4*D4/2)^2	=(A4+$H$4)^2+(B4+$H$4*E4)^2	h=	=(H1-H2)/H3
5	=(A5+$H$4/2)^2+(B5+$H$4*D5/2)^2	=(A5+$H$4)^2+(B5+$H$4*E5)^2		
6	=(A6+$H$4/2)^2+(B6+$H$4*D6/2)^2	=(A6+$H$4)^2+(B6+$H$4*E6)^2		
7	=(A7+$H$4/2)^2+(B7+$H$4*D7/2)^2	=(A7+$H$4)^2+(B7+$H$4*E7)^2		
8	=(A8+$H$4/2)^2+(B8+$H$4*D8/2)^2	=(A8+$H$4)^2+(B8+$H$4*E8)^2		
9	=(A9+$H$4/2)^2+(B9+$H$4*D9/2)^2	=(A9+$H$4)^2+(B9+$H$4*E9)^2		
10	=(A10+$H$4/2)^2+(B10+$H$4*D10/2)^2	=(A10+$H$4)^2+(B10+$H$4*E10)^2		
11	=(A11+$H$4/2)^2+(B11+$H$4*D11/2)^2	=(A11+$H$4)^2+(B11+$H$4*E11)^2		
12	=(A12+$H$4/2)^2+(B12+$H$4*D12/2)^2	=(A12+$H$4)^2+(B12+$H$4*E12)^2		
13	=(A13+$H$4/2)^2+(B13+$H$4*D13/2)^2	=(A13+$H$4)^2+(B13+$H$4*E13)^2		
14				

	A	B	C	D	E	F	G	H
1	Runge-Kutta Formula						b=	1
2	x(j)	y(j)	k(1,j)	k(2,j)	k(3,j)	k(4,j)	x(0)=	0
3	0	0	0	0.0025	0.002500016	0.010000063	n=	10
4	0.1	0.000333335	0.010000111	0.022500694	0.022502127	0.040006675	h=	0.1
5	0.2	0.002666875	0.040007112	0.062521783	0.062533558	0.090079571		
6	0.3	0.009003498	0.090081063	0.122682454	0.122729148	0.160452686		
7	0.4	0.021359447	0.160456226	0.203363317	0.20349399	0.251739628		
8	0.5	0.041791288	0.251746512	0.305457034	0.305756316	0.365236971		
9	0.6	0.072448125	0.365248731	0.430728406	0.431333095	0.503359068		
10	0.7	0.115660305	0.503377306	0.582332855	0.583460365	0.670278207		
11	0.8	0.174081004	0.670304196	0.765596188	0.767597115	0.872921065		
12	0.9	0.250907869	0.872954758	0.989263005	0.992722749	1.122626133		
13	1	0.350233742	1.122663674	1.267634078	1.273577737	1.438093656		
14								

إيجاد تكامل بالمحكاة Monte Carlo Integration :

سوف نستخدم المحاكاة (أو مايسمى بطريقة مونت كارلو) لإيجاد تكاملات من الشكل:

$$I = \int_a^b f(x)\,dx$$

والتي نقربها بالمجموع

$$I \approx \frac{(b-a)}{N}\sum_{i=1}^{N} f(x_i)$$

وذلك بإختيار x_i بطريقة عشوائية بدلا من إختيارها عند نقاط محددة. الخطأ هو من درجة $1/\sqrt{N}$ وهو عشوائي تماما ولايعتمد على درجة التكامل (أبعاد التكامل).

مثال:

سوف نقدر عدديا قيمة π بالتكامل التالي:

$$\pi = \int_{-1}^{1}\int_{-1}^{1} f(x,y)\,dxdy$$

حيث

$$f(x,y) = \begin{cases} 1, & x^2+y^2 \le 1 \\ 0, & otherwise \end{cases}$$

أي

$$\pi \approx \frac{4}{N}\sum_{i=1}^{N} f(x_i,y_i), \quad f(x,y) = \begin{cases} 1, & x^2+y^2 \le 1 \\ 0, & otherwise \end{cases}$$

	A	B	C
1	=IF(((RAND())^2)+((RAND())^2)<=1,1,0)	=COUNT(A:A)	=SUM(A:A)
2	=IF(((RAND())^2)+((RAND())^2)<=1,1,0)		=(4/B1)*C1
3	=IF(((RAND())^2)+((RAND())^2)<=1,1,0)		

	A	B	C
1	1	25036	19657
2	1		3.14059754
3	0		

مثال آخر:

أوجد $f(x) = \int_0^4 \sqrt{x}\,dx$

الخطوات:

1- نولد n رقم عشوائي $x_1, x_2, \ldots, x_n$ بحيث $x_i \sim U(0,4)$ لجميع قيم i .

2- أوجد $\hat{f} = \frac{1}{n}\sum_{i=1}^{n} f(x_i) = \frac{1}{n}\sum_{i=1}^{n} \sqrt{x_i}$.

3- يقرب التكامل بالعلاقة

$$\int_a^b f(x)dx \approx (b-a)\times \hat{f}$$

$$\int_0^4 \sqrt{x}\,dx \approx (4-0)\times \hat{f} = 4\hat{f}$$

يقدر الخطأ بالعلاقة

$$error \approx (b-a)\sqrt{\frac{\hat{f}^2 - (\hat{f})^2}{n}}, \quad \hat{f}^2 = \frac{1}{n}\sum_{i=1}^{n} f^2(x_i)$$

في صفحة من Excel أدخل التالي:

1- في A1 أكتب n وأدخل الأرقام من 1 إلى 1500 في المجال A2-A1501 .

2- في B1 أكتب x وأدخل في B2 الصيغة (RAND())*4= وانسخها حتى B1501.

3- في C1 أكتب f(x) وأدخل في C2 الصيغة =SQRT(B2) .

4- في D1 أكتب fHat وادخل في D2 الصيغة =AVERAGE(C:C).

5- في E1 أكتب Integration وأدخل في E2 الصيغة =4*D2.

6- أحسب التكامل الحقيقي وأحسب الخطأ.

	A	B	C	D	E	F	G
1	n	x	f(x)	fHat	Integration	True Value	Error
2	1	2.731632	1.652765	1.332545	5.330181	5.33333333	0.003152
3	2	0.66217	0.813738				

تمارين:

1- أوجد تكامل $f(x) = \frac{4}{1+x^2}$ على المجال $x \in [0,1]$.

2- أوجد تكامل $f(x) = \sqrt{x+\sqrt{x}}$ على المجال $x \in [0,1]$.

مثال آخر: $f(x,y) = 4 - x^2 - y^2$

أوجد $\int_0^{5/4}\left(\int_0^{5/4}\left(4-x^2-y^2\right)dy\right)dx$.

الخطوات:

1- نولد n نقطة عشوائية $(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$ بحيث $x_i \sim U\left(0,5/4\right)$ و $y_i \sim U\left(0,5/4\right)$ لجميع قيم i .

2- أوجد $\hat{f} = \frac{1}{n}\sum_{i=1}^{n} f\left(x_i,y_i\right) = \frac{1}{n}\sum_{i=1}^{n}(4-x_i^2-y_i^2)$.

3- يقرب التكامل بالعلاقة

$$\int_a^b\int_c^d f\left(x,y\right)dydx \approx (b-a)\times(d-c)\times\hat{f}$$

$$\int_0^{5/4}\int_0^{5/4}\left(4-x^2-y^2\right)dydx \approx \left(\frac{5}{4}-0\right)\times\left(\frac{5}{4}-0\right)\times\hat{f} = \frac{25}{16}\hat{f}$$

يقدر الخطأ بالعلاقة

$$error \approx (b-a)\times(d-c)\sqrt{\frac{\hat{f}^2-\left(\hat{f}\right)^2}{n}},\quad \hat{f}^2 = \frac{1}{n}\sum_{i=1}^{n} f^2\left(x_i,y_i\right)$$

في صفحة من Excel أدخل التالي:

1- في A1 أكتب n وأدخل الأرقام من 1 إلى 1500 في المجال A2-A1501 .
2- في B1 أكتب x وأدخل في B2 الصيغة (RAND())*(5/4)= وانسخها حتى B1501.
3- في C1 أكتب y وأدخل في C2 الصيغة (RAND())*(5/4)= وانسخها حتى C1501.
4- في D1 أكتب f(x) وأدخل في D2 الصيغة =4-B2*B2-C2*C2 .
5- في E1 أكتب fHat وادخل في E2 الصيغة =AVERAGE(D:D).
6- في E1 أكتب Integration وأدخل في E2 الصيغة =(5/4)*(5/4)*D2.
7- أحسب التكامل الحقيقي وأحسب الخطأ.

	A	B	C	D	E	F	G	H
1	n	x	y	f(x,y)	fHat	Integration	TrueValue	Error
2	1	0.59825	0.100495	3.631997	2.958536	4.622712	4.62298	0.000268
3	2	1.235862	0.640263	2.062707				

تمارين:

1- إذا كانت $f\left(x,y\right) = \sqrt{4-x^2-y^2}$ على المجال $x \in [0,5/4], y \in [0,5/4]$.

2- إذا كانت $f\left(x,y,z\right) = 4-x^2-y^2-z^2$

فأوجد

$$\int_0^{9/10}\left(\int_0^{1}\left(\int_0^{11/10}\left(4-x^{2}-y^{2}-z^{2}\right)dz\right)dy\right)dx$$

القيمة الحقيقية =2.9634 .

3- إذا كانت $f\left(x,y,z,u\right)=5-x^{2}-y^{2}-z^{2}-u^{2}$ فأوجد

$$\int_0^{4/5}\left(\int_0^{9/10}\left(\int_0^{1}\left(\int_0^{11/10}\left(5-x^{2}-y^{2}-z^{2}-u^{2}\right)du\right)dz\right)dy\right)dx$$

القيمة الحقيقية =2.99663 .

<u>مثال آخر:</u>

إيجاد التكامل للدالة $f\left(z\right)=\frac{1}{\sqrt{2\pi}}e^{-z^{2}/2}$ على المجال $z\in\left[-4,0\right]$.

في صفحة من Excel أدخل البيانات كماهو موضح

	A	B	C	D	E	F
1	n	x	f(x)	fHat	TrueValue	Error
2	1	=4*RAND()	=(1/SQRT(2*PI()))*EXP(-(B2*B2)/2)	=4*AVERAGE(C:C)	0.5	=E2-D2

الناتج:

	A	B	C	D	E	F
1	n	x	f(x)	fHat	TrueValue	Error
2	1	2.344258	0.02556	0.500305	0.5	-0.00031
3	2	0.463459	0.358318			

ملاحظة: التكامل السابق هو $P\left(z<0\right)$ حيث $z\sim N\left(0,1\right)$ على المجال $z\in\left(-\infty,0\right)$. (أخذنا 4– كمالانهاية لأن التكامل من $-\infty$ إلى 4– صغير جدا ويمكن إهماله).

الفصل الثاني

إستخدام EXCEL SOLVER في حل مسائل البرمجة الرياضية:

في هذا الفصل سوف نستخدم EXCEL SOLVER لحل مسائل في البرمجة الخطية والعددية والتربيعية والحركية. سوف نستعرض ذلك على مسائل متنوعة من مختلف المراجع المذكورة آخر الكتاب.

مثال 1:

مثال على البرمجة الخطية:

شركة تصنيع العاب أطفال تصنع نوعين من الألعاب S و Z في حزم تتكون من 12 لعبة وتصنع من نوع خاص من البلاستيك. الموارد المتاحة هي 1000 رطل من البلاستيك و 40 ساعة زمن الإنتاج. قدرة الشركة على تسويق المنتجات لاتزيد عن 700 حزمة إنتاجية. الطلب على النوع الأول من اللعب أكثر من النوع الثاني على الايزيد عن 350 حزمة. الجدول التالي يعطي متطلبات الموارد وقيم الأرباح:

Product	Profit per Dozen	Plastic (lb.) per Dozen	Production Time (min.) per Dozen
S	$8	2	3
Z	$5	1	4

النموذج:

$$
\begin{aligned}
Max \quad & 8x_1 + 5x_2 \\
St \quad & 2x_1 + x_2 \le 1000 \\
& 3x_1 + 4x_2 \le 2400 \\
& x_1 + x_2 \le 700 \\
& x_1 - x_2 \le 350 \\
& x_1, x_2 \ge 0
\end{aligned}
$$

الحل بواسطة EXCEL SOLVER:

أدخل البيانات التالية:

	A	B	C	D	E	F
1		S	Z			
2	Dozen	0	0			
3				Total		Limit
4	Profit	8	5	0		
5	Plastic	2	1	0	<=	1000
6	Prod. Time	3	4	0	<=	2400
7	Total	1	1	0	<=	700
8	Mix	1	-1	0	<=	350

في الخلية D4 أدخل التالي:

=SUMPRODUCT($B$2:$C$2,B4:C4)

ثم إسحبها للخلايا D5:D8 فينتج:

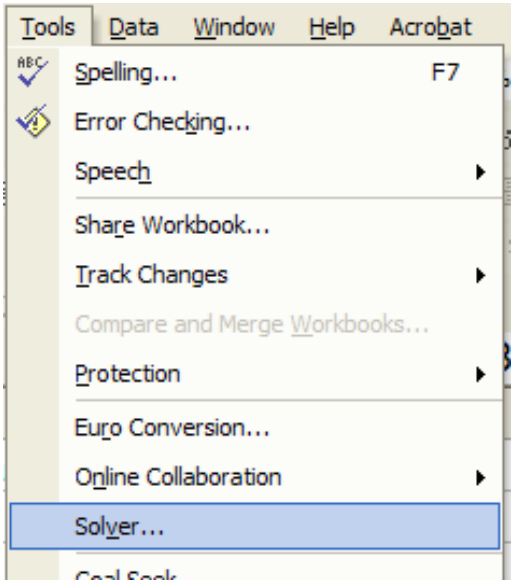
D4 fx =SUMPRODUCT($B$2:$C$2,B4:C4)

	A	B	C	D	E	F
1		S	Z			
2	Dozen	0	0			
3				Total		Limit
4	Profit	8	5	0		
5	Plastic	2	1	0	<=	1000
6	Prod. Time	3	4	0	<=	2400
7	Total	1	1	0	<=	700
8	Mix	1	-1	0	<=	350

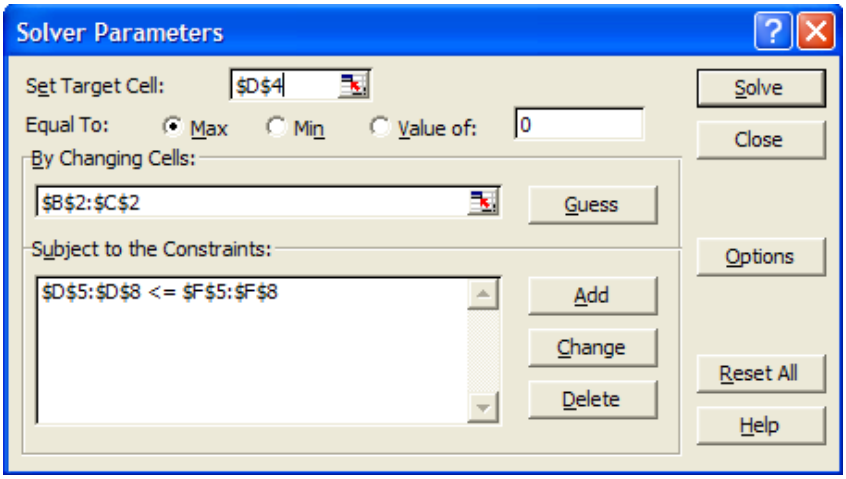
وفي شكل الصيغ

	A	B	C	D	E	F
1		S	Z			
2	Dozen	0	0			
3				Total		Limits
4	Profit	8	5	=SUMPRODUCT($B$2:$C$2,B4:C4)	<=	
5	Plastic	2	1	=SUMPRODUCT($B$2:$C$2,B5:C5)	<=	1000
6	Prod.Time	3	4	=SUMPRODUCT($B$2:$C$2,B6:C6)	<=	2400
7	Total	1	1	=SUMPRODUCT($B$2:$C$2,B7:C7)	<=	700
8	Mix	1	-1	=SUMPRODUCT($B$2:$C$2,B8:C8)	<=	350
9						

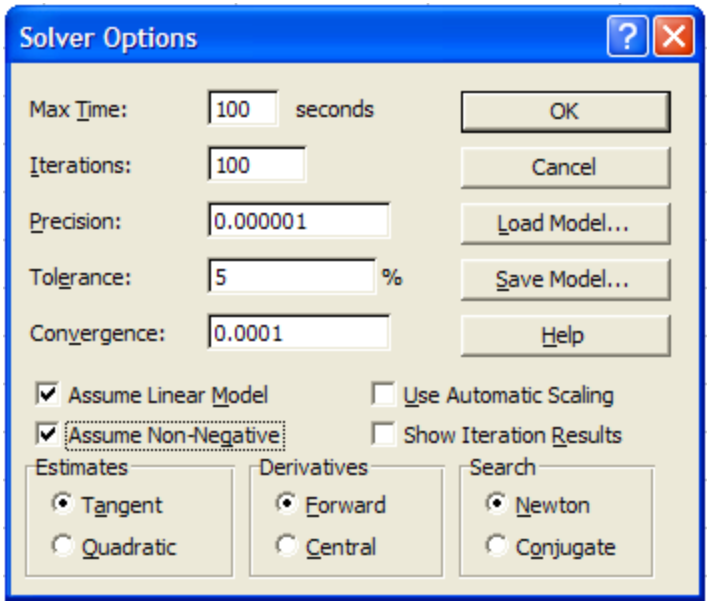
ضع المؤشر في الخلية D4 ثم من القائمة الرئيسة أختار



فتظهر النافذة



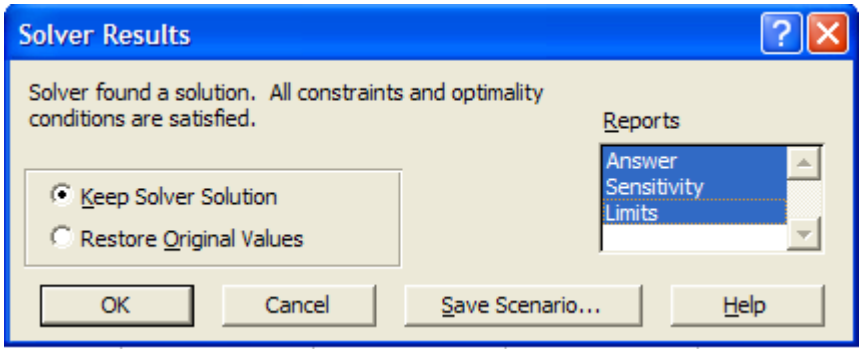
أضغط على Options فتظهر النافذة



تأكد من إختيار Assume Linear Model و Assume Non-Negative ثم أضغط OK ثم Solve فينتج

	A	B	C	D	E	F
1		S	Z			
2	Dozen	320	360			
3				Total		Limit
4	Profit	8	5	4360		
5	Plastic	2	1	1000	<=	1000
6	Prod. Time	3	4	2400	<=	2400
7	Total	1	1	680	<=	700
8	Mix	1	-1	-40	<=	350

ملاحظة: عند إنتهاء Solver تظهر نافذة



أختار تحت Reports الموضح بالتظليل فتظهر ثلاثة صفحات جديدة أحدها للإجابة و هي:

Microsoft Excel 11.0 Answer Report
Worksheet: [Book1]Sheet1
Report Created: 9/17/2005 9:54:04 PM

Target Cell (Max)

Cell	Name	Original Value	Final Value
$D$4	Profit Total	0	4360

Adjustable Cells

Cell	Name	Original Value	Final Value
$B$2	Dozen S	0	320
$C$2	Dozen Z	0	360

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$D$5	Plastic Total	1000	$D$5<=$F$5	Binding	0
$D$6	Prod. Time Total	2400	$D$6<=$F$6	Binding	0
$D$7	Total Total	680	$D$7<=$F$7	Not Binding	20
$D$8	Mix Total	-40	$D$8<=$F$8	Not Binding	390

الثانية لتحليل الحساسية

Microsoft Excel 11.0 Sensitivity Report
Worksheet: [Book1]Sheet1
Report Created: 9/17/2005 9:54:04 PM

Adjustable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$B$2	Dozen S	320	0	8	2	4.25
$C$2	Dozen Z	360	0	5	5.666666667	1

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$D$5	Plastic Total	1000	3.4	1000	100	400
$D$6	Prod. Time Total	2400	0.4	2400	100	650
$D$7	Total Total	680	0	700	1E+30	20
$D$8	Mix Total	-40	0	350	1E+30	390

والثالثة للنهايات

Microsoft Excel 10.0 Limits Report
Worksheet: [Book1]Limits Report 1
Report Created: 2/28/2003 8:42:28 PM

Cell	Target Name	Value
$D$4	Profit Total	4360

Cell	Adjustable Name	Value	Lower Limit	Target Result	Upper Limit	Target Result
$B$2	Dozen Space Rays	320	0	1800	320	4360
$C$2	Dozen Zappers	360	0	2560	360	4360

تمرين: من دراستك للبرمجة الخطية تحقق من الحل وفسر النتائج السابقة.

مثال 2:

مثال آخر على البرمجة الخطية:
بإستخدام EXCEL SOLVER حل التالي:

$$\begin{aligned} &Minimize && 0.6X_1 + 0.5X_2 \\ &ST && 20X_1 + 50X_2 \geq 100 \\ & && 25X_1 + 25X_2 \geq 100 \\ & && 50X_1 + 10X_2 \geq 100 \\ & && X_1, X_2 \geq 0 \end{aligned}$$

الحل:
جهز صفحة نشر كالتالي:

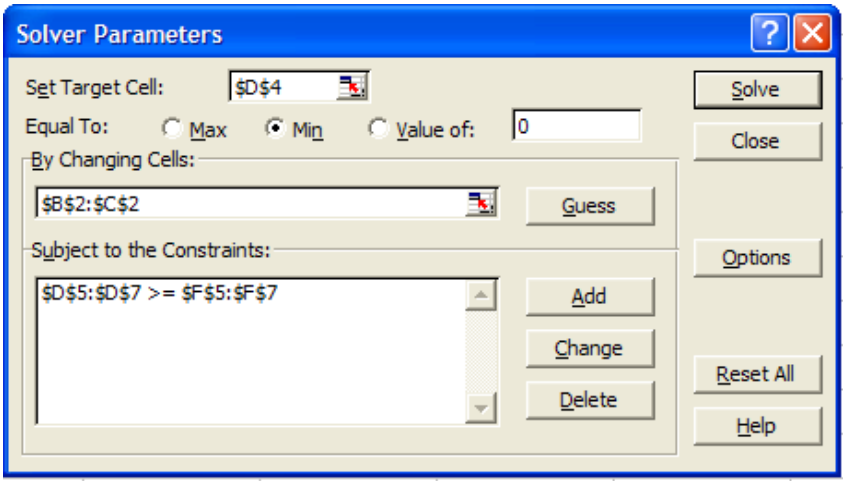
D4 | =SUMPRODUCT($B$2:$C$2,B4:C4)

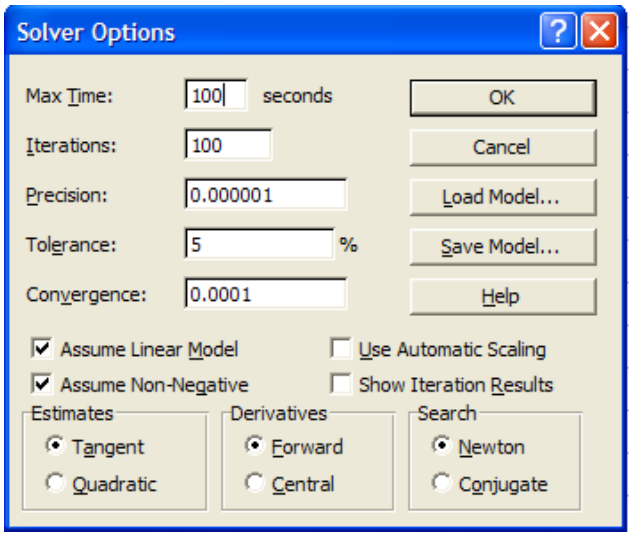
	A	B	C	D	E	F
1		X1	X2			
2	DecisionVar	0	0			
3				Total		
4	Objective	0.6	0.5	0		
5	Const. 1	20	50	0	>=	100
6	Const. 2	25	25	0	>=	100
7	Const. 3	50	10	0	>=	100

ويظهر في شكل الصيغ

	A	B	C	D	E	F
1		X1	X2			
2	DecisionVar	0	0			
3				Total		
4	Objective	0.6	0.5	=SUMPRODUCT($B$2:$C$2,B4:C4)		
5	Const. 1	20	50	=SUMPRODUCT($B$2:$C$2,B5:C5)	>=	100
6	Const. 2	25	25	=SUMPRODUCT($B$2:$C$2,B6:C6)	>=	100
7	Const. 3	50	10	=SUMPRODUCT($B$2:$C$2,B7:C7)	>=	100
8						

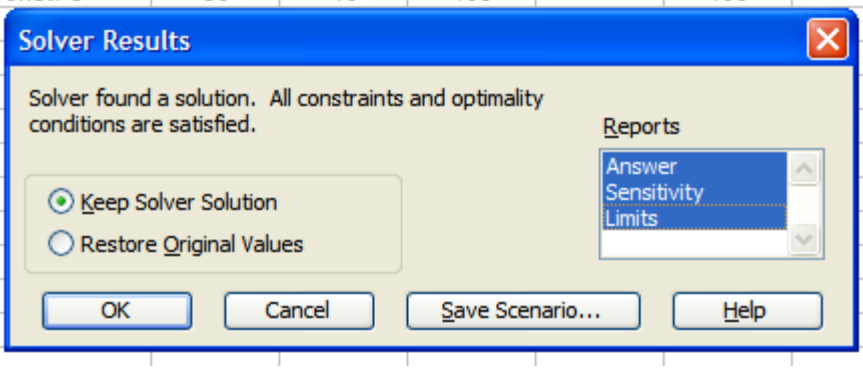
بوضع المؤشر في الخلية D4 وإختيار Solver من قائمة المواد تظهر النافذة التالية فنملؤها :


Solver Parameters
Set Target Cell:
$D$4
Equal To:
Max
Min
Value of:
0
By Changing Cells:
$B$2:$C$2
Guess
Subject to the Constraints:
$D$5:$D$7 >= $F$5:$F$7
Add
Change
Delete
Solve
Close
Options
Reset All
Help


Solver Options
Max Time:
100
seconds
Iterations:
100
Precision:
0.000001
Tolerance:
5
%
Convergence:
0.0001
OK
Cancel
Load Model...
Save Model...
Help
Assume Linear Model
Use Automatic Scaling
Assume Non-Negative
Show Iteration Results
Estimates
Tangent
Quadratic
Derivatives
Forward
Central
Search
Newton
Conjugate

D4 =SUMPRODUCT($B$2:$C$2,B4:C4)

	A	B	C	D	E	F	G
1		X1	X2				
2	DecisionVar	1.5	2.5				
3				Total			
4	Objective	0.6	0.5	2.15			
5	Const. 1	20	50	155	>=	100	
6	Const. 2	25	25	100	>=	100	
7	Const. 3	50	10	100	>=	100	



Microsoft Excel 10.0 Answer Report
Worksheet: [Book2]Sheet1
Report Created: 2/28/2003 9:21:11 PM

Target Cell (Min)

Cell	Name	Original Value	Final Value
$D$4	Objective Total	0	2.15

Adjustable Cells

Cell	Name	Original Value	Final Value
$B$2	Decision Var. X1	0	1.5
$C$2	Decision Var. X2	0	2.5

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$D$5	Const. 1 Total	155	$D$5>=$F$5	Not Binding	55
$D$6	Const. 2 Total	100	$D$6>=$F$6	Binding	0
$D$7	Const. 3 Total	100	$D$7>=$F$7	Binding	0

Microsoft Excel 10.0 Sensitivity Report
Worksheet: [Book2]Sheet1
Report Created: 2/28/2003 9:21:12 PM

Adjustable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$B$2	Decision Var. X1	1.5	0	0.6	1.9	0.1
$C$2	Decision Var. X2	2.5	0	0.5	0.1	0.38

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$D$5	Const. 1 Total	155	0	100	55	1E+30
$D$6	Const. 2 Total	100	0.019	100	150	23.91304348
$D$7	Const. 3 Total	100	0.0025	100	73.33333333	60

Microsoft Excel 10.0 Limits Report
Worksheet: [Book2]Limits Report 1
Report Created: 2/28/2003 9:21:12 PM

Cell	Target Name	Value
$D$4	Objective Total	2.15

Cell	Adjustable Name	Value	Lower Limit	Target Result	Upper Limit	Target Result
$B$2	Decision Var. X1	1.5	1.5	2.15	#N/A	#N/A
$C$2	Decision Var. X2	2.5	2.5	2.15	#N/A	#N/A

تمرين: يترك للطالب التحقق من الحل وتفسير النتائج.

تمرين: حل التالي بإستخدام Excel Solver

Minimize $18x_1 + 22x_2 + 10x_3 + 12x_4 + 10x_5 + 9x_6 + 40x_7 + 16x_8 + 50x_9 + 7x_{10}$

subject to

$$90x_1 + 110x_2 + 100x_3 + 90x_4 + 75x_5 + 35x_6 + 65x_7 + 100x_8 + 120x_9 + 65x_{10} \geq 420$$

$$6x_1 + 4x_2 + 2x_3 + 3x_4 + x_5 + x_7 + x_{10} \geq 5$$

$$20x_1 + 48x_2 + 12x_3 + 8x_4 + 30x_5 + 52x_7 + 250x_8 + 3x_9 + 26x_{10} \geq 400$$

$$3x_1 + 4x_2 + 5x_3 + 6x_4 + 7x_5 + 2x_6 + x_7 + 9x_8 + x_9 + 3x_{10} \geq 20$$

$$5x_1 + 2x_2 + 3x_3 + 4x_4 + x_7 + 3x_{10} \geq 12$$

$$2x_2 + 2x_3 + 2x_4 + 5x_5 + 3x_6 + 4x_8 + x_{10} \leq 20$$

$$270x_5 + 8x_6 + 12x_8 \leq 30$$

$$x_i \geq 0$$

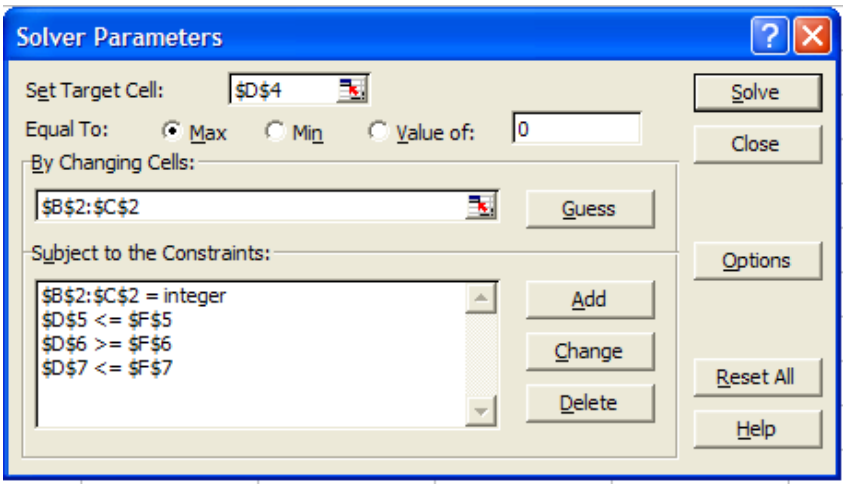
مثال 3:

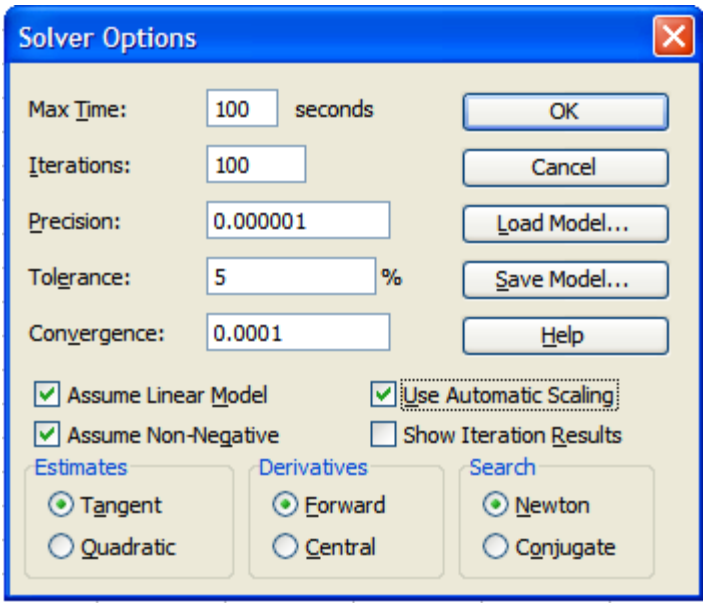
البرمجة العددية الصحيحة Integer Programming

حل التالي:

$$\begin{aligned} Max. \quad & 12000X_1 + 20000X_2 \\ St. \quad & 2X_1 + 6X_2 \le 27 \\ & X_2 \ge 2 \\ & 3X_1 + X_2 \le 19 \\ & X_1, X_2 \ge 0 \text{ and integers} \end{aligned}$$

	A	B	C	D	E	F
1		X1	X2			
2	Decision V	0	0			
3				Total		
4	Objective	12000	20000	0		
5	Const. 1	2	6	0	<=	27
6	Const. 2	0	1	0	>=	2
7	Const. 3	3	1	0	<=	19





لاحظ إختيار Use Automatic Scaling وذلك لإختلاف درجة الارقام حيث دالة الهدف تحوي ارقام من درجة الآلاف والقيود من درجة الآحاد.
وينتج الحل:

	A	B	C	D	E	F
1		X1	X2			
2	Decision V	4	3			
3				Total		
4	Objective	12000	20000	108000		
5	Const. 1	2	6	26	<=	27
6	Const. 2	0	1	3	>=	2
7	Const. 3	3	1	15	<=	19

Microsoft Excel 10.0 Answer Report
Worksheet: [Book2]Sheet1
Report Created: 2/28/2003 9:51:06 PM

Target Cell (Max)

Cell	Name	Original Value	Final Value
$D$4	Objective Total	0	108000

Adjustable Cells

Cell	Name	Original Value	Final Value
$B$2	Decision Var. X1	0	4
$C$2	Decision Var. X2	0	3

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$D$5	Const. 1 Total	26	$D$5<=$F$5	Not Binding	1
$D$6	Const. 2 Total	3	$D$6>=$F$6	Not Binding	1
$D$7	Const. 3 Total	15	$D$7<=$F$7	Not Binding	4
$B$2	Decision Var. X1	4	$B$2=integer	Binding	0
$C$2	Decision Var. X2	3	$C$2=integer	Binding	0

تمرين: من دراستك للبرمجة العددية تحقق من النتائج وفسرها.

تمرين : حل التالي وتحقق من النتائج.

$$\textit{Minimize} \quad 8x_1 + 10x_2 + 7x_3 + 6x_4 + 11x_5 + 9x_6$$

$$\textit{Subject to}$$

$$12x_1 + 9x_2 + 25x_3 + 20x_4 + 17x_5 + 13x_6 \geq 60$$

$$35x_1 + 42x_2 + 18x_3 + 31x_4 + 56x_5 + 49x_6 \geq 150$$

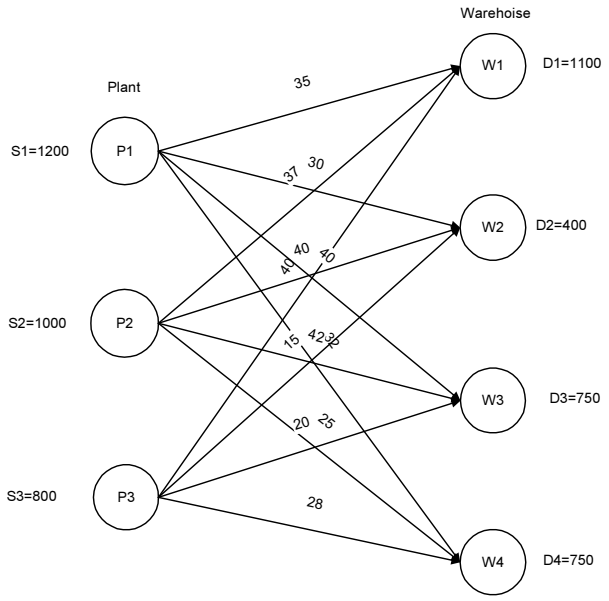
$$37x_1 + 53x_2 + 28x_3 + 24x_4 + 29x_5 + 20x_6 \geq 125$$

$$0 \leq x_1, x_2 \leq 1$$

مثال 4:

مثال على الشبكات والنقل:

حل مشكلة النقل للشبكة التالية:



والتي نشكلها على شكل البرمجة الخطية التالية

$Min\left(\text{Total Shipping Cost}\right)$

St :

Amount shipped from each source $\leq$ Supply at that source

Amount received at each destination = Demand at that destination

No negative shipments

إذا رمزنا بـ X_{ij} لعدد الوحدات التي تشحن من المصنع $i=1,2,3$ إلى المخزن $j=1,2,3,4$ فتكتب المشكلة على الشكل:

$$Min\ 35X_{11}+20X_{12}+40X_{13}+32X_{14}+37X_{21}+40X_{22}+42X_{23}+25X_{24}+40X_{31}+15X_{32}+20X_{33}+28X_{34}$$

$$St\quad X_{11}+X_{12}+X_{13}+X_{14} \leq 1200$$

$$X_{21}+X_{22}+X_{23}+X_{24} \leq 1000$$

$$X_{31}+X_{32}+X_{33}+X_{34} \leq 800$$

$$X_{11}+X_{21}+X_{31}=1100$$

$$X_{12}+X_{22}+X_{32}=400$$

$$X_{13}+X_{23}+X_{33}=750$$

$$X_{14}+X_{24}+X_{34}=750$$

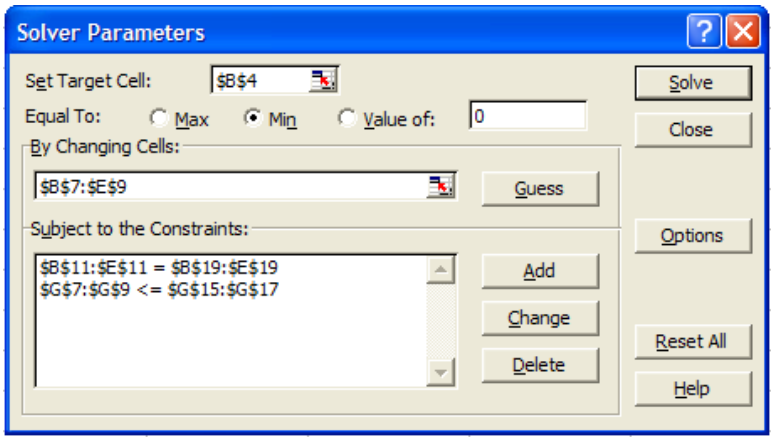
$$X_{ij} \geq 0,\quad for\ all\ i\ and\ j$$

	A	B	C	D	E	F	G
1	Network Transportation Problem						
2							
3	Solution						
4	Minimum Cost	0					
5			Shipments (Cases)				
6		P1	P2	P3	P4		Shipped
7	W1						0
8	W2						0
9	W3						0
10							
11	Received	0	0	0	0		
12							
13	Input						
14		P1	P2	P3	P4		Supply
15	W1	35	30	40	32		1200
16	W2	37	40	42	25		1000
17	W3	40	15	20	28		800
18							
19	Demand	1100	400	750	750		

شكل الصيغ

	A	B	C	D	E	F	G
1	Network Trans						
2							
3	Solution						
4	Minimum Cost	=SUMPRODUCT(B7:E9,B15:E17)					
5			Shipments (Cases)				
6		P1	P2	P3	P4		Shipped
7	W1	850	350	0	0		=SUM(B7:E7)
8	W2	250	0	0	750		=SUM(B8:E8)
9	W3	0	50	750	0		=SUM(B9:E9)
10							
11	Received	=SUM(B7:B9)	=SUM(C7:C9)	=SUM(D7:D9)	=SUM(E7:E9)		
12							
13	Input						
14		P1	P2	P3	P4		Supply
15	W1	35	30	40	32		1200
16	W2	37	40	42	25		1000
17	W3	40	15	20	28		800
18							
19	Demand	1100	400	750	750		
20							

وبإستخدام Solver



الحل يكون:

	A	B	C	D	E	F	G
1	Network Transportation Problem						
2							
3	Solution						
4	Minimum Cost	84000					
5			Shipments (Cases)				
6		P1	P2	P3	P4		Shipped
7	W1	850	350	0	0		1200
8	W2	250	0	0	750		1000
9	W3	0	50	750	0		800
10							
11	Received	1100	400	750	750		
12							
13	Input						
14		P1	P2	P3	P4		Supply
15	W1	35	30	40	32		1200
16	W2	37	40	42	25		1000
17	W3	40	15	20	28		800
18							
19	Demand	1100	400	750	750		

والتقرير:

Microsoft Excel 10.0 Answer Report
Worksheet: [Book3]Sheet1
Report Created: 2/28/2003 11:27:54 PM

Target Cell (Min)

Cell	Name	Original Value	Final Value
$B$4	Minimum Cost	0	84000

Adjustable Cells

Cell	Name	Original Value	Final Value
$B$7	W1 P1	0	850
$C$7	W1 P2	0	350
$D$7	W1 P3	0	0
$E$7	W1 P4	0	0
$B$8	W2 P1	0	250
$C$8	W2 P2	0	0
$D$8	W2 P3	0	0
$E$8	W2 P4	0	750
$B$9	W3 P1	0	0
$C$9	W3 P2	0	50
$D$9	W3 P3	0	750
$E$9	W3 P4	0	0

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$B$11	Received P1	1100	$B$11=$B$19	Not Binding	0
$C$11	Received P2	400	$C$11=$C$19	Not Binding	0
$D$11	Received P3	750	$D$11=$D$19	Not Binding	0
$E$11	Received P4	750	$E$11=$E$19	Not Binding	0
$G$7	W1 Shipped	1200	$G$7<=$G$15	Binding	0
$G$8	W2 Shipped	1000	$G$8<=$G$16	Binding	0
$G$9	W3 Shipped	800	$G$9<=$G$17	Binding	0

Microsoft Excel 10.0 Sensitivity Report
Worksheet: [Book3]Sheet1
Report Created: 2/28/2003 11:27:54 PM

Adjustable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$B$7	W1 P1	850	0	35	2	5
$C$7	W1 P2	350	0	30	5	17
$D$7	W1 P3	0	5	40	1E+30	5
$E$7	W1 P4	0	9	32	1E+30	9
$B$8	W2 P1	250	0	37	5	2
$C$8	W2 P2	0	8	40	1E+30	8
$D$8	W2 P3	0	5	42	1E+30	5
$E$8	W2 P4	750	0	25	9	1E+30
$B$9	W3 P1	0	20	40	1E+30	20
$C$9	W3 P2	50	0	15	17	5
$D$9	W3 P3	750	0	20	5	1E+30
$E$9	W3 P4	0	20	28	1E+30	20

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$B$11	Received P1	1100	37	1100	0	250
$C$11	Received P2	400	32	400	0	250
$D$11	Received P3	750	37	750	0	250
$E$11	Received P4	750	25	750	0	750
$G$7	W1 Shipped	1200	-2	1200	250	0
$G$8	W2 Shipped	1000	0	1000	1E+30	0
$G$9	W3 Shipped	800	-17	800	250	0

Microsoft Excel 10.0 Limits Report
Worksheet: [Book3]Limits Report 1
Report Created: 2/28/2003 11:27:54 PM

Cell	Target Name	Value
$B$4	Minimum Cost	84000

Cell	Adjustable Name	Value	Lower Limit	Target Result	Upper Limit	Target Result
$B$7	W1 P1	850	850	84000	850	84000
$C$7	W1 P2	350	350	84000	350	84000
$D$7	W1 P3	0	0	84000	0	84000
$E$7	W1 P4	0	0	84000	0	84000
$B$8	W2 P1	250	250	84000	250	84000
$C$8	W2 P2	0	0	84000	0	84000
$D$8	W2 P3	0	0	84000	0	84000
$E$8	W2 P4	750	750	84000	750	84000
$B$9	W3 P1	0	0	84000	0	84000
$C$9	W3 P2	50	50	84000	50	84000
$D$9	W3 P3	750	750	84000	750	84000
$E$9	W3 P4	0	0	84000	0	84000

تمرين: يترك للطالب التحقق من النتائج وتفسير المخرجات.

مثال 5:

مثال آخرلمشكلة نقل Transportation Problem

لشركة مصنعين لإنتاج معين والذي يتم توزيعه عن طريق ثلاثة مراكز توزيع. سعر إنتاج الوحدة متساوي في جميع المصانع ولكن تكلفة النقل من المصانع إلى مراكز التوزيع تختلف حسب البيانات في الجدول التالي:

	Distribution Center 1	Distribution Center 2	Distribution Center 3
Plant A	4	6	4
Plant B	6	5	2

عملية النقل تتم مرة في الأسبوع. خلال الأسبوع كل مصنع ينتج على الأكثر 60 وحدة وكل مركز توزيع يحتاج 40 وحدة للتوزيع.

المطلوب تحديد عدد الوحدات التي تنقل من كل مصنع إلى مراكز التوزيع بحيث تكون تكلفة النقل أقل مايمكن؟

الحل:

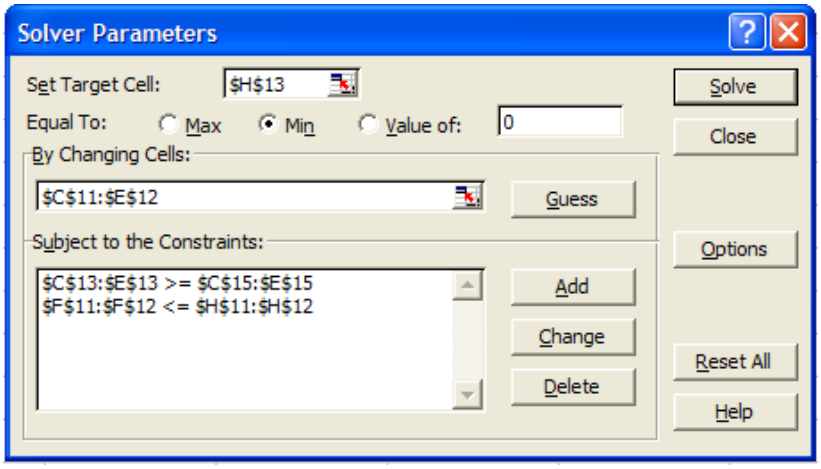
جهز صفحة نشر كالتالي:

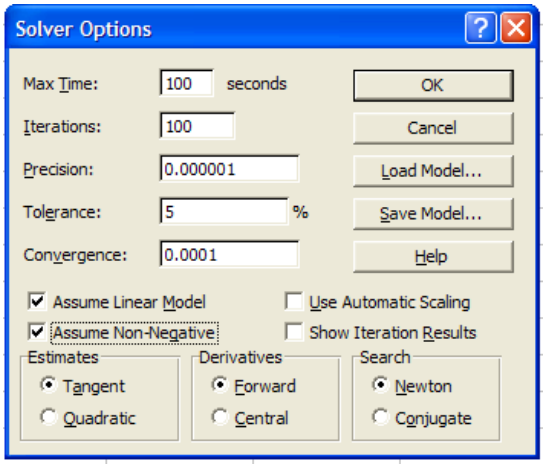
	A	B	C	D	E	F	G	H
1	A Distribution Problem							
2								
3			Distribution	Distribution	Distribution			
4		**Cost**	Center 1	Center 2	Center 3			
5		Plant A	4	6	4			
6		Plant B	6	5	2			
7								
8								
9		**Shipment**	Distribution	Distribution	Distribution			
10		**Quantities**	Center 1	Center 2	Center 3	Shipped		Available
11		Quantities	40	20	0	60	<=	60
12		Plant B	0	20	40	60	<=	60
13		Shipped	40	40	40	cost	=	460
14			>=	>=	>=			
15		Needed	40	40	40			

شكل الصيغ

	B	C	D	E	F	G	H
1							
2							
3		Distribution	Distribution	Distribution			
4	**Cost**	Center 1	Center 2	Center 3			
5	Plant A	4	6	4			
6	Plant B	6	5	2			
7							
8							
9	**Shipment**	Distribution	Distribution	Distribution			
10	**Quantities**	Center 1	Center 2	Center 3	Shipped		Available
11	Quantities				=SUM(C11:E11)	<=	60
12	Plant B				=SUM(C12:E12)	<=	60
13	Shipped	=SUM(C11:C12)	=SUM(D11:D12)	=SUM(E11:E12)	cost	=	=SUMPRODUCT(C5:E6,C11:E12)
14		>=	>=	>=			
15	Needed	40	40	40			

أفتح SOLVER وعبئ البيانات كالتالي:





وبالضغط على Solve ينتج الحل

	A	B	C	D	E	F	G	H
1	A	A Distri	A Distribution Problem					
2								
3			Distribution	Distribution	Distribution			
4		**Cost**	Center 1	Center 2	Center 3			
5		Plant A	4	6	4			
6		Plant B	6	5	2			
7								
8								
9		**Shipment**	Distribution	Distribution	Distribution			
10		**Quantities**	Center 1	Center 2	Center 3	Shipped		Available
11		Quantities	40	20	0	60	<=	60
12		Plant B	0	20	40	60	<=	60
13		Shipped	40	40	40	cost	=	460
14			>=	>=	>=			
15		Needed	40	40	40			

تقرير الحل

Microsoft Excel 10.0 Answer Report
Worksheet: [Book1]Sheet1
Report Created: 7/6/2003 6:36:12 PM

Target Cell (Min)

Cell	Name	Original Value	Final Value
$H$13	= Avaolable	0	460

Adjustable Cells

Cell	Name	Original Value	Final Value
$C$11	Quantities Center 1	0	40
$D$11	Quantities Center 2	0	20
$E$11	Quantities Center 3	0	0
$C$12	Plant B Center 1	0	0
$D$12	Plant B Center 2	0	20
$E$12	Plant B Center 3	0	40

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$C$13	Shipped Center 1	40	$C$13>=$C$15	Binding	0
$D$13	Shipped Center 2	40	$D$13>=$D$15	Binding	0
$E$13	Shipped Center 3	40	$E$13>=$E$15	Binding	0
$F$11	Quantities Shipped	60	$F$11<=$H$11	Binding	0
$F$12	Plant B Shipped	60	$F$12<=$H$12	Binding	0

تقرير الحساسية

Microsoft Excel 10.0 Sensitivity Report
Worksheet: [Book1]Sheet1
Report Created: 7/6/2003 6:36:12 PM

Adjustable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$C$11	Quantities Center 1	40	0	4	3	4
$D$11	Quantities Center 2	20	0	6	1	1
$E$11	Quantities Center 3	0	1	4	1E+30	1
$C$12	Plant B Center 1	0	3	6	1E+30	3
$D$12	Plant B Center 2	20	0	5	1	1
$E$12	Plant B Center 3	40	0	2	1	3

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$C$13	Shipped Center 1	40	4	40	0	40
$D$13	Shipped Center 2	40	6	40	0	20
$E$13	Shipped Center 3	40	3	40	0	20
$F$11	Quantities Shipped	60	0	60	1E+30	0
$F$12	Plant B Shipped	60	-1	60	20	0

تقرير النهايات

Microsoft Excel 10.0 Limits Report
Worksheet: [Book1]Limits Report 1
Report Created: 7/6/2003 6:36:12 PM

Cell	Target Name	Value
$H$13	= Avaolable	460

Cell	Adjustable Name	Value	Lower Limit	Target Result	Upper Limit	Target Result
$C$11	Quantities Center 1	40	40	460	40	460
$D$11	Quantities Center 2	20	20	460	20	460
$E$11	Quantities Center 3	0	0	460	0	460
$C$12	Plant B Center 1	0	0	460	0	460
$D$12	Plant B Center 2	20	20	460	20	460
$E$12	Plant B Center 3	40	40	460	40	460

تمرين: يترك للطالب التحقق من النتائج وتفسير المخرجات.

مثال6:

مثال على التخصيص (التحديد) Assignment Problem
يريد أحد المدربين لفريق سباحة تحديد سباحين لسباق مسابقة 200 متر سباحة متنوعة متتابعة لاربعة سباحين فقط كل منهم يسبح 50 متر من المسافة. الفريق يتكون من 5 أفراد والمدرب لديه بيانات عن أفضل أداء لكل منهم (بالثانية) في مسابقات سابقة كالتالي:

	Backstroke	Breaststroke	Butterfly	Freestyle
A	37.7	43.4	33.3	29.2
B	32.9	33.1	28.5	26.4
C	33.8	42.2	38.9	29.6
D	37	34.7	30.4	28.5
E	35.4	41.8	33.6	31.1

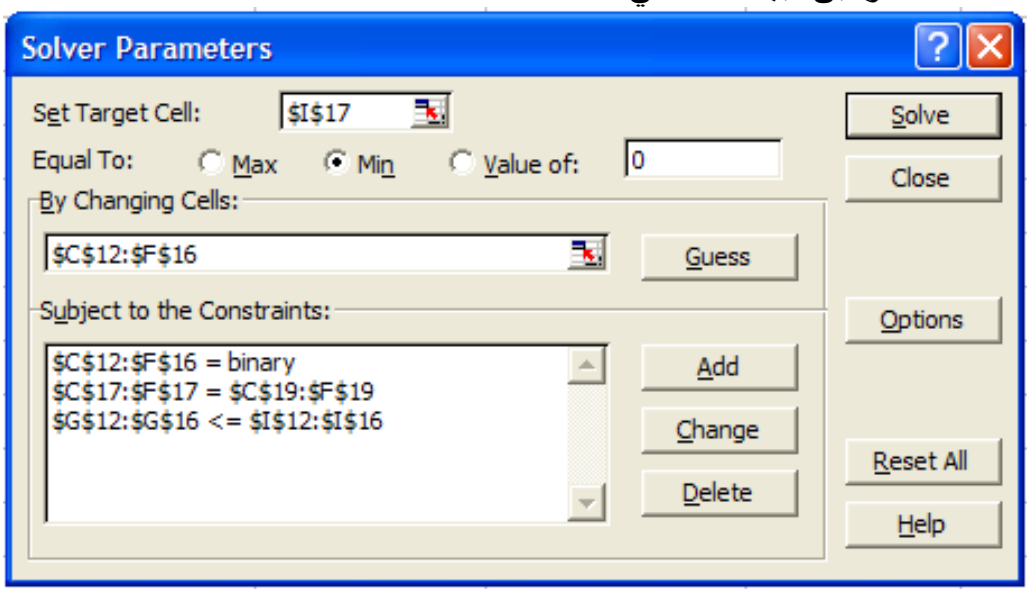
يريد المدرب تحديد أفضل فريق مكون من 4 سباحين لقطع مسافة السباق في أقل وقت ممكن.
الحل
جهز صفحة نشر كالتالي:

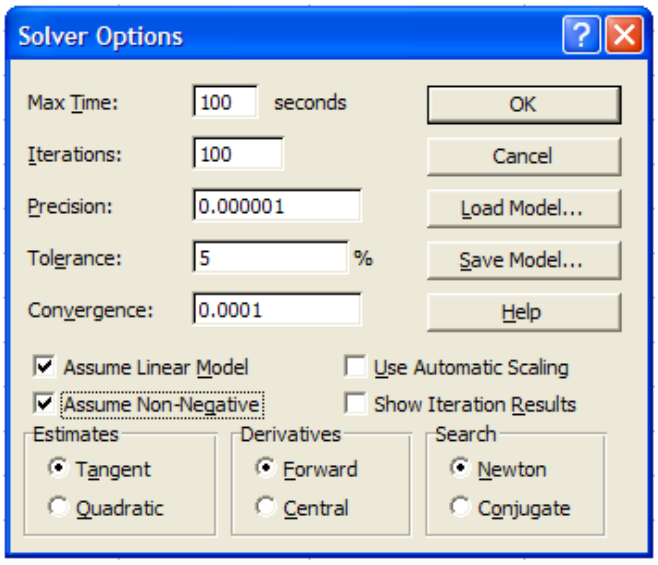
	A	B	C	D	E	F	G	H	I
1	An Assignment Problem								
2									
3		Best Times	Backstroke	Breaststroke	Butterfly	Freestyle			
4		A	37.7	43.4	33.3	29.2			
5		B	32.9	33.1	28.5	26.4			
6		C	33.8	42.2	38.9	29.6			
7		D	37	34.7	30.4	28.5			
8		E	35.4	41.8	33.6	31.1			
9									
10									
11		Assignment	Backstroke	Breaststroke	Butterfly	Freestyle			
12		A	0	0	0	0	0	<=	1
13		B	0	0	0	0	0	<=	1
14		C	0	0	0	0	0	<=	1
15		D	0	0	0	0	0	<=	1
16		E	0	0	0	0	0	<=	1
17			0	0	0	0	Time	=	0
18			=	=	=	=			
19			1	1	1	1			

شكل الصيغ

3	Backstroke	Breaststroke	Butterfly	Freestyle			
4	37.7	43.4	33.3	29.2			
5	32.9	33.1	28.5	26.4			
6	33.8	42.2	38.9	29.6			
7	37	34.7	30.4	28.5			
8	35.4	41.8	33.6	31.1			
9							
10							
11	Backstroke	Breaststroke	Butterfly	Freestyle			
12	0	0	0	0	=SUM(C12:F12)	<=	1
13	0	0	0	0	=SUM(C13:F13)	<=	1
14	0	0	0	0	=SUM(C14:F14)	<=	1
15	0	0	0	0	=SUM(C15:F15)	<=	1
16	0	0	0	0	=SUM(C16:F16)	<=	1
17	=SUM(C12:C16)	=SUM(D12:D16)	=SUM(E12:E16)	=SUM(F12:F16)	Time	=	=SUMPRODUCT(C4:F8,C12:F16)
18	=	=	=	=			
19	1	1	1	1			

أفتح SOLVER وعبئ البيانات كالتالي:





وبالضغط على Solve ينتج الحل

	A	B	C	D	E	F	G	H	I
1	**An Assignment Problem**								
2									
3		**Best Times**	Backstroke	Breaststroke	Butterfly	Freestyle			
4		A	37.7	43.4	33.3	29.2			
5		B	32.9	33.1	28.5	26.4			
6		C	33.8	42.2	38.9	29.6			
7		D	37	34.7	30.4	28.5			
8		E	35.4	41.8	33.6	31.1			
9									
10									
11		**Assignment**	Backstroke	Breaststroke	Butterfly	Freestyle			
12		A	0	0	0	1	1	<=	1
13		B	0	0	1	0	1	<=	1
14		C	1	0	0	0	1	<=	1
15		D	0	1	0	0	1	<=	1
16		E	0	0	0	0	0	<=	1
17			1	1	1	1	Time	=	126.2
18			=	=	=	=			
19			1	1	1	1			

تقرير الحل

Microsoft Excel 10.0 Answer Report
Worksheet: [Book2]Sheet1
Report Created: 7/6/2003 8:13:19 PM

Target Cell (Min)

Cell	Name	Original Value	Final Value
$I$17	=	0	126.2

Adjustable Cells

Cell	Name	Original Value	Final Value
$C$12	A Backstroke	0	0
$D$12	A Breaststroke	0	0
$E$12	A Butterfly	0	0
$F$12	A Freestyle	0	1
$C$13	B Backstroke	0	0
$D$13	B Breaststroke	0	0
$E$13	B Butterfly	0	1
$F$13	B Freestyle	0	0
$C$14	C Backstroke	0	1
$D$14	C Breaststroke	0	0
$E$14	C Butterfly	0	0
$F$14	C Freestyle	0	0
$C$15	D Backstroke	0	0
$D$15	D Breaststroke	0	1
$E$15	D Butterfly	0	0
$F$15	D Freestyle	0	0
$C$16	E Backstroke	0	0
$D$16	E Breaststroke	0	0
$E$16	E Butterfly	0	0
$F$16	E Freestyle	0	0

تمرين: يترك للطالب التحقق من النتائج وتفسير المخرجات.

مثال 7:

مثال تخصيص Assignment Problem:

الجدول التالي يمثل زمن النقل من خطوط التجميع إلى مناطق الفحص لقطع إلكترونية :

Assembly Line	Inspection Area				
	A	B	C	D	E
1	10	4	6	10	12
2	11	7	7	9	14
3	13	8	12	14	15
4	14	16	13	17	17
5	19	11	17	20	19

المطلوب هو تحديد الكيفية التي تنقل بها القطع من خطوط الإنتاج إلى مناطق الفحص في أقل زمن ممكن مع ملاحظة أن يخصص خط تجميع واحد لمنطقة فحص واحدة.

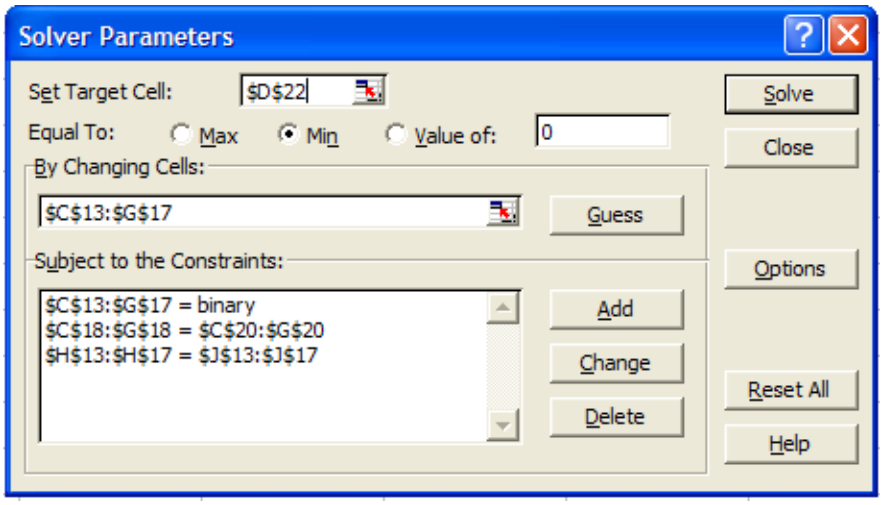
جهز صفحة نشر كالتالي:

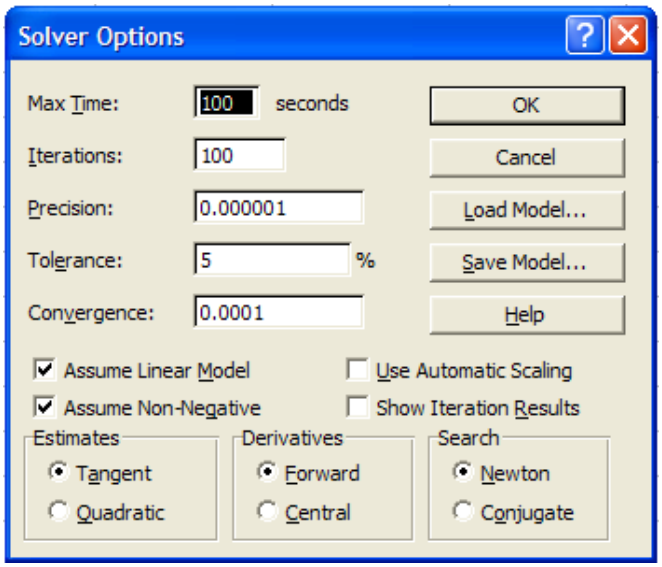
	A	B	C	D	E	F	G	H	I	J
1	**Assembly Line Assignment**									
2										
3		Assembly		Inspection	Area					
4		Line	A	B	C	D	E			
5		1	10	4	6	10	12			
6		2	11	7	7	9	14			
7		3	13	8	12	14	15			
8		4	14	16	13	17	17			
9		5	19	11	17	20	19			
10										
11		Assembly		Inspection	Area					
12		Line	A	B	C	D	E			
13		1	0	0	0	0	0	0	=	1
14		2	0	0	0	0	0	0	=	1
15		3	0	0	0	0	0	0	=	1
16		4	0	0	0	0	0	0	=	1
17		5	0	0	0	0	0	0	=	1
18			0	0	0	0	0			
19			=	=	=	=	=			
20			1	1	1	1	1			
21										
22			Min=	0						

شكل الصيغ

	C	D	E	F	G	H	I	J
1								
2								
3		Inspection	Area					
4	A	B	C	D	E			
5	10	4	6	10	12			
6	11	7	7	9	14			
7	13	8	12	14	15			
8	14	16	13	17	17			
9	19	11	17	20	19			
10								
11		Inspection	Area					
12	A	B	C	D	E			
13	0	0	0	0	0	=SUM(C13:G13)	=	1
14	0	0	0	0	0	=SUM(C14:G14)	=	1
15	0	0	0	0	0	=SUM(C15:G15)	=	1
16	0	0	0	0	0	=SUM(C16:G16)	=	1
17	0	0	0	0	0	=SUM(C17:G17)	=	1
18	=SUM(C13:C17)	=SUM(D13:D17)	=SUM(E13:E17)	=SUM(F13:F17)	=SUM(G13:G17)			
19	=	=	=	=	=			
20	1	1	1	1	1			
21								
22	Min=	=SUMPRODUCT(C5:G9,C13:G17)						
23								

أفتح SOLVER وعبئ البيانات كالتالي:





وبالضغط على Solve ينتج الحل

	A	B	C	D	E	F	G	H	I	J
1	**Assembly Line Assignment**									
2										
3		Assembly		Inspection	Area					
4		Line	A	B	C	D	E			
5		1	10	4	6	10	12			
6		2	11	7	7	9	14			
7		3	13	8	12	14	15			
8		4	14	16	13	17	17			
9		5	19	11	17	20	19			
10										
11		Assembly		Inspection	Area					
12		Line	A	B	C	D	E			
13		1	0	0	1	0	0	1	=	1
14		2	0	0	0	1	0	1	=	1
15		3	0	0	0	0	1	1	=	1
16		4	1	0	0	0	0	1	=	1
17		5	0	1	0	0	0	1	=	1
18			1	1	1	1	1			
19			=	=	=	=	=			
20			1	1	1	1	1			
21										
22			Min=	55						

وتقرير الحل

	Cell	Name	Original Value	Final Value
	$D$22	Min= =	0	55

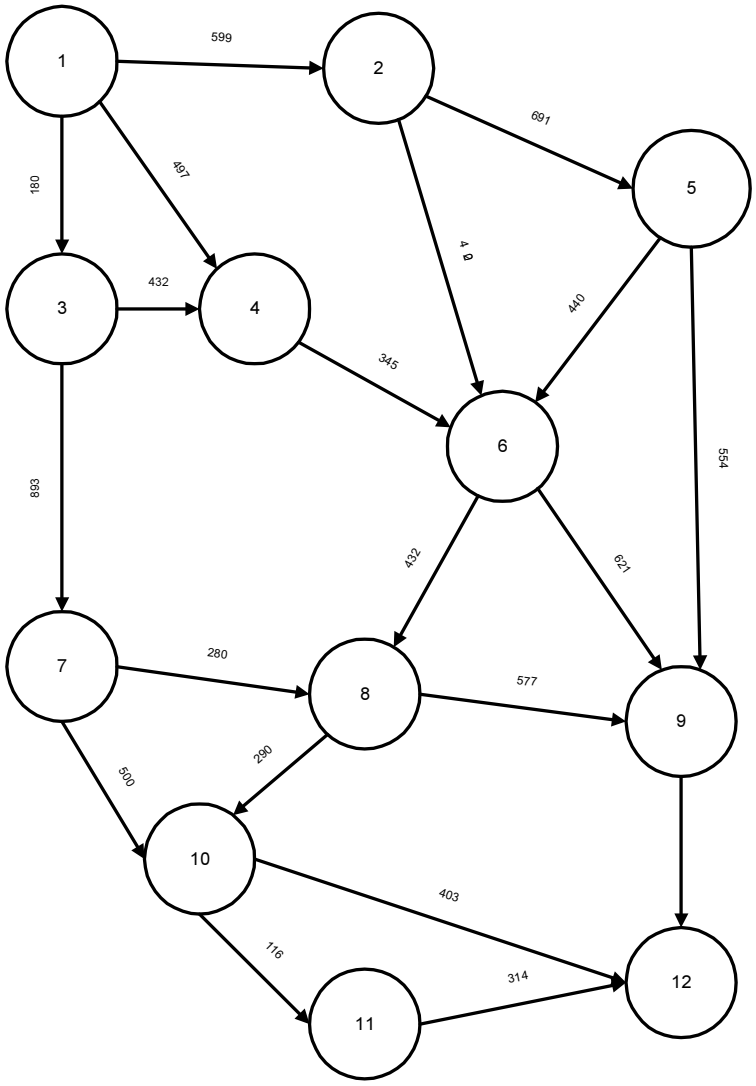
Adjustable Cells

Cell	Name	Original Value	Final Value
$C$13	A	0	0
$D$13	B	0	0
$E$13	C	0	1
$F$13	D	0	0
$G$13	E	0	0
$C$14	A	0	0
$D$14	B	0	0
$E$14	C	0	0
$F$14	D	0	1
$G$14	E	0	0
$C$15	A	0	0
$D$15	B	0	0
$E$15	C	0	0
$F$15	D	0	0
$G$15	E	0	1
$C$16	A	0	1
$D$16	B	0	0
$E$16	C	0	0
$F$16	D	0	0
$G$16	E	0	0
$C$17	A	0	0
$D$17	B	0	1
$E$17	C	0	0
$F$17	D	0	0
$G$17	E	0	0

مثال 8:

مشكلة أقصر طريق Shortest Path Problem:

الشبكة التالية تمثل خطوط سير ممكنة للسفر من العقدة 1 (عقدة البداية) إلى العقدة 12 (عقدة النهاية) والمراد تعيين أقصر مسافة لخط السير.



نضع الشبكة على شكل برمجة خطية كالتالي:

$$
\begin{aligned}
Min \quad & 599x_{1,2} + 180x_{1,3} + 497x_{1,4} + 691x_{2,5} + 420x_{2,6} + \\
& 432x_{3,4} + 893x_{3,7} + 345x_{4,6} + 440x_{5,6} + 554x_{5,9} + \\
& 432x_{6,8} + 621x_{6,9} + 280x_{7,8} + 500x_{7,10} + 577x_{8,9} + \\
& 290x_{8,10} + 268x_{9,12} + 116x_{10,11} + 403x_{10,12} + 314x_{11,12}
\end{aligned}
$$

$$
\begin{aligned}
ST \quad & x_{1,2} + x_{1,3} + x_{1,4} = 1 \\
& x_{2,5} + x_{2,6} - x_{1,2} = 0 \\
& x_{3,4} + x_{3,7} - x_{1,3} = 0 \\
& x_{4,6} - x_{1,4} - x_{3,4} = 0 \\
& x_{5,6} + x_{5,9} - x_{2,5} = 0 \\
& x_{6,8} + x_{6,9} - x_{2,6} - x_{4,6} - x_{5,6} = 0 \\
& x_{7,8} + x_{7,10} - x_{3,7} = 0 \\
& x_{8,9} + x_{8,10} - x_{6,8} - x_{7,8} = 0 \\
& x_{9,12} - x_{9,5} - x_{6,9} - x_{8,9} = 0 \\
& x_{10,11} + x_{10,12} - x_{7,10} - x_{8,10} = 0 \\
& x_{11,12} - x_{10,11} = 0 \\
& -x_{9,12} - x_{10,12} - x_{11,12} = 0 \\
All \quad & x_{i,j}\text{'s} = 0 \text{ or } 1
\end{aligned}
$$

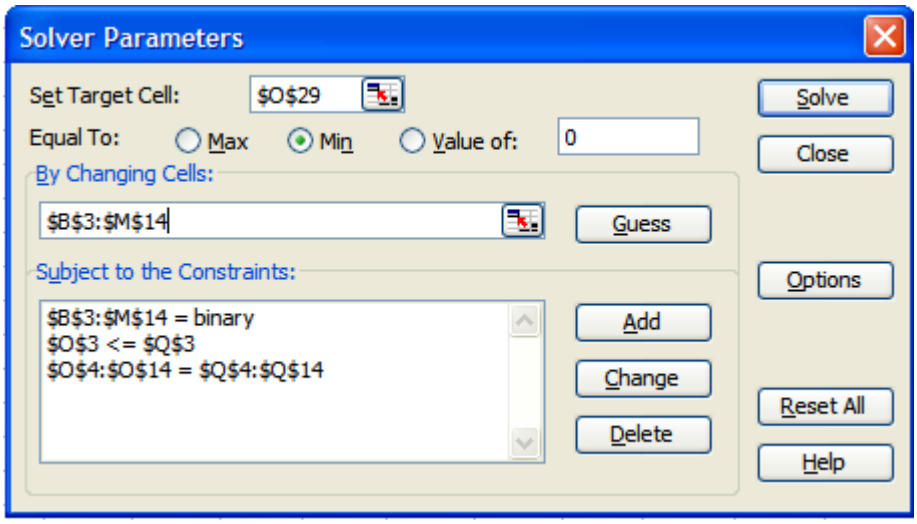
أدخل التالي في صفحة من إكسل

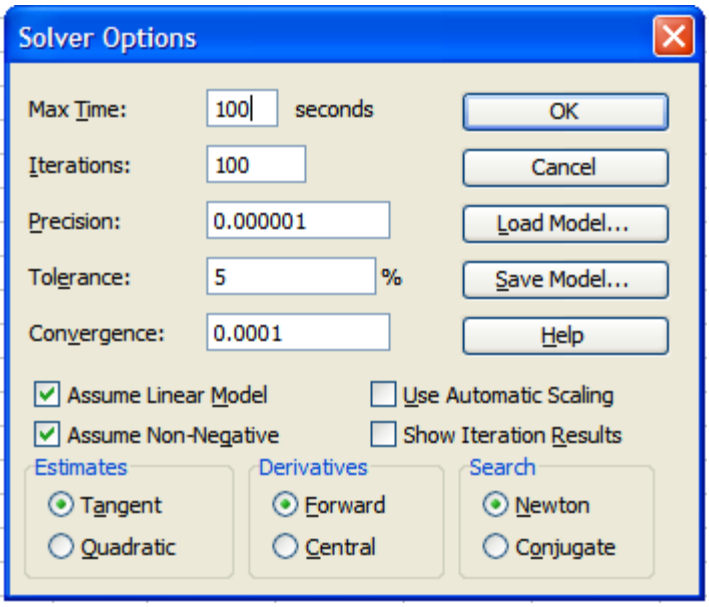
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Shortest Path Problem																
2	x	1	2	3	4	5	6	7	8	9	10	11	12				
3	1	0	0	0	0	0	0	0	0	0	0	0	0		0	<=	1
4	2	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
5	3	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
6	4	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
7	5	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
8	6	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
9	7	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
10	8	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
11	9	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
12	10	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
13	11	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
14	12	0	0	0	0	0	0	0	0	0	0	0	0		0	=	-1
15																	
16	x	1	2	3	4	5	6	7	8	9	10	11	12				
17	1	0	599	180	497	0	0	0	0	0	0	0	0				
18	2	0	0	0	0	691	420	0	0	0	0	0	0				
19	3	0	0	0	432	0	0	893	0	0	0	0	0				
20	4	0	0	0	0	0	345	0	0	0	0	0	0				
21	5	0	0	0	0	0	440	0	0	554	0	0	0				
22	6	0	0	0	0	0	0	0	432	621	0	0	0				
23	7	0	0	0	0	0	0	0	280	0	500	0	0				
24	8	0	0	0	0	0	0	0	0	577	290	0	0				
25	9	0	0	0	0	0	0	0	0	0	0	0	268				
26	10	0	0	0	0	0	0	0	0	0	0	116	403				
27	11	0	0	0	0	0	0	0	0	0	0	0	314				
28	12	0	0	0	0	0	0	0	0	0	0	0	0				
29															0		

الصيغ التالية تعرف القيود

O	P	Q
=C3+D3+E3	<=	1
=F4+G4-C3	=	0
=E5+H5-D3	=	0
=G6-E3-E5	=	0
=G7+J7-F4	=	0
=I8+J8+-G4-G6-G7	=	0
=I9+K9-H5	=	0
=J10+K10-I8-I9	=	0
=M11-J7-J8-J10	=	0
=L12+M12-K9-K10	=	0
=M13-L12	=	0
=-M11-M12-M13	=	-1
=SUMPRODUCT(B3:M14,B17:M28)		

أدخل التالي في نافذة حوار Solver





ينتج الحل

O29 =SUMPRODUCT(B3:M14,B17:M28)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Shortest Path Problem																
2	x	1	2	3	4	5	6	7	8	9	10	11	12				
3	1	0	0	2E-16	1	0	0	0	0	0	0	0	0		1	<=	1
4	2	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
5	3	0	0	0	0	0	0	2E-16	0	0	0	0	0		0	=	0
6	4	0	0	0	0	0	1	0	0	0	0	0	0		0	=	0
7	5	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
8	6	0	0	0	0	0	0	0	0	1	0	0	0		0	=	0
9	7	0	0	0	0	0	0	0	0	0	2E-16	0	0		0	=	0
10	8	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
11	9	0	0	0	0	0	0	0	0	0	0	0	1		0	=	0
12	10	0	0	0	0	0	0	0	0	0	0	0	2E-16		0	=	0
13	11	0	0	0	0	0	0	0	0	0	0	0	0		0	=	0
14	12	0	0	0	0	0	0	0	0	0	0	0	0		-1	=	-1
15																	
16	x	1	2	3	4	5	6	7	8	9	10	11	12				
17	1	0	599	180	497	0	0	0	0	0	0	0	0				
18	2	0	0	0	0	691	420	0	0	0	0	0	0				
19	3	0	0	0	432	0	0	893	0	0	0	0	0				
20	4	0	0	0	0	0	345	0	0	0	0	0	0				
21	5	0	0	0	0	0	440	0	0	554	0	0	0				
22	6	0	0	0	0	0	0	0	432	621	0	0	0				
23	7	0	0	0	0	0	0	0	280	0	500	0	0				
24	8	0	0	0	0	0	0	0	0	577	290	0	0				
25	9	0	0	0	0	0	0	0	0	0	0	0	268				
26	10	0	0	0	0	0	0	0	0	0	0	116	403				
27	11	0	0	0	0	0	0	0	0	0	0	0	314				
28	12	0	0	0	0	0	0	0	0	0	0	0	0				
29															1731		

Microsoft Excel 11.0 Answer Report
Worksheet: [Shortest Path.xls]Sheet1
Report Created: 11/24/2006 10:19:42 AM

Target Cell (Min)

Cell	Name	Original Value	Final Value
$O$29		0	1731

Adjustable Cells

Cell	Name	Original Value	Final Value
$B$3		0	0
$C$3		0	0
$D$3		0	2.22045E-16
$E$3		0	1
$F$3		0	0
$G$3		0	0
$H$3		0	0
$I$3		0	0
$J$3		0	0
$K$3		0	0
$L$3		0	0
$M$3		0	0
$B$4		0	0
$C$4		0	0
$D$4		0	0
$E$4		0	0
$F$4		0	0
$G$4		0	0
$H$4		0	0
$I$4		0	0
$J$4		0	0
$K$4		0	0
$L$4		0	0
$M$4		0	0
$B$5		0	0
$C$5		0	0
$D$5		0	0
$E$5		0	0
$F$5		0	0
$G$5		0	0
$H$5		0	2.22045E-16
$I$5		0	0
$J$5		0	0
$K$5		0	0
$L$5		0	0
$M$5		0	0
$B$6		0	0
$C$6		0	0
$D$6		0	0
$E$6		0	0
$F$6		0	0
$G$6		0	1
$H$6		0	0
$I$6		0	0

$J$6	0	0
$K$6	0	0
$L$6	0	0
$M$6	0	0
$B$7	0	0
$C$7	0	0
$D$7	0	0
$E$7	0	0
$F$7	0	0
$G$7	0	0
$H$7	0	0
$I$7	0	0
$J$7	0	0
$K$7	0	0
$L$7	0	0
$M$7	0	0
$B$8	0	0
$C$8	0	0
$D$8	0	0
$E$8	0	0
$F$8	0	0
$G$8	0	0
$H$8	0	0
$I$8	0	0
$J$8	0	1
$K$8	0	0
$L$8	0	0
$M$8	0	0
$B$9	0	0
$C$9	0	0
$D$9	0	0
$E$9	0	0
$F$9	0	0
$G$9	0	0
$H$9	0	0
$I$9	0	0
$J$9	0	0
$K$9	0	2.22045E-16
$L$9	0	0
$M$9	0	0
$B$10	0	0
$C$10	0	0
$D$10	0	0
$E$10	0	0
$F$10	0	0
$G$10	0	0
$H$10	0	0
$I$10	0	0

$J$10	0	0
$K$10	0	0
$L$10	0	0
$M$10	0	0
$B$11	0	0
$C$11	0	0
$D$11	0	0
$E$11	0	0
$F$11	0	0
$G$11	0	0
$H$11	0	0
$I$11	0	0
$J$11	0	0
$K$11	0	0
$L$11	0	0
$M$11	0	1
$B$12	0	0
$C$12	0	0
$D$12	0	0
$E$12	0	0
$F$12	0	0
$G$12	0	0
$H$12	0	0
$I$12	0	0
$J$12	0	0
$K$12	0	0
$L$12	0	0
$M$12	0	2.22045E-16
$B$13	0	0
$C$13	0	0
$D$13	0	0
$E$13	0	0
$F$13	0	0
$G$13	0	0
$H$13	0	0
$I$13	0	0
$J$13	0	0
$K$13	0	0
$L$13	0	0
$M$13	0	0
$B$14	0	0
$C$14	0	0
$D$14	0	0
$E$14	0	0
$F$14	0	0
$G$14	0	0
$H$14	0	0
$I$14	0	0

$J$14	0	0
$K$14	0	0
$L$14	0	0
$M$14	0	0

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$O$3		1	$O$3<=$Q$3	Binding	0
$O$4		0	$O$4=$Q$4	Not Binding	0
$O$5		0	$O$5=$Q$5	Not Binding	0
$O$6		0	$O$6=$Q$6	Not Binding	0
$O$7		0	$O$7=$Q$7	Not Binding	0
$O$8		0	$O$8=$Q$8	Not Binding	0
$O$9		0	$O$9=$Q$9	Not Binding	0
$O$10		0	$O$10=$Q$10	Not Binding	0
$O$11		0	$O$11=$Q$11	Not Binding	0
$O$12		0	$O$12=$Q$12	Not Binding	0
$O$13		0	$O$13=$Q$13	Not Binding	0
$O$14		-1	$O$14=$Q$14	Not Binding	0
$B$3		0	$B$3=binary	Binding	0
$C$3		0	$C$3=binary	Binding	0
$D$3		2.22045E-16	$D$3=binary	Binding	0
$E$3		1	$E$3=binary	Binding	0
$F$3		0	$F$3=binary	Binding	0
$G$3		0	$G$3=binary	Binding	0
$H$3		0	$H$3=binary	Binding	0
$I$3		0	$I$3=binary	Binding	0
$J$3		0	$J$3=binary	Binding	0
$K$3		0	$K$3=binary	Binding	0
$L$3		0	$L$3=binary	Binding	0
$M$3		0	$M$3=binary	Binding	0
$B$4		0	$B$4=binary	Binding	0
$C$4		0	$C$4=binary	Binding	0
$D$4		0	$D$4=binary	Binding	0
$E$4		0	$E$4=binary	Binding	0
$F$4		0	$F$4=binary	Binding	0
$G$4		0	$G$4=binary	Binding	0
$H$4		0	$H$4=binary	Binding	0
$I$4		0	$I$4=binary	Binding	0

$J$4	0	$J$4=binary	Binding	0
$K$4	0	$K$4=binary	Binding	0
$L$4	0	$L$4=binary	Binding	0
$M$4	0	$M$4=binary	Binding	0
$B$5	0	$B$5=binary	Binding	0
$C$5	0	$C$5=binary	Binding	0
$D$5	0	$D$5=binary	Binding	0
$E$5	0	$E$5=binary	Binding	0
$F$5	0	$F$5=binary	Binding	0
$G$5	0	$G$5=binary	Binding	0
$H$5	2.22045E-16	$H$5=binary	Binding	0
$I$5	0	$I$5=binary	Binding	0
$J$5	0	$J$5=binary	Binding	0
$K$5	0	$K$5=binary	Binding	0
$L$5	0	$L$5=binary	Binding	0
$M$5	0	$M$5=binary	Binding	0
$B$6	0	$B$6=binary	Binding	0
$C$6	0	$C$6=binary	Binding	0
$D$6	0	$D$6=binary	Binding	0
$E$6	0	$E$6=binary	Binding	0
$F$6	0	$F$6=binary	Binding	0
$G$6	1	$G$6=binary	Binding	0
$H$6	0	$H$6=binary	Binding	0
$I$6	0	$I$6=binary	Binding	0
$J$6	0	$J$6=binary	Binding	0
$K$6	0	$K$6=binary	Binding	0
$L$6	0	$L$6=binary	Binding	0
$M$6	0	$M$6=binary	Binding	0
$B$7	0	$B$7=binary	Binding	0
$C$7	0	$C$7=binary	Binding	0
$D$7	0	$D$7=binary	Binding	0
$E$7	0	$E$7=binary	Binding	0
$F$7	0	$F$7=binary	Binding	0
$G$7	0	$G$7=binary	Binding	0
$H$7	0	$H$7=binary	Binding	0
$I$7	0	$I$7=binary	Binding	0
$J$7	0	$J$7=binary	Binding	0
$K$7	0	$K$7=binary	Binding	0
$L$7	0	$L$7=binary	Binding	0
$M$7	0	$M$7=binary	Binding	0
$B$8	0	$B$8=binary	Binding	0
$C$8	0	$C$8=binary	Binding	0
$D$8	0	$D$8=binary	Binding	0
$E$8	0	$E$8=binary	Binding	0
$F$8	0	$F$8=binary	Binding	0
$G$8	0	$G$8=binary	Binding	0
$H$8	0	$H$8=binary	Binding	0
$I$8	0	$I$8=binary	Binding	0

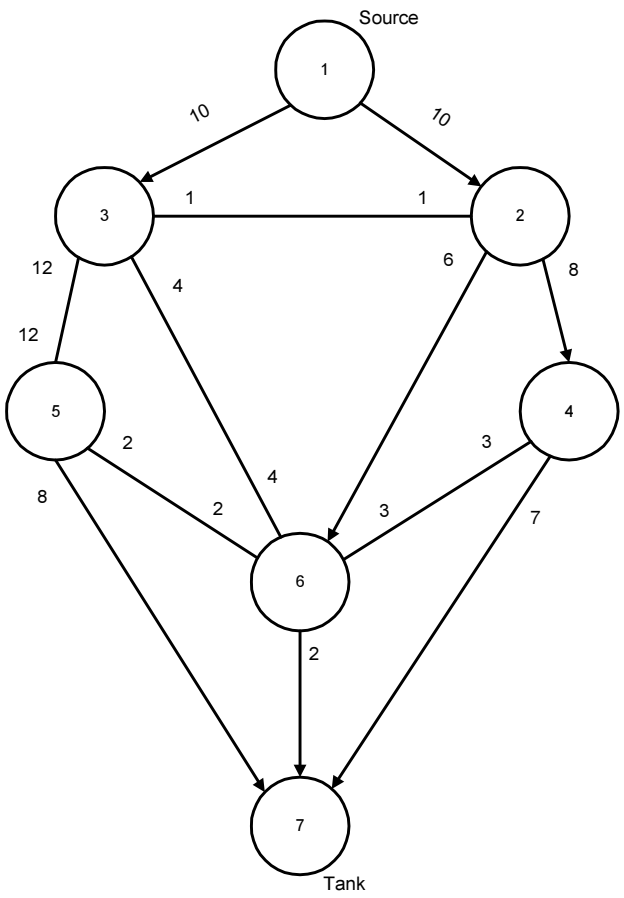
$J$8	1	$J$8=binary	Binding	0
$K$8	0	$K$8=binary	Binding	0
$L$8	0	$L$8=binary	Binding	0
$M$8	0	$M$8=binary	Binding	0
$B$9	0	$B$9=binary	Binding	0
$C$9	0	$C$9=binary	Binding	0
$D$9	0	$D$9=binary	Binding	0
$E$9	0	$E$9=binary	Binding	0
$F$9	0	$F$9=binary	Binding	0
$G$9	0	$G$9=binary	Binding	0
$H$9	0	$H$9=binary	Binding	0
$I$9	0	$I$9=binary	Binding	0
$J$9	0	$J$9=binary	Binding	0
$K$9	2.22045E-16	$K$9=binary	Binding	0
$L$9	0	$L$9=binary	Binding	0
$M$9	0	$M$9=binary	Binding	0
$B$10	0	$B$10=binary	Binding	0
$C$10	0	$C$10=binary	Binding	0
$D$10	0	$D$10=binary	Binding	0
$E$10	0	$E$10=binary	Binding	0
$F$10	0	$F$10=binary	Binding	0
$G$10	0	$G$10=binary	Binding	0
$H$10	0	$H$10=binary	Binding	0
$I$10	0	$I$10=binary	Binding	0
$J$10	0	$J$10=binary	Binding	0
$K$10	0	$K$10=binary	Binding	0
$L$10	0	$L$10=binary	Binding	0
$M$10	0	$M$10=binary	Binding	0
$B$11	0	$B$11=binary	Binding	0
$C$11	0	$C$11=binary	Binding	0
$D$11	0	$D$11=binary	Binding	0
$E$11	0	$E$11=binary	Binding	0
$F$11	0	$F$11=binary	Binding	0
$G$11	0	$G$11=binary	Binding	0
$H$11	0	$H$11=binary	Binding	0
$I$11	0	$I$11=binary	Binding	0
$J$11	0	$J$11=binary	Binding	0
$K$11	0	$K$11=binary	Binding	0
$L$11	0	$L$11=binary	Binding	0
$M$11	1	$M$11=binary	Binding	0
$B$12	0	$B$12=binary	Binding	0
$C$12	0	$C$12=binary	Binding	0
$D$12	0	$D$12=binary	Binding	0
$E$12	0	$E$12=binary	Binding	0
$F$12	0	$F$12=binary	Binding	0
$G$12	0	$G$12=binary	Binding	0
$H$12	0	$H$12=binary	Binding	0
$I$12	0	$I$12=binary	Binding	0

$J$12	0	$J$12=binary	Binding	0
$K$12	0	$K$12=binary	Binding	0
$L$12	0	$L$12=binary	Binding	0
$M$12	2.22045E-16	$M$12=binary	Binding	0
$B$13	0	$B$13=binary	Binding	0
$C$13	0	$C$13=binary	Binding	0
$D$13	0	$D$13=binary	Binding	0
$E$13	0	$E$13=binary	Binding	0
$F$13	0	$F$13=binary	Binding	0
$G$13	0	$G$13=binary	Binding	0
$H$13	0	$H$13=binary	Binding	0
$I$13	0	$I$13=binary	Binding	0
$J$13	0	$J$13=binary	Binding	0
$K$13	0	$K$13=binary	Binding	0
$L$13	0	$L$13=binary	Binding	0
$M$13	0	$M$13=binary	Binding	0
$B$14	0	$B$14=binary	Binding	0
$C$14	0	$C$14=binary	Binding	0
$D$14	0	$D$14=binary	Binding	0
$E$14	0	$E$14=binary	Binding	0
$F$14	0	$F$14=binary	Binding	0
$G$14	0	$G$14=binary	Binding	0
$H$14	0	$H$14=binary	Binding	0
$I$14	0	$I$14=binary	Binding	0
$J$14	0	$J$14=binary	Binding	0
$K$14	0	$K$14=binary	Binding	0
$L$14	0	$L$14=binary	Binding	0
$M$14	0	$M$14=binary	Binding	0

مثال 9:

مشكلة التدفق الأعظم Maximal Flow Problem:

الشبكة التالية تمثل تمديدات لأنابيب ماء في شبكة تمد خزان بالمياه



الجدول التالي يعطي سعة الأنابيب (1000 جالون/دقيقة)

		TO						
F		1	2	3	4	5	6	7
R	1	0	10	10	0	0	0	0
O	2	0	0	1	8	0	6	0
M	3	0	1	0	0	12	4	0
	4	0	0	0	0	0	3	7
	5	0	0	0	0	0	2	8
	6	0	0	4	3	2	0	2
	7	0	0	0	0	0	0	0

المطلوب تحديد مسار أقصى تدفق للخزان.

