

A Crash Course in Practical Data Analysis*

Sasha Hafner[†]

April 12, 2022

*For the latest version, visit <https://github.com/sashahafner/CCPDA>

[†]sasha@hafnerconsulting.com

Contents

1	Orientation	3
2	Data analysis steps	3
3	Spreadsheets and programming languages	4
4	Working with organized data	6
5	Tracking yourself	8
6	Inferential statistics	9
7	Random and systematic error	11
8	Uses and abuses of statistics	12
9	Classical linear models	13
10	Before you start	16
11	Example 1: Removal efficiency in treatment wetlands	16
12	Example 2: Comparing two BMP methods	22
13	Example 3: VOC emission from silage	28
14	Problem 1. Inoculum effects on BMP	38
15	Problem 2. Wood hardness and density	40
16	Problem 3. Fruit fly longevity and sexual activity	41
17	Problem 4: Growth and nitrate accumulation by <i>Lemna minor</i>	42
18	Bibliography	44

1 Orientation

Data analysis is essential in engineering research and practice. But the topic is confusing to many, and inaccurate or completely incorrect conclusions, wasted effort, and miserable students are just too common. This short book provides a concise introduction to data analysis from a very practical perspective. In it I try to explain and demonstrate some core concepts of data analysis and also data entry, manipulation, and related activities.

Much of this book is on relatively practical “nuts and bolts” of data analysis—including structuring your data, making data analysis reproducible, and the importance of visualization through plots. Where statistical models are covered, I focus on a single relatively simple set that can be used for many types of analyses: classical linear models. This is by no means a comprehensive introduction to *anything*, except maybe what I was thinking of over the days that I wrote it. However, it does, I think, contain several useful nuggets that could serve you well as you proceed in your academic and professional career. And if I manage to help only one of you better understand and better practice data analysis, well. . . that is a pretty crappy return on my investment! I hope it helps at least a dozen readers.

There are no real prerequisites for the material presented in this book, although some coursework in introductory statistics and at least a little experience with spreadsheets and some programming language would be helpful. The examples I present were carried out using R and LibreOffice Calc, but other tools would work as well. If you find that you cannot understand the output from the statistical models used here, or that the concepts are too complicated, spend some time with a good book on statistics first and then come back. I can recommend [Zar \[1999\]](#).

Much of the material you will see in the following pages is based on my own opinion, so I think it is fair for you to ask: “Who the hell are you?” I am a scientist with training in biology and engineering, with quite a bit of practical experience in data analysis. Presently I work on problems in environmental engineering from a modeling and data analysis perspective through my consultancy¹. I started working regularly with the R language and environment in 2007, and most weeks work with R daily. I have written several packages in R.²

For better or worse, I do not have a degree in statistics nor one in programming. I am definitely not the best person to explain advanced statistical theory, but this background comes with an advantage: it means I will focus on practical statistics that are relatively accessible.

2 Data analysis steps

You can think of the process of turning laboratory measurements into informative and useful results as occurring in 6 steps:

1. Data entry (manual) or data collection (automated)
2. Data processing
3. Data manipulation
4. Data checking

¹More details at www.hafnerconsulting.com.

²Including two on CRAN: biogas, for data processing and more in biogas research, and monitoR, for automated identification of animal vocalizations. See https://cran.r-project.org/web/packages/available_packages_by_name.html for these, or <https://github.com/sashahafner> for others.

5. Data visualization

6. Data analysis

Data entry is easy to do using spreadsheet programs, and Microsoft Excel or similar open source options are convenient for data storage as well. While text files are simpler to store, search, import, and manipulate, quickly checking and correcting data is often easier with a spreadsheet. In Section 4 below you can find some advice on how to organize data you enter into a file.

The term “**data processing**” is used here to refer to transforming “raw” measurements into quantities and units that are useful, e.g., converting measured biogas volume in biochemical methane potential, or even transforming electrode potential measurements into dissolved oxygen. And data manipulation is a broad term that could cover most of the steps listed above. But here I mean changing the structure of your data—the way they are organized within data objects—in order to use them in the next two steps. (The hip term for this process has become “data wrangling”.) These steps generally require more time and effort than those that follow. If you think “data science” is cool, you had better enjoy these types of tasks! Section 3 discusses some limitations of using spreadsheets for these operations.

Data checking should be done before analysis to find obvious input errors or missing values. Calculation of simple descriptive statistics (mean, standard deviation), and extraction of minimum and maximum values can be helpful here. In R, the `summary()`³ function is an easy way to do this. In Python, `describe()` can be used. Bivariate plots can also be helpful. Unfortunately, these approaches only highlight unusual values—they can not be used to ensure there are *no* errors in a dataset.

Data visualization is the processing of plotting data, to literally look for differences, trends, or in general, patterns. Please just accept here at the start that it is essential. It is simply impossible to understand relationships among variables and catch all problems with data without plots. The examples below should make all this clear.

Data analysis is the last step, and it includes hypothesis tests through application of statistical models, perhaps as well as more mundane calculation of summaries. In this book one set of flexible statistic models is described: classical linear models (Section 9).

This book will cover all of these six steps in at least a *little* detail.

3 Spreadsheets and programming languages

R (<https://www.r-project.org/>) is a programming language and a software environment for statistical computing (Fig. 1). It happens to be my favorite tool for data analysis, partially because it is really good, and partially because I was, by chance, introduced to it a decade or two ago. Python (<https://www.python.org/>) is also quite popular. Matlab has its own advantages, and is popular in academic settings, where it may actually be required for some course work. But it is not free or open-source, and it would not be unfair to say that for data analysis, it has been eclipsed by open-source alternatives.

Box 1. What the #@!*\$ is a script?

A *script* is just a text file with some code. When doing data analysis with a programming language, it is typical to enter and save your commands in scripts, which can be run or modified later.

³Or `dfsumm()` from <https://github.com/sashahafner/jumbled>.

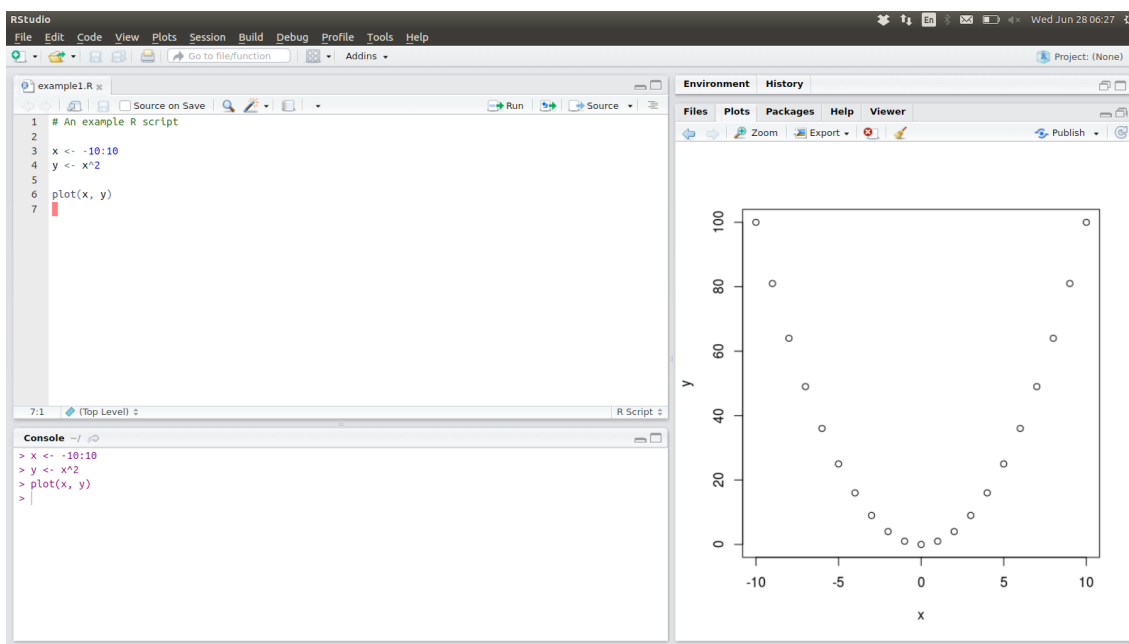


Figure 1: The RStudio IDE, which can be used for working with the R language. It is available for free for Windows, Mac, and Linux, and may be the easiest way to get started with R. You can find more details here: <https://www.rstudio.com/products/rstudio/download/>. Personally I find it much easier and more efficient to use the text editor Neovim with the Nvim-R plugin (<https://github.com/jalvesaq/Nvim-R>).

But I would guess that Microsoft Excel, a spreadsheet program, is more popular by orders of magnitude.⁴ Why? It is probably partially related to history, but one reason may be that it is very simple to *start* using Excel. Challenges come later. In contrast, getting started with a new programming language is not always easy. Also, Excel and other spreadsheet programs are cell-based. Users actually see and can manually manipulate their data. Use of a programming language requires some ability to mentally visualize data objects and their manipulation and relate these objects to symbolic variable names and commands. All this isn't trivial, but ultimately programming languages provide a lot of power for data analysis.

Based on my experience, I recommend spreadsheets for no more than data entry and storage. Anything else related to data analysis is better done in R or Python.

Box 2. Cell-based spreadsheets

Spreadsheet data analysis is cell-based. Results are generally dependent on many formulas distributed among many cells, and these cells may each have their own formulas. For even a simple statistical analysis with a small data set, this may mean hundreds of formulas. This feature makes it more difficult to write flexible “programs” and to find mistakes. In contrast, multiple rows of data are typically changed by a single line of R or Python code. Formulas in a single spreadsheet cell may refer to multiple cells, but it is difficult to make these references *scalable*, i.e., to still work correctly when the input data change by getting more variables or more observations. In contrast R or Python scripts are scalable by default, and can generally be applied to an expanded data set without any changes.

⁴The number of users is somewhere around 1 billion (<https://askwonder.com/research/number-google-sheets-users-worldwide-eoskdoxav>), while R and Python are probably below 10 million (<https://www.datanami.com/2019/08/15/is-python-strangling-r-to-death/>), although both estimates are both outdated and originally imprecise.

Box 3. Data manipulation in spreadsheets

Restructuring small datasets in a spreadsheet using cut-and-paste operations can be easy and quick. But manual manipulation becomes impractical for large datasets. And if original data change or there is a need to repeat manipulation, it is necessary to start again. In contrast, a script can simply be run again.

Although it takes a bit more effort to get started with a programming language, it takes little time to start getting dividends. You will complete tasks more quickly, probably with more accurate results, and automatically produce a detailed record of your analysis, which can be used again and again. This last point is important. There are several reasons why data analysis might need to be repeated, including correction of mistakes in the analysis itself or the original data.

R, Python, and similar languages are generally written in a script that can be saved, edited, shared, and re-run whenever changes to the analysis or input data change (see Box 1).

If you are a student trying to improve your data analysis skills and practice, what should you do? In general, I encourage students to learn a programming language, and the consensus seems to be that either R or Python is a good choice. Python and R interpreters (the software that actually reads your code and does what you tell it to do) can be downloaded and installed for free, and there is an immense population of free resources on both.

I know I won't convince many of you who are wedded to Excel to invest the necessary effort required for learning R or Python. But I do strongly encourage you to follow the advice in Section 5. And I might gently suggest that the time you would spend learning a little R or Python may quickly be repaid the first time you need to repeat data manipulation of a large dataset.

A major bonus that comes with trying a programming language is graphics capabilities. R and Python can be used to produce a huge variety of high-quality plots that are way beyond what is possible with existing spreadsheet programs. And when input data change, there is no need to copy/paste data or use a mouse to select new plot data.

Box 4. Recommendations: Spreadsheets vs. programming languages

Use spreadsheets for no more than data entry and storage. Use R or Python for everything else (see Section 2): data manipulation, checking, graphics, and analysis.

4 Working with organized data

To make your life easier and your research reproducible, the data you generate and work with should be well-organized. This refers to both organization *within* files, and organization *of* files. Data that are organized in a file in an unambiguous way are much more valuable than those that are not. They facilitate repeatable research and can vastly extend the useful life of your hard-earned measurements. Sadly, a quick search of datasets available online (e.g., Mendeley Data, <https://data.mendeley.com/research-data/>) shows that poorly organized and poorly described data are common.⁵ Of course, any electronic format is an improvement over data only stored on paper, but getting values into an electronic file is not in itself sufficient!

The following guidelines for data organization (Box 5) were originally developed for data that will be analyzed using R, Python, or similar software (see Section 3), but even if you plan to carry out all data analysis using spreadsheet software, they are still useful.

⁵Really it is a stretch to refer to some of the files available here as “datasets”.

	A	B	C	D	E	F	G	H	I	J	K
1	Initial samples (time = 0 d)										
2	CONTROL				VFAS (% of DM)						
3	Sample #	REP	WSC (% of DM)	NH3-N (% of DM)	% SUC	% LAC	%ACE	%1,2 PROP	%PROP	%ETOH	%BUT
4	782	1	6.8538168327	0.027563496279	0	0.02635782	0.10953209	0	0.18430902	0	0
5	783	2	4.8184928126	0.021054348548	0	0.03128378	0.08514032	0	0.10960043	0	0
6	784	3	6.4547007269	0.030083861071	0.02531126984	0.05508907	0.09859174	0	0.11831045	0	0
7	785	4	6.0182442559	0.031985419751	0	0.03177679	0.1161146	0	0.11535586	0	0
8											
9	LP				VFAS (% of DM)						
10	Sample #	REP	WSC (% of DM)	NH3-N (% of DM)	% SUC	% LAC	%ACE	%1,2 PROP	%PROP	%ETOH	%BUT
11	786	1	6.0149035626	0.026111100824	0.0223549376	0.05464205	0.1040861	0	0.13176641	0	0
12	787	2	6.1707269391	0.020955252073	0	0.02842632	0.08901928	0	0.13375278	0	0
13	788	3	6.0421891922	0.024968908673	0.01764531219	0.05091475	0.13863687	0	0.14247514	0	0
14	789	4	4.3164107859	0.021975191514	0	0.00696246	0.0830558	0	0.11771189	0	0
15											
16	PS				VFAS (% of DM)						
17	Sample #	REP	WSC (% of DM)	NH3-N (% of DM)	% SUC	% LAC	%ACE	%1,2 PROP	%PROP	%ETOH	%BUT
18	790	1	4.3121608914	0.02376575074	0	0.02583482	0.15956919	0	0.11794924	0	0
19	791	2	6.8534712874	0.024979879909	0.01764661214	0.03731931	0.10413928	0	0.13168528	0	0
20	792	3	6.641286909	0.034482802209	0	0.03147261	0.09311995	0	0.1360752	0	0
21	793	4	5.1577041535	0.022042234507	0	0.02399491	0.11490901	0	0.13771272	0	0
22											
23	LP+PS				VFAS (% of DM)						
24	Sample #	REP	WSC (% of DM)	NH3-N (% of DM)	% SUC	% LAC	%ACE	%1,2 PROP	%PROP	%ETOH	%BUT
25	794	1	5.5594442367	0.024309250671	0	0.03582544	0.06233151	0	0.12500716	0	0
26	795	2	6.5071930344	0.02839076182	0	0.03152608	0.07504792	0	0.13199742	0	0
27	796	3	4.6476545885	0.022590253341	0	0.02908559	0.0968207	0	0.12526258	0	0
28	797	4	3.7946128363	0.02367665723	0	0	0.16355376	0	0.12866802	0	0
29											
30											

Figure 2: An example of a poor data structure. Data are on composition of silage (fermented animal feed) from a factorial experiment.

