Stat 107: Introduction to Business and Financial Statistics

Homework 1 Sketch Solutions

Homework is graded on a 1-5 scale as detailed below. As there is often interpretation and programming issues with begin/end dates, homework is graded in a holistic way, and not always correct/incorrect.

- **5- Clear and correct, contains answers to all problems and logical explanations: 95-100%**
- **4- Clear and correct but might have a few wrong answers: 90-95%**
- **3- A number of incorrect answers or missing some parts: 80-90%**
- **2- Sloppy and/or incomplete, many wrong answers: 65%**
- **Missing many problems, questions that are answered are incorrect: 50%**
- **Homework was late or not turned in: 0%**

1) For this problem we want you to reproduce Figures 5.1-5.5 in Chapter 5 of the book *R by Example*.

   The deliverable for this question is a neat and organized print out (cut and paste the graphs) of figures 5.1-5.5 from the book *R by Example.*
   <u>Full credit if 5 figures are given:</u>

Figure 5.1

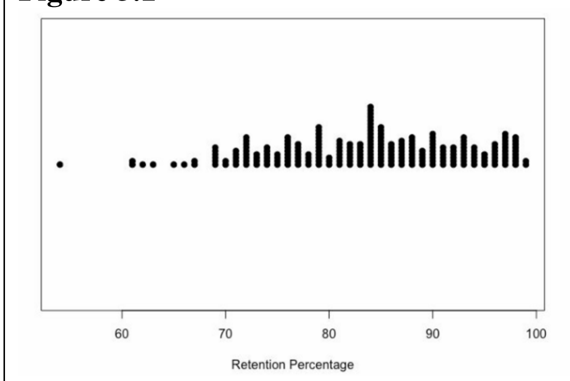


Figure 5.2

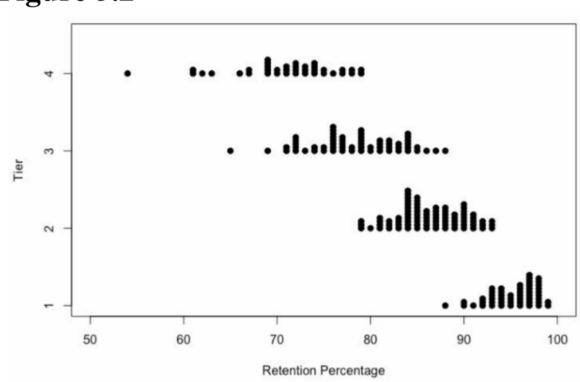


Figure 5.3

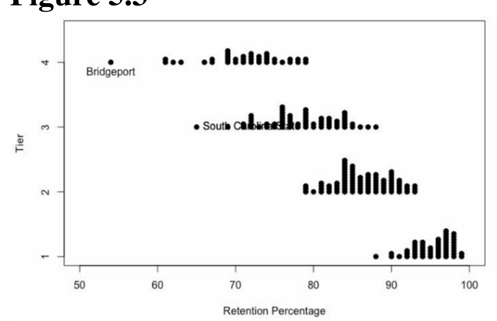


Figure 5.4

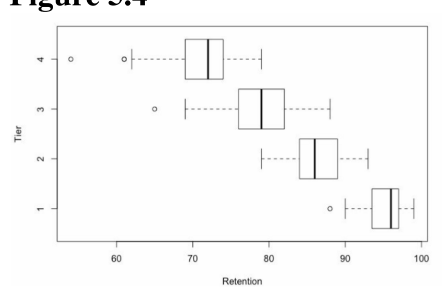
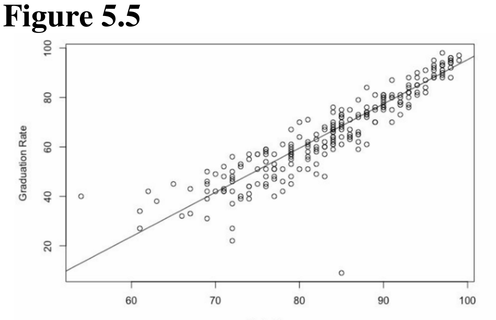


Figure 5.5



2) Read the handout on the website 'BestFirstRTutorial' and use R to do the following problems. Neatly write up your solution to each part, clearly showing how R was used and the resulting output. **Highlight** the answer to each question so it is easy to find in your output.

   a) Problems 1 and 2 on page 4
   b) Problem 1 page 5
   c) Problem 1 parts a and b page 8
   d) Problem 2 page 10 (the first problem 2 on page 10)
   e) Problem 1, page 11
   f) Problems 1 and 2, page 13-14
   g) Problem 2, page 16 [to make life easier write two functions, one to return the sum of x and another function to return the sum of x squared-you might want to do problem 3 for experience in writing simple functions then come back to this part.]

