Bio220 Lab Manual

Table Of Contents

Chapter 1: Introduction to R

Every quantitative experiment produce some numbers in the end. We need statistics to make sense of the data we collected. With the current amou nt of biological data and the rate of data production, we need computers to be able to process and make sense of the data from our experiments.

Programming languages are our way of communication with the computer to get it do what we want to be done. **R** is a programming language that is especially suitable for data manipulation and statistical analyses. It is commonly used in the field of Biology and so, there are a lot of high quality library packages written in R, which enable us do some specific analyses without having to write the whole code by ourselves. It is free and available on GNU/Linux, MacOS and Windows platforms.

Connect to https://cran.ncc.metu.edu.tr/ to download the installation file appropriate for your operating system, and install it using instructions given at the same web page.

Overview of RStudio Graphical User Interface

RStudio is a free and open-source integrated development environment (IDE). It is basically an interface between R and us. It makes R much easier and more convenient to use.

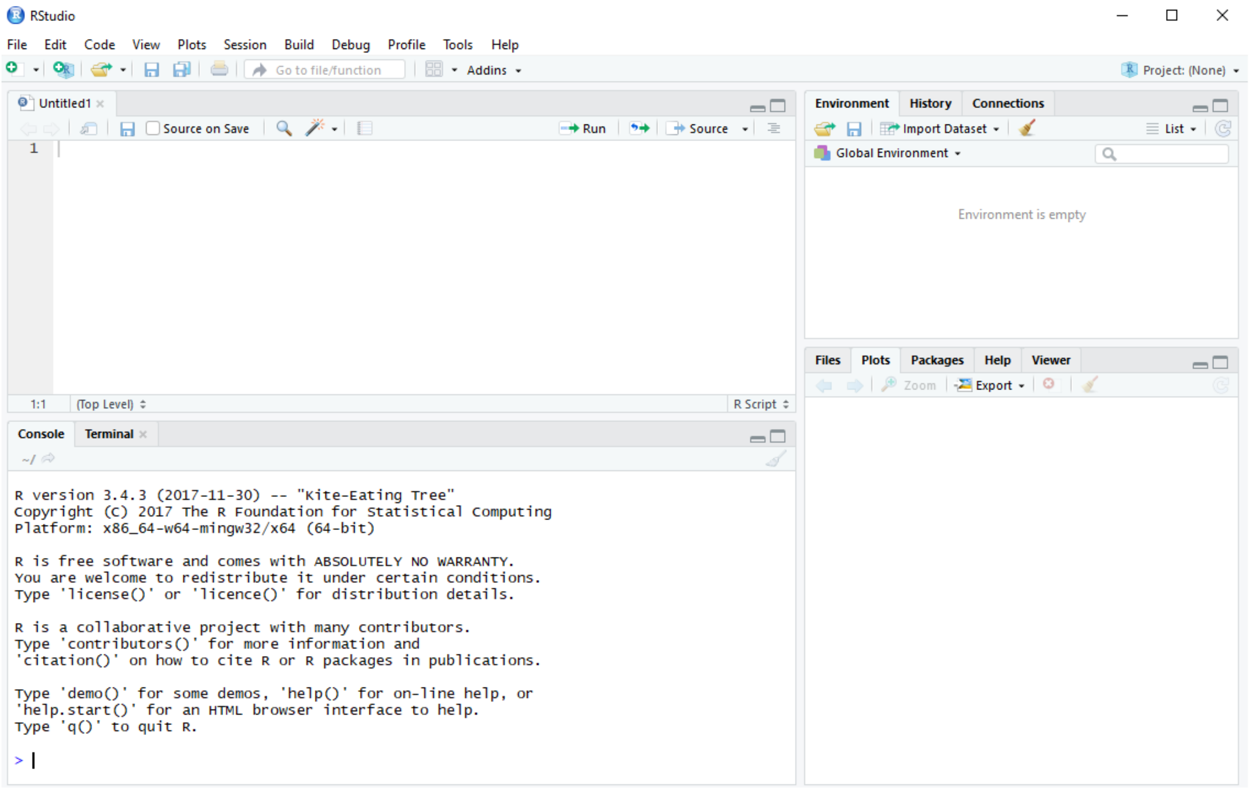
Download RStudio from https://www.rstudio.com/. Note that RStudio requires R pre-installed!

RStudio interface consists of 4 windows by default. The *upper left* window is a script editor. Script is a text file that contains code for a programming language. So, basically that window is like notepad of R. Multiple scripts can be opened in tabs. Codes can be written in script files and stored/modified to be used later.

The *lower left* window contains the console where we can write our code and get the immediate output. It is basically the naked R. We won't use other tabs in this window in this course.

The *upper right* window is the environment, where we will see the objects we create in following sections. Again, other tabs in that window will be rarely used.

The *lower right* window has multiple useful tabs including showing the plots that we will draw and user manuals of functions.



```
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

Working on the Console

Console is the place where the codes are executed and their results are printed. We can see the console on lower left window of default RStudio window.

For example, as a simple code, we can use *arithmetic operations* to use R like a calculator.

```
2+2  # Addition
2-2  # Substraction
2*2  # Multiplication
2/2  # Division
2^2  # Exponent
```

An **expression** is a combination of code elements that are ran together. It is usually a line of code, that may consist of *values*, *operators*, *functions* and *data objects*.

Example 1

- Calculate the following by typing the expression to the console and pressing enter.
  - `2+2/3`
  - `5^(19*4)`
- Observe how the input (code) and output (result) lines start differently.

If an incomplete expression is written, R console will prompt a new line starting with a + sign. You should either complete your expression or press Esc to cancel.

Objects to Store Data

In previous example we got our result printed on console. But in many cases we need to use the computed value[1] again. So, we will store values in **objects**[2] using the *assignment operator* =. [3]:

```
= # assignment operator
my_result = 2+2
```

Note that unlike the mathematical equal sign, = assignment operator is directional. Characters written to the left of the = sign is the object's name and to the right hand side is its value. Right hand side will be calculated first and then will be assigned to the left hand side after that.

If the object name is not used before in the current session, an object with that name will be created (then you can see its name in the 'Environment' tab). If an object with that name already exists, it will be *overwritten*. It will now have the newly assigned value and previous value will be lost. Object names can't start with a number or underscore. Similarly, some special words/symbols that are already meaningful for R can't be object names.

We can use the name of the object in the code instead of writing the value itself. To see the value of an object on the console, just write the name of the object and hit enter.

Example 2

- Assign 2 to an object called a.
- Create an object called b, with the value a times 5.
- Print value of b.
- Set value of a as 3.
- Print values of a and b again to check whether b has changed or not.

Functions and Operators

R comes with a diverse set of built-in **functions** for data manipulation, mathematical calculations, producing graphics and statistical analyses. Functions take input(s) an produce output(s). **Arguments** are the components of the input. For an imaginary function called 'fun', which has 2 arguments, usage is as follows:

```
fun(argument1=value1, argument2=value2)
```

Some functions don't have any arguments, but parantheses should still be there. Most functions need 1 or more arguments. If you know the order of the arguments, you don't need to write the argument name, such as:

[1] Objects can store anything in them, from single numbers to large datasets and even data in special (not designed to be read by humans) formats to be processed by R

[2] More specifically they are called variables, but to prevent confusion with a statistical variable we will simply call them objects.

[3] <- is another assignment operator. '=' causes less syntax errors. So we prefer that one

```
fun(value1, value2)
```

If the argument has a default value, it is not necessary to write the value. For the same imaginary function if argument 2 has a default value we can use like:

```
fun(argument1=value1) #or
fun(value1)
```

Often, the first argument of a function carries the data, and is called 'x'. Other arguments specify how the data will be used and have descriptive names. To get info about a function, its arguments and if they have default values or not, we can always look at the function's help page. To open the help page write function name after '?' and hit enter like `?fun`

If functions are within an expression, its output value is used *on that place*. We can assign output of a function to an object. We can also give functions as input for another function (function inside a function).

Here are some functions to be used this week:

```
ls()                 # List objects in current environment
rm()                 # Remove objects given as arguments
log(x= , base= )     # Calculate logarithm of `x` in base `base`
round(x= , digits= ) # Rounds `x` to `digits` digits after decimal point
sum()                # Sums all the values given as arguments
```

Example 3

- List the objects we created in this session
- Calculate log 5 without specifying the base
- What is the default base in this log function?
- Calculate log `a` (from previous example) on base 10
- Round your calculation to 2 decimals.
- Did the value of `a` changed?
- What can we do if we want to keep rounded value of `a`?

Operators can be thought as simple functions specified with special symbols. We have already seen the *arithmetic operators*. There are two other operator types we use frequently.

Comparison operators (also called relational operator) compares two values and tells if the given relation is `TRUE` or `FALSE`.

```
x1 == x2    # equal
x1 <  x2    # less
x1 >  x2    # greater
x1 <= x2    # less than or equal
x1 >= x2    # greater than or equal
```

Logical operators work on `TRUE`/`FALSE` type of values (may be called logical or Boolean type) and produce the same type of output based on the *truth table* given below. We will use 3 of these operators: *AND* (`&`), *OR* (`|`), *NOT*(negation) (`!`).

AND	Result	OR	Result	Negation	Result
`TRUE & TRUE`	`TRUE`	`TRUE | TRUE`	`TRUE`	`! TRUE`	`FALSE`
`TRUE & FALSE`	`FALSE`	`TRUE | FALSE`	`TRUE`	`! FALSE`	`TRUE`
`FALSE & FALSE`	`FALSE`	`FALSE | FALSE`	`FALSE`		

Several comparison and logical operators can be used in a single expression. Comparison operators are evaluated before logical operators. Still, using parentheses to specify precedence is better to improve readability of the code.

Example 4

- Let us define 4 numeric objects `x1`, `x2`, `x3`, `x4` as `log(204, base = 9)`, `log(402, base = 7)`, `log(954, base = 6)`, `log(866, base = 5)`, respectively.
- Check if `x1` is greater then `x2`.
- Check if `x3` is less then or equal to `x4`.
- Check if both of the conditions above hold true.
- Check if at least one of the conditions hold true.
- Check if first condition is true and second one is false.

Using R Scripts

Script is a text file that contains code for a programming language. Up to now we used R on console, where we ran single line of codes. Working on console quickly gets messy if you need multiple lines of (potentially dependent) codes. We need an editor like notepad to organize and edit the expressions. On the upper left part of the default RStudio window, there is a text editor that lets us write code as a script.

Code written in a script should be *executed* to take effect. To execute the current line, press the 'Run' icon at the top right of the tab or press Ctrl+Enter. To execute specific part of the code, select the part with cursor and run. The executed (evaluated) code will be printed on the console as well.

Example 5

- Open a new R script (`File->New File->R Script`)
- Write codes for the following:
  - Assign a number, let's say `39` to an object called `x`
  - Calculate logarithm of `x` on base `8` and assign the result to an object called `y`
- Check using `ls()`, if there are any objects called `x` or `y`
- Execute the codes in your script:
  - Select your lines of code with the mouse cursor
  - Press `Ctrl+Enter` or click `Run` on top of the script editor
- Check your environment again
- Now erase the first line from your script and check your environment again
- Save your script (`File->Save` or by pressing `Ctrl+S`)
  - You have to give your script a name and decide where to save it in your computer
- Later you can open your file (`File->Open File`) or `Ctrl+O`

A good script should also contain short explanations of what the code does. For that purpose we put **comments** in our scripts. A comment starts with a hash sign `#`. In any line, text that comes after `#` are not considered as code by R, hence we can write anything there. You can see that we have already used comments when introducing new codes.

Data Types and Structures

Data type defines what kind of values are within data and *data structure* organize a collection of values in a specific way.

Vector is the fundamental data structure in R. The objects we used so far are actually vectors with a single element. A vector is a 1 dimensional object holding an ordered set of values of a single type. `c()` function can be used to create a vector.

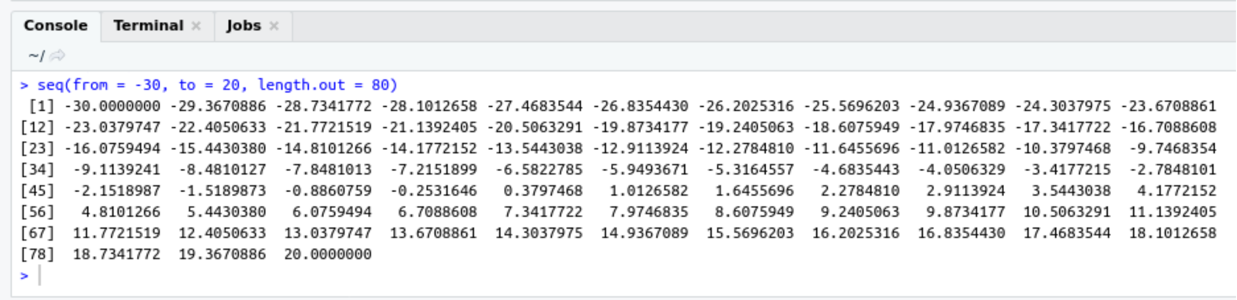
```
c(3, 5, 7, 11, 12)                     # a numeric vector
c("A","G","T","C")                     # a character vector
c(TRUE, FALSE, FALSE)                  # a logical vector
factor(c("a","b","b","c","a","b"),  # an object of class factor
       levels = c("a","b","c","d")) # with manually specified levels
```

A vector can be of types **integer**, **numeric**, **character** or **logical**.[4]

Integer types can be converted to **numeric** (real number) automatically without causing any problems. For that reason we will treat them as a single type and call both of them numeric. Numeric vectors that have a pattern can be created using the following functions:

```
4:15                            # numbers from 4 to 10 with increments of 1
8:-6                            # numbers from 8 to -6 with decrements of 1
seq(from= , to= , by= )         # defining sequence by increments/decrements
seq(from= , to= , length.out= ) # defining sequence by total number of elements
```

A vector looks as below in console output. Note the numbers in square brackets at beginning of each line, they show the index (order/rank) of the element in the beginning of the respective line.



```
> seq(from = -30, to = 20, length.out = 80)
 [1] -30.0000000 -29.3670886 -28.7341772 -28.1012658 -27.4683544 -26.8354430 -26.2025316 -25.5696203 -24.9367089 -24.3037975 -23.6708861
[12] -23.0379747 -22.4050633 -21.7721519 -21.1392405 -20.5063291 -19.8734177 -19.2405063 -18.6075949 -17.9746835 -17.3417722 -16.7088608
[23] -16.0759494 -15.4430380 -14.8101266 -14.1772152 -13.5443038 -12.9113924 -12.2784810 -11.6455696 -11.0126582 -10.3797468  -9.7468354
[34]  -9.1139241  -8.4810127  -7.8481013  -7.2151899  -6.5822785  -5.9493671  -5.3164557  -4.6835443  -4.0506329  -3.4177215  -2.7848101
[45]  -2.1518987  -1.5189873  -0.8860759  -0.2531646   0.3797468   1.0126582   1.6455696   2.2784810   2.9113924   3.5443038   4.1772152
[56]   4.8101266   5.4430380   6.0759494   6.7088608   7.3417722   7.9746835   8.6075949   9.2405063   9.8734177  10.5063291  11.1392405
[67]  11.7721519  12.4050633  13.0379747  13.6708861  14.3037975  14.9367089  15.5696203  16.2025316  16.8354430  17.4683544  18.1012658
[78]  18.7341772  19.3670886  20.0000000
> 
```

Character type is any text in quotation marks. When in quotation marks, numbers can also be stored as character type and they lose numeric properties, they will be treated as text/character.

[4] There are other types, namely "raw" and "complex", that we will never need in the scope of this course.

Logical type is `TRUE` and `FALSE` values. Although we can define a logical vector by writing `TRUE`/`FALSE` values within a `c()` function, most of the time we get this type of vector as a result of a comparison operation.

Logical and numeric data types are interconvertible. In arithmetic expressions `FALSE` is 0 and `TRUE` is 1. Conversely, in logical expressions, 0 is `FALSE`'and other numbers are `TRUE`.

In R, **categorical** variables can be processed in an object of class **factor** that stores values and levels (list of all possible values) as opposed to character vectors that only stores values. `factor()` function can be used to create a factor. First argument is the values to be stored (in vector form). Optionally, levels of the factor can be specified with the `levels` argument. If we do not specify the levels, R will take the unique set of values as levels. # Each object in R has some *attributes*, that provide information about the object. We may be interested in 3 attributes of the vectors: `class`, `names`, `length`.

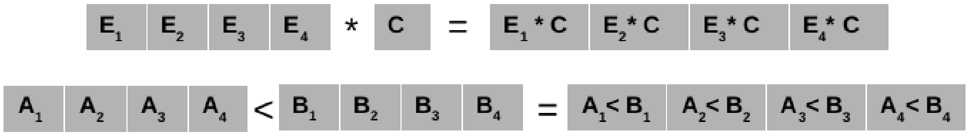
Each object has a **class**. Class of the fundamental object vector shows what type of data it stores. More complex objects like `factor`, `matrix` and `data frame` have their own class. `class()` function shows the class of an object.

Each element in a vector or factor can have a name. *Names* attribute show the names of the elements. By default these objects come with empty names, that is elements are not named but indexed by numbers. When a vector is printed in R console, each line starts with the index number of the first element on that line. If we have to give names to elements, we can use the `names()` function and pass names in vector format.

Another useful attribute is the *length* of the vector/factor. Using the `length()` function we can see how many elements a vector/factor has.

```
head(x)         # prints first few elements of an object,
                # by default first 6
class(x)        # shows class of the input object
length(x)       # shows number of elements in a 1-dim object
names(x)        # shows names of elements in a 1-dim object
names(x) = c("a","b","c")     # gives names a, b, c to elements
                              # of the object x
```

Operators work on vectors element-by-element. If a binary operation (operation that takes 2 inputs such as arithmetic operations we have seen) is given a vector and a single value, operation takes place between each element of the vector and that single value. If an operation is applied to two vectors of the same size, operation takes place between each respective element of the vectors with the same index number. These two cases are schematically shown below. If two vectors of different lengths are paired as operands, things may get complicated and they may behave differently for different functions/operators.


E_1 E_2 E_3 E_4 * C = E_1 * C E_2 * C E_3 * C E_4 * C
A_1 A_2 A_3 A_4 < B_1 B_2 B_3 B_4 = $A_1 < B_1$ $A_2 < B_2$ $A_3 < B_3$ $A_4 < B_4$

Example 6

vnames	ob1	ob2	ob3	ob4	pd1	pd2	pd3
v1	6.0	6.8	5.6	7.9	6.7	5.8	5.1
v2	4.5	4.7	6.7	6.2	7.1	5.5	6.9

- Create 2 separate vectors with the values given in the table above and store them in objects named `v1` and `v2`.
- Create another vector with the given character values and store it in an object called 'vnames'.
- Check classes and lengths of these 3 vectors.
- Assign values in `vnames` as names of the elements of both `v1` and `v2`.
- How many of the elements of `v1` is greater than the corresponding element in `v2`?
- How many of the elements of `v1` is greater than the corresponding element in `v2` and less than 6?
- In how many observations, at least one of the values from `v1` and `v2` is less than 6?
- Calculate natural logarithm of the values in `v1` and `v2`. How many of the `log(v2)/log(v1)` ratios are greater than 1?

Matrix is a 2 dimensional object storing values of a single type, whose places are determined by 2 index numbers, first one specifying the row number and second one specifying column number of the element. It can be generated using the function `matrix()` or combining existing vectors with `cbind()` or `rbind()`.

A matrix have attributes similar to a vector. Its class can be seen with the function `class()`. In order to see the type of values it stores, we need to check class of its columns, which we will see next week. As matrix is 2 dimensional, a length is not defined, instead number of columns (`ncol()`) and number of rows (`nrow()`) or both of them (`dim()`) can be retrieved.

Similar to vector, row names and column names of matrix is empty by default. Instead of names, index number are used to specify the elements. If needed, names can be assigned to rows and columns using the functions `rownames()` and `colnames()`.

```
matrix(c(1,2,3,4,5,6),        # create a matrix of values 1 to 6
       nrow = 2, ncol = 3)    # with 2 rows and 3 columns
                              # columns are filled first
cbind(v1, v2)                 # combine vectors column-wise
rbind(v1, v2)                 # combine vectors row-wise
dim(m1)          # get the number of rows and columns (in that order)
nrow(m1)         # get the number of rows of m1
ncol(m1)         # get the number of columns m1
rownames(m1) = c("r1","r2")      # assign row names to m1
colnames(m1) = c("c1","c2","c3") # assign column names m1
data.frame(v1, v2, v3) # create a data frame
                       # with given vectors as its columns
```

Data frame is another 2 dimensional object, better suited for storing data because it can store data of different types in its columns. The `data.frame()` function can be used to create a data frame from existing vectors. Within a column, different data types are not allowed and length of each column should be the same.

Attributes of data frame is almost the same as matrix, with slight differences in subsetting that we will see next week. We can see number of rows and columns and assign row names and column names of a data frame the same way we do for a matrix.

Example 7

- Combine `v1` and `v2` from previous example column-wise. Check class and dimensions of the new object.
- Combine `v1` and `v2` row-wise. Check class and dimensions of the new object.

- Create a data frame using `v1`, `v2` and a categorical variable which shows whether data is a new observation or previous data. “ob” stands for new observation and “pd” stands for previous data in the element’s names. Store this data frame in an object called `df1`.
- Can we represent it in 2 columns such that previous observations are in a column and new ones in another column? What prevents us from doing that?

a	a	a	b	b	b	c	c	c
6.4	5.3	7.3	7.4	7.4	4.7	5.8	5.7	6.1

- For the table above, create 2 data frames called `df2` and `df3` that represent data in 2 different ways. `df2` should have numerical values in one column and categorical values in another column. `df3` should consist of numerical values only and values that belong to different category levels should be in separate columns.
- Give meaningful names to rows and columns of `df2` and `df3`.

Exercises

1. Bacterial growth Suppose number of bacteria in a medium change according to the following formula:

$x_t = x_0 * (1 + r)^t$,

where t denotes number of generations, r the growth rate, x_0 initial number of bacteria and x_t number of bacteria at generation t.

- Consider a bacterial strain with an initial size 20 and growth rate 0.4. Calculate number of bacteria throughout first 8 generations. Store the bacteria numbers in a vector.
- Consider another bacterial strain with an initial size 10 and growth rate 0.6. Calculate number of bacteria throughout first 8 generations and store the bacteria numbers in another vector.
- Calculate the difference in population sizes of these strains for each generation.

2. Amplification of rounding error

- Calculate $\dfrac{(x-y)^2}{(x-y)^3}$, where $x = 3.79 * 2^{-0.84}$ and $y = 1.88 * log_2 2.18$. Round the result to 2 digits after decimal point and store it as `true_result`.
- Round x and y to 2 decimal places before the calculation and use the rounded values for the same calculation above. Store the result in an object called `early_rounded`.
- How much does `early_rounded` deviate from the `true_result`?

3. Exchanging values

- If you need to exchange values of two objects in such a situation that you can not memorize or re-compute those values, how can you do it? Show it on the x and y objects from previous exercise.

Hint: You can use a third object.

4. Difference or ratio? A study reports recovery rates of patients 4 days after an infection. Patients are infected with either one of six different bacterial strains and for each strain, a group of patients get an antibiotic treatment and a group did not get the treatment. Data shows ratio of recovered people.

	treatment	control
strain1	0.82	0.42
strain2	0.63	0.15
strain3	0.11	0.05
strain4	0.15	0.17
strain5	0.54	0.58
strain6	0.93	0.91

- Calculate ratio of treatment recovery to control recovery for each strain. That is, ratio of the recovery ratios.
- Calculate difference of treatment recovery from control recovery for each strain. That is, difference of the recovery ratios.
- Can we say that one of the measures is more informative than the other?
- If treatment recovery is higher than control recovery, we can not know whether there is evidence for effectiveness of the treatment, without making a statistical test. However if treatment recovery rate is lower than control, we can say that there is no evidence that treatment is effective. For how many of the strains, we can already say that treatment is not effective?
- In how many cases recovery rate in treatment group satisfies both; 0.2 greater than control and more than twice of control?

- When dealing with proportion data, logarithm of the data is commonly used in analysis. To recall arithmetics of logarithms, let us check that logarithm of ratio of two values equals to difference of logarithms of those two values.
  - Calculate log(treatment/control) and save the result.
  - Calculate log(treatment)-log(control) and save.
  - Check whether the results are equal or not.

5. Clutch size In 4 different regions, 3 nests of a bird species are checked and number of eggs are counted.

- In a data frame called `clutch_wide` store the following data.

region1	region2	region3	region4
2	2	1	4
3	6	2	5
1	8	1	7

- In another data frame called `clutch_tall` store the following data.

region	clutch_size
r1	2
r1	3
r1	1
r2	2
r2	6
r2	8
r3	1
r3	2
r3	1
r4	4
r4	5
r4	7

- Note that these 2 data frames store the same data in different representations. What are the pros and cons of these representations?
- How many nests have number of eggs between 3 and 6 (including 3 and 6)?
- How many nests are from regions 1 or 3 and have more than 2 eggs?

6. Some small tasks **a.**

Calculate below expression:

$4.6^2 * (8 - \frac{7}{18})^{0.8} - \log_7 32$

b.

Create an object called "excalibur_17" whose value is $\frac{\sqrt{289}}{2^{-5} * 17}$

c.

Create another object called "kek" with value is the result of following equation:

$(\frac{excalibur\_17}{8} - \log_2 34)^5$

d.

What is the result of:

$\frac{\sqrt{43}}{excalibur\_17} + \log_2{(kek^2)}$

e.

What is the sum of following elements:

$5, -17, 23, 8, kek^2, kek^3, 14.3$

f.

create the character object called “totoro” with value “mei”.

g.

check the class of all object you created (“excalibur_17”, “kek”, “totoro”)

h.

check whether “excalibur_17” is smaller than “kek” object

i.

What is the result of following expression and explain why

