HW2

Homework 2

Q1:

Table 1 displays the heights of the presidents and their opponents in the U.S. presidential election of 1948 through 2008.

(A)

> Q: Enter the data in R as a data frame with 3 columns: year, winner and opponent. Enter the height of the winner candidate in the column winner, the height of the opponent candidate in the column opponent. [5 points]
>
> A:

```
years <- seq(from = 1948, to = 2008, by = 4)
winnerHeights <- c(175, 179, 179, 183, 193, 182, 182, 177, 185, 185, 188, 189, 189, 182, 182, 185)
opponentHeights <- c(173, 178, 178, 182, 180, 180, 185, 183, 177, 180, 173, 188, 187, 185, 193, 175)
heightDf <- data.frame("year" = years, "winner" = winnerHeights, "opponent" = opponentHeights)
```

(B)

> Q: Create a dataframe with the same fields as before, but including an additional column called difference which is the differences in height between the winner and opponent in the column difference. [5 points]
>
> A:

```
require(dplyr)
```

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

```
heightDfDiff <- heightDf |>
  mutate("difference" = winner - opponent)
```

(C)

Q: Add a new column taller.won to your data frame in (a) with logical values (TRUE/FALSE) determining whether the taller candidate won the election. [5 points]

A:

```
heightDfDiff <- heightDfDiff |>
  mutate("taller.won" = winner > opponent)
```

(D)

Q: Use the table function to display percentages of TRUE/FALSE in the column taller.won. Interpret the result. [5 points]

A:

```
prop.table(table(heightDfDiff$taller.won))
```

```
##
## FALSE  TRUE
##  0.25  0.75
```

A majority (75%) of the General Elections in the United States from 1948 to 2008 had the taller candidate win.

(E)

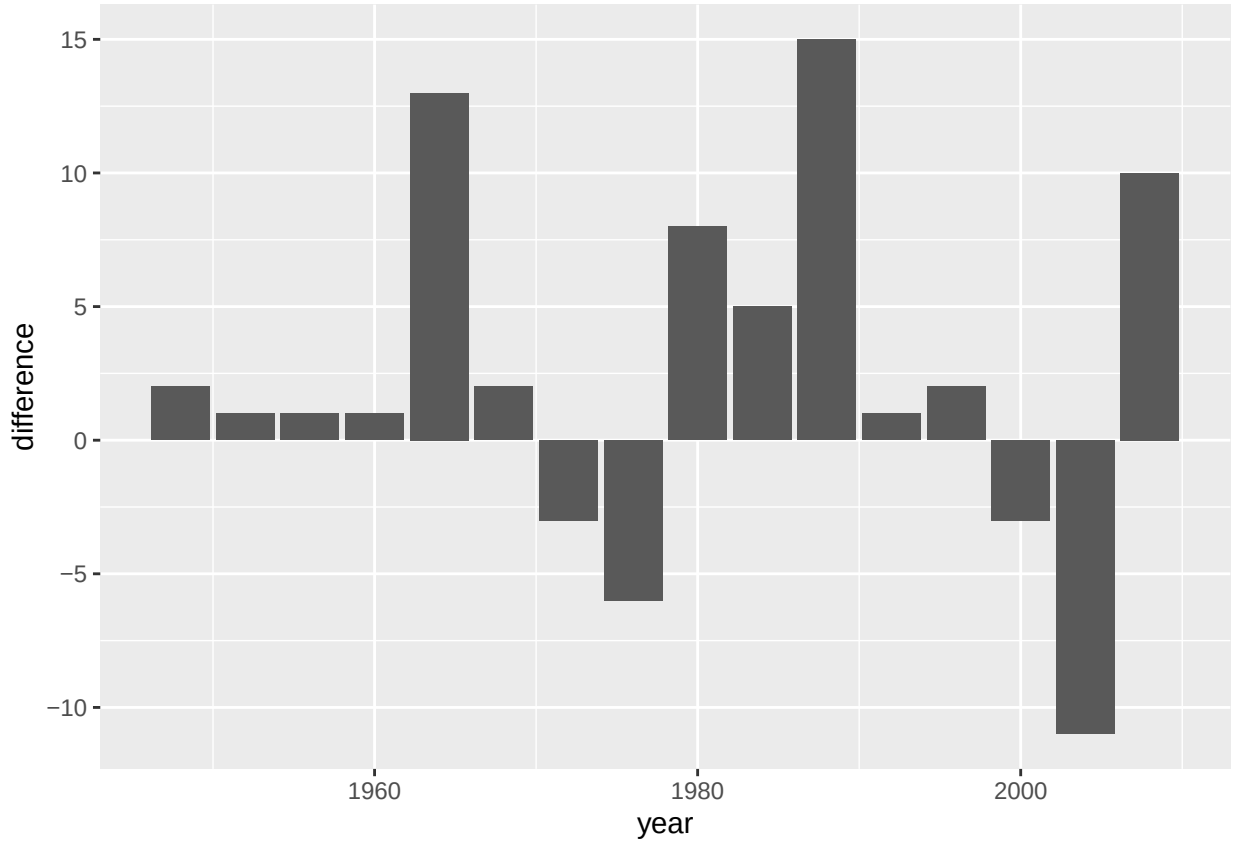
Q: Display a bar plot of the difference column. Use the rev function to reverse the order of the differences so that the election year is increasing from left to right. [5 points]

