

1. (20 points)

(a) Find the analytic MLE formula for exponential distribution $\exp(\lambda)$. Show that MLE is the same as MoM estimator here.

(b) A random sample of size 6 from the $\exp(\lambda)$ distribution results in observations:

1.636, 0.374, 0.534, 3.015, 0.932, 0.179. Find the MLE on this data set in two ways: by numerical optimization of the likelihood and by the analytic formula.

For (b): please give both values from the analytic MLE formula and numerical MLE solution on this data set. Also, please submit the R code for numerically finding the MLE.

A) Likelihood function for exp

$$lik(\lambda) = (\lambda e^{-\lambda x_1}) \dots (\lambda e^{-\lambda x_n}) = \lambda^n e^{-\lambda \sum_{i=1}^n x_i}$$

Log Likelihood function

$$\text{Log lik}(\lambda) = n \log \lambda - \lambda \sum_{i=1}^n x_i$$

Setting the derivative to zero

$$\frac{n}{\lambda} - \sum_{i=1}^n x_i = 0$$

$$\text{MLE} = \hat{\lambda}_n = \frac{n}{\sum_{i=1}^n x_i} = \frac{1}{\bar{X}}$$

MOM

Let X_1, \dots, X_n be a random sample from the exponential distribution with parameter λ . We estimate λ by matching the first moments of the population and those of the sample. The population mean is $\frac{1}{\lambda}$, the sample mean is \bar{X} . So the MoM estimator is $\hat{\lambda} = \frac{1}{\bar{X}}$

$$\text{B) Analytic MLE} - \hat{\lambda} = \frac{1}{\bar{X}} = \frac{6}{(1.636+0.374+0.534+3.015+0.932+0.179)} = .8995502$$

Numerical MLE –

Rscript:

```
lik<-function(lam) prod(dexp(c(1.636,0.374,0.534,3.015,0.932,0.179),
rate=lam))
nlik<-function(lam) -lik(lam)
optim(par=0.05, nlik)$par
```

Answer:

.8995502

2. (15 points)

A random sample of X_1, \dots, X_{53} , from the chi-square distribution with \mathbf{m} degree of freedom, has sample mean $\bar{X} = 100.8$ and sample standard deviation $s = 12.4$

(a) Find the point estimator of \mathbf{m} using the method of moments.

(b) Find a one-sided 90% lower confidence interval of \mathbf{m} .

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

A) The population mean is \mathbf{m} , the sample mean is \bar{X} .

So the MoM estimator is $\hat{\mathbf{m}} = \bar{X}$.

So $\hat{\mathbf{m}} = 100.8$

B) One sided lower confidence interval = $(\bar{X} + t_{\alpha, n-1} \frac{s}{\sqrt{n}}, \infty)$

$\alpha = 1 - \text{CI} = 1 - .9 = .1$

$(100.8 + t_{.1, 52} \frac{12.4}{\sqrt{52}}, \infty)$

Rscript:

`100.8 + qt(c(0.1), df=52)*(12.4/sqrt(52))`

one-sided 90% lower confidence interval of $\mathbf{m} = (98.56792, \infty)$

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.

(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.)

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

(d) Considering the CIs in parts (a)-(c), does the Zyxin gene express differently in ALL and AML patients?

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).

Rscript:

A)

```
data(golub, package = "multtest")
gol.fac <- factor(golub.cl, levels=0:1, labels=c("ALL", "AML"))
ZyxinALL <- golub[2124, gol.fac=="ALL"]
nALL <- length(ZyxinALL)
ZyxinAML <- golub[2124, gol.fac=="AML"]
nAML <- length(ZyxinAML)
nboot <- 1000
boot.xbar.ALL <- rep(NA, nboot)
boot.var.ALL <- rep(NA, nboot)
boot.xbar.AML <- rep(NA, nboot)
boot.var.AML <- rep(NA, nboot)
for (i in 1:nboot) {
  data.star.ALL <- ZyxinALL[sample(1:nALL, replace=TRUE)]
  boot.xbar.ALL[i] <- mean(data.star.ALL)
  boot.var.ALL[i] <- var(data.star.ALL)
  data.star.AML <- ZyxinAML[sample(1:nAML, replace=TRUE)]
  boot.xbar.AML[i] <- mean(data.star.AML)
  boot.var.AML[i] <- var(data.star.AML)
}
```

```
quantile(boot.xbar.ALL,c(0.025,0.975))
quantile(boot.var.ALL, c(0.025,0.975))
quantile(boot.xbar.AML,c(0.025,0.975))
quantile(boot.var.AML, c(0.025,0.975))
```

