

# Computational Stats Deliverable 2

*António Coelho, Gonçalo Arsénio, Sara Vigário, Tiago dos Santos*

*2018-11-08*

## Contents

<b>1</b>	<b>Problem 1</b>	<b>1</b>
1.1	1 a)	1
1.2	1 b)	2
1.3	1 c)	4
<b>2</b>	<b>Problem 2</b>	<b>8</b>
2.1	2 a)	8
2.2	2 b)	9
2.3	2 c)	10
2.4	2 d)	10
2.5	Discussion	13

## 1 Problem 1

Consider the following sample:

`c(7.0,3.5,11.9,8.9,10.1,1.2,1.1,7.9,12.9,1.3,5.2,5.1,3.9,2.5,10.4,6.2,-3.9)`

### 1.1 1 a)

Assume that the given sample originated from a random variable with a normal distribution whose parameter  $\sigma^2$  is unknown. Use the sample to determine the maximum-likelihood estimator of the parameter  $\sigma^2$ .

---

Maximum normal distribution Likelihood estimation.

Assuming that our sample follows a normal distribution as said on the exercise question, we can say that the probability density function follows this format

$$f_X(x_j) = (2\pi\sigma_0^2).e^{-\frac{1}{2}\cdot\frac{(x_j-\mu)^2}{\sigma_0^2}}$$

And to obtain the maximum likelihood estimation, we need to partially derivate the likelihood in order to the two parameters,  $\mu$  and  $\sigma^2$ , and discover the zeros.

$$\begin{aligned}L(\mu, \sigma^2, x_1, \dots, x_n) &= \prod_{j=1}^n f_X(x_j; \mu, \sigma^2) \\&= \prod_{j=1}^n (2\pi\sigma^2)^{-1/2} \cdot e^{-\frac{1}{2}\cdot\frac{(x_j-\mu)^2}{\sigma^2}} \\&= (2\pi\sigma^2)^{-n/2} \cdot e^{-\frac{1}{2\sigma^2} \sum_{j=1}^n (x_j-\mu)^2}\end{aligned}$$

Likelihood function:

$$l(\mu, \sigma^2; x_1, \dots, x_n) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{j=1}^n (x_j - \mu)^2$$

Derivates equals to zero

$$\frac{\partial}{\partial \mu} l(\mu, \sigma^2; x_1, \dots, x_n) = 0$$

$$\frac{\partial}{\partial \mu} l(\mu, \sigma^2; x_1, \dots, x_n) = \frac{\partial}{\partial \mu} \left( -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{j=1}^n (x_j - \mu)^2 \right) = \frac{1}{\sigma^2} \left( \sum_{j=1}^n x_j - n\mu \right)$$

$$\frac{1}{\sigma^2} \left( \sum_{j=1}^n x_j - n\mu \right) = 0 \equiv \left( \sum_{j=1}^n x_j - n\mu \right) = 0$$

Then:

$$\hat{\mu} = \frac{1}{n} \left( \sum_{j=1}^n x_j \right)$$

$$\frac{\partial}{\partial \sigma^2} l(\mu, \sigma^2; x_1, \dots, x_n) = 0$$

$$\frac{\partial}{\partial \sigma^2} l(\mu, \sigma^2; x_1, \dots, x_n) = \frac{\partial}{\partial \sigma^2} \left( -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{j=1}^n (x_j - \mu)^2 \right) = \frac{1}{2\sigma^2} \left[ \frac{1}{\sigma^2} \sum_{j=1}^n (x_j - \mu)^2 - n \right]$$

$$\frac{1}{2\sigma^2} \left[ \frac{1}{\sigma^2} \sum_{j=1}^n (x_j - \mu)^2 - n \right] = 0 \equiv \frac{1}{\sigma^2} \sum_{j=1}^n (x_j - \mu)^2 - n = 0$$

Then:

$$\widehat{\sigma^2} = \frac{1}{n} \sum_{j=1}^n (x_j - \mu)^2$$

```
sample<-c(7.0,3.5,11.9,8.9,10.1,1.2,1.1,7.9,12.9,1.3,5.2,5.1,3.9,2.5,10.4,6.2,-3.9)
```

```
norm.maximLikelihoodEst.var <- function(sample){
  n <- length(sample)
  sample.mean <- mean(sample)
  sum((sample - sample.mean)^2)/n
}
```

```
norm.maximLikelihoodEst.var(sample)
```

```
## [1] 18.84941
```

## 1.2 1 b)

Determine the Jackknife bias and variance of the estimator obtained in the item (a), and discuss the quality of the estimator.

We want to estimate the bias and variability of the estimator,  $\theta = t(F)$ . Since a good estimator not sensible to the sample is often of interest to know it faces minor perturbations in  $F$ . We can define:

$$L_t(y; F) = \lim_{\epsilon \rightarrow 0} \frac{t[(1 - \epsilon)F + \epsilon H_y] - t(F)}{\epsilon}$$

$$H_y = \begin{cases} 1, u \leq y \\ 0, u \geq y \end{cases} \quad (2.0)$$

as the influence fuction of  $t$  at  $F$ .