الحل:

بوضع المشكلة على شكل برمجة خطية

$$Max \quad x_{1,2} + x_{1,3}$$

St

$$x_{2,3} + x_{2,4} + x_{2,6} - x_{1,2} - x_{3,2} = 0$$
$$x_{3,2} + x_{3,5} + x_{3,6} - x_{1,3} - x_{2,3} - x_{6,3} = 0$$
$$x_{4,6} + x_{4,7} - x_{2,4} - x_{6,4} = 0$$
$$x_{5,6} + x_{5,7} - x_{3,5} - x_{6,5} = 0$$
$$x_{6,3} + x_{6,4} + x_{6,5} + x_{6,7} - x_{2,6} - x_{3,6} - x_{4,6} - x_{5,6} = 0$$
$$x_{1,2} \le 10$$
$$x_{1,3} \le 10$$
$$x_{2,3} \le 1$$
$$x_{2,4} \le 8$$
$$x_{2,6} \le 6$$
$$x_{3,2} \le 1$$
$$x_{3,5} \le 12$$
$$x_{3,6} \le 4$$
$$x_{4,6} \le 3$$
$$x_{4,7} \le 7$$
$$x_{5,6} \le 2$$
$$x_{5,7} \le 8$$
$$x_{6,3} \le 4$$
$$x_{6,4} \le 3$$
$$x_{6,5} \le 2$$
$$x_{6,7} \le 2$$

$$All \quad x_{i,j}\,'s \ge 0$$

ويدخل في صفحة من إكسل

C21 =SUMPRODUCT(B3:H9,B12:H18)

	A	B	C	D	E	F	G	H	I	J	K
1	Maximal Flow Problem										
2	x	1	2	3	4	5	6	7			
3	1	0	0	0	0	0	0	0			
4	2	0	0	0	0	0	0	0	0	=	0
5	3	0	0	0	0	0	0	0	0	=	0
6	4	0	0	0	0	0	0	0	0	=	0
7	5	0	0	0	0	0	0	0	0	=	0
8	6	0	0	0	0	0	0	0	0	=	0
9	7	0	0	0	0	0	0	0	0	<=	10
10									0	<=	10
11	x	1	2	3	4	5	6	7	0	<=	1
12	1	0	1	1	0	0	0	0	0	<=	8
13	2	0	0	0	0	0	0	0	0	<=	6
14	3	0	0	0	0	0	0	0	0	<=	1
15	4	0	0	0	0	0	0	0	0	<=	12
16	5	0	0	0	0	0	0	0	0	<=	4
17	6	0	0	0	0	0	0	0	0	<=	3
18	7	0	0	0	0	0	0	0	0	<=	7
19									0	<=	2
20									0	<=	8
21			0						0	<=	4
22									0	<=	3
23									0	<=	2
24									0	<=	2

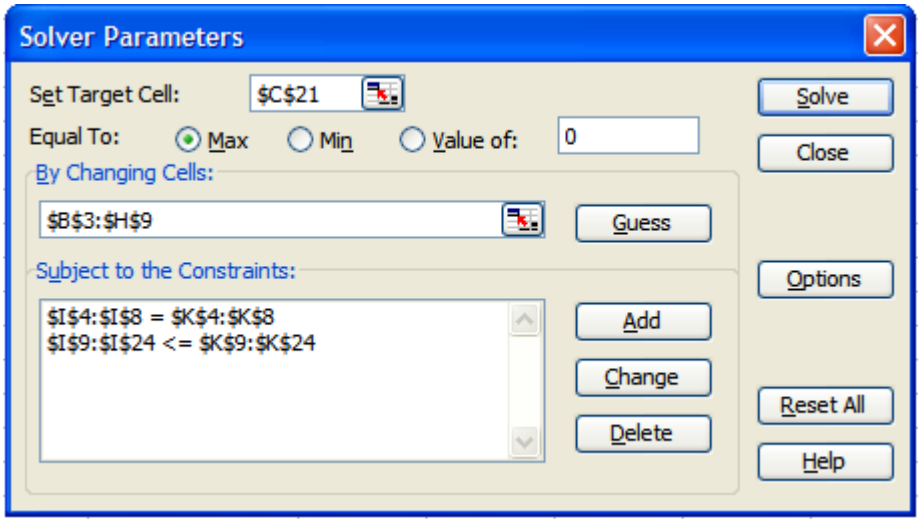
وصيغ القيود

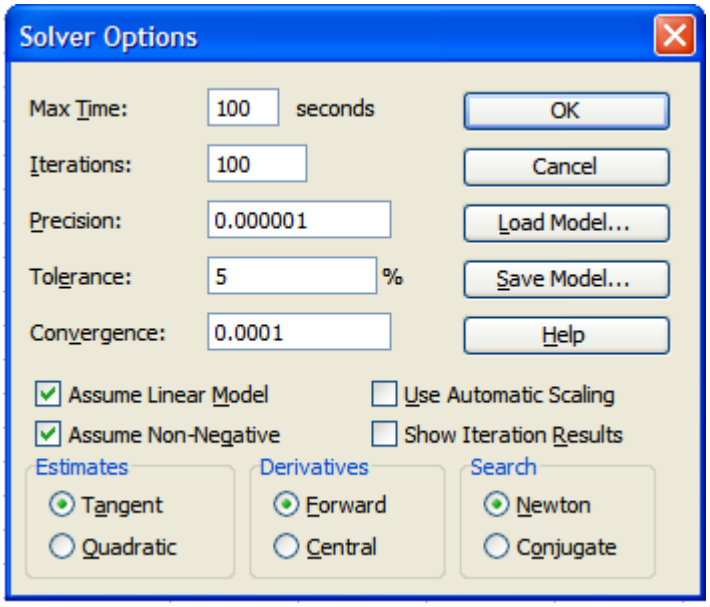
I	J	K
=D4+E4+G4-C3-C5	=	0
=C5+F5+G5-D3-D4-D8	=	0
=G6+H6-E4-E8	=	0
=G7+H7-F5-F8	=	0
=D8+E8+F8+H8-G4-G5-G6-G7	=	0
=C3	<=	10
=D3	<=	10
=D4	<=	1
=E4	<=	8
=G4	<=	6
=C5	<=	1
=F5	<=	12
=G5	<=	4
=G6	<=	3
=H6	<=	7
=G7	<=	2
=H7	<=	8
=D8	<=	4
=E8	<=	3
=F8	<=	2
=H8	<=	2

دالة الهدف هي

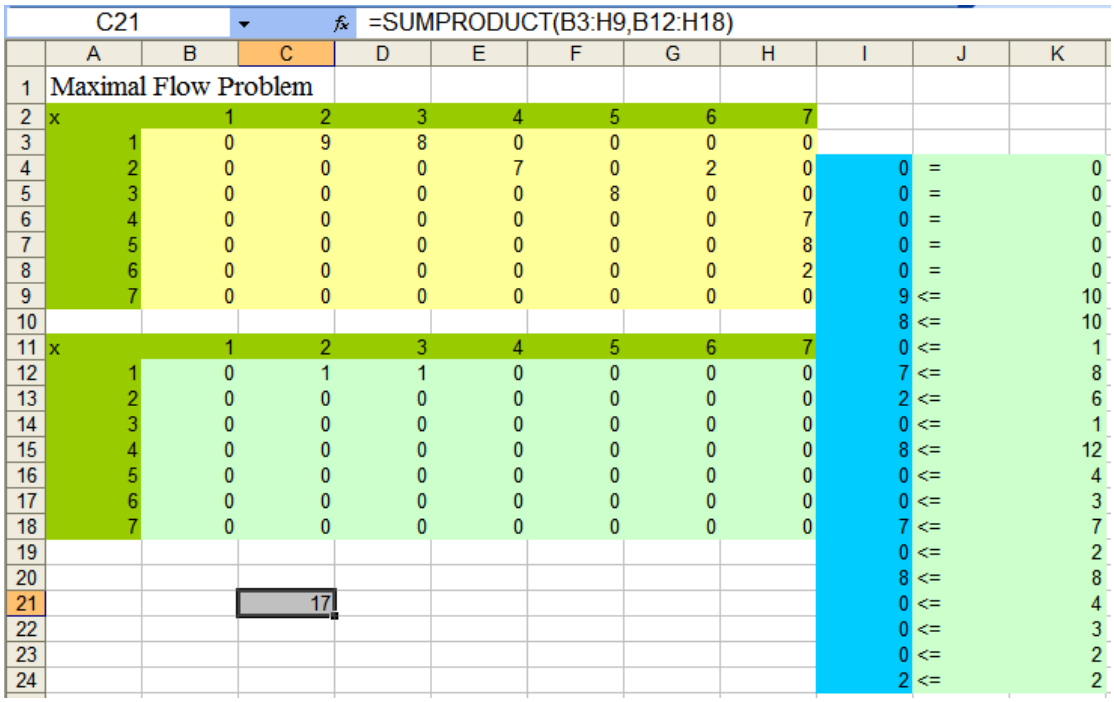
=SUMPRODUCT(B3:H9,B12:H18)

ويدخل التالي في نافذة حوار Solver





وينتج الحل التالي



C21 =SUMPRODUCT(B3:H9,B12:H18)

	A	B	C	D	E	F	G	H	I	J	K
1	Maximal Flow Problem										
2	x	1	2	3	4	5	6	7			
3	1	0	9	8	0	0	0	0			
4	2	0	0	0	7	0	2	0	0	=	0
5	3	0	0	0	0	8	0	0	0	=	0
6	4	0	0	0	0	0	0	7	0	=	0
7	5	0	0	0	0	0	0	8	0	=	0
8	6	0	0	0	0	0	0	2	0	=	0
9	7	0	0	0	0	0	0	0	9	<=	10
10									8	<=	10
11	x	1	2	3	4	5	6	7	0	<=	1
12	1	0	1	1	0	0	0	0	7	<=	8
13	2	0	0	0	0	0	0	0	2	<=	6
14	3	0	0	0	0	0	0	0	0	<=	1
15	4	0	0	0	0	0	0	0	8	<=	12
16	5	0	0	0	0	0	0	0	0	<=	4
17	6	0	0	0	0	0	0	0	0	<=	3
18	7	0	0	0	0	0	0	0	7	<=	7
19									0	<=	2
20									8	<=	8
21			17						0	<=	4
22									0	<=	3
23									0	<=	2
24									2	<=	2

Microsoft Excel 11.0 Answer Report
Worksheet: [Maximal Flow.xls]Sheet1
Report Created: 11/24/2006 10:38:48 AM

Target Cell (Max)

Cell	Name	Original Value	Final Value
$C$21		0	17

Adjustable Cells

Cell	Name	Original Value	Final Value
$B$3		0	0
$C$3		0	9
$D$3		0	8
$E$3		0	0
$F$3		0	0
$G$3		0	0
$H$3		0	0
$B$4		0	0
$C$4		0	0
$D$4		0	0
$E$4		0	7
$F$4		0	0
$G$4		0	2

$H$4	0	0
$B$5	0	0
$C$5	0	0
$D$5	0	0
$E$5	0	0
$F$5	0	8
$G$5	0	0
$H$5	0	0
$B$6	0	0
$C$6	0	0
$D$6	0	0
$E$6	0	0
$F$6	0	0
$G$6	0	0
$H$6	0	7
$B$7	0	0
$C$7	0	0
$D$7	0	0
$E$7	0	0
$F$7	0	0
$G$7	0	0
$H$7	0	8
$B$8	0	0
$C$8	0	0
$D$8	0	0
$E$8	0	0
$F$8	0	0
$G$8	0	0
$H$8	0	2
$B$9	0	0
$C$9	0	0
$D$9	0	0
$E$9	0	0
$F$9	0	0
$G$9	0	0
$H$9	0	0

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$I$4		0	$I$4=$K$4	Not Binding	0
$I$5		0	$I$5=$K$5	Not Binding	0
$I$6		0	$I$6=$K$6	Not Binding	0
$I$7		0	$I$7=$K$7	Not Binding	0
$I$8		0	$I$8=$K$8	Not	0

				Binding	
$I$9		9	$I$9<=$K$9	Not Binding	1
$I$10		8	$I$10<=$K$10	Not Binding	2
$I$11	x	0	$I$11<=$K$11	Not Binding	1
$I$12		7	$I$12<=$K$12	Not Binding	1
$I$13		2	$I$13<=$K$13	Not Binding	4
$I$14		0	$I$14<=$K$14	Not Binding	1
$I$15		8	$I$15<=$K$15	Not Binding	4
$I$16		0	$I$16<=$K$16	Not Binding	4
$I$17		0	$I$17<=$K$17	Not Binding	3
$I$18		7	$I$18<=$K$18	Binding	0
$I$19		0	$I$19<=$K$19	Not Binding	2
$I$20		8	$I$20<=$K$20	Binding	0
$I$21		0	$I$21<=$K$21	Not Binding	4
$I$22		0	$I$22<=$K$22	Not Binding	3
$I$23		0	$I$23<=$K$23	Not Binding	2
$I$24		2	$I$24<=$K$24	Binding	0

البرمجة الخطية و العددية المختلطة بواسطة SageMath

سوف نستعرض جميع الأمثلة السابقة مستخدمين SageMath

مثال 1:

$$
\begin{aligned}
Max \quad & 8x_1 + 5x_2 \\
St \quad & 2x_1 + x_2 \le 1000 \\
& 3x_1 + 4x_2 \le 2400 \\
& x_1 + x_2 \le 700 \\
& x_1 - x_2 \le 350 \\
& x_1, x_2 \ge 0
\end{aligned}
$$

```
lp1 = MixedIntegerLinearProgram( maximization = True)
x = lp1.new_variable(nonnegative = True)
lp1.set_objective( 8*x[1] + 5*x[2])
lp1.add_constraint(2*x[1] + x[2]  <= 1000)
lp1.add_constraint( 3* x[1] + 4*x[2]   <= 2400)
lp1.add_constraint(x[1] + x[2] <= 700 )
lp1.add_constraint(x[1]-  x[2]     <= 350 )
lp1.show()
lp1.solve()
lp1.get_values(x)
lp1.get_values(x[1])
lp1.get_values(x[2])
lp1.polyhedron()
```

مثال 2 :

$$\begin{aligned} &Minimize && 0.6X_1 + 0.5X_2 \\ &ST && 20X_1 + 50X_2 \geq 100 \\ & && 25X_1 + 25X_2 \geq 100 \\ & && 50X_1 + 10X_2 \geq 100 \\ & && X_1, X_2 \geq 0 \end{aligned}$$

```
lp2 = MixedIntegerLinearProgram( maximization = False)
x = lp1.new_variable(nonnegative = True)
lp2.set_objective( 0.6*x[1] + 0.5*x[2])
lp2.add_constraint(20*x[1] + 50*x[2]  >= 100)
lp2.add_constraint( 25* x[1] + 25*x[2] >= 100)
lp2.add_constraint(50*x[1] + 10*x[2] >= 100 )
lp2.show()
lp2.solve()
lp2.get_values(x)
lp2.polyhedron()
```

مثال 3:

$$\begin{aligned} Max.\quad & 12000X_1 + 20000X_2 \\ St.\quad & 2X_1 + 6X_2 \le 27 \\ & X_2 \ge 2 \\ & 3X_1 + X_2 \le 19 \\ & X_1, X_2 \ge 0 \text{ and integers} \end{aligned}$$

```
lp3 = MixedIntegerLinearProgram( maximization = True)
x = lp3.new_variable(nonnegative = True, integer = True)
lp3.set_objective( 12000*x[1] +20000*x[2])
lp3.add_constraint(2*x[1] + 6*x[2]  <= 27)
lp3.add_constraint( x[2] >= 2)
lp3.add_constraint(3*x[1] + x[2] <= 19 )
lp3.show()
lp3.solve()
lp3.get_values(x)
lp3.polyhedron()
```

مثال 4:

$$Min\ 35X_{11}+20X_{12}+40X_{13}+32X_{14}+37X_{21}+40X_{22}+42X_{23}+25X_{24}+40X_{31}+15X_{32}+20X_{33}+28X_{34}$$

$$\begin{aligned}
St\quad & X_{11}+X_{12}+X_{13}+X_{14} \le 1200\\
& X_{21}+X_{22}+X_{23}+X_{24} \le 1000\\
& X_{31}+X_{32}+X_{33}+X_{34} \le 800\\
& X_{11}+X_{21}+X_{31} = 1100\\
& X_{12}+X_{22}+X_{32} = 400\\
& X_{13}+X_{23}+X_{33} = 750\\
& X_{14}+X_{24}+X_{34} = 750\\
& X_{ij} \ge 0, \quad for\ all\ i\ and\ j
\end{aligned}$$

```
lpnet = MixedIntegerLinearProgram( maximization = False)
x = lpnet.new_variable(nonnegative = True)
lpnet.set_objective(35*x[1,1] + 20*x[1,2] + 40*x[1,3] + 32*x[1,4]
+ 37*x[2,1] + 40*x[2,2] + 42*x[2,3] + 25*x[2,4] + 40*x[3,1] +
15*x[3,2] + 20*x[3,3] + 28*x[3,4])
lpnet.add_constraint(x[1,1] + x[1,2] + x[1,3] + x[1,4]  <= 1200)
lpnet.add_constraint(x[2,1] + x[2,2] + x[2,3] + x[2,4]  <= 1000)
lpnet.add_constraint(x[3,1] + x[3,2] + x[3,3] + x[3,4]  <= 800)
lpnet.add_constraint(x[1,1] + x[2,1] + x[3,1]  == 1100)
lpnet.add_constraint(x[1,2] + x[2,2] + x[3,2]  == 400)
lpnet.add_constraint(x[1,3] + x[2,3] + x[3,3]  == 750)
lpnet.add_constraint(x[1,4] + x[2,4] + x[3,4]  == 750)
lpnet.solve()
lpnet.show()
lpnet.get_values(x)
```

```
Minimization:
  35.0 x_0 + 20.0 x_1 + 40.0 x_2 + 32.0 x_3 + 37.0 x_4 + 40.0 x_5 +
42.0 x_6 + 25.0 x_7 + 40.0 x_8 + 15.0 x_9 + 20.0 x_10 + 28.0 x_11
Constraints:
  x_0 + x_1 + x_2 + x_3 <= 1200.0
  x_4 + x_5 + x_6 + x_7 <= 1000.0
  x_8 + x_9 + x_10 + x_11 <= 800.0
  1100.0 <= x_0 + x_4 + x_8 <= 1100.0
  400.0 <= x_1 + x_5 + x_9 <= 400.0
  750.0 <= x_2 + x_6 + x_10 <= 750.0
  750.0 <= x_3 + x_7 + x_11 <= 750.0
Variables:
  x_0 is a continuous variable (min=0.0, max=+oo)
  x_1 is a continuous variable (min=0.0, max=+oo)
  x_2 is a continuous variable (min=0.0, max=+oo)
  x_3 is a continuous variable (min=0.0, max=+oo)
  x_4 is a continuous variable (min=0.0, max=+oo)
  x_5 is a continuous variable (min=0.0, max=+oo)
  x_6 is a continuous variable (min=0.0, max=+oo)
  x_7 is a continuous variable (min=0.0, max=+oo)
  x_8 is a continuous variable (min=0.0, max=+oo)
  x_9 is a continuous variable (min=0.0, max=+oo)
  x_10 is a continuous variable (min=0.0, max=+oo)
  x_11 is a continuous variable (min=0.0, max=+oo)
{(1, 1): 850.0,
 (1, 2): 350.0,
```

```
(1, 3): 0.0,
(1, 4): 0.0,
(2, 1): 250.0,
(2, 2): 0.0,
(2, 3): 0.0,
(2, 4): 750.0,
(3, 1): 0.0,
(3, 2): 50.0,
(3, 3): 750.0,
(3, 4): 0.0}
```

مثال 5:
مشكلة نقل

```
lpnet = MixedIntegerLinearProgram( maximization = False)
x = lpnet.new_variable(nonnegative = True)
lpnet.set_objective(4*x[1,1] + 6*x[1,2] + 6*x[2,1] +
5*x[2,2] + 4*x[3,1] + 2*x[3,2])
lpnet.add_constraint(x[1,1] + x[1,2]  >= 40)
lpnet.add_constraint(x[2,1] + x[2,2]  >= 40)
lpnet.add_constraint(x[3,1] + x[3,2]  >= 40)
lpnet.add_constraint(x[1,1] + x[2,1] + x[3,1]  <= 60)
lpnet.add_constraint(x[1,2] + x[2,2] + x[3,2]  <= 60)
lpnet.show()
lpnet.solve()
lpnet.get_values(x)
```

```
Minimization:
  4.0 x_0 + 6.0 x_1 + 6.0 x_2 + 5.0 x_3 + 4.0 x_4 + 2.0 x_5
Constraints:
  - x_0 - x_1 <= -40.0
  - x_2 - x_3 <= -40.0
  - x_4 - x_5 <= -40.0
  x_0 + x_2 + x_4 <= 60.0
  x_1 + x_3 + x_5 <= 60.0
Variables:
  x_0 is a continuous variable (min=0.0, max=+oo)
  x_1 is a continuous variable (min=0.0, max=+oo)
  x_2 is a continuous variable (min=0.0, max=+oo)
  x_3 is a continuous variable (min=0.0, max=+oo)
  x_4 is a continuous variable (min=0.0, max=+oo)
  x_5 is a continuous variable (min=0.0, max=+oo)
{(1, 1): 40.0,
 (1, 2): 0.0,
 (2, 1): 20.0,
 (2, 2): 20.0,
 (3, 1): 0.0,
 (3, 2): 40.0}
```

مثال 6:
مشكلة تخصيص

```
lpnet = MixedIntegerLinearProgram( maximization = False)
x = lpnet.new_variable(nonnegative = True, binary = True)
lpnet.set_objective(37.7*x[1,1] + 43.4*x[1,2] + 33.3*x[1,3] +
29.2*x[1,4] + 32.9*x[2,1] + 33.1*x[2,2] + 28.5*x[2,3] + 26.4*x[2,4] +
33.8*x[3,1] + 42.2*x[3,2] + 38.9*x[3,3] + 29.6*x[3,4] + 37.0*x[4,1] +
34.7*x[4,2] + 30.4*x[4,3] + 28.5*x[4,4] + 35.4*x[5,1] + 41.8*x[5,2] +
33.6*x[5,3] + 31.1*x[5,4])
lpnet.add_constraint(x[1,1] + x[1,2] + x[1,3] + x[1,4]  <= 1)
lpnet.add_constraint(x[2,1] + x[2,2] + x[2,3] + x[2,4]  <= 1)
lpnet.add_constraint(x[3,1] + x[3,2] + x[3,3] + x[3,4]  <= 1)
lpnet.add_constraint(x[4,1] + x[4,2] + x[4,3] + x[4,4]  <= 1)
lpnet.add_constraint(x[1,1] + x[2,1] + x[3,1] + x[4,1]  == 1)
lpnet.add_constraint(x[1,2] + x[2,2] + x[3,2] + x[4,2]  == 1)
lpnet.add_constraint(x[1,3] + x[2,3] + x[3,3] + x[4,3]  == 1)
lpnet.add_constraint(x[1,4] + x[2,4] + x[3,4] + x[4,4]  == 1)
lpnet.solve()

126.2


lpnet.show()

Minimization:
  37.7 x_0 + 43.4 x_1 + 33.3 x_2 + 29.2 x_3 + 32.9 x_4 + 33.1 x_5 +
28.5 x_6 + 26.4 x_7 + 33.8 x_8 + 42.2 x_9 + 38.9 x_10 + 29.6 x_11 +
37.0 x_12 + 34.7 x_13 + 30.4 x_14 + 28.5 x_15 + 35.4 x_16 + 41.8 x_17 +
33.6 x_18 + 31.1 x_19
Constraints:
  x_0 + x_1 + x_2 + x_3 <= 1.0
  x_4 + x_5 + x_6 + x_7 <= 1.0
  x_8 + x_9 + x_10 + x_11 <= 1.0
  x_12 + x_13 + x_14 + x_15 <= 1.0
  1.0 <= x_0 + x_4 + x_8 + x_12 <= 1.0
  1.0 <= x_1 + x_5 + x_9 + x_13 <= 1.0
  1.0 <= x_2 + x_6 + x_10 + x_14 <= 1.0
  1.0 <= x_3 + x_7 + x_11 + x_15 <= 1.0
Variables:
  x_0 is a boolean variable (min=0.0, max=1.0)
  x_1 is a boolean variable (min=0.0, max=1.0)
  x_2 is a boolean variable (min=0.0, max=1.0)
  x_3 is a boolean variable (min=0.0, max=1.0)
  x_4 is a boolean variable (min=0.0, max=1.0)
  x_5 is a boolean variable (min=0.0, max=1.0)
  x_6 is a boolean variable (min=0.0, max=1.0)
  x_7 is a boolean variable (min=0.0, max=1.0)
  x_8 is a boolean variable (min=0.0, max=1.0)
  x_9 is a boolean variable (min=0.0, max=1.0)
  x_10 is a boolean variable (min=0.0, max=1.0)
  x_11 is a boolean variable (min=0.0, max=1.0)
  x_12 is a boolean variable (min=0.0, max=1.0)
  x_13 is a boolean variable (min=0.0, max=1.0)
  x_14 is a boolean variable (min=0.0, max=1.0)
  x_15 is a boolean variable (min=0.0, max=1.0)
  x_16 is a boolean variable (min=0.0, max=1.0)
```

```
  x_17 is a boolean variable (min=0.0, max=1.0)
  x_18 is a boolean variable (min=0.0, max=1.0)
  x_19 is a boolean variable (min=0.0, max=1.0)

lpnet.get_values(x)

{(1, 1): 0.0,
 (1, 2): 0.0,
 (1, 3): 0.0,
 (1, 4): 1.0,
 (2, 1): 0.0,
 (2, 2): 0.0,
 (2, 3): 1.0,
 (2, 4): 0.0,
 (3, 1): 1.0,
 (3, 2): 0.0,
 (3, 3): 0.0,
 (3, 4): 0.0,
 (4, 1): 0.0,
 (4, 2): 1.0,
 (4, 3): 0.0,
 (4, 4): 0.0,
 (5, 1): 0.0,
 (5, 2): 0.0,
 (5, 3): 0.0,
 (5, 4): 0.0}
```

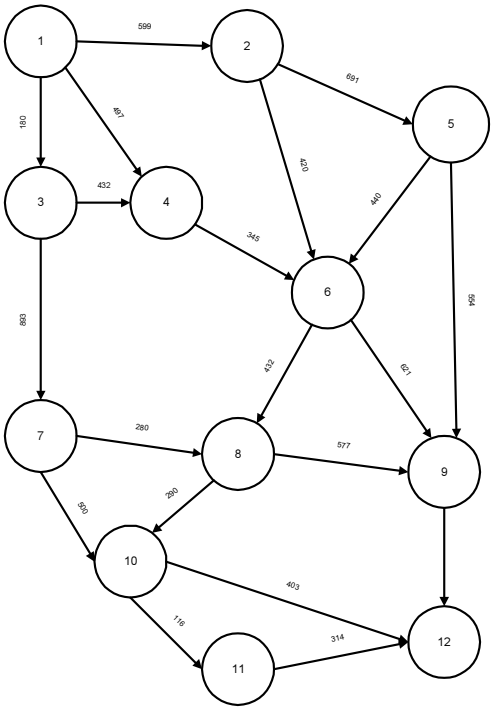
مثال 7:

مثال تخصيص

```
lpnet = MixedIntegerLinearProgram( maximization = False)
x = lpnet.new_variable(nonnegative = True, binary = True)
lpnet.set_objective(10*x[1,1] + 4*x[1,2] + 6*x[1,3] + 10*x[1,4] +
12*x[1,5] +11*x[2,1] + 7*x[2,2] + 7*x[2,3] + 9*x[2,4] + 14*x[2,5] +
13*x[3,1] + 8*x[3,2] + 12*x[3,3] + 14*x[3,4] + 15*x[3,5] + 14*x[4,1] +
16*x[4,2] + 13*x[4,3] + 17*x[4,4] + 17*x[4,5] +19*x[5,1] + 11*x[5,2] +
17*x[5,3] + 20*x[5,4] + 19*x[5,5] )

lpnet = MixedIntegerLinearProgram( maximization = False)
x = lpnet.new_variable(nonnegative = True, binary = True)
lpnet.set_objective(10*x[1,1] + 4*x[1,2] + 6*x[1,3] + 10*x[1,4] +
12*x[1,5] +11*x[2,1] + 7*x[2,2] + 7*x[2,3] + 9*x[2,4] + 14*x[2,5] +
13*x[3,1] + 8*x[3,2] + 12*x[3,3] + 14*x[3,4] + 15*x[3,5] + 14*x[4,1] +
16*x[4,2] + 13*x[4,3] + 17*x[4,4] + 17*x[4,5] +19*x[5,1] + 11*x[5,2] +
17*x[5,3] + 20*x[5,4] + 19*x[5,5] )

lpnet.add_constraint(x[1,1] + x[1,2] + x[1,3] + x[1,4] + x[1,5]  == 1)
lpnet.add_constraint(x[2,1] + x[2,2] + x[2,3] + x[2,4] + x[2,5]  == 1)
lpnet.add_constraint(x[3,1] + x[3,2] + x[3,3] + x[3,4] + x[3,5]  == 1)
lpnet.add_constraint(x[4,1] + x[4,2] + x[4,3] + x[4,4] + x[4,5]  == 1)
lpnet.add_constraint(x[5,1] + x[5,2] + x[5,3] + x[5,4] + x[5,5]  == 1)
lpnet.add_constraint(x[1,1] + x[2,1] + x[3,1] + x[4,1] + x[5,1]  == 1)
lpnet.add_constraint(x[1,2] + x[2,2] + x[3,2] + x[4,2] + x[5,2]  == 1)
lpnet.add_constraint(x[1,3] + x[2,3] + x[3,3] + x[4,3] + x[5,3]  == 1)
lpnet.add_constraint(x[1,4] + x[2,4] + x[3,4] + x[4,4] + x[5,4]  == 1)
lpnet.add_constraint(x[1,5] + x[2,5] + x[3,5] + x[4,5] + x[5,5]  == 1)
lpnet.solve()

55.0


lpnet.show()

Minimization:
  10.0 x_0 + 4.0 x_1 + 6.0 x_2 + 10.0 x_3 + 12.0 x_4 + 11.0 x_5 + 7.0
x_6 + 7.0 x_7 + 9.0 x_8 + 14.0 x_9 + 13.0 x_10 + 8.0 x_11 + 12.0 x_12 +
14.0 x_13 + 15.0 x_14 + 14.0 x_15 + 16.0 x_16 + 13.0 x_17 + 17.0 x_18 +
17.0 x_19 + 19.0 x_20 + 11.0 x_21 + 17.0 x_22 + 20.0 x_23 + 19.0 x_24
Constraints:
  1.0 <= x_0 + x_1 + x_2 + x_3 + x_4 <= 1.0
  1.0 <= x_5 + x_6 + x_7 + x_8 + x_9 <= 1.0
  1.0 <= x_10 + x_11 + x_12 + x_13 + x_14 <= 1.0
  1.0 <= x_15 + x_16 + x_17 + x_18 + x_19 <= 1.0
  1.0 <= x_20 + x_21 + x_22 + x_23 + x_24 <= 1.0
  1.0 <= x_0 + x_5 + x_10 + x_15 + x_20 <= 1.0
  1.0 <= x_1 + x_6 + x_11 + x_16 + x_21 <= 1.0
  1.0 <= x_2 + x_7 + x_12 + x_17 + x_22 <= 1.0
  1.0 <= x_3 + x_8 + x_13 + x_18 + x_23 <= 1.0
  1.0 <= x_4 + x_9 + x_14 + x_19 + x_24 <= 1.0
Variables:
  x_0 is a boolean variable (min=0.0, max=1.0)
  x_1 is a boolean variable (min=0.0, max=1.0)
  x_2 is a boolean variable (min=0.0, max=1.0)
  x_3 is a boolean variable (min=0.0, max=1.0)
```

```
  x_4 is a boolean variable (min=0.0, max=1.0)
  x_5 is a boolean variable (min=0.0, max=1.0)
  x_6 is a boolean variable (min=0.0, max=1.0)
  x_7 is a boolean variable (min=0.0, max=1.0)
  x_8 is a boolean variable (min=0.0, max=1.0)
  x_9 is a boolean variable (min=0.0, max=1.0)
  x_10 is a boolean variable (min=0.0, max=1.0)
  x_11 is a boolean variable (min=0.0, max=1.0)
  x_12 is a boolean variable (min=0.0, max=1.0)
  x_13 is a boolean variable (min=0.0, max=1.0)
  x_14 is a boolean variable (min=0.0, max=1.0)
  x_15 is a boolean variable (min=0.0, max=1.0)
  x_16 is a boolean variable (min=0.0, max=1.0)
  x_17 is a boolean variable (min=0.0, max=1.0)
  x_18 is a boolean variable (min=0.0, max=1.0)
  x_19 is a boolean variable (min=0.0, max=1.0)
  x_20 is a boolean variable (min=0.0, max=1.0)
  x_21 is a boolean variable (min=0.0, max=1.0)
  x_22 is a boolean variable (min=0.0, max=1.0)
  x_23 is a boolean variable (min=0.0, max=1.0)
  x_24 is a boolean variable (min=0.0, max=1.0)

lpnetget_values(x)

{(1, 1): 0.0,
 (1, 2): 0.0,
 (1, 3): 1.0,
 (1, 4): 0.0,
 (1, 5): 0.0,
 (2, 1): 0.0,
 (2, 2): 0.0,
 (2, 3): 0.0,
 (2, 4): 1.0,
 (2, 5): 0.0,
 (3, 1): 0.0,
 (3, 2): 0.0,
 (3, 3): 0.0,
 (3, 4): 0.0,
 (3, 5): 1.0,
 (4, 1): 1.0,
 (4, 2): 0.0,
 (4, 3): 0.0,
 (4, 4): 0.0,
 (4, 5): 0.0,
 (5, 1): 0.0,
 (5, 2): 1.0,
 (5, 3): 0.0,
 (5, 4): 0.0,
 (5, 5): 0.0}
```

مثال 8:
مشكلة أقصر طريق



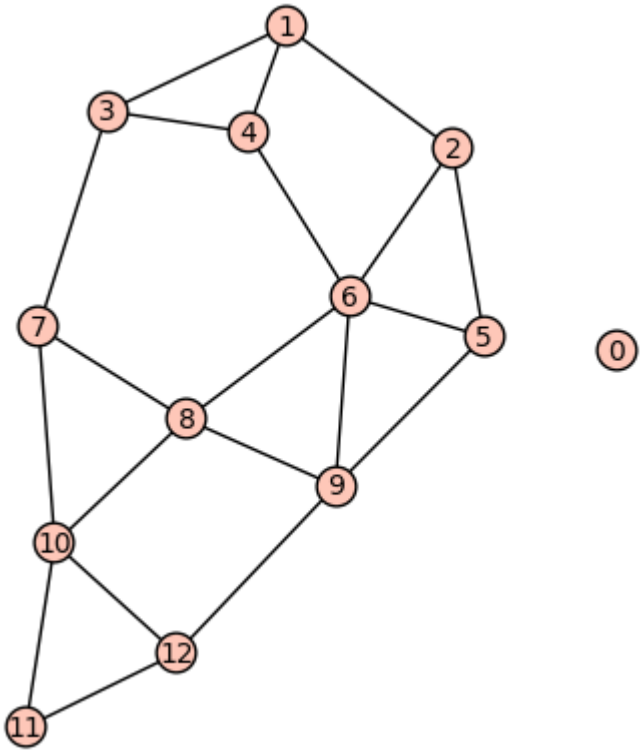
$$
\begin{aligned}
Min \quad & 599x_{1,2} + 180x_{1,3} + 497x_{1,4} + 691x_{2,5} + 420x_{2,6} + \\
& 432x_{3,4} + 893x_{3,7} + 345x_{4,6} + 440x_{5,6} + 554x_{5,9} + \\
& 432x_{6,8} + 621x_{6,9} + 280x_{7,8} + 500x_{7,10} + 577x_{8,9} + \\
& 290x_{8,10} + 268x_{9,12} + 116x_{10,11} + 403x_{10,12} + 314x_{11,12}
\end{aligned}
$$

$$
\begin{aligned}
ST \quad & x_{1,2} + x_{1,3} + x_{1,4} = 1 \\
& x_{2,5} + x_{2,6} - x_{1,2} = 0 \\
& x_{3,4} + x_{3,7} - x_{1,3} = 0 \\
& x_{4,6} - x_{1,4} - x_{3,4} = 0 \\
& x_{5,6} + x_{5,9} - x_{2,5} = 0 \\
& x_{6,8} + x_{6,9} - x_{2,6} - x_{4,6} - x_{5,6} = 0 \\
& x_{7,8} + x_{7,10} - x_{3,7} = 0 \\
& x_{8,9} + x_{8,10} - x_{6,8} - x_{7,8} = 0 \\
& x_{9,12} - x_{9,5} - x_{6,9} - x_{8,9} = 0 \\
& x_{10,11} + x_{10,12} - x_{7,10} - x_{8,10} = 0 \\
& x_{11,12} - x_{10,11} = 0 \\
& -x_{9,12} - x_{10,12} - x_{11,12} = 0 \\
& All \quad x_{i,j}\,'s = 0 \; or \; 1
\end{aligned}
$$

```
lpnet = MixedIntegerLinearProgram( maximization = False)
x = lpnet.new_variable(nonnegative = True, binary = True)
lpnet.set_objective(599*x[1,2] + 180*x[1,3] + 497*x[1,4] + 691*x[2,5] +
420*x[2,6] + 432*x[3,4] + 893*x[3,7] +

345*x[4,6] + 440*x[5,6] + 554*x[5,9] + 432*x[6,8] + 621*x[6,9] +
280*x[7,8] + 500*x[7,10] + 577*x[8,9] +

290*x[8,10] + 268*x[9,12] + 116*x[10,11] + 403*x[10,12] + 314*x[11,12])
lpnet.add_constraint(x[1,2] + x[1,3] + x[1,4]  == 1)
lpnet.add_constraint(x[2,5] + x[2,6] - x[1,2]  == 0)
lpnet.add_constraint(x[3,4] + x[3,7] - x[1,3]  == 0)
lpnet.add_constraint(x[4,6] - x[1,4] - x[3,4]  == 0)
lpnet.add_constraint(x[5,6] + x[5,9] - x[2,5]  == 0)
lpnet.add_constraint(x[6,8] + x[6,9] - x[2,6] - x[4,6] - x[5,6] == 0)
lpnet.add_constraint(x[7,8] + x[7,10] - x[3,7]  == 0)
lpnet.add_constraint(x[8,9] + x[8,10] - x[6,8] - x[7,8]  == 0)
lpnet.add_constraint(x[9,12] + x[9,5] - x[6,9] - x[8,9]  == 0)
lpnet.add_constraint(x[10,11] + x[10,12] - x[7,10] - x[8,10]  == 0)
lpnet.add_constraint(x[11,12] - x[10,11]  == 0)
lpnet.add_constraint(-x[9,12] - x[10,12] - x[11,12]  == -1)
lpnet.solve()
lpnet.get_values(x)

{(1, 2): 0.0,
 (1, 3): 0.0,
 (1, 4): 1.0,
 (2, 5): 0.0,
 (2, 6): 0.0,
 (3, 4): 0.0,
 (3, 7): 0.0,
 (4, 6): 1.0,
 (5, 6): 0.0,
 (5, 9): 0.0,
 (6, 8): 0.0,
 (6, 9): 1.0,
 (7, 8): 0.0,
 (7, 10): 0.0,
 (8, 9): 0.0,
 (8, 10): 0.0,
 (9, 5): 0.0,
 (9, 12): 1.0,
 (10, 11): 0.0,
 (10, 12): 0.0,
 (11, 12): 0.0}
```

الحل بإستخدام نظرية الرسم Graph Theory

```
g = Graph(12)
g.add_edge(1,2,599)
g.add_edge(1,3,180)
g.add_edge(1,4,497)
g.add_edge(2,5,691)
g.add_edge(2,6,420)
g.add_edge(3,4,432)
g.add_edge(3,7,893)
g.add_edge(4,6,345)
g.add_edge(5,6,440)
g.add_edge(5,9,554)
g.add_edge(6,8,432)
g.add_edge(6,9,621)
g.add_edge(7,8,280)
g.add_edge(7,10,500)
g.add_edge(8,9,577)
g.add_edge(8,10,290)
g.add_edge(9,12,268)
g.add_edge(10,11,116)
g.add_edge(10,12,403)
g.add_edge(11,12,314)
g.show()
from sage.graphs.distances_all_pairs import
shortest_path_all_pairs
shortest_path_all_pairs(g)
```



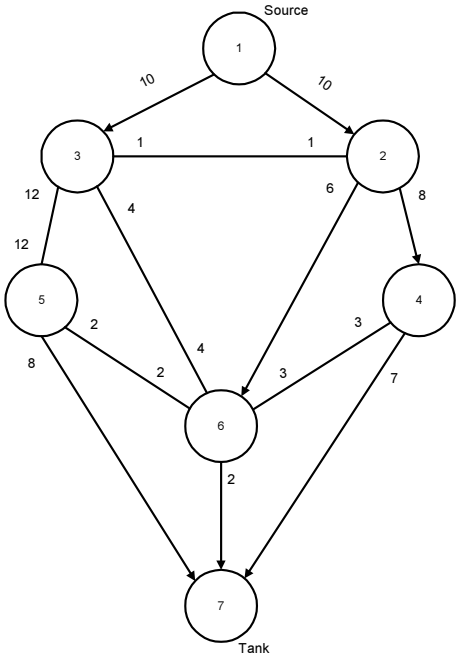

```
{0: {0: None,
  1: None,
  2: None,
  3: None,
  4: None,
  5: None,
  6: None,
  7: None,
  8: None,
  9: None,
  10: None,
  11: None,
  12: None},
 1: {0: None,
  1: None,
  2: 1,
  3: 1,
  4: 1,
  5: 2,
  6: 2,
  7: 3,
  8: 6,
  9: 5,
  10: 7,
  11: 10,
  12: 9},
 2: {0: None,
  1: 2,
  2: None,
  3: 1,
  4: 1,
```

```
 5: 2,
 6: 2,
 7: 3,
 8: 6,
 9: 5,
 10: 8,
 11: 12,
 12: 9},
3: {0: None,
 1: 3,
 2: 1,
 3: None,
 4: 3,
 5: 2,
 6: 4,
 7: 3,
 8: 7,
 9: 6,
 10: 7,
 11: 10,
 12: 10},
4: {0: None,
 1: 4,
 2: 1,
 3: 4,
 4: None,
 5: 6,
 6: 4,
 7: 3,
 8: 6,
 9: 6,
 10: 7,
 11: 10,
 12: 9},
5: {0: None,
 1: 2,
 2: 5,
 3: 1,
 4: 6,
 5: None,
 6: 5,
 7: 8,
 8: 6,
 9: 5,
 10: 8,
 11: 12,
 12: 9},
6: {0: None,
 1: 2,
 2: 6,
 3: 4,
 4: 6,
 5: 6,
 6: None,
 7: 8,
 8: 6,
 9: 6,
```

```
 10: 8,
 11: 10,
 12: 9},
7: {0: None,
 1: 3,
 2: 1,
 3: 7,
 4: 3,
 5: 6,
 6: 8,
 7: None,
 8: 7,
 9: 8,
 10: 7,
 11: 10,
 12: 10},
8: {0: None,
 1: 2,
 2: 6,
 3: 7,
 4: 6,
 5: 6,
 6: 8,
 7: 8,
 8: None,
 9: 8,
 10: 8,
 11: 10,
 12: 9},
9: {0: None,
 1: 2,
 2: 5,
 3: 4,
 4: 6,
 5: 9,
 6: 9,
 7: 8,
 8: 9,
 9: None,
 10: 8,
 11: 12,
 12: 9},
10: {0: None,
 1: 3,
 2: 6,
 3: 7,
 4: 3,
 5: 6,
 6: 8,
 7: 10,
 8: 10,
 9: 8,
 10: None,
 11: 10,
 12: 10},
11: {0: None,
 1: 3,
```

```
 2: 6,
 3: 7,
 4: 3,
 5: 9,
 6: 8,
 7: 10,
 8: 10,
 9: 12,
 10: 11,
 11: None,
 12: 11},
12: {0: None,
 1: 2,
 2: 5,
 3: 7,
 4: 6,
 5: 9,
 6: 9,
 7: 10,
 8: 9,
 9: 12,
 10: 12,
 11: 12,
 12: None}}
```

مثال 9:
مشكلة التدفق الأعظم



$Max \quad x_{1,2} + x_{1,3}$

St

$$x_{2,3} + x_{2,4} + x_{2,6} - x_{1,2} - x_{3,2} = 0$$

$$x_{3,2} + x_{3,5} + x_{3,6} - x_{1,3} - x_{2,3} - x_{6,3} = 0$$

$$x_{4,6} + x_{4,7} - x_{2,4} - x_{6,4} = 0$$

$$x_{5,6} + x_{5,7} - x_{3,5} - x_{6,5} = 0$$

$$x_{6,3} + x_{6,4} + x_{6,5} + x_{6,7} - x_{2,6} - x_{3,6} - x_{4,6} - x_{5,6} = 0$$

$$x_{1,2} \le 10$$

$$x_{1,3} \le 10$$

$$x_{2,3} \le 1$$

$$x_{2,4} \le 8$$

$$x_{2,6} \le 6$$

$$x_{3,2} \le 1$$

$$x_{3,5} \le 12$$

$$x_{3,6} \le 4$$

$$x_{4,6} \le 3$$

$$x_{4,7} \le 7$$

$$x_{5,6} \le 2$$

$$x_{5,7} \le 8$$

$$x_{6,3} \le 4$$

$$x_{6,4} \le 3$$

$$x_{6,5} \le 2$$

$$x_{6,7} \le 2$$

$All \quad x_{i,j}\text{'}s \ge 0$

```
lpnet = MixedIntegerLinearProgram( maximization = True)
x = lpnet.new_variable(nonnegative = True)
lpnet.set_objective(x[1,2] + x[1,3])
lpnet.add_constraint(x[2,3] + x[2,4] + x[2,6] - x[1,2] - x[3,2] == 0)
lpnet.add_constraint(x[3,2] + x[3,5] + x[3,6] - x[1,3] - x[2,3] -
x[6,3] == 0)
lpnet.add_constraint(x[4,6] + x[4,7] - x[2,4] - x[6,4] == 0)
lpnet.add_constraint(x[5,6] + x[5,7] - x[3,5] - x[6,5] == 0)
```

```
lpnet.add_constraint(x[6,3] + x[6,4] + x[6,5] + x[6,7] - x[2,6] -
x[3,6] - x[4,6] - x[5,6] == 0)
lpnet.add_constraint(x[1,2] <= 10)
lpnet.add_constraint(x[1,3] <= 10)
lpnet.add_constraint(x[2,3] <= 1)
lpnet.add_constraint(x[2,4] <= 8)
lpnet.add_constraint(x[2,6] <= 6)
lpnet.add_constraint(x[3,2] <= 1)
lpnet.add_constraint(x[3,5] <= 12)
lpnet.add_constraint(x[3,6] <= 4)
lpnet.add_constraint(x[4,6] <= 3)
lpnet.add_constraint(x[4,7] <= 7)
lpnet.add_constraint(x[5,6] <= 2)
lpnet.add_constraint(x[5,7] <= 8)
lpnet.add_constraint(x[6,3] <= 4)
lpnet.add_constraint(x[6,4] <= 3)
lpnet.add_constraint(x[6,5] <= 2)
lpnet.add_constraint(x[6,7] <= 2)
lpnet.solve()

17.0


lpnet.get_values(x)
{(1, 2): 7.0,
 (1, 3): 10.0,
 (2, 3): 0.0,
 (2, 4): 7.0,
 (2, 6): 0.0,
 (3, 2): 0.0,
 (3, 5): 8.0,
 (3, 6): 2.0,
 (4, 6): 0.0,
 (4, 7): 7.0,
 (5, 6): 0.0,
 (5, 7): 8.0,
 (6, 3): 0.0,
 (6, 4): 0.0,
 (6, 5): 0.0,
 (6, 7): 2.0}
```

مكتبة cvxopt لحل البرمجة الخطية و التربيعية

Solving a linear and quadratic programs with cvxopt

أولا: البرمجة الخطية LP:

`cvxopt.solvers.lp`(*c*, *G*, *h*[, *A*, *b*[, *solver*[, *primalstart*[, *dualstart*]]]])

لحل البرنامج الخطي الرئيسي و الثنائي primal and dual linear programs

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Gx + s = h \\ & Ax = b \\ & s \succeq 0 \end{aligned}$$

$$\begin{aligned} \text{maximize} \quad & -h^T z - b^T y \\ \text{subject to} \quad & G^T z + A^T y + c = 0 \\ & z \succeq 0. \end{aligned}$$

حيث x متغيرات القرار
c قيم دالة الهدف
G مصفوفة القيود و s المتغيرات الذائدة Slak
h الطرف الأيمن للقيود
A مصفوفة قيود للمسواة و b الطرف الأيمن

مثال:

solve the LP

$$\begin{aligned} \text{minimize} \quad & -4x_1 - 5x_2 \\ \text{subject to} \quad & 2x_1 + x_2 \le 3 \\ & x_1 + 2x_2 \le 3 \\ & x_1 \ge 0, \quad x_2 \ge 0. \end{aligned}$$

```
>>> import cvxopt
>>> from cvxopt import matrix, solvers
>>> c = matrix([-4., -5.])
>>> G = matrix([[2., 1., -1., 0.], [1., 2., 0., -1.]])
>>> h = matrix([3., 3., 0., 0.])
>>> sol = solvers.lp(c, G, h)
>>> print(sol['x'])
[ 1.00e+00]
[ 1.00e+00]
```

ثانيا: البرمجة التربيعية Quadratic Programming

`cvxopt.solvers.qp`(*P*, *q*[, *G*, *h*[, *A*, *b*[, *solver*[, *initvals*]]]])

لحل البرنامج التربيعي و الثنائي

$$\begin{aligned} \text{minimize} \quad & (1/2)x^T Px + q^T x \\ \text{subject to} \quad & Gx \le h \\ & Ax = b \end{aligned}$$

and

$$\begin{aligned} \text{maximize} \quad & -(1/2)(q + G^T z + A^T y)^T P^T (q + G^T z + A^T y) - h^T z - b^T y \\ \text{subject to} \quad & q + G^T z + A^T y \in \text{range}(P) \\ & z \succeq 0. \end{aligned}$$

مثال:

حل البرنامج التربيعي:

$$\begin{aligned} \text{minimize} \quad & 2x_1^2 + x_2^2 + x_1 x_2 + x_1 + x_2 \\ \text{subject to} \quad & x_1 \ge 0 \\ & x_2 \ge 0 \\ & x_1 + x_2 = 1 \end{aligned}$$

وهو على الشكل

$$\begin{aligned} \text{minimize} \quad & (1/2)x^T Px + q^T x \\ \text{subject to} \quad & Gx \le h \\ & Ax = b \end{aligned}$$

$$\text{minimize} \quad \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{bmatrix} 2.0 & 0.5 \\ 0.5 & 1.0 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1.0 & 1.0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

st

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \le \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1.0 & 1.0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1.0 \end{pmatrix}$$

حيث

$$P = \begin{bmatrix} 2.0 & 0.5 \\ 0.5 & 1.0 \end{bmatrix} \quad , q = \begin{pmatrix} 1.0 & 1.0 \end{pmatrix} \quad , G = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

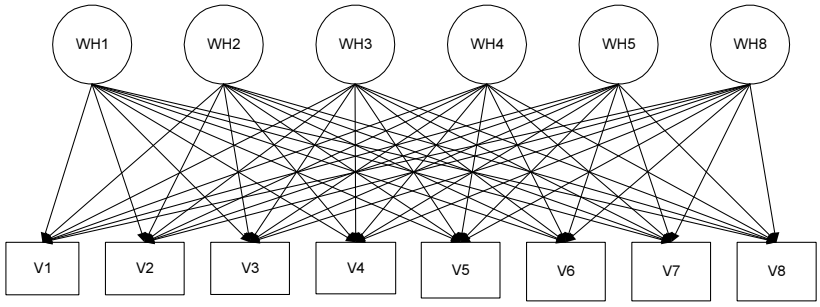
$$h = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad , A = \begin{pmatrix} 1.0 & 1.0 \end{pmatrix} \quad , b = \begin{pmatrix} 1.0 \end{pmatrix}$$

```
>>> import cvxopt
>>> from cvxopt import matrix, solvers
>>> P = 2*matrix([ [2, .5], [.5, 1] ])
>>> q = matrix([1.0, 1.0])
>>> G = matrix([[-1.0,0.0],[0.0,-1.0]])
>>> h = matrix([0.0,0.0])
>>> A = matrix([1.0, 1.0], (1,2))
>>> b = matrix(1.0)
>>> sol=solvers.qp(Q, p, G, h, A, b)
>>> print(sol['x'])
[ 2.50e-01]
[ 7.50e-01]
```

الفصل الثالث

مقدمة إلى لغة البرمجة الرياضية للأمثلية LINGO

سوف نستعرض قوة هذه اللغة بالمثال التالي الذي يستخدم الأمثلية في حل مشكلة نقل:
تمتلك شركة القطع الاسلكية 6 مخازن تمد 8 مشترين بقطع الغيار. لكل مخزن قدرة إمدادية (إستيعابية) لايستطيع تجاوزها كما ان كل زبون له طلب لابد أن يتحقق.
تريد الشركة تحديد عدد القطع التي تشحن من كل مخزن لكل زبون بحيث تكون تكلفة الشحن أقل مايمكن.
الشكل التالي يمثل المشكلة:



كما نلاحظ ان كل مخزن يستطيع الشحن لأي زبون وهذا يؤدي إلى 48 مسارشحن أو قوس. سوف نحتاج متغير لكل واحد من هذه الأقواس لتمثيل مقدار الشحنة المرسلة خلاله.

البيانات:

لدينا البيانات التالية:

1- بيانات عن المقدرة الإستيعابية:

Warehouse	Widgets On Hand
1	60
2	55
3	51
4	43
5	41
6	52

2- بيانات طلب الزبائن:

Vendor	Widget Demand
1	35
2	37
3	22
4	32
5	41
6	32
7	43
8	38

3- تكاليف الشحن للقطعة:

	V1	V2	V3	V4	V5	V6	V7	V8
WH1	6	2	6	7	4	2	5	9
WH2	4	9	5	3	8	5	8	2
WH3	5	2	1	9	7	4	3	3
WH4	7	6	7	3	9	2	7	1
WH5	2	3	9	5	7	2	6	5
WH6	5	5	2	2	8	1	4	3

دالة الهدف The Objective Function :

لتشكيل النموذج سوف نبدأ ببناء دالة الهدف. المطلوب تقليل تكلفة الشحن ولهذا سنستخدم المتغير $x(i,j)$ لكي يمثل عدد القطع التي تشحن من الخزن i إلى الزبون j ولهذا فإن دالة الهدف تكتب علي الشكل:

Minimize

$$6x(1,1)+2x(1,2)+6x(1,3)+7x(1,4)+4x(1,5)+2x(1,6)+5x(1,7)+9x(1,8)+$$
$$4x(2,1)+9x(2,2)+5x(2,3)+3x(2,4)+8x(2,5)+5x(2,6)+8x(2,7)+2x(2,8)+$$
$$5x(3,1)+2x(3,2)+x(3,3)+9x(3,4)+7x(3,5)+4x(3,6)+3x(3,7)+3x(3,8)+$$
$$7x(4,1)+6x(4,2)+7x(4,3)+3x(4,4)+9x(4,5)+2x(4,6)+7x(4,7)+x(4,8)+$$
$$2x(5,1)+3x(5,2)+9x(5,3)+5x(5,4)+7x(5,5)+2x(5,6)+6x(5,7)+5x(5,8)+$$
$$5x(6,1)+5x(6,2)+2x(6,3)+2x(6,4)+8x(6,5)+x(6,6)+4x(6,7)+3x(6,8)$$

كما نشاهد فإن دالة الهدف معقدة جدا وذلك فقط لـ 6 مخازن و 8 زبائن فكيف سوف تكون لو كان لدينا عشرات بل مئات المخازن وآلاف من الزبائن؟ كما أن إدخال مثل هذه الصيغة في برنامج حاسب سيكون بالتأكيد معرضا للخطأ.

الصيغة السابقة يمكن وضعها على شكل الصيغة الرياضية التالية:

$$Mimize \quad \sum_{i=1}^{6}\sum_{j=1}^{8} c_{ij}x(i,j)$$

حيث c_{ij} تكلفة شحن القطعة الواحدة من مخزن i إلى الزبون j .

وبالمثل فإن لغة النمذجة LINGO تسمح بتمثيل دالة الهدف بشكل مختصر سهل الكتابة والفهم كالتالي:

```
MIN = @SUM(LINKS(I,J): COST(I,J)*VOLUME(I,J));
```

حيث `VOLUME(I,J)` يمثل عدد القطع التي تشحن من المخزن i إلى الزبون j و `COST(I,J)` تكلفة شحن القطعة الواحدة من مخزن i إلى الزبون j و الجمع يكون على كل الروابط `LINKS(I,J)` بينهما. الجدول التالي مقارنة بين الصيغة الرياضية ولغة LINGO

Math Notation	LINGO Syntax
Minimize	MIN =
$\sum_{i=1}^{6}\sum_{j=1}^{8}$	@SUM(LINKS(I,J):)
c_{ij}	COST(I,J)
$x(i,j)$	VOLUME(I,J)

القيود Constraints:

بعد صياغتنا لدالة الهدف خطوتنا التالية صياغة القيود Constraints للنموذج. مجموعة القيود الأولى تضمن إستلام كل زبون لعدد القطع التي طلبها وسوف نسمي هذه القيود قيود الطلب Demand Constraints مجموعة القيود التالية والتي نسميها قيود السعة Capacity Constraints تضمن عدم شحن قطع من مخزن تزيد عن المقدار الذى يمتلكة.

بدأ بقيد الطلب للزبون الأول فإننا نحتاج إلى جمع كل الشحنات من جميع المخازن وجعلها مساوية لطلب الزبون الأول 35 قطعة أي

$$x(1,1)+x(2,1)+x(3,1)+x(4,1)+x(5,1)+x(6,1)=35$$

و هكذا لبقية الزبائن

$$x(1,2)+x(2,2)+x(3,2)+x(4,2)+x(5,2)+x(6,2)=37$$
$$x(1,3)+x(2,3)+x(3,3)+x(4,3)+x(5,3)+x(6,3)=22$$
$$x(1,4)+x(2,4)+x(3,4)+x(4,4)+x(5,4)+x(6,4)=32$$
$$x(1,5)+x(2,5)+x(3,5)+x(4,5)+x(5,5)+x(6,5)=41$$
$$x(1,6)+x(2,6)+x(3,6)+x(4,6)+x(5,6)+x(6,6)=32$$
$$x(1,7)+x(2,7)+x(3,7)+x(4,7)+x(5,7)+x(6,7)=43$$
$$x(1,8)+x(2,8)+x(3,8)+x(4,8)+x(5,8)+x(6,8)=38$$

و هنا ينطبق نفس التعليق الذي تبع صياغة دالة الهدف.

قيود الطلب تكتب بصيغة رياضية بسيطة كالتالي:

$$\sum_{i=1}^{6} x(i,j)=d_j,\ j=1,2,...,8$$

حيث d_j عدد القطع المطلوبة من الزبون j .

وبلغة النمذجة LINGO تكتب قيود الطلب

```
@FOR(VENDORS(J):
  @SUM(WAREHOUSES(I): VOLUME(I,J)) = DEMAND(J));
```

هذه العبارة بلغة النمذجة LINGO تمثل مجموعة القيود السابقة (بل يمكن أن تمثل مجموعة من آلاف الزبائن ومئآت المخازن ايضا). العبارة السابقة تدل على انه لكل الزبائن VENDORS فإن مجموع القطع التي تشحن VOLUME من كل مخزن WAREHOUSES إلى ذلك الزبون يجب أن تساوي لطلب DEMAND ذلك الزبون. لاحظ مدى تشابه هذه العبارة مع الصيغة الرياضية. الجدول التالي يعطى مقارنة بينهما:

Math Notation	LINGO Syntax
$j = 1,2,...,8$	@FOR(VENDORS(J):)
$\sum_{i=1}^{6}$	@SUM(WAREHOUSES(I):)
$x(i,j)$	VOLUME(I,J)
$=$	=
d_j	DEMAND(J)

سوف نصيغ الآن قيود السعة والتي تكتب بصيغة الرياضيات

$$\sum_{j=1}^{8} x(i,j) \le p_j, i = 1,2,...,6$$

حيث p_j هي سعة المخزن j .(لاحظ ان هذه الصيغة الرياضية تمثل مجموعة من 8 متراجحات)
وبلغة النمذجة LINGO تكتب قيود السعة

```
@FOR( WAREHOUSES( I):
 @SUM( VENDORS( J): VOLUME( I, J)) <= CAPACITY( I));
```

أي لكل مخزن WAREHOUSES فإن مجموع القطع المشحونة VOLUME للزبائن VENDORS يجب الا تزيد عن سعة CAPACITY ذلك المخزن.
الآن نكتب النموذج كاملا بلغة النمذجة LINGO كالتالي:

```
MODEL:

MIN = @SUM( LINKS( I, J):
    COST( I, J) * VOLUME( I, J));
@FOR( VENDORS( J):
    @SUM( WAREHOUSES( I): VOLUME( I, J)) = DEMAND( J));
@FOR( WAREHOUSES( I):
    @SUM( VENDORS( J): VOLUME( I, J)) <= CAPACITY( I));

END
```

لنسمي هذا Model: WIDGETS
لإكمال النموذج يجب ان نعرف البيانات فيما يسمى بقسم المجموعات Sets Section وقسم البيانات Data Section والتى نشرحها بالتفصيل كالتالي:

<u>تعريف المجموعات:</u>
عند قيامنا بالنمذجة في الحياة العملية فإننا نقابل مجموعة او اكثر من الأشياء التي لها علاقة ببعضها البعض مثل الزبائن ، المصانع ، السيارات ، الموظفين الخ. في الغالب إذا طبق قيد على عضو من المجموعة فإن هذا القيد يطبق بالتمام على الأعضاء الآخرين في المجموعة. هذا المبدأ هو من أساسيات لغة النمذجة LINGO إذ يعطى المقدرة على تعريف مجموعات من الأشياء المترابطة في قسم المجموعات.
قسم المجموعات يبدأ بكلمة الأساس Keyword التي هي SETS توضع في سطر منفردة وينتهي القسم بكلمة الأساس ENDSETS ايضا على سطر منفرد. بعد تعريف أعضاء المجموعة فإن LINGO يمتلك مجموعة من دوال الدورة Set Looping Functions (مثل @FOR) والتي تطبق بعض العمليات على جميع اعضاء المجموعة بإستخدام عبارة واحدة فقط.
في حالة نموذجنا فإننا قمنا بتكوين ثلاثة مجموعات هي:
1- المخازن Warehouses

2- الزبائن Vendors
3- توصيلات او أقواس الشحن من كل مخزن إلى كل زبون Shipping Arcs
وبلغة النمذجة LINGO تعرف هذه المجموعات كالتالي:

```
SETS:

   WAREHOUSES / WH1 WH2 WH3 WH4 WH5 WH6/: CAPACITY;
   VENDORS / V1 V2 V3 V4 V5 V6 V7 V8/   : DEMAND;
   LINKS( WAREHOUSES, VENDORS): COST, VOLUME;

ENDSETS
```

السطر الأول بعد كلمة الأساس SETS يقول ان المجموعة WAREHOUSES أعضائها هم WH1, WH2,…,WH6 كل منها له صفة Attribute تسمى CAPACITY . السطر التالي يعرف مجموعة الزبائن VENDORS والتي أعضائها V1,V2,…,V8 والتي لكل واحد منها الصفة DEMAND . المجموعة الأخيرة المسماة LINKS تمثل 48 توصيلة او خط شحن كل توصيلة لها صفات COST و VOLUME التابعة لها.
لاحظ الإختلاف في التركيب اللغوي Syntax بين المجموعة هذة والمجاميع السابقة فبوضع LINKS(WAREHOUSES, VENDORS) فإننا نخبر LINGO أن مجموعة LINKS تستنبط من المجموعتين WAREHOUSES و VENDORS وبالتالي فإن LINGO يولد 48 من الأزواج المرتبة (WAREHOUSES, VENDORS) ويجعلها أعضاء في المجموعة LINKS .

<u>إدخال البيانات:</u>
تدخل البيانات في قسم البيانات كالتالي:

```
DATA:

   CAPACITY = 60 55 51 43 41 52;
   DEMAND = 35 37 22 32 41 32 43 38;
   COST  =  6 2 6 7 4 2 5 9
            4 9 5 3 8 5 8 2
            5 2 1 9 7 4 3 3
            7 6 7 3 9 2 7 1
            2 3 9 5 7 2 6 5
            5 5 2 2 8 1 4 3;

ENDDATA
```

يبدأ قسم البيانات بكلمة الأساس DATA على سطر منفرد وينتهي بكلمة الأساس ENDDATA على سطر منفرد ايضا. كل من الصفات CAPACITY للمجموعة WAREHOUSES و DEMAND للمجموعة VENDORS اسند لها قيم بشكل مباشر. الصفة COST للمجموعة LINKS ثنائية الأبعاد يتم قرائتها بواسطة LINGO بزيادة الدورة الخارجية أسرع من الدورة الداخلية أي يقرأ القيم على الترتيب التالي:

```
COST(WH1,V1), COST(WH1,V2), COST(WH1,V3),…,COST(WH1,V8),
COST(WH2,V1), COST(WH2,V2), COST(WH2,V3),…,COST(WH2,V8),
…
COST(WH6,V1), COST(WH6,V2 ), COST(WH6,V3),…,COST(WH6,V8)
```

لاحظ اننا هنا أدخلنا البيانات بشكل مباشر ولكن يمكننا قرائتها من ملف او إستيرادها من صفحة نشر مثل EXCEL وسوف نتطرق لهذا لاحقا.

<u>النموذج بشكله الكامل:</u>

```
MODEL:
! A 6 Warehouse 8 Vendor Transportation Problem;
```

```
SETS:
   WAREHOUSES / WH1 WH2 WH3 WH4 WH5 WH6/: CAPACITY;
   VENDORS / V1 V2 V3 V4 V5 V6 V7 V8/   : DEMAND;
   LINKS( WAREHOUSES, VENDORS): COST, VOLUME;
ENDSETS
! The objective;
   MIN = @SUM( LINKS( I, J):
    COST( I, J) * VOLUME( I, J));
! The demand constraints;
   @FOR( VENDORS( J):
    @SUM( WAREHOUSES( I): VOLUME( I, J)) =
     DEMAND( J));
! The capacity constraints;
   @FOR( WAREHOUSES( I):
    @SUM( VENDORS( J): VOLUME( I, J)) <=
     CAPACITY( I));
! Here is the data;
DATA:
   CAPACITY = 60 55 51 43 41 52;
   DEMAND = 35 37 22 32 41 32 43 38;
   COST = 6 2 6 7 4 2 5 9
          4 9 5 3 8 5 8 2
          5 2 1 9 7 4 3 3
          7 6 7 3 9 2 7 1
          2 3 9 5 7 2 6 5
          5 5 2 2 8 1 4 3;
ENDDATA
END
```

لاحظ اننا أضفنا تعليقات Comments لشرح الأجزاء المختلفة من البرنامج وأسطر التعليق تبدأ بالرمز !

المخرجات:

```
Global optimal solution found at step:             20
 Objective value:                         664.0000

                     Variable           Value        Reduced Cost
                CAPACITY( WH1)        60.00000           0.0000000
                CAPACITY( WH2)        55.00000           0.0000000
                CAPACITY( WH3)        51.00000           0.0000000
                CAPACITY( WH4)        43.00000           0.0000000
                CAPACITY( WH5)        41.00000           0.0000000
                CAPACITY( WH6)        52.00000           0.0000000
                   DEMAND( V1)        35.00000           0.0000000
                   DEMAND( V2)        37.00000           0.0000000
                   DEMAND( V3)        22.00000           0.0000000
                   DEMAND( V4)        32.00000           0.0000000
                   DEMAND( V5)        41.00000           0.0000000
                   DEMAND( V6)        32.00000           0.0000000
                   DEMAND( V7)        43.00000           0.0000000
                   DEMAND( V8)        38.00000           0.0000000
                COST( WH1, V1)        6.000000           0.0000000
                COST( WH1, V2)        2.000000           0.0000000
```

COST(WH1, V3)	6.000000	0.0000000
COST(WH1, V4)	7.000000	0.0000000
COST(WH1, V5)	4.000000	0.0000000
COST(WH1, V6)	2.000000	0.0000000
COST(WH1, V7)	5.000000	0.0000000
COST(WH1, V8)	9.000000	0.0000000
COST(WH2, V1)	4.000000	0.0000000
COST(WH2, V2)	9.000000	0.0000000
COST(WH2, V3)	5.000000	0.0000000
COST(WH2, V4)	3.000000	0.0000000
COST(WH2, V5)	8.000000	0.0000000
COST(WH2, V6)	5.000000	0.0000000
COST(WH2, V7)	8.000000	0.0000000
COST(WH2, V8)	2.000000	0.0000000
COST(WH3, V1)	5.000000	0.0000000
COST(WH3, V2)	2.000000	0.0000000
COST(WH3, V3)	1.000000	0.0000000
COST(WH3, V4)	9.000000	0.0000000
COST(WH3, V5)	7.000000	0.0000000
COST(WH3, V6)	4.000000	0.0000000
COST(WH3, V7)	3.000000	0.0000000
COST(WH3, V8)	3.000000	0.0000000
COST(WH4, V1)	7.000000	0.0000000
COST(WH4, V2)	6.000000	0.0000000
COST(WH4, V3)	7.000000	0.0000000
COST(WH4, V4)	3.000000	0.0000000
COST(WH4, V5)	9.000000	0.0000000
COST(WH4, V6)	2.000000	0.0000000
COST(WH4, V7)	7.000000	0.0000000
COST(WH4, V8)	1.000000	0.0000000
COST(WH5, V1)	2.000000	0.0000000
COST(WH5, V2)	3.000000	0.0000000
COST(WH5, V3)	9.000000	0.0000000
COST(WH5, V4)	5.000000	0.0000000
COST(WH5, V5)	7.000000	0.0000000
COST(WH5, V6)	2.000000	0.0000000
COST(WH5, V7)	6.000000	0.0000000
COST(WH5, V8)	5.000000	0.0000000
COST(WH6, V1)	5.000000	0.0000000
COST(WH6, V2)	5.000000	0.0000000
COST(WH6, V3)	2.000000	0.0000000
COST(WH6, V4)	2.000000	0.0000000
COST(WH6, V5)	8.000000	0.0000000
COST(WH6, V6)	1.000000	0.0000000
COST(WH6, V7)	4.000000	0.0000000
COST(WH6, V8)	3.000000	0.0000000
VOLUME(WH1, V1)	0.0000000	5.000000
VOLUME(WH1, V2)	19.00000	0.0000000
VOLUME(WH1, V3)	0.0000000	5.000000
VOLUME(WH1, V4)	0.0000000	7.000000
VOLUME(WH1, V5)	41.00000	0.0000000
VOLUME(WH1, V6)	0.0000000	2.000000
VOLUME(WH1, V7)	0.0000000	2.000000
VOLUME(WH1, V8)	0.0000000	10.00000
VOLUME(WH2, V1)	1.000000	0.0000000
VOLUME(WH2, V2)	0.0000000	4.000000
VOLUME(WH2, V3)	0.0000000	1.000000

VOLUME(WH2, V4)	32.00000	0.0000000
VOLUME(WH2, V5)	0.0000000	1.000000
VOLUME(WH2, V6)	0.0000000	2.000000
VOLUME(WH2, V7)	0.0000000	2.000000
VOLUME(WH2, V8)	0.0000000	0.0000000
VOLUME(WH3, V1)	0.0000000	4.000000
VOLUME(WH3, V2)	11.00000	0.0000000
VOLUME(WH3, V3)	0.0000000	0.0000000
VOLUME(WH3, V4)	0.0000000	9.000000
VOLUME(WH3, V5)	0.0000000	3.000000
VOLUME(WH3, V6)	0.0000000	4.000000
VOLUME(WH3, V7)	40.00000	0.0000000
VOLUME(WH3, V8)	0.0000000	4.000000
VOLUME(WH4, V1)	0.0000000	4.000000
VOLUME(WH4, V2)	0.0000000	2.000000
VOLUME(WH4, V3)	0.0000000	4.000000
VOLUME(WH4, V4)	0.0000000	1.000000
VOLUME(WH4, V5)	0.0000000	3.000000
VOLUME(WH4, V6)	5.000000	0.0000000
VOLUME(WH4, V7)	0.0000000	2.000000
VOLUME(WH4, V8)	38.00000	0.0000000
VOLUME(WH5, V1)	34.00000	0.0000000
VOLUME(WH5, V2)	7.000000	0.0000000
VOLUME(WH5, V3)	0.0000000	7.000000
VOLUME(WH5, V4)	0.0000000	4.000000
VOLUME(WH5, V5)	0.0000000	2.000000
VOLUME(WH5, V6)	0.0000000	1.000000
VOLUME(WH5, V7)	0.0000000	2.000000
VOLUME(WH5, V8)	0.0000000	5.000000
VOLUME(WH6, V1)	0.0000000	3.000000
VOLUME(WH6, V2)	0.0000000	2.000000
VOLUME(WH6, V3)	22.00000	0.0000000
VOLUME(WH6, V4)	0.0000000	1.000000
VOLUME(WH6, V5)	0.0000000	3.000000
VOLUME(WH6, V6)	27.00000	0.0000000
VOLUME(WH6, V7)	3.000000	0.0000000
VOLUME(WH6, V8)	0.0000000	3.000000

Row	Slack or Surplus	Dual Price
1	664.0000	1.000000
2	0.0000000	-4.000000
3	0.0000000	-5.000000
4	0.0000000	-4.000000
5	0.0000000	-3.000000
6	0.0000000	-7.000000
7	0.0000000	-3.000000
8	0.0000000	-6.000000
9	0.0000000	-2.000000
10	0.0000000	3.000000
11	22.00000	0.0000000
12	0.0000000	3.000000
13	0.0000000	1.000000
14	0.0000000	2.000000
15	0.0000000	2.000000

أسماء القيود:

يوفر لنا LINGO المقدرة على تسمية القيود وهذا يعطى الميزات التالية، أولا تستخدم أسماء القيود في تقرير الحل وهذا يساعد على سهولة تفسير النتائج، ثانيا الكثير من رسائل الأخطاء تعطي القيود بأسمائها فإذا لم يكن للقيود أسماء فإن تعقب وإيجاد هذه الأخطاء سيكون صعبا جدا. ملاحظة: تسمية القيود غير إلزامية إذ أن LINGO يسمي القيود ذاتيا By Default ويعطيها أسماء من المؤشر الداخلي للقيود وقد يكون هذا الإسم بشكل لايدل إطلاقا عن القيد مما يجعل تفسير رسائل الأخطاء أكثر غموضا ولهذا فإننا نحث المنمذج على إستخدام أسماء مناسبة تدل بشكل ما على كل أنواع القيود المستخدمة في النموذج.

تسمية القيود سهل جدا إذ كل ما يتطلب هو إعطاء إسم له معنى في أول القيد محاط بأقواس مربعة [ ] الأسماء يجب أن تتبع قواعد لغة LINGO إذ يجب أن تبدأ بحرف أو أكثر من الحروف [A –Z] ويمكن ان تحوى ارقاما من [9 – 0] أو [_] ولايزيد عن 32 رمز.

أمثلة على أسماء القيود:

1- `[OBJECTIVE] MIN = X;` يعطي الإسم OBJECTIVE لدالة الهدف.

2-

```
@FOR( LINKS( I, J): [DEMAND_ROW]
 @SUM( SOURCES( I): SHIP( I, J)) >= DEMAND( J));
```

يعطي الإسم DEMAND_ROW لقيد الطلب في مشكلة النقل.

ولإعطاء أمثلة أكثر فإننا نعيد كتابة نموذجنا السابق مع إدخال أسماء للقيود

```
MODEL:
! A 6 Warehouse 8 Vendor Transportation Problem;
SETS:
   WAREHOUSES / WH1 WH2 WH3 WH4 WH5 WH6/: CAPACITY;
   VENDORS / V1 V2 V3 V4 V5 V6 V7 V8/   : DEMAND;
   LINKS( WAREHOUSES, VENDORS): COST, VOLUME;
ENDSETS
! The objective;
   [OBJECTIVE] MIN = @SUM( LINKS( I, J):
    COST( I, J) * VOLUME( I, J));
! The demand constraints;
   @FOR( VENDORS( J): [DEMAND_ROW]
    @SUM( WAREHOUSES( I): VOLUME( I, J)) = DEMAND( J));
! The capacity constraints;
   @FOR( WAREHOUSES( I): [CAPACITY_ROW]
    @SUM( VENDORS( J): VOLUME( I, J)) <= CAPACITY( I));
! Here is the data;
DATA:
   CAPACITY = 60 55 51 43 41 52;
   DEMAND = 35 37 22 32 41 32 43 38;
   COST = 6 2 6 7 4 2 5 9
          4 9 5 3 8 5 8 2
          5 2 1 9 7 4 3 3
          7 6 7 3 9 2 7 1
          2 3 9 5 7 2 6 5
          5 5 2 2 8 1 4 3;
ENDDATA
END
```

التقرير التالي اكثر سهولة في الفهم والتفسير عن التقرير السابق:

```
Row   Slack or Surplus   Dual Price
```

```
          OBJECTIVE        664.0000         1.000000
    DEMAND_ROW( V1)       0.0000000       -4.000000
    DEMAND_ROW( V2)       0.0000000       -5.000000
    DEMAND_ROW( V3)       0.0000000       -4.000000
    DEMAND_ROW( V4)       0.0000000       -3.000000
    DEMAND_ROW( V5)       0.0000000       -7.000000
    DEMAND_ROW( V6)       0.0000000       -3.000000
    DEMAND_ROW( V7)       0.0000000       -6.000000
    DEMAND_ROW( V8)       0.0000000       -2.000000
 CAPACITY_ROW( WH1)       0.0000000        3.000000
 CAPACITY_ROW( WH2)        22.00000        0.000000
 CAPACITY_ROW( WH3)       0.0000000        3.000000
 CAPACITY_ROW( WH4)       0.0000000        1.000000
 CAPACITY_ROW( WH5)       0.0000000        2.000000
 CAPACITY_ROW( WH6)       0.0000000        2.000000
```

عنوان النموذج:

يفضل أن يعطى النموذج عنوان لتعريفه في تقرير الحل ويكون العنوان في السطر التالي لكلمة الأساس :MODEL ويسبقه كلمة أساس TITLE وينتهي السطر بفاصلة منقوطة (;) مثل:

```
MODEL:
TITLE Widgets;
! A 6 Warehouse 8 Vendor Transportation Problem;
SETS:
   WAREHOUSES / WH1 WH2 WH3 WH4 WH5 WH6/: CAPACITY;
      .
      .
```

الفصل الرابع

أمثلة وحالات دراسة على لغة النمذجة LINGO :

هذه الأمثلة والحالات الدراسية سوف تشرح بالتفصيل في المحاضرات.

