

Laboratory exercises

The laboratory exercises outlined in the following pages are designed to allow you to develop some expertise in using statistical software (R) to analyze data. R is powerful statistical software but, like all software, it has its limitations. In particular, it is dumb: it cannot think for you, it cannot tell you whether the analysis you are attempting to do is appropriate or even makes any sense, and it cannot interpret your results.

General points to keep in mind

- Before attempting any statistical procedure, you must familiarize yourself with what the procedure is actually doing. This does not mean you actually have to know the underlying mathematics (although this certainly helps!), but you should at least understand the principles involved in the analysis. Therefore, before doing a laboratory exercise, read the appropriate section(s) in the lecture notes. Otherwise, the output from your analyses - even if done correctly - will seem like drivel.
- The laboratories are designed to complement the lectures, and *vice versa*. Owing to scheduling constraints, it may not be possible to synchronize the two perfectly. But feel free to bring questions about the laboratories to class, or questions about the lectures to the labs.
- Work on the laboratories at your own speed: some can be done much more quickly than others, and one laboratory need not correspond to one laboratory session. In fact, for some laboratories we have allotted two laboratory sessions. Although you will not be "graded" on the laboratories per se, be aware that completing the labs is essential. If you do not complete the labs, it is very unlikely that you will be able to complete the assignments and the final exam/term paper. So take these laboratories seriously!
- The objective of the first lab is to allow you to acquire or review the minimum knowledge required to complete the following laboratory exercises with R. There are always several methods to accomplish something in R, but you will only find simple ways in this manual. Those amongst you that want to go further will easily find many examples of more detailed and sophisticated methods. In particular, I point you to the following resources:

R for beginners

http://cran.r-project.org/doc/contrib/Paradis-rdebut_en.pdf

Using R for psychological research: A simple guide to an elegant package

<http://www.personality-project.org/r/>

An introduction to R

<http://cran.r-project.org/doc/manuals/R-intro.html>

If you prefer paper books, the CRAN web site has a commented list at:

<http://www.r-project.org/doc/bib/R-books.html>

Finally, I also recommend this short reference card:

R reference card by Tom Short

<http://cran.r-project.org/doc/contrib/Short-refcard.pdf>

What is R and why use it in this course

R is multiplatform free software forming a system for statistical computation and graphics. R is also a programming language specially designed for statistical data analysis. It is a dialect of the S language. S-Plus is another dialect of the S language, very similar to R, incorporated into a commercial package. S-Plus has a built-in graphical design interface that some find convivial.

R has 2 major advantages for this course. Initially, you will find that it also has one inconvenience. However, this “inconvenience” will rapidly force you to acquire very good working habits. So, I see it as a third advantage.

The first advantage is that you can install it freely on your personal computer(s). This is important because it is by doing analyses that you will learn and eventually master biostatistics. This implies that you have easy and unlimited access to a statistical software package.

The second advantage is that R can do everything in statistics. R was conceived to be extensible and has become the preferred tool for statisticians around the world. The question is not “Can R do this?” but rather “How can I do this in R?”. And Google is your friend.

No other software package offers you these two advantages.

The inconvenience of R is that one has to type commands (or copy and paste code) rather than use a menu and select options. If you do not know what command to use, nothing will happen. It is therefore not that easy when you start. However, it is possible to rapidly learn to make basic operations (open a data file, plot data, and run a simple

analysis). And once you understand the operating principle, you can easily find examples on the Web for more complex analyses and graphs for which you can adapt the code.

This is exactly what you will do in the first lab to familiarize yourself with R.

Why is this inconvenience really an advantage in my mind? Because this way of doing things is more efficient and will save you time on the long run. I guarantee it. Believe me, you will never do an analysis only once. As you'll proceed through analyses, you will find data entry errors, discover that the analysis must be run separately for subgroups, find extra data, have to rerun the analysis on transformed data, or you will make some analytical error along the way. If you use a graphical interface with menus, redoing an analysis implies that you relick here, enter values there, select some options, etc. Each of these steps is a potential source of error. If, instead, you use lines of codes, you only have to fix the code and submit to repeat instantaneously the entire analysis. And you can perfectly document what you did, leaving an audit trail for the future. This is how pros work and can document the quality of the results of their analyses.

Installation

To install R on a computer, go to <http://cran.r-project.org/>. You will find compiled versions (binaries) for your preferred operating system (Windows, MacOS, Linux).

The version I installed when I started these notes is 3.0.1 for Windows (36 Mbytes). I installed it with the default options, although along the way I installed quite a number of extra packages.

Note : R has already been installed on the lab computers (the version may be slightly different, but this should not matter).

General lab instructions

- Bring a USB key or equivalent so you can save your work. Alternatively, email your results to yourself.
- Read the lab exercise before coming to the lab. Read the R code and come with questions about the code.
- During pre-labs, listen to the special instructions
- Do the laboratory exercises at your own rhythm, in teams. Then, I recommend that you start (complete?) the lab assignment so that you can benefit from the presence of the TA or prof.

- During your analyses, copy and paste results in a separate document, for example in your preferred word processing program. Annotate abundantly
- Each time you shut down R, save the history of your commands (ex: labo1.1 rHistory, labo1.2.rHistory, etc). You will be able to redo the lab rapidly, get code fragments, or more easily identify errors.
- Create your own “library” of code fragments (snippets). Annotate it abundantly. You will thank yourself later.

And for the dilettantes or those completely refractory to programming...

If you find the absence of a graphical user interface too spartan or if you have recurring problems finding your way that make you waste too much time, then install and load the graphical interface distributed by CRAN and that works under Windows, Mac and Linux (so they say, although I only tested it under Windows).

The library (module) is called Rcmdr. It is not part of the base distribution, and you will need to install it first into R on your own computer (it has been installed already in the lab, but is not loaded automatically when you start R). To install this package, you need an internet connection. In the command window, enter the command:
install.packages("Rcmdr", dependencies=TRUE)

A window will open to let you choose one of the web sites that distribute Rcmdr. The download and installation will take several minutes because several other libraries are needed and will be automatically downloaded. Once the Rcmdr library will have been installed, you will need to start it to get to the graphical interface. To do so, simply enter the command:

library(Rcmdr)

A new window will open, with menus to access most of the common commands. One of the panels of the Rcmdr window is a log of the R commands created by Rcmdr as you make choices in the graphical user interface. Study this code to learn, maybe more easily, the R language.

Lab- R tutorial

After completing this laboratory exercise, you should be able to:

- Open R data files
- Import rectangular data sets
- Export R data to text files
- Verify that data were imported correctly
- Examine the distribution of a variable
- Examine visually and test for normality of a variable
- Calculate descriptive statistics for a variable
- Transform data

Importing and exporting data

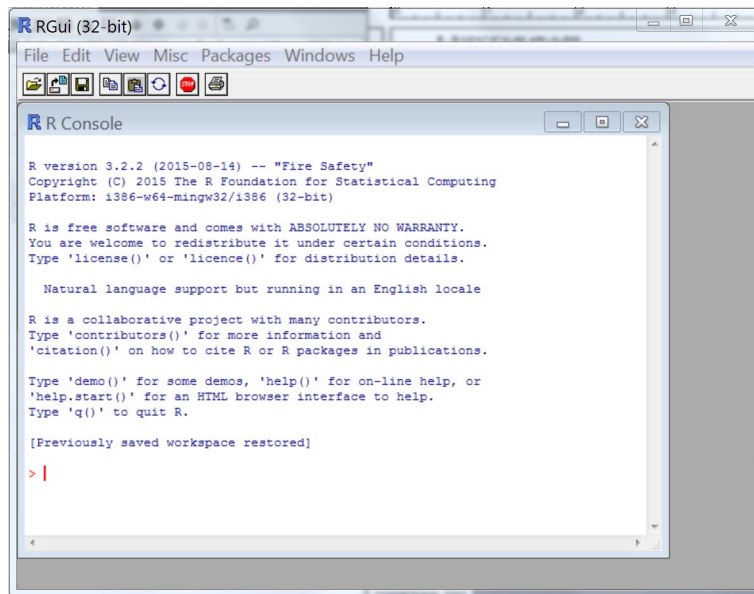
Open and save an R data file

Data for the lab exercises and assignments are provided to you in R format (files with extension .Rdata). To open these files, if your operating system allows it, you can simply click on the file to start a new R session. Alternatively, once R is started, you can type the following at the R console window:

```
load(file.choose())
```

to open a dialog box that will allow you to navigate to a folder on your computer to choose the file you want to load. After having made your selection (choose the file ErablesGatineau.Rdata), you will be returned to the R console window with no apparent change:

Figure 1.



To verify that the data were read and loaded properly, you can list all objects in memory with the `ls()` function, or get a more detailed description with `ls.str()`:

```
> ls()
[1] "ErablesGatineau"
> ls.str()
ErablesGatineau : 'data.frame': 100 obs. of 3 variables:
 $ station: Factor w/ 2 levels "A",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ diam : num 22.4 36.1 44.4 24.6 17.7 ...
 $ biom : num 732 1171 673 1552 504 ...
```

R confirms that the object `ErablesGatineau` is in memory. `ErablesGatineau` is a `data.frame` that contains 100 observations (lines) of 3 variables (columns): `station`, a variable of type `Factor` with 2 levels, and `diam` and `biom` that are 2 numeric variables.

Best practice: To ensure that the same file is used the next time you run your script, you should record the equivalent `load()` command line, as in: `load("C:/Users/hrundle/Dropbox/BIO 4158 - Biostats/Lab_data/ErablesGatineau.RData")`

Entering data in R

R is not the ideal environment to input data. It is possible, but the syntax is heavy and makes most people upset. Use your preferred worksheet program instead. It will be more efficient and less frustrating.

Cleaning up / correcting data

Another operation that can be frustrating in R. Our advice: unless you want to keep track of all corrections made (so that you can go back to the original data), do not change data in R. Return to the original data file (in a worksheet or database), correct the data there, and then reimport into R. It is simple to resubmit the few lines of code to reimport data. Doing things this way will leave you with a single version of your data file that has all corrections, and the code that allows you to repeat the analysis exactly.

Import data from Excel.

Save the data from Excel in a .csv format (**File>Save as**).

Import these data into R with the command `read.csv`.

For example, to create a data frame named `age` from a csv file on disk, you can type the command: :

```
>
age<-read.csv(file.choose())
```

(If you do this from the file called `age.csv` that I exported from the Excel file `age.xls`, and then type `age` to list the data frame contents, you will get:

```
> age
  ageclass sex count
1 0-9 female 17619
2 0-9 male 17538
3 10-19 female 17947
4 10-19 male 18207
5 20-29 female 21344
6 20-29 male 21401
7 30-39 female 19138
8 30-39 male 18837
9 40-49 female 13135
10 40-49 male 12568
11 50-59 female 11617
12 50-59 male 10661
13 60-69 female 11053
14 60-69 male 9374
15 70-79 female 7712
16 70-79 male 5348
17 80+ female 4114
18 80+ male 1926
```

Trap : Be careful if you use a language where the comma is used as a decimal point. By default, R uses decimal points and you will not get the result you expect. There is a modified version of `read.csv()` called `read.csv2()` that fixes this problem. Google it if in need.

Exporting data from R.


```
write.csv(mydata, file = "outfilename.csv", row.names =
FALSE)
```

where `mydata` is the data frame name to be exported and `outfile.csv` is the name of the file to be exported. Note that this file will be created in the working directory (which can be changed using the menu **File>Change dir**, or by the command `setwd()`)


Preliminary examination of data


The first step of data analysis is to examine the data at hand. This examination will tell you if the data were correctly imported, whether the numbers are credible, whether all data came in, etc. This initial data examination often will allow you to detect unlikely observations, possibly due to errors at the data entry stage. Finally, the initial plotting of the data will allow you to visualize the major trends that will be confirmed later by your statistical analysis.

The file `sturgeon.Rdata` contains data on sturgeons from the Saskatchewan River. These data were collected to examine how sturgeon size varies among sexes (`sex`), sites (`location`), and years (`year`).

 Make sure you start from a blank slate by removing all objects in memory by typing the command

```
rm(list=ls())
```


 Open the file `sturgeon.Rdata`.

 To get a list of the content of the data file loaded in memory, type the command

```
ls.str()
```

```
sturgeon : 'data.frame':      186 obs. of  9 variables:
 $ fklngth : num  37 50.2 28.9 50.2 45.6 ...
 $ totlngth: num  40.7 54.1 31.3 53.1 49.5 ...
 $ drlngth : num  23.6 31.5 17.3 32.3 32.1 ...
 $ rdwght  : num  15.95 NA 6.49 NA 29.92 ...
 $ age     : num  11 24 7 23 20 23 20 7 23 19 ...
 $ girth   : num  40.5 53.5 31 52.5 50 54.2 48 28.5 44 39 ...
 $ sex     : Factor w/ 2 levels "FEMALE","MALE": 2 1 2 1 2 1 1 2 2 2 ...
 $ location: Factor w/ 2 levels "CUMBERLAND","THE_PAS": 2 2 2 2 2 2 2 2 2 2 ...
 ...
 $ year    : Factor w/ 3 levels "1978","1979",...: 1 1 1 1 1 1 1 1 1 1 ...
```

Summary statistics

 To get summary statistics on the contents of the data frame `sturgeon`, type the command:

```
summary(sturgeon)
```

```
      fklngth      totlngth      drlngth
M n.   :24.96   M n.   :28.15   M n.   :14.33
1st Q. :41.00   1st Q. :43.66   1st Q. :25.00
```

```

Medi an : 44.06   Medi an : 47.32   Medi an : 27.00
Mean    : 44.15   Mean    : 47.45   Mean    : 27.29
3rd Qu. : 48.00   3rd Qu. : 51.97   3rd Qu. : 29.72
Max.    : 66.85   Max.    : 72.05   Max.    : 41.93
NA' s   :         NA' s   : 85.00   NA' s   : 13.00

  r dwght      age      gi r t h
M n.    : 4.73   M n.    : 7.00   M n.    : 11.50
1st Qu. : 18.09  1st Qu. : 17.00  1st Qu. : 40.00
Medi an : 23.10  Medi an : 20.00  Medi an : 44.00
Mean    : 24.87  Mean    : 20.24  Mean    : 44.33
3rd Qu. : 30.27  3rd Qu. : 23.50  3rd Qu. : 48.80
Max.    : 93.72  Max.    : 55.00  Max.    : 73.70
NA' s   : 4.00   NA' s   : 11.00  NA' s   : 85.00

sex      locat ion      year
FEMALE: 106  CUMBERLAND: 85  1978: 45
MALE   : 80  THE_PAS   : 101  1979: 68
                               1980: 73

```

For each variable, R lists the minimum, the maximum, the median that is the 50th percentile, here the 93rd value of the 186 observations ordered in ascending order, values at the first (25%) and third quartile (75%), and the number of missing values in the column.

Note that several variables have missing values (NA). Only the variables `fklength` (fork length), `sex`, `location`, and `year` have 186 observations.

Trap : Beware of missing values. Several R functions are sensitive to missing values and you will frequently have to do your analyses on data subsets without missing data, or by using optional parameters in various commands. We will get back to this, but you should always pay attention and take note of missing data when you do analyses.

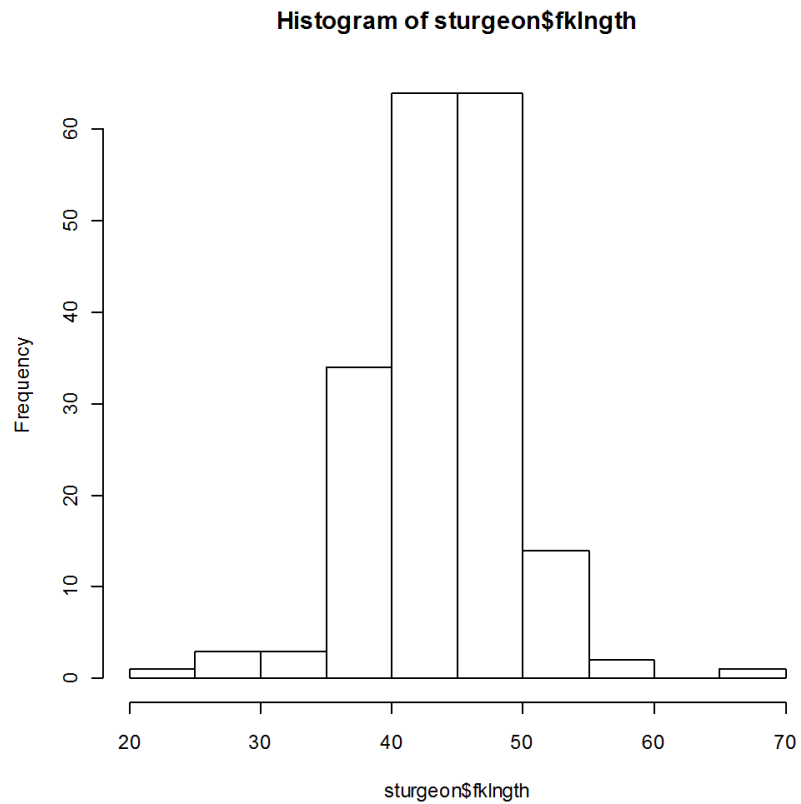
Histogram, empirical probability density, boxplot, and visual assessment of normality.

Let's look more closely at the distribution of `fklength`.

The command `hist()` will create a histogram. For the histogram of `fklength` in the `sturgeon` data frame, type the command:

```
hist(sturgeon$fklength)
```

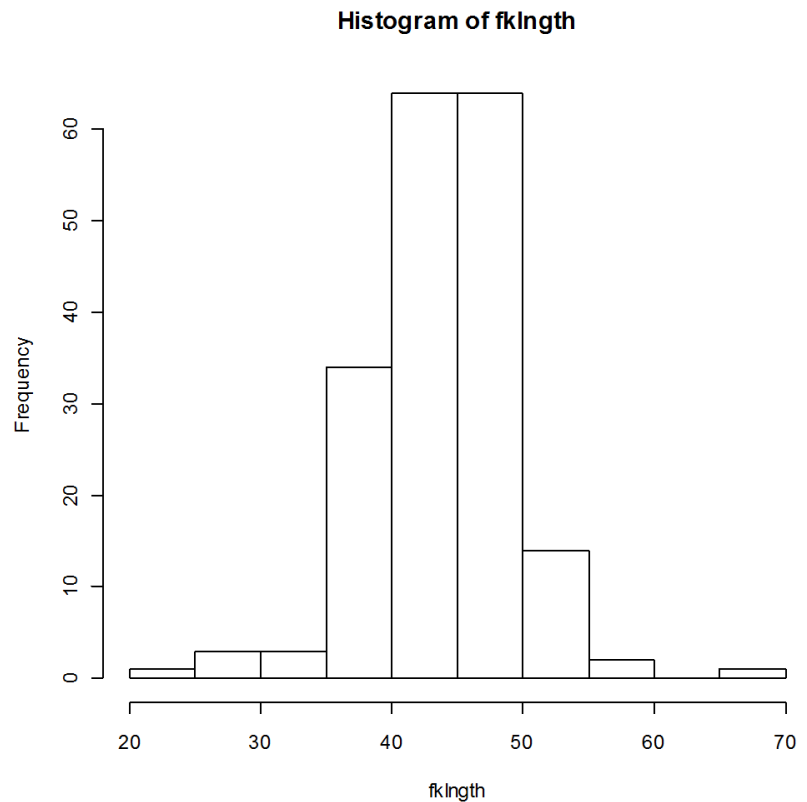
Figure 2.



The data appear to be approximately normal. This is good to know. Note that this syntax is a bit heavy as you need to prefix variable names by the data frame name. You can lighten the syntax by making the variables directly accessible by commands by typing the command `attach()`

```
attach(sturgeon)  
hist(fklnth)
```

Figure 3.



This histogram (Fig. 3) is a very classical representation of the distribution. Histograms are not perfect however because their shape partly depends on the number of bins used, more so for small samples. One can do better, especially if you want to visually compare the observed distribution to a normal distribution. But you need to come up with a bit of extra R code:

```
require(ggplot2)
# use "sturgeon" dataframe to make plot called mygraph
# and define x axis as representing fklngh
mygraph <- ggplot(sturgeon, aes(x = fklngh))
# add data to the mygraph ggplot
mygraph <- mygraph +

# add data density smooth
geom_density() +
# add rug (bars at the bottom of the plot)
geom_rug() +
# add black semitransparent histogram
geom_histogram(aes(y = ..density..),
               color = "black",
               alpha = 0.3) +
```

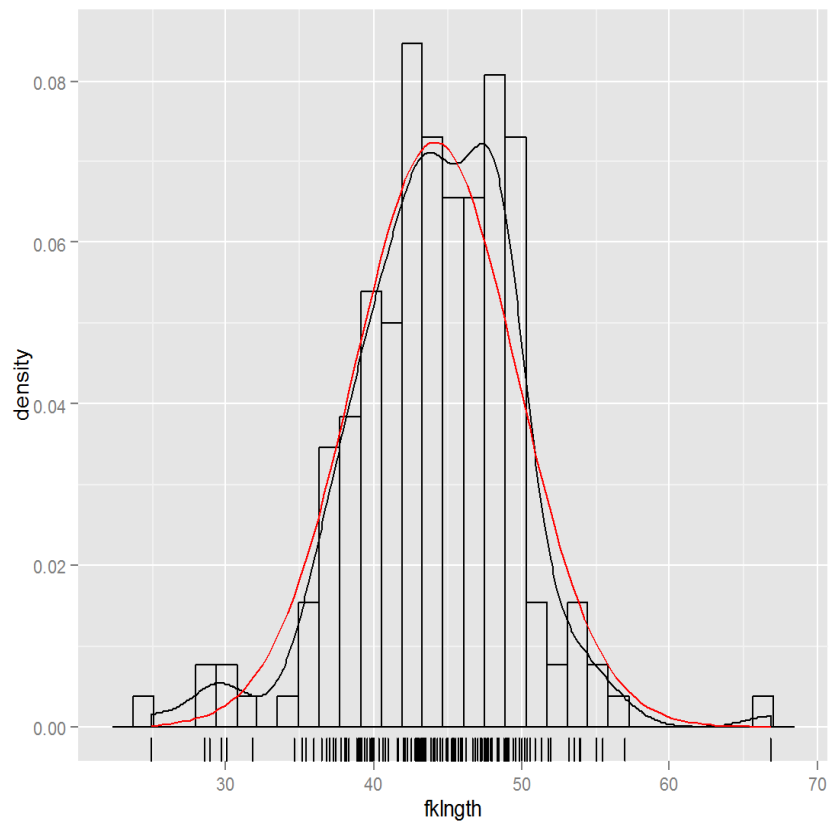
```

# add normal curve in red, with mean and sd from fklngth
stat_function(fun = dnorm,
             args = list(
               mean = mean(sturgeon$fklngth),
               sd = sd(sturgeon$fklngth)
             ),
             color = "red")
#display graph
mygraph

```

You should get :

Figure 4.



Each observation is represented by a short vertical bar below the x-axis (rug). The red line is the normal distribution with the same mean and standard deviation as the data. The other line is the empirical distribution, smoothed from the observations.

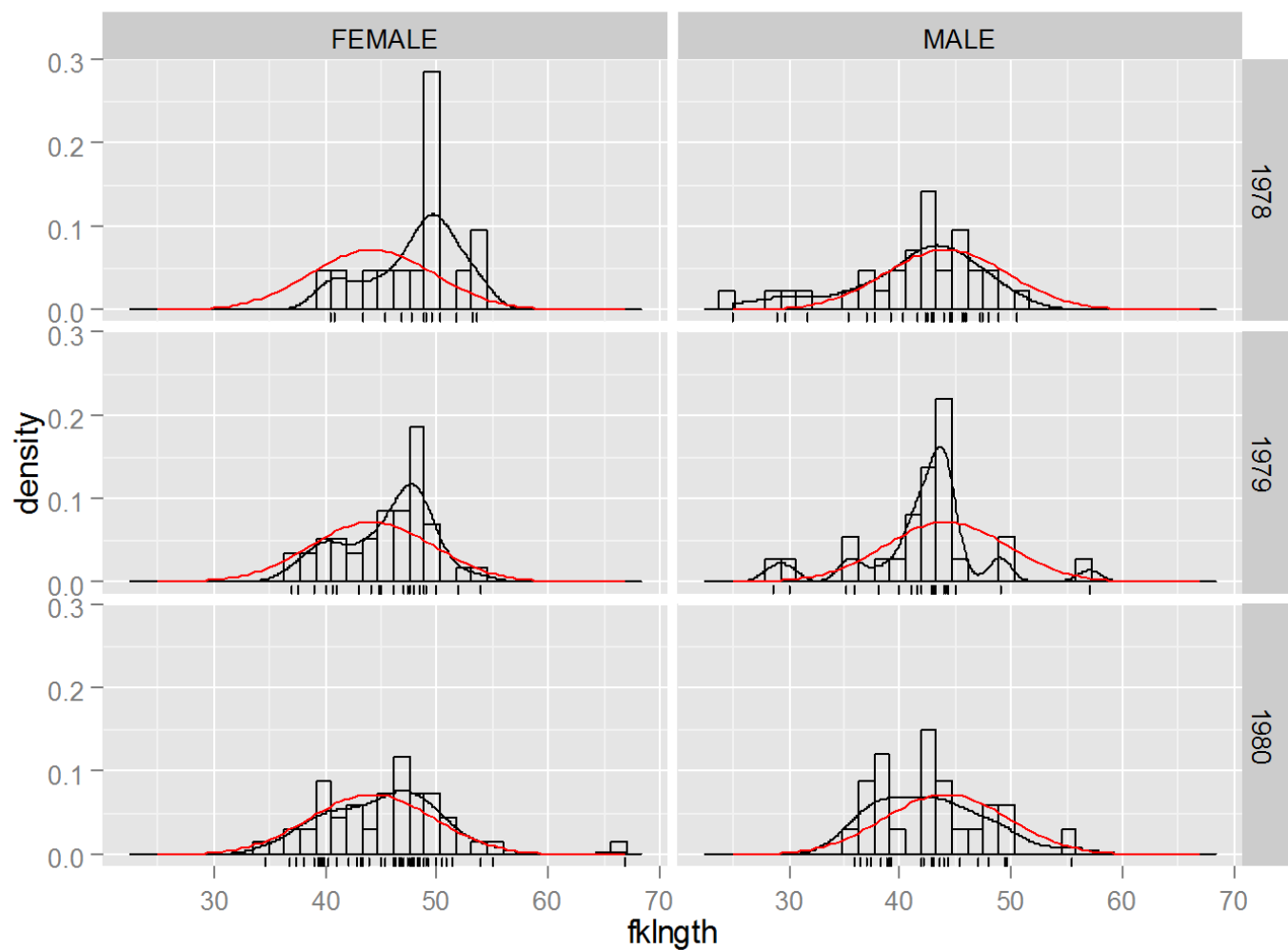
The ggplot object you just created (mygraph) can be further manipulated. For example, you can plot the distribution of fklngth per sex and year groups simply by adding a facet_grid() statement:

```

:
mygraph + facet_grid(year ~ sex)

```

Figure 5.

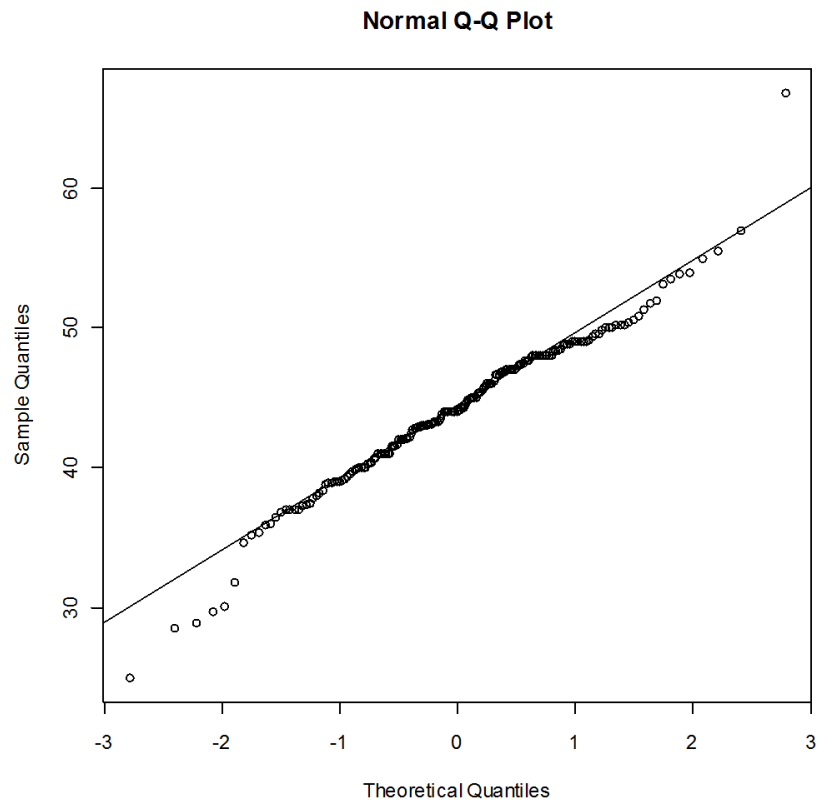


Each panel contains the data distribution for one sex that year, and the recurring red curve is the normal distribution for the entire data set. It can serve as a reference to help visually evaluate differences among panels.

Another way to visually assess normality of data is the QQ plot that is obtained by the pair of commands `qqnorm()` and `qqline()`,

```
qqnorm(fklngth)
qqline(fklngth)
```

Figure 6.



Perfectly normal data would follow the straight diagonal line. Here there are deviations in the tails of the distribution and a bit to the right of the center. Compare this representation to the two preceding graphs. You will probably agree that it is easier to visualize how data deviate from normality by looking at a histogram of an empirical probability density than by looking at the QQ plots. However, QQ plots are often automatically produced by various statistical routines and you should be able to interpret them. In addition, one can easily run a formal test of normality in R with the command `shapiro.test()` that computes a statistic (W) that measures how tightly data fall around the straight diagonal line of the QQ plot. If data fall perfectly on the line, then $W=1$. If W is much less than 1, then data are not normal.

For the `fklength` data :

```
shapiro.test(fklength)
```

```
Shapiro-Wilk normality test
```

```
data:  fklength  
W = 0.9722, p-value = 0.0009284
```

W is close to 1, but far enough to indicate a statistically significant deviation from normality.

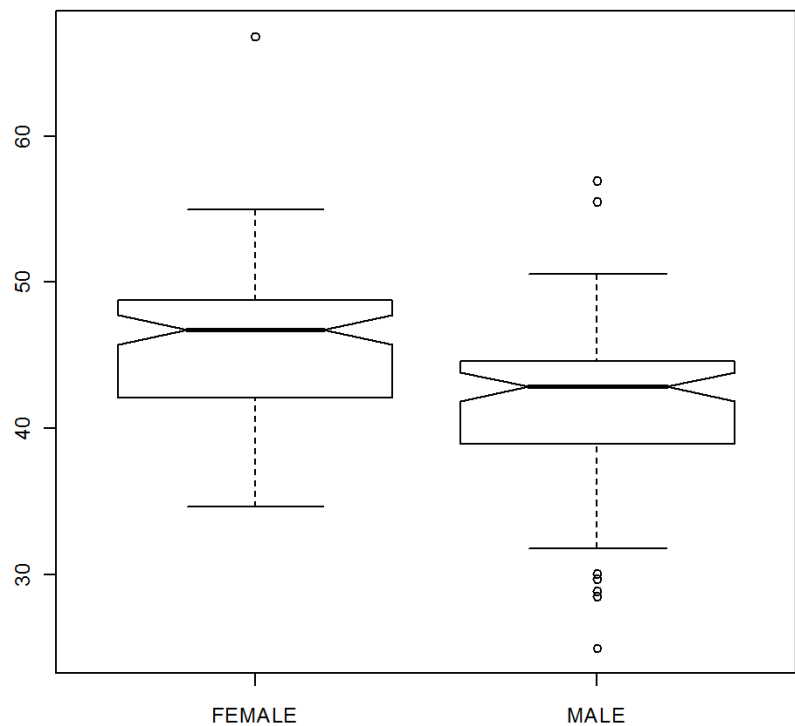
Visual examination of very large data sets is often made difficult by the superposition of data points. Boxplots are an interesting alternative.

The command

```
boxplot(fklngh~sex, notch=TRUE)
```

produces a boxplot of fklngh for each sex , and adds whiskers..

Figure 7.



The slightly thicker line inside the box of Fig. 7 indicates the median. The width of the notch is proportional to the uncertainty around the median estimate. One can visually assess the approximate statistical significance of differences among medians by looking at the overlap of the notches (here there is no overlap and one could tentatively conclude that the median female size is larger than the median male size). Boxes extend from the first to third quartile (the 25th to 75th percentile if you prefer). Bars (whiskers) extend above and below the

boxes from the minimum to the maximum observed value or, if there are extreme values, from the smallest to the largest observed value within 1.5x the interquartile range from the median. Observations exceeding the limits of the whiskers (hence further away from the median than 1.5x the interquartile range, the range between the 25th and 75th percentile) are plotted as circles. These are outliers, possibly aberrant data.

Scatterplots

In addition to histograms and other univariate plots, it is often informative to examine scatter plots. The command `plot(y~x)` produces a scatter plot of y on the vertical axis (the ordinate) vs x on the horizontal axis (abscissa).


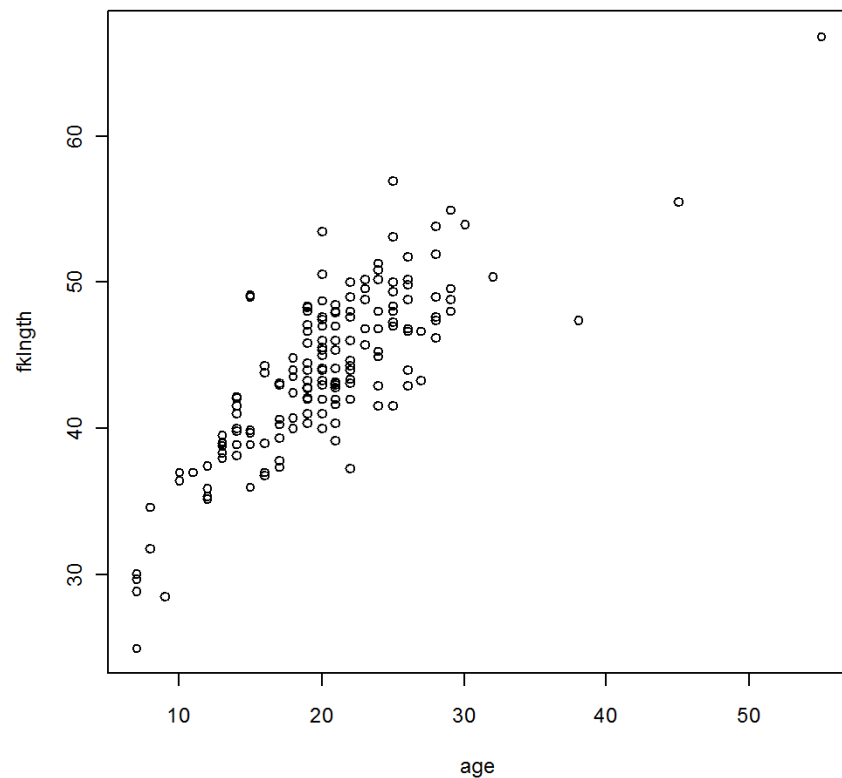
 Create a scatterplot of `fklngh` vs `age` using the `plot()` command. You should obtain:

Figure 8.



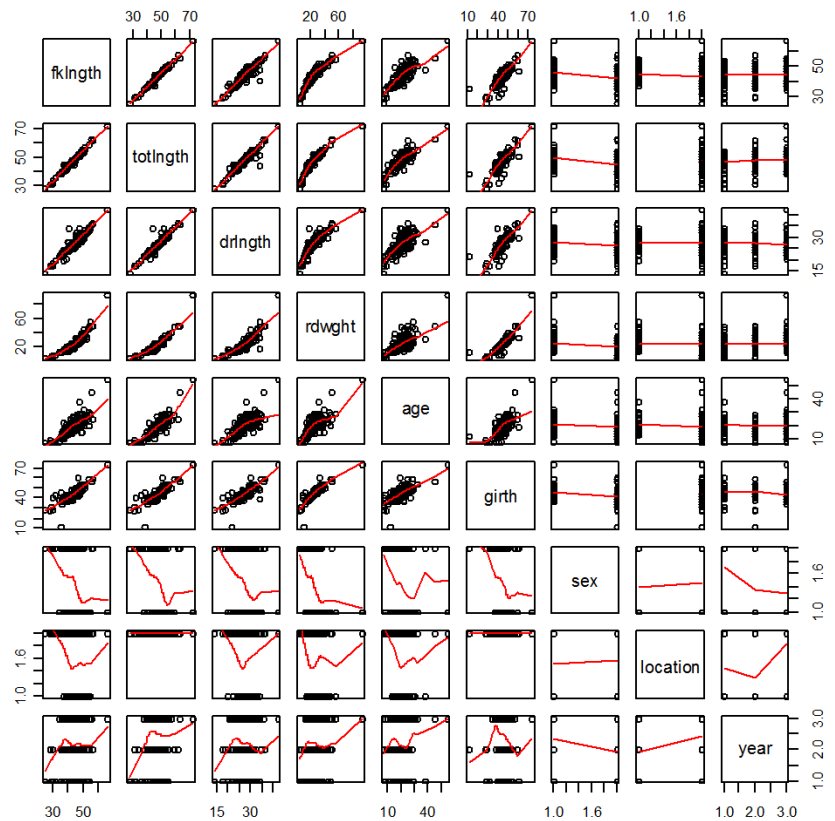
R has a function to create all pairwise scatterplots rapidly called `pairs()`. One of `pairs()` options is the addition of a lowess trace on each plot to that is a smoothed trend in the data.

To get the plot matrix with the lowess smooth for all variables in the sturgeon data frame, execute the command

```
pairs(sturgeon, panel=panel.smooth)
```

and you should get:

Figure 9.



Creating data subsets

You will frequently want to do analyses on some subset of your data..

The command `subset()` is what you need to isolate cases meeting some criteria. For example, to create a subset of the sturgeon data frame that contains only females caught in 1978, you could write: :

```
> sturgeon.female.1978<-subset(sturgeon, sex=="FEMALE" &  
year=="1978")  
> sturgeon.female.1978
```

```
      fklngth totlngth drlngth rdwght age girth  
2    50.19685 54.13386 31.49606   NA  24  53.5  
4    50.19685 53.14961 32.28346   NA  23  52.5
```

```

6  49.60630 53.93701 31.10236 35.86 23 54.2
7  47.71654 51.37795 33.97638 33.88 20 48.0
15 48.89764 53.93701 29.92126 35.86 23 52.5
105 46.85039      NA 28.34646 23.90 24  NA
106 40.74803      NA 24.80315 17.50 18  NA
107 40.35433      NA 25.59055 20.90 21  NA
109 43.30709      NA 27.95276 24.10 19  NA
113 53.54331      NA 33.85827 48.90 20  NA
114 51.77165      NA 31.49606 35.30 26  NA
116 45.27559      NA 26.57480 23.70 24  NA
118 53.14961      NA 32.67717 45.30 25  NA
119 50.19685      NA 32.08661 33.90 26  NA
123 49.01575      NA 29.13386 37.50 22  NA
sex location year
2  FEMALE  THE_PAS 1978
4  FEMALE  THE_PAS 1978
6  FEMALE  THE_PAS 1978
7  FEMALE  THE_PAS 1978
15 FEMALE  THE_PAS 1978
105 FEMALE CUMBERLAND 1978
106 FEMALE CUMBERLAND 1978
107 FEMALE CUMBERLAND 1978
109 FEMALE CUMBERLAND 1978
113 FEMALE CUMBERLAND 1978
114 FEMALE CUMBERLAND 1978
116 FEMALE CUMBERLAND 1978
118 FEMALE CUMBERLAND 1978
119 FEMALE CUMBERLAND 1978
123 FEMALE CUMBERLAND 1978

```

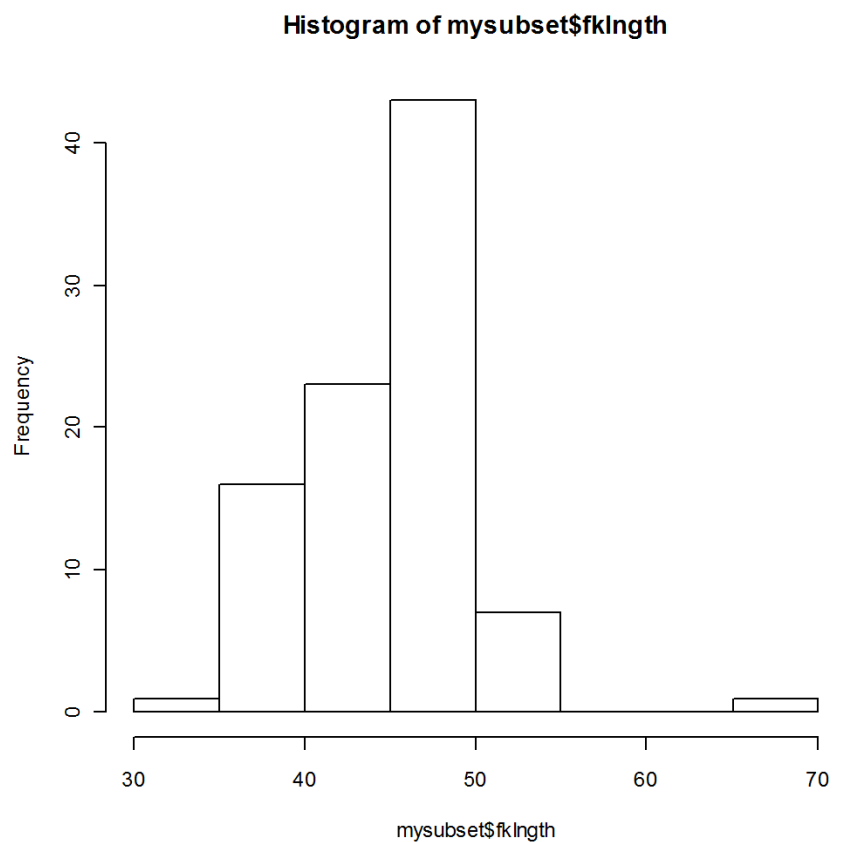
Trap: When using criteria to select cases, be careful of the `==` syntax to mean equal to. In this context, if you use a single `=`, you will not get what you want.

The following table lists the most common criteria to create expressions and their R syntax.

Operator	Meaning	Operator	Meaning
<code>==</code>	Equal to	<code>!=</code>	Not equal to
<code>></code>	Larger than	<code><</code>	Less than
<code>>=</code>	Larger than or equal	<code><=</code>	Less than or equal to
<code>&</code>	And	<code> </code>	Or

Using the commands `subset()` and `hist()`, create a histogram for females caught in 1979 and 1980 (hint: `sex=="FEMALE" & (year=="1979" | year=="1980")`)

Figure 10.



Data transformation

You will frequently transform raw data to better satisfy assumptions of statistical tests. R will allow you to do that easily..

The most used functions are probably:

`log10()`

`sqrt()`

`ifelse()`

You can use these functions directly within commands, create vector variables, or add columns in data frames.

To do a plot of the decimal log of `fklnth` vs `age`, you can simply use the `log10` function within the plot command:

```
plot(log10(fklnth)~age)
```

To create a vector variable, an orphan variable if you wish, one that is not part of a data frame, called `lfklnth` and corresponding too the decimal log of `fklnth`, simply enter:

```
1fklnth<-log10(fklnth)
```

If you want this new variable to be added to a data frame, then you must prefix the variable name by the data frame name and the \$ symbol. For example to add the variable `1fk1` containing the decimal log of `fklnth` to the `sturgeon` data frame, enter::

```
sturgeon$1fk1<-log10(fklnth)
```

`1fk1` will be added to the data frame `sturgeon` for the R session. Do not forget to save the modified data frame if you want to keep the modified version.

