

Some basic statistics with R

Ramon Diaz-Uriarte

Dept. Biochemistry, Universidad Autónoma de Madrid
Instituto de Investigaciones Biomédicas “Alberto Sols” (UAM-CSIC)
Madrid, Spain¹
<https://ligarto.org/rdiaz>

2024-11-20 (Release 1.3: Rev: 6a827c4)

Contents

1	License and copyright	7
2	Introduction	7
2.1	The PDF and the code	7
2.2	Warning: eternally provisional	7
1	Comparing two groups	8
3	Introduction to comparing two groups	8
3.1	Files we will use	8
4	Types of data	8
5	Looking at the data: plots	9
5.1	Plots to do	9
5.2	What are the plots telling you?	12
5.3	Relations between variables	14
6	Comparing two groups with a t-test	14
6.1	Ideas that should be clear from the t-test part	15
6.1.1	What p-values and hypothesis testing are and are not	16
6.2	Confidence intervals	16
6.3	Assumptions of the t-test	17

¹r.diaz@uam.es, rdiaz02@gmail.com

Some basic statistics with R

7	One vs two-tailed tests.	18
8	Power of a test	18
8.1	An analogy: should I take the umbrella?	19
8.2	The winner's curse	19
9	(Bio)equivalence testing and turning things around	20
10	Bayesian inference	22
11	Confidence intervals and p-values: a longer discussion	22
12	Paired tests	23
12.1	Paired t-test	23
12.2	Reshaping the data for a paired t-test	25
12.2.1	The shape of the data	25
12.2.2	Reshaping our data	26
12.2.3	Reshaping with <code>reshapeL2W</code>	26
12.2.4	Reshaping with <code>unstack</code>	27
12.2.5	Reshaping with <code>reshape</code>	29
12.2.6	Reshaping with <code>spread</code>	29
12.2.7	<code>dcast</code> from <i>reshape2</i> ?	30
12.2.8	Reshaping et al: final comments	30
12.3	The paired t-test	30
12.4	The paired t-test again: a single-sample t-test	31
12.5	Plots for paired data	33
12.6	Choosing between paired and two-sample t-tests	37
12.7	A first taste of linear models	37
13	One-sample t-test	39
14	Non-parametric procedures	40
14.1	Why and what.	40
14.2	Wilcoxon rank-sum test or Mann-Whitney U test: 2 independent samples	40
14.3	What a rank-sum Wilcoxon test is not	41
14.4	Wilcoxon signed-rank test: matched-pairs or single sample test	45
14.5	A bad (very bad, terrible!) way to choose between nonparametric and parametric procedures	46
14.6	Wilcoxon's paired test and interval data	47

Some basic statistics with R

15	Non-independent data	49
15.1	Multiple measures per subject	49
15.2	Nested or hierarchical sources of variation.	53
15.3	Non independent data: extreme cases.	53
15.4	More non-independences and other types of data.	54
16	Symmetry and the paired t-test	55
II	Linear models: ANOVA, regression, ANCOVA	58
17	Introduction to ANOVAs, regression, and linear models	58
17.1	Files we will use	58
18	Comparing more than two groups	59
18.1	Recoding variables	59
18.2	A boxplot	59
18.3	An ANOVA; some basic theory and output.	61
18.4	Confidence intervals for the parameters of the model	62
18.5	So which means are different? Multiple comparisons	63
18.5.1	And can I plot the means with s.e from the model?	68
18.5.2	This is a mess. What figures do I use?	69
18.5.3	Side note: Interpreting confidence intervals	71
18.5.4	Multiple comparisons, other contrasts, etc.	71
18.6	t-test as ANOVA	72
18.7	Several ways of obtaining summaries	72
18.8	Can you do an ANOVA with only one sample per group?	72
18.9	One way ANOVA: summary of steps	73
19	Multiple comparisons: FWER and FDR	74
19.1	Family-wise error rate.	74
19.2	False discovery rate (FDR)	75
19.3	Multiple comparisons: struck by lightning, roulette, and lottery	76
20	Two-way ANOVA	77
20.1	A very simple two-way ANOVA	77
20.1.1	No interaction model	77
20.1.2	Interaction model.	79
20.1.3	Parameters and degrees of freedom	80
20.2	Loading the cholesterol data set	80


Some basic statistics with R

20.2.1	<code>Anova</code> , <code>anova</code> , <code>aov</code> , <code>lm</code> , <code>summary</code> : what gives?	82
20.3	Interactions	82
20.4	An ANOVA without interactions	87
20.5	Getting ready for how things change with the order of factor	87
20.6	The order of factors	88
20.6.1	Type I, Type II, Type III: a few technical notes	91
20.7	Does order always matter?	95
20.8	One observation per cell	100
20.9	Another quick two-way example	102
20.10	ANOVA/linear models with more than two factors	104
20.11	Multiple comparisons of means in two-way ANOVA	105
20.11.1	Multiple comparisons in multi-way ANOVA: main messages	105
20.11.2	Multiple comparisons in two-way ANOVA: example with no interactions	107
20.11.3	Multiple comparisons in two-way ANOVA: example with interactions	110
20.12	Nonparametric alternatives	119
21	Simple linear regression	120
21.1	And how does it look like: should have plotted the data!	121
21.1.1	Transforming the data	122
21.2	Confidence intervals and predictions intervals or prediction and confidence bands	123
21.3	Confidence intervals for the parameters	126
22	Multiple regression	128
22.1	Introduction to multiple regression and the <code>cystfibr</code> data	128
22.2	Multiple regression example	128
22.3	R^2 and Adjusted R^2	132
22.4	Interactions between continuous variables	134
22.5	Confidence intervals, confidence bands	136
22.5.1	Confidence intervals, confidence bands: the bootstrap	138
22.6	Three way anovas, factors with more than two classes, etc	138
23	Continuous and discrete independent variables and ANCOVA	139
23.1	The general scenario for ANCOVA	139
23.2	A first example of ANCOVA with the <code>cystfibr</code> data	140
23.3	A parallel slopes model	144

Some basic statistics with R

23.4	Formally comparing models	145
23.5	ANCOVA with the birds and the reptiles	146
23.6	More examples	153
23.7	More variables	153
23.8	Parameters, coefficients	153
24	Interactions, summary	153
25	Diagnostics	155
25.1	Model diagnostics: why, how	155
25.2	Diagnostics: an example with a designed experiment with factors.	155
25.3	Diagnostics: examples with some of the regression models.	160
25.4	Diagnostics: more examples with regression and ANCOVA models	163
25.5	Diagnostics: more examples of non-constant variance (and other issues)	167
25.6	Diagnostics: a couple of examples from ANOVA models that are largely OK	169
25.7	Diagnostics: more examples with designed experiments	171
25.8	Diagnostics: further issues	179
26	Variable and model selection	181
26.1	Why model selection?	181
26.2	Variable selection: the summary	182
26.3	Model selection using AIC and step	183
26.3.1	A very brief introduction to AIC	183
26.3.2	Model selection using AIC: an example	185
26.4	Differences between model selection using AIC and model comparison using anova (and hypothesis testing using Anova)	188
26.5	Model selection or model averaging?	189
26.6	A large pemax model as an example.	189
27	Side issue: a interesting case where some things went wrong without us noticing	198
28	Experimental design matters	202
29	Covariate adjustment and a few comments about causal inference	203

Some basic statistics with R

30	Dealing with ratios	204
30.1	A misleading case with parallel lines	204
30.2	A misleading case where ratios differ	206
30.3	Diagnostics et al. and other warning signs	209
31	Additional reading and what next	210
A	What if we did not recode training?	210
B	Anova tables from  et al.: understanding the coefficients and parameters	211
B.1	Changing the reference in the one-way	213
B.2	Coefficients with two-ways	215
B.3	Other contrasts	221
B.4	Changing the reference in two-ways	223
B.5	Unbalanced case	223
C	A two-way ANOVA with interaction: SS, coefficients	223
C.1	Some intuition for the interaction: estimate of coefficient vs. Sums of Squares	227
D	Session info and packages used	228

1 License and copyright

This work is Copyright, ©, 2014-2022, Ramon Diaz-Uriarte, and is licensed under a **Creative Commons** Attribution-ShareAlike 4.0 International License: <http://creativecommons.org/licenses/by-sa/4.0/>.



All the original files for the document are available (again, under a Creative Commons license) from <https://github.com/rdiaz02/R-basic-stats>. (Note that in the github repo you will not see the PDF, or R files, nor many of the data files, since those are derived from the Rnw file).

2 Introduction

This document is a brief reviews of statistical approaches for comparing two groups (t-tests et al.) and linear models (anovas, regressions, etc). It assumes you know a little bit of R. You will probably need to install several additional packages as you go along; you will see a something like `library(something)` or `require(something)`.

2.1 The PDF and the code

The primary output of this document is a PDF. All the original files for the document are available (again, under a Creative Commons license —see section 1) from <https://github.com/rdiaz02/R-basic-stats>. (Note that in the github repo you will not see the PDF, HTML, or R files, since those are derived from the Rnw file).

For many commands I do not show the output (e.g., because it would just provide boring and space-filling output). However, make sure you type and understand it. You can copy and paste, of course, but I strongly suggest you type the code and change it, etc.

2.2 Warning: eternally provisional

This file can get changed often. When asking questions in class or in the forum, refer to the section numbers AND section names (these change less than the page numbers).

Part I

3 Introduction to comparing two groups

Someone in your lab has measured the expression of several genes from a set of patients with and without cancer. You are in charge of looking at the data and answering the question “Does the expression of the genes differ between patients with and without cancer?”.

3.1 Files we will use

- This one
- `P53.txt`
- `MYC.txt`
- `BRCA2.txt`

4 Types of data

We need to get this out of the way, as we will refer to it frequently. Data can be measured in different scales. From “less information to more information” we can organize scales this way:

Nominal or categorical scale We use a scale that simply differentiates different classes. For instance we can classify some objects around here, “computer”, “blackboard”, “pencil”, and we can give numbers to them (1 to computer, 2 to blackboard, etc) but the numbers have no meaning per se.

Some basic statistics with R

Binary data are in a nominal scale with only two classes: dead or alive (and we can give a 0 or a 1 to either), male or female, etc.

Lots of biological data are in a nominal scale. For instance, suppose you look at the types of repetitive elements in the genome, and give a 1 to SINEs, a 2 to LINEs, etc. Or you number the aminoacids from 1 (alanine) to 20 (valine). You can of course count how many are of type 1 (how many are alanines), etc, but it would make no sense to do averages and say “your average AA composition is 13.5”.

Ordinal scale The data can be ordered in the sense that you can say that something is larger or smaller than something else. For instance, you can rank your preference for food as: “chocolate > jamon serrano > toasted crickets > liver”. You might assign the value 1 to chocolate (your most preferred food) and a 4 to liver (the least preferred) but differences or ratios between those numbers have no meaning.

Some measurements in biology are of these kind. Name a few?

Interval or ratio scale You can take differences and ratios, and they do have meaning². If a subject has a value of 6 for the expression of gene PTEN, another a value of 3, and another a value of 1, then the first has six times more RNA of PTEN than the last, and two times more than the second.

We will try to be careful. With nominal data we will always try to keep them as “things without numbers”, so that we make no mistakes (i.e., keep the aminoacids names, not just a bunch of 1 to 20). Ordinal scale are trickier, and we will need to think about the type of data and the analyses.

5 Looking at the data: plots

We first need to import the data. Make sure you name it sensibly; for instance, dp53:

```
dp53 <- read.table("P53.txt", header = TRUE, stringsAsFactors = TRUE)
```

Notice the `stringsAsFactors = TRUE`: we want the strings to be turned into factors, so we ask for it.

The first step ever is to look at the data. In fact, here we can look at all the original data. So go take a look at the data.

5.1 Plots to do

For all except the trivially tiniest datasets we want to use graphics. Make sure you do the following plots:

²Some authors make a distinction between ratio and interval scales; I won't. The only difference is that ratio scales have a natural zero.

Some basic statistics with R

- Histogram for each gene, using condition ("cond") as the conditioning or grouping variable ("Plot by:").
- Boxplot, using condition ("cond") as the conditioning or grouping variable ("Plot by:").
- Plot of means (and make sure you get nicer axes labels).
- Stripchart, and make sure you use "jitter", not "stack": **can you tell for which one of the variables this matters a lot?**
- Density plots ("Density estimates")

We will load a few packages we need:

```
library(car)

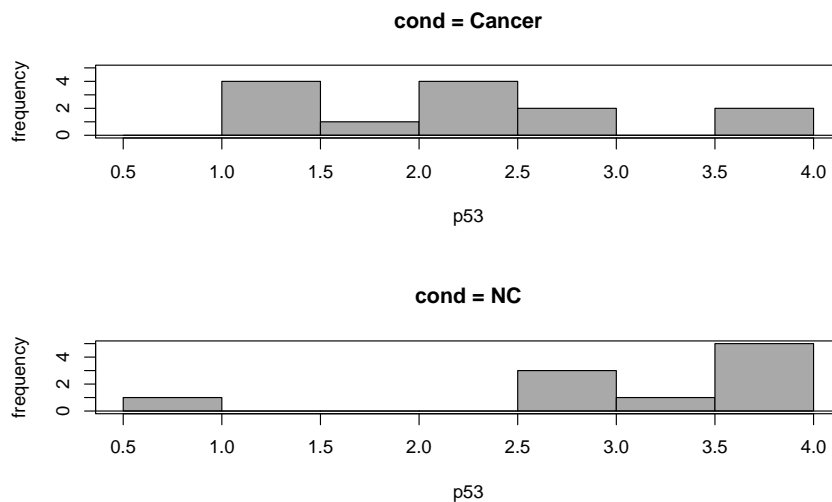
## Loading required package: carData

library(RcmdrMisc)

## Loading required package: sandwich
```

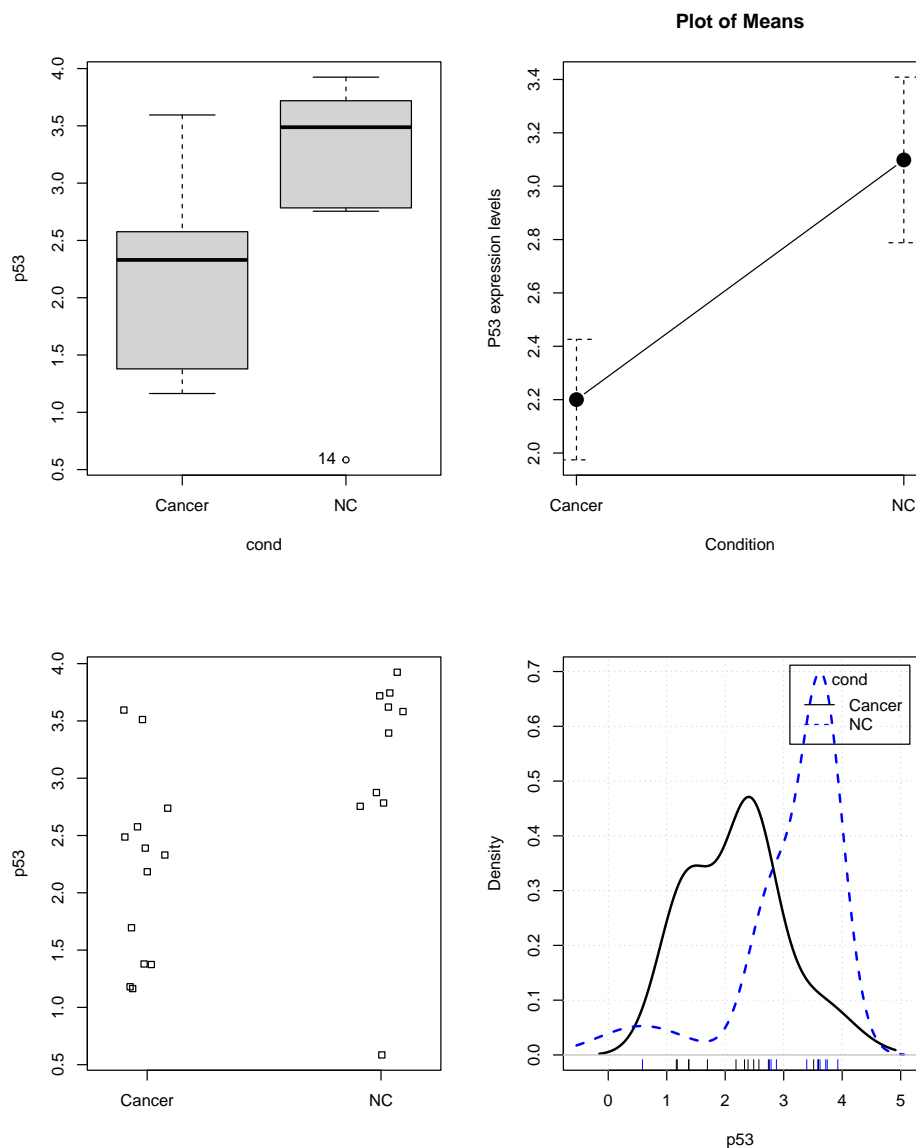
To get you going, I show all those for p53.

```
with(dp53, Hist(p53, groups = cond, col = "darkgray"))
```



```
op <- par(mfrow = c(2, 2)) ## to show 2 by 2 on the same figure
Boxplot(p53 ~ cond, data = dp53, id.method = "y")
plotMeans(dp53$p53, dp53$cond, error.bars = "se",
  xlab = "Condition", ylab = "P53 expression levels")
stripchart(p53 ~ cond, vertical = TRUE, method = "jitter",
  ylab = "p53", data = dp53)
densityPlot(p53 ~ cond, data = dp53, adjust = 1)
```

Some basic statistics with R



```
par(op)
```

Note that I used the `Boxplot`, but we could have used `boxplot`. The first is from `car` and provides added functionality. The `op <- par(mfrow = c(1,2))` and `par(op)` are a minor detail to restore options as they were.

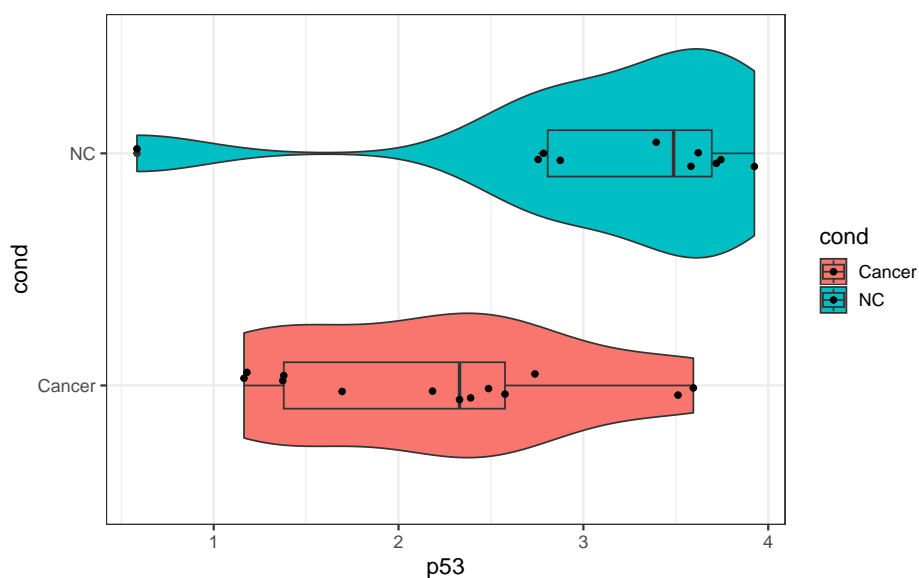
Likewise, I have used `Hist` and `plotMeans` from `RcmdrMisc`. They are not strictly needed (you can get these plots by other means), but they are extremely convenient.

And a violin + boxplot showing the points themselves, using package `ggplot2` (and we rotate it ... just because we can):

```
library(ggplot2)
tmpdf <- data.frame(x = dp53$cond, y = dp53$p53, z = dp53$cond)
```

Some basic statistics with R

```
theplot <- ggplot(data = tmpdf,  
                  aes(x = factor(x), y = y, fill = z)) +  
  geom_violin(position = position_dodge(width = 0.9)) +  
  geom_boxplot(width = 0.2) +  
  geom_jitter(colour = "black",  
              position =  
                position_jitterdodge(jitter.width = 0.25,  
                                      jitter.height = 0,  
                                      dodge.width = 0.9)) +  
  
  coord_flip() +  
  xlab("cond") +  
  ylab("p53") +  
  labs(fill = "cond") +  
  theme_bw(base_size = 14, base_family = "sans")  
print(theplot)  
rm(tmpdf, theplot)
```



5.2 What are the plots telling you?

Now, think about what these plots tell you:

1. The boxplots, is the “boxplot by group” more or less useful than the built-in one?
2. When was the stripchart specially useful? Could you see anything strange for brca2 without the stripchart?
3. Eye balling the plots, what variables do you think show differences between the two conditions? Wait! Think about at least:

Some basic statistics with R

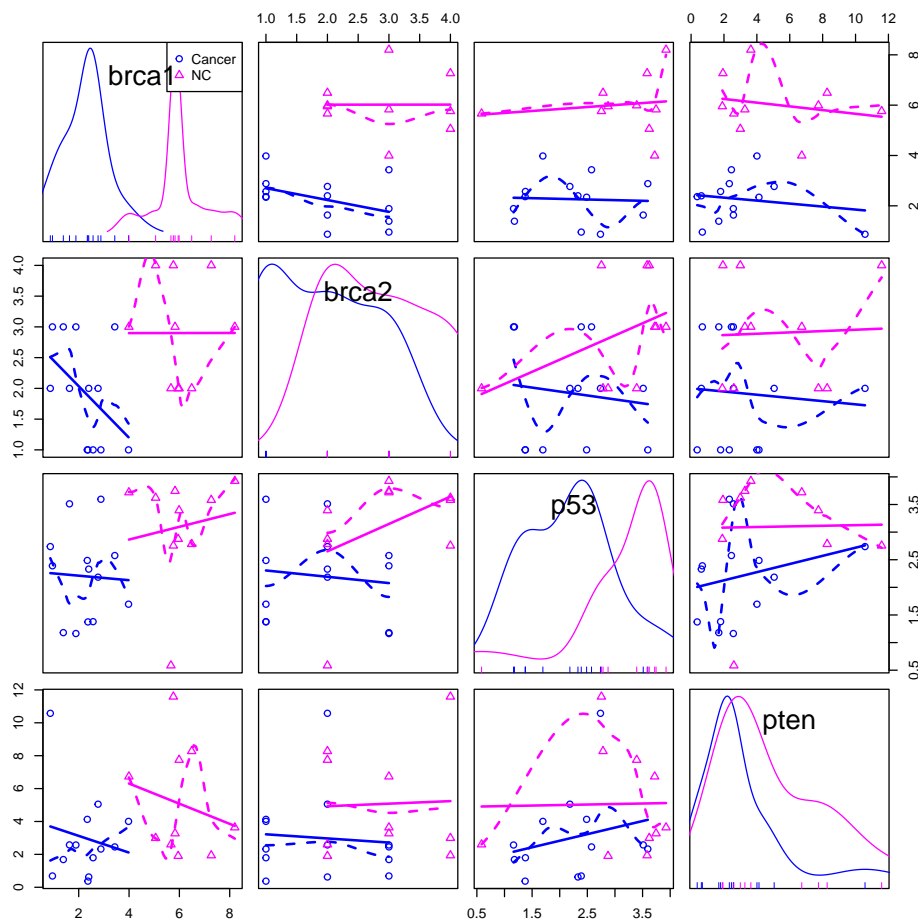
- (a) Differences in mean/median
- (b) Differences in dispersion (variance, IQR, etc)
- 4. Similar question again: what genes look like they have differential expression between the cancer and non-cancer patients?
- 5. Are density plots reasonable in these cases?

Some basic statistics with R

5.3 Relations between variables

We will focus on comparing two groups. But we have several variables (genes). An obvious thing to do is to look at how they are related AND display the different (two, in this case) groups.

```
scatterplotMatrix(~ brca1 + brca2 + p53 + pten | cond,  
  data = dp53)
```



We will not pursue this any further. But you know you can and probably want to look at these kinds of plots routinely.

6 Comparing two groups with a t-test

Let's start with `p53`.

```
t.test(p53 ~ cond, data = dp53)  
##  
## Welch Two Sample t-test
```

Some basic statistics with R

```
##
## data:  p53 by cond
## t = -2.3402, df = 17.402, p-value = 0.03142
## alternative hypothesis: true difference in means between group Cancer and group NC is not equal to 0
## 95 percent confidence interval:
##  -1.70635155 -0.08983306
## sample estimates:
## mean in group Cancer      mean in group NC
##           2.200308           3.098400
```

- What is this test for?
- Do you remember what the formula for the t-statistic look like? And why should this matter?
- What are the options given?
 - Equal variances? Does it make a difference? Any simple hint of whether you are using Welch's test or the equal-variances one?
 - Alternative hypothesis? One-tailed vs. two-tailed.
- Do results agree with the figures? What figures?
- Oh, in fact: can we interpret the results?
- The output gives a confidence interval: what is that?

(We will spend time in class making sure all this is understood if you have forgotten your stats classes).

Repeat the above with `brca1`.

6.1 Ideas that should be clear from the t-test part

A few ideas that should be clear after this section:

1. The difference between a sample and a population
2. That (most of the time) we use samples to make inferences about populations
3. What a statistic is; estimators (for example, the sample mean is an estimator) are a type of statistic ³
4. What a t-statistic is (a t-statistic is a test statistic used in hypothesis testing; a test statistic is a statistic used in hypothesis testing).
5. That statistics (and, thus, estimators) have distributions

³Briefly: a statistic is number that can be computed from a sample. An estimator is a function used to calculate, or estimate, a quantity of a probability distribution. The sample mean, with the function $\sum x/N$, is a function of the data and is used to estimate the true mean of a distribution using the data from a given sample.

Some basic statistics with R

6. The difference between standard deviation and standard error
7. That sampling introduces variability
8. That some procedures “ask more from the data” (e.g., interval data)
9. p-values
10. Null hypothesis
11. Distribution of the statistic under the null hypothesis
12. The logic behind a statistical test
13. The difference between estimation and hypothesis testing

If any one of the above is not clear, please ask it in class. Most of this, though, is material you have surely been taught before ☺ . So if any of this is unclear, please ask in class after having looked at the notes from your previous stats classes and, maybe, the Wikipedia.

6.1.1 What p-values and hypothesis testing are and are not

These ideas should be clear:

- We do not “confirm the null hypothesis”: we fail to reject (and failing to reject can be something trivial to achieve)
- The p-value is not the probability of the null hypothesis
- The p-value is not the probability of the alternative hypothesis
- The p-value can be used as a measure of strength of evidence **against** the null hypothesis. Remember the logic “either the null is false or something as (un)likely as the p-value happened”
- p-values are computed using models that make some assumptions
- Using p-values is often a much more sensible idea than saying “significant”
- Hair-splitting over, say, $p = 10^{-13}$ and $p = 10^{-16}$ makes no sense
- p-values are not the only tool we can use: do not forget confidence intervals!!

6.2 Confidence intervals

If this is not obvious to you, ask it in class: a figure that shows an estimate (e.g., a mean) and a 95% confidence interval, where the interval goes from, say, 1 to 2, **should not** be interpreted as saying that there is a 95% probability that the mean is between 1 and 2. That is not the correct interpretation of a confidence interval. Make sure you understand this!!!

In class we will very briefly cover how the confidence interval is constructed. In this case, it involves kind of turning around the logic of the p-value.

Some basic statistics with R

Please, read **now** and do the exercise in [\[See other file:\] CI-p-value-examples.pdf](#).

6.3 Assumptions of the t-test

One key assumption is **independence** of the data. This is the case for the t-test but is also the case for many statistical tests. We return to this below several times because it is a **crucial** assumption (e.g., section 15 and 25). Lack of independence is a serious and pervasive problem (and one form of non-independence is also referred to as “pseudoreplication”⁴).

When comparing two means, **equality of variances** is also important. But detecting differences in variances is complicated. Two practical solutions are: Welch’s test (the default in R) and using transformations. However, think about the meaning of a comparison of means when variances are hugely different: do you really want to do that?

What about normality? It matters, but not as much, especially as sample size gets larger. Deviations from normality because of skewness (asymmetry) can have a large effect. Deviations because of kurtosis (tails heavier than the normal) have very minor effects. That is why we often say “data are sufficiently close to normality”. And in the “sufficiently close” we worry mainly about asymmetries. And things tend to get “sufficiently close” as sample size grows larger (a consequence of the central limit theorem); how large is large enough? Oh, it depends on how far the distribution of the data is from the normal distribution, but often 10 is large enough, and 50 is generally well large enough (but in certain cases 100 might not be even close to large enough). (And if you want to play with the central limit theorem, there is a very nice demo available from R commander. It is the teaching demos, and once you load it, it will be available as “central limit theorem”; you can also use directly, without R commander, the *TeachingDemos* package, and call function `clt.examp`. There is also code in the repo to directly play with the central limit theorem; see file `central-limit-theorem-ex.R`).

Of course, although this should be obvious: when we talk about symmetry, normality, etc, we are talking about the data distribution of **each group separately**.

Finally, **outliers** can be a serious concern (by the way, outliers, or potential outliers — under some definition of outlier— get flagged by function `Boxplots` in R). In general, points very far from the rest of the points will have severe effects on the value computed for the mean (not the median —this is also related to why nonparametric procedures can be more robust). What to do, however, is not obvious. An outlier might be the consequence of an error in data recording. But it might also be a perfectly valid point and it might actually be the “interesting stuff”. Sometimes people carry out (and, of course, **should explicitly report**) analysis with and without the outlier; sometimes the same qualitative conclusions are reached, sometimes not.

⁴A major paper, a long while ago, in the journal *Ecological Monographs* by Hurlbert dealt with this and made “pseudoreplication” a well known term: Hurlbert, S.H. 2004. Pseudoreplication and the design of ecological field experiments. *Ecological Monographs*, 54, 187–211

Some basic statistics with R

Please, think carefully about what an outlier is before proceeding with your analysis but do not get into the habit of automatically getting rid of potential outliers. And whatever you do, you should report it.

The book by Rupert Miller “Beyond Anova” (Chapman and Hall, 1997) contains a great discussion of assumptions, consequences of deviations from them, what to do, etc.

7 One vs two-tailed tests

To be written. We will explain in class what the difference is. And why will discuss why, except for exceptional circumstances, using one-tailed tests with the t-test for comparing two groups is a bad idea.

8 Power of a test

If there is a true difference in means we would like to detect it. Power refers to our ability to reject the null when it is false. This figure might help; rows refer to the true state of the Universe and columns to your decision.

	Null hypothesis not rejected	Null hypothesis rejected
Means do not differ (H_0 is really true)	Correct	Type I error
Means differ (H_0 is really false)	Type II error	Correct

Power is $1 - \text{Type II error}$. Power is the probability of rejecting the null hypothesis when the null hypothesis is false.

How likely we are to detect a difference that really exists (power) depends on:

- The threshold we use for saying “the means differ” (α level, or Type I error)⁵.
- The sample size.
- The size of the effect (difference in means).
- The standard deviation.

Do you understand each one of these? We will not get into details. If you want, install [Rcmdr](#) and play with the “Teaching demos: Power of a test” (from the [RcmdrPlugin.TeachingDemos](#)). Or directly use [TeachingDemos](#) and run `run.power.examp(hscale = 1.5, vscale = 1.5, wait = FALSE)`.

There are simple, standard ways of computing power. But they require you to specify the above values, which are not always known (or easy to guess). The [IPSUR](#) and [RcmdrPlugin.IPSUR](#) packages provide some extra tools for power computations.

⁵There is, actually, a difference between p-value and α level; the p-value is a function of the data, is something you compute with a given procedure for a given data set; the α level or Type I error rate is a property of the procedure.

Some basic statistics with R

Power can be computed before hand to:

- Know if we are likely to find a difference if there is one (given our sample size and estimated effect sizes and standard deviations).
- Figure out if our sample size is OK given the desired power (and estimated effect sizes and standard deviations).

Please note: **it makes little sense** to compute the power of a test after the fact. It tells you nothing valuable⁶.

(A slightly more technical note: as is common in many stats intro texts, we have been mixing ideas that derive from Fisher's null hypothesis testing with ideas from Neyman and Pearson's alternative hypothesis and acceptance-rejection framework; we will not get further into this).

8.1 An analogy: should I take the umbrella?

If the trade-off between Type I and Type II errors is not clear, remember the analogy we used in class with respect to the umbrella (well, the behavior of taking or not the umbrella, as a function of meteorological predictions —with the **major** caveat that a p-value is **not** the probability of the null, whereas meteorological predictions are often the probability of, say, rain).

8.2 The winner's curse

When studies have low power, estimates of effects from tests that yield “significant” results are biased up (i.e., they are larger than they should be). In other words, for a given particular phenomenon, with low power studies, if you focus only on those papers with significant p-values, the effect estimates are too large (in absolute sense). So publication bias together with low power yields overestimates of effect sizes. If this is not clear, we will explain it in class.

Here is a interesting and recent example where the winner's curse can be playing a role: the possible use of botox to relieve depression <https://www.science.org/news/2021/06/can-botox-ease-depression-eliminating-frowns-researchers-have-doubts>.

The winner's curse is a **serious problem** (related to a bunch of others, such as reporting/publication bias and the reproducibility crisis.).

Oh, and the winner's curse is something that **can affect you too**; in other words, the winner's curse is not something that only happens to other researchers 😊.

⁶The paper by Hoenig and Heisey, “The abuse of power: the pervasive fallacy of power calculations for data analysis”, *The American Statistician*, 2001, 55: 19–24, explains this in more detail.

9 (Bio)equivalence testing and turning things around

We have set things up so that we **need strong enough evidence to reject the null** and we **use p-values of measures of strength of evidence AGAINST the null**. This is often what we want in science (why? think about how many papers with claims that are trivially generated under the null you want to read). But not always. And in many cases, in particular in issues related to public health, we might want to follow a precautionary principle (https://en.wikipedia.org/wiki/Precautionary_principle).

For example, we might want to say “We will only allow you to dump chlorine in the river if there is strong enough evidence that such action will not cause harm, for instance, will not increase fish mortality”. This is not something you can settle with p-values as we have used them. **Why??**

What can we do? We want to turn the process around. You would want a procedure for answering the following question: “Is there strong enough evidence that, if chlorine has an effect, the effect is no larger than an increase in fish mortality of 1%?”. This is like inverting the burden of proof: it is as if now we want evidence in favor of a hypothesis that says that things do not differ by more than a given, small, value (i.e., it seems we now want evidence in favor of what is often the null). In other words, we want strong evidence that the true value is inside the equivalence bounds, the bounds that say that “things are similar, or equivalent” (we have simplified things here, by just worrying about increases in fish mortality, but often we worry about deviations both up and down).

We can approach this problem as one of finding evidence against the (new null) hypothesis that things differ by more of the specified tolerance, in our case that 1% increase in fish mortality; in other words, that the true value falls outside the equivalence bounds. If we can reject our new null that the groups differ by more than a given threshold (that the true difference falls outside the equivalence bounds), we have established that they are equivalent. In some cases it is relatively straightforward to do this (such as with the TOST procedure), in many others it is not.

We will not pursue this any further. But, please, think of other examples of similar cases. Examples involving, for example, human disease and certain environmental changes, or secondary effects of drugs, or that two drugs (say, brand and generic) are for all practical purposes identical.

Oh, and, of course, this section emphasizes another idea: a large p-value is not evidence in favor of the null!!!

Figure 1, from Lakens et al. 2018⁷, shows equivalence testing alongside the usual null hypothesis significance testing, as well as other tests. Which of the four cases represented corresponds to the chlorine example in the notes? What case would represent examining if the effects of the brand name drug and a generic are similar?

⁷Lakens et al.(2018). Equivalence Testing for Psychological Research: A Tutorial. *Advances in Methods and Practices in Psychological Science*, 1(2): 259–269. <https://doi.org/10.1177/2515245918770963>

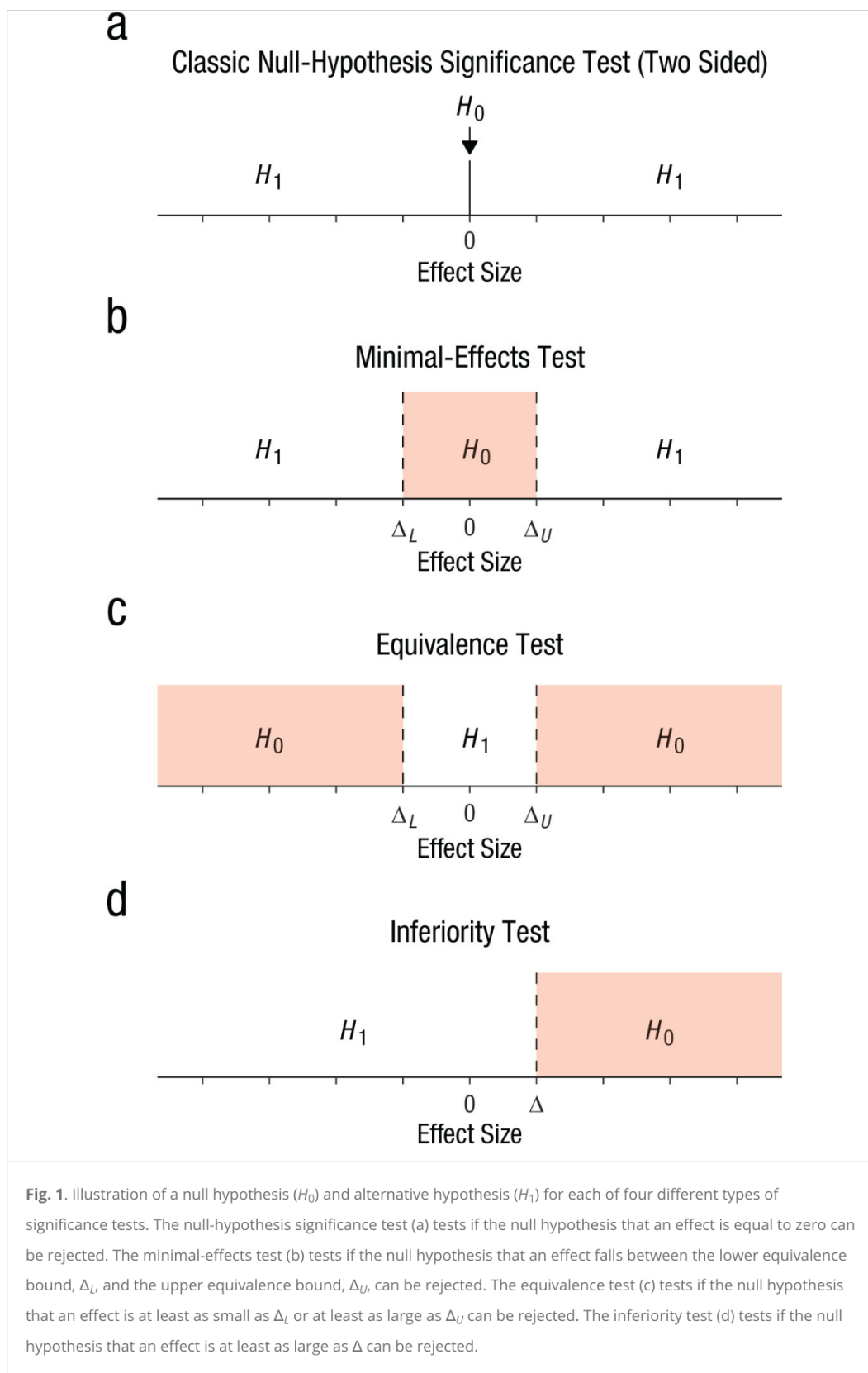


Figure 1 – Equivalence testing, from Lakens et al 2018

Figure 1 in Lakens et al.(2018). Equivalence Testing for Psychological Research: A Tutorial. *Advances in Methods and Practices in Psychological Science*, 1(2): 259–269. <https://doi.org/10.1177/2515245918770963>.

10 Bayesian inference

We will quickly review the following issues (please, make sure to look at your intro probability/stats notes):

- What is Bayes rule (or Bayes theorem)
- Using Bayes rule to obtain the posterior probability of a hypothesis given data. Why this is a very attractive idea.
- The problems of trying to do that: where is the prior for the hypothesis coming from?
- Why we will not spend more time on this.
- That Bayes rule is **NOT** controversial at all: it is **the procedure to use** in many cases, for example diagnostic testing (probability of having disease X, given that you got a positive test result). But using Bayes rule can be controversial or complicated for the cases we discuss here.
- That most of the p-values you see in the literature are, in fact, not “Bayesian p-values” but rather use frequentist statistics, so you must know how to use and interpret them.
- (And that we do not have conceptual or philosophical problems with at least many Bayesian approaches, and use them when we think appropriate; it is just that for this class we do not deem it appropriate to go in that direction).

11 Confidence intervals and p-values: a longer discussion

In class we will go over [\[See other file:\]](#) the file [confidence-intervals-p-values-interpretation.pdf](#).

You probably want to review the notes you took for [\[See other file:\]](#) [CI-p-value-examples.pdf](#).

12 Paired tests

Load now the data set called `MYC.txt`. As before, give it a reasonable name.

```
dmyc <- read.table("MYC.txt", header = TRUE)
```

Look at the data. Anything interesting?

There are actually two observations per subject, one from tumor tissue and one from non-tumor tissue. This is a classical setting that demands a paired t-test. **Why?** When answering this question think about the idea of using each subject as its own control.

How can we check the above?

```
all(with(dmyc, table(id, cond)) == 1)

## [1] TRUE
```

12.1 Paired t-test

Let's do the paired t-test.

```
myc.cancer <- dmyc$myc[dmyc$cond == "Cancer"]
myc.nc <- dmyc$myc[dmyc$cond == "NC"]
t.test(myc.nc, myc.cancer, paired = TRUE)

##
## Paired t-test
##
## data: myc.nc and myc.cancer
## t = 4.079, df = 11, p-value = 0.001823
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
## 0.432056 1.444777
## sample estimates:
## mean difference
## 0.9384167
```

Of course, this **crucially assumes** that the data are ordered the same way by patient: the first `myc.cancer` is the same patient as the same `myc.nc`, etc. This is the case in our data, but need not be. We can check it:

```
dmyc

##      myc   cond      id
## 1 38.289 Cancer bqysitlvpm
## 2 76.188 Cancer zuhxmiyfos
## 3 24.621 Cancer bpkmxwhtsg
```

Some basic statistics with R

```
## 4 54.079 Cancer qsmxexkcnw
## 5 43.832 Cancer uhbkijsnvw
## 6 0.300 Cancer efzpcboidt
## 7 31.055 Cancer trsyacmejh
## 8 58.402 Cancer hyqjownkue
## 9 29.723 Cancer ejmkobsqrh
## 10 6.190 Cancer mculjayvhw
## 11 52.626 Cancer ytwgsplaef
## 12 22.089 Cancer dchlnopykg
## 13 39.634      NC bqysitlvpm
## 14 78.361      NC zuhxmiyfos
## 15 24.396      NC bpkmxwhtsg
## 16 53.902      NC qsmxexkcnw
## 17 44.679      NC uhbkijsnvw
## 18 1.675      NC efzpcboidt
## 19 32.260      NC trsyacmejh
## 20 59.427      NC hyqjownkue
## 21 30.300      NC ejmkobsqrh
## 22 6.030      NC mculjayvhw
## 23 54.494      NC ytwgsplaef
## 24 23.497      NC dchlnopykg
```

However, to ensure that order is OK we could have pre-ordered the data by patient ID and by condition within patient (this second step isn't really needed in this case, but is in general):

```
dmyc0 <- dmyc[order(dmyc$id, dmyc$cond), ]
dmyc0

##      myc   cond      id
## 3 24.621 Cancer bpkmxwhtsg
## 15 24.396      NC bpkmxwhtsg
## 1 38.289 Cancer bqysitlvpm
## 13 39.634      NC bqysitlvpm
## 12 22.089 Cancer dchlnopykg
## 24 23.497      NC dchlnopykg
## 6 0.300 Cancer efzpcboidt
## 18 1.675      NC efzpcboidt
## 9 29.723 Cancer ejmkobsqrh
## 21 30.300      NC ejmkobsqrh
## 8 58.402 Cancer hyqjownkue
## 20 59.427      NC hyqjownkue
## 10 6.190 Cancer mculjayvhw
## 22 6.030      NC mculjayvhw
## 4 54.079 Cancer qsmxexkcnw
```


Some basic statistics with R

```
## 16 53.902      NC qsmeyxkcw
## 7  31.055 Cancer trsyacmejh
## 19 32.260      NC trsyacmejh
## 5  43.832 Cancer uhbkifsnvw
## 17 44.679      NC uhbkifsnvw
## 11 52.626 Cancer ytwgsplaef
## 23 54.494      NC ytwgsplaef
## 2  76.188 Cancer zuhxmiyfos
## 14 78.361      NC zuhxmiyfos

myc.cancer <- dmyc0$myc[dmyc0$cond == "Cancer"]
myc.nc <- dmyc0$myc[dmyc0$cond == "NC"]
t.test(myc.nc, myc.cancer, paired = TRUE)

##
## Paired t-test
##
## data: myc.nc and myc.cancer
## t = 4.079, df = 11, p-value = 0.001823
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  0.432056 1.444777
## sample estimates:
## mean difference
##      0.9384167
```

12.2 Reshaping the data for a paired t-test

The purpose of this section is to show you several different ways of reshaping data, but we will skip the details in class, unless you run into problems. In class we will definitely cover section [12.2.1](#), [12.2.2](#) and read section [12.2.8](#). The rest is left here so that you can refer to it when you need to do it with your own data, but, again, we will **not** cover it in class: you need to understand the source of the problem but the syntax for the solutions is not important (you have it here, and can refer to it when you need it).

12.2.1 The shape of the data

Often, when you know you are only going to do a paired test you can organize your data in a table structure like the one in [Table 1](#):

That looks ok, but is really a very limiting format for the data. Think about what you would do if you had additional “tumor” and “non-tumor” data per subject, or you had additional covariates for each measure, etc. That is why we often want to have the data in a “stacked” format as in [Table 2](#):

Some basic statistics with R

SubjectID	Tumor	Non-Tumor
pepe	23	45
maria	29	56
...

Table 1 – Paired data in a “unstacked” shape/format

SubjectID	Myc	Condition
pepe	23	tumor
pepe	45	nontumor
maria	29	tumor
maria	56	nontumor
...

Table 2 – Paired data in a “stacked” shape/format

This is arguably a much more useful way of storing the data for most general analyses and we can keep adding additional information if we need to. Whole papers have been written about these issues⁸, but we will stop here.

12.2.2 Reshaping our data

This is optional reading, and not a key part. This is about the mechanics of how to do the reshaping. So what follow are some general explanations. There is also a different summary available in the R-bioinfo-intro notes: <https://github.com/rdiaz02/R-bioinfo-intro>, in this file: <https://github.com/rdiaz02/R-bioinfo-intro/blob/master/reshape-long-wide.R>. **I will, at least initially, skip this section about the mechanics in class.**

So that is what we want: the values of myc of the same subject to be on the same row. We will do this reshaping in several different ways.

There are many alternative approaches for reshaping the data directly from the command line. Many, many, many⁹. And yes, it is unavoidable that this will get a little bit complicated. But manipulating and reshaping data must be done with a lot of care. You do not need to remember the details here, you just need to be aware of the issues.

12.2.3 Reshaping with `reshapeL2W`

This uses a function from the package `RcmdrMisc`. It is a simple front-end to function `reshape` but, by virtue of being simple, it might be worth taking a look if you don't do this thing often.

⁸For example, Wickham's “Tidy data”, in the *Journal of Statistical Software*: <http://www.jstatsoft.org/v59/i10>

⁹For instance: http://www.cookbook-r.com/Manipulating_data/Converting_data_between_wide_and_long_format/ or https://rpubs.com/bradleyboehmke/data_wrangling; these are heavily biased towards the “Hadley's way”.

Some basic statistics with R

```
(dmycWide <- reshapeL2W(dmyc, within="cond", id="id", varying="myc"))  
  
##           myc.Cancer myc.NC  
## bqysitlvp      38.289 39.634  
## zuhxmifos      76.188 78.361  
## bpkmxwhtsg     24.621 24.396  
## qsmxexkcw      54.079 53.902  
## uhbkijsnv      43.832 44.679  
## efzpcboidt      0.300  1.675  
## trsyacmejh     31.055 32.260  
## hyqjownkue     58.402 59.427  
## ejmkobsqrh     29.723 30.300  
## mculjayvhw      6.190  6.030  
## ytwgsplaef     52.626 54.494  
## dchlnopykg     22.089 23.497
```

12.2.4 Reshaping with `unstack`

We will first use the R function `unstack`. This is the natural reverse operation of “stacking”.

```
unstack(x = dmyc, form = myc ~ cond)  
  
##      Cancer      NC  
## 1  38.289 39.634  
## 2  76.188 78.361  
## 3  24.621 24.396  
## 4  54.079 53.902  
## 5  43.832 44.679  
## 6   0.300  1.675  
## 7  31.055 32.260  
## 8  58.402 59.427  
## 9  29.723 30.300  
## 10 6.190  6.030  
## 11 52.626 54.494  
## 12 22.089 23.497
```

That seems to do it but ... how can you be sure each row corresponds to the same subject? We are counting on it to work, but it might not if the data had not been ordered by subject id. (See also section 12.1). So we will be explicit here, ordering by condition and then id:

```
dmyc0 <- dmyc[order(dmyc$cond, dmyc$id), ]  
dmyc0  
  
##      myc      cond      id  
## 3  24.621 Cancer bpkmxwhtsg
```

Some basic statistics with R

```
## 1 38.289 Cancer bqysitlvpm
## 12 22.089 Cancer dchlnopykg
## 6 0.300 Cancer efzpcboidt
## 9 29.723 Cancer ejmkobsqrh
## 8 58.402 Cancer hyqjownkue
## 10 6.190 Cancer mculjayvhw
## 4 54.079 Cancer qsmxexkcw
## 7 31.055 Cancer trsyacmejh
## 5 43.832 Cancer uhbkifsnvw
## 11 52.626 Cancer ytwgsplaef
## 2 76.188 Cancer zuhxmiyfos
## 15 24.396 NC bpkmxwhtsg
## 13 39.634 NC bqysitlvpm
## 24 23.497 NC dchlnopykg
## 18 1.675 NC efzpcboidt
## 21 30.300 NC ejmkobsqrh
## 20 59.427 NC hyqjownkue
## 22 6.030 NC mculjayvhw
## 16 53.902 NC qsmxexkcw
## 19 32.260 NC trsyacmejh
## 17 44.679 NC uhbkifsnvw
## 23 54.494 NC ytwgsplaef
## 14 78.361 NC zuhxmiyfos
```

Now, we unstack:

```
merged2 <- unstack(dmyc0, form = myc ~ cond)
merged2

##      Cancer      NC
## 1 24.621 24.396
## 2 38.289 39.634
## 3 22.089 23.497
## 4 0.300 1.675
## 5 29.723 30.300
## 6 58.402 59.427
## 7 6.190 6.030
## 8 54.079 53.902
## 9 31.055 32.260
## 10 43.832 44.679
## 11 52.626 54.494
## 12 76.188 78.361
```

We could directly use the data, but we are missing the IDs. We will add them:

Some basic statistics with R

```
(merged2$id <- dmyc0$id[1:12])  
## [1] "bpkmxwhtsg" "bqysitlvp" "dchlnopykg" "efzpcboidt" "ejmkobsqrh"  
## [6] "hyqjownkue" "mculjayvhw" "qsmeyexkcw" "trsyacmejh" "uhbkifsnvw"  
## [11] "ytwgsplaef" "zuhxmiyfos"
```

Verify we are now ok comparing merged2 with dmyc.

12.2.5 Reshaping with `reshape`

This is another built-in function in R, `reshape`. This can be a complicated function to use, but it is extremely powerful. I'll use it here fully specifying all the arguments¹⁰:

```
(merged3 <- reshape(dmyc, direction = "wide", idvar = "id",  
                    timevar = "cond", v.names = "myc"))  
  
##           id myc.Cancer myc.NC  
## 1 bqysitlvp      38.289 39.634  
## 2 zuhxmiyfos      76.188 78.361  
## 3 bpkmxwhtsg      24.621 24.396  
## 4 qsmeyexkcw      54.079 53.902  
## 5 uhbkiifsnvw      43.832 44.679  
## 6 efzpcboidt        0.300  1.675  
## 7 trsyacmejh      31.055 32.260  
## 8 hyqjownkue      58.402 59.427  
## 9 ejmkobsqrh      29.723 30.300  
## 10 mculjayvhw        6.190  6.030  
## 11 ytwgsplaef      52.626 54.494  
## 12 dchlnopykg      22.089 23.497
```

12.2.6 Reshaping with `spread`

OK, so we will do it in yet another way. You need to install the `tidyr` package. If you don't have it, install it if you want. This is not a big deal: it is just to show you a fourth way.

```
library(tidyr)  
(merged4 <- spread(dmyc, cond, myc))  
  
##           id Cancer      NC  
## 1 bpkmxwhtsg 24.621 24.396  
## 2 bqysitlvp 38.289 39.634  
## 3 dchlnopykg 22.089 23.497  
## 4 efzpcboidt  0.300  1.675  
## 5 ejmkobsqrh 29.723 30.300
```

¹⁰In this case, you do not need to specify the `v.names` argument, for example, but I'd rather be explicit.

Some basic statistics with R

```
## 6  hyqjownkue 58.402 59.427
## 7  mculjayvhw  6.190  6.030
## 8  qsmeyekcnw 54.079 53.902
## 9  trsyacmejh 31.055 32.260
## 10 uhbkijsnvw 43.832 44.679
## 11 ytwgsplaef 52.626 54.494
## 12 zuhxmifos 76.188 78.361
```

You can compare with `merged3` and `merged2` and `merged`, and verify they have the same data.

12.2.7 `dcast` from `reshape2`?

As it says, you could try doing the reshaping using `dcast` from `reshape2`. Try it if you want to.

12.2.8 Reshaping et al: final comments

A few comments:

- This data reshaping is not strictly necessary from doing a paired t-test in R.
- We will see below a way to carry out these same analyses with a linear model (section 12.7). And no reshaping needed.
- Data manipulation can be tricky: you **must** be sure to get it right. What would I do? Do it in two different ways, and compare. Among those two I'd probably have `reshape` because, even if complicated, it allows you to be completely explicit about what is what.

12.3 The paired t-test

Just do a paired t-test. What is the result? Compare it with doing a t-test as if they were two independent samples. The differences are rather dramatic!

