Solution Manual

Professor Bradley Warner

Tuesday, September 30, 2016

These are my solutions to the problems assigned at the United States Air Force Academy for Math 377.

Chapter 1

Section 1.1 and 1.2

Problem 1.2 Load fastR and examine the data,

```
require('fastR')
```

```
## Warning: package 'fastR' was built under R version 3.2.2
```

```
names(littleSurvey)
```

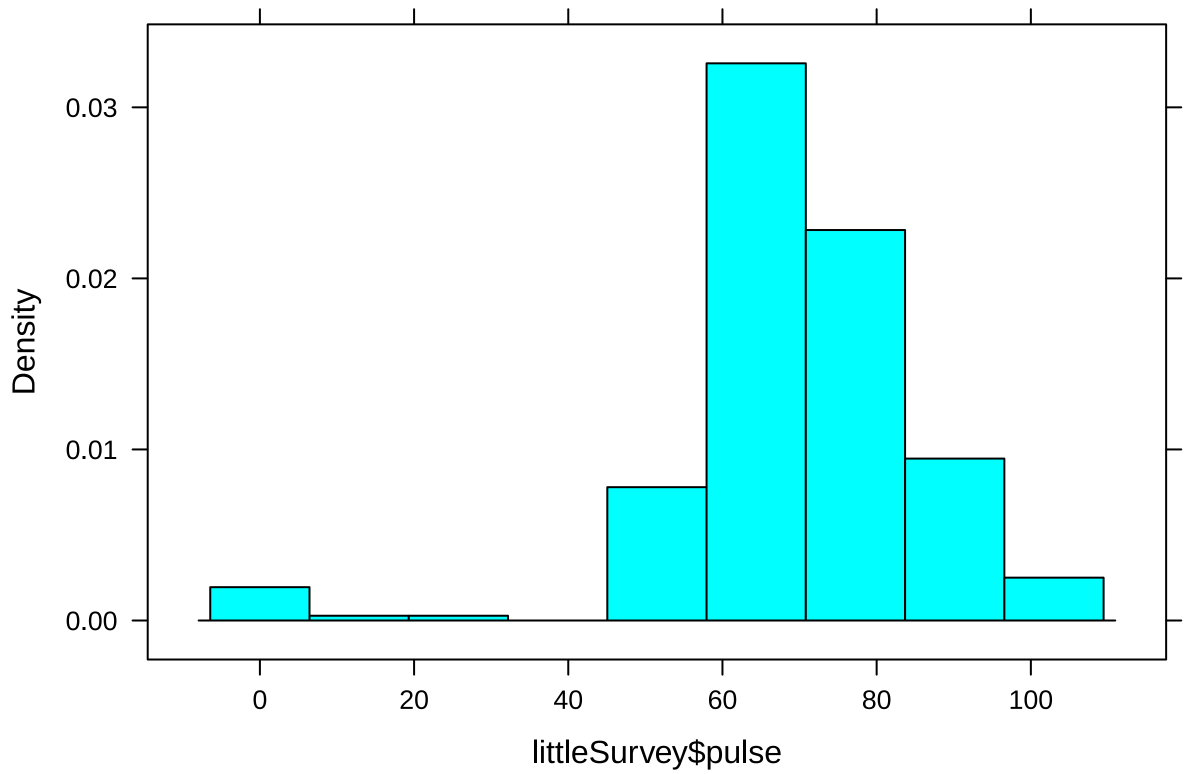
```
##  [1] "number"      "colorVer"    "color"       "otherColor"  "animalVer"
##  [6] "animal"      "otherAnimal" "pulseVer"    "pulse"       "TVver"
## [11] "tvBox"       "tvHours"     "surpriseVer" "surprise"    "playVer"
## [16] "play"        "diseaseVer"  "disease"     "homeworkVer" "homework"
```

```
head(littleSurvey)
```

```
##   number colorVer color otherColor animalVer   animal otherAnimal pulseVer
## 1     13       v1 other       blue        v1    other     penguin       v1
## 2     30       v1 black                   v1    other      monkey       v1
## 3     27       v2 other       Blue        v2    other     penguin       v1
## 4     30       v1 other       blue        v2    other     panther       v1
## 5     30       v1 black                   v1    other      monkey       v1
## 6     17       v2             BLUE        v1 elephant                   v1
##   pulse TVver tvBox tvHours surpriseVer surprise playVer play diseaseVer
## 1    72    v3   1-2       0          v1      yes      v1  yes         v2
## 2    72    v3  <NA>       0          v2     <NA>      v1 <NA>         v1
## 3    90    v1 other       3          v2       no      v2   no         v1
## 4    60    v1 other       0          v1       no      v2   no         v1
## 5    72    v3   2-4       0          v2       no      v1  yes         v1
## 6    52    v1 other       7          v2       no      v1  yes         v1
##   disease homeworkVer homework
## 1       B          v1        B
## 2    <NA>          v2     <NA>
## 3       A          v1        A
## 4       A          v2        A
## 5       A          v2        A
## 6       A          v2        A
```

Part a.

```r
histogram(littleSurvey$pulse)
```



There are some extremely low pulses including zero.

Part b.
I will pick pulses above 40.

```r
littleSurvey[littleSurvey$pulse<40,]
```

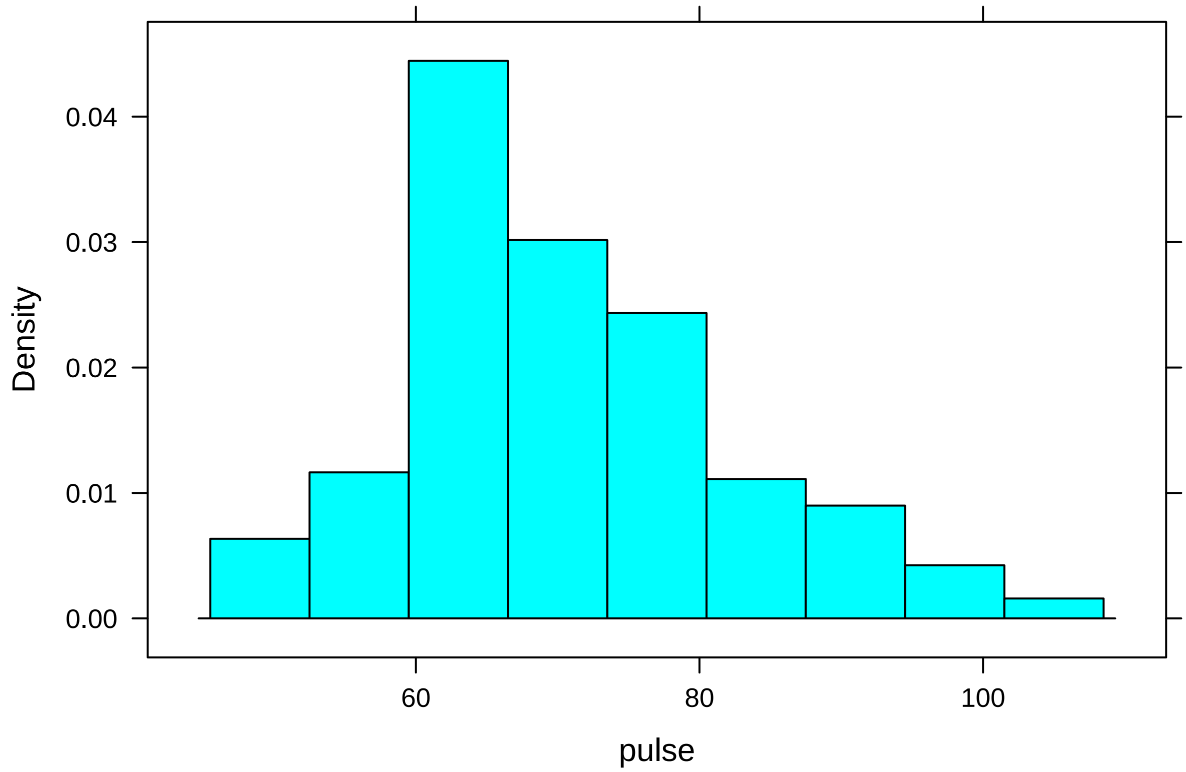
```
##     number colorVer color otherColor animalVer   animal otherAnimal
## 29      28       v1             Brown        v2 elephant
## 34      28       v1  <NA>       Brown        v2 elephant
## 127     19       v2 green                    v1    other       tiger
## 156      4       v2 other        pink        v1  giraffe
## 161     25       v1 other        blue        v2    other  panda bear
## 164      5       v2 other        pink        v1  giraffe
## 166      4       v1 green                    v2    other       Tiger
## 230      2       v2 other        blue        v1     lion
## 275      3       v1 other        Pink        v2  giraffe
##     pulseVer pulse TVver tvBox tvHours surpriseVer surprise playVer play
## 29        v1     0    v3    <1       0          v2      yes      v1   no
## 34        v1     0    v3    <1       0          v2      yes      v1   no
## 127       v1     0    v1 other       8          v2       no      v2  yes
## 156       v1     1    v3   1-2       0          v2      yes      v2  yes
```

```
## 161        v1     0     v2   1-2       0          v1      yes      v1  yes
## 164        v1    12     v1 other       4          v1       no      v1  yes
## 166        v1     0     v2    <1       0          v1      yes      v1  yes
## 230        v1     0     v3   2-4       0          v2       no      v1   no
## 275        v1    25     v2    >4       0          v2       no      v2  yes
##     diseaseVer disease homeworkVer homework
## 29          v2       B          v2        A
## 34          v2       B          v2        A
## 127         v1       B          v2        A
## 156         v2       A          v2        A
## 161         v1       A          v1        B
## 164         v2       B          v2        A
## 166         v1       A          v2        A
## 230         v2       A          v1        A
## 275         v1       B          v2        B
```

```r
subset(littleSurvey,pulse<40,pulse)
```

```
##     pulse
## 29      0
## 34      0
## 127     0
## 156     1
## 161     0
## 164    12
## 166     0
## 230     0
## 275    25
```

```r
histogram(~pulse,subset=pulse>=40,data=littleSurvey)
```



Part c.
The mean and median can be determined from the summary command.

```
summary(subset(littleSurvey,pulse>=40,pulse))
```

```
##      pulse
##  Min.   : 47.00
##  1st Qu.: 62.00
##  Median : 70.00
##  Mean   : 70.64
##  3rd Qu.: 78.00
##  Max.   :103.00
```

Or using inline r commands we have that Mean : 70.64 and Median : 70.00

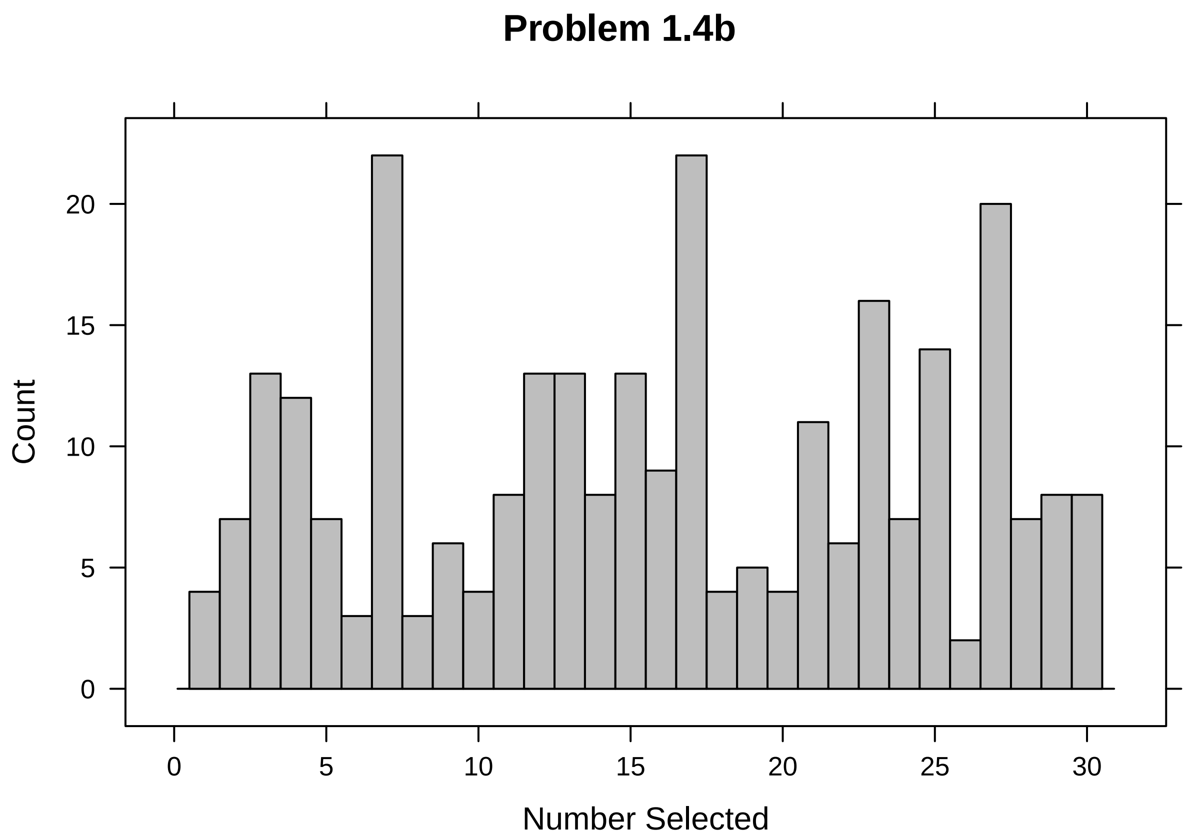
Problem 1.4 Part a.

```
table(littleSurvey$number)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
##  4  7 13 12  7  3 22  3  6  4  8 13 13  8 13  9 22  4  5  4 11  6 16  7 14
## 26 27 28 29 30
##  2 20  7  8  8
```

Part b.
The hard part is getting the bins. I used help on histogram to find a solution.

```
# ?histogram
histogram(littleSurvey$number, breaks = seq(0.5, 30.5), type = "count", xlab = "Number Selected",
    col = "grey", main = "Problem 1.4b")
```



Part c and d. I tried order first but it did not work. I used max and min but they only report the count. The which command was the key.

```
order(table(littleSurvey$number))
```

```
##  [1] 26  6  8  1 10 18 20 19  9 22  2  5 24 28 11 14 29 30 16 21  4  3 12
## [24] 13 15 25 23 27  7 17
```

```
max(table(littleSurvey$number))
```

```
## [1] 22
```

```
min(table(littleSurvey$number))
```

```
## [1] 2
```

```r
#?which
which(table(littleSurvey$number)==max(table(littleSurvey$number)))
```

```
##  7 17
##  7 17
```

```r
which(table(littleSurvey$number)==min(table(littleSurvey$number)))
```

```
## 26
## 26
```

so 7 and 17 were most common and 26 the least common.
Part e.
We have to using modular arithmetic

```r
5%%2
```

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

```r
littleSurvey$number%%2
```

```
##   [1] 1 0 1 0 0 1 1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1 1 0 0
##  [36] 1 0 1 1 1 1 1 0 1 0 0 0 0 0 1 1 1 1 0 0 1 1 0 1 0 0 1 1 1 1 1 1 0 1 1
##  [71] 1 1 1 1 1 0 0 0 0 0 1 1 0 1 1 1 0 1 0 0 1 0 1 0 1 1 1 0 0 0 0 0 1 1 0
## [106] 1 0 1 1 1 1 1 0 1 0 1 0 0 1 1 1 0 1 0 1 0 1 0 0 0 1 1 1 1 0 1 1 1 1 1
## [141] 1 1 0 0 1 1 1 1 1 0 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0
## [176] 1 0 1 1 1 0 0 1 1 0 1 1 0 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1 1 0 1 0 1 1
## [211] 0 0 1 1 1 0 1 0 1 1 0 0 1 0 1 1 1 1 1 0 0 1 0 1 1 1 0 1 1 0 1 1 1 1 0
## [246] 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1
```

```r
table(littleSurvey$number%%2)
```

```
##
##   0   1
##  97 182
```

```r
Prob1.4=table(littleSurvey$number%%2)
names(Prob1.4)=c('Even','Odd')
Prob1.4
```

```
## Even  Odd
##   97  182
```

Problem 1.6

The mean is a good measure if the data is well behaved in the sense of not having a small number of extreme values. This is where the median excels as a measure of cnetrality. Home prices or salaries are good examples of using the median in place of the mean. The mean is actually easier to calculate because the median requires sorting while the mean just requires addition. The mean also has better mathematical properties that we will learn about later in this course because of its additive nature.

Problem 1.8 Part a.
First we will experiment with the different summaries to gain insight.

```
fivenum(1:11)
```

```
## [1]  1.0  3.5  6.0  8.5 11.0
```

```
quantile(1:11)
```

```
##   0%  25%  50%  75% 100%
##  1.0  3.5  6.0  8.5 11.0
```

```
fivenum(c(rep(2,5),rep(4,5)))
```

```
## [1] 2 2 3 4 4
```

```
quantile(c(rep(2,5),rep(4,5)))
```

```
##   0%  25%  50%  75% 100%
##    2    2    3    4    4
```

```
fivenum(c(rep(2,5),rep(4,6)))
```

```
## [1] 2 2 4 4 4
```

```
quantile(c(rep(2,5),rep(4,6)))
```

```
##   0%  25%  50%  75% 100%
##    2    2    4    4    4
```

```
Prob1.8=c(2,2,3,3,4,4,5,5)
fivenum(Prob1.8)
```

```
## [1] 2.0 2.5 3.5 4.5 5.0
```

```
quantile(Prob1.8)
```

```
##   0%  25%  50%  75% 100%
## 2.00 2.75 3.50 4.25 5.00
```

The function fivenum is different if there are an even number of data points. In this case it is taking the median of the lower 50% for the 25th percentile and likewise for the 75th.

Problem 1.10 We denote the mean by

$$\bar{x} = \sum_{1}^{n} \frac{x_i}{n}$$

now

$$\sum_{i}^{n}(x_i - \bar{x}) = \sum_{i}^{n} x_i - \sum_{i}^{n} \bar{x}$$

but each of these terms are

$$n\bar{x}$$

so the total deviation from the mean is zero.

Problem 1.13 Let

$$SS(c) = \sum (x_i - c)^2$$

Since $SS(c)$ is a continuous function we will use the derivative to find the minimum.

$$\frac{dSS(c)}{dc} = \frac{d\sum(x_i - c)^2}{dc} = \sum \frac{d(x_i - c)^2}{dc} = \sum -2(x_i - c) = -2(\sum x_i - nc)$$

Setting equal to 0 and solving for c yields

$$-2(\sum x_i - nc) = 0$$

$$\sum x_i = nc$$

$$c = \bar{x}$$

You can verify it is a minimum by taking the second derivative and seeing the the result is $2n$ which is positive.

Section 1.3 and 1.4

First I will load the libraries needed.

```
require('fastR')
require("Hmisc")
```

Problem 1.14 Let's start with just two data points. If we consider the range as a length of 10, then the mean will be the midpoint of the two data points. The variance is maximized if we get the two data points as far as possible from the midpoint. Thus we pick one value as 0 and the other as 10. The same logic applies to all ten data points. Put five at 0 and five at 10.

```
Prob1.14=rep(c(0,10),5)
mean(Prob1.14)
```

```
## [1] 5
```

```
var(Prob1.14)
```

```
## [1] 27.77778
```

What if you had an odd number of data points?

Problem 1.15 The smallest variance possible is zero, since variance is a sum of squares it is non-negative. For a variance to be zero, all the values must be the same. Here is an example:

```
Prob1.15=rep(3,10)
var(Prob1.15)
```

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

Problem 1.16 First let's get familiar with the data set.

```r
str(pitching2005)
```

```
## 'data.frame':    653 obs. of  27 variables:
##  $ playerID: Factor w/ 606 levels "accarje01","acevejo01",..: 142 547 84 271 334 601 296 355 351 512
##  $ yearID  : int  2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 ...
##  $ stint   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ teamID  : Factor w/ 30 levels "ARI","ATL","BAL",..: 2 10 24 5 22 20 19 1 29 20 ...
##  $ lgID    : Factor w/ 2 levels "AL","NL": 2 1 1 1 2 1 2 2 1 1 ...
##  $ W       : int  0 0 0 1 3 4 0 4 1 5 ...
##  $ L       : int  1 2 0 1 1 0 0 1 3 1 ...
##  $ G       : int  5 2 2 32 6 40 33 27 13 67 ...
##  $ GS      : int  0 2 1 0 6 0 0 0 7 0 ...
##  $ CG      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SHO     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SV      : int  0 0 0 6 0 1 0 0 0 23 ...
##  $ IPouts  : int  15 34 6 118 124 174 69 91 136 235 ...
##  $ H       : int  6 15 1 34 31 64 22 21 49 53 ...
##  $ ER      : int  7 9 0 12 10 29 10 6 32 15 ...
##  $ HR      : int  2 1 0 3 2 6 2 2 7 3 ...
##  $ BB      : int  5 5 1 15 17 26 13 11 17 26 ...
##  $ SO      : int  3 7 1 50 26 44 23 31 34 72 ...
##  $ BAOpp   : logi  NA NA NA NA NA NA ...
##  $ ERA     : num  12.6 7.15 0 2.75 2.18 4.5 3.91 1.78 6.35 1.72 ...
##  $ IBB     : int  1 0 0 3 0 3 1 0 0 4 ...
##  $ WP      : int  0 1 0 4 0 2 0 1 7 1 ...
##  $ HBP     : int  0 1 0 1 3 8 2 1 7 2 ...
##  $ BK      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ BFP     : int  26 54 9 168 168 262 106 122 205 306 ...
##  $ GF      : int  1 0 1 18 0 15 4 10 2 47 ...
##  $ R       : int  7 9 0 15 10 34 12 6 34 17 ...
```

```r
head(pitching2005)
```

```
##    playerID yearID stint teamID lgID W L  G GS CG SHO SV IPouts  H ER HR
## 1 devinjo01   2005     1    ATL   NL 0 1  5  0  0   0  0     15  6  7  2
## 2 verlaju01   2005     1    DET   AL 0 2  2  2  0   0  0     34 15  9  1
## 3 campijo01   2005     1    SEA   AL 0 0  2  1  0   0  0      6  1  0  0
## 4 jenksbo01   2005     1    CHA   AL 1 1 32  0  0   0  6    118 34 12  3
## 5 maholpa01   2005     1    PIT   NL 3 1  6  6  0   0  0    124 31 10  2
## 6  yabuke01   2005     1    OAK   AL 4 0 40  0  0   0  1    174 64 29  6
##   BB SO BAOpp   ERA IBB WP HBP BK BFP GF  R
## 1  5  3    NA 12.60   1  0   0  0  26  1  7
## 2  5  7    NA  7.15   0  1   1  0  54  0  9
## 3  1  1    NA  0.00   0  0   0  0   9  1  0
## 4 15 50    NA  2.75   3  4   1  0 168 18 15
## 5 17 26    NA  2.18   0  0   3  0 168  0 10
## 6 26 44    NA  4.50   3  2   8  0 262 15 34
```

You can get help on this data set by typing

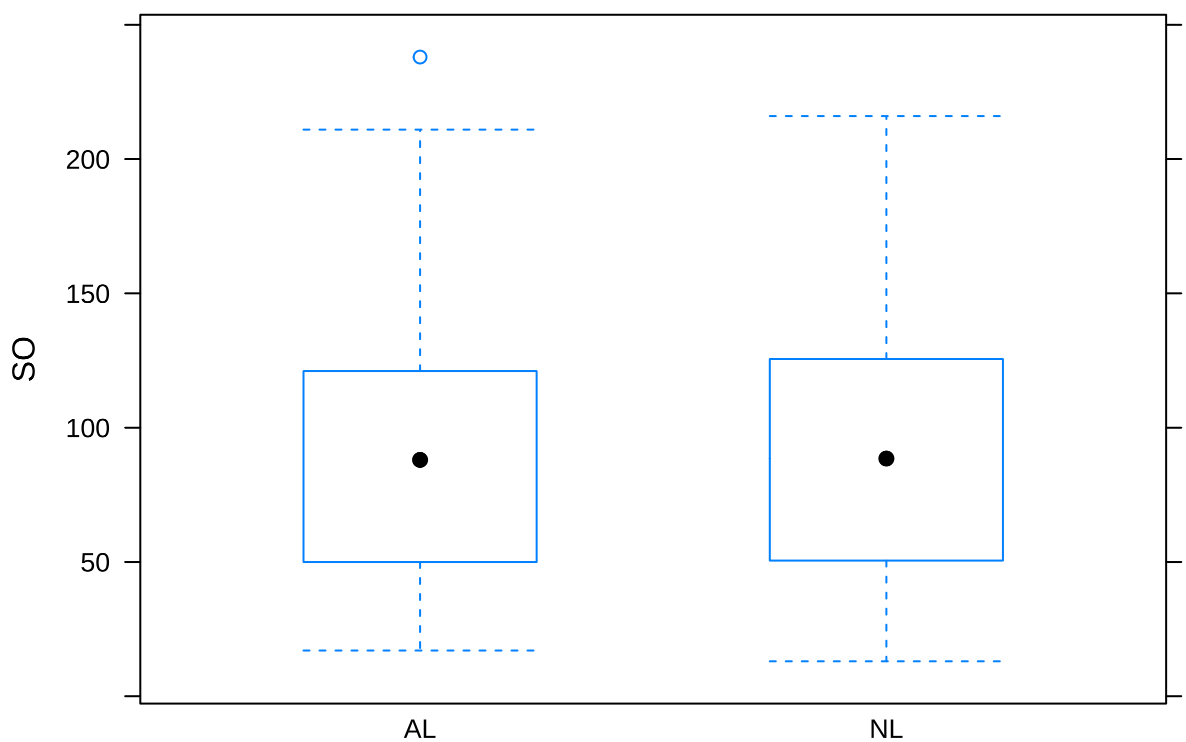
```
?pitching2005
```

I am interested in strike outs so that is the outcome of interest I will use for this problem.

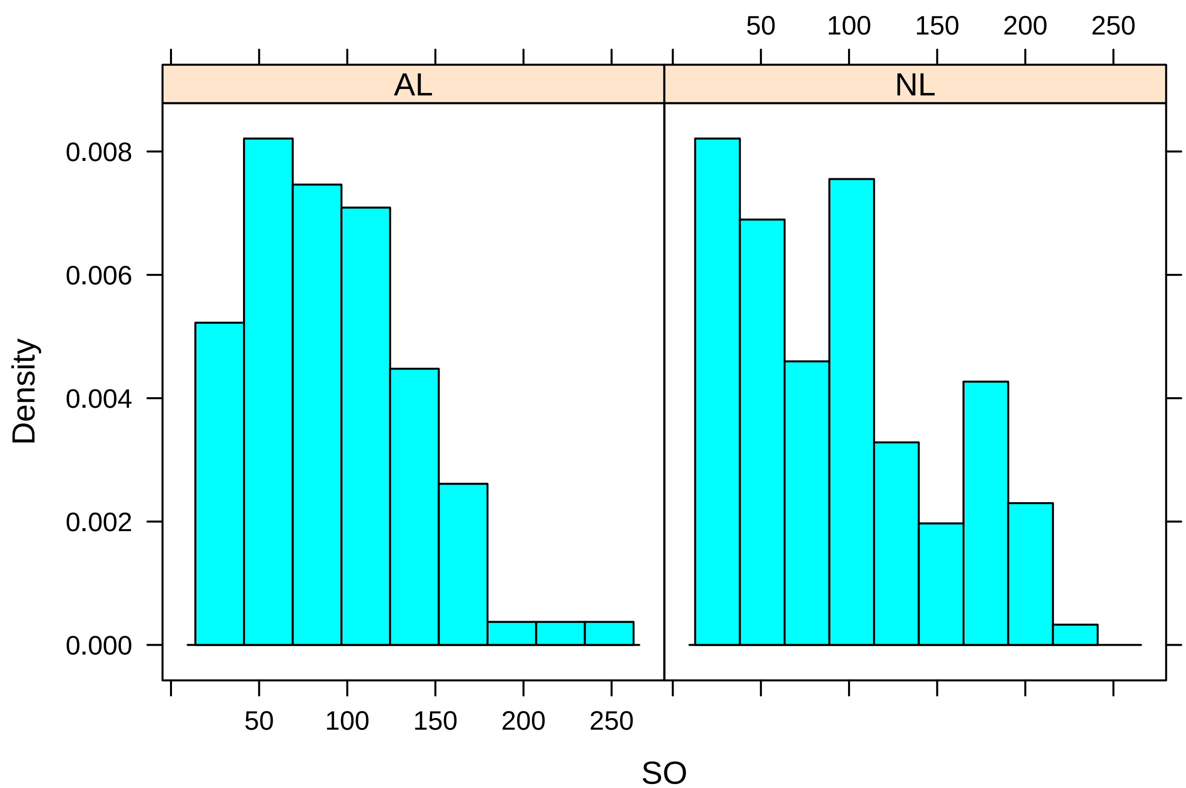
```r
summary(SO~lgID,data=pitching2005,subset=GS>4)
```

```
## SO    N=217
##
## +-------+--+---+--------+
## |       |  |N  |SO      |
## +-------+--+---+--------+
## |lgID   |AL| 97|91.07216|
## |       |NL|120|93.92500|
## +-------+--+---+--------+
## |Overall|  |217|92.64977|
## +-------+--+---+--------+
```

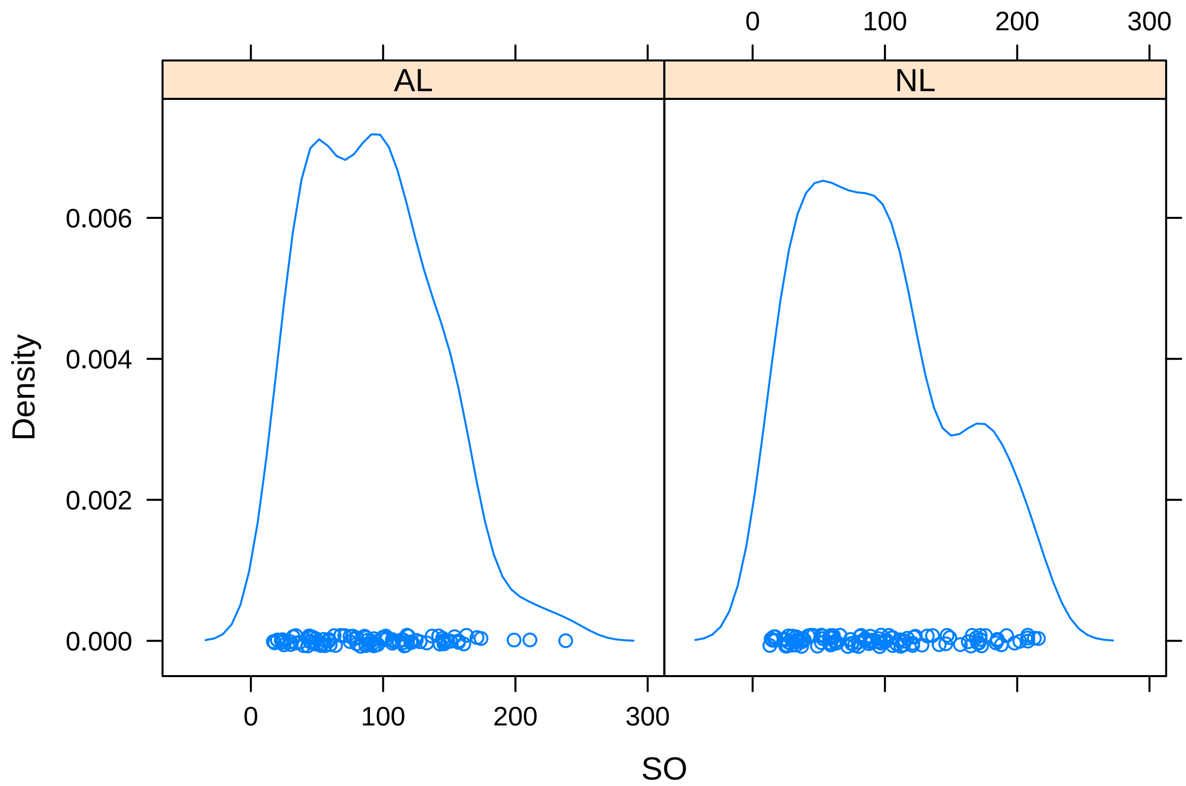
```r
bwplot(SO~lgID,data=pitching2005,subset=GS>4)
```



```r
histogram(~SO|lgID,data=pitching2005,subset=GS>4)
```




```r
densityplot(~SO|lgID,data=pitching2005,subset=GS>4)
```




```r
summary(SO~lgID,data=pitching2005,subset=GS>4,fun=favstats)
```

```
## SO    N=217
##
## +--------+--+---+----+-----+-------+------+----+--------+--------+---+--------+
## |        |  |N  |min |Q1   |median |Q3    |max |mean    |sd      |n  |missing |
## +--------+--+---+----+-----+-------+------+----+--------+--------+---+--------+
## |lgID    |AL| 97|17  |50.00|88.0   |121.00|238 |91.07216|47.43707| 97|0       |
## |        |NL|120|13  |51.25|88.5   |124.25|216 |93.92500|55.90135|120|0       |
## +--------+--+---+----+-----+-------+------+----+--------+--------+---+--------+
## |Overall|  |217|13  |50.00|88.0   |122.00|238 |92.64977|52.18971|217|0       |
## +--------+--+---+----+-----+-------+------+----+--------+--------+---+--------+
```

In terms of strike outs, there does not appear to be much difference in the two leagues.

Problem 1.17 Again, let's first get a sense for the data.

```r
head(batting)
```

```
##          player year stint team league   G  AB  R  H H2B H3B HR RBI SB CS
## 34289 abbotje01 2000     1  CHA     AL  80 215 31 59  15   1  3  29  2  1
## 34290 abbotpa01 2000     1  SEA     AL   2   5  1  2   1   0  0   0  0  0
## 34291 alcanis01 2000     1  BOS     AL  21  45  9 13   1   0  4   7  0  0
## 34292 alexama02 2000     1  BOS     AL 101 194 30 41   4   3  4  19  2  0
```

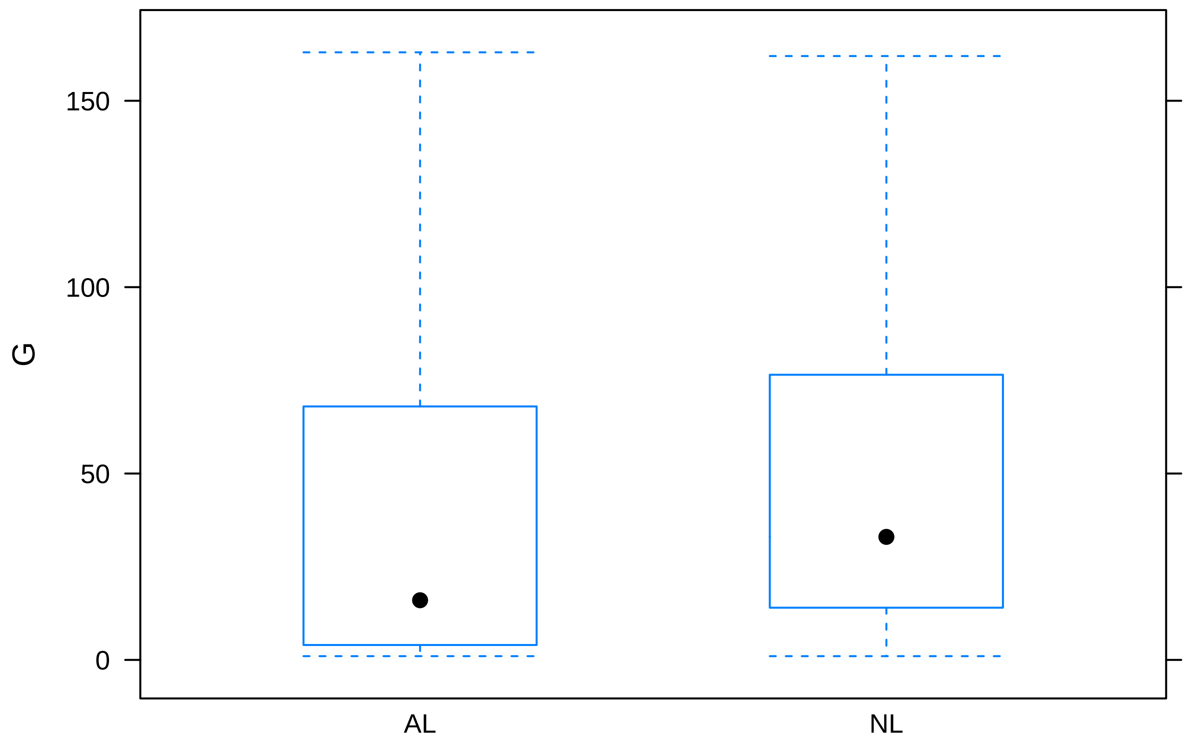
```
## 34293 alicelu01 2000     1  TEX     AL 139 540 85 159  25   8  6  63  1  3
## 34294 allench01 2000     1  MIN     AL  15  50  2  15   3   0  0   7  0  2
##       BB SO IBB HBP SH SF GIDP
## 34289 21 38   1   2  2  1    2
## 34290  0  1   0   0  1  0    0
## 34291  3  7   0   0  0  0    0
## 34292 13 41   0   0  2  0    0
## 34293 59 75   1   5  7  7   13
## 34294  3 14   0   1  0  1    1
```

```r
summary(batting)
```

```
##        player          year          stint             team      league
##  chenbr01 :  11   Min.   :2000   Min.   :1.000   SDN     : 307   AA:   0
##  micelda01:  10   1st Qu.:2001   1st Qu.:1.000   CLE     : 302   AL:3767
##  chrisja01:   9   Median :2002   Median :1.000   TEX     : 300   NL:4295
##  dejeami01:   9   Mean   :2002   Mean   :1.089   KCA     : 297
##  embreal01:   9   3rd Qu.:2004   3rd Qu.:1.000   COL     : 296
##  garcika01:   9   Max.   :2005   Max.   :4.000   BOS     : 288
##  (Other)  :8005                                  (Other):6272
##        G                AB            R                H
##  Min.   :  1.00   Min.   :  0   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  7.00   1st Qu.:  1   1st Qu.:  0.00   1st Qu.:  0.00
##  Median : 29.00   Median : 20   Median :  1.00   Median :  3.00
##  Mean   : 46.96   Mean   :124   Mean   : 17.27   Mean   : 32.87
##  3rd Qu.: 74.00   3rd Qu.:186   3rd Qu.: 23.00   3rd Qu.: 47.00
##  Max.   :163.00   Max.   :704   Max.   :152.00   Max.   :262.00
##
##       H2B              H3B                HR              RBI
##  Min.   : 0.000   Min.   : 0.0000   Min.   : 0.000   Min.   :  0.00
##  1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.:  0.00
##  Median : 0.000   Median : 0.0000   Median : 0.000   Median :  1.00
##  Mean   : 6.577   Mean   : 0.6848   Mean   : 3.955   Mean   : 16.44
##  3rd Qu.: 9.000   3rd Qu.: 1.0000   3rd Qu.: 4.000   3rd Qu.: 22.00
##  Max.   :59.000   Max.   :20.0000   Max.   :73.000   Max.   :160.00
##
##        SB               CS                BB               SO
##  Min.   : 0.000   Min.   : 0.0000   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.:  0.00   1st Qu.:  0.00
##  Median : 0.000   Median : 0.0000   Median :  1.00   Median :  6.00
##  Mean   : 2.047   Mean   : 0.9072   Mean   : 12.11   Mean   : 23.37
##  3rd Qu.: 1.000   3rd Qu.: 1.0000   3rd Qu.: 16.00   3rd Qu.: 35.00
##  Max.   :70.000   Max.   :24.0000   Max.   :232.00   Max.   :195.00
##
##       IBB               HBP               SH               SF
##  Min.   :  0.0000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:  0.0000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000
##  Median :  0.0000   Median : 0.000   Median : 0.000   Median : 0.000
##  Mean   :  0.9871   Mean   : 1.329   Mean   : 1.221   Mean   : 1.036
##  3rd Qu.:  1.0000   3rd Qu.: 1.000   3rd Qu.: 1.000   3rd Qu.: 1.000
##  Max.   :120.0000   Max.   :30.000   Max.   :24.000   Max.   :16.000
##  NA's   :1          NA's   :10                        NA's   :1
##       GIDP
##  Min.   : 0.000
```

```
##   1st Qu.: 0.000
##   Median : 0.000
##   Mean   : 2.846
##   3rd Qu.: 4.000
##   Max.   :32.000
##
```

```
bwplot(G~league,data=batting)
```




```
summary(G~league,data=batting,fun=favstats)
```

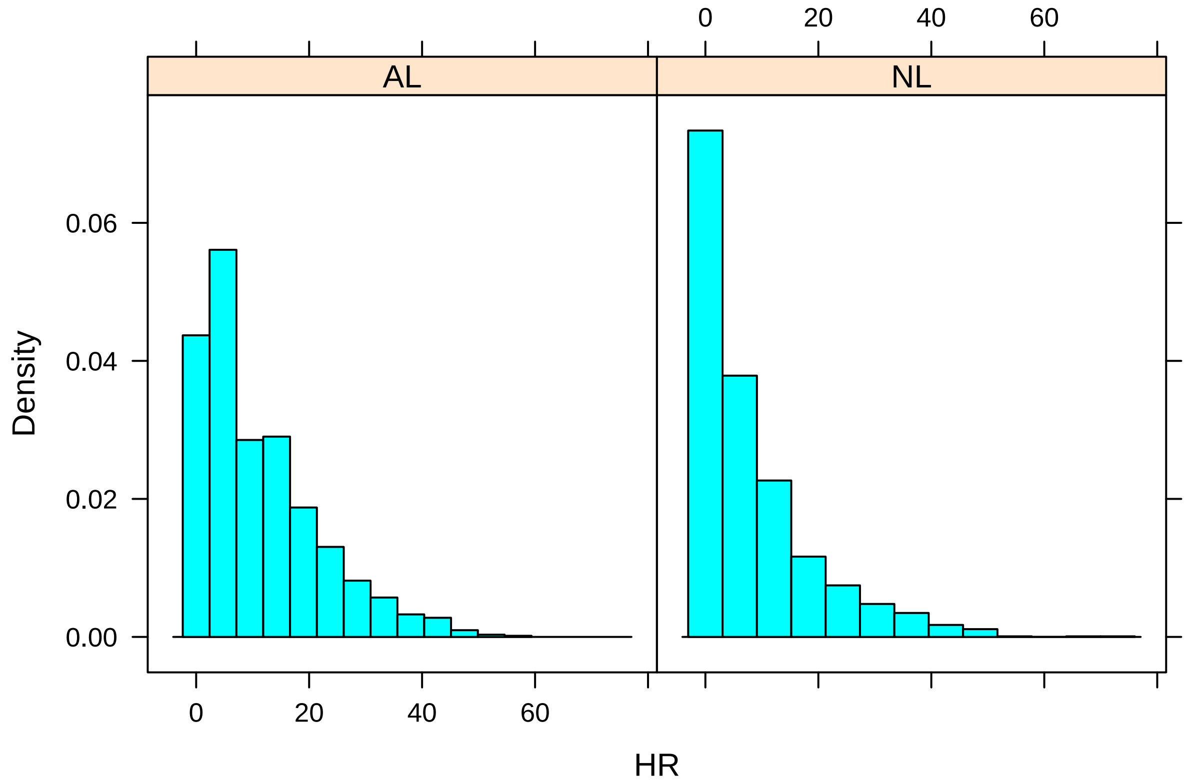
```
## G    N=8062
##
## +-------+--+----+----+---+-------+----+----+--------+--------+----+--------+
## |       |  |N   |min |Q1 |median |Q3  |max |mean    |sd      |n   |missing |
## +-------+--+----+----+---+-------+----+----+--------+--------+----+--------+
## |league |AA|   0|    |   |       |    |    |        |        |    |        |
## |       |AL|3767| 1  | 4 |16     |68.0|163 |41.93655|50.32067|3767| 0      |
## |       |NL|4295| 1  |14 |33     |76.5|162 |51.35856|46.23473|4295| 0      |
## +-------+--+----+----+---+-------+----+----+--------+--------+----+--------+
## |Overall|  |8062| 1  | 7 |29     |74.0|163 |46.95609|48.41282|8062| 0      |
## +-------+--+----+----+---+-------+----+----+--------+--------+----+--------+
```

Based on these summaries, subjectively I will use players who played at least 41 games. Since home runs are so exciting for the fans, I will use this as the outcome measure.

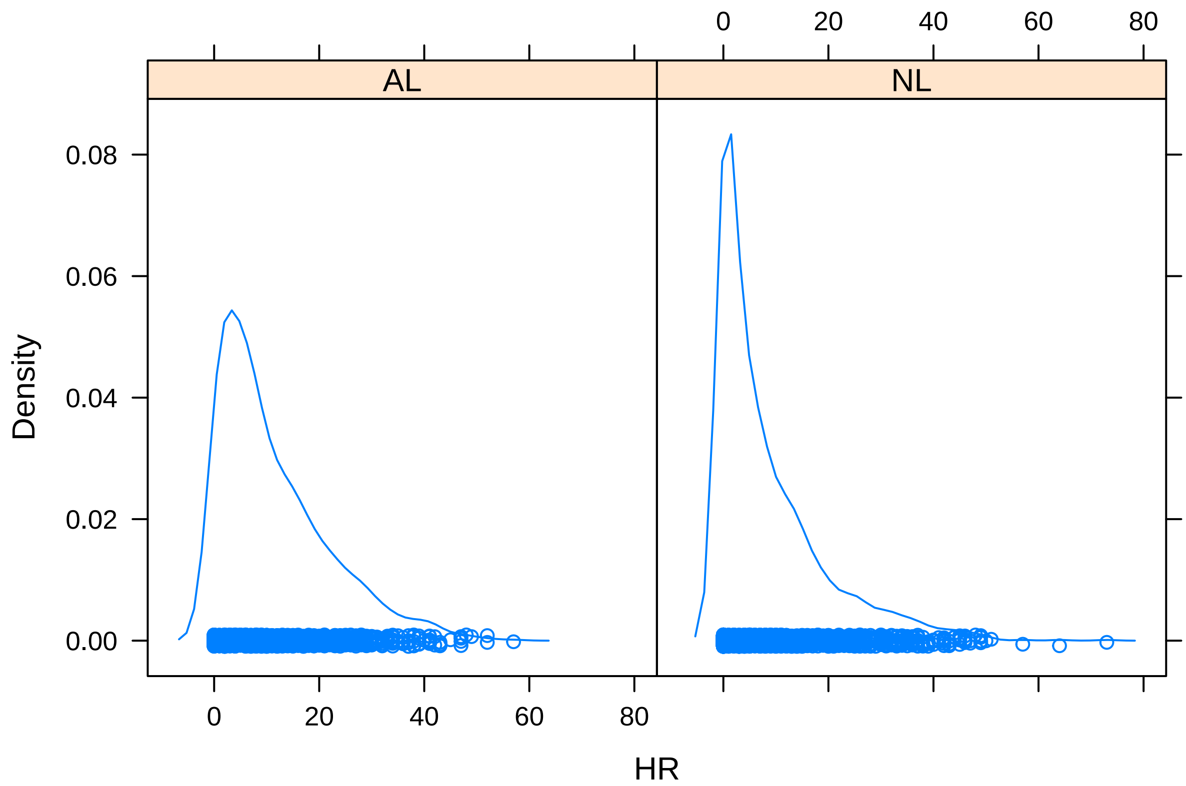
```r
summary(HR~league,data=batting,subset=G>40)
```

```
## HR    N=3184
##
## +-------+--+----+---------+
## |       |  |N   |HR       |
## +-------+--+----+---------+
## |league |AA|   0|         |
## |       |AL|1291|11.312161|
## |       |NL|1893| 8.435288|
## +-------+--+----+---------+
## |Overall|  |3184| 9.601759|
## +-------+--+----+---------+
```

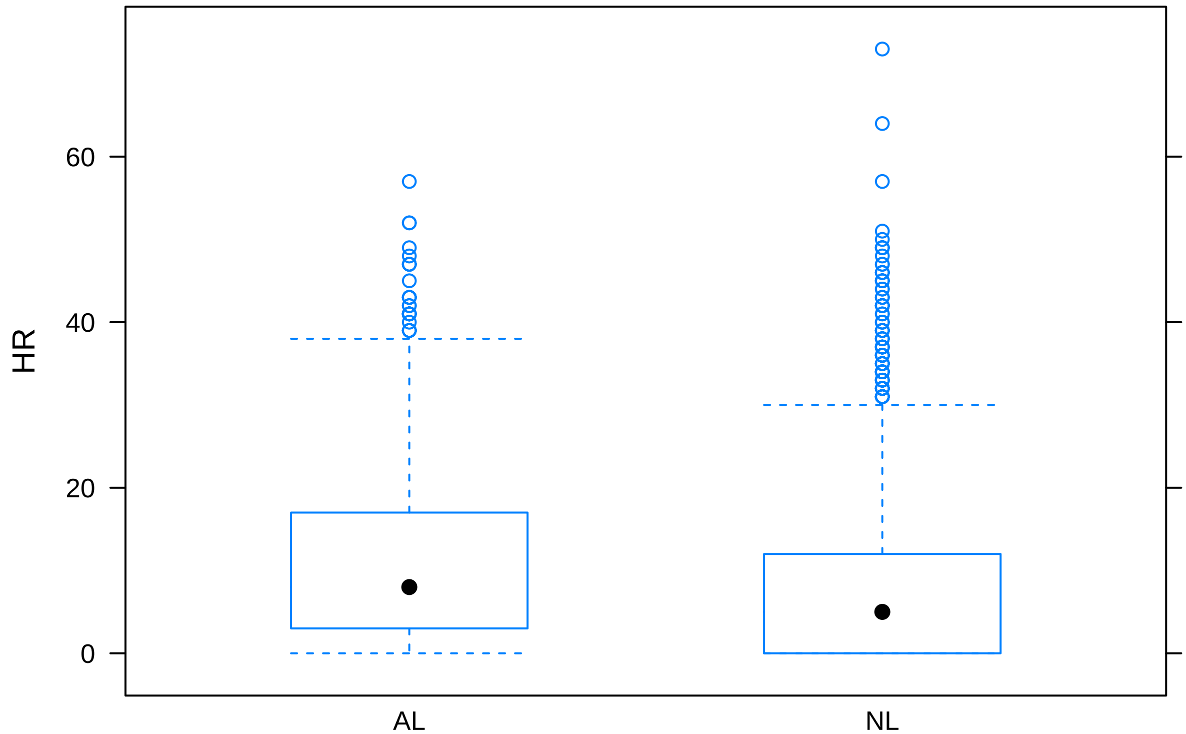
```r
histogram(~HR|league,data=batting,subset=G>40)
```




```r
densityplot(~HR|league,data=batting,subset=G>40)
```




```
bwplot(HR~league,data=batting,subset=G>40)
```



The American League with it designated hitter, appears to have more offense in the form of home runs.

Problem 1.19 Examine the data.

```
head(faithful)
```

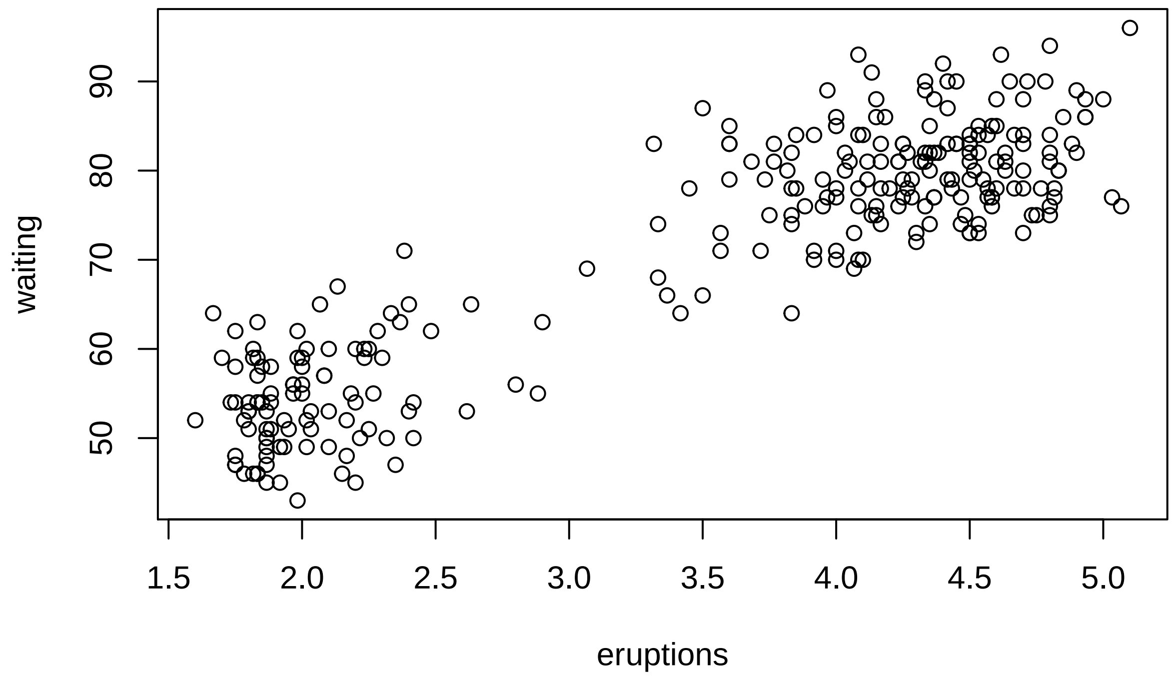
```
##   eruptions waiting
## 1     3.600      79
## 2     1.800      54
## 3     3.333      74
## 4     2.283      62
## 5     4.533      85
## 6     2.883      55
```

```
str(faithful)
```

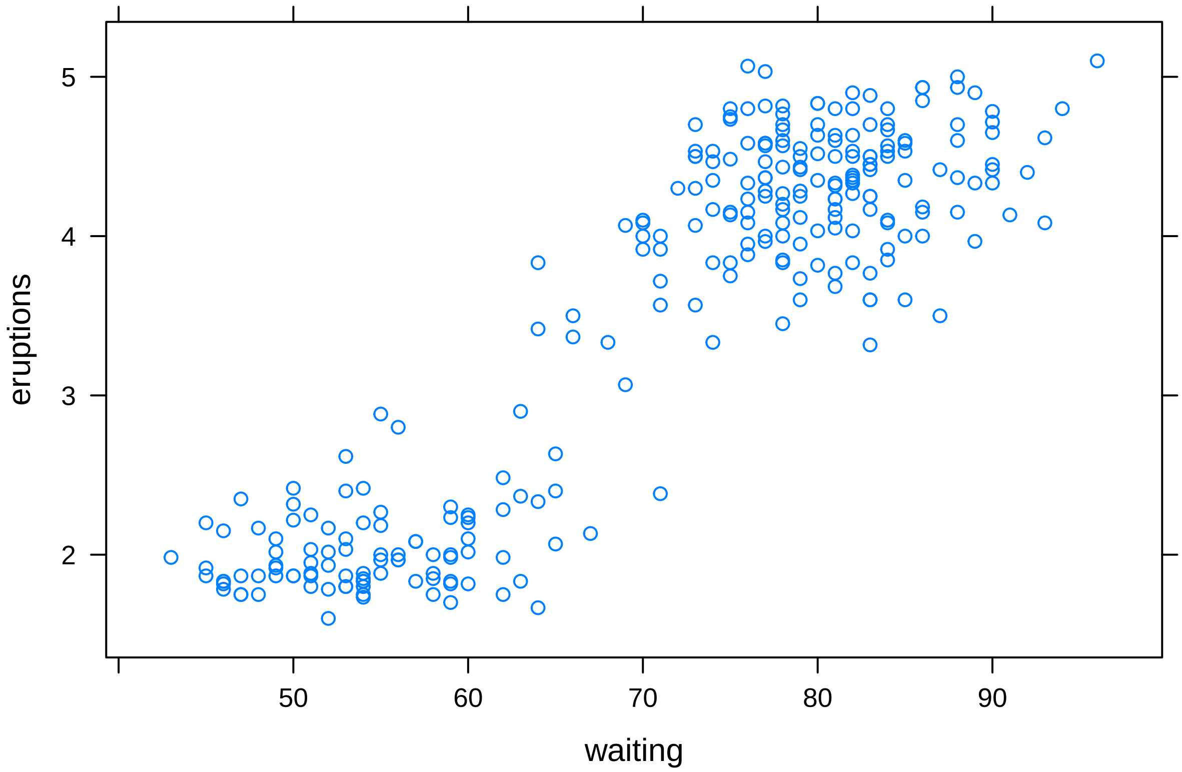
```
## 'data.frame':    272 obs. of  2 variables:
##  $ eruptions: num  3.6 1.8 3.33 2.28 4.53 ...
##  $ waiting  : num  79 54 74 62 85 55 88 85 51 85 ...
```

Part a requests a scatterplot. I have included two, the first using the base package and the second using the lattice package. The first uses the default order for the axes while in the second I selected the order.

```r
plot(faithful)
```



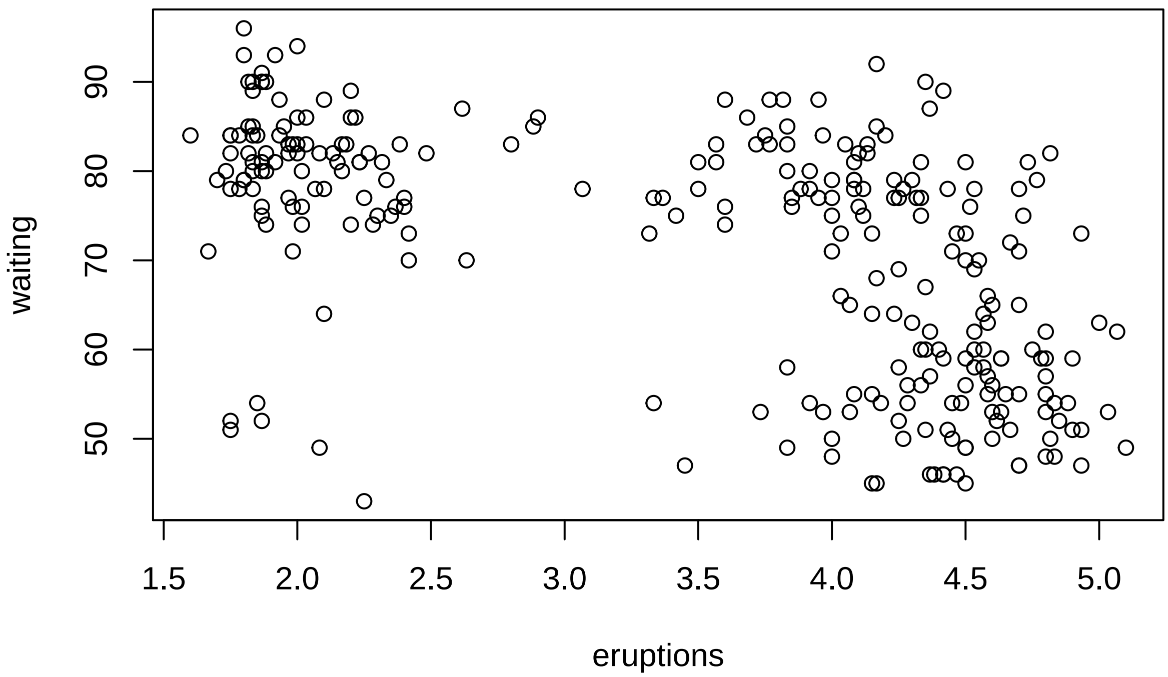

```r
xyplot(eruptions~waiting,faithful)
```



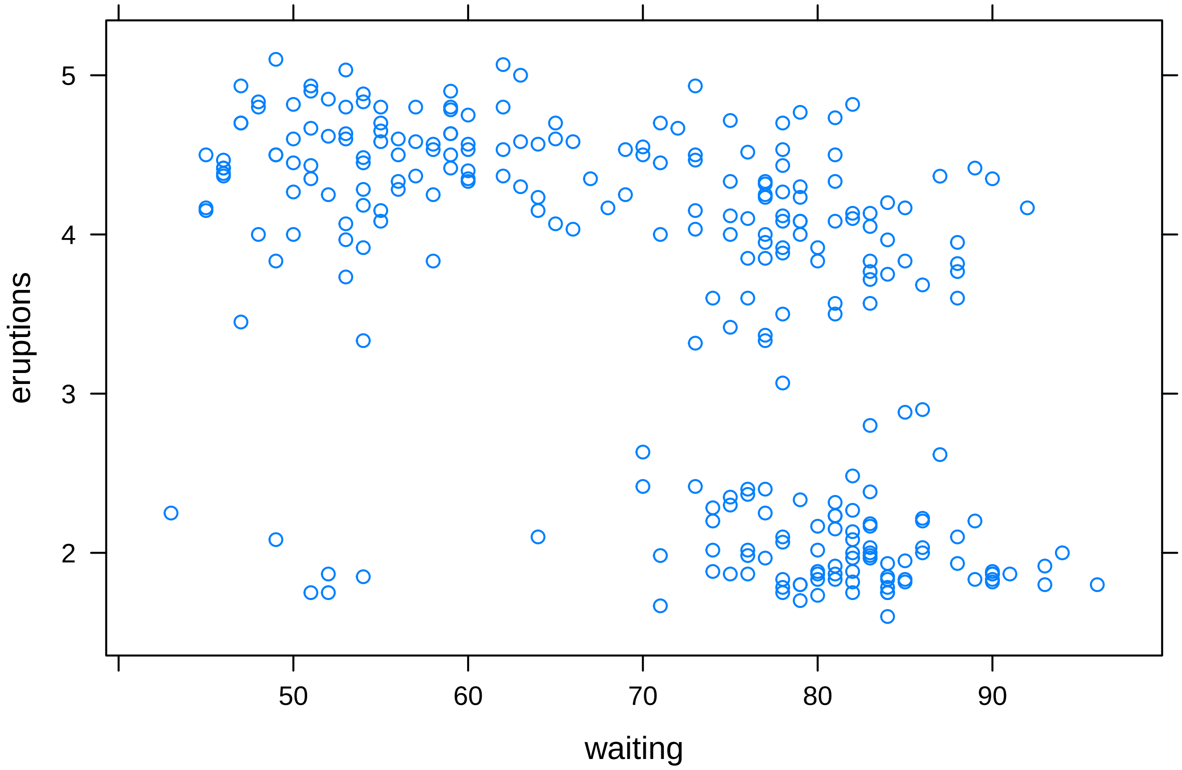
There appear to be two clusters in the data, one in the lower left and one in the upper right. There is also a strong linear correlation in that as waiting time increases the duration of eruptions increases.

Part b. By removing the first eruption time and the last waiting time we are reordering the data so that wait occurs first then eruption.

```
myfaithful=faithful
myfaithful[1:271,1]=myfaithful[2:272,1]
myfaithful=myfaithful[-272,]
plot(myfaithful)
```

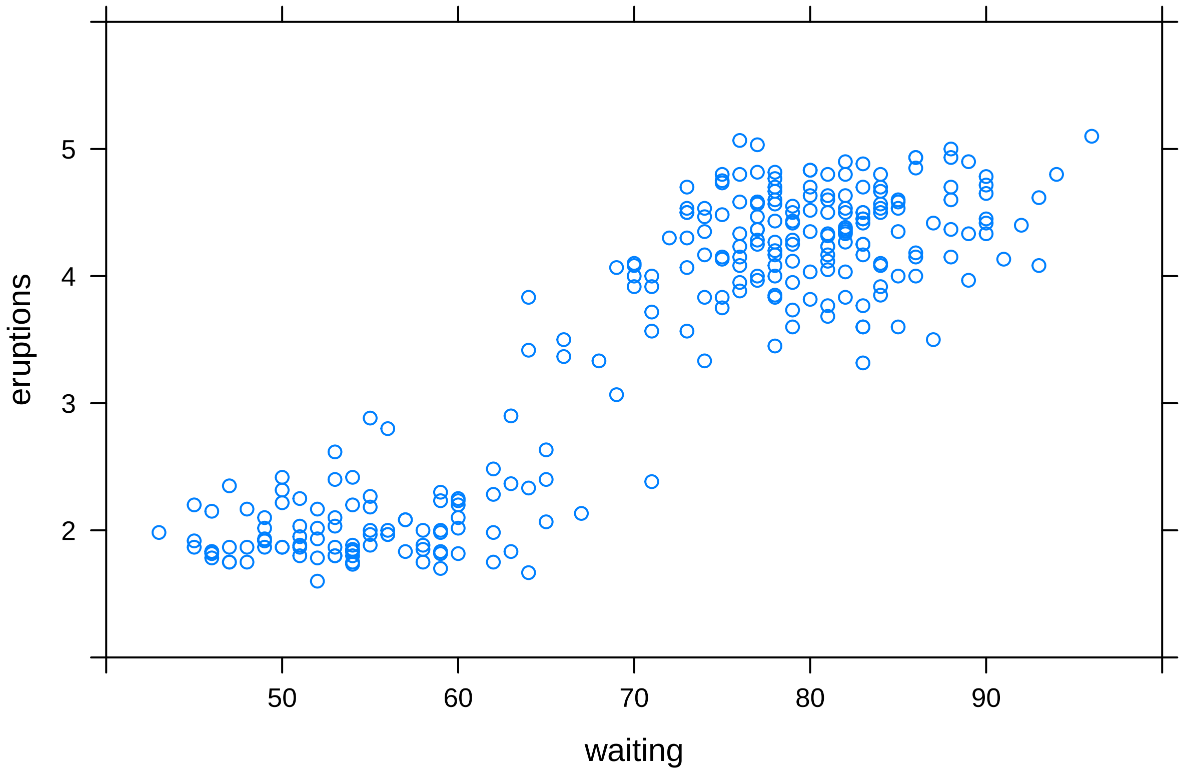



```
xyplot(eruptions~waiting,myfaithful)
```

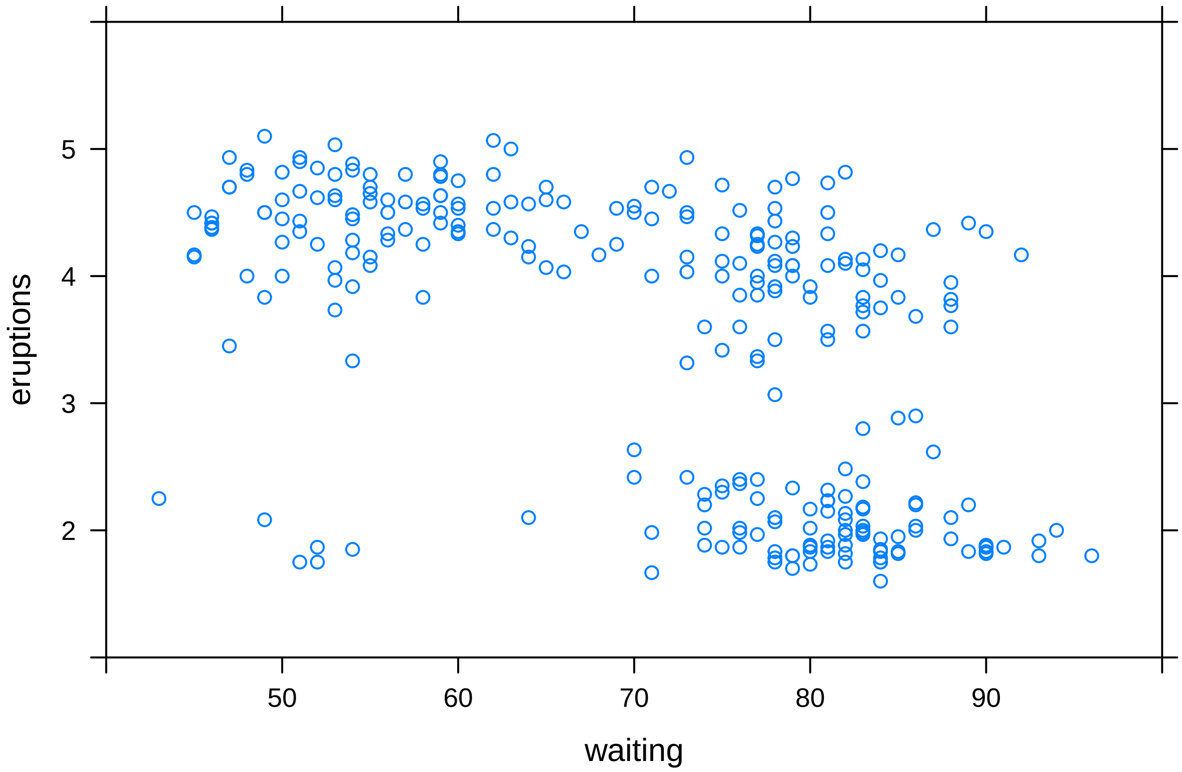


Let's use a common set of axes for both plots.

```
xyplot(eruptions~waiting,faithful,xlim=c(40,100),ylim=c(1,6))
```




```r
xyplot(eruptions~waiting,myfaithful,xlim=c(40,100),ylim=c(1,6))
```



Part c. Now there is not correlation between waiting and eruptions. so for this data set, to observe the relationship, it is important to associate the times in the appropriate manner. The wait time does not impact the length of the next eruption, but the length of the eruption impacts the wait until the next eruption.