For conditional transformations, you can use the function `ifelse()`. For example, to create a new variable called `dummy` with a value of 1 for males and 0 for females, you can use:

```
:
```

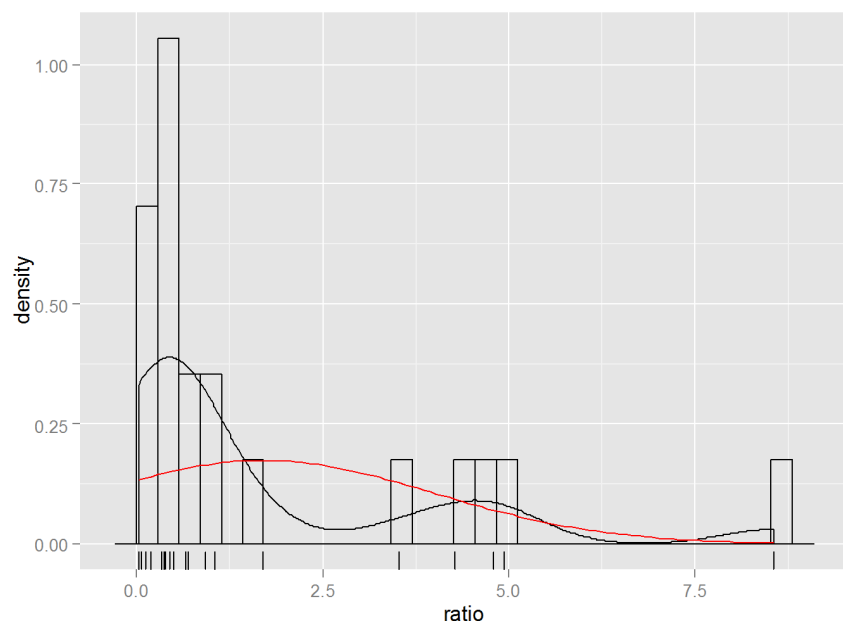
```
dummy<-ifelse(sex=="MALE",1,0)
```

Exercise

The file `salmonella.Rdata` contains numerical values for the variable called `ratio` for two environments (milieux: IN VITRO or IN VIVO) and for 3 strains (souches).

Examine the `ratio` variable and make a graph to visually assess normality for the SAUVAGE strain.

Figure 11.



Lab - Power analysis with G*Power

After completing this laboratory, you should

- be able to compute the power of a t-test with G*Power
- be able to calculate the required sample size to achieve a desired power level with a t-test
- be able to calculate the detectable effect size by a t-test given the sample size, the power and α
- understand how power changes when sample size increases, the effect size changes, or when α decreases
- understand how power is affected when you change from a two-tailed to a one-tailed test.

The theory

What is power?

Power is the probability of rejecting the null hypothesis when it is false.

Why do power analyses?

Assess strength of the evidence. Power analysis, performed after accepting a null hypothesis, can help assess the probability of rejecting the null if it were false, and if the magnitude of the effect was equal to that observed (or to any other given magnitude). This type of *a posteriori* analysis is very common.

Design better experiments. Power analysis, performed prior to conducting an experiment (but most often *after* a preliminary experiment), can be used to determine the number of observations required to detect an effect of a given magnitude with some probability (the power). This type of *a priori* experiment should be more common.

Estimate minimum detectable effect. Sampling effort is often pre-determined (when you are handed data of an experiment already completed), or extremely constrained (when logistics dictates what can be done). Whether it is *a priori* or *a posteriori*, power analysis can help you estimate, for a fixed sample size and a given power, what is the minimum effect size that can be detected.

Factors affecting power

For a given statistical test, there are 3 factors that affect power:

Decision criteria. Power is related to α , the probability level at which one rejects the null hypothesis. If this decision criteria is made very strict (i.e. if critical α is set to a very low value, like 0.1% or $p=0.001$), then power will be lower than if the critical α was less strict.


Sample size. The larger the sample size, the larger the power. As sample size increases, one's ability to detect small effect sizes as being statistically significant gets better.

Effect size. The larger the effect size, the larger the power. For a given sample size, the ability to detect an effect as being significant is higher for large effects than for small ones. Effect size measures how false the null hypothesis is.

What is G*Power?

G*Power is free software developed by quantitative psychologists from the University of Dusseldorf in Germany that is available at: <http://www.gpower.hhu.de/en.html>. It is available in MacOS and Windows versions.

G*Power will allow you to do power analyses for the majority of statistical tests we will cover during the term without making lengthy calculations and looking up long tables and figures of power curves. It is a really useful tool that you need to master.

 Download the software and install it on your computer and your workstation (if it is not there already).

How to use G*Power

General principle

Using G*Power generally involves 3 steps:

1. Choosing the appropriate test
2. Choosing one of the 5 types of available power analyses
3. Enter parameter values and press the **Calculate** button

Types of power analyses

A priori. Computes the sample size required given β , α , and the effect size. This type of analysis is useful when planning experiments.

Compromise. Computes α and β for a given α/β ratio, sample size, and effect size. Less commonly used (I have never used it myself) although it can be useful when the α/β ratio has meaning, for example when the cost of type I and type II errors can be quantified.

Criterion. Computes α for a given β , sample size, and effect size. In practice, I see little interest in this. Let me know if you see something I don't!

Post-hoc. Computes the power for a given α , effect size, and sample size. Used frequently to help in the interpretation of a test that is not statistically significant, but only if an effect size that is biologically significant is used (and not the observed effect size). Not relevant when the test is significant.

Sensitivity. Computes the detectable effect size for a given β , α , and sample size. Very useful at the planning stage of an experiment.

How to calculate effect size

G*Power can perform power analyses for several statistical tests. The metric for effect size depends on the test. Note that other software packages often use different effect size metrics and that it is important to use the correct one for each package. G*Power has an effect size calculator for many tests that only requires you to enter the relevant values. The following table lists the effect size metrics used by

G*Power for the various tests.

Test	Index of effect size	Calculation
t-test on Means	d	$d = \frac{ \mu_1 - \mu_2 }{\sqrt{(s_1^2 + s_2^2)/2}}$
t-test on Correlations	r	
Other t-tests	f	$f = \mu_1/\sigma$

F-test (ANOVA)

f

$$f = \frac{\sqrt{\frac{\sum_{i=1}^k (\mu_i - \mu)^2}{k}}}{\sigma}$$

Other F-tests

f²

$$f^2 = \frac{R_p^2}{1 - R_p^2}$$

where R_p^2 is the squared partial correlation coefficient

Chi-Square test

w

$$w = \sqrt{\frac{\sum_{i=1}^m (p_{0i} - p_{1i})^2}{p_{0i}}}$$

where p_{0i} is the proportion in category i predicted by the null (0) and alternative (1) hypothesis

Power analysis for a t-test on two independent means

The objective of this lab is to learn to use G*Power and understand how the 4 parameters of power analyses (α , β , sample size and effect size) are related to each other. For this, you will only use the standard t-test to compare two independent means. This is the test most used by biologists, you have all used it, and it will serve admirably for this lab. What you will learn today will be applicable to all other power analyses.

Jaynie Stephenson studied the productivity of streams in the Ottawa region. She has measured fish biomass in 36 streams, 18 on the Shield and 18 in the Ottawa Valley. She found that fish biomass was lower in streams from the valley (2.64 g/m², standard deviation=3.28) than from the Shield (3.31 g/m², standard deviation=2.79.). When she tested the null hypothesis that fish biomass is the same in the two regions by a t-test, she obtained:

Pool ed- Vari ance Two- Sampl e t - Test

t = -0.5746, df = 34, p- val ue = 0.5693

She therefore accepted the null hypothesis (since p is much larger than 0.05) and concluded that fish biomass is the same in the two regions.

Post-hoc analysis

Using the observed means and standard deviations, we can use G*Power to calculate the power of the two-tailed t-test for two independent means, using the observed effect size (the difference between the two means, weighted by the standard deviations) for $\alpha=0.05$.

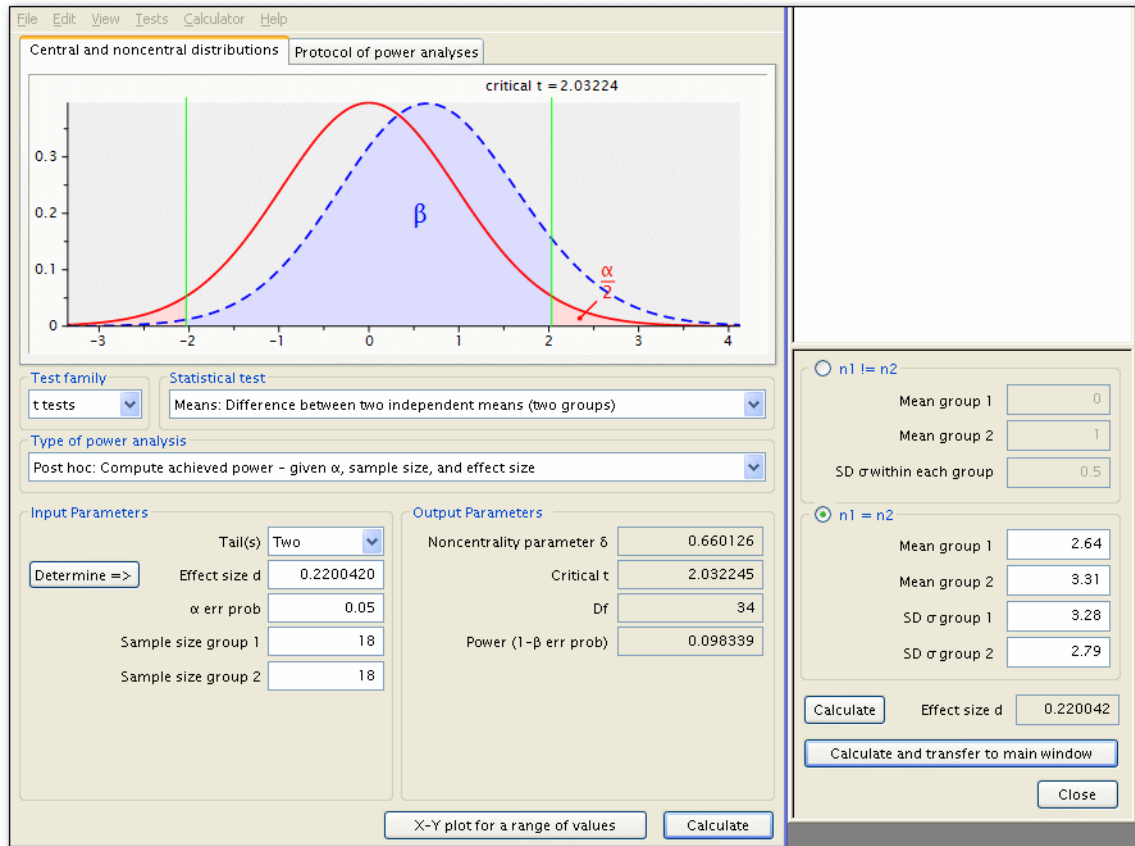
 Start G*Power.

1. In **Test family**, choose: t tests
2. For **Statistical test**, choose: Means: Difference between two independent means (two groups)
3. For **Type of power analysis**, choose: Post hoc: Compute achieved power - given α , sample size, and effect size
4. At **Input Parameters**,
in the box **Tail(s)**, chose: Two,
check that **α err prob** is equal to 0.05
Enter 18 for the **Sample size** of group 1 and of group 2

then, to calculate effect size (d), click on **Determine =>**
5. In the window that opens,

select **n1 = n2**, then
enter the two means(**Mean group 1 et 2**)
the two standard deviations(**SD s group 1 et 2**)
and click on **Calculate and transfer to main window**
6. After you click on the Calculate button in the main window, you should get the following:

Figure 1.



Let's examine this plot.

- The curve on the left, in red, corresponds to the expected distribution of the t-statistics when H_0 is true (i.e. when the two means are equal) given the sample size (18 per region) and the observed standard deviations.
- The vertical green lines correspond to the critical values of t for $\alpha = 0.05$ and a total sample size of 36 (2×18).
- The shaded pink regions correspond to the rejection zones for H_0 . If Jaynie had obtained a t-value outside the interval delimited by the critical values ranging from -2.03224 to 2.03224, she would then have rejected H_0 , the null hypothesis of equal means. In fact, she obtained a t-value of -0.5746 and concluded that the biomass is equal in the two regions.
- The curve on the right, in blue, corresponds to the expected distribution of the t-statistics if H_1 is true (here H_1 is that there is a difference in biomass between the two regions equal to $3.33 - 2.64 = 0.69 \text{ g/m}^2$, given the observed standard deviations). This distribution is what we should observe if H_1 was true and we repeated a large number of times the experiment using random

samples of 18 streams in each of the two regions and calculated a t-statistic for each sample. On average, we would obtain a t-statistic of about 0.6.

- Note that there is considerable overlap of the two distributions and that a large fraction of the surface under the right curve is within the interval where H_0 is accepted between the two vertical green lines at -2.03224 and 2.03224. This proportion, shaded in blue under the distribution on the right is labeled β and corresponds to the risk of type II error (accept H_0 when H_1 is true).
- Power is simply $1-\beta$, and is here 0.098339. Therefore, if the mean biomass differed by $0.69\text{g}/\text{m}^2$ between the two regions, Jaynie had only 9.8% chance of being able to detect it as a statistically significant difference at $\alpha=5\%$ with a sample size of 18 streams in each region.

Let's recapitulate: The difference in biomass between regions is not statistically significant according to the t-test. It is because the difference is relatively small relative to the precision of the measurements. It is therefore not surprising that that power, i.e. the probability of detecting a statistically significant difference, is small. Therefore, this analysis is not very informative.

Indeed, a post hoc power analysis using the observed effect size is not useful. It is much more informative to conduct a post hoc power analysis for an effect size that is **different** from the observed effect size. But what effect size to use? It is the biology of the system under study that will guide you. For example, with respect to fish biomass in streams, one could argue that a two fold change in biomass (say from 2.64 to $5.28\text{g}/\text{m}^2$) has ecologically significant repercussions. We would therefore want to know if Jaynie had a good chance of detecting a difference as large as this before accepting her conclusion that the biomass is the same in the two regions. So, what were the odds that Jaynie could detect a difference of $2.64\text{g}/\text{m}^2$ between the two regions? G*Power can tell you if you cajole it the right way.


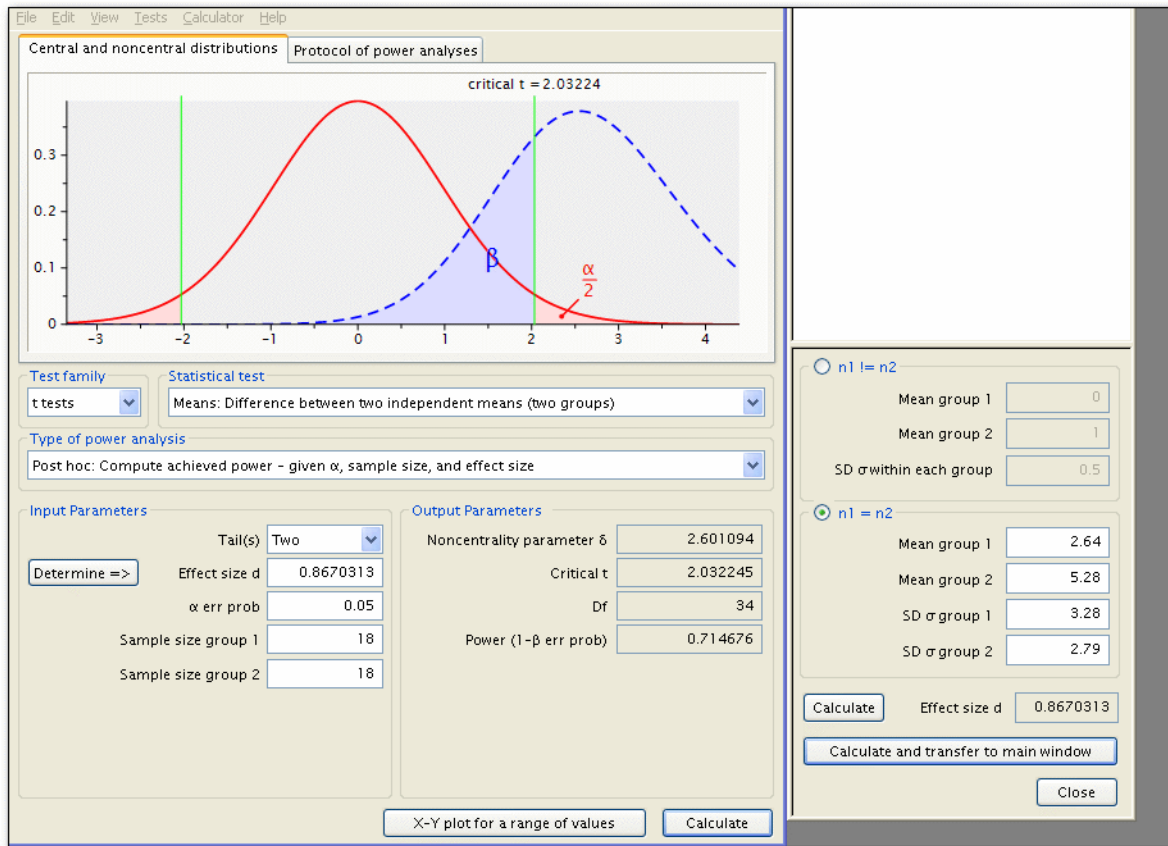
 Change the mean of group 2 to 5.28, recalculate effect size, and click on Calculate to obtain:

Figure 2.



The power is 0.71, therefore Jaynie had a reasonable chance (71%) of detecting a doubling of biomass with 18 streams in each region.

Note that this post hoc power analysis, done for an effect size considered biologically meaningful, is much more informative than the preceding one done with the observed effect size (which is what too many students do because it is the default of so many power calculation programs). Jaynie did not detect a difference between the two regions. There are two possibilities: 1) there is really no difference between the regions, or 2) the precision of measurements is so low (because the sample size is small and/or there is large variability within a region) that it is very unlikely to be able to detect even large differences. The second power analysis can eliminate this second possibility because Jaynie had 71% chances of detecting a doubling of biomass.

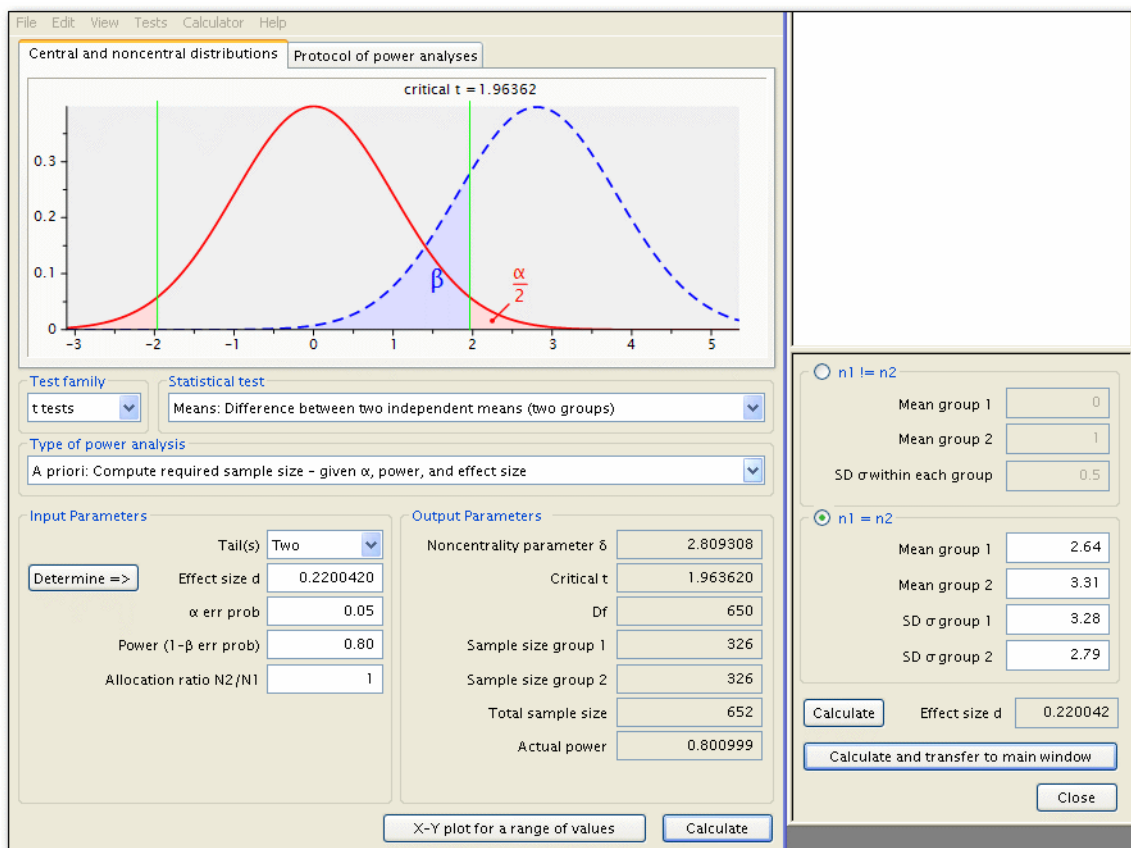
A priori analysis

Suppose that a difference in biomass of $3.31 - 2.64 = 0.67 \text{g/m}^2$ can be ecologically significant. The next field season should be planned so that Jaynie would have a good chance of detecting such a difference in

fish biomass between regions. How many streams should Jaynie sample in each region to have 80% of detecting such a difference (given the observed variability)?

☞ Change the type of power analysis in G*Power to **A priori: Compute sample size - given α , power, and effect size**. Ensure that the values for means and standard deviations are those obtained by Jaynie. Recalculate the effect size metric and enter 0.8 for power and you will obtain:

Figure 3.



Ouch! The required sample would be of 326 streams in each region! It would cost a fortune and require several field teams otherwise only a few dozen streams could be sampled over the summer and it would be very unlikely that such a small difference in biomass could be detected. Sampling fewer streams would probably be in vain and could be considered as a waste of effort and time: why do the work on several dozens of streams if the odds of success are that low?

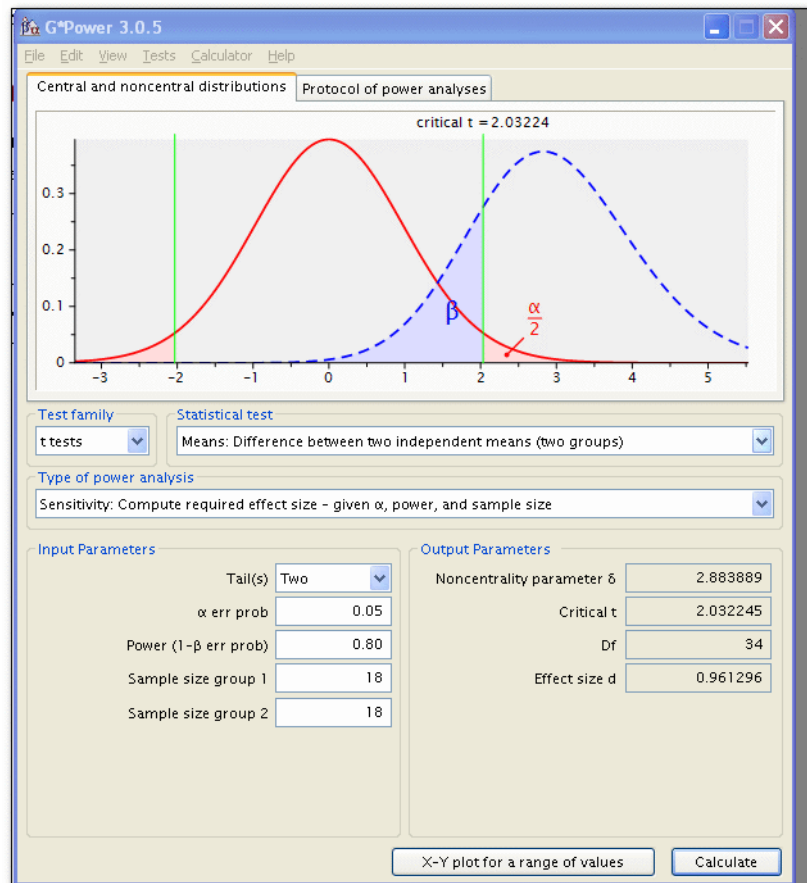
☞ If we recalculate for a power of 95%, we find that 538 streams would be required from each region. Increasing power means more work!

Sensitivity analysis - Calculate the detectable effect size

Given the observed variability, a sampling effort of 18 streams per region, and with $\alpha=0.05$, what effect size could Jaynie detect with 80% probability ($\beta=0.2$)?

☞ Change analysis type in G*Power to **Sensitivity: Compute required effect size - given α , power, and sample size** and check that sample size is 18 in each region. You should get:

Figure 4.



The detectable effect size for this sample size, $\alpha=0.05$ and $\beta=0.2$ (or power of 80%) is 0.961296. **Attention**, this effect size is the metric d and is dependent on sampling variability. Here, d is approximately equal to

$$d = \frac{|\bar{X}_1 - \bar{X}_2|}{\sqrt{\frac{s_1^2 + s_2^2}{2}}}$$

To convert this d value without units into a value for the detectable difference in biomass between the two regions, you need to multiply d by the denominator of the equation:

$$\begin{aligned} \text{detectable difference} &= |\bar{X}_1 - \bar{X}_2| = d \sqrt{\frac{s_1^2 + s_2^2}{2}} \\ \text{detectable difference} &= 0.961296 \sqrt{\frac{3.28^2 + 2.79^2}{2}} \\ \text{detectable difference} &= 2.927024 \end{aligned}$$

Therefore, with 18 streams per region, $\alpha=0.05$ and $\beta=0.2$ (so power of 80%), Jaynie could detect a difference of 2.93 g/m² between regions, a bit more than a doubling of biomass.

Important points to remember

- Post hoc power analyses are relevant only when the null hypothesis is accepted because it is impossible to make a type II error when rejecting H_0 .
- With very large samples, power is very high and minute differences can be statistically detected, even if they are not biologically significant.
- When using a stricter significance criteria ($\alpha < 0.05$) power is reduced.
- Maximizing power implies more sampling effort, unless you use a more liberal statistical criteria ($\alpha > 0.05$)
- The choice of β is somewhat arbitrary. $\beta = 0.2$ (power of 80%) is considered relatively high by most.

Assignment

Black fly larvae (Diptera: Simuliidae) were sampled in February at the outlet of two lakes of the Eastern Townships (Lakes Orford and Lovering). The body length of each larva was measured. The data are in the file `simulies.orford.lovering.Rdata`

The relationship between body length (L, in mm) and dry mass (M, in μg) for the dominant species (*P. mixtum/fuscum*) is:

$$M = 1.36L^{3.05}$$

1. Calculate the mean mass and its standard deviation at each site.
2. Using the mean mass of *P. mixtum/fuscum* at Lovering as a reference and the standard deviations observed at each site, calculate the power of a two-tailed t test for independent means
 - a) if the mass difference was 5 μg , $\alpha=0.05$, and that 100 larvae were sampled at each site
 - b) if the mass difference was 20 μg , $\alpha=0.05$, and that 100 larvae were sampled at each site
 - c) if the mass difference was 50 μg , $\alpha=0.05$, and that 100 larvae were sampled at each site
 - d) comment on how changing the mass difference affects power.
3. Calculate the required sample size to detect, with a two-tailed t-test for independent means, a difference of 50 μg between the means
 - a) with a power of 80% and $\alpha=0.05$
 - b) with a power of 80% and $\alpha=0.001$
 - c) with a power of 95% and $\alpha=0.05$
 - d) comment on how changing power and α affects the required sample size.
4. Calculate the detectable effect size (d) with a two-tailed t-test on independent means, given the observed standard deviations
 - a) with a power of 80%, $\alpha=0.05$, and measurements on 10 larvae at each site
 - b) with a power of 80%, $\alpha=0.05$, and measurements on 200 larvae at each site
 - c) with a power of 80%, $\alpha=0.05$, and measurements on 20 larvae at one site and 380 larvae at the second site
 - d) comment on how changing sample sizes affect d.
5. Calculate the difference in average mass, in μg , that can be detected given your estimate of the minimum detectable effect size (d) at 4a, b, and c.

Lab- Linear correlation and simple linear regression

After completing this laboratory exercise, you should be able to:


- Use R to produce a scatter plot of the relationship between two variables.
- Use R to carry out some simple data transformations.
- Use R to compute the Pearson product-moment correlation between two variables and assess its statistical significance.
- Use R to compute the correlation between pairs of ranked variables using the Spearman rank correlation and Kendall's tau.
- Use R to assess the significance of pairwise comparisons from a generalized correlation matrix using Bonferroni-adjusted probabilities.
- Use R do a simple linear regression
- Use R to test the validity of the assumptions underlying simple linear regression
- Use R to assess significance of a regression by the bootstrap method
- Quantify effect size in simple regression and perform a power analysis using G*Power

Scatter plots

Correlation and regression analysis should always begin with an examination of the data: this is a critical first step in determining whether such analyses are even appropriate for your data.

Suppose we are interested in the extent to which length of male sturgeon in the vicinity of The Pas and Cumberland House covaries with weight. To address this question, we look at the correlation between `fklength` and `rdwght`.

Recall that one of the assumptions in correlation analysis is that the relationship between the two variables is linear. To evaluate this assumption, a good first step is to produce a scatterplot:

 Using the file `sturgeon.Rdata`, load the sturgeon data frame and make a scatter plot of `rdwght` vs `fklength` fit with a locally weighted regression (Loess) smoother, and a linear regression

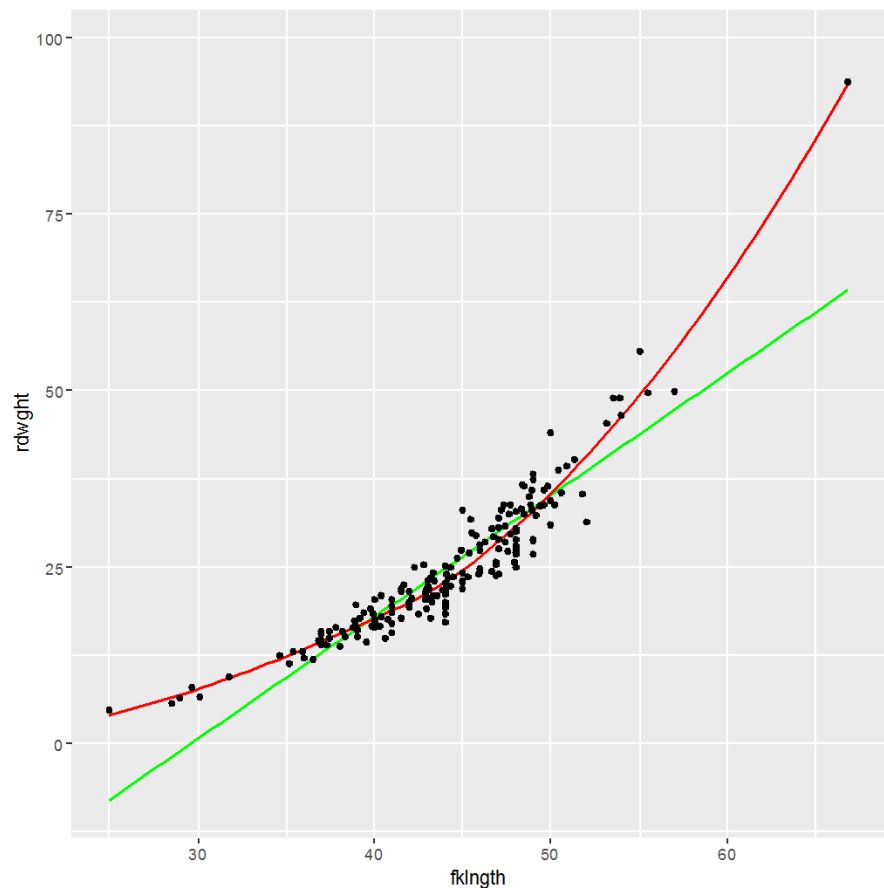
```

mygraph<-ggplot(
  data=sturgeon, # source of data
  aes(x=fklength, y=rdwght)) #aesthetics: x=fklength, y=rdwght
# plot data points, regression, loess trace
mygraph<-mygraph+
  stat_smooth(method=lm, se=FALSE, color="green") + # add
linear regression, but no SE shading
  stat_smooth(color="red")+ #add loess
  geom_point() # add data points
mygraph # display graph

```

Does this curve suggest a good correlation between the two? Based on visual inspection, does the relationship between these two variables appear linear?

Figure 1.

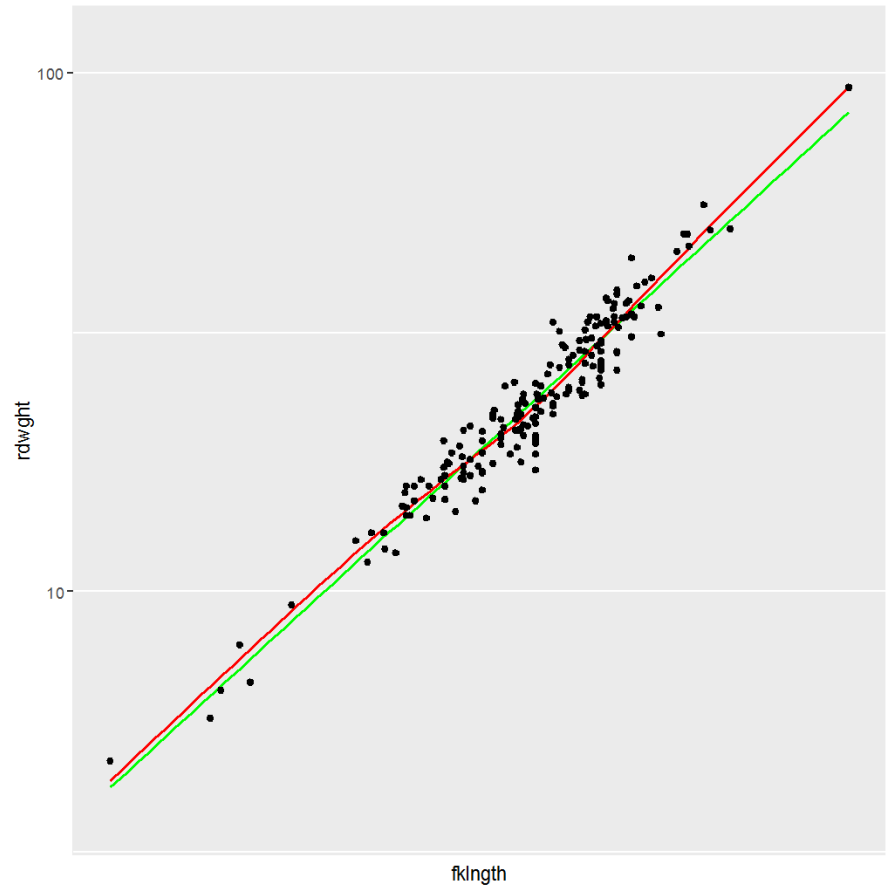


There is some evidence of nonlinearity, as the curve appears to have a positive second derivative (concave up). This notwithstanding, it does appear the two variables are highly correlated.

 Redo the scatterplot, but after logtransformation of both axes

```
# apply log transformation on defined graph
mygraph+scale_x_log10()+scale_y_log10()
```

Figure 2.



Compare the diagrams before and after the transformation. Since the relationship is more linear after transformation, correlation analysis should be done on the transformed data.

Data transformations and the product-moment correlation

Recall that another assumption underlying significance testing of the product-moment correlation is that the distribution of the two variables in question is bivariate normal. We can test to see whether each of the two variables are normally distributed using the same procedures outlined in the exercise on two-sample comparisons. If the

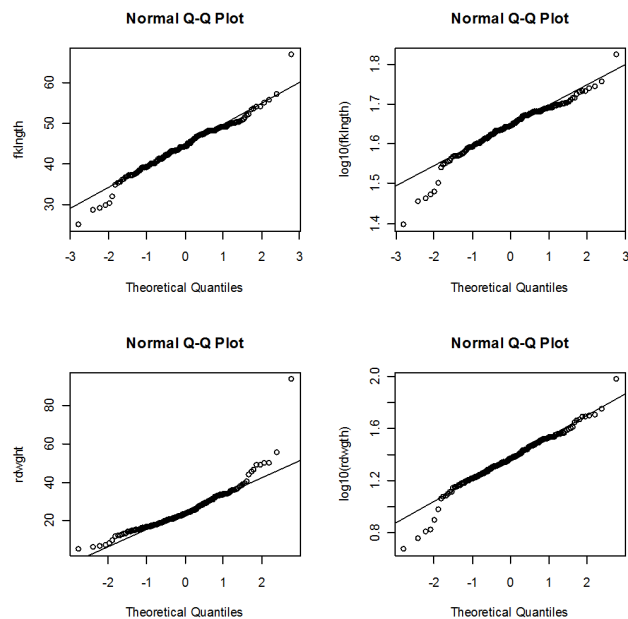
two variables are each normally distributed, then one is usually (relatively) safe in assuming the joint distribution is normal, although this needn't necessarily be true.

☞ Examine the distribution of the 4 variables (the two original variables and the log-transformed variables). What do you conclude from visual inspection of these plots?

The following graph contains the 4 QQ plots. It was produced by the code below that starts with the `par()` command to ensure that all 4 plots would appear together on the same page in 2 rows and 2 columns :

```
attach(sturgeon)
par(mfrow = c(2, 2))
qqnorm(fklength, ylab="fklength")
qqline(fklength)
qqnorm(log10(fklength), ylab="log10(fklength)")
qqline(log10(fklength))
qqnorm(rdwght, ylab="rdwght")
qqline(rdwght)
qqnorm(log10(rdwght), ylab="log10(rdwght)")
qqline(log10(rdwght))
par(mfrow = c(1, 1))
```

Figure 3.



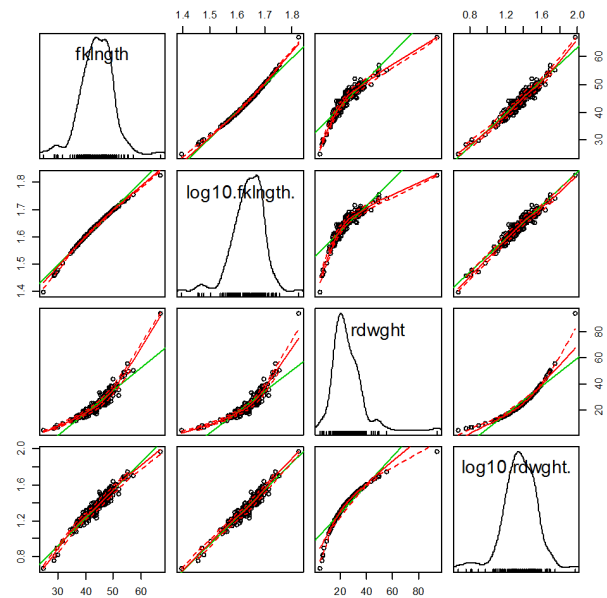
None of these distributions are perfectly normal, but deviations are mostly minor..

To generate a scatterplot matrix of all pairs of variables, with linear regression and loess traces, you can use `scatterplot.matrix()` (from the `car` library which you will have to install first):

```
scatterplot.matrix(~fklngth+log10(fklngth)+rdwght+log10(rdwght), reg.line=lm, smooth=TRUE, span=0.5, diagonal = 'density')
```

to get

Figure 4.



Next, calculate the Pearson product-moment correlation between each pair (untransformed and log transformed) using the `cor()` command. However, to do this, it will be easier if you first add your transformed data as columns in the `sturgeon` data frame:

```
lflnth<-log10(fklngth)
lrdwght<-log10(rdwght)
```

Then you can get the correlation matrix by:

```
cor(sturgeon[,c("fklngth", "lflnth", "lrdwght", "rdwght")],
    use="complete.obs")
```

Note the `use="complete.obs"` parameter. It tells R to keep only lines of the data frame where all variables were measured. If there are missing data, some lines will be removed, but correlations will be calculated for the same subset of cases for all pairs of variables. One could use, instead, `use="pairwise.complete.obs"`, to tell R to only eliminate observations when values are missing for this particular pair

of variables. In this situation, if there are missing values in the data frame, the sample size for pairwise correlations will vary. In general, I recommend you use the option **use="complete.obs"**, unless you have so many missing values that it eliminates the majority of your data.

Why is the correlation between the untransformed variables smaller than between the transformed variables?

```


      f k l n g t h   l f k l n g t h   l r d w g h t   r d w g h t
f k l n g t h  1. 0000000  0.9921435  0.9645108  0.9175435
l f k l n g t h 0.9921435  1.0000000  0.9670139  0.8756203
l r d w g h t  0.9645108  0.9670139  1.0000000  0.9265513
r d w g h t    0.9175435  0.8756203  0.9265513  1.0000000

```

Several things should be noted here. First, the correlation between fork length and round weight is high, regardless of which variables are used: so as might be expected, heavier fish are also longer, and vice versa. Second, the correlation is greater for the transformed variables than the untransformed variables. Why? Because the correlation coefficient is inversely proportional to the amount of scatter around a straight line. If the relationship is curvilinear (as it is for the untransformed data), the scatter around a straight line will be greater than if the relationship is linear. Hence, the correlation coefficient will be smaller.

Testing the significance of correlations and Bonferroni probabilities

It's possible to test the significance of individual correlations using the commands window. As an example, let's try testing the significance of the correlation between `lfklength` and `rdwght` (the smallest correlation in the above table).

 In the R script window, enter the following to test the correlation between `lfkfl` and `rdwght`:

```
>cor.test(sturgeon$lfklength, sturgeon$rdwght, alternative="two.sided", method="pearson")
```

```
Pearson's product-moment correlation
```

```

data:  sturgeon$lfklength and sturgeon$rdwght
t = 24.3223, df = 180, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8367345 0.9057199
sample estimates:
      cor
0.8756203

```


We see here that the correlation is highly significant ($p < 2.2e-16$), which is no surprise given how high the correlation coefficient is (0.8756).

It's important to bear in mind that when you are testing the probability of correlations, the probability of finding any one correlation that is "significant" by pure chance increases with the number of pairwise correlations examined. Suppose, for example, that you have five variables; there are then a total of 10 possible pairwise correlations, and from this set, you would probably not be surprised to find at least one that is "significant" purely by chance. One way of avoiding the problem is to adjust individual α levels for pairwise correlations by dividing by the number of comparisons, k , such that: $\alpha' = \alpha/k$ (Bonferroni probabilities), i.e. if initially, $\alpha = 0.05$ and there are a total of 10 comparisons, then $\alpha' = 0.005$.

In the above example where we examined correlations between `FKLNGTH` and `RDWGHT` and their log, it would be appropriate to adjust the α at which significance is tested by the total number of correlations in the matrix (in this case, 6, so $\alpha' = 0.0083$). Does your decision about the significance of the correlation between `lkgf1` and `rdwght` change?

Non-parametric correlations: Spearman's rank and Kendall's tau

The analysis done with the sturgeon data in the section above suggests that one of the assumptions of correlation, namely, bivariate normality, may not be valid for `fklngh` and `rdwght` nor for the log transforms of these variables. Finding an appropriate transformation is sometimes like looking for a needle in a haystack; indeed, it can be much worse simply because for some distributions, there is no transformation that will normalize the data. In such cases, the best option may be to go to a non-parametric analysis that does not assume bivariate normality or linearity. All such correlations are based on the ranks rather than the data themselves: two options available in R are Spearman and Kendall's tau.

 From the R console window, test the correlation between `fklngh` and `rdwght` using both the Spearman and Kendall's tau. The following commands will produce the correlations:

```
>cor.test(sturgeon$fklngh, sturgeon$rdwght, alternative="two.sided", method="spearman")
```

```
Spearman's rank correlation rho
```

```
data: sturgeon$fklngh and sturgeon$rdwght
S = 47971.33, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
```

```
0.9522546
```

```
>cor.test(sturgeon$fklngh, sturgeon$rdwght, alternative="two.sided", method="kendall")
```

```
Kendall's rank correlation tau
```

```
data: sturgeon$fklngh and sturgeon$rdwght
z = 16.3578, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.8208065
```

Contrast these results with those obtained using the Pearson product-moment correlation. Why the difference?

Test the non-parametric correlations on pairs of the transformed variables. You should immediately note that the non-parametric correlations are identical for untransformed and transformed variables. This is because we are using the ranks, rather than the raw data, and the rank ordering of the data does not change when a transformation is applied to the raw values.

Note that the correlations for Kendall's tau (0.820) are lower than for the Spearman rank (0.952) correlation. This is because Kendall's gives more weight to ranks that are far apart, whereas Spearman's weights each rank equally. Generally, Kendall's is more appropriate when there is more uncertainty about the reliability of close ranks.


