STAT 5703 Homework 1

Shijie He(sh3975), Yunjun Xia(yx2569), Shuyu Huang(sh3967)

2/6/2020

Exercise 1

No source code

Exercise 2

(7)

```
samplea = c()
s_a = c()
for (i in 1:500){
  x = rpois(50,5)
  s_a = c(s_a, var(x))
  samplea = c(samplea, mean(x))
}

mean_a = mean(samplea)
s_a = mean(s_a)

sampleb = c()
s_b = c()
for (i in 1:500){
  theta = rgamma(50, shape = 2.5, rate = 0.5)
  y = c()
  for (j in 1:50){
    y = c(y, rpois(1, theta[j]))
  }
  sampleb = c(sampleb, mean(y))
  s_b = c(s_b, var(y))
}

mean_b = mean(sampleb)
s_b = mean(s_b)

cat("Test statistic for model a is: ", sqrt(500/2)*(s_a-mean_a)/mean_a, "\n")
```

```
## Test statistic for model a is:  -0.2030737
```

```
cat("Test statistic for model b is: ", sqrt(500/2)*(s_b-mean_b)/mean_b)
```

```
## Test statistic for model b is:  31.26688
```

(8)

```
dpd = c(rep(0,1), rep(1,4), rep(2,15), rep(3,31), rep(4,39), rep(5,55), rep(6,54),
        rep(7,49), rep(8,47), rep(9,31), rep(10,16), rep(11, 9), rep(12, 8), rep(13, 4), rep(14, 3))

n = length(dpd)
s = var(dpd)
m = mean(dpd)

cat("Test statistic is: ", sqrt(n/2)*(s-m)/m, "\n")
```

```
## Test statistic is:  0.9786746
```

