Fezeka Nzama NZMFEZ001 Project 1

Project Aims & Objectives

In this study we aim explore samples, bootstrapping, hypothesis testing as well as confidence interval construction. In question 1, we conduct an exploratory analysis of the data, perform a 1-sample hypothesis test and construct a confidence interval for both a population mean and a population median. Question 2 focuses on 2-sample hypothesis testing and confidence interval construction, whilst also comparing the results obtained from bootstrapping to those obtained from the use of normal theory. Question 3 explores 4-sample ANOVA testing using bootstrapping.

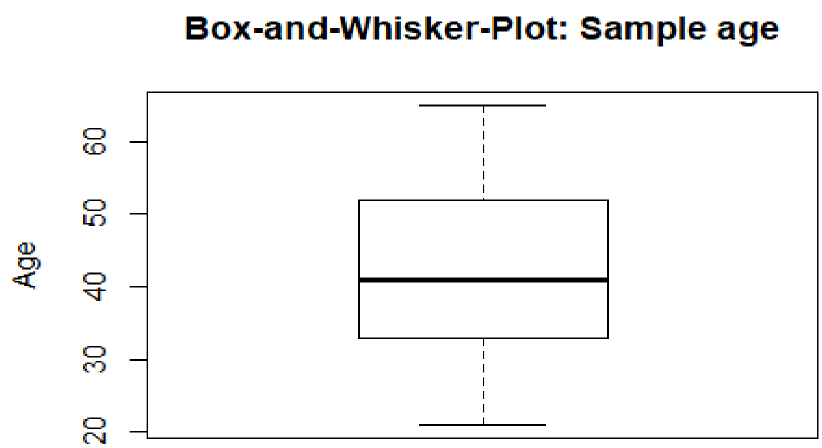
**Please notes that comprehensive code for all 3 questions as well as the answer for Question 3B are provided in the Appendix.*

Question 1

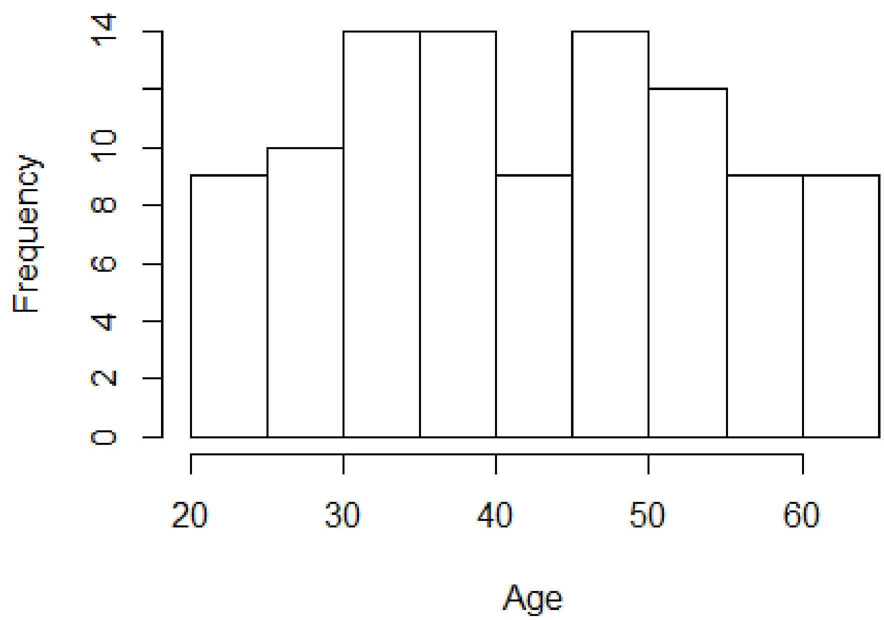
The following is an analysis of the Income and Expenditure survey conducted by Stats SA, with a focus on the age of the head of the household, and the total monthly income of the household (denoted by the hhinc variable).

