



**University of
Zurich^{UZH}**



MAKERERE UNIVERSITY

Data Analysis with R:

Lecture Slides (all)

Sonja Hartnack, Terence Odoch & Muriel Buri

October 2018

Lecture Slides for Day 1

Goals of the course

To be able to...

- import data sets to R
- describe data with R
- apply basic statistical tests in R
- some ideas for more advanced statistical tools ...
- simulate a data set similar to own research

Course schedule:

- Starting at 9:00am / 9:30am (?)
- Tea breaks in between
- Lunch break
- Teaching until 4.30pm (~ 5pm)

Optaining a certificate is conditional on:

- active participation in class
- attending at least 75 % of the course (lecture & exercises)
- short final exam (format to be defined)

Getting to know each other

- My name is ...
- I am doing a Master / a PhD in ...
- I hope to learn in this course how to
- My personal goal for this course is ...

How do we reach these goals

- hands on exercises with R:
 - `chickwts`
 - `ToothGrowth`
 - `bacteria`
 - `perulung`
 - ... and others.
- interactive discussions & student's present their own solutions
- ask us a lot of questions but also ask google for help!
- group work
- short motivational lectures

Get started with data set: chickwts

An experiment was conducted to measure and compare the effectiveness of various feed supplements on the growth rate of chickens.

```
# load data set "chickwts"
data(chickwts)
# the head(...) function shows the first 6 observations
head(chickwts)
```

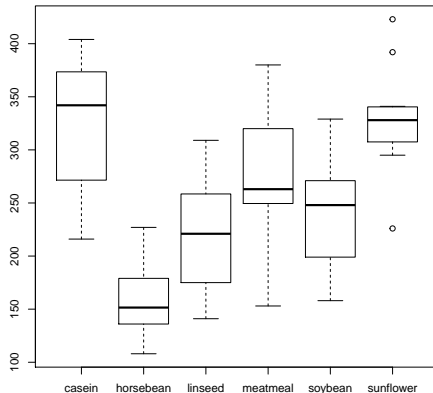
##	weight	feed
## 1	179	horsebean
## 2	160	horsebean
## 3	136	horsebean
## 4	227	horsebean
## 5	217	horsebean
## 6	168	horsebean

```
# FUNCTION - open bracket - DATA SET / VARIABLE - close bracket
```

Ideas for plotting the data

Ideas for plotting the data

```
# use x axis to show the categorical variable (feed),  
# y axis to represent the continuous variable (weight)  
# boxplot (y.cont.variable ~ x.cat.variable, data = dataset)  
# ?boxplot  
boxplot(weight ~ feed, data = chickwts)
```



Ideas for analysing the data

Ideas for analysing the data

```
anova <- aov(weight ~ feed, data = chickwts)
summary(anova)

##              Df Sum Sq Mean Sq F value    Pr(>F)
## feed           5 231129   46226   15.37 5.94e-10 ***
## Residuals     65 195556    3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(aov(weight ~ feed, data = chickwts))

##              Df Sum Sq Mean Sq F value    Pr(>F)
## feed           5 231129   46226   15.37 5.94e-10 ***
## Residuals     65 195556    3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Functionalities in R and RStudio

A hands on example

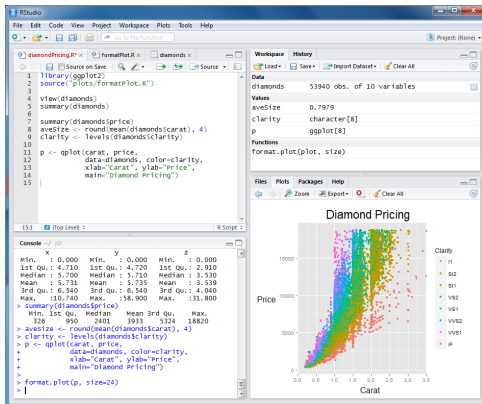


```
x <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
y <- c(20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30)
plot(x, y)
```

Functionalities in R and RStudio



- Source
- Console
- Environment, History, Files
- Files, Plots, Packages, Help





- Define manually a new folder called **RCourse** in your personal documents on your personal computer
- Know in which directory you are

```
getwd()
```

```
## [1] "/home/mburi/ownCloud/git/DataAnalysisWithR/Lectures"
```

- Set directory path

```
# back- and forslash is dependent on the system
```

```
setwd("C:/Users/muriel/Documents/RCourse/")
```

```
setwd("C:\\Users\\muriel\\Documents\\RCourse\\")
```

- Always clean up before starting with new R-Script

```
rm(list=ls()) # empty workspace, delete previously saved variables
```



```
?chickwts  
?boxplot
```

Also, have a look at the examples at the end of the help pages.

Exercise 1



A data frame in R: chickwts



chickwts[ROWS , COLUMNS]

	weight	feed
1	179	horsebean
2	160	horsebean
3	136	horsebean
4	227	horsebean
5	217	horsebean
6	168	horsebean
7	108	horsebean
8	124	horsebean
9	143	horsebean
10	140	horsebean
11	309	linseed
12	229	linseed
13	181	linseed

chickwts[6, 1]

	weight	feed
1	179	horsebean
2	160	horsebean
3	136	horsebean
4	227	horsebean
5	217	horsebean
6	168	horsebean
7	108	horsebean
8	124	horsebean
9	143	horsebean
10	140	horsebean
11	309	linseed
12	229	linseed
13	181	linseed

chickwts[11, 2]



Values of ...

