

The Statistical Sleuth in R:

Chapter 5

Kate Aloisio Ruobing Zhang Nicholas J. Horton*

January 24, 2019

Contents

1	Introduction	1
2	Diet and lifespan	2
2.1	Summary statistics and graphical display	2
2.2	One-way ANOVA	4
2.3	Pairwise comparisons	4
2.4	Other analyses	6
2.5	Residual analysis and diagnostics	7
3	Spock Conspiracy Trial	8
3.1	Summary statistics and graphical display	8
3.2	One-way ANOVA	10
3.3	Additional analyses	11
3.3.1	Kruskal-Wallis Nonparametric Analysis of Variance	13

1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Second Edition of the *Statistical Sleuth* (2002) 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.amherst.edu/~nhorton/sleuth>.

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, Amherst College, nhorton@amherst.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 **Sleuth2** package.

```
> install.packages('Sleuth2') # note the quotation marks
```

```
> require(Sleuth2)
```

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)

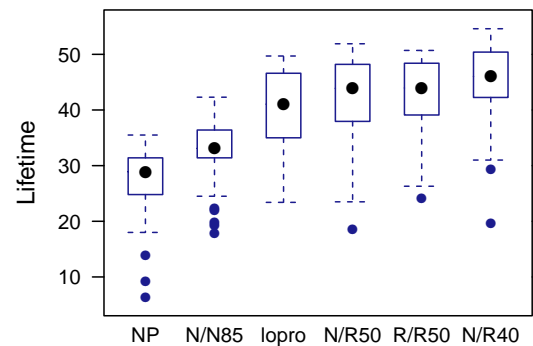
      Lifetime      Diet
Min.   : 6.4    NP    :49
1st Qu.:31.8   N/N85:57
Median :39.5   lopro:56
Mean   :38.8   N/R50:71
3rd Qu.:46.9   R/R50:56
Max.   :54.6   N/R40:60

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

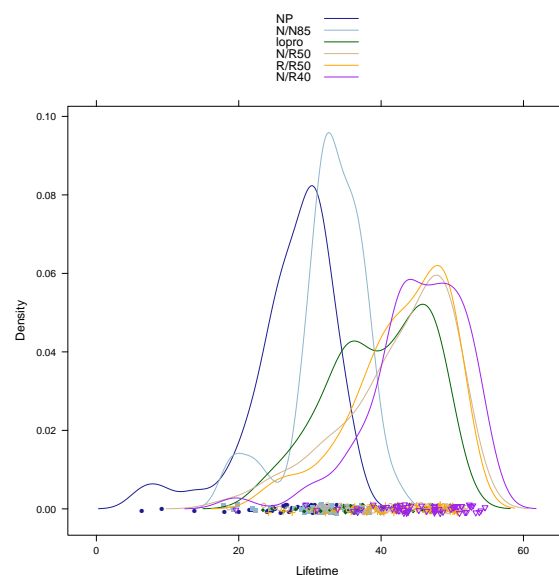
	Diet	min	Q1	median	Q3	max	mean	sd	n	missing
1	NP	6.4	24.8	28.9	31.4	35.5	27.4	6.13	49	0
2	N/N85	17.9	31.4	33.1	36.4	42.3	32.7	5.13	57	0
3	lopro	23.4	35.0	41.0	46.4	49.7	39.7	6.99	56	0
4	N/R50	18.6	38.0	43.9	48.2	51.9	42.3	7.77	71	0
5	R/R50	24.2	39.2	44.0	48.3	50.7	42.9	6.68	56	0
6	N/R40	19.6	42.3	46.0	50.3	54.6	45.1	6.70	60	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 evidence of a highly 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)	27.402	0.954	28.72	< 2e-16
DietN/N85	5.289	1.301	4.07	5.9e-05
Dietlopro	12.284	1.306	9.40	< 2e-16
DietN/R50	14.895	1.240	12.01	< 2e-16
DietR/R50	15.484	1.306	11.85	< 2e-16
DietN/R40	17.715	1.286	13.78	< 2e-16

```
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 121, 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(0, -1, 0, 1, 0, 0), conf.int=0.95)

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

```

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, 0, -1, 1, 0), conf.int=0.95)

              Estimate Std. Error t value Pr(>|t|) lower CI
Diet c=( 0 0 0 -1 1 0 )      0.589      1.19   0.493   0.622   -1.76
              upper CI
Diet c=( 0 0 0 -1 1 0 )      2.94
attr("class")
[1] "fit_contrast"

```

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

```

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

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

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

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

```

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, -1, 0, 0), conf.int=0.95)

              Estimate Std. Error t value Pr(>|t|) lower CI
Diet c=( 0 0 1 -1 0 0 )   -2.61      1.19  -2.19  0.0293  -4.96
              upper CI
Diet c=( 0 0 1 -1 0 0 )   -0.264
attr(,"class")
[1] "fit_contrast"
```

The results for (e) on page 116:

```
> # N/N85 vs NP (e)
> 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 )    5.29      1.3   4.07 5.95e-05   2.73
              upper CI
Diet c=( -1 1 0 0 0 0 )    7.85
attr(,"class")
[1] "fit_contrast"
```

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

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

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

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

2.4 Other analyses

We will next demonstrate how to calculate the quantities on page 120 (Display 5.6).

```

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

[1] 343

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

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

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

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

> 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.

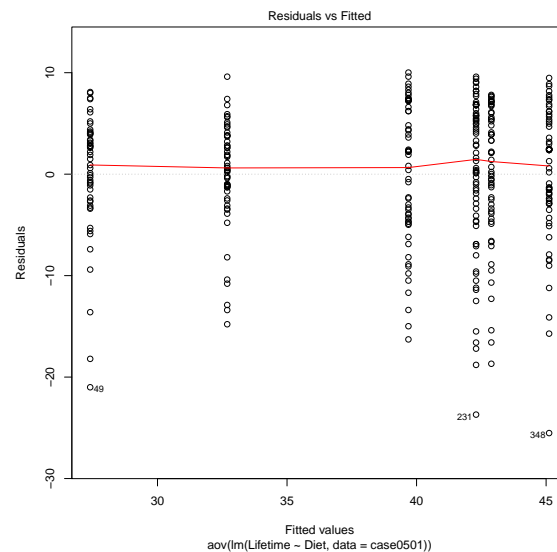
2.5 Residual analysis and diagnostics

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).

