chapter 4 : ESTIMATION

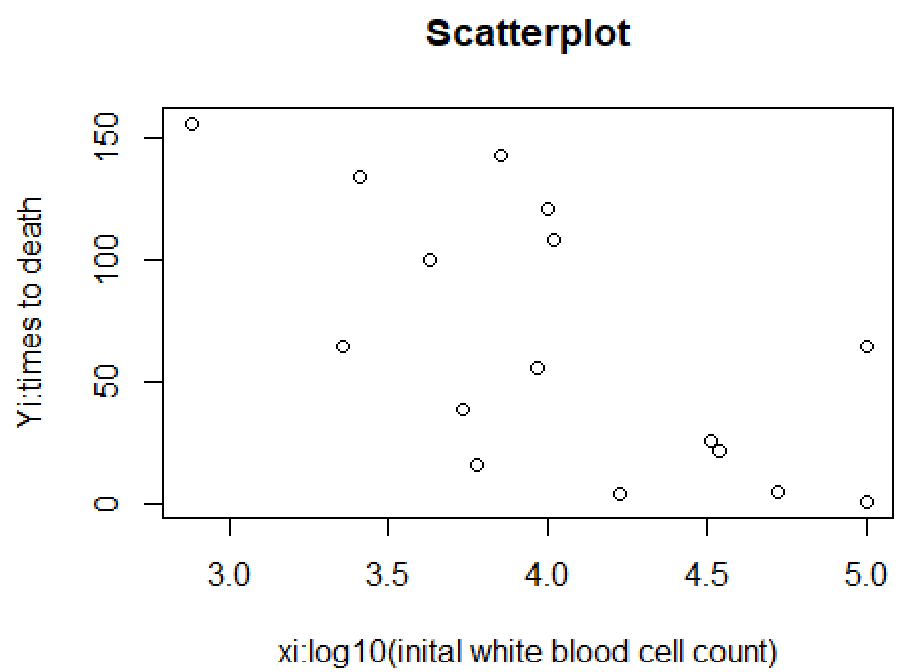
Kholoud Basalim
2024-10-14

Q4.2 page70:

```
# yi:times to death .
yi<-c(65,156,100,134,16,108,121,4,39,143,56,26,22,1,1,5,65)
yi

##  [1]  65 156 100 134  16 108 121   4  39 143  56  26  22   1   1   5  65

#xi:log10(inital white blood cell count).
xi<-c(3.36,2.88,3.63,3.41,3.78,4.02,4,4.23,3.73,3.85,3.97,4.51,4.54,5,5,4.72,5)
xi

##  [1] 3.36 2.88 3.63 3.41 3.78 4.02 4.00 4.23 3.73 3.85 3.97 4.51 4.54 5.00 5.00
## [16] 4.72 5.00
```

Plot y_i against x_i . Do the data show any trend?

```
#   Plot the times to death yi against log10(inital white blood cell count) xi.
plot(xi,yi,main = "Scatterplot", xlab = "xi:log10(inital white blood cell count)",
ylab = "Yi:times to death")
```



A possible specification for $E(Y)$ is

$$E(Y_i) = exp(\beta_1 + \beta_2 x_i),$$

which will ensure that $E(Y)$ is non-negative for all values of the parameters and all values of x. Which link function is appropriate in this case?

