

STAT/BIOSTAT 570: Coursework 2 Key

To be submitted to the course canvas site by 11:59pm Monday 18th October, 2021.

1. In this question we will begin to investigate the robustness of the OLS estimator to non-normality of the errors.

Consider the simple linear regression model

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, \dots, n, \quad (1)$$

where the error terms ϵ_i are such that $E[\epsilon_i] = 0$, $\text{var}(\epsilon_i) = \sigma^2$ and $\text{cov}(\epsilon_i, \epsilon_j) = 0$, $i \neq j$. In the following you will consider x_i with: $x_i \sim_{iid} N(20, 3^2)$, with $\beta_0 = 2$ and $\beta_1 = -2.5$ and $n = 15, 30$.

Simulate from model (1) with the error terms ϵ_i i.i.d. from the distributions:

- The normal distribution with mean 0 and variance 2^2 .
- The uniform distribution on the range $(-r, r)$ for $r = 2$.
- A skew normal distribution¹ with $\alpha = 5$, $\omega = 1$ and ξ chosen to give a distribution with mean zero.

- (a) Confirm numerically that the bias is zero.

Solution: For each $n = 15, 30$, we simulated a design matrix X which was held fixed for the remainder of the simulation study. Then, for each error distribution we simulated 20,000 errors. Table 1 below shows the empirical bias of β , its variance as estimated by least squares, and its theoretical variance as estimated across the simulated samples. Note in particular that under normal, uniform, and skewed normal errors, the estimators have bias close to 0.

The results are shown in following table (combined with part a, b, c and d)

¹If $\phi(x)$ and $\Phi(x)$ are the density function and distribution function of a standard normal then the skew normal distribution with location ξ and scale ω is

$$f(x) = \frac{2}{\omega} \phi\left(\frac{x - \xi}{\omega}\right) \Phi\left(\alpha \left(\frac{x - \xi}{\omega}\right)\right).$$

The mean of the distribution is $E[X] = \xi + \omega \delta \sqrt{\frac{2}{\pi}}$, where $\delta = \frac{\alpha}{\sqrt{1 + \alpha^2}}$.

		$n = 15$				$n = 30$			
		Bias	$\text{var}(\hat{\beta})$	$\widehat{\text{var}}(\hat{\beta})$	Coverage	Bias	$\text{var}(\hat{\beta})$	$\widehat{\text{var}}(\hat{\beta})$	Coverage
normal	β_0	0.051	14.987	14.982	0.951	0.013	6.647	6.722	0.952
	β_1	0.003	0.037	0.037	0.950	0.001	0.016	0.016	0.952
uniform	β_0	0.009	4.980	5.011	0.951	0.010	2.231	2.238	0.950
	β_1	0.001	0.012	0.012	0.951	0.000	0.005	0.005	0.949
skew.norm	β_0	0.005	1.501	1.468	0.948	0.008	0.647	0.652	0.950
	β_1	0.000	0.004	0.004	0.948	0.000	0.002	0.002	0.952
Cauchy	β_0	3.450	3080664.378	5140601.117	0.951	0.180	1050381.509	1580011.644	0.952
	β_1	0.161	9710.864	12569.479	0.952	0.041	2610.958	3710.863	0.953

Table 1: Empirical bias, standard errors, and 95% CI coverage based on 20,000 simulated datasets for each error distribution. Note that under normal, uniform, and skewed normal errors, the estimators have bias close to 0.

- (b) Compare the variance of the estimator as reported by least squares, with that which follows from the sampling distribution of the estimator (which you can estimate from the simulations).

Solution: Note in Table 1 that for all three error distributions, the estimator of the standard error appears unbiased.

- (c) Examine the distribution of the resultant estimators (across simulations) of β_0 and β_1 , in particular with respect to normality. For each parameter find the coverage probability of a 95% confidence interval, that is the proportion of times that the confidence intervals contain the true value.

Solutions: Figure 1c shows the Q-Q plots of the expected quantiles and the observed quantiles of the regression estimates. When $n = 15$, for the skewed distribution in particular the tail behaviour is a little off. When $n = 30$, we see that the estimators are approximately normal. The empirical coverage probabilities are shown in Table 1, they are close to the 95% we would expect for both values of n . Since σ^2 is estimated, we used the t_{n-2} quantiles to compute the confidence interval bounds for each error distribution. Note that if we use normal quantiles to compute confidence intervals, we will see conservative confidence intervals in the case of $n = 15$ due to small sample size.

- (d) **Bonus:** Can you “break” least squares? i.e., find a distribution of the errors (with mean zero) that produces poor confidence interval coverage?

Solutions: Using the Cauchy with no defined variance, we were not able to obtain poor coverage of the confidence intervals, yet estimation was very unstable. Some students chose a Bernoulli distribution for the errors, with very negative and positive values (and mean 0) and showed they broke the nominal 95% coverage.