I'll use `merged3`; you can use `merged2` or `merged4` if you want.

```
## Paired
t.test(merged3$myc.NC, merged3$myc.Cancer, alternative='two.sided',
       conf.level=.95, paired=TRUE)

##
## Paired t-test
##
## data: merged3$myc.NC and merged3$myc.Cancer
## t = 4.079, df = 11, p-value = 0.001823
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
```

Some basic statistics with R

```
## 0.432056 1.444777
## sample estimates:
## mean difference
## 0.9384167
```

```
## Two-sample
t.test(merged3$myc.NC, merged3$myc.Cancer, alternative='two.sided',
       conf.level=.95, paired=FALSE)

##
## Welch Two Sample t-test
##
## data: merged3$myc.NC and merged3$myc.Cancer
## t = 0.10365, df = 21.996, p-value = 0.9184
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -17.83752 19.71435
## sample estimates:
## mean of x mean of y
## 37.38792 36.44950
```

The last is of course the same as

```
t.test(myc ~ cond, alternative = 'two.sided', conf.level=.95,
       var.equal=FALSE, data=dmyc)

##
## Welch Two Sample t-test
##
## data: myc by cond
## t = -0.10365, df = 21.996, p-value = 0.9184
## alternative hypothesis: true difference in means between group Cancer and group NC is not equal to 0
## 95 percent confidence interval:
## -19.71435 17.83752
## sample estimates:
## mean in group Cancer      mean in group NC
## 36.44950 37.38792
```

(Is there a sign difference? Why? What happens if you change the order of NC and Cancer in the paired test?)

12.4 The paired t-test again: a single-sample t-test

What is the above test doing?

Create a new variable (e.g., call it “diff.nc.c”) that is the difference of myc in the NC and Cancer subjects and do a one-sample t-test.

Some basic statistics with R

```
diff.nc.c <- (myc.nc - myc.cancer)
t.test(diff.nc.c)

##
##  One Sample t-test
##
## data:  diff.nc.c
## t = 4.079, df = 11, p-value = 0.001823
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.432056 1.444777
## sample estimates:
## mean of x
## 0.9384167
```

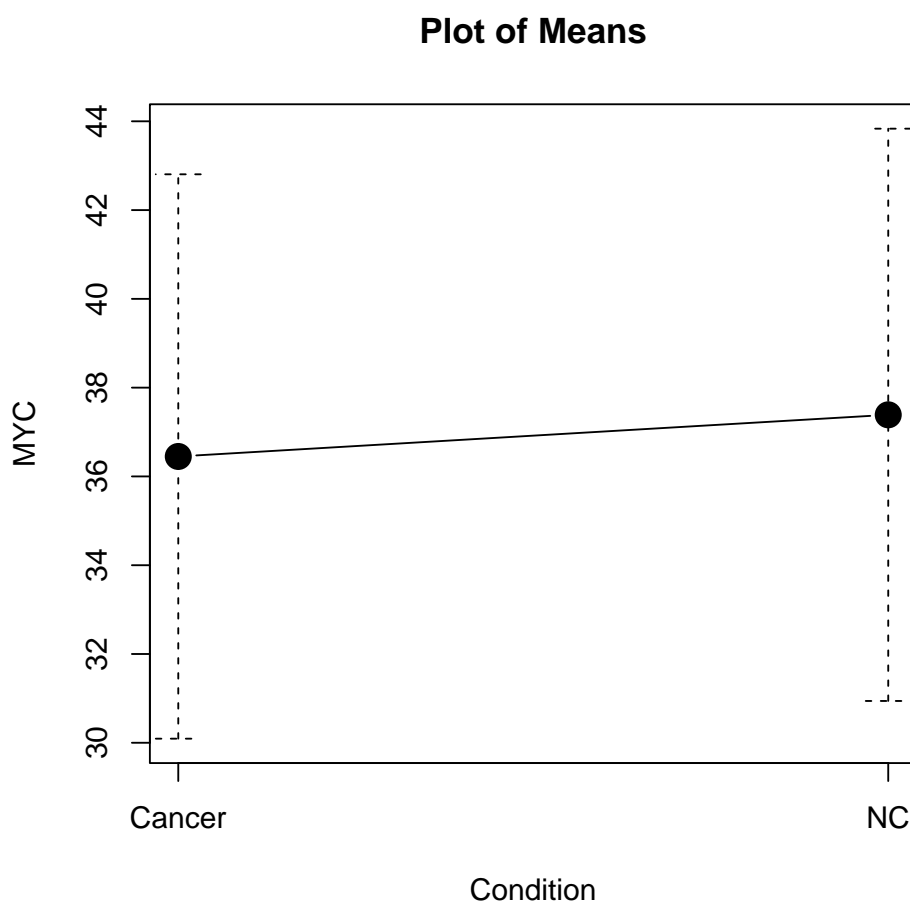
Compare the t-statistic, the degrees of freedom and the p-value with the paired t-test we did above. Again: what is it we are doing with the paired t-test? (Remember this, as this will be crucial when we use Wilcoxon's test).

Some basic statistics with R

12.5 Plots for paired data

What do you think of this plot?

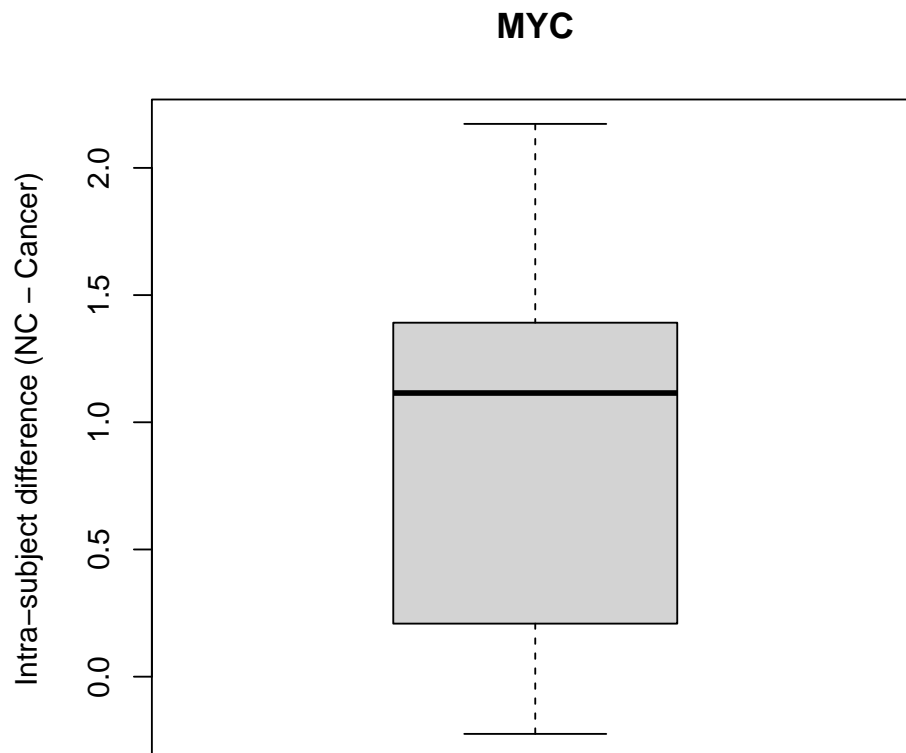
```
plotMeans(dmyc$myc, dmyc$cond, error.bars = "se", ylab = "MYC",  
          xlab = "Condition")
```



So what is a reasonable plot for paired data? A boxplot (or stripchart if few points) of the within-individual (or Intra-subject) differences is a very good idea.

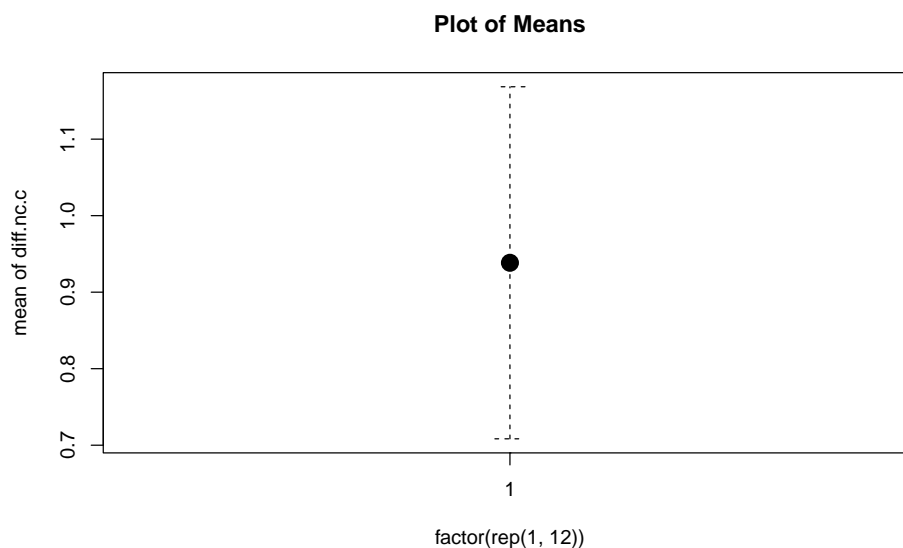
```
Boxplot( ~ diff.nc.c, data = merged3, xlab = "",  
        ylab = "Intra-subject difference (NC - Cancer)", main = "MYC")
```

Some basic statistics with R



Of course, we could do something like this if we wanted, which shows the mean of the intra-subject differences and its standard error:

```
plotMeans(diff.nc.c, factor(rep(1, 12)))
```

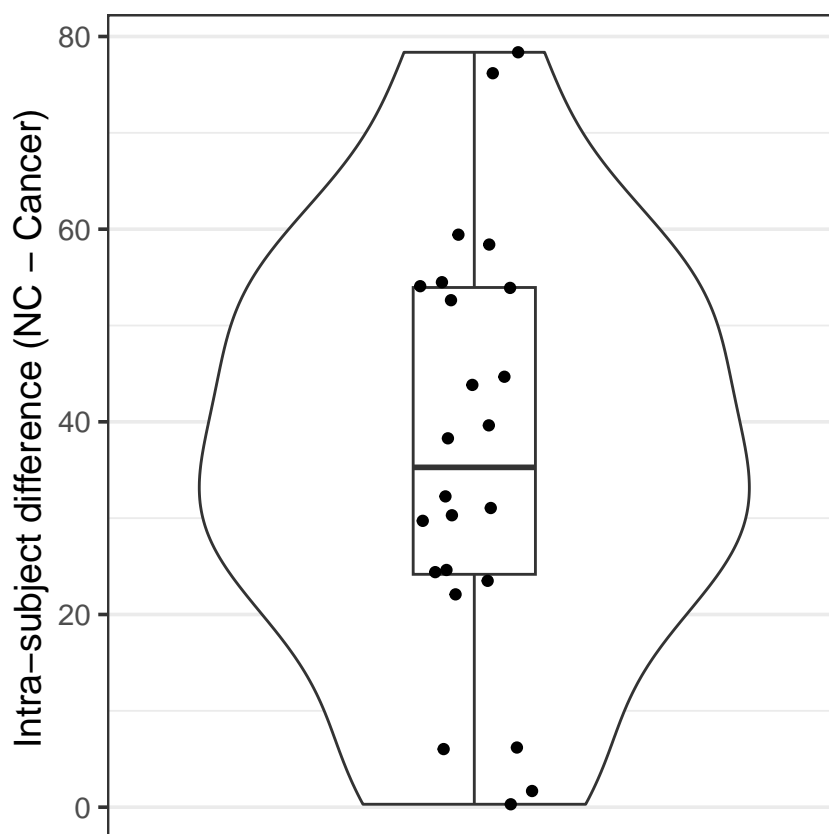


We can also create a violin plot with the boxplot inside and the points jittered, as we did before, using ggplot2:

Some basic statistics with R

```
library(ggplot2)

dftmp <- data.frame(y = dmycWide$diff.nc.c)
theplot <- ggplot(data = dftmp, aes(x = factor(1), y = y)) +
  geom_violin() + geom_boxplot(width = 0.2) +
  geom_jitter(colour = "black", width = 0.1, height = 0) +
  scale_x_discrete(breaks = NULL) +
  xlab("") +
  ylab("Intra-subject difference (NC - Cancer)") +
  theme_bw(base_size = 14, base_family = "sans") +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank())
print(theplot)
rm(dftmp, theplot)
```

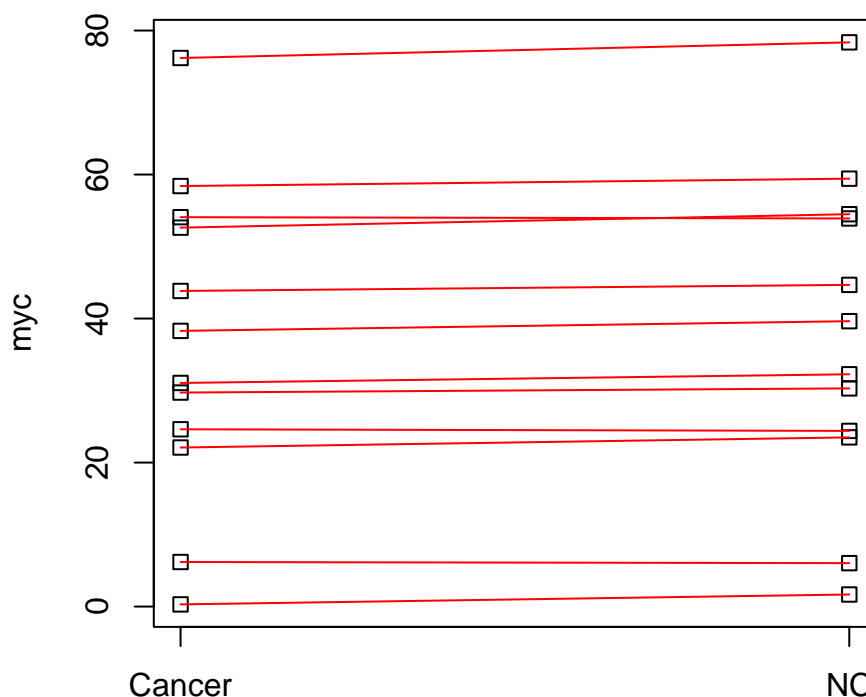


In this case, the violin plot is an overkill because we have too few data points. This is just to show it as an example.

Some basic statistics with R

A more elaborate plot directly shows both the NC and Cancer data, with a segment connecting the two observations of a subject (beware: this code does the job but it is not very efficient or elegant):

```
stripchart(myc ~ cond, vertical = TRUE, data = dmyc)
for(i in unique(dmyc$id))
  segments(x0 = 1, x1 = 2,
           y0 = dmyc$myc[dmyc$cond == "Cancer" & dmyc$id == i],
           y1 = dmyc$myc[dmyc$cond == "NC" & dmyc$id == i],
           col = "red")
```



Alternatively (though I do not see much of a gain here) we could have done:

```
stripchart(myc ~ cond, vertical = TRUE, data = dmyc)
f1 <- function(x) {
  segments(x0 = 1, x1 = 2,
           y0 = dmyc$myc[dmyc$cond == "Cancer" & dmyc$id == x],
           y1 = dmyc$myc[dmyc$cond == "NC" & dmyc$id == x],
           col = "red")
}
sapply(unique(dmyc$id), f1)
```

Now, look at these plots carefully, and understand that there are different sources of variation. In this particular example, there was a huge inter-subject variation in the expression of MYC; if we can control for it (e.g., with intra-subject measures) then we can detect what is, in relative terms, a small effect due to the condition.

Some basic statistics with R

(This is a short section, but it is very important. That is the reason I have added a few plots to beef it up. Seriously.)

12.6 Choosing between paired and two-sample t-tests

How the data were collected dictates the type of analysis. This is a crucial idea. You cannot conduct a two-sample t-test when your data are paired because your data are NOT independent (see also section 15).

This section is not about the analysis, but about the design. It is about “should I collect data so that data are paired or not?” A paired design controls for subject effects (correlation between the two measures of the same subject) but if there are none, then we are decreasing the degrees of freedom (by half): compare the degrees of freedom from the paired and the non-paired tests on the myc data. In most cases with biological data there really are subject effects that lead to large correlations of within-subject measurements. But not always.

This is a major topic (that cannot be done justice to in a paragraph). Sometimes the type of data are something you have no control over. But sometimes you do have control. How do you want to spend your money? Suppose you can sequence 100 exomes. Do you want to do 100 samples, 50 controls and 50 tumors, or do you want to do those same 100 exomes, but from 50 patients, getting tumor and non-tumor tissue? The answer is not always obvious. Go talk to a statistician.

12.7 A first taste of linear models

In the paired test, we implicitly have a model like this:

$$Expression.of.MYC = function(subject\ and\ condition) + \epsilon$$

which we make simpler (assuming additive contributions of each factor) as

$$Expression.of.MYC = effect.of.subject + effect.of.condition + \epsilon$$

Lets go and fit that model!

```
LinearModel.1 <- lm(myc ~ id + cond, data = dmyc)
summary(LinearModel.1)

##
## Call:
## lm(formula = myc ~ id + cond, data = dmyc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6173 -0.2224  0.0000  0.2224  0.6173
```

Some basic statistics with R

```
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  24.0393     0.4147  57.961 4.98e-15 ***
## idbqysitlvpm  14.4530     0.5635  25.647 3.66e-11 ***
## iddchlnopykg  -1.7155     0.5635  -3.044  0.01116 *
## idefzpcboidt -23.5210     0.5635 -41.739 1.82e-13 ***
## idejmkobsqrh   5.5030     0.5635   9.765 9.37e-07 ***
## idhyqjownkue  34.4060     0.5635  61.054 2.82e-15 ***
## idmculjayvhw -18.3985     0.5635 -32.649 2.65e-12 ***
## idqsmxexkcw  29.4820     0.5635  52.316 1.53e-14 ***
## idtrsyacmejh   7.1490     0.5635  12.686 6.56e-08 ***
## iduhbkifsnvw  19.7470     0.5635  35.041 1.23e-12 ***
## idytwgsplaef  29.0515     0.5635  51.553 1.80e-14 ***
## idzuhxmiyfos  52.7660     0.5635  93.634 < 2e-16 ***
## condNC         0.9384     0.2301   4.079  0.00182 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5635 on 11 degrees of freedom
## Multiple R-squared:  0.9997, Adjusted R-squared:  0.9993
## F-statistic: 2840 on 12 and 11 DF, p-value: < 2.2e-16

t.test(myc.nc, myc.cancer, paired = TRUE)

##
## Paired t-test
##
## data: myc.nc and myc.cancer
## t = 4.079, df = 11, p-value = 0.001823
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  0.432056 1.444777
## sample estimates:
## mean difference
##      0.9384167

## Note: the following syntax is no longer allowed.
## t.test(myc ~ cond, data = dmyc, paired = TRUE)
## This change in September 2023
## https://bugs.r-project.org/show\_bug.cgi?id=14359
## But this is OK
t.test(Pair(myc.nc, myc.cancer) ~ 1)

##
## Paired t-test
##
```

Some basic statistics with R

```
## data: Pair(myc.nc, myc.cancer)
## t = 4.079, df = 11, p-value = 0.001823
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  0.432056 1.444777
## sample estimates:
## mean difference
##      0.9384167
```

Run it and look at the line that has “cond” (so, using your common sense, ignore all the “id[blablabla]” lines). What is the t-value and the p-value for “cond”?

This is a linear model. And this particular one is also a two-way ANOVA.

```
Anova(LinearModel.1, type = "II")

## Anova Table (Type II tests)
##
## Response: myc
##           Sum Sq Df F value    Pr(>F)
## id       10815.9 11 3096.230 < 2.2e-16 ***
## cond         5.3  1   16.638  0.001823 **
## Residuals    3.5 11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Relate this to the figures above. This should all make sense. Regardless, linear models and ANOVAs will be covered in much more detail ... in the next part!

13 One-sample t-test

We have already used the one-sample t-test (section 12.4). You can of course compare the mean against a null value of 0, but you can also compare against any other value that might make sense (`mu` argument of the `t.test` function). Issues about assumptions are as for the two-sample (except, of course, there are no considerations of differences in variances among groups, since there is only one group). One-sample tests are probably not as common in many cases, as we are often interested in comparing two groups. But when you want to compare the mean of one group against a predefined value, the one-sample t-test is what you want.

14 Non-parametric procedures

14.1 Why and what

Non-parametric procedures refer to methods that do not assume any particular probability distribution for the data. These methods often proceed by replacing the original data by their ranks. They offer some protection against deviations from some assumptions (e.g., deviations from normality). But some times they might be testing a slightly different null hypothesis. Whether or not to use them can sometimes be a contentious issue. Some considerations that come into play are:

- Do the data look bad enough that a parametric procedure will lead to trouble?
- What about the relative efficiency of the non parametric procedure? (relative efficiency refers to the sample size required for two different procedures that have similar type I error rate to achieve the same power —more on power later). When assumptions are met, parametric methods do have more power (though not always a lot more). When assumptions are not met, nonparametric methods might have more power, or the correct Type I error, etc. Note that very, very, very small p-values are often not achievable with non-parametric methods (because of the way they work), and this is a concern with multiple testing adjustments in omics experiments (this will be covered in the next lesson).
- Is this test flexible enough? In particular, can I accommodate additional experimental factors?

We focus here on the Wilcoxon test. This is often the way to go when we have **ordinal** scale measurements and we want to compare two independent groups. Note, however, that the Wilcoxon test **requires interval scale data** for the single-sample Wilcoxon and the paired Wilcoxon (this is something that many websites and some textbooks get wrong); a simple illustration is shown in section 14.6.

However, **the independence assumption** is as important with Wilcoxon as with the t-test. Nonparametric tests are not “assumption-free” tests! (there is no such thing, in statistics or in life).

14.2 Wilcoxon rank-sum test or Mann-Whitney U test: 2 independent samples

This test applies to data of ordinal and interval scales. The basic logic is: put all the observations together, rank them, and examine if the sum of the ranks of one of the groups is larger (or smaller) than the sum of the ranks from the other¹¹. If this makes sense to you, you should understand why you can use both ordinal and interval scales.

¹¹The actual test statistic reported by R is not this one, but is one that is linearly related to it. Likewise, the equivalent Mann-Whitney U test uses another different statistic, but the test results are identical.

Some basic statistics with R

For example:

```
wilcox.test(p53 ~ cond, alternative="two.sided", data=dp53)

##
## Wilcoxon rank sum exact test
##
## data:  p53 by cond
## W = 22, p-value = 0.006473
## alternative hypothesis: true location shift is not equal to 0
```

14.3 What a rank-sum Wilcoxon test is not

Many people say “I’ll use a Wilcoxon test to compare the means”. Well ...the **Wilcoxon test is not a test of means**. It is often not even a test of medians. The Wilcoxon test can reject the null even if the medians are the same, and the Wilcoxon test can fail to reject the null even if the medians differ.

How is this possible? Please, make sure you read this:

https://www.graphpad.com/guides/prism/latest/statistics/stat_nonparametric_tests_dont_compa.htm

More details are available here: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1120984/> and in the Wikipedia entry for the Mann-Whitney U test (https://en.wikipedia.org/wiki/Mann%E2%80%93U_test). Many more details about permutation tests not testing what you might think they are testing are available, for example, from Christensen & Zabriskie, 2022. When Your Permutation Test is Doomed to Fail. *The American Statistician*, 76. <https://doi.org/10.1080/00031305.2021.1902856>

A few examples? Sure. Run the following in R:

```
## Will accept, medians differ
x <- c(rep(10, 1000), 11, rep(200, 1000))
y <- c(rep(10, 1000), 100, rep(200, 1000))
summary(x)

##      Min. 1st Qu.  Median     Mean 3rd Qu.     Max.
##       10       10       11      105     200      200

summary(y)

##      Min. 1st Qu.  Median     Mean 3rd Qu.     Max.
##       10       10      100      105     200      200

wilcox.test(x, y)

##
## Wilcoxon rank sum test with continuity correction
##
```

Some basic statistics with R

```
## data: x and y
## W = 2002000, p-value = 1
## alternative hypothesis: true location shift is not equal to 0

## Will accept, means differ
x <- c(rep(10, 1000), 1e9, rep(1000, 1000))
y <- c(rep(10, 1000), -1e9, rep(1000, 1000))
summary(x)

##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## 1.000e+01 1.000e+01 1.000e+03 5.003e+05 1.000e+03 1.000e+09

summary(y)

##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## -1.000e+09 1.000e+01 1.000e+01 -4.992e+05 1.000e+03 1.000e+03

wilcox.test(x, y)

##
## Wilcoxon rank sum test with continuity correction
##
## data: x and y
## W = 2004001, p-value = 0.9496
## alternative hypothesis: true location shift is not equal to 0

## Will reject, medians the same
x <- c(rep(10, 1000), 11, rep(12, 1000))
y <- c(rep(10, 1000), 11, rep(13, 1000))
summary(x)

##      Min. 1st Qu.  Median     Mean 3rd Qu.    Max.
##       10      10      11       11      12      12

summary(y)

##      Min. 1st Qu.  Median     Mean 3rd Qu.    Max.
##      10.0   10.0   11.0     11.5   13.0   13.0

wilcox.test(x, y)

##
## Wilcoxon rank sum test with continuity correction
##
## data: x and y
## W = 1502000, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0

## Will reject, means the same
x <- seq(from = 10, to = 20, length.out = 1000)
```

Some basic statistics with R

```
y <- c(rep(0, 999), 15000)
summary(x)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    10.0    12.5    15.0    15.0    17.5    20.0

summary(y)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##         0         0         0      15         0   15000

wilcox.test(x, y)

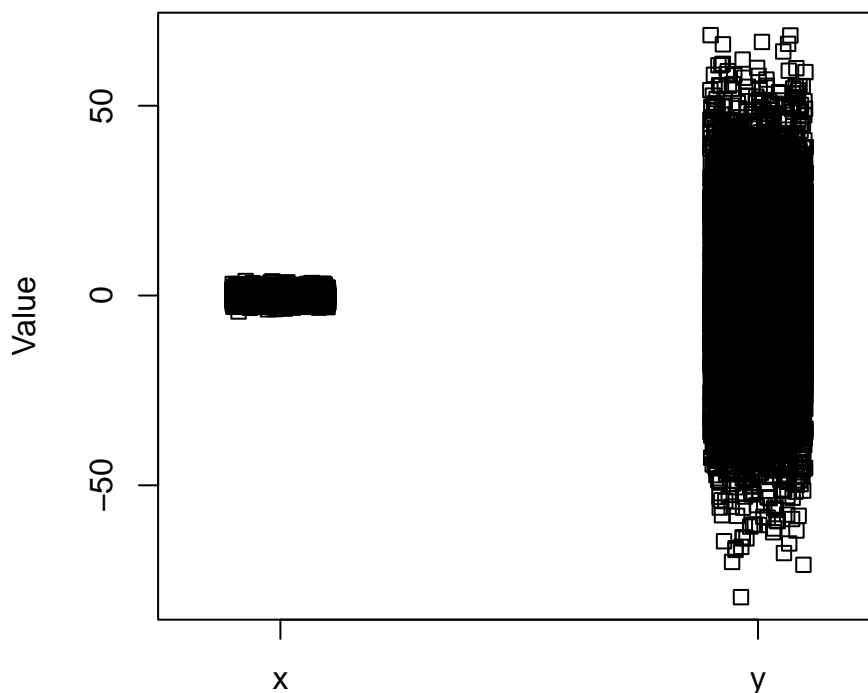
##
## Wilcoxon rank sum test with continuity correction
##
## data:  x and y
## W = 999000, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0

## Will not reject so
## contrary to Conover, not really testing  $F(x) = G(x)$ 
x <- rnorm(10000, mean = 0, sd = 1)
y <- rnorm(10000, mean = 0, sd = 20)
wilcox.test(x, y)

##
## Wilcoxon rank sum test with continuity correction
##
## data:  x and y
## W = 50401776, p-value = 0.3251
## alternative hypothesis: true location shift is not equal to 0

stripchart(c(x, y) ~ c(rep("x", 10000), rep("y", 10000)), vertical = TRUE,
           method = "jitter", ylab = "Value")
```

Some basic statistics with R



This is a very similar case

```
x <- rep(c(-2, -1, 1, 2), 1000)
y <- rep(c(-4, -2, 2, 4), 1000)
wilcox.test(x, y)

##
## Wilcoxon rank sum test with continuity correction
##
## data: x and y
## W = 8e+06, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
## Oh, you say you dislike the many identical values?
x <- x + rnorm(1000, 0, sd = 0.001)
y <- y + rnorm(1000, 0, sd = 0.002)
## No identical values
length(unique(c(x, y)))
## [1] 2000
wilcox.test(x, y)

##
## Wilcoxon rank sum test with continuity correction
##
## data: x and y
## W = 8003968, p-value = 0.9694
```

Some basic statistics with R

```
## alternative hypothesis: true location shift is not equal to 0

summary(x)

##      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
## -2.0027335 -1.2513779 -0.0001827  0.0000068  1.2511573  2.0029256

summary(y)

##      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
## -4.006275 -2.502308 -0.000378  0.000049  2.502711  4.005601
```

What is the Wilcoxon test testing? As the Wikipedia entry says “the null hypothesis that, for randomly selected values X and Y from two populations, the probability of X being greater than Y is equal to the probability of Y being greater than X .”

Summary: do not use a Wilcoxon expecting it to test for mean differences. And below we will emphasize another related message: do not use a Wilcoxon because of some ill-motivated fear of using the t-test.

14.4 Wilcoxon signed-rank test: matched-pairs or single sample test

When we have paired data, the idea here is to assess whether the within-pair differences are symmetrically distributed (generally around zero). As we are talking about symmetry, this means that this test can be used **only with interval scale** data (see also section 14.6).

The null hypothesis is that the differences are symmetrically distributed around some value (generally zero). We can therefore reject the null if the differences are symmetrical around some other value, or if the differences are not symmetrical around the posited value (as we said, generally 0), or a combination of both. Put differently, if we reject the null, we could be rejecting it for different reasons (asymmetry, symmetry around a value different from the one specified by the null), or combinations of those reasons¹². As usual, try to look at plots to disentangle what might be the case. It is often the case that, if we are dealing with within-individual differences, for example a “before vs. after” design, those within-individual differences will tend to be symmetric even if the center of the distribution shifts (regardless of the original distribution of the “before” measure —see also the example in 16), but not always.

¹²This is discussed, for example, in <https://stats.stackexchange.com/questions/348057/wilcoxon-signed-rank-symmetry-assumption>. Note, by the way, that this is not unique to this test: many tests can reject the null for a variety of reasons, and this is common in permutation-based tests.

Some basic statistics with R

How does the test work? In a nutshell, this is what the test does: for each pair, it computes the difference of the two measures (thus, taking differences must make sense, and it does not for ordinal data but it does for interval data). These differences (discarding the sign) are then ranked, and then the sum of the ranks of all the positive differences is computed and compared to the null distribution.

```
wilcox.test(myc.nc, myc.cancer, alternative = 'two.sided', paired = TRUE)

##
## Wilcoxon signed rank exact test
##
## data: myc.nc and myc.cancer
## V = 72, p-value = 0.006836
## alternative hypothesis: true location shift is not equal to 0
```

You can verify that the results are the same as doing a single sample Wilcoxon, or a Wilcoxon signed-rank test, on the within-subject differences.

```
wilcox.test(diff.nc.c)

##
## Wilcoxon signed rank exact test
##
## data: diff.nc.c
## V = 72, p-value = 0.006836
## alternative hypothesis: true location is not equal to 0
```

Before we move on: what is the role of symmetry in the paired t-test? To try to make progress thinking about this, remember that the paired t-test is the same as testing the mean of the within-individual differences against a pre-specified mean. Go to section [Symmetry and the paired t-test](#), 16 for an example.

14.5 A bad (very bad, terrible!) way to choose between nonparametric and parametric procedures

Don't do the following:

1. Carry out a test for normality (I guess for each group separately if and independent-samples t-test —testing for normality collapsing the two groups would be absolutely nonsensical).
2. If failed, use a nonparametric test. Otherwise a t-test.

There are several problems with the above procedure, among them:

- Tests for normality do not have a power of 1, and in fact can have very low power with small sample sizes.

Some basic statistics with R

- They can have power to detect a minor and irrelevant deviation from normality (e.g., slightly heavy tails) that have no effects on, say, a t-test. So we should really prefer a t-test, but we might be misled into doing a Wilcoxon.
- They might fail to detect a large deviation from normality in a very non-normal small sample where we really, given our ignorance, might want to conduct a Wilcoxon test.
- They lead to “sequential testing” problems, where the second p-value (the one from the t or Wilcoxon) cannot be simply interpreted at face value (as it is conditional on the normality test).

Even if this procedure is (or was) common in some circles, you now know why it is a bad idea. Please, do not do this.

Again, this is the type of “cargo-cult statistics”¹³ that does no body any good. Think about your data, what you want to test (is it means? or is it something else?), what are reasonable assumptions (heavy tails, asymmetries, etc) and whether the deviations are likely to cause trouble. The above simple-minded (brain-dead, actually) procedure is no substitute for thinking.

14.6 Wilcoxon’s paired test and interval data

If you understand the logic of what the two Wilcoxon tests are doing, you should understand why the one for the two-sample case works with ordinal data but the one for the matched pairs requires interval data. This explores the issue further.

In particular, if we were to transform the data using a monotonic non-linear transformation (e.g., a log transformation), the Wilcoxon test for two groups should never be affected, but the one for the paired case could be affected. Why? Because in the first case we rank the values, and the ranks of the values are the same as the ranks of the values after any monotonic transformation. However, the rank of the within-subject differences might differ if we use a transformation on the original data and then rank the within-subject differences.

```
## Without logs
wilcox.test(dmyc$myc[1:12], dmyc$myc[13:24], paired = TRUE)

##
## Wilcoxon signed rank exact test
##
## data: dmyc$myc[1:12] and dmyc$myc[13:24]
## V = 6, p-value = 0.006836
## alternative hypothesis: true location shift is not equal to 0

## After taking logs
wilcox.test(log(dmyc$myc[1:12]), log(dmyc$myc[13:24]), paired = TRUE)
```

¹³“going through the motions with scant understanding”; Stark and Saltelli, 2018. Cargo-cult statistics and scientific crisis. *Significance*, 15.

Some basic statistics with R

```
##  
## Wilcoxon signed rank exact test  
##  
## data: log(dmyc$myc[1:12]) and log(dmyc$myc[13:24])  
## V = 9, p-value = 0.01611  
## alternative hypothesis: true location shift is not equal to 0
```

Notice how the statistic and the p-value change. That would not happen if the test could be applied to ordinal data.

It is also easy to create examples that show that the one-sample Wilcoxon does require interval data. This is left as a simple exercise.

15 Non-independent data

15.1 Multiple measures per subject

Paired data are non-independent: they are associated via the subject, or id. There are other forms of dependency. Many. We will now look at the most common one: multiple measures per subject.

We will use the `BRCA2.txt` data set. Import it, etc. Look at the data carefully. It looks like there are 3 observations per subject in a total of 13 subjects. Basically, each subject has been measured 3 times. Thus, there are not 39 independent measures. Again, ask yourself: what is the true “experimental unit” (or observational unit)? It is arguably the subject, which is the one that has, or has not, cancer in this case. The 3 measures per subject are just that: multiple measures of the same experimental unit. This is an idea that should be crystal clear; make sure you understand it.

We can do a t-test ignoring this fact, but it would be wrong. Look at the degrees of freedom:

```
t.test(brca2 ~ cond, data = dbrca)

##
## Welch Two Sample t-test
##
## data: brca2 by cond
## t = -2.1969, df = 28.061, p-value = 0.03645
## alternative hypothesis: true difference in means between group Cancer and group NC is not equal to 0
## 95 percent confidence interval:
## -3.9309162 -0.1377338
## sample estimates:
## mean in group Cancer      mean in group NC
##           5.953208           7.987533
```

What can we do? The most elegant approaches are not something we can cover here¹⁴. However, fortunately for us in **this particular case**, all subjects have been measured the same number of times. Thus, we can simply take the mean per subject, and then do a t-test on those mean values per subject.

We want to aggregate (using the mean) the values of `brca2`, and we want to aggregate by “id”. However, we want to keep “cond” also. So we will aggregate by “cond” and “id”. In fact, this is a double check that each subject is in one, and only one, of the groups. Call this data “aggbrca”.

```
aggbrca <- aggregate(dbrca[,c("brca2")], drop = FALSE,
                     by = list(cond = dbrca$cond,
```

¹⁴A general approach is a mixed-effects linear model with random effects for subjects; alternatively, and in this simple case, we can use an ANOVA with the correct error term, for instance from a multistratum linear model similar to the ones used to analyze split-plot experiments.

Some basic statistics with R

```
id = dbrca$id), FUN = mean)
```

You can also call `aggregate` in a simpler way in this case:

```
aggbrc <- aggregate(brca2 ~ cond + id, data = dbrca, FUN = mean)
```

We want to double check that we have the exact same number of measures per subject. How? A simple procedure is to aggregate again (give the output another name) but using “length” instead of “mean”. Now we will be aggregating over only “id” (so we aggregate brca2 and cond) and also over both “id” and “cond”, to double check. Do you understand why we are doing this?

```
aggregate(brca2 ~ cond + id, data = dbrca, FUN = length)
```

```
##      cond      id brca2
## 1  Cancer btghoiazuc    3
## 2      NC cdxbfpeutk    3
## 3  Cancer glrehypdoi    3
## 4      NC iakfbsgurp    3
## 5  Cancer ifdqzsycev    3
## 6  Cancer jhemxnvrco    3
## 7  Cancer mextndbrqh    3
## 8      NC olpqrqeniu    3
## 9  Cancer rvmigxlht    3
## 10 Cancer wcmvlqudik    3
## 11      NC wcxzfppdsg    3
## 12      NC wqvjzkncl    3
## 13 Cancer zyipsexbhr    3
```

```
aggregate(brca2 ~ id, data = dbrca, FUN = length)
```

```
##      id brca2
## 1 btghoiazuc    3
## 2 cdxbfpeutk    3
## 3 glrehypdoi    3
## 4 iakfbsgurp    3
## 5 ifdqzsycev    3
## 6 jhemxnvrco    3
## 7 mextndbrqh    3
## 8 olpqrqeniu    3
## 9 rvmigxlht    3
## 10 wcmvlqudik    3
## 11 wcxzfppdsg    3
## 12 wqvjzkncl    3
## 13 zyipsexbhr    3
```

```
aggregate(. ~ id, data = dbrca, FUN = length)
```

Some basic statistics with R

```
##           id brca2 cond
## 1 btghoiazuc     3    3
## 2 cdxbfpeutk     3    3
## 3 glrehypdoi     3    3
## 4 iakfbsgurp     3    3
## 5 ifdqzsycev     3    3
## 6 jhemxnvrco     3    3
## 7 mextnnbrqh     3    3
## 8 olpzrqeniu     3    3
## 9 rvnmigxlht     3    3
## 10 wcmvlqudik     3    3
## 11 wcxzfpqdsq     3    3
## 12 wqvjzkncl     3    3
## 13 zyipsexbrh     3    3
```

Now, let's do a t-test on the aggregated data. Pay attention to the degrees of freedom (and the statistic and p-values):

```
t.test(brca2 ~ cond, alternative = 'two.sided', conf.level = .95,
       var.equal = FALSE,
       data = aggbrca)

##
## Welch Two Sample t-test
##
## data: brca2 by cond
## t = -1.2832, df = 8.0293, p-value = 0.2352
## alternative hypothesis: true difference in means between group Cancer and group NC is not equal to 0
## 95 percent confidence interval:
## -5.687861 1.619211
## sample estimates:
## mean in group Cancer      mean in group NC
##          5.953208          7.987533
```

You can see that doing things correctly makes, in this case, a large difference. When we do things incorrectly, we can end up believing that there is a strong effect when, really, there is no evidence of effect.

What if the subjects had been measured a different number of times? What would be the problems of simply taking the averages over subjects?

Of course, we could have done

```
(aggbrca3 <- aggregate(brca2 ~ cond + id, data = dbrca,
                      FUN = function(x) c(Mean = mean(x), N = length(x))))

##      cond      id brca2.Mean brca2.N
```

Some basic statistics with R

```
## 1 Cancer btghoiazuc 3.197667 3.000000
## 2 NC cdxbfpeutk 7.201667 3.000000
## 3 Cancer glrehypdoi 5.801667 3.000000
## 4 NC iakfbgurg 6.756000 3.000000
## 5 Cancer ifdqzsycev 7.395333 3.000000
## 6 Cancer jhemxnvrco 9.275667 3.000000
## 7 Cancer mextndbrqh 7.094000 3.000000
## 8 NC olpzrqeniu 9.918333 3.000000
## 9 Cancer rvmigxlht 8.742000 3.000000
## 10 Cancer wcmvlqudik 4.015000 3.000000
## 11 NC wcxzfpgdsg 11.713333 3.000000
## 12 NC wqvjzknsl 4.348333 3.000000
## 13 Cancer zyipsexbrh 2.104333 3.000000
```

```
## the first not that useful immediately
```

```
library(doBy)
```

```
(aggbrca4 <- summaryBy(brca2 ~ cond + id, data = dbrca,
                        FUN = function(x) c(Mean = mean(x), N = length(x))))
```

```
##      cond      id brca2.Mean brca2.N
## 1 Cancer btghoiazuc 3.197667      3
## 2 Cancer glrehypdoi 5.801667      3
## 3 Cancer ifdqzsycev 7.395333      3
## 4 Cancer jhemxnvrco 9.275667      3
## 5 Cancer mextndbrqh 7.094000      3
## 6 Cancer rvmigxlht 8.742000      3
## 7 Cancer wcmvlqudik 4.015000      3
## 8 Cancer zyipsexbrh 2.104333      3
## 9 NC cdxbfpeutk 7.201667      3
## 10 NC iakfbgurg 6.756000      3
## 11 NC olpzrqeniu 9.918333      3
## 12 NC wcxzfpgdsg 11.713333      3
## 13 NC wqvjzknsl 4.348333      3
```

```
t.test(brca2.Mean ~ cond,
       var.equal = FALSE,
       data = aggbrca4)
```

```
##
```

```
## Welch Two Sample t-test
```

```
##
```

```
## data: brca2.Mean by cond
```

```
## t = -1.2832, df = 8.0293, p-value = 0.2352
```

```
## alternative hypothesis: true difference in means between group Cancer and group NC is not equal
```

```
## 95 percent confidence interval:
```

Some basic statistics with R

```
## -5.687861  1.619211
## sample estimates:
## mean in group Cancer      mean in group NC
##              5.953208              7.987533

t.test(brca2[, "Mean"] ~ cond,
       var.equal = FALSE,
       data = aggbrca3)

##
## Welch Two Sample t-test
##
## data:  brca2[, "Mean"] by cond
## t = -1.2832, df = 8.0293, p-value = 0.2352
## alternative hypothesis: true difference in means between group Cancer and group NC is not equal
## 95 percent confidence interval:
## -5.687861  1.619211
## sample estimates:
## mean in group Cancer      mean in group NC
##              5.953208              7.987533
```

15.2 Nested or hierarchical sources of variation

A general way of thinking about these issues is that we can have nested, or hierarchical, levels of variation (e.g., multiple measures per cell, multiple cells per subject, multiple subjects for two different treatments) and it is crucial to understand what each level of variation is measuring (e.g., the difference between technical and biological variation) and to conduct the statistical analysis in a way that do incorporate this nestedness. A brief introductory 2-page paper with biomedical applications is Blainey et al., 2014, “Points of significance: Replication”, in *Nature Methods*, 2014, 11, 879–880, where they discuss issues of sample size choice at different levels.

By the way, not surprisingly, split-plot ANOVA and nested ANOVA are typical, traditional, ways of analyzing these kinds of designs. Many of these issues are, thus, naturally addressed in the context of linear models.

15.3 Non independent data: extreme cases

In the extreme, this problem can lead to data that should not be analyzed at all because there is, plainly, no statistical analysis that can be carried out. For instance, suppose we want to examine the hypothesis that consuming DHA during pregnancy leads to increased myelination in the hippocampus of the progeny. An experiment is set to test this idea, comparing the effects on myelination of mice that have been born to female mice with and without a DHA supplement¹⁵.

¹⁵This is based on an actual data set I was once asked to analyze. I've changed enough details — no DHA or anything similar. But the outcome was the same: no analyses were possible at all.

Some basic statistics with R

Now, a colleague comes to you with the data. She claims there are 40 data points, 20 from brains of newborn mice under the DHA supplement and 20 from brains of newborn mice without the supplement. OK, it looks good: it seems we can carry out an independent samples t-test with about 38 degrees of freedom. However, on asking questions, this is what you find out:

- A total of two female mice were used. One female was given food with DHA and one female without. They were fed the specified diet, and then they mated and got pregnant.
- From the litter for each female, five newborn mice were sacrificed immediately after birth, and four tissue slides were prepared from each brain (thus, 20 data points per female).

No statistical analysis can be conducted here to examine the effects of DHA: there is only one data point per experimental unit. We cannot do a t-test in the right way: there are no degrees of freedom because there is no way to estimate the variance. To put it simply, there is no way to tell whether any differences that could be seen are due to DHA or to the female herself or to any other factor that might have been confounded with the single female per treatment (color of the gloves of the technician or side in the animal room or how nice the male was while mating or ...).

It is this simple: **nothing** can be said from this data about the effects of DHA. (It is not even worth importing the data for this analysis. Maybe it is interesting to get a preliminary idea of the intra-brain variation in myelination, but not for the original question of the DHA effect).

There are some recent papers about issues like this that go over the pseudoreplication idea (e.g., Lazic, 2010, *BMC Neuroscience* and Lazic et al., 2018, *PLoS Biology*, "What exactly is 'N' in cell culture and animal experiments?") and this is probably a pervasive (but difficult to detect) problem. This kind of meaningless analysis can lead to lots of non-reproducible results.

Which brings us to the fundamental idea of **thinking carefully about the experimental design**, something we will take a quick look at in the linear models section.

15.4 More non-independences and other types of data

What if some subjects had cousins and brothers in the data set? And if some of them came from the same hospital and other from other hospitals? And ... ? This all lead to multilevel and possible crossed terms of variance. Mixed effects models can be used here. Go talk to a statistician (after looking at the rest of these notes).

However, for simple cases and to get going while we talk to the statistician, the approach we used above (collapsing the data over lower levels, leaving data that are independent at the experimental unit level) can some times be used in other scenarios. The independence assumption is actually crucial in many statistical analysis, be they t-

Some basic statistics with R

tests, ANOVAs, chi-squares, regressions, etc, etc. (As has been mentioned repeatedly, there are ways to incorporate or deal with the non-independence, for categorical, ordinal, and interval data, but they are far from trivial).

To make the point more clear, this is an example from the data for the TFM (“trabajo fin de master”) from a former student of BM-1¹⁶. Briefly, she was interested in the rates of chromosomal aberrations in different types of couples that went to a fertility clinic. For instance, suppose you want to examine incidence of aneuploidy in embryos from two groups of fathers, “younger fathers” and “older fathers”. The simplest idea here is to use a chi-square (χ^2) test to compare the frequency of aneuploidies between older and younger fathers (you will see chi-square tests in Lesson 4). But the problem is that each couple (each father in this case) contributes multiple embryos and we cannot simply do a chi-square counting embryos, as we would again run into a non-independence problem. A simple approach, especially if each father contributes the same number of embryos, is to calculate, for each father, the proportion of embryos with aneuploidies. And then, to examine if that per-father proportion of aneuploidies differs between older and younger fathers with, say, an independent samples t-test (possibly of suitably transformed data) or a two-samples Wilcoxon test.

16 Symmetry and the paired t-test

In the paired t-test symmetry of the within-subject differences is important. This we know, because the paired t-test is just a one-sample t-test applied to within subject differences (and the usual assumptions of the one-sample t-test apply, one of them being that the data have an approximately symmetrical distribution).

In the paired t-test, though, the within-individual differences (more generally, the within-experimental unit differences) are coming from, well, differences of, say, “before” and “after” distributions. How relevant is the distribution of those “before” and “after”? In other words, if we define within-individual-difference as W , with $w_i = u_i - v_i$ (where U could be the “after” and V the “before”), what is the relevance of the distributions of U and V ? What can we expect about W ?

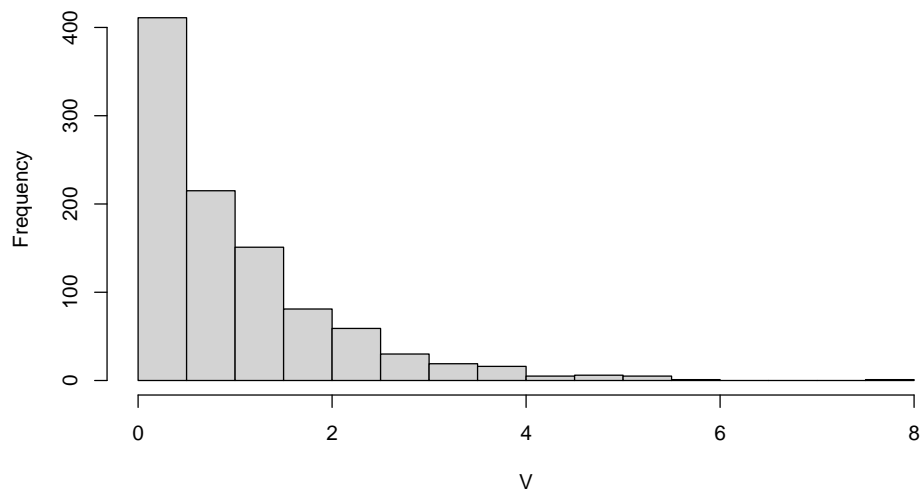
If you have access to it, Rupert Miller’s “Beyond Anova” (Chapman and Hall, 1997), bottom of p. 8 and p. 9 discusses this issue. In class, we will quickly discuss the code below:

```
## "num" random deviates from an exponential.  
## Play with num if you want. We use a very large num  
## to make it easier to see the patterns.  
num <- 1000  
V <- rexp(n = num, rate = 1)  
## The exponential is clearly asymmetric  
hist(V)
```

¹⁶This is true. I am not making it up.

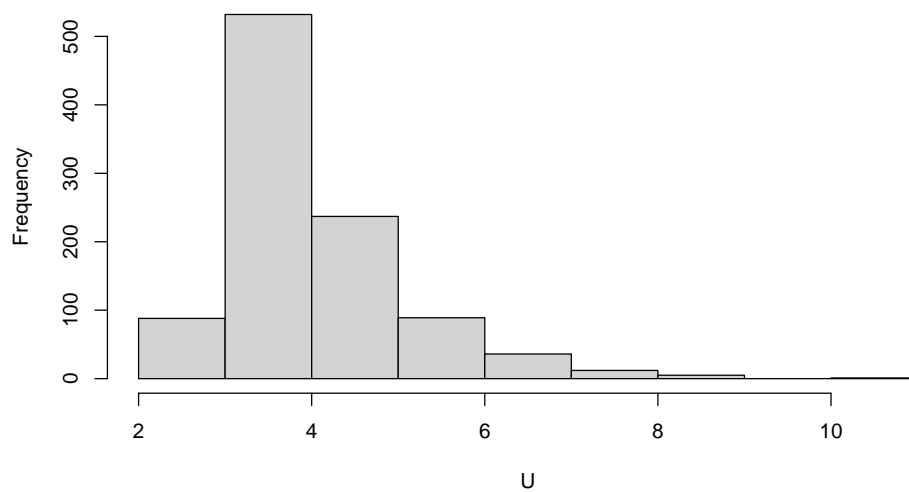
Some basic statistics with R

Histogram of V



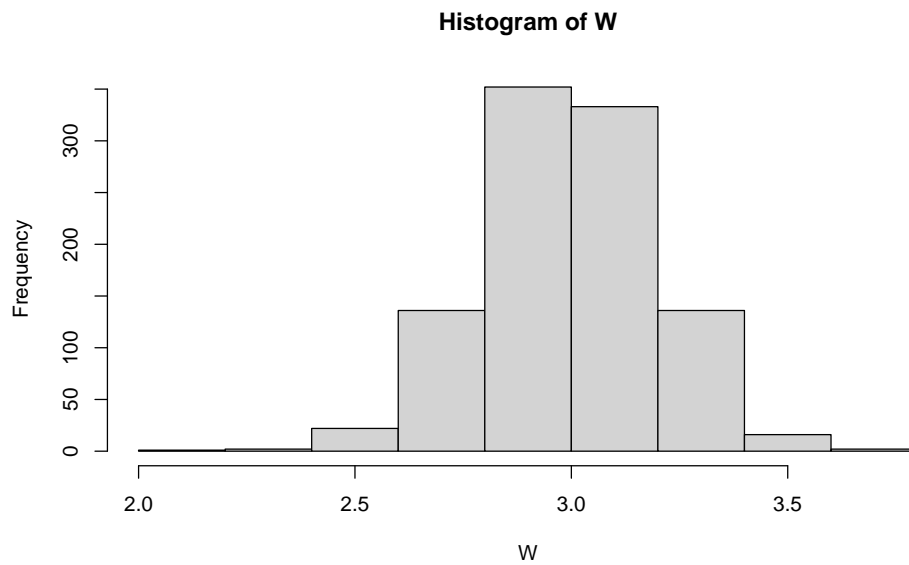
```
## Shift those numbers by adding 3, and a little bit of noise  
U <- V + 3 + rnorm(num, sd = 0.2)  
hist(U)
```

Histogram of U



```
## What about the differences?  
W <- U - V  
hist(W)
```


Some basic statistics with R



So in many cases skewness (asymmetry) tends to cancel out (e.g., for example when we add a constant value plus some random, symmetric noise).

And this emphasizes something else: when examining if assumptions are reasonable, think about what you are examining. Here, the distribution that matters is the distribution of the within-individual differences (W), not each of the separate distributions U or V : your test is about individual differences, W , not about U or V .

Part II

17 Introduction to ANOVAs, regression, and linear models

Linear models and their extensions (which include logistic regression, but also survival analysis, many classification problems, non-linear models, analysis of experiments, dealing with many types of dependent data, etc, etc) are one fundamental topic in statistics. Here, we will only scratch the surface. But you should come away from this lesson understanding that these methods are extremely powerful and flexible and that they can be used to address a huge variety of different research questions. You would spend your time wisely if you at least took a look at some of the references we provide at the end. To emphasize again: ANOVAs, regression, ANCOVAs, are just special type of linear models; the terminology is not that important, but I'll use those terms as they might be more familiar to you.

17.1 Files we will use

- This one
- `MIT.txt`
- `Cholesterol.txt`
- `AnAge_birds_reptiles.txt`
- `CystFibr2.txt`

18 Comparing more than two groups

18.1 Recoding variables

Import `MIT.txt` and call the object `dmit`. These are data about mitochondrial activity related to three different training regimes.

