Simple Linear models in R

Etienne Low-Décarie

2017-12-18

Based on sheet produced by Tom Cameron

0. Do it by hand...

Note this next can be cut and pasted in as a script, or alternatively, you can enter this all bit by bit because that way you're more likely to understand ...

Initial setup

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Loading required package: ggplot2
```

```
## Loading required package: tidyr
```

Read in the data

```
anova1<-read.csv("../Data/anova1.csv")
```

make sure the file type & path is correct make sure you have the data. Try to plot it

lets you see the format of the data, should location be a factor?

```
str(anova1)
```

```
## 'data.frame':    18 obs. of  2 variables:
##  $ location: int  1 1 1 1 1 1 1 1 1 2 ...
##  $ no_eggs : int  1 2 2 3 3 3 4 4 5 3 ...
```

make location a factor

```
anova1$location<-factor(anova1$location)
```

now set a column with the group means in

```
anova1 <- anova1 %>%
  group_by(location) %>%
  mutate(gpmeans=mean(no_eggs))
```

now set up a column with the grand mean in

```
anova1$grandmean<-mean(anova1$no_eggs)
```

now calculate the differences to calc the variances for within gps, between gps and total variance

```
anova1 <- within(anova1, {
  within<-anova1$no_eggs-gpmeans
  between<-anova1$gpmeans-grandmean
  total<-anova1$no_eggs-grandmean
})
```

within() allows you to work within an object, so this is equivalent to

```
anova1$within<-anova1$no_eggs-anova1$gpmeans
anova1$between<-anova1$gpmeans-anova1$grandmean
anova1$total<-anova1$no_eggs-anova1$grandmean
```

now square these distances to obliterate negative signs

```
anova1 <- within(anova1, {
  within<-within^2
  between<-between^2
  total<-total^2
})
```

now sum the squares to calc the sum of squares

```
withinSS<-sum(anova1$within)
betweenSS<-sum(anova1$between)
totalSS<-sum(anova1$total)
```

note you can do all these manipulations in single lines of code:

```
within<-sum((no_eggs-gpmeans)^2) between<-sum((gpmeans-grandmean)^2) total<-sum((no_eggs-grandmean)^2)
```

check you have estimates of the SS:

```
withinSS
```

```
## [1] 24
```

```
betweenSS
```

```
## [1] 18
```

```
totalSS
```

```
## [1] 42
```

now calculate the mean square as SS/df for total, you have 18 data points and have calculated the grandmean so that makes a df of 18-1=17 (or #putting it another way, knowing the grandmean and 17 data points you can calculate the 18th for bewteen group means, knowing one group mean and the grand mean you can calculate the 2nd

This is a very quick basic introduction to some simple linear models in R. Firstly there is an example where you do a one-way analysis "by hand" in R. Then a simple one way ANOVA and a regression for you to use as a means of getting to know R. The next few practicals will develop the skills you pick up here.

groupmean, so df=1 (=n groups-1) the error (within groups) df is estimated thus: for each group, if you #know the group mean and n-1 data points, you can calculate the final one. In this case we have 9 data in #each group, so 8df in each of two groups, so groups df=16. Check total df=within groups df+ between #groups df : 17=16+1

```
withinMS<-withinSS/16
betweenMS<-betweenSS/1
totalMS<-totalSS/17
```

```
withinMS
```

```
## [1] 1.5
```

```
betweenMS
```

```
## [1] 18
```

```
totalMS
```

```
## [1] 2.470588
```

check that the totalMS is the same as the variance for the dataset

```
var_egg <- var(anova1$no_eggs)
```

try this "if..then...else" statement

```
if(totalMS==var_egg) print ("yabadabadoo") else print ("D'oh!!!")
```

```
## [1] "yabadabadoo"
```

now calculate the F statistic as MS between/MS within

```
Fstat<-betweenMS/withinMS
```

now calculate P

```
pf(Fstat,1,16,lower.tail=F)
```

```
## [1] 0.003196953
```

you have now done an ANOVA "by hand" in R. Check it by doing it "properly":

```
m1<-lm(no_eggs~location, data=anova1)
```

create a linear model object using the "lm" command assign linear model to an object name, e.g. m1 or model 1 in assuming we can work with this data using a linear model – what assumptions?

