The Statistical Sleuth in R:
Chapter 1

Linda Loi   Ruobing Zhang   Kate Aloisio   Nicholas J. Horton*

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> install.packages('mosaic')               # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')               # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 1: Drawing Statistical Conclusions using R.

2 Motivation and Creativity

For Case Study 1: Motivation and Creativity, the following questions are posed: Do grading systems promote creativity in students? Do ranking systems and incentive awards increase productivity among employees? Do rewards and praise stimulate children to learn?

The data for Case Study 1 was collected by psychologist Teresa Amabile in an experiment concerning the effects of intrinsic and extrinsic motivation on creativity (page 2 of the *Sleuth*).

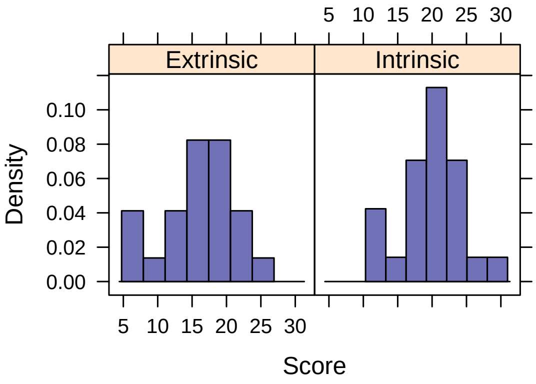
2.1 Statistical summary and graphical display

We begin by reading the data and summarizing the variables.

```
> summary(case0101)

     Score          Treatment
 Min.   : 5.0   Extrinsic:23
 1st Qu.:14.9   Intrinsic:24
 Median :18.7
 Mean   :17.9
 3rd Qu.:21.2
 Max.   :29.7
```

A total of 47 subjects with considerable experience in creative writing were randomly assigned to one of two treatment groups: 23 were placed into the "extrinsic" treatment group and 24 were placed into the "intrinsic" treatment group, as summarized in Display 1.1 (*Sleuth*, page 2)

```
> favstats(Score ~ Treatment, data=case0101)

Warning:  failed to assign NativeSymbolInfo for lhs since lhs is already defined in
the 'lazyeval' namespace
Warning:  failed to assign NativeSymbolInfo for rhs since rhs is already defined in
the 'lazyeval' namespace

  Treatment min   Q1 median   Q3  max mean   sd  n missing
1 Extrinsic   5 12.2   17.2 18.9 24.0 15.7 5.25 23       0
2 Intrinsic  12 17.4   20.4 22.3 29.7 19.9 4.44 24       0

> histogram(~ Score | Treatment, data=case0101)
```



```
> with(subset(case0101, Treatment=="Extrinsic"), stem(Score, scale=5))


  The decimal point is at the |

   5 | 04
   6 | 1
   7 |
   8 |
   9 |
  10 | 9
  11 | 8
  12 | 03
  13 |
  14 | 8
  15 | 0
  16 | 8
  17 | 2245
  18 | 577
  19 | 25
  20 | 7
  21 | 2
  22 | 1
  23 |
  24 | 0

> with(subset(case0101, Treatment=="Intrinsic"), stem(Score, scale=5))


  The decimal point is at the |

  12 | 009
  13 | 6
  14 |
  15 |
  16 | 6
  17 | 25
  18 | 2
  19 | 138
  20 | 356
  21 | 36
  22 | 126
  23 | 1
  24 | 03
  25 |
  26 | 7
  27 |
  28 |
  29 | 7
```

Similar output can be generated using the following code:

```
> maggregate(Score ˜ Treatment, data=case0101, FUN=stem)
```

The extrinsic group (n=23) has an average creativity score that is 4.1 points less than the intrinsic group (n=24). The extrinsic group has relatively larger spread than the intrinsic group (sd=5.25 for extrinsic group and sd=4.44 for intrinsic group). Both distributions are approximately normally distributed.

2.2 Inferential procedures (two-sample t-test)

```
> t.test(Score ~ Treatment, alternative="two.sided", data=case0101)


Welch Two Sample t-test

data:  Score by Treatment
t = -3, df = 40, p-value = 0.006
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.01 -1.28
sample estimates:
mean in group Extrinsic mean in group Intrinsic
                   15.7                    19.9
```

The two-sample t-test shows strong evidence that a subject would receive a lower creativity score for a poem written after the extrinsic motivation questionnaire than for one written after the intrinsic motivation questionnaire. The two-sided p-value is 0.006, which is small enough to reject the null hypothesis.

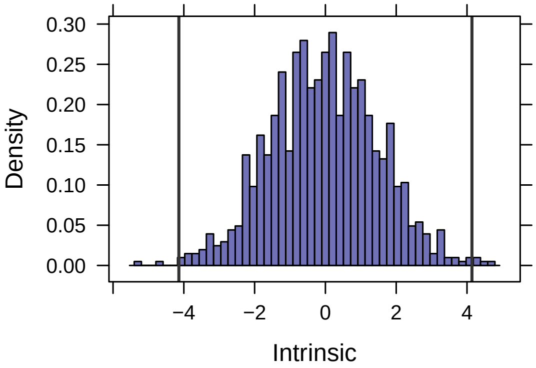
Thus, we can conclude that there is a difference between the population mean in the extrinsic group and the population mean in the intrinsic group; the estimated difference between these two scores is 4.1 points on the 0-40 point scale. A 95% confidence interval for the decrease in score due to having extrinsic motivation rather than intrinsic motivation is between -1.28 and -7.01 points (*Sleuth*, page 3).

```
> summary(lm(Score ~ Treatment, data=case0101))


Call:
lm(formula = Score ~ Treatment, data = case0101)

Residuals:
   Min     1Q Median     3Q    Max
-10.74  -2.98   1.06   2.96   9.82

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)           15.74       1.01   15.55   <2e-16 ***
TreatmentIntrinsic     4.14       1.42    2.93   0.0054 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.85 on 45 degrees of freedom
Multiple R-squared:  0.16,Adjusted R-squared:  0.141
F-statistic: 8.56 on 1 and 45 DF,  p-value: 0.00537
```

In the creativity study, the question of whether there is a treatment effect becomes a question of whether the parameter has a nonzero value. The value of the test statistic for the creativity scores is 4.14.

2.3 Permutation test

```
> diffmeans = diff(mean(Score ~ Treatment, data=case0101))
> diffmeans     # observed difference

Intrinsic
     4.14

> numsim = 1000      # set to a sufficient number
> nulldist = do(numsim)*diff(mean(Score~shuffle(Treatment), data=case0101))
> confint(nulldist)

Warning:  confint:  Using df=Inf.

       name lower upper level method estimate margin.of.error
1 Intrinsic -2.99  2.92  0.95 stderr     4.14            2.95

> # Display 1.8 Sleuth
> histogram(~ Intrinsic, nint=50, data=nulldist, v=c(-4.14,4.14))
```



As described in the *Sleuth* on page 12, if the group assignment changes, we will get different results. First, the test statistics will be just as likely to be negative as positive. Second, the majority of values fall in the range from -3.0 to +3.0. Third, only few of the 1,000 randomization produced test statistics as large as 4.14. This last point indicates that 4.14 is a value corresponding to an unusually uneven randomization outcome, if the null hypothesis is correct.

3 Gender Discrimination

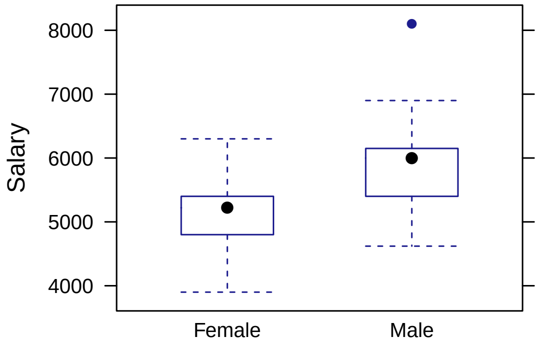
For Case Study 2: Gender Discrimination the following questions are posed: Did a bank discriminatorily pay higher starting salaries to men than to women? Display 1.3 (page 4 of the *Sleuth*) displays the beginning salaries for male and female skilled entry level clerical employees hired between 1969 and 1977.

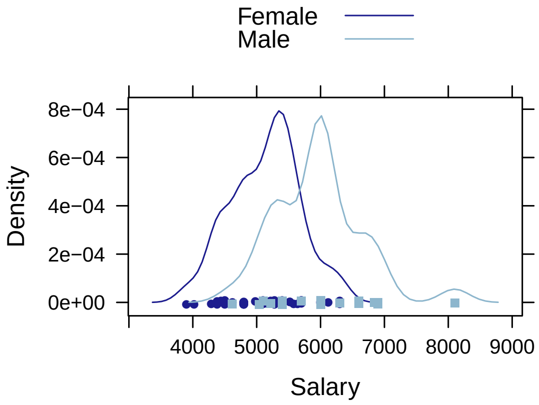
3.1 Statistical summary and graphical display

We begin by reading the data and summarizing the variables.

```
> summary(case0102) # Display 1.3 Sleuth p4

     Salary          Sex
 Min.   :3900   Female:61
 1st Qu.:4980   Male  :32
 Median :5400
 Mean   :5420
 3rd Qu.:6000
 Max.   :8100
```

```
> favstats(Salary ~ Sex, data=case0102)

     Sex  min   Q1 median   Q3  max mean  sd  n missing
1 Female 3900 4800   5220 5400 6300 5139 540 61       0
2   Male 4620 5400   6000 6075 8100 5957 691 32       0

> bwplot(Salary ~ Sex, data=case0102)
> densityplot(~ Salary, groups=Sex, auto.key=TRUE, data=case0102)
```

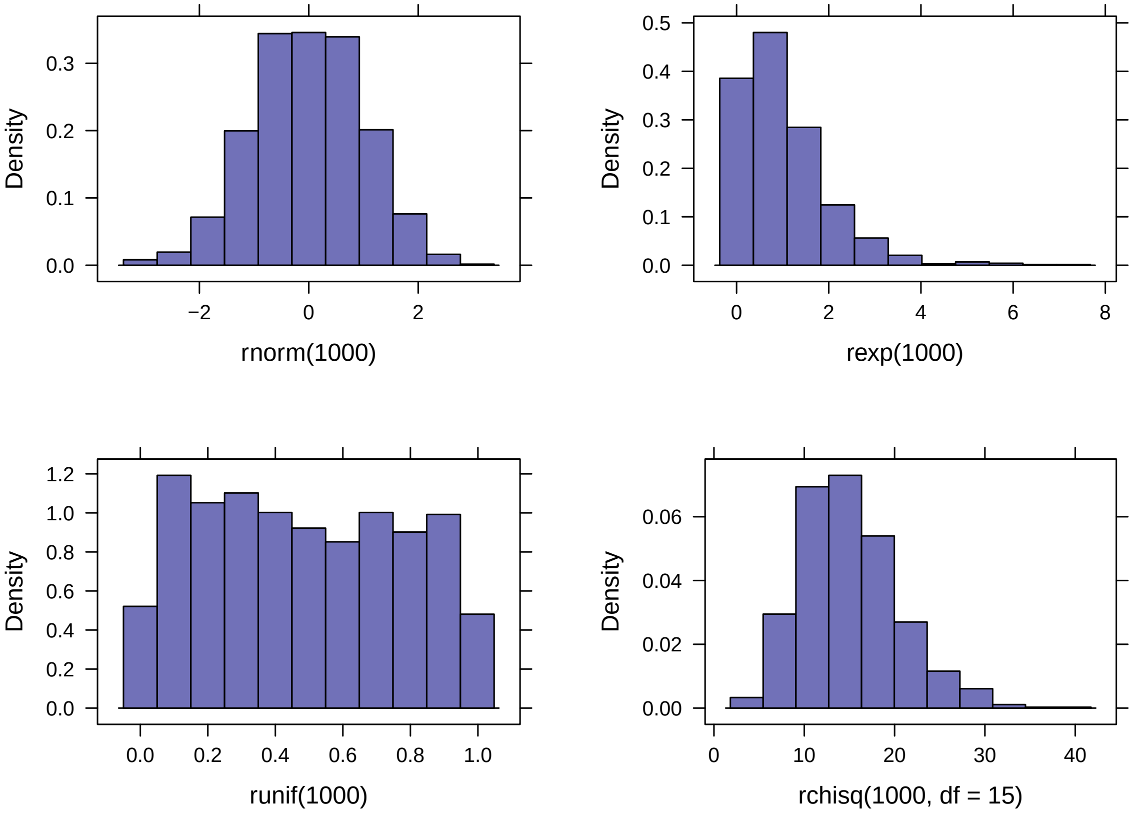




The 0 men have an average starting salary that is $818 more than the 61 women ($5957 vs $5139). Both distributions have similar spread (sd=$539.87 for women and sd=$690.73 for men) and distributions that are approximately normally distributed (see density plot). The key difference between the groups is the shift (as indicated by the parallel boxplots).

To show Display 1.13

```
> histogram(rnorm(1000))   # Normal
> histogram(rexp(1000))    # Long-tailed
> histogram(runif(1000))   # Short-tailed
> histogram(rchisq(1000, df=15)) # Skewed
```



3.2 Inferential procedures (two-sample t-test)

The *t*-test on page 4 of Sleuth can be replicated using the following commands (note that the equal-variance t-test is specified by `var.equal=TRUE` which is not the default).

```
> t.test(Salary ~ Sex, var.equal=TRUE, data=case0102)


Two Sample t-test

data:  Salary by Sex
t = -6, df = 90, p-value = 1e-08
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1076  -560
```

```
sample estimates:
mean in group Female   mean in group Male
                5139                 5957
```

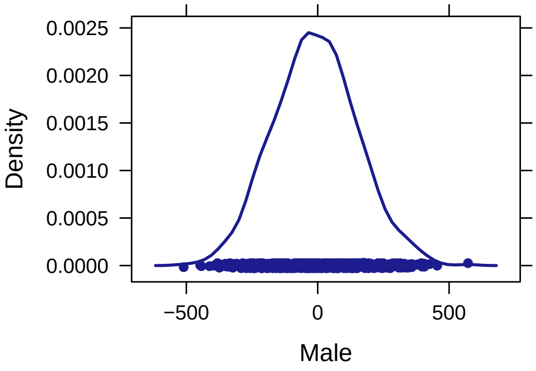
3.3 Permutation test

We undertake a permutation test to assess whether the differences in the center of these samples that we are observing are due to chance, if the distributions are actually equivalent back in the populations of male and female possible clerical hires. We start by calculating our test statistic (the difference in means) for the observed data, then simulate from the null distribution (where the labels can be interchanged) and calculate our p-value.

```
> obsdiff = diff(mean(Salary ~ Sex, data=case0102)); obsdiff

Male
 818
```

The labeling for the difference in means isn't ideal (but will be given as "Male" by R).

```
> numsim = 1000
> res = do(numsim) * diff(mean(Salary~shuffle(Sex), data=case0102))
> densityplot(~ Male, data=res)
> confint(res)

Warning:  confint:  Using df=Inf.

  name lower upper level method estimate margin.of.error
1 Male  -319   303  0.95 stderr      818             311
```



```
> larger = sum(with(res, abs(Male) >= abs(obsdiff)))
> larger
> pval = larger/numsim
> pval
```

Through the permutation test, we observe that the mean starting salary for males is significantly larger than the mean starting salary for females, as we never see a permuted difference in means close to our observed value. Therefore, we reject the null hypothesis ($p < 0.001$) and conclude that the salaries of the men are higher than that of the women back in the population.

```
> t.test(Salary ~ Sex, alternative="less", data=case0102)


Welch Two Sample t-test

data:  Salary by Sex
t = -6, df = 50, p-value = 2e-07
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf -583
sample estimates:
mean in group Female   mean in group Male
                5139                 5957
```

The p-value (< 0.001) from the two-sample t-test shows that the large difference between estimated salaries for males and females is unlikely to be due to chance.

The Statistical Sleuth in R:
Chapter 2

Linda Loi   Ruobing Zhang   Kate Aloisio   Nicholas J. Horton[*]

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

```
> install.packages('mosaic')               # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

[*] Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the Sleuth3 package.

```
> install.packages('Sleuth3')                    # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3, show.signif.stars=FALSE)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 2: Inference Using t-Distributions using R.

2 Evidence Supporting Darwin's Theory of Natural Selection

Do birds evolve to adapt to their environments? That's the question being addressed by Case Study 2.1 in the *Sleuth*.

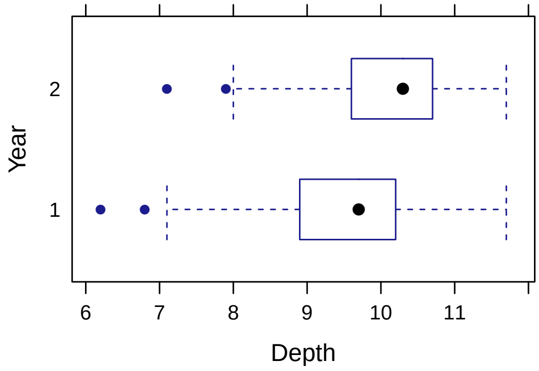
2.1 Statistical summary and graphical display

We begin by reading the data and summarizing the variables.

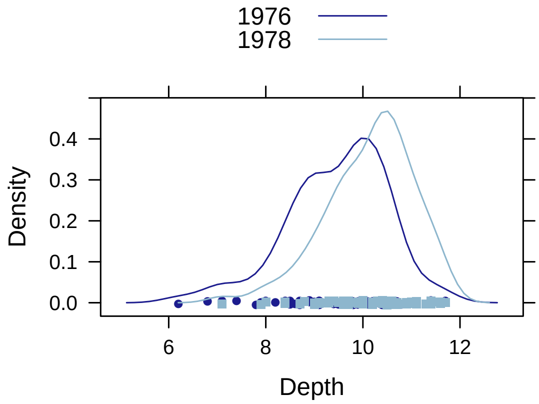
```
> summary(case0201)

      Year          Depth
 Min.   :1976   Min.   : 6.2
 1st Qu.:1976   1st Qu.: 9.1
 Median :1977   Median : 9.9
 Mean   :1977   Mean   : 9.8
 3rd Qu.:1978   3rd Qu.:10.5
 Max.   :1978   Max.   :11.7

> fav = favstats(Depth ~ Year, data=case0201); fav

  Year min  Q1 median   Q3  max  mean    sd  n missing
1 1976 6.2 8.9    9.7 10.2 11.7  9.47 1.035 89       0
2 1978 7.1 9.6   10.3 10.7 11.7 10.14 0.906 89       0
```

A total of 178 subjects are included in the data: 89 are finches that were caught in 1976 and 89 are finches that were caught in 1978. The following figure replicates Display 2.1 on page 30.

```
> bwplot(Year ~ Depth, data=case0201)
```



```
> densityplot(~ Depth, groups=Year, auto.key=TRUE, data=case0201)
```



The distributions are approximately normally distributed, with some evidence for a long left tail.

2.2 Inferential procedures (two-sample t-test)

First, we calculate the pooled SD and the standard error between these two different sample average (page 41, Display 2.8).

```
> # Calculate Pooled SD
> n1 = fav["1976", "n"]; n1

[1] NA

> n2 = fav["1978", "n"]; n2

[1] NA
```

```
> s1 = fav["1976", "sd"]; s1

[1] NA

> s2 = fav["1978", "sd"]; s2

[1] NA

> Sp = sqrt(((n1-1)*(s1)^2+(n2-1)*(s2)^2)/(n1+n2-2)); Sp

[1] NA

> # Calculate standard error
> SE = Sp*sqrt(1/n1+1/n2); SE

[1] NA
```

So the pooled SD is NA and the standard error is NA.

Based on this information, we can construct a 95% confidence interval (page 43, Display 2.9).

```
> Y1 = fav["1976", "mean"]; Y1

[1] NA

> Y2 = fav["1978", "mean"]; Y2

[1] NA

> Yd = Y2-Y1; Yd

[1] NA

> df = n1+n2-2; df

[1] NA

> qt = qt(0.975, df); qt

[1] NA

> hw = qt*SE; hw

[1] NA

> lower = Yd-hw; lower

[1] NA

> upper = Yd+hw; upper

[1] NA
```

So the 95% confidence interval of the difference between means is (NA, NA)

Now we want to calculate the t-statistic and p-value (as shown on page 46, Display 2.10).

```
> tstats = (Yd-0)/SE; tstats      # The hypothesis difference=0

[1] NA

> onepval = 1-pt(tstats, df); onepval

[1] NA

> twopval = 2*onepval; twopval

[1] NA
```

The one-sided p-value is approximately NA and the two-sided p-value is also approximately NA.

We can get the results of “Summary of Statistical Findings” (page 29) by using the following code:

```
> t.test(Depth ~ Year, var.equal=TRUE, data=case0201)


Two Sample t-test

data:  Depth by Year
t = -5, df = 200, p-value = 9e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.956 -0.381
sample estimates:
mean in group 1976 mean in group 1978
              9.47              10.14

> confint(lm(Depth ~ Year, data=case0201))

              2.5 %   97.5 %
(Intercept) -935.61 -366.488
Year           0.19    0.478
```

3 Anatomical Abnormalities Associated with Schizophrenia

Is the area of brain related to the development of schizophrenia? That’s the question being addressed by case study 2.2 in the *Sleuth*.

3.1 Statistical summary and graphical display

We begin by reading the data and summarizing the variables.

```
> summary(case0202)

   Unaffected       Affected    
 Min.   :1.25   Min.   :1.02  
 1st Qu.:1.60   1st Qu.:1.31  
 Median :1.77   Median :1.59  
 Mean   :1.76   Mean   :1.56  
 3rd Qu.:1.94   3rd Qu.:1.78  
 Max.   :2.08   Max.   :2.02  
```

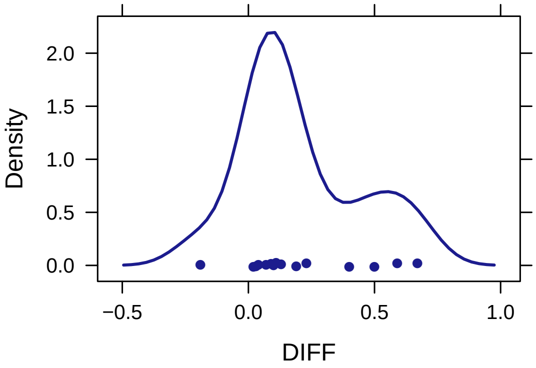
A total of 15 subjects are included in the data. There are 15 pairs of twins; one of the twins has schizophrenia, and the other does not. So there are 15 affected subjects and 15 unaffected subjects.

The difference in area of left hippocampus of these pairs of twins is:

```
> case0202 = transform(case0202, DIFF = Unaffected - Affected)
> favstats(~ DIFF, data=case0202)

   min    Q1 median    Q3  max  mean    sd  n missing
 -0.19 0.055   0.11 0.315 0.67 0.199 0.238 15       0
```

This matches the results on page 31, Display 2.2.

```
> densityplot(~ DIFF, data=case0202)
```



3.2 Inferential procedures (two-sample t-test)

We want to calculate the paired t-test and 95% confidence interval.

```
> # Calculate t-statistics
> difmean = mean(~ DIFF, data=case0202); difmean

[1] 0.199
```

```
> difsd = sd(~ DIFF, data=case0202); difsd

[1] 0.238

> difn = nrow(case0202); difn

[1] 15

> difSE = difsd/sqrt(difn); difSE

[1] 0.0615

> tscore = (difmean-0)/difSE; tscore         # hypothesis difference=0

[1] 3.23

> twopvalue = 2*(1-pt(tscore, difn-1)); twopvalue

[1] 0.00606

> # Construct confidence interval
> tstar = qt(0.975, difn-1); tstar

[1] 2.14

> schizolower = difmean - tstar*difSE; schizolower

[1] 0.0667

> schizoupper = difmean + tstar*difSE; schizoupper

[1] 0.331
```

So the two-sided p-value is approximately 0.006 and the 95% confidence interval is (0.07, 0.33). We can also get the results displayed on page 32 by conducting a paired t-test:

```
> with(case0202, t.test(Unaffected, Affected, paired=TRUE))

Warning in sub("^x$", deparse(x_lazy$expr), res$data.name):  argument 'replacement'
has length > 1 and only the first element will be used
Warning in sub("^x and y$", paste(deparse(x_lazy$expr), "and", deparse(y_lazy$expr)),
:  argument 'replacement' has length > 1 and only the first element will be used


Paired t-test

data:  c(1.94, 1.44, 1.56, 1.58, 2.06, 1.66, 1.75, 1.77, 1.78, 1.92,  and c(1.27, 1.63, 1.47,
t = 3, df = 10, p-value = 0.006
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.0667 0.3306
sample estimates:
mean of the differences
                  0.199
```

The Statistical Sleuth in R:
Chapter 3

Linda Loi   Ruobing Zhang   Kate Aloisio   Nicholas J. Horton*

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

[*] Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> install.packages('mosaic')               # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')               # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3, show.signif.stars=FALSE)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in *Sleuth* Chapter 3: A Closer Look at Assumptions using R.

2 Cloud Seeding to Increase Rainfall

Does seeding clouds lead to more rainfall? This is the question being addressed by case study 3.1 in the *Sleuth*.

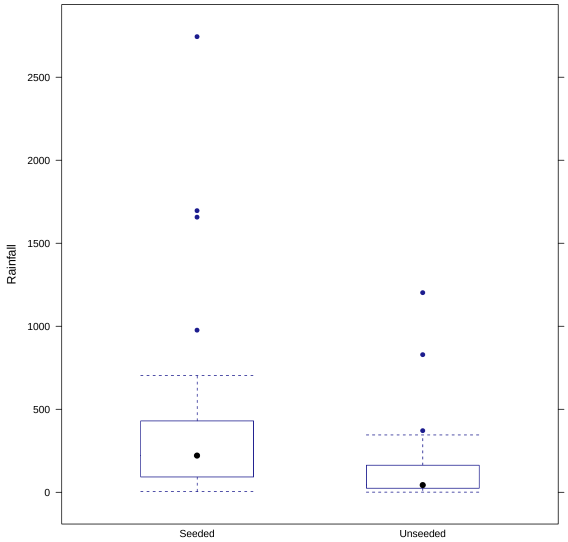
2.1 Summary statistics and graphical displays (untransformed)

We begin by reading the data and summarizing the variables.

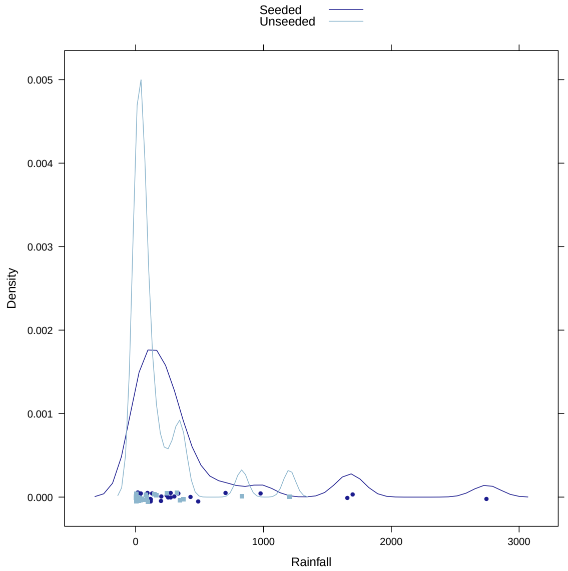
```
> summary(case0301)

    Rainfall        Treatment 
 Min.   :   1   Seeded  :26  
 1st Qu.:  29   Unseeded:26  
 Median : 117                
 Mean   : 303                
 3rd Qu.: 307                
 Max.   :2746                

> favstats(Rainfall ~ Treatment, data=case0301)

  Treatment min   Q1 median  Q3  max mean  sd  n missing
1    Seeded 4.1 98.1  221.6 406 2746  442 651 26       0
2  Unseeded 1.0 24.8   44.2 159 1203  165 278 26       0
```

A total of 52 subjects were included in this data: 26 seeded days and 26 unseeded days (Display 3.1, page 59).

```
> bwplot(Rainfall ~ Treatment, data=case0301)
```



```
> densityplot(~Rainfall, groups=Treatment, auto.key=TRUE, data=case0301)
```



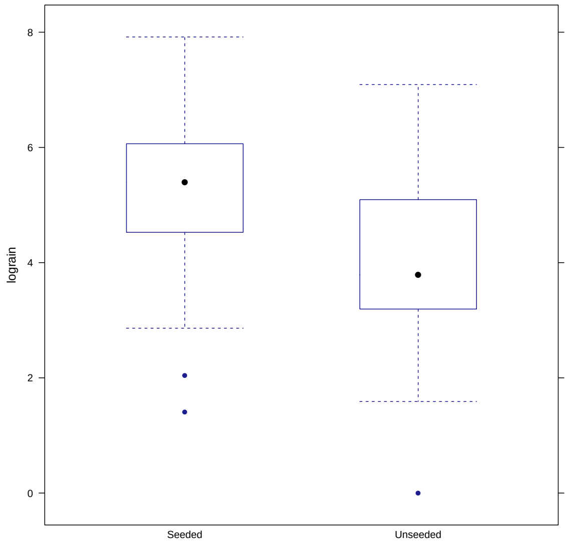
According to the boxplot and the density plot, the rainfall from seeded days seems to be larger than unseeded days. Both density curves are highly skewed to the right.

