Some Notes from the Book: Data Analysis and Graphics Using R: An Example-Based Approach by John Maindonald and W. John Braun

John L. Weatherwax[*]

October 8, 2004

[*]wax@alum.mit.edu

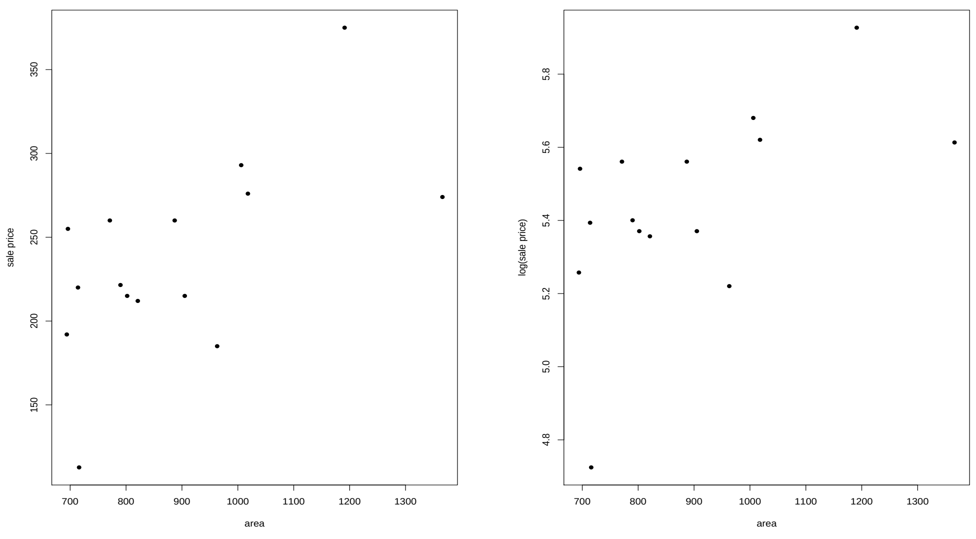


Figure 1: **Left:** Plots of price as a function of area. **Right:** Plots of log(price) as a function of area.

A brief introduction to R

Problem Solutions

Problem 1 (floor area)

See the `R` script `chap_1_prob_1.R`.

Problem 2 (orings)

See the `R` script `chap_1_prob_2.R`.

Problem 3 (`possum` data

Part (a): The command `str` gives the "structure" of an internal `R` expression. For example, the command `str` on the `possum` data gives

```
> library(DAAG)
> str(possum)
'data.frame':   104 obs. of  14 variables:
 $ case    : num  1 2 3 4 5 6 7 8 9 10 ...
```

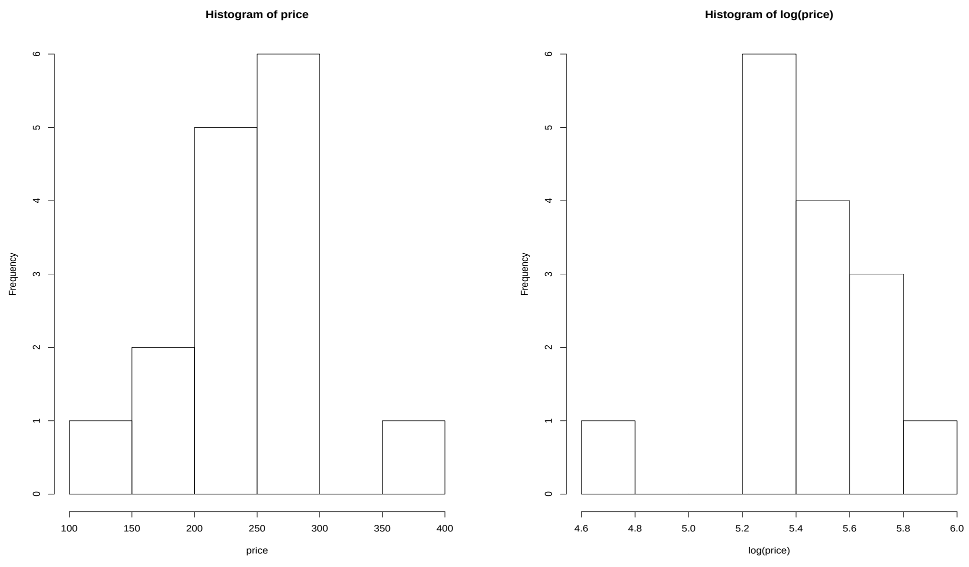


Figure 2: **Left:** Histogram of the price. **Right:** Histogram of the log(price). Notice that the “outlier” in each plot is in a different location. On the left we see the large priced house is the outlier, while on the right the smaller priced house is the outlier.

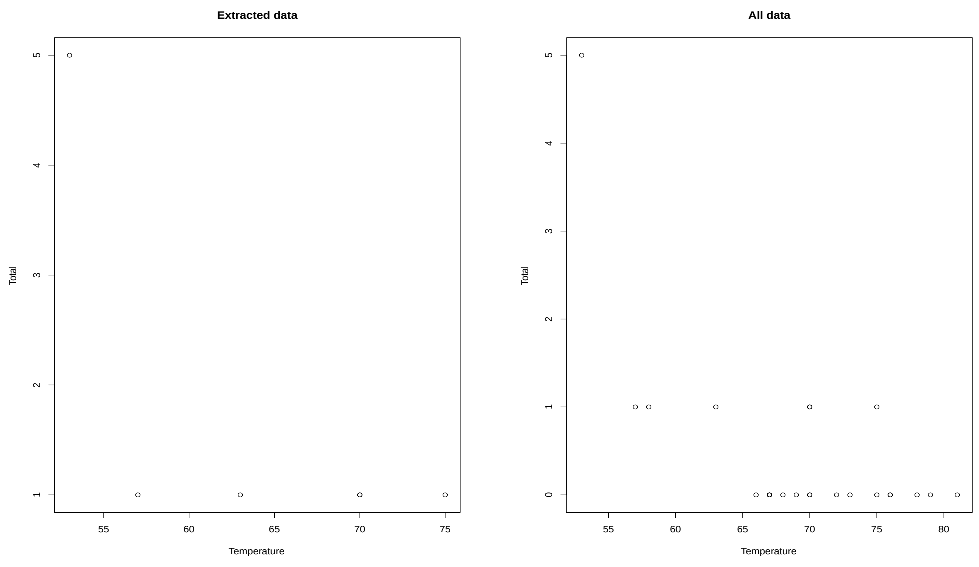


Figure 3: **Left:** Plots of the down sampled oring data **Right:** Plots of the entire oring data. Note that when we look at all of the data we get the impression that the chance of a failures is much larger than when we look at the smaller subset of the data. In fact the chance of failure seems to be nonzero over a much larger temperature range than might be concluded from the smaller data set.

```
 $ site    : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Pop     : Factor w/ 2 levels "Vic","other": 1 1 1 1 1 1 1 1 1 1 ...
 $ sex     : Factor w/ 2 levels "f","m": 2 1 1 1 1 1 2 1 1 1 ...
 $ age     : num  8 6 6 6 2 1 2 6 9 6 ...
 $ hdlngth : num  94.1 92.5 94 93.2 91.5 93.1 95.3 94.8 93.4 91.8 ...
 $ skullw  : num  60.4 57.6 60 57.1 56.3 54.8 58.2 57.6 56.3 58 ...
 $ totlngth: num  89 91.5 95.5 92 85.5 90.5 89.5 91 91.5 89.5 ...
 $ taill   : num  36 36.5 39 38 36 35.5 36 37 37 37.5 ...
 $ footlgth: num  74.5 72.5 75.4 76.1 71 73.2 71.5 72.7 72.4 70.9 ...
 $ earconch: num  54.5 51.2 51.9 52.2 53.2 53.6 52 53.9 52.9 53.4 ...
 $ eye     : num  15.2 16 15.5 15.2 15.1 14.2 14.2 14.5 15.5 14.4 ...
 $ chest   : num  28 28.5 30 28 28.5 30 30 29 28 27.5 ...
 $ belly   : num  36 33 34 34 33 32 34.5 34 33 32 ...
```

Part (b): We have the missing cases given by

```
> missing_cases = which( complete.cases(possum) != TRUE )
> missing_cases
[1] 41 44 46
```

Looking at these cases we have

```
> possum[ missing_cases, ]
     case site Pop sex age hdlngth skullw totlngth taill footlgth earconch  eye
BB36   41    2 Vic   f   5    88.4   57.0       83  36.5       NA     40.3 15.9
BB41   44    2 Vic   m  NA    85.1   51.5       76  35.5     70.3     52.6 14.4
BB45   46    2 Vic   m  NA    91.4   54.4       84  35.0     72.8     51.2 14.4
     chest belly
BB36  27.0  30.5
BB41  23.0  27.0
BB45  24.5  35.0
```

where we see, for example, that the first sample with NA is missing data in the column corresponding to footlgth.

Problem 4 (the data frame ais)

Part (b): We can find this information using

```
> table( sex=ais$sex, sport=ais$sport )
   sport
```

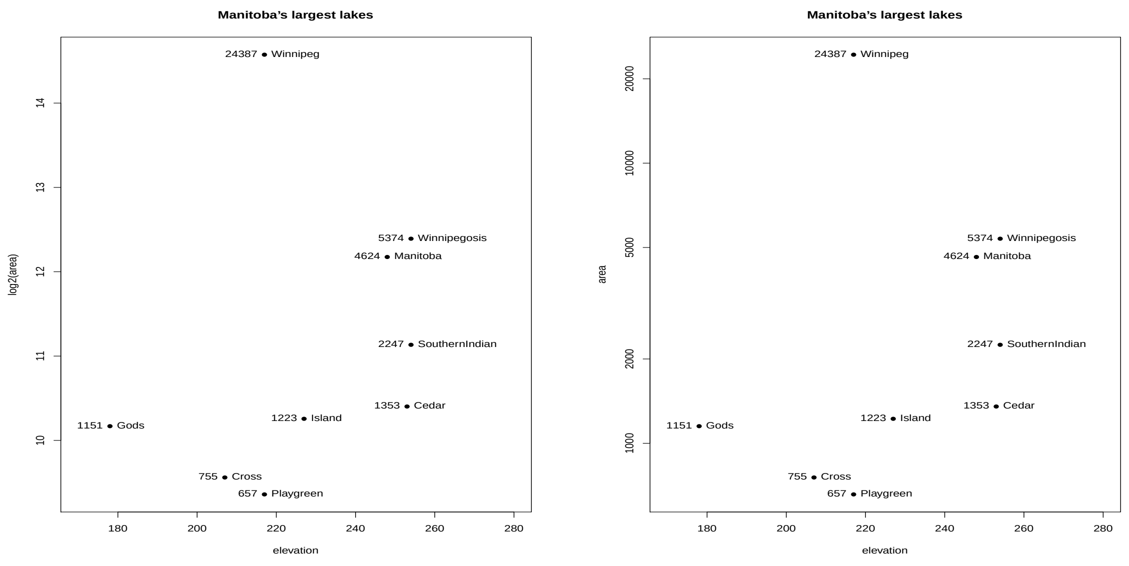


Figure 4: **Left:** Plots of log2(area) as a function of elevation. **Right:** Plots of log2(area) as a function of elevation (using the option `log=y`). These plots are identical, showing that in `R` one can quickly transform the x and y axis using a logarithm.

```
sex B_Ball Field Gym Netball Row Swim T_400m T_Sprnt Tennis W_Polo
  f     13     7   4      23  22    9     11       4      7      0
  m     12    12   0       0  15   13     18      11      4     17
```

We can then look for sports where there are significantly more men or women participants.

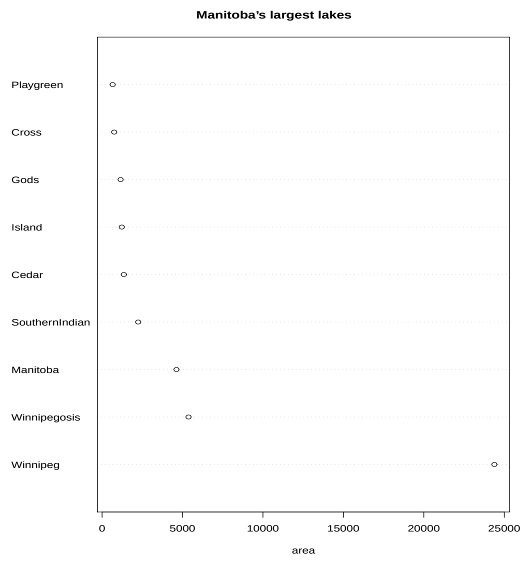
Problem 5 (the data frame `rainforest`)

See the `R` script `chap_1_prob_5.R`. When that script is run we find

```
                Valid Data NA Data Total Samples
Acacia mabellae         10       6            16
C. fraseri              12       0            12
Acmena smithii          11      15            26
B. myrtifolia           10       1            11
```

Problem 6 (the data frame `Manitoba.lakes`)

See the `R` script `chap_1_prob_6.R`. When that script is run produce plots given in Figure 4.



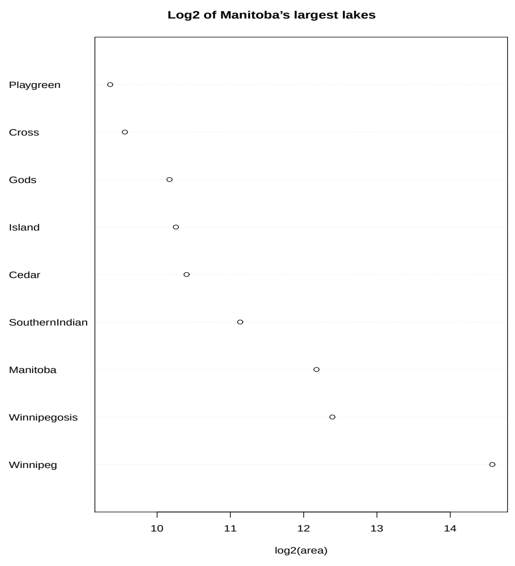


Figure 5: Using the `dotchart` function.

Problem 7 (using `dotchart`)

See the `R` script `chap_1_prob_7.R`. When that script is run produce plots given in Figure 5.

Problem 8 (using `dotchart`)

A lower bound is given by $sum(Manitoba.lakes\$area) = 41771$.