Page 4 - Problem 1

Define the variables

```
> x <- c(4, 2, 6)
> y <- c(1, 0, -1)

a) > length(x)
[1] 3
b) > sum(x)
[1] 12
c) > sum(x ^ 2)

[1] 56
d) > x + y
[1] 5 2 5
e) > x * y
[1] 4 0 -6
f) > x - 2
[1] 2 0 4
g) > x ^ 2
[1] 16 4 36
```

Page 4 - Problem 2

```
a) > 7:11
[1] 7 8 9 10 11
b) > seq(2, 9)
[1] 2 3 4 5 6 7 8 9
c) > seq(4, 10, by = 2)
[1] 4 6 8 10
d) > seq(3, 30, length = 10)
[1] 3 6 9 12 15 18 21 24 27 30
e) > seq(6, -4, by = -2)
[1] 6 4 2 0 -2 -4
```

Page 5 - Problem 1

Store the initial variable

```
> x <- c(5, 9, 2, 3, 4, 6, 7, 0, 8, 12, 2, 9)
a) > x[2]
[1] 9
b) > x[2:4]
[1] 9 2 3
c) > x[c(2, 3, 6)]
[1] 9 2 6
d) > x[c(1:5,10:12)]
[1]  5  9  2  3  4 12  2  9
e) > x[-(10:12)]
[1] 5 9 2 3 4 6 7 0 8
```

Page 8 - Problem 1

Initialize the requested matrices and verify output

```
> x = matrix(c(3, 2, -1, 1), nrow = 2, byrow = T)
> x


     [,1] [,2]
[1,]    3    2
[2,]   -1    1
> y = matrix(c(1, 4, 0, 0, 1, -1), nrow = 2, byrow = T)
> y
     [,1] [,2] [,3]
[1,]    1    4    0
[2,]    0    1   -1
a) > 2 * x
     [,1] [,2]
[1,]    6    4
[2,]   -2    2
b) > x * x
     [,1] [,2]
[1,]    9    4
[2,]    1    1
```

Page 10 - Problem 2

```
> data("mtcars")
> attach(mtcars)
> help(mtcars)
```

Mean Weight:

```
> mean(mtcars$wt)
[1] 3.21725
```

Mean Fuel Consumption:

```
> mean(mtcars$mpg)
[1] 20.09062
```

Page 11 - Problem 1

```
> dnorm(0.5, 2, sqrt(0.25))
a) [1] 0.008863697
> pnorm(2.5, 2, sqrt(0.25))
b) [1] 0.8413447
> qnorm(0.95, 2, sqrt(0.25))
c) [1] 2.822427
> pnorm(3, 2, sqrt(0.25)) - pnorm(1, 2, sqrt(0.25))
d) [1] 0.9544997
```

