Looking at data for TAM I

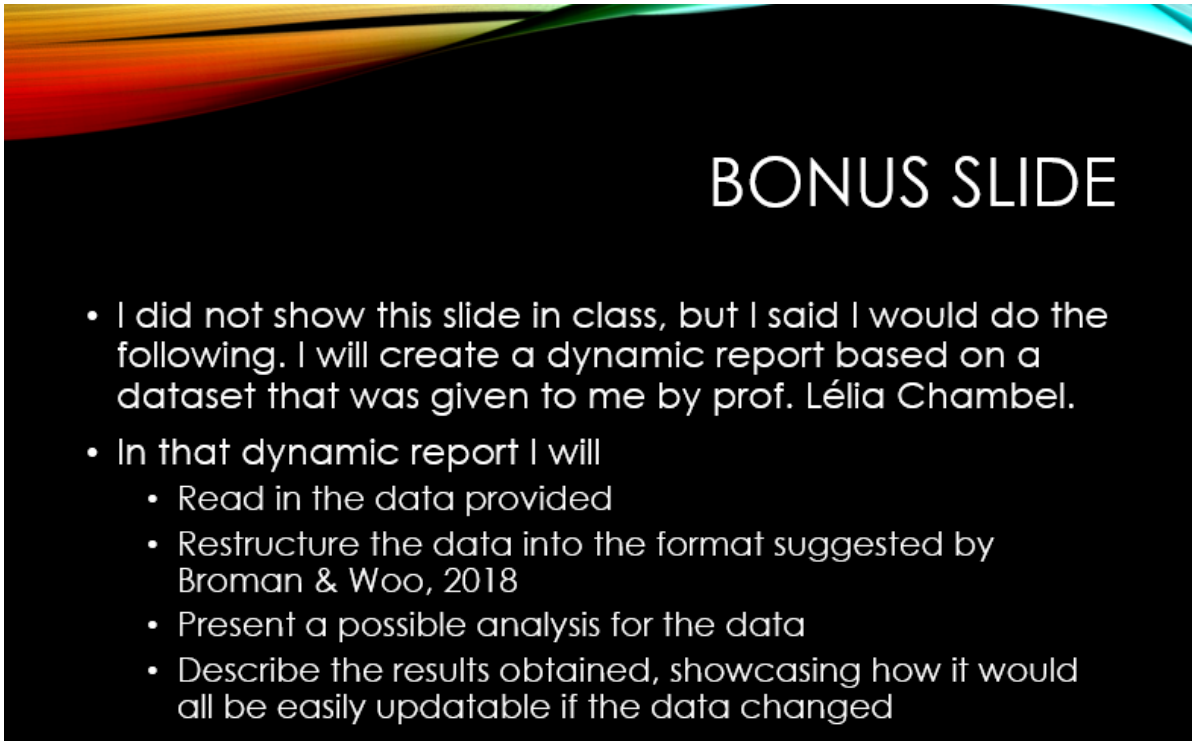
Introduction

```
library(readxl)
```

While preparing to deliver the lecture in Tópicos Avançados de Microbiologia I, we agreed it would be interesting to work over a dataset from Microbiologia to illustrate some of the concepts introduced in class.

Prof. Lélia Chambel (LC) suggested I could use a dataset they collected, to explore the growth of four different species of fungi under different conditions.

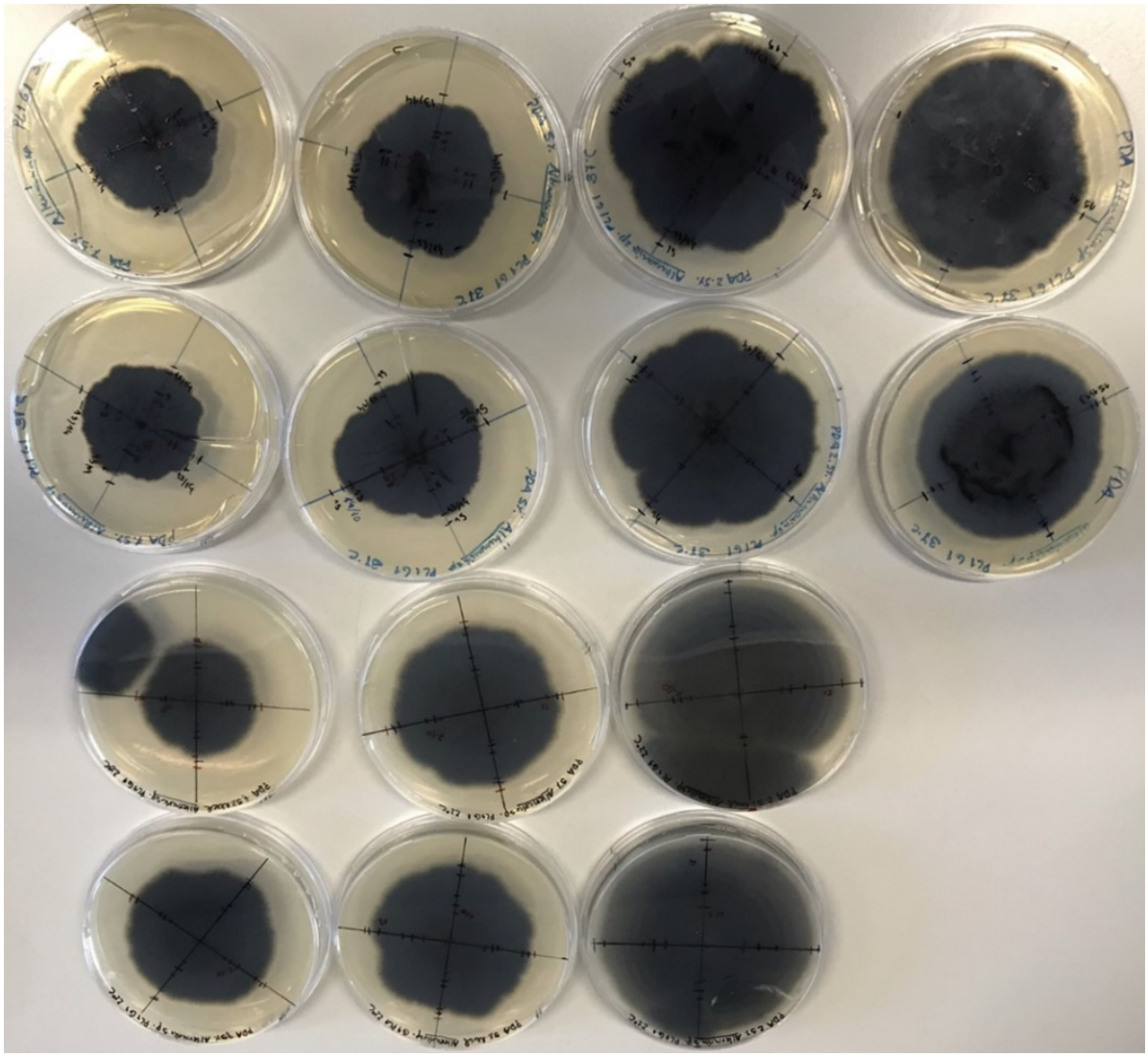
In this document I explore the dataset that was sent to me via email by LC on the Thu 9/14/2023 7:09 PM, to comply with the bonus slide I added to the material given to students:



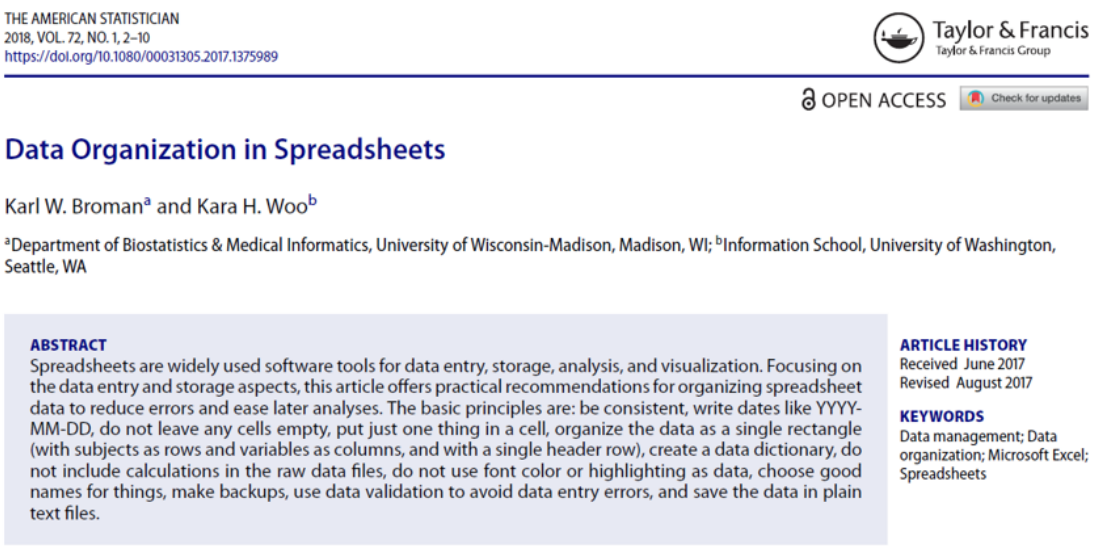
It contains, in Lélia's words, the following data:

"Sobre os dados é uma ótima ideia para passar da teoria à prática... Não sei se serão adequados mas, numa outra UC deste semestre, os alunos vão realizar um trabalho com fungos e avaliação do seu crescimento, ao longo do tempo, em diferentes condições (meios de cultura e temperaturas). Em anexo envio os dados obtidos, no ano anterior, para um dos fungos (analisamos 4 fungos diferentes). De uma forma muito geral, cada fungo é colocado em 4 meios diferentes (meio sem NaCl e meios com 3 concentrações diferentes de NaCl) e 3 temperaturas (22, 28, 37ºC). Em cada caso, o crescimento é medido ao longo de vários dias. O objetivo é avaliar o efeito do NaCl no crescimento dos fungos assim como da temperatura; perceber se os 4 fungos respondem de igual forma ou diferente. Estes dados são analisados utilizando uma análise de ANOVA (o colega Artur Lourenço faz essa análise com os alunos). Se te parecer que os dados são adequados para"outras análises" posso enviar os dos outros 3 fungos e podemos também falar um pouco pois será mais fácil explicar."

Here I reproduce an image with the fungi cultures:



So in this document I explore the actual growth data, highlighting some of the concepts and ideas I refer also in class, in particular, the fundamental aspects about how to structure one's data to facilitate effective data analysis, as described by Broman & Woo (2017).



THE AMERICAN STATISTICIAN
2018, VOL. 72, NO. 1, 2–10
https://doi.org/10.1080/00031305.2017.1375989

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Data Organization in Spreadsheets

Karl W. Broman[a] and Kara H. Woo[b]

[a]Department of Biostatistics & Medical Informatics, University of Wisconsin-Madison, Madison, WI; [b]Information School, University of Washington, Seattle, WA

ABSTRACT
Spreadsheets are widely used software tools for data entry, storage, analysis, and visualization. Focusing on the data entry and storage aspects, this article offers practical recommendations for organizing spreadsheet data to reduce errors and ease later analyses. The basic principles are: be consistent, write dates like YYYY-MM-DD, do not leave any cells empty, put just one thing in a cell, organize the data as a single rectangle (with subjects as rows and variables as columns, and with a single header row), create a data dictionary, do not include calculations in the raw data files, do not use font color or highlighting as data, choose good names for things, make backups, use data validation to avoid data entry errors, and save the data in plain text files.