Problem 9 (arguments to `rep`)

Part (a): The command gives

```
> rep( c(2,3,5), 4:2 )
[1] 2 2 2 2 3 3 3 5 5
```

Part (b): Use

```
> rep( c(4,3,2), c(4,4,4) )
 [1] 4 4 4 4 3 3 3 3 2 2 2 2
```

Part (c): Use `rep( c(3,1,1,5,7), length.out=50 )`

Part (d): Use `rep( c(3,1,1,5,7), each=4 )`. As another method is

```
c( rep( 3, length.out=4 ), rep( c(1,1), length.out=8 ), rep( 5, length.out=4 ), rep( 7, length.out=4 ) )
```

Problem 12

See the `R` script `chap_1_prob_12.R`.

Problem 13

See the `R` script `chap_1_prob_13.R`.

Problem 14

See the `R` script `chap_1_prob_14.R`.

Problem 15

See the `R` script `chap_1_prob_15.R`. Which makes the plot given in Figure 6. The number of samples in the largest age groups are very small indicating that they will have a large variance. It looks like the `unclass(age)` version seems to give a bit more information about the very high scores in that we can see the points explicitly (and their relationship to other points).

Problem 16

See the `R` script `chap_1_prob_16.R`.

Problem 17

See the `R` script `chap_1_prob_17.R`. We should use `seq(along=x)` when the input argument `x` could be `NULL`.

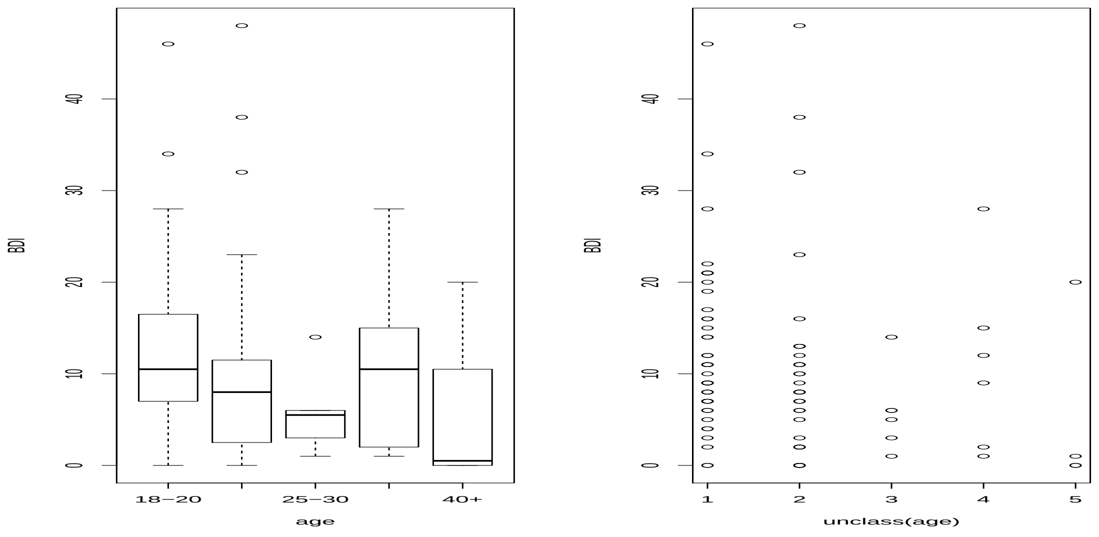


Figure 6: Plots for Problem 15.

Problem 18

See the R script chap_1_prob_18.R.

Problem 19

See the R script chap_1_prob_19.R. When we run that code we get

```
values window1 window2 window3
     0     141     192     200
     1     132      23      99
     2      30      65      11
     3      14      46       3
     5       7      10      14
     6       6       1       6
     7      15       8      12
```

From these window1 looks like it has more higher value prices than the other windows do (it has about 50 less 0 values).

Problem 22

See the R script chap_1_prob_21.R.

Figure 7: Age of possums at each site as a function of sex (male or female).

Styles of data analysis

Problem Solutions

Problem 1

We can do this with the code

```
library(DAAG)
library(lattice)

bwplot( site ~ age | sex, layout=c(2,1), data=possum )
```

What that code is run we get the result in Figure 7.

Problem 2

We can do this with the code

```
library(DAAG)

with( possum, stem(totlngth[ sex=="f" ]) )
with( possum, median( totlngth[ sex=="f" ] ) )
```

Problem 3

See the `R` script `chap_2_prob_3.R`.

Problem 4

We can do this with code like

```
library(DAAG)
library(lattice)

# I think plots like these look good for a first look at the data:
#
densityplot( ~rcc | sport, groups=sex, data=ais )
densityplot( ~hg | sport, groups=sex, data=ais )

# or we could try
#
bwplot( sport ~ rcc | sex, data=ais )
```

Problem 5

See the `R` script `chap_2_prob_5.R`. When we run that script we get the plot shown in Figure 8. The very long line is for the wren host which has smaller eggs than most of the other hosts.

Problem 6

See the `R` script `chap_2_prob_6.R`.

Problem 8

See the `R` script `chap_2_prob_8.R`.

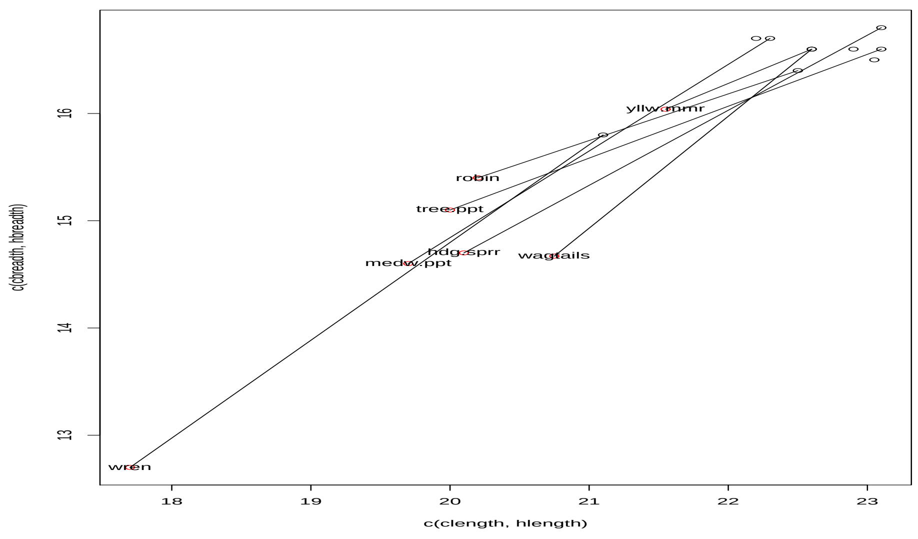


Figure 8: Egg $(length, breadth)$ plots for both cuckoo's (in black) and their host birds (in red). Lines connect the cuckoo's egg with the host bird's egg.

Problem 9 (a pooled estimate of cuckoo egg lengths)

See the `R` script `chap_2_prob_9.R`. We estimate the pooled standard deviation to be 0.9051987.

Problem 10

See the `R` script `chap_2_prob_10.R`. We estimate the three expressions for correlation as $-0.00534, 0.7794935, 0.7162994$. Plots of this data given in the book show that expressing body weight and brain weight directly gives very little explanation to most of the data. Taking the logarithm gives data that has a much more linear looking and only a few points do not fit the general linear trend well. The book suggests using the Spearman rank correlation when the two variables are monotonic but not necessarily *linear* i.e. could have a nonlinear trend. Based on this I would use the normal Pearson correlation but in log space.

Problem 13 (a plot of the density of the `galaxies` dataset

We can do this with the code

```
library(MASS)

plot( density( galaxies ) )
```

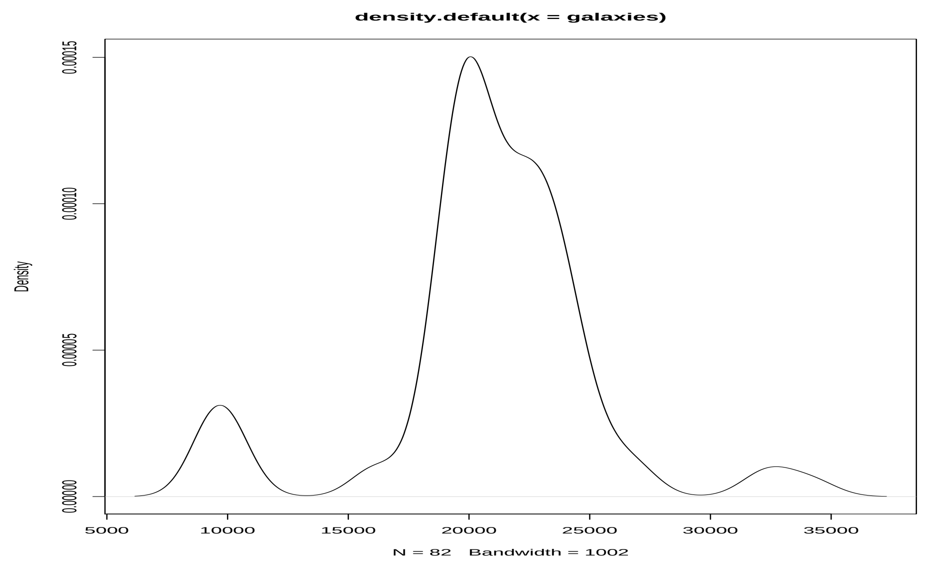


Figure 9: The density plot of the `galaxies` dataset.

When that code is run it produces the plot shown in Figure 9. We see evidence of clustering and a somewhat skewed (to the right) distribution

Problem 14

Part (a): See the `R` script `chap_2_prob_14.R`. Some of the variables have relatively different ranges. For example, we have that

```
> sapply( cpus[,2:9], range )
     syct  mmin  mmax cach chmin chmax perf estperf
[1,]   17    64    64    0     0     0    6      15
[2,] 1500 32000 64000  256    52   176 1150    1238
```

showing that the range of the `mmax` variable goes to 64000. As this is so much larger than the other variables we might want to apply a transformation of this data to map all variables on to the same scale.

Part (b): See Figure 10 where we display the scatter plot of log(perf) $\sim$ log(estperf). It looks like the variability about the best fit line gets smaller as log(perf) decreases.

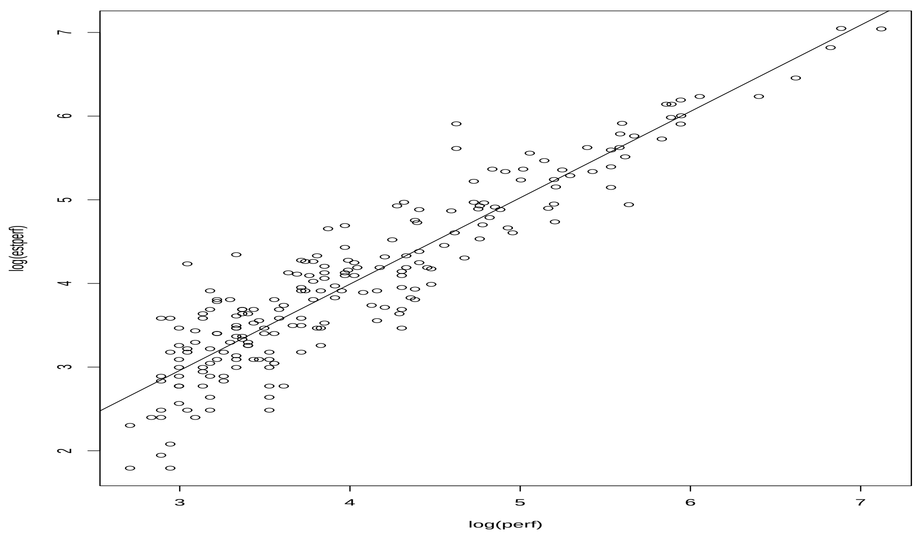


Figure 10: A plot of log(perf) $\sim$ log(estperf).

Statistical Models

Problem Solutions

Problem 3

See the R code chap_3_prob_3.R for an example of how to do this.

Problem 4

See the R code chap_3_prob_4.R.

Problem 5

See the R code chap_3_prob_5.R.

Problem 6

See the R code chap_3_prob_6.R.

Problem 8

If T is the time from now the next incident will occur in days and if we want the probability that the next accident happens in the next three weeks (21 days) we need to compute `pexp(21,rate=0.05)=0.65`.

Problem 9

This can be done with the R code `plot( density( rexp( 100, rate=0.2 ) ) )`. Running this code several times gives various density plots that look more or less like an exponential density (the difference is due to sampling error). Taking the mean of random samples like this gives a result like

```
> c( mean( rexp( 100, rate=0.2 ) ), 1/0.2 )
[1] 5.875883 5.000000
```

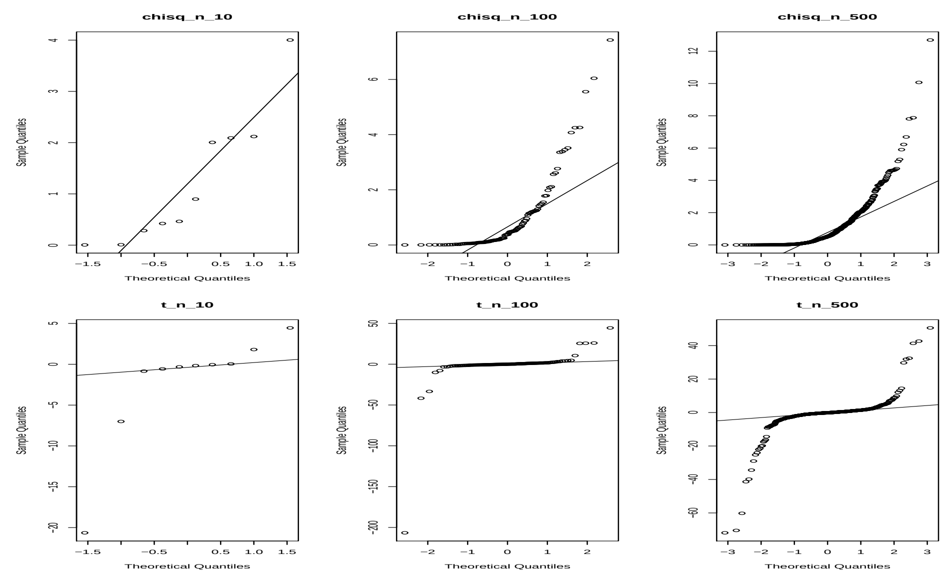


Figure 11: QQ plots of data drawn from a chi-squared distribution (first row of plots) and the t-distribution (second row of plots). The first column has $n = 10$ points, the second column has $n = 100$ points, the third columns has $n = 500$ points. For only a few number of points $n = 10$ data from these two distribution looks like it could be from a normal distribution. For larger samples sizes serious differences between the data generation process and the normal distribution are apparent.