مثال 1:

حل النموذج التالي بإستخدام LINGO

$$Max \quad 8x_1 + 5x_2$$

$$St \quad 2x_1 + x_2 \le 1000$$

$$3x_1 + 4x_2 \le 2400$$

$$x_1 + x_2 \le 700$$

$$x_1 - x_2 \le 350$$

$$x_1, x_2 \ge 0$$

نرتب النموذج في جدول كالتالي لتحديد المجموعات والبيانات:

	C1	C2	S
R1	2	1	1000
R2	3	4	2400
R3	1	1	700
R4	1	-1	350
P	8	5	

ويكون النموذج كالتالي

```
MODEL:
SETS:
     R / R1 R2 R3 R4 / : S;
     C / C1  C2 / : P , DV;
     RXC(R,C) : Coef;
ENDSETS
DATA:
     S = 1000 2400 700 350;
     P = 8  5;
     Coef = 2   1
            3    4
            1    1
            1   -1;
ENDDATA

MAX = @SUM( C(I): P(I) * DV(I));
@FOR( R(I): @SUM( C(J): Coef(I,J) * DV(J)) <= S(I));
```

التقرير:

```
Global optimal solution found at iteration:           3
  Objective value:                                4360.000
                      Variable           Value        Reduced Cost
                        S( R1)        1000.000            0.000000
                        S( R2)        2400.000            0.000000
                        S( R3)        700.0000            0.000000
                        S( R4)        350.0000            0.000000
                        P( C1)        8.000000            0.000000
                        P( C2)        5.000000            0.000000
                       DV( C1)        320.0000            0.000000
                       DV( C2)        360.0000            0.000000
                 COEF( R1, C1)        2.000000            0.000000
                 COEF( R1, C2)        1.000000            0.000000
                 COEF( R2, C1)        3.000000            0.000000
                 COEF( R2, C2)        4.000000            0.000000
                 COEF( R3, C1)        1.000000            0.000000
                 COEF( R3, C2)        1.000000            0.000000
                 COEF( R4, C1)        1.000000            0.000000
                 COEF( R4, C2)       -1.000000            0.000000
                           Row    Slack or Surplus      Dual Price
                             1        4360.000            1.000000
                             2        0.000000            3.40000
                             3        0.000000           0.4000000
                             4        20.00000            0.000000
                             5        390.0000            0.000000
```

مثال 2:

حل النموذج التالي بإستخدام LINGO :

$$\begin{aligned} Minimize \quad & 0.6X_1 + 0.5X_2 \\ ST \quad & 20X_1 + 50X_2 \geq 100 \\ & 25X_1 + 25X_2 \geq 100 \\ & 50X_1 + 10X_2 \geq 100 \\ & X_1, X_2 \geq 0 \end{aligned}$$

نرتب النموذج في جدول:

	C1	C2	S
R1	20	50	100
R2	25	25	100
R3	50	10	100
P	0.6	0.5	

ويكون النموذج:

```
MODEL:
SETS:
        R / R1 R2 R3 / : S;
        C / C1 C2 / : P , DV;
        RXC(R,C) : Coef;
ENDSETS
DATA:
        S= 100 100 100;
        P = 0.6 0.5;
        Coef = 20 50
                25 25
                50 10;
ENDDATA
MIN = @SUM( C(I) : P(I) * DV(I));
@FOR( R(I) : @SUM( C(J): Coef(I,J) *DV(J)) >= S(I));
```

التقرير:

```
Global optimal solution found at iteration:            3
  Objective value:                                 2.150000
                       Variable           Value        Reduced Cost
                         S( R1)         100.0000            0.000000
                         S( R2)         100.0000            0.000000
                         S( R3)         100.0000            0.000000
                         P( C1)        0.6000000            0.000000
                         P( C2)        0.5000000            0.000000
                        DV( C1)         1.500000            0.000000
                        DV( C2)         2.500000            0.000000
                  COEF( R1, C1)         20.00000            0.000000
                  COEF( R1, C2)         50.00000            0.000000
                  COEF( R2, C1)         25.00000            0.000000
                  COEF( R2, C2)         25.00000            0.000000
                  COEF( R3, C1)         50.00000            0.000000
                  COEF( R3, C2)         10.00000            0.000000
                             Row    Slack or Surplus      Dual Price
                               1         2.150000           -1.000000
                               2         55.00000            0.000000
                               3         0.000000           -0.1900000E-01
                               4         0.000000           -0.2500000E-02
```

مثال 3:

حل النموذج التالي بإستخدام LINGO :

$$\begin{aligned} Max. \quad & 12000X_1 + 20000X_2 \\ St. \quad & 2X_1 + 6X_2 \le 27 \\ & X_2 \ge 2 \\ & 3X_1 + X_2 \le 19 \\ & X_1, X_2 \ge 0 \text{ and integers} \end{aligned}$$

نرتب النموذج في جدول:

	C1	C2	LS	GS
LR1	2	6	24	
LR2	3	1	19	
GR1	0	1		2
P	12000	20000		

ويكتب النموذج كالتالي:

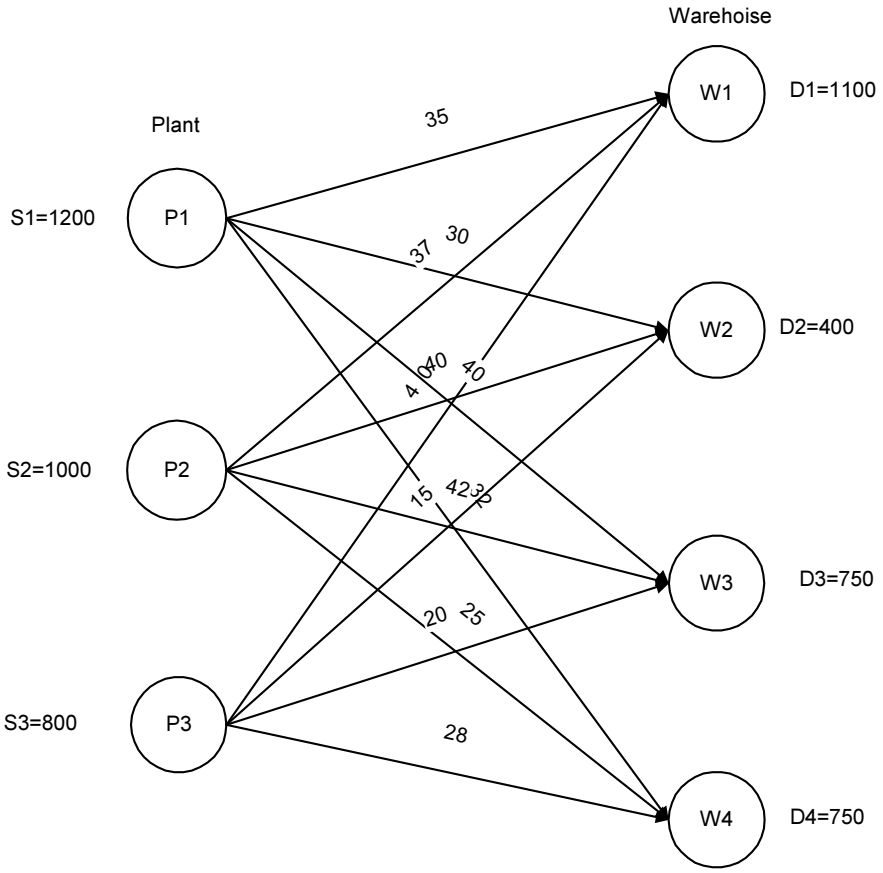
```
MODEL:
SETS:
     LR / LR1 LR2 / : LS;
     GR / GR1 / : GS;
     C / C1 C2 / : P , DV;
     LRXC(LR,C) : CO1;
     GRXC(GR,C) : CO2;
ENDSETS
DATA:
     LS = 27 19;
     GS = 2;
     P = 12000 20000;
     CO1 = 2 6
                 3 1;
     CO2 = 0 1;
ENDDATA
MAX = @SUM( C(I) : P(I) * DV(I));
@FOR(LR(I): @SUM( C(J): CO1(I,J) * DV(J)) <= LS(I));
@FOR(GR(I): @SUM( C(J): CO2(I,J) * DV(J)) >= GS(I));
@FOR(C(I) : @GIN(DV(I)));
```

التقرير:

```
Global optimal solution found at iteration:               0
  Objective value:                                  108000.0
                      Variable           Value        Reduced Cost
                      LS( LR1)        27.00000            0.000000
                      LS( LR2)        19.00000            0.000000
                      GS( GR1)        2.000000            0.000000
                        P( C1)        12000.00            0.000000
                        P( C2)        20000.00            0.000000
                       DV( C1)        4.000000           -12000.00
                       DV( C2)        3.000000           -20000.00
                 CO1( LR1, C1)        2.000000            0.000000
                 CO1( LR1, C2)        6.000000            0.000000
                 CO1( LR2, C1)        3.000000            0.000000
                 CO1( LR2, C2)        1.000000            0.000000
                 CO2( GR1, C1)        0.000000            0.000000
                 CO2( GR1, C2)        1.000000            0.000000
                           Row    Slack or Surplus      Dual Price
                             1        108000.0            1.000000
                             2        1.000000            0.000000
                             3        4.000000            0.000000
                             4        1.000000            0.000000
```

مثال 4:

حل مشكلة النقل التاليه بإستخدام LINGO :



والتي سبق أن كتبناه على شكل LP

$$Min\ 35X_{11}+30X_{12}+40X_{13}+32X_{14}+37X_{21}+40X_{22}+42X_{23}+25X_{24}+40X_{31}+15X_{32}+20X_{33}+28X_{34}$$

$$St \quad X_{11}+X_{12}+X_{13}+X_{14} \le 1200$$

$$X_{21}+X_{22}+X_{23}+X_{24} \le 1000$$

$$X_{31}+X_{32}+X_{33}+X_{34} \le 800$$

$$X_{11}+X_{21}+X_{31}=1100$$

$$X_{12}+X_{22}+X_{32}=400$$

$$X_{13}+X_{23}+X_{33}=750$$

$$X_{14}+X_{24}+X_{34}=750$$

$$X_{ij} \ge 0, \quad for\ all\ i\ and\ j$$

ونرتب النموذج في جدول:

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	SPL	DM
SR1	1	1	1	1									1200	
SR2					1	1	1	1					1000	
SR3									1	1	1	1	800	
DR1	1				1				1					1100
DR2		1				1				1				400
DR3			1				1				1			750
DR4				1				1				1		750
CST	35	30	40	32	37	40	42	25	40	15	20	28		

ويكتب النموذج:

```
MODEL:
SETS:
     SR / SR1 SR2 SR3 / : SPL;
     DR / DR1 DR2 DR3 DR4 / : DM;
     C / C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 /:
                          CST, DV;
     SRXC(SR,C) : SCO;
     DRXC(DR,C) : DCO;
ENDSETS
DATA:
     SPL = 1200 1000 800;
     DM = 1100 400 750 750;
     CST = 35 30 40 32 37 40 42 25 40 15 20 28;
     SCO = 1 1 1 1 0 0 0 0 0 0 0 0
           0 0 0 0 1 1 1 1 0 0 0 0
           0 0 0 0 0 0 0 0 1 1 1 1;
     DCO = 1 0 0 0 1 0 0 0 1 0 0 0
           0 1 0 0 0 1 0 0 0 1 0 0
           0 0 1 0 0 0 1 0 0 0 1 0
           0 0 0 1 0 0 0 1 0 0 0 1;
ENDDATA
MIN = @SUM( C(I): CST(I) *DV(I));
@FOR(SR(I): @SUM( C(J): SCO(I,J) * DV(J)) <= SPL(I));
@FOR(DR(I): @SUM( C(J): DCO(I,J) * DV(J))  = DM(I));
```

والتقرير: يترك للطالب كتمرين.

مثال 5:

حل مشكلة التخصيص التاليه بإستخدام LINGO :

المطلوب تخصيص منطقة فحص واحدة لكل خط تجميع بحيث نقلل ازمنة النقل من الخطوط الى المناطق المعطاة في الجدول التالي:

		Inspection Area				
		A	B	C	D	E
Assembly Line	1	10	4	6	10	12
	2	11	7	7	9	14
	3	13	8	12	14	15
	4	14	16	13	17	17
	5	19	11	17	20	19

والتي تكتب على شكل LP كالتالي

$$
\begin{aligned}
Min\ & 10X_{1A} + 4X_{1B} + 6X_{1C} + 10X_{1D} + 12X_{1E} + \\
& 11X_{2A} + 7X_{2B} + 7X_{2C} + 9X_{2D} + 14X_{2E} + \\
& 13X_{3A} + 8X_{3B} + 12X_{3C} + 14X_{3D} + 15X_{3E} + \\
& 14X_{4A} + 16X_{4B} + 13X_{4C} + 17X_{4D} + 17X_{4E} + \\
& 19X_{5A} + 11X_{5B} + 17X_{5C} + 20X_{5D} + 19X_{5E}
\end{aligned}
$$

$$
\begin{aligned}
St \quad & X_{1A} + X_{1B} + X_{1C} + X_{1D} + X_{1E} = 1 \\
& X_{2A} + X_{2B} + X_{2C} + X_{2D} + X_{2E} = 1 \\
& X_{3A} + X_{3B} + X_{3C} + X_{3D} + X_{3E} = 1 \\
& X_{4A} + X_{4B} + X_{4C} + X_{4D} + X_{4E} = 1 \\
& X_{5A} + X_{5B} + X_{5C} + X_{5D} + X_{5E} = 1 \\
& X_{1A} + X_{2A} + X_{3A} + X_{4A} + X_{5A} = 1 \\
& X_{1B} + X_{2B} + X_{3B} + X_{4B} + X_{5B} = 1 \\
& X_{1C} + X_{2C} + X_{3C} + X_{4C} + X_{5C} = 1 \\
& X_{1D} + X_{2D} + X_{3D} + X_{4D} + X_{5D} = 1 \\
& X_{1E} + X_{2E} + X_{3E} + X_{4E} + X_{5E} = 1 \\
& X_{ij} \geq 0, \quad for\ all\ i\ and\ j
\end{aligned}
$$

ونرتب النموذج في الجدول التالي:

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	SPL	DM
AL1	1	1	1	1	1																					1	
AL2						1	1	1	1	1																1	
AL3											1	1	1	1	1											1	
AL4																1	1	1	1	1						1	
AL5																					1	1	1	1	1	1	
IA1	1					1					1					1					1						1
IA2		1					1					1					1					1					1
IA3			1					1					1					1					1				1
IA4				1					1					1					1					1			1
IA5					1					1					1					1					1		1
CST	10	4	6	10	12	11	7	7	9	14	13	8	12	14	15	14	16	13	17	17	19	11	17	20	19		

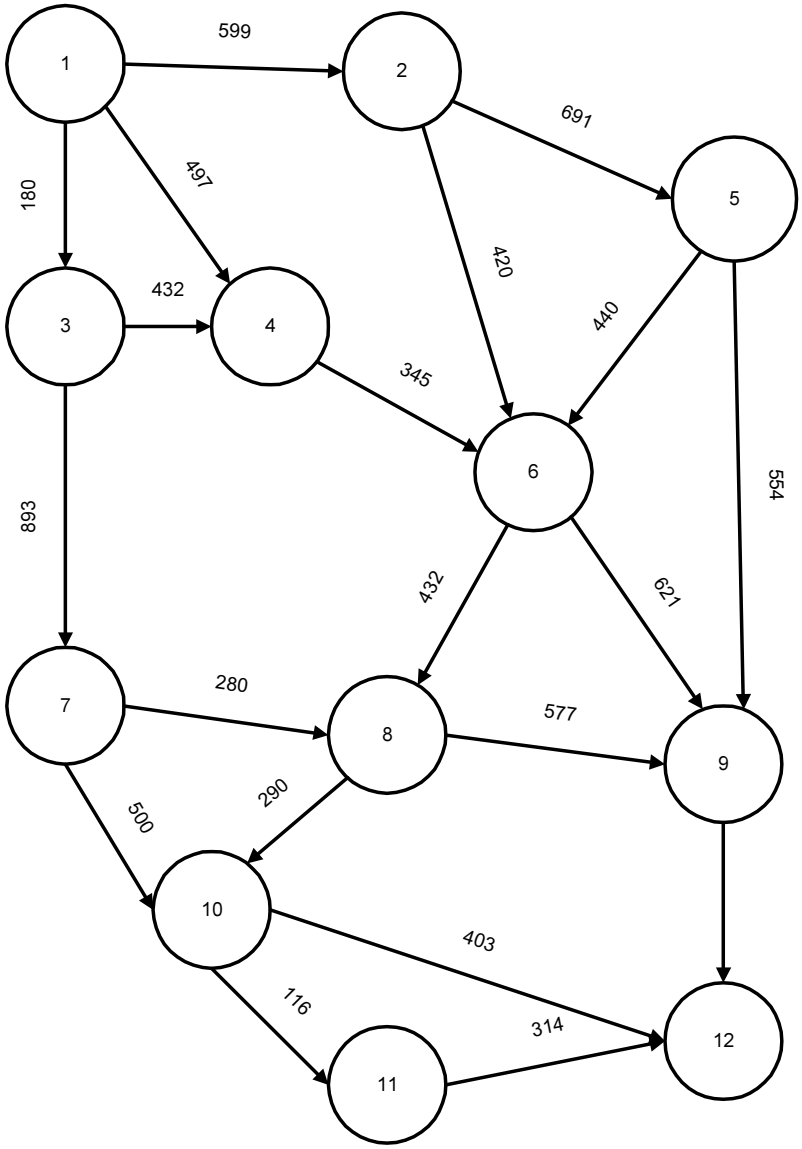
والنموذج:

```
MODEL:
SETS:
     AL / AL1 AL2 AL3 AL4 AL5 / : SPL;
     IA / IA1 IA2 IA3 IA4 IA5 / : DM;
     C / C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12
          C13 C14 C15 C16 C17 C18 C19 C20 C21
          C22 C23 C24 C25 / : CST, DV;
     ALXC(AL, C): SCO;
     IAXC(IA, C): DCO;
ENDSETS
DATA:
SPL = 1 1 1 1 1;
DM = 1 1 1 1 1;
CST = 10 4 6 10 12 11 7 7 9 14 13 8 12 14 15 14 16 13 17
      17 19 11 17 20 19;
SCO = 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1;
DCO = 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0
      0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0
      0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0
      0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0
      0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1;
ENDDATA
MIN = @SUM( C(I): CST(I) *DV(I));
@FOR (AL(I) : @SUM( C(J): SCO(I,J)* DV(J)) = SPL(I));
@FOR (IA(I) : @SUM( C(J): DCO(I,J) * DV(J)) = DM(I));
```

التقرير: يترك للطالب كتمرين.

تمرين: قارن بين نتائج LINGO و LINDO و WinQSB و Excel SOLVER للأمثلة السابقة.

مثال 6:
مشكلة أقصر طريق Shortest Path Problem: الشبكة التالية تمثل خطوط سير ممكنة للسفر من العقدة 1 (عقدة البداية) إلى العقدة 12 (عقدة النهاية) والمراد تعيين أقصر مسافة لخط السير.



لإيجاد مسافة أقصر طريق عبر الشبكة سوف نستخدم التكرار التالي للبرمجة الديناميكية:

$$F(i) = \underset{j}{MIN}\left[D(i,j) + F(j) \right]$$

حيث $F(i)$ أقل مسافة إنتقال من العقدة i إلى العقدة النهائية و $D(i,j)$ هي المسافة من العقدة i إلى العقدة j. أي أن أقل مسافة من العقدة i إلى العقدة النهائية هي تلك الأقل بين كل العقد التي يمكن

الوصول إليها عبر قوس واحد من i لمجموع المسافات من i إلى العقدة المجاورة مضاف إليها أقل مسافة من العقدة المجاورة إلى العقدة النهائية.
ونكتب النموذج كالتالي:

```
MODEL:
SETS:
! F(i) represents the shortest distance from node i to the
last node;
  NODE /1..12/: F;
  ARC(NODE, NODE)/
   1,2   1,3   1,4
   2,5   2,6
   3,4   3,7
   4,6
   5,6   5,9
   6,8   6,9
   7,8   7,10
   8,9   8,10
   9,12
   10,11 10,12
   11,12 /: D;
ENDSETS

! D(I,j) distance between node i and node j;
DATA:
  D =
      599   180    497
      691   420
      432   893
      345
      440   554
      432   621
      280   500
      577   290
      268
      116   403
      314;
ENDDATA
! If you are at node 12, then the distance to node 12 is 0;
 F( @SIZE(NODE)) = 0;
@FOR(NODE(i)| i #LT# @SIZE(NODE):
  F(i) = @MIN(ARC(i,j): D(i,j) + F(j)));
END
```

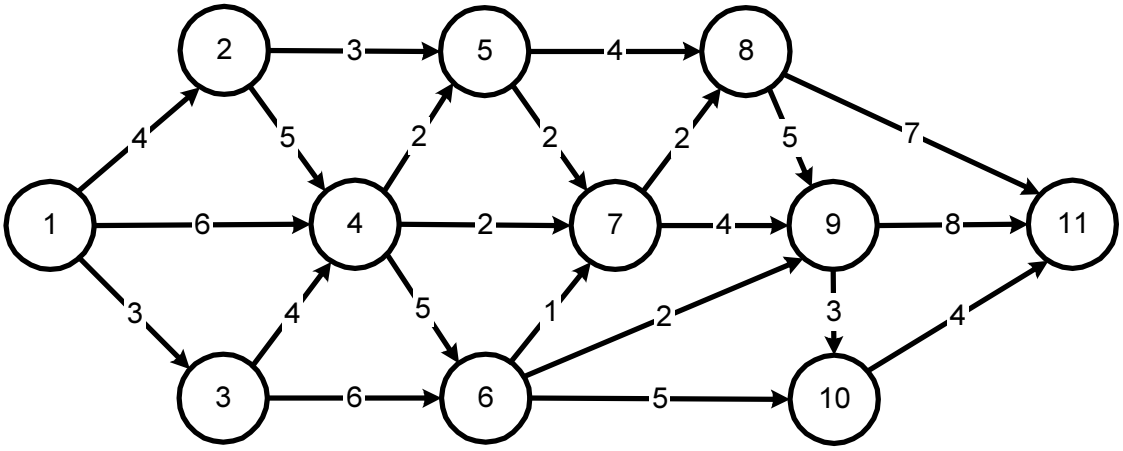
التقرير:

Feasible solution found at iteration: 0

Variable	Value
F(1)	1731.000
F(2)	1309.000
F(3)	1666.000
F(4)	1234.000
F(5)	822.0000
F(6)	889.0000
F(7)	903.0000
F(8)	693.0000
F(9)	268.0000
F(10)	403.0000
F(11)	314.0000
F(12)	0.000000
D(1, 2)	599.0000
D(1, 3)	180.0000
D(1, 4)	497.0000
D(2, 5)	691.0000
D(2, 6)	420.0000
D(3, 4)	432.0000
D(3, 7)	893.0000
D(4, 6)	345.0000
D(5, 6)	440.0000
D(5, 9)	554.0000
D(6, 8)	432.0000
D(6, 9)	621.0000
D(7, 8)	280.0000
D(7, 10)	500.0000
D(8, 9)	577.0000
D(8, 10)	290.0000
D(9, 12)	268.0000
D(10, 11)	116.0000
D(10, 12)	403.0000
D(11, 12)	314.0000

Row	Slack or Surplus
1	0.000000
2	0.000000
3	0.000000
4	0.000000
5	0.000000
6	0.000000
7	0.000000
8	0.000000
9	0.000000
10	0.000000
11	0.000000
12	0.000000

تمرين 1:
أوجد أقصر طريق من المصدر 1 إلى القرار 11 للشبكة التالية:

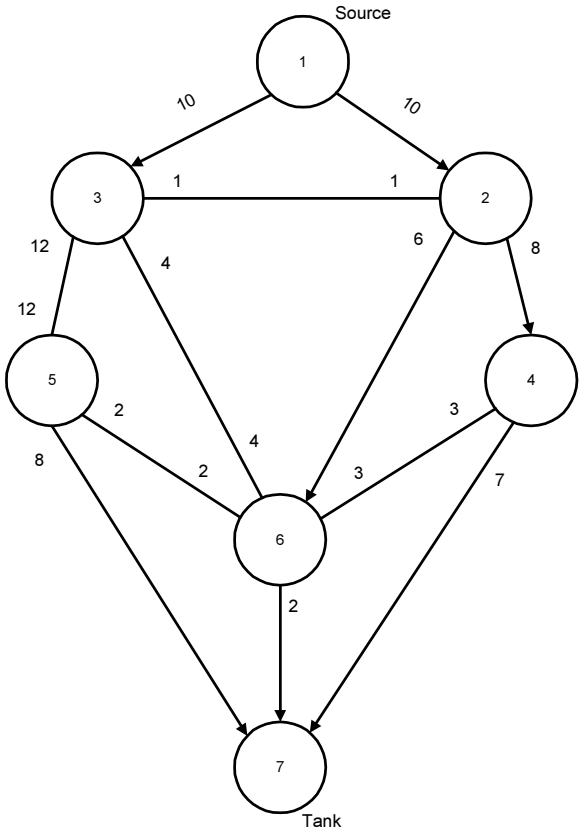


تمرين 2:
حل التمرين السابق بواسطة Excel Solver وقارن النتائج.

مثال 7:

مشكلة التدفق الأعظم Maximal Flow Problem:

الشبكة التالية تمثل تمديدات لأنابيب ماء في شبكة تمد خزان بالمياه



الجدول التالي يعطي سعة الأنابيب (1000 جالون/دقيقة)

			TO				
F		1	2	3	4	5	6	7
R	1		10	10				
O	2			1	8		6	
M	3		1			12	4	
	4						3	7
	5						2	8
	6			4	3	2		2
	7							

المطلوب تحديد مسار أقصى تدفق للخزان.

الحل:

```
! Max flow Problem;
! Given upper limits on the flow on each arc, find the maximum
  flow from node 1 to node 7;
  MODEL:
  SETS:
    NODE : ;
    ARC( NODE, NODE):
         UPPERLIM,  ! The distance matrix;
            X;  ! X( I, J) = 1 if we use ARC I, J;
  ENDSETS
  DATA:
     NODE = 1   2   3   4   5   6   7;! UPPERLIM matrix;
 UPPERLIM = 0   10  10  0   0   0   0 ! 1;
            0   0   1   8   0   6   0 ! 2;
            0   1   0   0   12  4   0 ! 3;
            0   0   0   0   0   3   7 ! 4;
            0   0   0   0   0   2   8 ! 5;
            0   0   4   3   2   0   2 ! 6;
            0   0   0   0   0   0   0;! 7;
  ENDDATA
!-------------------------------------------------------------;
   N = @SIZE( NODE);
  ! Maximize the flow into the last NODE;
  MAX =  @SUM( NODE( I)| I #LT# N : X( I, N));

  ! Upper limit on flow on each arc;
   @FOR( ARC( I, J):
       @BND( 0, X( I, J), UPPERLIM( I, J));
      );

  ! For NODE K, except first and last, ... ;
   @FOR( NODE( K)| K #GT# 1 #AND# K #LT# N:
  ! Inflow = outflow;
      @SUM( NODE( I)| I #NE# K: X( I, K)) =
      @SUM( NODE( J)| J #NE# K: X( K, J));
        );

  END
```

التقرير:

```
  Global optimal solution found.
  Objective value:                              17.00000
  Total solver iterations:                             7


                       Variable           Value        Reduced Cost
                              N        7.000000            0.000000
                  UPPERLIM( 1, 1)       0.000000            0.000000
                  UPPERLIM( 1, 2)       10.00000            0.000000
                  UPPERLIM( 1, 3)       10.00000            0.000000
                  UPPERLIM( 1, 4)       0.000000            0.000000
                  UPPERLIM( 1, 5)       0.000000            0.000000
                  UPPERLIM( 1, 6)       0.000000            0.000000
                  UPPERLIM( 1, 7)       0.000000            0.000000
                  UPPERLIM( 2, 1)       0.000000            0.000000
                  UPPERLIM( 2, 2)       0.000000            0.000000
```

UPPERLIM(2, 3)	1.000000	0.000000
UPPERLIM(2, 4)	8.000000	0.000000
UPPERLIM(2, 5)	0.000000	0.000000
UPPERLIM(2, 6)	6.000000	0.000000
UPPERLIM(2, 7)	0.000000	0.000000
UPPERLIM(3, 1)	0.000000	0.000000
UPPERLIM(3, 2)	1.000000	0.000000
UPPERLIM(3, 3)	0.000000	0.000000
UPPERLIM(3, 4)	0.000000	0.000000
UPPERLIM(3, 5)	12.00000	0.000000
UPPERLIM(3, 6)	4.000000	0.000000
UPPERLIM(3, 7)	0.000000	0.000000
UPPERLIM(4, 1)	0.000000	0.000000
UPPERLIM(4, 2)	0.000000	0.000000
UPPERLIM(4, 3)	0.000000	0.000000
UPPERLIM(4, 4)	0.000000	0.000000
UPPERLIM(4, 5)	0.000000	0.000000
UPPERLIM(4, 6)	3.000000	0.000000
UPPERLIM(4, 7)	7.000000	0.000000
UPPERLIM(5, 1)	0.000000	0.000000
UPPERLIM(5, 2)	0.000000	0.000000
UPPERLIM(5, 3)	0.000000	0.000000
UPPERLIM(5, 4)	0.000000	0.000000
UPPERLIM(5, 5)	0.000000	0.000000
UPPERLIM(5, 6)	2.000000	0.000000
UPPERLIM(5, 7)	8.000000	0.000000
UPPERLIM(6, 1)	0.000000	0.000000
UPPERLIM(6, 2)	0.000000	0.000000
UPPERLIM(6, 3)	4.000000	0.000000
UPPERLIM(6, 4)	3.000000	0.000000
UPPERLIM(6, 5)	2.000000	0.000000
UPPERLIM(6, 6)	0.000000	0.000000
UPPERLIM(6, 7)	2.000000	0.000000
UPPERLIM(7, 1)	0.000000	0.000000
UPPERLIM(7, 2)	0.000000	0.000000
UPPERLIM(7, 3)	0.000000	0.000000
UPPERLIM(7, 4)	0.000000	0.000000
UPPERLIM(7, 5)	0.000000	0.000000
UPPERLIM(7, 6)	0.000000	0.000000
UPPERLIM(7, 7)	0.000000	0.000000
X(1, 1)	0.000000	0.000000
X(1, 2)	7.000000	0.000000
X(1, 3)	10.00000	0.000000
X(1, 4)	0.000000	0.000000
X(1, 5)	0.000000	0.000000
X(1, 6)	0.000000	0.000000
X(1, 7)	0.000000	-1.000000
X(2, 1)	0.000000	0.000000
X(2, 2)	0.000000	0.000000
X(2, 3)	0.000000	0.000000
X(2, 4)	4.000000	0.000000
X(2, 5)	0.000000	0.000000
X(2, 6)	4.000000	0.000000
X(2, 7)	0.000000	-1.000000
X(3, 1)	0.000000	0.000000
X(3, 2)	1.000000	0.000000
X(3, 3)	0.000000	0.000000

X(3, 4)	0.000000	0.000000
X(3, 5)	6.000000	0.000000
X(3, 6)	3.000000	0.000000
X(3, 7)	0.000000	-1.000000
X(4, 1)	0.000000	0.000000
X(4, 2)	0.000000	0.000000
X(4, 3)	0.000000	0.000000
X(4, 4)	0.000000	0.000000
X(4, 5)	0.000000	0.000000
X(4, 6)	0.000000	0.000000
X(4, 7)	7.000000	-1.000000
X(5, 1)	0.000000	0.000000
X(5, 2)	0.000000	0.000000
X(5, 3)	0.000000	0.000000
X(5, 4)	0.000000	0.000000
X(5, 5)	0.000000	0.000000
X(5, 6)	0.000000	0.000000
X(5, 7)	8.000000	-1.000000
X(6, 1)	0.000000	0.000000
X(6, 2)	0.000000	0.000000
X(6, 3)	0.000000	0.000000
X(6, 4)	3.000000	0.000000
X(6, 5)	2.000000	0.000000
X(6, 6)	0.000000	0.000000
X(6, 7)	2.000000	-1.000000
X(7, 1)	0.000000	0.000000
X(7, 2)	0.000000	0.000000
X(7, 3)	0.000000	0.000000
X(7, 4)	0.000000	0.000000
X(7, 5)	0.000000	0.000000
X(7, 6)	0.000000	0.000000
X(7, 7)	0.000000	0.000000

Row	Slack or Surplus	Dual Price
1	0.000000	0.000000
2	17.00000	1.000000
3	0.000000	0.000000
4	0.000000	0.000000
5	0.000000	0.000000
6	0.000000	0.000000
7	0.000000	0.000000

مثال 8:
الإنحدار الخطي البسيط

```
MODEL:
 TITLE Linear Regression with one independent variable;
 !For this model we wish to
  predict Y with the equation: Y(i) = CONS + SLOPE * X(i);
 SETS:
  ! The OBS set contains the data points for X and Y;
  OBS/1..15/:
   Y,
   X;
  OBSN( OBS): XS, YS;
 ENDSETS

 DATA:
  Y =
   4360 4590 4520 4770 4760 5070 5230 5080 5550 5390 5670 5490 5810
6060 5940;
  X =
   1310 1313 1320 1322 1338 1340 1347 1355 1360 1364 1373 1376 1384
1395 1400;
 ENDDATA

 NK = @SIZE( OBS);

 XBAR = @SUM( OBS: X)/ NK;
 YBAR = @SUM( OBS: Y)/ NK;

 @FOR( OBS( I):
   XS( I) = X( I) - XBAR;
   YS( I) = Y( I) - YBAR;
    @FREE( XS(I)); @FREE( YS( I));
     );

 XYBAR = @SUM( OBSN: XS * YS);
 XXBAR = @SUM( OBSN: XS * XS);
 YYBAR = @SUM( OBSN: YS * YS);

  SLOPE = XYBAR/ XXBAR;
   CONS = YBAR - SLOPE * XBAR;
  RESID = @SUM( OBSN: ( YS - SLOPE * XS)^2);

  RSQRU = 1 - RESID/ YYBAR;
  RSQRA =  1 -  ( RESID/ YYBAR) * ( NK - 1)/( NK - 2);
  @FREE( CONS); @FREE( SLOPE);

END
```

النتائج:

Feasible solution found.
Total solver iterations: 0

Model Title: Linear Regression with one independent variable

Variable	Value
NK	15.00000
XBAR	1353.133
YBAR	5219.333
XYBAR	214541.3
XXBAR	11965.73
YYBAR	4068093.
SLOPE	17.92964
CONS	-19041.86
RESID	221443.7
RSQRU	0.9455657
RSQRA	0.9413785
Y(1)	4360.000
Y(2)	4590.000
Y(3)	4520.000
Y(4)	4770.000
Y(5)	4760.000
Y(6)	5070.000
Y(7)	5230.000
Y(8)	5080.000
Y(9)	5550.000
Y(10)	5390.000
Y(11)	5670.000
Y(12)	5490.000
Y(13)	5810.000
Y(14)	6060.000
Y(15)	5940.000
X(1)	1310.000
X(2)	1313.000
X(3)	1320.000
X(4)	1322.000
X(5)	1338.000
X(6)	1340.000
X(7)	1347.000
X(8)	1355.000
X(9)	1360.000
X(10)	1364.000
X(11)	1373.000
X(12)	1376.000
X(13)	1384.000
X(14)	1395.000
X(15)	1400.000
XS(1)	-43.13333
XS(2)	-40.13333
XS(3)	-33.13333
XS(4)	-31.13333
XS(5)	-15.13333
XS(6)	-13.13333
XS(7)	-6.133333
XS(8)	1.866667

XS(9)	6.866667
XS(10)	10.86667
XS(11)	19.86667
XS(12)	22.86667
XS(13)	30.86667
XS(14)	41.86667
XS(15)	46.86667
YS(1)	-859.3333
YS(2)	-629.3333
YS(3)	-699.3333
YS(4)	-449.3333
YS(5)	-459.3333
YS(6)	-149.3333
YS(7)	10.66667
YS(8)	-139.3333
YS(9)	330.6667
YS(10)	170.6667
YS(11)	450.6667
YS(12)	270.6667
YS(13)	590.6667
YS(14)	840.6667
YS(15)	720.6667

Row	Slack or Surplus
1	0.000000
2	0.000000
3	0.000000
4	0.000000
5	0.000000
6	0.000000
7	0.000000
8	0.000000
9	0.000000
10	0.000000
11	0.000000
12	0.000000
13	0.000000
14	0.000000
15	0.000000
16	0.000000
17	0.000000
18	0.000000
19	0.000000
20	0.000000
21	0.000000
22	0.000000
23	0.000000
24	0.000000
25	0.000000
26	0.000000
27	0.000000
28	0.000000
29	0.000000
30	0.000000
31	0.000000
32	0.000000
33	0.000000

34	0.000000
35	0.000000
36	0.000000
37	0.000000
38	0.000000
39	0.000000
40	0.000000
41	0.000000

مثال 9:

مشكلة الجراب Knapsack problem:

كمثال على مشكلة الجراب (حقيبة تحمل على الظهر) وهي مشكلة مهمة في التعبئة العسكرية وتجهيز الطائرات بأنواع مختلفة من الذخيرة والوقود.

لنفرض أنك تريد القيام بنزهة سيرا على الأقدام في أحد المرتفعات الجبلية وقد جهزت قائمة بالأشياء التي تحتاج إليها. كل عنصر من هذه الإشياء له وزن معين والحقيبة لاتستوعب أكثر من 15 كجم. ايضا من ضمن القائمة انت وضعت معيار أهمية لكل عنصر فبدت القائمة كالتالي:

Item	Weight	Rating
Ant Repellent	1	2
Pepsi	3	9
Blanket	4	3
Meat	3	8
Cakes	3	10
Football	1	6
Salad	5	4
Watermelon	10	10
---	---	
Sum	30	

المطلوب تحديد الأشياء المهمة التي تحملها معك مراعيا حجم الحقيبة.

```
MODEL:

SETS:
   ITEMS / ANT_REPEL, PEPSI, BLANKET,
    MEAT, CAKES, FOOTBALL, SALAD,
     WATERMELON/:
      INCLUDE, WEIGHT, RATING;
ENDSETS

DATA:
   WEIGHT RATING =
     1      2
     3      9
     4      3
     3      8
```

```
      3      10
      1       6
      5       4
     10      10;
    KNAPSACK_CAPACITY = 15;
ENDDATA

MAX = @SUM( ITEMS: RATING * INCLUDE);

@SUM( ITEMS: WEIGHT * INCLUDE) <=
 KNAPSACK_CAPACITY;

@FOR( ITEMS: @BIN( INCLUDE));

END
```

او

```
MODEL:

SETS:
   ITEMS :
      INCLUDE, WEIGHT, RATING;
ENDSETS

DATA:
   ITEMS =Bread, Juice, Blanket, Meat,
         Cake, Ball, Salad, Watermelon;
   WEIGHT RATING =
     1      2
     3      9
     4      3
     3      8
     3     10
     1      6
     5      4
    10     10;

   KNAPSACK_CAPACITY = 15;

ENDDATA

MAX = @SUM( ITEMS: RATING * INCLUDE);

@SUM( ITEMS: WEIGHT * INCLUDE) <=
 KNAPSACK_CAPACITY;

@FOR( ITEMS: @BIN( INCLUDE));

END
```

الحل:

```
  Global optimal solution found.
  Objective value:                              38.00000
```

```
Extended solver steps:                                  0
Total solver iterations:                                0


                     Variable           Value        Reduced Cost
            KNAPSACK_CAPACITY        15.00000            0.000000
              INCLUDE( BREAD)        1.000000           -2.000000
              INCLUDE( JUICE)        1.000000           -9.000000
            INCLUDE( BLANKET)        1.000000           -3.000000
               INCLUDE( MEAT)        1.000000           -8.000000
               INCLUDE( CAKE)        1.000000           -10.00000
               INCLUDE( BALL)        1.000000           -6.000000
              INCLUDE( SALAD)        0.000000           -4.000000
         INCLUDE( WATERMELON)        0.000000           -10.00000
               WEIGHT( BREAD)        1.000000            0.000000
               WEIGHT( JUICE)        3.000000            0.000000
            WEIGHT( BLANKET)         4.000000            0.000000
               WEIGHT( MEAT)         3.000000            0.000000
               WEIGHT( CAKE)         3.000000            0.000000
               WEIGHT( BALL)         1.000000            0.000000
              WEIGHT( SALAD)         5.000000            0.000000
         WEIGHT( WATERMELON)         10.00000            0.000000
               RATING( BREAD)        2.000000            0.000000
               RATING( JUICE)        9.000000            0.000000
            RATING( BLANKET)         3.000000            0.000000
               RATING( MEAT)         8.000000            0.000000
               RATING( CAKE)         10.00000            0.000000
               RATING( BALL)         6.000000            0.000000
              RATING( SALAD)         4.000000            0.000000
         RATING( WATERMELON)         10.00000            0.000000

                        Row    Slack or Surplus      Dual Price
                          1         38.00000            1.000000
                          2         0.000000            0.000000
```

الحل بواسطة SageMath

```
from sage.numerical.knapsack import knapsack
knapsack( [(1,2), (3,9), (4,3), (3,8), (3,10), (1,6), (5,4), (10,10)],
max=15)

[38.0, [(1, 2), (3, 9), (4, 3), (3, 8), (3, 10), (1, 6)]]
```

مثال 10:

نموذج سلسلة ماركوف Markov chain model:

نريد إيجاد حل حالة الإستقرار أي $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_n\}$ بحيث $\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$ و $\boldsymbol{\pi}\mathbf{e} = 1$ لمصفوفة الإنتقال:

$$\mathbf{P} = \begin{pmatrix} 0.75 & 0.1 & 0.05 & 0.1 \\ 0.4 & 0.2 & 0.1 & 0.3 \\ 0.1 & 0.2 & 0.4 & 0.3 \\ 0.2 & 0.2 & 0.3 & 0.3 \end{pmatrix}$$

البرنامج:

```
MODEL:
SETS:
STATE/ A B C D/: SPROB;
SXS( STATE, STATE): TPROB;
ENDSETS

DATA:
   TPROB = .75   .1   .05   .1000001
           .4    .2   .1    .3
           .1    .2   .4    .3
           .2    .2   .3    .3;
ENDDATA

  @FOR( STATE( J)| J #LT# @SIZE( STATE):
   SPROB( J) = @SUM( SXS( I, J): SPROB( I) *
    TPROB( I, J))
  );

  @SUM( STATE: SPROB) = 1;

END
```

لنجعل البرنامج يتحقق من أن كل سطر في المصفوفة له مجموع واحد صحيح ويحذر المستخدم في حالة عدم توفر هذا الشرط نضيف التالي قبل نهاية البرنامج:

```
  @FOR( STATE( I):
   @WARN( 'Probabilities in a row must sum to 1.',
    @ABS( 1 - @SUM( SXS( I, K): TPROB( I, K)))
     #GT# .000001);
  );
```

الحل:

```
  Feasible solution found.
  Total solver iterations:                         4
```

Variable	Value
SPROB(A)	0.4750000
SPROB(B)	0.1525000
SPROB(C)	0.1675000
SPROB(D)	0.2050000
TPROB(A, A)	0.7500000
TPROB(A, B)	0.1000000
TPROB(A, C)	0.5000000E-01
TPROB(A, D)	0.1000000
TPROB(B, A)	0.4000000
TPROB(B, B)	0.2000000
TPROB(B, C)	0.1000000
TPROB(B, D)	0.3000000
TPROB(C, A)	0.1000000
TPROB(C, B)	0.2000000
TPROB(C, C)	0.4000000
TPROB(C, D)	0.3000000
TPROB(D, A)	0.2000000
TPROB(D, B)	0.2000000
TPROB(D, C)	0.3000000
TPROB(D, D)	0.3000000

Row	Slack or Surplus
1	0.000000
2	0.000000
3	0.000000
4	0.000000

بإستخدام Sage Math

```
tp = matrix([[ 0.75,0.1,0.05,0.1], [0.4,0.2,0.1,0.3],
[0.1,0.2,0.4,0.3],[0.2,0.2,0.3,0.3]])
pp =tp^1000
pp
############ OR ####################
tp = matrix([[ 0.75,0.1,0.05,0.1], [0.4,0.2,0.1,0.3],
[0.1,0.2,0.4,0.3],[0.2,0.2,0.3,0.3]])
tp
p = vector([0.2, 0.3, 0.4, 0.1])
p
pp = p * tp^1000
pp

(0.475000000000023, 0.152500000000007, 0.167500000000008,
0.205000000000010)
```

```
################### OR ##############################
```

```
tp = matrix(QQ,[[ .75,.1,.05,.1], [.4,.2,.1,.3],
[.1,.2,.4,.3],[.2,.2,.3,.3]])
eigen = tp.eigenvectors_left()
eigen
pi = [k[1][0] for k in eigen if k[0] == 1][0]
pi = [k / sum(pi) for k in pi]
pi
```

```
[19/40, 61/400, 67/400, 41/200]
```

إدخال وإخراج البيانات في LINGO من ملف خارجي:

في التطبيقات العملية يكون حجم البيانات ضخم جدا ويصبح غير عمليا بل مستحيلا إدخال البيانات ضمن البرنامج ولهذا فإن LINGO يستطيع إدخال وإخراج البيانات عبر أوساط كثيرة منها ملفات *.ldt والتي هي ملفات نصية تحوي على جميع البيانات وخاصة بلغة LINGO كما أنة يدخل ويخرج البيانات عبر Excel وأي برنامج لقواعد المعلومات مثل Oracle و Access وغيرها.

سوف نتطرق أولا لإدخال وإخراج البيانات من ملفات *.ldt :

النموذج التالي يقرأ البيانات من الملف WIDGETS2.LDT والموجود في الدليل C:\LINGO7\ بإستخدام الأمر @FILE كالتالي:

```
MODEL:
SETS:
   WAREHOUSES / @FILE( 'WIDGETS2.LDT')/: CAPACITY;
   VENDORS / @FILE( 'WIDGETS2.LDT')/   : DEMAND;
   LINKS( WAREHOUSES, VENDORS): COST, VOLUME;
ENDSETS
   MIN = @SUM( LINKS( I, J):
    COST( I, J) * VOLUME( I, J));
   @FOR( VENDORS( J):
    @SUM( WAREHOUSES( I): VOLUME( I, J)) =
     DEMAND( J));
   @FOR( WAREHOUSES( I):
    @SUM( VENDORS( J): VOLUME( I, J)) <=
     CAPACITY( I));
DATA:
   CAPACITY = @FILE( 'WIDGETS2.LDT');
   DEMAND = @FILE( 'WIDGETS2.LDT');
   COST = @FILE( 'WIDGETS2.LDT');
   @TEXT('WIDGET2OUT.TXT')= VOLUME;
ENDDATA
END
```

الملف WIDGETS2.LDT له الشكل التالي:

```
!List of warehouses;
WH1 WH2 WH3 WH4 WH5 WH6 ~
!List of vendors;
V1 V2 V3 V4 V5 V6 V7 V8 ~
!Warehouse capacities;
60 55 51 43 41 52 ~
!Vendor requirements;
35 37 22 32 41 32 43 38 ~
!Unit shipping costs;
6 2 6 7 4 2 5 9
4 9 5 3 8 5 8 2
5 2 1 9 7 4 3 3
```

```
7  6  7  3  9  2  7  1
2  3  9  5  7  2  6  5
5  5  2  2  8  1  4  3
```

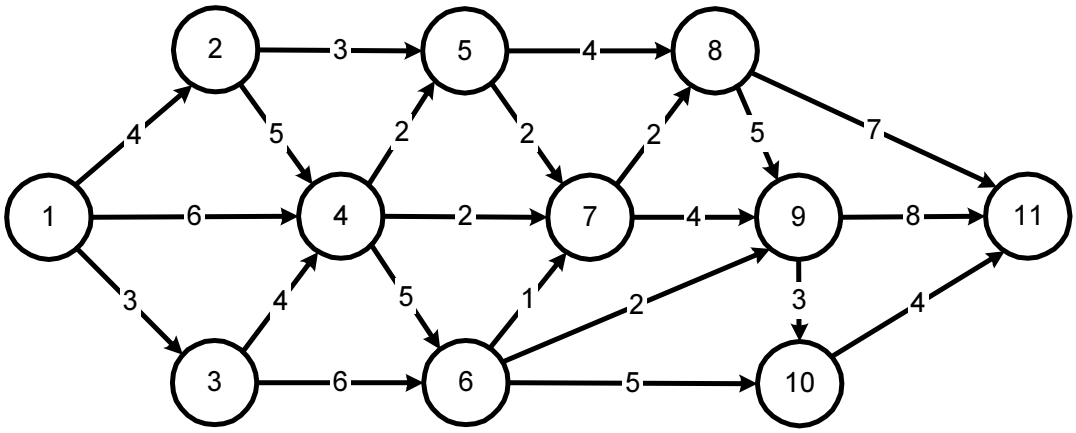
لاحظ (~) والتي تجعل البرنامج يقرأ السطر المناسب ثم يعود إلى الإجراء العادي حتى يقابل أمر @FILE مرة اخرى فيعود لقرائة البيان المناسب إبتدا من الموضع الذي توقف فية سابقا.
لإخراج قيم أي متغير إلى ملف نصي نستخدم الأمر @TEXT ففي المثال السابق اخرجنا قيم متغير القرار VOLUME بالأمر

```
@TEXT('WIDGET2OUT.TXT')= VOLUME;
```

الملف الناتج هو

```
0.0000000     19.000000     0.0000000     0.0000000
41.000000     0.0000000     0.0000000     0.0000000
1.0000000     0.0000000     0.0000000     32.000000
0.0000000     0.0000000     0.0000000     0.0000000
0.0000000     11.000000     0.0000000     0.0000000
0.0000000     0.0000000     40.000000     0.0000000
0.0000000     0.0000000     0.0000000     0.0000000
0.0000000     5.0000000     0.0000000     38.000000
34.000000     7.0000000     0.0000000     0.0000000
0.0000000     0.0000000     0.0000000     0.0000000
0.0000000     0.0000000     22.000000     0.0000000
0.0000000     27.000000     3.0000000     0.0000000
```

تمرين:
أوجد أقصر طريق من المصدر 1 إلى القرار 11 للشبكة التالية:



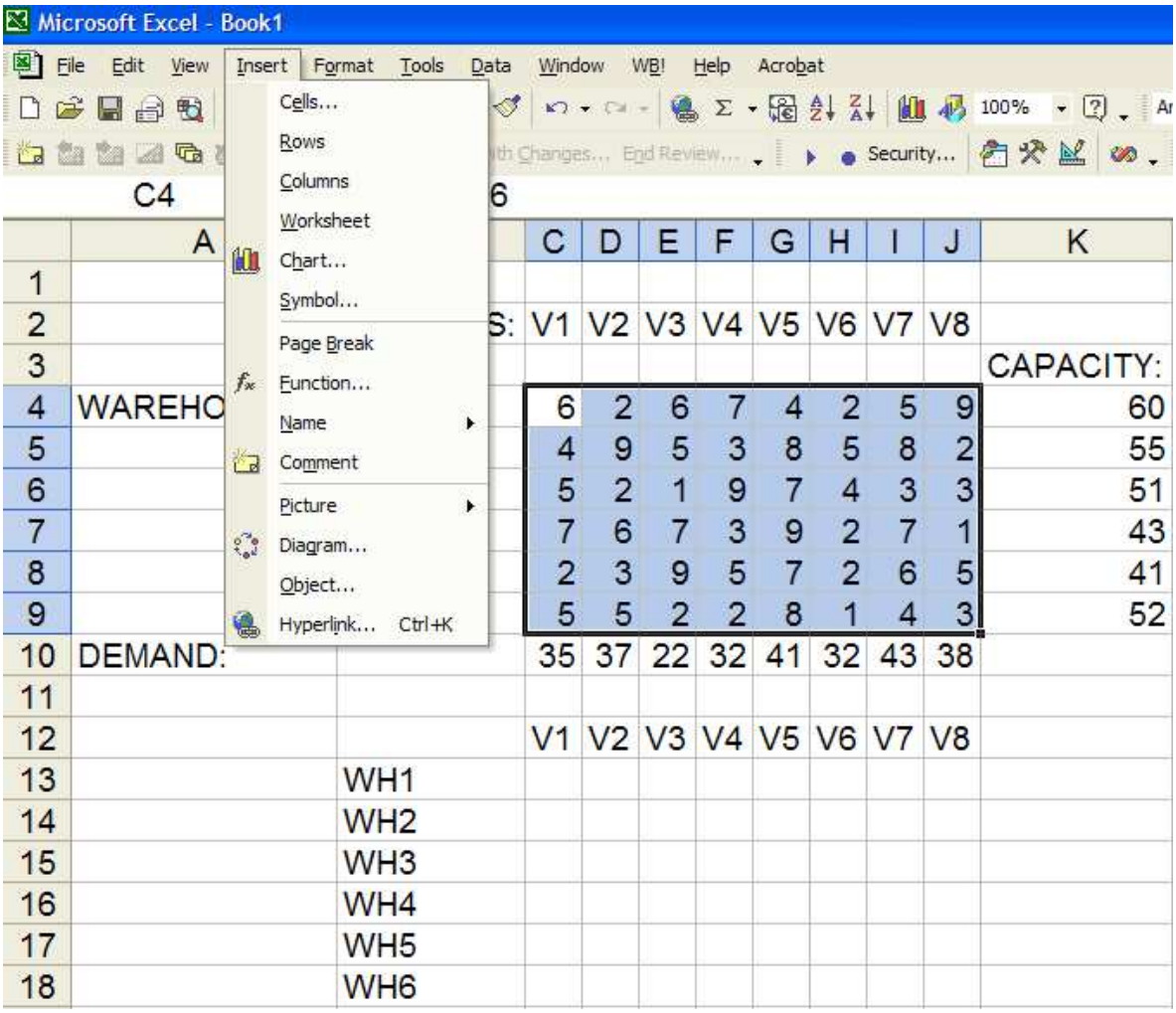
على ان تقرأ البيانات من ملف C:\\DATA\net1.ldt ويتم إخراج النتائج في ملف C:\\DATA\net1out.txt .

إدخال وإخراج البيانات في LINGO بإستخدام Excel:

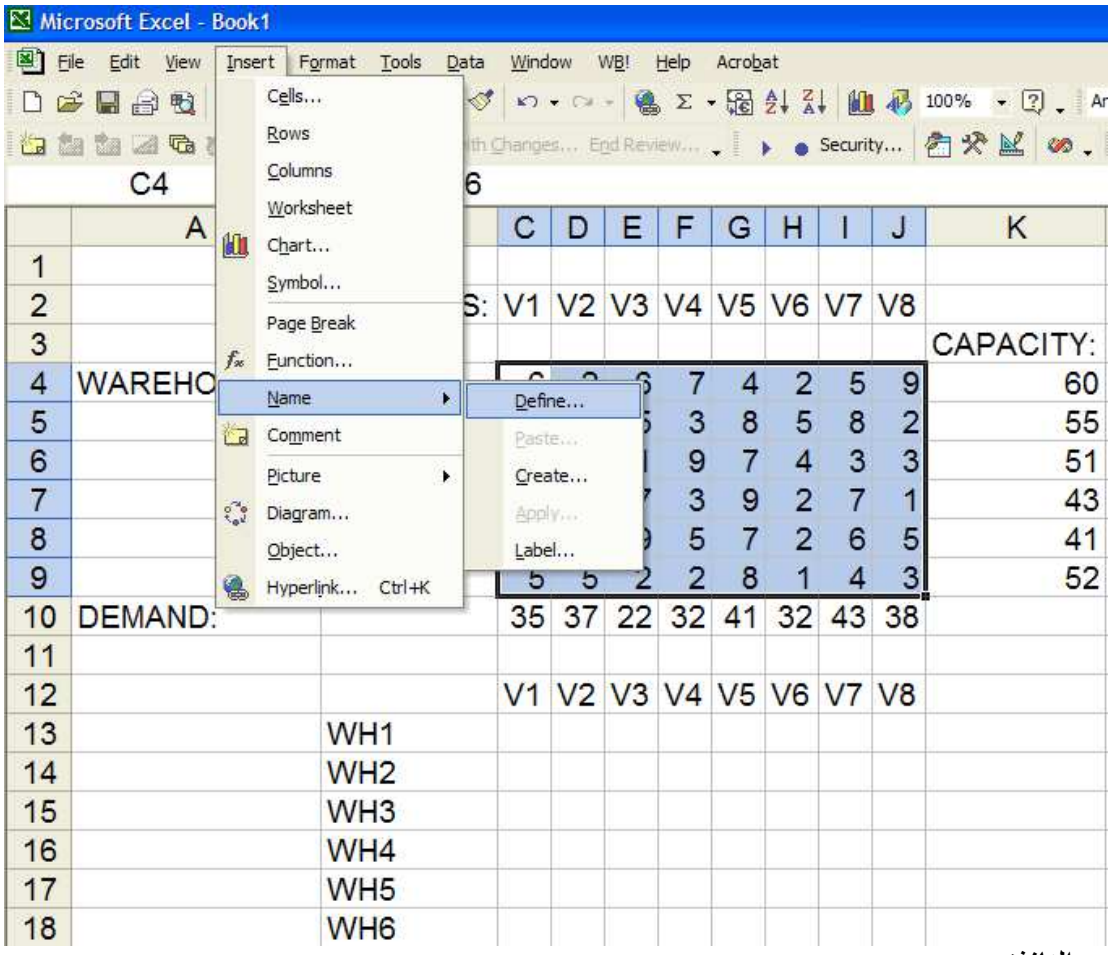
ندخل البيانات في Excel كالتالي:

	A	B	C	D	E	F	G	H	I	J	K
1											
2		VENDORS:	V1	V2	V3	V4	V5	V6	V7	V8	
3											CAPACITY:
4	WAREHOUSES:	WH1	6	2	6	7	4	2	5	9	60
5		WH2	4	9	5	3	8	5	8	2	55
6		WH3	5	2	1	9	7	4	3	3	51
7		WH4	7	6	7	3	9	2	7	1	43
8		WH5	2	3	9	5	7	2	6	5	41
9		WH6	5	5	2	2	8	1	4	3	52
10	DEMAND:		35	37	22	32	41	32	43	38	
11											
12			V1	V2	V3	V4	V5	V6	V7	V8	
13		WH1									
14		WH2									
15		WH3									
16		WH4									
17		WH5									
18		WH6									

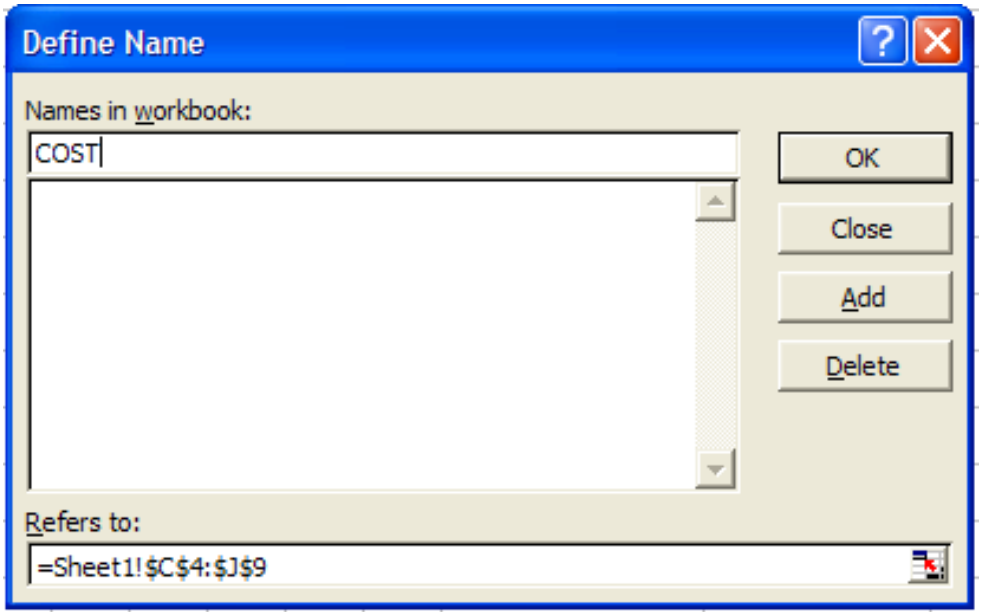
نختار مجال نريد تسميته كالتالي:



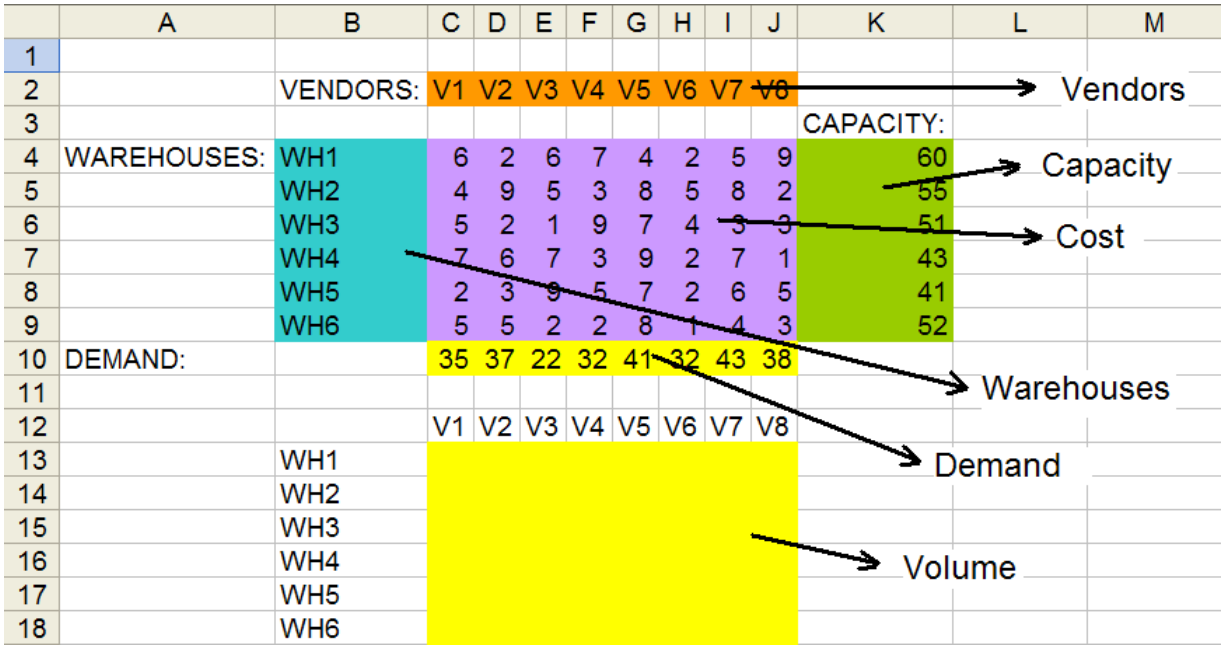
نختار Insert ثم Name ثم Define كما في النافذة التالية



فتظهر النافذة



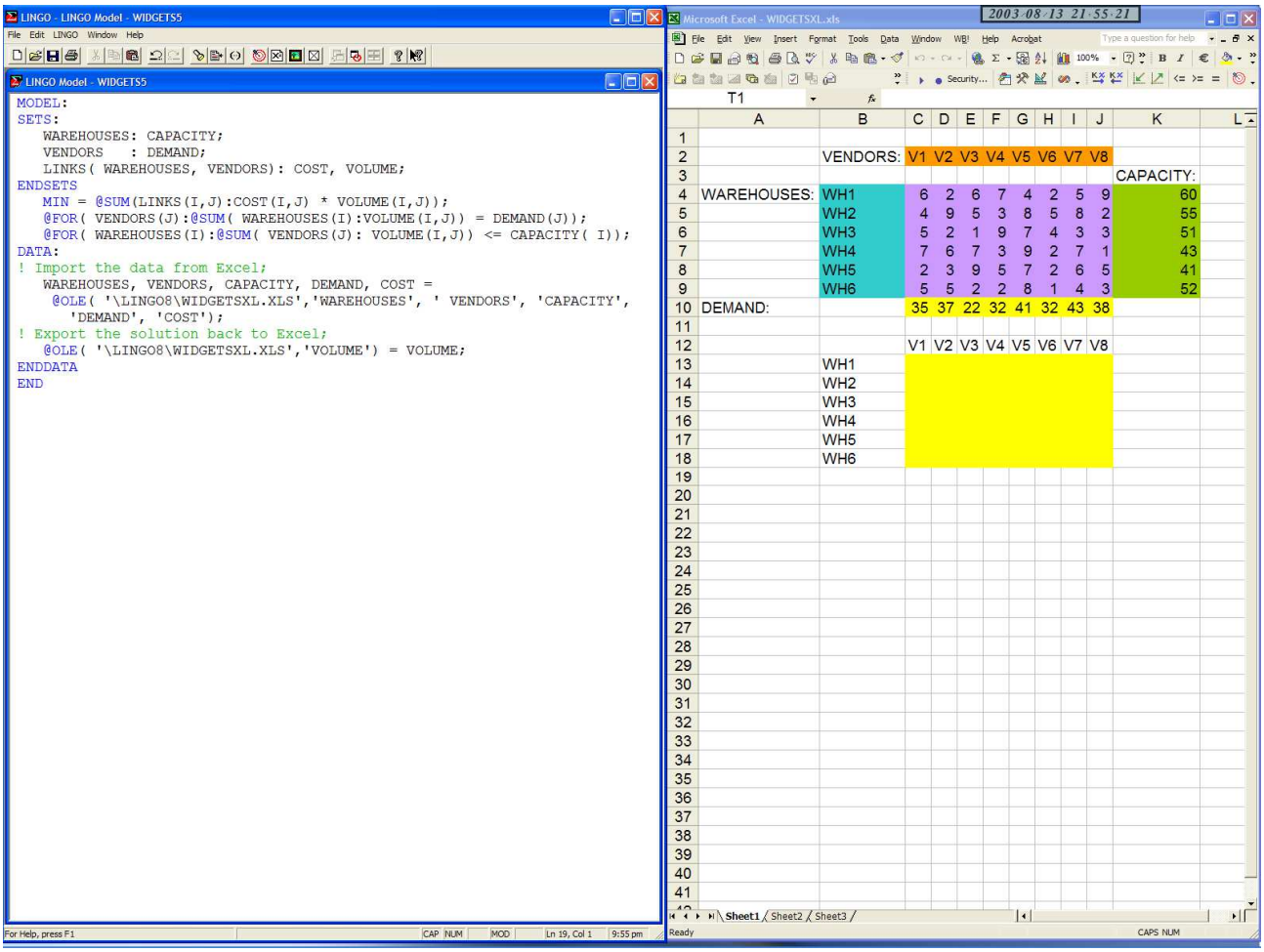
نعطي إسم للبيانات في هذا المجال (لاحظ Sheet1!$C$4:$J$9= التي تظهر ذاتيا).نسمي بقية مجالات بنفس الطريقة كالتالي:



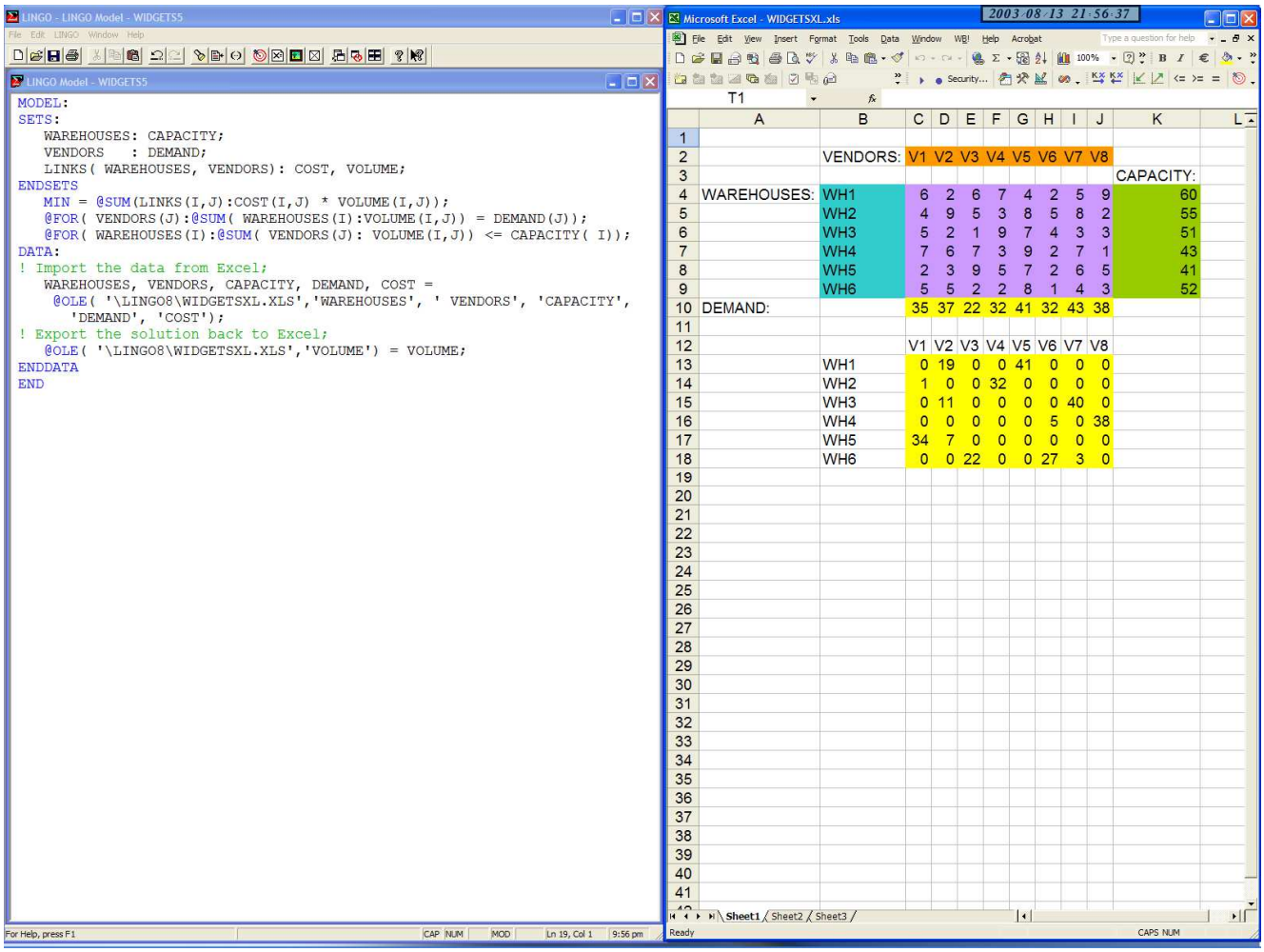
سمي الملف مثلا WIDGETSXL.XLS ثم خزنة في الدليل C:\LINGO7\
البرنامج التالي يقرأ البيانات من WIDGETSXL.XLS ومن ثم يكتب قيم متغير القرار VOLUME في المكان المحدد.

```
MODEL:
SETS:
   WAREHOUSES: CAPACITY;
   VENDORS    : DEMAND;
   LINKS( WAREHOUSES, VENDORS): COST, VOLUME;
ENDSETS
   MIN = @SUM( LINKS(I,J):COST(I,J) * VOLUME(I,J));
   @FOR(VENDORS(J):@SUM(WAREHOUSES(I):VOLUME(I,J)) = DEMAND( J));
   @FOR(WAREHOUSES(I):@SUM(VENDORS(J): VOLUME(I,J))<= CAPACITY(I));
DATA:
! Import the data from Excel;
  WAREHOUSES, VENDORS, CAPACITY, DEMAND, COST =
   @OLE( '\LINGO8\WIDGETSXL.XLS','WAREHOUSES', ' VENDORS',
         'CAPACITY','DEMAND', 'COST');
! Export the solution back to Excel;
   @OLE( '\LINGO8\WIDGETSXL.XLS','VOLUME') = VOLUME;
ENDDATA
END
```

الشكل التالي يبين LINGO و Excel قبل إجراء البرنامج



والشكل التالي يبين LINGO و Excel بعد إجراء البرنامج



مثال 1:

حل النموذج التالي بإستخدام LINGO وأدخل البيانات بإستخدام Excel .

$$Max \quad 8x_1 + 5x_2$$

$$St \quad 2x_1 + x_2 \le 1000$$

$$3x_1 + 4x_2 \le 2400$$

$$x_1 + x_2 \le 700$$

$$x_1 - x_2 \le 350$$

$$x_1, x_2 \ge 0$$

النموذج:

```
MODEL:
SETS:
    R : S;
    C : P, DV;
    RXC(R,C): Coef;
ENDSETS
DATA:
    R, C, S, P, Coef =
@OLE('\LINGO8\OR433EXP01.XLS','RSet','CSet','S','P','Coef');
ENDDATA
MAX = @SUM(C(I): P(I) * DV(I));
 @FOR(R(I): @SUM(C(J): Coef(I,J) * DV(J)) <= S(I));
END
```

أدخل التالي في Excelوسمي الملف or433exp01.xls

	A	B	C	D	E	F	G
1							
2	R=	R1	R2	R3	R4	<= Name this 'Rset'	
3	C=	C1	C2			<= Name this 'Cset'	
4	S=	1000	2400	700	350	<= Name this 'S'	
5	P=	8	5			<= Name this 'P'	
6	Coef=	2	1				
7		3	4			<= Name this 'Coef'	
8		1	1				
9		1	-1				
10							

تمرين: أجري البرنامج السابق وقارن الناتج.

طريقة اخرى لإدخال و إخراج البيانات من إكسل:

تدخل البيانات في إكسل كالسابق وتسمى مجالات البيانات بأسماء المتغيرات المطلوبة في LINGO

	A	B	C	D	E	F	G	H	I
1	PLANTS	P1	P2	P3	P4	P5	P6		
2	CUSTOMERS	C1	C2	C3	C4	C5	C6	C7	C8
3	CAPACITY	60	55	51	43	41	52		
4	DEMAND	35	37	22	32	41	32	43	38
5	COST	6	2	6	7	4	2	5	9
6		4	9	5	3	8	5	8	2
7		5	2	1	9	7	4	3	3
8		7	6	7	3	9	2	7	1
9		2	3	9	5	7	2	6	5
10		5	5	2	2	8	1	4	3
11									
12	VOLUME	0	0	0	0	0	0	0	0
13		0	0	0	0	0	0	0	0
14		0	0	0	0	0	0	0	0
15		0	0	0	0	0	0	0	0
16		0	0	0	0	0	0	0	0
17		0	0	0	0	0	0	0	0

يحفظ الملف بأي إسم ويبقى مفتوحا في إكسل ثم ندخل التالي في LINGO

```
MODEL:
SETS:
PLANTS: CAPACITY;
CUSTOMERS : DEMAND;
LINKS(PLANTS, CUSTOMERS): COST, VOLUME;
ENDSETS
MIN = @SUM(LINKS(I, J):
COST(I, J) * VOLUME(I, J));
@FOR(CUSTOMERS(J):@SUM(PLANTS(I):VOLUME(I, J)) = DEMAND(J));
@FOR(PLANTS(I):@SUM(CUSTOMERS(J): VOLUME(I, J))<= CAPACITY(I));
DATA:
PLANTS, CUSTOMERS, CAPACITY, DEMAND, COST =@OLE();
@OLE()= VOLUME;
ENDDATA
END
```

لاحظ كيفية إدخال البيانات

```
PLANTS, CUSTOMERS, CAPACITY, DEMAND, COST =@OLE();
```

وكيفية إخراج النتائج

```
@OLE()= VOLUME;
```

	A	B	C	D	E	F	G	H	I
1	PLANTS	P1	P2	P3	P4	P5	P6		
2	CUSTOMERS	C1	C2	C3	C4	C5	C6	C7	C8
3	CAPACITY	60	55	51	43	41	52		
4	DEMAND	35	37	22	32	41	32	43	38
5	COST	6	2	6	7	4	2	5	9
6		4	9	5	3	8	5	8	2
7		5	2	1	9	7	4	3	3
8		7	6	7	3	9	2	7	1
9		2	3	9	5	7	2	6	5
10		5	5	2	2	8	1	4	3
11									
12	VOLUME	0	19	0	0	41	0	0	0
13		1	0	0	32	0	0	0	0
14		0	11	0	0	0	0	40	0
15		0	0	0	0	0	5	0	38
16		34	7	0	0	0	0	0	0
17		0	0	22	0	0	27	3	0

```
Global optimal solution found.
Objective value:                              664.0000
Infeasibilities:                              0.000000
Total solver iterations:                            15


  Export Summary Report
  ---------------------
  Transfer Method:       OLE BASED
  Workbook:              Book1.xls
  Ranges Specified:                1
      VOLUME
  Ranges Found:                    1
  Range Size Mismatches:           0
  Values Transferred:             48

                     Variable           Value        Reduced Cost
                CAPACITY( P1)        60.00000            0.000000
                CAPACITY( P2)        55.00000            0.000000
                CAPACITY( P3)        51.00000            0.000000
                CAPACITY( P4)        43.00000            0.000000
                CAPACITY( P5)        41.00000            0.000000
                CAPACITY( P6)        52.00000            0.000000
                  DEMAND( C1)        35.00000            0.000000
                  DEMAND( C2)        37.00000            0.000000
                  DEMAND( C3)        22.00000            0.000000
                  DEMAND( C4)        32.00000            0.000000
                  DEMAND( C5)        41.00000            0.000000
                  DEMAND( C6)        32.00000            0.000000
                  DEMAND( C7)        43.00000            0.000000
                  DEMAND( C8)        38.00000            0.000000
                COST( P1, C1)        6.000000            0.000000
                COST( P1, C2)        2.000000            0.000000
                COST( P1, C3)        6.000000            0.000000
```

COST(P1, C4)	7.000000	0.000000
COST(P1, C5)	4.000000	0.000000
COST(P1, C6)	2.000000	0.000000
COST(P1, C7)	5.000000	0.000000
COST(P1, C8)	9.000000	0.000000
COST(P2, C1)	4.000000	0.000000
COST(P2, C2)	9.000000	0.000000
COST(P2, C3)	5.000000	0.000000
COST(P2, C4)	3.000000	0.000000
COST(P2, C5)	8.000000	0.000000
COST(P2, C6)	5.000000	0.000000
COST(P2, C7)	8.000000	0.000000
COST(P2, C8)	2.000000	0.000000
COST(P3, C1)	5.000000	0.000000
COST(P3, C2)	2.000000	0.000000
COST(P3, C3)	1.000000	0.000000
COST(P3, C4)	9.000000	0.000000
COST(P3, C5)	7.000000	0.000000
COST(P3, C6)	4.000000	0.000000
COST(P3, C7)	3.000000	0.000000
COST(P3, C8)	3.000000	0.000000
COST(P4, C1)	7.000000	0.000000
COST(P4, C2)	6.000000	0.000000
COST(P4, C3)	7.000000	0.000000
COST(P4, C4)	3.000000	0.000000
COST(P4, C5)	9.000000	0.000000
COST(P4, C6)	2.000000	0.000000
COST(P4, C7)	7.000000	0.000000
COST(P4, C8)	1.000000	0.000000
COST(P5, C1)	2.000000	0.000000
COST(P5, C2)	3.000000	0.000000
COST(P5, C3)	9.000000	0.000000
COST(P5, C4)	5.000000	0.000000
COST(P5, C5)	7.000000	0.000000
COST(P5, C6)	2.000000	0.000000
COST(P5, C7)	6.000000	0.000000
COST(P5, C8)	5.000000	0.000000
COST(P6, C1)	5.000000	0.000000
COST(P6, C2)	5.000000	0.000000
COST(P6, C3)	2.000000	0.000000
COST(P6, C4)	2.000000	0.000000
COST(P6, C5)	8.000000	0.000000
COST(P6, C6)	1.000000	0.000000
COST(P6, C7)	4.000000	0.000000
COST(P6, C8)	3.000000	0.000000
VOLUME(P1, C1)	0.000000	5.000000
VOLUME(P1, C2)	19.00000	0.000000
VOLUME(P1, C3)	0.000000	5.000000
VOLUME(P1, C4)	0.000000	7.000000
VOLUME(P1, C5)	41.00000	0.000000
VOLUME(P1, C6)	0.000000	2.000000
VOLUME(P1, C7)	0.000000	2.000000
VOLUME(P1, C8)	0.000000	10.00000
VOLUME(P2, C1)	1.000000	0.000000
VOLUME(P2, C2)	0.000000	4.000000
VOLUME(P2, C3)	0.000000	1.000000
VOLUME(P2, C4)	32.00000	0.000000

VOLUME(P2, C5)	0.000000	1.000000
VOLUME(P2, C6)	0.000000	2.000000
VOLUME(P2, C7)	0.000000	2.000000
VOLUME(P2, C8)	0.000000	0.000000
VOLUME(P3, C1)	0.000000	4.000000
VOLUME(P3, C2)	11.00000	0.000000
VOLUME(P3, C3)	0.000000	0.000000
VOLUME(P3, C4)	0.000000	9.000000
VOLUME(P3, C5)	0.000000	3.000000
VOLUME(P3, C6)	0.000000	4.000000
VOLUME(P3, C7)	40.00000	0.000000
VOLUME(P3, C8)	0.000000	4.000000
VOLUME(P4, C1)	0.000000	4.000000
VOLUME(P4, C2)	0.000000	2.000000
VOLUME(P4, C3)	0.000000	4.000000
VOLUME(P4, C4)	0.000000	1.000000
VOLUME(P4, C5)	0.000000	3.000000
VOLUME(P4, C6)	5.000000	0.000000
VOLUME(P4, C7)	0.000000	2.000000
VOLUME(P4, C8)	38.00000	0.000000
VOLUME(P5, C1)	34.00000	0.000000
VOLUME(P5, C2)	7.000000	0.000000
VOLUME(P5, C3)	0.000000	7.000000
VOLUME(P5, C4)	0.000000	4.000000
VOLUME(P5, C5)	0.000000	2.000000
VOLUME(P5, C6)	0.000000	1.000000
VOLUME(P5, C7)	0.000000	2.000000
VOLUME(P5, C8)	0.000000	5.000000
VOLUME(P6, C1)	0.000000	3.000000
VOLUME(P6, C2)	0.000000	2.000000
VOLUME(P6, C3)	22.00000	0.000000
VOLUME(P6, C4)	0.000000	1.000000
VOLUME(P6, C5)	0.000000	3.000000
VOLUME(P6, C6)	27.00000	0.000000
VOLUME(P6, C7)	3.000000	0.000000
VOLUME(P6, C8)	0.000000	3.000000

Row	Slack or Surplus	Dual Price
1	664.0000	-1.000000
2	0.000000	-4.000000
3	0.000000	-5.000000
4	0.000000	-4.000000
5	0.000000	-3.000000
6	0.000000	-7.000000
7	0.000000	-3.000000
8	0.000000	-6.000000
9	0.000000	-2.000000
10	0.000000	3.000000
11	22.00000	0.000000
12	0.000000	3.000000
13	0.000000	1.000000
14	0.000000	2.000000
15	0.000000	2.000000

إدخال و إخراج البيانات من قاعدة معلومات Database:

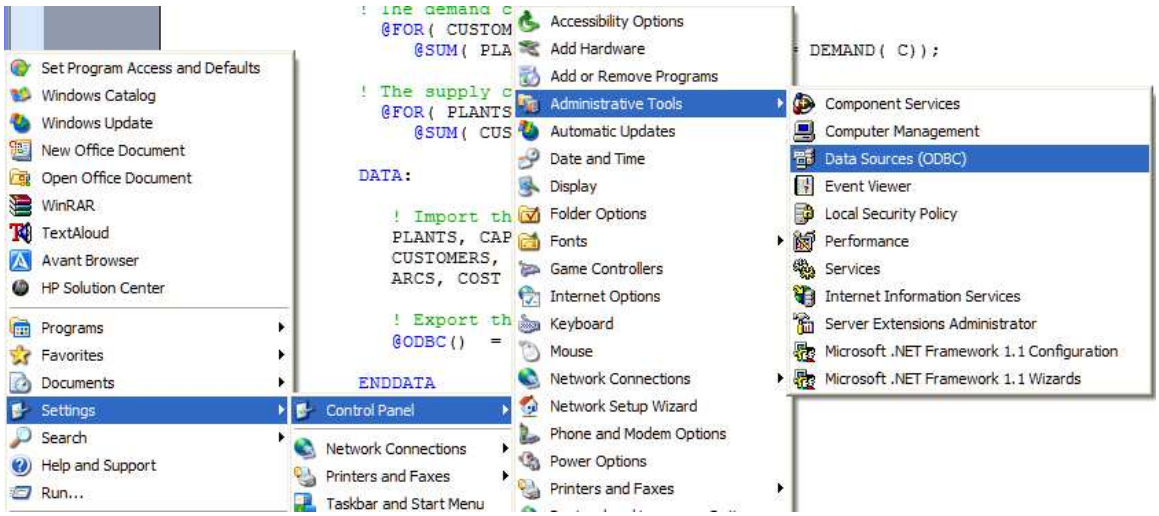
تعتبر صفحات النشر مثل Excel مناسبة لإدخال و إخراج كميات متوسطة من البيانات ولكن عند التعامل مع كميات كبيرة من البيانات فلا بد من إستخدام أنظمة إدارة قواعد المعلومات Database Management Systems (DBMS) وذلك لأن الكثير من جهات الأعمال تبقي بياناتها في قواعد بيانات. في LINGO يوجد إمكانية الربط بأي نظام إدارة قواعد بيانات والتي تمتلك سواقة إمكانية الربط بقاعدة بيانات مفتوحة Open Database Connectivity (ODBC) Driver (وهذا يشمل في الحقيقة جميع DBMS).

الدالة @ODBC في LINGO تقوم بإدخال و إخراج البيانات من أي DBMS وهي متاحة فقط لنظام التشغيل WINDOWS.