$$Y_i \sim exp\ (\theta_i)$$

Mean of Y_i: $E(Y_i) = \mu_i$

Variance of Y_i: $Var(Y_i) = \sigma^2$

(Mean)^2 =Variance

$$\boldsymbol{\mu_i = e^{\beta_1 + \beta_2 x_i}}$$

$$\boldsymbol{\log\ \mu_i = \beta_1 + \beta_2 x_i}$$

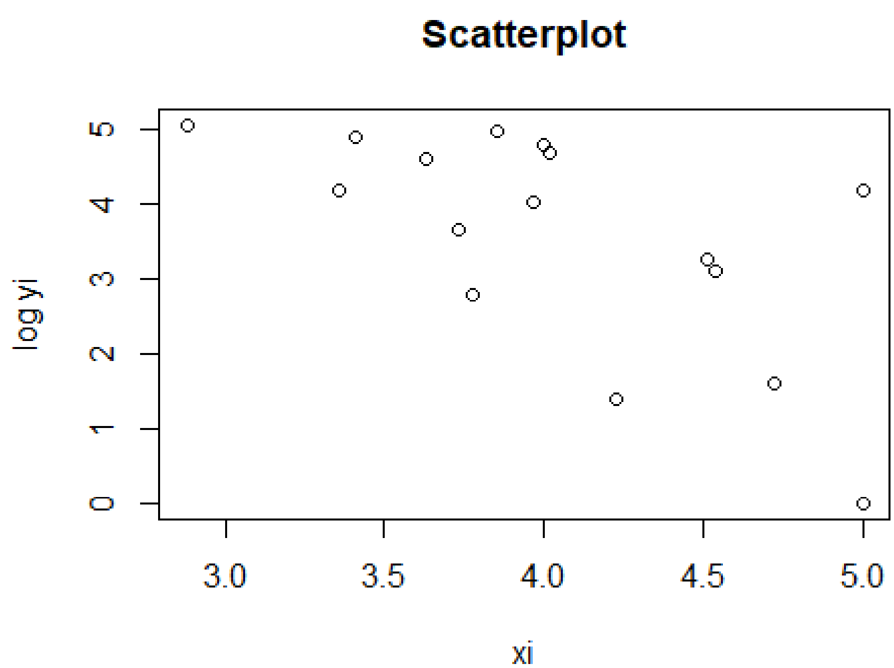
$$\boldsymbol{\log \mu_i = \eta_i}$$

$$\eta_i = \eta_i(x_i) = \beta_1 + \beta_2\ x_i$$

Link Function:

$$g(\boldsymbol{\mu_i}) = \boldsymbol{\log\ \mu_i}$$

```r
#Plot  log yi against log(i)  to examine this model.
plot(xi,log(yi), main = "Scatterplot", xlab = "xi", ylab = "log yi")
```



d) Fit a model with the equation for $E(Y_i)$ given in (b) and the Exponential distribution using appropriate statistical software.

We will use the iterative formula which is :

$$\mathbf{b}^{\mathbf{m+1}} = \mathbf{b}^{\mathbf{m}} + [\,\boldsymbol{\tau}(\mathbf{b}^{\mathbf{m}})]^{(-1)}\ \mathbf{U}(\mathbf{b}^{\mathbf{m}})\quad, \mathbf{m} = \mathbf{0}, \mathbf{1}, \mathbf{2}\ \ldots \ldots$$

We will start with an initial value $b^{(0)} = \begin{bmatrix} b_1^{(0)} \\ b_2^{(0)} \end{bmatrix} = \begin{bmatrix} 11 \\ -2 \end{bmatrix}$

```
# Initial values b0:
beta<-c(11,-2)
beta

## [1] 11 -2
```

Design matrix (X) :

```
#Design matrix (X) :
X=matrix(c(rep(1,17),xi),nrow=17,ncol = 2,byrow = F)
X

##       [,1] [,2]
##  [1,]    1 3.36
##  [2,]    1 2.88
##  [3,]    1 3.63
##  [4,]    1 3.41
##  [5,]    1 3.78
##  [6,]    1 4.02
##  [7,]    1 4.00
##  [8,]    1 4.23
##  [9,]    1 3.73
## [10,]    1 3.85
## [11,]    1 3.97
## [12,]    1 4.51
## [13,]    1 4.54
## [14,]    1 5.00
## [15,]    1 5.00
## [16,]    1 4.72
## [17,]    1 5.00
```

Transpose of design matrix (X^T) :

```
#Transpose of design matrix :
Xt<-t(X)
Xt

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## [1,] 1.00 1.00 1.00 1.00 1.00 1.00    1 1.00 1.00  1.00  1.00  1.00  1.00     1
## [2,] 3.36 2.88 3.63 3.41 3.78 4.02    4 4.23 3.73  3.85  3.97  4.51  4.54     5
##      [,15] [,16] [,17]
## [1,]     1  1.00     1
## [2,]     5  4.72     5
```