A:

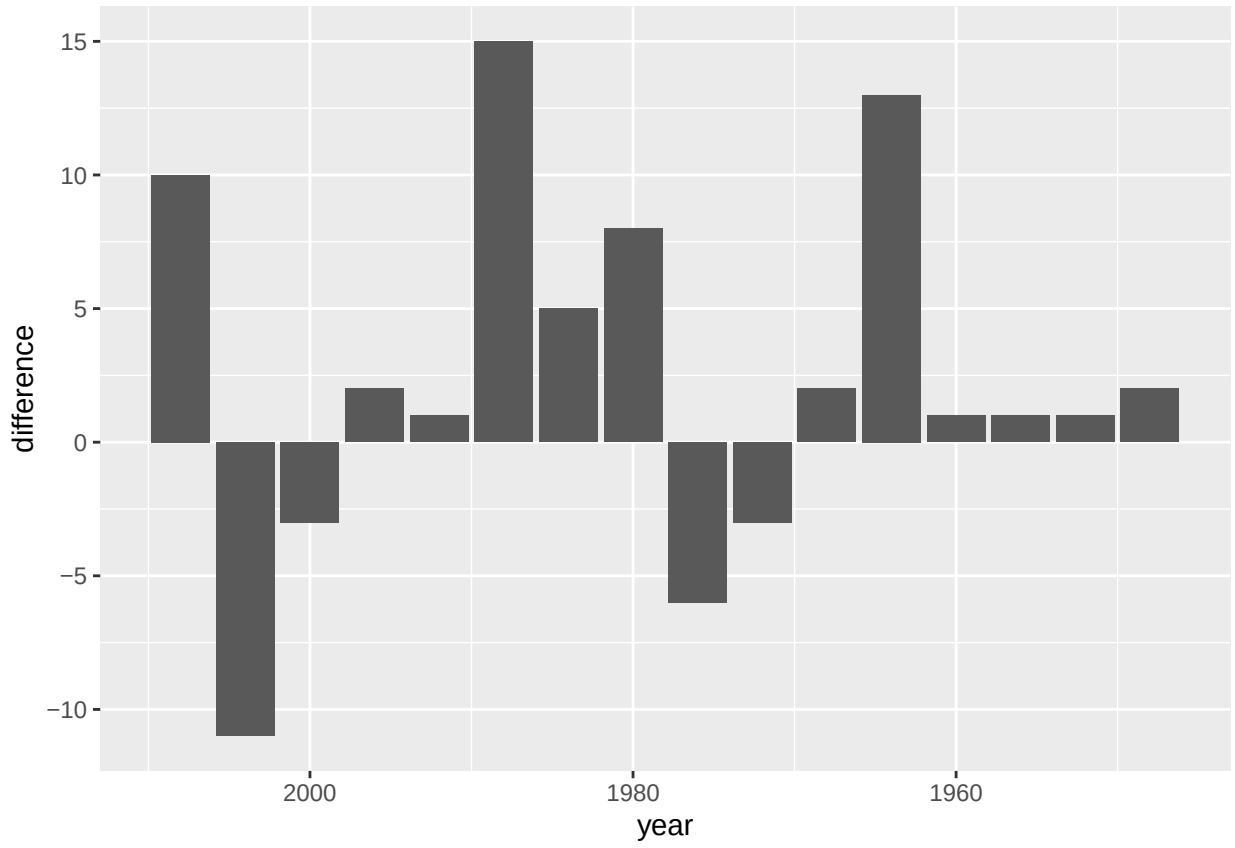
```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
p1<-ggplot(data=heightDfDiff, aes(x=year, y=difference)) +
  geom_bar(stat="identity")
p1
```




```
p2<-ggplot(data=heightDfDiff, aes(x=year, y=difference)) +
  geom_bar(stat="identity") +
  scale_x_reverse()
p2
```