```
dmit <- read.table("MIT.txt", header = TRUE)
```

Whoever entered the data, however, used a number for “training”, which is misleading, because this is really a categorical variable. The first thing we must do, then, is fix that.

Note the `factor` function call:

```
dmit$ftraining <- factor(dmit$training,  
                        labels = c('Morning', 'Lunch', 'Afternoon'))
```

As usual, make sure to look at the data set.

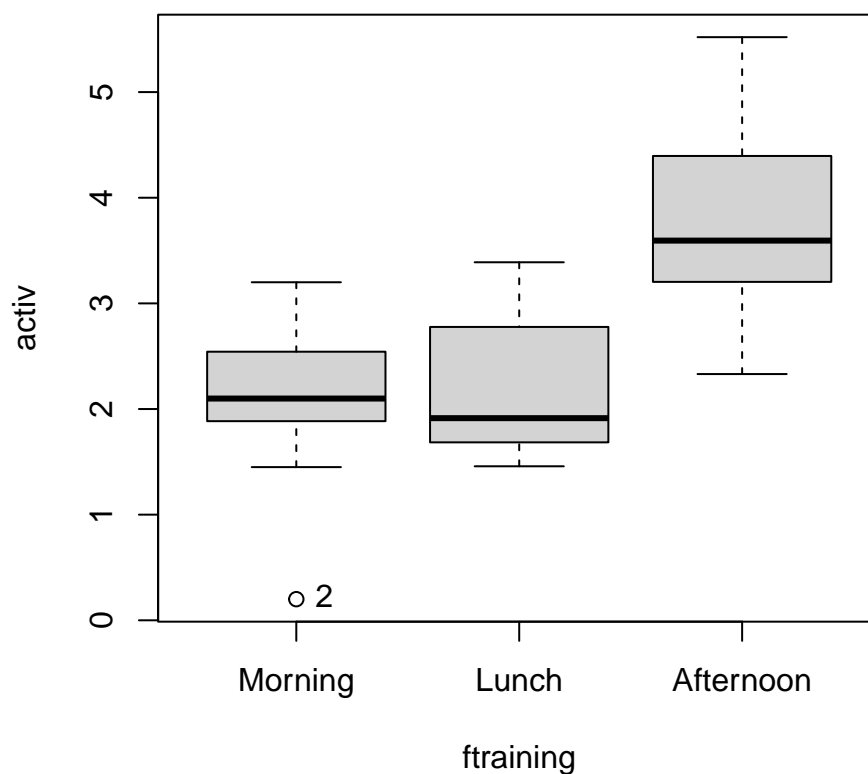
If we were to not recode the factor (or use the original “training”) it would be a disaster (look at the output in section [What if we did not recode training?](#)).

18.2 A boxplot

You are advised to also plot the data. For instance, this will do:

```
Boxplot(activ ~ ftraining, data = dmit, id.method = "y")
```

Some basic statistics with R

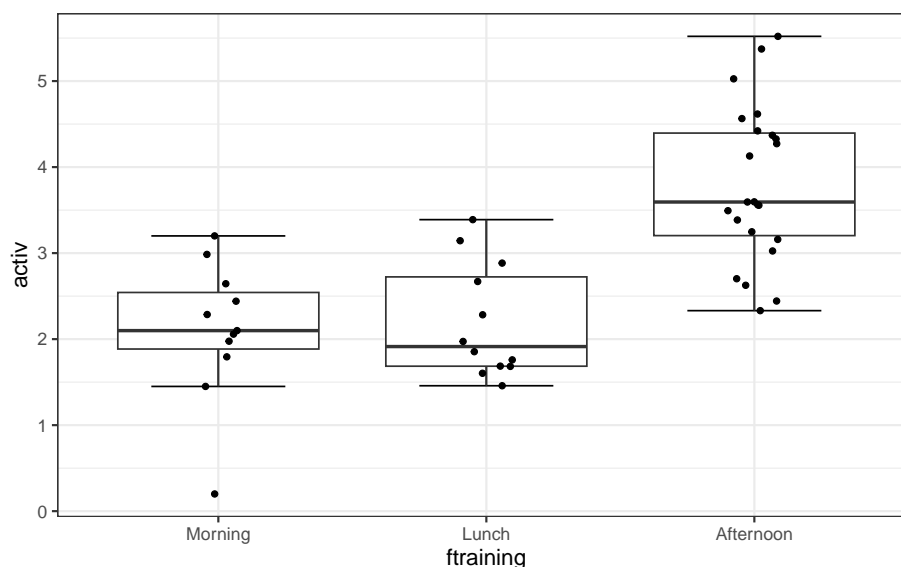


(The output might show a “2”; that is the identifier —row name— of a point that has been flagged as a potential outlier and we will silent that output from now on in these notes).

We can try a boxplot overimposing the observations (and, if you want, you can also try a violin plot) using `ggplot2`, as we have seen before:

```
require("ggplot2")
tmpdf <- data.frame(x = dmit$ftraining, y = dmit$activ)
theplot <- ggplot(data = tmpdf, aes(x = factor(x), y = y)) +
  stat_boxplot(geom = "errorbar", width = 0.5) +
  geom_boxplot(outlier.colour = "transparent") +
  geom_jitter(colour = "black", width = 0.1, height = 0) +
  xlab("ftraining") +
  ylab("activ") +
  theme_bw(base_size = 14, base_family = "sans")
print(theplot)
```

Some basic statistics with R



```
rm(theplot, tmpdf)
```

18.3 An ANOVA; some basic theory and output

We want to see if time of exercise makes any difference. Conducting three t-tests is not the best way to go here: our global null hypothesis is $\mu_{Morning} = \mu_{Lunch} = \mu_{Afternoon}$ and that is what ANOVA will allow us to test directly.

```
AnovaMIT <- aov(activ ~ ftraining, data = dmit)
summary(AnovaMIT)

##              Df Sum Sq Mean Sq F value   Pr(>F)
## ftraining      2  31.15   15.57   22.89 1.7e-07 ***
## Residuals     43  29.26    0.68
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Can you interpret the output?

Note: we are using `aov`. We could also use `lm` and then show the summary with function `anova`. Sometimes, these are just syntactic features —annoyances?— of R. I'll explain them in class. But the key concepts do not change. For example, go to “Statistics -> Fit models models -> Linear model” (Linear model, NOT regression: do you know why?). Then, go to “Models -> Hypothesis test -> ANOVA model”

```
LinearModel.1 <- lm(activ ~ ftraining, data = dmit)
Anova(LinearModel.1, type="II")

## Anova Table (Type II tests)
##
## Response: activ
```

Some basic statistics with R

```
##           Sum Sq Df F value    Pr(>F)
## ftraining 31.147  2  22.887 1.704e-07 ***
## Residuals 29.260 43
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In most of this course we will generally prefer this second route (i.e., functions `lm` with `Anova`).

[See other file:] We will cover this in more detail in class, in case you do not remember ANOVA. So read **now** the 2 and a half pages from Peter Dalgaard's book available in Moodle ([anova-basic-theory.pdf](#)), then read [anova-theory-even-simpler.pdf](#), and then re-read Dalgaard's. If there are questions, we will cover them in class.

Now, things to notice about the output:

- The two rows; one of them is the effect you are interested in (ftraining)
- The “Df” column: those are the degrees of freedom (three groups - 1 for “ftraining”). Do you know what degrees of freedom are? If not, ask in class.
- The two columns Sum Sq (Sum of Squares) and Mean Sq (Mean Squares). Sum of Squares is a quantity related to the variance. Mean Squares is obtained from the ratio of Sum Sq over Df. Then, we use Mean Sq to compare how much variance there is between groups related to the variance within groups: the F value is the ratio of Mean Sq of ftraining over Mean Sq of the residuals. The larger that F value, the more evidence there is of groups being different.
- There is a p-value associated with that F value. In this case it is very small.

By the way, notice how we created a “Model” which we called `AnovaMIT`. But we could have named it differently.

18.4 Confidence intervals for the parameters of the model

We can easily do

```
confint(AnovaMIT)

##           2.5 %    97.5 %
## (Intercept)  1.6014112 2.6045888
## ftrainingLunch -0.5985849 0.7902515
## ftrainingAfternoon 1.0844974 2.3041983
```

For one thing, we are all well aware that simply looking at p-values is not the only thing we want to do.

Some basic statistics with R

Sometimes `confint` will give us a lot of insight. But often it will not be easy to relate it back to our original scientific question. One of the reasons is that the actual parameters of the fitted model depend, well, on the parameterization (see details in section [Anova tables from `lm` et al.: understanding the coefficients and parameters](#), or just take it on faith). So, often, we will want to explicitly ask “Which means are different”, and that is what we do next.

18.5 So which means are different? Multiple comparisons

That small p-value leads us to reject the null hypothesis $\mu_{Morning} = \mu_{Lunch} = \mu_{Afternoon}$. So there is strong evidence that all three means are not equal. But which one(s) is(are) different from the other(s)?

Let us get some summary data. I will do it in two different ways. This is a convenient one, using a function from [RcmdrMisc](#):

```
numSummary(dmit$activ , groups = dmit$ftraining, statistics = c("mean", "sd"))

##           mean          sd data:n
## Morning  2.103000 0.8113702    11
## Lunch    2.198833 0.6608995    12
## Afternoon 3.797348 0.9013205    23
```

But of course, you can get a similar thing building what you want from scratch with, for instance, `aggregate`

```
with(dmit, aggregate(activ, list(Training = ftraining),
                        function(x) c(mean = mean(x),
                                       sd = sd(x),
                                       n = sum(!is.na(x)))
                        ))

##   Training    x.mean    x.sd    x.n
## 1  Morning  2.1030000 0.8113702 11.0000000
## 2   Lunch  2.1988333 0.6608995 12.0000000
## 3 Afternoon 3.7973478 0.9013205 23.0000000
```

or `by`:

```
with(dmit, by(activ, ftraining,
              function(x) c(mean = mean(x),
                             sd = sd(x),
                             n = sum(!is.na(x)))
              ))

## ftraining: Morning
##      mean          sd          n
## 2.1030000 0.8113702 11.0000000
```

Some basic statistics with R

```
## -----  
## ftraining: Lunch  
##      mean      sd      n  
## 2.1988333 0.6608995 12.0000000  
## -----  
## ftraining: Afternoon  
##      mean      sd      n  
## 3.7973478 0.9013205 23.0000000
```

You can get an idea: it seems that Morning and Lunch are very similar to each other, but Afternoon is very different. This agrees with the impression we got from the boxplot. But we would like a more formal procedure: we are going to compare all pairs of means and we will take into account that we are carrying out multiple comparisons (tests of pairs of means when the ANOVA is not significant are rarely justified¹⁷).

Comparing all pairs of means is done using the ANOVA model, so the results are not identical to comparing using t-tests (briefly: the estimate of the variance might be slightly different, and probably better).

Multiple testing corrections are needed because we are now conducting three separate tests (in general, if there are K groups and if you compare all pairs of means you carry out $\binom{K}{2} = \frac{K(K-1)}{2}$ tests). Here, we will control the family-wise error rate, the probability of falsely rejecting one or more tests over the family of tests performed—three in our case. The logic is somewhat like that of being struck by lightning: the chances of it happening are extremely small, but every year people die from lightning because the probability that at least one person is killed is huge since we have lots of people exposed to the risk. So even if all null hypotheses for our three tests are true:

- $\mu_{Morning} = \mu_{Lunch}$
- $\mu_{Morning} = \mu_{Afternoon}$
- $\mu_{Lunch} = \mu_{Afternoon}$

if we run the three comparisons, the chances of incorrectly rejecting at least one of the null hypotheses is larger than, say, 0.05 if we simply look at each one of the three p-values and keep any with a p-value ≤ 0.05 . You will see more about multiple testing below ([Multiple comparisons: FWER and FDR](#)).

Understanding what we just said is crucial. So we will use another two examples, a depressing and an uplifting one. If you understand them, you understand why we need multiple comparisons:

¹⁷Unless some specific, small number of, tests of specific pairs had been planned before the experiment.

Some basic statistics with R

- Russian roulette. (Important note: we of course strongly discourage playing Russian roulette!). Suppose you have two friends, “A” and “B”. “A” intends to play Russian roulette once a year, “B” intends to play it ten times a year¹⁸. Whose funeral are you likely to attend first?
- Lottery: we can give you either one lottery ticket or 20 lottery tickets (each with a different number). Assuming you have no problem with becoming rich, what option do you prefer?

Now, the following should be clear:

- If you test, say, 3 null hypotheses, and each one is true, and you reject any of the hypothesis at the 0.05 level, the probability that you will reject one or more null hypotheses when you shouldn't is larger than 0.05.
- If instead of testing 3 null hypotheses you test 20, the probability of rejecting one or more when you shouldn't is much larger than if you were just testing 3 null hypotheses.

If the above is still unclear, think about why it is more likely “B” will get killed first. Or why it is more likely you will become richer if we give you 20 different lottery tickets.

Back to our Anova.

So we will get both a plot and textual output¹⁹

```
library(multcomp) ## for glht

## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
##      geyser

## The next two lines carry out the multiple comparisons and the
## ones below plot them
```

¹⁸We say “intends to”: obviously, the first time that the gun fires is the last time the player plays the game.

¹⁹If you have used R Commander, you will see a lot of resemblance with the output produced by R Commander. I confess I cheated here; I got the commands below from R Commander, which I found very simple to do by fitting an ANOVA in the menu and then making sure to click on “Pair-wise comparisons of means”. However, that is just a detail that explains the names of the objects. The procedure using `glht` does not depend on R Commander.

Some basic statistics with R

```
Pairs <- glht(AnovaMIT, linfct = mcp(ftraining = "Tukey"))
summary(Pairs) # pairwise tests

##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = activ ~ ftraining, data = dmit)
##
## Linear Hypotheses:
##
##              Estimate Std. Error t value Pr(>|t|)
## Lunch - Morning == 0    0.09583    0.34434   0.278   0.958
## Afternoon - Morning == 0  1.69435    0.30240   5.603 <1e-05 ***
## Afternoon - Lunch == 0   1.59851    0.29375   5.442 <1e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

confint(Pairs) # confidence intervals

##
## Simultaneous Confidence Intervals
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = activ ~ ftraining, data = dmit)
##
## Quantile = 2.4235
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##
##              Estimate lwr      upr
## Lunch - Morning == 0    0.09583 -0.73866  0.93033
## Afternoon - Morning == 0  1.69435  0.96148  2.42722
## Afternoon - Lunch == 0   1.59851  0.88660  2.31043

cld(Pairs) # compact letter display
old.oma <- par(oma = c(0,5,0,0))
plot(confint(Pairs))
par(old.oma) ## restore graphics windows settings
```

Note that we can get some or most of that also from a call to `TukeyHSD`:

95% family-wise confidence level

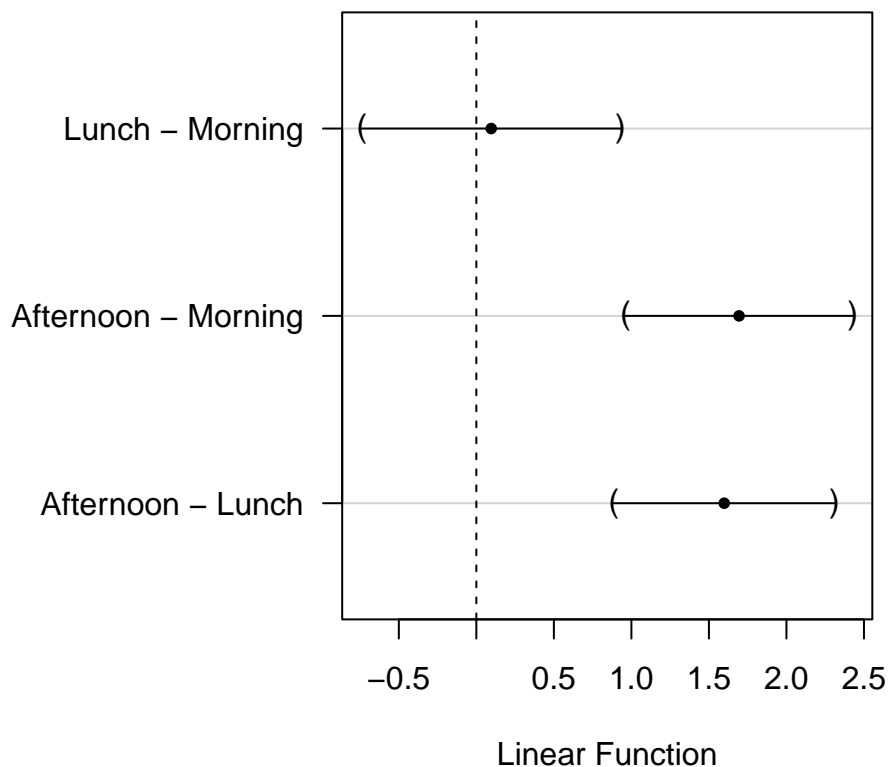


Figure 2 – Plot of pairwise differences with Tukey contrasts

```
TukeyHSD(aov(activ ~ ftraining, data = dmit))

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = activ ~ ftraining, data = dmit)
##
## $ftraining
##              diff          lwr          upr      p adj
## Lunch-Morning  0.09583333 -0.7400198  0.9316864  0.9582414
## Afternoon-Morning 1.69434783  0.9602867  2.4284089  0.0000041
## Afternoon-Lunch  1.59851449  0.8854440  2.3115850  0.0000070
```

More details are provided in section [Multiple comparisons of means in two-way ANOVA](#). A quick explanation of what is happening with the p-values in <https://stats.stackexchange.com/a/488655>.

Some basic statistics with R

Look carefully at the plot in Figure 2: for each difference (for each **contrast**), it shows the estimate and a 95% confidence interval around it. The plot title says “95% family-wise confidence interval”, and that indicates that multiple testing correction has been used. (You might want to make that even more explicit by using a title such as “95% family-wise confidence interval using Tukey contrasts”).

Given how far two of the contrasts are from 0.0, it seems those are highly significant differences.

Now, answer these questions (we will discuss them in class):

- If we had constructed **only** one of the confidence intervals (i.e., if we had not adjusted for multiple testing), that confidence interval would have been:
 - narrower
 - wider
- If we had constructed, and adjusted for multiple testing 10 confidence intervals instead of three they would have been
 - narrower
 - wider

Back to the numerical output.

The numerical output explicitly shows that we are using Tukey’s method and it shows the p-values of each contrast (each comparison), and it makes it clear that we are being reported adjusted p values. There is strong evidence of a difference between Afternoon and the other two levels, but no evidence of differences between Lunch and Morning.

18.5.1 And can I plot the means with s.e from the model?

Sure. A simple way of doing it is using the `allEffects` function from the *effects*:

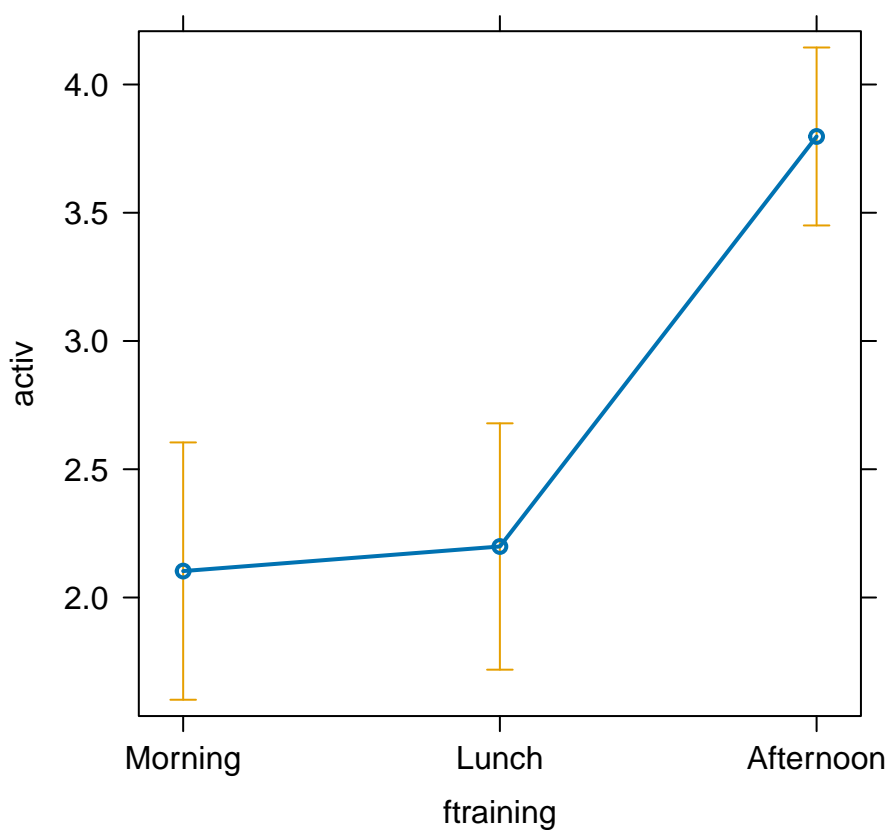
```
library(effects)

## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

Some basic statistics with R

```
plot(allEffects(AnovaMIT), ask = FALSE)
```

ftraining effect plot



That shows the estimates for each group and a 95% confidence interval (again, based on the whole ANOVA model). But from that figure it is not easy to tell which pairs differ, especially taking multiple comparisons into account.

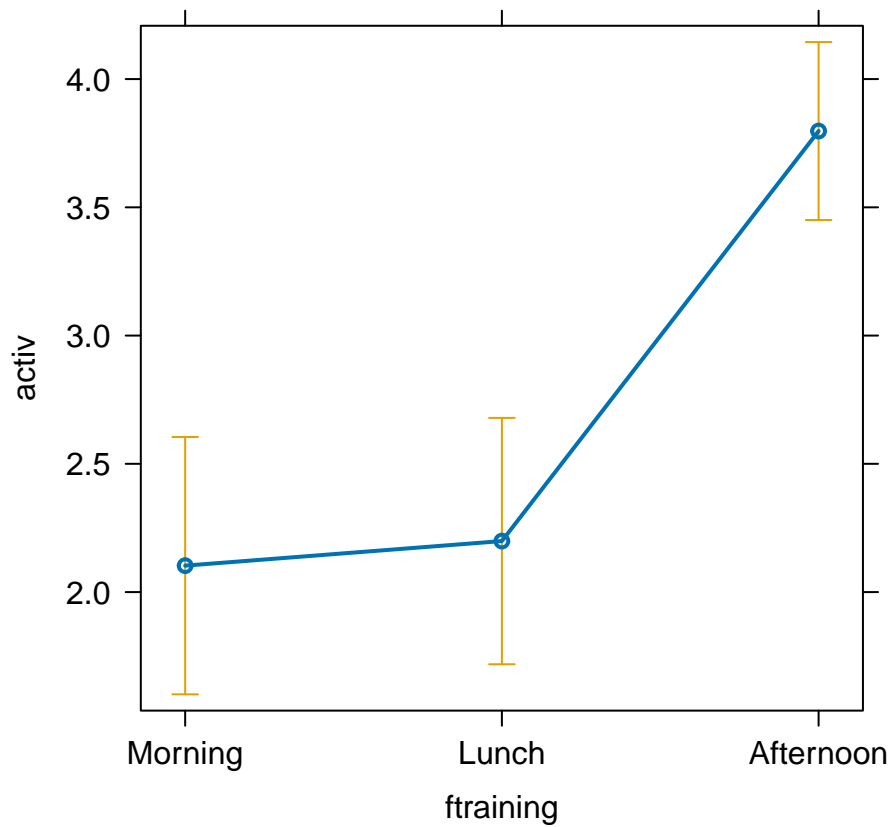
18.5.2 This is a mess. What figures do I use?

That is up to you :-). But this can work: present both the original means and the plot with the contrasts.

I would actually modify slightly the title of both figures, so that they look better:

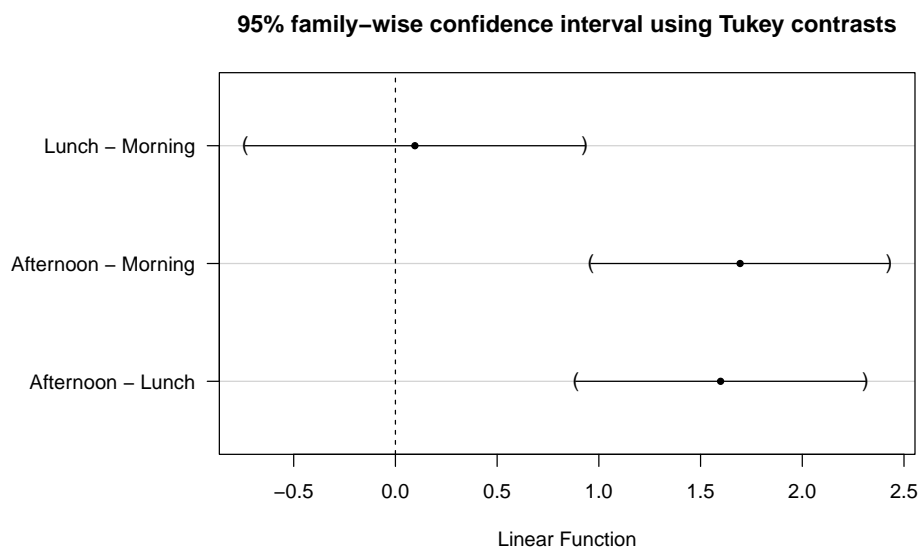
```
plot(allEffects(AnovaMIT), ask=FALSE, main = "Training: effect plot")
```

Training: effect plot



```
.Pairs <- glht(AnovaMIT, linfct = mcp(ftraining = "Tukey"))
tmp <- cld(.Pairs) ## silent assignment
old.oma <- par(oma=c(0,5,0,0))
plot(confint(.Pairs),
      main = "95% family-wise confidence interval using Tukey contrasts")
```

Some basic statistics with R



```
par(old.oma) ## restore graphics windows settings
```

18.5.3 Side note: Interpreting confidence intervals

If this is not obvious to you, ask it in class: a figure that shows an estimate (e.g., a mean) and a 95% confidence interval, where the interval goes from, say, 1 to 2, **should not** be interpreted as saying that there is a 95% probability that the mean is between 1 and 2. That is not the correct interpretation of a confidence interval. Make sure you understand this!!! (But we have discussed this already).

18.5.4 Multiple comparisons, other contrasts, etc

There is a wide literature on methods for adjusting for multiple comparisons in ANOVA and linear models. And sometimes a distinction is made between pre-planned and post-hoc comparisons. Tukey's approach is a widely accepted one (though there are others) and the distinction between pre-planned and post-hoc does not arise when researchers directly want to do all possible pairs right from the beginning. However, many of these issues can become important if you know, from the start, that some comparisons do not matter to you, and/or there are many groups. As well, we could be interested in other types of contrasts, for instance, that the mean of groups 2 and 3 is different from the mean of group 4, or comparing specific treatments against a control. Etc, etc. We will not get into this. A very good introduction, in addition to other references already mentioned, is chapter 3 of the on-line book "ANOVA and Mixed Models: A Short Intro Using R": <https://stat.ethz.ch/~meier/teaching/anova/contrasts-and-multiple-testing.html>.

Some basic statistics with R

18.6 t-test as ANOVA

Of course, in general, you can just carry out any two-group comparison as an ANOVA. There is nothing wrong with that (and there is a simple correspondence between a t statistic and an F statistic).

18.7 Several ways of obtaining summaries

For example with `aggregate`

```
with(dmit, aggregate(activ, list(Training = ftraining),
                        function(x) c(mean = mean(x),
                                      sd = sd(x),
                                      n = sum(!is.na(x)))
                        ))

##   Training      x.mean      x.sd      x.n
## 1  Morning  2.1030000  0.8113702 11.0000000
## 2   Lunch  2.1988333  0.6608995 12.0000000
## 3 Afternoon 3.7973478  0.9013205 23.0000000
```

or `by`:

```
with(dmit, by(activ, ftraining,
              function(x) c(mean = mean(x),
                            sd = sd(x),
                            n = sum(!is.na(x)))
              ))

## ftraining: Morning
##      mean      sd      n
## 2.1030000 0.8113702 11.0000000
## -----
## ftraining: Lunch
##      mean      sd      n
## 2.1988333 0.6608995 12.0000000
## -----
## ftraining: Afternoon
##      mean      sd      n
## 3.7973478 0.9013205 23.0000000
```

18.8 Can you do an ANOVA with only one sample per group?

Why or why not?

What happens here?

Some basic statistics with R

```
y <- c(1, 2, 3)
gr <- factor(c("g1", "g2", "g3"))
anova(lm(y ~ gr))

## Warning in anova.lm(lm(y ~ gr)): ANOVA F-tests on an essentially perfect
fit are unreliable

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value Pr(>F)
## gr         2      2      1      NaN    NaN
## Residuals  0      0      NaN
##
summary(aov(y ~ gr))

##           Df Sum Sq Mean Sq
## gr         2      2      1
```

And here?

```
y2 <- c(1, 2, 3, 4)
gr2 <- factor(c("g1", "g2", "g3", "g3"))
anova(lm(y2 ~ gr2))

## Analysis of Variance Table
##
## Response: y2
##           Df Sum Sq Mean Sq F value Pr(>F)
## gr2        2    4.5    2.25    4.5 0.3162
## Residuals  1    0.5    0.50
##
summary(aov(lm(y2 ~ gr2)))

##           Df Sum Sq Mean Sq F value Pr(>F)
## gr2        2    4.5    2.25    4.5 0.316
## Residuals  1    0.5    0.50
```

18.9 One way ANOVA: summary of steps

1. Enter the data.
2. Recode the factor (the independent variable), if needed.
3. Run the model.
4. Assess model diagnostics (see section [Diagnostics](#)).
5. Carry out comparisons between pairs of means with appropriate adjustment for multiple comparisons.

	Null hypothesis not rejected	Null hypothesis rejected
Means do not differ (H_0 true)	U	V
Means differ (H_0 false)	T	S

Table 3 – Multiple comparisons

In rows is the “truth” (how things really are), and in columns the output from our testing procedure (what we end up claiming or believing). The sum of all entries is the total number of comparisons made.

19 Multiple comparisons: FWER and FDR

19.1 Family-wise error rate

In section [So which means are different? Multiple comparisons](#) we covered multiple comparisons. Here we go into a little bit more detail before continuing with ANOVA. As in section [So which means are different? Multiple comparisons](#), suppose we are testing a number of null hypothesis. Table 3 shows a depiction of what we are concerned about, where the letters in each cell refer to the number of means tested that fall in each case.

In the example in section [So which means are different? Multiple comparisons](#) $U + V + T + S = 3$ (beware, 3 is the number of **hypothesis tests**, it is not the number of means; in our case, both are three, but Table 3 reflects number of hypothesis, not number of means). Procedures such as the one we used (Tukey) or Bonferroni or similar ones, try to control the probability that $V \geq 1$. They control what is called the “family-wise error rate” (FWER).

The intuitive idea is: “I want to control very tightly the probability of falsely rejecting any hypothesis”, and that is the same as saying “I want to control very tightly the probability that V is equal or larger than 1”. (I use the expression “control very tightly” because if we insist in “I NEVER want to falsely reject any null hypothesis” then ... we will never reject any null hypothesis). What Tukey, Bonferroni, and other procedures for controlling the family wise error rate provide are mechanisms for ensuring that $Pr(V \geq 1)$ is below a number you specify (e.g., 0.05).

Note that, in our usage of Tukey, we did not pre-specify the actual $Pr(V \geq 1)$. The procedure is run, and it gives us “adjusted p-values”. And what is an adjusted p-value? The classical paper from Wright, 1992, “Adjusted p-values for simultaneous inference”, *Biometrics*, 48: 1005–1013, has in p. 1006 this definition of adjusted p-value: “The adjusted P -value for a particular hypothesis within a collection of hypotheses, then, is the smallest overall (i.e., ‘experimentwise’) significance level at which the particular hypothesis would be rejected.” Read it very carefully. To get going, start reading the sentence covering “adjusted” and “collection of” (i.e., think about a single p-value for an experiment where a single null is tested); then move on to the multiple p-values case. It makes comparisons very simple; as Wright explains: “An adjusted P -value can be compared directly with any chosen significance level α :

Some basic statistics with R

If the adjusted P -value is less than or equal to α , the hypothesis is rejected.” (If you find this too advanced for now, don’t worry. But remember this once you feel more comfortable with these issues. And no, this paragraph will not be in the exam.)

19.2 False discovery rate (FDR)

There is a different approach to the multiple testing problem. In this approach we focus on controlling the fraction of false positives. The total number of null hypothesis we reject is $V + S$. The intuitive idea behind the control of the false discovery rate (**FDR**) is to bound (to set an upper limit to) the ratio $\frac{V}{V+S}$ ²⁰.

One key difference is that the FDR can be kept reasonably low (say, 0.01) even when it is almost sure that $V \geq 1$. When could this happen? For instance, when we are conducting tens of thousands of hypothesis tests. Again, the FDR will control the fraction of false discoveries whereas the control of the family wise error rate (FWER) is emphasizing that V don’t become 1 or more.

As we did with Tukey and the FWER procedures, we generally do not pre-specify the level of FDR we want to attain but, rather, we obtain “adjusted p-values”. The difference in the meaning of “adjusted” is that now these p-values are adjusted for FDR (not adjusted for control of the family wise error rate). So, when we deal with FDR, the adjusted p-value of an individual hypothesis is the lowest level of FDR for which the hypothesis is first included in the set of rejected hypotheses (e.g., Reiner et al., 2003, *Bioinformatics*)²¹.

The FDR is usually employed in screening procedures, where we are willing to allow some false discoveries, because we are screening over thousands of hypothesis. The cost of requiring $V = 0$ would be to miss many discoveries. One example? Suppose that you have measured the expression of 20000 genes in two sets of subjects some with colon cancer and some without. Now, you can do the equivalent of 20000 t-tests. So you will get 20000 p-values, and you will want to adjust those 20000 tests for multiple testing.

How do you adjust for multiple testing in R? This is easily done with the function `p.adjust`. When you are applying FDR you often have a collection of p-values already.

I will make a simple example up and will only use four p-values (not 20000) for the sake of simplicity. Suppose we have done a screening procedure, testing four genes. You get the p-values I show below. To use an FDR correction method I use `p.adjust` with the `method = “BH”` argument (BH is one of several possible types of FDR correction). To show what happens, I have then combined the two, side by side, so you can see the original p-value and the FDR-adjusted one.

²⁰There are several different approaches. The most common one is to control $FDR = E(Q)$ where $Q = V/(V + S)$ if $V + S > 0$ (and $Q = 0$ otherwise). But there are others, such as the $pFDR$, etc.

²¹Reiner et al. (2003). Identifying differentially expressed genes using false discovery rate controlling procedures. *Bioinformatics*, 19(3), 368–375. <https://doi.org/10.1093/bioinformatics/btf877>

Some basic statistics with R

```
p.values <- c(0.001, 0.01, 0.03, 0.05)
adjusted.p.values <- p.adjust(p.values, method = "BH")
cbind(p.values, adjusted.p.values)

##      p.values adjusted.p.values
## [1,]    0.001             0.004
## [2,]    0.010             0.020
## [3,]    0.030             0.040
## [4,]    0.050             0.050
```

How do we interpret this? Here I will only cover the very basics. But go back a couple of paragraphs, and re-read the definition of adjusted p-value for the FDR. So, for example, if we keep as “significant” all the genes with a p-value (not adjusted p-value, but p-value, so the last first three) ≤ 0.030 , the FDR (the expected number of false discoveries) will be 0.040 (the FDR-adjusted p-value for the gene with p -value of 0.03).

Note that the FDR applies not just to comparisons between means or t-tests, but to any kind of test (comparing variances, correlations, etc).

19.3 Multiple comparisons: struck by lightning, roulette, and lottery

This has been a short section, because we are skipping the technicalities and focus just on the big ideas. But this is a **VERY IMPORTANT** section to remember. When you do many tests, some of them might have low p-values just by chance and you need to adjust for this. If any gene with a low p-value is declared significant (regardless of the size of the collection of tests) you will be likely to start claiming that many purely chance results are “significant”. And you do not want that. Again, go back to the examples of the lottery and the Russian roulette if this is not completely obvious.

Remember that very rare events do happen, and they are almost certain to happen if the experiment is repeated many times (by the way, this is why most of us are not afraid of dying from lightning, even if every year some people do in fact die from lightning).

When you screen 20000 genes, you are running 20000 times the experiment of the p-value and the null hypothesis. And remember the rules: for one true null hypothesis, the probability of finding a p-value ≤ 0.05 is 0.05. Now imagine you do that 20000 times; you are almost certain to have many p-values ≤ 0.05 . (Same thing with lightning: even if the chances of dying from lightning are $\leq \frac{1}{300000}$, with millions of people on earth, some are almost sure to die from lightning).

There are many reviews about multiple testing, FDR, etc. You might want to take a look at a three-page one by W. Noble, in *Nature Biotechnology*, 2009, 27: 1135–1136, “How does multiple testing work”.

20 Two-way ANOVA

A key issue in models with two or more predictor variables (such as two-way ANOVA) are interactions. What is an interaction? The phenomenon of interaction should be familiar to you: it is very common in life in general, and in biology you might have previously seen it as epistasis in genetics; or take a look at the 'Interactions' section in the leaflet (prospectus) of any drug you take.

Before moving on, make sure you understand what interaction means. Interaction is also often referred to as non-additivity (and this will become clearer later).

20.1 A very simple two-way ANOVA

Let us create some fake data

```
set.seed(3)
df1 <- data.frame(y = runif(8),
                  A = rep(c("a1", "a2"), 4),
                  B = rep(c("b1", "b1", "b2", "b2"), 2))
```

```
df1

##           y  A  B
## 1 0.1680415 a1 b1
## 2 0.8075164 a2 b1
## 3 0.3849424 a1 b2
## 4 0.3277343 a2 b2
## 5 0.6021007 a1 b1
## 6 0.6043941 a2 b1
## 7 0.1246334 a1 b2
## 8 0.2946009 a2 b2
```

Some summaries of data:

```
(means <- with(df1, tapply(y, list(A, B), mean)))

##           b1           b2
## a1 0.3850711 0.2547879
## a2 0.7059552 0.3111676
```

And now, several ANOVA models. We will explain each of the terms in turn in class if this is not familiar to you:

20.1.1 No interaction model

Some basic statistics with R

```
m1 <- lm(y ~ A + B, data = df1)
anova(m1)

## Analysis of Variance Table
##
## Response: y
##          Df    Sum Sq  Mean Sq F value Pr(>F)
## A          1 0.071164  0.071164   1.9312 0.2233
## B          1 0.137850  0.137850   3.7410 0.1109
## Residuals  5 0.184244  0.036849
```

The above is the anova table; later, we will say many more things about it.

But for now notice also this output, that gives the estimated coefficients:

```
summary(m1)

##
## Call:
## lm(formula = y ~ A + B, data = df1)
##
## Residuals:
##      1      2      3      4      5      6      7      8
## -0.28316  0.16769  0.19628 -0.04956  0.15090 -0.03544 -0.06403 -0.08269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4512     0.1176   3.838  0.0121 *
## Aa2           0.1886     0.1357   1.390  0.2233
## Bb2          -0.2625     0.1357  -1.934  0.1109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.192 on 5 degrees of freedom
## Multiple R-squared:  0.5315, Adjusted R-squared:  0.3441
## F-statistic: 2.836 on 2 and 5 DF,  p-value: 0.1502
```

We will explain it in more detail later ([Anova tables from `lm` et al.: understanding the coefficients and parameters](#)), but for now remember we are using an additive model, so here we are estimating only three things from a four-means design. What three things?

We will not emphasize understanding the coefficients in this class; we just wanted to show them to you, so you can see we are actually estimating three things. More details in [Anova tables from `lm` et al.: understanding the coefficients and parameters](#), but those details **are not** needed.

Some basic statistics with R

20.1.2 Interaction model

Now all cell means are modeled:

```
m2 <- lm(y ~ A * B, data = df1)
anova(m2)

## Analysis of Variance Table
##
## Response: y
##          Df    Sum Sq  Mean Sq F value Pr(>F)
## A          1 0.071164  0.071164   1.9071 0.2394
## B          1 0.137850  0.137850   3.6942 0.1270
## A:B        1 0.034981  0.034981   0.9374 0.3878
## Residuals  4 0.149262  0.037316
```

In this case, is there evidence of interaction?

Again, we could ask for the estimated coefficients:

```
summary(m2)

##
## Call:
## lm(formula = y ~ A * B, data = df1)
##
## Residuals:
##      1      2      3      4      5      6      7      8
## -0.21703  0.10156  0.13015  0.01657  0.21703 -0.10156 -0.13015 -0.01657
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.3851     0.1366   2.819  0.0479 *
## Aa2            0.3209     0.1932   1.661  0.1720
## Bb2           -0.1303     0.1932  -0.674  0.5370
## Aa2:Bb2       -0.2645     0.2732  -0.968  0.3878
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1932 on 4 degrees of freedom
## Multiple R-squared:  0.6204, Adjusted R-squared:  0.3358
## F-statistic: 2.18 on 3 and 4 DF, p-value: 0.233
```

Check the interpretation of the interaction coefficient:

```
means[2, 2] -
  (means[1, 1] + coefficients(m2)[2] + coefficients(m2)[3])

##          Aa2
```

Some basic statistics with R

```
## -0.2645044
```

So: how are we estimating the coefficient for interaction? Again: we will not emphasize understanding the coefficients in this class; we just wanted to show them to you, so you can see we are actually estimating four things (in the previous example, we were estimating three) and how we measure the interaction. More details in [Anova tables from `lm` et al.: understanding the coefficients and parameters](#) and in [A two-way ANOVA with interaction: SS, coefficients](#) but those details **are not** needed.

Before we continue, however, let's do the following thought experiment. Suppose you have a two-way ANOVA. The first factor has four levels, the second factor has seven levels.

- How many coefficients will you be estimating in a model without interactions?
- How many coefficients will you be estimating in a model with interactions?

And could you think about how you would estimate the coefficients for the interactions? How many of those coefficient would there be? How are the number of coefficients related to the degrees of freedom for the interaction? (Again, further details in [A two-way ANOVA with interaction: SS, coefficients](#)).

20.1.3 Parameters and degrees of freedom

It should be clear how many parameters we are fitting and, thus, how many degrees of freedom we are using:

- One-way ANOVA
- Two-way ANOVA, with and without interactions
- Three-way ANOVA, with and without interactions

20.2 Loading the cholesterol data set

The following data come from an experiment about the effects of three diets and two cholesterol-controlling drugs in the reduction of cholesterol levels (note: the response variable is change in cholesterol, so the larger the value, the larger the reduction of cholesterol). As usual, read the data and look at them. Since the author used names for the levels within each of the two factors, Diet and Drug, we do not need to transform them into factors, in contrast to what we did in section [Recoding variables](#). Call the data `dcholest`.

(Note: since a few versions ago, R no longer converts characters to factors when reading with `read.table`. In some places that is inconsequential, in other places we will have to use factors. We will deal with it.)

```
dcholest <- read.table("Cholesterol.txt", header = TRUE)
```

Look at the output, which I comment below

Some basic statistics with R

```
## This fits the model. Pay attention to the "*"
cholestanova <- (lm(y ~ Diet*Drug, data=dcholest))
## This shows the ANOVA table. Notice the "Type II"
## And notice we are using function Anova with capital A
## which is a function from the car package.
Anova(cholestanova)

## Anova Table (Type II tests)
##
## Response: y
##           Sum Sq Df F value    Pr(>F)
## Diet       75.453  2   29.949 3.163e-08 ***
## Drug       32.261  1   25.610 1.433e-05 ***
## Diet:Drug  48.979  2   19.441 2.348e-06 ***
## Residuals 42.830 34
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Now we are shown the 3 by 2 table of means, standard deviations, and number
## of observations
tapply(dcholest$y, list(Diet=dcholest$Diet, Drug=dcholest$Drug),
       mean, na.rm=TRUE) # means

##      Drug
## Diet      A      B
##  HF 1.7280000 -0.588400
##  M1 0.7914286  4.055714
##  M2 2.5685556  5.318250

tapply(dcholest$y, list(Diet=dcholest$Diet, Drug=dcholest$Drug),
       sd, na.rm=TRUE) # std. deviations

##      Drug
## Diet      A      B
##  HF 0.5026165 0.4956474
##  M1 0.9572549 0.7641736
##  M2 1.5181591 1.3963718

tapply(dcholest$y, list(Diet=dcholest$Diet, Drug=dcholest$Drug),
       function(x) sum(!is.na(x))) # counts

##      Drug
## Diet A B
##  HF 4 5
##  M1 7 7
##  M2 9 8
```

Some basic statistics with R

20.2.1 `Anova`, `anova`, `aov`, `lm`, `summary`: what gives?

In R we often can use different ways of getting output from a linear model, including regression and ANOVA. This section briefly discusses issues specific to R. Most of this should be clear by the end of the course. These are they key messages. Refer to this section as needed while you use these notes.

- Function `lm` fits linear models, including regression and ANOVA's (most of them, not all).
- Function `aov` can also be used to fit ANOVA models. We do not use it to fit regression models. And most (not all, but not relevant for this course) of the models you can fit with `aov` you can fit with `lm`. As far as this course is concerned, their syntax is the same.
- `Anova` and `anova` give you ANOVA tables from models fitted with `lm`. The main difference between the two is that `Anova` gives, by default, what are called Type II sums of squares and tests, which we will use most of the time to deal with issues about order of factors. This will all be explained in detail.
- Function `anova` can also be used to compare models. We will see examples.
- Function `summary` on an object fitted with `aov` will also give you an ANOVA table. We will rarely use this (though it might be the code that some menus in R commander actually generate, and you might see it in other people's code).
- Function `summary` on an object fitted with `lm` will give, among others, a table of coefficients, not an ANOVA table.

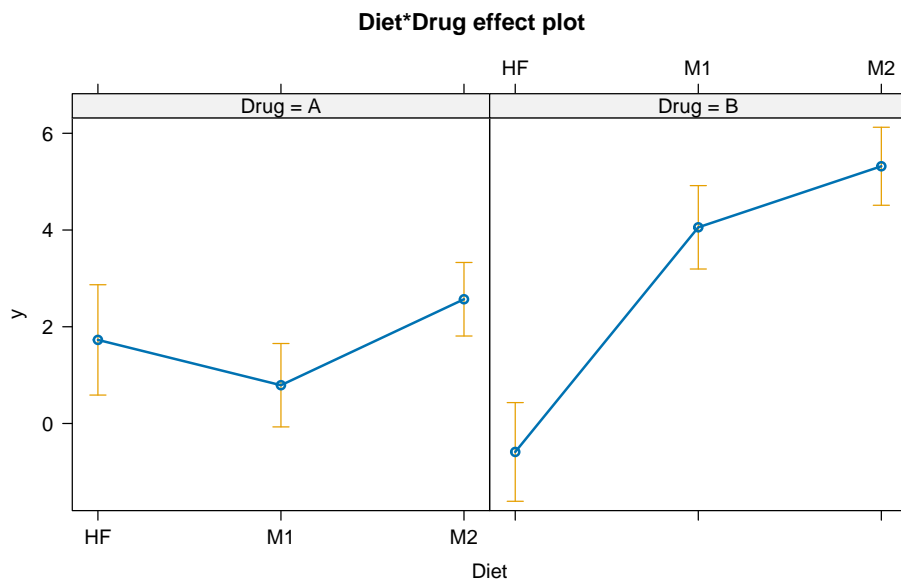
Yes, this is a lot of terminology (jargon). And yes, it is potentially confusing. This should not be, though, an issue for understanding the conceptual and statistical issues discussed.

20.3 Interactions

The Anova table for the cholesterol data shows very strong effects of interactions. Make sure you understand why we say this. We now want to understand those interactions. We will use an “Effects plot” to look at interactions:

```
plot(allEffects(cholestanova), ask = FALSE)
```

Some basic statistics with R



What do you see? Do you see the interaction in the plot? Basically, an interaction means that the effect of one variable depends on the effect of the other. In this case, even if Drug B overall leads to a larger change (decrease) in cholesterol, its effects depend on the Diet. This has practical consequences: is Drug B a better drug? It depends on the diet of the patient: for the HF (high fat) diet, Drug B is clearly worse than Drug A.

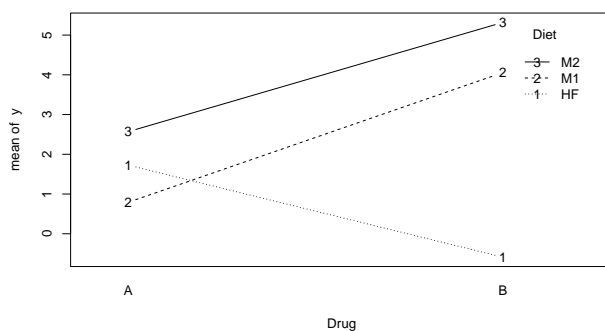
Interaction is also called “non-additivity” because the model deviates from a simple model like

$$y = \text{Drug} + \text{Diet}$$

as the effect of Drug depends on the value of Diet (or the other way around). As we have said, the phenomenon of interaction should be familiar to you: it is very common in life in general, and in biology you might have previously seen it as epistasis in genetics.

You can also see interaction plots using other functions from R. For example:

```
with(dcholest, interaction.plot(Drug, Diet, y, type = "b"))
```



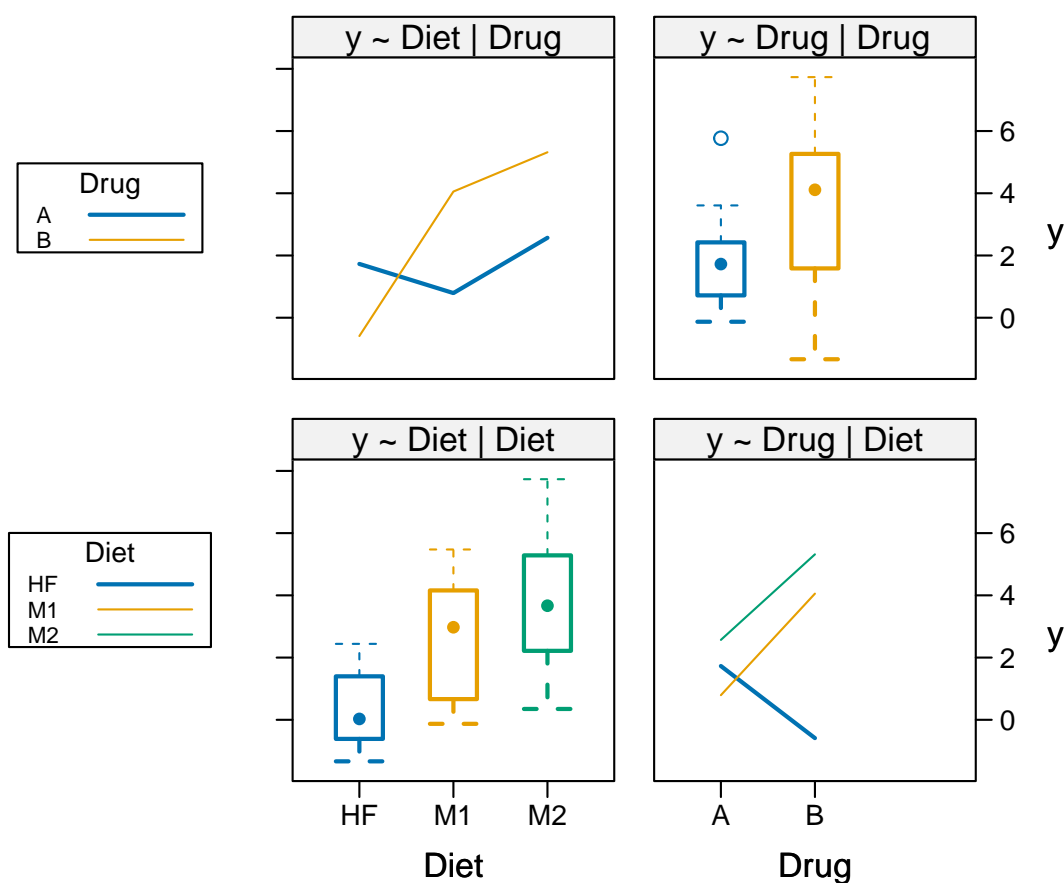
or using the [HH](#) package

Some basic statistics with R

```
library(HH)

## Loading required package: lattice
## Loading required package: grid
## Loading required package: latticeExtra
##
## Attaching package: 'latticeExtra'
## The following object is masked from 'package:ggplot2':
##
##     layer
## Loading required package: gridExtra
##
## Attaching package: 'HH'
## The following objects are masked from 'package:car':
##
##     logit, vif
## The following object is masked from 'package:BiocStyle':
##
##     latex
## The following object is masked from 'package:base':
##
##     is.R
interaction2wt(y ~ Diet + Drug, data = dcholest)
```

y: main effects and 2-way interactions

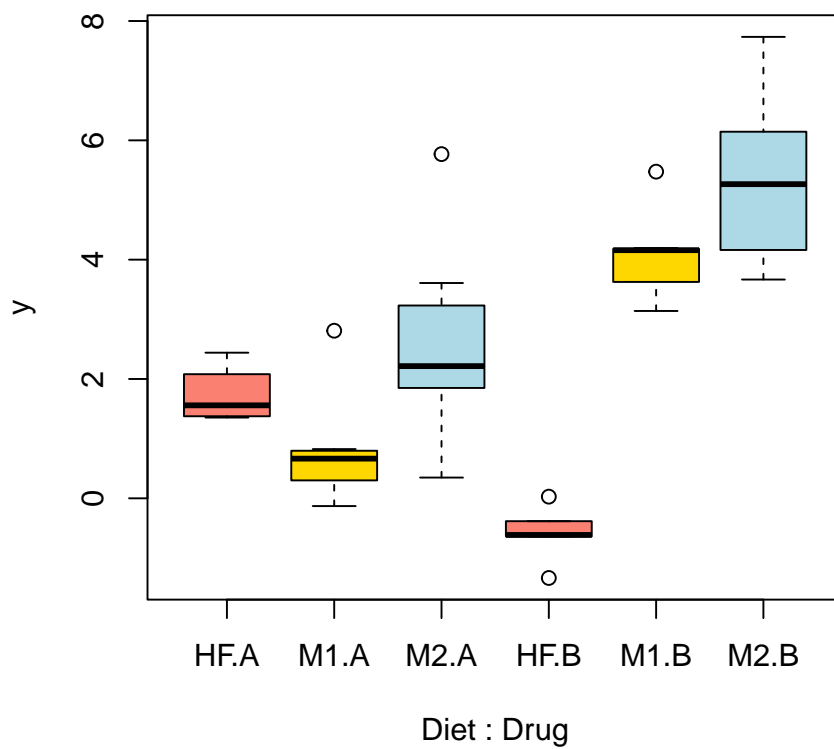
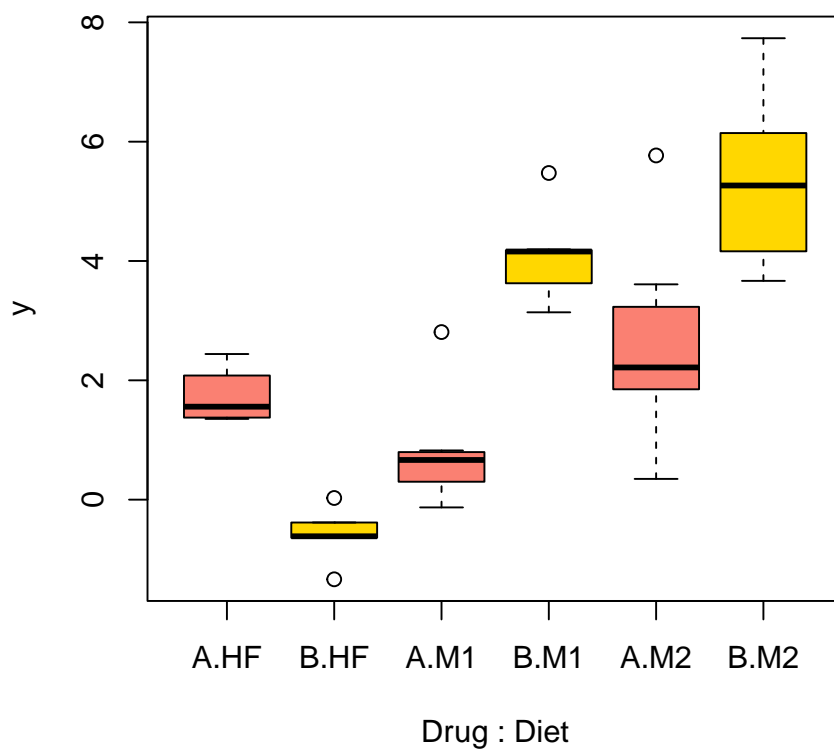


Notice how this last figure displays both main effects and interactions. So even if the main effect of Drug B is to lead to a larger change in cholesterol (as you can see in the upper right panel), Drug B actually leads to much smaller change in cholesterol if given to patients with diet HF (as seen in the bottom right panel). In fact, under Drug A, it seems that diet HF is actually slightly better than diet M1 (as seen in the bottom right panel or in the effects plot).