```
(TRUE + FALSE)*(FALSE+TRUE)
```

j.

Compare two expressions to check whether they are equal:

$"K" \text{ , } "k"$

k.

create variables “x” with value of `7` and “y” with value of `0`. What are the results of below expressions and explain why:

```
x & y
x & (y+1)
(x-7) | y
```

Chapter 2: Manipulating Data

This week, we will be learning how to import and export data using various acceptable file formats, and then how to manipulate and edit data that we have at hand whether that data is stored as a `vector`, `matrix` or a `data.frame`.

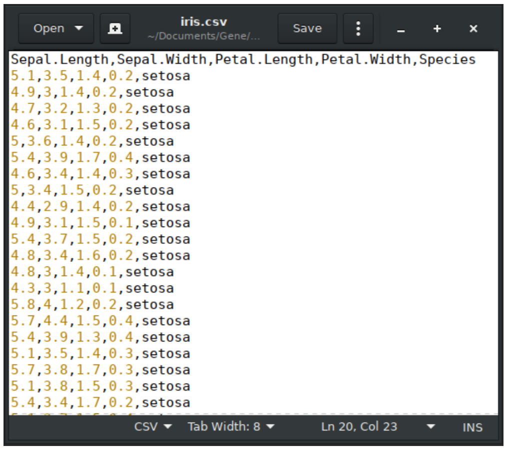
Importing and Exporting Data using `R`

From this week on, we will be dealing at least one data importing event in every week, since most of the time we will be using data from an external source to learn statistical concepts and perform analyses on. Because of that it is essential for us to learn data import/export process using `R` very well, but don't worry: there is not much to learn.

Before we start exercising about this topic, it is important that we first learn different file formats that `R` can read from and write to.

`.csv` file format

`.csv` file extension is an abbreviation and stands for "comma separated values". Other than columns in each line being delimited (separated, distinguished) by comma characters (`,`), it is actually a plain-text file, such that you can easily open them by Notepad or any other text editors. Below is a representation what you see if you open a `.csv` file as a plain text



```
Sepal.Length,Sepal.Width,Petal.Length,Petal.Width,Species
5.1,3.5,1.4,0.2,setosa
4.9,3,1.4,0.2,setosa
4.7,3.2,1.3,0.2,setosa
4.6,3.1,1.5,0.2,setosa
5,3.6,1.4,0.2,setosa
5.4,3.9,1.7,0.4,setosa
4.6,3.4,1.4,0.3,setosa
5,3.4,1.5,0.2,setosa
4.4,2.9,1.4,0.2,setosa
4.9,3.1,1.5,0.1,setosa
5.4,3.7,1.5,0.2,setosa
4.8,3.4,1.6,0.2,setosa
4.8,3,1.4,0.1,setosa
4.3,3,1.1,0.1,setosa
5.8,4,1.2,0.2,setosa
5.7,4.4,1.5,0.4,setosa
5.4,3.9,1.3,0.4,setosa
5.1,3.5,1.4,0.3,setosa
5.7,3.8,1.7,0.3,setosa
5.1,3.8,1.5,0.3,setosa
5.4,3.4,1.7,0.2,setosa
```

Of course, viewing `.csv` files this way is cumbersome and inefficent and makes it near impossible to edit or manipulate the data. Another way to open (and efficiently edit) `.csv` files is using MS Excel. It asks you how the columns are separated from each other when you first try to open a `.csv` file, and appropriately import the data given that you supply correct parameters to it. But since we will use `R` to analyze the data, it makes sense that we open and edit the data using `R` before analyzing or visualizing. there is a simple `read.csv()` function that allows us to do that. Let's inspect how its used and important parameters of the function.

```r
read.csv(file="filename.csv",     ## name of your .csv file
         header=TRUE,             ## whether the first line includes column names
         row.names = 1,           ## Nth column which actually is rownames
         stringsAsFactors = TRUE ## Auto-convert character columns into factors
         )
```

In order for us to be able to use the function with only the name of the `.csv` file, that file should be in your *working directory*, which is the default location (folder) on your computer that `R` looks into when you do anything that depends on a file. If a file is in your working directory you can access it through `R` functions by just providing the name of that file. The default working directory changes for different operating systems and sometimes even two windows computer might have different default `R` working directories. To check your current working directory, you can type `getwd()` in your `R` console. For windows computers, it is usually your `Documents` or `Belgelerim` folders, and for macOS its usually your Home folder (`/Users/yourUsername`).

Remember that, using this function allows us to internalize the data in the `.csv` file as a `data.frame` but without assigning the output of this function to an object, all the function does is internalize the data and print it to the `R` console. So if you read any data using this function, you have to assign the data to an object.

`read.csv()` function has an exporting counterpart, `write.csv()`, which does the exact opposite, and writes an already existing `data.frame` in your environment as a `.csv` file to your computer. This is handy when you have to share data with your colleagues, since a `.csv` file can be easily opened with `R` or MS Excel, if that's your colleague's choice of software. Let's see how its used in below box.

```
write.csv(x = myData, ## object name of your data.frame
          file = "outputFileName.csv", ## name of the .csv file you want to create
          row.names = TRUE ## whether the rownames of x should be included in .csv
         )
```

Example 1

- Download the the `possum.csv` data from ODTUClass
- Import the data using `read.csv()` and assign to an object called `my_data`
- Re-import the data by explicitly stating `row.names = 1` and `stringsAsFactors = TRUE` when using the `read.csv()` function and observe the differences between the two versions (assign the result of this re-importing to another object, `my_data_rownames`)

`.RData` files

`.RData` is a special and binary (not human-readable) data format that is special for `R` programming language. It is also compressed so the file size significantly decreases when you try to save a very large `data.frame` as an `.RData` file as opposed to a `.csv`. But the main advantage for our use case in this lab of using `.RData` files is its ability to hold more than one `R` object. We can save multiple `R` objects, or our entire environment to a single `.RData` file and share all the data directly with each other. Import and export functions when using this data type are `save()` and `load()`.

```
load(file="myData.RData")
save(obj1, file = "myData.RData")
save(list = c("obj1", "obj2", "objN"), ## list of object names to be saved
                                       ## as a character vector
     file = "myData.RData")
```

Since this file type is only meant to be read by `R` and able to hold more than one object, you don't have to (and shouldn't) try to assign the output value of `load()` function to an object. `load()` function retrieves all the information stored in `.RData` file and appropriately recreates the object(s) as they were saved in their original names.

Example 2

- Create an object called `myName` and type your name as its value
- Create an object called `myAge` and type your age as its value
- Create an object called `myHomeTown` and type your home town as its value
- Use the `data.frame` we imported from `.csv` in Example 1 together with the above 3 objects to save them as a single `.RData` file using `save()` function with appropriate arguments to the function.
- Wipe your whole environment using `rm(list = ls())`
- Recreate you environment using the `load()` function, specifying the name of the `.RData` file you just created.

Manipulating imported data

There are many possible reasons why we would want to manipulate the data we have:

- There might be incorrect naming/structure in data, such as column names might we wrong, or ordering of observations might be not correct.
- We migth want to divide the data into smaller, workable pieces to use in different functions, or just want to retrieve a portion of the data that we are actually interested in.

Retrieving different portions of a data is called "subsetting". You can remember the concept of "set/küme" and "subset/altküme" from your math classes in primary school. We will first start by subsetting.

Subsetting data

Subsetting can be applicable on any data type that has more than one element. With subsetting you can get individual element(s) from a vector or row(s)/column(s) from a `matrix/data.frame`. There are several attributes of data you can use to specify the element(s) you want to retrieve by subsetting:

- Name (names for a vector, and colnames/rownames for a matrix/dataframe)
- Index (the order in which an element appears in a data structre)
- Logical vector of same length as the data to be subsetted.

Subsetting with names

```
aVector["elementName"]           ## Get the element that named "elementName"
aDataFrame["rowName", ]          ## Get the whole row that named "rowName"
aDataFrame[, "colName"]          ## Get the whole column that named "colName"

aDataFrame["rowName", "colName"] ## Get the cell in the intersection of row
                                 ## "rowName" and column "colName"
```

Above is a summary of how can we use names when we want to subset our data. Methods shown for `data.frame` is also applicable for `matrix`. You can also use character vectors for subsetting with more than one name for any of the scenarios depicted above. An example for subsetting a vector would be `aVector[c("elA", "elB", "elC")]`.

Subsetting with indices.

```
aVector[5]                       ## Get the 5th element
aVector[-c(2,3)]                 ## Get all elements except 2nd and 3rd
aDataFrame[c(2,4,6), ]           ## Get the 2nd, 4th and 6th rows
aDataFrame[, c(2,4,6)]           ## Get the 2nd, 4th and 6th columns

aDataFrame[c(2,4,6), c(1,2,3)]   ## Get the 3x3 dataframe consisting of
                                 ## the intersection of 2nd, 4th, 6th
                                 ## rows and 1st, 2nd and 3rd column
```

Again, the examples shown for `data.frame` is also applicable for `matrix`.

Subsetting with TRUE/FALSE vectors Please note that you can type the short versions of `TRUE/FALSE` as `T/F` in `R`. Its totally acceptable.

For this descripton, imagine all the objects have dimensions of length 5.(e.g. vector with 5 elements, `data.frame` with 5 rows and 5 columns, etc.)

```
aVector[c(T,T,F,F,F)]            ## Get the first 2 elements, not last 3
aDataFrame[c(T,F,T,F,T), ]       ## Get the rows number 1,3,5
aDataFrame[, c(F,F,F,T,T)]       ## Get the columns number 4 and 5
```

Again, the examples shown for `data.frame` is also applicable for `matrix`. For using logical vectors for subsetting, **the vector you use should have the same length** as the dimension you are trying to apply the subsetting to.

For example, if you want to selectively retrieve the **columns** of a 5x10 (5 rows **10 columns**) `data.frame`, you should have a logical vector (consisting of `TRUE/FALSE` elements) **with length 10**. This is because when you want to use logical vectors for subsetting you are effectively saying that **you will supply an opinion on each and every element about whether you want that particular element to appear in the subset result or not.**

Mixed uses of subsetting approaches. For all three approaches above, you can mix and match them when you are subsetting a `data.frame` (as long as one approach used in rows, other in columns) . A statement like “I want the column named ‘Column A’ and rows numbered {1, 2, 3, . . . 10}” is therefore valid. We can use the notation `N:J` to create a numeric vector of {N, N+1, N+2, . . . , J} in a short and effective way.

Example 3

Download the `GuppyFatherSon.csv` data from ODTUClass.

- Read the data in using the appropriate function and save the object as `guppy` to your environment.
- Inspect the first few lines of the data using `head()`
- Using indices and names, retrieve the first 10 observations for the `sonAttractiveness` variable in the data.
- Create a logical vector using conditional statements to get where `fatherOrnamentation` is greater than 0.5, and store it in an object called `highly_ornamented`. How many `TRUE` values are there?
- Create a logical vector using conditional statements to get where `sonAttractiveness` is less than 0.5, and store it in an object calle `less_attractive`. How many `TRUE` values are there?
- Retrieve the actual values that corresponds to above two conditional statements using subsetting with those vectors. You may save them into their respective objects.
- What proportion of the observations (father/son pairs) are both highly ornamented (>0.5) and less attractive (<0.5)? (Hint: Compare the columns separately with 0.5, then using the appropriate logical operator retrieve cases with both True)

Manipulating/Editing data

There are multiple ways how you would manipulate or edit the data you have at hand. The most common ones are related to the column and rownames of the data and programmatically change/substitute the values in the data that satisfy a particular condition (Either mathematically or according to a character comparison.) We will start with substituing some values.

Substituting values using subsetting approaches. We can actually replace some values in the data using subsetting syntax. The most common way of doing this is replacing some mathematically unwanted values with a single replacement value (e.g. changing all negative values to zero, or changing missing values with zeros). Below is an example using subsetting with indices. We will see this with a real example after next section.

```r
aVector[c(1,2,5)] = 0   ## Replacing values of 1st,2nd and 5th elements with 0
```

Editing names (Review) and some other attributes. Now that we have learned subsetting, we can use it to edit specific elements of any vector. You have already seen how to set names for a vector or col/rownames for a matrix or data.frame. But in those examples lengths of the names vectors should have the same length as the original data. Using subsetting syntax, you can either add names only for the specific elements, or you can change individual elements in an already set names vector.

Another way in which we might to edit the data is to change the class of one particular column (variable) in a dataset. Sometimes R reads the dataset in the way that it sees the most convenient. For instance, the default behavior for the columns that has repeating character values is that reading them as factors. But in some cases this is simply not convenient, or in some cases it tries to read the integer columns as factors just because they contain repeated values. Having columns with class factor when you will use them as numeric and apply arithmetic operations on is simply inconvenient. In cases like this, we may want to convert between classes of columns, like converting a character column to numeric, or to a factor, or something else.

Class conversion (Type casting)

```r
newNumVec  = as.numeric(oldCharVec)               ## character to numeric
newCharVec = as.character(oldNumVec)              ## numeric to character
newFactVec = as.factor(oldCharVec)                ## character to factor
newNumVec  = as.numeric(as.character(oldFacVec))  ## factor to numeric
```

Example 4

We will use the possum data for this example you have already read the data into your environment using read.csv() in the first example.

- Inspect the first few lines of the data using head()
- Some column names are very cryptic and we want to change them to more understantable words. We want to change the totlngth to totalLength. Change that particular column name to the new desired name
- We have learned that observations from the site 3 and 4 are unreliable in the sense that that they were done by a new inexperienced scientist and in particular, sexes of the samples are mislabeled. We want to change all samples' sex value to m for those who come from sites 3 and 4.
- Assume that assesing the gender of possums that are not older than the age of 1 is not possible. However, we have sex values for 10 samples that are of age 1, hence below the suffient age after which we can have a reliable judgement about their sex. Change the sex values of those particular possums to u ,for unknown. (There might be errors regarding the class of the column we want to manipulate, we will handle them if necessary)
- Now that you have edited the data according to your needs, export the data using write.csv() into a new .csv file called possum_edited.csv

Helpful functions There are some other functions that are not directly related to subsetting or data manipulation but they are helpful along the way (like when creating names). We will re-demonstrate such functions when they are to be used in an example, but it is always interesting to learn!

paste() : This function is useful if you want to "paste" two or more character elements in a vector together programmatically. Let's say you want to create a vector like c("Student 1", "Student 2", "Student 3", . . . , "Student N") typing all these elements one by one is not really efficient. This is a very simple example where we may want to use the paste() function.

```r
paste(someVec,  ## Vector (or single char) to be the 1st half of the result
      otherVec, ## Vector we want to constitute the 2nd half of the result
      sep=" "   ## How two halves should be separated
)
```

Implementation of the "Student" example above:

```r
paste("Student", 1:10, sep=" ")
```

substr() : This function allows us to extract some parts of every character element in a vector. For example you can get c("dent 1", "dent 2", . . . , "dent N") from the student vector we created with paste() above by using this function.

```
substr(someVec,          ## Vector to get substrings from
       start = someNum,  ## nth element to start getting substring
       stop  = otherNum, ## jth element to stop getting substring
)
```

```
## First create the vector
studentVec = paste("Student", 1:10, sep=" ")

## Get the substrings
dentVec    = substr(studentVec, 4, 9 )
```

Exercises

load the provided w2_exercise.RData file into your environment to access the necessary datasets for the following two exercise questions.

1. In the dataset hospital you are given a mortality rate dataset for different hospitals or medical centers from United States. Data is comprised of 5 columns. In the Score column, there are mortality rates for the measurement categories that are denoted in Measure.ID and Measure.Name columns. First one is the abbreviated version of the latter one. In the Compared.to.National column each measurement is categorized into groups better (better than the national mortality rate, lower scores), worse (worse than the national mortality rate, higher scores) and noDiff (Not different when compared to national mortality rate)

a. Inspect the data with head()

b. Subset the data to retrieve the observations that belong to the MORT_30_HF group for the Measure.ID variable. This group corresponds to the observations for the mortality for heart failure patients. Save the result to the object heartFailure

c. Further subset the heartFailure data to retrieve the observations that belong to the better group for Compared.to.National variable. Save the result to the object better_heart_fail

d. Further subset the heartFailure data to retrieve the observations that belong to the worse group for Compared.to.National variable. Save the result to the object worse_heart_fail

e. Subset the heartFailure data to retrieve the observations that belong to the noDiff group for Compared.to.National variable. Save the result to the object noDiff_heart_fail

f. You know that if a mortality rate observation is better than the national average, it should have a low score, and vice versa. Observe if this is true by inspecting the Score columns of the three subsets you have created above. To do this use the mean() function over the Score columns of the objects better_heart_fail, worse_heart_fail and noDiff_heart_fail.

If you correctly created the 3 subsets, you should see a relationship like $better < noDiff < worse$ overall.

2. You will use the genome_annotation object for this example.

In the genome_annotation object, you are given genomic coordinates coordinates of genes, exons and transcripts from the human genome. Columns related to genomic coordinates hold information

- entry: gene, transcript or exon
- chromosome: Which chromosome that specific entry is located
- start and end: Start and end positions (Nth nucleotide on the chromosome) of that specific entry
- strand: which of the two DNA strands this entry is located
- gene_id: Unique identifier for the gene that entry belongs to
- transcript id: Unique identifier for the transcript belongs to

Valid values for gene_id and transcript_id look like "ENSG……." and "ENST…… ", respectively.

a. Since one gene might have more than one transcript, and each transcript contains introns and exons, transcript_id column should only be valid for exons and transcripts (not valid for gene entries)

Given this requirement, subset the dataset to retrieve transcript_id column where the entry column is gene and overwrite those transcript_id values by NA (See lab logs of this week for a similar operation on the my_grades vector)

b. Compare the average gene length of chromosomes 19 and 22. To do this, it might be easier to first create additional length column to the dataframe. Remember that we can access columns of a dataframe with the dataFrame$column notation (in addition to the dataFrame[,"column"] notation), and if we do addition or subtraction operations on equally size vectors, corresponding elements will be used pairwise in the operation. You can use the code chunk below to add a length column to your data.

```
genome_annotation$length = genome_annotation$end - genome_annotation$start + 1
```

After obtaining the length information:

- Subset the data to get the length column of entries that have 19 in the chromosome column **AND** have gene in the entry column. Save the resulting vector to an object called chr19_gene_lengths.
- Do the same as above but for chromosome 21, instead of chromosome 19, and save the resulting vector to an object called chr21_gene_lengths.
- Which chromosome have a higher average gene length?

Chapter 3: Displaying and Describing Data

Descriptive Statistics

Descriptive statistics shows the summary of the sample in terms of a quantitative measurement. It is different than inferential statistics (inductive statistics) in that descriptive statistics aims to summarize the sample while inferentical statistics uses the data to learn more about the population that the sample is thought to derived from.

For numerical samples, it is essential to know the **location** and the **spread** of the data. Location can be measured using mean, median or mode of the data while spread of the data is generally measured using standard deviation. For the categorical samples, the proportion is the most important descriptive statistics which shows the fraction of observations in a category. These summary statistics are very easy to calculate using their corresponding functions.

Mean, Median

Arithmetic **mean**, the most common metric to describe the location of a distribution, is defined as the sum of all measurements divided by the total number of observations. Population mean is designated by letter μ and the sample $\bar{Y}$ but the calculation is the same for both.

$$\bar{Y} = \frac{\sum_{i=1}^{n} Y_i}{n} \tag{1}$$

In R, `mean()` function calculates the mean of values given in a vector object. If the object contains missing values, function will print **NA** indicating there are missing values in the data. If we want to ignore the missing values and calculate the mean with the rest, we can specify this with the correct argument `na.rm = T`.

Median is another commonly used metric to describe the location of a distribution. Median is defined as the value on the middle of the data. When all the values are ordered, the value in the middle is the median. It is in fact the 0.5 quantile of the data. Mean and median is the same when the distribution is symmetric.

```
mean(x =,  # calculate arithmetic mean
     na.rm = F,) # false by default
median(x = , # calculate median
       na.rm = F,) # false by default
```

Example

- Calculate the mean current peak for pig data
- Calculate the mean body weight for male and female cats
- Calculate the mean heart weight for male and female cats

Standard Deviation, Variation

Standard deviation (sd) is used as a measure of spread of a distribution. The larger the standard deviation, the further the observations away from the mean. If standard deviation is small, most observations are closer to the mean. Sd is calculated from the **variance**, which is also a measure of spread, by simply taking square root of it. Calculating standard deviation for a population is slightly different than for a sample. Sample standard deviation is designated as `s` while population standard deviation is σ. There is no special symbol for variance, it is simply the s^2 or σ^2.

$$s = \sqrt{\frac{\sum (Y_i - \bar{Y})^2}{n-1}} \text{ , sample sd} \tag{2}$$

$$\sigma = \sqrt{\frac{\sum (Y_i - \bar{Y})^2}{n}} \text{ , population sd} \tag{3}$$

R always calculates sample standard deviation/variation with `sd()`/`var()` functions. If we have access to the whole population (which is not the case for most of the time), we cannot use `sd()` function. We can calculate sd by writing appropriate code.

```
# calculate sample standard deviation
sd(x =,
     na.rm = F) # false by default
var(x =, # calculates sample variance
    na.rm = F) # false by default
```

Coefficient of Variation

Coefficient of variation (cv) is defined as the standard deviation divided by the mean. In other words, standard deviation is scaled by the mean. cv is very useful when comparing standard deviations of two variables that have large difference in their means. There is no specific function in R to calculate cv, so we need to calculate it by ourselves.

```
# calculate coefficient of variation:
sd(mydat) / mean(mydat)
```

We can in fact retrieve some descriptive statistics with another function called `"summary()"`. It calculates mean, median, minimum, maximum, 1st and 3rd quartiles. Summary can take either a vector or a data frame as its argument. If a column of the data frame consists of a categorical variable, it will count each category.