```
# ... all columns of sixth observation:  
chickwts[6, ]  
  
# ... all columns of sixth to eleventh observation:  
chickwts[c(6:11), ]  
  
# ... all columns of sixth, eleventh and twentieth observation:  
chickwts[c(6, 11, 20), ]  
  
# ... all rows of first column (weight):  
chickwts[, 1]  
  
# ... all rows of second column (feed):  
chickwts[, 2]  
  
# or use the "$" sign as a reference to column "feed":  
chickwts$feed
```

Lecture Slides for Day 2

What is a data frame in R?

A data frame is used for storing a list of vectors of equal length. For example, the following variable `df` is a data frame containing three vectors `n`, `s`, `b`.

```
n <- c(2, 3, 5)
s <- c("aa", "bb", "cc")
b <- c(TRUE, FALSE, TRUE)
df <- data.frame(n, s, b) # df is a data frame
```

The characteristics of a data frame are:

- The column names should be non-empty.
- The row names should be unique.
- Each column should contain same number of data items.

Data frame in R

```
a <- c(1, 2, 3, 4)
```

```
a
```

```
## [1] 1 2 3 4
```

```
data.frame(a)
```

```
##      a
```

```
## 1 1
```

```
## 2 2
```

```
## 3 3
```

```
## 4 4
```

```
b <- c("d", "h", "h", "d")
```

```
dat <- data.frame(a, b)
```

```
dat
```

```
##      a b
```

```
## 1 1 d
```

```
## 2 2 h
```

```
## 3 3 h
```

```
## 4 4 d
```

Data frame in R: How to add a variable (var)

```
my.var <- c(1.3, 1.5, 1.8, 2.4)
# use "$" to refer to the additional vector variable
dat$my.var1 <- my.var
dat$my.var2 <- my.var
dat

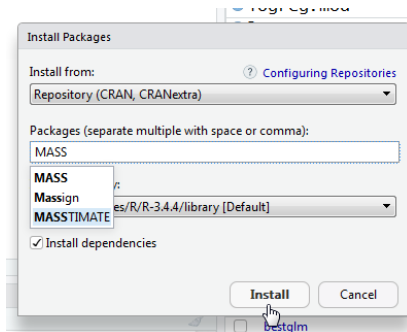
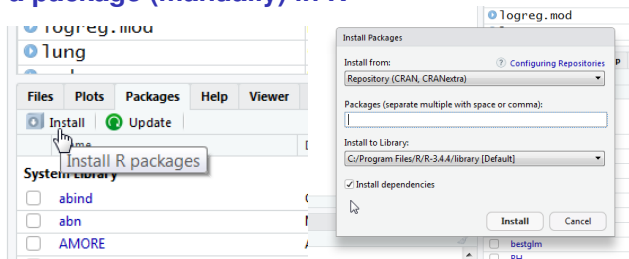
##      a b my.var1 my.var2
## 1 1 d      1.3      1.3
## 2 2 h      1.5      1.5
## 3 3 h      1.8      1.8
## 4 4 d      2.4      2.4

# What is the dimension (number of rows and columns) of our data frame?
dim(dat) # 4 rows and 4 columns

## [1] 4 4
```

Exercise 2

How to install a package (manually) in R



Using R is like cooking ...

Get into the kitchen	Change working directory
Get specialist electric tools into your kitchen (e.g. blender, ice-cream maker, etc.)	Install packages
Switch on your specialist electric tools	Load packages using the "library" function
Bring in your ingredients	Import data and save to R data frames
Check your ingredients	Use the function "summary" and basic tables to check your data for missing or implausible values (e.g. a number in a variable where "yes" or "no" are expected)
Chop things up (if required)	Split or filter data
Cook, using general and specialist tools	Carry out further descriptive and test statistics

How to install a package in R



```
# INSTALL package (only done ONCE!)  
install.packages("MASS")  
# LOAD package (whenever you use something from it!)  
library("MASS")  
data(bacteria)  
?bacteria
```

Exercise 3



How to google for getting help in R

- Google for **select observations in R**.



Objects are assigned values using $<-$,
an arrow formed out of $<$ and $-$. For example, the following
command assigns the value 1 to the object a.

```
a <- 1 # ALWAYS use "gets" assignment operator!  
# a = 1 # DO NOT USE the equal sign as the assignment operator!
```

After this assignment, the object a contains the value 1. Another
assignment to the same object will change its value.