Using (2.0) with the empirical function we get the empirical influence function:

$$I(y) = L_t(y; \hat{F}) \quad (2.1)$$

Applying an extension of Taylor's Theorem to  $t(\hat{F})$ :

$$t(\hat{F}) \simeq t(F) + \frac{1}{n} \sum_{j=1}^n I_j \quad (2.2)$$

So:

$$\theta - \hat{\theta} = -\frac{1}{n} \sum_{j=1}^n I_j \quad (2.3)$$

If we take  $\epsilon = -\frac{1}{n-1}$  we get a distribution with no weight on the point  $x_j$  and weight  $\frac{1}{n-1}$  on the rest of the sample,  $\hat{F}_{-j}$ . In practice this is having a sample of size  $n - 1$  by omitting  $x_j$  from the original sample.

So, the jackknife aproximation to the empirical influence value  $I_j$  is:

$$I_{jack;j} = (n - 1)[t(\hat{F}) - t(\hat{F}_{-j})] = (n - 1)(\theta - \hat{\theta}_{-j})$$

Consenquently:

$$b_{jack} = -\frac{1}{n} \sum_{j=1}^n I_{jack;j} \quad (2.1)$$

$$Var_{jack} = \frac{1}{n(n-1)} \left( \sum_{j=1}^n I_{jack;j}^2 - nb_{jack}^2 \right) \quad (2.2)$$

```
sample<-c(7.0,3.5,11.9,8.9,10.1,1.2,1.1,7.9,12.9,1.3,5.2,5.1,3.9,2.5,10.4,6.2,-3.9)
n <- length(sample)
sample.mean <- mean(sample)

ljack <- function(idx, sample, n){
  (n-1)*(norm.maximLikelihoodEst.var(sample) - norm.maximLikelihoodEst.var(sample[-idx]))
}

bias.jackknife <- -mean(sapply(1:n, ljack, sample, n))
bias.jackknife

## [1] -1.178088
```

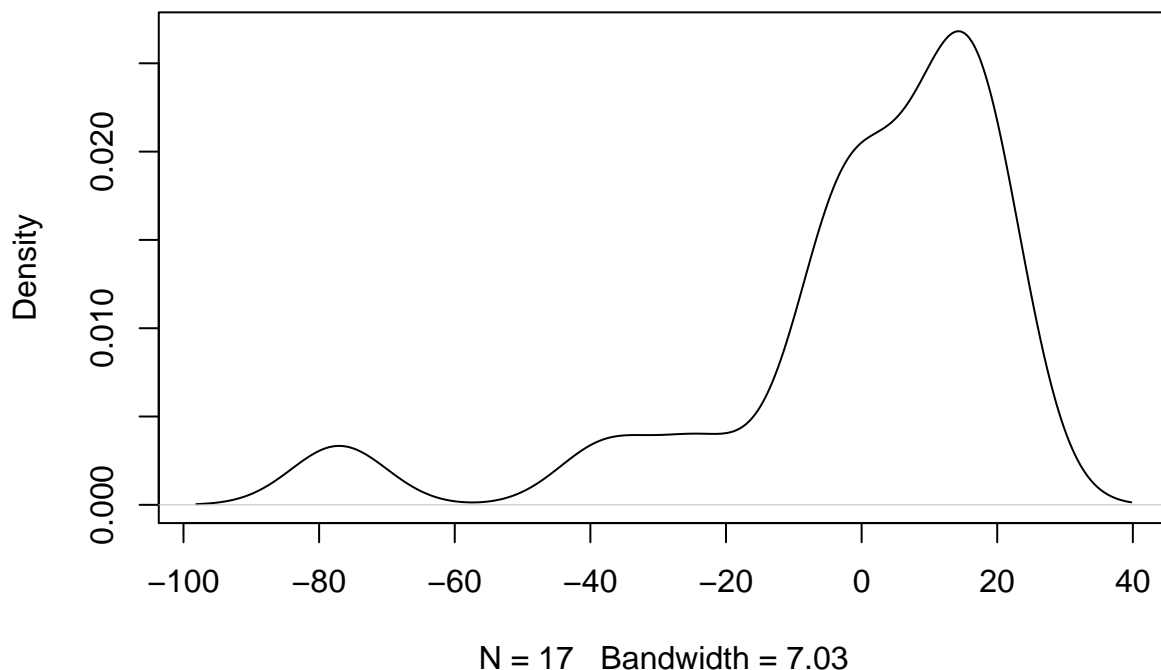
```
ljack.bias.distribution <- sapply(1:n, ljack, sample, n )

variance.jackknife <- 1/(n*(n-1))*sum(sapply(1:n,function(idx,sample,n){
  ljack(idx,sample,n)^2 - n*(bias.jackknife^2)
},sample,n))
variance.jackknife

## [1] 35.37843

plot(
  density(-ljack.bias.distribution)
  , main=paste0("Density Estimation of Jackknife Bias (mean = ",round(bias.jackknife,3)," )")
)
```

### Density Estimation of Jackknife Bias (mean = -1.178 )



### 1.3 1 c)

Consider the null hypothesis  $H_0$ : The sample was generated from a random variable with a normal distribution with parameters  $(\mu, \sigma^2) = (5, 5)$ . Use the test statistic's empirical distribution function to estimate the Kolmogorov-Smirnov test statistic's p-value. Explain why the test statistic is adequate considering the type of null hypothesis we are trying to test.