Problem 1.21 Let's explore the data.

```
head(utilities)
```

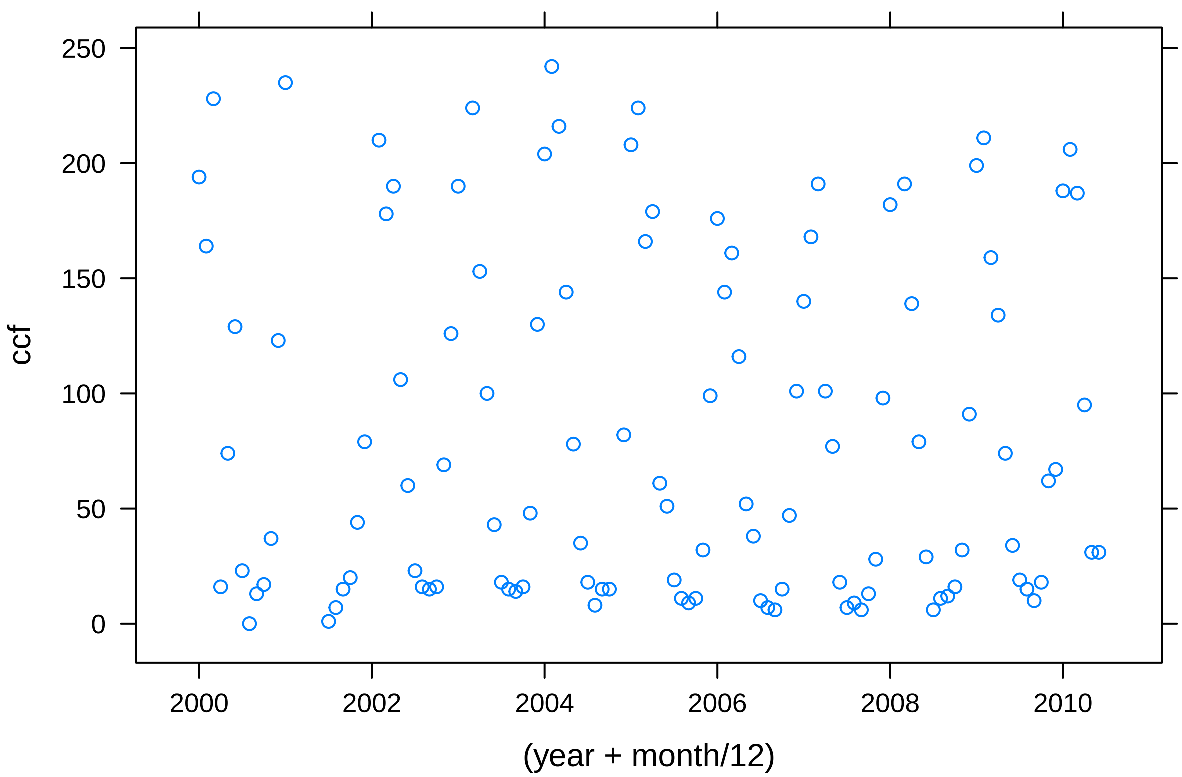
```
##   month day year temp kwh ccf thermsPerDay billingDays totalbill gasbill
## 1    12  29 1999   26 892 194          5.5          36    173.65  112.72
## 2     1  28 2000   18 533 164          5.6          30    139.18   95.88
## 3     2  26 2000   24 521 228          8.0          29    177.48  134.65
## 4     3  25 2000   41 554  16          0.6          28     61.27   15.32
## 5     4  28 2000   45 638  74          2.2          34    100.33   47.33
## 6     5  30 2000   60 700 129          4.1          32    153.32   89.87
##   elecbill             notes
## 1    68.25
## 2    43.30
## 3    42.83
## 4    45.95 bad meter reading
## 5    53.00
## 6    63.45
```

```r
str(utilities)
```

```
## 'data.frame':    117 obs. of  12 variables:
##  $ month       : int  12 1 2 3 4 5 6 7 8 9 ...
##  $ day         : int  29 28 26 25 28 30 24 26 24 25 ...
##  $ year        : int  1999 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
##  $ temp        : int  26 18 24 41 45 60 66 72 72 64 ...
##  $ kwh         : int  892 533 521 554 638 700 583 935 789 864 ...
##  $ ccf         : int  194 164 228 16 74 129 23 0 13 17 ...
##  $ thermsPerDay: num  5.5 5.6 8 0.6 2.2 4.1 0.9 0 0.4 0.5 ...
##  $ billingDays : int  36 30 29 28 34 32 25 32 29 32 ...
##  $ totalbill   : num  173.7 139.2 177.5 61.3 100.3 ...
##  $ gasbill     : num  112.7 95.9 134.7 15.3 47.3 ...
##  $ elecbill    : num  68.2 43.3 42.8 46 53 ...
##  $ notes       : Factor w/ 12 levels "","24.05 interim elec refund",..: 1 1 1 6 1 1 1 1 1 1 ...
```

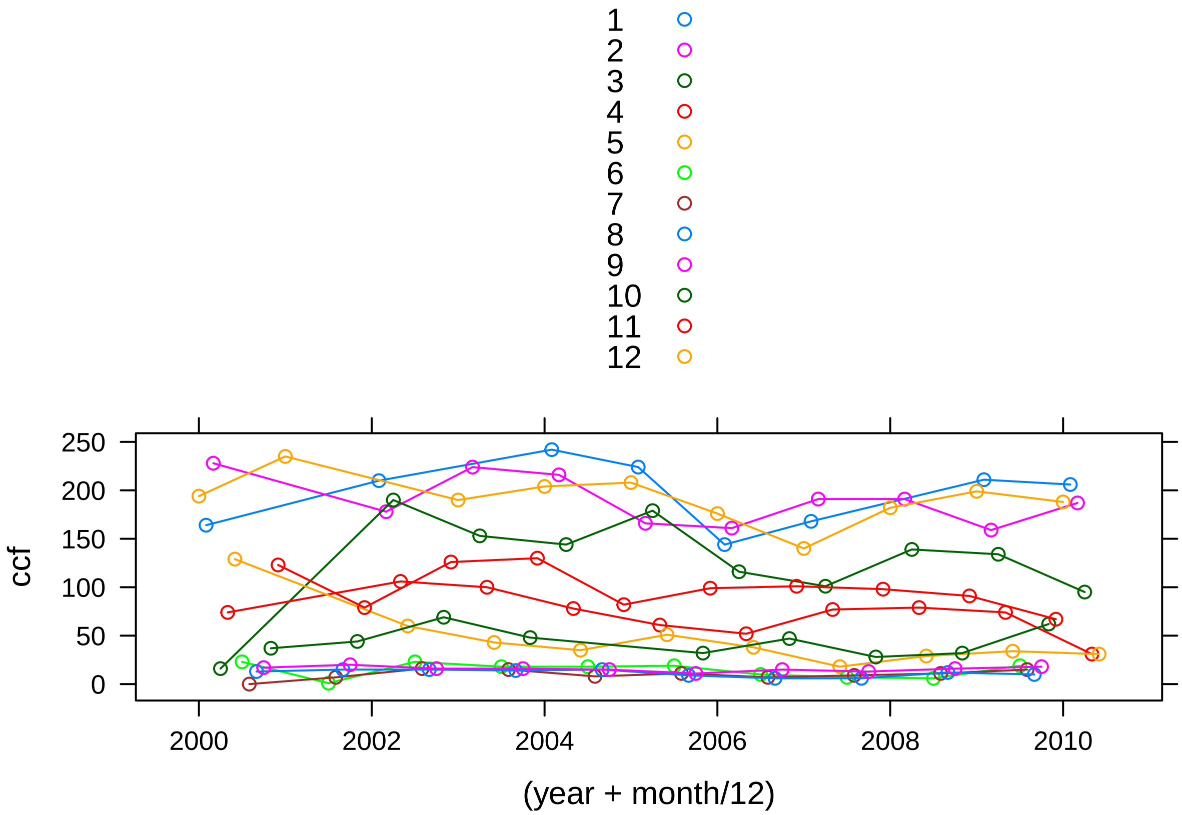
Part a.

```r
xyplot(ccf~(year+month/12),utilities)
```

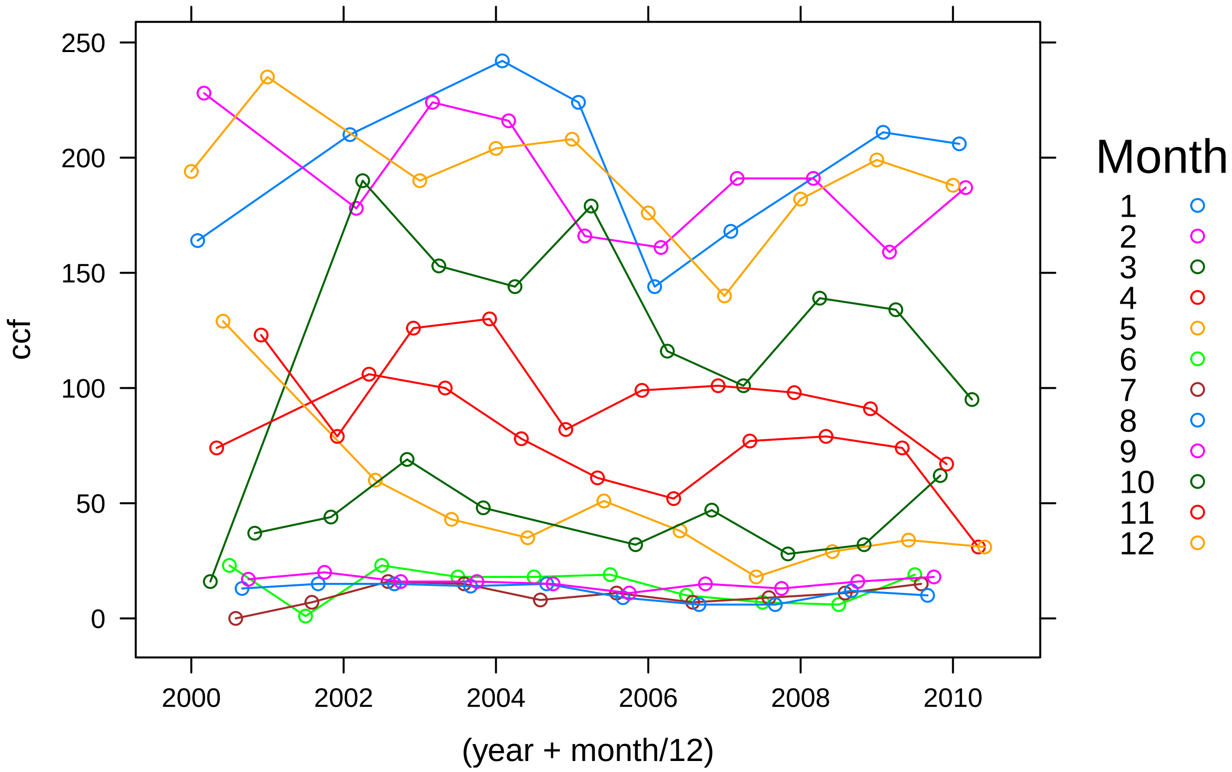


Part b.

```r
xyplot(ccf~(year+month/12),utilities,group=month,type=c("p","l"),auto.key=T)
```

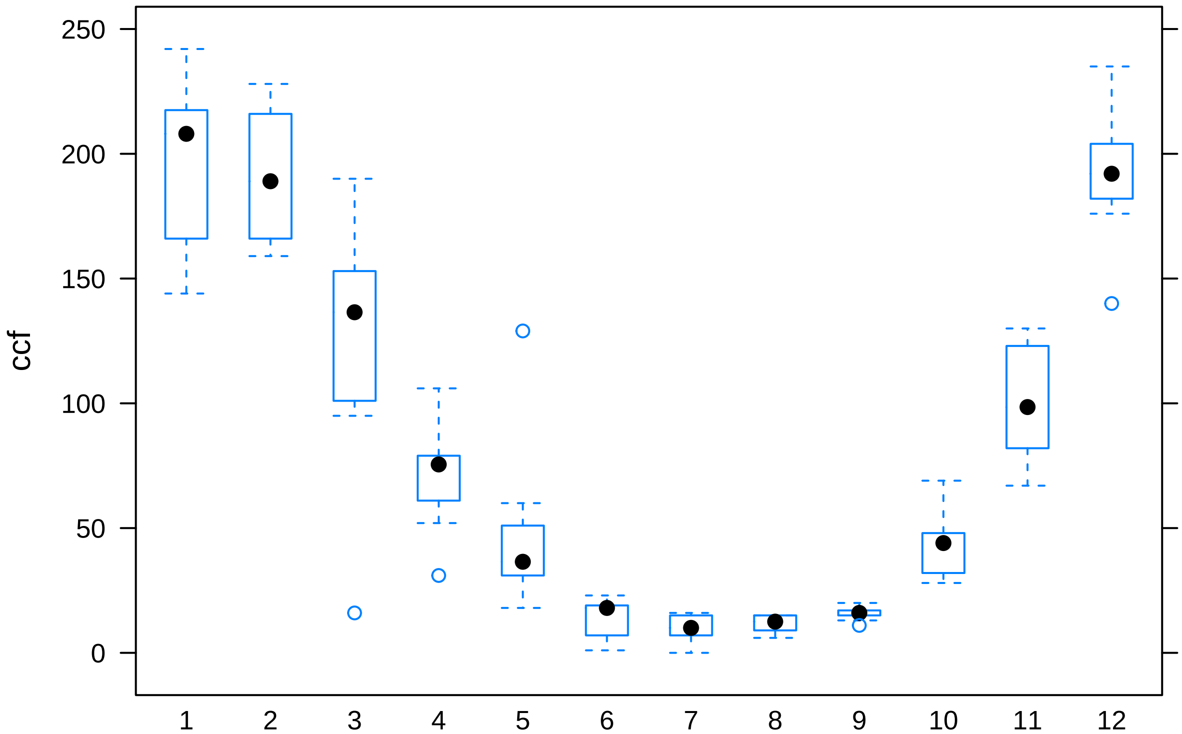



```
xyplot(ccf~(year+month/12),utilities,group=month,type=c("p","l"),auto.key=list(title="Month", space = "
```



Part c.

```
bwplot(ccf~factor(month),utilities)
```



Part d. The summer months have a low usage and little variation from year to year. The plot in part b and c helped with this observation. Month 3 had a bad reading; it does not appear that usage has changed over time except in month 3 where there is a downward trend.

Problem 1.25 Let's explore the data.

```
head(births78)
```

```
##     date births dayofyear
## 1 1/1/78   7701         1
## 2 1/2/78   7527         2
## 3 1/3/78   8825         3
## 4 1/4/78   8859         4
## 5 1/5/78   9043         5
## 6 1/6/78   9208         6
```

```
summary(births78)
```

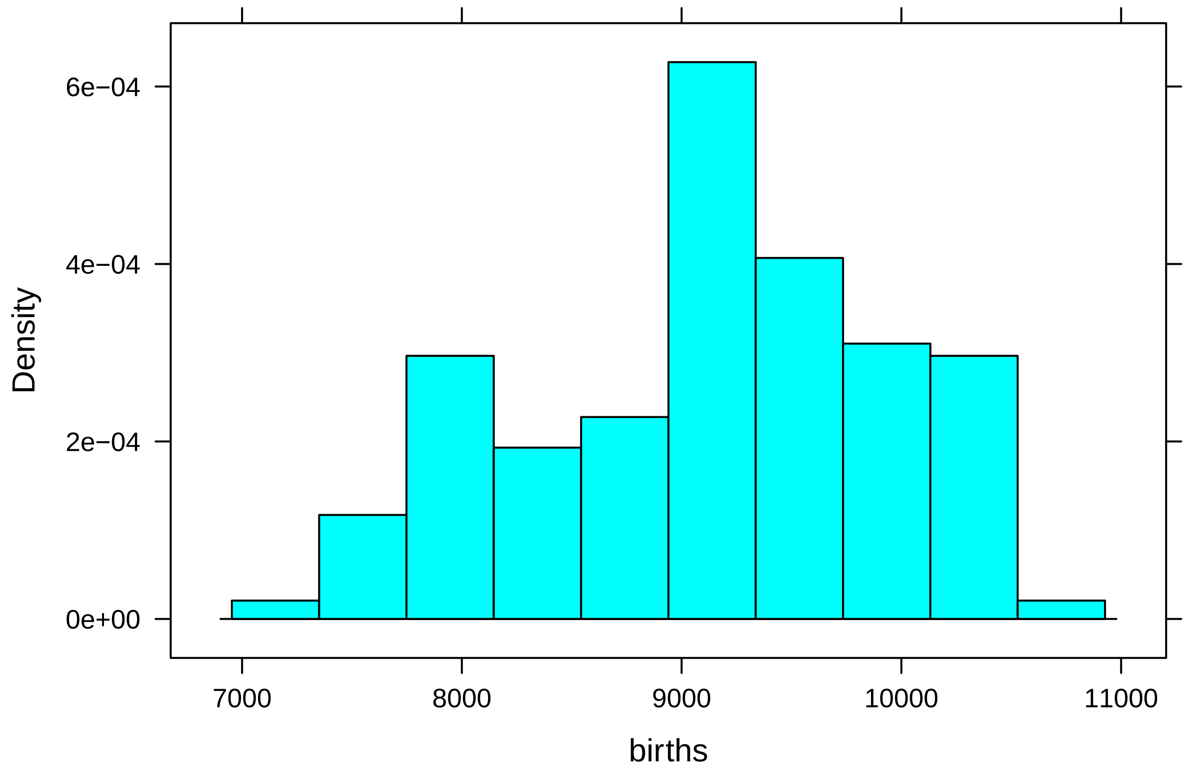
```
##        date          births          dayofyear
##  1/1/78 :  1   Min.   : 7135   Min.   :  1
##  1/10/78:  1   1st Qu.: 8554   1st Qu.: 92
##  1/11/78:  1   Median : 9218   Median :183
##  1/12/78:  1   Mean   : 9132   Mean   :183
##  1/13/78:  1   3rd Qu.: 9705   3rd Qu.:274
##  1/14/78:  1   Max.   :10711   Max.   :365
##  (Other):359
```

```r
str(births78)
```

```
## 'data.frame':    365 obs. of  3 variables:
##  $ date     : Factor w/ 365 levels "1/1/78","1/10/78",..: 1 12 23 26 27 28 29 30 31 2 ...
##  $ births   : int  7701 7527 8825 8859 9043 9208 8084 7611 9172 9089 ...
##  $ dayofyear: int  1 2 3 4 5 6 7 8 9 10 ...
```

Part a.

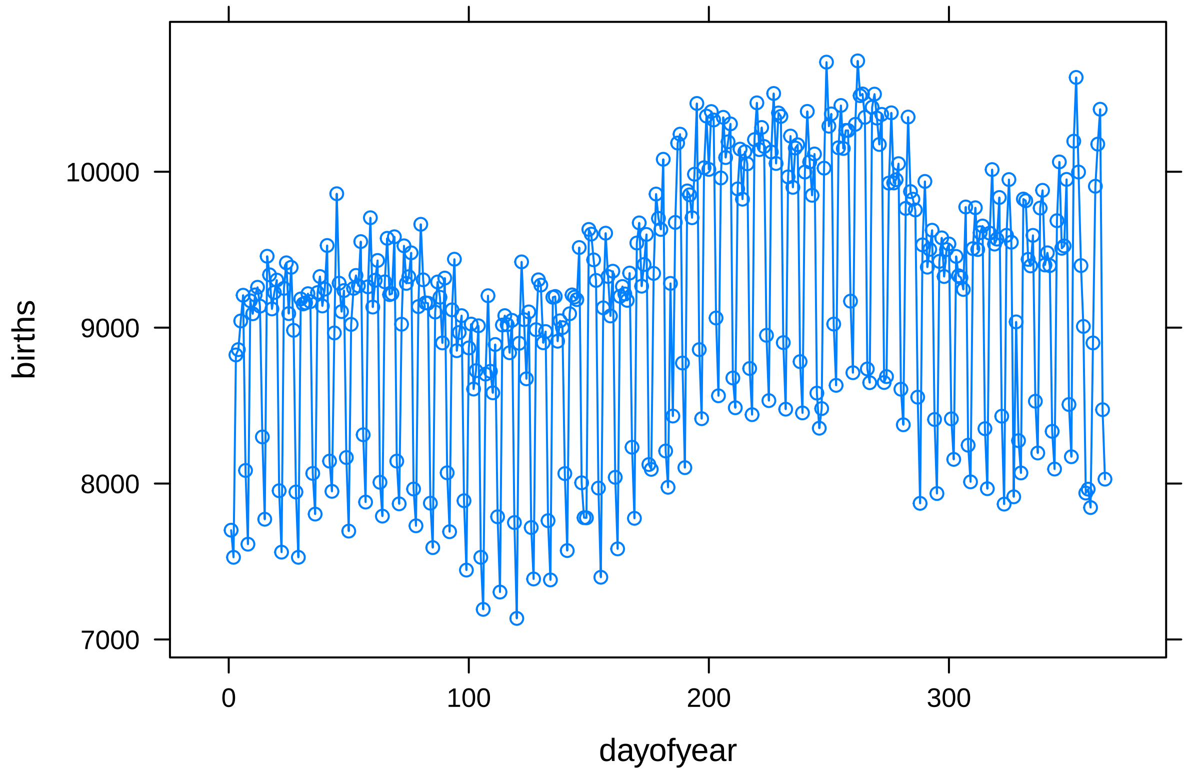
```r
histogram(~births,births78)
```



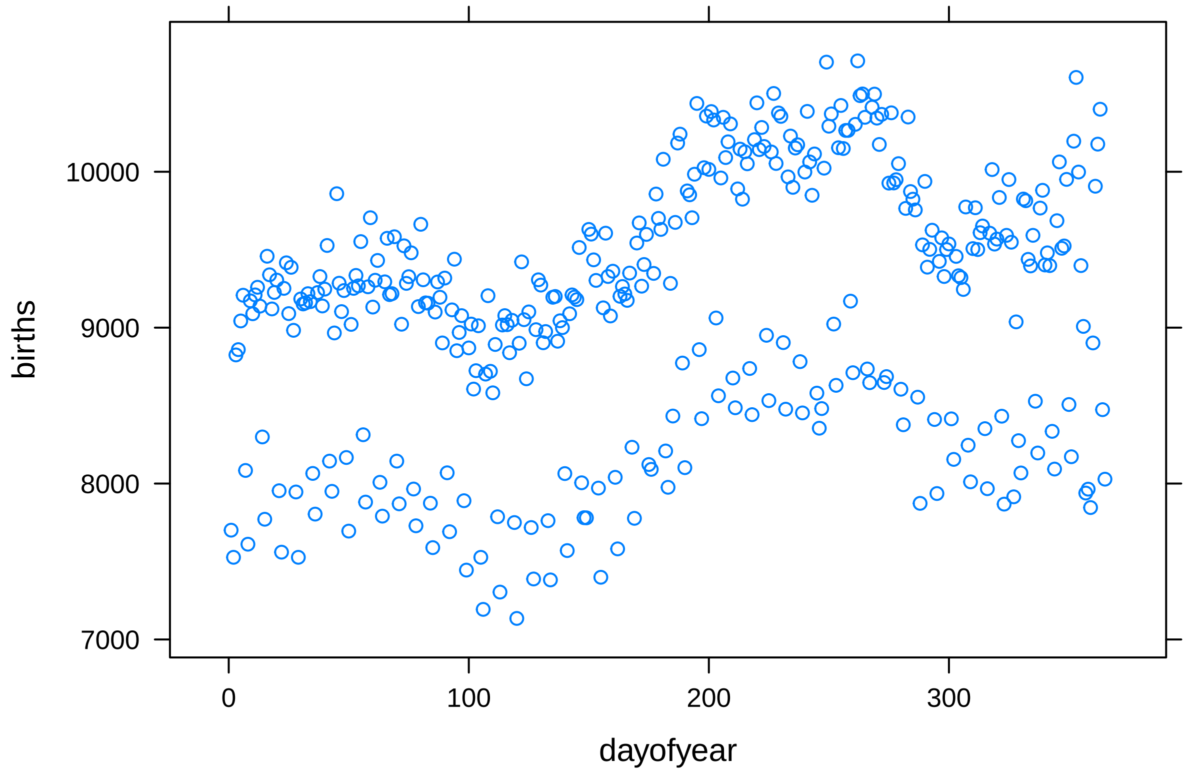
The births are not uniform. You can't tell from the histogram what time of the year more births occur, only that it is not consistent.

Part b.

```r
xyplot(births~dayofyear,births78,type=c("p","l"))
```



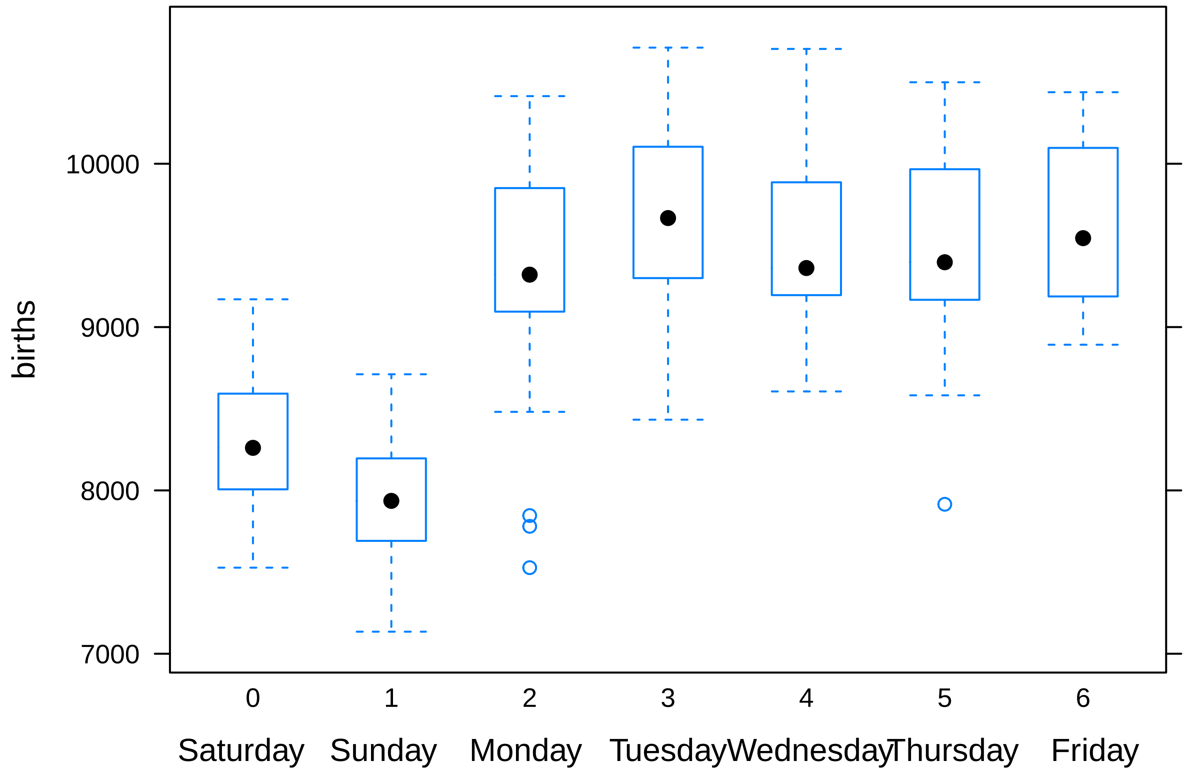

```
xyplot(births~dayofyear,births78)
```



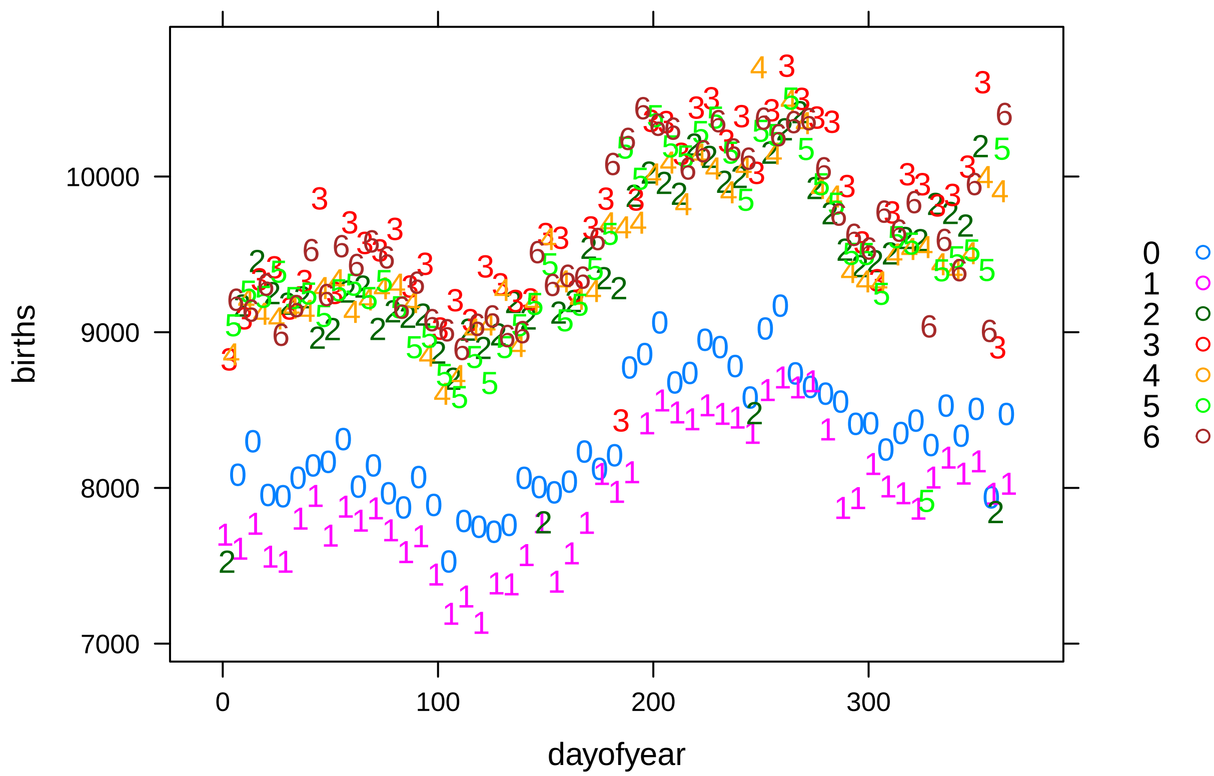
There are seasonal trends as well as two different bands. The bands probably correspond to weekend and/or holidays where there are a lower number of births. You also tend to get more births around 200 to 270 days into the year. This is roughly nine months after the previous holiday season where the weather is cold and people may be more in a festive mood.

Part c.
January 1, 1978 was a Sunday, I looked it up on line. Thus when I divide day of the year by 7 the remainder will tell me the day of the week; 1 will be a Sunday and 0 a Saturday. Thus I will use the group option to group the data by day of the week. The seasonal trends are observed on the time series plot.

```
bwplot(births~factor(I(dayofyear%%7)),births78,
       xlab=c("Saturday","Sunday","Monday","Tuesday","Wednesday","Thursday","Friday"))
```




```
xyplot(births~dayofyear,births78,group=dayofyear%%7,auto.key=list(space="right"),
       pch=c("0","1","2","3","4","5","6"),cex=1.5)
```




```r
summary(births~dayofyear%%7,data=births78)
```

```
## births    N=365
##
## +------------+-+---+--------+
## |            | |N  |births  |
## +------------+-+---+--------+
## |dayofyear%%7|0| 52|8309.327|
## |            |1| 53|7950.943|
## |            |2| 52|9371.327|
## |            |3| 52|9708.808|
## |            |4| 52|9498.019|
## |            |5| 52|9483.635|
## |            |6| 52|9625.788|
## +------------+-+---+--------+
## |Overall     | |365|9132.162|
## +------------+-+---+--------+
```

```r
summary(births~dayofyear%%7,data=births78,fun=favstats)
```

```
## births    N=365
##
## +------------+-+---+----+-------+-------+--------+-----+--------+--------+---+--------+
## |            | |N  |min |Q1     |median |Q3      |max  |mean    |sd      |n  |missing |
```

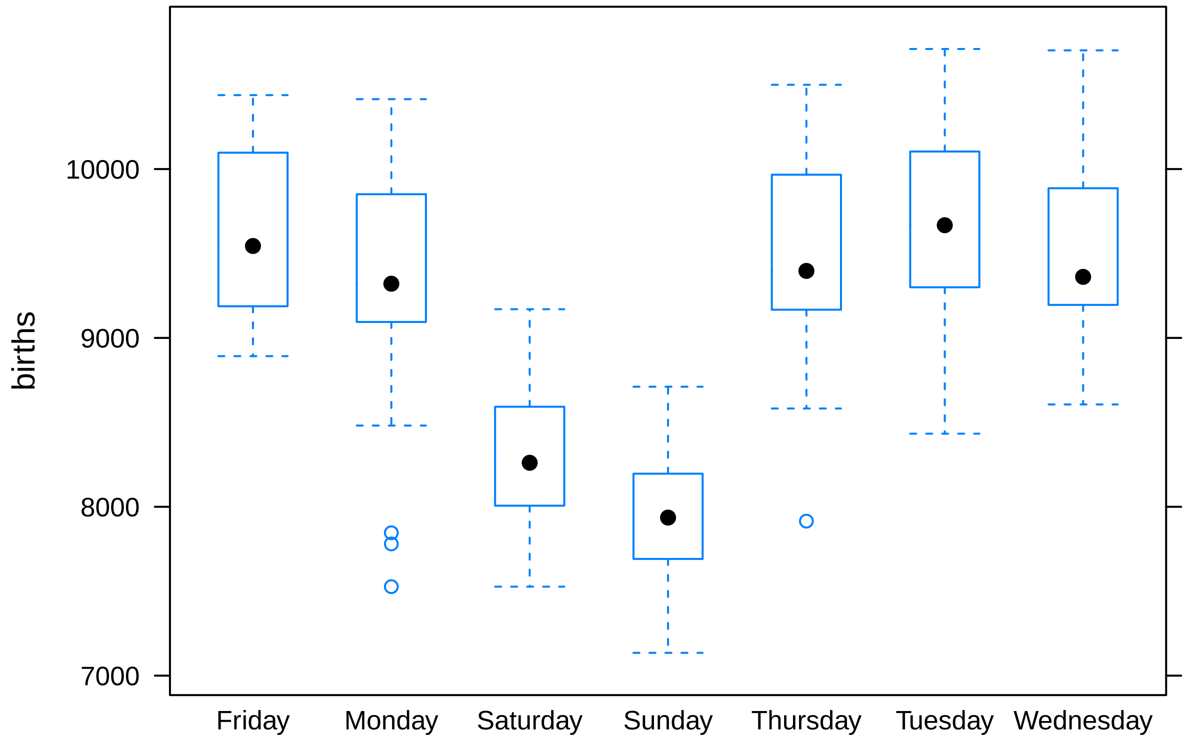
```
## +------------+-+---+----+-------+-------+--------+-----+--------+--------+---+--------+
## |dayofyear%%7|0| 52|7527|8007.25|8260.5 | 8586.25| 9170|8309.327|390.2555| 52|0       |
## |            |1| 53|7135|7691.00|7936.0 | 8196.00| 8711|7950.943|410.4367| 53|0       |
## |            |2| 52|7527|9097.25|9321.0 | 9838.00|10414|9371.327|608.3338| 52|0       |
## |            |3| 52|8433|9303.50|9667.5 |10083.50|10711|9708.808|526.5163| 52|0       |
## |            |4| 52|8606|9195.75|9361.5 | 9879.75|10703|9498.019|461.4187| 52|0       |
## |            |5| 52|7915|9171.00|9397.0 | 9957.75|10499|9483.635|551.0792| 52|0       |
## |            |6| 52|8892|9197.75|9544.5 |10088.50|10438|9625.788|487.6689| 52|0       |
## +------------+-+---+----+-------+-------+--------+-----+--------+--------+---+--------+
## |Overall     | |365|7135|8554.00|9218.0 | 9705.00|10711|9132.162|817.8821|365|0       |
## +------------+-+---+----+-------+-------+--------+-----+--------+--------+---+--------+
```

Another way to create the days of the week is to use the `as.Date` function. I will save the data to a temporary object and augment with the day of the week

```r
my_births78<-births78
my_births78$day_of_week<-weekdays(as.Date(births78$date,format = "%m/%d/%y"))
head(my_births78)
```

```
##     date births dayofyear day_of_week
## 1 1/1/78   7701         1      Sunday
## 2 1/2/78   7527         2      Monday
## 3 1/3/78   8825         3     Tuesday
## 4 1/4/78   8859         4   Wednesday
## 5 1/5/78   9043         5    Thursday
## 6 1/6/78   9208         6      Friday
```

```r
bwplot(births~day_of_week,my_births78)
```

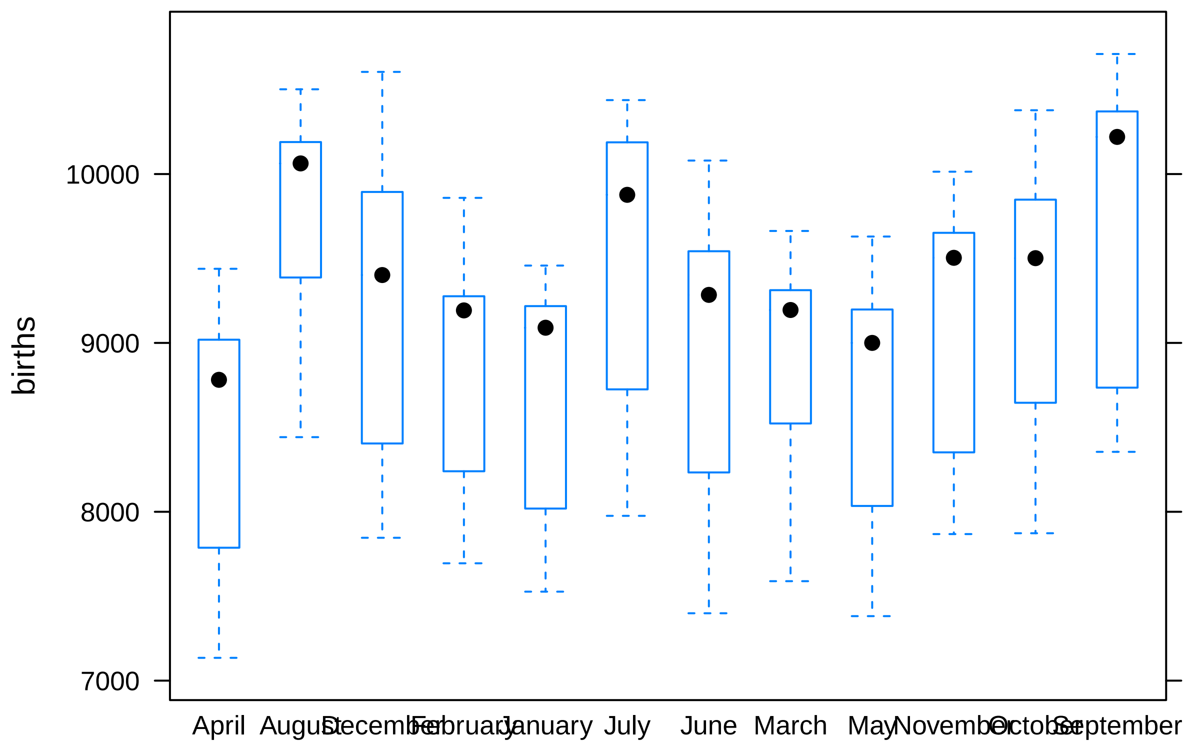


Now we can do the same thing with the month.

```
my_births78$month<-months(as.Date(births78$date,format = "%m/%d/%y"))
head(my_births78)
```

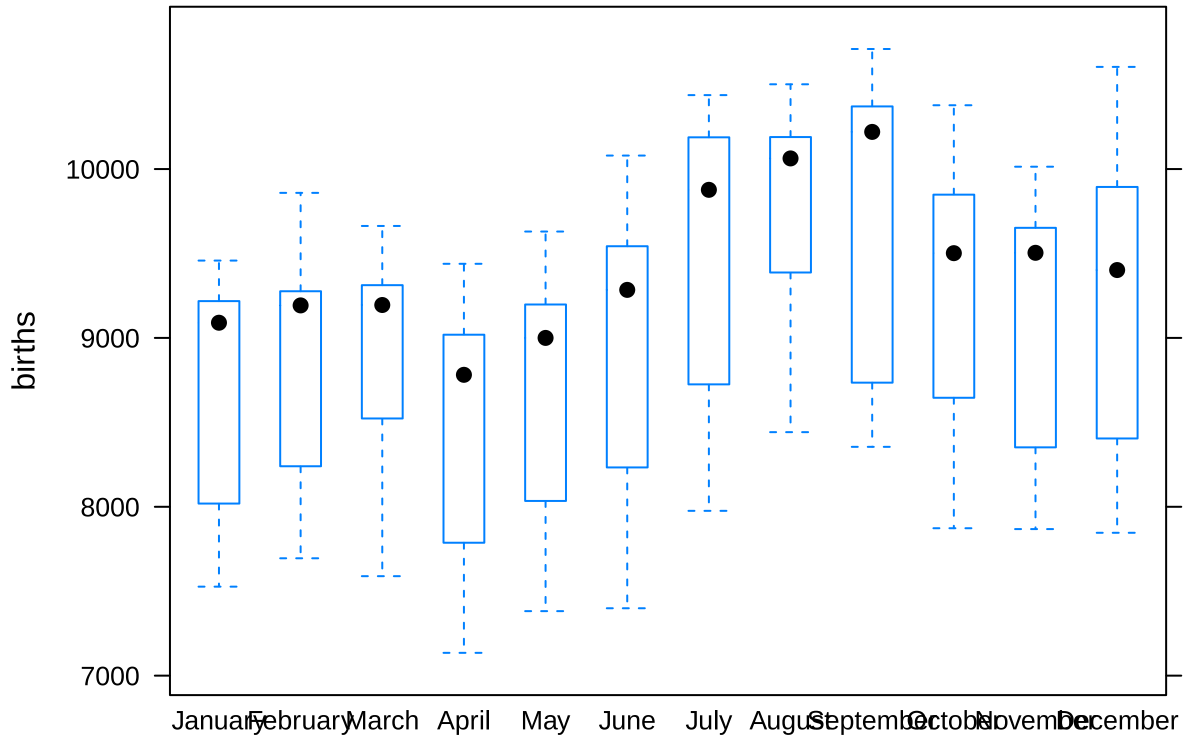
```
##     date births dayofyear day_of_week   month
## 1 1/1/78   7701         1      Sunday January
## 2 1/2/78   7527         2      Monday January
## 3 1/3/78   8825         3     Tuesday January
## 4 1/4/78   8859         4   Wednesday January
## 5 1/5/78   9043         5    Thursday January
## 6 1/6/78   9208         6      Friday January
```

```
bwplot(births~month,my_births78)
```



We may not like the alphabetic ordering on this plot. We can take care of this by telling R how to convert the character to a factor.

```
my_births78$month<-factor(my_births78$month,levels=c("January","February","March","April","May","June",
bwplot(births~month,my_births78)
```



These plots confirm my ideas about births.

Chapter 2

Section 2.1 and Appendix B

First I will load the libraries needed.

```
require('fastR')
require("Hmisc")
library(MASS)
```

Problem 2.1 Part a. $S = \{HHH, HHT, HTH, THH, THT, TTH, TTT\}$
b. $A = \{TTT, HTT, THT, TTH\}$,$B = \{TTT, HTT\}$,$C = \{TTT, TTH, THT, THH\}$
c. $A^c = \{HHH, HHT, HTH, THH\}$,$A \cap B = \{TTT, HTT\}$,$A \cup C = \{TTT, HTT, THT, TTH, THH\}$

Problem 2.2 First create a data object in R.

```
Prob2.2=data.frame(red=rep(1:6,each=6),blue=rep(1:6,times=6));Prob2.2
```

```
##    red blue
## 1    1    1
```

```
## 2    1    2
## 3    1    3
## 4    1    4
## 5    1    5
## 6    1    6
## 7    2    1
## 8    2    2
## 9    2    3
## 10   2    4
## 11   2    5
## 12   2    6
## 13   3    1
## 14   3    2
## 15   3    3
## 16   3    4
## 17   3    5
## 18   3    6
## 19   4    1
## 20   4    2
## 21   4    3
## 22   4    4
## 23   4    5
## 24   4    6
## 25   5    1
## 26   5    2
## 27   5    3
## 28   5    4
## 29   5    5
## 30   5    6
## 31   6    1
## 32   6    2
## 33   6    3
## 34   6    4
## 35   6    5
## 36   6    6
```

This can be used as the sample space where the first number is from red die and the second from the blue. b. Now using R we will find the events

```
Prob2.2[with(Prob2.2,red+blue>8),]
```

```
##     red blue
## 18   3    6
## 23   4    5
## 24   4    6
## 28   5    4
## 29   5    5
## 30   5    6
## 33   6    3
## 34   6    4
## 35   6    5
## 36   6    6
```

```r
Prob2.2[with(Prob2.2,red<blue),]
```

```
##    red blue
## 2    1    2
## 3    1    3
## 4    1    4
## 5    1    5
## 6    1    6
## 9    2    3
## 10   2    4
## 11   2    5
## 12   2    6
## 16   3    4
## 17   3    5
## 18   3    6
## 23   4    5
## 24   4    6
## 30   5    6
```

```r
Prob2.2[with(Prob2.2,blue==5),]
```

```
##    red blue
## 5    1    5
## 11   2    5
## 17   3    5
## 23   4    5
## 29   5    5
## 35   6    5
```

or

```r
A <- Prob2.2[with(Prob2.2,red+blue>=9),];A
```

```
##    red blue
## 18   3    6
## 23   4    5
## 24   4    6
## 28   5    4
## 29   5    5
## 30   5    6
## 33   6    3
## 34   6    4
## 35   6    5
## 36   6    6
```

```r
B <- Prob2.2[with(Prob2.2,blue>red),];B
```

```
##    red blue
## 2    1    2
## 3    1    3
## 4    1    4
```

```
## 5    1    5
## 6    1    6
## 9    2    3
## 10   2    4
## 11   2    5
## 12   2    6
## 16   3    4
## 17   3    5
## 18   3    6
## 23   4    5
## 24   4    6
## 30   5    6
```

```r
C <- Prob2.2[with(Prob2.2,blue==5),];C
```

```
##     red blue
## 5    1    5
## 11   2    5
## 17   3    5
## 23   4    5
## 29   5    5
## 35   6    5
```

c.

```r
Prob2.2[Prob2.2$red+Prob2.2$blue>8&Prob2.2$red<Prob2.2$blue,]
```

```
##     red blue
## 18   3    6
## 23   4    5
## 24   4    6
## 30   5    6
```

```r
Prob2.2[Prob2.2$red<Prob2.2$blue|Prob2.2$blue==5,]
```

```
##     red blue
## 2    1    2
## 3    1    3
## 4    1    4
## 5    1    5
## 6    1    6
## 9    2    3
## 10   2    4
## 11   2    5
## 12   2    6
## 16   3    4
## 17   3    5
## 18   3    6
## 23   4    5
## 24   4    6
## 29   5    5
## 30   5    6
## 35   6    5
```

```r
Prob2.2[Prob2.2$red+Prob2.2$blue>8&(Prob2.2$red<Prob2.2$blue|Prob2.2$blue==5),]
```

```
##    red blue
## 18   3    6
## 23   4    5
## 24   4    6
## 29   5    5
## 30   5    6
## 35   6    5
```

or

```r
Prob2.2[intersect(rownames(B),rownames(A)),]
```

```
##    red blue
## 18   3    6
## 23   4    5
## 24   4    6
## 30   5    6
```

```r
Prob2.2[union(rownames(B),rownames(C)),]
```

```
##    red blue
## 2    1    2
## 3    1    3
## 4    1    4
## 5    1    5
## 6    1    6
## 9    2    3
## 10   2    4
## 11   2    5
## 12   2    6
## 16   3    4
## 17   3    5
## 18   3    6
## 23   4    5
## 24   4    6
## 30   5    6
## 29   5    5
## 35   6    5
```

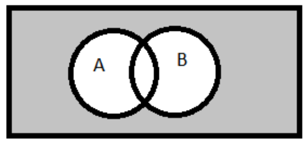
```r
Prob2.2[intersect(rownames(A),(union(rownames(B),rownames(C)))),]
```

```
##    red blue
## 18   3    6
## 23   4    5
## 24   4    6
## 29   5    5
## 30   5    6
## 35   6    5
```

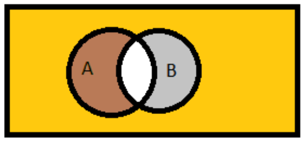
Problem B.1 $A \subseteq B$ When A is a subset of B, then every element of A is in B. Thus when we take the intersection of the two sets, we will get all the elements of A.

Problem B.2 $B \subseteq A$ Similar to Problem B.1 except it is the union, so we want A to be a superset of B.

Problem B.3 $A = B$ Two sets are equal when they are subsets of each other. Thus for the intersection to equal the union, they must have the same elements.



Problem B.4 First $(A \cup B)^c$



Now $(A^c \cap B^c)$
The union of A and B is everything inside both circles so its complement is everything outside of both circles. The complement of A is everything outside A and the complement and B is everything outside of B. The intersection of these is colored yellow in the second figure. Since the yellow and grey areas match, they are equal.

Problem B.6 It is the positive difference between a and b or it is 0.

Problem B.19

a.

```
B4x=c(0,1,2,3,4)
B4fx=c(1/6,1/3,1/4,1/6,1/12)
sum(B4fx)
```

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

b.

```
sum(B4x*B4fx)
```

```
## [1] 1.666667
```

```
fractions(sum(B4x*B4fx))
```

```
## [1] 5/3
```

c.

```r
sum(B4x^2*B4fx)
```

```
## [1] 4.166667
```

```r
fractions(sum(B4x^2*B4fx))
```

```
## [1] 25/6
```

Section 2.2 - 2.2.6

First I will load the libraries needed.

```r
library('fastR')
library("Hmisc")
```

Problem 2.6 In the denominator we are selecting 5 cards from 52 total where order does not matter and we sample without replacement. In the numerator, we first need to pick a card value and then from the 4 cards with that value we need to select 3. Then we pick the second face value from the remaining twelve and pick two cards from the 4.

```r
choose(13,1)*choose(4,3)*choose(12,1)*choose(4,2)/choose(52,5)
```

```
## [1] 0.001440576
```

Problem 2.7 This is a little more difficult problem. The denominator is the same as the previous problem. For the numerator, we select the two face values for the pairs and then pick two from each. Then we pick the remaining card from the 44, we could have done 44 choose 1 if we wanted. You don't want to do 13 choose 1 and then 12 choose 1 as this assumes order matters in that JJKK is different from KKJJ.

```r
choose(13,2)*choose(4,2)*choose(4,2)*choose(11,1)*choose(4,1)/choose(52,5)
```

```
## [1] 0.04753902
```

Problem 2.9 I will break it down for each. Start with the value the book has filled in to ensure we are doing the counting correctly.
1 suit

```r
choose(4,1)*choose(13,5)/choose(52,5)
```

```
## [1] 0.001980792
```

This matches the book. The zero suits is easy, it is just 0. Now 2 suits, we have to worry about whether the suits have 4 and 1 card or 3 and 2.

```r
choose(4,1) * choose(3,1) *choose(13,1)*choose(13,4)/ choose(52,5)+
choose(4,1)*choose(3,1)*choose(13,3)*choose(13,2)/ choose(52,5)
```

```
## [1] 0.1459184
```

Three suits is more difficult so let's do four suits.

```r
choose(4,1)*choose(13,2)*choose(13,1)*choose(13,1)*choose(13,1)/choose(52,5)
```

```
## [1] 0.2637455
```

For 3 suits we have to worry about having three cards of one suit with the remaining two suits of one card or two suits with two cards each and the third suit with one care.

```r
choose(4,2)*choose(13,2)*choose(13,2)*choose(2,1)*choose(13,1)/
choose(52,5)+choose(4,1)*choose(13,3)*choose(3,2)*choose(13,1)*choose(13,1)/choose(52,5)
```

```
## [1] 0.5883553
```

Or using complements, the easier way.

```r
1- 0.1459184- 0.2637455-0.001980792
```

```
## [1] 0.5883553
```

Problem 2.10 Part a.

Find the probability no one has the same birthday and then use complements to find the probability desired.

```r
1-prod(seq(356,365))/365^10
```

```
## [1] 0.1169482
```

Part b.
We want $1 - \prod_{i=1}^{n}(\frac{366-i}{365}) \geq 0.5$. I will write a function to make this easier.

```r
birthday=function(n){1-prod(seq(366-n,365))/365^n}
birthday(10)
```

```
## [1] 0.1169482
```

```r
cbind(people=15:30,prob=sapply(15:30,birthday))
```

```
##       people      prob
##  [1,]     15 0.2529013
##  [2,]     16 0.2836040
##  [3,]     17 0.3150077
##  [4,]     18 0.3469114
```

```
##  [5,]      19 0.3791185
##  [6,]      20 0.4114384
##  [7,]      21 0.4436883
##  [8,]      22 0.4756953
##  [9,]      23 0.5072972
## [10,]      24 0.5383443
## [11,]      25 0.5686997
## [12,]      26 0.5982408
## [13,]      27 0.6268593
## [14,]      28 0.6544615
## [15,]      29 0.6809685
## [16,]      30 0.7063162
```

Note: Exactly two people share the same birthday would be choose(n,2)*(365/365)*(1/365)*(364/365)...(365-(n-2))/365. This does not include more than 1 pair having the same birthday.

Problem 2.11 Part a.

I will scan my work in as a separate document. It is a lot of writing.

Part b.
Test code first.

```
str(births78)
```

```
## 'data.frame':    365 obs. of  3 variables:
##  $ date     : Factor w/ 365 levels "1/1/78","1/10/78",..: 1 12 23 26 27 28 29 30 31 2 ...
##  $ births   : int  7701 7527 8825 8859 9043 9208 8084 7611 9172 9089 ...
##  $ dayofyear: int  1 2 3 4 5 6 7 8 9 10 ...
```

```
sample(births78$dayofyear,2,replace=TRUE,prob=births78$births/sum(births78$births))
```

```
## [1] 337 255
```

```
length(unique(sample(births78$dayofyear,20,replace=TRUE,prob=births78$births/sum(births78$births))))
```

```
## [1] 18
```

```
replicate(10,length(unique(sample(births78$dayofyear,20,
  replace=TRUE,prob=births78$births/sum(births78$births)))))
```

```
##  [1] 19 19 20 18 18 20 19 20 20 19
```

```
replicate(10,length(unique(sample(births78$dayofyear,20,
  replace=TRUE,prob=births78$births/sum(births78$births))))<20)
```

```
##  [1] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE
```

```
sum(replicate(10,length(unique(sample(births78$dayofyear,20,
  replace=TRUE,prob=births78$births/sum(births78$births))))<20))
```

```
## [1] 2
```

I now have all the code I need to write a function.

```
birthday78=function(n,r){
  sum(replicate(r,length(unique(sample(births78$dayofyear,n,
    replace=TRUE,prob=births78$births/sum(births78$births))))<n))/r
}
```

Now we can calculate some probabilities.

```
birthday78(15,10000)
```

```
## [1] 0.2474
```

```
birthday78(20,10000)
```

```
## [1] 0.4122
```

```
birthday78(25,10000)
```

```
## [1] 0.5816
```

Problem 2.29 The key is to divide by the product of permutations of same letters.

```
factorial(10)/(factorial(3)*factorial(3)*factorial(2))
```

```
## [1] 50400
```

Section 2.2.7

First I will load the libraries needed.

```
library('fastR')
library("Hmisc")
library("MASS")
```

Problem 2.14 See additional pdf file as this problem has a picture.

Problem 2.15 From the definition of the probability of union, we have

$$P(A \cap B) = P(A) + P(B) - P(A \cup B)$$

but

$$P(A \cup B) \leq 1$$

so

$$P(A) + P(B) - P(A \cup B) \geq P(A) + P(B) - 1$$

thus

$$P(A \cap B) \geq P(A) + P(B) - 1$$

Problem 2.18 Setup the matrix in R

```
Prob2.18<-matrix(c(2,1,6,9),nrow=2,dimnames=list(Assembly_Line=c("One","Two"),Status=c("Bad","Good")));I
```

```
##              Status
## Assembly_Line Bad Good
##           One   2    6
##           Two   1    9
```

Part a.

```
sum(Prob2.18[,1])/sum(Prob2.18)
```

```
## [1] 0.1666667
```

```
fractions(sum(Prob2.18[,1])/sum(Prob2.18))
```

```
## [1] 1/6
```

Part b.

```
sum(Prob2.18[1,1])/sum(Prob2.18[1,])
```

```
## [1] 0.25
```

```
fractions(Prob2.18[1,1])/sum(Prob2.18[1,])
```

```
## [1] 1/4
```

Part c.

```
sum(Prob2.18[1,1])/sum(Prob2.18[,1])
```

```
## [1] 0.6666667
```

```
fractions(Prob2.18[1,1])/sum(Prob2.18[,1])
```

```
## [1] 2/3
```

Problem 2.21 See additional pdf file for solution.

Problem 2.24 Define the random variables as:
T = Test Result
C = Woman Carrier

Given $P(T = + \mid C = +) = .7$ and $P(T = - \mid C = -) = .9$. We want to find $P(C = + \mid T = +)$ first. To use Bayes Rule we need $P(C = +)$ which is given as 2/3. Notice that all of this is based on the given that the child has DMD; this makes the problem a little harder to understand.

$P(C = + \mid T = +) = \frac{P(T=+|C=+)P(C=+)}{P(T=+|C=+)P(C=+)+P(T=+|C=-)P(C=-)}$

```r
((.7*2/3)/(.7*2/3+.1/3))
```

```
## [1] 0.9333333
```

```r
fractions(((.7*2/3)/(.7*2/3+.1/3)))
```

```
## [1] 14/15
```

and the second part is to find $P(C = + \mid T = -)$

```r
((.3*2/3)/(.3*2/3+.9/3))
```

```
## [1] 0.4
```

```r
fractions(((.3*2/3)/(.3*2/3+.9/3)))
```

```
## [1] 2/5
```

Problem 2.25 $P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$

Section 2.3

First I will load the libraries needed.

```r
library('fastR')
library("Hmisc")
library("MASS")
```

Problem 2.30 Part a. Ten percent of the 100 items are defective. The purchaser will reject the shipment if in a sample of 4, one or more is defective. Thus the probability of rejecting is 1 - P(no defective items)

```r
1-choose(90,4)/choose(100,4)
```

```
## [1] 0.3483695
```

Now even though the assumptions of a binomial don't fit, let's see what the estimate would be if we had tried to use the binomial. In this case we will count a success as selecting a defective item. In 4 trials, we want to select 0

```r
pbinom(0,4,.1,lower.tail=FALSE)
```

```
## [1] 0.3439
```

```r
1-dbinom(0,4,.1)
```

```
## [1] 0.3439
```

This is close because the probability of defective is small.

Next let's try the negative binomial just to see how close it will be. Let's pick a success as a defective; in this case we want at least three or fewer failures before a success. This will lead to rejection of the shipment. Think about this carefully.

```
pnbinom(3,1,.1)
```

```
## [1] 0.3439
```

Part b.
I need to write a function to generate this plot. I will be using the first answer in part a, but I want to use multiple values for the probability of being defective. Here is my function:

```
myreject<-function(n){
  1-choose(100-n,4)/choose(100,4)
}
```

Let's test it.

```
myreject(10)
```

```
## [1] 0.3483695
```

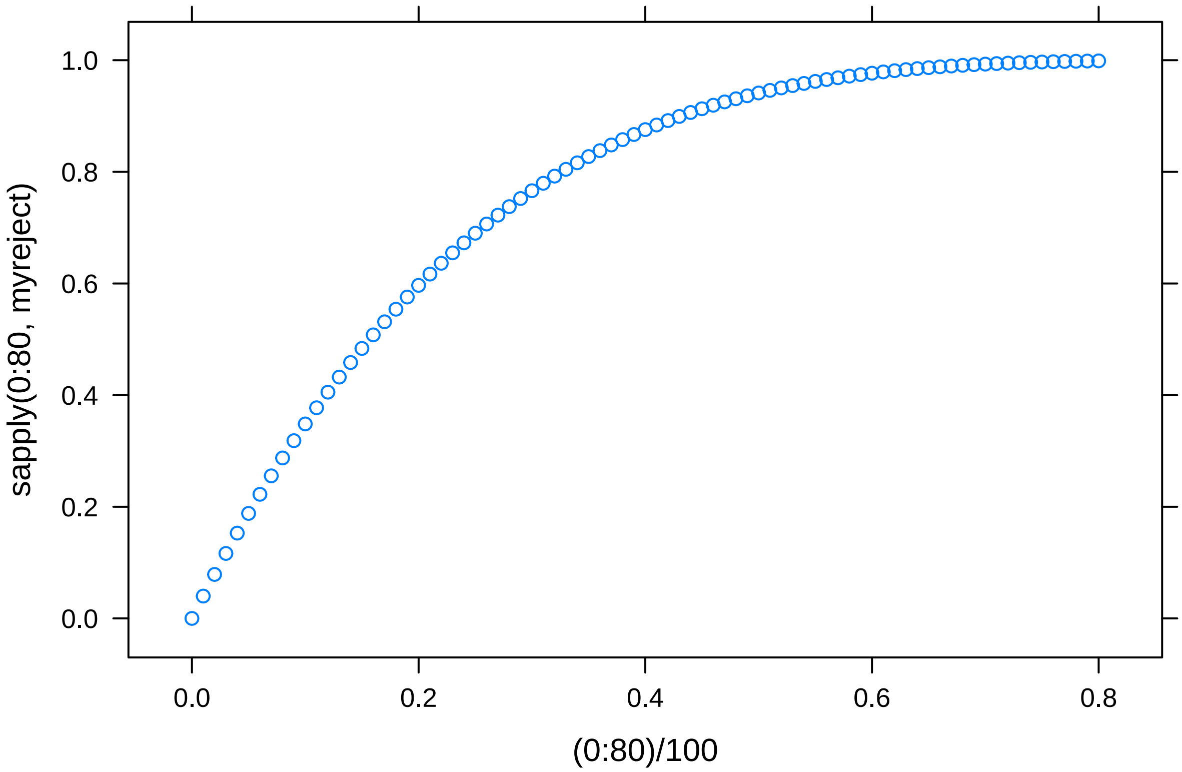
Now let's change the probability of being defective using a vector of values, we need the sapply function for this.

```
sapply(1:20,myreject)
```

```
##  [1] 0.04000000 0.07878788 0.11638837 0.15282597 0.18812488 0.22230910
##  [7] 0.25540233 0.28742804 0.31840943 0.34836945 0.37733081 0.40531594
## [13] 0.43234703 0.45844602 0.48363458 0.50793413 0.53136584 0.55395061
## [19] 0.57570912 0.59666176
```

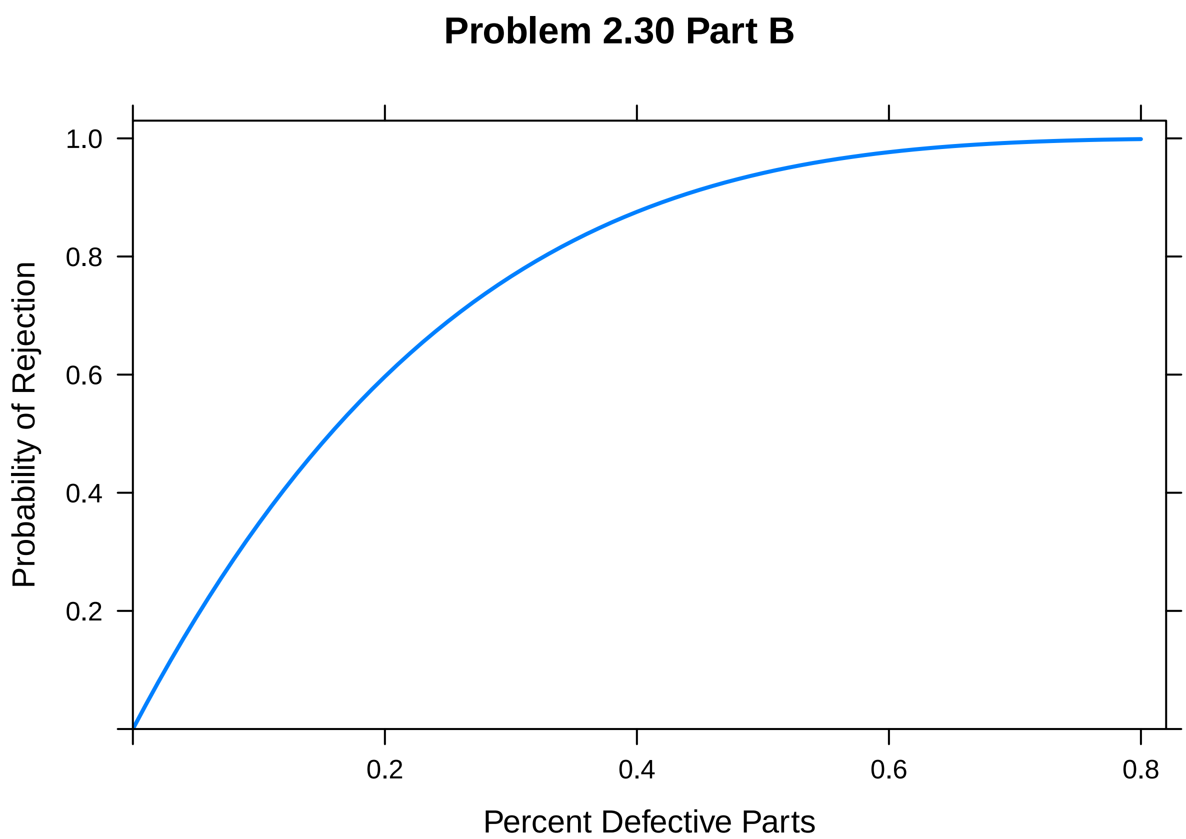
Everything seems to be in order, so let's plot

```
xyplot(sapply(0:80,myreject)~(0:80)/100)
```



Let's make it look professional. note in type the character is a lower case L for line.

```
xyplot(sapply(0:80,myreject)~(0:80)/100,type="l",
       xlab="Percent Defective Parts",ylab="Probability of Rejection",
       lwd=2,xlim=c(0,.82),ylim=c(0,1.03),main="Problem 2.30 Part B")
```



Problem 2.40 First we want the probability of at least one 6 in four rolls of a die. We will use the binomial for this.

```
pbinom(0,4,1/6,lower=FALSE)
```

```
## [1] 0.5177469
```

Or using a negative binomial where a success is rolling a 6.

```
pnbinom(3,1,1/6)
```

```
## [1] 0.5177469
```

This first game is slightly in favor of the person rolling the die.

For the second game, we have

```
pbinom(0,24,1/36,lower=FALSE)
```

```
## [1] 0.4914039
```

```
1-dbinom(24,24,35/36)
```

```
## [1] 0.4914039
```

or using the negative binomial

```
pnbinom(23,1,1/36)
```

```
## [1] 0.4914039
```

Problem 2.43 See pdf file with the worked out solution.

Problem 2.44 Part a. Need all 4 to be successful. Define X as the number of correct bits received out of 4. We want to find $P(X = 4)$.

```
dbinom(4,4,.95)
```

```
## [1] 0.8145062
```

Or let X be the number of correct bits until I get an incorrect bit. We want the $P(X > 3)$.

```
pnbinom(3,1,.05,lower=F)
```

```
## [1] 0.8145062
```

Part b. Now we have added bits so that if there are at most one incorrect bit, we can decode the transmission correctly. Thus define X as the number of correct bits out of 7. We want the $P(X > 5) = P(X \geq 6)$.

```
pbinom(5,7,.95,lower=F)
```

```
## [1] 0.9556195
```

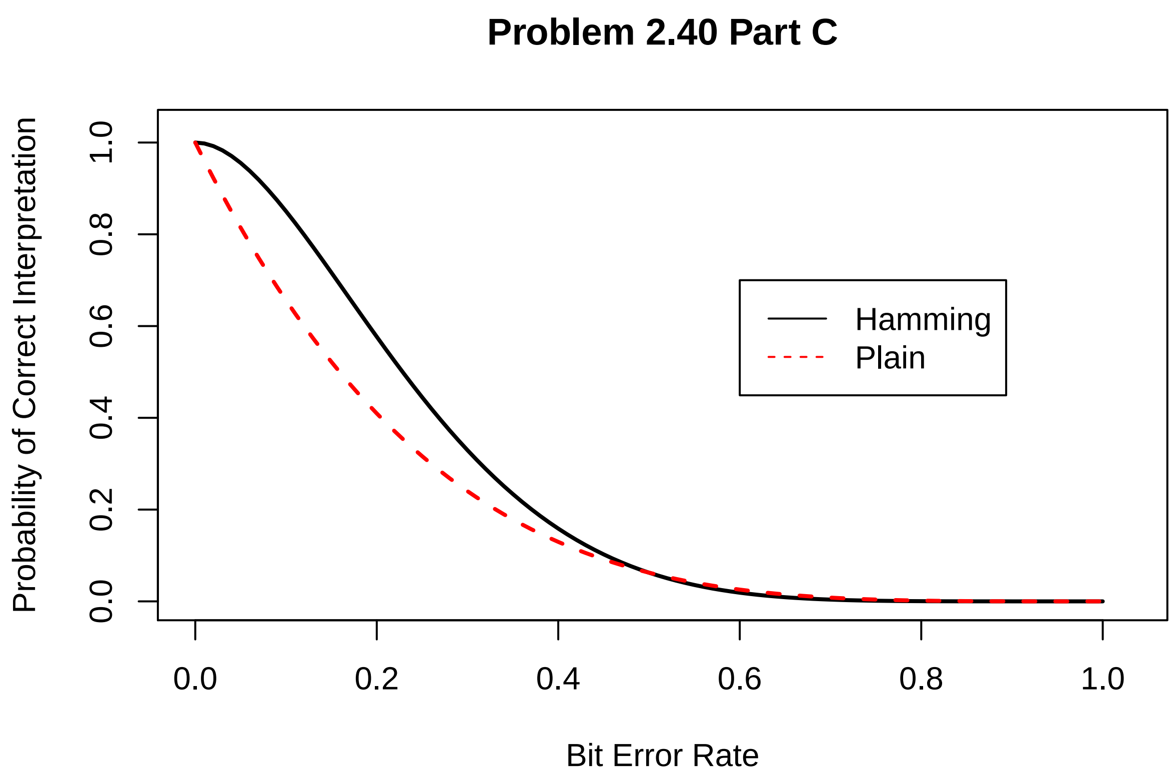
Or if we make a success an incorrect bit and Y = number of incorrect bits out of 7, then we want $P(Y \leq 1)$.

```
pbinom(1,7,.05)
```

```
## [1] 0.9556195
```

Part c.

```
Prob2.44=function(p){pbinom(5,7,1-p,low=F)}
plot((0:100)/100,sapply(seq(0,1,.01),Prob2.44),type="l",xlab="Bit Error Rate",ylab="Probability of Corr
points(seq(0,1,.01),dbinom(4,4,1-seq(0,1,.01)),type="l",lwd=2,col="red",lty=2)
legend(.6,.7,c("Hamming","Plain"),lty=c(1,2),col=c("black","red"))
```



Problem 2.45 This contains two binomial random variables, the first is the number of free throws made out of 10 trials and the second is the number of free throws made out of 20.

```
pbinom(8,10,.8,lower=F)
```

```
## [1] 0.3758096
```

```
pbinom(17,20,.8,lower=F)
```

```
## [1] 0.2060847
```

So Freddie has a better chance of make at least 9 out of 10 over at least 18 out of 20.

Problem 2.46 For this problem, define the random variable X as the number missed free throws until Freddie makes 10. The probability of success is .8.

Part a.

```
pnbinom(0,10,.8)
```

```
## [1] 0.1073742
```

Part b.

```r
pnbinom(4,10,.8,lower=F)
```

```
## [1] 0.1298396
```

Part c.

```r
pnbinom(4,10,.7,lower=F)
```

```
## [1] 0.4157988
```

Section 2.4

First I will load the libraries needed.

```r
library('fastR')
library("Hmisc")
```

Problem 2.51 The null hypothesis is that the population proportion for blue is .25. The alternative is that the population proportion is not 0.25. This is written as

$$H_0 : \pi = 0.25$$

$$H_a : \pi \neq 0.25$$

We need a significance level before we collect data. We will use 0.05, $\alpha = 0.05$.

The statistic is the number of times the spinner lands on blue out of 50 spins. Note that this statistic is a binomial random variable. For our problem the test statistic is 8.