2.2 Summary statistics and graphical display (transformed)

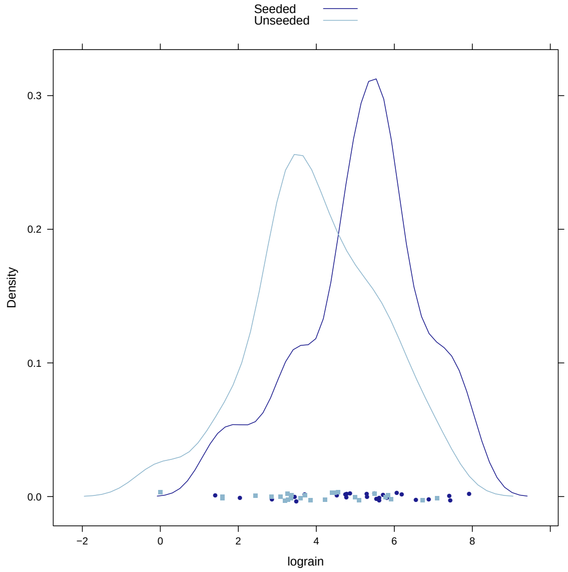
The skewness suggests that there is a need to apply a logarithmic transformation. The transformed data is shown on page 73 (Display 3.9).

```
> case0301 = transform(case0301, lograin=log(Rainfall))
> favstats(lograin ~ Treatment, data=case0301)

  Treatment  min   Q1 median   Q3  max mean   sd  n missing
1    Seeded 1.41 4.58   5.40 6.00 7.92 5.13 1.60 26       0
2  Unseeded 0.00 3.21   3.79 5.07 7.09 3.99 1.64 26       0
```

```
> bwplot(lograin ~ Treatment, data=case0301)
```



```
> densityplot(~lograin, groups=Treatment, auto.key=TRUE, data=case0301)
```



The log transformation reduces the skewness of these two distributions.

2.3 Inferential procedures (two-sample t-test)

```
> t.test(Rainfall ~ Treatment, var.equal=FALSE, data=case0301)


Welch Two Sample t-test

data:  Rainfall by Treatment
t = 2, df = 30, p-value = 0.05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  -4.76 559.56
sample estimates:
  mean in group Seeded mean in group Unseeded
                   442                    165

> t.test(Rainfall ~ Treatment, var.equal=TRUE, data=case0301)


Two Sample t-test

data:  Rainfall by Treatment
t = 2, df = 50, p-value = 0.05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  -1.43 556.22
sample estimates:
  mean in group Seeded mean in group Unseeded
                   442                    165
```

The following corresponds to the calculations on page 73.

```
> summary(lm(lograin ~ Treatment, data=case0301))


Call:
lm(formula = lograin ~ Treatment, data = case0301)

Residuals:
   Min     1Q Median     3Q    Max
-3.990 -0.745  0.162  1.019  3.102

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)          5.134      0.318   16.15   <2e-16
TreatmentUnseeded   -1.144      0.450   -2.54    0.014

Residual standard error: 1.62 on 50 degrees of freedom
Multiple R-squared:  0.115,Adjusted R-squared:  0.0969
F-statistic: 6.47 on 1 and 50 DF,  p-value: 0.0141

> ttestlog = t.test(lograin ~ Treatment, data=case0301); ttestlog


Welch Two Sample t-test

data:  lograin by Treatment
t = 3, df = 50, p-value = 0.01
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.241 2.047
sample estimates:
  mean in group Seeded mean in group Unseeded
                  5.13                   3.99
```

2.4 Interpretation of log model

The following code is used to calculate the “Statistical Conclusion” on page 59. First, we want to calculate the multiplier.

```
> obslogdiff = -diff(mean(lograin ~ Treatment, data=case0301)); obslogdiff
```

```
Unseeded
    1.14

> multiplier = exp(obslogdiff); multiplier

Unseeded
    3.14
```

Next we can calculate the 95% confidence interval for the multiplier.

```
> ttestlog$conf.int

[1] 0.241 2.047
attr(,"conf.level")
[1] 0.95

> exp(ttestlog$conf.int)

[1] 1.27 7.74
attr(,"conf.level")
[1] 0.95
```

3 Effects of Agent Orange on Troops in Vietnam

Is dioxin concentration related to veteran status? This is the question being addressed by case study 3.2 in the *Sleuth*.

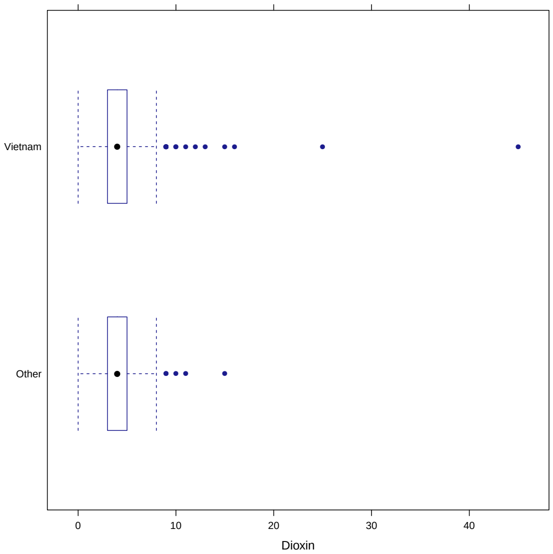
3.1 Summary statistics and graphical display

We begin by reading the data and summarizing the variables.

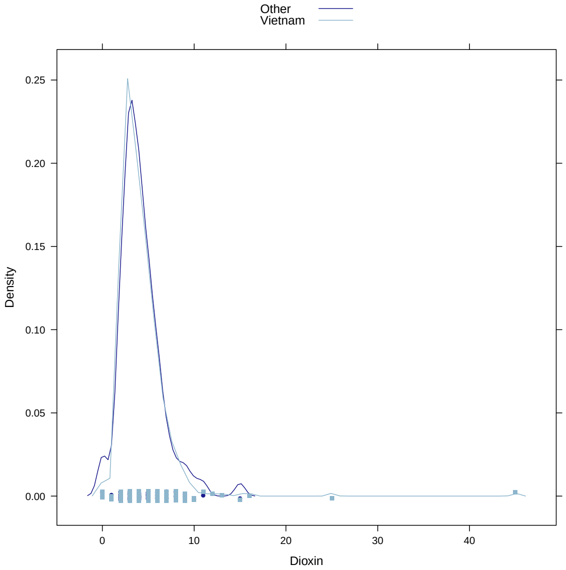
```
> summary(case0302)

     Dioxin        Veteran
 Min.   : 0.0   Other  : 97
 1st Qu.: 3.0   Vietnam:646
 Median : 4.0
 Mean   : 4.3
 3rd Qu.: 5.0
 Max.   :45.0

> favstats(Dioxin ˜ Veteran, data=case0302)

  Veteran min Q1 median Q3 max mean   sd   n missing
1   Other   0  3      4  5  15 4.19 2.30  97       0
2 Vietnam   0  3      4  5  45 4.26 2.64 646       0
```

A total of 743 veterans were included in this data: 646 served in Vietnam during 1967 and 1968 and 97 served in US or Germany during 1965 and 1971.

```
> bwplot(Veteran ~ Dioxin, data=case0302)
```



```
> densityplot(~Dioxin, groups=Veteran, auto.key=TRUE, data=case0302)
```



Both distributions are highly skewed to the right.

3.2 Inferential procedures (two-sample t-test)

The following code is used to calculate the “Statistical Conclusion” on page 62.

```
> t.test(Dioxin ~ Veteran, var.equal=TRUE, alternative="less", data=case0302)


Two Sample t-test

data:  Dioxin by Veteran
t = -0.3, df = 700, p-value = 0.4
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
  -Inf 0.392
sample estimates:
  mean in group Other mean in group Vietnam
                 4.19                  4.26

> t.test(Dioxin ~ Veteran, var.equal=TRUE, data=case0302)$conf.int

[1] -0.631  0.482
attr(,"conf.level")
[1] 0.95
```

So the one-sided p-value from a two-sample t-test is 0.396. The 95% confidence interval is (-0.63, 0.48). Notice that because of the way we ordered our variables, the confidence interval shown in our analysis is different from that of the book (our confidence intervals are inverse). This is of no consequence, as the difference between the groups is still the same.

3.3 Removing outliers

We will remove two extreme observations from the data. First we remove observation 646 and perform a t-test (Display 3.7, page 70).

```
> case0302.2 = case0302[-c(646), ]
> t.test(Dioxin ~ Veteran, alternative="less", data=case0302.2)


Welch Two Sample t-test

data:  Dioxin by Veteran
t = -0.05, df = 100, p-value = 0.5
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf  0.4
sample estimates:
  mean in group Other mean in group Vietnam
                 4.19                  4.20
```

Next we remove observations 645 and 646 and perform a t-test.

```
> dim(case0302)

[1] 743   2

> case0302.3 = case0302[-c(645, 646), ]
> dim(case0302.3)

[1] 741   2

> t.test(Dioxin ~ Veteran, alternative="less", data=case0302.3)


Welch Two Sample t-test

data:  Dioxin by Veteran
t = 0.09, df = 100, p-value = 0.5
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
  -Inf 0.429
sample estimates:
  mean in group Other mean in group Vietnam
                 4.19                  4.16
```

Notice that after removing these outliers, the *p*-value and the confidence interval have changed but the substantive conclusion is unchanged.

The Statistical Sleuth in R:
Chapter 4

Linda Loi   Ruobing Zhang   Kate Aloisio   Nicholas J. Horton[*]

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

```
> install.packages('mosaic')                    # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

[*] Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')                  # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3, show.signif.stars=FALSE)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 4: The Rank-Sum Test using R.

2 Space Shuttle O-Ring Failures

Does launch temperature tend to cause O-ring incidents? This is the question being addressed by case study 4.1 in the *Sleuth*.

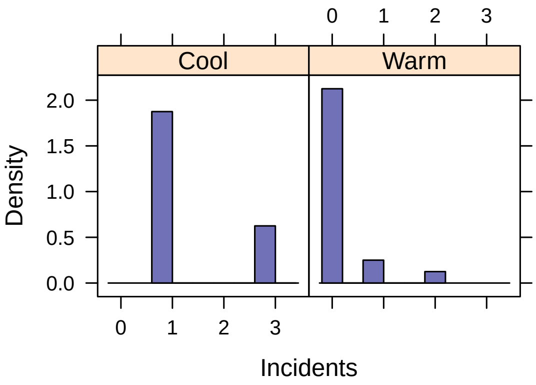
2.1 Summary statistics and graphical display

We begin by reading the data and summarizing the variables.

```
> summary(case0401)

   Incidents      Launch
 Min.   :0.000   Cool: 4
 1st Qu.:0.000   Warm:20
 Median :0.000
 Mean   :0.417
 3rd Qu.:1.000
 Max.   :3.000

> favstats(Incidents ~ Launch, data=case0401)

  Launch min Q1 median  Q3 max mean    sd  n missing
1   Cool   1  1      1 1.5   3  1.5 1.000  4       0
2   Warm   0  0      0 0.0   2  0.2 0.523 20       0
```

A total of 24 subjects are included in the data: 4 O-ring incidents when the temperature was cold and 20 incidents when the temperature was warm (Display 4.1, page 86).

```
> histogram(~ Incidents | Launch, data=case0401)
```



2.2 Permutation test on t-statistics

To replicate the permutation test performed on page 96 we use the following code, which first calculates the t-statistic of the observed outcome.

```
> t.test(Incidents ~ Launch, var.equal=TRUE, data=case0401)


Two Sample t-test

data:  Incidents by Launch
t = 4, df = 20, p-value = 8e-04
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.607 1.993
sample estimates:
mean in group Cool mean in group Warm
               1.5                0.2
```

We observe a test statistic of 3.888.

We want to get the total number of regroupings by calculating $C_{24,4}$.

```
> C244=factorial(24)/(factorial(4)*factorial(24-4)); C244

[1] 10626
```

There are a total of 1.063×10^4 regroupings with 8 possible (non-equiprobable) outcomes: (0, 0, 0, 0), (0, 0, 0, 1), (0, 0, 0, 2), (0, 0, 0, 3), (0, 0, 1, 1), (0, 0, 1, 2), (0, 0, 1, 3), (0, 0, 2, 3), (0, 1, 1, 1), (0, 1, 1, 2), (0, 1, 1, 3), (0, 1, 2, 3), (1, 1, 1, 1), (1, 1, 1, 2), (1, 1, 1, 3), (1, 1, 2, 3). Because the observed cold temperature outcomes was (1, 1, 1, 3), we will only examine the same or more extreme groupings, which are (1, 1, 2, 3) and (0, 1, 2, 3).

```
> # t.test for (1, 1, 2, 3)   # observations 1, 2, 4 and 24
> case0401$Incidents[c(1,2,4,24)]

[1] 1 1 3 2

> with(case0401, t.test(Incidents[c(1,2,4,24)], Incidents[-c(1,2,4,24)], var.equal=TRUE))


Two Sample t-test

data:  Incidents[c(1, 2, 4, 24)] and Incidents[-c(1, 2, 4, 24)]
t = 6, df = 20, p-value = 5e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.04 2.16
sample estimates:
mean of x mean of y
     1.75      0.15

> # t.test for (0, 1, 2, 3)   # observation 1, 4, 5 and 24
> case0401$Incidents[c(1,4,5,24)]

[1] 1 3 0 2

> with(case0401, t.test(Incidents[c(1,4,5,24)], Incidents[-c(1,4,5,24)], var.equal=TRUE))


Two Sample t-test

data:  Incidents[c(1, 4, 5, 24)] and Incidents[-c(1, 4, 5, 24)]
t = 4, df = 20, p-value = 8e-04
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.607 1.993
sample estimates:
mean of x mean of y
      1.5       0.2
```

The test statistic for (1, 1, 2, 3) is 5.952 and the test statistic for (0, 1, 2, 3) is 3.888.

We already know that the total number of regroupings is $C_{24,4}$=1.063×10^4. In order to calculate the p-value, we need to know the combinations of (1, 1, 1, 3), (2, 1, 2, 3) and (0, 1, 2, 3). There are 17 zeros, 5 ones, 1 two and 1 three.

For outcome (1, 1, 1, 3), we calculate C1113=$C_{5,3}$*$C_{1,1}$:

```
> C1113 = factorial(5)/(factorial(3)*factorial(5-3))*1; C1113

[1] 10
```

For outcome (1, 1, 2, 3), we calculate C1123=$C_{5,2}$*$C_{1,1}$*$C_{1,1}$:

```
> C1123 = factorial(5)/(factorial(2)*factorial(5-2))*1*1; C1123

[1] 10
```

For outcome (0, 1, 2, 3), we calculate C0123=$C_{17,1}$*$C_{5,1}$*$C_{1,1}$*$C_{1,1}$

```
> C0123 = 17*5*1*1; C0123

[1] 85
```

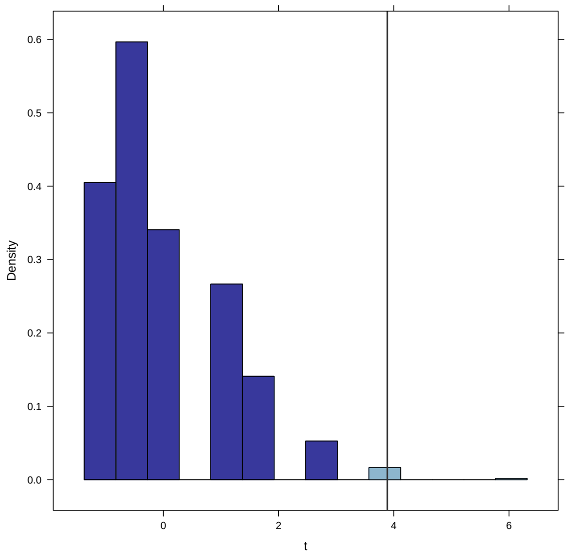
Now we can calculate the p-value as the proportion of the number of rearrangements that are as extreme or more extreme over the total number of rearrangements:

```
> onep = (C1113+C1123+C0123)/C244; onep

[1] 0.00988
```

The one-sided p-value from the permutation test on the t-statistic is 0.01.

Alternatively, we can approximate the p-value using the difference of means and simulating repeatedly from the null distribution (note that the book enumerates all of the possible outcomes to get an exact result).

```
> result = t.test(Incidents ~ Launch, var.equal=TRUE, data=case0401)$statistic; result

   t
3.89

> nulldist = do(10000)*t.test(Incidents ~ shuffle(Launch), var.equal=TRUE, data=case0401)$stati
> histogram(~ t, groups=t >= result, v=result, data=nulldist)
> tally(~ t >= result, format="proportion", data=nulldist)


  TRUE  FALSE
0.0101 0.9899
```



This simulation resulted in a p-value of 0.01.

3 Cognitive Load Theory in Teaching

Does use of modified instructional materials lead to quicker problem solving? That's the question being addressed by case study 4.2 in the *Sleuth*.

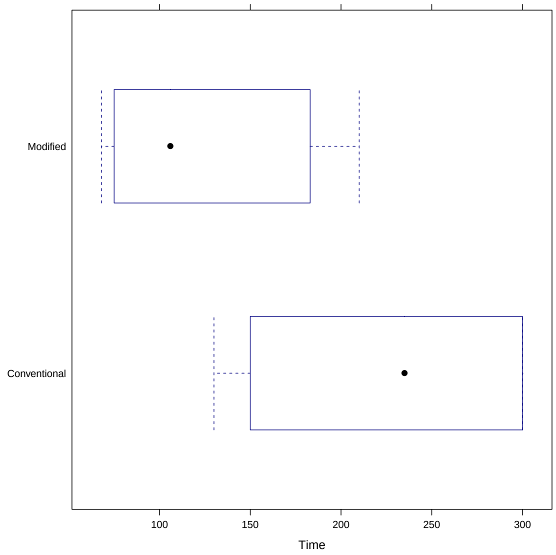
3.1 Summary statistics and graphical display

We begin by reading the data and summarizing the variables.

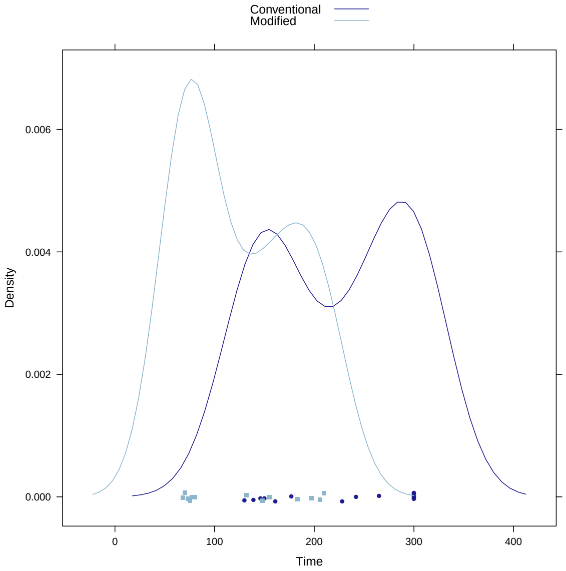
```
> summary(case0402)

      Time                Treatment       Censored    
 Min.   : 68   Conventional:14   Min.   :0.000  
 1st Qu.:118   Modified    :14   1st Qu.:0.000  
 Median :158                     Median :0.000  
 Mean   :175                     Mean   :0.179  
 3rd Qu.:232                     3rd Qu.:0.000  
 Max.   :300                     Max.   :1.000  

> favstats(Time ~ Treatment, data=case0402)

     Treatment min    Q1 median  Q3 max mean   sd  n missing
1 Conventional 130 152.8    235 300 300  224 70.5 14       0
2     Modified  68  75.5    106 176 210  125 56.6 14       0
```

A total of 28 subjects are included in the data: 14 students were assigned to modified instructional materials and 14 students were assigned to conventional materials.

```
> bwplot(Treatment ~ Time, data=case0402)
```



```
> densityplot(~ Time, groups=Treatment, auto.key=TRUE, data=case0402)
```



3.2 Rank-sum test

We can calculate the one-sided p-value by following rank-sum procedure. First, we try to find the statistic T (display 4.5, page 91):

```
> obsrank = rank(case0402$Time, ties.method="average"); obsrank

 [1]  1.0  2.0  3.0  4.0  5.0  6.5  6.5  9.0 12.0 14.0 17.0 18.0 19.0 20.0
[15]  8.0 10.0 11.0 13.0 15.0 16.0 21.0 22.0 23.0 26.0 26.0 26.0 26.0 26.0

> mt = sum(obsrank[1:14]); mt

[1] 137
```

Next we calculate the p-value using a normal approximation (Display 4.7, page 93).

```
> average = mean(obsrank); average

[1] 14.5

> sd = sd(obsrank); sd

[1] 8.2

> n = nrow(subset(case0402, Treatment=="Modified")); n

[1] 14

> MEANT = n * average; MEANT

[1] 203

> SDT = sd * sqrt((n^2)/(2*n)); SDT

[1] 21.7

> z = (mt-MEANT)/SDT; z

[1] -3.04

> p = pnorm(-abs(z)); p

[1] 0.00118
```

The one-sided p-value is 0.001.

Alternatively, we can use following code to calculate the Wilcoxon rank-sum test:

```
> wilcox.test(Time ~ Treatment, conf.int=TRUE, exact=TRUE, data=case0402)

Warning in wilcox.test.default(x = c(130L, 139L, 146L, 150L, 161L, 177L, :  cannot
compute exact p-value with ties
Warning in wilcox.test.default(x = c(130L, 139L, 146L, 150L, 161L, 177L, :  cannot
compute exact confidence intervals with ties
```

```
Wilcoxon rank sum test with continuity correction

data:  Time by Treatment
W = 200, p-value = 0.003
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
  57 160
sample estimates:
difference in location
                    94
```

So the one-sided *p*-value is 0.001. The 95% confidence interval is (57, 160). The book suggests that the 95% confidence interval should be (58, 159), which is slightly narrower than these results. Their procedure is on page 94.

The Statistical Sleuth in R:
Chapter 5

Linda Loi Kate Aloisio Ruobing Zhang Nicholas J. Horton*

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

```
> install.packages('mosaic')               # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')               # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme
> options(digits=3)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 5: Comparisons Among Several Samples using R.

2 Diet and lifespan

Does restricting the diet of female mice lead to increased lifespan? This is the question addressed in case study 5.1 in the *Sleuth*.

2.1 Summary statistics and graphical display

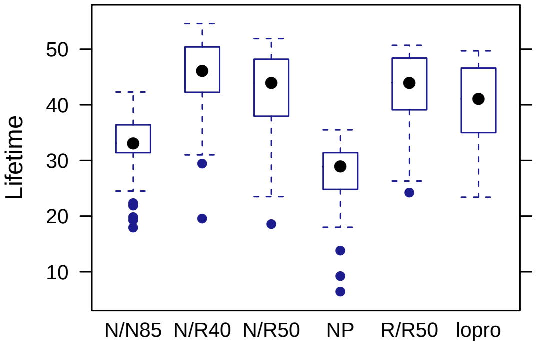
We begin by reading the data and summarizing the variables.

```
> summary(case0501)

    Lifetime       Diet
 Min.   : 6.4   N/N85:57
 1st Qu.:31.8   N/R40:60
 Median :39.5   N/R50:71
 Mean   :38.8   NP   :49
 3rd Qu.:46.9   R/R50:56
 Max.   :54.6   lopro:56

> favstats(Lifetime ~ Diet, data=case0501)
```

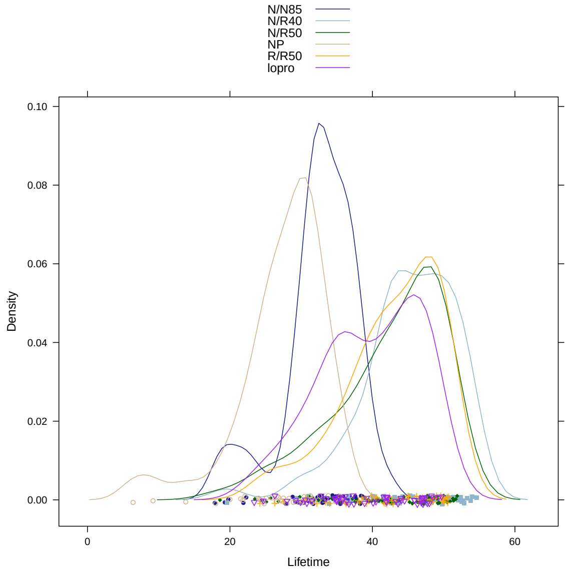
```
   Diet  min   Q1 median   Q3  max mean   sd  n missing
1 N/N85 17.9 31.4   33.1 36.4 42.3 32.7 5.13 57       0
2 N/R40 19.6 42.3   46.0 50.3 54.6 45.1 6.70 60       0
3 N/R50 18.6 38.0   43.9 48.2 51.9 42.3 7.77 71       0
4    NP  6.4 24.8   28.9 31.4 35.5 27.4 6.13 49       0
5 R/R50 24.2 39.2   44.0 48.3 50.7 42.9 6.68 56       0
6 lopro 23.4 35.0   41.0 46.4 49.7 39.7 6.99 56       0
```

There were a total of 349 female mice. These mice were randomly assigned to one of 6 diets. Their lifetimes were then recorded, as shown in Display 5.2 (page 115 of the *Sleuth*).

```
> bwplot(Lifetime ~ Diet, data=case0501) # Display 5.1
```



```
> densityplot(~ Lifetime, groups=Diet, auto.key=TRUE, data=case0501)
```



2.2 One-way ANOVA

First we fit the one way analysis of variance (ANOVA) model, using all of the groups.

```
> anova(lm(Lifetime ~ Diet, data=case0501))

Analysis of Variance Table

Response: Lifetime
           Df Sum Sq Mean Sq F value Pr(>F)
Diet        5  12734    2547    57.1 <2e-16
Residuals 343  15297      45
```

There is a strong statistically significant difference between the diets.