---

Hypothesis testing whose hypothesis are:

$$H_0 : X \sim \mathcal{N}(5, 5) \quad H_1 : X \not\sim \mathcal{N}(5, 5) \quad (1.0)$$

Since we don't know the probability distribution function of the test statistic,  $T(X) = D$ , we will use the empirical distribution function under the null hypothesis,  $H_0$ ,  $\hat{F}(x)_{H_0}$ ,

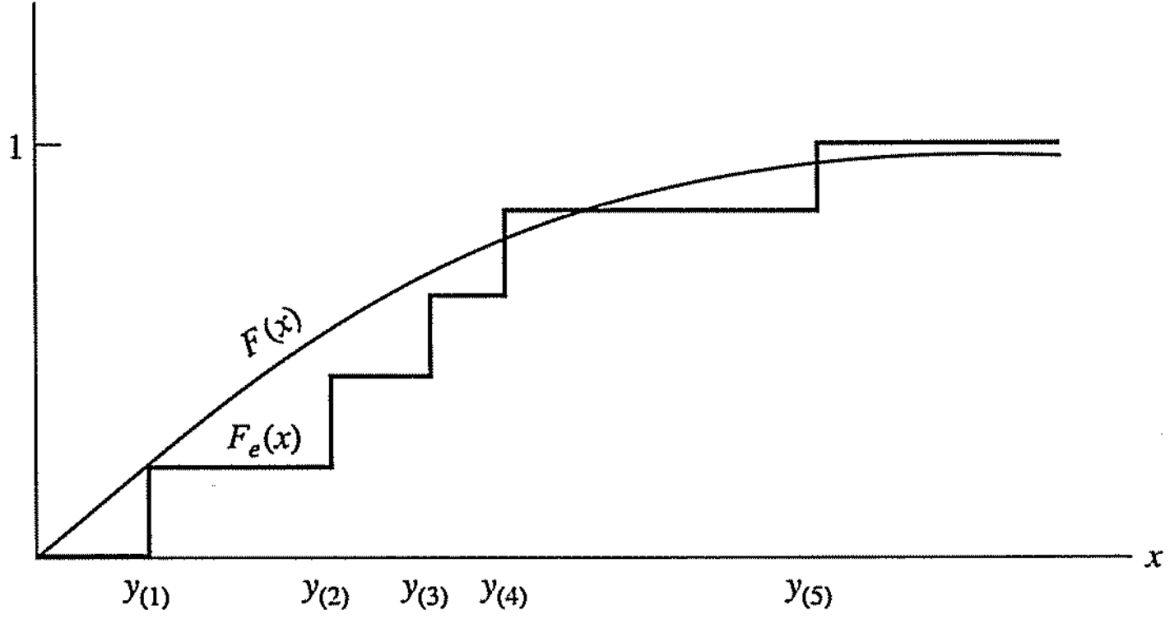


Figure 1: Graph showing the distances between continuous cdf and discrete empirical cdf (Ross, page 222, 4th edition)

$$\hat{F}(x)_{H_0} = \frac{\#\{i : x_i \leq x\}}{n} \quad (1.1)$$

with  $x_i$  being one of  $n$  observations of the sample

To do this, we need to use the Monte Carlo method.

So, we generate  $m$  samples of the random variable,  $X$  under  $H_0$ ,  $X \sim \mathcal{N}(5, 5)$ , and, for each sample, compute the test statistic:

$$x_1 : x_1^1, \dots, x_n^1 \rightarrow t_1 = T(x^1) \dots x_m : x_1^m, \dots, x_n^m \rightarrow t_m = T(x^m) \quad (1.2)$$

To compare the samples with the reference probability distribution,  $\mathcal{N}(5, 5)$ , we will use the Kolmogorov-Smirnov statistic.

In short, this statistic determines the degree of similarity between two given probability distributions through distance. This distance,  $D$ , is the the largest distance between the two given cumulative distribution functions.

Being  $F(x)$  the cumulative distribution function of the known distribution and  $y(j)$  the discontinuity points of  $\hat{F}(x)_{H_0}$ :

$$\begin{aligned} D &= \max_x |\hat{F}(x)_{H_0} - F(x)| \\ &= \max_x \{ \max_x \{ \hat{F}(x)_{H_0} - F(x) \}, \max_x \{ \hat{F}(x) - F(x)_{H_0} \} \} \\ &= \max_{j=1, \dots, n} \left\{ \frac{j}{n} - F(y(j)), F(y(j)) - \frac{j-1}{n} \right\} \end{aligned} \quad (1.3)$$

This is well illustrated in figure 1, where since both functions are monotonically increasing, a maximum distance  $D$  will only occur in the discontinuity points of  $\hat{F}(x)_{H_0}$  or  $y(j)$ .