To calculate a p-value we need the definition of a p-value; given that the null hypothesis is true, the p-value is the probability of observing the test statistics or more extreme. To do this by hand we need the probability of 8 or less given that the probability of success is 0.25. This is only the lower half of the contribution to the p-value.

```r
pbinom(8,50,.25)
```

```
## [1] 0.09159726
```

Next we need to know the upper half of the p-value, that is how many spins would need to be close to 50 to be considered unusual. We will experiment

```r
dbinom(8,50,.25)
```

```
## [1] 0.04634141
```

```r
cbind(signif((50:15),2),dbinom(50:15,50,.25))
```

```
##       [,1]         [,2]
##  [1,]   50 7.888609e-31
##  [2,]   49 1.183291e-28
##  [3,]   48 8.697191e-27
##  [4,]   47 4.174652e-25
##  [5,]   46 1.471565e-23
##  [6,]   45 4.061519e-22
##  [7,]   44 9.138417e-21
##  [8,]   43 1.723244e-19
##  [9,]   42 2.778732e-18
## [10,]   41 3.890224e-17
## [11,]   40 4.784976e-16
## [12,]   39 5.219974e-15
## [13,]   38 5.089474e-14
## [14,]   37 4.463077e-13
## [15,]   36 3.538583e-12
## [16,]   35 2.547780e-11
## [17,]   34 1.671980e-10
## [18,]   33 1.003188e-09
## [19,]   32 5.517535e-09
## [20,]   31 2.787807e-08
## [21,]   30 1.296330e-07
## [22,]   29 5.555702e-07
## [23,]   28 2.197028e-06
## [24,]   27 8.023927e-06
## [25,]   26 2.708075e-05
## [26,]   25 8.449195e-05
## [27,]   24 2.437268e-04
## [28,]   23 6.499381e-04
## [29,]   22 1.601633e-03
## [30,]   21 3.645096e-03
## [31,]   20 7.654701e-03
## [32,]   19 1.481555e-02
## [33,]   18 2.639020e-02
## [34,]   17 4.318396e-02
## [35,]   16 6.477595e-02
## [36,]   15 8.883558e-02
```

From this it looks like 17 or greater would be the values of interest since these all have a smaller probability than 8 success while 16 successes has a greater probability. The upper half of the contribution to the p-value would be

```r
pbinom(16,50,.25,lower=F)
```

```
## [1] 0.09830732
```

And the p-value is

```r
pbinom(8,50,.25)+pbinom(16,50,.25,lower=F)
```

```
## [1] 0.1899046
```

Using more sophisticated coding to do the same thing, based on the example in the book on page 62.

```r
sum(dbinom(0:50,50,0.25)
[dbinom(0:50,50,0.25) <= dbinom(8,50,0.25)])
```

```
## [1] 0.1899046
```

Running binom.test yields the same answer

```r
binom.test(8,50,.25)
```

```
##
##
##
## data:  8 out of 50
## number of successes = 8, number of trials = 50, p-value = 0.1899
## alternative hypothesis: true probability of success is not equal to 0.25
## 95 percent confidence interval:
##  0.07170077 0.29112631
## sample estimates:
## probability of success
##                   0.16
```

The conclusion is that based on the data if landing on blue occurred in 25% of all spins then the probability of 8 blues, or more extreme, out of 50 is 0.1899. Using a 5% significance level there is insufficient evidence to reject the hypothesis that the proportion of spins that land on blue is 0.25.

Problem 2.52 The null hypothesis is that the population proportion for Gus' use of the right paw is .50. The alternative is that the population proportion is not 0.50. This is written as

$$H_0 : \pi = 0.50$$

$$H_a : \pi \neq 0.50$$

We need a significance level before we collect data. We will use 0.05, $\alpha = 0.05$.

The statistic is the number of times Gus uses his right paw out of 10 attempts. Note that this statistic is a binomial random variable. For our problem the test statistic is 8.

To calculate a p-value use binom.test.

```r
binom.test(8,10,.5)
```

```
##
##
##
## data:  8 out of 10
## number of successes = 8, number of trials = 10, p-value = 0.1094
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.4439045 0.9747893
## sample estimates:
## probability of success
##                    0.8
```

The conclusion is that based on the data, assuming that Gus uses his right paw 50% of the time, then the probability of 8 uses of the right paw, or more extreme, out of 10 is 0.1094. Using a 5% significance level there is insufficient evidence to reject the hypothesis that Gus' use of his right paw is equal to the left paw.

Problem 2.53 There are several ways to answer this question but it is not a standard hypothesis test. The author of the book seems to be asking is the data too good, that is if green really occurred 3/4 of the time how likely is it that Mendel would get 428 green pea plants out of a sample of 580. Now if he got 435, this is exactly 3/4, then we would be really suspicious. One way to answer this question is to calculate the p-value, and if it is 1.0 or close to one, the data would be suspicious.

```
binom.test(428,580,.75)
```

```
##
##
##
## data:  428 out of 580
## number of successes = 428, number of trials = 580, p-value =
## 0.5022
## alternative hypothesis: true probability of success is not equal to 0.75
## 95 percent confidence interval:
##  0.7001255 0.7732932
## sample estimates:
## probability of success
##               0.737931
```

since the p-value is .5022, this does not raise any flags.

Another, equivalent, way to answer this question by finding the probability that the answer is extreme by finding the probability of having an observed value within 7 units of the hypothesized value of 435.

```
pbinom(442,580,.75)-pbinom(428,580,.75)
```

```
## [1] 0.4978299
```

This is a fairly high value, so based on 428 peas out of 580, I have no reason to believe that Mendel forged the data.

We could also simulate the experiment and determine how many times the number of green peas is within 7 of 435. This should give an answer close the value calculated from probability.

```r
temp=replicate(10000,sum(sample(c(1,0),580,replace=T,prob=c(.75,.25))))
sum(temp<=442&temp>=428)/10000
```

```
## [1] 0.535
```

Problem 2.55 Part a.
This problem is asking for us to find the power of the hypothesis test if the alternative is .55 and the null is .50. Let's assume we are using $\alpha = 0.05$, the problem statement leads us to this assumption by stating we will reject the null hypothesis if the p-value is less than 0.05. The first step is to find the critical values that will lead to rejection given the null hypothesis is true. For the lower end, we need the number of successes such that the probability of being less than or equal to that value is .025; that is I have split the .05 into two equal pieces.

```r
cbind(0:100,pbinom(0:100,200,.5))
```

```
##         [,1]         [,2]
##    [1,]    0 6.223015e-61
##    [2,]    1 1.250826e-58
##    [3,]    2 1.250888e-56
##    [4,]    3 8.298397e-55
##    [5,]    4 4.108338e-53
##    [6,]    5 1.619022e-51
##    [7,]    6 5.290204e-50
##    [8,]    7 1.474174e-48
##    [9,]    8 3.576236e-47
##   [10,]    9 7.672437e-46
##   [11,]   10 1.473854e-44
##   [12,]   11 2.560609e-43
##   [13,]   12 4.056888e-42
##   [14,]   13 5.902270e-41
##   [15,]   14 7.932088e-40
##   [16,]   15 9.897117e-39
##   [17,]   16 1.151611e-37
##   [18,]   17 1.254488e-36
##   [19,]   18 1.283765e-35
##   [20,]   19 1.237921e-34
##   [21,]   20 1.127930e-33
##   [22,]   21 9.734829e-33
##   [23,]   22 7.976368e-32
##   [24,]   23 6.217261e-31
##   [25,]   24 4.618699e-30
##   [26,]   25 3.275739e-29
##   [27,]   26 2.221524e-28
##   [28,]   27 1.442698e-27
##   [29,]   28 8.983927e-27
##   [30,]   29 5.371122e-26
##   [31,]   30 3.086568e-25
##   [32,]   31 1.706745e-24
##   [33,]   32 9.090400e-24
##   [34,]   33 4.667992e-23
##   [35,]   34 2.313108e-22
##   [36,]   35 1.106989e-21
```

```
## [37,]   36 5.120512e-21
## [38,]   37 2.291018e-20
## [39,]   38 9.921850e-20
## [40,]   39 4.161915e-19
## [41,]   40 1.692008e-18
## [42,]   41 6.670804e-18
## [43,]   42 2.551910e-17
## [44,]   43 9.477564e-17
## [45,]   44 3.418956e-16
## [46,]   45 1.198578e-15
## [47,]   46 4.085225e-15
## [48,]   47 1.354360e-14
## [49,]   48 4.369218e-14
## [50,]   49 1.372143e-13
## [51,]   50 4.196510e-13
## [52,]   51 1.250347e-12
## [53,]   52 3.630612e-12
## [54,]   53 1.027739e-11
## [55,]   54 2.837139e-11
## [56,]   55 7.640274e-11
## [57,]   56 2.007696e-10
## [58,]   57 5.149597e-10
## [59,]   58 1.289601e-09
## [60,]   59 3.153991e-09
## [61,]   60 7.535308e-09
## [62,]   61 1.759079e-08
## [63,]   62 4.013453e-08
## [64,]   63 8.951606e-08
## [65,]   64 1.952234e-07
## [66,]   65 4.163957e-07
## [67,]   66 8.687935e-07
## [68,]   67 1.773589e-06
## [69,]   68 3.543263e-06
## [70,]   69 6.928726e-06
## [71,]   70 1.326438e-05
## [72,]   71 2.486487e-05
## [73,]   72 4.564908e-05
## [74,]   73 8.209263e-05
## [75,]   74 1.446376e-04
## [76,]   75 2.497132e-04
## [77,]   76 4.225350e-04
## [78,]   77 7.008453e-04
## [79,]   78 1.139719e-03
## [80,]   79 1.817474e-03
## [81,]   80 2.842578e-03
## [82,]   81 4.361251e-03
## [83,]   82 6.565178e-03
## [84,]   83 9.698472e-03
## [85,]   84 1.406270e-02
## [86,]   85 2.001860e-02
## [87,]   86 2.798287e-02
## [88,]   87 3.841882e-02
## [89,]   88 5.181952e-02
## [90,]   89 6.868333e-02
```

```
## [91,]   90 8.948202e-02
## [92,]   91 1.146233e-01
## [93,]   92 1.444102e-01
## [94,]   93 1.790015e-01
## [95,]   94 2.183767e-01
## [96,]   95 2.623112e-01
## [97,]   96 3.103645e-01
## [98,]   97 3.618855e-01
## [99,]   98 4.160352e-01
## [100,]   99 4.718258e-01
## [101,]  100 5.281742e-01
```

This is too hard, let's let R do the heavy lifting.

```r
qbinom(.025,200,.5)
```

```
## [1] 86
```

```r
pbinom(85:87,200,.5)
```

```
## [1] 0.02001860 0.02798287 0.03841882
```

If we get 85 heads or less, then we reject. For the upper critical value, we have

```r
qbinom(.975,200,.5)
```

```
## [1] 114
```

```r
cbind(113:115,pbinom(113:115,200,.5,lower=F))
```

```
##      [,1]       [,2]
## [1,]  113 0.02798287
## [2,]  114 0.02001860
## [3,]  115 0.01406270
```

The critical values are 85 and 114. Now we calculate power which is the probability of rejecting given that the probability of success is .55. I will do it two equivalent ways.

```r
1-(pbinom(114,200,.55)-pbinom(85,200,.55))
```

```
## [1] 0.2619829
```

```r
pbinom(114,200,.55,lower=F)+pbinom(85,200,.55)
```

```
## [1] 0.2619829
```

Part b.
This is asking us to determine the impact of sample size on power. Before calculating, we think having more data, information, will increase the power. We repeat the same steps as part a.

```
qbinom(.025,400,.5)
```

```
## [1] 180
```

```
qbinom(.975,400,.5)
```

```
## [1] 220
```

And now the power

```
pbinom(220,400,.55,lower=F)+pbinom(179,400,.55)
```

```
## [1] 0.4806564
```

Part c.
For a sample size of 200, the power to detect a probability of success of .55 versus .5 is .262, and for a sample size of 400, it is .481. This question wants to know what sample size we need for a power of at least .90. We could do this by trial and error but I want to write a function.

First I will experiment with code from the previous problem

```
1-(pbinom(qbinom(.975,400,.5),400,.55)-pbinom(qbinom(.025,400,.5),400,.55))
```

```
## [1] 0.4806694
```

Now I have an idea of how to write the function

```
mypower=function(n){
  1-(pbinom(qbinom(.975,n,.5),n,.55)-pbinom(qbinom(.025,n,.5),n,.55))
  }
```

Testing the function

```
mypower(400)
```

```
## [1] 0.4806694
```

Making a table

```
cbind(SampleSize=500:600,Power=mypower(500:600))
```

```
##        SampleSize     Power
##   [1,]        500 0.5894618
##   [2,]        501 0.6084543
##   [3,]        502 0.5927692
##   [4,]        503 0.6116796
##   [5,]        504 0.5960572
##   [6,]        505 0.5803105
##   [7,]        506 0.5993258
```

```
##    [8,]          507 0.5836376
##    [9,]          508 0.6025749
##   [10,]          509 0.5869460
##   [11,]          510 0.6058044
##   [12,]          511 0.5902356
##   [13,]          512 0.6090145
##   [14,]          513 0.5935062
##   [15,]          514 0.6122050
##   [16,]          515 0.5967580
##   [17,]          516 0.6153760
##   [18,]          517 0.5999908
##   [19,]          518 0.6185274
##   [20,]          519 0.6032046
##   [21,]          520 0.6216592
##   [22,]          521 0.6063994
##   [23,]          522 0.6247715
##   [24,]          523 0.6095753
##   [25,]          524 0.6278642
##   [26,]          525 0.6127320
##   [27,]          526 0.6309373
##   [28,]          527 0.6158697
##   [29,]          528 0.6006530
##   [30,]          529 0.6189884
##   [31,]          530 0.6038326
##   [32,]          531 0.6220879
##   [33,]          532 0.6069937
##   [34,]          533 0.6251684
##   [35,]          534 0.6101363
##   [36,]          535 0.6282298
##   [37,]          536 0.6132603
##   [38,]          537 0.6312721
##   [39,]          538 0.6163657
##   [40,]          539 0.6342953
##   [41,]          540 0.6194525
##   [42,]          541 0.6372994
##   [43,]          542 0.6225207
##   [44,]          543 0.6402845
##   [45,]          544 0.6255703
##   [46,]          545 0.6432505
##   [47,]          546 0.6286012
##   [48,]          547 0.6461975
##   [49,]          548 0.6316135
##   [50,]          549 0.6491255
##   [51,]          550 0.6346072
##   [52,]          551 0.6520344
##   [53,]          552 0.6375823
##   [54,]          553 0.6229571
##   [55,]          554 0.6405388
##   [56,]          555 0.6259766
##   [57,]          556 0.6434766
##   [58,]          557 0.6289780
##   [59,]          558 0.6463958
##   [60,]          559 0.6319611
##   [61,]          560 0.6492965
```

```
##  [62,]        561 0.6349261
##  [63,]        562 0.6521786
##  [64,]        563 0.6378728
##  [65,]        564 0.6550422
##  [66,]        565 0.6408014
##  [67,]        566 0.6578872
##  [68,]        567 0.6437118
##  [69,]        568 0.6607138
##  [70,]        569 0.6466041
##  [71,]        570 0.6635218
##  [72,]        571 0.6494781
##  [73,]        572 0.6663114
##  [74,]        573 0.6523341
##  [75,]        574 0.6690826
##  [76,]        575 0.6551719
##  [77,]        576 0.6410719
##  [78,]        577 0.6579916
##  [79,]        578 0.6439556
##  [80,]        579 0.6607932
##  [81,]        580 0.6468215
##  [82,]        581 0.6635767
##  [83,]        582 0.6496696
##  [84,]        583 0.6663422
##  [85,]        584 0.6525000
##  [86,]        585 0.6690897
##  [87,]        586 0.6553127
##  [88,]        587 0.6718192
##  [89,]        588 0.6581077
##  [90,]        589 0.6745307
##  [91,]        590 0.6608849
##  [92,]        591 0.6772244
##  [93,]        592 0.6636445
##  [94,]        593 0.6799001
##  [95,]        594 0.6663865
##  [96,]        595 0.6825580
##  [97,]        596 0.6691108
##  [98,]        597 0.6851981
##  [99,]        598 0.6718175
## [100,]        599 0.6878204
## [101,]        600 0.6745067
```

The sample size will be large; let's let R figure how large without printing a huge table.

```r
which.min(abs(mypower(1:1500)-.9))
```

```
## [1] 1079
```

The closest value to .9 is a sample since of 1079, but let's focus in this area to find the best answer.

```r
cbind(SampleSize=1077:1081,Power=mypower(1077:1081))
```

```
##      SampleSize     Power
```

```
## [1,]       1077 0.8991071
## [2,]       1078 0.9048146
## [3,]       1079 0.8999742
## [4,]       1080 0.9056400
## [5,]       1081 0.9008344
```

The answer jumps around because of the discrete nature of the data. Thus we can never get exactly 0.05 in the rejection region. However, from the analysis, it appears that 1078 flips would suffice.

Problem 2.56 To achieve a power to detect an alternative hypothesis of the probability of heads being 0.55 versus the null of 0.50 with a sample size of 200 is .262. For a sample size of 400 it is .481. To achieve a power of at least .90, we need a sample size of 1078 flips.

Section 2.5

First I will load the libraries needed.

```r
library('fastR')
library("Hmisc")
```

Problem 2.60. Let f be the pmf of X and let $\mu = E(X)$. Then
Step 1 is $Var(X) = \sum(x-\mu)^2 f(x)$ this is the definition of variance
Step 2 is $\sum(x^2 - 2x\mu + \mu^2)f(x)$ this is just expanding the square
Step 3 is $\sum x^2 f(x) - \sum 2x\mu f(x) + \sum \mu^2 f(x))$ this is just distributing the sum
Step 4 is $E(X^2) - 2\mu \sum x f(x) + \mu^2 \sum f(x)$ this is definition of $E(X)$
Step 5 is $E(X^2) - 2\mu\mu + \mu^2$ this is the definition of $E(X)$ and the properties of a pmf
Step 6 is $E(X^2) - \mu^2 = E(X^2) - E(X)^2$ this is algebra and the definition of $E(X)$

Problem 2.61 Prove $Var(aX + b) = a^2 Var(X)$
Starting with $Var(aX + b)$ by Thm 2.5.7 we have

$$Var(aX + b) = E[(aX + b)^2] - [E(aX + b)]^2$$

$$= E(a^2X^2 + 2abX + b^2) - [E(aX + b)]^2$$

by Lemma 2.5.3

$$= a^2E(X^2) + 2abE(X) + b^2 - [aE(X) + b]^2$$
$$= a^2E(X^2) + 2abE(X) + b^2 - a^2E(X)^2 - 2abE(X) - b^2$$
$$= a^2E(X^2) - a^2E(X)^2$$
$$= a^2[E(X^2) - E(X)^2] = a^2Var(X)$$

Problem 2.64 The probability mass function is $f(x) = 1/n$ for $x = 1, 2, ..., n$ and zero otherwise. By definition

$$E(X) = 1/n(1) + 1/n(2) + ... + 1/n(n) = 1/n[1 + 2 + 3 + ... + n]$$

The sum in the brackets is a triangular number and is equal to $\frac{(n+1)n}{2}$.

Thus

$$E(X) = \frac{1}{n}\frac{(n+1)n}{2} = \frac{(n+1)}{2}$$

For variance we will use $Var(X) = E(X^2) - E(X)^2$

$$E(X^2) = 1/n(1^2) + 1/n(2^2) + ...1/n(n^2) = 1/n[1^2 + 2^2 + 3^2 + ... + n^2]$$

These are pyramidal numbers, see section B of the book. Thus we get

$$\frac{1}{n}\frac{n(n+1)(2n+1)}{6} = \frac{(n+1)(2n+1)}{6}$$

and

$$Var(X) = \frac{(n+1)(2n+1)}{6} - \frac{(n+1)^2}{4}$$

Problem 2.65 Part a. this is just problem 2.64 in words, it is a discrete uniform random variable. Thus $E(X) = \frac{(n+1)}{2}$.

Part b. The random variable X is the number of items out of 255 to be accessed until we find the desired value. This problem is difficult so let's break it down. Suppose our value is in position 128, then we only have to search once. If not then there are 127 items below and 127 items above, note that there will always be an odd number. It takes two steps if the item is at position 62 or 192. Now I see the pattern and since the location was random and equally likely we have

$$E(X) = 1(1/255) + 2(2/255) + 3(4/255) + ... + 8(128/255) = \sum_{i=1}^{8} i\frac{2^{i-1}}{255}$$

We cannot take more than 8 steps. Using R to calculate this sum for us

```
i<-(1:8)
sum(i*2^(i-1)/255)
```

```
## [1] 7.031373
```

For part a we have

```
(255+1)/2
```

```
## [1] 128
```

If we want to generalize the result in part b, notice that we have $2^n - 1$ number of elements to be searched. Thus we can write the expected value as

$$E(X) = 1(1/255) + 2(2/255) + 3(4/255) + ... + 8(128/255) + ... = \sum_{i=1}^{n} i\frac{2^{i-1}}{2^n - 1}$$

Problem 2.67 We want to compare $\frac{1}{3.5}$ with $E(1/X)$ where x is the number on a fair die. The probability mass function can take on the values 1,1/2,1/3,1/4,1/5, and 1/6 all with probability 1/6. Thus the expected value by definition is

$$E(\frac{1}{X}) = \sum_{i=1}^{6} \frac{1}{6}\frac{1}{x}$$

```
i<-(1:6)
1/6*sum(1/i)
```

```
## [1] 0.4083333
```

```
1/3.5
```

```
## [1] 0.2857143
```

Your expected value is better with taking one over the roll on the die, it is a little under 41 cents. However, note that half the time you will get less than 1/3.5, when you roll a 4, 5, or 6. The other half you will get more. Part of the answer to this question also relies on how risk adverse you are.

Section 2.6

Problem 2.71 The book discusses pairwise independence, for this problem that would mean that X and Y are independent, X and Z are independent, and Y and Z are independent. In this case we would have

$$f_{xy}(x,y) = f_x(x)f_y(y)$$

$$f_{xz}(x,z) = f_x(x)f_z(z)$$

and

$$f_{yz}(y,z) = f_y(y)f_z(z)$$

The book also discusses independence, where X, Y, and Z are all independent, mathematically

$$f_{xyz}(x,y,z) = f_x(x)f_y(y)f_z(z)$$

A third one not discussed is conditional independence, for example x and y are independent given z. Mathematically

$$f_{xy|z}(x,y) = f_{x|z}(x)f_{y|z}(y)$$

this must be true for each values of z.

Problem 2.76 Prove Lemma 2.6.12. To do this we need three things, the definition of covariance, Definition 2.6.10, the fact that $E(X) = \mu_x$, and property 1 of Theorem 2.6.7.

Starting with the definition of covariance we have

$$Cov(X,Y) = E(XY) - E(X)E(Y)$$

we next we use a standard technique of adding and subtracting the same term

$$Cov(X,Y) = E(XY) - E(X)E(Y) - \mu_x\mu_y + \mu_x\mu_y$$

rewriting using the general idea that $E(Y) = \mu_y$

$$Cov(X,Y) = E(XY) - E(X)\mu_y - \mu_x E(Y) + \mu_x\mu_y$$

and next, using that the expected value of a constant is just a constant,

$$Cov(X,Y) = E(XY) - E(\mu_y X) - E(\mu_x Y) + E(\mu_x \mu_y)$$

and finally from property 1 of Theorem 2.6.7, the expected value of a sum is the sum of the expected values

$$Cov(X,Y) = E(XY - \mu_y X - \mu_x Y + \mu_x \mu_y)$$

the last step is just algebra

$$Cov(X,Y) = E[(X - \mu_x)(Y - \mu_y)]$$

Problem 2.92 Part a. K is the number of kings in a 5 card hand and Q is the number of queens in a five card hand. The probability of 4 kings, P(K=4), is

```
choose(4,4)*choose(48,1)/choose(52,5)
```

```
## [1] 1.846893e-05
```

while the probability of 4 kings, given that we have 4 queens, $P(K = 4 \mid Q = 4)$, is zero. Therefore the random variable K and Q are not independent.

Part b. Find $P(K = 2 \mid Q = 2)$. First we need $P(K = 2, Q = 2)$

```
choose(4,2)*choose(4,2)*choose(44,1)/choose(52,5)
```

```
## [1] 0.0006094746
```

Next we need $P(Q = 2)$

```
choose(4,2)*choose(48,3)/choose(52,5)
```

```
## [1] 0.03992982
```

finally $P(K = 2 \mid Q = 2) = P(K = 2, Q = 2)/P(Q = 2)$

```
(choose(4,2)*choose(4,2)*choose(44,1)/choose(52,5))/(choose(4,2)*choose(48,3)/choose(52,5))
```

```
## [1] 0.01526364
```

Another way pointed out by one of the students is now that we know we have 2 kings, we have to pick 2 queens and one other card out of the remaining 48 cards.

```
(choose(4,2)*choose(44,1))/choose(48,3)
```

```
## [1] 0.01526364
```

Problem 2.95 A fair coin is tossed five times. Let Y be the number of heads in all five tosses. Let X be the number of heads in the first two tosses.

Part a. Are X and Y independent? No $P(X = 2) = 1/4$ and $P(X = 2 \mid Y = 0) = 0$.

Part b. Find pmf $f_{y|x}(y)$ First let's look at X=0, no heads in the first two flips. In this case, Y can only take the values 0, 1, 2, and 3; we can't get four or five heads in five flips if the first two flips are heads. Now let's complete the table

So $f_{y|x=0}(0) = f_{xy}(0,0)/f_x(0)$ but $P(X = 0) = f_x(0) = 1/4$ and $P(X = 0, Y = 0) = f_{xy}(0,0) = (1/2)^5$. Thus $f_{y|x=0}(0) = (1/2)^3$.

Next $f_{y|x=0}(1) = f_{xy}(0,1)/f_x(0)$ but $P(X = 0) = f_x(0) = 1/4$ and $P(X = 0, Y = 1) = f_{xy}(0,1) = 3 * (1/2)^5$. We have the three in there because there are three places for the one head to go, it cannot go in the first two spots. Thus $f_{y|x=0}(1) = 3 * (1/2)^3$.

by similar arguments $f_{y|x=0}(2) = 3 * (1/2)^3$, note we have 3 choose 2 spots to place the two heads, and $f_{y|x=0}(3) = (1/2)^3$.

This conditional distribution is a binomial random variable.

If you repeat for x=1 and x=2 you get similar results. In general we have

$$f_{y|x}(y) = \binom{3}{y-x}(1/2)^3; y \geq x$$

Part c. Find pmf $f_{x|y}(x)$

If Y=0, then X=0.

If Y=1, then X could be 0 or 1. So $f_{x|y=1}(0) = f_{xy}(0,1)/f_y(1)$ but $P(Y = 1) = f_y(1) = \binom{5}{1}(1/2)^5$ and $P(X = 0, Y = 1) = f_{xy}(0,1) = \binom{3}{1}(1/2)^5$. Thus $f_{x|y=1}(0) = \dfrac{\binom{3}{1}}{\binom{5}{1}}$

There is a slight change if X=1 when Y=1. In this case we only have two choices were to put the heads, it must be one of the first two positions. Thus $f_{x|y=1}(1) = \dfrac{\binom{2}{1}}{\binom{5}{1}}$

When Y=2 we get that X can be 0, 1, or 2 and $f_{x|y=2}(0) = \dfrac{\binom{3}{2}}{\binom{5}{2}}$

Notice, that we removed the $(1/2)^5$ as that will be in both the numerator and denominator. One way to think of it is that we have three tails and we have to put two of them in the first two spots. This is $\dfrac{\binom{3}{2}}{\binom{5}{2}}$.

For the remainder, we have 2 heads and we have to place them and 1 tail. This is $\dfrac{\binom{2}{2}\binom{1}{1}}{\binom{3}{3}}$ which is 1.

$$f_{x|y=2}(1) = \frac{\binom{2}{1}\binom{3}{1}}{\binom{5}{2}}$$

$$f_{x|y=2}(2) = \frac{\binom{2}{2}}{\binom{5}{2}}$$

When Y=3 we get that X can be 0, 1, or 2 and $f_{x|y=3}(0) = \frac{\binom{2}{2}}{\binom{5}{3}}$

$$f_{x|y=3}(1) = \frac{\binom{2}{1}\binom{3}{1}}{\binom{5}{3}}$$

$$f_{x|y=3}(2) = \frac{\binom{3}{2}}{\binom{5}{3}}$$

When Y=4 we get that X can be 1 or 2 and

$$f_{x|y=4}(1) = \frac{\binom{4}{1}\binom{1}{1}}{\binom{5}{4}}$$

$$f_{x|y=4}(2) = \frac{\binom{4}{2}}{\binom{5}{4}}$$

When Y=5 we get that X must be 2 and

$$f_{x|y=5}(1) = \frac{\binom{5}{2}}{\binom{5}{5}} = 1$$

This is a hypergeometric distribution and we will learn about it in section 2.7.

Section 2.7

Problem 2.42 This is a sampling without replacement but looks like a geometric where the success is on the last trial. We will use the product of hypergeometrics to solve this problem. This is similar to the idea of using a binomial for the first $n-1$ trials in a negative binomial and then just multiplying by the probability of success to account for the last trial that is a success.

For our problem, X can take the value of 1, 2, 3, and 4. For the first case when X equals 1 we have

$$\frac{\binom{2}{1}}{\binom{5}{1}}$$

since we have to get a match on the first grab.

When X equals 2, we have

$$\frac{\binom{3}{1}}{\binom{5}{1}}$$

for the first spot and

$$\frac{\binom{2}{1}}{\binom{4}{1}}$$

for the second.

To find the probability we multiply to get

$$\frac{\binom{3}{1}\binom{2}{1}}{\binom{5}{1}\binom{4}{1}}$$

When X equals 3, we have

$$\frac{\binom{3}{2}}{\binom{5}{2}}$$

for the first spot and

$$\frac{\binom{2}{1}}{\binom{3}{1}}$$

for the second.

To find the probability we multiply to get

$$\frac{\binom{3}{2}\binom{2}{1}}{\binom{5}{2}\binom{3}{1}}$$

Finally, for the last case of X equaling 4 we have

$$\frac{\binom{3}{3}\binom{2}{1}}{\binom{5}{2}\binom{2}{1}}$$

It is easy to see how to generalize to more than one success.

Problem 2.80 We are given that on average 6 customers arrive per hour.

Part a. The random variable X = the number of customers arriving in 20 minutes. This is a Poisson random variable. The parameter λ is the average number of customers in 20 minutes. since 6 arrive in an hour, 2 will arrive in 20 minutes. The probability statement is P(X=0).

```
dpois(0,2)
```

```
## [1] 0.1353353
```

Part b. Same λ but now we want P(X=2).

```
dpois(2,2)
```

```
## [1] 0.2706706
```

Part c. Now the random variable is the number of customers arriving in an hour which means λ is 6. We want P(X>6).

```
1-ppois(6,6)
```

```
## [1] 0.3936972
```

```
ppois(6,6,lower=FALSE)
```

```
## [1] 0.3936972
```

P(X<6)

```
ppois(5,6)
```

```
## [1] 0.4456796
```

and P(X=6)

```
dpois(6,6)
```

```
## [1] 0.1606231
```

Part d. The random variable is the number of customers in 4 hours nd λ is 24. We want $P(20 \leq X \leq 30)$

```
ppois(30,24)-ppois(19,24)
```

```
## [1] 0.7238911
```

Part e. If the business is a restaurant, then people could come together as a group so that the assumptions that only one individual will arrive in a small time interval is not met.

Problem 2.81 The random variable Y is the number of goals scored by Zetterberg in 89 games. The rate parameter λ for this problem is (206*89)/506 goals per 89 games. We want to find the probability P(X > 43).

```r
1-ppois(43,206*89/506)
```

```
## [1] 0.1156876
```

```r
ppois(43,206*89/506,lower=FALSE)
```

```
## [1] 0.1156876
```

I would say that the data does not support Coach Babcock's claim since it is not an unusually low probability.

Problem 2.85 The null hypothesis is that the proportion of people buying a new ticket when they lost cash is equal to the proportion of people who bought a new ticket when they lost the original ticket. The alternative is one-sided and is that the proportion of people buying a new ticket when they lost cash is greater than the proportion of people who bought a new ticket when they lost the original ticket.

Using a hypergeometric we have 22 people who lost cash, 23 who lost ticket, and 22 who bought a second ticket. If the proportion were the same, it would not matter who of the 22 second ticket buyers we assigned to the lost cash or lost ticket group. Our random variable is X the number of play attendees who lost cash out of 22. The test statistics is 13 and the p-value is $P(\geq 13)$.

```r
1-phyper(12,22,23,22)
```

```
## [1] 0.1490436
```

```r
phyper(12,22,23,22,lower=F)
```

```
## [1] 0.1490436
```

or we use the random variable X the number of people not attending the play who lost cash out of 23. Then we want $P(X \leq 9)$

```r
phyper(9,23,22,22)
```

```
## [1] 0.1490436
```

If you use fisher.test in R it will use the upper left hand corner cell as the reference so you must base the alternative on this.

```r
Prob2.85=rbind(c(9,14),c(13,9))
Prob2.85
```

```
##      [,1] [,2]
## [1,]    9   14
## [2,]   13    9
```

```r
fisher.test(Prob2.85,alt="less")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  Prob2.85
## p-value = 0.149
## alternative hypothesis: true odds ratio is less than 1
## 95 percent confidence interval:
##  0.000000 1.420837
## sample estimates:
## odds ratio
##  0.4533593
```

Problem 2.86 Same hypothesis test as problem 2.85 but now with more data

```
Prob2.86=rbind(c(61,69),c(103,44))
Prob2.86
```

```
##      [,1] [,2]
## [1,]   61   69
## [2,]  103   44
```

```
fisher.test(Prob2.86,alt="less")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  Prob2.86
## p-value = 7.215e-05
## alternative hypothesis: true odds ratio is less than 1
## 95 percent confidence interval:
##  0.0000000 0.5890587
## sample estimates:
## odds ratio
##  0.3790412
```

or in a more manual way

```
1-phyper(102,164,113,147)
```

```
## [1] 7.214999e-05
```

```
phyper(102,164,113,147,lower=FALSE)
```

```
## [1] 7.214999e-05
```

This second table gives stronger evidence that people respond differently to losing cash and losing a ticket.

Problem 2.88 Here is the original data

```r
convictions <- rbind(dizygotic=c(2,15), monozygotic=c(10,3))
colnames(convictions) <- c('convicted','not convicted')
convictions
```

```
##             convicted not convicted
## dizygotic           2            15
## monozygotic        10             3
```

```r
fisher.test(convictions, alternative = "less")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  convictions
## p-value = 0.0004652
## alternative hypothesis: true odds ratio is less than 1
## 95 percent confidence interval:
##  0.0000000 0.2849601
## sample estimates:
## odds ratio
## 0.04693661
```

using phyper it is

```r
phyper(2,17,13,12)
```

```
## [1] 0.0004651809
```

or

```r
phyper(2,12,18,17)
```

```
## [1] 0.0004651809
```

Now let's use the lower right cell, if monozygotic are convicted at higher rate we want the probability of 3 or less

```r
phyper(3,18,12,13)
```

```
## [1] 0.0004651809
```

Using fisher.test

```r
convictions2 <- rbind(monozygotic=c(3,10), dizygotic=c(15,2))
colnames(convictions2) <- c('not convicted','convicted')
convictions2
```

```
##             not convicted convicted
## monozygotic             3        10
## dizygotic              15         2
```

```r
fisher.test(convictions2, alternative = "less")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  convictions2
## p-value = 0.0004652
## alternative hypothesis: true odds ratio is less than 1
## 95 percent confidence interval:
##  0.0000000 0.2849601
## sample estimates:
## odds ratio
## 0.04693661
```

or using upper right cell from the original table, now we want more of the dizygotic not convicted.

```r
1-phyper(14,17,13,18)
```

```
## [1] 0.0004651809
```

Using fisher.test

```r
convictions3 <- rbind(dizygotic=c(15,2), monozygotic=c(3,10))
colnames(convictions3) <- c('not convicted','convicted')
convictions3
```

```
##             not convicted convicted
## dizygotic              15         2
## monozygotic             3        10
```

```r
fisher.test(convictions3, alternative = "greater")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  convictions3
## p-value = 0.0004652
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  3.509263      Inf
## sample estimates:
## odds ratio
##   21.30533
```

and finally the lower left of the original table

```r
phyper(9,13,17,12,lower=F)
```

```
## [1] 0.0004651809
```

using fisher.test

```r
convictions4 <- rbind(monozygotic=c(10,3), dizygotic=c(2,15))
colnames(convictions4) <- c('convicted','not convicted')
convictions4
```

```
##             convicted not convicted
## monozygotic        10             3
## dizygotic           2            15
```

```r
fisher.test(convictions4, alternative = "greater")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  convictions4
## p-value = 0.0004652
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  3.509263      Inf
## sample estimates:
## odds ratio
##   21.30533
```

Chapter 3

Section 3.1

Problem 3.1 Part a.

For a pdf, $f(x) \geq 0$ and $\int_{-\infty}^{\infty} f(x)dx = 1$ Thus

$$\int_{-\infty}^{\infty} f(x)dx = \int_{-2}^{2} k(x-2)(x+2)dx = 1$$

$$k\int_{-2}^{2}(x^2-4)dx = 1$$

$$k(\frac{x^3}{3} - 4x)|_{-2}^{2} = 1$$

$$k(\frac{8}{3} - 8 - (\frac{-8}{3} + 8)) == 1$$

$$k * (16/3 - 16) = 1$$

$$k = -3/32$$

or using R

```r
library(MASS)
```

```r
Prob3.1f<-function(x){(x-2)*(x+2)*(-2 <= x & x<=2)}
integrate(Prob3.1f,-2,2)
```

```
## -10.66667 with absolute error < 1.2e-13
```

```r
fractions(integrate(Prob3.1f,-2,2)$value)
```

```
## [1] -32/3
```

```r
fractions(1/integrate(Prob3.1f,-2,2)$value)
```

```
## [1] -3/32
```

Part b.

Find $P(X \geq 0)$. By definition

$$P(X \geq 0) = \int_0^2 (-3/32)(x^2 - 4)dx$$

$$(\frac{-3}{32})(\frac{x^3}{3} - 4x)|_0^2$$

$$(\frac{-3}{32})(\frac{8}{3} - 8)$$

$$(\frac{-3}{32})(\frac{8}{3} - \frac{24}{3})$$

$$(\frac{-3}{32})(-\frac{16}{3}) = \frac{1}{2}$$

Using R

```r
Prob3.1f<-function(x){(-3/32)*(x-2)*(x+2)*(-2 <= x & x<=2)}
integrate(Prob3.1f,-2,2)
```

```
## 1 with absolute error < 1.1e-14
```

```r
fractions(integrate(Prob3.1f,0,2)$value)
```

```
## [1] 1/2
```

Part c.

Find $P(X \geq 1)$. By definition

$$P(X \geq 1) = \int_1^2 (-3/32)(x^2 - 4)dx$$

$$(\frac{-3}{32})(\frac{x^3}{3} - 4x)|_1^2$$

$$(\frac{-3}{32})(\frac{8}{3} - 8 - (\frac{1}{3} - 4))$$

$$(\frac{-3}{32})(\frac{7}{3} - 4)$$

$$(\frac{-3}{32})(-\frac{5}{3}) = \frac{5}{32}$$

Using R

```r
fractions(integrate(Prob3.1f,1,2)$value)
```

```
## [1] 5/32
```

Part d.

Find $P(-1 \leq X \leq 1)$. By definition

$$P(X \geq 1) = \int_{-1}^{1} (-3/32)(x^2 - 4)dx$$

$$(\frac{-3}{32})(\frac{x^3}{3} - 4x)|_{-1}^{1}$$

$$(\frac{-3}{32})(\frac{1}{3} - 4 - (\frac{-1}{3} + 4))$$

$$(\frac{-3}{32})(\frac{2}{3} - 8)$$

$$(\frac{-3}{32})(-\frac{22}{3}) = \frac{11}{16}$$

Using R

```r
fractions(integrate(Prob3.1f,-1,1)$value)
```

```
## [1] 11/16
```

Problem 3.3 An example of a random variable is I flip a coin, if it is heads I record a -1, if it is tails, I generate a random number from a continuous uniform distribution over the interval [0,1] and record it. This random variable does not have a pmf or pdf, but it does have a cdf as I can calculate $P(X \leq x)$ For example $P(X \leq 0) = \frac{1}{2}$ and $P(X \leq \frac{1}{2}) = \frac{1}{2} + \frac{1}{2} * \frac{1}{2} = \frac{3}{4}$.

Problem 3.5 The pdf for an exponential random variable is

$$f(x) = \lambda e^{-\lambda x}$$

By definition of median, we need to find x such that $P(X \leq x) = 0.5$. So

$$P(X \leq x) = \int_{0}^{x} \lambda e^{-\lambda y} dy = 0.5$$

$$1 - e^{-\lambda x} = 0.5$$

$$e^{-\lambda x} = 0.5$$

$$-\lambda x = ln(.5)$$

$$x = \frac{-ln(.5)}{\lambda}$$

$$x = \frac{ln(2)}{\lambda}$$

By similar arguments, the first quartile is

$$x = \frac{-ln(3/4)}{\lambda}$$

and the third quartile is

$$x = \frac{-ln(1/4)}{\lambda}$$

Problem 3.7 Let's take two exponential distributions. For the first one, we leave it alone. For the second, we change the pdf at the point $x = 1$ to be zero. These have the same cdfs and thus are equal in distribution.

Additional Problem 1. To solve this we want to find the cdf of Y.

$$P(Y \leq y)$$

$$= P(\sqrt{X} \leq y)$$

and since X is an exponential random variable, we get

$$= P(X \leq y^2) = 1 - e^{-y^2}$$

Now to find the pdf of Y, we differentiate with respect to Y to get

$$\frac{d}{dy}(1 - e^{-y^2}) = 2ye^{-y^2}$$

The pdf of Y is $2ye^{-y^2}$ for $y \geq 0$ and 0 otherwise.

Additional Problem 2. We can solve this as either a Poisson or exponential. I will do it both ways for reference. First as a Poisson. Let X be the number of cars entering the north gate in the next 5 minutes. For this $\lambda = 4$ and we want $P(X = 0)$. Using R we get

```
round(dpois(0,4),4)
```

```
## [1] 0.0183
```

Now as an exponential. Let Y be the time in minutes until the next car enters the north gate. We want $P(Y \geq 5)$ and $\lambda = 48/60$. Using R we get

```
round(1-pexp(5,rate=48/60),4)
```

```
## [1] 0.0183
```

Section 3.2

Homework

This is my homework for Section 3.2 of the book.

Problem 3.8 Prove

$$Var(X) = E(X^2) - E(X)^2$$

Start with the definition of variance

$$Var(X) = E[(X - \mu)^2]$$

where

$$\mu = E(X)$$

Expanding the product and using the that the expected value of a sum is the sum of the expected values, this comes from the fact that expectation is an integral and integration is a linear operator, we get

$$Var(X) = E[X^2 - 2\mu X + \mu^2] = E(X^2) + E(-2\mu X) + E(\mu^2)$$

Note: Using the integral definition of expectation, we see that

$$E[X^2 - 2\mu X] = \int_{-\infty}^{\infty} [x^2 - 2\mu x] f(x) dx$$

$$= \int_{-\infty}^{\infty} x^2 f(x) dx - 2\mu \int_{-\infty}^{\infty} x f(x) dx$$

$$= E(X^2) - 2\mu E(X)$$

Next, the constants can come outside of the expected value

$$= E(X^2) - 2\mu E(X) + \mu^2$$

but

$$\mu = E(X)$$

so

$$Var(X) = E(X^2) - 2\mu E(X) + \mu^2 = E(X^2) - 2E(X)^2 + E(X)^2 = E(X^2) - E(X)^2$$

Problem 3.10 For the random variable X let $F(x) = x^2/4$ on $[0, 2]$.

Part a. Find $P(X \leq 1)$. This is $F(1) = 1^2/4 = 1/4$. Using R

```
library(MASS)
```

```
Prob3.10=function(x){x^2/4}
Prob3.10(1)
```

```
## [1] 0.25
```

```
fractions(Prob3.10(1))
```

```
## [1] 1/4
```

Part b. Find $P(0.5 \leq X \leq 1.0)$. This is $P(X \leq 1) - P(X \leq .05)$ or $F(1) - F(.5) = 1^2/4 - (1/2)^2/4 = 3/16$. Using R

```
fractions(Prob3.10(1)-Prob3.10(.5))
```

```
## [1] 3/16
```

Part c. Find $P(X \geq 1.5)$. This is $1 - P(X < 1.5) = 1 - P(X \leq 1.5)$ or $1 - F(1.5) = 1 - (1.5)^2/4 = 7/16$
Using R

```
fractions(1-Prob3.10(1.5))
```

```
## [1] 7/16
```

Part d. Find the median of X. By definition, the median is the value x such that $P(X \leq x) = .5$. Solving we have $F(x) = .5$ or $x^2/4 = .5$. Thus the median is $\sqrt{2}$
Using R

```
uniroot(function(x)Prob3.10(x)-.5,lower=0,upper=2)$root
```

```
## [1] 1.414213
```

```
sqrt(2)
```

```
## [1] 1.414214
```

Just for fun, let's see what fraction R gives us for the irrational number $\sqrt{2}$.

```
fractions(sqrt(2))
```

```
## [1] 8119/5741
```

Part e. Find the pdf of X. This is just the derivative of the CDF by the fundamental theorem of Calculus.

$$f(x) = \frac{d}{dx}\frac{x^2}{4} = \frac{x}{2}$$

and the domain is still $[0, 2]$.

Part f. Find $E(X)$. By definition $E(X) = \int_{-\infty}^{\infty} x f(x) dx$ Thus we get

$$E(X) = \int_0^2 x\frac{x}{2}dx = \int_0^2 \frac{x^2}{2}dx$$
$$= \frac{x^3}{6}|_0^2 = \frac{2^3}{6} - 0 = \frac{8}{6} = \frac{4}{3}$$

Using R

```
Prob3.10f<-function(x){(x^2/2)*(0 <= x & x<=2)}
integrate(Prob3.10f,0,2)
```

```
## 1.333333 with absolute error < 1.5e-14
```

```
fractions(integrate(Prob3.10f,0,2)$value)
```

```
## [1] 4/3
```

Part g. Find $Var(X)$. We first need $E(X^2)$ and by definition

$$E(X^2) = \int_0^2 x^2\frac{x}{2}dx = \int_0^2 \frac{x^3}{2}dx$$
$$= \frac{x^4}{8}|_0^2 = \frac{2^4}{8} - 0 = \frac{16}{8} = 2$$
$$Var(X) = E(X^2) - E(X)^2 = 2 - (\frac{4}{3})^2 = \frac{2}{9}$$

Using R

```
Prob3.10g<-function(x){(x^3/2)*(0 <= x & x<=2)}
fractions(integrate(Prob3.10g,0,2)$value)
```

```
## [1] 2
```

```
fractions(integrate(Prob3.10g,0,2)$value-(integrate(Prob3.10f,0,2)$value)^2)
```

```
## [1] 2/9
```

Problem 3.12 The time X between two randomly selected consecutive cars in a traffic flow model is modeled with the pdf $f(x) = \frac{k}{x^4}$ on $[0, \infty]$.

Part a. Determine the value of k. From properties of a pdf,

$$\int_{-\infty}^{\infty} f(x)dx = \int_{1}^{\infty} \frac{k}{x^4}dx = 1$$

$$\lim_{a\to\infty} \int_{1}^{a} \frac{k}{x^4}dx = 1$$

so

$$\lim_{a\to\infty} k\frac{1}{-3x^3}|_1^a = 1$$

$$k \lim_{a\to\infty} \frac{1}{-3a^3} - \frac{1}{(-3)1^3} = 1$$

but

$$\lim_{a\to\infty} \frac{1}{-3a^3} = 0$$

so

$$\frac{k}{3} = 3$$

or

$$k = 3$$

Checking using R

```
Prob3.12=function(x){3/(x^4)}
integrate(Prob3.12,1,Inf)$value
```

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

Part b. Find the CDF of X. By definition $F(x) = P(X \leq X)$. Thus

$$F(x) = P(X \leq X) = \int_{-\infty}^{x} f(y)dy = \int_{1}^{x} \frac{3}{x^4}dx$$

$$\frac{-1}{x^3}|_1^x = \frac{-1}{x^3} - \frac{-1}{1^3} = 1 - \frac{1}{x^3}$$

Thus $F(x) = 1 - \frac{1}{x^3}$ on $[1, \infty]$.

Part c. Find $P(2 \leq X \leq 3)$. This is $P(X \leq 3) - P(X \leq 2)$ or $F(3) - F(2)$. Thus

$$P(2 \leq X \leq 3) = 1 - \frac{1}{3^3} - (1 - \frac{1}{2^3}) = \frac{1}{8} - \frac{1}{27} = \frac{19}{216}$$

Using R

```r
Prob3.12cdf=function(x){1-1/(x^3)}
fractions(Prob3.12cdf(3))
```

```
## [1] 26/27
```

```r
fractions(Prob3.12cdf(2))
```

```
## [1] 7/8
```

```r
fractions(Prob3.12cdf(3)-Prob3.12cdf(2))
```

```
## [1] 19/216
```

Part d. Find $E(X)$. By definition $E(X) = \int_{-\infty}^{\infty} xf(x)dx$ Thus we get

$$E(X) = \int_1^{\infty} x\frac{3}{x^4}dx$$

$$= \lim_{a\to\infty}\int_1^a \frac{3}{x^3}dx$$

$$= \lim_{a\to\infty}\frac{3}{(-2)x^2}|_1^a$$

$$= \lim_{a\to\infty}\frac{3}{-2a^3} - \frac{3}{(-2)1^3}$$

but

$$\lim_{a\to\infty}\frac{3}{-2a^3} = 0$$

so

$$E(X) = \frac{3}{2}$$

Using R

```r
Prob3.12d<-function(x){(3/x^3)*(1 <= x)}
integrate(Prob3.12d,1,Inf)
```

```
## 1.5 with absolute error < 1.7e-14
```

```r
fractions(integrate(Prob3.12d,1,Inf)$value)
```

```
## [1] 3/2
```

Depending on the time units, this expected value tells us the average time between cars. If the unit is hours, there is not much traffic; however, it the units are seconds, then there is heavy traffic.

Part e. Find the median of X. By definition, the median is the value x such that $P(X \leq x) = .5$. Solving we have $F(x) = .5$ or $1 - \frac{1}{x^3} = .5$. Thus the median is $\sqrt[3]{2}$
Using R

```
uniroot(function(x)Prob3.12cdf(x)-.5,lower=0,upper=2)$root
```

```
## [1] 1.259923
```

```
2^(1/3)
```

```
## [1] 1.259921
```

The median tells us the time between cars such that 50% of the times are less than this value. It is close to the mean but slightly less. This tells us that the distribution is slightly skewed to the larger times.

Part f. Find $\sqrt{Var(X)}$ so we first have to find $Var(X)$. We first need $E(X^2)$ and by definition By definition

$$E(X^2) = \int_{-\infty}^{\infty} x^2 f(x)dx$$

$$E(X^2) = \int_{1}^{\infty} x^2 \frac{3}{x^4}dx$$

$$= \lim_{a\to\infty} \int_{1}^{a} \frac{3}{x^2}dx$$

$$= \lim_{a\to\infty} \frac{-3}{x}|_1^a$$

$$= \lim_{a\to\infty} \frac{-3}{a} - \frac{-3}{1^3}$$

but

$$\lim_{a\to\infty} \frac{-3}{a} = 0$$

so

$$E(X^2) = 3$$

$$Var(X) = E(X^2) - E(X)^2 = 3 - (\frac{3}{2})^2 = \frac{3}{4}$$

Thus the standard deviation is $\sqrt{\frac{3}{4}}$. Using R

```
Prob3.12f<-function(x){(3/x^2)*(1 <= x)}
fractions(integrate(Prob3.12f,1,Inf)$value)
```

```
## [1] 3
```

```
fractions(integrate(Prob3.12f,1,Inf)$value-(integrate(Prob3.12d,1,Inf)$value)^2)
```

```
## [1] 3/4
```

```
sqrt(3/4)
```

```
## [1] 0.8660254
```

Section 3.3

Homework

This is my homework for Section 3.3 of the book.

Problem 3.14 Given the pmf of the discrete uniform distribution, find the moment generating function. The probability mass function for the discrete uniform is

$$f(x) = \begin{cases} \frac{1}{n} & \text{if } x \in 1, 2, ..., n \\ 0 & \text{otherwise} \end{cases}$$

From the definition of moment generating function, we have

$$M_X(t) = E[e^{tX}] = \sum e^{tx} f(x)$$

For our problem we have

$$M_X(t) = \sum_{i=1}^{n} \frac{1}{n}(e^{tx_i}) = \frac{1}{n}(e^{1t} + e^{2t} + e^{3t} + ... + e^{nt})$$

That is the moment generating function but I will try to simplify using standard techniques from Calculus. Let

$$S = e^t + e^{2t} + ... + e^{nt}$$

Then

$$e^t S = e^{2t} + e^{3t} + ... + e^{(n+1)t}$$

and thus

$$S - e^t S = S(1 - e^t) = e^t - e^{(n+1)t}$$

so

$$S = \frac{(e^t - e^{(n+1)t})}{(1 - e^t)}$$

Finally, the moment generating function is

$$M_X(t) = \frac{1}{n} \frac{(e^t - e^{(n+1)t})}{(1 - e^t)}$$

Even though the problem does not ask for it, let's find the mean of the discrete random variable. First, I will take the derivative of the moment generating function with respect to x. This is

$$M_X^{'}(t) = \frac{1}{n}(e^t + 2e^{2t} + ...ne^{nt})$$

and now evaluate at $t = 0$, yielding

$$\mu = M_X^{'}(0) = \frac{1}{n}(e^0 + 2e^0 + ...ne^0) = \frac{1}{n}(1 + 2 + ... + n) = \frac{1}{n}\frac{n(n+1)}{2} = \frac{(n+1)}{2}$$

Problem 3.19 Given the moment generating function

$$M_X(t) = (1 - \pi_1 - \pi_2) + \pi_1 e^t + \pi_2 e^{2t}$$

find the mean and variance of X. To find the mean, take the derivative of the moment generating function and evaluate at $t = 0$.

$$M_X^{'}(t) = \pi_1 e^t + 2\pi_2 e^{2t}$$

$$\mu = M_X^{'}(0) = \pi_1 e^0 + 2\pi_2 e^0 = \pi_1 + 2\pi_2$$

To find the variance, I will use the property

$$Var(X) = E(X^2) - E(X)^2$$

Now $E(X^2)$ is

$$M_X^{''}(t) = \pi_1 e^t + 4\pi_2 e^{2t}$$
$$\mu_2^{'} = M_X^{''}(0) = \pi_1 + 4\pi_2$$

and

$$Var(X) = \sigma^2 = \mu_2^{'} - (\mu_1)^2 = \pi_1 + 4\pi_2 - (\pi_1 + 2\pi_2)^2$$

Problem 3.30(a) Given

$$M_W(t) = \left(\frac{e^t + 1}{2}\right)^{10}$$

identify the distribution. Looking at the moment generating functions in the back of the book, the moment generating function for a binomial is

$$M_X(t) = (pe^t + 1 - p)^n$$

This looks like the mgf for W where the number of trials is 10 and the probability of success is 1/2.

Section 3.4

Homework

This is my homework for Section 3.4 of the book.

Problem 3.29 Using the model that the scores are distributed as a normal with mean 500 and standard deviation of 110, I need to find the proportion or equivalently the probability that a student scores 800 or greater $P(X \geq 800)$. This students will get truncated to a score of 800. Notice that the overall distribution will be partial continuous, scores between 200 and 800, and partially discrete, there is a finite probability at the point 800. This is similar to what problem 3.3 asked you to formulate.

I could standardize this problem as $P(X \geq 800) = P\left(Z \geq \frac{800-500}{110}\right)$. I will now use R to solve the problem

```
pnorm(800,500,110,low=F)
```

```
## [1] 0.003193012
```

```
1-pnorm(800,500,110)
```

```
## [1] 0.003193012
```

```
pnorm((800-500)/110,low=F)
```

```
## [1] 0.003193012
```

Thus about 0.3% of the population of students taking the SAT will earn a perfect score.

Problem 3.31 Given $X \sim Gamma(\alpha, \lambda)$ find the distribution of $Y = 3X$. The shortcut is to use Lemma 3.4.12 which states that if $X \sim Gamma(\alpha, \lambda)$ then $3X \sim Gamma(\alpha, \lambda/3)$.

For learning, we will use the moment generating function to confirm this result.

$$M_X(t) = \left[\frac{\lambda}{\lambda - t}\right]^{\alpha}$$

Let $Y = 3X$ and so

$$M_Y(t) = E[e^{tY}] = E[e^{3tX}] = M_X(3t) = \left[\frac{\lambda}{\lambda - 3t}\right]^{\alpha}$$

simplifying

$$M_Y(t) = \left[\frac{\frac{\lambda}{3}}{\frac{\lambda}{3} - t}\right]^{\alpha}$$

which I recognize as the moment generating function for a gamma, thus $Y \sim Gamma\left(\alpha, \frac{\lambda}{3}\right)$.

since it is a good day and I am feeling well, I will also try the CDF-method. The pdf for X is

$$f(x) = \frac{\lambda^{\alpha} x^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)} \text{ for } x \geq 0$$

A modification of the CDF-method is to use the pdf directly but then to multiply by the derivative of x with respect to y, this is what happens when I take the derivative of the CDF of Y. Thus

$$X = \frac{Y}{3}$$

$$\frac{dX}{dY} = \frac{1}{3}$$

$$f_Y(y) = f_X(\frac{y}{3}) * \frac{dX}{dY} = \frac{\lambda^{\alpha} \frac{y}{3}^{\alpha-1} e^{-\lambda \frac{y}{3}}}{\Gamma(\alpha)} \frac{1}{3} \text{ for } \frac{y}{3} \geq 0$$

and simplifying

$$f_Y(y) == \frac{\left(\frac{\lambda}{3}\right)^{\alpha} y^{\alpha-1} e^{-\frac{\lambda}{3} y}}{\Gamma(\alpha)} \text{ for } y \geq 0$$

Problem 3.32 For this problem, I need the variance for the exponential, uniform, and beta distributions.

Part a.
The variance for an exponential is $Var(X) = \frac{1}{\lambda^2}$. The probability statement is $P(1 - 1 \leq X \leq 1 + 1)$

```
pexp(2,1)-pexp(0,1)
```

```
## [1] 0.8646647
```

Part b.

For this problem, the mean is 1/2 and the standard deviation is 1/2, $\sqrt{(\frac{1}{4})}$. I need $P(0 \leq X \leq 1)$

```
pexp(1/2+1/2,2)-pexp(1/2-1/2,2)
```

```
## [1] 0.8646647
```

```r
pexp(1,2)-pexp(0,2)
```

```
## [1] 0.8646647
```

Part c.

For the uniform, $E(X) = \frac{(a+b)}{2}$ and $Var(X) = \frac{(b-a)^2}{12}$

```r
punif(1/2+sqrt(1/12))-punif(1/2-sqrt(1/12))
```

```
## [1] 0.5773503
```

Part d.

For the beta, $E(X) = \frac{\alpha}{(\alpha+\beta)}$ and $Var(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

```r
pbeta(2/6+sqrt(8/(36*7)),2,4)-pbeta(2/6-sqrt(8/(36*7)),2,4)
```

```
## [1] 0.652183
```

Problem 3.33 Part a.

For a weibull, $E(X) = \beta\Gamma(1+\frac{1}{\alpha})$ and $Var(X) = \beta^2\left[\Gamma(1+\frac{2}{\alpha}) - \left(\Gamma(1+\frac{1}{\alpha})\right)^2\right]$.

The mean and variance are:

```r
3*gamma(1+1/2)
```

```
## [1] 2.658681
```

```r
3^2*(gamma(1+2/2)-gamma(1+1/2)^2)
```

```
## [1] 1.931417
```

Part b.

The median is the 0.5-quantile

```r
qweibull(.5,2,3)
```

```
## [1] 2.497664
```

Part c.

Find $P(X \leq E(X))$

```r
pweibull(3*gamma(3/2),2,3)
```

```
## [1] 0.5440619
```

Part d.

Find $P(1.5 \leq X \leq 6)$

```r
pweibull(6,2,3)-pweibull(1.5,2,3)
```

```
## [1] 0.7604851
```

Part e.

Find the probability X is within one standard deviation of the mean.

```r
Prob3.33mean=3*gamma(1+1/2)
Prob3.33stddev=sqrt(3^2*(gamma(1+2/2)-gamma(1+1/2)^2))
pweibull(Prob3.33mean+Prob3.33stddev,2,3)-pweibull(Prob3.33mean-Prob3.33stddev,2,3)
```

```
## [1] 0.6743336
```

```r
pweibull(3*gamma(3/2)+sqrt(9*(gamma(2)-gamma(3/2)^2)),2,3)-pweibull(3*gamma(3/2)-sqrt(9*(gamma(2)-gamma
```

```
## [1] 0.6743336
```

Problem 3.34 Part a

For this problem $X \sim Beta(5, 2)$. For the beta, $E(X) = \frac{\alpha}{(\alpha+\beta)}$ and $Var(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.

```r
Prob3.34mean=5/(2+5)
Prob3.34var=(2*5)/((2+5)^2*(2+5+1))
Prob3.34mean
```

```
## [1] 0.7142857
```

```r
Prob3.34var
```

```
## [1] 0.0255102
```

Part b.

The median is the 0.5-quantile

```r
qbeta(.5,5,2)
```

```
## [1] 0.73555
```

Part c.

Find $P(X \leq E(X))$

```r
pbeta(Prob3.34mean,5,2)
```

```
## [1] 0.451555
```

Part d.

Find $P(.2 \leq X \leq .4)$

```r
pbeta(.4,5,2)-pbeta(.2,5,2)
```

```
## [1] 0.03936
```

Part e.

Find the probability X is within one standard deviation of the mean.

```r
pbeta(Prob3.34mean+sqrt(Prob3.34var),5,2)-pbeta(Prob3.34mean-sqrt(Prob3.34var),5,2)
```

```
## [1] 0.6620119
```

Section 3.5 and 3.6

Homework

This is my homework for Sections 3.5 and 3.6 of the book.

Problem 3.38 This problem is easy because the author tells us the distribution by not masking the name on the y-axis. Let's reason our way through each figure

Part a.

The fact that the curve is flat on the ends means that the normal has larger values than the sampled distribution for the same quantile. Reflecting on this, this means that the normal has longer tails. In the middle the points seem to lie along the reference line along if we could zoom in we may see the points above the line when we are left of 0 on the x-axis and below the line we we are left of 0 on the x-axis. This would indicate that the sampled distribution does not have the same peak as a normal.

Part b.

The large quantile values for the sampled distribution tend to be much larger than those of the normal but on the other end, for small quantiles the sampled distribution tend to be smaller than those from the normal. This means that the distribution is skewed to the right.

Part c.

This similar to part b.

Part d.

The large and small quantile values for the sampled distribution are much larger than the normal while the middle values lie on the line. This indicates that the sampled distribution is is symmetric but with long tails as compared to the normal.

Problem 3.39 The data is count data, so we will get many repeated values. I will plot on a normal-quantile plot.

Load the fastR library first.

```r
library(fastR)
```

Quickly look at the data

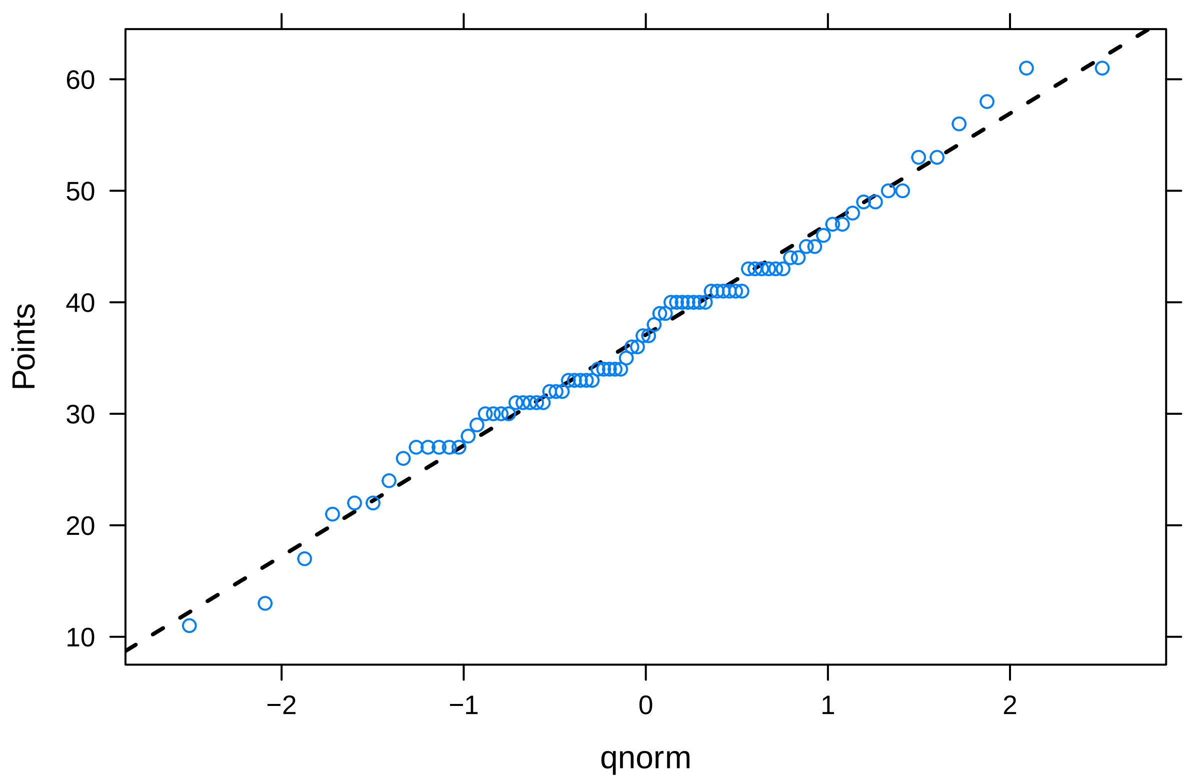
```
head(Jordan8687)
```

```
##   Game Points
## 1    1     50
## 2    2     41
## 3    3     34
## 4    4     33
## 5    5     39
## 6    6     34
```

```
summary(Jordan8687)
```

```
##        Game           Points
##  Min.   : 1.00   Min.   :11.00
##  1st Qu.:21.25   1st Qu.:31.00
##  Median :41.50   Median :37.00
##  Mean   :41.50   Mean   :37.09
##  3rd Qu.:61.75   3rd Qu.:43.00
##  Max.   :82.00   Max.   :61.00
```

```
xqqmath(~Points,Jordan8687,fitline=TRUE)
```



The fit is not bad, a normal distribution would make an acceptable model for the data.

Problem 3.40 Part a.