Box 5. Guidelines for organizing data within a file

1. Header rows are only present at the top of the file
2. Each column contains a single variable
3. Each row contains a single observation
4. Each file (or worksheet) contains a single block of data

This is probably best shown by example. See the files `silage_comp_original.xlsx` and `silage_comp_restruct.xlsx` for an example. Half of the original file is shown below in Fig. 2. This file violates rules 1, 2, 3 (although it is not clear in Fig. 2, there is another set of block of data to the right), and 4. This structure is pretty easy to understand and a person not familiar with the experiment could interpret it without much trouble. But it would be very difficult to read the data into e.g., R and work with them.

The restructured file, the contents of which are shown in Fig. 3, in contrast, would be easy to work with. It follows all of the rules listed above. The only feature that that may seem a bit odd is the use of multiple header rows. This turns out to be a convenient approach, however. The first two rows provide information for understanding the data, including units and more details on the analytes. These “extra” headers are simply skipped when reading these data into R or Python. The header in row 3 contains short names easy to use in code.

Sometimes researchers have a inclination to avoid repetition in data files, and so find the value in column B in Fig. 3 to be inappropriate. Perhaps this has to do with a focus on data entry efficiency. If you have this perspective, please try to get over it! For data files, the goal isn't to produce

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1							VFAS (% of DM)							
2	Treatment	Time (d)	Sample #	REP	WSC (% of DM)	NH3-N (% of DM)	% SUC	% LAC	%ACE	%1,2 PROP	%PROP	%ETOH	%BUT	
3	trt	time	samp	rep	wsc	nh3	suc	lac	ace	prop12	prop	etoh	but	
4	CONTROL	0	782	1	6.854	0.028	0.000	0.026	0.110	0.000	0.184	0.000	0	
5	CONTROL	0	783	2	4.818	0.021	0.000	0.031	0.085	0.000	0.110	0.000	0	
6	CONTROL	0	784	3	6.455	0.030	0.025	0.055	0.099	0.000	0.118	0.000	0	
7	CONTROL	0	785	4	6.018	0.032	0.000	0.032	0.116	0.000	0.115	0.000	0	
8	LP	0	786	1	6.015	0.026	0.022	0.055	0.104	0.000	0.132	0.000	0	
9	LP	0	787	2	6.171	0.021	0.000	0.028	0.089	0.000	0.134	0.000	0	
10	LP	0	788	3	6.042	0.025	0.018	0.051	0.139	0.000	0.142	0.000	0	
11	LP	0	789	4	4.316	0.022	0.000	0.007	0.083	0.000	0.118	0.000	0	
12	PS	0	790	1	4.312	0.024	0.000	0.026	0.160	0.000	0.118	0.000	0	
13	PS	0	791	2	6.853	0.025	0.018	0.037	0.104	0.000	0.132	0.000	0	
14	PS	0	792	3	6.641	0.034	0.000	0.031	0.093	0.000	0.136	0.000	0	
15	PS	0	793	4	5.158	0.022	0.000	0.024	0.115	0.000	0.138	0.000	0	
16	LP+PS	0	794	1	5.559	0.024	0.000	0.036	0.062	0.000	0.125	0.000	0	
17	LP+PS	0	795	2	6.507	0.028	0.000	0.032	0.075	0.000	0.132	0.000	0	
18	LP+PS	0	796	3	4.648	0.023	0.000	0.029	0.097	0.000	0.125	0.000	0	
19	LP+PS	0	797	4	3.795	0.024	0.000	0.000	0.164	0.000	0.129	0.000	0	
20	CONTROL	119	798	1	0.884	0.088	0.154	4.575	0.959	0.000	0.000	0.707	0	
21	CONTROL	119	799	2	0.686	0.068	0.150	4.337	0.958	0.000	0.000	1.108	0.073	
22	CONTROL	119	800	3	0.922	0.091	0.135	4.350	1.242	0.159	0.000	0.722	0	
23	CONTROL	119	801	4	0.843	0.055	0.141	4.440	0.872	0.000	0.000	0.798	0	
24	LP	119	802	1	0.521	0.091	0.121	4.201	0.810	0.089	0.000	0.846	0.153	
25	LP	119	803	2	0.674	0.079	0.138	4.456	1.124	0.000	0.000	0.932	0	
26	LP	119	804	3	0.586	0.069	0.147	4.487	1.620	0.000	0.000	1.174	0.061	
27	LP	119	805	4	0.641	0.090	0.137	4.658	1.220	0.172	0.554	0.000	0.067	
28	PS	119	806	1	2.103	0.077	0.098	4.560	1.014	0.000	0.000	0.235	0	
29	PS	119	807	2	1.033	0.027	0.041	2.553	1.234	0.163	0.000	0.072	0	
30	PS	119	808	3	1.515	0.073	0.169	4.661	1.372	0.235	0.000	0.455	0	
31	PS	119	809	4	1.964	0.080	0.060	3.776	1.001	0.000	0.000	0.161	0	
32	LP+PS	119	810	1	0.800	0.069	0.053	5.735	0.999	0.000	0.000	0.208	0	
33	LP+PS	119	811	2	1.086	0.040	0.047	5.920	1.105	0.000	0.000	0.212	0	
34	LP+PS	119	812	3	0.656	0.040	0.060	5.848	1.080	0.000	0.000	0.411	0	
35	LP+PS	119	813	4	0.892	0.087	0.074	6.213	1.108	0.000	0.000	0.413	0	
36														

Figure 3: A better way to structure a file containing the data shown in Fig. 2. This file was manually manipulated using a mouse and keyboard in a spreadsheet program.

something beautiful.

Data that don't follow some of these rules can sometimes be restructured (reshaped) using R, Python, etc. "Manual" restructuring via cut-and-paste etc. in a spreadsheet, or even manipulation of a text file may be the only option in other cases (as with the silage data described above). If so, take care and be sure to save a copy of the original file so one could check the accuracy of the restructuring later.

Organizing files themselves presents its own significant challenges. Try to use a consistent structure for your projects. Avoid accumulating numerous copies of a file. If you worry about making changes that you will later want to undo, consider making a switch from working in spreadsheets to working with a programming language, and see Section 5. If you receive data from a collaborator or a public source, it is good practice to save a read-only copy of the original, e.g., in a sub-directory (folder) named "data/original".

5 Tracking yourself

I think it is a bit strange to have a mobile phone record my location and activities, but I am completely in favor of tracking my every move when it comes to data analysis. In fact, doing so is essential for *reproducible research*. Ideally, any conclusion you write in a paper or report should be based on an analysis that can be checked, repeated, and corrected. Data processing and manipulation are both prone to problems in this area.

Working with scripts in a programming language immediately solves part of this problem for you. Why?—because a script (see Box 1) contains a detailed description of exactly what you did. This is a major advantage of using a programming language for your work, instead of a spreadsheet.

For complex analyses, or at least after you have a little experience with a new programming language, I recommend going one step further, and tracking *changes* to your scripts. I like to use Git and GitHub for this, but there are alternatives. These are especially useful if you collaborate with others on data analysis.

Box 6. Git and GitHub

Git is a “version control system” originally used for software code. Git is an application that allows developers to track all changes in code down to the character, work on multiple versions simultaneously, and collaborate efficiently. In its simplest sense, GitHub (<https://github.com>) is simply a place for storing Git repositories. But it includes a convenient web interface and handy tools, and is now used for much more than software code (Fig. 4). All this works really well for data analysis code as well!

```

Showing 1 changed file with 6 additions and 5 deletions.

R/summBg.R
@@ -22,11 +22,6 @@ summBg <- function(
-   quiet = FALSE)
- {
-   # Argument revisions
-   if (tolower(when) == 'latest') {
-     extrap <- TRUE
-   }
-   # For "vectorized" calls, lapply-like behavior
-   if(class(vol)[1] == 'list') {
-@@ -184,8 +179,14 @@ summBg <- function(
-   checkArgClassValue(rate.crit, 'character', c('net', 'gross', 'total'))
-   checkArgClassValue(show.obs, 'logical')
-   checkArgClassValue(sort, 'logical')
+   182 +
+   183 + # Argument revisions
+   184 + if (tolower(when) == 'latest') {
+   185 +   extrap <- TRUE
+   186 + }
-   187 187 #-----
-   188 188

```

Figure 4: Example of some changes in an R script tracked with Git displayed on the GitHub website. The red code was deleted, and the green code added.

Spreadsheets do not facilitate repeating or checking any of the steps listed above, but this problem is especially acute for manual data manipulation. For calculations underlying data processing and analysis, sure, one could find the formulas underlying the calculations, and follow the cell references to check everything. But there are so many damn cells, each with its own formula! Code, in contrast, is concise by its nature. You can find a bit more discussion on this difference in Section 3. If you insist on sticking with a spreadsheet for data analysis, what can you do? In the least, you can keep a log of major changes within each spreadsheet, in a dedicated worksheet. Here it can be helpful to include the date, file name, your name, in addition to a description of the changes. This is helpful for files used for only data entry as well—if you delete or correct a cell, record that change!

6 Inferential statistics

The term “statistics” can mean a few things, but here I mean statistical methods and statistical models used for data analysis. Here I am also focused on *inferential* statistics, i.e., methods and

models meant to make some kind of inference.⁶ For example, you might like to know if your new sludge treatment method improves biogas yields. In some cases statistics can help answer a question like this, and in others, they are worse than useless. To help understand why, it can be useful to think about a basic concept in inferential statistics (Box 7).

Box 7. Inferential statistics: the basic concept

Inferential statistical methods are based comparing a measured *difference* in some variable to *random error* in that variable. If the observed difference is large compared to the error, we can *infer* that a true difference exists in the (real or hypothetical) larger population.

Typically, it is some difference in a response variable, that we are interested in. Thinking about sludge, it might be the size of the difference in methane yield between your new treatment and a reference treatment (or no treatment). Typically, we use the *mean* or average difference as our best estimate of the effect. We might have measured the results shown in Fig. 5.

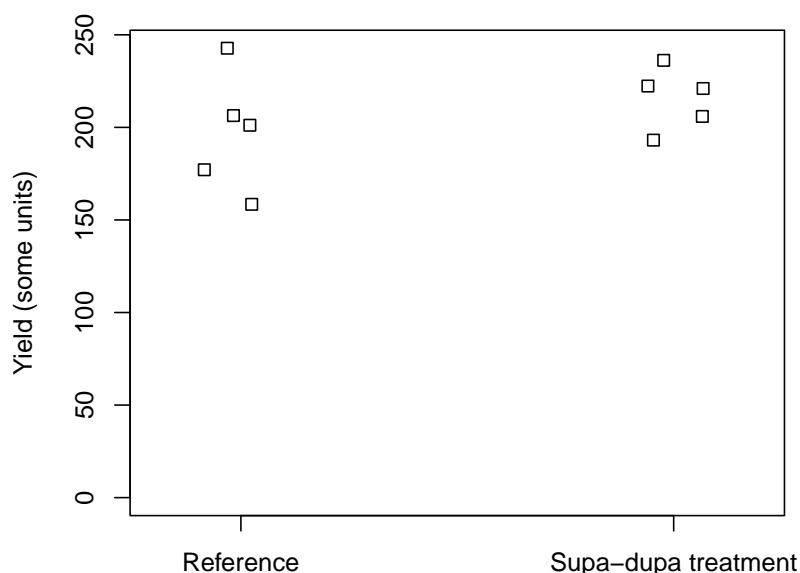


Figure 5: A comparison of biogas yields from sludge for two different treatments. Each point represents a single unit of observation.

Our mean values in this case are 197 (reference) and 216 (the new treatment). So there is a clear difference, right?! No! There is always a difference, and we might need inferential statistics to determine *if a difference really means anything*. In this case, we can immediately see that there is no meaningful difference without even applying a statistical model. Why? Because the size of the difference is small compared to the random error.⁷ There are no statistics needed to tell us this, making this example the first case where statistics are **not** useful.

The “population” about which we make inferences can be challenging to think about.

⁶Descriptive statistics, e.g., mean and standard deviation, are also useful, but the distinction should be clear.

⁷And we have a small sample. With a large number of observations we might be able to confirm the presence of a difference.

Box 8. The population concept

Statistics are used to draw conclusions about a *population*, which may be real or imaginary. The *sample* used in any experiment should reflect the population of interest.

For observational studies, where the population is real, there is no challenge. For example, if you are interested in some characteristic of engineering students, the population might be all engineering students in Germany. For experimental studies, the population is often imaginary (also called “hypothetical” or “potential” [Zar, 1999, p. 17]). Ideally, we would like to design experiments that provide information about how a process would function elsewhere. For example, if we actually found a 30% increase in methane yield from the supa-dupa process, we might hope that the same process would provide a similar improvement if applied to any sludge. But variation in sludge composition could make this unlikely, so expecting this response is risky. Perhaps then we should consider only sludge with similar characteristics, and our hypothetical population is all secondary (waste activated) sludge from municipal wastewater treatment plants. Perhaps we need to consider solids retention time also. Figure 6 in Section 8 below shows that the true population is not always as broad as we might expect.

7 Random and systematic error

Random error is important in inferential statistics, but what is it anyway? Random error is the sum of all the sources of error in our measurements which we do not completely understand and cannot completely control. In the plot above (Fig. 5), the variation in the vertical position of the points is a representation of the random error. We call it random because we cannot predict its value for any one particular observation (i.e., any single point in the plot above). But we can estimate its magnitude from our measurements. And if we assume it follows some kind of distribution, we can use this information in statistical tests. It is typical to assume (effectively define) that the expected mean value of random error is zero, and to estimate the value of random error inherent in our measurements from the observed error. When you calculate the standard deviation from a sample, you are typically quantifying random error.

Systematic error is different. It is generally repeatable, and it may be possible to assign it to a particular cause. For example, technician effects may be systematic errors. Depending on the experimental design and the analysis approach, one particular source of error could be identified (or not identified) as either random or systematic. If you are confused the examples below may help.

Random error is an integral part of inferential statistics, but systematic error can cause real problems. In applying statistical models we commonly assume that individual observations are *independent* (see Section 9, for example). If we want to make an estimate of the magnitude of random error, which is needed for a hypothesis test, we need to be sure that the appropriate random error is reflected in our measurements. If our observations are not in fact independent, results may be affected by the presence of systematic error. For example, if you wanted to compare two sludge treatments, and decided to apply one treatment in your lab and ask a friend in a different country to test the other one, you would be asking for trouble! It would be nearly certain that some of the observed differences between the two samples were due to systematic errors that were laboratory-dependent. Statistical methods are very powerful for detecting this type of effect, but assigning it to the treatment would be a major mistake.