As previously stated,  $D$  is applied to every Monte Carlo sample. So, applying (1.3) to (1.1) we get:

$$\begin{aligned} T(x^1) = D_1 &= \max_{j=1,\dots,n} \left\{ \frac{j}{n} - F(y_1(j)), F(y_1(j)) - \frac{j-1}{n} \right\} \\ &\dots \\ T(x^2) = D_2 &= \max_{j=1,\dots,n} \left\{ \frac{j}{n} - F(y_2(j)), F(y_2(j)) - \frac{j-1}{n} \right\} \end{aligned} \quad (1.4)$$

and to the original sample:

$$T(x) = d = \max_{j=1,\dots,n} \left\{ \frac{j}{n} - F(y(j)), F(y(j)) - \frac{j-1}{n} \right\} \quad (1.5)$$

Finally, the  $p - \hat{value}$  is estimated:

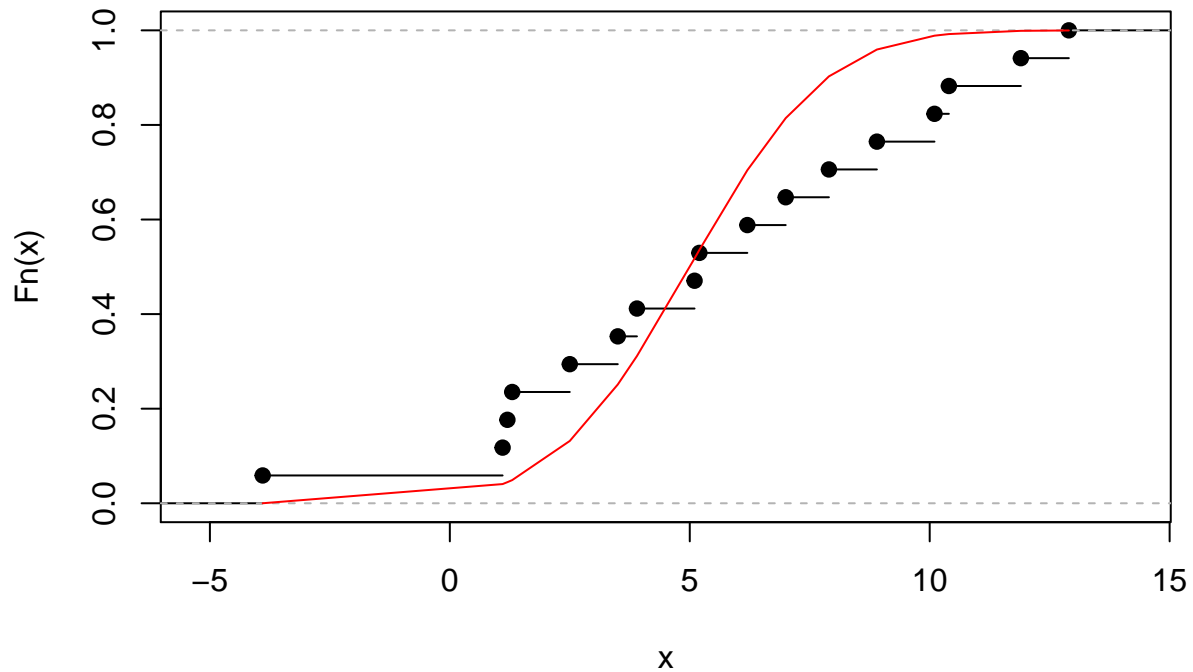
$$p - \hat{value} = P(D \geq d | H_0) = \frac{\#\{D_k : D_k \geq d\}}{n} \quad (1.6)$$

```
KS_statDist<-function(sample,mean,sd)
{
  sample<-sort(sample)
  n<-length(sample)
  DVecA<-rep(0, n)
  DVecB<-rep(0, n)
  for(j in c(1:n))
  {
    DVecA[j]<-(j/n)-pnorm(sample[j],mean,sd)
    DVecB[j]<-pnorm(sample[j],mean,sd)-((j-1)/n)
  }
  DVecAA <- sapply(1:n,function(idy){(j/n)-pnorm(sample[j],mean,sd)})
  DVecBB <- sapply(1:n,function(idy){pnorm(sample[j],mean,sd)-((j-1)/n)})
  return(max(c(DVecA,DVecB)))
}

#Kolmogorov-Smirnov test
sample.sort <- sort(sample,decreasing = F)
d<-KS_statDist(sample.sort,5,sqrt(5))

#plot
EmpCDF<-ecdf(sample.sort)
NormCDF<-pnorm(sample.sort,5,sqrt(5))
plot(EmpCDF,main="Kolmogorov-Smirnov Test")
lines(sample.sort,NormCDF,col="red")
```

