Homework 2

Steve Harms

September 10, 2017

Exercise 1

a)

```
#I manually entered the numeric heights without units, and generate the year as a d
ecreasing sequence

year <- seq(from = 2008, to = 1948, by = -4)
winner <- rev(c(175, 179, 179, 183, 193, 182, 182, 177, 185, 185, 188, 189, 189, 18
2, 182, 185))
opponent <- rev(c(173, 178, 178, 182, 180, 180, 185, 183, 177, 180, 173, 188, 187,
185, 193, 175))

#Combine my 3 columns into a data frame
elections <- data.frame(year, winner, opponent)
```

b)

```
#Create a new data frame with an additional field
electionsdiff <- data.frame(year, winner, opponent, difference = winner - opponent)
```

c)

```
#add a column of logical values to our data frame from (a)
elections$taller.won <- (winner - opponent > 0)
```

d)

```
#Create a frequency table of the logical values in the new column. I use prop.table
to get the proportions
percentages <- prop.table(table(elections$taller.won))
percentages
```

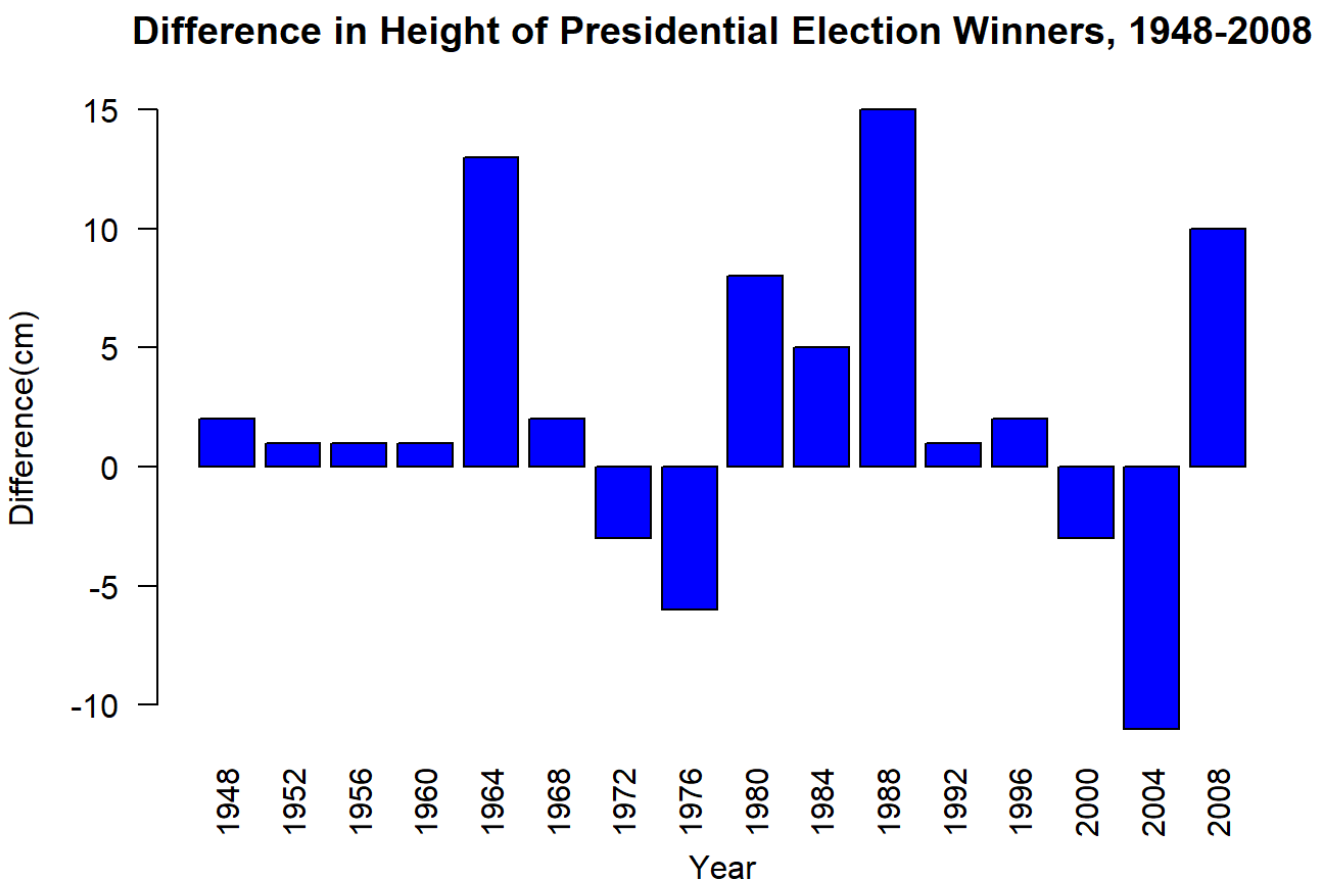
```
##
## FALSE  TRUE
##  0.25  0.75
```