Question 1A) The following is an exploratory analysis of the sample age.

```
#The five number summary of the sample age.
summary(age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   21.00   33.00   41.00   42.58   52.00   65.00

#The variance and standard deviation of the sample age.
var(age)

## [1] 147.6804

sd(age)

## [1] 12.15238

boxplot(age, ylab = "Age", main="Box-and-Whisker-Plot: Sample age")

hist(age, xlab = "Age", main = "Histogram showing the sample age of the head
of the household")
```

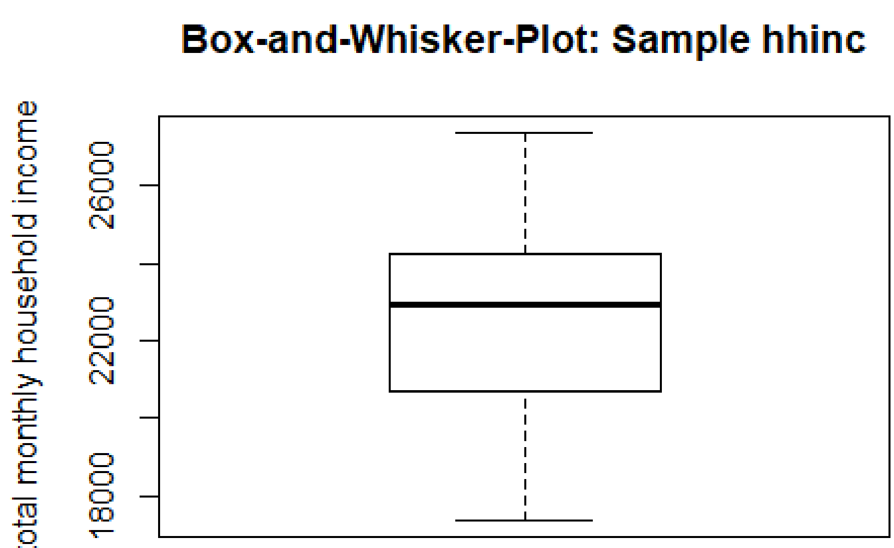


"Histogram showing the sample age of the head of the household

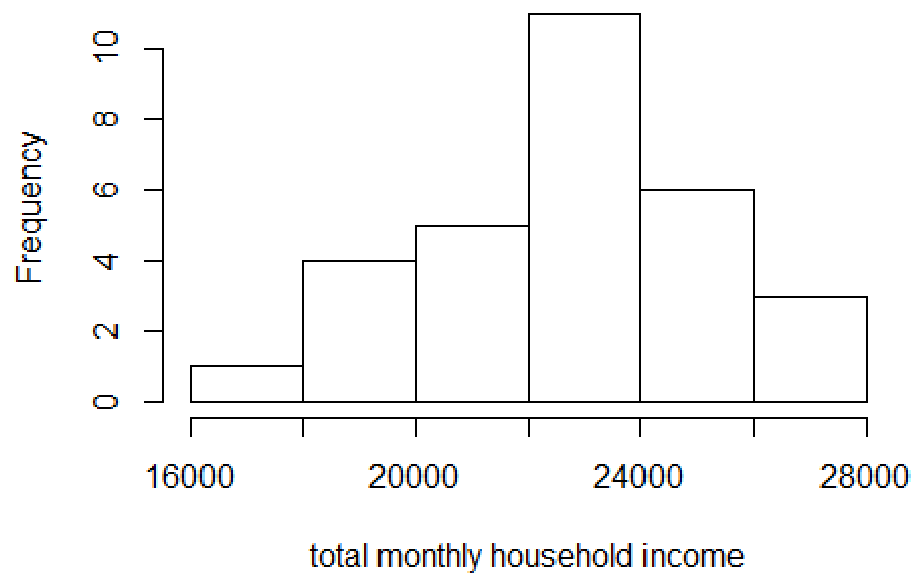


The following is an exploratory analysis of the sample hhinc.

```
#The five number summary of the sample total monthly income of the household.
summary(hhinc)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17372   20752   22912   22649   24182   27362

#The variance and standard deviation of the sample total monthly income of th
e household.
var(hhinc)

## [1] 6200217

sd(hhinc)

## [1] 2490.024

#The boxplot for the sample total monthly income of the household.
boxplot(hhinc, ylab="total monthly household income", main="Box-and-Whisker-P
lot: Sample hhinc")

hist(hhinc, xlab = "Household monthly income", main = "Histogram showing the
sample total monthly income in a household")
```



Histogram showing the sample total monthly income of the household



Question 1B)