I will explore the data first

```r
str(pheno)
```

```
## 'data.frame':    2333 obs. of  13 variables:
##  $ id    : int  1002 1009 1012 1015 1018 1023 1032 1036 1043 1048 ...
##  $ t2d   : Factor w/ 2 levels "case","control": 1 1 2 1 2 1 1 1 1 1 ...
##  $ bmi   : num  32.9 27.4 30.5 32.5 28.3 ...
##  $ sex   : Factor w/ 2 levels "F","M": 1 1 2 2 1 1 1 1 2 2 ...
##  $ age   : num  70.8 53.9 53.9 66.3 53.9 ...
##  $ smoker: Factor w/ 4 levels "former","never",..: 1 2 1 1 4 2 2 2 1 1 ...
##  $ chol  : num  4.57 7.32 5.02 6.42 4.3 6.23 5.03 5.07 6.46 7.14 ...
##  $ waist : num  112 93.5 104 120 84 ...
##  $ weight: num  85.6 77.4 94.6 100.1 75.2 ...
##  $ height: num  161 168 176 175 163 ...
##  $ whr   : num  0.987 0.94 0.933 0.98 0.832 ...
##  $ sbp   : num  135 158 143 155 149 135 134 142 149 147 ...
##  $ dbp   : num  77 88 89 88 89 83 91 90 91 91 ...
```

```r
head(pheno)
```

```
##     id     t2d      bmi sex      age  smoker chol waist weight height
## 1 1002    case 32.85994   F 70.76438  former 4.57 112.0   85.6  161.4
## 2 1009    case 27.39085   F 53.91896   never 7.32  93.5   77.4  168.1
## 3 1012 control 30.47048   M 53.86161  former 5.02 104.0   94.6  176.2
## 4 1015    case 32.53680   M 66.27415  former 6.42 120.0  100.1  175.4
## 5 1018 control 28.30366   F 53.94632 regular 4.30  84.0   75.2  163.0
## 6 1023    case 35.19037   F 57.18630   never 6.23  98.5   90.2  160.1
##         whr sbp dbp
## 1 0.9867841 135  77
## 2 0.9396985 158  88
## 3 0.9327354 143  89
## 4 0.9795918 155  88
## 5 0.8316832 149  89
## 6 0.8140496 135  83
```

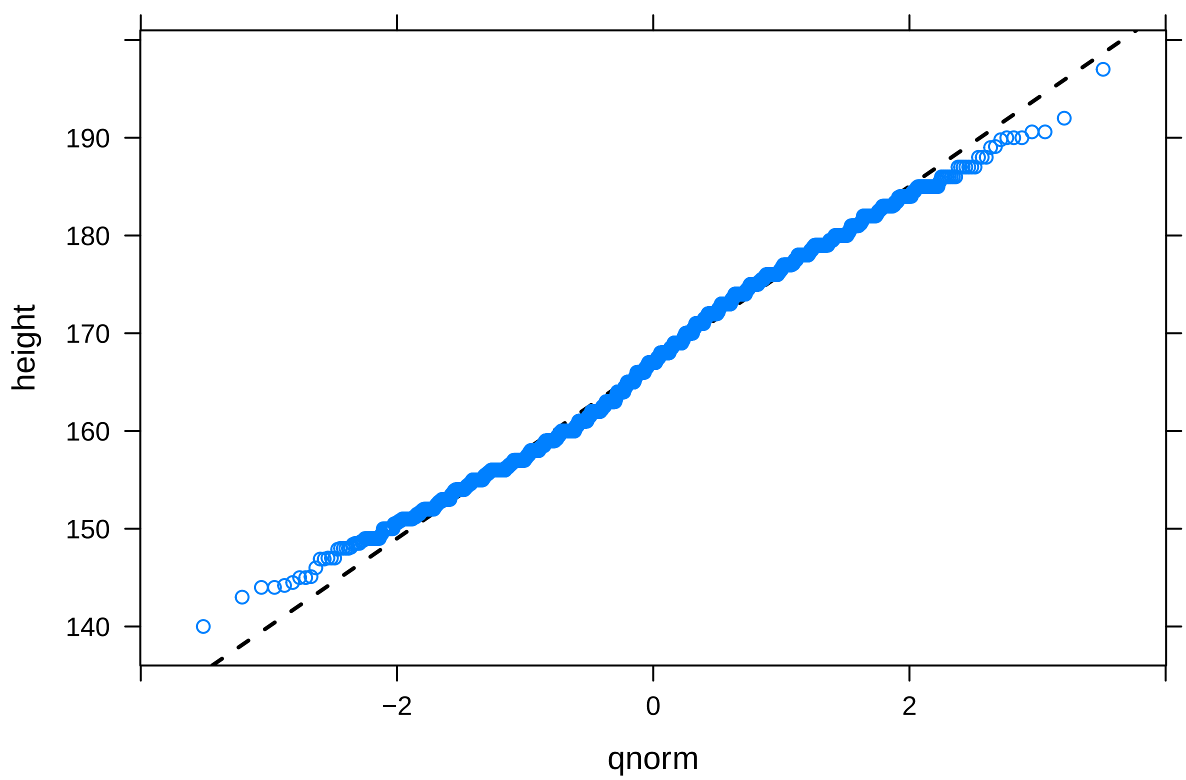
```r
summary(pheno)
```

```
##        id             t2d            bmi         sex          age
##  Min.   : 1002   case   :1161   Min.   :16.00   F:1107   Min.   :40.77
##  1st Qu.: 4027   control:1172   1st Qu.:25.40   M:1226   1st Qu.:58.00
##  Median : 6462                  Median :28.16            Median :64.00
##  Mean   : 6710                  Mean   :28.61            Mean   :63.23
##  3rd Qu.: 9764                  3rd Qu.:31.22            3rd Qu.:69.67
##  Max.   :10652                  Max.   :51.07            Max.   :84.89
##                                 NA's   :70
##        smoker          chol           waist           weight
##  former    : 352   Min.   : 2.490   Min.   : 59.00   Min.   : 35.00
##  never     : 560   1st Qu.: 5.010   1st Qu.: 87.50   1st Qu.: 69.70
##  occasional:   9   Median : 5.670   Median : 96.00   Median : 78.70
```

```
##  regular    : 115   Mean   : 5.762   Mean   : 96.49   Mean   : 79.92
##  NA's       :1297   3rd Qu.: 6.400   3rd Qu.:105.00   3rd Qu.: 89.10
##                     Max.   :15.210   Max.   :147.00   Max.   :151.10
##                     NA's   :112      NA's   :78       NA's   :69
##      height          whr               sbp              dbp
##  Min.   :140   Min.   :0.6729   Min.   : 93.5   Min.   : 39.00
##  1st Qu.:160   1st Qu.:0.8600   1st Qu.:132.0   1st Qu.: 76.00
##  Median :167   Median :0.9300   Median :145.0   Median : 83.00
##  Mean   :167   Mean   :0.9254   Mean   :147.0   Mean   : 83.12
##  3rd Qu.:174   3rd Qu.:0.9867   3rd Qu.:160.0   3rd Qu.: 90.00
##  Max.   :197   Max.   :1.2424   Max.   :237.0   Max.   :129.50
##  NA's   :84    NA's   :79       NA's   :78      NA's   :78
```

A normal-quantile plot

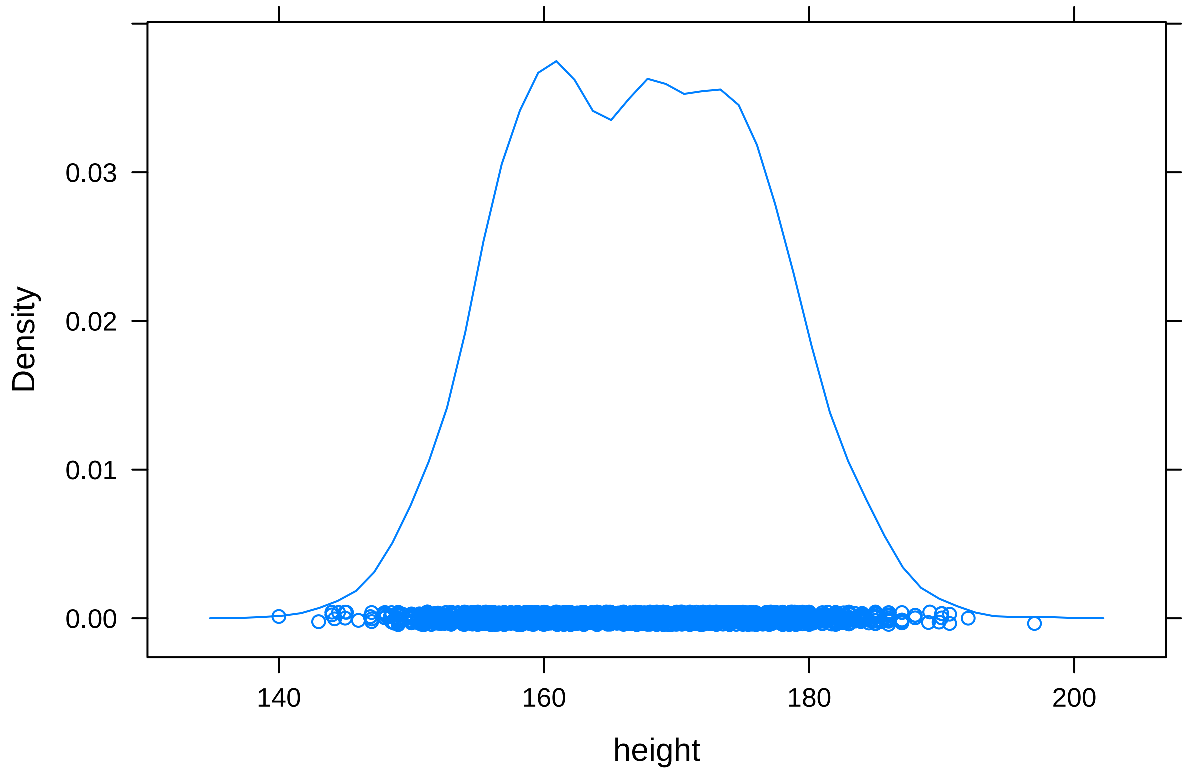
```
xqqmath(~height,pheno,fitline=TRUE)
```



This is S-shaped so the tails of a normal are longer than the observed and the normal has a sharper peak.

Next I will plot a density plot

```
densityplot(~height,pheno)
```

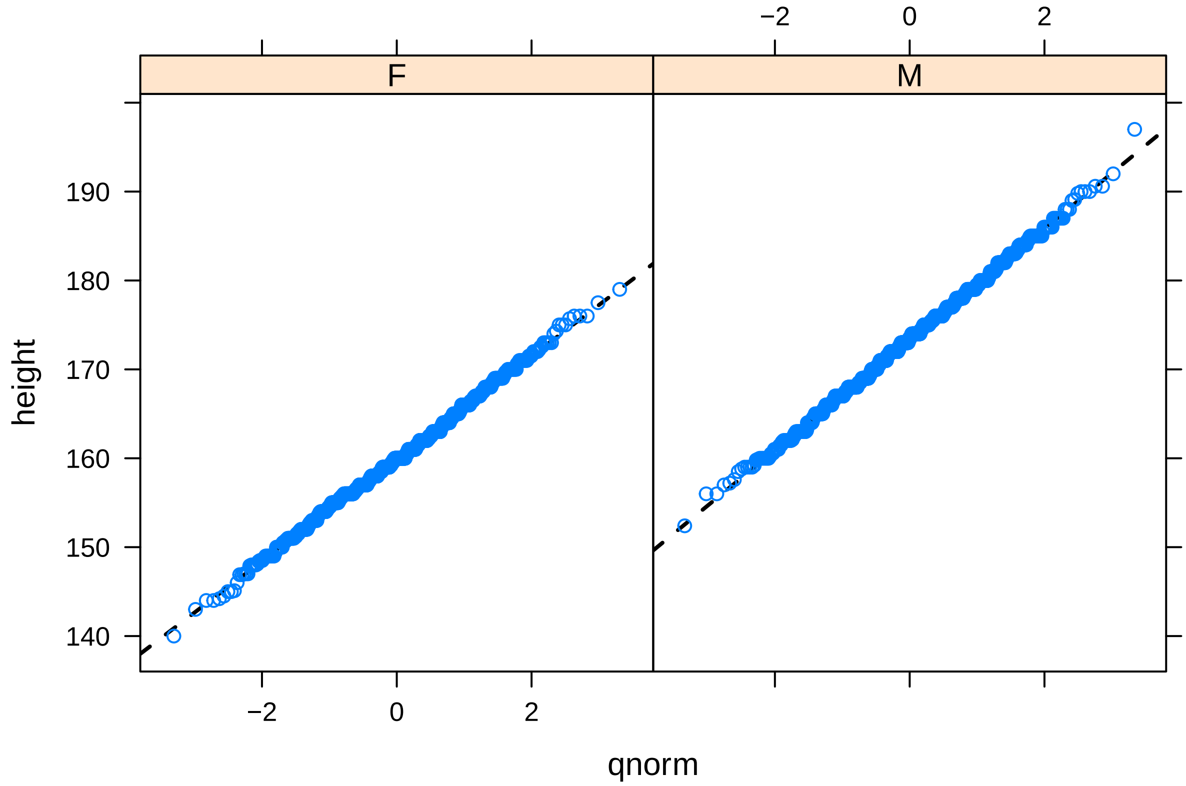


It appears the data is bi-modal.

Part b.

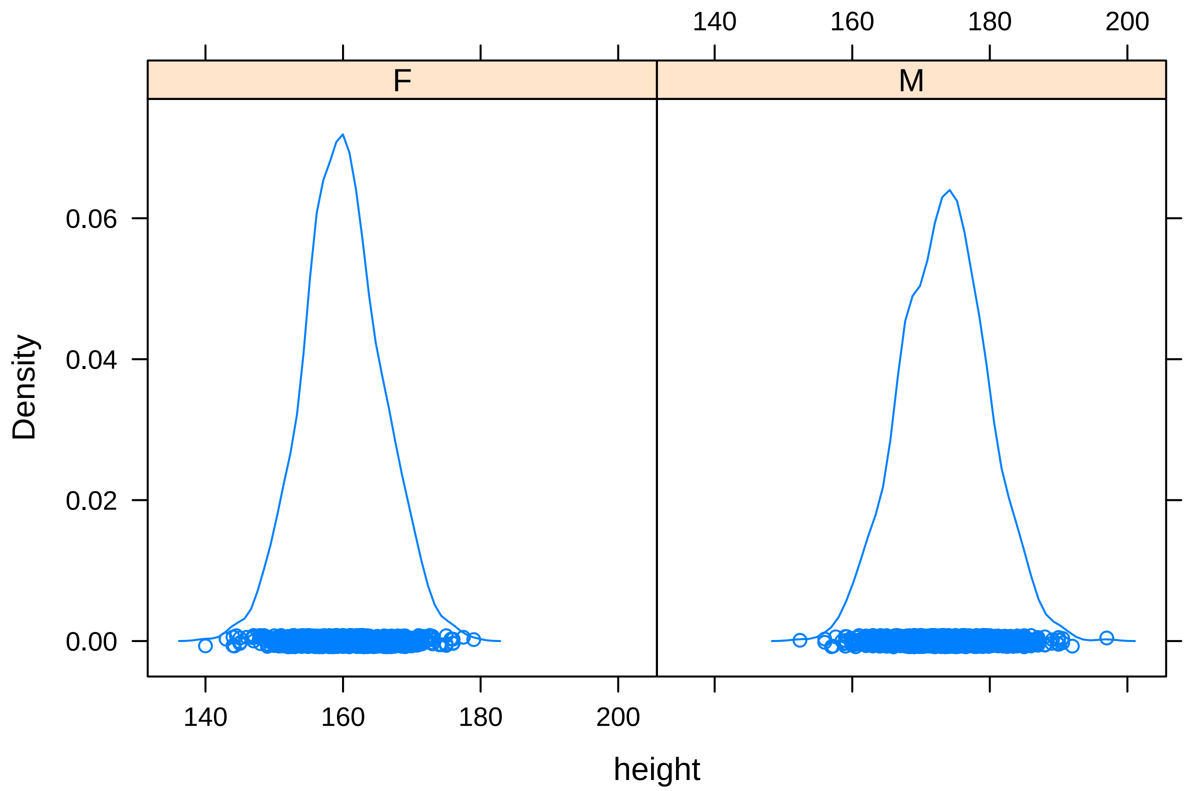
As suggested, maybe the difference between genders is leading to the bi-modal nature of the data. I will generate a normal-quantile plot for each gender.

```
xqqmath(~height|sex,pheno,fitline=TRUE)
```



And a densityplot for each gender.

```
densityplot(~height|sex,pheno)
```

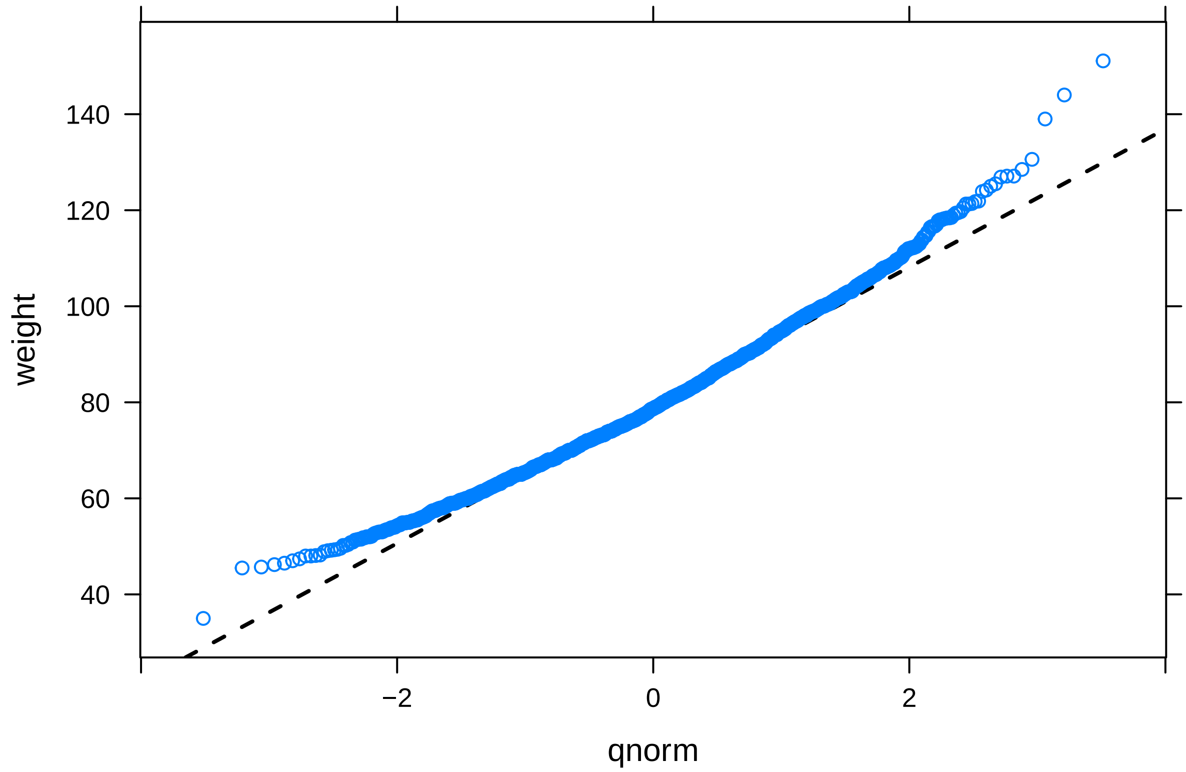


It appears that within gender, the height is normally distributed.

Problem 3.41 I am repeating problem 3.40 for weight.

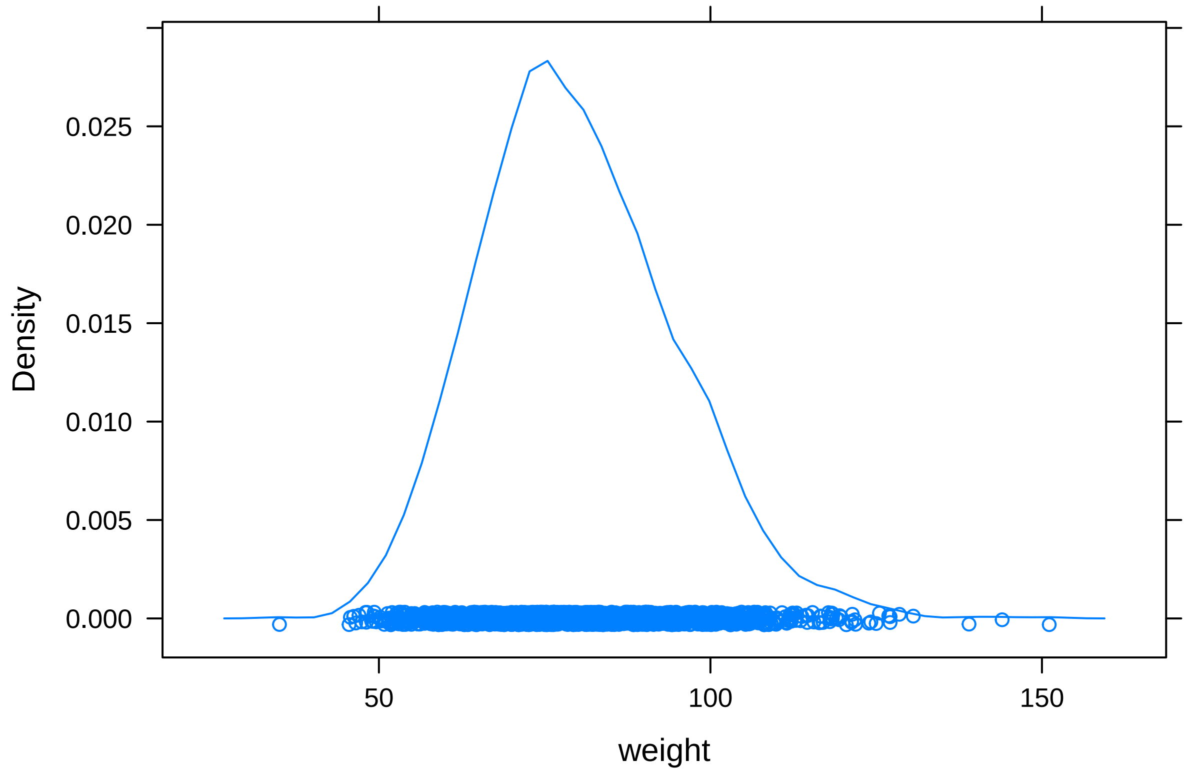
Part a.

```
xqqmath(~weight,pheno)
```



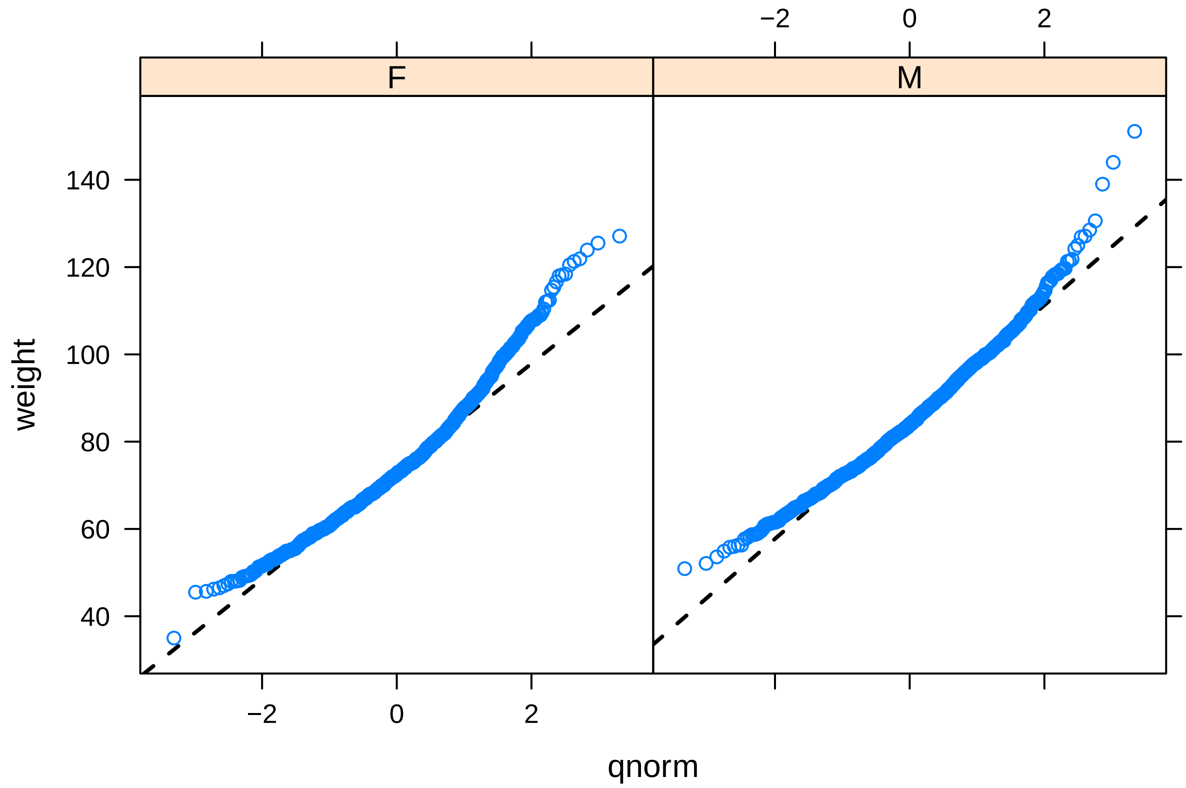
In this case it appears that the data is skewed to the right.

```
densityplot(~weight,pheno)
```

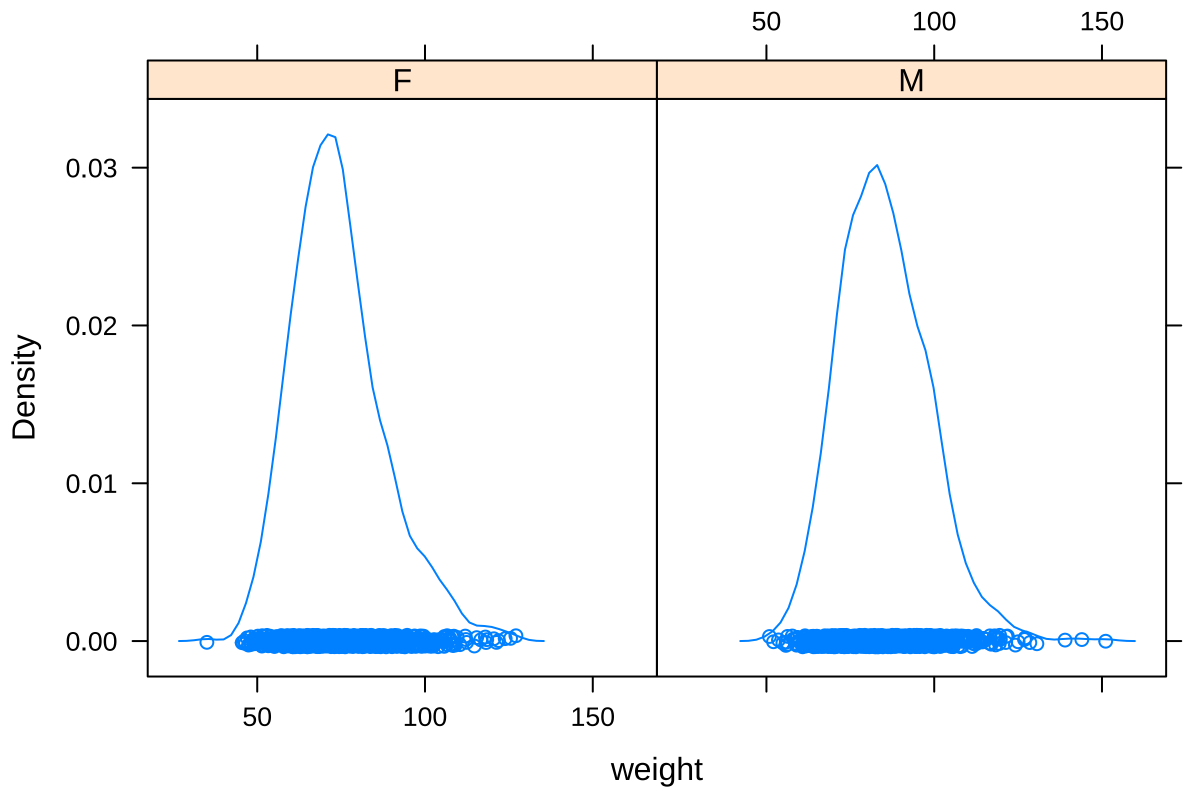


Part b.

```
xqqmath(~weight|sex,pheno)
```




```
densityplot(~weight|sex,pheno)
```



Even within gender, the weights are skewed to the right. There is a limit to how little a person can weigh relative to the mean but on the other end they can be grossly overweight.

Section 3.7

Homework

This is my homework for Sections 3.7 of the book.

Problem 3.44 Part a.

We know that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)dxdy = 1$$

Thus

$$k\int_0^1\int_0^1 x^2y^3dxdy = 1$$

$$k\int_0^1 \frac{x^3}{3}|_0^1 y^3dy = 1$$

$$\frac{k}{3}\int_0^1 y^3dy = 1$$

$$\frac{k}{3}\left(\frac{y^4}{4}|_0^1\right) = 1$$

$$\frac{k}{12} = 1$$
$$k = 12$$

Part b.

Find $P(X < Y)$.
It is helpful to plot the domain of this problem and identify the region of interest, the shaded region in the picture. First draw the line with equality and then determine which side of the line is correct for the inequality.

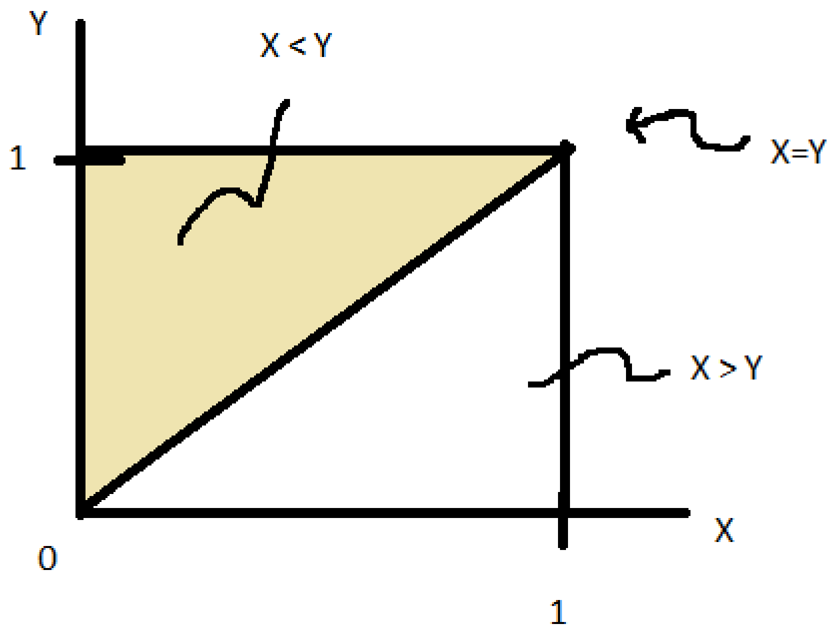


Figure 1: Figure 3.44b

A common mistake is to select the wrong region, pick a point where $X < Y$ such as (.25,.5) and plot.

Now using knowledge from Calculus 3, find the probability by integrating. If we choose to integrate x first then we have:

$$12\int_0^1\int_0^y x^2y^3dxdy$$
$$12\int_0^1 \frac{x^3}{3}|_0^y y^3dy$$
$$4\int_0^1 y^6dy$$
$$4\left(\frac{y^7}{7}|_0^1\right)$$
$$\frac{4}{7}$$

Integrating y first is a little more work:

$$12\int_0^1\int_x^1 x^2y^3dydx$$

$$12\int_0^1 \frac{y^4}{4}|_x^1 x^2dx$$

$$3\int_0^1 x^2(1-x^4)dx$$

$$3\int_0^1 (x^2-x^6)dx$$

$$3\left(\frac{x^3}{3}-\frac{x^7}{7}\right)|_0^1$$

$$3\left(\frac{1}{3}-\frac{1}{7}\right)$$

$$1-\frac{3}{7}$$

$$\frac{4}{7}$$

Part c.

Are X and Y independent?
Find $f_X(x)$:

$$f_X(x)=\int_0^1 12x^2y^3dy$$

$$f_X(x)=12x^2\left(\frac{y^4}{4}\right)|_0^1$$

$$f_X(x)=3x^2 \text{ for } 0\leq x\leq 1$$

By similar work, we find that

$$f_Y(y)=4y^3 \text{ for } 0\leq y\leq 1$$

Thus X and Y are independent because

$$f_{X,Y}(x,y)=f_X(x)f_Y(y)$$

Problem 3.45 Part a.

The joint pdf is

$$f_{X,Y}(x,y)=1 \text{ for } 5\leq x\leq 6, 5\leq y\leq 6$$

This is a uniform distribution.

Part b.

Find $P(X<5.5, Y<5.5)$.

Again, draw a picture of the domain and the region of interest.

For this problem, we have a cube of base 1/2, width 1/2, and height 1. Thus the probability, volume, is 1/4.
If you wanted to integrate, you would use

$$\int_5^{5.5}\int_5^{5.5} 1dydx$$

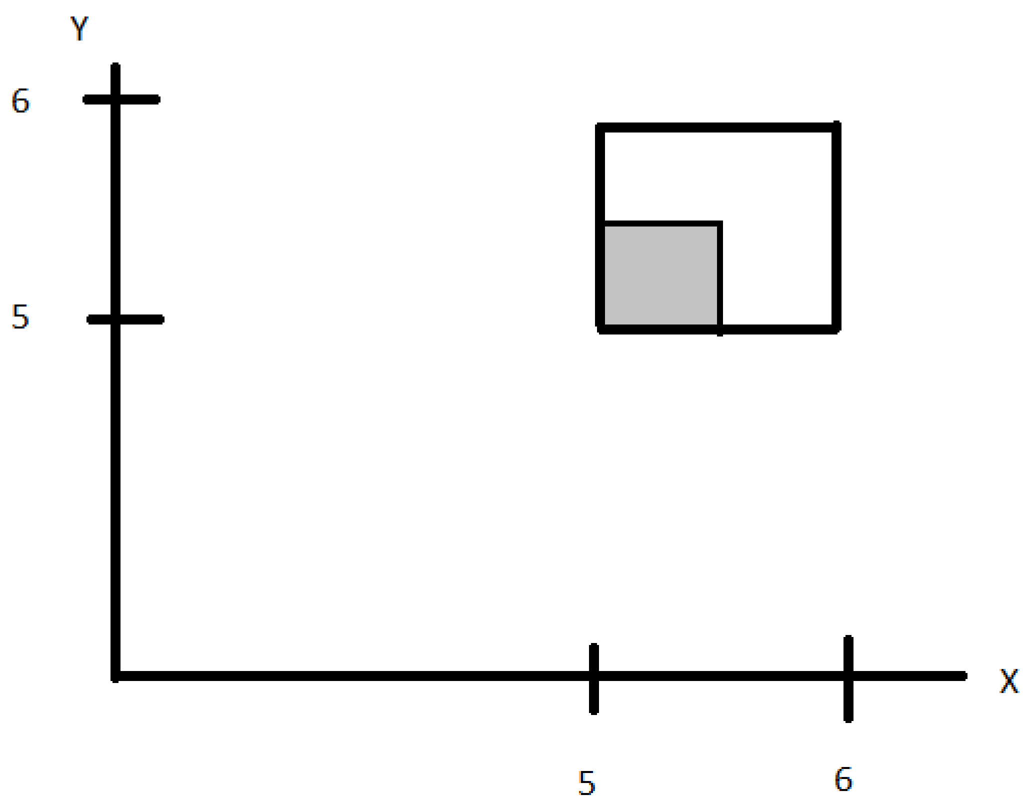


Figure 2: Figure 3.45b

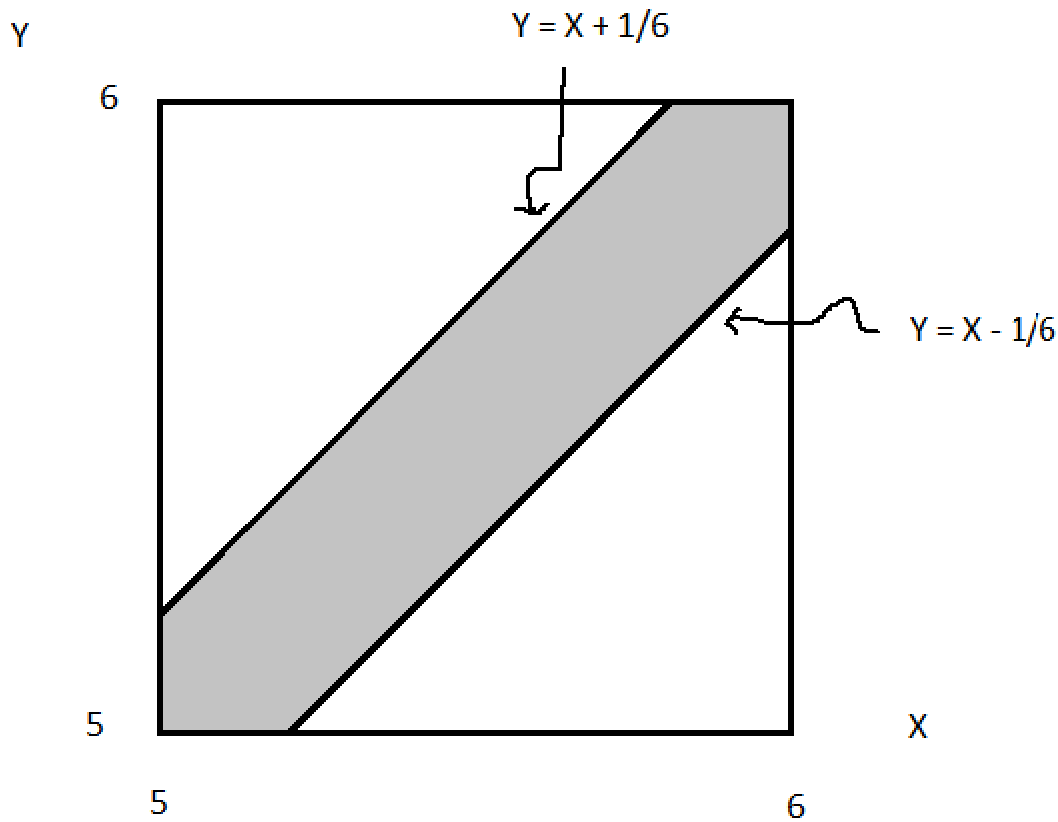


Figure 3: Figure 3.45c

Part c.

We want $P(|X-Y| \leq 10)$. A picture of the domain and region of interest is below.

By a geometric argument, it is easier to find the area of the two triangles and then subtract from 1 to get the area of the shaded region. I am interested in area since the height is 1 and I will multiply by 1 to get the volume and thus the probability.

The area of the triangle is

$$\frac{5}{6}\frac{5}{6}\frac{1}{2} = \frac{25}{72}$$

Thus the probability is

$$1 - 2\left(\frac{25}{72}\right) = \frac{11}{36}$$

Problem 3.47 Part a.

Since the volume must be one and we have a uniform the height must be the inverse of the area. Therefore, the joint pdf is

$$f_{X,Y}(x,y) = \frac{1}{\pi R^2} \text{ for } x^2 + y^2 \leq R^2$$

Part b.

Find $P(\sqrt{X^2+Y^2} \leq R/2)$ Draw a picture of the domain and region of interest.

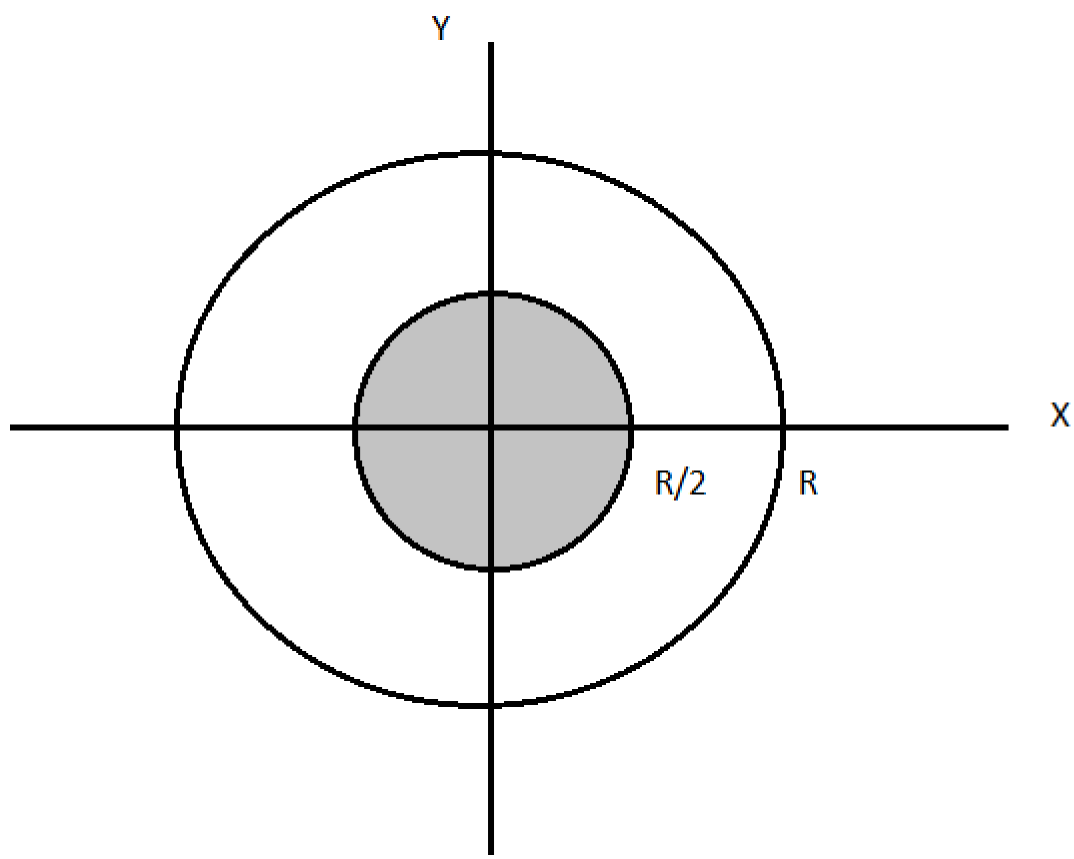


Figure 4: Figure 3.47b

By a geometric argument, the area of the shaded circle is $\frac{\pi R^2}{4}$ and thus the probability is

$$\left(\frac{\pi R^2}{4}\right)\left(\frac{1}{\pi R^2}\right)=\frac{1}{4}$$

.

Part c.

Find $P(|X-Y|\leq R)$ Draw a picture of the domain and region of interest.

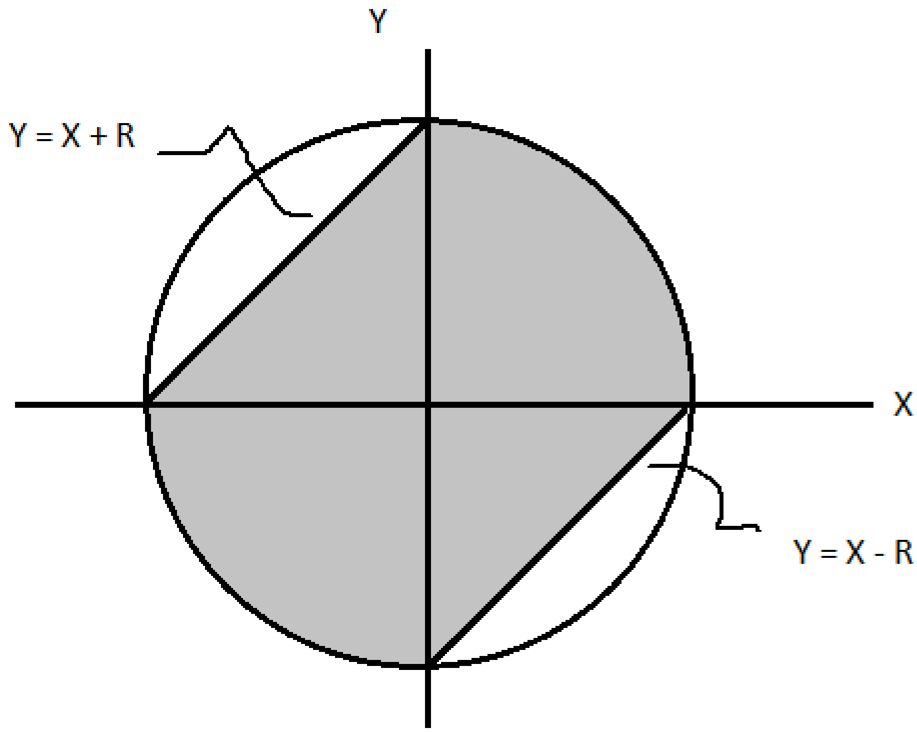


Figure 5: Figure 3.47c

By a geometric argument, we have two of the four quarters of the circle so they have volume 1/2. Next I will calculate the area of one of the triangles then double it since we have two and multiply by the height. The base and width of the triangle is R, thus it's area is $\frac{R^2}{2}$. Multiply by the pdf, the height, and doubling yields

$$2\left(\frac{R^2}{2}\right)\left(\frac{1}{\pi R^2}\right)=\frac{1}{\pi}$$

. Thus the answer is

$$P(|X-Y|\leq R)=\frac{1}{2}+\frac{1}{\pi}$$

.

Part d.

Find the marginal pdf of X.

By definition

$$f_X(x)=\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}}\frac{1}{\pi R^2}dy=\frac{1}{\pi R^2}\left(2\sqrt{R^2-x^2}\right)\ \text{for}\ -R\leq x\leq R$$

Part e.

Since the domain is not rectangular, Y depends on X and the variables are not independent.

You could also make the geometric argument that $P\left(Y > \frac{R}{\sqrt{2}}\right) > 0$ but $P\left(Y > \frac{R}{\sqrt{2}} | X > \frac{R}{\sqrt{2}}\right) = 0$ and thus not independent.

Finally, you could find the marginal of Y and demonstrate that the joint is not the product of the marginals.

Problem 3.50 Given R is Ralph's score and C is Claudia's score and

$$R \sim N(100, 20)$$

$$C \sim N(110, 15)$$

answer the questions.

Part a.

Since 150 is $2\frac{1}{2}$ standard deviations above Ralph's mean and $2\frac{2}{3}$ standard deviations above Claudia's mean, Ralph has the higher probability.

Using R

```
1-pnorm(150,100,20)
```

```
## [1] 0.006209665
```

```
1-pnorm(150,110,15)
```

```
## [1] 0.003830381
```

Part b.

Define a new random variable $D = R - C$ we want to find $P(D > 0)$. Since R and C are independent normals, their sum is normal, we could use the moment generating functions to prove this. Thus $D \sim N(100 - 110, \sqrt{400 + 225})$.

```
1-pnorm(0,-10,sqrt(400+225))
```

```
## [1] 0.3445783
```

So Claudia has the higher probability of winning.

Another way to complete this problem is to find the joint probability density function. Since R and C are independent, the joint is the product of the marginal distributions. These marginals are both the pdf for a normal and for ease of notation we will denote the respectively as $f_R(r)$ and $f_C(c)$. If we want the probability that Ralph beats Claudia we integrate the joint probability density over the region of the Cartesian plane. This is

$$\int_{-\infty}^{\infty} \int_{-\infty}^{r} f_{RC}(r, c) dc dr$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{r} f_R(r) f_C(c) dc dr$$

$$= \int_{-\infty}^{\infty} f_R(r) \int_{-\infty}^{r} f_C(c) dc dr$$
$$= \int_{-\infty}^{\infty} f_R(r) F_C(r) dr$$

Where $F_C(r)$ is the cdf of C evaluated at Ralph's score. We can integrate this with a numeric integrator. First we define our function which is a product of the pdf of R and the cdf of C.

```
RvsC<-function(x){dnorm(x,100,20)*(pnorm(x,110,15))}
```

Now integrate

```
integrate(RvsC,-Inf,Inf)$value
```

```
## [1] 0.3445783
```

Part c.

Define a new random variable $W = R+R+R-C-C-C$, note that we can't use $W = 3R-3C$ as this would just take one game and triple the score. Again, W is a normal random variable. $W \sim N(-30, \sqrt{1200+675})$.

Find $P(W > 0)$

```
1-pnorm(0,-30,sqrt(1875))
```

```
## [1] 0.2442112
```

Claudia dominates even more in this scenario.

Part d.

The addition and subtraction of normals is normal. This is the important mathematical idea. Anytime you are adding experimental results, such as scores from multiple events. If the original variables are normal, you have a method to make probability statements about the sum and difference.

Part e.

This is now a binomial T the number of games out of three that Ralph wins. The probability of success was found in part b. We want to find $P(T \geq 2)$.

```
1-pbinom(1,3,1-pnorm(0,-10,sqrt(400+225)))
```

```
## [1] 0.2743761
```

Problem 3.51 Since this is a sum of random variables, I will use the moment generating function to find the distribution. For a Poisson

$$M(t) = e^{-\lambda+\lambda e^t}$$

Thus

$$M_X(t) = e^{-\lambda_1+\lambda_1 e^t}$$
$$M_Y(t) = e^{-\lambda_2+\lambda_2 e^t}$$

Let $W = X + Y$

$$M_W(t) = E\left[e^{tW}\right] = E\left[e^{t(X+Y)}\right]$$

Since X and Y are independent

$$M_W(t) = E\left[e^{t(X+Y)}\right] = E\left[e^{tX}\right] E\left[e^{tY}\right] = e^{-\lambda_1+\lambda_1 e^t} e^{-\lambda_2+\lambda_2 e^t} = e^{-(\lambda_1+\lambda_2)+(\lambda_1+\lambda_2)e^t}$$

This is the moment generating function of a Poisson, therefore

$$W \sim Poisson\left(\lambda_1 + \lambda_2\right)$$

More generally, the sum of Poisson random variables is Poisson.

Chapter 4

Section 4.2

This is my homework for section 4.2 of the book.

Problem 4.1 Given $X \sim Binom(1, \pi)$ find a method of moment estimator for π.

We know that $E(X) = n\pi$ for a binomial. Here n is the number of trials, which is 1. It may be confusing to think of n as the number of trials and also n as the largest subscript on the random variable X. These are could be two different values and it would be better to give them different names but we will work with what was given to us. Now we have a sample size of n, so the method of moments estimator is derived by equating the sample moment with its corresponding distribution moment.

$$E(X) = \hat{\mu}_1$$

or

$$E(X) = \pi$$

and

$$\hat{\mu}_1 = \bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}$$

Therefore

$$\hat{\pi} = \bar{x}$$

Remember that the random variable X takes on the values of 0 or 1. Below is a simulation to show the merits of this estimator. I am sampling from a binomial with trial size 1 and probability of success of 0.6, this is arbitrary. The sample size is 20.

```
set.seed(111111)
sampledata<-rbinom(16,1,.6)
mean(sampledata)
```

```
## [1] 0.5625
```

For this example, our estimate of the probability of success is 0.5625. Later we will learn how to bound this estimate by accounting for its variation.

Problem 4.2 Given $X \sim Unif(-\theta, \theta)$ explain why the method of moments method is not applicable for this problem.

The expected value $E(X)$ for this uniform is $\frac{\theta+(-\theta)}{2}$ which is 0 and does not contain the parameter θ. Thus when we setup the equation for the method of moments

$$0 = \bar{x}$$

there is no parameter θ in the expression to solve for. Thus the method of moments does not produce an estimator for this problem.

You could use the second moments around the mean however. This would yield

$$\frac{(\theta - (-\theta))^2}{12} = \hat{\mu}_2^{'}$$

and thus

$$\hat{\theta} = \frac{\sqrt{12\hat{\mu}_2^{'}}}{2}$$

Problem 4.3 Given $X \sim Unif(0, \theta)$, estimate how often $\hat{\theta}$ is incorrect by meeting the condition $\hat{\theta} < max(\mathbf{X})$.

First I will create a function where I sample from a standard Uniform $Uinf(0, 1)$ and let the user pick the sample size, default is 6, and the number of simulation runs, default is 1000. I also calculate the method of moments estimator for this problem, $2\bar{x}$, and compare with the maximum value in the sample. I return the proportion of simulation runs where the estimator is less than the maximum value.

```
uni.mom=function(n=6,runs=1000,theta=1){
  counter=0
  for(i in 1:runs){
    x=runif(n,max=theta)
    if(2*mean(x)<max(x))counter=counter+1
  }
  counter/runs
}
```

Let's answer the question.

```
set.seed(2016)
uni.mom(6,10000)
```

```
## [1] 0.2245
```

```
uni.mom(12,10000)
```

```
## [1] 0.3
```

```
uni.mom(24,10000)
```

```
## [1] 0.3506
```

It appears that as the sample size increases, the probability of having an incorrect estimator also increases and is roughly in the one-third range.

I don't like my code for the function as I had to use a for loop. I want to try it again with more terse code.

```r
uni.mombest=function(n=6,runs=1000,theta=1){
  x=lapply(rep(n,runs),runif,max=theta)
  sum(2*sapply(x,mean)<sapply(x,max))/runs
}
```

and now answer the question again

```r
set.seed(2016)
uni.mombest(6,10000)
```

```
## [1] 0.2245
```

```r
uni.mombest(12,10000)
```

```
## [1] 0.3
```

```r
uni.mombest(24,10000)
```

```
## [1] 0.3506
```

Now that is some nice code but working with Sutton Hernandez we came up with

```r
uni.mommostbestest=function(n=6,runs=1000,theta=1){
  sum(apply(replicate(runs,runif(n,0,theta)),2,function(x)sum((2*mean(x)<max(x)))))/runs
}
```

```r
set.seed(2016)
uni.mommostbestest(6,10000)
```

```
## [1] 0.2245
```

```r
uni.mommostbestest(12,10000)
```

```
## [1] 0.3
```

```r
uni.mommostbestest(24,10000)
```

```
## [1] 0.3506
```

One line of code in the function! I love R.

Another method is to do the following

```r
prob4.3<-function(n=6,theta=1){
x<-runif(n,max=theta)
return(as.numeric(2*mean(x)<max(x)))
}
```

and then run in replicate

```r
set.seed(2016)
sum(replicate(10000,prob4.3(n=6)))/10000
```

```
## [1] 0.2245
```

```r
sum(replicate(10000,prob4.3(n=12)))/10000
```

```
## [1] 0.3
```

```r
sum(replicate(10000,prob4.3(n=24)))/10000
```

```
## [1] 0.3506
```

Problem 4.5 Generate a function in R to find the sample moments. Give it an option to find the moments centered around the sample mean.

```r
moment=function(k,x,centered=(k>1)){
  newx=na.omit(x)
  if(centered & k==1){
    print("Do not use a centered first sample moment")
    return()
  }
  if(centered){
    meanx=mean(newx)
    ans=sum((newx-meanx)^k)/length(newx)
  } else ans=sum(newx^k)/length(newx)
  ans
}
```

Let's test the function:

```r
set.seed(788887)
moment(1,c(3,5),centered=TRUE)
```

```
## [1] "Do not use a centered first sample moment"

## NULL
```

```r
moment(1,c(3,5))
```

```
## [1] 4
```

```r
moment(1,c(3,5,NA))
```

```
## [1] 4
```

```r
x=rnorm(20)
mean(x)
```

```
## [1] 0.1045961
```

```r
moment(1,x)
```

```
## [1] 0.1045961
```

```r
var(x)*(length(x)-1)/length(x)
```

```
## [1] 0.5394629
```

```r
moment(2,x)
```

```
## [1] 0.5394629
```

My sample moment function appears to be working well.

Problem 4.7 We are going to find a beta distribution to the free throw percentage of a basketball league. The beta should be a good model as it is used for percentages. First I will examine the data prior to starting the problem to get a better understanding of it.

```r
library(fastR)
head(miaa05)
```

```
##   Number              Player GP GS Min AvgMin  FG FGA FGPct FG3 FG3A
## 1     14 Brian Schaefer..... 25 19 769   30.8 146 366 0.399  67  185
## 2     32 Billy Collins Jr... 25 19 641   25.6 119 285 0.418  41  131
## 3      5 Mike Lewis......... 25 18 553   22.1  99 162 0.611   0    2
## 4     30 Adam Novak......... 20 13 453   22.6  95 163 0.583   3    3
## 5     24 Jeff Nokovich...... 25 17 702   28.1  38 109 0.349   7   31
## 6     44 Steve Thornton..... 22  5 356   16.2  48  84 0.571   2    4
##   FG3Pct FT FTA FTPct Off Def Tot RBG PF FO   A TO Blk Stl Pts PTSG
## 1  0.362 66  94 0.702  24  42  66 2.6 37  1  96 69   1  40 425 17.0
## 2  0.313 37  60 0.617  18  41  59 2.4 51  0  37 35   1  19 316 12.6
## 3  0.000 47  63 0.746  58  81 139 5.6 65  1  29 40   6  26 245  9.8
## 4  1.000 45  64 0.703  52  79 131 6.6 42  2  47 25   5  33 238 11.9
## 5  0.226 36  60 0.600  20  60  80 3.2 63  2 104 49   3  52 119  4.8
## 6  0.500 19  29 0.655  23  52  75 3.4 37  0  11 21  11  13 117  5.3
```

```r
str(miaa05)
```

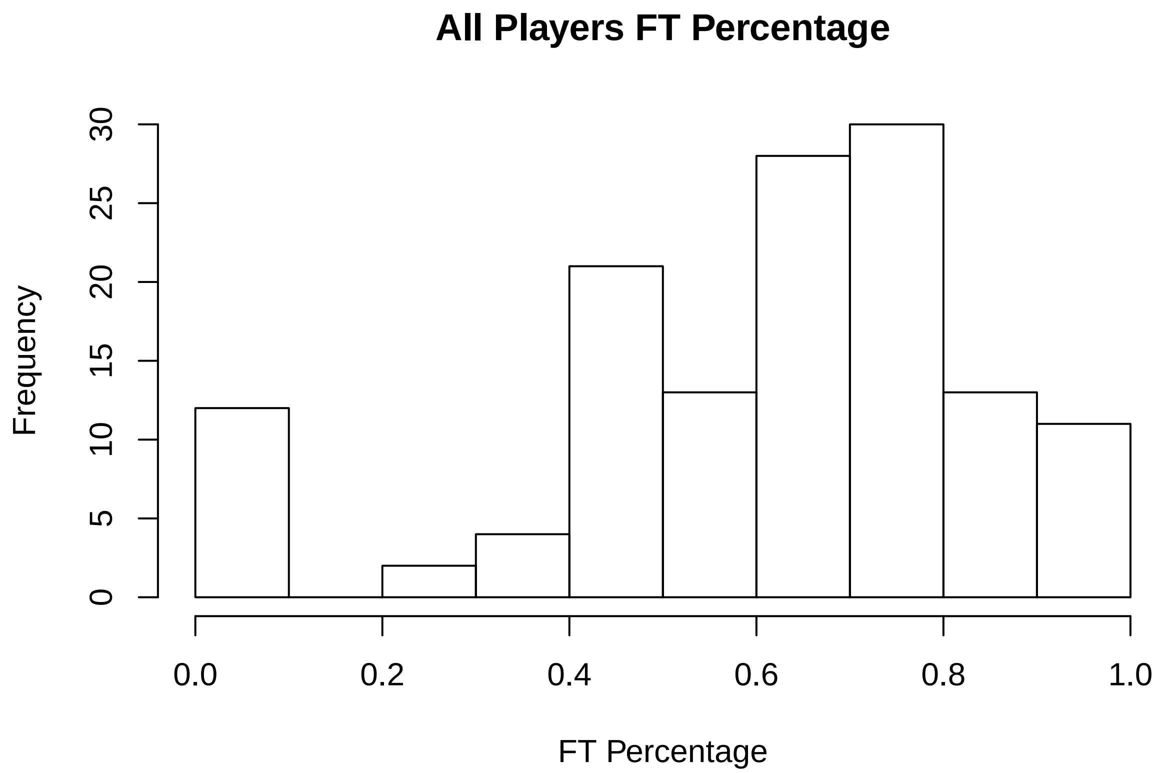
```
## 'data.frame':    134 obs. of  27 variables:
##  $ Number: int  14 32 5 30 24 44 4 34 10 22 ...
##  $ Player: Factor w/ 134 levels "Aaron Rehrer.......",..: 26 15 95 5 64 118 8 13 87 40 ...
##  $ GP    : int  25 25 25 20 25 22 25 24 23 21 ...
##  $ GS    : int  19 19 18 13 17 5 6 9 6 11 ...
##  $ Min   : int  769 641 553 453 702 356 349 361 299 228 ...
```

```
##  $ AvgMin: num  30.8 25.6 22.1 22.6 28.1 16.2 14 15 13 10.9 ...
##  $ FG    : int  146 119 99 95 38 48 30 31 33 20 ...
##  $ FGA   : int  366 285 162 163 109 84 84 93 80 42 ...
##  $ FGPct : num  0.399 0.418 0.611 0.583 0.349 0.571 0.357 0.333 0.412 0.476 ...
##  $ FG3   : int  67 41 0 3 7 2 22 18 7 0 ...
##  $ FG3A  : int  185 131 2 3 31 4 58 51 30 0 ...
##  $ FG3Pct: num  0.362 0.313 0 1 0.226 0.5 0.379 0.353 0.233 0 ...
##  $ FT    : int  66 37 47 45 36 19 10 11 5 10 ...
##  $ FTA   : int  94 60 63 64 60 29 15 14 11 28 ...
##  $ FTPct : num  0.702 0.617 0.746 0.703 0.6 0.655 0.667 0.786 0.455 0.357 ...
##  $ Off   : int  24 18 58 52 20 23 15 16 18 16 ...
##  $ Def   : int  42 41 81 79 60 52 26 41 40 25 ...
##  $ Tot   : int  66 59 139 131 80 75 41 57 58 41 ...
##  $ RBG   : num  2.6 2.4 5.6 6.6 3.2 3.4 1.6 2.4 2.5 2 ...
##  $ PF    : int  37 51 65 42 63 37 37 48 53 20 ...
##  $ FO    : int  1 0 1 2 2 0 0 2 1 0 ...
##  $ A     : int  96 37 29 47 104 11 15 19 16 11 ...
##  $ TO    : int  69 35 40 25 49 21 31 18 22 25 ...
##  $ Blk   : int  1 1 6 5 3 11 2 0 22 6 ...
##  $ Stl   : int  40 19 26 33 52 13 10 5 10 7 ...
##  $ Pts   : int  425 316 245 238 119 117 92 91 78 50 ...
##  $ PTSG  : num  17 12.6 9.8 11.9 4.8 5.3 3.7 3.8 3.4 2.4 ...
```

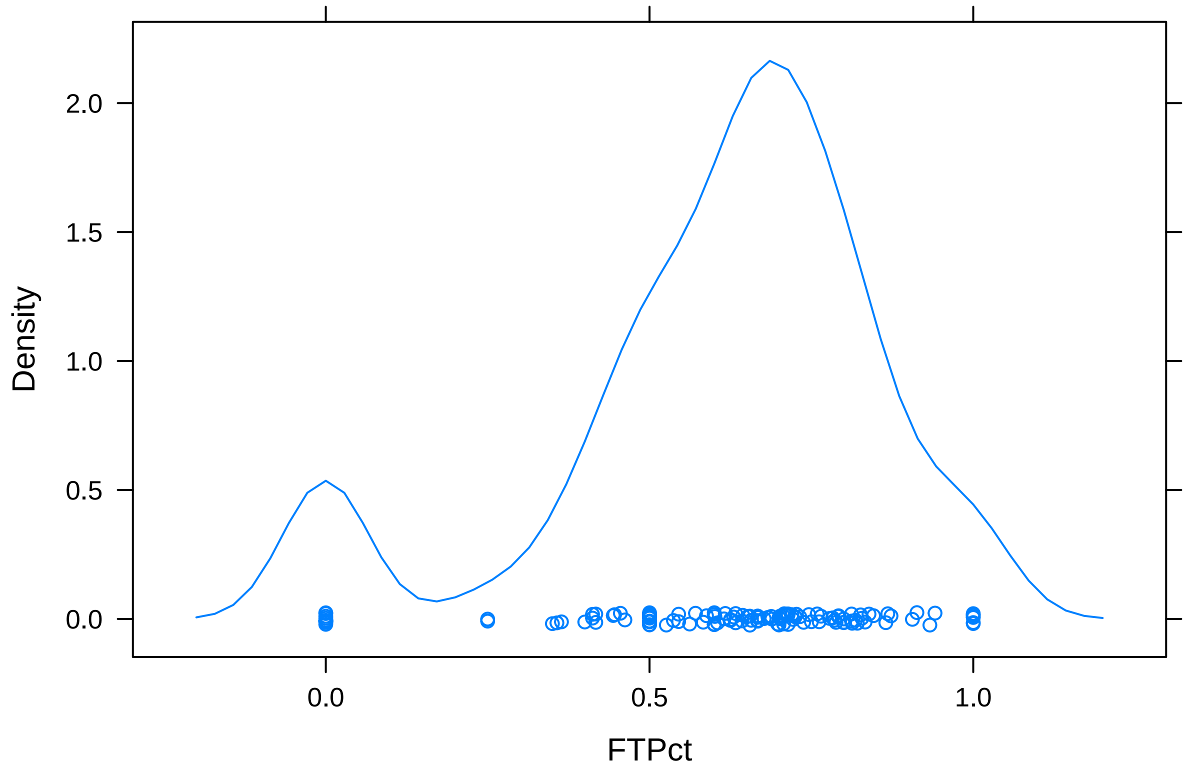
```
summary(miaa05$FTPct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.5000  0.6620  0.6091  0.7642  1.0000
```

```
hist(miaa05$FTPct,main="All Players FT Percentage",xlab="FT Percentage")
```




```
densityplot(~FTPct,miaa05)
```



There are several player who had a zero free throw percentage, this may cause some problems with our analysis. But I will proceed anyway.

A shortcut to estimating the parameters for the Beta is to use the author's code found in the snippet mom-beta01, execute

```
snippet('mom-beta01')
```

and it will add the beta.mom function to your working directory. Since I am using an RMarkdown file, the snippet command will not work and thus I will load the function directly.

```
beta.mom <- function(x,lower=0.01,upper=100) {
     x.bar <- mean (x)
     n <- length(x)
     v <- var(x) * (n-1) / n
     R <- 1/x.bar - 1

     f <- function(a){# note: undefined when a=0
         R * a^2 / ( (a/x.bar)^2 * (a/x.bar + 1) ) - v
     }

     u <- uniroot(f,c(lower,upper))

     return( c(shape1=u$root, shape2=u$root * R) )
 }
```

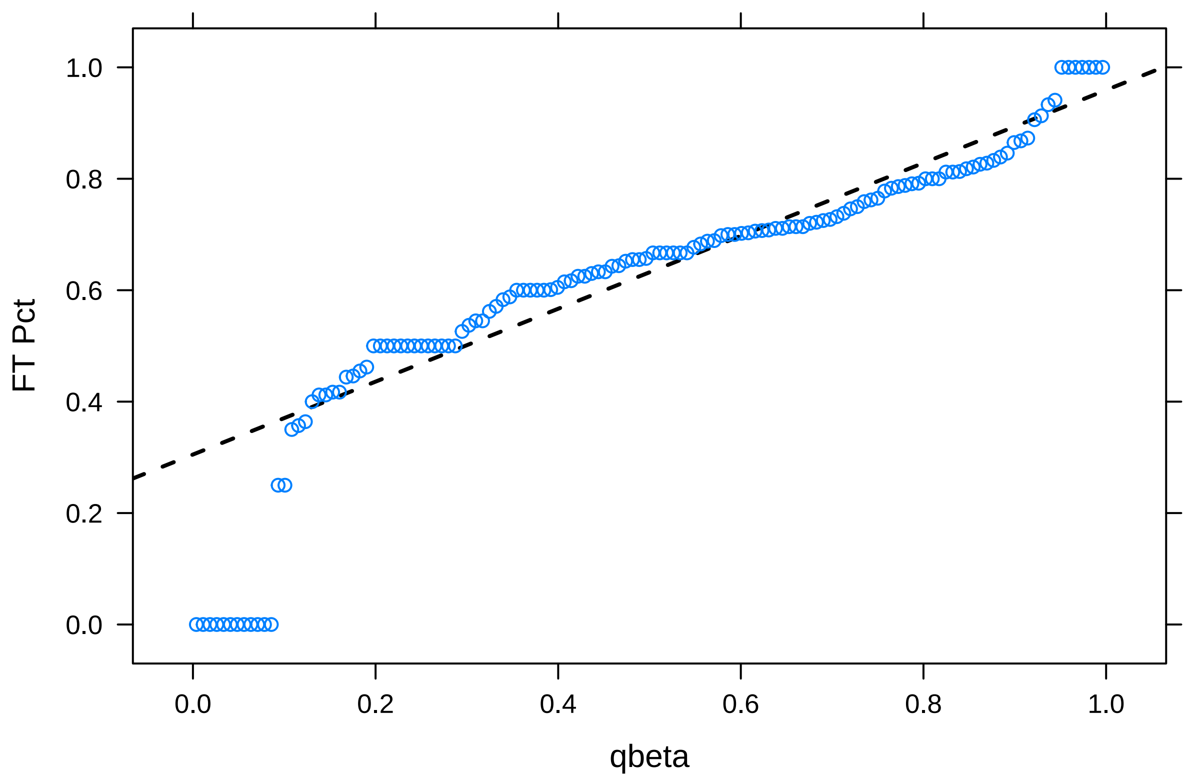
Next I will estimate the parameters of the beta distribution from the MIAA05 data.

```r
beta.mom(miaa05$FTPct)
```

```
##   shape1   shape2
## 1.766544 1.133652
```

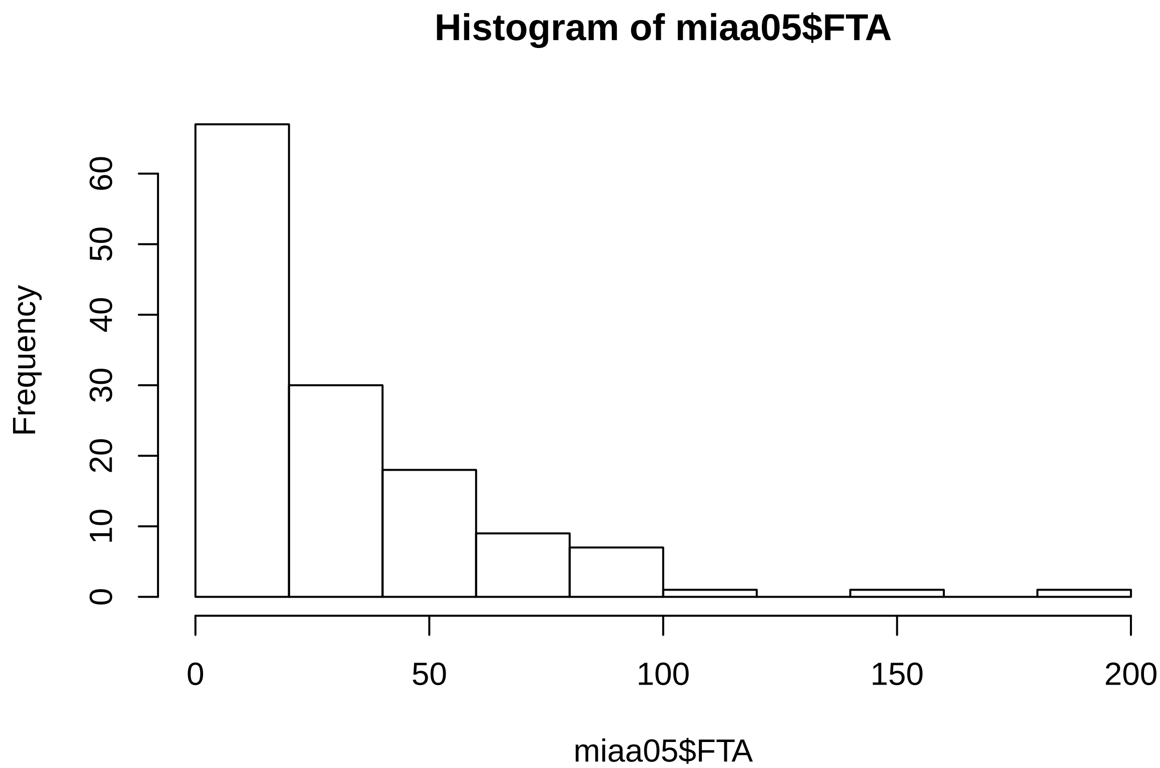
Using a quantile-quantile plot, I will access how well the data fits the theoretical model.

```r
xqqmath(~FTPct,miaa05,
  dist=function(x)qbeta(x,beta.mom(x)[1],beta.mom(x)[2]),
  xlab='qbeta',ylab="FT Pct")
```



As I suspected, there are a number of players with a 0 free throw percentage and that is impacting the fit. I want to remove players that have not attempted many free throws. The variable FTA gives me the number of free throws attempted.

```r
hist(miaa05$FTA)
```

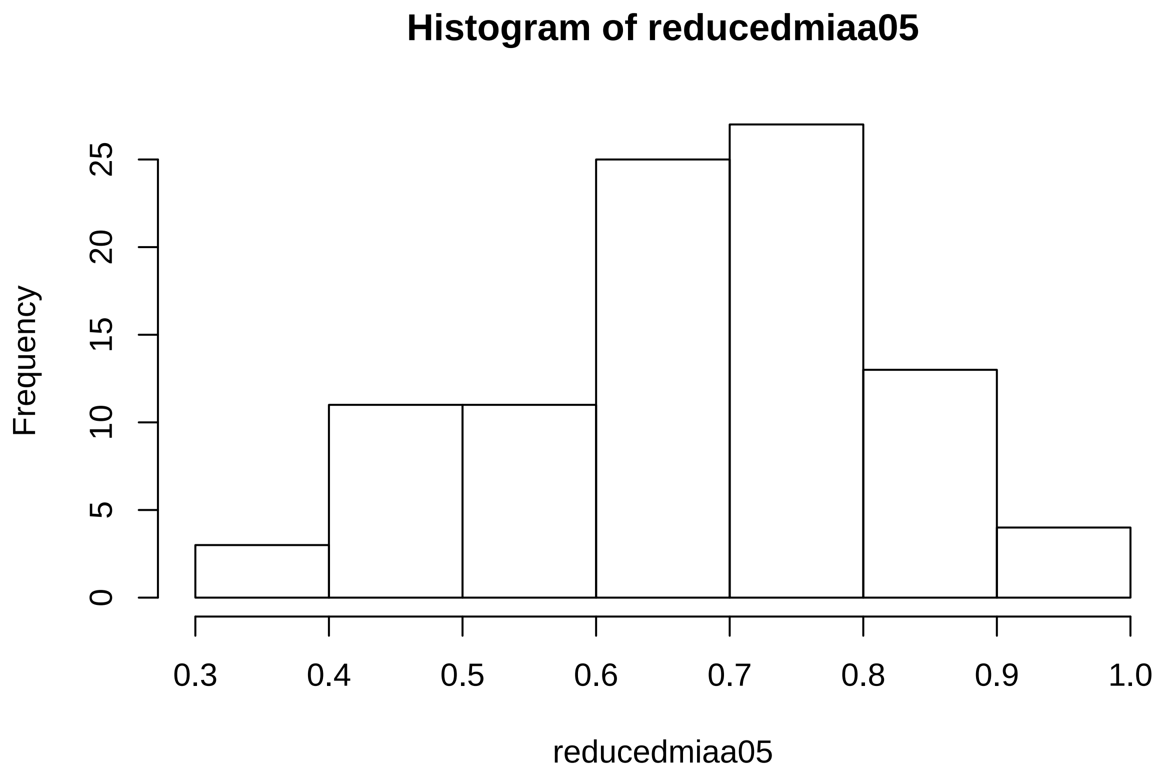



```
table(miaa05$FTA)
```

```
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
##   9   1  13   1   2   5   1   2   3   3   4   2   4   2   2   1   4   4
##  18  19  20  21  22  23  24  27  28  29  30  31  32  33  35  36  37  38
##   1   1   2   2   3   3   1   1   3   3   3   1   1   1   1   1   1   2
##  39  40  41  42  45  46  50  51  53  54  56  57  59  60  62  63  64  68
##   2   1   2   1   2   1   1   2   2   1   1   1   1   3   1   4   1   1
##  71  75  82  84  85  90  94  98  99 119 153 191
##   1   1   1   1   1   1   1   1   1   1   1   1
```

I will only keep players who have attempted 10 or more free throws.

```
reducedmiaa05<-miaa05$FTPct[miaa05$FTA>=10]
hist(reducedmiaa05)
```

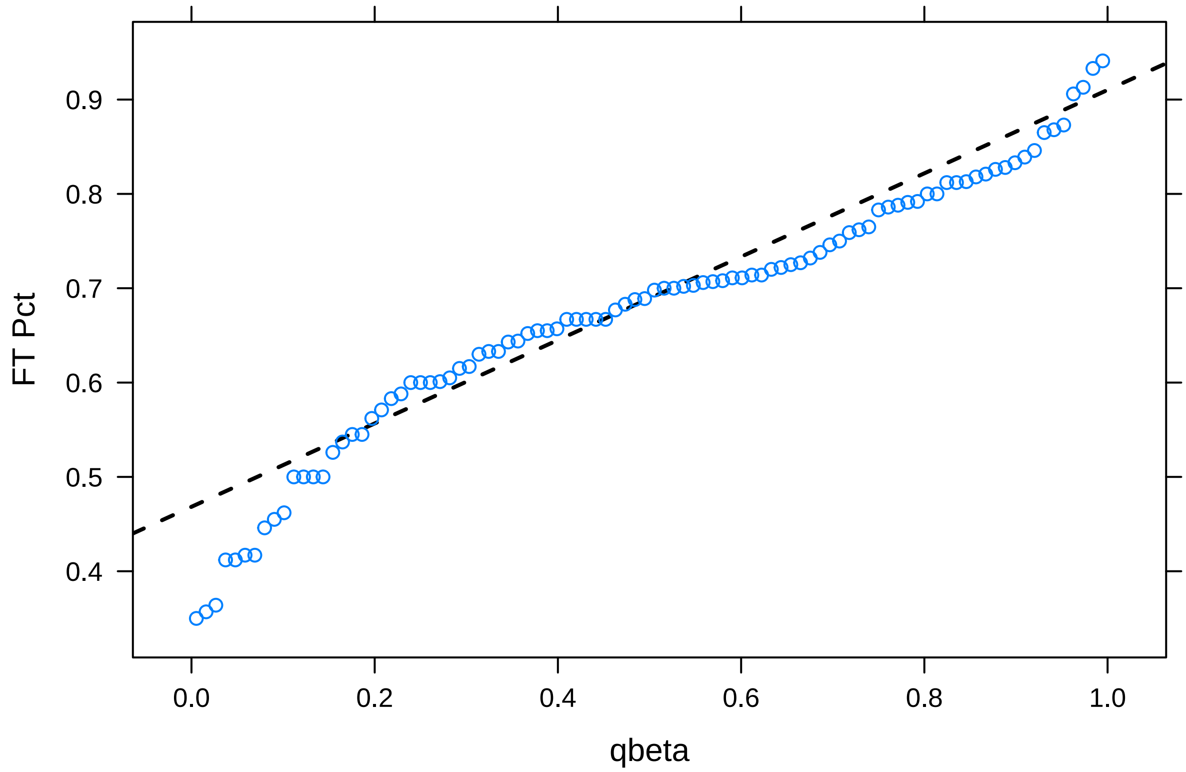


Now, I will assess the fit again.

```
beta.mom(reducedmiaa05)
```

```
##   shape1   shape2
## 7.302737 3.530133
```

```
xqqmath(~reducedmiaa05,
  dist=function(x)qbeta(x,beta.mom(x)[1],beta.mom(x)[2]),
  xlab='qbeta',ylab="FT Pct")
```



The fit is better but the data still tends to be longer in the tails as compared to the theoretical distribution.