Finally, a boxplot can also help show the interaction. I will use two different ones, that differ by the order in which factors are specified (one or the other might be easier to decode visually):

```
par(mfrow = c(2, 1))
boxplot(y ~ Drug * Diet, data = dcholest, col = c("salmon", "gold"))
boxplot(y ~ Diet * Drug, data = dcholest,
        col = c("salmon", "gold", "lightblue"))
```

Some basic statistics with R



Some basic statistics with R

Given these results (the strong interaction, that can even revert effects of one factor), it makes little sense to report any global main effects and we would rarely be interested in interpreting the significance (or not) of the Diet or Drug term. In general, **in the presence of interactions, we often refrain from interpreting main effects; this is one consequence of what is often referred to as the “marginality principle”²²**. We return to this later in *“ANOVA/linear models with more than two factors” (section 20.10)*.

20.4 An ANOVA without interactions

Could we fit a model without interactions? Yes, of course. The idea is to change the “*” by a “+” (and we will see that again when we deal with multiple regression et al. in section *Interactions between continuous variables*).

```
amodelnoint <- (lm(y ~ Diet + Drug, data=dcholest))
Anova(amodelnoint)

## Anova Table (Type II tests)
##
## Response: y
##           Sum Sq Df F value    Pr(>F)
## Diet       75.453  2  14.793 2.046e-05 ***
## Drug       32.261  1  12.650 0.001074 **
## Residuals  91.809 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There are, however, good reasons to start fitting a model **with** interactions first, and **only** if there are no interactions, fit a simpler, additive model.

20.5 Getting ready for how things change with the order of factor

The next section introduces a common phenomenon that is sometimes surprising. We will try to provide some intuition about why this phenomenon happens.

These are the key steps of the argument (take a piece of paper, and draw a bunch of two-by-two boxes with the sample sizes in each cell; **really, do this now**):

1. Suppose you do a two-way ANOVA where the dependent variable (Y) is “awake in class” and your predictor variables are sex (female or male) and coffee in the morning (yes or no).

²²Properly formulated, which also generally involves using other types of contrasts —such as `contr.sum`, in R parlance— marginal tests in the presence of interactions, what are called Type III, can make sense, but are not always of interest. See also footnote 23 in section *The order of factors* for some entries and references.

Some basic statistics with R

2. Suppose you have, in the sample, 10 women, all of whom drank coffee, and 10 men, none of whom drank coffee. Can you say anything about the effect of sex that is not saying something (or even everything) about the effect of coffee?
3. Now, suppose instead that the design is perfectly balanced: 5 women who drank coffee, 5 women who did not drink coffee, 5 men who drank coffee, 5 men who did not drink coffee. If I told you “I measured a woman”, do you know if she is also a coffee drinker or not?
4. Now, think about intermediate scenarios.
5. Was the above a silly example? OK, replace the Y by “cardiovascular disease” and the predictor variables by “smoking” (yes and no) and “exercise” (yes and no). Can you say anything about the effects of exercise if all the people in your sample who exercise are also non-smokers? (Repeat this with other pairs of variables, such as diet and exercise, gene expression and age, gene expression and sex, etc, etc).

Some of this might still seem unclear after the above exercise. Think about doing a regression of body height on the length of both the left and right arms (we will actually do something very similar later: *“Multiple regression example” (section 22.2)*). And think about how unbalanced data, with categorical predictor variables, is somewhat similar to inducing a correlation between variables. We will discuss this in class.

Anyway, let's go and see this happening!

20.6 The order of factors

Let's pretend there are no interactions. We can do that by creating a data set without the “HF” subjects.

```
dcholest2 <- subset(dcholest, subset = Diet != "HF")
```

Now do a two-way ANOVA:

```
cholest2anova <- (lm(y ~ Diet*Drug, data = dcholest2))
Anova(cholest2anova)

## Anova Table (Type II tests)
##
## Response: y
##          Sum Sq Df F value    Pr(>F)
## Diet       17.879  1 11.7483  0.001967 **
## Drug       68.809  1 45.2150 3.216e-07 ***
## Diet:Drug   0.507  1  0.3335  0.568417
## Residuals 41.089 27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Some basic statistics with R

So no evidence whatsoever of interactions. For simplicity, we can go and refit the model without the interaction. We will actually fit two models, which differ only by the order in which we give Diet and Drug in the formula and we will call them lm1 and lm2

```
lm1 <- lm(y ~ Diet + Drug, data = dcholest2)
lm2 <- lm(y ~ Drug + Diet, data = dcholest2)
```

Look at the ANOVA tables:

```
anova(lm1)

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Diet         1 15.897   15.897   10.701 0.002842 **
## Drug         1 68.809   68.809   46.318 2.156e-07 ***
## Residuals    28 41.597    1.486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(lm2)

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Drug         1 66.827   66.827   44.983 2.793e-07 ***
## Diet         1 17.879   17.879   12.035 0.001708 **
## Residuals    28 41.597    1.486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As you can see, the F statistic and the p-value are different!!! What gives here?

Now, use Type II sums of squares:

```
Anova(lm1)

## Anova Table (Type II tests)
##
## Response: y
##           Sum Sq Df F value    Pr(>F)
## Diet         17.879  1  12.035 0.001708 **
## Drug         68.809  1  46.318 2.156e-07 ***
## Residuals    41.597 28
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Some basic statistics with R

```
Anova(lm2)

## Anova Table (Type II tests)
##
## Response: y
##          Sum Sq Df F value    Pr(>F)
## Drug       68.809  1  46.318 2.156e-07 ***
## Diet       17.879  1  12.035 0.001708 **
## Residuals  41.597 28
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nothing changes between those two. But if you look carefully, the F value (and p-value) of the Type II Sums of Squares ANOVA table are the same as those for the term that enters last in the Type I (those produced via `anova`, without a capital “A”).

By the way, R might tell you something might be going on in here if you look carefully:

```
aov1 <- aov(y ~ Diet + Drug, data=dcholest2)
aov1 ## Notice the "Estimated effects may be unbalanced"

## Call:
## aov(formula = y ~ Diet + Drug, data = dcholest2)
##
## Terms:
##          Diet      Drug Residuals
## Sum of Squares 15.89687 68.80930  41.59668
## Deg. of Freedom      1        1      28
##
## Residual standard error: 1.21885
## Estimated effects may be unbalanced
```

This is an **extremely common phenomenon** when the design is not perfectly balanced (with categorical independent variables) or there are correlations (with continuous covariates, as in regression). What is happening?

- Type II sums of squares (similar to t-statistics from a linear model) show what that term contributes, **given all the rest** are already in the model (“given all the rest”: all the rest that do not include this term, so no interactions with this term). In other words, given all the other terms (that do not include this term) have already been taken into account. This is actually the output we would get from comparing two models, one with all terms, and one with all terms except the term in question. (Always assuming interactions with the term in question are zero). The package `car`, by default, gives you this via `Anova`. I routinely use `Anova`.
- Type I (or sequential) sums of squares do not. They are sequential, in the order shown in the output. R, by default, gives you this via `anova`.

Some basic statistics with R

- (More details about Type I and Type II Sums of Squares are provided in the Appendix in “*Type I, Type II, Type III: a few technical notes*” (section 20.6.1), but you can skip it. This issue is also discussed, by focusing on the key issues from an applied perspective, in chapter 5 of Fox and Weisberg’s “An R companion to applied regression, 3rd edition”.)

And this, of course, affects models with two, three, . . . factors. Always pay attention to what it is you are being reported (and beware that the defaults used by R need not the same as those used by SPSS, SAS, etc.)²³.

Some of this might still seem mysterious. Think about doing a regression of body height on the length of both the left and right arms. And think about how unbalanced data, with categorical independent variables, is somewhat similar to inducing a correlation between variables. We will discuss this in class.

20.6.1 Type I, Type II, Type III: a few technical notes

(Skip this if you want. I leave it here in case you want to get back to it. And yes, we have not mentioned Type III much (but see [Interactions](#)).

I take this from Nancy Reid’s <http://www.utstat.utoronto.ca/reid/sta442f/2009/typeSS.pdf>:

“Let $R(\cdot)$ represent the residual sum of squares for a model, so for example $R(A,B,AB)$ is the residual sum of squares fitting the whole model, $R(A)$ is the residual sum of squares fitting just the main effect of A , and $R(1)$ is the residual sum of squares fitting just the mean.”

Type I

$$A: SS(A) = R(1) - R(A)$$

$$B: SS(B|A) = R(A) - R(A, B)$$

$$AB: SS(AB|A, B) = R(A, B) - R(A, B, AB)$$

Type II

$$A: SS(A|B) = R(B) - R(A, B)$$

$$B: SS(B|A) = R(A) - R(A, B)$$

$$AB: SS(AB|A, B) = R(A, B) - R(A, B, AB)$$

We assume no significant interaction

Type III

²³More details about Type I and Type II Sums of Squares are provided in the Appendix in “*Type I, Type II, Type III: a few technical notes*” (section 20.6.1), but you can skip it. This issue is also discussed, by focusing on the key issues from an applied perspective, in chapter 5 of Fox and Weisberg’s “An R companion to applied regression, 3rd edition” and Fox’s “Applied regression analysis and generalized linear models”. An email-length discussion of these topics can be found here <https://stat.ethz.ch/pipermail/r-help/2006-August/111854.html> and <https://stat.ethz.ch/pipermail/r-help/2006-August/111927.html>.

Some basic statistics with R

A: $SS(A|B, AB) = R(B, AB) - R(A, B, AB)$

B: $SS(B|A, AB) = R(A, AB) - R(A, B, AB)$

AB: $SS(AB|A, B) = R(A, B) - R(A, B, AB)$

Interaction need not be absent. But this does not mean these SS are interpretable. (And beware of the contrasts you use —section [Other contrasts](#)— if you want to take a look at coefficients. Note also possible issues related to changes in results if you center the predictors in models with continuous predictors). We will not use Type III in this notes. Why? Read the paper by Bill Venables “Exegeses on linear models”, or Oyvind Langsrud, 2003, “ANOVA for unbalanced data: Use Type II instead of Type III sums of squares”, *Statistics and Computing*, Volume 13, Number 2, pp. 163-167, or the many debates about Type III SS, or chapter 5 of Fox and Weisberg’s “An R companion to applied regression”, as well as Fox’s book on linear models (mentioned in Fox and Weisberg). There are a few ways to obtain Type III in R, the easiest by using `Anova(x, type = “III”)`, and also by using `drop1` with the right contrasts).

And some numbers here:

```
m_diet <- lm(y ~ Diet, data = dcholest)
m_drug <- lm(y ~ Drug, data = dcholest)
m_diet_drug <- lm(y ~ Diet + Drug, data = dcholest)
m_int <- lm(y ~ Diet * Drug, data = dcholest)

summary(m_diet)

##
## Call:
## lm(formula = y ~ Diet, data = dcholest)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.514 -1.648  0.003  1.454  3.872
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4411     0.6104   0.723   0.4744
## DietM1        1.9825     0.7824   2.534   0.0156 *
## DietM2        3.4214     0.7549   4.532 5.92e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.831 on 37 degrees of freedom
## Multiple R-squared:  0.359, Adjusted R-squared:  0.3243
## F-statistic: 10.36 on 2 and 37 DF, p-value: 0.0002675

anova(m_diet)
```

Some basic statistics with R

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Diet        2  69.476   34.738    10.36 0.0002675 ***
## Residuals   37 124.070    3.353
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m_drug)

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Drug        1  26.285   26.285    5.9716 0.01929 *
## Residuals   38 167.262    4.4016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m_diet_drug)

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Diet        2  69.476   34.738   13.621 3.938e-05 ***
## Drug        1  32.261   32.261   12.650 0.001074 **
## Residuals   36  91.809    2.550
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m_diet, m_diet_drug)

## Analysis of Variance Table
##
## Model 1: y ~ Diet
## Model 2: y ~ Diet + Drug
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1      37 124.070
## 2      36  91.809  1    32.261 12.65 0.001074 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## 124.1: RSS from model m_diet
## 91.8 : RSS from model m_diet_drug
## 32.3 = 124.1 - 91.8:
## SS(drug|diet) = R(m_diet) - R(m_diet_drug)
```

Some basic statistics with R

```
anova(m_drug, m_diet_drug)

## Analysis of Variance Table
##
## Model 1: y ~ Drug
## Model 2: y ~ Diet + Drug
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1      38 167.262
## 2      36  91.809  2    75.453 14.793 2.046e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##  $SS(diet|drug) = R(m\_drug) - R(m\_diet\_drug)$ 
##  $75.5 = 167.3 - 91.8$ 
Anova(m_diet_drug)

## Anova Table (Type II tests)
##
## Response: y
##           Sum Sq Df F value    Pr(>F)
## Diet       75.453  2  14.793 2.046e-05 ***
## Drug       32.261  1  12.650  0.001074 **
## Residuals  91.809 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## But also the sum SS here for drug and diet
Anova(m_int)

## Anova Table (Type II tests)
##
## Response: y
##           Sum Sq Df F value    Pr(>F)
## Diet       75.453  2  29.949 3.163e-08 ***
## Drug       32.261  1  25.610 1.433e-05 ***
## Diet:Drug  48.979  2  19.441 2.348e-06 ***
## Residuals  42.830 34
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(You can also take a look at http://md.psych.bio.uni-goettingen.de/mv/unit/lm_cat/lm_cat_unbal_ss_explained.html: that uses a slightly different notation.)

Some basic statistics with R

20.7 Does order always matter?

Nope. When the design is balanced, order does not matter²⁴.

This is slightly more advanced material. Skip it on first reading. If you want a very simple message: **unless you know what your are doing and/or you know your data fulfills certain properties, always expect order to matter.**

If you want to continue reading, let's proceed with some details then. The following is an example. For the sake of the exposition, I will here simulate the data, so everything is clear and in the open.

```
set.seed(1)
sex <- factor(rep(c("Male", "Female"), c(20, 20)))
drug <- factor(rep(rep(c("A", "B"), c(10, 10)), 2))
y <- rep(c(10, 13, 12, 16), rep(10, 4))
y <- y + rnorm(length(y), sd = 1.5)
y.data <- data.frame(y, sex, drug)
```

First, some basic stats about those data. Notice the perfect balance:

```
with(y.data, tapply(y, list(sex, drug), function(x) sum(!is.na(x))))

##           A  B
## Female 10 10
## Male   10 10

with(y.data, tapply(y, list(sex, drug), mean))

##           A          B
## Female 11.79949 16.18110
## Male   10.19830 13.37327
```

Just by eye, it seems the difference between sexes is around 2, and the difference between drugs of about 4. And no, there is no interaction:

```
summary(lm(y ~ sex * drug, data = y.data))

##
## Call:
## lm(formula = y ~ sex * drug, data = y.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.6953 -0.6854  0.1639  0.9228  2.1946
##
```

²⁴Technically, orthogonality of the design matrix does not require identical cell counts; if row/column counts are proportional, for instance, we should be OK. But, for simplicity, many of our examples when order does not matter use balanced examples.

Some basic statistics with R

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   11.7995     0.4322  27.304 < 2e-16 ***
## sexMale       -1.6012     0.6112  -2.620  0.0128 *
## drugB         4.3816     0.6112   7.169 1.97e-08 ***
## sexMale:drugB -1.2066     0.8643  -1.396  0.1712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.367 on 36 degrees of freedom
## Multiple R-squared:  0.7436, Adjusted R-squared:  0.7222
## F-statistic: 34.8 on 3 and 36 DF, p-value: 9.746e-11
```

Fit two models, simply changing the order (we assume no interaction, as shown above).

```
m1 <- lm(y ~ sex + drug, data = y.data)
m2 <- lm(y ~ drug + sex, data = y.data)
```

And we also fit two small models, one only with sex, the other only with drug:

```
msex <- lm(y ~ sex, data = y.data)
mdrug <- lm(y ~ drug, data = y.data)
```

Now, the output for the coefficients for m1 and m2 is the same (these are always the coefficients as if entered last in the model):

```
summary(m1)

##
## Call:
## lm(formula = y ~ sex + drug, data = y.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.9970 -0.7100  0.0357  0.8676  2.4963
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   12.1012     0.3790  31.927 < 2e-16 ***
## sexMale       -2.2045     0.4377  -5.037 1.26e-05 ***
## drugB         3.7783     0.4377   8.633 2.15e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.384 on 37 degrees of freedom
```


Some basic statistics with R

```
## Multiple R-squared:  0.7297, Adjusted R-squared:  0.7151
## F-statistic: 49.95 on 2 and 37 DF,  p-value: 3.079e-11

summary(m2)

##
## Call:
## lm(formula = y ~ drug + sex, data = y.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.9970 -0.7100  0.0357  0.8676  2.4963
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.1012     0.3790   31.927 < 2e-16 ***
## drugB         3.7783     0.4377    8.633 2.15e-10 ***
## sexMale      -2.2045     0.4377   -5.037 1.26e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.384 on 37 degrees of freedom
## Multiple R-squared:  0.7297, Adjusted R-squared:  0.7151
## F-statistic: 49.95 on 2 and 37 DF,  p-value: 3.079e-11
```

So nothing new up to here. Now look at what happens if we get the coefficients for the small models, those with only sex or only drug:

```
summary(msex)

##
## Call:
## lm(formula = y ~ sex, data = y.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.9743 -1.9275 -0.3339  1.9228  4.0477
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.9903     0.5302   26.39 < 2e-16 ***
## sexMale      -2.2045     0.7498   -2.94  0.00556 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.371 on 38 degrees of freedom
```

Some basic statistics with R

```
## Multiple R-squared:  0.1853, Adjusted R-squared:  0.1639
## F-statistic: 8.645 on 1 and 38 DF,  p-value: 0.005556

summary(mdrug)

##
## Call:
## lm(formula = y ~ drug, data = y.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.0992 -1.1956  0.0094  1.1359  3.2608
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.9989     0.3965  27.741 < 2e-16 ***
## drugB         3.7783     0.5607   6.738 5.57e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.773 on 38 degrees of freedom
## Multiple R-squared:  0.5444, Adjusted R-squared:  0.5324
## F-statistic: 45.41 on 1 and 38 DF,  p-value: 5.569e-08
```

In both cases, the estimate is the same from the model with the two factors, or with only a single factor. For example, the differences between sexes are of about 2.2 (the coefficient that says "sexMale") and the differences between drugs of about 3.8 (the coefficient that says "drugB"). However, the standard error and, thus, the t value and the p-value change.

Again, the key is to understand that even if the coefficient does not change whether or not the other factor is included in the model (and it does not change because there is complete balance here), the t statistic and the p-value do change. Why? Because the other factor explains a large part of variance, and thus makes the residual standard error much smaller if we include it in the model.

And what about the ANOVA tables?

```
anova(m1)

## Analysis of Variance Table
##
## Response: y
##          Df Sum Sq Mean Sq F value    Pr(>F)
## sex       1  48.599   48.599   25.371 1.258e-05 ***
## drug      1 142.754  142.754   74.527 2.149e-10 ***
## Residuals 37   70.873    1.915
```

Some basic statistics with R

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

anova(m2)

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## drug        1 142.754  142.754   74.527 2.149e-10 ***
## sex          1  48.599   48.599   25.371 1.258e-05 ***
## Residuals   37  70.873    1.915
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Order (when we include both factors, of course) does not change anything. Why? Because the contributions of each factor do not depend at all on the other (i.e., the Mean Squares of each factor does not depend on the other). And since the F is the ratio of the Mean Squares of the factor over the Mean Squares of the residuals (and this is whatever is left after we have fitted everything), the order does not affect the F statistic or the p-value.

Of course, an "Anova" (Type II tests) would show the same:

```
Anova(m1)

## Anova Table (Type II tests)
##
## Response: y
##           Sum Sq Df F value    Pr(>F)
## sex          48.599  1  25.371 1.258e-05 ***
## drug        142.754  1  74.527 2.149e-10 ***
## Residuals    70.873 37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To understand this better, look at the anova tables for the models with only one factor:

```
anova(msex)

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## sex          1  48.599   48.599   8.6447 0.005556 **
## Residuals   38 213.627    5.622
## ---
```

Some basic statistics with R

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

anova(mdrug)

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## drug        1 142.75  142.754   45.406 5.569e-08 ***
## Residuals  38  119.47    3.144
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice how the Mean Sq for each factor is the same as in the previous tables. So the Mean Squares for Sex do not depend on whether or not drug is in the model. But the F statistic (and the p-value) do change a lot. Why? Because what changes a lot are the Mean Sq. of the residuals. And why is that? Because the other factor, the one we have not included, does indeed explain a lot of variability, but in these two last tables, since the other factor is not in the model, that variability is included now in the error term.

So, to summarize: when there is balance, order does not change a thing if we include both factors in the model. However, having or not the other factor in the model can make a difference for the standard errors, the residual standard errors, and thus the p-values.

20.8 One observation per cell

Briefly: try not to fit models with a single observation per cell. You will not be able to test interactions. And, of course . . . that would be a tiny sample size anyway.

So far, we have had more than one observation for each combination of levels (i.e., more than one observation per cell, where cell means each of the “places” or “boxes” in a table that shows the combinations of treatments). What if we only had one?

Let us simulate some data:

```
set.seed(3)
df1 <- data.frame(y = runif(6),
                  A = rep(c("a1", "a2", "a3"), 2),
                  B = rep(c("b1", "b2"), rep(3, 2)))

df1

##           y  A  B
## 1 0.1680415 a1 b1
## 2 0.8075164 a2 b1
## 3 0.3849424 a3 b1
## 4 0.3277343 a1 b2
```

Some basic statistics with R

```
## 5 0.6021007 a2 b2
## 6 0.6043941 a3 b2
```

Some summaries of data:

```
(means <- with(df1, tapply(y, list(A, B), mean)))

##           b1           b2
## a1 0.1680415 0.3277343
## a2 0.8075164 0.6021007
## a3 0.3849424 0.6043941
```

We can fit the additive model (but only 1 df left)

```
m1 <- lm(y ~ A + B, data = df1)
anova(m1)

## Analysis of Variance Table
##
## Response: y
##           Df    Sum Sq  Mean Sq F value Pr(>F)
## A           2 0.209224  0.104612   3.9552 0.2018
## B           1 0.005030  0.005030   0.1902 0.7053
## Residuals   2 0.052898  0.026449

summary(m1)

##
## Call:
## lm(formula = y ~ A + B, data = df1)
##
## Residuals:
##          1          2          3          4          5          6
## -0.05089  0.13166 -0.08077  0.05089 -0.13166  0.08077
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.21893    0.13279   1.649   0.241
## Aa2           0.45692    0.16263   2.810   0.107
## Aa3           0.24678    0.16263   1.517   0.268
## Bb2           0.05791    0.13279   0.436   0.705
##
## Residual standard error: 0.1626 on 2 degrees of freedom
## Multiple R-squared:  0.802, Adjusted R-squared:  0.505
## F-statistic: 2.7 on 3 and 2 DF, p-value: 0.2818
```

But we can't really fit the interaction model:

Some basic statistics with R

```
m2 <- lm(y ~ A * B, data = df1)
anova(m2)

## Warning in anova.lm(m2): ANOVA F-tests on an essentially perfect fit
are unreliable

## Analysis of Variance Table
##
## Response: y
##          Df    Sum Sq  Mean Sq F value Pr(>F)
## A          2 0.209224  0.104612     NaN   NaN
## B          1 0.005030  0.005030     NaN   NaN
## A:B        2 0.052898  0.026449     NaN   NaN
## Residuals  0 0.000000      NaN
##
summary(m2)

##
## Call:
## lm(formula = y ~ A * B, data = df1)
##
## Residuals:
## ALL 6 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.16804         NaN     NaN     NaN
## Aa2            0.63947         NaN     NaN     NaN
## Aa3            0.21690         NaN     NaN     NaN
## Bb2            0.15969         NaN     NaN     NaN
## Aa2:Bb2       -0.36511         NaN     NaN     NaN
## Aa3:Bb2        0.05976         NaN     NaN     NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1, Adjusted R-squared:      NaN
## F-statistic:      NaN on 5 and 0 DF,  p-value: NA
```

Do you understand what is going on?

20.9 Another quick two-way example

We will use the `coking` data from the `ISwR` package (the one associated with Dalgaard's book *Introductory statistics with R*). You can access that data using the menu "Data -> Data in packages". Please, look at the description of the data. Now, we will fit a model and summarize it.

Some basic statistics with R

You should make sure what kinds of variables these are (i.e., do we have factors or numeric variables for `width` and `temp`?)

```
library(ISwR)

##
## Attaching package: 'ISwR'
## The following object is masked from 'package:survival':
##
##      lung

ck1 <- lm(time ~ width * temp, data = coking)
Anova(ck1)

## Anova Table (Type II tests)
##
## Response: time
##           Sum Sq Df F value    Pr(>F)
## width      123.143  2 222.102 3.312e-10 ***
## temp        17.209  1  62.076 4.394e-06 ***
## width:temp    5.701  2  10.283 0.002504 **
## Residuals     3.327 12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(ck1)

## Analysis of Variance Table
##
## Response: time
##           Df Sum Sq Mean Sq F value    Pr(>F)
## width       2 123.143   61.572 222.102 3.312e-10 ***
## temp        1  17.209   17.209  62.076 4.394e-06 ***
## width:temp   2   5.701    2.851  10.283 0.002504 **
## Residuals  12   3.327    0.277
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

First, make sure you understand the degrees of freedom and the rest of the output. What can you say from the above results? Would you have wanted to analyze temperature as a numerical variable? Would it have made any difference here?

Is this a balanced design? If we were to fit a model without interactions, would the order of factors change the output? Would fitting a model without interactions be justified here?

20.10 ANOVA/linear models with more than two factors

We will not present detailed examples, but life is filled with them. Think about cholesterol: to the experiment with drug and diet add a third factor: an exercise program. Or maybe a fourth factor too: a stress reduction program. Or maybe . . .

So let us gain some practice with them. Suppose we have a model with three factors, so a three-way ANOVA, with factors U, V, W. Before we continue, please write down the rows of the ANOVA table; do not fill up the numbers of SS, etc, but write down the rows. For example, one row will have “V”, another “U:V:W”, etc, etc. Fill up also the column with degrees of freedom (d.f.), assuming that U has two levels (U_1 , U_2), V has three, and W has four. Really, write down that table.

Make sure you can understand why we have this table and what each row would be testing.

Suppose we find out that:

- There is no evidence of three-way interaction.
- There is no evidence of interaction U-V.
- There is no evidence of interaction U-W.
- Only the interaction V-W is significant.

So that the next exercise is easy to do, make an annotation as “highly significant” or “obviously not significant” in the interactions according to what we wrote above (e.g., the row with the U-V-W interaction will be labelled “obviously not significant”).

Now, following what explained before (“the marginality principle”), where we said (“*Interactions*” (section 20.3)) “in the presence of interactions, we often refrain from interpreting main effects”, we will not test main effects that are marginal to any significant interaction. This means that:

- We will only examine the main effect of U.
- We will not examine the main effect of V, because there is a significant interaction V-W; in other words, the effect of V changes with the level of W.
- We will not examine the main effect of W, because there is a significant interaction V-W; in other words, the effect of W changes with the level of V. (This is exactly the same as in the previous line).

Let us repeat what we just did to make sure we understand what is going on. Write down what rows we would see in the ANOVA tables for the following cases; write down the terms and the d.f. Yes, **really, do this now**:

- An experiment with three factors, A, B, C. A has 3 levels, B has four, C has five. We fit a model without any interactions.
- As above, but the model is one with all possible interactions.

Some basic statistics with R

- As in the first case, but the model is one with interactions between A and B, and interactions between A and C, and interactions between B and C (but no three-way interaction).
 - Now, suppose the A-C interaction is NOT at all significant. What main effects can you test?
 - Now, suppose the A-C and A-B interactions are NOT at all significant. What main effects can you test?

Finally, repeat once more on your own. An experiment with four factors, the first with 2 levels, the second with 7 levels, the third with 4 levels, the fourth with 5 levels. Write the table, degrees of freedom, and think about hypotheses you can test depending on what interactions are or not significant.

20.11 Multiple comparisons of means in two-way ANOVA

This section is way too long for this course. We will cover it quickly, focusing on the key conceptual issues. But we leave the code, examples, and more advanced comments for your future reference. Why is this that long, then? Because we want to emphasize that **“p values are not enough”**: in many/most cases, you will want to look at confidence intervals and other measures of uncertainty around the parameters you have estimated and the contrasts (differences in means) of interest. And, of course, you will need to correct for multiple testing when you, well, conduct multiple tests.

In two-way and, more generally, multi-way ANOVA, computing and displaying multiple comparisons is more complicated than in one-way ANOVA. Among other questions you should ask yourself, probably some of the first ones are: At what levels of one variable will I be comparing the other? How does lack of balance affect the contrasts? Do I really want all possible contrasts? A good place to start reading on these issues is chapter 14 (and section 5.3.2) of Everitt and Hothorn's “A handbook of statistical analysis using R, 2nd ed”, and then the documentation of [multcomp](#).

20.11.1 Multiple comparisons in multi-way ANOVA: main messages

Again, it is the main messages that you need to understand. You can skip sections [20.11.2](#) and [20.11.3](#) and use them as reference when you need it. (Nope, those two sections will not be in the exam).

- In two-way (or three-way or . . . , generally, multi-way) ANOVA **without interactions** carrying out multiple comparisons between pairs of means is straightforward.
 - You might need to pay attention to which procedure (function) you use when the data are unbalanced.

Some basic statistics with R

- When there are interactions, this is more complicated. For example, at what level of each variable do you want to estimate the differences in the other? A very simple numerical example illustrating the problem is shown below in *"Comparisons between means when there are interactions: a simple example of why we need to specify at what level of the other variable/of the interaction" (section 20.11.1)*.
- Finally, in both cases, you might want to think if carrying out comparisons between **all** pairs of means is what you really want to do. (Of course, you **must** report what you really did: carrying out comparisons between all pairs and reporting only a few is incorrect, leads to biases, and is scientifically dishonest and poor practice).

Comparisons between means when there are interactions: a simple example of why we need to specify at what level of the other variable/of the interaction

This is an intuitive explanation of the problem (which is what we want for this class). Suppose we have the model

$$Y \sim A * B$$

where both A and B have two levels (A: a1, a2; B: b1, b2)

Now, suppose these are the means of each of the cells (e.g., 3 is the average of observations that have a1 and b1):

Level of A	Level of B	Mean
a1	b1	3
a1	b2	5
a2	b1	8
a2	b2	2

Please, draw a figure (by hand) with those means, like any of the interaction plots we have seen in class. Please, **really draw that figure**. Actually, **draw 2 figures**: the first will have A in the X (abscissas) axis, the second will have B in the X axis. You will see that in both plots the lines clearly cross.

Now, suppose we want this: "95% confidence interval for the difference between the means of a1 and a2". Sounds simple enough but ...

... at what level of B do you want that? Because, even without thinking about the confidence interval, the difference between the means of a1 and a2 ($a1 - a2$) is:

$$3 - 8 = -5 \quad \text{at } b1$$

$$5 - 2 = 3 \quad \text{at } b2$$

Now, please repeat the above for B. **Really, do it now**: compute the difference between b1 - b2 at each level of A.

Some basic statistics with R

So "what is the difference between a1 and a2" depends on the level of B (and this is not news: this is because there is interaction). Thus, returning a confidence interval (CI) for that difference requires us to specify at what level of B we want the CI. And the same thing happens if we want a CI for the difference between the two levels of B (at what level of A do we want it?).

But then, how do we even see plots for CI under interactions (*"Multiple comparisons in two-way ANOVA: example with interactions" (section 20.11.3)*)? Because it is possible to return that CI at different levels of the other factors. We just need to say "at this level of the other factor". For example at "the average" (because we can define the "average effect of A" and "the average effect of B", and "the average effect of the interaction", in a model with interactions). But computing the CI at this "average" might not be what you want to do. With data that look like the one above, with such extreme crossing of lines, that might not make much sense. Alternatively, you might compute the difference and confidence interval at some other level of the other factor that is more relevant for your question. Whether or not it makes sense to compute the CI at the average level of the interaction, or at what level it makes sense to do it is **highly context dependent**: this depends on how strong the interactions are (e.g., are effects being reverted, as in this example?), and what your scientific question is. Regardless, how that average of B/average of A/average of the interaction is defined and computed, how the average difference at the average of the other variable is computed, how that is done in R, how to specify other levels, etc, do not matter to us here.

What matters is understanding that, with interactions, asking a question about "the difference between levels of A" (or "a confidence interval around the difference between levels of A") cannot be answered without saying something about the level of B at which we want to compute that difference.

And this, by the way, is strongly related to the "marginality principle" we have already discussed twice (20.3, 20.10).

Now you can skip sections 20.11.2 and 20.11.3 and continue to section 20.12.

20.11.2 Multiple comparisons in two-way ANOVA: example with no interactions

I will show a few examples below. We will use function `mmc` from package *HH* as well as function `glht` from package *multcomp*. I will skip most of the syntax here.

For the sake of illustration, let's pretend (even when we know it is wrong!!!) that there is no interaction between Drug and Diet in the cholesterol data.

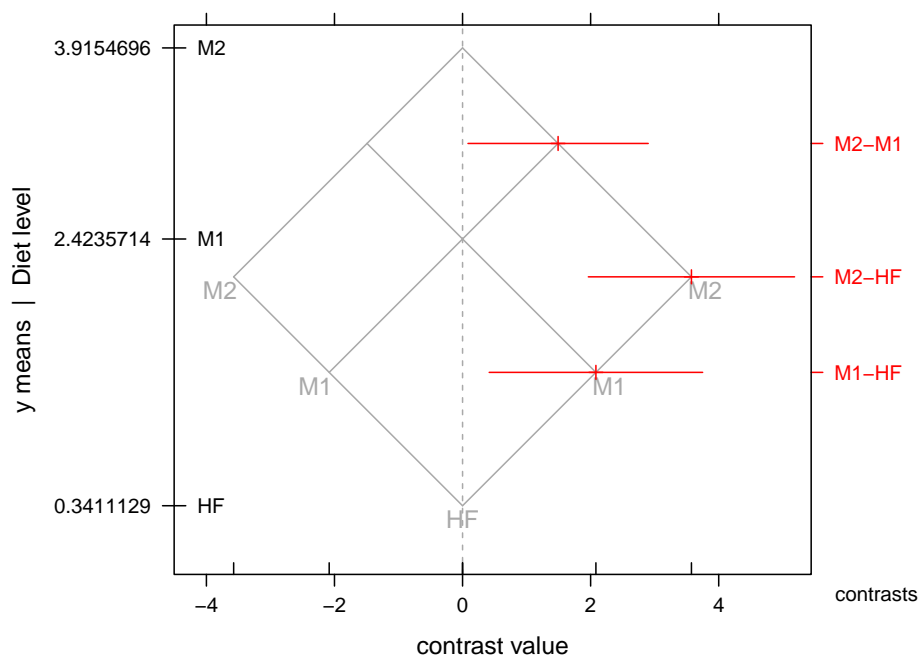
```
## We must make sure we are using factors with mmc
dc2 <- dcholest
dc2$Drug <- as.factor(dc2$Drug)
dc2$Diet <- as.factor(dc2$Diet)
clmA <- lm(y ~ Diet + Drug, data = dc2)
cholA_mmc <- mmc(clmA, focus = "Diet")
```

Some basic statistics with R

```
cholA_mmc

## Tukey contrasts
## Fit: lm(formula = y ~ Diet + Drug, data = dc2)
## Estimated Quantile = 2.4407
## 95% family-wise confidence level
## $mca
##      estimate      stderr      lower      upper      height
## M2-M1 1.491898 0.5765392 0.0847389 2.899058 3.169521
## M2-HF 3.574357 0.6597161 1.9641874 5.184526 2.128291
## M1-HF 2.082459 0.6828711 0.4157749 3.749142 1.382342
## $none
##      estimate      stderr      lower      upper      height
## M2 3.9154696 0.3876035 2.9694457 4.861494 3.9154696
## M1 2.4235714 0.4268032 1.3818728 3.465270 2.4235714
## HF 0.3411129 0.5330590 -0.9599245 1.642150 0.3411129

mmcplot(cholA_mmc)
```



In this two-way ANOVA we might want to compare, simultaneously, the levels of Diet and Drug (see the documentation of [multcomp](https://cran.r-project.org/web/packages/multcomp/vignettes/multcomp-examples.pdf) (in particular, <https://cran.r-project.org/web/packages/multcomp/vignettes/multcomp-examples.pdf>)).

```
KA1 <- glht(clmA, mcp(Diet = "Tukey"))$linfct
KA2 <- glht(clmA, mcp(Drug = "Tukey"))$linfct
clmA_glh <- glht(clmA, linfct = rbind(KA1, KA2))
```

Some basic statistics with R

```
summary(clmA_glh)

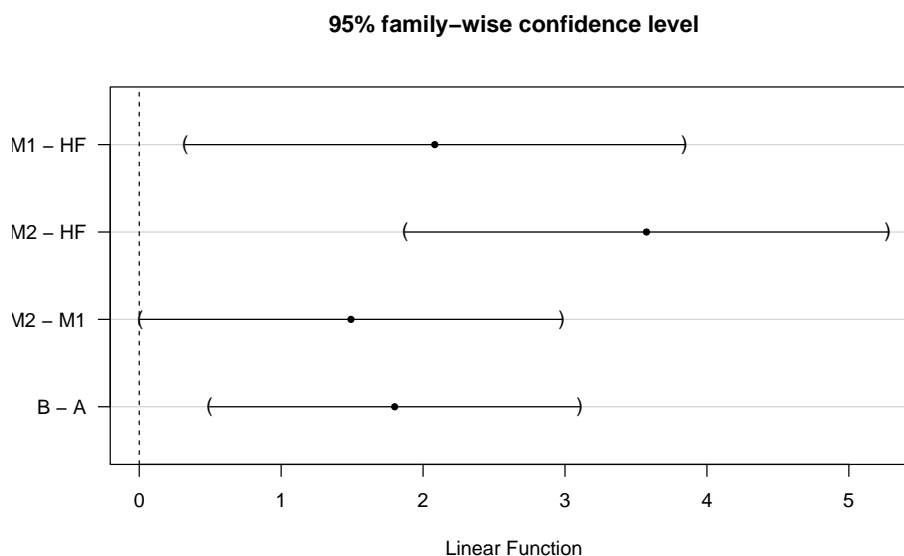
##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = y ~ Diet + Drug, data = dc2)
##
## Linear Hypotheses:
##           Estimate Std. Error t value Pr(>|t|)
## M1 - HF == 0    2.0825      0.6829   3.050  0.01573 *
## M2 - HF == 0    3.5744      0.6597   5.418 < 0.001 ***
## M2 - M1 == 0    1.4919      0.5765   2.588  0.04875 *
## B - A == 0      1.8000      0.5061   3.557  0.00403 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

## Note that these are wider than the ones above, as we should expect.
confint(clmA_glh)

##
## Simultaneous Confidence Intervals
##
## Fit: lm(formula = y ~ Diet + Drug, data = dc2)
##
## Quantile = 2.5764
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##           Estimate lwr      upr
## M1 - HF == 0  2.082459 0.323102 3.841815
## M2 - HF == 0  3.574357 1.874657 5.274057
## M2 - M1 == 0  1.491898 0.006496 2.977300
## B - A == 0    1.799968 0.496107 3.103829

plot(clmA_glh)
```

Some basic statistics with R



You can also obtain the above using `TukeyHSD` (there are further comments about the differences below —bottom line here: when sample sizes are not very unbalanced, the results will be very, very similar)²⁵.

You can stop reading here, but remember that there is extra material below if you needed it.

20.11.3 Multiple comparisons in two-way ANOVA: example with interactions

But we know there are interactions. Let us use that model to compare levels of Diet.

```
clm <- lm(y ~ Diet*Drug, data = dc2)
chol_mmc <- mmc(clm, focus = "Diet")

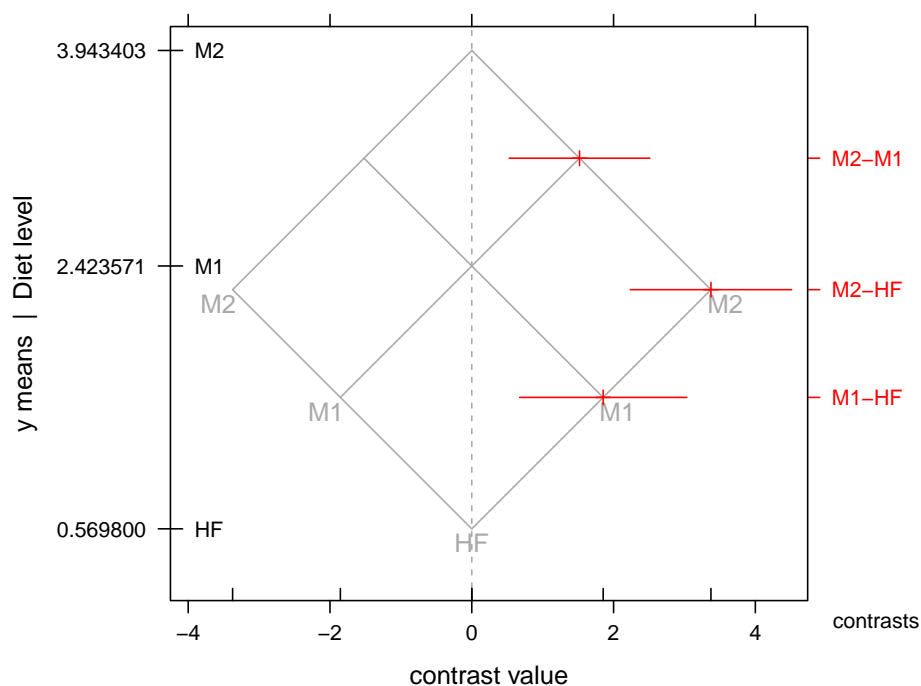
chol_mmc

## Tukey contrasts
## Fit: lm(formula = y ~ Diet * Drug, data = dc2)
## Estimated Quantile = 2.449133
## 95% family-wise confidence level
## $mca
##      estimate   stderr   lower   upper   height
## M2-M1 1.519831 0.4053834 0.5269933 2.512669 3.183487
## M2-HF 3.373603 0.4648369 2.2351552 4.512050 2.256601
## M1-HF 1.853771 0.4813467 0.6748892 3.032654 1.496686
## $none
##      estimate   stderr   lower   upper   height
## M2 3.943403 0.2726852 3.2755603 4.611245 3.943403
## M1 2.423571 0.2999641 1.6889192 3.158224 2.423571
```

²⁵Though the plots might not look as nice or you might need to do more work to put all plots on the same page

Some basic statistics with R

```
## HF 0.569800 0.3764520 -0.3521812 1.491781 0.569800
mmcplot(chol_mmc)
```



What is happening with the interactions? If you read the help of `mmc` you will see it mentions options `interaction_average` and `covariate_average`.

What if we set those to false? Notice the warning, and see the differences!

```
chol_mmc2 <- mmc(c1m,
  linfct = mcp(Diet = "Tukey",
    `interaction_average` = FALSE,
    `covariate_average` = FALSE))

## Warning in mcp2matrix(model, linfct = linfct): covariate interactions
## found - default contrast might be inappropriate

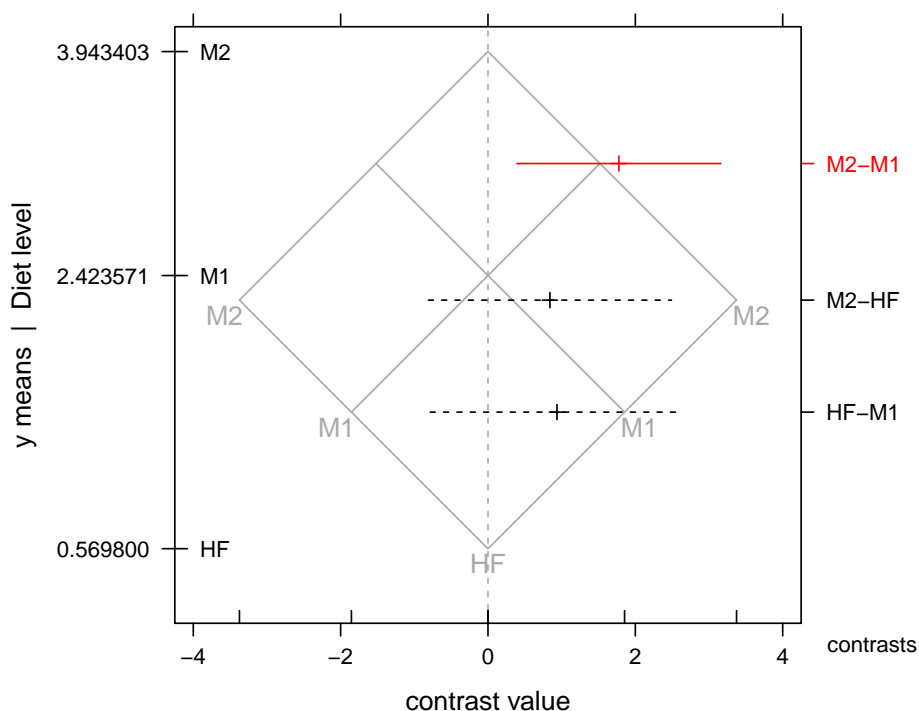
chol_mmc2

## Tukey contrasts
## Fit: lm(formula = y ~ Diet * Drug, data = dc2)
## Estimated Quantile = 2.44608
## 95% family-wise confidence level
## $mca
##      estimate   stderr   lower   upper   height
## M2-M1 1.7771270 0.5656178 0.3935804 3.160674 3.183487
## M2-HF 0.8405556 0.6744563 -0.8092185 2.490330 2.256601
## HF-M1 0.9365714 0.7034783 -0.7841928 2.657336 1.496686
```

Some basic statistics with R

```
## $none
##      estimate      stderr      lower      upper      height
## M2 3.943403 0.2726852 3.2763929 4.610413 3.943403
## M1 2.423571 0.2999641 1.6898351 3.157308 2.423571
## HF 0.569800 0.3764520 -0.3510318 1.490632 0.569800

mmcplot(chol_mmc2)
```



Using directly `glht` to compare both Diets and Drugs:

```
K1 <- glht(clm, mcp(Diet = "Tukey"))$linfct
## Warning in mcp2matrix(model, linfct = linfct): covariate interactions
## found - default contrast might be inappropriate

K2 <- glht(clm, mcp(Drug = "Tukey"))$linfct
## Warning in mcp2matrix(model, linfct = linfct): covariate interactions
## found - default contrast might be inappropriate

K1B <- glht(clm, mcp(Diet = "Tukey",
                    interaction_average = TRUE,
                    covariate_average = TRUE))$linfct

K2B <- glht(clm, mcp(Drug = "Tukey",
                    interaction_average = TRUE,
                    covariate_average = TRUE))$linfct
```


Some basic statistics with R

```
confint(glht(clm, linfct = rbind(K1, K2)))

##
## Simultaneous Confidence Intervals
##
## Fit: lm(formula = y ~ Diet * Drug, data = dc2)
##
## Quantile = 2.5512
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##           Estimate lwr      upr
## M1 - HF == 0 -0.9366 -2.7313  0.8582
## M2 - HF == 0  0.8406 -0.8801  2.5612
## M2 - M1 == 0  1.7771  0.3341  3.2201
## B - A == 0   -2.3164 -4.2372 -0.3956

confint(glht(clm, linfct = rbind(K1B, K2B)))

##
## Simultaneous Confidence Intervals
##
## Fit: lm(formula = y ~ Diet * Drug, data = dc2)
##
## Quantile = 2.5834
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##           Estimate lwr      upr
## M1 - HF == 0  1.8538  0.6103  3.0973
## M2 - HF == 0  3.3736  2.1727  4.5745
## M2 - M1 == 0  1.5198  0.4726  2.5671
## B - A == 0   1.2325  0.2797  2.1853

## Compare with previously computed ones
## A few change sign as we do HF - M1 instead
## of M1 - HF
## chol_mmc2
## chol_mmc
```

But in models with interactions, we might be interested in doing other things. For example, comparing the levels of Drug within levels of Diet. There are examples in the documentation of *multcomp* (see <https://cran.r-project.org/web/packages/multcomp/vignettes/multcomp-examples.pdf>).