ARTICLE HISTORY
Received June 2017
Revised August 2017

KEYWORDS
Data management; Data organization; Microsoft Excel; Spreadsheets

I strongly recommend anyone having to deal with data and Excel to read said paper.

To begin with, this is a dynamic report in RMarkdown (https://bookdown.org/yihui/rmarkdown/), which means that if the data were to change, the analysis and reporting would remain the same, we would just have to recompile the .Rmd. If additional data would be obtained, it should be just a matter of making small modifications to the code to deal with that, and then recompile the document.

Another advantage of it being a dynamic report in RMarkdown is that it can be output into different formats just by recompiling the document, including htlm, pdf or Word.

If you want to obtain a template to do your own RMarkdown documents, you can find one here:

https://github.com/TiagoAMarques/RMarkdownTemplate

Having dynamic documents allows one to quickly adapt to changes while making the analysis fully transparent, and hence easily reproducible.

In the following I will:

- Read the data in
- Manipulate the data
- Explore the data
- Implement some regression models to explore the effects of the different factors on the fungi growth
- Conclude with some final thoughts

Reading the data

First things first, lets read the data in.

The data is in an Excel workshhet, but in a format that makes it hard to read and to analyse in an automated and integrated versatile way. Following the guidelines of Broman & Woo (2017), and with LC's permission, I will use this data to illustrate how one could format data to make it amenable to any analysis one might want to conduct later.

In doing so I will prepare a document that might be distributed to the students in Tópicos Avançados de Microbiologia 1, to illustrate the points made by Broman & Woo (2017) regarding how to organize the data in spreadsheets.

A first thing to do is to define a suitable data object to hold all the data, which should be, as Broman & Woo (2017) suggest, a rectangular data structure with rows as records and columns as variables. Note that we will need at the very least columns for:

- size, the response variable, which I label `size`, and then, potential explanatory covariates
- species, as there are 4 different species, which I label `Sp` (we only have data for 1 pecies, Alternaria sp., but making it explicit would allow to add other species later)
- temperature, as we have 3 different temperatures (22, 28, 37^{o}C), the which I label `temp`
- concentration, as we have a medium without NaCl and mediums with 3 different concentrations of NaCl, which I label `conc`
- time, as measurements are made over time for the same sampling units, which I label `time`

we will also had a column to hold the number of the replicate (there are about 4 per combination, see below), which I will label `rep`.

Notice how in doing so I have considered two of the suggestions in Broman & Woo (2017), by:

- using variable names that are simple but intuitive to others, to ensure that the workflow is as transparent as possible, and
- creating a data dictionary, ensuring that others reading this document will know what everything is.

```
fungus<-data.frame(Sp=character(0),temp=numeric(0),conc=numeric(0),time=numeric(0),size=nu
```

Now, we can read all the data from the Excel file. This requires many lines of code, since the data is spread across multiple sub-tables and different sheets in the original document:

You might think that all these lines of code correspond to lots of work. This was necessary given the original format. but if you look at the code, it is mostly copy-paste of the same structure for each table, pointing at where to find the data for each replicate within each sheet

(corresponding to different NaCl concentrations), where results for the multiple temperatures are separated in sub-tables.

If you do NOT want to see all the above lines in your report, just change the `chunk` option `echo` in the .Rmd file from `echo=TRUE` to `echo=FALSE`.

Now that now we have the data in the format of Broman & Woo (2017), we can do anything we might want with it with a couple lines of code.

We also export the data as a flat text file (a .txt) for safety, as Broman & Woo (2017) suggest. And as they also suggest, do not forget to keep backup copies of your precious original data. The last thing you want is to have to redo all the analysis because a laptop is lot or some file gets corrupted. If something goes wrong, having a safety copy of the full dataset in fundamental.

```
write.table(fungus,file="fungus.txt",quote=FALSE)
```

Exploring the data

We start by exploring the data, making sure we have what we wanted to have. We can take a peak at the first few lines of data

```
head(fungus,n=10)
```

```
# A tibble: 10 x 6
   time   size Sp          temp  conc rep
   <chr> <dbl> <chr>      <dbl> <dbl> <chr>
 1 6        49 Alternaria    22     0 R1
 2 7        51 Alternaria    22     0 R1
 3 8        51 Alternaria    22     0 R1
 4 12       60 Alternaria    22     0 R1
 5 13       75 Alternaria    22     0 R1
 6 14       75 Alternaria    22     0 R1
 7 6        48 Alternaria    22     0 R2
 8 7        54 Alternaria    22     0 R2
 9 8        54 Alternaria    22     0 R2
10 12       57 Alternaria    22     0 R2
```

as well as the last few lines of data

```
tail(fungus,n=10)
```

```
# A tibble: 10 x 6
   time   size Sp          temp  conc rep
   <chr> <dbl> <chr>      <dbl> <dbl> <chr>
 1 13       40 Alternaria    37   7.5 R46
 2 14       45 Alternaria    37   7.5 R46
 3 5        20 Alternaria    37   7.5 R47
 4 6        28 Alternaria    37   7.5 R47
 5 7        28 Alternaria    37   7.5 R47
 6 14       48 Alternaria    37   7.5 R47
 7 5        21 Alternaria    37   7.5 R48
 8 6        27 Alternaria    37   7.5 R48
 9 7        27 Alternaria    37   7.5 R48
10 14       47 Alternaria    37   7.5 R48
```

if something wrong happened in the import, that might help diagnose it.