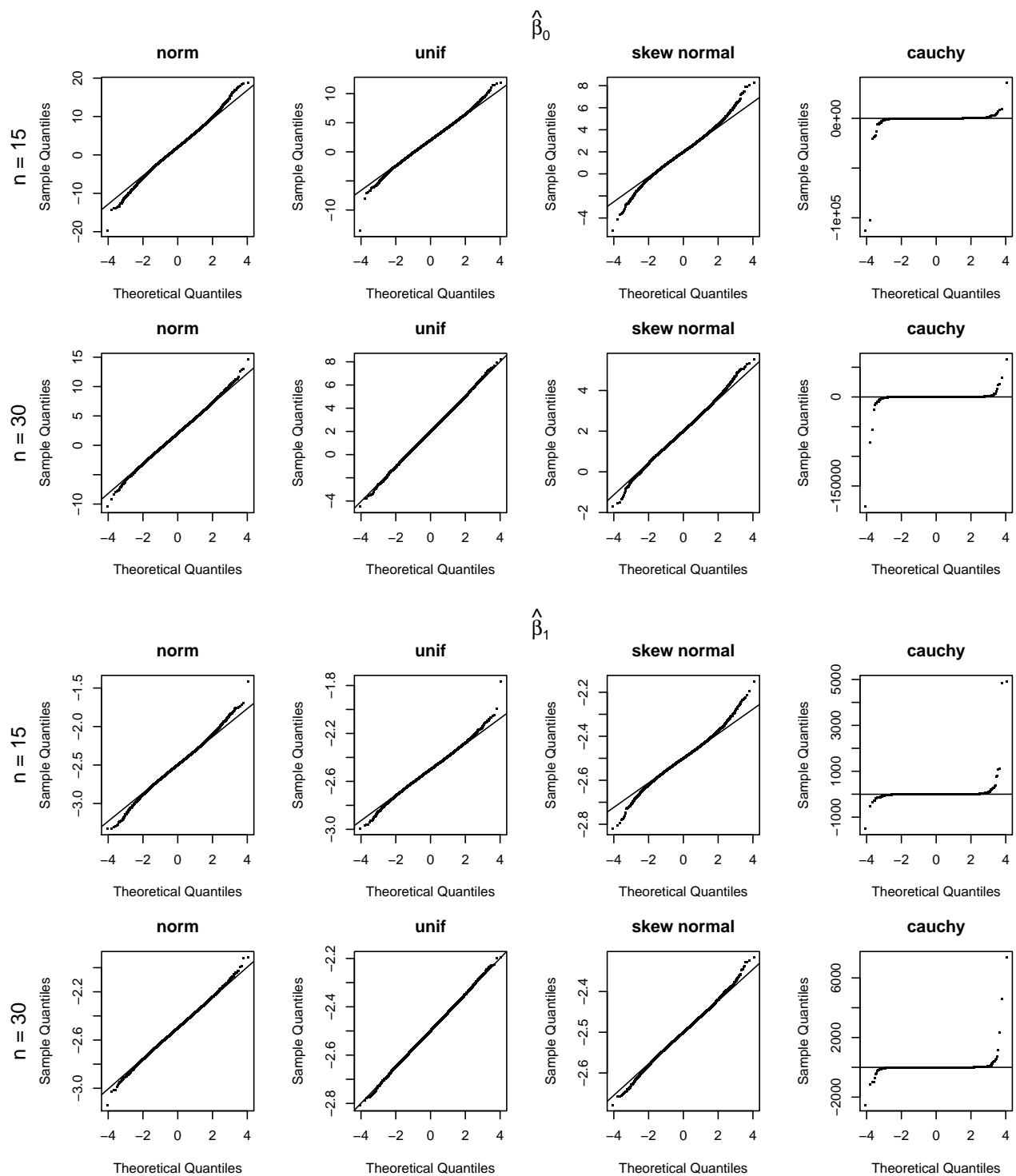


Figure 1: QQ plots of observed estimators versus normal quantiles. The three error distributions specified in the question are close to the asymptotic normal behaviour when $n = 30$. On the contrary, for the Cauchy distribution the normality of the estimators does not hold.

[Note: You should simulate the set of x values once, and then use in all subsequent simulations.]