75% of our elections were won by the taller candidate, so based on this table we might believe that a taller candidate is more likely to become president. However, given small sample size and the context, I wouldn't believe this to be actually true unless I was controlling for many other factors.

e)

e)

```
#Create a bar plot of height differences. X-axis labeled with year, and I color it
blue. rev() reverses the order of the data
barplot(rev(electionsdiff$difference), names.arg= rev(elections$year), axes = TRUE,
        xlab = "Year", ylab = "Difference(cm)", main = "Difference in Height of Pr
esidential Election Winners, 1948-2008",
        col = "BLUE", cex.names = .98, las = 2)
```



f)

```
#Write a new file of our data, making sure to separate by ",". I change the column
names to be more understandable.
#The file will appear in whatever you set your working directory as.
write.table(elections, file = "heights.csv", sep = ",",
            row.names = FALSE, col.names = c("Year", "Winner Height (cm)", "Loser
Height (cm)", "Taller Won?"))
```

Exercise 2

```
#I downloaded the file into the working directory, reading from Canvas does not wor
k well
students <- read.table(file = "students.txt", header = TRUE,
                       sep = "", col.names = c("height", "shoe size", "gender", "p
opulation"))
```

a)

```
#Calculating some summary statistics for our data set
mean(students$height)
```

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

```
mean(students$shoe.size)
```

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

```
sd(students$height)
```

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

```
sd(students$shoe.size)
```

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

So our students' height has mean 169.765 cm with standard deviation 7.579 cm, and shoe size has mean 40.471 with standard deviation 2.695

b)

```
#a summary of the gender column will give us counts by gender
summary(students$gender)
```

```
## female   male
##      9      8
```

So we have 9 females and 8 males

c)

```
attach(students)

#Recode to colors, I convert to character type first
population <- as.character(population)
population[population == "kuopio"] <- "blue"
population[population == "tampere"] <- "red"

#just as an example we could also do the recoding in one command:
population2 <- ifelse(students$population == "kuopio", "blue", "red")

#Combine data into a new data frame
students_new <- data.frame(height, shoe.size, gender, population)
```

d)

```
#Subset the data
male <- subset(students, gender == "male")
female <- subset(students, gender == "female")

#Write the subsets into text files, removing the quotes from character strings
write.table(male, file = "male.txt", row.names = FALSE, col.names = TRUE, quote =
FALSE)
write.table(female, file = "female.txt", row.names = FALSE, col.names = TRUE, quot
e = FALSE)
```

e)

```
#Subset the data based on the median. Problem didn't specify what to do with median
so I just leave it out of the subsets
med <- median(height)
below <- subset(students, height < med)
abovem <- subset(students, height > med)

#Write the subsets into 2 new files
write.table(abovem, file = "abovem.csv", row.names = FALSE, col.names = TRUE, quot
e = FALSE, sep = ",")
write.table(below, file = "below.csv", row.names = FALSE, col.names = TRUE, quote
= FALSE, sep = ",")
```

Exercise 3

a)

```
#Read in the data. I keep column names given in the file for now
cars <- read.table(file = "http://maitra.public.iastate.edu/stat579/datasets/cars.d
at", header = TRUE)
```

b)