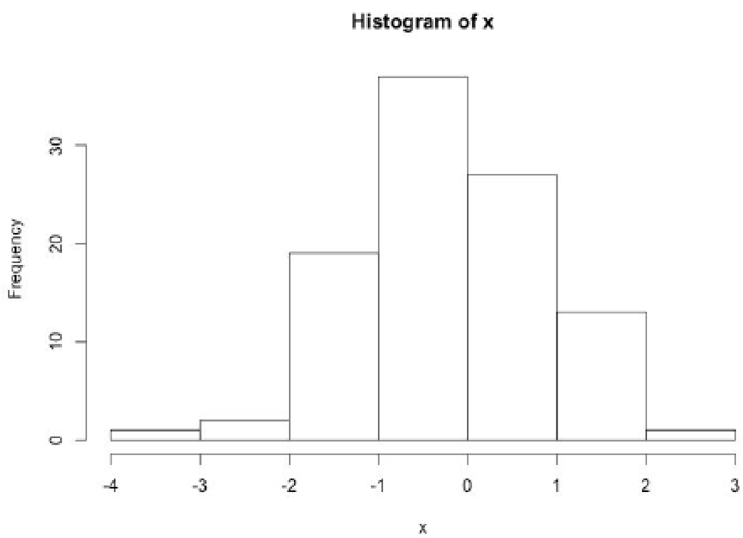
Page 13 - Problem 1

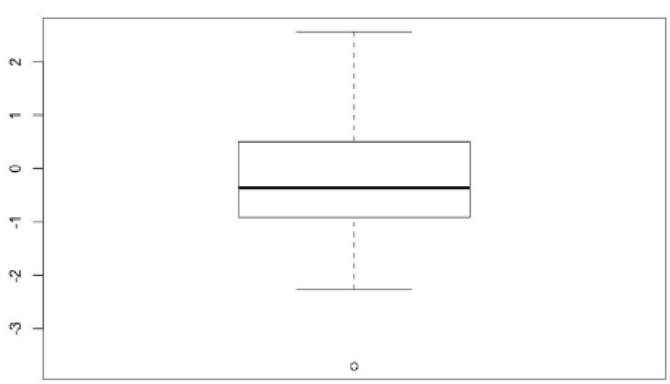
Data generated:

```
> x <- rnorm(100)
```

Histogram:

```
> hist(x)
```

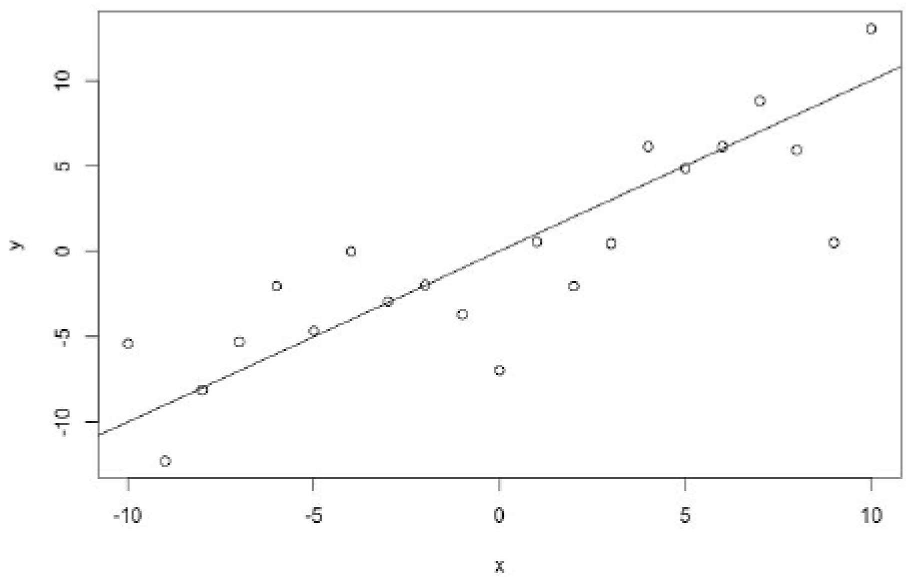




Page 14 - Problem 2

Data to generate results:

```
> x <- (-10):10
> n <- length(x)
> y <- rnorm(n, x, 4)
> plot(x, y)
> abline(0, 1)
```

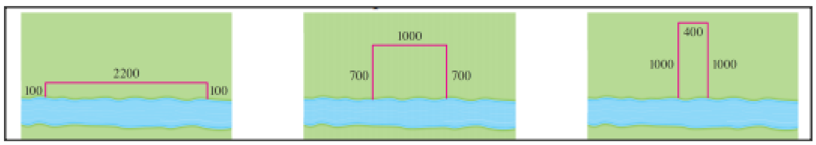


Page 16, Problem 2

```
myfun=function(x) {
print(paste("The sum of the x values is",sum(x)))
print(paste("The sum of the x*x values is",sum(x*x)))
}
```

3) We will revisit optimization after we start portfolio theory, but since it is easy enough to do in R [for the most basic problems] let's spend a little time with the idea since it is a good example of writing a function, plotting and then calling an R routine [we use the phrase R routine and R function interchangeably, but don't want you to get confused between the function you write as opposed to a built in R function]. Broadly speaking, optimization is the process of finding the minimum or maximum of an objective function.