The sturgeons in this sample were collected using nets and baited hooks of a certain size. What impact do you think this method of collection had on the shapes of the distributions of `fklngh` and `rdwght`? Under these circumstances, do you think correlation analysis is appropriate at all?

Note that correlation analysis assumes that *each variable is randomly sampled*. In the case of sturgeon, this is not the case: baited hooks and nets will only catch sturgeon above a certain minimum size. Note that in the sample, there are no small sturgeons, since the fishing gear targets only larger fish. Thus, we should be very wary of the correlation coefficients associated with our analysis, as the inclusion of smaller fish may well change our estimate of these correlations.

Simple linear regression

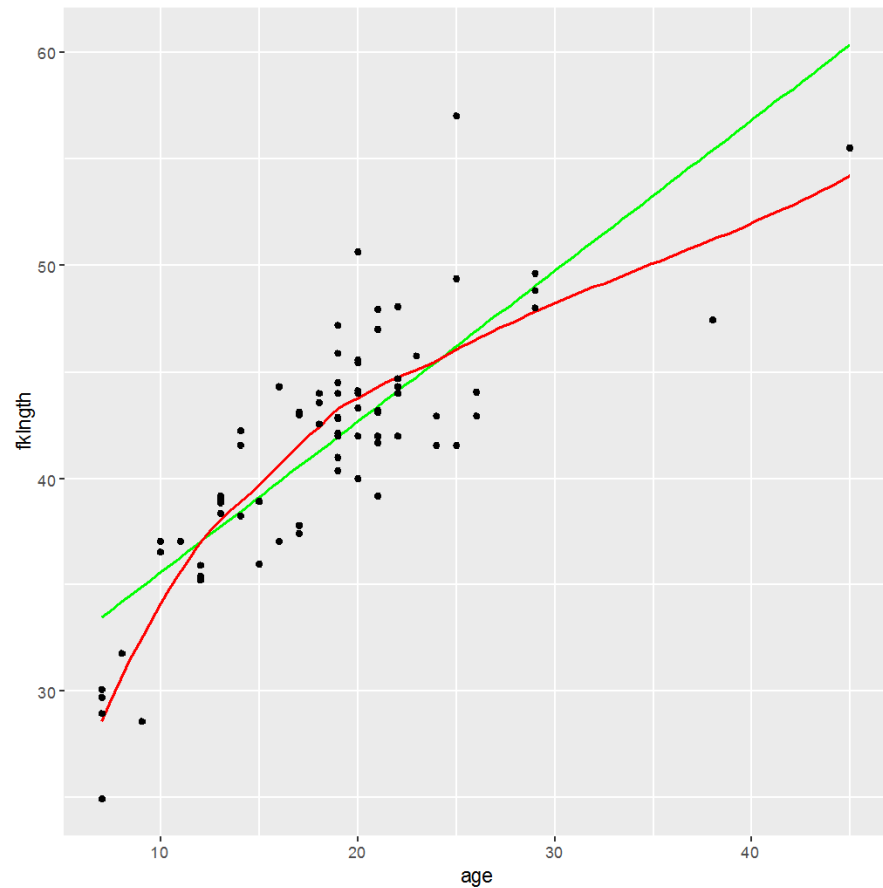
In correlation analysis we are interested in how pairs of variables covary: However, in regression analysis, we are attempting to estimate a model that predicts a variable (the dependent variable) from another variable (the independent variable)..

As with any statistical analysis, the best way to begin is by looking at your data. If you are interested in the relationship between two variables, say, Y and X, produce a plot of Y versus X just to get a "feel" for the relationship.

 The data file `sturgeon.Rdata` contains data for sturgeons collected from 1978-1980 at Cumberland House, Saskatchewan and The Pas, Manitoba. Make a scatterplot of `flngth` (the dependent variable) versus `age` (the independent variable) for males and add a linear regression and a loess smoother. What do you conclude from this plot?

```
sturgeon.male <- subset(sturgeon, subset=sex=="MALE")
mygraph<-ggplot(
  data=sturgeon.male, # source of data
  aes(x=age, y=flngth) #aesthetics: y=flngth, x=age
  # plot data points, regression, loess trace
mygraph<-mygraph+
  stat_smooth(method="lm", se=FALSE, color="green") + # add linear regression,
  but no SE shading
  stat_smooth(color="red") + #add loess
  geom_point() # add data points
mygraph # display graph
```

Figure 5.



This suggests that the relationship between age and fork length is not linear.

Suppose that we want to know the growth rate of male sturgeon. One estimate (perhaps not a very good one) of the growth rate is given by the slope of the fork length - age regression.

First, let's run the regression with the `lm()` command, and save its results in an object called `RegModel.1`

```
RegModel.1 <- lm(fklngh~age, data=sturgeon.male)
```

Nothing appears on the screen, but don't worry, it all got saved in memory.

To see the statistical results, type:

```
summary(RegModel.1)
```

Call :

```
lm(formula = fklngh ~ age, data = sturgeon.male) 1
```

Residuals: **2**

Mn	1Q	Median	3Q	Max
-8.4936	-2.2263	0.1849	1.7526	10.8234

```

Coefficient s: ③
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 28.50359    1.16873   24.39 <2e-16 ***
age         0.70724     0.05888   12.01 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: ④ 3.307 on 73 degrees of freedom
(5 observations deleted due to missingness)

Multiple R-squared: 0.664 ⑤, Adjusted R-squared: 0.6594 ⑥

F-statistic: 144.3 on 1 and 73 DF, p-value: < 2.2e-16 ⑦

```

- ① A friendly reminder of the model fitted and the data used.
- ② General statistics about the residuals around the fitted model
- ③ Fitted model parameter estimates, standard errors, t values and associated probabilities
- ④ Square root of the residual variance
- ⑤ Coefficient of determination. It corresponds to the proportion of the total variance of the dependent variable that is accounted for by the regression (i.e. by the independent variable).
- ⑥ The adjusted R-squared accounts for the number of parameters in the model. If you want to compare the performance of several models with different numbers of parameters, this is the one to use.
- ⑦ This is the test of the overall significance of the model. In the simple regression case, this is the same as the test of the slope of the regression.

The estimated regression equation is therefore:

$$\text{Fklength} = 28.50359 + 0.70724 * \text{age}$$

Given the highly significant F-value of the model (or equivalently the highly significant t-value for the slope of the line), we reject the null hypothesis that there is no relationship between fork length and age.

Testing regression assumptions

Simple model I regression makes four assumptions:

1. the X variable is measured without error;
2. the relationship between Y and X is linear;
3. that for any value of X, the Y's are independently and normally distributed;

4. the variance of Y for fixed X is independent of X.

Having done the regression, we can now test the assumptions. For most biological data, the first assumption is almost never valid; usually there is error in both Y and X. This means that in general, slope estimates are biased, but predicted values are unbiased. However, so long as the error in X is small relative to the range of X in your data, the fact that X has an associated error is not likely to influence the outcome dramatically. On the other hand, if there is substantial error in X, regression results based on a model I regression may give poor estimates of the functional relationship between Y and X. In this case, more sophisticated regression procedures must be employed which are, unfortunately, beyond the scope of this course.

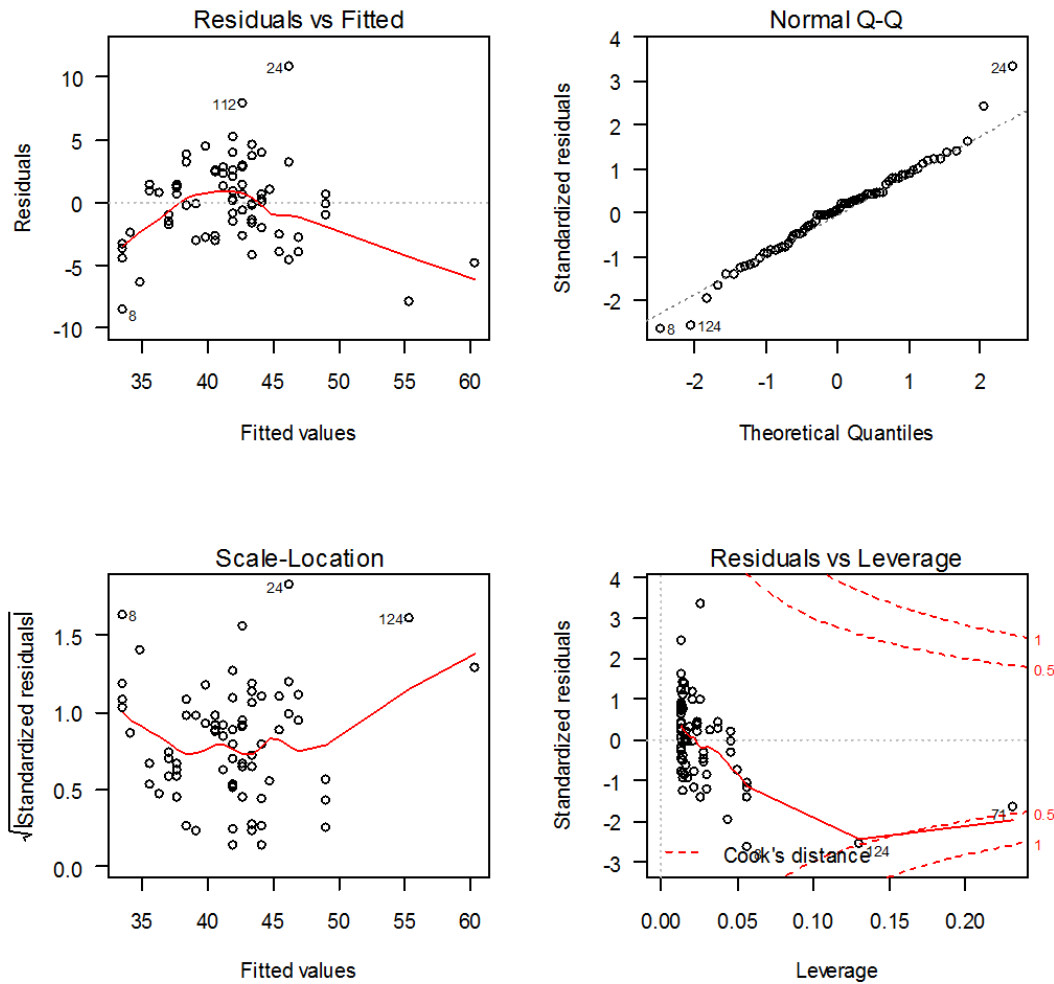
The other assumptions of a model I regression can, however, be tested, or at least evaluated visually. The `plot()` command can display diagnostics for linear models.

```
par(mfrow = c(2, 2), las=1)
plot(RegModel.1)
```

The `par()` command is used here to tell R to display 2 rows and 2 columns of graphs per page (there are 4 diagnostic graphs for linear models generated automatically), and the last statement is to tell R to rotate the labels of the Y axis so that they are perpendicular to the Y axis. (Yes, I know, this is not at all obvious.)

You will get:

Figure 6.



The first graph (upper left) can tell you about linearity, normality, and homoscedasticity of the residuals. It shows the deviations around the regression vs the predicted values. Remember that the scatterplot (*fork length vs age*) suggested that the relationship between fork length and age is not linear. Very young and very old sturgeons tended to fall under the line, and fish of average age tended to be a bit above the line. This is exactly what the residual vs fitted plot shows. The red line is a lowess trace through these data. If the relationship was linear, it would be approximately flat and close to 0. The scatter of residuals tells you a bit about their normality and homoscedasticity, although this graph is not the best way to look at these properties. The next two are better.

The second graph is to assess the normality of the residuals. It is a QQ plot of the residuals. If the residuals were normally distributed, they would fall very close to the diagonal line. Here, we see it is mostly the case, except in the tails.

The third graph, bottom left and titled Scale-Location, helps with assessing homoscedasticity. It plots the square root of the absolute value of the standardized residual (residual divided by the standard error of the residuals, this scales the residuals so that their variance is 1) as a function of the fitted value. This graph can help you visualize whether the spread of the residuals is constant or not. If residuals are homoscedastic, then the average will not change with increasing fitted values. Here, there is slight variability, but it is not monotonous (i.e. it does not increase or decrease systematically) and there is no strong evidence against the assumption of homoscedasticity.

The fourth graph, bottom right, plots the residuals as a function of leverage and can help detecting the presence of outliers or points that have a very strong influence on the regression results. The leverage of a point measures how far it is from the other points, but only with respect to the independent variable. In the case of simple linear regression, it is a function of the difference between the observation and the mean of the independent variable. You should look more closely at any observation with a leverage value that is greater than: $2(k+1)/n$, where k is the number of independent variables (here 1), and n is the number of observations. In this case there is 1 independent variable, 75 observations, and points with a leverage higher than 0.053 may warrant particular scrutiny. The plot also gives you information about how the removal of a point from the data set would change the predictions. This is measured by the Cook's distance, illustrated by the red lines on the plot. A data point with a Cook distance larger than 1 has a large influence.

Trap: Note that R automatically labels the most extreme cases on each of these 4 plots. It does not mean that these cases are outliers, or that you necessarily need be concerned with them. In any data set, there will always be a minimum and a maximum residual.

So, what is the verdict about the linear regression between `fklnngth` and `age`? It fails the linearity, possibly fails the normality, passes homoscedasticity, and this does not seem to be too strongly affected by some bizarre points.

Formal tests of regression assumptions

In my practice, I seldom use formal tests of regression assumptions and mostly rely on graphs of the residuals to guide my decisions. To my knowledge, this is what most biologists and data analysts do. However, in my early analyst life I was not always confident that I was interpreting these graphs correctly and wished that I had a formal test or a statistic quantifying the degree of deviation from the regression assumptions.

The `lmtest` R package, not part of the base R installation, but available from CRAN, contains a number of tests for linearity and homoscedasticity. And one can test for normality using the Shapiro-Wilk test seen previously.

☞ Install the `lmtest` package from CRAN.

☞ Run the following commands:

```
library(lmtest)
bptest(RegModel.1)
dwtest(RegModel.1)
resettest(RegModel.1)
shapiro.test(residuals(RegModel.1))
```

to get:

studentized Breusch-Pagan test **1**

```
data: RegModel.1
BP = 1.1765, df = 1, p-value = 0.2781
```

Durbin-Watson test **2**

```
data: RegModel.1
DW = 2.242, p-value = 0.849
alternative hypothesis: true autocorrelation is greater than 0
```

RESET test **3**

```
data: RegModel.1
RESET = 14.544, df1 = 2, df2 = 71, p-value = 5.082e-06
```

```
> shapiro.test(residuals(RegModel.1))
```

Shapiro-Wilk normality test **4**

```
data: residuals(RegModel.1)
W = 0.9804, p-value = 0.2961
```

- ❶ The Breusch-Pagan test examines whether the variability of the residuals is constant with respect to increasing fitted values. A low p value is indicative of heteroscedasticity. Here, the p value is high, and supports my visual assessment that the homoscedasticity assumption is met by these data.
- ❷ The Durbin-Watson test can detect serial autocorrelation in the residuals. Under the assumption of no autocorrelation, the D statistic is 2. This test can detect violation of independence of observations (residuals), although it is not foolproof. Here there is no problem identified.
- ❸ The RESET test is a test of the assumption of linearity. If the linearity assumption is met, the RESET statistic will be close to 1. Here, the statistic is much larger (14.54), and very highly significant. This confirms our visual assessment that the relationship is not linear.
- ❹ And the Shapiro-Wilk normality test on the residual confirms that the deviation from normality of the residuals is not large.

Data transformations in regression

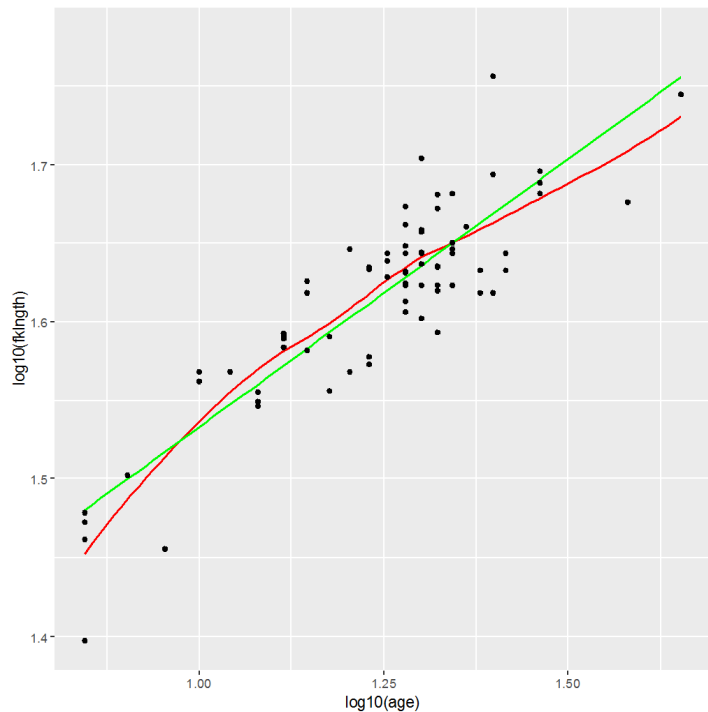
The analysis above revealed that the linearity assumption underlying regression analysis is not met by the `fklength-age` data. If we want to use regression analysis, data transformations are required:

☞ Let's plot the log-transformed data,

```
ggplot(data=sturgeon.male,
       aes(x=log10(age), y=log10(fklength)))+
  geom_smooth(color="red")+
  geom_smooth(method="lm", se=FALSE, color="green")+
  geom_point()
```

We get:

Figure 7.



and fit the linear regression model on the log-transformed variables:

```
RegModel.2 <- lm(log10(fklength)~log10(age), data=sturgeon.male)
summary(RegModel.2)
```

```
Call:
lm(formula = log10(fklength) ~ log10(age), data = sturgeon.male)

Residuals:
    Min       1Q   Median       3Q      Max
-0.0827935 -0.0168373 -0.0007188  0.0211016  0.0874465

Coefficients:
(Intercept)  1.19199   0.02723   43.77  <2e-16 ***
log10(age)    0.34086   0.02168   15.72  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

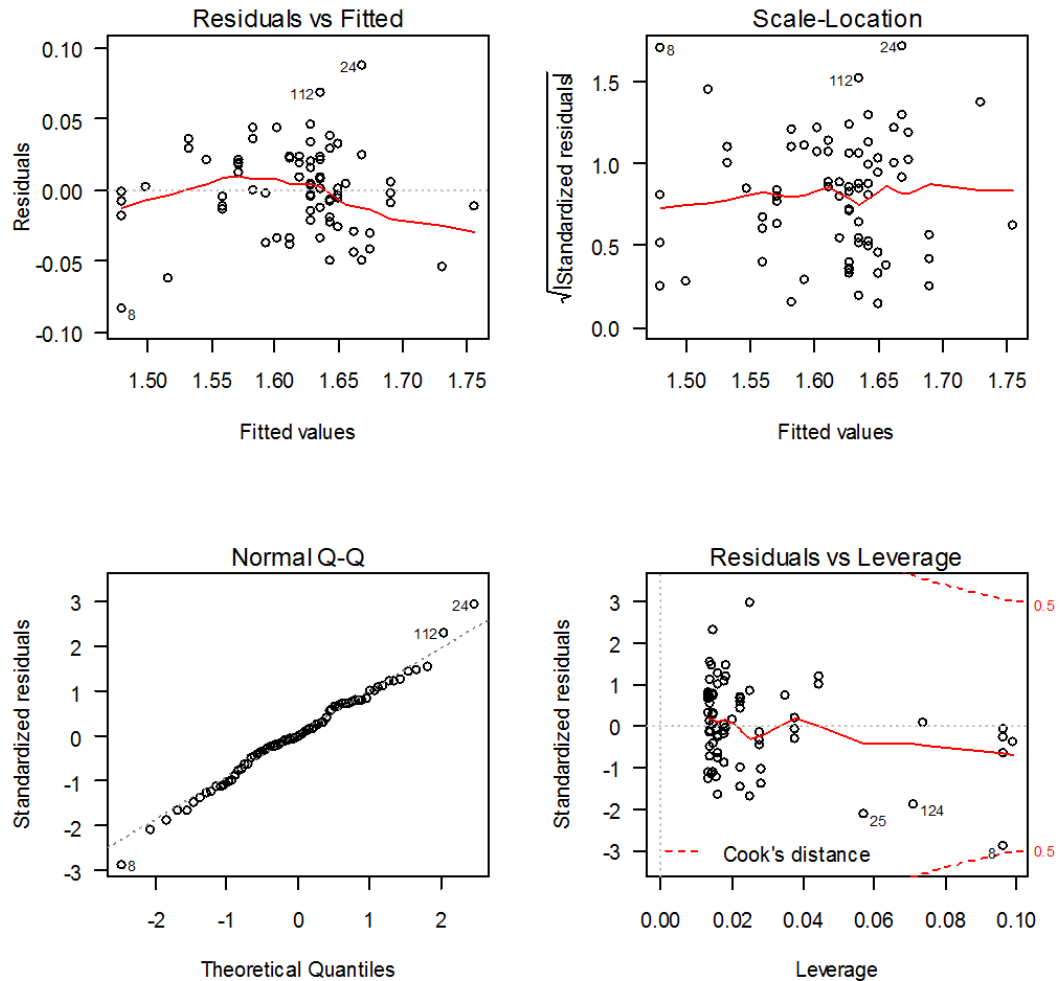
Residual standard error: 0.03015 on 73 degrees of freedom
(5 observations deleted due to missingness)
Multiple R-squared:  0.772,    Adjusted R-squared:  0.7688
F-statistic: 247.1 on 1 and 73 DF,  p-value: < 2.2e-16
```

Note that by using the log transformed data, the proportion of variation explained by the regression has increased by 10% (from 0.664 to 0.772), a substantial increase.

So the relationship has become more linear. Good. Let's look at the residual diagnostic plots:

```
plot(RegModel.2)
```

Figure 8.



So things appear a little better than before, although still not ideal. For example, the Residual vs fitted plot still suggests a potential nonlinearity. The QQ plot is nicer than before, indicating that residuals are more normally distributed after the log-log transformation. There is no indication of heteroscedasticity. And, although there are still a few points with somewhat high leverage, none have a Cook's distance above 0.5. It thus seems that transforming data improved things: more linear, more normal, less dependence on extreme data.

Do the formal tests support this visual assessment?

```

bptest (RegModel . 2)
dwtest (RegModel . 2)
resettest (RegModel . 2)
shapiro.test (residuals (RegModel . 2))

> bptest (RegModel . 2)

      studentized Breusch-Pagan test

data:  RegModel . 2
BP = 0.14282, df = 1, p-value = 0.7055

> dwtest (RegModel . 2)

      Durbin-Watson test

data:  RegModel . 2
DW = 2.1777, p-value = 0.6134
alternative hypothesis: true autocorrelation is greater than 0

> resettest (RegModel . 2)

      RESET test

data:  RegModel . 2
RESET = 4.4413, df1 = 2, df2 = 71, p-value = 0.01523

> shapiro.test (residuals (RegModel . 2))

      Shapiro-Wilk normality test

data:  residuals (RegModel . 2)
W = 0.98998, p-value = 0.8246

```

Indeed, they do: residuals are still homoscedastic (Breusch-Pagan test), show no autocorrelation (Durbin-Watson test), are normal (Shapiro-Wilk test), and they are more linear (p value of the RESET test is now 0.015, instead of 0.000005). Linearity has improved, but is still violated somewhat.

Dealing with outliers

In this case, there are no real clear outliers. Yes, observations 8, 24, and 112 are labeled as the most extreme in the last set of residual diagnostic plots. But they are still within what I consider the "reasonable" range. But how does one define a limit to the reasonable? When is an extreme value a real outlier we have to deal with? Opinions vary about the issue, but I favor conservatism.

My rule is that, unless the value is clearly impossible or an error in data entry, I do not delete "outliers"; rather, I analyze all my data. Why? Because, I want my data to reflect natural or real variability. Indeed, variability is often what interests biologists the most.

Keeping extreme values is the fairest way to proceed, but it often creates other issues. These values will often be the main reason why the data fail to meet the assumptions of the statistical analysis. One solution is to run the analysis with and without the outliers, and compare the results. In many cases, the two analyses will be qualitatively similar: the same conclusions will be reached, and the effect size will not be very different. Sometimes, however, this comparison will reveal that the presence of the outliers changes the story. The logical conclusion then is that the results depend on the outliers and that the data at hand are not very conclusive.

As an example, let's rerun the analysis after eliminating observations labeled 8, 24, and 112:

```
RegModel.3 <- lm(log10(fklength)~log10(age),
data=sturgeon.male, subset=! (rownames(sturgeon.male) %in%
c('8', '24', '112'))
summary(RegModel.3)
```

to obtain:

```
Call:
lm(formula = log10(fklength) ~ log10(age), data = sturgeon.male,
    subset = !(rownames(sturgeon.male) %in% c("8", "24", "112")))

Residuals:
    Min       1Q   Median       3Q      Max
-0.0691634 -0.0173903  0.0009862  0.0185900  0.0476466

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.22676    0.02431   50.46  <2e-16 ***
log10(age)   0.31219    0.01932   16.16  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

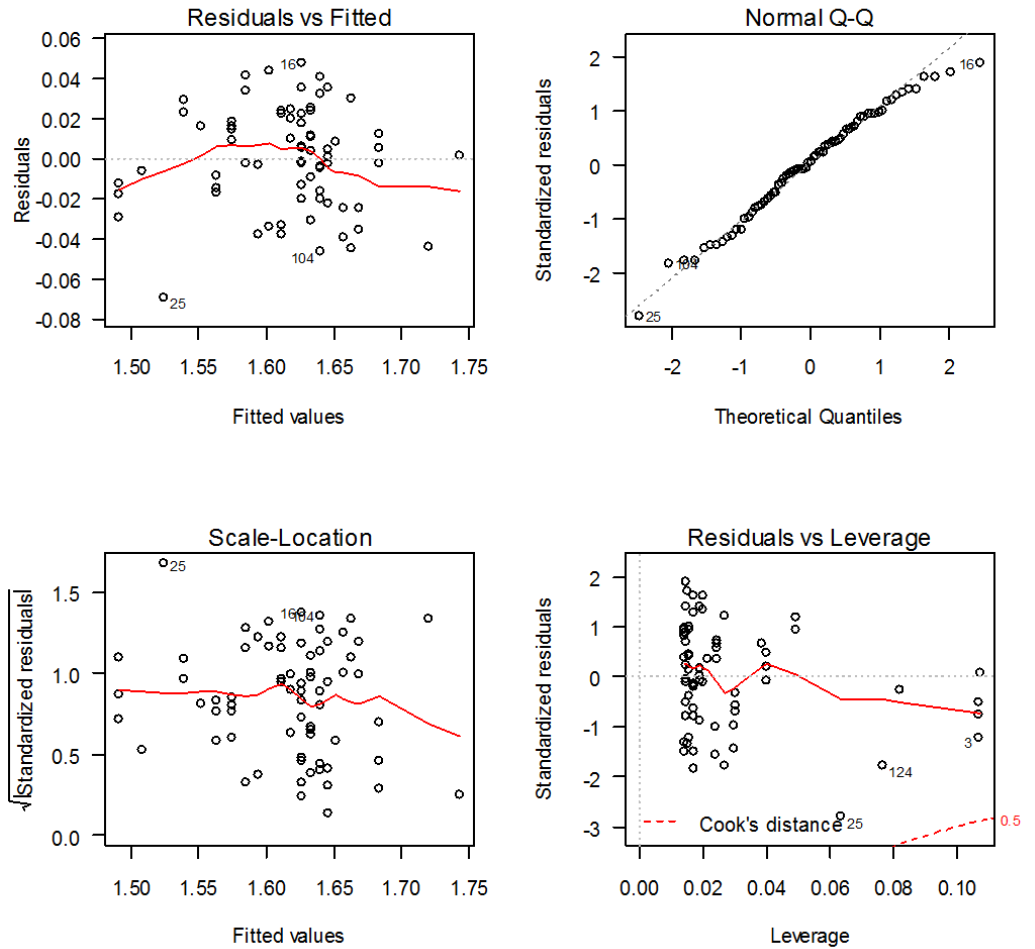
Residual standard error: 0.02554 on 70 degrees of freedom
(5 observations deleted due to missingness)
Multiple R-squared:  0.7885,    Adjusted R-squared:  0.7855
F-statistic: 261 on 1 and 70 DF,  p-value: < 2.2e-16
```

The intercept, slope, and R squared are about the same, and the significance of the slope is still astronomical. Removing the "outliers" has little effect in this case.

As for the diagnostic residual plots and the formal tests of assumptions:

```
plot(RegModel.3)
library(lmtest)
sturgeon.male.subset <- subset(sturgeon, subset=!
(rownames(sturgeon.male) %in% c('8', '24', '112')))
bptest(RegModel.3)
dwtest(RegModel.3)
resettest(RegModel.3)
shapiro.test(residuals(RegModel.3))
```

Figure 9.



```

> bpt est ( RegMdel . 3)

studentized Breusch-Pagan test

dat a: RegMdel . 3
BP = 0.3001, df = 1, p-value = 0.5838

> dwt est ( RegMdel . 3)

Durbin-Watson test

dat a: RegMdel . 3
DW = 2.0171, p-value = 0.5074
alternative hypothesis: true autocorrelation is greater than 0

> reset test ( RegMdel . 3)

RESET test

dat a: RegMdel . 3
RESET = 3.407, df 1 = 2, df 2 = 68, p-value = 0.0389

> shapiro.test ( residuals ( RegMdel . 3) )

Shapiro-Wilk normality test
    
```

```
data: residuals(RegModel.3)
W = 0.98318, p-value = 0.4502
```

No real difference either. Overall, this suggests that the most extreme values do not have undue influence on the results.

Quantifying effect size in regression and power analysis

Biological interpretation differs from statistical interpretation. Statistically, we conclude that size increase with age (i.e. the slope is positive and different from 0). But this conclusion alone does not tell if the difference between young and old fish is large. The slope and the scatterplot are more informative than the p-value here. The slope (in log-log space) is 0.34. This means that for each unit increase of X ($\log_{10}(\text{age})$), there is an increase of 0.34 units of $\log_{10}(\text{fklngth})$. In other words, when age is multiplied by 10, fork length is multiplied by about 2 ($10^{0.34}$). Hmmm, length increases more slowly than age. This slope value (0.34) is an estimate of raw effect size. It measures how much length changes with age.

Power to detect a given slope

You can compute power with G*Power for some slope value that you deem of sufficient magnitude to warrant detection.

1. Go to **t Tests: Linear bivariate regression: One group, size of slope.**
2. Select **Post hoc: Compute achieved power- given α , sample size, and effect size**

For example, suppose that sturgeon biologists deem that a slope of 0.1 for the relationship between $\log_{10}(\text{fklngth})$ and $\log_{10}(\text{age})$ is meaningful and you wanted to estimate the power to detect such a slope with a sample of 20 sturgeons. Results from the log-log regression contain most of what you need:

summary(RegModel.2)

```
Call:
lm(formula = log10(fklngth) ~ log10(age), data = sturgeon.male)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.0827935 -0.0168373 -0.0007188  0.0211016  0.0874465
```

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.19199    0.02723   43.77 <2e-16 ***
log10(age)   0.34086    0.02168   15.72 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03015 on 73 degrees of freedom
(5 observations deleted due to missingness)
Multiple R-squared:  0.772,    Adjusted R-squared:  0.7688
F-statistic: 247.1 on 1 and 73 DF,  p-value: < 2.2e-16

```

Note the Residual standard error value (0.03015). You will need this.

The other thing you need is an estimate of the standard deviation of $\log_{10}(\text{age})$.

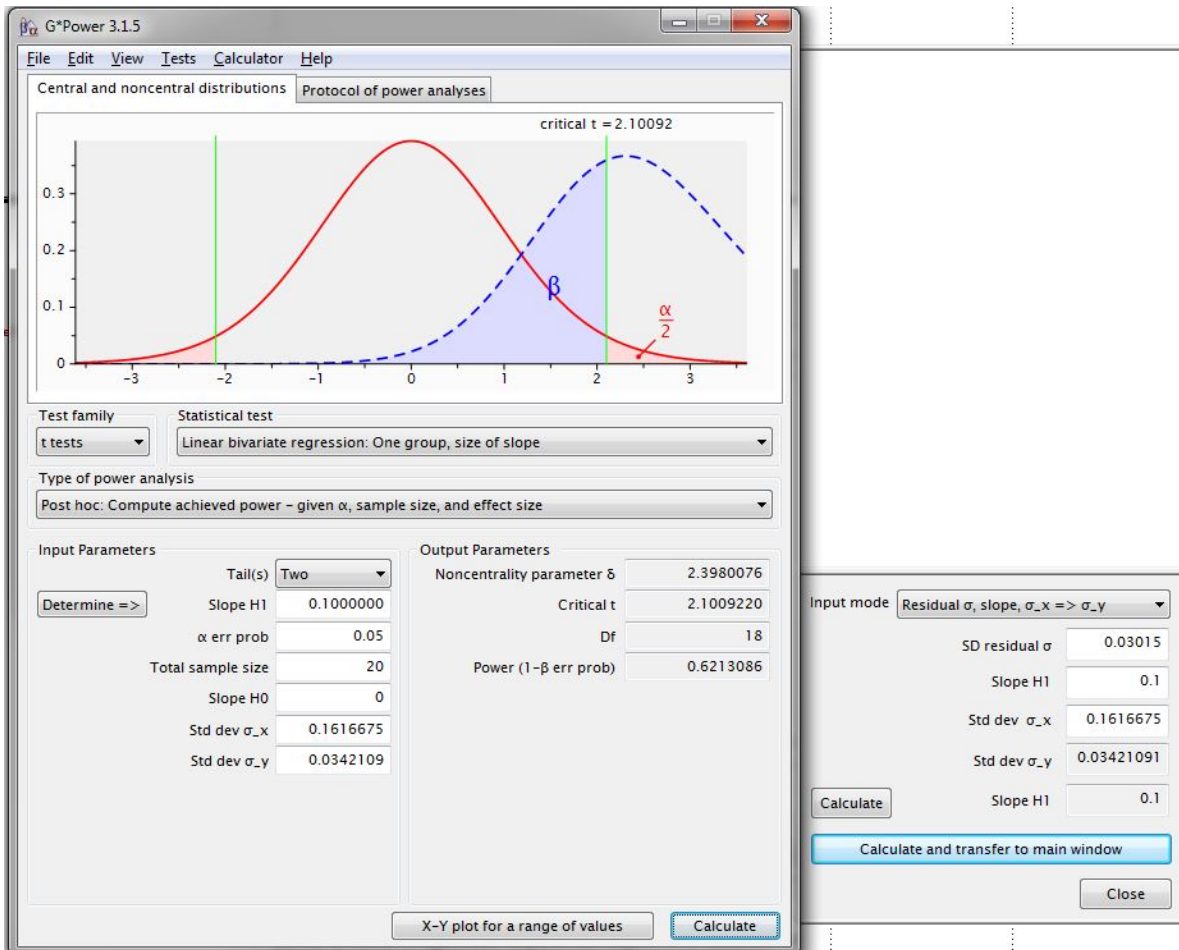
R can (of course) compute it. Be careful, the `sd()` function will return NA if there are missing values. You can get around this by adding `na.rm=TRUE`:

```
sd(log10(sturgeon.male$age), na.rm=TRUE)
[1] 0.1616675
```

You can then enter these values (slope to be detected, sample size, alpha, standard deviation of the independent variable) to calculate another quantity that G*Power needs (standard deviation of y) using the Determine panel. Finally you can calculate Power. The filled panels should look like this:

Note: The SD of y can't just be taken from the data because if the slope changes (e.g. H1) then this will change the SD of y. SD y needs to be estimated from the observed scatter around the line and the hypothesized slope).

Figure 10.

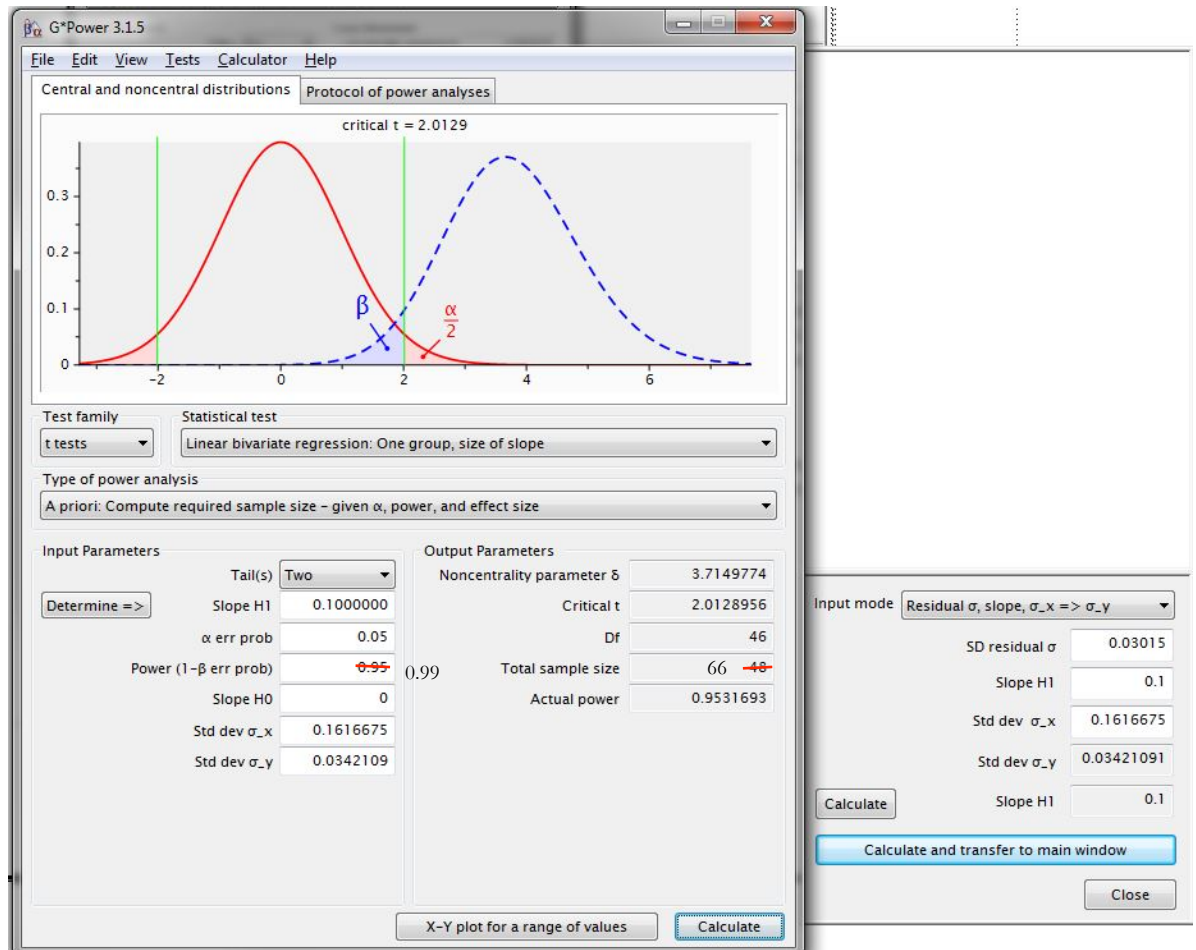


Power to detect a significant slope, if the slope is 0.1, variability of data points around the regression is like in our sample, for a sample of 20 sturgeons, with $\alpha=0.05$ is 0.621. Only about 2/3 of samples of that size would detect a significant effect of age on fkglnth.

Sample size required to achieve desired power

To estimate the sample size required to achieve 99% power to detect a slope of 0.1 (in log-log space), with $\alpha=0.05$, you simply change the type of analysis:

Figure 11.



By increasing sample size to 66, with the same assumptions as before, power increases to 99%.

Bootstrapping the simple linear regression with R

A non-parametric test for the intercept and slope of a linear regression can be obtained by bootstrapping:

```
# Bootstrap 95% CI for regression coefficients
library(boot)
# function to obtain regression weights
bs <- function(formula, data, indices) {
  d <- data[indices,] # allows boot to select sample
  fit <- lm(formula, data=d)
  return(coef(fit))
}
# bootstrapping with 1000 replications
```

```

results <- boot(data=sturgeon.male, statistic=bs,
               R=1000, formula=log10(fklength)~log10(age))

# view results
results
> results

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = sturgeon.male, statistic = bs, R = 1000, formula = log10(fklength)
~ log10(age))

Bootstrap Statistics:
      original 1      bias 2      std. error 3
t1*  1.1919926 -8.936048e-05  0.03532614
t2*  0.3408557  1.116266e-04  0.02789170

```

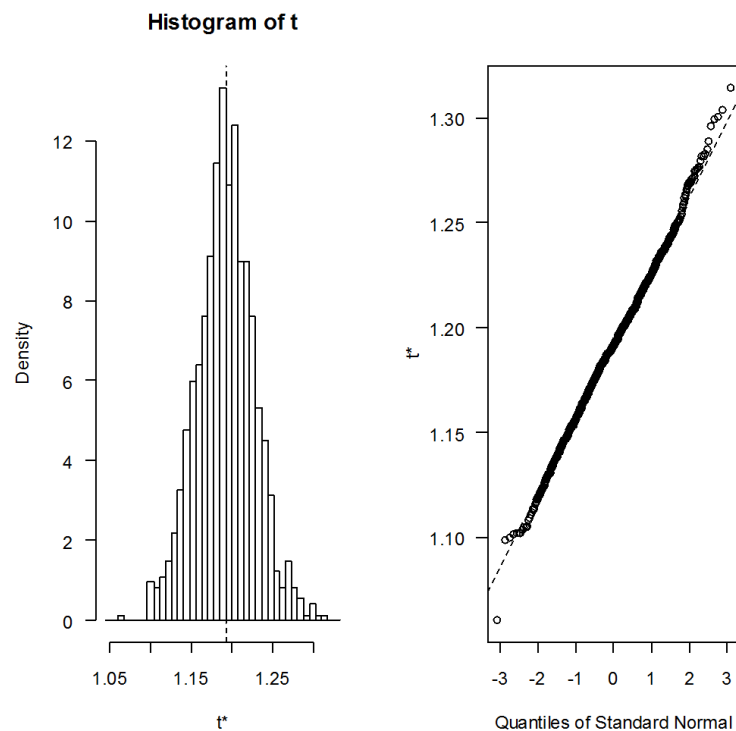
1 For each parameter in the model (here the intercept is labeled t1* and the slope of the regression line is labeled t2*), you obtain the original parameter estimate (on all non-bootstrapped data)

2 bias is the difference between the mean value of all bootstrap estimates and the original value

3 the standard error of the bootstrap estimate

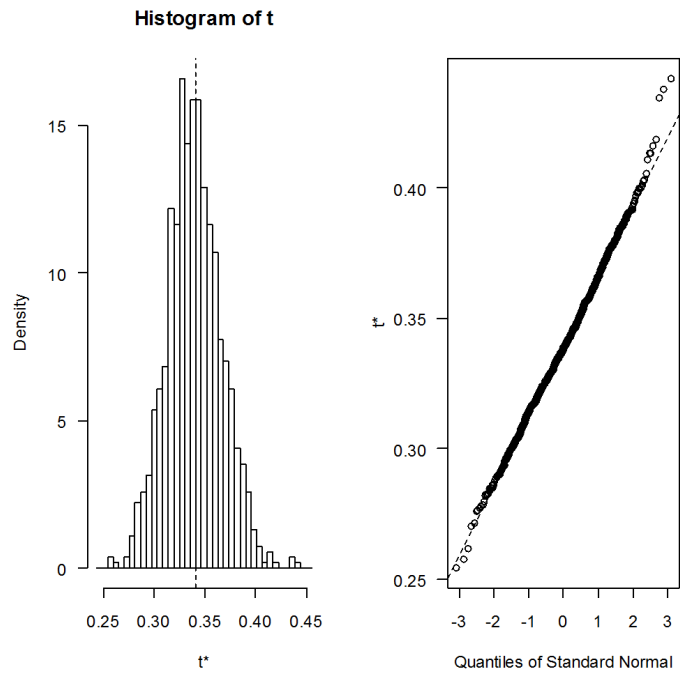
```
plot(results, index=1) # intercept
```

Figure 12.