In the following section we construct a 95% confidence interval for the sample age. To do this we will use bootstrapping to simulate the sampling process. Additionally, we make use of the bootstrap assumption: $\overline{X}$ - $\mu \sim \overline{X}_b$ - $\overline{X}$

```
#Using the vector of sorted bootstrap means we find the upper and lower bound of the set.
lowBound = agesortm[4000*.025]
uppBound = agesortm[4000*.975]
truLow=(uppBound*-1)+(2*mean(age))
truUpp =(-1*lowBound)+(2*mean(age))

cat("The 95% confidence interval of the mean age is (",truLow,",",truUpp,")")

## The 95% confidence interval of the mean age is ( 40.14 , 44.98 )
```

The 95% confidence interval can be interpreted as being the probability of stating the correct bounds. Thus, in repeated sampling, bounds between 40.14 and 44.98 were observed 95% of the time, hence the sample mean age can be said to sit within those bounds in 95% of observations.

Question 1C)
In the following section we use bootstrapping to test the hypothesis that the population mean age for the head of the household is less than or equal to 43. We will make use of the bootstrap assumpion: $\overline{X}$ - $\mu \sim \overline{X}_b$ - $\overline{X}$

H0: $\mu \leq 43$
H1: $\mu > 43$

```
#calculate the sampling error assuming the null hypothesis is true
SE = mean(age)-43
```

Therefore assuming $\mu \leq 43$:
$\overline{X}$ - $\mu \leq$ -0.42.

Applying the bootstrap assumption :
$\overline{X}_b$ - $\overline{X} \leq$ -0.42
$\overline{X}_b \leq$ -0.42 + $\overline{X}$

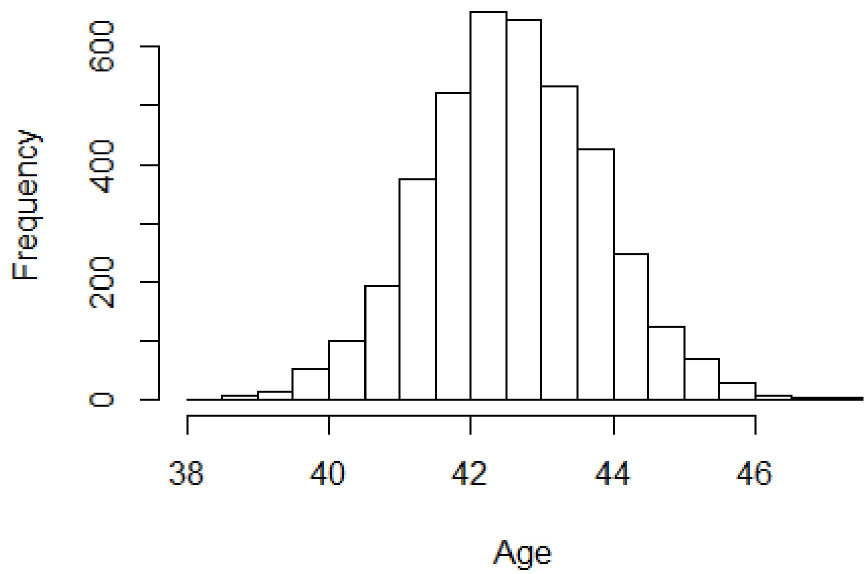
```
bMean = SE + mean(age)
index = match(bMean, agesortm)

#The pvalue for our hypothesis test is be calculated by summing up the number
of bootstrap means that are greater than bMean and dividing that by the total
number of bootstrap means

pvalue = (4000 - index)/4000
pvalue

## [1] 0.6375
```

Our observed p-value of 0.6375 does not indicate that there is a significant difference in our assumed value for μ and the true population mean age. Thus, assuming the null hypothesis is true there isn't significant evidence that it is unlikely that the sample we drew would come from a population with that value for μ. As such we cannot reject **H0**.