Some basic statistics with R

Or comparing, simultaneously, the levels of Diet and Drug (again, see the vignette):

```
K1 <- glht(clm, mcp(Diet = "Tukey"))$linfct
## Warning in mcp2matrix(model, linfct = linfct): covariate interactions
## found - default contrast might be inappropriate

K2 <- glht(clm, mcp(Drug = "Tukey"))$linfct
## Warning in mcp2matrix(model, linfct = linfct): covariate interactions
## found - default contrast might be inappropriate

K1A <- glht(clm, mcp(Diet = "Tukey",
                    interaction_average = TRUE,
                    covariate_average = TRUE))$linfct

K2A <- glht(clm, mcp(Drug = "Tukey",
                    interaction_average = TRUE,
                    covariate_average = TRUE))$linfct

summary(glht(clm, linfct = rbind(K1, K2)))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = y ~ Diet * Drug, data = dc2)
##
## Linear Hypotheses:
##           Estimate Std. Error t value Pr(>|t|)
## M1 - HF == 0  -0.9366    0.7035  -1.331   0.4612
## M2 - HF == 0   0.8406    0.6745   1.246   0.5137
## M2 - M1 == 0   1.7771    0.5656   3.142   0.0119 *
## B - A == 0    -2.3164    0.7529  -3.077   0.0140 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

summary(glht(clm, linfct = rbind(K1A, K2A)))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = y ~ Diet * Drug, data = dc2)
##
## Linear Hypotheses:
##           Estimate Std. Error t value Pr(>|t|)
## M1 - HF == 0   1.8538    0.4813   3.851 0.00189 **
## M2 - HF == 0   3.3736    0.4648   7.258 < 1e-04 ***
```

Some basic statistics with R

```
## M2 - M1 == 0    1.5198      0.4054    3.749  0.00251 **
## B - A == 0      1.2325      0.3688    3.342  0.00752 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

## Compare with previously computed one
chol_mmc2

## Tukey contrasts
## Fit: lm(formula = y ~ Diet * Drug, data = dc2)
## Estimated Quantile = 2.44608
## 95% family-wise confidence level
## $mca
##      estimate   stderr      lower    upper   height
## M2-M1 1.7771270 0.5656178  0.3935804 3.160674 3.183487
## M2-HF 0.8405556 0.6744563 -0.8092185 2.490330 2.256601
## HF-M1 0.9365714 0.7034783 -0.7841928 2.657336 1.496686
## $none
##      estimate   stderr      lower    upper   height
## M2 3.943403 0.2726852  3.2763929 4.610413 3.943403
## M1 2.423571 0.2999641  1.6898351 3.157308 2.423571
## HF 0.569800 0.3764520 -0.3510318 1.490632 0.569800

chol_mmc

## Tukey contrasts
## Fit: lm(formula = y ~ Diet * Drug, data = dc2)
## Estimated Quantile = 2.449133
## 95% family-wise confidence level
## $mca
##      estimate   stderr      lower    upper   height
## M2-M1 1.519831 0.4053834 0.5269933 2.512669 3.183487
## M2-HF 3.373603 0.4648369 2.2351552 4.512050 2.256601
## M1-HF 1.853771 0.4813467 0.6748892 3.032654 1.496686
## $none
##      estimate   stderr      lower    upper   height
## M2 3.943403 0.2726852  3.2755603 4.611245 3.943403
## M1 2.423571 0.2999641  1.6889192 3.158224 2.423571
## HF 0.569800 0.3764520 -0.3521812 1.491781 0.569800
```

Note that many of those can be obtained, too, using the [TukeyHSD](#) function. For example, as simple as this:

```
## Fit as aov
clmaov <- aov(y ~ Diet * Drug, data = dc2)
```

Some basic statistics with R

```
## Compare to, for example, chol_mmc
TukeyHSD(clmaov, which = "Diet")

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = y ~ Diet * Drug, data = dc2)
##
## $Diet
##          diff          lwr          upr          p adj
## M1-HF 1.982460 0.8074131 3.157507 0.0006297
## M2-HF 3.421418 2.2876675 4.555169 0.0000000
## M2-M1 1.438958 0.4463700 2.431546 0.0031980

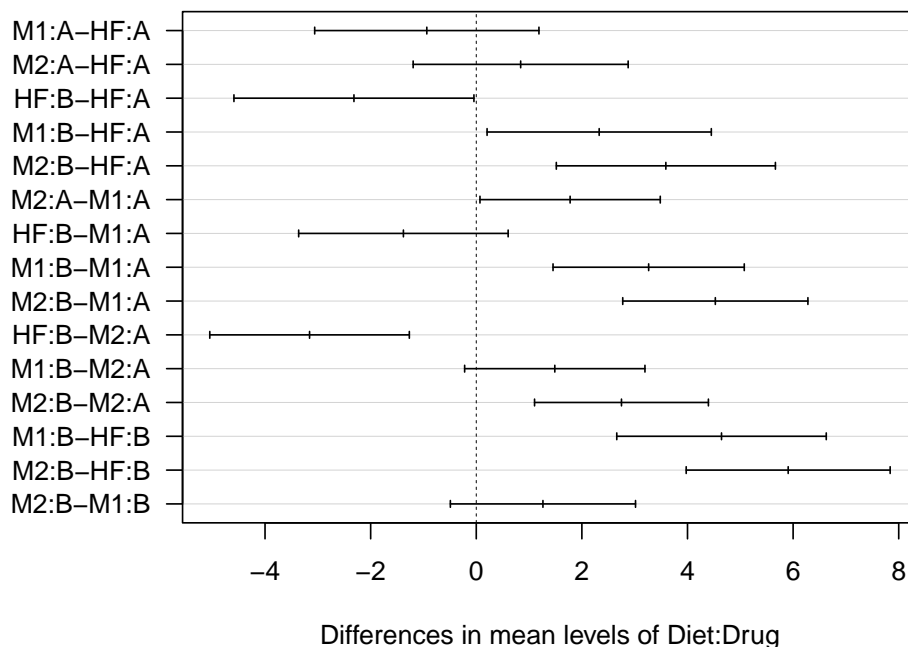
## We want all possible contrasts for all combinations
TukeyHSD(clmaov, which = "Diet:Drug")

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = y ~ Diet * Drug, data = dc2)
##
## $`Diet:Drug`
##          diff          lwr          upr          p adj
## M1:A-HF:A -0.9365714 -3.05983918 1.18669633 0.7658397
## M2:A-HF:A 0.8405556 -1.19511673 2.87622784 0.8110085
## HF:B-HF:A -2.3164000 -4.58884664 -0.04395336 0.0436082
## M1:B-HF:A 2.3277143 0.20444653 4.45098204 0.0248396
## M2:B-HF:A 3.5902500 1.51579952 5.66470048 0.0001206
## M2:A-M1:A 1.7771270 0.06995548 3.48429849 0.0373350
## HF:B-M1:A -1.3798286 -3.36338262 0.60372548 0.3115396
## M1:B-M1:A 3.2642857 1.45355689 5.07501454 0.0000635
## M2:B-M1:A 4.5268214 2.77359078 6.28005208 0.0000001
## HF:B-M2:A -3.1569556 -5.04644817 -1.26746294 0.0002052
## M1:B-M2:A 1.4871587 -0.22001278 3.19433024 0.1176209
## M2:B-M2:A 2.7496944 1.10363449 4.39575440 0.0002058
## M1:B-HF:B 4.6441143 2.66056024 6.62766833 0.0000005
## M2:B-HF:B 5.9066500 3.97544171 7.83785829 0.0000000
## M2:B-M1:B 1.2625357 -0.49069493 3.01576636 0.2761644

## Plot them
## But make sure y-axis labels are horizontal
## and y-axis labels fit
op <- par(las = 1, mar = c(5, 8, 4, 4))
plot(TukeyHSD(clmaov, which = "Diet:Drug"))
```

Some basic statistics with R

95% family-wise confidence level



```
par(op) ## return graphical parameters to previous stage
```

```
## Though if we only care about the contrasts
```

```
## between all factor combinations maybe we really just want this?
```

```
TukeyHSD(aov(y ~ Diet:Drug, data = dc2))
```

```
## Tukey multiple comparisons of means
```

```
## 95% family-wise confidence level
```

```
##
```

```
## Fit: aov(formula = y ~ Diet:Drug, data = dc2)
```

```
##
```

```
## $`Diet:Drug`
```

```
##          diff          lwr          upr      p adj
## M1:A-HF:A -0.9365714 -3.05983918  1.18669633 0.7658397
## M2:A-HF:A  0.8405556 -1.19511673  2.87622784 0.8110085
## HF:B-HF:A -2.3164000 -4.58884664 -0.04395336 0.0436082
## M1:B-HF:A  2.3277143  0.20444653  4.45098204 0.0248396
## M2:B-HF:A  3.5902500  1.51579952  5.66470048 0.0001206
## M2:A-M1:A  1.7771270  0.06995548  3.48429849 0.0373350
## HF:B-M1:A -1.3798286 -3.36338262  0.60372548 0.3115396
## M1:B-M1:A  3.2642857  1.45355689  5.07501454 0.0000635
## M2:B-M1:A  4.5268214  2.77359078  6.28005208 0.0000001
## HF:B-M2:A -3.1569556 -5.04644817 -1.26746294 0.0002052
## M1:B-M2:A  1.4871587 -0.22001278  3.19433024 0.1176209
```

Some basic statistics with R

```
## M2:B-M2:A 2.7496944 1.10363449 4.39575440 0.0002058
## M1:B-HF:B 4.6441143 2.66056024 6.62766833 0.0000005
## M2:B-HF:B 5.9066500 3.97544171 7.83785829 0.0000000
## M2:B-M1:B 1.2625357 -0.49069493 3.01576636 0.2761644

## Diet and Drug. Compare to summary(glht(clm, linfct = rbind(K1A, K2A)))
TukeyHSD(clmaov, which = c("Diet", "Drug"))

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = y ~ Diet * Drug, data = dc2)
##
## $Diet
##          diff          lwr          upr      p adj
## M1-HF 1.982460 0.8074131 3.157507 0.0006297
## M2-HF 3.421418 2.2876675 4.555169 0.0000000
## M2-M1 1.438958 0.4463700 2.431546 0.0031980
##
## $Drug
##          diff          lwr          upr      p adj
## B-A 1.792321 1.071032 2.51361 1.48e-05
```

A virtue of **TukeyHSD** is its simplicity. In models without interactions, results of these procedures are often very, very similar, though not necessarily identical (with unbalanced designs, specially heavily unbalanced, **glht** is often preferable — see for example the discussion in Everitt and Hothorn, chapters 5 and 14).

In models with interactions, **glht** allows us to control what to do with averaging over interactions and it will not necessarily give the same results as **TukeyHSD** (if there is no unbalance in sample sizes, results will often be very similar if we use `interaction_average = TRUE`). See further discussion, for example, here: <https://stat.ethz.ch/pipermail/r-help/2012-January/299623.html>. But this is, as we anticipated, a complicated issue. What comparisons are you interested in when there are interactions?

Finally, another package that might be worth checking is **emmeans** (<https://cran.r-project.org/web/packages/emmeans/index.html>). In many cases, or in other types of models, not covered in this course (e.g., mixed-effects, generalized linear models, etc) it might be easier or more flexible than the packages used above (see, for instance, <https://cran.r-project.org/web/packages/emmeans/vignettes/comparisons.html>). The latest edition of Fox and Weisberg's "An R companion to applied regression" makes extensive use of **emmeans** (to obtain confidence intervals and many other things).

20.12 Nonparametric alternatives

Are there nonparametric versions of the above procedures? For the one-way ANOVA the Kruskal-Wallis test is popular. For two-way designs with one observation per cell the Friedman test and the Quade test. But testing interactions is not easy; one needs to use more sophisticated approaches as in permutation tests conditioning on permuting only within rows or columns, etc. Moreover, as was the case with the Wilcoxon test, nonparametric and permutation procedures might not be testing what you think they are testing, and they can be sensitive to features you are not interested in, or insensitive to what you are really interested in. These procedures, of course, are sometimes what needs to be done, but you should really understand what they are really doing and why an ANOVA will not do. We will not pursue this any further.

21 Simple linear regression

This is another form of a linear model. But, now, the independent variable is continuous. So we will fit a line:

$$Y = \alpha + \beta X + \epsilon$$

where Y is, as usual, the dependent variable, X the independent, β is the slope and α the intercept (this is just the equation for a line). The simple linear regression procedure will estimate α and β , finding values $(\hat{\alpha}, \hat{\beta})$ that produce a **best fitting line** (note: it is a line, not an arbitrary curve).

We will use a subset of data from the AnAge data set (Animal Ageing and Longevity Database) (accessed on 2014-08-19) from <http://genomics.senescence.info/species/>. This file contains longevity, metabolic rate, body mass, and a variety of other life history variables. The data I provide you are a small subset that includes only some birds and reptiles.

Read the full data and call it `anage_a_r` (the a and r stand for aves and reptilia, the proper Class names).

We want to take the log of all the relevant continuous variables (yes, you would not know this before hand, but I do, so create those new variables now to avoid going back later)²⁶.

```
anage_a_r$logMetabolicRate <- log(anage_a_r$Metabolic.rate..W.)
anage_a_r$logBodyMass <- log(anage_a_r$Body.mass..g.)
anage_a_r$logLongevity <- log(anage_a_r$Maximum.longevity..yrs.)
```

For now, we will only use the birds. So use subsetting to keep only birds and call it `anage_a`

We want to model metabolic rate as a function of body mass (note that this data set is rather nice, because column names are nicely labeled and include information about units). **Beware:** what we are going to do is not correct, as the data are not independent (species share common ancestors, and they are related in varying degrees, as any phylogenetic tree would show you, and as you should be able to tell from looking at the names of some species). So we are violating the assumption of independence. What we are doing here is just for the sake of the example, and because this is a nice set of data²⁷.

```
metab <- lm(Metabolic.rate..W. ~ Body.mass..g., data = anage_a)
summary(metab)
```

²⁶Creating these new variables is not really necessary in general for fitting models. But some functions from the `HH` package lead to problems if we don't.

²⁷This can be done correctly, incorporating phylogenetic information in the regression model, but this is way out of the scope of this class. It is a really fascinating topic, though! This is often referred to as using the comparative method in evolutionary biology.

Some basic statistics with R

```
##
## Call:
## lm(formula = Metabolic.rate..W. ~ Body.mass..g., data = anage_a)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.2390 -0.3386 -0.2095  0.1578  3.8380
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5300123   0.0694005    7.637 1.74e-12 ***
## Body.mass..g. 0.0025673   0.0001133   22.663 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7975 on 164 degrees of freedom
## (1020 observations deleted due to missingness)
## Multiple R-squared:  0.758, Adjusted R-squared:  0.7565
## F-statistic: 513.6 on 1 and 164 DF, p-value: < 2.2e-16
```

The row of the output that says “(Intercept)” gives you the estimate of the intercept. The t-statistic (under “t value”) is testing that the intercept is zero. And it is not. But tests about the intercept are rarely interesting (except for cases with a natural and meaningful 0). The second line is more interesting: that is the slope, how much metabolic rate increases per unit increase in body mass (of course, to interpret this we need to know the units!). And the t-statistic tests if the slope is 0. There is certainly strong evidence that Metabolic rate increases with body mass.

Do you know what “R-squared” refers to (we explain this in more detail in section “*R² and Adjusted R²*” (section 22.3))? And the rest of the output?

By the way, did you see the note about missingness? Do you know what that means?

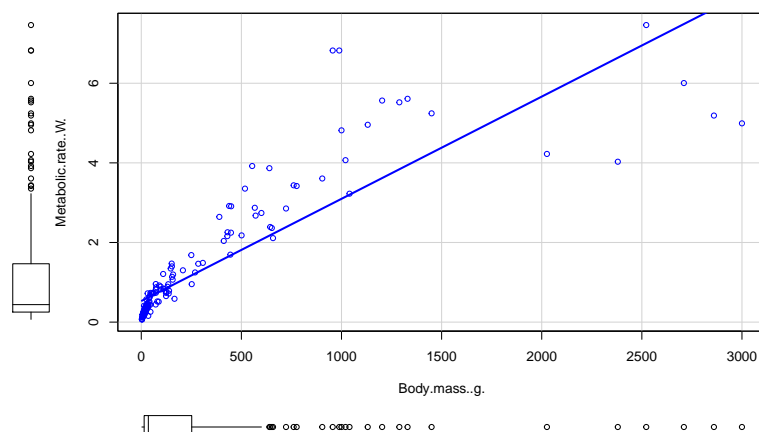
21.1 And how does it look like: should have plotted the data!

Eh!!! We should probably have plotted the data as the first thing. First, let’s do a scatterplot (you might want to not show the spread ²⁸):

```
scatterplot(Metabolic.rate..W. ~ Body.mass..g.,
            smooth = FALSE,
            data = anage_a)
```

²⁸I find the spread information to be confusing. But I like to leave the smoothed line, as it can help me see errors in the model specification.

Some basic statistics with R



21.1.1 Transforming the data

Hummm... That plot does not look good. OK, let's refit a model, but this time let's transform both the dependent and independent variables with a log (why a log? theory and previous empirical evidence from the field of allometry and life history suggest that it is a reasonable way to go).

First fit the model: ²⁹

```
metablog <- lm(logMetabolicRate ~ logBodyMass, data = anage_a)
summary(metablog)

##
## Call:
## lm(formula = logMetabolicRate ~ logBodyMass, data = anage_a)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00686 -0.14349  0.01545  0.16584  0.61638
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.15949    0.04895  -64.55  <2e-16 ***
## logBodyMass   0.65037    0.01095   59.38  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2452 on 164 degrees of freedom
## (1020 observations deleted due to missingness)
## Multiple R-squared:  0.9556, Adjusted R-squared:  0.9553
## F-statistic: 3527 on 1 and 164 DF, p-value: < 2.2e-16
```

²⁹You could have fitted the model as

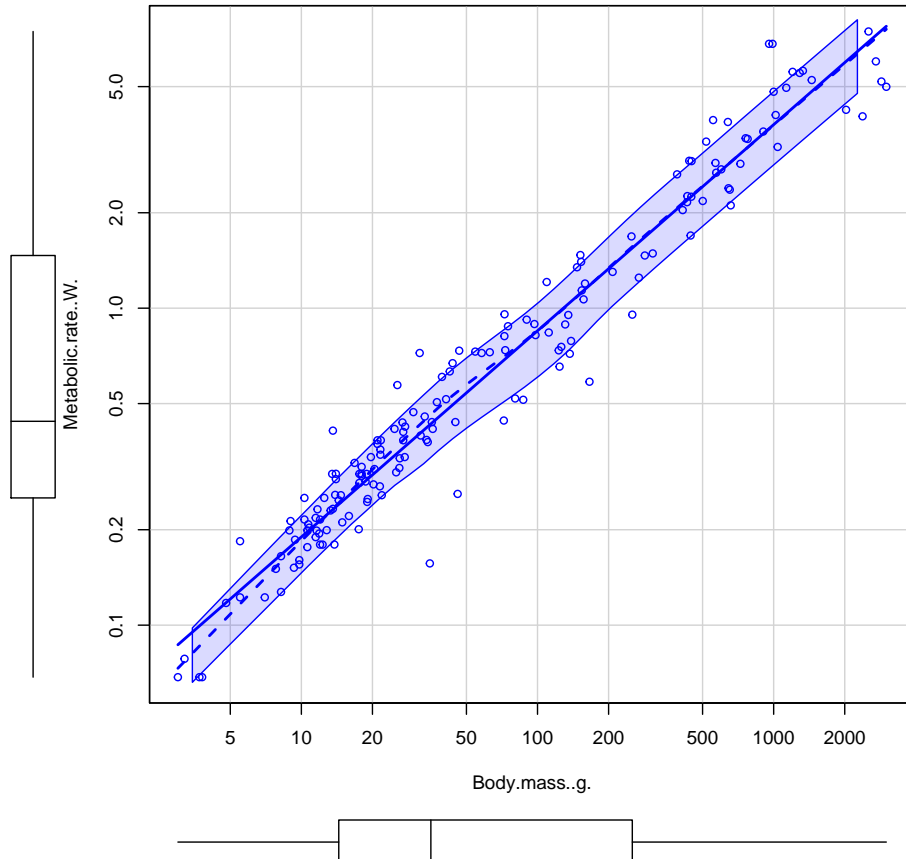
```
metablog<-lm(log(Metabolic.rate..W.)~log(Body.mass..g.),data=anage_a)
```

and that would have been fine. But then, functions `ci.plot` and `ancova` from `HH` choke.

Some basic statistics with R

Now plot it:

```
scatterplot(Metabolic.rate..W. ~ Body.mass..g., log = "xy",  
            smooth = TRUE, boxplots = 'xy',  
            data = anage_a)
```



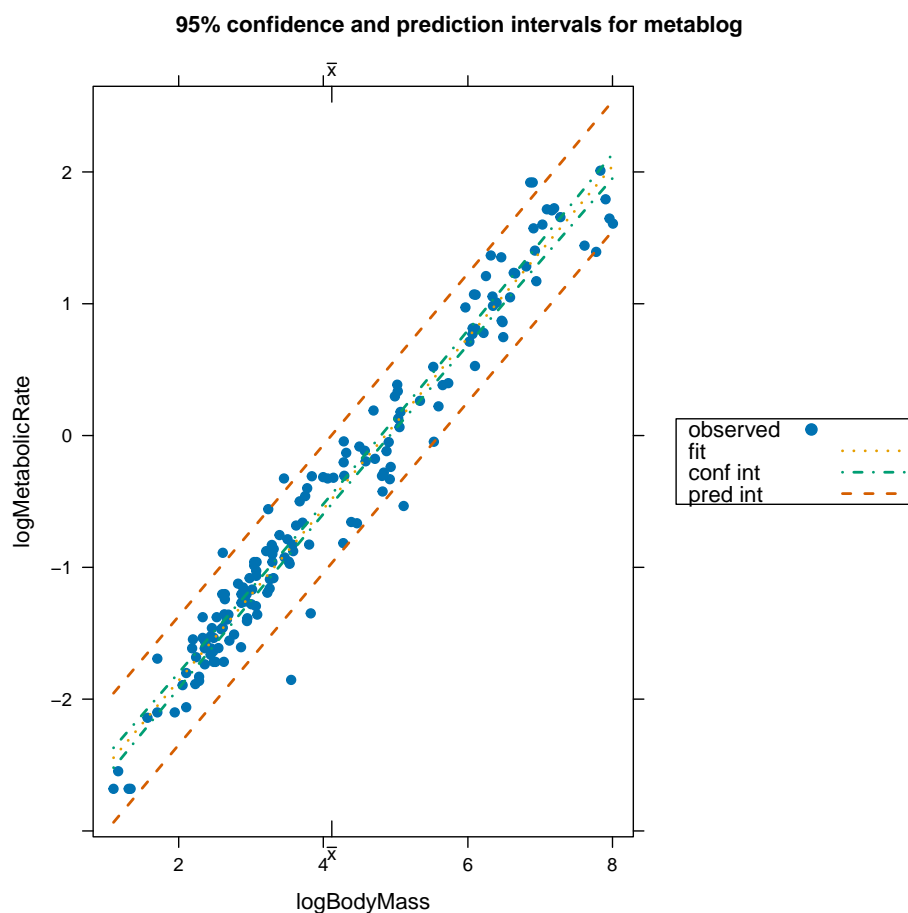
This is much, much better. We will address this issue more formally below (section [Diagnostics](#)). Notice that the call to `scatterplot` uses the original variables but the axis are in log-scale, which is nicer than directly plotting (in linear scale) the log-transformed variables: you can see the original values.

But this was a particularly simple example since I told you how to transform the data. You should be asking yourself: how do I know what transformation to use? Often, theory (should allometric patterns scale with the log? shouldn't we use the square root for phenomena that take place on surfaces? etc) can guide us. Otherwise, there are procedures to try to identify transformations, including some diagnostic plots that can help (e.g., component+residual plots, section [Diagnostics](#)).

21.2 Confidence intervals and predictions intervals or prediction and confidence bands

Some basic statistics with R

```
library(HH)
ci.plot(metablog)
```

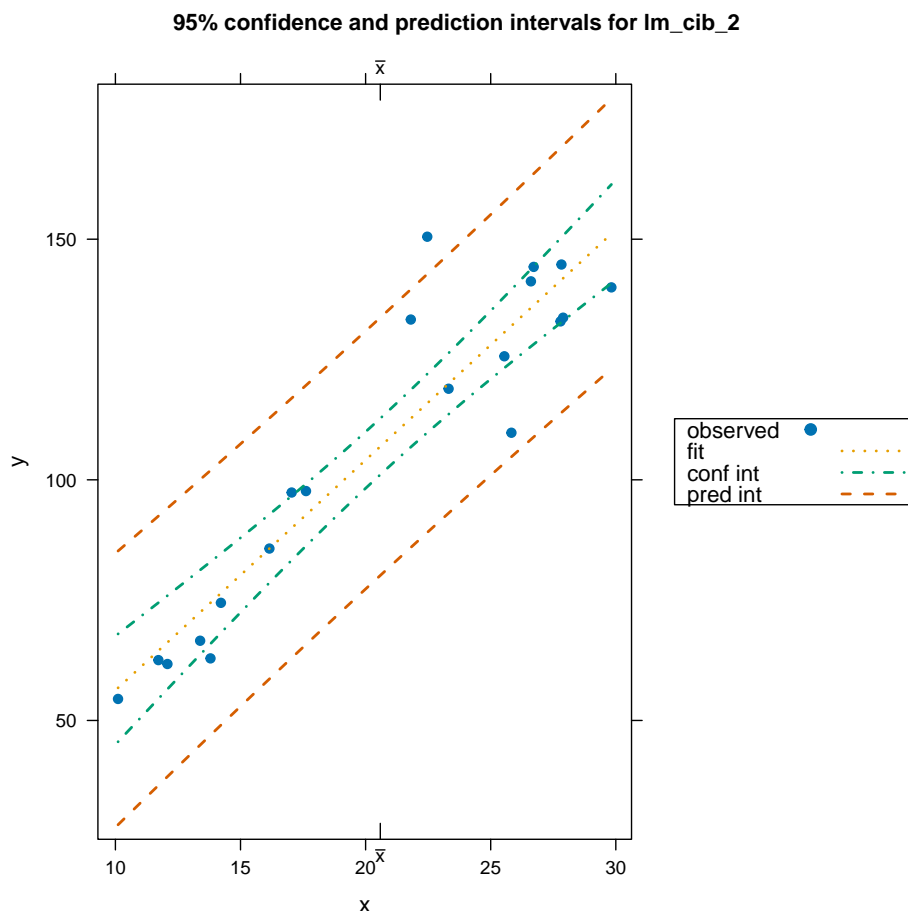


What are the bands? Make sure you understand the difference between the confidence interval and the prediction interval bands.

Maybe an example with more noise will help:

```
N <- 20
x <- runif(N, min = 10, max = 30)
y <- 3 + 5 * x + rnorm(N, sd = 10)
dummy_data <- data.frame(y = y, x = x)
rm(y, x)
lm_cib_2 <- lm(y ~ x, data = dummy_data)
ci.plot(lm_cib_2)
```

Some basic statistics with R



You can run the previous code changing the value of N , to get an intuition of the difference between the two bands.

Basically:

- Confidence interval bands are for the regression line itself which is the same as saying that they are for the expected value of the response variable. (We are modeling $E[y] = \alpha + \beta x$). The more uncertain we are about the overall trend, about the line, the wider the confidence interval bands will be.
- Prediction interval bands are for the observations; so, in addition to the uncertainty around the regression line, we have the σ , the variance of the observations around their mean value, around $E[y]$.
- To clarify the above, think about this: suppose you take a huge sample, say of size 10 million, of people and do a regression of body mass on body height. You will estimate the regression line with very little uncertainty. In other words, you will be very, very certain about where $E[\text{body_mass}]$ is given body_height . But ... no matter your sample size, there is quite a bit of variation in body mass for a given body height. So the prediction bands will be relatively wide, to accomodate this fact.

Some basic statistics with R

This is, by the way, the reason why you can be very certain about an overall trend (the expected value) and yet be unable to really predict any one specific individual's actual value. It is crucial to understand the difference between predicting the mean and predicting a specific individual's value.

In contrast, if σ is very, very, very tiny, then the prediction bands will be very, very close to the confidence interval bands.

21.3 Confidence intervals for the parameters

For this course, what follows is of much less interest, but just so that you have it here. You can skip it if you want.

You can of course obtain confidence intervals for the parameters of the model:

```
confint(metablog)

##                2.5 %    97.5 %
## (Intercept) -3.256139 -3.062837
## logBodyMass  0.628743  0.671992
```

Those are confidence intervals for the parameters themselves. The estimates of the parameters, however, are correlated. This means that testing each parameter on its own can lead to different answers from testing both at once. Let us show a joint confidence ellipse. I follow here Faraway's "Linear models with R" and Fox and Weisberg's "An R companion to applied regression". (There is no need for you to try to replicate this)

```
## Correlation of estimated coefficients
round(cov2cor(vcov(metablog)), 3)

##           (Intercept) logBodyMass
## (Intercept)      1.000      -0.921
## logBodyMass     -0.921      1.000

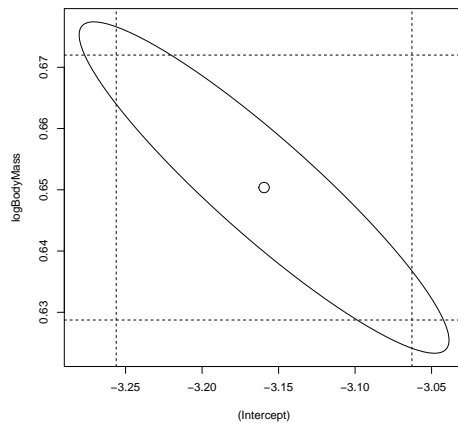
## Plot of joint and each-at-time CIs
library(ellipse)

##
## Attaching package: 'ellipse'
##
## The following object is masked from 'package:car':
##
##     ellipse
##
## The following object is masked from 'package:graphics':
##
##     pairs

plot(ellipse(metablog), type = "l")
```

Some basic statistics with R

```
points(coef(metablog)[1], coef(metablog)[2], pch = 1, cex = 2)
abline(v = confint(metablog)[1, ], lty = 2)
abline(h = confint(metablog)[2, ], lty = 2)
```



An alternative to `ellipse` is `car`'s `confidenceEllipse`.

(And no, there is nothing wrong with this correlation. It actually makes a lot of sense. Think about tilting the regression line, keeping the center of mass fixed, so play with increasing or decreasing the slope: what happens with the intercept?)

22 Multiple regression

22.1 Introduction to multiple regression and the cystfibr data

In the previous section we had

$$Y = \alpha + \beta X + \epsilon$$

Now we can have two or more independent variables:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \dots + \epsilon$$

I will use a dataset that is a small subset from the original `cystfibr` data set from package *ISwR* (by Peter Dalgaard; this package is also material to accompany Dalgaard's book "Introductory statistics with R")

Import the dataset.

```
cystfibr2 <- read.table("CystFibr2.txt", header = TRUE)
```

The meaning of the variables is (this is copied verbatim from the help of the original dataset):

```
'age' a numeric vector, age in years.  
'sex' a numeric vector code, 0: male, 1:female.  
'height' a numeric vector, height (cm).  
'weight' a numeric vector, weight (kg).  
'pemax' a numeric vector, maximum expiratory pressure.
```

22.2 Multiple regression example

For the multiple regression we model

$$pemax = \alpha + \beta_1 age + \beta_2 height + \beta_3 weight + \epsilon$$

```
mcyst <- lm(pemax ~ age + height + weight, data=cystfibr2)  
summary(mcyst)  
  
##  
## Call:  
## lm(formula = pemax ~ age + height + weight, data = cystfibr2)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -43.675 -21.566   3.229  16.274  48.068   
##
```


Some basic statistics with R

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 64.65555   82.40935   0.785   0.441
## age         1.56755    3.14363   0.499   0.623
## height     -0.07608    0.80278  -0.095   0.925
## weight      0.86949    0.85922   1.012   0.323
##
## Residual standard error: 27.41 on 21 degrees of freedom
## Multiple R-squared:  0.4118, Adjusted R-squared:  0.3278
## F-statistic: 4.901 on 3 and 21 DF,  p-value: 0.009776

confint(mcyst)

##              2.5 %      97.5 %
## (Intercept) -106.7240711 236.035161
## age         -4.9699874   8.105082
## height      -1.7455449   1.593379
## weight      -0.9173634   2.656339
```

You should be able to interpret all output without problems.

Now, use ANOVA tables with Type I sums of squares and Type II sums of squares:

```
anova(mcyst)

## Analysis of Variance Table
##
## Response: pemax
##      Df Sum Sq Mean Sq F value    Pr(>F)
## age    1 10098.5  10098.5  13.4371 0.001441 **
## height 1   182.3    182.3   0.2426 0.627427
## weight 1   769.6    769.6   1.0240 0.323082
## Residuals 21 15782.2    751.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

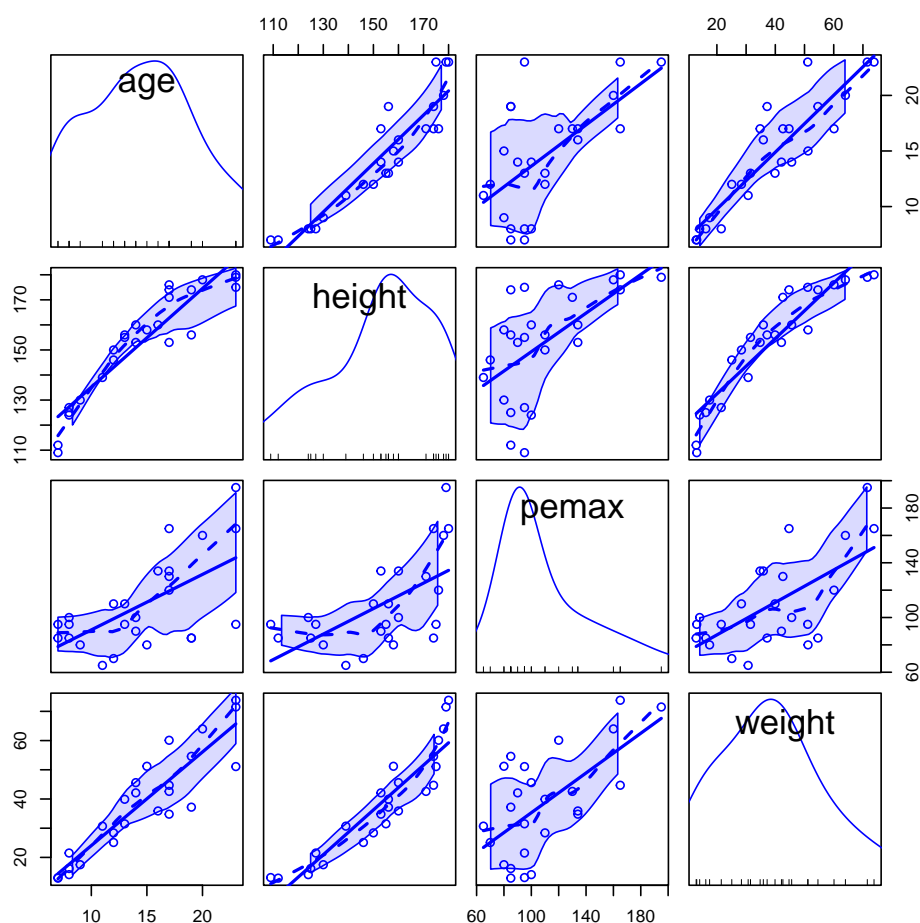
Anova(mcyst)

## Anova Table (Type II tests)
##
## Response: pemax
##      Sum Sq Df F value    Pr(>F)
## age      186.9  1  0.2486 0.6232
## height    6.8  1  0.0090 0.9254
## weight   769.6  1  1.0240 0.3231
## Residuals 15782.2 21
```

Some basic statistics with R

Are you surprised? Age seems highly significant with the sequential sums of squares (when entered first and using `anova`, so Type I) but not when we test it after all other terms in the model (`Anova`, or Type II). Why? One possible explanation is that there is correlation between explanatory variables, and that the information of age relevant for predicting pmax is already contained in the height and weight. That age, height, and weight are correlated is easy to check with a scatterplot matrix:

```
scatterplotMatrix(~ age+height+pemax+weight,
  data = cystfibr2)
```



In fact, if you refit the model and now put height first, and do a sequential test you find ... that height seems significant:

```
anova(lm(pemax ~ height + weight + age, data = cystfibr2))

## Analysis of Variance Table
##
## Response: pemax
##          Df Sum Sq Mean Sq F value    Pr(>F)
## height    1  9634.6   9634.6  12.8200 0.001763 **
## weight    1  1228.9   1228.9   1.6352 0.214935
```

Some basic statistics with R

```
## age          1   186.9   186.9   0.2486 0.623214
## Residuals 21 15782.2   751.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And similar if you place weight first.

```
anova(lm(pemax ~ weight + height + age, data = cystfibr2))

## Analysis of Variance Table
##
## Response: pemax
##           Df Sum Sq Mean Sq F value    Pr(>F)
## weight     1 10827.2 10827.2 14.4067 0.001058 **
## height     1    36.4    36.4  0.0484 0.827949
## age        1   186.9   186.9  0.2486 0.623214
## Residuals 21 15782.2   751.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Is this a problem? Well, you are trying to model pemax as a function of three variables, but those three variables are very highly correlated among themselves. Is this a common phenomenon: yes, it is rather common.

Oh, and you might have realized that we are seeing another manifestation of the problem of the order of factors we saw with ANOVA.

So, to summarize, when predictor variables are correlated:

- Even if each (or subsets of) predictor seem strongly associated to the outcome, their p-values could be large, and the sign of the coefficient reverted, when we fit all of the correlated predictors.
- The overall model (as given by the overall F statistic or the R^2) might indicate that the model is doing a decent job (certainly much better than fitting just a single mean), and yet the individual p-values might suggest that no individual predictor is relevant.

What model should we choose? We will return to some of these questions in section [“Variable and model selection” \(section 26\)](#). For now, remember that prediction and interpretation are not necessarily the same objective (and, in fact, are often at odds with each other). The model with the best predictive power is often not the model that gives the best insight into the biological (or whatever) process. (And none of these might be, either, the models that are best for causal prediction; more on this later: [“Covariate adjustment and a few comments about causal inference” \(section 29\)](#)).

22.3 R^2 and Adjusted R^2

For this class, these are the key messages:

- R^2 (R-squared) is the proportion of variability in the dependent variable accounted for (or explained) by the model. It is also the square of the correlation between observed and predicted (predicted according to the model) values of the dependent variable.
- But adding predictor variables that actually explain nothing will never decrease R^2 . The adjusted R^2 accounts for this (adding a predictor will only increase adjusted R^2 if the predictor makes some contribution to improving predictive value). Note that the (unadjusted) R^2 is the relative change in residual sums of squares, whereas the adjusted R^2 is the relative change in residual variance. In general, the adjusted R^2 is a better number to look at (and note it can, in models that explain nothing, become negative)³⁰.

The following shows the calculations but the previous summary is enough for this class (i.e., you can skip what follows).

```
summary(mcyst)

##
## Call:
## lm(formula = pemax ~ age + height + weight, data = cystfibr2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -43.675 -21.566   3.229  16.274  48.068
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  64.65555   82.40935   0.785   0.441
## age           1.56755    3.14363   0.499   0.623
## height       -0.07608    0.80278  -0.095   0.925
## weight        0.86949    0.85922   1.012   0.323
##
## Residual standard error: 27.41 on 21 degrees of freedom
## Multiple R-squared:  0.4118, Adjusted R-squared:  0.3278
## F-statistic: 4.901 on 3 and 21 DF,  p-value: 0.009776

cor(fitted(mcyst), cystfibr2$pemax)^2

## [1] 0.4118275
```

³⁰There are additional issues that might need to be considered. For example, the default R^2 that R gives you, both adjusted and unadjusted, should not be used in models without an intercept (i.e., regression through the origin). And the unadjusted R^2 has the virtue that it can be computed for many other models, since it is just the square of the correlation of predicted and observed. We will not discuss these issues here.

Some basic statistics with R

```
all.equal(
  cor(fitted(mcyst), cystfibr2$pemax)^2,
  summary(mcyst)$r.squared
)

## [1] TRUE

##
all.equal(summary(mcyst)$sigma^2,
  sum(residuals(mcyst)^2)/21)

## [1] TRUE

## and where is the (25 = 21 - 4) 4 from
21 == (length(residuals(mcyst)) - length(coefficients(mcyst)))

## [1] TRUE

all.equal(
  (var(cystfibr2$pemax) - summary(mcyst)$sigma^2)/var(cystfibr2$pemax),
  summary(mcyst)$adj.r.squared)

## [1] TRUE

## Also could see as

ssresid <- sum(residuals(mcyst)^2)
sstot  <- sum( (cystfibr2$pemax - mean(cystfibr2$pemax))^2 )

## Multiple R^2
1 - (ssresid/sstot)

## [1] 0.4118275

length(resid(mcyst))

## [1] 25

## Adjusted R^2
1 - ( (ssresid/(25 - 4))/(sstot/(25 - 1)) )

## [1] 0.3278028
```

22.4 Interactions between continuous variables

Can we add interactions? Yes, of course. Interactions between continuous variables, however, are harder to visualize: they represent curved surfaces, because the slope of one of the variable changes as the other variable changes, whereas an additive model is just a plane—or hyperplane³¹.

How do we interpret coefficients? These are a few hints (you can skip this if you want). Suppose we have

$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_{1,2} x_1 x_2 + \epsilon$ (where the term $\beta_{1,2} x_1 x_2$ is literally the product of $\beta_{1,2}$ times the product of x_1 and x_2). Now, in a table of coefficients, what is the meaning of β_1 ? That is a table that “shows every variable when everything else is also in the model” (and that includes the interaction). You can think of β_1 as just the best term that solves the above equation ($y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_{1,2} x_1 x_2 + \epsilon$). But think about the interpretation of “a coefficient tells me how fast the response changes when the predictor changes by one unit”. Let’s do that: $\frac{\partial y}{\partial x_1} = \beta_1 + \beta_{1,2} x_2$. Notice how x_2 is there: how fast y changes with x_1 is also a function of x_2 . And this differs from the model without interaction where we have: $\frac{\partial y}{\partial x_1} = \beta_1$.

And a simple numerical example (again, skip if you want):

```
mah <- lm(pemax ~ age + height, data = cystfibr2)
mahi <- lm(pemax ~ age * height, data = cystfibr2)

## Note how the coefficients are VERY different
summary(mah)

##
## Call:
## lm(formula = pemax ~ age + height, data = cystfibr2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -44.817 -17.883   3.815  18.275  53.824
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.8600    68.2493   0.262   0.796
## age          2.7178     2.9325   0.927   0.364
## height       0.3397     0.6900   0.492   0.627
##
## Residual standard error: 27.43 on 22 degrees of freedom
```

³¹If you want to play around with a regression plane or regression surfaces, go to “Graphs”, “3D graph”, “3D scatterplot”. In options, you can choose a plane or three different surfaces—you might need to play with the degrees of freedom if you get errors. You can zoom in and out with the mouse wheel and move/rotate the figure. Try doing that during your TFM defense! Of course, that allows only for up to one dependent variable and two independent ones: our brains do not seem ready for 4D and higher.

Some basic statistics with R

```
## Multiple R-squared:  0.3831, Adjusted R-squared:  0.3271
## F-statistic: 6.832 on 2 and 22 DF,  p-value: 0.00492

summary(mahi)

##
## Call:
## lm(formula = pemax ~ age * height, data = cystfibr2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -49.722  -9.579  -4.036   11.503   43.160
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  221.70208   103.62638    2.139   0.0443 *
## age         -25.00128    11.63450   -2.149   0.0435 *
## height       -0.69135     0.75216   -0.919   0.3685
## age:height    0.15376     0.06285    2.447   0.0233 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.77 on 21 degrees of freedom
## Multiple R-squared:  0.52, Adjusted R-squared:  0.4514
## F-statistic: 7.583 on 3 and 21 DF,  p-value: 0.001276

## Note that the SS of age and height are the same in both
## though the RSS in mahi is smaller.

Anova(mah)

## Anova Table (Type II tests)
##
## Response: pemax
##           Sum Sq Df F value Pr(>F)
## age           646.2  1  0.8589 0.3641
## height        182.3  1  0.2424 0.6274
## Residuals 16551.8 22

Anova(mahi, type = "II")

## Anova Table (Type II tests)
##
## Response: pemax
##           Sum Sq Df F value Pr(>F)
## age           646.2  1  1.0535 0.3164
## height        182.3  1  0.2973 0.5913
## age:height  3671.6  1  5.9863 0.0233 *
## Residuals  12880.2 21
```

Some basic statistics with R

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

## SS of height same as for type II of mah and mahi
anova(lm(lm(pemax ~ age + height, data = cystfibr2)))

## Analysis of Variance Table
##
## Response: pemax
##           Df Sum Sq Mean Sq F value    Pr(>F)
## age         1 10098.5  10098.5  13.4225 0.001365 **
## height      1   182.3    182.3   0.2424 0.627384
## Residuals  22 16551.8    752.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## SS of age same as for type II of mah and mahi
anova(lm(lm(pemax ~ height + age, data = cystfibr2)))

## Analysis of Variance Table
##
## Response: pemax
##           Df Sum Sq Mean Sq F value    Pr(>F)
## height      1  9634.6   9634.6  12.8060 0.001676 **
## age         1   646.2    646.2   0.8589 0.364108
## Residuals  22 16551.8    752.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## But the coefficients in mahi are from a model that includes the
## interaction.
## Thus, the tests of coefficients of age and height in mahi are
## not the same as the tests of age and height in the ANOVA tables (Type
## II) for models mah and mahi.
```

22.5 Confidence intervals, confidence bands

This is also provided so that you have it in here if you need it, but you can skip it.

With multiple regression, visualizing confidence intervals and prediction intervals becomes much harder (we are no longer in 2D). There are ways of visualizing 3D plots (see [Interactions between continuous variables](#)), but you will probably need to decide what exactly you want to plot and for what. For example, predictions with respect to each variable in different panels. You can easily obtain, for each observation, its predicted (fitted) value and its confidence or prediction values. For example (showing just the first five values)

Some basic statistics with R

```
predict(mcyst, interval = "confidence", level = 0.95)[1:5, ]

##           fit           lwr           upr
## 1 78.72565 47.93282 109.51847
## 2 78.32350 51.11680 105.53020
## 3 80.02144 60.35352  99.68936
## 4 81.77128 62.70768 100.83489
## 5 86.22740 66.13730 106.31751

predict(mcyst, interval = "prediction", level = 0.95)[1:5, ]

## Warning in predict.lm(mcyst, interval = "prediction", level = 0.95): predictions
on current data refer to _future_ responses

##           fit           lwr           upr
## 1 78.72565 13.93036 143.5209
## 2 78.32350 15.15361 141.4934
## 3 80.02144 19.71341 140.3295
## 4 81.77128 21.65762 141.8849
## 5 86.22740 25.78037 146.6744
```

Side note: pay attention to the warning: the predictions should not be used to assess how well the model predicts the data used for the fitting itself.

With those values, you could then construct the plots you might want.

Similar comments apply to plots of confidence intervals for the parameters: visualizing the multidimensional ellipses will not be easy. Obtaining the confidence intervals and the matrix of correlations is simple, though:

```
confint(mcyst)

##              2.5 %      97.5 %
## (Intercept) -106.7240711 236.035161
## age         -4.9699874   8.105082
## height      -1.7455449   1.593379
## weight      -0.9173634   2.656339

round(cov2cor(vcov(mcyst)), 3)

##              (Intercept)    age height weight
## (Intercept)      1.000  0.421 -0.976  0.561
## age              0.421  1.000 -0.557 -0.362
## height           -0.976 -0.557  1.000 -0.512
## weight           0.561 -0.362 -0.512  1.000
```

(Note: we already knew that mcyst is probably not a great model!!)

Some basic statistics with R

22.5.1 Confidence intervals, confidence bands: the bootstrap

A final comment: for real, one might want to use confidence intervals using the bootstrap instead of `confint`. See section 5.1.3 in Fox and Weisberg's "An R companion to applied regression, 3rd ed." or section 3.6 in Faraway's "Linear models with R, 2nd ed."

22.6 Three way anovas, factors with more than two classes, etc

There is nothing conceptually new, but the accounting gets more complicated.

23 Continuous and discrete independent variables and ANCOVA

23.1 The general scenario for ANCOVA

Right now, nothing should stop us from thinking about models where the right hand side contains both continuous and discrete variables. ANCOVA refers to this mixture of ANOVA and regression and stands for “Analysis of covariance”³². Regardless, all of ANOVA, regression, and ANCOVA are especial types of linear models.

Suppose you have a response variable, Y . And two predictor variables; variable A , which is discrete, and has, say, two levels (a_1 , a_2). And variable X , which is continuous. Now, get a piece of paper and draw the following cases, in turn. Really, **do draw this**.

- The relationship between Y and X increases faster in a_1 than in a_2 . Thus, the regression lines for a_1 and a_2 are not parallel (and will eventually cross each other).
- The relationships between Y and X change at the same rate for a_1 and a_2 . Thus, they are parallel lines. But individuals of group a_1 with value $X = x$ have a larger value of Y than individual of group a_2 for that value of X . So, as we said, lines are parallel, but they are separate. They have different intercepts.
- As the previous case, but now the intercept is the same. So you only see one line.
- There is not relationship between Y and X in any of the groups. (Or, if you insisted on putting a line, it would be of slope 0).

ANCOVA is a procedure for differentiating the above scenarios, starting from the first and moving down. It also works when the discrete group has more than two levels. And, for that matter, we can deal with more than one discrete variable and more than one continuous one, etc, etc: these are just linear models. As said several times already, we will use the name ANCOVA as this might be familiar. But once you understand the process, these are just linear models that allow to flexibly model scientific problems even if they no longer fit in the simple ANCOVA framework.

Oh, and if you want some specific names for variables to draw the above figures, you can do as follows:

- Y is alertness, A is sex (male, female), X is amount of coffee.
- Y is metabolic rate, A is tetrapod group (bird, reptile, amphibian, mammal), X is body mass.
- Y is number of somatic mutations, A is type of tissue, X is age.

³²But this does not mean that we are comparing covariances, as in comparing correlations, between groups; we are comparing groups after adjusting for possible covariates, if that is warranted by the absence of interactions between the continuous and discrete predictors.

Some basic statistics with R

- etc, etc.

23.2 A first example of ANCOVA with the cystfibr data

Let's fit an ANCOVA with the cystic fibrosis data set. The independent variables are sex (discrete, obviously) and age. However, sex is coded with 0/1 and we want it to be a factor, explicitly. Let's recode it:

```
cystfibr2$sex <- factor(cystfibr2$sex, labels = c('Male', 'Female'))
```

Now, fit a model.

```
mcyst2 <- lm(pemax ~ age * sex, data = cystfibr2)
summary(mcyst2)

##
## Call:
## lm(formula = pemax ~ age * sex, data = cystfibr2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -54.901 -12.447   5.069  15.099  45.099
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    54.185     20.806   2.604  0.01656 *
## age             4.162       1.281   3.249  0.00384 **
## sexFemale       5.683     37.968   0.150  0.88243
## age:sexFemale  -1.313       2.602  -0.505  0.61911
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.25 on 21 degrees of freedom
## Multiple R-squared:  0.419, Adjusted R-squared:  0.336
## F-statistic: 5.048 on 3 and 21 DF, p-value: 0.008655

confint(mcyst2)

##              2.5 %    97.5 %
## (Intercept)  10.916180 97.454261
## age          1.497428  6.825642
## sexFemale    -73.274310 84.641307
## age:sexFemale -6.724126  4.098292

Anova(mcyst2)

## Anova Table (Type II tests)
##
```

Some basic statistics with R

```
## Response: pemax
##           Sum Sq Df F value    Pr(>F)
## age       8819.5  1 11.8802 0.002417 **
## sex        955.4  1  1.2870 0.269386
## age:sex    189.0  1  0.2546 0.619111
## Residuals 15589.7 21
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

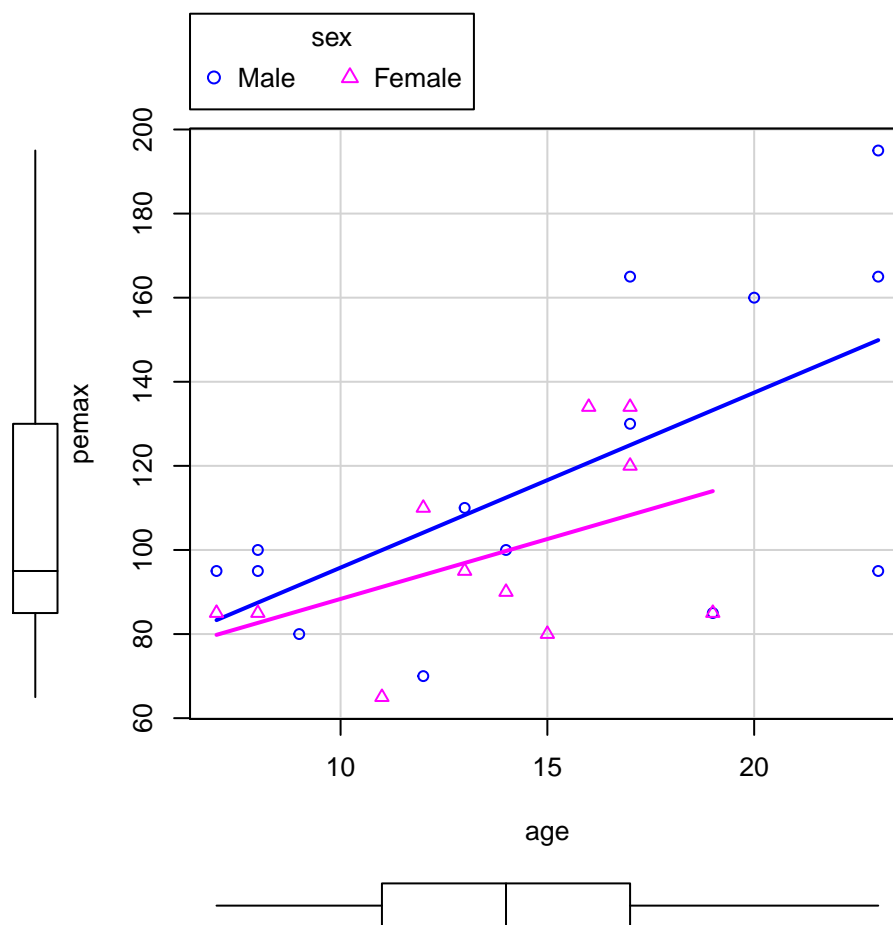
Note we added a “*”, so an interaction between a continuous and a discrete variable. Here there is no evidence of interaction.

Back to the interactions. What would an interaction have looked like? Different slopes for each group.

Look at the plots:

```
scatterplot(pemax~age | sex,
            boxplots='xy',
            smooth = FALSE,
            by.groups=TRUE,
            data=cystfibr2)
```

Some basic statistics with R



(yes, the best fitting slopes are slightly different, but they are not significantly different, as shown by the “age:sex[T.Female]” term in the model above, so no evidence for different slopes).

The different intercepts are captured by the term “sex[T.Female]” (that is not significant in this example either). Anyway, we can often have models where we have no evidence of different slopes (no interaction), but evidence of different intercepts: these means parallel lines (we will see one below: [ANCOVA with the birds and the reptiles](#)).

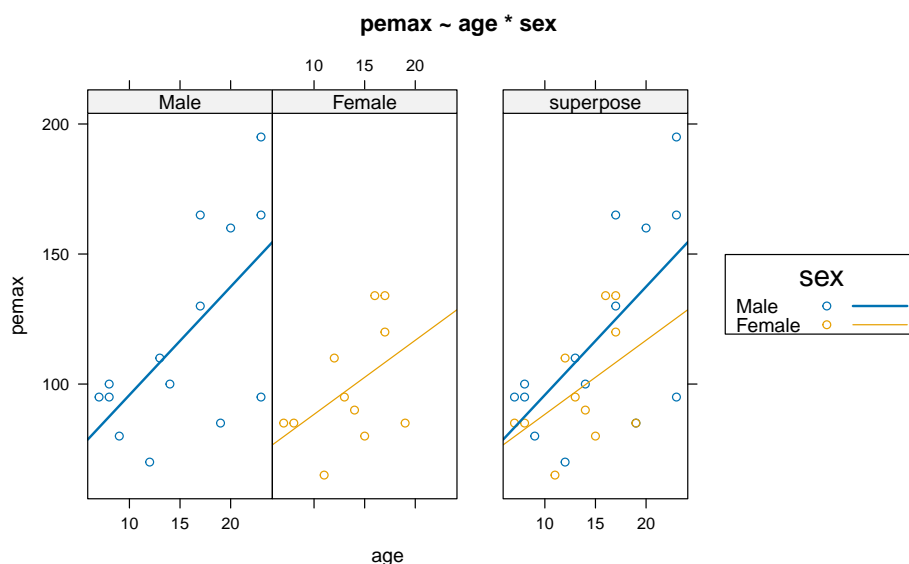
One can visualize this also with the function `ancova` in package *HH*, so we do not need to load it again); `ancova` also produces an anova table (with sequential sums of squares, Type I):

```
ancova(pemax ~ age * sex, data = cystfibr2)

## Analysis of Variance Table
##
## Response: pemax
##          Df Sum Sq Mean Sq F value    Pr(>F)
## age       1 10098.5  10098.5   13.6031 0.001366 **
```

Some basic statistics with R

```
## sex      1  955.4  955.4  1.2870 0.269386
## age:sex   1  189.0  189.0  0.2546 0.619111
## Residuals 21 15589.7  742.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



Note that we are using `ancova` just to produce plots. The analysis are directly available using `lm`, `anova`, etc.

And, of course, do not forget to look at the output from `confint`. As you see, all the information (summary of the model, confidence intervals, figures) points in the same direction.

One final thing to notice here: at the beginning of this section we have used “Anova” and “summary(lm)” and the p-values for age and sex are not identical³³. Why? Because Type II sums of squares (the default of `Anova`) respect the marginality principle: age and sex are tested after each other, but **not** after the interaction is in the model (this is in contrast to what happens in the `summary(lm)` output). The marginality principle and what Type II Sums of Squares do applies not only to Ancova, but to linear models in general³⁴.

Finally, can you tell if `ancova` is using sequential or Type II sums of squares? If you use `ancova` and change the order of age and sex in the specification of the model, does the numeric output differ?

³³Neither is it the case that the sqrt of the F for those terms is identical to t statistics, which is a property of the relationship between t statistics and F statistics with one degree of freedom in the numerator

³⁴Type III Sums of Squares, which you can ask for when using `Anova` would give, for p-values, output identical to the one of `lm` even if there is an interaction term in the model —barring possible differences due to the contrasts used, if we have factors; we will not get into any of this.