```
# calculate summary statistics
summary(x = )
```

Example

- Calculate the sd of current peak for pig data
- Calculate the sd of body weight for male and female cats
- Calculate the sd of heart weight for male and female cats
- Calculate the coefficient of variation for body weight for male and female cats
- Compare their sd and cv. What can you comment on them?

Displaying Data

Displaying data is an essential process in data analysis to reveal patterns in the form of graphs. Therefore, before doing any statistical analysis, first step is to plot and visually inspect the data. Throughout the course, we will mostly deal with numerical and categorial data. Depending on the data type, we can display frequencies, differences between variables, association between variables etc.

Histogram

A numerical data can be visualized in histogram graphs. We use histograms for only one numerical variable. Histogram can be plotted by first dividing the data into equal intervals (called bins) and counting how many observations there are in each interval. Height of the bars in histogram specifies the number of observations in those intervals.

We will use a basic function "`hist()`" to plot the histograms that will calculate the above steps automatically. First argument of the "`hist()`" function should be a numerical object to be plotted. Plotting functions have so many arguments, most of which are the same for all of them, such as arguments specifying color, title, names etc. There is another important argument of "`hist()`" function called "`breaks = `" that defines the intervals. If we do not specify the number of breaks, the function will automatically calculate the number of intervals that would be best suited for the data.

```
# plot the histogram of the data specifying arguments;
hist(x = , # data to be plotted
     breaks = , # number of breaks/intervals
     main = , # title of the plot
     col = , # color of the plot
     xlim = , # range of x axis
     ylim = , # range of y axis
     xlab = , # label of x axis
     ylab = , # label of y axis)
```

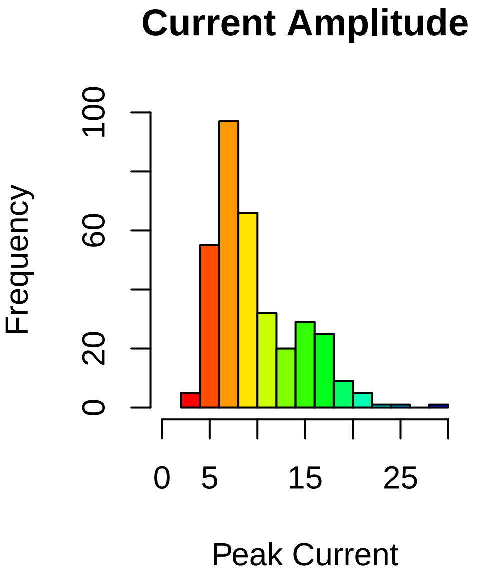
Number of breaks (intervals) is an important parameter for histogram. We can try different break numbers and choose one. To compare two plots, we can draw them side by side by changing graphical parameters. "`par()`" function controls almost all graphical parameters but we will only use "`mfrow = c('#ofrows','#ofcols')`" that divides the plot window into different parts. After drawing the plot, do not forget to change back to default parameters.

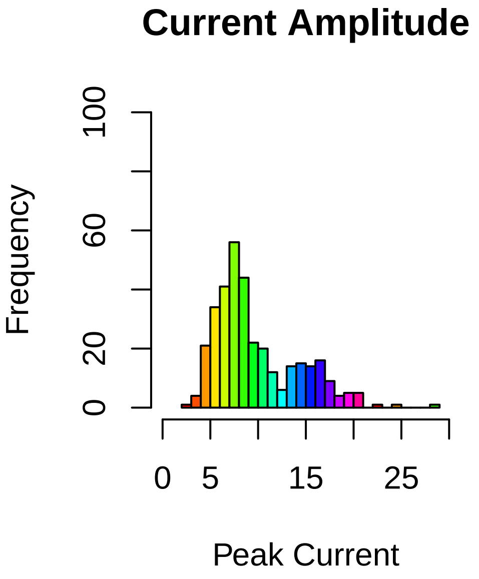
```
par(mfrow = c(1,2)) # 1 row, 2 column; two plot side by side
par(mfrow = c(1,3 )) # 1r, 3c; three plots side by side
par(mfrow = c(2,2)) # 2r,2c; four plots
par(mfrow = c(1,1)) # default, one plot in one window
```

There are various additional functions that can add texts, lines, legends etc. to the existing plot. Some of the useful functions are "`abline()`" that adds straight lines; "`legend()`" that adds legend; "`points()`" that adds points in given positions. Keep in mind that there has to be an existing plot in the plotting area before using these functions.

```
plot(1:10, pch = 19, col = "darkred") # plot some graph
abline(a = , # the intercept in y axis
       b = ,) # slope of the line
abline(h = ) # draw a horizontal line intercepting y axis at 'h'
abline(v = ) # draw a vertical line intercepting x axis at 'v'
```

Below graphs are plotted using the same data (‘paulsen.csv’) but with different bin numbers.[5]





Example Brains adult guinea pigs were sectioned and researchers measured the peak spontaneous current amplitude to decide whether current flows were multimodel which indicates that peak currents are combination of many smaller currents[6].

- Download the data ‘paulsen.csv’ and copy it to your working directory
- Read the file and store it in an object called ‘pig’
- How many columns does the data have?
- Show the first few lines of the data on the console
- How many guinea pigs they used in the study?
- Check the class of ‘pig’ object and class of its variable/s?
- Draw the histogram of measured observations; specify title, color, x axis label, y axis label, x axis range, y axis range and the number of intervals
- Calculate the mean and median of the data, and add them as lines on to the plot.
- What can you conclude about the data looking at the histogram? Comment on the data whether it is unimodel/bimodel, symmetric/skewed etc?

[5] For color argument, rainbow() function was used

[6] reference: Paulsen, O. and Heggelund, P. (1994) The quantal size at retinogeniculate synapses determined from spontaneous and evoked EPSCs in guinea-pig thalamic slices. Journal of Physiology, 480, 505–511.

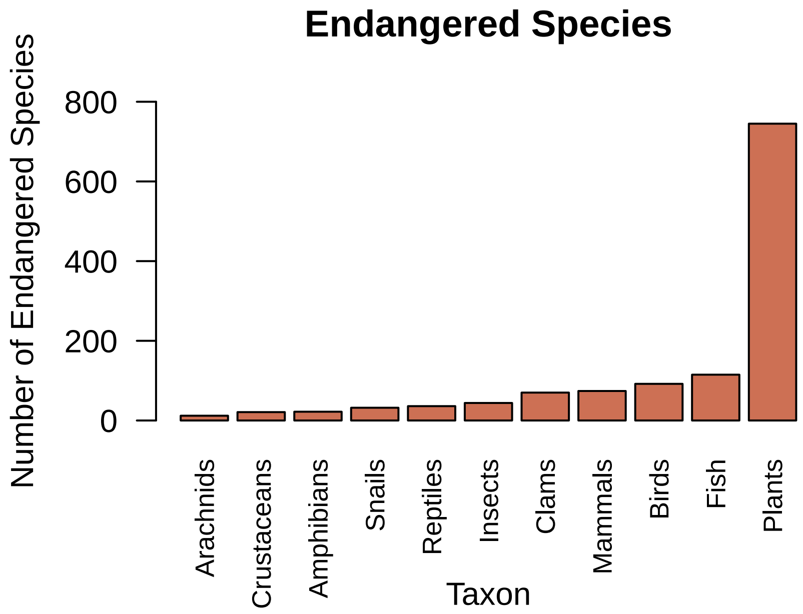
Barplot

Barplot is similar to histogram in the sense that it shows the frequencies. However, we use barplot to visualize categorical data while we use histogram for numerical data. Barplot can be plotted by counting the observations for each category and then drawing bars for each category whose heights showing the counts.

Function to draw barplot is "`barplot()`" whose arguments are similar to those of "`hist()`" function but with a slight differences. For example, we can print the names for each category using "`names.arg =` " argument.

```
# plot the barplot of the data;
barplot(height = , # numerical object describing bar heights
        names.arg = , #  vector describing category names
        horiz = , # logical; if false plot is drawn vertically
        las = , # numeric; draw labels horiz. or vertical
        sub = , # Sub text on x axis
        col =, main = , xlab = ,ylab = ,xlim = , ylim =)
```

Below plot is generated using "`endangeredsp.csv`" data. Categories are ordered with the increasing number of observations.



Example In a study, researchers estimated the number of endangered species for each taxon. Data is given in 'endangeredsp.csv' file.

- Download the data 'endangeredsp.csv' and copy it to your working directory
- Read the file and store it in an object called 'endsp'
- Show how many taxons there are
- Draw the barplot of the data; specify title, color, category labels, x axis label, y axis label, y axis range
- Try to produce the above plot by sorting the categories from lowest to highest count (you need to use sort and order functions)

Boxplot

Boxplot is used to display numerical data. While histogram shows the frequency distribution in terms of divided intervals, boxplot makes use of descriptive statistics[7] of the data. Moreover, when we have more than one numerical data, we can draw boxplot of both variables side by side to compare their distribution.

Boxplot uses a summary of the data, hence it is more compact than a histogram. This property makes it more suitable when we want to compare distribution of a numerical (continous or pseudocontinous) variable from different categories.

We will use `boxplot()` function to draw boxplots along with its arguments, many of which are again the same as histogram and barplot. There are two ways of syntax[8] to produce the same boxplot. First one is the **formula notation** in which we provide two vectors to the `boxplot()` function separated with the "~" (tilde) sign. On the left hand side of the tilde sign should be the vector of values to be plotted while on the right hand side should be the vector categorizing the data. Since we will mostly deal with data frames, numerical and categorizing values will be given in different columns of a data frame.

```
# num.column: numeric values to be plotted
# categorizing.column: grouping variable
boxplot(num.column ~ categorizing.column,
        names = , # names of categories
        col = , xlab = , ylab = ) # etc.
```

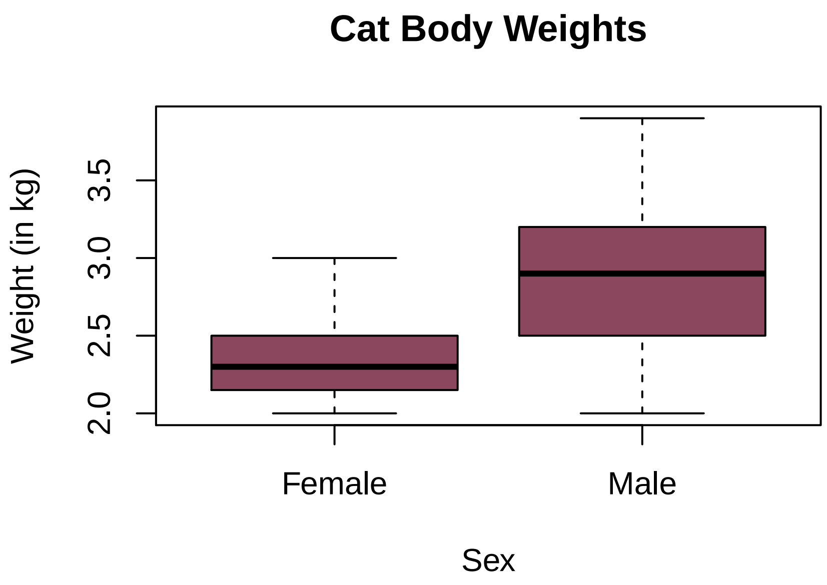
Other way of drawing the boxplot is when we have two numerical variables in two different vectors. Those vectors can be on different columns of a data frame or stored in different vector objects. In this case, we do not need a categorizing variable to group the numerical data because they are already given in two separate vectors. We can draw boxplot of such variables by listing them inside `boxplot()` function using comma to separate the variables.

```
# num.vec1: numeric variable 1
# num.vec2: numeric variable 2
# num.vec3: numeric variable 3 ...
boxplot(num.vec1, num.vec2, num.vec3,
        col = , xlab = , ylab = ) # etc.
```

Below plot is generated using `"cats.csv"` data. There are two categories; female and male which were plotted as two boxplots in one frame.

[7] 1st quartile, median, 3rd quartile and a range estimate along with the outlier values

[8] set of rules that accomplish a task in a programming language



Example In a study of domestic cats, researchers measured the body weight and heart weight of adult cats. Data is given in 'cats.csv' file

- Download the data 'cats.csv' and copy it to your working directory
- Read the file and store it in an object
- Check how many columns and rows there are
- Show how many cats are male/female
- To draw boxplot, let's say we are interested in the weights of female and male cats. We will ignore height measurements fow now. So, draw boxplot of weights for both groups (male and females)
- Draw boxplot with heights this time.
- Specify title, color, x axis label, y axis label, category labels

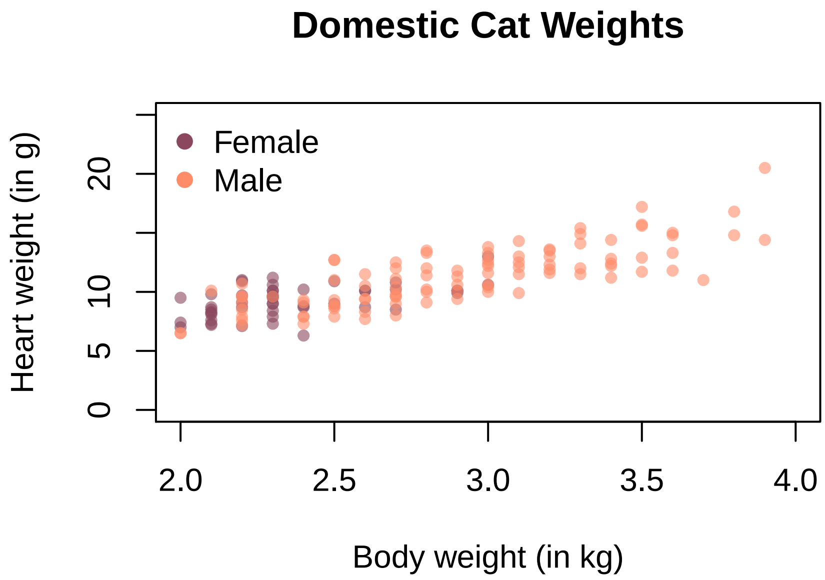
Scatterplot

Scatterplot is used to show the association between two numeric variables. The plot is produced by placing the values of the one variable on x-coordinate (by convention: explanatory variable) and the values of the other variable on the y-coordinate (response variable). Thus, each observation is represented by a point on scatterplot. Observing the pattern of the points on the plot, we can draw conclusion about two variables whether their association is poisitive or negative.

To draw scatterplot, we will use "`plot()`" function. Again, most of the arguments of this function is the same as the arguments of previous plotting functions. First argument of "`plot()`" function is the varible that will go into the x axis, second argument is the variable that will go into the y axis. In addition, we can specify the shape and size of the points in "`pch =`" and "`cex =`" arguments, respectively.

```
# scatterplot
plot(x = , # x axis values
     y = , # y axis values
     pch = , # numeric, shape of the points
     cex = , # numeric, size of the points
     col = , xlab = , ylab = ) # etc.
```

Below plot is generated using height measurement of "cats.csv" data. Legend is added using "legend()" function. Check the help page of the "legend()" function to understand how the legend is added.



Example To draw scatterplot we will again use 'cats' data. Remember that the data contains body and heart measurements of adult domestic cats. Let's say we are interested in whether there is an association between body and heart weights

- Draw the scatterplot of body weight (on x axis) vs heart weight (on y axis)
- Specify title, color, x axis label, y axis label, x axis range, y axis range, choose a type of point

Exporting Plots

There are several ways to save a plot in different formats. Most commonly used formats are png and/or pdf for which we have "png()" and "pdf()" functions. These functions open a connection to a specified file. First argument of the function specifies the path and the file name, other arguments are about height, width of the plots etc. but we will not deal with them. After writing all the plotting codes, one has to use "dev.off()" function to close the connection. Otherwise, R will write all the plots to that specified file.

```
pdf(file = "myplot.pdf",) # opens connection to myplot.pdf file
par(mfrow = c(2,2)) # divide the plotting area into four
hist() # draw some plots
plot()
boxplot()
hist()
dev.off() # close the connection

png(file = "myplot2.pdf",) # opens connection to myplot.pdf file
```

```
plot() # draw the plot on first page of the file
plot() # draw the plot on the second page etc.
dev.off() # close the connection
```

Example Save all your previous plots into a file

- For the pig data, divide the window into four parts and draw four histograms with different number of breaks and save it a pdf file ‘pighist.pdf’
- Save endangered species barplot and scatterplot of cats data to a single file on different pages

Exercises

1. 'desertbird' data contains the informatin about bird species living in desert. Each row is a bird species and the corresponding column is the number of individuals found in the desert for that species.

- Is the data numeric or categorical?
- How many bird species are there in the data?
- Which type of plot would you use to visualize the data?
- Draw the appropriate plot of the data and specify the following arguments:
- x axis label, y axis label, title, color, labels, y axis range

2. 'spideramputation' data contains speeds of male spiders, before and after their one arm is amputated. In a sipder species, male spiders has been observed to voruntarily amputate one of their organs just before sexual maturity. The relevant question is that does males do this to increase their speed performance (because their weight will decrease) to find a female partner. To test the hypothesis, researchers recorded the speed of male spiders before and after one of their legs amputated.

- How many spiders are used in the study for both groups?
- Draw an appropriate plot for the data and specify the following arguments:
- x axis label, y axis label, title, color, labels
- By just looking at the graph, which group seems to have higher speed?
- Calculate the mean and sd for both groups
- We have to do a statistical analysis before jumping to any conclusion but by observing the plot and the the mean and sd values, what can you comment on the hypothesis?

3. For this exercise, we will use one of the data that is inside R but not visible to us unless we call for that data. Even if do not see the 'cars' object in your environment, type 'cars' on your console to retrieve it. Data consist of the speed of cars and the distances taken to stop.

- Draw the appropriate plot for the 'cars' data with the arguments specified for previous questions.
- Comment on the plot. Are distance and speed are positively/negatively associated?

4. From R datasets again, we will use 'DNase' dataset. Type 'DNase' in your console to retrieve the data. It shows the data about an ELISA assay for the protein DNase in rat serum. The experiment is repeated 11 times. First column shows the experiment number. Second column shows the known concentration of the protein. Third column shows the measured optical density in the assay.

- Check the class of each column
- Retrieve concentration and density values for 'Run 3' and save it to 'DNase.r3' object (you can give any name you want to the object). Also retrieve same values for 'Run 5' and save it to 'DNase.r5' object. Repeat this process for 'Run 8' and 'Run 11'
- Divide plotting window into four parts (2 rows, 2 columns)
- Draw the conc. vs density plot for both the runs that you subsetted above in a single window. Specify the arguments that would make the plot high quality
- Do the runs seem to be consistent with each other?
- We could draw all the runs, not just the above 4 runs, in a single window area using only one ploting function but it is out of scope of this course. If you are curious, you can find the function in the help page of the DNase dataset.

Chapter 4: Probability Distributions

This week, we start with learning to use probability distribution functions to calculate probabilities of some events. There are many distributions that model specific cases of random events. We will get to know 3 important distributions and how to use them.

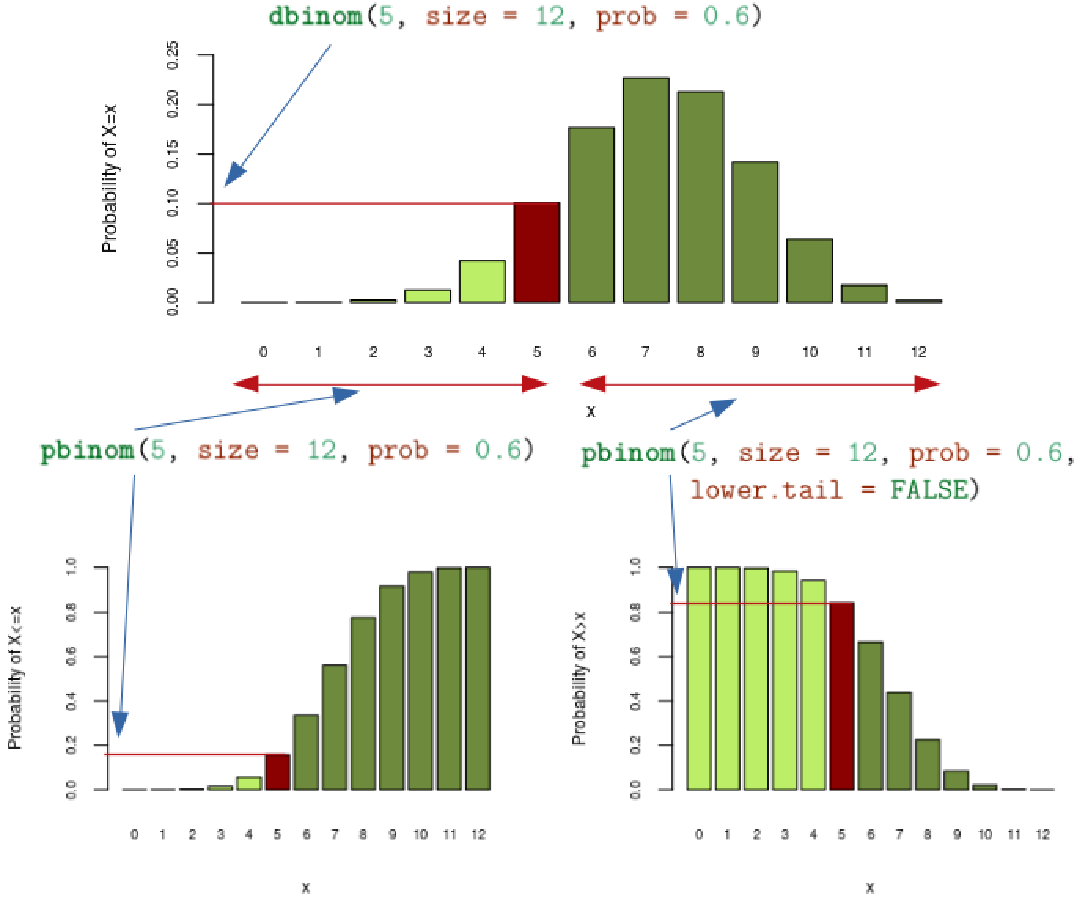
Then, we proceed with sampling distributions. They are probability distributions themselves, giving probability of not a single element, but a sample of elements drawn from a population. As we will be dealing with samples in any kind of statistical analysis, sampling distributions play a key role in the whole field of statistical inference.

Calculating probabilities

Binomial distribution

A random variable representing the number of elements belonging to a category, when a certain number of elements are drawn independently from a population, which consists of elements belonging to either one of two possible categories, is distributed according to binomial distribution. One of the categories is arbitrarily called "**success**", and each independent draw is called a "**trial**". **Probability of success** is the proportion of elements in the population, that belong to the "success" category. The binomial distribution gives probabilities of each possible outcome (number of successes) for a given number of trials and for a specific probability of success.

Probability mass function `dbinom()` gives probability of observing an outcome being equal to `x` and **cumulative distribution function** `pbinom()` gives sum of probabilities of $X \leq x$ or $X > x$ for a given `x`.



```
dbinom(5, size = 12, prob = 0.6)
```

```
pbinom(5, size = 12, prob = 0.6)
```

```
pbinom(5, size = 12, prob = 0.6,
    lower.tail = FALSE)
```

We can also calculate probabilities of multiple outcomes at once. Instead of an input number, we can write a vector of numbers and can get corresponding probability values.