```

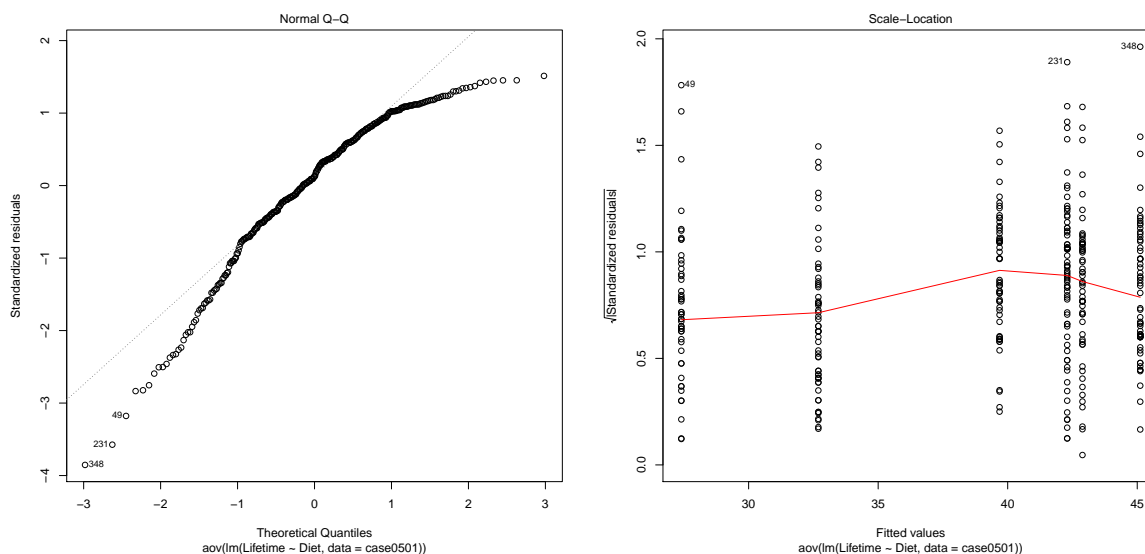
> aov1 = aov(lm(Lifetime ~ Diet, data=case0501))
> plot(aov1, which=1)

```



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.

```
> summary(case0502)
```

Percent	Judge
Min. : 6.4	Spock's:9
1st Qu.:20.0	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))
```

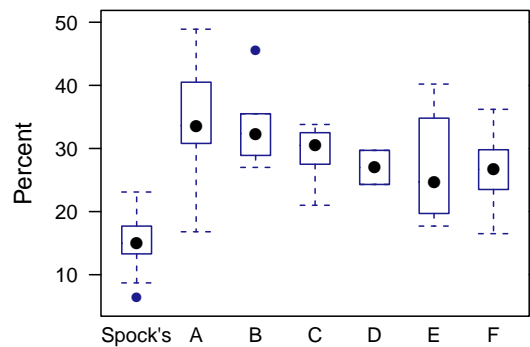
```
> favstats(Percent ~ Judge, data=case0502)
```

	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

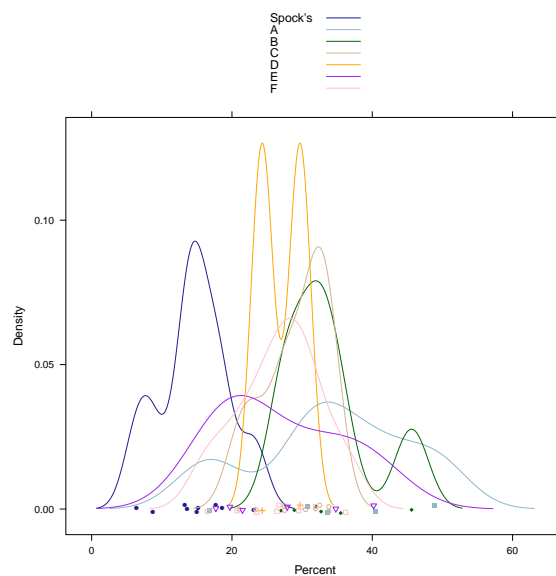
2	A	16.8	30.8	33.6	40.5	48.9	34.1	11.94	5	0
3	B	27.0	29.7	32.4	34.8	45.6	33.6	6.58	6	0
4	C	21.0	27.5	30.5	32.5	33.8	29.1	4.59	9	0
5	D	24.3	25.6	27.0	28.4	29.7	27.0	3.82	2	0
6	E	17.7	20.2	24.7	33.1	40.2	27.0	9.01	6	0
7	F	16.5	23.5	26.7	29.8	36.2	26.8	5.97	9	0

There were a total of 46 venires. They compared Spock's judge with 6 other judges. The percent 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(twoJudge), data=case0502), lm(Percent ~ as.factor(Judge), data=
Analysis of Variance Table

Model 1: Percent ~ as.factor(twoJudge)
Model 2: Percent ~ as.factor(Judge)
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     44 2191
2     39 1864  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=
          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
attr(,"class")
[1] "fit_contrast"

```

```
> # 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