```
a <- 5
```

Examples of assigned objects: single number



```
a <- 1
b <- 2
c <- a + b # c = 3
c

## [1] 3
```

Examples of assigned objects: vector



```
a <- c(1, 2, 3, 4, 5)
```

```
b <- 1
```

```
c <- a + b
```

```
c
```

```
## [1] 2 3 4 5 6
```

Examples of assigned objects: model



```
anova_model <- aov(weight ~ feed, data = chickwts)
summary(anova_model)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## feed           5 231129   46226    15.37 5.94e-10 ***
## Residuals     65 195556     3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Examples of assigned objects: data frame



```
bac <- bacteria
str(bac) # $ week: int  0 2 4 11 0 2 6 11 0 2 ...

## 'data.frame': 220 obs. of  6 variables:
## $ y   : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 1 2 2 2 ...
## $ ap  : Factor w/ 2 levels "a","p": 2 2 2 2 1 1 1 1 1 1 ...
## $ hilo: Factor w/ 2 levels "hi","lo": 1 1 1 1 1 1 1 1 2 2 ...
## $ week: int  0 2 4 11 0 2 6 11 0 2 ...
## $ ID  : Factor w/ 50 levels "X01","X02","X03",...: 1 1 1 1 2 2 2 2 3 3 ...
## $ trt : Factor w/ 3 levels "placebo","drug",...: 1 1 1 1 3 3 3 3 2 2 ...

bac_sub <- subset(bac, week == 2)
str(bac_sub) # $ week: int  2 2 2 2 2 2 2 2 2 2 ...

## 'data.frame': 44 obs. of  6 variables:
## $ y   : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 1 2 2 2 ...
## $ ap  : Factor w/ 2 levels "a","p": 2 1 1 2 2 1 1 2 2 2 ...
## $ hilo: Factor w/ 2 levels "hi","lo": 1 1 2 2 2 2 1 1 2 1 ...
## $ week: int  2 2 2 2 2 2 2 2 2 2 ...
## $ ID  : Factor w/ 50 levels "X01","X02","X03",...: 1 2 3 4 5 6 7 8 9 11 ...
## $ trt : Factor w/ 3 levels "placebo","drug",...: 1 3 2 1 1 2 3 1 1 1 ...
```



The `str` function displays the structure of an R object. One line for each "basic" structure is displayed.

```
## 'data.frame': 44 obs. of 6 variables:  
## $ y : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 1 2 2 2 ...  
## $ ap : Factor w/ 2 levels "a","p": 2 1 1 2 2 1 1 2 2 2 ...  
## $ hilo: Factor w/ 2 levels "hi","lo": 1 1 2 2 2 2 1 1 2 1 ...  
## $ week: int 2 2 2 2 2 2 2 2 2 2 ...  
## $ ID : Factor w/ 50 levels "X01","X02","X03",...: 1 2 3 4 5 6 7 8 9 11 ...  
## $ trt : Factor w/ 3 levels "placebo","drug",...: 1 3 2 1 1 2 3 1 1 1 ...
```

Exercise 4



- numeric

```
data(ToothGrowth)
ToothGrowth$len[1:6]

## [1]  4.2 11.5  7.3  5.8  6.4 10.0

class(ToothGrowth$len[1:6])

## [1] "numeric"
```

- integers

```
bacteria$week[1:6]

## [1]  0  2  4 11  0  2

class(bacteria$week[1:6])

## [1] "integer"
```

- (un/ordered) factor

```
chickwts$feed[1:6]

## [1] horsebean horsebean horsebean horsebean horsebean horsebean
## Levels: casein horsebean linseed meatmeal soybean sunflower

levels(chickwts$feed)[1:3]

## [1] "casein"      "horsebean"  "linseed"
```



Ordinal variables are represented as ordered factors:

```
bac_growth <- c("none", "+", "++", "+", "+++", "+", "none") # vector
bac_growth <- factor(bac_growth, levels = c("none", "+", "++", "+++"),
                     order = TRUE)

bac_growth

## [1] none +      ++      +      +++     +      none
## Levels: none < + < ++ < +++

#
mood <- c("OK", "Well", "Super", "Super", "Don't ask", "OK") # vector
mood <- factor(mood, levels = c("Don't ask", "Well", "OK", "Super"),
               order = TRUE)

mood

## [1] OK          Well          Super          Super          Don't ask OK
## Levels: Don't ask < Well < OK < Super
```

Exercise 5



Exercise 6





- numeric variable
- integer variable
- variable with two levels (binary factor)
- ordered variable with **more than** two levels (ordinal)
- unordered variable with **more than** two levels (nominal)

Rules for importing data into R (from Excel)



- First row of excel sheet contains **variable names**:
y, ap, hilo, week, ID, trt.
- Columns of excel sheet represent **variables**.
- Rows of excel sheet represent **observations per individual** (except for the first row).

	A	B	C	D	E	F	G	H	I	J
1	id	fev1	age	height	sex	resp	symptoms			
2	1	1.56	9.593	124.8	0	0				
3	2	1.18	7.491	111	1	0				
4	3	1.87	9.864	135.7	0	0				
5	4	1.49	8.588	119.1	0	0				
6	5	1.62	8.967	120.9	1	0				
7	6	2.11	9.293	134.3	0	1				
8	7	1.73	9.574	122.1	1	0				
9	8	1.47	8.493	122.6	0	1				
10	9	1.83	8.468	126.8	1	0				
11	10	1.41	9.029	126	0	0				
12	11	1.27	8.274	128	0	0				
13	12	1.34	8.416	127	0	0				
14	13	1.64	9.629	133.7	0	0				
15	14	1.57	8.622	125.5	1	0				
16	15	1.51	9.033	125.9	1	0				
17	16	1.25	8.643	122.3	0	0				



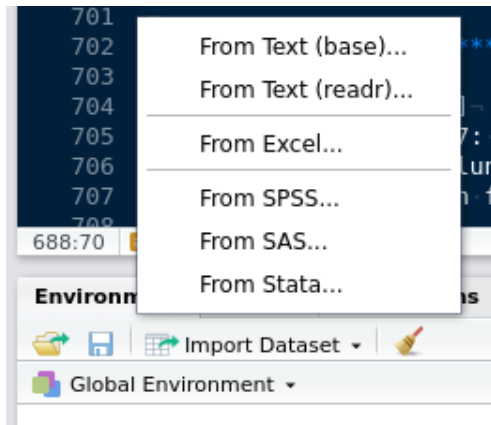
Variable names should ...