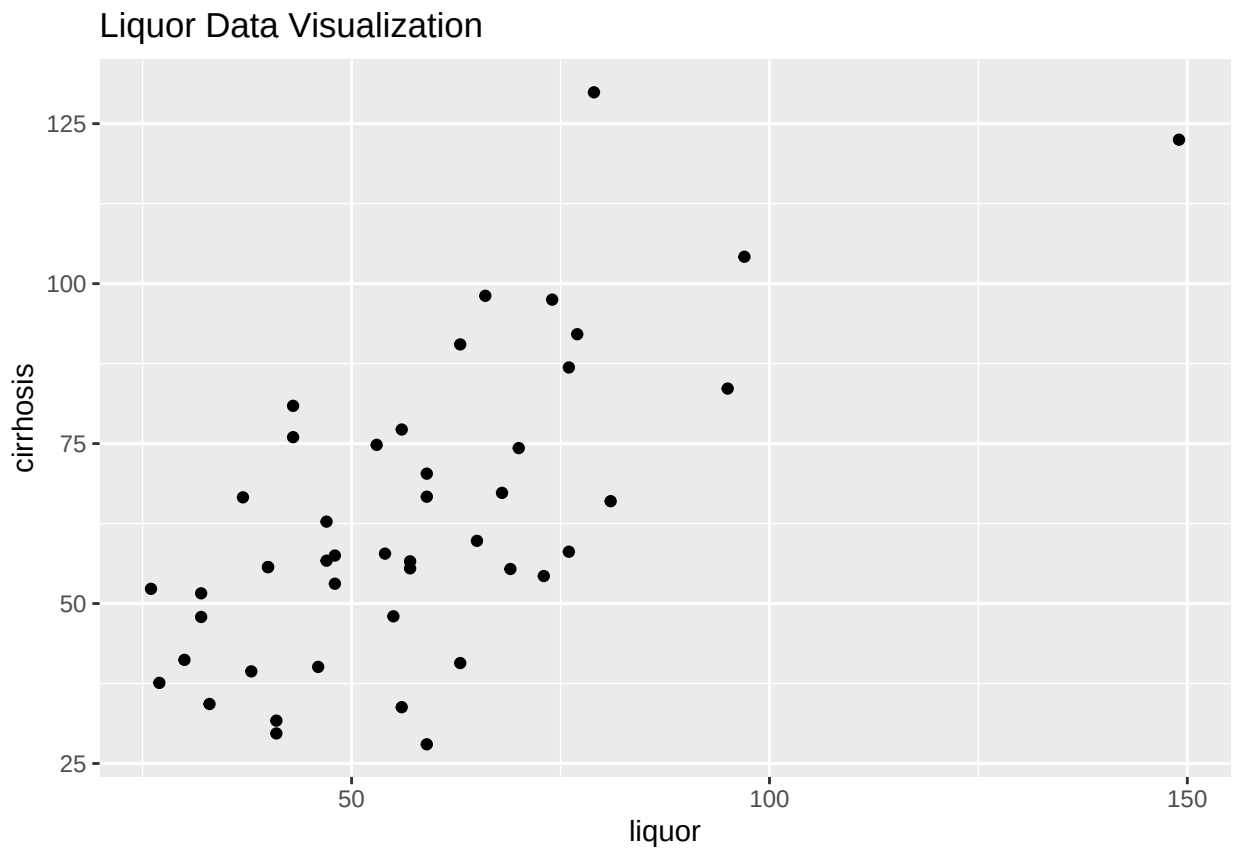
Exercise 3

No Source Code

Exercise 4

1. Visualize the data and discuss the pertinence of fitting a straight line to this data set.

```
ggplot(data=liver, aes(x = liquor, y =cirrhosis) )+
  geom_point()+
  ggtitle('Liquor Data Visualization')
```



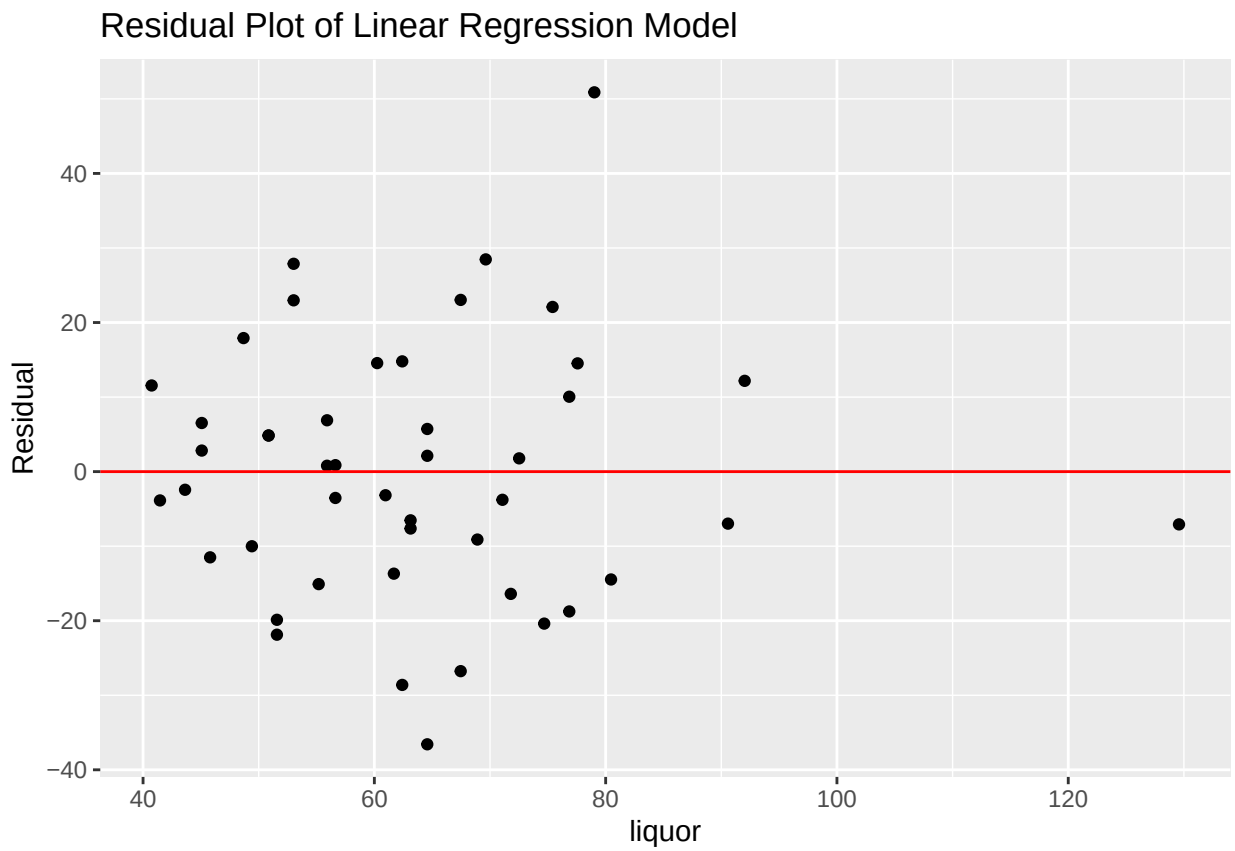
5.

```
liquor_fit<-lm(cirrhosis~liquor, data=liver)
summary(liquor_fit)
```

```
##
## Call:
## lm(formula = cirrhosis ~ liquor, data = liver)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.577 -11.127  -0.821  11.179  50.878
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.9649     7.1847   3.057  0.00379 **
## liquor        0.7222     0.1168   6.185  1.8e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.34 on 44 degrees of freedom
## Multiple R-squared:  0.4651, Adjusted R-squared:  0.4529
## F-statistic: 38.26 on 1 and 44 DF,  p-value: 1.803e-07
```

