Stat Computing HW1

Yaniv Bronshtein

1/30/2022

Import the necessary libraries

```
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
```

```
library(ggplot2)
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

Problem 1

Read data for Q1

```
df <- read.table('/Users/yanivbronshtein/Coding/Rutgers/Statistical_Computing_Repo/data/q1_data.txt',
                 header=TRUE)
```

Write an R code to compute the "average" blood pressure. (ABP) defined as a weighted average of the diastolic blood pressure and the systolic blood pressure. Since the heart spends more time in its relaxed state (diastole), the diastolic pressure is weighted two-thirds, and the systolic blood pressure is weighted one-third. Therefore, the average blood pressure could be computed by multiplying the diastolic blood pressure by 2/3,

and the systolic blood pressure by 1/3 and adding the two. An equivalent expression would be the diastolic pressure plus one-third of the difference between the systolic and diastolic pressures. Using either definition, add ABP to the data set.

```
df <- df %>% mutate(ABP=SBP/3 + DBP*(2/3))
```

Problem 2

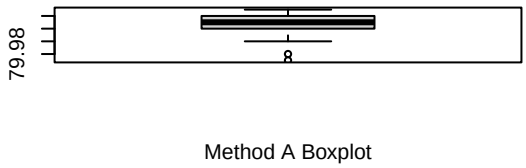
Rice (1995, p. 390) gives the following data (Natrella, 1963) on the latent heat of the fusion of ice (cal/gm): Method A: 79.98 80.04 80.02 80.04 80.03 80.03 80.04 79.97 80.05 80.03 80.02 80.00 80.02 Method B: 80.02 79.94 79.98 79.97 79.97 80.03 79.95 79.97 (a) Inspect the data graphically in various ways, for example, boxplots, Q-Q plots and histograms.

```
meth_a = c(79.98, 80.04, 80.02, 80.04, 80.03, 80.03, 80.04, 79.97, 80.05, 80.03,
           80.02, 80.00, 80.02)

meth_b = c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03, 79.95, 79.97)
```

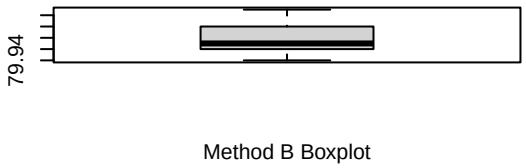
```
par(mfrow = c(3,2))
boxplot(meth_a, xlab="Method A Boxplot")
boxplot(meth_b, xlab="Method B Boxplot")
hist(meth_a)
hist(meth_b)

# qqnorm(meth_a, pch = 1, frame = FALSE)
# qqline(meth_a, col = "steelblue", lwd = 2)
qqPlot(meth_a)
```

```
## [1] 8 1
```

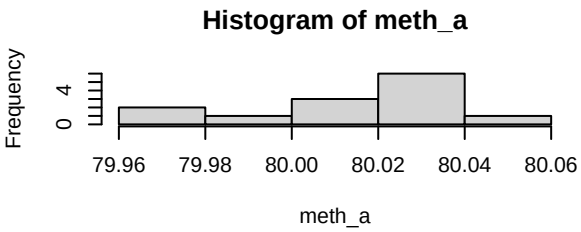
```
# qqnorm(meth_b, pch = 1, frame = FALSE)
# qqline(meth_b, col = "steelblue", lwd = 2)
qqPlot(meth_b)
```

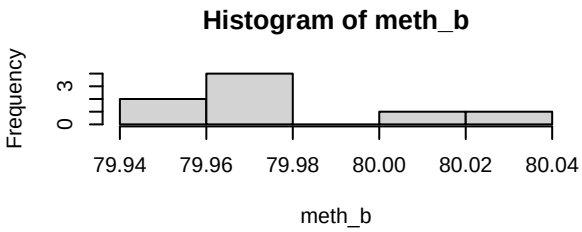


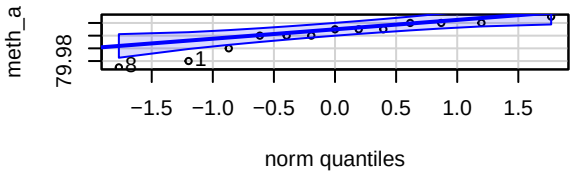
Method A Boxplot

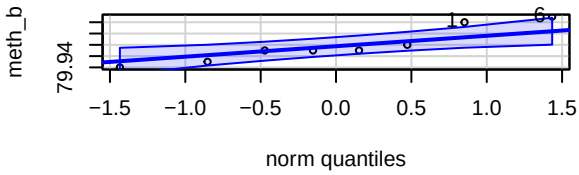


Method B Boxplot










```
## [1] 6 1
```

(b) Assuming normality, test the hypothesis of equal means, both with and without making the as- sumption of equal variances. Try a two sample t-test anova is for 3 or more samples.

1. With equal var.
2. Without equal var.