- start with a letter (not a number):
y, ap, hilo, week, ID, trt
- longer variables names should be separated with dots:
`time.in.weeks`
- do not use special characters, such as /, #, @, &, *, ...

How to import external data files into R?



- > Import Dataset > **From Text (base)...** > CSV Files (.csv)
or
- > Import Dataset > **From Excel...**





- Environment (upper right corner)
- > Import Dataset > **From Text (base)...** > CSV Files (.csv)

```
perulung_ems <- read.csv("perulung_ems.csv", row.names = 1,  
                        sep = ";")  
lung <- data.frame(perulung_ems)
```

- > Import Dataset > **From Text (base)...** > Text Files (.txt)
- > Import Dataset > **From Excel...** > Excel Files (.xlsx)

```
install.packages("readxl")  
library("readxl")  
perulung_ems <- read_excel("perulung_ems.xlsx")  
lung <- data.frame(perulung_ems)  
head(lung)
```

How to import .txt and .csv files into R? (1/2)

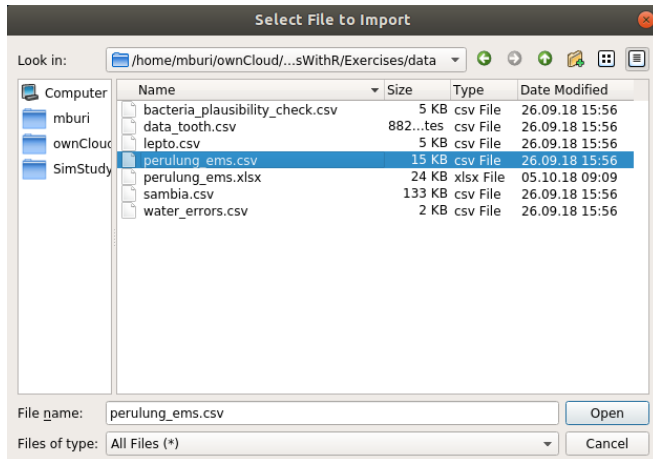


- Environment (upper right corner)
- > Import Dataset > From Text (base)... > CSV Files (.csv)

How to import .txt and .csv files into R? (1/2)



- Environment (upper right corner)
- > Import Dataset > From Text (base)... > CSV Files (.csv)



How to import .txt and .csv files into R? (2/2)



Import Dataset

Name:

Encoding:

Heading: ☒ Yes ☐ No

Row names:

Separator:

Decimal:

Quote:

Comment:

na.strings:

☒ Strings as factors

Input File

```
id;fev1;age;height;sex;respsymptoms
1;1.56;9.593;124.8;0;0
2;1.18;7.491;111.1;0
3;1.87;9.864;135.7;0;0
4;1.49;8.588;119.1;0;0
5;1.62;8.967;120.9;1;0
6;2.11;9.293;134.3;0;1
7;1.73;9.574;122.1;1;0
8;1.47;8.493;122.6;0;1
9;1.83;8.468;126.8;1;0
10;1.41;9.029;126.0;0;0
11;1.27;8.274;128.0;0;0
12;1.34;8.416;127.0;0;0
13;1.64;9.629;133.7;0;0
14;1.57;8.622;125.5;1;0
```

Data Frame

id	fev1	age	height	sex	respsymptoms
1	1.56	9.593	124.8	0	0
2	1.18	7.491	111.0	1	0
3	1.87	9.864	135.7	0	0
4	1.49	8.588	119.1	0	0
5	1.62	8.967	120.9	1	0
6	2.11	9.293	134.3	0	1
7	1.73	9.574	122.1	1	0
8	1.47	8.493	122.6	0	1
9	1.83	8.468	126.8	1	0
10	1.41	9.029	126.0	0	0
11	1.27	8.274	128.0	0	0
12	1.34	8.416	127.0	0	0
13	1.64	9.629	133.7	0	0
14	1.57	8.622	125.5	1	0

```
perulung_ems <- read.csv("perulung_ems.csv", row.names = 1,
                        sep = ";")
lung <- data.frame(perulung_ems)
```

How to import .xlsx files into R? (1/3)

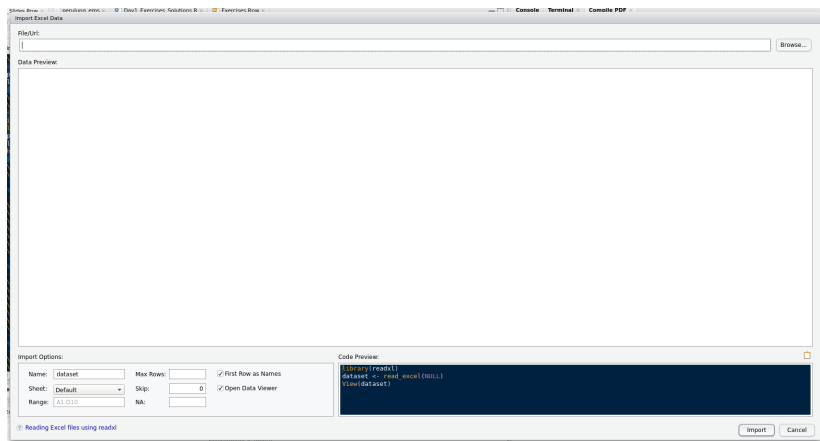


- Environment (upper right corner)
- > Import Dataset > From Excel... > Excel Files (.xlsx)