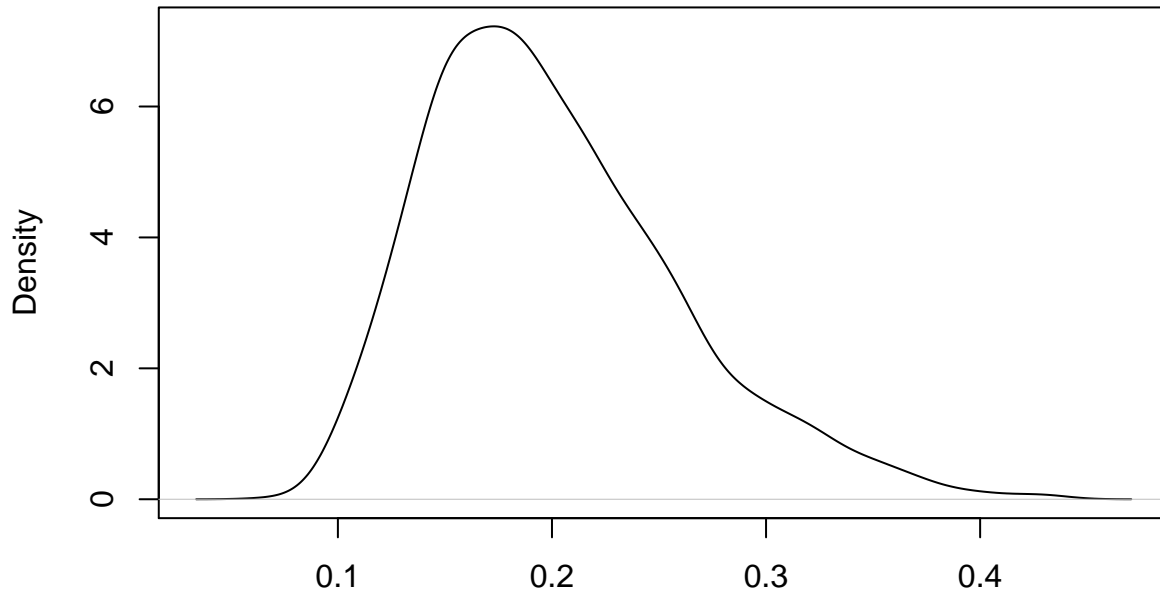
## Kolmogorov–Smirnov Test



```
MCsampleNum<-2000
m<-length(sample.sort)
DVec<-rep(0, m)
pValueNumerator<-0
for (k in c(1:MCsampleNum))
{
  #D calculation of the generated MC samples
  DVec[k]<-KS_statDist(rnorm(m,5,sqrt(5)),5,sqrt(5))
  if(DVec[k]>=d)
    pValueNumerator<-pValueNumerator+1
}

DVec.df <- data.frame(
  pts = DVec
)
library(ggplot2)
p <- ggplot2::ggplot(DVec.df, aes(x=pts)) +
  geom_density()
plot(density(DVec))
```

### density.default(x = DVec)



N = 2000 Bandwidth = 0.01166

```
#p-value calculation  
pValue<-(pValueNumerator+1)/(MCsampleNum+1)  
pValue
```

```
## [1] 0.169915
```

## 2 Problem 2

### 2.1 2 a)

Assume that the given sample originated from a random variable with a exponential distribution whose parameter  $\lambda$  is unknown. Use the sample to determine the maximum-likelihood estimator of the parameter  $\lambda$

---

Maximum normal distribution Likelihood estimation.

Assuming that our sample follows a exponential distribution as said on the exercise question, we can say that the probability density function follows this format

$$f(x; \lambda) = \lambda e^{-\lambda x}$$

And to obtain the maximum likelihood estimation, we need to derivate the likelihood function in order to  $\lambda$  and discover the zeros.

$$\begin{aligned} L(\lambda, x_1, \dots, x_n) &= \prod_{j=1}^n f_X(x_j; \lambda) \\ &= \prod_{j=1}^n \lambda e^{-\lambda x_j} \end{aligned}$$



$$= \lambda e^{-\lambda \sum_{j=1}^n x_j}$$

$$l(\lambda; x) = n \ln(\lambda) - \lambda \left( \sum_{j=1}^n x_j \right)$$

$$\frac{d}{d\lambda} l(\lambda; x) = 0 \equiv \frac{n}{\lambda} - \sum_{j=1}^n x_j = 0 \equiv \lambda = \frac{1}{\frac{\sum_{j=1}^n x_j}{n}}$$

which means

$$\hat{\lambda} = \frac{1}{\bar{X}}$$

```
sample <- c(0.05,0.03,0.19,0.14,0.12,0.03,0.08,0.19,0.07,0.01,0.24,0.10,0.03,0.31)

exp.maximLikelihoodEst.var <- function(sample){
  1/mean(sample)
}

exp.maximLikelihoodEst.var(sample)

## [1] 8.805031
```

## 2.2 2 b)

Determine the Bootstrap bias and variance of the estimator obtained in the item (a).

