Datamining 101 HW1 - Group J

Problem 1.A

First column is quantitative; the second is qualitative.

```
str(read.csv('data/myfirstdata.csv'))
'data.frame':     1999 obs. of  2 variables:
 $ X0  : int  0 0 1 0 1 0 1 0 1 3 ...
 $ X0.1: Factor w/ 22 levels "0","1","10","11",..: 14 2 11 1 11 2 2 1 1
11 ...
```

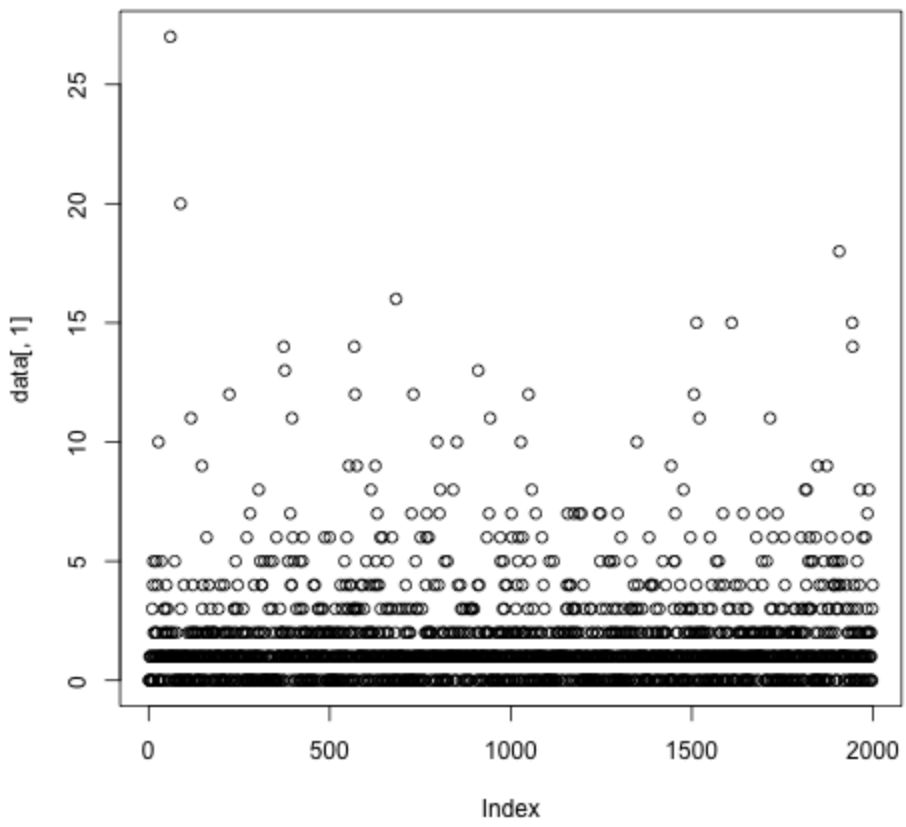
Problem 1.B

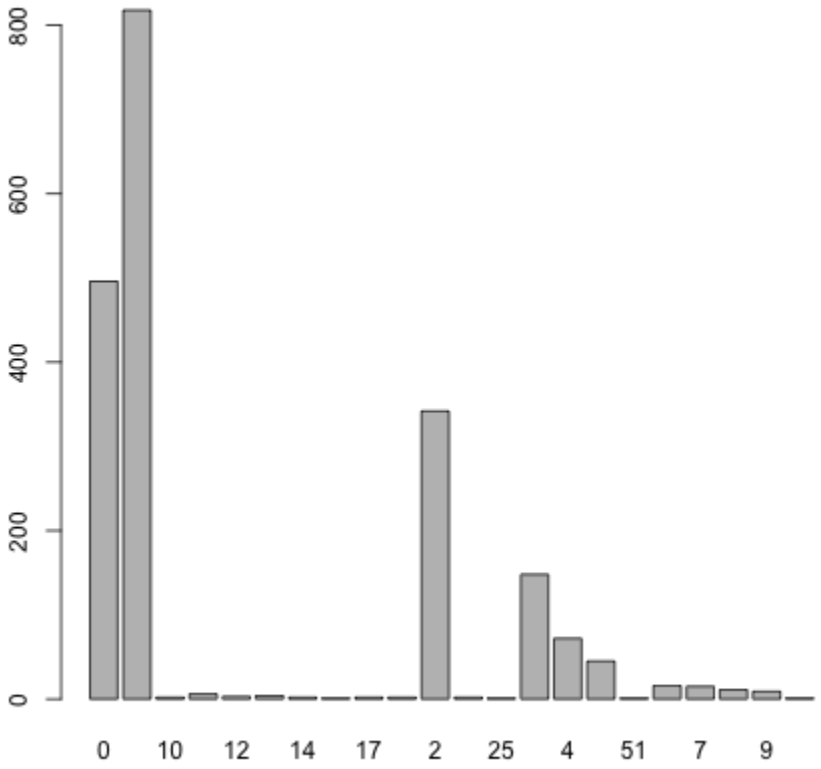
The "R Data Import/Export" manual section 2.1.9 titled "Classes for the variables" states that R attempts to convert data to the following types in a decending order of preference: "logical, integer, numeric and complex, moving on if any entry is not missing and cannot be converted. If all of these fail, the variable is converted to a factor." On line 1463 of the input file "myfirstdata.csv" the second column contains the value "two" which cannot be converted to logical, integer, numeric or complex and as a result the values for the entire column are imported as "factor".

Problem 1.C

```