```
# Equal variance
t.test(meth_a, meth_b, var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  meth_a and meth_b
## t = 3.4722, df = 19, p-value = 0.002551
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01669058 0.06734788
## sample estimates:
## mean of x mean of y
##  80.02077  79.97875
```

```
# Unequal variance
t.test(meth_a, meth_b)
```

```
##
##  Welch Two Sample t-test
##
## data:  meth_a and meth_b
## t = 3.2499, df = 12.027, p-value = 0.006939
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  0.01385526 0.07018320
## sample estimates:
## mean of x mean of y
##  80.02077  79.97875
```

(c) Compare the result with a Wilcoxon/Mann-Whitney nonparametric two sample test.

```
wilcox.test(meth_a, meth_b, alternative='two.sided')
```

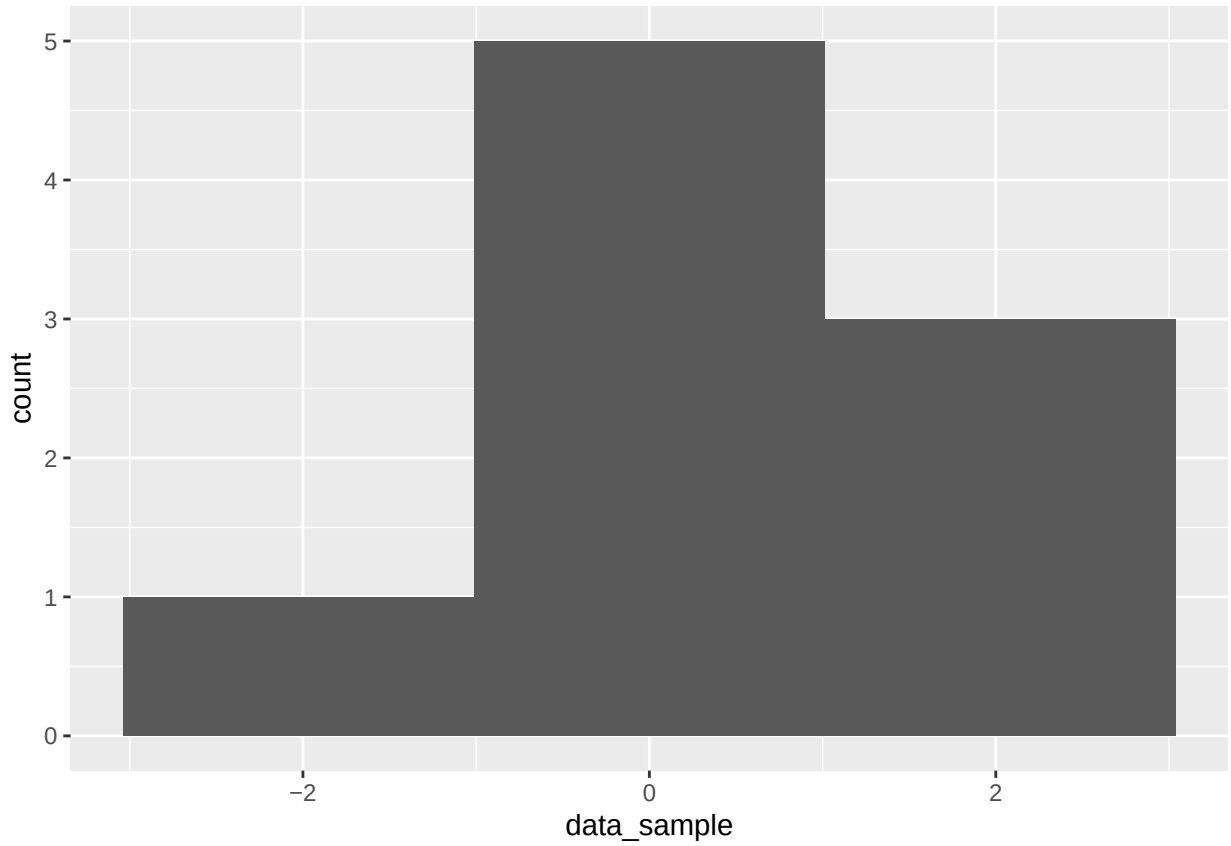
```
## Warning in wilcox.test.default(meth_a, meth_b, alternative = "two.sided"):
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  meth_a and meth_b
## W = 89, p-value = 0.007497
## alternative hypothesis: true location shift is not equal to 0
```

[Remark: For (a), solve it both by paper and pencil and also using a computer. For (b) and (c), use a computer for your calculation, but write out the corresponding formulas]

Question 3