A farmer has 2400 feet of fencing and wants to fence off a rectangular field that borders a straight river. He needs no fence along the river. What are the dimensions of the field that has the largest area?



a) Write a function in R for the area of the field as a function of x [what is x? Why only x?]

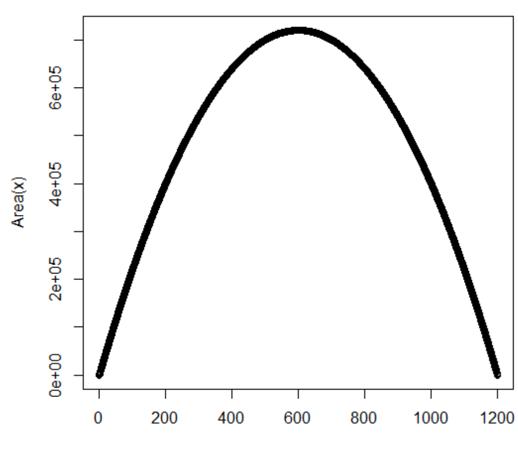
x is the length on either side (the width)

```
Area = function(x)
 return(x*(2400-2*x))
```

b) What is the range of possible values for x?
0 to 1200

c) Plot the function in part (a) .

```
> x=seq(0,1200,.1)
> plot(x,Area(x))
```



d) Find the maximum value [set maximum=TRUE in the optimize function]

```
> optimize(Area , lower=0, upper=1200, maximum=TRUE)
$maximum
[1] 600

$objective
[1] 720000
```

e) Write up your solution clearly.

Hence the optimal width is 600 feet, which gives an enclosed area of 720,000 ft2

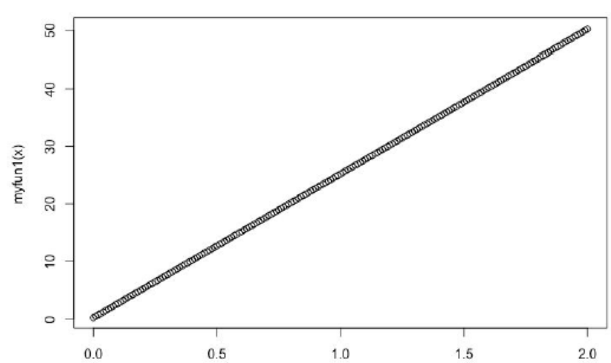
4) Remember integration from your high school calculus class? Ok, neither do I. Luckily R can do integration for us. Now I haven't used an integral in a long time so not sure how useful this is, but it's educational thing to do.

   a. Write a function in R to generate `f(x)=0.2+25x`. Call your function `myfun1`. **Deliverable is your code.**
   b. R can vectorize commands. That is it can compute the function on a vector at once. Generate a sequence in R from 0 to 2 by `x=seq(0,2,.01)`.
      i. **Deliverable.** In words, what does the `seq` command do?
      ii. **Deliverable.** Generate a plot of x versus f(x).
      iii. **Deliverable.** Calculate the area under the curve by hand.
      iv. **Deliverable.** Show the function call `integrate(myfun1,0,1)`.
      v. **Deliverable.** Do your answers to (iii) and (iv) agree?

   c. Write a function in R to generate `f(x)=1/log(x+4)`. Call your function `myfun2`.
      i. **Deliverable.** Generate a plot of x versus f(x).
      ii. **Deliverable.** Show the function call `integrate(myfun2,-1,3)`.
      iii. Go to wolframalpha.com and enter this code `integrate(1/log(x+4),-1,3)`. Does the answer agree with R?

   d. In R, execute the command `integrate(dnorm,0,Inf)`. What is the answer? What is it calculating?

a) Function for generating the line:
   > myfun1 = function(x) { return (0.2 + (25 * x)) }
b) Generating a sequence from 0 to 2 with a 0.01 step increment:
   > x <- seq(0, 2, 0.01)
   i) The seq(...) command takes as an input the start value of a range and the end value, inclusive, and generates values in between specified by the third parameter, which is the step size, in this case 0.01. Thus, the output is 0, 0.01, 0.02, 0.03, etc... until 2. This data is returned as a vector.
   ii) Plot:
   > plot(x, myfun1(x))



iii)