By default, the use of the linear model (regression) function displays the pairwise differences between the first group and each of the other groups. Note that the overall test of the model is the same.

```
> summary(lm(Lifetime ~ Diet, data=case0501))


Call:
lm(formula = Lifetime ~ Diet, data = case0501)

Residuals:
    Min      1Q  Median      3Q     Max
-25.517  -3.386   0.814   5.183  10.014

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   32.691      0.885   36.96  < 2e-16
DietN/R40     12.425      1.235   10.06  < 2e-16
DietN/R50      9.606      1.188    8.09  1.1e-14
DietNP        -5.289      1.301   -4.07  5.9e-05
DietR/R50     10.194      1.257    8.11  8.9e-15
Dietlopro      6.994      1.257    5.57  5.2e-08

Residual standard error: 6.68 on 343 degrees of freedom
Multiple R-squared:  0.454,Adjusted R-squared:  0.446
F-statistic: 57.1 on 5 and 343 DF,  p-value: <2e-16
```

The reference group is *NP*, followed by *N/N85, lopro, N/R50, R/R50, N/R40.*

2.3 Pairwise comparisons

Next we used contrasts for the results on page 122, Display 5.7, and part **(a)** on page 115:

```
> require(gmodels)

Loading required package:  gmodels

> # N/N85 vs N/R50
> fit.contrast(lm(Lifetime ~ Diet, data=case0501), "Diet", c(-1, 0, 1, 0, 0, 0), conf.int=0.95

                        Estimate Std. Error t value Pr(>|t|) lower CI
Diet c=( -1 0 1 0 0 0 )     9.61       1.19    8.09 1.06e-14     7.27
                        upper CI
Diet c=( -1 0 1 0 0 0 )     11.9
```

The results for **(b)** on page 115-116:

```
> # N/R50 vs R/R50 (b)
> fit.contrast(lm(Lifetime ~ Diet, data=case0501), "Diet", c(0, 0, -1, 0, 1, 0), conf.int=0.95

                        Estimate Std. Error t value Pr(>|t|) lower CI
Diet c=( 0 0 -1 0 1 0 )    0.589       1.19   0.493    0.622    -1.76
                        upper CI
Diet c=( 0 0 -1 0 1 0 )     2.94
```

The results for **(c)** on page 116:

```
> # N/R40 vs N/R50 (c)
> fit.contrast(lm(Lifetime ~ Diet, data=case0501), "Diet", c(0, -1, 1, 0, 0, 0), conf.int=0.95

                        Estimate Std. Error t value Pr(>|t|) lower CI
Diet c=( 0 -1 1 0 0 0 )    -2.82       1.17   -2.41   0.0166    -5.12
                        upper CI
Diet c=( 0 -1 1 0 0 0 )   -0.516

> # N/N85 vs N/R40
> fit.contrast(lm(Lifetime ~ Diet, data=case0501), "Diet", c(-1, 1, 0, 0, 0, 0), conf.int=0.95

                        Estimate Std. Error t value Pr(>|t|) lower CI
Diet c=( -1 1 0 0 0 0 )     12.4       1.24    10.1 4.96e-21       10
                        upper CI
Diet c=( -1 1 0 0 0 0 )     14.9
```

The results for **(d)** on page 116:

```
> # N/R50 vs N/R50 lopro (d)
> fit.contrast(lm(Lifetime ~ Diet, data=case0501), "Diet", c(0, 0, -1, 0, 0, 1), conf.int=0.95

                        Estimate Std. Error t value Pr(>|t|) lower CI
Diet c=( 0 0 -1 0 0 1 )    -2.61       1.19   -2.19   0.0293    -4.96
```

```
                        upper CI
Diet c=( 0 0 -1 0 0 1 )   -0.264
```

The results for **(e)** on page 116:

```
> # N/N85 vs NP (e)
> fit.contrast(lm(Lifetime ~ Diet, data=case0501), "Diet", c(-1, 0, 0, 1, 0, 0), conf.int=0.95

                        Estimate Std. Error t value Pr(>|t|) lower CI
Diet c=( -1 0 0 1 0 0 )    -5.29        1.3   -4.07 5.95e-05    -7.85
                        upper CI
Diet c=( -1 0 0 1 0 0 )    -2.73
```

Another way of viewing these results is through a model table, which displays the differences between the grand mean and the group means.

```
> model.tables(aov(lm(Lifetime ~ Diet, data=case0501)))

Tables of effects

 Diet
    N/N85 N/R40 N/R50    NP  R/R50   lopro
   -6.106  6.32   3.5 -11.4  4.089  0.8886
rep 57.000 60.00  71.0  49.0 56.000 56.0000
```

Another way of calculating the above results is done with the following code:

```
> mean(Lifetime ~ Diet, data=case0501)-mean(~ Lifetime, data=case0501)

  N/N85   N/R40   N/R50      NP   R/R50   lopro
 -6.106   6.320   3.500 -11.395   4.089   0.889
```

2.4 Other analyses

We will next demonstrate how to calculate the quantities on 121 (Display 5.6).

```
> df = length(case0501$Diet) - length(unique(case0501$Diet)); df

[1] 343

> sdvals = with(case0501, tapply(Lifetime, Diet, sd)); sdvals

N/N85 N/R40 N/R50    NP R/R50 lopro
 5.13  6.70  7.77  6.13  6.68  6.99

> nvals = with(case0501, tapply(Lifetime, Diet, length)); nvals
```

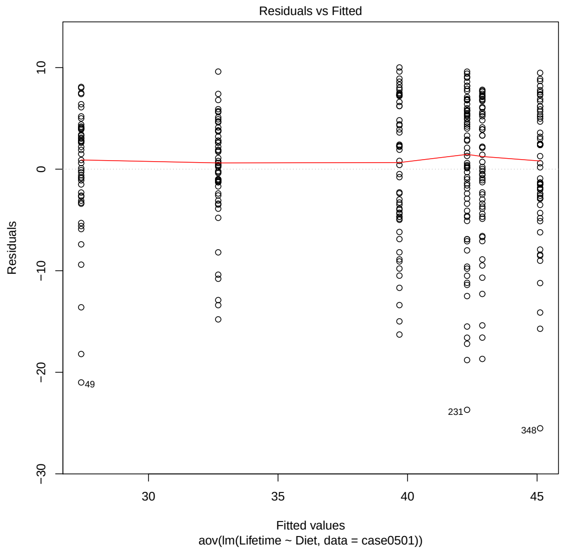
```
N/N85 N/R40 N/R50    NP  R/R50 lopro
   57    60    71    49    56    56

> pooledsd = sum(sdvals*nvals)/sum(nvals); pooledsd

[1] 6.63
```

Note that the pooled standard deviation reported in chapter 5 is not the same as the root MSE from the ANOVA. For the rest of this document we will use the ANOVA estimate of the root mean squared error.

2.5 Residual analysis and diagnostics

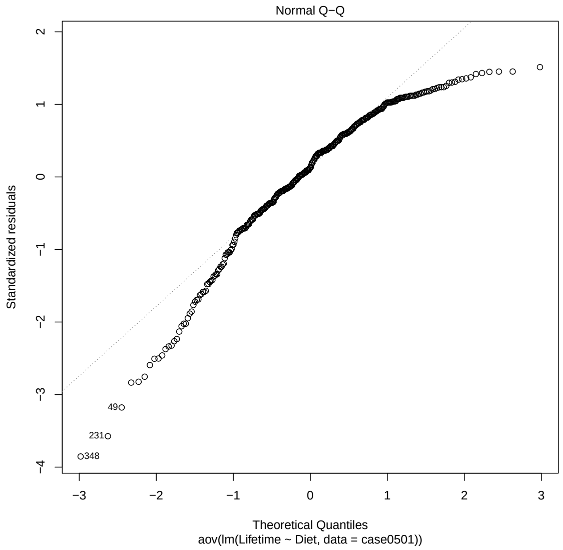
The residuals versus fitted graph does not demonstrate dramatic lack of fit (though some of the mice had very small residuals). The following figure is akin to Display 5.14 (page 132).

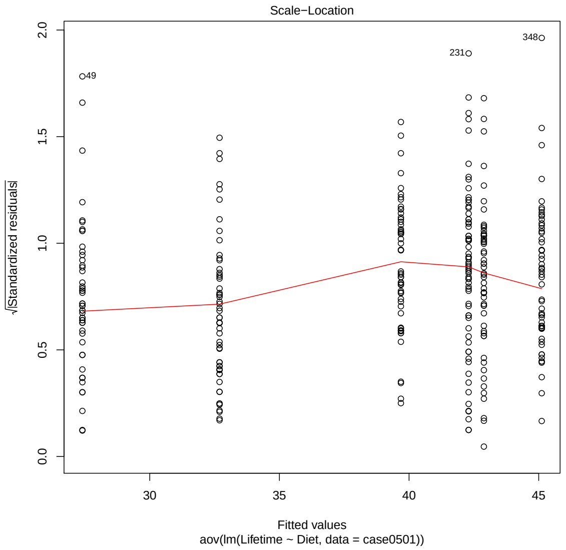
```
> aov1 = aov(lm(Lifetime ~ Diet, data=case0501))
> plot(aov1, which=1)
```



The quantile plot of the residuals indicates that the normality assumption may be violated.

```
> plot(aov1, which=2)
> plot(aov1, which=3)
```





3 Spock Conspiracy Trial

Did Dr. Benjamin Spock have a fair trial? More specifically, were women underrepresented on his jury pool? This is the question considered in case study 5.2 in the *Sleuth*.

3.1 Summary statistics and graphical display

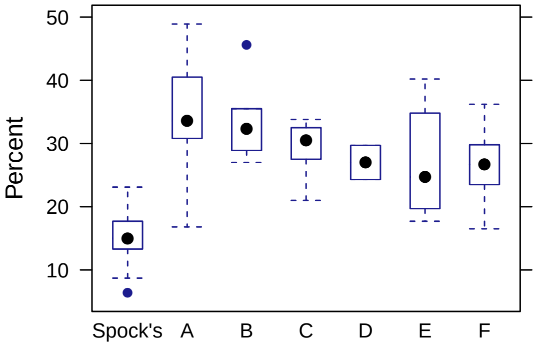
We begin by reading the data and summarizing the variables.

```
> case0502 = transform(case0502, Judge = factor(Judge, levels = c("Spock's", "A", "B", "C", "D
> summary(case0502)

    Percent          Judge
 Min.   : 6.4   Spock's:9
 1st Qu.:19.9   A      :5
 Median :27.5   B      :6
 Mean   :26.6   C      :9
 3rd Qu.:32.4   D      :2
 Max.   :48.9   E      :6
                F      :9

> case0502$Judge = with(case0502, as.factor(Judge))
> favstats(Percent ~ Judge, data=case0502)

    Judge  min   Q1 median   Q3  max mean    sd n missing
1 Spock's  6.4 13.3   15.0 17.7 23.1 14.6  5.04 9       0
2       A 16.8 30.8   33.6 40.5 48.9 34.1 11.94 5       0
3       B 27.0 29.7   32.4 34.8 45.6 33.6  6.58 6       0
```

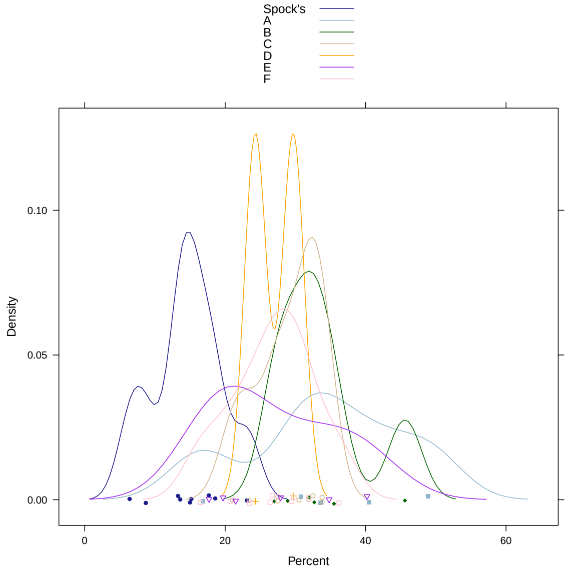
```
4        C 21.0 27.5   30.5 32.5 33.8 29.1  4.59 9        0
5        D 24.3 25.7   27.0 28.3 29.7 27.0  3.82 2        0
6        E 17.7 20.1   24.7 33.1 40.2 27.0  9.01 6        0
7        F 16.5 23.5   26.7 29.8 36.2 26.8  5.97 9        0
```

There were a total of 46 venires. They compared Spock's judge with 6 other judges. The precent of women within each venire was recorded as shown in Display 5.4 (page 117 of the *Sleuth*).

```
> bwplot(Percent ~ Judge, data=case0502) # Display 5.5 (page 118)
```



```
> densityplot(~ Percent, groups=Judge, auto.key=TRUE, data=case0502)
```



3.2 One-way ANOVA

First we fit the one way analysis of variance (ANOVA) model, with all of the groups. These results are summarized on page 118 and shown in Display 5.10 (page 127).

```
> aov1 = anova(lm(Percent ~ Judge, data=case0502)); aov1

Analysis of Variance Table

Response: Percent
          Df Sum Sq Mean Sq F value  Pr(>F)
Judge      6   1927     321    6.72 6.1e-05
Residuals 39   1864      48
```

By default, the use of the linear model (regression) function displays the pairwise differences between the first group and each of the other groups. Note that the overall test of the model is the same.

```
> summary(lm(Percent ~ Judge, data=case0502))


Call:
lm(formula = Percent ~ Judge, data = case0502)

Residuals:
   Min     1Q Median     3Q    Max
-17.32  -4.37  -0.25   3.32  14.78

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    14.62       2.30    6.34  1.7e-07
JudgeA         19.50       3.86    5.06  1.1e-05
JudgeB         18.99       3.64    5.21  6.4e-06
JudgeC         14.48       3.26    4.44  7.2e-05
JudgeD         12.38       5.41    2.29   0.0275
JudgeE         12.34       3.64    3.39   0.0016
JudgeF         12.18       3.26    3.74   0.0006

Residual standard error: 6.91 on 39 degrees of freedom
Multiple R-squared:  0.508,Adjusted R-squared:  0.433
F-statistic: 6.72 on 6 and 39 DF,  p-value: 6.1e-05
```

```
> model.tables(aov(lm(Percent ~ Judge, data=case0502)))

Tables of effects

 Judge
    Spock's     A     B     C      D      E      F
     -11.96 7.537 7.034 2.517 0.4174 0.3841 0.2174
rep    9.00 5.000 6.000 9.000 2.0000 6.0000 9.0000
```

Then we can fit the one way analysis of variance F-test of whether the mean percentage is the same for judges A-F (page 118).

```
> with(subset(case0502, Judge!="Spock's"), anova(lm(Percent ~ Judge)))

Analysis of Variance Table

Response: Percent
          Df Sum Sq Mean Sq F value Pr(>F)
Judge      5    326    65.3    1.22   0.32
Residuals 31   1661    53.6
```

3.3 Additional analyses

Now we will demonstrate how to fit the reduced model comparing Spock's judge to a combination of the other judges. First we create a 2 level version of the grouping variable.

```
> case0502$twoJudge = as.character(case0502$Judge)
> case0502$twoJudge[case0502$Judge!="Spock's"] = "notspock"
> tally(twoJudge ~ Judge, format="count", data=case0502)

          Judge
twoJudge   Spock's A B C D E F
  Spock's        9 0 0 0 0 0 0
  notspock       0 5 6 9 2 6 9
```

Recall that the book calculates the extra sum of squares as (2,190.90 - 1864.45)/(44-39)) / (1864.45 / 39) = 1.37, with df 5 and 39. P(F > 1.366) = 0.26 (page 130). Below are the calculations for the results found on page 128.

```
> numdf1 = aov1["Residuals", "Df"]; numdf1 # Within

[1] 39

> ss1 = aov1["Residuals", "Sum Sq"]; ss1 # Within

[1] 1864

> aov2 = anova(lm(Percent ~ as.factor(twoJudge), data=case0502)); aov2

Analysis of Variance Table

Response: Percent
                    Df Sum Sq Mean Sq F value Pr(>F)
as.factor(twoJudge)  1   1601    1601    32.1  1e-06
Residuals           44   2191      50
```

```
> df2 = aov2["Residuals", "Df"]; df2 # Spock and others

[1] 44

> ss2 = aov2["Residuals", "Sum Sq"]; ss2 # Spock and others

[1] 2191

> Fstat = ((ss2 - ss1)/(df2 - numdf1)) / (ss1 / numdf1); Fstat

[1] 1.37

> 1-pf(Fstat, length(levels(case0502$Judge))-2, numdf1)

[1] 0.258
```

We can also compare the two models using ANOVA (Display 5.12, page 130).

```
> anova(lm(Percent ~ as.factor(Judge), data=case0502), lm(Percent ~ as.factor(twoJudge), data=c

Analysis of Variance Table

Model 1: Percent ~ as.factor(Judge)
Model 2: Percent ~ as.factor(twoJudge)
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     39 1864
2     44 2191 -5      -326 1.37   0.26
```

There are some other ways to compare whether the other judges differ from Dr. Spock's judge in their female composition using contrasts.

```
> # test all of the other judges vs. Spock's judge using a contrast page 118
> fit.contrast(lm(Percent ~ Judge, data=case0502), "Judge", c(-6, 1, 1, 1, 1, 1, 1), conf.int=0

                           Estimate Std. Error t value Pr(>|t|) lower CI
Judge c=( -6 1 1 1 1 1 1 )     89.9       15.9    5.67 1.49e-06     57.8
                           upper CI
Judge c=( -6 1 1 1 1 1 1 )      122

> # calculate the 95% confidence interval for Dr. Spock's jury female composition page 118
> estimable(lm(Percent ~ Judge, data=case0502), c(1,0,0,0,0,0,0), conf.int=0.95)

                Estimate Std. Error t value DF Pr(>|t|) Lower.CI Upper.CI
(1 0 0 0 0 0 0)     14.6        2.3    6.34 39 1.72e-07     9.96     19.3
```

3.3.1 Kruskal-Wallis Nonparametric Analysis of Variance

For the results of the Kruskal-Wallis test on page 136 we can use the following code:

```
> kruskal.test(Percent ~ Judge, data=case0502)


Kruskal-Wallis rank sum test

data:  Percent by Judge
Kruskal-Wallis chi-squared = 20, df = 6, p-value = 0.001
```

The Statistical Sleuth in R:
Chapter 6

Linda Loi Ruobing Zhang Kate Aloisio Nicholas J. Horton[*]

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated knitr reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the mosaic package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

[*] Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> install.packages('mosaic')                # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')               # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 6: Linear Combinations and Multiple Comparisons of Means using R.

2 Discrimination Against the Handicapped

Do equivalent candidates with the same qualifications but different disabilities get treated differentially? This is the question addressed in case study 6.1 in the *Sleuth*.

2.1 Summary statistics and graphical display

We begin by reading the data and summarizing the variables.

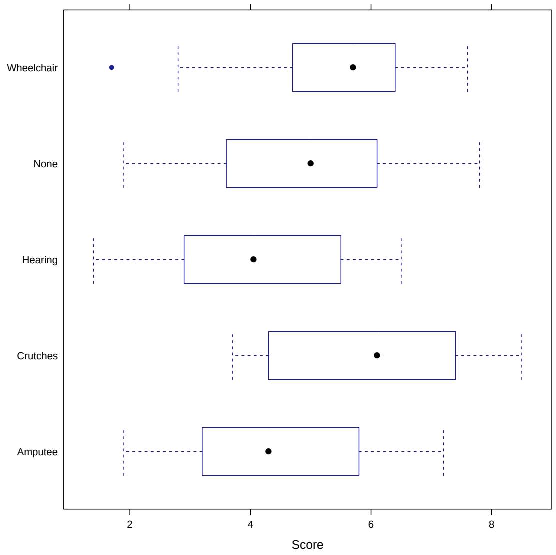
```
> case0601$Handicap = relevel(case0601$Handicap, ref="Amputee")
> summary(case0601)

     Score              Handicap
 Min.   :1.40   Amputee   :14
 1st Qu.:3.70   Crutches  :14
 Median :5.05   Hearing   :14
 Mean   :4.93   None      :14
 3rd Qu.:6.10   Wheelchair:14
 Max.   :8.50

> favstats(Score ~ Handicap, data=case0601)
```

```
    Handicap min   Q1 median   Q3 max mean   sd  n missing
1    Amputee 1.9 3.30   4.30 5.72 7.2 4.43 1.59 14       0
2   Crutches 3.7 4.50   6.10 7.15 8.5 5.92 1.48 14       0
3    Hearing 1.4 3.02   4.05 5.30 6.5 4.05 1.53 14       0
4       None 1.9 3.73   5.00 6.05 7.8 4.90 1.79 14       0
5 Wheelchair 1.7 4.73   5.70 6.35 7.6 5.34 1.75 14       0
```

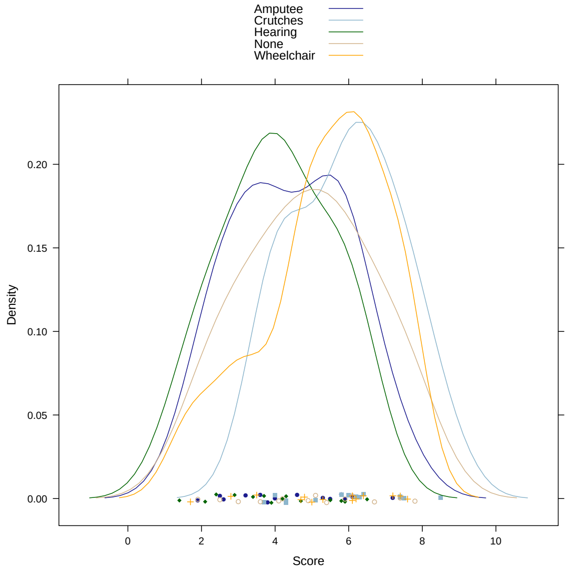
A total of 70 undergraduate students from a U.S. university were randomly assigned to view the tapes, 14 to each tape. The five kinds of tapes are: *None*, *Amputee*, *Crutches*, *Hearing* and *Wheelchair*. After reviewing the tape, each subject rated the qualifications of the application on 0-10 scale. Among the five handicap conditions, the *Crutches* group gave the highest mean score, while the *Hearing* group gave the lowest mean score. This is summarized on page 150 and in Display 6.1 of the *Sleuth*.

```
> with(subset(case0601, Handicap=="None"), stem(Score, scale=2))


  The decimal point is at the |

  1 | 9
  2 | 5
  3 | 06
  4 | 129
  5 | 149
  6 | 17
  7 | 48

> with(subset(case0601, Handicap=="Amputee"), stem(Score, scale=2))


  The decimal point is at the |

  1 | 9
  2 | 56
  3 | 268
  4 | 06
  5 | 3589
  6 | 1
  7 | 2

> with(subset(case0601, Handicap=="Crutches"), stem(Score, scale=1))


  The decimal point is at the |

  3 | 7
  4 | 033
  5 | 18
  6 | 0234
  7 | 445
  8 | 5

> with(subset(case0601, Handicap=="Hearing"), stem(Score, scale=2))


  The decimal point is at the |

  1 | 4
  2 | 149
  3 | 479
  4 | 237
  5 | 589
  6 | 5

> with(subset(case0601, Handicap=="Wheelchair"), stem(Score, scale=2))


  The decimal point is at the |

  1 | 7
  2 | 8
  3 | 5
  4 | 78
  5 | 03
  6 | 1124
  7 | 246
```

```
> bwplot(Handicap ~ Score, data=case0601)
```



```
> densityplot(~ Score, groups=Handicap, auto.key=TRUE, data=case0601)
```



The stem plots show the applicant qualification scores given by objectives. The boxplots and the density plots show that all the distributions are approximately normally distributed.

2.2 One-way ANOVA

First we fit the one way analysis of variance (ANOVA) model, using all of the groups. This corresponds to the interpretations on page 151.

```
> anova(lm(Score ~ Handicap, data=case0601))
```

```
Analysis of Variance Table

Response: Score
          Df Sum Sq Mean Sq F value Pr(>F)
Handicap   4   30.5    7.63    2.86   0.03
Residuals 65  173.3    2.67
```

The p-value provides some evidence that subjects rate qualifications differently according to handicap status.

By default, the use of the linear model (regression) function displays the pairwise differences between the first group and each of the other groups. Note that the overall test of the model is the same.

```
> summary(lm(Score ~ Handicap, data=case0601))


Call:
lm(formula = Score ~ Handicap, data = case0601)

Residuals:
   Min     1Q Median     3Q    Max
-3.643 -1.209  0.114  1.329  2.900

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)           4.429      0.436   10.15    5e-15
HandicapCrutches      1.493      0.617    2.42    0.018
HandicapHearing      -0.379      0.617   -0.61    0.542
HandicapNone          0.471      0.617    0.76    0.448
HandicapWheelchair    0.914      0.617    1.48    0.143

Residual standard error: 1.63 on 65 degrees of freedom
Multiple R-squared:  0.15,Adjusted R-squared:  0.0974
F-statistic: 2.86 on 4 and 65 DF,  p-value: 0.0301
```

The reference group here is *Amputee*, followed by *Crutches*, *Hearing*, *None* and *Wheelchair*.

Another way of viewing these results is through a model table, which displays the differences between the grand mean and the group means.

```
> model.tables(aov(Score ~ Handicap, data=case0601))

Tables of effects

 Handicap
Handicap
   Amputee   Crutches    Hearing       None Wheelchair
   -0.5000     0.9929    -0.8786    -0.0286     0.4143
```

Or by:

```
> mean(Score ~ Handicap, data=case0601)-mean(~ Score, data=case0601)

   Amputee   Crutches    Hearing       None Wheelchair
   -0.5000     0.9929    -0.8786    -0.0286     0.4143
```

2.3 Contrasts and linear combination

The Tukey-Kramer test is a reasonable method for these data. We can use this to verify the calculation on page 151.

```
> TukeyHSD(aov(lm(Score ~ Handicap, data=case0601)), "Handicap", ordered=TRUE, c(0,1,-1,0,0), 

  Tukey multiple comparisons of means
    95% family-wise confidence level
    factor levels have been ordered

Fit: aov(formula = lm(Score ~ Handicap, data = case0601))

$Handicap
                     diff    lwr  upr p adj
Amputee-Hearing     0.379 -1.353 2.11 0.972
None-Hearing        0.850 -0.882 2.58 0.644
Wheelchair-Hearing  1.293 -0.439 3.02 0.235
Crutches-Hearing    1.871  0.140 3.60 0.028
None-Amputee        0.471 -1.260 2.20 0.940
Wheelchair-Amputee  0.914 -0.817 2.65 0.578
Crutches-Amputee    1.493 -0.239 3.22 0.123
Wheelchair-None     0.443 -1.289 2.17 0.952
Crutches-None       1.021 -0.710 2.75 0.469
Crutches-Wheelchair 0.579 -1.153 2.31 0.881
```

Based on the Tukey-Kramer procedure, the difference is estimated to be higher for the *Crutches* tapes.

Next, we calculate the comparison of *Amputee/Hearing* to *Crutches/Wheelchair*.

```
> require(gmodels)
> fit.contrast(lm(Score ~ Handicap, data=case0601), "Handicap", c(-1, 1, -1, 0, 1), conf.int=0

                           Estimate Std. Error t value Pr(>|t|) lower CI
Handicap c=( -1 1 -1 0 1 )     2.79      0.873    3.19  0.00218     1.04
                           upper CI
Handicap c=( -1 1 -1 0 1 )     4.53
```

The results indicate a statistically significant difference between the average scores given to the *Wheelchair* and *Crutches* handicaps and the average scores given to the *Amputee* and *Hearing* handicaps.

To verify the calculations on page 155 we used the following contrast:

```
> fit.contrast(lm(Score ~ Handicap, data=case0601), "Handicap", c(-0.5, 0.5, -0.5, 0, 0.5), co

                                   Estimate Std. Error t value Pr(>|t|)
Handicap c=( -0.5 0.5 -0.5 0 0.5 )     1.39      0.436    3.19  0.00218
                                   lower CI upper CI
Handicap c=( -0.5 0.5 -0.5 0 0.5 )    0.521     2.26
```

Other multiple comparison procedures could also be implemented. The following shows the calculation on page 164.

```
> require(agricolae)

Loading required package:  agricolae

> LSD.test(aov(lm(Score ~ Handicap, data=case0601)), "Handicap")   # LSD
> HSD.test(aov(lm(Score ~ Handicap, data=case0601)), "Handicap")   # Tukey-Kramer
> LSD.test(aov(lm(Score ~ Handicap, data=case0601)), "Handicap", p.adj=c("bonferroni"))  # Bon
> scheffe.test(aov(lm(Score ~ Handicap, data=case0601)), "Handicap")    # Scheffe
```

The "Significant Difference" in each test result is the "95% interval half-width" described in the book.

3 Pre-existing Preference of Fish

Was Charles Darwin right that sexual selection is driven by females? This is the question addressed in case study 6.2 in the *Sleuth*.

3.1 Summary statistics and graphical display

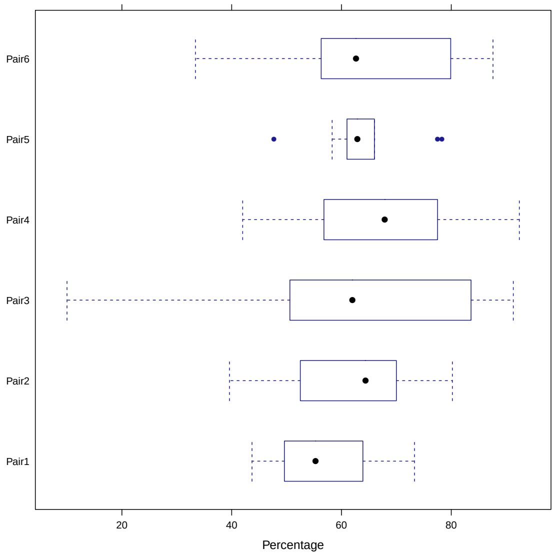
We begin by reading the data and summarizing the variables.

```
> summary(case0602)

   Percentage       Pair         Length
 Min.   :10.0   Pair1:16   Min.   :28.0
 1st Qu.:53.1   Pair2:14   1st Qu.:31.0
 Median :61.5   Pair3:17   Median :34.0
 Mean   :62.1   Pair4:14   Mean   :32.8
 3rd Qu.:71.8   Pair5: 9   3rd Qu.:34.0
 Max.   :92.4   Pair6:14   Max.   :35.0

> favstats(Percentage ~ Pair, data=case0602)
```

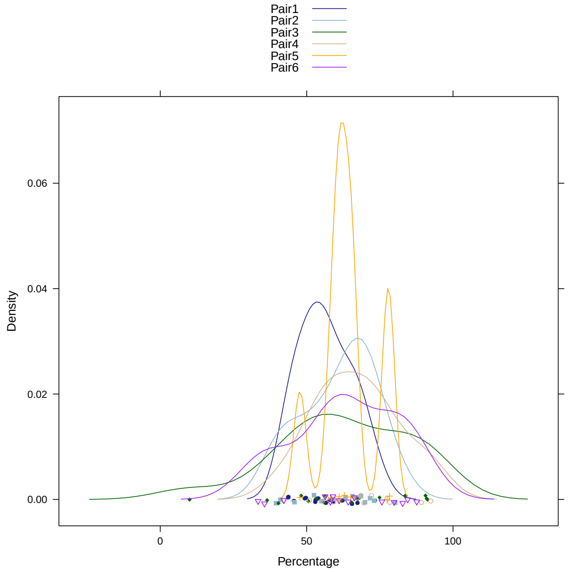
```
   Pair  min   Q1 median   Q3  max mean    sd  n missing
1 Pair1 43.7 49.7   55.3 63.1 73.3 56.4  9.02 16       0
2 Pair2 39.6 53.1   64.4 69.6 80.2 60.9 12.48 14       0
3 Pair3 10.0 50.6   62.0 83.6 91.3 62.4 22.29 17       0
4 Pair4 42.0 57.2   67.9 76.2 92.4 67.0 14.33 14       0
5 Pair5 47.7 61.0   62.9 66.0 78.3 64.2  9.41  9       0
6 Pair6 33.4 56.7   62.7 78.9 87.6 63.3 17.68 14       0
```

A total of 84 female fish were involved in this experiment, which is shown on page 153.

```
> bwplot(Pair ~ Percentage, data=case0602)
```



```
> densityplot(~ Percentage, groups=Pair, auto.key=TRUE, data=case0602)
```



Besides the distribution of pair 5, all distributions of other pairs are approximately normally distributed.