```
dbinom(0:12, size = 12, prob = 0.6)
```

```
##  [1] 1.677722e-05 3.019899e-04 2.491417e-03 1.245708e-02 4.204265e-02
##  [6] 1.009024e-01 1.765791e-01 2.270303e-01 2.128409e-01 1.418940e-01
## [11] 6.385228e-02 1.741426e-02 2.176782e-03
```

Example

In a family with 5 children, what is the probability of having

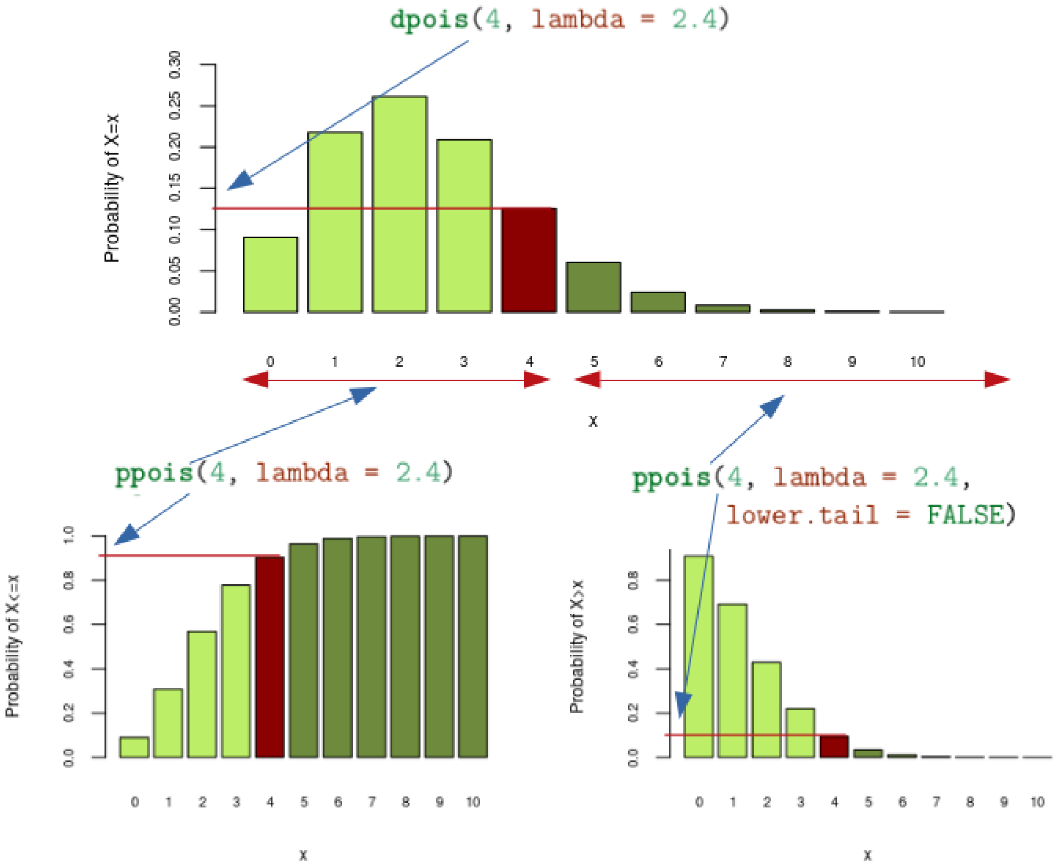
- exactly 2 boys?
- at least 2 boys?
- more than 2 boys?
- less than 2 boys?
- at most 2 boys?

(Assume that probability of having a boy is 0.5)

Poisson distribution

A random variable representing number of events occuring independently in a fixed time or space interval, with a constant **rate** of occurence is distributed according to Poisson distribution.

Probability mass function `dpois()` gives probability of observing an outcome being equal to `x` and **cumulative distribution function** `ppois()` gives sum of probabilities of $X \leq x$ or $X > x$ for a given `x`.



Similar to the previous section, we can write a vector of numbers and can get corresponding probability values.

```
dpois(0:10, lambda = 2.4)
```

```
##  [1] 0.0907179533 0.2177230879 0.2612677055 0.2090141644 0.1254084986
##  [6] 0.0601960793 0.0240784317 0.0082554623 0.0024766387 0.0006604370
## [11] 0.0001585049
```

Example

In a cell culture, on average 1.3 cell divisions occur per minute. You observe the culture for 5 minutes. What is the probability of observing

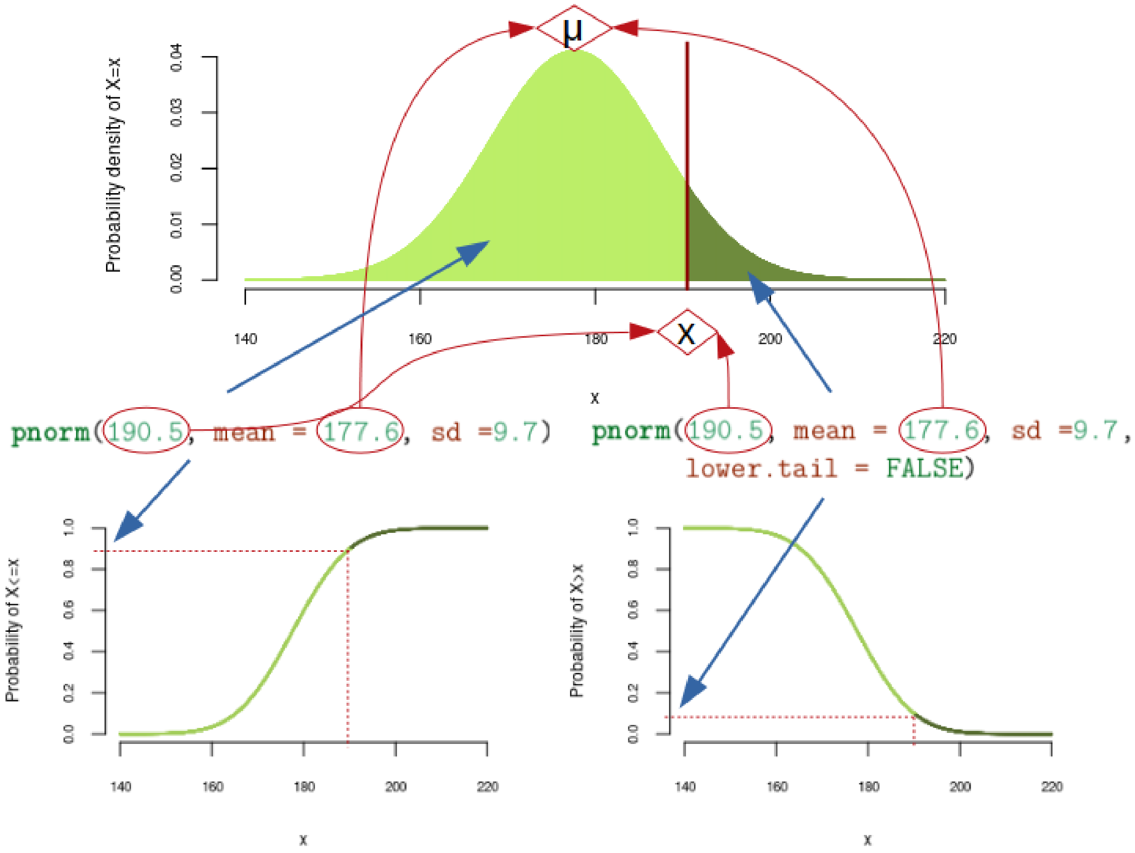
- exactly 4 divisions?
- at least 4 divisions?
- more than 4 divisions?
- less than 4 divisions?
- at most 4 divisions?

Normal distribution

Normal distribution is a continous probability distribution in contrast to the previous two distributions, which are discrete. Many quantities in nature follow a normal distribution and we will see in next section why this is the case and why this distribution is very important.

Each specific normal distibution is characterised by its **mean** and **standard deviation**.

When our random variable is a continous one, calculating probability of a single outcome is meaningless / not defined. We will talk always talk about probability of observing an outcome for a range of values, or in other words, probability of an outcome being between two values. **Cumulative distribution function** `pnorm()` gives probability of observing an outcome $X \leq x$ or $X > x$ for a given value of `x`.

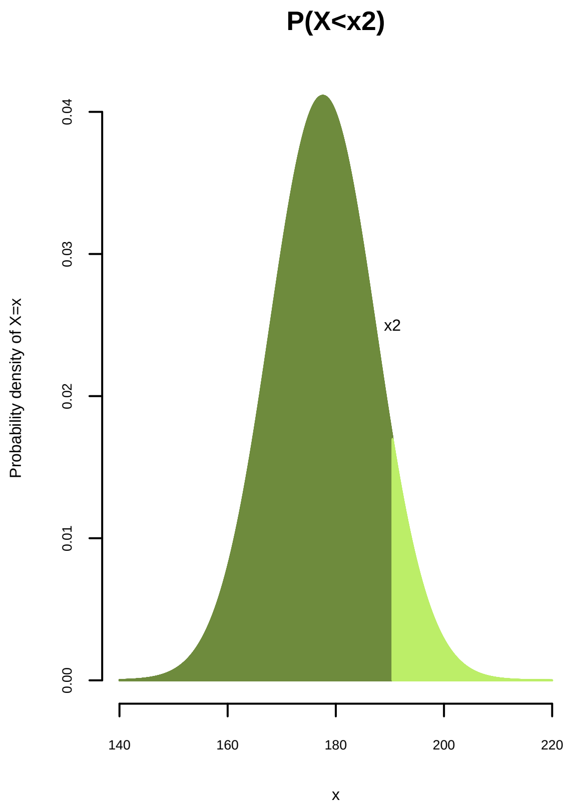


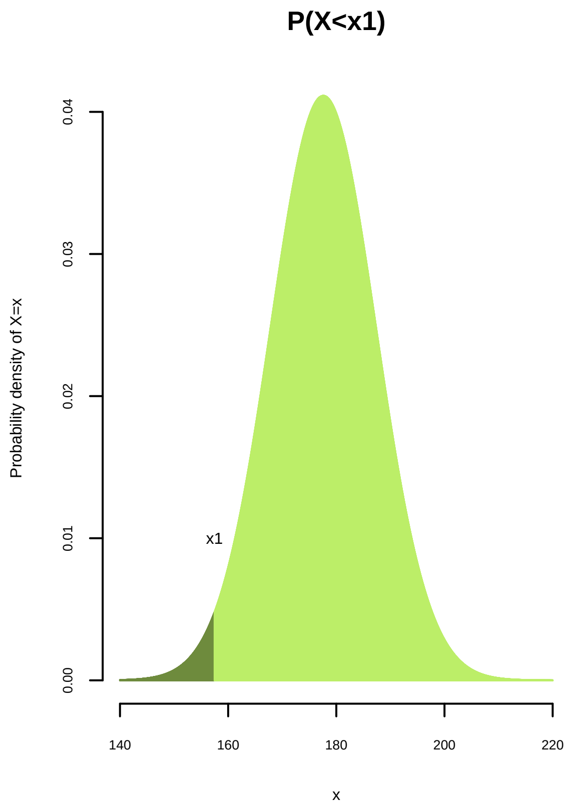
```
pnorm(190.5, mean = 177.6, sd =9.7)
```

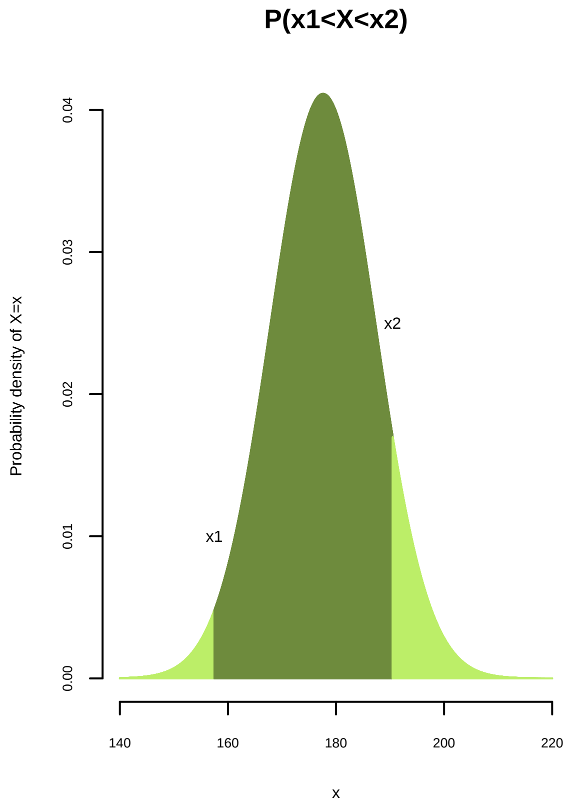
```
pnorm(190.5, mean = 177.6, sd =9.7,
      lower.tail = FALSE)
```

If we need the probability for $x_1 < X < x_2$, we can substract the cumulative probability up to the lower bound $P(X < x_1)$ from the cumulative probability up to the upper bound $P(X < x_2)$.

```
pnorm(x2, mean = 177.6, sd =9.7) - pnorm(x1, mean = 177.6, sd =9.7)
```


P(X<x2)
Probability density of X=x
0.00
0.01
0.02
0.03
0.04
x2
140
160
180
200
220
x


P(X<x1)
Probability density of X=x
0.00
0.01
0.02
0.03
0.04
x1
140
160
180
200
220
x


P(x1<X<x2)
Probability density of X=x
0.00
0.01
0.02
0.03
0.04
x1
x2
140
160
180
200
220
x

Example

Male astronauts at NASA should be between 157.5 and 190.5 cm tall. Height of men in US is normally distributed with a mean = 177.6 and standard deviation = 9.7. What is the probability for a random US male to be,

- too short for NASA?
- too tall for NASA?
- eligible for NASA?
- not eligible for NASA?

Empirical distributions

When we have data from a whole population, we can calculate probabilities of outcomes by simply dividing number of elements that satisfy the condition for the outcome to the number of all elements in the population. In cases when we have partial data from the population, we may use it as an approximate probability distribution.

Example

- Import the file "KenyaFinches.csv"
- Check distribution of the `beaklengths`
- What is the probability of observing a finch with beaklength - Less than 8 cm? - Greater than 11 cm? - Between 8 and 10 cm?

---

for loop

Looping is used for repeated execution of a set of commands. R has several built-in loop commands and the easiest one is the `for` loop. `For` loop iterates through a vector of integers, each time running the set of commands written inside of it. Below is the syntax structure of a `for` loop:

```
for(i in 1:n){
  # do something to be repeated n times
  # i can be used within these codes to
  # do the repeated operation on different values
}
```

---

if/else statement

When we want to execute a piece of code depending on a condition, we can use `if(){}else{}` statement. The example below includes some random operations to show how the if/else statement works.

```
# the cpndition expression can be anything that produces a TRUE/FALSE value
if(conditionExpression){
  # do something if condition is true
  } else{
```

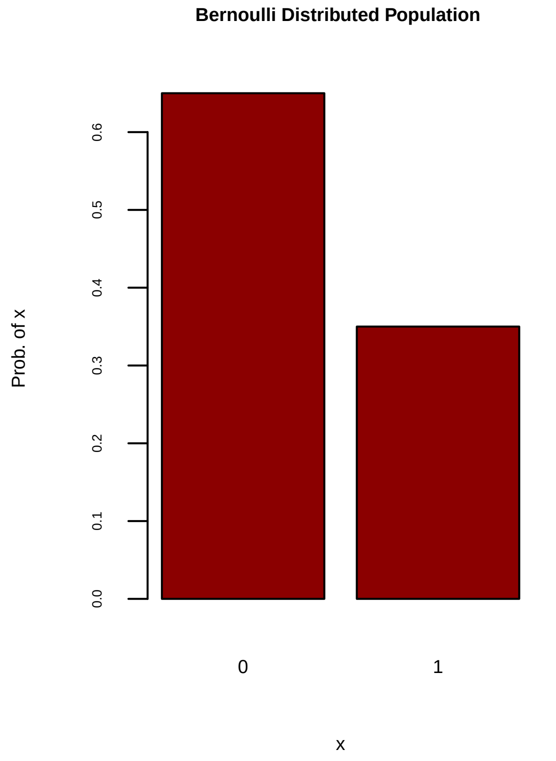
```
  # do another thing if condition is false
}
```

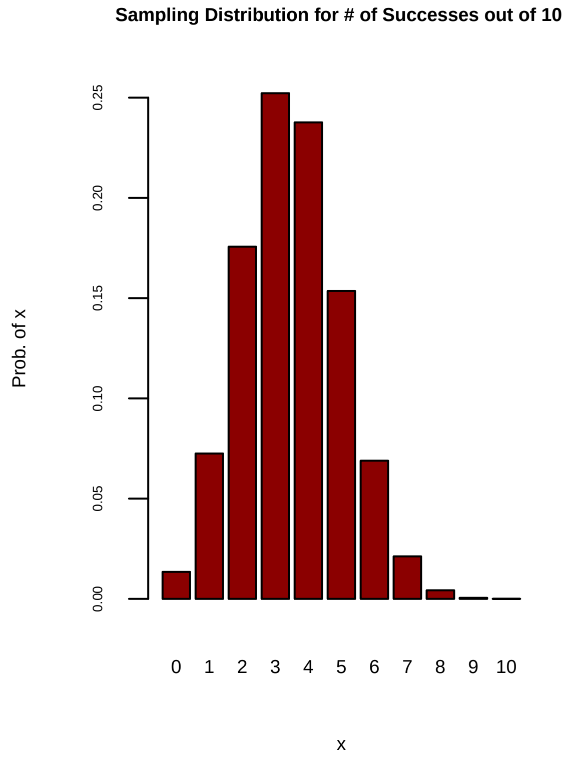
If we want to execute a piece of code when given condition is true, but we don't want to do anything if the condition is false, we can use `if(){}` statement by without the `else{}`.

Sampling distributions

When we take a sample from a population and calculate a characteristic from that sample, probabilities corresponding to possible outcomes of that sample characteristic is given in a sampling distribution.

Perhaps the simplest example of a sampling distribution is the binomial distribution. It is the probability distribution for number of successes in samples taken from a population with Bernoulli distribution. A Bernoulli distributed population consists of elements that belong to one of two possible categories.



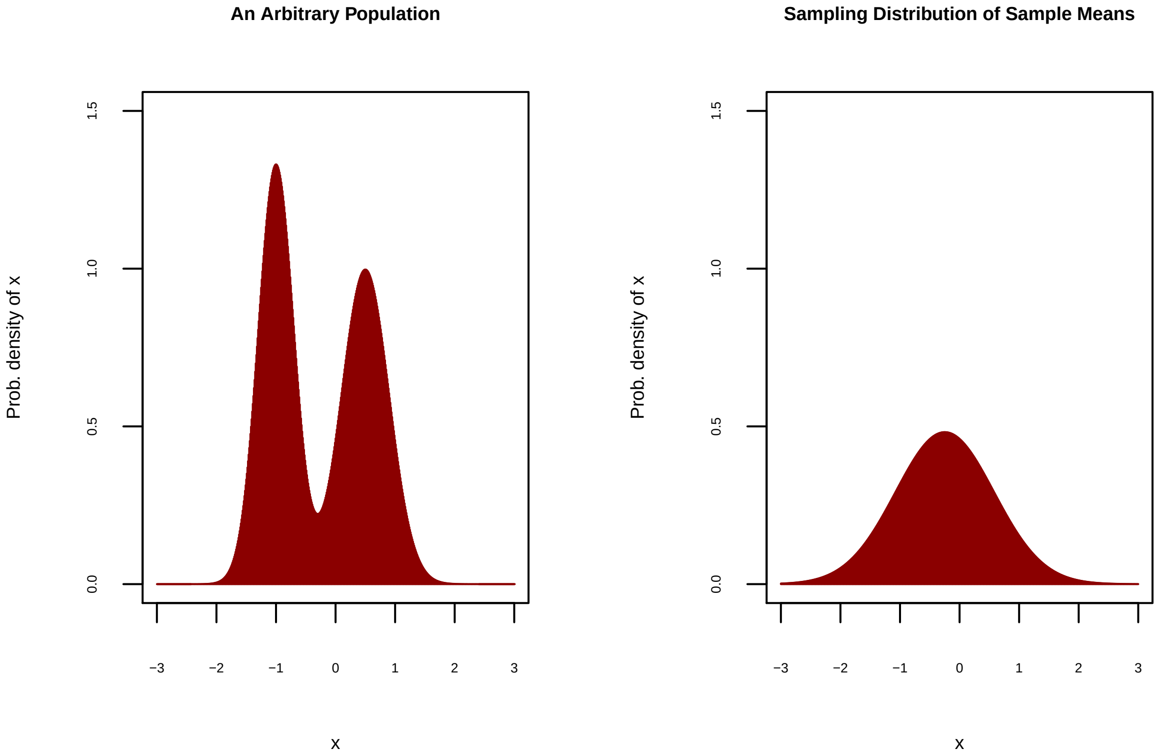