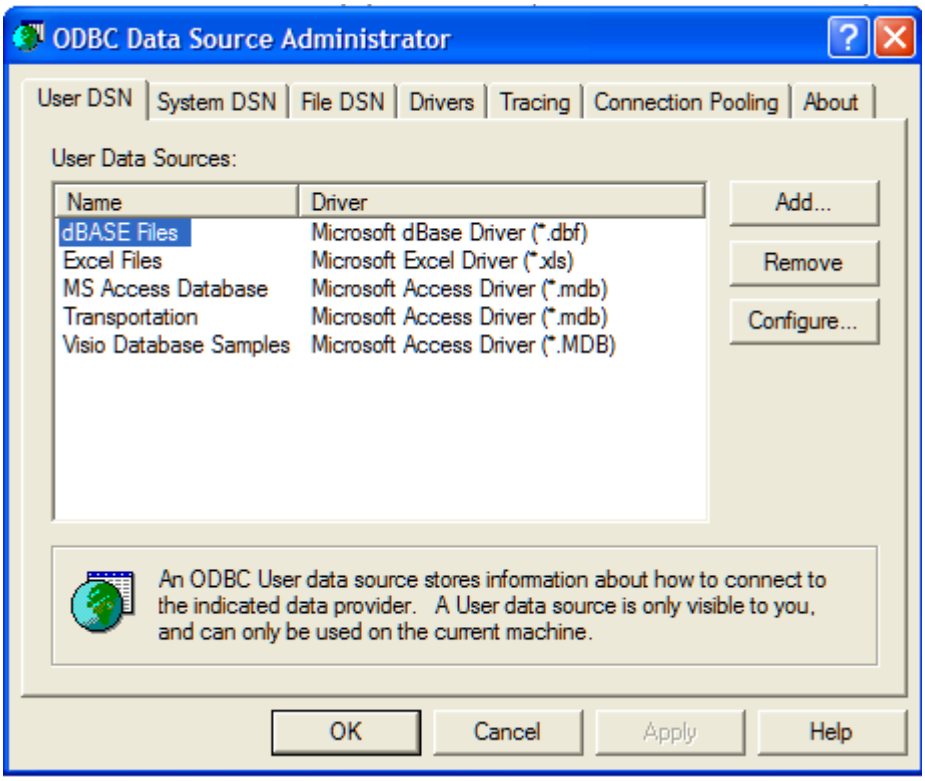
سوف نستخدم مثال موجود في ملف SAMPLES\TRANDB.LG4 لتفصيل ذلك

```
MODEL:
   ! A 3 Plant, 4 Customer Transportation Problem;

   ! Data is retrieved from an ODBC link.  You *MUST*
     use the ODBC Administrator to register one of the
     supplied databases under the name "Transportation"
     in order to get this model to run.  Refer to Chapter
     10 for more details.;

   TITLE Transportation;

   SETS:
      PLANTS: CAPACITY;
      CUSTOMERS: DEMAND;
      ARCS( PLANTS, CUSTOMERS): COST, VOLUME;
   ENDSETS

   ! The objective;
     [OBJ] MIN = @SUM( ARCS: COST * VOLUME);

   ! The demand constraints;
     @FOR( CUSTOMERS( C):
        @SUM( PLANTS( P): VOLUME( P, C)) >= DEMAND( C));

   ! The supply constraints;
     @FOR( PLANTS( P):
        @SUM( CUSTOMERS( C): VOLUME( P, C)) <= CAPACITY( P));

   DATA:

      ! Import the data via ODBC;
      PLANTS, CAPACITY    = @ODBC();
      CUSTOMERS, DEMAND   = @ODBC();
      ARCS, COST          = @ODBC();

      ! Export the solution via ODBC;
      @ODBC()  = VOLUME;

   ENDDATA

END
```

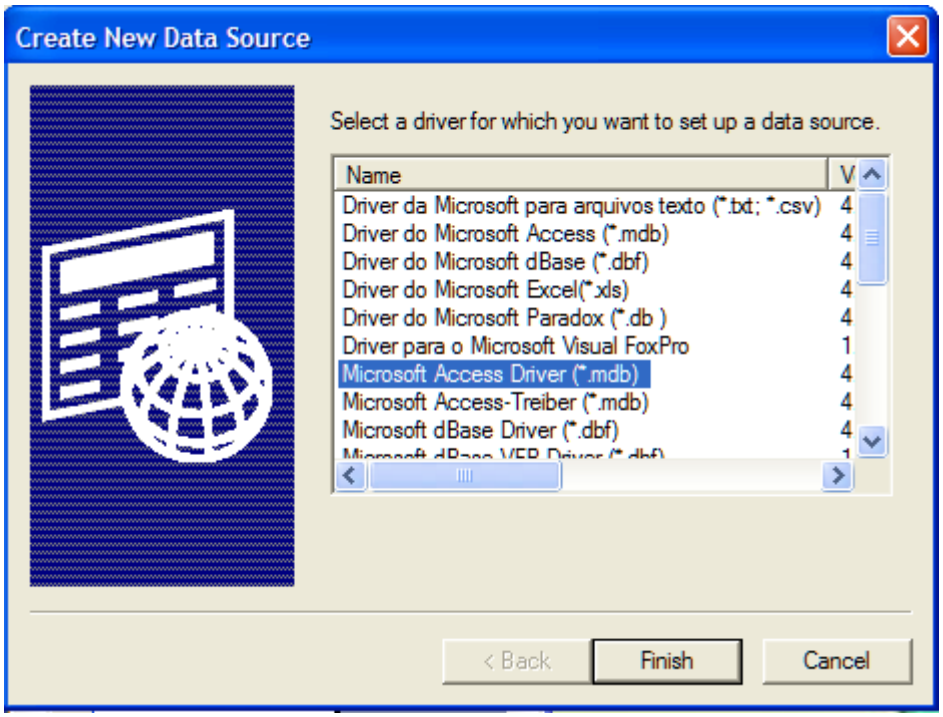
نلاحظ في قسم البيانات لهذا النموذج أننا نستخدم الدالة ODBC@ لكي نقوم بتأسيس توصيل لمصدر بيانات ODBC لكي نسترجع كل البيانات و لكي نرجع المستخرجات.
سوف نشرح كيفية تعريف مصدر بيانات ODBC لكي يستخدمها LINGO.
مصدر بيانات ODBC هو قاعدة معلومات بحيث:
1- موجودة في DBMS يمتلك سواقة ODBC Driver.
2- يكون قد تم تسجيلها مع مدير (ODBCA) ODBC Adminstrator.
مدير رابط قاعدة المعلومات المفتوحة ODBC يمكن الوصول اليه من لوحة التحكم في Windows ويتم تسجيل قاعدة معلومات من برنامج Access أو Oracle كالتالي:
1- من لوحة التحكم Control Panel أفتح Adminstrative Tools ثم Data Sources (ODBC) :



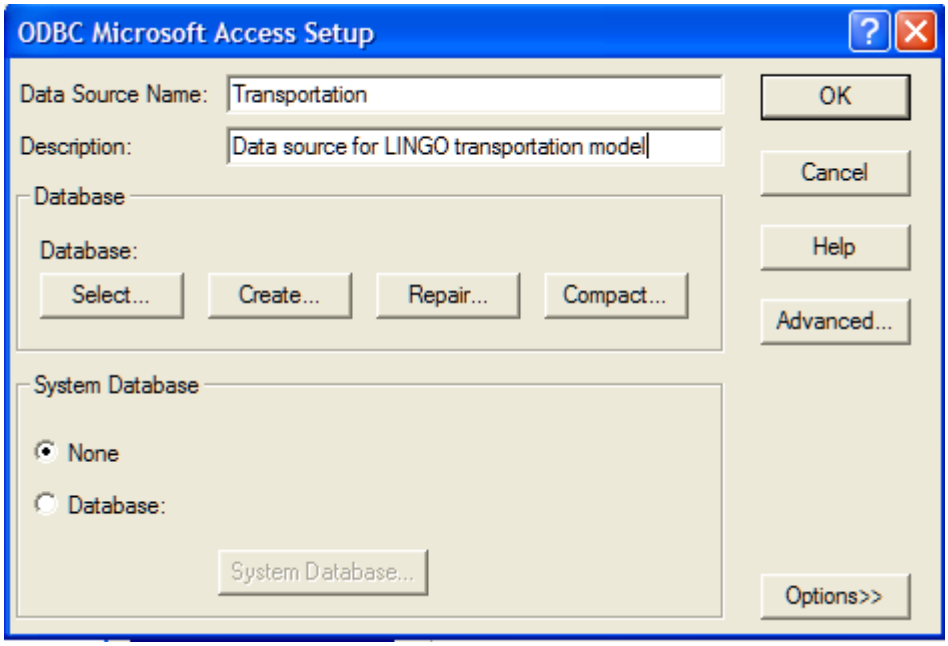
فتظهر النافذة



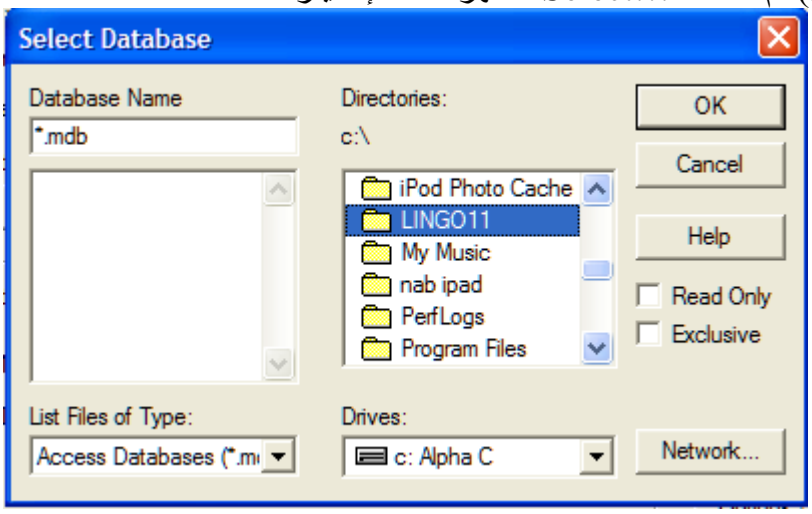
الآن اضغط على ...Add فيظهر



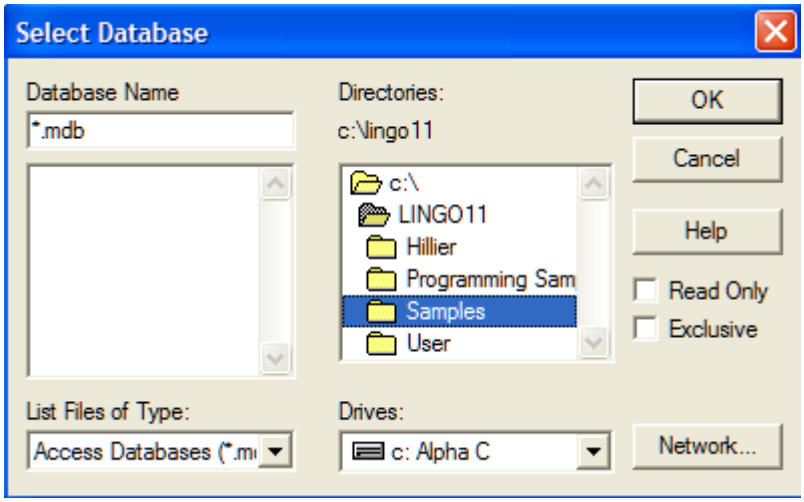
نختار Microsoft Access Driver(*.mdb) ثم نضغط Finish فتظهر النافذة



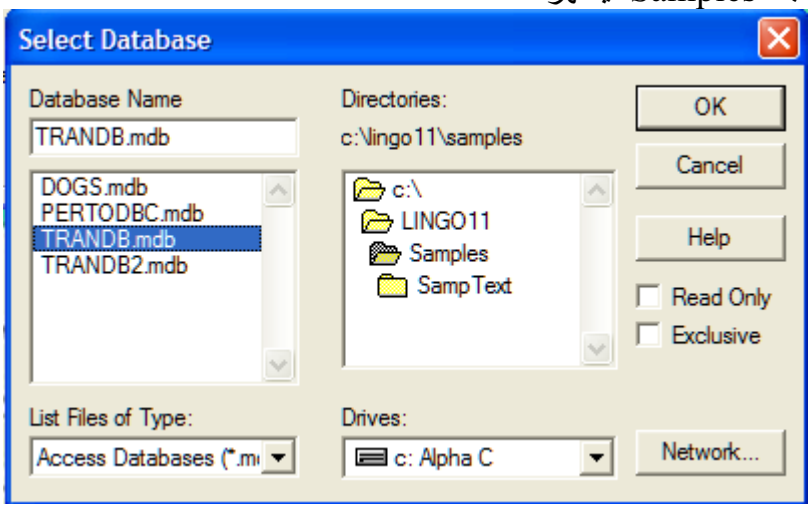
ندخل اسم معرف لمصدر البيانات وليكن Transportation ووصف بسيط كما هو موضح أعلاه (وهذا إختياري) ثم نضغط Select… فتظهر نافذة الإختيار



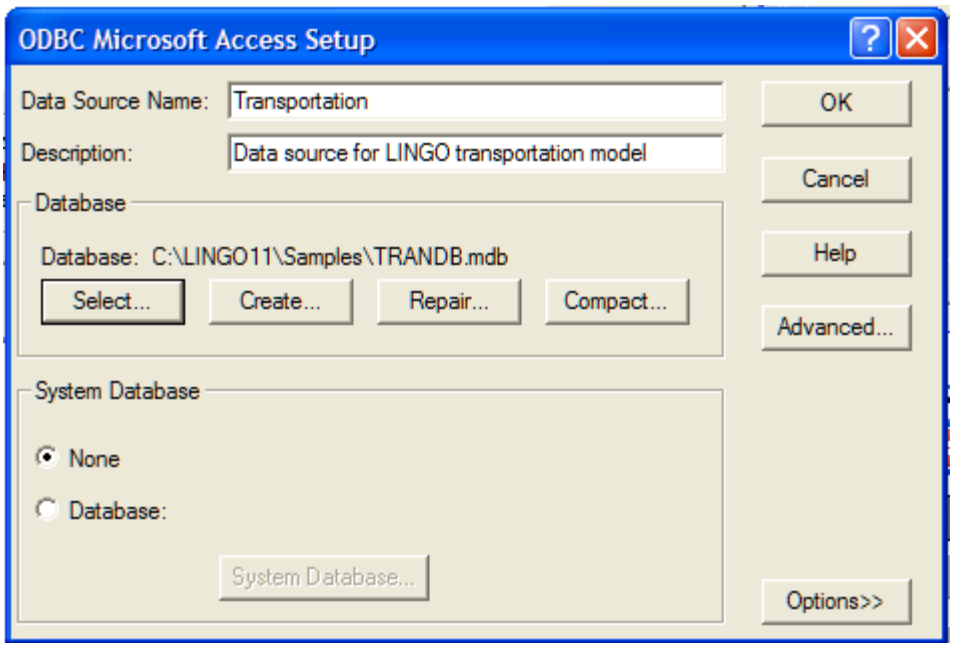
لاحظ أخترنا مجلد LINGO11 (النسخة الحالية) ثم نضغط على الجلد فيفتح



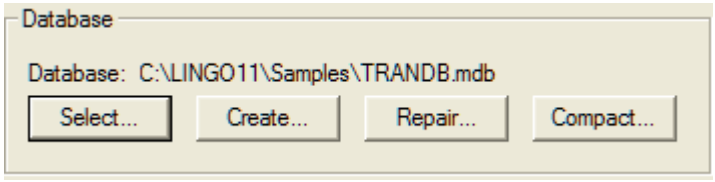
نضغط على المجلد Samples فيظهر



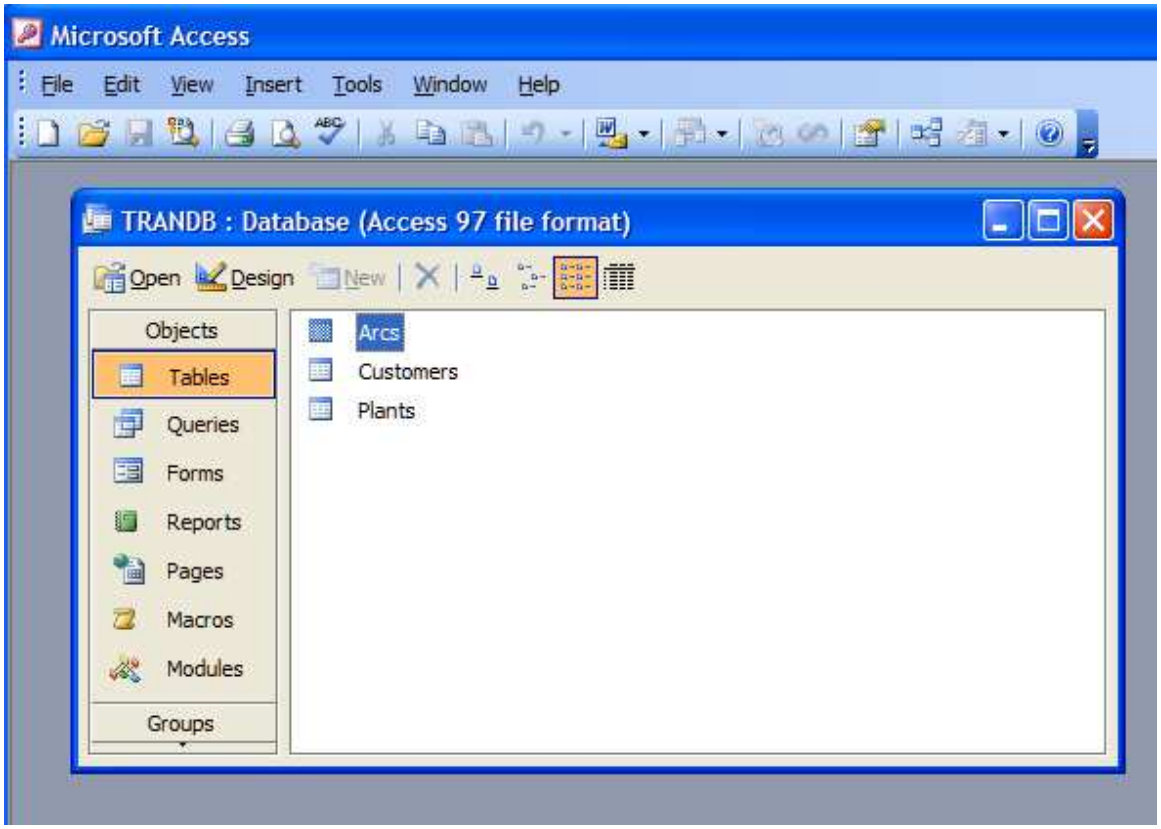
نختار الملف TRANDB.mbd ثم OK



لاحظ



ثم OK و OK
وهكذا تم تسجيل قاعدة البيانات التي سنستخدمها في المثال.
لننظر لقاعدة البيانات في Access



لنفتح Arcs فنرى

Arcs : Table

	Plants	Customers	Cost	Volume
▶	Plant1	Cust1	6	2
	Plant1	Cust2	2	17
	Plant1	Cust3	6	1
	Plant1	Cust4	7	0
	Plant2	Cust1	4	13
	Plant2	Cust2	9	0
	Plant2	Cust3	5	0
	Plant2	Cust4	3	12
	Plant3	Cust1	8	0
	Plant3	Cust2	8	0
	Plant3	Cust3	1	21
	Plant3	Cust4	5	0
*			0	0

و Customers فنرى

Customers : Table

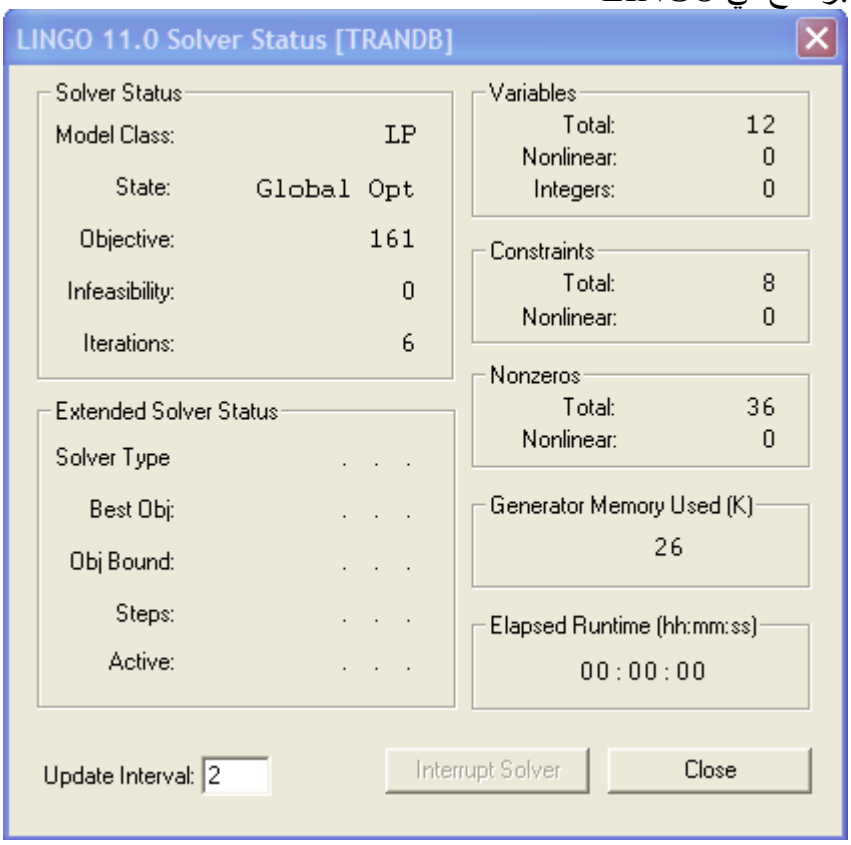
	Customers	Demand
▶	Cust1	15
	Cust2	17
	Cust3	22
	Cust4	12
*		0

وأخيرا Plants

Plants : Table

	Plants	Capacity
▶	Plant1	30
	Plant2	25
	Plant3	21
*		0

الآن نجري البرنامج في LINGO

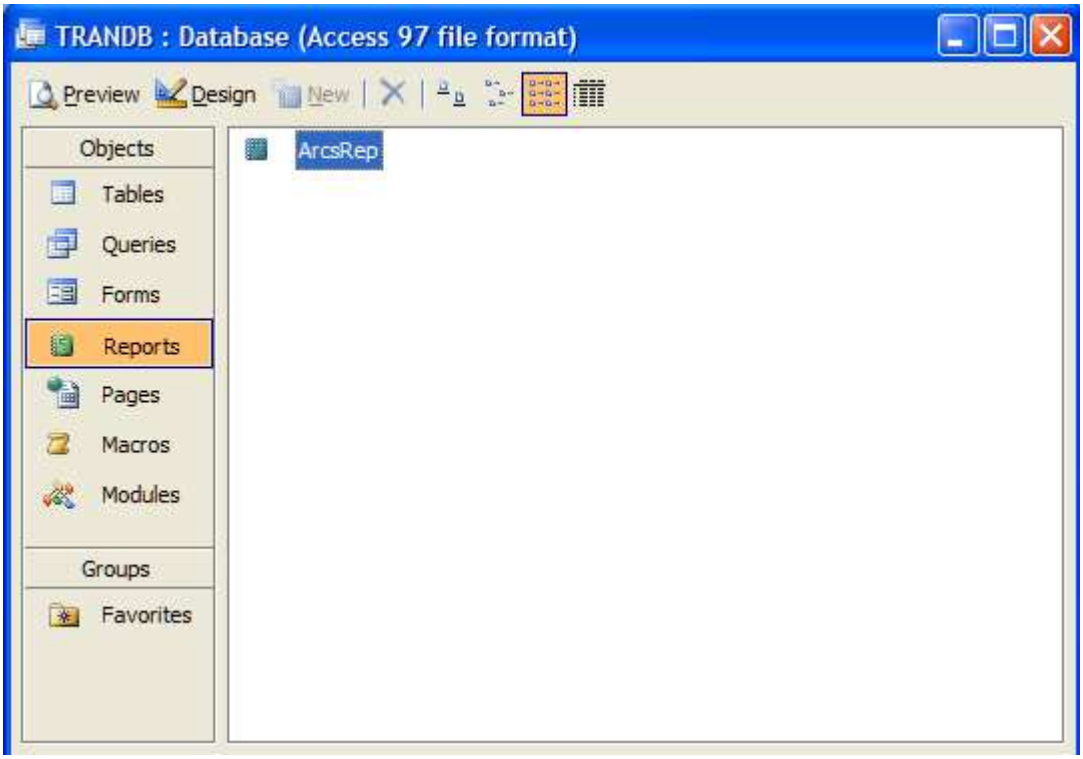


و النتائج:

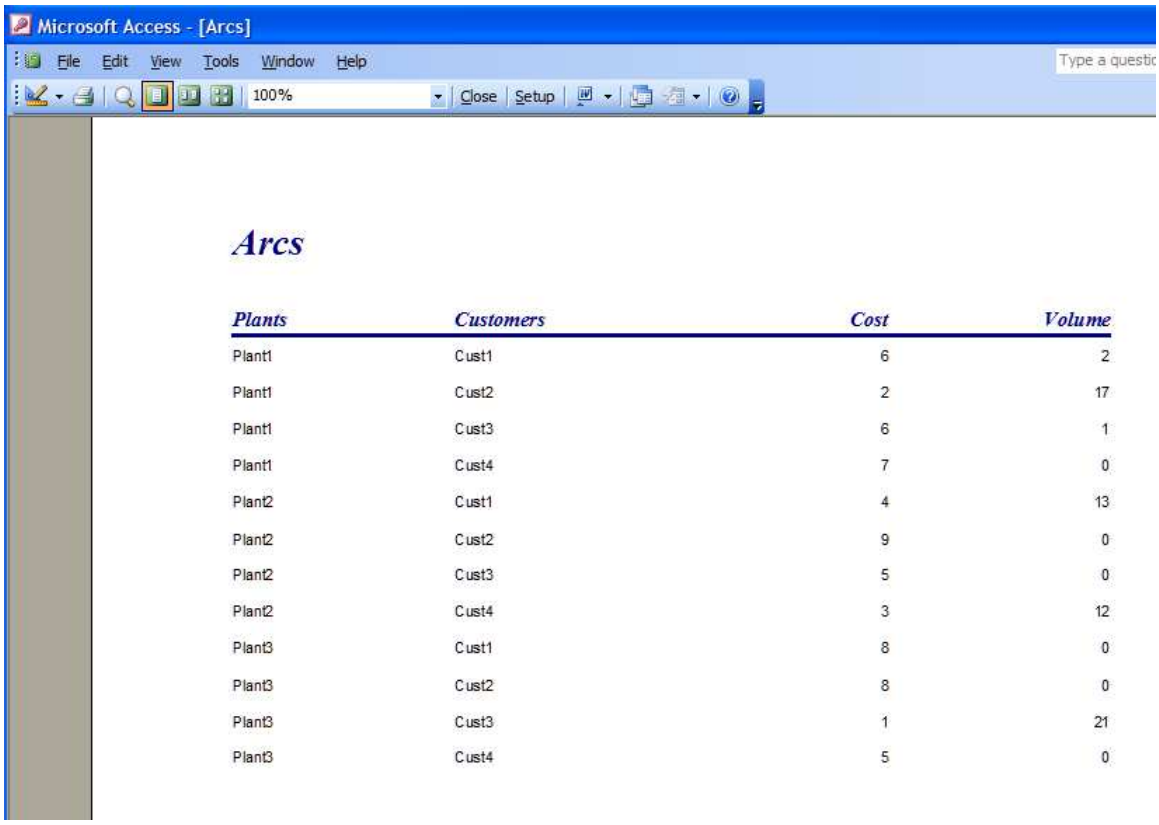
```
Global optimal solution found.
Objective value:                              161.0000
Infeasibilities:                              0.000000
Total solver iterations:                             6


  Export Summary Report
  ---------------------
  Transfer Method:       ODBC BASED
  ODBC Data Source:      Transportation
  Data Table Name:       ARCS
  Columns Specified:              1
      VOLUME
  LINGO Column Length:           12
  Database Column Length:        12


Model Title: Transportation

                     Variable           Value        Reduced Cost
            CAPACITY( PLANT1)        30.00000            0.000000
            CAPACITY( PLANT2)        25.00000            0.000000
            CAPACITY( PLANT3)        21.00000            0.000000
               DEMAND( CUST1)        15.00000            0.000000
               DEMAND( CUST2)        17.00000            0.000000
               DEMAND( CUST3)        22.00000            0.000000
               DEMAND( CUST4)        12.00000            0.000000
          COST( PLANT1, CUST1)        6.000000            0.000000
          COST( PLANT1, CUST2)        2.000000            0.000000
          COST( PLANT1, CUST3)        6.000000            0.000000
          COST( PLANT1, CUST4)        7.000000            0.000000
          COST( PLANT2, CUST1)        4.000000            0.000000
          COST( PLANT2, CUST2)        9.000000            0.000000
          COST( PLANT2, CUST3)        5.000000            0.000000
          COST( PLANT2, CUST4)        3.000000            0.000000
          COST( PLANT3, CUST1)        8.000000            0.000000
          COST( PLANT3, CUST2)        8.000000            0.000000
          COST( PLANT3, CUST3)        1.000000            0.000000
          COST( PLANT3, CUST4)        5.000000            0.000000
        VOLUME( PLANT1, CUST1)        2.000000            0.000000
        VOLUME( PLANT1, CUST2)        17.00000            0.000000
        VOLUME( PLANT1, CUST3)        1.000000            0.000000
        VOLUME( PLANT1, CUST4)        0.000000            2.000000
        VOLUME( PLANT2, CUST1)        13.00000            0.000000
        VOLUME( PLANT2, CUST2)        0.000000            9.000000
        VOLUME( PLANT2, CUST3)        0.000000            1.000000
        VOLUME( PLANT2, CUST4)        12.00000            0.000000
        VOLUME( PLANT3, CUST1)        0.000000            7.000000
        VOLUME( PLANT3, CUST2)        0.000000            11.00000
        VOLUME( PLANT3, CUST3)        21.00000            0.000000
        VOLUME( PLANT3, CUST4)        0.000000            5.000000

                           Row    Slack or Surplus      Dual Price
                           OBJ        161.0000           -1.000000
                             2        0.000000           -6.000000
```

3	0.000000	-2.000000
4	0.000000	-6.000000
5	0.000000	-5.000000
6	10.00000	0.000000
7	0.000000	2.000000
8	0.000000	5.000000

والنتئج في Access

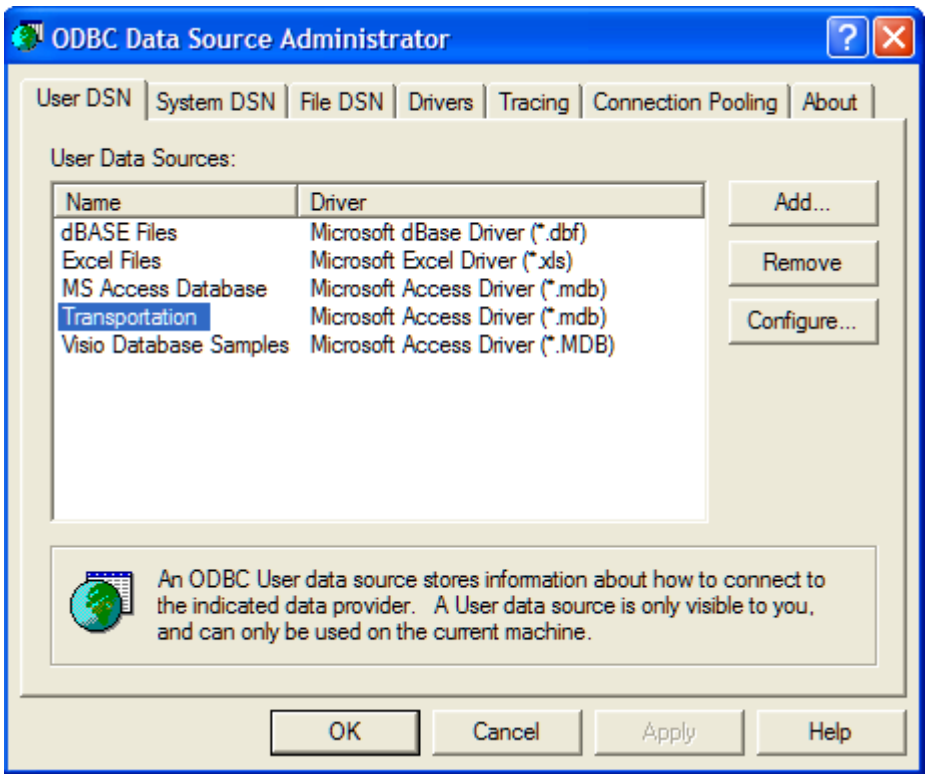


و

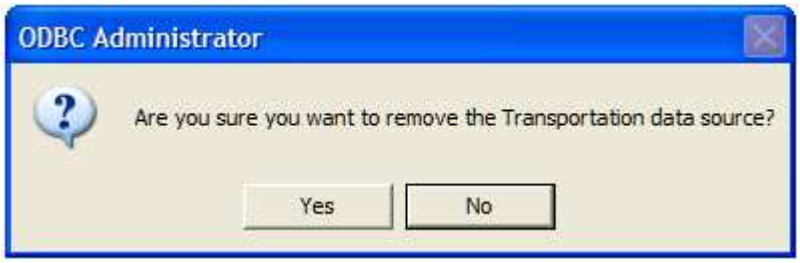


النموذج السابق طبق على 4 زبائن و3 مصانع. سوف نجري النموذج السابق لتطبيقه على 200 زبون و 50 مصنعا بدون أي تغيير للبرنامج بل فقط تسجيل قاعدة بيانات جديدة كالتالي:

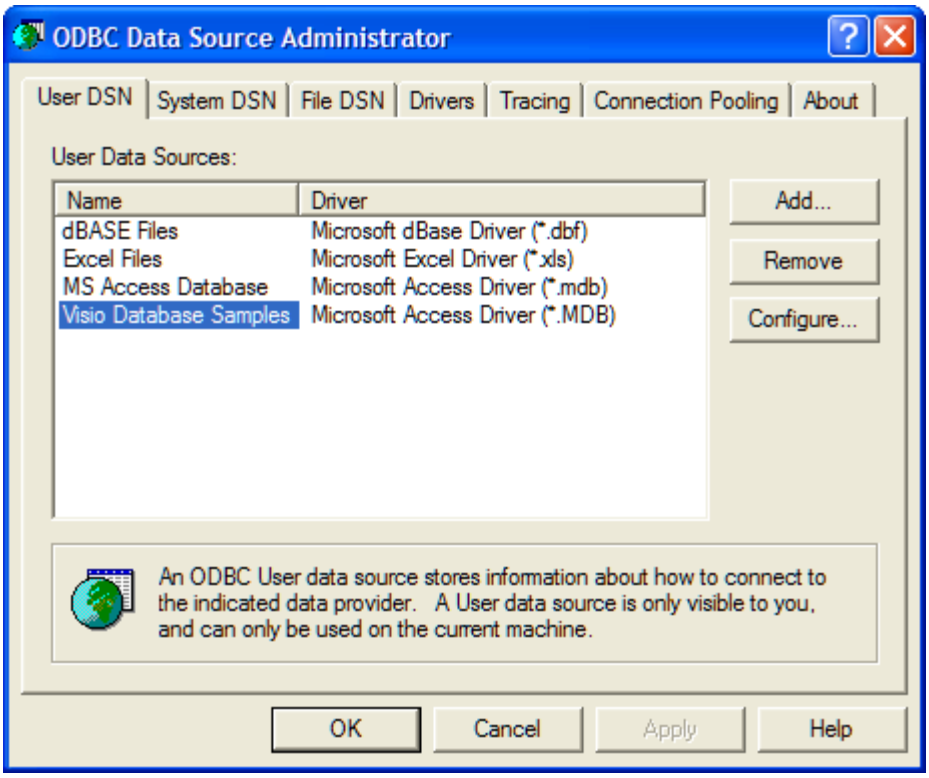
1- نزيل قاعدة البيانات السابقة بإختارها ثم الضغط على Remove



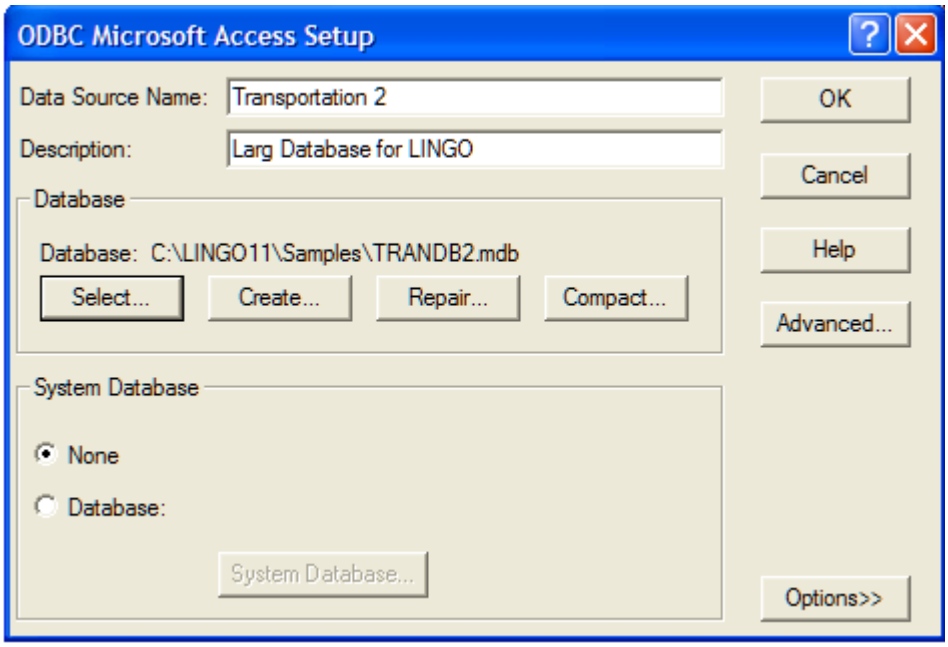
فتظهر النافذة



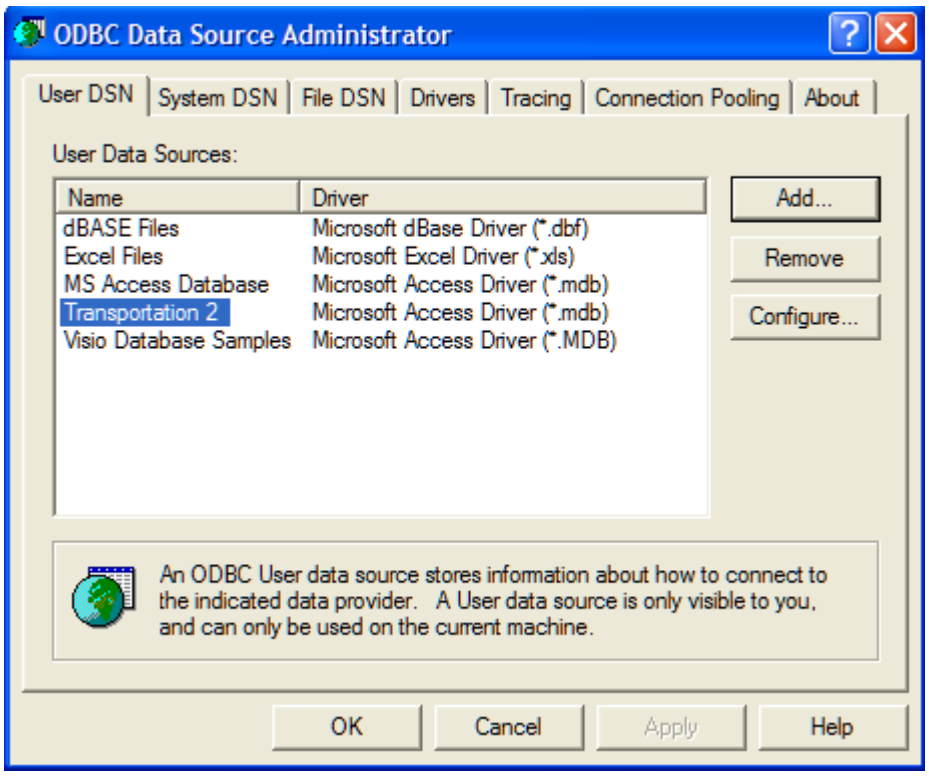
نضغط Yes



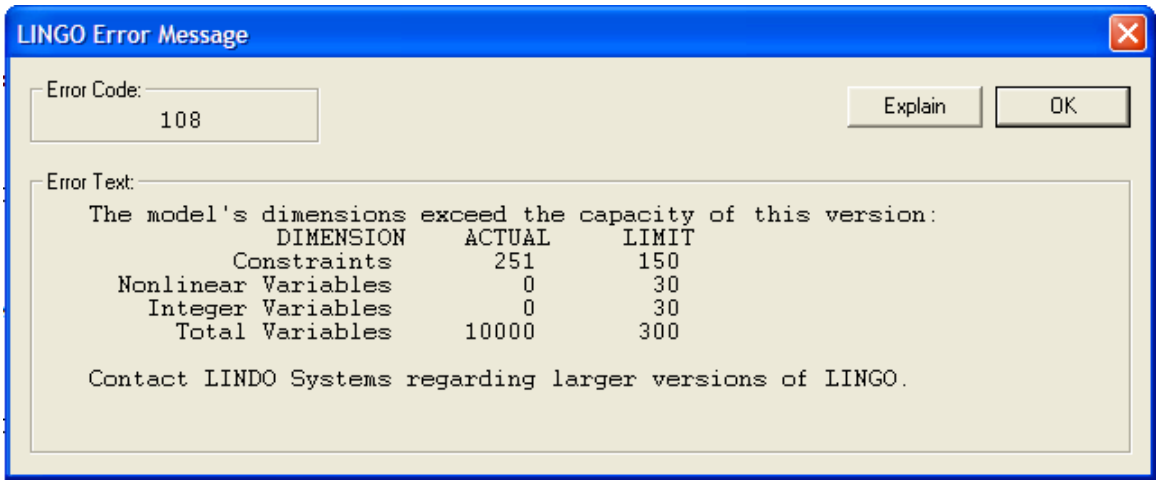
الآن نسجل قاعدة البيانات الجديدة كالسابق



لاحظ الإسم الجديد لقاعدة البيانات

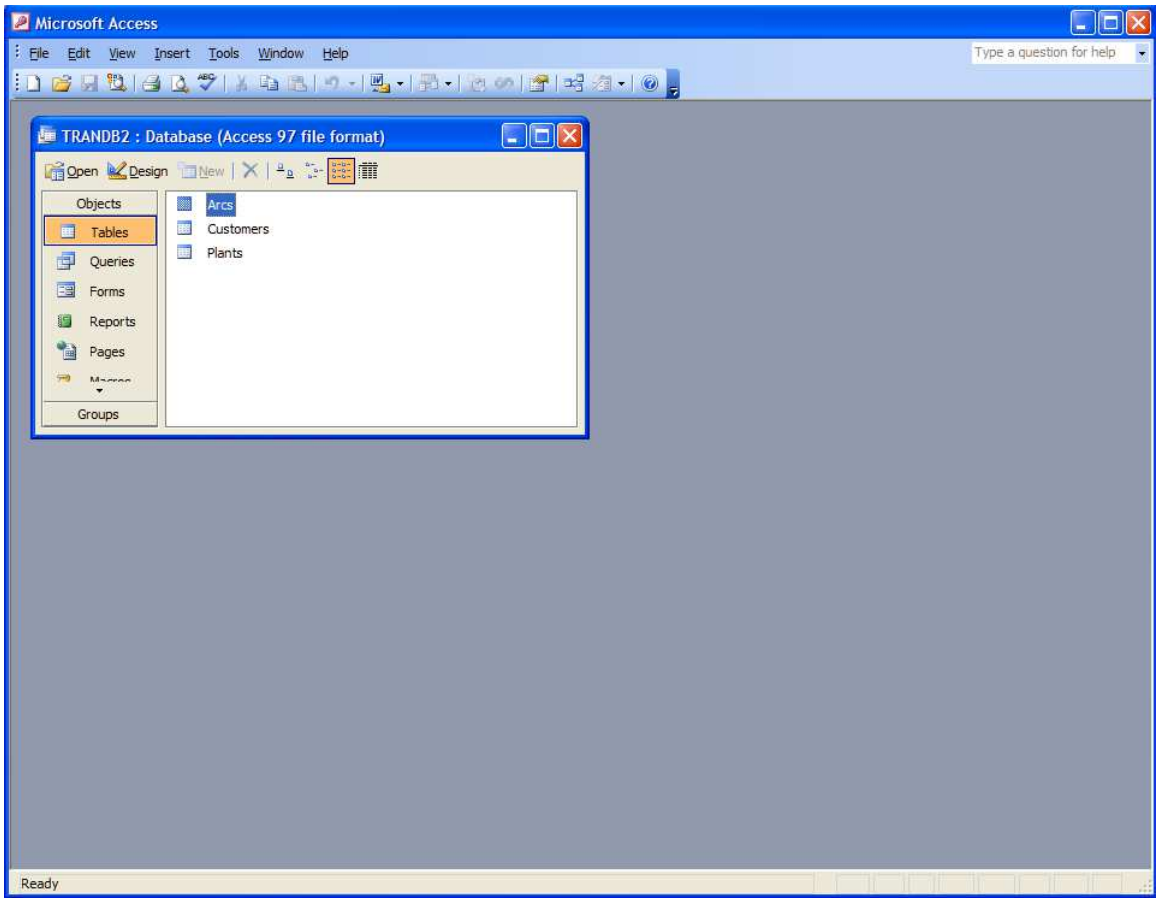


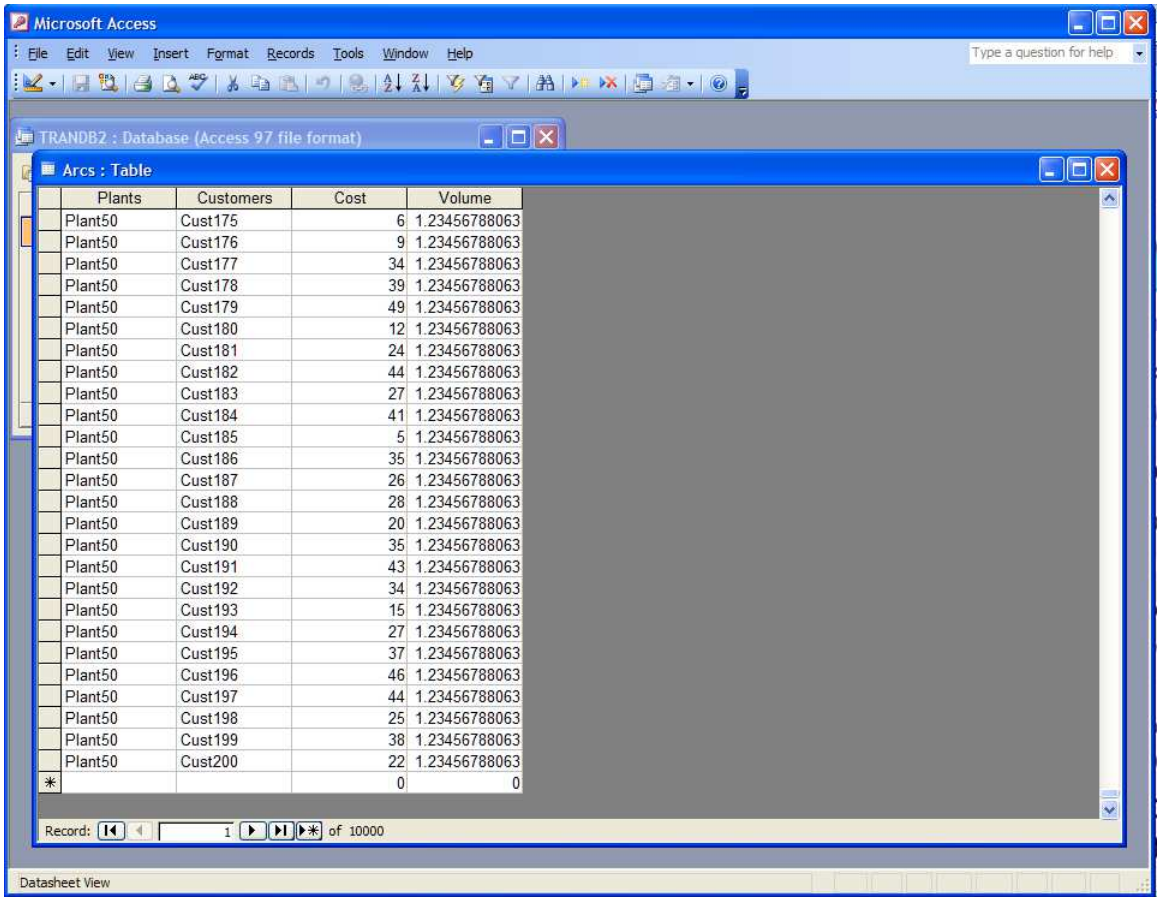
عند إجراء البرنامج نتج



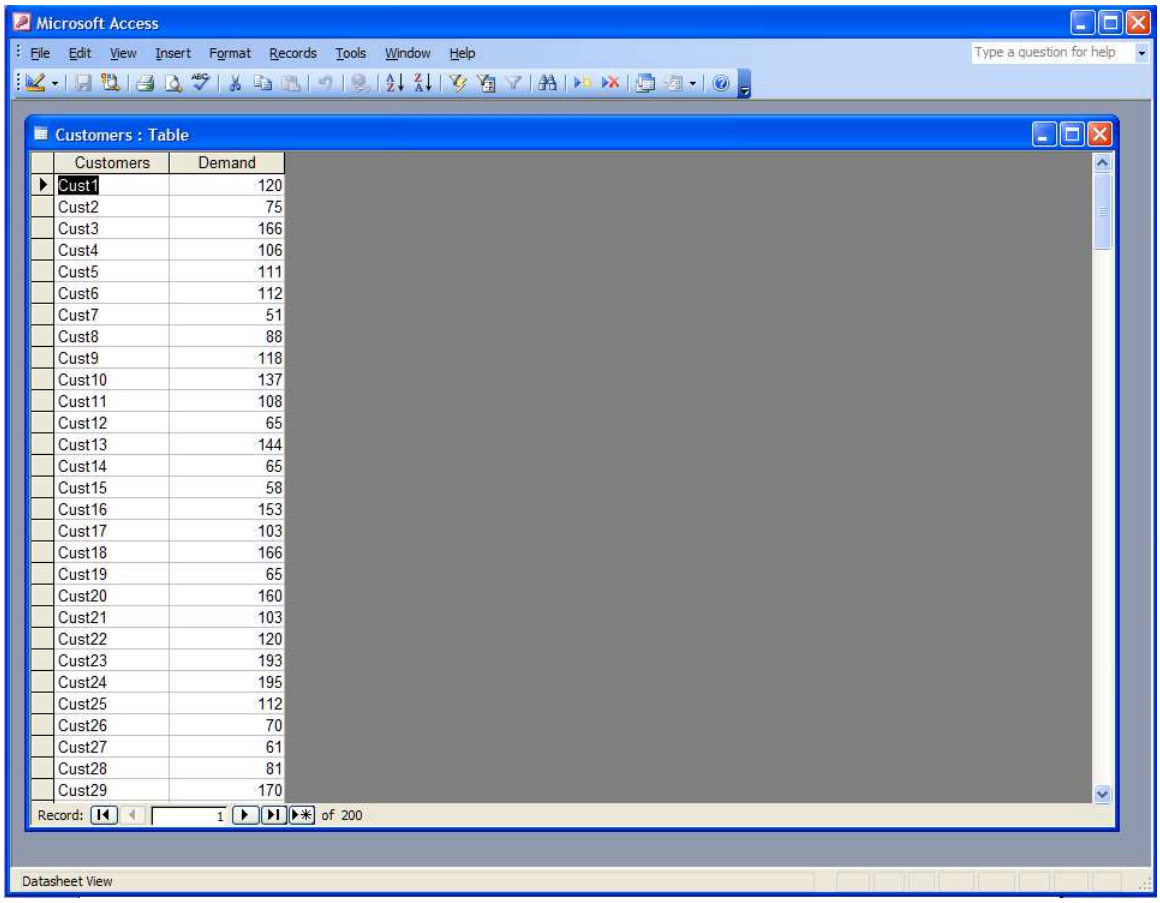
حجم النموذج (عدد القيود والمتغيرات) أكبر من نسخة الطالب المجانية.

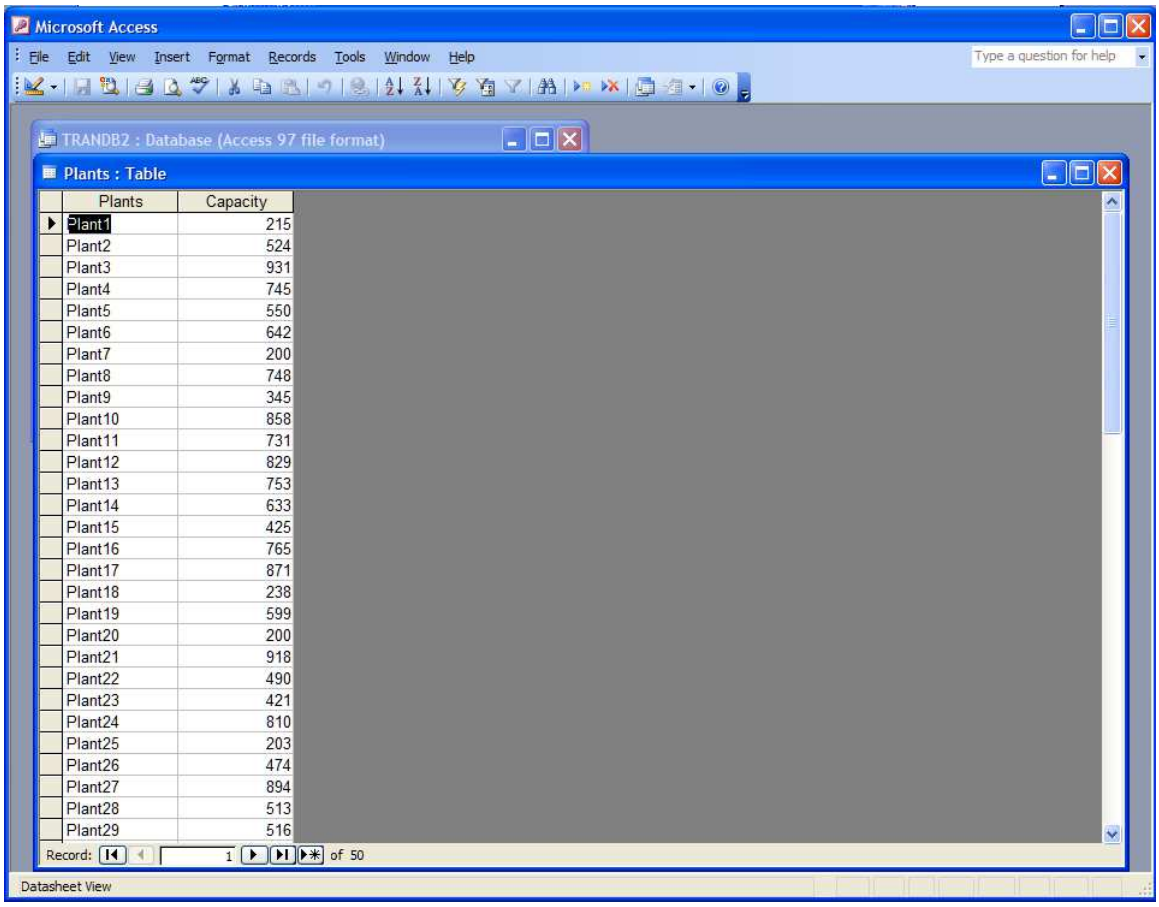
قاعدة البيانات في Access:





Plants	Customers	Cost	Volume
Plant50	Cust175	6	1.23456788063
Plant50	Cust176	9	1.23456788063
Plant50	Cust177	34	1.23456788063
Plant50	Cust178	39	1.23456788063
Plant50	Cust179	49	1.23456788063
Plant50	Cust180	12	1.23456788063
Plant50	Cust181	24	1.23456788063
Plant50	Cust182	44	1.23456788063
Plant50	Cust183	27	1.23456788063
Plant50	Cust184	41	1.23456788063
Plant50	Cust185	5	1.23456788063
Plant50	Cust186	35	1.23456788063
Plant50	Cust187	26	1.23456788063
Plant50	Cust188	28	1.23456788063
Plant50	Cust189	20	1.23456788063
Plant50	Cust190	35	1.23456788063
Plant50	Cust191	43	1.23456788063
Plant50	Cust192	34	1.23456788063
Plant50	Cust193	15	1.23456788063
Plant50	Cust194	27	1.23456788063
Plant50	Cust195	37	1.23456788063
Plant50	Cust196	46	1.23456788063
Plant50	Cust197	44	1.23456788063
Plant50	Cust198	25	1.23456788063
Plant50	Cust199	38	1.23456788063
Plant50	Cust200	22	1.23456788063
		0	0


Microsoft Access
File Edit View Insert Format Records Tools Window Help
Type a question for help
Customers : Table
Customers | Demand
Cust1 120
Cust2 75
Cust3 166
Cust4 106
Cust5 111
Cust6 112
Cust7 51
Cust8 88
Cust9 118
Cust10 137
Cust11 108
Cust12 65
Cust13 144
Cust14 65
Cust15 58
Cust16 153
Cust17 103
Cust18 166
Cust19 65
Cust20 160
Cust21 103
Cust22 120
Cust23 193
Cust24 195
Cust25 112
Cust26 70
Cust27 61
Cust28 81
Cust29 170
Record: 1 of 200
Datasheet View


Microsoft Access
File Edit View Insert Format Records Tools Window Help
Type a question for help
TRANDB2 : Database (Access 97 file format)
Plants : Table
Plants
Capacity
Plant1 215
Plant2 524
Plant3 931
Plant4 745
Plant5 550
Plant6 642
Plant7 200
Plant8 748
Plant9 345
Plant10 858
Plant11 731
Plant12 829
Plant13 753
Plant14 633
Plant15 425
Plant16 765
Plant17 871
Plant18 238
Plant19 599
Plant20 200
Plant21 918
Plant22 490
Plant23 421
Plant24 810
Plant25 203
Plant26 474
Plant27 894
Plant28 513
Plant29 516
Record: 1 of 50
Datasheet View

الفصل الخامس

طرق حسابية بإستخدام R:

ملاحظة: أنظر للملحق صفحة 144 لكي تطلع على مقدمة للغة النمذجة الرياضية R.

1- تعريف دوال

Create a function from a formula

تستخدم الدالة makeFun لتعريف دوال رياضية ولها التركيب التالي:

```
makeFun(object, ...)
```

الحجة object هي شيئ لتوليد الدالة منه. الـ ... حجج إضافية على شكل var = val والتي تتكفل بالقيم الإفتراضية لمدخلات الدلة.

Examples

```
library(mosaic)

f <- makeFun( sin(x^2 * b) ~ x & y & a); f

g <- makeFun( sin(x^2 * b) ~ x & y & a, a=2 ); g

h <- makeFun( a * sin(x^2 * b) ~ b & y, a=2, y=3); h

model <- lm(wage ~ poly(exper,degree=2), data=CPS85)
fit <- makeFun(model)
xyplot(wage ~ exper, data=CPS85)
plotFun(fit(exper) ~ exper, add=TRUE)
model <- glm(wage ~ poly(exper,degree=2), data=CPS85,
family=gaussian)
fit <- makeFun(model)
xyplot(wage ~ exper, data=CPS85)
plotFun(fit(exper) ~ exper, add=TRUE)
```

```
model <- nls( wage ~ A + B * exper + C * exper^2,
data=CPS85, start=list(A=1,B=1,C=1) )
fit <- makeFun(model)
xyplot(wage ~ exper, data=CPS85)
plotFun(fit(exper) ~ exper, add=TRUE)
```

2- المشتقات العددية

Numerical Derivatives

الدالة numD تكون أو تعطي المشتقة العددية لتعبير رياضي.

ولها التركيب التالي:

```
numD(formula, ..., .hstep = NULL, add.h.control = FALSE)

 setInterval(C, wrt, h)

 setCorners(C, var1, var2, h)

 dfdx(.function, .wrt, .hstep)

 d2fdxdy(.function, .var1, .var2, .hstep)

 d2fdx2(.function, .wrt, .hstep)

 numerical.first.partial(f, wrt, h, av)

 numerical.second.partial(f, wrt, h, av)

 numerical.mixed.partial(f, var1, var2, h, av)
```

الحجج:

Arguments

formula تعبير رياضي

... معالم إضافية وغالبا ماتكون القيم الإفتراضية للمعالم الرياضية.

hstep. الخطوة العددية للفروق المحدودة.

add.h.control تهي الدالة الناتجة ليكون لها خطوة عددية لضمان التقارب.

Examples

```
library(mosaic)
g = numD( a*x^2 + x*y ~ x, a=1)
g(x=2,y=10)

gg = numD( a*x^2 + x*y ~ x&x, a=1)
gg(x=2,y=10)

ggg = numD( a*x^2 + x*y ~ x&y, a=1)
ggg(x=2,y=10)

h = numD( g(x=x,y=y,a=a) ~ y, a=1)
h(x=2,y=10)

f = numD( sin(x)~x, add.h.control=TRUE)
plotFun( f(3,.hstep=h)~h, hlim=range(.00000001,.000001))
ladd( panel.abline(cos(3),0))
```

3- التفاضل العددي

Numerical Differentiation

الدالة fderiv تعطي التفاضل العددي من الرتب 1 وحتى 4 مستخدمة تقريب الفروق المحدودة Finite Difference.

ولها التركيب التالي:

```
fderiv(f, x, n = 1, h = 0, method="central", ...)
```

الحجج:

f الدالة المطلوب تفاضلها.

x النقطة او النقاط التي سيتم عندها التفاضل.

n درجة التفاضل وهي من 1 وحتى 4 .

h طول او حجم الخطوة والقيمة إفتراضية 0 (يجعل البرنامج يختارها ذاتيا).

method واحد من:

“central”, “forward”, or “backward”.

... حجج إضافية وقيم إفتراضية.

Examples

```
library(pracma)
## Not run:
f <- sin
xs <- seq(-pi, pi, length.out = 100)
ys <- f(xs)
y1 <- fderiv(f, xs, n = 1, method = "backward")
y2 <- fderiv(f, xs, n = 2, method = "backward")
```

```
y3 <- fderiv(f, xs, n = 3, method = "backward")
y4 <- fderiv(f, xs, n = 4, method = "backward")
plot(xs, ys, type = "l", col = "gray", lwd = 2,
     xlab = "", ylab = "", main = "Sinus and its
Derivatives")
lines(xs, y1, col=1, lty=2)
lines(xs, y2, col=2, lty=3)
lines(xs, y3, col=3, lty=4)
lines(xs, y4, col=4, lty=5)
grid()
## End(Not run)
```

4- التكامل العددي

Numerical integration

طرق التكامل العددي بإستخدام:

1- Euler

2- Runge-Kutta 4th order

3- Monte Carlo

4- RK4 with discontinuities

5- Ramos method.

و لهم التراكيب التالية:

```
euler(x0, t, f, p, dt)
RK4(x0, t, f, p, dt)
rnum(x0, t, f, p, dt, n)
RK4D(x0, t, f, p, dt, g, tz)
ramos(x0, t, f, p, dt)
```

الدالة euler تستخدم فقط للأغراض التعليمية (التوضيحية) وتستخدم لكي نوجد X في النموذج

$dX/dt = f(t,p,X)$

الدالة RK4 تستخدم طريقة RK ذات الرتبة 4 لحل النموذج

$dX/dt = f(t,p,X)$

عدديا.

الدلة rnum تشبة أو تقوم على نفس مبدأ euler.

الدالة RK4D تتعامل مع الإنقطاعات أو الفجوات في الدالة المطلوب تكاملها.

الدالة ramos تستخدم طريقة رامس للتكامل.

الحجج:

x0 الشرط الأولي

initial condition

t الأزمنة للإخراج

times for output

f دالة النموذج (الطرف الأيسر للمعادلة التفاضلية العادية

right hand side of ODE

p معالم.

dt الخطوات الزمنية.

n عدد الإنجازات.

g دالة الإنقطاعات.

tz أزمنة الإنقطاعات.

Examples

```
library(seem)
# Euler
 model <- list(f=expon)
 t <- seq(0,10,1); dt <- 0.001
 p <- 0.1; X0 <- 1
 X <- euler(X0, t, model$f, p, dt)
 # Runge-Kutta
 model <- list(f=expon)
 t <- seq(0,10,1); dt <- 0.001
 p <- 0.1; X0 <- 1
 X <- RK4(X0, t, model$f, p, dt)
 # Stochastic
 model <- list(f=expon)
 t <- seq(0,10,1); dt <- 0.001
 p <- c(0.1,0.01); X0 <- 1
 X <- rnum(X0, t, model$f, p, dt,n=20)
 # RK4 with discontinuities
 model <- list(f=expon,z=expon.z,g=expon.g)
 t <- seq(0,100,1); dt <- 0.01
 p <- c(0.02,10,0,-10); X0 <- 100
 X <- RK4D(X0, t, model$f, p, dt, model$g, model$z(t,p,X))
```

5- المشتقة وعكسها

Derivative and Anti-derivative operators

عمال لحساب المشتقة وعكسها كدالة. ولها التراكيب التالية:

```
 D(formula, ..., .hstep = NULL, add.h.control = FALSE)

antiD(formula, ..., lower.bound = 0,force.numeric = FALSE)
```

```
makeAntiDfun(.function, .wrt, from,.tol =
.Machine$double.eps^0.25)

numerical.integration(f, wrt, av, args, vi.from, ciName =
"C", .tol)
```

الحجج:

formula صيغة.

... القيم الإفتراضية.

hstep. المسافات الأفقية بين النقاط المستخدمة لحساب المشتقات العددية.

add.h.control قيمة منطقية تستخدم لحساب المشتقة العددية.

lower.bound الحد الأدنى لحساب المشتقة العددية.

force.numeric إذا كان TRUE يجرى تكامل عددي حتى لو كان يوجد تكامل رمزي symbolic integral.

function. الدالة المراد تكاملها.

wrt. صف من الحروف يسمي متغير التكامل.

from القيمة الإفتراضية للحد الأدنى لمجال التكامل.

f دالة.

wrt صف حرفي يسمي المتغير.

av قائمة من الحجج الممررة للدالة.

args القيم الإفتراضية للمعالم إذا وجدت.

tol. الحد العددي المسموح للخطأ.

Examples

```
library(mosaic)
D(sin(t) ~ t)
D(A*sin(t) ~ t )
D(A*sin(2*pi*t/P) ~ t, A=2, P=10) # default values for
parameters.
```

```
f <- D(A*x^3 ~ x + x, A=1) # 2nd order partial -- note,
it's a function of x
f(x=2)
f(x=2,A=10) # override default value of parameter A
g <- D(f(x=t, A=1)^2 ~ t)  # note: it's a function of t
g(t=1)
gg <- D(f(x=t, A=B)^2 ~ t, B=10)  # note: it's a function
of t and B
gg(t=1)
gg(t=1, B=100)
f <- makeFun(x^2~x)
D(f(cos(z))~z) #will look in user functions also
antiD( a*x^2 ~ x)
antiD( A/x~x )
F <- antiD( A*exp(-k*t^2 ) ~ t, A=1, k=0.1)
F(t=Inf)
one = makeFun(1~x&y)
by.x = antiD( one(x=x, y=y) ~x)
by.xy = antiD(by.x(x=sqrt(1-y^2), y=y)~y)
4*by.xy(y=1) #area of quarter circle
```

6- المشتقات الرمزية

Symbolic Derivatives

تعطي المشتقات الرياضية لبعض التعابير. ولها الركيب التالي:

```
symbolicD(formula, ..., .order = NULL)
```

الحجج:

formula تعبير رياضي.

... معالم إضافية وغالبا ما تكون القيم الإفتراضية للمعالم الرياضية.

order. رقم يحدد درجة المشتقة لمتغير واحد.

Examples

```
library(mosaic)
symbolicD( a*x^2 ~ x)
symbolicD( a*x^2 ~ x&x)
symbolicD( a*sin(x)~x, .order=4)
symbolicD( a*x^2*y+b*y ~ x, a=10, b=100 )
```

7- رسم تعابير رياضية

Plotting mathematical expressions

تستخدم الدالة plotFun لرسم تعابير رياضية لمتغير او اثنين ولها التركيب التالي:

```
plotFun(object, ..., add = FALSE, xlim = NULL,
   ylim = NULL, npts = NULL, ylab = NULL, xlab = NULL,
   zlab = NULL, filled = TRUE, levels = NULL,
   nlevels = 10, labels = TRUE, surface = FALSE,
   groups = NULL, col.regions = topo.colors, type = "l",
   alpha = NULL)
```

الحجج:

object تعبير رياضي أو دالة لمتغير واحدة.

add إذا كان TRUE يضاف الرسم إلى رسم حالي.

xlim حدود المحور الافقي.

ylim حدود المحور العامودي.

npts عدد النقاط التي يتم رسمها.

xlab إسم المحور الافقي.

ylab إسم المحور العامودي.

zlab إسم المحور الثالث للرسوم السطحية surface-plot.

col لون الرسم.

filled ملي المناطق في الرسم باللون.

levels مستويات الرسم لرسوم الكنتور.

nlevels عدد الكنتورات المرسومة.

labels إذا كانت FALES فلا تسمى الكنتورات.

surface يرسم رسم سطحي بدل الرسم الكنتوري.

col.regions متجه من الالوان أو دالة.

type نوع الرسم.

alpha عدد من 0 (شفاف) إلى 1 (غير شفاف) لألوان الملئ.

groups حجة التجميع لرسوم الشبكية Lattice graphics.

... معالم إضافية.

Examples

```
library(mosaic)
plotFun( a*sin(x^2)~x, xlim=range(-5,5), a=2 )
plotFun( u^2 ~ u, ulim=c(-4,4) )
plotFun( y^2 ~ y, ylim=c(-2,20), y.lim=c(-4,4) )
plotFun( x^2 -3 ~ x, xlim=c(-4,4), grid=TRUE )
ladd( panel.abline(h=0,v=0,col='gray50') )
plotFun( (x^2 -3) * (x^2 > 3) ~ x, type='h', alpha=.1,
lwd=4, col='lightblue', add=TRUE )
plotFun( sin(x) ~ x,
   groups=cut(x, findZeros(sin(x) ~ x, within=10)$x),
   col=c('blue','green'), lty=2, lwd=3, xlim=c(-10,10) )
plotFun( sin(2*pi*x/P)*exp(-k*t)~x+t,k=2,P=.3)
f <- rfun( ~ u & v )
plotFun( f(u=u,v=v) ~ u & v, u.lim=range(-3,3),
v.lim=range(-3,3) )
plotFun( u^2 + v < 3 ~ u & v, add=TRUE, npts=200 )
```

```
model <- lm(wage ~ poly(exper,degree=2), data=CPS85)
fit <- makeFun(model)
xyplot(wage ~ exper, data=CPS85)
plotFun(fit(exper) ~ exper, add=TRUE, lwd=8)
plotFun(fit, add=TRUE, lwd=2, col='white')
plotFun( sin(k*x)~x, k=0.01 )
```

حل مشاكل الأمثلية بواسطة R:

سوف نستخدم ثلاثة حزم رياضية لإستعراض مقدرات لغة النمذجة الرياضية R في حل مشاكل الأمثلية.
أول هذه الحزم هي حزمة 'Package 'Rglpk وهو واجة إستخدام في R لعدة GNU Linear Programing Kit للبرمجة الخطية المسماة GLPK وهي حزمة مجانية والتي تقوم بحل مشاكل واسعة النطاق في البرمجة خطية والبرمجة العددية والبرمجة المختلطة وغيرها.
الحزمة الثانية هي حزمة 'Package 'lpSolve وهي حزمة متوفرة مجانا لحل مشاكل البرمجة الخطية والعددية والمختلطة ومشاكل التخصيص والنقل وغيرها.
الحزمة الثالثة هي حزمة 'Package 'linprog ويستخدم لحل مشاكل البرمجة الخطية ومشاكل الأمثلية بإستخدام خوارزم السمبلكس Simplex Algorithm.

أولا: حزمة Rglpk وتستخدم كالتالي:

```
Rglpk_solve_LP(obj, mat, dir, rhs, types = NULL, max =
FALSE, bounds = NULL, verbose = FALSE)
```

حيث:

obj متجة يحوي معامل دالة الهدف.
mat متجه أو مصفوفة معامل القيود.
dir متجة نصي وفيه إتجاهات القيود ويجب ان تكون ">", "=>", "<", "=<", or "==".
rhs الطرف الأيمن من القيود.
types متجة يبين انواع متغيرات الهدف ويجب ان يكون "B" للثنائي و "C" للمستمر و "I" للصحيح والقيمة الإفتراضية هي "C".
max قيمة منطقية تعطي إتجاه الأمثلية فالقيمة TRUE تعني تعظيم maximize دالة الهدف و FALSE تعني التصغير minimize.
bounds وقيمتها الإفتراضية NULL أو قائمة بعناصر عليا upper ودنيا lower تحوي فهارس indices والحدود التابعة لها لمتغيرات الهدف القيمة الإفتراضية لكل متغير هي بين 0 و ∞.
verbose قيمة منطقية لكي نخبر الحلال solver بأن يعطي مخرجات إضافية أو لا والقيمة الإفتراضية FALSE.

المخرجات:

قائمة تحوي الحل الأمثل optimal solution مع التالي:
solution الحل وهو متجه للعوامل المثلى optimal coefficients.
objval قيمة دالة الهدف عند الحل الأمثل.
status وهو عدد صحيح يعطي معلومات الحالة عن الحل ويعطي 0 إذا كان الحل أمثل.
وقيمة غير صفرية غير ذلك.

حل المثال 1:

$$
\begin{aligned}
Max \quad & 8x_1 + 5x_2 \\
St \quad & 2x_1 + x_2 \le 1000 \\
& 3x_1 + 4x_2 \le 2400 \\
& x_1 + x_2 \le 700 \\
& x_1 - x_2 \le 350 \\
& x_1, x_2 \ge 0
\end{aligned}
$$

```
> library(Rglpk)
> obj <- c(8, 5)
> mat <- matrix (c(2,1,3,4,1,1,1,-1),nrow=4,byrow=TRUE)
> dir <- c("<=", "<=", "<=", "<=")
> rhs <- c(1000,2400,700,350)
> Rglpk_solve_LP(obj, mat, dir, rhs, max = TRUE)
$optimum
[1] 4360

$solution
[1] 320 360

$status
[1] 0
```

مثال 2:

$$
\begin{aligned}
Max. \quad & 12000X_1 + 20000X_2 \\
St. \quad & 2X_1 + 6X_2 \le 27 \\
& X_2 \ge 2 \\
& 3X_1 + X_2 \le 19 \\
& X_1, X_2 \ge 0 \text{ and integers}
\end{aligned}
$$

```
> library(Rglpk)
> obj <- c(12000,20000)
> mat <- matrix (c(2,6,0,1,3,1),nrow=3,byrow=TRUE)
> dir <- c("<=", ">=", "<=")
> rhs <- c(27,2,19)
> s1 <- Rglpk_solve_LP(obj, mat, dir, rhs, types = "I", max
= TRUE)
> print(s1)
$optimum
[1] 108000
```

```
$solution
[1] 4 3

$status
[1] 0

>
```

ثانيا الحل بإستخدام الحزمة linprog

لحل النظام

$$minimize: c'x,$$
$$subject\ to\ Ax \le b,\ x \ge 0$$

وتستخدم كالتالي:

```
solveLP( cvec, bvec, Amat, maximum=FALSE, maxiter=1000,
zero=1e-10, lpSolve=FALSE, verbose=FALSE )
```

حيث:

cvec المتجه c ويحوي n عنصر.
bvec المتجة b ويحوي m عنصر.
Amat المصفوفة A بأبعاد n x m.
maximum قيمة منطقية لتعظيم أو تصغير دالة الهدف والقيمة الإفتراضية تصغير.
maxiter أقصى عدد للتكرار.
zero تخبر البرنامج في ان يضع الأرقام التي هي اصغر من هذا العدد مساوية للصفر.
lpSolve قيمة منطقية لإخبار البرنامج لإستخدام الحزمة 'lpSolve' أو لا. القيمة الإفتراضية لا.
verbose قيمة منطقية إختيارية لإخبار البرنامج لطباعة النتائج المرحلية أو لا. والقيمة الإفتراضية لا.

حل مثال 1:

```
> library( linprog )
> cvec <- c(8,5)
> bvec <- c(1000,2400,700,350)
> Amat <- rbind( c( 2, 1),c( 3,4 ),c( 1, 1 ), c( 1, -1 ))
> res <- solveLP( cvec, bvec, Amat, TRUE )
> print.solveLP( res )


Results of Linear Programming / Linear Optimization
```

```
Objective function (Maximum):
[1] 4360

Iterations in phase 1: 0
Iterations in phase 2: 3

Basic Variables
    opt
1   320
2   360
S 3  20
S 4 390

Constraints
   max actual diff dual price dual.reg
1 1000   1000    0        3.4      400
2 2400   2400    0        0.4      650
3  700    680   20        0.0       20
4  350    -40  390        0.0      390

All Variables (including slack variables)
    opt c     min c    max c marg. marg.reg.
1   320 8  3.750000 10.00000    NA        NA
2   360 5  4.000000 10.66667    NA        NA
S 1   0 0        NA       NA  -3.4       400
S 2   0 0        NA       NA  -0.4       650
S 3  20 0        NA  2.00000   0.0        20
S 4 390 0 -0.666667  2.42857   0.0       390
```

ثالثا الحل بإستخدام الحزمة lpSolve

وتستخدم كالتالي:

```
lp (direction = "min", objective.in, const.mat, const.dir,
const.rhs, transpose.constraints = TRUE, int.vec,
presolve=0, compute.sens=0, binary.vec, all.int=FALSE,
all.bin=FALSE, scale = 196, dense.const, num.bin.solns=1,
use.rw=FALSE)
```

حيث:

direction صف حرفي يعطي إتجاة الأمثلية تصغير (وهو القيمة الإفتراضية) أو تعظيم.
objective.in متجه عددي يحوي معاملات دالة الهدف.

const.mat مصفوفة عددية لمعاملات القيود صف واحد لكل قيد وعمود واحد لكل متغير قرار (ماعدى في حالة transpose.constraints = FALSE).
const.dir متجه من النصوص يعطي إتجاه القيد ويجب أن تكون أحد التالي ("=" "<=" "<" ">=" "==" "=>").
const.rhs متجه من القيم العددية للجة اليمنى من القيود.
transpose.constraints في العادة كل قيد يشغل سطر من const.mat وهذه المصفوفة تحتاج إلى عكس transpose قبل تمريرها لخوارزم الأمثلية. ولهذا في حالة مصفوفة قيود كبيرة جدا قد يكون من الأفضل وضع القيود في المصفوفة عمود بعمود وفي هذه الحالة نضع transpose.constraints = FALSE.
int.vec متجه عددي يعطي مؤشرات indices المتغيرات المطلوب أن تكون صحيحة.
presolve عدد القيمة الإفتراضية له 0.
compute.sens عدد لحساب الحساسية قيمته الإفتراضية 0 أي لا.
binary.vec متجه عددي يعطي مؤشرات indices المتغيرات المطلوب أن تكون ثنائية.
all.int قيمة منطقية. هل كل المتغيرات يجب أن تكون صحيحة؟ القيمة الإفتراضية: FALSE.
all.bin قيمة منطقية. هل كل المتغيرات يجب أن تكون ثنائية؟ القيمة الإفتراضية: FALSE.
scale عدد صحيح لقيمة القياس (لجعل جميع القيم العددية بنفس القياس) القيمة الإفتراضية 196 و 0 لعدم تغيير القياس.
dense.const مرصوصة array كثيفة للقيود ذات ثلاثة أعمدة. وتهمل في حال إعطاء قيمة const.mat .
num.bin.solns عدد صحيح. إذا كان all.bin = TRUE فإن المستخدم يستطيع طلب العدد المعطى من الحلول المثلى.
use.rw منطقي إذا كان TRUE وكان num.bin.solns >1 فهو يعطي كل حل على إنفراد.

المخرجات:

عن طريق النداء: lp.object.

مثال 1:

```
> library(lpSolve)
> f.obj <- c(8, 5)
> f.con <- matrix (c(2,1,3,4,1,1,1,-1),nrow=4,byrow=TRUE)
> f.dir <- c("<=","<=","<=","<=")
> f.rhs <- c(1000,2400,700,350)
> s1 <- lp ("max", f.obj, f.con, f.dir, f.rhs)
> lp ("max", f.obj, f.con, f.dir, f.rhs)$solution
[1] 320 360
> print(s1)
Success: the objective function is 4360
>
```

مثال:

$$max: x_1 + 9x_2 + x_3$$
$$s.t: \quad x_1 + 2x_2 + 3x_3 \le 9$$
$$3x_1 + 2x_2 + 2x_3 \le 15$$

```
> library(lpSolve)
> f.obj <- c(1, 9, 3)
> f.con <- matrix (c(1, 2, 3, 3, 2, 2), nrow=2, byrow=TRUE)
> f.dir <- c("<=", "<=")
> f.rhs <- c(9, 15)
> lp ("max", f.obj, f.con, f.dir, f.rhs)
Success: the objective function is 40.5
> lp ("max", f.obj, f.con, f.dir, f.rhs)$solution
[1] 0.0 4.5 0.0
```

تحليل الحساسية:

```
> lp ("max", f.obj, f.con, f.dir, f.rhs,
compute.sens=TRUE)$sens.coef.from
[1] -1e+30  2e+00 -1e+30
> lp ("max", f.obj, f.con, f.dir, f.rhs,
compute.sens=TRUE)$sens.coef.to
[1] 4.50e+00 1.00e+30 1.35e+01

> lp ("max", f.obj, f.con, f.dir, f.rhs,
compute.sens=TRUE)$duals
[1]   4.5   0.0  -3.5   0.0 -10.5
> lp ("max", f.obj, f.con, f.dir, f.rhs,
compute.sens=TRUE)$duals.from
[1]  0e+00 -1e+30 -1e+30 -1e+30 -6e+00
> lp ("max", f.obj, f.con, f.dir, f.rhs,
compute.sens=TRUE)$duals.to
[1] 1.5e+01 1.0e+30 3.0e+00 1.0e+30 3.0e+00
```

إعادة الإجراء مع جعل جميع المتغيرات صحيحة:

```
> lp ("max", f.obj, f.con, f.dir, f.rhs, int.vec=1:3)
Success: the objective function is 37
> lp ("max", f.obj, f.con, f.dir, f.rhs,
int.vec=1:3)$solution
[1] 1 4 0
> lp ("max", f.obj, f.con, f.dir, f.rhs, int.vec=1:3,
compute.sens=TRUE)$duals
[1] 1 0 0 7 0
```

البرمجة العددية لمشكلة تخصيص:

يستخدم الأمر

```
lp.assign (cost.mat, direction = "min", presolve = 0,
compute.sens = 0)
```

حيث:

cost.mat مصفوفة التكلفة العنصر ij هو تكلفة تخصيص المصدر i للمقصد j.
direction متجة نصوص ذا طول 1 يحوي إما "min" وهي القيمة الإفتراضية أو "max".
presolve قيمة عددية وتوضع مساوية للصفر.
compute.sens قيمة عددية وإفتراضيا تساوي الصفر وأي قيمة غير صفرية تؤدي لحساب الحساسية.

مخرجات: lp.object.

مثال:

مصفوفة التكلفة:

$$\begin{pmatrix} 2 & 7 & 7 & 1 \\ 7 & 7 & 2 & 9 \\ 7 & 3 & 8 & 8 \\ 2 & 2 & 10 & 2 \end{pmatrix}$$

```
> assign.costs <- matrix (c(2, 7, 7, 2, 7, 7, 3, 2, 7, 2,
8,10, 1, 9, 8, 2), 4, 4)
> assign.costs
     [,1] [,2] [,3] [,4]
[1,]    2    7    7    1
[2,]    7    7    2    9
[3,]    7    3    8    8
[4,]    2    2   10    2
> lp.assign (assign.costs)$solution
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    1
[2,]    0    0    1    0
[3,]    0    1    0    0
[4,]    1    0    0    0
>
```

الأمر lp.object يعطي LP شيئ وهو هيكل لـ lp شيئ. وهو قائمة تحوي التالي:

direction إتجاه الأفضلية كما في المدخل.

x.count عدد متغيرات القرار في دالة الهدف.
objective متجه معاملات دالة الهدف كما في المدخل.
const.count عدد القيود المدخلة.
constraint مصفوفة القيود كما ادخلت.
int.count عدد المتغيرات الصحيحة.
int.vec متجه مؤشرات المتغيرات الصحيحة كما ادخلت.
objval قيمة دالة الهدف عند الأمثلية.
solution متجة المعاملات المثلى.
num.bin.solns مؤشر عددي لعدد الحلول الناتجة.
status مؤشر عددي 0 = نجاح و 2= لايوجد حل ممكن.

برمجة مشكلة نقل:

يستخدم الأمر:

```
lp.transport (cost.mat, direction="min", row.signs,
row.rhs, col.signs, col.rhs, presolve=0, compute.sens=0,
integers = 1:(nc*nr) )
```

حيث:

cost.mat مصفوفة التكلفة العنصر *ij* هو تكلفة تخصيص المصدر *i* للمقصد *j*.
direction متجة نصوص ذا طول 1 يحوي إما "min" وهي القيمة الإفتراضية أو "max".
row.signs متجة من النصوص يعطى إتجاهات قيود الأسطر وكل قيمة يجب أن تكون أحد التالي:
">=" ">" "==" "=" "=<" "<".
row.rhs متجة من القيم العددية للجهة اليمنى من قيود الأسطر.
col.signs متجة من النصوص يعطى إتجاهات قيود الأعمدة وكل قيمة يجب أن تكون أحد التالي:
">=" ">" "==" "=" "=<" "<".
col.rhs متجة من القيم العددية للجهة اليمنى من قيود الأعمدة.
presolve عدد القيمة الإفتراضية له 0.
compute.sens عدد لحساب الحساسية قيمته الإفتراضية 0 أي لا.
integers متجه من الأعداد الصحيحة عنصره i يعطي مؤشر المتغير الصحيح رقم i والذي طوله هو عدد المتغيرات الصحيحة. اقيمة الإفتراضية جميع المتغيرات صحيحة.

المخرجات: lp.object.

مثال:

```
> library(lpSolve)
> costs <- matrix(10000,8,5)
> costs[4,1] <- costs[-4,5] <- 0
> costs[1,2] <- costs[2,3] <- costs[3,4] <- 7
> costs[1,3] <- costs[2,4] <- 7.7
```

```
> costs[5,1] <- costs[7,3] <- 8
> costs[1,4] <- 8.4
> costs[6,2] <- 9
> costs[8,4] <- 10
> costs[4,2:4] <- c(.7, 1.4, 2.1)
> row.signs <- rep ("<", 8)
> row.rhs <- c(200, 300, 350, 200, 100, 50, 100, 150)
> col.signs <- rep (">", 5)
> col.rhs <- c(250, 100, 400, 500, 200)
> lp.transport (costs, "min", row.signs, row.rhs,
col.signs, col.rhs)
Success: the objective function is 7790
> lp.transport (costs, "min", row.signs, row.rhs,
col.signs, col.rhs)$solution
     [,1] [,2] [,3] [,4] [,5]
[1,]    0  100  100    0    0
[2,]    0    0  200  100    0
[3,]    0    0    0  350    0
[4,]  200    0    0    0    0
[5,]   50    0    0    0   50
[6,]    0    0    0    0   50
[7,]    0    0  100    0    0
[8,]    0    0    0   50  100
>
```

الأمر print.lp يقوم بطباعة lp.object .
ويستخدم

```
print(x,…)
```

حيث x يساوي lp.object.

مثال 3:

$$
\begin{aligned}
&Min\ 35X_{11}+20X_{12}+40X_{13}+32X_{14}+37X_{21}+40X_{22}+42X_{23}+25X_{24}+40X_{31}+15X_{32}+20X_{33}+28X_{34}\\
&St \quad X_{11}+X_{12}+X_{13}+X_{14} \le 1200\\
&\qquad X_{21}+X_{22}+X_{23}+X_{24} \le 1000\\
&\qquad X_{31}+X_{32}+X_{33}+X_{34} \le 800\\
&X_{11}+X_{21}+X_{31} = 1100\\
&X_{12}+X_{22}+X_{32} = 400\\
&X_{13}+X_{23}+X_{33} = 750\\
&X_{14}+X_{24}+X_{34} = 750\\
&X_{ij} \ge 0, \quad for\ all\ i\ and\ j
\end{aligned}
$$

$$
\begin{pmatrix}
35 & 20 & 40 & 32\\
37 & 40 & 42 & 25\\
40 & 15 & 20 & 28
\end{pmatrix}
$$

```
> library(lpSolve)
> costs <- matrix
(c(35,20,40,32,37,40,42,25,40,15,20,28),nrow=3,byrow=TRUE)
> row.signs <- rep ("<=", 3)
> row.rhs <- c(1200,1000,800)
> col.signs <- rep ("=", 4)
> col.rhs <- c(1100, 400,750, 750)
> s2 <- lp.transport (costs, "min", row.signs, row.rhs,
col.signs, col.rhs)
> s2
Success: the objective function is 80500
> summary(s2)
              Length Class  Mode
direction      1     -none- numeric
rcount         1     -none- numeric
ccount         1     -none- numeric
costs         13     -none- numeric
rsigns         3     -none- numeric
rrhs           3     -none- numeric
csigns         4     -none- numeric
crhs           4     -none- numeric
objval         1     -none- numeric
int.count      1     -none- numeric
integers      12     -none- numeric
solution      12     -none- numeric
presolve       1     -none- numeric
compute.sens   1     -none- numeric
sens.coef.from 12     -none- numeric
sens.coef.to   12     -none- numeric
duals          12     -none- numeric
duals.from     12     -none- numeric
duals.to       12     -none- numeric
status         1     -none- numeric
> s2[12]
$solution
     [,1] [,2] [,3] [,4]
[1,]  850  350    0    0
[2,]  250    0    0  750
[3,]    0   50  750    0

>
> library(Rglpk)
> obj <- c(35,20,40,32,37,40,42,25,40,15,20,28)
> mat <- matrix (c(1,1,1,1,0,0,0,0,0,0,0,0,
+  0,0,0,0,1,1,1,1,0,0,0,0,
+  0,0,0,0,0,0,0,0,1,1,1,1,
```

```
+  1,0,0,0,1,0,0,0,1,0,0,0,
+  0,1,0,0,0,1,0,0,0,1,0,0,
+  0,0,1,0,0,0,1,0,0,0,1,0,
+  0,0,0,1,0,0,0,1,0,0,0,1),nrow=7,byrow=TRUE)
> dir <- c(rep("<=",3),rep("==",4))
> rhs <- c(1200,1000,800,1100,400,750,750)
> s3 <- Rglpk_solve_LP(obj, mat, dir, rhs, types = "I")
> s3
$optimum
[1] 80500

$solution
 [1] 850 350   0   0 250   0   0 750   0  50 750   0

$status
[1] 0

>
```

```
> library(lpSolve)
> costs <- matrix
(c(35,20,40,32,37,40,42,25,40,15,20,28),nrow=3,byrow=TRUE)
> row.signs <- rep ("<=", 3)
> row.rhs <- c(1200,1000,800)
> col.signs <- rep ("=", 4)
> col.rhs <- c(1100, 400,750, 750)
> s2 <- lp.transport (costs, "min", row.signs, row.rhs,
col.signs, col.rhs)
> print(s2)
Success: the objective function is 80500
>
```

مثال آخرلمشكلة نقل Transportation Problem

لشركة مصنعين لإنتاج معين والذي يتم توزيعه عن طريق ثلاثة مراكز توزيع. سعر إنتاج الوحدة متساوي في جميع المصانع ولكن تكلفة النقل من المصانع إلى مراكز التوزيع تختلف حسب البيانات في الجدول التالي:

	Distribution Center 1	Distribution Center 2	Distribution Center 3
Plant A	4	6	4
Plant B	6	5	2

عملية النقل تتم مرة في الأسبوع. خلال الأسبوع كل مصنع ينتج على الأكثر 60 وحدة وكل مركز توزيع يحتاج 40 وحدة للتوزيع.

