Chutian Chen:cc4515

Congcheng Yan: cy2550

Mingrui Liu: ml4404

Exercise 2

```
import numpy as np
from scipy.misc import derivative
from scipy.stats import norm
from sympy import diff, symbols
import random

random.seed(1111)
model_1 = np.random.poisson(5,500*50)
m1,v1 = np.mean(model_1), np.var(model_1,ddof = 1)
def model2_gen(n):
    a = []
    for i in range(n):
        a = a + list(np.random.poisson(np.random.gamma(2.5,2,1),50))
    return np.array(a)
model_2 = model2_gen(500)
m2,v2 = np.mean(model_2), np.var(model_2,ddof = 1)
print("Model A: ", np.sqrt(500/2)*(v1-m1)/m1)
print("Model B: ",np.sqrt(500/2)*(v2-m2)/m2)
norm.ppf(0.975)

f = [1,4,15,31,39,55,54,49,47,31,16,9,8,4,3]
data = []
for i,fre in enumerate(f):
    for j in range(fre):
        data.append(i)
data = np.array(data)
m3,v3 = np.mean(data), np.var(data,ddof = 1)
np.sqrt(len(data)/2)*(v3-m3)/m3
```

Exercise 4

```
library(ggplot2)
data <- read.csv('liver.csv')
ggplot(data) + geom_point(aes(x = liquor, y = cirrhosis))
```

```r
model <- lm(cirrhosis~liquor, data = data)
summary(model)

predict(model,data.frame(liquor=180))

res <- resid(model)
ggplot() + geom_point(aes(x = data$liquor, y = res))

residual <- sum(res^2)
sxx <- sum(data$liquor^2) - nrow(data)*(mean(data$liquor)^2)
interval <- qt(0.975,nrow(data)-2) * sqrt(residual/(nrow(data)-2)/sxx)
l <- as.numeric(coef(model)['liquor'])-interval
r <- as.numeric(coef(model)['liquor'])+interval
print(c(l,r))

s_hat <- sd(res)*(nrow(data)-1)/(nrow(data)-2)
interval2 <- qnorm(0.975)*s_hat/sqrt(sxx)
l2 <- as.numeric(coef(model)['liquor'])-interval2
r2 <- as.numeric(coef(model)['liquor'])+interval2
print(c(l2,r2))

# Bootstrap 95% CI
library(boot)

bootstrap <- function(data, indices) {
  d <- data[indices,] # allows boot to select sample
  fit <- lm(cirrhosis~liquor, data=d)
  return(coef(fit)['liquor'])
}
# bootstrapping with 1000 replications
set.seed(31415)
results <- boot(data=data, statistic=bootstrap, R=1000)

# view results
ggplot() + geom_histogram(aes(x = results$t), bins = 50)
# get 95% confidence interval
boot.ci(results, type="bca")

library(dplyr)
try <- function(i) {
  if (i == nrow(data)) data_loo <- data[0:(i-1),]
  else data_loo <- data[c(0:(i-1),(i+1):(nrow(data))),]
  return(corr(data_loo)-corr(data))
}
corr_d <- unlist(Map(try,1:nrow(data)))
```

```
data$corr_d <- corr_d
plot(corr_d)
data[order(corr_d),][1,]
```

Exercise 5

```
library(ggplot2)
freq <- c(179,51,17,6,8,1,0,2)
y <- seq(2,9,1)
data <- as.data.frame(y)
data$freq <- freq
log_mle <- function(miu) {
  likehood <- function(miu,y) {
    return(-miu+y*log(miu)-log(factorial(y))-log(1-exp(-miu)-miu*exp(-miu)))
  }
  return(sum(likehood(miu,data$y)*data$freq))
}
miu <- seq(0.8,1.6,0.1)
mle <- sapply(miu,log_mle)
ggplot() + geom_line(aes(x = miu, y = mle)) + xlab("mu")

library(optim.functions)
initial <- c(1)
optim(initial, a <- function(x){return(-log_mle(x))}, method = "BFGS")

m <- 1.398
i <- ((exp(m)-1)^2-m^2*exp(m))/m/(exp(m)-1-m)^2
interval <- qnorm(0.975)/sqrt(sum(data$freq)*i)
print(c(m-interval,m+interval))
```