Problem 10

See the R code chap_3_prob_10.R and Figure 11.

Problem 11

See the R code chap_3_prob_11.R. Since the variance of the data is not very close in value to the value of the data mean we could conclude that a Poisson model is not appropriate for this data.

Problem 12

See the R code chap_3_prob_12.R. When we run that code we get

```
> source('chap_3_prob_12.R')
 [1] 2 3 4 5 5 5 5 5 5 5 5 5 5 5 5
 [1] 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 [1] 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 [1] 2 3 2 3 2 3 4 5 5 5 5 5 5 5 5
 [1] 2 1 2 3 2 1 0 0 0 0 0 0 0 0 0
 [1] 2 3 4 3 4 5 5 5 5 5 5 5 5 5 5
 [1] 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 [1] 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 [1] 2 3 4 5 5 5 5 5 5 5 5 5 5 5 5
 [1] 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0
```

showing various outcomes.

Problem 13

See the `R` code `chap_3_prob_13.R`.

A Review of Inference Concepts

Problem Solutions

Problem 1 (the `nswdemo` dataset

See the `R` code `chap_4_prob_1.R`. When that code is run we get the following output

```
> source('chap_4_prob_1.R')
[1] "Part (a): Control group confidence interval:"
[1] 2530.773 3522.593
[1] "Part (a): Treated group confidence interval:"
[1] 2509.407 3622.789
[1] "Part (b): Control group confidence interval:"
[1] 4544.861 5635.236
[1] "Part (b): Treated group confidence interval:"
[1] 5185.685 6767.019
[1] "Part (b): Confidence interval of the difference"
[1] -1844.77823    72.17077
```

From the above output we see that there does not seem to be much evidence for a difference between the control and the treated groups in 1975 but there seems to be more of a difference between the two groups. To test this idea in Figure 12 we product a box and whiskers plot of the `re78` data. Given the above box and whiskers plot there does not seem to be much difference between the the two groups.

Problem 2 (the t critical values vs. the degrees of freedom)

See the `R` code `chap_4_prob_2.R`. When that code is run we get the output shown in Figure 13. The log transformation seems to create a better looking plot.

Problem 3 (possible values for the p-value)

See the `R` code `chap_4_prob_3.R`.

Problem 4 (the distribution of p-values)

See the `R` code `chap_4_prob_4.R`. When that code is run we get the output shown in Figure 14. The plot shown indicates that when drawing only $n = 10$ random variates the p-value from

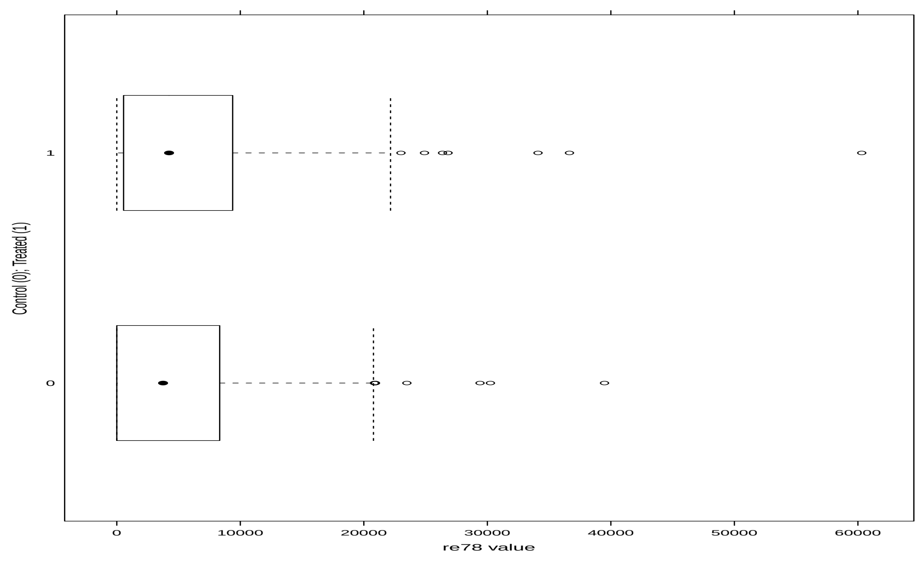


Figure 12: A visual comparison of the differences in re78 between the control (not treated) and the treated group in the nswdemo dataset.

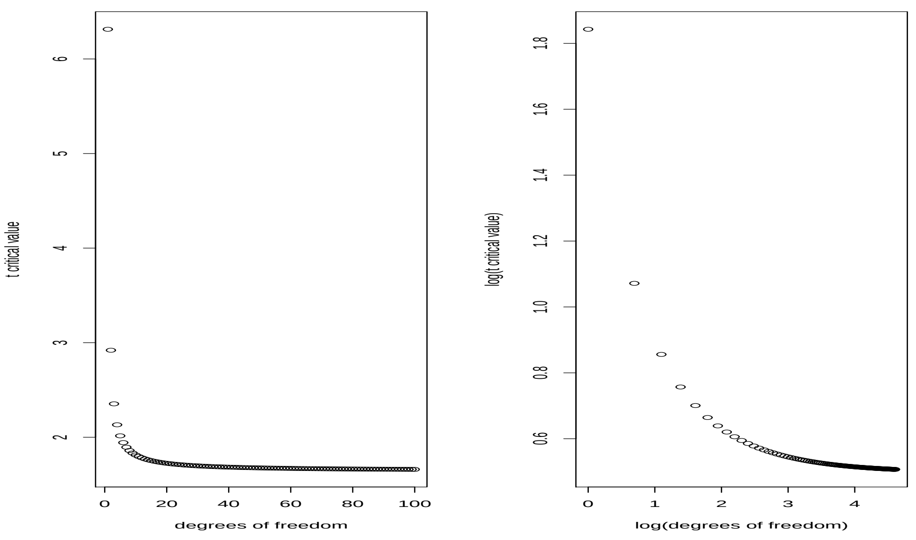


Figure 13: The critical values of the t statistics as a function of the number of degrees of freedom.

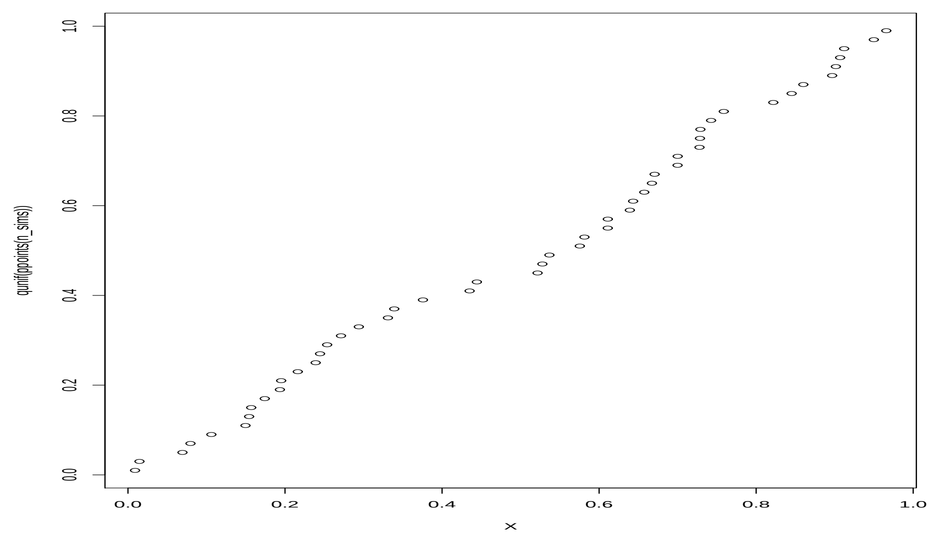


Figure 14: A *qq* plot of the distribution of p-values when computing the mean of 10 Gaussian random variables with *actual* mean zero.

a t-test seems to be drawn from a uniform distribution on $[0, 1]$. This is an indication that the p-value in this small sample size case is not very informative.

Problem 5 (more outliers of a normal or a t-distribution)

See the `R` code `chap_4_prob_5.R`. When that code is run we get the output shown in Figure 15. The t-distribution is a heavier tailed distribution than then normal and thus we expect to have more outliers flagged when we consider the various boxplots. We see this phenomena when we look at Figure 15.

Problem 6 (a time series with autocorrelation)

See the `R` code `chap_4_prob_6.R`. When that code is run we get the output shown in Figure 16.

Problem 7 (more series with autocorrelation)

See the `R` code `chap_4_prob_7.R`.

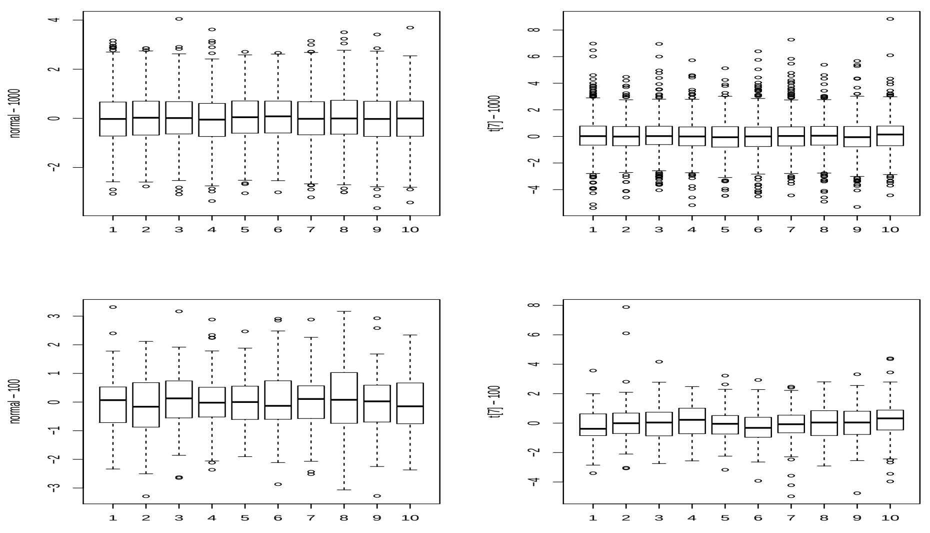


Figure 15: Looking for outliers using the R `boxplot` function.

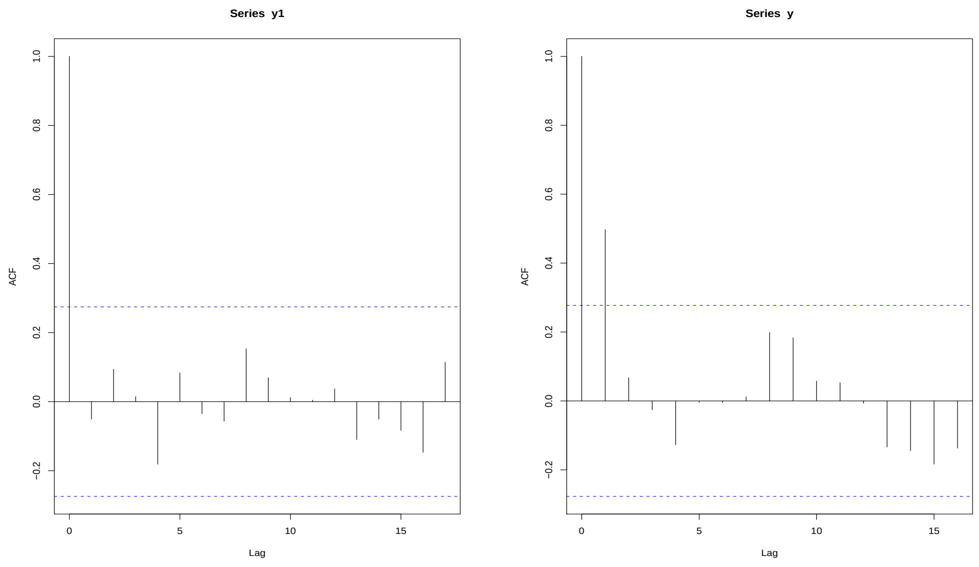


Figure 16: The time series `t1` has no serial autocorrelation (left), while the autocorrelation for `y` does (right). Note the significant spike at the lag $k = 1$ in the second plot.

Problem 8 (the `nswpsid3` dataset)

See the `R` code `chap_4_prob_8.R` for some of the calculations requested for this problem.

Problem 9 (acupuncture)

Part (a): From the table given it looks like whether a patient has the true acupuncture or the sham acupuncture there is not much difference in the resulting treatment outcome i.e. whether the treatment is actually able to actually reduce in headaches. For the patients that did not receive any acupuncture treatment (the column "Waiting list") it looks like a significant number of them *did not* have any reduction in headache symptoms. Perhaps the act of having *any* treatment makes people more likely to have reduced symptoms.

Problem 10 (using the command `mosaicplot`)

The `rareplants` data looks like

```
> rareplants
    D   W WD
CC 37 190 94
CR 23  59 23
RC 10 141 28
RR 15  58 16
```

and in `chap_4_prob_10.R` we use the `mosaicplot` command to get the output shown in Figure 17.

Problem 13 (the `overlapDensity` function)

See the `R` code `chap_4_prob_13.R` for some of the calculations requested for this problem.

Problem 14 (the z-transform in computing bootstrap CI of the correlation)

See the `R` code `chap_4_prob_14.R` for some of the calculations requested for this problem. When that code is run we get the following output

```
> source('chap_4_prob_14.R')
```

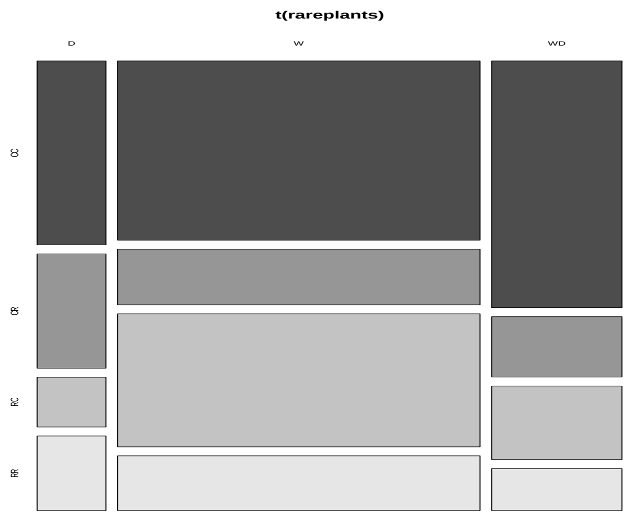


Figure 17: The output of the `mosaicplot` on the `rareplants` dataset.

```
[1] "CI: with the z-transform"
[1] 0.4686113 0.7102245
[1] "CI: without the z-transform"
[1] 0.4714532 0.7123547
```

From this output we see that there is not much difference in the two calculations. Thus it looks like the z-transformation is not needed here.