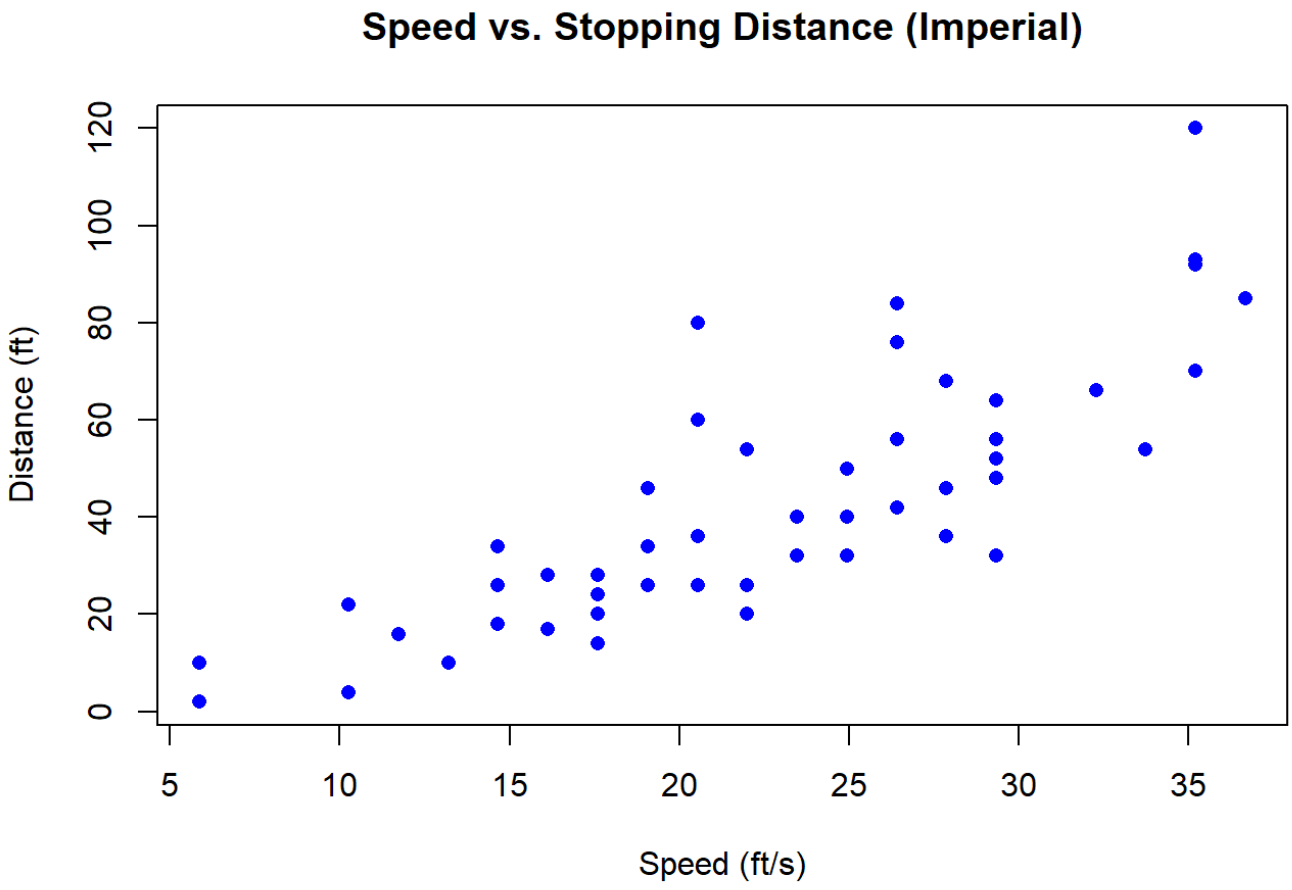
```
#attach the data frame
attach(cars)
```

c)

```
#convert to feet per second
fps <- speed * 5280 / (60*60)
```

d)

```
#Plot speed vs. distance
plot(y = dist, x = fps, type = "p", ylab = "Distance (ft)", xlab = "Speed (ft/s)",
main = "Speed vs. Stopping Distance (Imperial)",
     pch = 16, col = "blue")
```




```
#Save the plot for later
plot1 <- recordPlot()

#Save to pdf file
dev.copy2pdf(file = "Speed vs Distance fps.pdf", device = plot1)
```

```
## png
##   2
```

e)