```
plot(results, index=2) # log10(age)
```

Figure 13.



The distribution of the bootstrapped estimates is rather Gaussian, with only small deviations in the tails (where it counts for confidence intervals...). One could use the standard error of the bootstrap estimates to calculate a symmetrical confidence interval as mean \pm t SE. But, given that R can as easily calculate a bias-corrected adjusted (BCa) confidence interval, or one based on the actual distribution, (Percentile) why not have it do it all:

```
# get 95% confidence intervals
boot.ci(results, type = "all", index = 1)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "all", index = 1)

Interval s :
Level      Normal          Basic
95%      ( 1.124, 1.256 )    ( 1.123, 1.255 )

Level      Percentile          BCa
95%      ( 1.129, 1.261 )    ( 1.104, 1.247 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
Warning message:
In boot.ci(results, type = "all", index = 1) :
  bootstrap variances needed for studentized intervals
```

```
boot.ci(results, type = "all", index = 2)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
```

Based on 1000 bootstrap replicates

```

CALL :
boot.ci(boot.out = results, type = "all", index = 2)

Intervals :
Level      Normal          Basic
95% ( 0.2898, 0.3946 ) ( 0.2902, 0.3931 )

Level      Percentile      BCa
95% ( 0.2886, 0.3915 ) ( 0.2966, 0.4044 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
Warning message:
In boot.ci(results, type = "all", index = 2) :
  bootstrap variances needed for studentized intervals

```

Here the 4 types of CI that R managed to calculate are essentially the same. Had data been violating more strongly the standard assumptions (normality, homoscedasticity), then the different methods (Normal, Basic, Percentile, and BCa) would have diverged more. In that case, which one is best? BCa has the favor of most, currently.

Lab- Two - sample comparisons

After completing this laboratory exercise, you should be able to:


- Use R to visually examine data.
- Use R to compare the means of two normally distributed samples.
- Use R to compare the means of two non-normally distributed samples.
- Use R to compare the means of two paired samples

Visual examination of sample data

One of the first steps in any type of data analysis is to visualize your data with plots and summary statistics, to get an idea of underlying distributions, possible outliers, and trends in your data. This often begins with plots of the data, such as histograms, probability plots, and box plots, that allow you to get a feel for whether your data are normally distributed, whether they are correlated one to the other, or whether there are any suspicious looking points that may lead you to go back to the original data file to check for errors.

Suppose we want to test the null hypothesis that the size, as indexed by fork length (`fklength` - the length, in cm, from the tip of the nose to the base of the fork in the caudal fin), of sturgeon at The Pas and Cumberland House is the same. To begin, we have a look at the underlying distributions of the sample data to get a feel for whether the data are normally distributed in each sample. We will not actually test for normality at this point; the assumption of normality in parametric analyses refers always to the residuals and not the raw data themselves. However, if the raw data are non-normally distributed, then you usually have good reason to suspect that the residuals also will be non-normally distributed.

An excellent way to visually compare a data distribution to a normal distribution is to superimpose a histogram of the data and a normal curve. To do so, we must proceed in two steps: 1) tell R that we want to make a histogram with a density curve superimposed, 2) tell R that we want this to be done for both locations.

 Using the data file `sturgeon.Rdata`, generate histograms for `fklength` data at The Pas and Cumberland House.

```
require(ggplot2)

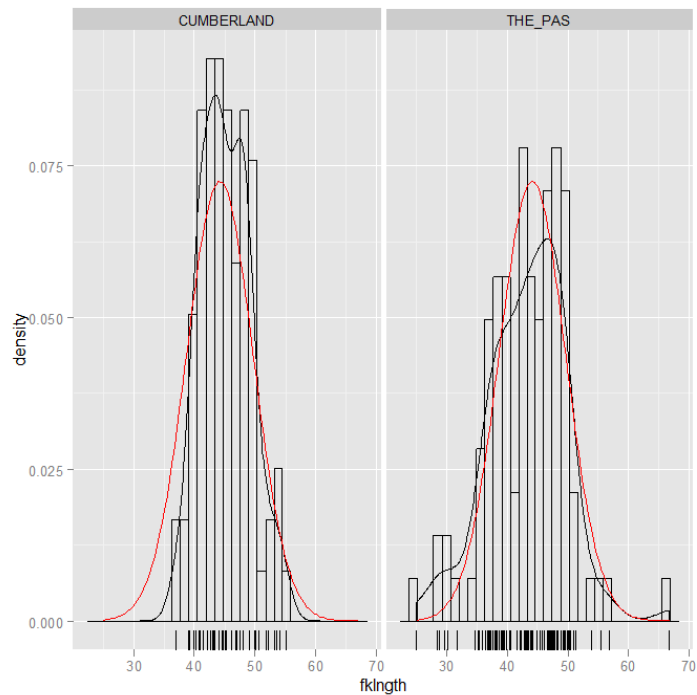
# use "sturgeon" dataframe to make plot called mygraph
```

```

# and define x axis as representing fklength
mygraph <- ggplot(sturgeon, aes(x = fklength, xlab="Fork
length (cm)"))
# add data to the mygraph ggplot
mygraph <- mygraph +
  # add data density smooth
  geom_density() +
  # add rug (bars at the bottom of the plot)
  geom_rug() +
  # add black semitransparent histogram
  geom_histogram(aes(y = ..density..),
                 color = "black",
                 alpha = 0.3) +
  # add normal curve in red, with mean and sd from fklength
  stat_function(fun = dnorm,
               args = list(
                 mean = mean(sturgeon$fklength),
                 sd = sd(sturgeon$fklength)
               ),
               color = "red")
#display graph, by location
mygraph+facet_grid(.~location)

```

Figure 1.



Based on your visual inspection, are the two samples normally distributed?

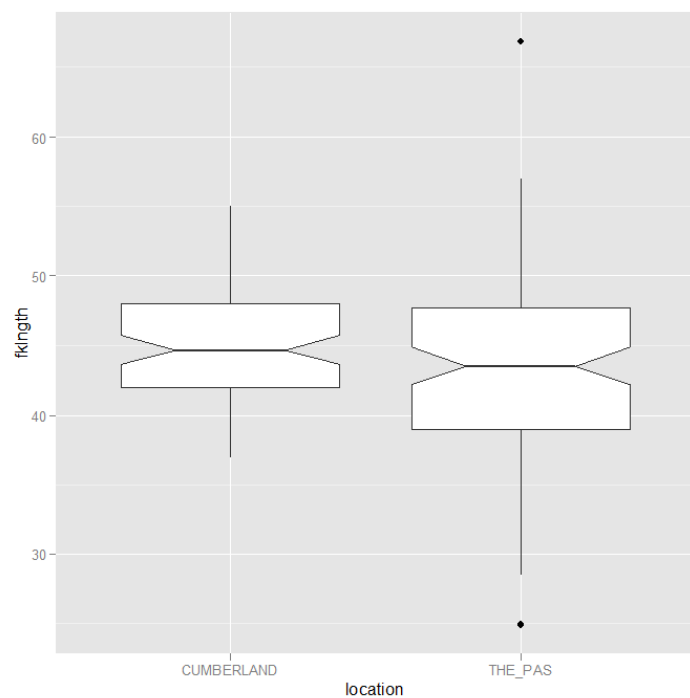
Visual inspection of these plots suggests that this variable is approximately normally distributed in each sample.

Since we are interested in finding out if mean fish size differs among the two locations, it is probably also a good idea to generate a graph that compares the two groups of data. A box plot works well for this.

📊 Generate a box plot of `fklength` grouped by `location`. What do you conclude about differences in size among the two locations?

```
ggplot(data=sturgeon, aes(x=location,
y=fklength)) + geom_boxplot(notch=TRUE)
```

Figure 2.



It would appear as though there are not big differences in fish size among the two locations, although fish size at The Pas looks to be more variable, with a bigger range in size and outliers (defined as values $> 1.5 \times$ inter-quartile range) at both ends of the distribution.

Comparing means of two independent samples: parametric and non-parametric comparisons

📊 Test the null hypothesis that the mean `fklength` of The Pas and Cumberland House samples are the same. What do you conclude?

```

> # t-test assuming equal variances
> t.test(fklngh~location, alternative='two.sided',
conf.level=.95,
+   var.equal=TRUE, data=sturgeon)

      Two Sample t-test

data:  fklngh by location
t = 2.1359, df = 184, p-value = 0.03401
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1308307 3.2982615
sample estimates:
mean in group CUMBERLAND      mean in group THE_PAS
      45.08439                43.36984

>
> # t-test assuming unequal variances
> t.test(fklngh~location, alternative='two.sided',
conf.level=.95,
+   var.equal=FALSE, data=sturgeon)

      Welch Two Sample t-test

data:  fklngh by location
t = 2.2201, df = 169.804, p-value = 0.02774
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1900117 3.2390804
sample estimates:
mean in group CUMBERLAND      mean in group THE_PAS
      45.08439                43.36984

>
> # non-parametric wilcoxon test (aka Mann-Whitney U test)
> wilcox.test(fklngh ~ location, alternative="two.sided",
data=sturgeon)

      Wilcoxon rank sum test with continuity correction

data:  fklngh by location
W = 4973, p-value = 0.06296
alternative hypothesis: true location shift is not equal to 0

>

```

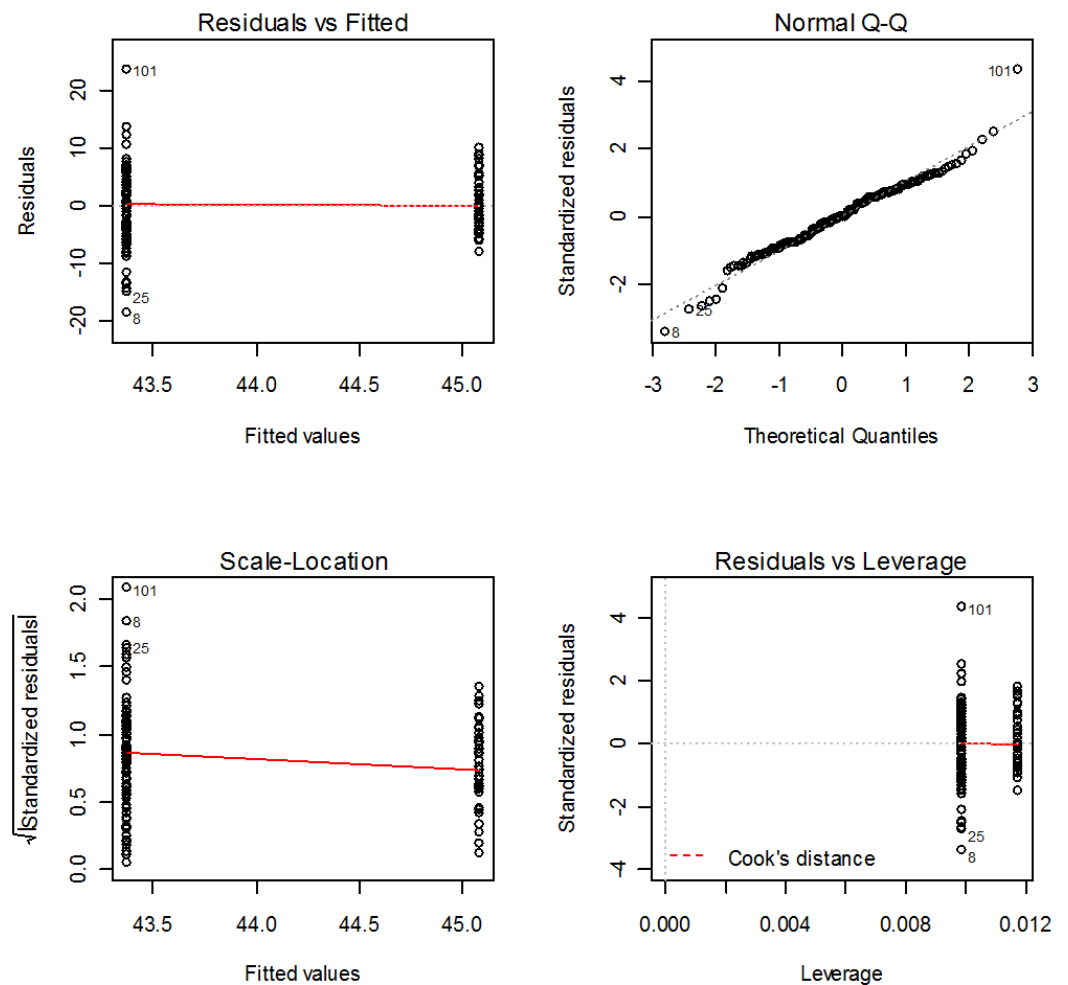
Based on the t -test, we would reject the null hypothesis, i.e. there is a significant (but not highly significant) difference in mean fork length between the two populations.

Note that using the Wilcoxon rank sum test, we do not reject the null hypothesis. The two different tests therefore give us two different results. The significant difference obtained using the t -test may, at least in part, be due to deviations from normality or homoscedasticity; on the other hand, the non-significant difference obtained using the U -statistic may be due to the fact that for fixed sample size, the power of a non-parametric test is lower than the corresponding parametric test. Given the p values obtained from both tests, and the fact that for samples of this size (84 and 101), the t -test is comparatively robust with respect to non-normality, I would be inclined to reject the null hypothesis. In practice to avoid P-hacking, you should decide which test is appropriate first and then apply and interpret it, or if you decide to do both you should present results of both and interpret accordingly.

Before accepting the results of the parametric t -test and rejecting the null hypothesis that there is no difference in size between the two locations, one should do some sort of assessment to determine if the model fits the assumption of normally distributed residuals and equal variances. Preliminary examination of the raw data suggested the data appeared roughly normal but there might be problems with variances (since the spread of data for The_Pas was much greater than for Cumberland). We can examine this more closely by looking at the residuals. An easy way to do so, is to fit a linear model and use the residual diagnostic plots:

```
AnovaModel.1 <- lm(fklnlngth ~ location, data=sturgeon)
par(mfrow = c(2, 2))
plot(AnovaModel.1)
```

Figure 3.



The first plot above shows the spread of the residuals around the estimated values for the two groups and allows us to get a feel for whether there are problems with the assumption of homogeneity of variances. If the variances were equal, the vertical spread of the two clusters of points should be about the same. The above plot shows that the vertical spread of the group with the smaller mean is greater than it is for the larger mean, suggesting again that there are problems with the variances. We can test this formally by examining the mean differences in the absolute value of the residuals.

The second graph above is a normal QQ plot (or probability plot) of the residuals of the model. Note that these generally fall on a straight line, suggesting there is no real problem with normality. We can do a formal test for normality on the residuals using the Shapiro-Wilk test.

```
shapiro.test(residuals(AnovaModel.1))
```

```
Shapiro-Wilk normality test
data: residuals(AnovaModel.1)
W = 0.9747, p-value = 0.001858
```

Hummm. The test indicates that the residuals are not normal. But, given that (a) the distribution is not very far (at least visually) from normal, and that (b) the number of observations in each location is reasonably large (i.e. >30), we do not need to be overly concerned with this violation of the normality assumption.:

How about equality of variance?

```
> require(car)
> leveneTest(AnovaModel.1)
```

```
Levene's Test for Homogeneity of Variance
  Df F value Pr(>F)
group 1 11.514 0.0008456 ***
    184
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> require(lmtest)
> bptest(AnovaModel.1)
```

```
studentized Breusch-Pagan test
data: AnovaModel.1
BP = 8.8015, df = 1, p-value = 0.00301
>
```

The above are the results of two tests implemented in R (in the `car` and `lmtest` packages) that can be used to test for equal variances in t-tests or linear models involving only discontinuous or categorical independent variables. **Doing the two of them is overkill.** There is

not much to prefer one test over another. Levene test is possibly the better known. It tests whether the mean of absolute values of the residuals differs among groups. The Breusch-Pagan test has the advantage of being applicable to more linear models (it can deal with regression-type continuous independent variables, at least to some extent). It tests whether the studentized (i.e. scaled by their sd estimate) squared residuals vary with the independent variables in a linear model. In this case, both indicate that variances are unequal.

On the basis of these results, we conclude that there is evidence (albeit weak) to reject the null hypothesis of no difference in `fklngh` by `location`. We have modified the t -test to accommodate unequal variances, and are satisfied that the assumption of normally distributed residuals is sufficiently met. Thus, it appears that `fklngh` at Cumberland is greater than `fklngh` at The Pas.

Bootstrap and permutation tests to compare 2 means

Bootstrap and permutation tests can be used to compare means (or other statistics) between pairs of samples. The general idea is simple, and it can be implemented in more ways than I can count.. Here, I use existing tools and the fact that a comparison of means can be construed as a test of a linear model. We will be able to use similar code later on when we fit more complex (but fun!) models.

```
library(boot)
```

The first section defines the function that I called `bs` that simply extracts coefficients from a fitted model:

```
# function to obtain model coefficients for each iteration
bs <- function(formula, data, indices) {
  d <- data[indices, ]
  fit <- lm(formula, data = d)
  return(coef(fit))
}
```

The second section with the `boot()` command is where the real work is done: take data in `sturgeon`, bootstrap $R=1000$ times, each time fit the model `fklngh` vs `location`, and keep the values calculated by the `bs` function.

```
# bootstrapping with 1000 replications
results <- boot(data = sturgeon, statistic = bs, R = 1000,
  formula = fklngh ~ location)

# view results
```

results

ORDINARY NONPARAMETRIC BOOTSTRAP

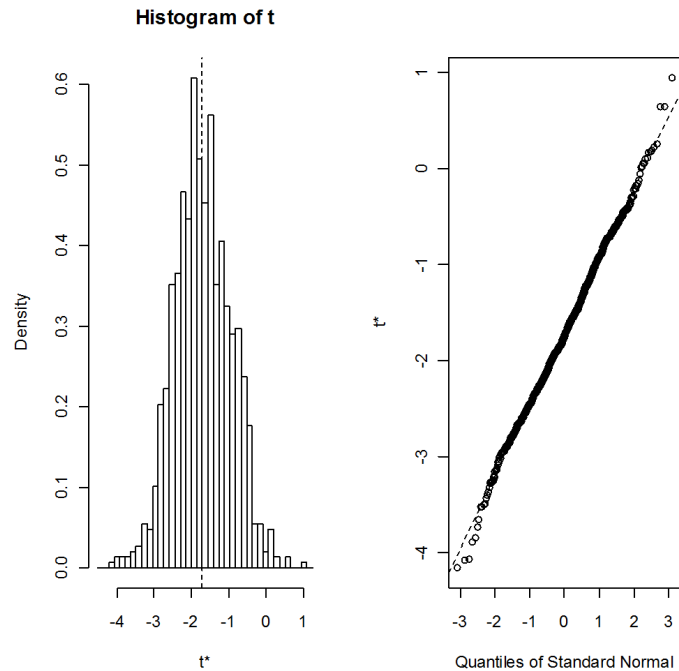
```
Call :
boot(data = sturgeon, statistic = bs, R = 1000, formula = fklngth ~
      location)
```

```
Bootstrap Statistics :
      original      bias      std. error
t1* 45.084391 -0.005634742  0.4164833
t2* -1.714546  0.012011519  0.7510652
```

So we get the original estimates for the two coefficients in this model: the mean at the first (alphabetical) location, Cumberland, and the difference in means between Cumberland and The Pas). It is the second parameter, the difference between means, which is of interest here.

```
plot(results, index = 2)
```

Figure 4.



```
# get 95% confidence intervals
```

```
boot.ci(results, type = "bca", index = 2)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates
```

```
CALL :
boot.ci(boot.out = results, type = "bca", index = 2)
```

```
Interval s :
Level      BCa
95%      (-3.052, -0.211 )
Calculations and Intervals on Original Scale
```

The 95% CI for the difference between the two means does not include 0. Hence, the bootstrap test indicates that the two means are not equal

Permutation tests for linear models can easily be done using the lmPerm library:

```
require(lmPerm)
myModelProb <- lmp(fklngh ~ location, data = sturgeon, perm
= "Prob", Ca=0.0001, maxIter=5000)
```

The lmp() function does all the work for us. Here it is run with the option perm to control the stopping rule used. Option Prob stops the sampling when the estimated standard deviation of the p-value falls below some fraction of the estimated. It is one of many stopping rules that one could use to do permutations on a subset of all the possibilities (because it would take foreeeever to do them all, even on your fast machine).

```
summary(myModelProb)
```

```
Call :
lmp(formula = fklngh ~ location, data = sturgeon, perm = "Prob")

Residuals:
    Mn      1Q  Median      3Q     Max
-18.40921 -3.75370 -0.08439  3.76598 23.48055

Coefficients:
              Estimate Iter 1 Pr(Prob) 2
location1    0.8573 1117    0.0824 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 5.454 on 184 degrees of freedom
Multiple R-Squared:  0.02419,    Adjusted R-squared:  0.01889
F-statistic: 4.562 on 1 and 184 DF,  p-value: 0.03401 3
```

1 In this case, the Prob stopping rule stopped after 1117 iterations. Note that this number will vary each time you run this snippet of code. These are random permutation results, so expect variability.

2 The estimated probability associated to H_0 is 0.0824. The observed difference in fklngh between the two locations was larger than the permuted differences in about $(1-0.0824=\text{about } 92\%)$ of the 1117 cases. Mind you, 1117 permutations is not a large number, so

small p values can't be expected to be very precise. If it is critical that you get more precise p values, more permutations would be needed. Two parameters can be tweaked: `maxIter`, the maximum number of iterations (default=5000), and `Ca`, that stops iterations when estimated standard error of the estimated p is less than $Ca \cdot p$. Default 0.1.

③ The rest is the standard output for the model fitted to the data, with the standard parametric test. Here the p-value, assuming all assumptions are met, is 0.034.

Comparing the means of paired samples

In some experimental designs, individuals are measured twice: common examples are the measurement of the same individual at two different times during development, or of the same individual subjected to two different experimental treatments. In these cases, the two samples are not independent (they include the same individuals), and a paired comparison must be made.

The file `skulldat.Rdata` shows measurements of lower face width of 15 North American girls measured at age 5 and again at age 6 years (data from Newman and Meredith, 1956).

Let's first run a standard t-test comparing the face width at age 5 and 6, not taking into account that the data are not independent and that they are consecutive measurements on the same individuals.

```
> t.test(width~age, alternative='two.sided',
+ conf.level=.95, var.equal=TRUE, data=skulldat )
```

```
Two Sample t-test

data: width by age
t = -1.7812, df = 28, p-value = 0.08573
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.4300032  0.0300032
sample estimates:
mean in group 5 mean in group 6
 7.461333      7.661333

>
```

Now, let's run the appropriate paired t-test. What do you conclude? Compare this with the previous result and explain any differences.

```
> t.test(skulldat$width5, skulldat$width6, alterna-
+ tive='two.sided',
+ conf.level=.95, paired=TRUE)
```

```
Paired t-test

data: skulldat$width5 and skulldat$width6
```

```

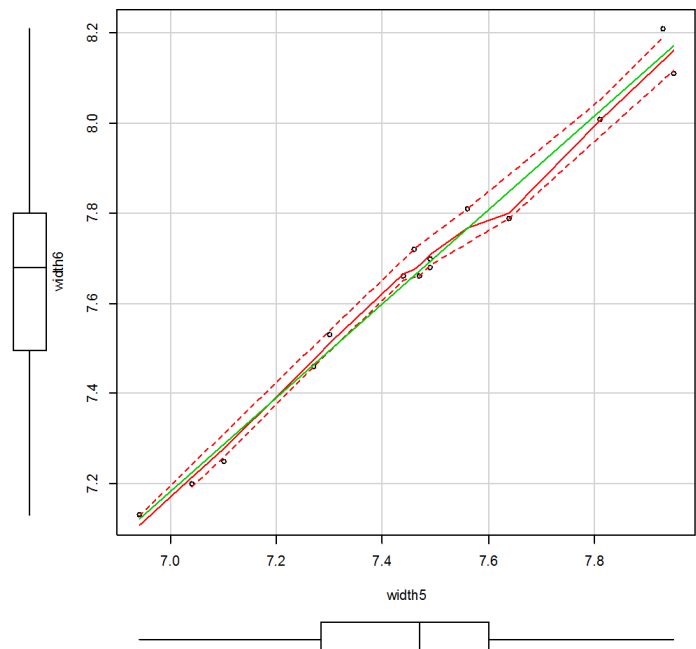
t = -19.5411, df = 14, p-value = 1.473e-11
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2241710 -0.1798290
sample estimates:
mean of the differences
 -0.202

```


The first analysis above assumes that the two samples of girls at age 5 and 6 are independent samples, whereas the second analysis assumes that the same girl is measured twice, once at age 5 and once at age 6 years.

Note that in the former case, we accept the null based on $p = .05$, but in the latter we reject the null. In other words, the appropriate (paired sample) test shows a very significant effect of age, whereas the inappropriate one does not. The reason is because there is a strong correlation between face width at age 5 and face width at age 6:

Figure 5.



with $r = .993$. In the presence of correlation, the standard error of the pairwise difference in face width at age 5 and 6 is much smaller than the standard error of the difference between the mean face width at age 5 and 6. Thus, the associated t -statistic will be much larger for a paired sample test, i.e. the power of the test is much greater, and the p values are smaller.

 Repeat the above procedure with the nonparametric alternative, the Wilcoxon signed-rank test. What do you conclude?

```

> wilcox.test(skulldat$width5, skulldat$width6, alterna-
+   paired=TRUE)
Warning in wilcox.test.default(skulldat$width5, skulldat$width6, alternative
= "two.sided", :
cannot compute exact p-value with ties

Wilcoxon signed rank test with continuity correction

data:  skulldat$width5 and skulldat$width6
V = 0, p-value = 0.0007211
alternative hypothesis: true location shift is not equal to 0

>

```

So, we reach the same conclusion as we did using the paired sample *t*-test and conclude there are significant differences in skull sizes of girls aged 5 and 6 (what a surprise!).

But, wait a minute. We have used two-tailed tests here. But, given what we know about how children grow, a one-tail hypothesis would be preferable. This can be done by changing the alternative option. One uses the alternative hypothesis to decide if it is "less" or greater". Here, we expect that if there is an effect (i.e the alternative hypothesis), width5 will be less than width6.

```

> t.test(skulldat$width5, skulldat$width6, alterna-
+   conf.level=.95, paired=TRUE)

Paired t-test

data:  skulldat$width5 and skulldat$width6
t = -19.5411, df = 14, p-value = 7.363e-12
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
-Inf -0.1837930
sample estimates:
mean of the differences
-0.202

>
> wilcox.test(skulldat$width5, skulldat$width6, alterna-
+   exact=TRUE, paired=TRUE)
Warning in wilcox.test.default(skulldat$width5, skulldat$width6, alternative
= "less", :
cannot compute exact p-value with ties

Wilcoxon signed rank test with continuity correction

data:  skulldat$width5 and skulldat$width6
V = 0, p-value = 0.0003606
alternative hypothesis: true location shift is less than 0

```

References

- Bumpus, H.C. (1898) The elimination of the unfit as illustrated by the introduced sparrow, *Passer domesticus*. Biological Lectures, Woods Hole Biology Laboratory, Woods Hole, 11 th Lecture: 209 - 226.
- Newman, K.J. and H.V. Meredith. (1956) Individual growth in skeletal bigonial diameter during the childhood period from 5 to 11 years of age. Amer. J. Anat. 99: 157 - 187.

Lab- Single classification (one-way) ANOVA

After completing this laboratory exercise, you should be able to:


- Use R to do a one-way parametric ANOVA with multiple comparisons
- Use R to test the validity of the parametric ANOVA assumptions
- Use R to perform a one-way non-parametric ANOVA
- Use R to transform your data so that the assumptions of parametric ANOVA are met.

One-way ANOVA with multiple comparisons


The one-way ANOVA is the multi-group analog of the t -test, which is used to compare two groups/levels. It makes essentially the same assumptions, and in the case of two groups/levels, is in fact mathematically equivalent to the t -test.

In 1960 - 1962, the Grand Rapids Dam was built on the Saskatchewan River upstream of Cumberland House. There are anecdotal reports that during dam construction, a number of large sturgeon were stranded and died in shallow pools. Surveys of sturgeon were carried out in 1954, 1958, 1965 and 1966 with fork length (`fklength`) and round weight (`rdwght`) being recorded (not necessarily both measurements for each individual). These data are in the data file `dam10dat.Rdata`.

Visualizing the data

 Using `dam10dat.Rdata`, you must first change the data type of the numerical variable `year`, so that R recognizes that we wish to treat this variable as a factor variable and not a continuous variable.

```
Dam10dat$year<-as.factor(Dam10dat$year)
ls.str()
attach(Dam10dat)
```

 Next, have a look at the `fklength` data, just as we did in the last lab for t -tests. Create a histogram with density line grouped by year to get a feel for what's happening with your data. What can you say about these data?

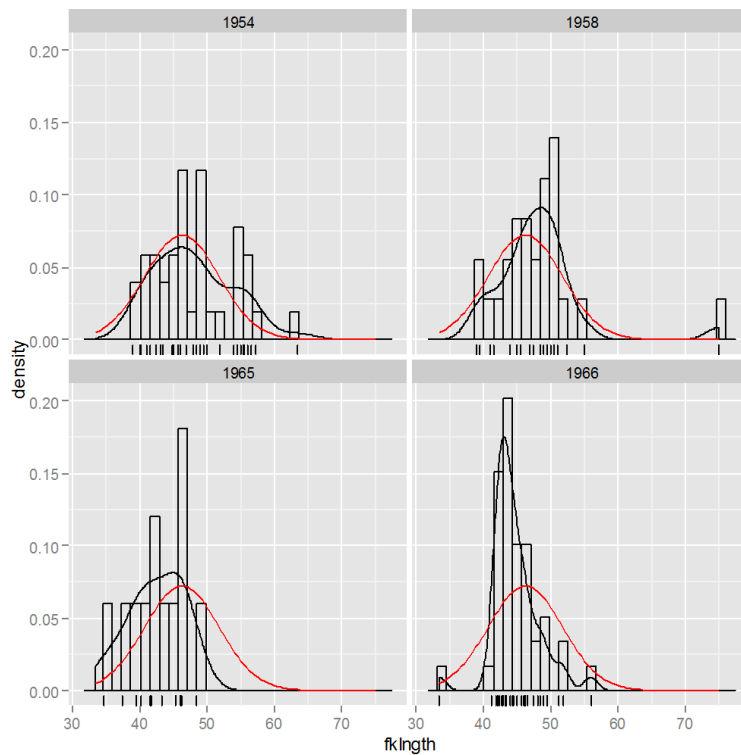
```
require(ggplot2)
```

```

# use "Dam10dat" dataframe to make plot called mygraph
# and define x axis as representing fklnth
mygraph <- ggplot(Dam10dat, aes(x = fklnth))
# add data to the mygraph ggplot
mygraph <- mygraph + xlab("Fork length (cm)") +

# add data density smooth
geom_density() +
# add rug (bars at the bottom of the plot)
geom_rug() +
# add black semitransparent histogram
geom_histogram(aes(y = ..density..),
               color = "black",
               alpha = 0.3) +
# add normal curve in red, with mean and sd from fklnth
stat_function(fun = dnorm,
             args = list(
               mean = mean(Dam10dat$fklnth),
               sd = sd(Dam10dat$fklnth)
             ),
             color = "red")
#display graph, by year
mygraph+facet_wrap(~year, ncol=2)


```



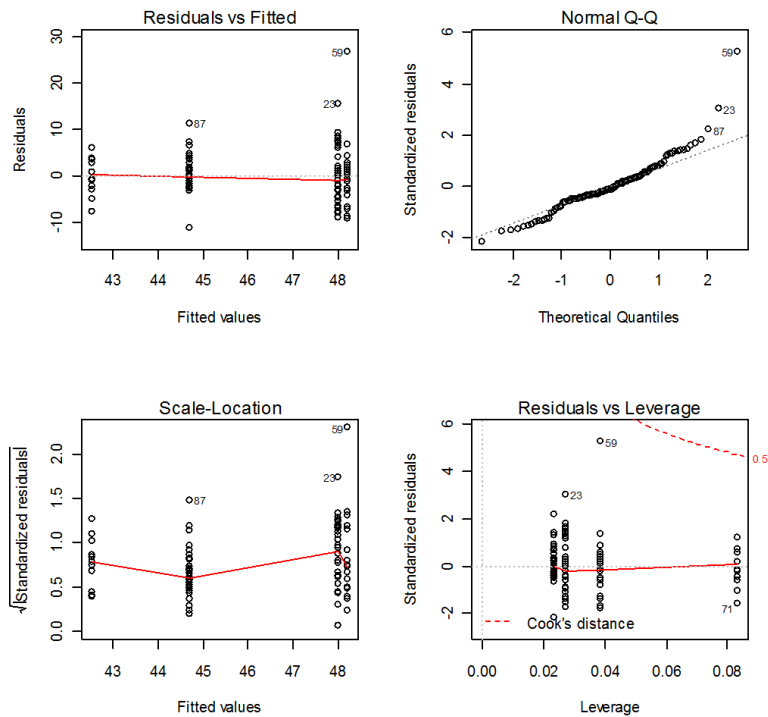
It appears as though there may have been a small drop in `FKLENGTH` after the construction of the dam, but the data are variable and the effects are not clear. There might also be some problems with normality in the 1954 and 1966 samples, and it looks as though there are outliers in the 1958 and 1966 samples. Let's proceed with testing the assumptions of the ANOVA by running the analysis and looking at the residuals.

Testing the assumptions of a parametric ANOVA

Parametric one-way ANOVAs have three major assumptions: (1) the residuals are normally distributed; (2) the error variance is the same for all groups (homoscedasticity); and (3) the residuals are independent. These assumptions must be tested before we can accept the results of any parametric ANOVA.


 Carry out a one-way ANOVA on `fklength` by `year` and produce the residual diagnostic plots

```
# Fit anova model and plot residual diagnostics
# but first, save current par and set graphic page to hold 4
graphs
opar <- par(mfrow = c(2, 2))
anova.modell <- lm(fklength ~ year, data=Dam10dat)
plot(anova.modell)
par(opar)
```



Judging from the plots, it looks as though there may be problems with both normality and variance heterogeneity. Note that there is one point (case 59) with large expected values and a large residual that appear to lie well off the line: this is the outlier we noted earlier. This point might be expected to inflate the variance for the group it belongs to.

Formal tests may also provide some insight as to whether we should be concerned about normality and variance heterogeneity.


 Perform a normality test on the residuals from the ANOVA.

```
shapiro.test(residuals(anova.model1))
```

Shapiro-Wilk normality test

```
data: residuals(anova.model1)
W = 0.9157, p-value = 1.63e-06
```

This test confirms our suspicions from the probability plot: the residuals are not normally distributed. Recall, however, that the power here is high, so only small deviations from normality are required to reject the null.

 Next, test for homoscedasticity:

```
require(car)
leveneTest(fklngh ~ year, data=Dam10dat)
```

```
Levene's Test for Homogeneity of Variance
      Df F value Pr(>F)
group  3  2.8159 0.04234 *
      114
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

The probability value tells you that you can reject the null hypothesis that there is no difference in variances among years. Thus, we conclude there is evidence that the variances in the groups are not equal.

Performing the ANOVA

Let's look at the results of the ANOVA, assuming for the moment that assumptions are met well enough :

```
> summary(anova.model1)

Call:
lm(formula = fklngh ~ year)

Residuals:
    Min       1Q   Median       3Q      Max
-11.2116  -2.6866  -0.7116   2.2103  26.7885

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  48.0243 1      0.8566  56.061 < 2e-16 ***
year 1958     0.1872    1.3335   0.140  0.88859
year 1965    -5.5077    1.7310  -3.182  0.00189 **
year 1966    -3.3127    1.1684  -2.835  0.00542 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.211 2 on 114 degrees of freedom

Multiple R-squared:  0.1355, 3    Adjusted R-squared:  0.1128

F-statistic: 5.957 on 3 and 114 DF,  p-value: 0.0008246 4
```

1 Note the 4 coefficients printed. They can be used to obtain the predicted values for the model (i.e. the group means). The mean `fklngh` for the first year (1954) is 48.0243. The coefficients for the 3 other years are the difference between the mean for that year and for 1954. So, the mean for 1965 is $(48.0243 - 5.5077 = 42.5166)$. For each estimated coefficient, there is a standard error, a t-value and associated probability (for H_0 that the coefficient is 0). Note here that coefficients for 1965 and 1966 are both negative and significantly less than 0. Fish were smaller after the construction of the dam than in

1954. Take these p-values with a grain of salt: these are not corrected for multiple comparisons, and they constitute only a subset of the possible comparisons. In general, I pay little attention to this part of the output and look more at what comes next.

② The square root of the variance of the residuals (observed minus fitted values) corresponds to the amount of variability that is unexplained by the models (here an estimate of how much size varied among fish, once corrected for differences among years)

③ The R-squared is the proportion of the variance of the dependent variable that can be explained by the model. Here the model explains only 13.5% of the variability. Size differences among year are relatively small compared to the ranges of sizes that can occur within years. This corresponds well to the visual impression left by the histograms of fklngth per year.

④ This is the p-value for the "omnibus" test, the test that all means are equal. Here it is much smaller than 0.05 and hence we would reject H_0 and conclude that fklngth varies among the years.


The `anova()` command produces the standard ANOVA table that contains most of the same information.:

```
> anova(anova.model 1)
Analysis of Variance Table

Response: fklngth
          Df Sum Sq Mean Sq F value    Pr(>F)
year       3  485.26  161.75   5.9574 0.0008246 ***
Residuals 114 3095.30   27.15
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

The total variability in fklngth sums of square is partitioned into what can be accounted for by year (485.26) and what is left unexplained as residual variability (3095.30). Year indeed explains $(485.26 / (3095.30 + 485.26)) = 0.1355$ or 13.55% of the variability). The mean square of the residuals is their variance.

Performing multiple comparisons of means test

 The `pairwise.t.test()` function can be used to compare means and adjust (or not) probabilities for multiple comparisons by choosing one of the options for `p.adj`:

```
pairwise.t.test(Dam10dat$fklngth, Dam10dat$year, p.adj =
"none")
pairwise.t.test(Dam10dat$fklngth, Dam10dat$year, p.adj =
"bonf")
```

```
pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj =
"holm")
pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj =
"fdr")
```

```
> pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj =
"none")
```

This compares means of all pairs, not adjusting probabilities.

```
Pairwise comparisons using t tests with pooled SD
data: Dam10dat$fklngh and Dam10dat$year
      1954 1958 1965
1958 0.8886 -    -
1965 0.0019 0.0022 -
1966 0.0054 0.0079 0.1996
```

Option "bonf" adjusts the p-values according to the Bonferroni correction. In this case, since there are 6 p-values calculated, it amounts to simply multiplying the uncorrected p-values by 6 (unless the result is above 1, in that case, let $\text{adj } p = 1$).

```
Pairwise comparisons using t tests with pooled SD
data: Dam10dat$fklngh and Dam10dat$year
      1954 1958 1965
1958 1.000 -    -
1965 0.011 0.013 -
1966 0.033 0.047 1.000
P value adjustment method: bonferroni
```

Option "holm" is the sequential Bonferroni correction, where the p-values are ranked from ($i=1$) smallest to (N) largest. The correction factor for p-values is then $(N-i+1)$. Here, for example, we have $N=6$ pairs that are compared. The lowest uncorrected p-value is 0.0019 for 1954 vs 1965. The corrected p-value becomes $0.0019 \times (6-1+1) = 0.011$. The second lowest p-value is 0.0022. The corrected p-value is therefore $0.0022 \times (6-2+1) = 0.011$. For the highest p-value, the correction is $(N-N+1) = 1$, hence it is equal to the uncorrected probability.

```
pairwise.t.test(Dam10dat$fklngh, Dam10dat$year, p.adj = "holm")
Pairwise comparisons using t tests with pooled SD
data: Dam10dat$fklngh and Dam10dat$year
      1954 1958 1965
1958 0.889 -    -
1965 0.011 0.011 -
1966 0.022 0.024 0.399
P value adjustment method: holm
```

The "fdr" option is for controlling the false discovery rate. .