23.3 A parallel slopes model

We could have started by fitting a model with parallel slopes:

```
mcyst0 <- lm(pemax ~ age + sex, data = cystfibr2)
summary(mcyst0)

##
## Call:
## lm(formula = pemax ~ age + sex, data = cystfibr2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -52.423 -13.617   5.637  17.485  47.577
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   59.027     18.146   3.253  0.00365 **
## age           3.843       1.096   3.507  0.00199 **
## sexFemale    -12.632     10.944  -1.154  0.26081
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.78 on 22 degrees of freedom
## Multiple R-squared:  0.412, Adjusted R-squared:  0.3585
## F-statistic: 7.706 on 2 and 22 DF,  p-value: 0.002907

confint(mcyst0)

##              2.5 %    97.5 %
## (Intercept) 21.394568 96.659396
## age         1.570351  6.116243
## sexFemale   -35.328603 10.065321

Anova(mcyst0)

## Anova Table (Type II tests)
##
## Response: pemax
##              Sum Sq Df F value    Pr(>F)
## age         8819.5  1 12.2969 0.001992 **
## sex          955.4  1  1.3321 0.260810
## Residuals 15778.7 22
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Make sure you understand the differences with respect to the previous model. What are we fitting here? What are we saying, biologically, in this model?

23.4 Formally comparing models

In this case, the sequential anova table (as produced by `ancova` or `anova`) suggests that we could simplify our model a lot, and use one with a single intercept and slope (i.e., just like a simple linear regression as in section [Simple linear regression](#)) as neither “sex” by itself nor the interaction is relevant. We will do that, and we will then do a global model comparison to verify the simplified model is a reasonable one:

```
mcyst3 <- lm(pemax ~ age, data = cystfibr2)
anova(mcyst3, mcyst2)

## Analysis of Variance Table
##
## Model 1: pemax ~ age
## Model 2: pemax ~ age * sex
##   Res.Df  RSS Df Sum of Sq    F Pr(>F)
## 1      23 16734
## 2      21 15590  2    1144.4 0.7708 0.4753
```

This is a comparison of two models using an F test: it tests whether the larger (more complex, with more terms) model is significantly better than the smaller one. It clearly shows that, in this case, the larger model (the one with both a main effect of “sex” and an interaction) is not significantly better than the one without “sex”. So we can just keep model `mcyst3`: there is no statistical evidence of `mcyst2` being any better. Again, notice how we have compared two models that, in this case, differed by the presence of several terms: the small model had only age, the large had sex and the interaction of sex and age. Being able to compare (nested —see below) models that differ in several terms at once is really handy and there is no need to proceed by dropping one term at a time.

Three additional comments here:

- These tests only make sense for nested models (where the terms of one of the models is a subset of terms of the other). Beware that R does not check this, and you could easily do meaningless things³⁵.
- Whether you type `anova(mcyst2, mcyst3)` or `anova(mcyst3, mcyst2)` is inconsequential for the F statistic and p-values.
- This was a very clear-cut case. Often, people will proceed in steps: first check no interaction and then, later, and if no interaction, check if there is a need for different intercepts.

In fact, we could have done this in a more step-by-step way:

```
anova(mcyst0, mcyst2)

## Analysis of Variance Table
```

³⁵There are ways to compare non-nested models using other procedures, for instance based on AIC.

Some basic statistics with R

```
##  
## Model 1: pemax ~ age + sex  
## Model 2: pemax ~ age * sex  
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)  
## 1      22 15779  
## 2      21 15590   1      189 0.2546 0.6191
```

where we compare the model and without interaction (though this is the same, of course, as the term for interaction in `Anova(mcyst2)`).

And then

```
anova(mcyst3, mcyst0)  
  
## Analysis of Variance Table  
##  
## Model 1: pemax ~ age  
## Model 2: pemax ~ age + sex  
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)  
## 1      23 16734  
## 2      22 15779   1    955.43 1.3321 0.2608
```

but, again, this is just the same as the term for `sex` in `Anova(mcyst0)`.

23.5 ANCOVA with the birds and the reptiles

We will use the longevity and metabolic rate data we used in section [Simple linear regression](#), but now the full one: `anage_a_r`. We will go pretty fast here (this is just a kind of review), and will examine two things:

- If the relationship between metabolic rate and body mass is different between reptiles and birds.
- If the relationship between longevity and body mass is different between reptiles and birds.

Beware: as we said before, what we are going to do is not correct, as the data are not independent (species share common ancestors, and they are related in varying degrees, as any phylogenetic tree would show you, and as you should be able to tell from looking at the names of some species). What we are doing here is just for the sake of the example, and because this is a nice set of data³⁶.

³⁶This can be done correctly, incorporating phylogenetic information in the regression model, but this is way out of the scope of this class. It is a really fascinating topic, though! This is often referred to as using the comparative method in evolutionary biology.

Some basic statistics with R

We will see interactions, parallel and non-parallel slopes, and more comparison of models. Again, these analyses are not fully correct as we ignore phylogenetic relatedness. But they are nice to illustrate a couple of points. We will directly use log transformations (again, theory and previous empirical evidence indicate this is the way to go —and we looked at a couple of different models already).

First, metabolic rate vs. body mass allowing for interaction with “Class” (bird vs. reptile):

```
metab_b_r <- lm(logMetabolicRate ~ logBodyMass * Class, data = anage_a_r)
summary(metab_b_r)

##
## Call:
## lm(formula = logMetabolicRate ~ logBodyMass * Class, data = anage_a_r)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26958 -0.14488  0.01647  0.18083  0.62902
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.15949    0.05366  -58.88  <2e-16 ***
## logBodyMass      0.65037    0.01201   54.17  <2e-16 ***
## ClassReptilia   -2.93984    0.25722  -11.43  <2e-16 ***
## logBodyMass:ClassReptilia -0.04577    0.04488   -1.02    0.309
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2688 on 174 degrees of freedom
## (1548 observations deleted due to missingness)
## Multiple R-squared:  0.9576, Adjusted R-squared:  0.9569
## F-statistic: 1310 on 3 and 174 DF, p-value: < 2.2e-16

confint(metab_b_r)

##              2.5 %      97.5 %
## (Intercept) -3.2653954 -3.05358006
## logBodyMass  0.6266718  0.67406317
## ClassReptilia -3.4475184 -2.43216507
## logBodyMass:ClassReptilia -0.1343460  0.04279745
```

The output is clear: parallel lines (i.e., different intercepts, but same slope). Note that this is not a silly or irrelevant biological detail: metabolic rates scales with body mass in the same way in an endothermic group (birds) and a ectothermic one (reptiles), but their metabolic rates are not the same (birds' are higher).

Some basic statistics with R

We could simplify this model to a model without the interaction (so single slope, two intercepts):

```
metab_b_r_2 <- lm(logMetabolicRate ~ logBodyMass + Class, data = anage_a_r)
summary(metab_b_r_2)

##
## Call:
## lm(formula = logMetabolicRate ~ logBodyMass + Class, data = anage_a_r)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.28901 -0.14756  0.01835  0.18542  0.63129
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.14600    0.05201  -60.49  <2e-16 ***
## logBodyMass   0.64709    0.01157   55.93  <2e-16 ***
## ClassReptilia -3.18852    0.08201  -38.88  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2688 on 175 degrees of freedom
## (1548 observations deleted due to missingness)
## Multiple R-squared:  0.9573, Adjusted R-squared:  0.9569
## F-statistic: 1964 on 2 and 175 DF, p-value: < 2.2e-16

confint(metab_b_r_2)

##              2.5 %    97.5 %
## (Intercept) -3.2486446 -3.043349
## logBodyMass  0.6242576  0.669925
## ClassReptilia -3.3503788 -3.026665

anova(metab_b_r_2, metab_b_r)

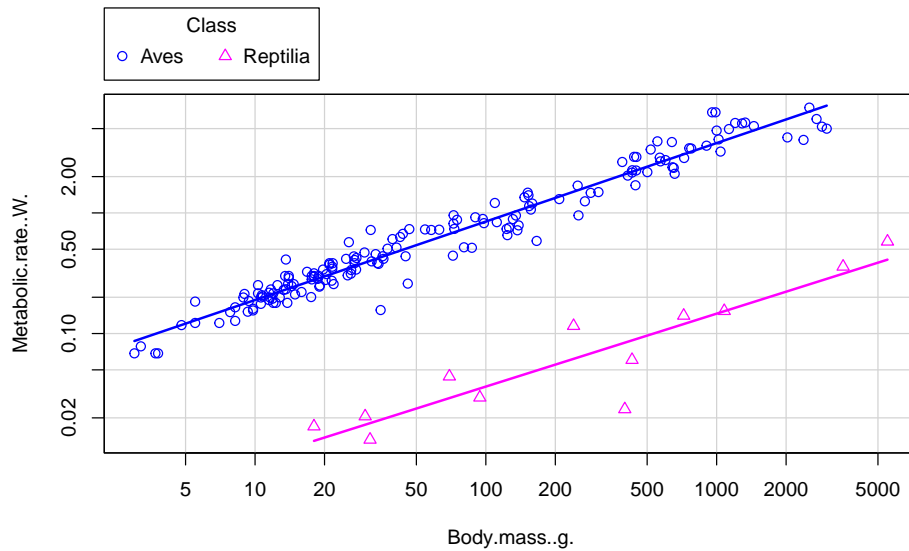
## Analysis of Variance Table
##
## Model 1: logMetabolicRate ~ logBodyMass + Class
## Model 2: logMetabolicRate ~ logBodyMass * Class
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      175 12.646
## 2      174 12.571  1  0.075167 1.0404 0.3091
```

(of course, the model comparison via `anova` here is really unneeded: we know what the p-value and F values should be, since we are only removing the interaction, for which we already saw a test).

Let's see the plots:

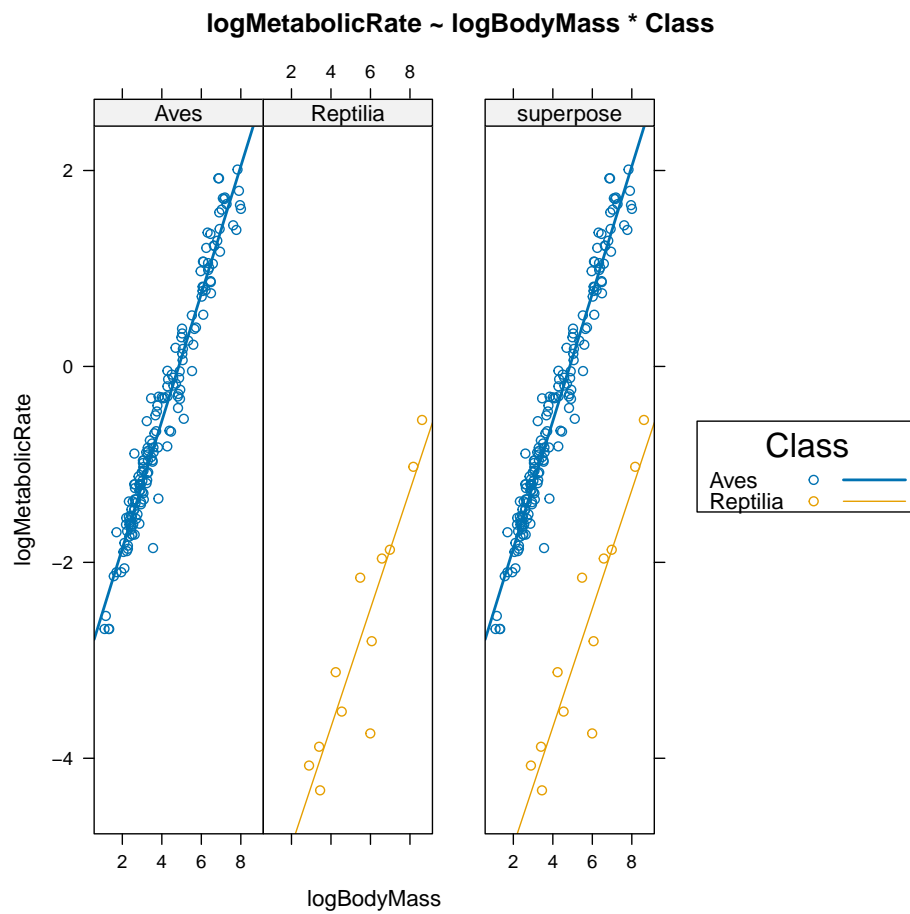
Some basic statistics with R

```
scatterplot(Metabolic.rate..W.~Body.mass..g. | Class,  
            log="xy", smooth=FALSE,  
            by.groups=TRUE,  
            data=anage_a_r)
```



Some basic statistics with R

```
ancova(logMetabolicRate ~ logBodyMass * Class,  
       data = anage_a_r)
```



Some basic statistics with R

What about longevity?

```
longev_b_r <- lm(logLongevity ~ logBodyMass * Class, data = anage_a_r)
summary(longev_b_r)

##
## Call:
## lm(formula = logLongevity ~ logBodyMass * Class, data = anage_a_r)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49667 -0.21790  0.01212  0.20800  0.91414
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.78882    0.08159   21.923 < 2e-16 ***
## logBodyMass      0.21821    0.01820   11.991 < 2e-16 ***
## ClassReptilia    -0.98582    0.42928  -2.296  0.02289 *
## logBodyMass:ClassReptilia 0.25611    0.08052   3.181  0.00175 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4028 on 168 degrees of freedom
## (1554 observations deleted due to missingness)
## Multiple R-squared:  0.5402, Adjusted R-squared:  0.532
## F-statistic: 65.8 on 3 and 168 DF, p-value: < 2.2e-16

confint(longev_b_r)

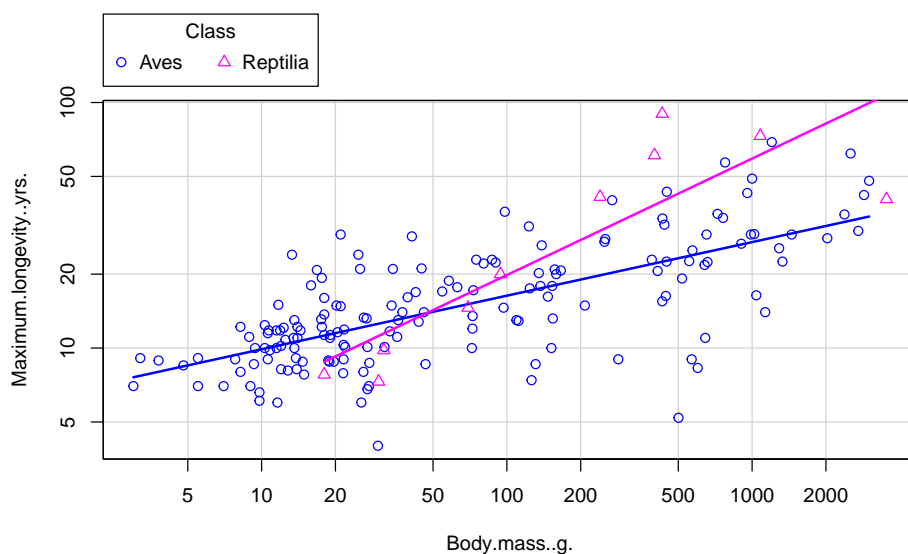
##              2.5 %      97.5 %
## (Intercept)  1.62773582  1.9498983
## logBodyMass  0.18228304  0.2541336
## ClassReptilia -1.83329842 -0.1383351
## logBodyMass:ClassReptilia 0.09714182  0.4150738
```

In this case, lines are not parallel: the rate of change of longevity with body mass is faster in reptiles than in birds (note the coefficient “logBodyMass:Class[T.Reptilia]”).

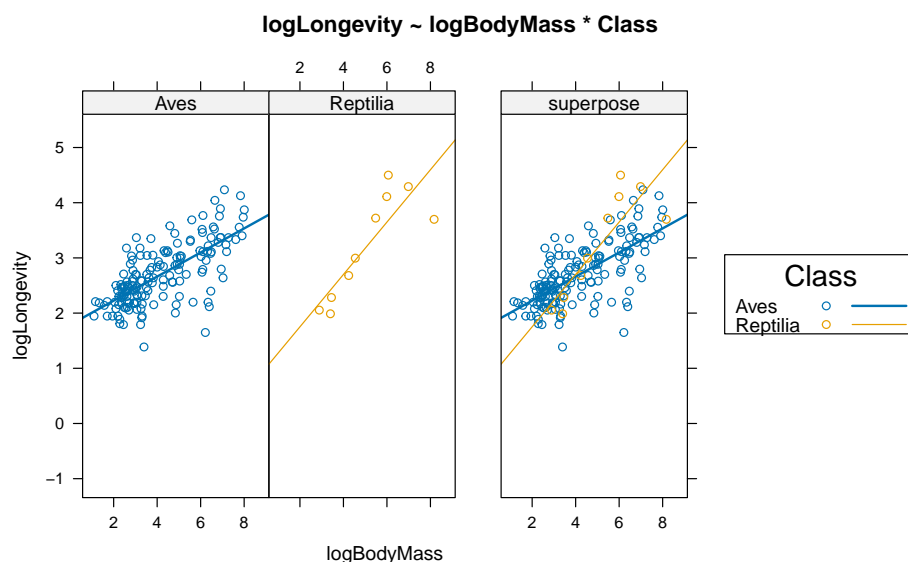
How do things look?

```
scatterplot(Maximum.longevity..yrs.~Body.mass..g. | Class,
            log="xy", smooth=FALSE,
            by.groups=TRUE,
            data=anage_a_r)
```

Some basic statistics with R



```
ancova(logLongevity ~ logBodyMass * Class, data = anage_a_r)
```



Note: the tightness of the data around the lines in this model for longevity is not nearly as good as for metabolic rate, which probably does make biological sense.

If we were to remove “Class” and refit a simpler model we would see that the larger model is clearly better when using `anova` to compare the two models:

```
longev_b_r_2 <- lm(logLongevity ~ logBodyMass, data = anage_a_r)
anova(longev_b_r_2, longev_b_r)

## Analysis of Variance Table
##
## Model 1: logLongevity ~ logBodyMass
## Model 2: logLongevity ~ logBodyMass * Class
```


Some basic statistics with R

```
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1     170 29.806
## 2     168 27.258  2    2.5477 7.8512 0.00055 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

23.6 More examples

Section 12.7 of Dalgaard's “Introductory statistics with R” contains a beautiful and detailed example of ANCOVA, including transformation of variables, using the `hel` dataset included in *ISwR*.

To discuss in class: suppose you had four groups (call it Factor4) and one continuous variable (call it X). How many lines would you see in the figures? What would the ANOVA table(s) look like? (Think about rows and degrees of freedom).

23.7 More variables

We can extend the models to incorporate more variables, add interactions, etc. Interactions can involve more than two variables, can involve continuous and discrete variables, etc, etc.

23.8 Parameters, coefficients

There is nothing conceptually new here. If you want the details, go to section [Anova tables from `lm` et al.: understanding the coefficients and parameters](#) and make sure things make sense with these new models: how we interpret coefficients and how many parameters we have.

24 Interactions, summary

The general pattern is always the same: the effect of one independent variable (say, A) depends on the setting of the other independent variable (say, B) with which it interacts. In other words, to say something about how a change of variable A affects the outcome, you need to also know the setting or value of variable B.

The three main types are:

Between factors There is one level for each combination of the factors, as in the example in section [Interactions](#).

Between a factor and a continuous variable What we saw in ANCOVA, in section [Continuous and discrete independent variables and ANCOVA](#): a different slope for each group. (Parallel slopes is not an interaction).

Some basic statistics with R

Between two continuous variables We mentioned this in section [Interactions between continuous variables](#): slope changes as we move the other variable which gives curved surfaces.

25 Diagnostics

25.1 Model diagnostics: why, how

Wait!!! Do our models make sense? These are models, so we can, and should, check some of their basic assumptions. We cannot do justice to this **very important topic**.

In general, for linear models (ANOVAs, regressions, etc) we want to check:

- Constant variance (across groups or over the range of the independent variables). Often referred to as “homoscedasticity” (where “heteroscedasticity” is the opposite).
- Linearity (for regression).
- Approximate normality of residuals.
- Possible highly influential points (i.e., do our results depend on one or two points that are driving the model one way or another?).
- Possible outliers.

Independence is also a crucial assumption. But often, checking independence is very difficult from the data themselves (or at least from the data we have been using, anyway). For example, lack of independence among data is the reason why the analysis with the AnAge data set are not really correct.

Before looking at the plots, two concepts should be clear:

Fitted value The predicted, or fitted value: if we have an equation like $Y = \alpha + \beta_1 X + \beta_2 Z$, then the fitted values are the \hat{Y} for the observed combinations of X and Z (with the values of α and β returned from the model). It is just what the model predicts.

Residual Basically, the difference between the observed and the fitted value. There are different types of residuals (residuals, standardized and studentized being the most common).

25.2 Diagnostics: an example with a designed experiment with factors

We will first use the fake data from a perfectly balanced experimental design, the data used in section [Does order always matter?](#). The diagnostic plots from these kinds of designed experiments can look slightly different from those for regression, which makes sense if you think about it. I will recreate the data here. However, to make things more interesting, I will create a very large outlier:

```
## Create the data
set.seed(1)
sex <- factor(rep(c("Male", "Female"), c(20, 20)))
```

Some basic statistics with R

```
drug <- factor(rep(rep(c("A", "B"), c(10, 10)), 2))
y <- rep(c(10, 13, 12, 16), rep(10, 4))
y <- y + rnorm(length(y), sd = 1.5)
y.data <- data.frame(y, sex, drug)
y.data[1, 1] <- 25
## Fit the model
myAdditive <- lm(y ~ sex + drug, data = y.data)
myInteract <- lm(y ~ sex * drug, data = y.data)
## What are they saying?
summary(myAdditive)

##
## Call:
## lm(formula = y ~ sex + drug, data = y.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.3955 -1.1085 -0.1430  0.4823 13.9079
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.4996     0.7523   16.615 < 2e-16 ***
## sexMale      -1.4075     0.8687   -1.620  0.11368
## drugB         2.9813     0.8687    3.432  0.00149 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.747 on 37 degrees of freedom
## Multiple R-squared:  0.2802, Adjusted R-squared:  0.2413
## F-statistic: 7.201 on 2 and 37 DF, p-value: 0.002283

summary(myInteract)

##
## Call:
## lm(formula = y ~ sex * drug, data = y.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.6953 -1.1203 -0.2659  0.8848 13.2077
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.799490   0.849216   13.895  4.9e-16 ***
## sexMale      -0.007218   1.200973   -0.006  0.995238
## drugB         4.381605   1.200973    3.648  0.000829 ***
```

Some basic statistics with R

```
## sexMale:drugB -2.800610 1.698433 -1.649 0.107861
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.685 on 36 degrees of freedom
## Multiple R-squared:  0.3307, Adjusted R-squared:  0.275
## F-statistic: 5.93 on 3 and 36 DF, p-value: 0.002143
```

```
oldpar <- par(oma=c(0,0,3,0), mfrow=c(2,2))
plot(myAdditive)
par(oldpar)
```

$\text{lm}(y \sim \text{sex} + \text{drug})$

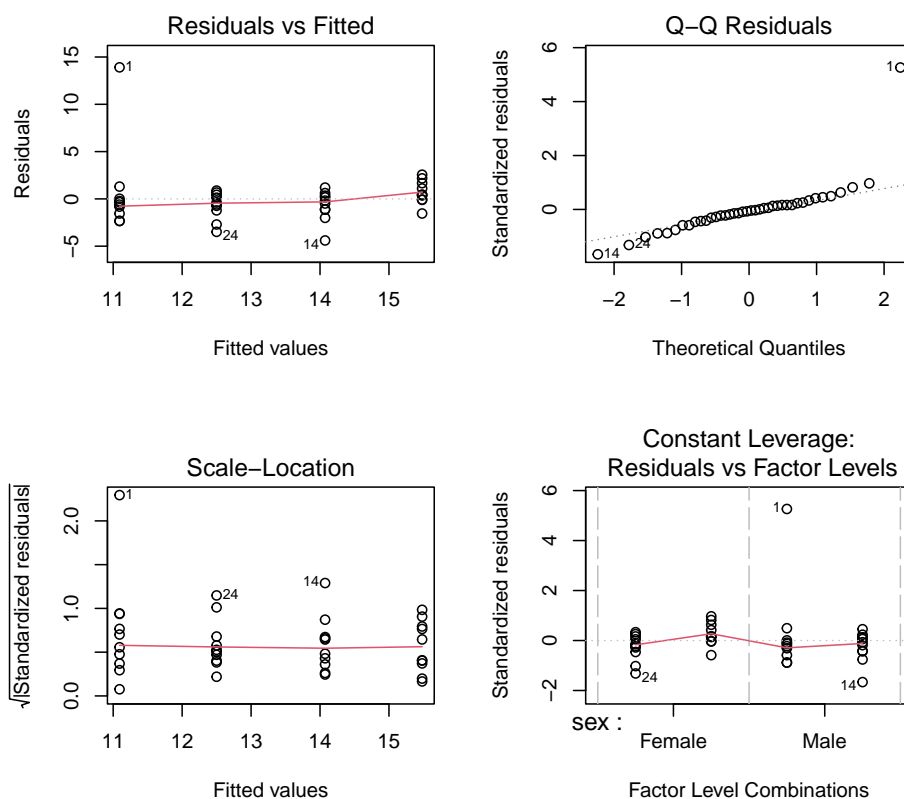


Figure 3 – Model diagnostics for designed experiment, additive model

```
oldpar <- par(oma=c(0,0,3,0), mfrow=c(2,2))
plot(myInteract)
par(oldpar)
```

First, look at the plots. Try to see what they are about. See how there is one point that stands out in several plots, look at the regularity of the patterns.

Some basic statistics with R

$\text{lm}(y \sim \text{sex} * \text{drug})$

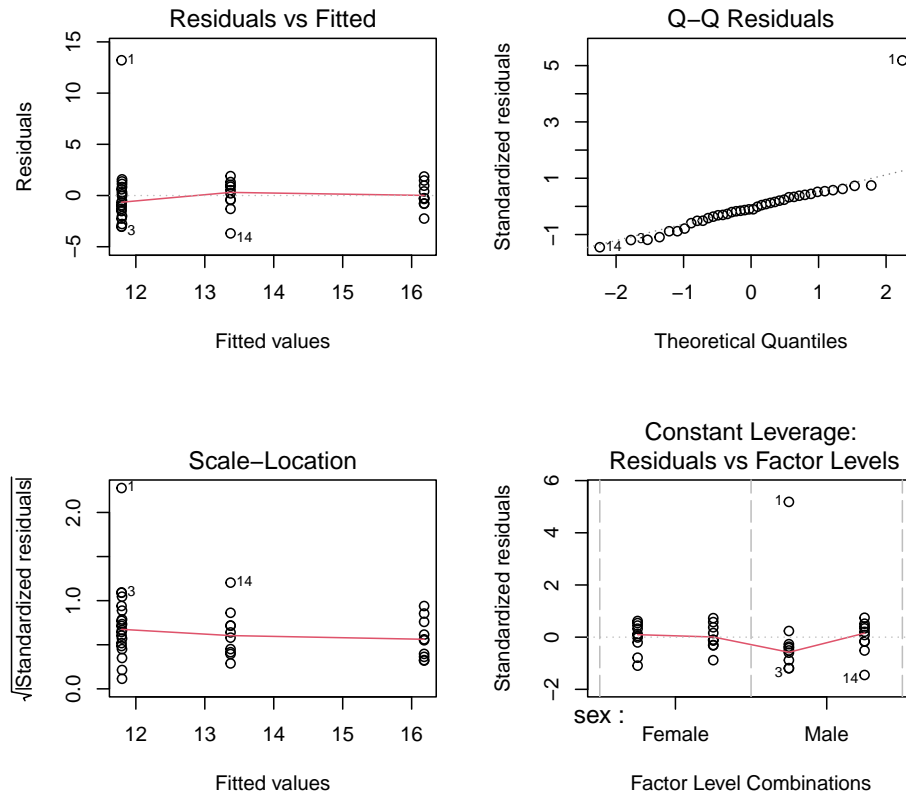


Figure 4 – Model diagnostics for designed experiment, interaction model

The plot on the upper left is used to judge if the functional form of the model makes sense, especially for regression models (not so much for designed experiments with factors, but it is still helpful). This figure also helps spots systematic changes in variance (i.e., violations of homoscedasticity). But for this, the bottom left plot is better which, in this case, does not suggest anything serious, except for the outlier.

The upper right plot (a “q-q plot”) is used to assess approximate normality of the residuals: you want points to more or less lie along the dotted line (with some allowance for deviations in the tails). Here the “q-q plot” is not perfect (even if we discount the outlier), even when data did come from a normal; this is totally normal (remember, we are sampling).

The bottom right differs in regression models and experiments with factors. Here you are shown residuals vs. factor level combinations. (The idea of leverage that makes a lot of sense in regression —see below— is not that useful here). Notice how this plot shows four vertical lines in both models, the additive and the interaction one, but plots on the upper left and bottom left differ in the apparent number of vertical lines. Do you understand why?

Some basic statistics with R

An issue about syntax: if you want the plots to show, in the upper part of the figure, the actual model fitted, you might need to increase the margins. That is what I do with the `par(oma = c(0, 0, 3, 0))` (I actually do that inside the call to also allow to plot several figures at once: `par(oma=c(0,0,3,0), mfrow=c(2,2))`)

25.3 Diagnostics: examples with some of the regression models

We will now use the two simple regression models we fitted in section [Simple linear regression](#). Make the first one (`metab`) active and go to “Models”, “Graphs”, “Basic diagnostic plots”.

```
oldpar <- par(oma=c(0,0,3,0), mfrow=c(2,2))
plot(metab)
par(oldpar)
```

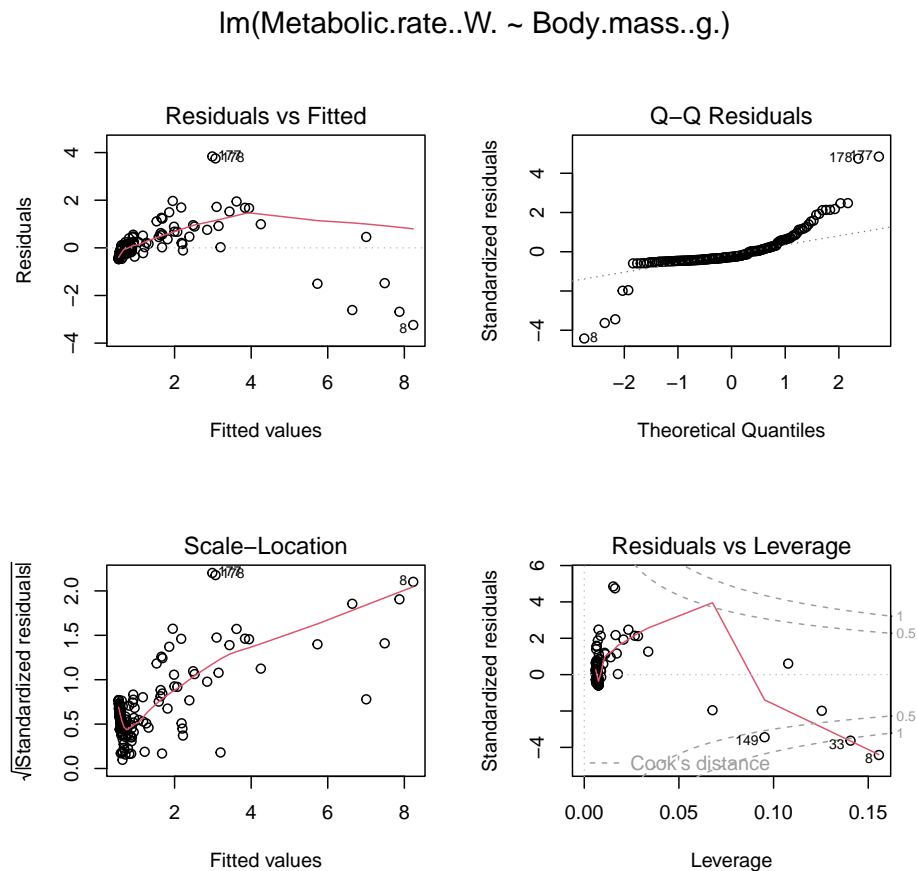


Figure 5 – Model diagnostics for metabolic rate model without log transformation

Now you can see why with regression models we can use the plot on the upper left to judge if the functional form of the model makes sense: if the relationship is really linear as modeled, you should see no systematic pattern here, but we see it (in this case, it suggests that our model is predicting too small a metabolic rate at intermediate values, and the opposite at large values, which suggests a curvilinear relationship). Again, this figure also helps spot systematic changes in variance (i.e., violations of homoscedasticity). But for this, the bottom left plot is better and it does suggest that violations of homoscedasticity are present (though, right now, with such strong evidence of non-linearity, this is not a surprise).

Some basic statistics with R

What about the “q-q plot”: things do not look great here, and this distribution has several very large residuals, is heavy tailed, and also somewhat skewed.

The bottom right can be hard to interpret: it shows two quantities (residuals and leverage) that, together, are part of Cook's distance. Cook's distance measures the effect of individual points on the fitted coefficients (values of Cook's distance larger than 1 usually indicate a possibly very influential point, but any point with a widely outstanding Cook's distance deserves a closer look). The plot called “Influence plots” is very similar to this one. However, I often find it simpler to look at plots Cook's distance (function `cooks.distance`).

Now, repeat the above with the model that has taken logs:

```
oldpar <- par(oma=c(0,0,3,0), mfrow=c(2,2))
plot(metablog)
par(oldpar)
```

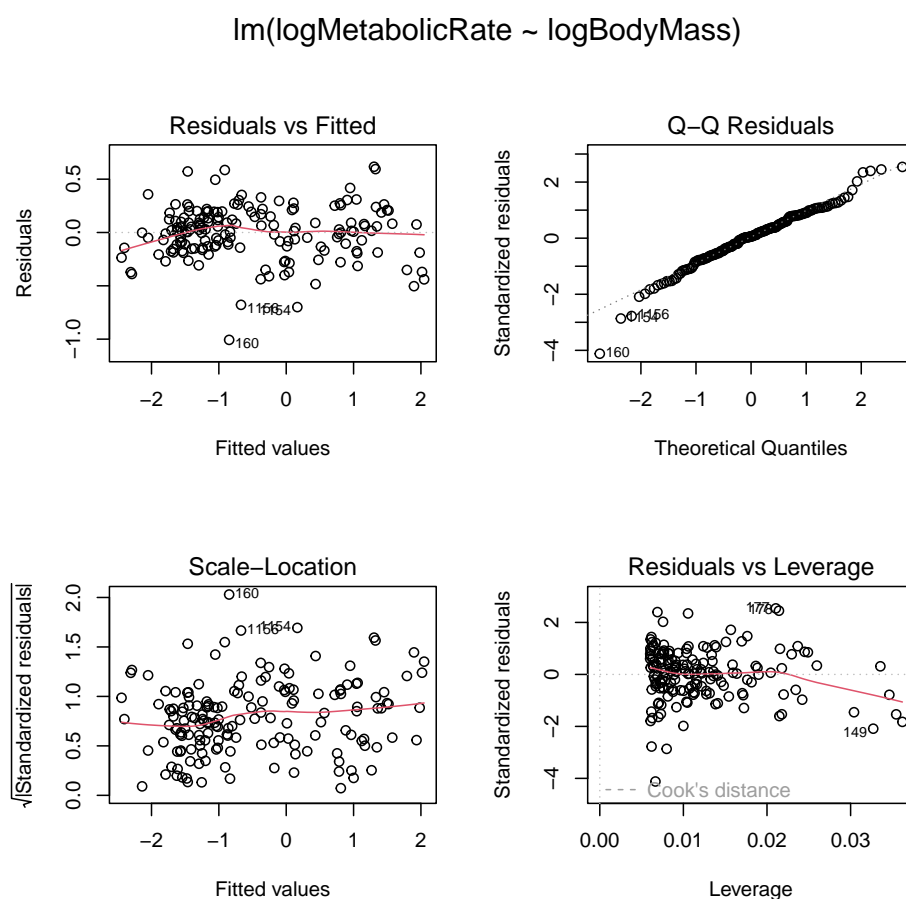


Figure 6 – Model diagnostics for metabolic rate model after log transformation

and you will see that all diagnostics look much better.

Some basic statistics with R

You should also take a look at the diagnostics for some of the other models we have fitted, specifically `longev_b_r` and `metab_b_r` (or `metab_b_r_2`) (section [ANCOVA with the birds and the reptiles](#)). The metabolism one are OK, but the longevity model is not fully satisfactory (small problems in all diagnostic plots, and one potentially influential value). `mcyst2` (section [Continuous and discrete independent variables and ANCOVA](#)) and `mcyst` (section [Multiple regression](#)), the two models we fitted to the cystic fibrosis data, both look relatively decent, although there could be some concerns about increases in variance with fitted values. However, it is not easy to tell, because of the relatively few points. This, of course, makes sense: if there are few points, we will only be able to detect if a model is a bad one if it really is very bad.

`cholestanova`, from section [Two-way ANOVA](#), looks relatively OK (remember, this is an ANOVA, and there were six combinations of levels, and thus the discrete values you observe in two of the plots). However, the `qqplot`³⁷ of residuals is a little bit ugly, but it is hard to tell from relatively so little data³⁸. Finally, `AnovaModel.1`, from section [An ANOVA; some basic theory and output](#), looks just fine. Please look at all of these yourself to see them.

³⁷“`qqplot`”: “quantile-quantile plot”

³⁸Again, this is an interesting case, since these data are simulated from a normal distribution

25.4 Diagnostics: more examples with regression and ANCOVA models

$\text{lm}(\text{tumor.size} \sim \text{exposure} * \text{drug})$

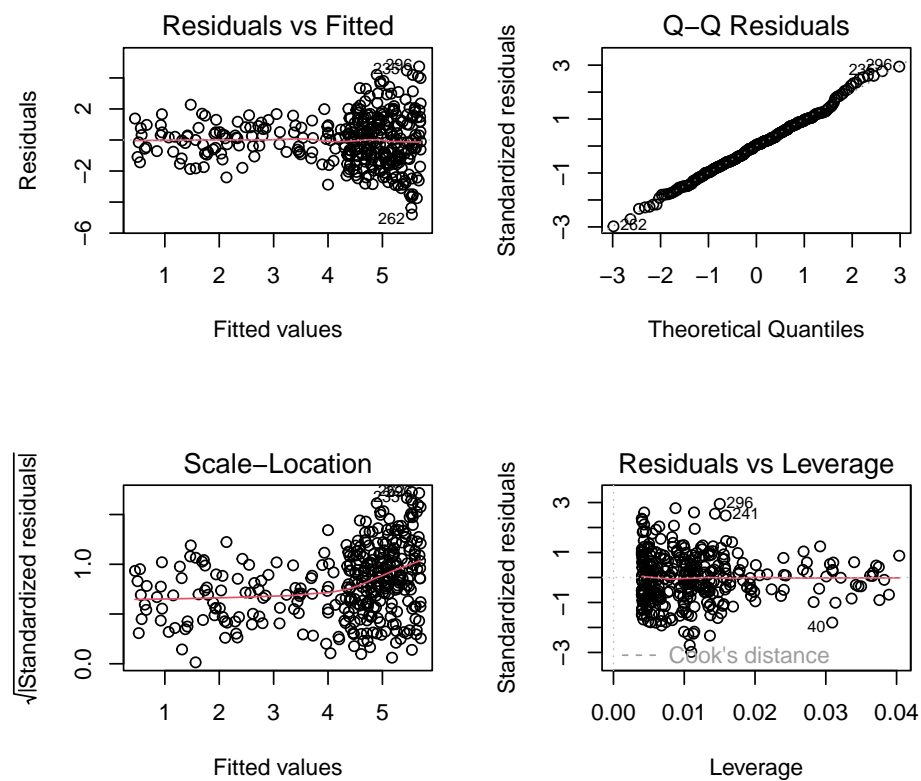


Figure 7 – Non-constant variance in ANCOVA model

Some basic statistics with R

Some plots to try to understand the patterns in the last residual plots. These are just some initial steps, the sort of thing I'd do if I found the above patterns; what could be happening? And here, the next plots clearly show what is happening. Can you tell what is going on?

```
## Analysis of Variance Table
##
## Response: tumor.size
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## exposure    1 582.78   582.78 221.8174 < 2.2e-16 ***
## drug         1  54.42    54.42  20.7137 7.394e-06 ***
## exposure:drug 1  17.46    17.46   6.6459 0.01035 *
## Residuals   346 909.05     2.63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

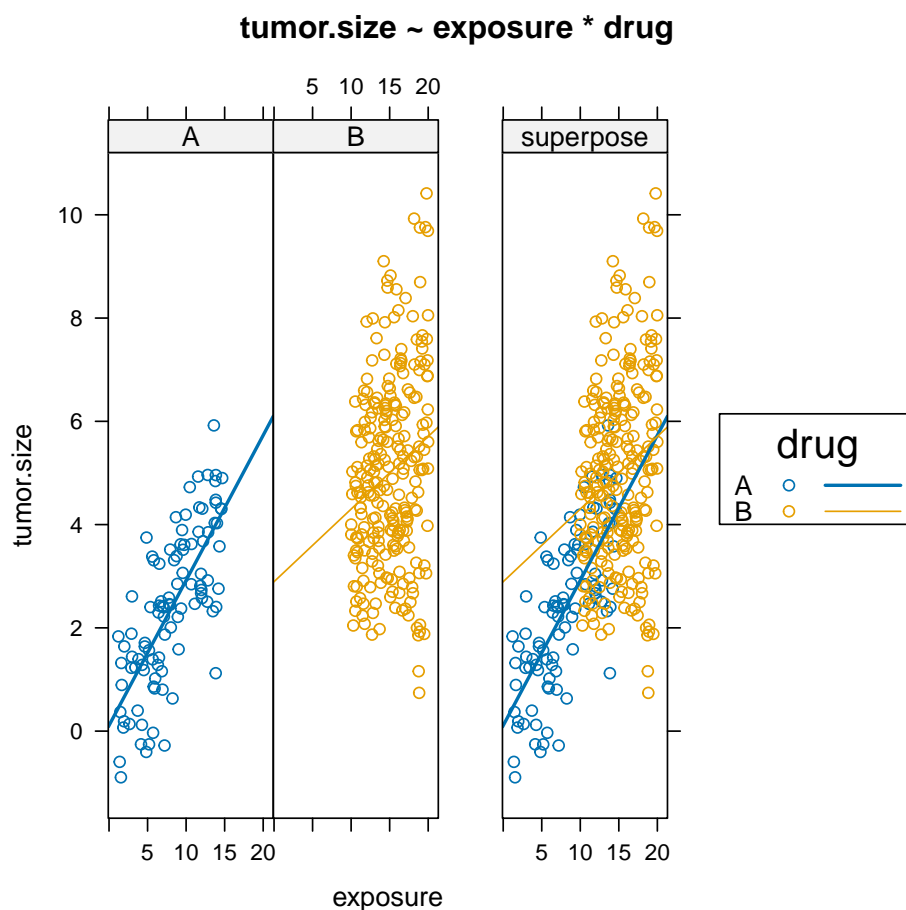


Figure 8 – Trying to make sense of those patterns, 1

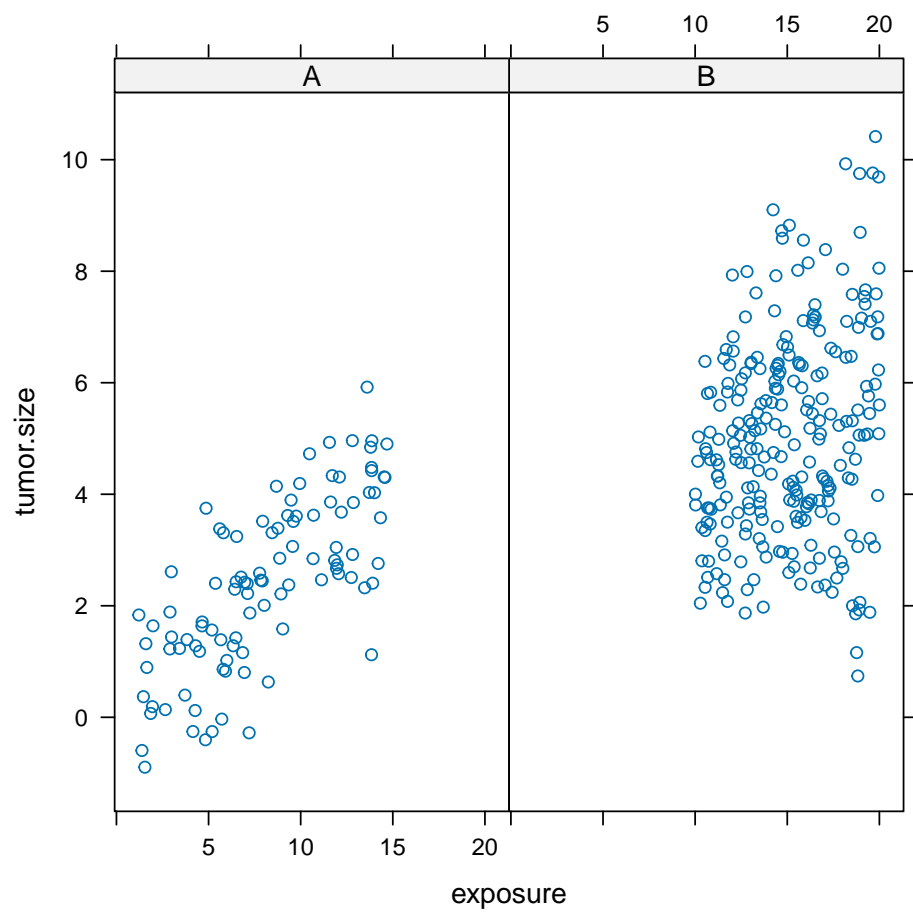


Figure 9 – Trying to make sense of those patterns, 1b

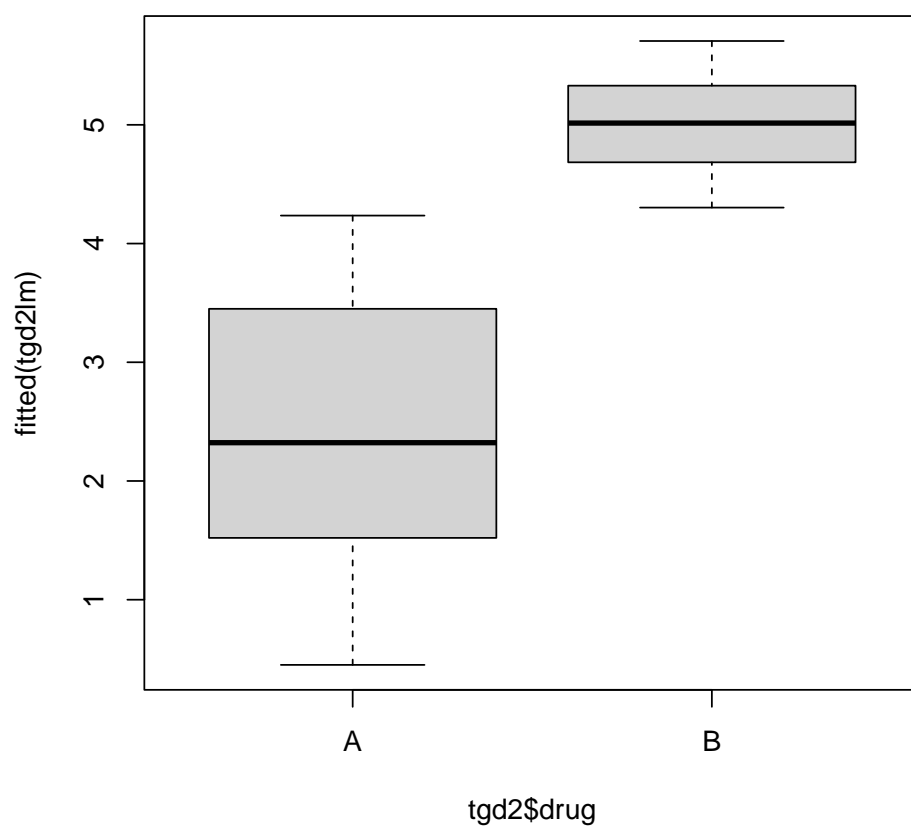


Figure 10 – [Trying to make sense of those patterns, 2](#)

25.5 Diagnostics: more examples of non-constant variance (and other issues)

The next example presents fairly common patterns:

$$\text{lm}(y1 \sim x1)$$

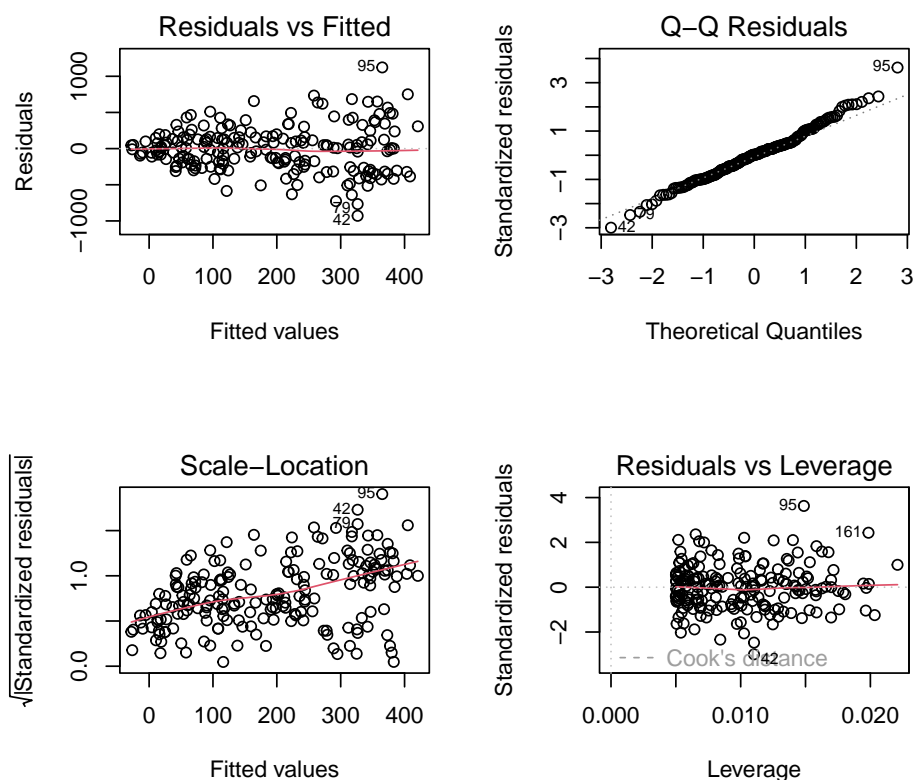


Figure 11 – Another example with non-constant variance in a regression model

Some basic statistics with R

The next is a contrived example, but one that shows things are clearly wrong (look at the bottom left).

$$\text{lm}(y1 \sim x1)$$

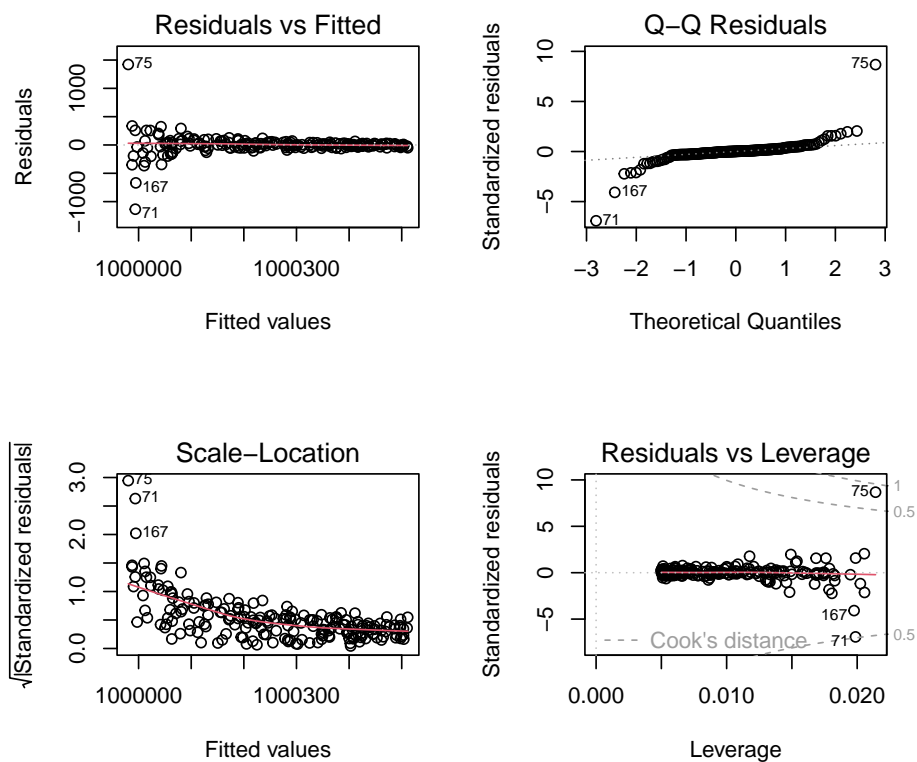


Figure 12 – Yet another example of non-constant variance (plus other issues —these is a contrived example))

25.6 Diagnostics: a couple of examples from ANOVA models that are largely OK

$\text{lm}(\text{hdl} \sim \text{smoking} * \text{sex})$

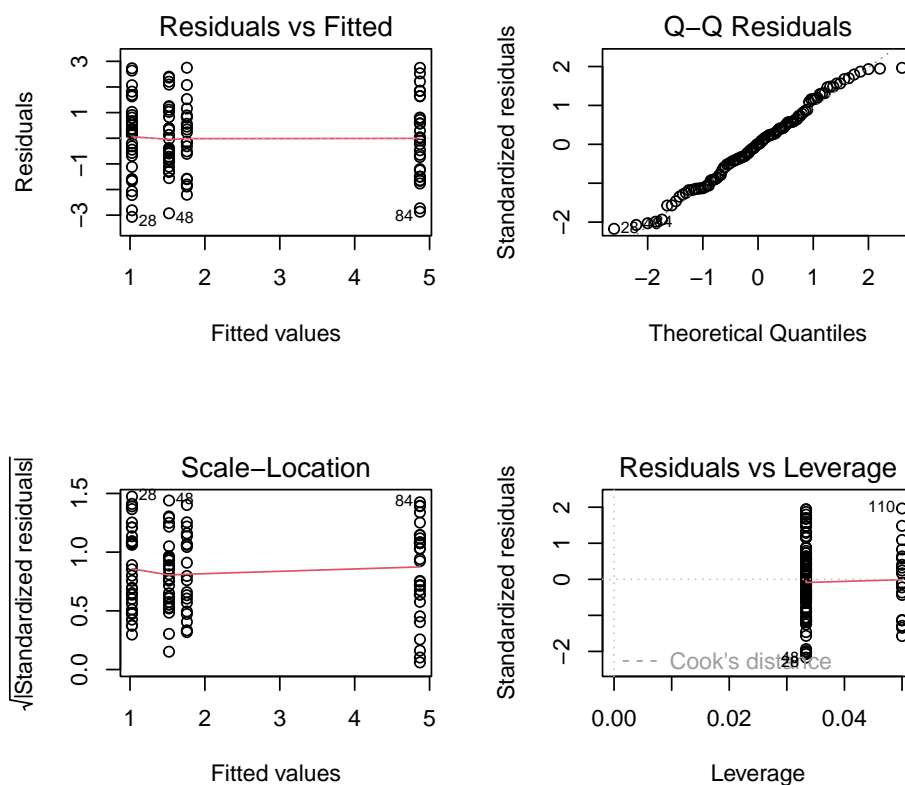


Figure 13 – Diagnostic plots from an ANOVA model with interactions and unbalanced data

Some basic statistics with R

In the next plot what we probably want to understand is why we have those three large residuals. Those are leading to the pattern in the upper left diagnostic figure.

$\text{lm}(\text{dsdd} \sim \text{drug} * \text{exercise})$

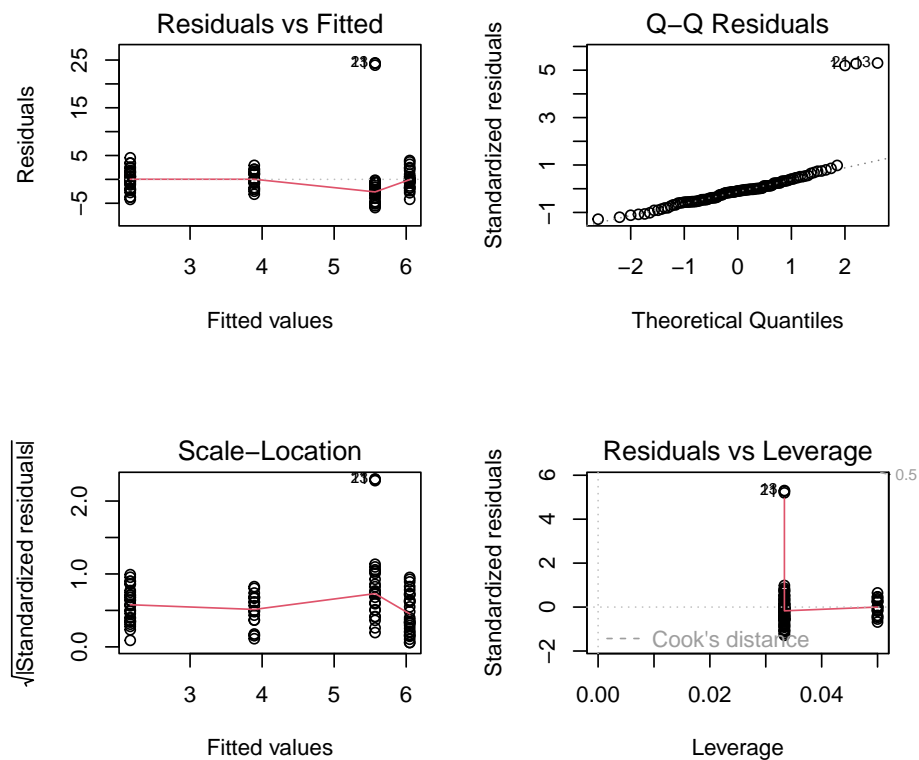


Figure 14 – Diagnostic plots from another ANOVA model with interactions and unbalanced data

25.7 Diagnostics: more examples with designed experiments

Follow the text and the graphics. We will discuss this in class.

We will create a data set where the true model is one where there really is an interaction. And we will fit both an additive model (i.e., a model without interaction) and a model with interaction.