data <- read.csv('data/myfirstdata.csv')
png("hw_1_problem_1_c_col_1.png")
plot(data[,1])
dev.off()

png("hw_1_problem_1_c_col_2.png")
plot(data[,2])
dev.off()
```


data[, 1]
0
5
10
15
20
25
0
500
1000
1500
2000
Index



Problem 1.D

#VALUE!

Problem 2.A

```
sample(read.csv('data/twomillion.csv', header=FALSE)[,1], 10000,
replace=TRUE)
```

Problem 2.B

```
data = sample(read.csv('data/twomillion.csv', header=FALSE)[,1], 10000,
replace=TRUE)
mean(data):  9.422355
max(data):  17.03436
min(data):  2.268879
var(data):  3.961212
quantile(data, .25):  8.100489
```

Problem 2.C

```
data = read.csv('data/twomillion.csv', header=FALSE)[,1]
mean(data):  9.45103
max(data):  18.96657
min(data):  -0.1115070
var(data):  4.001492
quantile(data, .25):  8.103118
```

Problem 2.D

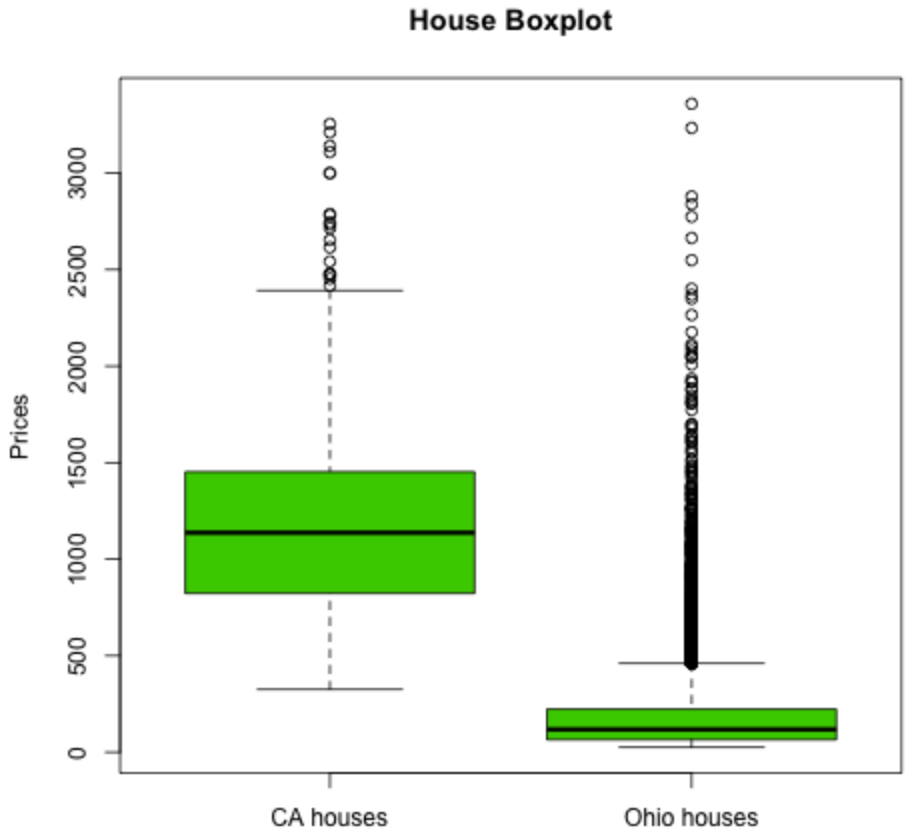
```
write.csv(data, 'data/twomillion-sample.csv')
=AVERAGE(B:B):  9.422354942
=MAX(B:B):  17.0343598
=MIN(B:B):  2.268878857
=VAR(B:B):  3.961211682
=QUARTILE(B:B,0.25):  2.268878857
```