8 Uses and abuses of statistics

Everyone knows that *replication* is important in statistics.⁸ But it must be the right kind of replication. Replicating measurements on one single experimental unit that received some treatment and one single unit that did not can give you a lot of data, but you would both tend to underestimate random error and assign a likely systematic difference to your treatment!⁹ A statistical model alone cannot tell you about the effect of the treatment in this case, and complex fancy-sounding approaches do not solve the problem (although many try, as numerous published papers show). Basic principles are important!

In other cases, it may be acceptable to apply a statistical model, but unnecessary. When a measured difference is much larger than the random error, there may not be a need to apply a statistical model. This is common in engineering, where most experiments are designed (and resulting data are experimental and not observational) and treatments have large effects.

Even when statistics are useful, they are not the complete solution. The *magnitude* of the difference is more important. In fact, we can turn the basic idea of inferential statistics on its head for many types of experiments by considering this: What is the probability that any two physical, chemical, or biological treatments you might work with to improve some process have the *exact* same effect on the process? Essentially nill. It is the magnitude of the difference that is important, so we must remember to consider both the size of the difference and the evidence we have about whether it reflects a true difference between treatments etc. Avoid feeling so satisfied by finding a “statistically significant” difference that you ignore the size of the difference. Unfortunately this very mistake is not rare (perhaps deliberately so) in research articles.

Box 9. Advice on multiple comparison tests

Avoid making many comparisons in your data analysis, especially if the predictor variable is quantitative by nature. In this case, consider using regression. Otherwise, focus on the important comparisons, possibly to a control level.

Even if you are cautious in avoiding unnecessary hypothesis tests, conventional interpretation of statistical results is being questioned, and this discussion is worth paying attention to. The use of *p*-values and in particular, the use of an arbitrary cutoff for assessing “statistical significance” has been strongly criticized over the past few decades, and statisticians have proposed (and argued about) alternatives [Wasserstein et al., 2019].¹⁰ Advice includes completely dropping the use of a fixed cutoff (commonly called α) as well as the term “statistically significant”. Instead, we might consider reporting actual *p*-values, using confidence intervals, and always considering the magnitude of any difference.

I’ll end this section with one last reminder about the limitations of experimental work. Even very clear differences observed in your laboratory may not work out the same way in other laboratories, or at pilot or full scale. Why not? For starters, there are always differences between technicians and laboratories, and particularly for biological processes or even biological conversions, many differences are difficult to measure or even observe. I and some collaborators have recently looked into the reproducibility of kinetic results extracted from BMP tests in the lab.¹¹ In many cases, results

⁸Although it is not always required in a strict sense. For example, a repeated-measures design can be considered a two-factor analysis of variance without replication [Zar, 1999]. And when using regression, it is not necessary to have multiple replicates for each level of the response variable.

⁹This is related to the concept of *pseudoreplication* or false replication, which has been discussed extensively [Hurlbert, 1984].

¹⁰This paper, actually an editorial in a special issue on the topic, provides a very interesting summary of the problem and proposed solutions [Wasserstein et al., 2019].

¹¹You can find a short presentation on this topic here: <https://www.bioenergie-events.de/cmp/program/short-presentations>.

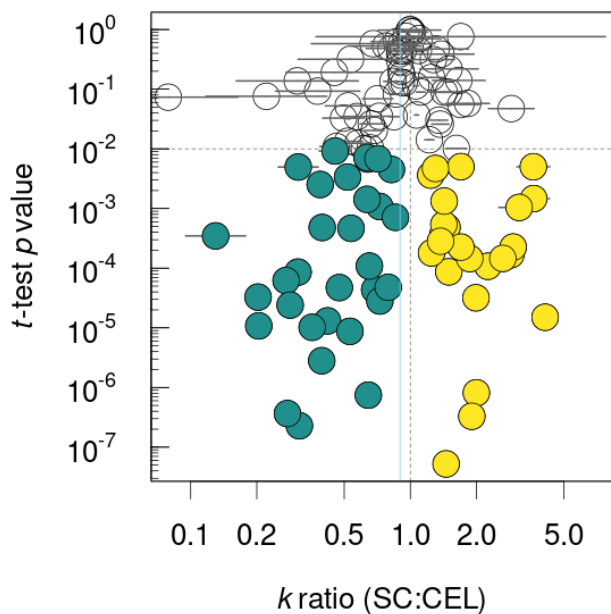


Figure 6: A demonstration of a complete lack of reproducibility in inferential statistics results among laboratories. Individual labs measured first-order rate constant for two substrates, and the ratio of these values is shown on the x axis. In many cases, differences (i.e., a ratio above or below 1) were “statistically significant” based on a t -test, shown by p -values below 0.05.

from individual laboratories could be used to show “highly significantly different” conversion rates between two substrates, e.g., p -values well below 0.001. But these results did not carry over to other laboratories, some of which had similarly “significant” results, but for the *opposite* difference (Fig. 6)! So be modest, be careful, and accept that your results just may be wrong.

9 Classical linear models

In this book I will present a single statistical method: linear regression, which we can perhaps more accurately refer to as a set of methods called “classical linear models” You can use classical linear models for much more than just simple or multiple linear regression, but the heart of the calculations are the same as in linear regression, with an analysis of variance (ANOVA) table fitted on top in some cases. I’ve selected this method because I think it is extremely flexible and useful, and because I tend to use it frequently. In this short book, I don’t have the option of including many different approaches, but another reason to focus on a single method is to use it to support what I think is an important piece of advice: focus on methods you understand.

Box 10. Advice: Don't use methods you don't understand

Access to powerful statistical models has increased in recent years, and code from the Internet can easily be copied and modified to apply to a new problem. However, I would advise you to avoid using statistical methods that you do not understand. In many cases, simpler methods will give clearer results and reduce the risk of mistakes.

It is likely that many statistical analyses you will need to do can be done using classical linear models, and they are relatively easy to understand and apply.

In R, several classical statistical models can be implemented using one function: `lm` (for linear model).¹² Python has some similar functions, including `ols()` in the statsmodel package (<https://www.statsmodels.org/v0.10.2/>). The `lm` function can be used for simple and multiple linear regression, analysis of variance (ANOVA), and analysis of covariance (ANCOVA). With data transformations and polynomials, `lm()` can easily handle (some) non-normal error distributions and non-linear responses.

The arguments for `lm` are

```
args(lm)

# function (formula, data, subset, weights, na.action, method = "qr",
#       model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE,
#       contrasts = NULL, offset, ...)
# NULL
```

The first argument, `formula`, is where you specify the basic structure of the statistical model. This approach is used in other R functions as well, such as `glm`, `gam`, and others. Venables et al. has a useful list of example formulas—some examples are repeated below. In these examples, the variables `x`, `y`, and `z` are continuous, and `A`, `B`, and `C` are factors. The response variable is `y`, and the others are predictor variables.

<code>y ~ x</code>	Simple linear regression of <code>y</code> on <code>x</code>
<code>y ~ x + z</code>	Multiple regression of <code>y</code> on <code>x</code> and <code>z</code>
<code>y ~ poly(x, 2)</code>	Second order orthogonal polynomial regression
<code>y ~ x + I(x^2)</code>	Second order polynomial regression
<code>y ~ A</code>	Single factor ANOVA
<code>y ~ A + B</code>	Two-factor ANOVA
<code>y ~ A + B + A:B</code>	Two-factor ANOVA with interaction
<code>y ~ A*B</code>	Two-factor ANOVA with interaction
<code>y ~ (A + B + C)^2</code>	Three-factor ANOVA with all first-order interactions
<code>y ~ (A + B + C)^2 - B:C</code>	As above but without <code>B:C</code> interaction
<code>y ~ A + x</code>	ANCOVA

Of course all this applies to R, but formulas in Python are similar, and even for spreadsheet programs, the concepts still apply.

¹²In R you can load help files by simply typing `?` followed by the name of the function, so `?lm` will bring up a help file on this handy function.

Box 11. Response and predictor variables

The term “dependent” variable has been widely used to refer to the variable that is (or hypothesized to be) dependent on (affected by) some other variables, called “independent”. The terms “response” and “predictor” variables doesn’t imply a dependency or not, and are therefore better terms to use.

Linear regression calculations can be used for analysis of variance (ANOVA). But predictor variables—categorical in this case, or factors—must be converted to dummy variables first.

Box 12. What are dummy variables?

Continuous predictor variables can be used as-is in regression. But factors (categorical variables) don’t work like this. Instead, with n levels, at least $n - 1$ binary (typically values of 0 or 1) “dummy variables” are used. In effect, each level of the factor is turned into its own variable. Dummy variables make it easy to handle categorical predictors through regression.

Transformations can make non-linear responses or non-normal error distributions easy to handle. In some cases, true non-linear regression is a better approach, however.

Box 13. Logarithmic transformations

A logarithmic (or log) transformation of the response variables does two important things to your model. First, it makes the effects of predictor variables *multiplicative* [Steel, 1997], so a fixed absolute change in the predictor causes (or is correlated with) a relative change in the response. Second, it changes the assumed error distribution from normal to lognormal. In many cases both are needed together, conveniently. If this is not the case, generalized linear models are an alternative [McCullagh and Nelder, 1989]. Predictor variables can also be transformed, giving three general types of model in addition to true “linear” models: log-linear, linear-log, log-log. Take care with interpretation of parameter values!

Polynomial regression can be useful for some non-linear responses.

When you apply classical linear models, you should be aware of the assumptions that are employed every time model coefficients, p -values, or confidence intervals are returned.

1. Errors are normally distributed
2. Variance is constant
3. Observations are independent

For linear regression in particular, there are two more assumptions.

1. The actual relationship is linear
2. Error in predictor variables is negligible

Some of these assumptions can be evaluated before even entering data, but others can only be evaluated after a model has been fit, somewhat ironically. Functions available for the R language make it very easy to evaluate assumptions, but even spreadsheets can be used for the task with a bit of effort. You can find an excellent introduction to the use of linear models by Julian Faraway, available for both R [Faraway, 2005] and now Python as well [Faraway, 2020].

10 Before you start

Some of the issues introduced above should be considered whenever analyzing data. I can think of a few other points that are important as well. Before analyzing your data, make some plots, think about your experiment, and ask yourself the questions listed below. This list is not explained in great detail here, but it builds on the material from earlier sections and is referred to in the examples.

1. What research question would you like you to answer? Is your sample appropriate for your question? The sample should reflect the population that you are interested in.
2. What is the unit of observation? What is the thing on which you made measurements? Clearly describe it.
3. Do you have replication, and it is the right type? Are the observations independent, apart from whatever factor you would like to test?
4. Are there systematic errors present in your data that could affect the results? If all observations have the same systematic error (e.g., you made the measurements instead of your more experienced colleague) there is generally no reason to expect an effect on a comparison. If systematic error is associated with the experimental factor, however, you have a problem that statistical models cannot (easily) solve.
5. Have you plotted your data? Explain what you see. Do you really need to apply a statistical model?
6. Is your experimental factor continuous or categorical by nature? The answer determines the way you should analyze your data and interpret results.
7. What type of a relationship do you expect (or see) between your treatments and the response variable(s)? Additive? Multiplicative? How can your approach to data analysis accommodate this? Are transformations needed? Polynomials?
8. Is there any reason to expect that errors are not normally distributed? Should you consider a transformation? It has been argued that log-normal distributions are more common than normal [Diwakar, 2017].

11 Example 1: Removal efficiency in treatment wetlands

Let's, finally, work on an example. Two treatment wetlands were created and used to compare the efficacy of wastewater treatment by two species of plants: *Phragmites australis* and *Cyperus papyrus* [García-Ávila, 2020]. The data are in the csv file `wetlands.csv` (text file with comma separators). I will use R to plot the data.

First, let's load a handy function for summarizing datasets.¹³

```
source("functions/dfsumm.R")
```

```
w1 <- read.csv('data/wetlands.csv')
```

¹³You can download this function from <https://github.com/sashahafner/jumbled>.


```
dfsumm(wl)

#
# 83 rows and 6 columns
# 83 unique rows
#
#      date parameter      unit influent  eff_cp
# Class      character character character numeric numeric
# Minimum    04-15 Alkalinity          0.31    2.1
# Maximum    07-08      TSS      µS/cm 1.6e+11 2.8e+09
# Mean       05-27    NH3.N      mg/L 5.4e+09 1.36e+08
# Unique (excl. NA)      7      12      6      75      74
# Missing values      0      0      0      4      4
# Sorted      TRUE      FALSE      FALSE      FALSE      FALSE
#
#      eff_pa
# Class      numeric
# Minimum      2.4
# Maximum    3.5e+09
# Mean       2.45e+08
# Unique (excl. NA)      72
# Missing values      4
# Sorted      FALSE
```

These data are in a structure midway between “long” and “wide”. I’ll reshape it first, and for that I need an add-on package. I’ll load the graphics and date/time packages as well.

```
library(reshape2)
library(ggplot2)
library(lubridate)
library(rmarkdown)
library(dplyr)
```

I’ll reshape these data in a couple ways.

```
wll <- melt(wl, id.vars = c('date', 'parameter', 'unit'),
            measure.vars = c('influent', 'eff_cp', 'eff_pa'),
            value.name = 'value', variable.name = 'source')
ww <- dcast(wll, date + source ~ parameter, value.var = 'value')
```

And I’ll get day of the year for plotting.

```
ww$date <- mdy(paste(ww$date, '2000'))
ww$doy <- yday(ww$date)
```

We can look at the data now. Here are the first few rows.

```
head(ww)

#      date source Alkalinity  BOD5   COD  EC      FC NH3.N
# 1 2000-04-15 influent      210.2 102.50 205.04 680 1.6e+10 22.30
```

```
ggplot(wv, aes(doy, NH3.N, colour = source)) +
  geom_line() +
  geom_point() +
  labs(x = 'Day of year', y = 'Total ammonia N conc. (mg/L)')
```