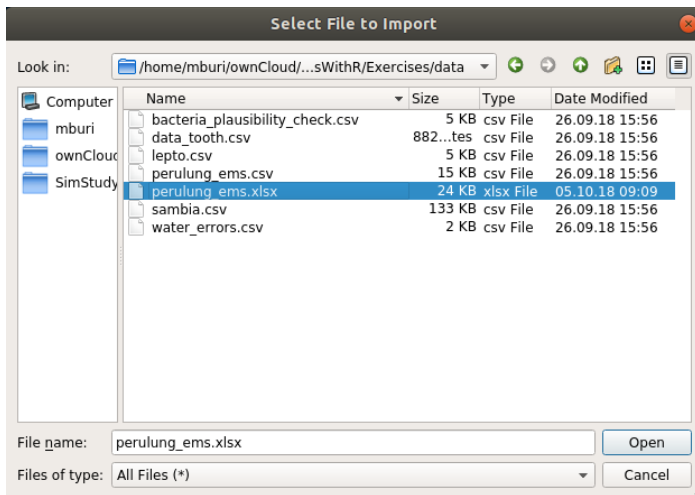
How to import .xlsx files into R? (1/3)



- Environment (upper right corner)
- > Import Dataset > From Excel... > Excel Files (.xlsx)



How to import .xlsx files into R? (2/3)



How to import .xlsx files into R? (3/3)



Import Excel Data

File Path: Update

Data Preview:

id	fec1	age	height	sex	respsymptoms
(double)	(double)	(double)	(double)	(double)	(double)
1	1.56	9.593	126.8	0	0
2	1.38	7.483	111.0	1	0
3	1.87	9.864	135.7	0	0
4	1.49	8.588	119.1	0	0
5	1.62	8.967	120.9	1	0
6	2.31	9.293	134.3	0	1
7	1.73	9.574	122.1	1	0
8	1.47	8.493	122.6	0	1
9	1.83	8.488	126.8	1	0
10	1.41	9.629	126.0	0	0
11	1.27	8.274	128.0	0	0
12	1.34	8.416	127.0	0	0
13	1.64	9.629	135.7	0	0
14	1.57	8.822	125.5	1	0
15	1.51	9.633	125.9	1	0
16	1.25	8.643	122.3	0	0
17	1.55	9.167	126.5	1	0
18	2.23	9.035	123.0	1	1
19	2.09	10.090	136.5	0	0
20	1.38	8.658	119.3	0	1
21	1.76	9.602	129.3	1	0
22	1.71	9.676	130.3	0	0

Previewing first 50 entries.

Import Options:

Name: Max Rows:
Sheet: Skip: ☒ First Row as Names
Range: NA: ☒ Open Data Viewer

Code Preview:

```
library(readxl)
perulung_ems <- read_excel("Exercises/data/perulung_ems.xlsx")
View(perulung_ems)
```

[Reading Excel files using readxl](#)

```
perulung_ems <- read_excel("perulung_ems.xlsx")
lung <- data.frame(perulung_ems)
head(lung)
```



Data from a study of lung function among children living in a deprived suburb of Lima, Peru. Data taken from Kirkwood and Sterne, 2nd edition.

Variables:

- fev1: in liter, "forced expiratory volume in 1 second" measured by a spirometer. This is the maximum volume of air which the children could breath out in 1 second
- age: in years
- height: in cm
- sex: 0 = girl, 1 = boy
- respsymp: respiratory symptoms experienced by the child over the previous 12 months

Lecture Slides for Day 3

Why do we need Statistics?

Repeatability of results:

Statistical science allows us to estimate what might happen if an experiment was repeated - but without having to actually repeat it!

Why do we need Statistics?

- Study results must be shown to be robust, i.e. real and not due to random chance
- Best way to demonstrate this is to repeat the same experiment/study many times each with **different** subjects (animals) drawn from the **same study population** and show that the result is truly repeatable
- It is generally totally impractical, in terms of both time and resources, to repeat an experiment many times!

Why do we need Statistics?

- Instead of repeating the experiment many times **probability theory i.e. statistics** is used to **estimate** what might have happened if the experiment had been repeated
- A mathematical model is used to fill this “data gap”
- Generally the most difficult task in statistics is to decide what “model” is most appropriate for a given experiment

What is Statistics? - A definition

A set of analytical tools designed to quantify uncertainty

- If an experiment or procedure is repeated, how likely is it that the new results will be similar to those already observed?
- What is the likely variation in results if the experiment was repeated?