(F)

Q: Using the function write.table() perhaps, save the table obtained in 1(c) above in a new file called heights.csv. (Note that you do not have to turn in the created file, just the complete function that you used.) Make sure that the created file is as you would like. [5 points]

A:

```
names(heightDfDiff)[5] <- "tallerwon"
write.table(heightDfDiff, file = "heights.csv", row.names = FALSE, col.names = TRUE, sep = ",")
```

Q2:

Read the dataset students.txt available in the Datasets section of Canvas into an R object named as students. [5 points]

(A)

Q: What is the mean height and shoesize of all the students? What about their standard deviations? [5 points]

A:

```
studentDf <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/STAT 5790/PS/PS2/students
```

```
mean(studentDf$height)
```

```
## [1] 169.7647
```

```
mean(studentDf$shoesize)
```

```
## [1] 40.47059
```

```
sd(studentDf$height)
```

```
## [1] 7.578996
```

```
sd(studentDf$shoesize)
```

```
## [1] 2.695312
```

(B)

Q: How many female students are in the sample? How many male students? [5 points]

A:

```
require(dplyr)
studentDf |>
  filter(gender == "male") |>
  nrow()
```

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

```
studentDf |>
  filter(gender == "female") |>
  nrow()
```

```
## [1] 9
```

(C)

Q: Recode the population variable with color names (kuopio = blue, tampere = red), and create a new dataset called studentsnew. [5 points

A:

```
studentsnew <- studentDf |>
  mutate("population" = case_when(population == "kuopio" ~ "blue",
                        population == "tampere" ~ "red"))
```

(D)

Q: Make two subsets of the dataset students. Split it in two according to gender and export the two datasets to female.txt and male.txt files. [5 points]

A:

```
female <- subset(studentsnew, gender == "female")
male <- subset(studentsnew, gender == "male")

write.table(male, file = "male.txt", row.names = FALSE, col.names = TRUE, sep = ",")
write.table(female, file = "female.txt", row.names = FALSE, col.names = TRUE, sep = ",")
```

```
boysWillBeBoys <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/STAT 5790/PS/PS2/mal
girlsWillBeGirls <- boysWillBeBoys <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/S
```

(E)

Q: make two new datasets of the dataset students that containing individuals below and above the median height. Export the two new datasets to below.csv and abovem.csv files. [5 points]

A:

```
medianHeight <- median(studentsnew$height)

below <- subset(studentsnew, height < medianHeight)
above <- subset(studentsnew, height > medianHeight)

write.table(below, file = "below.csv", row.names = FALSE, col.names = TRUE, sep = ",")
write.table(above, file = "above.csv", row.names = FALSE, col.names = TRUE, sep = ",")
```

```
asAbove <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/STAT 5790/PS/PS2/above.csv"
```

Q3:

We are given two 83×108 slices of the (same) human brain. The first slice, contained in anat.dat, a file available on Canvas, contains the anatomic information of the brain, revealing its structure, as per a Magnetic Resonance Image (MRI), while the second file, located at activ.dat also available on Canvas, is the probability that that particular location (pixel) in the slice is activated in response to the tapping of a finger. The objective of this exercise is to display the anatomic structure of the brain, overlaid with the functional image to provide context and understanding of which structures of the brain slice respond to the application of a stimulus (tapping of one's right index finger against the thumb). To do so, you may use demo(image) to get ideas, or the help file of the following functions: scan, matrix, contour, image. The function rev may also perhaps be needed.