```
summary(m1) #look at the summary function applied to a linear model
```

```
##
## Call:
## lm(formula = no_eggs ~ location, data = anova1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##     -2     -1      0      1      2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0000     0.4082   7.348 1.64e-06 ***
## location2     2.0000     0.5774   3.464   0.0032 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.225 on 16 degrees of freedom
## Multiple R-squared:  0.4286, Adjusted R-squared:  0.3929
## F-statistic:    12 on 1 and 16 DF,  p-value: 0.003197
```

```
anova(m1) # compare the summary function output to the anova function
```

```
## Analysis of Variance Table
##
## Response: no_eggs
##           Df Sum Sq Mean Sq F value   Pr(>F)
## location   1     18    18.0      12 0.003197 **
## Residuals 16     24     1.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exercise 1. One way ANOVA

An experiment was conducted to compare the effect of three amino acids: carnosine, histidine and imidazole on ascorbate-iron induced lipid peroxidation in rat liver microsomes. The results are below (in arbitrary units of rate). Do the different amino acids produce different amounts of peroxidation?

1. Input your 15 data into a single vector (`perox<-...`); then set up your factor column with 5x1, 5x2 and 5x3 (`amino<-gl(3,5)` or `amino<-rep(c("carn","hist","imid"),each=5))` Use ?gl or ?rep to explore what they mean.
2. Create a dataframe `perox_amino` with your two columns.
3. Do your lm with your model model1<-lm(perox~amino, data=perox_amino).
4. Check that the residuals are normally distributed and equal variance among groups (use `plot(model1))`
5. Do the different amino acids produce different amounts of peroxidation?
6. Use the `names()` command to look at the model object. Type each name in turn (e.g. model1$coefficients) and try and work out what they all mean.
7. Use `?lm` and explore the help files associated with lm .* What happens when you use the predict command (try `?predict`)? Syntax would be `predict(model1)`? .* How does this compare to `model1$fitted.values`? .* How do the fitted values relate to the model1$coefficients$?. $*$ $How do the model1$coefficients relate to what appears in the `summary(model1)` table?
8. The coefficients of a fitted model are used (with associated SEs) to compare groups. The comparisons of coefficients is known as a contrast. Type `options(contrasts=c("contr.sum","contr.poly"))`, run the model again, and look at the coefficients again (using summary(model1)). Can you work out (and use help!) what the coeffs mean? Putting the contrasts back to the default (`contrasts=c("contr.treatment","contr.poly")`) and try and work out (by looking at the coefficients) how the `contrast.sum` and `contrast.treatment` differ. How can you use the coefficients to compare groups?

```
perox <- c(13.9,
14.5,
13.4,
13.8,
14.1,
13.5,
```

```
12.9,
13.2,
12.7,
12.8,
14.1,
13.7,
13.5,
14.2,
13.6)

amino<-rep(c("carn","hist","imid"),each=5)

perox_amino <- data.frame(perox,amino)
```

2. Regression

The contractile vacuole of protozoa can often be observed under the microscope. In an experiment, protozoa were placed in different concentrations of water and the rate of contractile vacuole contractions observed. What, if any, is the relationship between contractile vacuole contractions and water salinity?

1. Always begin by plotting the data to see what a relationship looks like. Use lm to fit a model to it, with the model looking at contraction rate as a function of concentration. What is the equation describing the relationship? Does the vacuole work differently at different concentrations? If so, why?
2. Plot the graph and add the fitted line using the command abline(modelname) (where "modelname" is the name you gave your model)
3. set up a new data frame with `new <- data.frame(conc = seq(10, 30, 5))`

4. use this new data frame to predict values of contraction rate that are at different concentrations than used in the experiment by predict(modelname,new). Hint, you will get funny error messages if your new data has a different variable name to that in your model.