What is Statistics? - A definition

The key scientific purpose of statistics

- to provide **evidence** of the existence of some “effect” of scientific interest
- i.e. evidence based medicine

As a reminder: The importance of study design

Even the most sophisticated statistical analyses cannot rescue a poorly designed study

→ unreliable results

→ inability to answer the main research question

Putting Statistics in Context

- The vast majority of analyses can be done in a straightforward fashion - just remember and always use common sense as a guide - be skeptical!
- It is very easy to get “lost” in the statistical software and technical jargon, which differs markedly between different software packages. Terminology can also differ greatly between textbooks...
- Wikipedia is as good a resource as any for finding out about different statistical tests and terminology

Exploratory Data Analysis

- It is crucially important to explore your data fully before considering any “formal” statistical analyses

Exploratory Data Analysis

- It is crucially important to explore your data fully before considering any “formal” statistical analyses
- What explorations are done depends on the objective of the study - the research question(s)

Exploratory Data Analysis

- It is crucially important to explore your data fully before considering any “formal” statistical analyses
- What explorations are done depends on the objective of the study - the research question(s)
- Helps to decide what kind of formal statistical analyses might be most appropriate for the data available

Exploratory Data Analysis

- It is crucially important to explore your data fully before considering any “formal” statistical analyses
- What explorations are done depends on the objective of the study - the research question(s)
- Helps to decide what kind of formal statistical analyses might be most appropriate for the data available
- What a simple descriptive analysis **does not** provide is evidence of whether the observed treatment effects are large enough to be notable once sampling variation has been accounted - that is the role of formal analyses, e.g. hypothesis testing

Summary Statistics

Continuous (Integers / Numeric)

- Mean - a measure of location. Always examine the average value of the response variable(s) for the different “treatment” effects in your data
- Median - a robust single value summary of a set of data (50% quantile point) - most useful in highly skewed data or data with outliers
- Standard deviation (sd) - a measure of spread, how variable the data are
- Standard error of the mean (se) - an estimate of how far the sample mean is likely to be from the population mean
- and others: min, max, range, IQR, ...



```
mean(x) # mean
```

```
median(x) # median
```

```
sd(x) # standard deviation
```

```
min(x) # minimum
```

```
max(x) # maximum
```

```
range(x) # range
```

```
IQR(x) # interquartile range
```

Continuous Data Summaries

standard deviation

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

standard error

$$se = \frac{s}{\sqrt{n}}$$

Correlation coefficient

Combination of continuous and continuous

Correlation coefficient a measure association between two continuous variables (common but somewhat limited)

Pearson's correlation coefficient r

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

\bar{X} : mean of variable x

\bar{Y} : mean of variable y



```
# Test for Association/Correlation Between  
# Paired Samples  
cor.test(data$x, data$y, method = "pearson")  
cor.test(data$x, data$y, method = "spearman")  
  
# Scatterplot(s)  
pairs(data$x ~ data$y)  
pairs(data)
```

Summary Statistics

Continuous and ordinal variables



```
tapply(data$x.cont, data$y.fac, mean)
```

```
tapply(data$x.cont, data$y.fac, median)
```

```
tapply(data$x.cont, data$y.fac, sd)
```



- Median - a robust single value summary of a set of data (50% quantile point) - most useful in highly skewed data or data with outliers
- e.g. 10th and 90th percentile - a measure of spread, how variable the data are

```
quantile(x, probs = c(0.1, 0.9))
```

- proportions - e.g. percentage per grade

```
prop.table(table(data$x.fac))  
prop.table(table(data$x.fac, data$y.fac))
```



- proportions - percentage within the different categories
- contingency tables e. g. 2×2

```
table(data$x.fac)
table(data$x.fac, data$y.fac)
prop.table(table(data$x.fac))
prop.table(table(data$x.fac, data$y.fac))
```


Exercise 8



How to deal with missing values in R? (1/3)

- In R, missing values are represented by the symbol **NA** (not available).
- Impossible values (e. g., dividing by zero) are represented by the symbol **NaN** (not a number).
- Ask yourself why a **NA** and / or **NaN** occurs!

How to deal with missing values in R? (2/3)

- Testing for Missing Values

```
vec1 <- c(1, 2, 3, NA)
is.na(vec1) # returns a vector (FALSE, FALSE, FALSE, TRUE)
# The TRUE indicates the position of the NA in vec1.
```

- Recoding Values to Missing

```
# recode specific values (e. g. 0.001) to missing for variable x
# select rows where x is 0.001 and recode value in column x with NA
dat$x[dat$x == 0.001] <- NA
```

How to deal with missing values in R? (3/3)

- Excluding Missing Values from specific function calls

```
a <- c(1, 2, NA, 3)
mean(a) # returns NA
mean(a, na.rm=TRUE) # returns 2
```

- Check for complete cases with function `complete.cases(...)`

```
# list rows of data that have missing values
dat[!complete.cases(dat),]
subdat <- dat[complete.cases(dat),]
```

- Create new dataset without missing data with function `na.omit(...)`

```
new.dat <- na.omit(dat)
```

How to check your data for plausibility?

- Ask yourself what can go wrong?
- Implausible values?
- Impossible values?
- Logical errors?

Exercise 9A: Plausibility Checks



Exercise 9B: Missing Values



Exercise 10



Lecture Slides for Day 4

Basic Statistical Tests

Study data is collected for a purpose - to answer one or more specific scientific questions. The classical way to perform a formal statistical analyses of these data is to formulate these research questions into statistical **hypothesis tests**.

In this section we will go through a simple example in detail to highlight some of the important concepts - the general approach for more complex analyses is exactly same. *Note: the precise technical details are much less important than the concepts!*

Simple Example - One Population

After six weeks will the mean weight of a chicken be more than 250 grams?