```
data_sample = c(-1.43, -0.95, -0.19, 0.02, 0.14, 0.83, 1.35, 1.46, 2.62)
df_data_sample = as.data.frame(data_sample)

ggplot(data=df_data_sample, mapping=aes(x=data_sample)) +
   geom_histogram(bins = 3)
```



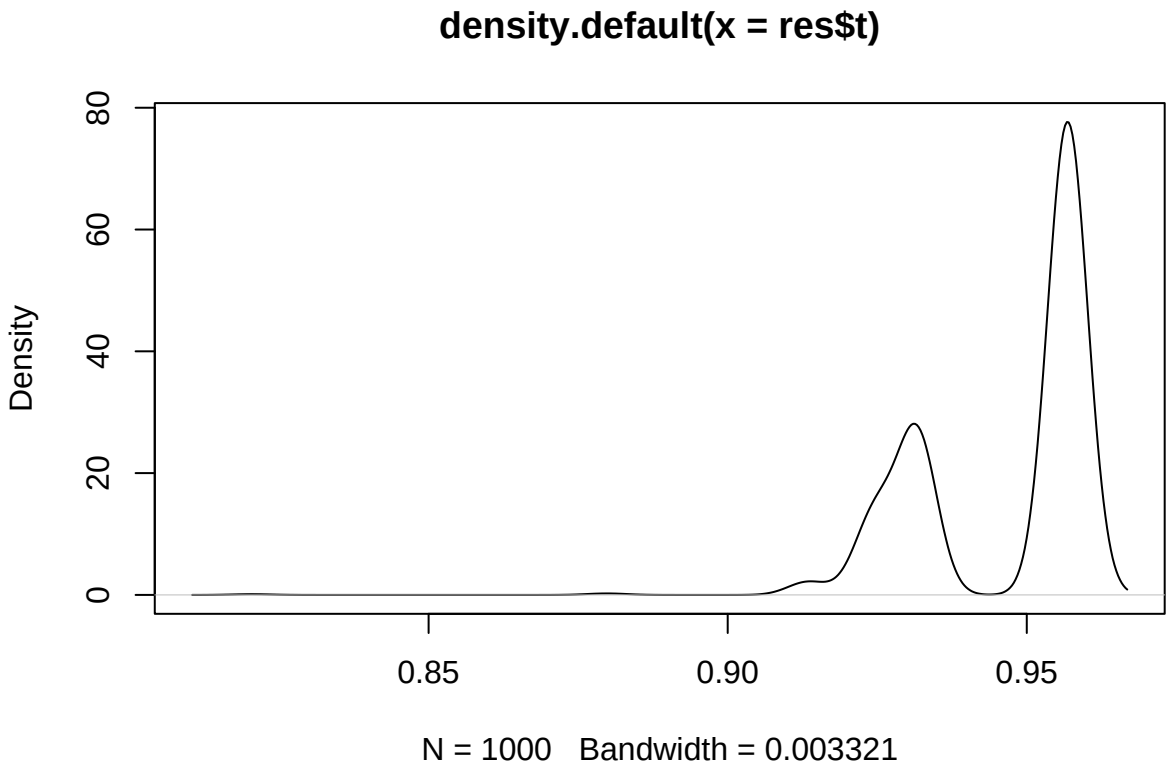
Problem 5

(b). Let the underlying truth be $\theta = 1$. Suppose we are only able to get n=60 samples. Generate your own observations $x_1, ..., x_{60}$. Based on them we can do bootstrap and get the bootstrap MLE $\hat{\theta}*$. Bootstrap for N=1000 times and plot the density of $\hat{\theta}^*$ Compare the distribution of $\hat{\theta}^{MLE}$ and $\hat{\theta}*$ according to the density plot you obtain. Are they symmetric or asymmetric?

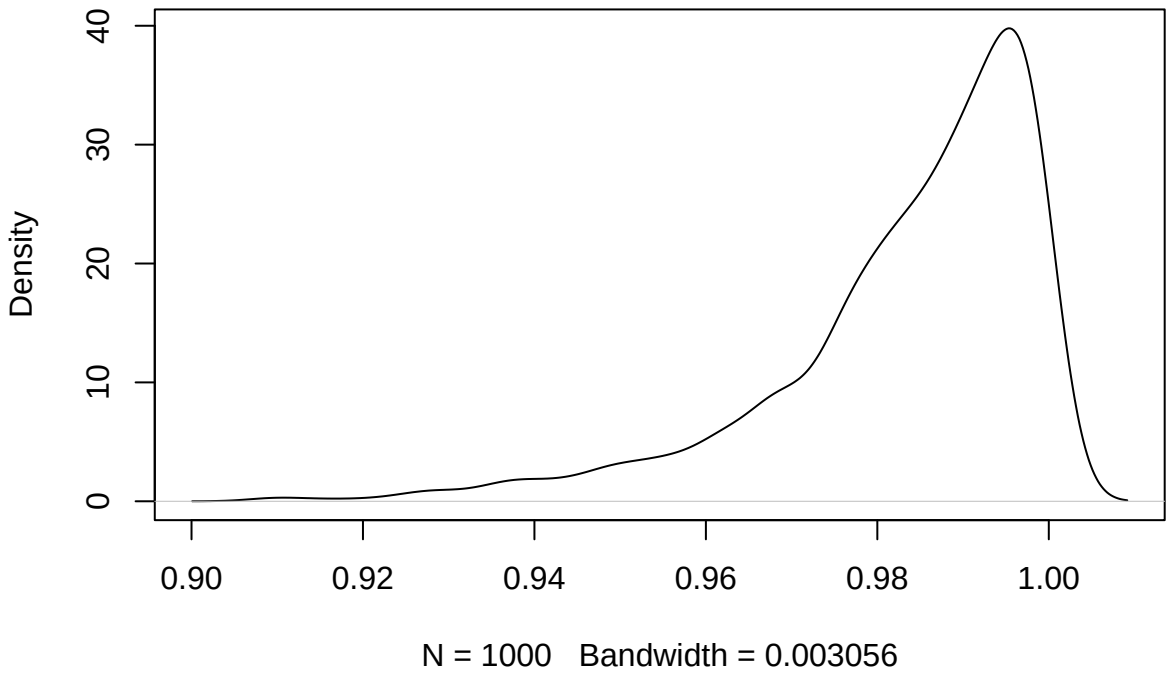
```
set.seed(101)
X <- runif(60,0,1)
uniform.boot <- boot(X, function(x, i) max(x[i]), R = 1000)