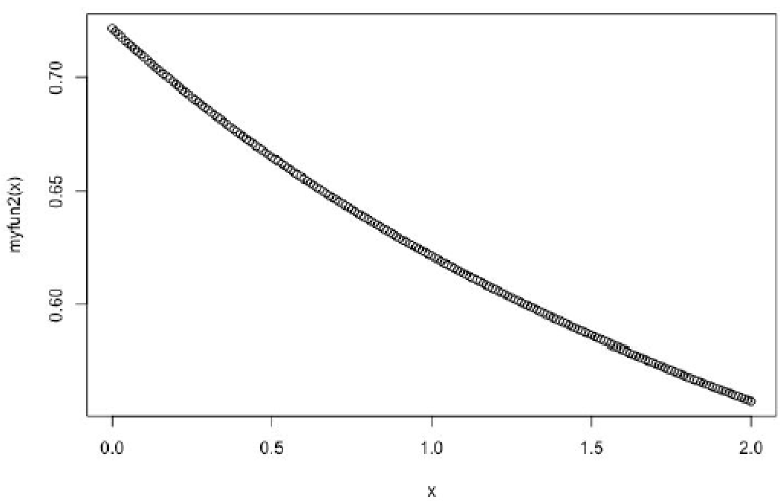
$$\int_0^2 (0.2 + 25x)\mathrm{d}x = \left[0.2x + \frac{25}{2}x^2\right]_0^2$$
$$= 50.4$$

iv) Using the builtin integrate function:
\> integrate(myfun1, 0, 2)
50.4 with absolute error < 5.6e-13

v) The answers that were computed by hand and using the builtin functions agree

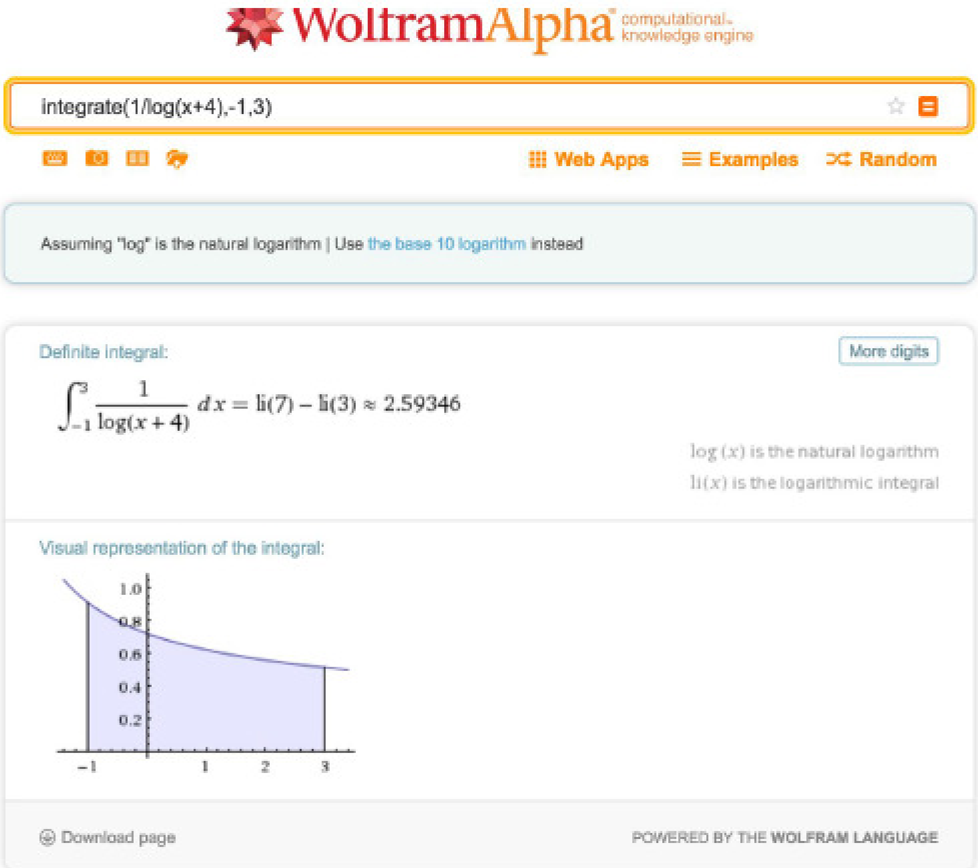
c) Function generation and plot:

i) > myfun2 <- function(x) { return (1 / log(x + 4)) }
\> plot(x, myfun2(x))



ii) Using the builtin integrate function on the range of values from -1 to 3:
\> integrate(myfun2, -1, 3)
2.593463 with absolute error < 6.9e-14

iii) The result from wolfram alpha agree with the results computed by R



d) The computation of the normal distribution integral:

```
> integrate(dnorm, 0, Inf)
0.5 with absolute error < 4.7e-05
```

This is the case because the result is that there is a 50% chance of picking a value between 0 and infinity for a standard distribution of values given that the valid range of values is -∞ to +∞. Since half of those values are between 0 and +∞ thus one has a 50% change P(0.5) of choosing a value on that side, which is what the integral in this case represents.

5) Install the quantmod package in R using the command
`install.packages("quantmod")`

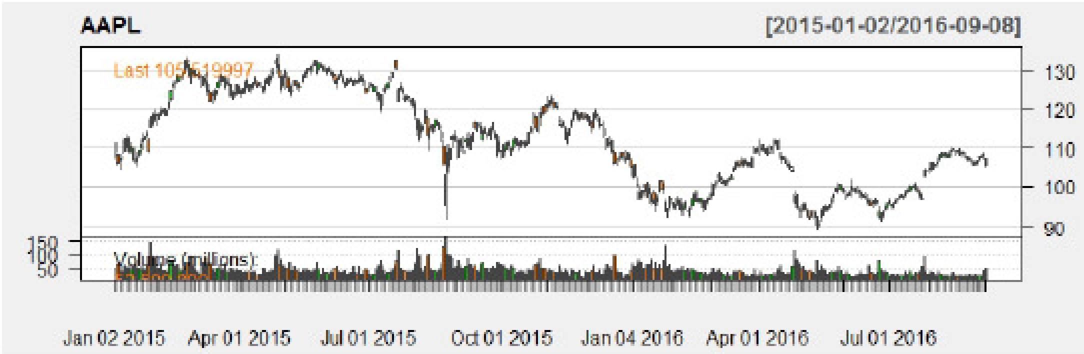
Note: we want to always work with *adjusted* closing prices (adjusted for stock splits and dividends) instead of the usual closing prices. This is done as follows:

```
getSymbols("AAPL", from = "2015-01-01")
AAPL.a =adjustOHLC(AAPL)
```