The matrix of working weights $W = diag\left[\frac{1}{var(y_i)}\left(\frac{\partial \mu}{\partial \eta}\right)^2\right]$ is :

```
#The matrix of working weights (W) is :
W<-diag(c(rep(1,17)),nrow = 17,ncol = 17)
W

##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    1    0    0    0    0    0    0    0    0     0     0     0     0
##  [2,]    0    1    0    0    0    0    0    0    0     0     0     0     0
##  [3,]    0    0    1    0    0    0    0    0    0     0     0     0     0
##  [4,]    0    0    0    1    0    0    0    0    0     0     0     0     0
##  [5,]    0    0    0    0    1    0    0    0    0     0     0     0     0
##  [6,]    0    0    0    0    0    1    0    0    0     0     0     0     0
##  [7,]    0    0    0    0    0    0    1    0    0     0     0     0     0
##  [8,]    0    0    0    0    0    0    0    1    0     0     0     0     0
##  [9,]    0    0    0    0    0    0    0    0    1     0     0     0     0
## [10,]    0    0    0    0    0    0    0    0    0     1     0     0     0
## [11,]    0    0    0    0    0    0    0    0    0     0     1     0     0
## [12,]    0    0    0    0    0    0    0    0    0     0     0     1     0
## [13,]    0    0    0    0    0    0    0    0    0     0     0     0     1
## [14,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [15,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [16,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [17,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##       [,14] [,15] [,16] [,17]
##  [1,]     0     0     0     0
##  [2,]     0     0     0     0
##  [3,]     0     0     0     0
##  [4,]     0     0     0     0
##  [5,]     0     0     0     0
##  [6,]     0     0     0     0
##  [7,]     0     0     0     0
##  [8,]     0     0     0     0
##  [9,]     0     0     0     0
## [10,]     0     0     0     0
## [11,]     0     0     0     0
## [12,]     0     0     0     0
## [13,]     0     0     0     0
## [14,]     1     0     0     0
## [15,]     0     1     0     0
## [16,]     0     0     1     0
## [17,]     0     0     0     1
```

The information matrix $\tau = X^t * W * X$ is :

```
##Information matrix, Tau= Xt*W*X (Multiply the matrices):
tau<- Xt%*%W%*%X
tau

##       [,1]     [,2]
## [1,] 17.00  69.6300
## [2,] 69.63 291.4571
```

The score statistics are: (U1 and U2):

```
#The score statistics are: (U1 and U2
U1<-sum(-1+(yi/exp(beta[1]+beta[2]*xi)))
U2<-sum(-xi+(yi*xi/exp(beta[1]+beta[2]*xi)))
```

The vector of scores $\boldsymbol{U} = \begin{bmatrix} \boldsymbol{U_1} \\ \boldsymbol{U_2} \end{bmatrix}$

```
U<-matrix(c(U1,U2))
U

##           [,1]
## [1,]  43.14648
## [2,] 195.23871
```

We make the following Calculation:

$$\mathbf{b^{m+1}} = \mathbf{b^m} + [\,\boldsymbol{\tau}(\mathbf{b^m})]^{-1}\ \mathbf{U}(\mathbf{b^m})$$

```
#We make the following Calculation:
b<-beta+solve(tau)%*%U
b

##           [,1]
## [1,] 1.4248231
## [2,] 0.9574105
```

Repeat the steps but change the initial value to the last value of beta you obtained. Stop the iteration process if you get the same value of beta in two successive steps.

The following table summarizes the results of the iterative procedure:

m	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
b_1^m	11	1.4245							8.4775	8.4775
b_2^m	-2	0.9575							-1.1093	-1.1093

Since $b^{(8)} = b^{(9)} = \begin{bmatrix} 8.4775 \\ -1.1093 \end{bmatrix}$, the iteration algorithm was terminated at step (8) .

The final approximation of MLE of $\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$ $is\ \ b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 8.4775 \\ -1.1093 \end{bmatrix}$

OR By using Loop

```
# yi:times to death .
yi<-c(65,156,100,134,16,108,121,4,39,143,56,26,22,1,1,5,65)
yi

##  [1]  65 156 100 134  16 108 121   4  39 143  56  26  22   1   1   5  65

#xi:log10(inital white blood cell count).
xi<-c(3.36,2.88,3.63,3.41,3.78,4.02,4,4.23,3.73,3.85,3.97,4.51,4.54,5,5,4.72,5)
xi

##  [1] 3.36 2.88 3.63 3.41 3.78 4.02 4.00 4.23 3.73 3.85 3.97 4.51 4.54 5.00 5.00
## [16] 4.72 5.00

# Initial values
beta <- matrix(c(11, -2))
epsilon <- 1e-6
max_iter <- 100
```