res <- uniform.boot

boot_estim <- res$t0
cat("Bootstrap MLE estimate: ", boot_estim, "\n")
```

```
## Bootstrap MLE estimate:  0.9568375
```

```
plot(density(res$t))
```




```
uniform.mle <- function(){
  arr <- list()
  #Resample each time
  for (i in 1:1000){
    arr[i] = max(runif(60,0,1)) }
  plot(density(unlist(arr)))
  cat("MLE Estimate:", max(unlist(arr)), ",\n")
  return(arr)
}

uniform.mle_arr <- uniform.mle()
```




```
## MLE Estimate: 0.9999886 ,
```

According to the density plots, both are assymetric since we have skew. (c). Since theta hat MLE is max(X1, X2, . . . , X60) from a continuous uniform distribution, the probability that this statistic equals one is zero.

The probability that the two estimates are equal is 0.648. One only needs to look at the the counts:

```
table(res$t)
```

```
##
## 0.820436094887555 0.879795730113983 0.913383485749364 0.913478652015328
##                 1                 2                 8                10
## 0.923318882007152 0.924804413458332 0.931634427979589 0.956837461562827
##                20                90               221               648
```

To get the probability, we perform the calculation below which amounts to 648/10000

```
prob <-sum(res$t==max(X))/1000
prob
```

```
## [1] 0.648
```

(d). Based on the 1000 $\hat{\theta}^*$ obtained in (b), build the bootstrap 90% confidence interval for θ. Simulate the data for N=1000 times and get the 90% confidence interval for θ based on $\hat{\theta}^{MLE}$. Compare these two intervals.

```
confint90 <- boot.ci(res, conf=0.90, type="norm")
confint90
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = res, conf = 0.9, type = "norm")
```

```
##
## Intervals :
## Level      Normal
## 90%   ( 0.9429,  0.9912 )
## Calculations and Intervals on Original Scale
```

Now let us do without bootstrap

```
###
#Note: We use the t-distribution since we have over 30 samples
err <- 1.67 * sd(unlist(uniform.mle_arr)/(sqrt(100)))

unif_no_bootMean <- mean(unlist(uniform.mle_arr))
cat("Confidence interval no bootstrap:", c(unif_no_bootMean-err,unif_no_bootMean+err))
```

```
## Confidence interval no bootstrap: 0.9807871 0.9859646
```

(e) The max we find through the bootstrap method is our sample space. Bootstrapping will never reach this value. The bootstrap central limit theorem will thus not hold because of the gap.