Problem 2.E

```
It will only fit 1048576 rows.
```

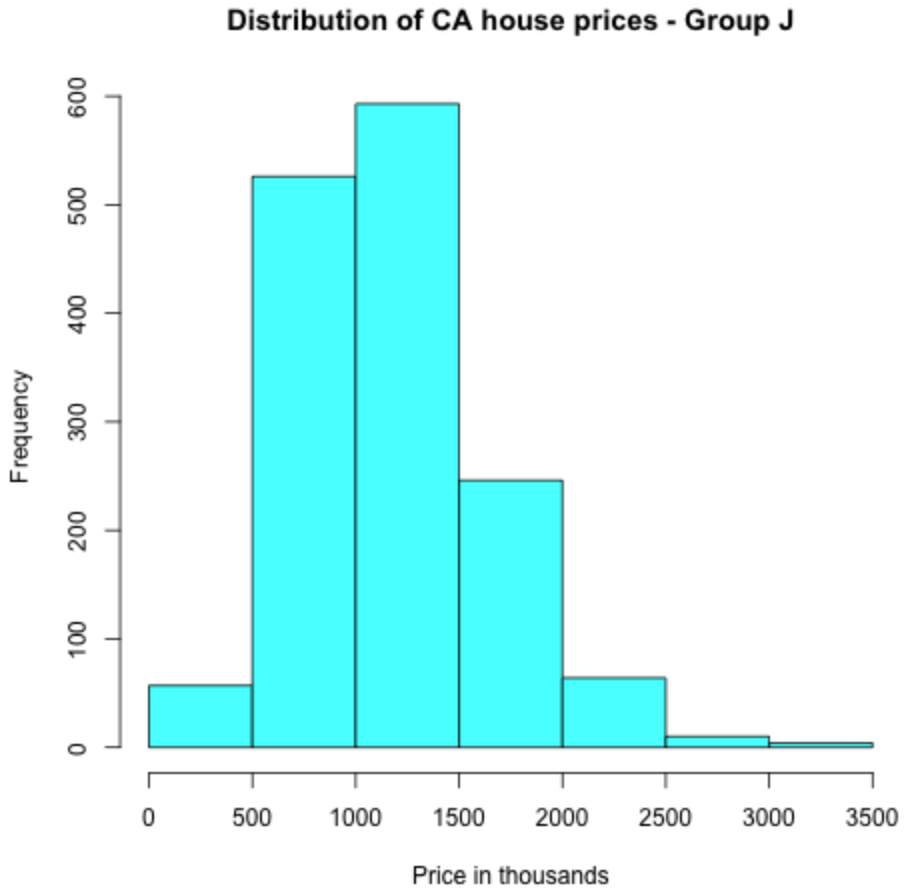
Problem 3.A

```
dataOH = read.csv('data/OH_house_prices.txt', header=FALSE)
dataCA = read.csv('data/CA_house_prices.txt', header=FALSE)
png("hw_1_problem_3_a.png")
boxplot(dataCA[,1],dataOH[,1], main=" House Boxplot ", names=c("CA houses",
      "Ohio houses"), ylab="Prices", col=3)
dev.off()
```



