Homework 2

2022-10-19

Question 1

```
set.seed(195021)
x = rexp(n=50, rate=2)
```

1.1)

```
x_bar = mean(x)

f = function(lamb, x_bar, n){return(- n*log(lamb) + lamb*n*x_bar) }
ans = optimise(f=f, n=length(x), x_bar=x_bar, interval=c(0,100000))
cbind(ans$minimum, 1/x_bar)
```

```
##         [,1]   [,2]
## [1,] 3.2472 3.2472
```

My estimates is same as $\frac{1}{\bar{x}}$

1.2)

```
library(numDeriv)
lambda = ans$minimum
sigma <- sqrt(solve(hessian(f, lambda, x_bar=x_bar, n=length(x))))
upper <- lambda + 1.96 * sigma
lower <- lambda - 1.96 * sigma
cat(lower, upper, '\n')
```

```
## 2.347122 4.147278
```

The approximate 95% CI for the estimate is $[2.347122, 4.147278]$.

Question 2

2.1)

Using the gout data set, fit a logistic regression for gout using sex, age, and race as predictors (for this you can use glm(), don't forget the link!).

```
data = read.csv('../../DATA/goutData.txt', sep = ' ')
data$gout = ifelse(data$gout == 'Y', 1, 0)
data$sex = ifelse(data$sex == 'F', 1, 0)
data$race = ifelse(data$race == 'B', 1, 0)
model = glm(gout~sex+age+race,family=binomial(link=logit), data=data)
summary(model)
```

```
##
## Call:
```

```
## glm(formula = gout ~ sex + age + race, family = binomial(link = logit),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8294  -0.4256  -0.3537  -0.2713   2.6115
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.37823    2.40051  -3.490 0.000483 ***
## sex         -0.44915    0.38821  -1.157 0.247288
## age          0.09239    0.03644   2.536 0.011227 *
## race         0.74329    0.41914   1.773 0.076168 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 213.11  on 399  degrees of freedom
## Residual deviance: 203.05  on 396  degrees of freedom
## AIC: 211.05
##
## Number of Fisher Scoring iterations: 5
```

2.2)

Calculate the covariance matrix:

```
COV = vcov(model)
```

Compute the predicted risk and 95% CI:

```
test = data.frame(race=c('W', 'W', 'B', 'B'),
                  sex=c('M', 'F', 'M', 'F'),
                  age=c(55, 55, 55, 55))
result = test
test$sex = ifelse(test$sex == 'F', 1, 0)
test$race = ifelse(test$race == 'B', 1, 0)
result$predicted_risk = predict(model, test, type = "response")

test.matrix = as.matrix(cbind(1, test$sex, test$age, test$race)) # include the intercept
SEeta = diag(sqrt(test.matrix %*% COV %*% t(test.matrix)))

log_odds = predict(model, test)
log_odds_upper = log_odds + 1.96 * SEeta
log_odds_lower = log_odds - 1.96 * SEeta

risk_upper = exp(log_odds_upper)/(1+exp(log_odds_upper))
risk_lower = exp(log_odds_lower)/(1+exp(log_odds_lower))

result$percent95CI = rep(0, 4)
for (i in 1:4){
  result$percent95CI[i] = paste('(', risk_lower[i], ',', risk_upper[i],  ')')
}
```

```
result
```

```
##   race sex age predicted_risk                                    percent95CI
## 1    W   M  55     0.03568382  ( 0.0143371592240793 , 0.0860392310566476 )
## 2    W   F  55     0.02307028 ( 0.00896344415281758 , 0.0580776176387717 )
## 3    B   M  55     0.07219612   ( 0.0273233949591836 , 0.177327511035585 )
## 4    B   F  55     0.04730937   ( 0.0184214349436718 , 0.116138541574751 )
```

Note, the `risk` here means the probability of gout for each of the person in the test dataset and it do not means odds.

Question 3

Use 1,000 bootstrap samples to estimate the SE:

```
data = read.csv('../../DATA/goutData.txt', sep = ' ')
data$gout = ifelse(data$gout == 'Y', 1, 0)
data$sex = ifelse(data$sex == 'F', 1, 0)
data$race = ifelse(data$race == 'B', 1, 0)

model = glm(gout~sex+age+race,family=binomial(link=logit), data=data)
set.seed(123)
n = dim(data)[1]

B = 1000 # numbrer of bootstrap samples
SEetameans = matrix(rep(0, B*4), nrow = B)

test = data.frame(race=c('W', 'W', 'B', 'B'),
                  sex=c('M', 'F', 'M', 'F'),
                  age=c(55, 55, 55, 55))
test$sex = ifelse(test$sex == 'F', 1, 0)
test$race = ifelse(test$race == 'B', 1, 0)
test.matrix = as.matrix(cbind(1, test$sex, test$age, test$race))

for(i in 1:B){
  index = sample(1:n, size = n, replace = TRUE)
  modelQ3 = glm(gout~sex+age+race,family=binomial(link=logit), data=data[index, ])
  COV = vcov(modelQ3)
  SEeta = diag(sqrt(test.matrix %*% COV %*% t(test.matrix)))

  SEetameans[i, ] = SEeta
}
```

The estimated SE obtained by bootstrap is:

```
SEeta = colMeans(SEetameans)
SEeta
```

```
## [1] 0.4941875 0.5105432 0.5399246 0.5166063
```

The 95% CIs:

```
log_odds = predict(model, test)
log_odds_upper = log_odds + 1.96 * SEeta
log_odds_lower = log_odds - 1.96 * SEeta

risk_upper = exp(log_odds_upper)/(1+exp(log_odds_upper))
```

```r
risk_lower = exp(log_odds_lower)/(1+exp(log_odds_lower))

answerQ3 = data.frame(percent95CI_from_Q2=result$percent95CI)
answerQ3$percent95CI_from_Q3 = rep(0, 4)

for (i in 1:4){
  answerQ3$percent95CI_from_Q3[i] = paste('(', risk_lower[i], ',', risk_upper[i],  ')')
}

answerQ3
```

```
##                            percent95CI_from_Q2
## 1  ( 0.0143371592240793 , 0.0860392310566476 )
## 2 ( 0.00896344415281758 , 0.0580776176387717 )
## 3   ( 0.0273233949591836 , 0.177327511035585 )
## 4   ( 0.0184214349436718 , 0.116138541574751 )
##                            percent95CI_from_Q3
## 1   ( 0.0138526477155667 , 0.088821141074193 )
## 2 ( 0.00860700669025921 , 0.0603580265522246 )
## 3   ( 0.0262961064222747 , 0.183145240013323 )
## 4   ( 0.0177209099752763 , 0.120253320795713 )
```

We can see from the results that the 95% confidence interval obtained by bootstrap is almost the same as the result obtained from the second question.