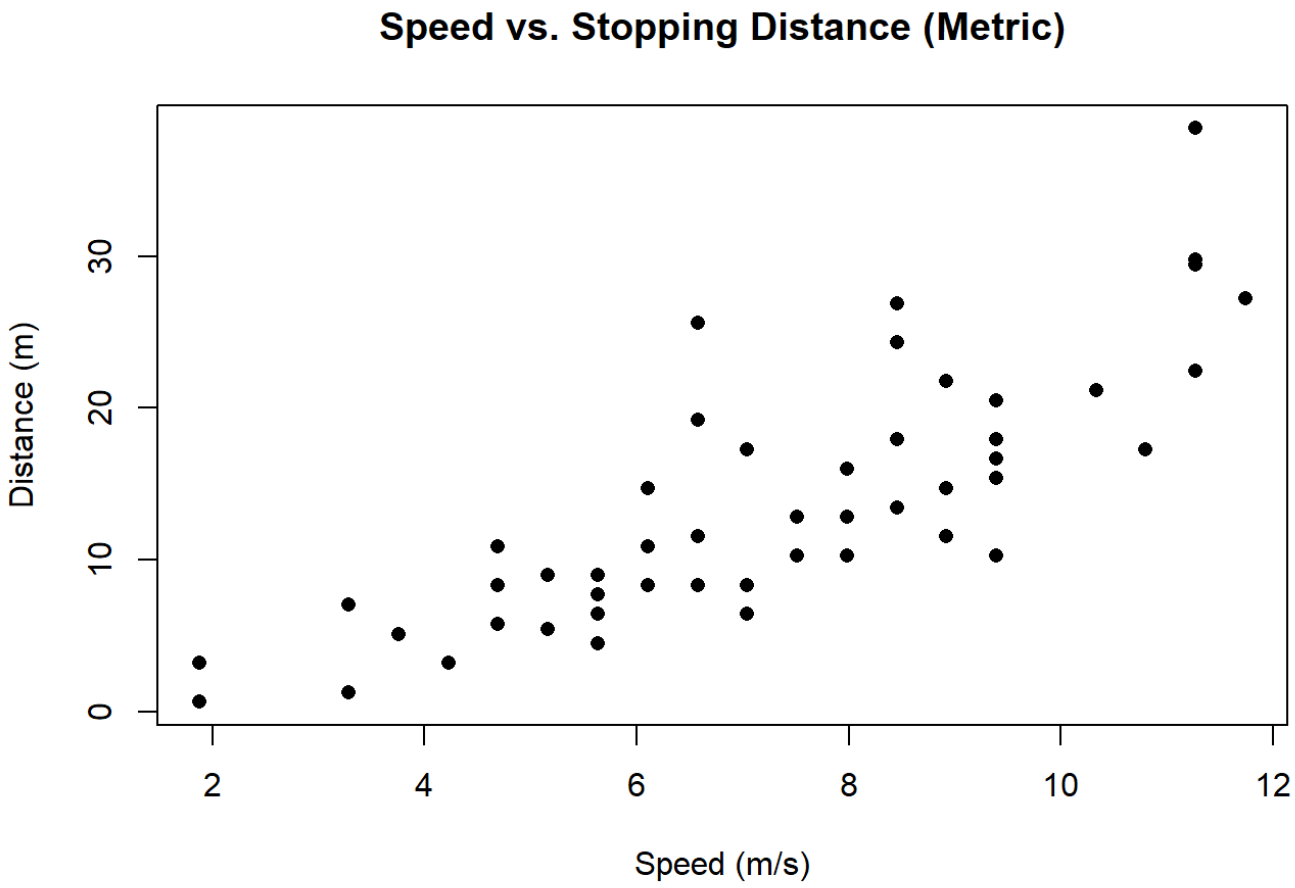
```
#Convert speeds and distance into metric
#Distance in meters
distance.metric <- dist*(1.6903*1000)/5280

#Speed in km/hr
speed.metric <- speed*1.6903

#Speed in meters/second
speed.ms <- fps*1.6903*1000/5280
```

f)

```
#Detach data frame
detach(cars)
```

g)

```
#Plot Speed against distance again, now in metric
plot(y = distance.metric, x = speed.ms, type = "p", ylab = "Distance (m)", xlab =
"Speed (m/s)", main = "Speed vs. Stopping Distance (Metric)",
     pch = 16, col = "black")
```




```
plot2 <- recordPlot()
```

h)

```
#Print to pdf
dev.copy2pdf(file = "Speed vs Distance metric.pdf", device = plot2)
```

```
## png
##   2
```

(Note I saved the first plot when I created it in a previous step)

It's clear from the two plots that there is a positive linear relationship between the speed of the car and the distance required to stop. THe only difference between the the two plots is the units, the trend is the exact same.