```
> pairwise.t.test(Dam10dat$fklength, Dam10dat$year, p.adj = "fdr")
```

Pairwise comparisons using t tests with pooled SD

```
data: Dam10dat$fklength and Dam10dat$year
```

```
      1954  1958  1965
1958 0.8886 -      -
1965 0.0066 0.0066 -
1966 0.0108 0.0119 0.2395
```

The four post-hoc tests here tell us the same thing: differences are all between two groups of years: 1954/58 and 1965/66, since all comparisons show differences between the 50's and 60's but no differences within the 50's or 60's. So, in this particular case, the conclusion is not affected by the choice of adjustment method.

But in other situations, you will observe contradictory results. Which one to choose? Unadjusted p-values are certainly suspect when there are multiple tests. On the other hand, the traditional Bonferroni correction is very conservative, and becomes even more so when there are a large number of comparisons. Recent work suggest that the fdr approach may be a good compromise when there are a lot of comparisons.

The Tukey method of multiple comparisons is one of the most popular and is easily performed with R (note, however, that there is a pesky bug that manifests itself when the independent variable can look like a number rather than a factor, hence the little pirouette with paste to add a space character before the first digit):

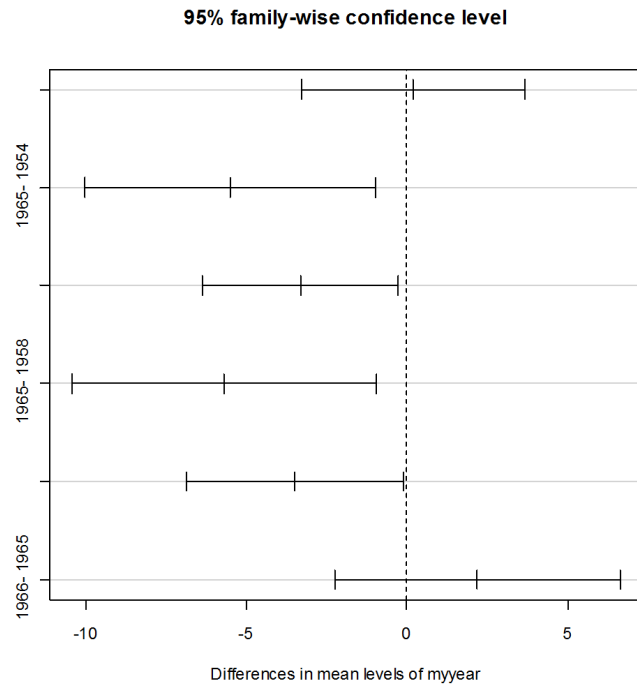
```
Dam10dat$myyear <- as.factor(paste(" ", Dam10dat$year))
TukeyHSD(aov(fklength ~ myyear))
```

```
      Tukey multiple comparisons of means
      95% family-wise confidence level
```

```
Fit: aov(formula = fklength ~ myyear)
```

```
$myyear
      diff      lwr      upr      p adj
1958- 1954  0.1872141 -3.289570  3.6639986 0.9990071
1965- 1954 -5.5076577 -10.021034 -0.9942809 0.0100528
1966- 1954 -3.3126964  -6.359223 -0.2661701 0.0274077
1965- 1958 -5.6948718 -10.436304 -0.9534397 0.0116943
1966- 1958 -3.4999106  -6.875104 -0.1247171 0.0390011
1966- 1965  2.1949612  -2.240630  6.6305526 0.5710111
```

```
plot(TukeyHSD(aov(fklength ~ myyear)))
```



The confidence intervals, corrected for multiple tests by the Tukey method, are plotted for differences among years. Unfortunately, the labels are not all printed because they would overlap, but the order is the same as in the preceding table.

The library `multcomp` can produce a better plot version, but requires a bit more code:

```
# Alternative way to compute Tukey multiple comparisons
# set up a one-way ANOVA
```

```
anova.fkl.vs.year <- aov(aov(fklngth ~ year))
```

```
# set up all-pairs comparisons for factor `year`
library(multcomp)
```

```
meandiff <- glht(anova.fkl.vs.year, linfct = mcp(year =
"Tukey"))
```

```
confint(meandiff)
```

Simultaneous Confidence Intervals

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = aov(fklngth ~ year))

Estimated Quantile = 2.5936
95% family-wise confidence level

Linear Hypotheses:

	Estimate	lwr	upr
1958 - 1954 == 0	0.1872	-3.2713	3.6457

```

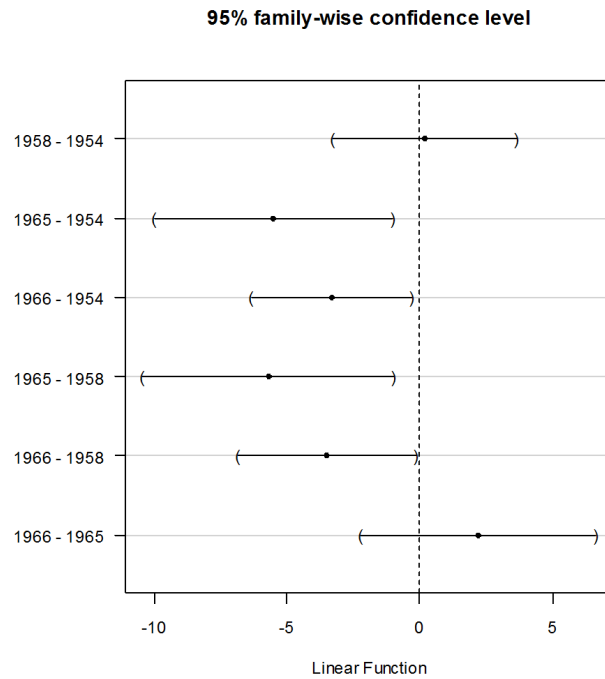
1965 - 1954 == 0 -5.5077 -9.9973 -1.0180
1966 - 1954 == 0 -3.3127 -6.3432 -0.2822
1965 - 1958 == 0 -5.6949 -10.4114 -0.9783
1966 - 1958 == 0 -3.4999 -6.8574 -0.1424
1966 - 1965 == 0  2.1950 -2.2173  6.6073

```

```

old.par <- par(mai = c(1, 1.25, 1, 1))
plot(meandiff)
par(old.par)

```

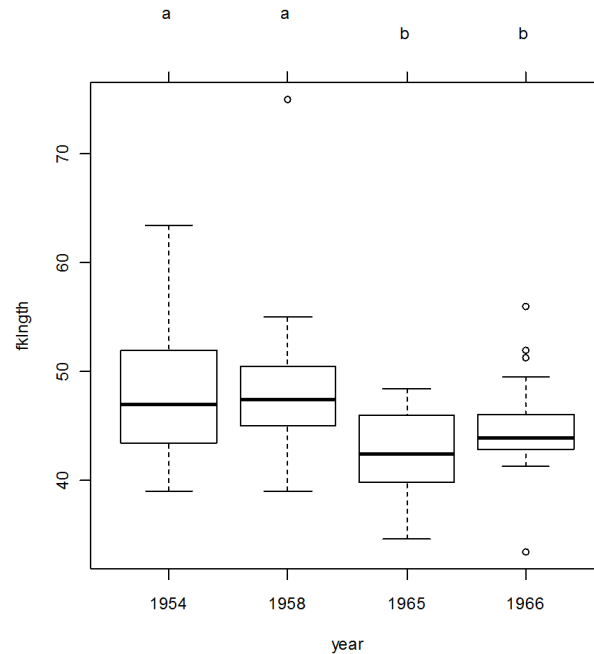


This is better. Also useful is a plot the means and their confidence intervals with the Tukey groupings shown as letters above:

```

# Compute and plot means and Tukey results
means <- glht(anova.fkl.vs.year, linfct = mcp(year =
"Tukey"))
cimeans <- cld(means)
# use sufficiently large upper margin
old.par <- par(mai = c(1, 1, 1.25, 1))
# plot
plot(cimeans)
par(old.par)

```



Note the letters appearing on top. Years labelled with the same letter do not differ significantly.

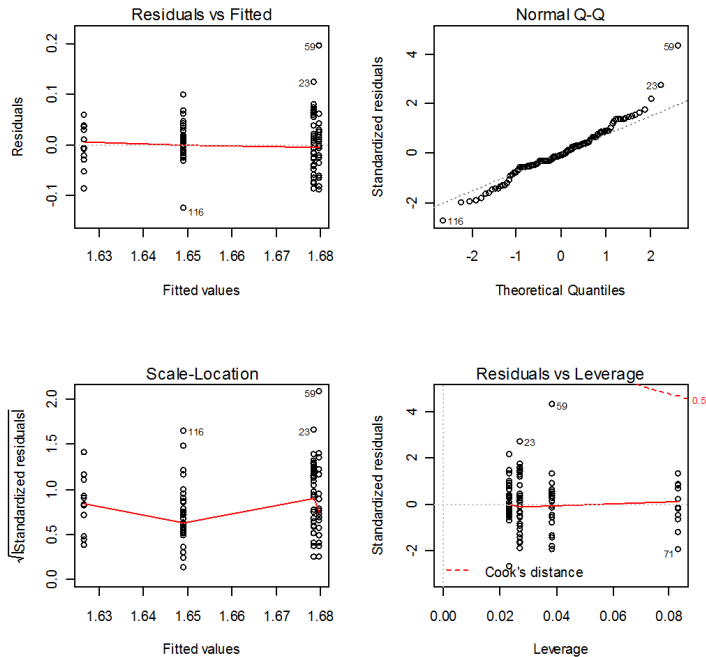
Data transformations and non-parametric ANOVA

In the above example to examine differences in `fklngth` among years, we detected evidence of non-normality and variance heterogeneity. If the assumptions underlying a parametric ANOVA are not valid, there are several options: (1) if sample sizes in each group are reasonably large, parametric ANOVA is reasonably robust with respect to the normality assumption, for the same reason that the t-test is, so the results are probably not too bad; (2) we can transform the data; (3) we can go the non-parametric route.

☞ Repeat the one-way ANOVA in the section above, but this time run the analysis on the \log_{10} `fklngth`. With this transformation, do some of the problems encountered previously disappear?

```
# Fit anova model on log10 of fklngth and plot residual diagnostics
opar <- par(mfrow = c(2, 2))
anova.model2 <- lm(log10(fklngth) ~ year, data=Dan10dat)
plot(anova.model2)
par(opar)
```

Looking at the residuals, we have:




Things look barely better than before. Running the Wilks-Shapiro test for normality on the residuals, we get:

```
Shapiro-Wilk normality test
data: residuals(anova.model.2)
W = 0.962, p-value = 0.002048
```

Running Levene's test on the absolute value of the residuals we get:

```
Levene's Test for Homogeneity of Variance
  Df F value Pr(>F)
group 3  2.6611 0.05148 .
    114
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, it would appear that we still have some problems with the assumption of normality and are just on the border line of meeting the assumption of homogeneity of variances. You have several choices here: 1) try to find a different transformation to satisfy the assumptions, 2) assume the data are close enough to meeting the assumptions, or 3) perform a non-parametric ANOVA.

 The most commonly used non-parametric analog of the parametric one-way ANOVA is the Kruskal-Wallis one-way ANOVA. Perform a Kruskal-Wallis one-way ANOVA of `length`, and compare these results to the parametric analysis above. What do you conclude?

```
kruskal.test(fklength ~ year, data=Dam10dat)
```

```
      Kruskal-Wallis rank sum test

data:  fklength and year
Kruskal-Wallis chi-squared = 15.7309, df = 3, p-value = 0.001288
```

So, the conclusion is the same as with the parametric ANOVA: we reject the null that the mean rank is the same for each year. Thus, despite violation of one or more assumptions, the parametric analysis is telling us the same thing as the non-parametric analysis: the conclusion is, therefore, quite robust.

Dealing with outliers

Our preliminary analysis of the relationship between `FLNGTH` and `Year` suggested there might be some outliers in the data. These were evident in the box plots of `FKLENGTH` by `YEAR` and flagged as cases 59, 23 and 87 in the residual probability plot and residual-fit plot. In general, you have to have very good reasons for removing outliers from a data set (e.g., you know there was a mistake made in the data collection/entry). However, it is often useful to know how the analysis changes if you remove the outliers from the data set.

- Repeat the original ANOVA of `FKLENGTH` by `YEAR` but work with a subset of the data without the outliers.

Have any of the conclusions changed?

```
Damsubset<-Dam10dat[-c(23,59,87),] #removes obs 23, 59 and 87
aov.Damsubset <- aov(fklength ~ as.factor(year), Damsubset)
summary(aov.Damsubset)
```

```
              Df Sum Sq Mean Sq F value Pr(>F)
as.factor(year)  3  367.51  122.50  6.8942 0.000267 ***
Residuals      111 1972.36   17.77
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
shapiro.test(residuals(aov.Damsubset))
```

```
      Shapiro-Wilk normality test

data:  residuals(aov.Damsubset)
W = 0.9853, p-value = 0.2448
```

```
leveneTest(log10(fklength) ~ year, Damsubset)
Levene's Test for Homogeneity of Variance
      Df F value Pr(>F)
group  3  3.8144 0.01206 *
```

```
111
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Elimination of three outliers, in this case, makes things better in terms of the normality assumption, but does not improve the variances. Moreover, the fact that the conclusion drawn from the original ANOVA with outliers retained does not change upon their removal reinforces the fact that there is no good reason to remove the points..

Instead of a Kruskal-Wallis rank-based test, a permutation test could be used (see supplied R script).

Permutation test

This is a more complex way of doing a permutation that we used when `lmPerm` was not available.

R commands for permutation test in one-way ANOVA

```
#####
# Permutation Test for one-way ANOVA
# modified from code written by David C. Howell
# http://www.uvm.edu/~dhowell/StatPages/Mbr_e_Stuff/Permutation%20Anova/PermutationAnova.htm

# set desired number of permutations
nrEPS <- 500
# to simplify reuse of this code, copy desired dataframe to mydata
mydata <- Dam0dat
# copy model formula to myformula
myformula <- as.formula("fklngth ~ year")
# copy dependent variable vector to mydep
mydep <- mydata$fklngth
# copy independent variable vector to myindep
myindep <- as.factor(mydata$year)

#####
# You should not need to modify code chunk below
#####
# Compute observed F value for original sample
mod1 <- lm(myformula, data=mydata) # Standard Anova
ANOVA <- summary(aov(mod1)) # Save summary to variable
observedF <- ANOVA[[1]]$Fvalue[1] # Save observed F value

# Print standard ANOVA results
cat(" The standard ANOVA for these data follows ",
    "\n")
print(ANOVA, "\n")
cat("\n")
cat("\n")

print("Resampling as in Manly with unrestricted sampling of observations. ")
# Now start resampling

Fboot <- numeric(nrEPS) # initialize vector to receive permuted values
Fboot[1] <- observedF
for (i in 2:nrEPS) {
  newdependent <- sample(mydep, length(mydep)) # randomize dep var
  mod2 <- lm(newdependent ~ myindep) # refit model
  b <- summary(aov(mod2))
  Fboot[i] <- b[[1]]$Fvalue[1] # store F stats
}
permprob <- length(Fboot[Fboot >= observedF]) / nrEPS
cat(" The permutation probability value is: ", permprob,
    "\n")

# end of code chunk for permutation
```

lmPerm version of permutation test

```
## lmPerm version of permutation test
require(lmPerm2)
# for generality, copy desired dataframe to mydata
# and model formula to myformula
mydata<- Dam10dat
myformula<- as.formula("fklngth ~ year")

# Fit desired model on the desired dataframe
mymodel <- lm(myformula, data=mydata)
# Calculate permutation p-value
anova(lm(myformula, data=mydata, perm = "Prob", center=FALSE,
Ca=0.001))
```

Lab- Multiway ANOVA: factorial and nested designs

After completing this laboratory exercise, you should be able to:

- Use R to do parametric ANOVAs for 2-way factorial designs with replication.
- Use R to do 2-way factorial design ANOVA without replication
- Use R to do parametric ANOVAs for nested designs with replication.
- Use R to do non-parametric 2-way ANOVAs
- Use R to do multiway pairwise comparisons

Be aware that there are a large number of possible ANOVA designs, many of which can be handled by R: this laboratory is designed only to introduce you to some of the more common ones.

Two-way factorial design with replication

Many experiments are designed to investigate the joint effects of several different factors: in a two-way ANOVA, we examine the effect of two factors, but in principle the analysis can be extended to three, four or even five factors, although interpreting the results from 4- and 5-way ANOVAs can be very difficult.

Suppose that we are interested in the effects of two factors: `location` (Cumberland House and The Pas) and `sex` (male or female) on sturgeon size (data can be found in `Stu2wdat.Rdata`). Note that because the sample sizes are not the same for each group, this is an unbalanced design. Note also that there are missing data for some of the variables, meaning that not every measurement was made on every fish.

Fixed effects ANOVA (Model I)


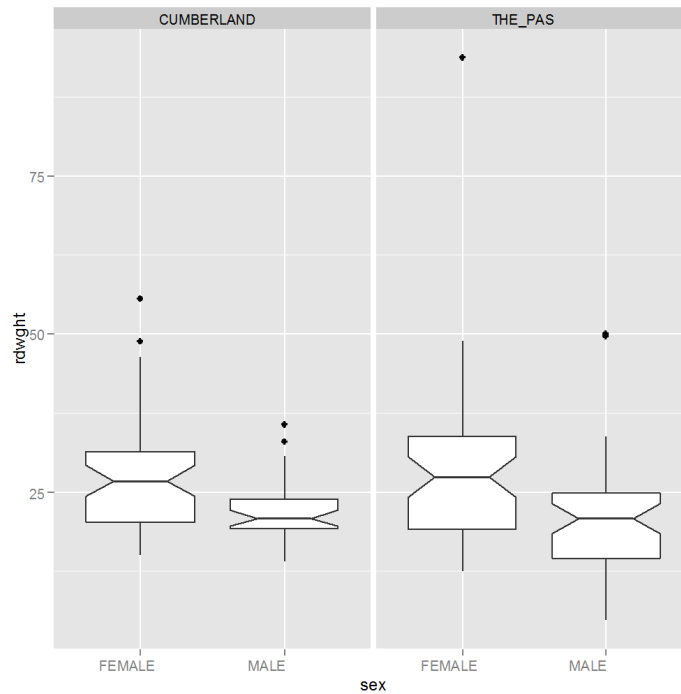

 Begin by having a look at the data by generating box plots of `rdwght` for `sex` and `location` from the file `Stu2wdat.Rdata`.

Figure 1.




From this, it appears as though females might be larger at both locations. It's difficult to get an idea of whether fish differ in size between the two locations. The presence of outliers on these plots suggests there might be problems meeting normality assumptions for the residuals.

 Generate summary statistics for `rdwght` by sex and location.

```
aggregate(rdwght ~sex+location, data=Stu2wdat, FUN = "summary")
  sex      location rdwght.M n. rdwght.1st Qu. rdwght.Median rdwght.3rd Qu. rdwght.Max.
1 FEMALE  CUMBERLAND  15.10    20.40    26.80    27.37
31.40    55.60
2 MALE    CUMBERLAND  14.00    19.22    20.85    22.14
23.90    35.60
3 FEMALE  THE_PAS     12.54    19.14    27.39    27.98
33.88    93.72
4 MALE    THE_PAS     4.73    14.63    20.79    20.65
24.94    49.94
>
```

The summary statistics confirm our interpretation of the box plots: females appear to be larger than males, and differences in fish size between locations are small.

 Using the file `Stu2wdat.Rdata`, do a two-way factorial ANOVA:

```
# Fit anova model and plot residual diagnostics
# but first, save current par and set graphic page to hold 4
graphs
```

```

opar <- par(mfrow = c(2, 2))

anova.model1 <- lm(rdwght ~ sex + location + sex:location,
  contrasts = list(sex = contr.sum, location = contr.sum),
  data = Stu2wdat)
anova(anova.model1)

```

Analysis of Variance Table

```

Response: rdwght
          Df Sum Sq Mean Sq F value    Pr(>F)
sex         1  1839.6   1839.6  18.6785 2.569e-05 ***
location    1     4.3     4.3   0.0433  0.8355
sex:location 1    48.7    48.7   0.4944  0.4829
Residuals  178 17530.4    98.5
---
Sgnif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Trap. Be careful here. R gives you the sequential sums of squares (Type I) and associated Mean squares and probabilities. These are **not** to be trusted unless the design is perfectly balanced. In this case, there are varying numbers of observations across sex and location combinations and therefore the design is not balanced.

What you want are the partial sums of squares (type III). The easiest way to get them is to use the `Anova()` function in the `car` package (note the subtle difference, `Anova()` is not the same as `anova()`, remember case matters in R.). However, this is not enough by itself. To get the proper values for the type III sums of square, one also needs to specify contrasts, hence the cryptic `contrasts = list(sex = contr.sum, location = contr.sum)` above...

```

require(car)
Anova(anova.model1, type = 3)

```

Anova Table (Type III tests)

```

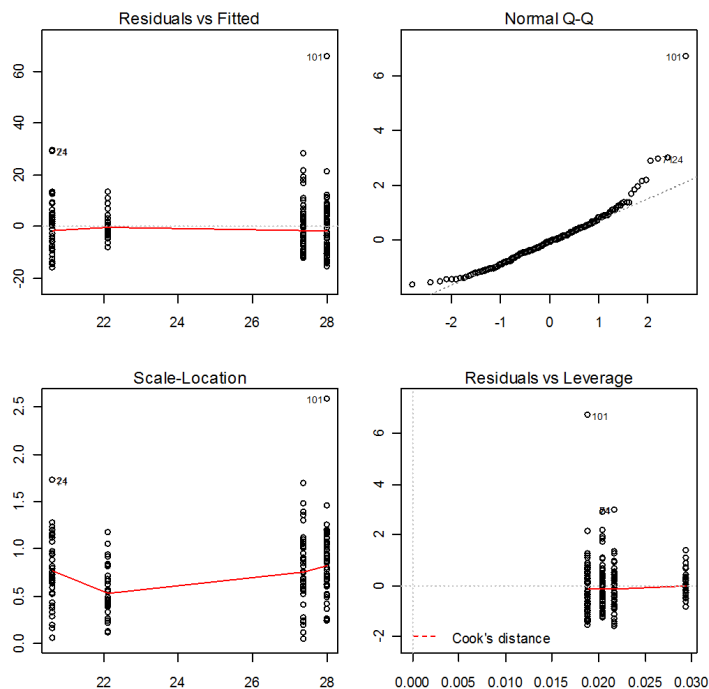
Response: rdwght
          Sum Sq Df F value    Pr(>F)
(Intercept) 106507 1 1081.4552 < 2.2e-16 ***
sex           1745  1  17.7220 4.051e-05 ***
location      9   1  0.0891  0.7656
sex:location  49   1  0.4944  0.4829
Residuals   17530 178
---
Sgnif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

On the basis of the ANOVA, there is no reason to reject two null hypotheses: (1) that the effect of `sex` (if any) does not depend on `location` (no interaction), and (2) that there is no difference in the size of sturgeon (pooled over `sex`) between the two `locations`. On the other hand, we reject the null hypothesis that there is no difference in size between male and female sturgeon (pooled over `location`), precisely as expected from the graphs.

```
plot(anova.model1)
```

Figure 2.



As usual, we cannot accept the above results without first ensuring that the assumptions of ANOVA are met. Examination of the residuals plots above shows that the residuals are reasonably normally distributed, with the exception of three potential outliers flagged on the QQ plot (cases 101, 24, & 71; the latter two are on top of one another). However, Cook's distances are not large for these (the 0.5 contour is not even visible on the plot), so there is little indication that these are a concern. The residuals vs fit plot shows that the spread of residuals is about equal over the range of the fitted values, again with the exception of a few cases. When we test for normality of residuals we get:

```
shapiro.test(residuals(anova.model1))
```

```
Shapiro-Wilk normality test
```

```
data: residuals(anova.model1)
W = 0.8721, p-value = 2.619e-11
```

So, there is evidence of non-normality in the residuals.

We will use the Levene's test to examine the assumption of homogeneity of variances, just as we did with the 1-way anova.

```
require(car)
```

```
leveneTest(rdwght ~ sex * location, data = Stu2wdat)
```


```
Levene's Test for Homogeneity of Variance
```

```
  Df F value Pr(>F)
group 3  3.8526 0.01055 *
 178
```

```
---
```

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

If the assumption of homogeneity of variances was valid, we would be accepting the null that the mean of the absolute values of residuals does not vary among levels of `sex` and `location` (i.e., `group`). The above table shows that the hypothesis is rejected and we conclude there is evidence of heteroscedasticity. All in all, there is some evidence that several important assumptions have been violated. However, whether these violations are sufficiently large to invalidate our conclusions remains to be seen.


 Repeat this procedure using the data file `Stu2mdat.Rdata`. Now what do you conclude? Suppose you wanted to compare the sizes of males and females: in what way would these comparisons differ between `Stu2wdat.Rdata` and `Stu2mdat.Rdata`?

Note that in this case, we see that at Cumberland House, females are larger than males, whereas the opposite is true in The Pas (you can confirm this observation by generating summary statistics). What happens with the ANOVA (remember, you want Type III sum of squares)?

Anova Table (Type III tests)

```
Response: rdwght
              Sum Sq Df F value    Pr(>F)
(Intercept) 106507   1 1081.4552 < 2.2e-16 ***
sex           49     1   0.4944    0.4829
location      9     1   0.0891    0.7656
sex:location 1745    1  17.7220 4.051e-05 ***
Residuals    17530 178
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case, the interaction term `sex:location` is significant but the main effects are not significant.

 You might find it useful here to generate plots for the two data files to compare the interactions between `sex` and `location`. The effect plot shows the relationship between means for each combination of factors (also called cell means). Generate an effect plot for the two models using the `allEffects` command from the `effects` library:

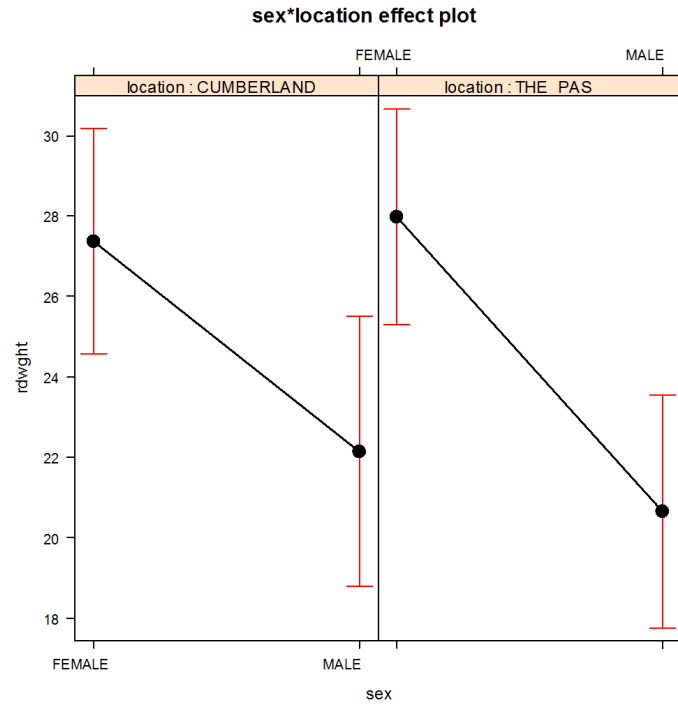
```
library(effects)
allEffects(anova.model1)
model: rdwght ~ sex + location + sex:location

sex*location effect
      location
sex    CUMBERLAND  THE_PAS
```

FEMALE	27.37347	27.97717
MALE	22.14118	20.64652

```
plot(allEffects(anova.model1), 'sex:location')
```

Figure 3.



```
allEffects(anova.model2)
```

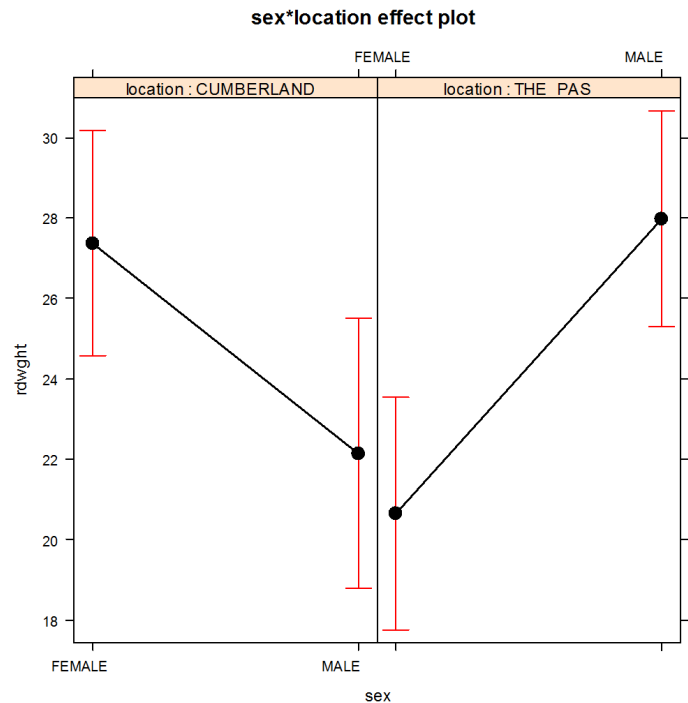
```
model: rdwght ~ sex + location + sex:location
```

```
sex*location effect
```

sex	location	
	CUMBERLAND	THE_PAS
FEMALE	27.37347	20.64652
MALE	22.14118	27.97717

```
plot(allEffects(anova.model2), 'sex:location')
```

Figure 4.



There is a very large difference between the results from `Stu2wdat` and `Stu2mdat`. In the former case, because there is no significant interaction, we can essentially pool over the levels of factor 1 (`sex`, say) to test for the effects of `location`, or over the levels of factor 2 (`location`) to test for the effects of `sex`. In fact, if we do so and simply run a one-way ANOVA on the `Stu2wdat` data with `sex` as the grouping variable, we get:

```
Anova(aov(rdwght ~ sex, data = Stu2wdat), type = 3)
Anova Table (Type III tests)
```

Response: rdwght

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	78191	1	800.440	< 2.2e-16 ***
sex	1840	1	18.831	2.377e-05 ***
Residuals	17583	180		

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

Note that here the residual sum of squares (17583) is only slightly higher than for the 2-way model (17530), simply because, in the 2-way model, only a small fraction of the explained sums of squares is due to the `location` main effect or the `sex:LOCATION` interaction.

On the other hand, if you try the same trick with `stu2mdat`, you get:

```
Anova(aov(rdwght ~ sex, data = Stu2mdat), type = 3)
Anova Table (Type III tests)
```

```

Response: rdwght
      Sum Sq Df F value Pr(>F)
(Intercept) 55251 1 515.0435 <2e-16 ***
sex          113 1  1.0571 0.3053
Residuals   19309 180
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


```

Here, the residuals sum of squares (19309) is much larger than in the 2-way model (17530), because most of the explained sums of squares is due to the interaction. Note that if we did this, we would conclude that male and female sturgeons don't differ in size. But in fact they do: it's just that the difference is in different directions, depending on location. This is why it is always dangerous to try and make too much of main effects in the presence of interactions!

Mixed effects ANOVA (Model III)

We have neglected an important component in the above analyses, and that is related to the type of ANOVA model we wish to run. In this example, `Location` could be considered a random effect, whereas `sex` is a fixed effect (because it is "fixed" biologically), and so this model could be treated as a mixed model (Model III) ANOVA.

Note that in these analyses, R treats analyses by default as Model I ANOVA, so that the main effects and the interaction are tested over the residuals mean square. Recall, however, that in a Model III ANOVA, main effects are tested over the interaction mean square or the pooled interaction mean square and residual mean square (depending on which statistician you consult!)

 Working with the `Stu2wdat` data, rebuild the ANOVA table for `rdwght` for the situation in which `location` is a random factor and `sex` is a fixed factor. To do this, you need to recalculate the F-ratios for the `sex` and `location` terms using the `sex:location` interaction mean square instead of the residual mean square. This is most easily accomplished by hand, making sure you are working with the Type III Sums of squares ANOVA table.


Anova Table (Type III tests)

```

Response: rdwght
      Sum Sq Df F value Pr(>F)
(Intercept) 106507 1 1081.4552 < 2.2e-16 ***
sex          1745 1  17.7220 4.051e-05 ***
location     9 1  0.0891 0.7656
sex:location 49 1  0.4944 0.4829
Residuals   17530 178
---

```

For `sex`, the new ratio of mean squares is $(1745/1)/(49/1)=35.6$

 To assign a probability to the new F-value, enter the following in the commands window: `1-pf(F, df1, df2)`, where `F` is the newly calculated F-value, and `df1` and `df2` are the degrees of freedom of the numerator (`sex`) and denominator (`SEX:location`), respectively.

```
1-pf(35.6, 1, 1)
[1] 0.1057152
```


Note that the p value for `sex` is now non-significant. This is because the error MS of the initial ANOVA is smaller than the interaction MS, but mostly because the number of degrees of freedom of the denominator of the F test has dropped from 178 to 1. In general, a drop in the denominator degrees of freedom makes it much more difficult to reach significance.

2-way factorial ANOVA without replication

In some experimental designs, there are no replicates within data cells: perhaps it is simply too expensive to obtain more than one datum per cell. A 2-way ANOVA is still possible under these circumstances, but there is an important limitation.

Because there is no replication within cells, there is no error variance: we have simply a row sum of squares, a column sum of squares, and a remainder sum of squares. This has important implications: if there is an interaction in a Model III ANOVA, only the fixed effect can be tested (over the remainder MS); for Model I ANOVAs, or for random effects in Model III ANOVAs, it is not appropriate to test main effects over the remainder unless we are sure there is no interaction.

A limnologist studying Round Lake in Algonquin Park takes a single temperature (`temp`) reading at 10 different depths (`depth`, in m) at four times (`date`) over the course of the summer. Her data are shown in `Nr2wdat.Rdata`.

 Do a two-way unreplicated ANOVA using `temp` as the dependent variable, `date` and `depth` as the factor variables (you will need to recode `depth` to tell R to treat this variable as a factor). Note that there is no interaction term included in this model.

```
anova.model4<-lm(temp ~ date + as.factor(depth), data =
nr2wdat)
Anova(anova.model4, type = 3)
```

Anova Table (Type III tests)

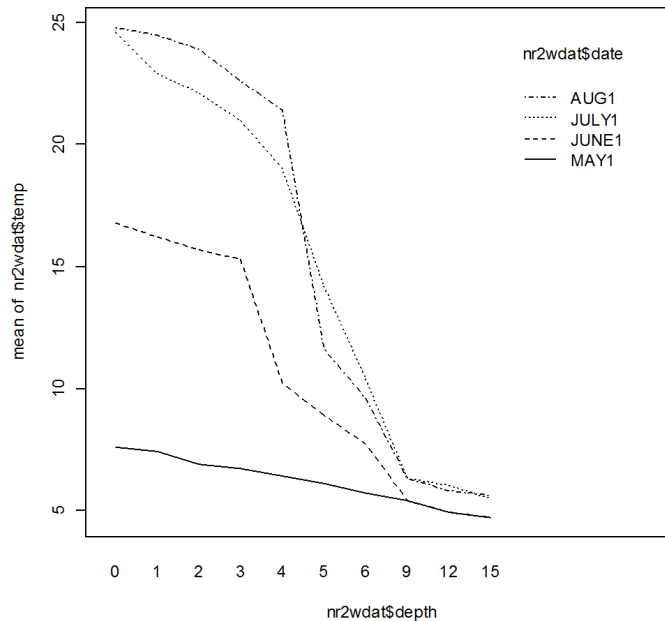
```
Response: temp
          Sum Sq Df F value Pr(>F)
```

```
(Intercept)    1511.99    1 125.5652 1.170e-11 ***
date           591.15    3 16.3641 2.935e-06 ***
as.factor(depth) 1082.82    9  9.9916 1.450e-06 ***
Residuals      325.12   27
```

Assuming that this is a Model III ANOVA (date random, depth fixed), what do you conclude? (Hint: you may want to generate an interaction plot of temp versus depth and month, just to see what's going on.)

```
interaction.plot(nr2wdat$depth, nr2wdat$date, nr2wdat$temp)
```

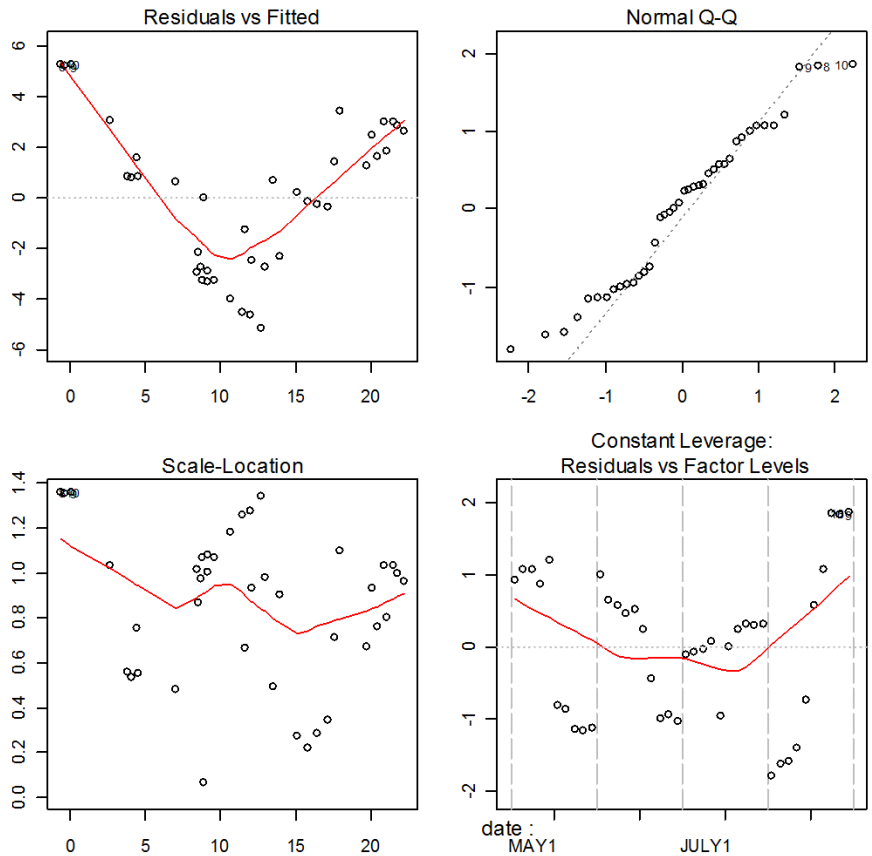
Figure 5.



There is a highly significant decrease in temperature as depth increases. To test the effect of month (the (assumed) random factor), we must assume that there is no interaction between depth and month, i.e. that the change in temperature with depth is the same for each month. This is a dubious assumption: if you plot temperature against depth for each month, you should see that the temperature profile becomes increasingly non-linear as the summer progresses (i.e. the thermocline develops), from almost a linear decline in early spring to what amounts to a step decline in August. In other words, the relationship between temperature and depth does change with month, so that if you were to use the above fitted model to estimate, say, the temperature at a depth of 5 m in July, you would not get a particularly good estimate.

In terms of residual diagnostics, have a look at the residuals probability plot and residuals vs fitted values plot.

Figure 6.




Testing the residuals for normality, we get $p=0.16$, so that the normality assumption seems to be O.K. In terms of heteroscedasticity, we can only test among months, using depths as replicates (or among depths using months as replicates). Using depths as replicates within months, we find

```
leveneTest(temp ~ date , data = nr2wdat)
Levene's Test for Homogeneity of Variance
  Df F value Pr(>F)
group 3  17.979 2.679e-07 ***
```

So there seems to be some problem here, as can be plainly seen in the above plot of residuals vs fit. All in all, this analysis is not very satisfactory: there appears to be some problems with the assumptions, and the assumption of no interaction between depth and date would appear to be invalid.

Nested designs

A common experimental design occurs when each major group (or treatment) is divided into randomly chosen subgroups. For example, a geneticist interested in the effects of genotype on desiccation resistance in fruit flies might conduct an experiment with larvae of three different genotypes. For each genotype (major group), she sets up three environmental chambers (sub-groups, replicates within groups) with a fixed temperature humidity regime, and in each chamber, she has five larvae for which she records the number of hours each larvae survived.

 The file `Nestdat.Rdata` contains the results of just such an experiment. The file lists three variables: `genotype`, `chamber` and `survival`. Run a nested ANOVA with `survival` as the dependent variable, `genotype/chamber` as the independent variables (this is the shorthand notation for a chamber effect nested under genotype).

```
nestdat$chamber <- as.factor(nestdat$chamber)
anova.nested <- lm(survival ~ genotype/chamber, data = nestdat)
```

What do you conclude from this analysis? What analysis would (should) you do next? (Hint: if there is a non-significant effect of chambers within genotypes, then you can increase the power of between-genotype comparisons by pooling over chambers within genotypes, although not everyone (Dr. Rundle included) agrees with such pooling.) Do it! Make sure you check your assumptions!

Analysis of Variance Table

Response: survival

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
genotype	2	2952.22	1476.11	292.6081	<2e-16 ***
genotype:chamber	6	40.65	6.78	1.3432	0.2639
Residuals	36	181.61	5.04		

We conclude from this analysis that there is no (significant) variation among chambers within genotypes, but that the null hypothesis that all genotypes have the same desiccation resistance (as measured by survival) is rejected (Test of genotype using MS genotype:chamber as denominator: $F = 1476.11/6.78 = 217.7153$, $P < 0.0001$). In other words, genotypes differ in their survival.

Since the chambers within genotypes effect is non-significant, we may want to pool over chambers to increase our degrees of freedom:

```
anova.simple <- lm(survival ~ genotype, data = nestdat)
anova(anova.simple)
```

Analysis of Variance Table

Response: survival

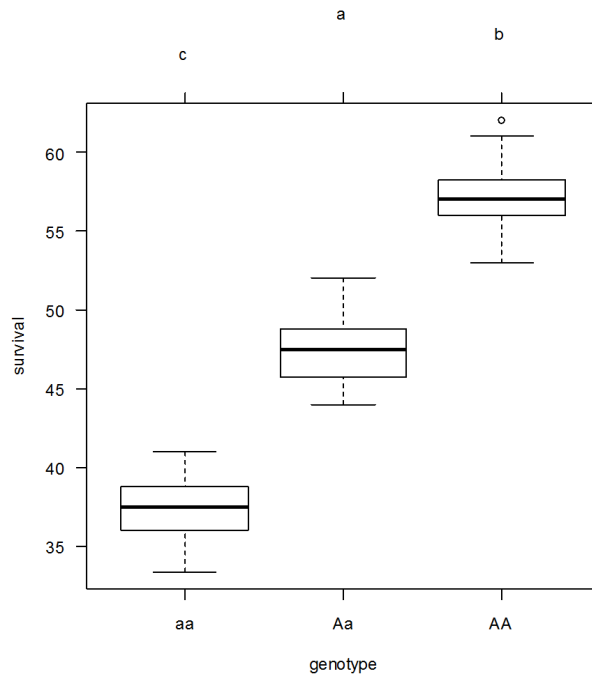
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
genotype	2	2952.22	1476.11	278.93	< 2.2e-16 ***
Residuals	42	222.26	5.29		

Thus, we conclude that there is significant variation among the three genotypes in desiccation resistance.

A box plot of survival across genotypes shows clearly that there is significant variation among the three genotypes in desiccation resistance. This can be combined with a formal Tukey multiple comparison test:

```
par(mfrow=c(1,1))
# Compute and plot means and Tukey CI
means <- glht(anova.simple, linfct = mcp(genotype =
"Tukey"))
cimeans <- cld(means)
# use sufficiently large upper margin
old.par <- par(mai = c(1, 1, 1.25, 1))
# plot
plot(cimeans, las=1) # las option to put y-axis labels
```

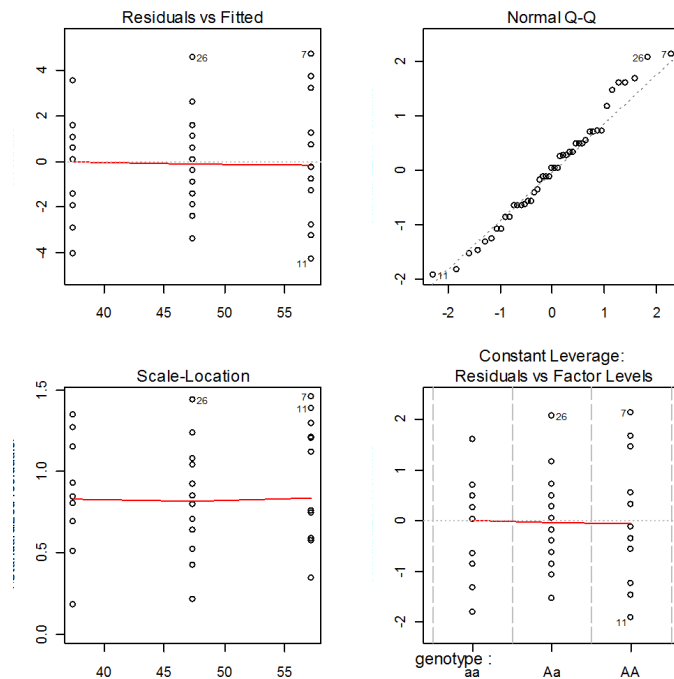
Figure 7.



So, we conclude from the Tukey analysis and plot that desiccation resistance (R), as measured by larval survival under hot, dry conditions, varies significantly among all three genotypes with $R(AA) > R(Aa) > R(aa)$.

Before concluding this, however, we must test the assumptions. Here are the residual plots and diagnostics for the one-way (unnested) design:


Figure 8.



So, all the assumptions appear to be valid, and the conclusion reached above still holds. Note that if you compare the residual mean squares of the nested and one-way ANOVAs (5.04 vs 5.29), they are almost identical. This is not surprising, given the small contribution of the chamber `in` genotype effect to the explained sum of squares.

Two-way non-parametric ANOVA

Two-way non-parametric ANOVA is an extension of the non-parametric one-way methods discussed previously. The basic procedure is to rank all the data in the sample from smallest to largest, then carry out a 2-way ANOVA on the ranks. This can be done either for replicated or unreplicated data:

 Using the data file `Stu2wdat.Rdata`, do a two-factor ANOVA to examine the effects of `sex` and `location` on `rank(rdwght)`.

```
aov.rank <- aov(rank(rdwght) ~ sex * location,
  contrasts = list(sex = contr.sum,
    location = contr.sum), data = Stu2wdat)
Anova(aov.rank, type = 3)
```


Anova Table (Type III tests)

```

Response: rank(rdwght)
          Sum Sq Df F value    Pr(>F)
(Intercept) 1499862  1 577.8673 < 2.2e-16 ***
sex          58394  1 22.4979 4.237e-06 ***
location     1128  1  0.4347  0.5105
sex:location 1230  1  0.4738  0.4921
Residuals   472383 182

```

The Scheirer-Ray-Hare extension of the Kruskal-Wallis test is done by computing a statistic H given by the effect sums of squares (SS) divided by the total MS. The latter can be calculated as the variance of the ranks.. We compute an H statistic for each term. The H -statistics are then compared to a theoretical χ^2 distribution using the command line: `1-pchisq(H, df)`, where H and df are the calculated H -statistics and associated degrees of freedom, respectively.

 Use the ANOVA table based on ranks to test the effects of `sex` and `location` on `rdwght`. What do you conclude? How does this result compare with the result obtained with the parametric 2-way ANOVA done before?

To calculate the Scheirer-Ray-Hare extension to the Kruskal-Wallis test, you must first calculate the total mean square (MS), i.e. the variance of the ranked data. In this case, there are 186 observations, their ranks are therefore the series 1, 2, 3, ..., 186. The variance can be calculated simply as `var(1:186)` (Isn't R neat? Cryptic maybe, but neat.)

So we can compute the H statistic for each term:

$$H_{\text{sex}} = 58394/\text{var}(1:186) = 20.14$$

$$H_{\text{location}} = 1128/\text{var}(1:186) = 0.39$$

$$H_{\text{sex:location}} = 1230/\text{var}(1:186) = 0.42$$

And convert these statistics into p-values:

$$\text{for sex: } 1\text{-pchisq}(20.14, 1) = 7.197556\text{e-}06$$

$$\text{location } 1\text{-pchisq}(0.39, 1) = 0.5322994$$

$$\text{sex:location } 1\text{-pchisq}(0.42, 1) = 0.516937$$


Note that these results are the same as those obtained in our original two-way parametric ANOVA. Despite the reduced power, we still find significant differences between the sexes, but still no interaction and no effect due to location.

There is, however, an important difference. Recall that in the original parametric ANOVA, there was a significant effect of `sex` when we considered the problem as a Model I ANOVA. However, if we consider it as Model III, the significant `sex` effect could in principle disappear, because the df associated with the interaction MS are much smaller than the df associated with the Model I error MS. In this case, however, the interaction MS is about half that of the error MS. So, the significant `sex` effect becomes even more significant if we analyze the problem as a Model III ANOVA. Once again, we see the importance of specifying the appropriate ANOVA design.

