Module 5 Homework

1. (20 points)
A random sample of size 6 from the exp(λ) distribution results in observations:
1.433, 0.524, 0.384, 4.515, 1.852, 0.429. Find the MLE on this data set in two
ways:

a) by numerical optimization of the likelihood (please include R code) and

Answer)

> **lik<-function(lam) prod(dexp(c(1.433, 0.524, 0.384, 4.515, 1.852, 0.429),
> rate=lam))**
>
> **nlik<-function(lam) -lik(lam)**
>
> **optim(par=1, nlik)$par**

Output)

```
[1] 0.6566895
```

b) by the analytic formula.

Answer)

> **n<- 6**
>
> **Observations<- sum(1.433, 0.524, 0.384, 4.515, 1.852, 0.429)**
>
> **analytic <- n/Observations**
>
> **analytic**

Output)

```
[1] 0.6566707
```

2. (15 points)

A random sample $X_1, X_2, ..., X_{75}$ follows chi-square distribution with **m** degree of freedom, has sample mean $\bar{X} = 98.6$ and sample standard deviation $s = 9.4$.

(a) Find the point estimator of **m** using the method of moments.

2) (a) is (very) easy.

Answer)

a)

    #In chi-square Distribution the population mean(m) is equal to sample mean(X)

    m=98.6

(b) Find a one-sided 90% lower confidence interval of **m**.

Answer)

    CI<- 98.6-qt(0.10, 74)*(9.4/sqrt(74))
    CI

Output)

    [1] 100.013

Please provide the formulas and the derivations together with your numerical answer.

Problem 3 (35 points)

On the Golub et al. (1999) data set, analyze the Zyxin gene expression data separately for the ALL and AML groups.

(a) Find the bootstrap 95% CIs for the mean and for the variance of the gene expression in each group separately.

Answer)

```
#mean and variance for ALL
data(golub, package = "multtest")

ZyxinALL<-golub[2124,1:27]
n<-length(ZyxinALL)
nboot<-1000
boot.xbarmALL<-rep(NA,nboot)
boot.xbarvALL<-rep(NA,nboot)
for(i in 1:nboot) {
  data.star<-ZyxinALL[sample(1:n,replace=TRUE)]
  boot.xbarmALL[i]<-mean(data.star)
  boot.xbarvALL[i]<-var(data.star)
}
quantile(boot.xbarmALL,c(0.025,0.975))
quantile(boot.xbarvALL,c(0.025,0.975))
```

Output)

```
> quantile(boot.xbarmALL,c(0.025,0.975))
       2.5%        97.5%
-0.56294390 -0.04944982
> quantile(boot.xbarvALL,c(0.025,0.975))
      2.5%      97.5%
0.3571430 0.6535835
```

#mean and variance for AML

```
ZyxinAML<-golub[2124,28:38]
n<-length(ZyxinAML)
nboot<-1000
boot.xbarmAML<-rep(NA,nboot)
boot.xbarvAML<-rep(NA,nboot)
for(i in 1:nboot) {
  data.star<-ZyxinAML[sample(1:n,replace=TRUE)]
  boot.xbarmAML[i]<-mean(data.star)
  boot.xbarvAML[i]<-var(data.star)
  }
quantile(boot.xbarmAML,c(0.025,0.975))
quantile(boot.xbarvAML,c(0.025,0.975))
```

Output)

```
> quantile(boot.xbarmAML,c(0.025,0.975))
     2.5%     97.5%
1.384754 1.797642
> quantile(boot.xbarvAML,c(0.025,0.975))
      2.5%        97.5%
0.05193054 0.20493144
```

(b) Find the parametric 95% CIs for the mean and for the variance of the gene expression in each group separately. (You need to choose the appropriate approximate formula to use: z-interval, t-interval or chi-square interval.)

Answer)

parametric mean for ALL and AML

```
gol.fac <- factor(golub.cl,levels=0:1,labels=c("ALL","AML"))
xmALL <- golub[2124, gol.fac=="ALL"]
nmALL<-length(xmALL)
ci.mALL <- mean(xmALL)+qt(c(0.025,0.957),df=nmALL-
1)*sd(xmALL)/sqrt(nmALL)

gol.fac <- factor(golub.cl,levels=0:1,labels=c("ALL","AML"))
xmAML <- golub[2124, gol.fac=="AML"]
nmAML<-length(xMAML)
ci.mAML <- mean(xmAML)+qt(c(0.025,0.957),df=nmAML-
1)*sd(xmAML)/sqrt(nmAML)
print(ci.mALL)
print(ci.mAML)
```

Output)

```
> print(ci.mALL)
[1] -0.58073875 -0.04653493
> print(ci.mAML)
[1] 1.339698 1.797752
```

