

The Statistical Sleuth in R:

Chapter 13

Kate Aloisio

Ruobing Zhang

Nicholas J. Horton*

October 27, 2024

Contents

1	Introduction	1
2	Intertidal seaweed grazers	2
2.1	Data coding, summary statistics and graphical display	2
2.2	Models	5
2.3	Linear combinations	8
3	Pygmalion effect	11
3.1	Statistical summary	11
3.2	Graphical presentation	11
3.3	Two way ANOVA (fit using multiple linear regression model)	12
3.4	Randomization Methods	15

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>).

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, Amherst College, nhorton@amherst.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 **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 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 = transform(case1301, logitcover = 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 f :16	1st Qu.: -2.314
Median :22.5	B3	:12 fF :16	Median : -1.237
Mean :28.6	B4	:12 L :16	Mean : -1.233
3rd Qu.:42.2	B5	:12 Lf :16	3rd Qu.: -0.313
Max. :95.0	B6	:12 LfF:16	Max. : 2.944
	(Other):24		

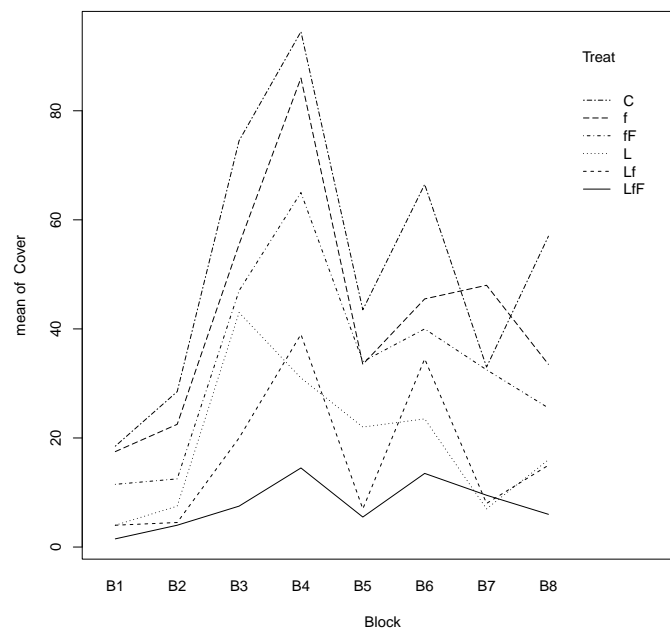
```
> favstats(logitcover~Treat, data=case1301)
```

	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	f	-2.091	-0.8119	-0.4898	0.09007	2.0907	-0.3137	1.0748	16	0
3	fF	-2.197	-1.7762	-0.5325	-0.30237	0.9946	-0.8214	0.9599	16	0
4	L	-3.178	-2.4784	-1.6964	-0.90838	0.3228	-1.7120	1.0215	16	0
5	Lf	-3.476	-2.9444	-2.1530	-1.25519	0.2819	-2.0044	1.1399	16	0
6	LfF	-4.595	-2.9444	-2.7515	-2.28453	-1.2657	-2.7247	0.8310	16	0

There were a total of 96 rock plots free of seaweed. These plots were 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 377 of the *Sleuth*).

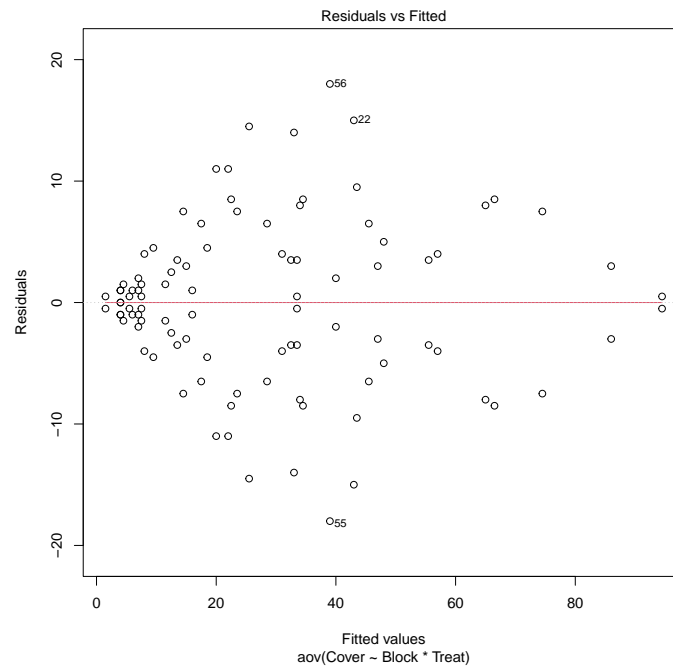
We can check for evidence of nonadditivity using interaction plots. For a figure akin to Display 13.7 on page 383 we can use the following code:

```
> with(case1301, interaction.plot(Block, Treat, Cover))
```