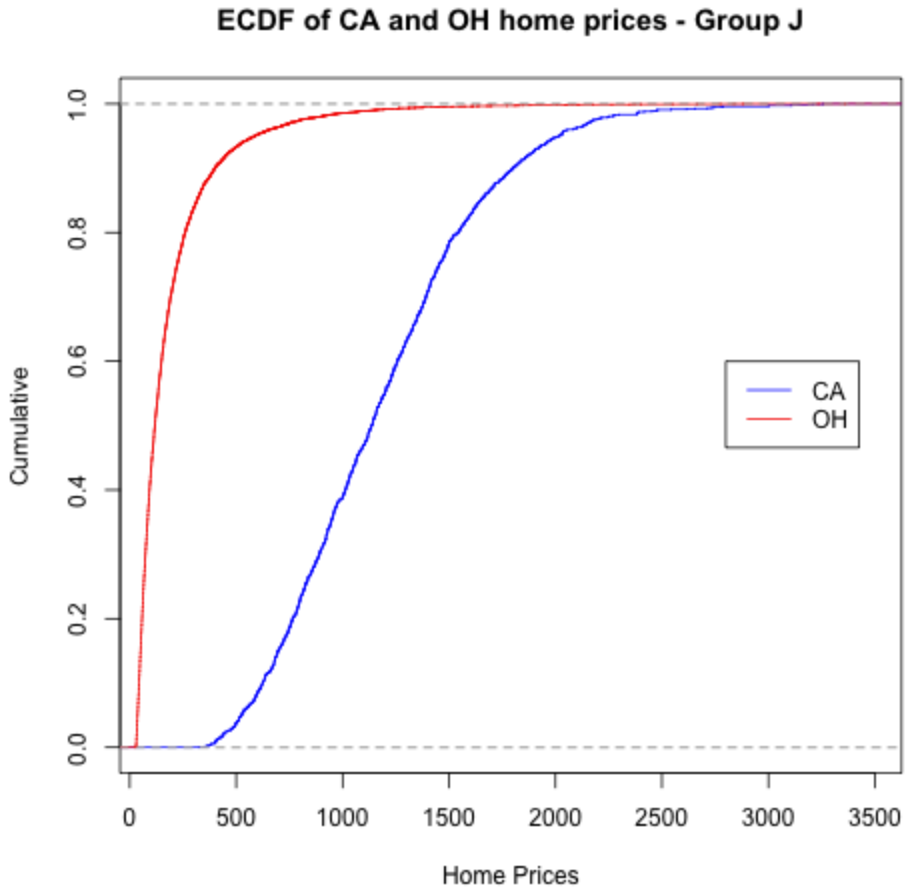
Problem 3.B

```
dataCA = read.csv('data/CA_house_prices.txt', header=FALSE)
png("hw_1_problem_3_b.png")
hist(dataCA[,1], breaks=c(0, 500, 1000, 1500, 2000, 2500, 3000, 3500),
xlab="Price in thousands", main="Distribution of CA house prices - Group J",
col=5)
dev.off()
```