We can also see the structure of the data

```
str(fungus)
```

```
tibble [240 x 6] (S3: tbl_df/tbl/data.frame)
 $ time: chr [1:240] "6" "7" "8" "12" ...
 $ size: num [1:240] 49 51 51 60 75 75 48 54 54 57 ...
 $ Sp  : chr [1:240] "Alternaria" "Alternaria" "Alternaria" "Alternaria" ...
 $ temp: num [1:240] 22 22 22 22 22 22 22 22 22 22 ...
 $ conc: num [1:240] 0 0 0 0 0 0 0 0 0 0 ...
 $ rep : chr [1:240] "R1" "R1" "R1" "R1" ...
```

and look at an omnibus summary of the data

```
summary(fungus)
```

```
     time                size             Sp                  temp
 Length:240         Min.   :14.00   Length:240         Min.   :22
 Class :character   1st Qu.:31.00   Class :character   1st Qu.:22
 Mode  :character   Median :44.00   Mode  :character   Median :28
                    Mean   :47.46                      Mean   :29
                    3rd Qu.:63.00                      3rd Qu.:37
                    Max.   :85.00                      Max.   :37
      conc            rep
 Min.   :0.000   Length:240
```

```
 1st Qu.:1.875   Class :character
 Median :3.750   Mode  :character
 Mean   :3.750
 3rd Qu.:5.625
 Max.   :7.500
```

Something weird is happening with the time variable, as it seems like there is 1 value is not a number, which created issues with the import. We can track the issue to cell X35 in worksheet "PDA", where a "i" was recorded instead of a, presumably, 5.

That violates a few of the rules from Broman & Woo (2017), namely that one should be consistent, and using only one type of data for any given variable. It was just a slip up, but it could be enough to break a bioinformatics pipeline.

Respecting the rule that says an analyst should not change the original data, he should always ask the data owner to change the data instead and work from a clean master data file, but here I will change the data in R and then continue the analysis. I assume that instance of "i" was supposed to be 5. Note that I will use implicitly conditional code, such that if later the data gets corrected, my code still works the same.

```
fungus$time[fungus$time=="i"]<-5
fungus$time<-as.numeric(as.character(fungus$time))
```

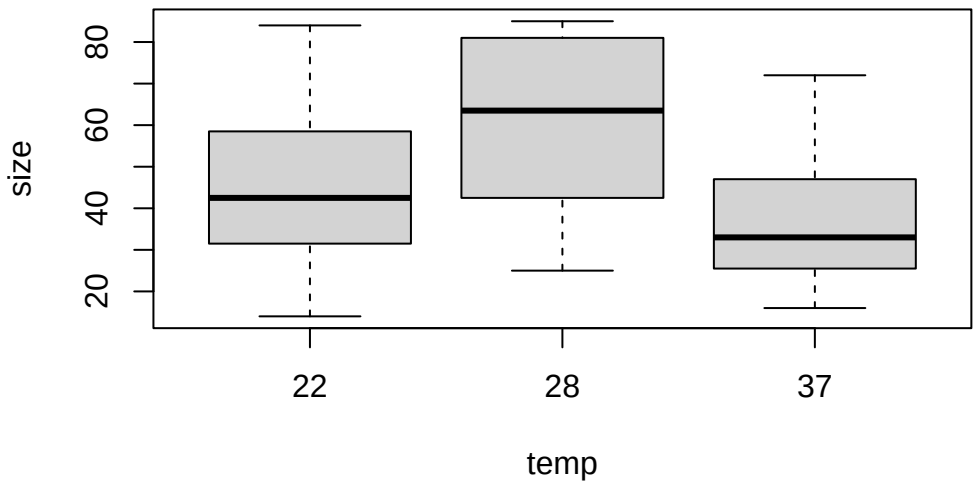
it all looks as it should, therefore we proceed.

As noted, with this structure we can do anything we'd like with it.

As examples, we could see the sizes as a function of temperatures, concentrations, or times, pooled across all the other factors.

First„size as a function of temperatures,

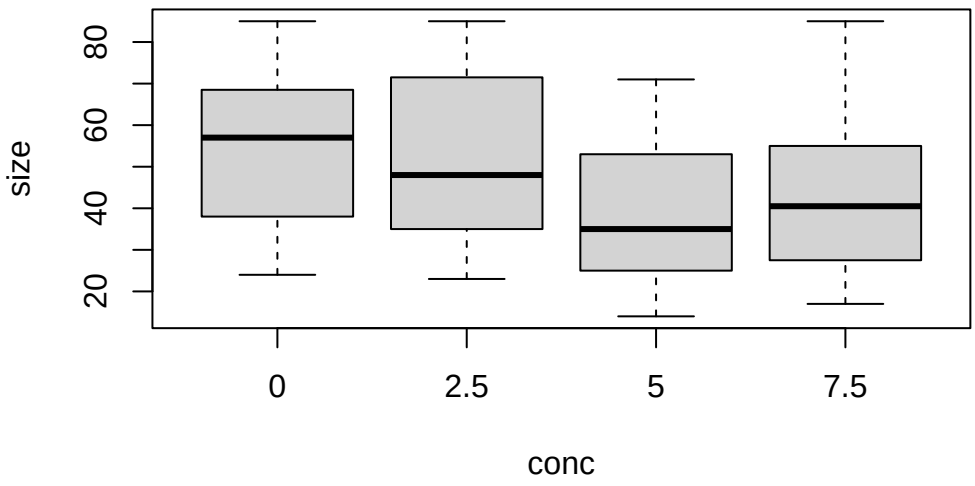
```
with(fungus,boxplot(size~temp))
```



which seems to show that sizes were larger at intermediate temperatures.