8. Plot the residuals of your least squares fit.

```
ggplot(liquor_fit)+
  geom_point(aes(x=.fitted, y=.resid))+
  ylab('Residual')+
  xlab('liquor')+
  ggtitle("Residual Plot of Linear Regression Model")+
  geom_hline(yintercept=0, color='red')
```



9. Give an exact 95% confidence interval for β assuming that the noise terms are i.i.d. normal. Compare it with a 95% asymptotic confidence interval that does not assume that the errors are normal. Discuss briefly their relative merits.

```
alpha = 0.05
n=46
t=qt(1-alpha/2, n-2)
se = 0.1168 #according to output in question 5
LB=0.7222-t*se
UB=0.7222+t*se
c(LB, UB)
```

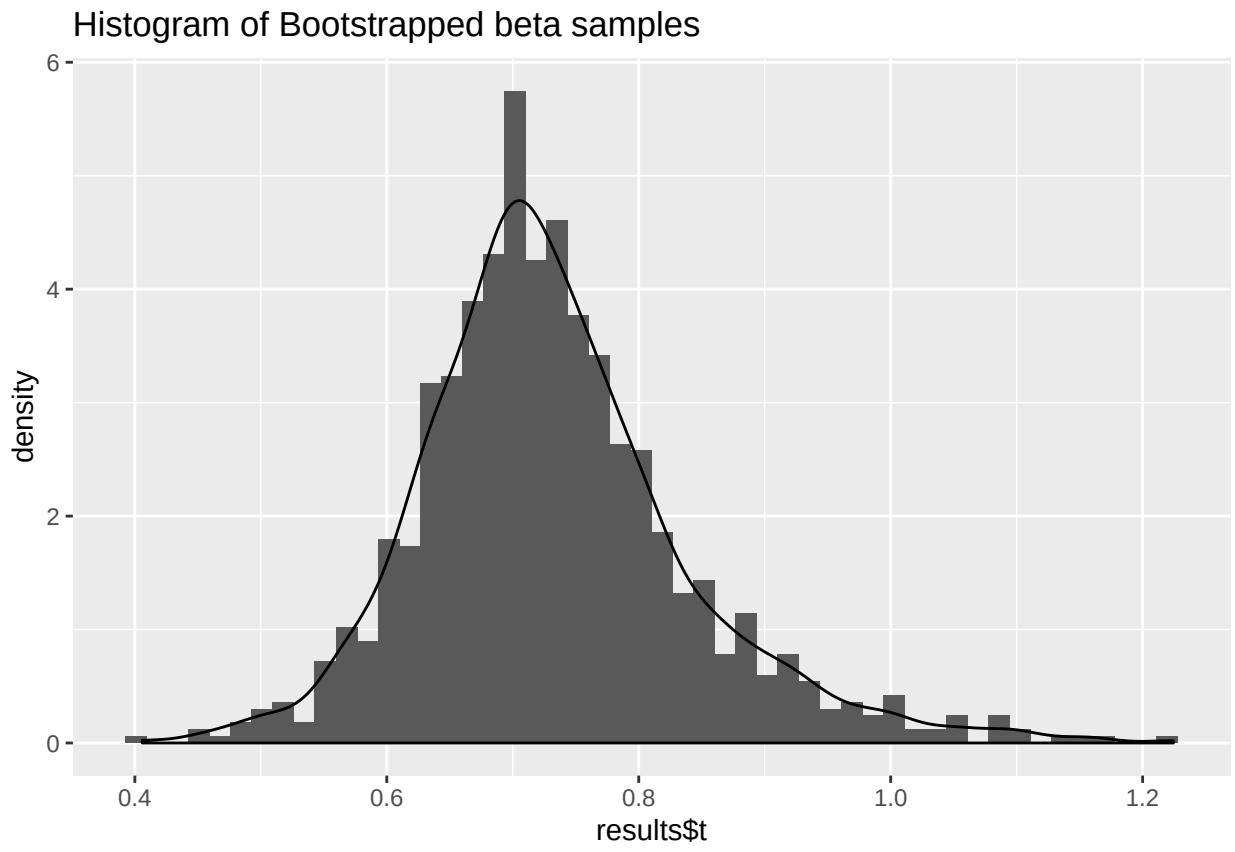
```
## [1] 0.4868051 0.9575949
```

```
n=nrow(liver)
sighat=sqrt(sum(resid(liquor_fit)^2)/n)
Sxx=sum((liver$liquor-mean(liver$liquor))^2)
interval = qnorm(0.975)*sighat/sqrt(Sxx)
betahat = as.numeric(coef(liquor_fit)['liquor'])
LB2= betahat-interval
UB2= betahat+interval
c(LB2,UB2)
```

```
## [1] 0.4984011 0.9460696
```