[Hint: to simulate from a skew normal distribution you may use the function `rsn` in the `sn` package.]

```
##### R CODE Question 1 #####
####
library(sn)
gen.eps <- function(X, dist, b0=2, b1=-2.5){
  n <- dim(X)[1]
  if(dist=="norm") {
    eps <- rnorm(n, 0, 2)
  }else if(dist=="unif") {
    eps <- runif(n, -2, 2)
  }else if(dist=="skew") {
    eps <- rsn(n, xi=-5/sqrt(1+5^2)*sqrt(2/pi), omega=1, alpha=5)
  }else if(dist=="cauchy"){
    eps <- rt(n, df=1)
  }else{
    cat("wrong start")
    break}
  b <- c(b0,b1)
  Y <- X%*%b + eps
  b.hat <- solve(t(X)%*%X) %*% t(X) %*% Y
  # print(b.hat)
  # fit <- lm(Y~0+X)
  # print(fit)
  Y.hat <- X%*%b.hat
  sig2.hat <- sum((Y-Y.hat)^2)/(n-2)
  # if(dist == "norm") print(sig2.hat)
  b.se <- sqrt(diag(sig2.hat * solve(t(X)%*%X)))
  list(beta=b.hat, var=b.se)
}

do.one <- function(n,mu=20,sig=3,print=F){
  x.vec <- rnorm(n, mean=mu, sd=sig)
  X <- cbind(1, x.vec)
  method <- c("norm","unif","skew","cauchy")
  rst <- matrix(NA,ncol = 4, nrow = 4)
```

```

colnames(rst) <- method
for(ss in method){
test <- gen.eps(X,ss)
rst[,ss] <- c(test$beta, test$var)
}
if(print==T)print(rst)
return(as.vector(rst))
}
set.seed(2)
result.15 <- replicate(20000, do.one(15))
result.30 <- replicate(20000, do.one(30))

```

```

make.table <- function(result, dist, n){
if(dist=="norm") {
tbl <- result[1:4,]
}else if(dist=="unif") {
tbl <- result[5:8,]
}else if(dist=="skew") {
tbl <- result[9:12,]
}else if(dist=="cauchy"){
tbl <- result[13:16,]
}
# se <- sqrt(tbl[3:4,])
t.stat <- tbl[1:2,]/tbl[3:4,]
ci1 <- tbl[1,] + tbl[3,] %o% c(qt(0.025,df=n-2), qt(0.975,df=n-2))
ci2 <- tbl[2,] + tbl[4,] %o% c(qt(0.025,df=n-2), qt(0.975,df=n-2))
cover1 <- 2>=ci1[,1] & 2<=ci1[,2]
cover2 <- -2.5>=ci2[,1] & -2.5<=ci2[,2]
mean <- apply(tbl,1,mean)
var <- apply(tbl,1,var)
var.hat <- apply(tbl^2,1,mean)
rst <- cbind(bias=abs(mean[1:2]-c(2,-2.5)),var=var[1:2],var.hat=var.hat[3:4],
cover=c(mean(cover1),mean(cover2)))
return(rst)
}

```

```

table <- cbind(rbind(make.table(result.15,"norm",15),
make.table(result.15,"unif",15),
make.table(result.15,"skew",15),

```

```

make.table(result.15,"cauchy",15)),
rbind(make.table(result.30,"norm",30),
make.table(result.30,"unif",30),
make.table(result.30,"skew",30),
make.table(result.30,"cauchy",30)))
library(xtable)
print(xtable(table, digits=3))

method=c("norm","unif","skew normal","cauchy")
pdf("qqplot.pdf",height=9.5, width=8.5)
par(mar=c(4,4,3,2)+0.1, oma=c(0,2,1,0))
layout(matrix(c(1:8, rep(9,4), 10:17), 5, 4, byrow=TRUE), heights=c(6,6,1,6,6))

for(i in 1:4){
qqnorm(result.15[i*4-3,],main=method[i], pch='.')
qqline(result.15[i*4-3,])
if(i == 1) mtext(paste("n =", 15), side=2, line=4.5)
}

for(i in 1:4){
qqnorm(result.30[i*4-3,],main=method[i], pch='.')
qqline(result.30[i*4-3,])
if(i == 1) mtext(paste("n =", 30), side=2, line=4.5)
}
mtext(expression(hat(beta)[0]), side=3, line=-1, outer=TRUE)
par(mar=c(0,0,0,0))
plot.new()
par(mar=c(4,4,3,2)+0.1)
for(i in 1:4){
qqnorm(result.15[i*4-2,],main=method[i], pch='.')
qqline(result.15[i*4-2,])
if(i == 1) mtext(paste("n =", 15), side=2, line=4.5)
}
for(i in 1:4){
qqnorm(result.30[i*4-2,],main=method[i], pch='.')
qqline(result.30[i*4-2,])
if(i == 1) mtext(paste("n =", 30), side=2, line=4.5)
}
mtext(expression(hat(beta)[1]), side=3, line=-38, outer=TRUE)
dev.off()

```

2. Consider the exponential regression problem with independent responses

$$p(y \mid \lambda_i) = \lambda_i e^{-\lambda_i y}, \quad y > 0$$

and $\log \lambda_i = \beta_0 + \beta_1 x_i$ for given covariates $x_i, i = 1, \dots, n$. We wish to estimate the 2×1 regression parameter $\beta = [\beta_0, \beta_1]^\top$ using MLE.

- Find expressions for the likelihood function $L(\beta)$, log likelihood function $l(\beta)$, score function $S(\beta)$ and Fisher's information matrix $I(\beta)$.
- Find expressions for the maximum likelihood estimate $\hat{\beta}$. If no closed form solution exists, then instead provide a functional form that could be simply implemented for solution.
- For the data in Table 2, numerically maximize the likelihood function to obtain estimates of β . These data consist of the survival times (y) of rats as a function of concentrations of a contaminant (x). Find the asymptotic covariance matrix for your estimate using the information $I(\beta)$. Provide a 95% confidence interval for each element of β_0 and β_1 .