3.2 One-way ANOVA

First we fit the one way analysis of variance (ANOVA) model, using all of the groups:

```
> anova(lm(Percentage ~ Pair, data=case0602))

Analysis of Variance Table

Response: Percentage
          Df Sum Sq Mean Sq F value Pr(>F)
Pair       5    939     188    0.79   0.56
Residuals 78  18637     239
```

The p-value is not small, and does not provide much evidence that the mean percentage of time with the yellow-sword male significantly differed from one male pair to another back in the population.

By default, the use of the linear model (regression) function displays the pairwise differences between the first group and each of the other groups. Note that the overall test of the model is the same.

```
> summary(lm(Percentage ~ Pair, data=case0602))


Call:
lm(formula = Percentage ~ Pair, data = case0602)

Residuals:
   Min      1Q Median      3Q     Max
-52.43  -8.41   0.25  10.86  28.87

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    56.41       3.86   14.60   <2e-16
PairPair2       4.48       5.66    0.79    0.431
PairPair3       6.02       5.38    1.12    0.267
PairPair4      10.59       5.66    1.87    0.065
PairPair5       7.80       6.44    1.21    0.229
PairPair6       6.93       5.66    1.22    0.224

Residual standard error: 15.5 on 78 degrees of freedom
Multiple R-squared:  0.048,Adjusted R-squared:  -0.0131
F-statistic: 0.786 on 5 and 78 DF,  p-value: 0.563
```

The reference group here is pair 1, followed by pairs 2-6. Another way of viewing these results is through a model table, which displays the differences between the grand mean and the group means.

```
> model.tables(aov(Percentage ~ Pair, data=case0602))

Tables of effects

 Pair
     Pair1  Pair2   Pair3  Pair4 Pair5  Pair6
    -5.722 -1.243  0.3008  4.871 2.083  1.207
rep 16.000 14.000 17.0000 14.000 9.000 14.000
```

Or by:

```
> mean(Percentage ~ Pair, data=case0602)-mean(~ Percentage, data=case0602)

 Pair1  Pair2  Pair3  Pair4  Pair5  Pair6
-5.722 -1.243  0.301  4.871  2.083  1.207
```

3.3 Contrasts and linear combination

We can calculate the values on page 152 and Display 6.5 on page 158 using contrasts.

```
> require(gmodels)
> lc = fit.contrast(lm(Percentage ~ Pair, data=case0602), "Pair", c(5, -3, 1, 3, -9, 3), conf.

                         Estimate Std. Error t value Pr(>|t|) lower CI
Pair c=( 5 -3 1 3 -9 3 )    -25.1       54.8  -0.458    0.648     -134
```

```
                         upper CI
Pair c=( 5 -3 1 3 -9 3 )     83.9

> t=round(lc[, "t value"], 2); t

[1] -0.46

> pt(t, 78, lower.tail=TRUE)

[1] 0.323
```

The *t*-value is -0.46 and the one-sided *p*-value is 0.32.

```
> mean(mean(Percentage ~ Pair, data=case0602))

[1] 62.4

> t.test(mean(Percentage ~ Pair, data=case0602))


One Sample t-test

data:  mean(Percentage ~ Pair, data = case0602)
t = 40, df = 5, p-value = 1e-07
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 58.6 66.1
sample estimates:
mean of x
     62.4
```

The estimated mean percentage of time spent with the yellow-sword male is 62.378%. The one-sided *p*-value$<$ 0.0001, and the 95% confidence interval is (58.637%, 66.119%).

The Statistical Sleuth in R:
Chapter 7

Linda Loi Ruobing Zhang Kate Aloisio Nicholas J. Horton*

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> install.packages('mosaic')                    # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the Sleuth3 package.

```
> install.packages('Sleuth3')                    # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=4)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 7: Simple Linear Regression: A Model for the Mean using R.

2 The Big Bang

Is there relation between distance and radial velocity among extra-galactic nebulae? This is the question addressed in case study 7.1 in the *Sleuth*.

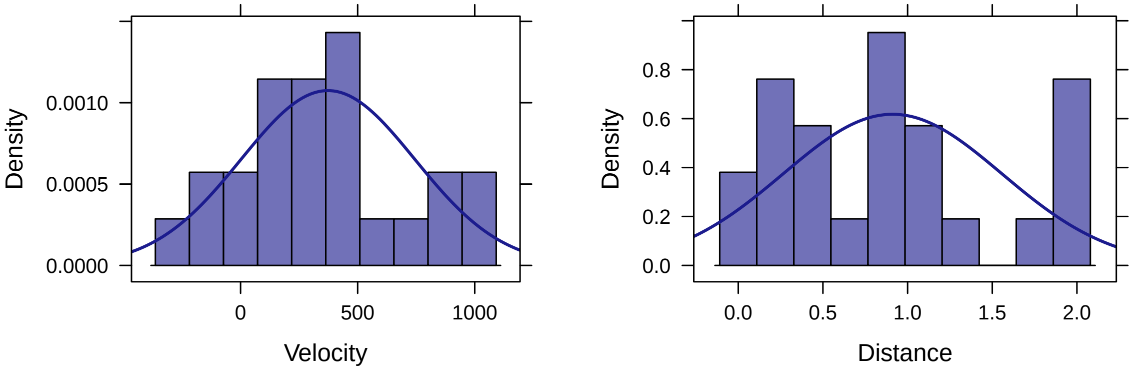
2.1 Summary statistics and graphical display

We begin by reading the data and summarizing the variables.

```
> summary(case0701)

    Velocity        Distance
 Min.   :-220   Min.   :0.030
 1st Qu.: 165   1st Qu.:0.407
 Median : 295   Median :0.900
 Mean   : 373   Mean   :0.911
 3rd Qu.: 538   3rd Qu.:1.175
 Max.   :1090   Max.   :2.000
```

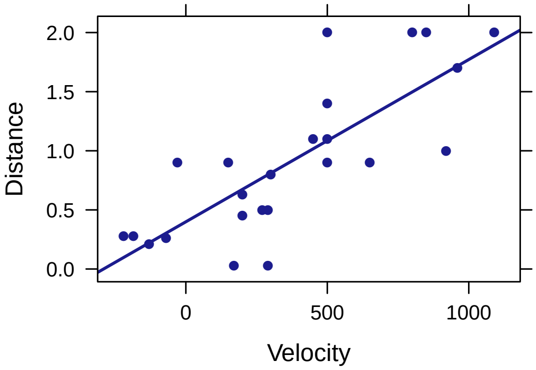
A total of 24 nebulae are included in this data.

```
> histogram(~ Velocity, type='density', density=TRUE, nint=10, data=case0701)
> histogram(~ Distance, type='density', density=TRUE, nint=10, data=case0701)
```



The density plots show that the distributions for the two variables are fairly symmetric, but more uniform than normally distributed.

```
> xyplot(Distance ~ Velocity, type=c("p", "r"), data=case0701)
```



The scatterplot is displayed on page 177 of the *Sleuth*. It indicates that there is a linear statistical relationship between distance and velocity.

2.2 The simple linear regression model

The following code presents the results interpreted on page 186 of the *Sleuth*.

```
> lm1 = lm(Distance ~ Velocity, data=case0701)
> summary(lm1)


Call:
lm(formula = Distance ~ Velocity, data = case0701)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-0.7672 -0.2352 -0.0108  0.2108  0.9146

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.399170   0.118666    3.36   0.0028
Velocity    0.001372   0.000228    6.02  4.6e-06

Residual standard error: 0.406 on 22 degrees of freedom
Multiple R-squared:  0.623,Adjusted R-squared:  0.605
F-statistic: 36.3 on 1 and 22 DF,  p-value: 4.61e-06
```

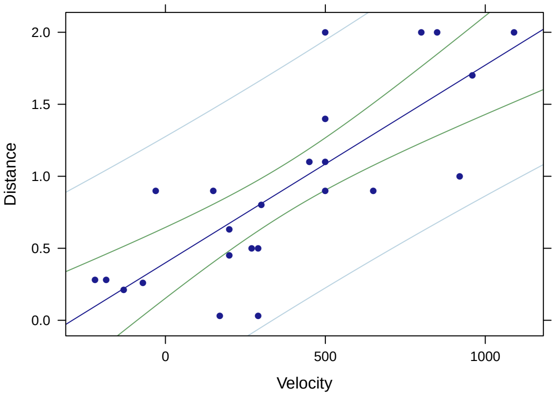
The estimated parameter for the intercept is 0.3992 megaparsecs and the estimated parameter for velocity is 0.0014 megaparsecs/(km/sec). The estimated mean function is $\hat{\mu}$ (distance|velocity) = 0.3992 + 0.0014 * velocity. The estimate of residual standard error is 0.4056 megaparsecs with 22 degrees of freedom. These results are also presented by Display 7.9 (page 187).

```
> fitted(lm1)

      1       2       3       4       5       6       7       8       9
0.63248 0.79717 0.22076 0.30310 0.14528 0.09724 0.67365 0.79717 0.76972
     10      11      12      13      14      15      16      17      18
0.67365 0.81089 0.35800 1.29124 0.60503 1.08537 1.66179 1.01675 1.08537
     19      20      21      22      23      24
1.08537 1.71668 1.08537 1.56572 1.49710 1.89509

> resid(lm1)^2

        1         2         3         4         5         6         7
0.3629818 0.5885477 0.0001157 0.0018578 0.0181508 0.0334009 0.0500202
        8         9        10        11        12        13        14
0.0883092 0.0727491 0.0019055 0.0001187 0.2937659 0.1530651 0.0870064
       15        16        17        18        19        20        21
0.0343636 0.4379599 0.0069299 0.0002139 0.0989894 0.0002783 0.8365403
       22        23        24
0.1886019 0.2529120 0.0110051

> sum(resid(lm1)^2)

[1] 3.62

> sum(resid(lm1)^2)/sum((fitted(lm1)-mean(~Distance, data=case0701))^2)

[1] 0.6062
```

Display 7.8 (page 186) shows the list of fitted values and residuals for this model. The sum of all the squared residuals is 3.62 and R-squared is 0.6062.

We can also display 95% confidence bands for the model line and the predicted values, the following graph is akin to Display 7.11 (page 191).

```
> xyplot(Distance ~ Velocity, panel=panel.lmbands, data=case0701)
```



2.3 Inferential Tools

First, we test β_0 (the intercept). From the previous summary, we know that the two-sided p-value for the intercept is 0.0028. This p-value is small enough for us to reject the null hypothesis that the estimated parameter for the intercept equals 0 (page 188).

Next we want to examine β_1. The current β_1 for $\hat{\mu}\,(\mathrm{Y}|\mathrm{X}) = \beta_0 + \beta_1 * \mathrm{X}$ is 0.0014, and we want to get the β_1 for $\hat{\mu}\,(\mathrm{Y}|\mathrm{X}) = \beta_1 * \mathrm{X}$, a model with no intercept (page 188).

```
> # linear regression with no intercept
> lm2 = lm(Distance ~ Velocity-1, data=case0701)
> summary(lm2)


Call:
lm(formula = Distance ~ Velocity - 1, data = case0701)

Residuals:
    Min      1Q  Median      3Q     Max
-0.7677 -0.0691  0.2295  0.4606  1.0393

Coefficients:
         Estimate Std. Error t value Pr(>|t|)
Velocity 0.001921   0.000191      10    7e-10

Residual standard error: 0.488 on 23 degrees of freedom
Multiple R-squared:  0.814,Adjusted R-squared:  0.806
```

```
F-statistic:  101 on 1 and 23 DF,  p-value: 7.05e-10

> confint(lm2)

            2.5 %   97.5 %
Velocity 0.001526 0.002317
```

Without the intercept, the new estimate for β_1 is 0.0019 megaparsec-second/km. The standard error is 1.91×10^{-4} megaparsecs with 23 degrees of freedom. The 95% confidence interval is (0.0015, 0.0023). Because 1 megaparsec-second/km = 979.8 billion years, the confidence interval could be written as 1.49 to 2.27 billion years, and the best estimate is 1.88 billion years (page 188).

3 Meat Processing and pH

Is there a relationship between postmortem muscle pH and time after slaughter? This is the question addressed in case study 7.2 in the *Sleuth*.

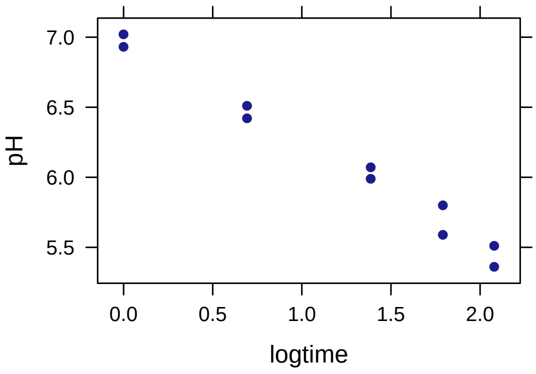
3.1 Summary statistics and graphical display

We begin by reading the data and summarizing the variables.

```
> summary(case0702)

      Time           pH
 Min.   :1.0   Min.   :5.36
 1st Qu.:2.0   1st Qu.:5.64
 Median :4.0   Median :6.03
 Mean   :4.2   Mean   :6.12
 3rd Qu.:6.0   3rd Qu.:6.49
 Max.   :8.0   Max.   :7.02
```

A total of 10 steer carcasses are included in this data as shown in Display 7.3, page 179.

```
> logtime = log(case0702$Time)
> xyplot(pH ~ logtime, data=case0702)
```



The above scatterplot indicates a negative linear relationship between pH and log(Time).

3.2 The simple linear regression model

We fit a simple linear regression model of pH on log(time) after slaughter. The estimated mean function will be $\hat{\mu}\,(\text{pH}|\text{logtime}) = \beta_0 + \beta_1 * \log(\text{Time})$.

```
> lm3 = lm(pH ~ logtime, data=case0702)
> summary(lm3)


Call:
lm(formula = pH ~ logtime, data = case0702)

Residuals:
    Min      1Q  Median      3Q     Max
-0.1147 -0.0589  0.0209  0.0361  0.1166

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.9836     0.0485   143.9  6.1e-15
logtime      -0.7257     0.0344   -21.1  2.7e-08

Residual standard error: 0.0823 on 8 degrees of freedom
Multiple R-squared:  0.982,Adjusted R-squared:  0.98
F-statistic:  444 on 1 and 8 DF,  p-value: 2.7e-08

> beta0 = coef(lm3)["(Intercept)"]; beta0

(Intercept)
      6.984

> beta1 = coef(lm3)["logtime"]; beta1
```

```
logtime
-0.7257

> sigma = summary(lm3)$sigma; sigma

[1] 0.08226
```

The $\hat{\beta}_0$ is 6.9836 and the $\hat{\beta}_1$ is -0.7257. The $\hat{\sigma}$ is 0.0823 (page 189).

3.3 Inferential Tools

With the previous information, we can calculate the 95% confidence interval for the estimated mean pH of steers 4 hours after slaughter (Display 7.10, page 189):

```
> mu = beta0+beta1*log(4); mu

(Intercept)
      5.978

> n = nrow(case0702)
> mean = mean(~logtime, data=case0702)
> sd = sd(~logtime, data=case0702)
> se = sigma*sqrt(1/n+(log(4)-mean)^2/((n-1)*sd)); se

[1] 0.0267

> upper = mu+qt(0.975, df=8)*se; upper

(Intercept)
      6.039

> lower = mu-qt(0.975, df=8)*se; lower

(Intercept)
      5.916
```

Or we can use the following code to get the same result:

```
> predict(lm3, interval="confidence")[5,]

  fit   lwr   upr
5.978 5.916 6.040
```

So the 95% confidence interval for estimated mean is (5.92, 6.04).

Next, we can calculate the 95% prediction interval for a steer carcass 4 hours after slaughter (Display 7.12, page 193):

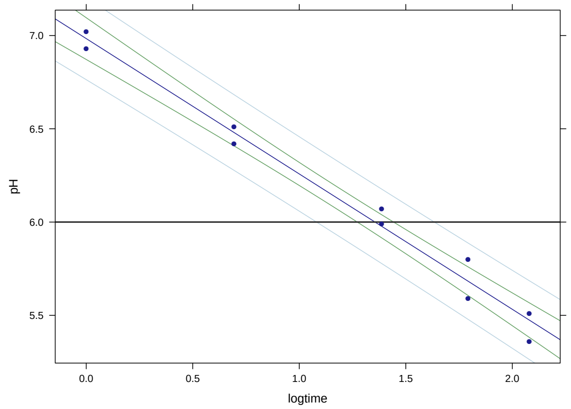
```
> pred = beta0+beta1*log(4); pred

(Intercept)
      5.978

> predse = sigma*sqrt(1+1/n+(log(4)-mean)^2/((n-1)*sd)); predse

[1] 0.08648

> predupper = pred+qt(0.975, df=8)*predse; predupper

(Intercept)
      6.177

> predlower = pred-qt(0.975, df=8)*predse; predlower

(Intercept)
      5.778
```

Or we can use the following code to get the 95% prediction interval for a steer carcass 4 hours after slaughter:

```
> predict(lm3, interval="prediction")[5,]

Warning in predict.lm(lm3, interval = "prediction"):  predictions on current data refer
to _future_ responses

  fit   lwr   upr
5.978 5.778 6.177
```

So the 95% prediction interval is (5.78, 6.18).

```
> xyplot(pH ~ logtime, abline=(h=6), data=case0702, panel=panel.lmbands)
```



The 95% prediction band is presented as Display 7.4 (page 180).

The Statistical Sleuth in R: Chapter 8

Kate Aloisio   Ruobing Zhang   Nicholas J. Horton*

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> install.packages('mosaic')                # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')               # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=4)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 8: A Closer Look at Assumptions for Simple Linear Regression using R.

2 Island Area and Number of Species

What is the relationship between the area of islands and the number of animal and plant species living on them? This is the question addressed in case study 8.1 in the *Sleuth*.

2.1 Summary statistics and graphical display

We begin by reading the data and summarizing the variables.

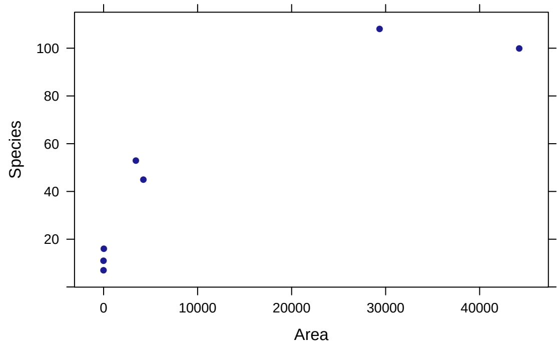
```
> case0801

   Area Species
1 44218     100
2 29371     108
3  4244      45
4  3435      53
5    32      16
6     5      11
7     1       7

> summary(case0801)
```

```
      Area          Species     
 Min.   :    1   Min.   :  7.0  
 1st Qu.:   18   1st Qu.: 13.5  
 Median : 3435   Median : 45.0  
 Mean   :11615   Mean   : 48.6  
 3rd Qu.:16808   3rd Qu.: 76.5  
 Max.   :44218   Max.   :108.0  
```

A total of 7 islands are included in this data as displayed in Display 8.1 (page 208).

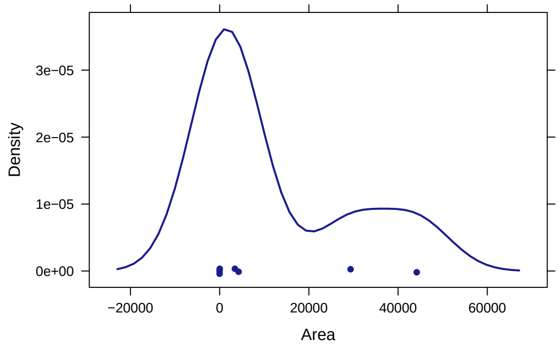
We can then observe the relationship between the area and the number of species for these islands with a scatterplot, akin to the top figure in Display 8.2 (page 209).

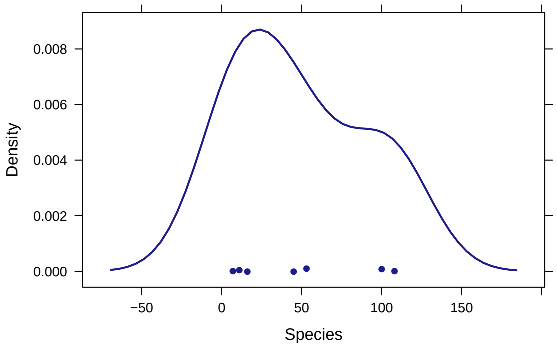
```
> xyplot(Species ~ Area, data=case0801)
```



It appears that the relationship with the observed values may not be linear, therefore we need to check the normality assumption to determine if transformations are nessessary.

```
> densityplot(~ Area, data=case0801)
> densityplot(~ Species, data=case0801)
```



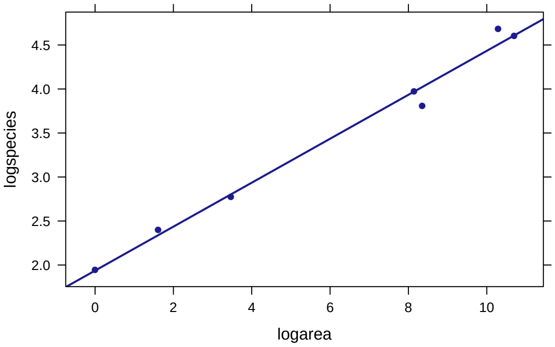


Since neither of these appear to be approximately normal, we log transformed both the the variables.

```
> case0801$logarea = with(case0801, log(Area))
> case0801$logspecies = with(case0801, log(Species))
```

Then we can create a log-log-scatterplot for these two variables, akin to the bottom figure in Display 8.2 (page 209).

```
> xyplot(logspecies ~ logarea, type = c("p", "r"), data=case0801)
```



2.2 Simple Linear Model

We first fit the model for $\mu\{\log(\text{Species})|\log(\text{Area})\} = \beta_0 + \beta_1 * \log(\text{Area})$.

```
> lm1 = lm(logspecies ~ logarea, data=case0801)
> summary(lm1)


Call:
lm(formula = logspecies ~ logarea, data = case0801)

Residuals:
        1         2         3         4         5         6         7
-0.002136  0.176975 -0.215487  0.000947 -0.029244  0.059543  0.009402

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.9365     0.0881    22.0  3.6e-06
logarea       0.2497     0.0121    20.6  5.0e-06

Residual standard error: 0.128 on 5 degrees of freedom
Multiple R-squared:  0.988,Adjusted R-squared:  0.986
F-statistic:  425 on 1 and 5 DF,  p-value: 4.96e-06
```

Thus our estimated equation becomes, $\hat{\mu}\{\log(\text{Species})|\log(\text{Area})\} = 1.94 + 0.25 * \log(\text{Area})$.

Next we calculate the 95% confidence interval for the estimates, note that the `logarea` 95% confidence interval is interpreted in the "Statistical Conclusion" on page 208:

```
> confint(lm1)

             2.5 % 97.5 %
(Intercept) 1.7100 2.1631
logarea     0.2186 0.2808
```

To interpret this log-log model the *Sleuth* notes that if $\hat{\mu}\{\log(\mathrm{Y})|\log(\mathrm{X})\} = \beta_0 + \beta_1 * \log(\mathrm{X})$ then $\mathrm{Median}\{\mathrm{Y}|\mathrm{X}\} = \exp(\beta_0)X^{\beta_1}$ (page 216). For this example the researchers are interested in a doubling effect (2^{β_1}). Therefore to obtain the 95% confidence interval for the multiplicative factor in the median we used the following code:

```
> 2^confint(lm1)

            2.5 % 97.5 %
(Intercept) 3.272  4.479
logarea     1.164  1.215
```

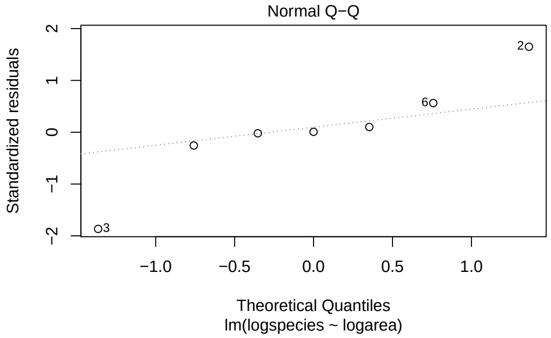
Thus for this model the estimated median number of species is 1.19 ($2^{0.25}$) with a 95% confidence interval between (1.16, 1.21). These match the numbers found on page 217.

2.3 Assessment of Assumptions

First we will have to assume independence from the information given. As seen in the above density plots, the observations for each variable were not normally distributed, once we preformed a log transformation the distribution of the values became more approximately normal.

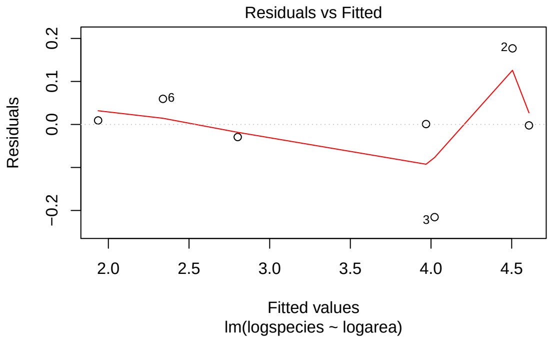
Next we can check for linearity.

```
> plot(lm1, which=2)
```



Lastly we can assess the assumption of equal variance.

```
> plot(lm1, which=1)
```



3 Breakdown Times for Insulating Fluid Under Different Voltages

How does the distribution of breakdown time depend on voltage? This is the question addressed in case study 8.2 in the *Sleuth*.

3.1 Summary statistics and graphical display

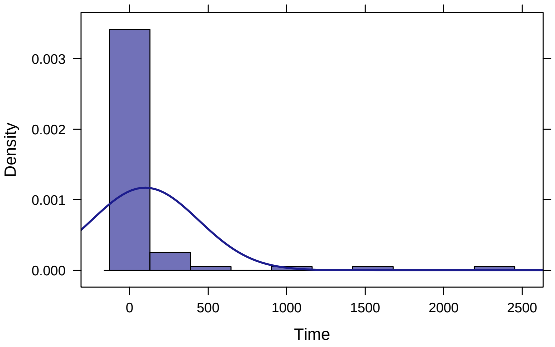
We begin by reading the data and summarizing the variables.

```
> summary(case0802)

      Time              Voltage          Group   
 Min.   :   0.1   Min.   :26.0   Group1: 3  
 1st Qu.:   1.6   1st Qu.:31.5   Group2: 5  
 Median :   6.9   Median :34.0   Group3:11  
 Mean   :  98.6   Mean   :33.1   Group4:15  
 3rd Qu.:  38.4   3rd Qu.:36.0   Group5:19  
 Max.   :2323.7   Max.   :38.0   Group6:15  
                                 Group7: 8  
```

A total of 76 samples of insulating fluids are included in this data. Each sample was placed in one of 7 groups representing different degrees of voltage. Each group varried in sample size as shown in Display 8.2 (page 209).

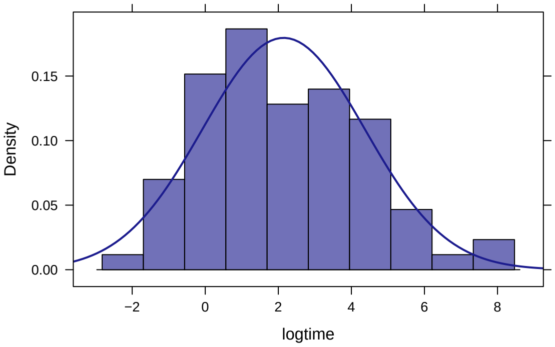
Before we can fit the simple linear regression model we need to assess the assumption of normality through density plots.

```
> histogram(~ Time, type='density', density=TRUE, nint=10, data=case0802)
```



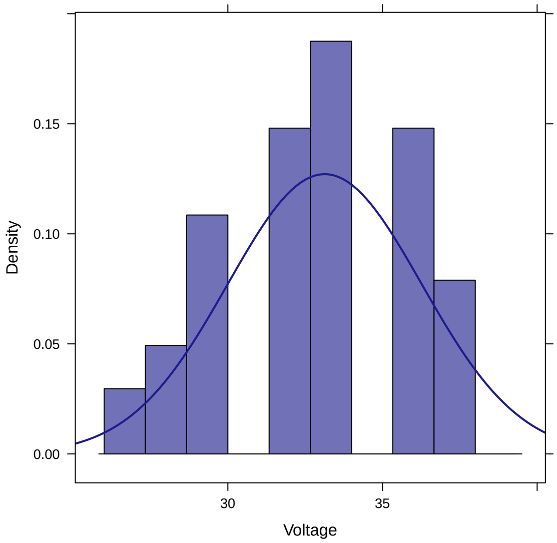
It appears that the distribution of `Time` is highly skewed with a long right tail. Therefore one possible transformation would be to take the log of the `Time` observations.