```
# some functions for generating random samples

sample(x=vector_to_sample-_from, size = sample_size,
       replace = FALSE, prob = NULL)
      # if we need sampling with replacement set repalce = TRUE
      # if drawing each element has different probabilities,
      # give a vector of probabilities

rbinom(n = sample_size, size = number_of_trials, prob = prob_of_success)
rpois(n = sample_size, lambda = average)
rnorm(n = sample_size, mean = mean_of_sampled_pop, sd = sd_of_pop)
```

Central limit theorem

A particularly useful sampling distribution is the distribution of sample means. Frequently we want to make inference about population mean using a sample mean. We can calculate the probability of getting the sample mean we have if a hypothesized population mean is true, by using the distribution of sample means.

Central limit theorem states that, irrespective of the population's distribution, means of samples taken from that population converge to a normal distribution with the same mean as the original population and with $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$, when the sample size n gets larger. $\sigma_{\bar{x}}$ is standard deviation of the distribution of sample means and has a special name **standard error of mean**.



We will not attempt to prove the central limit theorem, but demonstrate it using a simulation. From some arbitrary populations, we can take many samples and calculate their mean values. Using a large number of samples we can approximate a probability distribution of sample means.

Example

- Use the `beaklength` of `KenyaFinches` data from previous example. Recall its distribution.
- Take a sample of size 20 from that population and record its mean.
- Repeat the above step for 1000 times, record each mean as a new element of a vector.
- Plot the distribution of sample means.
- In a 2*2 plotting area, draw 4 distributions of sample means with sample sizes 10, 20, 50, 100.
- For each of the 4 vectors that store sample means, calculate their means and standard deviations. Compare them with the mean and standard deviation of the original population.

Approximate probability calculation for a sample mean

Central limit theorem states that distribution of sample means is approximately a normal distribution with

$$\text{mean} = \mu_{\bar{x}} = \mu$$
$$\text{sd} = \sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

When we know these 2 population parameters μ and σ, we can calculate probability of drawing a sample with a specified mean.

Example

Calculate the probabilities below in 2 different ways. 1st using the sampling distributions we generated in previous example, 2nd using the normal approximation. What is the probability of catching,

- 5 random finches, with mean beaklength less than 8 cm?
- 20 random finches, with mean beaklength less than 8 cm?
- 30 random finches, with mean beaklength between 8-9 cm?
- 10 random finches, with mean beaklength greater than 9 cm?

If you see a sample of 40 finches with a mean beaklength of 11.2 cm, would you suspect that the mean beaklength of the population is actually not 8.74?

Exercises

1. Obesity rate

- A group of researchers want to survey obesity rates and report that 461 out of randomly selected 2428 people are obese. What is the probability of obtaining this result, if in reality 15% of the population is obese (if in reality each random person has 0.15 probability of being obese)?
- A more meaningful question would be to ask for the probability of observing 461 *or more* obese people out of 2428. Calculate that probability, with the same probability of success as above (0.15).

2. Seedlings On a field experiment, abundance of seedlings on a large number of rectangular plots with same size and same properties are observed. On average there was 1.32 seedlings per plot. What is the probablity that a randomly selected plot has,

- no seedlings
- exactly 1 seedling
- more than 1 seedling
- 3 or more seedlings
- less then 2 seedlings
- 2 or less seedlings

3. Fish From a fish population with normally distributed ($\mu = 82$, $\sigma = 6.7$) weight values (in grams),

- what is the probability that you catch a fish heavier than 85 grams?

You catch 25 random fish from the same population, what is the probability that,

- all of them are heavier than 85?
- their mean weight is greater than 85?

4. Birth weight Human birth weights are normally distributed with $\mu = 3339$ grams and $\sigma = 612$.

What is the probability that **a** random newborn baby weighs,

- less than 3000 grams?
- between 3500 and 4500 grams?

What is the probability that randomly selected 40 newborn babies have a mean weight,

- less then 3000 grams?
- between 3500 and 4500 grams?
- more than 500 grams less, or more than 500 grams greater than mean weight of baby population?

5. Rain What is the probability of

- 180 rainy days a year (360 or 365 days, your choice), if 50% of the days are rainy
- more than 180 rainy days a year, if 80% of the days are rainy
- getting wet less than 10 days a summer (you decide how long summer lasts), if 13% of the summer days are rainy, and whenever it rains you toss a coin to decide to take an umbrella or not. (computationaly produce a probability estimate)

6. Beans From a population of yellow and green beans with a 7:3 ratio, you want to sample 20 beans, and count number of yellow ones. Generate a sampling distribution of 500 such samples. Compute how many of these samples have less then 10 yellow beans.

7. Birth weight factors Let us simulate a hypothetical situation where birth weights are determined by the sum of 5 genetical and 2 environmental factors.

Each genetical factor is the presence/absence of an independent allele and presence of each of the alleles add 700 gram to baby's weight and absence contributes 0. Presence probabilities of the 5 independent alleles are given as 0.2, 0.3, 0.5, 0.6 and 0.9.

One of the environmental factors is uniformly distributed and ranges between 200 and 900 grams of addition to baby's weight. The second environmental factor is distributed according to Poisson distribution and its mean is adding 500 grams.

Simulate the situation by following the given steps:

- Randomly select an element from the vector c(700,0) with the probabilities c(0.2,0.8). Store it in a variable called g1.
- Randomly select an element from the vector c(700,0) with the probabilities c(0.3,0.7). Store it in a variable called g2.
- Randomly select an element from the vector c(700,0) with the probabilities c(0.5,0.5). Store it in a variable called g3.
- Randomly select an element from the vector c(700,0) with the probabilities c(0.6,0.4). Store it in a variable called g4.
- Randomly select an element from the vector c(700,0) with the probabilities c(0.9,0.1). Store it in a variable called g5.
- Randomly select a number between 200 and 900, so that probability of drawing each value from this interval is equal to each other. Store it in a variable called e1. (Use the runif() function)
- Randomly select a number from Poisson distribution with the mean 500. Store it in a variable called e2. (Use rpois() function)
- Sum the variables g1,g2,g3,g4,g5,e1 and e2. This is a simulated birth weight for a single baby.
- Write a for loop to repeat everything above 1000 times. Store each sum as an element of a vector. This vector will be our sampling distribution.
- Draw a histogram of the generated sampling distribution.

Chapter 5: Estimating with Uncertainty - Hypothesis Testing

Estimation

Estimation is the process in which we infer a population parameter from samples drawn from it. Up to this week, you have learned to estimate some population parameters (like μ - the population mean) using samples or sampling distributions. This week we will learn to estimate such population parameters with the added uncertainty, a declared measure of how we are confident about our estimation. Former approach is called **point estimation** whereas the latter one is called **interval estimation**. Then, we will go on to form -and test- some hypotheses about the parent populations so that we can ask and answer biologically significant questions about our biological data that we may have at hand in the future.

In real world, we can collect only small samples or how adequate they may seem in size, they usually add up to really small portion of the original data. This is just how things work. But that's okay as long as you have a way to express the uncertainty of the estimate made from your collected samples. This is where the **confidence intervals** come into the play.

Confidence Intervals

Confidence intervals will be named after the degree at which you are confident the actual population parameter lies in the range you report. An x% confidence interval is a numerical range in which you are x% confident about your estimation regarding the population parameter. The most widely used one is **95% interval** for the **population mean**, and we are mostly going to utilize this one.

Remember the concept **standard error of the mean** from last week's lab sessions. Of course you are not going to be able to access the whole population's data in real-world examples, last week's animation example was just a simulation to show you the relationship between the population and the sampling distribution which is created by drawing repeated samples from an arbitrary population.

Now, last week it is shown to you that where the population standard deviation is σ, the size of the samples that you are drawing from this population is N, **standard eror of mean** of your sample approximates to

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}}$$

according to the **central limit theorem**. Since you will most likely not be able to know what σ is, you can use the sample's standard deviation (s) to get an approximation for it.

Once you have the sample mean and $\sigma_{\bar{x}}$ in your hands, it is trivial to calculate a 95% confidence interval for the population mean. There is something called the **2SE rule** . It states that if you go $2\sigma_{\bar{x}}$ from your sample mean in both directions, you got your 95% confidence interval. Which means that the interval is defined as the numerical range

$$[\bar{x} - 2\sigma_{\bar{x}}, \bar{x} + 2\sigma_{\bar{x}}]$$

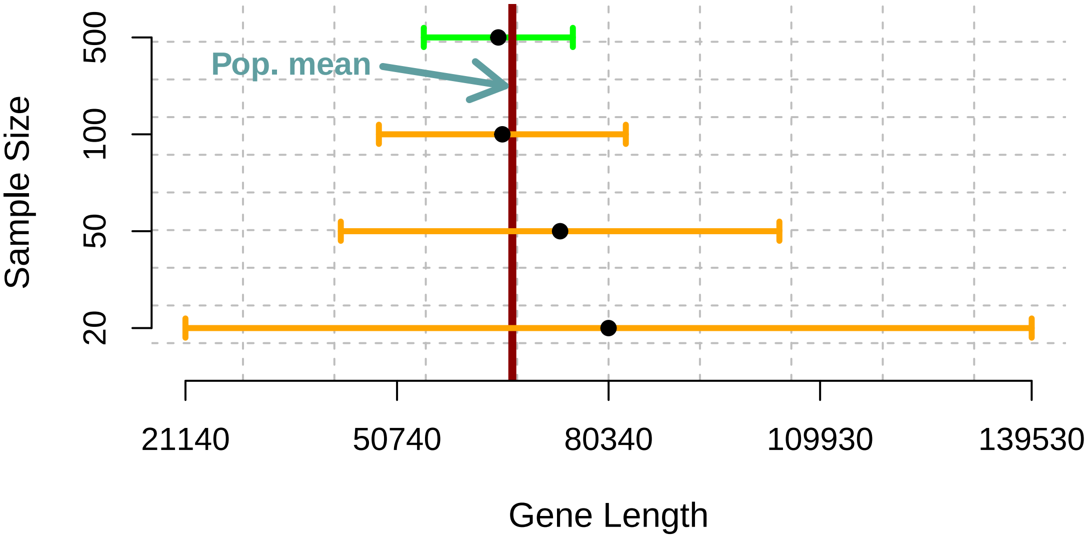
When you inspect the formula for SE, you can see that in the denominator, there is the sample size which should tell you that as you increase your sample size, your **SE** will get smaller and therefore your confidence interval should get **narrower**, which in turn will leave you with a more **precise** estimation of the μ.

Example 1: Human gene lengths

In the dataset `humangenes` there is length data for all the protein coding genes in human genome curated from an annotation file that is couple of years old. So there might be slight changes if you were to be look up actual gene lengths online. Since it includes potentially all genes, this dataset will be our population.

1. Calculate the population mean and save it to an object called `popmean`
2. Draw a sample from this data with sample size 10 assign in to an object called `s10`
3. Calculate the 95% confidence interval for population mean from this sample and save to the object `s10_ci`
4. Draw another sample with size of 100 and save it to an object called `s20`
5. Calculate the 95% confidence interval for population mean from this sample and save to the object `s20_ci`
6. Repeat the steps 2-3 for a sample of sizes $50, 100, 500$.
7. Compare resuling CIs with the actual population mean. Do they include the population mean? How are their width compared to each other?

Below,there's a visual example of the results from such an exercise described above.



Hypothesis testing

Stating the hypotheses

In a hypothesis test, there should be two **exclusive** hypotheses about data. The null hypothesis (H_0) and the alternative hypothesis (H_A). The null hypothesis is almost always refers to the situation of no change ($i.e$ no difference, no preference, no relation, no correlation) and is always the one that is being tested.

In other words, you test the null hypothesis using the observation from your sample, and report the probability of that observation if the null hypothesis is true.

The null distribution

In order to perform a hypothesis test, we need a null distribution that is the probability distribution -of the sample statistics (i.e. mean) in which the **null hypothesis is true**. Null distribution can be obtained as a means of creating a sampling distribution -assuming you have access the whole population or the data is really simple that you can simulate it- or can be selected from a collection of pre-defined mathematical distributions (binomial, poisson, normal *etc.*) taking into account of the nature of your data.

As you have learned last week, `R` has some pre-made functions for calculating probabilities with such distributions. And you will see in following weeks, it also has some pre-made hypothesis test functions so you don't need to worry about performing an hypothesis test in the way that we have learned this week (A little bit too long, but very cautious and informative). But, before we go on with more complex examples in the following weeks, we will learn the basics and the reasoning behind the process of hypothesis testing in this week first.

One-sided vs. Two-sided tests

An hypothesis test can be performed in both one-sided and two-sided fashion. In a two-sided test, you are interested in the deviations in **either direction** from the mean, whereas in a one-sided test, you select the direction in which the observations deviate from the mean.

Example 2: Right and left handed toads

Researchers sampled 18 random toads(kind of frog) from the wild and recorded which forelimb each toad used to remove the baloon from their heads that the researchers put on. The question is that do right-handed and left-handed toads occur with equal frequency in the toad population (H_0), or is one type more frequent than the other, as in the human population (H_a)? Out of 18 toads, **14** were recorded to be right-handed and 4 left-handed. Is this observation an evidence of handedness in toads?

Using above example, we will go on with rest of the subject.

The team of researchers observed **14** right-handed toads in their sample. Remember their initial question: they want to test if right handed toads and right handed toads **occur in equal frequency** in nature by taking a sample from a wild population. In this case, if one group were to be more common in the nature, it wouldn't matter whether the unfairness favors the right or the left handedness would it? In other words, having too many right-handed toads would be as bad as having too many left-handed toads in their sample. Therefore, their hypotheses should be like this:

- H_0: $p_{right} = 0.5$ (right handedness is equally likely as the left-handedness)
- H_A: $p_{right} \neq 0.5$, either right or left handedness is favored in the nature.

Under the null hypothesis , you would expect 9 right handed ones out of 18 toads but researchers observed **14**. So the difference between the mean (expected value) and the observation is **5**.

As you can figure out from the hypotheses above, this is a two-sided test. Because any significant deviation from the mean would falsify the null hypothesis regardless of its direction. In a hypothesis test, you should sum up the proportions or the probabilities that represent deviations that are **at least as extreme as your observation**. Since we will perform a two-sided test, we should sum up the probabilities that represent

- $9 + 5$ right handed toads and more
- $9 - 5$ left handed toads and less

because extreme/deviant observation defines the extreme values in either tail of the distribution in our example. Summing up above probabilities will give you the probability of encountering an observation like ours if the H_0 is true. This is also called the **p-value** of the test.

Example 2 (contn'd)

To test the hypotheses:

- $H_0 : p = 0.5$
- $H_A : p \neq 0.5$

Apply the following steps:

1. Using the `sample()` function and and for loops, create a sampling distribution drawn from the vector `c('R', 'L')` with the sample size size of 18 and record the number of "R" elements (right handed toads) in each sample. Use 1000 repetitions in the for loop. This will be our null distribution.
2. Plot the null distribution. Mark our observation (14 right handed toads out of 18 toads) using the `abline()` function.
3. Use the `table()` function to convert your null distribution into a frequency table. You can save the result into an object called `toads_freq`.
4. Sum over the appropriate tails of the null distribution in accordance with the hypotheses.

You calculated a p-value using appropriate tails of the distribution in accordance with your hypotheses, but you still need a rule for making a decision regarding the truth of the null hypothesis based on the p-value. At this point, α value comes to your rescue. It is the threshold value after which you reject that your H_0 is true. In other words, you should reject your null hypothesis if your p-value is **less than** (less probable) the α value. Most common common practice is using the rule $\alpha = 0.05$. This value is also called the **significance level** in the context of hypothesis testing.

Let's see one more example regarding hypothesis tests but using a known probability function instead of the `sample()` function this time.

Like the cumulative probability functions you have seen for binomial, poisson and normal distributions (`pbinom()`, `ppois()` and `pnorm()` respectively), there are also functions that lets you draw a random sample from those probability distributions. they start with `r`(for random) instead of `p`. (`rbinom()`, `rpois()` and `rnorm()`)

Example 3: Catching fish

You want to go fishing and decided to buy a fishing rod from a store. The seller showed you a fishing rod and claimed that an average skilled person can catch 2 fish on average on the seashore in one day using this fishing rod but if you go fishing in the open sea you can catch 5 fish on average in one day (for the sake of the example, let's assume that there is only one place allowed for fishing on seashore and open sea).

The seller seemed to be convincing and you decided to buy that fishing rod. In the following two consecutive days, you went to fishing on seashore and open sea and you caught 3 fish in total. Was the seller telling you the truth (and you just had bad luck) or s/he just lied to you to sell the fishing rod? (You can use the poisson distribution to model the number of fishes you can catch in a given day)

1. State the null and the alternative hypotheses.
2. Create the null distribution using a for loop with 1000 repetitions doing below steps in each repetition:
   - Draw two different poisson samples of size 1 with $\lambda = 2$ (seashore) and $\lambda = 5$ (open sea). Since the probabilites change depending on the place you are fishing, you should use two different poission distributions for the two days.
   - Sum the outcomes of the two possion samplings and save them into a vector outside the for loop. (This will be your simulated total catched fish in two days for each trial)

3. Plot the null distribution and mark your observation (3 fish) on the plot using `abline()` function.
4. Convert the null distribution vector to a frequency table using `table()` function. save the result into an object called `fishing_freq`.
5. Sum the appropriate tail(s) of the null distribution from converted table to get a p-value.

Type I and Type II errors

Even if you did your best to come up with really good hypotheses, collecting truly random samples and be careful not to make any calculation errors, a significant p-value ($p < \alpha$) does not always mean that your null hypothesis is actually false. Just by chance, you may have rejected a true H_0. This type of error is called a **Type I error**. Actually, if you think about what the α stands for this makes sense. $\alpha = 0.05$ means that the chance of you randomly rejecting a true hypothesis is 5%.

Of course, if needed you can decrease your α even more for a more stringent rule of rejecting a null hypothesis. For example you say that I want to be 99% or even 99.5% sure that the H_0 is actually false if I reject it (α values 0.01 and 0.005 respectively).But imagine this scenario, just because you wanted to be sure that the H_0 it is actually false when you reject it, you now face the risk of failing to reject an actually false H_0 (which happened to have a slightly higher p-value than your super-small α).

This second type of error, **failing to reject a false H_0**, is called a **Type II error**. Outcomes of the Type I and Type II errors are called **False positives** and **False negatives**, respectively. Since rejecting a H_0 means your H_A can be true, meaning you got a **positive** result in your test somehow. Like a drug that nearly destroys an aggressive tumor, when compared to the null distribution of all the other drugs which are not so effective. But this positive result is actually not true, it is the result of an error, hence the term **False** positive.

You see, this is just like the two ends of a seesaw. Selecting a proper value for the α is always a tradeoff. If you don't mind occasional false negatives but your positive results just **must** be authentic, then you may want to select a lower value for α than the usual. Or, if you are desperately in need of any positive results in a particular hypothesis test in a way that you are fine with double-checking your positive results auxillary methods after you obtain them, but **you cannot afford to miss an actual positive**, then you can use a higher value for α for his purpose.

Example: Type I errors

In the dataset `type1_pvals` you are given p-values of the repeated hypothesis tests (1000 times) similar to the toad example we have seen earlier using approach like this in each step:

1. Take a random sample of size 18 out of two outcomes(right or left handedness) with each outcome has an equal probablity.
2. Sum the number of the first outcome out of 18 to use it as the test statistic.
3. Perform a hypothesis test using the test statistic and specify that alternative hypothesis is that the 1st outcome has an higher probability to occur. Save the p-value

Now, using this data,

- Choose an appropriate visual way of displaying it.
- Based on the looks of the distribution, are there any significant p-values?
- Calculate the percentage of significant p-values. What percent of the 1000 hypothesis test yielded a result that resulted us to reject a true null hypothesis?

Exercises

Data required for the exercises are stored in the `w5_excData.RData` in ODTUClass.

Cereal sugar content data This dataset contains measurements of sugar content in frosted flakes breakfast cereal. We want to find the mean sugar content of cereal. Data is a sample from population of all flake cereals. The data is given to you in the `cereal` object.

- Calculate the point and interval estimators (95% CI) for population mean.
- If the sample size was smaller, would the range of 95%CI be narrower or wider?

Mirror image of flowers (6.4) A mud plant has a mechanism to avoid self-fertilization that in some individuals female flowers deflects to the left and to the right in others. The male is on the opposite side. To investigate the genetics of this variation, researchers crossed pure strains of left- and right-handed flowers, producing only right-handed plants in the next generation. This F1 generation were then crossed with each other. The expectation under a simple model of inheritance would be that their offsprings should consist of left- and right-handed flowers in a 1:3 ratio. Of 27 offsprings from this cross, 6 were left-handed and 21 were right-handed. Does this result support the simple genetic model?

The null hypothesis is that left:right ratio is 1:3 (proportion of left-handed individuals in the offspring population is p=1/4).
The alternative hypothesis is that this ratio is not 1:3.

- What is the expected number of left-handed flowers out of 27 for this model?
- Is this a two-tailed or one-tailed test?
- Test this hypothesis using sampling distribution.

A new cell culture medium You work in a research lab that uses human cell lines on a regular basis. For the last couple of months you were sloppy with your lab work and as a results of this, you are really behind the schedule with your experiments. This situation makes you really nervous, especially a big collaborators meeting coming up in two weeks. Even if you start to take things seriously today, you still wouldn't be able to finish the work by the due-date. "If only there was a way to grow these cells much faster. Maybe then I could finish my work on time." you hopelessly daydream.

Later, when you were browsing the web you see an advertisement from a shady looking not well known company selling biological lab supplies. The ad is about a new culture medium that is tested extensively regarding its toxicity to the cells and other important factors. It is certified that this medium is not harmful to cells and is at least nutrient as other standard media. On top of that, they claim that using this medium will make you get a considerably high yield of cells in the same amount of time when compared to other standard cell culture media. But there is no or any test results regarding this high-yield effect of the product.

You know that on average you expect to see 10 cell divisions per 100 cells in a given day. The ad claims that this medium can give you up to 1.5 fold increased yield (10 can jump up to 15 if the claim is true).

The claims seem tempting to you and you decide to order a small testing sample. You grow your cells in this medium and count the divisions you observed and found that you observed **12** divisions per 100 cell per day.

You want to perform a hypothesis test to see **if this medium really increases the rate of cell division**.

- What kind of a theoretical distribution you can use to model this kind of data. (Number of cell divisions per day)
- What should be your hypotheses?
- Using the cumulative probability function (e.g. `pnorm()` for normal, `ppois()` for poisson) of your selected distribution, calculate the sum of tail probabilities in accordance with your hypotheses.

- Based on the p-value, can you confidently say that this medium increases the cell division rate? (use $\alpha = 0.05$)

Quest for finding a good apartment You are in the market for renting a new house to live in during your education in METU, but you have limited budget in a way that spending more than TRY1000 for rent would be really problematic for you. You started your apartment hunting from 100.Yil neighborhood (Near campus) but you saw that rents there are way out of your budget. You decide to look for further neighborhoods for an apartment.

You find a nice looking one with a relatively low rent around kugulupark. It would be a really nice home, but you are a little bit concerned about your transportation to the campus everyday. You share this concern with the realtor (Emlakci) and he says you don't need to worry about the transportaion. There is bus stop in 2 minutes walking distance by which buses and mini-buses pass frequently and all stop by guvenpark/kizilay. You know that from kizilay its easy to get to the campus. The realtor claims that on average **5 different buses pass per 10 minutes** by the stop near home so you it's impossible for you to have problems with your transportation.

Being an enthusiastic student of this course, you decide to test this claim using a very basic hypothesis approach. After meeting with the realtor and telling him you will think about this offer, you go and wait at the bus stop he mentioned for 10 minutes and record how many buses go by. You only saw 2 buses passing by that stop in 10 minutes.

- What kind of a pre-defined probability distribution you think you can use for this type of data?
- What are your hypotheses
- Do you think that the realtor is telling you the truth?

Titanic data The data 'titanic' contains the information on the survival status, sex, age, and passenger class of 1309 passengers in the Titanic disaster. Several questions can be asked and different hypotheses can be tested on the data. 'passengerClass' column has informatin about whether the passanger is the 1st, 2nd or 3rd class. Also, 'survived' column contains information abot whether the passenger survived the accident or not. ref: https://vincentarelbundock.github.io/Rdatasets/doc/carData/TitanicSurvival.html

- Calculate how many first class passengers there were in Titanic.
- Calculate how many of the first class passengers survived.
- Calculate how many of the first class passengers did not survive.
- Using your observation (calculated in above steps) test the null hypothesis that probability of survival is 0.5 for the 1st class passengers.
  - State null and alternative hypotheses.
  - Calculate the expected number of survived first class passengers under null hypothesis.
  - Create null distribution (sampling distribution) under H_0. Draw the histogram of the distribution (drawing barplot would be more appropriate but it is not that much important, just keep in mind that this is a discrete distribution.)
  - Calculate the p-value using sampling distribution.
  - What can you conclude from the test result given that significance level is 0.05?
- Do you think the actual probability of survival is higher or smaller than 0.5 for 1st class passengers?
- Repeat all above questions for second class passengers.

- Repeat all above questions for third class passengers.
- Do you think there rich people had higher probability of survival on the titanic accident?
- Is there a better way of comparing probability of survival of three passenger classes? What would be the problem with conducting 3 independent tests? (This is about multiple testing problem, think about the last topic we discussed in the lab regarding Type I and Type II errors.)

Obsessively fair boardgame startup (Visit the course ODTUclass page for the link to the visual aid to this example)

Consider that there is a new boardgame company with an exceptional new game that could change the whole market. Their game includes element of luck-like most games do- in certain parts. To decide in those moments they want to put a game-coin in the box. They want this coin to be completely fair because they afraid that a cheap, poorly made unfair coin may spoil the whole thing and their dream of being millonaires can die on the vine.

So they spend their resources to develop a fancy and really really, fair coin. Luckily they obtain a set of premade gold standard game-coins with the help of an inside-man that works in Hasbro^{TM} and start to test their first batch of coins using hasbro ones as reference. They come up with an approach like this:

- Make 1000 coin-toss trials tossing 100 reference coins in each trial and record the total number of heads to use it as a *null distribution*
- Toss the 100 coins from their first batch also and record the total number of heads. (They only have 100 coins for now.)
- Using the null distribution they test whether their coins are as fair as the reference coins.

You will help them to perform the tests.Using the dataset `gamecoins` perform below tasks:

- Subset the data twice in accordance with the hypotheses stated above and our observation (62 heads). Save those two subset results to objects naming them whatever you like.
- Use `sum()` function over the correct column in both subsets to calculate the total number of counts. As you know there were 100 coins in the process of creating this null distribution. but those 100 coins were tossed **1000** times. So the sum of `count` column should be 1000 in the original dataset. You can double-check this again using `sum()` function.
- Devide both total count results for your subset by the grand total to obtain respective probabilities.
- Finally, sum those two probabilities up. This is your p-value. Remember that this value represents the probability of the H_0 being true. So what can you say about the null hypothesis based on this probability?

Follow-up (if you reject the null hypothesis above) We had to reject the H_0 so it seems like that their coins favoring one of the sides. And they know that their designer friend that helped the company design all the physical parts of the game (coins, pawns, etc.) and the game board. They were very happy of the overall brave design of the parts especially the coin. They are really proud of how the coin looks. Wooden base material and two different abstract mythological animal figures for head and tail sides made with silver metal inlay material.

After getting suspicious results with the coins, they remember that their designer warned them like this:

"Because the figure on the tails side is a little bit more complicated and elaborate, there may be slight over-use of the silver inlay material on that side"

Basically, she told them to be extra cautious about the coin favoring the heads side since the tails side may be heavier.

Having this in mind, they want to test their coins again. This time they want to **check if there is actually a bias towards heads** coming up with their coins. Null distribution and their observation is still the same, they just need a new set of hypotheses.

- What kind of a hypothesis test is should be? (One sided / Two sided)
- What are the null and alternative hypotheses?
- What is the p-value and what is your final decision on the subject based on the resulting p-value? Are the coins actually heavier on the tails side? (Use $\alpha = 0.05$)

Chapter 6: Analysing Proportions

Last week, basics of hypothesis testing and confidence intervals were covered. The fundamental step in hypothesis testing is to generate a null distribution.A null distribution can be constructed either using sampling distribution like we did in previous week or we can use a theoretical distribution that fits to our data. Sampling distribution can be modelled by a particular theoretical distribution under certain assumptions. There are many theoretical distributions that can be used instead of sampling distributions for hypothesis testing. In previous weeks, binomial and poisson distributions and their related functions were introduced to calculate the probabilities for random samples. For hypothesis testing, there are defined functions in R that calculates p-value from a theoretical distribution.

In this section, we will describe how to test hypotheses about a population proportion, in other words categorial data that has two outcomes; traditionally called success and failure. Choosing which outcome to be success or failure does not make any difference in the test result, i.e. does not affect p balue, but it will affect how we interpret the result. Binomial test is required for hypothesis testing of a proportion using binomial distribution. The function used for binomial test in R is `binom.test()`.