Exercise 4)

a)

```
#Read the data into R
pottery <- read.table(file = "http://www.public.iastate.edu/~maitra/stat579/datasets/pottery.dat", header = TRUE)
```

b)

Note that in the output below, each row provides the summary statistics for each element at each site (4 sites total = 4 rows). Site C only had 2 observations so the IQR isn't really applicable for that data. The aggregate function doesn't give standard deviation so I call it separately. This could also be done with the tapply() function if we wanted to do so.

```
#I use the base aggregate() function to find summary statistics by site,
#much faster than subsetting and then finding summary statistics for each
#Note that summary won't give us standard devitaion, so I call that function as wel
l

aggregate(pottery[,-6], by = list(pottery$Site), FUN = summary)
```

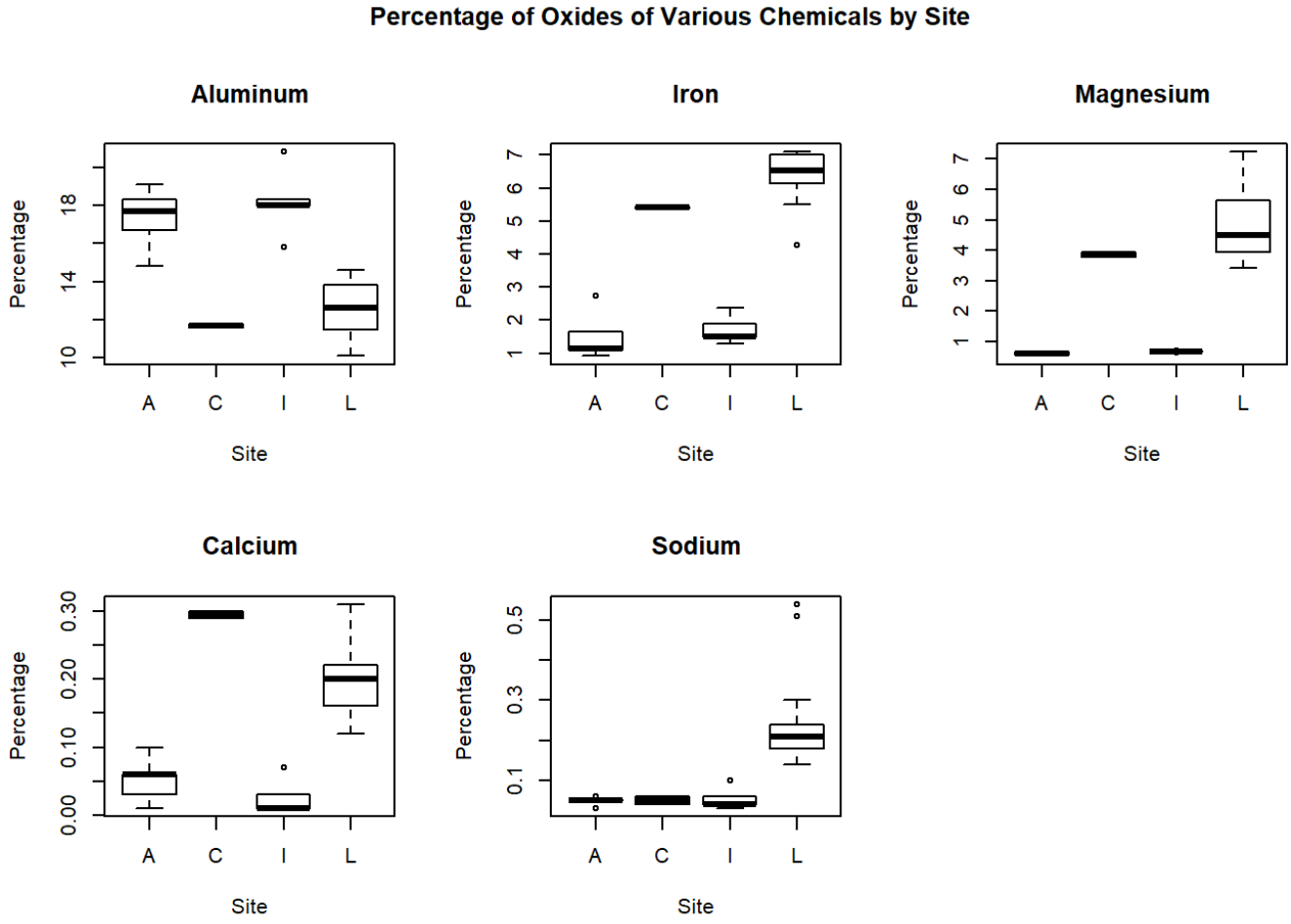
```
##   Group.1 Al.Min. Al.1st Qu. Al.Median Al.Mean Al.3rd Qu. Al.Max. Fe.Min.
## 1       A   14.80      16.70     17.70   17.32      18.30   19.10   0.920
## 2       C   11.60      11.65     11.70   11.70      11.75   11.80   5.390
## 3       I   15.80      18.00     18.00   18.18      18.30   20.80   1.280
## 4       L   10.10      11.52     12.60   12.56      13.70   14.60   4.260
##   Fe.1st Qu. Fe.Median Fe.Mean Fe.3rd Qu. Fe.Max. Mg.Min. Mg.1st Qu.
## 1      1.120     1.140   1.512      1.640   2.740   0.530      0.560
## 2      5.402     5.415   5.415      5.428   5.440   3.770      3.812
## 3      1.500     1.510   1.712      1.880   2.390   0.630      0.670
## 4      6.162     6.540   6.372      6.980   7.090   3.430      4.020
##   Mg.Median Mg.Mean Mg.3rd Qu. Mg.Max. Ca.Min. Ca.1st Qu. Ca.Median
## 1     0.600   0.606      0.670   0.670  0.0100     0.0300    0.0600
## 2     3.855   3.855      3.898   3.940  0.2900     0.2925    0.2950
## 3     0.670   0.674      0.680   0.720  0.0100     0.0100    0.0100
## 4     4.485   4.826      5.608   7.230  0.1200     0.1625    0.2000
##   Ca.Mean Ca.3rd Qu. Ca.Max. Na.Min. Na.1st Qu. Na.Median Na.Mean
## 1  0.0520     0.0600  0.1000  0.0300     0.0500    0.0500  0.0480
## 2  0.2950     0.2975  0.3000  0.0400     0.0450    0.0500  0.0500
## 3  0.0260     0.0300  0.0700  0.0300     0.0400    0.0400  0.0540
## 4  0.2021     0.2200  0.3100  0.1400     0.1850    0.2100  0.2507
##   Na.3rd Qu. Na.Max.
## 1     0.0500  0.0600
## 2     0.0550  0.0600
## 3     0.0600  0.1000
## 4     0.2375  0.5400
```

```r
#Standard deviations below
aggregate(pottery[,-6], by = list(pottery$Site), FUN = sd)
```

```
##   Group.1        Al         Fe         Mg          Ca         Na
## 1       A 1.6589153 0.73601630 0.06348228 0.034205263 0.01095445
## 2       C 0.1414214 0.03535534 0.12020815 0.007071068 0.01414214
## 3       I 1.7753873 0.43597018 0.03209361 0.026076810 0.02792848
## 4       L 1.3770689 0.78556377 1.08822090 0.058201054 0.12262916
```