(A)

Q: Read in the data from each of the two files into R and store each dataset as a matrix of appropriate dimensions. Note that both files have NA's to represent where there is no structure in the brain slice (outside the brain). [8 points]

A:

```
# two 83×108 slices -> nrow = 83, ncol = 108
# firstSlice <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/STAT 5790/PS/PS2/anat.
#   as.matrix(nrows = 83, ncols = 108)
anat <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/STAT 5790/PS/PS2/anat.dat") |>
  as.matrix(nrows = 83, ncols = 108)
sum(is.na(anat))
```

```
## [1] 0
```

```
activ <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/STAT 5790/PS/PS2/activ.dat")
  as.matrix(nrows = 83, ncols = 108)
sum(is.na(activ))
```

```
## [1] 3354
```

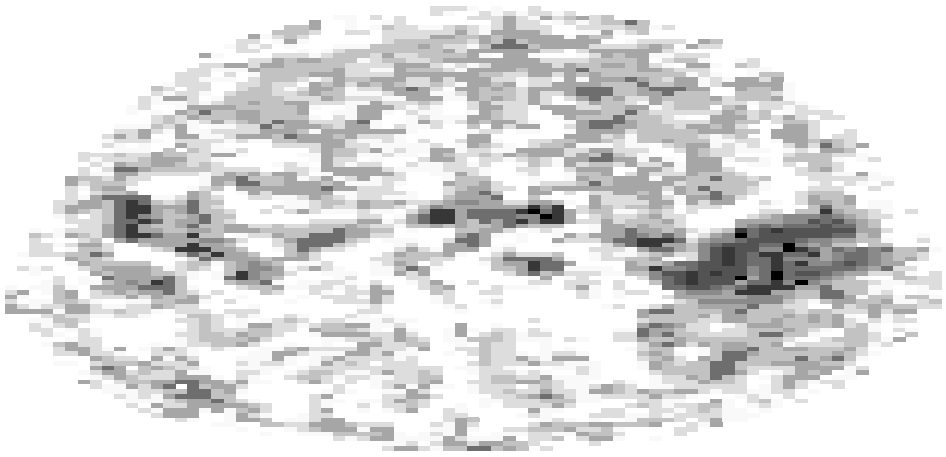
```
# secondSlice <-
```

(B)

Q: Provide an image of the activation data using a grayscale color map. Make sure that you set axes = F, and also perhaps make sure that the lighter scale of the image map represents low values. [6 points]

A:

```
image(activ, col=rev(gray(0:200/200)), axes=FALSE)
legend(grconvertX(0.5, "device"), grconvertY(1, "device"),
    c("0","0.25", ".5", "0.75", "1"), fill = rev(gray(c(0, 0.25, 0.5, 0.75, 1))), xpd = NA)
```

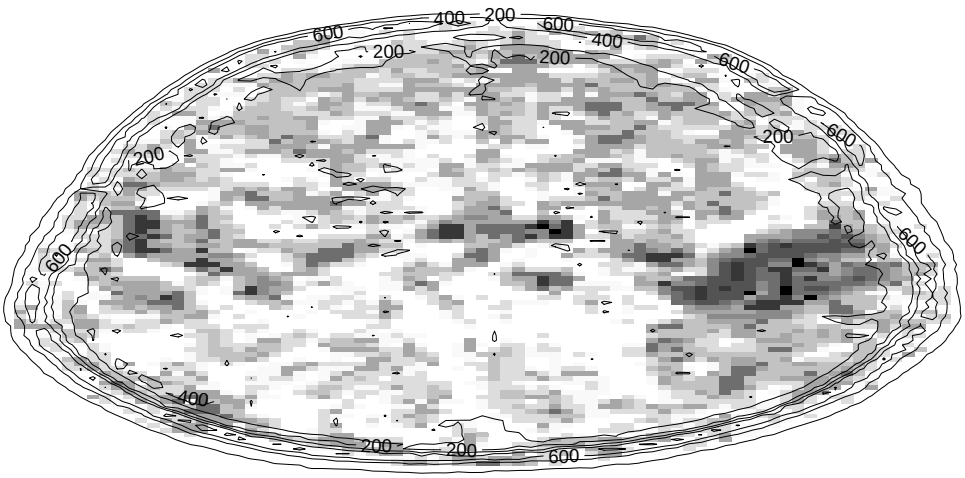


(C)

Q: On the image plot above, overlay a contour plot of the anatomic structure of the brain. Note that there is no need to label the contour lines, and also that there are a few options to control the levels of the contour plots as well as the line width. Note that overlaying means to add a figure atop an existing plot. [6 points]

A:

```
image(activ, col=rev(gray(0:200/200)), axes=FALSE)
contour(anat, nlevels = 5, add = TRUE, lwd = 0.5)
```



