Processing and Analysis of Biological Data

The Linear Model II: Analysis of Variance

Øystein H. Opedal

1 Nov 2022

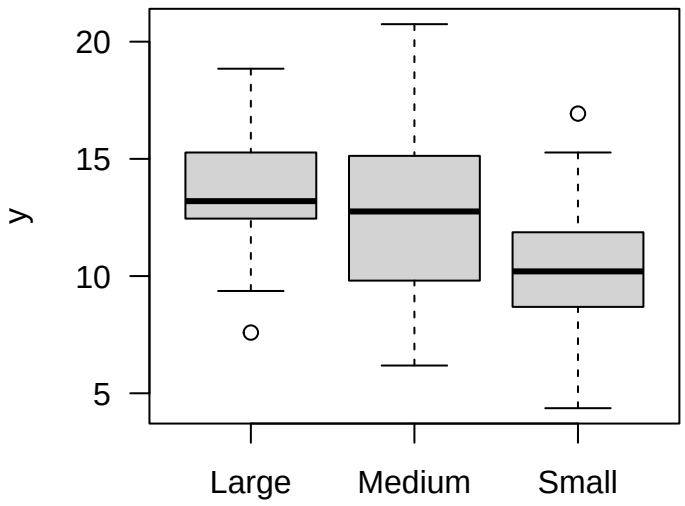
Analysis of variance (ANOVA)

When our predictor variables are categorical (factors), linear models are used to perform analyses of variance. The parameter estimation works in much the same way as in regression, except that instead of estimating regression slopes, we are estimating group effects.

Let's simulate some data, fit a linear model, and perform an ANOVA.

```
set.seed(100)
groups = as.factor(rep(c("Small", "Medium", "Large"), each=50))
x = c(rnorm(50, 10, 3), rnorm(50, 13, 3), rnorm(50, 14, 3))

plot(groups, x, las=1, xlab="")
```

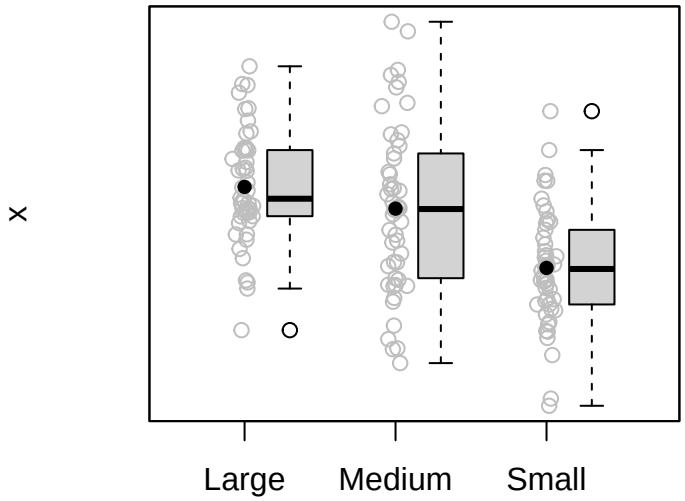


Plots like this are called 'boxplots', and can be useful for visualising the distribution of data across factor levels or other groups. With the default settings, the boxes range from the 1st to the 3rd quartile, i.e. they

span 50% of the data. The thick lines show the median, the 'whiskers' extend to 1.5 times the inter-quantile range, and individual circles show outliers. This representations allows us to assess whether the data are roughly normally distributed within each group, an assumption of the ANOVA model.

Boxplots are not directly useful for discussing ANOVA and variance partitioning though, and plots that show all the data can be more informative. Here is a rather elaborate plot combining a scatterplot (with values slightly 'jittered' along the x-axis for clarity) and a boxplot.

EXTRA EXERCISE: Reproduce a plot similar to this for the ANOVA exercise.



The aim of our ANOVA analysis is to evaluate whether the variance among groups is greater than the variance within groups, or more so than expected by chance. We fit the model just as before.

```
m = lm(x~groups)
anova(m)
```

```
## Analysis of Variance Table
##
## Response: x
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## groups      2  319.97 159.985  19.591 2.866e-08 ***
## Residuals 147 1200.43   8.166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA table contains a lot of information. First, we learn about the number of degrees of freedom for each variable. For the `groups` variable (our focal factor), the 2 degrees of freedom is the number of groups in our data (3) - 1. The minus 1 comes from the fact that we had to estimate the mean in the data to obtain our sums of squares (the sum of the square deviations of data points from their group means). Similarly for residual degrees of freedom, we have 150 - 2 - 1, where the 2 comes from estimating the two contrasts (difference of group 2 and 3 from group 1), and the 1 is still the estimated mean.

The `Sum Sq` are the sums of squares, i.e. the sum of the squared deviations of each observation from the grand mean. The total sum of squared (SS_T) divided by $n-1$ gives the total variance of the sample.