Question 1D)
In the following section we construct a histogram for the bootstrap means for the age of the head of the household.

```
hist(agesortm, xlab = "Age", main = "Histogram showing the distribution of th
e bootstrap means for the age of the head of the household variable")
```

Histogram showing the distribution of the bootstrap means for the age of the head of the household



The histogram takes on a shape that is very similar to that of the normal distribution. However, it is visibly more peaked, unlike the normal distribution. Thus one can assume that the bootstrap means follow a t-distribution. Additionally, the histogram indicates that the sample mean age for the head of the household lies between 42 and 43 years of age, this is the inverval at which the histogram reaches it's peak. The histogram also shows that the range of ages for head of the household extends from about 38 to 48, with most of the bootstrap means concentrated around the range from 41 to 44.

Question 1E)
In the following section we construct a 90% confidence interval for the median household income ($\varepsilon_{0.5}$). We make use of bootstrapping to do this, as well as the bootstrap assumption applied to medians rather than means in this case.

```
 #lower and upper median bound
lowMed = hhincM[4000*.05]
uppMed = hhincM[4000*.95]
popLow = -(uppMed - median(hhinc))+median(hhinc)
popUpp = - (lowMed - median(hhinc))+median(hhinc)
cat("(",popLow,",",popUpp,")")

## ( 22171.43 , 23771.28 )
```

The median is often used instead of the mean when estimating the probable value of a random variable when the data tends to exhibits outliers which may skew the mean as the median is a more robust measure. The 90% confidence interval for the population household income median is the frequency in repeated sampling of stating correct bounds for the population household income median. In this case the confidence interval is (22171.43, 23771.28). Thus in repeated samples,the sample median lies within that range 90% of the time.

Question 2

In this question we compare the marketing strategies to sell a new Apple juice concentrate employed in two cities with the aim of identifying which (if any) is more effective. In the first city the emphasis of the marketing communication is on convenience, whilst in the second city the emphasis is on price. The number of packages sold per week has been recorded for each city, and these numbers are used as an indicator of the success of each marketing strategy.

Question 2A)

In this section we test whether there is a significant difference between the means in the city where convenience is emphasisied versus the city where the price is the focal point of the marketing material. To do this we will use bootstrapping and the bootstrap assumption: $\overline{X}$ - $\mu \sim \overline{X}_b$ - $\overline{X}$

(Variances are assumed to be equal)

H0: $\mu_1 = \mu_2$
H1: $\mu_1 \neq \mu_2$

```
samplingErr = mean(city1)-mean(city2)-0
samplingErr

## -20.4

uppTail = -samplingErr
```

Therefore assuming **H0** is true:
-20.4 < $\overline{X}_1$ - $\overline{X}_2$ - (μ_1 - μ_2) < 20.4
-20.4 < $\overline{X}_1$ - $\overline{X}_2$ - 0 < 20.4

Applying the bootstrap assumption :
-20.4 < $\overline{X}_{b1}$ - $\overline{X}_{b2}$ - 0 < 20.4

P-val = Pr[($\overline{X}_{b1}$ - $\overline{X}_{b2}$) < -20.4] + Pr[($\overline{X}_{b1}$ - $\overline{X}_{b2}$) > 20.4]

```
greaterThan = match(uppTail, sortCityDiff)
lessThan = match(samplingErr, sortCityDiff)
pval = ((5000-greaterThan)+(lessThan))/5000
pval

## 0.6014
```

Our observed p-value of 0.6014 does not indicate that there is a significant difference in the mean number of packages sold in City 1 (where convenience was emphasisied) as opposed to those sold in City 2 (where the emphasis was on the price). Thus we cannot reject the null hypothesis, meaning that from this result we cannot assume that the marketing employed in each city made a significant difference to the number of units of Apple juice sold.

Question 2B)
In the following section we construct a confidence interval for the difference in population means of packages of Apple Juice sold in City 1 as opposed to City 2.

```
lowBound = sortCityDiff[5000*.025]
uppBound = sortCityDiff[5000*.975]

#calculation of the true population mean difference bounds
PlowBound = (uppBound*-1)+(mean(city1)-mean(city2))
PuppBound = (lowBound*-1)+(mean(city1)-mean(city2))
confi = cat("(",PlowBound,",",PuppBound,")")

## ( -98.1 , 59.6 )
```

The 95% confidence interval can be interpreted as being the probability of stating the correct bounds of the difference in means between city1 and city2. Thus, in repeated sampling, bounds between -98.1 and 59.6 were observed 95% of the time, hence the sample mean difference in packgaes sold of Apple Juice can be said to sit within those bounds in 95% of observations.