Problem 15 (paired observations)

See the `R` code `chap_4_prob_15.R`. When we run that command we get the output shown in Figure 18. The pot index 2 and 4 appear to have different values for the median difference between the `cross` and `self` variables.

Problem 16

See the `R` code `chap_4_prob_16.R`.

Problem 17

See the `R` code `chap_4_prob_17.R`. When that code is run we get an output like the following

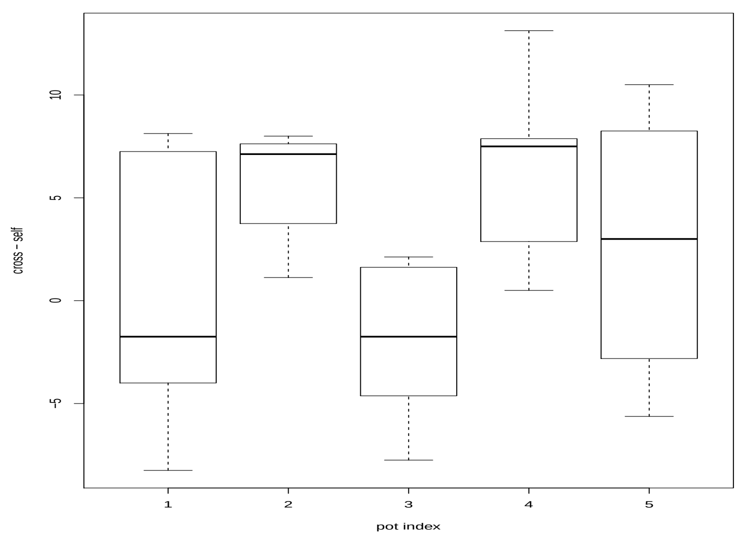


Figure 18: The median values and inter-quartile range of the difference between the `cross` and `self` in the `mignonette` dataset.

```
[1] "rate=  1.00; standard error of median (bootstrapped)   0.112211"
[1] "rate=  1.00; standard error of median (normal approximation)   0.125331"
```

Thus the bootstrapped estimate and the normal approximation seem to be quite close. If we increase the `rate` parameter in the exponential data generation (so that the data is more skewed) the comparison between the two methods is much worse

```
[1] "rate= 10.00; standard error of median (bootstrapped)   0.011221"
[1] "rate= 10.00; standard error of median (normal approximation)   0.125331"
```

Problem 19

See the `R` code `chap_4_prob_19.R`.

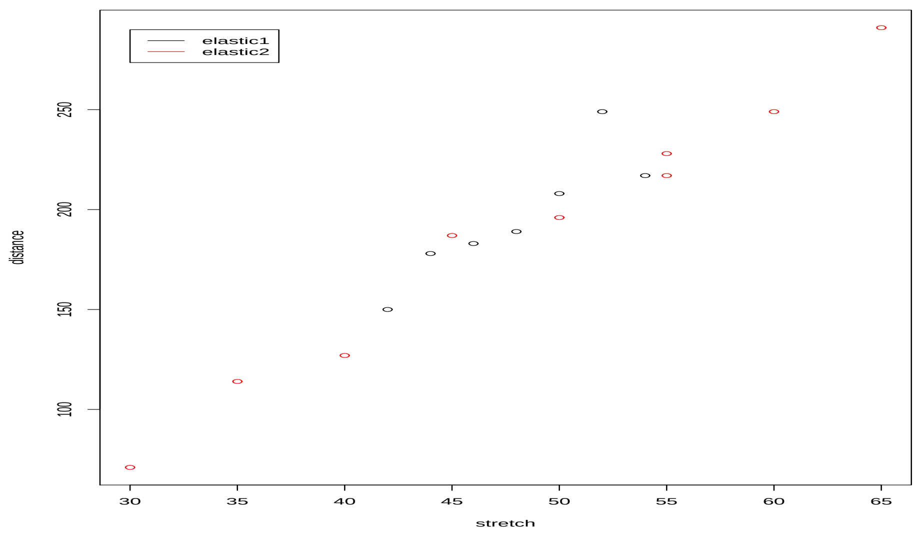


Figure 19: The black dots correspond to data from the `elastic1` data frame and the red dots correspond to data from the `elastic2` data frame.

Statistical Models

Problem Solutions

Problem 1

See the `R` code `chap_5_prob_1.R` and Figure 19. From this plot it looks like both sets of data lie on the same line.

Problem 2

See the `R` code `chap_5_prob_2.R`. The result of a `lm` call in `R` gives a field called `fitted.values` that gives the fitted values for each model. The standard error of these fitted values can be obtained by calling the `R` function `predict` with the `se.fit=TRUE` option. When called on the `lm1` object we get

```
> predict( lm1, se.fit=TRUE )
$fit
       1        2        3        4        5        6        7
183.1429 235.7143 196.2857 209.4286 170.0000 156.8571 222.5714

$se.fit
```

```
[1]  6.586938 10.621119  5.891537  6.586938  8.331891 10.621119  8.331891

$df
[1] 5

$residual.scale
[1] 15.58754
```

where we see the standard error for each fitted value. The two R^2 values are 0.799 and 0.981 for the data `elastic1` and `elastic2`. The better R^2 value of `elastic2` (qualitatively) is due to the fact that this data looks to be linear over a longer range of the `stretch`. In other words, it looks like the data in `elastic1` is a subset of that in `elastic2`.

Next we run the robust regression command `rlm` from the `MASS` package. Perhaps what is most interesting is how the robust regression changes the estimate of the linear models coefficients. For the data set `elastic1` the estimates of the linear intercept and the linear slope for ordinary least squares and robust least squares is given by

```
lm Coefficients:                                   | rlm Coefficients:
            Estimate Std. Error t value Pr(>|t|)  |             Value    Std. Error t value
(Intercept) -119.143     70.943  -1.679  0.15391  | (Intercept) -95.7472  60.6901    -1.5776
stretch        6.571      1.473   4.462  0.00663  | stretch       6.0397   1.2600     4.7934
```

The value of the slope is about the same while the intercept is somewhat different. For the `elastic2` data the same comparison gives

```
lm Coefficients:                                    | rlm Coefficients:
             Estimate Std. Error t value Pr(>|t|)  |             Value     Std. Error t value
(Intercept) -100.9167    15.6102  -6.465 0.000345  | (Intercept) -103.0550   14.4955    -7.1094
stretch        5.9500     0.3148  18.899 2.89e-07  | stretch        5.9752    0.2924    20.4379
```

Here the coefficients are much more similar.

Problem 3

See the `R` code `chap_5_prob_3.R` and Figure 20. From this plot we see that the quadratic fit is not much better than the linear fit. We can study if the quadratic fit is a better match to the data by using the `R` `anova` command to compare the two models. We find

```
> anova(m1,m2)
Analysis of Variance Table

Model 1: dist ~ speed
Model 2: dist ~ speed + I(speed^2)
```

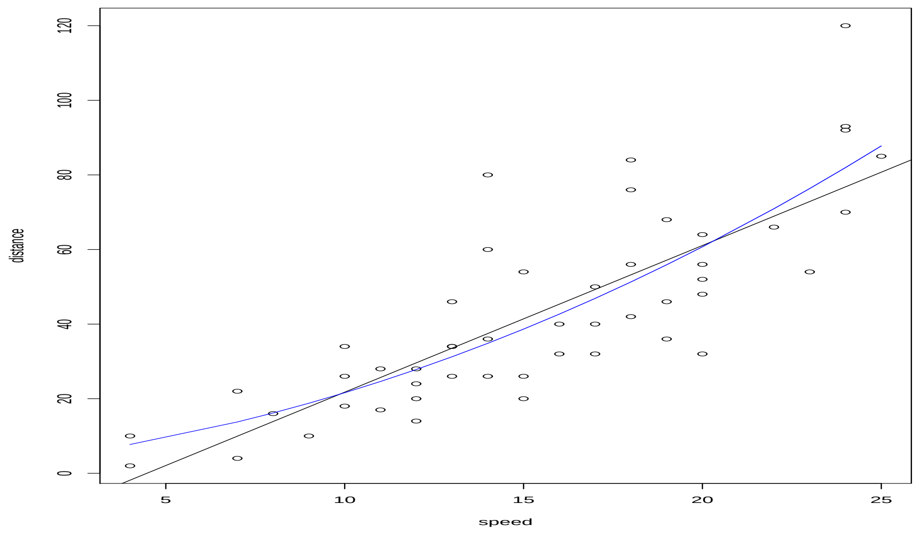


Figure 20: The black curve correspond to the linear fit to the data in the `cars` data frame and the blue curve correspond to a quadratic fit to the same data.

```
  Res.Df   RSS Df Sum of Sq     F Pr(>F)
1     48 11354
2     47 10825  1    528.81 2.296 0.1364
```

Thus there is a 13% chance that the quadratic models improvement is due to chance (i.e. not a real improvement). Given how large this percentage is I would opt to take the simpler (i.e. linear) model. We can look at the `summary` command on the quadratic model. Part of what we find is

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.47014   14.81716   0.167    0.868
speed        0.91329    2.03422   0.449    0.656
I(speed^2)   0.09996    0.06597   1.515    0.136
```

We first see that the estimated quadratic coefficient is very small and that the t value is relatively small. This indicates that again the addition of a quadratic term is perhaps not needed. The solutions presented to this problem argue that adding a quadratic term is still helpful (and valid to include) in that

- It does actually reduce the residual sum of squares.
- There is a clear pattern in the plot of the residuals as a function of the fitted values for the linear model that is not seen in the quadratic model.

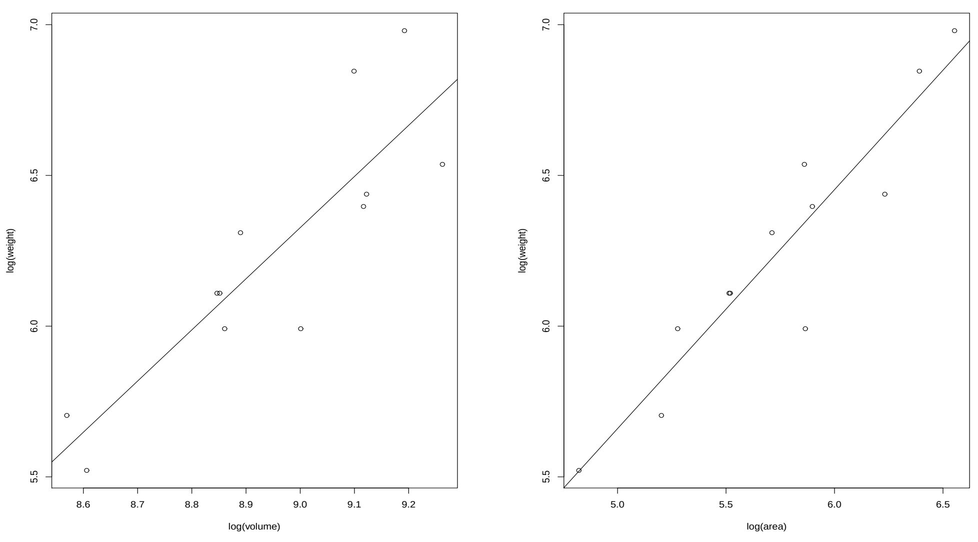


Figure 21: **Left:** Plots of the linear regression of weight on volume. **Right:** Plots of the linear regression for weight on area.

Problem 4 (the density of pages)

See the `R` code `chap_5_prob_4.R` and Figure 21. From these two plots we see that both linear fits look reasonable. The output from the `summary` command give that for the linear regression `log(weight) ~ log(volume)` has an R^2 of 0.7557 while for the linear regression `log(weight) ~ log(area)` has an R^2 of 0.8746. This and other indicators (F-statistic) indicate that the linear regression of weight on area is the better model fit.

Part (d): When we replace `log(density)` with `log(weight)` we again see that the regression on `area` seems to be the better fit.

Problem 12

See the `R` code `chap_5_prob_12.R` and Figure 22. In that figure we draw the linear regression coefficients that depend on how much relative error there is in x compared to y. Thus we can get quite different results (i.e. slopes) depending on how much noise there is in x relative to y.

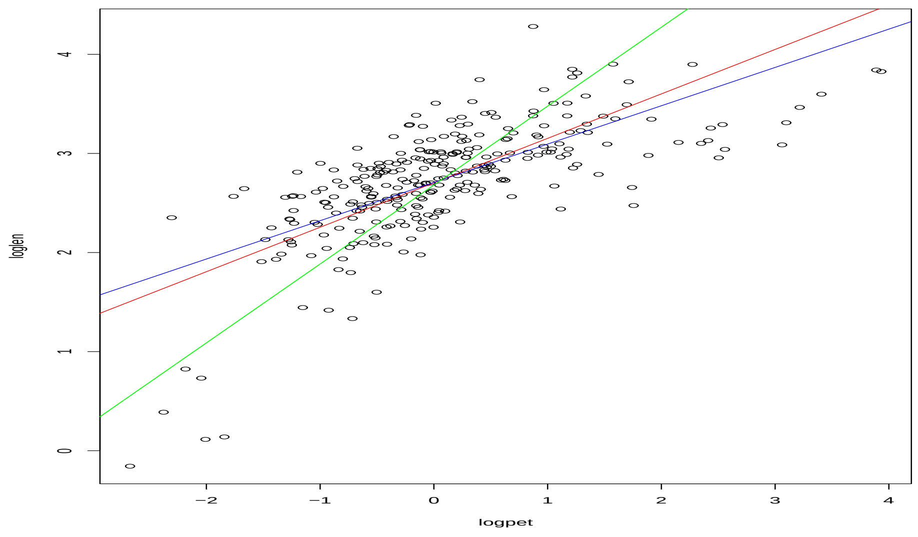


Figure 22: Data from the `leafshape` frame, specifically the `logpet` and `loglen` columns in black. We then plot the best fit line when there is equal amount of error in x and y in red, 10 times more error in y in blue, and 10 times more error in x in green.

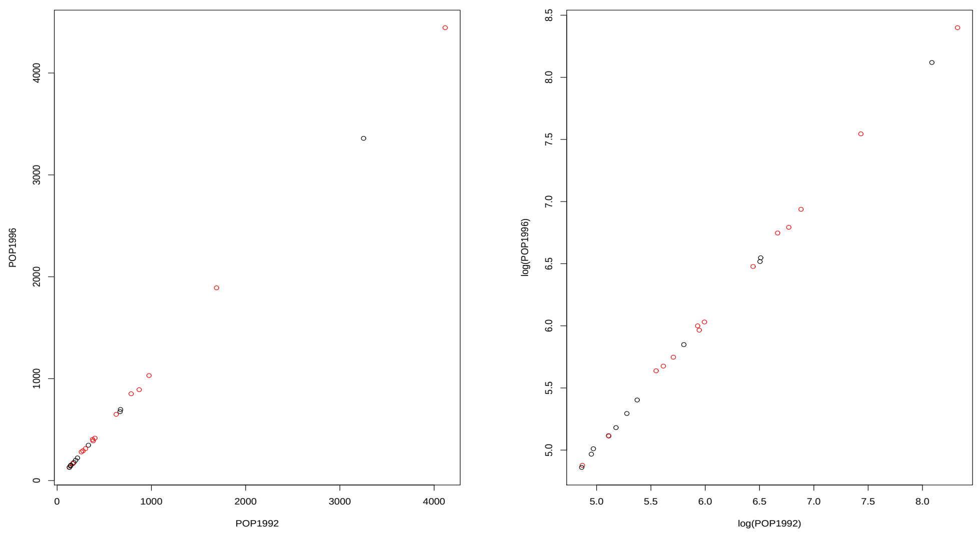


Figure 23: **Left:** Plot of `POP1996` vs. `POP1992`. **Right:** Plot of the *logarithms* of `POP1996` vs. `POP1992`.

Multiple Linear Regression

Problem Solutions

Problem 1 (modeling the `cities` dataset)

See the `R` code in `chap_6_prob_1.R`. When we plot the two suggested plots we get the results shown in Figure 23. From this plot it looks like taking the logarithm seems to spread the data apart a bit more. The four "summary" plots from the two linear models indicate that the second model has fewer outliers and seems to have data that is more easily modeled in the linear framework.

Problem 2 (head production in cement)

See the `R` code in `chap_6_prob_2.R`. A scatter plot of the predictors and the response is shown in Figure 24. From that plot it looks like several of the predictors are highly correlated. For example `x2` and `x4`. Thus we might expect that we should not include each of these variables in the model if we wish to have reasonable ability to estimate the coefficients of the model. In fact the "summary" of a linear model withe all datum as a predictor indicates that we cannot extract valid estimates of the coefficients