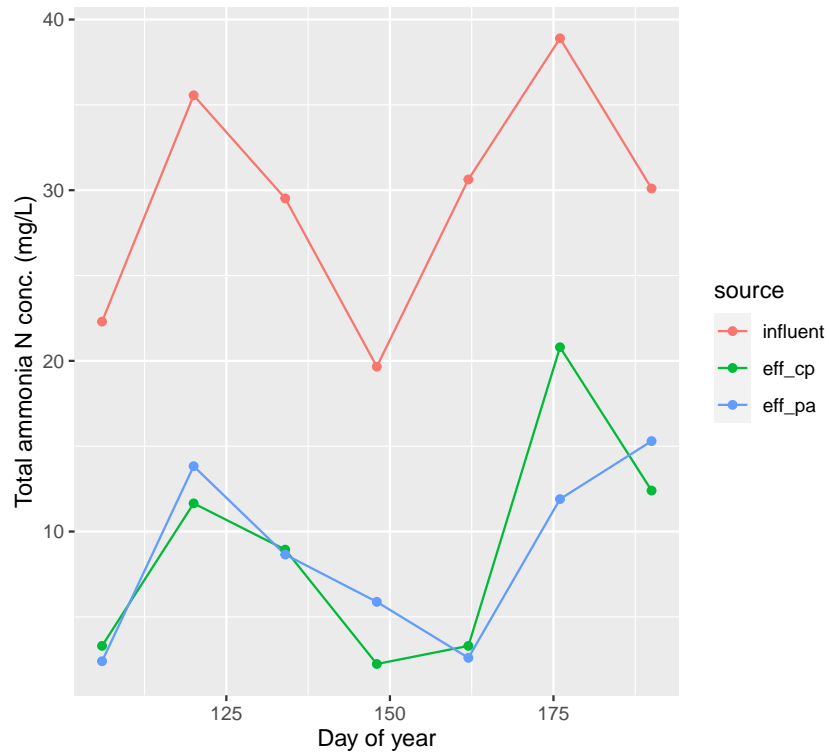


Figure 7: Ammonia concentration in influent and effluent from the two wetlands, showing removal

#	2	2000-04-15	eff_cp	126.6	29.97	89.89	545	1.6e+09	3.30
#	3	2000-04-15	eff_pa	111.6	49.90	78.04	521	1.6e+09	2.40
#	4	2000-04-29	influent	NA	89.50	280.00	772	NA	35.56
#	5	2000-04-29	eff_cp	NA	14.85	67.00	634	NA	11.65
#	6	2000-04-29	eff_pa	NA	20.30	99.00	665	NA	13.83
#									
#	1	0.605	6.79	1.6e+10	26.7	5.01	55	106	
#	2	2.105	6.19	1.6e+09	26.1	3.07	58	106	
#	3	7.615	5.96	1.6e+09	26.1	4.14	22	106	
#	4	NA	6.94	NA	23.1	7.42	78	120	
#	5	NA	6.32	NA	22.8	3.21	144	120	
#	6	NA	6.45	NA	23.0	3.75	82	120	

So we have influent composition, and the composition of effluent for the wetland with *Phragmites* and the one with *Cyperis*. Let's assume we are interested in ammonia. I'll plot ammonia concentrations over time.

It is clear that both wetlands remove ammonia—effluent concentrations are always below influent around the same date. But can we compare the two plant species? Let's start thinking about the questions in Section 10 before we answer that question.

1. We might be interested to know if the ammonia removal efficiency of wetlands planted with these two plant species differs. And we should also want to know what the removal efficiency is, i.e., its magnitude.
2. The unit of observation is a single sampling time (date) for a single wetland.
3. Yes, we have replication: we have multiple measurements for each wetland. Is it the right kind though? Umm. . .

Are observations independent? No—all the blue observations are from a single wetland, for example. Well, is the replication the right kind then? Probably not, but it depends on the question we want to answer! Presumably we are interested in whether ammonia removal differs between the two species *in general*. That is, our hypothetical population is a bunch of similar constructed wetlands with one of these two species. But to answer this question using inferential statistics, we would need multiple replicated *wetlands* for each species! So no, we do not have the right kind of replication. If we proceeded to apply a statistical model here we would really be comparing these two particular wetlands. Could we be sure that any difference is due to the plants and not some other difference between the two wetlands, e.g., retention time, some other plants, degree of aeration? Probably not.

Unfortunately this situation may be unavoidable when it is difficult or expensive to create or treat some unit of observation—think about pilot-scale reactors, for example! We do have the option of assuming that the plants are the main cause of differences between these two wetlands, but such a decision should be supported. This could be done by e.g., running the wetlands in parallel for some time before adding plants, to show there is no difference. Or, evidence could also come from other studies that show small variability among replicate wetlands. Regardless, this leap of faith should be explicitly described, if it is employed.

We can get a hint that something is amiss here by thinking about this: Are we guaranteed to see a “significant” difference eventually if we have a large enough sample size? Well, since it is impossible for two individual wetlands to be exactly the same, yes. In this particular example, we also have the problem of autocorrelation within time series measurements; measurements made closer together are more likely to be similar.

In this particular case, we do not have enough knowledge to know if it is appropriate, and we didn’t invest the money and effort in collecting the data, so it is easy to simply say they cannot be used in this way. Furthermore, we can see in the plot that there is no consistent difference—so there is clearly no reason to apply a statistical model here. The only value I see in these data is in presenting the estimates of average removal efficiency. To calculate mean values, it is easiest if we have the different wetlands in separate columns.

```
head(wl)
```

#	date	parameter	unit	influent	eff_cp	eff_pa
# 1	04-15	pH		6.79	6.19	5.96
# 2	04-15	Temperature	°C	26.70	26.10	26.10
# 3	04-15	Alkalinity	mg/L, CaCO3	210.20	126.60	111.60
# 4	04-15	EC	µS/cm	680.00	545.00	521.00
# 5	04-15	TSS	mg/L	55.00	58.00	22.00
# 6	04-15	BOD5	mg/L	102.50	29.97	49.90

Let’s focus on ammonia and a few other variables where removal efficiency makes sense.

```
levels(w1$parameter)

# NULL

w2 <- subset(w1, parameter %in% c('TSS', 'BOD5', 'COD', 'NO3.N', 'NH3.N', 'TP'))
```

The hydraulic retention time in these wetlands is only 1 d, so it is not unreasonable to assume influent and effluent samples collected on the same day are related.

```
w2$reff_cp <- 100 * (1 - w2$eff_cp / w2$influent)
w2$reff_pa <- 100 * (1 - w2$eff_pa / w2$influent)
```

```
head(w2)
```

#	date	parameter	unit	influent	eff_cp	eff_pa	reff_cp
# 5	04-15	TSS	mg/L	55.000	58.000	22.000	-5.454545
# 6	04-15	BOD5	mg/L	102.500	29.970	49.900	70.760976
# 7	04-15	COD	mg/L	205.040	89.890	78.040	56.159774
# 8	04-15	NO3.N	mg/L	0.605	2.105	7.615	-247.933884
# 9	04-15	NH3.N	mg/L	22.300	3.300	2.400	85.201794
# 10	04-15	TP	mg/L	5.010	3.070	4.140	38.722555
#		reff_pa					
# 5		60.00000					
# 6		51.31707					
# 7		61.93913					
# 8		-1158.67769					
# 9		89.23767					
# 10		17.36527					

And we can calculate mean values and standard deviation.

```
w3 <- melt(w2, measure.vars = c('reff_cp', 'reff_pa'),
           value.name = 'reff', variable.name = 'wetland')

wlsumm <- summarise(group_by(w3, parameter, wetland), reff_mean = mean(reff),
                   reff_sd = sd(reff))

# 'summarise()' has grouped output by 'parameter'. You can override using the '.groups'
argument.
```

```
kable(wlsumm, digits = 1)
```

parameter	wetland	reff_mean	reff_sd
BOD5	reff_cp	80.1	12.2
BOD5	reff_pa	74.3	17.9
COD	reff_cp	68.4	12.2
COD	reff_pa	63.5	11.6
NH3.N	reff_cp	72.2	16.3
NH3.N	reff_pa	71.6	14.9
NO3.N	reff_cp	NA	NA
NO3.N	reff_pa	NA	NA
TP	reff_cp	48.3	17.2
TP	reff_pa	41.5	18.3
TSS	reff_cp	12.6	66.8
TSS	reff_pa	52.8	36.8

Nitrate values are missing. We can exclude missing values but then we should also add the number of measurements.

```
wlsumm <- summarise(group_by(w3, parameter, wetland), nn = sum(!is.na(reff)),
  reff_mean = mean(reff, na.rm = TRUE),
  reff_sd = sd(reff, na.rm = TRUE))

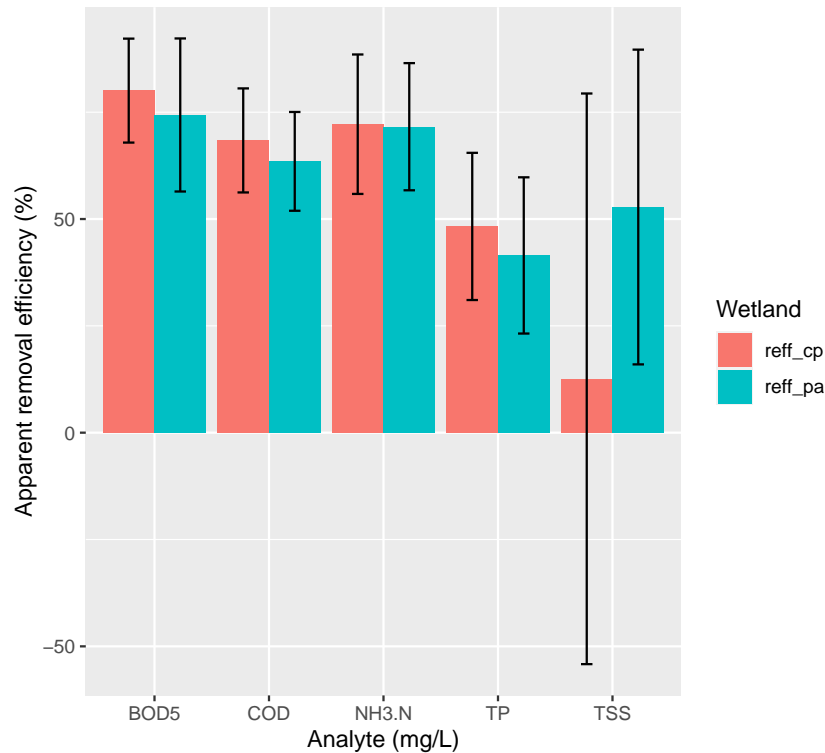
# 'summarise()' has grouped output by 'parameter'. You can override using the '.groups'
argument.
```

```
kable(wlsumm, digits = 1)
```

parameter	wetland	nn	reff_mean	reff_sd
BOD5	reff_cp	7	80.1	12.2
BOD5	reff_pa	7	74.3	17.9
COD	reff_cp	7	68.4	12.2
COD	reff_pa	7	63.5	11.6
NH3.N	reff_cp	7	72.2	16.3
NH3.N	reff_pa	7	71.6	14.9
NO3.N	reff_cp	6	-1054.7	961.0
NO3.N	reff_pa	6	-1275.4	891.8
TP	reff_cp	7	48.3	17.2
TP	reff_pa	7	41.5	18.3
TSS	reff_cp	7	12.6	66.8
TSS	reff_pa	7	52.8	36.8

Not surprisingly, nitrate removal was negative—presumably it is produced by nitrification. Let's omit it for a plot.

```
ggplot(subset(wlsumm, parameter != 'NO3.N'), aes(parameter, reff_mean, fill = wetland)) +
  geom_bar(position = position_dodge(), stat = 'identity') +
  geom_errorbar(aes(ymin = reff_mean - reff_sd, ymax = reff_mean + reff_sd),
    position = position_dodge(0.9), width = 0.2) +
  labs(x = 'Analyte (mg/L)', y = 'Apparent removal efficiency (%)',
    fill = 'Wetland')
```



So there are some “statistics”—average removal efficiency and some estimate of variability over time for these **two** wetlands. There is no evidence here of meaningful differences among the two plant species. If we were interested in this topic, we had better start designing more wetlands!

12 Example 2: Comparing two BMP methods

I did some work on the evaluation of a new method for measurement of biochemical methane potential (BMP) a couple years ago [Justesen et al., 2019]. We were interested in determining if our new method gave different results from other methods. Let’s get the data.

```
bb <- read.csv('data/BMP_comp.csv')
```

This is a small dataset, so we can look at the whole thing.

```
bb
```

```
#   method time.d id substrate    bmp
# 1   grav   31.1 L1 Cellulose 363.1848
# 2   grav   31.1 L3 Cellulose 360.1335
# 3   grav   31.1 E1  Ethanol 725.7272
# 4   grav   31.1 E2  Ethanol 712.9015
# 5   grav   31.1 E3  Ethanol 762.2732
# 6   grav   31.1 A1           A 376.4508
# 7   grav   31.1 A2           A 382.8212
# 8   grav   31.1 A3           A 376.9564
```

```
# 9    grav  31.1 B1      B 375.0472
# 10   grav  31.1 B2      B 381.1704
# 11   grav  31.1 B3      B 386.2744
# 12   grav  31.1 C1      C 524.2979
# 13   grav  31.1 C2      C 521.2255
# 14   grav  31.1 C3      C 518.9827
# 15   gdt   31.1 L1 Cellulose 371.8001
# 16   gdt   31.1 L3 Cellulose 400.9151
# 17   gdt   31.1 E1 Ethanol 620.4517
# 18   gdt   31.1 E2 Ethanol 708.3848
# 19   gdt   31.1 E3 Ethanol 662.8819
# 20   gdt   31.1 A1      A 378.2669
# 21   gdt   31.1 A2      A 384.3896
# 22   gdt   31.1 A3      A 361.8799
# 23   gdt   31.1 B1      B 400.7561
# 24   gdt   31.1 B2      B 380.8994
# 25   gdt   31.1 B3      B 310.7199
# 26   gdt   31.1 C1      C 434.8386
# 27   gdt   31.1 C2      C 520.3938
# 28   gdt   31.1 C3      C 533.3696
```

Continuing with the questions posed in Section 10, the sample should be appropriate for the question. The response variable is BMP measured for a single bottle using a particular method (gravimetric, **grav** or the new one, GD-BMP, **gdt**). The experimental unit is a single bottle. It looks like we have replication—three bottles for all substrates except cellulose, for which we have two. Apart from receiving the same substrates, the bottles were independent. These measurements were carried out by a single group of researchers at a single laboratory, so undoubtedly there are important systematic errors. But we will assume they apply equally to all bottles. The experimental factor—measurement method—is categorical by nature. Importantly, it seems that both of the methods were applied to each bottle (the **id** column has a unique bottle identifier).

Even though we can easily look at the entire data frame, let's get in the habit of looking at a summary of any new data set.