```
SS_T = 319.97+1200.43
SS_T/(150-1)
```

```
## [1] 10.20403
```

```
var(x)
```

```
## [1] 10.20403
```

We can easily get the proportion of variance explained by the `groups` variable, which is the same as the r^2 for the model.

```
319.97/SS_T
```

```
## [1] 0.2104512
```

The `Mean Sq` is the variance attributable to each variable, conventionally called the mean sum of squares (the sum of squares divided by the degrees of freedom). The F-ratio is computed as the mean sum of squares for the focal variable divided by the mean residual sum of squares. Thus, it represents the ratio of the among-group variance to the within-group variance, but also the sample size which gives the residual degrees of freedom and thus all else being equal, larger sample gives lower mean sum of squares and thus higher F-ratios and lower P-values.

In an ANOVA, a statistically significant result such as the one above indicates that at least one group mean is different from the others. To further assess which groups are different, we can extract the typical summary table of the linear model.

```
summary(m)
```

```
##
## Call:
## lm(formula = x ~ groups)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5887 -1.5596 -0.0987  1.6274  7.9729
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   13.7006     0.4041  33.901  < 2e-16 ***
## groupsMedium  -0.9277     0.5715  -1.623    0.107
## groupsSmall   -3.4561     0.5715  -6.047 1.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.858 on 147 degrees of freedom
## Multiple R-squared:  0.2105, Adjusted R-squared:  0.1997
## F-statistic: 19.59 on 2 and 147 DF,  p-value: 2.866e-08
```

This contains some of the same information as the ANOVA table, but we now also obtain parameter estimates. The first parameter, the intercept, corresponds to the estimated mean for the first level of the `groups` factor. In this example this happens to be 'Large', because L comes before M and S in the alphabet. The next two estimates represents *contrasts* from the reference group, and the associated hypothesis tests tests the null hypothesis that the group has the same mean as the reference group.

The summary table also gives us directly the r^2, which is a simple ANOVA is defined as $1 - SS_E/SS_T$, where SS_E is the sum of squares for the error term (residuals), and SS_T is the total sum of squared. (Control question: why could we do it even more simply above?).

The parameter estimates allow us to quantify the effect size, i.e. the magnitude of the difference between the groups. A useful way to report such differences is to compute the % difference (the contrast divided by the mean of the reference group, here $3.277/13.4642 = 0.243$), so that we can say that 'Small individuals were 24.3% smaller than large individuals'.

Note that if we want a different reference group, we can change the order of the factor levels.

```
groups = factor(groups, levels=c("Small", "Medium", "Large"))
m = lm(x~groups)
summary(m)
```

```
##
## Call:
## lm(formula = x ~ groups)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5887 -1.5596 -0.0987  1.6274  7.9729
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   10.2445     0.4041  25.349  < 2e-16 ***
## groupsMedium   2.5284     0.5715   4.424 1.88e-05 ***
## groupsLarge    3.4561     0.5715   6.047 1.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.858 on 147 degrees of freedom
## Multiple R-squared:  0.2105, Adjusted R-squared:  0.1997
## F-statistic: 19.59 on 2 and 147 DF,  p-value: 2.866e-08
```

Sometimes we also want to suppress the intercept of the model, and thus estimate the mean and standard error for each level of the predictor. We can do this by adding `-1` to the model formula (what comes after the `~` sign). This could be useful for example, if we wanted to obtain the estimated mean for each group, associated for example with a 95% confidence interval.

```
m = lm(x~groups-1)
summary(m)$coef
```

```
##              Estimate Std. Error  t value     Pr(>|t|)
## groupsSmall  10.24453  0.4041333 25.34937 1.586298e-55
## groupsMedium 12.77295  0.4041333 31.60578 1.980446e-67
## groupsLarge  13.70064  0.4041333 33.90129 2.276884e-71
```

```
confint(m)
```

```
##                   2.5 %   97.5 %
## groupsSmall   9.445865 11.04319
## groupsMedium 11.974287 13.57161
## groupsLarge  12.901979 14.49930
```

Tukey

The linear model III: two-way ANOVA

Analyses of variance can also be performed with more than one factor variable. If we have two factors, we can talk about two-way ANOVA, and so on. A typical example from biology is when we have performed a factorial experiment, and want to assess the effects of each experimental factor and their potential interaction.

With two factors, a full model can be formulated as `y ~ factor1 * factor2`. Recall that in Rsyntax, the * means both main effects and their interaction, while a : means only the interaction. A detectable interaction term in this model would indicate that the effect of factor 1 depends on the level of factor 2 (and *vice versa*). If we are analysing an experiment where we have manipulated both temperature and nitrogen supply, an interaction would mean that the effect of temperature depend on the nitrogen level.

Data exercise: analysing a factorial experiment