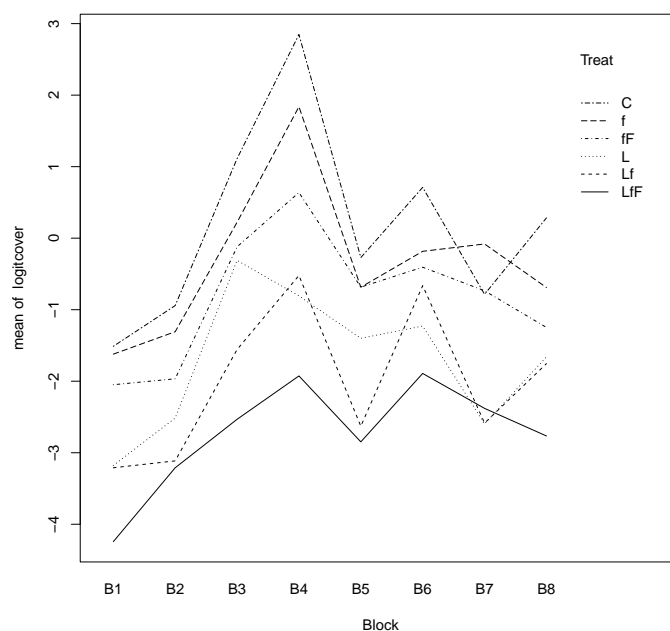
This figure shows evidence of nonadditivity. However as the authors note the type of nonadditivity seen in this figure may be removed by transformations. In addition, the residual plot from the saturated model (shown below and is akin to Display 13.8 on page 384) has a distinct funnel shape, also indicating 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 385.

```
> 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 385 (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 387.

```
> anova(lm(logitcover ~ Block+Treat, data=case1301))
```

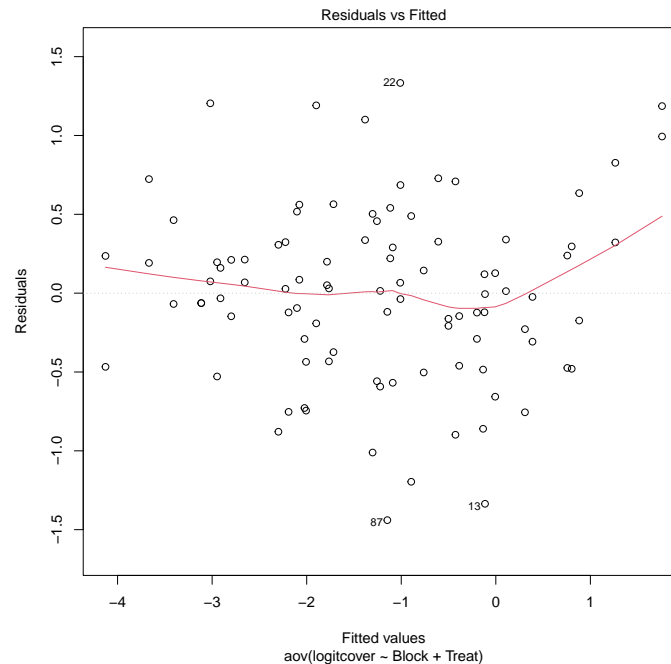
Analysis of Variance Table