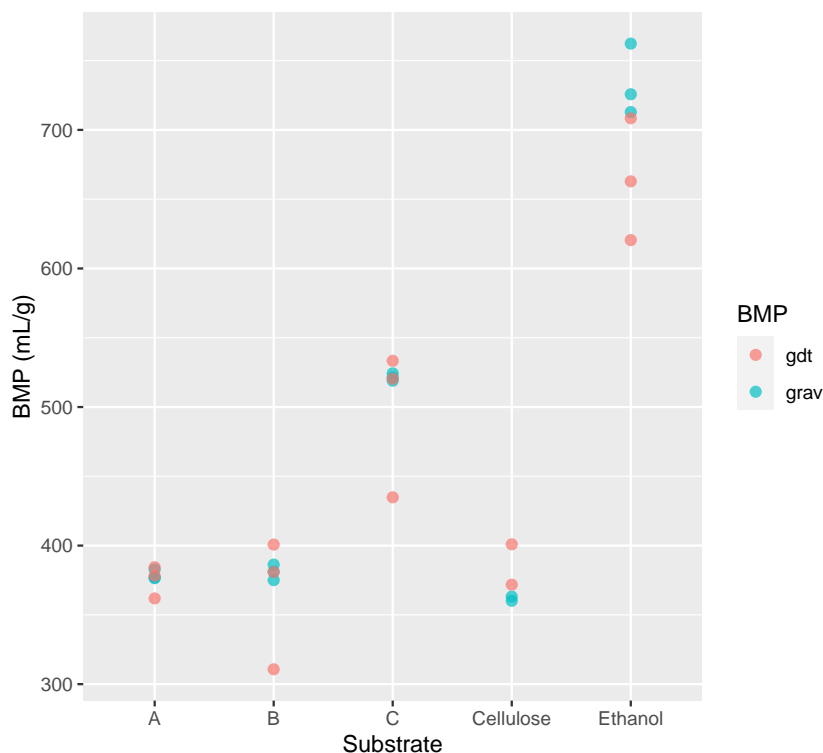
```
dfsumm(bb)
```

```
#
# 28 rows and 5 columns
# 28 unique rows
#
#      method  time.d      id substrate      bmp
# Class      character numeric character character numeric
# Minimum      gdt    31.1      A1      A      311
# Maximum      grav    31.1      L3  Ethanol    762
# Mean          gdt    31.1      C1      C      473
# Unique (excl. NA)  2      1      14      5      28
# Missing values      0      0      0      0      0
# Sorted          FALSE    TRUE    FALSE    FALSE    FALSE
```

No values missing, we have two levels of **method** as expected, and BMP varies from 300 to above 700. Everything seems OK.

Let's plot these data.

```
ggplot(bb, aes(substrate, bmp, colour = method)) +
  geom_point(alpha = 0.7, size = 2) +
  labs(x = 'Substrate', y = 'BMP (mL/g)', colour = 'BMP')
```



So what do we see? Is there evidence of differences between the methods? Can we compare them? There is in fact virtually no evidence that the methods differ; results overlap for all substrates except ethanol. Should we conclude then that the two methods give identical results? Hell no! In fact, we should assume from the start that they do not. These two methods are based on different principles and each almost certainly have their own biases. So perhaps our question should be changed to “How different are the two methods?” The plot seems to suggest that the answer is “not very different, at least in comparison to measurement error”. We might take this as our conclusion and stop here. But we could try to make a quantitative estimate of how similar the two methods are likely to be.

Something else we can see in this plot is that the variability of the GD method is higher than the gravimetric method. Is this meaningful? Well, here we see that for five different substrates, GD results were much more variable for all of them. That is pretty strong evidence that the difference is real. Later, perhaps, we can apply a statistical test. But it is worth pointing out that this difference could be an impediment to applying statistical models for comparing BMP, because we would typically have to assume that variance is constant.

Thinking more about the difference between the methods, we should consider the paired nature of the measurements. Let’s calculate a difference between the two methods for each individual bottle. That will remove some of the random error associated with bottles, and, conveniently, eliminates the problem with unequal variance. See how experimental design can help you? In general, a paired approach is more powerful, that is, more likely to show a clear difference when in fact one exists.

Box 14. Paired observations

Paired observations result from making two measurements on the same experimental unit (subject), typically under two different conditions or after application of two different treatments. In general the approach is more *powerful* than if these measurements were made on different units of observation, because some of the random error is eliminated. With more than two measurements, this approach can be called “repeated measures”.

```
library(tidyr)
```

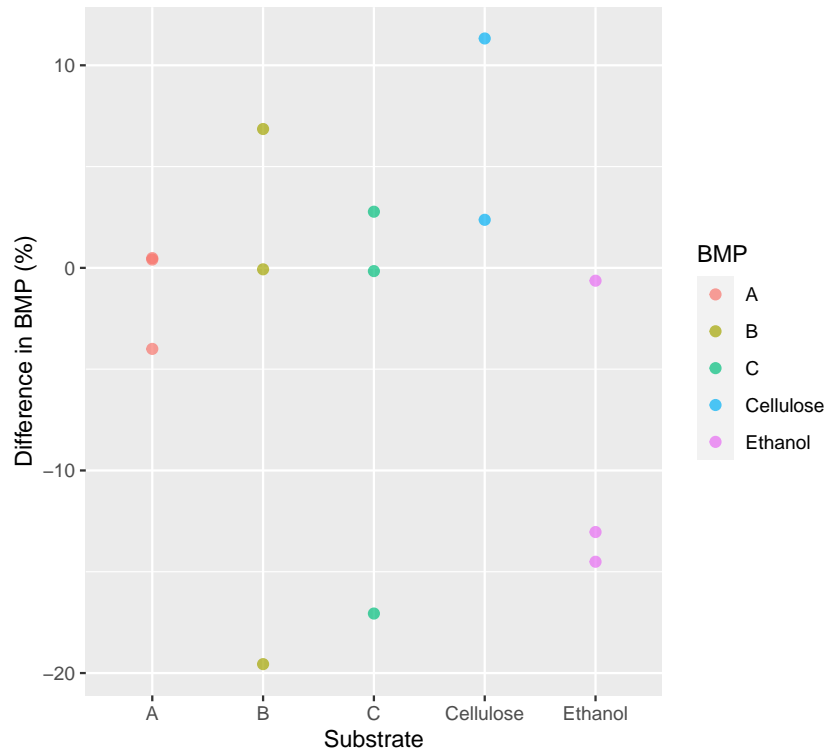
```
bw <- spread(bb, method, bmp)
head(bw)
```

```
#   time.d id substrate      gdt      grav
# 1   31.1 A1          A 378.2669 376.4508
# 2   31.1 A2          A 384.3896 382.8212
# 3   31.1 A3          A 361.8799 376.9564
# 4   31.1 B1          B 400.7561 375.0472
# 5   31.1 B2          B 380.8994 381.1704
# 6   31.1 B3          B 310.7199 386.2744
```

We can now easily calculate a difference for each bottle, and I'll add a relative difference (% of gravimetric result) as well.

```
bw$diff <- bw$gdt - bw$grav
bw$rdiff <- 100 * bw$diff / bw$grav
```

```
ggplot(bw, aes(substrate, rdiff, colour = substrate)) +
  geom_point(alpha = 0.7, size = 2) +
  labs(x = 'Substrate', y = 'Difference in BMP (%)', colour = 'BMP')
```



Here as well, there is no evidence of a consistent difference between the methods, possibly excluding ethanol. We can still conclude then, without fitting a statistical model, that there is no real evidence of a systematic difference between the methods.

But, maybe we would like to say how large a difference *might* exist. We could use these data for this, and can (finally) apply a statistical model! Let's use the relative difference as the response variable, because it likely to be less variable than the absolute differences. Still, we should include the fact that bottles with the same substrate are not independent. We'll fit a classical linear model using the `lm()` function in R. Because `substrate` is a factor (categorical variable) and not continuous, R will automatically create dummy variables for us. Essentially we are carrying out analysis of variance (ANOVA) here.

```
mod1 <- lm(rdiff ~ substrate - 1, data = bw)
summary(mod1)
```

```
#
# Call:
# lm(formula = rdiff ~ substrate - 1, data = bw)
#
# Residuals:
```

	Min	1Q	Median	3Q	Max
	-15.301	-4.268	1.482	4.612	11.114

```
#
# Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
substrateA	-1.036	5.354	-0.193	0.851
substrateB	-4.259	5.354	-0.795	0.447
substrateC	-4.817	5.354	-0.900	0.392
substrateCellulose	6.848	6.557	1.044	0.324

	A	B	C	D	E	F	G	H	I	J	K	L
1	Time (g)	Bottle ID	Substrate	Grav. BMP (mL/g)	GD BMP (mL/g)	Dummy cell	Dummy ethanol	Dummy A	Dummy B	Dummy C	Absolute difference (mL/g)	Relative difference (% of grav.)
2	31.1	L1	Cellulose	363.18	371.80	1	0	0	0	0	8.62	2.37
3	31.1	L3	Cellulose	360.13	400.92	1	0	0	0	0	40.78	11.32
4	31.1	E1	Ethanol	725.73	620.45	0	1	0	0	0	-105.28	-14.51
5	31.1	E2	Ethanol	712.90	708.38	0	1	0	0	0	-4.52	-0.63
6	31.1	E3	Ethanol	762.27	662.88	0	1	0	0	0	-99.39	-13.04
7	31.1	A1	A	376.45	378.27	0	0	1	0	0	1.82	0.48
8	31.1	A2	A	382.82	384.39	0	0	1	0	0	1.57	0.41
9	31.1	A3	A	376.96	361.88	0	0	1	0	0	-15.08	-4.00
10	31.1	B1	B	375.05	400.76	0	0	0	1	0	25.71	6.85
11	31.1	B2	B	381.17	380.90	0	0	0	1	0	-0.27	-0.07
12	31.1	B3	B	386.27	310.72	0	0	0	1	0	-75.55	-19.56
13	31.1	C1	C	524.30	434.84	0	0	0	0	1	-89.46	-17.06
14	31.1	C2	C	521.23	520.39	0	0	0	0	1	-0.83	-0.16
15	31.1	C3	C	518.98	533.37	0	0	0	0	1	14.39	2.77
16												

Figure 8: BMP comparison data ready for analysis in LibreOffice Calc (a spreadsheet program).

```
# substrateEthanol      -9.393      5.354  -1.755    0.113
#
# Residual standard error: 9.273 on 9 degrees of freedom
# Multiple R-squared:  0.3856, Adjusted R-squared:  0.0443
# F-statistic:  1.13 on 5 and 9 DF,  p-value: 0.4104
```

Not surprisingly, there is no evidence of a difference. But how large could a difference be? We can get confidence intervals to tell us this.

```
confint(mod1)

#               2.5 %    97.5 %
# substrateA      -13.146300 11.074698
# substrateB      -16.369173  7.851825
# substrateC      -16.927203  7.293795
# substrateCellulose -7.984173 21.680371
# substrateEthanol -21.503363  2.717635
```

We can say then, that at the 95% confidence, any difference between the methods is probably smaller than 17%, with the exception of ethanol. That is useful! If we want more information or a better estimate of any probable difference between the methods, we would need to carry out additional experiments. For this new method, this is precisely what was later done, with comparisons made within and even among other laboratories, with measurements made by different technicians [Justesen et al., 2019].

Could this analysis have been done using a spreadsheet? Yes, but with some difficulty, and an unfortunate lack of clarity and reproducibility. In LibreOffice Calc, it is necessary to first calculate the response variable (relative difference), as in the R code above, but then to add columns with “dummy variables” for substrate. This is shown in Fig. 8, and also in the file “BMP_comp.ods” in the spreadsheets directory.

To actually fit the regression model, the “Data” menu is selected, then “Statistics”, and finally, “Regression”. Variables and options are selected are shown in Fig. 9.

Figure 9: Inputs required for analysis of the BMP data in LibreOffice Calc.

13 Example 3: VOC emission from silage

The data in `ethanol_emis.xlsx` are on ethanol emission from maize silage (fermented cattle feed) measured in a simple wind tunnel. Emission was measured from 15 cm thick samples of silage taken from bunker silos, where silage is stored for weeks or months. The measurements were part of a crossed factorial experiment designed for evaluating the effect of temperature and wind speed at the silage surface on emission rate. The response variable is in the last column: `emis.n` (for emission, normalized), and is the fraction of initial ethanol mass lost over 12 hours of emission. Temperature and relative humidity were controlled using an environmental chamber. Air speed was controlled using a blower and a system of valves. The target value is given in `speed.tar` while the actual value is in `speed`. Silage density was not controlled, but was determined because it affects porosity, which could affect emission rate. Silage gas-phase porosity was determined from density and dry matter content.

The primary question we were interested in was how do temperature and air speed affect ethanol emission?

```
library(tidyverse)
library(readxl)
```

Read in the data.

```
et <- read_excel("data/ethanol_emis.xlsx", skip = 1)
```

Check data

```
dfsumm(et)

#
# 27 rows and 15 columns
# 27 unique rows
```

```

#           id      sample      box thick.samp  temp.c
# Class      numeric      character character      numeric numeric
# Minimum      124 C 2009JULY16A      A      0.15      5
# Maximum      170 C2009JUNE30D      D      0.15      35
# Mean         141 C2009JULY13C      B      0.15      19.4
# Unique (excl. NA) 27      27      4      1      3
# Missing values 0      0      0      0      0
# Sorted      TRUE      FALSE      FALSE      TRUE      FALSE
#           rh.tar speed.tar headspace  speed      dm
# Class      numeric      numeric      numeric numeric numeric
# Minimum      50      0.05      0.01  0.042      28.9
# Maximum      95      5      0.1  5.07      35.8
# Mean         70.4      1.4  0.0353  1.43      33.5
# Unique (excl. NA) 3      3      3      27      27
# Missing values 0      0      0      0      0
# Sorted      FALSE      FALSE      FALSE      FALSE      FALSE
#           c.etch.i  rho.d  por.g  emis.t  emis.n
# Class      numeric numeric numeric numeric numeric
# Minimum      1540      184  0.213  0.216  0.016
# Maximum      3930      306  0.54  6.53  0.501
# Mean         2990      266  0.304  1.52  0.141
# Unique (excl. NA) 27      27      27      27      27
# Missing values 0      0      0      0      0
# Sorted      FALSE      FALSE      FALSE      FALSE      FALSE

```

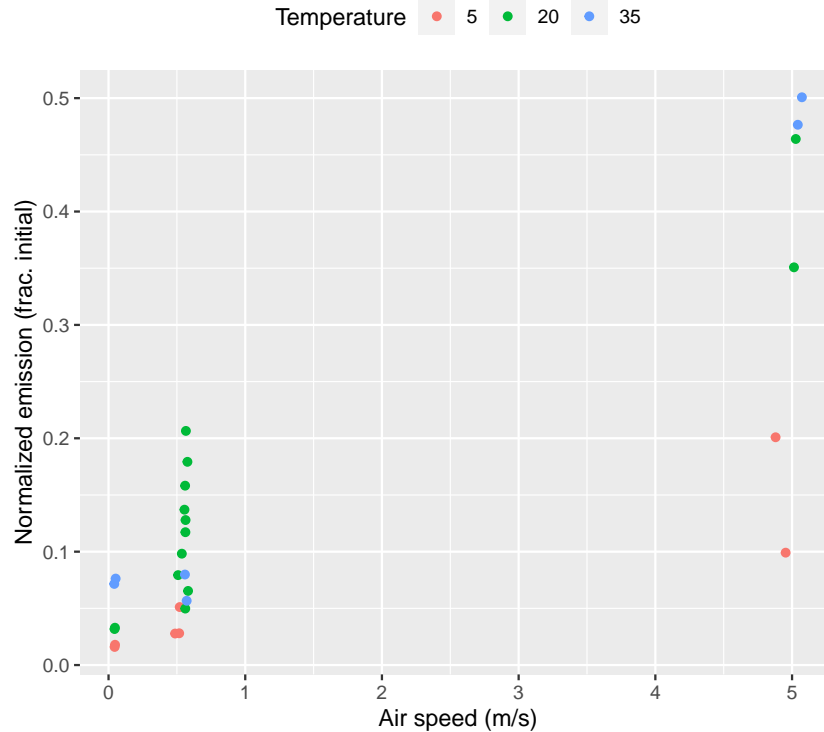
Everything looks OK. No missing values. All the variables we are interested in are numeric, except `box`, which is character. We will need to convert it to a factor later. Notice the ranges—`emis.n` is all between 0.016 and 0.501.

A plot is the best place to start.