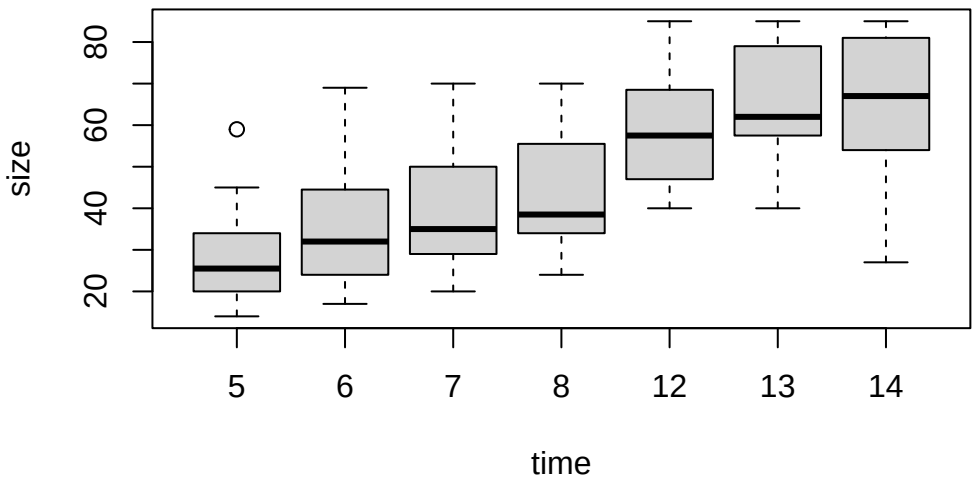
Then we focus on concentration of NACl:

```
with(fungus,boxplot(size~conc))
```



which seems to show that the larger the NaCl concentration, the smaller the sizes, and finally we look at the times

```
with(fungus,boxplot(size~time))
```



We can see a clear effect of increasing size with time, as would be expected.

Of course, interpreting those plots by themselves is not sensible, because once conditioning on a single factor, we ignore the variability and the sources of dependency across the other factors. As an example, if looking at the results from the different temperatures, we naively ignore that observations occur within replicates over time.

We can take a look at the number of observations per factor level

```
with(fungus,table(temp,conc))
```

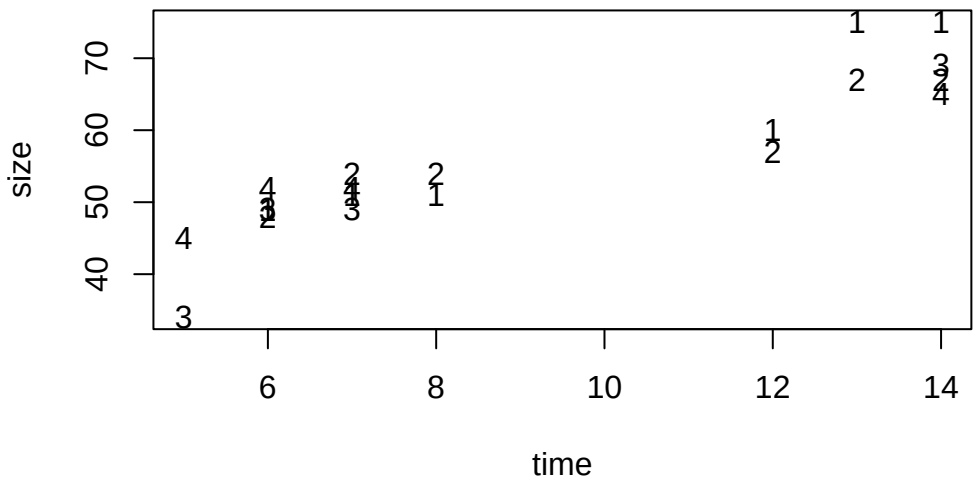
```
    conc
temp  0 2.5  5 7.5
  22 20  20 20  20
  28 20  20 20  20
  37 20  20 20  20
```

and we can see everything is almost perfectly balanced, with 20 observations for each of 3 by $4 = 12$ factorial combinations (just one with 21 observations).

However, as noted, we must be careful, as these do not correspond to independent data points. The data correspond to in general 4 replicates measured at 5 time points, and an adequate analysis must incorporate that correlation structure.

For illustration, We can take a look at one of the 12 factorial combinations, say for temperature = 22 and NaCl concentration = 0.