Question 2C) In this section we will be testing the samples from our two cities for equality of variance using bootstrapping.

H0: $\sigma_1 = \sigma_2$
H1: $\sigma_1 \neq \sigma_2$

```
sampleFstat = (var(city1))/(var(city2))

#cityFs is the bootstrapped Fstats calculated from the data - see appendix fo
r detail
pvalV= length(cityFs[cityFs>=sampleFstat])/5000
pvalV

## [1] 0.4974
```

We found an observed p-value of 0.4974 which does not indicate a significant difference between the variance in packages of Appe juice sold in City 1 as opposed to City 2. Thus we cannot reject the null hypothesis, as there isn't signicficant evidence to suggest that the two variances are different.

Question 2D)
In this section we calculate the results obtained from normal theory and bootstrapping.

```
#the p-value calculated using normal theory for the equality of means test
Tp_value = t.test(city1,city2)
Tp_value$p.value
```

```
## [1] 0.6286176

#the p-value calculated using normal theory for the equality of variance test
pf(sampleFstat, 9,9)

## [1] 0.5064688
```

Using normal theory we found that for the equality of means test p-value = 0.6286176 as opposed to the result obtained from bootstrapping of p-value = 0.6014. This shows the p-value obtained via bootstrapping to be very close to that obtained using normal theory. Similarly, when testing for equality of variance the bootstrapped p-value = 0.4974, whilst normal theory gave a p-value = 0.5064688 again showing similarity in the p-value obtained by bootstrapping to normal theory. Thus both sets of results show that normal theory is consistent to results obtained by repeated sampling and observation and as such validate the use of normal theory in hypothesis testing as a means to estimate the p-value when repeated sampling is not possible.

Question 3

In this section we will review data provided by the Internal Revenue Service. The IRS wishes to improve the wording and format of it's tax return form so as to make them easier to fill out. As such the IRS conducted an experiment where 120 individuals were grouped into 4 groups of 30 and asked to fill out tax return forms whilst being timed. Each group received a different form (3 groups getting new forms, whilst one used the old form).

We wish to descern whether there is a significant difference in the mean times of each group and thus determine if there is significant evidence to motivate a change in the wording and format of the forms. We will do this using the bootstrapped ANOVA method.

H_0: $\mu_i = \mu$
H_1: $\mu_i \neq \mu$

The following statistics were obtained from the provided sample data:
SSE = 1.186726710^{5}
SST = 1.244403310^{4}
F-stat = 4.1594525

```
p_value=length(Ratios[Ratios>Forig])/B
p_value

## [1] 0.0075
```

Our Bootstrapping process resulted in a p-value of 0.0075. This p-value is significantly small and thus suggests that there is significant difference in mean times for filling in the form. As such we can reject **H_0** at a significance level of 1%.

From this test we can see that there may be value in altering the wording and formatting of the tax return forms, as at least one of the forms had a mean fill out time which was different to the others. This thus provides motivation for further study to be done around this data to determine where the differences actually lie, ie. which form or forms had a different mean time. From this further investigation we would then be able to determine which form has the best wording and formatting and advise the IRS to adopt that form for use.

Plagiarism Statement: The work presented is all my own.