```
> summary(m0)
```

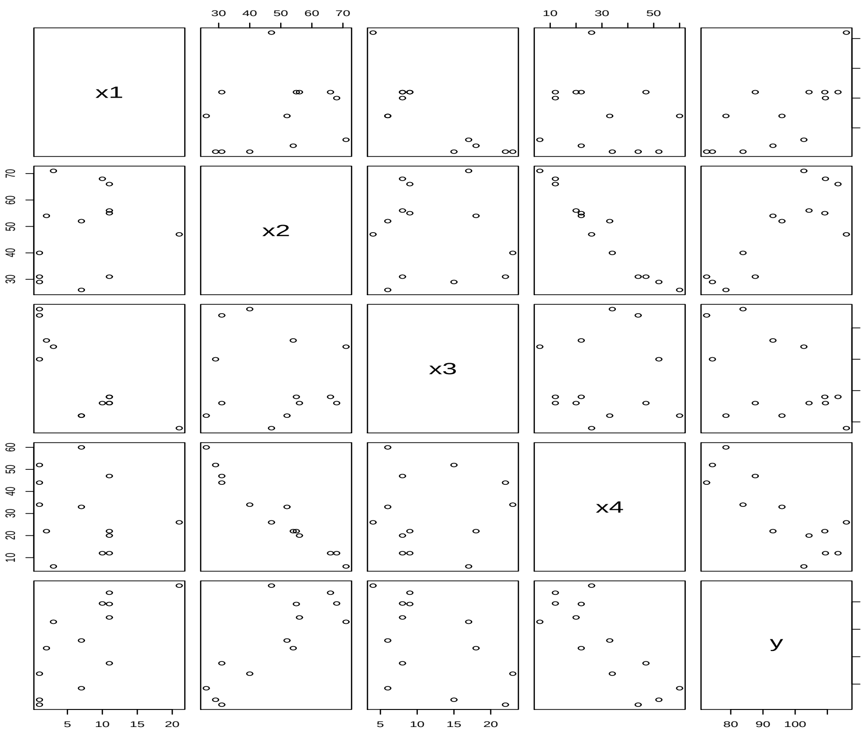


Figure 24: A `pairs` plot of the `cement` data frame.

```
Call: lm(formula = y ~ ., data = cement)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  62.4054    70.0710   0.891   0.3991
x1            1.5511     0.7448   2.083   0.0708 .
x2            0.5102     0.7238   0.705   0.5009
x3            0.1019     0.7547   0.135   0.8959
x4           -0.1441     0.7091  -0.203   0.8441
---

Residual standard error: 2.446 on 8 degrees of freedom
Multiple R-squared: 0.9824,     Adjusted R-squared: 0.9736
F-statistic: 111.5 on 4 and 8 DF,  p-value: 4.756e-07
```

Notice that the model as a whole is significant but none of the coefficients are. If we use the `stepAIC` command to sequentially remove predictors we find the the smallest AIC is obtained when we drop the variable `x3`. This variable also has one of the higher correlation with another variable (`x1`). Looking at the correlations between variables we get

```
> cor( cement )
           x1          x2           x3           x4           y
```

```
x1  1.0000000  0.2285795 -0.82413376 -0.24544511  0.7307175
x2  0.2285795  1.0000000 -0.13924238 -0.97295500  0.8162526
x3 -0.8241338 -0.1392424  1.00000000  0.02953700 -0.5346707
x4 -0.2454451 -0.9729550  0.02953700  1.00000000 -0.8213050
y   0.7307175  0.8162526 -0.53467068 -0.82130504  1.0000000
```

We see that x2 and x4 are almost perfectly correlated. Dropping the term x4 we get the model

```
> summary(m2)

Call: lm(formula = y ~ x1 + x2 + x3, data = cement)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 48.19363    3.91330  12.315 6.17e-07 ***
x1           1.69589    0.20458   8.290 1.66e-05 ***
x2           0.65691    0.04423  14.851 1.23e-07 ***
x3           0.25002    0.18471   1.354    0.209
---

Residual standard error: 2.312 on 9 degrees of freedom
Multiple R-squared: 0.9823,     Adjusted R-squared: 0.9764
F-statistic: 166.3 on 3 and 9 DF,  p-value: 3.367e-08
```

Thus we are now accurately estimating the coefficients for x1 and x2. Since the coefficient of x3 is estimated poorly we might consider dropping it. When we perform the suggested log transformation and then look at the summary plots we see that this model has a few more outliers than the original (untransformed) model. Since the number of data points this model has is so small it is probably best to stick with the untransformed model.

Problem 3 (hill climbing data)

For this problem see the R code in chap_6_prob_3.R. Here we learn two models on hill climbing data. The first is using the data frame nihills and the second is from the data frame hills2000. We then compute the variance in the predictions of each model with the true values of log(time). Since the linear model when fitting this target using the hills2000 data is explicitly minimizing an expression like this we expect the variance obtained when we build a model using the exact data to be smaller. Indeed this is what we observe

```
[1] "var using nihills=   0.019447; using hills2k=   0.014041"
```

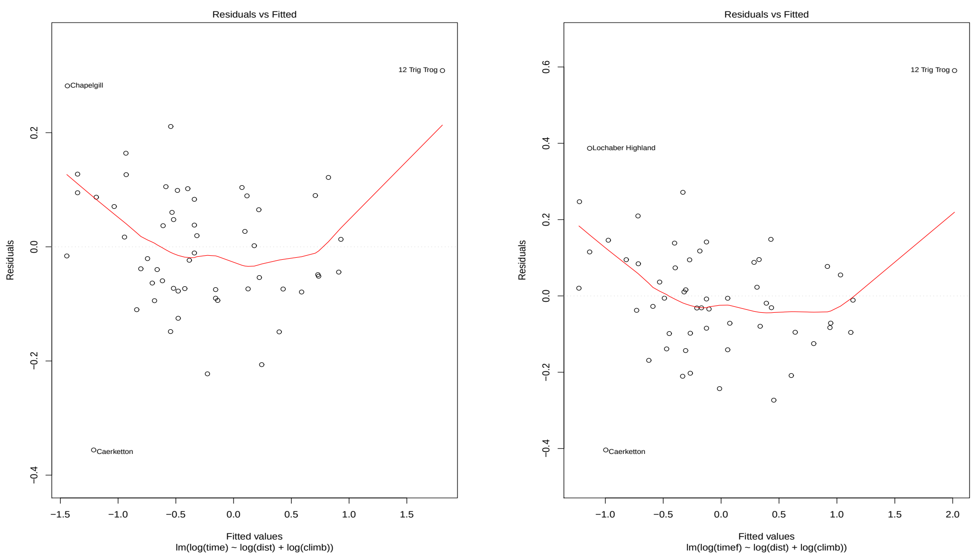


Figure 25: **Left:** The first linear model diagnostic plot for male hill climbers in the `hills2000` dataset. From the first plots in this series some potential outliers might be `Chapelgill`, `12 Trig Trog`, and `Caerketton`. **Right:** The first linear model diagnostic plot for female hill climbers in the `hills2000` dataset. The races `Caerketton` and `12 Trig Trog` still stand out as potential outliers along with the race `Lochaber Highland`.

Problem 4 (the `hills2000` dataset)

For this problem see the `R` code in `chap_6_prob_4.R`. Since the `hills` dataset does not have the field `timef` we will fit models for male and female walkers using the `hills2000` dataset. When we run that script we get the two plots shown in Figure 25. From these plots we would refit our model with the data for `Caerketton` and `12 Trig Trog` omitted. If we refit our two models over male and female racers we get two sets of coefficients for the given models For the male racers we get

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.96700    0.16722  -23.72  < 2e-16 ***
log(dist)    0.73829    0.03175   23.26  < 2e-16 ***
log(climb)   0.31815    0.02696   11.80 3.39e-16 ***
```

While for the female racers we get

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.68805    0.21708 -16.990  < 2e-16 ***
log(dist)    0.70018    0.04074  17.188  < 2e-16 ***
log(climb)   0.31782    0.03494   9.096 3.53e-12 ***
```

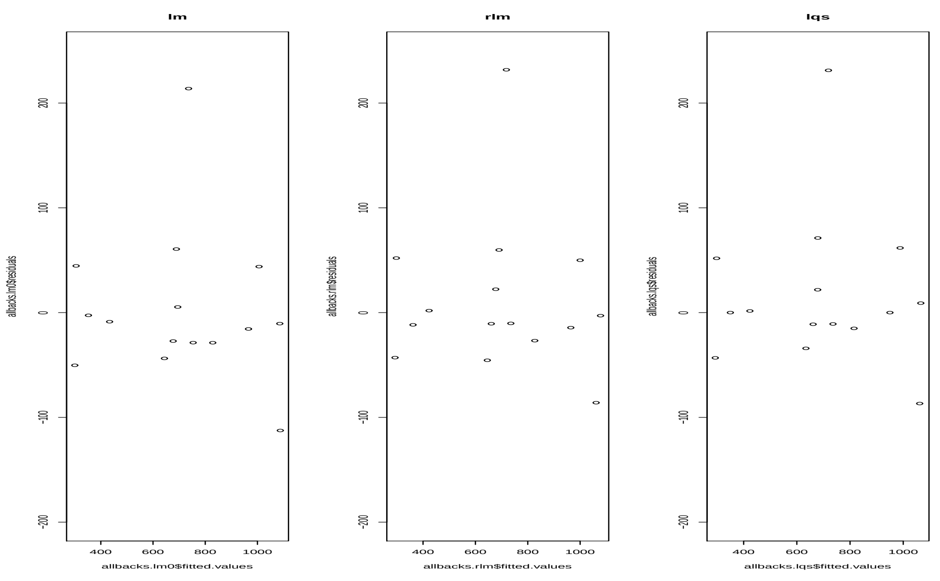


Figure 26: Residual vs. fitted value plots for various linear modeling techniques. Outliers are easily seen as data with very large y values.

Notice that the coefficiets of `log(dist)` and `log(climb)` are equal (within the accuracy of their standard errors) while the intercept term is not.

Problem 5 (robust linear models)

Since it was decided in the section of the book that an intercept term is not needed we will perform regression fits using `lm`, `rlm`, and `lqs` without one. In the `R` code in `chap_6_prob_5.R` we first produce residual vs. fitted value plots for each of the three models. These are displayed in Figure 26. This is a nice diagnostic plot for looking for outliers in the data. From each model we see that the data points 13 (large positive residual) and 11 (relatively large negative residual) could be considered outliers. Because the robust fitting methods places less emphasis on the point 13 it ends up having a larger residual value under the two robust models. We might consider dropping these points and observing the effect that this procedure has on our model coefficients when they were included (or not).