```
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)
```

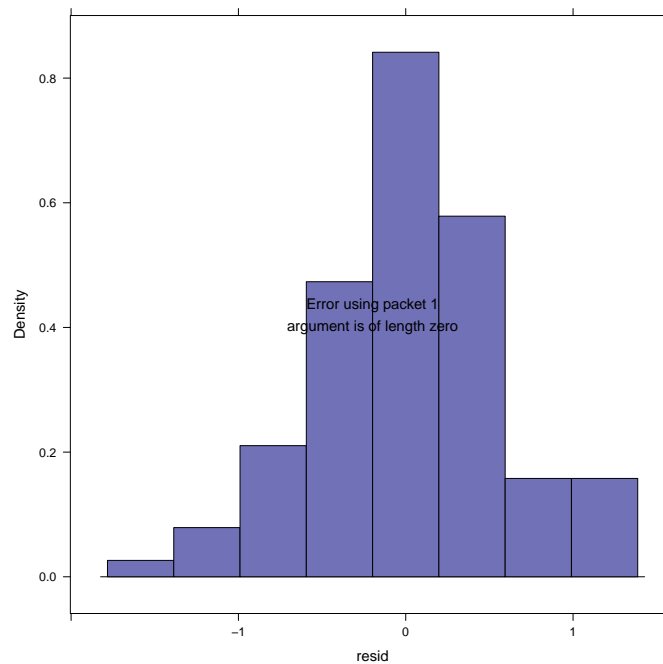


From this plot it appears that the linearity assumption seems reasonable.

We will need to assume independence based on the information given.

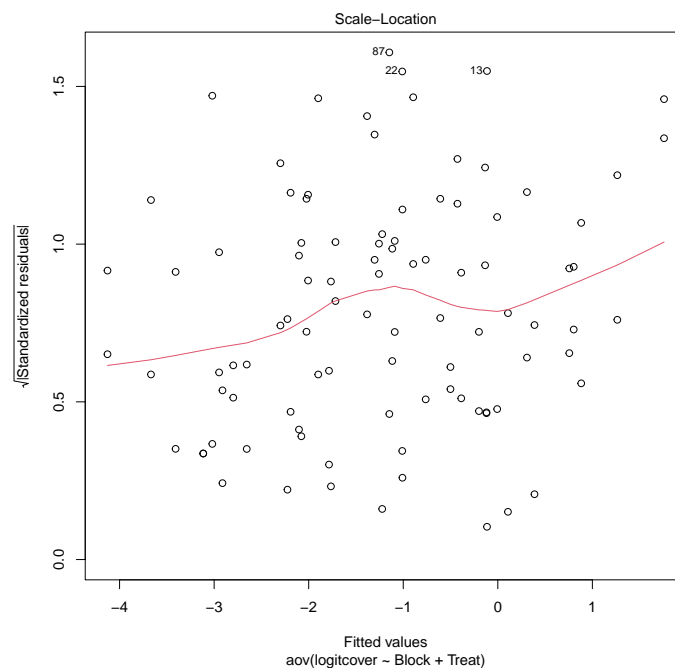
Next we will assess the normality assumption for the additive model.

```
> case1301$resid = residuals(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)
```

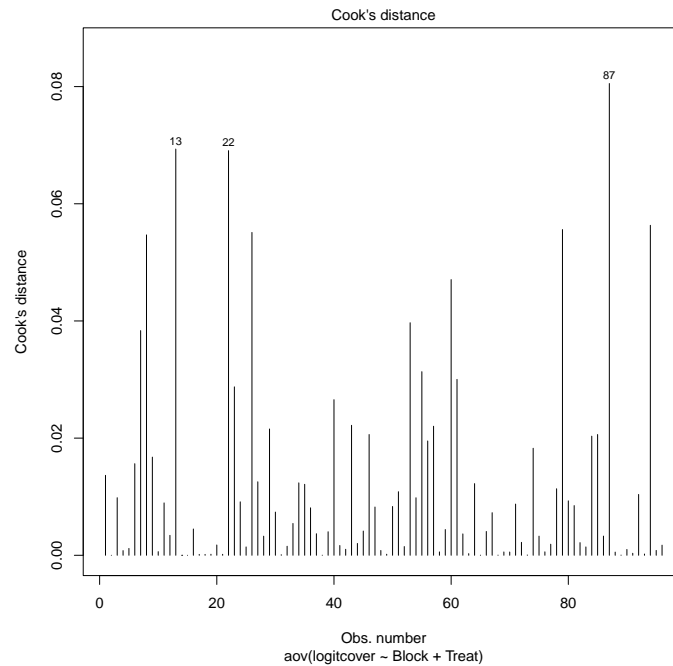


From this figure, the assumption of equal variance seems to be somewhat problematic, as seen

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	-1.336
22	58	B3	L	0.3228	1.333
87	7	B4	LfF	-2.5867	-1.440

2.3 Linear combinations

First we can observe the Block and Treatment averages and the Block and Treatment effects from Display 13.12 (page 388).

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      f      fF      L      Lf      LfF
1.4131  0.9190  0.4112 -0.4794 -0.7718 -1.4921

Block:Treat
  Treat
Block C      f      fF      L      Lf      LfF
  B1 -0.2892  0.0951  0.1755 -0.0629  0.1972 -0.1157
  B2 -0.1797 -0.0509 -0.2013  0.1406 -0.1663  0.4576
  B3  0.2303 -0.1658 -0.0007  0.6996 -0.2540 -0.5094
  B4  1.0899  0.5743 -0.1179 -0.6724 -0.0947 -0.7791
  B5 -0.2650 -0.1850  0.3241  0.4996 -0.4376  0.0638
  B6 -0.0918 -0.4920 -0.2067 -0.1392  0.7185  0.2112
  B7 -0.6709  0.5274  0.3807 -0.5903 -0.2862  0.6394
  B8  0.1763 -0.3030 -0.3536  0.1250  0.3231  0.0322