```

ggplot(et, aes(speed, emis.n, colour = factor(temp.c))) +
  geom_point() +
  labs(x = 'Air speed (m/s)', y = 'Normalized emission (frac. initial)',
       colour = 'Temperature') +
  theme(legend.position = 'top')

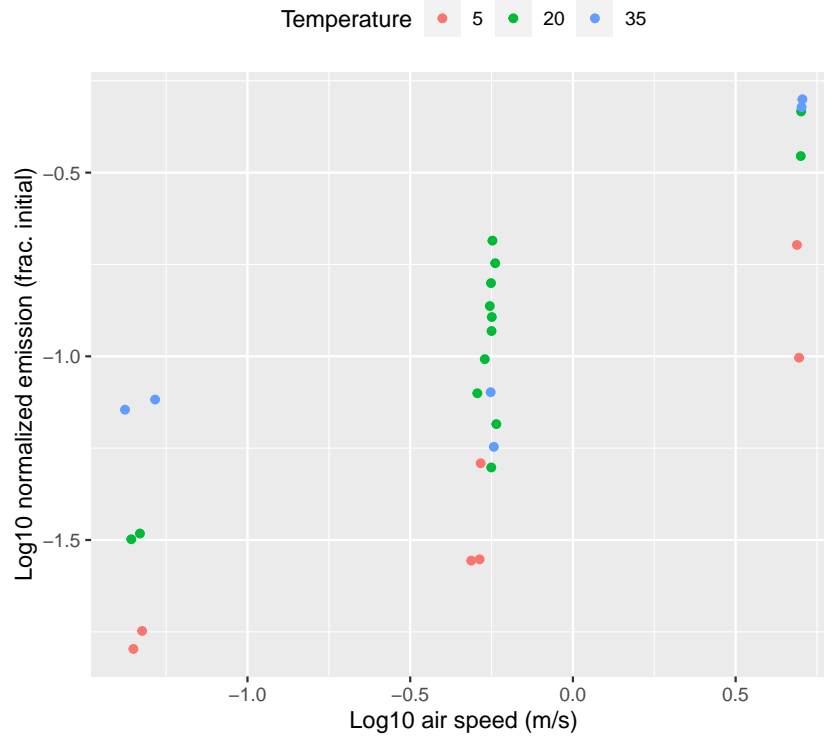
```



Clearly air speed, and probably temperature, affected emission of ethanol. But let's pause and think about a few other things here.

First, what kind of relationships should we expect? Thinking about this will help us display the data in the best way, and later, fit the most appropriate model. You can assume that ethanol emission is related to its volatility, which we can quantify using Henry's law constant. How does volatility respond to temperature? It sure isn't linear. In fact, a common assumption is that Henry's law constant changes by a factor of 2 with every 10°C change in temperature. The form of this statement suggests a logarithmic relationship, so we should log transform emission. How about air speed? We might expect that emission rate is at least partially controlled by the mass transfer coefficient from the surface. If we think about correlations for forced convection, the Reynolds number raised to the power of 0.5, i.e., $h \propto Re^{0.5}$. A log-log model can be used for this relationship, and we can get that by a log transformation of air speed in addition to emission.

```
ggplot(et, aes(log10(speed), log10(emis.n), colour = factor(temp.c))) +
  geom_point() +
  labs(x = 'Log10 air speed (m/s)', y = 'Log10 normalized emission (frac. initial)',
       colour = 'Temperature') +
  theme(legend.position = 'top')
```



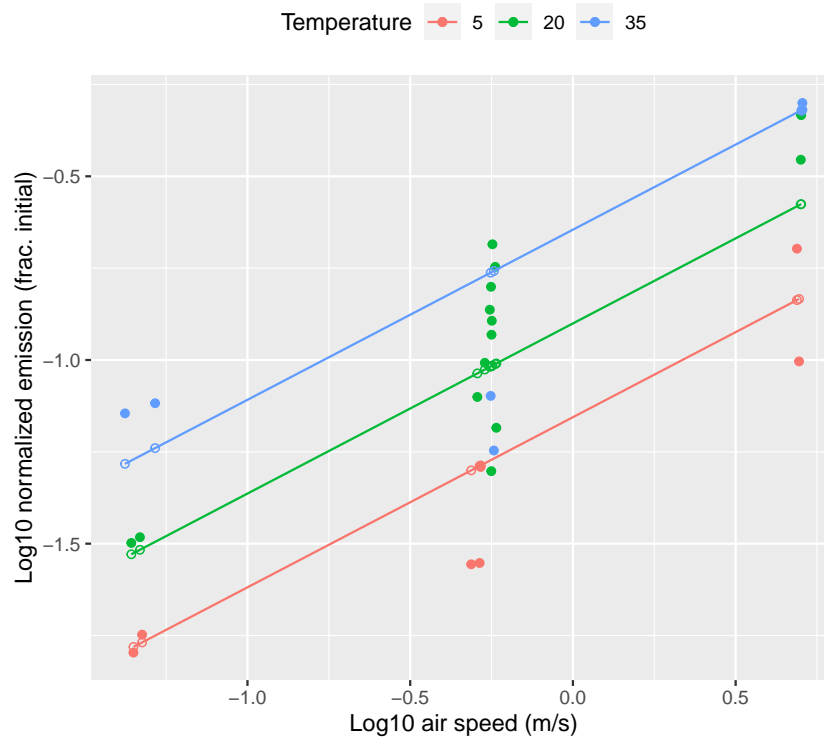
```
mod1 <- lm(log10(emis.n) ~ temp.c + log10(speed), data = et)

summary(mod1)

#
# Call:
# lm(formula = log10(emis.n) ~ temp.c + log10(speed), data = et)
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -0.48859 -0.11700  0.02118  0.13010  0.32998
#
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -1.240841   0.086574 -14.333 2.91e-13 ***
# temp.c         0.017015   0.003844   4.426 0.000179 ***
# log10(speed)   0.463134   0.058796   7.877 4.15e-08 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.2076 on 24 degrees of freedom
# Multiple R-squared:  0.7745, Adjusted R-squared:  0.7557
# F-statistic: 41.2 on 2 and 24 DF, p-value: 1.733e-08

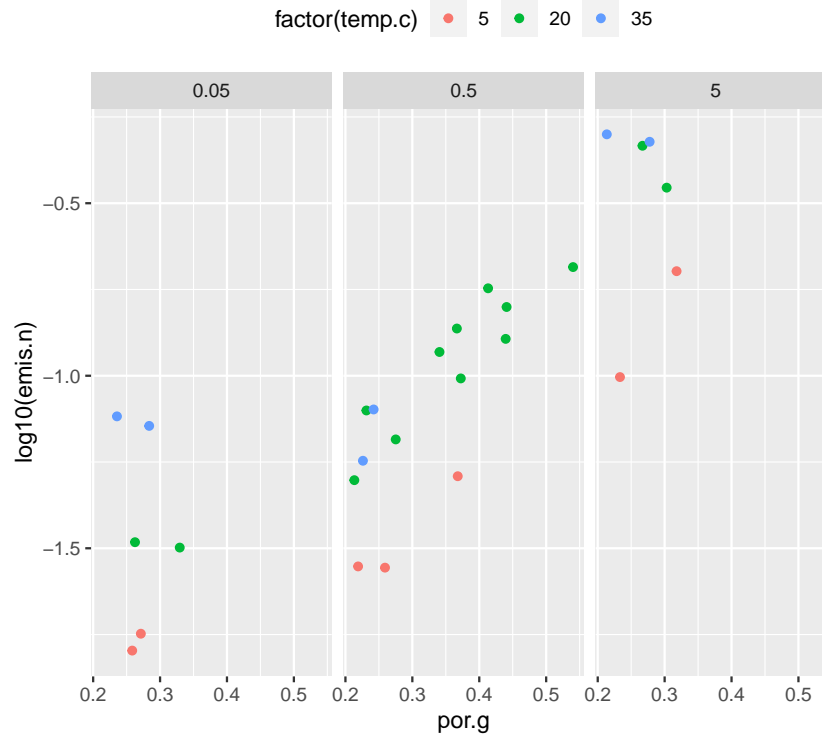
et$pred1 <- 10^predict(mod1)
et$resid1 <- resid(mod1)
```

```
ggplot(et, aes(log10(speed), log10(emis.n), colour = factor(temp.c))) +
  geom_point() +
  geom_point(aes(y = log10(pred1)), pch = 1) +
  geom_line(aes(y = log10(pred1))) +
  labs(x = 'Log10 air speed (m/s)', y = 'Log10 normalized emission (frac. initial)',
       colour = 'Temperature') +
  theme(legend.position = 'top')
```



So we could end here. But how about porosity? It was not controlled, but we might expect that it could affect volatilization, because volatile compounds could travel through gas pores. Wouldn't it be great if it explained much of the variability not clearly related to the other experimental factors? Let's take a look.

```
ggplot(et, aes(por.g, log10(emis.n), colour = factor(temp.c))) +
  geom_point() +
  facet_wrap(~ speed.tar) +
  theme(legend.position = 'top')
```

There is very clear correlation and we should include it!

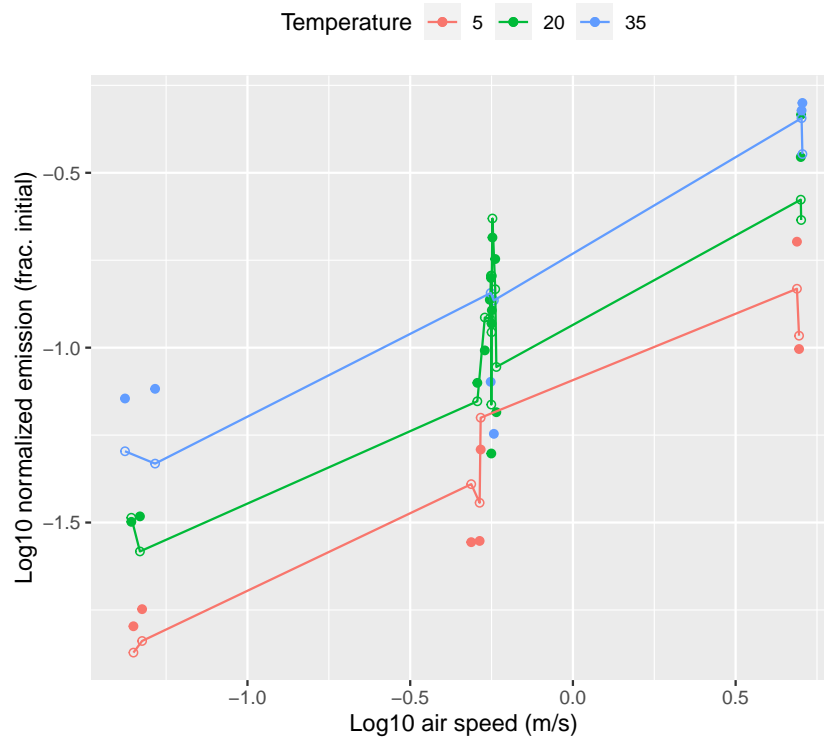
```
mod2 <- lm(log10(emis.n) ~ temp.c + log10(speed) + por.g, data = et)

summary(mod2)

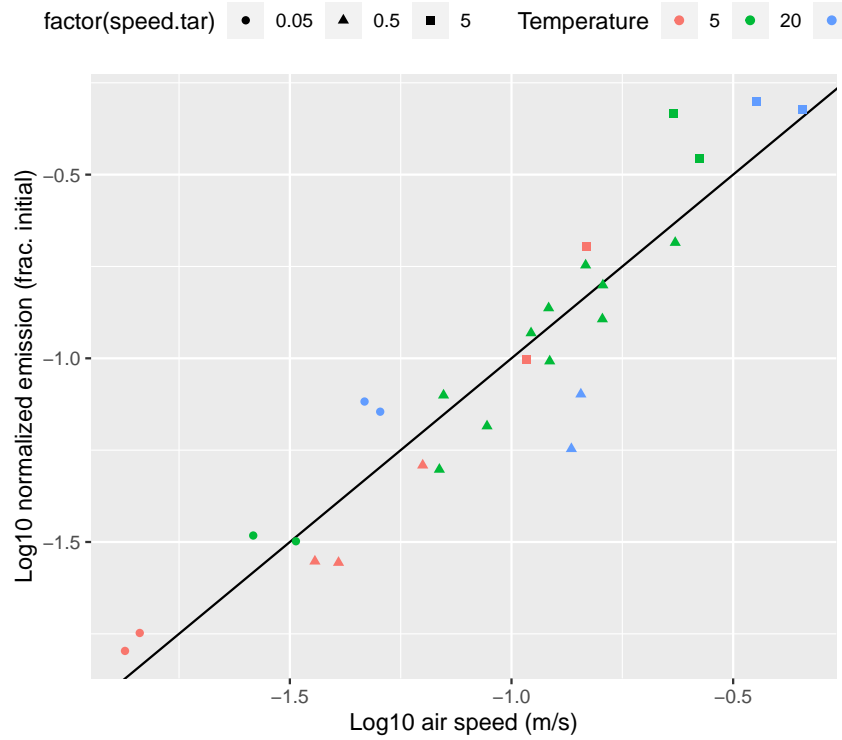
#
# Call:
# lm(formula = log10(emis.n) ~ temp.c + log10(speed) + por.g, data = et)
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -0.38144 -0.09606  0.02184  0.09561  0.30146
#
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept) -1.757115   0.135913  -12.928 4.93e-12 ***
# temp.c       0.018219   0.002926   6.227 2.36e-06 ***
# log10(speed) 0.462940   0.044546  10.392 3.68e-10 ***
# por.g        1.622569   0.374118   4.337 0.000243 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.1573 on 23 degrees of freedom
# Multiple R-squared:  0.8759, Adjusted R-squared:  0.8597
# F-statistic: 54.12 on 3 and 23 DF, p-value: 1.405e-10