```
> case0802$logtime=with(case0802, log(Time))
> histogram(~ logtime, type='density', density=TRUE, nint=10, data=case0802)
```



Now the observations are approximately normally distributed.

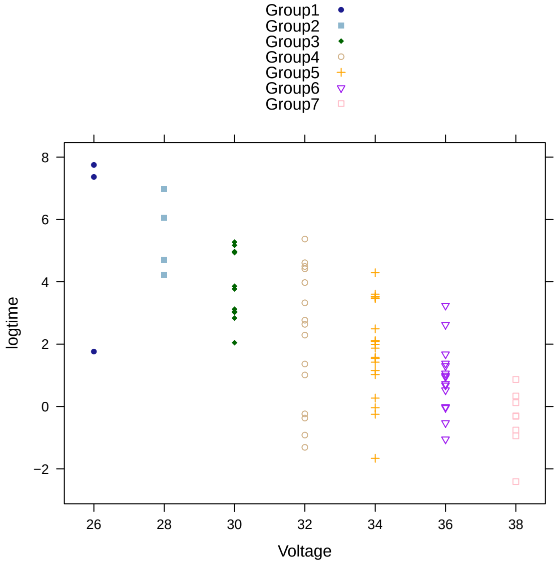
```
> histogram(~ Voltage, type='density', density=TRUE, nint=10, data=case0802)
```



The distribution of `Voltage` seems to be approximately normal.

Next we can observe the relationship between log(`Time`) and `Voltage`, the following figure is akin to Display 8.4 (page 211).

```
> xyplot(logtime ~ Voltage, groups=Group, auto.key=TRUE, data=case0802)
```



3.2 Simple linear regression models

The model that the researchers want to analyse is $\mu\{\log(\text{Time})|\text{Voltage}\} = \beta_0 + \beta_1 * \text{Voltage}$

```
> lm1 = lm(logtime ~ Voltage, data=case0802)
> summary(lm1)
```

```
Call:
lm(formula = logtime ˜ Voltage, data = case0802)

Residuals:
   Min     1Q Median     3Q    Max 
-4.029 -0.692  0.037  1.209  2.651 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  18.9555     1.9100    9.92  3.1e-15
Voltage      -0.5074     0.0574   -8.84  3.3e-13

Residual standard error: 1.56 on 74 degrees of freedom
Multiple R-squared:  0.514,Adjusted R-squared:  0.507 
F-statistic: 78.1 on 1 and 74 DF,  p-value: 3.34e-13
```

Therefore the estimated model is $\hat{\mu}\{\log(\text{Time})|\text{Voltage}\} = 18.96 + (\text{-}0.51)* \log(\text{Area})$. The R^2 for the model is 51.36%, as discussed on page 222.

For the interpretation of the model we first exponentiate the estimated coefficients since the response variable is logged as shown on page 216.

```
> exp(coef(lm1))

(Intercept)     Voltage 
  1.707e+08   6.021e-01 
```

Thus a 1 kV increase in volatge is associated with a multiplicative change in median breakdown time of 0.6.

Next we can calculate the 95% confidence interval for β_0 and β_1.

```
> confint(lm1)

              2.5 % 97.5 %
(Intercept) 15.1497 22.761
Voltage     -0.6217 -0.393
```

For the interpetation of the model we next need to exponentiat the above 95% confidence interval.

```
> exp(confint(lm1))

                2.5 %    97.5 %
(Intercept) 3.797e+06 7.675e+09
Voltage     5.370e-01 6.750e-01
```

Thus the 95% confidence interval for the multiplicative change in median breakdown time is (0.54, 0.68) as interpreted on page 216.

Next we can assess the fit using the Analysis of Variance (ANOVA). The ANOVA results below match those in the top half of Display 8.8 (page 219).

```
> anova(lm1)

Analysis of Variance Table

Response: logtime
          Df Sum Sq Mean Sq F value  Pr(>F)
Voltage    1    190   190.2    78.1 3.3e-13
Residuals 74    180     2.4
```

We can then compare this with a model with separate means for each group.

```
> lm2 = lm(logtime ~ as.factor(Voltage), data=case0802)
> summary(lm2)


Call:
lm(formula = logtime ~ as.factor(Voltage), data = case0802)

Residuals:
   Min     1Q Median     3Q    Max
-3.868 -0.819  0.074  1.122  3.143

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)             5.624      0.916    6.14  4.7e-08
as.factor(Voltage)28   -0.294      1.159   -0.25  0.80019
as.factor(Voltage)30   -1.802      1.034   -1.74  0.08571
as.factor(Voltage)32   -3.395      1.004   -3.38  0.00118
as.factor(Voltage)34   -3.838      0.986   -3.89  0.00023
as.factor(Voltage)36   -4.722      1.004   -4.70  1.3e-05
as.factor(Voltage)38   -6.048      1.074   -5.63  3.6e-07

Residual standard error: 1.59 on 69 degrees of freedom
Multiple R-squared:  0.531,Adjusted R-squared:  0.49
F-statistic:   13 on 6 and 69 DF,  p-value: 8.87e-10
```

This model has a F-statistic of 13 with a p-value < 0.0001, as shown in the bottom half of Display 8.8 (page 218).

Another way of viewing this model is with the ANOVA.

```
> anova(lm2)

Analysis of Variance Table

Response: logtime
                   Df Sum Sq Mean Sq F value  Pr(>F)
as.factor(Voltage)  6    196    32.7      13 8.9e-10
Residuals          69    174     2.5
```

Note that the values for the `Residuals` can also be found in the bottom half of Display 8.8 (page 219).

The F-statistic and its associated p-value for the lack-of-fit discussion on page 220 can be calculated by comparing the two models with an ANOVA.

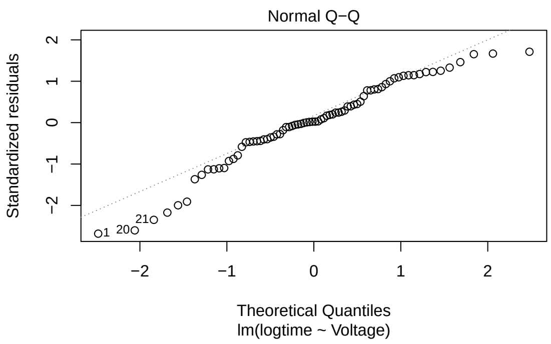
```
> anova(lm1, lm2)

Analysis of Variance Table

Model 1: logtime ~ Voltage
Model 2: logtime ~ as.factor(Voltage)
  Res.Df RSS Df Sum of Sq   F Pr(>F)
1     74 180
2     69 174  5      6.33 0.5   0.77
```

3.3 Assessment of Assumptions

First we will have to assume independence for the information given. As seen in the above density plot the observations for `Time` was not normally distributed, once we preformed a log transformation the distribution of the values became more approximately normal.

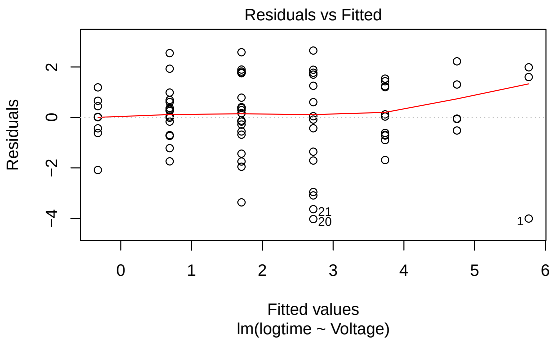
Next we can check for linearity, the following figure is akin to the right side graph in Display 8.14 (page 226).

```
> plot(lm1, which=2)
```



Lastly we can assess the assumption of equal variance.

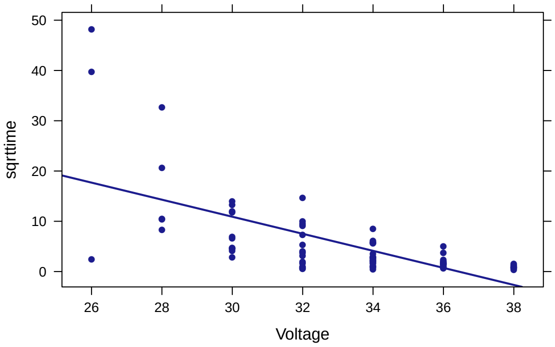
```
> plot(lm1, which=1)
```



3.4 Other transformations

The *Sleuth* also discusses the use of a square root transformation for the breakdown time. The following figure is a scatterplot of the square root of breakdown time versus voltage, akin to the left figure in Display 8.7 (page 215).

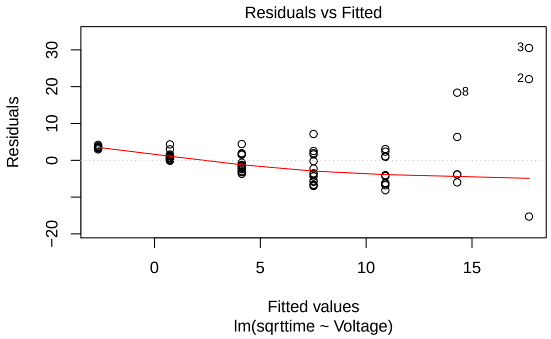
```
> case0802$sqrttime = with(case0802, sqrt(Time))
> xyplot(sqrttime ~ Voltage, type=c("p", "r"), data=case0802)
```



We can assess this transformation by observing the residual plot based on the simple linear regression fit, akin to the right figure in Display 8.7 (page 215).

```
> lm3 = lm(sqrttime ~ Voltage, data=case0802)
> summary(lm3)


Call:
lm(formula = sqrttime ~ Voltage, data = case0802)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-15.285  -3.711   0.142   2.040  30.514

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   61.784      7.777    7.94  1.6e-11
Voltage       -1.696      0.234   -7.26  3.3e-10

Residual standard error: 6.35 on 74 degrees of freedom
Multiple R-squared:  0.416,Adjusted R-squared:  0.408
F-statistic: 52.7 on 1 and 74 DF,  p-value: 3.25e-10

> plot(lm3, which = 1)
```



The Statistical Sleuth in R: Chapter 9

Linda Loi Kate Aloisio Ruobing Zhang Nicholas J. Horton*

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> install.packages('mosaic')                # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')               # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 9: Multiple Regression using R.

2 Effects of light on meadowfoam flowering

Do different amounts of light affect the growth of meadowfoam (a small plant used to create seed oil)? This is the question addressed in case study 9.1 in the *Sleuth*.

2.1 Data coding, summary statistics and graphical display

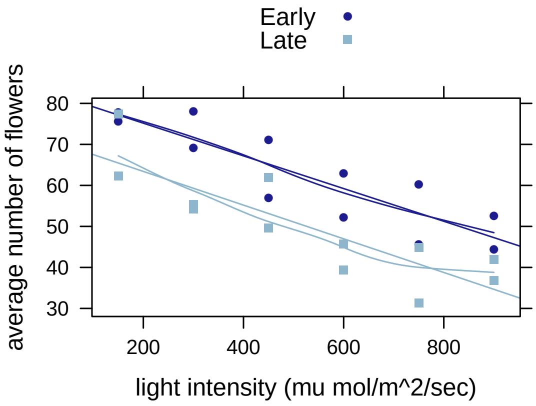
We begin by reading the data, clarifying the data, and summarizing the variables.

```
> head(case0901)

  Flowers Time Intensity
1    62.3    1       150
2    77.4    1       150
3    55.3    1       300
4    54.2    1       300
5    49.6    1       450
6    61.9    1       450

> case0901 = transform(case0901, Time = factor(ifelse(case0901$Time > 1, "Early", "Late")))
> summary(case0901)
```

```
    Flowers        Time      Intensity
 Min.   :31.3   Early:12   Min.   :150
 1st Qu.:45.4   Late :12   1st Qu.:300
 Median :54.8              Median :525
 Mean   :56.1              Mean   :525
 3rd Qu.:64.5              3rd Qu.:750
 Max.   :78.0              Max.   :900

> favstats(Flowers ~ Intensity | Time, data=case0901)

        Time min    Q1 median   Q3  max mean     sd  n missing
1  150.Early 75.6 76.1   76.7 77.2 77.8 76.7  1.556  2       0
2  300.Early 69.1 71.3   73.5 75.8 78.0 73.5  6.293  2       0
3  450.Early 57.0 60.5   64.0 67.6 71.1 64.0  9.970  2       0
4  600.Early 52.2 54.9   57.5 60.2 62.9 57.5  7.566  2       0
5  750.Early 45.6 49.3   53.0 56.6 60.3 53.0 10.394  2       0
6  900.Early 44.4 46.4   48.5 50.6 52.6 48.5  5.798  2       0
7   150.Late 62.3 66.1   69.8 73.6 77.4 69.8 10.677  2       0
8   300.Late 54.2 54.5   54.8 55.0 55.3 54.8  0.778  2       0
9   450.Late 49.6 52.7   55.8 58.8 61.9 55.8  8.697  2       0
10  600.Late 39.4 41.0   42.5 44.1 45.7 42.5  4.455  2       0
11  750.Late 31.3 34.7   38.1 41.5 44.9 38.1  9.617  2       0
12  900.Late 36.8 38.1   39.3 40.6 41.9 39.3  3.606  2       0
13     Early 44.4 52.5   61.6 72.2 78.0 62.2 12.117 12       0
14      Late 31.3 41.3   47.7 56.9 77.4 50.1 12.919 12       0
```

A total of 24 meadowfoam plants were included in this data. There were 12 treatment groups - 6 light intensities at each of the 2 timing levels (Display 9.2, page 239 of the *Sleuth*). The following code generates the scatterplot of the average number of flowers per plant versus the applied light intensity for each of the 12 experimental units akin to Display 9.3 on page 240.

```
> xyplot(Flowers ~ Intensity, groups=Time, type=c("p", "r", "smooth"),
+        data=case0901, auto.key=TRUE,
+        xlab="light intensity (mu mol/m^2/sec)", ylab="average number of flowers")
```



2.2 Multiple linear regression model

We next fit a multiple linear regression model that specifies parallel regression lines for the mean number of flowers as a function of light intensity as interpreted on page 239.

```
> lm1 = lm(Flowers ~ Intensity+Time, data=case0901)
> summary(lm1)


Call:
lm(formula = Flowers ~ Intensity + Time, data = case0901)

Residuals:
   Min     1Q Median     3Q    Max 
 -9.65  -4.14  -1.56   5.63  12.16 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  83.46417    3.27377   25.49  < 2e-16
Intensity    -0.04047    0.00513   -7.89    1e-07
TimeLate    -12.15833    2.62956   -4.62  0.00015

Residual standard error: 6.44 on 21 degrees of freedom
Multiple R-squared:  0.799,Adjusted R-squared:  0.78 
F-statistic: 41.8 on 2 and 21 DF,  p-value: 4.79e-08

> confint(lm1, level=.95) # 95% confidence intervals

              2.5 %  97.5 %
(Intercept)  76.6560 90.2723
Intensity    -0.0511 -0.0298
TimeLate    -17.6268 -6.6899
```

We can also fit a multiple linear regression with an interaction between light intensity and timing of its initiation as shown in Display 9.14 (page 260) and interpreted on page 239.

```
> lm2 = lm(Flowers ~ Intensity*Time, data=case0901)
> summary(lm2)


Call:
lm(formula = Flowers ~ Intensity * Time, data = case0901)

Residuals:
   Min     1Q Median     3Q    Max 
 -9.52  -4.28  -1.42   5.47  11.94 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         83.14667    4.34330   19.14  2.5e-14
Intensity           -0.03987    0.00744   -5.36  3.0e-05
TimeLate           -11.52333    6.14236   -1.88    0.075
Intensity:TimeLate  -0.00121    0.01051   -0.12    0.910

Residual standard error: 6.6 on 20 degrees of freedom
Multiple R-squared:  0.799,Adjusted R-squared:  0.769 
F-statistic: 26.5 on 3 and 20 DF,  p-value: 3.55e-07
```

3 Why do some mammals have large brains?

What characteristics predict large brains in mammals? This is the question addressed in case study 9.2 in the *Sleuth*.

3.1 Data coding and summary statistics

We begin by reading the data and summarizing the variables.

```
> case0902 = transform(case0902, logbrain = log(Brain))
> case0902 = transform(case0902, logbody = log(Body))
> case0902 = transform(case0902, loggest = log(Gestation))
> case0902 = transform(case0902, loglitter = log(Litter))
```

```
> summary(case0902)

             Species        Brain            Body          Gestation  
 Aardvark        : 1   Min.    :   0   Min.    :   0   Min.   : 16  
 Acouchis        : 1   1st Qu.:  13   1st Qu.:   2   1st Qu.: 63  
```

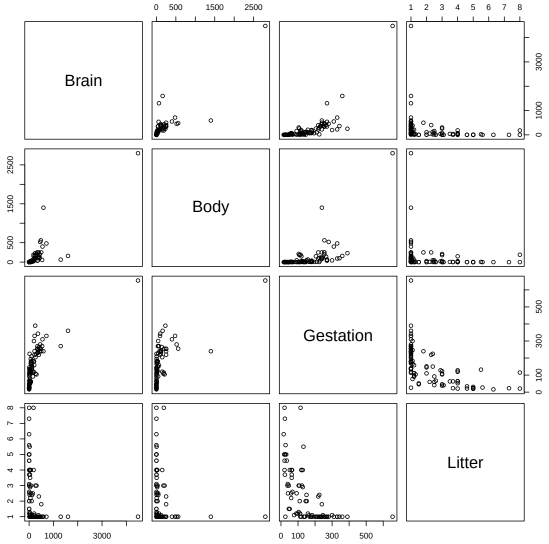
```
African elephant: 1   Median :  74   Median :   9   Median :134
Agoutis         : 1   Mean   : 219   Mean   : 108   Mean   :151
Axis deer       : 1   3rd Qu.: 260   3rd Qu.:  95   3rd Qu.:226
Badger          : 1   Max.   :4480   Max.   :2800   Max.   :655
(Other)         :90
    Litter        logbrain        logbody         loggest
Min.   :1.00   Min.   :-0.80   Min.   :-4.07   Min.   :2.77
1st Qu.:1.00   1st Qu.: 2.53   1st Qu.: 0.73   1st Qu.:4.14
Median :1.20   Median : 4.30   Median : 2.19   Median :4.89
Mean   :2.31   Mean   : 3.86   Mean   : 2.13   Mean   :4.71
3rd Qu.:3.20   3rd Qu.: 5.56   3rd Qu.: 4.55   3rd Qu.:5.42
Max.   :8.00   Max.   : 8.41   Max.   : 7.94   Max.   :6.48

  loglitter
Min.   :0.000
1st Qu.:0.000
Median :0.182
Mean   :0.598
3rd Qu.:1.162
Max.   :2.079
```

A total of 96 mammals were included in this data. The average values of brain weight, body weight, gestation length, and litter size for each of the species were calculated and presented in Display 9.4 (page 241 of the *Sleuth*).

3.2 Graphical presentation

The following displays a simple (unadorned) pairs plot, akin to Display 9.10 on page 255.

```
> smallds = subset(case0902, select=c("Brain", "Body", "Gestation", "Litter"))
> pairs(smallds)
```

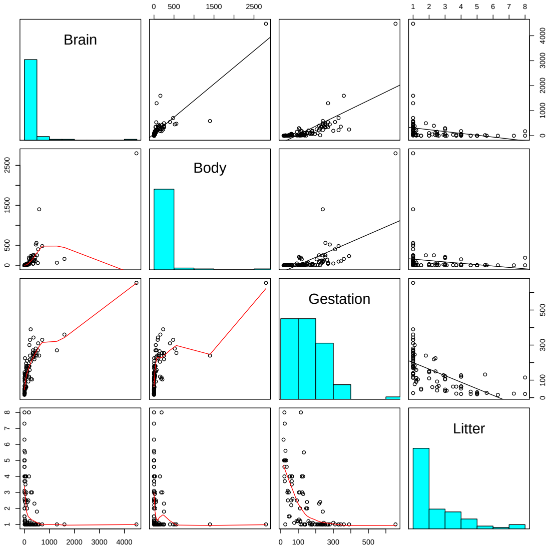


We can make it fancier if we like.

```
>   panel.hist = function(x, ...)
+   {
+     usr = par("usr"); on.exit(par(usr))
+     par(usr = c(usr[1:2], 0, 1.5) )
+     h = hist(x, plot=FALSE)
+     breaks = h$breaks; nB = length(breaks)
+     y = h$counts; y = y/max(y)
+     rect(breaks[-nB], 0, breaks[-1], y, col="cyan", ...)
+   }
>
> panel.lm = function(x, y, col=par("col"), bg=NA,
+                     pch=par("pch"), cex=1, col.lm="red", ...)
+ {
+   points(x, y, pch=pch, col=col, bg=bg, cex=cex)
+   ok = is.finite(x) & is.finite(y)
+   if (any(ok))
+     abline(lm(y[ok] ~ x[ok]))
+ }
```

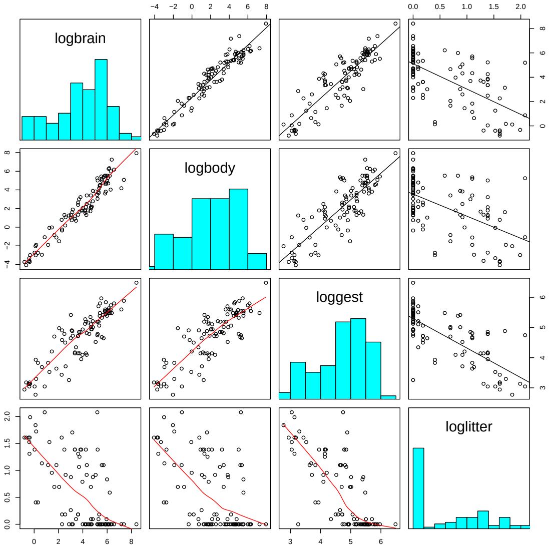
Below is a somewhat fancier pairs plot.

```
>   pairs(~ Brain+Body+Gestation+Litter,
+         lower.panel=panel.smooth, diag.panel=panel.hist,
+         upper.panel=panel.lm, data=case0902)
```



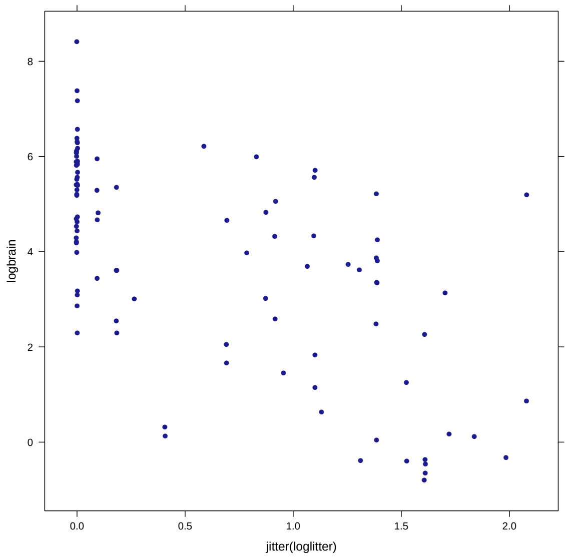
Here is an even fancier pairs plot using the log-transformed variables, akin to Display 9.11 on page 256.

```
>   pairs(~ logbrain+logbody+loggest+loglitter,
+                lower.panel=panel.smooth, diag.panel=panel.hist,
+                upper.panel=panel.lm, data=case0902)
```



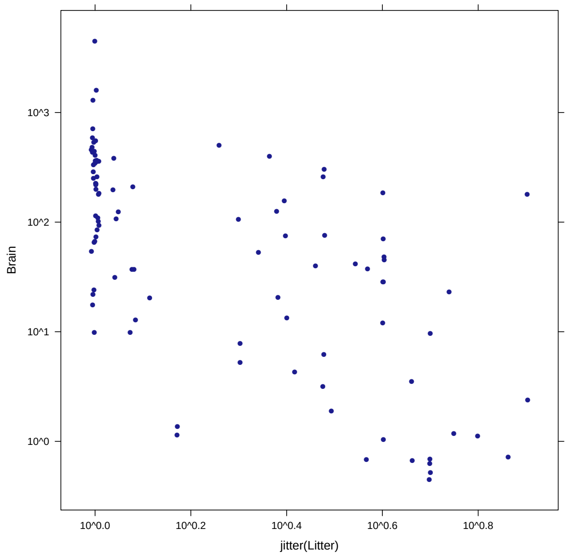
The following displays a jittered scatterplot of log brain weight as a function of log litter size, akin to Display 9.12 on page 258.

```
>   xyplot(logbrain ~ jitter(loglitter), data=case0902)
```



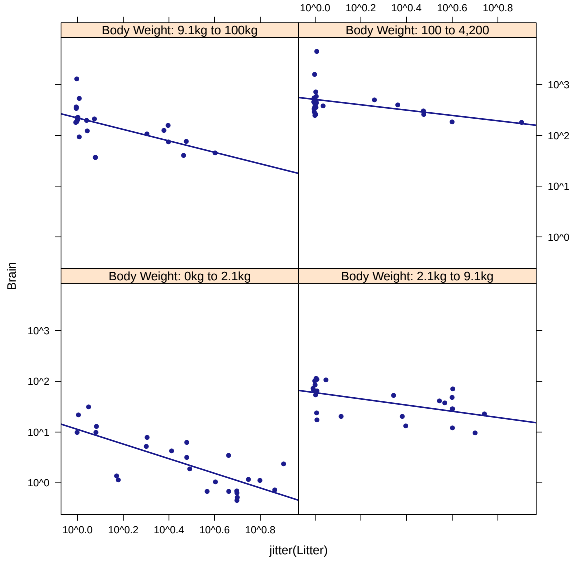
Below displays a jittered scatterplot using the original data on a log-transformed axis, akin to Display 9.12 on page 258.

```
>   xplot(Brain ~ jitter(Litter), scales=list(y=list(log=TRUE),
+                                            x=list(log=TRUE)), data=case0902)
```



The following displays a jittered scatterplot using the original data stratified by body weight on a log-transformed axis, akin to Display 9.13 on page 259.

```
> case0902$weightcut = cut(case0902$Body, breaks=c(0, 2.1, 9.1, 100, 4200), labels=c("Body Weig
> xyplot(Brain ~ jitter(Litter) | weightcut,
+        scales=list(y=list(log=TRUE), x=list(log=TRUE)), type=c("p", "r"), data=case0902)
```



3.3 Multiple linear regression model

The following model is interpreted on page 240 and shown in Display 9.15 (page 260).

```
> lm1 = lm(logbrain ~ logbody+loggest+loglitter, data=case0902)
> summary(lm1)


Call:
lm(formula = logbrain ~ logbody + loggest + loglitter, data = case0902)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9541 -0.2964 -0.0311  0.2811  1.5749

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.8548     0.6617    1.29   0.1996
logbody       0.5751     0.0326   17.65   <2e-16
loggest       0.4179     0.1408    2.97   0.0038
loglitter    -0.3101     0.1159   -2.67   0.0089

Residual standard error: 0.475 on 92 degrees of freedom
Multiple R-squared:  0.954,Adjusted R-squared:  0.952
F-statistic:  632 on 3 and 92 DF,  p-value: <2e-16
```

The Statistical Sleuth in R:
Chapter 10

Linda Loi Kate Aloisio Ruobing Zhang Nicholas J. Horton[*]

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

```
> install.packages('mosaic')               # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

[*] Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')                    # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 10: Inferential Tools for Multiple Regression using R.

2 Galileo's data on the motion of falling bodies

Galileo investigated the relationship between height and horizontal distance. This is the question addressed in case study 10.1 in the *Sleuth*.

2.1 Data coding, summary statistics and graphical display

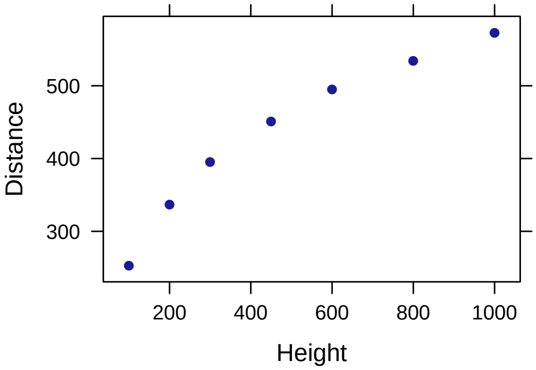
We begin by reading the data and summarizing the variables.

```
> summary(case1001)

    Distance        Height
 Min.   :253   Min.   : 100
 1st Qu.:366   1st Qu.: 250
 Median :451   Median : 450
 Mean   :434   Mean   : 493
 3rd Qu.:514   3rd Qu.: 700
 Max.   :573   Max.   :1000

> favstats(~ Distance, data=case1001)

 min  Q1 median  Q3 max mean  sd n missing
 253 366    451 514 573  434 113 7       0
```

There we a total of 7 trials of Galileo's experiment. For each trial, he recorded the initial height and then measured the horizontal distance as shown in Display 10.1 (page 272).