Answer:

Mean 95% CI for ALL patients:
-0.56657678 -0.02116587

Variance 95% CI for ALL patients:
0.3452325 0.6551290

Mean 95% CI for AML patients:
1.390031 1.793037

Variance 95% CI for AML patients:
0.04729564 0.20544064

B) Rscript:

```
mean(ZyxinALL)+qt(c(0.025,0.975),df=nALL-
1)*sd(ZyxinALL)/sqrt(nALL)
mean(ZyxinAML)+qt(c(0.025,0.975),df=nAML-
1)*sd(ZyxinAML)/sqrt(nAML)
(nALL-1)*((sd(ZyxinALL))^2)/qchisq(c(0.975,0.025),df=nALL-1)
(nAML-1)*((sd(ZyxinAML))^2)/qchisq(c(0.975,0.025),df=nAML-1)
```

Answers:

Mean 95% CI for ALL patients:
-0.580738750 -0.008846435

Variance 95% CI for ALL patients:
0.3240441 0.9812951

Mean 95% CI for AML patients:
1.339698 1.833638

Variance 95% CI for AML patients:
0.06597815 0.41621602

C) Rscript:

```
data(golub, package = "multtest")
gol.fac <- factor(golub.cl,levels=0:1,labels=c("ALL","AML"))
ZyxinALL<-golub[2124, gol.fac=="ALL"]
```

```

nALL<-length(ZyxinALL)
ZyxinAML<-golub[2124, gol.fac=="AML"]
nAML<-length(ZyxinAML)
nboot<-1000
boot.xmed.ALL <- rep(NA, nboot)
boot.xmed.AML <- rep(NA, nboot)
for (i in 1:nboot) {
  data.star.ALL <- ZyxinALL[sample(1:nALL,replace=TRUE)]
  boot.xmed.ALL[i]<-median(data.star.ALL)
  data.star.AML <- ZyxinAML[sample(1:nAML,replace=TRUE)]
  boot.xmed.AML[i]<-median(data.star.AML)
}
quantile(boot.xmed.ALL,c(0.025,0.975))
quantile(boot.xmed.AML,c(0.025,0.975))

```

Answers:

Median 95% CI for ALL patients:

-0.73507 0.31432

Median 95% CI for AML patients:

1.22814 1.82829

d)

Considering the CIs in parts (a)-(c), does the Zyxin gene express differently in ALL and AML patients?

Yes gene express differently in ALL and AML patients as seen from the different mean values of gene expression obtained from bootstrap and parametric CI.

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: $(\frac{49\bar{x}^2}{\chi_{0.95,49}^2}, \frac{49\bar{x}^2}{\chi_{0.05,49}^2})$

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

(b) Run the Monte Carlo simulation for $\text{nsim}=1000$ 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.

(c) Considering your result in part (b), which one of these two CI formulas should you use in practice? Can you explain the pattern observed in (b)?

Rscript: A)

```
n<-50
nsim<-1000
xmean<-matrix(NA, nrow=nsim, ncol=2)
xvar <- matrix(NA, nrow=nsim, ncol=2)
for (i in 1:nsim){
  x<- rpois(n,lambda=0.1)
  xmean[i,<- mean(x) +qt(c(.05,.95),df =n-1) * sqrt(mean(x)/n)
  xvar[i,<- 49*(sd(x)^2)/qchisq(c(.95,.05),df=n-1)
}
mean((xmean[,1]<.1)&(.1<xmean[,2]))
mean((xvar[,1]<.1)&(.1<xvar[,2]))
```

B)

Answers:

$\lambda=0.1$:

Coverage probability of mean CI – 0.85

Coverage probability of Variance CI – 0.556

$\lambda=1$

Coverage probability of mean CI – 0.908

Coverage probability of Variance CI – 0.835

$\lambda=10$

Coverage probability of mean CI – 0.924

Coverage probability of Variance CI – 0.907

C)

I think Mean CI would be better if we are considering smaller values of lambda and for higher values of lambda we can use any of the two CI as both become almost similar.

Pattern observed here:

For mean CI there is slight change as the lambda values are increased so we can say Coverage probabilities of mean CI is more stable.

On the other hand variance CI is increasing with increasing CI so it becomes more stable as lambda value is increased.