```

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      f      fF      L      Lf      LfF
0.1805 -0.3137 -0.8214 -1.7120 -2.0044 -2.7247

Block:Treat
  Treat
Block C      f      fF      L      Lf      LfF
  B1 -1.512 -1.622 -2.049 -3.178 -3.210 -4.243
  B2 -0.942 -1.308 -1.966 -2.515 -3.114 -3.210
  B3  1.112  0.222 -0.121 -0.311 -1.557 -2.533

```

```

B4  2.848  1.838  0.638 -0.807 -0.522 -1.926
B5 -0.272 -0.686 -0.684 -1.399 -2.629 -2.848
B6  0.711 -0.184 -0.406 -1.229 -0.664 -1.891
B7 -0.785 -0.081 -0.735 -2.597 -2.585 -2.380
B8  0.284 -0.690 -1.248 -1.660 -1.754 -2.766

```

To answer specific questions of interest regarding subgroup comparisons we can use linear combinations. The *Sleuth* proposes five questions as detailed on pages 289-390. The code for results of these questions is displayed below and these results are also interpreted on pages 389-390 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)      Treatf      TreatfF      TreatL      TreatLf      TreatLfF
   -1.2226      -0.4941     -1.0019     -1.8925     -2.1849     -2.9052
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, -1/2, 1/2, 0, -1/2, 1/2))
> small = rbind('Small fish' = c(-1/2, 1/2, 0, -1/2, 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, -1/2, -1/2, -1, 1/2, 1/2))
> limpetsLarge = rbind('Limpets X Large' = c(0, 1, -1, 0, -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 upper CI
TreatLimpets X Small 0.09549      0.2593  0.3682  0.7136  -0.4203  0.6113

> fit.contrast(lm1, "Treat", limpetsLarge, conf.int=.95)

              Estimate Std. Error t value Pr(>|t|) lower CI upper CI
TreatLimpets X Large -0.2125      0.2994 -0.7097  0.4799  -0.8081  0.383

```

To attain the confidence intervals discussed in the “Summary of Statistical Findings” (page 376) 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 expected excellence affect performance? More specifically, does telling a manager that some of the supervisees are “superior” affect the supervisor’s perception of their performance (Pygmalion effect)? 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	Pygmalion:10	Min. :59.5
C2 : 3	Control :19	1st Qu.:69.2
C4 : 3		Median :73.9
C5 : 3		Mean :74.1
C6 : 3		3rd Qu.:78.9
C7 : 3		Max. :89.8
(Other):11		

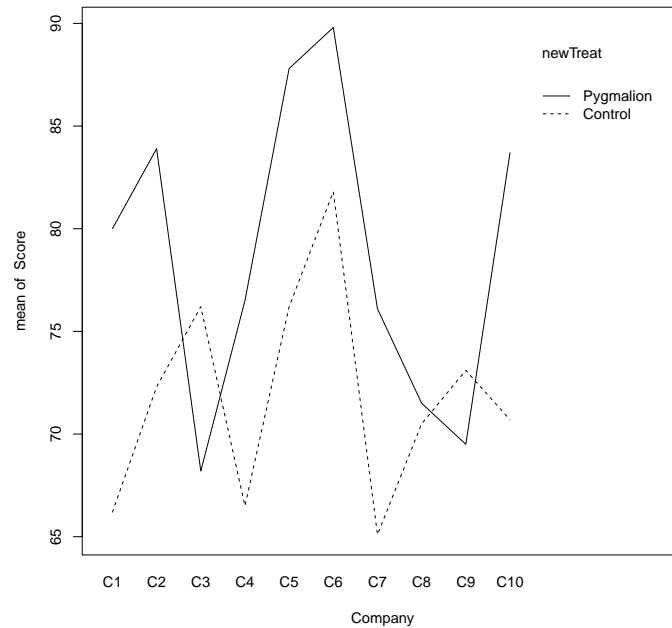
```
> case1302$newTreat = relevel(case1302$Treat, ref="Control")
```

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 378 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 392.

```
> 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 394).