```
                     volume    wo 11/13        area    wo 11/13
coefficients(lm0)   0.7289086   0.7172330   0.4808668   0.5059043
coefficients(rlm)   0.7111266   0.7159097   0.5168193   0.4936729
coefficients(lqs)   0.7116612   0.7011605   0.4849406   0.5077369
```

We see that when we remove the two outliers the coefficient estimate are more closely clustered.

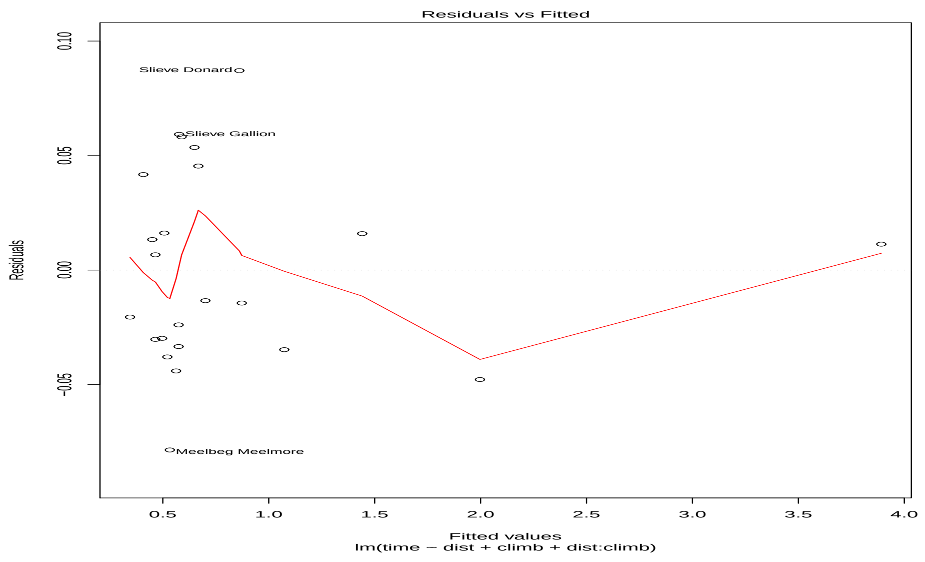


Figure 27: Residual vs. fitted value plots for various the model with the interaction term of `dist:climb`. It looks like the two points `Slieve Donard` and `Meelbeg Meelmore` might be outliers.

Problem 6 (modeling the `nihills` dataset with an interaction term)

See the `R` code in `chap_6_prob_6.R` for this problem. The `anova` command gives the following comparison between the two models

```
> anova( nihills.lm, nihills2.lm )
Analysis of Variance Table

Model 1: time ~ dist + climb
Model 2: time ~ dist + climb + dist:climb
  Res.Df      RSS Df Sum of Sq      F    Pr(>F)
1     20 0.189361
2     19 0.039361  1      0.15 72.406 6.623e-08 ***
```

This would indicate that the reduction in `RSS` is "sufficient" when we include the interaction term and it should be kept in the model. If we next produce a residual vs. fitted value plot we get the result shown in Figure 27. Based on that plot we identify a few points that might be outliers, remove them and refit the two models. When we do that the `anova` command gives

```
> anova( nihills.lm, nihills2.lm )
Analysis of Variance Table

Model 1: time ~ dist + climb
Model 2: time ~ dist + climb + dist:climb
  Res.Df      RSS Df Sum of Sq      F    Pr(>F)
1     18 0.180804
2     17 0.024907  1   0.15590 106.40 9.805e-09 ***
```

Again we see that the interaction term results in a significant reduction in RSS and thus should be kept. This gives stronger evidence that the interaction term might be a real effect. This gives motivation for taking the logarithm of all variables (turn products into sums).

Problem 7 (variance inflation factors)

See the R code in chap_6_prob_7.R for this problem. When that script is run we get the following variance inflation factors for the two variables

```
> vif( litters.lm )
bodywt  lsize
 11.33  11.33
```

There is almost a linear relationship between `lsize` and `bodywt` which cause such large values for variance inflation factors.

Problem 8 (using `lm.ridge`)

See the R code in chap_6_prob_8.R for this problem. When that script is run we break the dataset `litters` into three groups to use for cross validation. For each value of λ we estimate the mean square error and the one sigma uncertainty in that value. Plots of estimate of the mean square error as a function of λ are given in Figure 28.

Part (a): The coefficients of the ridge regression linear model and the normal linear model are given by

```
 (Intercept)       bodywt        lsize   method
 0.381983207  0.005518766 -0.001065681   ridge regression
 0.178246962  0.024306344  0.006690331   standard linear regression
```

Note that the coefficients seem to be quite different. One would need to look at the standard errors to be sure that the difference was significant.

Part (b): When we compute a Monte-Carlo estimate of the prediction accuracy for the ridge regression linear model (with $\lambda = 10$) and the normal linear regression model give

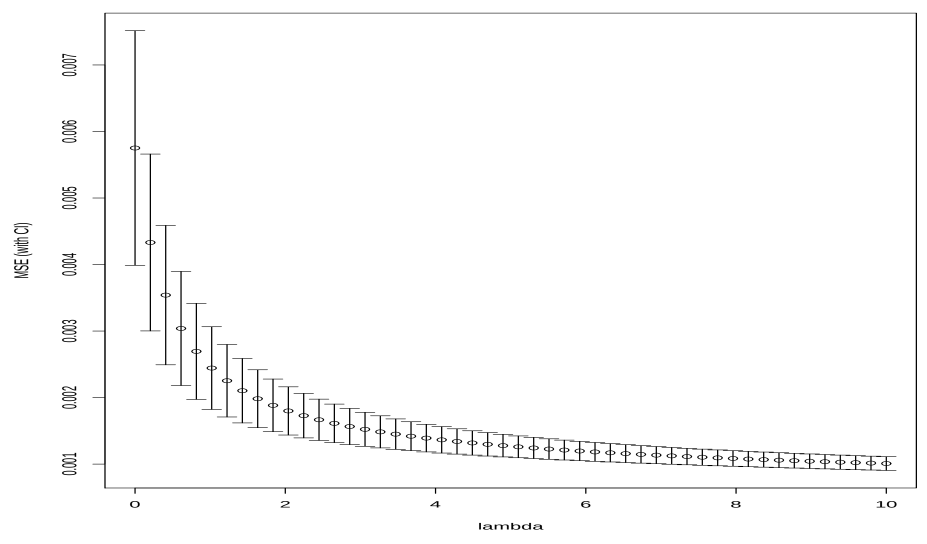


Figure 28: Plots of the estimate of the mean square error as a function of regularization parameter λ. It looks like the larger the value of λ the smaller the mean square error. The largest value of λ considered was $\lambda = 10$.

```
predition estimate        5%        95%     method
         0.4103799 0.4063087 0.4152301    ridge regresssion
         0.4153171 0.4102774 0.4194162    standard linear regresssion
```

Using the standard output from the function `lm.predict` for this sample we get

```
        fit       lwr       upr
1 0.4152947 0.4062582 0.4243312
```

These estimates look rather close.

Problem 9

See the `R` code in `chap_6_prob_9.R` for this problem. We get ranges (the difference between the maximum and the minimum value of the variable) for the two variables in the datasets given by

```
dist climb range_ratio    data_frame
16.4 8025  489.32         nihills
44.5 7000  157.30         hills2000
```

Since the ratio of these two ranges is largest for the `nihills` data set we would expect that it would be the harder data set to fit with a linear model. When we perform the two fits suggested we get (just the residual fit components of the `summary` command) for the log model and quadratic model on the `nihills` data frame

```
Residual standard error: 0.0766 on 20 degrees of freedom
Multiple R-squared: 0.9831,     Adjusted R-squared: 0.9814
F-statistic: 582.7 on 2 and 20 DF,  p-value: < 2.2e-16

Residual standard error: 0.05753 on 20 degrees of freedom
Multiple R-squared: 0.9948,     Adjusted R-squared: 0.9943
F-statistic:  1929 on 2 and 20 DF,  p-value: < 2.2e-16
```

This looks like the quadratic model has a much larger F-statistics for the same degrees of freedom. For the `hills2000` data frame we get

```
Residual standard error: 0.1207 on 53 degrees of freedom
Multiple R-squared: 0.9706,     Adjusted R-squared: 0.9695
F-statistic:   874 on 2 and 53 DF,  p-value: < 2.2e-16

Residual standard error: 0.1992 on 53 degrees of freedom
Multiple R-squared: 0.9714,     Adjusted R-squared: 0.9704
F-statistic: 901.3 on 2 and 53 DF,  p-value: < 2.2e-16
```

This again indicates that the quadratic model looks better *in-sample* for this data frame.

Problem 10

See the `R` code in `chap_6_prob_10.R` for this problem. When we run that script we generate the plots given in Figure 29. We would find the points 9, 15, 20 as possible outliers to the linear model. When we plot the residuals vs. fitted values of the suggested linear model as a function of `x7` and separated by the value of `x11` we see that the residuals for the cases where `x11=1` are more tightly clustered than the ones when `x11=0`. Thus our model for `x11=1` should be a better model then the one for `x11=0`.

Problem 11

See the `R` code in `chap_6_prob_10.R` for this problem. In the first part of this code we duplicate the books observation that when the model is incorrectly specified the estimated coefficients can be in gross error. When we make the change to make

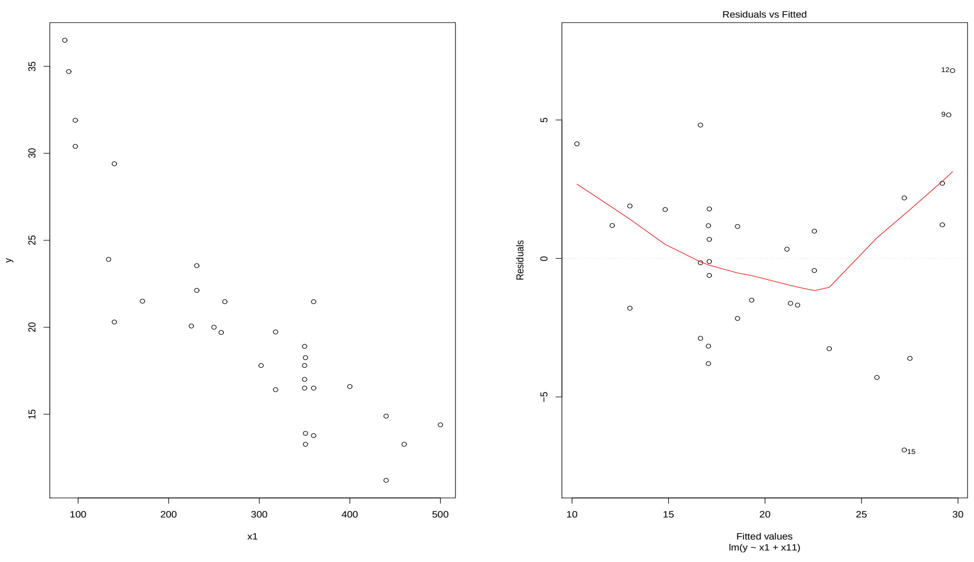


Figure 29: **Left:** A scatter plot of `y` as a function of `x1`. **Right:** The residual vs. fitted plot for the model `y ~ x1 + x11`, where `x11` is a factor. The curvature of the data smooth could indicate that quadratic terms should be included in the model.

```
x2 <- rbinom(10, 1, x1)
```

we should not see a qualitative difference in output. The coefficients from the two models extracted give

```
(Intercept) factor(x2)1
   2.201276    2.609256

(Intercept)          x1 factor(x2)1
0.001037955 4.956124428 1.031046806
```

We see that in the first case the intercept and the coefficient of `x2` are wrong while in the full specified model they are "correct" again. If we change `x2` to be

```
x2 <- rbinom(10, 1, 0.5)
```

Now the value of `x2` does not depend on `x1` at all and its inclusion in the model is independent of `x1`. The coefficients of the linear models become

```
(Intercept) factor(x2)1
  3.2963770   0.7711775
```

```
(Intercept)          x1 factor(x2)1
 0.03392788  4.93702354  1.02600691
```

Now we get reasonable estimate of the coefficient of x2. If we consider taking more data we should be able to estimate this coefficient even better. With 10000 data points the incorrect model (when not including the term x1 gives coefficients of

```
(Intercept) factor(x2)1
   2.495551    1.010454
```

which is close to true coefficient of 1.

Problem 12 (a different nonlinear model)

It seems like this problem is just asking us to estimate a different nonlinear model. I don't think there is anyway to related the estimated coefficients between the two models. If one model is correct and second model is incorrect then the estimated coefficients in the incorrect model will have "noise" introduced due to the incorrect model. This will make their estimate potentially very wrong.

Problem 14

See the R code in chap_6_prob_14.R for this problem. For the second model (the one with the cube root) the absolute values of the residuals are much smaller (order 1 vs. order 100) and the QQ-plot has fewer samples in the tails. These two facts indicate that the cube root model seems to be the better model. The fact that the coefficients are so different is simply a reflection on the fact that each is a coefficient in a different model and stands for a different thing.

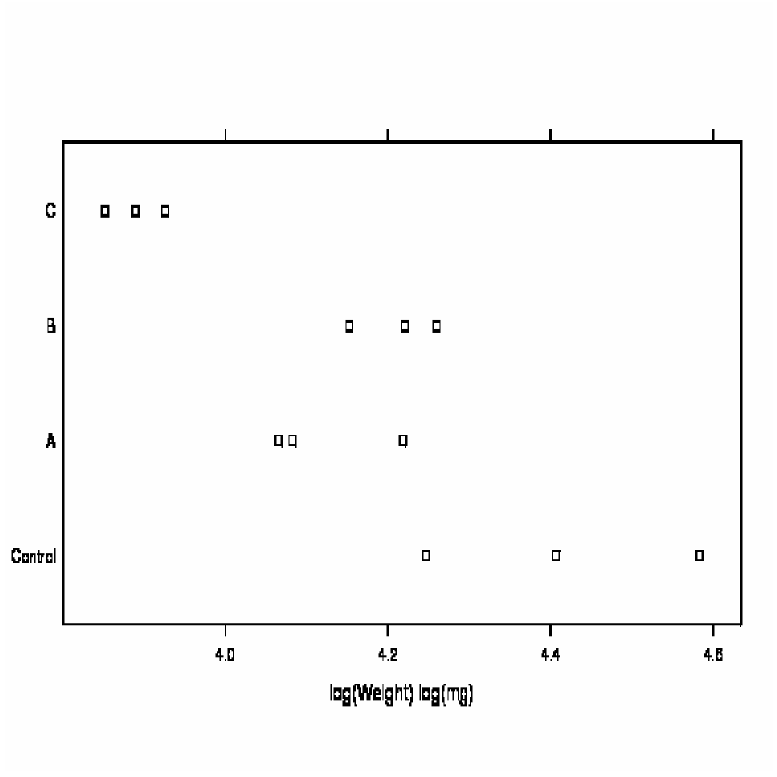


Figure 30: The distribution of log `weights` for the `sugar` dataset.

Exploiting the Linear Model Framework

Problem Solutions

Problem 1 (reducing variance with $\log(\cdot)$)

See the `R` code in `chap_7_prob_1.R`. We take the logarithm of `weight` and then apply the `aov` function to estimate the means of the three treatments (relative to the mean of the control weights). We plot the distribution of log weights in Figure 30. For the log weight the output of the `summary.lm` we get

```
> summary.lm(sugar.aov)

Call:
aov(formula = logWeight ~ trt, data = sugar)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.41224    0.05738  76.900 9.11e-13 ***
trtA        -0.29015    0.08114  -3.576 0.007232 **
```

```
trtB        -0.20109    0.08114  -2.478 0.038220 *
trtC        -0.52291    0.08114  -6.444 0.000200 ***

Residual standard error: 0.09938 on 8 degrees of freedom
Multiple R-squared: 0.8426,     Adjusted R-squared: 0.7835
F-statistic: 14.27 on 3 and 8 DF,  p-value: 0.001414
```

The output from the above seems to be slightly better than the output obtained when we don't take the logarithm, in that the F-statistics above is slightly larger and the t values are slightly larger. Again the differences in the mean from the control for each treatments are significant at the 5% level.

Problem 2 (adding a quadratic term)

See the `R` code in `chap_7_prob_2.R`. The `anova` command gives the following