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_i	6.1	4.2	0.5	8.8	1.5	9.2	8.5	8.7	6.7	6.5	6.3	6.7	0.2	8.7	7.5
y_i	0.8	3.5	12.4	1.1	8.9	2.4	0.1	0.4	3.5	8.3	2.6	1.5	16.6	0.1	1.3

Table 2: Survival times y_i and concentrations of a contaminant x_i for $i = 1, \dots, 15$.

- Plot the log-likelihood function $l(\beta_0, \beta_1)$ and compare with the log of the asymptotic normal approximation to the sampling distribution of the MLE.
- Find the maximum likelihood estimate $\hat{\beta}_0$ under the null hypothesis $H_0 : \beta_1 = 0$.
- Perform score, likelihood ratio, and Wald tests of the null hypothesis $H_0 : \beta_1 = 0$ with $\alpha = 0.05$. In all cases explicitly indicate the formula you use to compute the test statistic.
- Summarize the results of the estimation and hypothesis testing presented above in a manner that would address the question of whether increasing concentrations of the contaminant had an effect on a rat's life expectancy.

Solution:

- 3 points** Preliminaries: For independent response variables Y_i and known predictor variables $X_i, i = 1, \dots, n$ we have

$$\begin{aligned}
Y_i &\sim f_i(y; \lambda_i) = \lambda_i e^{-\lambda_i y}, \quad y > 0 \\
\log(\lambda_i) &= \beta_0 + \beta_1 X_i \\
\lambda_i &= e^{\beta_0} e^{\beta_1 X_i}
\end{aligned}$$

Under the above assumptions, the expectation of Y_i is $E(Y_i) = \frac{1}{\lambda_i} = e^{-\beta_0} e^{-\beta_1 X_i}$. In order to use the chain rule in differentiating the log likelihood, we note

$$\begin{aligned}
\frac{\partial \lambda_i}{\partial \beta_0} &= e^{\beta_0} e^{\beta_1 X_i} = \lambda_i \\
\frac{\partial \lambda_i}{\partial \beta_1} &= X_i e^{\beta_0} e^{\beta_1 X_i} = X_i \lambda_i
\end{aligned}$$

(i) Likelihood function

$$L(\boldsymbol{\beta}) = \prod_{i=1}^n \lambda_i e^{-\lambda_i Y_i} = \prod_{i=1}^n \exp\{\beta_0 + \beta_1 X_i - e^{\beta_0} e^{\beta_1 X_i} Y_i\}$$

(ii) Log likelihood function

$$\begin{aligned}
l(\vec{\beta}) &= \sum_{i=1}^n \log(\lambda_i) - \lambda_i Y_i = \sum_{i=1}^n \beta_0 + \beta_1 X_i - e^{\beta_0} e^{\beta_1 X_i} Y_i \\
&= n\beta_0 + \beta_1 \sum_{i=1}^n X_i - e^{\beta_0} \sum_{i=1}^n e^{\beta_1 X_i} Y_i
\end{aligned}$$

(iii) Score function

$$\begin{aligned}
U_1(\boldsymbol{\beta}) &= \frac{\partial l(\boldsymbol{\beta})}{\partial \beta_0} = \sum_{i=1}^n \frac{\partial}{\partial \beta_0} (\log(\lambda_i) - \lambda_i Y_i) \\
&= \sum_{i=1}^n \frac{\partial}{\partial \lambda_i} (\log(\lambda_i) - \lambda_i Y_i) \frac{\partial \lambda_i}{\partial \beta_0} \\
&= \sum_{i=1}^n \left(\frac{1}{\lambda_i} - Y_i \right) (\lambda_i) = \sum_{i=1}^n (1 - \lambda_i Y_i) \\
&= n - e^{\beta_0} \sum_{i=1}^n e^{\beta_1 X_i} Y_i
\end{aligned}$$

$$\begin{aligned}
U_2(\boldsymbol{\beta}) &= \frac{\partial l(\boldsymbol{\beta})}{\partial \beta_1} = \sum_{i=1}^n \frac{\partial}{\partial \beta_1} (\log(\lambda_i) - \lambda_i Y_i) \\
&= \sum_{i=1}^n \frac{\partial}{\partial \lambda_i} (\log(\lambda_i) - \lambda_i Y_i) \frac{\partial \lambda_i}{\partial \beta_1} \\
&= \sum_{i=1}^n \left(\frac{1}{\lambda_i} - Y_i \right) (X_i \lambda_i) = \sum_{i=1}^n (X_i - X_i \lambda_i Y_i) \\
&= \sum_{i=1}^n X_i - \sum_{i=1}^n X_i e^{\beta_0} e^{\beta_1 X_i} Y_i
\end{aligned}$$