```
binom.test(x =, # number of successes
           n =, # number of trials
           p =, # hypothesized probability of success
           alternative =, # alt. hypthesis; indicates one-sided/two.sided
           conf.level = ) # confidence level
```

Example

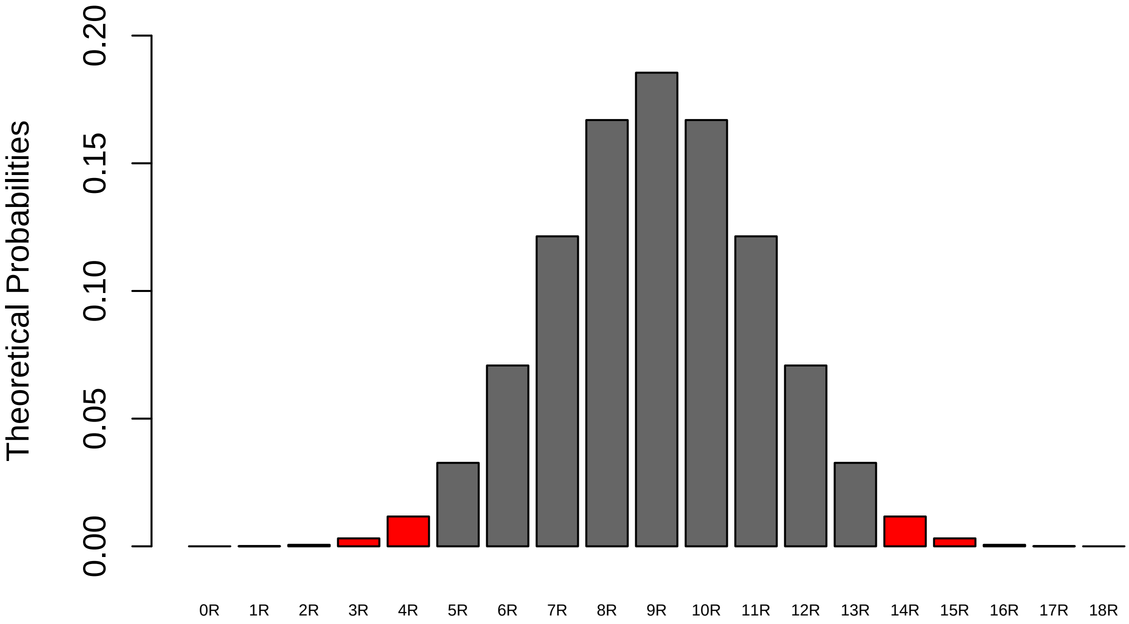
- Consider right and left handed toads example from previous week. There were 18 toads sampled from wild and measured handedness on them. 14 toads were recorded to be right-handed. Based on this observation, is there a handedness in toads?
- Perform a hypothesis test choosing success as right-handed toads.
- Perform the same test using left-handed toads and check if there is any difference in p value and CI.
- Perform a hypothesis test for **right-handedness** in toads.

How binom.test() function calculates p-value?

The function constructs the null distribution using the binomial equation for each possible outcome. Since there are 18 toads in total, there will be 19 outcomes for right-handed toads: 0 right-handed, 1 right-handed, 2 right-handed,… 18 right-handed toads. You can also calculate the probabilities for each outcome using the binomial formula:

$\binom{n}{p} * p^x * (1-p)^{n-x}$, n: sample size, p: probability of success, x: number of success

Recall that "`dbinom()`" function will calculate probability for a given outcome using above formula. We want to calculate probabilities for all outcomes as a vector from 0 to 19. Below barplot is constructed using `dbinom()` function.



P value is calculated by summing the correct probabilities of observations on both sided (colored in red). Alternatively, we can only sum one tail and multiply it with 2 to get the same answer. This is true because binomial distribution is symmetrical. However, we should not use this approach if we were to use a sampling distribution (what we did last week) because random sampling by its nature contain randomness. Sampling distribution will not be perfectly symmetrical.

Confidence intervals for proportions

There are several methods to calculate confidence interval (CI) for proportion, two of which are presented in the course book: Wald method and Agresti-Coull method. In R, we do not need to calculate the confidence interval, "`binom.test()`" function output gives the 95% CI (α being 0.05) by default. If you check the help page of the function, it states that confidence interval is obtained using the method given in Clopper and Pearson (1934) article. However, this method gives sligtly wider CI than the two methods mentioned in the book. If we want to calculate CI with other methods, we need to use the given formulas. To obtain a confidence interval other than 95% CI, we need to specify the value as an argument of the function which is illustrated in the above function ('conf.level=' argument).

In hypothesis testing, we can either use p-value or confidence interval to decide whether we should reject H_o or not. You already have learn that if p-value is smaller than the α, we reject H_0. We can deduce the same conclusion using confidence interval. For example, take α as 0.05, and calculate 95% CI. If CI does not include H_0, then we reject H_0.

Example

- In human genome, there are 781 genes on X chromosome and 19506 genes on other chromosomes. X chromosome covers the 5.2% of the whole genome. This indicates that if genes are distributed evenly through the chromosomes, one should expect 5.2% of the genes to be located on X chromosome.
- Test if observed ratio is in accordance with the expected ratio. Since, we do not indicate any direction, one has to make a two sided test.
- State null and alternative hypotheses
- Use the appropriate hypothesis test function

- Observing the result of the test, do you reject or fail to reject the null hypothesis?
- What is 95% confidence interval?
- What is 99% confidence interval?
- Consider that someone hypotheses X chromosome should contain less number of genes claiming that there are less sex related genes. Using this information, perform a new hypothesis test followed by indicating null and alternative hypotheses.

Exercises

1. You want to cultivate oranges and before making a big investment, you want to test the survival rate of orange seedlings with a small sample. From a seedling producer you buy 20 seedlings. Producer tells you that at least 80% of their products survive to next year. After a year you observe that 14 of your 20 seedlings are alive. Would you say that producer is wrong and less than 80% of their products survive to next year? Make a hypothesis test (alpha=0.05) and state what the following are:

- Population
- Sample and sample size
- Test statistic
- Hypotheses
- P-value
- Conclusion If the producer had not make any claim about survival rate, and we want to get an estimate for survival rate of orange seedlings, based on our sample, we can get a confidence interval for that.
- Calculate the 95% confidence interval.
- If we need a more precise (narrower interval), what can we change?

2. Suppose that you want to afforest an arid area with pine trees. To do that you are going to purchase pine saplings from a company. Company claims that at most 20% of the saplings will dry out. Before afforesting whole area with these saplings, you want to test whether company tells the truth. You purchase 40 saplings and plant them. After several months, you observe that only 2 of the saplings dried out.

- Perform a hypothesis test using your sample to decide whether proportion of dried-out saplings is less than 0.2
- State hypotheses
- According to test result, state your conclusion

3. (Assignment Problem 21) In a test of Murphy's Law, pieces of toast were buttered on one side and then dropped. Murphy's Law predicts that they will land butter side down. Outof 9821 total slices of toast dropped, 6101 landed butter-side down.

- What is a 95% CI for the probability of a piece of toast landing butter-side down?
- Using the result of first part, is it reasonable to believe that there is a 50:50 chance of the toast landing butter-side down or butter-side up?

Chapter 7: Fitting Expected Probabilities to Frequency Data

When we want to test if number of observations belonging to categories are according to some expected proportions, what we would do is called a goodness of fit test. Null hypothesis of goodness of fittest is that observed frequencies fit to expected frequencies (and the difference between observed and expected values are due to chance). Alternative hypothesis is that they do not fit the expected frequencies. Therefore, goodness of fit test is always a one tailed test with alternative being "greater" representing the deviations from a good fit.

Goodness of fit test

The statistic of the test is $\chi^2 = \Sigma_i (Observed_i - Expected_i)^2 / Expected_i$

which measures the discrepancy between observed and expected frequencies. Expected frequencies are calculated from a distribution stated in the null hypothesis. Calculated test statistic is tested against a distribution called χ^2 distribution to calculate the p value. R has `chisq.test()` function that tests observed frequencies against the supplied expected probabilities, always performing a one tailed test. Another function `pchisq()` calculates cumulative probability corresponding to given χ^2 values with given `degrees of freedom`. It can be used to calculate p value for a manually calculated χ^2 value and manually determined degrees of freedom. Before doing a goodness of fit test in R, there are 2 things to consider. First we need to check if any of the expected frequencies is less than 5. If this is the case, categories should be merged so that none of the categories has an expected frequency less than 5. Second consideration is to decide on degrees of freedom. If expected probabilities are retrieved independent of the data, $df = n - 1$, where n is the number of categories. If expected probabilities are calculated using some information from the data, another degree of freedom is lost for each parameter calculated form the data. In that case $df = n - 1 - x$, where n is again number of categories and x is number of parameters calculated from the data.

```r
# an easy chi_squared test which is not flexible
chisq.test(x = vector_of_observed_frequencies,
      p = vector_of_expected_probabilities)

# a flexible way for calculating p value of
# a chi_squared test
pchisq(q = calculated_test_stat,
       df = degrees_of_freedom,
       lower.tail = FALSE)

# a simple way of getting number of observations
# belonging to each category
table(vector_of_categorical_values)
```

Example 1

'days_of_birth' data gives information on day of the week on which babies was born, for 350 babies. Under a proportional model, one would expect that babies should be born at the same frequency on all seven days of the week.

- Calculate the frequency of birth on each week day
- Visualize the distribution
- Under the proportional model, calculate the expected probabilities for the each day of the week. One would expect that the number of births on each day should reflect the number of days in a given year. Assume that frequency of each day in that year is all equal
- Test whether observed frequencies fit to expected probabilities or not
  - State the null and alternative hypotheses
  - Conduct the test
  - Make a decision based on the p value
  - State a statistical and biological conclusion

Example 2

Let us consider a field experiment about abundance/distribution of a fungus species. Researchers divide a piece of land into a 10 by 10 grid, and count how many individual fungi grow on each of the 100 areas. Their question is, are they distributed randomly (according to Poisson distribution) or not. They have obtained the following data:

Number of fungi per area	0	1	2	3	4
Number of areas with that many fungi	32	36	29	1	2

- Calculate the expected probabilities using `dpois()` function for all outcomes
  - Lambda is the mean number of outcomes for all grids:
  - (0x32 + 1x36 + 2x29 + 3x1 + 4x2)/(32 + 36 + 29 + 1 + 2)
- Calculate the expected frequencies for all outcomes
- If an expected count is below 5, combine it with the second rare outcome
- Test whether observed frequencies fit to expected probabilities or not
  - Calculate the df
  - Calculate the chi squared statistic
  - Calculate the p value using `pchisq()` function
  - Make a decision based on the p value
  - State a statistical and biological conclusion

Example 3

Consider the phenotype frequencies of F1 generation from a dihybrid cross and perform a goodness of fit test to Mendelian expected ratios. Observed frequencies of the 4 categories are 53, 17, 22 and 8. Expected ratios are 9:3:3:1

Exercises

1.

A researcher studies prey preferences of a lynx population. Out of the 40 observed prey animals caught by lynx; 19 were mice, 10 birds and 11 hares. Based on this data, can we say that there is a prey preference of lynx, or are the prey caught equal in frequencies?

- Population
- Sample and sample size
- Test statistic
- Hypotheses
- P value
- Conclusion

2.

We want to study habitat preference of a finch species. We release 58 individuals from captivity and record to which of the 5 available habitats they fly. "habitats.RData" file contains data from the experiment. The experiment area consists of:

Habitat	A	B	C	D	E
Coverage	0.42	0.08	0.19	0.03	0.28

We want to see, if they fly to habitats randomly (proportional to the coverage of the habitats) or do they have a preference. Make an appropriate hypothesis test. (You need to merge the 2 categories with lowest frequency.)

3.

Genotype frequencies in a locus is given as follows:

28 AA, 85 Aa, 29 aa

Is this locus in Hardy-Weinberg equilibrium?

Hint:

- First calculate allele frequency of A from data (p(A))
- Then calculate expected frequencies as: p(A)*p(A); 2*p(A)*p(a); p(a)*p(a)
- Test goodness of fit of the observed genotype frequencies to the expected probabilities.

Chapter 8: Contingency Analysis

Associating Two Categorical Variables

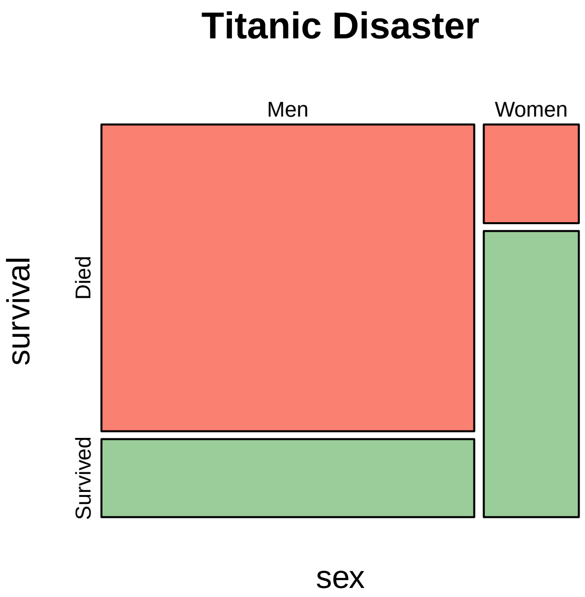
When there are two categorical variables that we want to analyse, we test for associations between those variables. In other words, we want to determine whether one variable is contingent ("bağımlı" in turkish) on the other. We can test this association using contingency analysis. Specificly, contingency analysis is used to investigate the independence of variables. If the two variables are associated, this implies that they are not independent of each other. Contingency test can also be used for more than two categorical variables.

There are two ways of testing association between categorical variables. If there are only two categorical variables and there are only two categories in each variable, we can calculate *odds ratio and confidence interval for it (or we can use Fisher's Exact Test to test the significance of the odds ratio). Other test we can use is the χ^2 contingency test which does not have a restriction on the number of variables and the categories.

Mosaic Plot

One can visualize categorical variables in a different type of plot called **mosaic plot**. Mosaic plot uses the area of rectangle to represent the relative frequency of occurence in each combination of variables.

Below mosaic plot shows the frequencies of each categorical combination of two variables (sex and survival status). The plot represents the Titanic disaster data where survival status of every man and women in the ship recored after the Titanic sank. It can be clearly observed that women had a lower probability of death than men. If there were **no association** between the sex of the individual and his/her survival status, we would expect their survival probability to be the same. therefore, the data indicates that sex and survival were **not independent**. We have to use a statistical test to decide whether this association is statistically significant.



Example 1: Titanic Disaster Data

Download and load the 'week8.RData' into your environment. 'titanic' data contains represents the survival status of every man and woman in the ship.

- Calculate the frequency of each category in each variable.
- Draw the mosaic plot of the data as shown above.

- Add labels to the plot including title, x-axis label, y-axis label, color etc.

Odds Ratio

Odds ratio can be calculated when there are only two categorical variables, each consisting of two categories. In the above example we have a 'sex' variable which has two categories: `men` and `women`. And, the other variable is `survival` which also has two categories: `survived` and `died`. Odds ratio compares the proportion of each categories between two groups.

Odds, Odds Ratio

Odds of success are the probability of success (survival in this example) occuring divided by the probability of failure (death). To calculate the probability of success (which is represented by 'p' below equation) we need to calculate frequencies.

$$O = \frac{p}{1-p}$$

To quantify the difference between the odds of success in two variables, we can use **odds ratio (OR)**. Odds ratio is simply calculated by calculating the ratio of odds of success of the two categories.

$$OR = \frac{O_1}{O_2}$$

If the odds ratio is equal to 1, we could say that odds of success (survival in the above example) is equal for both groups (men and women). In other words, two variables are independent of each other (or they are not associated; no association between sex and survival) OR > 1 means that group 1 (women) have higher odds of (success) survival than group 2 (men). Obviously, the opposite trend (OR < 1) would then mean that group 2 (men) have higher odds of success (survival) than group 1.

One can also calculate odds ratio using an alternative way. For example, if the contingency table is given as follows:

Var2 Var1	**Category A**	**Category B**
Category 1	a	b
Category 2	c	d

The odds ratio would then equal to

$$OR = \frac{a \times d}{b \times c}$$

Example 2: Titanic Disaster Data

Using the same `titanic` data from previous example:

- Calculate the odds of survival for women and men.
- Calculate the odds of dying for women and and men.
- Compare the odds of survival between women and men.
- Calculate the actual odds ratio of survival between women and men.
- Did any of two sexes haave higher odds of survival?

The χ^2 contingency Test

When there are more than two categories for a variable, we cannot calculate odds ratio. Therefore we need a different statistical method. χ^2 Contingency test is used to test association between two categorical variables. Last week, we used goodness-of-fit approach for binomial and poisson distributions. Here, we will use goodness-of-fit to the null model of **independence** of the variables. χ^2 contingency test is actually a specific case of gooddness-of-fit test. Since we are interested in only the association between variables, we do not need to fit a model to the data. Therefore, expected frequencies are always calculated from the data itself. Consider the following contincency table

Var2 Var1	**Category A**	**Category B**
Category 1	a (cell #1)	b (cell #3)
Category 2	c (cell #2)	d (cell #4)
Category 3	e (cell #3)	f (cell #5)

Assuming two variables are independent of each other the expected frequency for a cell in the table can be calculated as

$$\frac{\sum row \times \sum column}{N}$$

where $\sum row$ and $\sum column$ are the sum of whole row and column that particular cell belongs to, and N is the sample size (sum of whole table). For example for the cell #5, the expected frequency would be

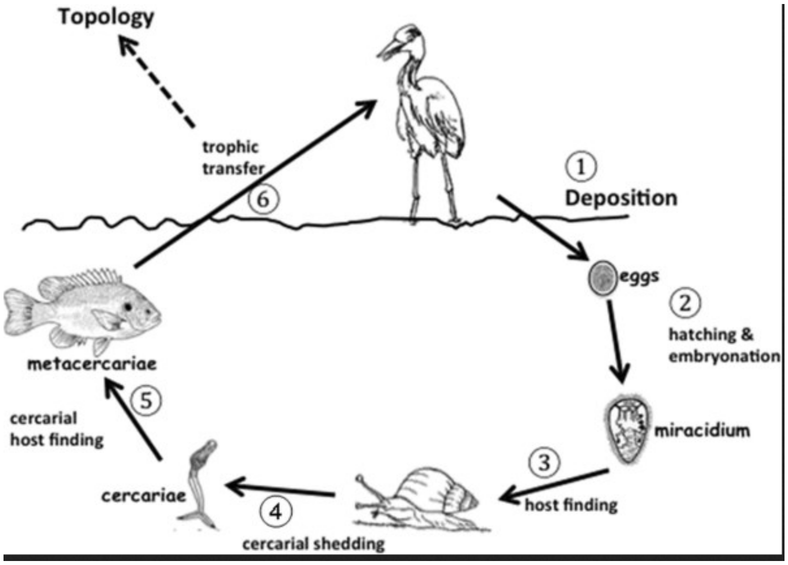
$$\frac{(b+d+f)(c+d)}{N}$$

Once every expected frequency is calculated using the equation above, χ^2 statistic can be calculated as usual. We can still use the `chisq.test()` function for the contingency test with which we won't have to calculate the expected frequencies.

As for every type of χ^2 test, we still need to specify a degrees of freedom. *df* for the contingency test is calculated using the formula $(ncol - 1)(nrow - 1)$

Example 2: `worm` data

The data we are going to use is stored in the `worm` object. Trematode is a species of worm that infect killifish and form encyst in its brain. In later stages of its life cycle, it is transferred to predator birds that eat the killfish. Researchers obsrved that infected fish spend excessive time near the water surface making themselves more vulnerable to the bird predator. This can be advantageous to the worm because it can pass to the predator bird to continue its life cycle. Researchers tested the hypothesis that the worm infection in killifish influences risk of predation by birds. They set up the experiment by dividing fish into three categories: uninfected, lightly infected and heavily infected and then recording how many of the fish are being eaten by the predator birds.



Using this data:

- Summarize the data to obtain frequencies for each category.
- Visualize the data using an appropriate plot.
- Calculate expected frequencies for (uninfected & eaten), (highly infected & eaten) and (highly infected & not eaten) categories.
- State null and alternative hypothesis for the test that researchers want to conduct.
- Test your hypotheses.
- What is your test statistic?
- According to the test result, what is your biological conclusion?

Fisher's Exact Test

Fisher's Exact test is specifically designed for 2x2 contingency tables. It is an improvement over the χ^2 contingency test when the assumptions of the χ^2 test are not met. Let's recall the said assumptions:

- None of the **expected** frequencies should be less than one
- No more than 20% of the categories should have **expected** frequencies less than 5.

As a rule of thumb, if the expected frequencies too low to conduct the χ^2 contingency test for an 2x2 table, we will use the Fisher's Exact test. It is called exact beause it calculates the exact p-value whereas the χ^2 test calculates a high-accuracy *estimate* for p-value. Test statistic of this test is the odds ratio. In `R`, we can use the `fisher.test()` function. Let's do the next example to to learn about the Fisher's test.

Example 4: Vampire bats

The data we are going to use is stored in the `vampire` object. The vampire bat found in Costa Rica, feeds on the blood of domestic cattle. The cat preferes cows to bulls, which suggests that the bats might respond to a hormonal signal. To investigate this, researchers compared the vampire bat attacks on cows in their estrous ("heat") with attacks on cows not in their estrous on a particular night. The relevant question would be: do cows in estrous have higher chance of being attacked when compared to the ones that are not in estrous?

- State the null and the alternative hypotheses.

- Create the frequency table for this data.
- Perform Fisher's Exact test and state your conclusion.
- Now perform χ^2 test with the same data for the same hypotheses.
- Calculate the expected frequency for bitten cows in estrous.
- Do the results of two tests in aggreement with each other?

Exercises

1. A study was performed on 200 students to investigate whether Vitamin A supplementation was effective in preventing colds during winter. Randomly chosen 100 students were given daily doses of the vitamin and another randomly chosen 100 students were given placebos. The number of students getting getting cold in winter was computed accross the two groups and the results are given in the following table:

	Cold	NoCold	Total
Vitamin A	15	85	100
Placebo	25	75	100
Total	40	160	200

Using this data:

- What statistical test should be used for testing the possible relationship between taking Vitamins or not and getting cold or not?
- State your hypotheses.
- Visualize the data with an appropriate plot.
- Perform your test of choice. What is your biological conclusion based on the test results?

1. A study was conducted on the efficacy and safety of zidovudine (AZT) in reducing the risk of mother to infant AIDS transmission. 363 HIV$^+$ women were randomized to receiving either AZT or an existing treatment. Of the 180 given AZT, 13 gave birth to children who tested positive for HIV within 18 months. Of the 183 given the conventional treatment, 40 gave birth to children who tested positive for HIV within 18 months.

	AZT	Other
HIV +	13	40
HIV -	167	143

- What is the relevant questiong to ask for this data?
- What are the null and the alternative hypotheses?
- Visualize the data with `mosaicplot()`. Put HIV status on the y axis and the drug treatment on the x axis. The order of the x axis should be "other" and then "AZT. The order of the y axis should be"NoHIV" and then "HIV"
- Perform an appropriate statistical test and state your biological conclusion.

1. Practice problem 9.6 from the book: Between 20 and 25 violent acts are portrayed per hour in children's TV programming. A study onthe possible link between TV viewing and aggression followed the TV viewing habits of children between 1 and 10 uears old. Of these children, 88 watched less than one hour of TV per day, 386 watched 1-3 hours per day, 233 wathced more than 3 hourd per day. 8 years later, researchers evaluated the kids to see if they had a police record or had assaulted another person resulting in injury. The number of aggressive individuals from three TV watching groups were 5, 87, 67 respectively.

I didn't put the data in a tabled form like above. as a practice, let's try to create the data from scratch (in the form of raw data like the ones we saw in the in-class examples, not in the table form) If you can't create the data, just copy and paste the code snippet below to obtain the data in your environment.