There are 71 observations in `chickwts` from which to answer this question. This can be formulated into a statistical hypothesis test. A hypothesis test has two parts, the null hypothesis and the alternative hypothesis. This is typically written as follows:

$$H_0 : \mu \leq 250$$

$$H_A : \mu > 250$$

where μ is the mean weight in the **population** of chickens from which the sample of 71 chickens was drawn. Remember - we know the mean weight in the sample of chickens is greater than 250 it is the **population** of chickens which we are interested in.

Simple Example - One Population

After six weeks will the mean weight of a chicken be at least 250 grams?

$$H_0 : \mu \leq 250$$

$$H_A : \mu > 250$$

The null hypothesis (H_0) is the default situation, sometimes called the “state of nature”. In a treatment-control trial, H_0 is typically that the effect of the treatment is not different from the control. In this example our default position is that the mean weight of chickens is ≤ 250 . This is called a single-sided hypothesis test.



We now analyse the 71 observations to see whether there is evidence to **REJECT** the null hypothesis H_0 , and if the null hypothesis is rejected then we can conclude that the available evidence supports the alternative hypothesis.

```
t.test(chickwts$weight, mu = 250)
```

Note that hypothesis testing is concerned with finding evidence in support of the null hypothesis H_0 - the default situation - rather than evidence in favour of the alternative hypothesis.

One Sample t-test

For the chicken weights data an appropriate formal analyses is to use a **one-sample t-test**, why this test is appropriate will be discussed later. This analysis involves calculating a simple summary statistic - called a *t*-statistic - which we do entirely from the observed data.

$$T_{obs} = \frac{\bar{x} - \mu}{s/\sqrt{n}}$$

where \bar{x} is the sample mean, s the sample standard deviation and μ is the population mean in the null hypothesis which we wish to test for. We then look up the value of T_{obs} in a set of statistical tables/computer to see what the answer is to our research question.

Important concept - sampling

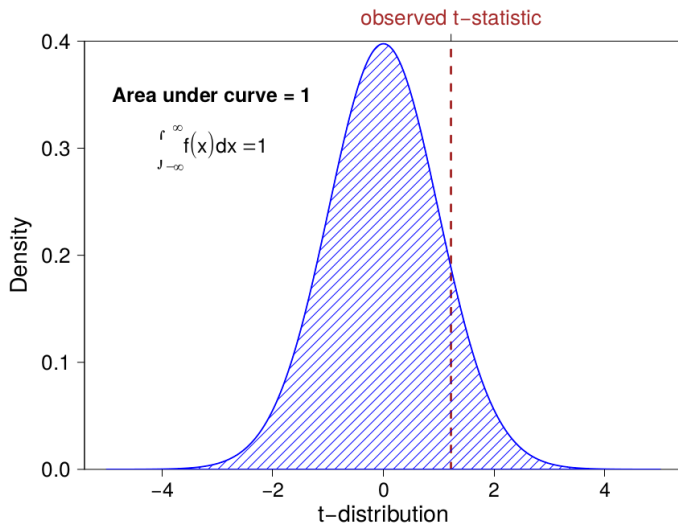
Why is $T_{obs} = \frac{\bar{x} - \mu}{s / \sqrt{(n)}}$ called a t -statistic?

If another sample of 71 chickens from the same population were weighed then the values for \bar{x} and s would be different, and hence the value for T_{obs} . If this was repeated many times and a histogram/Q-Q/P-P plot produced of the values for T_{obs} then this would follow the shape of a known distribution - **student-t probability distribution**. It is this piece of mathematics - knowing what the sampling distribution of T_{obs} is - which allows us to infer information about the population of chickens from which our original 71 chickens were sampled - without actually having to collect lots and lots of other samples of chickens! Mathematical theory is used to fill this data gap.

$$T_{obs} = \frac{261.31 - 250}{78.07 / \sqrt{71}} = 1.22$$

Put the values for the sample mean and standard deviation into the t-statistic formula along with the $\mu = 250$. We now look up the value of this in a t-distribution reference table. All this calculation will be done for you in R but it is important to understand the general process as this is the same for hypothesis testing in other more complex analyses.

One Sample, one-sided, t-test



Important concept - p -values

- The result of a hypothesis test is usually communicated in the form of a **p -value**
- The interpretation of a p -value is of crucial importance - it is the *probability that the test statistic takes values at least as extreme as that observed **assuming that H_0 is true***
- Exactly what **at least as extreme as** refers to depends on the alternative hypothesis H_A .
- This may sound rather abstract but it is usually obvious in practice

Simple Example - One Population

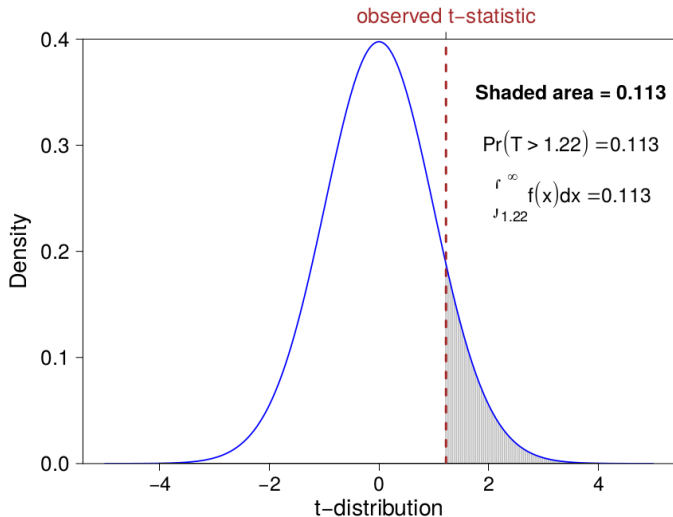
After six weeks will the mean weight of a chicken be at least 250 grams?

$$H_0 : \mu \leq 250$$

$$H_A : \mu > 250$$

The alternative hypothesis is $\mu > 250$ so in this test **at least as extreme as** in the definition of a p -value is the probability of observing a t -statistic which is > 1.22 **assuming that H_0 is true** - this is why 250 was used for μ when calculating T_{obs} .