Multiple comparisons

Further hypothesis testing in multiway ANOVAs depends critically on the outcome of the initial ANOVA. If you are interested in comparing groups of marginal means (that is, means of treatments for one factor pooled over levels of the other factor, e.g., between male and female sturgeon pooled over location), this can be done exactly as outlined for multiple comparisons for one-way ANOVAs. For comparison of individual cell means, you must specify the interaction as the group variable.

The file `wmdat2.Rdata` shows measured oxygen consumption (`o2cons`) of two species (`species = A, B`) of limpets at three different concentrations of seawater (`conc = 100, 75, 50%`) taken from Sokal and Rohlf, 1995, p. 332.

 Run a 2-way factorial ANOVA on `wmdat2` data, using `o2cons` as the dependent variable and `species` and `conc` as the factors. What do you conclude?

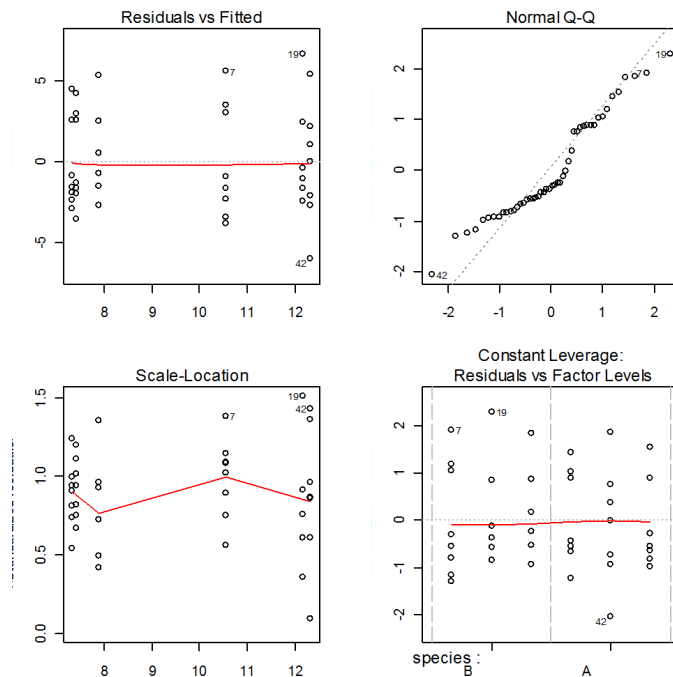
The ANOVA table is shown below. Technically, because the sample sizes in individual cells are rather small, this analysis should be repeated using a non-parametric ANOVA. For the moment, let's stick with the parametric analysis.

Anova Table (Type III tests)

```
Response: o2cons
          Sum Sq Df F value    Pr(>F)
(Intercept) 4441.7  1 464.6163 < 2.2e-16 ***
species       16.6  1   1.7404  0.1942381
conc         181.3  2   9.4833  0.003993 ***
species:conc   23.9  2   1.2514  0.2965616
Residuals    401.5 42
```

Look at the diagnostic plots:

Figure 9.



Homoscedasticity looks ok, but normality less so.. Testing for normality, we get:

Shapiro-Wilk normality test

```
data: residuals(anova.model5)
W = 0.9369, p-value = 0.01238
```

So there is evidence of non-normality, but otherwise everything looks O.K. Since the ANOVA is relatively robust with respect to non-normality, we proceed, but if we wanted to reassure ourselves, we could run a non-parametric ANOVA, and get the same answer.

☞ On the basis of the ANOVA results obtained above, which means would you proceed to compare? Why?

Overall, we conclude that there are no differences among species, and that the effect of concentration does not depend on species (no interaction). Since there is no interaction and no main effect due to species, the only comparison of interest is among salinity concentrations:

```
# fit simplified model
anova.model6 <- lm(o2cons ~ conc, data = wmcdata2)
# Make Tukey multiple comparisons
TukeyHSD(anova.model6)
```

```

Tukey multiple comparisons of means
 95% family-wise confidence level

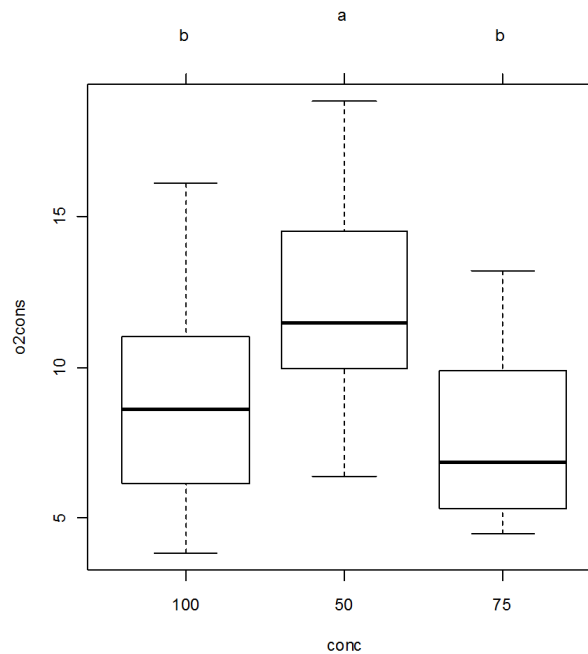
Fit: aov(formula = o2cons ~ conc, data = wmc2dat2)

$conc
      diff      lwr      upr    p adj
50-100  3.25500  0.5692518  5.940748 0.0141313
75-100 -1.38125 -4.0669982  1.304498 0.4325855
75-50  -4.63625 -7.3219982 -1.950502 0.0003793


par(mfrow=c(1,1))
# Graph of all comparisons for conc
tuk <- glht(anova.model6, linfct = mcp(conc = "Tukey"))
# extract information
tuk.cld <- cld(tuk)
# use sufficiently large upper margin
old.par <- par(mai = c(1, 1, 1.25, 1))
# plot
plot(tuk.cld)
par(old.par)

```

Figure 10.



So there is evidence of a significant difference in oxygen consumption at a reduction in salinity to 50% of regular seawater, but not at a reduction of only 25%.

 Repeat the analysis described above using `wmc2dat2.Rdata`. How do your results compare with those obtained for `wmc2dat2.Rdata`?

Using `wmc2dat2.Rdata`, we get:

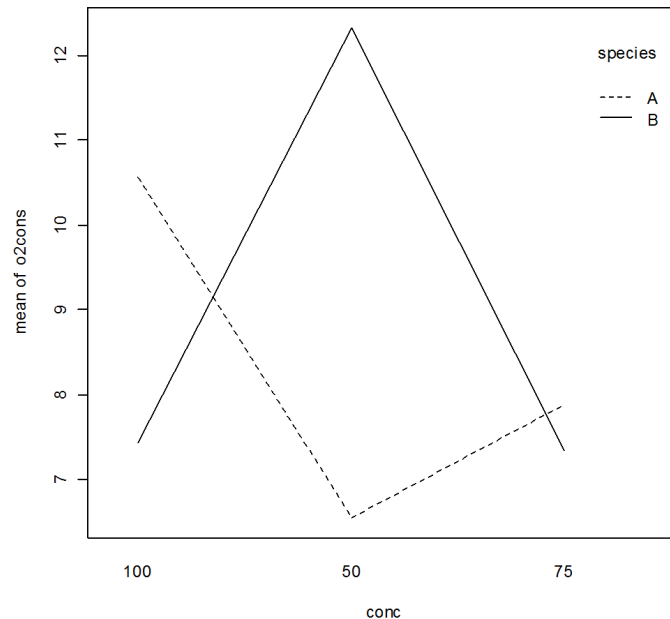
Anova Table (Type III tests)

```
Response: o2cons
              Sum Sq Df F value    Pr(>F)
(Intercept) 3618.2  1 381.8998 < 2.2e-16 ***
species      5.8    1  0.6162  0.4368642
conc        29.0    2  1.5283  0.2287237
species:conc 168.2  2  8.8742  0.0006101 ***
Residuals   397.9  42
```

Here there is a large interaction effect, and consequently, there is no point in comparing marginal means. This is made clear by examining an interaction plot:

```
with(wmc2dat2, interaction.plot(conc, species, o2cons))
```

Figure 11.



Working still with the `wmc2dat2.Rdata` data set, compare individual cell means (6 in all), with the **Bonferroni** adjustment. To do this, it is helpful to create a new variable to indicate all the combinations of species and conc:

```
wmc2dat2$species.conc <- paste(wmc2dat2$species,
                                wmc2dat2$conc)
```

Then we can conduct pairwise bonferroni comparisons:

```
with(wmc2dat2, pairwise.t.test(o2cons, species.conc,
                                p.adj = "bonf"))
```

Pairwise comparisons using t tests with pooled SD

data: o2cons and species.conc

	A 100	A 50	A 75	B 100	B 50
A 50	1.000	-	-	-	-
A 75	1.000	0.124	-	-	-
B 100	0.737	0.056	1.000	-	-
B 50	1.000	1.000	0.096	0.043	-
B 75	0.647	0.048	1.000	1.000	0.036

P value adjustment method: bonferroni

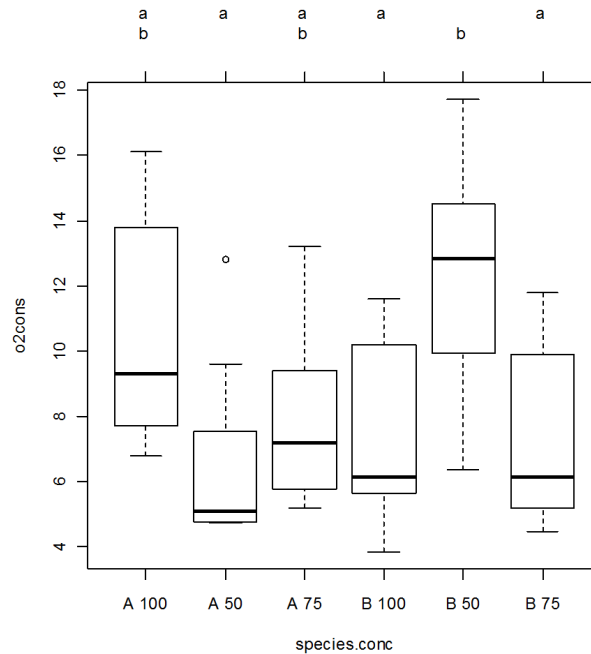
These comparisons are a little more difficult to interpret, but the analysis essentially examines for differences among seawater concentrations within species A and for differences among concentrations within species B. We see here that the `o2cons` at 50% seawater for species B is significantly different from that of 75% and 100% seawater for species B, whereas there are no significant differences in `o2cons` for species A across all seawater concentrations.

I find these outputs rather unsatisfying because they show only p-values, but no indication of effect size. One can get both the conclusion from the multiple comparison procedure and an indication of effect size from the graph produced with the following code:

```
# fit one-way anova comparing all combinations of species.conc combinations
anova.modelx <- aov(o2cons ~ species.conc, data = wmc2dat2)

tuk2 <- glht(anova.modelx, linfct = mcp(species.conc =
"Tukey"))
# extract information
tuk2.cld <- cld(tuk2)
# use sufficiently large upper margin
old.par <- par(mai = c(1, 1, 1.25, 1))
# plot
plot(tuk2.cld)
par(old.par)
```

Figure 12.



Note that in this analysis, we have used the error $MS = 9.474$ from the original model to contrast cell means. Recall, however, that this assumes that in fact we are dealing with a Model I ANOVA, which may or may not be the case (`conc` is certainly a fixed factor, but `species` might be either fixed or random).

Permutation test for two-way ANOVA

When data do not meet the assumptions of the parametric analysis in two- and multiway ANOVA, as an alternative to the non-parametric ANOVA, it is possible to run permutation tests to calculate p-values. The `lmPerm` package does this easily.

```
#####
## lmPerm version of permutation test
require(lmPerm)
# for generality, copy desired dataframe to mydata
# and model formula to myformula
mydata <- Stu2wdat
myformula <- as.formula("rdwght ~ sex+locati on+sex:locati on")

# Fit desired model on the desired dataframe
mymodel <- lm(myformula, data=mydata)
# Calculate permutation p-value
anova(lmPerm(myformula, data=mydata, perm = "Prob", center=FALSE, Ca=0.001))
```

lmPerm was orphaned for a while and the code below, while clunkier, provided an alternative way of doing it. You would have to adapt it for other situations.

```
#####
# Permutation test for two way ANOVA
# Ter Braak creates residuals from cell means and then permutes across
# all cells
# This can be accomplished by taking residuals from the full model
# modified from code written by David C. Howell
# http://www.uvm.edu/~dhowell/St at Pages/ Mbr e_ St uff / Per mut at i on % 20 Anova / Per m
TestsAnova. ht ml

nr eps = 500
dependent <- St u2wdat $r dwght
factor1 <- as. f act or ( St u2wdat $sex)
factor2 <- as. f act or ( St u2wdat $l ocat i on)

my. dat a f r a m e <- dat a. f r a m e (dependent , f act or 1 , f act or 2)
my. dat a f r a m e. noNA <- my. dat a f r a m e [compl et e. cases (my. dat a f r a m e) ,
]

mod <- lm (dependent ~ f act or 1 + f act or 2 + f act or 1 : f act or 2 ,
dat a = my. dat a f r a m e. noNA)
res <- mod $resi dual s
TBi nt <- numer i c (nr eps)
TB1 <- numer i c (nr eps)
TB2 <- numer i c (nr eps)

ANOVA <- summar y (aov (mod))
cat (" The standard ANOVA for these data follows ",
"\n")
F1 <- ANOVA [ [ 1 ] ] $ " F val ue" [ 1 ]
F2 <- ANOVA [ [ 1 ] ] $ " F val ue" [ 2 ]
Fi nt er act <- ANOVA [ [ 1 ] ] $ " F val ue" [ 3 ]
pri nt (ANOVA)
cat ("\n")
cat ("\n")

TBi nt [ 1 ] <- Fi nt er act
for (i in 2: nr eps) {
newdat <- sampl e (res , lengt h (res) , repl ace = FALSE)
modb <- summar y (aov (newdat ~ f act or 1 + f act or 2 + f act or 1 : f act or 2 ,
dat a = my. dat a f r a m e. noNA) )
TBi nt [ i ] <- modb [ [ 1 ] ] $ " F val ue" [ 3 ]
TB1 [ i ] <- modb [ [ 1 ] ] $ " F val ue" [ 1 ]
TB2 [ i ] <- modb [ [ 1 ] ] $ " F val ue" [ 2 ]
}
probl nt <- lengt h (TBi nt [ TBi nt >= Fi nt er act ]) / nr eps
prob1 <- lengt h (TB1 [ TB1 >= F1 ]) / nr eps
prob2 <- lengt h (TB2 [ TB2 >= F2 ]) / nr eps
cat ("\n")
cat ("\n")
pri nt (" Resampling as inter Braak with unrestricted sampling of cell residu-
al s. ")
cat (" The probability for the effect of Interaction is ",
probl nt , "\n")
cat (" The probability for the effect of Factor 1 is ",
prob1 , "\n")
cat (" The probability for the effect of Factor 2 is ",
prob2 , "\n")
```

Bootstrap for two-way ANOVA

In most cases, permutation tests will be more appropriate than bootstrap in ANOVA designs. However, for the sake of completeness, I have a snippet of code to do bootstrap for you:

```
#####

# Bootstrap for two-way ANOVA
# You possibly want to edit bootfunction.mod1 to return other values
# Here it returns the standard coefficients of the fitted model
```

```

# Requires boot library
#

nr eps = 5000
dependent <- St u2wdat $r dwght
factor 1 <- as.f actor (St u2wdat $sex)
factor 2 <- as.f actor (St u2wdat $l ocat ion)

my.dat a.f rame <- dat a.f rame(dependent, factor 1, factor 2)
my.dat a.f rame.noNA <- my.dat a.f rame[comple.t cases(my.dat a.f rame), ]

require(boot)
# Fit model on observed data
mod1 <- aov(dependent ~ factor 1 + factor 2 + factor 1:factor 2,
            dat a = my.dat a.f rame.noNA)

# Bootstrap 1000 time using the residuals bootstrapping methods to
# keep the same unequal number of observations for each level of the indep.
var.

fit <- fitted(mod1)
e <- residuals(mod1)
X <- model.matrix(mod1)

bootfunction.mod1 <- function(dat a, indices) {
  y <- fit + e[indices]
  boot mod <- lm(y ~ X)
  coefficients(boot mod)
}

boot results <- boot (my.dat a.f rame.noNA, bootfunction.mod1,
                    R = 1000)

boot results

# Calculate 90% CI and plot bootstrap estimates separately
# for each model parameter

boot.ci(boot results, conf = 0.9, index = 1)
plot(boot results, index = 1)

boot.ci(boot results, conf = 0.9, index = 3)
plot(boot results, index = 3)

boot.ci(boot results, conf = 0.9, index = 4)
plot(boot results, index = 4)

boot.ci(boot results, conf = 0.9, index = 5)
plot(boot results, index = 5)

```


Lab- Multiple regression

After completing this laboratory exercise, you should be able to:

- Use R to fit a multiple regression model, and compare the adequacy of several models using inferential and information theoretic criteria
- Use R to test hypotheses about the effects of different independent variables on the dependent variable of interest.
- Use R to evaluate multicollinearity among (supposedly) independent variables and its effects.
- Use R to do curvilinear (polynomial) regression.

Multiple regression: points to keep in mind

Multiple regression models are used in cases where there is one dependent variable and several independent, continuous variables. In many biological systems, the variable of interest may be influenced by several different factors, so that accurate description or prediction requires that several independent variables be included in the regression model.

Before beginning, be aware that multiple regression takes time to learn well. Beginners should keep in mind several important points:

1. An overall regression model may be statistically significant even if none of the individual regression coefficients in the model are (caused by multicollinearity);
2. A multiple regression model may be "nonsignificant" even though some of the individual coefficients are "significant" (caused by overfitting);
3. Unless "independent" variables are uncorrelated in the sample, different model selection procedures may yield different results.

The data: let's first have a look

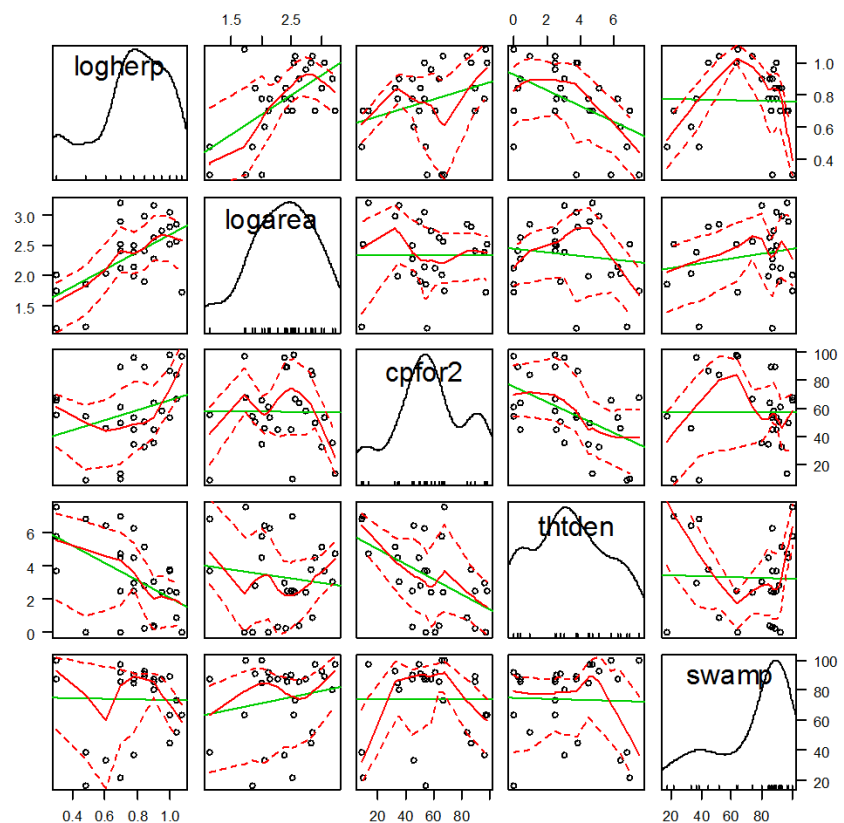
The file `Mregdat.Rdata` contains data of the richness (number of species) of four groups: birds (`bird`, and its log transform `logbird`), plants (`plant`, `logpl`), mammals (`mammal`, `logmam`), herptiles (`herptile`, `logherp`), and the total species richness of all four groups combined (`totsp`, `logtot`) in 30 wetlands in the Ottawa-Cornwall-Kingston area. Also shown are the latitude and longitude of the wetland (`lat`, `long`), its area (`logarea`), the percentage of the wetland

covered by water at all times during the year (*swamp*), the percentage of forested land within 1 km of the wetland (*cpfor2*), and the density (in m/hectare) of hard-surface roads within 1 km of the wetland (*thtden*).

We will focus on herptiles for this exercise, so we better first have a look at how this variable is distributed and correlated to the potential independent variables:


```
require(car)
scatterplotMatrix(~logherp + logarea + cpfor2 + thtden +
  swamp,
  reg.line = lm, smooth = TRUE, span = 0.5, diagonal = "density",
  data = mydata)
```

Figure 1.



Multiple regression models from scratch

We begin the multiple regression exercise by considering a situation with one dependent variable and three (possibly) independent variables. First, we will start from scratch and build a multiple regression model based on what we know from building simple regression models. Next, we will look at automated methods of building multiple regressions models using simultaneous, forward, and backward stepwise procedures.

 Using the `mydata` subset of the `Mregdat.Rdata` data file, regress `logherp` on `logarea`.

On the basis of the regression, what do you conclude?

```
model.loga<-lm(logherp ~ logarea, data = mydata)
summary(model.loga)
```

```
Call:
lm(formula = logherp ~ logarea, data = mydata)

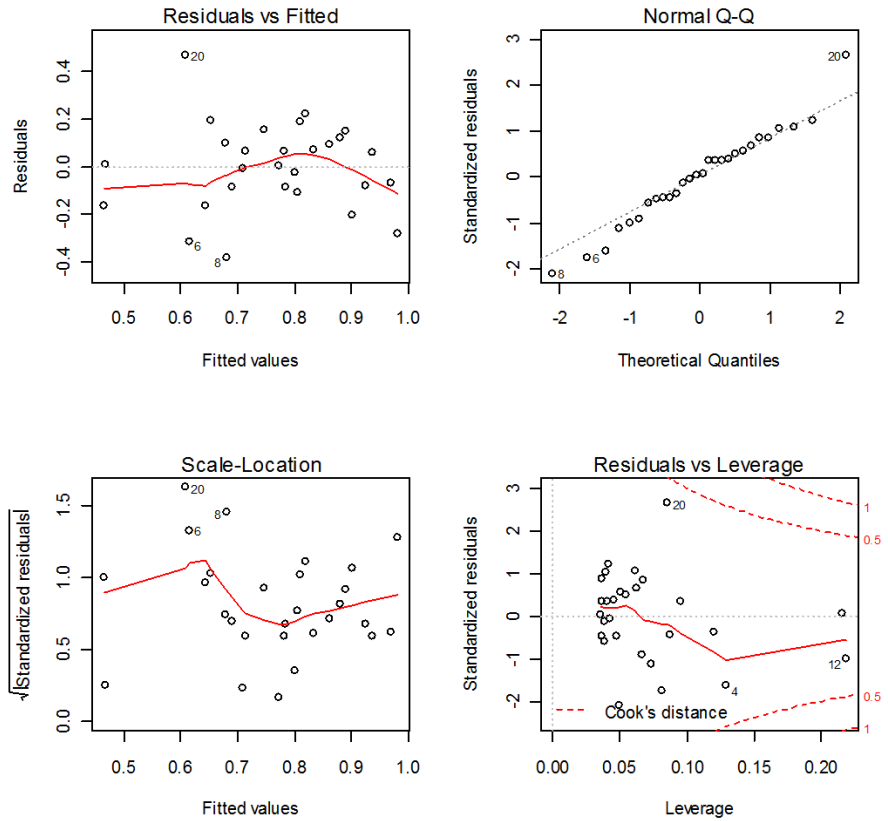
Residuals:
    Min       1Q   Median       3Q      Max
-0.38082 -0.09265  0.00763  0.10409  0.46977

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.18503    0.15725   1.177 0.249996
logarea      0.24736    0.06536   3.784 0.000818 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1856 on 26 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared:  0.3552,    Adjusted R-squared:  0.3304
F-statistic: 14.32 on 1 and 26 DF,  p-value: 0.0008185

opar <- par(mfrow(2, 2))
plot(model.loga)
par(opar)
```

Figure 2.



It looks like there is a positive relationship between herptile species richness and wetland area: the larger the wetland, the greater the number of species. Note, however, that about 2/3 of the observed variability in species richness among wetlands is not "explained" by wetland area ($R^2 = .355$). Residual analysis shows no major problems with normality, heteroscedasticity or independence of residuals.

☞ Rerun the above regression, this time replacing `logarea` with `cpfor2` as the independent variable, such that the expression in the formula field reads: `Logherp~cpfor2`. What do you conclude?


```
Call:
lm(formula = logherp ~ cpfor2, data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max
-0.4909 -0.1027  0.0588  0.1603  0.2516

Coefficients:
(Intercept)  0.609197  0.104233  5.845 3.68e-06 ***
cpfor2       0.002706  0.001658  1.632  0.115
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.2202 on 26 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared: 0.09289, Adjusted R-squared: 0.058
F-statistic: 2.662 on 1 and 26 DF, p-value: 0.1148
```

According to this result, we would accept the null hypothesis, and conclude that there is no relationship between herptile density and the proportion of forest on adjacent lands. But what happens when we enter both variables into the regression simultaneously?

 Rerun the above regression one more time, this time adding both independent variables into the model at once, such that `logherp~logarea+cpfor2`. What do you conclude?

```
Call:
lm(formula = logherp ~ logarea + cpfor2, data = mydata)


Residuals:
    Min       1Q   Median       3Q      Max
-0.40438 -0.11512  0.01774  0.08187  0.36179

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.027058   0.166749   0.162 0.872398
logarea      0.247789   0.061603   4.022 0.000468 ***
cpfor2       0.002724   0.001318   2.067 0.049232 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.175 on 25 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared: 0.4493, Adjusted R-squared: 0.4052
F-statistic: 10.2 on 2 and 25 DF, p-value: 0.0005774
```

Now we reject both null hypotheses that the slope of the regression of `logherp` on `logarea` is zero and that the slope of the regression of `logherp` on `cpfor2` is zero.

Why is `cpfor2` a significant predictor of `logherp` in the combined model when it was not significant in the simple linear model? The answer lies in the fact that it is sometimes necessary to control for one variable in order to detect the effect of another variable. In this case, there is a significant relationship between `logherp` and `logarea` that masks the relationship between `logherp` and `cpfor2`. When both variables are entered into the model at once, the effect of `logarea` is controlled for, making it possible to detect a `cpfor2` effect (and vice versa).

 Run another multiple regression, this time substituting `THTDEN` for `cpfor2` as an independent variable (`Logherp~Logarea+thtden`).

```
Call:
lm(formula = logherp ~ logarea + thtden, data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max

```

```
-0.31583 -0.12326 0.02095 0.13201 0.31674
```

```
Coefficient s:
```

```
Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.37634 0.14926 2.521 0.018437 *
logarea 0.22504 0.05701 3.947 0.000567 ***
thtden -0.04196 0.01345 -3.118 0.004535 **
```

```
---
```


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

```
Residual standard error: 0.1606 on 25 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared: 0.5358, Adjusted R-squared: 0.4986
F-statistic: 14.43 on 2 and 25 DF, p-value: 6.829e-05
```

In this case we reject the null hypotheses that there are no effects of wetland area (`logarea`) and road density (`thtden`) on herptile richness (`logherp`). Note here that road density has a negative effect on richness, whereas wetland area and forested area (`cpfor2`; results from previous regression) both have positive effects on herptile richness.

The R^2 of this model is even higher than the previous multiple regression model, reflecting a higher correlation between `logherp` and `thtden` than between `logherp` and `cpfor2` (if you run a simple regression between `logherp` and `thtden` and compare it to the `cpfor2` regression you should be able to detect this).

Thus far, it appears that herptile richness is related to wetland area (`logarea`), road density (`thtden`), and possibly forest cover on adjacent lands (`cpfor2`). But, does it necessarily follow that if we build a regression model with all three independent variables, that all three will show significant relationships? No, because we have not yet examined the relationship between `logarea`, `cpfor2` and `thtden`. Suppose, for example, two of the variables (say, `cpfor2` and `thtden`) are perfectly correlated. Then the `thtden` effect is nothing more than the `cpfor2` effect (and vice versa), so that once we include one or the other in the regression model, none of the remaining variability would be explained by the third variable.

 Fit a regression model with `logherp` as the dependent variable and `logarea`, `cpfor2` and `thtden` as the independent variables. What do you conclude?

```
Call:
```

```
lm(formula = logherp ~ logarea + cpfor2 + thtden, data = mydata)
```

```
Residuals:
```

```
Min      1Q  Median      3Q      Max
-0.30729 -0.13779  0.02627  0.11441  0.29582
```

```
Coefficient s:
```

```
Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.284765 0.191420 1.488 0.149867
logarea 0.228490 0.057647 3.964 0.000578 ***
cpfor2 0.001095 0.001414 0.774 0.446516
thtden -0.035794 0.015726 -2.276 0.032055 *
---
```

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

Residual standard error: 0.1619 on 24 degrees of freedom
(2 observations deleted due to missingness)

Multiple R-squared: 0.5471 ², Adjusted R-squared: 0.4904
F-statistic: 9.662 on 3 and 24 DF, p-value: 0.0002291

Several things to note here:

① The regression coefficient for `cpfor2` has become non-significant: once the variability explained by `logarea` and `thtden` is removed, a non-significant part of the remaining variability is explained by `cpfor2`.

② The R^2 for this model (.547) is only marginally larger than the R^2 for the model with only `logarea` and `thtden` (.536, which is again consistent with the non-significant coefficient for `cpfor2`.

Note also that although the regression coefficient for `thtden` has not changed much from that obtained when just `thtden` and `logarea` were included in the fitted model (-.036 vs -.042, the standard error for the regression coefficient for `thtden` has increased slightly, meaning the estimate is less precise. If the correlation between `thtden` and `cpfor2` was greater, the change in precision would also be greater.

We can compare the fit of the last two models (i.e., the model with all 3 variables and the model with only `logarea` and `thtden`) to decide which model is best to include.

```
anova(model.loga.cpfor2.thtden, model.loga.thtden)
```

Analysis of Variance Table

```
Model 1: logherp ~ logarea + cpfor2 + thtden
```

```
Model 2: logherp ~ logarea + thtden
```

	Res. Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	24	0.62937				
2	25	0.64508	-1	-0.01571	0.599	0.4465

Note that this is the identical result we obtained via the t-test of the effect of `cpfor2` in the model with all 3 variables above as they are testing the same thing (this should make sense to you). From this analysis, we would conclude that the full model with all three variables included does not offer a significant improvement in fit over the model with only `logarea` and `thtden`. This isn't surprising given that we already know that we cannot reject the null hypothesis of no effect of `cpfor2` in the full model.

Overall, we would conclude, on the basis of these analyses, that:

1. Given the three variables `thtden`, `logarea` and `cpfor2`, the best model is one that includes the first two variables.


2. There is evidence of a negative relationship between herptile richness and the density of roads on adjacent lands.
3. There is evidence that the larger the wetland area, the greater the herptile species richness.

Note that by "best", I don't mean the best possible model, I mean the best one given the three predictor variables we started with. It seems pretty clear that there are other factors controlling richness in wetlands, since even with the "best" model, almost half of the variability in richness is unexplained.

Stepwise multiple regression procedures

There are a number of techniques available for selecting the multiple regression model that best suits your data. When working with only three independent variables it is often sufficient to work through the different combinations of possible variables yourself, until you are satisfied you have fit the best model. This is, essentially, what we did in the first section of this lab. However, the process can become tedious when dealing with numerous independent variables, and you may find an automatic procedure for fitting models to be easier to work with.

Stepwise regression in R relies on the Akaike Information Criterion, as a measure of goodness of fit ($AIC=2k + 2\ln(L)$) where k is the number of regressors, and L is the maximized value of the likelihood function for the model). This is a statistic that rewards prediction precision while penalizing model complexity. If a new model has an AIC lower than that of the current model, the new model is a better fit to the data.

 Still working with the `Mregdat` data, run a stepwise multiple regression on the same set of variables:

```
model.loga.cpfor2.thtden<-lm(logherp ~ logarea + cpfor2 +
thtden, data = mydata)
# Stepwise Regression
library(MASS)
step <- stepAIC(model.loga.cpfor2.thtden, direction="both")
step$anova # display results
```

```
Start : AIC=-98.27 1
logherp ~ logarea + cpfor2 + thtden
```

	Df	Sum of Sq	RSS	AIC 2
- cpfor2	1	0.016	0.645	-99.576
<none>			0.629	-98.267
- thtden	1	0.136	0.765	-94.794
- logarea	1	0.412	1.041	-86.167

```
Step: AIC=-99.58
logherp ~ logarea + thtden
```

	Df	Sum of Sq	RSS	AIC
<none>			0.645	-99.576
+ cpfor2	1	0.016	0.629	-98.267
- thtden	1	0.251	0.896	-92.376
- logarea	1	0.402	1.047	-88.013

```
> step$anova # display results
Stepwise Model Path
Analysis of Deviance Table
```

```
Initial Model:
logherp ~ logarea + cpfor2 + thtden
```

```
Final Model:
logherp ~ logarea + thtden
```

Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
1			24	0.6293717	-98.26666
2 - cpfor2	1	0.01570813	25	0.6450798	-99.57640

Examining the output, we find:

- 1 R calculated the AIC for the starting model (here the full model with the 3 independent variables).
- 2 The AIC for models where terms are deleted.. Note here that the only way to reduce the AIC is to drop cpfor2.
- 3 The AIC for models where terms are added or deleted from the model selected in the first step (i.e. $\text{logherp} \sim \text{logarea} + \text{thtden}$). Note that none of these models are better.

Instead of starting from the full (saturated) model and removing and possibly re-adding terms (i.e. `direction = "both"`), one can start from the null model and only add terms:

```
# Forward selection approach
model.null<-lm(logherp ~ 1, data = mydata)
step <- stepAIC(model.null, scope=~. + logarea + cpfor2 + thtden, direction="forward")
step$anova # display results
```

How does it differ from the first procedure?

```
Start: AIC=-82.09
logherp ~ 1
```

	Df	Sum of Sq	RSS	AIC
+ logarea	1	0.494	0.896	-92.376
+ thtden	1	0.342	1.047	-88.013
+ cpfor2	1	0.129	1.260	-82.820
<none>			1.390	-82.091

```
Step: AIC=-92.38
logherp ~ logarea
```

```

      Df Sum of Sq    RSS    AIC
+ t h t den  1      0.251  0.645 -99.576
+ c p f o r 2  1      0.131  0.765 -94.794
<none>
      0.896 -92.376

```

```

Step:  AIC=-99.58
logherp ~ logarea + t h t den

```

```

      Df Sum of Sq    RSS    AIC
<none>
      0.645 -99.576
+ c p f o r 2  1      0.016  0.629 -98.267

```

```
> step$anova # display results
```

```

Stepwise Model Path
Analysis of Deviance Table

```

```

Initial Model:
logherp ~ 1

```

```

Final Model:
logherp ~ logarea + t h t den

```

```

      Step Df  Deviance Resid. Df Resid. Dev      AIC
1
2 + logarea  1  0.4935233      26  0.8960048 -92.37639
3 + t h t den  1  0.2509250      25  0.6450798 -99.57640
>

```

You should first notice that the final result is the same as the default stepwise regression and as what we got building the model from scratch. In forward selection, R first fits the least complex model (i.e. with only an intercept), and then adds variables, one by one, according to AIC statistics. Thus, in the above example, the model was first fit with only an intercept. Next, `logarea` was added, followed by `t h t den`. `cpfor2` was not added because it would make AIC increase to above that of the model fit with the first two variables.

Generally speaking, when doing multiple regressions, it is good practice to try several different methods (e.g. all regressions, stepwise, and backward elimination, etc.) and see whether you get the same results. If you don't, then the "best" model may not be so obvious, and you will have to think very carefully about the inferences you draw. In this case, regardless of whether we use automatic, or forward/backward stepwise regression, we arrive at the same model.

When doing multiple regression, always bear in mind the following:

1. Different procedures may produce different "best" models, i.e. the "best" model obtained using forward stepwise regression needn't necessarily be the same as that obtained using backward stepwise. It is good practice to try several different methods and see whether you end up with the same result. If you don't, it is almost invariably due to multicollinearity among the independent variables.
2. Be wary of stepwise regression. As the authors of SYSTAT, another commonly used statistical package, note, "Stepwise regression is probably the most abused computerized statistical

technique ever devised. If you think you need automated stepwise regression to solve a particular problem, you probably don't. Professional statisticians rarely use automated stepwise regression because it does not necessarily find the "best" fitting model, the "real" model, or alternative "plausible" models. Furthermore, the order in which variables enter or leave a stepwise program is usually of no theoretical significance. You are always better off thinking about why a model could generate your data and then testing that model."

3. Remember that just because there is a significant regression of Y on X doesn't mean that X causes Y: correlation does not imply causation!

Detecting multicollinearity

Multicollinearity is the presence of correlations among independent variables. In extreme cases (perfect collinearity) it will prevent you from fitting some models. When collinearity is not perfect, it reduces your ability to test for the effect of individual variables, **but does not affect the ability of the model to predict.**

The help file for the HH package contains this clear passage about one of the indices of multicollinearity, the variance inflation factors:

"A simple diagnostic of collinearity is the variance inflation factor, VIF, one for each regression coefficient (other than the intercept). Since the condition of collinearity involves the predictors but not the response, this measure is a function of the X's but not of Y. The VIF for predictor i is $1/(1-R_i^2)$, where R_i^2 is the R^2 from a regression of predictor i against the remaining predictors. If R_i^2 is close to 1, this means that predictor i is well explained by a linear function of the remaining predictors, and, therefore, the presence of predictor i in the model is redundant. Values of VIF exceeding 5 are considered evidence of collinearity: The information carried by a predictor having such a VIF is contained in a subset of the remaining predictors. If, however, all of a model's regression coefficients differ significantly from 0 (p -value $< .05$), a somewhat larger VIF may be tolerable."

VIFs indicate by how much the variance of each regression coefficient is increased by the presence of collinearity.

Trap. There are several `vif()` functions (I know of at least three in the packages `car`, `HH` and `DAAG`) and I do not know how they differ.

To quantify multicollinearity, one can simply call the `vif()` function from the package `car`:

```
require(car)
vif(model.loga.cpf2.thtden)
```

```
logarea  cpfor2  tht den
1.022127 1.344455 1.365970
```

Here there is no evidence that multicollinearity is a problem since all vif are close to 1.

Polynomial regression

In the regression models considered so far, we have assumed that the relationship between the dependent and independent variables is linear. If not, in some cases it can be made linear by transforming one or both variables. On the other hand, for many biological relationships no transformation in the world will help, and we are forced to go with some sort of non-linear regression method.

The simplest type of nonlinear regression method is polynomial regression, in which you fit regression models that include independent variables raised to some power greater than one, e.g. X^2 , X^3 , etc.


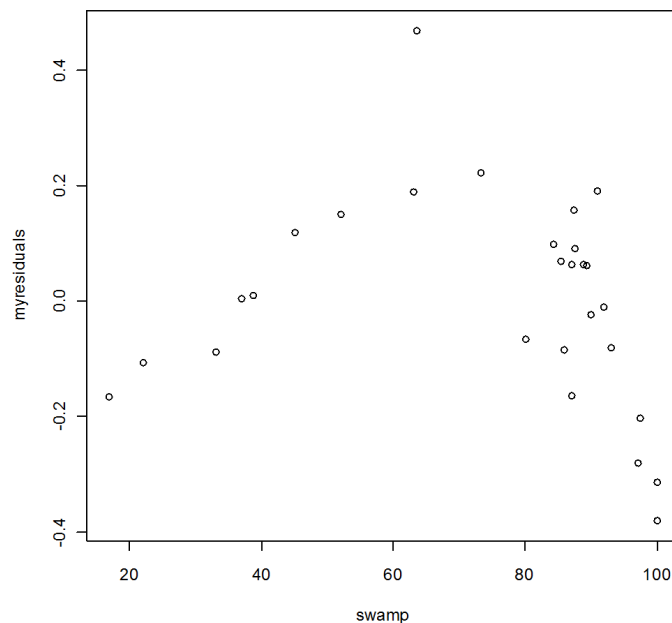
 Plot the relationship between the residuals of the logherp-logarea regression and swamp.

Figure 3.



Visual inspection of this graph suggests that there is a strong, but highly nonlinear, relationship between these two variables. Try regressing the residuals of the logherp-logarea regression on swamp. What do you conclude?

```
Call:
lm(formula = myresiduals ~ swamp, data = mydata.noNA)

Residuals:
    Min       1Q   Median       3Q      Max
-0.350880 -0.138189  0.003131  0.108485  0.458021

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.084571  0.109265   0.774   0.446
swamp       -0.001145  0.001403  -0.816   0.422

Residual standard error: 0.1833 on 26 degrees of freedom
Multiple R-squared:  0.02498,    Adjusted R-squared:  -0.01252
F-statistic: 0.666 on 1 and 26 DF,  p-value: 0.4219
```

In other words, the fit is terrible, even though you can see from the graph that there is in fact quite a strong relationship between the two - it's just that it is a non-linear relationship. (If you look at your residuals from this model, you will see strong evidence of nonlinearity, as expected.)

Rerun the above regression but add a second term in the **Formula** field to represent $(swamp)^2$. The expression should appear as: `residuals~SWAMP+SWAMP^2`. What do you conclude? What does examination of the residuals from this multiple regression tell you?

```
Call:
lm(formula = myresiduals ~ swamp + I(swamp^2), data = mydata.noNA)

Residuals:
    Min       1Q   Median       3Q      Max
-0.181185 -0.085350  0.007377  0.067327  0.242455

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.804e-01  1.569e-01  -4.975 3.97e-05 ***
swamp        3.398e-02  5.767e-03   5.892 3.79e-06 ***
I(swamp^2)  -2.852e-04  4.624e-05  -6.166 1.90e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1177 on 25 degrees of freedom
Multiple R-squared:  0.6132,    Adjusted R-squared:  0.5823
F-statistic: 19.82 on 2 and 25 DF,  p-value: 6.972e-06
```

It is clear that once the effects of area are controlled for, a considerable amount of the remaining variability in herptile richness is explained by swamp, in a nonlinear fashion. If you examine the residuals, you will see that compared to the linear model, the fit is much better.

Based on the results from the above analyses, how would you modify the regression model arrived at above? What, in your view, is the

“best” overall model? Why? How would you rank the various factors in terms of their effects on herptile species richness?