```
violence = data.frame(TV = factor(rep(c("<1 hour", "1-3 hours", ">3 hours"),
                                      times = c(88,386,233))),
                      Agressiveness = factor(rep(c("agressive", "notagressive",
                                                   "agressive", "notagressive",
                                                   "agressive", "notagressive"),
                                                 times = c(5,88-5,87,
                                                           386-87,
                                                           67,233-67))))
head(violence)
```

```
##        TV Agressiveness
## 1 <1 hour     agressive
## 2 <1 hour     agressive
## 3 <1 hour     agressive
## 4 <1 hour     agressive
## 5 <1 hour     agressive
## 6 <1 hour  notagressive
```

```
summary(violence)
```

```
##          TV           Agressiveness
##  <1 hour  : 88   agressive   :159
##  >3 hours :233   notagressive:548
##  1-3 hours:386
```

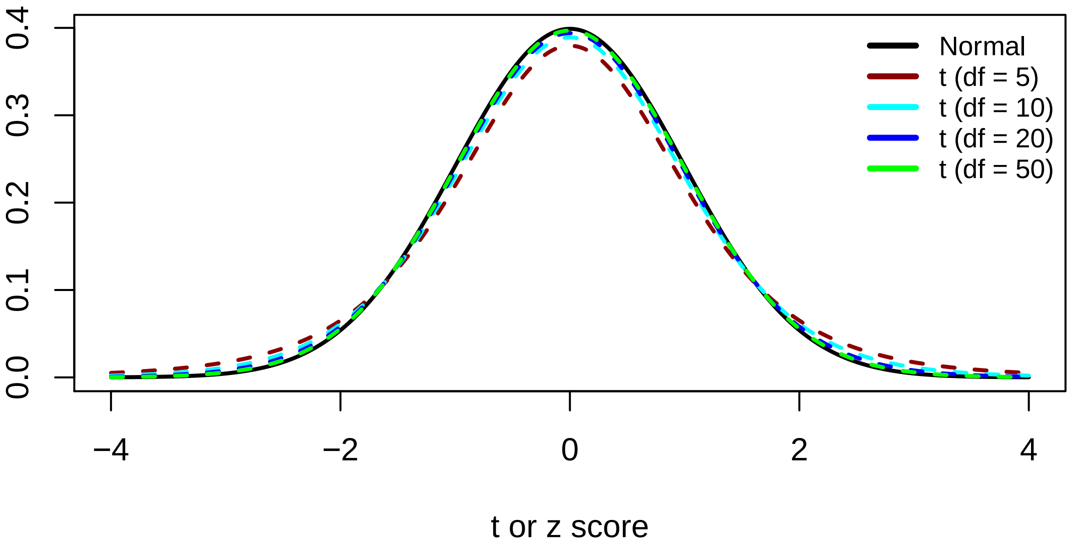
- How many variables are there in the data, what are their names?
- How many categories each variable has in the data?
- Produce a contingency table from the `violence` object.
- Plot the data with `mosaicplot()`. Put aggressiveness on the x axis and TV on the y axis. The order in x axis hould be "<1 hour", "1-3 hours" and ">3 hours". For y axis "notagressive" and "aggressive".
- Perform an appropriate statistical test to decide whether childhood TV viewing is associated with future aggressiveness.
- What are the null and the alternative hypotheses in your test?
- Check the assumptions of the test you just conducted.
- Does this prove that TV watching causes increased aggression in kids? Explain your reasoning.

Chapter 9: Inference for a Normal Distribution (t-test)

t-Distribution

In this week's lab, and in the following weeks, we will mostly perform hypothesis testing on variables that are **normally distributed**. If you recall the NASA astronaut hiring example from the probability distributions week, we need to know population parameters (μ, σ) to use the normal distribution in order to have a null distribution to calculate probabilities from. In real world examples researchers are not able to access the data for the whole population most of the time, which means they cannot know σ for calculating the standard error. This is where the t-distribution comes to their rescue: it has a slightly different formula in which they can use the standard deviation of the sample they have.

This new distribution has a degrees of freedom represented by $N - 1$ where N is the sample size. It has slightly fatter tails when compared to normal distribution which is implemented as a means of compensation since we calculate SE from sample and there is always an effect of sampling error. In other words, extreme values are given a little bit more probability in this distribution when compared to standard normal distribution. Lets see this difference in a simple plot of probability density functions of both distributions:



As you can see, t-distribution with very small df has fatter tails when compared to normal distribution. As the df increases, resulting t-distribution gets really close to the normal distribution. For this reason, we can use the t-distribution and thus the **t-test** when we are confident that the variable we are measuring is normally distributed in the original population.

t-test

There are different types of t-test you may want to use depending on your data and the questions you have about those data. But all address some questions regarding the population mean(s). Here we will see all of them one by one, starting **one-sample t-test**.

One-sample t-test

If you want to test your sample against the hypothesis regarding what the population mean is, you can use one-sample t-test. When performing a one-sample t-test, there is the implicit assumption that the variable you are testing is normally distributed. If not, you shouldn't use t-test with your data.

Now before we move on, here is how you can use the `t.test()` function in the one-sample t-test context:

```r
t.test(x  = data$variable #name of the variable that is being tested,
       mu = popMean #Population mean in accordance with the hypotheses,
       alternative = c('less', 'greater', 'two.sided') #direction of HA
       # Use one of the 3 options above. The default behavior is to perform
       # a two-sided test
       )
```

Let's illustrate its use with a real world example regarding mean body temperature of the human population:

Example 1: Human body temperature

The mean temperature of normal human body is 98.6°F. Body temperature measurements are obtained from randomly chosen healthy human sample. The data is given in `temp` object.

- Visually inspect the data to see if it seems like the assumption(s) of t-test is met.
- If it is appropriate, use `t.test()` function to test the hypotheses of:
  - H_0: $\mu = 98.6$
  - H_A: $\mu \neq 98.6$
- Based on the p-value, what can you say about your hypotheses?

Using t-test to compare two population means

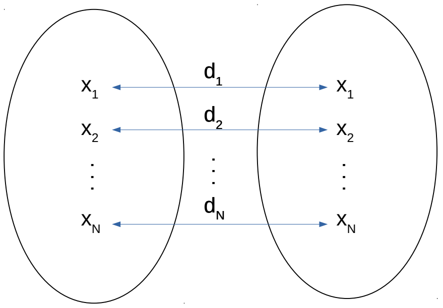
The above example was for testing a hypothesis regarding on population mean using your one sample. If you want to test a hypothesis regarding two different populations an their means, using samples drawn from each one of them, then you can use two different types of t-test. These are named **paired t-test** and **two-sample t-test**. We will start with the paired t-test first.

Paired t-test

When pairs of observations from the two samples of yours are dependent with each other, you should use the paired t-test. A basic example for such a case can be: Mean heart rate measurements from a sample of people before and after they have been in an extensive program of cardio. People we are taking measurements from are the same in both cases, the only thing that changes is that those same people worked out for some time before the second measurements take place.

Above, two samples are illustrated with two sets. Each observation denoted with x_i. When you have N samples in both groups, you have N number of pairwise differences (denoted as d_i). Paired t-test actually tests if mean of those all d values are significantly different than zero or not using a simple one-sample t-test in the background. Because of that, differences between the observations should distribute normally for us to use the paired t-test.

```r
t.test(formula = data$numVariable ~ data$groupVarible,
       paired=TRUE,
       alternative = c("less", "greater", "two.sided"))
```



You can see that the notation changes a little bit when you perform a paired t-test. Since now you have two different groups, you need a way to express this grouping to R. Above illustrated approach is called the formula notation, which follows the same logic as the notation that you saw when learning the `boxplot()` function.

Now, let's solve an example regarding paired t-test using the `t.test()` function:

Example 2: Blackbird serum antibody levels

To see the effect of testosterone on blackbird immune system, researchers measured the serum antibody levels before and after they have placed an implant that secretes testosterone to those same birds' body. Related data is given in the `blackbird` object. Using the data:

1. Inspect the dataset to see what each column corresponds to
2. Check the assumptions of the paired t-test visually to see if the data is suitable to apply this test on.
3. If so, use the `t.test()` function accordingly to test **whether there is any difference in terms of blood antibody levels between before and after groups at all**
4. If not, try the steps 1-3 with the logged version of the data.
5. What is your conclusion about the t-test result? Does testosterone change the bird's immunity?

Two-sample t-test

When you have two **independent** samples representing two groups that are going to be tested against each other, you should use the two-sample t-test. Since the observations in two samples are not paired with each other, sample sizes of the two groups don't have to be exactly the same. Having balanced samples in terms of size would provide us more reliable results nevertheless. In addition to the normality assumption, two-sample t-test needs that two populations that the two samples were drawn from have **equal variances**.

```
t.test(formula = data$numVariable ~ data$groupVarible,
       var.equal = TRUE,
       alternative = c("less", "greater", "two.sided"))
```

You may have noticed that we specifically say that we have equal variances with the `var.equal = TRUE` part in the above usage example. This is because although traditional t-test needs equal variances there are some mathematical corrections that can be applied if not. R assumes that variances are not equal if we don't specify the `var.equal = TRUE` and apply those corrections. We normally don't want that so `var.equal` should be set to TRUE if we can confirm that from our samples. To test if the variances are not significantly different from one another, you can use the `var.test()` function which performs the F-test for equal variances. Significant p-value indicates that two variances are not equal. If the result of your `var.test()` call says that two variances are different, you can use a different variant of t-test, **Welch's t-test** which will be shown in the next part.

Example 3: Horned lizards

Researchers tested the idea that long spikes provides lizards with better protection against predators, birds. They identified 30 horned lizards that have been killed by birds and measured their horn lengths. As a comparison group, they measured the horn lengths of 154 living lizards that were still alive and well.

1. Inspect the dataset to see what each column corresponds to
2. Check the assumptions of two sample t-test visually to see if the data is suitable to apply this test on.
3. If so, use the `t.test()` function accordingly to test whether the horn length **increases** the survival rate.
4. What is your conclusion about the t-test result? Does horn length increase survival rate?

Welch's (approximate) t-test

If you want to conduct a two-sample t-test but two variances are not equal (F-test/`var.test()` gives a significant p-value) you can use the `t.test()` with the added option of applying the **welch correction** which can specified as `var.equal = FALSE` inside the `t.test()` function. If `var.equal` is not set to `TRUE` explicitly, `R` will assume the variances are not equal and perform the t-test with welch correction automatically.

To illustrate the importance of the equality of variances, let's conduct a simple simulation to see how it affects the results. For the simulation,

- draw two random samples from normal distribution with the same mean but with different standard deviations.
- perform `t.test()` with both `var.equal=T` and `var.equal=F` arguments.
- store p-values of each test.
- repeat the above process 10k times using a `for` loop.

Example 4: t-test vs welch corrected t-test

Using above `simulation` data

- Plot the p-value distributions of both tests to see 'α' level differences.
- Calculate the proportion of significant p-values in both cases (both columns) at α=0.05. Are you seeing a difference?

- You know that in each trial (each row) both tests were performed on samples that are coming from populations with **same** mean. considering what t-test normally makes a conclusion about, do you think that this difference is normal?

Exercises

1. Dolphins In Northern Hemisphere, dolphins swim predominantly in a counterclockwise direction while sleeping. A group of researchers wanted to know whether the opposite was true for dolphins in the Southern Hemisphere. They watched eight sleeping dolphins and recorded the percentage of time that the dolphins swam clockwise. Assume that this is a random sample and that this variable has a normal distribution.

You can copy paste below code to save the data into your environment:

```
dolphins = c(77.7, 84.8, 79.4, 84.0, 99.6, 93.6, 89.4, 97.2)
```

- What is the mean percentage of swimming among those 8 dolphins that are being sampled?
- Test the alternative hypothesis that Southern Hemisphere dolphins spend more than half of their time while swimming clockwise. ($H_A : \mu > 0.5$)
- Using same test function as the above step, can you say what is the 95% confidence interval for the mean percentage swimming clockwise in S. Hemisphere?
- Again using the same function, this time modifying an argument, find what is the 99% confidence interval for the same population parameter.
- Based on your p-value from step 2, what can you conclude?

2. PlantGrowth One of the built-in `R` datasets, `PlantGrowth` contains results from an experiment to compare yields obtained under a control and two different fertilizer treatment condition. Using this data:

- Check the overall structure of data using `head()`
- Since there are 3 different groups (Control, treatments 1&2) you cannot use t-test directly on this data using formula notation. Subset the data into two parts that will correspond to the yield values of control and the treatment 2. Using `t.test()` function with appropriate arguments test that if treatment 2 increases the crop yields and make a conclusion based on the p-value.

3. Weddell seals Weddell seals live in the Antarctic and feed on fish during long dives in freezing water. The seals benefit form these feeding dives, but the food they gain comes at a metabolic cost. The dives are exhausting. A set of researchers wanted to know whether feeding was also energetically expensive, in addition to the exertion of a regular dive. They determined the metabolic cost of dives by measuring the oxygen use of seals as they surfaced for air after a dive. Using 10 seals, they measured the metabolic cost of 10 feeding dives and with the **same 10 seals**, they also measured a non-feeding dive. The data are given as follows:

```
seal = data.frame(Nonfeeding = c(42.2,51.7,59.8,66.5,81.9,82.0,81.3,81.3,96.0,104.1),
                  Feeding = c(71.0,77.3,82.6,96.1,106.6,112.8,121.2,126.4,127.5,143.1))
```

You can copy-paste the above code to obtain the data into your environment.

- What is the mean change in oxygen consumption in feeding dives compared with the non-feeding dives?
- Test whether feeding increases the metabolic cost of a dive ($H_A : \mu_d > 0$)
- What is the 99% confidence interval for the mean change for this difference?

Chapter 10: Handling Violations of Assumptions

The hypothesis testing methods we have learned so far for numerical variables require data to be normally distributed. Sometimes data may not meet this assumptions. What should we do in those cases? There are three alternatives to be considered:

- Ignore the violations of the assumptions
- Transform the data
- Use non-parametric test

These three options should be considered in the written order. If the sample size is high enough and violations of the assumptions are not too drastic, we can ignore the violations. If the deviation from normality is too high, we should consider transforming the data. If the transformed data has a distribution similar to normal distribution, we can contine with the parametric test. However, sometimes transformation may not work for data to be normally distributed. The last option we can use is to use a non-parametric test.

Detecting Deviations from Normality

Histogram

The very first thing we should check is to draw frequency distribution (histogram) of the data.

Quantile-Quantile (QQ) Plots

We can draw normal quantile plot to detect departures from normality. Normal quantile plot compares each quantile of the data with the quantile from normal distribution. We will use `qqnorm` function to draw normal quantile plot.

```
qqnorm(data_to_be_tested, col = , pch = )
qqline(data_to_be_tested) # adds straight expected line
```

Shapiro-Wilk test for Normality

Checking distribution of the data using histogram and qqplots is a subjective way of deciding whether the data is normally distributed or not. Shapiro-Wilk test is a formal test which is a goodness-of-fit test to normal distribution. It tests the null hypothesis that:

- Ho: data is a sample from a population having normal distribution
- Ha: data is a sample from a population not having normal distribution

---

```
shapiro.test(data_to_be_tested)
```

Example 1: Marine Reserves

Marine organisms are important for biological conservation. Marine reserves are marine protected areas that has protection against fishing. Scientists matched each of 32 marine reserves to a control location which was the site of the reserve before it became protected or a similar unprotected site nearby. They measured protection by biomass ratio, which is the total mass of all marine plants and animals in a protected unit area divided by the unproteted unit area. If the protection had no effect, biomass ratio would be one. If the biomass ratio would be higher than one, then it would mean the protection was beneficial. The data is given in 'marine' object.

- Check the data for normality by drawing its histogram, qq-plot and applying Shapiro test.
- Why or why not can we assume that data is normally distributed?

Using shapiro test sometimes might be misleading. If the sample size is very large, the test will be very powerful and detect very small deviations from normality even though the data look normally distributed and result in rejection of null hypothesis. And if the sample size is too small, power of the test will be too low and can not detect samples drawn from non-normally distributed populations.

Data Transformation

Data tranformation is one way of handling data that do not meet the assumptions of a statistical test. The aim of the transformation is to fit the data to normal distribution and to make standard deviations similar in two groups. Tranformation should be done to every data point exactly the same way. The most frequently used transformations are listed below in order of popular usage:

- log transform: for ratio data, right skewed data
- Other tranformation methods: arcsine tranform, square root transform, square tranform, antilog transform, reciprocal transform

---

Example 1 cont'd

- Try log tranformation to 'biomass' data
- Check if log transformed data can be assumed to be drawn from a normal distribution

---

Before performing log tranformation, make sure that all the values are positive. Logarithm of zero is not defined. Therefore, if there is zero in your data, add +1 to every data point and then take the logarithm.

Non-Parametric Tests

If the second option does not work for data to have normal distribution, we can use non-parametric tests for our last resort. Parametric tests make assumptions about the distribution, whereas non-parametric tests do not make assumptions about the distribution from which sample is taken.

Non-parametric tests are typically less powerful than parametric tests. Therefore, they are used as a last option. Most of the parametric tests use ranks instead of actual data points.

Wilcoxon signed-rank test (for one-sample t-test and paired t-test)

Wilcoxon signed-rank test evaluates whether the median of a population is equal to a null hypothesized value. `wilcox.test()` function performs this test. Use of its arguments are the same as `t.test()` function from last week. Assumption of this test is that distribution of measurements is symmetric around the median (no skewness).

```
# 1 sample usage
wilcox.test(x = data$variable, mu = nullHypothesis,
            alternative = c('less','greater','two.sided'))
# paired design usage, option 1
wilcox.test(formula = data$numVariable ~ data$groupVarible,
            paired = TRUE, mu = nullHypothesis,
            alternative = c("less", "greater", "two.sided"))
# paired design usage, option 2
wilcox.test(x = subsetted_sample1, y = subsetted_sample2,
            paired = TRUE, mu = nullHypothesis,
            alternative = c("less", "greater", "two.sided"))
# paired design usage, option 3
wilcox.test(x = subsetted_sample1 - subsetted_sample2,
            mu = nullHypothesis,
            alternative = c("less", "greater", "two.sided"))
# note that "mu" is by default 0
```

Example 2: Autism

A study assessed the effectiveness of a new drug designed to reduce repetitive behaviors in children affected with autism. A total of 8 children with autism enrolled in the study and the amount of time that each child is engaged in repetitive behavior during three hour observation periods are measured both before treatment and then again after taking the new medication for a periof of 1 week. The data is given in ‘autism’ object.

- Write the hypotheses (be careful about the order of the groups)
- How is the data organised, i.e. how are the groups given in the object?
- Calculate the pair-wise differences (be careful to be consistent about the order of groups)
- Check if we can assume that the differences are normally distributed
- Log-transform the differences by calculating logarithm of each value
- Check if we can assume that log transformed differences are normally distributed
- Choose whether to use t-test or wilcox test
- Apply the chosen test and interpret its result

Wilcoxon rank-sum test (Mann-Whitney U test) (for two-sample t-test)

Mann-Whitney U test compares the distributions of two groups. It has only one assumption that distributions of the two groups have same shapes. The same function `wilcox.test()` performs this test with the appropriate usage of arguments.

```
# 2 sample usage, option 1
wilcox.test(x = subsetted_sample1, y = subsetted_sample2,
            mu = nullHypothesis,
            alternative = c('less','greater','two.sided'))

# 2 sample usage, option 2
wilcox.test(formula = data$numVariable ~ data$groupVarible,
            mu = nullHypothesis,
            alternative = c("less", "greater", "two.sided"))
```

```
    # note that "mu" is by default 0
```

Example 3: Sexual Cannibalism

The sage cricket has an unusual form of mating. During mating, the male offers his fleshy hind wings to the female to eat. The wounds are not fatal, but a male with already nibbled wings is less likely to be chosen by females he meets subsequently. Females get some nutrition from feeding on the wings, which raises the question "Are females more likely to mate if they are hungary?" Researchers answered this question by randomly dividing 24 females into two groups: one group of 11 females was starved for at least two days and another group of 13 females was fed during the same period. Finally, each female was put separately into a cage with a single new male, the waiting time to mating was recorded. The data are given in 'cannibalism' object.

- How is the data organised, i.e. how are the groups given in the object?
- Using the `levels()` function check how R handles the order of the groups
- Write the hypotheses
- Check if we can assume that the data are normally distributed (check for the two groups separately)
- Decide whether a data transformation is required or not
- Choose whether to use t-test or wilcox test
- Apply the chosen test and interpret its result

Exercises

Question 1 (from r-bloggers.com)

A company sells the same product in two types of stores: classical and self-service stores. The data about income earned in each type of store are as follows:

Classical stores: 50, 50, 60, 70, 75, 80, 90, 85

Self-service: 55, 75, 80, 90, 105, 65

On the level of significance of 95%, is there a difference in income among different types of stores?

- State null and alternative hypotheses
- Check the assumptions of the test
- Decide whether you should use a parametric or non-parametric test (consider also sample size)

Question 2

Accounting data for sales showed that in randomly selected 15 stores the quantities of products sold are:

509, 517, 502, 629, 830, 911, 847, 803, 727, 853, 757, 730, 774, 718, 904

Unsatisfied with those results, a company decided to start advertising campaign. After the campaign finished, the amount of products sold in these same stores were:

517, 508, 523, 730, 821, 940, 818, 821, 842, 842, 709, 688, 787, 780, 901

Did the advertizing campaign produce statistically significant results?

- State your hypotheses
- Draw histogram and qqplot of the data
- Perform an appropriate statistical test.

Question 3

R has datasets that are loaded into environment by default. Write '`UKDriverDeaths`' and the data will be automatically loaded. Its class is different than what we see as a matrix in class. Simply convert the data to a numeric vector (using `as.numeric` function) and save it to an object. You can type '`?UKDriverDeaths`' to see the explanation of the data.

- Draw histogram of the data. What can you say about the distribution.
- Draw qqplot and add expected line on it.
- Perform shapiro test
- What can you say about the distribution of the data? Is it normally distributed?
- Try `log2` transformation on the data and repeat above steps again. Is data normally distributed now?
- Test whether mean of total car deaths is different than `1600`.
- Try to add density line to the histogram of the original data. Here is how to do it:

– Draw the histogram with probabilities, not frequencies (I draw the histogram with 40 breaks).
– Use '`density`' function and add it ass a line to the existing plot. You can use '`?density`' to see how the function works.

Chapter 11: Analysis of Variance (ANOVA)

Stripchart

Stripchart is a graphical representation similar to boxplot. Instead of showing median, quartiles and range of data, obervations themselves are shown on the graph. It is a convenient way of comparing distribution of multiple groups, when number of observations are low.