Viés:

$$E(\hat{\Psi} - \Psi) \approx b_{boot}(\hat{\Psi}) = \frac{1}{R} \cdot \sum_{r=1}^R (\hat{\Psi}_r^* - \hat{\Psi})$$

```
R <- 999
bias.bootstrap <- mean(sapply(1:R, function(idx,sample){
  sample.psi <- exp.maximLikelihoodEst.var(sample)
  sample.bootstrap <- rexp(length(sample),sample.psi)
  sample.bootstrap.psi <- exp.maximLikelihoodEst.var(sample.bootstrap)

  sample.bootstrap.psi - sample.psi
},sample))
bias.bootstrap

## [1] 0.6977537
```

$$Var(\hat{\Psi} - \Psi) \approx Var_{boot} = \frac{1}{R-1} \sum_{r=1}^R (\hat{\Psi}_r^* - \hat{\Psi}^*)^2$$

```
set.seed(1)

bootstrap.var <- function(sample,R){
  estimated.bootstrap <- sapply(1:R, function(idx,sample){
    exp.maximLikelihoodEst.var(rexp(length(sample),exp.maximLikelihoodEst.var(sample)))
  },sample)
```

```

    sum((estimated.bootstrap - mean(estimated.bootstrap))^2)/(R-1)
}

variance.bootstrap <- bootstrap.var(sample,R)
variance.bootstrap

```

```
## [1] 8.171176
```

## 2.3 2 c)

Construct a basic Bootstrap confidence interval for  $\lambda$

Confidence interval

$$P(a_\alpha < \hat{\theta} - \theta < a_{1-\alpha})$$

The confidence Interval can be describe like this:  $]\hat{\theta} - a_{1-\alpha}, \hat{\theta} - a_\alpha[$

But,as we do not know the distribution of  $\hat{\theta} - \theta$  we will use the empirical distribution function of  $\hat{\theta}^* - \hat{\theta}$  to estimate  $a_\alpha$  and  $a_{1-\alpha}$

That being said:  $\hat{a}_\alpha = \theta_{((R+1)\alpha)}^* - \hat{\theta}$

$$\hat{a}_{\alpha-1} = \theta_{((R+1)(\alpha-1))}^* - \hat{\theta}$$

```

alpha <- 0.025/2
set.seed(1)
distance.bootstrap <- sapply(1:R, function(idx,sample){
  sample.phi <- exp.maximLikelihoodEst.var(sample)
  sample.bootstrap <- rexp(length(sample),sample.phi)
  sample.bootstrap.phi <- exp.maximLikelihoodEst.var(sample.bootstrap)

  sample.bootstrap.phi - sample.phi
},sample)
distance.bootstrap <- c(distance.bootstrap,0)
distance.bootstrap.sorted <- sort(distance.bootstrap, decreasing = F)

ahat_left <- distance.bootstrap.sorted[(R+1)*alpha]
ahat_right <- distance.bootstrap.sorted[(R+1)*(1-alpha)]

```

```
## left value: -3.790893
## right value: 8.600578
```

## 2.4 2 d)

Construct a Studentized-Bootstrap confidence interval for  $\lambda$  (Use the Bootstrap variance estimator to estimate the Bootstrap samples's variance).

Studentized-Bootstrap confidence Interval can be describe like this:

$$]\hat{\theta} - z_{((R+1)(1-\alpha))}^* \sqrt{v}, \hat{\theta} - z_{((R+1)\alpha)}^r \sqrt{v} [$$