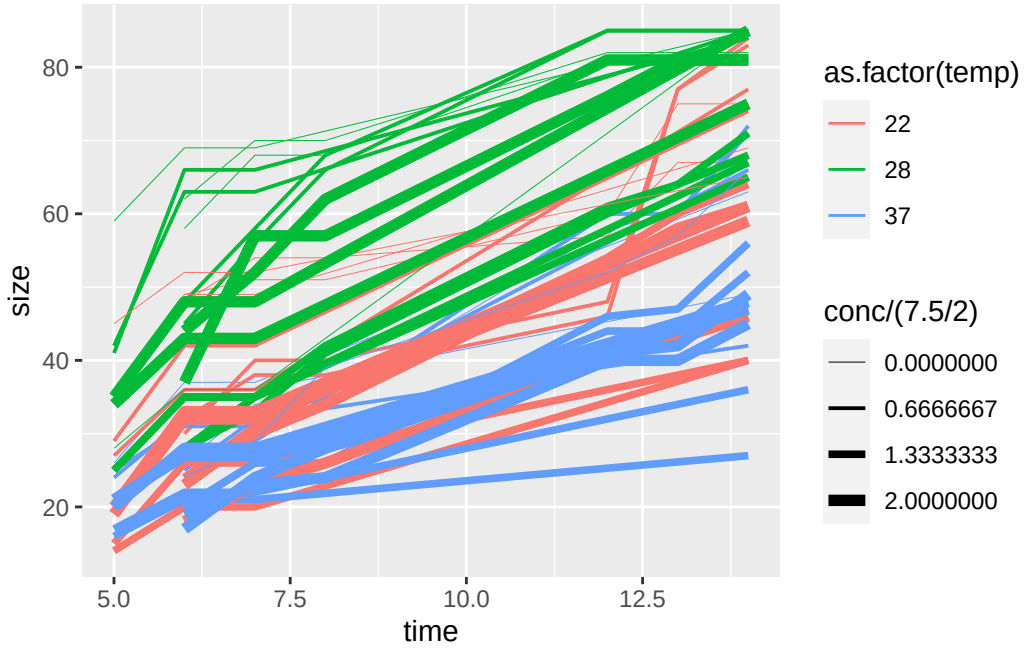
```
with((fungus[fungus$temp==22 & fungus$conc==0,]),plot(time,size,pch=substr(rep,2,2)))
```



It would be great to make a single plot with all the data, with lines joining the points corresponding to the different replicates, and say line widths representing levels of temperature, and line type representing the amount of NaCl.

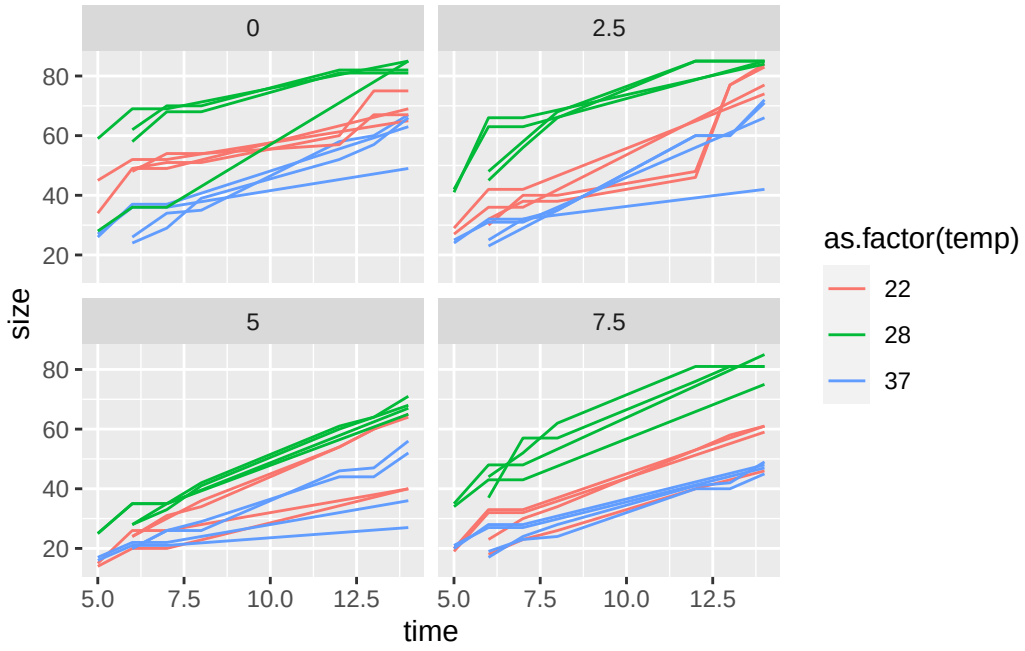
This is easily done using `ggplot2` capabilities, which allow us to look at the data from multiple angles. As an example, all the data in a single plot

```
library(ggplot2)
ggplot(data=fungus,aes(x=time,y=size,color=as.factor(temp),linewidth=conc/(7.5/2),group=re
geom_line()+scale_linewidth(range = c(0, 2),breaks=unique(fungus$conc/(7.5/2)))
```



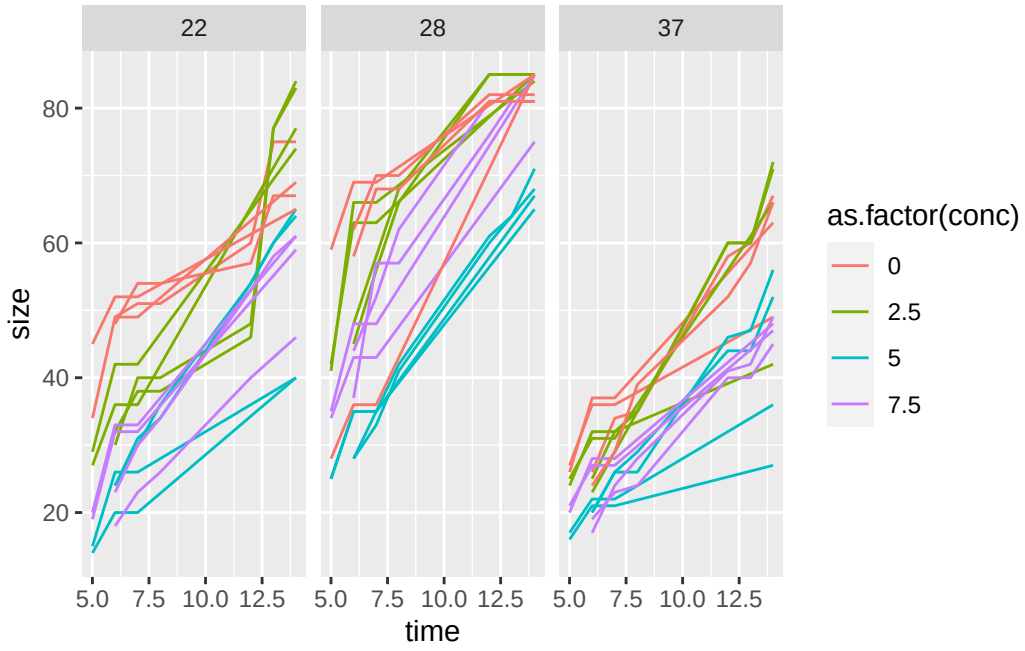
or if we prefer to separate for easier viewing, separate by level of concentration, and using color to highlight temperature

```
ggplot(data=fungus,aes(x=time,y=size,color=as.factor(temp),group=rep))+
geom_line()+facet_wrap(~conc)
```