(iv) Observed information $I^*(\boldsymbol{\beta})$

$$\begin{aligned}
I_{11}^*(\boldsymbol{\beta}) &= -\frac{\partial^2}{\partial \beta_0^2} l(\boldsymbol{\beta}) = -\frac{\partial}{\partial \beta_0} U_1(\boldsymbol{\beta}) = \sum_{i=1}^n e^{\beta_0} e^{\beta_1 X_i} Y_i \\
I_{12}^*(\boldsymbol{\beta}) &= -\frac{\partial^2}{\partial \beta_1 \partial \beta_0} L(\boldsymbol{\beta}) = -\frac{\partial}{\partial \beta_1} U_1(\boldsymbol{\beta}) = \sum_{i=1}^n X_i e^{\beta_0} e^{\beta_1 X_i} Y_i \\
I_{21}^*(\boldsymbol{\beta}) &= I_{12}(\boldsymbol{\beta}) = \sum_{i=1}^n X_i e^{\beta_0} e^{\beta_1 X_i} Y_i \\
I_{22}^*(\boldsymbol{\beta}) &= -\frac{\partial^2}{\partial \beta_1^2} L(\boldsymbol{\beta}) = -\frac{\partial}{\partial \beta_1} U_2(\boldsymbol{\beta}) = \sum_{i=1}^n X_i^2 e^{\beta_0} e^{\beta_1 X_i} Y_i
\end{aligned}$$

(v) Fisher's (expected) information $I(\boldsymbol{\beta}) = EI^*(\boldsymbol{\beta})$

$$\begin{aligned}
I_{11}(\boldsymbol{\beta}) &= EI_{11}(\boldsymbol{\beta}) = \sum_{i=1}^n e^{\beta_0} e^{\beta_1 X_i} E(Y_i) \\
&= \sum_{i=1}^n e^{\beta_0} e^{\beta_1 X_i} e^{-\beta_0} e^{-\beta_1 X_i} = n
\end{aligned}$$

$$I_{12}(\boldsymbol{\beta}) = EI_{12}(\boldsymbol{\beta}) = \sum_{i=1}^n X_i e^{\beta_0} e^{\beta_1 X_i} E(Y_i) = \sum_{i=1}^n X_i$$

$$I_{21}(\boldsymbol{\beta}) = I_{12}(\boldsymbol{\beta}) = \sum_{i=1}^n X_i$$

$$I_{22}(\boldsymbol{\beta}) = EI_{22}(\boldsymbol{\beta}) = \sum_{i=1}^n X_i^2 e^{\beta_0} e^{\beta_1 X_i} E(Y_i) = \sum_{i=1}^n X_i^2$$

(b) **1 point** The maximum likelihood estimate $\hat{\boldsymbol{\beta}}$ will satisfy $U(\hat{\boldsymbol{\beta}}) = 0$. Hence

$$U_1(\hat{\beta}) = 0 \Rightarrow \hat{\beta}_0 = \log \left(\frac{n}{\sum_{i=1}^n e^{\hat{\beta}_1 X_i} Y_i} \right)$$

$$U_2(\hat{\beta}) = 0 \Rightarrow \sum_{i=1}^n e^{\hat{\beta}_1 X_i} Y_i \sum_{i=1}^n X_i - n \sum_{i=1}^n X_i e^{\hat{\beta}_1 X_i} Y_i = 0$$

Note the second equation cannot be represented as a closed form solution for $\hat{\beta}_1$.

(c) **2 points**

- (i) 95% confidence intervals for the regression parameters using the expected information are given by:

```

      MLE      low.lim      up.lim
beta0 -2.8209327 -3.9712909 -1.6705746
beta1  0.3013552  0.1293685  0.4733418

```

where the estimated covariance matrix using the expected information was

```

[,1]      [,2]
[1,]  0.34448471 -0.046251617
[2,] -0.04625162  0.007700047

```

- (ii) 95% confidence intervals for the regression parameters using the observed information are given by:

```

      MLE      low.lim      up.lim
beta0 -2.8209327 -4.0151390 -1.626726
beta1  0.3013552  0.1212733  0.481437

```

where the estimated covariance matrix using the observed information was

```

[,1]      [,2]
[1,]  0.37133865 -0.050722322
[2,] -0.05072232  0.008444341

```

- (d) **1 point** Profile log-likelihoods and their asymptotic approximations for univariate and bivariate parameters are given in Figure 2 below.