By use Loop

```
############### By use Loop ###############

# Iterative process
for (iter in 1:max_iter) {
##Design matrix (X)
X=matrix(c(rep(1,17),xi),nrow=17,ncol = 2,byrow = F)
#Transpose of design matrix :
Xt<-t(X)
#Working weights matrix (W)= Diagonal matrix having 17 rows and 17 columns :
W<-diag(c(rep(1,17)),nrow = 17,ncol = 17)
#Information matrix , Tau= Xt*W*X
# Multiply the matrices.
Tau<-Xt %*% W %*% X
# to calculate the inverse of Tau:
Tau_inver<-solve(Tau)
#The score statistics are: (U1 and U2)
U1<-sum(-1+(yi/exp(beta[1]+beta[2]*xi)))
U2<-sum(-xi+(yi*xi/exp(beta[1]+beta[2]*xi)))
#The vector of scores (U)
U<-matrix(c(U1,U2))
U
#iterative equation to find an approximate estimate of beta: b1,b2
b<- beta+(Tau_inver %*% U)
b
# Check for convergence
if (max(abs(b - beta)) < epsilon ) {
  break
}

# Update beta for the next iteration
```

```
beta <- b

# Print iteration results (optional)
cat("Iteration", iter, ": beta =", t(b), "\n")
}

## Iteration 1 : beta = 1.424823 0.9574105
## Iteration 2 : beta = 4.058176 0.1854641
## Iteration 3 : beta = 6.027011 -0.4071352
## Iteration 4 : beta = 7.550717 -0.8500827
## Iteration 5 : beta = 8.297601 -1.060796
## Iteration 6 : beta = 8.461092 -1.105153
## Iteration 7 : beta = 8.476365 -1.109021
## Iteration 8 : beta = 8.477422 -1.10928
## Iteration 9 : beta = 8.477493 -1.109297
## Iteration 10 : beta = 8.477497 -1.109298
```

Fit the model described in (c) using statistical software

```
model<-glm(yi~xi ,family =Gamma(link = "log"))
summary(model, dispersion =1)

##
## Call:
## glm(formula = yi ~ xi, family = Gamma(link = "log"))
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.4775     1.6034   5.287 9.13e-05 ***
## xi           -1.1093     0.3872  -2.865   0.0118 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.9388638)
##
##     Null deviance: 26.282  on 16  degrees of freedom
## Residual deviance: 19.457  on 15  degrees of freedom
## AIC: 173.97
##
## Number of Fisher Scoring iterations: 8
```

- **Find the 95% confidence interval for β_1 and β_2 is:**

$$\mathbf{b_1 \pm 1.96\ \ se(b_1)} >> \quad \mathbf{8.4775 \pm 1.96 * 1.6034}$$

$$\mathbf{b_2 \pm 1.96\ \ se(b_2)} >> \mathbf{-1.1093 \pm 1.96 * 0.3872}$$

```
CI_of_beta <- confint.default(model,level=0.95)
CI_of_beta
```

```
##                  2.5 %     97.5 %
## (Intercept)  5.334837 11.6201502
## xi          -1.868283 -0.3503104
```

Find approximately the variance-covariance matrix of the MLE $\boldsymbol{b}$:

Final approximate of the inverse of the information matrix evaluated at $\boldsymbol{b}$ is:

For obtaining the estimated variance-covariance matrix of parameter estimate s in a fitted model:

The variance-covariance matrix of the **MLE** $\boldsymbol{b}$ ($\mathbf{cov(b) = \tau^{-1}}$) is

```
Tauinver<-vcov(model, dispersion =1)
Tauinver

##             (Intercept)         xi
## (Intercept)   2.7383886 -0.6542095
## xi           -0.6542095  0.1597237
```

$$\boldsymbol{\tau^{-1} = cov(\hat{\beta}) = cov(b) = \begin{bmatrix} \mathbf{var(b_1)} & \mathbf{cov(b_1, b_2)} \\ \mathbf{cov(b_1, b_2)} & \mathbf{var(b_2)} \end{bmatrix}}$$

$$\boldsymbol{cov(b) = \begin{bmatrix} 2.7383886 & -0.6542095 \\ -0.6542095 & 0.1597237 \end{bmatrix}}$$