or conversely, separate by temperature level, and using colour to highlight concentration

```
ggplot(data=fungus,aes(x=time,y=size,color=as.factor(conc),group=rep))+
geom_line()+facet_wrap(~temp)
```



Having now explored what the data looks like, we move on to try to model it.

Modelling the data

The endgame of many analysis is a model, which will be able to provide a means to make inferences based on the data, i.e., make genaral statements about the population from where the samples were collected.

To model the data data at hand will probably require a model with random effects, where the replicate itself is a random effect.

Since the data are sizes, hence strictly positive, a Gamma response variable might be a sensible option.

Nonetheless, we will start also by exploring Gaussian responses, as if we can keep it Normal, things are usually far easier.

As a first try, we consider a simple GLM, where we use a long link function to ensure positive predictions from the model. Remember, with a standard linear model predictions are not constraied in any way, hence, for a response variable that is strictly possible, one could obtain inadmissible estimates, which pose obvious problems.

```r
glmGam<-glm(size~time+conc+as.factor(temp),family=Gamma(link="log"),data=fungus)
glmGaul<-glm(size~time+conc+as.factor(temp),family=gaussian(link="log"),data=fungus)
glmGau<-glm(size~time+conc+as.factor(temp),family=gaussian,data=fungus)
summary(glmGam)
```

```
Call:
glm(formula = size ~ time + conc + as.factor(temp), family = Gamma(link = "log"),
    data = fungus)

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        3.110620   0.042016  74.033  < 2e-16 ***
time               0.088870   0.003537  25.124  < 2e-16 ***
conc              -0.049625   0.004340 -11.434  < 2e-16 ***
as.factor(temp)28  0.324668   0.029713  10.927  < 2e-16 ***
as.factor(temp)37 -0.196904   0.029713  -6.627 2.32e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.03531566)

    Null deviance: 44.9228  on 239  degrees of freedom
Residual deviance:  8.8602  on 235  degrees of freedom
AIC: 1710.6

Number of Fisher Scoring iterations: 4
```

```r
summary(glmGaul)
```

```
Call:
glm(formula = size ~ time + conc + as.factor(temp), family = gaussian(link = "log"),
    data = fungus)

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        3.204717   0.045130  71.011  < 2e-16 ***
time               0.078373   0.003429  22.858  < 2e-16 ***
conc              -0.039628   0.004113  -9.634  < 2e-16 ***
```

```
as.factor(temp)28  0.269472   0.026225  10.276  < 2e-16 ***
as.factor(temp)37 -0.191269   0.032728  -5.844 1.69e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 78.73116)

    Null deviance: 93578  on 239  degrees of freedom
Residual deviance: 18502  on 235  degrees of freedom
AIC: 1735.9

Number of Fisher Scoring iterations: 5
```

```
summary(glmGau)
```

```
Call:
glm(formula = size ~ time + conc + as.factor(temp), family = gaussian,
    data = fungus)

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        15.6923     1.8583   8.444 3.19e-15 ***
time                4.0497     0.1564  25.885  < 2e-16 ***
conc               -2.1200     0.1919 -11.045  < 2e-16 ***
as.factor(temp)28  15.3875     1.3142  11.709  < 2e-16 ***
as.factor(temp)37  -8.0125     1.3142  -6.097 4.40e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 69.08251)

    Null deviance: 93578  on 239  degrees of freedom
Residual deviance: 16234  on 235  degrees of freedom
AIC: 1704.5

Number of Fisher Scoring iterations: 2
```