et$pred2 <- 10^predict(mod2)
et$resid2 <- resid(mod2)
```

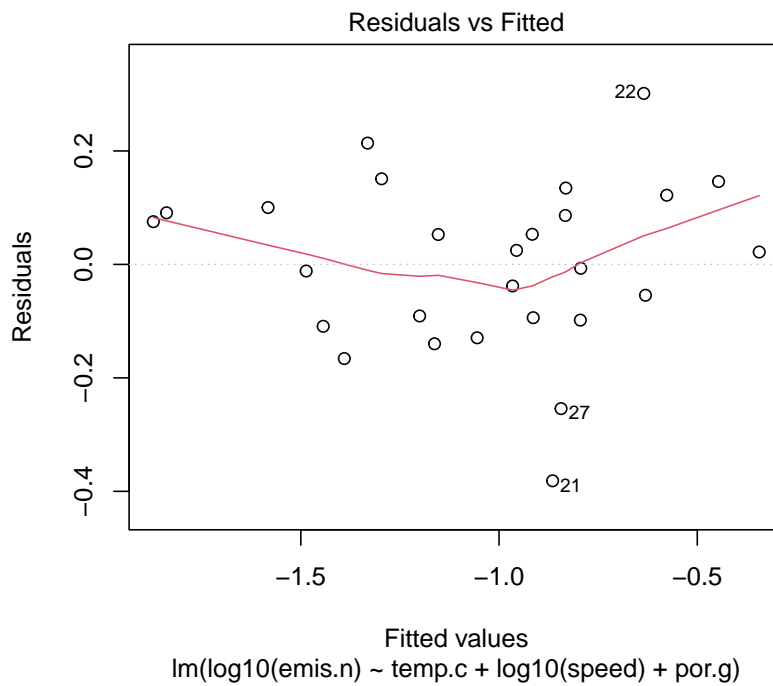
```
ggplot(et, aes(log10(speed), log10(emis.n), colour = factor(temp.c))) +
  geom_point() +
  geom_point(aes(y = log10(pred2)), pch = 1) +
  geom_line(aes(y = log10(pred2))) +
  labs(x = 'Log10 air speed (m/s)', y = 'Log10 normalized emission (frac. initial)',
       colour = 'Temperature') +
  theme(legend.position = 'top')
```

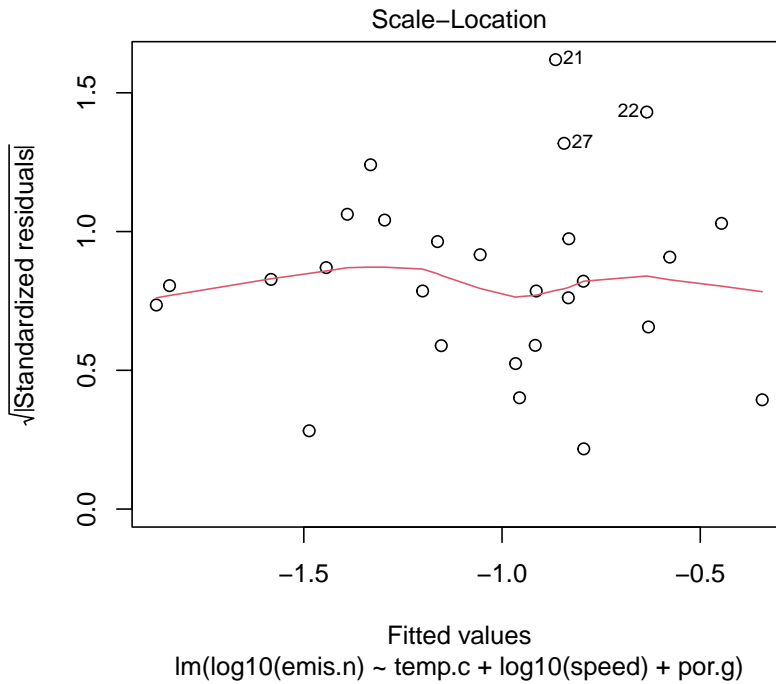
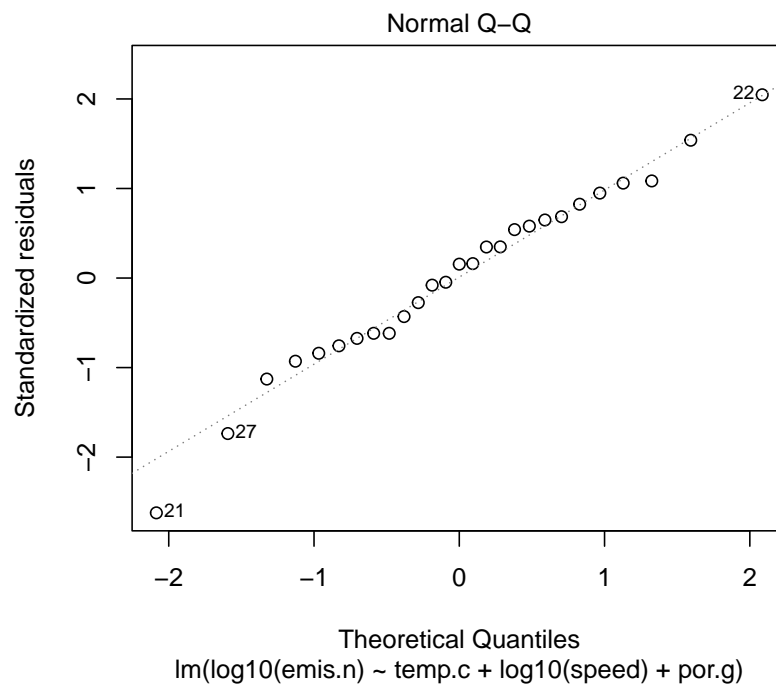


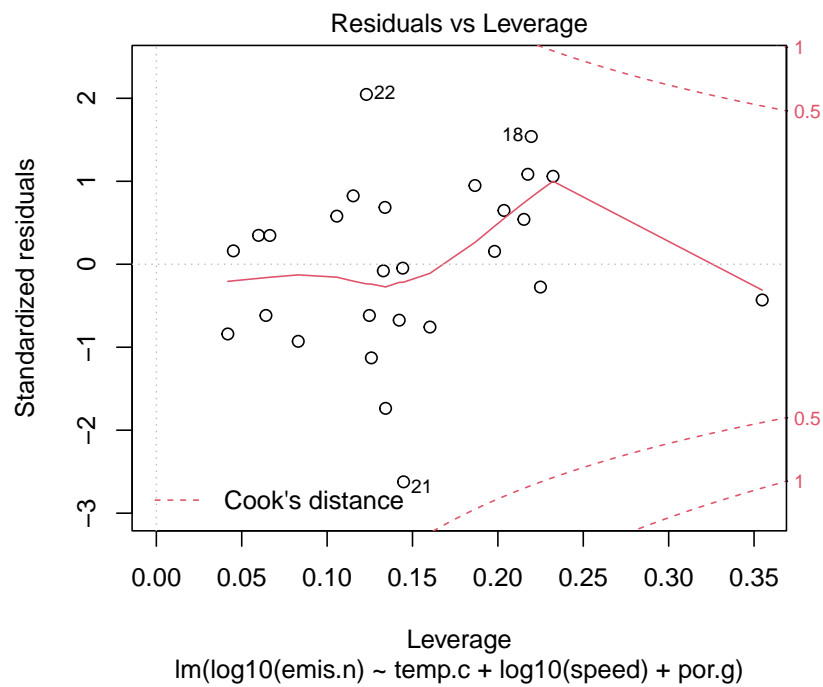
```
ggplot(et, aes(log10(pred2), log10(emis.n), colour = factor(temp.c),
               shape = factor(speed.tar))) +
  geom_abline(intercept = 0, slope = 1) +
  geom_point() +
  labs(x = 'Log10 air speed (m/s)', y = 'Log10 normalized emission (frac. initial)',
       colour = 'Temperature') +
  theme(legend.position = 'top')
```



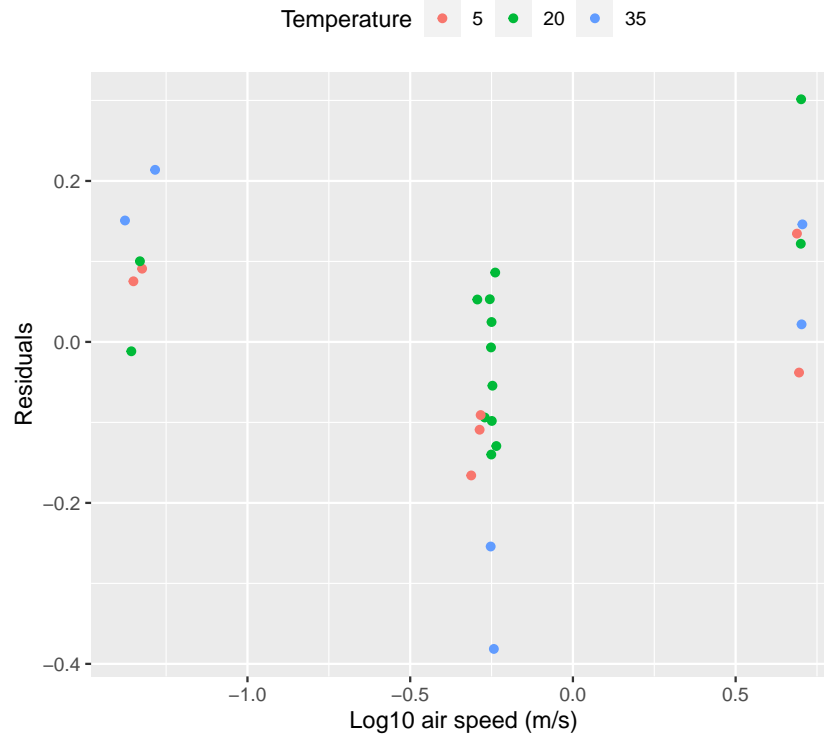
```
plot(mod2, ask = FALSE)
```







```
ggplot(et, aes(log10(speed), resid2, colour = factor(temp.c))) +
  geom_point() +
  labs(x = 'Log10 air speed (m/s)', y = 'Residuals',
       colour = 'Temperature') +
  theme(legend.position = 'top')
```



That's an informative plot.

14 Problem 1. Inoculum effects on BMP

Koch et al. [2017] studied the effect of inoculum origin on biochemical methane potential (BMP) for four substrates. Data are given in the file `BMP_inoc.csv`, where the unit of observation is a single BMP bottle. Take a look at the data and answer these questions:

1. Did BMP depend on inoculum type?
2. Did any effect vary by substrate?

The original data are in a intermediate structure, with replicates across columns.

```
bi <- read.csv('data/BMP_inoc.csv')
```

```
bi
```

#	substrate	inoc	BMP1	BMP2	BMP3	BMP4	BMP5	BMP6	BMP7	BMP8
# 1	Sewage Sludge	WWTP	293.8	272.8	303.9	260.2	275.7	276.6	309.9	330.1
# 2	Maize	WWTP	319.7	320.2	344.5	324.7	328.3	338.6	324.8	351.9
# 3	Food Waste	WWTP	453.9	444.5	462.9	451.1	453.9	473.7	423.8	419.5
# 4	Cellulose	WWTP	333.3	315.6	341.0	322.8	330.4	338.9	338.9	343.0
# 5	Sewage Sludge	ABP	294.8	294.2	293.9	267.0	269.6	272.5	332.4	319.8
# 6	Maize	ABP	320.1	325.6	348.6	362.5	343.8	412.5	326.6	330.9

```
# 7      Food Waste  ABP 441.1 432.2 466.2 490.0 398.3 429.3 423.3 432.5
# 8      Cellulose  ABP 344.1 347.7 374.8 348.5 351.3 378.0 354.9 367.5
# 9  Sewage Sludge BWTP 296.6 307.6 307.5 309.1 315.0 319.4 342.3 325.0
# 10      Maize BWTP 328.2 341.6 356.8 339.4 357.3 372.6 336.6 339.5
# 11      Food Waste BWTP 459.0 450.8 484.4 453.2 449.3 483.8 442.9 429.7
# 12      Cellulose BWTP 379.0 389.4 376.8 360.1 357.0 389.0 362.5 369.7
#      BMP9
# 1 328.3
# 2 352.1
# 3 432.0
# 4 350.0
# 5 319.4
# 6 335.5
# 7 439.8
# 8 366.9
# 9 347.1
# 10 356.0
# 11 458.2
# 12 376.7
```

This structure could work well in a spreadsheet analysis. For analysis in R, the structure can be changed to long using the `gather()` function.

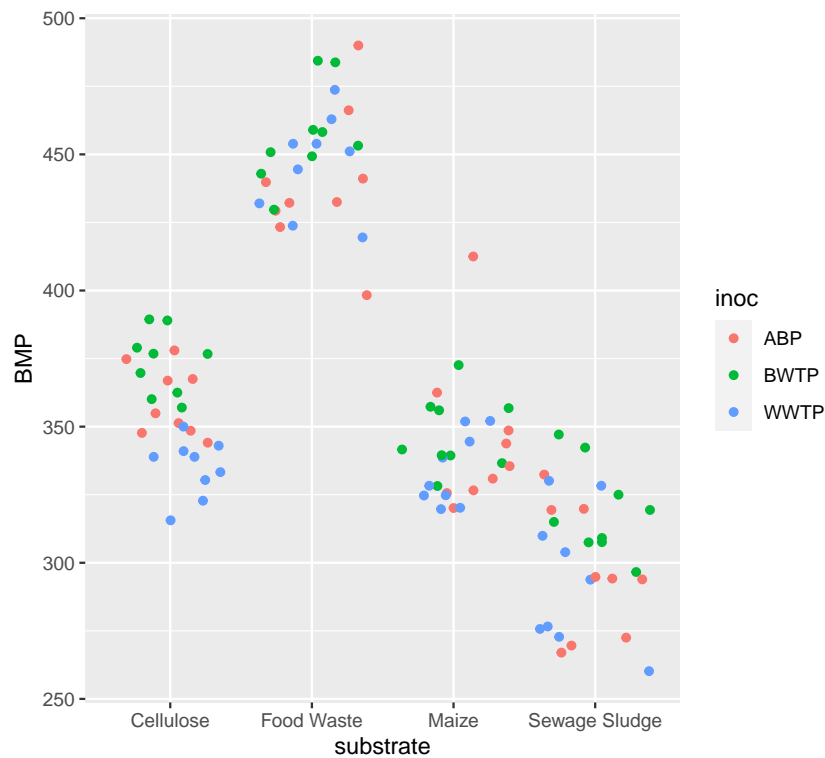
```
library(tidyr)
```

```
bil <- gather(bi, key = 'rep', value = 'BMP', contains('BMP'))
head(bil)
```

```
#      substrate inoc rep  BMP
# 1 Sewage Sludge WWTP BMP1 293.8
# 2      Maize WWTP BMP1 319.7
# 3  Food Waste WWTP BMP1 453.9
# 4  Cellulose WWTP BMP1 333.3
# 5 Sewage Sludge ABP BMP1 294.8
# 6      Maize ABP BMP1 320.1
```

Here are the values, with a single point representing a BMP value from a single bottle.