```
stripchart(x = , # data to be plotted
           method = , # how points will be plotted: jitter/overplot
           vertical = , # logical: T/F
```

Example

Download and load the 'week11.RData' into your environment. 'dat' and 'dat2' objects contain a hypothetical data of three groups.

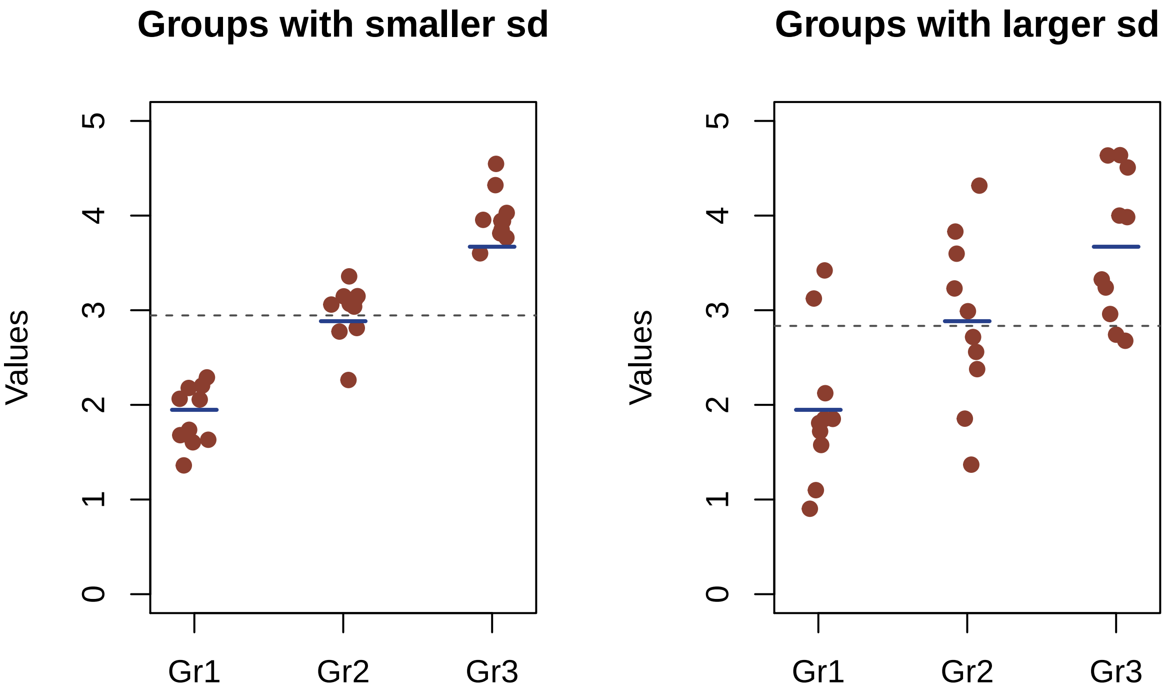
- Draw the stripchart of both data side by side.
- Draw the plots verticallly
- Add labels to the plot including title, x-axis label, y-axis label, color, point types etc.

Analysis of Variance

When we have more than two groups to compare their means with t-test, we have to have three hypotheses and apply three different tests to decide which group(s) differ from others. This analysis brings problems with itself because each time we conduct a test, we increase the probability of committing Type I error. Therefore, we need an alternative approach to test difference among groups means by using only one statistical test; called analysis of variance (ANOVA). Analysis of variance compares the means of more than two groups by analyzing variation in the data.

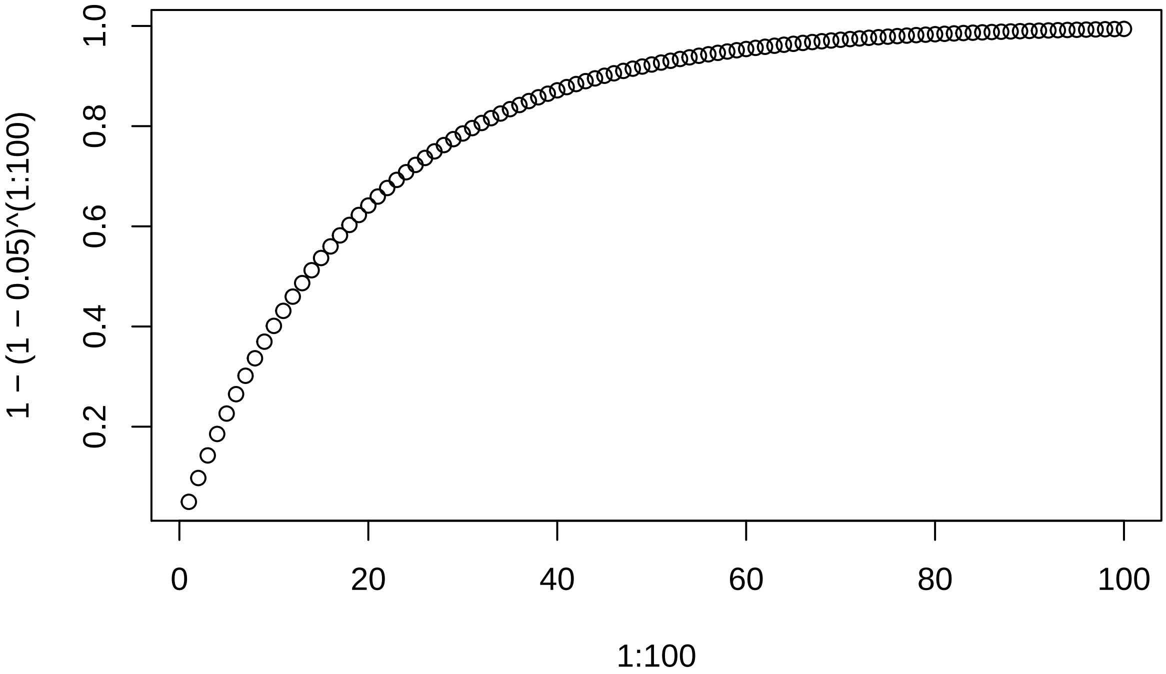
Null hypothesis in ANOVA is that all the groups have the same mean. Alternative hypothesis is that at least one of the groups differ from others. Since ANOVA cannot decide which group differs from others, we need other tests, called posthoc tests, to find the group having different mean.

Basically, ANOVA divides total variation in the data into two parts. One source of variation is due to the deviation of each observation from its group mean (error term, variation of each dot to blue line(group means)). Other source of variation is due to deviation between group means (blue lines) and grand mean(dashed line). If variation due to groups (deviation among groups) is significantly higher than variation inside the groups (error term), we reject null hypothesis. Observing below plots without doing any statistical analysis, notice that the variation inside the groups (error term) on the left panel is smaller than those of the right panel. Therefore, we can be more confident that groups on the left panel differ from each other (in fact at least one of them), while on the right panel groups seem to overlap and we are less certain whether their means actually differ or nor.



if we perform m hypothesis tests, what is the probability of at least 1 false positive? $1-(1-a)^m$

plot test number vs at least one FP rate:



ANOVA is performed using two functions. aov() makes the calculations and its output contains more information than it prints to the console. That information can be extracted using a second function: anova(). We store output of aov() in a variable, and give that variable as argument to anova() to retrieve the anova table.

```
dat.aov = aov(Numeric.Var. ~ Grping.Var., data = data)
anova(dat.aov)
```

The test statistics for the ANOVA is the variance ratio; F value. It is defined by the ratio of MS_{group} by MS_{error}. The significance of F value is calculated using F distribution. Similar to χ^2 distribution, F distribution is one sided. If null hypothesis is correct, F value should lie close to one. If null hypothesis is not correct, F value should be greater than one.

R-squared value R^2 is defined as the variability explained by the group differences. It can be calculated using anova table. R-squared is defined by the ratio of group sum of squares (SSg) divided by the total SS (sum of group SS and residual SS).

$$R^2 = \frac{SS_{group}}{SS_{group}+SS_{error}}$$

Example

In a study, it was hypothesized that human circadian clock can be reset by exposing the back of the knee to light. To test this hypothesis, an experiment was established that measure the circadian rhytme by daily melatonin production in 22 subjects randomly assigned to one of three light treatments. Subjects were awakened from sleep and light was exposed to either their eyes or knees or neither(control). Effect of treatment was measured two days later by the magnitude of phase shift in each subject's daily melatonin production. Data is given in 'circadian' object.

- Draw the stripchart of the data to visually inspect the groups.
- Calculate anova statistics of the data using necessary functions.
- what are the SS (sum of squares) for groups and erros?
- In plot of above example, comment which R^2 value would be greater?
- Calculate R^2 value using anova table.

Kruskal-Wallis test

If we think that assumptions of Anova are not met, we can use the non-parametric alternative called Kruskal-Wallis test.

```
kruskal.test(Numeric.Var. ~ Grping.Var., data = data)
```

Posterior tests

If we get a significant result from Anova, most of the time we want to make further analysis to see which group or groups cause the inequality of means. These tests make use of values calculated during ANOVA. There are different cases of posterior testing.

Unplanned case

When we have no prior idea about or concern about specific groups that disrupt equality, we will test for each pair-wise combination of groups. Tukey's test is one of the commonly used tests that control both the type 1 error rate due to multiple testing and power of the test due to multiple testing correction.

Output of the aov() function is given as argument to Tukey's test. As output, this functions returns us for each of the pairwise comparisons, a confidence interval for difference between two means, and a p-value for the test of equality of the two means.

```r
TukeyHSD(anov.result)
```

Example

- Make an unplanned comparison between all groups in circadian data.

Planned case

When we have a prior concern about difference between two specific groups, we can get away by making a single comparison. In circadian example, researchers are mainly interested in the difference between control and knee groups. They seek evidence supporting that knee group has lower values than control group. So, the null hypothesis is that knee group's mean is equal or higher than control, alternative hypotosis is that knee group's mean is lower than control.

In this case we do not need multiple tests and a singe t-test will suffice. However, we want to make use of the data from all three groups to enhance power of our tests. Hence, we will make a slightly modified t-test. Instead of standard deviation estimated from 2 samples, we will use MSe from 3 samples. So, the formula for standard error,SE:

$SE = \sqrt{MS_{error} * (\frac{1}{n_1} + \frac{1}{n_2})}$

and the rest is the same as t-test; t value will be then:

$t = \frac{\bar{Y_i} - \bar{Y_j}}{SE}$

Degrees of freedom for planned comparison will be the same as ANOVA test. Then, p-value can be calculated with these information using 'pt()' function.

Example

- Make a planned comparison between control and knee groups in circadian data.

Random Effects Anova

Sometimes the purpose of the analysis is not the difference between specific groups per se, but to reveal if some kind of grouping has an explanatory role. In those cases, the grouping variable does not consist of fixed values, but it has randomly chosen values from a larger set of possible values. We will make use of difference within and between randomly chosen groups to make an inference about the role of all such groups.

For the case of 1-way Anova (one explanatory variable), calculations of both fixed-effect and random-effect set-ups are the same. So we can analyse the data for the example of walking stick beetles in the same way.

Example

The walking stick, a wingless insect, was used in an experiment to study its adaptation to different plant species. Different traits were measured using digital photographs of specimens collected from the study. The traits were measured using a computer, so researchers were concerned about the measurement error. They took two separate photos of each specimen, and measured femur length on both photos for all specimens. If measurements differ between the two photos (taken from same specimen), this implies a measurement error. The relevant question is here how large was the measurement error compared with real variation among traits of individuals? Data is given in 'stick' object.

- Calculate anova statistics for stick data.
- Are differences in femur length due to measurement error or the real variation in the population?

Difference between mean measurements of random insects are significantly greater than difference within same insect's repeated measurements. So we can say that, although there is some variation (error) between different measurements of the same insects, those errors are small in comparison to difference between different insects.

Calculating repeatability

In random effect anova, both sources of variation in data (within groups and between groups) are random. Remember that in fixed-effects anova, within groups variation is due to error, hence random. However, between groups variation is not random because groups are fixed, predetermined. Therefore, MS_{error} and MS_{group} have different meanings in random effect anova.

Note: In the case of random effects Anova, we are not interested in difference between specific groups, hence between specific insects. If we get a significant result, our aim in a posterior analysis is then, how much of the total variance is due to the grouping variable.

Within group variation in the population, called σ^2, is estimated by MS_{error}. This represents the variance due to measurement error. Variation between groups in random effect anova are also random because groups (for example, individual insects chosen in the previous example) are chosen randomly. Since the sample represents a random sample of the population, we can use MS_{group} to calcuate variance in population, called σ_A^2, (variance in femur length in previous example). σ^2 and σ_A^2 are called variance components in random effect anova. Variance in population is estimated by:

$s_A^2 = \frac{MS_{group} - MS_{error}}{n}$

We will make use of the "Mean Squared Error" (Residuals) and "Mean Squared Group" valuess from the Anova table to calculate repeatability. Repeatability is specific to random effect anova. It represents the percent of variance due to group differences. High repeatability indicates within group variance is low which means low measurement error.

$repeatability = \frac{s_A^2}{s_A^2 + MS_{error}}$

Exercises

Load the data “Exercise11.RData”.

```
load("Exercise11.RData")
```

1. In the “disorders” data.frame, there are expression levels of PLP1 gene (normally distributed) from 45 subjects with 3 different conditions.

a. Based on this data, can you conclude that subjects with different conditions have different mean PLP1 expression levels? ##### b. If there are some significant differences, which of the groups have different means?

c. What proportion of the total variation is due to these conditions?

2. Katiaho et al. took 12 male dung beetles, and mated each of them to 3 different female beetles. They calculated an “average body-condition score” for offsprings of each female. It is known that the calculated score values are normally distributed.

a. Plot a *vertical* stripchart of the data (with open red circles, pch=1)

b. Based on this data, can we conclude that offsprings from the same father are more similar to each other?

c. Calculate repeatability of this trait (body-condition).

3. An important issue in conservation biology is how dispersal among populations influences the persistence of species in a fragmanted landscape. Four treatments were used to manipulate seed dispersal by changing the distance among experimental plant populations. The data below are the number of generations that the populations persisted (survived) in 5 replicates of each treatment.

Isolated: 6, 6, 8, 6, 9
Long: 5, 7, 7, 5, 15
Medium: 12, 4, 18, 4, 11
Continous: 9, 13, 16, 21, 11

Test whether any of the treatments have a different mean persistance, when it is given that survival times are not normally distributed.

4. A sequence of partially interbreeding populations are called a ring species, if neighbouring populations interbreed with each other, but the populations at both ends of the sequence do not interbreed.

Researchers measure red color density at some bird populations’ plumage (feathers). These populations form a ring species, and they want to know if the plumage of the most distant populations differ from each other. (Density values are normally distributed.)

a. Test if the populations differ with respect to plumage.

b. Test if the most distant two populations (A and D) differ with respect to plumage.

Chapter 12: Correlation

Correlation

Correlation analysis is a kind of hypothesis test for detecting whether 2 numerical variables are correlated linearly or not. The test statistic, correlation coefficient (sometimes called rho), is calculated that shows the degree of association. Correlation coefficient can be between -1 and 1, values closer to -1 correspond to a negative/inverse relation and values closer to 1 correspond to a positive/direct relation between the tested variables. If the variables are not associated (independent), correlation coefficient values are expected to be around 0.

The null hypothesis in correlation analysis is that the variables are independent (not associated). It can also be expressed as correlation coefficient = 0. The alternative hypothesis is that they are associated, or correlation coefficient != 0.

Similar to other hypothesis tests, there are parametric and non-parametric alternatives. Parametric alternative is called Pearson correlation and it assumes that the samples are taken from normally distributed populations. Non-parametric alternative does not have assumption of normality and it is called Spearman correlation.

Assumptions

- Random samples from the population.
- Two variables have bivariate normal distribution in 3D:
  - Linear relationship between the two variables
  - Variables have normal distribution

Pearson Correlation

`cor.test()` function performs a Pearson correlation analysis.

```
cor.test(var1, var2)
```

Order of the variables is not important, as correlation analysis does not assume a cause/effect relation between the variables. We can retrieve only the correlation coefficient or only the p-value using $ sign on the output object.

```
corr_coeff = cor.test(var1, var2)
corr_coeff$est
corr_coeff$p.value
```

Spearman Correlation

```
cor.test(var1, var2, method = 'spearman')
```

Example:Inbred Wolf Puppies In the data frame 'wolves', for some pairs of wolves, their inbreeding coefficient and number of their puppies that have survived a year are given. We want to test whether inbreeding coefficient and number of surviving puppies are associated or not.

- Draw a scatter plot of the data.
- Check the data for normality by applying Shapiro test.
- Write the null and alternative hypotheses.

- Apply a correlation test and interpret its results.

Example: Nicolas Cage Nicolas data contains information about the number of films Nicolas Cage appeared through years 1999 and 2009. Data also contains the number of people drowned in pools in those years. This data is an example of 'correlation does not imply causation' principle. We want to test whether there is a correlation between Nicolas Cage appearing in the movies and the people drowning in pools, and how to make a conclusion.

- Draw a scatter plot of the data
- Check the data for normality by applying Shapiro test.
- Write the null and alternative hypotheses.
- Apply a correlation test and interpret its results.
- After applying the test, you will see that there is a significant high correlation between those two variables. Can you say that people drowning in pools is affected by the number of films Nicolas Cage appears?

Example: left handed people advantage(book chp16 assignment problem) Left handed people have an advantage in many sports, and it has been suggested that lefthandedness might have been advantageous in hand-to-hand fights in early societies. To explore this potential advantage, researchers compared the frequency of left-handed individuals in traditional societies with measures of the level of violence in those societies.

- Draw a scatter plot of homicide rate against lefthanded percentage.
- Check the data for normality using histograms and shapiro tests.
- Write null and alternative hypotheses.
- Apply a transformation is necessary.
- Perform the appropriate correlation test and interpret the result.

Example: age of covid-19 deaths(from worlometer.com (5 May 2020)) Corona data contains the information about the deathrate of different age groups globally from covid-19 pandemic. We want to test whether there is a correlation between age and deaths from covid-19.

- Draw a scatter plot of the data.
- Check the data for normality using histograms and shapiro tests.
- Write null and alternative hypotheses.
- Apply a transformation is necessary.
- Perform the appropriate correlation test and interpret the result.

Exercises

1. Researchers want to test if looking similar to a poisonous fish species (from the same habitat) is beneficial for a fish species in terms of keeping predators away. For testing this, they set up 4 dummy plastic fishes with varying degree of similarity (resemblence) to the poisonous pufferfish, and counted number of predators that come to close proximity of the plastic fish, within 5 minutes. Degree of resemblence is encoded from 1 to 4, where 1 is most similar. If there is such a beneficial effect, we expect a positive association, where most similar (1 or closer to 1) fish attract least number of predators and least similar (4 or closer to 4) fish attract higher number of predators.

- Check the "pufferfish" data.frame. How many observations and how many variables are there?
- Visualise the data.
- Check the assumption of normality for both variables.
- Make a correlation analysis, write the hypothesis and select the appropriate test.
- Make a statistical and a biological conclusion.

2. In the height_weight data.frame, there is made-up data of height and weight measurements of people from 2 different job types. Both of the variables are normally distributed, you do not need to check for assumption of normality.

- Check whether height and weight values are correlated among job 1
- Check whether height and weight values are correlated among all individuals in the data set.
- Compare results of the 2 correlation tests. Which one has a correlation coefficient with higher magnitude? Which one has a more significant p-value? Is this discrepancy expected? How can we explain this difference?

Chapter 13: Linear Regression

Similar to correlation analysis of last chapter, regression analysis tests an association between 2 numerical variables. In regression analysis, however, a cause/effect relation is assumed. One of the variables should be specified as independent/explanatory variable (X) and one of them should be dependent/response variable (Y). Note that, this should be based on prior information, and decision of which one of the variables determines the other one is not a part of regression analysis.

Here we will study simple linear regression, and in this case a line equation is fitted to the data. Then significance (usefulness) of that line equation is tested. If the calculated line equation significantly explains values of the response variable (Y) based on values of explanatory variable (X), we can use it to predict values of Y that correspond to values of X, which are not observed in the sample.

Assumptions of the linear regression is that, for each X there exist a normal distribution of Y values, whose means are on a straight line. And these distributions have equal variances. In other words, X explains changes in Y, their relation is linear and residuals are normally distributed with equal variances.

A line can be fitted for any given data. After fitting the line, we will make a hypothesis test, where the null hypothesis is that slope of the regression line $= 0$ (hence X does not explain Y) and alternative hypothesis is that slope of the regression line $\neq 0$ (X does explain Y).

Calculating Regression Line

`lm()` function which stands for linear model, calculates a regression line from the data. Formula notation should be used to input the variables. It is useful to store the output in an object, as we will need to use the output further.

```
regression_model = lm(Y_variable~X_variable, data = AnalysedDataFrame)
```

Checking Assumptions

Linear regression model assumes that the residual terms are normally distributed, we can check this assumption by retrieving residual values from the line fitting process before, and applying Shapiro test to those residual values.

```
residuals = regression_model$residuals
shapiro.test(residuals)
```

In addition, we have to check (at least) visually whether variances of the residuals are different for different values of X.

Checking Significance of Regression Line

There are 2 alternative ways to test significance of the calculated line. `summary()` function describes the information from the line fitting process. It also gives anova and t-test results for the null hypothesis that slope $= 0$, both of them will produce the same p-value. Alternatively, using the `anova` function, we can get the anova table for the regression line.

```
summary(regression_model)
# or
anova(regression_model)
```

Predictions Based on Regression Line

If the fitted regression line is significant, it can be used to predict new value of Y. We can retrieve the predicted values together with their interval estimates, accounting for the uncertainty.

```
predict(regression_model, data.frame(ExplanatoryColumnName = c(some_values)),
    interval =  "prediction")
```

Example: Chickadee

In the data frame 'chickadee', for some predator bird species, their average mass and average alarm song length of the prey birds are given. You want to know whether predator mass does explain alarm song length of the prey bird.

- Calculate the regression line.
- Draw a scatter plot of the data. Then, add the calculated line to the plot.
- Draw a scatter plot of the residuals. Check for uneven distribution of residuals for different X values.
- Check the residuals for normality by applying Shapiro test.
- Write the null and alternative hypotheses.
- Test whether the calculated regression model significantly explain the relation of X and Y.
- Predict song length values that correspond to mass values 0.6 and 0.8.

Exercises

Question 1

In a field experiment, effect of nutrient type of soil on plant species diversity is studied. Researchers want to know whether increasing nutrient type changes number of plant species, and if it changes, how can plant species number be predicted from nutrient type.

Use the "plantnutrition" data.frame. Calculate the regression line. Visualize the data and add the regression line to the plot. Check if residuals are normally distributed. Check if the slope of the regression line is significant. Calculate the predicted confidence interval of number of plant species when nutrient type is 2.

Question 2

In the height_weight data.frame, there is made-up data of height and weight measurements of people from 2 different job types. We want to predict weight values from height. Y values for each X are normally distributed, you do not need to check for assumption of normality.

- Calculate and check significance of the regression line for "O1" only and then for all observations. (We will calculate 2 distinct models)
- For both of the models, predict weight values that correspond to height values 165 and 175. Compare the prediction intervals. Are the prediction intervals for height 165 and height 175 intersecting or not? Are they different in 2 models? What can be a reason for a potential difference?