Answer)

gol.fac <- factor(golub.cl,levels=0:1,labels=c("ALL","AML"))
xvALL <- golub[2124, gol.fac=="ALL"]
nvALL<-length(xvALL) #sample size n
ci.vALL <- var(xvALL)+qt(c(0.025,0.957),df=nvALL-1)*sd(xvALL)/sqrt(nvALL)

gol.fac <- factor(golub.cl,levels=0:1,labels=c("ALL","AML"))
xvAML <- golub[2124, gol.fac=="AML"]
nvAML<-length(xvAML) #sample size n
ci.vAML <- var(xvAML)+qt(c(0.025,0.957),df=nvAML-1)*sd(x)/sqrt(nvAML)
print(ci.vALL)
print(ci.vAML)

Output)

```
> print(ci.vALL)
[1] 0.2365522 0.7707560
> print(ci.vAML)
[1] -0.1118260  0.3462277
```

(c) Find the bootstrap 95% CI for the <u>median</u> gene expression in both groups separately.

Answer)

#median for ALL

```
ZyxinmdALL<-golub[2124,1:27]
n<-length(ZyxinmdALL)
nboot<-1000
boot.xbarmdALL <- rep(NA, nboot)
for (i in 1:nboot) {
  data.star <- ZyxinmdALL[sample(1:n,replace=TRUE)]
  boot.xbarmdALL[i]<-median(data.star)
}
quantile(boot.xbarmdALL,c(0.025,0.975))
```

Output)

```
> quantile(boot.xbarmdALL,c(0.025,0.975))
    2.5%     97.5%
-0.73507   0.31432
```

Answer)

 # median for AML

```
ZyxinmdAML<-golub[2124,28:38]
n<-length(ZyxinmdAML)
nboot<-1000
boot.xbarmdAML <- rep(NA, nboot)
for (i in 1:nboot) {
  data.star <- ZyxinmdAML[sample(1:n,replace=TRUE)]
  boot.xbarmdAML[i]<-median(data.star)
```

```
        }
        quantile(boot.xbarmdAML,c(0.025,0.975))
```

Output)

```
> quantile(boot.xbarmdAML,c(0.025,0.975))
    2.5%    97.5%
1.22814 1.82829
```

Please provide numerical answers for each part. Please also submit your R codes used for the calculations (**the R code should be clearly labeled and separated for each part**).

4. (30 points)

For a random sample of 50 observations from Poisson distribution, we have two ways to construct a 90% CI for the parameter λ.

(1) Since the Poisson mean is λ, we can use the interval for the sample mean ($\bar{X} + t_{0.05,49}\sqrt{\frac{\bar{X}}{50}}, \bar{X} + t_{0.95,49}\sqrt{\frac{\bar{X}}{50}}$).

(2) Since the Poisson variance is also λ, we can use the interval for the sample variance directly: ($\frac{49s^2}{\chi^2_{0.95,49}}, \frac{49s^2}{\chi^2_{0.05,49}}$).

(a) Write a R-script to conduct a Monte Carlo study for the coverage probabilities of the two CIs. That is, to generate nsim=1000 such data sets from the Poisson distribution. Check the proportion of the CIs that contains the true parameter λ.

Answer)

```
nsim <- 1000
MCsim<- function(nsim, lambda) {
  cov1<-cov2<-rep(NA,nsim) # create empty matrices to store data
  for (i in 1:nsim) {
    x<- rpois(50, lambda) # The question says Poisson
    xbar<- mean(x) #find mean of x
```

```
    Xsd<- sd(x)# find sd of x
    CI1<-c(xbar+(qt(0.05, 49)*sqrt(xbar/50)), xbar+qt(0.95,
49)*sqrt(xbar/50)) #use the formula for conf interval for mean given in
the question
    CI2<-c(49*(Xsd^2)/qchisq(0.95, 49), 49*(Xsd^2)/qchisq(0.05, 49))
    cov1[i]<-(CI1[1]<lambda)&(lambda<CI1[2])
    cov2[i]<-(CI2[1]<lambda)&(lambda<CI2[2])
  }

  print(paste("When lambda=", lambda, ": coverage for first CI is",
mean(cov1), ", coverage for second CI is", mean(cov2), ".")) # Just to
keep your output presentable
}
```

(b) Run the Monte Carlo simulation for nsim=1000 runs, at three different
parameter values: $\lambda=0.1$, $\lambda=1$ and $\lambda=10$. Report the coverage probabilities of
these two CIs at each of the three parameter values.

Answer)

    MCsim(1000, 0.1)
    MCsim(1000, 1)
    MCsim(1000, 10)

Output)

```
> MCsim(1000, 0.1)
[1] "When lambda= 0.1 : coverage for first CI is 0.867 , c
overage for second CI is 0.552 ."
> MCsim(1000, 1)
[1] "When lambda= 1 : coverage for first CI is 0.908 , cov
erage for second CI is 0.849 ."
> MCsim(1000, 10)
[1] "When lambda= 10 : coverage for first CI is 0.906 , co
verage for second CI is 0.899 ."
```