```
ggplot(bil, aes(substrate, BMP, colour = inoc)) +
  geom_jitter(height = 0)
```



15 Problem 2. Wood hardness and density

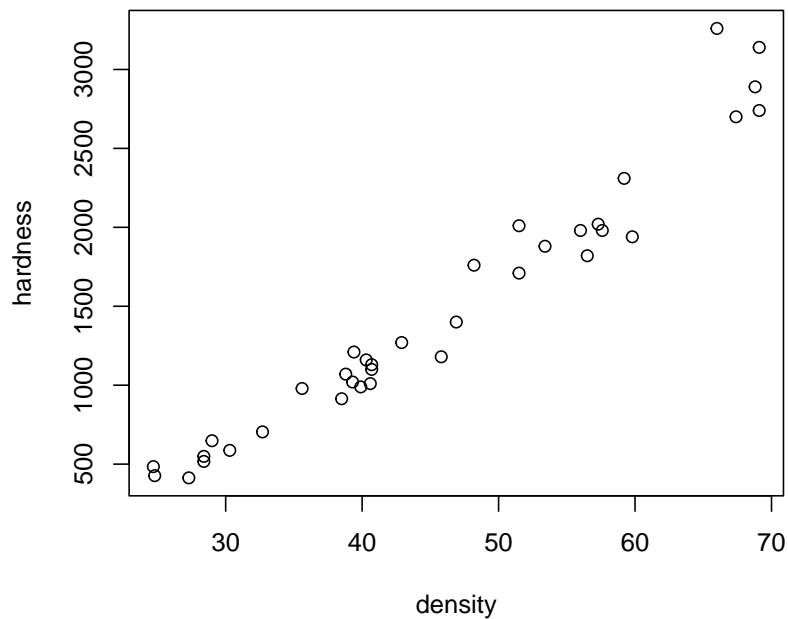
Faraway [2005] presented some data on the hardness of some Australian woods, and I've copied these into the file `janka.csv`.

```
hard <- read.csv("data/janka.csv")
dfsumm(hard)

#
# 36 rows and 2 columns
# 36 unique rows
#
#           density hardness
# Class          numeric  integer
# Minimum         24.7      413
# Maximum         69.1     3260
# Mean            45.7     1180
# Unique (excl. NA)  32       35
# Missing values     0        0
# Sorted            TRUE     FALSE
```

Let's start out by seeing what the data look like.

```
plot(hardness ~ density, data = hard)
```

We might be interested in doing two things with these data: determining if wood hardness (difficult to measure) is related to wood density (easy to measure), and, if so, predicting hardness from the density. Are these data experimental or observational? Try to fit an appropriate regression model to these data, and take a look at the residuals to check the structure. Can you improve it?

16 Problem 3. Fruit fly longevity and sexual activity

The data in the file `fruitfly.csv` are from an experiment on fruitfly longevity and are also from [Faraway \[2005\]](#). The original objective of this famous experiment was to assess the effect of sexual activity (manipulated by controlling the number of females placed with a single male, `activity` column) on fruitfly longevity (how long the flies live, `longevity` column). But longevity is known to be correlated with thorax length (`thorax` column).

```
ff <- read.csv('data/fruitfly.csv')
head(ff)
```

```
#   thorax longevity activity
# 1  0.68         37    many
# 2  0.68         49    many
# 3  0.72         46    many
# 4  0.72         63    many
# 5  0.76         39    many
# 6  0.76         46    many
```

1. How might you plot these data to assess the effect of activity?

2. How can you fit a statistical model that utilizes the correlation with thorax length to increase power?
3. What approach should you use to compare the levels of activity to each other?

17 Problem 4: Growth and nitrate accumulation by *Lemna minor*

Duckweeds are very tiny floating plants that can be used for wastewater treatment and recovery of nitrogen. Harvested material can be used as an animal feed. [Devlamynck et al. \[2020\]](#) measured biomass production and nitrate accumulation in a duckweed species *Lemna minor*. The data are in `lemna.csv`. Use them to explore the following questions.

1. Did medium affect growth (`grow`)?
2. Did medium affect NO_3^- accumulation (`N03.accum`)?
3. Is NO_3^- accumulation related to NO_3^- concentration in the medium (`N03.med`)?

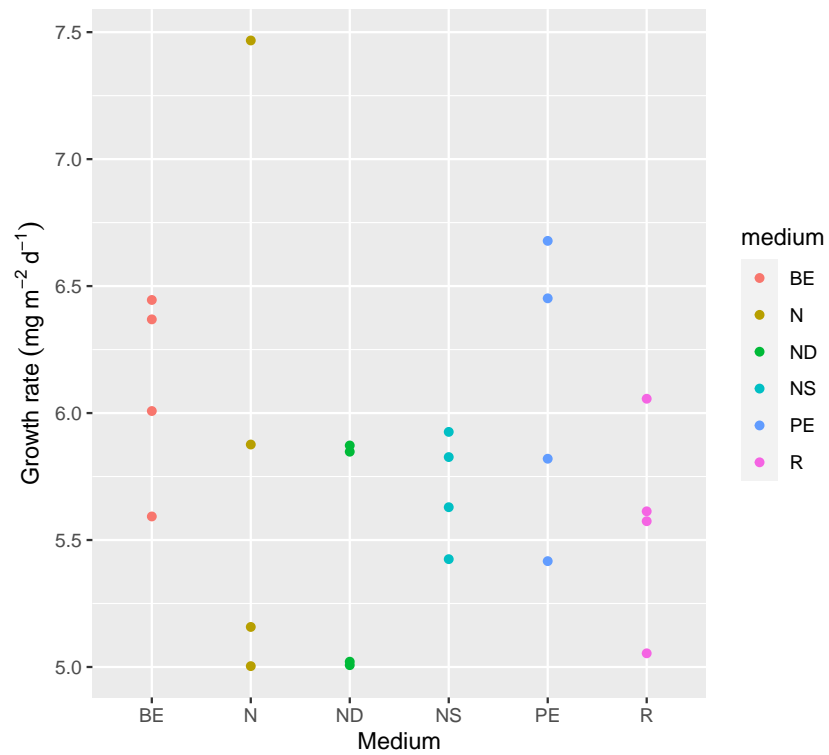
```
lem <- read.csv('data/lemna.csv')
```

```
summary(lem)
```

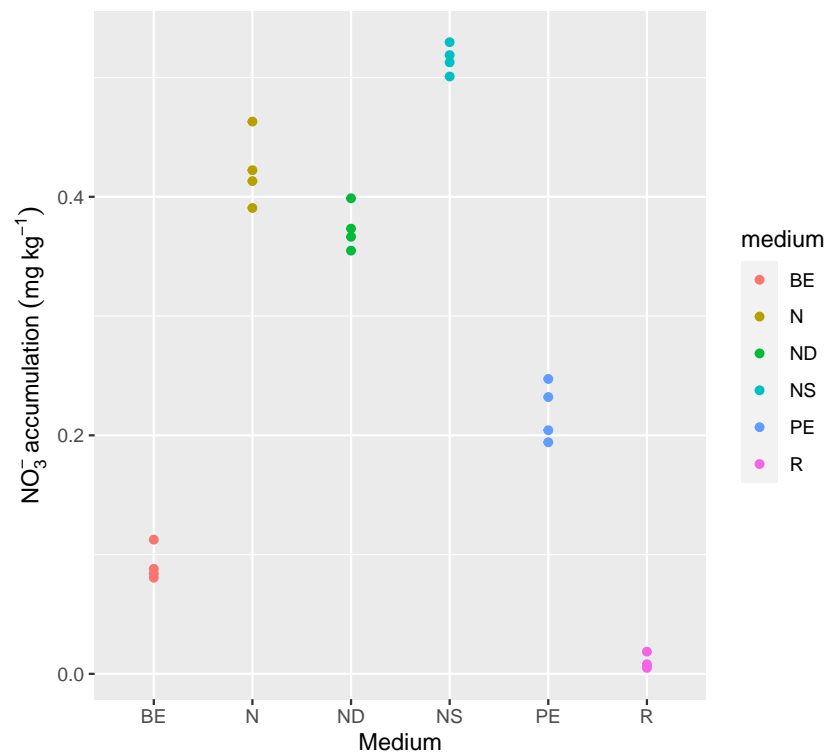
```
# med.descrip      medium      grow
# Length:24      Length:24      Min.   :5.003
# Class :character Class :character 1st Qu.:5.423
# Mode  :character Mode  :character Median :5.823
#                                     Mean  :5.797
#                                     3rd Qu.:6.020
#                                     Max.   :7.467
#   N03.accum      pH.med      N03.med
# Min.   :0.005025 Min.   :5.760 Min.   : 0.009594
# 1st Qu.:0.087042 1st Qu.:6.388 1st Qu.: 2.215323
# Median :0.301076 Median :7.390 Median : 4.138554
# Mean   :0.271930 Mean   :7.461 Mean   : 7.795348
# 3rd Qu.:0.415473 3rd Qu.:8.525 3rd Qu.: 9.410879
# Max.   :0.529639 Max.   :9.632 Max.   :27.129694
```

```
library(ggplot2)
```

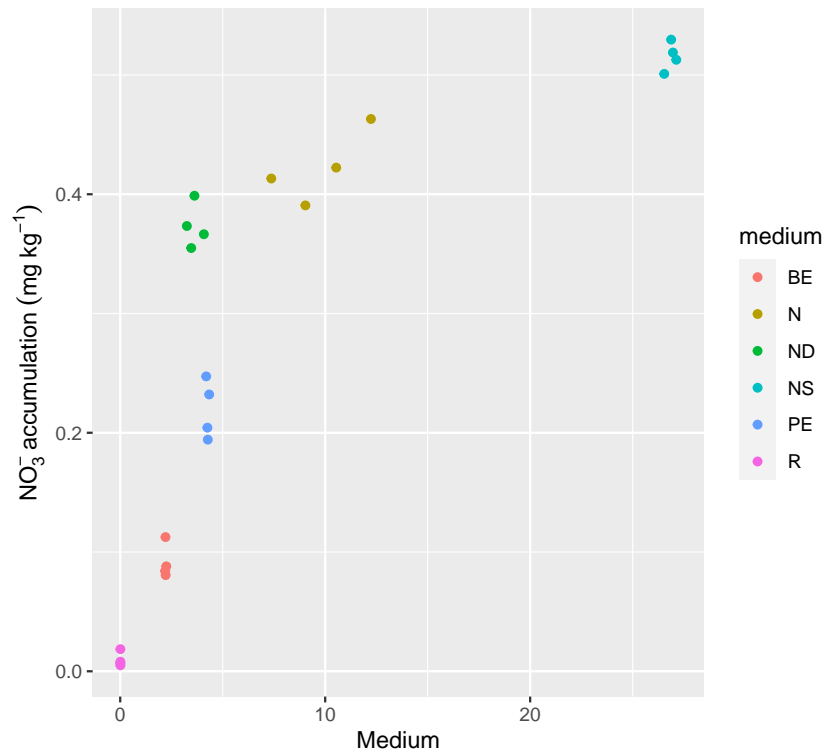
```
ggplot(lem, aes(medium, grow, colour = medium)) +
  geom_point() +
  labs(x = 'Medium', y = expression('Growth rate' ~ (mg ~ m ^ '-2' ~ d ^ '-1')))
```



```
ggplot(lem, aes(medium, NO3.accum, colour = medium)) +
  geom_point() +
  labs(x = 'Medium', y = expression(NO[3]^{-1} \sim 'accumulation' \sim (\text{mg} \sim \text{kg}^{-1})))
```



```
ggplot(lem, aes(NO3.med, NO3.accum, colour = medium)) +
  geom_point() +
  labs(x = 'Medium', y = expression(NO[3]^{-1}~'accumulation'~(mg~kg^{-1})))
```



18 Bibliography

- R. Devlamynck, M. Fernandes de Souza, M. Bog, J. Leenknecht, M. Eeckhout, and E. Meers. Effect of the growth medium composition on nitrate accumulation in the novel protein crop *Lemna minor*. *Ecotoxicology and Environmental Safety*, 206:111380, Dec. 2020. ISSN 0147-6513. doi: 10.1016/j.ecoenv.2020.111380. URL <https://www.sciencedirect.com/science/article/pii/S0147651320312173>.
- R. Diwakar. An evaluation of normal versus lognormal distribution in data description and empirical analysis. *Practical Assessment, Research & Evaluation*, 22:1–15, Dec. 2017. ISSN 1531-7714. URL <http://pareonline.net/getvn.asp?v=22&n=13>.
- J. J. Faraway. *Linear Models with R*. Number v. 63 in Texts in Statistical Science. Chapman & Hall/CRC, Boca Raton, 2005. ISBN 1-58488-425-8.
- J. J. Faraway. *Linear Models with Python*. Chapman and Hall/CRC, Boca Raton, 1st edition edition, Dec. 2020. ISBN 978-1-138-48395-8.
- F. García-Ávila. Treatment of municipal wastewater by vertical subsurface flow constructed wetland: Data collection on removal efficiency using *Phragmites Australis* and *Cyperus Papyrus*. *Data in Brief*, 30:105584, June 2020. ISSN 2352-3409. doi: 10.1016/j.dib.2020.105584. URL <https://www.sciencedirect.com/science/article/pii/S2352340920304789>.

- S. H. Hurlbert. Pseudoreplication and the Design of Ecological Field Experiments. *Ecological Monographs*, 54(2):187–211, 1984. ISSN 0012-9615. doi: 10.2307/1942661. URL <https://www.jstor.org/stable/1942661>.
- C. G. Justesen, S. Astals, J. R. Mortensen, R. Thorsen, K. Koch, S. Weinrich, J. M. Triolo, and S. D. Hafner. Development and validation of a low-cost gas density method for measuring biochemical methane potential (BMP). *Water*, 11(12):2431, Dec. 2019. doi: 10.3390/w11122431. URL <https://www.mdpi.com/2073-4441/11/12/2431>.
- K. Koch, T. Lippert, and J. E. Drewes. The role of inoculum’s origin on the methane yield of different substrates in biochemical methane potential (BMP) tests. *Bioresource Technology*, 243 (Supplement C):457–463, Nov. 2017. ISSN 0960-8524. doi: 10.1016/j.biortech.2017.06.142.
- P. McCullagh and J. A. Nelder. *Generalized Linear Models, Second Edition*. Chapman and Hall/CRC, second edition, Aug. 1989. ISBN 0-412-31760-5.
- R. G. D. Steel. *Principles and Procedures of Statistics: A Biometrical Approach*. McGraw-Hill, 1997. ISBN 978-0-07-061028-6.
- R. L. Wasserstein, A. L. Schirm, and N. A. Lazar. Moving to a World Beyond “ $p < 0.05$ ”. *The American Statistician*, 73(sup1):1–19, Mar. 2019. ISSN 0003-1305. doi: 10.1080/00031305.2019.1583913. URL <https://doi.org/10.1080/00031305.2019.1583913>.
- J. H. Zar. *Biostatistical Analysis*. Prentice Hall, Upper Saddle River, N.J, 4th ed edition, 1999. ISBN 0-13-081542-X.