Appendix

```
#Question 1 code & logic

#reading in data
mydata = read.table("F:\\sta3030 - Project 1\\incexp.txt", header = TRUE)
age = mydata$AGE[118:(118+99)]
hhinc = mydata$HHINCOME[118:(118+29)]


#Question 1B - Bootstrapping
#The 'agebstr' variable is the matrix in which we will store bootstrap sample
age.
agebstr<-matrix(0,nrow=4000, ncol=100)
for(i in 1:4000){
  samp = sample(age, size=100, replace = TRUE)
  agebstr[i,] = samp
}

#The 'agebstrm' is a vector containing the mean of each bootstrap sample, whi
lst 'agesortm' contains the means sorted in ascending order.
agebstrm = apply(agebstr, 1, mean)
agesortm = sort(agebstrm)

#the following is the logic used in calculating the population confidence int
erval
#xb = bootstrap mean, ageMean = smaple mean, popMean = populaion meanS
#Pr[lowBound<XB<uppBound]
#Pr[lowBound-ageMean<Xb-ageMean<uppBound-ageMean]
#Pr[lowBound-ageMean<agemean-popMean<uppBound-ageBound]
#Pr[-(uppBound-ageMean)<popMean-ageMean<-(lowBound-ageMean)]
#Pr[-(uppBound-ageMean)+ageMean<popMean<-(lowBound-ageMean)+ageMean]

#Question 1E - Bootstrapping
#hhincM is a vector in whch the household income bootstrapp medians are store
d
hhincM<-vector("numeric", length = 4000)

#bootstrapping
for(i in 1:4000){
  samp =sort(sample(hhinc, size=30, replace = TRUE))
  #find the median value in the bootstrap
  hhincM[i]=median(samp)
}

#sorting the vector
hhincM = sort(hhincM)
```

```r
#Question 2 code & logic

#reading in required data
quest2 = read.table("F:\\sta3030 - Project 1\\A1.txt",header= TRUE)
city1=quest2$City1
summary(city1)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   444.0   482.0   552.0   555.5   613.5   712.0

city2 =quest2$City2
summary(city2)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   464.0   529.5   557.0   575.9   632.5   759.0

#Question 2A)

#A combined data set is created be combining the data from the two cities
#Bootstraps of the combined are created and the means of the bootstraps are s
tored in the bootstrap mean vectors(city1B and city2B). The mean differences
are also then computed and stored in vector cityDiff
combined = c(city1,city2)
city1B <-vector(mode="numeric", length = 5000)
city2B <-vector(mode="numeric", length = 5000)
cityDiff <-vector(mode="numeric", length = 5000)
for(i in 1:5000){
  samp = sample(combined, size = 20, replace = TRUE)
  city1B[i] = mean(samp[1:10])
  city2B[i] = mean(samp[11:20])
  cityDiff[i] = city1B[i]-city2B[i]
}

sortCityDiff =sort(cityDiff)

#Question 2B)
#95% confint - find index of (5000*.025) & (5000*.975)

#lowBound<bmean(1-2) - smean(1-2)<uppBound     **note that is smean(1-2)=0
#lowBound<bmean(1-2) <uppBound
#lowBound<smean(1-2)- u(1-2)<uppBound
#-uppBound<u(1-2)-smean(1-2)<-lowBound
#therefore: -uppBound+smean(1-2)<u(1-2)<-lowBound+smean(1-2)


#Question 2C)
#the following code block was used to create a vector (cityFs) containing the
bootstrap f-stats of city one and two
cityFs <-vector(mode="numeric", length = 5000)
for(i in 1:5000){
```

```r
  samp = sample(combined, size = 20, replace = TRUE)
  city1v= c(samp[1:10])
  varCity1v = var(city1v) #calculate variance of bootstrap of city1
  city2v= c(samp[11:20])
  varCity2v=var(city2v)   #calculate variance of bootstrap of city2
  cityFs[i] = varCity1v/varCity2v  #calculate Fstat and retain that
}

#Question 3

#reading the data
quest3 = read.table("F:\\sta3030 - Project 1\\aov5.txt",header= TRUE)
quest3

##    Form_1 Form_2 Form_3 Form_4
## 1      23     88    116    103
## 2      59    114    123    122
## 3      68     81     64    105
## 4     122     41    136     73
## 5      74    108     99     87
## 6      90     92    156     81
## 7      70     52    175    120
## 8      87     54     93    169
## 9     155    103     77    130
## 10    120     50     88     56
## 11    124    135     91    101
## 12    103     76    118    143
## 13     54    143     86    106
## 14     90    124    164    129
## 15    124    151    101    104
## 16     80     96     74    169
## 17     69     76    124     69
## 18    123    128    137     76
## 19     76     60     69     55
## 20     71    127    136    138
## 21     94    109    127    122
## 22    167    122    135    139
## 23     69     88     97    138
## 24    105    109    103    132
## 25     98     90     86     99
## 26     73     56    121     64
## 27     79    105     98     89
## 28     61     64     59    128
## 29    121    127     91    127
## 30     56    104     61    161

#treatment groups
form1 = quest3$Form_1
form2 = quest3$Form_2
form3 = quest3$Form_3
```

```
form4 = quest3$Form_4

#number pof treatment groups
k = 4
#number of participants per group
ni = 30
#total no. participants
N = 30*4
#overall mean
Y.. = (sum(form4,form3, form2, form1))/N
#Y..

B = 4000

#calculate values for the provided sample
sSSE=sSST=0
#treatment groups placed in list
forms = list(form1, form2, form3, form4)
#mean of each treatment group
Yi. = vector( mode = "numeric", length = 4)
for(i in 1:k){
  Yi.[k]=mean(forms[[k]])
  #sum((forms[[k]]- Yi.[k])^2) does the (Yij-Yi)^2 for all values in that loc
ked list
  #vector minus a vector basically
  sSSE=sSSE+sum((forms[[k]]- Yi.[k])^2)
  sSST =sSST+ni*((Yi.[k]-Y..)^2)
}
Forig = (sSST/(k-1))/(sSSE/(N-1))

#bootstraps
combined = unlist(forms)  #combined list of all our sample data
Ratios = vector("numeric") #empty vector to store boostrap F-stats
sampleToPrint = list("numeric", length = 3)
for (i in 1:B){
  bSSE=bSST=0
  Yi.=vector("numeric")
  samp = sample(combined, size = 120, replace = TRUE)
  bstrp = list("numeric", length = k)
  bstrp[[1]]= samp[1:30]
  bstrp[[2]]= samp[31:60]
  bstrp[[3]]= samp[61:90]
  bstrp[[4]]= samp[91:120]
  if(i<4){
    sampleToPrint[[i]]=samp
  }
  Y.. = mean(unlist(bstrp))
  for(j in 1:k)
  {Yi.[j]=mean(bstrp[[j]])
```

```
  bSSE=bSSE+sum((bstrp[[j]]-Yi.[j])^2)
  bSST=bSST+ni*(Yi.[j]-Y..)^2
  }
  Ratios[i]=(bSST/(k-1))/(bSSE/(N-k))
}
```