Problem 4.9 Given the data, use method of moments to estimate parameters from an exponential and gamma.

First I will enter the data

```
Prob4.9<-c(16,34,53,75,93,120,150,191,240,339)
```

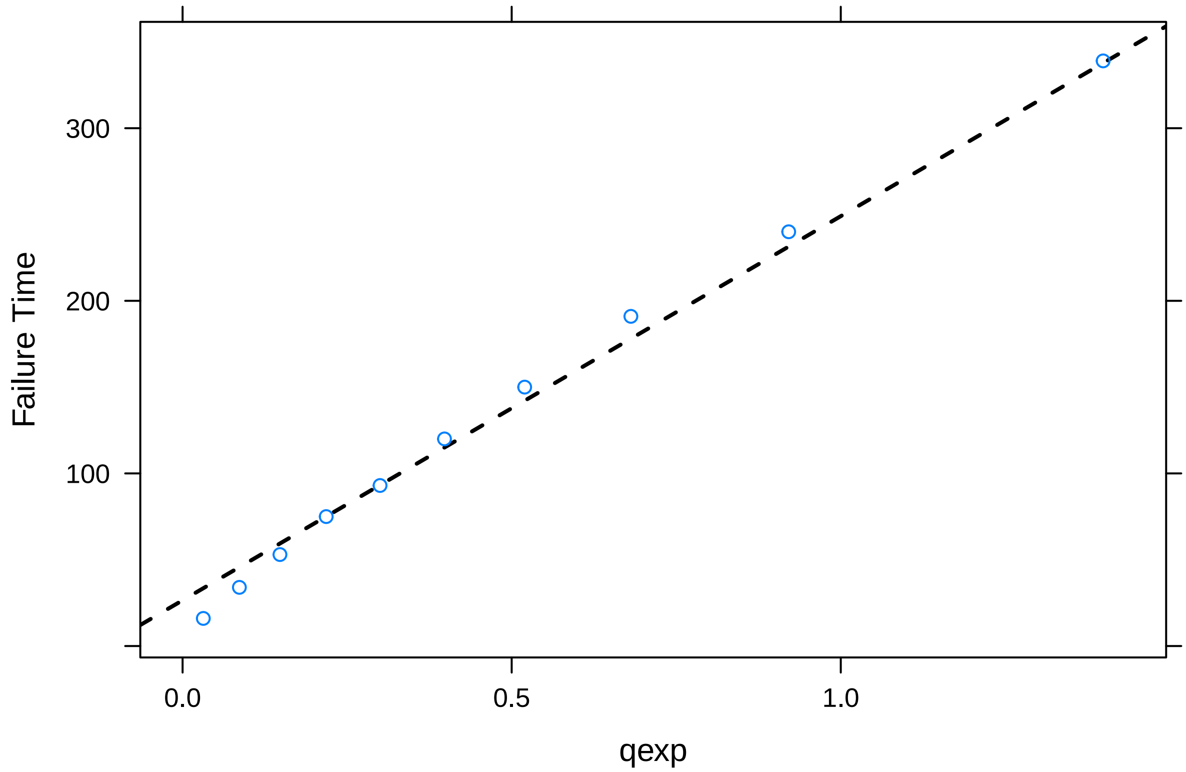
For an exponential $\hat{\lambda} = \frac{1}{\bar{x}}$ and using R

```
lamdahat=1/mean(Prob4.9);lamdahat
```

```
## [1] 0.007627765
```

I will assess the fit using a quantile-quantile plot

```
xqqmath(~Prob4.9,
        dist=function(x)qexp(x,1/mean(x)),
        xlab='qexp',ylab="Failure Time")
```



Next, I will estimate the parameters of a gamma.

First note that if $X \sim Gamma(\alpha, \lambda)$ then

$$E(X) = \mu_1 = \frac{\alpha}{\lambda}$$

and

$$Var(X) = \mu_2^{'} = \frac{\alpha}{\lambda^2}$$

Next, equating sample and population moments

$$\frac{\alpha}{\lambda} = \hat{\mu}_1 = \bar{x}$$

and

$$\frac{\alpha}{\lambda^2} = \hat{\mu}_2^{'}$$

Thus

$$\alpha = \lambda\bar{x}$$

substituting into the second expression

$$\hat{\mu}_2^{'} = \frac{\alpha}{\lambda^2} = \frac{\lambda\bar{x}}{\lambda^2}$$

Thus

$$\hat{\lambda} = \frac{\bar{x}}{\hat{\mu}_2^{'}}$$

and

$$\hat{\alpha} = \frac{\bar{x}^2}{\hat{\mu}_2^{'}}$$

In R I have, using my moment function

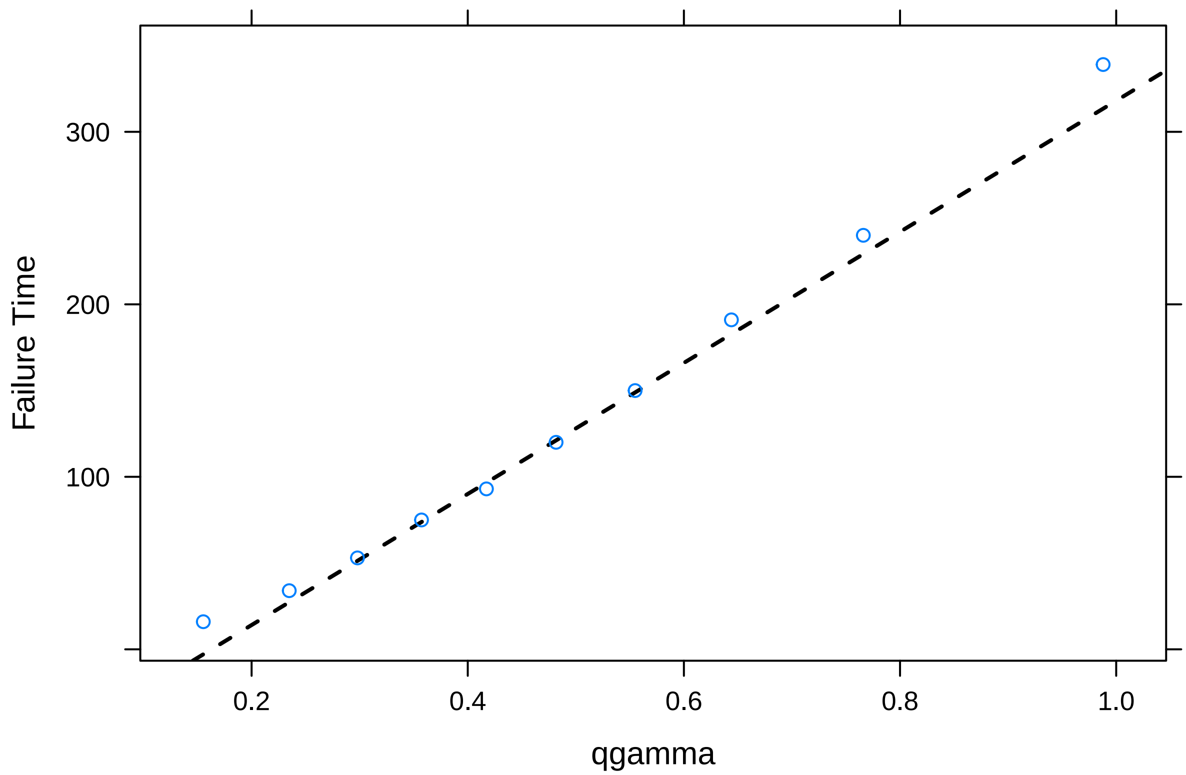
```
lamdahat=moment(1,Prob4.9)/moment(2,Prob4.9);lamdahat
```

```
## [1] 0.01416916
```

```
alphahat=lamdahat*moment(1,Prob4.9);alphahat
```

```
## [1] 1.857577
```

```
xqqmath(~Prob4.9,
 dist=function(x)qgamma(x,shape=moment(1,x)^2/moment(2,x),rate=moment(1,x)/moment(2,x)),
       xlab='qgamma',ylab="Failure Time")
```



or if you want to use the built-in functions mean and var:

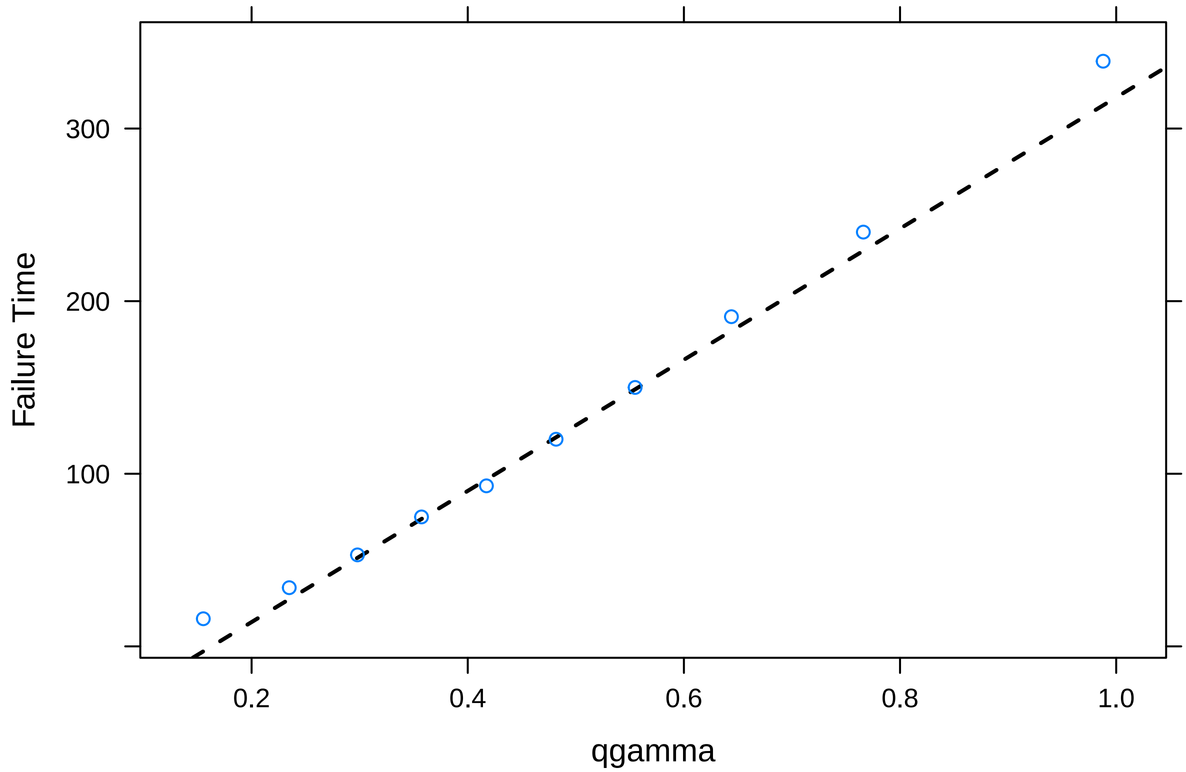
```
lamdahat=mean(Prob4.9)/var(Prob4.9)*(length(Prob4.9)/(length(Prob4.9)-1));lamdahat
```

```
## [1] 0.01416916
```

```
alphahat=lamdahat*mean(Prob4.9);alphahat
```

```
## [1] 1.857577
```

```
xqqmath(~Prob4.9,
       dist=function(x)qgamma(x,shape=mean(x)^2/var(x)*(length(x)/(length(x)-1)),rate=mean(x)/var(x)*(le
       xlab='qgamma',ylab="Failure Time")
```



The gamma appears to be a better fit.

Section 4.3

This is my homework for section 4.3 of the book.

Problem 4.11 Determine if the estimator in Examples 4.2.1 and 4.2.2 are unbiased. That is, determine from $X \sim U(0,\theta)$ and the estimate of θ as $\hat{\theta} = 2\bar{X}$ if $\hat{\theta}$ is unbiased, where X is the random uniform variable. I will determine $E(\hat{\theta})$

$$E(\hat{\theta}) = E(2\bar{X}) = 2E(\bar{X})$$

by definition

$$\bar{X} = \sum_{i=1}^{n} \frac{X_i}{n}$$

Thus

$$2E(\bar{X}) = 2E\left[\sum_{i=1}^{n} \frac{X_i}{n}\right] = \frac{2}{n}\sum_{i=1}^{n} E\left(X_i\right)$$

For a uniform $U(a,b)$

$$E(X) = \frac{a+b}{2}$$

and thus for this problem

$$E(X) = \frac{\theta}{2}$$

Substituting back in, I get

$$E(\hat{\theta}) = 2E(\bar{X}) = \frac{2}{n}\sum_{i=1}^{n} E(X_i) = \frac{2}{n}\sum_{i=1}^{n}\frac{\theta}{2} = \frac{2}{n}\frac{n\theta}{2} = \theta$$

The estimator in Example 4.2.1 and 4.2.2 is unbiased.

Problem 4.13 Let $\overrightarrow{w} = \langle w_1, w_2, .., w_n\rangle$ be a vector of fixed numbers, (weights). For a sample $\overrightarrow{X} = \langle X_1, X_2, .., X_n\rangle$ let the weighted sum be defined by

$$\bar{X}_w = \sum_{i=1}^{n} w_i X_i$$

Part a.

Find conditions on $\overrightarrow{w}$ so that $\bar{X}_w$ is an unbiased estimator of $\mu = E(X)$. By definition

$$E\left(\bar{X}_w\right) = E\left(\sum_{i=1}^{n} w_i X_i\right) = \sum_{i=1}^{n} w_i E\left(X_i\right) = \sum_{i=1}^{n} w_i \mu = \mu\sum_{i=1}^{n} w_i$$

The definition of unbiased means that

$$E\left(\bar{X}_w\right) = \mu$$

So

$$\mu = \mu\sum_{i=1}^{n} w_i$$

if

$$\sum_{i=1}^{n} w_i = 1$$

Part b.

Find $Var\left(\bar{X}_w\right)$. Since the sample is iid, independent and identically distributed

$$Var\left(\bar{X}_w\right) = Var\left(\sum_{i=1}^{n} w_i X_i\right) = \sum_{i=1}^{n} w_i^2 Var\left(X_i\right) = Var(X)\sum_{i=1}^{n} w_i^2$$

Part c. Show that the variance of the estimator is smallest when the weights are equal. First, I will examine the case when $n = 2$.

$$Var\left(\bar{X}_w\right) = w_1^2 Var\left(X\right) + w_2^2 Var\left(X\right)$$

But

$$w_1 + w_2 = 1$$

or

$$w_1 = 1 - w_2$$

Substituting back in

$$Var\left(\bar{X}_w\right) = \left(1 - w_2\right)^2 Var\left(X\right) + w_2^2 Var\left(X\right) = \left(1 - 2w_2 + w_2^2\right) Var\left(X\right) + w_2^2 Var\left(X\right) = \left(1 - 2w_2 + 2w_2^2\right) Var\left(X\right)$$

To find the minimum, I will differentiate with respect to w, since it is a continuous variable, and set equal to zero.

$$\frac{d}{dw_2}\left(1-2w_2+2w_2^2\right)Var\left(X\right)=Var\left(X\right)\left(-2+4w_2\right)=0$$

Thus it must be the case

$$(-2+4w_2)=0$$

or

$$w_2=\frac{1}{2}$$

This is a minimum because the second derivative is positive. Finally

$$w_1=1-w_2=1-\frac{1}{2}=\frac{1}{2}$$

I showed that in the case of $n=2$ the minimum variance is achieved when the weights are equal.

To prove it for the case $n>2$ I will use an induction argument. The base case, $n=2$, is true. We assume that for an arbitrary n, that the variance is a minimum when all the weights are equal. For the $n+1$ case we have

$$w_1+w_2+...+wn+w_{n+1}=1$$

but the firs n weights are equal since it minimizes the variance for the case n. Thus

$$nw+w_{n+1}=1$$

or

$$w_{n+1}=1-nw$$

Now we have

$$nw^2Var(X)+w_{n+1}^2Var(X)$$

and we want to minimize this expression. Substituting back in yields

$$nw^2Var(X)+(1-nw)^2Var(X)$$

$$nw^2Var(X)+(1-2nw+n^2w^2)Var(X)=(1-2nw+(n^2+n)w^2)Var(X)$$

Again, differentiating and setting equal to zero yields

$$-2n+2(n^2+n)w=0$$

$$w=\frac{n}{n^2+n}=\frac{n}{n(n+1)}=\frac{1}{(n+1)}$$

Thus all the weights are equal.

Problem 4.15 This is an awesome problem, and extremely useful. Let's take care of some notation first. Let θ be the true proportion of people who would answer true to version A and let π be the probability of selecting versions A.

Part a.
Letρ be the probability that a randomly selected person will answer true to the question. Thus probability of answer true is the probability someone receives version A, π and answers true, θ, these events are independent, or a person receives version B, $1-\pi$ and would answer true, $1-\theta$. The or can be handled as a sum since the events are mutually exclusive. Thus

$$\rho=\pi\theta+(1-\pi)(1-\theta)=\pi\theta+1-\pi-\theta+\pi\theta=(2\pi\theta-\theta)+(1-\pi)=(2\pi-1)\theta+(1-\pi)$$

Part b.

We can find θ by solving the equation in part a.

$$\theta = \frac{\rho + \pi - 1}{2\pi - 1}$$

Part c.

Let

$$\hat{\rho} = \frac{X}{n}$$

then show

$$E\left(\hat{\rho}\right) = \rho$$

$$E\left(\hat{\rho}\right) = E\left(\frac{\hat{X}}{n}\right) = \frac{1}{n}E(X)$$

But X is a binomial random variable with parameters n and ρ so $E(X) = n\rho$. Finally

$$E\left(\hat{\rho}\right) = \frac{1}{n}E(X) = \frac{1}{n}n\rho = \rho$$

Thus it is an unbiased estimator. This is true for either iid sampling or a simple random sample since the lack of independence in simple random sampling does not change the properties of $E(X)$.

Part d.

An natural estimator would be the result we obtained in part b but substituting the estimator from part c. Thus

$$\hat{\theta} = \frac{\hat{\rho} + \pi - 1}{2\pi - 1}$$

Now, let's see if it unbiased

$$E\left(\hat{\theta}\right) = E\left(\frac{\hat{\rho} + \pi - 1}{2\pi - 1}\right) = \frac{E\left(\hat{\rho}\right)}{2\pi - 1} + \frac{\pi - 1}{2\pi - 1}$$

But $\hat{\rho}$ is unbiased

$$E\left(\hat{\theta}\right) = \frac{\rho}{2\pi - 1} + \frac{\pi - 1}{2\pi - 1} = \theta$$

Thus

$$\hat{\theta} = \frac{\hat{\rho} + \pi - 1}{2\pi - 1}$$

is unbiased estimator of θ.

Part e.

Assuming an independent and identically distributed sample with

$$\hat{\rho} = \frac{X}{n}$$

$$Var\left(\hat{\rho}\right) = Var\left(\frac{X}{n}\right) = \frac{1}{n^2}Var(X)$$

But, again X is a binomial random variable with parameters n and ρ so $Var(X) = n\rho(1-\rho)$. Finally

$$Var\left(\hat{\rho}\right) = \frac{1}{n^2}Var(X) = \frac{1}{n^2}n\rho(1-\rho) = \frac{\rho(1-\rho)}{n}$$

Part f.

Find $Var(\hat{\theta})$. Using the results from the previous parts, we have

$$Var(\hat{\theta}) = Var\left(\frac{\hat{\rho}+\pi-1}{2\pi-1}\right)$$

$$Var(\hat{\theta}) = \left(\frac{Var\left(\hat{\rho}\right)}{(2\pi-1)^2}\right) = \frac{\rho(1-\rho)}{n(2\pi-1)^2}$$

Part g.

They are both consistent since the converge in quadratic mean. That is they are both unbiased and both have $lim_{n\to\infty}Var(\hat{\rho}) = 0$ and $lim_{n\to\infty}Var(\hat{\theta}) = 0$.

Problem 4.16 First enter the data

```
Prob4.16=c(1,2,4,4,9)
```

Part a.
I could not determine how to get the combinations I wanted with the commands in the base package of R. So I went and searched on the internet to find that I needed the package `gtools`. I will load that package first.

```
library(gtools)
```

Because of the symmetry of the problem when I am sampling without replacement, I can do this problem in two equivalent ways. First I will find all the combinations for this data set as the book suggests.

```
(Prob4.16combin<-combinations(5,2,Prob4.16,set=F))
```

```
##       [,1] [,2]
##  [1,]    1    2
##  [2,]    1    4
##  [3,]    1    4
##  [4,]    1    9
##  [5,]    2    4
##  [6,]    2    4
##  [7,]    2    9
##  [8,]    4    4
##  [9,]    4    9
## [10,]    4    9
```

Note: # Note I could use `(Prob4.16combin<-combn(Prob4.16,2))` from the base package, but this will not help me when I try the second method or for part c.

Next I want the mean of each row so I will use the apply function

```
(Prob4.16means<-apply(Prob4.16combin,1,mean))
```

```
##  [1] 1.5 2.5 2.5 5.0 3.0 3.0 5.5 4.0 6.5 6.5
```

Finally, I will calculate the mean and variance using the definition of the mean and variance of a discrete random variable, note that each has the same probability of being selected so this is a discrete uniform with probability 1/10.

```r
(mu1<-sum(Prob4.16means*1/length(Prob4.16means)))
```

```
## [1] 4
```

```r
(v1<-sum((Prob4.16means-mu1)^2/length(Prob4.16means)))
```

```
## [1] 2.85
```

Now, I was wondering why order did not matter, that is why did we use a combination. We should be able to do this using a permutation. Now each value has probability 1/20 but I get the same result. Here is the code:

```r
(Prob4.16perm<-permutations(5,2,Prob4.16,set=F))
```

```
##       [,1] [,2]
##  [1,]    1    2
##  [2,]    1    4
##  [3,]    1    4
##  [4,]    1    9
##  [5,]    2    1
##  [6,]    2    4
##  [7,]    2    4
##  [8,]    2    9
##  [9,]    4    1
## [10,]    4    2
## [11,]    4    4
## [12,]    4    9
## [13,]    4    1
## [14,]    4    2
## [15,]    4    4
## [16,]    4    9
## [17,]    9    1
## [18,]    9    2
## [19,]    9    4
## [20,]    9    4
```

```r
(Prob4.16meansa<-apply(Prob4.16perm,1,mean))
```

```
##  [1] 1.5 2.5 2.5 5.0 1.5 3.0 3.0 5.5 2.5 3.0 4.0 6.5 2.5 3.0 4.0 6.5 5.0
## [18] 5.5 6.5 6.5
```

```r
(mu1a<-sum(Prob4.16meansa*1/length(Prob4.16meansa)))
```

```
## [1] 4
```

```r
(v1a<-sum((Prob4.16meansa-mu1a)^2/length(Prob4.16meansa)))
```

```
## [1] 2.85
```

Part b.

By Corollary 4.3.3. The mean in part a should equal the mean of the original data

```r
mean(Prob4.16)
```

```
## [1] 4
```

```r
mu1
```

```
## [1] 4
```

And the variance of the sampling mean should be the population mean divided by the sample size n and multiplied by the population correction factor.

First we need the population variance

```r
(varpop<-sum((Prob4.16-mean(Prob4.16))^2/length(Prob4.16)))
```

```
## [1] 7.6
```

Now find the variance from the formula in Corollary 4.3.3

```r
varpop/2*(length(Prob4.16)-2)/(length(Prob4.16)-1)
```

```
## [1] 2.85
```

```r
v1
```

```
## [1] 2.85
```

The formulas are verified for this problem.

Part c.

For this part we need to sample with replacement so that we all repeats of the data values. Here are all the permutations with repeats allowed:

```r
permutations(5,2,Prob4.16,set=F,repeats=T)
```

```
##       [,1] [,2]
##  [1,]    1    1
##  [2,]    1    2
##  [3,]    1    4
##  [4,]    1    4
##  [5,]    1    9
##  [6,]    2    1
##  [7,]    2    2
##  [8,]    2    4
##  [9,]    2    4
## [10,]    2    9
## [11,]    4    1
## [12,]    4    2
## [13,]    4    4
## [14,]    4    4
```

```
## [15,]    4    9
## [16,]    4    1
## [17,]    4    2
## [18,]    4    4
## [19,]    4    4
## [20,]    4    9
## [21,]    9    1
## [22,]    9    2
## [23,]    9    4
## [24,]    9    4
## [25,]    9    9
```

Now I will find the means of all the samples.

```
Prob4.16cmeans<-apply(permutations(5,2,Prob4.16,set=F,repeats=T),1,mean)
```

The mean of this sample distribution is

```
(muiid<-sum(Prob4.16cmeans*1/length(Prob4.16cmeans)))
```

```
## [1] 4
```

```
mean(Prob4.16)
```

```
## [1] 4
```

Again, the same mean as the population. For variance we do not need the population correction factor.

```
sum((Prob4.16cmeans-muiid)^2/length(Prob4.16cmeans))
```

```
## [1] 3.8
```

```
varpop/2
```

```
## [1] 3.8
```

Section 4.4

This is my homework for section 4.4 of the book.

Problem 4.14 We want $P(|\bar{X} - \mu| \leq 2)$

We know that from an iid sample of size 16 from $X \sim \text{Norm}(\mu, 10)$ and that $\bar{X}$ will be $\bar{X} \sim \text{Norm}(\mu, 10/\sqrt{16})$ The distribution of $\bar{X} - \mu$ is $\bar{X} - \mu \sim \text{Norm}(0, 10/\sqrt{16})$

Thus the probability is

```
pnorm(2,0,10/4)-pnorm(-2,0,10/4)
```

```
## [1] 0.5762892
```

Problem 4.19 The wording on this problem is a little difficult, but here is the idea; we want to find the sample size n for a binomial such that the central limit theorem will be reasonable. One way to do this is to make sure that the mean $\pm$ 3 standard deviation is with the range of the binomial, which is $[0, n]$. This sample size will depend on the probability of success since decreasing the probability of success, π, while keeping the sample size, n, constant will move the center of the distribution towards the lower bound of 0. Thus n must increase. The rule of thumb is given $\pi < 0.5$ then we need $n\pi > 10$. Note, if $\pi > .5$ then we need $n(1-\pi) > 10$. Here is why:
Starting with the given

$$n\pi \geq 10$$

we have

$$n\pi \geq 10 > 9$$

so

$$\sqrt{n\pi} > 3$$

next multiplying by $\sqrt{n\pi}$ we get

$$n\pi > 3\sqrt{n\pi}$$

Since $(1-\pi) \leq 1$,

$$n\pi > 3\sqrt{n\pi} > 3\sqrt{n\pi(1-\pi)}$$

or

$$n\pi - 3\sqrt{n\pi(1-\pi)} > 0$$

Thus the mean minus 3 standard deviations will stay greater than 0 when $n\pi \geq 10$. This is the basis for the rule of thumb.

The rule of thumb was actually derived backwards from this deviation.

Problem 4.20 Given a sample size of 950 with 450 yes votes, we want to test if the true unknown population proportion is different from 0.5. Here are the hypotheses

$$H_0 : \pi = 0.5$$

$$H_A : \pi \neq 0.5$$

We will use a default level of significance of 0.05, $\alpha = 0.05$.

From what we learned in Chapter 2, we can test this using the `binom.test` function:

```
binom.test(450,950)
```

```
##
##
##
## data:  450 out of 950
## number of successes = 450, number of trials = 950, p-value =
## 0.1118
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.4415309 0.5060011
## sample estimates:
## probability of success
##              0.4736842
```

Based on the data and a level of significance of 0.05 the probability of 450 yes votes out of 950 respondents or more extreme given that the true population probability of a yes vote is 0.5 is 0.1118386. Thus we fail to reject the null hypothesis that the proportion of yes votes is 0.5, the vote is too close to call.

We could also test this claim using the central limit theorem. In this case $X \sim N(n\pi, \sqrt{n\pi(1-\pi)})$. Below is the probability calculation for half of the p-value in R:

```
pnorm(450,950*.5,sqrt(950*.5*(1-.5)))
```

```
## [1] 0.05237874
```

Since it is a two-sided test, we must double to obtain the p-value

```
2*pnorm(450,950*.5,sqrt(950*.5*(1-.5)))
```

```
## [1] 0.1047575
```

This is slightly smaller than what we got from the `binom.test`, this is because as the book discussed it is anti-conservative. Let's apply the continuity correction to improve the result.

```
2*pnorm(450+.5,950*.5,sqrt(950*.5^2))
```

```
## [1] 0.1118867
```

Instead of doing this manually, we could have used the R function '`prop.test` to get the same results

```
prop.test(450,950) #with continuity correction
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  450 out of 950
## X-squared = 2.5274, df = 1, p-value = 0.1119
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.4415808 0.5060030
## sample estimates:
##         p
## 0.4736842
```

```
prop.test(450,950,correct=F) #no continuity correction
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  450 out of 950
## X-squared = 2.6316, df = 1, p-value = 0.1048
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.4421033 0.5054771
## sample estimates:
##         p
## 0.4736842
```

Problem 4.21 Since the parent population is normal, the average will be normal. That is, if

$$X \sim N(\mu, \sigma)$$

then

$$\bar{X} \sim N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

The probability calculations are now straight forward as the actual mass will equal μ. We could use any value for the mass or we could define the probability this way

$$P(|\bar{X} - \mu| \leq 0.02)$$

or

$$P(-0.02 \leq \bar{X} - \mu \leq 0.02)$$

The distribution of $\bar{X} - \mu$ is $\sim N\left(0, \frac{0.02}{\sqrt{3}}\right)$ Now the calculation

```
pnorm(.02,mean=0,sd=.02/sqrt(3))-pnorm(-.02,mean=0,sd=.02/sqrt(3))
```

```
## [1] 0.9167355
```

Now increase the sample size to 4

```
pnorm(.02,mean=0,sd=.02/sqrt(4))-pnorm(-.02,mean=0,sd=.02/sqrt(4))
```

```
## [1] 0.9544997
```

Section 4.5

This is my homework for section 4.5 of the book.

Problem 4.22 Here is my `z.test` function, I opted for the "fancy" option in the book using `htest` class.

```
z.test<-function(x, alternative = c("two.sided", "less", "greater"),
                  mu=0,
                  sigma,
                  conf.level = 0.95, ... ){

  if(missing(sigma)) stop("You must specify a Standard Deviation of the population")

  DNAME <- deparse(substitute(x))

  alternative <- match.arg(alternative)

  n <- length(x)
  z <- (mean(x)-mu)/(sigma/sqrt(n))

  out <- list(statistic=c(z=z))
  class(out) <- 'htest'

  out$parameter <- c(n=n,"Std. Dev." = sigma,
```

```
                    "Std. Dev. of the sample mean" = sigma/sqrt(n))

  out$p.value <- switch(alternative,
                   two.sided = 2*pnorm(abs(z),lower.tail=FALSE),
                   less = pnorm(z),
                   greater = pnorm(z, lower.tail=FALSE) )

  out$conf.int <- switch(alternative,
                         two.sided = mean(x) +
                           c(-1,1)*qnorm(1-(1-conf.level)/2)*sigma/sqrt(n),
                         less = c(-Inf, mean(x)+qnorm(conf.level)*sigma/sqrt(n)),
                         greater = c(mean(x)-qnorm(conf.level)*sigma/sqrt(n), Inf)
                         )
  attr(out$conf.int, "conf.level") <- conf.level

  out$estimate <- c("mean of x" = mean(x))
  out$null.value <- c("mean" = mu)
  out$alternative <- alternative
  out$method <- "One Sample z-test"
  out$data.name <- DNAME
  names(out$estimate) <- paste("mean of", out$data.name)

  return(out)
}
```

Let's test the z.test function

```
set.seed(2098)
(testdata<-rnorm(15))
```

```
##  [1]  0.291450646  0.055964811  1.212737686 -0.007210643  0.242920352
##  [6] -1.178096425 -0.480846352 -1.182356054 -1.339077665 -1.328265308
## [11] -0.751965271 -0.193539372  1.587085258 -0.163215828  0.772041459
```

```
mean(testdata)
```

```
## [1] -0.1641582
```

```
sd(testdata)/sqrt(15)
```

```
## [1] 0.234344
```

```
z.test(testdata,sigma=1)
```

```
##
##  One Sample z-test
##
## data:  testdata
## z = -0.63578, n = 15.0000, Std. Dev. = 1.0000, Std. Dev. of the
## sample mean = 0.2582, p-value = 0.5249
```

```
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.6702187  0.3419023
## sample estimates:
## mean of testdata
##       -0.1641582
```

Problem 4.25 Part a.

This question is asking us to perform a hypothesis test. The hypothesis is

$$H_0 : \pi = 0.95$$

$$H_A : \pi \neq 0.95$$

This problem is a binomial but I estimate $\hat{\pi}$ with $\bar{X}$, thus I will use the CLT and the normal approximation. That is $\bar{X} \dot{\sim} N\left(\pi, \sqrt{\frac{\pi(1-\pi)}{n}}\right)$. I will reject if the p-value is less than 0.05. That is $P(\text{Data or more extreme} \mid \pi = .95) < .05$. This is a two-sided test so I can find this from checking just the lower tail and upper tail. I first show the lower tail and then the upper

$$P(\bar{X} < ? \mid \pi = .95) \leq .025$$

This means I need the .025 quantile of the $N\left(\pi, \sqrt{\frac{\pi(1-\pi)}{n}}\right)$.

```
qnorm(.025,mean=.95,sd=sqrt(.95*.05/10000))
```

```
## [1] 0.9457284
```

and the upper

```
qnorm(.975,mean=.95,sd=sqrt(.95*.05/10000))
```

```
## [1] 0.9542716
```

Thus if the simulated coverage rate is outside of the interval [0.9457284,0.9542716], then I reject that the true coverage rate is 0.95.

Part b.

Looking at the simulations, the beta with a sample size of 2 was too high.

Part c.

I am not sure I understand this question but I think the author wants to create a band. Thus we want to check that $\pi > .96$ or $\pi < .94$. Thus I will repeat part a but for two different hypothesis tests. First

$$H_0 : \pi = 0.96$$

$$H_A : \pi > 0.96$$

```
qnorm(.95,mean=.96,sd=sqrt(.96*.04/10000))
```

```
## [1] 0.9632232
```

and

$$H_0 : \pi = 0.94$$
$$H_A : \pi < 0.94$$

```
qnorm(.05,mean=.94,sd=sqrt(.94*.06/10000))
```

```
## [1] 0.9360937
```

Thus if the simulated coverage rate is outside of the interval [0.9360937,0.9632232], then I reject that the true coverage rate is in the interval [.94,.96]. I am concerned about combining these two hypothesis tests using .05 for both. These in some sense are simultaneous and I would need to adjust alpha for each to get an over all alpha of 0.05. A simple adjustment is to use the Bonferroni correction which we have not learned. In this case, I use $\frac{\alpha}{2}$. Thus a better answer might be

```
qnorm(.975,mean=.96,sd=sqrt(.96*.04/10000))
```

```
## [1] 0.9638407
```

```
qnorm(.025,mean=.94,sd=sqrt(.94*.06/10000))
```

```
## [1] 0.9353453
```

Thus if the simulated coverage rate is outside of the interval [0.9353453,0.9638407], then I reject that the true coverage rate is in the interval [.94,.96].

Part d.

Again, the beta of sample size 2 is suspect.

Problem 4.26 Part a.

The confidence interval is created from the form $\bar{x} \pm 1.96\text{SE}$, thus the mean is the midpoint of the interval.

```
(54.7+11.2)/2
```

```
## [1] 32.95
```

Part b.

The hypotheses would be

$$H_0 : \mu = 0$$
$$H_A : \mu \neq 0$$

Since the 95% confidence interval does not include zero, the p-value would be less than 0.05.

Problem 4.27 I think my friend is completely wrong. We are talking about the mean weight of the birds and not an individual bird weight. In fact, we do not know the distribution of the weights of individual birds. We are 95% confident that the mean weight of a bird is between 1363 and 1727 grams.

Problem 4.28 The 95% one-sided upper confidence interval places all 0.05% in the upper tail, so it is

```r
10+qnorm(.95)*3/5
```

```
## [1] 10.98691
```

The interval would be $(-\infty$,10.9869).

And the 95% one-sided lower bound is

```r
10-qnorm(.95)*3/5
```

```
## [1] 9.013088
```

Notice to test

$$H_0 : \mu = \mu_0$$
$$H_A : \mu < \mu_0$$

we need the one-sided upper confidence bound of the form

$$(-\infty, L)$$

since if $L < \mu_0$, we could reject. If we used a one-sided lower confidence bound of the form (L, ∞) then if $L > \mu_0$ we would fail to reject and if $L < \mu_0$ we are inconclusive as it is possible for both $\mu < \mu_0$ and $\mu > \mu_0$.

Similar analysis could be performed for

$$H_0 : \mu = \mu_0$$
$$H_A : \mu > \mu_0$$

Section 4.6 and 4.7

This is my homework for section 4.6 and 4.7 of the book.

Problem 4.30 Show that if $\hat{\theta}^2$ is an unbiased estimator of θ^2 then in all interesting cases $\hat{\theta}$ is a biased estimator of θ.

First from the definition of unbiased, we have

$$\hat{\theta}^2 = \theta^2$$

and the definition of variance

$$Var(\hat{\theta}) = E(\hat{\theta}^2) - E(\hat{\theta})^2$$

. This is the key step, the penultimate step, in the derivation. I came to understand that I need this because I need a link between $\hat{\theta}^2$ and $\hat{\theta}$.
Now, a trivial estimator is one that has no variance, it does not change with the data. Thus all interesting cases we have

$$Var(\hat{\theta}) > 0$$

Since $\hat{\theta}^2$ is an unbiased estimator of θ^2, we have

$$Var(\hat{\theta}) = \theta^2 - E(\hat{\theta})^2$$

If $\hat{\theta}$ were unbiased we would have

$$Var(\hat{\theta}) = \theta^2 - E(\hat{\theta})^2 = \theta^2 - \theta^2 = 0$$

Thus $\hat{\theta}$ must be biased.

Since sample variance is an unbiased estimator, sample standard deviation is a biased estimator.

Problem 4.39 Generate 95% confidence intervals for sepal width for each of the species in the `iris` data set.

Let's look at the data first

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Since I don't know the variance, I must use a t-test to find the confidence intervals.

```
tapply(iris$Sepal.Width,iris$Species,t.test)
```

```
## $setosa
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 63.946, df = 49, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  3.320271 3.535729
## sample estimates:
## mean of x
##     3.428
##
##
## $versicolor
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 62.419, df = 49, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  2.68082 2.85918
## sample estimates:
## mean of x
##      2.77
##
##
## $virginica
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 65.208, df = 49, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
```

```
## 95 percent confidence interval:
##  2.882347 3.065653
## sample estimates:
## mean of x
##     2.974
```

If you want to subset and just get the intervals, we could use

```
temp<-tapply(iris$Sepal.Width,iris$Species,t.test)
temp[1][[1]]$conf.int
```

```
## [1] 3.320271 3.535729
## attr(,"conf.level")
## [1] 0.95
```

```
temp[2][[1]]$conf.int
```

```
## [1] 2.68082 2.85918
## attr(,"conf.level")
## [1] 0.95
```

```
temp[3][[1]]$conf.int
```

```
## [1] 2.882347 3.065653
## attr(,"conf.level")
## [1] 0.95
```

From these confidence, since they do not overlap, we can say that the mean sepal width for each species is different.

```
tapply(iris$Sepal.Length/iris$Sepal.Width,iris$Species,t.test)
```

Problem 4.41

```
## $setosa
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 87.544, df = 49, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  1.436439 1.503936
## sample estimates:
## mean of x
##  1.470188
##
##
```

```
## $versicolor
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 66.809, df = 49, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  2.095418 2.225386
## sample estimates:
## mean of x
##  2.160402
##
##
## $virginica
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 63.855, df = 49, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  2.160258 2.300647
## sample estimates:
## mean of x
##  2.230453
```

Using the ratio of sepal length to sepal width, setosa is different from versicolor and virginica but the latter two are possible not different.

Problem 4.42 Part a.

To compute a one-sided confidence interval when σ is unknown, use the t distribution instead of the z and use $\frac{s}{\sqrt{n}}$ instead of $\frac{\sigma}{\sqrt{n}}$. Thus a one-sided upper confidence interval when the population standard deviation is unknown is

$$\bar{x} + t_{1-\alpha,n-1}\frac{s}{\sqrt{n}}$$

Part b.

Here is my code in R

```r
length(iris$Sepal.Length[iris$Species=="versicolor"])
```

```
## [1] 50
```

```r
mean(iris$Sepal.Length[iris$Species=="versicolor"])+qt(.95,49)*sd(iris$Sepal.Length[iris$Species=="vers
```

```
## [1] 6.058384
```

Part c.

Checking using the `t.test` command

```r
t.test(iris$Sepal.Length[iris$Species=="versicolor"],alt="less")
```

```
##
##  One Sample t-test
##
## data:  iris$Sepal.Length[iris$Species == "versicolor"]
## t = 81.318, df = 49, p-value = 1
## alternative hypothesis: true mean is less than 0
## 95 percent confidence interval:
##      -Inf 6.058384
## sample estimates:
## mean of x
##     5.936
```

Notice that I had to use lower for the hypothesis test to generate an upper bound.

Problem 4.44 Let's examine the data set first

```r
str(chickwts)
```

```
## 'data.frame':    71 obs. of  2 variables:
##  $ weight: num  179 160 136 227 217 168 108 124 143 140 ...
##  $ feed  : Factor w/ 6 levels "casein","horsebean",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```r
head(chickwts)
```

```
##   weight      feed
## 1    179 horsebean
## 2    160 horsebean
## 3    136 horsebean
## 4    227 horsebean
## 5    217 horsebean
## 6    168 horsebean
```

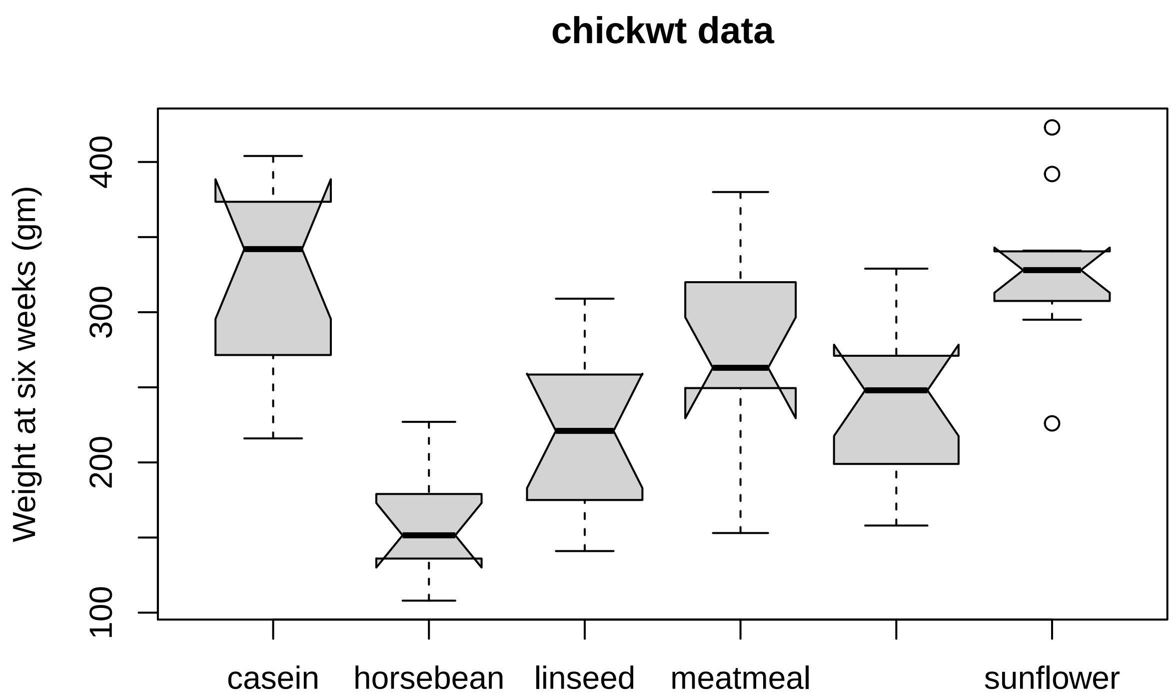
```r
summary(chickwts)
```

```
##      weight             feed
##  Min.   :108.0   casein   :12
##  1st Qu.:204.5   horsebean:10
##  Median :258.0   linseed  :12
##  Mean   :261.3   meatmeal :11
##  3rd Qu.:323.5   soybean  :14
##  Max.   :423.0   sunflower:12
```

And visually

```r
boxplot(weight ~ feed, data = chickwts, col = "lightgray",
        varwidth = TRUE, notch = TRUE, main = "chickwt data",
        ylab = "Weight at six weeks (gm)")
```

```
## Warning in bxp(structure(list(stats = structure(c(216, 271.5, 342, 373.5, :
## some notches went outside hinges ('box'): maybe set notch=FALSE
```



Part a

Generate 95% confidence intervals for each feed weight. Notice that the boxplots already did this for us and allows a comparison visually. I will use the `tapply` command

```r
tapply(chickwts$weight,chickwts$feed,t.test)
```

```
## $casein
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 17.397, df = 11, p-value = 2.373e-09
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  282.6440 364.5226
## sample estimates:
## mean of x
##  323.5833
##
##
## $horsebean
##
```

```
##  One Sample t-test
##
## data:  X[[i]]
## t = 13.115, df = 9, p-value = 3.599e-07
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  132.5687 187.8313
## sample estimates:
## mean of x
##     160.2
##
##
## $linseed
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 14.507, df = 11, p-value = 1.62e-08
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  185.561 251.939
## sample estimates:
## mean of x
##    218.75
##
##
## $meatmeal
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 14.151, df = 10, p-value = 6.113e-08
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  233.3083 320.5099
## sample estimates:
## mean of x
##  276.9091
##
##
## $soybean
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 17.034, df = 13, p-value = 2.848e-10
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  215.1754 277.6818
## sample estimates:
## mean of x
##  246.4286
##
##
```

```
## $sunflower
##
##  One Sample t-test
##
## data:  X[[i]]
## t = 23.331, df = 11, p-value = 1.018e-10
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  297.8875 359.9458
## sample estimates:
## mean of x
##  328.9167
```

The printout is long, so we could simplify using our own confidence interval function

```
myt.ci=function(data, conf.level = 0.95,alternative = c("two.sided", "less", "greater")) {
   DNAME<-deparse(substitute(data))
   df = length(data) - 1
   n <- length(data)
   stdev<-sd(data)
   alternative <- match.arg(alternative)
   CONFINT <- switch(alternative,
                         two.sided = mean(data) +
                           c(-1,1)*qt(1-(1-conf.level)/2,df)*stdev/sqrt(n),
                         less = c(-Inf, mean(data)+qt(conf.level,df)*stdev/sqrt(n)),
                         greater = c(mean(data)-qt(conf.level,df)*stdev/sqrt(n), Inf)
   )
   attr(CONFINT,"conf.level")<-conf.level
   structure(list(conf.int=CONFINT,data.name=DNAME),class="htest")
 }
```

and then apply to the data set

```
tapply(chickwts$weight,chickwts$feed,myt.ci)
```

```
## $casein
##
##
##
## data:  X[[i]]
##
## 95 percent confidence interval:
##  282.6440 364.5226
##
##
## $horsebean
##
##
##
## data:  X[[i]]
##
## 95 percent confidence interval:
##  132.5687 187.8313
```

```
##
##
## $linseed
##
##
##
## data:  X[[i]]
##
## 95 percent confidence interval:
##  185.561 251.939
##
##
## $meatmeal
##
##
##
## data:  X[[i]]
##
## 95 percent confidence interval:
##  233.3083 320.5099
##
##
## $soybean
##
##
##
## data:  X[[i]]
##
## 95 percent confidence interval:
##  215.1754 277.6818
##
##
## $sunflower
##
##
##
## data:  X[[i]]
##
## 95 percent confidence interval:
##  297.8875 359.9458
```

Part b.

To conclude that a feed is different, I want to check that the confidence intervals do not overlap. There are many cases of this, for example, sunflower is different from soybean, linseed, and horsebean. If the confidence intervals overlap, then there is no conclusion as it is possible they are different. This pseudo test is conservative.

Part c.

Let's load a couple of packages that will help check some assumptions.

```
library(fastR)
library(Hmisc)
```

First a summary of the data

```r
summary(weight~feed,chickwts)
```

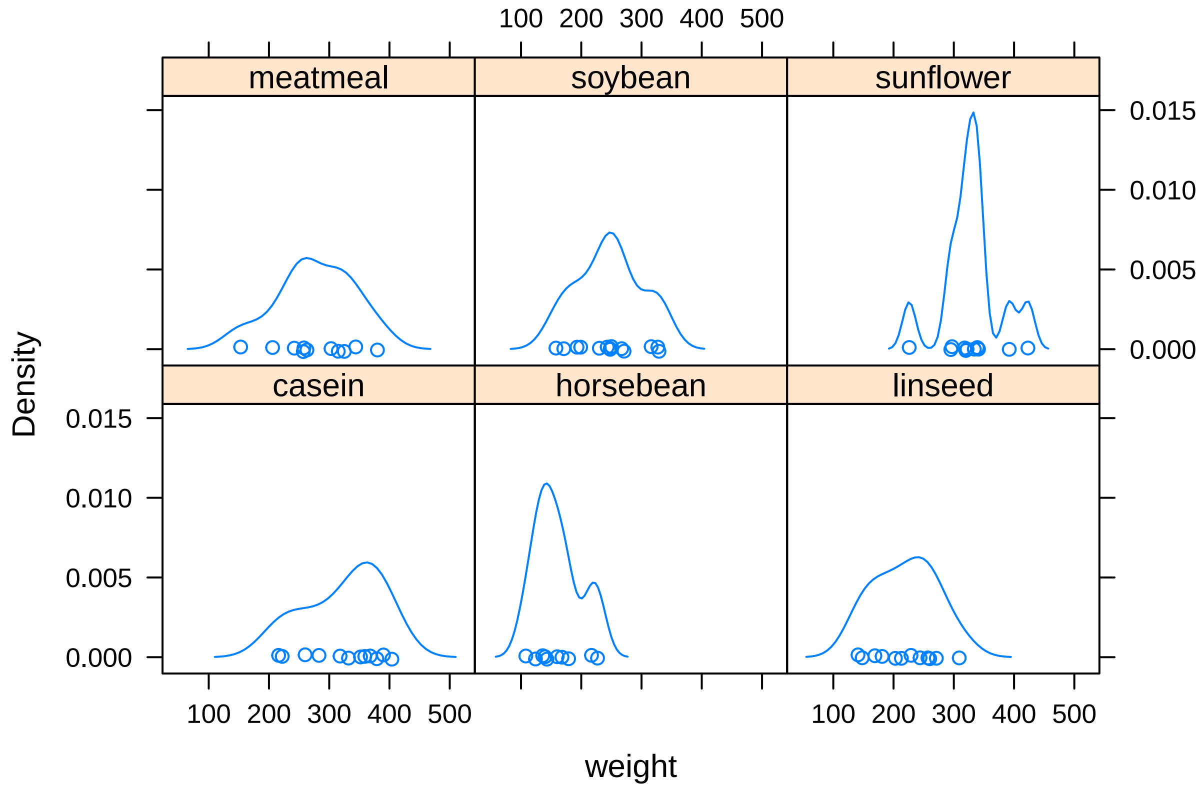
```
## weight    N=71
##
## +-------+---------+--+--------+
## |       |         |N |weight  |
## +-------+---------+--+--------+
## |feed   |casein   |12|323.5833|
## |       |horsebean|10|160.2000|
## |       |linseed  |12|218.7500|
## |       |meatmeal |11|276.9091|
## |       |soybean  |14|246.4286|
## |       |sunflower|12|328.9167|
## +-------+---------+--+--------+
## |Overall|         |71|261.3099|
## +-------+---------+--+--------+
```

```r
favstats(weight~feed,data=chickwts)
```

```
##        feed min     Q1 median     Q3 max     mean       sd  n missing
## 1    casein 216 277.25  342.0 370.75 404 323.5833 64.43384 12       0
## 2 horsebean 108 137.00  151.5 176.25 227 160.2000 38.62584 10       0
## 3   linseed 141 178.00  221.0 257.75 309 218.7500 52.23570 12       0
## 4  meatmeal 153 249.50  263.0 320.00 380 276.9091 64.90062 11       0
## 5   soybean 158 206.75  248.0 270.00 329 246.4286 54.12907 14       0
## 6 sunflower 226 312.75  328.0 340.25 423 328.9167 48.83638 12       0
```

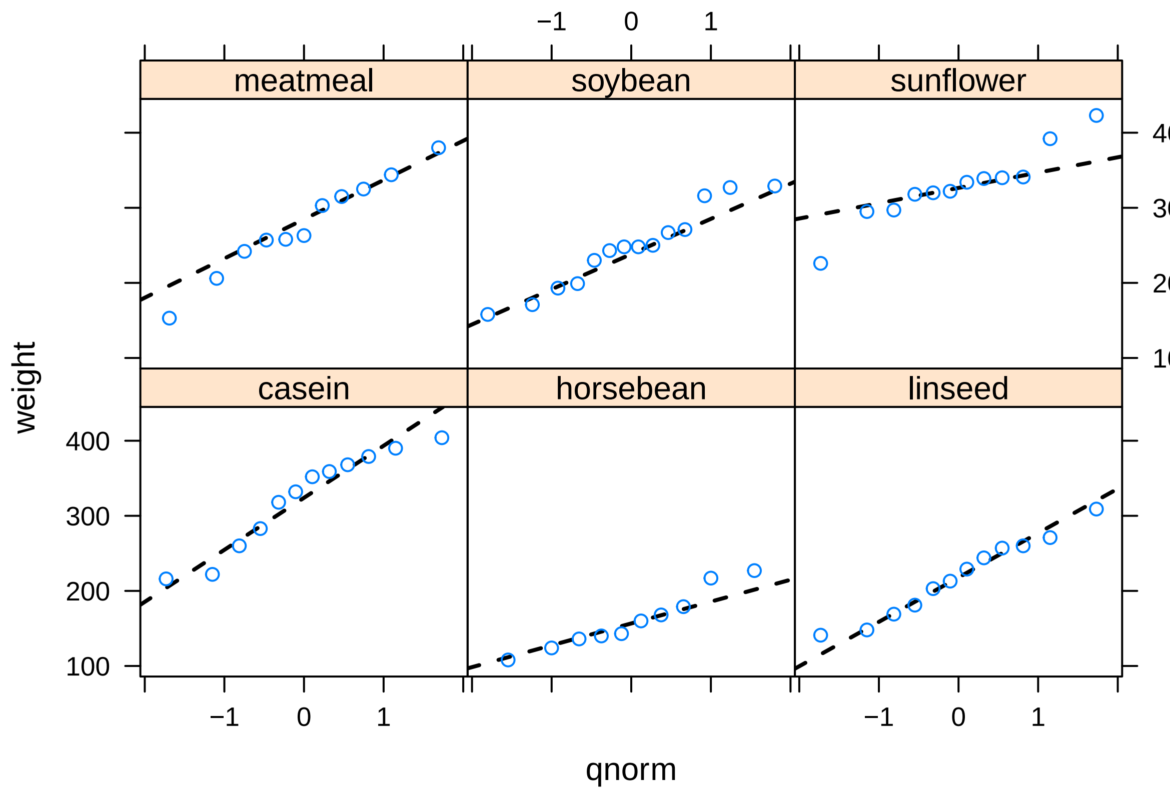
To use the t test we are assume the data is independent and identically distributed from a normal population. For this problem the means may vary from feed to feed but the distribution should still be norm. Let's visually inspect the normal assumption for each feed type. The independence is hard to check without knowing how the data was collected.

```r
densityplot(~weight|feed,data=chickwts)
```



The data is small so density plots may not be the best. Next let's use a quantile-quantile plot

```
xqqmath(~weight|feed,data=chickwts)
```



Not bad, there might be some slight skewing in the horsebeen and meatmeal feed and sunflower has longer tails.

Section 4.8 and 4.9

This is my homework for section 4.8 and 4.9 of the book.

Problem 4.50 First I will load the libraries that I think I will need

```
library(fastR)
library(Hmisc)
```

The data collection description contains some new terms, please see the wiki page on crossover designs. Randomization is used to help with the independence assumption. The double blind condition means that the patients and researcher do not know if the patient is receiving a placebo or treatment. Here is an interesting article from the New York Times on the importance of placebos.

Part a

Conduct a paired t-test to compare the two treatments.

First I will look at the data.

```
head(endurance)
```

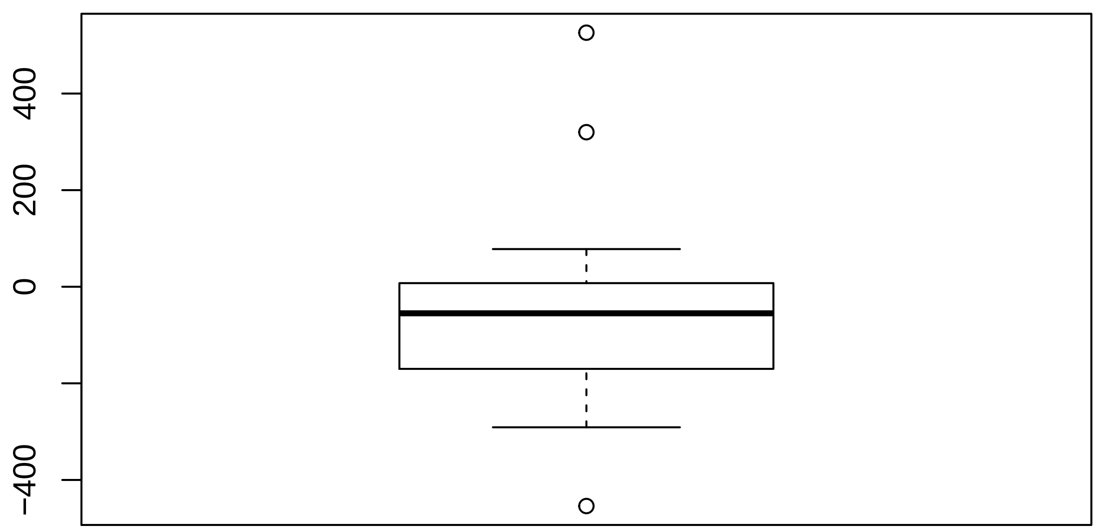
```
##   Vitamin   First Placebo
## 1     145 Vitamin     417
## 2     185 Placebo     279
## 3     387 Vitamin     678
## 4     593 Placebo     636
## 5     248 Vitamin     170
## 6     245 Placebo     699
```

```r
str(endurance)
```

```
## 'data.frame':    15 obs. of  3 variables:
##  $ Vitamin: int  145 185 387 593 248 245 349 902 159 122 ...
##  $ First  : Factor w/ 2 levels "Placebo","Vitamin": 2 1 2 1 2 1 2 1 2 1 ...
##  $ Placebo: int  417 279 678 636 170 699 372 582 363 258 ...
```

and a visual display

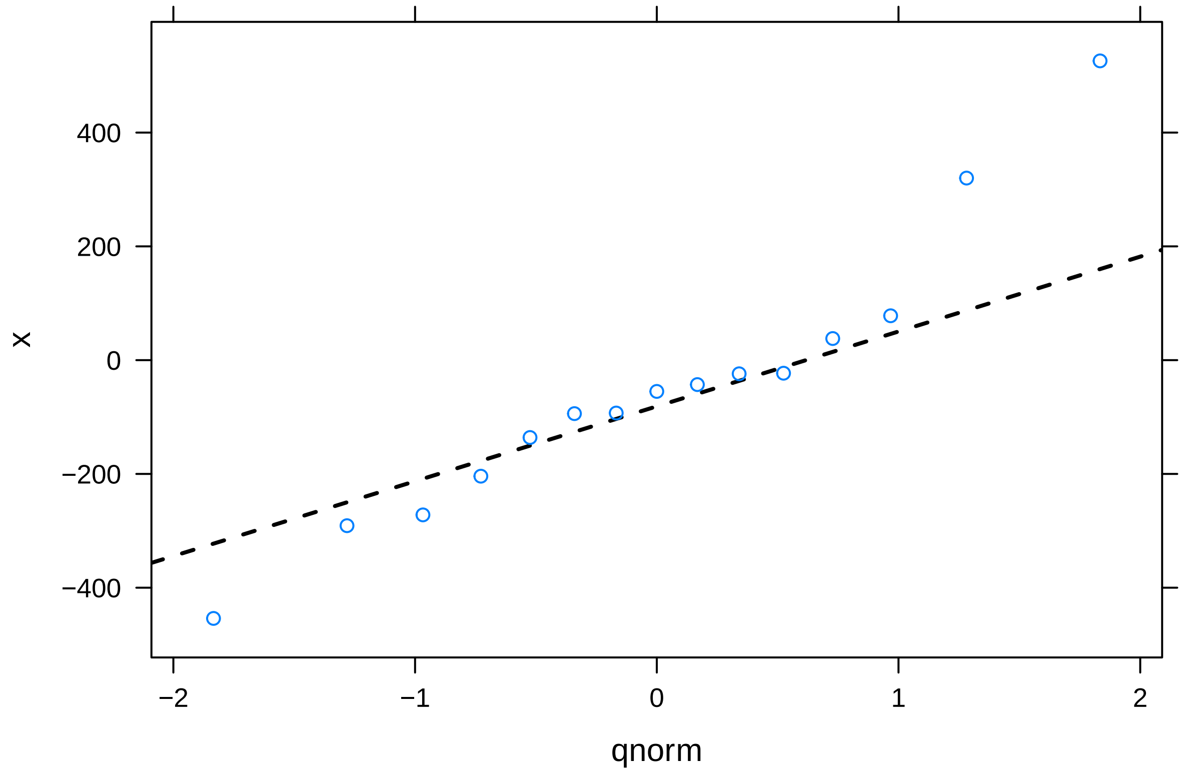
```r
boxplot(endurance$Vitamin-endurance$Placebo)
```



and perhaps a better plot

```r
xqqmath(endurance$Vitamin-endurance$Placebo)
```

```
## Warning in qqmath.numeric(x, data = data, panel = panel, ...): explicit
## 'data' specification ignored
```



These are visual plots of the difference since each row is an individual and each individual receives both treatments, placebo and vitamin. The data may have some unusually large values and some skewness.

Now I will conduct the hypothesis test where the null hypothesis is that the difference in the mean treatment effect is zero.

```
t.test(endurance$Vitamin,endurance$Placebo,paired=T)
```

```
##
##  Paired t-test
##
## data:  endurance$Vitamin and endurance$Placebo
## t = -0.78538, df = 14, p-value = 0.4453
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -180.82308   83.88975
## sample estimates:
## mean of the differences
##               -48.46667
```

or equivalently

```
t.test(endurance$Vitamin-endurance$Placebo)
```

```
##
```

```
##  One Sample t-test
##
## data:  endurance$Vitamin - endurance$Placebo
## t = -0.78538, df = 14, p-value = 0.4453
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -180.82308   83.88975
## sample estimates:
## mean of x
## -48.46667
```

Since the p-value is greater than 0.05, we fail to reject that the difference in mean strength between the vitamin and placebo is zero.

I am concerned about the assumption that the distribution of the difference is normal. From the boxplot above, this is somewhat questionable.

Part b.

This section has us using a transform of the original data. I think the hope is to address the skewness issue that is apparent in the original data.

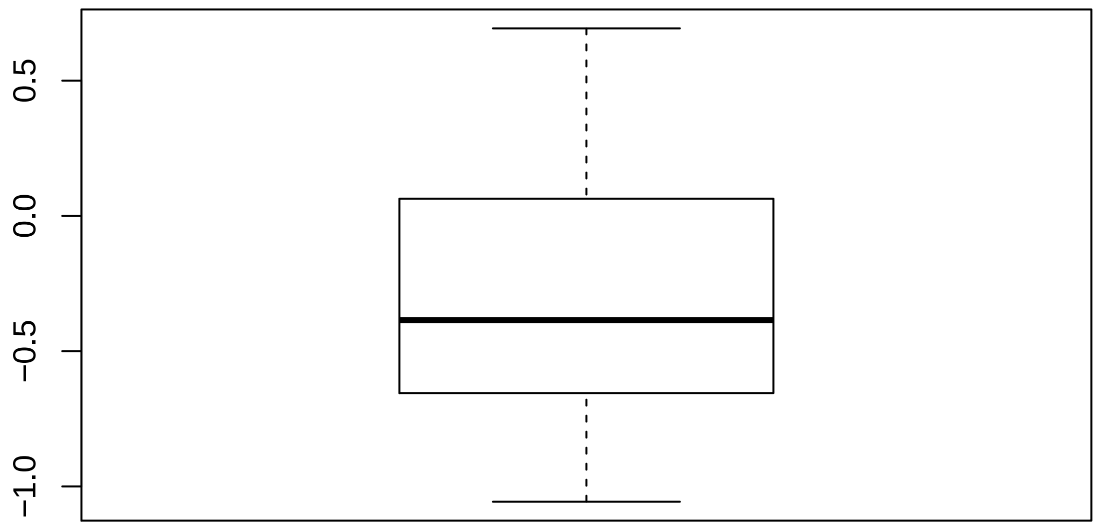
First, I will transform the data

```
endurancelog<-transform(endurance,LogDiff=log(Vitamin)-log(Placebo))
head(endurancelog)
```

```
##   Vitamin   First Placebo     LogDiff
## 1     145 Vitamin     417 -1.05635248
## 2     185 Placebo     279 -0.41085596
## 3     387 Vitamin     678 -0.56072259
## 4     593 Placebo     636 -0.07000416
## 5     248 Vitamin     170  0.37763031
## 6     245 Placebo     699 -1.04839253
```

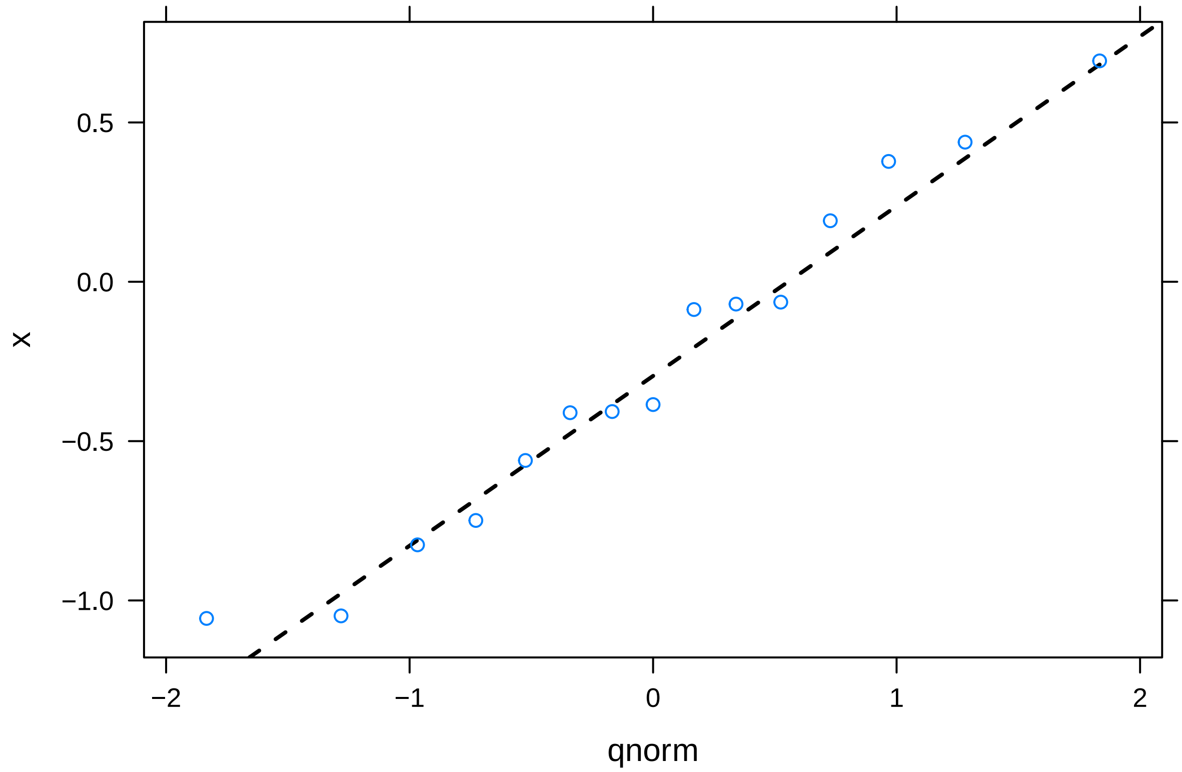
Next, I will plot and summarize the transformed data.

```
boxplot(endurancelog$LogDiff)
```




```r
xqqmath(endurancelog$LogDiff)
```

```
## Warning in qqmath.numeric(x, data = data, panel = panel, ...): explicit
## 'data' specification ignored
```




```r
summary(endurancelog)
```

```
##     Vitamin           First       Placebo          LogDiff
##  Min.   : 117.0   Placebo:7   Min.   :170.0   Min.   :-1.05635
##  1st Qu.: 172.0   Vitamin:8   1st Qu.:268.0   1st Qu.:-0.65483
##  Median : 245.0               Median :363.0   Median :-0.38532
##  Mean   : 344.7               Mean   :393.2   Mean   :-0.26425
##  3rd Qu.: 368.0               3rd Qu.:554.0   3rd Qu.: 0.06386
##  Max.   :1052.0               Max.   :699.0   Max.   : 0.69315
```

This transformation seems to help with the assumptions of the t-test.