10. Generate 1000 bootstrap samples and use them to compute a 95% bootstrap confidence interval for β. Plot the bootstrap distribution that you obtained and compare your bootstrap confidence interval with the two obtained in point 9.

```
library(boot)
set.seed(5703)
bootstrap <- function(liver, indices) {
  d <- liver[indices,] # allows boot to select sample
  fit<-lm(cirrhosis~liquor, data=d)
  return(coef(fit)['liquor'])
}
results <- boot(data=liver, statistic=bootstrap, R=1000)
ggplot()+
  geom_histogram(aes(x=results$t, y=..density..), bins=50)+
  geom_density(aes(x=results$t, y=..density..))+
  ggtitle('Histogram of Bootstrapped beta samples')
```




```
boot.ci(results, type="bca")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "bca")
##
```

```
## Intervals :
## Level       BCa
## 95%   ( 0.5667,  1.0430 )
## Calculations and Intervals on Original Scale
```

11.

```
p1 =cor(liver$cirrhosis,liver$liquor)
p1
```

```
## [1] 0.6819694
```

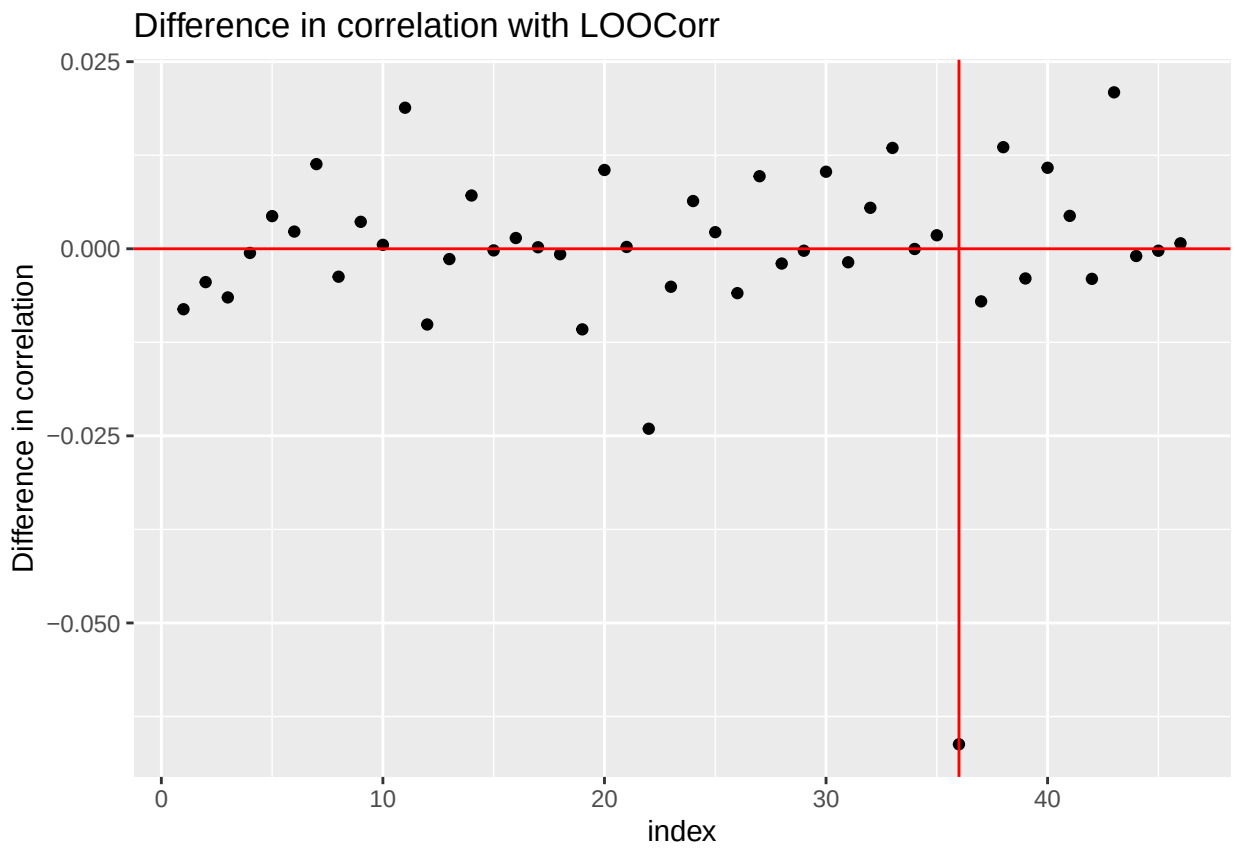
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
LOOCorr <- function(i){
  dftmp <- liver[-c(i),]
  cor(dftmp$cirrhosis, dftmp$liquor) - cor(liver$cirrhosis,liver$liquor)
}
corr_diff <- unlist(Map(LOOCorr, 1:nrow(liver)))
livercopy<-liver
livercopy$corr_diff<-corr_diff
maxindex=which.max(abs(livercopy$corr_diff))
ggplot(livercopy)+
  geom_point(aes(x=seq.int(1,nrow(livercopy)), y=corr_diff))+
  xlab('index')+
  ylab('Difference in correlation')+
  ggtitle('Difference in correlation with LOOCorr')+
  geom_hline(yintercept=0, color='red')+
  geom_vline(xintercept=maxindex, color='red')
```