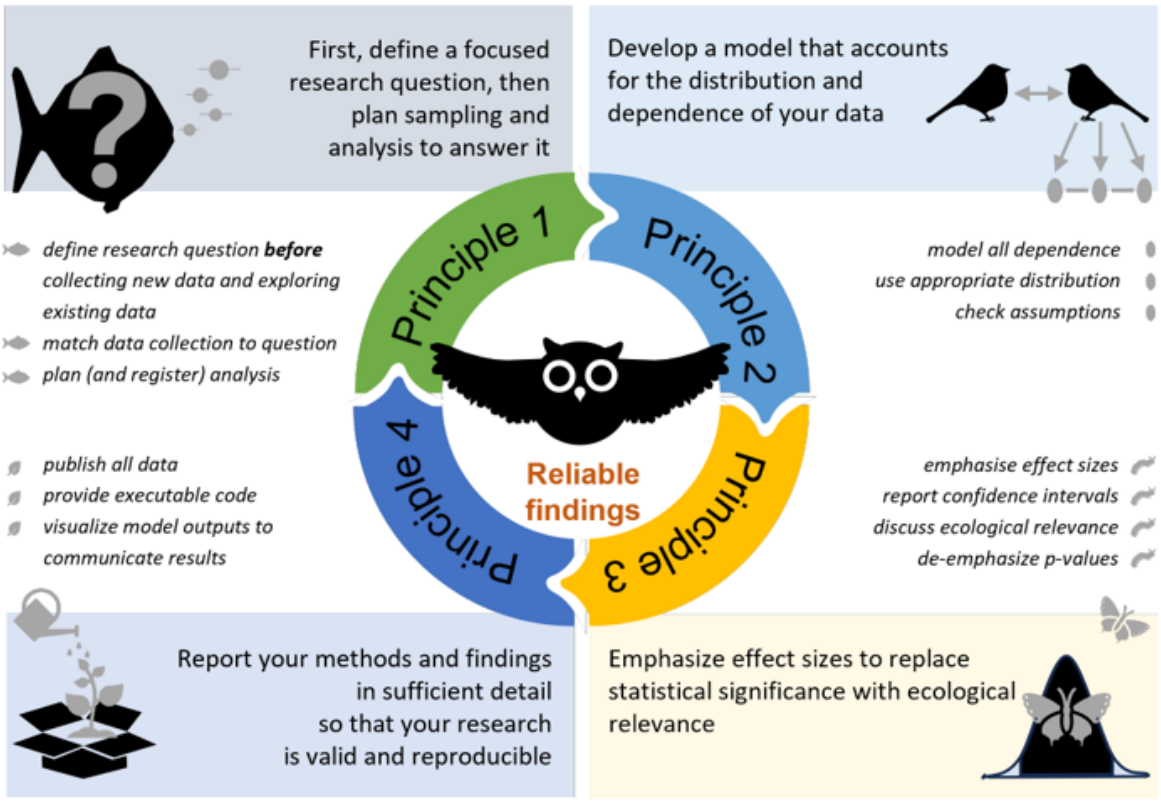
```
AIC(glmGau,glmGaul,glmGam)
```

```
         df      AIC
glmGau    6 1704.510
glmGaul   6 1735.887
glmGam    6 1710.597
```

As noted above, such analysis would ignore the dependencies in the observations over time within replicates, treating all the data points as independent.

Interestingly, the Gaussian model without a log link seems to be preferred, which could mean a Linear Model might be enough for this data.

I borrow here on an image presented in class, from Popovic et al. (in press). Four principles for improved statistical ecology. Methods in Ecology and Evolution.



highlighting that one of the key aspects of effective statistical data analysis is to consider a model that respects the structure and dependencies of the data.

Therefore, we attempt a GLMM with replicate as a random effect, to account for the fact that there is (temporal) autocorrelation in the data

```r
library(MASS)
glmm1<-glmmPQL(fixed=size~time+conc+as.factor(temp),family=gaussian,data=fungus,random=~1|
```

```
iteration 1
```

```
iteration 2
```

```r
summary(glmm1)
```

```
Linear mixed-effects model fit by maximum likelihood
  Data: fungus
  AIC BIC logLik
   NA  NA     NA

Random effects:
 Formula: ~1 | rep
        (Intercept) Residual
StdDev:    5.866081  5.91983

Variance function:
 Structure: fixed weights
 Formula: ~invwt
Fixed effects:  size ~ time + conc + as.factor(temp)
                     Value Std.Error  DF  t-value p-value
(Intercept)       16.059423 2.3161601 191  6.93364  0.0000
time               3.993666 0.1166033 191 34.25001  0.0000
conc              -2.101667 0.3368367  44 -6.23943  0.0000
as.factor(temp)28 15.170946 2.3061630  44  6.57844  0.0000
as.factor(temp)37 -7.967814 2.3061630  44 -3.45501  0.0012
 Correlation:
                  (Intr) time   conc   a.()28
time              -0.455
conc              -0.545  0.000
as.factor(temp)28 -0.498  0.000  0.000
as.factor(temp)37 -0.498  0.000  0.000  0.500

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.36037030 -0.60887767  0.03654345  0.57463095  2.66569230
```

```
Number of Observations: 240
Number of Groups: 48
```

All the models lead to similar conclusions. As would be expected, the size increases over time. As one increases the concentration of NaCl the growth seems to start from a lower level, while for intermediate temperatures the growths seems to start from a higher level.

We could now investigate further subtleties, as potential differences in slopes as a function of interaction between factors or in relation with the random effects, but that does not seem to be an obvious necessity, and further, is beyhond what was the objective of this document, to illustrate the advantages of having data in the format of Broman & Woo (2017).

Additionally, while we only looked at data from one species, we were told there was data from 3 additional species. One could:

1. repeat this procedure for the other 3 species
2. explore a simple integrated model for all species, eventually considering interactions between species and the factors

Conclusion

This document describes in a single integrated workflow, hence easily reproduced by others, a full analysis pipeline, considering:

1. Data import
2. Data manipulation
3. Data exploration
4. Data analysis
5. Data interpretation

By doing so in a single document that describes all the steps and includes, as needed, note, some or all the code to repeat them, we have enabled transparency and reproducibility. We did so while illustrating several of the principles

This is the kind of thing that I would like to see students being able to do with their own data.

While learning how to work with dynamic documents involves many concepts and tools and might seem to be behind an insurmountable steep learning curve at first, once you know your ways around R, that becomes a relatively easy task, with tangible and intangible dividends on the initial investment to be obtained over the course of a lifetime dealing with data.

If you are looking for an easy way to start learning R by yourself, there are many excelent and free options online, but I leave here a link to my own startup kit:

https://github.com/TiagoAMarques/AnIntro2RTutorial

Enjoy the learning!

Broman, K.W. & Woo, K.H. (2017). Data organization in spreadsheets. *The American Statistician*, **72**, 2–10.