One Sample, one-sided, t-test



Research Question

The purpose of this hypothesis test analysis is to answer a very specific scientific question:

After six weeks will the mean weight of a chicken be more than 250 grams?

So what is our answer?

The p -value for this hypothesis test is 0.113. Based on this value we can either **reject** H_0 and conclude that the mean weight of chickens in the population is likely to be greater than 250 grams or else we can **accept** H_0 where the mean chicken weight is less than 250 grams.

Research Question - be pragmatic with p -values

By convention a p -value of less than 0.05 is considered to provide reasonable evidence for rejecting H_0 . A p -value of between 0.05 and 0.1 might be considered as weak evidence against H_0 . Values of less than 0.01 are generally considered as very strong evidence for rejecting H_0 . It is **always** best to provide a p -value in any analyses to let the reviewer/client see the strength of evidence rather than simply claiming statistically significant findings!

Communicating Results of Hypothesis Tests

Transparency is essential - the devil can be in the detail - which at the very least should comprise:

- i. what hypothesis was being tested - be clear and precise
- ii. what statistical test was used
- iii. what the p -value is
- iv. what the treatment effect is (more later).

This is particularly crucial if the analyses are to be given to someone *e/se* to then make a judgment on the scientific significance.

Two-sided Tests: One Population

After six weeks will the mean weight of a chicken be equal to 250 grams?

This is now a two sided hypothesis test:

$$H_0 : \mu = 250$$

$$H_A : \mu \neq 250$$

This time the hypothesis test is asking how much evidence is there in our sample data to conclude that in the population of all chickens the mean weight is not equal to 250 grams.

Two-sided Tests: One Population



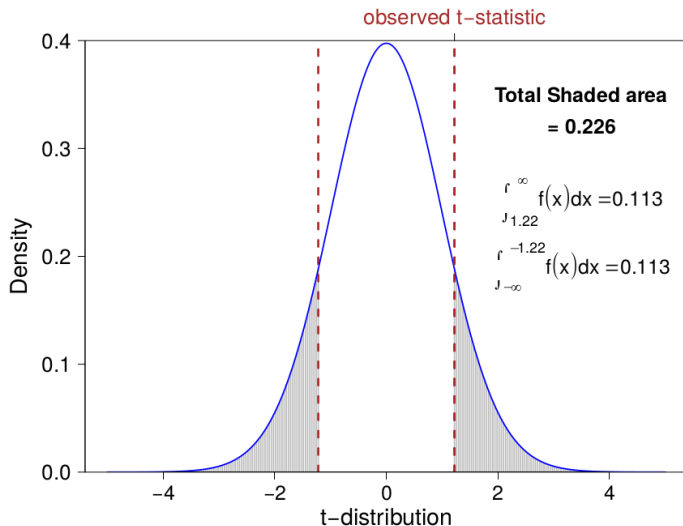
```
t.test(chickwts$weight, mu = 250)

##
##  One Sample t-test
##
## data:  chickwts$weight
## t = 1.2206, df = 70, p-value = 0.2263
## alternative hypothesis: true mean is not equal to 250
## 95 percent confidence interval:
##  242.8301 279.7896
## sample estimates:
## mean of x
##  261.3099
```

Two-sided Tests

A two-sided test is similar to a one-sided test - the key difference is in what is now defined as **at least as extreme** in the definition of the p -value. This time the alternative hypothesis refers to observing a value of **either** $\bar{x} > 250$ or $\bar{x} < -250$ **assuming that H_0 is true**, which using the t-test approach is equivalent to the probability of observing $T_{obs} > 1.22$ or $T_{obs} < -1.22$ which we can again look up in reference tables.

One Sample, two-sided, t-test



Two-sided Tests

- The two-sided t-test has a p -value which is exactly double the single sided test
- Think! - intuitively the p -value should be less for a single sided test as the research question you are asking is much narrower e.g. greater than 250 grams, as opposed to whether the mean chicken weight might be **either** less than 250 grams **or greater** than 250 grams.

→ You are using the same amount of information (71 observations) to answer a narrower research question and so all else being equal you should expect a “more powerful” analyses (e.g. a lower p -value all else being equal)

Exercise 11



Chi-square Test

There are two very commonly used statistical tests for testing dependence between two categorical variables: Chi-squared test & Fisher's exact test.

The Chi-square Distribution

Exact Fisher Test: Permutation Test

Exercise 12



Plotting in R





- Continuous data
 - Histogram
 - Boxplot
- Nominal / Ordinal data
 - Barplot
 - Mosaicplot
 - Scatterplots

Exercise 13A and 13B



Exercise 14