We can start to explore this relationship by creating a scatterplot of Galileo's horizontal distances versus initial heights. The following graph is akin to Display 10.2 (page 273).

```
> xyplot(Distance ~ Height, data=case1001)
```



2.2 Models

The first model that we created is a cubic model as interpreted on page 273 and summarized in Display 10.13 (page 291).

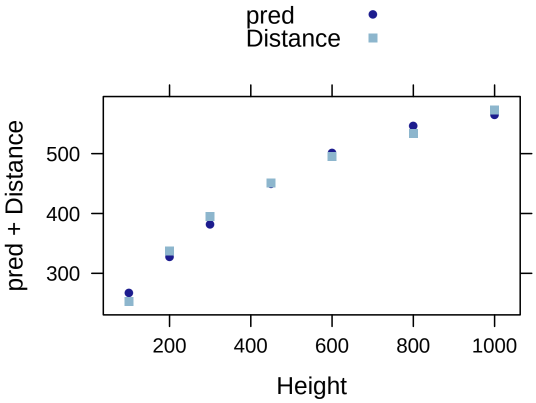
```
> lm1 = lm(Distance ~ Height+I(Height^2)+I(Height^3), data=case1001); summary(lm1)


Call:
lm(formula = Distance ~ Height + I(Height^2) + I(Height^3), data = case1001)

Residuals:
      1       2       3       4       5       6       7 
-2.4036  3.5809  1.8917 -4.4688 -0.0804  2.3216 -0.8414 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.56e+02   8.33e+00   18.71  0.00033
Height       1.12e+00   6.57e-02   16.98  0.00044
I(Height^2) -1.24e-03   1.38e-04   -8.99  0.00290
I(Height^3)  5.48e-07   8.33e-08    6.58  0.00715

Residual standard error: 4.01 on 3 degrees of freedom
Multiple R-squared:  0.999,Adjusted R-squared:  0.999 
F-statistic: 1.6e+03 on 3 and 3 DF,  p-value: 2.66e-05
```

We next decrease the polynomial for *Height* by one degree to obtain a quadratic model as interpreted on page 273 and summarized in Display 10.7 (page 281). This model is used for most of the following results.

```
> lm2 = lm(Distance ~ Height+I(Height^2), data=case1001); summary(lm2)


Call:
lm(formula = Distance ~ Height + I(Height^2), data = case1001)

Residuals:
     1      2      3      4      5      6      7
-14.31   9.17  13.52   1.94  -6.18 -12.61   8.46

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.00e+02   1.68e+01   11.93  0.00028
Height       7.08e-01   7.48e-02    9.47  0.00069
I(Height^2) -3.44e-04   6.68e-05   -5.15  0.00676

Residual standard error: 13.6 on 4 degrees of freedom
Multiple R-squared:  0.99,Adjusted R-squared:  0.986
F-statistic:  205 on 2 and 4 DF,  p-value: 9.33e-05
```

The following figure presents the predicted values from the quadratic model using the original data points akin to Display 10.2 (page 273).

```
> case1001$pred = predict(lm2)
> xyplot(pred+Distance ~ Height, auto.key=TRUE, data=case1001)
```



To obtain the expected values of $\hat{\mu}\,(\text{Distance}|\text{Height} = 0)$ and $\hat{\mu}\,(\text{Distance}|\text{Height} = 250)$, we used the `predict()` command with the quadratic model as shown in Display 10.7 (page 281).

```
> predict(lm2, interval="confidence", data.frame(Height=c(0, 250)))

  fit lwr upr
1 200 153 246
2 356 337 374
```

We can also verify the above confidence interval calculations with the following code:

```
> 355.1+c(-1, 1)*6.62*qt(.975, 4)

[1] 337 373
```

To verify numbers on page 284, an interval for the predicted values , we used the following code:

```
> predict(lm2, interval="predict", data.frame(Height=c(0, 250)))

  fit lwr upr
1 200 140 260
2 356 313 398
```

Lastly, we produced an ANOVA for the quadratic model interpreted on page 288 (Display 10.11).

```
> anova(lm2)

Analysis of Variance Table

Response: Distance
            Df Sum Sq Mean Sq F value Pr(>F)
Height       1  71351   71351   383.6  4e-05
I(Height^2)  1   4927    4927    26.5 0.0068
Residuals    4    744     186
```

3 Echolocation in bats

How do bats make their way about in the dark? Echolocation requires a lot of energy. Does it depend on mass and species? This is the question addressed in case study 10.2 in the *Sleuth*.

3.1 Data coding, summary statistics and graphical display

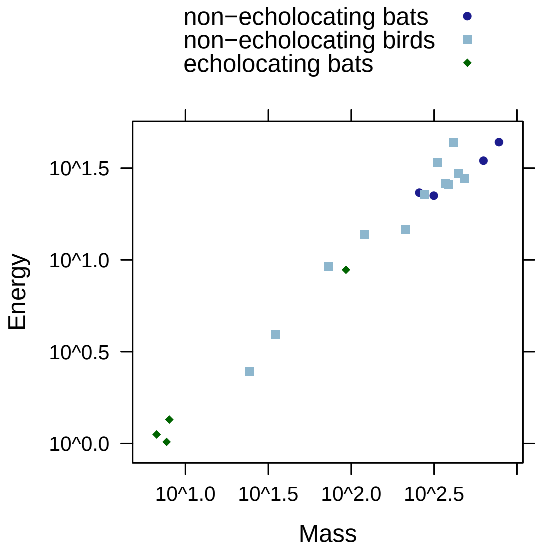
We begin by reading the data, performing transformations where necessary and summarizing the variables.

```
> case1002 = transform(case1002, Type = factor(Type, levels = c("non-echolocating bats","non-e
> case1002$logmass = log(case1002$Mass); case1002$logenergy = log(case1002$Energy)
> summary(case1002)

      Mass                            Type        Energy         logmass
 Min.   :  7   non-echolocating bats : 4   Min.   : 1.0   Min.   :1.90
 1st Qu.: 63   non-echolocating birds:12   1st Qu.: 7.6   1st Qu.:4.10
 Median :266   echolocating bats     : 4   Median :22.6   Median :5.58
 Mean   :263                               Mean   :19.5   Mean   :4.89
 3rd Qu.:391                               3rd Qu.:28.2   3rd Qu.:5.97
 Max.   :779                               Max.   :43.7   Max.   :6.66
   logenergy
 Min.   :0.02
 1st Qu.:1.98
 Median :3.12
 Mean   :2.48
 3rd Qu.:3.34
 Max.   :3.78

> favstats(Mass ~ Type, data=case1002)

                    Type   min     Q1 median    Q3 max  mean    sd  n
1  non-echolocating bats 258.0 300.75 471.50 665.8 779 495.0 249.6  4
2 non-echolocating birds  24.3 108.20 302.50 391.0 480 263.2 165.2 12
3      echolocating bats   6.7   7.45   7.85  29.2  93  28.9  42.8  4
  missing
1       0
2       0
3       0

> favstats(Energy ~ Type, data=case1002)

                    Type   min   Q1 median    Q3   max  mean    sd  n
1  non-echolocating bats 22.40 23.1  29.05 37.02 43.70 31.05 10.15  4
2 non-echolocating birds  2.46 12.6  24.35 28.23 43.70 21.15 12.52 12
3      echolocating bats  1.02  1.1   1.24  3.22  8.83  3.08  3.84  4
  missing
1       0
2       0
3       0
```

A total of 20 flying vertebrates were included in this study. There were 4 echolocating bats, 4 non-echolocating bats, and 12 non-echolocating birds. For each subject their *mass* and *flight energy expenditure* were recorded as shown in Display 10.3 (page 274).

We can next observe the pattern between log(energy expenditure) as a function of log(body mass) for each group with a scatterplot. The following figure is akin to Display 10.4 (page 275).

```
> xyplot(Energy ~ Mass, group=Type, scales=list(y=list(log=TRUE),
+     x=list(log=TRUE)), auto.key=TRUE, data=case1002)
```



3.2 Multiple regression

We first evaluate a multiple regression model for log(energy expenditure) given type of species and log(body mass) as defined on page 276 and shown in Display 10.6 (page 277).

```
> lm1 = lm(logenergy ~ logmass+Type, data=case1002); summary(lm1)


Call:
lm(formula = logenergy ~ logmass + Type, data = case1002)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.2322 -0.1220 -0.0364  0.1257  0.3446 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 -1.5764     0.2872   -5.49  5.0e-05
logmass                      0.8150     0.0445   18.30  3.8e-12
Typenon-echolocating birds   0.1023     0.1142    0.90     0.38
Typeecholocating bats        0.0787     0.2027    0.39     0.70

Residual standard error: 0.186 on 16 degrees of freedom
Multiple R-squared:  0.982,Adjusted R-squared:  0.978 
F-statistic:  284 on 3 and 16 DF,  p-value: 4.46e-14
```

Next, we calculate confidence intervals for the coefficients which are interpreted on page 278.

```
> confint(lm1)

                             2.5 % 97.5 %
(Intercept)                 -2.185 -0.967
logmass                      0.721  0.909
Typenon-echolocating birds -0.140  0.344
Typeecholocating bats      -0.351  0.508

> exp(confint(lm1))

                            2.5 % 97.5 %
(Intercept)                 0.112   0.38
logmass                     2.056   2.48
Typenon-echolocating birds 0.870   1.41
Typeecholocating bats      0.704   1.66
```

Since the significance of a model depends on which variables are included, the *Sleuth* proposes two other models, one only looking at the type of flying animal and the other allows the three groups to have different straight-line regressions with *mass*. These two models are displayed below and discussed on pages 278-279.

```
> summary(lm(logenergy ~ Type, data=case1002))


Call:
lm(formula = logenergy ~ Type, data = case1002)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.8872 -0.3994  0.0236  0.4932  1.5253 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                   3.396      0.422    8.04  3.4e-07
Typenon-echolocating birds   -0.609      0.488   -1.25  0.22885
Typeecholocating bats        -2.743      0.597   -4.59  0.00026

Residual standard error: 0.845 on 17 degrees of freedom
Multiple R-squared:  0.595,Adjusted R-squared:  0.548 
F-statistic: 12.5 on 2 and 17 DF,  p-value: 0.000458

> summary(lm(logenergy ~ Type * logmass, data=case1002))


Call:
lm(formula = logenergy ~ Type * logmass, data = case1002)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-0.2515 -0.1264 -0.0095  0.0812  0.3284

Coefficients:
                                   Estimate Std. Error t value Pr(>|t|)
(Intercept)                          -0.202      1.261   -0.16    0.875
Typenon-echolocating birds           -1.378      1.295   -1.06    0.305
Typeecholocating bats                -1.268      1.285   -0.99    0.341
logmass                               0.590      0.206    2.86    0.013
Typenon-echolocating birds:logmass    0.246      0.213    1.15    0.269
Typeecholocating bats:logmass         0.215      0.224    0.96    0.353

Residual standard error: 0.19 on 14 degrees of freedom
Multiple R-squared:  0.983,Adjusted R-squared:  0.977
F-statistic:  163 on 5 and 14 DF,  p-value: 6.7e-12
```

To construct the confidence bands discussed on page 282 and shown in Display 10.9 (page 283) we used the following code:

```
> pred = predict(lm1, se.fit=TRUE, newdata=data.frame(Type=c("non-echolocating birds", "non-ec
> pred.fit = pred$fit[1]; pred.fit

   1
2.28

> pred.se = pred$se.fit[1]; pred.se

     1
0.0604

> multiplier = sqrt(4*qf(.95, 4, 16)); multiplier

[1] 3.47

> lower = exp(pred.fit-pred.se*multiplier); lower

   1
7.92

> upper = exp(pred.fit+pred.se*multiplier); upper

 1
12
```

```
> # for the other reference points
> pred2 = predict(lm1, se.fit=TRUE, newdata=data.frame(Type=c("non-echolocating bats", "non-ecl
> pred3 = predict(lm1, se.fit=TRUE, newdata=data.frame(Type=c("echolocating bats", "echolocati
>
> table10.9 = rbind(c("Intercept estimate", "Standard error"), round(cbind(pred2$fit, pred2$se

  [,1]                 [,2]
  "Intercept estimate" "Standard error"
1 "2.1767"             "0.1144"
2 "3.3064"             "0.0931"
1 "2.2553"             "0.1277"
2 "3.3851"             "0.1759"
```

Next we can assess the model by evaluating the extra sums of squares F-test for testing the equality of intercepts in the parallel regression lines model as shown in Display 10.10 (page 287).

```
> lm2 = lm(logenergy ~ logmass, data=case1002)
> anova(lm2, lm1)

Analysis of Variance Table

Model 1: logenergy ~ logmass
Model 2: logenergy ~ logmass + Type
  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1     18 0.583
2     16 0.553  2    0.0296 0.43   0.66
```

We can also compare the full model with interaction terms and the reduced model (without interaction terms) with the extra sum of squares F-test as described in Display 10.12 (page 290).

```
> lm3 = lm(logenergy ~ logmass*Type, data=case1002)
> anova(lm3, lm1)

Analysis of Variance Table

Model 1: logenergy ~ logmass * Type
Model 2: logenergy ~ logmass + Type
  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1     14 0.505
2     16 0.553 -2   -0.0484 0.67   0.53
```

Another way to test the equality of the groups is by using linear combinations which we can attain using the `estimable()` command as follows. These results can be found on page 276 and 289.

```
> require(gmodels)
> estimable(lm1, c(0, 0, -1, 1))

           Estimate Std. Error t value DF Pr(>|t|)
(0 0 -1 1)  -0.0236      0.158   -0.15 16    0.883
```

The Statistical Sleuth in R:
Chapter 11

Linda Loi Kate Aloisio Ruobing Zhang Nicholas J. Horton*

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> install.packages('mosaic')              # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')              # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3, show.signif.stars=FALSE)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 11: Model Checking and Refinement using R.

2 Alcohol metabolism in men and women

How do men and women metabolise alcohol? This is the question addressed in case study 11.1 in the *Sleuth*.

2.1 Data coding, summary statistics and graphical display

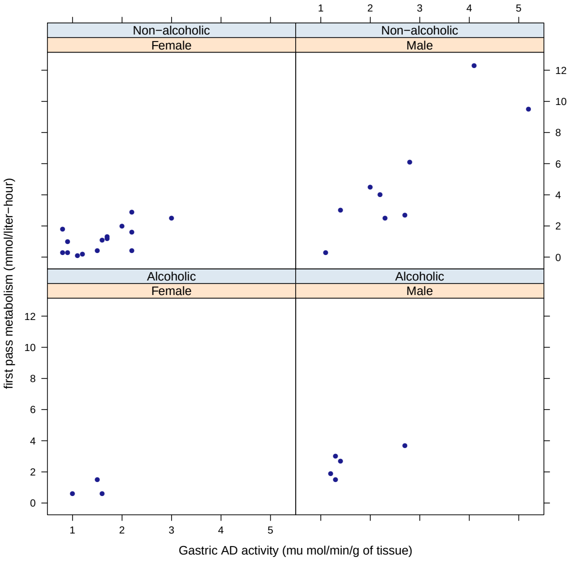
We begin by reading the data and summarizing the variables.

```
> summary(case1101)

    Subject         Metabol          Gastric          Sex    
 Min.   : 1.0   Min.   : 0.10   Min.   :0.80   Female:18  
 1st Qu.: 8.8   1st Qu.: 0.60   1st Qu.:1.20   Male  :14  
 Median :16.5   Median : 1.70   Median :1.60              
 Mean   :16.5   Mean   : 2.42   Mean   :1.86              
 3rd Qu.:24.2   3rd Qu.: 2.92   3rd Qu.:2.20              
 Max.   :32.0   Max.   :12.30   Max.   :5.20              
          Alcohol  
 Alcoholic    : 8  
 Non-alcoholic:24  
```

A total of 32 volunteers were included in this data. There were 18 females and 14 males, as recorded on Display 11.1 (page 311 of the *Sleuth*).

The following is a graphical display of the variables akin to Display 11.2 (page 312).

```
> xyplot(Metabol ~ Gastric | Sex+Alcohol, data=case1101, auto.key=TRUE,
+   xlab="Gastric AD activity (mu mol/min/g of tissue)",
+   ylab="first pass metabolism (mmol/liter-hour)")
```



2.2 Multiple regression

First we can fit a full model for estimating *metabolism* given a subjects *gastric AD activity*, whether they are *alcoholic* and *gender*. This first model is summarized on page 321 (Display 11.9).

```
> case1101 = transform(case1101, Sex = factor(Sex, levels = c("Male", "Female")))
> case1101 = transform(case1101, Alcohol = factor(Alcohol, levels = c("Non-alcoholic", "Alcohol
> lm1 = lm(Metabol ~ Gastric+Sex+Alcohol+Gastric*Sex+Sex*Alcohol+Gastric*Alcohol+Gastric*Sex*Al


Call:
lm(formula = Metabol ~ Gastric + Sex + Alcohol + Gastric * Sex +
    Sex * Alcohol + Gastric * Alcohol + Gastric * Sex * Alcohol,
    data = case1101)

Residuals:
   Min     1Q Median     3Q    Max
-2.429 -0.619 -0.047  0.515  3.652

Coefficients:
                                   Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)                             -1.660      1.000   -1.66    0.110
Gastric                                  2.514      0.343    7.32  1.5e-07
SexFemale                                1.466      1.333    1.10    0.282
AlcoholAlcoholic                         2.552      1.946    1.31    0.202
Gastric:SexFemale                       -1.673      0.620   -2.70    0.013
SexFemale:AlcoholAlcoholic              -2.252      4.394   -0.51    0.613
Gastric:AlcoholAlcoholic                -1.459      1.053   -1.39    0.179
Gastric:SexFemale:AlcoholAlcoholic       1.199      2.998    0.40    0.693

Residual standard error: 1.25 on 24 degrees of freedom
Multiple R-squared:  0.828,Adjusted R-squared:  0.777
F-statistic: 16.5 on 7 and 24 DF,  p-value: 9.35e-08
```

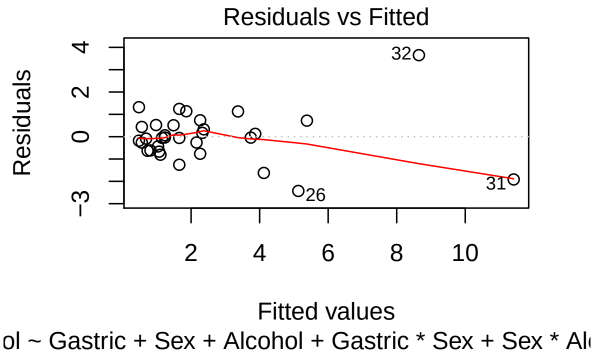
Next we can calculate a number of model diagnostics, including leverage, studentized resids and Cook's distance (pages 325-327).

```
> require(MASS)
```

```
> case1101 = transform(case1101, hat = hatvalues(lm1))
> case1101 = transform(case1101, studres = studres(lm1))
> case1101 = transform(case1101, cooks = cooks.distance(lm1))
> # display a particular row
> case1101[31,]

   Subject Metabol Gastric  Sex       Alcohol   hat studres cooks
31      31     9.5     5.2 Male Non-alcoholic 0.601   -2.72   1.1
```

The following is a residual plot for the full model akin to Display 11.7 (page 319).

```
> plot(lm1, which=1)
```



From these diagnostics it appears that observations 31 and 32 may be influential points. Therefore, we next re-fit the full model excluding these two observations. The following results are found in Display 11.9 and discussed on page 321.

```
> case11012 = case1101[-c(31, 32),]
> lm2 = lm(Metabol ~ Gastric+Sex+Alcohol+Gastric*Sex+Sex*Alcohol+Gastric*Alcohol+Gastric*Sex*Al


Call:
lm(formula = Metabol ~ Gastric + Sex + Alcohol + Gastric * Sex +
    Sex * Alcohol + Gastric * Alcohol + Gastric * Sex * Alcohol,
    data = case11012)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8076 -0.5701 -0.0466  0.4976  1.4002

Coefficients:
                                   Estimate Std. Error t value Pr(>|t|)
(Intercept)                          -0.680      1.309   -0.52   0.6088
Gastric                               1.921      0.608    3.16   0.0046
SexFemale                             0.486      1.467    0.33   0.7436
AlcoholAlcoholic                      1.572      1.812    0.87   0.3949
Gastric:SexFemale                    -1.081      0.721   -1.50   0.1483
SexFemale:AlcoholAlcoholic           -1.272      3.467   -0.37   0.7172
Gastric:AlcoholAlcoholic             -0.866      0.963   -0.90   0.3784
Gastric:SexFemale:AlcoholAlcoholic    0.606      2.316    0.26   0.7961

Residual standard error: 0.941 on 22 degrees of freedom
Multiple R-squared:  0.685,Adjusted R-squared:  0.585
F-statistic: 6.83 on 7 and 22 DF,  p-value: 0.000226
```

2.3 Refining the Model

This section addresses the process of refining the model. We first tested the lack of fit for the removal of `Alcohol` as shown in Display 11.13 (page 329).

```
> lm3 = lm(Metabol ~ Gastric+Sex+Gastric*Sex, data=case11012); summary(lm3)


Call:
lm(formula = Metabol ~ Gastric + Sex + Gastric * Sex, data = case11012)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5962 -0.6025 -0.0408  0.4759  1.6473

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)         0.0695     0.8019    0.09   0.9316
Gastric             1.5654     0.4074    3.84   0.0007
SexFemale          -0.2668     0.9932   -0.27   0.7904
Gastric:SexFemale  -0.7285     0.5394   -1.35   0.1885

Residual standard error: 0.882 on 26 degrees of freedom
Multiple R-squared:  0.673,Adjusted R-squared:  0.635
F-statistic: 17.8 on 3 and 26 DF,  p-value: 1.71e-06

> anova(lm3, lm2) # page 322

Analysis of Variance Table

Model 1: Metabol ~ Gastric + Sex + Gastric * Sex
Model 2: Metabol ~ Gastric + Sex + Alcohol + Gastric * Sex + Sex * Alcohol +
    Gastric * Alcohol + Gastric * Sex * Alcohol
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     26 20.2
2     22 19.5  4      0.74 0.21   0.93
```

Next we assessed a model without an intercept which is scientifically plausible as summarized in Display 11.14 (page 329).

```
> lm4 = lm(Metabol ~  Gastric+Gastric:Sex - 1, data=case11012); summary(lm4)


Call:
lm(formula = Metabol ~ Gastric + Gastric:Sex - 1, data = case11012)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6171 -0.6075 -0.0262  0.4772  1.6230

Coefficients: (1 not defined because of singularities)
                  Estimate Std. Error t value Pr(>|t|)
Gastric              0.726      0.121    5.99  1.9e-06
Gastric:SexMale      0.873      0.174    5.02  2.6e-05
Gastric:SexFemale       NA         NA      NA       NA

Residual standard error: 0.852 on 28 degrees of freedom
Multiple R-squared:  0.877,Adjusted R-squared:  0.868
F-statistic: 99.9 on 2 and 28 DF,  p-value: 1.8e-13

> anova(lm4, lm3)

Analysis of Variance Table
```

```
Model 1: Metabol ~ Gastric + Gastric:Sex - 1
Model 2: Metabol ~ Gastric + Sex + Gastric * Sex
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     28 20.3
2     26 20.2  2     0.094 0.06   0.94
```

Note that the “Summary of Statistical Findings” section (page 312) is based on this final model.

3 Blood brain barrier

Neuroscientists working to better understand the blood brain barrier have infused rats with cells to induce brain tumors. This is the topic addressed in case study 11.2 in the *Sleuth*.

3.1 Data coding and summary statistics

We begin by reading the data, performing transformations where needed and summarizing the variables.

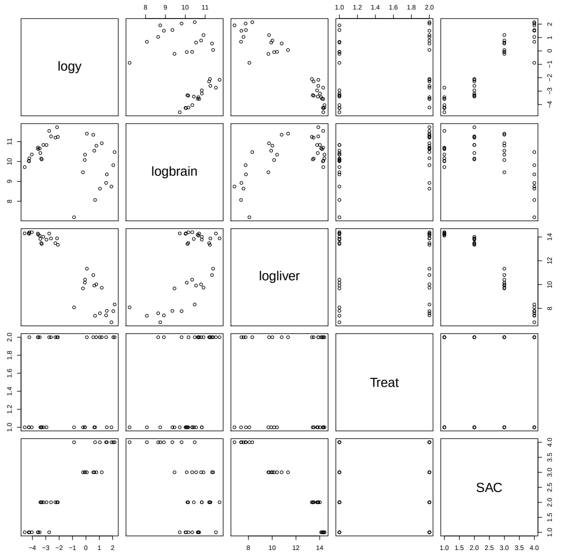
```
> names(case1102)

[1] "Brain"     "Liver"     "Time"      "Treatment" "Days"      "Sex"
[7] "Weight"    "Loss"      "Tumor"

> case1102 = transform(case1102, Y = Brain/Liver)
> case1102 = transform(case1102, logliver = log(Liver))
> case1102 = transform(case1102, logbrain = log(Brain))
> case1102 = transform(case1102, SAC = as.factor(Time))
> case1102 = transform(case1102, logy = log(Brain/Liver))
> case1102 = transform(case1102, logtime = log(Time))
> case1102 = transform(case1102, Treat = relevel(Treatment, ref="NS"))
> summary(case1102)

     Brain            Liver              Time       Treatment      Days
 Min.   :  1334   Min.   :    928   Min.   : 0.5   BD:17     Min.   : 9
 1st Qu.: 19281   1st Qu.:  16210   1st Qu.: 1.1   NS:17     1st Qu.:10
 Median : 32572   Median : 643965   Median : 3.0             Median :10
 Mean   : 39965   Mean   : 668776   Mean   :23.5             Mean   :10
 3rd Qu.: 50654   3rd Qu.:1318557   3rd Qu.:24.0             3rd Qu.:10
 Max.   :123730   Max.   :1790863   Max.   :72.0             Max.   :11
     Sex          Weight          Loss            Tumor            Y
 Female:26   Min.   :184   Min.   :-4.90   Min.   : 25   Min.   :0.01
 Male  : 8   1st Qu.:225   1st Qu.: 1.20   1st Qu.:136   1st Qu.:0.03
             Median :240   Median : 3.95   Median :166   Median :0.12
             Mean   :242   Mean   : 3.64   Mean   :183   Mean   :1.50
             3rd Qu.:259   3rd Qu.: 5.97   3rd Qu.:223   3rd Qu.:1.95
```

```
              Max.   :298   Max.   :12.80   Max.   :484   Max.   :8.55
    logliver        logbrain       SAC          logy           logtime
 Min.   : 6.83   Min.   : 7.20   0.5:9   Min.   :-4.58   Min.   :-0.69
 1st Qu.: 9.69   1st Qu.: 9.86   3  :9   1st Qu.:-3.39   1st Qu.:-0.25
 Median :13.37   Median :10.39   24 :8   Median :-2.13   Median : 1.10
 Mean   :11.61   Mean   :10.23   72 :8   Mean   :-1.39   Mean   : 1.86
 3rd Qu.:14.09   3rd Qu.:10.83           3rd Qu.: 0.67   3rd Qu.: 3.18
 Max.   :14.40   Max.   :11.73           Max.   : 2.15   Max.   : 4.28
 Treat
 NS:17
 BD:17
```

A total of 34 rats were included in this experiment. Each rat was given either the barrier solution (n = 17) or a normal saline solution (n = 17). Then variables of interest were calculated and are displayed in Display 11.4 (page 314 of the *Sleuth*).

We can graphically relationships between the variables using a pairs plot.