```
livercopy[maxindex,]
```

```
##    liquor cirrhosis   corr_diff
## 36    149     122.5 -0.06621551
```

Exercise 5

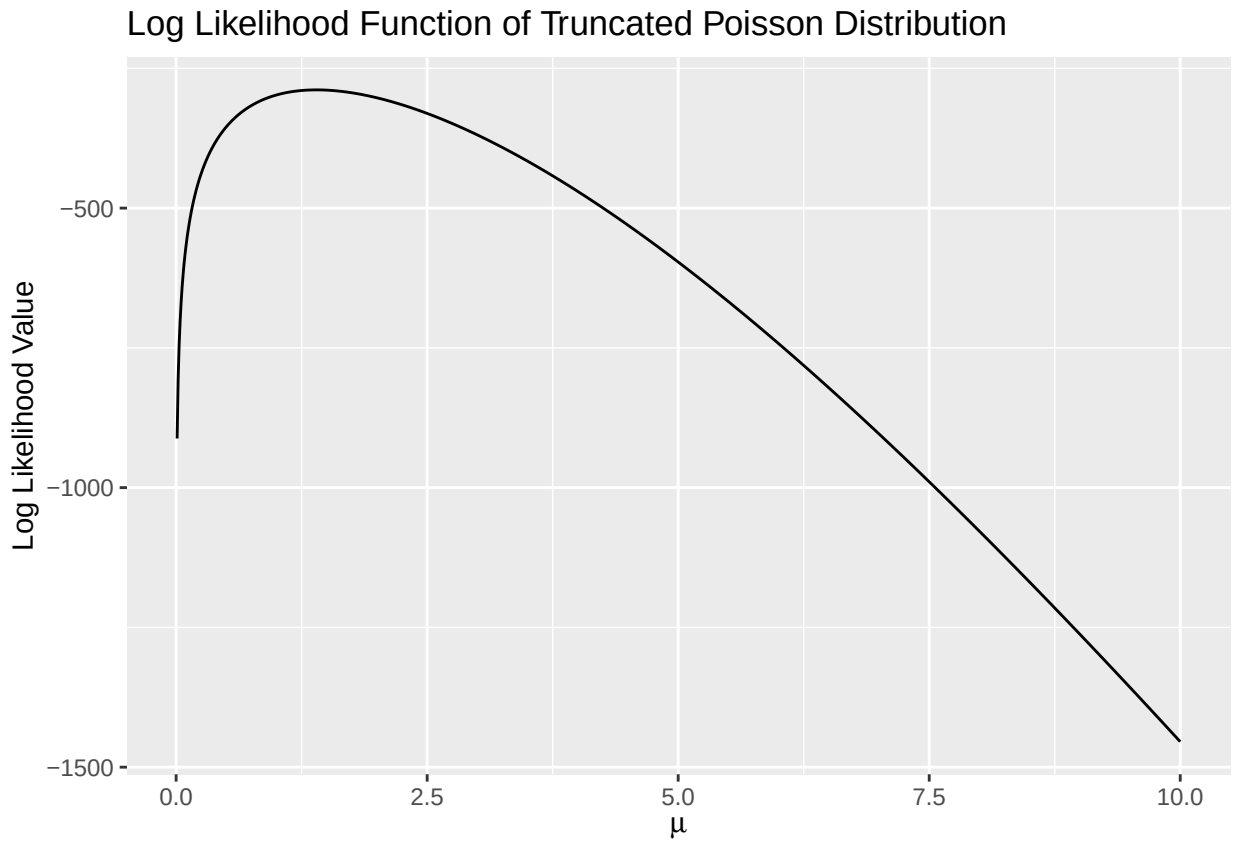
Part 3

Part 4

```
obs_freq <- c(179,51,17,6,8,1,0,2)
k <- seq(2,9,1)
table1_data <- data.frame('k'=k, 'obs_freq'=obs_freq)
obs_data <- numeric()
for (i in 1:8){
  obs_data <- c(obs_data, rep(table1_data[i,1],table1_data[i,2]))
}
```

```
log_likelihood <- function(data, mu){
  n <- length(data)
  log_llh <- -n*log(1-exp(-mu)-mu*exp(-mu))-mu*n+sum(data*log(mu))-sum(log(factorial(data)))
}
```

```r
list_mu <- seq(0.01,10,0.01)
list_log_llh <- numeric()
for (i in 1:length(list_mu)){
  temp <- log_likelihood(obs_data, list_mu[i])
  list_log_llh <- c(list_log_llh,temp)
}
df <- data.frame('mu'=list_mu, 'llh'=list_log_llh)
ggplot(df, aes(x=mu,y=llh))+
  geom_line()+
  ggtitle('Log Likelihood Function of Truncated Poisson Distribution')+
  labs(x=expression(mu), y='Log Likelihood Value')
```



Part 5

```r
neg_log_likelihood <- function(mu){
  result <- -log_likelihood(data=obs_data, mu)
  return(result)
}
mle(neg_log_likelihood, start = list(mu=1), method = "BFGS")
```

```
##
## Call:
## mle(minuslogl = neg_log_likelihood, start = list(mu = 1), method = "BFGS")
```

```
##
## Coefficients:
##       mu
## 1.398391
```

Part 6

```
mu_mle <- 1.398391
fisher_info <- ((1-exp(-mu_mle))^2-mu_mle^2 * exp(-mu_mle))/
  (mu_mle * (1-exp(-mu_mle)-mu_mle*exp(-mu_mle))^2)
fisher_info
```

```
## [1] 0.3616344
```

Part 7

```
z <- 1.96
n <- 264
lower <- mu_mle-z/sqrt(n*fisher_info)
upper <- mu_mle+z/sqrt(n*fisher_info)
c(lower, upper)
```

```
## [1] 1.197796 1.598986
```