```
Model 1: depression ~ weight
Model 2: depression ~ weight + I(weight^2)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1      8 362.93
2      7 298.58  1    64.359 1.5089  0.259
```

The large p-value indicates the quadratic term is not needed.

Problem 3 (`elastic1` and `elastic2`)

See the `R` code in `chap_7_prob_3.R`. For this problem we follow the steps outlined in the books section 7.3 "fitting multiple lines". There we consider four models of increasing complexity and then select one based on statistical tests. In this problem there are two types of elastic each of which is considered a factor. See Figure 19 for a plot of the data corresponding to the two types of elastic. The four models we consider are

- distance is estimated using a constant mean (independent of type).
- distance is estimated using a constant mean and slope (independent of type).
- distance is estimated using different constants and a common slope.
- distance is estimated using different constants and different slopes.

For each of these different models we can use the `R` command `anova` to select the final model. When we run the above script we get the output

```
> anova( elastic.lm1, elastic.lm2, elastic.lm3, elastic.lm4 )
Analysis of Variance Table

Model 1: distance ~ 1
Model 2: distance ~ stretch
Model 3: distance ~ stretch + type
Model 4: distance ~ stretch + type + stretch:type
  Res.Df   RSS Df Sum of Sq        F    Pr(>F)
1     15 46122
2     14  2549  1     43573 264.3302 1.542e-09 ***
3     13  2017  1       532   3.2249   0.09773 .
4     12  1978  1        39   0.2381   0.63435
```

From the above table it looks like the change in adding different intercepts is not significant and neither is migrating to models where each elastic type has a different slope. Thus we would prefer a single (one intercept and one slope) model for both types of elastic.

Problem 4 (the toycars dataset)

See the R code in chap_7_prob_4.R. For the suggested model we find an R^2 given by 0.9451. When we plot the first residual we find that the 17th sample appears to be a large outlier. This means that the data is in error, or the model is in error, or both. If we trust our model, we should remove this sample and refit the model. If we fit the second suggested model (different slopes for each car type) we get an R^2 given by 0.9413 (smaller than before). The 17th sample appears to be less of an outlier. This means that if we assume our data is correct this means that this new model is perhaps the preferred model.

Problem 5 (the cuckoos dataset)

See the R code in chap_7_prob_5.R. When we perform the suggested comparison and then run the anova command we get the following output

```
> anova( cuckoos.lm_a, cuckoos.lm_b, cuckoos.lm_c )
Analysis of Variance Table

Model 1: length ~ breadth
Model 2: length ~ breadth + species
Model 3: length ~ breadth + species + breadth:species
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1    118 101.923
2    113  79.114  5   22.8096 6.5711 2.244e-05 ***
3    108  74.978  5    4.1356 1.1914    0.3182
```

This indicates that adding specific intercepts (and keeping a constant slope) seems to result in a valid model improvement.

Problem 6 (the cuckoos dataset with robust regression)

When we use the R function anova on the outputs of the robust regression plots we get

```
> anova( cuckoos.rlm_a, cuckoos.rlm_b, cuckoos.rlm_c )
Analysis of Variance Table

Model 1: length ~ breadth
Model 2: length ~ breadth + species
Model 3: length ~ breadth + species + breadth:species
  Res.Df     RSS Df Sum of Sq F Pr(>F)
1        101.935
2         79.458      22.4768
3         76.189       3.2691
```

The numerical output of this command looks very similar to that from the non-robust lm output. This indicates that the model "B" (different intercepts for each species with a common slope) may still be the preferred model even when using robust regression.

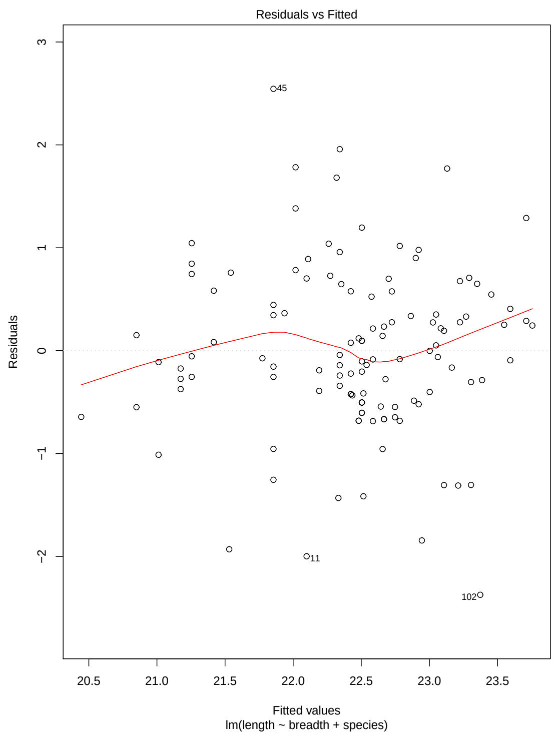
If we compare the coefficients computed using the two procedures lm and the rlm we get