المطلوب تحديد عدد الوحدات التي تنقل من كل مصنع إلى مراكز التوزيع بحيث تكون تكلفة النقل أقل مايمكن؟

```
> library(lpSolve)
> costs <- matrix(c(4,6,4,6,5,2),2,3,byrow=TRUE)
> row.signs <- rep ("<=", 2)
> row.rhs <- c(60,60)
> col.signs <- rep ("==", 3)
> col.rhs <- c(40,40,40)
> lp.transport (costs, "min", row.signs, row.rhs,
col.signs, col.rhs)
Success: the objective function is 460
> lp.transport (costs, "min", row.signs, row.rhs,
col.signs, col.rhs)$solution
     [,1] [,2] [,3]
[1,]   40   20    0
[2,]    0   20   40
>
```

مثال تخصيص Assignment Problem:

الجدول التالي يمثل زمن النقل من خطوط التجميع إلى مناطق الفحص لقطع إلكترونية :

	Inspection Area				
Assembly Line	A	B	C	D	E
1	10	4	6	10	12
2	11	7	7	9	14
3	13	8	12	14	15
4	14	16	13	17	17
5	19	11	17	20	19

المطلوب هو تحديد الكيفية التي تنقل بها القطع من خطوط الإنتاج إلى مناطق الفحص في أقل زمن ممكن مع ملاحظة أن يخصص خط تجميع واحد لمنطقة فحص واحدة.

```
> cost2 <- matrix
(c(10,4,6,10,12,11,7,7,9,14,13,8,12,14,15,14,16,13,17,17,19
,11,17,20,19), 5, 5, byrow=TRUE)
> lp.assign (cost2,direction = "min")$solution
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    1    0    0
[2,]    0    0    0    1    0
[3,]    0    0    0    0    1
[4,]    1    0    0    0    0
```

```
[5,]    0    1    0    0    0
>
```

مشكلة الجراب:

لدينا n من الأشياء. المتجة من المتغيرات الثنائية $x_j,\left(j=1,...,n\right)$ والذي له الخاصية التالية:

$$x_j=\begin{cases}1 & if\ object\ j\ is\ selected\\ 0 & otherwise\end{cases}$$

فإذا كانت $p_j,\left(j=1,...,n\right)$ مقياس أهمية العنصر (شيئ) j و $w_j,\left(j=1,...,n\right)$ حجم (او وزن) العنصر (شيئ) j و c حجم أوسعة الحقيبة الظهرية فالمشكلة هي في إختيار من بين كل المتجهات الثنائية x والتي تحقق القيود

$$\sum_{j=1}^{n} w_j x_j \le c$$

تلك التي تعظم maximizes دالة الهدف

$$\sum_{j=1}^{n} p_j x_j$$

أي

$$\text{maximize } \sum_{j=1}^{n} p_j x_j$$

$$\text{subject to } \sum_{j=1}^{n} w_j x_j \le c$$

$$x_j = 0 \quad or \quad 1, \quad j=1,...,n$$

مثال:

$$n=8$$

$$p_j=\left(15,100,90,60,40,15,10,1\right)$$

$$w_j=\left(2,20,20,30,40,30,60,10\right)$$

$$c=102$$

```
> library(lpSolve)
> f.obj <- c(15,100,90,60,40,15,10,1)
> f.con <- matrix(c(2,20,20,30,40,30,60,10),1,8,byrow=TRUE)
> f.dir <- c("<=")
> f.rhs <- c(102)
> s1 <- lp ("max", f.obj, f.con, f.dir,
f.rhs,binary.vec=1:8)
> lp ("max", f.obj, f.con, f.dir,
f.rhs,binary.vec=1:8)$solution
```

```
[1] 1 1 1 1 0 1 0 0
> s1
Success: the objective function is 280
>
```

مثال آخر:

لنفرض أنك تريد القيام بنزهة سيرا على الأقدام في أحد المرتفعات الجبلية وقد جهزت قائمة بالأشياء التي تحتاج إليها. كل عنصر من هذه الإشياء له وزن معين والحقيبة لاتستوعب أكثر من 15 كجم. ايضا من ضمن القائمة انت وضعت معيار أهمية لكل عنصر فبدت القائمة كالتالي:

Item	Weight	Rating
Ant Repellent	1	2
Pepsi	3	9
Blanket	4	3
Meat	3	8
Cakes	3	10
Football	1	6
Salad	5	4
Watermelon	10	10
Sum	30	

المطلوب تحديد الأشياء المهمة التي تحملها معك مراعيا حجم الحقيبة.

```
> library(lpSolve)
> f.obj <- c(2,9,3,8,10,6,4,10)
> f.con <- matrix(c(1,3,4,3,3,1,5,10),1,8,byrow=TRUE)
> f.dir <- c("<=")
> f.rhs <- c(15)
> s1 <- lp ("max", f.obj, f.con, f.dir,
f.rhs,binary.vec=1:8)
> lp ("max", f.obj, f.con, f.dir,
f.rhs,binary.vec=1:8)$solution
[1] 1 1 1 1 1 1 0 0
> s1
Success: the objective function is 38
>
```

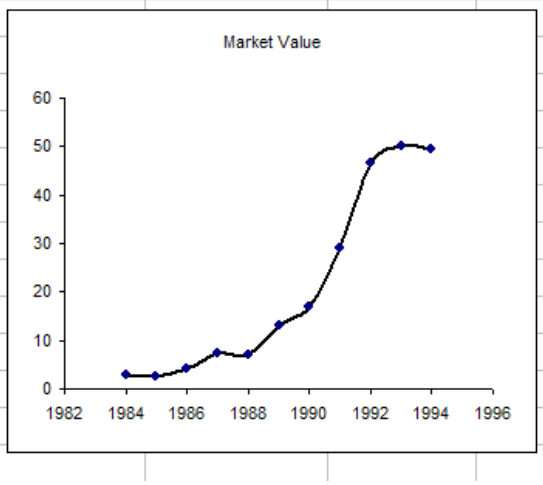
تطبيق المنحنيات وتقدير المعالم Curve Fitting and Parameters Estimation:

البيانات التالية هي نسبة قيمة السوق لمنتجات مكروسوفت و إنتل من سنة 1984حتى سنة 1994

Year	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994
Market Value	3.0	2.5	4.0	7.5	7.0	13.0	17.0	29.0	46.5	50.0	49.5

ندخل هذه البيانات على صفحة نشر ونرسمها.

	A	B	C	D
1	Year	Market Value	Base Year	
2	1984	3	0	
3	1985	2.5	1	
4	1986	4	2	
5	1987	7.5	3	
6	1988	7	4	
7	1989	13	5	
8	1990	17	6	
9	1991	29	7	
10	1992	46.5	8	
11	1993	50	9	
12	1994	49.5	10	
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				



واضح أن المنحني على شكل حرف S وهذه خاصية المنحنى اللوجستي والذي له الشكل الرياضي:

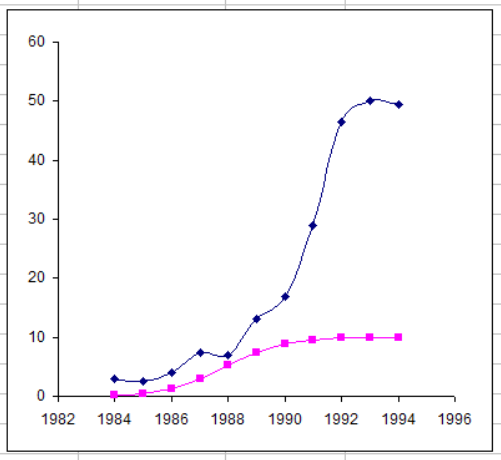
$$x(t)=\frac{a}{1+be^{-ct}},\ t\geq 0$$

من البيانات السابقة نريد تعيين قيم المعالم a و b و c والتي تعطي أفضل تطبيق للبيانات ولذلك ندخل البيانات في صفحة نشر كالتالي:

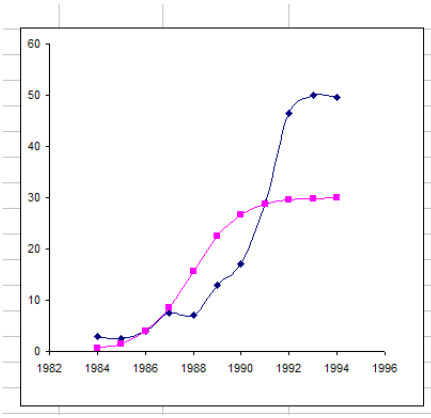
	A	B	C	D	E	F
1	Year	Base Year	Market Va	Fit	Error	
2	1984	0	3	=$F$15/(1+$F$16*EXP(-$F$17*B2))	=(D2-C2)^2	
3	1985	1	2.5	=$F$15/(1+$F$16*EXP(-$F$17*B3))	=(D3-C3)^2	
4	1986	2	4	=$F$15/(1+$F$16*EXP(-$F$17*B4))	=(D4-C4)^2	
5	1987	3	7.5	=$F$15/(1+$F$16*EXP(-$F$17*B5))	=(D5-C5)^2	
6	1988	4	7	=$F$15/(1+$F$16*EXP(-$F$17*B6))	=(D6-C6)^2	
7	1989	5	13	=$F$15/(1+$F$16*EXP(-$F$17*B7))	=(D7-C7)^2	
8	1990	6	17	=$F$15/(1+$F$16*EXP(-$F$17*B8))	=(D8-C8)^2	
9	1991	7	29	=$F$15/(1+$F$16*EXP(-$F$17*B9))	=(D9-C9)^2	
10	1992	8	46.5	=$F$15/(1+$F$16*EXP(-$F$17*B10))	=(D10-C10)^2	
11	1993	9	50	=$F$15/(1+$F$16*EXP(-$F$17*B11))	=(D11-C11)^2	
12	1994	10	49.5	=$F$15/(1+$F$16*EXP(-$F$17*B12))	=(D12-C12)^2	
13				Error sum of squares	=SUM(E2:E12)	
14						
15					a=	10
16					b=	50
17					c=	1

لاحظ أننا أخذنا السنة 1984 كسنة أساس وساويناها بالصفر (المنحنى اللوجستي يجب أن يبدأ من الصفر)، ندخل قيم أولية للمعالم $a=10$ و $b=50$ و $c=1$ ونرسم Market Value و Fit مع الزمن.

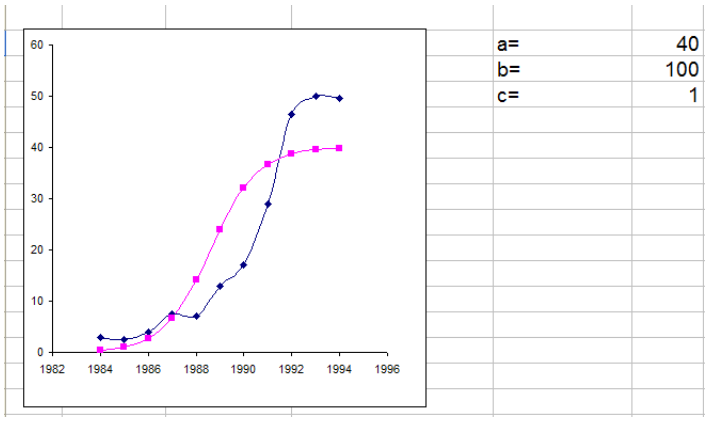
	A	B	C	D	E	F
1	Year	Base Year	Market Val	Fit	Error	
2	1984	0	3	0.196078431	7.861976	
3	1985	1	2.5	0.515624132	3.937748	
4	1986	2	4	1.287537486	7.357453	
5	1987	3	7.5	2.865860462	21.47525	
6	1988	4	7	5.219800734	3.169109	
7	1989	5	13	7.480005851	30.47034	
8	1990	6	17	8.897291025	65.65389	
9	1991	7	29	9.563940746	377.7604	
10	1992	8	46.5	9.835035655	1344.32	
11	1993	9	50	9.938673512	1604.91	
12	1994	10	49.5	9.977351447	1562.04	
13				Error sum of squ	5028.955	
14						
15					a=	10
16					b=	50
17					c=	1
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						



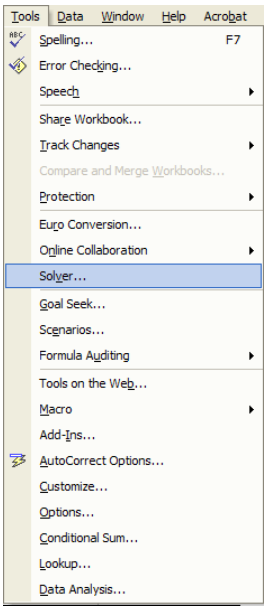
لاحظ أن المنحنى للقيم المطبقة بعيد عن المنحنى الناتج من البيانات لهذا نجرب قيم مختلفة للمعالم كالتالي:



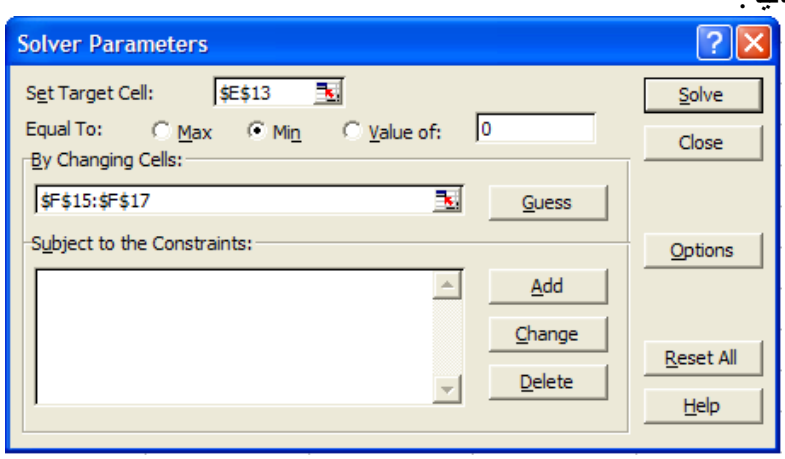
	a=	30
	b=	50
	c=	1



من الشكلين السابقين نجد أن القيم الأولية للمعالم $a=40$ و $b=100$ و $c=1$ مناسبة، نضع المؤشر في الخلية المراد تصغيرقيمتها E13 ونختار Solver من قائمة الأدوات كالتالي:

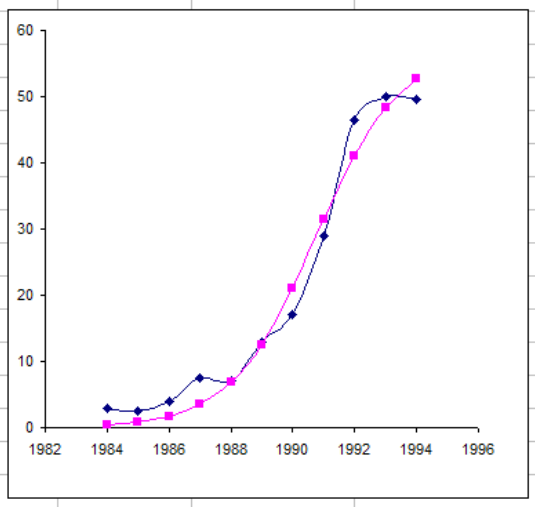


فتظهر النافذة التالية:



لتصغيرقيمته $E$13 (لاحظ العنونة المطلقة) نختار Min ثم في صندوق إختيار القيم التى تغير نختار المجال الموجودة به تقديرات المعالم الأولية أي $F$15:$F$17 ثم Solve فتظهر النتائج التالية:

	A	B	C	D	E	F
1	Year	Base Year	Market Value	Fit	Error	
2	1984	0	3	0.414542265	6.684591701	
3	1985	1	2.5	0.852705443	2.713579359	
4	1986	2	4	1.739946887	5.107840073	
5	1987	3	7.5	3.493689512	16.05052372	
6	1988	4	7	6.800502541	0.039799236	
7	1989	5	13	12.51540951	0.234827947	
8	1990	6	17	21.04921791	16.39616568	
9	1991	7	29	31.36782219	5.606581939	
10	1992	8	46.5	41.08403009	29.3327301	
11	1993	9	50	48.3020243	2.883121473	
12	1994	10	49.5	52.77515711	10.72665407	
13				Error sum of squares	95.7764153	
14						
15					a=	57.76
16					b=	138.3
17					c=	0.729
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						



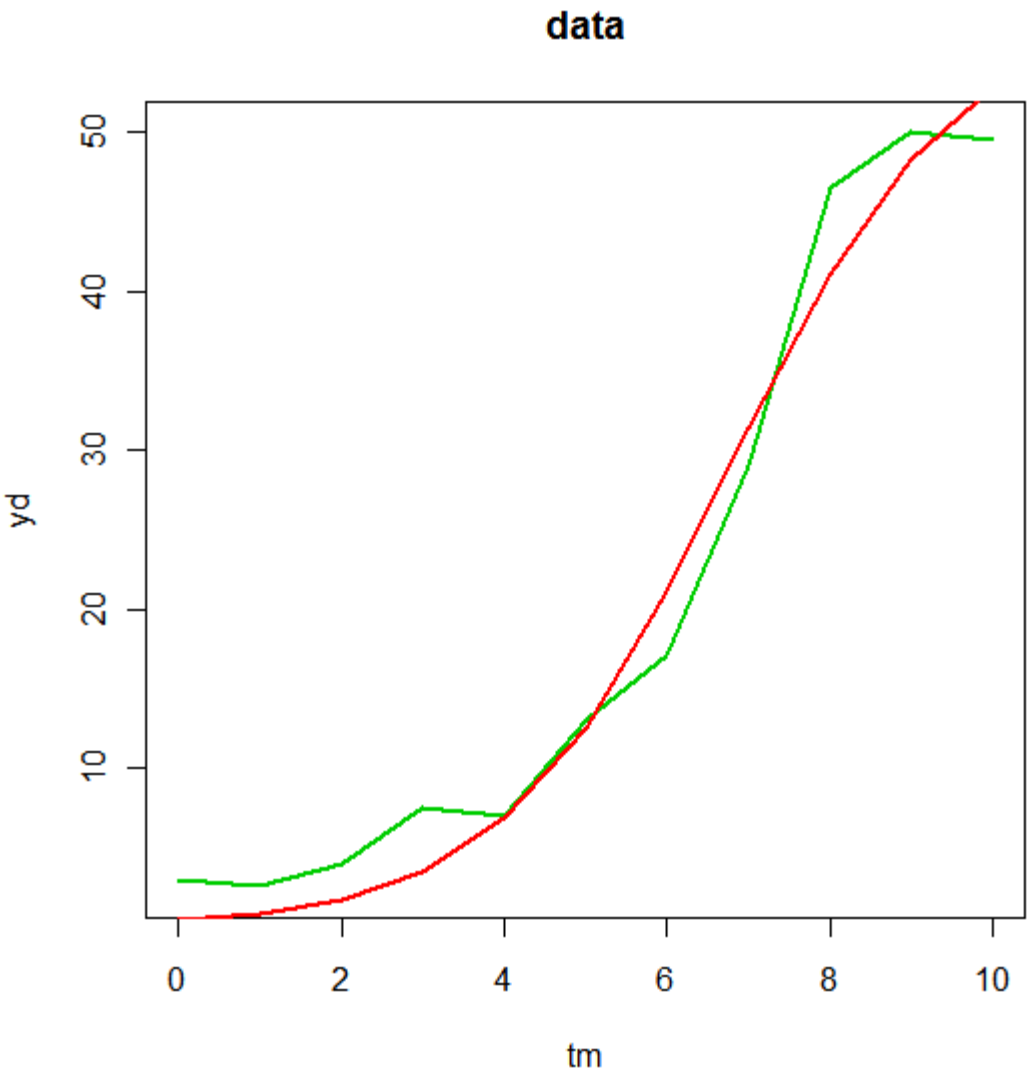
وهذا أفضل تطبيق للمنحنى على البيانات المعطاة. أي أن البيانات تنطبق بشكل جيد على المنحنى

$$x(t)=\frac{57.76}{1+138.3e^{-0.729t}},\ t\geq 0$$

بإستخدام R

```
> library(minpack.lm)
> tm = seq(0,10,length=11)
> getPred <- function(parS, xx) parS$a/(1+parS$b*exp(-parS$c*xx))
> yd = c(3.0,2.5,4.0,7.5,7.0,13.0,17.0,29.0,46.5,50.0,49.5)
> residFun <- function(p, observed, xx) observed - getPred(p,xx)
> parStart <- list(a=40,b=50, c=1)
> nls.out <- nls.lm(par=parStart, fn = residFun, observed = yd,xx = tm,
control = nls.lm.control(nprint=1))
It.    0, RSS =    1180.49, Par. =         40         50          1
It.    1, RSS =    577.266, Par. =    40.0868    69.6474   0.866728
```

```
It.    2, RSS =    274.692, Par. =    47.0803    89.6054   0.704542
It.    3, RSS =    112.953, Par. =    59.7216    98.4505   0.649334
It.    4, RSS =    98.0102, Par. =    56.4299      125.1   0.725997
It.    5, RSS =    95.9791, Par. =    58.3408    127.124   0.711536
It.    6, RSS =    95.8031, Par. =    57.6869    135.903   0.727385
It.    7, RSS =    95.7785, Par. =    57.8297    136.901   0.726888
It.    8, RSS =    95.7766, Par. =     57.758    138.048   0.728667
It.    9, RSS =    95.7764, Par. =    57.7661    138.209   0.728741
It.   10, RSS =    95.7764, Par. =    57.7606    138.317   0.728896
It.   11, RSS =    95.7764, Par. =    57.7609    138.337   0.728912
> plot(tm,yd, main="data",type="l",col=3,lwd=2)
> lines(tm,getPred(as.list(coef(nls.out)), tm), col=2, lwd=2)
> summary(nls.out)

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a  57.7609     6.1406   9.406 1.34e-05 ***
b 138.3368   105.2750   1.314 0.225255
c   0.7289     0.1431   5.092 0.000939 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.46 on 8 degrees of freedom
Number of iterations to termination: 11
Reason for termination: Relative error in the sum of squares is at most
`ftol'.
>
>
```



ملاحظات:

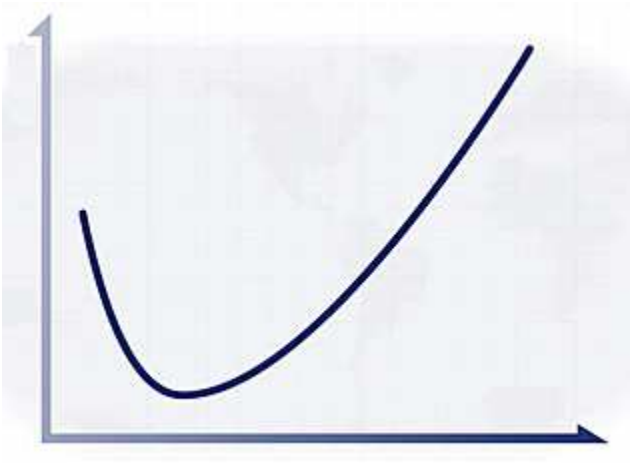
تطبيق المنحنيات في R يعطي تفاصيل أكثر عن مقدرات المعالم:

```
Parameters:
  Estimate Std. Error t value Pr(>|t|)
a  57.7609     6.1406   9.406 1.34e-05 ***
b 138.3368   105.2750   1.314 0.225255
c   0.7289     0.1431   5.092 0.000939 ***
```

مثال آخر لتطبيق المنحنيات

ظاهرة المنحنى J

كثير من الأنظمة الطبيعية تبدي ظاهرة منحنى J والذي فيه يتفاعل متغيرين حالة State Variables بحيث تتناقص قيمة المتغير التابع مع زيادة المتغير المستقل حتى يصل إلى نقطة حرجة ثم يعود في الزيادة ويتخطى نقطة تفاعلة السابقة.



ومن الأمثلة الشهيرة على ظاهرة منحنى J هو تأثير تخفيض سعر الصرف لعملة على ميزان التعاملات مقاس بالعملة المحلية والذي سوف يزداد سوء في فترة صدمة القرار ثم يعود للتحسن. تخفيض معدل الصرف سوف يؤدي لصادرات رخيصة وواردات غالية مما يجعل التوازن الحالى أكثر سوء ولكن بعد فترة فإن حجم الصادرات سوف يبدأ في الإرتفاع بسبب رخص أسعارها بالنسبة للمشتري الأجنبي كما أن المستهلك المحلي سوف يقلل من شرائه للبضائع المستوردة التي أصبحت غالية الثمن. وبالتدريج فإن ميزان المدفوعات أو التعاملات أو التجاري سوف يتحسن.

المنحنى J مثلا قد يعني أن قيمة الإعتماد المالي المجازف Venture Fund تتناقص دائما و المعدل الضمني للربح Internal Rate of Return (IRR) يصبح سالبا في السنوات القليلة القادمة قبل أن يبدأ كليهما في الصعود. ففرضية منحنى J تعني إنتكاس قصير الأمد يتبعه تحسن طويل الأمد.

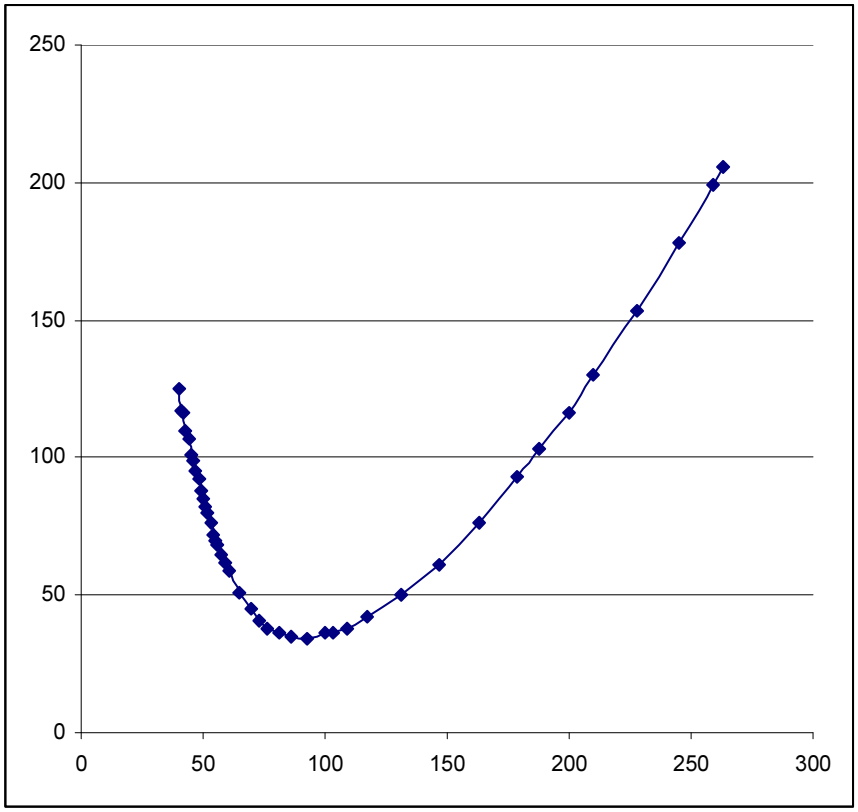
مثال:

البيانات التالية لمتغيري حالة

x	y
40	125
41	117
42	116
43	110
44	107
45	101
46	99
47	95
48	92
49	88
50	85
51	82
52	80
53	76
54	72
55	70
56	68
57	65
59	62
61	59
65	51
70	45
73	41
76	38
81	36
86	35
93	34
100	36
103	36
109	38
117	42

131	50
147	61
163	76
179	93
188	103
200	116
210	130
228	153
245	178
259	199
263	206

الشكل التالي للعلاقة بين x و y



سوف نطبق المنحنى J بالعلاقة الرياضية

$$y = a + bx + cx^2$$

لقيم حقيقية للمعالم a و b و c و $x, y \in R$.

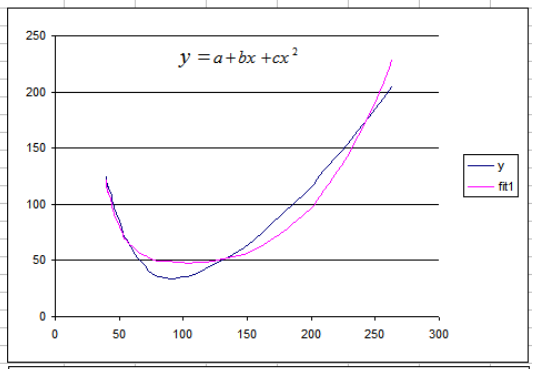
وكذلك العلاقة

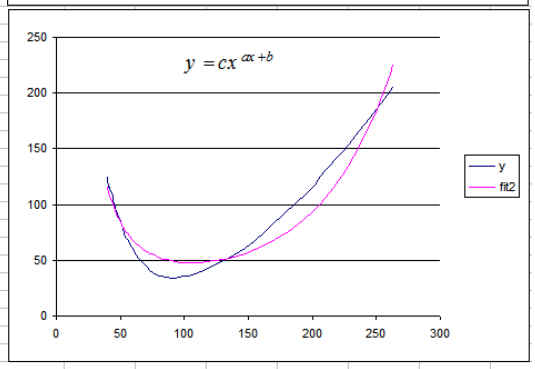
$$y = cx^{ax+b}$$

لقيم حقيقية للمعالم a و b و c و $x, y \in R$.

بإستخدام إكسل نجد

	A	B	C	D	E	F	G	H
1	x	y	fit1	error1	errorS1	fit2	error2	errorS2
2	40	125	121.134	3.865992	14.94589	115.892202	9.107798	82.95198
3	41	117	111.8725	5.127457	26.29082	111.634619	5.365381	28.78731
4	42	116	110.6834	5.31662	28.26645	107.684934	8.315066	69.14032
5	43	110	104.1711	5.828858	33.97558	104.014648	5.985352	35.82443
6	44	107	101.0006	5.999361	35.99233	100.598522	6.401478	40.97892
7	45	101	95.00033	5.999673	35.99608	97.4141385	3.585861	12.8584
8	46	99	93.01958	5.980424	35.76547	94.4415339	4.558466	20.77961
9	47	95	89.25657	5.743433	32.98703	91.6628808	3.337119	11.13636
10	48	92	86.51267	5.487332	30.11081	89.06222	2.93778	8.630551
11	49	88	83.0151	4.984899	24.84922	86.6252304	1.37477	1.889992
12	50	85	80.47028	4.529715	20.51832	84.339031	0.660969	0.43688
13	51	82	78.01079	3.989211	15.91381	82.1920106	-0.19201	0.036868
14	52	80	76.39028	3.609718	13.03007	80.1736805	-0.17368	0.030165
15	53	76	73.34776	2.65224	7.034379	78.2745466	-2.27455	5.173562
16	54	72	70.45692	1.543081	2.381098	76.4859987	-4.486	20.12418
17	55	70	69.02601	0.973986	0.948648	74.8002129	-4.80021	23.04204
18	56	68	67.63303	0.366971	0.134667	73.2100678	-5.21007	27.14481
19	57	65	65.65702	-0.65702	0.431673	71.7090698	-6.70907	45.01162
20	59	62	63.6816	-1.6816	2.827769	68.9512964	-6.9513	48.32052
21	61	59	61.7915	-2.7915	7.792454	66.4852105	-7.48521	56.02838
22	65	51	57.28133	-6.28133	39.45509	62.2931473	-11.2931	127.5352
23	70	45	54.1274	-9.1274	83.3095	58.1724355	-13.1724	173.5131
24	73	41	52.24263	-11.2426	126.3968	56.1778749	-15.1779	230.3679
25	76	38	50.86505	-12.865	165.5094	54.4798877	-16.4799	271.5867
26	81	36	49.73986	-13.7399	188.7838	52.2053874	-16.2054	262.6146
27	86	35	48.97966	-13.9797	195.431	50.5079571	-15.508	240.4967
28	93	34	48.05948	-14.0595	197.669	48.9134285	-14.9134	222.4104
29	100	36	48.12997	-12.13	147.1361	48.0621774	-12.0622	145.4961
30	103	36	47.87577	-11.8758	141.034	47.8919683	-11.892	141.4189
31	109	38	48.06891	-10.0689	101.383	47.8629291	-9.86293	97.27737
32	117	42	48.90788	-6.90788	47.71878	48.4118648	-6.41186	41.11201
33	131	50	51.21032	-1.21032	1.464863	50.8385996	-0.8386	0.703249
34	147	61	55.64221	5.357787	28.70588	55.8199062	5.180094	26.83337
35	163	76	64.02731	11.97269	143.3452	63.3334695	12.66653	160.441
36	179	93	76.28977	16.71023	279.2317	73.841618	19.15838	367.0436
37	188	103	84.81769	18.18231	330.5966	81.3383292	21.66167	469.228
38	200	116	97.29583	18.70417	349.846	93.4867323	22.51327	506.8472
39	210	130	112.7732	17.22676	296.7612	105.846636	24.15336	583.385
40	228	153	142.1011	10.89894	118.787	134.589007	18.41099	338.9646
41	245	178	179.8845	-1.88453	3.551449	171.858327	6.141673	37.72014
42	259	199	216.2253	-17.2253	296.7095	212.583972	-13.584	184.5243
43	263	206	229.3244	-23.3244	544.0262	226.294471	-20.2945	411.8656
44				sse =	4197.045			5579.712
45	a =		50.46	0.003673				
46	b =		-0.08473	-2.21514				
47	c =		0.00474	238486.3				





الصيغ:

لتطبيق المعادلة الأولى نضع في C2

=$B$45+$B$46*A2+$B$47*B2*B2

لتطبيق المعادلة الثانية نضع في F2

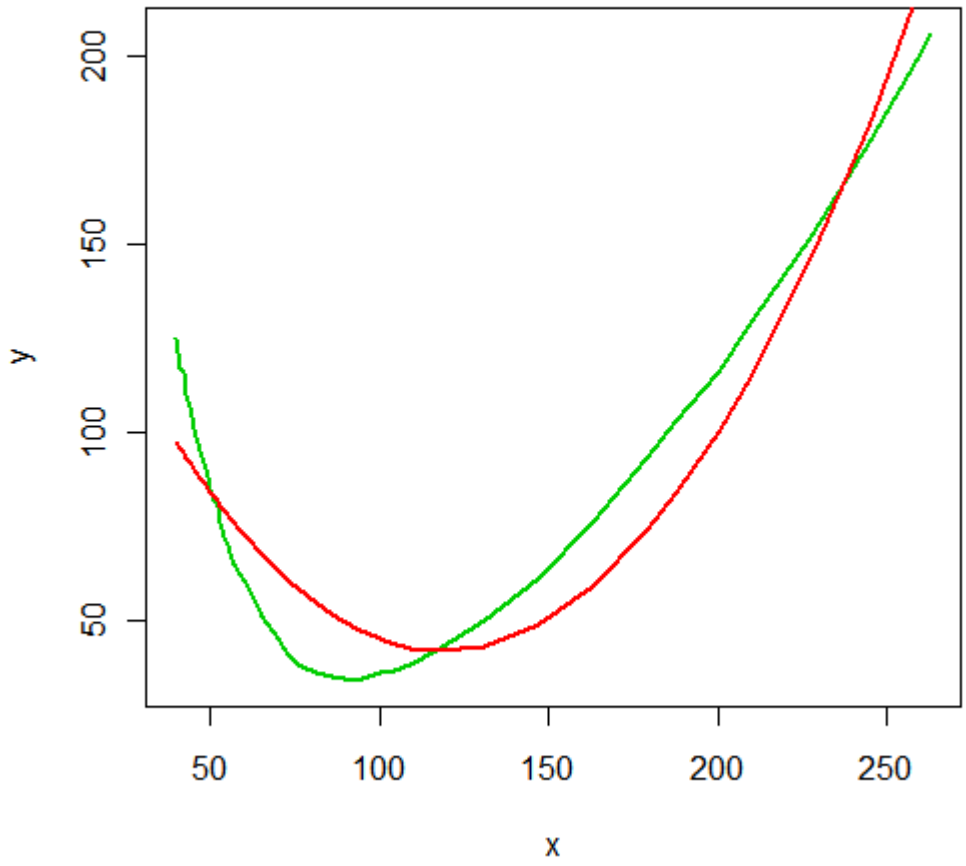
=$C$47*(A2)^($C$45*A2+$C$46)

تمرين:

طبق المعادلتين على البيانات وأستخرج النتائج وأي تطبيق أفضل؟

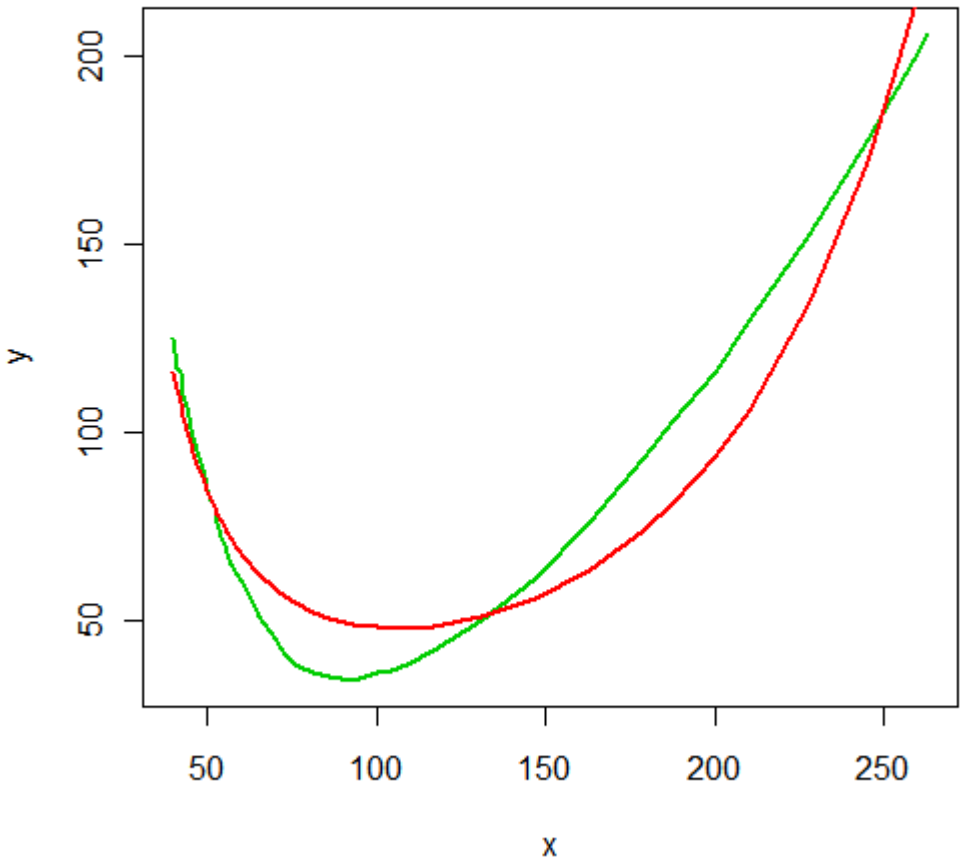
بإستخدام R:

```
> jcurve = read.csv("C:/Documents and Settings/amb/Desktop/ts in R/ts
data/j curve data.csv", header=TRUE)
> x =jcurve[,1]
> y =jcurve[,2]
> library(minpack.lm)
> getPred <- function(parS, xx) parS$a+parS$b*xx+ parS$c*xx*xx
> residFun <- function(p, observed, xx) observed - getPred(p,xx)
> parStart <- list(a=40,b=50, c=1)
> nls.out <- nls.lm(par=parStart, fn = residFun, observed = y ,xx = x,
control = nls.lm.control(nprint=1))
It.    0, RSS = 3.64588e+10, Par. =         40         50          1
It.    1, RSS =    7942.28, Par. =    167.652   -2.11772  0.00889644
It.    2, RSS =    7942.28, Par. =    167.652   -2.11772  0.00889644
> plot(x,y,type="l",col=3,lwd=2)
> lines(x,getPred(as.list(coef(nls.out)), x), col=2, lwd=2)
> summary(nls.out)

Parameters:
    Estimate Std. Error t value Pr(>|t|)
a  1.677e+02  8.857e+00   18.93  < 2e-16 ***
b -2.118e+00  1.692e-01  -12.52  3.1e-15 ***
c  8.896e-03  5.968e-04   14.91  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.27 on 39 degrees of freedom
Number of iterations to termination: 2
Reason for termination: Relative error between `par' and the solution
is at most `ptol'.
>
```

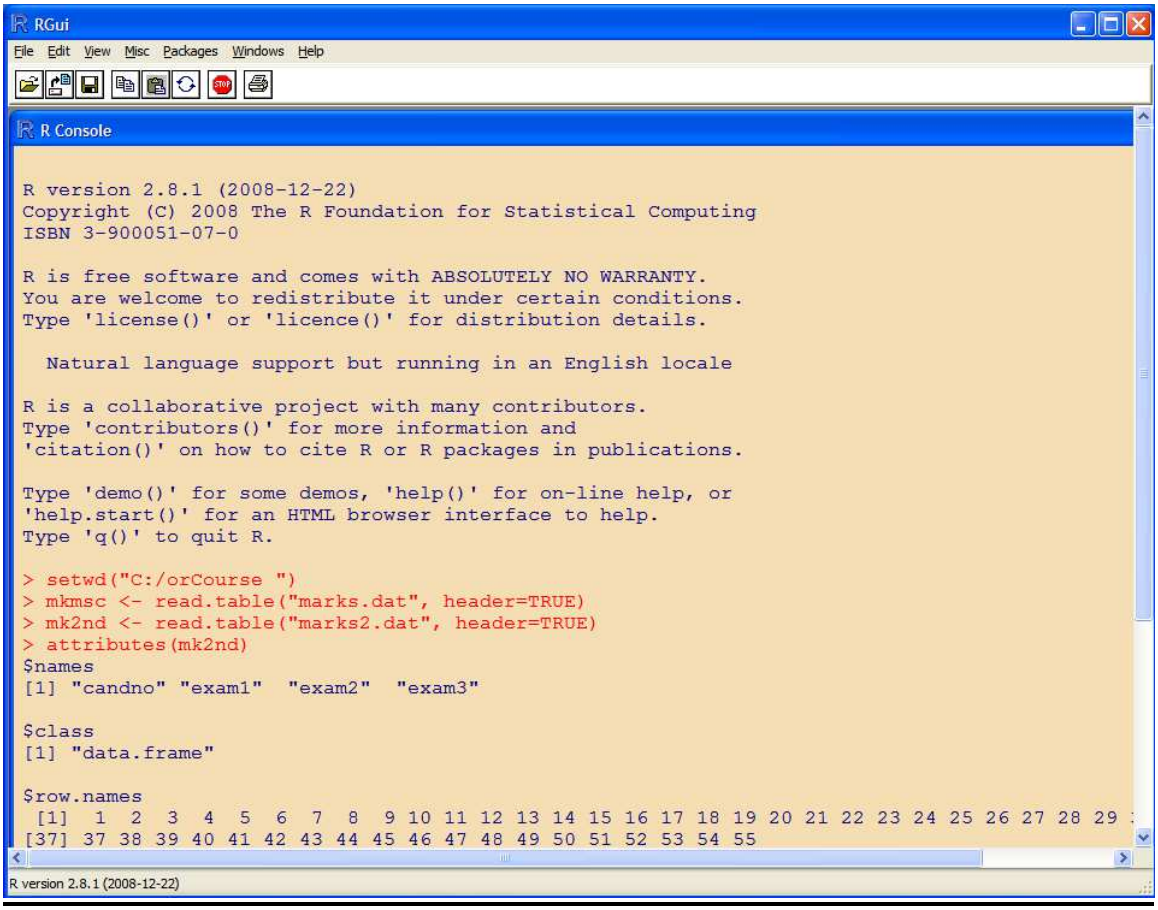



```
>
> getPred <- function(parS, xx) parS$c*((xx)^(parS$a*xx+parS$b))
> residFun <- function(p, observed, xx) observed - getPred(p,xx)
> parStart <- list(a=0.001,b=-2, c=200000)
> nls.out <- nls.lm(par=parStart, fn = residFun, observed = y ,xx = x,
control = nls.lm.control(nprint=1))
It.    0, RSS =     167377, Par. =      0.001         -2     200000
It.    1, RSS =    78730.8, Par. =  0.0038509   -2.06646     135199
It.    2, RSS =    6700.23, Par. =  0.00366134   -2.14178     173335
It.    3, RSS =    5752.83, Par. =  0.00364086   -2.19055     213088
It.    4, RSS =    5592.78, Par. =  0.00367427    -2.2158     237623
It.    5, RSS =    5579.71, Par. =  0.00367349   -2.21553     238867
```

```
It.    6, RSS =    5579.71, Par. =  0.00367368   -2.21566     238978
It.    7, RSS =    5579.71, Par. =  0.00367367   -2.21566     238974
> plot(x,y,type="l",col=3,lwd=2)
> lines(x,getPred(as.list(coef(nls.out)), x), col=2, lwd=2)
> summary(nls.out)

Parameters:
    Estimate Std. Error t value Pr(>|t|)
a  3.674e-03  2.042e-04  17.990   <2e-16 ***
b -2.216e+00  1.477e-01 -14.999   <2e-16 ***
c  2.390e+05  1.297e+05   1.842   0.0731 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.96 on 39 degrees of freedom
Number of iterations to termination: 7
Reason for termination: Relative error in the sum of squares is at most
`ftol'.
>
```


y
200
150
100
50
50
100
150
200
250
x

ملحق:
مقدمة للغة النمذجة الرياضية R



البيانات وإدارة الدورة وإحصائآت بسيطة:

أفكار أساسية:

أشياء *Objects*:

تعمل R على أشياء Objects. جميع الأشياء تمتلك الخواص التالية والتي نطلق عليها الصفات الجوهرية intrinsic attributes :

- نمط أو صيغة mode: ويبين نوع الشيئ - والصيغ الممكنة هنا هي: عددي numeric أو مركب complex أو منطقي logical أو نص character أو قائمة list.

- طول length: عدد المكونات والتي تصنع الشيئ.

وببساطة الشيئ هو طريقة مناسبة لتخزين المعلومات. ففي النمذجة الرياضية نحتاج لتخزين مشاهدات لمتغير ذو أهمية. وهذا يتم عن طريق إستخدام متجه عددي. لاحظ انه لايوجد قيم مفردة *Scalar* في R فالعدد الفردي هو متجه عددي له طول 1. المتجهات نشير إليها كتركيبات ذرية *Atomic Structures* حيث جميع مكوناتها لها نفس النمط أو الصيغة.

إذا كان الشيئ يخزن معلومات فإننا نحتاج إلى تسميته بحيث نستطيع الإشارة اليه لاحقا. يطلق على إسم الشيئ معرف *Identifier* والمعرف يتم إختيارة بواسطة المستخدم. ونستطيع إختيار أي إسم معرف على أن تتوفر فيه الشروط التالية:

- أي تشكيلة من الحروف والأرقام و النقطة ويستحسن أن يكون الإسم واصفا لطبيعة الشيئ.
- لايبدء المعرف بعدد أو نقطة فمثلا المعرف moonbase3.sample مقبول ولكن 3moonbase.sample أو moonbase3.sample. غير مقبولة.
- المعرفات حساسة لنوع الحرف *Case Sensitive* فمثلا moon.sample تختلف عن moon.Sample
- بعض النصوص تعتبر قيم محفوظة مثل c و q و C و D و F و I و T. ولهذا يجب تجنب إستخدامها كمعرفات.

في العادة نحن نهتم بمجموعات بيانات *Data Sets* تتكون من عدة متغيرات ففي R مجموعات البيانات تمثل بشيئ يعرف بإطار البيانات *Data Frame*. وكجميع الأشياء إطار البيانات له الخواص الجوهرية نمط و طول. إطار البيانات له نمط قائمة وطول إطار البيانات هو عدد المتغيرات التي يحويها. وكبقية الأشياء الكبيرة يمتلك إطار البيانات صفات اخرى خلاف النمط والطول وهذه تسمى صفات غير جوهرية *Non-intrinsic Attributes* مثل:

- أسماء names: وهي أسماء المتغيرات التي تكون مجموعة البيانات.
- أسماء.سطور row.names وهي أسماء الأفراد التي تتبع لهم المشاهدات.
- فئة class: يمكن إعتبار هذه الصفة كتوصيف مفصل عن نوع الشيئ وفي هذه الحالة الفئة هي data.frame

صفة الفئة تخبر بعض الدوال *generic functions* كيف تتعامل مع الشيئ فمثلا أشياء من فئة data.frame تعرض على الشاشة بطريقة خاصة.

دوال و حجج وقيم مرجعة

Functions, arguments and return values

تعمل R بواسطة إستدعاء دوال *calling functions* وفكرة الدالة معروفة حيث تأخذ الدالة مجموعة من المدخلات *inputs* وتقوم بتطبقهم *map* لمخرج *output* واحد. في R المدخلات والمخرجات هي أشياء. نطلق على المدخلات حجج *arguments* والمخرجات قيم مرجعة *return values*. الكثير من الدوال المهمة والمفيدة موجودة في تركيب R بينما هناك اخرى موجودة في حزم يمكن تحميلها عند اللزوم وسوف نتحدث عنها لاحقا.
وأحد أهم ميزات R هو سهولة تعريف المستخدم لدواله الخاصة والتي سنتحدث عنها لاحقا. الدوال تعتبر أيضا أشياء ونمط الدالة هو function . أحينا الدوال لها تأثيرات جانبية *side-effects* فمثلا إستدعاء دالة قد يسبب رسم بياني وفي كثير من الأحيان يكون التأثير الجانبي أهم من القيمة المرجعة.

فضاء العمل و دليل العمل

Workspace and working directories

خلال جلسة في R يتم توليد عدد من الأشياء. فمثلا يمكن أن نولد متجهات و إطارات بيانات ودوال خلال هذه الجلسة هذه الأشياء تخزن في منطقة من ذاكرة الحاسب يطلق عليها فضاء عمل *workspace*. إذا أردنا حفظ الأشياء لإستخدام لاحق فإننا نوعز لـ R لكتابتهم في ملف في دليل العمل *working directory* الحالي. لاحظ الفرق الأشياء التي في الذاكرة مؤقتة تفقد بعد الخروج من النظام بينما الملفات دائمة حيث تخزن على القرص الصلب. إدارة الأشياء والملفات هي جزء مهم في إستخدام R بشكل فعال.

المتجهات والتخصيص

Vectors and assignment

يمكن إحداث المتجهات في محث التحكم *command prompt* بإستخدام دالة الربط *concatenation function* c(…) والتي لها البناء اللغوي *syntax* التالي:

c(*object1,object2,…*)

هذه الدالة تأخذ الحجج من نفس النمط أو الصنف وتعيد متجها يحوي هذة القيم

```
> c(1,2,3)
[1] 1 2 3
> c("Ali","Bet","Cat")
[1] "Ali" "Bet" "Cat"
```

لكي نستطيع إستخدام المتجهات يجب إعطائها معرف (طبعا نحن لانريد ان نكتب المتجه من جديد كلما نريد إستخدامه) ولكي نعطي إسم معرف للمتجه نستخدم عامل التخصيص *assignment operator* <-

```
name  <-  expression
```

name الآن يشير لشيئ قيمته هي نتيجة تقييم العبارة *expression*

```
> numbers <- c(1,2,3)
> people <- c("Ali","Bet","Cat")
> numbers
[1] 1 2 3
> people
[1] "Ali" "Bet" "Cat"
```

الرمز # في أول السطر يخبر R أن مايتبع هو تعقيب أو تعليق *comment*.
نستطيع إجراء عمليات بسيطة على المتجهات كالتالي:

```
> c(1,2,3)+c(4,5,6)
[1] 5 7 9
> numbers + numbers
[1] 2 4 6
> numbers - c(8,7.5,-2)
[1] -7.0 -5.5 5.0
> c(1,2,4)*c(1,3,3)
[1] 1 6 12
> c(12,12,12)/numbers
[1] 12 6 4
```

لاحظ في المثال السابق أن عمليتي الضرب والقسمة أجريت عنصر بعنصر.
ملاحظة: نستطيع إستدعاء الأوامر التي سبق أن أدخلناها بإستخدام زر السهم أعلى ↑ .
نتيجة أي عملية حسابية يمكن أن يعطى لها معرف لكي يمكن إستخداما لاحقاً

```
> calc1 <- numbers + c(8,7.5,-2)
> calc2 <- calc1 * calc1
> calc1
[1] 9.0 9.5 1.0
> calc2
[1] 81.00 90.25 1.00
> calc1 <- calc1 + calc2
> calc1
[1] 90.00 99.75 2.00
> calc2
[1] 81.00 90.25 1.00
```

إذا حاولنا جمع متجهين لها أطوال مختلفة فإن R تستخدم قاعدة إعادة إستخدام أو قاعدة التدوير *recycling rule* حيث يتم تكرار المتجة الأصغر حتى تصبح الأطوال واحدة

```
> small <- c(1,2)
> large <- c(0,0,0,0,0,0)
> large + small
[1] 1 2 1 2 1 2
```

ملاحظة: لمسح الأشياء من فضاء العمل نستخدم

```
> rm(list=objects())
```

مثال بسيط:

أوزان 5 خراف بالكيلو جرام هي **84.5, 72.6, 75.7, 94.8, 71.3** سوف نضع هذه القيم في متجه ونجري عليها بعض العمليات

```
> weight <- c(84.5, 72.6, 75.7, 94.8, 71.3)
> weight
[1] 84.5 72.6 75.7 94.8 71.3
> total <- sum(weight)
> numobs <- length(weight)
> meanweight <- total/numobs
> meanweight
[1] 79.78
...
> mean(weight)
[1] 79.78
```

أطر البيانات Data Frames:

إطار البيانات هو شيئ *object* في R والذي يمكن إعتباره كتمثيل لمجموعة بيانات *data set*. يتكون إطار البيانات من متغيرات (متجهات عامودي) لها نفس الطول و كل سطر يعود أو يتبع وحدة تجربة *experimental unit*. البناء اللغوي *Syntax* لتكوين إطار بيانات هو

```
name <- data.frame(variable1, variable2,
...)
```

وأي متغير في إطار بيانات يمكن الوصول إليه أو إستخدامه بإستعمال الرمز $ كالتالي:

```
name$variable
```

بعد تكوين إطار بيانات نستطيع مشاهدتة وتحريره في شكل صفحة نشر *spreadsheet* مستخدمين الأمر fix(...) (أو بشكل مكافئ data.entry(...)) ويمكن إضافة متغيرات جديدة لإطار البيانات عن طريق التخصيص.

عودة للمثال السابق:

لنفترض بألإضافة للأوزان لدينا إرتفاع الخراف مقاسة بالسنتيمتر كالتالي:
75.4 84.9 77.2 71.8 86.5 سوف ندخل هذه البيانات في متغير وكذلك سوف نكون تركيبة تربط بين الوزن والإرتفاع لنفس الوحدة التجريبية (هنا خروف) ويتم هذا عن طريق إضافة كل متغير إلى إطار بيانات وسوف نسمي هذا الإطار sheep ثم ننظر إليه بإستخدام fix(sheep)

```
> height <- c(86.5, 71.8, 77.2, 84.9, 75.4)
> sheep <- data.frame(weight, height)
> mean(sheep$height)
[1] 79.16
> fix(sheep)
```

لنفترض ان لدينا مقاس آخر لمتغير الطول بالسنتيمتر **130.4, 100.2,** **109.4, 140.6, 101.4** ونضيفه لإطار البيانات كالتالي:

```
> sheep$backlength <- c(130.4, 100.2, 109.4, 140.6,
                        101.4)
```

تمرين: أنظر لإطار البيانات لتشاهد ماذا حدث.

تحليل وصفي:

نستطيع الحصول على مجموعة من الإحصائآت الوصفية بإستخدام الدالة summary(...) وحجة الأمر ممكن أن تكون متغير واحد أو إطار بيانات. يتكون المخرج من جدول كالتالي:

```
> summary(sheep$weight)
Min.    1st Qu.   Median    Mean    3rd Qu.   Max.
71.30   72.60     75.70     79.78   84.50     94.80


> summary(sheep)
weight              height              backlength
Min. : 71.30        Min. : 71.80        Min. : 100.2
1st Qu. : 72.60     1st Qu. : 75.40     1st Qu. : 101.4
Median : 75.70      Median : 77.20      Median : 109.4
Mean : 79.78        Mean : 79.16        Mean : 116.4
3rd Qu. : 84.50     3rd Qu. : 84.90     3rd Qu. : 130.4
Max. : 94.80        Max. : 86.50        Max. : 140.6

> IQR(sheep$height)
[1] 9.5
> sd(sheep$backlength)
[1] 18.15269
```

نظام المساعدة في R:

توجد عدة طرق للحصول على المساعدة في R.

- إذا كان هناك أي تسائل عن دالة معينة نستخدم ؟ يتبعها إسم الدالة فتظهر نافذة المساعدة لتلك الدالة مثل:

```
> ?mean
> ?setwd
> ?t.test
```

- للحصول على مساعدة عامة نستخدم help.start() والتي تظهر نافذة تسمح بتصفح المعلومات التي تحتاج إليها.

إدارة الجلسة ووضوحية الرؤية Session management and visibility:

جميع الأشياء *objects* التي كونت خلال جلسة في R تخزن في فضاء عمل *workspace* في الذاكرة ونستطيع أن نشاهد الأشياء التي في فضاء العمل في الوقت الحالي بإستخدام الأمر ()object (لاحظ القوسين فهي مهمه جدا لكي يعمل الأمر) في المثال السابق نجد

```
> objects()
[1] "height" "meanweight" "numobs" "sheep" "total"
[6] "weight"
```

المعلومات في المتغيرات height و weight هي الآن مغلفة *encapsulated* في إطار البيانات sheep ولهذا سوف نقوم بتنظيف فضاء العمل وذلك بإزالة المتغيرات height و weight وبقية الأشياء التي لم نعد في حاجتها بإستخدام الدالة (...)rm وفي المثال السابق

```
> rm(height,weight,meanweight,numobs,total)
> objects()
[1] "sheep"
```

المتغيرات height و weight نستطيع الوصول إليها الآن فقط من خلال إطار البيانات sheep كالتالي

```
> weight
Error: Object "weight" not found
> sheep$weight
[1] 84.5 72.6 75.7 94.8 71.3
```

وميزة تغليف *encapsulation* المعلومات هي في كوننا نستطيع تكوين إطار بيانات آخر مثلا لنقل بقر cow لهو متغيرات height و weight بدون خلط أو غموض. في حال إرادتنا إستخدام المتغيرات في إطار البيانات sheep بشكل مكثف نجعلها مرئية من خط الأمر بإستخدام الأمر (...)attach وبعد الإنتهاء من هذه المتغيرات نعميها بالأمر ()detach (لاحظ الأقواس) بحيث تصبح المتغيرات المغلفة غير مرئية

```
> weight
Error: Object "weight" not found
> attach(sheep)
> weight
[1] 84.5 72.6 75.7 94.8 71.3
```

```
> detach()
> weight
Error: Object "weight" not found
```

نستطيع تخزين فضاء العمل الحالي في ملف متى شئنا ومن الممارسات الجيدة تكوين مجلد *folder* لأي مشروع جديد نقوم به. سوف نكون مجلد أو دليل عمل *Working directory* كالتالي:

- كون مجلد وسمه مثلا orCourse في :C (او اي قرص آخر)
- أجعله دليل عمل

```
> setwd("C:/orCourse")
```

وأمر تخزين فضاء العمل هو save.image(…) وفي حال عدم تحديد إسم للملف فإن البرنامج يعطيه الإسم Rdata. (لاحظ النقطة قبل الإسم). وطبعا من الأفضل إستخدام إسم يدل على طبيعة المشروع مثل

- تخزين فضاء العمل الحالي

```
> save.image("example1.Rdata")
```

- سرد الملف الموجود في دليل العمل

```
> dir()
```

الأمر dir() يعطي سرد لجميع الملفات في دليل العمل الحالي سوف تلاحظ أن مجلدك orCourse يحوي الآن ملف example1.Rdata هذا الملف يحوي أشياء objects من الجلسة الحالية أي إطار البيانات sheep من المهم جدا التفريق بين أشياء objects (والتي هي موجودة في فضاء العمل والتي نسردها بالأمر object()) وفضاء العمل (والذي يمكن أن يخزن في ملف محتوى مجلد والذي يسرد بالأمر dir()).

إستيراد بيانات Importing data:

عمليا نحتاج إلى إستيراد المعلومات من مجاميع بيانات *data sets* كبيرة جدا والتي هي مخزنة بشكل إلكتروني. سوف نستخدم بيانات مخزنة على شكل نص *text*. المتغيرات موجودة في أعمدة وأول سطر في العمود يحوي إسم المتغير. الملف sheep.dat يحوي أوزان وأطوال 100 خروف تم إختيارها عشوائيا. سوف نقوم بنسخ هذا الملف إلى دليل العمل orCourse. المعلومات تقرأ في R بإستخدام الدالة read.table(...). هذه الدالة ترجع إطار بيانات.

- استرجع البيانات من مجلد عام

```
C:\Documents and Settings\amb\Desktop
copy C:\Documents and Settings\...\sheep.dat
```

- قم بلصق الملف في المجلد

```
paste C:\orCourse\
```

- إقرأ البيانات في R

```
> sheep2 <- read.table("sheep.dat", header=TRUE)
```

إستخدام header = TRUE يعطينا إطار بيانات يكون فيه السطر الأول من ملف البيانات مستخدما كمعرف لأعمدة مجموع البيانات. إذا أغفلنا header = TRUE فسوف يعتبر السطر الأول كأحد المشاهدات المعطاة.

إختبار فرضيات A hypothesis test:

سوف نختبر الفرضية أن متوسط مجتمع الخراف $\mu = 80\,kg$ أي

$H_0 : \mu = 80,$

$H_1 : \mu \neq 80.$

سوف نستخدم مستوى معنوية 5% أي $\alpha = 0.05$ وبفرض أن أوزان الخراف لها توزيع طبيعي بتباين غير معروف لهذا نستخدم إختبار ***t*** (بذيلين) بإستخدام الدالة t.test(...) كالتالي:

```
> attach(sheep2) # to make variables accessible
> t.test(weight, mu=80)
One Sample t-test
data: weight
t = 2.1486, df = 99, p-value = 0.03411
alternative hypothesis: true mean is not equal to 80
95 percent confidence interval:
```

```
80.21048 85.29312
sample estimates:
mean of x
82.7518
```

للإختبار أي من الفرضيات البديلة (ذيل واحد)

$H_1 : \mu > 80,$

$H_1 : \mu < 80.$

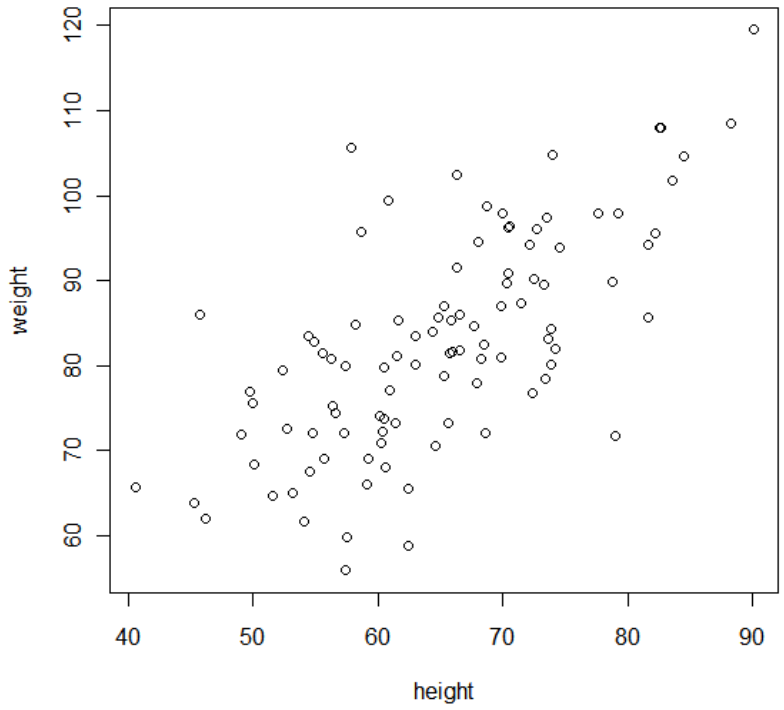
نستخدم

```
> t.test(weight, mu=80, alternative="greater")
> t.test(height, mu=66, alternative="less")
```

نموذج خطي A linear model:

سوف نستخدم الإنحدار الخطي لإيجاد علاقة بين الوزن (صعب القياس) والطول (سهل القياس) ولكن اولا نرسم المتغيرين في رسم أو مخطط إنتشار بإستخدام الدالة plot(…) كالتالي:

```
> plot(height,weight)
```



لاحظ أن متغير محور x يعطى في الدالة أولا.

لتطبيق نماذج خطية نستخدم الدالة lm(...) (لاحظ أن هذه الدالة مرنة جدا وتستخدم هنا في أبسط أشكالها ولها إستخدامات اخرى نتطرق لها في حينها) لتطبيق إنحدار خطي بسيط للوزن على الطول

```
> lmres <- lm(weight ~ height)
```

نلاحظ شيئين:

- الحجة weight ~ height تسمى صيغة نموذج *model formula*. الرمز ~ يعني " موصوف بـ " *described by* فالأمر هنا يطلب نموذج خطي والذي فيه الوزن يوصف بالطول سوف نتكلم أكثر عن صيغ النموذج لاحقا.
- بعد إدخال الدالة بحججها والضغط على إدخال لم يحدث شيئ فقط ظهر محفذ الأوامر > ولم تخرج نتائج؟ ذلك لأن R تعمل بوضع جميع المعلومات في شيئ *object* يعاد بواسطة الدالة lm(...) وهذا يعرف بإسم شيئ النموذج *model object* وفي هذا المثال فإننا قمنا بتخزين المعلومات في شيئ نموذج يسمى lmres والذي يمكن مشاهدتة وأستخدامه بإستخدام دوال مستخلصة *extractor functions*.

أسهل طريقة لإستخلاص معلومة هي بطباعة معرف (إسم) شيئ نموذج وفي هذا المثال lmres كما يمكننا إستخدام الدالة summary(...) لإعطاء تفصيلات أكثر أو الدالة abline(...) لتوليد رسم للخط المطبق.

```
> lmres

Call:
lm(formula = weight ~ height)

Coefficients:
(Intercept)       height
    26.0319       0.8724

> summary(lmres)

Call:
lm(formula = weight ~ height)

Residuals:
     Min       1Q   Median       3Q      Max
-23.1146  -6.1125  -0.1758   5.6578  29.1479

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 26.03185    6.05581   4.299 4.06e-05 ***
height       0.87239    0.09204   9.479 1.64e-15 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

Residual standard error: 9.298 on 98 degrees of freedom
Multiple R-squared: 0.4783,     Adjusted R-squared: 0.473
F-statistic: 89.84 on 1 and 98 DF,  p-value: 1.64e-15
```

الخروج والعودة لفضاء عمل مخزن Quitting and returning to a saved workspace:

قبل إنهاء جلسة من الأفضل إزالة أي أشياء *objects* غير أساسية ثم تخزين فضاء العمل في دليل العمل الحالي

```
> save.image("sheep2.Rdata")
> quit()
```

لاحظ عند الخروج فإن R يسأل فيما إذا كان يراد تخزين فضاء العمل ولو وافقنا فإنه سوف يعيد تخزين فضاء العمل في ملف *Rdata*. الإفتراضي ولهذا عند طلب التخزين نرفض تخزينه مرة أخري (هذا إجراء إختياري في R).
لكي نعود لفضاء العمل الذي تم تخزينه نعيد تشغيل R ونعيد تعريف دليل العمل ثم نستخدم الدالة load(…) لتحميل فضاء العمل

```
> setwd("C:/orCourse ")
> load("sheep2.Rdata")
> objects()
```

نلاحظ وجود جميع الأشياء من الجلسة السابقة.

خواص الأشياء Properties of objects:

الدالة length(…) تعطي طول المتجه فمثلا

```
> length(sheep2)
[1] 2
> length(plot)
[1] 1
```

طول إطار بيانات هو عدد المتغيرات التي يحويها الإطار. يمكن إيجاد طول دالة ولكن ليس له معنى. الدالة mode(…) تعطي طور الشيئ بينما الدالة attributes(…) تعطي الصفات غير الجوهرية

```
> mode(sheep2)
[1] "list"
> attributes(sheep2)
$names
[1] "weight" "height"
$class
[1] "data.frame"
$row.names
```

```
[1] "1" "2" "3" # ... etc to "100"
> mode(plot)
[1] "function"
> attributes(plot)
NULL
> mode(lmres)
[1] "list"
> attributes(lmres)
$names
[1] "coefficients" "residuals" "effects" "rank"
[5] "fitted.values" "assign" "qr" "df.residual"
[9] "xlevels" "call" "terms" "model"
$class
[1] "lm"
```

لاحظ برغم أن كلا منهما من طور *list* ألا أن إطارات البيانات من فئة *data.frame* بينما شيئ النموذج الخطي فهو من فئة *lm*. وهذا يؤثر على الطريقة التي يظهر بها على الشاشة عند طباعة المعرف

```
>  sheep2
    weight height
1    75.19  56.34
2   101.80  83.57
3    74.51  56.58
.
.

> lmres

Call:
lm(formula = weight ~ height)

Coefficients:
(Intercept)       height
    26.0319       0.8724
```

لقد تمت طباعة إطار البيانات كليا بينما اعطي مخرجات قليلة لشيئ نموذج خطي.
نستطيع أن نجبر R لكي يعامل كليهما كقائمة list بإستخدام الدالة unclass(…)
ولنحاول التالي:

```
> unclass(sheep2)
```

```
$weight
  [1]  75.19 101.80  74.51  81.42  83.20  80.84  81.67  90.82 102.51 104.87  96.03
 [12]  86.97  87.08  83.92  69.00  67.57  81.89  98.79  80.74  61.61  84.64  72.29
 [23]  82.41  80.20  73.82  64.95  70.64  94.60  72.08  65.60  97.95  61.97  84.81
 [34]  80.21  73.18 108.50  79.46  76.84  74.10  78.72  59.80  76.92  85.28  96.27
 [45]  82.77 108.00  97.90  77.99  91.60  97.37  78.40 107.93  89.74  83.57  65.76
 [56]  90.13  73.25  77.18  95.66 105.70  85.66  94.22 104.60  72.03  65.99  81.16
 [67]  93.84  86.07  95.56  89.84  55.90  85.40  89.59  72.07  64.76  68.09  86.05
 [78]  81.95  75.62  99.48  84.30  81.49  79.84  72.53 119.50  83.53  70.92  58.87
 [89]  96.47  63.89  87.33  81.01  97.90  79.97  69.10  68.37  71.81  85.68  94.27
[100]  71.90

$height
  [1] 56.34 83.57 56.58 55.60 73.59 56.24 65.93 70.42 66.28 73.99 72.76 69.87 65.33
 [14] 64.40 59.20 54.59 66.53 68.67 68.27 54.06 67.66 60.38 68.54 62.96 60.48 53.20
 [27] 64.56 68.01 54.76 62.47 77.62 46.20 58.23 73.84 65.58 88.21 52.42 72.40 60.17
 [40] 65.30 57.55 49.76 65.82 70.41 54.89 82.67 70.00 67.91 66.26 73.51 73.42 82.56
 [53] 70.27 63.01 40.57 72.50 61.38 60.98 58.64 57.91 81.64 72.12 84.47 57.33 59.14
 [66] 61.51 74.59 45.81 82.23 78.81 57.40 61.60 73.24 68.65 51.60 60.55 66.49 74.25
 [79] 50.03 60.82 73.89 65.74 60.50 52.72 90.05 54.40 60.25 62.46 70.50 45.35 71.46
 [92] 69.82 79.17 57.46 55.67 50.15 78.97 64.84 81.64 49.11

attr(,"row.names")
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
 [21]  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
 [41]  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
 [61]  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
 [81]  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
>
```

> unclass(lmres)

```
$coefficients
(Intercept)      height
 26.0318530   0.8723918

$residuals
          1           2           3           4           5           6
 0.007594467   2.862366459  -0.881779559   6.883164379  -7.031163636   5.744833644
          7           8           9          10          11          12
-1.878642646   3.354318288  18.656020233  14.289879655   6.522921537  -0.015866236
         13          14          15          16          17          18
 4.054792418   1.706116768  -8.677446007  -6.085719929  -2.182077711  12.851003893
         19          20          21          22          23          24
-4.850039398 -11.583352289  -0.417880416  -6.416868300  -3.415585177  -0.757639078
         25          26          27          28          29          30
-4.974107478  -7.493095363 -11.713465916   9.236782464  -1.724026530 -14.930167108
         31          32          33          34          35          36
 4.203097515  -4.366352945   7.978774014 -10.239261579 -10.063305525   5.514468628
         37          38          39          40          41          42
 7.697370221 -12.353017424  -4.423666028  -4.279035829 -16.437999580   7.477932340
         43          44          45          46          47          48
 1.827320449   8.813042206   8.852562539   9.847519056  10.800722833  -7.285978359
         49          50          51          52          53          54
 7.763468068   7.208627706 -11.682857034   9.873482151   2.405177054   2.568741334
         55          56          57          58          59          60
 4.335212743   0.849743398  -6.329260075  -2.050303365  18.471093386  29.147939381
         61          62          63          64          65          66
-11.593917417   5.271252272   4.877213863  -4.016073390 -11.635102501   1.467328995
         67          68          69          70          71          72
 2.736444590  20.073879847  -2.208628563  -4.945048696 -20.207140814   5.628813735
         73          74          75          76          77          78
-0.335826515 -13.851548272  -6.287268524 -10.765174902   2.012817960  -8.856942206
         79          80          81          82          83          84
 5.942386561  20.389279319  -6.192881168  -1.892888209   1.028444687   0.505652689
```

```
          85           86           87           88           89           90
14.909267764 10.040034508 -7.673457370 -21.651443190  8.934526946 -1.704819937
          91           92           93           94           95           96
-1.042969157 -5.932246647  2.800890265  3.810515680 -5.497903045 -1.412300452
          97           98           99          100
-23.114631380  3.082264387 -2.983917417  3.024986993
```

$effects

```
 (Intercept)        height
-827.5180000 -88.1319145  -0.7006440   7.0875994  -7.2544388   5.9340527
  -1.9198022   3.2064095  18.6065394  14.0570945   6.3193795  -0.1506989
   4.0278978   1.7013328  -8.5586006  -5.8572723  -2.2375022  12.7447011
  -4.9468322 -11.3423040  -0.5001706  -6.3260773  -3.5187972  -0.7281873
  -4.8856939  -7.2316007 -11.7220539   9.1461711  -1.4996206 -14.8890656
   3.8840095  -3.9384340   8.1206810 -10.4684805 -10.0961439   4.9436046
   7.9774093 -12.5480005  -4.3278823  -4.3052172 -16.2799256   7.8212127
   1.7887761   8.6653711   9.0738777   9.4083679  10.6627995  -7.3742122
   7.7144628   6.9872545 -11.9020905   9.4369462   2.2608345   2.5970044
   4.8969843   0.6523828  -6.2622439  -1.9737773  18.6032527  29.2974544
 -12.0085805   5.0829261   4.3952679  -3.8527690 -11.5148306   1.5312544
   2.4893945  20.5110710  -2.6373188  -5.2924288 -20.0455006   5.6905994
  -0.5507805 -13.9573755  -5.9877340 -10.6784256   1.9583444  -9.0959088
   6.2792477  20.4696094  -6.4232888  -1.9295306   1.1163827   0.7785593
  14.2946579  10.2729994  -7.5795756 -21.6101039   8.7847161  -1.2566924
  -1.2156039  -6.0658905   2.4449512   3.9707294  -5.2951323  -1.0782923
 -23.4658155   3.0670194  -3.3985805   3.3837210
```

$rank

```
[1] 2
```

$fitted.values

```
       1        2        3        4        5        6        7        8
 75.18241 98.93763 75.39178 74.53684 90.23116 75.09517 83.54864 87.46568
       9       10       11       12       13       14       15       16
 83.85398 90.58012 89.50708 86.98587 83.02521 82.21388 77.67745 73.65572
      17       18       19       20       21       22       23       24
 84.07208 85.93900 85.59004 73.19335 85.05788 78.70687 85.82559 80.95764
      25       26       27       28       29       30       31       32
 78.79411 72.44310 82.35347 85.36322 73.80403 80.53017 93.74690 66.33635
      33       34       35       36       37       38       39       40
 76.83123 90.44926 83.24331 102.98553 71.76263 89.19302 78.52367 82.99904
      41       42       43       44       45       46       47       48
 76.23800 69.44207 83.45268 87.45696 73.91744 98.15248 87.09928 85.27598
      49       50       51       52       53       54       55       56
 83.83653 90.16137 90.08286 98.05652 87.33482 81.00126 61.42479 89.28026
      57       58       59       60       61       62       63       64
 79.57926 79.23030 77.18891 76.55206 97.25392 88.94875 99.72279 76.04607
      65       66       67       68       69       70       71       72
 77.62510 79.69267 91.10356 65.99612 97.76863 94.78505 76.10714 79.77119
      73       74       75       76       77       78       79       80
 89.92583 85.92155 71.04727 78.85517 84.03718 90.80694 69.67761 79.09072
```

```
      81        82        83        84        85        86        87        88
 90.49288  83.38289  78.81156  72.02435 104.59073  73.48997  78.59346  80.52144
      89        90        91        92        93        94        95        96
 87.53547  65.59482  88.37297  86.94225  95.09911  76.15948  74.59790  69.78230
      97        98        99       100
 94.92463  82.59774  97.25392  68.87501

$assign
[1] 0 1

$qr
$qr
    (Intercept)         height
1         -10.0 -6.501660e+02
2           0.1 -1.010233e+02
3           0.1 -7.570349e-02
4           0.1 -8.540422e-02
5           0.1  9.267347e-02
6           0.1 -7.906905e-02
7           0.1  1.684939e-02
8           0.1  6.129458e-02
9           0.1  2.031394e-02
10          0.1  9.663295e-02
11          0.1  8.445755e-02
12          0.1  5.585029e-02
13          0.1  1.091017e-02
14          0.1  1.704377e-03
15          0.1 -4.976889e-02
16          0.1 -9.540191e-02
17          0.1  2.278862e-02
18          0.1  4.397184e-02
19          0.1  4.001236e-02
20          0.1 -1.006482e-01
21          0.1  3.397415e-02
22          0.1 -3.808841e-02
23          0.1  4.268501e-02
24          0.1 -1.254976e-02
25          0.1 -3.709854e-02
26          0.1 -1.091611e-01
27          0.1  3.288170e-03
28          0.1  3.743870e-02
29          0.1 -9.371913e-02
30          0.1 -1.740012e-02
31          0.1  1.325652e-01
32          0.1 -1.784520e-01
33          0.1 -5.937063e-02
34          0.1  9.514815e-02
35          0.1  1.338485e-02
36          0.1  2.373925e-01
37          0.1 -1.168821e-01
38          0.1  8.089401e-02
39          0.1 -4.016714e-02
40          0.1  1.061321e-02
41          0.1 -6.610175e-02
42          0.1 -1.432127e-01
43          0.1  1.576054e-02
44          0.1  6.119559e-02
45          0.1 -9.243230e-02
46          0.1  1.825537e-01
47          0.1  5.713712e-02
48          0.1  3.644883e-02
49          0.1  2.011597e-02
50          0.1  9.188157e-02
51          0.1  9.099069e-02
52          0.1  1.814648e-01
53          0.1  5.980977e-02
54          0.1 -1.205482e-02
```

```
55         0.1 -2.341818e-01
56         0.1  8.188388e-02
57         0.1 -2.818971e-02
58         0.1 -3.214919e-02
59         0.1 -5.531216e-02
60         0.1 -6.253821e-02
61         0.1  1.723580e-01
62         0.1  7.812238e-02
63         0.1  2.003714e-01
64         0.1 -6.827946e-02
65         0.1 -5.036281e-02
66         0.1 -2.690288e-02
67         0.1  1.025722e-01
68         0.1 -1.823125e-01
69         0.1  1.781983e-01
70         0.1  1.443447e-01
71         0.1 -6.758655e-02
72         0.1 -2.601200e-02
73         0.1  8.920892e-02
74         0.1  4.377387e-02
75         0.1 -1.249990e-01
76         0.1 -3.640564e-02
77         0.1  2.239267e-02
78         0.1  9.920662e-02
79         0.1 -1.405400e-01
80         0.1 -3.373299e-02
81         0.1  9.564308e-02
82         0.1  1.496864e-02
83         0.1 -3.690057e-02
84         0.1 -1.139125e-01
85         0.1  2.556061e-01
86         0.1 -9.728267e-02
87         0.1 -3.937525e-02
88         0.1 -1.749911e-02
89         0.1  6.208647e-02
90         0.1 -1.868659e-01
91         0.1  7.158923e-02
92         0.1  5.535536e-02
93         0.1  1.479082e-01
94         0.1 -6.699263e-02
95         0.1 -8.471131e-02
96         0.1 -1.393522e-01
97         0.1  1.459285e-01
98         0.1  6.059807e-03
99         0.1  1.723580e-01
100        0.1 -1.496468e-01
attr(,"assign")
[1] 0 1

$qraux
[1] 1.100000 1.191463

$pivot
[1] 1 2

$tol
[1] 1e-07

$rank
[1] 2
```

```
attr(,"class")
[1] "qr"

$df.residual
[1] 98

$xlevels
list()

$call
lm(formula = weight ~ height)

$terms
weight ~ height
attr(,"variables")
list(weight, height)
attr(,"factors")
       height
weight      0
height      1
attr(,"term.labels")
[1] "height"
attr(,"order")
[1] 1
attr(,"intercept")
[1] 1
attr(,"response")
[1] 1
attr(,".Environment")
<environment: R_GlobalEnv>
attr(,"predvars")
list(weight, height)
attr(,"dataClasses")
   weight    height
"numeric" "numeric"

$model
     weight height
1    75.19  56.34
2   101.80  83.57
3    74.51  56.58
4    81.42  55.60
```

5	83.20	73.59
6	80.84	56.24
7	81.67	65.93
8	90.82	70.42
9	102.51	66.28
10	104.87	73.99
11	96.03	72.76
12	86.97	69.87
13	87.08	65.33
14	83.92	64.40
15	69.00	59.20
16	67.57	54.59
17	81.89	66.53
18	98.79	68.67
19	80.74	68.27
20	61.61	54.06
21	84.64	67.66
22	72.29	60.38
23	82.41	68.54
24	80.20	62.96
25	73.82	60.48
26	64.95	53.20
27	70.64	64.56
28	94.60	68.01
29	72.08	54.76
30	65.60	62.47
31	97.95	77.62
32	61.97	46.20
33	84.81	58.23
34	80.21	73.84
35	73.18	65.58
36	108.50	88.21
37	79.46	52.42
38	76.84	72.40
39	74.10	60.17
40	78.72	65.30
41	59.80	57.55
42	76.92	49.76
43	85.28	65.82
44	96.27	70.41
45	82.77	54.89
46	108.00	82.67
47	97.90	70.00
48	77.99	67.91
49	91.60	66.26
50	97.37	73.51
51	78.40	73.42
52	107.93	82.56
53	89.74	70.27
54	83.57	63.01
55	65.76	40.57
56	90.13	72.50
57	73.25	61.38
58	77.18	60.98
59	95.66	58.64
60	105.70	57.91
61	85.66	81.64
62	94.22	72.12
63	104.60	84.47
64	72.03	57.33
65	65.99	59.14
66	81.16	61.51
67	93.84	74.59
68	86.07	45.81
69	95.56	82.23
70	89.84	78.81
71	55.90	57.40
72	85.40	61.60
73	89.59	73.24
74	72.07	68.65
75	64.76	51.60

```
76   68.09  60.55
77   86.05  66.49
78   81.95  74.25
79   75.62  50.03
80   99.48  60.82
81   84.30  73.89
82   81.49  65.74
83   79.84  60.50
84   72.53  52.72
85  119.50  90.05
86   83.53  54.40
87   70.92  60.25
88   58.87  62.46
89   96.47  70.50
90   63.89  45.35
91   87.33  71.46
92   81.01  69.82
93   97.90  79.17
94   79.97  57.46
95   69.10  55.67
96   68.37  50.15
97   71.81  78.97
98   85.68  64.84
99   94.27  81.64
100  71.90  49.11

>
```

لاحظ أن الجزء الأخير من `> unclass(lmres)` يشبه مخرجات sheep2 وذلك لأن lmres يحوي صفة ($model) والذي هو إطار بيانات مكون من متغيرات نحن نضعها في النموذج. نستطيع إيجاد فئة أي شيئ مستخدمين الدالة class(…)

```
> class(lmres$model)
[1] "data.frame"
```

وهذه ليست النهاية لأن الطور *mode* هو طريقة مريحة لإخفاء حقيقة معقدة. في الحقيقة R تعمل على نوع *type* الشيئ وفي بعض الأحيان يكون نوع أكثر تحديدا ودقة من طور *mode* ولكن في حالات اخرى فإن الخاصيتين متطابقة. نستطيع إيجاد نوع شيئ بإستخدام الدالة typeof(…) والطريقة التي تخزن بها تعطى بالدالة storage.mode(…) كالتالي:

```
> typeof(sheep2)
[1] "list"
> typeof(sheep2$weight)
[1] "double"
> z <- 1 + 1i
> typeof(z)
[1] "complex"
> typeof(plot)
[1] "closure"
> storage.mode(plot)
[1] "function"
```

بعض الدوال:

```
> class(lmres$model)
[1] "data.frame"
> names(sheep2)
[1] "weight" "height"
> attributes(lmres)
$names
 [1] "coefficients"  "residuals"     "effects"
"rank"          "fitted.values"
 [6] "assign"        "qr"            "df.residual"
"xlevels"       "call"
[11] "terms"         "model"

$class
[1] "lm"

> lmres$residuals
           1            2            3            4
5            6
  0.007594467   2.862366459  -0.881779559   6.883164379  -
7.031163636   5.744833644
           7            8            9           10
11           12
 -1.878642646   3.354318288  18.656020233  14.289879655
6.522921537  -0.015866236  ….
> lmres$effects
 (Intercept)      height
-827.5180000  -88.1319145   -0.7006440    7.0875994   -7.2544388
5.9340527

  -1.9198022    3.2064095   18.6065394   14.0570945    6.3193795
-0.1506989

   4.0278978    1.7013328   -8.5586006   -5.8572723   -2.2375022
12.7447011 …
> lmres$coefficients
(Intercept)      height
 26.0318530   0.8723918
```

الحجج المسماه والقيم الإفتراضية Named arguments and default values:

في التالي نحتاج إلى تحميل بعض الملفات من دليل العمل

```
> setwd("C:/orCourse")
> mkmsc <- read.table("marks.dat", header=TRUE)
> mk2nd <- read.table("marks2.dat", header=TRUE)
```

```
> attach(mk2nd)
```

في لغة R الدوال التالية

```
> plot(exam1, exam2)
> plot(exam2, exam1)
```

الاولى ترسم exam1 على محور x و exam 2 على محور y بينما الثانية تعكس الرسم وهذا هو الإختيار الإفتراضي *default* في R.
لتغيير الإختيار الإفتراضي نستخدم حجج مسماة *named arguments* كالتالي

```
> plot(y=exam2, x=exam1)
> plot(x=exam1, y=exam2)
```

وكلاهما يعطي الرسم نفسه. الحجج المسماه تفيد أكثر في عدم الإلتزام بالترتيب أو القيم الإفتراضية فمثلا

```
> t.test(exam1)
> t.test(exam1, mu=30)
> t.test(exam1, alternative="greater")
> t.test(exam1, mu=31.5, alternative="less")
> t.test(mu=30, exam1)
> t.test(x=exam1, mu=30)
```

لاحظ أن القيمة الإفتراضية لـ mu في t.test هي الصفر. و alternative = "two.sided"

الصفة names:

كما ذكرنا سابقا إطارات البيانات لها صفة *names* والتي تشير إلى أسماء المتغيرات في ذلك الإطار. ويمكن سرد الأسماء بإستخدام الدالة names(...)

```
> names(mkmsc)
[1] "courseA" "courseB" "courseC"
```

الغرض من العامل $ هو تمكيننا من الوصول مباشرة إلى جزء- شيئ *sub-object* مسرد بواسطة names(...) ويمكننا تغيير الأسماء والمحتويات للجزء - شيئ عن طريق الإسناد فلنفترض اننا نريد تغيير courseA و courseB و courseC بالأسماء التالية or221 و or241 و or342

```
> names(mkmsc) <- c("or221", "or241", "or342")
> names(mkmsc)
[1] "or221" "or241" "or342"
> mkmsc$or221
[1] 52 71 44 90 23 66
```

ونستطيع إضافة 2 لكل قيمة

```
> mkmsc$or221 <- mkmsc$or221 + 2
> mkmsc$or221
[1] 54 73 46 92 25 68
```

صفات غير جوهرية أخرى Other non-intrinsic attributes:

الصفات غير الجوهرية الأخرى يمكن تغييرها بالإسناد فمثلا في إطار بيانات نريد تغيير أسماء الأسطر الإفتراضية والتي هي غالبا 1 و 2 و 3 و الخ إلى أسماء ذات معنى

```
> row.names(mkmsc)
[1] "1" "2" "3" "4" "5" "6"
> row.names(mkmsc) <- c("Ali", "Badr", "Ahmad",
+"Bakur", "Saeed", +"Faris")
> mkmsc
      or221 or241 or342
Ali      54    67    71
Badr     73    80    84
Ahmad    46    55    55
Bakur    92    76    68
Saeed    25    49    52
Faris    68    61    61
>
```

ملاحظة: في R يكمل السطر الطويل إلى السطر التالي بوضع علامة + في بداية السطر التالي.