Find approximate of the information matrix evaluated at $\boldsymbol{b}$

```
Tau_at_b<-solve(vcov(model, dispersion =1))
Tau_at_b

##             (Intercept)      xi
## (Intercept)  17.00         69.6300
## log(i)       69.63         291.4571
```

$$\boldsymbol{\tau = cov(U) = \begin{bmatrix} \mathbf{var(U_1)} & \mathbf{cov(U_1, U_2)} \\ \mathbf{cov(U_1, U_2)} & \mathbf{var(U_2)} \end{bmatrix}}$$

$$\boldsymbol{cov(U) = \begin{bmatrix} 17 & 69.63 \\ 69.63 & 291.4571 \end{bmatrix}}$$

e) For the model fitted in (d), compare the observed values y_i and fitted values $\widehat{y_\iota} = e^{\widehat{\beta_1}+\widehat{\beta_2}\,x_i}$ and use the standardized residuals $r_i = (y_i - \widehat{y_\iota})/\widehat{y_\iota}$ to investigate the adequacy of the model.

```
yhat<-exp(8.4775-1.1093*xi)  #or yhat=fitted.values(m,dispersion =1)
yhat

##  [1] 115.61342 196.90394  85.69042 109.37551  72.55508  55.59615  56.84339
##  [8]  44.04276  76.69304  67.13429  58.76691  32.28352  31.22684  18.74637
## [15]  18.74637  25.57471  18.74637

y<-yi
y

##  [1]  65 156 100 134  16 108 121   4  39 143  56  26  22   1   1   5  65

ri<-(y-yhat)/yhat
ri

##  [1] -0.43778151 -0.20773551  0.16699164  0.22513715 -0.77947788  0.94258047
##  [7]  1.12865547 -0.90917917 -0.49147930  1.13005890 -0.04708284 -0.19463562
## [13] -0.29547788 -0.94665633 -0.94665633 -0.80449437  2.46733841

test_statistc<- sum(ri^2)
test_statistc

## [1] 14.08317

chi_table<-qchisq(1-0.05,17-2)
chi_table

## [1] 24.99579
```

1-Hypothesis:

$$H_0: \textit{Model fit data well} \quad vs \quad H_A: \textit{Model does not fit data well}$$

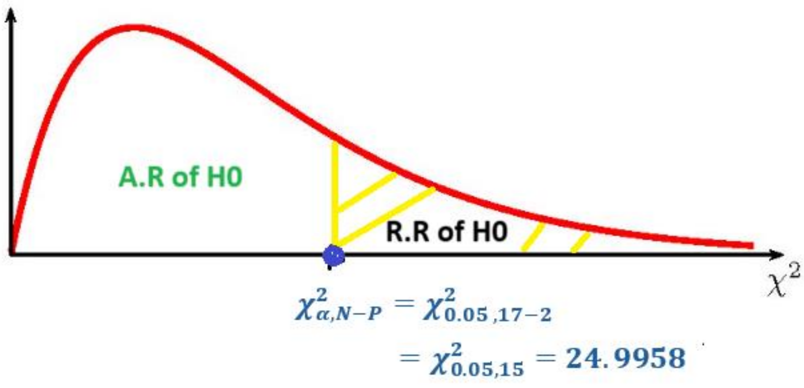
2- Test statistics:

$$\chi^2 = \sum_{i=1}^{17} r_i^2 = 14.0832$$

3- Rejection region of H_0:

$$N = \#\ \textit{of observation} \quad \textit{and} \quad P = \#\ \textit{of parameters}$$

$$\chi^2_{\alpha,N-P} = \chi^2_{0.05\,,17-2} = \chi^2_{0.05,15} = 24.9958$$



4- Since $\sum_{i=1}^{17} r_i^2 < \chi^2$ >> ($\sum_{i=1}^{17} r_i^2 \in Accept\ Area$), we conclude that the model is adequate for describing the data .

Not:

We may use one of the following iterative equations to find an approximate estimate of $\beta = \begin{bmatrix}\beta_0 \\ \beta_1\end{bmatrix}$:

$$b^{(m+1)} = b^{(m)} + [\mathcal{J}^{(m)}]^{-1}\, U^{(m)}$$

or

$$b^{(m+1)} = \left[X^t\, W^{(m)} X\right]^{-1} X^t\, W^{(m)} z^{(m)}$$