Now for the t-test

```r
t.test(endurancelog$LogDiff)
```

```
##
##  One Sample t-test
##
## data:  endurancelog$LogDiff
## t = -1.8968, df = 14, p-value = 0.07868
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.56304724  0.03455067
## sample estimates:
```

```
##  mean of x
## -0.2642483
```

or if I did not want to create a transformed data set

```
t.test(log(endurance$Vitamin),log(endurance$Placebo),paired=T)
```

```
##
##  Paired t-test
##
## data:  log(endurance$Vitamin) and log(endurance$Placebo)
## t = -1.8968, df = 14, p-value = 0.07868
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.56304724  0.03455067
## sample estimates:
## mean of the differences
##              -0.2642483
```

Again, since the p-value is greater than 0.05, we fail to reject that the difference in mean strength between the vitamin and placebo is zero.

Part c.

The author is having us experiment with different transformations. If we take the log of the quotient we should get the same answer as part b.

```
t.test(log(endurance$Vitamin/endurance$Placebo))
```

```
##
##  One Sample t-test
##
## data:  log(endurance$Vitamin/endurance$Placebo)
## t = -1.8968, df = 14, p-value = 0.07868
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.56304724  0.03455067
## sample estimates:
##  mean of x
## -0.2642483
```

But perhaps we could just look at the quotient by itself but in this case the null hypothesis would have the quotient at 1

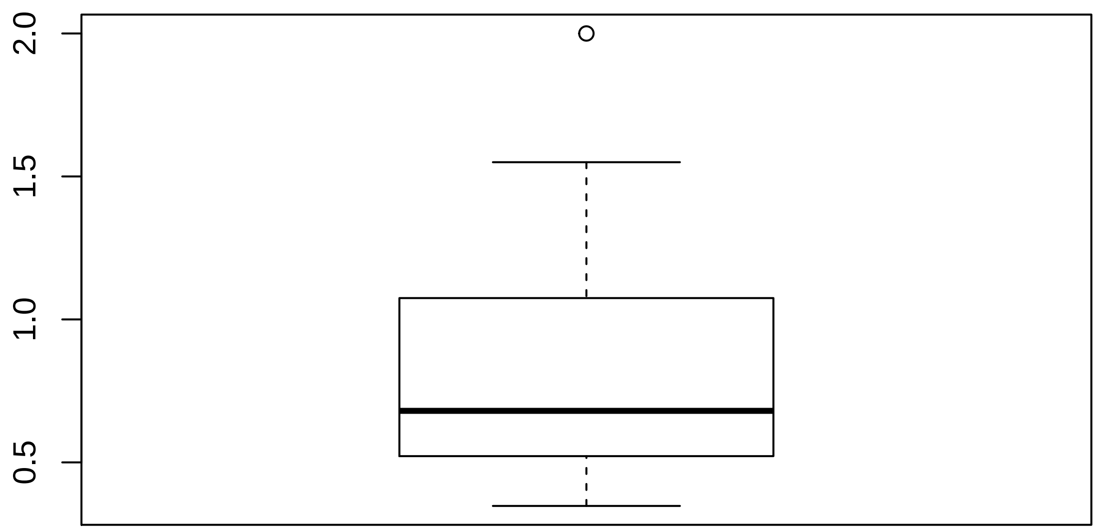
```
t.test(endurance$Vitamin/endurance$Placebo,mu=1)
```

```
##
##  One Sample t-test
##
## data:  endurance$Vitamin/endurance$Placebo
## t = -0.95832, df = 14, p-value = 0.3542
## alternative hypothesis: true mean is not equal to 1
```

```
## 95 percent confidence interval:
##  0.6104985 1.1489252
## sample estimates:
## mean of x
## 0.8797119
```

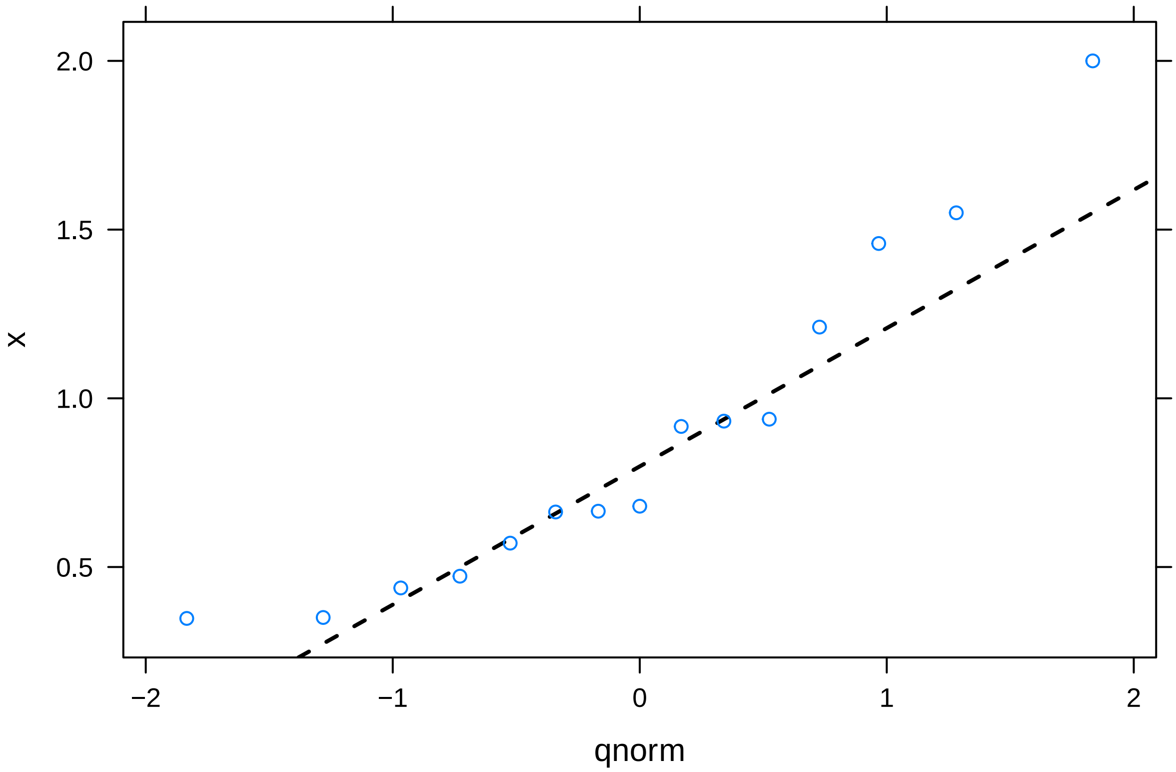
This did not help

```r
boxplot(endurance$Vitamin/endurance$Placebo)
```




```r
xqqmath(endurance$Vitamin/endurance$Placebo)
```

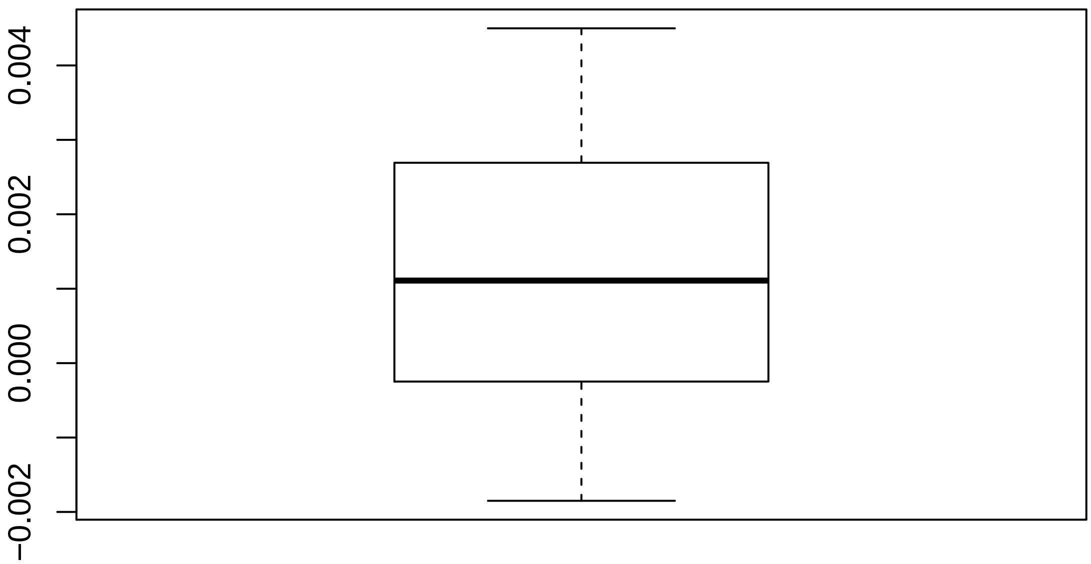
```
## Warning in qqmath.numeric(x, data = data, panel = panel, ...): explicit
## 'data' specification ignored
```



Part d.

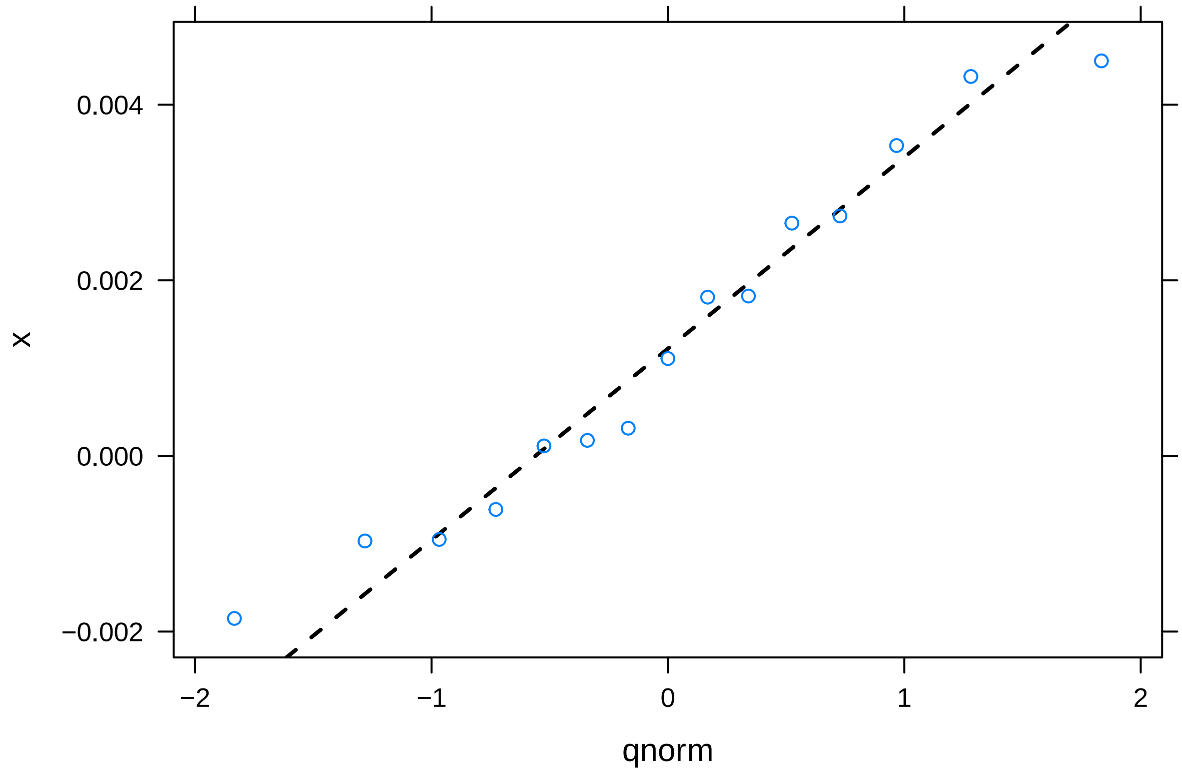
Another transformation attempt

```
boxplot(1/endurance$Vitamin-1/endurance$Placebo)
```




```r
xqqmath(1/endurance$Vitamin-1/endurance$Placebo)
```

```
## Warning in qqmath.numeric(x, data = data, panel = panel, ...): explicit
## 'data' specification ignored
```



This seems to help with the assumptions

```
summary(1/endurance$Vitamin-1/endurance$Placebo)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0018500 -0.0002478  0.0011090  0.0012470  0.0026920  0.0044980
```

And the t-test

```
t.test(1/endurance$Vitamin,1/endurance$Placebo,paired=T)
```

```
##
##  Paired t-test
##
## data:  1/endurance$Vitamin and 1/endurance$Placebo
## t = 2.4111, df = 14, p-value = 0.03022
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.0001377454 0.0023561868
## sample estimates:
## mean of the differences
##             0.001246966
```

or

```r
t.test(1/endurance$Vitamin-1/endurance$Placebo)
```

```
##
##  One Sample t-test
##
## data:  1/endurance$Vitamin - 1/endurance$Placebo
## t = 2.4111, df = 14, p-value = 0.03022
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.0001377454 0.0023561868
## sample estimates:
##   mean of x
## 0.001246966
```

In this case we would reject the null hypothesis of no difference in the reciprocal means. This would be a difficult transformation to explain to a decision maker.

Part e.

To run the sign test, I must generate a binomial variable by comparing vitamin to placebo and then testing with an exact binomial test.

```r
binom.test(sum(endurance$Vitamin>endurance$Placebo),length(endurance$Vitamin))
```

```
##
##
##
## data:  sum(endurance$Vitamin > endurance$Placebo) out of length(endurance$Vitamin)
## number of successes = 4, number of trials = 15, p-value = 0.1185
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.07787155 0.55100324
## sample estimates:
## probability of success
##              0.2666667
```

This test assumes that the differences are independent and identically distributed. These are hard to verify and rely on the test design.

Since the p-value is greater than 0.05, we fail to reject that the difference in mean strength between the vitamin and placebo is zero.

Even though it is easy to use `binom.test` for this problem, I could write my own sign test function

```r
#My sign test
sign.test=function(x, y = NULL, md = 0,
  alternative =  c("two.sided", "less", "greater"), conf.level = 0.95){
  if(is.null(y)) y=rep(md,length(x))
  if(sum(which(x==y))!=0){
    xx=x
    yy=y
    x=xx[-1*which(xx==yy)]
    y=yy[-1*which(xx==yy)]
  }
```

```r
  ans=binom.test(sum(x>y),length(x),alternative=alternative,conf.level=conf.level)
  ans$method="Sign Test"
  return(ans)
}
```

And then test

```r
sign.test(endurance$Vitamin,endurance$Placebo)
```

```
##
##  Sign Test
##
## data:  sum(x > y) out of length(x)
## number of successes = 4, number of trials = 15, p-value = 0.1185
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.07787155 0.55100324
## sample estimates:
## probability of success
##              0.2666667
```

The advantage of my code is that I can handle ties better than using `binom.test`.

As a side note, if I am interested in looking at vitamin and placebo separately, then the stack command works well.

```r
stack(endurance[,c(1,3)])
```

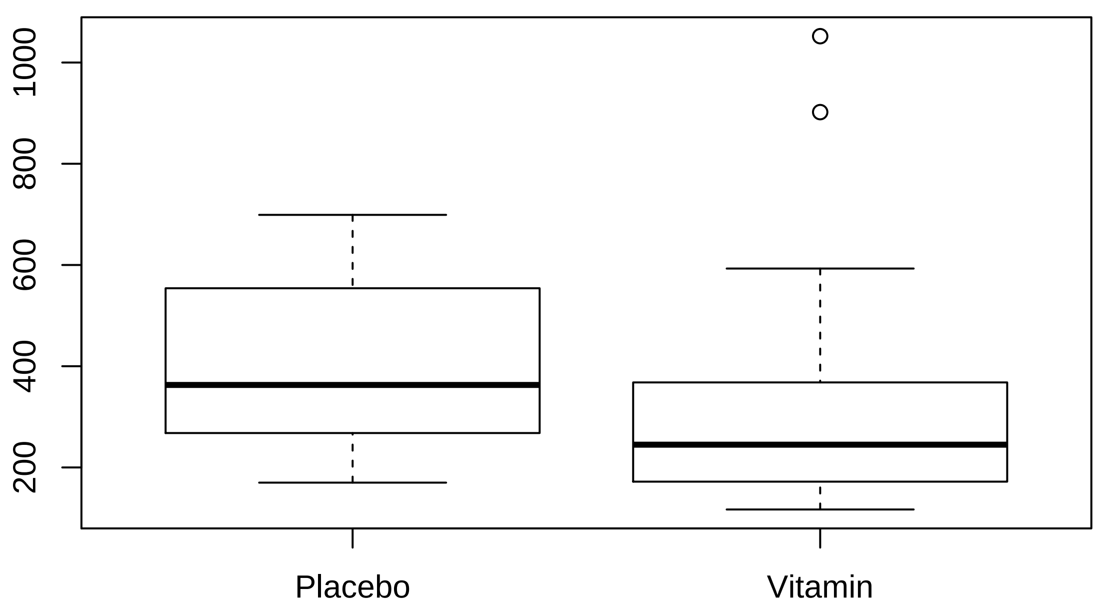
```
##    values     ind
## 1     145 Vitamin
## 2     185 Vitamin
## 3     387 Vitamin
## 4     593 Vitamin
## 5     248 Vitamin
## 6     245 Vitamin
## 7     349 Vitamin
## 8     902 Vitamin
## 9     159 Vitamin
## 10    122 Vitamin
## 11    264 Vitamin
## 12   1052 Vitamin
## 13    218 Vitamin
## 14    117 Vitamin
## 15    185 Vitamin
## 16    417 Placebo
## 17    279 Placebo
## 18    678 Placebo
## 19    636 Placebo
## 20    170 Placebo
## 21    699 Placebo
## 22    372 Placebo
## 23    582 Placebo
```

```
## 24    363 Placebo
## 25    258 Placebo
## 26    288 Placebo
## 27    526 Placebo
## 28    180 Placebo
## 29    172 Placebo
## 30    278 Placebo
```

```
summary(values~ind,data=stack(endurance[,c(1,3)]),fun=favstats)
```

```
## values    N=30
##
## +-------+-------+--+----+---+-------+------+----+--------+--------+--+--------+
## |       |       |N |min |Q1 |median |Q3    |max |mean    |sd      |n |missing |
## +-------+-------+--+----+---+-------+------+----+--------+--------+--+--------+
## |ind    |Placebo|15|170 |268|363.0  |554.00| 699|393.2000|186.8243|15|0       |
## |       |Vitamin|15|117 |172|245.0  |368.00|1052|344.7333|285.7308|15|0       |
## +-------+-------+--+----+---+-------+------+----+--------+--------+--+--------+
## |Overall|       |30|117 |185|278.5  |498.75|1052|368.9667|238.4760|30|0       |
## +-------+-------+--+----+---+-------+------+----+--------+--------+--+--------+
```

```
boxplot(values~ind,data=stack(endurance[,c(1,3)]))
```



Part f.

To determine which is the best analysis, you have to determine how the models satisfy the assumptions and if the test answers the original research question. The analysis in part a is a little suspect in meeting the assumptions. The analysis in parts b and d are better at meeting the assumptions, but make it difficult to answer the original research question. The sign test assumes the least but thus is more conservative. I would select the sign test as the best choice for this problem.

Problem 4.51 Joe flips a coin 200 times and records 115 heads.

Part a.

Give a 95% confidence interval for the true proportion of heads.

Enter the data

```
x=115
n=200
```

From a Wald

```
x/n+c(-1,1)*qnorm(.975)*sqrt(x/n*(1-x/n)/n)
```

```
## [1] 0.5064888 0.6435112
```

Score

```
(x/n +qnorm(.975)^2/(2*n)+
      c(-1,1)*qnorm(.975)*
      sqrt(x/n*(1-x/n)/n+(qnorm(.975)/(2*n))^2))/
      (1+qnorm(.975)^2/(n))
```

```
## [1] 0.5057093 0.6414639
```

Wilson, which is the Wald with 2 added successes and failures

```
(x+2)/(n+4)+c(-1,1)*qnorm(.975)*sqrt((x+2)/(n+4)*(1-(x+2)/(n+4))/(n+4))
```

```
## [1] 0.5056629 0.6413959
```

Part b.

Running the code

```
prop.test(115,200,correct=F)
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  115 out of 200
## X-squared = 4.5, df = 1, p-value = 0.03389
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.5057093 0.6414639
## sample estimates:
##     p
## 0.575
```

```r
prop.test(115,200,correct=T)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  115 out of 200
## X-squared = 4.205, df = 1, p-value = 0.0403
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.5032062 0.6438648
## sample estimates:
##     p
## 0.575
```

It appears that `prop.test` without the continuity correction is the score confidence interval. With the continuity correction it is none of the three methods.

The exact confidence interval from `binom.test` is

```r
binom.test(115,200)
```

```
##
##
##
## data:  115 out of 200
## number of successes = 115, number of trials = 200, p-value =
## 0.04004
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5033041 0.6444388
## sample estimates:
## probability of success
##                  0.575
```

This is the Clopper-Pearson confidence interval.

Note fastR has wald.ci and wilson.ci built in but as of 30 Oct 14, the Wilson confidence interval has an error.

```r
wald.ci(115,200)
```

```
## [1] 0.5064888 0.6435112
## attr(,"conf.level")
## [1] 0.95
```

```r
wilson.ci(115,200)
```

```
## [1] 0.5056629 0.6413959
## attr(,"conf.level")
## [1] 0.95
```

The error is to not divide by $n+4$ in the standard error.

```r
wilson.ci
```

```
## function (x, n = 100, conf.level = 0.95)
## {
##     alpha = 1 - conf.level
##     p = (x + 2)/(n + 4)
##     zstar <- -qnorm(alpha/2)
##     interval <- p + c(-1, 1) * zstar * sqrt(p * (1 - p)/(n +
##         4))
##     attr(interval, "conf.level") <- conf.level
##     return(interval)
## }
## <environment: namespace:fastR>
```

Problem 4.52 In this study, 45 coffee drinkers sampled fresh brewed coffee versus gourmet instant coffee with 14 preferring the instant, 26 the fresh brewed and 5 with no preference.

Part a.

If the sample size is small relative to the number of no responses then how we treat them could have a big impact. If we simply group them into one category, it would lead to a bias. Dropping them will reduce the sample size. Finally, if no response is related to one of the other responses instead of just being random, then we could have a bias in just dropping them.

Part b.

We could just drop them or randomly assign them to the two outcomes based on the empirical proportion in each category. In a more complex manner, we could try to develop a predictive model as to why there is no response and use this to evaluate the data. This would potentially require more variables to be measured.

The other consideration is how the participants were asked for their response. For example, if they were asked "Do you prefer the fresh brewed coffee?", then a no preference means no they do not prefer it. They would get grouped with the gourmet instant coffee as not preferring fresh brewed coffee.

Part c.

Again, it is important to understand the hypothesis. If the hypothesis is which do you prefer fresh or gourmet coffee, then I would drop them or assign them randomly to each group.

If the question is do you prefer fresh brewed coffee, then I would group them with the gourmet coffee drinkers.

Part d.

I have to set up the hypothesis. Let π be the proportion of drinkers that favor fresh brewed coffee. The hypothesis test is

$$H_o : \pi = 0.5$$

$$H_a : \pi > 0.5$$

First, I will see what happens if I drop the no preference responses.

Using the sign test implemented with `binom.test`

```r
binom.test(26,40,alt="greater")
```

```
##
##
##
## data:  26 out of 40
## number of successes = 26, number of trials = 40, p-value = 0.04035
## alternative hypothesis: true probability of success is greater than 0.5
## 95 percent confidence interval:
##  0.5080545 1.0000000
## sample estimates:
## probability of success
##                   0.65
```

Based on the data and a level of significance of 0.05, if there was no preference for fresh brewed coffee, then the probability of getting 26 or more drinkers out of 40 who prefer fresh brewed coffee is 0.04. Thus I reject the hypothesis that there is no preference in favor of the hypothesis that drinkers prefer fresh brewed coffee. We can also see this from the lower confidence bound of 0.5081. This means we are 95% confident that the proportion of drinkers who prefer fresh brewed coffee is at least 0.5081.

Now the question appears to be, do you prefer fresh brewed coffee? Let's put the 5 no preference into the analysis.

Using the sign test implemented with `binom.test`

```
binom.test(26,45,alt="greater")
```

```
##
##
##
## data:  26 out of 45
## number of successes = 26, number of trials = 45, p-value = 0.1856
## alternative hypothesis: true probability of success is greater than 0.5
## 95 percent confidence interval:
##  0.4445067 1.0000000
## sample estimates:
## probability of success
##              0.5777778
```

No we fail to reject. Based on the data and a level of significance of 0.05, if there was no preference for fresh brewed coffee, then the probability of getting 26 or more drinkers out of 45 who prefer fresh brewed coffee is 0.1856. Thus I fail to reject the hypothesis that there is no preference in favor of the hypothesis that drinkers prefer fresh brewed coffee. We can also see this from the lower confidence bound of 0.4445. This means we are 95% confident that the proportion of drinkers who prefer fresh brewed coffee is at least 0.4445.

Finally, let's put 3 into the fresh brewed and 2 into the instant and see the results.

```
binom.test(29,45,alt="greater")
```

```
##
##
##
## data:  29 out of 45
## number of successes = 29, number of trials = 45, p-value = 0.03623
## alternative hypothesis: true probability of success is greater than 0.5
## 95 percent confidence interval:
```

```
## 0.5112937 1.0000000
## sample estimates:
## probability of success
##              0.6444444
```

The choice we make has a big impact on the results for this problem.

Section 4.10

This is my homework for section 4.10 of the book.

Let's load libraries to help with the homework.

```
library(fastR)
library(Hmisc)
```

Problem 4.53 Let's look at the data first

```
str(scent)
```

```
## 'data.frame':    21 obs. of  12 variables:
##  $ id     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ sex    : Factor w/ 2 levels "F","M": 2 1 2 2 2 1 1 1 2 1 ...
##  $ smoker : Factor w/ 2 levels "N","Y": 1 2 1 1 1 2 1 1 1 1 ...
##  $ opinion: Factor w/ 3 levels "indiff","neg",..: 3 2 3 2 2 3 3 3 3 1 ...
##  $ age    : int  23 43 43 32 15 37 26 35 26 31 ...
##  $ first  : Factor w/ 2 levels "scented","unscented": 2 1 2 1 2 1 2 1 2 1 ...
##  $ u1     : num  38.4 46.2 72.5 38 82.8 33.9 50.4 35 32.8 60.1 ...
##  $ u2     : num  27.7 57.2 57.9 38 57.9 32 40.6 33.1 26.8 53.2 ...
##  $ u3     : num  25.7 41.9 51.9 32.2 64.7 31.4 40.1 43.2 33.9 40.4 ...
##  $ s1     : num  53.1 54.7 74.2 49.6 53.6 51.3 44.1 34 34.5 59.1 ...
##  $ s2     : num  30.6 43.3 53.4 37.4 48.6 35.5 46.9 26.4 25.1 87.1 ...
##  $ s3     : num  30.2 56.7 42.4 34.4 44.8 42.9 42.7 24.8 25.1 59.2 ...
```

```
summary(scent)
```

```
##        id      sex    smoker   opinion        age              first
##  Min.   : 1   F:10   N:13   indiff: 4   Min.   :15.00   scented  :10
##  1st Qu.: 6   M:11   Y: 8   neg   : 7   1st Qu.:26.00   unscented:11
##  Median :11                 pos   :10   Median :35.00
##  Mean   :11                             Mean   :36.57
##  3rd Qu.:16                             3rd Qu.:43.00
##  Max.   :21                             Max.   :65.00
##        u1              u2              u3              s1
##  Min.   :32.80   Min.   :26.80   Min.   :25.70   Min.   :28.30
##  1st Qu.:40.90   1st Qu.:38.00   1st Qu.:37.20   1st Qu.:49.60
##  Median :49.50   Median :46.80   Median :43.20   Median :53.60
##  Mean   :53.92   Mean   :48.94   Mean   :47.18   Mean   :55.53
##  3rd Qu.:60.10   3rd Qu.:57.90   3rd Qu.:58.00   3rd Qu.:67.30
##  Max.   :93.80   Max.   :91.90   Max.   :77.40   Max.   :77.50
##        s2              s3
```

```
##  Min.   : 25.10   Min.   :24.80
##  1st Qu.: 36.80   1st Qu.:34.40
##  Median : 44.00   Median :42.90
##  Mean   : 48.31   Mean   :43.32
##  3rd Qu.: 53.40   3rd Qu.:48.40
##  Max.   :126.60   Max.   :64.50
```

```
head(scent)
```

```
##   id sex smoker opinion age     first   u1   u2   u3   s1   s2   s3
## 1  1   M      N     pos  23 unscented 38.4 27.7 25.7 53.1 30.6 30.2
## 2  2   F      Y     neg  43   scented 46.2 57.2 41.9 54.7 43.3 56.7
## 3  3   M      N     pos  43 unscented 72.5 57.9 51.9 74.2 53.4 42.4
## 4  4   M      N     neg  32   scented 38.0 38.0 32.2 49.6 37.4 34.4
## 5  5   M      N     neg  15 unscented 82.8 57.9 64.7 53.6 48.6 44.8
## 6  6   F      Y     pos  37   scented 33.9 32.0 31.4 51.3 35.5 42.9
```

Part a.

There are many questions a researcher could ask with this data set
1. Does smoking impact the results?
2. Does the order, unscented versus scented, impact the results?
3. Is there a difference in performance between males and females?
4. Is there a learning curve, does the time on the third trial decrease relative to the first?
5. Are scented times less than unscented times?

Part b.

If we are talking about mean performance, then we could answer the scented versus unscented using a paired t-test. We don't know the two sample t-test, but we could use it on the other questions. We could use a permutation test on any of these questions.

Part c.

I will do a couple of test in R using a simulation to find empirical p-values.

First, I will test if men are different from women. For this I will use the mean performance as a metric. For the test I will use the first trial of unscented.

The hypothesis is

$$H_0 : \mu_{male} = \mu_{female}$$

$$H_0 : \mu_{male} \neq \mu_{female}$$

Where μ is the mean performance on the first maze with the unscented mask and I am assuming they are independent and identically distributed with the exception that the mean may be different.

Next ,I will get the data I need in a smaller data frame

```
(prob4.53a=scent[,c(1,2,7)])
```

```
##    id sex   u1
## 1   1   M 38.4
## 2   2   F 46.2
## 3   3   M 72.5
## 4   4   M 38.0
## 5   5   M 82.8
## 6   6   F 33.9
```

```
## 7   7   F 50.4
## 8   8   F 35.0
## 9   9   M 32.8
## 10 10   F 60.1
## 11 11   F 75.1
## 12 12   F 57.6
## 13 13   F 55.5
## 14 14   M 49.5
## 15 15   M 40.9
## 16 16   M 44.3
## 17 17   M 93.8
## 18 18   M 47.9
## 19 19   F 75.2
## 20 20   F 46.2
## 21 21   M 56.3
```

```r
names(prob4.53a)[3]="Time"
```

Now I need a test statistic, I will use the absolute value of the differences in means. From the data this value is

```r
(teststatperm=abs(diff(tapply(prob4.53a$Time,prob4.53a$sex,mean))))
```

```
##         M
## 0.7709091
```

Now under the null hypothesis, if the two means were the same then the labels of male and female could be changed and it would not matter. Thus let's change the male and female labels and test how rare our observed data is.

I will experiment with some code to get what I need

```r
with(prob4.53a,abs(diff(tapply(Time,sex,mean))))
```

```
##         M
## 0.7709091
```

```r
prob4.53a$sex
```

```
##  [1] M F M M M F F F M F F F F M M M M M F F M
## Levels: F M
```

```r
sample(prob4.53a$sex)
```

```
##  [1] F F M M F F F F M M M M F F M F M M F M M
## Levels: F M
```

```r
with(prob4.53a,abs(diff(tapply(Time,sample(sex),mean))))
```

```
##        M
## 14.02455
```

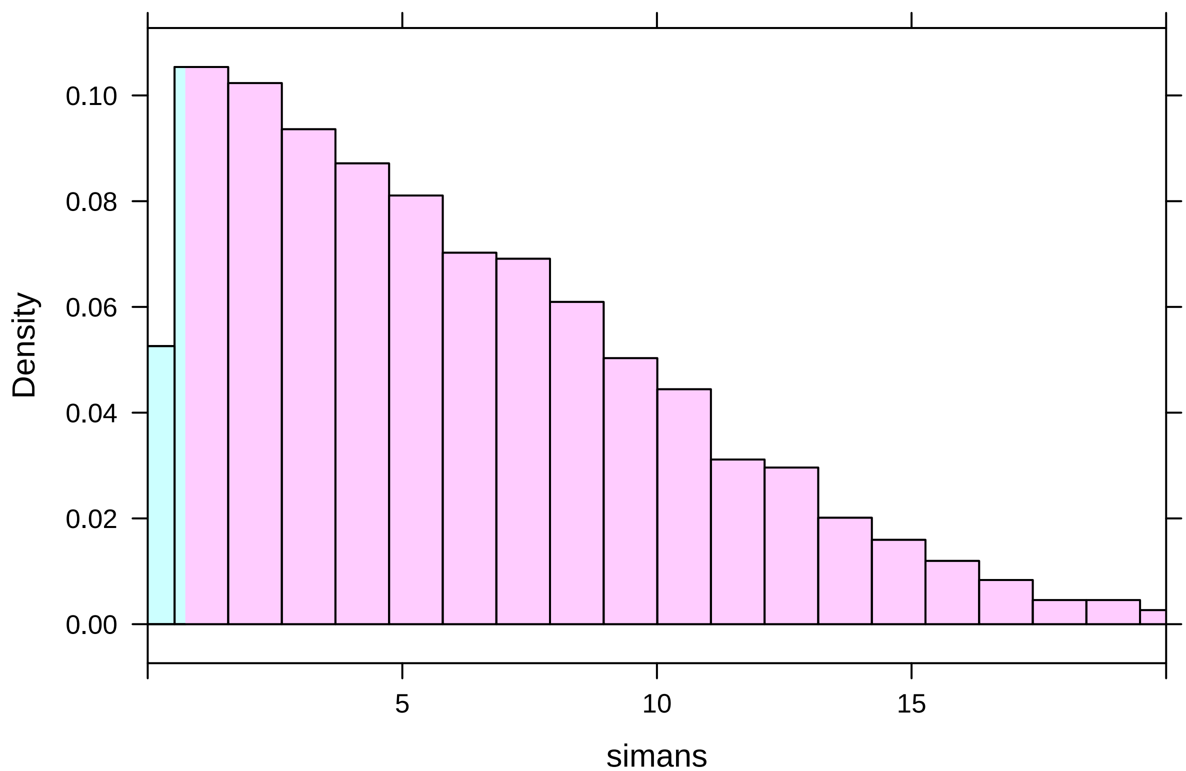
The `with` command allows me to change the order of gender in the data frame. Now I will simulate the data under the assumptions of the null hypothesis.

```
numsims=5000
simans=with(prob4.53a,replicate(numsims,abs(diff(tapply(Time,sample(sex),mean)))))
sum(simans>=teststatperm)/numsims
```

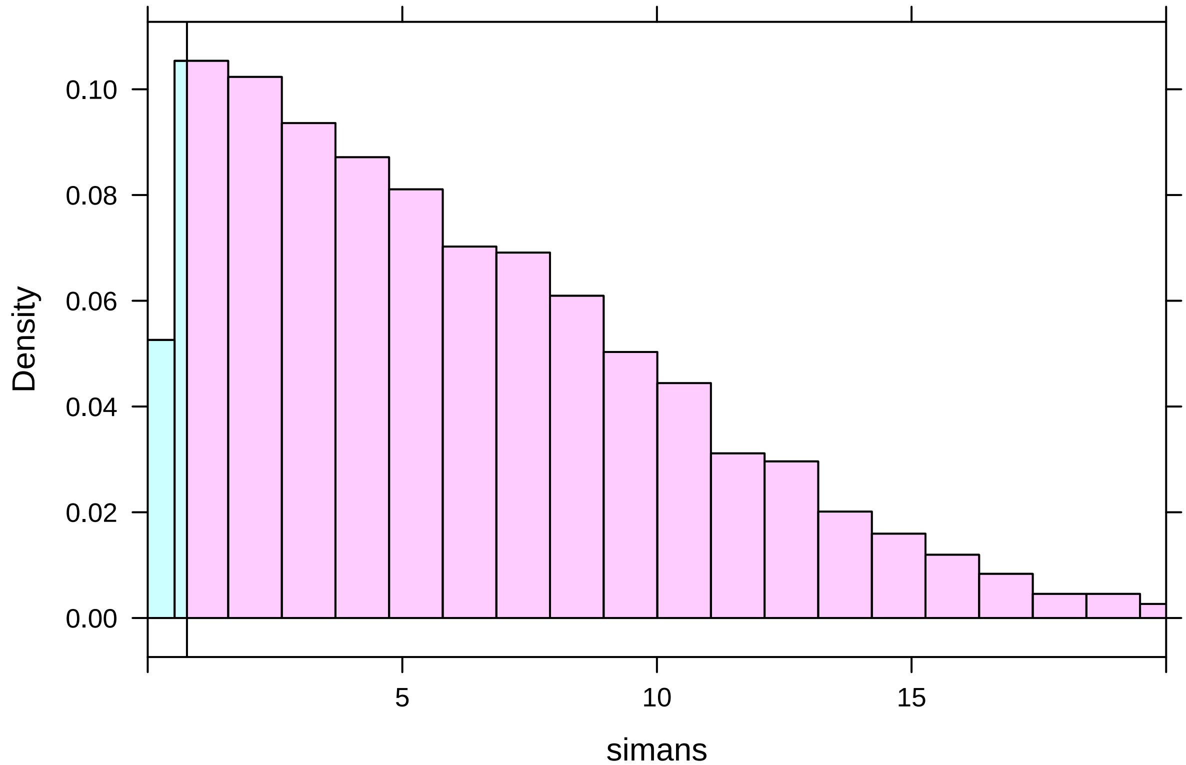
```
## [1] 0.9202
```

Next I will plot the results

```
histogram(simans,n=25, xlim=c(0, 20),
          groups=simans >= teststatperm, pch=16, cex=.8)
```




```
ladd(panel.abline(v=teststatperm))
```



If the mean time to complete the first maze with an unscented mask were the same for men and women, then the probability of getting a difference of .77 or more extreme, is 0.92, thus I fail to reject.

I could also test this in R with a two sample t-test, we did not learn about this.

```
t.test(Time~sex,data=prob4.53a,var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  Time by sex
## t = -0.10032, df = 19, p-value = 0.9211
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.85397  15.31216
## sample estimates:
## mean in group F mean in group M
##        53.52000        54.29091
```

Note that the p-value is about the same.

Next, I will test whether the scented time is different from the unscented. I will use a paired t-test. Also I will use the times from the third trial.

```
prob4.53b=scent[,c(1,9,12)]
names(prob4.53b)[c(2,3)]=c("Un","Scent")
prob4.53b
```

```
##    id   Un Scent
## 1   1 25.7  30.2
## 2   2 41.9  56.7
## 3   3 51.9  42.4
## 4   4 32.2  34.4
## 5   5 64.7  44.8
## 6   6 31.4  42.9
## 7   7 40.1  42.7
## 8   8 43.2  24.8
## 9   9 33.9  25.1
## 10 10 40.4  59.2
## 11 11 58.0  42.2
## 12 12 61.5  48.4
## 13 13 44.6  32.0
## 14 14 35.3  48.1
## 15 15 37.2  33.7
## 16 16 39.4  42.6
## 17 17 77.4  54.9
## 18 18 52.8  64.5
## 19 19 63.6  43.1
## 20 20 56.6  52.8
## 21 21 58.9  44.3
```

And now the paired t-test

```r
t.test(prob4.53b$Un,prob4.53b$Scent,paired=T)
```

```
##
##  Paired t-test
##
## data:  prob4.53b$Un and prob4.53b$Scent
## t = 1.3571, df = 20, p-value = 0.1899
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.069097  9.773859
## sample estimates:
## mean of the differences
##                3.852381
```

We could also use the sign test

```r
binom.test(sum(prob4.53b$Un>prob4.53b$Scent),length(prob4.53b$Un))
```

```
##
##
##
## data:  sum(prob4.53b$Un > prob4.53b$Scent) out of length(prob4.53b$Un)
## number of successes = 12, number of trials = 21, p-value = 0.6636
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.3402063 0.7818031
## sample estimates:
## probability of success
##              0.5714286
```

If I wanted to generate a permutation test for this, the code would be a little different. Here is my try

```
(time<-c(prob4.53b$Un,prob4.53b$Scent))
```

```
##  [1] 25.7 41.9 51.9 32.2 64.7 31.4 40.1 43.2 33.9 40.4 58.0 61.5 44.6 35.3
## [15] 37.2 39.4 77.4 52.8 63.6 56.6 58.9 30.2 56.7 42.4 34.4 44.8 42.9 42.7
## [29] 24.8 25.1 59.2 42.2 48.4 32.0 48.1 33.7 42.6 54.9 64.5 43.1 52.8 44.3
```

```
(teststatperm<-abs(mean(time[1:21]-time[22:42])))
```

```
## [1] 3.852381
```

```
#or
(abs(mean(diff((time),lag=21))))
```

```
## [1] 3.852381
```

```
numsims=5000
simans<-replicate(numsims,abs(mean(diff(sample(time),lag=21))))
sum(simans>=teststatperm)/numsims
```

```
## [1] 0.3022
```

This p-value is a little larger than the p-value from the paired t-test although the conclusion is the same. I don't think we did this correct. With paired data we have to keep the data together and we did not do this. We did a two-sample test and thus are picking up sample to sample variance. We need to sample the pair and then arbitarily change the label of scented and unscented. All that would happen is that we would change the sign of the difference.

Let's do the problem again but as a paired. Let's find the differences.

```
(prob4.53bdiff<-prob4.53b$Un-prob4.53b$Scent)
```

```
##  [1]  -4.5 -14.8   9.5  -2.2  19.9 -11.5  -2.6  18.4   8.8 -18.8  15.8
## [12]  13.1  12.6 -12.8   3.5  -3.2  22.5 -11.7  20.5   3.8  14.6
```

```
(teststatperm<-abs(mean(prob4.53b$Un-prob4.53b$Scent)))
```

```
## [1] 3.852381
```

Let's do one sample

```
set.seed(212)
abs(mean(sample(c(-1,1),21,replace=TRUE)*prob4.53bdiff))
```

```
## [1] 5.395238
```

I think we are ready

```
set.seed(212)
numsims=5000
simans<-replicate(numsims,abs(mean(sample(c(-1,1),21,replace=TRUE)*prob4.53bdiff)))
sum(simans>=teststatperm)/numsims
```

```
## [1] 0.1956
```

Problem 4.54 There is not `gossett` data set in the `fastR` package so here is the data needed

Regular Kiln
1 1903 2009
2 1935 1915
3 1910 2011
4 2496 2463
5 2108 2180
6 1961 1925
7 2060 2122
8 1444 1482
9 1612 1542
10 1316 1443
11 1511 1535

I am entering it in R using the following command

```
gosset=data.frame(Regular=c(1903,1935,1910,2496,2108,1961,2060,1444,1612,1316,1511),
        Kiln=c(2009,1915,2011,2463,2180,1925,2122,1482,1542,1443,1535))
```

Now, I will examine the data

```
str(gosset)
```

```
## 'data.frame':    11 obs. of  2 variables:
##  $ Regular: num  1903 1935 1910 2496 2108 ...
##  $ Kiln   : num  2009 1915 2011 2463 2180 ...
```

```
summary(gosset)
```

```
##     Regular          Kiln     
##  Min.   :1316   Min.   :1443  
##  1st Qu.:1562   1st Qu.:1538  
##  Median :1910   Median :1925  
##  Mean   :1841   Mean   :1875  
##  3rd Qu.:2010   3rd Qu.:2066  
##  Max.   :2496   Max.   :2463  
```

Part a.

Plotting the seeds in adjacent plots removes the location variation from the analysis.

Part b.

I will run a paired t-test on the data to determine if there is a statistically significant difference in seed preparation method.

```
t.test(gosset$Regular,gosset$Kiln,paired=T)
```

```
##
##  Paired t-test
##
## data:  gosset$Regular and gosset$Kiln
## t = -1.6905, df = 10, p-value = 0.1218
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -78.18164  10.72710
## sample estimates:
## mean of the differences
##               -33.72727
```

Based on the data with $\alpha = .05$, if there is no difference between yield of regular and kiln dried seeds, the probability of my data or more extreme is 12.18%. I fail to reject the hypothesis of no difference in yield of regular and kiln dried seeds.

Problem 4.56 I want to look at the variables used in Example 4.10.1.

```
golfballs
```

```
##   1   2   3   4
## 137 138 107 104
```

```
rgolfballs[1:4,1:10]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]  129  125  118  134  132  122  116  113  113   112
## [2,]  125  117  123  104  138  143  132  133  132   119
## [3,]  108  126  130  120  111  109  122  117  104   118
## [4,]  124  118  115  128  105  112  116  123  137   137
```

The data object `rgolfballs` has simulated data where each column is a simulation.

Next I want to repeat the analysis in the book

```
teststat=function(x){diff(range(x))}
statTally(golfballs,rgolfballs,teststat)
```

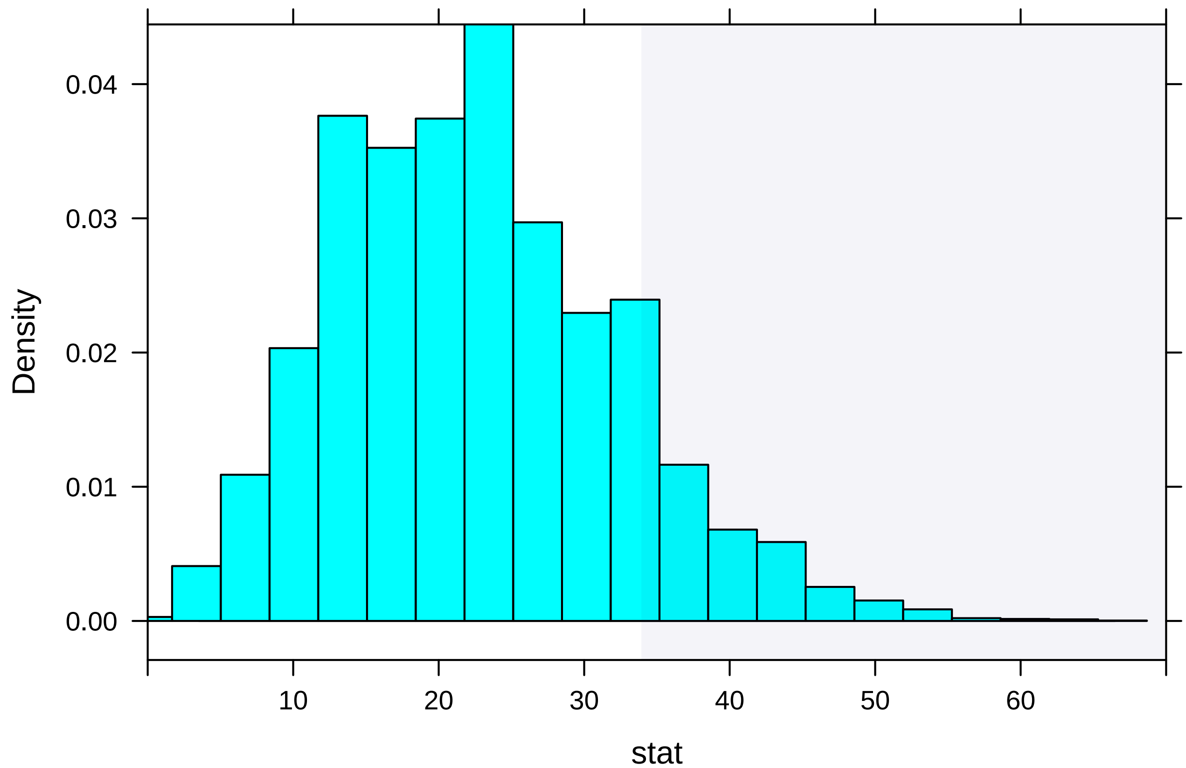
```
## Null distribution appears to be asymmetric. (p = 0.0105)
```

```
##
## Test statistic applied to sample data =  34
```

```
##
## Quantiles of test statistic applied to random data:
```

```
## 50% 90% 95% 99%
##  22  35  40  48
```

```
##
## Of the 10001 samples (1 original + 10000 random),

##
##  193 ( 1.93 % ) had test stats = 34

##
##  1348 ( 13.48 % ) had test stats >= 34

##
```



The first test statistics I will use is the ratio of max to min and I will also use the code from `fastR`

```
teststat=function(x){max(x)/min(x)}
statTally(golfballs,rgolfballs,teststat)
```

```
## Null distribution appears to be asymmetric. (p = 4.85e-12)

##
## Test statistic applied to sample data =  1.327

##
## Quantiles of test statistic applied to random data:
```

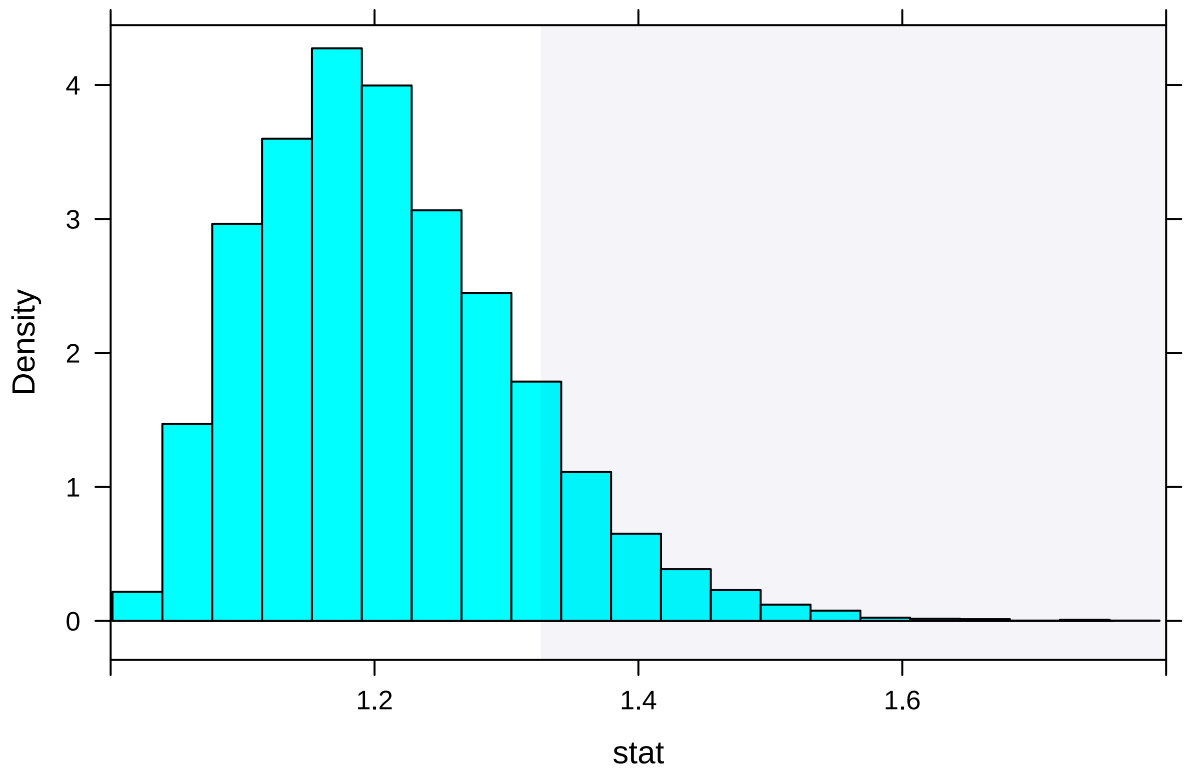
```
##      50%      90%      95%      99%
## 1.196429 1.340000 1.390000 1.490428
```

```
##
## Of the 10001 samples (1 original + 10000 random),
```

```
##
##  23 ( 0.23 % ) had test stats = 1.327
```

```
##
##  1229 ( 12.29 % ) had test stats >= 1.327
```

```
##
```



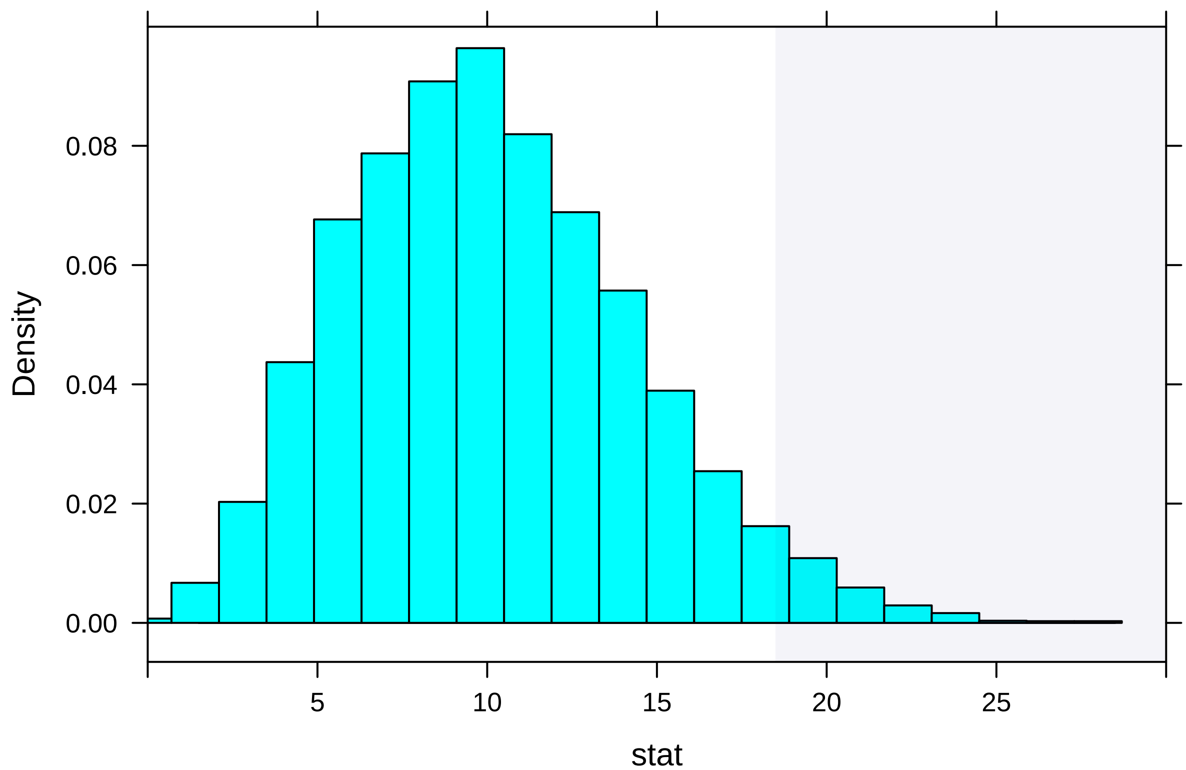
Next, I will use the standard deviation as my test statistic.

```
teststat=function(x){sd(x)}
statTally(golfballs,rgolfballs,teststat)
```

```
## Null distribution appears to be asymmetric. (p = 0.000591)
```

```
##
## Test statistic applied to sample data =  18.52
```

```
##
## Quantiles of test statistic applied to random data:

##       50%       90%       95%       99%
##  9.814955 15.800844 17.710637 21.362116

##
## Of the 10001 samples (1 original + 10000 random),

##
##  4 ( 0.04 % ) had test stats = 18.52

##
##  365 ( 3.65 % ) had test stats >= 18.52

##
```

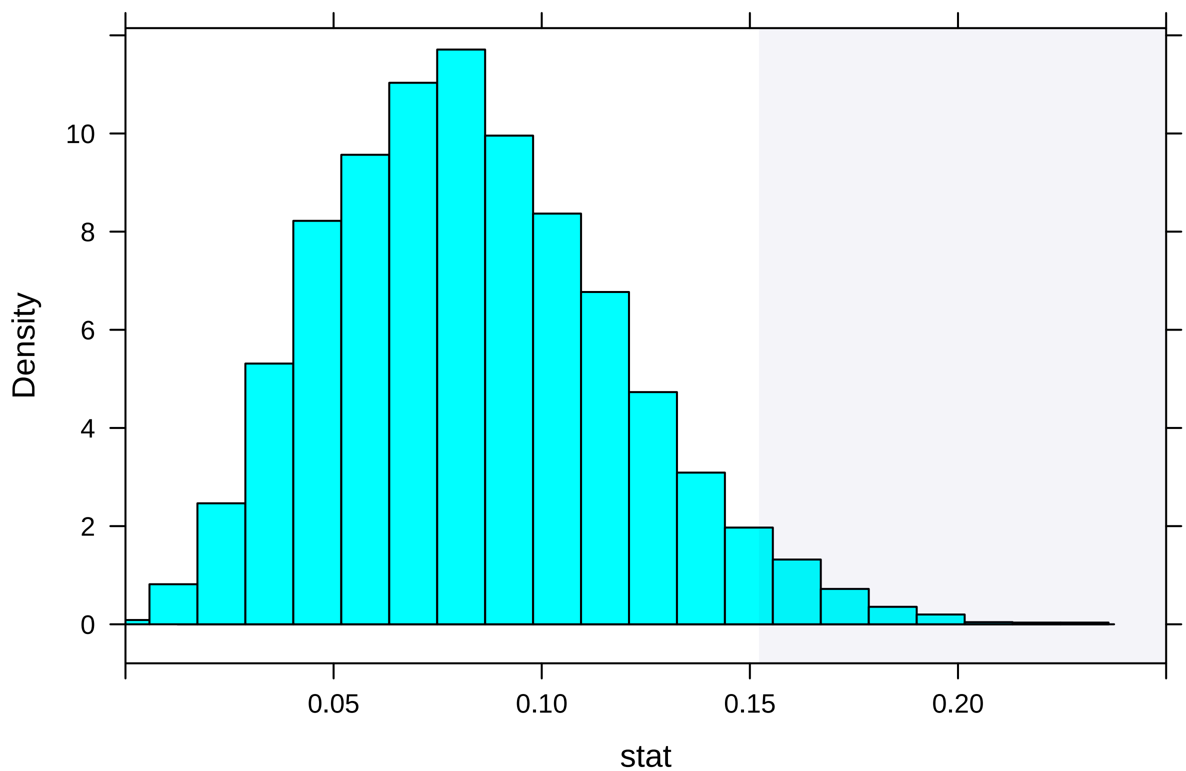


And finally, I will use standard deviation over mean

```
teststat=function(x){sd(x)/mean(x)}
statTally(golfballs,rgolfballs,teststat)
```

```
## Null distribution appears to be asymmetric. (p = 0.000584)

##
## Test statistic applied to sample data =  0.1524
```

```
##
## Quantiles of test statistic applied to random data:

##        50%        90%        95%        99%
## 0.08078152 0.13004810 0.14576656 0.17581988

##
## Of the 10001 samples (1 original + 10000 random),

##
##  4 ( 0.04 % ) had test stats = 0.1524

##
##  365 ( 3.65 % ) had test stats >= 0.1524

##
```



Problem 4.57 Part a.
Find the power of the test statistic in Example 4.10.1 using a true population of .3,.3,.2,.2.

```
teststatbook=function(x){diff(range(x))}
```

Now I need to simulate data from the hypothesized alternative distribution, remember the null was that all probabilities were equal.

```r
rmultinom(n=10,size=486,prob=c(.3,.3,.2,.2))
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]  141  152  140  160  151  153  138  150  153   157
## [2,]  154  150  134  137  149  155  139  139  134   137
## [3,]   95   96  111   98   87   99  106   95   92   104
## [4,]   96   88  101   91   99   79  103  102  107    88
```

Now, I need to determine the rejection criteria to use in calculating power. Looking at the data in Example 4.10.1, I see that the 95-th percentile is 40. Thus I would reject if the test statistic were 40 or greater. Here is the simulated power

```r
set.seed(2016)
sum(apply(rmultinom(n=10000,size=486,prob=c(.3,.3,.2,.2)), 2, teststatbook)>=40)/10000
```

```
## [1] 0.9574
```

Part b.

For my tests

```r
sum(apply(rmultinom(n=10000,size=486,prob=c(.3,.3,.2,.2)), 2, teststat)>=1.39)/10000
```

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

```r
sum(apply(rmultinom(n=10000,size=486,prob=c(.3,.3,.2,.2)), 2, teststat)>=17.71)/10000
```

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

```r
sum(apply(rmultinom(n=10000,size=486,prob=c(.3,.3,.2,.2)), 2, teststat)>=0.145766)/10000
```

```
## [1] 0.9758
```

I am interested in how sensitive the results are to the alternative hypothesis distribution

```r
sum(apply(rmultinom(n=10000,size=486,prob=c(.5,.1,.3,.1)), 2, teststatbook)>=40)/10000
```

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

```r
sum(apply(rmultinom(n=10000,size=486,prob=c(.25,.25,.27,.23)), 2, teststatbook)>=40)/10000
```

```
## [1] 0.1692
```

If I use the null, I should get α of 0.05.

```r
sum(apply(rmultinom(n=10000,size=486,prob=c(.25,.25,.25,.25)), 2, teststatbook)>=40)/10000
```

```
## [1] 0.0561
```

Chapter 5

Section 5.1

Problem 5.1 Example 5.1.4 of the textbook does a nice job deriving the maximum likelihood estimator for a $U(0,\theta)$. The maximum likelihood estimator, MLE, is the maximum data value, $\hat{\theta} = max(X)$. The data from example 4.2.2 of the book are `0.2 0.9 1.9 2.2 4.7 5.1`, thus the MLE of θ is $\hat{\theta} = 5.1$.

The advantage of the MLE for θ is that $\hat{\theta} = 5.1$ will never be less than any data value. This was not the case for the method of moments estimator.

Problem 5.2 Given X is an independent and identically distributed sample from

$$f(x;\theta) = (\theta+1)x^{\theta}\text{on [0,1]}.$$

Part a. Find the method of moments estimator for θ.

First, I need $E(X)$ which by definition is

$$\int_0^1 x(\theta+1)x^{\theta}dx = (\theta+1)\int_0^1 x^{\theta+1}dx$$

$$= \frac{(\theta+1)}{(\theta+2)}x^{\theta+2}|_0^1 = \frac{(\theta+1)}{(\theta+2)}\left(1^{\theta+2} - 0^{\theta+2}\right) = \frac{(\theta+1)}{(\theta+2)}$$

I will set this equal to the first sample mean about the origin and solve for θ

$$\frac{(\theta+1)}{(\theta+2)} = \bar{X}$$

$$\hat{\theta} + 1 = \bar{X}\hat{\theta} + 2\bar{X}$$

$$\hat{\theta}\left(1 - \bar{X}\right) = 2\bar{X} - 1$$

$$\hat{\theta} = \frac{2\bar{X} - 1}{1 - \bar{X}}$$

Part b.

The likelihood function is

$$L(\theta;x) = \prod_{i=1}^{n}\left[(\theta+1)x_i^{\theta}\right]$$

It is easier to use the log-likelihood

$$l(\theta;x) = log\left(\prod_{i=1}^{n}\left[(\theta+1)x_i^{\theta}\right]\right) = \sum_{i=1}^{n} log\left[(\theta+1)x_i^{\theta}\right] = \sum_{i=1}^{n} log\left[(\theta+1)\right] + \sum_{i=1}^{n} log\left[x_i^{\theta}\right] = nlog\left[(\theta+1)\right] + \theta\sum_{i=1}^{n} log\left[x_i\right]$$

Since the log-likelihood is a continuous function of θ we can find the maximum by using differentiation.

$$\frac{dl}{d\theta} = \frac{n}{(\theta+1)} + \sum_{i=1}^{n} log\left[x_i\right] = 0$$

$$\theta + 1 = \frac{-n}{\sum_{i=1}^{n} log\left[x_i\right]}$$

$$\hat{\theta} = -1 - \frac{n}{\sum_{i=1}^{n} log\left[x_i\right]}$$

Part c.

Given the data in the problem, find the method of moments estimate.

```
Prob5.2=c(0.9,0.078,.93,.64,.45,.85,.75,.93,.98,.78)
(2*mean(Prob5.2)-1)/(1-mean(Prob5.2))
```

```
## [1] 1.687316
```

Part d.

Find the maximum likelihood estimate

```
-1-length(Prob5.2)/sum(log(Prob5.2))
```

```
## [1] 1.098544
```

Problem 5.9 Repeat Example 5.1.7 using numeric methods.

```
library(fastR)
```

First, I need to create the log-likelihood function.

```
loglik=function(theta,x){(2*x[1]+x[2])*log(theta)+(2*x[3]+x[2])*log(1-theta)}
```

Now, I will enter the data and then find the maximum using a numeric solver

```
Prob5.9=c(83,447,470)
summary(nlmax(loglik,p=sqrt(83/(83+447+470)),Prob5.9))
```

```
##
##        Maximum: -1232.5431
##       Estimate:0.3064995
##       Gradient:-0.0001377884
##     Iterations: 3
##
## Relative gradient is close to zero, current iterate is probably an
## approximate solution.[Code=1]
```

```
nlmax(loglik,p=sqrt(83/(83+447+470)),Prob5.9)$estimate
```

```
## [1] 0.3064995
```

Problem 5.12 Let X and Y be continuous random variables. We define a new random variable Z as a mixture of X and Y where we sample from X α proportion of the time.

Part a.
Find the CDF for X.

$$F_Z(z) = P(Z \leq Z) = \alpha P(X \leq z) + (1-\alpha)P(Y \leq z)$$

Part b. Find the pdf of Z

$$\begin{aligned} f_Z(z) = \frac{dF_Z(z)}{dz} &= \alpha\frac{dF_X(z)}{dz} + (1-\alpha)\frac{dF_Y(z)}{dz} \\ &= \alpha\frac{dF_X(z)}{dx}\frac{dx}{dz} + (1-\alpha)\frac{dF_Y(z)}{dy}\frac{dy}{dz} \\ &= \alpha f_X(z)\frac{dx}{dz} + (1-\alpha)f_Y(z)\frac{dy}{dz} \end{aligned}$$

Since Z is a simple linear combination of X and Y

$$\frac{dx}{dz} = 1$$

and

$$\frac{dy}{dz} = 1$$

Thus the probability density function for Z is

$$f_Z(z) = \alpha f_X(z) + (1-\alpha)f_Y(z)$$

Part c.

Find $P(W < 12)$

$$P(W < 12) = .3P(X < 12) + .7P(Y < 12)$$

Using R we have

```
.3*pnorm(12,8,2)+.7*pnorm(12,16,3)
```

```
## [1] 0.3570228
```

Section 5.2

Problem 5.17 Suppose that X is a random variable that has the following pdf

$$f(x;\theta) = \begin{cases} (\theta+1)x^\theta & 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

where $\theta \geq 1$. We want to test the hypothesis

$$H_0 : \theta = 0$$

$$H_a : \theta > 0$$

for a sample size of 1.

Part a.

When θ is zero, under the null, we have a uniform distribution in which all values of x are equally likely. If *theta* is greater than zero, we have a polynomial for the pdf. Thus larger values of x will have larger density and thus we should expect larger x value under the alternative. Thus we should reject if x is large.

Part b.

Under the null hypothesis, the pdf is

$$f(x;\theta) = 1$$

The p-value is the probability of the data or more extreme given that the null hypothesis is true. More extreme under the alternative is a larger x value. Thus the p-value is

$$P(X \geq .2 \mid \theta = 0) = 1 - P(X < .2 \mid \theta = 0) = 1 - .2 = .8$$

Part c.

Using similar reasoning as part b. The p-value is

$$P(X \geq .9 \mid \theta = 0) = 1 - P(X < .9 \mid \theta = 0) = 1 - .9 = .1$$

Part d.

For a significance level of $\alpha = 0.05$ the rejection criteria would be

$$p - value < .05 \Rightarrow P(X \geq x \mid \theta = 0) < .05 \Rightarrow X \geq 0.95$$

Part e.

Find the power of the test if $\theta = 1$. I need to find the probability of rejecting, given the alternative hypothesis is true;

$$P(X \geq 0.95 \mid \theta = 1)$$

This is, where I substituted $\theta = 1$ into the pdf,

$$= \int_{.95}^{1} 2x dx$$

$$= x^2|_{.95}^{1}$$

$$= (1 - .95^2)$$

```
(1-.95^2)
```

```
## [1] 0.0975
```

This is a small power, so a high probability of a type II error. But again, our sample size is only 1, so we would not expect power to be very high.