المتتابعات المنسقة Regular sequences:

المتتابعات المنسقة هي متتابعات من الأعداد أو الحروف والتي تتبع نمط او نسق ثابت وهي مفيدة في إختيار أجزاء من متجة وفي توليد قيم للمتغيرات الوصفية. هناك عدة طرق لتوليد متتابعات منها مولد متتابعة *sequence generator*:

```
> 1:10
[1] 1 2 3 4 5 6 7 8 9 10
> 10:1
[1] 10 9 8 7 6 5 4 3 2 1
> 2*1:10
[1] 2 4 6 8 10 12 14 16 18 20
> 1:10 + 1:20
[1] 2 4 6 8 10 12 14 16 18 20 12 14 16 18 20 22 24
26 28 30
> 1:10-1
[1] 0 1 2 3 4 5 6 7 8 9
> 1:(10-1)
```

```
[1] 1 2 3 4 5 6 7 8 9
```

لاحظ أن (:) له افضلية على العملية الحسابية.

الدالة seq تعطي مقدار اكبر من المرونة في توليد متتابعات ولها البناء التالي

seq(from, to, by, length, along)

الحجج from و to مفسرة ذاتيا الحجة by تدل على مقدار الزيادة و length عدد الوحدات في المتتابعة. لاحظ انه لاتستخدم جميع هذه الحجج معا وإلا سيسبب هذا في زيادة في التعيين وينتج عنه خطأ. الحجة along تسمح لمتجه يذكر والذي طوله هو الطول المطلوب للمتتابعة

```
> seq(1,10)
[1] 1 2 3 4 5 6 7 8 9 10
> seq(to=10, from=1)
[1] 1 2 3 4 5 6 7 8 9 10
> seq(1,10,by=0.5)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5
7.0 7.5 8.0
[16] 8.5 9.0 9.5 10.0
> seq(1,10,length=19)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5
7.0 7.5 8.0
[16] 8.5 9.0 9.5 10.0
> seq(1,10,length=19,by=0.25)
Error in seq.default(1, 10, length = 19, by = 0.25)
:
Too many arguments
> seq(1,by=2,length=6)
[1] 1 3 5 7 9 11
> seq(to=30,length=13)
[1] 18 19 20 21 22 23 24 25 26 27 28 29 30
> seq(to=30)
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25
[26] 26 27 28 29 30
```

الدالة rep تولد متتابعات بتكرارات معينة

```
> rep(1, times = 3)
[1] 1 1 1
> rep((1:3), each =5)
[1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
```

المتجهات المنطقية Logical vectors:

القيم المنطقية في R هي TRUE و FALSE
المقارنة تعمل بإستخدام == للمساواة و =! لعدم المساواة و < لأكبر من و =< لأكبرمن أو يساوي و > لأقل من و => لأقل من أو يساوي
العمال المنطقية & لـ AND و | لـ OR و ! لـ NOT

```
> tf1 <- c(TRUE,TRUE,FALSE,FALSE)
> tf2 <- c(TRUE,FALSE,TRUE,FALSE)
> tf1 & tf2
[1] TRUE FALSE FALSE FALSE
> tf1 | tf2
[1] TRUE TRUE TRUE FALSE
> (tf1 & !tf2) | (!tf1 & tf2)
[1] FALSE TRUE TRUE FALSE
```

تأشير متجه وإختيار مجموعة جزئية Indexing vectors and subset selection:

أحيانا نرغب في إستخدام جزء من البيانات أو مجموعة جزئية والتي تحقق شرط معين ولذلك تستخدم [ ] لكي تؤشر لجزء من المتجه ويعرف بعامل التذيل *subscripting operator* والمتجه مابين القوسين يعرف بإسم متجه التأشير *indexing vector* فإذا كان متجه التأشير له قيم صحيحة (غالبا متتابعة متسقة) فإن العناصر المؤشرة التابعة لقيم المتجة المؤشر يتم إختيارها وإذا وجدت علامه - في المقدمة فعندئذ كل العناصر ماعدى الذين تم تأشيرهم سيتم إختيارهم

```
> attach(mkmsc)
> or221
[1] 54 73 46 92 25 68
> or221[c(1,2,6)]
[1] 54 73 68
> or221[1:4]
[1] 54 73 46 92
> or221[-(1:4)]
[1] 25 68
> or221[seq(6,by=-2)]
[1] 68 92 73
```

```
> or221[rep((1:3),each=2)]
[1] 54 54 73 73 46 46
> row.names(mkmsc)[1:3]
[1] "Ali"   "Badr"  "Ahmad"
```

نلاحظ أننا نستطيع عمل التالي:

- إيجاد علامات الطلاب في or221 التي أعلى من 70
- علامات طلاب or221 والذين حصلوا على أقل من 65 في or241
- علامات الطلاب الذين مجموع علاماتهم الكلي أقل من 200

```
> or221[or221>70]
[1] 73 92
> or241[or221<65]
[1] 67 55 49
> or342[(or221+or241+or342)<200]
[1] 71 55 52 61
> or342[or221>50 & or241>50 & or342>50]
[1] 71 84 68 61
```

التالي يستخرج الأسماء بدل العلامات

```
> row.names(mkmsc)[or221>70]
[1] "Badr"  "Bakur"
> row.names(mkmsc)[or241<50 | or221<50 | or342<50]
[1] "Ahmad" "Saeed"
> names(mkmsc)[c(sum(or221), sum(or241),
sum(or342)) > 350]
[1] "or221" "or241" "or342"
```

متتابعات الحروف أو النصوص Character sequences:

لنفترض أنه لدينا أزواج من النقاط $\{(x_1,y_1),(x_2,y_2),\ldots,(x_5,y_5)\}$ ونريد تمثيلها في R لذلك نستخدم الدالة paste(…) والتي تجمع الحجج وتعيد حرف *character*. الحجة sep لتحديد مانريد إدخاله بين الحجج

```
> paste(c("x","y"), rep(1:5,each=2), sep="")
 [1] "x1" "y1" "x2" "y2" "x3" "y3" "x4" "y4" "x5"
"y5"
> str1 <- paste("(x", 1:5, sep="")
```

```
> str2 <- paste("y", 1:5, ")", sep="")
> label1 <- paste(str1,str2,sep=",")
> label1
[1] "(x1,y1)" "(x2,y2)" "(x3,y3)" "(x4,y4)"
"(x5,y5)"
> mode(label1)
[1] "character"
```

القوائم Lists:

القائمة هي مجموعة مرتبة من الأشياء. الأشياء الموجودة في قائمة يشار إليها كمكونات *components* وهذه المكونات قد تكون من أطوار مختلفة. تولد القوائم بإستخدام الدلة list(...) والقيمة المرجعة هي قائمة بمكونات من الحجج التي اعطيت للقائمة

```
> mklst <- list("BSc exam marks", c(10,6,2005),
mkmsc)
> mode(mklst)
[1] "list"
> length(mklst)
[1] 3
> mklst
[[1]]
[1] "BSc exam marks"

[[2]]
[1]   10    6 2005

[[3]]
      or221 or241 or342
Ali      54    67    71
Badr     73    80    84
Ahmad    46    55    55
Bakur    92    76    68
Saeed    25    49    52
Faris    68    61    61
```

نستطيع الآن إستخدام عامل التأشير [ ] للوصول إلى أجزاء من القائمة

```
> mklst[2]
```

```
[[1]]
[1]   10    6 2005
```

```
> mode(mklst[2])
[1] "list"
> length(mklst[2])
[1] 1
```

لاحظ أن نتيجة إستخدام التأشير على قائمة هو قائمة أخرى. إذا أردنا أن نضيف 4 إلى يوم فإننا لانستطيع أن نستخدم [ ] لأن mklst[2] ليست قيمة عددية

```
> mklst[2]+c(4,0,0)
Error in mklst[2] + c(4, 0, 0) : non-numeric
argument to binary operator
```

ولكي نتجنب هذا الخطأ نستخدم أقواس مضاغفة [[ ]] مثل

```
> mklst[[2]]
[1] 10 6 2005
> mode(mklst[[2]])
[1] "numeric"
> mklst[[2]] <- mklst[[2]] + c(4,0,0)
> mklst
```

في الحقيقة بما أن mklst[[2]] متجه عددي فإننا نستطيع إستخدام أي عنصر فيه مباشرة ونضيف 4 إليه

```
> mklst[[2]][1] <- mklst[[2]][1] + 4
```

لاحظ أن العامل [ ] إذا طبق على قائمة فإنه يعيد قائمة والعامل [[ ]] يعيد عنصر من القائمة وطور القيمة المعادة هو طور العنصر. العلامات label الإفتراضية لعناصر قائمة هي 1 و 2 و 3 و ... ولاتوضح ماتحويه القائمة ويكون علينا تذكر إسم كل عنصر في القائمة وهذا سهلا لقوائم تحوي عنصر أو عنصرين ولكن للقوائم التي تحوي عدد كبير من العناصر يكون من الصعب تذكر أسماء كل العناصر وترتيبها في القائمة لهذا من الإفضل إعطاء أسماء لعناصر القائمة مثلا كالتالي

```
> names(mklst) <- c("title", "date", "marks")
```

ويمكننا الآن الإشارة إلى عناصر القائمة إما بإستخدام العامل [[ ]] أو العامل $

```
> mklst[[2]]
[1] 18 6 2005
> mklst[["date"]]
[1] 18 6 2005
> mklst$date
[1] 18 6 2005
```

لاحظ بإستخدامنا للعامل $ نستطيع الإشارة إلى إسم مباشرة ولانحتاج لإستخدام "". كما يمكن إعطاء الأسماء عند تعريف القائمة مثل

```
> mklst <- list("title"="MSc exam marks",
+"date"=c(10,6,2005),
+ "marks"=mkmsc)
```

الرسومات البسيطة في R:

تمتلك R مقدرات رسم قوية ومرنة. سوف نستعرض هنا بعضا من هذه المقدرات.

بعض الرسومات الوصفية:

سوف نقوم برسم المدرج التكراري لأحد المتغيرات وذلك بإستخدام الدالة hist(...) كالتالي:

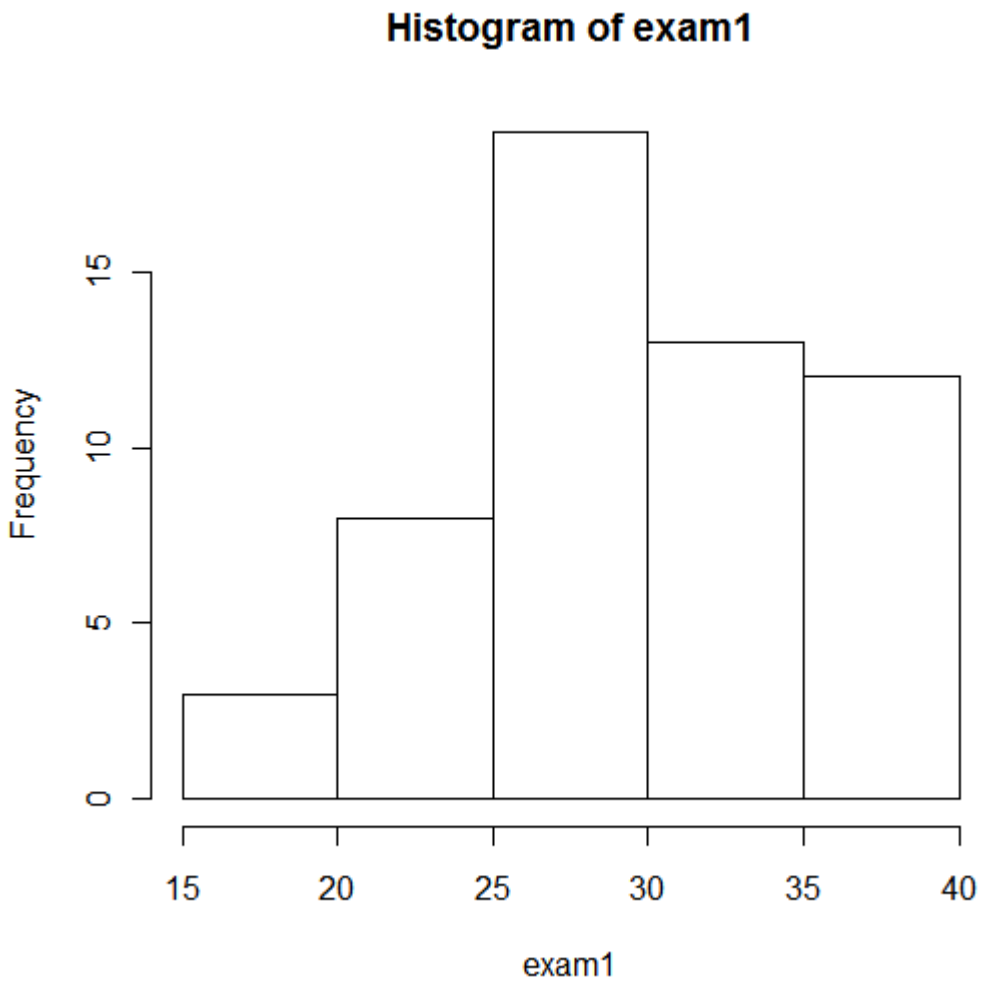
أولا نحدد دليل العمل ونقرأ ملف البيانات

```
> setwd("C:/orCourse")
> mk2nd <- read.table("marks2.dat", header=TRUE)
> attributes(mk2nd)
$names
[1] "candno" "exam1"  "exam2"  "exam3"

$class
[1] "data.frame"

$row.names
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
16 17 18 19 20 21 22
[23] 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40 41 42 43 44
[45] 45 46 47 48 49 50 51 52 53 54 55
```

```
> attach(mk2nd)
```

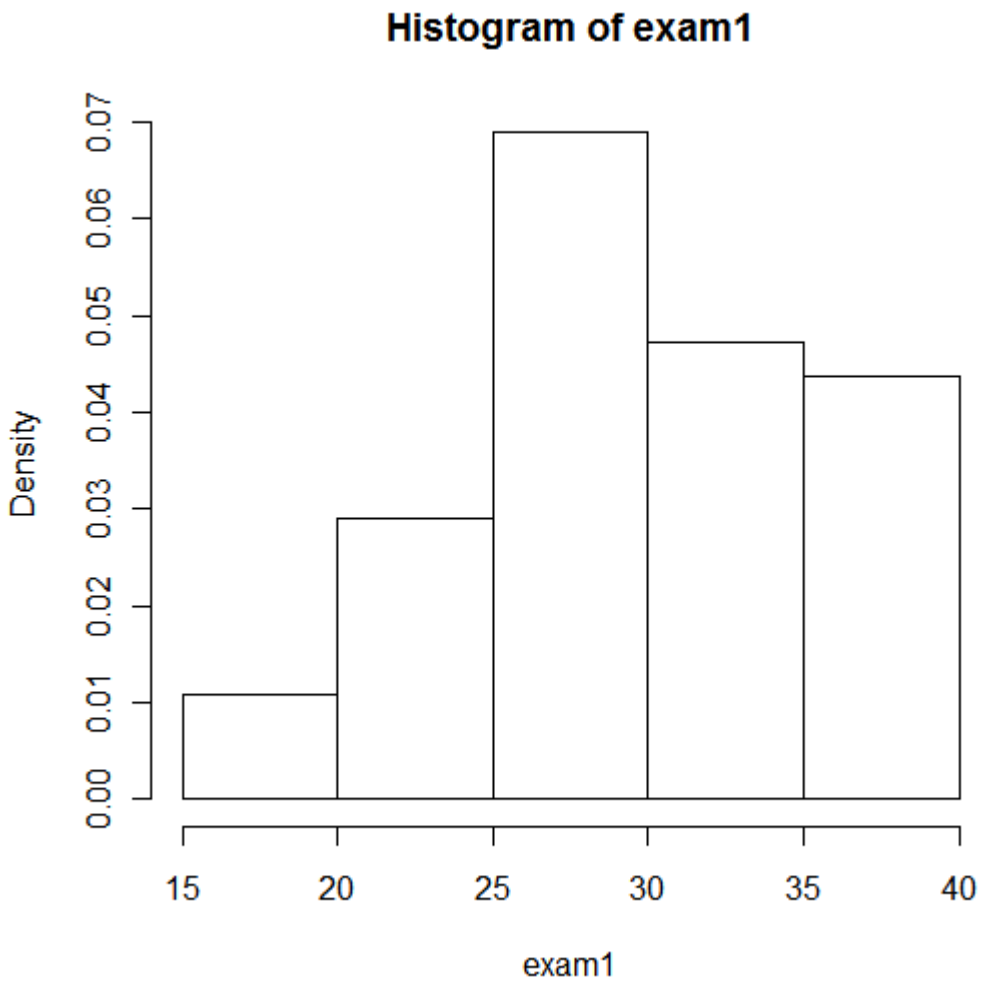
ونرسم المدرج التكراري للمتغير exam1

```
> hist(exam1)
```



لاحظ هنا تم رسم التكرارات. لرسم الإحتمالات ندخل

```
> hist(exam1, probability = TRUE)
```

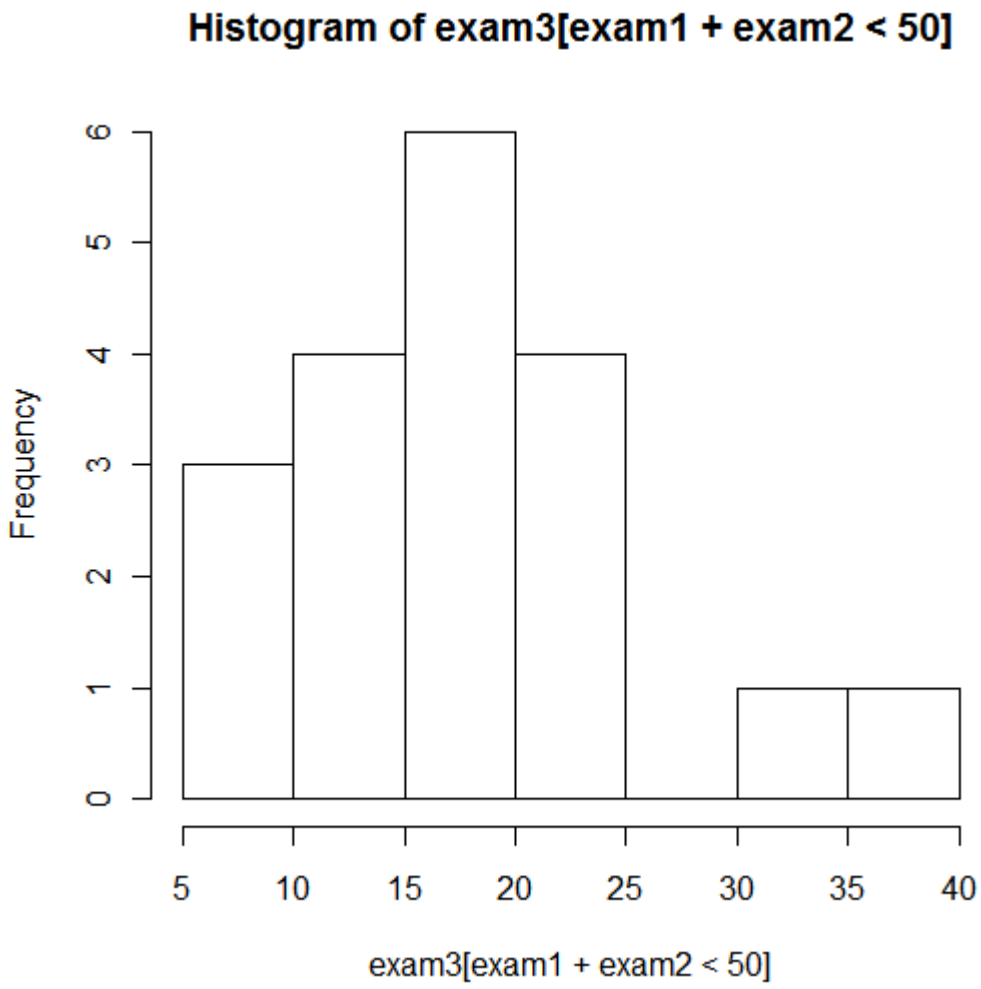


نريد فئآت أكثر مثلا 10 فئآت

```
> hist(exam1, nclass=10)
```

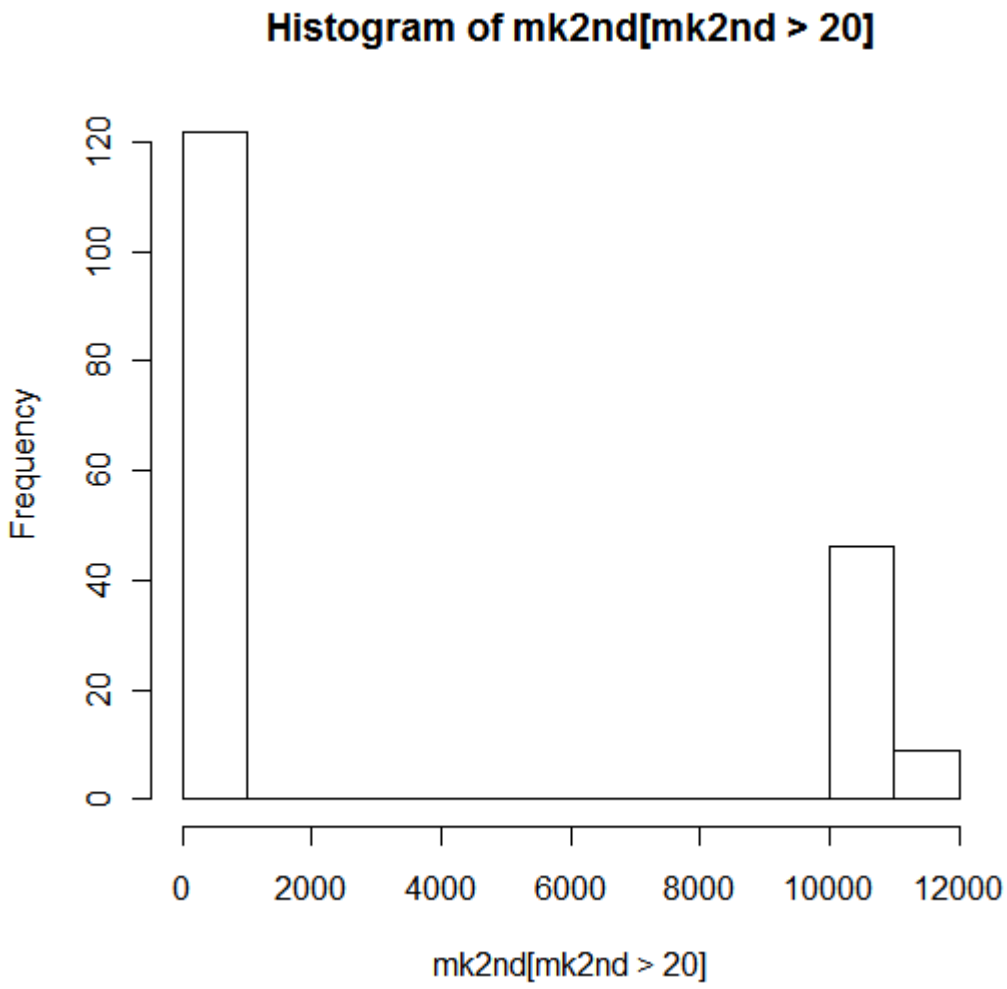
كما أننا نستطيع إختيار أجزاء كالتالي

```
> hist(exam3[exam1+exam2 <50])
```



أو

```
> hist(mk2nd[mk2nd>20])
```

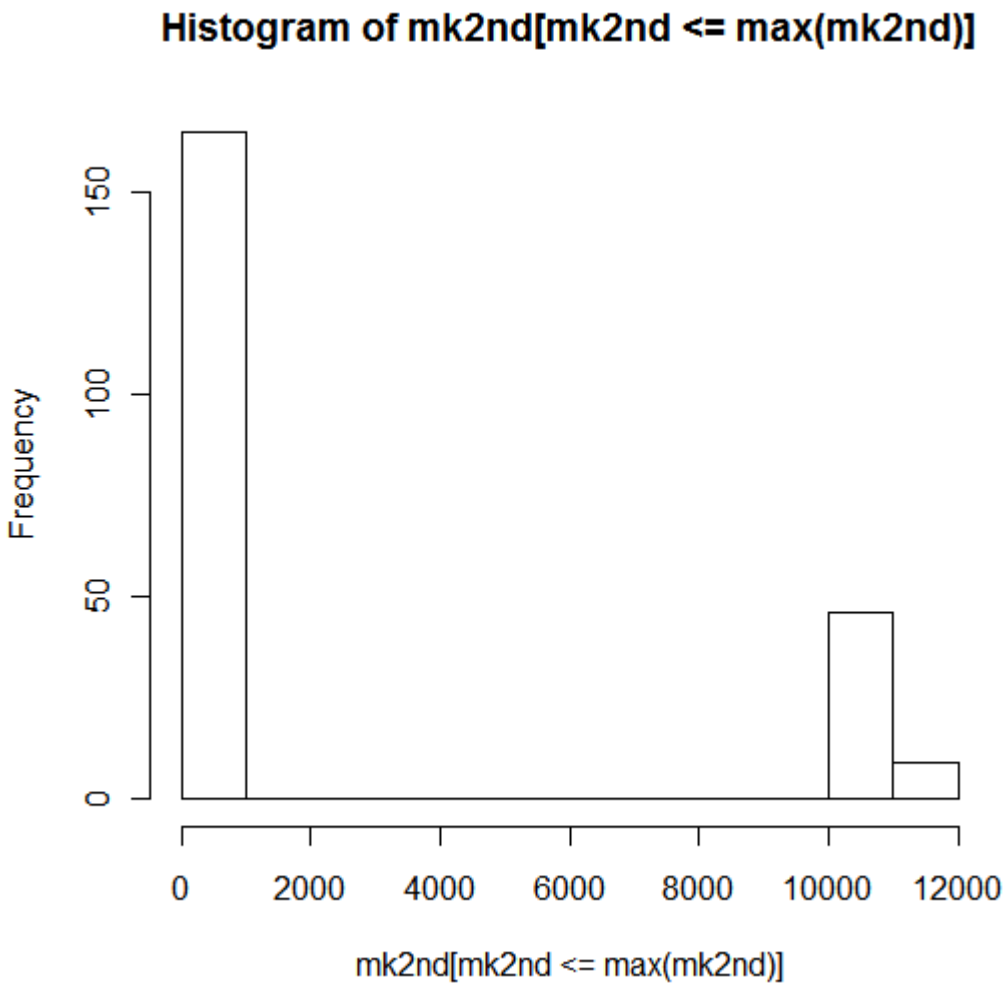


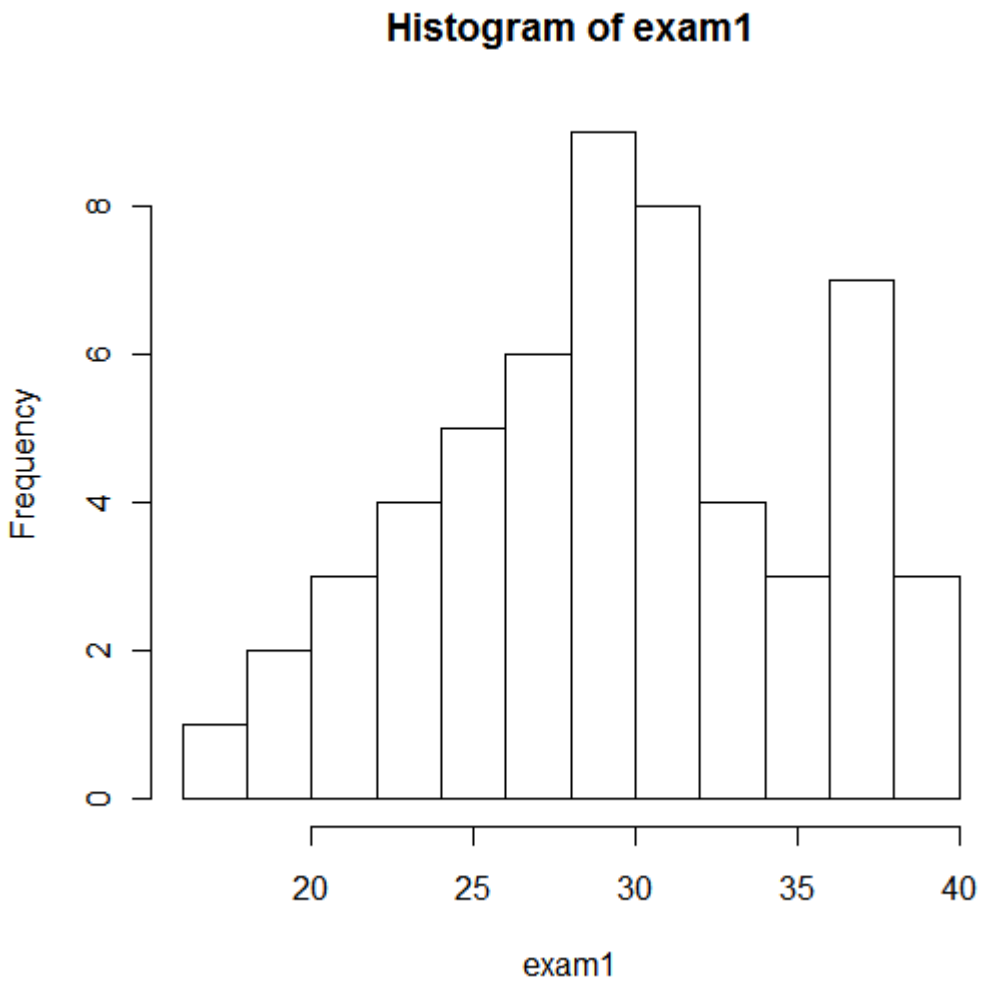
لاحظ أننا لانستطيع رسم إطار بيانات مباشرة

```
> hist(mk2nd)
Error in hist.default(mk2nd) : `x' must be numeric
```

ولكن نستطيع التالي

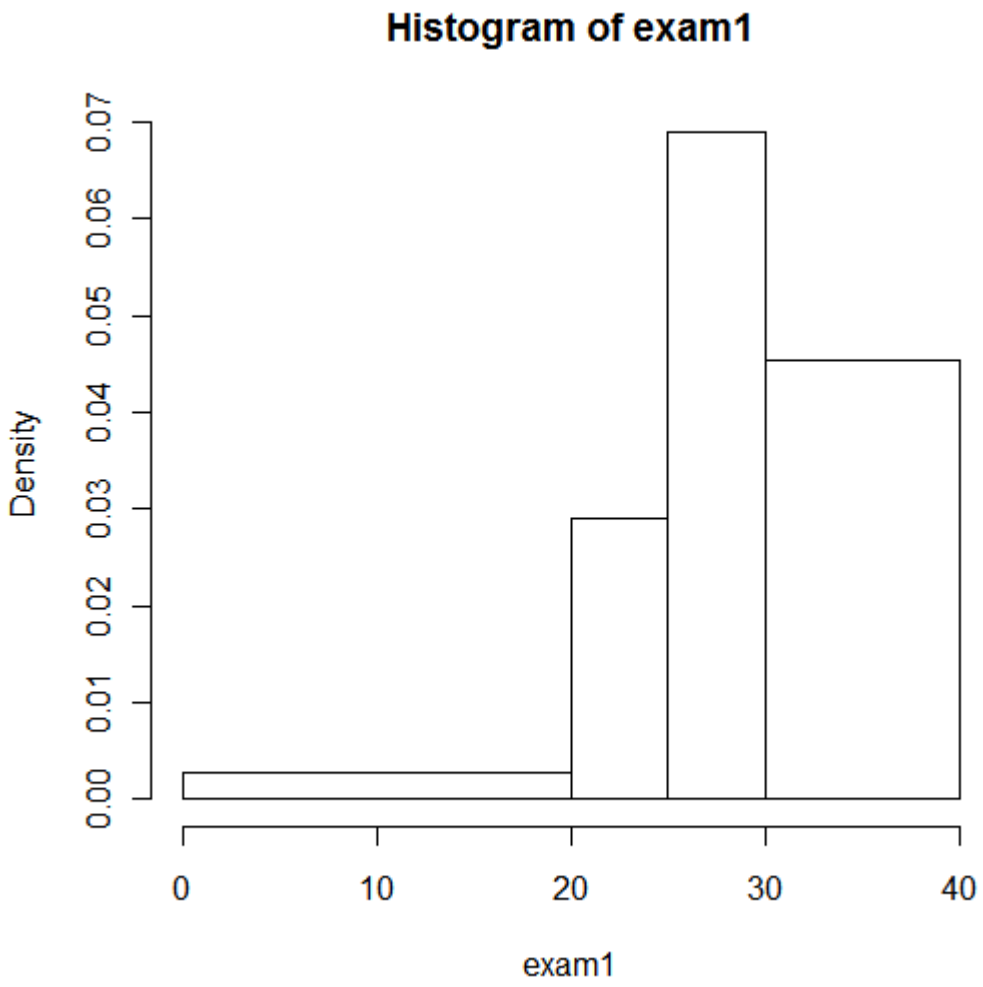
```
> hist(mk2nd[mk2nd<=max(mk2nd)])
```


Histogram of mk2nd[mk2nd <= max(mk2nd)]
Frequency
0
50
100
150
0
2000
4000
6000
8000
10000
12000
mk2nd[mk2nd <= max(mk2nd)]



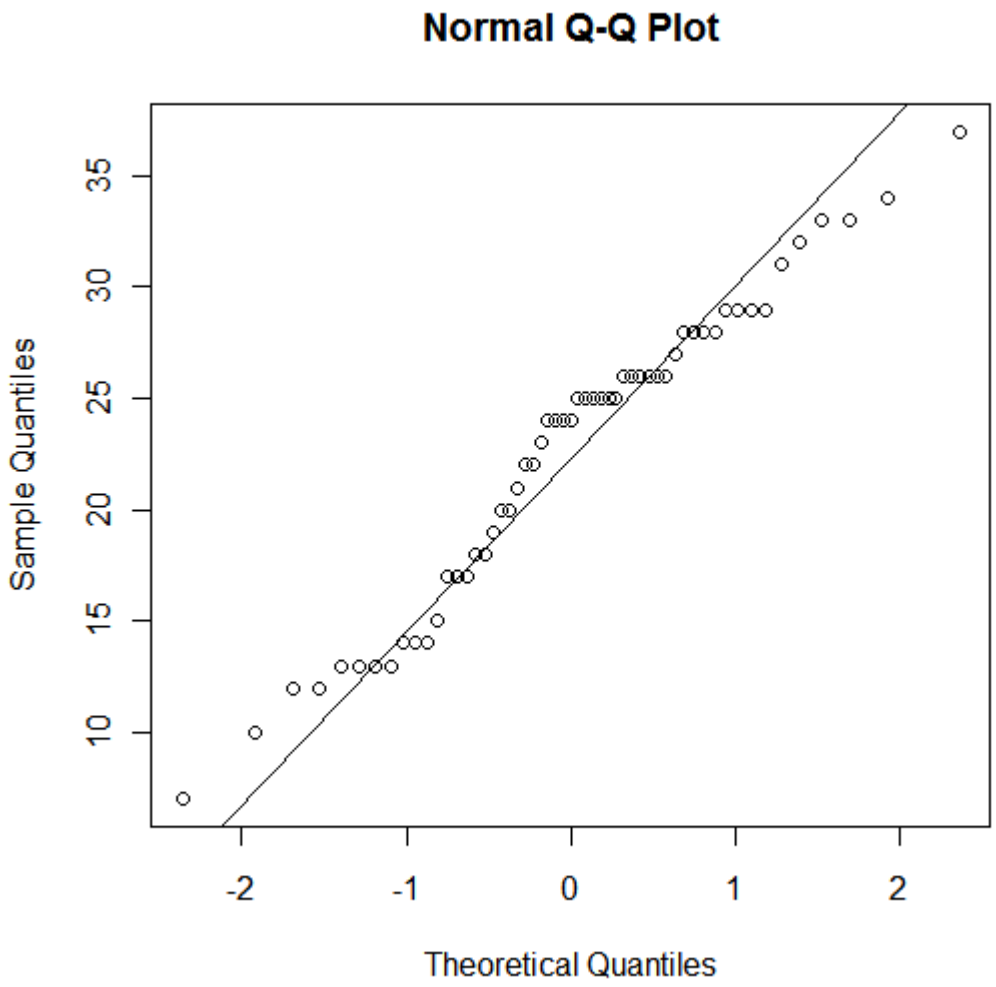
نغير حدود الفئآت

```
> hist(exam1, breaks=c(0,20,25,30,40))
```



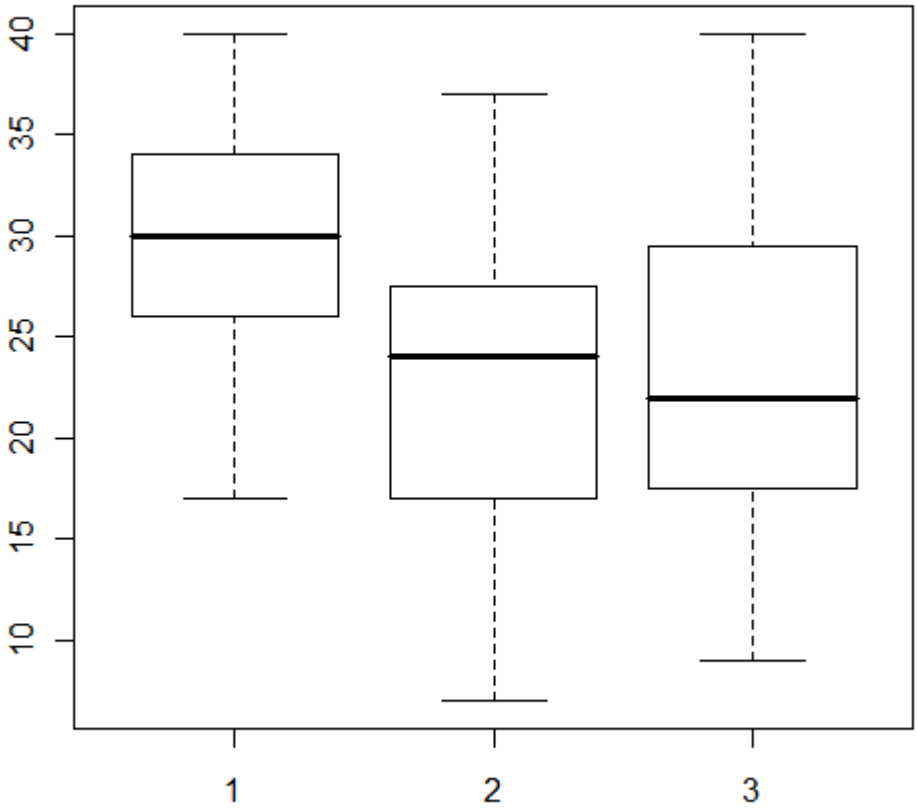
لمقارنة مجموعة بيانات بالتوزيع الطبيعي نستخدم الدالة qqnorm(...) والتي ترسم ربيعات البيانات ضد ربيعات التوزيع الطبيعي والدالة qqline(...) سوف يقوم برسم خط مستقيم على الرسم السابق يمر بالربيعين الأول والثالث وبالتالي إذا كانت البيانات طبيعية فإن نقاط الرسم الأول ستكون قريبة من الخط المستقيم

```
> qqnorm(exam2)
> qqline(exam2)
```



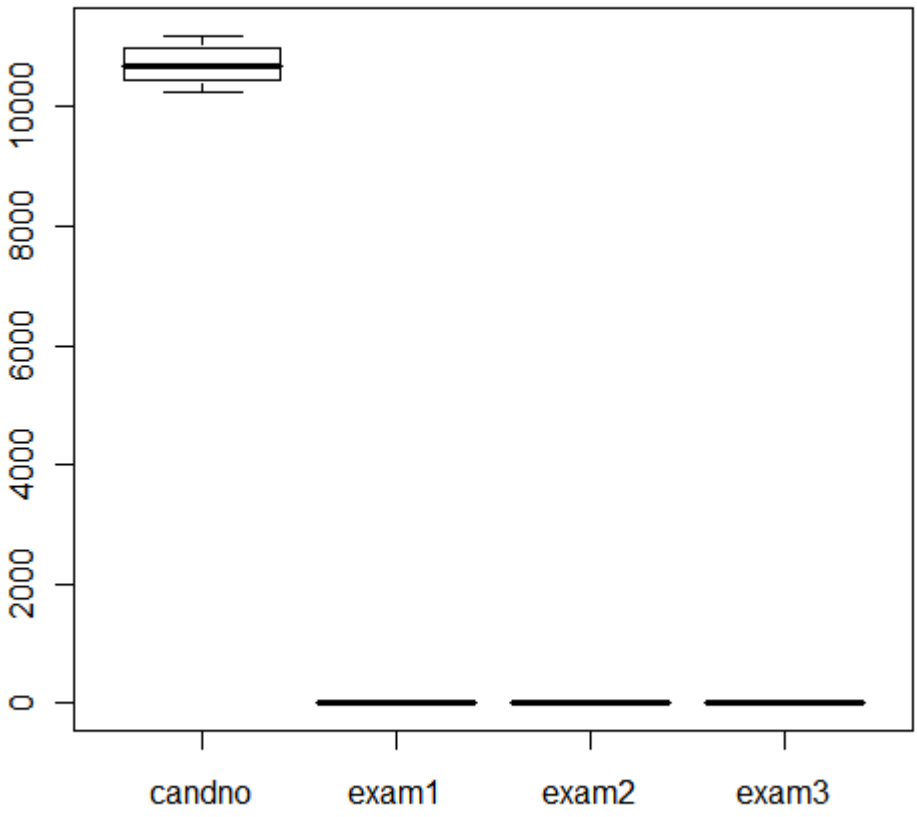
رسومات الصندوق Boxplots

```
> boxplot(exam1,exam2,exam3)
```

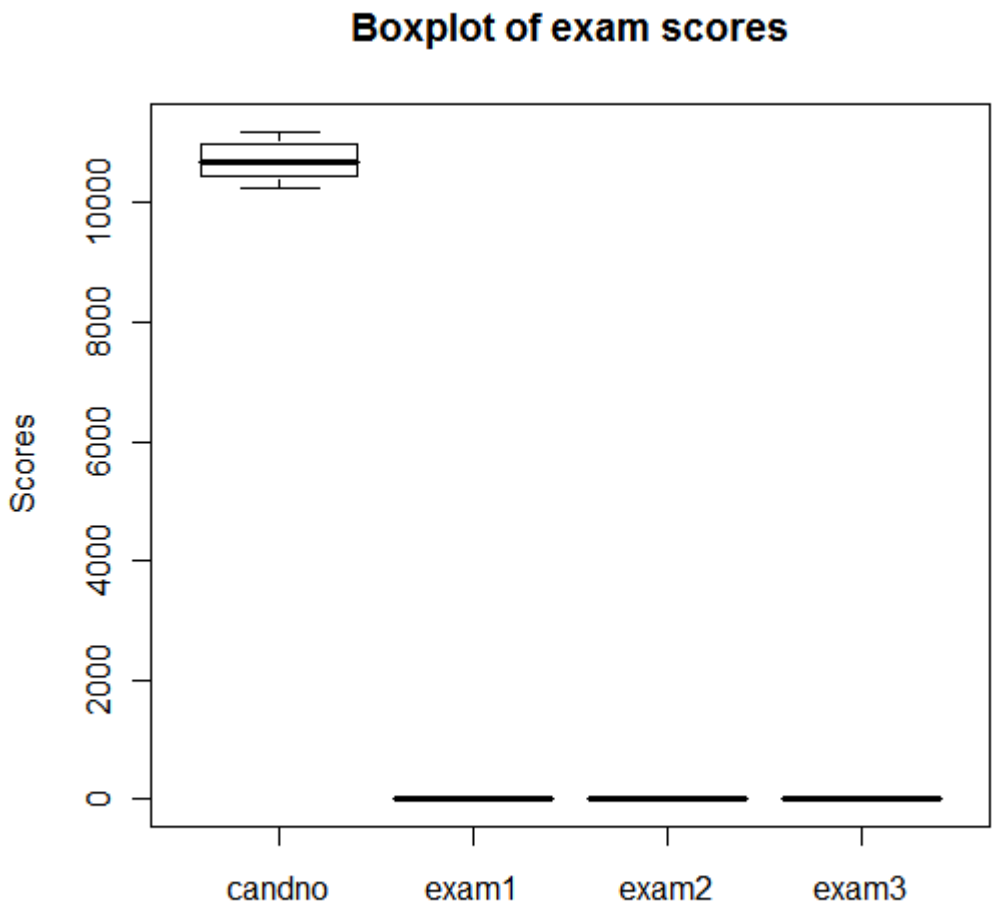


لاحظ التالي:

```
> boxplot(mk2nd)
```



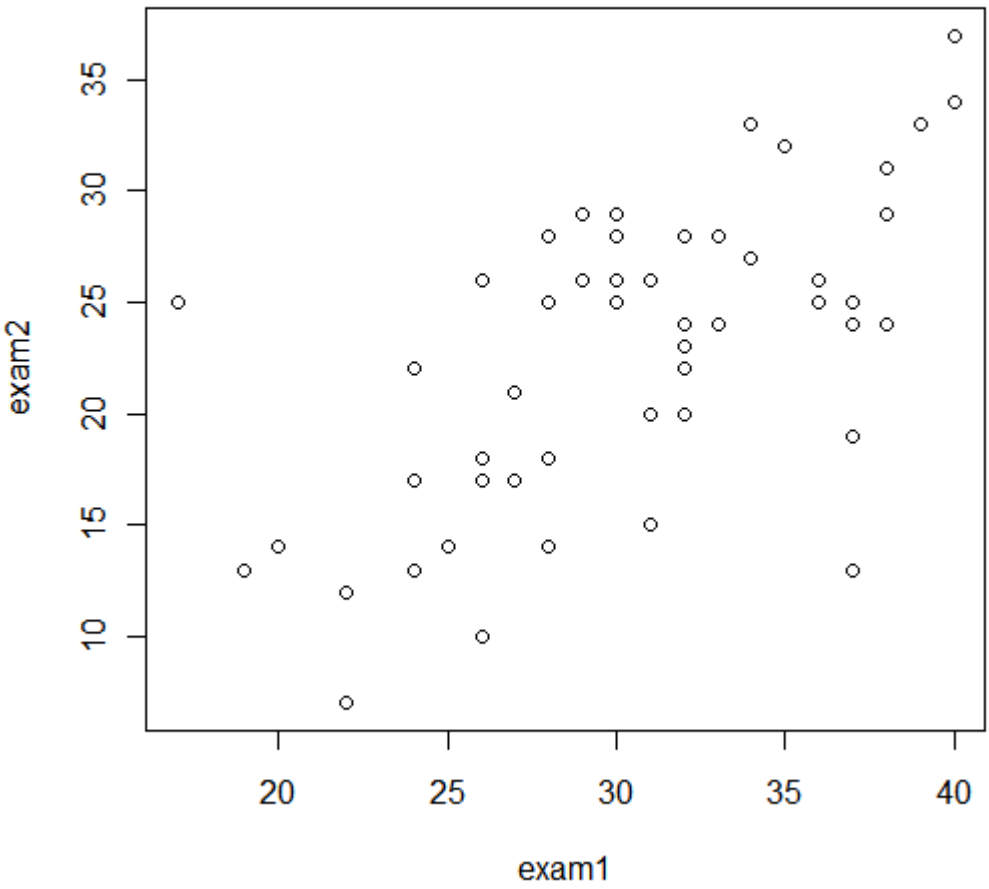

```
> boxplot(mk2nd, main="Boxplot of exam scores",
+ylab="Scores")
```



أنواع الرسومات ومعالم الرسومات والتفاعلية Plot types, graphics parameters and interactivity :

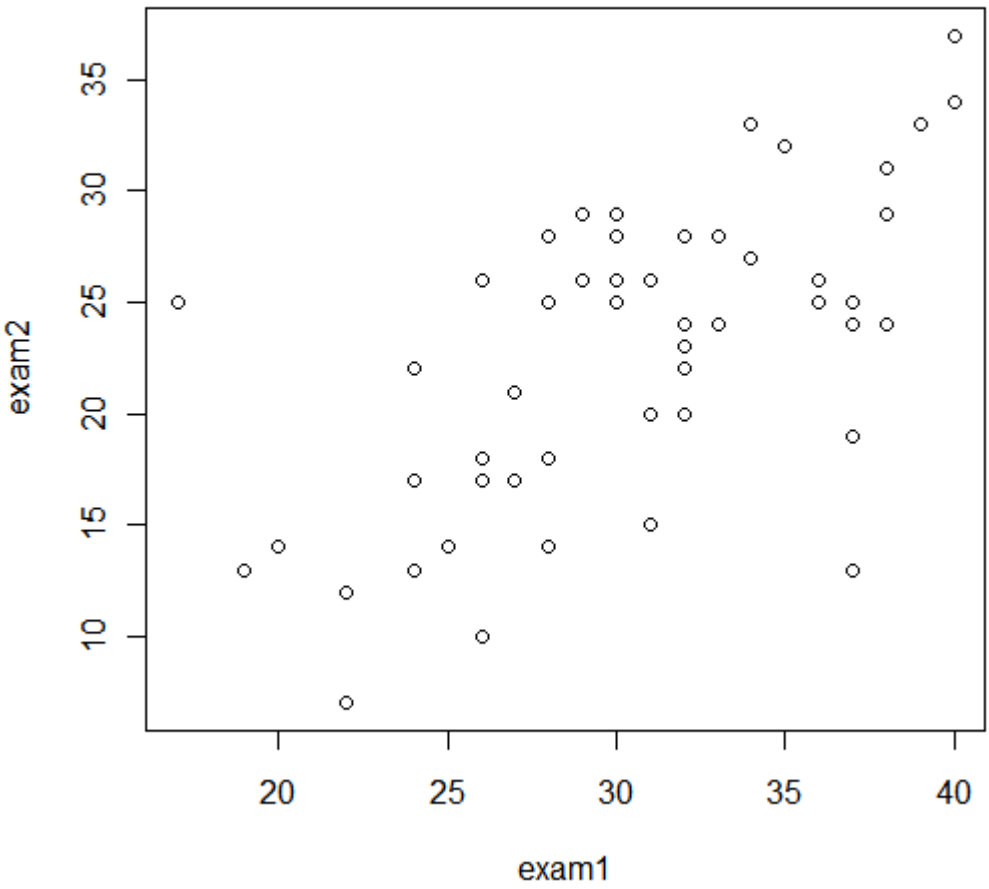
نوع الرسم الإفتراضي في R بسيط جدا ومناسب ولكن قد نحتاج أحيانا أن نغيره ولهذا تستخدم حجة type في دالة الرسم لاحظ التالي:

```
> plot(exam1,exam2)
```



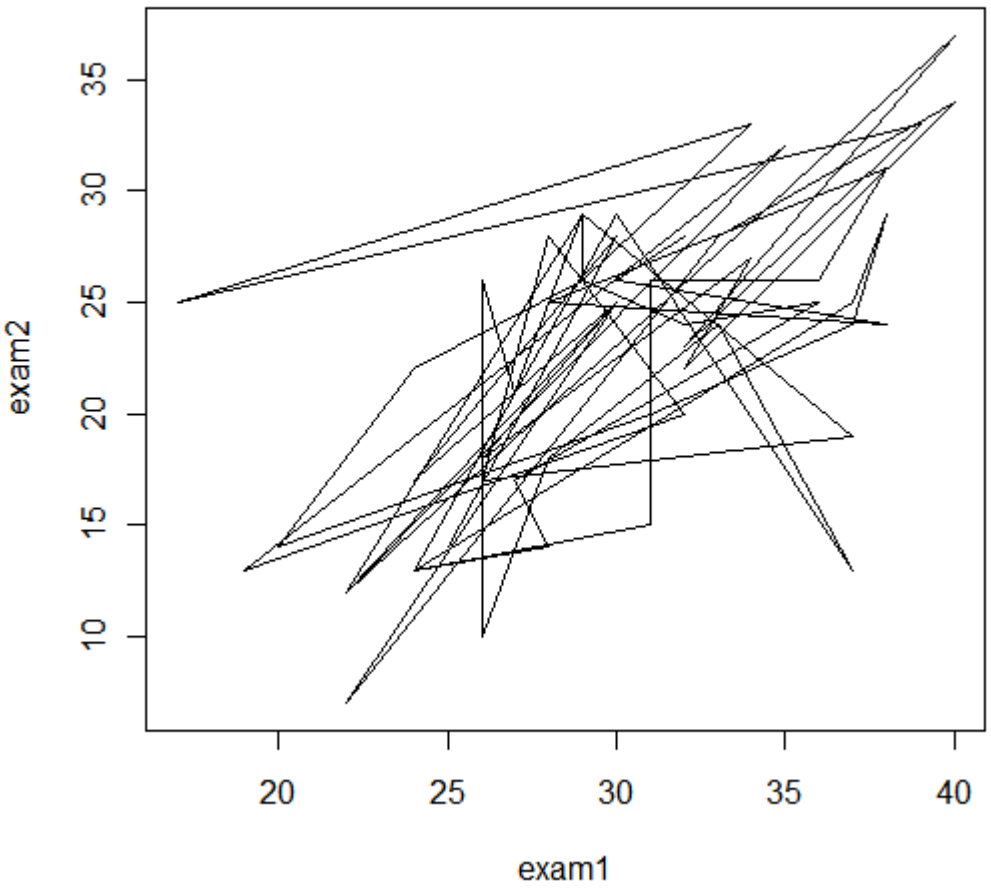

```
> plot(exam1, exam2, type="p") # points (the
default)
```

p تعني نقاط و هي القيم الإفتراضية



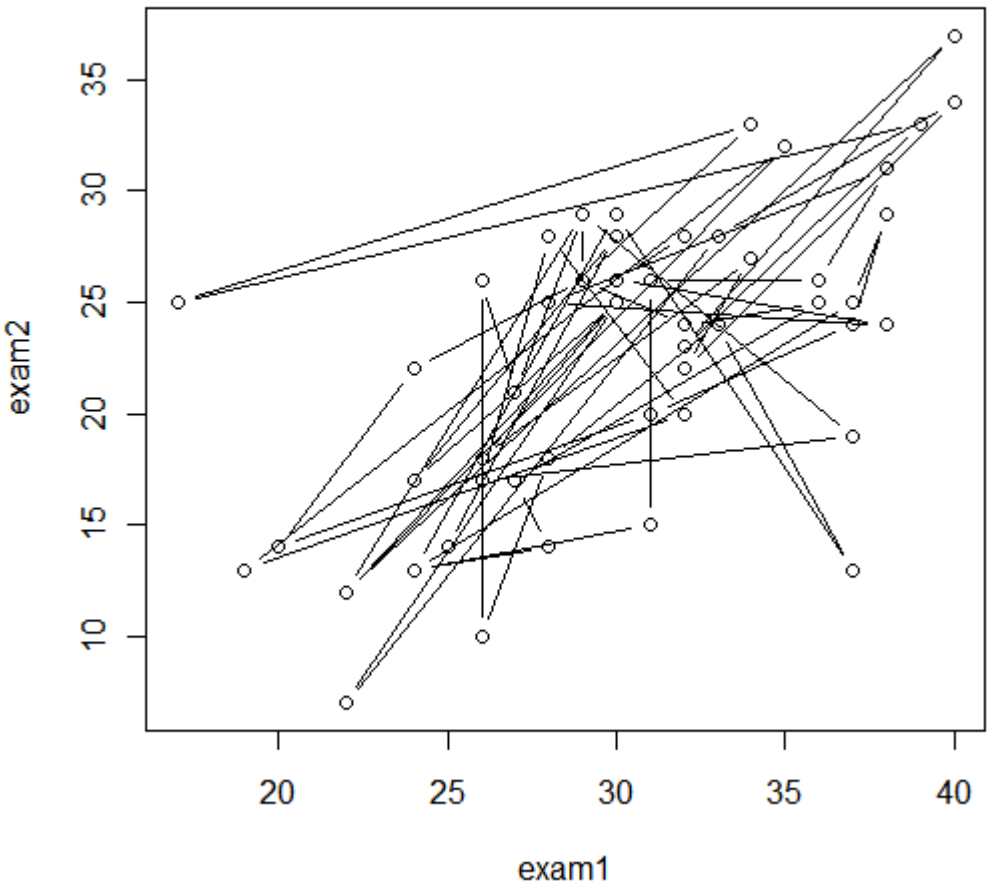

```
> plot(exam1, exam2, type="l") # lines
```

l من line أي خط



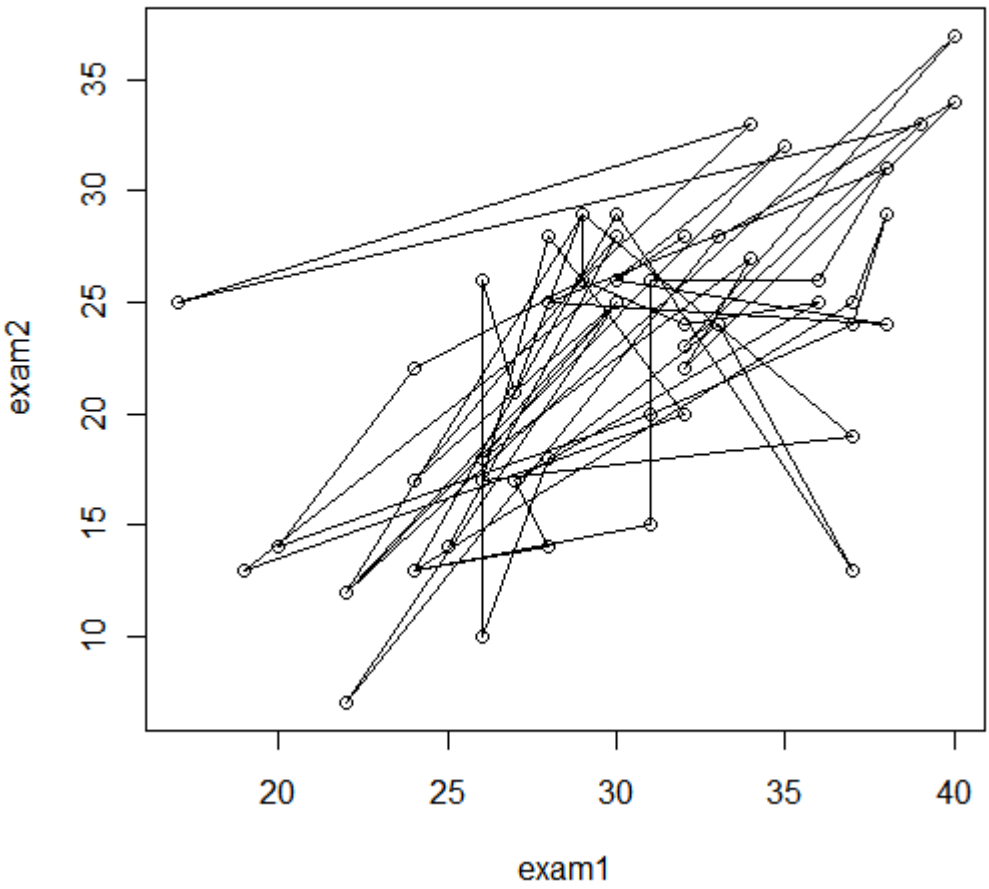

```
> plot(exam1, exam2, type="b") # both lines and
points
```

b من both أي كلا من نقاط و خطوط

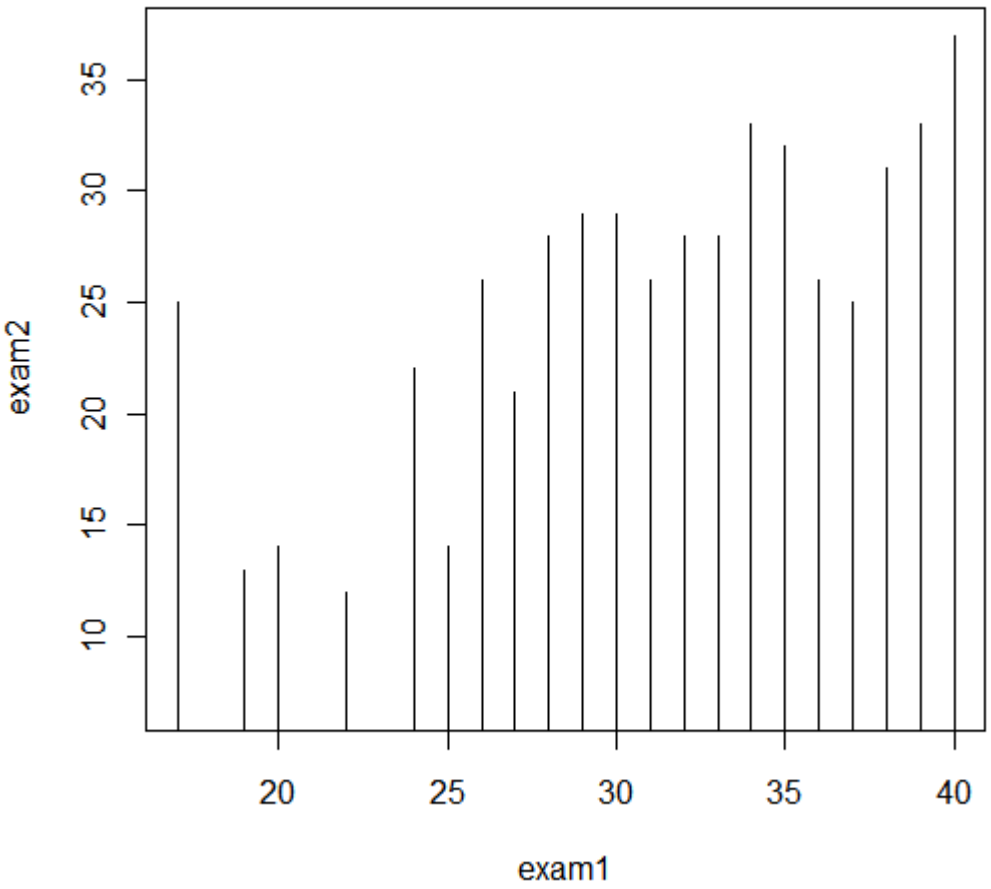



```
> plot(exam1, exam2, type="o") # overlaid lines on
points
```

o من overlaid وهو وضع النقاط على الخطوط

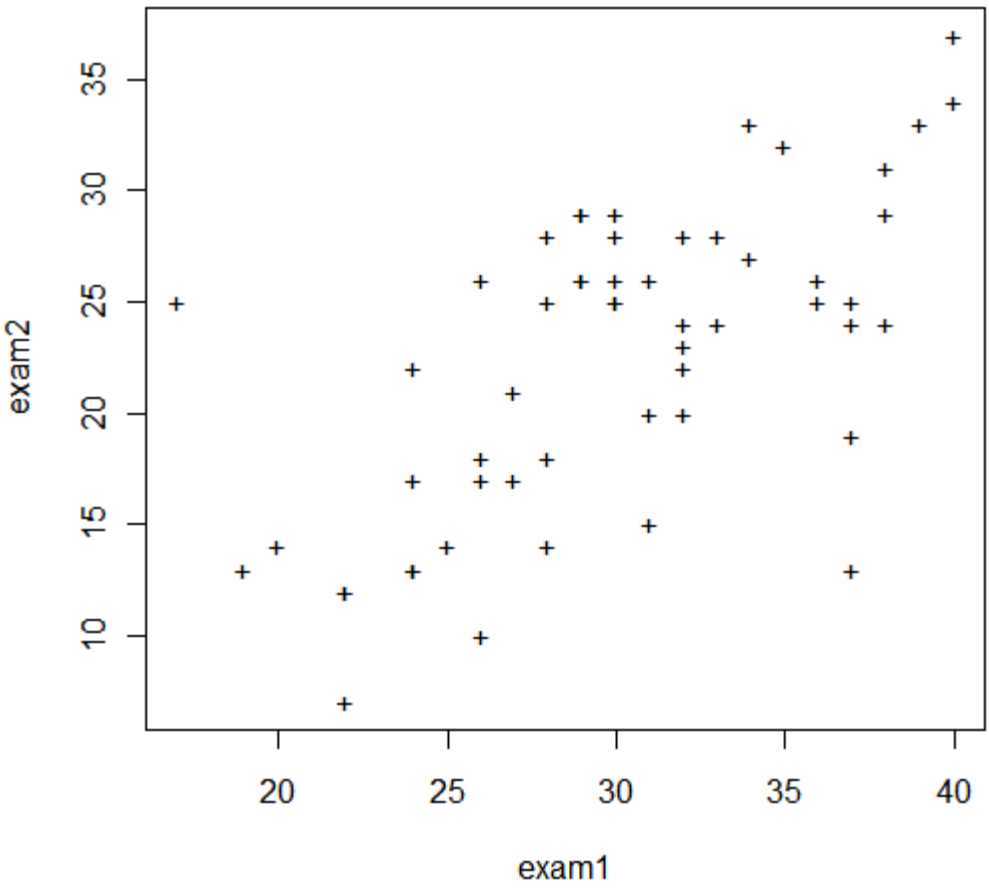



```
> plot(exam1, exam2, type="h") # high density
(vertical lines)
```



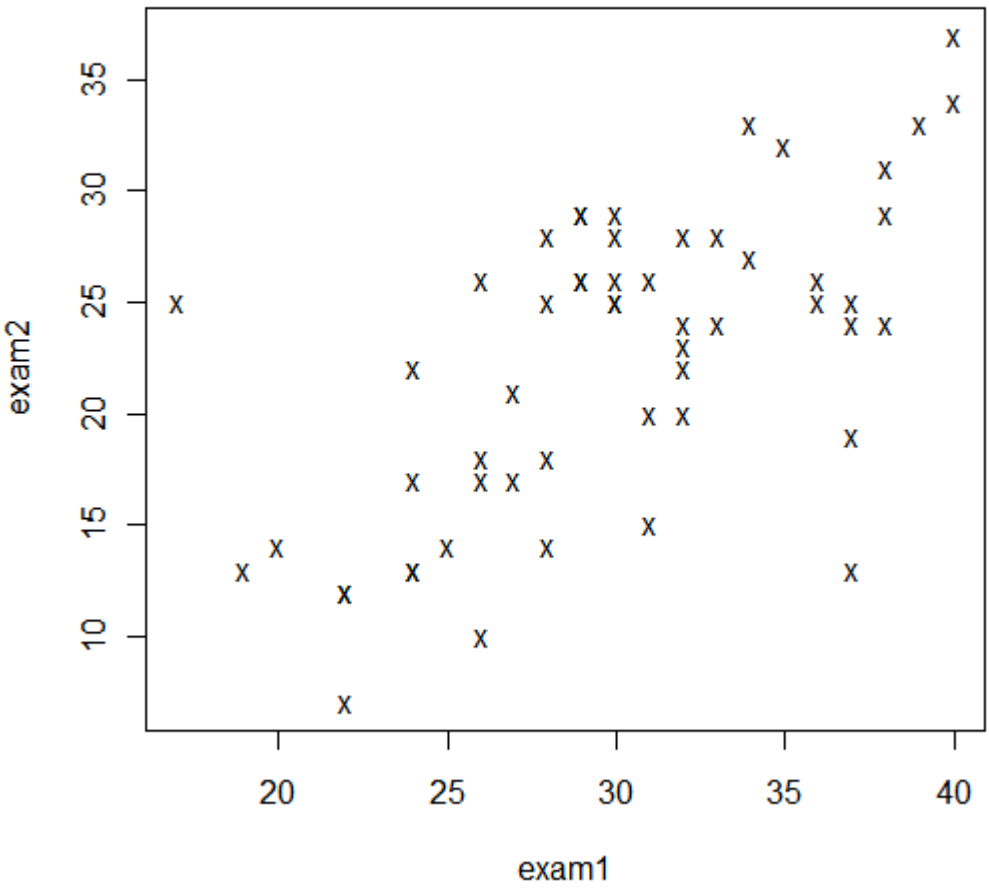
نستطيع تغيير حرف الرسم الإفتراضي *plotting character* والذي هو o بإستخدام الحجة pch إلى أي رمز منها

```
> plot(exam1, exam2, pch="+")
```



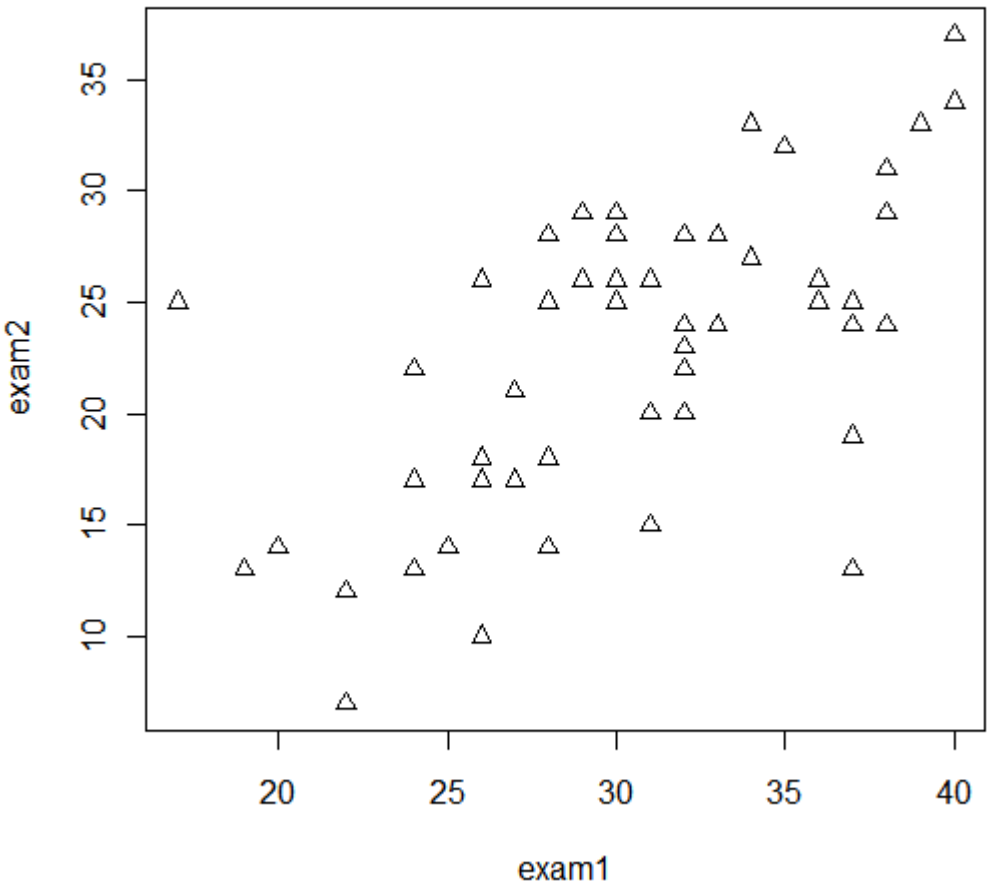
و

```
> plot(exam1, exam2, pch="x")
```



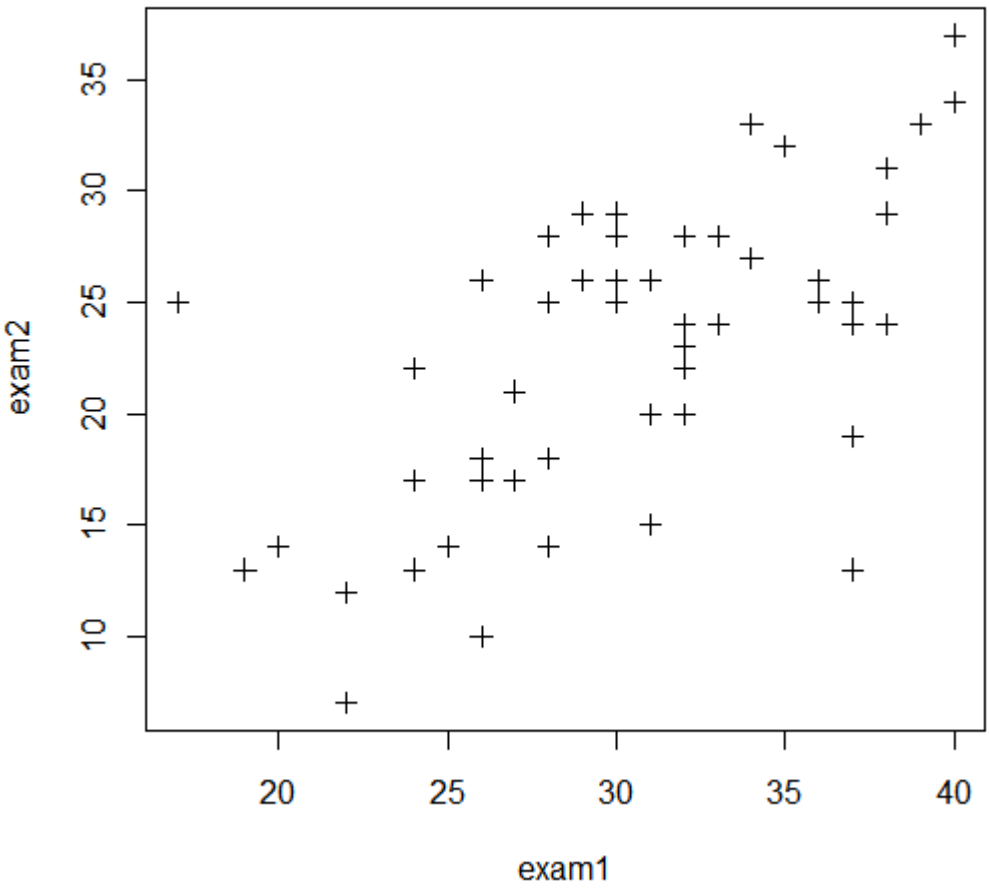
و

```
> plot(exam1, exam2, pch=2)
```



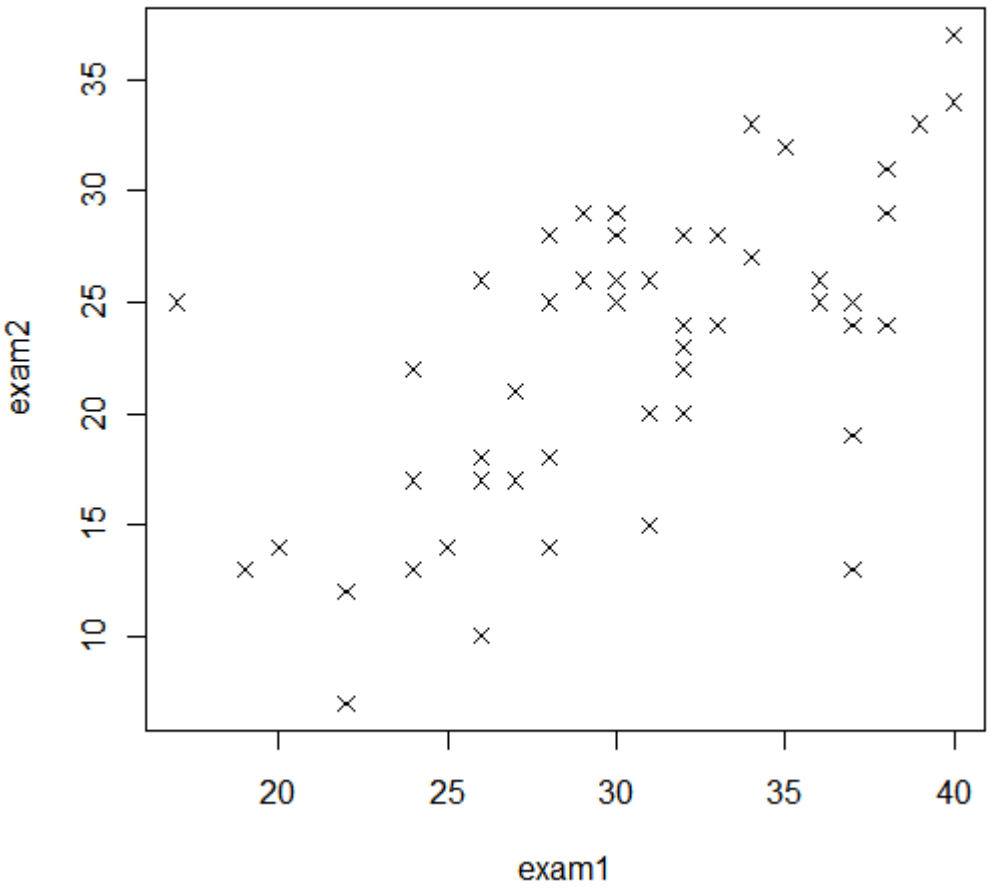
و

```
> plot(exam1, exam2, pch=3)
```



و

```
> plot(exam1, exam2, pch=4)
```



إذا اردنا تغيير معلم رسم بصورة دائمة نستخدم الدالة par(...) وكذلك يستخدم لسرد معالم الرسم

```
> par()
$xlog
[1] FALSE

$ylog
[1] FALSE

$adj
[1] 0.5

$ann
[1] TRUE
```

```
$ask
[1] FALSE

$bg
[1] "transparent"

$bty
[1] "o"

$cex
[1] 1

$cex.axis
[1] 1

$cex.lab
[1] 1

$cex.main
[1] 1.2

$cex.sub
[1] 1

$cin
[1] 0.15 0.20

$col
[1] "black"

$col.axis
[1] "black"

$col.lab
[1] "black"

$col.main
[1] "black"

$col.sub
```

```
[1] "black"

$cra
[1] 14.4 19.2

$crt
[1] 0

$csi
[1] 0.2

$cxy
[1] 0.03437083 0.05328007

$din
[1] 5.604166 5.593749

$err
[1] 0

$family
[1] ""

$fg
[1] "black"

$fig
[1] 0 1 0 1

$fin
[1] 5.604166 5.593749

$font
[1] 1

$font.axis
[1] 1

$font.lab
[1] 1
```

```
$font.main
[1] 2

$font.sub
[1] 1

$lab
[1] 5 5 7

$las
[1] 0

$lend
[1] "round"

$lheight
[1] 1

$ljoin
[1] "round"

$lmitre
[1] 10

$lty
[1] "solid"

$lwd
[1] 1

$mai
[1] 1.02 0.82 0.82 0.42

$mar
[1] 5.1 4.1 4.1 2.1

$mex
[1] 1
```

```
$mfcol
[1] 1 1

$mfg
[1] 1 1 1 1

$mfrow
[1] 1 1

$mgp
[1] 3 1 0

$mkh
[1] 0.001

$new
[1] FALSE

$oma
[1] 0 0 0 0

$omd
[1] 0 1 0 1

$omi
[1] 0 0 0 0

$pch
[1] 1

$pin
[1] 4.364166 3.753749

$plt
[1] 0.1463197 0.9250557 0.1823464 0.8534078

$ps
[1] 12

$pty
```

```
[1] "m"

$smo
[1] 1

$srt
[1] 0

$tck
[1] NA

$tcl
[1] -0.5

$usr
[1] 0 1 0 1

$xaxp
[1] 0 1 5

$xaxs
[1] "r"

$xaxt
[1] "s"

$xpd
[1] FALSE

$yaxp
[1] 0 1 5

$yaxs
[1] "r"

$yaxt
[1] "s"

>
```

لمعرفة القيم الإفتراضية للون والحرف

```
> par(c("pch", "col"))
$pch
[1] 1

$col
[1] "black"

>
```

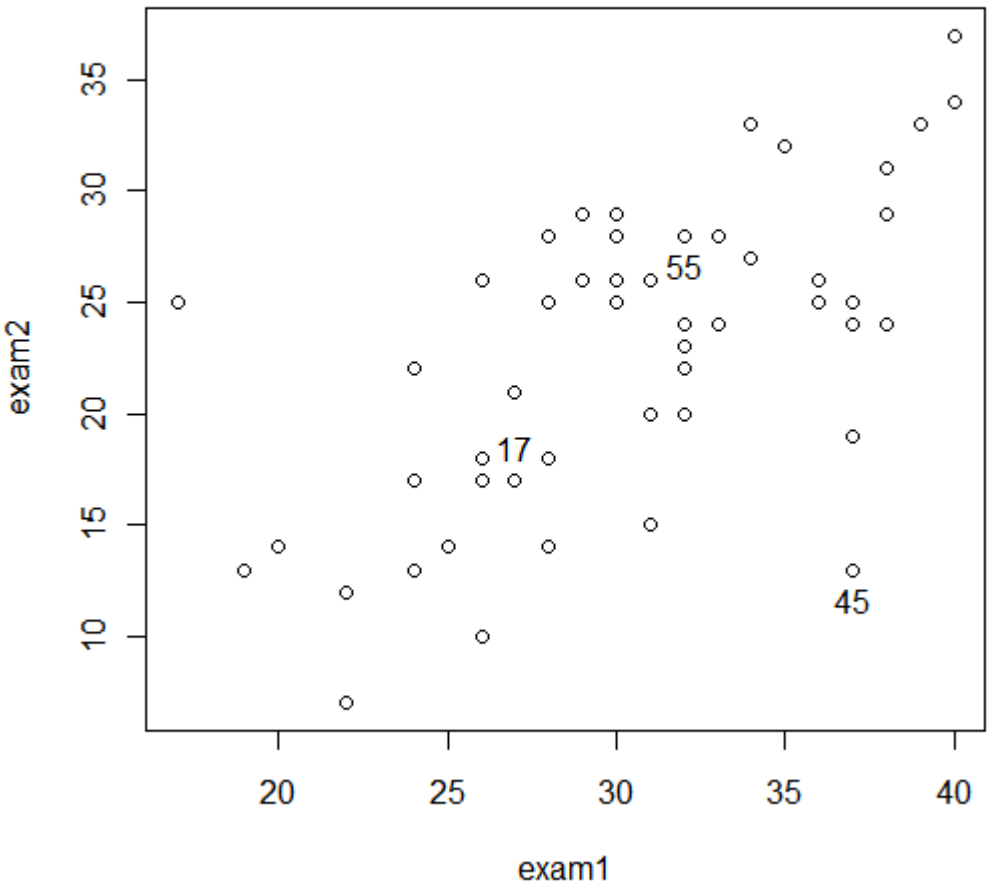
لتغيير الحرف الإفتراضي إلى *

```
> par(pch="*")
```

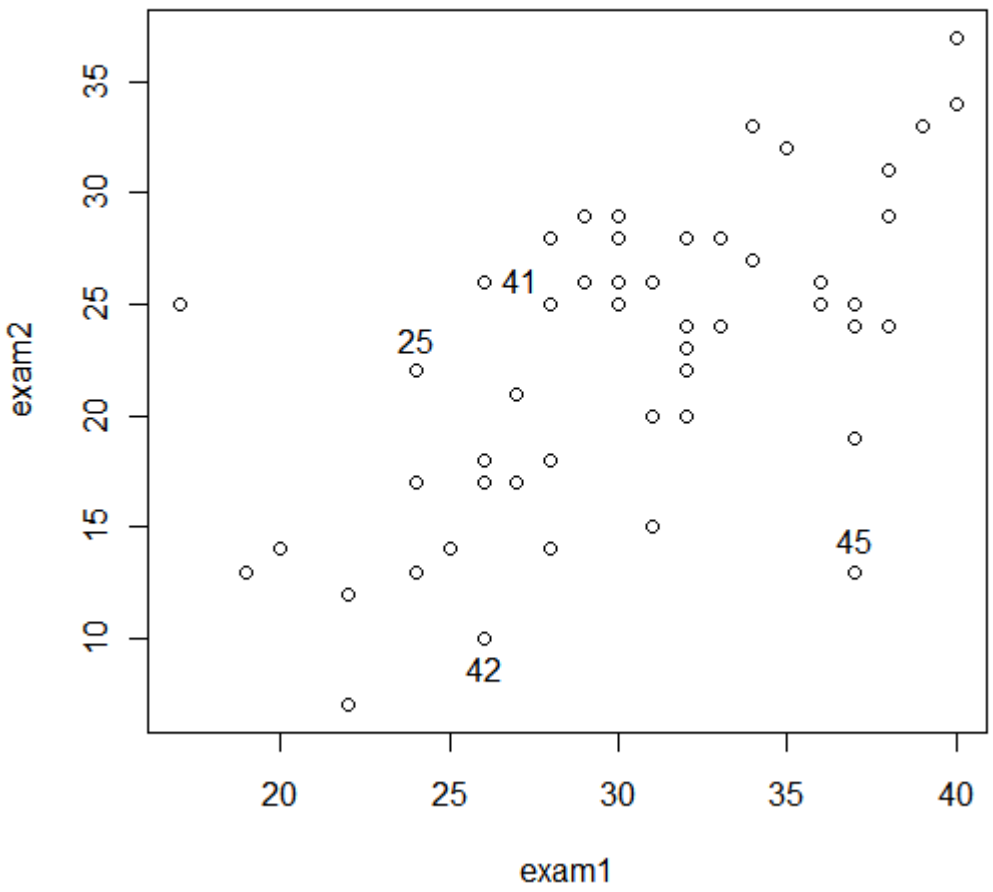
الرسومات التفاعلية:

تمتلك R عدد من إمكانيات الرسم التفاعلي وأحد هذه هو الدالة identify(…) والتي تمكن من تأشير نقاط مهمة في الرسم. لاحظ بعد إدخال الدالة identify(…) سوف ينتظر البرنامج حتى يستطيع المستخدم إختيار النقاط مستخدما الفارة وتوقف العملية بإستخدام الزر الأيمن للفارة

```
> plot(exam1,exam2)
> identify(exam1,exam2)
[1] 17 45 55
>
```


exam2
10
15
20
25
30
35
20
25
30
35
40
exam1
55
17
45

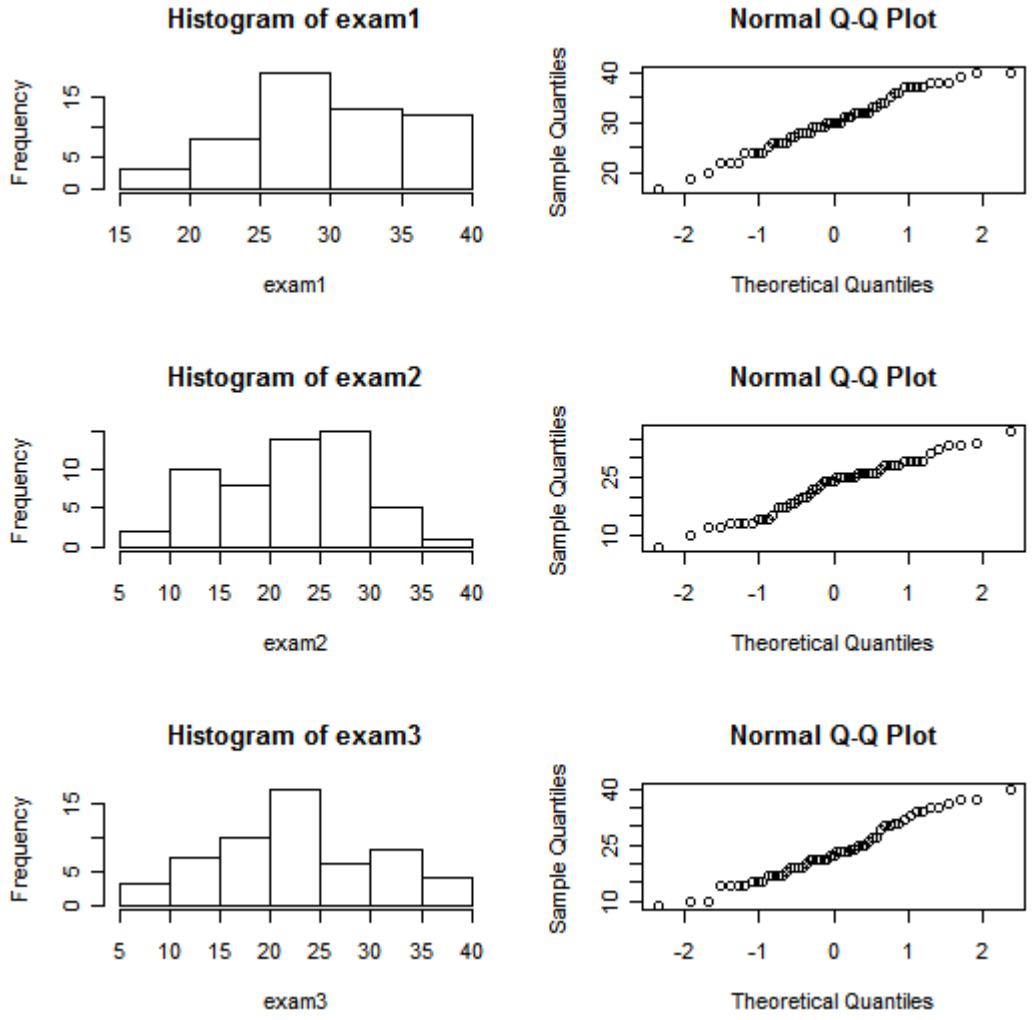
```
> plot(exam1,exam2)
> identify(exam1,exam2,row.names(mk2nd))
[1] 25 41 42 45
>
```



في الإمكان وضع أكثر من رسم واحد في الصفحة وذلك بإستخدام المعلم mfrow والذي تكون قيمته لمتجة عددي صحيح ذا بعدين يعطي عدد الأسطر والأعمدة لاحظ نتيجة التالي:

```
> par(mfrow=c(3,2))
> hist(exam1)
> qqnorm(exam1)
> hist(exam2)
> qqnorm(exam2)
> hist(exam3)
> qqnorm(exam3)
```

```
> par(mfrow=c(1,1))
```



لاحظ الأمر الأخير الذي يعيد الرسم للقيمة الإفتراضية.
جرب التالي:

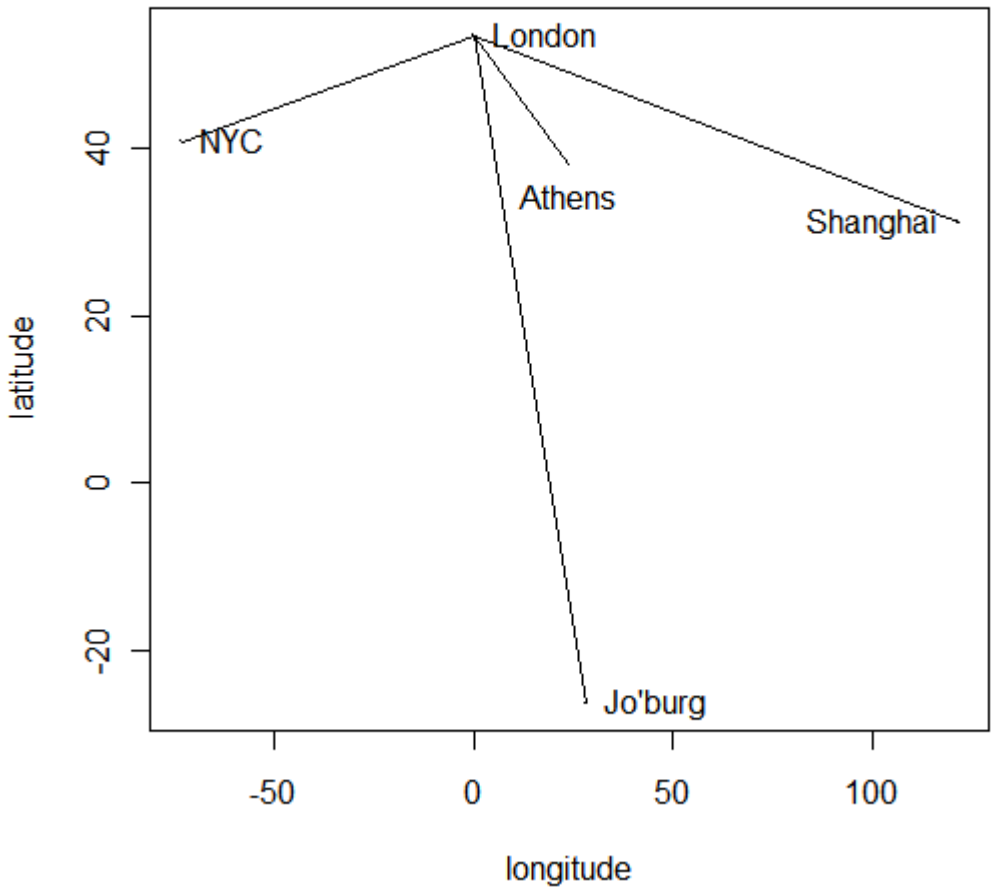
```
> par(mfcol=c(3,2))
```

والذي يستخدم mfcol بدلا من mfrow.
تمرين: أدرس التالي

```
> citynames <- c("Athens", "Jo'burg", "London",
"NYC", "Shanghai")
> longitude <- c(23.72, 28.07, -0.08, -73.47,
121.47)
> latitude <- c(37.97, -26.20, 53.42, 40.78, 31.17)
> cities <- data.frame(latitude,longitude)
> row.names(cities) <- citynames
> cities
```

```
          latitude longitude
Athens       37.97     23.72
Jo'burg     -26.20     28.07
London       53.42     -0.08
NYC          40.78    -73.47
Shanghai     31.17    121.47
> rm(latitude,longitude,citynames)
> attach(cities)
> plot(longitude,latitude,type="n")
> points(longitude,latitude,pch=".")
> text(longitude[1], latitude[1],
row.names(cities)[1],pos=1)
> text(longitude[2], latitude[2],
row.names(cities)[2],pos=4)
> text(longitude[3], latitude[3],
row.names(cities)[3],pos=4)
> text(longitude[4], latitude[4],
row.names(cities)[4],pos=4)
> text(longitude[5], latitude[5],
row.names(cities)[5],pos=2)
> lines(c(longitude[1],longitude[3]),
c(latitude[1],latitude[3]))
> lines(c(longitude[2],longitude[3]),
c(latitude[2],latitude[3]))
> lines(c(longitude[4],longitude[3]),
c(latitude[4],latitude[3]))
> lines(c(longitude[5],longitude[3]),
c(latitude[5],latitude[3]))
>
```

الرسم الناتج:



لاحظ التالي:

1- أمر الرسم `plot(longitude,latitude,type="n")` سوف يقوم برسم الإحداثيات ولكن type = "n" سوف يمنع رسم أي شيئ آخر.

