

11.9 Appendix: An Example of R Output for ANOVA

by EV Nordheim, MK Clayton & BS Yandell, September 20, 2004

Here we briefly indicate how R can be used to perform the ANOVA analysis for several examples in this chapter.

11.9.1 One-way ANOVA

We begin with the `oneway.test` command. To use this command it is necessary to have the data entered in two columns, one column for the treatment number and another for the data. We have already prepared file `drug.dat` in this fashion (available in the course data area).

```
> drug = read.table("drug.dat")
> group = factor(drug$V1)
> y = drug$V2
```

Column V1 contains the treatment group number and column V2 contains the drug response. The `factor` command explicitly states that `group` is a categorical factor with discrete levels, not a number.

We can use the `onewayaoov.test` command to generate the desired output. This command is slightly different from the earlier test commands, as it requires a “formula”, in this case `y~group`. The tilde (`~`) separates the response (`y`) from the treatment identifier (`group`).

```
> oneway.test(y ~ group, var.equal = TRUE)
```

One-way analysis of means

```
data: y and group
F = 7.3837, num df = 4, denom df = 33, p-value = 0.000231
```

This output is rather condensed, showing only what you “need to know” for the test. You can get a more standard looking ANOVA table, but without the TOTAL line by the following:

```
> drug.lm = lm(y ~ group)
> anova(drug.lm)
```

Analysis of Variance Table

```
Response: y
      Df Sum Sq Mean Sq F value    Pr(>F)
group   4  5.7028   1.4257   7.3837 0.000231 ***
Residuals 33  6.3719   0.1931
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This latter, more complicated method uses the linear model (`lm`) command. First, we explicitly need the treatment `group` to be categorical, a `factor` in R language (we created it this way above). Second, we use the formula in the `lm` command to calculate a linear model object. Finally, we print the ANOVA table using the `anova` command. This method has its advantages as it opens up many new avenues for data analysis.

11.9.2 Graphical Summaries for ANOVA

We can use R to produce stem and leaf displays or box plots and to plot the standard deviations versus the means to check assumptions. We can also use the `lm` model object created above to check the residuals.

Here are some shorthand ways to get stem and leaf displays and boxplots:

```
> tmp = tapply(y, group, stem)
```

```
The decimal point is at the |
```

```
6 | 0
6 | 5689
7 | 3
```

```
The decimal point is at the |
```

```
5 | 79
6 | 2234
6 | 7
7 | 0
```

```
The decimal point is at the |
```

```
4 | 9
5 | 34
5 | 5578
6 | 03
```

```
The decimal point is at the |
```

```
5 | 699
6 | 13
6 | 69
```

The decimal point is at the |

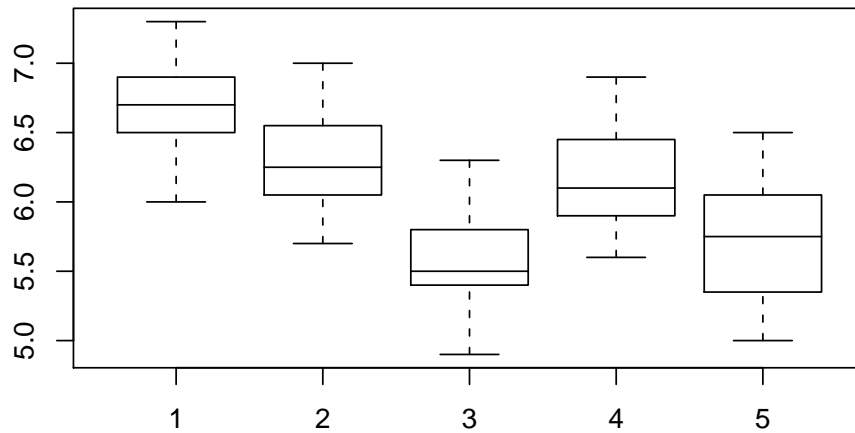
5 | 034

5 | 789

6 | 2

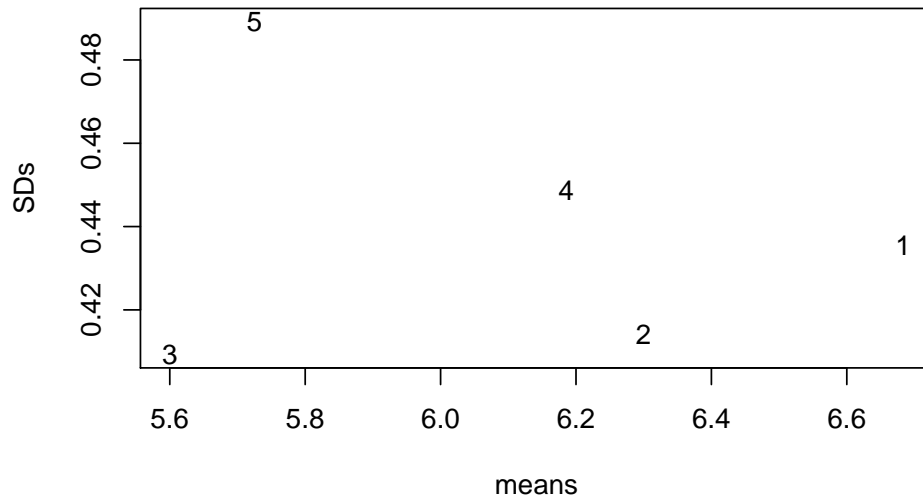
6 | 5