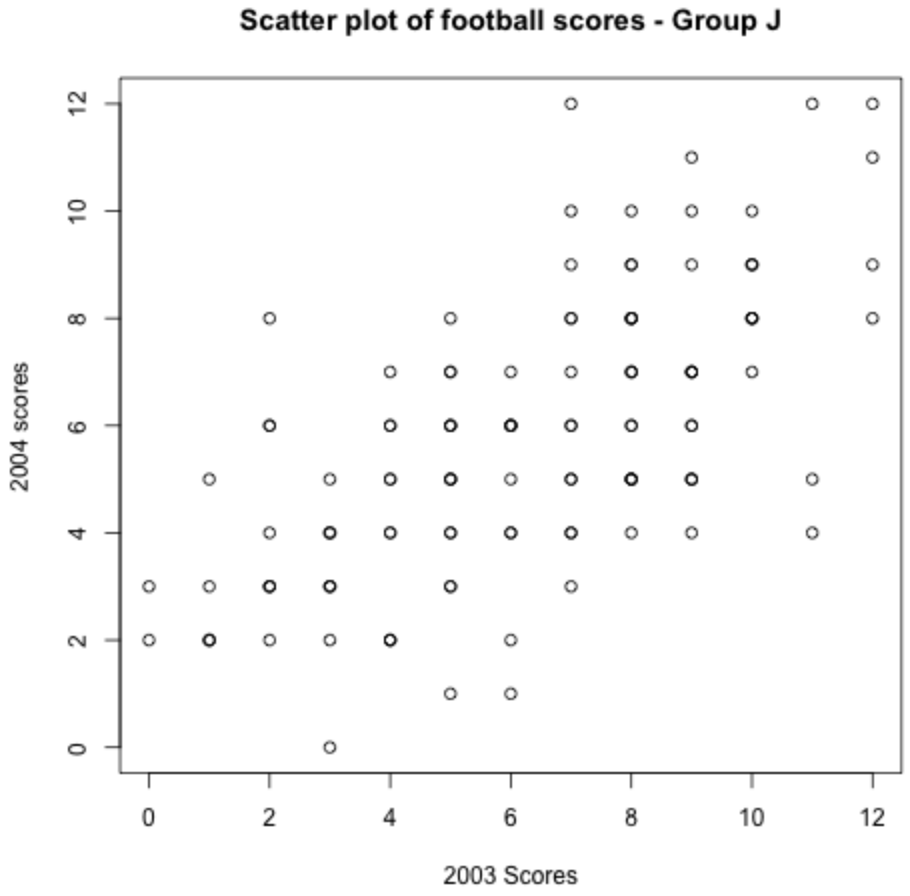
Problem 3.C

```
png("hw_1_problem_3_c.png")
plot(ecdf(dataCA[,1]), verticals=TRUE, do.p=FALSE, main="ECDF of CA and OH
home prices - Group J", xlab="Home Prices", ylab="Cumulative", col="blue")
lines(ecdf(dataOH[,1]), verticals=TRUE, do.p=FALSE, col.h="red", col.v="red",
lwd=1)
legend(2800, .6, c("CA","OH"), col=c("blue","red"), lwd=c(1,1))
dev.off()
```



Problem 4.A

```
data = read.csv("data/football.txt")
png("hw_1_problem_4_a.png")
plot(data[,2], data[,3], main="Scatter plot of football scores - Group J",
xlab="2003 Scores", ylab="2004 scores")
dev.off()
```



Problem 4.B

The plot points represent discrete values. Many points in a scatter plot may be plotted over top of one another. No example was given in class for dealing with with this issue. Possible remedies include using a small offset, or multiple colors.

Problem 4.C

```
0.6537691
```

Problem 4.D

```
cor(data[,2], mapply(function(x) x + 10, data[,3]))
0.6537691
```

Problem 5.A

```
ages <- c(19,23,30,30,45,25,24,20)
sd(ages)
8.315218
```

Problem 5.B

```
ages <- c(19,23,30,30,45,25,24,20)
mean <- sum(ages) / length(ages)
sumDeviantionsSq <- sum(mapply(function(x) (x - mean)^2, ages))
sqrt(sumDeviantionsSq / (length(ages) - 1))
8.315218
```

Problem 5.C

```
sd(mapply(function(x) x + 10, c(19,23,30,30,45,25,24,20)))
8.315218 - It doesn't change
```

Problem 5.D

```
sd(mapply(function(x) x * 100, c(19,23,30,30,45,25,24,20)))
831.5218 - SD increases by 100x
```