In light of these results, we might want to try and fit a model which includes `logarea`, `thtden`, `cpfor2`, `swamp` and `swamp^2`:

```
Call:
lm(formula = logherp ~ logarea + cpfor2 + thtden + swamp + I(swamp^2),
    data = mydata)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.201797 -0.056170 -0.002072  0.051814  0.205626
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.203e-01  1.813e-01  -1.766   0.0912 .
logarea      2.202e-01  3.893e-02   5.656  1.09e-05 ***
cpfor2      -7.864e-04  9.955e-04  -0.790   0.4380
thtden      -2.929e-02  1.048e-02  -2.795   0.0106 *
swamp        3.113e-02  5.898e-03   5.277  2.70e-05 ***
I(swamp^2)  -2.618e-04  4.727e-05  -5.538  1.45e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.1072 on 22 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared:  0.8181,    Adjusted R-squared:  0.7767
F-statistic: 19.78 on 5 and 22 DF,  p-value: 1.774e-07
```

Note that on the basis of this analysis, we should probably drop `cpfor2` and refit using the remaining variables:

```
Call:
lm(formula = logherp ~ logarea + thtden + swamp + I(swamp^2),
    data = mydata)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.19621 -0.05444 -0.01202  0.07116  0.21295
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.461e-01  1.769e-01  -1.957   0.0626 .
logarea      2.232e-01  3.842e-02   5.810  6.40e-06 ***
thtden      -2.570e-02  9.364e-03  -2.744   0.0116 *
swamp        2.956e-02  5.510e-03   5.365  1.89e-05 ***
I(swamp^2)  -2.491e-04  4.409e-05  -5.649  9.46e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.1063 on 23 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared:  0.8129,    Adjusted R-squared:  0.7804
F-statistic: 24.98 on 4 and 23 DF,  p-value: 4.405e-08
```

How about multicollinearity in this model?

```
> vif(model.polynomial.reduced)
logarea      thtden      swamp I(swamp^2)
 1.053193   1.123491  45.845845  45.656453
```

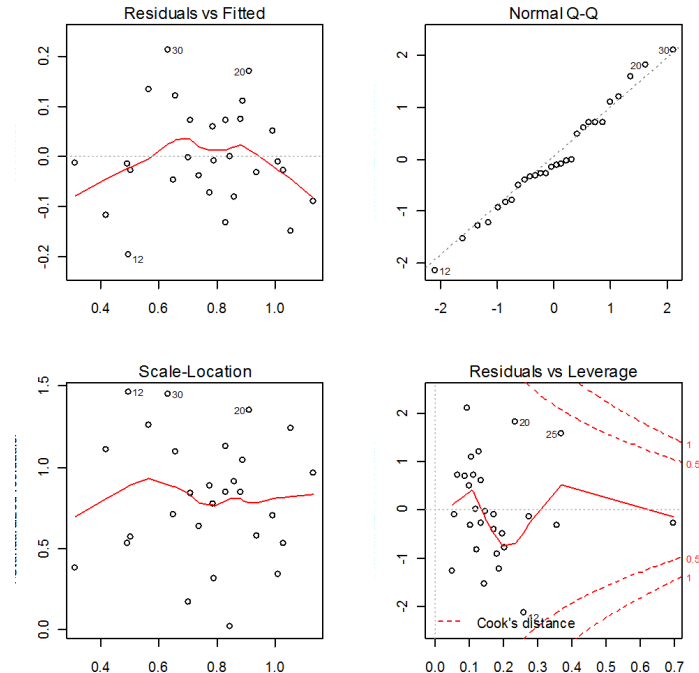
VIF for the two swamp terms are much higher than the standard threshold of 5. However, this is expected for polynomial terms, and not really a concern given that both terms are highly significant in the model. The high VIF means that these two coefficients are not estimated precisely, but using both in the model still allows to make a good prediction (i.e. account for the response to swamp).

Checking assumptions of a multiple regression model

All the model selection techniques or the manual model crafting assumes that the standard assumptions (independence, normality, homoscedasticity, linearity) are met. Given that a large number of models can be fitted, it may seem that testing the assumptions at each step would be an herculean task. However, it is generally sufficient to examine the residuals of the full (saturated) model and of the final model. Terms not contributing significantly to the fit do not affect residuals much, and therefore, the residuals to the full model, or the residuals to the final model, are generally sufficient.

Let's have a look at the diagnostic plots for the final model:

Figure 4.



Everything looks about right here. For the skeptic, let's run the formal tests:

```
shapiro.test(residuals(model.polynomial.reduced))
```

```

Shapiro-Wilk normality test
data: residuals(model.polynomial.reduced)
W = 0.9837, p-value = 0.9278

```

The residuals do not deviate from normality. Good.

```

>
> #Homoscedasticity
> bptest(model.polynomial.reduced)

```

```

studentized Breusch-Pagan test
data: model.polynomial.reduced
BP = 3.8415, df = 4, p-value = 0.4279

```

No deviation from homoscedasticity either. Good.

```

> #Serial autocorrelation
> dwtest(model.polynomial.reduced)

```

```

Durbin-Watson test
data: model.polynomial.reduced
DW = 1.725, p-value = 0.2095
alternative hypothesis: true autocorrelation is greater than 0

```

No serial correlation in the residuals, so no evidence of non-independence.

```

>
#Linearity
resettest(model.polynomial.reduced, type = "regressor", data = mydata)

```

```

RESET test
data: model.polynomial.reduced
RESET = 0.9823, df1 = 8, df2 = 15, p-value = 0.4859

```

And no significant deviation from linearity. So it seems that all is fine.

Visualizing effect size

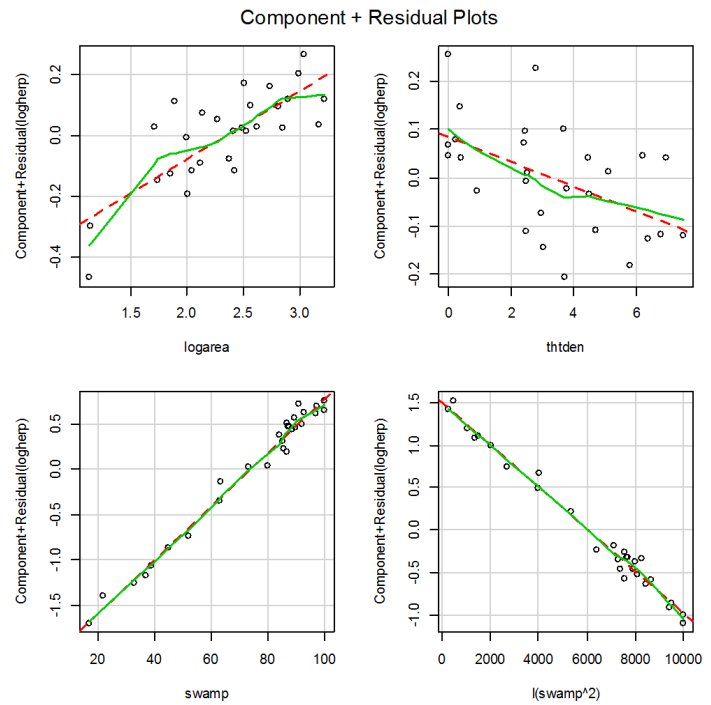
How about effect size? How is that measured or viewed? The regression coefficients can be used to measure effect size, although it may be better to standardize them so that they become independent of measurement units. But a graph is often useful as well. In this context, some of the most useful graphs are called partial residual plots (or component + residual plots). These plots show how the dependent variable, corrected for other variables in the model, varies with each individual variable. Let's have a look:

```

# Evaluate visually linearity and effect size
# component + residual plot
cr.plots(model.polynomial.reduced, one.page=TRUE, ask=FALSE)

```

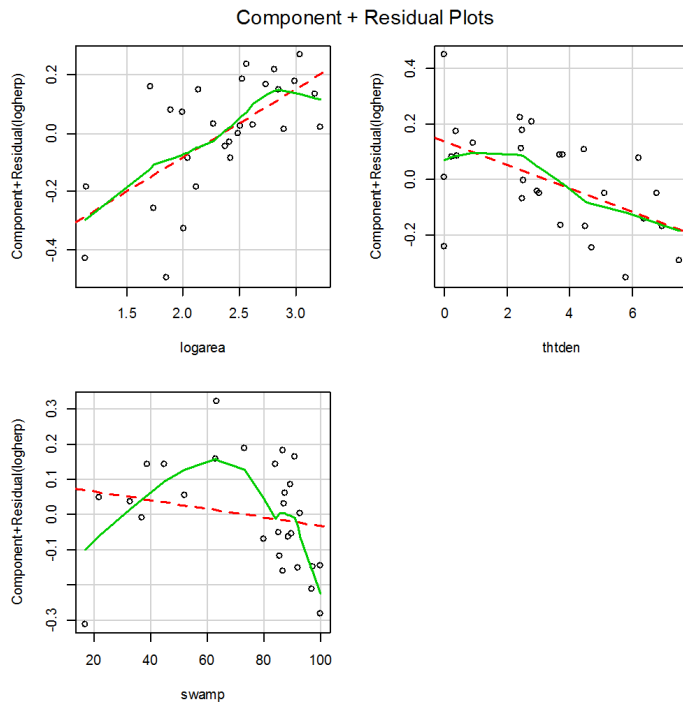
Figure 5.



Note that the vertical scale varies among plots. For thtden, the dependent variable (\log_{10} (herptile richness)) varies by about 0.4 units over the range of thtden in the sample. For logarea, the variation is about 0.6 log units. For swamp, it is a bit tricky since there are two terms and they have opposite effect (leading to a peaked relationship), so the plots are less informative. However, there is no deviation from linearity to be seen.

To illustrate what these graphs would look like if there was deviation from linearity, let's drop the squared swamp term and produce the graphs and run the reset test:

Figure 6.



The lack of linearity along the gradient of swamp becomes obvious. And the RESET test also detects a violation from linearity:

RESET t est

```
data: g
RESET = 6.7588, df 1 = 6, df 2 = 18, p-value = 0.0007066
```

Testing for interactions

When there are multiple independent variables one should always be ready to assess interactions. In most multiple regression contexts this is somewhat difficult because adding interaction terms increases overall multicollinearity and because in many cases there are not enough observations to test all interactions, or the observations are not well balanced to make powerful tests for interactions.

Going back to our final model, see what happens if one tries to fit the fully saturated model with all interactions:

```
fullmodel.withinteractions<-lm(logherp ~ logarea * cpfor2 *
thtden * swamp * I(swamp^2), data= Mregdat)
```

```
summary(fullmodel.withinteractions)
```

```
Call:
lm(formula = logherp ~ logarea * cpfor2 * thtden * swamp * I(swamp^2),
    data = Mregdat)
```

```
Residuals:
```

ALL 28 residuals are 0: no residual degrees of freedom

Coefficients: (4 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-5.948e+03	NA	NA	NA
logarea	3.293e+03	NA	NA	NA
cpfor2	7.080e+01	NA	NA	NA
thtden	9.223e+02	NA	NA	NA
swamp	1.176e+02	NA	NA	NA
I (swamp^2)	-3.517e-01	NA	NA	NA
logarea:cpfor2	-3.771e+01	NA	NA	NA
logarea:thtden	-4.781e+02	NA	NA	NA
cpfor2:thtden	-1.115e+01	NA	NA	NA
logarea:swamp	-7.876e+01	NA	NA	NA
cpfor2:swamp	-1.401e+00	NA	NA	NA
thtden:swamp	-1.920e+01	NA	NA	NA
logarea:I (swamp^2)	5.105e-01	NA	NA	NA
cpfor2:I (swamp^2)	3.825e-03	NA	NA	NA
thtden:I (swamp^2)	7.826e-02	NA	NA	NA
swamp:I (swamp^2)	-2.455e-03	NA	NA	NA
logarea:cpfor2:thtden	5.359e+00	NA	NA	NA
logarea:cpfor2:swamp	8.743e-01	NA	NA	NA
logarea:thtden:swamp	1.080e+01	NA	NA	NA
cpfor2:thtden:swamp	2.620e-01	NA	NA	NA
logarea:cpfor2:I (swamp^2)	-5.065e-03	NA	NA	NA
logarea:thtden:I (swamp^2)	-6.125e-02	NA	NA	NA
cpfor2:thtden:I (swamp^2)	-1.551e-03	NA	NA	NA
logarea:swamp:I (swamp^2)	-4.640e-04	NA	NA	NA
cpfor2:swamp:I (swamp^2)	3.352e-05	NA	NA	NA
thtden:swamp:I (swamp^2)	2.439e-04	NA	NA	NA
logarea:cpfor2:thtden:swamp	-1.235e-01	NA	NA	NA
logarea:cpfor2:thtden:I (swamp^2)	7.166e-04	NA	NA	NA
logarea:cpfor2:swamp:I (swamp^2)	NA	NA	NA	NA
logarea:thtden:swamp:I (swamp^2)	NA	NA	NA	NA
cpfor2:thtden:swamp:I (swamp^2)	NA	NA	NA	NA
logarea:cpfor2:thtden:swamp:I (swamp^2)	NA	NA	NA	NA

Residual standard error: NaN on 0 degrees of freedom
(2 observations deleted due to missingness)

Multiple R-squared: 1, Adjusted R-squared: NaN
F-statistic: NaN on 27 and 0 DF, p-value: NA

Indeed, it is not possible to include all 32 terms with only 28 observations. There are not enough data points, R square is one, and the model perfectly overfits the data.

If you try to use an automated routine to "pick" the best model out of this soup, R complains:

```
step(fullmodel.withinteractions)
```

```
Error in step(fullmodel.withinteractions) :  
  AIC is -infinity for this model, so 'step' cannot proceed
```

Does this mean you can forget about potential interactions and simply accept the final model without a thought? No. You simply do not have enough data to test for all interactions. But there is a compromise worth attempting, comparing the final model to a model with a subset of the interactions, say all second order interactions, to check whether the inclusion of these interactions improves substantially the fit:

```
full.model.2ndinteractions<- lm(logherp ~ logarea + cpfor2 +  
thtden + swamp + I (swamp^2)  
+ logarea:cpfor2 + logarea:thtden + logarea:swamp  
+ cpfor2:thtden + cpfor2:swamp
```

```

+ thtden:swamp
, data= mydata)

summary(full.model.2ndinteractions)

Call:
lm(formula = logherp ~ logarea + cpf or 2 + tht den + swamp + I(swamp^2) +
    logarea:cpf or 2 + logarea:tht den + logarea:swamp + cpf or 2:tht den +
    cpf or 2:swamp + tht den:swamp, data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max
-0.216880 -0.036534  0.003506  0.042990  0.175490

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.339e-01  6.325e-01   0.686 0.502581
logarea      -1.254e-01  2.684e-01  -0.467 0.646654
cpf or 2     -9.344e-03  7.205e-03  -1.297 0.213032
tht den      -1.833e-01  9.035e-02  -2.028 0.059504 .
swamp        3.569e-02  7.861e-03  4.540 0.000334 ***
I(swamp^2)   -3.090e-04  7.109e-05  -4.347 0.000500 ***
logarea:cpf or 2  2.582e-03  2.577e-03  1.002 0.331132
logarea:tht den  7.017e-02  3.359e-02  2.089 0.053036 .
logarea:swamp  -5.290e-04  2.249e-03  -0.235 0.816981
cpf or 2:tht den -2.095e-04  6.120e-04  -0.342 0.736544
cpf or 2:swamp   4.651e-05  5.431e-05  0.856 0.404390
tht den:swamp   2.248e-04  4.764e-04  0.472 0.643336
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.108 on 16 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared:  0.8658,    Adjusted R-squared:  0.7735
F-statistic: 9.382 on 11 and 16 DF,  p-value: 4.829e-05

```

This model fits the data slightly better than the "final" model (it explains 86.6% of the variance in logherp, compared to 81.2% for the "final" model without interactions), but has twice as many parameters. If you look at the individual coefficients, some weird things happen: for example, the sign for logarea has changed. This is one of the symptoms of multicollinearity. Let's look at the variance inflation factors:

```

vif(full.model.2ndinteractions)

```

	logarea	cpf or 2	tht den	swamp	I (swamp^2)
logarea	49.86060	78.49622	101.42437	90.47389	115.08457
logarea:cpf or 2	66.97792	71.69894	67.27034	14.66814	29.41422
tht den:swamp	20.04410				

Ouch. All VIF are above 5, not only the ones involving the swamp terms. This model is not very satisfying it seems. Indeed the AIC for the two models indicate that the model with interactions has less information than the full model (remember, models with the lowest AIC value are to be preferred):

```

> AIC(full.model)
[1] -38.3433
> AIC(full.model.2ndinteractions)
[1] -34.86123

```

The `anova()` command can be used to test whether the addition of all interaction terms improves the fit significantly:

```
anova(full.model, full.model.2ndinteractions)
```

```
Analysis of Variance Table
```

```
Model 1: logherp ~ logarea + cpfor2 + tht den + swamp + I(swamp^2)
Model 2: logherp ~ logarea + cpfor2 + tht den + swamp + I(swamp^2) + loga-
rea:cpfor2 +
  logarea:tht den + logarea:swamp + cpfor2:tht den + cpfor2:swamp +
  tht den:swamp
  Res. Df    RSS Df Sum of Sq    F Pr(>F)
1      22 0.252820
2      16 0.186507  6  0.066314 0.9481  0.489
```

This test indicates that the addition of interaction terms did not reduce significantly the residual variance around the full model. How about a comparison with the final model without cpfor2?

```
anova(model.polynomial.reduced, full.model.2ndinteractions)
```

```
Analysis of Variance Table
```

```
Model 1: logherp ~ logarea + tht den + swamp + I(swamp^2)
Model 2: logherp ~ logarea + cpfor2 + tht den + swamp + I(swamp^2) + loga-
rea:cpfor2 +
  logarea:tht den + logarea:swamp + cpfor2:tht den + cpfor2:swamp +
  tht den:swamp
  Res. Df    RSS Df Sum of Sq    F Pr(>F)
1      23 0.259992
2      16 0.186507  7  0.073486 0.9006  0.5294
```

And this comparison suggests that our final model does not make worse predictions than the full model with interactions.

Dredging and the information theoretical approach

One of the main critiques of stepwise methods is that the p-values are not strictly correct because of the large number of tests that are actually done. This is the multiple testing problem. In building linear models (multiple regression for example) from a large number of independent variables, and possibly their interactions, there are so many possible combinations that if one were to use Bonferroni type corrections, it would make tests very conservative.

An alternative, very elegantly advocated by Burnham and Anderson (2002, Model selection and multimodel inference: a practical information-theoretic approach. 2nd ed), is to use AIC (or better the AICc that is more appropriate for samples where the number of observations is less than about 40 times the number of variables) to rank potential models, and identify the set of models that are the best ones. One can then average the parameters across models, weighting using the probability that it is the best model to obtain coefficients that are more robust and less likely to be unduly affected by multicollinearity.

This approach has been implemented in the package MuMIn

```
library(MuMIn)
```

```
dd <- dredge(full.model.2ndinteractions)
```

Object dd will contain all possible models using the terms of our full model with 2nd order interactions.

Then, we can have a look at the subset of models that have an AICc within 4 units from the lowest AICc model. (Burnham and Anderson suggest that models that deviate by more than 4 AICc units have very little empirical support):

```
> # get models within 4 units of AICc from the best model
top.models.1 <- get.models(dd, subset = delta < 4)
avgmodel1 <- model.avg(top.models.1) # compute average parameters
summary(avgmodel1) #display averaged model
confint(avgmodel1) #display CI for averaged coefficients
```

```
Call:
model.avg.default(object = top.models.1)
```

Component model s: **1**

	df	logLik	AICc	Delta	Weight 2
23457	7	27.78	-35.95	0.00	0.34
2345	6	25.78	-35.56	0.39	0.28
123457	8	28.30	-33.02	2.93	0.08
234578	8	28.26	-32.95	3.00	0.08
12345	7	26.17	-32.74	3.21	0.07
23458	7	26.06	-32.51	3.44	0.06
234567	8	27.88	-32.17	3.78	0.05
23456	7	25.79	-31.99	3.97	0.05

Term codes:

cpfor2	I (swamp^2)	logarea	swamp	thtden
1	2	3	4	5
logarea:swamp	logarea:thtden	swamp:thtden		
6	7	8		

Model-averaged coefficients:
(full average) **3**

	Estimate	Std. Error	Adjusted SE	z value	Pr(> z)
(Intercept)	-2.075e-01	2.484e-01	2.593e-01	0.800	0.424
logarea	1.314e-01	1.185e-01	1.222e-01	1.076	0.282
swamp	3.193e-02	6.125e-03	6.438e-03	4.960	7e-07 ***
I(swamp^2)	-2.676e-04	4.904e-05	5.154e-05	5.193	2e-07 ***
thtden	-6.843e-02	5.324e-02	5.459e-02	1.254	0.210
logarea:thtden	2.139e-02	2.506e-02	2.565e-02	0.834	0.404
cpfor2	-1.202e-04	4.710e-04	4.886e-04	0.246	0.806
swamp:thtden	-3.277e-05	1.419e-04	1.475e-04	0.222	0.824
logarea:swamp	4.378e-05	5.378e-04	5.676e-04	0.077	0.939

(conditional average)

	Estimate	Std. Error	Adjusted SE	z value	Pr(> z)
(Intercept)	-2.075e-01	2.484e-01	2.593e-01	0.800	0.4236
logarea	1.314e-01	1.185e-01	1.222e-01	1.076	0.2820
swamp	3.193e-02	6.125e-03	6.438e-03	4.960	7e-07 ***
I(swamp^2)	-2.676e-04	4.904e-05	5.154e-05	5.193	2e-07 ***
thtden	-6.843e-02	5.324e-02	5.459e-02	1.254	0.2100
logarea:thtden	3.924e-02	2.125e-02	2.251e-02	1.743	0.0813 .
cpfor2	-8.187e-04	9.692e-04	1.027e-03	0.797	0.4253
swamp:thtden	-2.402e-04	3.127e-04	3.313e-04	0.725	0.4684
logarea:swamp	4.462e-04	1.664e-03	1.762e-03	0.253	0.8001

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

Relative variable importance:

	I(swamp^2)	logarea	swamp	thtden	logarea:thtden	cpfor2
Importance:	1.00	1.00	1.00	1.00	0.55	0.15
N containing models:	8	8	8	8	4	2
	swamp:thtden	logarea:swamp				
Importance:	0.14	0.10				
N containing models:	2	2				

	2.5 %	97.5 %
(I n t e r c e p t)	-0.7157333646	0.3007147516
l o g a r e a	-0.1080048582	0.3708612563
s w a m p	0.0193158426	0.0445532538
l (s w a m p ^ 2)	-0.0003686653	-0.0001666418
t h t d e n	-0.1754184849	0.0385545120
l o g a r e a : t h t d e n	-0.0048800385	0.0833595106
c p f o r 2	-0.0028313465	0.0011940283
s w a m p : t h t d e n	-0.0008894138	0.0004090457
l o g a r e a : s w a m p	-0.0030067733	0.0038991294

- 1 You first get the list of the models with an AICc within the desired 4 units of the best model. The variables that are included in the model are coded with the key just below.
- 2 For each model, in addition to the AICc, the Akaike weights are calculated. These estimate the probability that this model is the best one. Here we can see that the first (best) model has only 34% chances of being really the best one.
- 3 For the subset of models, weighted averages (using Akaike weights) for model parameters are calculated, with 95% CI. Note that, by default, terms missing from a model are assumed to have a coefficient of 0.

Bootstrapping multiple regression

When data do not meet the assumptions of normality and homoscedasticity and it is not possible to transform the data to meet the assumptions, bootstrapping can be used to compute confidence intervals for coefficients. If the distribution of the bootstrapped coefficients is symmetrical and approximately Gaussian, then empirical percentiles can be used to estimate the confidence limits.

The following code, using library simpleboot has been designed to be easily modifiable and will compute CI using empirical percentiles. Following this is an easier approach using the library boot that will calculate several different bootstrap confidence limits.

```
#####
Bootstrap analysis the simple way with library simpleboot

# Define model to be bootstrapped and the data source used
mymodel<-lm(logherp ~ logarea + thtden + swamp + I(swamp^2),
data = mydata)

# Set the number of bootstrap iterations
nboot<-1000

require(simpleboot)

# R is the number of bootstrap iterations
# Setting rows to FALSE indicates resampling of residuals
mysimpleboot<-lm.boot(mymodel, R = nboot, rows = FALSE)
```

```

# Extract bootstrap coefficients
myresults<-sapply(mysimpleboot$boot.list, function (x)
x$coef)

# Transpose matrix so that lines are bootstrap iterations
and columns are coefficients
tmyresults<-t(myresults)

# Plot histograms of bootstrapped coefficients

ncoefs<-length(data.frame(tmyresults))

par(mfrow=c(2,1), mai=c(0.5, 0.5, 0.5, 0.5), ask = TRUE)

for (i in 1 : ncoefs) {
  lab<-colnames(tmyresults)[i]
  x<-tmyresults[,i]

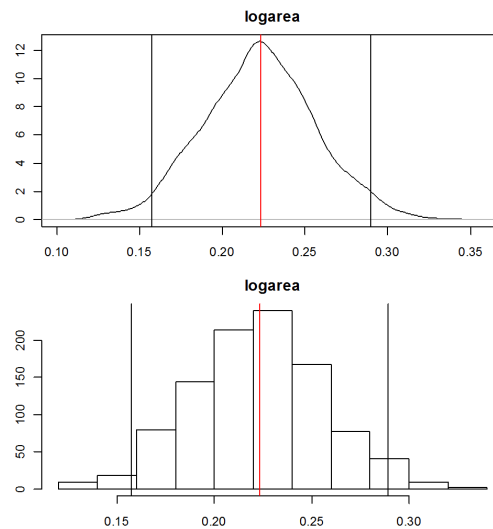
  plot(density(x),main=lab, xlab="")
  abline(v=mymodel$coef[i], col = "red")
  abline(v=quantile(x, c(0.025, 0.975)))
  hist(x, main=lab, xlab="")
  abline(v=quantile(x, c(0.025, 0.975)))
  abline(v=mymodel$coef[i], col = "red")
}

par(mfrow=c(1,1), ask = FALSE)

```

When run, it will pause to let you have a look at the distribution for each coefficient in the model by producing plots like:

Figure 7.



The top plot is the probability density function and the bottom one is the histogram of the bootstrap estimates for the coefficient. On these plots, the red line indicate the value of the parameter in the ordinary analysis, and

the two vertical black lines mark the limits of the 95% confidence interval. Here the CI does not include 0 and one can conclude that the effect of logarea on logherp is significantly positive.

Precise values for the limits can be obtained by:

```
> # Display empirical bootstrap quantiles (not corrected
for bias)
> p <- c(0.005,0.01, 0.025, 0.05, 0.95, 0.975, 0.99, 0.995)
> apply(tmyresults, 2, quantile, p)
      (Intercept) logarea   tht den   swamp   I(swamp^2)
0.5% -0.735605927 0.1321902 -0.048153235 0.01794482 -0.0003529401
1%   -0.706786582 0.1444830 -0.045895772 0.01886999 -0.0003384226
2.5% -0.644649399 0.1575567 -0.043016668 0.02015046 -0.0003263934
5%   -0.600986378 0.1679613 -0.039140810 0.02166009 -0.0003152855
95%  -0.075274798 0.2801058 -0.012076119 0.03777323 -0.0001859188
97.5% -0.035536091 0.2894439 -0.009660800 0.03929003 -0.0001715579
99%  -0.004244384 0.3014986 -0.007446500 0.04091320 -0.0001612558
99.5% 0.031621494 0.3095612 -0.006314632 0.04301388 -0.0001566147
```

These confidence limits are not reliable when the distribution of the bootstrap estimates deviate from Gaussian. If they do,, then it is preferable to compute so-called bias-corrected accelerated (BCa) confidence limits. The following code does just that:

```
#####
# Bootstrap analysis in multiple regression with BCa confi-
dence intervals
# Preferable when parameter distribution is far from normal

# Bootstrap 95% BCa CI for regression coefficients
library(boot)

# function to obtain regression coefficients for each itera-
tion
bs <- function(formula, data, indices) {
  d <- data[indices,] # allows boot to select sample
  fit <- lm(formula, data=d)
  return(coef(fit))
}

# bootstrapping with 1000 replications
results <- boot(data=mydata, statistic=bs,
  R=1000, formula=logherp ~ logarea + tht den + swamp +
I(swamp^2))

# view results
results
plot(results, index=1) # intercept
plot(results, index=2) # logarea
plot(results, index=3) # tht den
plot(results, index=4) # swamp
plot(results, index=5) # swamp^2

# get 95% confidence intervals
```

```

boot.ci(results, type="bca", index=1) # intercept
boot.ci(results, type="bca", index=2) # logarea
boot.ci(results, type="bca", index=3) # thtden
boot.ci(results, type="bca", index=4) # swamp
boot.ci(results, type="bca", index=5) # swamp^2

results

ORDINARY NONPARAMETRIC BOOTSTRAP

Call :
boot(data = mydata, statistic = bs, R = 1000, formula = logherp ~
      logarea + thtden + swamp + I(swamp^2))

Bootstrap Statistics :
      original      bias      std. error
t1*  -0.3460988043  2.613804e-04  2.346177e-01
t2*   0.2232323394 -5.971629e-03  5.222778e-02
t3*  -0.0256957141 -3.946859e-04  9.411536e-03
t4*   0.0295628938  4.057821e-04  7.148861e-03
t5*  -0.0002490924 -2.438465e-06  5.341945e-05

```

It will also display the standard graphs for each coefficient, and the resulting BCa CI. For example, for logarea:

```

> boot.ci(results, type="bca", index=2) # logarea
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 2)

Intervals :
Level      BCa
95% ( 0.1212, 0.3225 )
Calculations and Intervals on Original Scale

```

Note that the BCa interval is from 0.12 to 0.32, whereas the simpler percentile interval is 0.16 to 0.29. BCa interval here is longer on the low side, and shorter on the high side, which it should be given the distribution of bootstrap estimates.

Permutation test

Permutation tests are more rarely performed in multiple regression contexts than bootstrap. But here is code to do it

```

#####
#####
# Permutation in multiple regression
#
# using lperm library

require(lmPerm)

# Fit desired model on the desired dataframe
mymodel<-lm(logherp ~ logarea + thtden + swamp + I(swamp^2),
data = mydata)

```

```
mymodelProb<-lmp(logherp ~ logarea + thtden + swamp +  
I(swamp^2), data = mydata, perm="Prob")
```

```
summary(mymodel)  
summary(mymodelProb)
```


Lab- ANCOVA and GLM

After completing this laboratory exercise, you should be able to:

- Use R to do an analysis of covariance (ANCOVA) and interpret statistical models that have both continuous and categorical independent variables (GLM)
- Use R to test the assumptions underlying GLMs
- Use R to compare model fits
- Use R to do bootstrap and permutation tests on GLM type models including both continuous and categorical independent variables.

GLM

GLM stands for General Linear Model. GLMs are statistical models that can be written as $Y=XB+E$, where Y is a vector (or matrix) containing the dependent variable, X is a matrix of independent variables, B is a matrix of estimated parameters, and E is the vector (or matrix) of independent, normally distributed and homoscedastic residuals. All tests we have seen to date (t-test, simple linear regression, One-Way ANOVA, Multiway ANOVA, and multiple regression) are GLMs. Note that all models we have encountered until now contain only one type of variable (either continuous or categorical) as independent variables. In this laboratory exercise, you will fit models that have both type of independent variables. These models are also GLMs.

ANCOVA

ANCOVA stands for Analysis of Covariance. It is a type of GLM where there is one (or more) continuous independent variable (sometimes called a covariate) and one (or more) categorical independent variable. In the traditional treatment of ANCOVA in biostatistical textbooks, the ANCOVA model does not contain interaction terms between the continuous and categorical independent variables. Hence, the traditional ANCOVA analysis assumes that there is no interaction, and is preceded by a test of significance of interactions, equivalent to testing that the slopes (coefficients for the continuous independent variables) do not differ among level of the categorical independent variables (a test for homogeneity of slopes).

Some people, me included, use the term ANCOVA a bit more loosely for any GLM that involves both continuous and categorical variables. Be aware that, depending on the author, ANCOVA may refer to a model with or without interaction terms.

Homogeneity of slopes

In many biological problems, a question arises as to whether the slopes of two or more regression lines are significantly different; for example, whether two different insecticides differ in their efficacy, whether males and females differ in their growth curves, etc. These problems call for direct comparisons of slopes. GLMs (ANCOVAs) can test for equality of slopes (homogeneity of slopes).

Remember that there are two parameters that describe a regression line, the intercept and the slope. The ANCOVA model (sensus stricto) tests for homogeneity of intercepts, but the starting point for the analysis is a test for homogeneity of slopes. This test can be performed by fitting a model with main effects for both the categorical and continuous independent variables, plus the interaction term(s), and testing for significance of the addition of the interaction terms.

Case 1 - Size as a function of age (equal slopes example)

Using the file `anc1dat.Rdata`, test the hypothesis that female and male sturgeon at The Pas over the years 1978-1980 have the same observed growth rate, defined as the slope of the regression of \log_{10} of fork length, `fklngh`, on the \log_{10} age, `lage`.

First, let's have a look at the data. It would help to draw the regression line and a lowess trace to better assess linearity. Being fancy, one could also use more of R magic to spruce up the axis legends (note the use of `expression()` to get subscripts):

```
mydata<-anc1dat
require(ggplot2)
myplot<-ggplot(data=mydata, aes(x=lage,
y=log10(fklngh)))+facet_grid(.~sex)+geom_point()
myplot
myplot<-myplot+
  stat_smooth(method = lm, se=FALSE)+
  stat_smooth(se=FALSE, color="red")

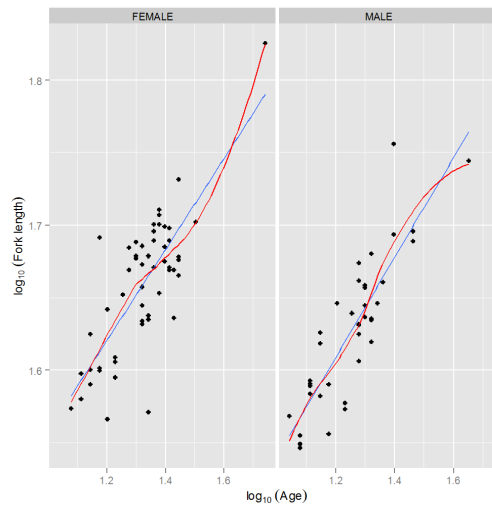
update_labels(myplot, list(
```

```

y = expression(log[10]~(Fork~length)),
x = expression(log[10]~(Age))
))
)

```

Figure 1.



The log-log transformation makes the relationship linear and, at first glance, there is no issues with assumptions of GLMs (although this should be confirmed by appropriate examination of the residuals). Let's fit the full model with both main effects and the interaction:

```

require(car)
model.full<-lm(lfkl ~ sex + l age + sex:l age, data = mydata)
Anova(model.full, type = 3)

```

Anova Table (Type III tests)

Response: lfkl

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	0.64444	1	794.8182	< 2.2e-16 ***
sex	0.00041	1	0.5043	0.4795
l age	0.07259	1	89.5312	4.588e-15 ***

sex:l age	0.00027	1	0.3367	0.5632 1
-----------	---------	---	--------	-----------------

Residuals 0.07135 88

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

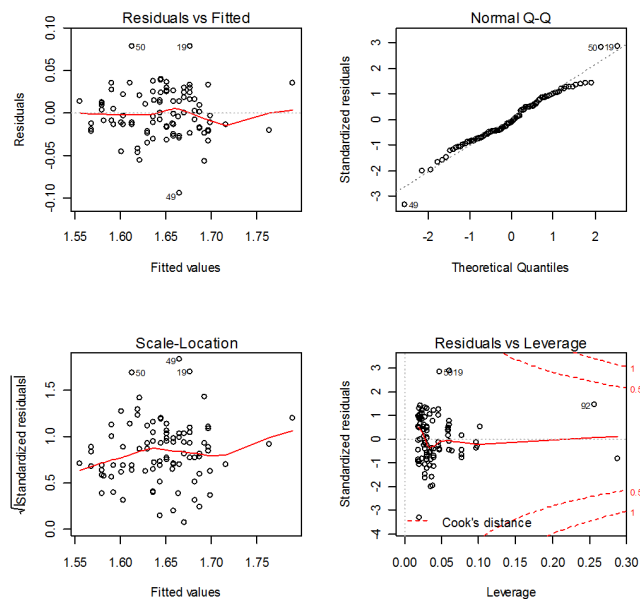
Residuals 88 0.0713501 0.0008108

1

Probability of observing an `l age:sex` interaction this strong or stronger under the null hypothesis that slope of the relationship between fork length and age does not vary between the sexes, or equivalently that the difference in fork length between males and females (if it exists) does not vary with age (and providing the assumptions of the analysis have been met).

Trap. Note that I used the `Anova()` function with an uppercase "a" (from the `car` library) instead of the "built in" `anova()` (with a lowercase "a") command to get the results using Type III sums of squares. The type III (partial) sums of squares are calculated as if each variable was the last entered in the model and correspond to the difference in explained SS between the full model and a reduced model where only that variable is excluded. The standard `anova()` function returns Type I (sequential) SS, calculated as each variable is added sequentially to a null model with only an intercept. In rare cases, the type I and type III SS are equal (when the design is perfectly balanced and there is no multicollinearity). In the vast majority of cases, they will differ, and I recommend that you always use the Type III SS in your analyses.

On the basis of this analysis, we would accept the null hypotheses that: (1) the slope of the regression of $\log(\text{fork length})$ on $\log(\text{age})$ is the same for males and females (the interaction term is not significant); (2) that the intercepts are also the same for the two sexes (the sex term is also not significant). But before accepting these conclusions, we should test the assumptions in the usual way.



With respect to normality, things look O.K., although there are several points in the top right corner that appear to lie off the line. We could also run a Wilk-Shapiro normality test and find $W = 0.9764$, $p = 0.09329$, also suggesting this assumption is valid. Homoscedasticity seems fine too, but if you want further evidence of

this, you can run one of the tests. Here I use the Breusch-Pagan test, which is appropriate when some of the independent variables are continuous (Levene's test is for categorical independent variables only):

```
bptest(model.full)
```

which yields:

```
studentized Breusch-Pagan test

data:  model.full
BP = 0.99979, df = 3, p-value = 0.8013
```


Since the null is that the residuals are homoscedastic, and p is rather large, the test confirm the visual assessment.

Further, there is no obvious pattern in the residuals, which implies there is no problem with the assumption of linearity. This too can be formally tested:

```
resettest(model.full, power = 2:3, type = "regressor",
           data = mydata)
RESET test
```

```
data:  model.full
RESET = 0.5986, df1 = 2, df2 = 86, p-value = 0.5519
```

The last assumption in this sort of analysis is that the covariate (in this case, *lage*) has no measurement error. We really have no way of knowing whether this assumption is justified, although multiple aging of fish by several different investigators usually gives ages that are within 1-2 years of each other, which is within the 10% considered by most to be the maximum for Type I modelling. Note that there is no "test" that you can do with the data to determine what the error is, at least in this case. If we had replicate ages for individual fish, it could be estimated quantitatively

 You will notice that there is one datum with a large studentized residual, i.e. an outlier (case 49). Eliminate this datum from your data file and rerun the analysis. Do your conclusions change?

```
model.full.no49<-lm(lfkl ~ sex + lage + sex:lage, data =
mydata[c(-49),])
Anova(model.full.no49, type=3)
```


Anova Table (Type III tests)

```
Response: lfkl
          Sum Sq Df F value Pr(>F)
(Intercept) 0.64255  1 895.9394 <2e-16 ***
sex          0.00038  1  0.5273 0.4697
lage        0.07378  1 102.8746 <2e-16 ***
sex:lage    0.00022  1  0.3135 0.5770
Residuals   0.06239 87
---
```

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

So the conclusion does not change if the outlier is deleted (not surprising as Cook's distance is low for this point reflecting its low leverage). Since there is no good reason to delete this data point, and since (at least qualitatively) our conclusions do not change, it is probably best to go with the full data set. A test of the assumptions for the refit model (with the outlier removed) shows that all are met, and no more outliers are detected. (I won't report these analyses, but you can and should do them just to assure yourself that everything is O.K.)

Case 2 - Size as a function of age (different slopes example)

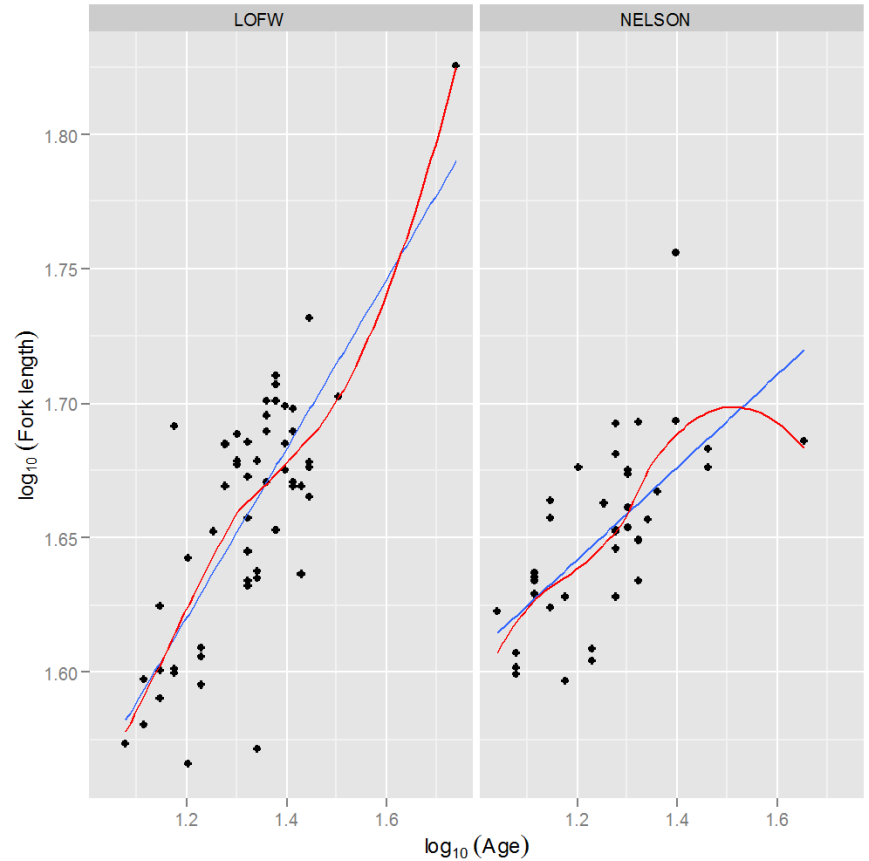
 The file `anc3dat.Rdata` records data on male sturgeon collected at two locations (`locate`), Lake of the Woods, in northwestern Ontario, and the Churchill River in northern Manitoba. Using the same procedure as outlined above (with `locate` as the categorical variable instead of `sex`), test the null hypothesis that the slope of the regression of `lflkl` on `lage` is the same in the two locations. What do you conclude?

```
mydata<-anc3dat

require(ggplot2)
myplot<-ggplot(data=mydata, aes(x=lage,
y=log10(fklngth)))+facet_grid(.~locate)+geom_point()
myplot
myplot<-myplot+
  stat_smooth(method = lm, se=FALSE)+
  stat_smooth(se=FALSE, color="red")

update_labels(myplot, list(
  y = expression(log[10]~(Fork~length)),
  x = expression(log[10]~(Age))
))
```

Figure 2.



```
model.full<-lm(lfkl ~ lage + locate + lage:locate, data = mydata)
```

```
Anova(model.full, type = 3)
```

Anova Table (Type III tests)

Response: lfkl

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	0.62951	1	1078.632	< 2.2e-16 ***
lage	0.07773	1	133.185	< 2.2e-16 ***
locate	0.00968	1	16.591	0.0001012 ***
lage:locate	0.00909	1	15.575	0.0001592 ***
Residuals	0.05136	88		

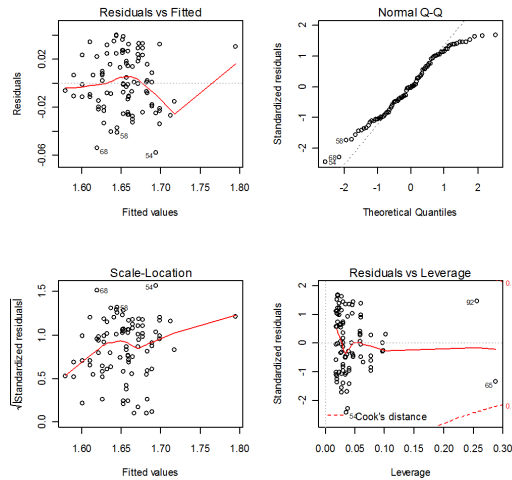
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

In this case, we reject the null hypotheses that (1) the slopes of the regressions are the same in the two locations; and (2) that the intercepts are the same in the two locations. In other words, if we want to predict the fork length of a sturgeon of a particular age (accurately) we need to know from which location it came. The fact that we reject the null hypothesis that the slopes of the `lfkl ~ lage`

regressions are the same in both locations means that we should be doing individual regressions for each location separately (that is in fact what the full model is fitting).

But we are jumping the gun here. Before you can trust these p values, you need to confirm that assumptions are met:

Figure 3.



If you analyze the residuals (in the way described above), you will find that there is no problem with the linearity assumption, nor the homoscedasticity assumption ($BP = 1.2267, p = 0.268$). However, Wilk-Shapiro test of normality of residuals is suspicious ($W=0.97, p = 0.03$). Given the relatively large sample size ($N=92$), this normality test has high power and the magnitude of deviation from normality does not appear to be large. Considering the robustness of GLM to non-normality with large samples, we should not be overly concerned with this violation.