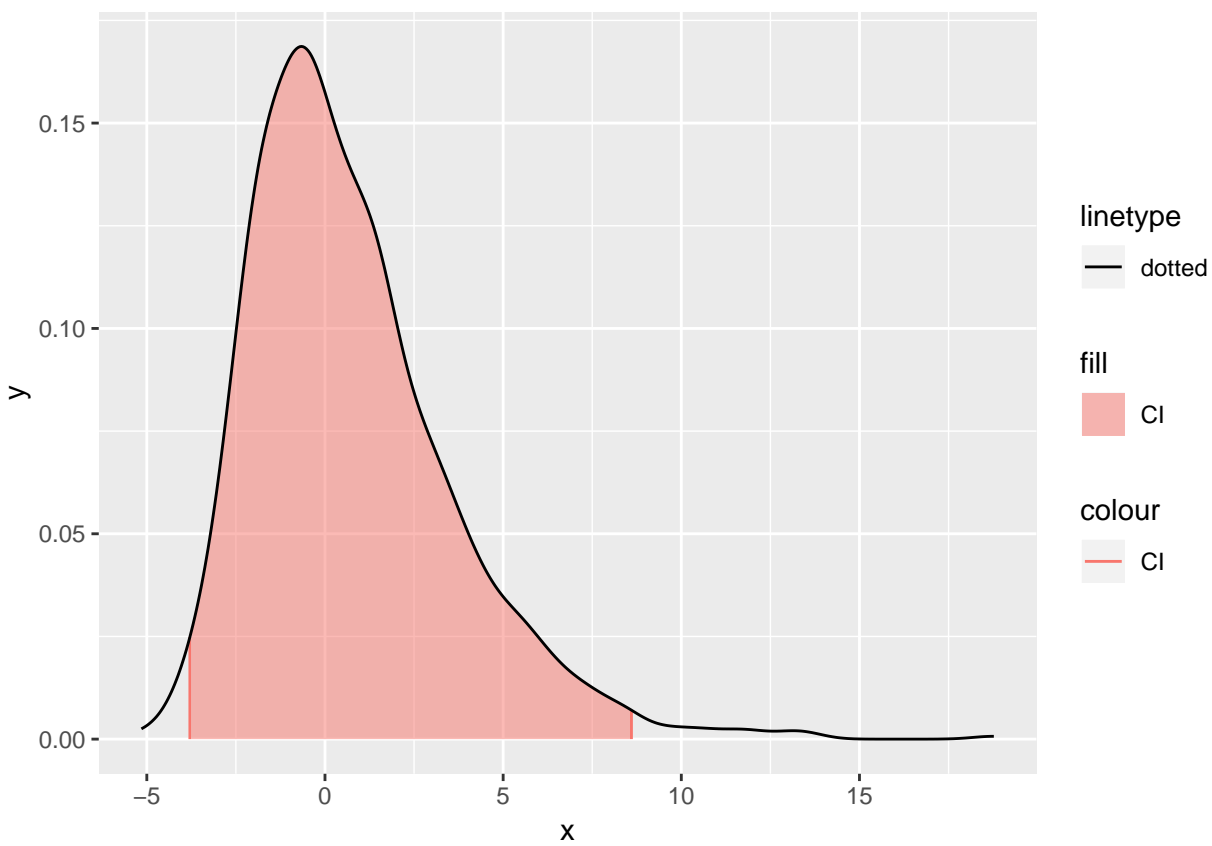


Figure 2: Density estimation of the pivot statistic  $\hat{\theta} - \theta$

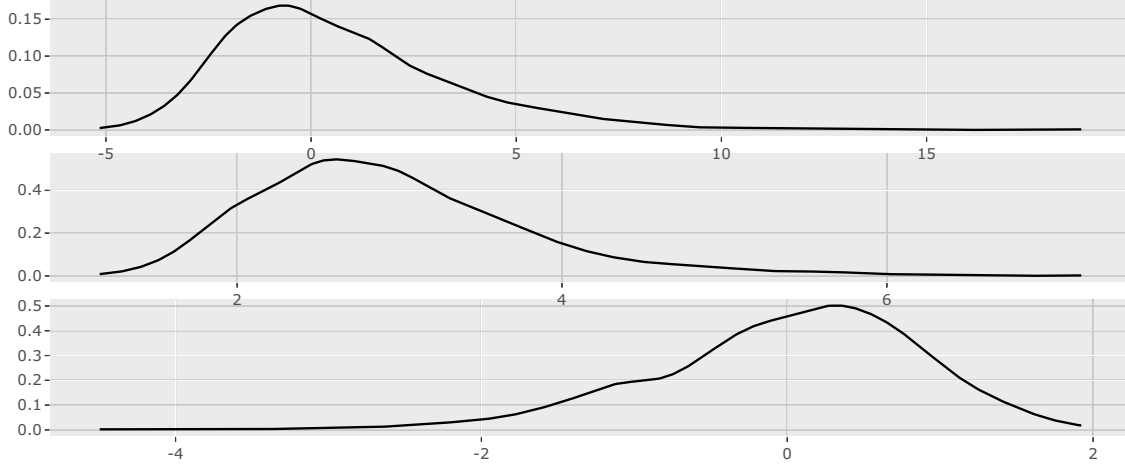


Figure 3: Density estimations of pivot stat  $\hat{\theta} - \theta$ , the bootstrap variance and the pivot stat  $z$

The pivotal statistic used is the following:

$$\hat{z} = \frac{\hat{\theta} - \theta}{\sqrt{\text{var}(\hat{\theta} - \theta)}}$$

assuming that  $\text{var}(\hat{\theta} - \theta)$  can be estimated by  $\text{Var}_{boot}(\hat{\theta})$  then

$$\hat{z}_r^* = \frac{\hat{\theta}_r^* - \hat{\theta}}{\sqrt{v_r^*}}$$

```
alpha <- 0.025/2

set.seed(1)
vs <- sapply(1:R, function(idx,sample){
  R <- 999
  sample.phi <- exp.maximLikelihoodEst.var(sample)
  sample.bootstrap <- rexp(length(sample),sample.phi)
  sample.bootstrap.phi <- exp.maximLikelihoodEst.var(sample.bootstrap)

  v <- bootstrap.var(sample.bootstrap,R)
  return(sqrt(v))
},sample)
set.seed(1)
distance.bootstrap.studantized <- sapply(1:R, function(idx,sample){
  R <- 999
  sample.phi <- exp.maximLikelihoodEst.var(sample)
  sample.bootstrap <- rexp(length(sample),sample.phi)
  sample.bootstrap.phi <- exp.maximLikelihoodEst.var(sample.bootstrap)

  v <- bootstrap.var(sample.bootstrap,R)

  (sample.bootstrap.phi - sample.phi)/sqrt(v)
},sample)
```