Problem 5.18 Find the likelihood ratio test for Problem 5.17. First I need the likelihood under the null hypothesis. Under the null, the pdf is $f(x;\theta) = 1$ and thus the likelihood function is

$$L(\theta; x) = \prod_{i=1}^{n} 1 = 1$$

Next, I need the maximum likelihood estimator of θ

The likelihood function is

$$L(\theta; x) = \prod_{i=1}^{n} (\theta + 1) x_i^{\theta}$$

The log-likelihood is

$$\ell(\theta; x) = \sum_{i=1}^{n} ln(\theta + 1) + \theta \sum_{i=1}^{n} ln(x_i) = nln(\theta + 1) + \theta \sum_{i=1}^{n} ln(x_i)$$

Since θ is a real non-negative number, I will find the maximum by differentiation

$$\frac{d\ell}{d\theta} = \frac{n}{\theta + 1} + \sum_{i=1}^{n} ln(x_i) = 0$$

$$\theta + 1 = \frac{-n}{\sum_{i=1}^{n} ln(x_i)} \Rightarrow \hat{\theta} = -1 - \frac{n}{\sum_{i=1}^{n} ln(x_i)}$$

The test statistic for the likelihood ration test is

$$\lambda = \frac{L(\theta = 0, x)}{L(\hat{\theta}, x)} = \frac{1}{\prod_{i=1}^{n} (\hat{\theta} + 1) x_i^{\hat{\theta}}}$$

where

$$\hat{\theta} = -1 - \frac{n}{\sum_{i=1}^{n} ln(x_i)}$$

and I would find a p-value from

$$-2ln(\lambda) = 2[nln(\hat{\theta} + 1) + \hat{\theta} \sum_{i=1}^{n} ln(x_i)] \sim \chi^2(1)$$

Problem 5.22 The maximum likelihood estimators for μ and σ from a normal distribution are

$$\hat{\mu} = \bar{x}$$

$$\hat{\sigma^2} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}$$

For the likelihood ratio test, I need

$$\frac{L(\hat{\mu}, \sigma_0)}{L(\hat{\mu}, \hat{\sigma})}$$

where

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}}$$

Now

$$L(\Omega_0) = \prod_{i=1}^{n} \frac{1}{\sigma_0 \sqrt{2\pi}} e^{\frac{-(x_i - \bar{x})^2}{2\sigma_0^2}}$$

and

$$\ell(\Omega_0) = \sum_{i=1}^{n} ln \left(\frac{1}{\sigma_0 \sqrt{2\pi}} e^{\frac{-(x_i - \bar{x})^2}{2\sigma_0^2}} \right)$$

$$= -\sum_{i=1}^{n} ln(\sigma_0) + \sum_{i=1}^{n} ln \left(\frac{1}{\sqrt{2\pi}} \right) + \sum_{i=1}^{n} \left(\frac{-(x_i - \bar{x})^2}{2\sigma_0^2} \right)$$

Likewise

$$L(\Omega) = \prod_{i=1}^{n} \frac{1}{\hat{\sigma}\sqrt{2\pi}} e^{\frac{-(x_i-\bar{x})^2}{2\hat{\sigma}^2}}$$

and

$$\ell(\Omega) = -\sum_{i=1}^{n} ln(\hat{\sigma}) + \sum_{i=1}^{n} ln\left(\frac{1}{\sqrt{2\pi}}\right) + \sum_{i=1}^{n}\left(\frac{-(x_i-\bar{x})^2}{2\hat{\sigma}^2}\right)$$

The test statistic is

$$-2ln(\Lambda) = 2\left(\ell(\Omega) - \ell(\Omega_0)\right)$$

$$= 2\left[-\sum_{i=1}^{n} ln(\hat{\sigma}) + \sum_{i=1}^{n}\left(\frac{-(x_i-\bar{x})^2}{2\hat{\sigma}^2}\right) + \sum_{i=1}^{n} ln(\sigma_0) + \sum_{i=1}^{n}\left(\frac{(x_i-\bar{x})^2}{2\sigma_0^2}\right)\right]$$

To simplify, note

$$\hat{\sigma^2} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n} \Rightarrow \sum_{i=1}^{n}(x_i-\bar{x})^2 = n\hat{\sigma^2}$$

so

$$\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{2\hat{\sigma^2}} = \frac{n}{2}$$

and

$$\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{2\sigma_0^2} = \frac{n\hat{\sigma^2}}{2\sigma_0^2}$$

Thus

$$-2ln(\Lambda) = 2nln\left(\frac{\sigma_0}{\hat{\sigma}}\right) - n + \frac{n\hat{\sigma^2}}{\sigma_0^2} \sim \chi^2(1)$$

Section 5.4

```
library(fastR)
```

Problem 5.19 In Chapter 4, problem 4.56, I did the following tests in R

```
teststat<-function(x){max(x)/min(x)}
statTally(golfballs,rgolfballs,teststat)
teststata<-function(x){sd(x)}
statTally(golfballs,rgolfballs,teststata)
teststatb<-function(x){sd(x)/mean(x)}
statTally(golfballs,rgolfballs,teststatb)
```

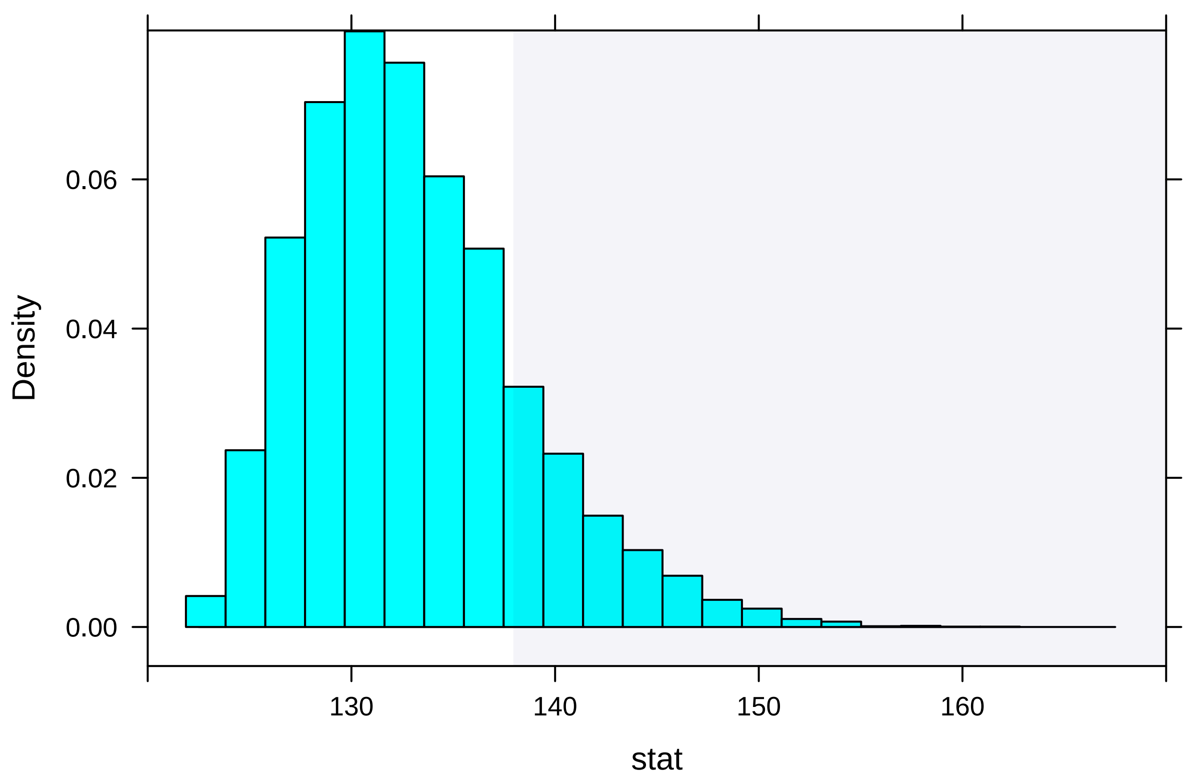
My three test statistics were a ratio of max to min, the standard deviation, and a ratio of the standard deviation to the mean. Now, I will experiment with statistics that use less of the data, the max and the min

```
statTally(golfballs,rgolfballs,max)
```

```
## Null distribution appears to be asymmetric. (p = 2.24e-27)
```

```
##
## Test statistic applied to sample data =  138
```

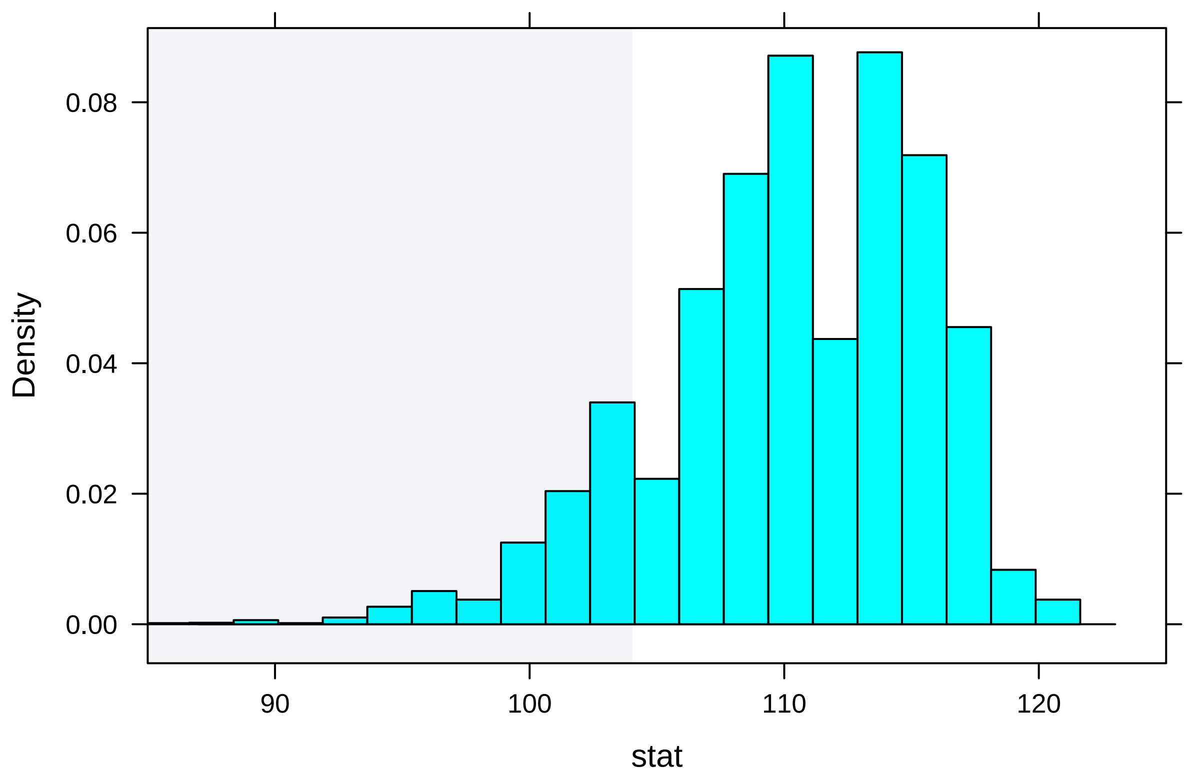
```
##
## Quantiles of test statistic applied to random data:

## 50% 90% 95% 99%
## 132 140 143 149

##
## Of the 10001 samples (1 original + 10000 random),

##
##  360 ( 3.6 % ) had test stats = 138

##
##  1869 ( 18.69 % ) had test stats >= 138

##
```




```
statTally(golfballs,rgolfballs,min)
```

```
## Null distribution appears to be asymmetric. (p = 1.18e-15)

##
## Test statistic applied to sample data =  104
```

```
##
## Quantiles of test statistic applied to random data:

## 50% 90% 95% 99%
## 111 117 118 119

##
## Of the 10001 samples (1 original + 10000 random),

##
##  339 ( 3.39 % ) had test stats = 104

##
##  1413 ( 14.13 % ) had test stats <= 104

##
```

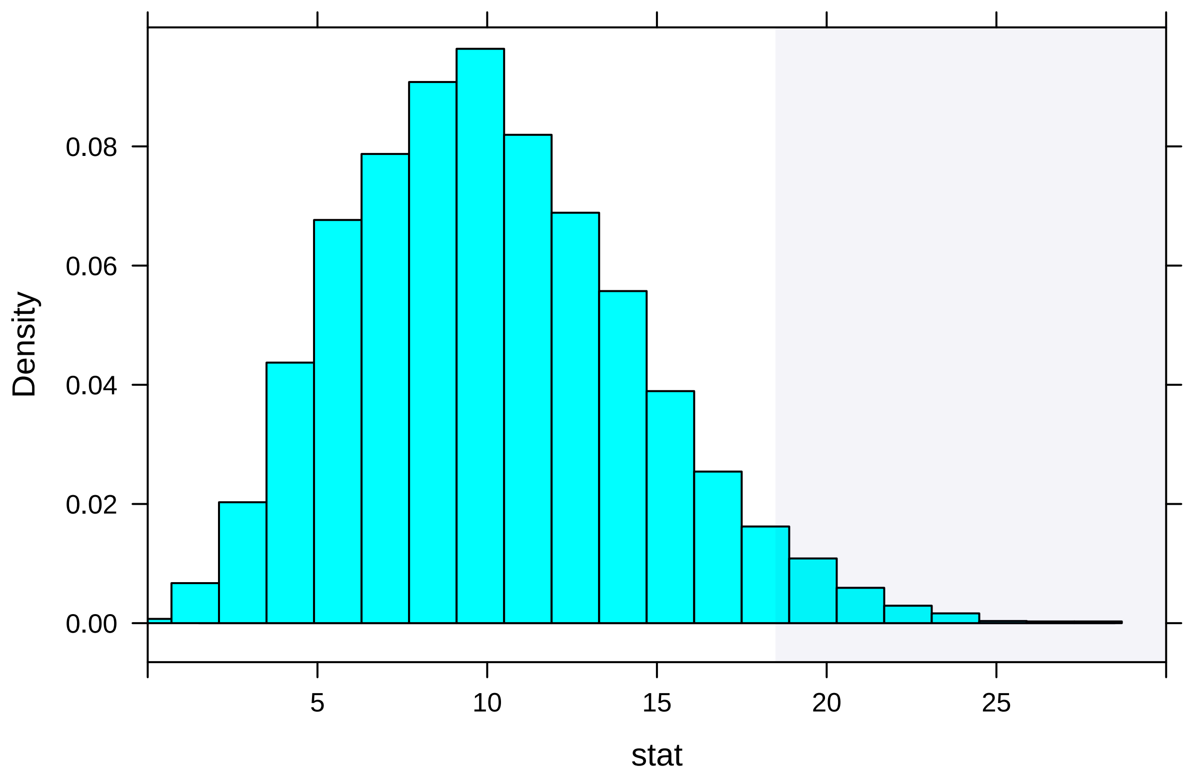


As the author points out, those test that use more of the data, tend to have smaller p-values and more power. Thus, from problem 4.56, we would expect using the standard deviation as the test statistic, would give us a smaller p-value than using the maximum or minimum. I will run the code from problem 4.56 again to verify.

```
teststata<-function(x){sd(x)}
statTally(golfballs,rgolfballs,teststata)
```

```
## Null distribution appears to be asymmetric. (p = 0.000591)
```

```
##
## Test statistic applied to sample data =  18.52

##
## Quantiles of test statistic applied to random data:

##       50%       90%       95%       99%
##  9.814955 15.800844 17.710637 21.362116

##
## Of the 10001 samples (1 original + 10000 random),

##
##  4 ( 0.04 % ) had test stats = 18.52

##
##  365 ( 3.65 % ) had test stats >= 18.52

##
```

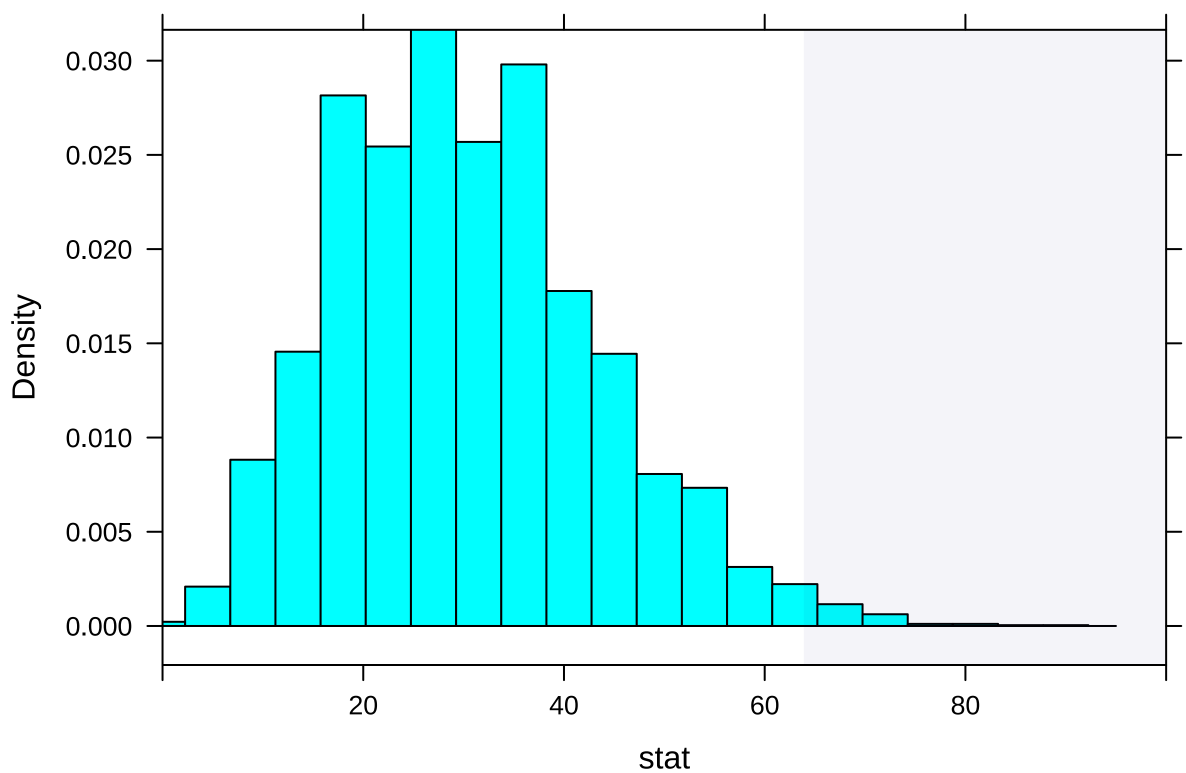


And the simulation confirms the result. In fact, we now reject the null hypothesis.

Finally, I will run two of the test proposed on page 279 of the text. First using the sum of the absolute deviations

```
teststatc<-function(x){sum(abs(x-mean(x)))}
statTally(golfballs,rgolfballs,teststatc)
```

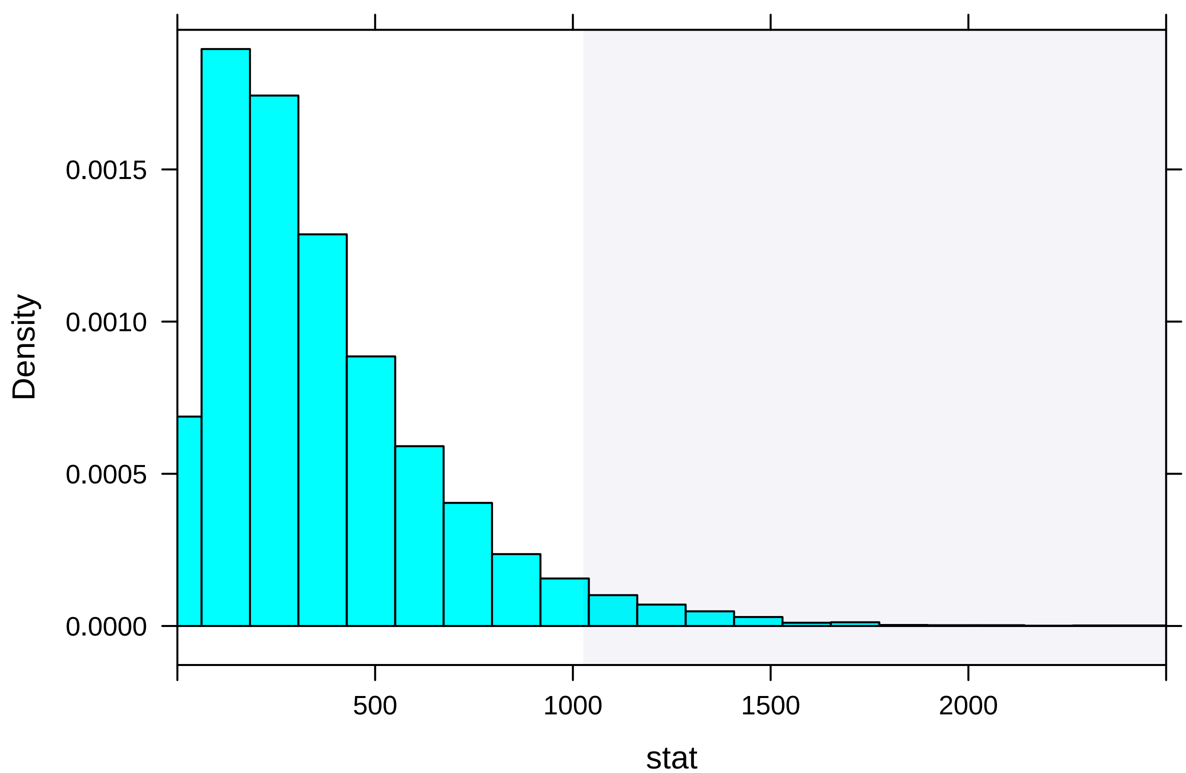
```
## Null distribution appears to be asymmetric. (p = 1.64e-10)

##
## Test statistic applied to sample data =  64

##
## Quantiles of test statistic applied to random data:

## 50% 90% 95% 99%
##  29  48  54  65

##
## Of the 10001 samples (1 original + 10000 random),

##
##  30 ( 0.3 % ) had test stats = 64

##
##  139 ( 1.39 % ) had test stats >= 64

##
```



Next, the sum of the squared deviations

```
teststatd<-function(x){sum((x-mean(x))^2)}
statTally(golfballs,rgolfballs,teststatd)
```

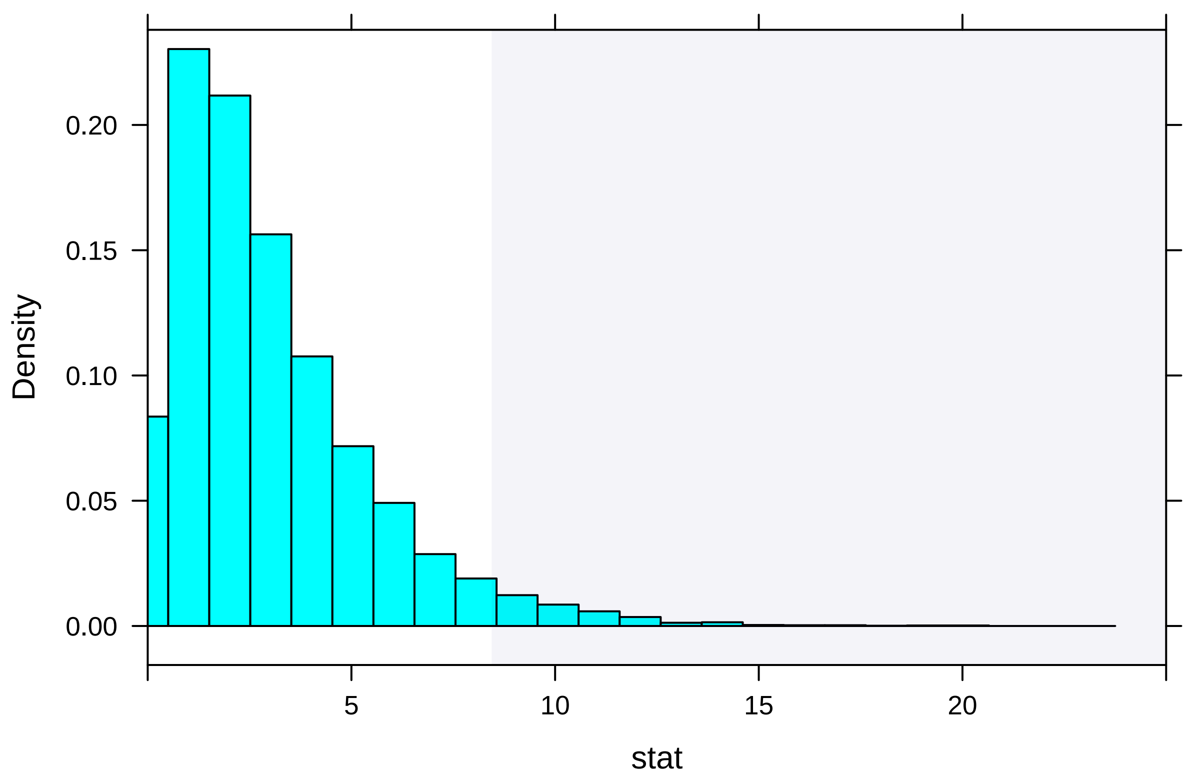
```
## Null distribution appears to be asymmetric. (p = 4.65e-38)

##
## Test statistic applied to sample data =  1029

##
## Quantiles of test statistic applied to random data:

##     50%     90%     95%     99%
##  289.00  749.00  941.00 1369.02

##
## Of the 10001 samples (1 original + 10000 random),

##
##  4 ( 0.04 % ) had test stats = 1029

##
##  365 ( 3.65 % ) had test stats >= 1029

##
```



Then the sum of the squared deviation, normalized by the mean

```
teststate<-function(x){sum((x-mean(x))^2/mean(x))}
statTally(golfballs,rgolfballs,teststate)
```

```
## Null distribution appears to be asymmetric. (p = 4.16e-38)

##
## Test statistic applied to sample data =  8.469

##
## Quantiles of test statistic applied to random data:

##       50%       90%       95%       99%
##  2.378601  6.164609  7.744856 11.267654

##
## Of the 10001 samples (1 original + 10000 random),

##
##  4 ( 0.04 % ) had test stats = 8.469

##
##  365 ( 3.65 % ) had test stats >= 8.469

##
```



Finally, I will run the Pearson chi-squared test

```r
chisq.test(golfballs)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  golfballs
## X-squared = 8.4691, df = 3, p-value = 0.03725
```

Here is the same test with estimated p-value

```r
E<-rep(486/4,4)
teststatf<-function(x){sum((x-E)^2/E)}
statTally(golfballs,rgolfballs,teststatf)
```

```
## Null distribution appears to be asymmetric. (p = 4.16e-38)
```

```
##
## Test statistic applied to sample data =  8.469
```

```
##
## Quantiles of test statistic applied to random data:
```

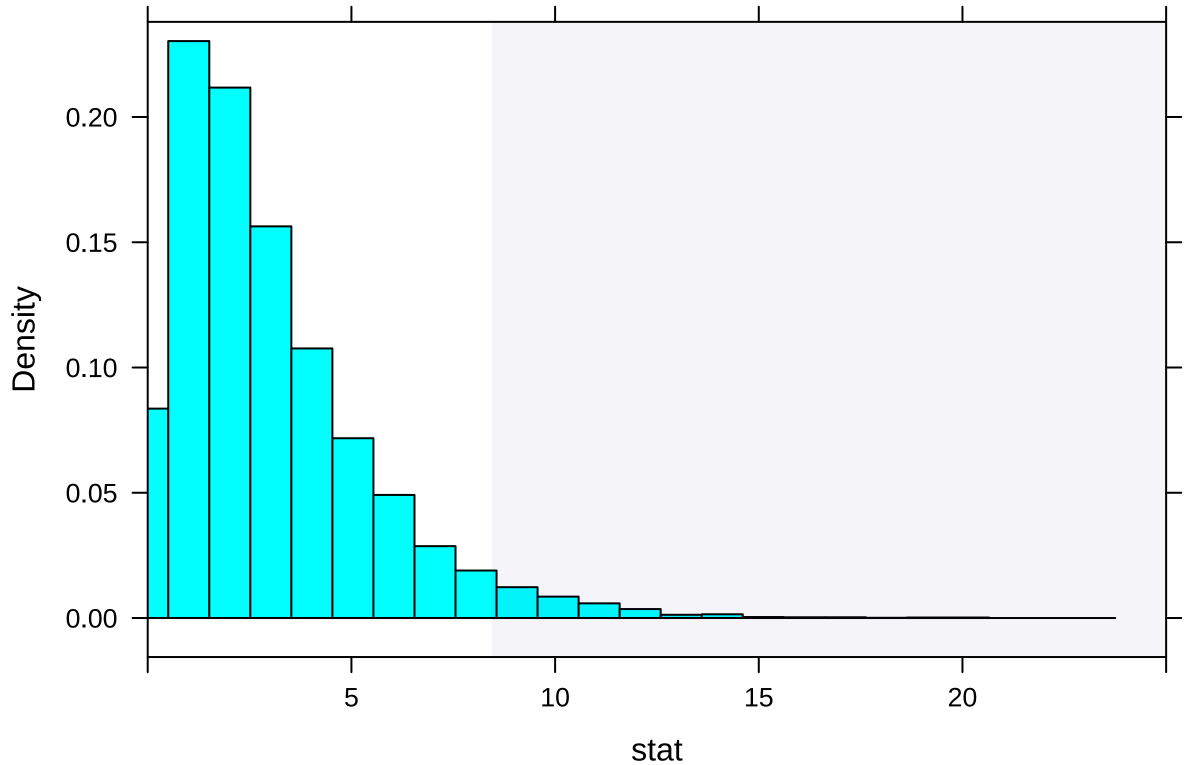
```
##       50%       90%       95%       99%
##  2.378601  6.164609  7.744856 11.267654
```

```
##
## Of the 10001 samples (1 original + 10000 random),
```

```
##
##  4 ( 0.04 % ) had test stats = 8.469
```

```
##
##  365 ( 3.65 % ) had test stats >= 8.469
```

```
##
```



For completeness, the likelihood ration test is

```
(G<-2*sum(golfballs*log(golfballs/E)))
```

```
## [1] 8.498805
```

```
1-pchisq(G,df=3)
```

```
## [1] 0.03675295
```

or an estimated p-value

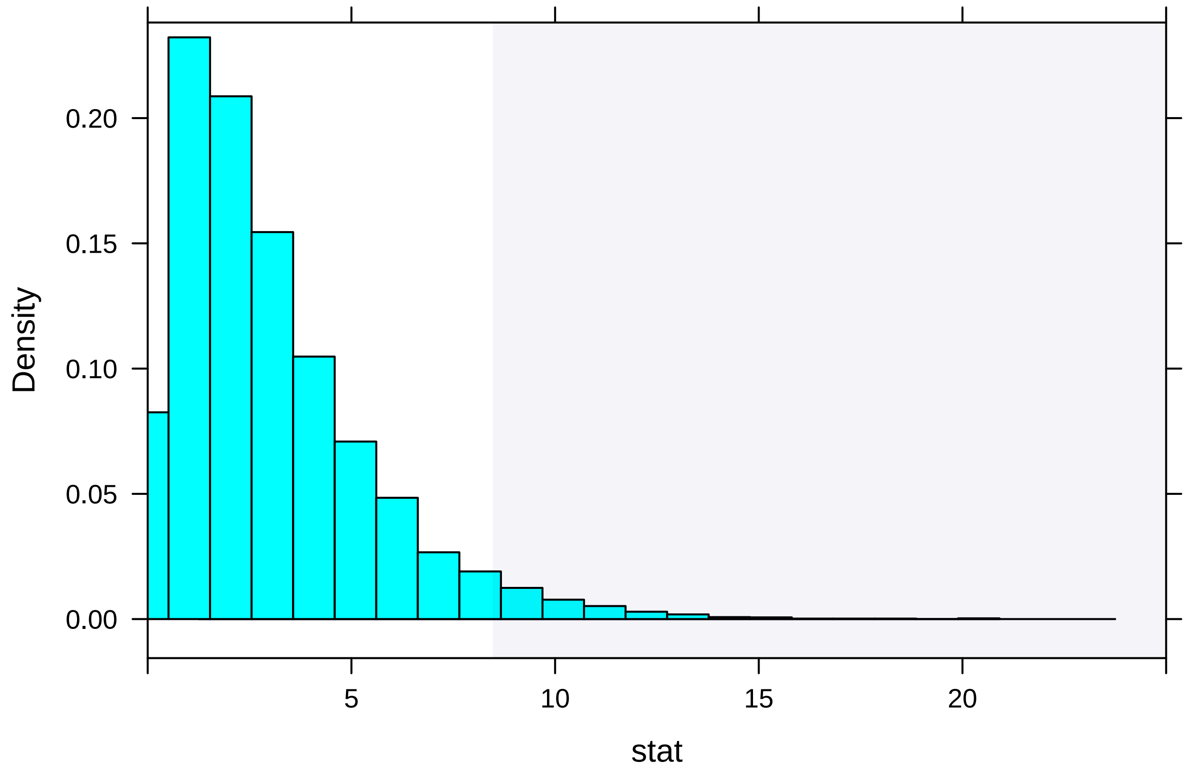
```
teststatg<-function(x){2*sum(x*log(x/E))}
statTally(golfballs,rgolfballs,teststatg)
```

```
## Null distribution appears to be asymmetric. (p = 6.81e-38)
```

```
##
## Test statistic applied to sample data =  8.499
```

```
##
## Quantiles of test statistic applied to random data:
```

```
##       50%       90%       95%       99%
##  2.380827  6.149516  7.764158 11.212036

##
## Of the 10001 samples (1 original + 10000 random),

##
##  1 ( 0.01 % ) had test stats = 8.499

##
##  356 ( 3.56 % ) had test stats >= 8.499

##
```



Problem 5.20 Repeat Example 5.4.9 using normal, gamma, and Weibull.

First enter the data

```
data <- c(18.0,6.3,7.5,8.1,3.1,0.8,2.4,3.5,9.5,39.7,
           3.4,14.6,5.1,6.8,2.6,8.0,8.5,3.7,21.2,3.1,
           10.2, 8.3,6.4,3.0,5.7,5.6,7.4,3.9,9.1,4.0)
```

Next, using the code from the book, I will bin the data and set cut-points

```r
n <- length(data)
cutpts <- c(0,2,4,7,12,Inf)
# cutpts<-c(0,4,8,12,Inf) Try these cut-points to see the impact
bin.data <- cut(data, cutpts)
```

To check that my code is correct, I will repeat the analysis from Example 5.4.9 using the exponential distribution.

```r
theta.hat <- 1/mean(data); theta.hat
```

```
## [1] 0.125261
```

```r
p <- diff(pexp(cutpts,theta.hat))
e <- n * p
o <- table(bin.data)
print(cbind(o,e))
```

```
##           o        e
## (0,2]     1 6.648167
## (2,4]    10 5.174896
## (4,7]     6 5.693899
## (7,12]    9 5.810061
## (12,Inf]  4 6.672975
```

```r
lrt  <- 2 * sum(o * log(o/e)); lrt
```

```
## [1] 13.79823
```

```r
pearson <- sum( (o-e)^2/e ); pearson
```

```
## [1] 12.1361
```

```r
1-pchisq(lrt, df=3)              # df = (5 - 1) - 1 [anti-conservative]
```

```
## [1] 0.003193062
```

```r
1-pchisq(pearson,df=3)
```

```
## [1] 0.0069312
```

```r
1-pchisq(lrt, df=4)              # df = 5 - 1       [conservative]
```

```
## [1] 0.007967651
```

```r
1-pchisq(pearson,df=4)
```

```
## [1] 0.0163674
```

Now, I will repeat using the normal. First, I will obtain the maximum likelihood estimates for the mean and standard deviation, both numerically and from the known formulas

```r
logliknorm<-function(theta,x){sum(dnorm(x,mean=theta[1],sd=theta[2],log=T))}
summary(nlmax(logliknorm,p=c(6,5),x=data))
```

```
##
##        Maximum: -102.5538
##       Estimate:7.983329 7.385526
##       Gradient:-2.136079e-08 -8.658671e-08
##     Iterations: 11
##
## Relative gradient is close to zero, current iterate is probably an
## approximate solution.[Code=1]
```

```r
mean(data)
```

```
## [1] 7.983333
```

```r
sqrt(sum((data-mean(data))^2)/length(data))
```

```
## [1] 7.38553
```

Next, I will calculate the p-values for the goodness of fit

```r
theta.hat<-c(mean(data),sqrt(sum((data-mean(data))^2)/length(data)))
p<-diff(pnorm(cutpts,mean=theta.hat[1],sd=theta.hat[2]))
n<-length(data)
e<-n*p
o<-table(bin.data)
print(cbind(o,e))
```

```
##           o        e
## (0,2]     1 2.072032
## (2,4]    10 2.576881
## (4,7]     6 4.566448
## (7,12]    9 7.790692
## (12,Inf]  4 8.798106
```

```r
lrt<-2*sum(o*log(o/e))
pearson<-sum((o-e)^2/e)
1-pchisq(lrt,5-1-2)
```

```
## [1] 3.320649e-06
```

```r
1-pchisq(pearson,5-1-2)
```

```
## [1] 3.384584e-06
```

```r
1-pchisq(lrt,5-1)
```

```
## [1] 4.52118e-05
```

```
1-pchisq(pearson,5-1)
```

```
## [1] 4.601775e-05
```

These p-values are even smaller, thus I reject the claim that the data comes from a normal distribution.

Now the gamma

```
loglikgamma<-function(theta,x){sum(dgamma(x,theta[1],theta[2],log=T))}
summary(nlmax(loglikgamma,p=c(6,5),x=data))
```

```
##
##        Maximum: -89.28777
##       Estimate:1.8978913 0.2377317
##       Gradient:-2.597523e-06  2.293326e-05
##     Iterations: 17
##
## Relative gradient is close to zero, current iterate is probably an
## approximate solution.[Code=1]
```

And the p-values

```
theta.hat<-nlmax(loglikgamma,p=c(6,5),x=data)$estimate
```

```
## Warning in dgamma(x, theta[1], theta[2], log = T): NaNs produced

## Warning in dgamma(x, theta[1], theta[2], log = T): NaNs produced

## Warning in nlm(g, ...): NA/Inf replaced by maximum positive value

## Warning in dgamma(x, theta[1], theta[2], log = T): NaNs produced

## Warning in dgamma(x, theta[1], theta[2], log = T): NaNs produced

## Warning in nlm(g, ...): NA/Inf replaced by maximum positive value

## Warning in dgamma(x, theta[1], theta[2], log = T): NaNs produced

## Warning in dgamma(x, theta[1], theta[2], log = T): NaNs produced

## Warning in nlm(g, ...): NA/Inf replaced by maximum positive value

## Warning in dgamma(x, theta[1], theta[2], log = T): NaNs produced

## Warning in dgamma(x, theta[1], theta[2], log = T): NaNs produced

## Warning in nlm(g, ...): NA/Inf replaced by maximum positive value

## Warning in dgamma(x, theta[1], theta[2], log = T): NaNs produced

## Warning in dgamma(x, theta[1], theta[2], log = T): NaNs produced

## Warning in nlm(g, ...): NA/Inf replaced by maximum positive value
```

```r
p<-diff(pgamma(cutpts,theta.hat[1],theta.hat[2]))
n<-length(data)
e<-n*p
o<-table(bin.data)
print(cbind(o,e))
```

```
##            o        e
## (0,2]      1 2.957842
## (2,4]     10 5.282845
## (4,7]      6 7.589399
## (7,12]     9 8.141385
## (12,Inf]   4 6.028529
```

```r
lrt<-2*sum(o*log(o/e))
pearson<-sum((o-e)^2/e)
1-pchisq(lrt,5-1-2)
```

```
## [1] 0.04292376
```

```r
1-pchisq(pearson,5-1-2)
```

```
## [1] 0.03662676
```

```r
1-pchisq(lrt,5-1)
```

```
## [1] 0.1780619
```

```r
1-pchisq(pearson,5-1)
```

```
## [1] 0.1577506
```

Again, I reject that the data comes from a gamma distribution.

Finally the Weibull

```r
loglikwei<-function(theta,x){sum(dweibull(x,theta[1],theta[2],log=T))}
summary(nlmax(loglikwei,p=c(6,5),x=data))
```

```
##
##        Maximum: -90.60571
##       Estimate:1.290669 8.726359
##       Gradient: 1.689004e-05 -2.688650e-06
##     Iterations: 35
##
## Relative gradient is close to zero, current iterate is probably an
## approximate solution.[Code=1]
```

```
theta.hat=nlmax(loglikwei,p=c(6,5),x=data)$estimate
p<-diff(pweibull(cutpts,theta.hat[1],theta.hat[2]))
n<-length(data)
e<-n*p
o<-table(bin.data)
print(cbind(o,e))
```

```
##           o        e
## (0,2]     1 4.162154
## (2,4]    10 5.020055
## (4,7]     6 6.680493
## (7,12]    9 7.500429
## (12,Inf]  4 6.636868
```

```
lrt<-2*sum(o*log(o/e))
pearson<-sum((o-e)^2/e)
1-pchisq(lrt,5-1-2)
```

```
## [1] 0.01184546
```

```
1-pchisq(pearson,5-1-2)
```

```
## [1] 0.01252948
```

```
1-pchisq(lrt,5-1)
```

```
## [1] 0.06438966
```

```
1-pchisq(pearson,5-1)
```

```
## [1] 0.06740448
```

Problem 5.21 Test the goodness of fit of the data to the model in Example 5.1.7. The model is

AA	Aa	aa
θ^2	$2\theta(1-\theta)$	$(1-\theta)^2$

and the data is

AA	Aa	aa
83	447	470

I will create variables with the observed values and the expected, in Example 5.1.7, the maximum likelihood estimate of θ was found to be 0.3065. using the invariance property of maximum likelihood estimators, which is not taught in the book, we can find the maximum likelihood estimators for each cell by substituting 0.3065 for θ.

```r
o<-c(83,447,470)
theta.hat<-.3065
probs<-c(theta.hat^2,2*theta.hat*(1-theta.hat),(1-theta.hat)^2)
```

First using built-in code

```r
chisq.test(o,p=probs)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  o
## X-squared = 2.6501, df = 2, p-value = 0.2658
```

And then step by step

```r
e=sum(o)*probs
print(cbind(o,e))
```

```
##        o         e
## [1,]  83  93.94225
## [2,] 447 425.11550
## [3,] 470 480.94225
```

```r
lrt=2*sum(o*log(o/e))
pearson=sum((o-e)^2/e)
1-pchisq(lrt,3-1)
```

```
## [1] 0.2610967
```

```r
1-pchisq(pearson,3-1)
```

```
## [1] 0.265792
```

We fail to reject that our data fits the Hardy-Weinberg model. Note: we used the conservative estimate of the p-value. Many books would use the anit-conservative and subtract one more degree of freedom for estimating θ in the null.

Section 5.5

Problem 5.23 The following is the hypothesized probability mass function for the problem

Starchy-Green	Starchy-White	Sugary-Green	Sugary-White
$\frac{1}{4}(2+\theta)$	$\frac{1}{4}(1-\theta)$	$\frac{1}{4}(1-\theta)$	$\frac{1}{4}\theta$

for $1 \le \theta \le 1$.

Part a.

The likelihood function is

$$L(\theta) = \left[\frac{1}{4}(2+\theta)\right]^{x_1} \left[\frac{1}{4}(1-\theta)\right]^{x_2} \left[\frac{1}{4}(1-\theta)\right]^{x_3} \left[\frac{1}{4}\theta\right]^{x_4}$$

and the log likelihood is

$$\ell(\theta) = x_1 log(1/4) + x_1 log(2+\theta) + (x_2+x_3)log(1/4) + (x_2+x_3)log(1-\theta) + x_4 log(1/4) + x_4 log(\theta)$$

Maximizing the log likelihood with respect to θ

$$\frac{d\ell}{d\theta} = \frac{x_1}{2+\theta} - \frac{x_2+x_3}{1-\theta} + \frac{x_4}{\theta} = 0$$

or

$$\frac{x_1}{2+\theta} + \frac{x_4}{\theta} = \frac{x_2+x_3}{1-\theta}$$

Solving for θ

$$\theta * 1-\theta)x_1 + (1-\theta)(2+\theta)x_4 = \theta(2+\theta)(x_2+x_3)$$

This is a quadratic in θ

$$(x_1+x_2+x_3+x_4)\theta^2 + (2x_2+2x_3-x_1+x_4)\theta - 2x_4 = 0$$

so

$$\hat{\theta} = \frac{-(2x_2+2x_3-x_1+x_4) \pm \sqrt{(2x_2+2x_3-x_1+x_4)^2 - 4(x_1+x_2+x_3+x_4)(-2x_4)}}{2(x_1+x_2+x_3+x_4)}$$

For this problem

$$x_1 = 1997, x_2 = 906, x_3 = 904, \text{ and } x_4 = 32$$

thus

$$\sum x_i = 3839 \text{ and } (2x_2+2x_3-x_1+x_4) = 1655$$

Finally

$$\hat{\theta} = \frac{-1655 \pm \sqrt{1655^2 - 4(3839)(-2)(32)}}{2(3839)} = .035712, -.46681$$

or

$$\hat{\theta} = .035712$$

Let's check in R

```
library(fastR)
```

```
Prob5.23<-c(1997,906,904,32)
loglike523<-function(theta,x){x[1]*log(2+theta)+(x[2]+x[3])*log(1-theta)+x[4]*log(theta)}
summary(nlmax(loglike523,p=.5,x=Prob5.23))$estimate
```

```
##
##        Maximum: 1247.1050
##       Estimate:0.03571182
##       Gradient:0.0006004939
##     Iterations: 9
##
## Relative gradient is close to zero, current iterate is probably an
## approximate solution.[Code=1]
```

```
## NULL
```

```r
(Prob523mle<-nlmax(loglike523,p=.5,x=Prob5.23)$estimate)
```

```
## [1] 0.03571182
```

Part b.

Test

$$H_o : \theta = .05$$
$$H_a : \theta \neq .05$$

The likelihood ratio test is

$$LRT = \Lambda = \frac{L(.05)}{L(\hat{\theta})}$$

and $-2log(\Lambda)$ is

$$2[\ell(\hat{\theta}) - \ell(.05)] \dot{\sim} \chi^2(1)$$

```r
2*(loglike523(Prob523mle,Prob5.23)-loglike523(.05,Prob5.23))
```

```
## [1] 4.566447
```

```r
1-pchisq(2*(loglike523(Prob523mle,Prob5.23)-loglike523(.05,Prob5.23)),1)
```

```
## [1] 0.03260412
```

We will repeat this two more times so let's write a function

```r
Prob523LRT <- function(thetamle,thetanull,data) {
 teststat <- 2 * (loglike523(thetamle,data) - loglike523(thetanull,data))
 pvalue <- 1 - pchisq(teststat,df=1)
 return( c(Teststat = teststat, p.value = pvalue))
 }
```

Test it

```r
Prob523LRT(Prob523mle,.05,Prob5.23)
```

```
##   Teststat    p.value
## 4.56644699 0.03260412
```

Part c.

```r
Prob523LRT(Prob523mle,.03,Prob5.23)
```

```
##  Teststat   p.value
## 0.9970687 0.3180208
```

Part d.

```r
Prob523LRT(Prob523mle,.07,Prob5.23)
```

```
##     Teststat      p.value
## 2.127940e+01 3.969742e-06
```

Part e.

The null hypothesis is that the data fits the model in part a. We have not specified θ so we use the maximum likelihood estimate. We find the expected values by using by multiplying the probabilities times the total sample size.

```r
(e5.23<-sum(Prob5.23)*(1/4)*c(2+Prob523mle,1-Prob523mle,1-Prob523mle,Prob523mle))
```

```
## [1] 1953.77442  925.47558  925.47558   34.27442
```

```r
print(cbind(o=Prob5.23,e=e5.23))
```

```
##         o          e
## [1,] 1997 1953.77442
## [2,]  906  925.47558
## [3,]  904  925.47558
## [4,]   32   34.27442
```

```r
(sum((Prob5.23-e5.23)^2/e5.23))
```

```
## [1] 2.015438
```

```r
1-pchisq(sum((Prob5.23-e5.23)^2/e5.23),2) #3 free parameters in Ha and 1 in Ho
```

```
## [1] 0.3650508
```

```r
2*sum(Prob5.23*log(Prob5.23/e5.23))
```

```
## [1] 2.018721
```

```r
1-pchisq(2*sum(Prob5.23*log(Prob5.23/e5.23)),2)
```

```
## [1] 0.3644519
```

Problem 5.27 First I will enter the data

```r
o5.27<-c(315,101,108,32)
```

The hypothesis is

$$H_o : \pi_1 = 9/16, \pi_2 = 3/16, \pi_3 = 3/16, \text{ and } \pi_4 = 1/16$$

$$H_a : \text{At least one not true}$$

Using the built in function

```r
chisq.test(o5.27,p=c(9,3,3,1),rescale=T)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  o5.27
## X-squared = 0.47002, df = 3, p-value = 0.9254
```

Or manually

```r
e5.27<-c(9,3,3,1)/16*sum(o5.27)
1-pchisq(sum((o5.27-e5.27)^2/e5.27),3)
```

```
## [1] 0.9254259
```

Using a LRT with the multinomial

```r
1-pchisq(2*sum(o5.27*log(o5.27/e5.27)),3)
```

```
## [1] 0.9242519
```

The p-value is large so we fail to reject. The large p-value means the data fits the model well. Some people have questioned how well this data fits the model bringing into question where the data was falsified in some manner.

Let's simulate power for this test

```r
rmultinom(2,556,prob=c(1,1,1,1))
```

```
##      [,1] [,2]
## [1,]  162  128
## [2,]  133  132
## [3,]  112  151
## [4,]  149  145
```

```r
chisqteststat=function(x){sum((x-e5.27)^2/e5.27)}
chisqteststat(o5.27)
```

```
## [1] 0.470024
```

```r
sum(o5.27)
```

```
## [1] 556
```

```r
sum(apply(rmultinom(n=10000,size=556,prob=c(1,1,1,1)), 2, chisqteststat)>=qchisq(.95,3))/10000
```

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

```r
sum(apply(rmultinom(n=10000,size=556,prob=c(6,2,2,1)), 2, chisqteststat)>=qchisq(.95,3))/10000
```

```
## [1] 0.6159
```

If the alternative is a multinomial with equal probabilities, this test will have a high power. If we change the probabilities only slightly from 0.5625, 0.1875, 0.1875, 0.0625 to 0.375, 0.125, 0.125, 0.0625, the power drops quickly. This test has low power.

Problem 5.28 First let's examine the two data sets

```r
names(fusion1)
```

```
##  [1] "id"        "marker"    "markerID" "allele1"  "allele2"  "genotype"
##  [7] "Adose"     "Cdose"     "Gdose"    "Tdose"
```

```r
names(pheno)
```

```
##  [1] "id"     "t2d"    "bmi"    "sex"    "age"    "smoker" "chol"
##  [8] "waist"  "weight" "height" "whr"    "sbp"    "dbp"
```

```r
dim(pheno)
```

```
## [1] 2333   13
```

```r
str(pheno)
```

```
## 'data.frame':    2333 obs. of  13 variables:
##  $ id    : int  1002 1009 1012 1015 1018 1023 1032 1036 1043 1048 ...
##  $ t2d   : Factor w/ 2 levels "case","control": 1 1 2 1 2 1 1 1 1 1 ...
##  $ bmi   : num  32.9 27.4 30.5 32.5 28.3 ...
##  $ sex   : Factor w/ 2 levels "F","M": 1 1 2 2 1 1 1 1 2 2 ...
##  $ age   : num  70.8 53.9 53.9 66.3 53.9 ...
##  $ smoker: Factor w/ 4 levels "former","never",..: 1 2 1 1 4 2 2 2 1 1 ...
##  $ chol  : num  4.57 7.32 5.02 6.42 4.3 6.23 5.03 5.07 6.46 7.14 ...
##  $ waist : num  112 93.5 104 120 84 ...
##  $ weight: num  85.6 77.4 94.6 100.1 75.2 ...
##  $ height: num  161 168 176 175 163 ...
##  $ whr   : num  0.987 0.94 0.933 0.98 0.832 ...
##  $ sbp   : num  135 158 143 155 149 135 134 142 149 147 ...
##  $ dbp   : num  77 88 89 88 89 83 91 90 91 91 ...
```

```r
str(fusion1)
```

```
## 'data.frame':    2331 obs. of  10 variables:
##  $ id      : int  9735 10158 9380 9691 10050 4794 10520 9872 9838 9659 ...
##  $ marker  : Factor w/ 1 level "RS12255372": 1 1 1 1 1 1 1 1 1 1 ...
##  $ markerID: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ allele1 : int  3 3 3 3 3 3 3 3 3 3 ...
```

```
##  $ allele2 : int  3 3 4 3 3 3 3 3 3 4 ...
##  $ genotype: Factor w/ 3 levels "GG","GT","TT": 1 1 2 1 1 1 1 1 1 2 ...
##  $ Adose   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cdose   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gdose   : int  2 2 1 2 2 2 2 2 2 1 ...
##  $ Tdose   : int  0 0 1 0 0 0 0 0 0 1 ...
```

As suggested in the problem, I will merge the data sets

```
fusionlm=merge(fusion1,pheno,by="id",all.x=F,ally=F)
```

To check, I will run the two cross tabulations

```
xtabs(~t2d+genotype,fusionlm)
```

```
##          genotype
## t2d        GG  GT  TT
##   case    737 375  48
##   control 835 309  27
```

```
xtabs(~t2d+Gdose,fusionlm)
```

```
##          Gdose
## t2d         0   1   2
##   case     48 375 737
##   control  27 309 835
```

Part a.

Looking at the cases, they have higher counts with the T allele so it is the risk allele.

Part b.

$$H_o: \text{SNP and Type II Diabetes are independent}$$
$$H_a: \text{There is a relationship between SNP and Type II diabetes}$$

```
chisq.test(xtabs(~t2d+genotype,fusionlm))
```

```
## 
##  Pearson's Chi-squared test
## 
## data:  xtabs(~t2d + genotype, fusionlm)
## X-squared = 18.306, df = 2, p-value = 0.0001059
```

```
xchisq.test(xtabs(~t2d+genotype,fusionlm))
```

```
## 
##  Pearson's Chi-squared test
## 
## data:  x
```

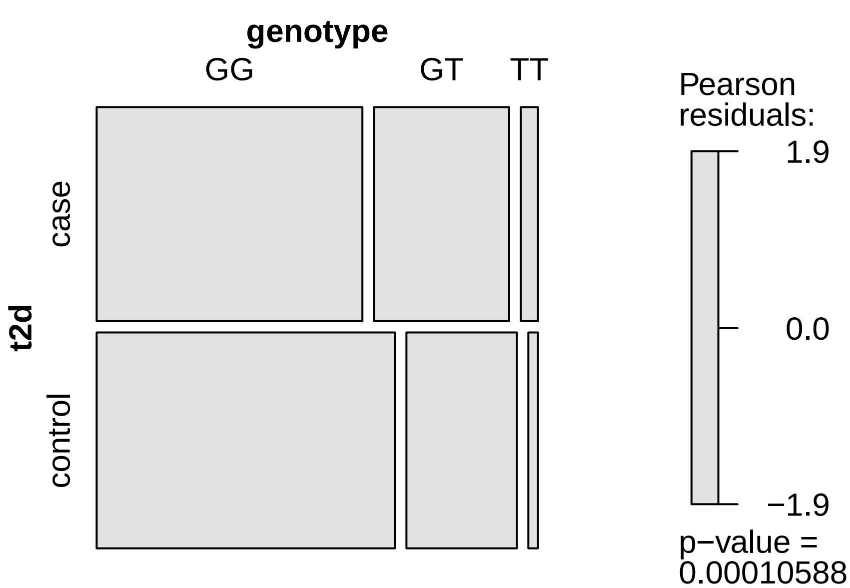
```
## X-squared = 18.306, df = 2, p-value = 0.0001059
##
##    737      375      48
## (782.29) (340.39) ( 37.32)
##   [2.62]   [3.52]   [3.05]
## <-1.62>  < 1.88>  < 1.75>
##
##    835      309      27
## (789.71) (343.61) ( 37.68)
##   [2.60]   [3.49]   [3.03]
## < 1.61>  <-1.87>  <-1.74>
##
## key:
##   observed
##   (expected)
##   [contribution to X-squared]
##   <residual>
```

Based on the p-value, we reject the null and conclude there is a relationship between genotype and type II diabetes.

Here is a plot of the data

```r
library(vcd)
```

```r
mosaic(~t2d+genotype,fusionlm,shade=T)
```



Part c.

Since there are over 300,000 other SNPs, we have to worry about the higher number of Type I errors that could occur. Using an $\alpha = 0.05$ we would expect to find $300000 * .05 = 15000$ SNPs that we think there is an association when in fact there was none. Using the Bonferroni adjustment, we would not reject unless the p-value were less than 0.05/300000. In this problem we would fail to reject. There are less conservative measures to account for the multiple comparison problem but are not discussed in the book.

Chapter 6

Section 6.2

Problem 6.2 Let $\vec{Y} =< Y_1, Y_2, Y_3 >$ and let $\vec{v} =< 1, 2, -3 >$. Suppose $Y_i \sim_{iid} \text{Norm}(5, 2)$.

Part a.
what is the length of $\vec{v}$?

$$|v| = \sqrt{1^2 + 2^2 + (-3)^2} = \sqrt{14}$$

Part b.

Is $\vec{v} \perp \vec{1}$?

$$\vec{v} \cdot \vec{1} = 1 * 1 + 1 * 2 + 1 * (-3) = 3 - 3 = 0$$

Yes the two vectors are orthogonal

Part c.

What is the distribution of $\vec{v} \cdot \vec{Y}$?

$$\vec{v} \cdot \vec{Y} = Y_1 + 2Y_2 - 3Y_3$$

But each Y_i is normal so sum of normals is normal. The expected value is

$$E(\vec{v} \cdot \vec{Y}) = E(Y_i)(1 + 2 - 3) = 0$$

and the variance is, using independence,

$$V(\vec{v} \cdot \vec{Y}) = Var(Y_i)(1^2 + 2^2 + (-3)^2) = 4 * (1 + 4 + 9) = 4 * 14$$

The standard deviation is

$$2 * \sqrt{14}$$

so

$$\vec{v} \cdot \vec{Y} \sim \text{Norm}(0, 2 * \sqrt{14})$$

Problem 6.4 Part a. Let's look at the two data sets:

```
library(fastR)
library(DAAG)
library(Hmisc)
library(lattice)
library(MASS)
```

```
describe(rubberband)
```

```
## rubberband
##
##  2  Variables      16  Observations
## ---------------------------------------------------------------------------
## Stretch
##       n missing  unique    Info    Mean
##      16       0       4    0.94     3.5
##
## 2 (4, 25%), 3 (4, 25%), 4 (4, 25%), 5 (4, 25%)
## ---------------------------------------------------------------------------
## Distance
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##      16       0      16       1   286.3   196.5   204.5   240.0   285.0
##     .75     .90     .95
##   342.2   359.0   371.2
##
##           186 200 209 216 248 263 267 273 297 303 331 340 349 350 368 381
## Frequency   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## %           6   6   6   6   6   6   6   6   6   6   6   6   6   6   6   6
## ---------------------------------------------------------------------------
```

```r
describe(elasticband)
```

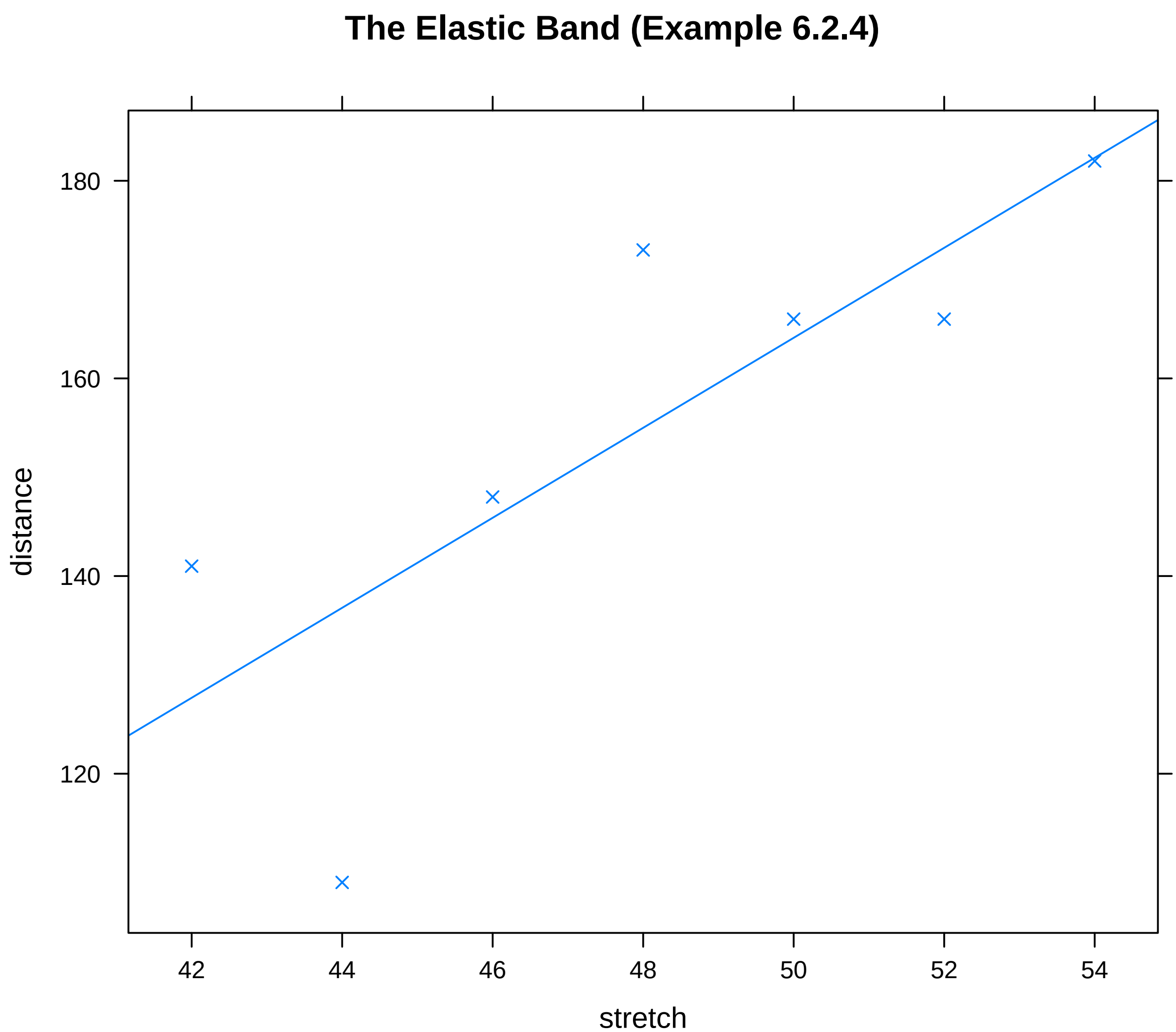
```
## elasticband
##
##  2  Variables      7  Observations
## ---------------------------------------------------------------------------
## stretch
##       n missing  unique    Info    Mean
##       7       0       7       1      48
##
##           42 44 46 48 50 52 54
## Frequency  1  1  1  1  1  1  1
## %         14 14 14 14 14 14 14
## ---------------------------------------------------------------------------
## distance
##       n missing  unique    Info    Mean
##       7       0       6    0.98     155
##
##           109 141 148 166 173 182
## Frequency   1   1   1   2   1   1
## %          14  14  14  29  14  14
## ---------------------------------------------------------------------------
```

The rubber band data set has more observations so potentially a less biased estimate of error and lower variance for parameter estimates. In addition, since the predictors are replicated we have a better estimate of error and are less susceptible to extreme points. At each stretch length we have an ability to estimate error and thus reproducibility.

Part b.

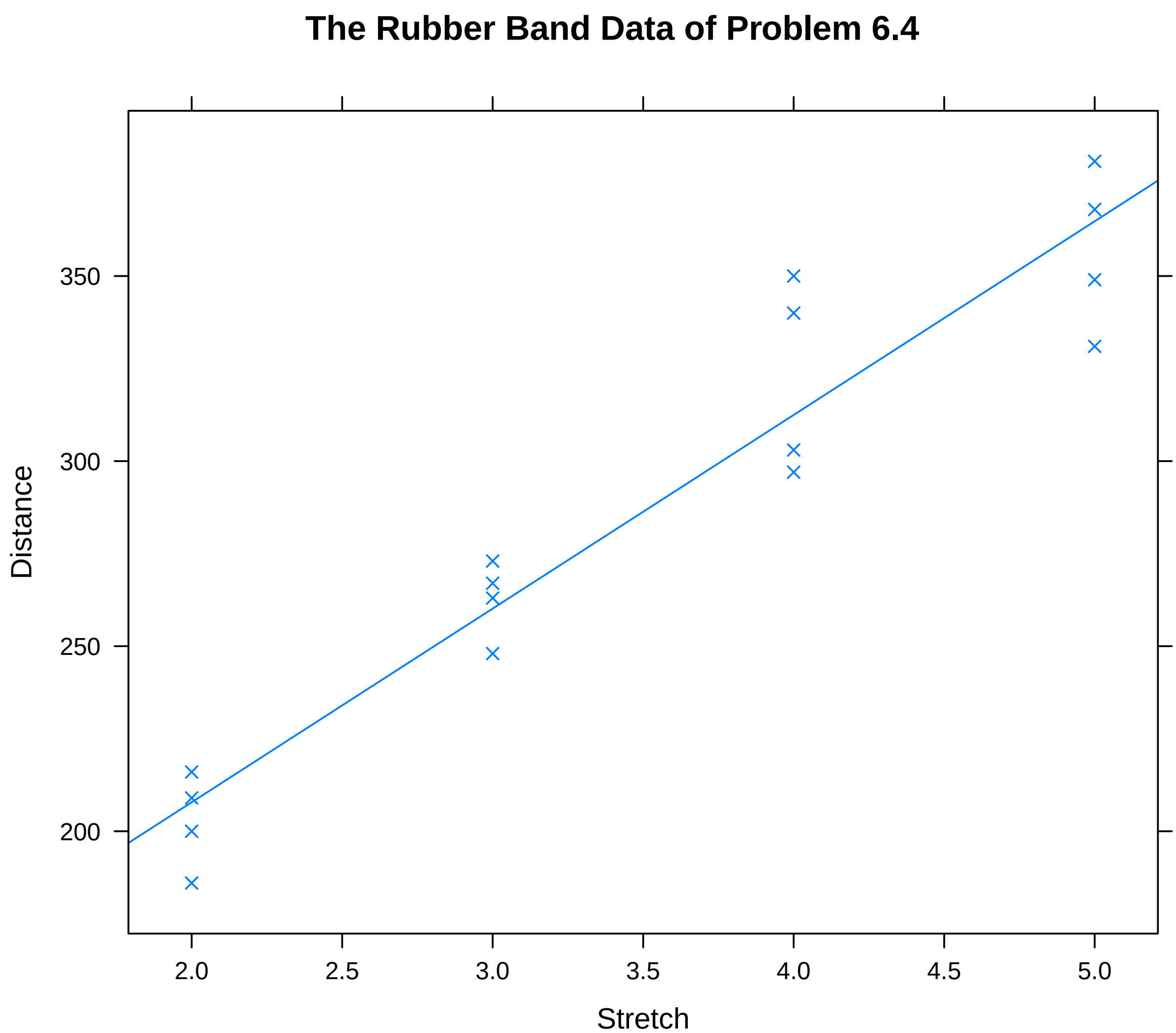
Here are plots of the data:

```
xyplot(distance~stretch,data=elasticband,type=c("p","r"),pch=4,main="The Elastic Band (Example 6.2.4)")
```



and

```
xyplot(Distance~Stretch,data=rubberband,type=c("p","r"),pch=4,main="The Rubber Band Data of Problem 6.4
```



Finally the regression models:

```
summary(lm(distance~stretch,data=elasticband))
```

```
##
## Call:
## lm(formula = distance ~ stretch, data = elasticband)
##
## Residuals:
##        1        2        3        4        5        6        7
##   2.1071  -0.3214  18.0000   1.8929 -27.7857  13.3214  -7.2143
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -63.571     74.332  -0.855   0.4315
## stretch        4.554      1.543   2.951   0.0319 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 16.33 on 5 degrees of freedom
## Multiple R-squared:  0.6352, Adjusted R-squared:  0.5622
## F-statistic: 8.706 on 1 and 5 DF,  p-value: 0.03186
```

```r
summary(lm(Distance~Stretch,data=rubberband))
```

```
##
## Call:
## lm(formula = Distance ~ Stretch, data = rubberband)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.800 -12.981   2.013   9.344  37.525
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  103.175     15.732   6.558 1.27e-05 ***
## Stretch       52.325      4.282  12.220 7.40e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.15 on 14 degrees of freedom
## Multiple R-squared:  0.9143, Adjusted R-squared:  0.9082
## F-statistic: 149.3 on 1 and 14 DF,  p-value: 7.401e-09
```

The second model has smaller p-value for the slope coefficient and the fit seems to be stronger based on adjusted R^2.

Problem 6.5 Part a.

Find an estimate for the model parameter using least squares.

The least squares equation is

$$\sum_{i=1}^{n}(y_i - \beta_1 x_i)^2 = S(\beta_1)$$

Finding the least squares estimate implies

$$\frac{\partial S(\beta_1)}{\partial \beta_1} = \sum_{i=1}^{n} 2(y_i - \beta_1 x_i)(-x_i)$$

$$= -2\sum_{i=1}^{n}(y_i x_i - \beta_1 x_i^2) =$$

$$\sum_{i=1}^{n}(y_i x_i) = \beta_1 \sum_{i=1}^{n} x_i^2$$

$$\hat{\beta_1} = \frac{\sum_{i=1}^{n}(y_i x_i)}{\sum_{i=1}^{n} x_i^2}$$

We must check that this is a minimum, I will leave the details to you.

Part b.

Find an estimate for the model parameter using maximum likelihood.

Starting with a distributional assumption of normality

$$\vec{\varepsilon} == \vec{Y} - \beta_1 \vec{X} \sim N(0, \sigma)$$

The likelihood function is

$$L(\beta_1, \sigma : \vec{X}, \vec{Y}) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-(y_i - \beta_1 x_i)^2/2\sigma^2}$$

or the log-likelihood

$$\ell = -\sum_{i=1}^{n} \left[log(\sigma) - 1/2 log(2\pi) - \frac{(y_i - \beta_1 x_i)^2}{2\sigma^2} \right]$$

Next

$$\frac{\partial \ell}{\partial \beta_1} = \frac{-1}{\sigma^2} \sum_{i=1}^{n} (y_i - \beta_1 x_i)(-x_i) = 0$$

$$\hat{\beta_1} = \frac{\sum_{i=1}^{n}(y_i x_i)}{\sum_{i=1}^{n} x_i^2}$$

and

$$\frac{\partial \ell}{\partial \sigma} = \sum_{i=1}^{n} \left[\frac{-1}{\sigma} + \frac{1}{\sigma^3}(y_i - \beta_1 x_i)^2 \right] = 0$$

$$\frac{-n}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{n} \left((y_i - \beta_1 x_i)^2 \right) = 0$$

$$\hat{\sigma^2} = \frac{\sum_{i=1}^{n}(y_i - \beta_1 x_i)^2}{n} = \frac{\sum_{i=1}^{n}(e_i)^2}{n}$$

Part c.