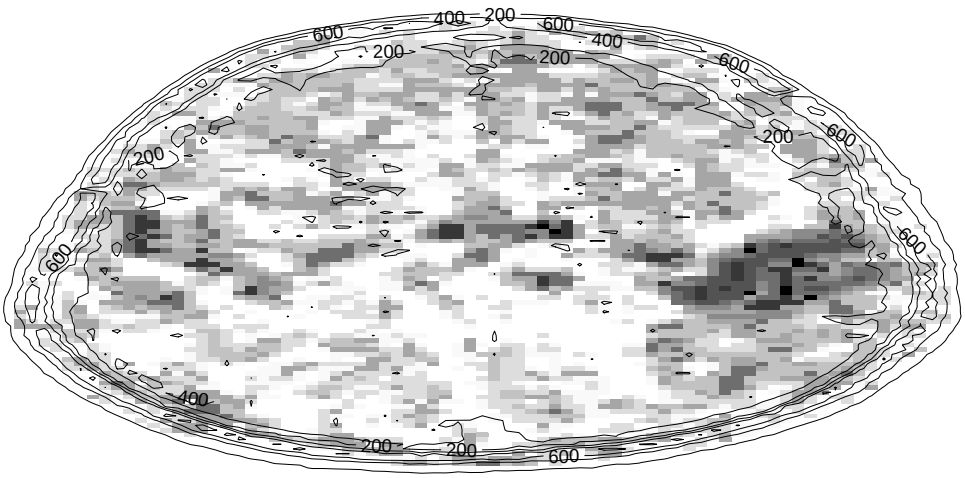
(D)

Q: What to turn in: Turn in the final plots and also the final R codes that you used for the problem.

A:

```
# Read Data
anat <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/STAT 5790/PS/PS2/anat.dat") |>
  as.matrix(nrows = 83, ncols = 108)
activ <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/STAT 5790/PS/PS2/activ.dat")
  as.matrix(nrows = 83, ncols = 108)

# Initial Imaging
image(activ, col=rev(gray(0:200/200)), axes=FALSE)
legend(grconvertX(0.5, "device"), grconvertY(1, "device"),
    c("0","0.25", ".5", "0.75", "1"), fill = rev(gray(c(0, 0.25, 0.5, 0.75, 1))), xpd = NA)
# Add Contour
contour(anat, nlevels = 5, add = TRUE, lwd = 0.5)
```


200
400
600
200
400
600
200
600
200
600
600
400
200
200
600

Q4:

Romano-British Pottery. Samples from Romano-British pottery were taken at four sites in the United Kingdom. The dataset is available on Canvas at pottery.dat. Twenty-six samples of Romano-British pottery were found at four different kiln sites in Wales, Gwent and the New Forest. The sites are Llanederyn (L), Caldicot (C), Island Thorns (I), and Ashley Rails (A). The other variables are the percentage of oxides of various metals measured by atomic absorption spectro-photometry. The data were collected to see if different sites contained pottery of different chemical compositions. A chemical analysis of the pottery was performed to measure the percentage of five metal oxides present in each sample.

1. Al: Percentage of aluminum oxide in sample
2. Fe: Percentage of iron oxide in sample
3. Mg: Percentage of magnesium oxide in sample
4. Ca: Percentage of calcium oxide in sample
5. Na: Percentage of sodium oxide in sample
6. Site: Site where pottery sample was collected

(A)

Q: Read in the dataset into R. [2 points]

A:

```
pottery <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/STAT 5790/PS/PS2/pottery.da
```

(B)

Q: For each site, provide summaries of chemical content using means, medians, standard deviations and interquartile ranges (if applicable). [8 points]