- (e) **1 point** From part (b), we have that under $H_0 : \beta_1 = 0$,

$$\hat{\beta}_0^* = \log \left(\frac{n}{\sum_{i=1}^n Y_i} \right) = -1.44299$$

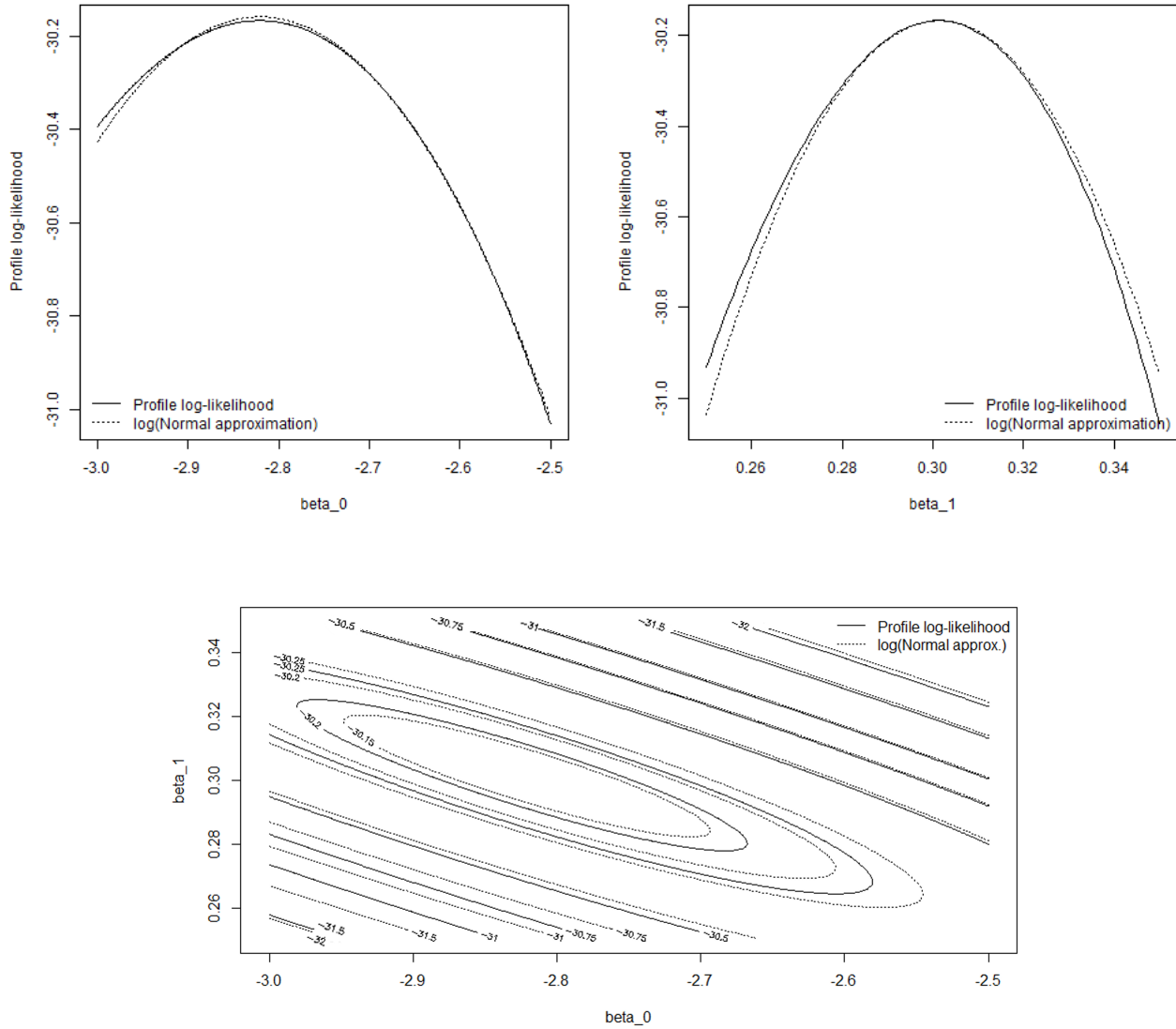


Figure 2: Top-left: profile log-likelihood for β_0 with $\beta_1 = \hat{\beta}_1$. Top-right: profile log-likelihood for β_1 with $\hat{\beta}_0$. Bottom: profile log-likelihood for (β_0, β_1) .

(f) **2 points** Testing the hypothesis $H_0 : \beta_1 = 0$

(ia) The score statistic using the expected information was calculated as

$$R \equiv U(\hat{\beta}_0^*, 0)^T I^{-1}(\hat{\beta}_0^*, 0)_{2,2} U(\hat{\beta}_0^*, 0) = 15.26$$

where R follows a χ_1^2 distribution under H_0 , yielding a p-value of $< .001$.

(ib) The score statistic using the observed information was calculated as

$$R^* \equiv U(\hat{\beta}_0^*, 0)^T I^{*-1}(\hat{\beta}_0^*, 0)_{2,2} U(\hat{\beta}_0^*, 0) = 13.67$$

where R^* follows a χ_1^2 distribution under H_0 , yielding a p-value of $< .001$.

(ii) The likelihood ratio statistic was calculated as

$$T \equiv -2 \log \left(\frac{L(\hat{\beta}_0^*, 0)}{L(\hat{\beta})} \right) = 12.96$$

where T follows a χ_1^2 distribution under H_0 , yielding a p-value of $< .001$.