Enter the data into R

```
x=c(1,2,3,4)
y=c(2,3,5,6)
```

The maximum likelihood and least squares estimate are

```
sum(x*y)/sum(x*x)
```

```
## [1] 1.566667
```

```
fractions(sum(x*y)/sum(x*x))
```

```
## [1] 47/30
```

The maximum likelihood estimate for σ is

```
sqrt(sum((y-47/30*x)^2)/length(x))
```

```
## [1] 0.302765
```

with the `lm()` command in R

```r
summary(lm(y~0+x))
```

```
##
## Call:
## lm(formula = y ~ 0 + x)
##
## Residuals:
##       1       2       3       4
##  0.4333 -0.1333  0.3000 -0.2667
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x  1.56667    0.06383   24.55 0.000148 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3496 on 3 degrees of freedom
## Multiple R-squared:  0.995,  Adjusted R-squared:  0.9934
## F-statistic: 602.5 on 1 and 3 DF,  p-value: 0.0001483
```

```r
summary(lm(y~0+x))$sigma*(sqrt(3/4))
```

```
## [1] 0.302765
```

```r
sqrt(sum((summary(lm(y~0+x))$residuals)^2)/4)
```

```
## [1] 0.302765
```

```r
anova(lm(y~0+x))
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          1 73.633  73.633  602.45 0.0001483 ***
## Residuals  3  0.367   0.122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Problem 6.6 Equation 6.9

$$\frac{\sum(y_i - \bar{y})x_i}{\sum x_i(x_i - \bar{x})}$$

and Equation 6.10

$$\frac{\sum(y_i - \bar{y})(x_i - \bar{x})}{\sum(x_i - \bar{x})^2}$$

Start with Equation 6.10 and expand numerator and denominator

$$\frac{\sum(y_i - \bar{y})x_i - \sum(y_i - \bar{y})\bar{x}}{\sum(x_i^2 - 2x_i\bar{x} + \bar{x}^2)}$$

Looking only at the second part of the numerator

$$-\sum(y_i-\bar{y})\bar{x} = -\sum y_i\bar{x} + \sum \bar{y}\bar{x}$$
$$= -n\bar{y}\bar{x} + n\bar{y}\bar{x} = 0$$

Thus the numerator is

$$\sum(y_i-\bar{y})x_i$$

The denominator is

$$\sum(x_i^2 - 2x_i\bar{x} + \bar{x}^2)$$
$$= \sum x_i(x_i-\bar{x}) + \sum(\bar{x}^2 - x_i\bar{x})$$
$$= \sum x_i(x_i-\bar{x}) + n\bar{x}^2 - \bar{x}\sum x_i$$
$$= \sum x_i(x_i-\bar{x}) + n\bar{x}^2 - \bar{x}n\bar{x}$$
$$= \sum x_i(x_i-\bar{x})$$

Thus Equation 6.10 is

$$\frac{\sum(y_i-\bar{y})x_i}{\sum x_i(x_i-\bar{x})}$$

which is Equation 6.9. They are equivalent.

Section 6.3

Problem 6.7 Since we are only proving part a, we start with

$$\vec{u}\cdot\vec{Y} \sim N(\vec{u}\cdot\vec{\mu}, \sqrt{\vec{u}\cdot\vec{\sigma^2}})$$

where

$$\vec{\sigma^2} = \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \vdots \\ \sigma_n^2 \end{bmatrix}$$

and

$$|\vec{u}| = 1$$

It has been demonstrated before, using moment generating functions, that the linear combination of independent normally distributed random variables is also normally distributed.

In part a, we let all standard deviation be equal. That is

$$\sigma_j = \sigma$$

and

$$\vec{\sigma^2} = \begin{bmatrix} \sigma^2 \\ \sigma^2 \\ \vdots \\ \sigma^2 \end{bmatrix}$$

Now

$$\vec{u}\cdot\vec{\sigma^2} = u_1^2\sigma^2 + u_2^2\sigma^2 + \ldots + u_n^2\sigma^2$$

But $\vec{u}$ is a unit vector, so

$$(u_1^2 + u_2^2 + \ldots + u_n^2) = 1$$
$$\sigma^2(u_1^2 + u_2^2 + \ldots + u_n^2) = \sigma^2$$

Thus

$$\vec{u} \cdot \vec{Y} \sim N(\vec{u} \cdot \vec{\mu}, \sigma)$$

In part b, it is given that

$$\vec{u} \perp \vec{1} \Rightarrow \vec{u} \cdot \vec{1} = u_1 * 1 + u_2 * 1 + \ldots + u_n * 1 = 0$$

We also have

$$\mu_j = \mu$$

Thus

$$\vec{u} \cdot \vec{\mu} = u_1\mu + u_2\mu + \ldots + u_n\mu = \mu(u_1 + u_2 + \ldots + u_n) = 0$$

In this case Thus

$$\vec{u} \cdot \vec{Y} \sim N(0, \sigma)$$

Problem 6.8 Given $\vec{a}, \vec{b},$ and $\vec{c}$ and $\vec{a} \perp \vec{c} \Rightarrow \vec{a} \cdot \vec{c} = 0$, then find

$$\vec{a} \cdot \vec{b}$$

but

$$\vec{a} \cdot \vec{b} = \vec{a} \cdot \vec{b} + \vec{a} \cdot \vec{c}$$

since

$$\vec{a} \cdot \vec{c} = 0$$

Thus

$$\vec{a} \cdot \vec{b} = \vec{a} \cdot (\vec{b} + \vec{c})$$

Also

$$\vec{a} \cdot \vec{b} = \vec{a} \cdot \vec{b} - \vec{a} \cdot \vec{c}$$
$$\vec{a} \cdot \vec{b} = \vec{a} \cdot (\vec{b} - \vec{c})$$

Problem 6.25 An ANOVA table from the R output in problem 6.25

Response: y

Name	Df	Sum Sq	Mean Sq	F	Pr(>F)
x	1	21.876	21.876	2.48	0.132
Residuals	18	158.76	8.82		

Problem 6.26 Part a.

95% CI for mean ACT

```
library(fastR)
```

```
t.test(actgpa$ACT)$conf.int[1:2]
```

```
## [1] 24.24932 27.90453
```

Part b.

95% CI for mean GPA

```
t.test(actgpa$GPA)$conf.int[1:2]
```

```
## [1] 3.185408 3.578515
```

or

```
predict(lm(GPA~ACT,data=actgpa),newdata=data.frame(ACT=mean(actgpa$ACT)),interval="confidence")
```

```
##        fit      lwr      upr
## 1 3.381962 3.263842 3.500081
```

Different because a different estimate of error is being used.

Part c.

The adjusted-R^2 is

```
summary(lm(GPA~ACT,data=actgpa))$r.squared
```

```
## [1] 0.6547641
```

Part d.

Confidence interval for mean GPA with an ACT of 25

```
predict(lm(GPA~ACT,data=actgpa),newdata=data.frame(ACT=25),interval="confidence")
```

```
##        fit      lwr      upr
## 1 3.288243 3.166694 3.409792
```

Part e.

Prediction interval for GPA with an ACT of 30

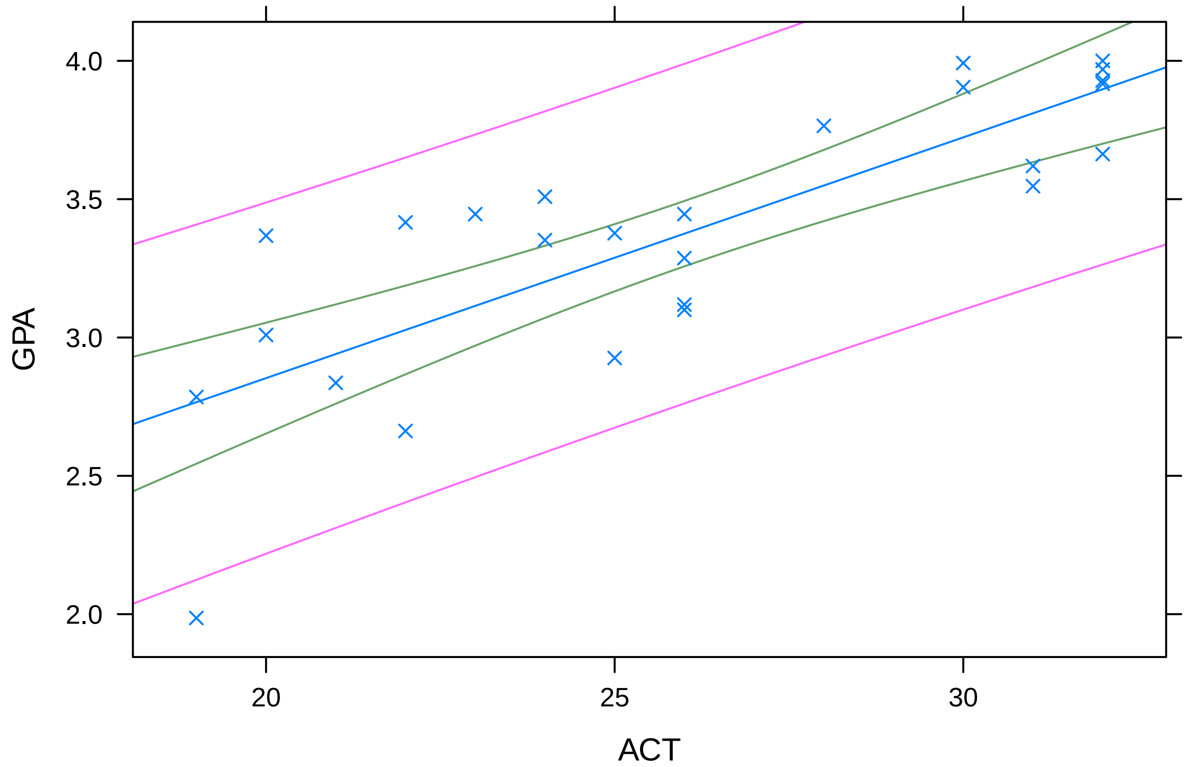
```
predict(lm(GPA~ACT,data=actgpa),newdata=data.frame(ACT=30),interval="prediction")
```

```
##        fit      lwr      upr
## 1 3.723365 3.100775 4.345955
```

Part f.

Concerns

```
xyplot(GPA~ACT,data=actgpa,panel=panel.lmbands,pch=4)
```



There is a significant amount of unexplained variance, due to other factors besides ACT scores. Thus many prediction intervals will be outside of the possible range of gpa, such as exceeding 4. We need to use more predictors.

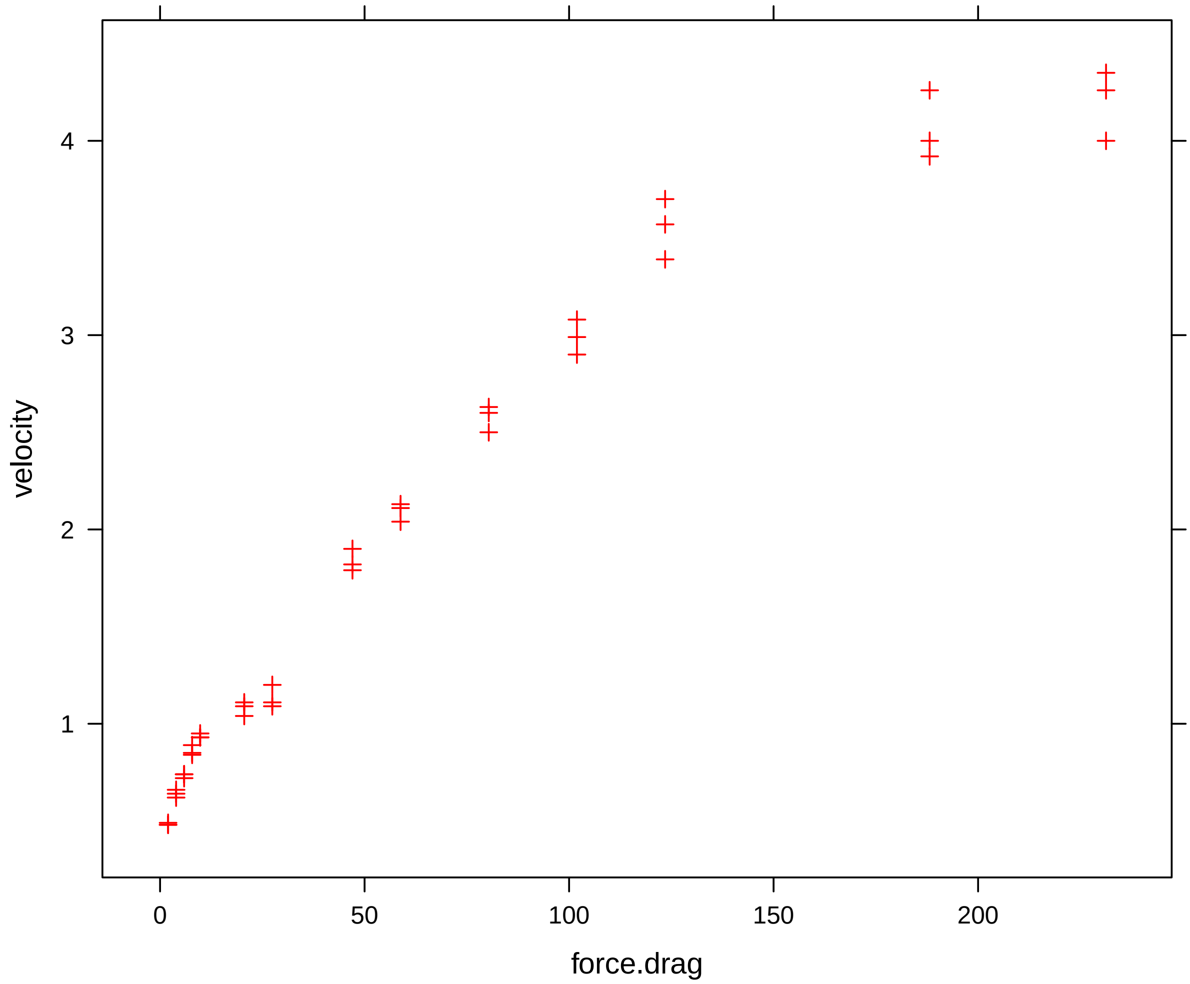
Section 6.4

Problem 6.27 Add fastR library

```
library(fastR)
```

Plot drag versus velocity. We plot velocity on the y-axis since this experiment controlled drag and measured velocity:

```
xyplot(velocity~force.drag,data=drag,col="red",pch=3)
```



Now the models

```
model6.27a=lm(velocity~force.drag,drag)
summary(model6.27a)
```

```
##
## Call:
## lm(formula = velocity ~ force.drag, data = drag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85389 -0.18618 -0.06583  0.27413  0.73302
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.8056051  0.0704308   11.44 3.49e-14 ***
## force.drag  0.0175038  0.0007365   23.77  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.3353 on 40 degrees of freedom
## Multiple R-squared:  0.9339, Adjusted R-squared:  0.9322
## F-statistic: 564.9 on 1 and 40 DF,  p-value: < 2.2e-16
```

```
model6.27b=lm(velocity~sqrt(force.drag),drag)
summary(model6.27b)
```

```
##
## Call:
## lm(formula = velocity ~ sqrt(force.drag), data = drag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39839 -0.11834  0.05261  0.09688  0.50245
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.035856   0.054832  -0.654    0.517
## sqrt(force.drag)  0.290979   0.006807  42.748   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1908 on 40 degrees of freedom
## Multiple R-squared:  0.9786, Adjusted R-squared:  0.978
## F-statistic:  1827 on 1 and 40 DF,  p-value: < 2.2e-16
```

```
model6.27c=lm(velocity~I(force.drag^2),drag)
summary(model6.27c)
```

```
##
## Call:
## lm(formula = velocity ~ I(force.drag^2), data = drag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0539 -0.5619 -0.2408  0.5504  1.3303
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.299e+00  1.223e-01   10.63 3.23e-13 ***
## I(force.drag^2) 7.019e-05  6.808e-06   10.31 7.96e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6818 on 40 degrees of freedom
## Multiple R-squared:  0.7266, Adjusted R-squared:  0.7197
## F-statistic: 106.3 on 1 and 40 DF,  p-value: 7.96e-13
```

```
model6.27d=lm(velocity~exp(force.drag),drag)
summary(model6.27d)
```

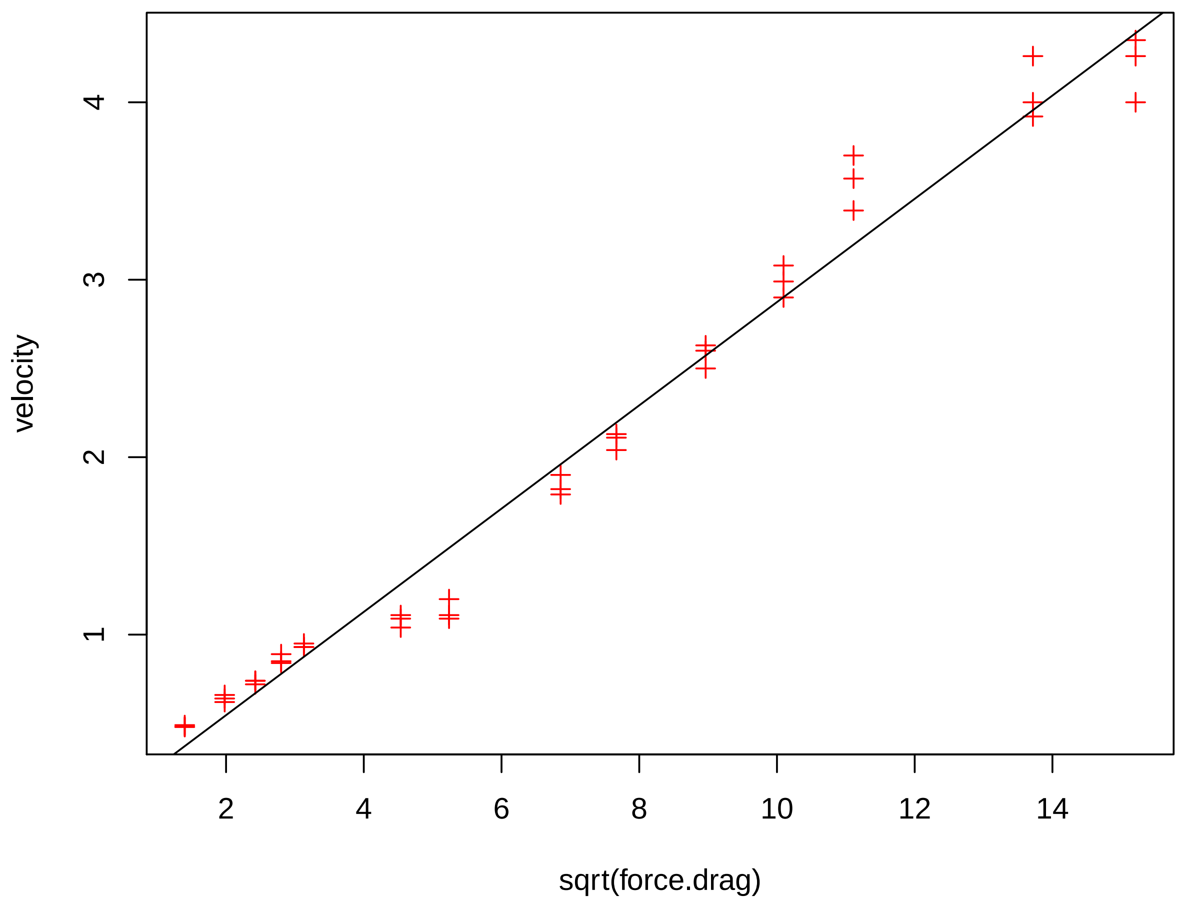
```
##
## Call:
## lm(formula = velocity ~ exp(force.drag), data = drag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2874 -0.9074 -0.3854  0.8076  2.4926
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.767e+00  1.817e-01   9.730 4.23e-12 ***
## exp(force.drag) 8.771e-101 2.447e-101   3.584 0.000909 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.134 on 40 degrees of freedom
## Multiple R-squared:  0.2431, Adjusted R-squared:  0.2241
## F-statistic: 12.84 on 1 and 40 DF,  p-value: 0.0009094
```

```r
model6.27e=lm(velocity~log(force.drag),drag)
summary(model6.27e)
```

```
##
## Call:
## lm(formula = velocity ~ log(force.drag), data = drag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.81947 -0.41079 -0.02638  0.35609  0.79359
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.76888    0.18269  -4.209 0.000141 ***
## log(force.drag)  0.80868    0.04994  16.192  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4744 on 40 degrees of freedom
## Multiple R-squared:  0.8676, Adjusted R-squared:  0.8643
## F-statistic: 262.2 on 1 and 40 DF,  p-value: < 2.2e-16
```

It appears that the second model is the best based on adjusted r^2. Let's plot the data and the line

```r
plot(velocity~sqrt(force.drag),data=drag,col="red",pch=3)
abline(model6.27b)
```



Problem 6.28 Let's do a summary of the data.

```r
summary(drag)
```

```
##       time            mass           height      velocity    
##  Min.   :0.500   Min.   : 0.200   Min.   :1.0   Min.   :0.480  
##  1st Qu.:0.700   1st Qu.: 0.800   1st Qu.:1.0   1st Qu.:0.860  
##  Median :0.900   Median : 3.800   Median :1.5   Median :1.495  
##  Mean   :1.017   Mean   : 6.621   Mean   :1.5   Mean   :1.941  
##  3rd Qu.:1.175   3rd Qu.:10.400   3rd Qu.:2.0   3rd Qu.:2.967  
##  Max.   :2.100   Max.   :23.600   Max.   :2.0   Max.   :4.350  
##    force.drag    
##  Min.   :  1.96  
##  1st Qu.:  7.84  
##  Median : 37.24  
##  Mean   : 64.89  
##  3rd Qu.:101.92  
##  Max.   :231.28
```

```
favstats(velocity~height,data=drag)
```

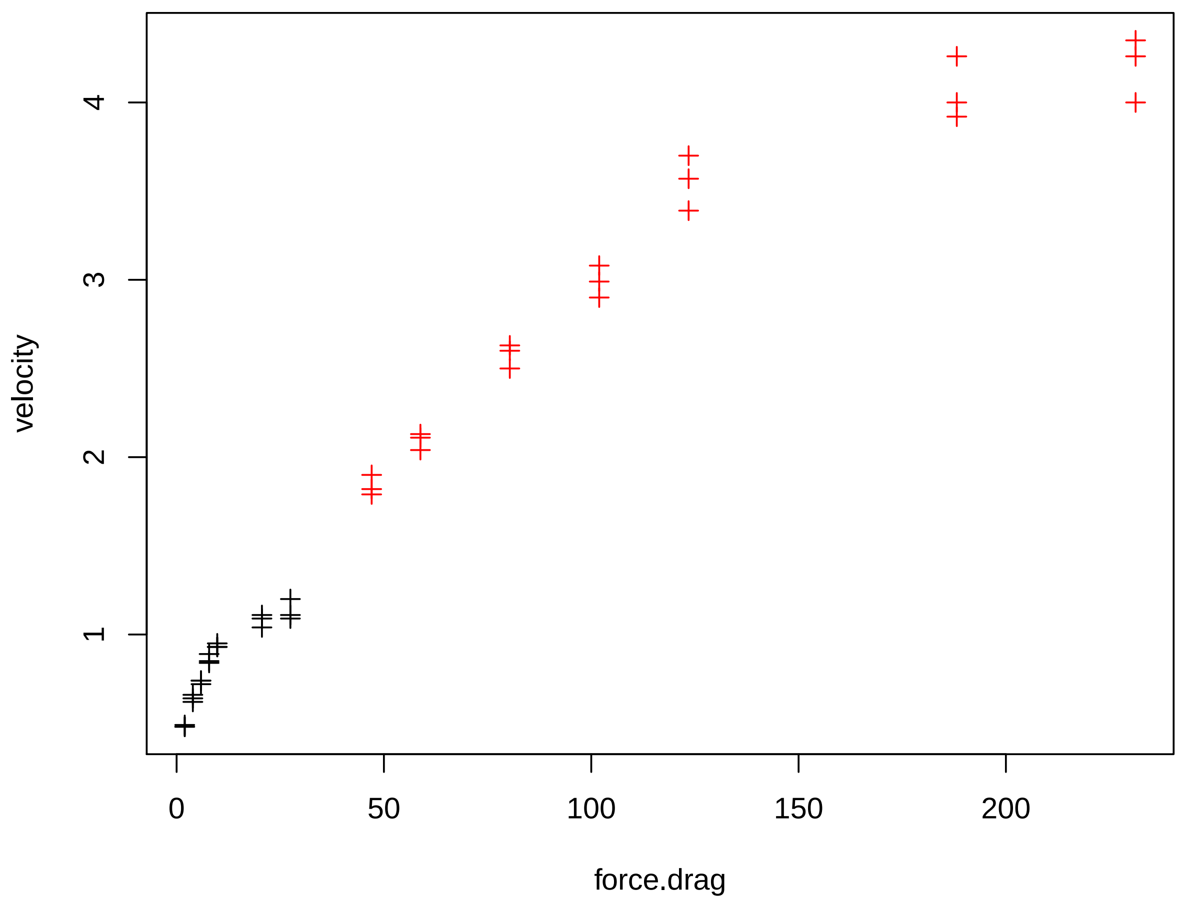
```
##   height  min   Q1 median   Q3  max      mean        sd  n missing
## 1      1 0.48 0.66   0.85 1.04 1.20 0.8380952 0.2245132 21       0
## 2      2 1.79 2.13   2.99 3.92 4.35 3.0447619 0.8907728 21       0
```

```
favstats(force.drag~height,data=drag)
```

```
##   height   min    Q1 median     Q3    max   mean        sd  n missing
## 1      1  1.96  3.92   7.84  20.58  27.44  11.06  8.928993 21       0
## 2      2 47.04 58.80 101.92 188.16 231.28 118.72 64.803337 21       0
```

There appears that when the distance between sensors is 1 meter, there is much less variation. Let's look at a plot.

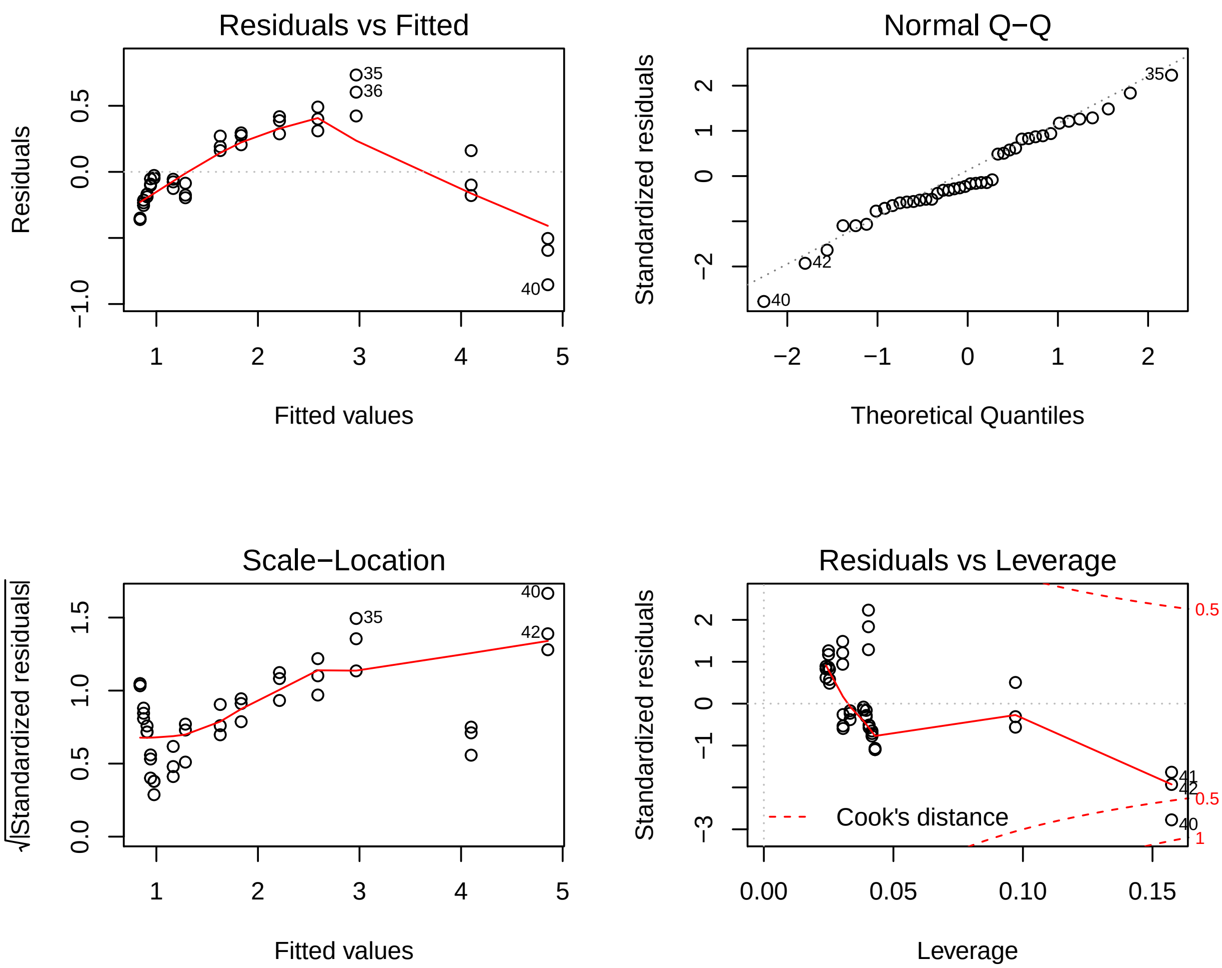
```
plot(velocity~(force.drag),data=drag,col=drag$height,pch=3)
```



The plot confirms that there is not much variability for the lower height. This could be a function of the timing mechanism or a function of the problem itself.

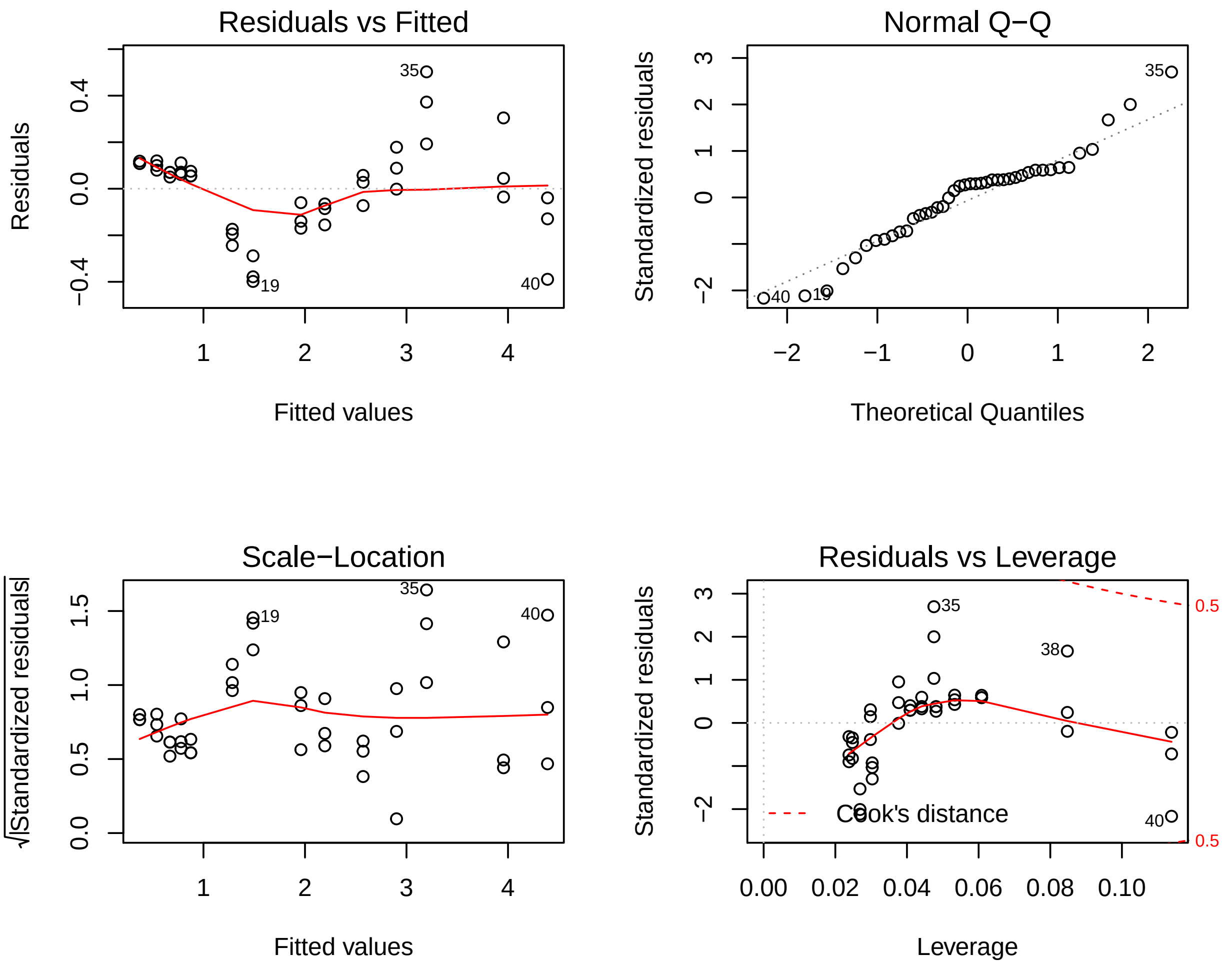
Problem 6.29 Let's look at the diagnostics on the base regression model versus the best model

```
par(mfrow=c(2,2))
plot(model6.27a)
```



Again we see in third plot that the variance increases with drag.

```
par(mfrow=c(2,2))
plot(model6.27b)
```



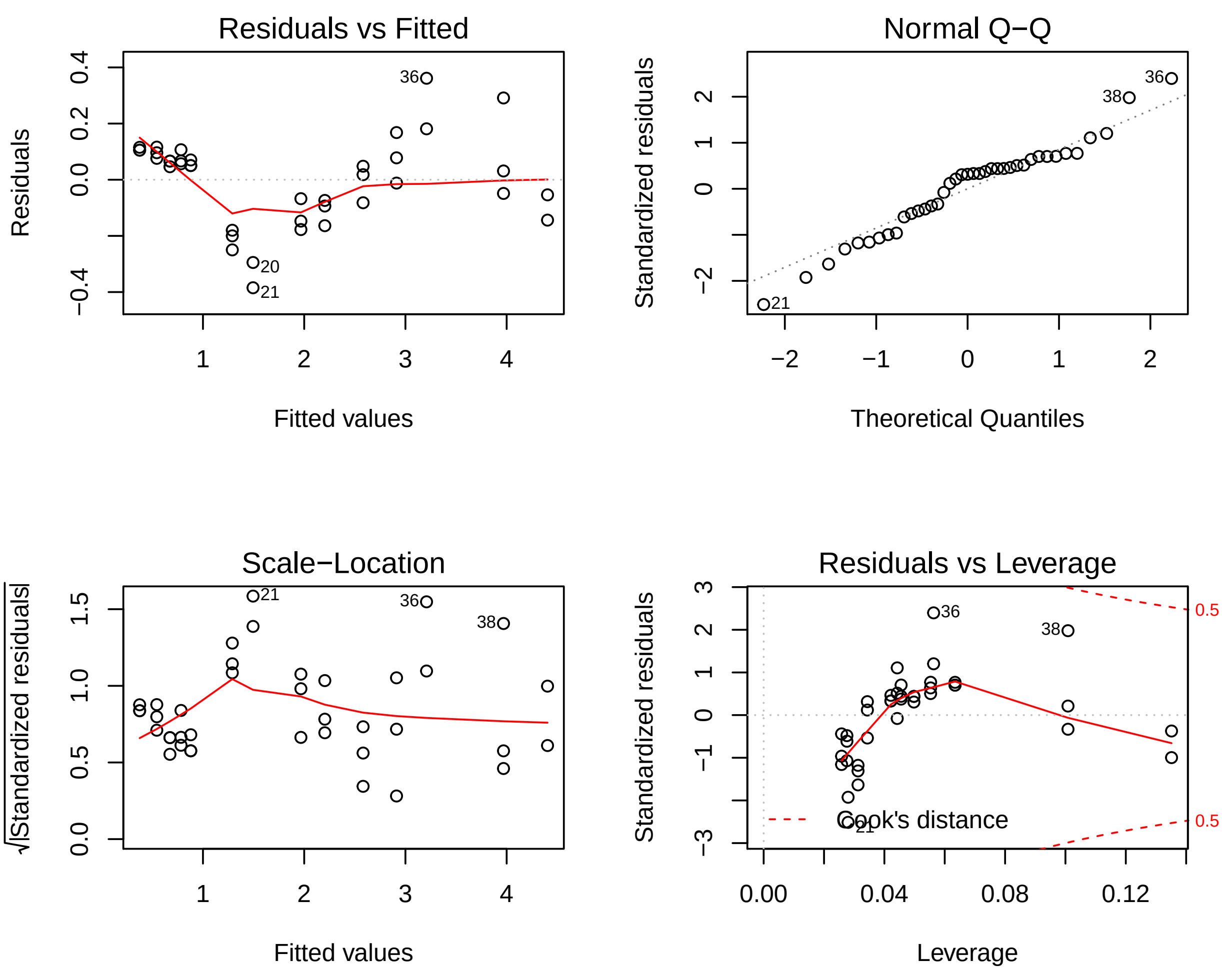
Now let's remove observations 19, 35, and 40.

```
model6.27f=lm(velocity~sqrt(force.drag),drag[c(-19,-35,-40),])
summary(model6.27f)
```

```
##
## Call:
## lm(formula = velocity ~ sqrt(force.drag), data = drag[c(-19,
##     -35, -40), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38466 -0.08780  0.04808  0.08708  0.36135
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.033889   0.045771   -0.74    0.464
## sqrt(force.drag)  0.291801   0.005905   49.42   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1552 on 37 degrees of freedom
## Multiple R-squared:  0.9851, Adjusted R-squared:  0.9847
## F-statistic:  2442 on 1 and 37 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(model6.27f)
```

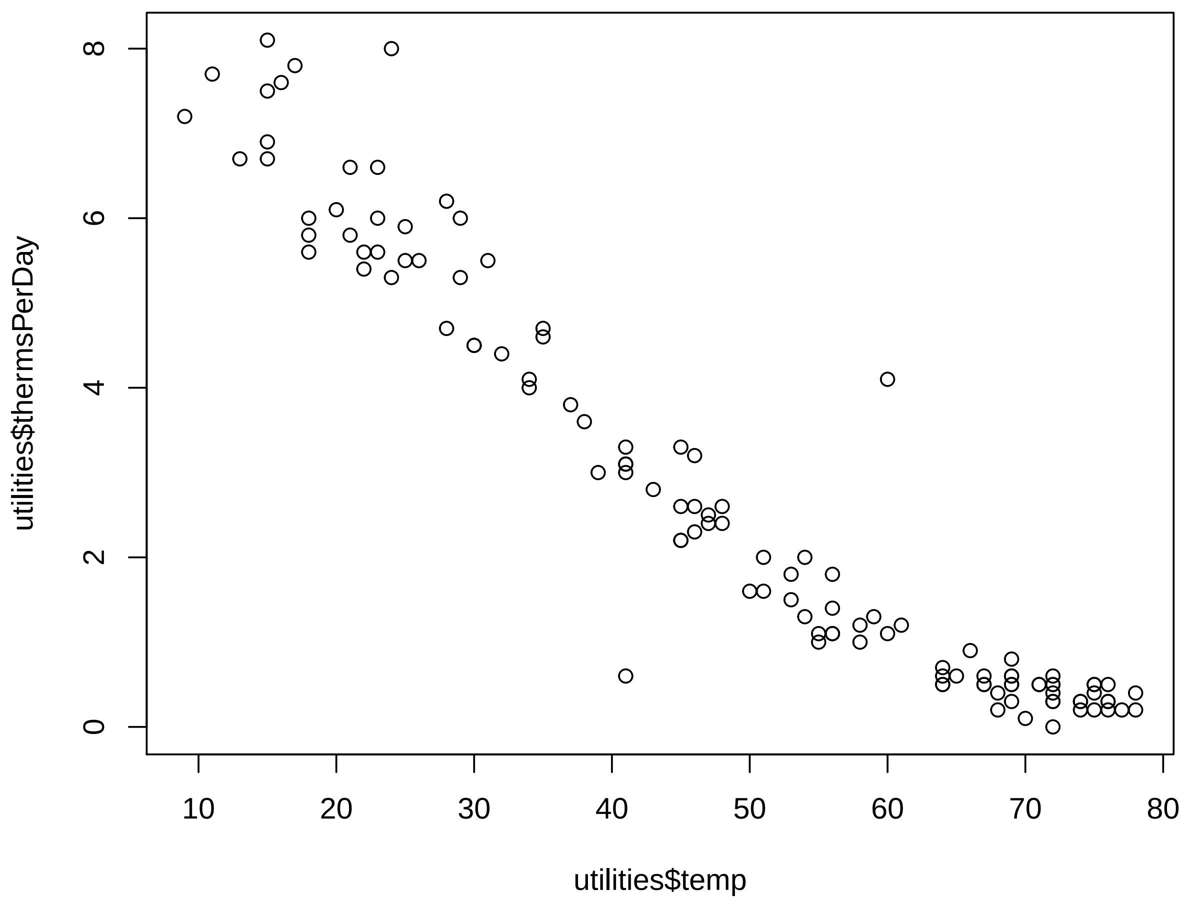


We may still have some problems with a couple of data points but we will stop here.

Problem 6.32 Part a

First plot the data

```
par(mfrow=c(1,1))
plot(utilities$temp,utilities$thermsPerDay)
```



The data points at a temp of around 40 and 60 standout. I will sort the data to look at these two points.

```
utilities[ order(utilities[,4]),c(4,7)]
```

```
##     temp thermsPerDay
## 101    9          7.2
## 13    11          7.7
## 79    13          6.7
## 43    15          8.1
## 54    15          7.5
## 90    15          6.7
## 113   15          6.9
## 44    16          7.6
## 32    17          7.8
## 2     18          5.6
## 89    18          6.0
## 100   18          5.8
## 114   20          6.1
## 22    21          6.6
## 65    21          5.8
```

```
## 67    22           5.6
## 112   22           5.4
## 20    23           6.6
## 53    23           6.0
## 102   23           5.6
## 3     24           8.0
## 78    24           5.3
## 31    25           5.5
## 42    25           5.9
## 1     26           5.5
## 21    28           6.2
## 91    28           4.7
## 33    29           5.3
## 55    29           6.0
## 66    30           4.5
## 77    30           4.5
## 56    31           5.5
## 103   32           4.4
## 30    34           4.1
## 68    34           4.0
## 41    35           4.6
## 45    35           4.7
## 12    37           3.8
## 80    38           3.6
## 99    39           3.0
## 4     41           0.6
## 64    41           3.1
## 76    41           3.1
## 88    41           3.0
## 115   41           3.3
## 52    43           2.8
## 5     45           2.2
## 23    45           3.3
## 92    45           2.6
## 110   45           2.2
## 34    46           3.2
## 81    46           2.6
## 111   46           2.3
## 29    47           2.4
## 104   47           2.5
## 19    48           2.4
## 46    48           2.6
## 75    50           1.6
## 18    51           1.6
## 24    51           2.0
## 40    53           1.5
## 69    53           1.8
## 11    54           1.3
## 57    54           2.0
## 93    55           1.0
## 98    55           1.1
## 35    56           1.4
## 58    56           1.8
## 63    56           1.1
```

```
## 116   56            1.1
## 47    58            1.2
## 87    58            1.0
## 70    59            1.3
## 6     60            4.1
## 117   60            1.1
## 105   61            1.2
## 10    64            0.5
## 17    64            0.7
## 48    64            0.6
## 74    64            0.5
## 82    65            0.6
## 7     66            0.9
## 36    67            0.6
## 50    67            0.5
## 97    67            0.5
## 86    68            0.4
## 94    68            0.2
## 25    69            0.8
## 28    69            0.5
## 39    69            0.5
## 62    69            0.3
## 106   69            0.6
## 109   69            0.6
## 14    70            0.1
## 51    71            0.5
## 107   71            0.5
## 8     72            0.0
## 9     72            0.4
## 27    72            0.5
## 37    72            0.5
## 49    72            0.3
## 59    72            0.6
## 108   72            0.3
## 61    74            0.3
## 71    74            0.3
## 83    74            0.2
## 16    75            0.5
## 38    75            0.5
## 85    75            0.2
## 96    75            0.4
## 15    76            0.2
## 26    76            0.5
## 84    76            0.3
## 95    76            0.3
## 73    77            0.2
## 60    78            0.4
## 72    78            0.2
```

I don't like to remove data. However, observations 4 and 6 tend to be further out. I will run a model with them and one without. Note: it looks like some type of transformation will be in order due to a perceived curvature and there may be a problem with equal variance. We have not discussed how to address these issues.

Part c

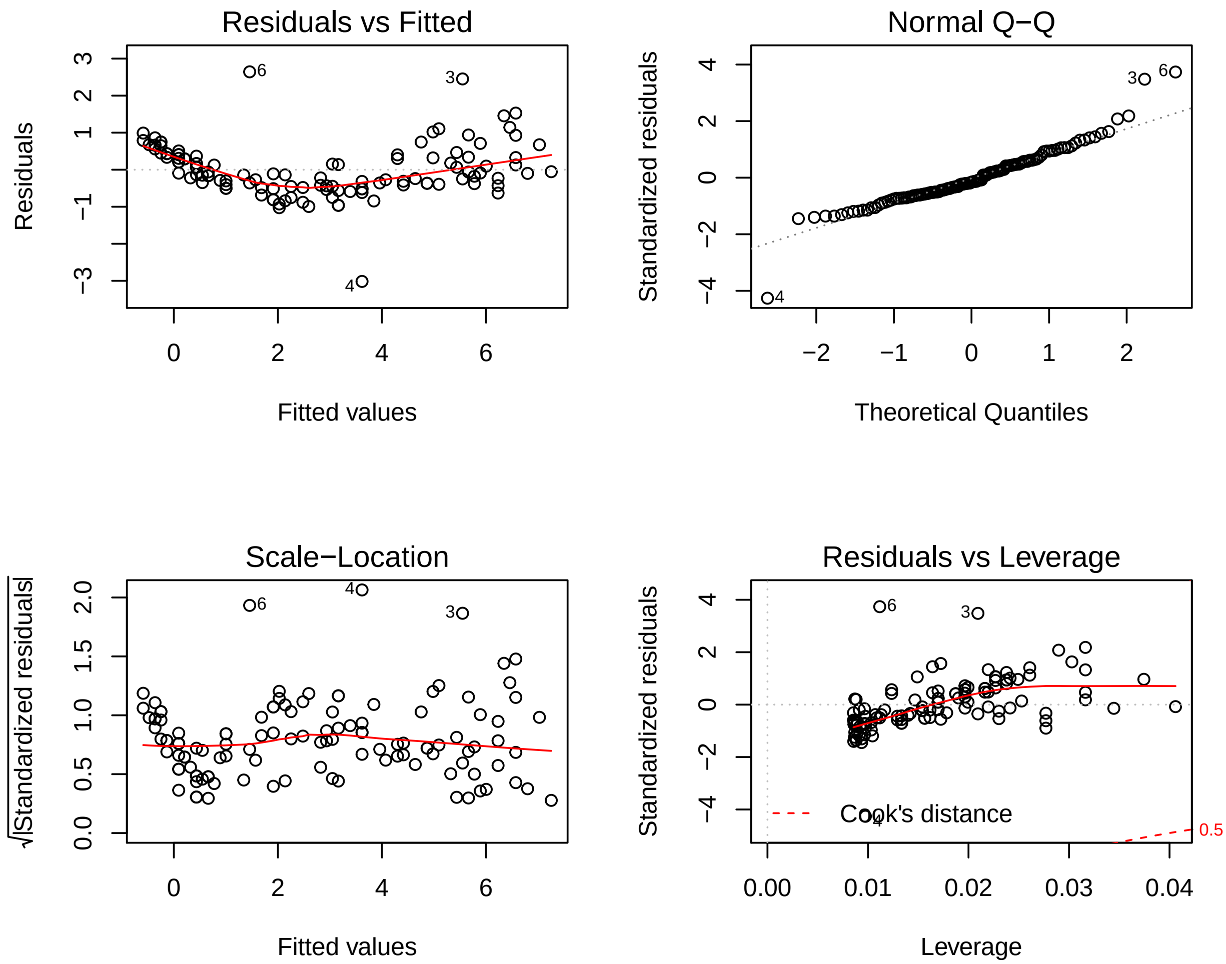
First a model with all the data points.

```
model6.32a=lm(thermsPerDay~temp,data=utilities)
summary(model6.32a)
```

```
##
## Call:
## lm(formula = thermsPerDay ~ temp, data = utilities)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0166 -0.4307 -0.1117  0.4015  2.6429
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.276530   0.169551   48.81   <2e-16 ***
## temp        -0.113658   0.003212  -35.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7115 on 115 degrees of freedom
## Multiple R-squared:  0.9159, Adjusted R-squared:  0.9152
## F-statistic:  1252 on 1 and 115 DF,  p-value: < 2.2e-16
```

and the diagnostic plots:

```
par(mfrow=c(2,2))
plot(model6.32a)
```



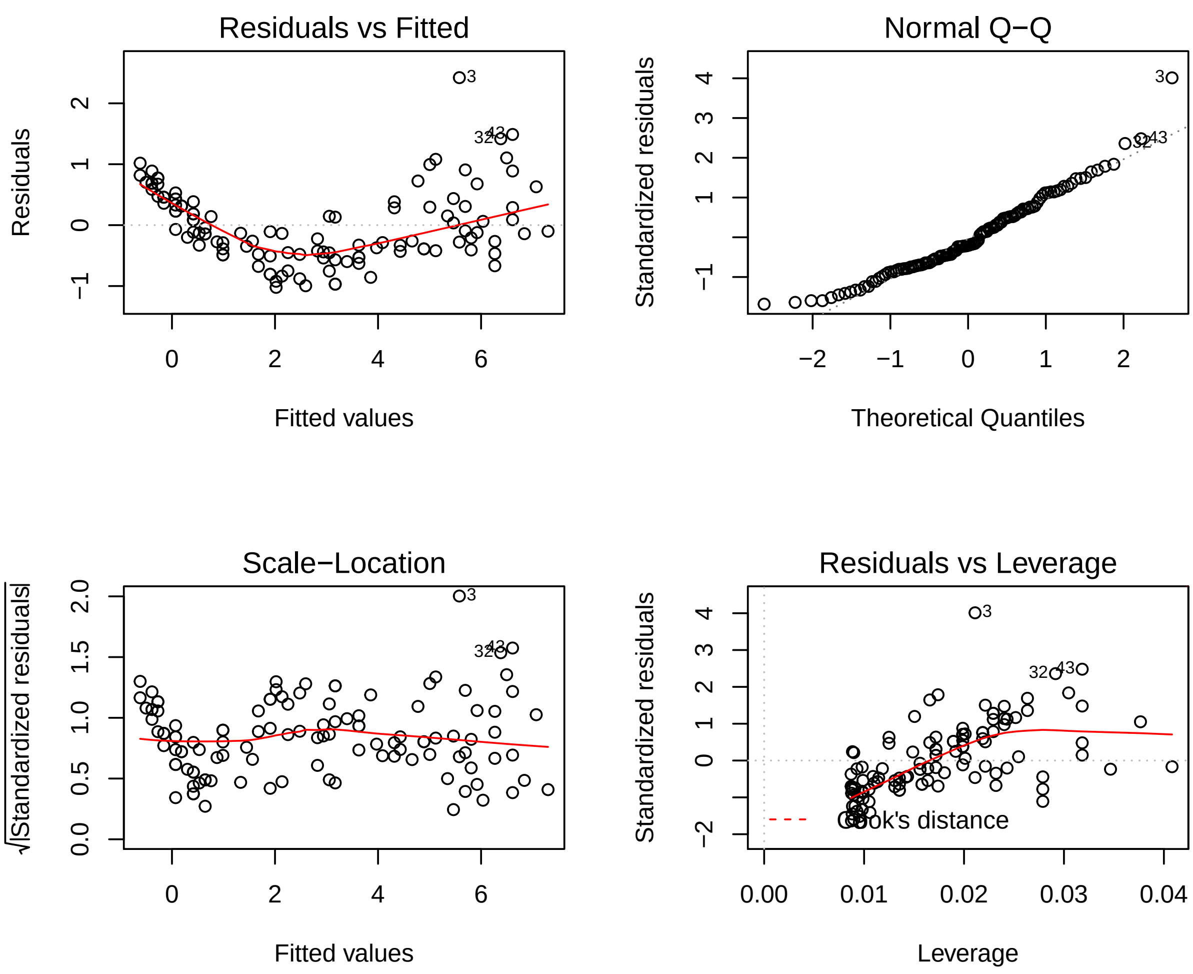
We clearly see that observations 4 and 6 are a problem. It also now appears that observation 3 might also be a problem. The analysis without the two observations, 4 and 6:

```r
model6.32b=lm(thermsPerDay~temp,data=utilities,subset=c(-4,-6))
summary(model6.32b)
```

```
##
## Call:
## lm(formula = thermsPerDay ~ temp, data = utilities, subset = c(-4,
##     -6))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0217 -0.4355 -0.1230  0.3841  2.4213
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.33256    0.14575   57.17   <2e-16 ***
## temp        -0.11474    0.00276  -41.58   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6101 on 113 degrees of freedom
## Multiple R-squared:  0.9387, Adjusted R-squared:  0.9381
## F-statistic:  1729 on 1 and 113 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(model6.32b)
```



Yes, observation 3 appears to be an outlier. In addition, we have curvature and perhaps some heteroscedasticity.

Part d

The interpretation is that the average number of thermal units used per day decrease by -.115 units for every increase of 1 degree in the average daily temperature. Our model explains 93.9% of the variation in thermal units used per day.

Appendix A

Problem A.1 The following code

```
odds<-1+2*(0:4)
```

will generate the first 5 odd numbers and save to an object named odds.
Now checking using R:

```
odds<-1+2*(0:4);odds
```

```
## [1] 1 3 5 7 9
```

The following code

```
primes<-c(2,3,5,7,11,13)
```

will save the first 6 primes into an object named primes
Now checking using R:

```
primes<-c(2,3,5,7,11,13);primes
```

```
## [1]  2  3  5  7 11 13
```

The following code

```
length(odds)
```

will report the number of elements in the object odds, 5.
Now checking using R:

```
length(odds)
```

```
## [1] 5
```

The following code

```
length(primes)
```

will report the number of elements in the object primes, 6.
Now checking using R:

```
length(primes)
```

```
## [1] 6
```

The following code

```
odds+1
```

will add 1 to each element in odds.
Now checking using R:

```
odds+1
```

```
## [1]  2  4  6  8 10
```

The following code

```
odds+primes
```

will add the elements of each object by position. Thus 1 in odds will be added to 2 in primes to return 3. Since primes is one element longer, the elements in odds are used again so that the first element in odds is added to the last element in primes.
Now checking using R:

```
odds+primes
```

```
## Warning in odds + primes: longer object length is not a multiple of shorter
## object length
```

```
## [1]  3  6 10 14 20 14
```

The following code

```
odds*primes
```

will do the same as the previous except multiply.
Now checking using R:

```
odds*primes
```

```
## Warning in odds * primes: longer object length is not a multiple of shorter
## object length
```

```
## [1]  2  9 25 49 99 13
```

The following code

```
odds>5
```

will return a logical vector indicating if the element in odds is greater than 5.
Now checking using R:

```
odds>5
```

```
## [1] FALSE FALSE FALSE  TRUE  TRUE
```

The following code

```
sum(odds>5)
```

will return a the number of elements greater than 5 by adding TRUEs as 1 and FALSEs as zeros.
Now checking using R:

```r
sum(odds>5)
```

```
## [1] 2
```

The following code

```r
sum(primes<5|primes>9)
```

will return a the number of elements in primes less 5 or greater than 9.
Now checking using R:

```r
sum(primes<5|primes>9)
```

```
## [1] 4
```

The following code

```r
odds[3]
```

will return a the third element of odds, 5.
Now checking using R:

```r
odds[3]
```

```
## [1] 5
```

The following code

```r
odds[-3]
```

will return all elements of odds except the third one.
Now checking using R:

```r
odds[-3]
```

```
## [1] 1 3 7 9
```

The following code

```r
primes[odds]
```

will return the first, third, fifth, seventh, and ninth elements of primes. Since primes only has 6 elements, the last two will be NAs.
Now checking using R:

```r
primes[odds]
```

```
## [1]  2  5 11 NA NA
```

The following code

```
primes[primes >=7]
```

will return the elements in primes that are greater than or equal to 7.
Now checking using R:

```
primes[primes >=7]
```

```
## [1]  7 11 13
```

The following code

```
sum(primes[primes > 5])
```

will return the sum of the elements in primes that are greater than 5.
Now checking using R:

```
sum(primes[primes > 5])
```

```
## [1] 31
```

The following code

```
sum(odds[odds > 5])
```

will return the sum of the elements in odds that are greater than 5.
Now checking using R:

```
sum(odds[odds > 5])
```

```
## [1] 16
```

Problem A.2 First look at the data to understand the variables and values

```
library(fastR)
library(Hmisc)
head(ChickWeight,20)
```

```
## Grouped Data: weight ~ Time | Chick
##    weight Time Chick Diet
## 1      42    0     1    1
## 2      51    2     1    1
## 3      59    4     1    1
## 4      64    6     1    1
## 5      76    8     1    1
## 6      93   10     1    1
## 7     106   12     1    1
## 8     125   14     1    1
```

```
## 9      149   16      1     1
## 10     171   18      1     1
## 11     199   20      1     1
## 12     205   21      1     1
## 13      40    0      2     1
## 14      49    2      2     1
## 15      58    4      2     1
## 16      72    6      2     1
## 17      84    8      2     1
## 18     103   10      2     1
## 19     122   12      2     1
## 20     138   14      2     1
```

```
describe(ChickWeight)
```

```
## ChickWeight
##
##  4  Variables      578  Observations
## ---------------------------------------------------------------------------
## weight
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     578       0     212       1   121.8    41.0    47.7    63.0   103.0
##     .75     .90     .95
##   163.8   223.6   264.0
##
## lowest :  35  39  40  41  42, highest: 331 332 341 361 373
## ---------------------------------------------------------------------------
## Time
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     578       0      12    0.99   10.72       0       2       4      10
##     .75     .90     .95
##      16      20      21
##
##            0  2  4  6  8 10 12 14 16 18 20 21
## Frequency 50 50 49 49 49 49 49 48 47 47 46 45
## %          9  9  8  8  8  8  8  8  8  8  8  8
## ---------------------------------------------------------------------------
## Chick
##       n missing  unique
##     578       0      50
##
## lowest : 18 16 15 13 9 , highest: 49 46 50 42 48
## ---------------------------------------------------------------------------
## Diet
##       n missing  unique
##     578       0       4
##
## 1 (220, 38%), 2 (120, 21%), 3 (120, 21%), 4 (118, 20%)
## ---------------------------------------------------------------------------
```

The last time period is 21, subset on only time 21 then sort by weight.

```
A.2=ChickWeight[ChickWeight$Time==21,]
A.2[order(A.2$weight),]
```

```
## Grouped Data: weight ~ Time | Chick
##     weight Time Chick Diet
## 268     74   21    24    2
## 155     96   21    13    1
## 107     98   21     9    1
## 220    117   21    20    1
## 119    124   21    10    1
## 194    142   21    17    1
## 376    147   21    33    3
## 340    150   21    30    2
## 48     157   21     4    1
## 72     157   21     6    1
## 208    157   21    19    1
## 244    167   21    22    2
## 131    175   21    11    1
## 256    175   21    23    2
## 424    178   21    37    3
## 304    192   21    27    2
## 518    196   21    45    4
## 496    200   21    43    4
## 36     202   21     3    1
## 472    204   21    41    4
## 12     205   21     1    1
## 143    205   21    12    1
## 542    205   21    47    4
## 24     215   21     2    1
## 412    220   21    36    3
## 60     223   21     5    1
## 316    233   21    28    2
## 566    237   21    49    4
## 530    238   21    46    4
## 292    251   21    26    2
## 352    256   21    31    3
## 578    264   21    50    4
## 280    265   21    25    2
## 167    266   21    14    1
## 448    272   21    39    3
## 484    281   21    42    4
## 436    290   21    38    3
## 84     305   21     7    1
## 364    305   21    32    3
## 328    309   21    29    2
## 460    321   21    40    3
## 554    322   21    48    4
## 232    331   21    21    2
## 388    341   21    34    3
## 400    373   21    35    3
```

The heaviest chick was 35 with a weight of 373 on diet 3.

The lightest chick was 24 with a weight of 74 on diet 2.

Problem A.3 First we need to find the chick that do not have complete data

```
xtabs(~Time+Chick,data=ChickWeight)
```

```
##     Chick
## Time 18 16 15 13 9 20 10 8 17 19 4 6 11 3 1 12 2 5 14 7 24 30 22 23 27 28
##   0   1  1  1  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   2   1  1  1  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   4   0  1  1  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   6   0  1  1  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   8   0  1  1  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   10  0  1  1  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   12  0  1  1  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   14  0  0  1  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   16  0  0  0  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   18  0  0  0  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   20  0  0  0  1 1  1  1 1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##   21  0  0  0  1 1  1  1 0  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1
##     Chick
## Time 26 25 29 21 33 37 36 31 39 38 32 40 34 35 44 45 43 41 47 49 46 50 42
##   0   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   2   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   4   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   6   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   8   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   10  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   12  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   14  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   16  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   18  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   20  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  1  1  1  1  1  1  1  1
##   21  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  1  1  1  1  1  1  1  1
##     Chick
## Time 48
##   0   1
##   2   1
##   4   1
##   6   1
##   8   1
##   10  1
##   12  1
##   14  1
##   16  1
##   18  1
##   20  1
##   21  1
```

There are 5 chicks with incomplete data, let's find them by finding columns with zeros in it.

```
apply(xtabs(~Time+Chick,data=ChickWeight)==0,2,sum)
```

```
## 18 16 15 13  9 20 10  8 17 19  4  6 11  3  1 12  2  5 14  7 24 30 22 23 27
## 10  5  4  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
## 28 26 25 29 21 33 37 36 31 39 38 32 40 34 35 44 45 43 41 47 49 46 50 42 48
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0
```

```
colnames(xtabs(~Time+Chick,data=ChickWeight))[apply(xtabs(~Time+Chick,data=ChickWeight)==0,2,sum)!=0]
```

```
## [1] "18" "16" "15" "8"  "44"
```

```
chicknum<-c(18,16,15,8,44)
```

Now create a new dataframe without those 5 chicks.

```
A.3<-ChickWeight
for(i in chicknum){
  A.3<-A.3[A.3$Chick!=i,]
}
xtabs(~Time+Chick,data=A.3)
```

```
##     Chick
## Time 13 9 20 10 17 19 4 6 11 3 1 12 2 5 14 7 24 30 22 23 27 28 26 25 29 21
##   0   1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   2   1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   4   1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   6   1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   8   1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   10  1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   12  1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   14  1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   16  1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   18  1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   20  1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##   21  1 1  1  1  1  1 1 1  1 1 1  1 1 1  1 1  1  1  1  1  1  1  1  1  1  1
##     Chick
## Time 33 37 36 31 39 38 32 40 34 35 45 43 41 47 49 46 50 42 48
##   0   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   2   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   4   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   6   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   8   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   10  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   12  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   14  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   16  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   18  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   20  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   21  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

We will now get the absolute amount of weight gained using three different methods for reference.

```
#tapply(ChickWeight$weight,ChickWeight$Chick,FUN=function(x){max(x)-min(x)})
aggregate(weight~Chick,A.3,FUN=function(x){max(x)-min(x)})
```

```
##    Chick weight
## 1     13     55
## 2      9     58
## 3     20     76
## 4     10     83
## 5     17    100
## 6     19    114
## 7      4    118
## 8      6    119
## 9     11    141
## 10     3    163
## 11     1    163
## 12    12    164
## 13     2    175
## 14     5    182
## 15    14    225
## 16     7    264
## 17    24     34
## 18    30    115
## 19    22    126
## 20    23    132
## 21    27    153
## 22    28    194
## 23    26    209
## 24    25    225
## 25    29    270
## 26    21    291
## 27    33    117
## 28    37    137
## 29    36    188
## 30    31    214
## 31    39    230
## 32    38    249
## 33    32    264
## 34    40    280
## 35    34    300
## 36    35    332
## 37    45    156
## 38    43    158
## 39    41    162
## 40    47    169
## 41    49    197
## 42    46    198
## 43    50    223
## 44    42    239
## 45    48    283
```

```
tapply(A.3$weight,A.3$Chick,FUN=function(x){max(x)-min(x)})
```

```
##  13   9  20  10  17  19   4   6  11   3   1  12   2   5  14   7  24  30
##  55  58  76  83 100 114 118 119 141 163 163 164 175 182 225 264  34 115
##  22  23  27  28  26  25  29  21  33  37  36  31  39  38  32  40  34  35
## 126 132 153 194 209 225 270 291 117 137 188 214 230 249 264 280 300 332
##  45  43  41  47  49  46  50  42  48
```

```
## 156 158 162 169 197 198 223 239 283
```

```
chicklevel<-as.numeric(as.character(unique(A.3$Chick)))
abschick=rep(NA,45)
for (i in 1:length(unique(A.3$Chick))){
  abschick[i]<-max(A.3$weight[A.3$Chick==chicklevel[i]])-min(A.3$weight[A.3$Chick==chicklevel[i]])
}
abschick
```

```
##  [1] 163 175 163 118 182 119 264  58  83 141 164  55 225 100 114  76 291
## [18] 126 132  34 225 209 153 194 270 115 214 264 117 300 332 188 137 249
## [35] 230 280 162 239 158 156 198 169 283 197 223
```

Next we will find the percent increase in weight

```
tapply(A.3$weight,A.3$Chick,FUN=function(x){round((x[12]-x[1])/x[1]*100,2)})
```

```
##     13      9     20     10     17     19      4      6     11      3
## 134.15 133.33 185.37 202.44 238.10 265.12 273.81 282.93 306.98 369.77
##      1     12      2      5     14      7     24     30     22     23
## 388.10 400.00 437.50 443.90 548.78 643.90  76.19 257.14 307.32 306.98
##     27     28     26     25     29     21     33     37     36     31
## 392.31 497.44 497.62 562.50 692.31 727.50 276.92 334.15 464.10 509.52
##     39     38     32     40     34     35     45     43     41     47
## 547.62 607.32 643.90 682.93 731.71 809.76 378.05 376.19 385.71 400.00
##     49     46     50     42     48
## 492.50 495.00 543.90 569.05 725.64
```

Now to get plots by weight, we need to aggregate by diet and chick

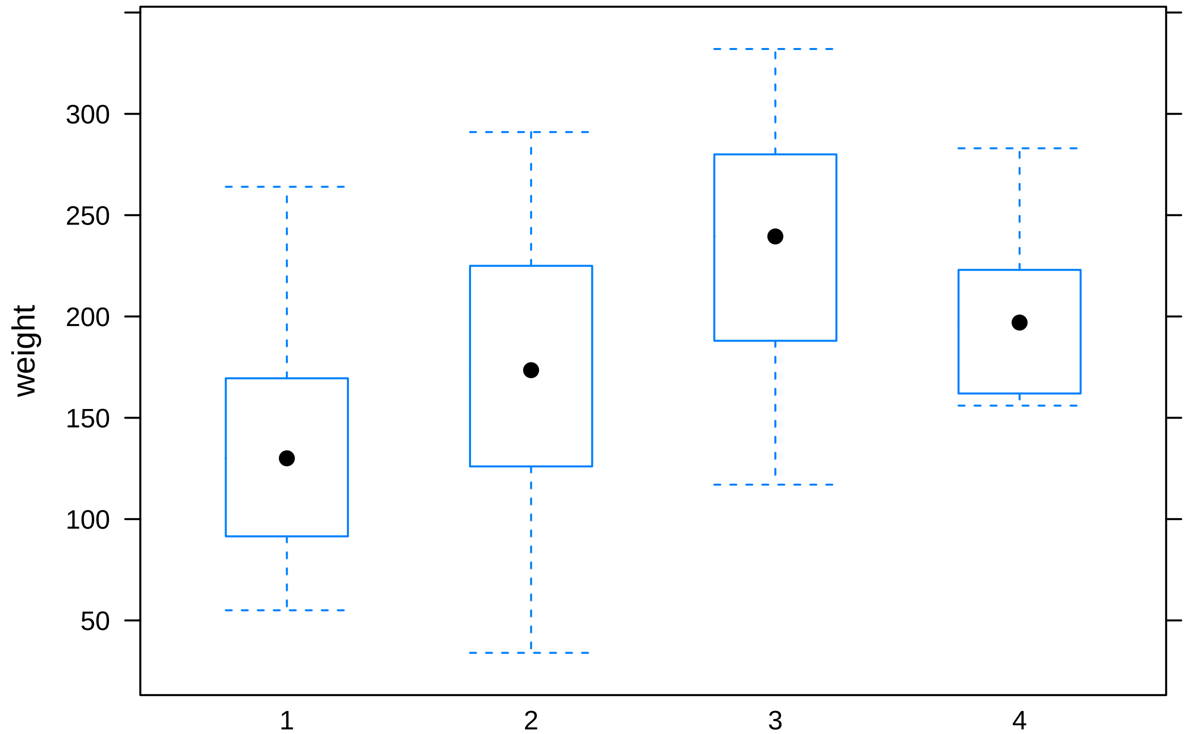
```
A.3sum<-aggregate(weight~Chick+Diet,A.3,FUN=function(x){max(x)-min(x)})
A.3sum
```

```
##    Chick Diet weight
## 1     13    1     55
## 2      9    1     58
## 3     20    1     76
## 4     10    1     83
## 5     17    1    100
## 6     19    1    114
## 7      4    1    118
## 8      6    1    119
## 9     11    1    141
## 10     3    1    163
## 11     1    1    163
## 12    12    1    164
## 13     2    1    175
## 14     5    1    182
## 15    14    1    225
## 16     7    1    264
## 17    24    2     34
## 18    30    2    115
```

```
## 19     22     2     126
## 20     23     2     132
## 21     27     2     153
## 22     28     2     194
## 23     26     2     209
## 24     25     2     225
## 25     29     2     270
## 26     21     2     291
## 27     33     3     117
## 28     37     3     137
## 29     36     3     188
## 30     31     3     214
## 31     39     3     230
## 32     38     3     249
## 33     32     3     264
## 34     40     3     280
## 35     34     3     300
## 36     35     3     332
## 37     45     4     156
## 38     43     4     158
## 39     41     4     162
## 40     47     4     169
## 41     49     4     197
## 42     46     4     198
## 43     50     4     223
## 44     42     4     239
## 45     48     4     283
```

Finally the plots

```
bwplot(weight~Diet,A.3sum)
```




```
summary(weight~Diet,A.3sum)
```

```
## weight    N=45
##
## +-------+-+--+--------+
## |       | |N |weight  |
## +-------+-+--+--------+
## |Diet   |1|16|137.5000|
## |       |2|10|174.9000|
## |       |3|10|231.1000|
## |       |4| 9|198.3333|
## +-------+-+--+--------+
## |Overall| |45|178.7778|
## +-------+-+--+--------+
```

Diet 3 tends to have the largest weight gain but there is also a large amount of variance.

Problem A.4 This is a function to find the third largest value with some error handling.

```
third=function(x,k=3){
if(length(x)<3)return(print("Less than 3 values in vector"))
if(!is.numeric(x))return(print("Data is not numeric"))
sort(x,decreasing=T)[k]
}
```

Now test

```r
third(c("a","b","c"))
```

```
## [1] "Data is not numeric"
```

```r
third(c("a","b"))
```

```
## [1] "Less than 3 values in vector"
```

```r
third(c(1,2,3,4))
```

```
## [1] 2
```

```r
third(c(1,4))
```

```
## [1] "Less than 3 values in vector"
```