A:

```
# Sites
# "A", "C", "I", "L"

require(dplyr)
potteryStatsSiteA <- pottery |>
  filter(Site == "A") |>
  summarize(meanAl = mean(Al),
            medianAl = median(Al),
            stdDevAl = sd(Al),
            iqrAl = IQR(Al),

            meanFe = mean(Fe),
            medianFe = median(Fe),
            stdDevFe = sd(Fe),
            iqrFe = IQR(Fe),

            meanMg = mean(Mg),
            medianMg = median(Mg),
            stdDevMg = sd(Mg),
```

```
            iqrMg = IQR(Mg),

            meanCa = mean(Ca),
            medianCa = median(Ca),
            stdDevCa = sd(Ca),
            iqrCa = IQR(Ca),

            meanNa = mean(Na),
            medianNa = median(Na),
            stdDevNa = sd(Na),
            iqrNa = IQR(Na)
            )

potteryStatsSiteC <- pottery |>
  filter(Site == "C") |>
  summarize(meanAl = mean(Al),
            medianAl = median(Al),
            stdDevAl = sd(Al),
            iqrAl = IQR(Al),

            meanFe = mean(Fe),
            medianFe = median(Fe),
            stdDevFe = sd(Fe),
            iqrFe = IQR(Fe),

            meanMg = mean(Mg),
            medianMg = median(Mg),
            stdDevMg = sd(Mg),
            iqrMg = IQR(Mg),

            meanCa = mean(Ca),
            medianCa = median(Ca),
            stdDevCa = sd(Ca),
            iqrCa = IQR(Ca),

            meanNa = mean(Na),
            medianNa = median(Na),
            stdDevNa = sd(Na),
            iqrNa = IQR(Na)
            )

potteryStatsSiteI <- pottery |>
  filter(Site == "I") |>
  summarize(meanAl = mean(Al),
            medianAl = median(Al),
            stdDevAl = sd(Al),
            iqrAl = IQR(Al),

            meanFe = mean(Fe),
            medianFe = median(Fe),
            stdDevFe = sd(Fe),
            iqrFe = IQR(Fe),
```

```
            meanMg = mean(Mg),
            medianMg = median(Mg),
            stdDevMg = sd(Mg),
            iqrMg = IQR(Mg),

            meanCa = mean(Ca),
            medianCa = median(Ca),
            stdDevCa = sd(Ca),
            iqrCa = IQR(Ca),

            meanNa = mean(Na),
            medianNa = median(Na),
            stdDevNa = sd(Na),
            iqrNa = IQR(Na)
            )

potteryStatsSiteL <- pottery |>
  filter(Site == "L") |>
  summarize(meanAl = mean(Al),
            medianAl = median(Al),
            stdDevAl = sd(Al),
            iqrAl = IQR(Al),

            meanFe = mean(Fe),
            medianFe = median(Fe),
            stdDevFe = sd(Fe),
            iqrFe = IQR(Fe),

            meanMg = mean(Mg),
            medianMg = median(Mg),
            stdDevMg = sd(Mg),
            iqrMg = IQR(Mg),

            meanCa = mean(Ca),
            medianCa = median(Ca),
            stdDevCa = sd(Ca),
            iqrCa = IQR(Ca),

            meanNa = mean(Na),
            medianNa = median(Na),
            stdDevNa = sd(Na),
            iqrNa = IQR(Na)
            )
```

```
SiteA <- t(round(potteryStatsSiteA, 3))
SiteC <- t(round(potteryStatsSiteC, 3))
SiteI <- t(round(potteryStatsSiteI, 3))
SiteL <- t(round(potteryStatsSiteL, 3))
```

```
cbind.data.frame(SiteA, SiteC, SiteI, SiteL)
```

```
##           SiteA  SiteC  SiteI  SiteL
## meanAl   17.320 11.700 18.180 12.564
## medianAl 17.700 11.700 18.000 12.600
## stdDevAl  1.659  0.141  1.775  1.377
## iqrAl     1.600  0.100  0.300  2.175
## meanFe    1.512  5.415  1.712  6.372
## medianFe  1.140  5.415  1.510  6.540
## stdDevFe  0.736  0.035  0.436  0.786
## iqrFe     0.520  0.025  0.380  0.818
## meanMg    0.606  3.855  0.674  4.826
## medianMg  0.600  3.855  0.670  4.485
## stdDevMg  0.063  0.120  0.032  1.088
## iqrMg     0.110  0.085  0.010  1.588
## meanCa    0.052  0.295  0.026  0.202
## medianCa  0.060  0.295  0.010  0.200
## stdDevCa  0.034  0.007  0.026  0.058
## iqrCa     0.030  0.005  0.020  0.057
## meanNa    0.048  0.050  0.054  0.251
## medianNa  0.050  0.050  0.040  0.210
## stdDevNa  0.011  0.014  0.028  0.123
## iqrNa     0.000  0.010  0.020  0.052
```

(C)