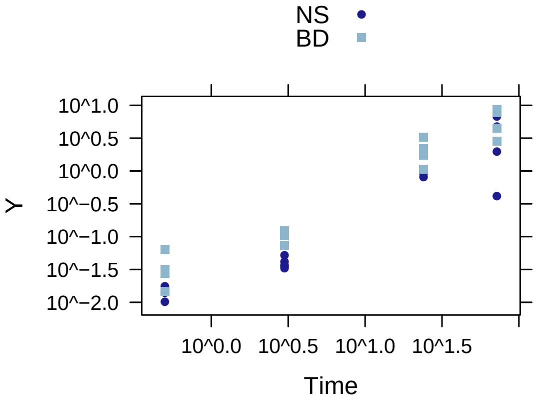
```
> smallds = subset(case1102, select=c("logy", "logbrain","logliver","Treat", "SAC"))
> pairs(smallds)
```



3.2 Graphical presentation

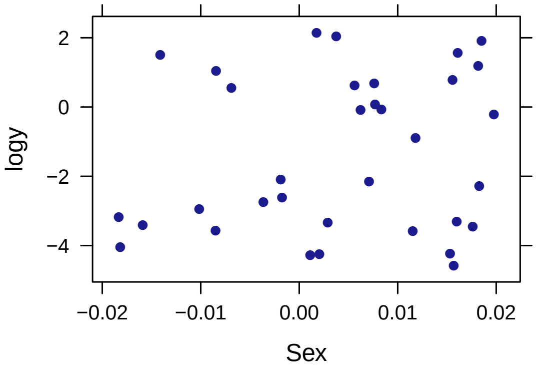
The following displays a scatterplot of log ratio (Y) as a function of log time, akin to Display 11.5 on page 315.

```
> xyplot(Y ~ Time, group=Treat, scales=list(y=list(log=TRUE),
+   x=list(log=TRUE)), auto.key=TRUE, data=case1102)
```

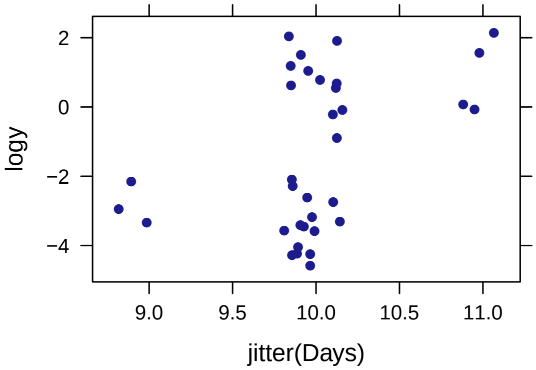


The following graphs are akin to the second and third plots in Display 11.16 on page 333.

```
> case1102=transform(case1102, female = ifelse(Sex=="F", 1, 0))
> xyplot(logy ~ jitter(female), xlab="Sex", data=case1102)
```



```
> xyplot(logy ~ jitter(Days), data=case1102)
```



3.3 Multiple regression

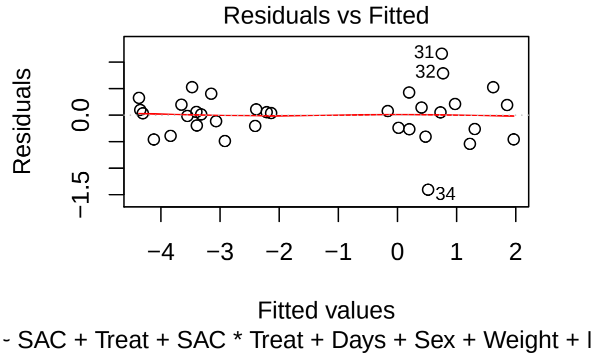
We first fit a model that reflects the initial investigation. This is the proposed model from page 317.

```
> lm1 = lm(logy ~ SAC+Treat+SAC*Treat+Days+Sex+
+   Weight+Loss+Tumor, data=case1102); summary(lm1)


Call:
lm(formula = logy ~ SAC + Treat + SAC * Treat + Days + Sex +
    Weight + Loss + Tumor, data = case1102)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4056 -0.2559  0.0458  0.1957  1.1583

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.836741   3.391046   -1.13    0.271
SAC3           1.015463   0.399578    2.54    0.019
SAC24          4.337135   0.477836    9.08  1.0e-08
SAC72          5.010605   0.454953   11.01  3.5e-10
TreatBD        0.795999   0.378970    2.10    0.048
Days          -0.036987   0.295645   -0.13    0.902
SexMale        0.001295   0.373368    0.00    0.997
Weight        -0.000558   0.005330   -0.10    0.918
Loss          -0.059544   0.030422   -1.96    0.064
Tumor          0.001551   0.001226    1.26    0.220
SAC3:TreatBD   0.179831   0.551964    0.33    0.748
SAC24:TreatBD -0.386047   0.585450   -0.66    0.517
SAC72:TreatBD  0.379104   0.569242    0.67    0.513
```

```
Residual standard error: 0.564 on 21 degrees of freedom
Multiple R-squared:  0.96,Adjusted R-squared:  0.937
F-statistic: 41.9 on 12 and 21 DF,  p-value: 6.45e-12
```

We can then display a residual plot to assess the fit of the above model. This is provided in Display 11.6 (page 318).

```
> plot(lm1, which=1)
```



3.4 Refining the model

Lastly, we fit a refined model. These results can be found in Display 11.17 (page 334).

```
> lm2 = lm(logy ˜ SAC+Treat, data=case1102); summary(lm2)


Call:
lm(formula = logy ˜ SAC + Treat, data = case1102)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.7402 -0.1755 -0.0178  0.2477  1.0551 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -4.302      0.205  -21.01  < 2e-16
SAC3           1.134      0.252    4.50  0.00010
SAC24          4.257      0.259   16.43  3.1e-16
SAC72          5.154      0.259   19.89  < 2e-16
TreatBD        0.797      0.183    4.35  0.00016

Residual standard error: 0.533 on 29 degrees of freedom
```

```
Multiple R-squared:  0.951,Adjusted R-squared:  0.944
F-statistic:  140 on 4 and 29 DF,  p-value: <2e-16

> anova(lm2, lm1)

Analysis of Variance Table

Model 1: logy ~ SAC + Treat
Model 2: logy ~ SAC + Treat + SAC * Treat + Days + Sex + Weight + Loss +
    Tumor
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     29 8.23
2     21 6.68  8      1.55 0.61   0.76
```

The Statistical Sleuth in R: Chapter 12

Linda Loi   Kate Aloisio   Ruobing Zhang   Nicholas J. Horton[*]

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

[*] Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> install.packages('mosaic')               # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')               # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=4)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 12: Strategies for Variable Selection using R.

2 State Average SAT Scores

What variables are associated with state SAT scores? This is the question addressed in case study 12.1 in the *Sleuth*.

2.1 Summary statistics

We begin by reading the data and summarizing the variables.

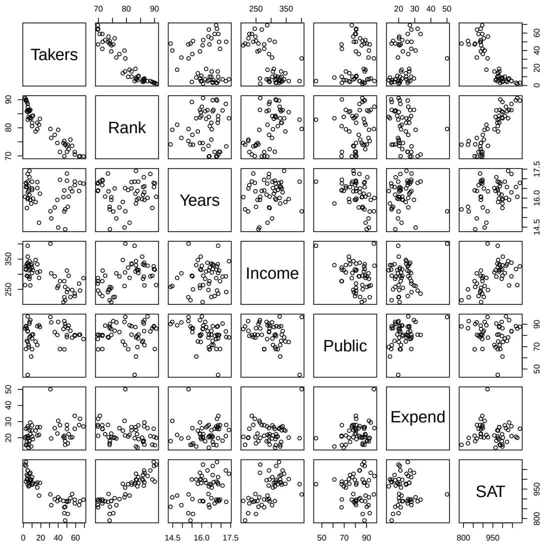
```
> summary(case1201)

        State         SAT             Takers           Income
 Alabama   : 1   Min.   : 790   Min.   : 2.00   Min.   :208
 Alaska    : 1   1st Qu.: 889   1st Qu.: 6.25   1st Qu.:262
 Arizona   : 1   Median : 966   Median :16.00   Median :295
 Arkansas  : 1   Mean   : 948   Mean   :26.22   Mean   :294
 California: 1   3rd Qu.: 998   3rd Qu.:47.75   3rd Qu.:325
 Colorado  : 1   Max.   :1088   Max.   :69.00   Max.   :401
 (Other)   :44
     Years          Public          Expend          Rank
 Min.   :14.4   Min.   :44.8   Min.   :13.8   Min.   :69.8
 1st Qu.:15.9   1st Qu.:76.9   1st Qu.:19.6   1st Qu.:74.0
 Median :16.4   Median :80.8   Median :21.6   Median :80.8
 Mean   :16.2   Mean   :81.2   Mean   :23.0   Mean   :80.0
 3rd Qu.:16.8   3rd Qu.:88.2   3rd Qu.:26.4   3rd Qu.:85.8
 Max.   :17.4   Max.   :97.0   Max.   :50.1   Max.   :90.6
```

The data are shown on page 347 (display 12.1). A total of 50 state average SAT scores are included in this data.

2.2 Dealing with Many Explanatory Variables

The following graph is presented as Display 12.4, page 356.

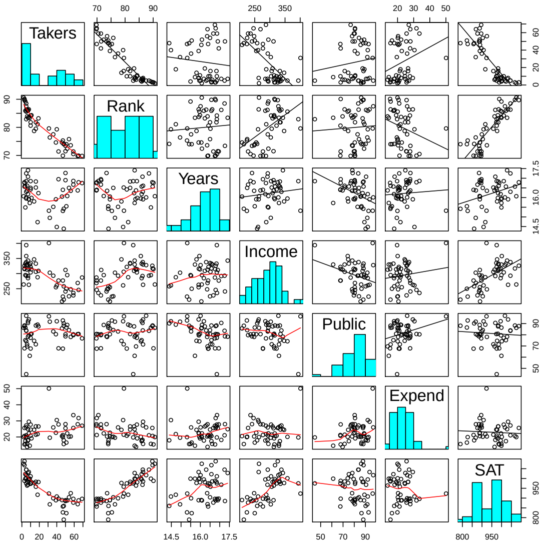
```
> pairs(~ Takers+Rank+Years+Income+Public+Expend+SAT, data=case1201)
```



We can get a fancier graph using following code:

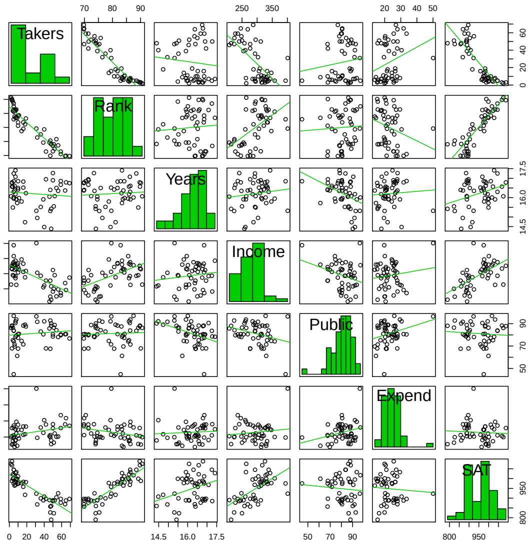
```
>   panel.hist = function(x, ...)
+   {
+     usr = par("usr"); on.exit(par(usr))
+     par(usr = c(usr[1:2], 0, 1.5) )
+     h = hist(x, plot=FALSE)
+     breaks = h$breaks; nB = length(breaks)
+     y = h$counts; y = y/max(y)
+     rect(breaks[-nB], 0, breaks[-1], y, col="cyan", ...)
+   }
>
> panel.lm = function(x, y, col=par("col"), bg=NA,
+                     pch=par("pch"), cex=1, col.lm="red", ...)
+ {
+   points(x, y, pch=pch, col=col, bg=bg, cex=cex)
+   ok = is.finite(x) & is.finite(y)
+   if (any(ok))
+     abline(lm(y[ok] ~ x[ok]))
+ }
```

```
> pairs(~ Takers+Rank+Years+Income+Public+Expend+SAT,
+       lower.panel=panel.smooth, diag.panel=panel.hist,
+       upper.panel=panel.lm, data=case1201)
```



An alternative graph can be generated using the car package.

```
> require(car)
> scatterplotMatrix(~ Takers+Rank+Years+Income+Public+Expend+SAT, diagonal="histogram", smooth
```



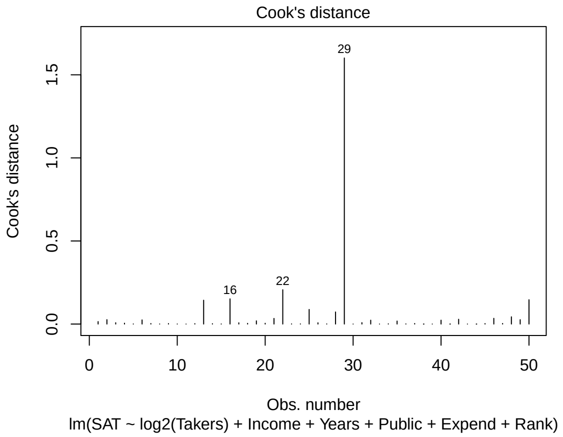
Based on the scatterplot, we choose the logarithm of percentage of SAT takers and median class rank to fit our first model (page 355-357):

```
> lm1 = lm(SAT ~ Rank+log(Takers), data=case1201)
> summary(lm1)


Call:
lm(formula = SAT ~ Rank + log(Takers), data = case1201)

Residuals:
   Min     1Q Median     3Q    Max
-94.46 -17.31   5.32  22.82  48.47

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   882.08     224.13    3.94  0.00027
Rank            2.40       2.33    1.03  0.30898
log(Takers)   -45.19      14.06   -3.21  0.00236

Residual standard error: 31.1 on 47 degrees of freedom
Multiple R-squared:  0.815,Adjusted R-squared:  0.807
F-statistic:  103 on 2 and 47 DF,  p-value: <2e-16
```

From the regression output, we observe that these two variables can explain 81.5% of the variation.

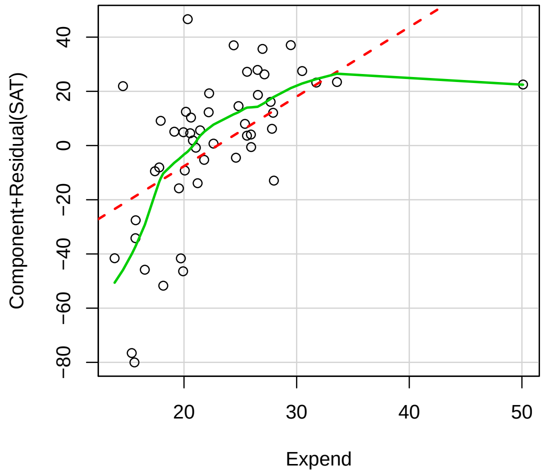
Next we fit a linear regression model using all variables and create the partial residual plot presented on page 357 as Display 12.5:

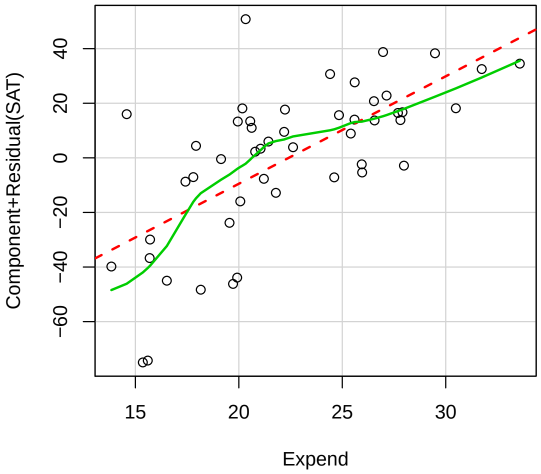
```
> lm2 = lm(SAT ~ log2(Takers)+Income+Years+Public+Expend+Rank, data=case1201)
> summary(lm2)


Call:
lm(formula = SAT ~ log2(Takers) + Income + Years + Public + Expend +
    Rank, data = case1201)

Residuals:
   Min      1Q Median      3Q     Max
-61.11  -8.60   2.86  14.77  53.40

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  407.5399   282.7633    1.44   0.1567
log2(Takers) -26.6429    11.0572   -2.41   0.0203
Income        -0.0359     0.1301   -0.28   0.7841
Years         17.2181     6.3201    2.72   0.0093
Public        -0.1130     0.5624   -0.20   0.8417
Expend         2.5669     0.8064    3.18   0.0027
Rank           4.1143     2.5017    1.64   0.1073

Residual standard error: 24.9 on 43 degrees of freedom
Multiple R-squared:  0.892,Adjusted R-squared:  0.877
F-statistic: 59.2 on 6 and 43 DF,  p-value: <2e-16

> plot(lm2, which=4)
```



According to the Cook's distance plot, obs 29 (Alaska) seems to be an influential outlier. We may consider removing this observation from the dataset.

```
> case1201r = case1201[-c(29),]
> lm3 = lm(SAT ~ log2(Takers) + Income+ Years + Public + Expend + Rank, data=case1201r)
> anova(lm3)

Analysis of Variance Table

Response: SAT
             Df Sum Sq Mean Sq F value  Pr(>F)
log2(Takers)  1 199007  199007  390.63 < 2e-16
Income        1    785     785    1.54  0.2214
Years         1   5910    5910   11.60  0.0015
Public        1   5086    5086    9.98  0.0029
Expend        1  10513   10513   20.64 4.6e-05
Rank          1   2679    2679    5.26  0.0269
Residuals    42  21397     509

> crPlots(lm2, term = ~ Expend) # with Alaska
> crPlots(lm3, term = ~ Expend) # without Alaska
```





The difference between these two slopes indicates that Alaska is an influential observation. We decide to remove it from the original dataset.

2.3 Sequential Variable Selection

The book uses F-statistics as the criterion to perform the procedures of forward selection and backward elimination presented on page 359. As mentioned on page 359, the entire forward selection procedure required the fitting of only 16 of the 64 possible models presented on Display 12.6 (page 360). These 16 models utilized Expenditure and log(Takers) to predict SAT scores.Further, as mentioned on page 359, the entire backward selection procedure required the fitting of only 3 models of the 64 possible models. These 3 models used Year, Expenditure, Rank and log(Takers) to predict SAT scores.

To the best of our knowledge, RStudio is not equipped to perform stepwise regressions using F-statistics. Instead, we demonstrate this proceduring using AIC criterion and get the final model using the following code. Note that we choose log(Taker) as our preliminary predictor for forward selection, because it has the largest F-value when we fitted lm3.

```
> # Forward Selection
> lm4 = lm(SAT ~ log2(Takers), data=case1201r)
> stepAIC(lm4, scope=list(upper=lm3, lower=~1),
+   direction="forward", trace=FALSE)$anova

Stepwise Model Path
Analysis of Deviance Table

Initial Model:
SAT ~ log2(Takers)

Final Model:
SAT ~ log2(Takers) + Expend + Years + Rank


      Step Df Deviance Resid. Df Resid. Dev   AIC
1                             47      46369 339.8
2 + Expend  1    20523        46      25846 313.1
3  + Years  1     1248        45      24598 312.7
4   + Rank  1     2676        44      21922 309.1

> # Backward Elimination
> stepAIC(lm3, direction="backward", trace=FALSE)$anova

Stepwise Model Path
Analysis of Deviance Table

Initial Model:
SAT ~ log2(Takers) + Income + Years + Public + Expend + Rank

Final Model:
SAT ~ log2(Takers) + Years + Expend + Rank


      Step Df Deviance Resid. Df Resid. Dev   AIC
1                             42      21397 311.9
2 - Public  1     20.0        43      21417 309.9
3 - Income  1    505.4        44      21922 309.1
```

```
> # Stepwise Regression
> stepAIC(lm3, direction="both", trace=FALSE)$anova

Stepwise Model Path
Analysis of Deviance Table

Initial Model:
SAT ~ log2(Takers) + Income + Years + Public + Expend + Rank

Final Model:
SAT ~ log2(Takers) + Years + Expend + Rank


      Step Df Deviance Resid. Df Resid. Dev   AIC
1                             42      21397 311.9
2 - Public  1     20.0        43      21417 309.9
3 - Income  1    505.4        44      21922 309.1
```

Thus, the final model includes log(Takers), Expenditure, Years and Rank.

```
> lm5 = lm(SAT ~ log2(Takers) + Expend + Years + Rank, data=case1201r)
> summary(lm5)


Call:
lm(formula = SAT ~ log2(Takers) + Expend + Years + Rank, data = case1201r)

Residuals:
   Min     1Q Median     3Q    Max
-52.30  -9.92   0.60  11.88  59.20

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   399.115    232.372    1.72   0.0929
log2(Takers)  -26.409      8.259   -3.20   0.0026
Expend          3.996      0.764    5.23  4.5e-06
Years          13.147      5.478    2.40   0.0207
Rank            4.400      1.899    2.32   0.0252

Residual standard error: 22.3 on 44 degrees of freedom
Multiple R-squared:  0.911,Adjusted R-squared:  0.903
F-statistic:  112 on 4 and 44 DF,  p-value: <2e-16
```

The final model can explain 91.1% percent or the variation of SAT. All of the explanatory variables are statistically significant at the $\alpha = .05$ level.

2.4 Model Selection Among All Subsets

The Cp-statistic can be an useful criterion to select model among all subsets. We'll give an example about how to calculate this statistic for one model, which includes log(Takers), Expenditure, Years and Rank.

```
> sigma5 = summary(lm5)$sigma^2 # sigma-squared of chosen model
> sigma3 = summary(lm3)$sigma^2 # sigma-squared of full model
> n = 49 # sample size
> p = 4+1 # number of coefficients in model
> Cp=(n-p)*sigma5/sigma3+(2*p-n)
> Cp

[1] 4.031
```

The Cp statistic for this model is 4.0312.

Alternatively, the Cp statistic can be calculated using the following command:

```
> require(leaps)
> explanatory = with(case1201r, cbind(log(Takers), Income, Years, Public, Expend, Rank))
> with(case1201r, leaps(explanatory, SAT, method="Cp"))$which[27,]

    1     2     3     4     5     6
 TRUE FALSE  TRUE FALSE  TRUE  TRUE
```

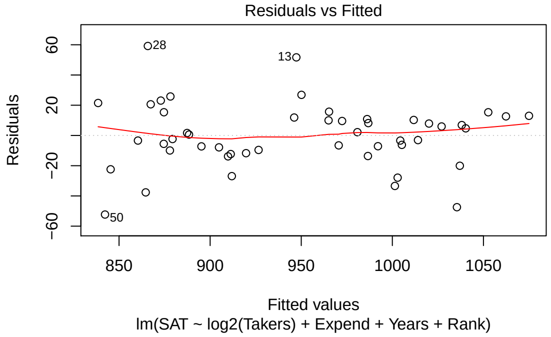
This means that the 27th fitting model includes log(Takers), Years and Expend.

```
> with(case1201r, leaps(explanatory, SAT, method="Cp"))$Cp[27]

[1] 4.031
```

The Cp statistic for this model is 4.0312. This will be the the ”tyer” point on the Display 12.9, page 365.

We use the following code to generate the graph presented as Display 12.14 on page 372.

```
> plot(lm5, which=1)
```



From the scatterplot, we see that obs 28 (New Hampshire) has the largest residual, while obs 50 (Sorth Carolina) has the smallest.

2.5 Contribution of Expend

Display 12.13 (page 363) shows the contribution of Expend to the model.

```
> lm7 = lm(SAT ~ Expend, data=case1201r)
> summary(lm7)


Call:
lm(formula = SAT ~ Expend, data = case1201r)

Residuals:
   Min     1Q Median     3Q    Max 
-162.5  -57.7   17.0   46.6  141.4 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  961.724     49.888   19.28   <2e-16
Expend        -0.592      2.178   -0.27     0.79

Residual standard error: 72.2 on 47 degrees of freedom
Multiple R-squared:  0.00157,Adjusted R-squared:  -0.0197 
F-statistic: 0.074 on 1 and 47 DF,  p-value: 0.787

> lm8 = lm(SAT ~ Income + Expend, data=case1201r)
> summary(lm8)


Call:
lm(formula = SAT ~ Income + Expend, data = case1201r)

Residuals:
   Min     1Q Median     3Q    Max 
-91.15 -38.41  -2.58  27.29 159.52 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  604.682     73.209    8.26  1.2e-10
Income         1.127      0.196    5.73  7.2e-07
Expend         0.672      1.695    0.40     0.69

Residual standard error: 55.7 on 46 degrees of freedom
Multiple R-squared:  0.418,Adjusted R-squared:  0.392 
F-statistic: 16.5 on 2 and 46 DF,  p-value: 3.95e-06
```

3 Sex Discrimination in Employment

Do females receive lower starting salaries than similarly qualified and similarly experience males and did females receive smaller pay increases than males? These are the questions explored in case 12.2 in the *Sleuth*.

3.1 Summary Statistics

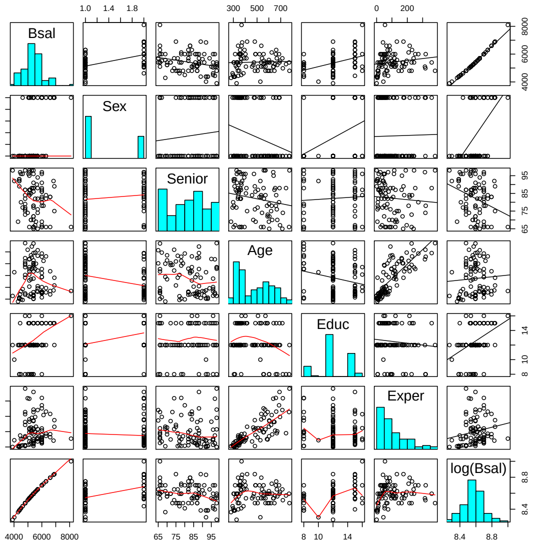
We begin by summarizing the data.

```
> summary(case1202)

      Bsal           Sal77            Sex         Senior           Age       
 Min.   :3900   Min.   : 7860   Female:61   Min.   :65.0   Min.   :280  
 1st Qu.:4980   1st Qu.: 9000   Male  :32   1st Qu.:74.0   1st Qu.:349  
 Median :5400   Median :10020               Median :84.0   Median :468  
 Mean   :5420   Mean   :10393               Mean   :82.3   Mean   :474  
 3rd Qu.:6000   3rd Qu.:11220               3rd Qu.:90.0   3rd Qu.:590  
 Max.   :8100   Max.   :16320               Max.   :98.0   Max.   :774  
      Educ           Exper      
 Min.   : 8.0   Min.   :  0.0  
 1st Qu.:12.0   1st Qu.: 35.5  
 Median :12.0   Median : 70.0  
 Mean   :12.5   Mean   :100.9  
 3rd Qu.:15.0   3rd Qu.:144.0  
 Max.   :16.0   Max.   :381.0  
```

The data is shown on page 350-351 as display 12.3. A total of 93 employee salaries are included: 61 females and 32 males.

Next we present a full graphical display for the variables within the dataset and the log of the beginning salary variable.

```
> pairs(~ Bsal+Sex+Senior+Age+Educ+Exper+log(Bsal),
+       lower.panel=panel.smooth, diag.panel=panel.hist,
+       upper.panel=panel.lm, data=case1202)
```



Through these scatterplots it appears that beginning salary should be on the log scale and the starting model without the effects of gender will be a saturated second-order model with 14 variables including Seniority, Age, Education, Experience, as main effects, quadratic terms, and their full interactions.

3.2 Model Selection

To determine the best subset of these variables we first compared Cp statistics. Display 12.11 shows the Cp statistics for models that meet 'good practice' and have small Cp values. We will demonstrate how to calculate the Cp statistics for the two models with the lowest Cp statistics discussed in "Identifying Good Subset Models" on pages 367-368.

The first model includes Seniority, Age, Education, Experience, and the interactions between Seniority and Education, Age and Education, and Age and Experience. The second model includes Seniority, Age, Education, Experience, and the interactions between Age and Education and Age and Experience.

```
> require(leaps)
> explanatory1 = with(case1202, cbind(Senior, Age, Educ, Exper, Senior*Educ, Age*Educ, Age*Expe
> # First model (saexnck)
> with(case1202, leaps(explanatory1, log(Bsal), method="Cp"))$which[55,]

   1    2    3    4    5    6    7
TRUE TRUE TRUE TRUE TRUE TRUE TRUE

> with(case1202, leaps(explanatory1, log(Bsal), method="Cp"))$Cp[55]

[1] 8

> # second model (saexck)
> with(case1202, leaps(explanatory1, log(Bsal), method="Cp"))$which[49,]
```

```
    1     2     3     4     5     6     7 
 TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE 

> with(case1202, leaps(explanatory1, log(Bsal), method="Cp"))$Cp[49]

[1] 8.124
```

This first model has a Cp statistic of 8. Compared to the second model with a Cp statistic of 8.12.

We can also compare models using the BIC, we will next compare the second model with a thrid model defined as *saexyc* = Seniority + Age + Education + Experience + Experience^2 + Age*Education.

```
> BIC(lm(log(Bsal) ~ Senior+Age+Educ+Exper+Age*Educ+Age*Exper, data=case1202))

[1] -140.2

> BIC(lm(log(Bsal) ~ Senior+Age+Educ+Exper+(Exper)^2+Age*Educ, data=case1202))

[1] -131.3
```

Thus our final model is the second model, summarized below.

```
> lm1 = lm(log(Bsal) ~ Senior + Age + Educ + Exper + Age*Educ + Age*Exper, data=case1202)
> summary(lm1)


Call:
lm(formula = log(Bsal) ~ Senior + Age + Educ + Exper + Age * 
    Educ + Age * Exper, data = case1202)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.2817 -0.0476  0.0132  0.0605  0.2341 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.89e+00   2.45e-01   32.21  < 2e-16
Senior      -3.15e-03   1.04e-03   -3.04  0.00313
Age          1.24e-03   4.02e-04    3.09  0.00270
Educ         7.20e-02   1.67e-02    4.31  4.3e-05
Exper        2.86e-03   6.67e-04    4.28  4.8e-05
Age:Educ    -1.02e-04   3.15e-05   -3.25  0.00166
Age:Exper   -3.72e-06   1.02e-06   -3.65  0.00044

Residual standard error: 0.0974 on 86 degrees of freedom
```

```
Multiple R-squared:  0.469,Adjusted R-squared:  0.431
F-statistic: 12.6 on 6 and 86 DF,  p-value: 3.58e-10
```

3.3 Evaluating the Sex Effect

After selecting the model *saexck* = Seniority + Age + Education + Experience + Age*Education + Age*Experience we can add the sex indicator variable as summarized on page 360.

```
> lm2 = lm(log(Bsal) ~ Senior + Age + Educ + Exper + Age*Educ + Age*Exper + Sex, data=case1202)
> summary(lm2)


Call:
lm(formula = log(Bsal) ~ Senior + Age + Educ + Exper + Age *
    Educ + Age * Exper + Sex, data = case1202)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17822 -0.05197 -0.00203  0.05301  0.20466

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.16e+00   2.21e-01   36.99  < 2e-16
Senior      -3.48e-03   9.09e-04   -3.83  0.00024
Age          9.15e-04   3.57e-04    2.56  0.01218
Educ         4.23e-02   1.57e-02    2.70  0.00836
Exper        2.18e-03   5.98e-04    3.65  0.00045
SexMale      1.20e-01   2.29e-02    5.22  1.3e-06
Age:Educ    -5.46e-05   2.91e-05   -1.88  0.06402
Age:Exper   -3.23e-06   8.96e-07   -3.61  0.00052

Residual standard error: 0.0853 on 85 degrees of freedom
Multiple R-squared:  0.598,Adjusted R-squared:  0.564
F-statistic:   18 on 7 and 85 DF,  p-value: 1.79e-14
```

In contrast to the book, our reference group is Male, therefore the median male salary is estimated to be 1.13 times as large as the median female salary, adjusted for the other variables.