Given that the assumptions appear sufficiently met, we can accept the results as calculated by R. All terms in the model are significant (location, lage, and the interaction). This full model is equivalent to fitting separate regressions for each location. To get the coefficients of these regression, one can either fit the two regressions on data subsets for each location, or extract the fitted coefficients from the full model:

model.full

```
Call:
lm(formula = lfk1 ~ lage + locat e + lage:locat e, data = mydata)
```

Coefficients:

(Intercept)	lage	locat eNELSON
1.2284	0.3253	0.2207
lage:locat eNELSON		
-0.1656		

By default, the variable `locate` in the model is internally encoded as 0 for the location that comes first alphabetically (LofW) and 1 for the other (Nelson). So the regression equations for each location become:

For LofW:

$$l_{fkl} = 1.2284 + 0.3253 \text{ l age} + 0.2207 (0) - 0.1656 (0) (l_{age})$$

$$l_{fkl} = 1.2284 + 0.3253 \text{ l age}$$

For Nelson:

$$l_{fkl} = 1.2284 + 0.3253 \text{ l age} + 0.2207 (1) - 0.1656 (1) (l_{age})$$

$$l_{fkl} = 1.4491 + 0.1597 \text{ l age}$$

You can convince yourself that this is the same as fitting 2 regressions separately:

```
by(mydata, mydata[,"locate"], function(x) lm(lfkl~lage, data=x))
```

```
l o c a t e : L C F W
```

```
Call :
```

```
lm(formula = lfkl ~ lage, data = x)
```

```
Coefficient s:
(Intercept)      l age
      1.2284      0.3253
```

```
-----
l o c a t e : N E L S O N
```

```
Call :
```

```
lm(formula = lfkl ~ lage, data = x)
```

```
Coefficient s:
(Intercept)      l age
      1.4491      0.1597
```

The ANCOVA model

If the test for homogeneity of slopes indicates that the two or more slopes are not significantly different, i.e. there is no significant interaction between the categorical and continuous variable, then a single slope parameter can be fit. How about the intercepts? Do they differ among levels of the categorical variable? There are two school of thoughts on how to proceed to test for equality of intercepts when slopes are equal:

The old school fits a reduced model, with the categorical and continuous variables, but no interactions (this is the ANCOVA model, *sensus stricto*) and uses the partitioned sums of squares to test for significance, say with the `Anova()` function. This approach is the one presented in many statistical textbooks.

Others simply use the full model results, and test significance of each term from the partial sums of squares. This approach has the advantage of being faster as only one model needs to be fitted to make all inferences. However, this approach is less powerful. In most practical cases, it does not matter unless one has very complex models with a large number of terms and higher level interactions and that many of these terms are not significant. My suggestion is that you use the faster approach first, and use the traditional approach only when you accept the null hypothesis for equal intercepts. Why? Since the faster approach is less powerful, if you nevertheless reject H_0 , then this conclusion will not be changed, only reinforced, by using the traditional approach.

Here I will compare the old school and the other approach. Recall that we want to assess equality of intercepts **once we determined that slopes are equal**. Test for equality of intercepts when slopes differ (or, if you prefer, when there is a significant interaction) are rarely directly meaningful, are often misinterpreted, and should rarely be conducted.

Going back to `anc1dat.Rdata`, comparing the relationships between `lfkl` and `lage` among sexes, we obtained the following results for the full model with interactions:

```
Anova Table (Type III tests)
Response: lfkl
      Sum Sq Df F value Pr(>F)
(Intercept) 0.64444 1 794.8182 < 2.2e-16 ***
sex          0.00041 1  0.5043  0.4795
lage        0.07259 1 89.5312 4.588e-15 ***
sex:lage     0.00027 1  0.3367  0.5632
Residuals   0.07135 88
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We already concluded that the slope of the regression for males and females does not differ (the interaction `sex:lage` is not significant). Note that the p-values associated with `sex` (0.4795) is not significant either.

For the old-school approach, one would fit a reduced model (the sensus stricto ANCOVA model):

```
model.ancova<-lm(lfkl ~ sex + lage , data = mydata)
Anova(model.ancova, type = 3)
summary(model.ancova)
```

to get:

```
Anova Table (Type III tests)
Response: lfkl
      Sum Sq Df F value Pr(>F)
```

```

(Intercept) 1.13480 1 1410.1232 <2e-16 ***
sex          0.00149 1    1.8513 0.1771
lage        0.14338 1   178.1627 <2e-16 ***
Residuals   0.07162 89
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Call:
lm(formula = lfk1 ~ sex + lage, data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max
-0.0939916 -0.0184574 -0.0008762  0.0224911  0.0811609


Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.225533   0.032636  37.552 <2e-16 ***
sexMALE     -0.008473   0.006228  -1.361  0.177
lage        0.327253   0.024517  13.348 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02837 on 89 degrees of freedom
Multiple R-squared:  0.696,    Adjusted R-squared:  0.6892
F-statistic: 101.9 on 2 and 89 DF,  p-value: < 2.2e-16

```

According to this test, sex is not significant and therefore we can conclude that the intercept does not vary significantly between males and females. Note that the p-value is lower this time (0.1771 vs 0.4795), reflecting the higher power of this old-school approach. However, the conclusion remains qualitatively the same: intercepts do not differ.

So we accept the null hypothesis that the intercepts are the same for the two sexes. Running the residual diagnostics, we find no problems with linearity, independence, homogeneity of variances, and normality.

 You will notice, in the above analysis that the residuals plots flag three data points (cases 19, 49, and 50) as having high residuals. These points are a bit worrisome, and may be having a disproportionate effect on your analysis. Eliminate these "outliers" and re-run the analysis. Now what do you conclude?

```

model.ancova.nooutliers<-lm(lfk1 ~ sex + lage ,
  data = mydata[c(-49, -50, -19),])
Anova(model.ancova.nooutliers, type=3)
summary(model.ancova.nooutliers)

```

Anova Table (Type III tests)

```

Response: lfk1
            Sum Sq Df  F value  Pr(>F)
(Intercept)  1.09160  1 1896.5204 < 2e-16 ***
sex          0.00232  1    4.0374 0.04764 *
lage        0.13992  1  243.0946 < 2e-16 ***
Residuals   0.04950 86
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Call:
lm(formula = lfk1 ~ sex + lage, data = mydata[c(-49, -50, -19),
])

```

```


Residuals:
    Min       1Q   Median       3Q      Max
-0.0583972 -0.0184692 -0.0009756  0.0206964  0.0402875

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.224000   0.028106  43.549  <2e-16 ***
sexMALE     -0.010823   0.005386  -2.009   0.0476 *
lage        0.328604   0.021076  15.591  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02399 on 86 degrees of freedom
Multiple R-squared:  0.7706,    Adjusted R-squared:  0.7653
F-statistic: 144.4 on 2 and 86 DF,  p-value: < 2.2e-16

```

Well, well. Now we would, according to convention, reject the null hypothesis, and conclude that in fact, the intercepts of the regressions for the two sexes are different! This is a qualitatively different result from that obtained using all the data. Why? There are two possible reasons: (1) the "outliers" have significant impacts on the fitted regression lines, so that the intercepts of the lines change depending on whether the "outliers" are included (or not); (2) the exclusion of the outliers increases the precision, i.e. reduces the standard error of the intercept estimates, and therefore increases the likelihood that the two intercepts will in fact be "statistically" different. (1) is unlikely, since none of the outliers had high leverage (hence Cook's distances were not large), so (2) is more likely, and you can verify this by fitting separate regressions for each sex with and without these three outliers. If you do, you will notice that the estimated intercepts for each sex do not change very much, but the standard errors of these intercepts change quite a lot.

 Fit separate regressions by sex with vs. without the outliers. Pay attention to the intercepts.

```

> by(mydata, mydata[,"sex"], function(x) lm(lfkl~lage,
data=x)) mydata[, "sex"]: FEMALE

Call:
lm(formula = lfkl ~ lage, data = x)

Coefficients:
(Intercept)      lage
  1.2426         0.3143

-----
----- mydata[, "sex"]: MALE

Call:
lm(formula = lfkl ~ lage, data = x)

Coefficients:
(Intercept)      lage
  1.197         0.343

```

```

> by(mydata, mydata[, "sex"], function(x) lm(lfkl~lage,
data=x[c(-49, -50, -19),]))
mydata[, "sex"]: FEMALE

Call:
lm(formula = lfkl ~ lage, data = x[c(-49, -50, -19), ])

Coefficients:
(Intercept)      lage
  1.2403         0.3153

-----
----- mydata[, "sex"]: MALE

Call:
lm(formula = lfkl ~ lage, data = x[c(-49, -50, -19), ])

Coefficients:
(Intercept)      lage
  1.1936         0.3466

```

Difference in intercepts are indeed very small. This suggests that increased precision is sufficient to make the observed effect (difference in intercepts) significant. We didn't get the SE's above.

Let's get these using a common regression model:

```

> model.linear.nooutliers<-lm(lfkl ~ lage , data = mydata[c(-49, -50, -19),])
> summary(model.linear.nooutliers)

Call:
lm(formula = lfkl ~ lage, data = mydata[c(-49, -50, -19), ])

Residuals:
    Min       1Q   Median       3Q      Max
-0.055567 -0.017809 -0.002944  0.021272  0.044972

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.20378     0.02670   45.09 <2e-16 ***
lage         0.34075     0.02054   16.59 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02441 on 87 degrees of freedom
Multiple R-squared:  0.7598,    Adjusted R-squared:  0.7571
F-statistic: 275.2 on 1 and 87 DF,  p-value: < 2.2e-16

> model.linear<-lm(lfkl ~ lage , data = mydata)
> summary(model.linear)

Call:
lm(formula = lfkl ~ lage, data = mydata)

Residuals:
    Min       1Q   Median       3Q      Max
-0.090915 -0.018975 -0.002587  0.021270  0.085273

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.21064     0.03089   39.19 <2e-16 ***
lage         0.33606     0.02376   14.14 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0285 on 90 degrees of freedom
Multiple R-squared:  0.6897,    Adjusted R-squared:  0.6863
F-statistic: 200.1 on 1 and 90 DF,  p-value: < 2.2e-16


```

Dropping these three points did indeed increase the precision with which the intercept was estimated.

Note: It is often the case that by eliminating outliers, new outliers appear. This is simply because the "outlier" designation is usually based on a standardized residual: if you eliminate a couple of outliers, then the residual sums of squares decreases, i.e. the "average" (absolute) residual decreases. Thus, points which were not "far from the average" when the original average residual was comparatively large (i.e. were not "outliers"), may now become so because the average residual has been decreased. Remember also that as you eliminate outliers, N decreases, and the increase in R^2 may be more than compensated for by decreased power. So be wary of eliminating outliers!

Comparing model fits

As we have just seen, the process of fitting models to data is usually an iterative one. That is, there are, more often than not, several competing models that may be used to fit the data and it is left to the analyst to decide which model best balances goodness of fit (which we are usually trying to maximize) and complexity (which we are usually trying to minimize). In general, the strategy to use in regression and anova is to choose a simpler model when doing so does not reduce the goodness-of-fit by a significant amount. R can compute an F-statistic to compare the fit of two models. The null hypothesis in this situation is that there is no difference in goodness of fit between the models.

 Working with the `Anc1dat` data set, compare the fit of the ANCOVA and common simple regression models:

```
anova(model.ancova,model.linear)
```

```
Analysis of Variance Table
```

```
Model 1: l f k l ~ sex + l age
```

```
Model 2: l f k l ~ l age
```

	Res. Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	89	0.071623				
2	90	0.073113	-1	-0.001490	1.8513	0.1771

The **anova()** function can compare the differences in sum of squares and degrees of freedom between the simpler and more complex models, takes the ratio of these two values to generate a mean square, and divides this by the mean square of the more complex model to generate an F-statistic. In the above case, there is insufficient

evidence to reject the null hypothesis and we conclude that the simpler model, which is the simple linear regression, is the best model for these data. (Because these models differ by only the presence vs. absence of a single factor (sex), the P-value is the same as the p-value for sex in model 1.)

Repeat the above procedure with the `ANC3DAT` data, rerunning the full ANCOVA with interaction (`LFKL~LAGE+LOCATE+LAGE:LOCATE`) and without interaction (`LFKL~LAGE+LOCATE`), saving the model objects as you did above. Compare the fits of the two models. What do you conclude?

```
model.full.anc3dat<-lm(lfkl ~ lage + locate + lage:locate,
data = anc3dat)
model.ancova.anc3dat<-lm(lfkl ~ lage + locate , data =
anc3dat)
anova(model.full.anc3dat,model.ancova.anc3dat)
```

Analysis of Variance Table

Model 1: lfkl ~ lage + locate + lage:locate

Model 2: lfkl ~ lage + locate

	Res. Df	RSS Df	Sum of Sq	F	Pr(>F)
1	88	0.051358			
2	89	0.060448	-1	-0.009090	15.575 0.0001592 ***

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

In this case there is sufficient evidence to reject the null hypothesis and conclude that the full model with interaction is the best model to fit to the `Anc3dat` data. This is as we expected, given the fact that we found the interaction to be significant in the original analysis of the data. While no new information is gained from this model comparison in this case, this approach can be more usefully employed to compare nested models that differ in more than one term.

Bootstrap

```
#####
#####
# Bootstrap analysis

# Bootstrap analysis BCa confidence intervals
# Preferable when parameter distribution is far from normal

# Bootstrap 95% BCa CI for regression coefficients
library(boot)

# To simplify future modifications of the code in this file,
# copy the data to a generic mydata dataframe
mydata<-anc3dat

# create a myformula variable containing the formula for the
model to be fitted
myformula<-as.formula(lfkl ~ lage + locate + lage:locate)
```

```

# function to obtain regression coefficients for each iteration
bs <- function(formula, data, indices) {
  d <- data[indices, ]
  fit <- lm(formula, data = d)
  return(coef(fit))
}

# bootstrapping with 1000 replications
results <- boot(data = mydata, statistic = bs, R = 1000,
  formula = myformula)

# view results
results
for (i in 1:length(results$t0)) {
  plot(results, index = i)
}

# get 95% confidence intervals
for (i in 1:length(results$t0)) {
  print(boot.ci(results, type = "bca", index = i))
}

```

Permutation test

```

#####
#####
# Permutation test
#
# using lmperm library

# To simplify future modifications of the code in this file,
# copy the data to a generic mydata dataframe
mydata<-anc3dat

# create a myformula variable containing the formula for the
model to be fitted
myformula<-as.formula(lfkl ~ lage + locate + lage:locate)

require(lmPerm2)

# Fit desired model on the desired dataframe
mymodel <- lm(myformula , data = mydata)

# Calculate p-values for each term by permutation
# Note that lmp centers numeric variable by default, so to
get results that are
# consistent with standard models, it is necessary to set
center=FALSE

```

```
mymodelProb <- lmp(myformula, data = mydata, center=FALSE,  
  perm = "Prob")
```

```
summary(mymodel) summary(mymodelProb)  
summary(aovp(myformula,data=mydata))
```


Lab- Analysis of frequency data: Contingency Tables, Log-Linear Models, and Poisson Regression

After completing this laboratory exercise, you should be able to:

- Create and manipulate data files in R to analyze count data.
- Use R to test an external hypothesis about a particular population using count data.
- Use R to test for independence in two-way tables
- Use R to fit Poisson regression and log-linear models to count data

Organizing the data: 3 forms

Some biological experiments yield count data, e.g., the number of plants infected by a plant pathogen under different exposure regimes, the number of male and female turtles hatched under different incubation temperature treatments (in turtles, sex is temperature dependent!), etc. Usually the statistical issue here is whether the proportion of individuals in different categories (e.g., infected versus uninfected, male versus female, etc.) differs significantly among treatments. To examine this question, we can set up a data file that lists the number of individuals in each category. There are 3 ways to do this. You should be able to decide which one is appropriate, and how to convert between them with R.

The file `USPopSurvey.RData` contains the results of a 1980 U.S. population survey of a mid-eastern town:

```
# Get US Survey data, in "Frequency" form
load(url("http://www.antoinemorin.com/biostat/s/2013/USPopSurvey.RData")) #
online version
```

```
> USPopSurvey
  ageclass sex frequency
1    0-9 female 17619
2   10-19 female 17947
3   20-29 female 21344
4   30-39 female 19138
5   40-49 female 13135
6   50-59 female 11617
7   60-69 female 11053
8   70-79 female  7712
9    80+ female  4114
10   0-9  male 17538
11  10-19  male 18207
12  20-29  male 21401
13  30-39  male 18837
14  40-49  male 12568
```

```

15  50- 59  mal e    10661
16  60- 69  mal e     9374
17  70- 79  mal e     5348
18      80+  mal e     1926
>

```

Note that there are 18 lines and 3 columns in this file. Each line lists the number of individuals (frequency) of a given sex and age class. There are $\text{sum}(\text{USPopSurvey}\$frequency)=239539$ individuals that were classified into the 18 (2 sexes x 9 age classes) categories. This way of presenting data is the **frequency form**. It is a compact way to present the data when there are only categorical variables.

When there are continuous variables, the frequency form can't be utilized (or provides no gain since each observation could possibly have a different values for the continuous variable(s)). Data have therefore to be stored in **case form** where each observation (individual) represents one line in the data file, and each variable is a column. Conveniently, the `vcdExtra` package includes the `expand.dft()` function to convert from the frequency to case form. For example, to create a data frame with 239539 lines and 2 columns (sex and ageclass),:

```

USPopSurvey.caseform<-expand.dft(USPopSurvey, freq="frequency")
head(USPopSurvey.caseform)
ageclass sex
1 0-9 female
2 0-9 female
3 0-9 female
4 0-9 female
5 0-9 female
6 0-9 female

tail(USPopSurvey.caseform)
ageclass sex
239534 80+ male
239535 80+ male
239536 80+ male
239537 80+ male
239538 80+ male
239539 80+ male

```

Finally, these data can also be represented in **table form** (contingency table) where each variable is represented by a dimension of the n-dimensional table (here, for example, rows could represent each age class, and columns each sex), and the cells of the resulting table contain the frequencies. The table form can be created from the case or frequency form by the `xtabs()` command with slightly different syntax:

```

# convert case form to table form
xtabs(~ageclass+sex, USPopSurvey.caseform)
sex
ageclass female male
0-9 17619 17538
10-19 17947 18207
20-29 21344 21401
30-39 19138 18837
40-49 13135 12568
50-59 11617 10661
60-69 11053 9374
70-79 7712 5348

```

```

80+      4114  1926

# convert frequency form to table form
xtabs(frequency~ageclass+sex, data=USPopSurvey)
sex
ageclass female male
0- 9      17619 17538
10- 19    17947 18207
20- 29    21344 21401
30- 39    19138 18837
40- 49    13135 12568
50- 59    11617 10661
60- 69    11053  9374
70- 79     7712  5348
80+      4114  1926
    
```

Figure 1. Tools for converting among different forms for categorical data

From this	To This		
	Case form	Frequency form	Table form
Case form		<code>xtabs(~A+B)</code>	<code>table(A,B)</code>
Frequency form	<code>expand.dft(X)</code>		<code>xtabs(count~A+B)</code>
Table form	<code>expand.dft(X)</code>	<code>as.data.frame(X)</code>	

Graphs for contingency tables and testing for independence

Contingency tables can be used to test for independence. By this we mean to answer the question: Is the classification of observations according to one variable (say, sex) independent from the classification by another variable (say, ageclass). In other words, is the proportion of males and females independent of age, or does it vary among age classes?

The `vcd` package includes a `mosaic()` function useful to graphically display contingency tables:

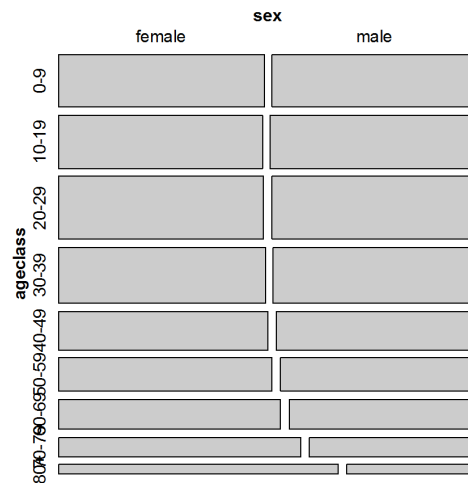
```

library(vcd)

USTable<-xtabs(frequency~ageclass+sex, data=USPopSurvey) # save the table
form as USTable dataframe

# Mosaic plot of the contingency table
mosaic(USTable)
    
```

Figure 2.



Mosaic plots represent the proportion of observations in each combination of categories (here there are 18 categories, 2 sexes x 9 age classes). Categories with a higher proportion of observations are represented by larger rectangles. Visually, one can see that males and females are approximately equal for young age classes, but that the proportion of females increases quite a bit amongst the elders.

The Chi square test can be used to test the null hypothesis that the proportion of males and females does not differ among age classes:

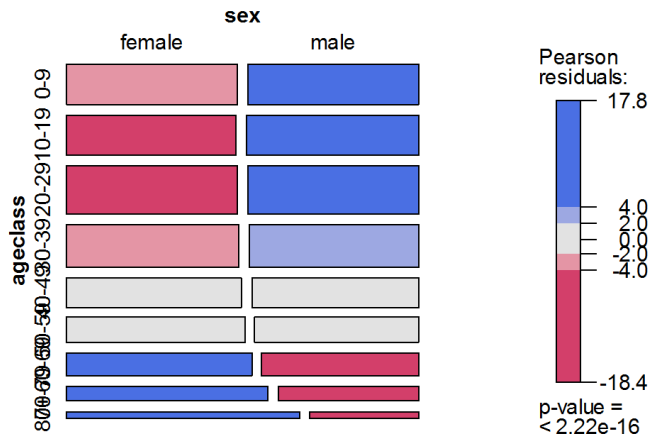
```
# Test of independence
chi sq.test(USTable) # runs chi square test of independence of sex and age class
Pearson's Chi-squared test
data: USTable
X-squared = 1162.584, df = 8, p-value < 2.2e-16
```

From this we conclude there is ample evidence to reject the null hypothesis that `ageclass` and `sex` are independent, which isn't particularly surprising.

The mosaic plot from the `vcd` package can be shaded to show the categories that contribute most to the lack of independence:

```
# Mosaic plot of the contingency table with shading
mosaic(USTable, shade=TRUE)
```

Figure 3.



The shading of each rectangle is proportional to the extent that observed frequencies deviate from what would be expected if sex and age class were independent. The age classes 40-49 and 50-59 have a sex ratio about equal to the overall sex:ratio for the entire dataset, and appear in grey. There are more young males and old females than expected if sex ratio did not change with age, and these rectangles are coded in blue. On the other hand, there are fewer young females and old males than if sex ratio did not change with age, and these rectangles are red coded. Note that the p-value printed on the right of the graph is for the chi-square test that assumes that observations are independent.

The estimation of p-value associated with the chi square statistic is less than ideal when expected frequencies are small in some of the cells, particularly for 2x2 contingency tables. Two options are then preferred, depending on the number of observations. For large samples, like in this example with more than 200,000 cases(!), a Monte Carlo approach is suggested and can be obtained by adding `simulate.p.value=TRUE` as an argument to the `chisq.test()` function:

```
# Monte-carlo estimation of p value (better for large n)
chisq.test(USTable, simulate.p.value=TRUE, B=10000)

      Pearson's Chi-squared test with simulated p-value (based on 10000
      replicates)

data:  USTable
X-squared = 1162.584, df = NA, p-value = 9.999e-05
```

Here, the simulation was done $B=10000$ times, and the chi square value observed with the data was never exceeded so p is estimated as $1/10001=9.999e-05$, which is much larger than the p -value estimated from the theoretical chi square distribution ($p < 2.2e-16$). This difference in p -value is at least partly an artifact of the number of simulations. To estimate p values as small as $1e-16$, at least 10^{16} simulations must be run. And I am not THAT patient.

For small tables with relatively low expected frequencies, Fisher's exact test can be run to test for independence. This result is unbiased if row and column totals are fixed, but is conservative (i.e. it will incorrectly fail to reject the null more often than expected) if row and/or column totals are not fixed.

But this test will fail for large samples, like in this example:

```
> #Fisher exact test for contingency tables (small samples and small tables)
> fisher.test(USTable) # fails here because too many observations
Error in fisher.test(USTable) : FEXACT error 40.
Out of workspace.
```

Log-linear models as an alternative to Chi square test for contingency tables

By now, hopefully, you have learned to appreciate the flexibility and generality of general linear models and you realize that the t -test is a special, simple, case of a linear model with one categorical independent variable. The analysis of contingency tables by chi square test can similarly be generalized. Indeed, generalized linear models for poisson distributed data can be used when the dependent variable are frequencies (count data) and the independent variables can be categorical only (like for contingency tables, these are also called log-linear models), continuous only (Poisson regression), or a combination of categorical and continuous independent variables (this, too is a Poisson regression, but with added categorical variables, analogous to an ANCOVA *sensu lato*).

Such models predict the natural log frequency of observations given the independent variables. Like for linear models assuming normality of residuals, one can assess the overall quality of the fit (by AIC for example), and the significance of terms (say by comparing the fit of models including or excluding particular terms). One can even, if desired, obtain estimates of the parameters for each model term, with confidence intervals and p -values for the null hypothesis that the value of the parameter is 0.

The `glm()` function with the option `family=poisson()` allows the estimation, by maximum likelihood, of linear models for count data.

One "peculiarity" of fitting such models to contingency table data is that generally the only terms of interest are the **interactions**. Going back to the population survey data in frequency form, with sex and ageclass as independent variables, one can fit a glm model by:

```
mymodel <- glm(frequency~sex*ageclass, family=poisson(), data=USPopSurvey)
> summary(mymodel)
```

```
Call:
glm(formula = frequency ~ sex * ageclass, family = poisson(),
     data = USPopSurvey)

Deviance Residuals:
 [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

Coefficients:
            (Intercept)              9.776733  0.007534 1297.730 < 2e-16 ***
sexmale
ageclass10-19      0.018445  0.010605   1.739  0.0820 .
ageclass20-29      0.191793  0.010179  18.842 < 2e-16 ***
ageclass30-39      0.082698  0.010441   7.921 2.36e-15 ***
ageclass40-49     -0.293697  0.011528 -25.477 < 2e-16 ***
ageclass50-59     -0.416508  0.011951 -34.850 < 2e-16 ***
ageclass60-69     -0.466276  0.012134 -38.428 < 2e-16 ***
ageclass70-79     -0.826200  0.013654 -60.511 < 2e-16 ***
ageclass80+       -1.454582  0.017316 -84.004 < 2e-16 ***
sexmale:ageclass10-19  0.018991  0.014981   1.268  0.2049
sexmale:ageclass20-29  0.007275  0.014400   0.505  0.6134
sexmale:ageclass30-39 -0.011245  0.014803  -0.760  0.4475
sexmale:ageclass40-49 -0.039519  0.016416  -2.407  0.0161 *
sexmale:ageclass50-59 -0.081269  0.017136  -4.742 2.11e-06 ***
sexmale:ageclass60-69 -0.160154  0.017633  -9.083 < 2e-16 ***
sexmale:ageclass70-79 -0.361447  0.020747 -17.422 < 2e-16 ***
sexmale:ageclass80+   -0.754343  0.029598 -25.486 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for poisson family taken to be 1)

```
Null deviance: 5.3611e+04 on 17 degrees of freedom
Residual deviance: -1.6116e-12 on 0 degrees of freedom
AIC: 237.31
```

Number of Fisher Scoring iterations: 2

Fitting the full model, with the sex:ageclass interaction, allows the proportion of males and females to vary among ageclass levels, and hence to estimate exactly the frequencies for each combination of sex and ageclass (note that the deviance residuals are all 0's and that the Residual deviance is also approximately zero).

A masochist can use the coefficient table to obtain the predicted values for sex and ageclass categories by summing the appropriate coefficients. The predicted values, like for multiway ANOVA model, are obtained by combining the coefficients. Remembering that the first level of a factor (alphabetically) is used as a reference, here the coefficient for the intercept (9.776733) is the predicted value for the natural log of the number of observations for females in the first alphabetical ageclass (0 to 9). Indeed $e^{9.776733}$ is approximately equal to

17619, the observed number of females in that age class. For example, for males in the 80+ ageclass, calculate the antilog of the coefficient for the intercept (for female in the youngest age class) plus the coefficient for sexmale (equal to the difference between ln frequency of females and males overall), plus the coefficient for the ageclass 80+ corresponding in the difference in frequency on average between the oldest and reference ageclass, plus the coefficient for the interaction terms sexmale:ageclass80+ (corresponding to the difference in the proportion of male for this ageclass compared to the youngest ageclass), so $\ln(\text{frequency})=9.776733-0.004608-1.454582-0.754343=7.5632$, and the frequency is equal to $e^{7.5632}=1926$

Although there are numerous p values in this output, they are not really helpful. To test whether the effect of sex on observed frequency is the same across ageclass levels, i.e. is sex and age are independent, one needs to fit a model where the interaction sex:ageclass is removed, and see how badly this affects the fit.

The Anova function of the car package provides a handy shortcut:

```
> library(car)
> Anova(nymodel, type=3, test = "LR")
Analysis of Deviance Table (Type III tests)

Response: frequency
          LR Chi sq Df Pr(>Chi sq)
sex          0.2   1  0.6657
ageclass 21074.6   8 <2e-16 ***
sex:ageclass 1182.2   8 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The use of type=3 and test="LR" ensures that the test performed to compare the full and reduced models is the Likelihood Ratio Chi-Square using the Residual deviance, and that it is a partial test, not a sequential one.

According to these tests, there is no main effect of sex ($p=0.667$) but there is a main effect of ageclass and a significant sex:ageclass interaction. The significant interaction means that the effect of sex on frequency varies with ageclass, or that the sex ratio varies with age. The main effect of ageclass means that the frequency of individuals varies with age (i.e. some ageclass are more populous than others), The absence of a main effect of sex suggests that there are approximately the same frequency of males and females in this sample (although, since there is an interaction, you have to be careful in making this assertion. It is "true" overall, but appears incorrect for individual age categories).

Testing an external hypothesis

The above test of independence is that of an **internal** hypothesis because the proportions used to calculate the expected frequencies assuming independence of sex and ageclass come from the data (i.e. the overall proportion of males in females in the entire dataset, and the proportions of individuals in each ageclass, males and females combined).

To test the (external) null hypothesis that the sex ratio is 1:1 for the youngest individuals (ageclass 0-9), one has to compute the 2 X 2 table of observed and expected frequencies. The expected frequencies are obtained simply by summing male and female frequencies and dividing by two.

R program to create and analyze a 2x2 table to test an external hypothesis

```
### Produce obs vs exp for 0-9 age class
observed = c(17619, 17538)      # observed frequencies
expected = c(0.5, 0.5)         # expected proportions
chisq.test(x = observed, p = expected, correct=FALSE)
```

Chi-squared test for given probabilities

```
data: observed
x-squared = 0.18662, df = 1, p-value = 0.6657
```

In the past, for 2 X 2 tables Yates's correction was frequently employed (this can be done by changing the above code to: `correct=TRUE`) but it has since been shown to be overly conservative and is no longer recommended (although it doesn't affect the results in this particular instance). Better is a Fisher's exact test if the total number of cases is <200 (which is not the case here), or a randomization.

These data are not particularly good for testing the null hypothesis that the sex ratio at birth is 1:1 because the first age category is too coarse. It is entirely possible that at birth there is an unequal sex-ratio, but there is compensatory age-specific mortality (e.g. more males at birth, but reduced survivorship among males in the first 9 years of life relative to females). In this case, the sex ratio at birth is NOT 1:1, but we still accept the null hypothesis based on age class 0-9.

Poisson regression to analyze multi-way tables

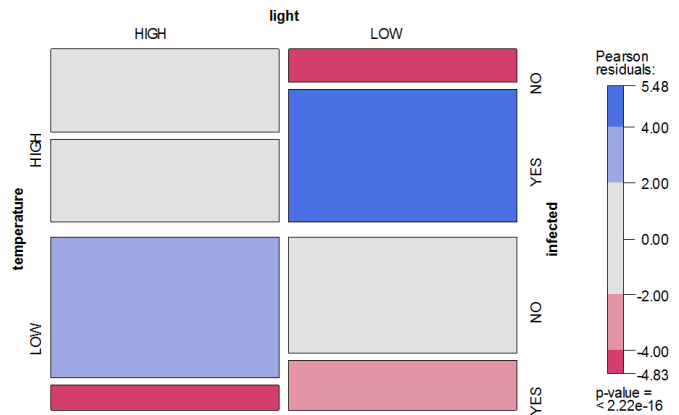
The principle of testing for independence through interactions can be extended to multi-way tables, that is, tables in which more than two criteria are used to classify observations. For example, suppose that we wanted to test the effect of temperature (two levels: high and low) and light (two levels: high irradiance and low irradiance) on the number of plants infected by a plant pathogen (two levels: infected and non-infected). In this case we would need a three-way table with three criteria (infection status, temperature, and light).

Fitting log linear models to frequency data involves testing of different models by comparing them with the full (saturated) model. A series of simplified models is produced, each model missing one of the interactions of interest, and the fit of each simplified model is compared to that of the full model. If the fit does not change much, then the term eliminated does not have much influence on the frequencies, whereas if the resulting model provides a significantly worse fit, then the term is important. As with two-way tables, the terms of interest are the interactions, not the main effects, if what we are testing for is independence of different factors.

The file `loglin.Rdata` contains the frequencies (`frequency`) of infected and non-infected plants (`infected`) at low and high temperature (`temperature`) and low and high light (`light`). To graph the data and determine if infected status depends on light and temperature, one can construct a mosaic plot and a loglinear model.

```
# Convert from frequency form to table form for mosaic plot
loglinTable<-xtabs(frequency~temperature+light+infected, data=loglin)
# Create mosaic plot to look at data
library(vcd)
mosaic(loglinTable, shade=TRUE)
```

Figure 4.



The symmetrical experimental design with the same number of observations made at the two levels of light and of temperature is apparent in the above plot in the overall equal area occupied by the observations in each of the four quadrants. What is of interest, the infected status, appears to vary among the quadrants (i.e. levels of light and temperature). For example, the red rectangles in the lower left and upper right quadrants indicates that there were fewer infected plants at high light and low temperature (bottom left), and fewer uninfected plants at low light and high temperature than if the infected level was not affected by light and temperature. The p-value at the bottom of the color scale represents a test of independence equivalent to testing the full model against a reduced model including only the main effect of temperature, light, and infected status on the (ln) number of observations.

```
# Fit full model
full.model <- glm(frequency~temperature*light*infected, family=poisson(),
data=loglin)

# Test partial effect of terms in full model
require(car)
Anova(full.model, type=3, test="LR")
```

Analysis of Deviance Table (Type III tests)

Response: frequency

	LR Chi sq	Df	Pr (>Chi sq)
temperature	9.1786	1	0.0024487 **
light	13.2829	1	0.0002678 ***
infected	0.0000	1	1.0000000
temperature:light	5.6758	1	0.0172008 *
temperature:infected	29.0612	1	7.013e-08 ***
light:infected	20.2687	1	6.729e-06 ***
temperature:light:infected	1.0840	1	0.2978126

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 'Not significant'
of S-PLUS output


The probabilities associated with each term in the full model are here calculated by comparing the fit of the full model to that of a model with this particular term removed. As is typical in log-linear model analyses, many of the tests here are not interesting. If the biological question is about how infected status varies with other conditions, then the only informative terms are the interaction terms involving infected status.

There are therefore only 3 terms of interest: `temperature:infected`, `light:infected`, and the 3-way interaction `temperature:light:infected`.

- The `temperature:infected` significant interaction implies that infection status is not independent of temperature. Indeed the mosaic plot shows that the proportion of infected cases is higher at high temperature.
- The `light:infected` significant interaction implies that infection status is not independent of light. The mosaic plot also indicates that the proportion of infected plants is larger at low light levels.
- The `temperature:light:infected` interaction is not significant. This implies that the previous 2 effects do not vary between levels of the third variable. So there is no evidence that the effect of light on infection status varies at the two temperatures, or that the effect of temperature on infection status varies between the two light levels. We should therefore drop this term and refit before evaluating the 2-way interactions (small increase in power).

Exercise

We will now work with the `sturgdat` data set to test the hypothesis that number of fish caught is independent of location, year, and gender. Before the analysis, the data will have to be reshaped to be in suitable format for fitting a log-linear model.

 Open `sturgdat.Rdata`, then use the `table()` function to summarize the data according to number of individuals by `sex`, `location`, and `year`. Save this object as `sturgdat.table`. Make a mosaic plot of the data.

```
load(url("http://www.antonio-nemorin.com/biostat/2010/Sturgdat.Rdata")) #
online version, data in case form

# Reorganize data from case form to table form
sturgdat.table <- with(sturgdat, table(sex, year, location))
# display the table
sturgdat.table
```

```

, , location = CUMBERLAND
      year
sex   1978   1979   1980
FEMALE 10    30    11
MALE   14    14     6

, , location = THE_PAS
      year
sex   1978   1979   1980
FEMALE 5     12    38
MALE  16    12    18

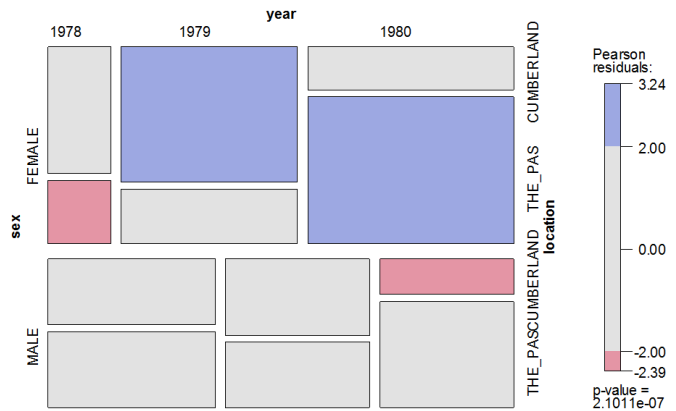
# Create data frame while converting from table form to frequency form
sturgdta.freq<-as.data.frame(sturgdta.table)
#display data frame
sturgdta.freq

  sex      year location Freq
1 FEMALE 1978  CUMBERLAND  10
2 MALE   1978  CUMBERLAND  14
3 FEMALE 1979  CUMBERLAND  30
4 MALE   1979  CUMBERLAND  14
5 FEMALE 1980  CUMBERLAND  11
6 MALE   1980  CUMBERLAND   6
7 FEMALE 1978  THE_PAS     5
8 MALE   1978  THE_PAS    16
9 FEMALE 1979  THE_PAS    12
10 MALE  1979  THE_PAS    12
11 FEMALE 1980  THE_PAS    38
12 MALE  1980  THE_PAS    18

# Look at the data as mosaic plot
library(vcd)
# mosaic using the table created above
mosaic(sturgdta.table, shade=TRUE)

```

Figure 5.



Using the frequency form of the table, fit the full log-linear model just as we did with the `loglin` data set and produce the anova table with chi square statistics for the terms in the model. Is the 3-way interaction significant (`location:year:sex`)? Does sex ratio change between locations or among years?

Fitting the full, three-way model:

```
# Fit full model
full.model <- glm(Freq~sex*year*location, data=sturdat.freq, family="poisson")
summary(full.model)
library(car)
Anova(full.model, type=3)
```

```
Call:
glm(formula = Freq ~ sex * year * location, family = "poisson",
     data = sturdat.freq)
```

```
Deviance Residuals:
 [1]  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Coefficients:

value	Pr(> z)	Estimate	Std. Error	z
(Intercept)		2.30259	0.31623	
7.281	3.3e-13 ***			
sexMALE		0.33647	0.41404	
0.813	0.41641			
year 1979		1.09861	0.36515	
3.009	0.00262 **			
year 1980		0.09531	0.43693	
0.218	0.82732			
locationTHE_PAS		-0.69315	0.54772	-
1.266	0.20569			
sexMALE : year 1979		-1.09861	0.52554	-
2.090	0.03658 *			
sexMALE : year 1980		-0.94261	0.65498	-
1.439	0.15011			
sexMALE : locationTHE_PAS		0.82668	0.65873	
1.255	0.20950			
year 1979 : locationTHE_PAS		-0.22314	0.64550	-
0.346	0.72957			
year 1980 : locationTHE_PAS		1.93284	0.64593	
2.992	0.00277 **			
sexMALE : year 1979 : locationTHE_PAS		-0.06454	0.83986	-
0.077	0.93875			
sexMALE : year 1980 : locationTHE_PAS		-0.96776	0.87942	-
1.100	0.27114			

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

(Dispersion parameter for poisson family taken to be 1)

```
Null deviance: 5.7176e+01 on 11 degrees of freedom
Residual deviance: 4.6629e-15 on 0 degrees of freedom
AIC: 77.28
```

Analysis of Deviance Table (Type III tests)

Response: Freq

	LR Chi sq	Df	Pr(>Chi sq)
sex	0.6698	1	0.4131256
year	13.8895	2	0.0009637 ***
location	1.6990	1	0.1924201
sex:year	4.6930	2	0.0957024 .
sex:location	1.6323	1	0.2013888
year:location	25.2580	2	3.276e-06 ***
sex:year:location	1.6677	2	0.4343666

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

This is a three-way table, with three factors: sex, location and year. Thus, the “saturated” or “full” loglinear model includes 7 terms: the three main effects (sex, location and year), the three 2-way interactions (sex:year, sex:location and year:location) and the one 3-way interaction (sex:year:location). The null deviance is 57.17574, the residual deviance of the full model is, not surprisingly, 0. The deviance explained by the three-way interaction, 1.66773 (which

is, in fact, a chi square statistic with two degrees of freedom), is not significant, and we are therefore justified in fitting the model without this term.

What does this mean? It means that, if there are any 2-way interactions, they do not depend on the level of the third variable. For example, if indeed the sex ratio of sturgeon varies among years (a `sex:year` interaction), that it varies in the same manner at the two locations. This in turn means that in testing for two-way interactions, we are (statistically) justified in pooling (summing) over the levels of the third variable. This is analog to what can be done in multiway ANOVA when high order interactions are not significant. For example, in testing for a `sex:location` effect, we can pool over `year`, to produce a 2 X 2 table whose cell counts are the total number of sturgeon of a given sex at a given location captured over the three years 1978-1980. By increasing cell counts, we increase statistical power, which is desirable.

☞ If we adjust the model without the 3-way interaction, we get:

Analysis of Deviance Table (Type III tests)

Response: Freq

	LR Chi sq	Df	Pr (>Chi sq)
sex	1.8691	1	0.1715807
year	15.1289	2	0.0005186 ***
location	1.5444	1	0.2139568
sex:year	15.5847	2	0.0004129 ***
sex:location	2.1762	1	0.1401583
year:location	28.3499	2	6.981e-07 ***

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

We can see that the `sex:location` interaction does not explain a significant portion of the deviance, whereas the two others do. Sex ratio does not vary among locations, but it does among years. The `year:location` is also significant (see below for its meaning).

Should you try to simplify the model further? Real statisticians are divided on this question. All agree that keeping insignificant terms in the model may cost some power. On the other hand, removing non significant interactions can lead to difficulty interpreting answers when observations are not well balanced (i.e. there is colinearity among model terms).

☞ Refit the model, this time excluding the `sex:location` interaction.

Analysis of Deviance Table (Type III tests)

Response: Freq

	LR Chi sq	Df	Pr (>Chi sq)
sex	5.0970	1	0.0239677 *
year	16.1226	2	0.0003155 ***
location	0.2001	1	0.6546011

```
sex:year      13.9883  2  0.0009173 ***
year:location 26.7534  2  1.551e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now the remaining two interactions are significant. It looks as though this is the "best" model.

On the basis of the above analysis, the simplest model is:

$$\ln[f_{(ijk)}] = \text{location} + \text{sex} + \text{year} + \text{sex:year} + \text{location:year}.$$

How are these effects interpreted biologically? Remember, as in tests of independence, we are not interested in main effects, only the interactions. For example, the main effect `location` tells us that the total number of sturgeon caught (pooled over both sexes and all years 1978-1980) varied between the two locations. This is not surprising and uninteresting given that we have no information on the sampling effort. However, the `sex:year` interaction tells us that over the 3 year period, the sex-ratio of the harvest changed, and it changed in more or less the same fashion in the two locations, which is a rather interesting result. The `location:year` effect tells us that the total number of sturgeon harvested not only changed over the years, but that this change varied between locations. This could be caused by a different fishing effort at one station over the years, or to a negative impact at one station only on one year. Whatever the cause, it affected males and females similarly since the 3 way interaction is not significant.