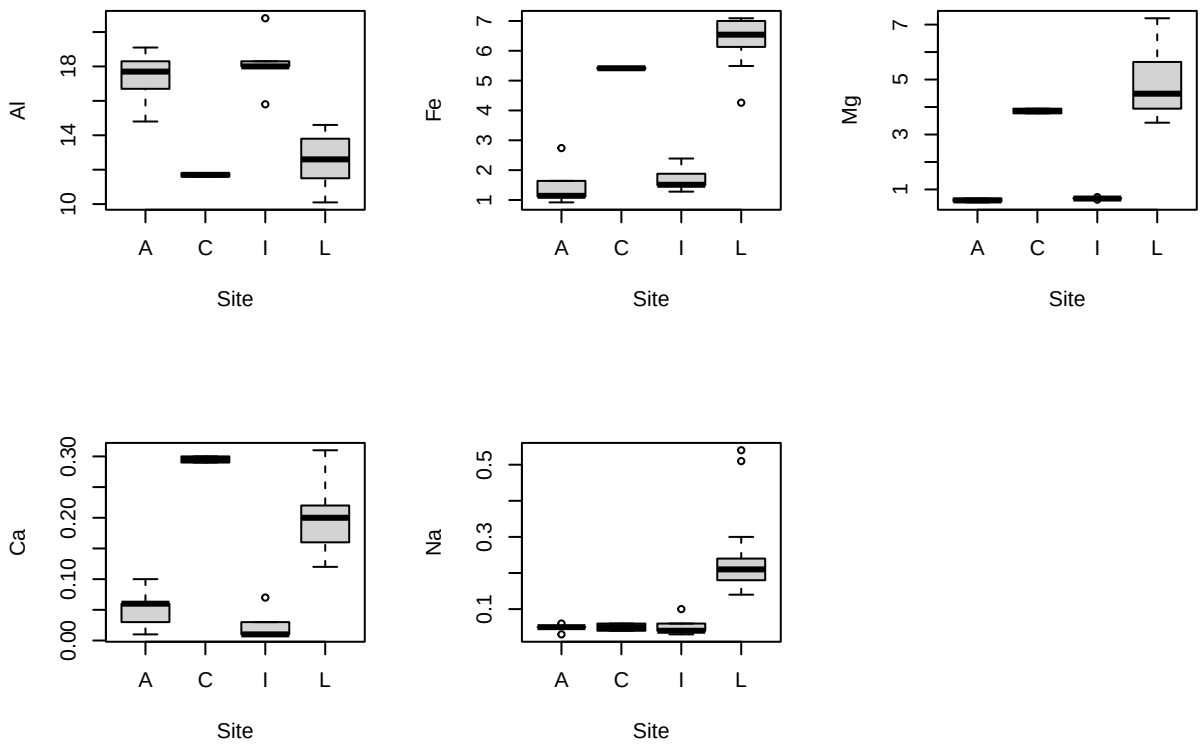
> Q: Use the function boxplot to provide boxplot displays of the site-wise composition of each chemical. Hand in these plots, all on one page. (To do this, use the graphical function par before plotting. The function:

par(mfrow=c(2,3))

divides the plotting region into 2 rows and three columns. Thus, successive calls to a plot would fill the region one by one. [10 points]

A:

```
par(mfrow=c(2,3))
boxplot(Al ~ Site, data = pottery)
boxplot(Fe ~ Site, data = pottery)
boxplot(Mg ~ Site, data = pottery)
boxplot(Ca ~ Site, data = pottery)
boxplot(Na ~ Site, data = pottery)
```



(D)

Q: Discuss the characteristics of the distributions of the percentages of different metal oxides, as provided by the above boxplots and summary measures. [2 points]

A:

Overall comparisons: For three of the five elements, Site L has the highest concentrations of Fe, Mg, and Na. Typically, sites tend to have roughly equal mean and medians for a particular element. Typically sites C and I (and on occasion A) have a small range in their IQR, especially compared to Site L. Site L tends to contain the largest range of observed values (based on IQR comparisons across elements).

Al: Site I has the highest average and median for Al, though it also has the largest standard deviation.

Fe: Site L has the highest average and median for Fe, as well as the largest standard deviation.

Mg: Site L has the highest average and median for Mg, and a standard deviation more than twice that of other sites for this particular element.

Ca: Site A has the highest mean and median for Ca, but has significantly less (less than half the second lowest) standard deviation than the other sites for this particular element.

Na: Site L has the highest average and median for Na, as well as the largest standard deviation in Na values compared to the other sites.