```
> 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
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
CompanyC10	4.50	7.20	0.62	0.548
newTreatPygmalion	13.80	8.82	1.56	0.152
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
CompanyC10:newTreatPygmalion	-0.80	12.48	-0.06	0.950

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 395
```

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
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
CompanyC10	4.2333	5.3697	0.79	0.441
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)
```

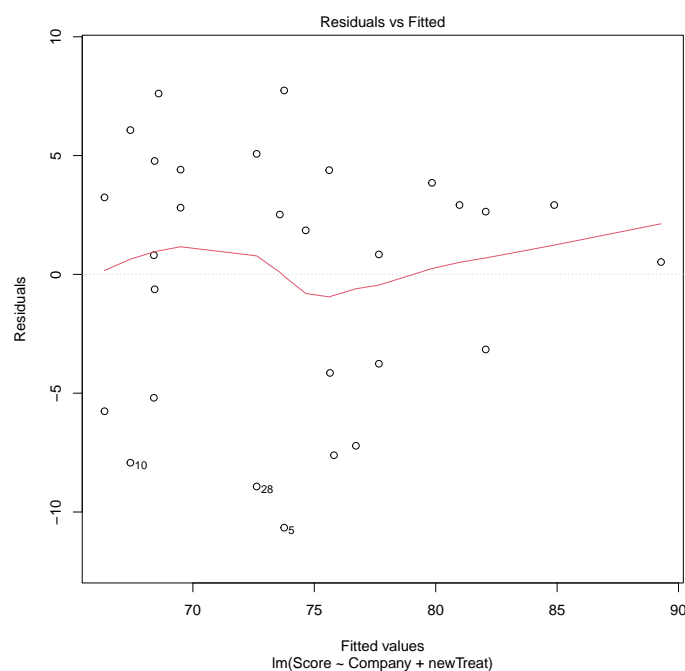
Analysis of Variance Table

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 395.

```
> plot(lm2, which=1)
```



3.4 Randomization Methods

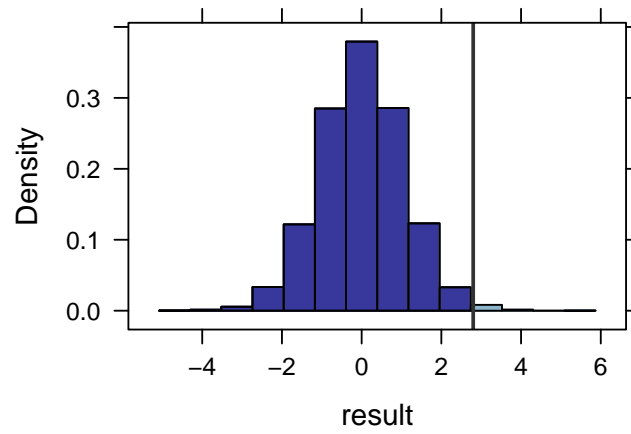
As introduced in Chapter 4, we can construct a randomization distribution by considering the distribution of a test statistic over all possible ways the randomization could have turned out. For the Pygmalion data we can construct a randomization distribution for the t -statistic of the treatment effect as discussed on pages 397-398.

```
> mod = lm(Score ~ Company+newTreat, data=case1302)
> obs = summary(mod)$coefficients["newTreatPygmalion", "t value"]
> obs

[1] 2.799

> nulldist = do(10000) * summary(lm(Score ~ shuffle(Company)+shuffle(newTreat),
+   data=case1302))$coefficients["shuffle(newTreat)Pygmalion", "t value"]
> histogram(~ result, groups=result >= obs, v=obs, data=nulldist)
> # akin to Display 13.20 page 398
> tally(~ result >= obs, format="proportion", data=nulldist)

result >= obs
  TRUE  FALSE
0.0066 0.9934
```



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 0.0066.