The Statistical Sleuth in R:
Chapter 13

Linda Loi   Kate Aloisio   Ruobing Zhang   Nicholas J. Horton*

June 15, 2016

Contents

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Third Edition of the *Statistical Sleuth* (2013) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/sleuth3`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> install.packages('mosaic')                    # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth3` package.

```
> install.packages('Sleuth3')                    # note the quotation marks
```

```
> require(Sleuth3)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=4, show.signif.stars=FALSE)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 13: The Analysis of Variance for Two-Way Classifications using R.

2 Intertidal seaweed grazers

This wicked complicated trial is a subset of a factorial design (6 of the possible 2 by 2 by 2 combination of factors) plus blocking. This randomized block design is analyzed in case study 13.1 in the *Sleuth*.

2.1 Data coding, summary statistics and graphical display

We begin by reading the data, performing the necessary transformations and summarizing the variables.

```
> # logit transformation
> case1301$logitcover = with(case1301, log(Cover/(100-Cover)))
```

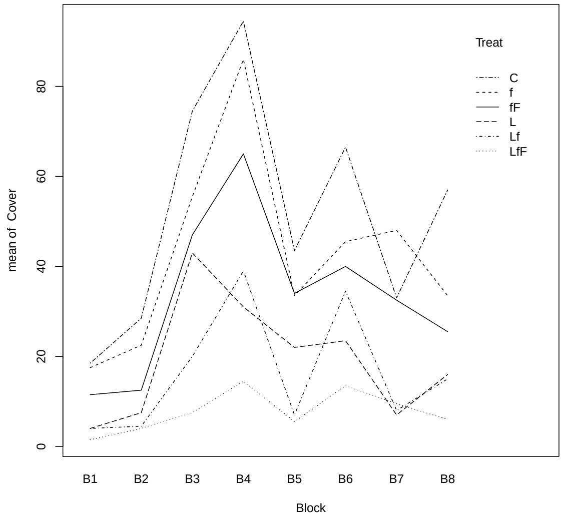
```
> summary(case1301)

     Cover           Block     Treat      logitcover   
 Min.   : 1.0   B1     :12   C  :16   Min.   :-4.595  
 1st Qu.: 9.0   B2     :12   L  :16   1st Qu.:-2.314  
 Median :22.5   B3     :12   Lf :16   Median :-1.237  
 Mean   :28.6   B4     :12   LfF:16   Mean   :-1.233  
 3rd Qu.:42.2   B5     :12   f  :16   3rd Qu.:-0.313  
 Max.   :95.0   B6     :12   fF :16   Max.   : 2.944  
                (Other):24                            
```

```
> favstats(logitcover~Treat, data=case1301)

  Treat    min      Q1  median       Q3     max    mean     sd  n missing
1     C -1.815 -0.7995  0.1201  0.80579  2.9444  0.1805 1.3990 16       0
2     L -3.178 -2.4784 -1.6964 -0.90838  0.3228 -1.7120 1.0215 16       0
3    Lf -3.476 -2.9444 -2.1530 -1.25519  0.2819 -2.0044 1.1399 16       0
4   LfF -4.595 -2.9444 -2.7515 -2.28453 -1.2657 -2.7247 0.8310 16       0
5     f -2.091 -0.8119 -0.4898  0.09007  2.0907 -0.3137 1.0748 16       0
6    fF -2.197 -1.7762 -0.5325 -0.30237  0.9946 -0.8214 0.9599 16       0
```

There were a total of 96 rock plots free of seaweed. These plots where split into 8 blocks based on location. Each block contained 12 plots. Then 6 treatments were randomly assigned to plots within each block. Therefore there were two plots per treatment within each block, as shown in Display 13.2 (page 387 of the *Sleuth*).

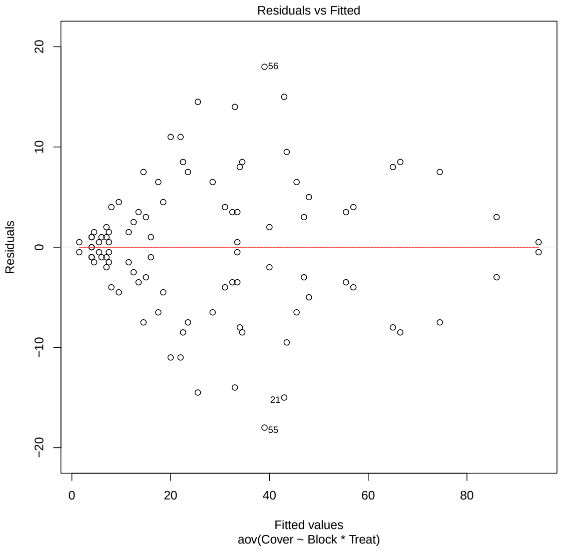
We can check for evidence of nonadditivity using interaction plots. For a figure akin to Display 13.7 on page 393 we can use the following code:

```
> with(case1301, interaction.plot(Block, Treat, Cover))
```



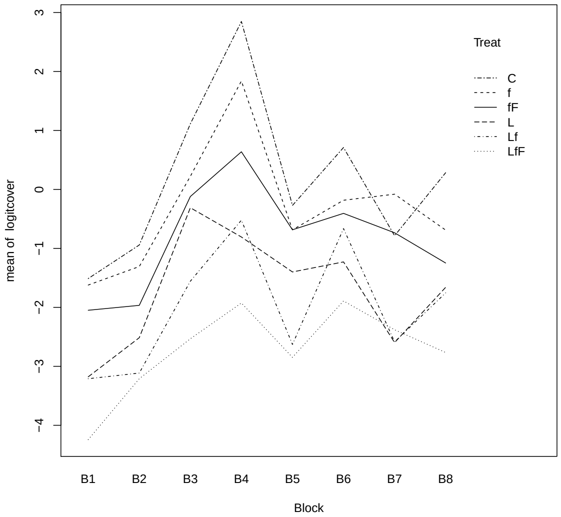
This figure shows evidence of nonadditivity. However as the authors note the type of nonadditivity seen in this figure may be removed by transformations. In addition, the residual plot from the saturated model (shown below and is akin to Display 13.8 on page 394) has a distinct funnel shape, also indicating a need for transformation.

```
> plot(aov(Cover ~ Block*Treat, data=case1301), which=1)
```



After the log transformation, we can then observe an interaction plot on the log transformed data akin to Display 13.9 on page 395.

```
> with(case1301, interaction.plot(Block, Treat, logitcover))
```



2.2 Models

Then we can create an ANOVA for the nonadditive model estimating the log of the seaweed regeneration ratio as summarized on page 395 (Display 13.10).

```
> anova(lm(logitcover ~ Block*Treat, data=case1301))
```

```
Analysis of Variance Table

Response: logitcover
            Df Sum Sq Mean Sq F value Pr(>F)
Block        7   76.2   10.89   35.96 <2e-16
Treat        5   97.0   19.40   64.06 <2e-16
Block:Treat 35   15.2    0.44    1.44   0.12
Residuals   48   14.5    0.30
```

This model has an R^2 of 92.84%, an adjusted R^2 of 85.83%, and an estimated SD of 0.5503. Notice that the interaction term has a large p-value, 0.1209, suggesting that the data may be more consistent with an additive model.

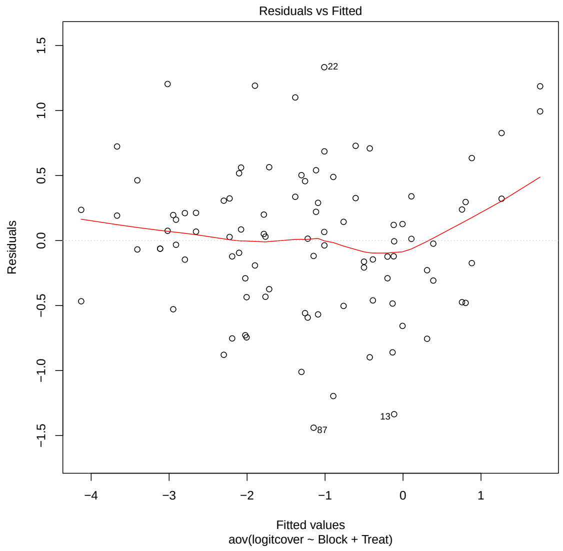
We can then compare these results to an ANOVA for the additive model estimating the log of the seaweed regeneration ratio as shown in Display 13.11 on page 397.

```
> anova(lm(logitcover ~ Block+Treat, data=case1301))

Analysis of Variance Table

Response: logitcover
          Df Sum Sq Mean Sq F value Pr(>F)
Block      7   76.2   10.89    30.4 <2e-16
Treat      5   97.0   19.40    54.1 <2e-16
Residuals 83   29.8    0.36
```

This model has an R^2 of 85.34%, an adjusted R^2 of 83.22%, and an estimated SD of 0.5989.

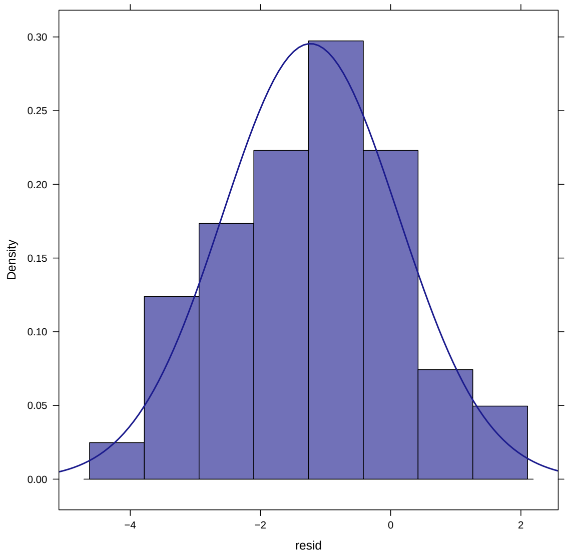
Next we can assess the fit of the additive model through diagnostic plots. First we can check the linearity assumption.

```
> plot(aov(logitcover ~ Block+Treat, data=case1301), which=1)
```



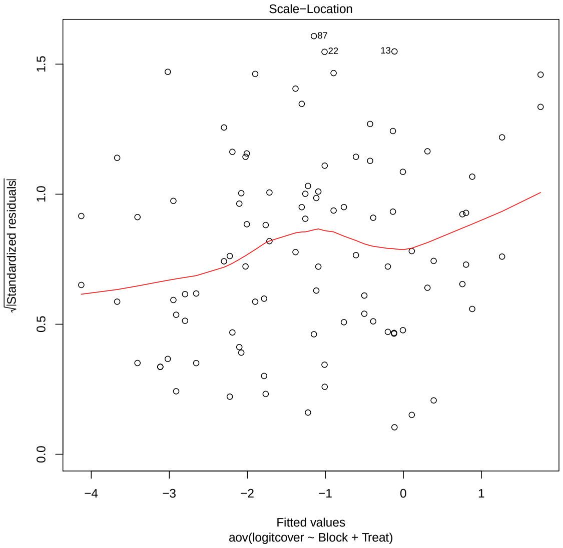
From this plot is appears that the linearity assumption seems reasonable.
We will need to assume independence based on the information given.
Next we will assess the normality assumption for the additive model.

```
> case1301$resid = predict(aov(logitcover ~ Block+Treat, data=case1301))
> histogram(~ resid, type='density', density=TRUE, data=case1301)
```



From this figure normality seems reasonable as well.
Now we can assess equality of variance.

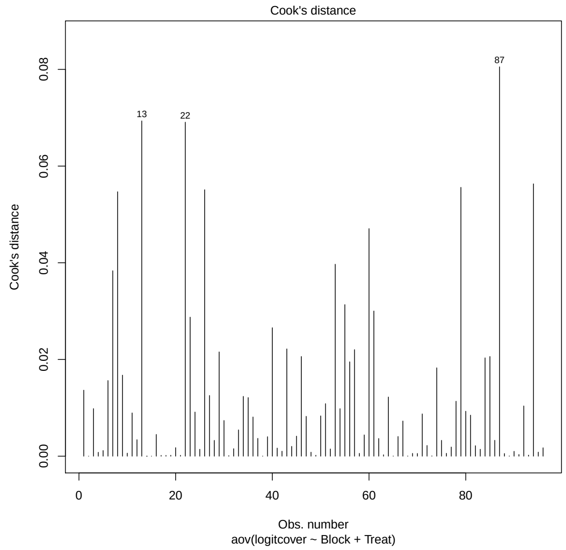
```
> plot(aov(logitcover ~ Block+Treat, data=case1301), which=3)
```



From this figure, the assumption of equal variance seems to be somewhat problematic, as seen in the curvature of the lowess line.

Lastly we can look for influential points and/or high leverage with the additive model.

```
> plot(aov(logitcover ~ Block+Treat, data=case1301), which=4)
```



From this figure we can obtain certain plots that appear to be influential points.

```
> case1301[c(13, 22, 87),]

   Cover Block Treat logitcover   resid
13    19    B7     C    -1.4500 -0.1141
22    58    B3     L     0.3228 -1.0105
87     7    B4   LfF    -2.5867 -1.1471
```

2.3 Linear combinations

First we can observe the Block and Treatment averages and the Block and Treatment effects from Display 13.12 (page 398).

For the effects we used:

```
> model.tables(aov(lm(logitcover ~ Block*Treat, data=case1301)), type="effects")

Tables of effects

 Block
Block
     B1      B2      B3      B4      B5      B6      B7      B8
-1.4031 -0.9432  0.7015  1.5776 -0.1871  0.6220 -0.2946 -0.0731

 Treat
```

```
Treat
      C       L      Lf     LfF       f      fF 
 1.4131 -0.4794 -0.7718 -1.4921  0.9190  0.4112 

 Block:Treat 
     Treat
Block C       L       Lf      LfF     f       fF     
   B1 -0.2892 -0.0629  0.1972 -0.1157  0.0951  0.1755
   B2 -0.1797  0.1406 -0.1663  0.4576 -0.0509 -0.2013
   B3  0.2303  0.6996 -0.2540 -0.5094 -0.1658 -0.0007
   B4  1.0899 -0.6724 -0.0947 -0.7791  0.5743 -0.1179
   B5 -0.2650  0.4996 -0.4376  0.0638 -0.1850  0.3241
   B6 -0.0918 -0.1392  0.7185  0.2112 -0.4920 -0.2067
   B7 -0.6709 -0.5903 -0.2862  0.6394  0.5274  0.3807
   B8  0.1763  0.1250  0.3231  0.0322 -0.3030 -0.3536
```

For the means we changed the `type` attribute to `"means"`:

```
> model.tables(aov(lm(logitcover ~ Block*Treat, data=case1301)), type="means")

Tables of means
Grand mean
          
-1.233 

 Block 
Block
     B1      B2      B3      B4      B5      B6      B7      B8 
-2.6357 -2.1758 -0.5311  0.3450 -1.4197 -0.6106 -1.5272 -1.3057 

 Treat 
Treat
      C       L      Lf     LfF       f      fF 
 0.1805 -1.7120 -2.0044 -2.7247 -0.3137 -0.8214 

 Block:Treat 
     Treat
Block C      L      Lf     LfF    f      fF    
   B1 -1.512 -3.178 -3.210 -4.243 -1.622 -2.049
   B2 -0.942 -2.515 -3.114 -3.210 -1.308 -1.966
   B3  1.112 -0.311 -1.557 -2.533  0.222 -0.121
   B4  2.848 -0.807 -0.522 -1.926  1.838  0.638
   B5 -0.272 -1.399 -2.629 -2.848 -0.686 -0.684
   B6  0.711 -1.229 -0.664 -1.891 -0.184 -0.406
   B7 -0.785 -2.597 -2.585 -2.380 -0.081 -0.735
   B8  0.284 -1.660 -1.754 -2.766 -0.690 -1.248
```

To answer specific questions of interest regarding subgroup comparisons we can use linear combinations. The *Sleuth* proposes five questions as detailed on pages 299-400. The code for results of these questions is displayed below and these results are also interpreted on pages 399-400 and summarized in Display 13.13. For this model the reference group is *control* followed by *f, fF, L, Lf, LfF*.

```
> require(gmodels)
> lm1 = lm(logitcover ~ Treat+Block, data=case1301); coef(lm1)

(Intercept)      TreatL     TreatLf    TreatLfF      Treatf     TreatfF
    -1.2226     -1.8925     -2.1849     -2.9052     -0.4941     -1.0019
    BlockB2     BlockB3     BlockB4     BlockB5     BlockB6     BlockB7
     0.4600      2.1046      2.9807      1.2160      2.0251      1.1085
    BlockB8
     1.3300

> large = rbind('Large fish' = c(0, 0, -1/2, 1/2, -1/2, 1/2))
> small = rbind('Small fish' = c(-1/2, -1/2, 1/2, 0, 1/2, 0))
> limpets = rbind('Limpets' = c(-1/3, 1/3, 1/3, 1/3, -1/3, -1/3))
> limpetsSmall = rbind('Limpets X Small' = c(1, -1, 1/2, 1/2, -1/2, -1/2))
> limpetsLarge = rbind('Limpets X Large' = c(0, 0, -1, 1, 1, -1))
> fit.contrast(lm1, "Treat", large, conf.int=.95)

                Estimate Std. Error t value Pr(>|t|) lower CI upper CI
TreatLarge fish   -0.614     0.1497  -4.101 9.54e-05  -0.9118  -0.3162

> fit.contrast(lm1, "Treat", small, conf.int=.95)

                Estimate Std. Error t value Pr(>|t|) lower CI upper CI
TreatSmall fish  -0.3933     0.1497  -2.627  0.01026   -0.691 -0.09549

> fit.contrast(lm1, "Treat", limpets, conf.int=.95)

             Estimate Std. Error t value  Pr(>|t|) lower CI upper CI
TreatLimpets   -1.829     0.1222  -14.96 2.778e-25   -2.072   -1.586

> fit.contrast(lm1, "Treat", limpetsSmall, conf.int=.95)

                     Estimate Std. Error t value Pr(>|t|) lower CI
TreatLimpets X Small  0.09549     0.2593  0.3682   0.7136  -0.4203
                     upper CI
TreatLimpets X Small   0.6113

> fit.contrast(lm1, "Treat", limpetsLarge, conf.int=.95)

                     Estimate Std. Error t value Pr(>|t|) lower CI
TreatLimpets X Large  -0.2125     0.2994 -0.7097   0.4799  -0.8081
                     upper CI
TreatLimpets X Large    0.383
```

To attain the confidence intervals discussed in the “Summary of Statistical Findings” (page 386) we need to exponential the lower and upper bounds of the above 95% confidence intervals. Therefore, for the limpets estimation, the corresponding 95% confidence interval is (0.126, 0.205). The resulting large fish 95% confidence interval is (0.402, 0.729). Lastly for the estimation of the regeneration ratio for small fish the 95% confidence interval is (0.501, 0.909).

3 Pygmalion effect

Does telling a manager that some of the supervisees are superior affect their perceived performance? This is the question addressed in case study 13.2 in the *Sleuth*.

3.1 Statistical summary

We begin by reading the data and summarizing the variables.

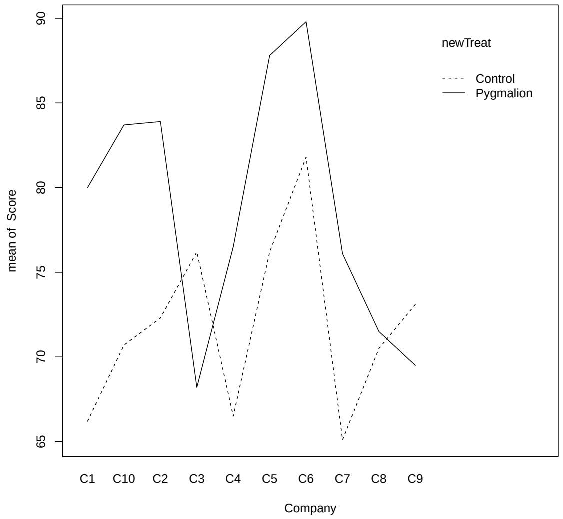
```
> summary(case1302)

   Company          Treat         Score
 C1      : 3   Control  :19   Min.   :59.5
 C10     : 3   Pygmalion:10   1st Qu.:69.2
 C2      : 3                  Median :73.9
 C4      : 3                  Mean   :74.1
 C5      : 3                  3rd Qu.:78.9
 C6      : 3                  Max.   :89.8
 (Other):11

> case1302$newTreat = relevel(case1302$Treat, ref="Control")
```

There were a total of 29 platoons. For each of the 10 companies, one platoon received the Pygmalion treatment and two platoons were control, with the exception of one company that only had one control platoon. Therefore, there were 10 Pygmalion platoons and 19 control platoons. As shown in Display 13.3 (page 388 of the *Sleuth*).

3.2 Graphical presentation

The following figure displays an interaction plot for the Pygmalion dataset, akin to Display 13.14 on page 402.

```
> with(case1302, interaction.plot(Company, newTreat, Score))
```



3.3 Two way ANOVA (fit using multiple linear regression model)

We can then use multiple linear regression models for the additive and nonadditive models and compare them using the two-way ANOVA.

The following is similar to Display 13.16 (page 404).

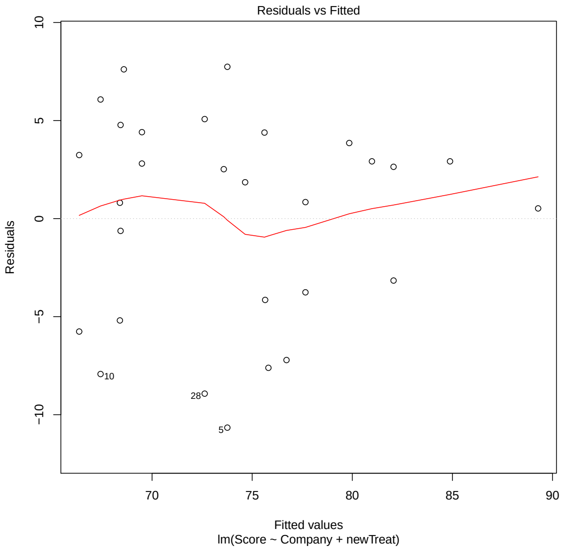
```
> lm1 = lm(Score ~ Company*newTreat, data=case1302); summary(lm1)


Call:
lm(formula = Score ~ Company * newTreat, data = case1302)

Residuals:
   Min     1Q Median     3Q    Max 
  -9.2   -2.3    0.0    2.3    9.2 

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                     66.20       5.09   13.00  3.9e-07
CompanyC10                       4.50       7.20    0.62    0.548
CompanyC2                        6.10       7.20    0.85    0.419
CompanyC3                       10.00       8.82    1.13    0.286
CompanyC4                        0.30       7.20    0.04    0.968
CompanyC5                       10.00       7.20    1.39    0.198
CompanyC6                       15.60       7.20    2.17    0.059
CompanyC7                       -1.10       7.20   -0.15    0.882
CompanyC8                        4.30       7.20    0.60    0.565
CompanyC9                        6.90       7.20    0.96    0.363
```

```
newTreatPygmalion               13.80       8.82    1.56    0.152
CompanyC10:newTreatPygmalion    -0.80      12.48   -0.06    0.950
CompanyC2:newTreatPygmalion     -2.20      12.48   -0.18    0.864
CompanyC3:newTreatPygmalion    -21.80      13.48   -1.62    0.140
CompanyC4:newTreatPygmalion     -3.80      12.48   -0.30    0.768
CompanyC5:newTreatPygmalion     -2.20      12.48   -0.18    0.864
CompanyC6:newTreatPygmalion     -5.80      12.48   -0.46    0.653
CompanyC7:newTreatPygmalion     -2.80      12.48   -0.22    0.827
CompanyC8:newTreatPygmalion    -12.80      12.48   -1.03    0.332
CompanyC9:newTreatPygmalion    -17.40      12.48   -1.39    0.197

Residual standard error: 7.2 on 9 degrees of freedom
Multiple R-squared:  0.739,Adjusted R-squared:  0.188
F-statistic: 1.34 on 19 and 9 DF,  p-value: 0.336

> lm2 = lm(Score ~ Company+newTreat, data=case1302); summary(lm2) # Display 13.18 page 406


Call:
lm(formula = Score ~ Company + newTreat, data = case1302)

Residuals:
   Min     1Q Median     3Q    Max 
-10.66  -4.15   1.85   3.85   7.74 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        68.3932     3.8931   17.57  8.9e-13
CompanyC10          4.2333     5.3697    0.79    0.441
CompanyC2           5.3667     5.3697    1.00    0.331
CompanyC3           0.1966     6.0189    0.03    0.974
CompanyC4          -0.9667     5.3697   -0.18    0.859
CompanyC5           9.2667     5.3697    1.73    0.102
CompanyC6          13.6667     5.3697    2.55    0.020
CompanyC7          -2.0333     5.3697   -0.38    0.709
CompanyC8           0.0333     5.3697    0.01    0.995
CompanyC9           1.1000     5.3697    0.20    0.840
newTreatPygmalion   7.2205     2.5795    2.80    0.012

Residual standard error: 6.58 on 18 degrees of freedom
Multiple R-squared:  0.565,Adjusted R-squared:  0.323
F-statistic: 2.33 on 10 and 18 DF,  p-value: 0.0564

> anova(lm1)

Analysis of Variance Table
```

```
Response: Score
                 Df Sum Sq Mean Sq F value Pr(>F)
Company           9    671      75    1.44  0.299
newTreat          1    339     339    6.53  0.031
Company:newTreat  9    311      35    0.67  0.722
Residuals         9    467      52

> anova(lm2)

Analysis of Variance Table

Response: Score
          Df Sum Sq Mean Sq F value Pr(>F)
Company    9    671      75    1.72  0.156
newTreat   1    339     339    7.84  0.012
Residuals 18    779      43

> anova(lm2, lm1)

Analysis of Variance Table

Model 1: Score ~ Company + newTreat
Model 2: Score ~ Company * newTreat
  Res.Df RSS Df Sum of Sq    F Pr(>F)
1     18 779
2      9 467  9       312 0.67   0.72
```

Lastly we can observe the residual plot from the fit of the additive model, akin to Display 13.17 on page 405.

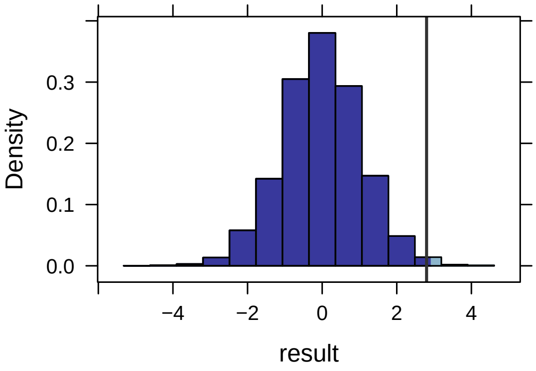
```
> plot(lm2, which=1)
```



3.4 Randomization Methods

As introduced in Chapter 4, we can construct a randomization distribution by considering the distribution of a test statistic over all possible ways the randomization could have turned out. For the Pygmalion data we can construct a randomization distribution for the *t*-statistic of the treatment effect as discussed on pages 407-408.

```
> obs = summary(lm(Score ~ Company+newTreat, data=case1302))$coefficients["newTreatPygmalion",
> nulldist = do(10000) * summary(lm(Score ~ shuffle(Company)+shuffle(newTreat), data=case1302)
> histogram(~ result, groups=result >= obs, v=obs, data=nulldist) # akin to Display 13.20 page
> tally(~ result >= obs, format="proportion", data=nulldist)


  TRUE  FALSE
0.0056 0.9944
```



From this simulation we observed that the proportion of t-statistics that were as extreme or more extreme than our observed t-statistic (2.799) is 0.0056.