c)

```r
#Build boxplots for each chemical, subset by site
#First, set up the plotting area to include enough space for all of the plots and a title
par(mfrow = c(2,3), oma=c(0,0,2,0))

#Now, call the boxplot function for each chemical
boxplot(pottery$Al~pottery$Site, main = "Aluminum", xlab = "Site", ylab = "Percentage")
boxplot(pottery$Fe~pottery$Site, main = "Iron", xlab = "Site", ylab = "Percentage")
boxplot(pottery$Mg~pottery$Site, main = "Magnesium", xlab = "Site", ylab = "Percentage")
boxplot(pottery$Ca~pottery$Site, main = "Calcium", xlab = "Site", ylab = "Percentage")
boxplot(pottery$Na~pottery$Site, main = "Sodium", xlab = "Site", ylab = "Percentage")

#And a title for good measure
title("Percentage of Oxides of Various Chemicals by Site", outer = TRUE)
```



d)

Based on the box plots, it looks like Fe, Ca, and Mg have an inverse relationship with Al, in that sites where Al was found in high amounts meant Mg, Ca, and Fe were found in small amounts (and vice versa). Sodium was found in very low percentage amounts in 3 of the 4 sites, while Aluminum was found in somewhat high percentages at all 4 sites. For the other 3 chemicals, the percentage varied by site. I think this means that sites where Fe, Mg, and Ca had higher concentrations, it was because they were being used as substitutes for Aluminum. It also appears that we have a few outliers and that those outliers were fairly consistent across sites.