(iiia) The Wald statistic using the expected information was calculated as

$$W \equiv \hat{\beta}_1^T \left[I^{-1}(\hat{\beta}_0^*, 0)_{2,2} \right]^{-1} \hat{\beta}_1 = 11.79$$

where W follows a χ_1^2 distribution under H_0 , yielding a p-value of $< .001$.

(iiib) The Wald statistic using the observed information was calculated as

$$W^* \equiv \hat{\beta}_1^T \left[I^{*-1}(\hat{\beta}_0^*, 0)_{2,2} \right]^{-1} \hat{\beta}_1 = 13.16$$

where W^* follows a χ_1^2 distribution, yielding a p-value of $< .001$.

- (g) **3 points** In our assumed model we have $E[Y|\beta, x] = \exp(-\beta_0 - \beta_1 x)$. Then, the estimated mean life expectancy for rats unexposed to the contaminant was 16.79 months ($e^{-\hat{\beta}_0}$), and the survival times of rats was found to be significantly associated with concentration of the contaminant ($P < .0001$). Also, the ratio of mean life expectancy of rats with change in contaminant concentration from $x - 1$ to x is $\frac{E(Y|x)}{E(Y|x-1)} = \frac{e^{-\beta_0 - \beta_1 x}}{e^{-\beta_0 - \beta_1 (x-1)}} = e^{-\beta_1}$ which is estimated as $e^{-0.30} = 74\%$ (95% CI: $e^{-0.47}$ to $e^{-0.13} = 62\%$ to 88%). That is, a population of rats exposed to contamination at a concentration 1 unit higher than an other group would be estimated to have a life expectancy 26% (95% CI: 12% to 38%) lower than that of the group with 1 unit lower level of exposure to the contaminant.

```

##### R CODE Question 2 #####
##### Define functions to calculate the loglikelihood, score and info matrix #####
#####
##
getLogL <- function(x, y, beta0, beta1){
  n <- length(x)
  n*beta0 + beta1*sum(x) - sum(exp(beta0+beta1*x)*y)
}

getScore <- function(x, y, beta0, beta1){
  n <- length(y)
  U1 <- n - sum(exp(beta0+beta1*x)*y)
  U2 <- sum(x) - sum(x*y*exp(beta0+beta1*x))
  c( U1, U2 )
}

getInfo <- function(x, y, beta0, beta1, type="expected"){
  n <- length(y)
  if ( type == "expected" ){
    I.11 <- n
    I.12 <- I.21 <- sum( x )
    I.22 <- sum( x^2 )
  }
  else if ( type == "observed" ){
    I.11 <- sum( exp( beta0 + beta1 * x ) * y )
    I.12 <- I.21 <- sum( x * y * exp(beta0 + beta1 * x ) )
    I.22 <- sum( x^2 * y * exp( beta0 + beta1 * x ) )
  }
  return( matrix( c( I.11, I.12, I.21, I.22 ), nrow=2 ) )
}
##
#####
##### Function to calculate confidence intervals for regression parameters #####
#####
##
getCIbeta <- function( beta, varBeta ){
  low.lim <- beta - qnorm( .975 ) * sqrt( diag( varBeta ) )
  up.lim <- beta + qnorm( .975 ) * sqrt( diag( varBeta ) )
  rslt <- cbind( beta, low.lim, up.lim )
  dimnames( rslt ) <- list( c( "beta0", "beta1" ), c( "MLE", "low.lim", "up.lim" ) )
}

```

```

    rslt
  }
  ##
  #####
  #####  Read in the data  #####
  #####
  ##
  x <- c( 6.1, 4.2, 0.5, 8.8, 1.5, 9.2, 8.5, 8.7, 6.7, 6.5, 6.3, 6.7, 0.2, 8.7, 7.5 )
  y <- c( 0.8, 3.5, 12.4, 1.1, 8.9, 2.4, 0.1, 0.4, 3.5, 8.3, 2.6, 1.5, 16.6, 0.1, 1.3 )
  ##
  #####
  #####  Part c:  Obtaining MLEs and C.I.'s  #####
  #####
  ##
  minusLogL <- function(beta){
    -getLogL(x,y,beta[1],beta[2])
  }
  mle <- optim(par=c(0,0),fn=minusLogL)$par
  varBetaExp <- solve( getInfo( x, y, mle[1], mle[2], type="expected" ) )
  varBetaObs <- solve( getInfo( x, y, mle[1], mle[2], type="observed" ) )
  getCIbeta( mle, varBetaExp )
  varBetaExp
  getCIbeta( mle, varBetaObs )
  varBetaObs
  ##
  #####
  #####  Part d:  Plot profile log-likelihoods and (log) asymptotic ditributions  #####
  #####
  ##
  beta0 <- seq(-3.0, -2.5, by=.001)
  beta1 <- seq(0.25, 0.35, by=.001)
  ##
  #####  Profile log-likelihood for beta0  #####
  ##
  png("beta0.png")
  rslt <- NULL
  for ( i in beta0 ){
    rslt <- rbind( rslt, c( i, mle[2], getLogL( x, y, i, mle[2] ) ) )
  }
  # Fix beta1= MLE, mle[2]