```
Coefficients:       (lm results)                      (rlm results)
                    Estimate Std. Error t value     Value   Std. Error t value
(Intercept)          9.51559    3.01765   3.153      9.1745  2.7468     3.3401
breadth              0.81117    0.17951   4.519      0.8344  0.1634     5.1068
speciesmeadow.pipit -0.80125    0.25610  -3.129     -0.9052  0.2331    -3.8829
speciespied.wagtail -0.01324    0.31454  -0.042     -0.0338  0.2863    -0.1180
speciesrobin        -0.30310    0.31137  -0.973     -0.3140  0.2834    -1.1079
speciestree.pipit    0.04490    0.31143   0.144      0.0940  0.2835     0.3317
specieswren         -1.23912    0.35300  -3.510     -1.2691  0.3213    -3.9496
```

There is some small changes in the numerical value of the coefficients.

Problem 7 (polynomial regression on the geophones dataset)

See the R code in chap_7_prob_7.R. When that code is run we get the plot shown in Figure 32 (left). The fourth order model seems to curve up at the two extremes of the plot. The data seems to also curve up somewhat, an understanding of the problem domain would help to determine which model is more likely.



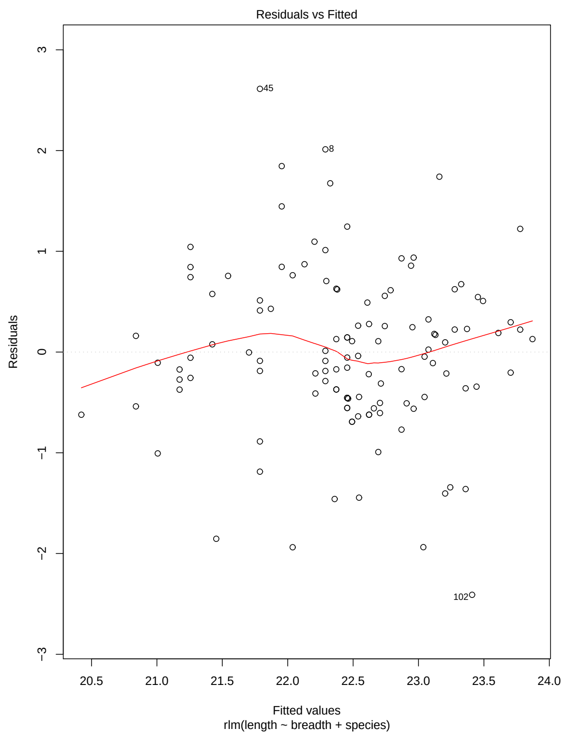


Figure 31: Residuals vs. fitted values for the model in two regressions **Left:** from the linear model `lm` **Right:** from the robust regression `rlm`. There does not seem to be much difference.

Problem 8 (spline regression on the `geophones` dataset)

See the `R` code in `chap_7_prob_8.R`. When that code is run we get the plot shown in Figure 32 (right). In general the spline models fit the data better than polynomial models.

Problem 9 (worldRecords)

See the `R` code in `chap_7_prob_9.R`. There does not seem to be much difference between the models.

Problem 10 (applying lowess smoothing)

See the `R` code in `chap_7_prob_10.R`.

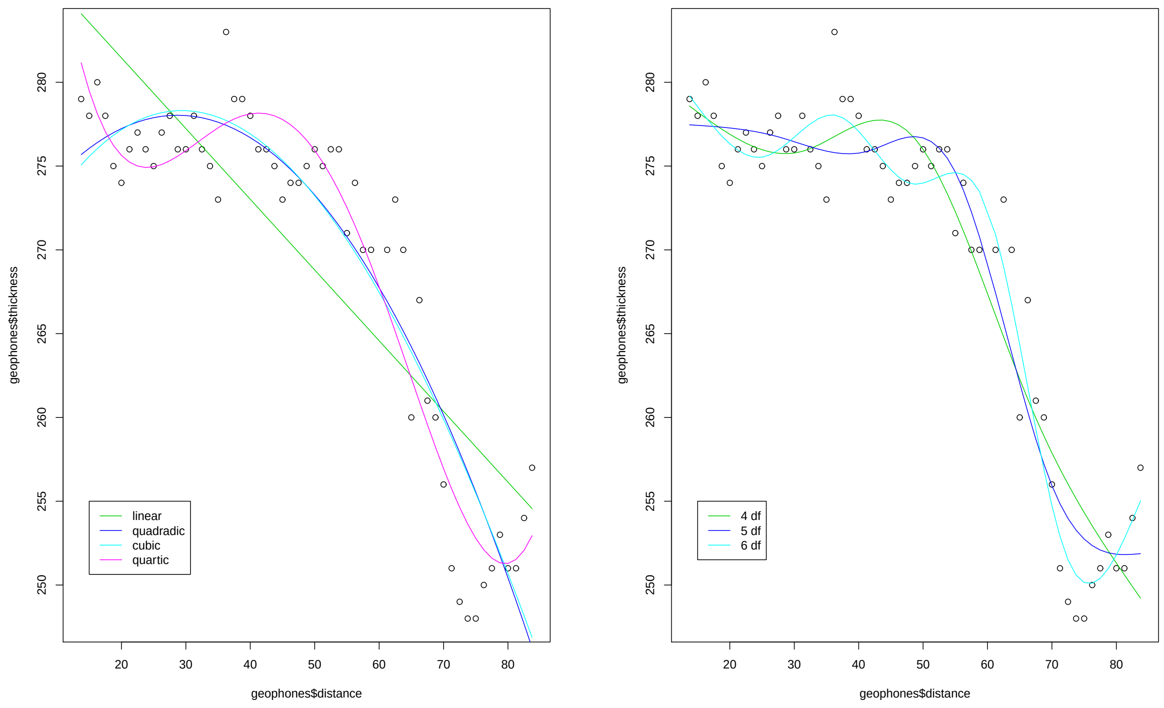


Figure 32: Raw data for the `geophones` dataset. **Left:** Plots of the fitted values for various polynomial models for Problem 7. **Right:** Plots of the fitted values for various spline models for Problem 8.

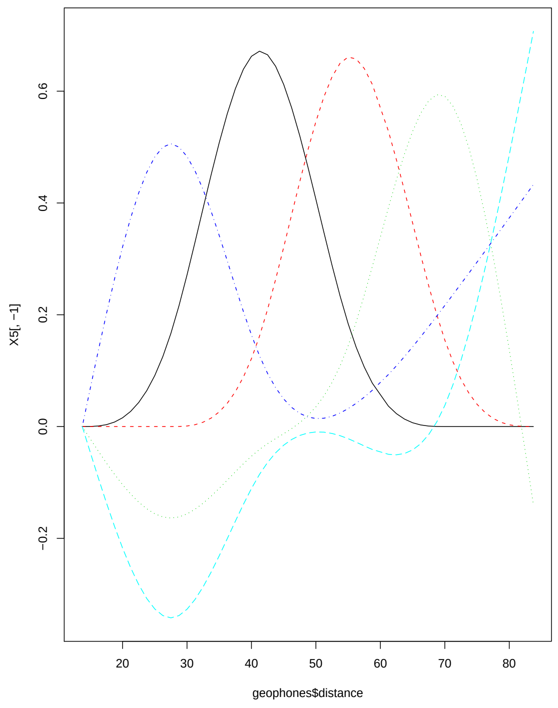


Figure 33: Plots of the spline basis function selected for Problem 12.

Problem 11 (influential outliers?)

See the R code in `chap_7_prob_11.R`. From the first plot (residual vs. fitted values) it looks like the data samples 19, 39, and 42 might be outliers. As there are several samples in the lower left of the plot with negative residuals of the same magnitude ≈ 5 we might *not* actually consider these points as extream.

Problem 12 (the spline basis curves)

See the R code in `chap_7_prob_12.R`. When that code is run we get the plot shown in Figure 33.

Problem 13 (using `s` in the `mgcv` library)

See the R code in `chap_7_prob_13.R`. When that code is run we get the plot shown in Figure 34. We notice from that plot that when we set the option $k = 20$ the smoothed curve is overfitting the data a bit. One could use cross-validation to select an optimal value for k.

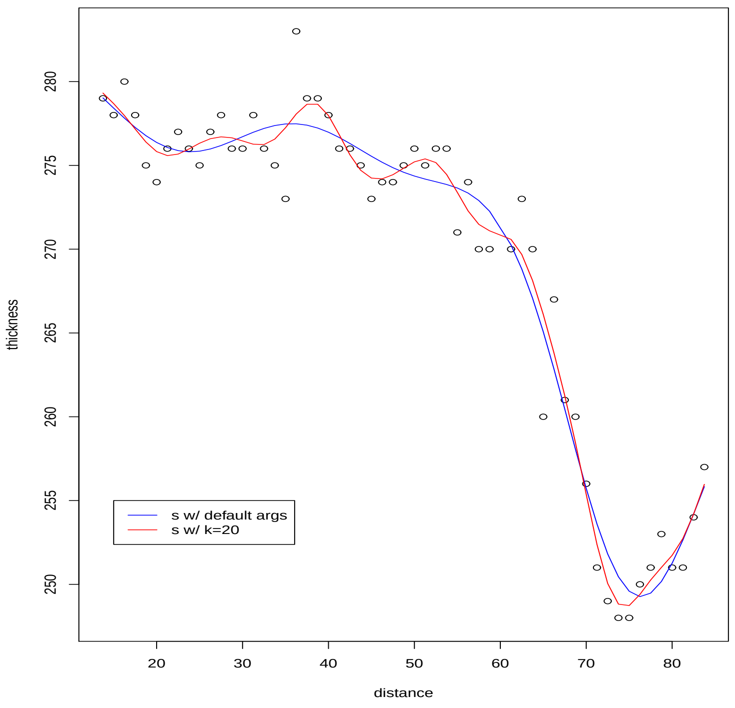


Figure 34: Plots of the spline smoothing from `mgcv`'s function `s` for Problem 13.

Problem 15 (the `wages1833` data set)

See the `R` code in `chap_7_prob_15.R`. When that code is run we get the plot shown in Figure 35. The curves seem very different.

Problem 16 (merkats)

See the `R` code in `chap_7_prob_16.R`. Due to the nonlinear nature of the data points we choose to model the points using `loess`. As there are not very many data points its hard to objectively pick a smoothing parameter f.

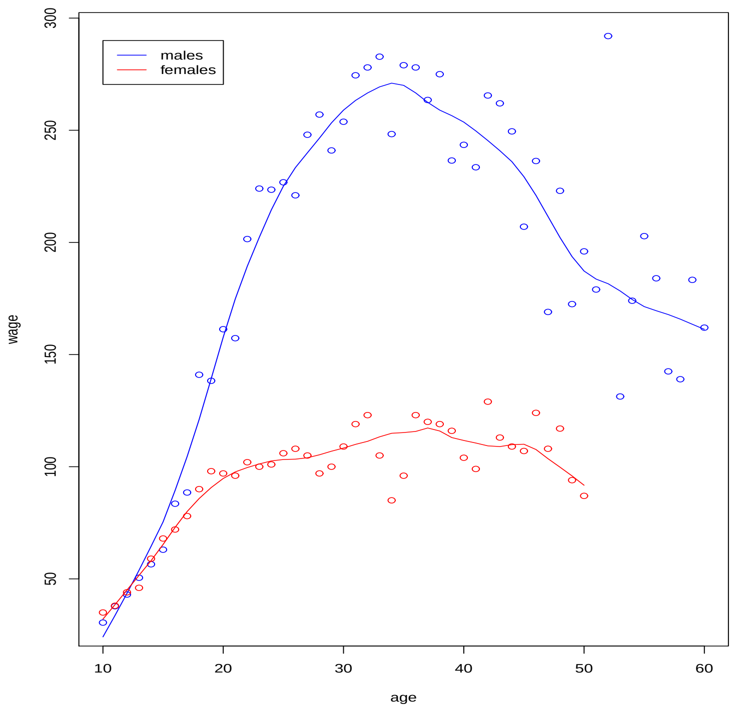


Figure 35: A lowess smooth of the male/female wage data from the `wages1833` data set of Problem 15.

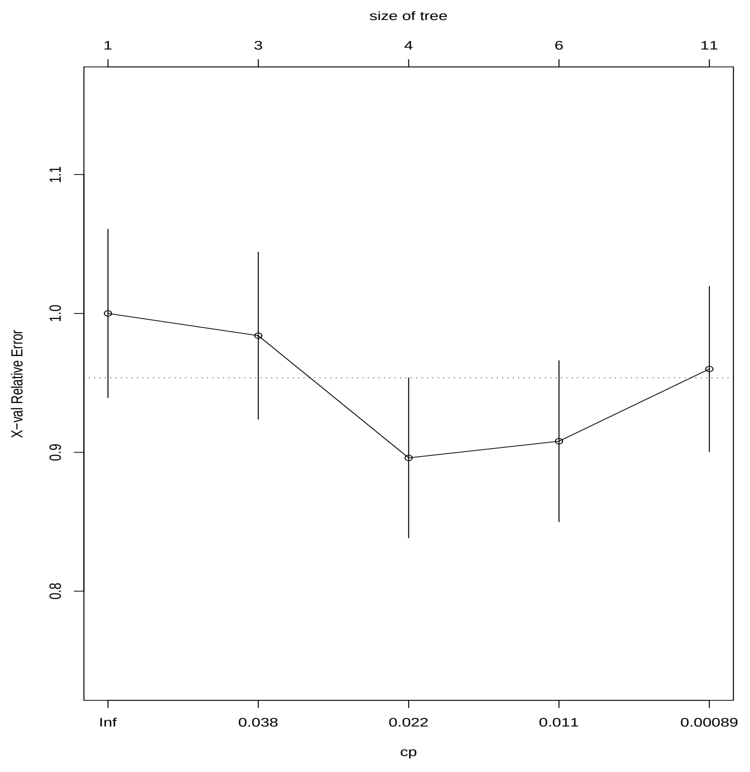


Figure 36: The "cp" plot of the `rpart` fit on the `head.injury` dataset starting with an initially very small value for `cp` so that the tree over fits (this is corrected by pruning with the one-standard deviation rule).

Tree-based classification and regression

Problem Solutions

Problem 1 (using `rpart`)

See the `R` code in `chap_11_prob_1.R`. When we run that code we get the plot shown in Figure 36. We see that the minimum cross-validated error is either for a tree of size 4. If we consider the tree of size 4 then the one-standard deviation rule: "choose the smallest tree who's error is less than the minimum error plus one standard deviation" this would pick the same tree size of 4.

Problem 2 (`rpart` on the `monica` dataset)

See the `R` code in `chap_11_prob_2.R`. When we run that code we first use `plotcp` to produce a tree that has too many splits (the tree overfits the data). When we run that code we get the plot shown in Figure 37 (left). From the produced plot we see that using the one-standard

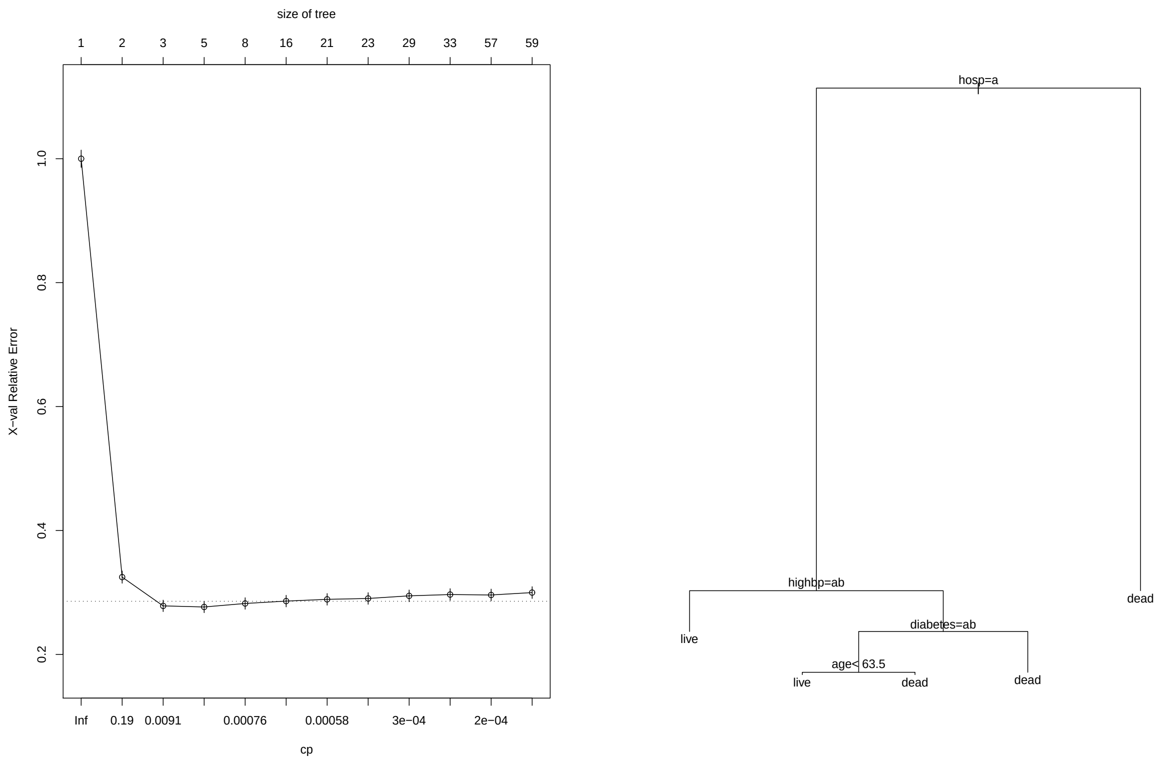


Figure 37: **Left:** The initial "cp" plot of the `rpart` fit on the `monica` dataset. The one-standard deviation rule is then used to select the tree size. The selected tree size using this rule has 3-5 nodes. **Right:** The resulting final tree.

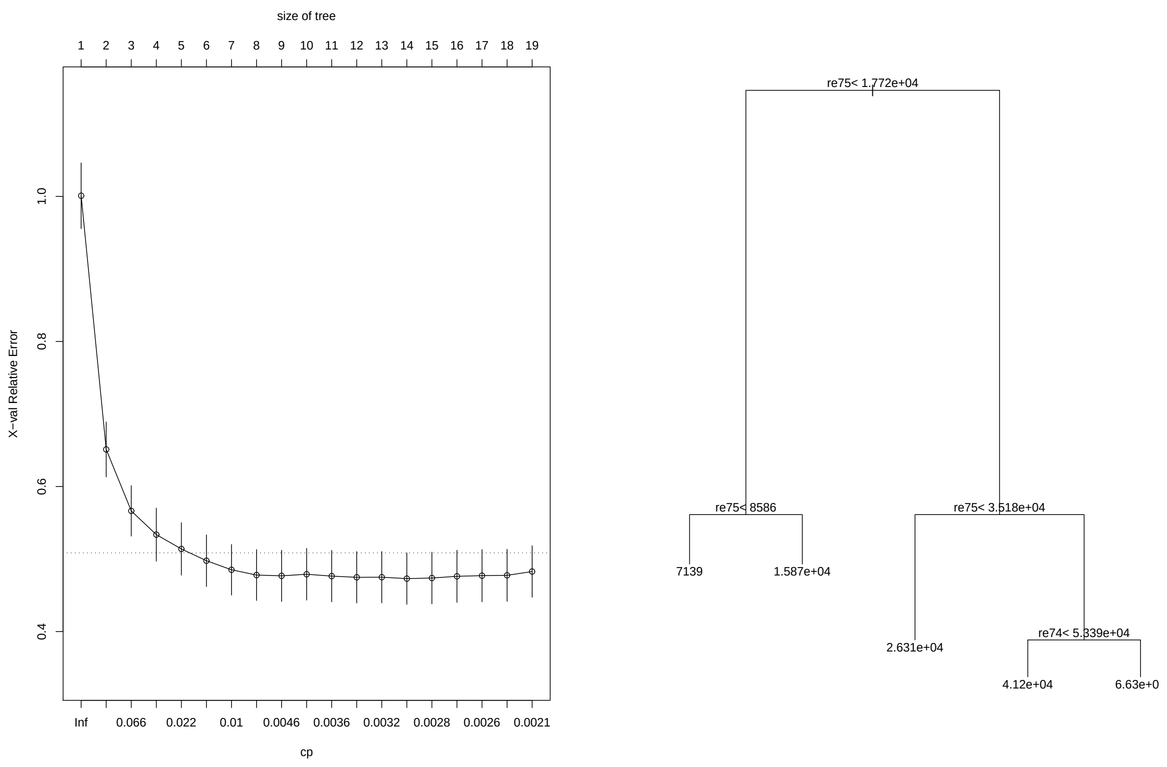


Figure 38: **Left:** The initial "cp" plot of the `rpart` fit on the `nswpsid1` dataset. The one-standard deviation rule is then used to select the tree size. The selected tree size using this rule has 6 nodes. **Right:** The resulting final tree.

deviation rule `plotcp` says to pick the tree of size 3-5. We use the command `prune` to extract this tree. Plotting the resulting tree gives the plot shown in Figure 37 (right).

Problem 3 (using `rpart` on the `nswpsid1` dataset)

See the `R` code in `chap_11_prob_3.R`. When we run that code we first use `plotcp` to produce a tree that has too many splits (the tree overfits the data). Using the one-standard deviation rule we would then want to select a tree of size 6 and $cp \approx 0.016$. When this script is run we get the two plots shown in Figure 38.

Problem 5 (running `randomForests`)

See the `R` code in `chap_11_prob_5.R`. When this script is run we get the plot shown in Figure 39. See the caption for that figure for additional explanation.

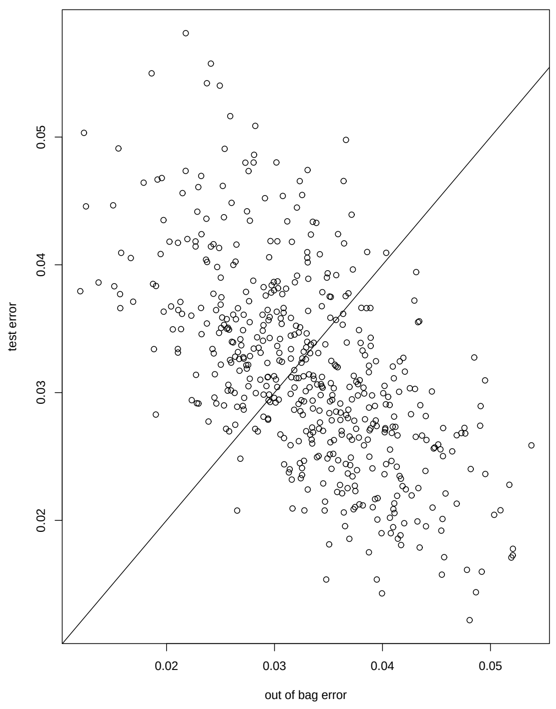


Figure 39: Plots of the test error rate vs. the out of bag (OOB) error rate when using a random forest to predict the `biops` data set. The line $y = x$ is also drawn. There does not seem to be a consistent difference between the OOB accuracy and the test accuracy.

Problem 6-8 (the expected number of duplicate elements)

See the `R` code in `chap_11_prob_6_to_8.R`.

Problem 9 (training on a smaller dataset)

See the `R` code in `chap_11_prob_9.R`. When this is run we get the following accuracy output for the two methods

```
> randomForest(as.factor(type) ~ ., data=Pima.tr)
        OOB estimate of  error rate: 27%
Confusion matrix:
     No Yes class.error
No  109  23   0.1742424
Yes  31  37   0.4558824
> randomForest(formula = as.factor(type) ~ ., data = Pima.tr[rowsamp,      ])
        OOB estimate of  error rate: 8.5%
Confusion matrix:
     No Yes class.error
No  126   8  0.05970149
Yes   9  57  0.13636364
```

The low classification accuracy is due to repeated elements in the rows sampled.