```
> boxplot(split(y, group))
```



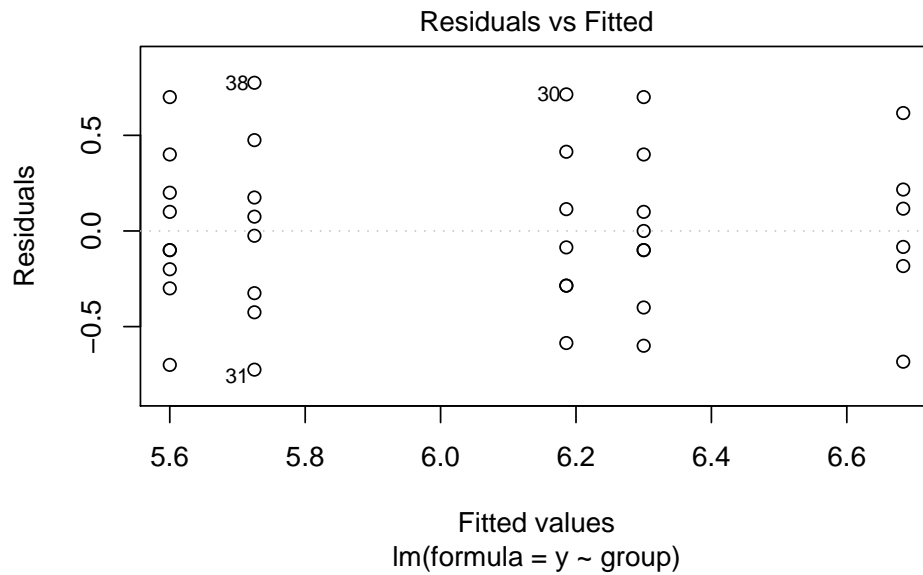
It can be revealing to plot the means against the SDs to see if there is any pattern of variability. This can suggest an appropriate transformation.

```
> plot(tapply(y, group, mean), tapply(y, group, sd), xlab = "means",  
+       ylab = "SDs", pch = levels(group))
```



In addition, we can use the `lm` fit to the one-way ANOVA to check for patterns in the residuals. There are actually four possible plots, but we show only the first one here.

```
> plot(drug.lm, which = 1)
```



11.9.3 Nonparametric one-way ANOVA

The nonparametric one-way ANOVA can be performed using the `kruskal.test` command. We illustrate the latter with the fungus data from Section 11.7.

```
> fungus = c(1.75, 1.25, 1, 2.75, 1.25, 2.5, 1.5, 3.75, 2, 1.75,
+ 2.5, 3, 2.75, 4.25, 3, 3.5, 2.75, 2.25, 4)
> trt = c(rep(1, 5), rep(2, 8), rep(3, 6))
> kruskal.test(fungus, trt)
```

```
      Kruskal-Wallis rank sum test
```

```
data:  fungus and trt
```

```
Kruskal-Wallis chi-squared = 8.479, df = 2, p-value = 0.01441
```

We can also use formula in this command. For instance, the nonparametric ANOVA for the drug problem is found by `kruskal.test(V2~V1,drug)`, yielding a p-value of 0.0015.

11.9.4 Levene's test of unequal variance for groups

We can compute Levene's test for multiple groups using the following home-made command. It uses some subtle R commands and is probably only for the adventurous reader. This is a more general version of the two-sample Levene's test shown in Appendix 10.

```
> levene.test = function(data, v1 = "V1", v2 = "V2") {
+   levene.trans = function(data) {
+     ## find median for group of data
+     ## subtract median; take absolute value
+     a = sort(abs(data - median(data)))
+     ## if odd sample size, remove exactly one zero
+     if (length(a)%%2)
+       a[a != 0 | duplicated(a)]
+     else a
+   }
+   ## set up data frame with transformed data for anova
+   V2 = lapply(split(drug[[v2]], drug[[v1]]), levene.trans)
+   V1 = rep(seq(length(V2)), lapply(V2, length))
+   levdata = data.frame(V1 = factor(V1), V2 = unsplit(V2, V1))
+   ## perform one-way anova on transformed data
+   cat("Overall ANOVA for Levene's Test\n")
+   print(anova(lm(V2 ~ V1, levdata)))
+   ## perform pairwise T-tests on transformed data
+   cat("\nPairwise Levene's Tests\n")
+   pairwise.t.test(levdata$V2, levdata$V1, p.adjust = "none")
+ }
```

We illustrate this ANOVA version of the Levene's test on the drug data:

```
> levene.test(drug)
```

```
Overall ANOVA for Levene's Test  
Analysis of Variance Table
```

```
Response: V2
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
V1	4	0.04681	0.01170	0.1632	0.9554
Residuals	31	2.22208	0.07168		

```
Pairwise Levene's Tests
```

```
$method
```

```
[1] "t tests with pooled SD"
```

```
$data.name
```

```
[1] "levdata$V2 and levdata$V1"
```

```
$p.value
```

	1	2	3	4
2	0.9089767	NA	NA	NA
3	0.8863667	0.7812375	NA	NA
4	0.5936598	0.4943337	0.6685501	NA
5	0.6893972	0.5793268	0.7812375	0.8638533

```
$p.adjust.method
```

```
[1] "none"
```

```
attr(,"class")
```

```
[1] "pairwise.htest"
```