```
## See how diagnostics suggest missing interaction or
## at least suggest something is wrong.
set.seed(1)
sex <- factor(rep(c("Male", "Female"), c(20, 20)))
drug <- factor(rep(rep(c("A", "B"), c(10, 10)), 2))
y <- rep(c(8, 16, 10, 12), rep(10, 4))
y <- y + rnorm(length(y), sd = 1.5)
y.data1 <- data.frame(y, sex, drug)
rm(y, sex, drug)
with(y.data1, tapply(y, list(sex, drug), mean))

##              A          B
## Female 9.799490 12.18110
## Male   8.198304 16.37327

## Fit the model
myAdditive2 <- lm(y ~ sex + drug, data = y.data1)
myInteract2 <- lm(y ~ sex * drug, data = y.data1)
summary(myAdditive2)

##
## Call:
## lm(formula = y ~ sex + drug, data = y.data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.695 -1.712 -0.232  1.685  3.343
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.3512     0.5535  15.088 < 2e-16 ***
## sexMale       1.2955     0.6391   2.027  0.0499 *
## drugB         5.2783     0.6391   8.259 6.42e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.021 on 37 degrees of freedom
## Multiple R-squared:  0.6615, Adjusted R-squared:  0.6432
## F-statistic: 36.16 on 2 and 37 DF, p-value: 1.979e-09
```

Some basic statistics with R

```
summary(myInteract2)

##
## Call:
## lm(formula = y ~ sex * drug, data = y.data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.6953 -0.6854  0.1639  0.9228  2.1946
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      9.7995     0.4322  22.676 < 2e-16 ***
## sexMale          -1.6012     0.6112  -2.620  0.012796 *
## drugB             2.3816     0.6112   3.897  0.000407 ***
## sexMale:drugB     5.7934     0.8643   6.703  8.08e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.367 on 36 degrees of freedom
## Multiple R-squared:  0.8494, Adjusted R-squared:  0.8369
## F-statistic: 67.7 on 3 and 36 DF, p-value: 7.158e-15
```

We will now show diagnostic plots for the additive model and the model with interaction.

```
par(oma=c(0,0,3,0), mfrow = c(2, 3))
plot(myAdditive2, which = c(1:5)) ## look at first plot
```

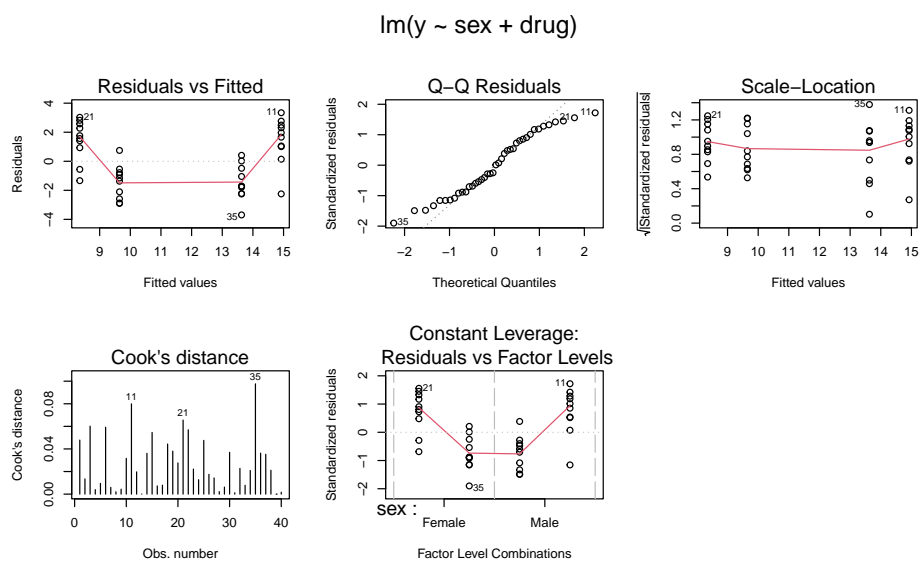


Figure 15 – Additive model, myAdditive2, when there is interaction

Some basic statistics with R

```
par(oma=c(0,0,3,0), mfrow = c(2, 3))  
plot(myInteract2, which = c(1:5))
```

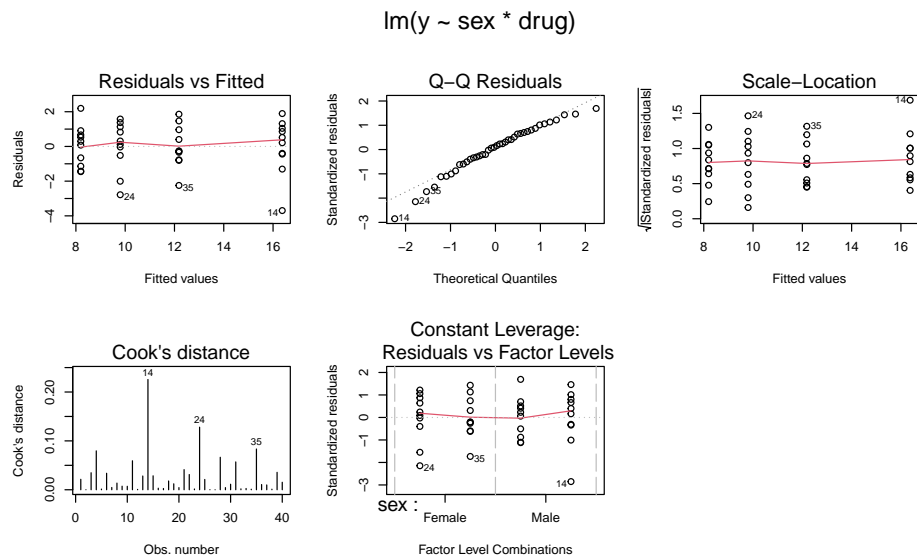


Figure 16 – Interaction model, myInteract2, when there is interaction

Some basic statistics with R

Now, we will create a data set where there is a point with a large value for Cook's statistic. The data are perfectly balanced and the large Cook's distance is, thus, the result of a large residual.

```
#####  
## Large cook's in anova  
set.seed(1)  
sex <- factor(rep(c("Male", "Female"), c(20, 20)))  
drug <- factor(rep(rep(c("A", "B"), c(10, 10)), 2))  
y <- rep(c(8, 12, 11, 15), rep(10, 4))  
y <- y + rnorm(length(y), sd = 1.5)  
y.data2 <- data.frame(y, sex, drug)  
rm(y, sex, drug)  
## create a large outlier  
  
y.data2[1, 1] <- 30  
with(y.data2, tapply(y, list(sex, drug), mean))  
  
##           A           B  
## Female 10.79949 15.18110  
## Male   10.49227 12.37327  
  
## ## Fit the model  
myAdditive2b <- lm(y ~ sex + drug, data = y.data2)  
## myInteract <- lm(y ~ sex * drug, data = y.data2)  
## summary(myAdditive)  
## summary(myInteract)  
  
## ## diagnostics, all of them except 6th  
## we actually have a large Cook's distance  
par(oma=c(0,0,3,0), mfrow = c(2, 3))  
plot(myAdditive2b, which = 1:5)
```

Some basic statistics with R

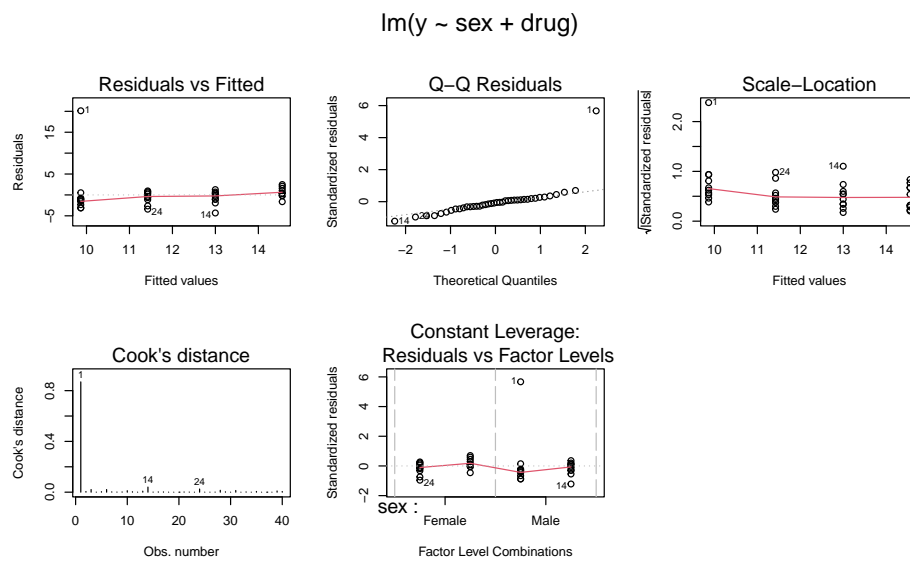


Figure 17 – [myAdditive2b](#): large Cook with balanced data

Some basic statistics with R

Now, remove the offending value (**BEWARE: this is not necessarily what you should do in real life!! This is just for the sake of showing the diagnostic plots**). Pay attention at the change in the plots (and we no longer have constant leverage).

```
## model with and without first obs
summary(lm(y ~ sex + drug, data = y.data2))

##
## Call:
## lm(formula = y ~ sex + drug, data = y.data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.3205 -1.1996 -0.2456  0.5103 20.1329
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   11.425      1.011   11.300 1.48e-13 ***
## sexMale       -1.558      1.167   -1.334  0.1903
## drugB          3.131      1.167    2.682  0.0109 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.692 on 37 degrees of freedom
## Multiple R-squared:  0.1952, Adjusted R-squared:  0.1517
## F-statistic: 4.487 on 2 and 37 DF,  p-value: 0.018

summary(lm(y ~ sex + drug, data = y.data2[-1, ]))

##
## Call:
## lm(formula = y ~ sex + drug, data = y.data2[-1, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.7763 -0.6446  0.1305  0.9171  2.1582
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   10.8805      0.3727  29.197 < 2e-16 ***
## sexMale       -2.6458      0.4341  -6.094 5.20e-07 ***
## drugB          4.2196      0.4341   9.720 1.32e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.355 on 36 degrees of freedom
```


Some basic statistics with R

```
## Multiple R-squared:  0.7813, Adjusted R-squared:  0.7691
## F-statistic: 64.29 on 2 and 36 DF,  p-value: 1.314e-12
```

```
#### Diagnostics if we remove the offending value
par(oma=c(0,0,3,0), mfrow = c(2, 3))
plot(lm(y ~ sex + drug, data = y.data2[-1, ]), which = 1:5)
```

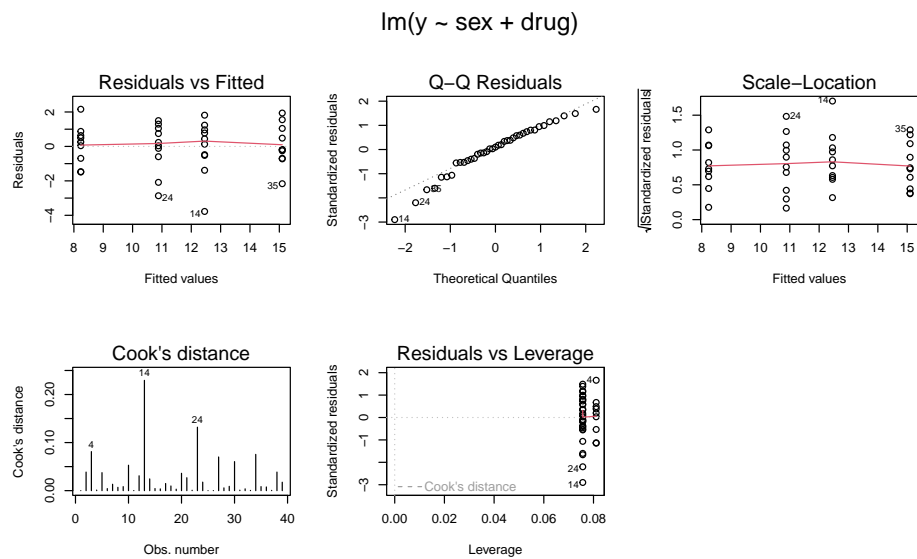


Figure 18 – [myAdditive2b](#) removing the large offending value

Some basic statistics with R

Now, create unbalance in the data by removing two observations (but not the one with large residual). Notice how we get the plot of residuals vs. leverage and we can clearly see the observation with the large Cook's distance.

```
## The model with two observations missing
## create unbalance
y.data3 <- y.data2[-c(35, 40), ]
with(y.data3, tapply(y, list(sex, drug), mean))

##           A           B
## Female 10.79949 15.34147
## Male   10.49227 12.37327

myAdditive3 <- lm(y ~ sex + drug, data = y.data3)
```

```
par(oma=c(0,0,3,0), mfrow = c(2, 3))
plot(myAdditive3, which = 1:5) ## see Cook's
```

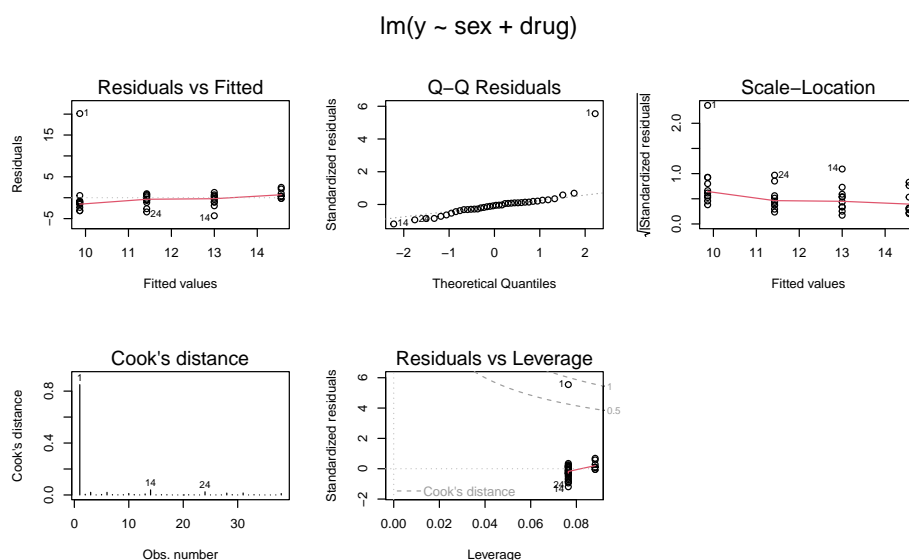


Figure 19 – myAdditive3: missing two observations

Some basic statistics with R

```
## A few more (maybe technical) details. Statistics for influence,  
## the ``hat'' matrix.  
(hii2b <- lm.influence(myAdditive2b, do.coef = FALSE)$hat) ## constant  
(hii3 <- lm.influence(myAdditive3, do.coef = FALSE)$hat) ##nope  
  
diff(range(hii2b)) ## constant indeed  
diff(range(hii3)) ## not constant
```

25.8 Diagnostics: further issues

For you to read and play around further:

- An extended qqplot is available from package [car](#), [qqPlot](#) (and in R Commander under “Residual quantile comparison plots”).
- “Component+Residual” (or partial residual³⁹) plots allow us to examine, in models with multiple regressors, deviations from linearity and could suggest the appropriate transformation. “CERES plots” are a variation of “Component + Residual” plots that work well even if relationships are strongly nonlinear. Both also available from [car](#).
- Various diagnostics related to [dfbetas](#) allow us to identify influential observations in specific terms of the model.
- Variance inflation factors help us detect possible problems caused by collinearity (correlations between independent variables).
- Added variable plots are particularly useful in multiple regression problems with multiple independent variables; they can help to identify influential points (which are easily masked with multiple variables) and can also help to try to find a good functional relationship (but Component+Residual are more useful here). Again, available from [car](#).
- A variety of numerical tests and diagnostics are also available (e.g., tests for nonlinearity or for homoscedasticity).
- Package [car](#) and the accompanying book by Fox and Weisberg “An R companion to applied regression, 3rd ed” contain excellent and detailed comments about those and other diagnostics, and examples of how to use the functions in the [car](#) package. Read chapter 8 of that book. It will explain you how to use them as well as possible remedial measures. Actually, read the complete book. In the meantime, you can also take a look, for a summary, at the very nice sections 8.3 to 8.5 in Kabakoff’s “R in action.”

³⁹These plots, for each variable, shows a plot of the independent variable, say x , on the horizontal axis and, on the ordinates, the “partial residuals” $= residual + \hat{\beta}x$.

Some basic statistics with R

- There is in CRAN a very interesting package, *gvlma* (<https://cran.r-project.org/web/packages/gvlma/>), by Peña and Slate, that implements the methods in their paper from 2006 “Global Validation of Linear Model Assumptions”, *J. American Statistical Association*, 101(473):341-354. This offers a global testing procedure that allows further examination of each of the key assumptions. Here, I have preferred to start by looking carefully at each one of the traditional diagnostic plots. But you will most likely want to take a look at this package.
- What if diagnostics identify a problem? The usual procedures are making sure the model is right, and possibly transforming either the response or some of the predictors, and maybe using more complex models (e.g., for modeling the variance, etc). But **before** fitting a model, think about it carefully and what is an appropriate biological model of the phenomenon. That might dictate, for example, reasonable *a priori* transformations of the response or predictors, or what the functional form should be. (For example, modeling metabolic rate as a function of body mass was a bad idea; there are many strong arguments to suggest a log-log relationship is the way to go).

26 Variable and model selection

26.1 Why model selection?

You say you want to select variables? Select them for what? Prediction? Interpretation? Manipulation? The first thing to realize is that procedures based purely in statistical significance criteria might select “statistically important” variables (under some suitable definition of “important”), but those need not be the most relevant from a biological point of view, or causally, etc.

This is a very quick summary of some of the reasons why we fit statistical models and, then, maybe conduct variable and/or model selection:

- Interpretation/understanding/insight: why are things this way? What factors are relevant for a process and why? This is trying to gain scientific understanding in the sense of “uncovering the truth”. Is sleep deprivation what causes whatever? Or is it differences in light exposure? Or ... This is often what much (not all!!!) scientific modeling is about.
- Prediction: predict a variable given others. This might be totally unrelated to intervention (which is often what causal inference is after) or understanding/gaining insight into a process.

Example: banks might want to filter customers unlikely to return a credit. They are not really trying to correct anyone’s behavior, they are not trying to understand why people ask for money they cannot return. They just want to predict (so as not to offer them certain financial products).

Or you might want estimate response to a treatment based on the gene expression profile of patients. You are not trying to understand why that gene’s over-expression or repression leads to that response, just predict, so as to give each patient the best treatment. (In other words, you are not trying to “understand the true relationship between gene expression and response to treatment”).

Many other models are of this kind. For example, many meteorological models are like this: if you see AEMET’s predictions you only care about them being right (and they often are). You are not trying to modify anything (i.e., “let’s try to make of today a rainy day”) nor understand.

Incidentally, that we care just about prediction is what makes models of high complexity, including deep learning, with millions of parameters that our brains can’t just wrap their heads around, fine for many these tasks⁴⁰.

- Causal inference, intervention: what would happen in Y if we manipulated X? For example, if we were to get people to eat legumes twice a week, how many colon cancers could we prevent?

⁴⁰Yes, we can argue that interpretable AI and machine learning models are important, for practical reasons —better predictions—, for ethical reasons, for legal reasons, and for conceptual reasons.

Some basic statistics with R

Yes, causal inference and intervention on the one hand and understanding on the other are often very closely related. But they are not the same. For example, you might have a relatively good understanding of a particular process, and yet be unable to predict exactly what an intervention will do because of the presence of modifier variables.

And sometimes we might not understand the mechanisms in detail⁴¹ and, yet, be able to conduct very successful interventions if we are able to estimate the causal effect of a variable⁴².

And intervention is not totally unrelated to successful prediction either, of course. Elements of models that play a role in meteorological and climatological prediction are also present in causal models that are used to guide (or try to guide, unfortunately without much success) actions to prevent the disaster of climate change; we are asking here “what variables should we change, and to what values, to achieve this result or prevent this other result?”

But models that might be excellent for predictions of current outcome might be very bad at prediction under new circumstances; causal inference and intervention are after the last objective whereas you looking at AEMET in the morning is you concerned about prediction-without-trying-to-modify-or-understand..

And, maybe counterintuitively, some models that are “closer to the truth” in the sense of “they incorporate all the variables that matter” can actually be much worse than simpler (i.e., “less true” models) for making predictions. Please, think about this **now!** Come up with examples of your own; why could this be? If this is not clear, **ask about this in class.**

26.2 Variable selection: the summary

With regards to the statistical procedures, we will summarize it as follows: please, please, please, distrust automated variable selection procedures that rely on p-values or F-statistics of individual variables (in all their variants, such as stepwise, etc). These procedures rarely do what you want to do, are often extremely unstable, etc. They are trivial to implement, and that makes them very common, but they are rarely what your scientific question needs. Make sure you use subject-matter guidance to, well, guide you on what are sensible hypotheses to test, and in what order. Subject-matter knowledge might dictate what are and are not candidates for deletion and in what order.

Careful model comparison, for instance using something like `anova(model1, model2)` , as we have seen (e.g., section [Formally comparing models](#) or [ANCOVA with the birds and the reptiles](#)), might be a good idea. Notice, again, that `anova(model1, model2)` is all about comparing models.

⁴¹When one has an adequate understanding of mechanisms is actually a difficult question.

⁴²Briefly: the effect that changing, by manipulation, a variable from one state or value to another has on the outcome.

Some basic statistics with R

The book by Frank Harrell, “Regression modeling strategies” contains great discussions of these topics, including using the bootstrap to assess stability, etc.

26.3 Model selection using AIC and `step`

If you really need automated or semiautomated procedures, then reasonable strategies use model-comparison criteria such as AIC (e.g., with the `step` function) are much more sensible. But the focus, now, is different: we are doing **model selection**, not variable selection. And we are trying to achieve a different objective: we are trying to **find the best model to make predictions**. In other words, we are trying to find the model that, when applied to new data, will give the best predictions (we will explain what we mean by “best predictions”).

26.3.1 A very brief introduction to AIC

AIC stands for Akaike Information Criterion. This is the AIC of a model:

$$-2 \ln(\hat{L}) - 2k$$

where k is the number of parameters, \hat{L} is the likelihood of the model at the maximum likelihood parameters (and \ln is the base e logarithm). (Many definitions of AIC multiply the above by -1 , as we will see below, but that is irrelevant; so if with our definition better models are those with larger AIC, if you multiply by -1 , better models are those with smaller AIC).

And what is the likelihood? You should know this already. If not, ask in class, but essentially, $L(\theta)$, is the probability of the data under parameter θ (it is better to think of the joint density, as that includes, then, continuous distributions). For a data set, we can find the value of θ , the parameters, that makes that likelihood as large as it can be; this would be the maximum likelihood estimate of the parameter(s) θ . So in the expression above, \hat{L} is the value of the likelihood when evaluated with the parameters that make that likelihood as large as possible, i.e., at the value(s) that make the data as likely as possible. In a regression example, this would be the value of the likelihood when the regression coefficients (and the intercept) take the values estimated by your regression software.

What is AIC computing then? The log of the likelihood penalized by the number of parameters. In other words, AIC is showing the trade-off between good fit (to the current data) and number of parameters: models with many parameters might be able to adjust the data better (i.e., have larger likelihood), but slight increases in likelihood might not justify some of those parameters. So AIC says something like “larger likelihood is good, but do not get carried away adding too many parameters”. With too many parameters, the model might be just too complex. Too complex for what? Not for the present data (the likelihood increased), but too complex for making good predictions in other data sets. Too many parameters increase the risk of **overfitting**.

These are three key features of AIC:

Some basic statistics with R

1. AIC (under certain assumptions) is an unbiased estimate of the predictive accuracy of the model: the expected likelihood of the model when applied to new data (see Figure 20, from Otsuka (2023), *Thinking about statistics. The philosophical foundations.*).
2. AIC measures (again, under certain assumptions) how close to the true values are the predictions of the model⁴³.
3. AIC is equivalent to leave-one-out cross-validation for (many) linear models (including linear regression and mixed-effects).

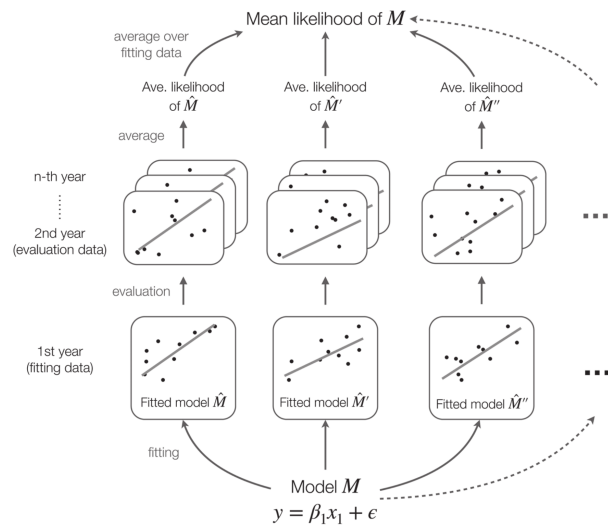


FIGURE 4.2 A (hypothetical) calculation of a model's mean likelihood. The predictive performance of a fitted model \hat{M} can be evaluated by averaging its likelihood with respect to many datasets of the same nature. The mean likelihood of the model M is obtained by repeating this process over different initial datasets, using fitted models $\hat{M}', \hat{M}'', \dots$ (note these have different regression slopes due to randomness in the fitting data). Since this calculation is infeasible in reality, AIC aims to estimate it from a single dataset.

Figure 20 – AIC as estimate of expected likelihood

From J. Otsuka (2023), *Thinking about statistics. The philosophical foundations.*, Routledge, Figure 4.2, p. 116.

The importance of the above three results is huge.

Now, suppose we have a bunch of models, and we measure the AIC of each. Model selection via AIC will tell us to prefer the model with the largest AIC. (Note: largest if we define AIC as above; we should prefer the model with the smallest AIC if we define AIC as $AIC = -2 \ln(\hat{L}) + 2k$).

The key idea is that AIC allows us to select models so as to keep the model that has (among the ones we consider) the best predictive performance in future samples.

Before we continue, it is crucial that we understand that:

⁴³Here, “how close” is evaluated using the Kullback-Leibler divergence, a measure of distance between probability distributions.

Some basic statistics with R

1. The model thus selected (i.e., the model with the best AIC) need not be the model that has “all and every parameter that really affects the outcome”: “smaller models”, models that do not include some parameters, might actually work better. Make sure you understand this.

To repeat: for a given sample size, the best model according to AIC might tell you NOT to include a variable that really has an effect (that is “significant”). Not all variables that “really have an effect” necessarily improve predictive performance; similarly, if the relationship is really slightly quadratic, adding the parameter for the quadratic term not necessarily improves the predictive performance. Yes, this means that “true models” (in terms of variables, parameters, etc) are not necessarily the best models for predictive performance, and simpler models are often better. Can you think of a reason why this could be the case? (Hint: you need to estimate the model from data.)

2. The size (the number of parameters) of the selected model depends on the sample size: with larger samples sizes, we can often afford to fit larger models, and thus with larger sample sizes AIC will often select larger models. (Larger can mean “more variables” or “more parameters” if we include, say, non-linear relationships, etc).

A great, book-length, treatment of AIC is Burnham and Anderson, 2002, *Model Selection and Multimodel Inference: A practical information-theoretic approach*, 2nd ed., Springer-Verlag.

A conceptual discussion of AIC and its philosophical implications is given in pp. 109–124 of Otsuka, 2023, *Thinking about statistics. The philosophical foundations*, Routledge.

There are other criteria related to AIC. For example, for small sample sizes we might want to use `AIC_c`; other criteria such as BIC seem very similar, though they have different objectives; etc. We will not get into further details here.

26.3.2 Model selection using AIC: an example

The actual data set used here is discussed, also in the context of variable selection, by P. Dalgaard in chapter 11 of “Introductory statistics with R”.

Of course, using model diagnostics (section [Diagnostics](#)) with the initial and final models is always a necessity.

Let’s give some examples (look at them carefully, and compare with what we did by hand, and with whether or not these are the models you would use).

We will use the `step` function. As is customary, we are defining $AIC = -2\log(\hat{L}) + 2k$, so we want to minimize AIC.

```
step(mcyst2, direction = "both")  
## Start: AIC=168.89
```

Some basic statistics with R

```
## pemax ~ age * sex
##
##           Df Sum of Sq  RSS   AIC
## - age:sex  1       189 15779 167.19
## <none>                 15590 168.89
##
## Step: AIC=167.19
## pemax ~ age + sex
##
##           Df Sum of Sq  RSS   AIC
## - sex      1       955.4 16734 166.66
## <none>                 15779 167.19
## + age:sex  1       189.0 15590 168.89
## - age      1      8819.5 24598 176.29
##
## Step: AIC=166.66
## pemax ~ age
##
##           Df Sum of Sq  RSS   AIC
## <none>                 16734 166.66
## + sex    1       955.4 15779 167.19
## - age    1     10098.5 26833 176.46
##
## Call:
## lm(formula = pemax ~ age, data = cystfibr2)
##
## Coefficients:
## (Intercept)          age
##      50.408         4.055
```

```
step(mcyst, direction = "both")

## Start: AIC=169.19
## pemax ~ age + height + weight
##
##           Df Sum of Sq  RSS   AIC
## - height  1         6.75 15789 167.21
## - age     1       186.86 15969 167.49
## - weight  1       769.60 16552 168.38
## <none>                 15782 169.19
##
## Step: AIC=167.2
## pemax ~ age + weight
##
##           Df Sum of Sq  RSS   AIC
```

Some basic statistics with R

```
## - age      1      216.51 16006 165.54
## - weight   1      945.19 16734 166.66
## <none>                                15789 167.21
## + height   1         6.75 15782 169.19
##
## Step:  AIC=165.55
## pemax ~ weight
##
##           Df Sum of Sq  RSS    AIC
## <none>                16006 165.54
## + age      1      216.5 15789 167.21
## + height   1        36.4 15969 167.49
## - weight   1    10827.2 26833 176.46
##
## Call:
## lm(formula = pemax ~ weight, data = cystfibr2)
##
## Coefficients:
## (Intercept)      weight
##      63.546         1.187
```

```
## In metab, we drop the interaction
step(metab_b_r, direction = "both")

## Start:  AIC=-463.77
## logMetabolicRate ~ logBodyMass * Class
##
##           Df Sum of Sq  RSS    AIC
## - logBodyMass:Class  1  0.075167 12.646 -464.71
## <none>                                12.571 -463.77
##
## Step:  AIC=-464.71
## logMetabolicRate ~ logBodyMass + Class
##
##           Df Sum of Sq  RSS    AIC
## <none>                12.646 -464.71
## + logBodyMass:Class  1    0.075 12.571 -463.77
## - Class              1   109.234 121.880 -63.42
## - logBodyMass        1   226.058 238.704  56.23
##
## Call:
## lm(formula = logMetabolicRate ~ logBodyMass + Class, data = anage_a_r)
##
## Coefficients:
## (Intercept)  logBodyMass  ClassReptilia
```

Some basic statistics with R

```
##           -3.1460           0.6471           -3.1885

## But nothing can be dropped here
step(longev_b_r, direction = "both")

## Start:  AIC=-308.85
## logLongevity ~ logBodyMass * Class
##
##              Df Sum of Sq    RSS    AIC
## <none>                27.258 -308.85
## - logBodyMass:Class   1    1.6414 28.900 -300.79
##
## Call:
## lm(formula = logLongevity ~ logBodyMass * Class, data = anage_a_r)
##
## Coefficients:
##              (Intercept)              logBodyMass
##                1.7888                0.2182
##      ClassReptilia logBodyMass:ClassReptilia
##        -0.9858                0.2561
```

26.4 Differences between model selection using AIC and model comparison using `anova` (and hypothesis testing using `Anova`)

The **major difference** is that when we compare models using `anova`, or examine each of the terms in a fitted model with, say, `Anova`, **we are conducting statistical hypothesis testing**⁴⁴. In contrast, the AIC criterion is not used to conduct hypothesis testing, but is a criterion related to predictive performance; so when we carry out **model selection using AIC we are trying to find the best model where “best” is “best from the point of view of prediction”**.

A second difference is that when we use `anova` and `Anova` to compare models or assess the significance of different variables, we generally do not (should not) do this with tens or hundreds of models and variables. We are testing some specific hypotheses and doing it “manually”. In contrast, using model selection with AIC (or similar criteria) the procedure will run automatically and can potentially compare hundreds of models.

A third difference is that most automated procedures such as `step` with AIC will add or remove a “complete variable”. However, with `Anova` (and `glht`, and others) you can test specific hypotheses of interest to you, including hypotheses where, say, the average of two levels of a factor are equal to the third, etc.

⁴⁴The hypotheses tests we conduct with `Anova` are, in fact, comparisons of models, where we test the null hypothesis that the smaller model is good enough.

Some basic statistics with R

Finally, note that `anova` should only be used to compare nested models (even if `Anova` and others you can test hypotheses of interest to you). `step` will follow greedy comparison rules for moving from one model to another, but if you insist you could look at the AIC of a large collection of models (or all the possible models for a set of variables) and compare between models even if they are not nested versions of each other. Again, doing this using AIC might be sensible because the objective is prediction, not hypothesis testing.

Simplifying the issues a little bit, hypothesis testing and model building using tools such as `anova` and `Anova` are things you do when you want to understand a phenomenon, whereas model selection using criteria such as AIC is what you do when you want to build models with the best predictive performance.

As a consequence of the above, of course, sometimes using `step` is something that would make no sense and you would not even consider, for example in a clean, clear-cut experiment with two factors: you want to use an ANOVA, no need for `step`. And, similarly, with thousands of possible variables, running thousands of manually run model tests would not make sense if you are trying to build a good predictive model; use a procedure that will try to find the best model from the prediction point of view.

26.5 Model selection or model averaging?

To finish this section: why select one single model? If we are interested in prediction, it could make a lot more sense to obtain predictions from several models, and weight those predictions using AIC weights. This is discussed at length in the Burnham and Anderson book mentioned above (and in many other places).

26.6 A large pemax model as an example

A large pemax model:

```
mcystL <- lm(pemax ~ age * height * weight * sex, data = cystfibr2)
summary(mcystL)

##
## Call:
## lm(formula = pemax ~ age * height * weight * sex, data = cystfibr2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -45.211  -7.080   1.627  10.216  25.901
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.128e+02  5.469e+02   0.755   0.470
## age          -1.302e+02  1.238e+02  -1.052   0.320
```

Some basic statistics with R

```
## height          1.755e+00  2.842e+00   0.618   0.552
## weight          -1.031e+01  3.651e+01  -0.282   0.784
## sexFemale       7.002e+01  1.055e+03   0.066   0.949
## age:height      5.549e-01  5.627e-01   0.986   0.350
## age:weight      1.580e+00  4.088e+00   0.386   0.708
## height:weight   1.383e-02  1.711e-01   0.081   0.937
## age:sexFemale   2.064e+02  2.056e+02   1.004   0.342
## height:sexFemale -6.785e+00  8.754e+00  -0.775   0.458
## weight:sexFemale -5.038e+01  7.230e+01  -0.697   0.504
## age:height:weight -6.410e-03  2.022e-02  -0.317   0.758
## age:height:sexFemale -8.199e-01  1.138e+00  -0.721   0.489
## age:weight:sexFemale -7.534e-01  5.018e+00  -0.150   0.884
## height:weight:sexFemale 4.499e-01  4.761e-01   0.945   0.369
## age:height:weight:sexFemale -5.172e-03  2.661e-02  -0.194   0.850
##
## Residual standard error: 24.55 on 9 degrees of freedom
## Multiple R-squared:  0.7979, Adjusted R-squared:  0.461
## F-statistic: 2.369 on 15 and 9 DF,  p-value: 0.09685

Anova(mcystL)

## Anova Table (Type II tests)
##
## Response: pemax
##
##          Sum Sq Df F value    Pr(>F)
## age          379.6  1  0.6299 0.44780
## height      3281.3  1  5.4453 0.04449 *
## weight       117.5  1  0.1950 0.66918
## sex           0.6  1  0.0010 0.97595
## age:height    694.8  1  1.1529 0.31088
## age:weight    457.2  1  0.7588 0.40635
## height:weight   0.1  1  0.0002 0.98864
## age:sex      2595.5  1  4.3072 0.06777 .
## height:sex     818.5  1  1.3583 0.27379
## weight:sex     548.9  1  0.9108 0.36482
## age:height:weight 308.0  1  0.5112 0.49276
## age:height:sex   410.1  1  0.6806 0.43072
## age:weight:sex   861.9  1  1.4303 0.26227
## height:weight:sex 722.4  1  1.1988 0.30200
## age:height:weight:sex 22.8  1  0.0378 0.85017
## Residuals      5423.4  9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

mcystL_r <- step(mcystL, direction = "both")

## Start:  AIC=166.49
```

Some basic statistics with R

```
## pemax ~ age * height * weight * sex
##
##              Df Sum of Sq    RSS    AIC
## - age:height:weight:sex  1    22.777 5446.2 164.59
## <none>                    5423.4 166.49
##
## Step:  AIC=164.59
## pemax ~ age + height + weight + sex + age:height + age:weight +
##      height:weight + age:sex + height:sex + weight:sex + age:height:weight +
##      age:height:sex + age:weight:sex + height:weight:sex
##
##              Df Sum of Sq    RSS    AIC
## - age:height:weight      1    308.04 5754.2 163.97
## - age:height:sex         1    410.11 5856.3 164.41
## <none>                    5446.2 164.59
## - height:weight:sex      1    722.39 6168.6 165.71
## - age:weight:sex         1    861.92 6308.1 166.27
## + age:height:weight:sex  1     22.78 5423.4 166.49
##
## Step:  AIC=163.97
## pemax ~ age + height + weight + sex + age:height + age:weight +
##      height:weight + age:sex + height:sex + weight:sex + age:height:sex +
##      age:weight:sex + height:weight:sex
##
##              Df Sum of Sq    RSS    AIC
## - age:height:sex         1    266.60 6020.8 163.10
## - height:weight:sex      1    451.26 6205.5 163.86
## <none>                    5754.2 163.97
## - age:weight:sex         1    575.27 6329.5 164.35
## + age:height:weight      1    308.04 5446.2 164.59
##
## Step:  AIC=163.1
## pemax ~ age + height + weight + sex + age:height + age:weight +
##      height:weight + age:sex + height:sex + weight:sex + age:weight:sex +
##      height:weight:sex
##
##              Df Sum of Sq    RSS    AIC
## - height:weight:sex      1    184.73 6205.5 161.86
## - age:weight:sex         1    499.74 6520.5 163.10
## <none>                    6020.8 163.10
## - age:height             1    694.76 6715.6 163.83
## + age:height:sex         1    266.60 5754.2 163.97
## + age:height:weight      1    164.53 5856.3 164.41
##
## Step:  AIC=161.86
```

Some basic statistics with R

```
## pemax ~ age + height + weight + sex + age:height + age:weight +
##   height:weight + age:sex + height:sex + weight:sex + age:weight:sex
##
##              Df Sum of Sq    RSS    AIC
## - height:weight      1      0.13 6205.7 159.86
## <none>                      6205.5 161.86
## - age:height          1    698.55 6904.1 162.53
## - height:sex          1    848.51 7054.1 163.06
## + height:weight:sex    1    184.73 6020.8 163.10
## + age:height:weight    1     35.13 6170.4 163.72
## + age:height:sex       1      0.07 6205.5 163.86
## - age:weight:sex       1   1098.83 7304.4 163.93
##
## Step:  AIC=159.86
## pemax ~ age + height + weight + sex + age:height + age:weight +
##   age:sex + height:sex + weight:sex + age:weight:sex
##
##              Df Sum of Sq    RSS    AIC
## <none>                      6205.7 159.86
## - height:sex          1    867.89 7073.6 161.13
## + height:weight       1      0.13 6205.5 161.86
## + age:height:sex      1      0.07 6205.6 161.86
## - age:weight:sex      1   1216.89 7422.6 162.34
## - age:height          1   1647.87 7853.5 163.75
##
summary(mcystL_r)
##
## Call:
## lm(formula = pemax ~ age + height + weight + sex + age:height +
##   age:weight + age:sex + height:sex + weight:sex + age:weight:sex,
##   data = cystfibr2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -45.339  -8.127   1.488   6.848  30.308
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    266.7363    237.7810   1.122   0.2808
## age           -81.3842     29.8518  -2.726   0.0164 *
## height          1.4308      2.2207   0.644   0.5298
## weight         -3.5865      3.4352  -1.044   0.3142
## sexFemale      58.5371    164.0673   0.357   0.7266
## age:height      0.3375      0.1751   1.928   0.0744 .
## age:weight      0.2323      0.1671   1.390   0.1864
```


Some basic statistics with R

```
## age:sexFemale      39.1226    15.2540    2.565    0.0225 *
## height:sexFemale   -3.2630     2.3319   -1.399    0.1835
## weight:sexFemale    4.8224     5.1124    0.943    0.3615
## age:weight:sexFemale -0.4751     0.2868   -1.657    0.1198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.05 on 14 degrees of freedom
## Multiple R-squared:  0.7687, Adjusted R-squared:  0.6035
## F-statistic: 4.653 on 10 and 14 DF,  p-value: 0.004757

Anova(mcystL_r)

## Anova Table (Type II tests)
##
## Response: pemax
##           Sum Sq Df F value    Pr(>F)
## age           35.2  1  0.0795 0.78211
## height       3281.3  1  7.4027 0.01657 *
## weight        15.7  1  0.0354 0.85350
## sex           22.6  1  0.0510 0.82462
## age:height    1647.9  1  3.7176 0.07437 .
## age:weight     552.8  1  1.2472 0.28289
## age:sex       2653.3  1  5.9858 0.02823 *
## height:sex     867.9  1  1.9580 0.18350
## weight:sex    1630.9  1  3.6793 0.07572 .
## age:weight:sex 1216.9  1  2.7453 0.11977
## Residuals      6205.7 14
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Manually reducing it. Be careful here!!!
mcystL_r2 <- update(mcystL_r, . ~ . -age:weight:sex - height:sex - age:weight)
Anova(mcystL_r2)

## Anova Table (Type II tests)
##
## Response: pemax
##           Sum Sq Df F value    Pr(>F)
## age           15.6  1  0.0333 0.857423
## height        74.7  1  0.1589 0.695124
## weight        31.7  1  0.0673 0.798388
## sex            2.4  1  0.0051 0.943792
## age:height   4488.8  1  9.5478 0.006646 **
## age:sex     4609.9  1  9.8051 0.006082 **
## weight:sex  4455.8  1  9.4775 0.006810 **
## Residuals   7992.5 17
```

Some basic statistics with R

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

anova(mcystL_r2, mcystL_r)

## Analysis of Variance Table
##
## Model 1: pemax ~ age + height + weight + sex + age:height + age:sex +
##      weight:sex
## Model 2: pemax ~ age + height + weight + sex + age:height + age:weight +
##      age:sex + height:sex + weight:sex + age:weight:sex
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      17 7992.5
## 2      14 6205.7  3    1786.8 1.3437 0.3003
```

A final one, with another type of data (but make sure to read section [Side issue: a interesting case where some things went wrong without us noticing](#), as this is probably not a sensible model)

```
data(Prestige)
## do ?Prestige

mI <- lm(prestige ~ women * education * type, data = Prestige)

## Notice degrees of freedom of three and four way interactions
Anova(mI)

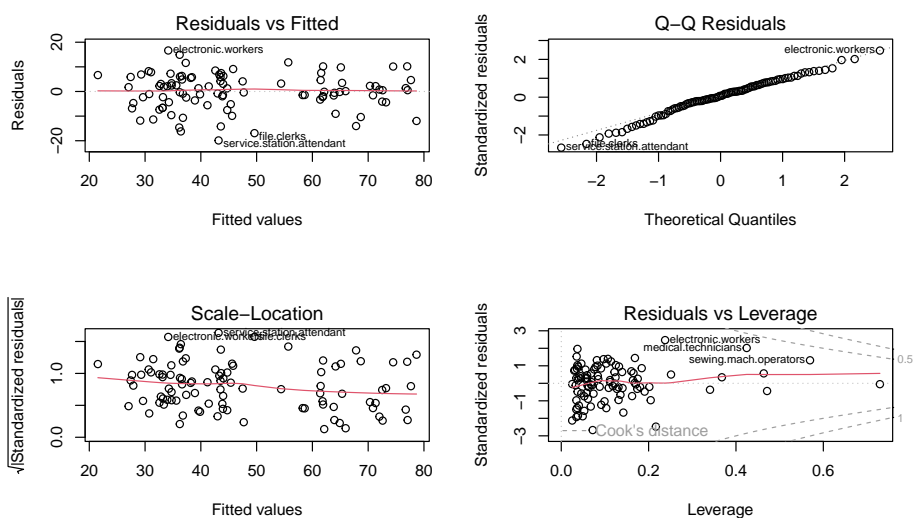
## Anova Table (Type II tests)
##
## Response: prestige
##
##           Sum Sq Df F value    Pr(>F)
## women           311.4  1  5.2158  0.024846 *
## education       2459.2  1 41.1945 7.241e-09 ***
## type             812.3  2  6.8040  0.001806 **
## women:education    8.1  1  0.1356  0.713573
## women:type         8.6  2  0.0724  0.930186
## education:type    153.3  2  1.2836  0.282284
## women:education:type 176.9  2  1.4816  0.233011
## Residuals       5133.9 86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(mI)

##
## Call:
## lm(formula = prestige ~ women * education * type, data = Prestige)
##
```

Some basic statistics with R

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8779  -3.9954   0.5177   5.3517  16.6454
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.563470   10.655718    0.147  0.883692
## women            -0.125292    0.329658   -0.380  0.704832
## education         4.208728    1.230381    3.421  0.000958 ***
## typeprof        -1.728122   21.372379   -0.081  0.935743
## typewc          -17.693514   31.784544   -0.557  0.579198
## women:education    0.007752    0.040960    0.189  0.850333
## women:typeprof     1.250643    0.817127    1.531  0.129554
## women:typewc      -0.361104    0.716650   -0.504  0.615636
## education:typeprof  0.813183    1.807114    0.450  0.653850
## education:typewc   1.331432    3.024920    0.440  0.660930
## women:education:typeprof -0.095989  0.068361   -1.404  0.163879
## women:education:typewc  0.031253    0.070406    0.444  0.658234
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.726 on 86 degrees of freedom
## (4 observations deleted due to missingness)
## Multiple R-squared:  0.8189, Adjusted R-squared:  0.7957
## F-statistic: 35.35 on 11 and 86 DF, p-value: < 2.2e-16
## Check model!!! Always check models using standard diagnostics!!!
op <- par(mfrow = c(2, 2))
plot(mI)
```



Some basic statistics with R

```
## Hummm...
vif(mI)

##              women              education              typeprof
##          173.90365              18.58783              162.16776
##          typewc              women:education              women:typeprof
##          297.88140              349.31827              424.28346
##          women:typewc              education:typeprof              education:typewc
##          630.10116              233.17133              330.62234
## women:education:typeprof              women:education:typewc
##          560.97523              769.17182

## The model selection itself
mI_r <- step(mI, direction = "both")

## Start:  AIC=411.95
## prestige ~ women * education * type
##
##              Df Sum of Sq    RSS    AIC
## - women:education:type  2      176.9 5310.8 411.27
## <none>                      5133.9 411.95
##
## Step:  AIC=411.27
## prestige ~ women + education + type + women:education + women:type +
##          education:type
##
##              Df Sum of Sq    RSS    AIC
## - women:type      2       8.648 5319.5 407.43
## - women:education  1       8.096 5318.9 409.42
## - education:type   2     153.257 5464.1 410.06
## <none>                      5310.8 411.27
## + women:education:type  2     176.897 5133.9 411.95
##
## Step:  AIC=407.43
## prestige ~ women + education + type + women:education + education:type
##
##              Df Sum of Sq    RSS    AIC
## - women:education  1       5.687 5325.2 405.53
## - education:type   2     163.080 5482.6 406.39
## <none>                      5319.5 407.43
## + women:type      2       8.648 5310.8 411.27
##
## Step:  AIC=405.53
## prestige ~ women + education + type + education:type
##
##              Df Sum of Sq    RSS    AIC
```

Some basic statistics with R

```
## - education:type 2 157.747 5482.9 404.39
## <none> 5325.2 405.53
## + women:education 1 5.687 5319.5 407.43
## - women 1 311.369 5636.5 409.10
## + women:type 2 6.239 5318.9 409.42
##
## Step: AIC=404.39
## prestige ~ women + education + type
##
##          Df Sum of Sq  RSS   AIC
## <none>          5482.9 404.39
## + education:type 2 157.75 5325.2 405.53
## + women:education 1 0.35 5482.6 406.39
## - women 1 257.14 5740.0 406.89
## + women:type 2 18.57 5464.3 408.06
## - type 2 818.02 6300.9 414.02
## - education 1 2635.06 8118.0 440.85

Anova(mI_r)

## Anova Table (Type II tests)
##
## Response: prestige
##          Sum Sq Df F value    Pr(>F)
## women      257.1  1  4.3615 0.039493 *
## education 2635.1  1 44.6954 1.685e-09 ***
## type        818.0  2  6.9375 0.001555 **
## Residuals 5482.9 93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

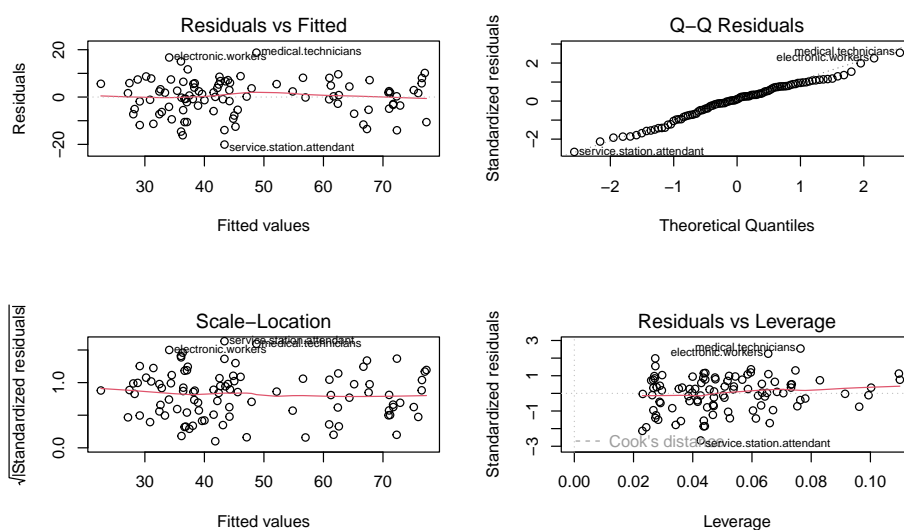
summary(mI_r)