2- الحجة pos تأخذ القيمة 1 لأعلى و 2 لليسار و 3 لأسفل و 4 لليمين. لتنسيق وضع النصوص يمكن إستخدام الحجج adj و offset وتترك كتمرين.

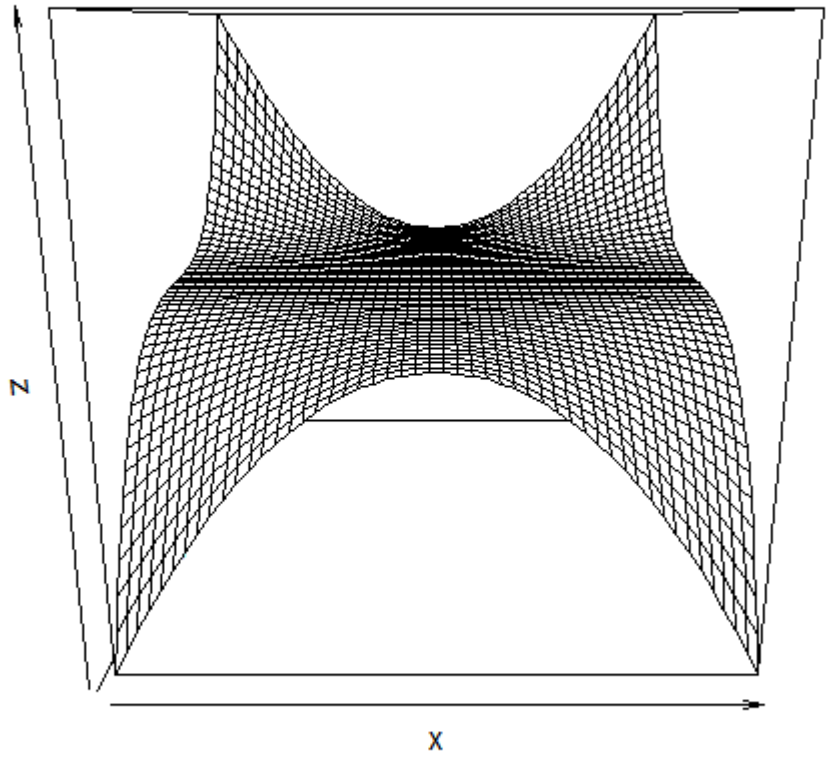
الرسم ثلاثي الأبعاد 3-D:

سوف نقوم برسم السطح $f(x,y)=x^2y^3$ في المجال $x\in[-1,1]$ و $y\in[-1,1]$

ولكي نقوم بهذا نولد متتابعة منسقة من 50 نقطة للفترة $[-1,1]$ ثم نستخدم الدالة outer (للضرب الخارجي) لتوليد مصفوفة 50X50 لنقاط على السطح

```
> x <- seq(-1,1,length=50)
> y <- seq(-1,1,length=50)
```

```
> z <- outer(x^2,y^3)
> contour(x,y,z)
> image(x,y,z)
> persp(x,y,z)
```



يمكن تحسين شكل الرسم وذلك بتغيير بعض القيم الإفتراضية.

أساسيات R essentials :

في أي جلسة من جلسات R يقوم البرنامج بحفظ الأوامر المدخلة من لوحة المفاتيح في منطقة من الذاكرة buffer ويمكن إستعادة هذه الأوامر بأحد المفاتيح ↑ أو ↓ وقد يكون من المفيد تخزين هذة الأوامر للإستخدام في جلسة أخرى لهذا نحفظ هذا التاريخ بالأمر:

```
> savehistory("filename.Rhistory")
```

وتستعاد بالأمر:

```
> loadhistory("filename.Rhistory")
```

بعد إستعادة الأوامر يمكن إستخدامها بأحد المفاتيح ↑ أو ↓ أما إذا شئنا إجراء هذه الأوامر نستخدم الأمر:

```
> source("filename ", echo=TRUE)
```

مسار البحث The search path

عند إمرار معرف لدالة تقوم R بالبحث عن الشيئ التابع للمعرف. أول مكان تبحث فيه R هو فضاء العمل (والذي يعرف أيضا بالإسم البيئة الشاملة *global environment* ويرمز له GlobalEnv.) إذا لم يوجد هذا الشيئ في فضاء العمل سيقوم R بالبحث في أي قائمة مضافة (بما فيها أطر البيانات) والحزم المحملة بواسطة المستخدم والحزم المحملة ذاتيا ومن هنا تتضح عمل attach(…) الحقيقي حيث أنها تضع قائمة في مسار البحث و detach(…) تزيلها من المسار. الدالة search() تعطي مسار البحث وإعطاء إسم أي بيئة أو موقعها لدالة الأشياء سوف يقوم بسرد الأشياء في تلك البيئة

```
> search()
> objects(package:MASS)
> objects(4)
```

حزم R packages:

وهي عبارة عن ملفات تحوي برامج إضافية متخصصة تضاف إلى R لإعطائها مقدرات لحل مجال أكبر من المشاكل لاتتوفر في أساس R. وفي مقدورنا سرد قائمة بالحزم المتوفرة وكذلك تحميل هذه الحزم وتجديدها بإستخدام الأمر **library(…)**. الأمر التالي يسرد كل الحزم المتوفرة

```
> library()
Packages in library 'C:/PROGRA~1/R/R-28~1.1/library':
```

```
base                    The R Base Package
boot                    Bootstrap R (S-Plus) Functions (Canty)
class                   Functions for Classification
cluster                 Cluster Analysis Extended Rousseeuw et al.
codetools               Code Analysis Tools for R
datasets                The R Datasets Package
foreign                 Read Data Stored by Minitab, S, SAS, SPSS,
                        Stata, Systat, dBase, ...
graphics                The R Graphics Package
grDevices               The R Graphics Devices and Support for
                        Colours and Fonts
grid                    The Grid Graphics Package
KernSmooth              Functions for kernel smoothing for Wand &
                        Jones (1995)
lattice                 Lattice Graphics
MASS                    Main Package of Venables and Ripley's MASS
methods                 Formal Methods and Classes
mgcv                    GAMs with GCV/AIC/REML smoothness
estimation
                        and GAMMs by PQL
nlme                    Linear and Nonlinear Mixed Effects Models
nnet                    Feed-forward Neural Networks and
Multinomial
                        Log-Linear Models
R.methodsS3             Utility function for defining S3 methods
R.oo                    R object-oriented programming with or
                        without references
rpart                   Recursive Partitioning
Sim101                  Sim 101 Package for WSC 2008
spatial                 Functions for Kriging and Point Pattern
                        Analysis
splines                 Regression Spline Functions and Classes
stats                   The R Stats Package
stats4                  Statistical Functions using S4 Classes
survival                Survival analysis, including penalised
                        likelihood.
tcltk                   Tcl/Tk Interface
tools                   Tools for Package Development
utils                   The R Utils Package
```

الأمر التالي يحمل الحزمة MASS

```
> library(MASS)
```

ونستطيع مشاهدة محتواه

```
> library(help=MASS)

                Information on package 'MASS'

Description:
```

```
Bundle:             VR
Priority:           recommended
Contains:           MASS class nnet spatial
Version:            7.2-45
Date:               2008-12-07
Depends:            R (>= 2.4.0), grDevices, graphics, stats,
                    utils
Suggests:           lattice, nlme, survival
Author:             S original by Venables & Ripley. R port by
                    Brian Ripley <ripley@stats.ox.ac.uk>,
                    following earlier work by Kurt Hornik and
                    Albrecht Gebhardt.
Maintainer:         Brian Ripley <ripley@stats.ox.ac.uk>
BundleDescription:  Functions and datasets to support Venables
                    and Ripley, 'Modern Applied Statistics with
                    S' (4th edition).
License:            GPL-2 | GPL-3
URL:                http://www.stats.ox.ac.uk/pub/MASS4/
Packaged:           Tue Dec 9 11:11:17 2008; ripley
Package:            MASS
Description:        The main library and the datasets
Title:              Main Package of Venables and Ripley's MASS
LazyLoad:           yes
LazyData:           yes
Built:              R 2.8.1; i386-pc-mingw32; 2008-12-22
                    09:22:56; windows

Index:

Functions:
=========

Null                    Null Spaces of Matrices
addterm                 Try All One-Term Additions to a Model
anova.negbin            Likelihood Ratio Tests for Negative
Binomial GLMs
area                    Adaptive Numerical Integration
bandwidth.nrd           Bandwidth for density() via Normal
Reference
                          Distribution
bcv                     Biased Cross-Validation for Bandwidth
Selection
boxcox                  Box-Cox Transformations for Linear Models
con2tr                  Convert Lists to Data Frames for use by
Trellis
confint-MASS            Confidence Intervals for Model Parameters
contr.sdif              Successive Differences contrast coding
corresp                 Simple Correspondence Analysis
cov.rob                 Resistant Estimation of Multivariate
Location and
                          Scatter
```

```
cov.trob                Covariance Estimation for Multivariate t
Distribution
denumerate              Transform an Allowable Formula for
'loglm'
                          into one for 'terms'
dose.p                  Predict Doses for Binomial Assay model
dropterm                Try All One-Term Deletions from a Model
eqscplot                Plots with Geometrically Equal Scales
fitdistr                Maximum-likelihood Fitting of Univariate
Distributions
fractions               Rational Approximation
gamma.dispersion        Calculate the MLE of the Gamma Dispersion
Parameter
                          in a GLM Fit
gamma.shape             Estimate the Shape Parameter of the Gamma
Distribution
                          in a GLM Fit
ginv                    Generalized Inverse of a Matrix
glm.convert             Change a Negative Binomial fit to a GLM
fit
glm.nb                  Fit a Negative Binomial Generalized
Linear Model
glmmPQL                 Fit Generalized Linear Mixed Models via
PQL
hist.scott              Plot a Histogram with Automatic Bin Width
Selection
huber                   Huber M-estimator of Location with MAD
Scale
hubers                  Huber Proposal 2 Robust Estimator of
Location
                          and/or Scale
isoMDS                  Kruskal's Non-metric Multidimensional
Scaling
kde2d                   Two-Dimensional Kernel Density Estimation
lda                     Linear Discriminant Analysis
ldahist                 Histograms or Density Plots of Multiple
Groups
lm.gls                  Fit Linear Models by Generalized Least
Squares
lm.ridge                Ridge Regression
loglm                   Fit Log-Linear Models by Iterative
                        Proportional Scaling
logtrans                Estimate log Transformation Parameter
lqs                     Resistant Regression
mca                     Multiple Correspondence Analysis
mvrnorm                 Simulate from a Multivariate Normal
Distribution
negative.binomial       Family function for Negative Binomial
GLMs
newcomb                 Newcomb's Measurements of the Passage
Time of Light
```

```
pairs.lda               Produce Pairwise Scatterplots from an
'lda' Fit
parcoord                Parallel Coordinates Plot
plot.lda                Plot Method for Class 'lda'
plot.mca                Plot Method for Objects of Class 'mca'
polr                    Proportional Odds Logistic Regression
predict.lda             Classify Multivariate Observations by
Linear
                          Discrimination
predict.lqs             Predict from an lqs Fit
predict.mca             Predict Method for Class 'mca'
predict.qda             Classify from Quadratic Discriminant
Analysis
qda                     Quadratic Discriminant Analysis
rational                Rational Approximation
renumerate              Convert a Formula Transformed by
'denumerate'
rlm                     Robust Fitting of Linear Models
rms.curv                Relative Curvature Measures for Non-
Linear Regression
rnegbin                 Simulate Negative Binomial Variates
sammon                  Sammon's Non-Linear Mapping
stdres                  Extract Standardized Residuals from a
Linear Model
stepAIC                 Choose a model by AIC in a Stepwise
Algorithm
studres                 Extract Studentized Residuals from a
Linear Model
summary.loglm           Summary Method Function for Objects of
Class 'loglm'
summary.negbin          Summary Method Function for Objects of
Class 'negbin'
summary.rlm             Summary Method for Robust Linear Models
synth.tr                Synthetic Classification Problem
theta.md                Estimate theta of the Negative Binomial
by Deviance
theta.ml                Estimate theta of the Negative Binomial
by Maximum
                          Likelihood
theta.mm                Estimate theta of the Negative Binomial
by Moments
truehist                Plot a Histogram
ucv                     Unbiased Cross-Validation for Bandwidth
Selection
width.SJ                Bandwidth Selection by Pilot Estimation
of Derivatives
write.matrix            Write a Matrix or Data Frame


Datasets:
========
```

Aids2	Australian AIDS Survival Data
Animals	Brain and Body Weights for 28 Species
Belgian-phones	Belgium Phone Calls 1950-1973
Boston	Housing Values in Suburbs of Boston
Cars93	Data from 93 Cars on Sale in the USA in 1993
Cushings	Diagnostic Tests on Patients with Cushing's Syndrome
DDT	DDT in Kale
GAGurine	Level of GAG in Urine of Children
Insurance	Numbers of Car Insurance claims
Melanoma	Survival from Malignant Melanoma
OME	Tests of Auditory Perception in Children with OME
Pima.tr	Diabetes in Pima Indian Women
Rabbit	Blood Pressure in Rabbits
Rubber	Accelerated Testing of Tyre Rubber
SP500	Returns of the Standard and Poors 500
Sitka	Growth Curves for Sitka Spruce Trees in 1988
Sitka89	Growth Curves for Sitka Spruce Trees in 1989
Skye	AFM Compositions of Aphyric Skye Lavas
Traffic	Effect of Swedish Speed Limits on Accidents
UScereal	Nutritional and Marketing Information on US Cereals
UScrime	The Effect of Punishment Regimes on Crime Rates
VA	Veteran's Administration Lung Cancer Trial
abbey	Determinations of Nickel Content
accdeaths	Accidental Deaths in the US 1973-1978
anorexia	Anorexia Data on Weight Change
bacteria	Presence of Bacteria after Drug Treatments
beav1	Body Temperature Series of Beaver 1
beav2	Body Temperature Series of Beaver 2
biopsy	Biopsy Data on Breast Cancer Patients
birthwt	Risk Factors Associated with Low Infant Birth Weight
cabbages	Data from a cabbage field trial
caith	Colours of Eyes and Hair of People in Caithness
cats	Anatomical Data from Domestic Cats
cement	Heat Evolved by Setting Cements
chem	Copper in Wholemeal Flour
coop	Co-operative Trial in Analytical Chemistry
cpus	Performance of Computer CPUs
crabs	Morphological Measurements on Leptograpsus Crabs

drivers	Deaths of Car Drivers in Great Britain 1969-84
eagles	Foraging Ecology of Bald Eagles
epil	Seizure Counts for Epileptics
farms	Ecological Factors in Farm Management
fgl	Measurements of Forensic Glass Fragments
forbes	Forbes' Data on Boiling Points in the Alps
galaxies	Velocities for 82 Galaxies
gehan	Remission Times of Leukaemia Patients
genotype	Rat Genotype Data
geyser	Old Faithful Geyser Data
gilgais	Line Transect of Soil in Gilgai Territory
hills	Record Times in Scottish Hill Races
housing	Frequency Table from a Copenhagen Housing Conditions Survey
immer	Yields from a Barley Field Trial
leuk	Survival Times and White Blood Counts for Leukaemia Patients
mammals	Brain and Body Weights for 62 Species of Land Mammals
mcycle	Data from a Simulated Motorcycle Accident
menarche	Age of Menarche data
michelson	Michelson's Speed of Light Data
minn38	Minnesota High School Graduates of 1938
motors	Accelerated Life Testing of Motorettes
muscle	Effect of Calcium Chloride on Muscle Contraction in Rat Hearts
nlschools	Eighth-Grade Pupils in the Netherlands
npk	Classical N, P, K Factorial Experiment
npr1	US Naval Petroleum Reserve No. 1 data
oats	Data from an Oats Field Trial
painters	The Painter's Data of de Piles
petrol	N. L. Prater's Petrol Refinery Data
quine	Absenteeism from School in Rural New South Wales
road	Road Accident Deaths in US States
rotifer	Numbers of Rotifers by Fluid Density
ships	Ships Damage Data
shoes	Shoe wear data of Box, Hunter and Hunter
shrimp	Percentage of Shrimp in Shrimp Cocktail
shuttle	Space Shuttle Autolander Problem
snails	Snail Mortality Data
steam	The Saturated Steam Pressure Data
stormer	The Stormer Viscometer Data
survey	Student Survey Data
topo	Spatial Topographic Data
waders	Counts of Waders at 15 Sites in South Africa
whiteside	House Insulation: Whiteside's Data

```
wtloss                  Weight Loss Data from an Obese Patient
```

يلاحظ أن بعض الحزم تحوي مجاميع بيانات ونستطيع معرفة البيانات الموجودة بالأمر **data()** والذي يسرد مجاميع البيانات للحزم المحملة حاليا

```
> data()
Data sets in package 'datasets':

AirPassengers         Monthly Airline Passenger Numbers 1949-1960
BJsales               Sales Data with Leading Indicator
BJsales.lead (BJsales)
                      Sales Data with Leading Indicator
BOD                   Biochemical Oxygen Demand
CO2                   Carbon Dioxide uptake in grass plants
ChickWeight           Weight versus age of chicks on different
                      diets
DNase                 Elisa assay of DNase
EuStockMarkets        Daily Closing Prices of Major European
Stock
                      Indices, 1991-1998
Formaldehyde          Determination of Formaldehyde
HairEyeColor          Hair and Eye Color of Statistics Students
Harman23.cor          Harman Example 2.3
Harman74.cor          Harman Example 7.4
Indometh              Pharmacokinetics of Indomethicin
InsectSprays          Effectiveness of Insect Sprays
JohnsonJohnson        Quarterly Earnings per Johnson & Johnson
                      Share
LakeHuron             Level of Lake Huron 1875-1972
LifeCycleSavings      Intercountry Life-Cycle Savings Data
Loblolly              Growth of Loblolly pine trees
Nile                  Flow of the River Nile
Orange                Growth of Orange Trees
OrchardSprays         Potency of Orchard Sprays
PlantGrowth           Results from an Experiment on Plant Growth
Puromycin             Reaction velocity of an enzymatic reaction
Seatbelts             Road Casualties in Great Britain 1969-84
Theoph                Pharmacokinetics of theophylline
Titanic               Survival of passengers on the Titanic
ToothGrowth           The Effect of Vitamin C on Tooth Growth in
                      Guinea Pigs
UCBAdmissions         Student Admissions at UC Berkeley
UKDriverDeaths        Road Casualties in Great Britain 1969-84
UKgas                 UK Quarterly Gas Consumption
USAccDeaths           Accidental Deaths in the US 1973-1978
USArrests             Violent Crime Rates by US State
USJudgeRatings        Lawyers' Ratings of State Judges in the US
                      Superior Court
USPersonalExpenditure
                      Personal Expenditure Data
VADeaths              Death Rates in Virginia (1940)
```

```
WWWusage              Internet Usage per Minute
WorldPhones           The World's Telephones
ability.cov           Ability and Intelligence Tests
airmiles              Passenger Miles on Commercial US Airlines,
                      1937-1960
airquality            New York Air Quality Measurements
anscombe              Anscombe's Quartet of "Identical" Simple
                      Linear Regressions
attenu                The Joyner-Boore Attenuation Data
attitude              The Chatterjee-Price Attitude Data
austres               Quarterly Time Series of the Number of
                      Australian Residents
beaver1 (beavers)     Body Temperature Series of Two Beavers
beaver2 (beavers)     Body Temperature Series of Two Beavers
cars                  Speed and Stopping Distances of Cars
chickwts              Chicken Weights by Feed Type
co2                   Mauna Loa Atmospheric CO2 Concentration
crimtab               Student's 3000 Criminals Data
discoveries           Yearly Numbers of Important Discoveries
esoph                 Smoking, Alcohol and (O)esophageal Cancer
euro                  Conversion Rates of Euro Currencies
euro.cross (euro)     Conversion Rates of Euro Currencies
eurodist              Distances Between European Cities
faithful              Old Faithful Geyser Data
fdeaths (UKLungDeaths)
                      Monthly Deaths from Lung Diseases in the UK
freeny                Freeny's Revenue Data
freeny.x (freeny)     Freeny's Revenue Data
freeny.y (freeny)     Freeny's Revenue Data
infert                Infertility after Spontaneous and Induced
                      Abortion
iris                  Edgar Anderson's Iris Data
iris3                 Edgar Anderson's Iris Data
islands               Areas of the World's Major Landmasses
ldeaths (UKLungDeaths)
                      Monthly Deaths from Lung Diseases in the UK
lh                    Luteinizing Hormone in Blood Samples
longley               Longley's Economic Regression Data
lynx                  Annual Canadian Lynx trappings 1821-1934
mdeaths (UKLungDeaths)
                      Monthly Deaths from Lung Diseases in the UK
morley                Michaelson-Morley Speed of Light Data
mtcars                Motor Trend Car Road Tests
nhtemp                Average Yearly Temperatures in New Haven
nottem                Average Monthly Temperatures at Nottingham,
                      1920-1939
occupationalStatus    Occupational Status of Fathers and their
                      Sons
precip                Annual Precipitation in US Cities
presidents            Quarterly Approval Ratings of US Presidents
pressure              Vapor Pressure of Mercury as a Function of
                      Temperature
```

```
quakes                  Locations of Earthquakes off Fiji
randu                   Random Numbers from Congruential Generator
                        RANDU
rivers                  Lengths of Major North American Rivers
rock                    Measurements on Petroleum Rock Samples
sleep                   Student's Sleep Data
stack.loss (stackloss)
                        Brownlee's Stack Loss Plant Data
stack.x (stackloss)     Brownlee's Stack Loss Plant Data
stackloss               Brownlee's Stack Loss Plant Data
state.abb (state)       US State Facts and Figures
state.area (state)      US State Facts and Figures
state.center (state)
                        US State Facts and Figures
state.division (state)
                        US State Facts and Figures
state.name (state)      US State Facts and Figures
state.region (state)
                        US State Facts and Figures
state.x77 (state)       US State Facts and Figures
sunspot.month           Monthly Sunspot Data, 1749-1997
sunspot.year            Yearly Sunspot Data, 1700-1988
sunspots                Monthly Sunspot Numbers, 1749-1983
swiss                   Swiss Fertility and Socioeconomic
Indicators
                        (1888) Data
treering                Yearly Treering Data, -6000-1979
trees                   Girth, Height and Volume for Black Cherry
                        Trees
uspop                   Populations Recorded by the US Census
volcano                 Topographic Information on Auckland's
Maunga
                        Whau Volcano
warpbreaks              The Number of Breaks in Yarn during Weaving
women                   Average Heights and Weights for American
                        Women

Data sets in package 'MASS':

Aids2                   Australian AIDS Survival Data
Animals                 Brain and Body Weights for 28 Species
Boston                  Housing Values in Suburbs of Boston
Cars93                  Data from 93 Cars on Sale in the USA in
1993
Cushings                Diagnostic Tests on Patients with Cushing's
                        Syndrome
DDT                     DDT in Kale
GAGurine                Level of GAG in Urine of Children
Insurance               Numbers of Car Insurance claims
Melanoma                Survival from Malignant Melanoma
OME                     Tests of Auditory Perception in Children
                        with OME
```

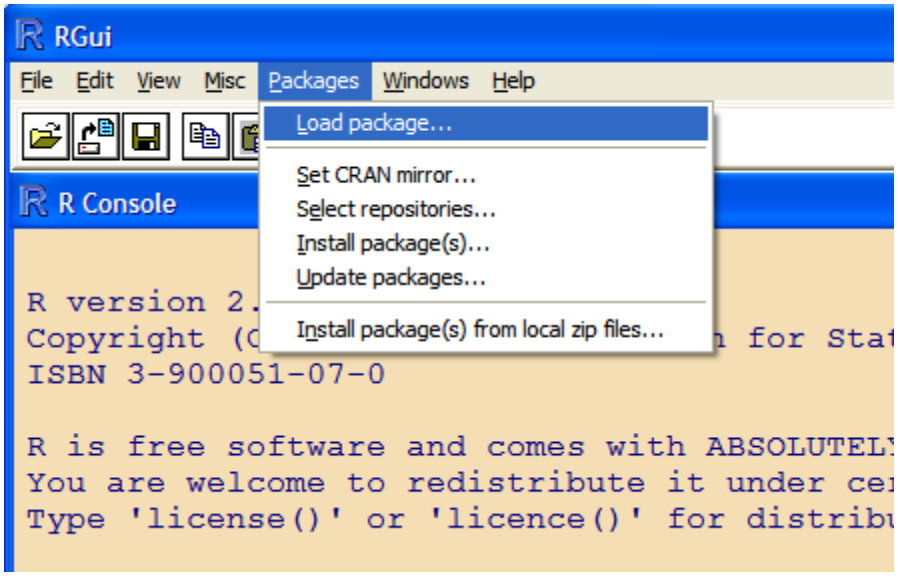
Pima.te	Diabetes in Pima Indian Women
Pima.tr	Diabetes in Pima Indian Women
Pima.tr2	Diabetes in Pima Indian Women
Rabbit	Blood Pressure in Rabbits
Rubber	Accelerated Testing of Tyre Rubber
SP500	Returns of the Standard and Poors 500
Sitka	Growth Curves for Sitka Spruce Trees in 1988
Sitka89	Growth Curves for Sitka Spruce Trees in 1989
Skye	AFM Compositions of Aphyric Skye Lavas
Traffic	Effect of Swedish Speed Limits on Accidents
UScereal	Nutritional and Marketing Information on US Cereals
UScrime	The Effect of Punishment Regimes on Crime Rates
VA	Veteran's Administration Lung Cancer Trial
abbey	Determinations of Nickel Content
accdeaths	Accidental Deaths in the US 1973-1978
anorexia	Anorexia Data on Weight Change
bacteria	Presence of Bacteria after Drug Treatments
beav1	Body Temperature Series of Beaver 1
beav2	Body Temperature Series of Beaver 2
biopsy	Biopsy Data on Breast Cancer Patients
birthwt	Risk Factors Associated with Low Infant Birth Weight
cabbages	Data from a cabbage field trial
caith	Colours of Eyes and Hair of People in Caithness
cats	Anatomical Data from Domestic Cats
cement	Heat Evolved by Setting Cements
chem	Copper in Wholemeal Flour
coop	Co-operative Trial in Analytical Chemistry
cpus	Performance of Computer CPUs
crabs	Morphological Measurements on Leptograpsus Crabs
deaths	Monthly Deaths from Lung Diseases in the UK
drivers	Deaths of Car Drivers in Great Britain 1969-84
eagles	Foraging Ecology of Bald Eagles
epil	Seizure Counts for Epileptics
farms	Ecological Factors in Farm Management
fgl	Measurements of Forensic Glass Fragments
forbes	Forbes' Data on Boiling Points in the Alps
galaxies	Velocities for 82 Galaxies
gehan	Remission Times of Leukaemia Patients
genotype	Rat Genotype Data
geyser	Old Faithful Geyser Data
gilgais	Line Transect of Soil in Gilgai Territory
hills	Record Times in Scottish Hill Races
housing	Frequency Table from a Copenhagen Housing Conditions Survey

```
immer                   Yields from a Barley Field Trial
leuk                    Survival Times and White Blood Counts for
                        Leukaemia Patients
mammals                 Brain and Body Weights for 62 Species of
                        Land Mammals
mcycle                  Data from a Simulated Motorcycle Accident
menarche                Age of Menarche in Warsaw
michelson               Michelson's Speed of Light Data
minn38                  Minnesota High School Graduates of 1938
motors                  Accelerated Life Testing of Motorettes
muscle                  Effect of Calcium Chloride on Muscle
                        Contraction in Rat Hearts
newcomb                 Newcomb's Measurements of the Passage Time
                        of Light
nlschools               Eighth-Grade Pupils in the Netherlands
npk                     Classical N, P, K Factorial Experiment
npr1                    US Naval Petroleum Reserve No. 1 data
oats                    Data from an Oats Field Trial
painters                The Painter's Data of de Piles
petrol                  N. L. Prater's Petrol Refinery Data
phones                  Belgium Phone Calls 1950-1973
quine                   Absenteeism from School in Rural New South
                        Wales
road                    Road Accident Deaths in US States
rotifer                 Numbers of Rotifers by Fluid Density
ships                   Ships Damage Data
shoes                   Shoe wear data of Box, Hunter and Hunter
shrimp                  Percentage of Shrimp in Shrimp Cocktail
shuttle                 Space Shuttle Autolander Problem
snails                  Snail Mortality Data
steam                   The Saturated Steam Pressure Data
stormer                 The Stormer Viscometer Data
survey                  Student Survey Data
synth.te                Synthetic Classification Problem
synth.tr                Synthetic Classification Problem
topo                    Spatial Topographic Data
waders                  Counts of Waders at 15 Sites in South
Africa
whiteside               House Insulation: Whiteside's Data
wtloss                  Weight Loss Data from an Obese Patient


Use 'data(package = .packages(all.available = TRUE))'
to list the data sets in all *available* packages.
```

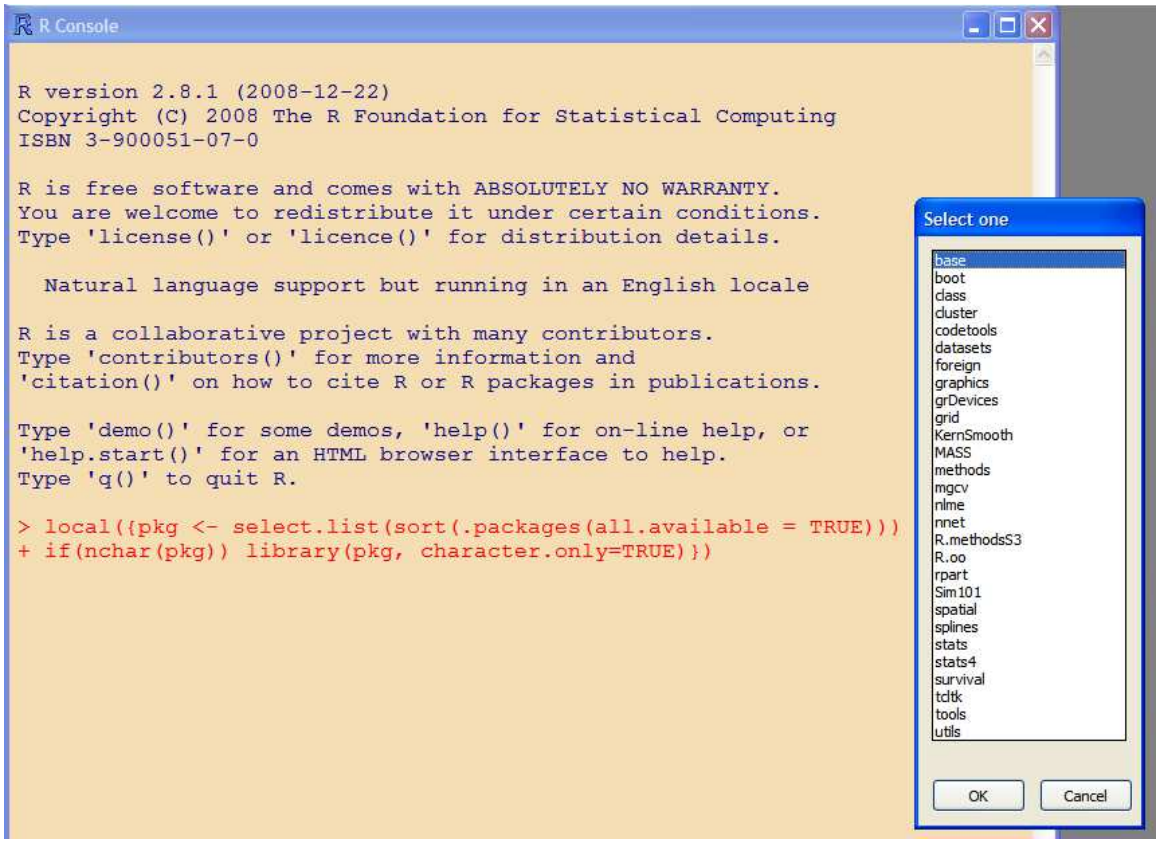
وإضافة مجموعة بيانات إلى حزمة

```
> data(mydata, package=MASS)
```

الحوار التالي لتحميل حزمة



تظهر نافذة تعرض جميع الحزم المتوفرة للإختيار منها

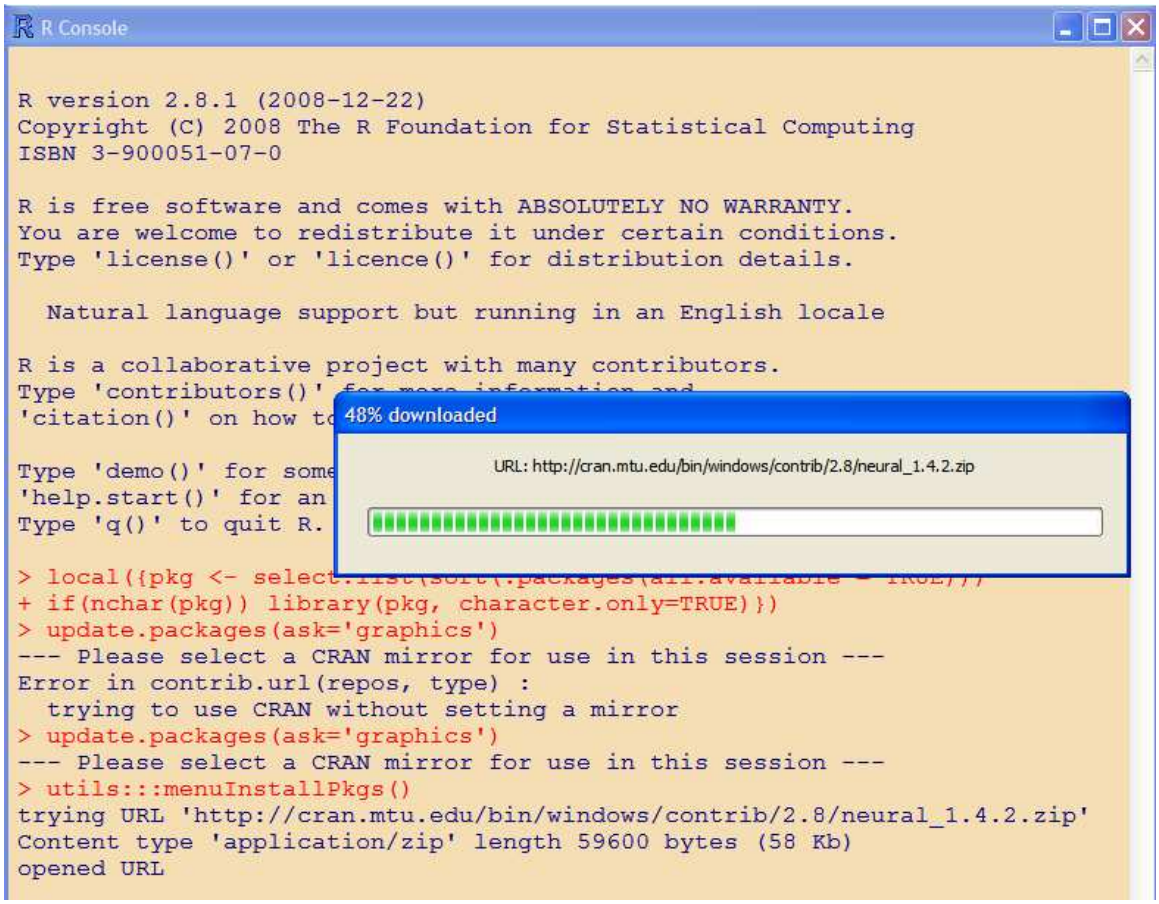



```
> update.packages(ask='graphics')
--- Please select a CRAN mirror for use in this session ---
```

بعد إختيار الحزمة يطلب البرنامج إختيار الجهة التي سيتم التحميل منها

ثم يعرض أسماء الحزم الموجودة والتي تحتاج لتجديد

ثم يقوم البرنامج بإنزال وتجهيز الحزمة




```
> update.packages(ask='graphics')
--- Please select a CRAN mirror for use in this session ---
Warning: package 'VR' is in use and will not be installed
trying URL
'http://cran.mtu.edu/bin/windows/contrib/2.8/cluster_1.11.13.zip'
Content type 'application/zip' length 518967 bytes (506 Kb)
opened URL
downloaded 506 Kb

trying URL
'http://cran.mtu.edu/bin/windows/contrib/2.8/lattice_0.17-22.zip'
Content type 'application/zip' length 926180 bytes (904 Kb)
opened URL
downloaded 904 Kb

trying URL
'http://cran.mtu.edu/bin/windows/contrib/2.8/survival_2.35-4.zip'
Content type 'application/zip' length 2694031 bytes (2.6 Mb)
opened URL
```

```
downloaded 2.6 Mb

package 'cluster' successfully unpacked and MD5 sums checked
package 'lattice' successfully unpacked and MD5 sums checked
Warning: cannot remove prior installation of package 'lattice'
package 'survival' successfully unpacked and MD5 sums checked

The downloaded packages are in
        C:\Documents and Settings\amb\Local
Settings\Temp\RtmptdGcgZ\downloaded_packages
updating HTML package descriptions
>
```

```
> utils:::menuInstallPkgs()
```

```
trying URL
'http://cran.mtu.edu/bin/windows/contrib/2.8/neural_1.4.2.zip'
Content type 'application/zip' length 59600 bytes (58 Kb)
opened URL
downloaded 58 Kb

package 'neural' successfully unpacked and MD5 sums checked

The downloaded packages are in
        C:\Documents and Settings\amb\Local
Settings\Temp\Rtmp1dLn2q\downloaded_packages
updating HTML package descriptions
```

```
> library(neural)
> library(help = neural)
>
```

```
                Information on package 'neural'

Description:

Package:        neural
Version:        1.4.2
Title:          Neural Networks
Author:         Ádám Nagy
Maintainer:     Ádám Nagy <sodzsu@yahoo.com>
Description:    RBF and MLP neural networks with graphical user
                interface
License:        GPL version 2 or later
Built:          R 2.8.1; ; 2008-12-14 01:45:38; windows

Index:

mlp                     MLP neural network recalling method
```

```
mlptrain                MLP neural network training method
rbf                     RBF neural network recalling method
rbftrain                RBF neural network training method
letters_train           dataset for pattern recognizing
letters_out             dataset for pattern recognizing
letters_recall          dataset for pattern recognizing
```

الصفوف والمصفوفات Arrays and matrices:

الصف في R يتكون من متجة بيانات والذي يعطي مكونات مصفوفة ومتجة بعد.
والمصوفة هي صف ذو بعدين. متجه البعد يمتلك عنصرين وهي عدد الأسطر وعدد الأعمدة. يمكن توليد مصفوفة بالدالة (...)arry مع الحجة dim توضع كمتجه ذا طول 2. أو يمكن إستخدام الدالة (...)matrix مع ملاحظة أن الأعمدة تحدد أولا

```
> m1 <- array(1:15,dim = c(3,5))
> m1
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    7   10   13
[2,]    2    5    8   11   14
[3,]    3    6    9   12   15
> m2 <- matrix(1:15,3,5)
> m2
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    7   10   13
[2,]    2    5    8   11   14
[3,]    3    6    9   12   15

> m3 <- 1:15
> dim(m3) <- c(3,5)
> m3
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    7   10   13
[2,]    2    5    8   11   14
[3,]    3    6    9   12   15

> c(class(m1), class(m2), class(m3))
[1] "matrix" "matrix" "matrix"
> array(1:15, dim=c(3,6)) # recycling rule used
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    4    7   10   13    1
```

```
[2,]    2    5    8   11   14    2
[3,]    3    6    9   12   15    3
> array(1:15, dim=c(2,4)) # extra values discarded
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
```

إستخدام العمال الحسابية على صفوف متشابهة سينتج عنه إجراء حسابات عنصر بعنصر وهذا لابأس به في جمع مصفوفة مع أخرى ولكنه غير صحيح في عملية الضرب

```
> m1+m1
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    8   14   20   26
[2,]    4   10   16   22   28
[3,]    6   12   18   24   30
> m1*m1
     [,1] [,2] [,3] [,4] [,5]
[1,]    1   16   49  100  169
[2,]    4   25   64  121  196
[3,]    9   36   81  144  225
```

المصفوفات المتوافقة *conformal matrices* (وهي تلك التي يكون عدد الأعمدة في الأولى يساوي عدد السطور في الثانية) يمكن أن يوجد حاصل ضربها بإستخدام العامل **%*%** . قلب المصفوفة أو عكسها وكذلك حل نظام خطي من المعادلات يمكن أن يتم بإستخدام الدالة **solve(...)** فمثلا لكي نحل النظام $\boldsymbol{ax} = \boldsymbol{b}$ نستطيع أن نستخدم solve(A,b) والدالة solve(A) تعطي مقلوب **A**.

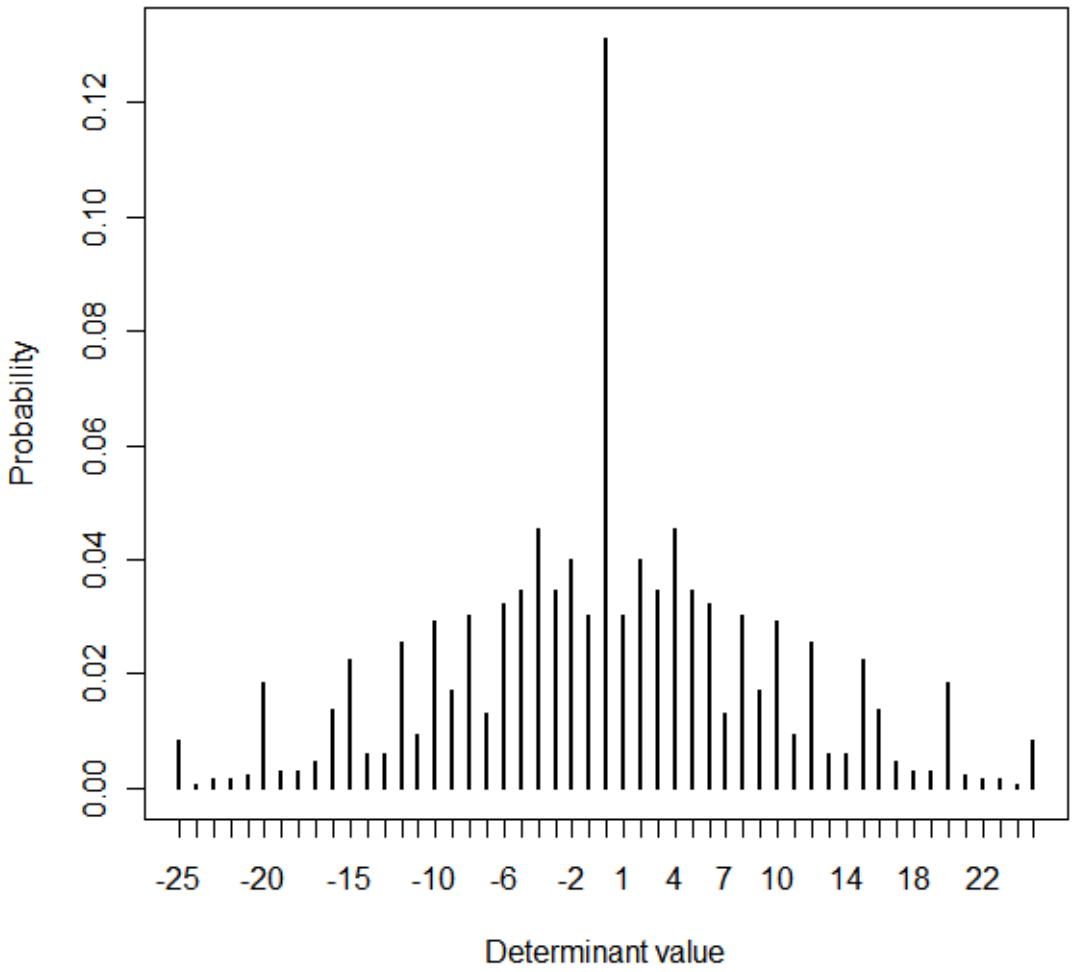
```
> m4 <- array(1:3, c(4,2))
> m4
     [,1] [,2]
[1,]    1    2
[2,]    2    3
[3,]    3    1
[4,]    1    2

> m5 <- array(3:8, c(2,3))
> m5
     [,1] [,2] [,3]
[1,]    3    5    7
```

```
[2,]    4    6    8

> m4 %*% m5
     [,1] [,2] [,3]
[1,]   11   17   23
[2,]   18   28   38
[3,]   13   21   29
[4,]   11   17   23

> m6 <- array(c(1,3,2,1),c(2,2))
> m6
     [,1] [,2]
[1,]    1    2
[2,]    3    1
> v1 <- array(c(1,0), c(2,1))
> v1
     [,1]
[1,]    1
[2,]    0
> solve(m6,v1)
     [,1]
[1,] -0.2
[2,]  0.6

> solve(m6) # inverts m6
     [,1] [,2]
[1,] -0.2  0.4
[2,]  0.6 -0.2

> solve(m6) %*% v1 # does the same as solve(m6,v1)
     [,1]
[1,] -0.2
[2,]  0.6
```

الضرب الخارجي outer product لصفين هو مصفوفة تتولد من كل تراكيب ممكنة لعناصرهم. الحجتين الأولى والثانية في الدلة **outer(...)** هي المتجهين والحجة الثالثة هي العملية المطلوب القيام بها (القيمة الإفتراضية هي الضرب) وهنا نحن نبسط الأمر لأن الضرب الخارجي يعمل على متجهات بأبعاد عامة (وليس متجهات وحسب) والمثال التالي يوضح ذلك. المطلوب توليد دالة الكتلة للمصفوفة 2X2 والتي عناصرها تأتي من

توزيع متساوي متقطع على المجموعة {0,…,5} المحددة هي على الشكل AD - BC حيث A و B و C و D موزعة توزيع متساوي. دالة الكتلة يمكن حسابها عن طريق إحصاء جميع القيم الممكنة الناتجة وتستخدم الدالة **table(…)** التي تكون جدول توزيع تكراري لحجته

```
> m7 <- outer(0:5,0:5) # every possible value of AD
and BC
> m7
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    0
[2,]    0    1    2    3    4    5
[3,]    0    2    4    6    8   10
[4,]    0    3    6    9   12   15
[5,]    0    4    8   12   16   20
[6,]    0    5   10   15   20   25
> freq <- table(outer(m7,m7,"-")) # frequency for
all values of AD-BC
> freq
> freq

-25 -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 -14 -13 -12 -11 -10  -9  -8  -7  -6  -5  -4
 11   1   2   2   3  24   4   4   6  18  29   8   8  33  12  38  22  39  17  42  45  59
 -3  -2  -1   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
 45  52  39 170  39  52  45  59  45  42  17  39  22  38  12  33   8   8  29  18   6   4
 19  20  21  22  23  24  25
  4  24   3   2   2   1  11
> plot(freq/sum(freq), xlab="Determinant value",
+ylab = "Probability")
```



يوجد العديد من العمليات على المصفوفات في R مثل **t(...)** لإيجاد المنقول *transpose* و **nrow(...)** وتعطي عدد الأسطر و **ncol(...)** تعطي عدد الأعمدة والدالة **rbind(...)** و **cbind(...)** تربط الأسطر او الأعمدة معا. والدالة **eigen(...)** لإيجاد الجذور والمتجهات المميزة *eigenvectors and eigenvalues* للمصفوفة وتقوم بتفكييك القيمة الشاذة *singular value decomposition* مستخدمين الدالة **svd(...)** وهكذا.

إدخال بيانات من الإنترنت Data from the Internet:

نستخدم **read.table()** لجلب وإدخال بيانات من الإنترنت كالتالي:

```
> data <-read.table("http://www.stat.ucla.edu/~vlew
/stat130a/datasets/twins.csv",header=TRUE,sep=",")
```

حيث **header=TRUE** تخبر R بأن البيانات تشمل أسماء المتغيرات في أول سطر.
و **sep=","** تخبر R أن البيانات مفصولة بواسطة فاصلة (" , ") .

Quick R

دليل العمل:

Working directory:

```
setwd("../scripts")
getwd()
```

المساعدة:

Help:

```
help(x)
?x
help.search("x")
help.start()
demo()
demo(graphics)
```

تثبيت وتحميل الحزم:

Installing and loading the package:

```
install.packages("spuRs")
library(spuRs)
```

العمليات الحسابية:

Arithmetic:

```
(1 + 1/100)^100
17%%5
17%/%5
exp(1)
options(digits = 16)
exp(1)
pi
sin(pi/6)
```

المتغيرات:

Variables:

```
x <- 100
x
(1 + 1/x)^x
x <- 200
(1 + 1/x)^x
(y <- (1 + 1/x)^x)
n <- 1
n <- n + 1
n
```

الدوال:

Functions:

```
seq(from = 1, to = 9, by = 2)
seq(from = 1, to = 9)
seq(1, 9, 2)
seq(to = 9, from = 1)
seq(by = -2, 9, 1)
x <- 9
seq(1, x, x/3)
```

المتجهات:

Vectors:

```
(x <- seq(1, 20, by = 2))
(y <- rep(3, 4))
(z <- c(y, x))
(x <- 100:110)
i <- c(1, 3, 2)
x[i]
j <- c(-1, -2, -3)
x[j]
x <- c()
length(x)
```

العمليات الجبرية على المتجهات:

Algebraic operations on vectors:

```
x <- c(1, 2, 3)
y <- c(4, 5, 6)
x * y
x + y
y^x
c(1, 2, 3, 4) + c(1, 2)
(1:10)^c(1, 2)
2 + c(1, 2, 3)
2 * c(1, 2, 3)
(1:10)^2
sqrt(1:6)
mean(1:6)
sort(c(5, 1, 3, 4, 2))
```

مثال: المتوسط و التباين

Example: mean and variance

```
x <- c(1.2, 0.9, 0.8, 1, 1.2)
x.mean <- sum(x)/length(x)
x.mean - mean(x)
x.var <- sum((x - x.mean)^2)/(length(x) - 1)
x.var - var(x)
```

مثال: تكامل عددي بسيط

Example: simple numerical integration

```
dt <- 0.005
t <- seq(0, pi/6, by = dt)
ft <- cos(t)
(I <- sum(ft) * dt)
I - sin(pi/6)
```

مثال: النهايات الاسية

Example: exponential limit

```
x <- seq(10, 200, by = 10)
y <- (1 + 1/x)^x
plot(x, y)
```

القيم المفقودة:

Missing data:

```
a <- NA # assign NA to variable A
is.na(a) # is it missing?
a <- c(11,NA,13) # now try a vector
is.na(a) # is it missing?
mean(a) # NAs can propagate
mean(a, na.rm = TRUE) # NAs can be removed
```

المصفوفات:

Matrices:

```
(A <- matrix(1:6, nrow = 2, ncol = 3, byrow = TRUE))
A[1, 3] <- 0
A[, 2:3]
(B <- diag(c(1, 2, 3)))
(A <- matrix(c(3, 5, 2, 3), nrow = 2, ncol = 2))
(B <- matrix(c(1, 1, 0, 1), nrow = 2, ncol = 2))
A %*% B
A * B
(A.inv <- solve(A))
A %*% A.inv
A^(-1)
A[2]


is.matrix(x)
is.vector(x)
A <- as.matrix(x)
as.vector(A)
```

فضاء العمل:

The workspace:

```
ls()
objects()
rm(x)
rm(list = ls())
save.image(file = "fname")
save(x, y, file = "fname")
```

```
load(file = "fname")
savehistory(file = "fname")
loadhistory(file = "fname")
```

برمجة أساسية:

Basic programming:

```
# find the zeros of a2*x^2 + a1*x + a0 = 0
# clear the workspace
rm(list=ls())
# input
a2 <- 1
a1 <- 4
a0 <- 2
# calculation
root1 <- (-a1 + sqrt(a1^2 - 4*a2*a0))/(2*a2)
root2 <- (-a1 - sqrt(a1^2 - 4*a2*a0))/(2*a2)
# output
show(c(root1, root2))
```

التشعب بواسطة if:

Branching with if:

```
if (logical_expression) {
expression_1
...
}
if (logical_expression) {
expression_1
...
} else {
expression_2
...
}
```

مثال: جذور المعادلة التربيعية

Example: roots of a quadratic

```
# find the zeros of a2*x^2 + a1*x + a0 = 0
# clear the workspace
rm(list=ls())
# input
a2 <- 1
a1 <- 4
a0 <- 5
# calculate the discriminant
discrim <- a1^2 - 4*a2*a0
# calculate the roots depending on the value of the
discriminant
```

```
if (discrim > 0) {
roots <- c( (-a1 + sqrt(a1^2 - 4*a2*a0))/(2*a2),
(-a1 - sqrt(a1^2 - 4*a2*a0))/(2*a2) )
} else {
if (discrim == 0) {
roots <- -a1/(2*a2)
} else {
roots <- c()
}
}
# output
show(roots)
```

الدوران بواسطة for:

Looping with for:

```
for (x in vector) {
expression_1
...
}
```

مثال: جمع عناصر متجه

Example: summing a vector

```
(x_list <- seq(1, 9, by = 2))
sum_x <- 0
for (x in x_list) {
+ sum_x <- sum_x + x
+ cat("The current loop element is", x, "\n")
+ cat("The cumulative total is", sum_x, "\n")
+ }
sum(x_list)
```

مضروب n:

Example: n factorial

```
# Calculate n factorial
# clear the workspace
rm(list=ls())
# Input
n <- 6
# Calculation
n_factorial <- 1
for (i in 1:n) {
n_factorial <- n_factorial * i
}
# Output
```

```
show(n_factorial)
```

الدران بواسطة while:

Looping with while:

```
while (logical_expression) {
expression_1
...
}
```

مثال: ارقام فيبونتشي

Example: Fibonacci numbers

لنعتبر أرقام فيبونتشي ., . . ,F_1, F_2 والتي تعرف تكرايا من القاعدة

$F_1 = 1,$
$F_2 = 1,$
$F_n = F_{n-1} + F_{n-2}$ for $n \geq 2.$

البرنامج:

```
# calculate the first Fibonacci number greater than 100
# clear the workspace
rm(list=ls())
# initialise variables
F <- c(1, 1) # list of Fibonacci numbers
n <- 2 # length of F
# iteratively calculate new Fibonacci numbers
while (F[n] <= 100) {
# cat("n =", n, " F[n] =", F[n], "\n")
n <- n + 1
F[n] <- F[n-1] + F[n-2]
}
# output
cat("The first Fibonacci number > 100 is F(", n, ") =",
F[n], "\n")
F
```

البرمجة المعتمدة على المتجه:

Vector-based programming:

مثال: جمع مربعات أول n عدد بإستخدام الدوران

Example: sum of the first n squares using a loop

```
n <- 100
S <- 0
for (i in 1:n) {
+ S <- S + i^2
+ }
```

S

مثال: جمع مربعات أول n عدد بإستخدام العمليات المتجهية

Example: sum of the first n squares using vector operations:

```
sum((1:n)^2)

ifelse:
ifelse(test, A, B)

x <- c(-2, -1, 1, 2)
ifelse(x > 0, "Positive", "Negative")
```

pmin and pmax:

وهي الشكل المتجهي للدوال min و max

```
pmin(c(1, 2, 3), c(3, 2, 1), c(2, 2, 2))
```

الإدخال من ملف:

Input from a file:

```
list.files(path = "dir.name")
scan(file = "", what = 0, n = -1, sep = "", skip = 0, quiet
= FALSE)
```

مثال: إدخال من ملف

Example: file input

```
# Calculate median and quartiles.
# Clear the workspace
rm(list=ls())
# Input
# We assume that the file file_name consists of numeric
values
# separated by spaces and/or newlines
file_name = "../data/data1.txt"
# Read from file
data <- scan(file = file_name)
# Calculations
n <- length(data)
data.sort <- sort(data)
data.1qrt <- data.sort[ceiling(n/4)]
data.med <- data.sort[ceiling(n/2)]
data.3qrt <- data.sort[ceiling(3*n/4)]
```

```
# Output
cat("1st Quartile:", data.1qrt, "\n")
cat("Median: ", data.med, "\n")
cat("3rd Quartile:", data.3qrt, "\n")
```

OR:

```
quantile(scan("../data/data1.txt"), (0:4)/4)
```

إدخال من لوحة المفاتيح:

Input from the keyboard:

مثال: جذور المعادلة التربيعية

Example: roots of a quadratic

```
# find the zeros of a2*x^2 + a1*x + a0 = 0
# clear the workspace
rm(list=ls())
# input
cat("find the zeros of a2*x^2 + a1*x + a0 = 0\n")
a2 <- as.numeric(readline("a2 = "))
a1 <- as.numeric(readline("a1 = "))
a0 <- as.numeric(readline("a0 = "))
# calculate the discriminant
discrim <- a1^2 - 4*a2*a0
# calculate the roots depending on the value of the
discriminant
if (discrim > 0) {
roots <- (-a1 + c(1,-1) * sqrt(a1^2 - 4*a2*a0))/(2*a2)
} else {
if (discrim == 0) {
roots <- -a1/(2*a2)
} else {
roots <- c()
}
}
# output
if (length(roots) == 0) {
cat("no roots\n")
} else if (length(roots) == 1) {
cat("single root at", roots, "\n")
} else {
cat("roots at", roots[1], "and", roots[2], "\n")
}
```

الإخراج لملف:

Output to a file:

```
write(x, file = "data", ncolumns = if(is.character(x)) 1
else 5,
append = FALSE)

(x <- matrix(1:24, nrow = 4, ncol = 6))
write(t(x), file = "../results/out.txt", ncolumns = 6)

cat(..., file = "", sep = " ", append = FALSE)
```

```
dump("x", file = "../results/x.txt")
```

الرسم:

Plotting:

```
plot(x, y, type)
```

مثال:

Example:

```
x <- seq(0, 5, by = 0.01)
y.upper <- 2 * sqrt(x)
y.lower <- -2 * sqrt(x)
y.max <- max(y.upper)
y.min <- min(y.lower)
plot(c(-2, 5), c(y.min, y.max), type = "n", xlab = "x",
+ ylab = "y")
lines(x, y.upper)
lines(x, y.lower)
abline(v = -1)
points(1, 0)
text(1, 0, "focus (1, 0)", pos = 4)
text(-1, y.min, "directrix x = -1", pos = 4)
title("The parabola y^2 = 4*x")
```

عدة رسومات:

Multiple graphs:

```
par(mfrow = c(nr, nc))
par(mfcol = c(nr, nc))
```

مثال:

example:

```
par(mfrow = c(2, 2))
curve(x * sin(x), from = 0, to = 100, n = 1001)
curve(x * sin(x), from = 0, to = 10, n = 1001)
curve(x * sin(x), from = 0, to = 1, n = 1001)
curve(x * sin(x), from = 0, to = 0.1, n = 1001)
par(mfrow = c(1, 1))
```

البرمجة بالدوال:

Programming with functions:

الدوال:

Functions:

الدالة لها الشكل:

```
name <- function(argument_1, argument_2, ...) {
```

```
expression_1
expression_2
...
return(output)
}
```

لإستخدام (نداء) أو إجراء دالة

```
name(x1, x2, ...)
```

مثال: جذور المعادلة التربيعية

Example: roots of a quadratic

```
quad3 <- function(a0, a1, a2) {
# find the zeros of a2*x^2 + a1*x + a0 = 0
if (a2 == 0 && a1 == 0 && a0 == 0) {
roots <- NA
} else if (a2 == 0 && a1 == 0) {
roots <- NULL
} else if (a2 == 0) {
roots <- -a0/a1
} else {
# calculate the discriminant
discrim <- a1^2 - 4*a2*a0
# calculate the roots depending on the value of the
discriminant
if (discrim > 0) {
roots <- (-a1 + c(1,-1) * sqrt(a1^2 - 4*a2*a0))/(2*a2)
} else if (discrim == 0) {
roots <- -a1/(2*a2)
} else {
roots <- NULL
}
}
return(roots)
}
```

المخرج:

output:

```
rm(list = ls())
quad3(1, 0, -1)
quad3(1, -2, 1)
quad3(1, 1, 1)
```

مثال: n إختيار r (توافيق)

Example: n choose r

```
n_factorial <- function(n) {
```

```
# Calculate n factorial
n_fact <- prod(1:n)
return(n_fact)
}
n_choose_r <- function(n, r) {
# Calculate n choose r
n_ch_r <- n_factorial(n)/n_factorial(r)/n_factorial(n-r)
return(n_ch_r)
}
```

المخرج:

output:

```
rm(list = ls())
n_choose_r(4, 2)
n_choose_r(6, 4)
```

الصعيد:

Scope:

داخل دالة تتواجد فقط ضمن تلك الدالة Variable و المتغيرات Argumentsالحجج داخل دالة فهذا المتغير لايكون له وجود خارج xأي اننا لو عرفنا وأستخدمنا المتغير تلك الدالة. إذا وجدت متغيرات لها نفس الإسم داخل وخارج دالة فهذان المتغيرين مختلفان تماما وليس بينهما أي صلة.

مثال توضيحي:

Illustration:

```
test <- function(x) {
+ y <- x + 1
+ return(y)
+ }
test(1)

[1] 2

x

Error: Object "x" not found

y

Error: Object "y" not found

y <- 10
test(1)
```

```
[1] 2

y

[1] 10
```

جزء البرنامج والذي يكون متغير معرف ضمنه يسمى صعيده The variable scope المتغيرات المعرفة داخل دالة ليمكن رؤيتها خارج الدالة و العكس صحيح.

مثال:

```
test2 <- function(x) {
+ y <- x + z
+ return(y)
+ }
z <- 1
test2(1)

[1] 2

z <- 2
test2(1)

[1] 3
```

الحجج الإختيارية والقيم الإفتراضية:

Optional arguments and default values:

لإعطاء الحجة argument_1 القيمة الإفتراضية x1 نستخدم argument_1 = x1 داخل معرف الدالة.

R يسند حجج للمتغيرات من اليسار إلا إذا سميت الحجة.

مثال توضيحي:

Illustration:

```
test3 <- function(x = 1, y = 1, z = 1) {
+ return(x * 100 + y * 10 + z)
+ }
test3(2, 2)

[1] 221

test3(y = 2, z = 2)

[1] 122
```

البرمجة المتجهية بإستخدام الدوال:

Vector-based programming using functions:

كثير من دوال R متجهية Vectorised Functions أي إذا اعطي مدخل متجهي فإن الدالة تعمل على كل عنصر منفصلا في المتجة ويعاد مخرج متجهي. ويعد هذا صفة قوية لـ R والتي تسمح بوضع برنامج مدمج وفعال وسهل القرائة.

لتسهيل البرمجة المتجهية يوجد في R عائلة من الدوال القوية والمرنة والتي تمكن من توجيه Vectorisation الدوال المعرفة بالمستخدم:

apply, sapply, lapply, tapply, and mapply.

تأثير **sapply(X, FUN)** تطبيق الدالة **FUN** على كل عنصر من عناصر المتجه **X** أي أن **sapply(X, FUN)** تعيد متجه يكون عنصره رقم **i** قيمة التعبير **FUN(X[i])** إذا كان للدالة **FUN** حجج اخرى غير **X[i]** فإنه بالإمكان شملها بإستخدام **sapply(X, FUN, ...)** والتي تعيد **FUN(X[i], ...)** كعنصرها رقم **i**.لتتجيه أكثر من حجة واحدة نستخدم **mapply**.

إذا اردنا نطبيق دالة تأخذ حجج متجهية لكل من الأسطر (أو الأعمدة) لمصفوفة عندا نستخدم الدالة **apply**.

مثال: كثافة الأعداد الأولية

Example: density of primes

سوف نكتب دالة والتي تختبر إذا كان عدد صحيح معطى أولي أو لا. ثم نستخدم **sapply** لنطبق الدالة على المتجه **2:n** وبهذا نعرف كل الأعداد الأولية التي أقل من أو تساوي **n** .

سوف نستخدم الدالة **cumsum(x)** والتي تعطي المجوع التراكمي لـ **x** كمتجه.

سوف نطبقه على متجه منطقي يحوي قيم **TRUE/FALSE** والذي تجبره R ليصبح متجه 1/0 (أي R coerces into a 1/0 vector) قبل حساب المجمع التراكمي.

```
# estimate the density of primes (using a very inefficient
algorithm)
# clear the workspace
rm(list=ls())
prime <- function(n) {
# returns TRUE if n is prime
# assumes n is a positive integer
if (n == 1) {
is.prime <- FALSE
} else if (n == 2) {
is.prime <- TRUE
} else {
is.prime <- TRUE
```

```
for (m in 2:(n/2)) {
if (n %% m == 0) is.prime <- FALSE
}
}
return(is.prime)
}
# input
# we consider primes <= n
n <- 1000
# calculate the number of primes <= m for m in 2:n
# num.primes[i] == number of primes <= i+1
m.vec <- 2:n
primes <- sapply(m.vec, prime)
num.primes <- cumsum(primes)
# output
# plot the actual prime density against the theoretical
limit
par(mfrow = c(1, 2))
plot(m.vec, num.primes/m.vec, type = "l",
main = "prime density", xlab = "n", ylab = "")
lines(m.vec, 1/log(m.vec), col = "red")
plot(m.vec, num.primes/m.vec*log(m.vec), type = "l",
main = "prime density * log(n)", xlab = "n", ylab = "")
par(mfrow = c(1, 1))


prime <- function(n) {
# returns TRUE if n is prime
# assumes n is a positive integer
if (n == 1) {
is.prime <- FALSE
} else if (n == 2) {
is.prime <- TRUE
} else {
is.prime <- TRUE
m <- 2
m.max <- sqrt(n) # only want to calculate this once
while (is.prime && m <= m.max) {
if (n %% m == 0) is.prime <- FALSE
m <- m + 1
}
}
return(is.prime)
}
```

البرمجة التكرارية:

Recursive programming:

البرنامج التكراري هو ببساطة البرنامج الذي يستخدم نفسه. وهذه الخاصية مفيدة لأن الكثير من الخوارزمات تكرارية بطبيعتها.

مثال: مضروب n

Example: n factorial

```
nfact2 <- function(n) {
# calculate n factorial
if (n == 1) {
cat("called nfact2(1)\n")
return(1)
} else {
cat("called nfact2(", n, ")\n", sep = "")
return(n*nfact2(n-1))
 }
 }
nfact2(6)

called nfact2(6)
called nfact2(5)
called nfact2(4)
called nfact2(3)
called nfact2(2)
called nfact2(1)

[1] 720
```

مثال: منخل (مصفاة أو فلتر) إراتوسثيين

Example: Sieve of Eratosthenes

منخل إراتوسثيين هو عبارة عن خوارزم لإيجاد كل الأعداد الأولية التي هي أقل من أو تساوي عدد معطى n.

```
primesieve <- function(sieved, unsieved) {
# finds primes using the Sieve of Eratosthenes
# sieved: sorted vector of sieved numbers
# unsieved: sorted vector of unsieved numbers
# cat("sieved", sieved, "\n")
# cat("unsieved", unsieved, "\n")
p <- unsieved[1]
n <- unsieved[length(unsieved)]
if (p^2 > n) {
return(c(sieved, unsieved))
} else {
unsieved <- unsieved[unsieved %% p != 0]
sieved <- c(sieved, p)
```

```
return(primesieve(sieved, unsieved))
}
}
```

output:

```
rm(list = ls())
primesieve(c(), 2:200)
```

تراكيب البيانات المطورة:

Sophisticated data structures:

العوامل:

Factors:

مثال توضيحي:

Illustration:

```
hair <- c("blond", "black", "brown", "brown", "black",
"gray","none")
is.character(hair)

[1] TRUE

is.factor(hair)

[1] FALSE

hair <- factor(hair)
levels(hair)

[1] "black" "blond" "brown" "gray" "none"

hair <- factor(hair, levels = c("black", "gray", "brown",
+ "blond", "white", "none"))
table(hair)
hair
black gray brown blond white none
2     1    2     1     0     1

hair

[1] blond black brown brown black gray none
Levels: black gray brown blond white none

as.vector(hair)
```

```
[1] "blond" "black" "brown" "brown" "black" "gray" "none"

as.numeric(hair)

[1] 4 1 3 3 1 2 6

c(hair, 5)

[1] 4 1 3 3 1 2 6 5

x <- factor(c(0.8, 1.1, 0.7, 1.4, 1.4, 0.9))
as.numeric(x) # does not recover x

[1] 2 4 1 5 5 3

as.numeric(levels(x))[x] # does recover x

[1] 0.8 1.1 0.7 1.4 1.4 0.9

as.numeric(as.character(x)) # does recover x

[1] 0.8 1.1 0.7 1.4 1.4 0.9

table(hair[hair == "gray" | hair == "none"])

black gray brown blond white none
0     1    0     0     0     1

table(hair[hair == "gray" | hair == "none", drop = TRUE])

gray none
1    1
```

مثال توضيحي:

Illustration:

```
phys.act <- c("L", "H", "H", "L", "M", "M")
phys.act <- factor(phys.act, levels = c("L", "M", "H"),
+ ordered = TRUE)
is.ordered(phys.act)

[1] TRUE

phys.act[2] > phys.act[1]

[1] TRUE
```

```
phys.act <- factor(phys.act, levels = c("L", "M", "H"),
+ labels = c("Low", "Medium", "High"), ordered = TRUE)
table(phys.act)
phys.act
Low Medium High
2   2      2
which(phys.act == "High")

[1] 2 3
```

أطر البيانات:

Dataframes:

القوائم Lists و أطر البيانات **Dataframes** صممت لكي تخزن تراكيب بيانات أكثير تعقيدا من المصفوفات.
إطار البيانات عبارة عن قائمة فصلت خصيصا لمواجهة الحاجة العملية لتمثيل مجاميع بيانات متعددة المتغيرات Multivariate Datasets. وهي قائمة من متجهات مقيدة بشرط أن يكون لها نفس الطول.
كل متجه (أو عمود) يعود إلى أو يمثل متغير في البيانات وكل سطر يعود أو يمثل مشاهدة واحدة من البيانات. وكل متجه يمكن أن يكون أي من الأطوار الأساسية للشيئ (basic modes of object).
اطر البيانات الضخمة تقرأ عادة في R من ملف بإستخدام الدالة **read.table** والتي لها الشكل التالي:

```
read.table(file, header = FALSE, sep = "")
```

الدالة **read.table** ترجع أو تعيد إطار بيانات. يوجد شكلين آخرين للدالة **read.table** هما **read.csv(file)** للبيانات المفصولة بفاصلة ولها الشكل:

```
read.table(file, header = TRUE, sep = ",").
```

و الدالة **read.delim(file)** وهي لقرائة بيانات مفصولة بمؤشر الجدولة (Tab) ولها الشكل:

```
read.delim (file, header = TRUE, sep = "\t").
```

مثال:

Example:

```
ufc <- read.csv("../data/ufc.csv")
head(ufc)
     plot tree species dbh.cm height.m
1    2    1    DF      39     20.5
2    2    2    WL      48     33.0
3    3    2    GF      52     30.0
4    3    5    WC      36     20.7
5    3    8    WC      38     22.5
6    4    1    WC      46     18.0
tail(ufc)
    plot tree species dbh.cm height.m
331 143  1    GF      28.0   21.0
332 143  2    GF      33.0   20.5
333 143  7    WC      47.8   20.5
334 144  1    GF      10.2   16.0
335 144  2    DF      31.5   22.0
336 144  4    WL      26.5   25.0
```

كل عمود أو متغير في إطار البيانات له إسم وحيد ونستطيع إستخراج هذا المتغير بواسطة إسم إطار البيانات وإسم العمود وبينهم علامة الدولار أي

```
x <- ufc$height.m
x[1:5]
[1] 20.5 33.0 30.0 20.7 22.5
```

الأشكال **ufc[[5]]**, **ufc$height.m** و **ufc[["height.m"]]** متكافئة

ويمكن إستخراج عناصر من إطار البيانات مباشرة بإستخدام مؤشر مصفوفي

```
 ufc[1:5, 5]
[1] 20.5 33.0 30.0 20.7 22.5
```

التأشير **ufc[4:5]** مكافئ لـ **ufc[c("dbh.cm", "height.m")].**

```
diam.height <- ufc[4:5]
diam.height[1:5, ]
dbh.cm height.m
1 39 20.5
2 48 33.0
3 52 30.0
4 36 20.7
5 38 22.5
is.data.frame(diam.height)
[1] TRUE

mode(ufc)
[1] "list"
mode(ufc[5])
[1] "list"
mode(ufc[[5]])
[1] "numeric"
```

إضافة لقرائة إطار بيانات من ملف فإنه يمكننا تكوين إطار بيانات من مجموعة من المتجهات و/أو اطربيانات موجودة سابقا بإستخدام الدالة **data.frame** والتي لها الشكل:

data.frame(col1 = x1, col2 = x2, ..., df1, df2, ...)

هنا **col1** و **col2** الخ هي اسماء اعمدة (وتكتب كحروف بدون علامات إقتباس " أو ') و **x1** و **x2** الخ متجهات لها نفس الطول و **df1** و **df2** اطر بيانات والتي يجب ان تكون اطوال اعمدتها نفس اطوال **x1** و **x2** .

ايضا يمكننا إضافة متغير داخل اطار بيانات بإعطائه إسم و إسناد قيمة له.

مثال:

```
ufc$volume.m3 <- pi * (ufc$dbh.cm/200)^2 * ufc$height/2
mean(ufc$volume.m3)
[1] 1.93294
```

لاحظ ان:
ufc$volume.m3 و **ufc[6]** و **ufc["volume.m3"]** و **ufc[[6]]** و **ufc[["volume.m3"]]** جميعها متكافئة.
الأمر **names(df)** يعطي أسماء إطار البيانات **df** كمتجه من الحروف. لتغيير الأسماء نعيد إسناد اسماء جديدة لها مثل:

```
(ufc.names <- names(ufc))
[1] "plot" "tree" "species" "dbh.cm" "height.m"
[6] "volume.m3"
names(ufc) <- c("P", "T", "S", "D", "H", "V")
names(ufc)
[1] "P" "T" "S" "D" "H" "V"
names(ufc) <- ufc.names
```

صفات إطار البيانات:

Dataframe attributes:

```
names(df)
dim(df)
row.names(df)
```

الدالة subset:

The function subset:

```
fir.height <- subset(ufc, subset = species %in% c("DF",
"GF"),
+ select = c(plot, tree, height.m))
head(fir.height)
plot tree height.m
1 2 1 20.5
3 3 2 30.0
7 4 2 17.0
8 5 2 29.3
9 5 4 29.0
10 6 1 26.0
```

للمتجهين **x** و **y** والذين لهم نفس الطور التعبير **x %in% y** يعطي متجة منطقي له نفس طول **x** وعنصرة الـ **i** يكون **TRUE** فقط وفقط إذا كان **x[i]** عنصر من عناصر **y**.

لكتابة إطار بيانات في ملف نستخدم:

To write a dataframe to a file we use

```
write.table(x, file = "", append = FALSE, sep = " ",
row.names = TRUE, col.names = TRUE)
```

ويمكن إزالة الأسطر التي تحوي قيم مفقودة بإستخدام الدالة **na.omit**.

التوصيل:

Attaching:

لتوصيل اطار بيانات لفضاء عمل

```
attach(ufc)
max(height.m[species == "GF"])
[1] 47
```

ولفصله

```
height.m <- 0 #vandalism
max(height.m)
[1] 0
max(ufc$height.m)
[1] 47
detach(ufc)
max(ufc$height.m)
[1] 47
```

القوائم:

Lists:

لقد ذكرنا أن المتجة هو مجموعة مفهرسة أو مجدولة Indexed set من الأشياء وجميع عناصره لها نفس النوع (عددي أو حرفي أو منطقي ألخ) والذي سميناة نوع Mode المتجه.

القائمة هي أيضا مجموعة مفهرسة من الأشياء ولكنها بعكس المتجه فإن عناصرها يمكن أن تكون من أنواع مختلفة وقد تضمن قوائم اخرى ونوع القائمة هو قائمة وتستخدم القوائم في R لتجميع وتخزين مخرجات دوال معقدة. لاحظ أن أطر البيانات هي حالة خاصة من القوائم.

تولد الفائمة بإستخدام الأمر list(...) بحجج مفصولة بفواصل.

تستخدم الأقواس المربعة المفردة لإختيار قوائم فرعية أو ثانوية و الأقواس المربعة المضاعفة لإختيار عنصر واحد.