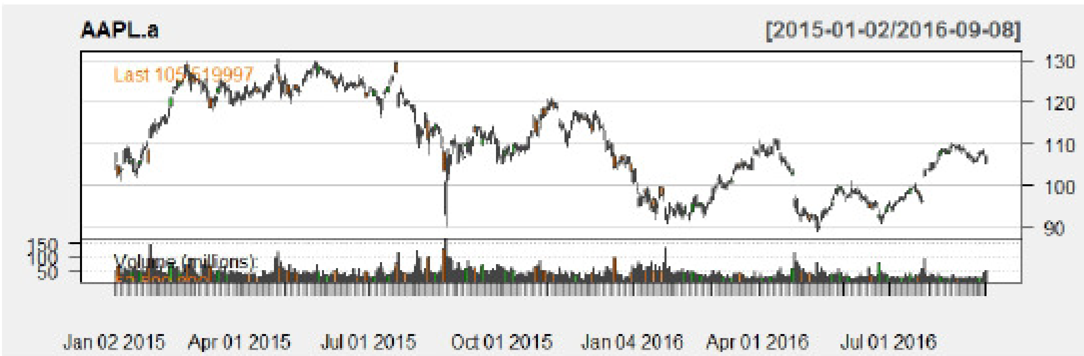
a) Using R, get AAPL's data from 2015-01-01 (see above). There is no deliverable for this part of the question.
b) Produce a plot of AAPL's (Apple) unadjusted closing price for the last year using the command `candleChart(AAPL,theme = "white")`
c) Produce a plot of AAPL's (Apple) adjusted closing price for the last year using the command `candleChart(AAPL.a,theme = "white")`
d) How do the plots in (b) and (c) differ?
e) What day of the year had the lowest trading volume for AAPL? Does this make sense? [see page 318 of the R Cookbook for how to find the min and max of a vector].
f) Go to finance.yahoo.com and find the mean volume (over the last three months) for AAPL.
g) Make a histogram of daily volume and overlay a density estimate on it (see Recipe 10.19 from the R Cookbook). Doe the data look Normal? Should it be Normal?
h) Find the mean and median of daily volume over the last three months for AAPL from R [get the data using [Use `getSymbols("AAPL", from = "2016-06-01")`]
i) Compare the mean values from (f) and (g). Are they similar?
j) Are the mean and median from (h) the same or different? What does that imply?
k) What is the correlation between AAPL's daily closing price and the daily closing price of QQQ (the Nasdaq 100 Index)?
   The relevant R command is `cor(Ad(QQQ),Ad(AAPL))`
l) What is the correlation between AAPL's daily volume and the daily volume of QQQ?
m) People who use technical analysis on the stock market often look to the 200 day moving average (often call the simple moving average) to measure long term trends. If a stock price is above its 200 day moving average it's a buy and above the long term trend. If its below its 200 day moving average it's a sell. Run the following R code and discuss the resultant graph in terms of buying and selling AAPL.

```
getSymbols("AAPL",from="2010-05-01")
AAPL.a=adjustOHLC(AAPL)
candleChart(AAPL.a, multi.col = TRUE, theme = "white")
addSMA(n=200)
```

5b)



5c)



5d) No discernible difference

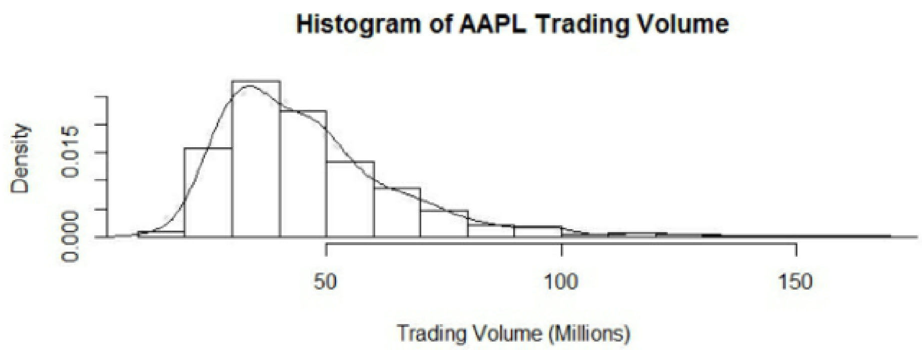
5e)

```
> AAPL$AAPL.Close[which.min(AAPL.a$AAPL.Close),]
           AAPL.Close
2016-05-12      90.34
```

Minimum closing price occurred on May 12, 2016.

5f) 32.91M

5g)



No, trading volume does not follow a Normal distribution - it is right-skewed instead of symmetrical about the mean. This makes sense, because periods of high trading activity coincide with earnings announcements or other momentous news, which occur only sporadically throughout the year. This is why you see a very long right tail at much lower frequency.

5h)

```
> getSymbols("AAPL", from = "2016-06-09")
[1] "AAPL"
> mean(AAPL$AAPL.Volume)
[1] 32911334
> median(AAPL$AAPL.Volume)
[1] 29065600
```

Mean trading volume over the last 3 months is 32.9M; median is 29.1M.

5i) The mean values are identical to 1 decimal place.

5j) The median is smaller than the mean, implying a right-skewed distribution.

5k)

```
> getSymbols("AAPL", from = "2015-01-01", to = "2016-09-01")
> getSymbols("QQQ", from = "2015-01-01", to = "2016-09-01")
> cor(Ad(QQQ),Ad(AAPL))
             AAPL.Adjusted
QQQ.Adjusted     0.1939789
```

5m)

If one adopted this strategy, one would generally make money if one held on to the stock for some time. However, there is the risk of buying the stock at inflection points i.e. at its peak before it begins a descent or at the trough before it begins an ascent.