Question 3B)

```
#3 bootstrap samples generated -
print(sampleToPrint)

## [[1]]
##   [1] 175  96  70 167 122  76 109 124  59  61 120  90  23  54  99  76  90
101
##  [19] 169  64 104 127 104  64 139  90  80  59  59  56 118 169 156  77  76
132
##  [37] 124 103 103 127 137 122  69  60  97  96  99  87 122  97 118  64 122
52
##  [55]  54 105 124 122  81  77  56  81  87 105  56 175  90  88  23 138 128
155
##  [73]  87 156 139  96  90  59 128  55  64  50 169 103  68  64 103 128  81
50
##  [91]  91  81 121  80 121  69  76  64 137 161  98 137 122 122  76 120 137
99
## [109]  54 101  59  56 104 104  56 128 161 121 104  90
##
## $length
##   [1] 120  69 109 161 123 127  70  88  69 135  97 135 124 135 161  90  91
169
##  [19] 127  41 123  87  87 129  76  69  79 124  64  56 120 155 130 120  76
88
##  [37] 108 105 155 109  88  87  87  73 136 127 103 127 169  76  99  56 124
103
##  [55] 169 127 167 169 143 137 120  89  76 169  23 114 106 127 101  94  86
101
##  [73]  71  76 123 151  59  60 129  64  77 122 136 103 128  64  64 143  64
130
##  [91] 106  81  79 156  86  88  88  73 137  41 127  90  70  41  73 143 118
135
## [109] 143  88 137  76  64  92 138 132  52  86 130  69
##
## [[3]]
##   [1]  98  76  69 127 135  68 108  61  92  88  91 143  81  55  81 135  90
90
##  [19] 156 114 161  64 124 127  97 114 127 136  55 124  52  60  74  76  64
69
##  [37]  77  61  88 105 122  76 151 155 122  99  61  73 138  81 103 109 103
89
##  [55] 108  90  76 105 155 122  87 123 169  90 105 123  76 104 127 103 124
74
```

```
##  [73]  56 108  90  80  60  54 167  76 143 109  80 114  99  54 104  56 105
81
##  [91]  87 175  56 132  93 143 127 143 104 161  99  90 128 155  64 103  52
74
## [109]  41 121  86  73 122  50 129  71 122 128  69 124
```