أمثلة:

Examples:

```
> my.list <- list("one", TRUE, 3, c("f", "o", "u", "r"))
> my.list[[2]]
[1] TRUE
> mode(my.list[[2]])
```

```
[1] "logical"
> my.list[2]
[[1]]
[1] TRUE
> mode(my.list[2])
[1] "list"
> my.list[[4]][1]
[1] "f"
> my.list[4][1]
[[1]]
[1] "f" "o" "u" "r"
```

عندما تعرض قائمة تستخدم R أقواس مربعة مضاعفة [[1]] و [[2]] الخ لتبين عناصر القائمة و أقواس مربعة وحيدة [1] و [2] الخ لتبين عناصر المتجه.

```
> my.list <- list(first = "one", second = TRUE, third = 3,
+ fourth = c("f", "o", "u", "r"))
> names(my.list)
[1] "first" "second" "third" "fourth"
> my.list$second
[1] TRUE
> names(my.list) <- c("First element", "Second element",
+ "Third element", "Fourth element")
> my.list$"Second element"
[1] TRUE
> x <- "Second element"
> my.list[[x]]
[1] TRUE
```

لتمهيد قائمة أي تحويلها إلى متجة نستخدم الأمر unlist(x)

```
> x <- list(1, c(2, 3), c(4, 5, 6))
> unlist(x)
[1] 1 2 3 4 5 6
```

إذا كانت القائمة تحوي قوائم داخلها فإن هذه القوائم تمهد أيضا إلا إذا أمرنا بغير ذلك بإستخدام الحجة recursive = FALSE.

```
> lm.xy <- lm(y ~ x, data = data.frame(x = 1:5, y = 1:5))
> mode(lm.xy)
[1] "list"
> names(lm.xy)
[1] "coefficients" "residuals" "effects" "rank"
[5] "fitted.values" "assign" "qr" "df.residual"
[9] "xlevels" "call" "terms" "model"
```

لاحظ هنا لايهمنا كيفية تطبيق الخط المستقيم ولكن نلاحظ أن الدلة lm أعادة قائمة. العنصر الأول المسمى coefficients هو متجة يعطي a و b والثاني المسمى residuals متجه يعطي قيم $y_i - a\ x_i - b$ لجميع قيم i الثالث والمسمى fitted.values متجه يعطي $a\ x_i + b$ وهكذا.

عائلة apply

The apply family

يوجد في R العديد من التقنيات لمعالجة القوائم و اطر البيانات منها عدة دوال والتي تمكن من تطبيق دالة على جميع عناصر (أو مجموعة مختارة) القائمة او اطار البيانات.

الدالة tapply

ولها التركيب التالي:

```
tapply(X, INDEX, FUN, ...),
```

حيث الحجج:

X المتجه الذي ستطبق عليه الدالة.

INDEX عامل factor له نفس طول X والذي يستخدم لتجميع (group) عناصر X.

FUN الدلة المطبقة.

تطبيق دوال على قوائم بإستخدام lapply و sapply

Applying functions to lists lapply and sapply

يمكن إستخدام sapply و apply لتطبيق دالة على متجه أو صف array مثلا يمكن حساب مجاميع الأسط و الأعمدة لمصفوفة.

لتطبيق دالة على قائمة نستخدم إما sapply أو lappy. الدالة lapply(X,FUN,...) تطبق الدالة FUN على جميع عناصر القائمة X وتعطي قائمة. الدالة sapply(X,FUN,...) تطبق الدالة FUN على جميع عناصر X سواء كانت X قائمة أو متجه.

مقدمة للإمكانات الإحصائية في R:

بعض الدوال والأوامر الأولية:

إضافة حزمة

```
install.packages('ggplot2')

install.packages("Rcmdr")
```

لتحميل حزمة للعمل بدوالها

```
library(ggplot2)
```

لتجديد الحزم

```
update.packages()
```

للمساعدة لمعرفة إستخدام دالة

```
help(functionname)
```

طريقتين للمساعدة عن إستخدام mean

```
help(mean)
?mean
```

للمساعدة في معرفة حجج دالة

```
args(functionname)
```

مثال لإستخدام args

```
args(mean)
function (x, ...)
NULL

args(sd)
function (x, na.rm = FALSE)
NULL
```

إعطاء مثال لإستخدام دالة

```
example(functionname)

example(mean)
mean> x <- c(0:10, 50)
mean> xm <- mean(x)
mean> c(xm, mean(x, trim = 0.1))
[1] 8.75 5.50
```

```
mean> mean(USArrests, trim = 0.2)
Murder Assault UrbanPop Rape
7.42 167.60 66.20 20.16
```

اجر هذا الأمر لتوجه لصفحة تعطي كتيبات ومراجع و أشياء اخرى عن R

```
help.start()
```

للبحث عن أمر في مواقع R

```
http://rseek.org
http://stackoverflow.com/
http://stats.stackexchange.com/
RSiteSearch("key phrase")
```

مثلا

```
RSiteSearch("canonical correlation")
```

لمعرفة دليل العمل الحالي

```
getwd()
```

لتحديد دليل عمل

```
setwd("R:\\data")
setwd("R:/data")
```

لسرد محتويات دليل العمل

```
dir()
```

إدخال بيانات:

قرائة بيانات على شكل جدول ووضعها في مجموعة بيانات تسمى dfrm

```
dfrm = read.table("filename.txt")
```

مثال

```
dfrm = read.table("statisticians.txt", sep=" ")
```

تقرأ بيانات من الملف النصي statisticians.txt (مثلا يحوي بيانات رقمية ونصية الخ)

لمعرفة صنف المتغير V1 الموجود في مجموعة البيانات dfrm والذي نستخدمه عن طريق التأشير له dfrm$V1

```
class(dfrm$V1)
```

للقرائة من ملف يحوي السطر أسماء المتغيرات

```
dfrm = read.table("statisticians.txt", header=TRUE)
```

للقرائة من ملف csv

```
tbl = read.csv("filename")
```

للقرائة من ملف csv لايحوي أسماء متغيرات

```
tbl = read.csv("filename", header=FALSE)
```

للقرائة من ملف يختار من مجلد

```
data = read.csv(file.choose(), header = TRUE)
```

للقرائة من ملف على الإنترنت

```
hs0 <-
read.table("http://www.ats.ucla.edu/stat/R/notes/hs
0.csv", header=T, sep=",")
```

قرائة بيانات اخرى من الإنترنت

```
 data =
read.csv("http://wiki.stdout.org/rcookbook/Data
input and output/Loading data from a
file/datafile.csv")

data
```

```
   First    Last  Sex Number
1 Currer    Bell    F      2
2    Dr.   Seuss    M     49
3        Student <NA>     21

data2 =
read.csv("http://wiki.stdout.org/rcookbook/Data
input and output/Loading data from a file/datafile-
noheader.csv", header=FALSE)

data2
      V1      V2   V3 V4
1 Currer    Bell    F  2
2    Dr.   Seuss    M 49
3        Student <NA> 21

names(data2) <- c("First","Last","Sex","Number")
data2
   First    Last  Sex Number
1 Currer    Bell    F      2
2    Dr.   Seuss    M     49
3        Student <NA>     21
>
```

إدخال بيانات من ملفات Excel:

تحمل مكتبة gdata أولا

```
library(gdata)

data = read.xls("data.xls")
```

إدخال بيانات من ملفات SPSS و SAS
تحمل مكتبة foreign أولا

```
library(foreign)
data = read.spss("data.sav", to.data.frame=TRUE)
data = read.sas("data.sas", to.data.frame=TRUE)
```

إدخال بيانات من لوحة المفاتيح أو لوحة النسخ

```
data = read.csv(stdin())
data = read.table(stdin(), header=TRUE)

data = read.csv('clipboard')
data = read.table('clipboard', header=TRUE)
```

مثال:

```
a<-read.csv(stdin())
gender,id,race,ses,schtyp,prgtype,read,write,math,s
cience,socst
0,70,4,1,1,general,57,52,41,47,57
1,121,4,2,1,vocati,68,59,53,63,61
0,86,4,3,1,general,44,33,54,58,31
0,141,4,3,1,vocati,63,44,47,53,56
```

طباعة البيانات

```
print(data, row.names=FALSE)
```

كتابة البيانات إلى لوحة النسخ أو للشاشة

```
write.csv(df, 'clipboard', row.names=FALSE)
write.csv(data, stdout(), row.names=FALSE)
```

تخزين البيانات

```
write.csv(data, "data.csv", row.names=FALSE)
write.table(data, "data.csv", sep="\t",
row.names=FALSE, col.names=FALSE)
write.csv(data, "data.csv", row.names=FALSE, na="")
```

تخزين البيانات في شكل نصي ثم إسترجاعها

```
dump("data", "data.Rdmpd")
source("data.Rdmpd")
```

تخزين البيانات في شكل خاص بـ R ثم إسترجاعها

```
save("data", file="data.RData")
load("data.RData")
```

معالجة المتغيرات

إدخال قيم بشكل متجه (بدون حفظ القيم)

```
c(1,1,2,3,5,8,13,21)
```

إدخال قيم بشكل متجه وحفظها في v1 و v2

```
v1 = c(1,2,3)
v2 = c(4,5,6)
```

ضم المتجهين v1 و v2 (بدون حفظ القيم)

```
c(v1,v2)
```

إدخال بيانات في متجه x

```
x = c(0,1,1,2,3,5,8,13,21,34)
```

إيجاد متوسط x (بدون حفظ القيمة)

```
mean(x)
[1] 8.8
```

إيجاد وسيط x (بدون حفظ القيمة)

```
median(x)
[1] 4
```

إيجاد الإنحراف المعياري لـ x (بدون حفظ القيمة)

```
sd(x)
[1] 11.03328
```

إيجاد تباين x (بدون حفظ القيمة)

```
var(x)
[1] 121.7333
```

إيجاد دالة من x و تحفظ في المتجة y

```
y <- log(x+1)
```

إيجاد ترابط x مع y

```
cor(x,y)
[1] 0.9068053
```

إيجاد تغاير x مع y

```
cov(x,y)
[1] 11.49988
```

متجه جديد x (يستبدل المتجه السابق) يحوي قيمة مفقودة missing value والتي يرمز لها في R بالرمز NA

```
x <- c(0,1,1,2,3,NA)
```

لاحظ أن دالتي المتوسط و الإنحراف المعياري تعطي NA

```
mean(x)
[1] NA
sd(x)
[1] NA
```

بإعطاء الأمر الجانبي na.rm=TRUE أي إستبعاد القيم المفقودة نتحصل على القيم للدوال المطلوبة بعد إزالة القيم المفقودة

```
mean(x, na.rm=TRUE)
[1] 1.4
sd(x, na.rm=TRUE)
[1] 1.140175
```

لمشاهدة البيانات في المجموعة dframe

```
print(dframe)
     small    medium    big
1 0.6739635 10.526448 99.83624
2 1.5524619 9.205156 100.70852
3 0.3250562 11.427756 99.73202
4 1.2143595 8.533180 98.53608
5 1.3107692 9.763317 100.74444
6 2.1739663 9.806662 98.58961
7 1.6187899 9.150245 100.46707
8 0.8872657 10.058465 99.88068
9 1.9170283 9.182330 100.46724
10 0.7767406 7.949692 100.49814
```

حساب إحصائآت المتغيرات في المجموعة dframe

```
summary(dframe)

mean(dframe)
  small   medium   big
1.245040 9.560325 99.946003

sd(dframe)
  small     medium   big
0.5844025 0.9920281 0.8135498

var(dframe)
          small     medium     big
small 0.34152627 -0.21516416 -0.04005275
medium -0.21516416 0.98411974 -0.09253855
big -0.04005275 -0.09253855 0.66186326

cor(dframe)
         small      medium    big
small 1.00000000 -0.3711367 -0.08424345
```

```
medium -0.37113670 1.0000000 -0.11466070
big -0.08424345 -0.1146607 1.00000000

cov(dframe)
         small      medium       big
small 0.34152627 -0.21516416 -0.04005275
medium -0.21516416 0.98411974 -0.09253855
big -0.04005275 -0.09253855 0.66186326
```

مثال:

سوف نستخدم البيانات

```
hs0 <-
read.table("http://www.ats.ucla.edu/stat/R/notes/hs
0.csv", header=T, sep=",")
```

لسرد الأسطر من 1 إلى 20

```
hs0[1:20, ]
```

لسرد أسماء المتغيرات

```
names(hs0)
```

للعمل على متغيرات الملف (البيانات) فقط

```
attach(hs0)
```

لإيقاف العمل على متغيرات الملف

```
detach()

getOption("digits")
options(digits=2)
by(hs0, prgtype, mean, na.rm=T)
by(hs0, prgtype, sd, na.rm=T)
m   <- tapply(write, prgtype, mean)
v   <- tapply(write, prgtype, var)
med <- tapply(write, prgtype, median)
n   <- tapply(write, prgtype, length)
sd  <- tapply(write, prgtype, sd)
cbind(mean=m, var=v, std.dev=sd, median=med, n=n)
hist(write)
# load trellis graphics
library(lattice)
# trellis graphs
histogram(~write, hs0, type="count")
# histogram of write by gender
histogram(~write | gender, hs0, type="count")
```

```
# change the number of bins to 15
hist(write, breaks=15)
```

الجداول التكرارية

```
table(ses)
tab1<-table(gender, ses)
# row proportions
prop.table(tab1,1)
# column proportions
prop.table(tab1,2)
# row frequencies
rowSums(tab1)
# column frequencies
colSums(tab1)
```

الترابط ورسوم الإنتشار

```
# shorthand way of referring to read, write, math,
science
read.sci <- hs0[ , 7:10]
# correlation of a pair of variables
cor(write, math)
cor(write, science)
cor(write, science, use="complete.obs")
# correlation matrix
plot(math, write)
# scatter plot matrix
plot(read.sci)
cor(read.sci, use="complete.obs")
plot(math, write)
> # correlation of a pair of variables
> cor(write, math)
> cor(write, science)
> cor(write, science, use="complete.obs")
> # correlation matrix
> plot(math, write)
> # scatter plot matrix
> # shorthand way of referring to read, write,
math, science
> read.sci <- hs0[ , 7:10]
> plot(read.sci)
> cor(read.sci, use="complete.obs")
> cor(read.sci, use="pairwise.complete.obs")
```

```
> plot(math, write)
> # scatter plot matrix
>  hs1 <-
read.table("http://www.ats.ucla.edu/stat/R/notes/hs
1.csv", header=T, sep=",")
> hs1.read.well <- hs1[hs1$read >= 60, ]
> mean(hs1.read.well$read)
> mean(hs1$read)
> hs2<-hs1[, c("read", "write")]
> # another way of doing the same thing
> hs3<-hs1[, c(7, 8)]
> names(hs3)
> hs2.drop<-hs1[, -c(7, 8)]
> names(hs2.drop)
> attach(hs1)
> hsfemale<-hs1[female==1, ]
> hsmale<-hs1[female==0, ]
> dim(hsfemale)
> dim(hsmale)
> hs.all<-rbind(hsfemale, hsmale)
> dim(hs.all)
> hs.demo<-hs1[, c("id", "ses", "female", "race")]
> hs.scores<-hs1[, c("id", "read", "write", "math",
"science")]
> dim(hs.demo)
> dim(hs.scores)
> hs.merge <- merge(hs.demo, hs.scores, by="id",
all=T)
> head(hs.merge)
> dim(hs.merge)
> hs.merge1 <- merge(hs.demo, hs.scores, by.x="id",
by.y="id", all=T)
```

إنشاء إطار بيانات

```
dfrm = data.frame(v1, v2, v3, f1, f2)
```

أو

```
dfrm = as.data.frame(list.of.vectors)
```

الإشارة لعامود n بالذات في إطار البيانات

```
dfrm[[n]]
```

يعطي إطار بيانات يتكون من العمود n من إطار البيانات dfrm

```
dfrm[n]
```

يعطي إطار بيانات يتكون من الأعمدة n1, n2, …,nk من إطار البيانات dfrm

```
dfrm[c(n1, n2, ..., nk)]
```

ويمكن إستخدام التأشير التحتي subscrip على الطريقة المصفوفية (سنشرحها لاحقا) لإختيار عمود أو أكثر. فالتالي يعطي العمود n

```
dfrm[, n]
```

والتالي يعطي إطار بيانات مكون من الأعمدة في المواقع n1, n2, …,nk من إطار البيانات dfrm

```
dfrm[, c(n1, n2, ..., nk)]
```

مثال:

```
suburbs = read.table(file.choose(), header = TRUE)

suburbs
          city  county state     pop
1      Chicago    Cook    IL 2853114
2      Kenosha Kenosha    WI   90352
3       Aurora    Kane    IL  171782
4        Elgin    Kane    IL   94487
5         Gary    Lake    IN  102746
6       Joliet Kendall    IL  106221
7   Naperville  DuPage    IL  147779
8    Arlington    Cook    IL   76031
9  Bolingbrook    Will    IL   70834
10      Cicero    Cook    IL   72616
11    Evanston    Cook    IL   74239
12     Hammond    Lake    IN   83048
13    Palatine    Cook    IL   67232
14  Schaumburg    Cook    IL   75386
15      Skokie    Cook    IL   63348
```

```
16    Waukegan    Lake    IL   91452
suburbs[[1]]

[1] "Chicago" "Kenosha" "Aurora" "Elgin"
[5] "Gary" "Joliet" "Naperville" "Arlington
Heights"
[9] "Bolingbrook" "Cicero" "Evanston" "Hammond"
[13] "Palatine" "Schaumburg" "Skokie" "Waukegan"

suburbs[1]
city
1 Chicago
2 Kenosha
3 Aurora
4 Elgin
5 Gary
6 Joliet
7 Naperville
8 Arlington Heights
9 Bolingbrook
10 Cicero
11 Evanston
12 Hammond
13 Palatine
14 Schaumburg
15 Skokie
16 Waukegan

suburbs[c(1,3)]
city pop
1 Chicago 2853114
2 Kenosha 90352
3 Aurora 171782
4 Elgin 94487
5 Gary 102746
6 Joliet 106221
7 Naperville 147779
8 Arlington Heights 76031
9 Bolingbrook 70834
10 Cicero 72616
```

```
11 Evanston 74239
12 Hammond 83048
13 Palatine 67232
14 Schaumburg 75386
15 Skokie 63348
16 Waukegan 91452

suburbs[,1]
[1] "Chicago" "Kenosha" "Aurora" "Elgin"
[5] "Gary" "Joliet" "Naperville" "Arlington
Heights"
[9] "Bolingbrook" "Cicero" "Evanston" "Hammond"
[13] "Palatine" "Schaumburg" "Skokie" "Waukegan"

suburbs[,c(1,4)]
city pop
1 Chicago 2853114
2 Kenosha 90352
3 Aurora 171782
4 Elgin 94487
5 Gary 102746
6 Joliet 106221
7 Naperville 147779
8 Arlington Heights 76031
9 Bolingbrook 70834
10 Cicero 72616
11 Evanston 74239
12 Hammond 83048
13 Palatine 67232
14 Schaumburg 75386
15 Skokie 63348
16 Waukegan 91452
```

إنشاء إطار بيانات عن طريق con:

```
data = read.table(header=T, con = textConnection('
  subject sex size
        1   M    7
        2   F    6
        3   F    9
        4   M   11
  '))
```

```
close(con)

data

  subject sex size
1       1   M    7
2       2   F    6
3       3   F    9
4       4   M   11
```

التأشير السلبي Negative Indexing:

```
x = c(1,4,4,3,2,2,3)
x
[1] 1 4 4 3 2 2 3

x[-1]
[1] 4 4 3 2 2 3
```

يعطي المتجه بدون العنصر الأول.

```
x[-1:-3]
[1] 3 2 2 3
```

يعطي المتجه بدون العناصر من 1 إلى 3

```
x[-length(x)]
[1] 1 4 4 3 2 2
```

يعطي المتجه بدون العنصر الأخير.

بعض الدوال المفيدة:

1) الدالة rep

```
y = rep(1, 10)
y
 [1] 1 1 1 1 1 1 1 1 1 1

z = rep(1:5, 4)
z
 [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

w = rep(1:5, each=4)
w
 [1] 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5

abc = rep(factor(LETTERS[1:3]), 5)
```

```
abc
 [1] A B C A B C A B C A B C A B C
Levels: A B C
```

2) الدالة ls

```
ls()
 [1] "abc"     "con"     "data"    "df"      "let"
"n"       "suburbs" "v"       "w"       "x"
[11] "y"       "z"
```

3) الدالة rm لمسح أو إزالة متجه

```
rm(v)
ls()
 [1] "abc"     "con"     "data"    "df"      "let"
"n"       "suburbs" "w"       "x"       "y"
[11] "z"
```

4) دالة إيجاد حجم (طول) متجه

```
length(x)
[1] 1
length(y)
[1] 10
length(n)
[1] 4
```

5) عدد الأسطر وعدد الأعمدة في إطار بيانات أو متجه أو مصفوفة

```
nrow(df)
[1] 4
ncol(df)
[1] 2
dim(df)
[1] 4 2
```

الإحصاء الوصفي:

```
data <- read.table(header=T, con <-
textConnection('
 subject sex condition before after change
       1   F   placebo   10.1   6.9   -3.2
       2   F   placebo    6.3   4.2   -2.1
       3   M   aspirin   12.4   6.3   -6.1
       4   F   placebo    8.1   6.1   -2.0
       5   M   aspirin   15.2   9.9   -5.3
       6   F   aspirin   10.9   7.0   -3.9
       7   F   aspirin   11.6   8.5   -3.1
       8   M   aspirin    9.5   3.0   -6.5
       9   F   placebo   11.5   9.0   -2.5
      10   M   placebo   11.9  11.0   -0.9
      11   F   aspirin   11.4   8.0   -3.4
      12   M   aspirin   10.0   4.4   -5.6
      13   M   aspirin   12.5   5.4   -7.1
      14   M   placebo   10.6  10.6    0.0
      15   M   aspirin    9.1   4.3   -4.8
      16   F   placebo   12.1  10.2   -1.9
      17   F   placebo   11.0   8.8   -2.2
      18   F   placebo   11.9  10.2   -1.7
      19   M   aspirin    9.1   3.6   -5.5
      20   M   placebo   13.5  12.4   -1.1
      21   M   aspirin   12.0   7.5   -4.5
      22   F   placebo    9.1   7.6   -1.5
      23   M   placebo    9.9   8.0   -1.9
      24   F   placebo    7.6   5.2   -2.4
      25   F   placebo   11.8   9.7   -2.1
      26   F   placebo   11.8  10.7   -1.1
      27   F   aspirin   10.1   7.9   -2.2
      28   M   aspirin   11.6   8.3   -3.3
      29   F   aspirin   11.3   6.8   -4.5
      30   F   placebo   10.3   8.3   -2.0
 '))
close(con)

library(plyr)
cdata <- ddply(data, .(sex, condition), summarise,
```

```
                N      = length(change),
                change = mean(change),
                sd     = sd(change),
                se     = sd(change) /
sqrt(length(change)) )

dataNA <- data
dataNA$change[11:14] <- NA

cdata <- ddply(dataNA, .(sex, condition),
summarise,
                N      = sum(!is.na(change)),
                change = mean(change, na.rm=TRUE),
                sd     = sd(change, na.rm=TRUE),
                se     = sd(change, na.rm=TRUE) /
sqrt(sum(!is.na(change))) )
```

الإنحدار والترابط
Regression and correlation

نولد بيانات للعمل عليها:

```
set.seed(955)
xvar <- 1:20 + rnorm(20,sd=3)
zvar <- 1:20/4 + rnorm(20,sd=2)
yvar <- -2*xvar + xvar*zvar/5 + 3 + rnorm(20,sd=4)

df <- data.frame(x=xvar, y=yvar, z=zvar)

cor(df$x, df$y)
cor(df)
round(cor(df),2)

fit <- lm(y ~ x, data=df)
fit <- lm(df$y ~ df$x)
summary(fit)


fit2 <- lm(y ~ x + z, data=df)
fit2 <- lm(df$y ~ df$x + df$z)
```

التفاعل:

```
interactions:
fit3 <- lm(y ~ x * z, data=df)
fit3 <- lm(y ~ x + z + x:z, data=df)
```

إختبار t
t-test

سوف نستخدم البيانات sleep الموجودة في الحزمة

```
sleep
extra group ID
 0.7     1  1
-1.6     1  2
-0.2     1  3
-1.2     1  4
-0.1     1  5
```

```
 3.4     1  6
 3.7     1  7
 0.8     1  8
 0.0     1  9
 2.0     1 10
 1.9     2  1
 0.8     2  2
 1.1     2  3
 0.1     2  4
-0.1     2  5
 4.4     2  6
 5.5     2  7
 1.6     2  8
 4.6     2  9
 3.4     2 10

sleep.wide <- data.frame(ID=1:10,
group1=sleep$extra[1:10],
group2=sleep$extra[11:20])

t.test(extra ~ group, sleep)
t.test(sleep.wide$group1, sleep.wide$group2)

t.test(extra ~ group, sleep, var.equal=TRUE)
t.test(sleep.wide$group1, sleep.wide$group2,
var.equal=TRUE)
```

إختبار t المتزاوج

Paired-sample t-test

```
t.test(extra ~ group, sleep, paired=TRUE)
t.test(sleep.wide$group1, sleep.wide$group2,
paired=TRUE)

t.test(sleep.wide$group1 - sleep.wide$group2, mu=0,
var.equal=TRUE)
```

إختبار t لعينة واحدة

One-sample t-test

```
t.test(sleep$extra, mu=0)
```

إختبارات التكرار

Frequency tests

```
data <- read.table(header=TRUE, con <-
textConnection('
 condition result
   control      0
   control      0
   control      0
   control      0
 treatment      1
   control      0
   control      0
 treatment      0
 treatment      1
   control      1
 treatment      1
 treatment      1
 treatment      1
 treatment      1
 treatment      0
   control      0
   control      1
   control      0
   control      1
 treatment      0
 treatment      1
 treatment      0
 treatment      0
   control      0
 treatment      1
   control      0
   control      0
 treatment      1
 treatment      0
 treatment      1
 '))
close(con)
```

إختبارات حسن المطابقة

Tests of goodness-of-fit

إختبار مربع كاي

Chi-square test

```
ct <- table(data$result)
# An alternative is to manually create the table
#ct <- matrix(c(17,13), ncol=2)
#colnames(ct1) <- c("0", "1")
chisq.test(ct)
ct <- table(data$result)
pt <- c(.75, .25)
chisq.test(ct, p=pt)
```

إختبارات الإستقلال

Tests of independence

إختبار مربع كاي

Chi-square test

```
ct <- table(data$condition, data$result)
chisq.test(ct)
chisq.test(ct, correct=FALSE)
```

```
data <- read.table(header=T, con <-
textConnection('
 subject sex   age before after
       1   F   old    9.5   7.1
       2   M   old   10.3  11.0
       3   M   old    7.5   5.8
       4   F   old   12.4   8.8
       5   M   old   10.2   8.6
       6   M   old   11.0   8.0
       7   M young    9.1   3.0
       8   F young    7.9   5.2
       9   F   old    6.6   3.4
      10   M young    7.7   4.0
      11   M young    9.4   5.3
      12   M   old   11.6  11.3
      13   M young    9.9   4.6
      14   F young    8.6   6.4
      15   F young   14.3  13.5
      16   F   old    9.2   4.7
      17   M young    9.8   5.1
      18   F   old    9.9   7.3
      19   F young   13.0   9.5
      20   M young   10.2   5.4
      21   M young    9.0   3.7
      22   F young    7.9   6.2
      23   M   old   10.1  10.0
      24   M young    9.0   1.7
      25   M young    8.6   2.9
      26   M young    9.4   3.2
      27   M young    9.7   4.7
      28   M young    9.3   4.9
      29   F young   10.7   9.8
      30   M   old    9.3   9.4
'))
close(con)
```

تحليل التباين في إتجاه واحد بين المعالجات

One way between ANOVA

```
# One way between:
# IV: sex
# DV: before
aov.before.sex <- aov(before ~ sex, data=data)
summary(aov.before.sex)
```

تحليل التباين في إتجاهين بين المعالجات

Two way between ANOVA

```
# 2x2 between:
# IV: sex
# IV: age
# DV: after
# These two calls are equivalent
aov.after.sex.age <- aov(after ~ sex*age,
data=data)
aov.after.sex.age <- aov(after ~ sex + age +
sex:age, data=data)
summary(aov.after.sex.age)
```

أمثلة:

قبل العمل على بيانات ومتغيرات جديدة نقوم بإزالة المتغيرات الحالية

```
rm(list=ls())
```

ندخل البيانات الجديدة ونفعلها

```
hs1 <-
read.table("http://www.ats.ucla.edu/stat/R/notes/hs
1.csv", header=T, sep=",")
attach(hs1)
```

إختبار مربع كاي للإستقلال

```
tab1 <- table(female, ses)
# chi-square test of independence
summary(tab1)
```

إختبار t لعينة واحدة

```
t.test(write, mu=50)
```

إختبار t المتزاوج

```
t.test(write, read, paired=TRUE)
```

فحص التباين لإختبار t

```
by(write, female, var)
tapply(write, female, var)
# assuming equal variances
t.test(write~female, var.equal=TRUE)

# assuming unequal variances
t.test(write~female, var.equal=FALSE)
```

تحليل التباين:

```
anova(lm(write~factor(prog)))
summary(aov(write~factor(prog)))

m2<-lm(write~factor(prog)*factor(female))
anova(m2)

anova(lm(write~factor(prog) + read))
summary(aov(write~factor(prog) + read))
```

تحليل الإنحدار:

```
summary(lm(write~female+read))

lm2 <- lm(write~read+socst)
summary(lm2)
# plotting diagnostic plots of lm2
plot(lm2)

write[1:20]
fitted(lm2)[1:20]
resid(lm2)[1:20]
```

طرق عددية: Numerical Methods

المشتقات العددية

Numerical Derivatives

Examples

```
library(mosaic)
g = numD( a*x^2 + x*y ~ x, a=1)
g(x=2,y=10)
gg = numD( a*x^2 + x*y ~ x&x, a=1)
gg(x=2,y=10)
ggg = numD( a*x^2 + x*y ~ x&y, a=1)
ggg(x=2,y=10)
h = numD( g(x=x,y=y,a=a) ~ y, a=1)
h(x=2,y=10)
f = numD( sin(x)~x, add.h.control=TRUE)
plotFun( f(3,.hstep=h)~h, hlim=range(.00000001,.000001))
ladd( panel.abline(cos(3),0))
```

R Summary Examples

```
==========Getting Help in R==========
?read.table
find(lowess)
apropos(lm)

============Worked Examples of Functions===
example(lm)

=========Demonstrations of R Functions=========
demo(persp)
demo(graphics)
demo(Hershey)
demo(plotmath)

==================Libraries in R=============
library(spatial)

===============Contents of Libraries=======
library(help=spatial)
objects(grep("spatial",search()))

=======Installing Packages and Libraries=========
install.packages("akima")
install.packages("chron")
install.packages("Ime4")
```

```
install.packages("mcmc")
install.packages("odesolve")
install.packages("spdep")
install.packages("spatstat")
install.packages("tree")

==================Data Editor================
library(MASS)
attach(bacteria)
fix(bacteria)

==================Significance Stars============
options(show.signif.stars=FALSE)
options( digits = 3)
========To stop multiple graphs whizzing by====
par(ask=TRUE)

================Good Housekeeping============
objects()
search()
ls()
rm(list=ls())
rm(x,y,z)
detach(worms)

====Summary Information from Vectors by Groups=====
data<-read.table("c:\\temp\\daphnia.txt",header=T)
attach(data)
names(data)
tapply(Growth.rate,Detergent,mean)
tapply(Growth.rate,list(Water,Daphnia),median)
with(data, function( ...))
with(OrchardSprays,boxplot(decrease~treatment))
library(MASS)
with(bacteria,tapply((y=="n"),trt,sum))

with(mammals,plot(body,brain,log="xy"))
reg.data<-read.table("c:\\temp\\regression.txt",header=T)

with (reg.data, {
                 model<-lm(growth~tannin)
                 summary(model) } )
summary(lm(growth~tannin,data=reg.data))

=======To see the names of the dataframes in the built-in
package called datasets====
data()
```

```
==========To see all available data sets=========
data(package = .packages(all.available = TRUE))



=============Sorting, Ranking and Ordering=======
houses<-read.table("c:\\temp \\houses.txt",header=T)
attach(houses)
names(houses)
[1] "Location" "Price"

ranks<-rank(Price)
sorted<-sort(Price)
ordered<-order(Price)

view<-data.frame(Price,ranks,sorted,ordered)
view

===============Matrices=======================
X<-matrix(c(1,0,0,0,1,0,0,0,1),nrow=3)
X

vector<-c(1,2,3,4,4,3,2,1)
V<-matrix(vector,byrow=T,nrow=2)
V

====================Data Input=============
y <- c (6,7,3,4,8,5,6,2)

x<-scan() (enter data, to finish return)


===============Data Input from Files==========
data<-read.table("c:\\temp\\regression.txt",header=T)
map<-read.table("c:\\temp\\bowens.csv",header=T,sep=",")
map<-read.csv("c:\\temp\\bowens.csv")
read.delim("c:\\temp\\file.txt")

===============Browsing to Find Files==========
data<-read.table(file.choose(),header=T)

=============Reading from clipboard========
data <-read.table("clipboard", sep = " "))


==============Setting the Working Directory=====
```

```
setwd("c:\\temp")
getwd()

=====================Dataframes==============
worms<-read.table(file.choose(), header = T)
attach(worms)
names(worms)
summary(worms)
by(worms,Vegetation,mean)

==============Subscripts and Indices=========
worms[3,5]
worms[14:19,7]
worms[1:5,2:3]
worms[3,]
worms[,3]
worms[,c(1,5)]

==================Sorting Dataframes===========
worms[order(Slope),]
worms[rev(order(Slope)),]
worms[order(Vegetation,Worm.density),]

====Omitting Rows Containing Missing Values, NA====
na.omit(data)
new.frame<-na.exclude(data)

===Dataframe with Row Names instead of Row Numbers=
worms<-read.table(file.choose(),header=T,row.names=1)

====Summarizing the Contents of Dataframes =======
aggregate(worms[,c(2,3,5,7)],by=list(veg=Vegetation),mean)
aggregate(worms[,c(2,3,5,7)],by=list(veg=Vegetation,d=Damp)
,mean)
========Adding Normal Curve to histogram========

x<-rnorm(300)
h<-hist(x, breaks=10, col="red", xlab="Simulated Data",
   main="Histogram with Normal Curve")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2)
====another example=========
n <- 1000
x <- rnorm(n)
h <- hist(x, col="red",freq=TRUE)
```

```
dx <- min(diff(h$breaks))
curve(n*dx*dnorm(x), add=TRUE, col="blue", lwd=2)

===Classical Tests===Plots for testing normality ==
x<-exp(rnorm(30))
qqnorm(x)
qqline(x,lty=2)

================Test for normality==============
x<-exp(rnorm(30))
shapiro.test(x)
wilcox.test(x,mu=0)
hist(x)
skew(x)
kurtosis(x)

==========quantiles of the F distribution=========
qf(0.975,9,9)

======F test to compare two variances========
x<-exp(rnorm(30))
y<-exp(rnorm(30))
var.test(x,y)

====Bartlett test of homogeneity of variances =====
bartlett.test(x,y)

===Comparing two means==Student's t test======
qt(0.975,18)
t.test(x,y)

==================Tests on paired samples=========
t.test(x,y,paired=T)

=======Chi-squared contingency tables========
qchisq(0.95,1)
count<-matrix(c(38,14,11,51),nrow=2)
chisq.test(count)
chisq.test(count,correct=F)
chisq.test(count,correct=F)$expected



============Test of contingency=========
chisq.test(c(10,3,2,6))
chisq.test(c(10,3,2,6),p=c(0.2,0.2,0.3,0.3))
die<-ceiling(runif(100,0,6))
```

```
table(die)
chisq.test(table(die))

=======Fisher's Exact Test for Count Data ======
x<-as.matrix(c(6,4,2,8))
dim(x)<-c(2,2)
fisher.test(x)

=========Correlation and covariance============
var(x,y) ====covariance of x and y===
cor(x,y)

=====Pearson's product-moment correlation========
cor.test(Summer, Winter)
cor.test(x,y,method="spearman")

=============Kolmogorov-Smirnov test============
ks.test(A,B)

=========Regression====very simple intro===
reg<-lm(y ~ x)
summary(reg)
lm.influence(lm(growth~tannin))
influence.measures(lm(growth ~ tannin))
model<-lm(Fruit ~ Root,subset=(Grazing=="Grazed")) ====
na.action = na.omit
model<-lm(Fruit ~ Grazing*Root,na.action=na.fail)
AIC(model)
model.1<-lm(Fruit ~ Grazing*Root)
model.2<-lm(Fruit ~ Grazing+Root)
AIC(model.1, model.2)

==================Leverage===========
leverage(x1)
plot(leverage(x1),type="h")
abline(0.66,0,lty=2)
points(leverage(x1))

======Extracting components of the model using $ ==
model$coef
model$fitted
model$resid
model$df

==Extracting components from the summary.aov table=
summary.aov(model)
summary.aov(model)
```

```
          Df   Sum Sq Mean Sq F value Pr(>F)
tannin         1    88.817 88.817  30.974  0.000846
Residuals      7    20.072 2.867

summary.aov(model)[[1]][[1]]
[1] 1 7

summary.aov(model)[[1]][[2]]
[1] 88.81667 20.07222

summary.aov(model)[[1]][[3]]
[1] 88.816667 2.867460

summary.aov(model)[[1]][[4]]
[1] 30.97398 NA

summary.aov(model)[[1]][[5]]
[1] 0.0008460738 NA

model<-lm(biomass ~ clipping)
summary.aov(model)
summary.lm(model)
anova(model6,model7)

===========Prediction using the fitted model==
model<-lm(growth~tannin)
predict(model,list(tannin=5.5))
predict(model,list(tannin=c(3.3,4.4,5.5,6.6)))

===============Model checking========
par(mfrow=c(2,2))
plot(model)

==========The Multiple Regression Model=========
model7<-lm(log(ozone) ~ temp + wind + rad + I(temp^2) +
I(wind^2))
summary(model7)

====================Step reg=======
model10<-step(model1)
summary(model10)
pairs(pollute,panel=panel.smooth)

================One-Way ANOVA========
results<-read.table("c:\\temp\\yields.txt",header=T)
attach(results)
```

```
names(results)
sapply(list(sand,clay,loam),mean)
y<-c(sand,clay,loam)
soil<-factor(rep(1:3,c(10,10,10)))
sapply(list(sand,clay,loam),var)
plot(soil,y,names=c("sand","clay","loam"),ylab="yield")
summary(aov(y~soil))
plot(aov(y~soil))
summary.lm(aov(y~soil))
summary.lm(model1)
summary.aov(model2)


> x<-rnorm(150)
> h<-hist(x,breaks=15)
> xhist<-c(min(h$breaks),h$breaks)
> yhist<-c(0,h$density,0)
> xfit<-seq(min(x),max(x),length=40)
> yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
> plot(xhist,yhist,type="s",ylim=c(0,max(yhist,yfit)))
> lines(xfit,yfit)
>


curve(dnorm(x,0,1),xlim=c(-3.5,3.5))
curve(dnorm(x,0,1),-3.5,3.5)


> dat <- rnorm(30, mean = 82, sd = 7)
> hist(dat, prob = TRUE)
> curve(dnorm(x, mean = mean(dat), sd = sd(dat)), add =
TRUE)
> # Why not throw a kernel density estimate on there too
> lines(density(dat), col = "red")
>
> library(descr)
Warning message:
package 'descr' was built under R version 2.15.1
> dat <- rnorm(30, mean = 82, sd = 7)
> histkdnc(dat)
> dat <- rnorm(3000, mean = 82, sd = 7)
> histkdnc(dat)
>
x <- rnorm(1000)
hist(x, freq = FALSE, col = "grey")
curve(dnorm, col = 2, add = TRUE)
```

Glossary

- Basic arithmetic operators:
  ( ) brackets in arithmetic calculations
  ^ raise to the power of
  / division
  ***** multiplication
  + addition
  - subtraction
- Assignment operator
  <− assigns the value of the expression on the right to the identifer on the left.
- Concatenation function containing those values.
  c(...) argument is a set of values; returns a vector
- Simple statistical functions
  sum(...)
  mean(...)
  median(...)
  range(...)
  sd(...) standard deviation
  mad(...) mean absolute deviation
  IQR(...) inter-quartile range
  min(...) minimum
  max(...) maximum
- Help
  ? ... will provide information on function usage and arguments
  help.start() opens html help files
- Session management
  objects() lists objects in workspace
  ls() does the same as objects()
  rm(...) removes the objects listed from workspace
  rm(list = objects()) removes all objects from workspace
  getwd() returns the current working directory
  setwd(...) set the working directory
  dir() list files in current directory
  save.image(...) saves workspace to specified file
  quit() ends R session
  load(...) loads a saved workspace from file
- Load data from text file
  read.table(...) argument is a file in the working directory. Return value is a data frame.
- Editing data
  fix(...) or **data.entry(...)** brings up spreadsheet like environment for editing data.
- Scatterplot
  plot(...) is a versatile plotting command, draws a scatterplot when two arguments are given.
- Linear model fitting
  lm(...) can fit a variety of linear models, argument is a model formula.
- Basic inference
  t.test(...) Student's t-test
  var.test(...) F test to compare two variances
  chisq.test(...) Pearson's χ^2 test for count data

cor.test(...) test for association

- Object properties **length(...)**
  mode(...)
  typeof(...)
  class(...)
  attributes(...) returns all non-intrinsic attributes
- Object contents **summary(...)** information on an object; output depends on class
  str(...) compact details of object structure
  dput(...) text listing of object

- Regular sequences : sequence generator gives sequence with unit steps between left and right arguments
  seq(...) more general regular sequence generator, can set step (by) and/or length of sequence (length)
  rep(...) for generating sequence with repeated values
- Logical values TRUE and FALSE
- Comparison and logical operators == equality
  != inequality
  < less than
  > greater than
  <= less than or equal to
  >= greater than or equal to
  ! negation
  & and (&& scalar version)
  | or (|| scalar version)
- Accessing elements [...] contents of brackets a vector of integers, a vector of logical values or a vector of names.
- Accessing a list component [[...]] contents of brackets a number or a name.
- Accessing a named component $
- Referring to list components directly **attach(...)** brings components of a list into view; use **detach()** to tidy up.
- Generating a vector of strings **paste(...)** converts any non-character arguments into characters and puts them together into a vector of mode character.
- Generating lists **list(...)** put arguments (which may be named) into a list.
- A selection of plots **hist(...)** histogram
  qqnorm(...) normal qq-plot
  boxplot(...)
  contour(...) 3D contours
  image(...) 3D coloured blocks
  persp(...) 3D wire frame
- Some low level plotting commands **points(...)**
  lines(...)
  text(...)

- Some graphics parameters **pch** plotting character
  col colour
  mfrow multi-figure page specification mfcol

- Storing commands **savehistory(...)**
  loadhistory(...)
  source(...)

- Packages **library(...)** list and load packages
  data(...) add data from a package to workspace
- Search path **search()** lists environments in the search path in the order in which theywill be searched. Passing the position in search path or name of an environments to **objects(...)** lists objects in that environment.
- Arrays **matrix(...)** to generate a matrix (2D array)
  array(...) to generate a general array.

أسئلة وإجابات الإختبارات السابقة

بسم الله الرحمن الرحيم

جامعة الملك سعود
قسم الإحصاء وبحوث العمليات
مادة 433 بحث (طرق الحسابات في بحوث العمليات)
الإختبار النهائي للفصل الأول 1424/1423 هـ

الزمن 3 ساعات
أجب على سؤالين من الأسئلة التالية:

السؤال الأول: بإستخدام EXCEL أجب التالي:

أ) بتقريب المشتقتين الأولى و الثانية بثلاثة نقاط أوجد عدديا المشتقتين الأولى و الثانية للدالة $f(x)=e^{x}\sin(x)$ عند $x=0$ و $h=0.01$.

ب) قرب الدالة $f(x)=\ell n(x)$ بكثيرة حدود حول النقطة $x_0=1$.

ج) بإستخدام المحاكاة أوجد التكامل العددي للتالي:

$$\int_0^{\pi}\sin(x)\,dx$$

القيمة الحقيقية هي 2.

د) حل المعادلة التفاضلية التالية:

$$\frac{dx}{dt}+tx=t,\quad x(0)=0$$

في الفترة $[0,1]$.

السؤال الثاني: بإستخدام EXCEL SOLVER حل التالي:

$$1-\ Minimize\quad 8x_1+10x_2+7x_3+6x_4+11x_5+9x_6$$

Subject to

$$12x_1+\ 9x_2+25x_3+20x_4+17x_5+13x_6\geq 60$$
$$35x_1+42x_2+18x_3+31x_4+56x_5+49x_6\geq 150$$
$$37x_1+53x_2+28x_3+24x_4+29x_5+20x_6\geq 125$$
$$0\leq x_1,x_2\leq 1$$

السؤال الثالث: بإستخدام لغة البرمجة الرياضية للأمثلية LINGO حل المشكلة التالية:

مع توضيح المجوعات والبيانات والمتغيرات والهدف والقيود.

شركة تصنيع حلويات تسوق 4 أصناف من المكسرات المشكلة. الأصناف الأربعة تعرف بألأسماء التجارية Pawn و Knight و Bishop و King. كل صنف يحتوي على نسبة محددة من اللوز Peanuts و الكاشو Cashews. الجدول التالي يعطي عدد الاونسات من اللوز والكاشو في كل رطل من الأصناف الأربعة وسعربيع الرطل لكل صنف:

	Pawn	Knight	Bishop	King
Peanuts (oz.)	15	10	6	2
Cashews (oz.)	1	6	10	14
Selling Price ($)	2	3	4	5

أبرمت الشركة عقد مع جهة لتموينها 750 رطل من اللوز و 250 رطل من الكاشو يوميا لأغراض الإنتاج. مشكلة الشركة هي تحديد عدد الأرطال التي تنتجها كل يوم من كل صنف بحيث تحقق أقصى دخل بدون تجاوز الكميات الموردة يوميا من اللوز والكاشو.

بعض الصيغ المساعدة:

تقريب المشتقات عدديا:

تقريب المشتقة الأولى بثلاثة نقاط

$$f'(x)=\frac{f(x+h)-f(x-h)}{2h}+O(h^2)$$

تقريب المشتقة الثانية بثلاثة نقاط

$$f''(x)=\frac{f(x+h)-2f(x)+f(x-h)}{h^2}+O(h^2)$$

تقرب الدوال بكثيرة حدود:

تقرب أي دالة $f(x)$ والتي تكون جميع مشتقاتها موجودة ومعرفة حتى المشتقة النونية (n) بكثيرة حدود $p_{n-1}(x)$ كالتالي:

$$p_{n-1}(x)=\sum_{k=0}^{n-1}\frac{f^{(k)}(x_0)}{k!}(x-x_0)^k$$

والتي تسمى بكثيرات حدود تايلور. العلاقة السابقة يمكن حسابها تكراريا كالتالي:

$$p_j(x)=p_{j-1}(x)+\frac{f^{(j)}}{j!}(x-x_0)^j$$

حل المعادلة التفاضلية العادية:

صيغة أويلر Euler's Formula

$$k_1=f(x_j,y_j)$$

$$k_2=f\left(x_j+\frac{h}{2},y_j+\frac{h}{2}k_1\right)$$

$$y_{j+1}=y_j+h\,k_2$$

إيجاد تكامل بالمحكاة:

تستخدم المحاكاة (أو مايسمى بطريقة مونت كارلو) لإيجاد تكاملات من الشكل:

$$I=\int_a^b f(x)\,dx$$

والتي تقرب بالمجموع

$$I\approx\frac{(b-a)}{N}\sum_{i=1}^{N}f(x_i)$$

وذلك بإختيار x_i بطريقة عشوائية بدلا من إختيارها عند نقاط محددة. الخطأ هو من درجة $1/\sqrt{N}$ وهو عشوائي تماما ولايعتمد على درجة التكامل (أبعاد التكامل).

بسم الله الرحمن الرحيم

قسم الإحصاء وبحوث العمليات

جامعة الملك سعود

إختبار فصلي أول للفصل الصيفي 1424/1423 هـ

المادة 433 بحث (طرق الحسابات في بحوث العمليات)

الزمن 2 ساعة: أجب على جميع الأسئلة التالية:

السؤال الأول:

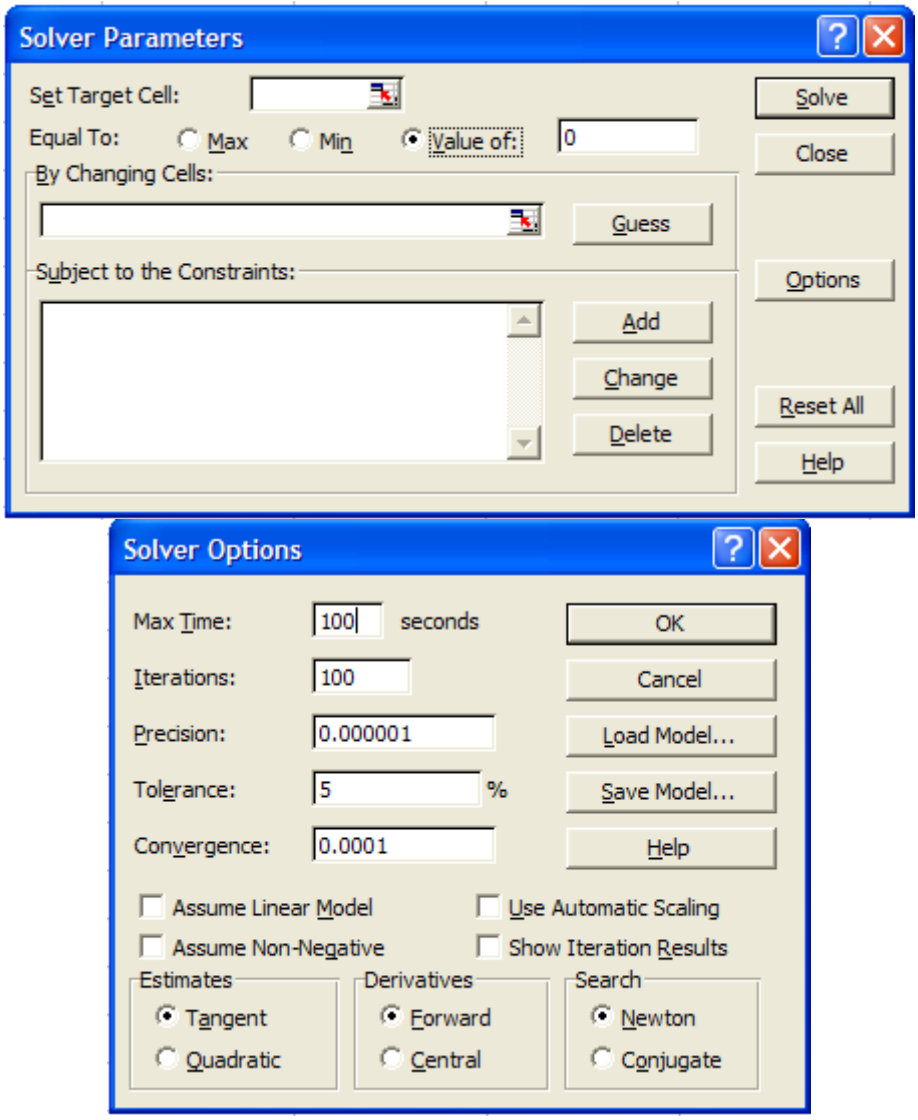
لنموذج صفحة النشر التالي :

	A	B	C	D	E	F	G	H
1		XYZ COMPUTER CORPORATION PRODUCTION PLAN						
2								
3	Product:		Standard	Deluxe				
4							PROFIT:	
5	Quantity to Produce:		0	0				
6							$0	
7								
8	Profit per Unit:		$300	$500				
9								
10			Product Component Requirements					
11								
12	Components:		Quantity Required:		Total		Number	
13			Standard	Deluxe	Usage		In Stock	
14								
15	Standard Tower		1	0	0		60	
16	Deluxe Tower		0	1	0		50	
17	Hard Drive		1	2	0		120	
18								

أ) أكتب نموذج صفحة النشر على شكل رياضي.

ب) أشرح عمل النموذج.

ج) املئ الفراغات في نوافذ Solver التالية



د) حل النموذج بإستخدام Solver

هـ) حل النموذج السابق بإستخدام WinQSB. وقارن النتائج مع الفقرة السابقة.

السؤال الثاني:

شركة أخشاب لها ثلاثة مواقع لقطع الخشب هي A و B و C كما أن لديها ثلاثة مواقع لقص وتصنيع الخشب هي 1 و 2 و 3 . الجداول التالية تعطي كميات الخشب المقطوع في كل موقع قطع والطاقة الإنتاجية لكل موقع تصنيع وتكلفة نقل الخشب من مواقع القطع إلى مواقع التصنيع:

جدول 1: كميات الخشب المقطوع في كل موقع بالطن

الموقع	كمية الخشب المقطوع بالطن
A	1600
B	1600
C	1600

جدول 2: الطاقة الإنتاجية لكل مصنع بالطن

المصنع	الطاقة الإنتاجية بالطن
1	2000
2	1400
3	1500

جدول 3: تكلفة نقل الطن الواحد من مواقع القطع إلى مواقع التصنيع

من / إلى	1	2	3
A	175	225	250
B	150	100	100
C	300	275	200

بإستخدام Solver و WinQSB حدد الكميات التي تنقل من كل موقع قطع إلى مواقع التصنيع بحيث تكون تكلفة النقل أقل مايمكن.

بسم الله الرحمن الرحيم

قسم الإحصاء وبحوث العمليات
جامعة الملك سعود

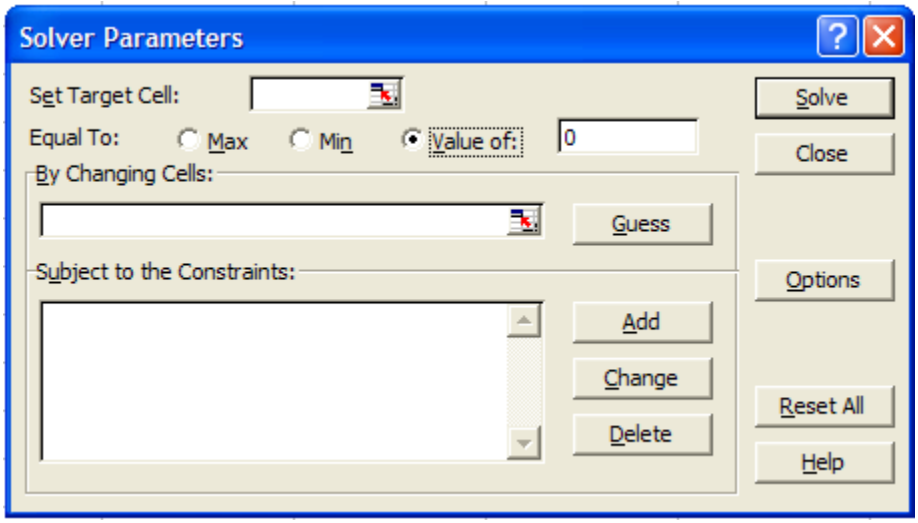
إختبار فصلي أول للفصل الصيفي 1424/1423 هـ
المادة 433 بحث (طرق الحسابات في بحوث العمليات)
الزمن 2 ساعة: أجب على جميع الأسئلة التالية:

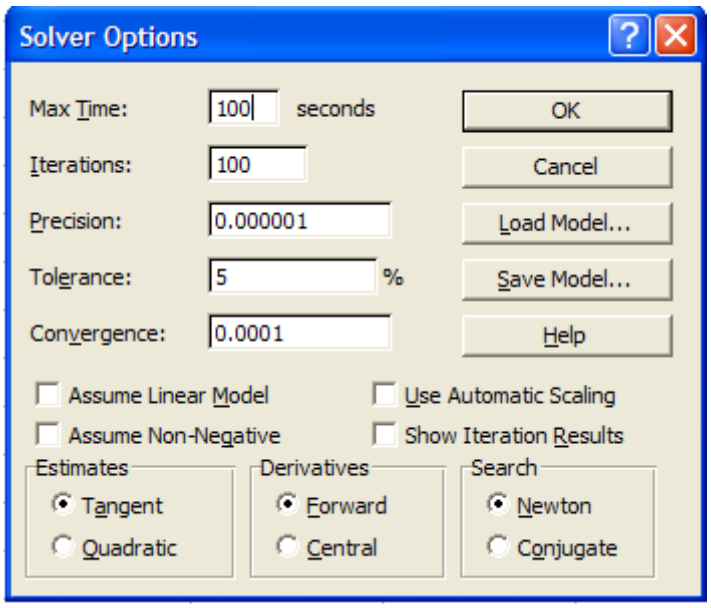
السؤال الأول:

لنموذج صفحة النشر التالي :

	A	B	C	D	E	F	G
1		Milk	Beans	Orange			
2		0	0	0			
3	Min Cost	2	0.2	0.25	0		
4	Niacin	3.2	4.9	0.8	0	>=	13
5	Thiamin	1.12	1.3	0.19	0	>=	1.5
6	Vitamin C	32	0	93	0	>=	45
7							

أ) أكتب نموذج صفحة النشر على شكل رياضي.
ب) أشرح عمل النموذج.
ج) املئ الفراغات في نوافذ Solver التالية



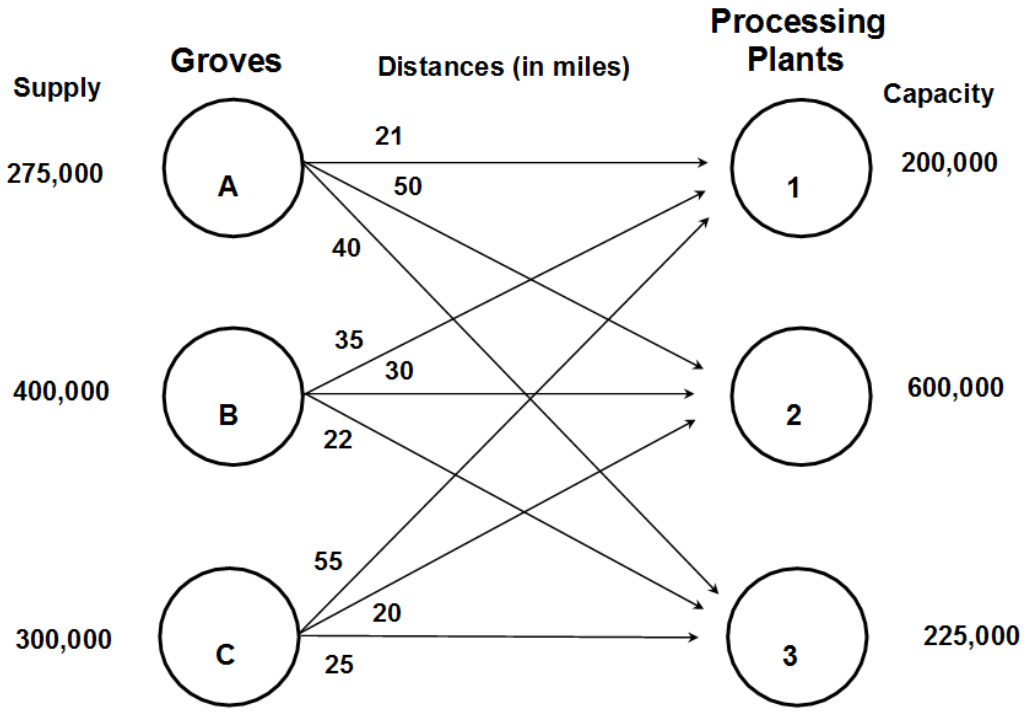


د) حل النموذج بإستخدام Solver

هـ) حل النموذج السابق بإستخدام WinQSB. وقارن النتائج مع الفقرة السابقة.

السؤال الثاني:

بإستخدام Solver و WinQSB حل مسألة النقل التالية:



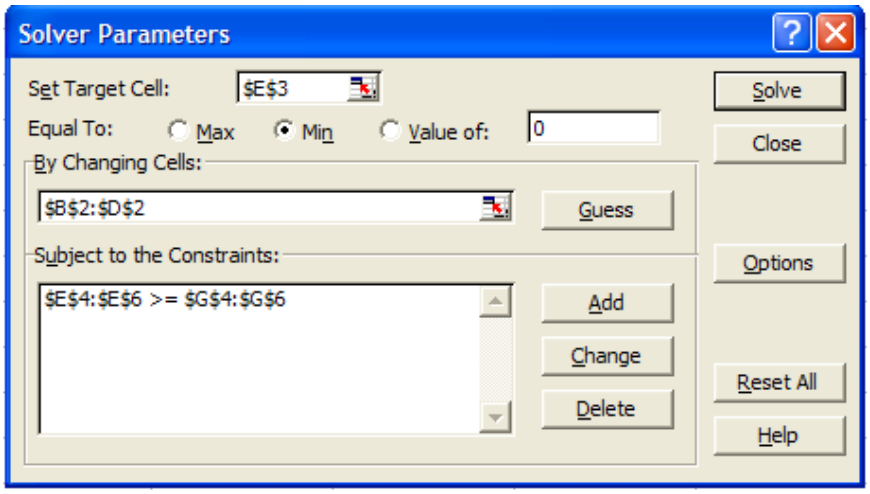
بسم الله الرحمن الرحيم
قسم الإحصاء وبحوث العمليات
جامعة الملك سعود
إجابات محتملة لإختبار فصلي أول للفصل الصيفي 1424/1423 هـ
المادة 433 إحص (طرق الحسابات في بحوث العمليات)

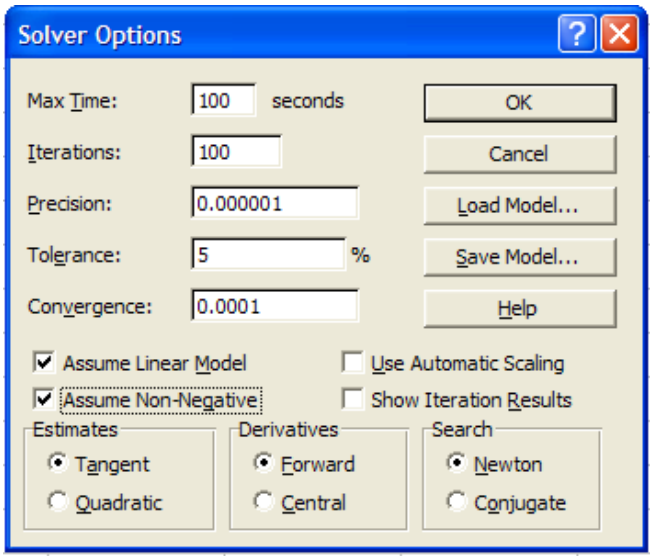
إجابة السؤال الأول:
ا)

$$
\begin{aligned}
Min \quad & 2x_1 + 0.2x_2 + 0.25x_3 \\
St \quad & 3.2x_1 + 4.9x_2 + 0.8x_3 \geq 13 \\
& 1.12x_1 + 1.3x_2 + 0.19x_3 \geq 1.5 \\
& 32x_1 \qquad\quad + 93x_3 \geq 45 \\
& x_1, x_2, x_3 \geq 0
\end{aligned}
$$

ب) نموذج لتقليل تكلفة وجبة تتكون من الحليب والفول وعصير البرتقال تحت شروط ان لاتقل كميات بعض الفيتامينات (3 فيتامينات) في مكونات الوجبة عن قيم معطاة.

ج)





(د

	A	B	C	D	E	F	G
1		Milk	Beans	Orange			
2		0	2.574062	0.483871			
3	Min Cost	2	0.2	0.25	0.63578		
4	Niacin	3.2	4.9	0.8	13	>=	13
5	Thiamin	1.12	1.3	0.19	3.438216	>=	1.5
6	Vitamin C	32	0	93	45	>=	45

Microsoft Excel 10.0 Answer Report
Worksheet: [Book1]Sheet1
Report Created: 8/2/2003 9:14:15 PM

Target Cell (Min)

Cell	Name	Original Value	Final Value
$E$3	Min Cost	0	0.635780118

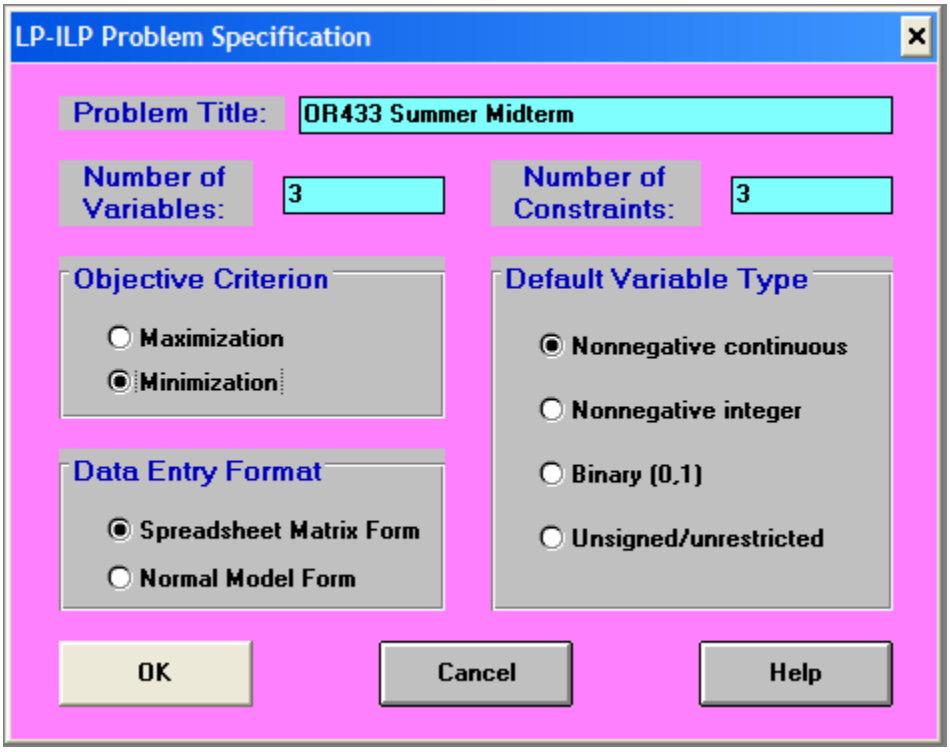
Adjustable Cells

Cell	Name	Original Value	Final Value
$B$2	Milk	0	0
$C$2	Beans	0	2.574061883
$D$2	Orange	0	0.483870968

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$E$4	Niacin	13	$E$4>=$G$4	Binding	0
$E$5	Thiamin	3.438215932	$E$5>=$G$5	Not Binding	1.938215932
$E$6	Vitamin C	45	$E$6>=$G$6	Binding	0

هـ)

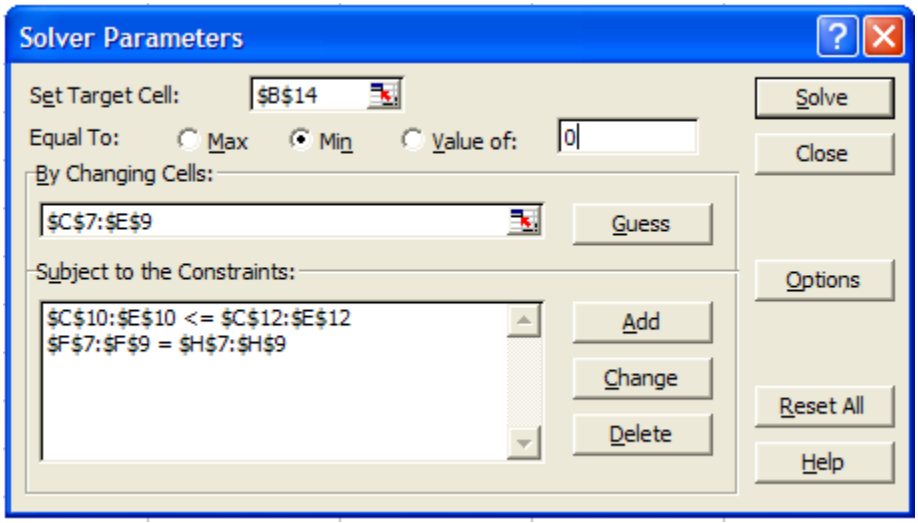


Variable -->	X1	X2	X3	Direction	R. H. S.
Minimize	2	0.2	0.25		
C1	3.2	4.9	0.8	>=	13
C2	1.12	1.3	0.19	>=	1.5
C3	32	0	93	>=	45
LowerBound	0	0	0		
UpperBound	M	M	M		
VariableType	Continuous	Continuous	Continuous		

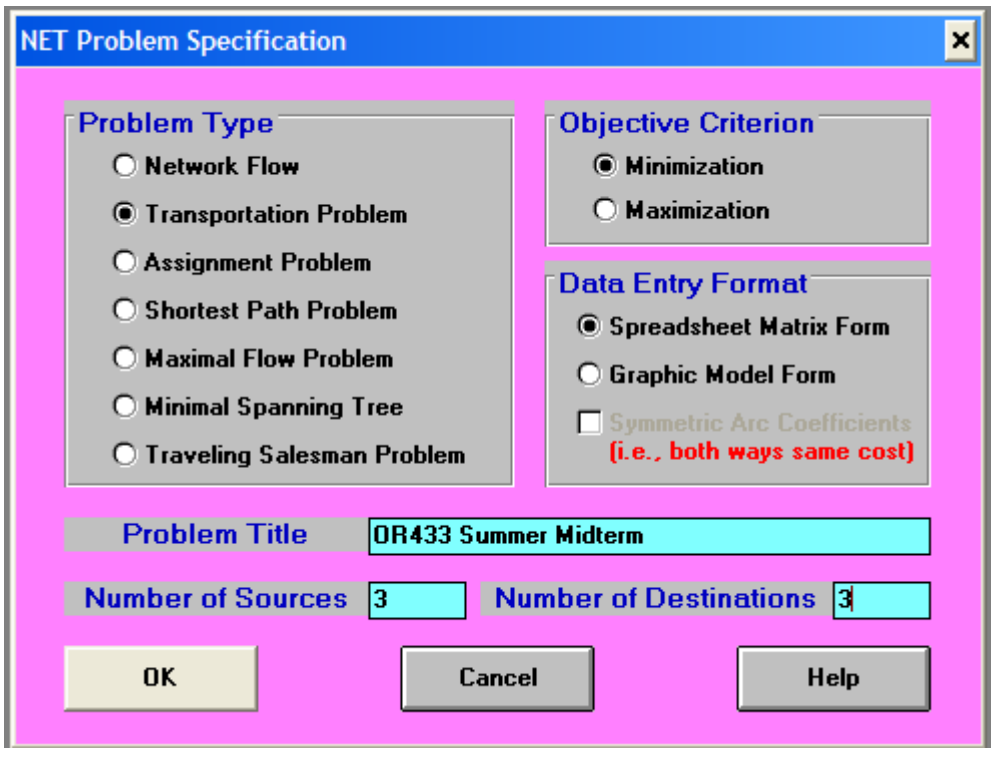
	21:23:48		Saturday	August	02	2003		
	Decision Variable	Solution Value	Unit Cost or Profit c[j]	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c[j]	Allowable Max. c[j]
1	X1	0	2.0000	0	1.7946	at bound	0.2054	M
2	X2	2.5741	0.2000	0.5148	0	basic	0	1.5313
3	X3	0.4839	0.2500	0.1210	0	basic	0.0327	5.4656
	Objective	Function	(Min.) =	0.6358				
	Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price	Allowable Min. RHS	Allowable Max. RHS
1	C1	13.0000	>=	13.0000	0	0.0408	5.6944	M
2	C2	3.4382	>=	1.5000	1.9382	0	-M	3.4382
3	C3	45.0000	>=	45.0000	0	0.0023	0	1,511.2500

إجابة السؤال الثاني:

	A	B	C	D	E	F	G	H
1								
2			1	2	3			
3		A	175	225	250			
4		B	150	100	100			
5		C	300	275	200			
6								
7		A	0	0	0	0	=	1600
8		B	0	0	0	0	=	1600
9		C	0	0	0	0	=	1600
10			0	0	0			
11			=	=	=			
12			2000	1400	1500			
13								
14	Min Cost =	0						



	A	B	C	D	E	F	G	H
1								
2			1	2	3			
3		A	175	225	250			
4		B	150	100	100			
5		C	300	275	200			
6								
7		A	1600	0	0	1600	=	1600
8		B	200	1400	0	1600	=	1600
9		C	100	0	1500	1600	=	1600
10			1900	1400	1500			
11			=	=	=			
12			2000	1400	1500			
13								
14	Min Cost =	780000						
15								



From \ To	Destination 1	Destination 2	Destination 3	Supply
Source 1	175	225	250	1600
Source 2	150	100	100	1600
Source 3	300	275	200	1600
Demand	2000	1400	1500	

08-02-2003	From	To	Shipment	Unit Cost	Total Cost	Reduced Cost
1	Source 1	Destination 1	1600	175	280000	0
2	Source 2	Destination 1	200	150	30000	0
3	Source 2	Destination 2	1400	100	140000	0
4	Source 3	Destination 1	100	300	30000	0
5	Source 3	Destination 3	1500	200	300000	0
6	Unfilled_Demand	Destination 1	100	0	0	0
	Total	Objective	Function	Value =	780000	

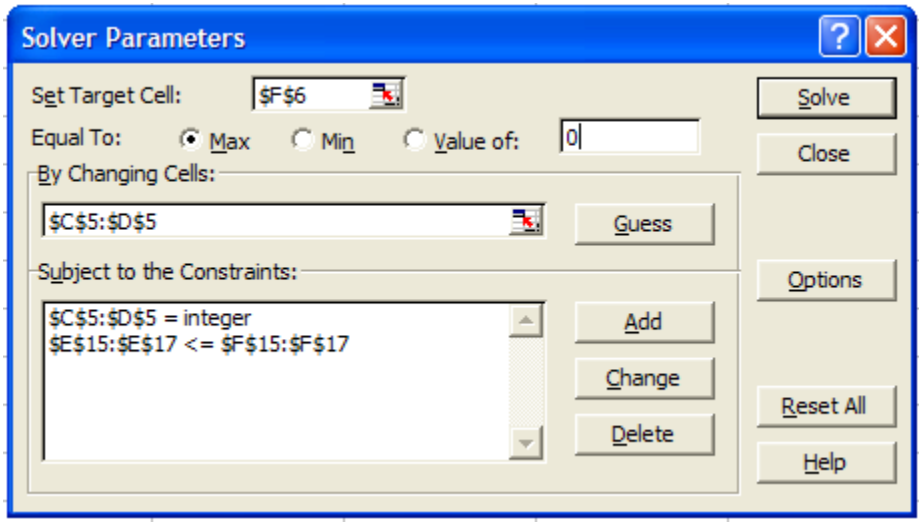
إجابة السؤال الأول:

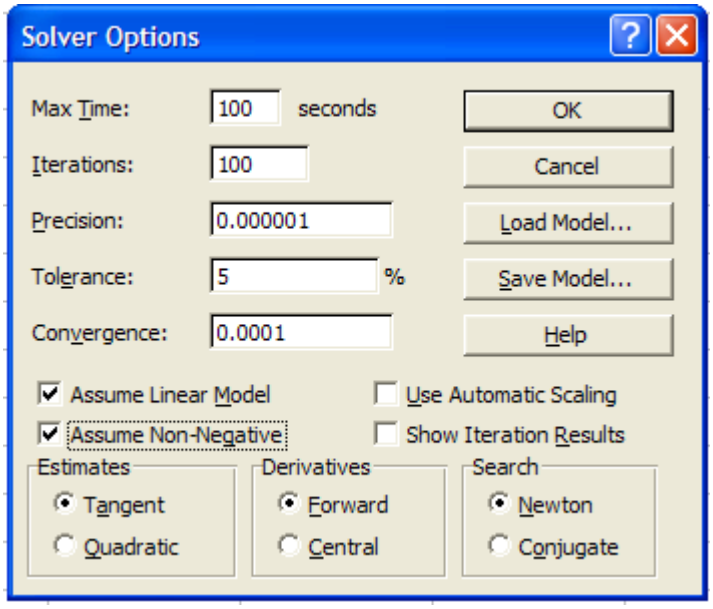
أ)

$$
\begin{aligned}
Max \quad & 33x_1 + 500x_2 \\
St \quad & x_1 \leq 60 \\
& x_2 \leq 50 \\
& x_1 + 2x_2 \leq 120 \\
& x_1, x_2 \geq 0, \text{ and integers}
\end{aligned}
$$

ب) نوعين من الحاسبات يطلب العدد الذي ينتج من كل نوع بحيث نحقق أعلى ربح مع الأخذ في الإعتبار كميات قطع التجميع المتاحة للتصنيع.

ج)





(د

	A	B	C	D	E	F
1		XYZ COMPUTER CORPORATION PRODUCTION PLAN				
2						
3	Product:		Standard	Deluxe		
4						Profit:
5	Quantity to Produce:		60	30		
6						33000
7						
8	Profit per Unit:		300	500		
9						
10			Product Component Requirements			
11						
12	Components:		Quantity Required:		Total	Number
13			Standard	Deluxe	Usage	in Stock
14						
15	Standard Tower		1	0	60	60
16	Deluxe Tower		0	1	30	50
17	Hard Drive		1	2	120	120
18						

Microsoft Excel 10.0 Answer Report
Worksheet: [Book3]Sheet1
Report Created: 8/2/2003 11:02:41 PM

Target Cell (Max)

Cell	Name	Original Value	Final Value
$F$6	Profit:	33000	33000

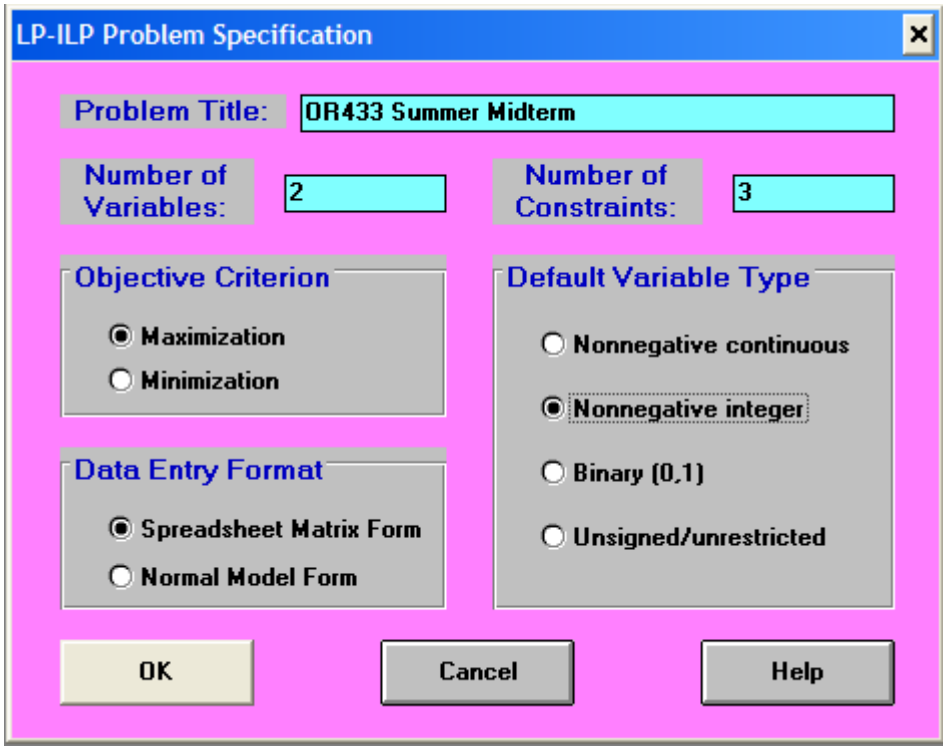
Adjustable Cells

Cell	Name	Original Value	Final Value
$C$5	Quantity to Produce: Standard	60	60
$D$5	Quantity to Produce: Deluxe	30	30

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$E$15	Standard Tower Usage	60	$E$15<=$F$15	Binding	0
$E$16	Deluxe Tower Usage	30	$E$16<=$F$16	Not Binding	20
$E$17	Hard Drive Usage	120	$E$17<=$F$17	Binding	0
$C$5	Quantity to Produce: Standard	60	$C$5=integer	Binding	0
$D$5	Quantity to Produce: Deluxe	30	$D$5=integer	Binding	0

هـ)

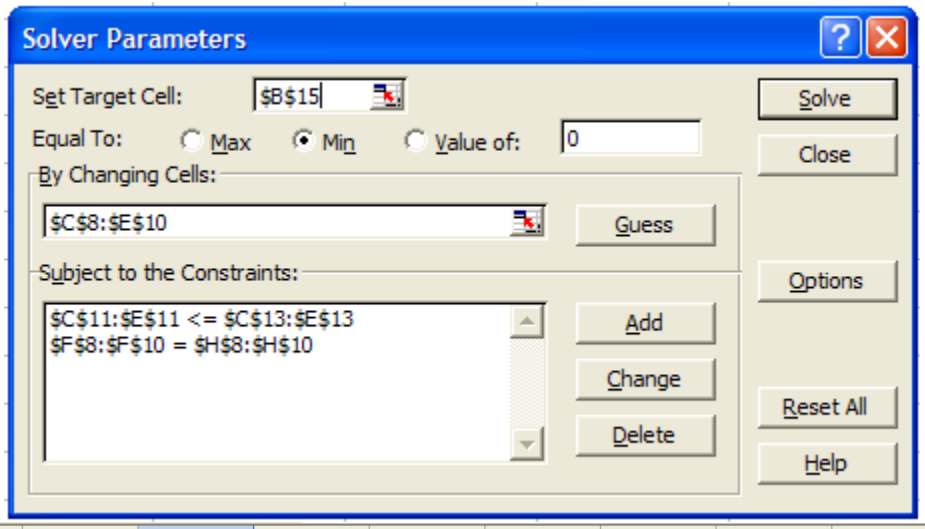


Variable -->	X1	X2	Direction	R. H. S.
Maximize	300	500		
C1	1	0	<=	60
C2	0	1	<=	50
C3	1	2	<=	120
LowerBound	0	0		
UpperBound	M	M		
VariableType	Integer	Integer		

	23:08:29		Saturday	August	02	2003		
	Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c(j)	Allowable Max. c(j)
1	X1	60.0000	300.0000	18,000.0000	0	basic	250.0000	M
2	X2	30.0000	500.0000	15,000.0000	0	basic	0	600.0000
	Objective	Function	(Max.) =	33,000.0000				
	Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price	Allowable Min. RHS	Allowable Max. RHS
1	C1	60.0000	<=	60.0000	0	50.0000	20.0000	120.0000
2	C2	30.0000	<=	50.0000	20.0000	0	30.0000	M
3	C3	120.0000	<=	120.0000	0	250.0000	60.0000	160.0000

إجابة السؤال الثاني:

	A	B	C	D	E	F	G	H
1								
2			1	2	3			
3		A	21	50	40			
4		B	35	30	22			
5		C	55	20	25			
6								
7								
8		A	0	0	0	0		275000
9		B	0	0	0	0		400000
10		C	0	0	0	0		300000
11			0	0	0			
12								
13			200000	600000	225000			
14								
15	Min Dist.	0						



	A	B	C	D	E	F	G	H
1								
2			1	2	3			
3		A	21	50	40			
4		B	35	30	22			
5		C	55	20	25			
6								
7								
8		A	200000	0	75000	275000		275000
9		B	0	250000	150000	400000		400000
10		C	0	300000	0	300000		300000
11			200000	550000	225000			
12								
13			200000	600000	225000			
14								
15	Min Dist.	24000000						

Microsoft Excel 10.0 Answer Report
Worksheet: [Book3]Sheet2
Report Created: 8/2/2003 11:19:31 PM

Target Cell (Min)

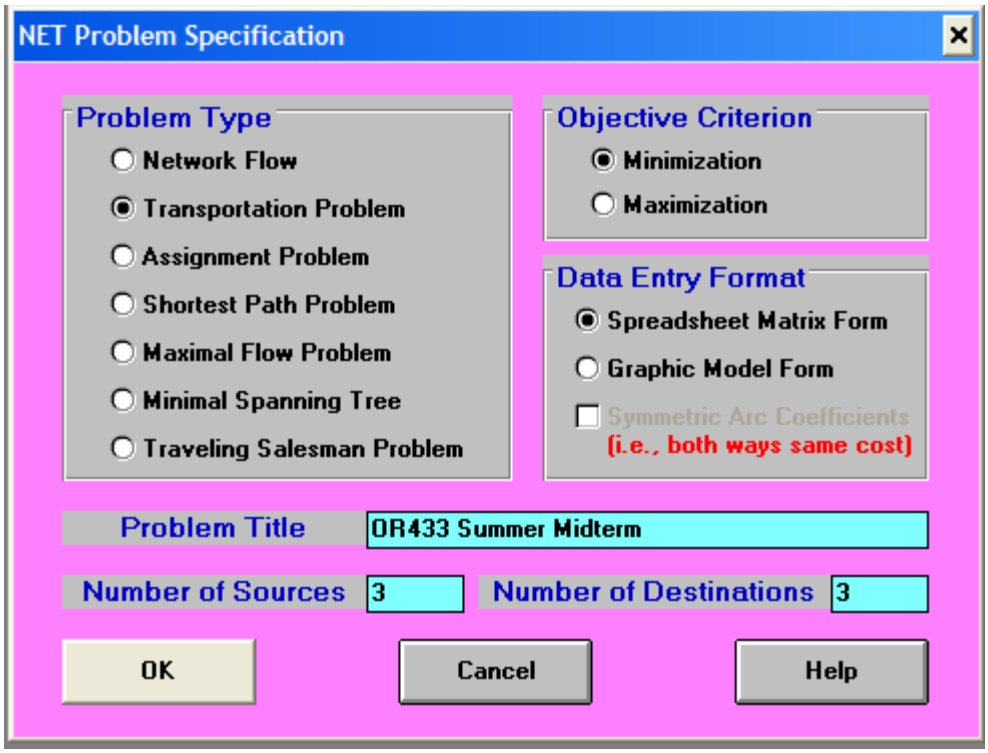
Cell	Name	Original Value	Final Value
$B$15	Min Dist. C	0	24000000

Adjustable Cells

Cell	Name	Original Value	Final Value
$C$8	A	0	200000
$D$8	A	0	0
$E$8	A	0	75000
$C$9	B	0	0
$D$9	B	0	250000
$E$9	B	0	150000
$C$10	C	0	0
$D$10	C	0	300000
$E$10	C	0	0

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$C$11		200000	$C$11<=$C$13	Binding	0
$D$11		550000	$D$11<=$D$13	Not Binding	50000
$E$11		225000	$E$11<=$E$13	Binding	0
$F$8	A	275000	$F$8=$H$8	Not Binding	0
$F$9	B	400000	$F$9=$H$9	Not Binding	0
$F$10	C	300000	$F$10=$H$10	Not Binding	0



From \ To	Destination 1	Destination 2	Destination 3	Supply
Source 1	21	50	40	275000
Source 2	35	30	22	400000
Source 3	55	20	25	300000
Demand	200000	600000	225000	

08-02-2003	From	To	Shipment	Unit Cost	Total Cost	Reduced Cost
1	Source 1	Destination 1	200000	21	4200000	0
2	Source 1	Destination 3	75000	40	3000000	0
3	Source 2	Destination 2	250000	30	7500000	0
4	Source 2	Destination 3	150000	22	3300000	0
5	Source 3	Destination 2	300000	20	6000000	0
6	Unfilled_Demand	Destination 2	50000	0	0	0
	Total	Objective	Function	Value =	2.4E+07	

المراجع:

1- Lawrence, John A., Jr, and Pasternack, Barry A. (2002) *Applied Management Science*: Modeling, Spreadsheet Analysis, and Communication for Decision Making. Second Edition. Wiley.
2- Lawrence, John A., Jr, and Pasternack, Barry A. (1998) *Applied Management Science*: A Computer-Integrated Approach for Decision Making. Wiley.
3- Microsoft EXCEL Online Help.
4- WINQSB Online Help and Manual.
5- LINDO Online Help.
6- LINGO Online Help.
7- The R Manuals at http://cran.r-project.org/