##
## Call:
## lm(formula = prestige ~ women + education + type, data = Prestige)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.0752  -5.0445   0.7207   5.5826  18.7587
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.43988    5.73899  -0.077   0.9391
## women       -0.05783    0.02769  -2.088   0.0395 *
## education    4.43388    0.66321   6.685 1.68e-09 ***
## typeprof     7.31595    4.22235   1.733   0.0865 .
```

Some basic statistics with R

```
## typewc      -3.13070    2.86916   -1.091    0.2780
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.678 on 93 degrees of freedom
## (4 observations deleted due to missingness)
## Multiple R-squared:  0.8066, Adjusted R-squared:  0.7983
## F-statistic: 96.95 on 4 and 93 DF,  p-value: < 2.2e-16

## Check the model again
plot(mI_r)
```



```
par(op)
```

27 Side issue: a interesting case where some things went wrong without us noticing

This once happened in a class. This seems innocuous if we just quickly throw together a model:

```
mI2 <- lm(prestige ~ income * women * education * type, data = Prestige)
```

Look at the degrees of freedom: aren't you surprised? (look at, for example, the women and type terms):

```
Anova(mI2)

## Anova Table (Type II tests)
##
```

Some basic statistics with R

```
## Response: prestige
##
```

	Sum Sq	Df	F value	Pr(>F)
## income	789.00	1	23.3529	7.112e-06 ***
## women	699.16	3	6.8979	0.0003694 ***
## education	699.54	1	20.7050	2.057e-05 ***
## type	893.32	4	6.6101	0.0001324 ***
## income:women	33.14	1	0.9807	0.3252424
## income:education	7.88	1	0.2331	0.6306681
## women:education	30.36	1	0.8986	0.3462316
## income:type	653.40	2	9.6697	0.0001859 ***
## women:type	140.32	2	2.0766	0.1326023
## education:type	0.72	2	0.0107	0.9893462
## income:women:education	100.42	1	2.9722	0.0888832 .
## income:women:type	136.80	2	2.0245	0.1393087
## income:education:type	82.02	2	1.2138	0.3029069
## women:education:type	140.18	2	2.0745	0.1328633
## income:women:education:type	2.05	2	0.0303	0.9701334
## Residuals	2500.16	74		

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

The dfs seem to be wrong, and yet there are no warnings. In addition, the Sums of Squares do not match what we would get by hand:

```
m_1 <- lm(prestige ~ type * income * education, data = Prestige)
m_2 <- lm(prestige ~ type * income * education + women, data = Prestige)
## does not match women SS
## See, e.g. here for a long example of doing this by hand
## http://www.dwoll.de/r/ssTypes.php#codeII
sum(residuals(m_1)^2) - sum(residuals(m_2)^2)

## [1] 167.2936
```

As explained by John Fox (<https://stat.ethz.ch/pipermail/r-help/2018-December/460705.html>) to a question I asked, we are “(...) in the small region between where `lm()` detects a singularity and the projections used by `Anova()` break down.”

So the first thing to do is not to try to fit such a model in this ill-conditioned problem. We could see some signs of trouble by doing

```
e <- eigen(vcov(mI2))$values
max(e)/min(e)

## [1] 2.776285e+17
```

That is very close to a singular matrix.

Another warning sign, if you look for it, are the VIFs:

Some basic statistics with R

```
vif(mI2)

##              income              women
##          1421.263          4370.920
##          education          typeprof
##          299.789          1382.935
##          typewc              income:women
##          16035.172          7448.875
##          income:education          women:education
##          4601.475          9439.734
##          income:typeprof          income:typewc
##          3294.393          11787.641
##          women:typeprof          women:typewc
##          5102.885          18201.262
##          education:typeprof          education:typewc
##          2540.470          18944.229
##          income:women:education          income:women:typeprof
##          21084.678          11411.431
##          income:women:typewc          income:education:typeprof
##          13708.260          7668.479
##          income:education:typewc          women:education:typeprof
##          13354.436          8769.521
##          women:education:typewc income:women:education:typeprof
##          23087.971          25477.733
##          income:women:education:typewc
##          18694.371
```

In this case, if we center the data, as suggested by John Fox in the above email, we no longer see the above problems:

```
Prestige.c <- within(Prestige, {
  income <- income - mean(income)
  education <- education - mean(education)
  women <- women - mean(women)
})

mI2.c <- lm(prestige ~ income * women * education * type, data = Prestige.c)

## Now dfs are OK
Anova(mI2.c)

## Anova Table (Type II tests)
##
## Response: prestige
##              Sum Sq Df F value    Pr(>F)
## income      789.00  1 23.3529 7.112e-06 ***
```


Some basic statistics with R

```
## women                167.29  1  4.9516 0.0291142 *
## education            699.54  1 20.7050 2.057e-05 ***
## type                 744.30  2 11.0150 6.494e-05 ***
## income:women         33.14   1  0.9807 0.3252424
## income:education      7.88   1  0.2331 0.6306681
## women:education      30.36   1  0.8986 0.3462316
## income:type          653.40  2  9.6697 0.0001859 ***
## women:type           140.32  2  2.0766 0.1326023
## education:type        0.72   2  0.0107 0.9893462
## income:women:education 100.42  1  2.9722 0.0888832 .
## income:women:type     136.80  2  2.0245 0.1393087
## income:education:type  82.02  2  1.2138 0.3029069
## women:education:type  140.18  2  2.0745 0.1328633
## income:women:education:type 2.05  2  0.0303 0.9701334
## Residuals            2500.16 74
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Still huge
e <- eigen(vcov(mI2.c))$values
max(e)/min(e)

## [1] 3.645149e+14

m_1.c <- lm(prestige ~ type * income * education, data = Prestige.c)
m_2.c <- lm(prestige ~ type * income * education + women, data = Prestige.c)
## These do match women SS
sum(residuals(m_1.c)^2) - sum(residuals(m_2.c)^2)

## [1] 167.2936

## And how do the VIFs look?
vif(mI2.c)

##                income                women
##            810.57887            668.11751
##            education            typeprof
##            461.31022            122.58224
##            typewc                income:women
##            87.97808            616.43906
##            income:education        women:education
##            1180.24886            476.46515
##            income:typeprof          income:typewc
##            775.87657            79.04575
##            women:typeprof          women:typewc
##            236.86008            279.43939
##            education:typeprof      education:typewc
##            209.98770            16.74007
```

Some basic statistics with R

```
##          income:women:education      income:women:typeprof
##                798.76217                542.03329
##          income:women:typewc      income:education:typeprof
##                236.06942                1346.89012
##          income:education:typewc      women:education:typeprof
##                12.53033                276.27179
##          women:education:typewc income:women:education:typeprof
##                34.80346                722.53046
## income:women:education:typewc
##                26.05042
```

The message here: you can fit large models if you need to, but you need to be careful of what you do. R provides many warnings of things probably going wrong, but you still need to do your work. This includes, among other things, checking diagnostics, possible ill-conditioning problems and VIFs, etc.

28 Experimental design matters

All of the data we have seen so far have been relatively straightforward. Things aren't always this way. Suppose a situation such as this:

- 20 mice.
- 10 assigned to drug A, 10 assigned to drug B.
- Each mouse in one leg gets a corticoid ointment, on another leg gets a placebo ointment.
- What is the experimental unit?

There are, in fact, two experimental units: mouse and leg within mouse. To compare drugs we use mice. To compare ointment: we should use leg within mouse. Interactions? Can be studied, yes. How do we analyze this? This example, nicely balanced, is a classical example of a split-plot design (a type of ANOVA with multiple strata). But designs like this, and others more general, or like this but unbalanced, or like this but with additional covariates, etc, are nowadays analyzed using mixed-effects models.

This also relates to questions we asked in section on “Non-independent data” [Non-independent data](#): What if we had repeated measures on the same subjects over time? Or if we had some data that came from brothers, cousins, etc?

And how would you go about designing an experiment from scratch? What should you randomize over and what should you block over? Should you use a factorial design? Matched pairs? Should each one of two technicians each take care of half of the samples, randomly assigned to each, or should one technician deal with all male sample and the other with all the female samples? Should we start mice in each one of the four different diets in different days of the week, or should we have similar

sized groups of each diet starting every day of the week? Do you give treatment A to all even numbered samples and treatment B to all odd numbered ones, or do you randomize order? Etc, etc. And, of course, how should we allocate sample sizes to the different **levels of variation**?

Understanding what is the experimental unit is **absolutely crucial**. And sometimes things are complicated: talk to (collaborate with) a statistician as soon as you can. Some high-profile mistakes in the literature are derived from misunderstanding experimental design (a somewhat amusing half-page comment about this by G. Churchill in *Science*, 2013, v. 343, p. 370). In fact, some mistakes made during the experimental design phase just cannot be corrected later⁴⁵.

29 Covariate adjustment and a few comments about causal inference

Should we use all covariates available to us in our models? It depends on what we want to do. If all we care is prediction, then maybe yes. If we care about interpretation, probably not. In particular, variables that are effects of the outcome variable (i.e., variables that are, causally, downstream from our “y”) are often variables we do not want to adjust for. Likewise for variables that are downstream from both the “y” and other predictor variables.

We have been using language a bit to casually (I wrote “casually”, not “causally” :-). But sometimes we will read our models as saying “a unit increase in variable X1 is associate with these many units change in Y”. At other times, we will want (and even be able to say) “a unit increase in variable X1 causes these many units change in Y”. These differences in wording also emphasize the possible difficulties in understanding the meaning of phrases like “holding variable X2 constant, a unit change in variable X1 causes whatever”.

That said, it is possible sometimes to use data, both experimental and observational, to make causal claims. Yes, we wrote observational, too. And it is actually good news that we can (sometimes, and under certain assumptions) estimate the effects of causes, since many important questions that concern us are inherently causal. For example “will doing X minimize the risks of being infected by the coronavirus?”. An many of these questions (or aspects of these questions) are ones for which gathering experimental data is just not possible (ethically, logistically, etc).

The literature on causal inference has grown quite a bit in the last 10 years. Some recommended readings, in approximate order of increasing difficulty (of the reading itself, or of working through all of the material) are:

- Pearl and Mackenzie, 2018, “The book of why”.

⁴⁵R. Fisher, one of the fathers of modern statistics, as well as modern quantitative and evolutionary genetics, is often quoted in this context because he said “To consult the statistician after an experiment is finished is often merely to ask him to conduct a post mortem examination. He can perhaps say what the experiment died of.”

Some basic statistics with R

- Rosenbaum, 2017, “Observation and experiment. An introduction to causal inference” (Rosenbaum does not use DAGs)
- Pearl, Glymour, Jewell, 2016, “Causal inference in statistics: a primer” [Beware with errata: avoid early printings of the book and make sure to look at the errata page anyway!]
- Hernán and Robins. 2020. “Causal inference: what if”. Available from <https://www.hsph.harvard.edu/miguel-hernan/causal-inference-book/>.
- Morgan and Winship, “Counterfactuals and causal inference: methods and principles for social research” [Yes, basically all examples are from sociology and political science, but that should not be a problem.]

[See other file:] A separate PDF is available for this topic, [covars-interpr-causal.pdf](#). Please, read it **now**.

30 Dealing with ratios

We will only cover this in class if we have time. But many of you deal with this issue, so you might want to read it anyway ☺ .

In biology (as well as in other disciplines) it is common for researchers to use ratios of variables to try to standardize/normalize a variable. This procedure looks deceptively simple, but it is not. Here, we will explore some of the problems⁴⁶. You can find more details⁴⁷ in this commentary from Curran-Everett: <http://ajpadvan.physiology.org/cgi/doi/10.1152/advan.00053.2013>.

Before we start, though, note that this section should not really be necessary after previous sections :-).

So that nothing is hidden, I will simulate the data here, but in these notes I won't provide details about how/why the data have been simulated that way (the code is below; look at it if you want). We will pretend there is a response variable, called “Y”, two groups (“g1” and “g2”) and another variable, called “Z”. Z might be some reporter protein, or something that can be taken as a proxy for cell volume, etc.

```
set.seed(1) ## irrelevant, but so that we all
            ## get the same numbers
n <- 20
sd <- 0.5
```

30.1 A misleading case with parallel lines

⁴⁶I thank Alba Concepción, a former student of BM-1, for asking me questions that prompted me to write this section. She also provided the link to Curran-Everett's paper

⁴⁷With discussion about errors in the X variable, an issue we have not covered here

Some basic statistics with R

```
z1 <- runif(n, 1, 10)
t1 <- factor(rep(c("g1", "g2"), rep(n, 2)))
y1 <- 2 * z1 + 3 * as.numeric(t1) + rnorm(n, 0, sd)
data1 <- data.frame(Y = y1, Z = z1, Group = t1, Ratio = y1/z1)
```

```
par(mfrow = c(1, 2))
with(data1, plot(Ratio ~ Group))
with(data1, plot(Y ~ Z, col = c("red", "blue")[Group]))
abline(lm(Y ~ Z, subset(data1, Group == "g1")), col = "red")
abline(lm(Y ~ Z, subset(data1, Group == "g2")), col = "blue")
legend(x = 3, y = 20, legend = c("g1", "g2"), col = c("red", "blue"),
      pch = 1)
```

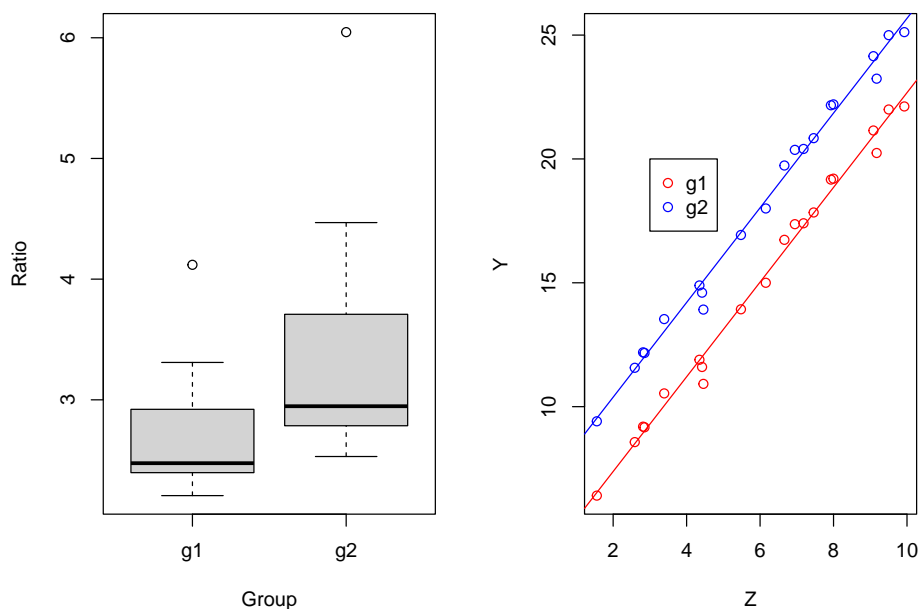


Figure 21 – Parallel slopes but ratio differences?

```
summary(lm(Y ~ Z * Group, data = data1))

##
## Call:
## lm(formula = Y ~ Z * Group, data = data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.15961 -0.17470  0.08719  0.29409  0.52719
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.559e+00  2.670e-01  13.329 1.72e-15 ***
```

Some basic statistics with R

```
## Z          1.912e+00  4.108e-02  46.538 < 2e-16 ***
## Groupg2    3.000e+00  3.776e-01   7.945 1.97e-09 ***
## Z:Groupg2  4.146e-16  5.809e-02   0.000      1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.461 on 36 degrees of freedom
## Multiple R-squared:  0.9925, Adjusted R-squared:  0.9919
## F-statistic: 1585 on 3 and 36 DF,  p-value: < 2.2e-16

t.test(Ratio ~ Group, data = data1)

##
## Welch Two Sample t-test
##
## data:  Ratio by Group
## t = -2.8577, df = 29.435, p-value = 0.007759
## alternative hypothesis: true difference in means between group g1 and group g2 is not equal to 0
## 95 percent confidence interval:
##  -1.1059664 -0.1836097
## sample estimates:
## mean in group g1 mean in group g2
##           2.676243           3.321031

## This is the equivalent to the t-test, except the
## t-test used by default by R does not assume equal variances
## You could also use t.test(..., var.equal = TRUE)
summary(aov(Ratio ~ Group, data = data1))

##           Df Sum Sq Mean Sq F value Pr(>F)
## Group      1  4.158    4.158    8.166 0.00689 **
## Residuals 38 19.346    0.509
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The lines are perfectly parallel, but the test for ratios says they differ. Why? Because it is forcing a regression through the origin. Note the the linear model does get the results right: there are no differences in the rate of change of Y relative to Z, but the groups differ in intercept. As you can see, the analysis with ratios is misleading.

30.2 A misleading case where ratios differ

Generate the data:

```
set.seed(123)
sd <- 0.1
z2 <- seq(from = 1, to = 3, length.out = n)
```

Some basic statistics with R

```
ya <- z2 + rnorm(n, 0, sd)
yb <- 0.5 * z2 + 1 + rnorm(n, 0, sd)
y <- c(ya, yb)
tf <- factor(rep(c("g1", "g2"), rep(n, 2)))
z <- rep(z2, 2)
data2 <- data.frame(Y = y, Z = z, Group = tf, Ratio = y/z)
```

The figure and analysis:

```
par(mfrow = c(1, 2))
with(data2, plot(Ratio ~ Group))
with(data2, plot(Y ~ Z, col = c("red", "blue")[Group]))
abline(lm(Y ~ Z, subset(data2, Group == "g1")), col = "red")
abline(lm(Y ~ Z, subset(data2, Group == "g2")), col = "blue")
legend(x = 1.5, y = 2.5, legend = c("g1", "g2"), col = c("red", "blue"),
       pch = 1)
```

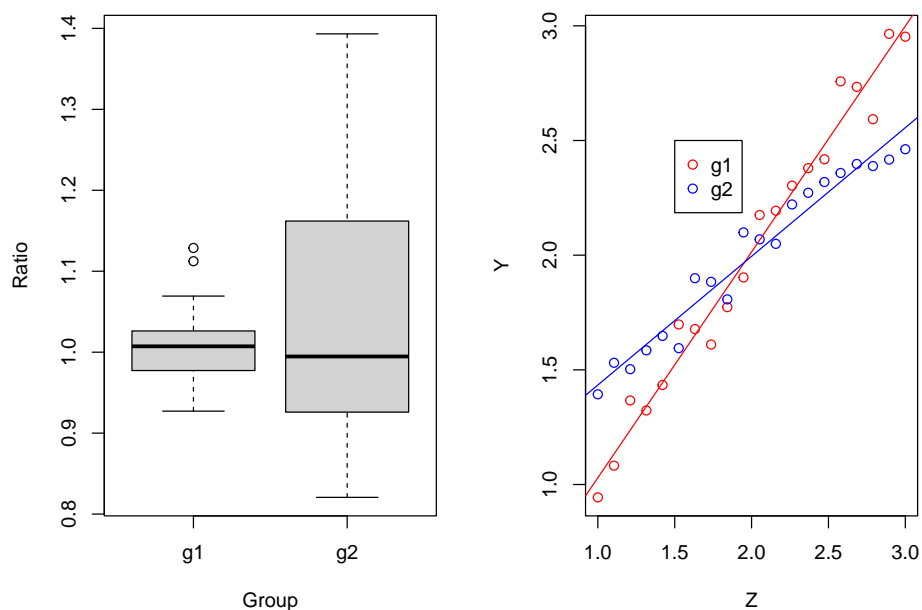


Figure 22 – Different slopes and no ratio differences?

```
summary(lm(Y ~ Z * Group, data = data2))

##
## Call:
## lm(formula = Y ~ Z * Group, data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.198791 -0.052996 -0.003768  0.049134  0.173353
```

Some basic statistics with R

```
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.04465    0.06808   0.656    0.516
## Z            0.98476    0.03258  30.230 < 2e-16 ***
## Groupg2      0.82820    0.09629   8.601 2.96e-10 ***
## Z:Groupg2    -0.42374    0.04607  -9.198 5.52e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08843 on 36 degrees of freedom
## Multiple R-squared:  0.9711, Adjusted R-squared:  0.9687
## F-statistic: 403.6 on 3 and 36 DF,  p-value: < 2.2e-16

t.test(Ratio ~ Group, data = data2) ## or do an ANOVA, as above

##
## Welch Two Sample t-test
##
## data:  Ratio by Group
## t = -0.90546, df = 22.839, p-value = 0.3747
## alternative hypothesis: true difference in means between group g1 and group g2 is not equal to
## 95 percent confidence interval:
##  -0.11757840  0.04600486
## sample estimates:
## mean in group g1 mean in group g2
##           1.008799           1.044586
```

The ratios do differ, if we do not force them through the origin: the rate of change of Y relative to Z differs between the two groups. But just comparing the ratios will not detect it. Again, a linear model does detect it just fine: you see a strong interaction between slope and group. And, again, the analysis of ratios is misleading.

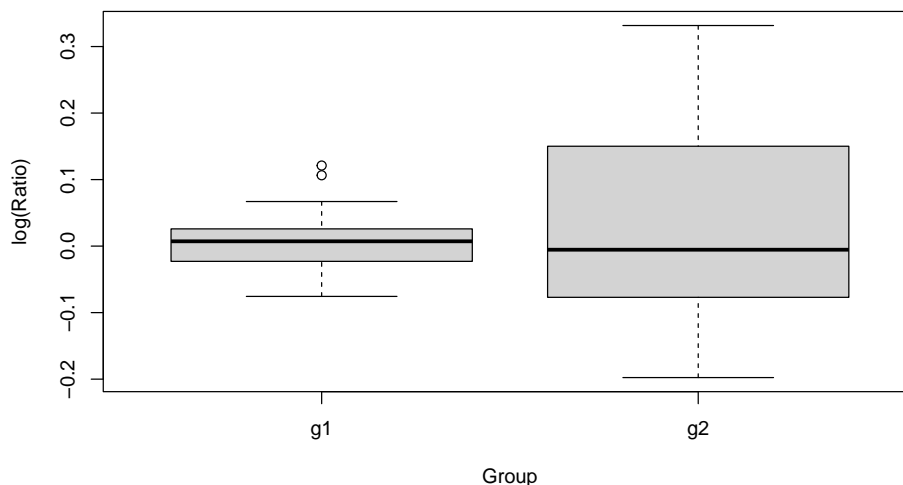
Log-transforming the ratio, using non-parametric statistics, etc, won't help; the problem is using the ratio.

```
t.test(log(Ratio) ~ Group, data = data2)

##
## Welch Two Sample t-test
##
## data:  log(Ratio) by Group
## t = -0.66269, df = 23.28, p-value = 0.514
## alternative hypothesis: true difference in means between group g1 and group g2 is not equal to
## 95 percent confidence interval:
##  -0.10049597  0.05170603
## sample estimates:
```


Some basic statistics with R

```
## mean in group g1 mean in group g2
##      0.007431795      0.031826766
with(data2, plot(log(Ratio) ~ Group))
```



```
wilcox.test(Ratio ~ Group, data = data2)

##
## Wilcoxon rank sum exact test
##
## data: Ratio by Group
## W = 196, p-value = 0.9254
## alternative hypothesis: true location shift is not equal to 0
```

30.3 Diagnostics et al. and other warning signs

If you look closely, the data show warning signals, such as possible differences in variances of ratios. And if you have no idea about the model, you might be able to compare several and use diagnostics, as explained in section [Diagnostics](#) to choose among models. But, in general, ratios are not the way to analyze the data. And using diagnostics might require a decently large sample size. Summary: only use ratios if you really know what you are doing and have good reasons to do it. Otherwise, use linear models.

By the way: we have covered just two very simple examples. Things can of course get a lot more complicated with non-linear relationships, etc.

31 Additional reading and what next

Many books have been devoted to linear models, ANOVA, et al. And in R (not necessarily through R Commander) there is a wide variety of procedures implemented. To begin with, look at the great book by Fox and Weisberg “An R companion to applied regression” (now in its third edition) and Faraway’s “Linear models with R” (in its second edition). Take also a look at chapters 6 and 7 of Dalgaard’s “Introductory statistics with R” and chapters 8 and 9 of Kabakoff’s “R in action”. The on-line book “ANOVA and Mixed Models: A Short Intro Using R” (<https://stat.ethz.ch/~meier/teaching/anova/index.html>) includes ANOVA and mixed models, and includes discussion of experimental designs we have not covered (though sometimes mentioned) such as split-plot designs and complete and incomplete block designs.

This should get you going. From here, you will probably want to look at generalized linear models, mixed effects models, nonlinear models, generalized additive models, and survival analysis, which are some key extensions of linear models that you might want to use with your own data.

A What if we did not recode training?

Suppose we had not recoded training but we had fitted a linear model. This would have happened:

```
lmMITnofactor <- lm(activ ~ training, data = dmit)
summary(lmMITnofactor)

##
## Call:
## lm(formula = activ ~ training, data = dmit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.61093 -0.72914 -0.06266  0.64855  1.86234
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8876     0.3810   2.330  0.0245 *
## training      0.9234     0.1584   5.829 6.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8802 on 44 degrees of freedom
## Multiple R-squared:  0.4357, Adjusted R-squared:  0.4229
## F-statistic: 33.97 on 1 and 44 DF, p-value: 6.024e-07
```

Some basic statistics with R

```
Anova(lmMITnofactor)

## Anova Table (Type II tests)
##
## Response: activ
##           Sum Sq Df F value    Pr(>F)
## training  26.320  1  33.973 6.024e-07 ***
## Residuals 34.088 44
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

See how the degrees of freedom for training make no sense.

B Anova tables from `lm` et al.: understanding the coefficients and parameters

We've said this: ANOVAs are a type of linear models. Thus, in R (and in most statistical packages) you can get the output for an ANOVA by different routes. Let us make sure we understand this, and can interpret the output from using the different routes available to us. In fact, we have jumped from using `aov` to `lm` several times by now.

This is also a good time to recap and make sure we understand ideas we have used like “more complex models” and “number of parameters”. Before we see the examples with code below, make sure we can understand how many parameters (and how many degrees of freedom are taken by the model) and what they represent in models like:

- A simple linear regression like $y \sim x$.
- A multiple linear regression like $y \sim x + z$.
- A one-way ANOVA like the one for the training regimes, with three possible training regimes: $y \sim \text{ftraining}$.
- A two-way ANOVA with interactions, like the one about Drug (two levels) and Diet (three levels): $y \sim \text{Drug} * \text{Diet}$.
- A two-way ANOVA without interactions, like one we might fit to the Drug and Diet data: $y \sim \text{Drug} + \text{Diet}$.

Note: the details about the exact numerical value of the coefficients can be skipped in a first reading (or in a second reading, or for the exam :-). What you definitely need to understand is how many parameters we are estimating.

If you want to continue, then let us now see a simple example:

```
asAnova <- aov(activ ~ ftraining, data = dmit)
summary(asAnova)
```

Some basic statistics with R

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## ftraining   2  31.15   15.57   22.89 1.7e-07 ***
## Residuals  43  29.26    0.68
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Not let us fit the same model using `lm`.

```
asLm <- lm(activ ~ ftraining, data = dmit)
anova(asLm)

## Analysis of Variance Table
##
## Response: activ
##           Df Sum Sq Mean Sq F value    Pr(>F)
## ftraining   2 31.147  15.5737   22.887 1.704e-07 ***
## Residuals  43 29.260   0.6805
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA table is the same. I've used `anova` but I could have used `Anova`.

But what is this output?

```
asLm

##
## Call:
## lm(formula = activ ~ ftraining, data = dmit)
##
## Coefficients:
##           (Intercept)      ftrainingLunch  ftrainingAfternoon
##           2.10300           0.09583           1.69435

summary(asLm)

##
## Call:
## lm(formula = activ ~ ftraining, data = dmit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9030 -0.5151 -0.1642  0.5647  1.7227
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.10300    0.24872   8.455 1.08e-10 ***
## ftrainingLunch  0.09583    0.34434   0.278   0.782
```

Some basic statistics with R

```
## ftrainingAfternoon 1.69435    0.30240    5.603 1.38e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8249 on 43 degrees of freedom
## Multiple R-squared:  0.5156, Adjusted R-squared:  0.4931
## F-statistic: 22.89 on 2 and 43 DF,  p-value: 1.704e-07
```

We are being shown the fitted coefficients. They are expressed as deviations with respect to a baseline level, in this case the first level:

```
means <- with(dmit, tapply(activ, ftraining, mean)) ## instead of "tapply",
                                                    ## "by" or "aggregate"
                                                    ## would
                                                    ## also work here

means[1]

## Morning
##    2.103

means[2] - means[1]

##      Lunch
## 0.09583333

means[3] - means[1]

## Afternoon
## 1.694348

rm(means) ## let's remove it, so as not to leave
          ## garbage around
```

This is the default parameterization in R. But it is not the only one available.

Of course, the above applies to more than one factor, and to factors with an arbitrary number of levels.

B.1 Changing the reference in the one-way

We can change the reference, and that will change what we are comparing against:

```
dmitb <- dmit
dmitb$ftraining2 <- factor(dmitb$ftraining,
                           levels = c("Afternoon", "Lunch", "Morning"))
## check
with(dmitb, table(ftraining, ftraining2))

##           ftraining2
```

Some basic statistics with R

```
## ftraining  Afternoon Lunch Morning
## Morning      0      0      11
## Lunch        0     12      0
## Afternoon    23      0      0
```

Now, rerun the analysis

```
asLm2 <- lm(activ ~ ftraining2, data = dmitb)
anova(asLm2) ## no difference, of course

## Analysis of Variance Table
##
## Response: activ
##          Df Sum Sq Mean Sq F value    Pr(>F)
## ftraining2  2 31.147  15.5737   22.887 1.704e-07 ***
## Residuals 43 29.260   0.6805
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(asLm2)

##
## Call:
## lm(formula = activ ~ ftraining2, data = dmitb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9030 -0.5151 -0.1642  0.5647  1.7227
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.7973     0.1720   22.077 < 2e-16 ***
## ftraining2Lunch  -1.5985     0.2938   -5.442 2.36e-06 ***
## ftraining2Morning -1.6943     0.3024   -5.603 1.38e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8249 on 43 degrees of freedom
## Multiple R-squared:  0.5156, Adjusted R-squared:  0.4931
## F-statistic: 22.89 on 2 and 43 DF, p-value: 1.704e-07

meansb <- with(dmitb, tapply(activ, ftraining2, mean))
meansb[1]

## Afternoon
##  3.797348

meansb[2] - meansb[1]
```

Some basic statistics with R

```
##      Lunch
## -1.598514

meansb[3] - meansb[1]

##      Morning
## -1.694348

rm(meansb)
```

If we go back to [A very simple two-way ANOVA](#) we can also do the same and change the reference, and that changes what we are comparing. For instance, let us recode A and run the model again in those data:

```
df1b <- df1
df1b$A <- factor(df1b$A, levels = c("a2", "a1"))
table(df1b$A, df1b$A) ## double check

##
##      a1 a2 a3
## a2    0  2  0
## a1    2  0  0
```

```
summary(lm(y ~ A + B, data = df1b))

##
## Call:
## lm(formula = y ~ A + B, data = df1b)
##
## Residuals:
##      1      2      4      5
## -0.09128  0.09128  0.09128 -0.09128
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.71624    0.15810   4.530   0.138
## Aa1           -0.45692    0.18255  -2.503   0.242
## Bb2           -0.02286    0.18255  -0.125   0.921
##
## Residual standard error: 0.1826 on 1 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.8626, Adjusted R-squared:  0.5879
## F-statistic:  3.14 on 2 and 1 DF,  p-value: 0.3706
```

B.2 Coefficients with two-ways

What about two-way models? Lets us first play with another fake data set:

Some basic statistics with R

```
y <- c(1:9, 20, 21, 22)
X <- rep(rep(c("x1", "x2"), c(3, 3)), 2)
U <- rep(c("u1", "u2"), c(6, 6))
anova(lm(y ~ U * X)) ## so strong evidence of interaction

## Analysis of Variance Table
##
## Response: y
##          Df Sum Sq Mean Sq F value    Pr(>F)
## U          1     363      363    363 5.964e-08 ***
## X          1     192      192    192 7.115e-07 ***
## U:X         1       75       75     75 2.457e-05 ***
## Residuals   8         8         1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## The cell means
tapply(y, list(X, U), mean)

##      u1 u2
## x1  2  8
## x2  5 21

## The estimates
summary(lm(y ~ X * U))

##
## Call:
## lm(formula = y ~ X * U)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
##     -1.00    -1.00     0.00     1.00     1.00
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.0000    0.5774   3.464 0.00852 **
## Xx2            3.0000    0.8165   3.674 0.00627 **
## Uu2            6.0000    0.8165   7.348 8.01e-05 ***
## Xx2:Uu2        10.0000    1.1547   8.660 2.46e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1 on 8 degrees of freedom
## Multiple R-squared:  0.9875, Adjusted R-squared:  0.9828
## F-statistic: 210 on 3 and 8 DF, p-value: 6.053e-08
```


Some basic statistics with R

```
## The intercept is the first cell mean.  
## The right and bottom cell:  
## the X2:U2 =  
21 - (2 + 3 + 6)  
  
## [1] 10  
  
## The coefficient for x2 is the difference between the intercept and  
## the first column of the second row:  
5 - 2  
  
## [1] 3  
  
## The coefficient for u2 is the difference between the intercept  
## and the first row of the second column:  
8 - 2  
  
## [1] 6
```

So things here are easy: a saturated model with interactions has a many parameters as cell means and once we figure out what is the reference, we can see what each coefficient means.

What about additive models? This is slightly more complicated:

```
summary(mxu <- lm(y ~ X + U))  
  
##  
## Call:  
## lm(formula = y ~ X + U)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -3.5    -2.5     0.0     2.5     3.5   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)  -0.500      1.518  -0.329  0.749468      
## Xx2           8.000      1.753   4.563  0.001361 **     
## Uu2          11.000      1.753   6.274  0.000145 ***    
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 3.037 on 9 degrees of freedom  
## Multiple R-squared:  0.8699, Adjusted R-squared:  0.841   
## F-statistic: 30.09 on 2 and 9 DF,  p-value: 0.0001033  
  
model.matrix(mxu)
```

Some basic statistics with R

```
##      (Intercept) Xx2 Uu2
## 1          1    0    0
## 2          1    0    0
## 3          1    0    0
## 4          1    1    0
## 5          1    1    0
## 6          1    1    0
## 7          1    0    1
## 8          1    0    1
## 9          1    0    1
## 10         1    1    1
## 11         1    1    1
## 12         1    1    1
## attr(,"assign")
## [1] 0 1 2
## attr(,"contrasts")
## attr(,"contrasts")$X
## [1] "contr.treatment"
##
## attr(,"contrasts")$U
## [1] "contr.treatment"

mean(y) ## 9

## [1] 9

## What is the first cell? the overall mean with the effect of X:1 and
## U:1, which is the overall mean minus half the effects of X:2 and U:2

9 - 11/2 - 8/2 ## Where 11 and 8 are coming from the fitted model
## [1] -0.5

fitted(mxu) ## Yes: fitted for first group are the intercept term

##      1      2      3      4      5      6      7      8      9     10     11     12
## -0.5 -0.5 -0.5  7.5  7.5  7.5 10.5 10.5 10.5 18.5 18.5 18.5

## But why the 11 and the 8 above?
## Again, look here:
(mxu <- tapply(y, list(X, U), mean))

##      u1 u2
## x1  2  8
## x2  5 21

## or here
aggregate(y ~ X * U, FUN = mean)

##      X  U  y
```

Some basic statistics with R

```
## 1 x1 u1 2
## 2 x2 u1 5
## 3 x1 u2 8
## 4 x2 u2 21

## And our model says: an overall mean, and row and column deviations.
## Let's write it.
## Row effects, or effect of X:2 (effect of X:1 = - effect of X:2)
## is the average of the row differences at each column
## or average of (5 - 2) and (21 - 8) which is the effect of X:2
0.5 * ((5 - 2) + (21 - 8)) # = 8

## [1] 8

## or, similarly
mean(mxu[2, ] - mxu[1, ])

## [1] 8

## Similar for column effects, or the effect of U:2
## effect of U:2:
0.5 * ((8 - 2) + (21 - 5)) # = 11

## [1] 11

## or, similarly
mean(mxu[, 2] - mxu[, 1])

## [1] 11

## So the first cell is the first cell UNDER that additive model. You can't
## just put the first cell mean in there.
```

What if we go to the data in section [A very simple two-way ANOVA](#)? Again, in the interaction case things are easiest:

```
(means <- with(df1, tapply(y, list(A, B), mean)))

##           b1           b2
## a1 0.1680415 0.3277343
## a2 0.8075164 0.6021007
## a3 0.3849424 0.6043941
```

```
summary(m2 <- lm(y ~ A * B, data = df1))

##
## Call:
## lm(formula = y ~ A * B, data = df1)
##
## Residuals:
```

Some basic statistics with R

```
## ALL 6 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.16804         NaN     NaN     NaN
## Aa2          0.63947         NaN     NaN     NaN
## Aa3          0.21690         NaN     NaN     NaN
## Bb2          0.15969         NaN     NaN     NaN
## Aa2:Bb2      -0.36511         NaN     NaN     NaN
## Aa3:Bb2       0.05976         NaN     NaN     NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1, Adjusted R-squared:      NaN
## F-statistic:      NaN on 5 and 0 DF,  p-value: NA

## each main effect
means[1, 2] - means[1, 1]

## [1] 0.1596928

means[2, 1] - means[1, 1]

## [1] 0.6394749

## interaction
means[2, 2] - ( means[1, 1] + (m2$coefficients[2] + m2$coefficients[3]))

##           Aa2
## -0.4223165
```

What about additive model?

```
## Overall mean:
mean(df1$y)

## [1] 0.4824549

## Note what are the estimates of the effects of A and B: the mean deviation
## of the second level from the first:
## A2
mean(means[2, ] - means[1, ])

## [1] 0.4569206

## B2
mean(means[, 2] - means[, 1])

## [1] 0.05790959

## Intercept
mean(df1$y) -
```

Some basic statistics with R

```
0.5 * (mean(means[2, ] - means[1, ])) -  
0.5 * mean(means[, 2] - means[, 1])  
  
## [1] 0.2250398
```

B.3 Other contrasts

We are using `contr.treatment`, the default in R. There are other types. In particular, and when we want to estimate effects in the presence of interactions, we probably want to use `contr.sum`. We will not pursue this any further here.

To give you a quick taste, this might do:

(We will use `contr.Sum` from `car`, since clearer labeling)

```
opt <- options(contrasts = c("contr.Sum", "contr.poly"))  
m11 <- lm(y ~ A + B, data = df1)  
anova(m11)  
  
## Analysis of Variance Table  
##  
## Response: y  
##          Df    Sum Sq  Mean Sq F value Pr(>F)  
## A          2 0.209224  0.104612   3.9552 0.2018  
## B          1 0.005030  0.005030   0.1902 0.7053  
## Residuals  2 0.052898  0.026449  
  
summary(m11)  
  
##  
## Call:  
## lm(formula = y ~ A + B, data = df1)  
##  
## Residuals:  
##          1          2          3          4          5          6  
## -0.05089  0.13166 -0.08077  0.05089 -0.13166  0.08077  
##  
## Coefficients:  
##          Estimate Std. Error t value Pr(>|t|)  
## (Intercept)  0.48245    0.06639   7.267  0.0184 *  
## A[S.a1]     -0.23457    0.09390  -2.498  0.1298  
## A[S.a2]      0.22235    0.09390   2.368  0.1414  
## B[S.b1]     -0.02895    0.06639  -0.436  0.7053  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.1626 on 2 degrees of freedom
```

Some basic statistics with R

```
## Multiple R-squared:  0.802, Adjusted R-squared:  0.505
## F-statistic:    2.7 on 3 and 2 DF,  p-value: 0.2818
```

```
(overallMean <- mean(df1$y))

## [1] 0.4824549

mA <- with(df1, tapply(y, A, mean))
mA - overallMean

##           a1           a2           a3
## -0.23456697  0.22235365  0.01221332

mB <- with(df1, tapply(y, B, mean))
mB - overallMean

##           b1           b2
## -0.02895479  0.02895479
```

Interaction now (note the estimates!)

```
m12 <- lm(y ~ A * B, data = df1)
anova(m12)

## Warning in anova.lm(m12):  ANOVA F-tests on an essentially perfect fit
are unreliable

## Analysis of Variance Table
##
## Response: y
##           Df    Sum Sq  Mean Sq F value Pr(>F)
## A           2  0.209224  0.104612     NaN   NaN
## B           1  0.005030  0.005030     NaN   NaN
## A:B          2  0.052898  0.026449     NaN   NaN
## Residuals   0  0.000000      NaN

summary(m12)

##
## Call:
## lm(formula = y ~ A * B, data = df1)
##
## Residuals:
## ALL 6 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.48245         NaN     NaN     NaN
## A[S.a1]       -0.23457         NaN     NaN     NaN
```

Some basic statistics with R

```
## A[S.a2]          0.22235      NaN      NaN      NaN
## B[S.b1]          -0.02895      NaN      NaN      NaN
## A[S.a1]:B[S.b1]  -0.05089      NaN      NaN      NaN
## A[S.a2]:B[S.b1]  0.13166      NaN      NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1, Adjusted R-squared:      NaN
## F-statistic:      NaN on 5 and 0 DF,  p-value: NA
```

Return contrasts to usual state

```
options(opt)
```

B.4 Changing the reference in two-ways

If we change the levels or references, of course the interpretation of the coefficients must also change. This is nothing new relative to what we saw for the one-way.

B.5 Unbalanced case

These examples have used balanced data. Things get trickier with unbalanced data. The idea of this section is to get an intuitive understanding of what the numbers mean, but then for real you'll do this with a computer.

C A two-way ANOVA with interaction: SS, coefficients

Here are some details about the sums of squares of the interaction term or a two-way Anova with interactions. This section is not required material.

Some on-line material that covers this is (only the first uses R explicitly):

- [https://stats.libretexts.org/Courses/Taft_College/PSYC_2200%3A_Elementary_Statistics_for_Behavioral_and_Social_Sciences_\(Oja\)/Unit_2%3A_Mean_Differences/13%3A_Factorial_ANOVA_\(Two-Way\)/13.03%3A_Two-Way_ANOVA_Summary_Table/13.03.01%3A_Calculating_Sum_of_Squares_for_the_Factorial_ANOVA_Summary_Table](https://stats.libretexts.org/Courses/Taft_College/PSYC_2200%3A_Elementary_Statistics_for_Behavioral_and_Social_Sciences_(Oja)/Unit_2%3A_Mean_Differences/13%3A_Factorial_ANOVA_(Two-Way)/13.03%3A_Two-Way_ANOVA_Summary_Table/13.03.01%3A_Calculating_Sum_of_Squares_for_the_Factorial_ANOVA_Summary_Table)
- <https://www.itl.nist.gov/div898/handbook/prc/section4/prc437.htm>
- <https://www.stat.purdue.edu/~boli/stat512/lectures/topic7.pdf>

Create a data set

```
y <- c(2, 3, 4, 0, 1, 2, 6, 7, 8, 1, 2, 3)
A <- c(rep("a1", 6), rep("a2", 6))
```

Some basic statistics with R

```
B <- rep(c(rep("b1", 3), rep("b2", 3)), 2)
df1 <- data.frame(y, A = factor(A), B = factor(B))
## Do not leave junk around and avoid confusion
rm(y, A, B)

## Create another data that is identical, but uses a different
## level as reference for parameterization
df2 <- df1
df2$A <- factor(df2$A, levels = c("a2", "a1"))
```

Table of cell means:

```
xtabs(y ~ A + B, data = aggregate(y ~ A + B, data = df1, mean))

##      B
## A    b1 b2
## a1   3  1
## a2   7  2
```

Fit the model

```
m1 <- lm(y ~ A * B, data = df1)

## Same model
m2 <- lm(y ~ A * B, data = df2)
```

Of course, we have identical ANOVA tables

```
## Identical ANOVA tables
Anova(m1)

## Anova Table (Type II tests)
##
## Response: y
##          Sum Sq Df F value    Pr(>F)
## A          18.75  1    18.75 0.0025116 **
## B          36.75  1    36.75 0.0003018 ***
## A:B           6.75  1     6.75 0.0317125 *
## Residuals    8.00  8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova(m2)

## Anova Table (Type II tests)
##
## Response: y
```


Some basic statistics with R

```
##           Sum Sq Df F value    Pr(>F)
## A           18.75  1    18.75 0.0025116 **
## B           36.75  1    36.75 0.0003018 ***
## A:B           6.75  1     6.75 0.0317125 *
## Residuals     8.00  8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

But the actual coefficients differ, of course, since the reference is different:

```
S(m1)

## Call: lm(formula = y ~ A * B, data = df1)
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0000     0.5774   5.196 0.000826 ***
## Aa2           4.0000     0.8165   4.899 0.001195 **
## Bb2          -2.0000     0.8165  -2.449 0.039969 *
## Aa2:Bb2       -3.0000     1.1547  -2.598 0.031712 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1 on 8 degrees of freedom
## Multiple R-squared:  0.8861
## F-statistic: 20.75 on 3 and 8 DF,  p-value: 0.0003946
##   AIC   BIC
## 39.19 41.61

S(m2)

## Call: lm(formula = y ~ A * B, data = df2)
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.0000     0.5774  12.124 1.98e-06 ***
## Aa1          -4.0000     0.8165  -4.899 0.001195 **
## Bb2          -5.0000     0.8165  -6.124 0.000282 ***
## Aa1:Bb2       3.0000     1.1547   2.598 0.031712 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1 on 8 degrees of freedom
## Multiple R-squared:  0.8861
## F-statistic: 20.75 on 3 and 8 DF,  p-value: 0.0003946
##   AIC   BIC
```

Some basic statistics with R

```
## 39.19 41.61
```

The interaction coefficient for “m1” is the deviation of the cell for a2 and b2 from the expected under the additive model. Thus it is $2 - (3 + 4 - 2)$, where 3 is the reference, a1, b1; 4 is the change from going from a1 to a2 (from 3 to 7); and 2 is the change from going from b1 to b2 (3 to 1). This we saw before.

In “m2” we use a different reference and measure the coefficient as the deviation of a1:b2 from the expected: $1 - (7 - 4 - 5)$ (7 is the reference, a2:b1; and -4 and -5 are the deviations from moving to a1 and b2, respectively).

What about the SS, sums of squares, of the interaction? Let us write our model again for the means

$$\mu_{i,j} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{i,j}$$

where $(\alpha\beta)_{i,j}$ is the interaction.

How do we compute $(\alpha\beta)_{i,j}$? Just write them as deviations from the expected from the additive model:

$$(\alpha\beta)_{i,j} = \mu_{i,j} - (\mu + \alpha_i + \beta_j)$$

But we can write the interactions in terms of the marginal means in the table of means, the means of the columns and rows:

$$(\alpha\beta)_{i,j} = \mu_{i,j} - \mu_{i.} - \mu_{.j} + \mu$$

(The above are equivalent because $\alpha_i = \mu_{i.} - \mu$ and $\beta_j = \mu_{.j} - \mu$).

In our case we have:

- $\mu_{a1.} = 2$
- $\mu_{a2.} = 4.5$
- $\mu_{.b1} = 5$
- $\mu_{.b2} = 1.5$

Do it with R

```
addmargins(xtabs(y ~ A + B,
                 data = aggregate(y ~ A + B, data = df1, mean)),
           FUN = mean)

## Margins computed over dimensions
## in the following order:
## 1: A
```

Some basic statistics with R

```
## 2: B
##      B
## A      b1    b2 mean
## a1    3.00  1.00 2.00
## a2    7.00  2.00 4.50
## mean  5.00  1.50 3.25
```

Now, the sums of squares for the interaction are obtained in the usual way (see Dalgaard's text for the one-way version)

$$n \sum_{ij} (\widehat{\alpha\beta})_{i,j}^2$$

where n are the number of observations per cell, 3 in our case.

That is, of course, the same as

$$n \sum_{i,j} (\hat{\mu}_{i,j} - \hat{\mu}_{i.} - \hat{\mu}_{.j} + \hat{\mu})^2$$

In our case we have

$$3 * ((3 - 2 - 5 + 3.25)^2 + (1 - 2 - 1.5 + 3.25)^2 + (7 - 5 - 4.5 + 3.25)^2 + (2 - 1.5 - 4.5 + 3.25)^2) = \\ 3 * ((-0.75)^2 + 0.75^2 + 0.75^2 + (-0.75)^2) = 3 * 4 * 0.75^2$$

C.1 Some intuition for the interaction: estimate of coefficient vs. Sums of Squares

Why did we have an estimate of the coefficient of interaction of 3 (or -3, depending on the reference) and yet the SS is $3 * 4 * (0.75)^2$?

When we compute the sums of squares for interaction we first fit a model with an overall (or common) mean, and rows and columns effects and then estimate the deviations of all cells from it to find the sums of squares. In other words, the deviations that are squared and then summed are the deviations of all the observed cell means from a model with a common mean (the overall mean) and rows and columns effects (the marginal means) that combine additively. We are computing the variation due to interaction, the variation in cell means that deviates from the additive model. In our case, when we computed the SS each of those cell deviations terms were 0.75 or -0.75.

When we computed the coefficient of the interaction, say `Aa1:Bb2` we subtracted the mean of one cell (e.g., bottom right) from the sum of another cell (top left) as reference and the other two (bottom left and top right) as the ones that dictate the effect. So the reference and the other two cells were modelled perfectly, and all the deviation was placed in the remaining cell.

Some basic statistics with R

We just said that when we computed the SS each of those cell deviations terms were 0.75 or -0.75 ; this is the same as “spreading” the coefficient of 3 (or -3) over all four cells equally: $3/4 = 0.75$. The 3 (or the -3), in contrast, has just collapsed all of those differences in all the cells in a single cell. When computing predicted (fitted) values it makes no difference, but this is not what we want for computing sums of squares.

Note that the estimate of the coefficient 3 or -3 can change if we use the first cell or the second or ... as reference when we estimate the coefficient and this is why the coefficient is labelled as `Aa2:Bb2` or `Aa1:Bb2`, or whatever is appropriate depending on what you used as reference. None of this change of Coefficients matters when we compute the SS.

Finally, note that the estimate of the interaction, e.g., `Aa2:Bb2` is -3 , but it has an standard error estimate. The *t*-statistic is -2.598 . And its square is ... 6.75, exactly the same as the F statistic for the interaction, as we have 1 degree of freedom.

D Session info and packages used

This is the information about the version of R and packages used when producing this document:

```
sessionInfo()

## R version 4.4.1 Patched (2024-10-12 r87227)
## Platform: x86_64-pc-linux-gnu
## Running under: Debian GNU/Linux trixie/sid
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/openblas-openmp/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-openmp/libopenblas-r0.3.28.so; LAPACK version 3.11.0
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8      LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8  LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8     LC_NAME=C
##  [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Madrid
## tzcode source: system (glibc)
##
## attached base packages:
## [1] grid      tools      stats      graphics  grDevices
## [6] utils     datasets  methods   base
```

Some basic statistics with R

```
##
## other attached packages:
## [1] ellipse_0.5.0      ISwR_2.0-9
## [3] HH_3.1-52          gridExtra_2.3
## [5] latticeExtra_0.6-30 lattice_0.22-6
## [7] effects_4.2-2      multcomp_1.4-26
## [9] TH.data_1.1-2      MASS_7.3-61
## [11] survival_3.7-0     mvtnorm_1.3-1
## [13] doBy_4.6.24        tidyr_1.3.1
## [15] ggplot2_3.5.1      RcmdrMisc_2.9-1
## [17] sandwich_3.1-1     car_3.1-3
## [19] carData_3.0-5      BiocStyle_2.33.1
## [21] patchSynctex_0.1-4 stringr_1.5.1
## [23] knitr_1.48
##
## loaded via a namespace (and not attached):
## [1] DBI_1.2.3           deldir_2.0-4
## [3] readxl_1.4.3        rlang_1.1.4
## [5] magrittr_2.0.3      e1071_1.7-16
## [7] compiler_4.4.1      reshape2_1.4.4
## [9] png_0.1-8           vctrs_0.6.5
## [11] pkgconfig_2.0.3     fastmap_1.2.0
## [13] backports_1.5.0     labeling_0.4.3
## [15] utf8_1.2.4          promises_1.3.0
## [17] rmarkdown_2.28      haven_2.5.4
## [19] nloptr_2.1.1        tinytex_0.53
## [21] purrr_1.0.2         xfun_0.48
## [23] Rmpfr_0.9-5         later_1.3.2
## [25] highr_0.11          gmp_0.7-5
## [27] jpeg_0.1-10         Deriv_4.1.6
## [29] broom_1.0.7         cluster_2.1.6
## [31] R6_2.5.1            vcd_1.4-13
## [33] stringi_1.8.4       RColorBrewer_1.1-3
## [35] boot_1.3-31         rpart_4.1.23
## [37] lmtest_0.9-40       cellranger_1.1.0
## [39] estimability_1.5.1  Rcpp_1.0.13
## [41] modelr_0.1.11       zoo_1.8-12
## [43] base64enc_0.1-3     httpuv_1.6.15
## [45] Matrix_1.7-0        splines_4.4.1
## [47] nnet_7.3-19         tidyselect_1.2.1
## [49] rstudioapi_0.16.0   abind_1.4-8
## [51] yaml_2.3.10         codetools_0.2-20
## [53] plyr_1.8.9          tibble_3.2.1
## [55] shiny_1.9.1         withr_3.0.1
## [57] evaluate_1.0.1      foreign_0.8-87
```

Some basic statistics with R

```
## [59] proxy_0.4-27      survey_4.4-2
## [61] pillar_1.9.0      BiocManager_1.30.25
## [63] checkmate_2.3.2   nortest_1.0-4
## [65] insight_0.20.5    generics_0.1.3
## [67] hms_1.1.3         munsell_0.5.1
## [69] scales_1.3.0      minqa_1.2.8
## [71] xtable_1.8-4      leaps_3.2
## [73] class_7.3-22      glue_1.8.0
## [75] Hmisc_5.1-3       interp_1.1-6
## [77] data.table_1.16.2 lme4_1.1-35.5
## [79] forcats_1.0.0     cowplot_1.1.3
## [81] mitools_2.4       colorspace_2.1-1
## [83] nlme_3.1-166      htmlTable_2.4.3
## [85] Formula_1.2-5     cli_3.6.3
## [87] fansi_1.0.6       dplyr_1.1.4
## [89] gtable_0.3.5      digest_0.6.37
## [91] htmlwidgets_1.6.4 farver_2.1.2
## [93] htmltools_0.5.8.1 lifecycle_1.0.4
## [95] mime_0.12         microbenchmark_1.5.0
```