```

```

plot( rslt[,1], rslt[,3], type="l", xlab="beta_0", ylab="Profile log-likelihood" )

temp <- log( dnorm( rslt[,1], mle[1], sd=sqrt(varBetaExp[1,1])) )
std.temp <- ( temp - mean( temp ) ) / sd( temp )
lines( rslt[,1], std.temp * sd( rslt[,3]) + mean( rslt[,3] ), lty=3 )
legend('bottomleft', legend=c("Profile log-likelihood", "log(Normal approximation)" ),
      bty='n')
dev.off()

##
##### Profile log-likelihood for beta1 #####
##
png("beta1.png")
rslt <- NULL
for ( i in beta1 ){
  rslt <- rbind( rslt, c( mle[1], i, getLogL( x, y, mle[1], i ) ) )
}
plot( rslt[,2], rslt[,3], type="l", xlab="beta_1", ylab="Profile log-likelihood" )
temp <- log( dnorm( rslt[,2], mle[2], sd=sqrt( varBetaExp[2,2] ) ) )
std.temp <- ( temp - mean( temp ) ) / sd( temp )
lines( rslt[,2], std.temp * sd( rslt[,3]) + mean( rslt[,3] ), lty=3 )
legend('bottomright', legend=c("Profile log-likelihood", "log(Normal approximation)" ),
      bty='n')
dev.off()

##
##### Profile log-likelihood for (beta0, beta1)
##

rslt <- matrix(nrow=length(beta0),ncol=length(beta1))
for ( i in 1:length(beta0) ){
  for ( j in 1:length(beta1) ){
    rslt[i,j] <- getLogL( x, y, beta0[i], beta1[j] )
  }
}

temp<- matrix(nrow=length(beta0),ncol=length(beta1))
for ( i in 1:length(beta0) ){
  for ( j in 1:length(beta1) ){
    temp[i,j] <- log( 1/sqrt(2*pi*det(varBetaExp)) * exp(-0.5*(c(beta0[i],beta1[j])-
mle)%*(solve(varBetaExp)%*(c(beta0[i],beta1[j])-mle))))
  }
}

```



```

}
std.temp <- ( temp - mean( temp ) ) / sd( temp )
png("beta01.png")
contour(beta0,beta1,rslt, xlab="beta_0", ylab="beta_1",method="edge",
levels = c(-32,-31.5,-31,-30.75,-30.5,-30.25,-30.2,-30))
contour(beta0,beta1,std.temp * sd(rslt) + mean(rslt),lty=3,add=TRUE,method="edge",
levels = c(-32,-31.5,-31,-30.75,-30.5,-30.25,-30.2,-30.15),
labels=c(-32,-31.5,-31,-30.75,"",-30.25,-30.2,-30.15))
legend('topright', legend=c("Profile log-likelihood", "log(Normal approx.)" ),
bty='n',lty=c(1,3))
dev.off()
##
#####
##### Part e: Score, LR, and Wald tests of Ho: beta = 0 #####
beta0.r <- log( length(y) / sum( y ) )
##
#####
##### Part f: Score, LR, and Wald tests of Ho: beta1 = 0 #####
#####
##
##### Score test #####
##
## Using expected info ##
##
U <- getScore( x, y, beta0.r, 0 )
I <- getInfo( x, y, beta0.r, 0, type="expected" )
score.stat <- t( U[2] ) %*% solve( I )[2,2] %*% U[2]
p.value <- 1 - pchisq( score.stat, 1 )
c( score.stat, p.value )
##
## Using observed info ##
##
U <- getScore( x, y, beta0.r, 0 )
I <- getInfo( x, y, beta0.r, 0, type="observed" )
score.stat <- t( U[2] ) %*% solve( I )[2,2] %*% U[2]
p.value <- 1 - pchisq( score.stat, 1 )
c( score.stat, p.value )
##
##### LR test #####
##

```

```

logL0 <- getLogL(x,y,beta0=beta0.r,beta1=0)
logL1 <- getLogL(x,y,beta0=mle[1],beta1=mle[2])
LR.stat <- -2*(logL0-logL1)
p.value <- 1 - pchisq( LR.stat, 1 )
c( LR.stat, p.value )
##
##### Wald test #####
##
## Using expected info #####
##
I <- getInfo( x, y, beta0.r, 0, type="expected" )
wald.stat <- t( mle[2] ) %*% solve( I )[2,2]^{-1} %*% mle[2]
p.value <- 1 - pchisq( wald.stat, 1 )
c( wald.stat, p.value )
##
## Using observed info #####
##
I <- getInfo( x, y, beta0.r, 0, type="observed" )
wald.stat <- t( mle[2] ) %*% solve( I )[2,2]^{-1} %*% mle[2]
p.value <- 1 - pchisq( wald.stat, 1 )
c( wald.stat, p.value )

```