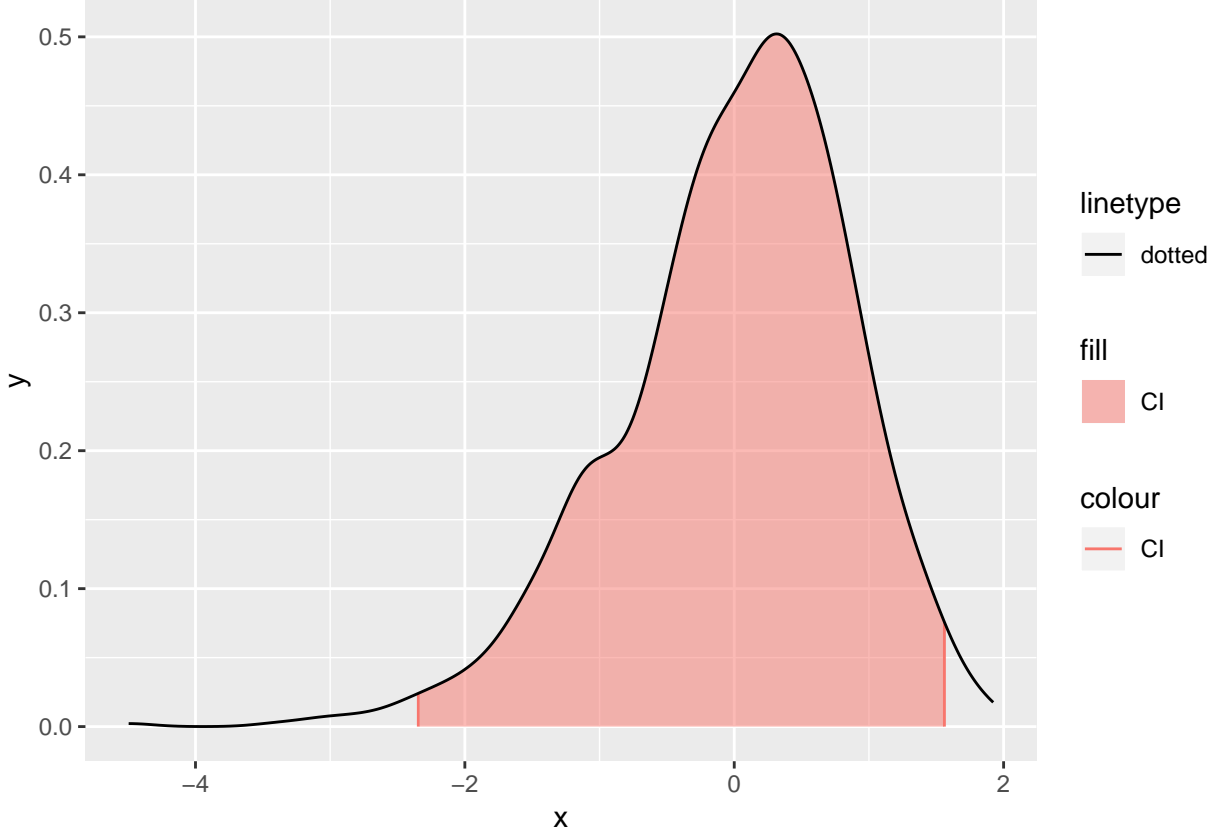


Figure 4: Density estimation of the pivot statistic  $z$

```
distance.booststrap.studantized <- c(distance.booststrap.studantized,0)
distance.booststrap.studantized.sorted <- sort(distance.booststrap.studantized, decreasing = F)

ahat.stdzd_left <- distance.booststrap.studantized.sorted[(R+1)*alpha]
ahat.stdzd_right <- distance.booststrap.studantized.sorted[(R+1)*(1-alpha)]
```

## 2.5 Discussion

Discuss the results of items (c) and (d). Choose  $\alpha = 0.025$  as the significance level for the confidence intervals.

---

The distributions in figures 2 and 4 do not present the same shape and this is due of the term  $v_r^*$  from  $z_r^* = \frac{\hat{\theta}_r^* - \hat{\theta}}{\sqrt{v_r^*}}$ . This change of shape can also be understood in figure 3 since this occurs because the term  $v_r^*$  acts like a regularizator.