```
dat = read.csv("datasets/butterflies.csv")
names(dat)
```

```
## [1] "LarvalID"        "LarvalHost"      "Sex"             "MaternalHost"
## [5] "MotherID"        "DevelopmentTime" "AdultWeight"     "GrowthRate"
```

Data exercise: Interpreting linear-model analyses

Flowers are integrated phenotypes, which means that the different parts of the flowers are generally covarying with each other so that large flowers have e.g. both longer petals and longer sepals. Evolutionary botanists are interested in these patterns of covariation among floral parts, because they can affect for example the fit of flowers to their pollinators. We will work with a dataset on flower measurements from 9 natural populations in Costa Rica.

The traits are

- ASD: anther-stigma distance (mm)
- GAD: gland-anther distance (mm)
- GSD: gland-stigma distance (mm)
- LBL: lower bract length (mm)
- LBW: lower bract width (mm)
- UBL: upper bract length (mm)
- UBW: upper bract width (mm)
- GW: gland width (mm)
- GA: gland area (mm^2)

The traits have known or assumed functions. Anther-stigma distance is important for the ability of self-pollination, gland-anther distance and gland-stigmas distance affect the fit of flowers to pollinators, the upper and lower bracts are advertisements (think petals in other flowers), and the gland produces the the reward for pollinators.

The first step in any data analysis in always to explore the data. Make a series of histograms and plots. How are the data distributed? Are there any problematic outliers? How are patterns of trait correlations? Which traits are (proportionally) more variable?

What about differences between populations? Are any of the traits detectably different? By 'detectably' I mean statisticaly significant, but because the 's word' is so often misused, I find it safer to say detectably. At the very least, say 'statistically significant', to make clear that you do not (necessarily) imply any biological significance.

To get started, the following lines reads the data.

```
blossoms = read.csv("datasets/blossoms/blossoms.csv")
names(blossoms)
```

```
##  [1] "pop"   "patch" "ASD"   "GAD"   "GSD"   "LBL"   "LBW"   "UBL"   "UBW"
## [10] "GW"    "GA"
```

To summarize the data per population, the `apply` family of functions are useful. To call a function for each level of a factor, such as computing the mean for each population, we can use `tapply`.

```
tapply(blossoms$UBW, blossoms$pop, mean, na.rm=T)
```

```
##       S1      S11      S12       S2      S20      S27       S7       S8
## 17.37067 17.90706 16.82120 19.35714 20.94882 18.64091 18.68200 21.10600
##       S9
## 20.45882
```

A couple of packages are also very useful for producing complete summaries. I use `plyr` and `reshape2`. You could also consider learning some of the more modern things such as `tidyverse`.

```
library(plyr)
library(knitr)
popstats = ddply(blossoms, .(pop), summarize,
                 LBWm = mean(LBW, na.rm=T),
                 LBWsd = sd(LBW, na.rm=T),
                 GSDm = mean(GSD, na.rm=T),
                 GSDsd = sd(GSD, na.rm=T),
                 ASDm = mean(ASD, na.rm=T),
                 ASDsd = sd(ASD, na.rm=T))
popstats[,-1] = round(popstats[,-1], 2)
kable(popstats)
```

pop	LBWm	LBWsd	GSDm	GSDsd	ASDm	ASDsd
S1	18.32	2.13	4.75	0.73	2.56	1.20
S11	18.34	3.68	4.57	0.63	3.16	0.89
S12	17.35	1.34	5.02	0.90	2.66	0.84
S2	20.09	2.62	5.01	0.60	3.87	1.03

pop	LBWm	LBWsd	GSDm	GSDsd	ASDm	ASDsd
S20	21.78	2.58	4.91	0.52	6.32	1.71
S27	19.39	2.09	5.14	0.62	2.98	1.08
S7	19.24	3.76	5.08	0.65	3.92	1.06
S8	20.74	3.10	4.89	0.64	4.52	1.20
S9	20.78	3.68	4.57	0.74	4.05	0.90

After exploring and summarizing the data, fit some linear models to estimate the slopes of one trait on another. Interpret the results. Do the analysis on both arithmetic and log scale. Choose traits that belong to the same vs. different functional groups, can you detect any patterns? Produce tidy figures that illustrate the results. Hint: once you have produced a scatterplot, you can add more points (e.g. for a different variable) by using the `points()` function.