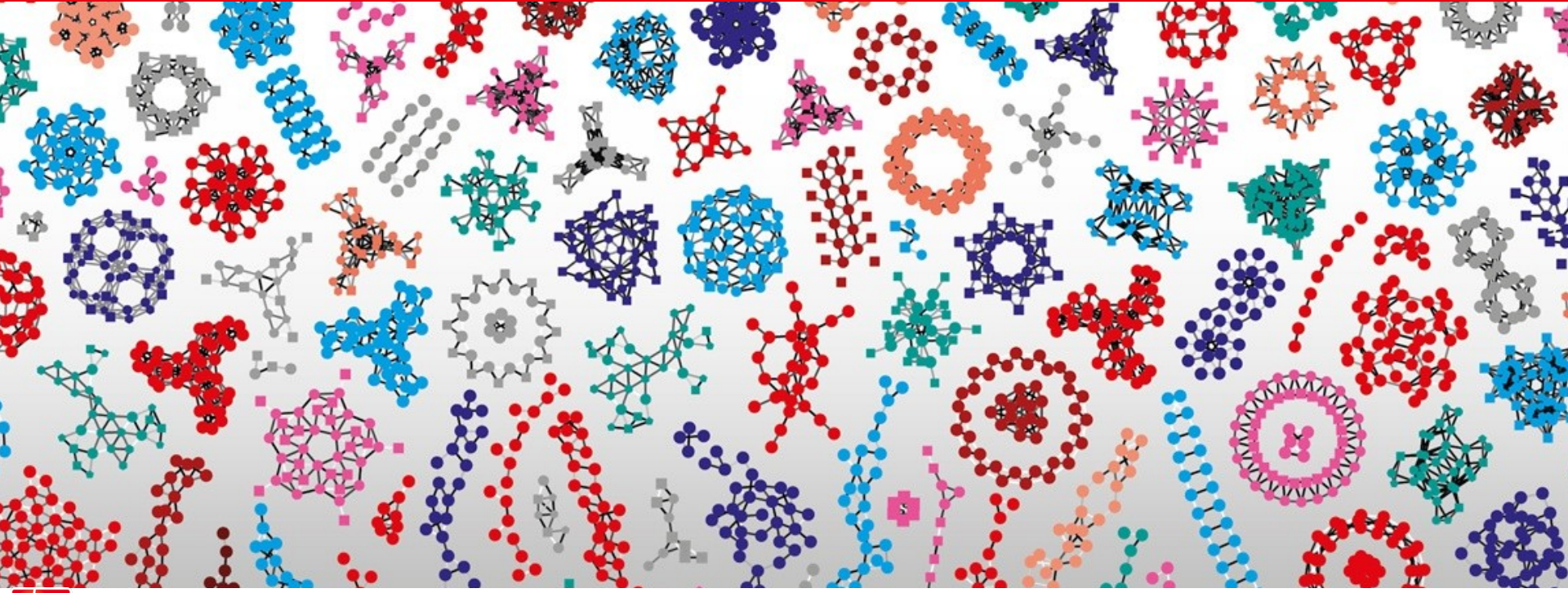


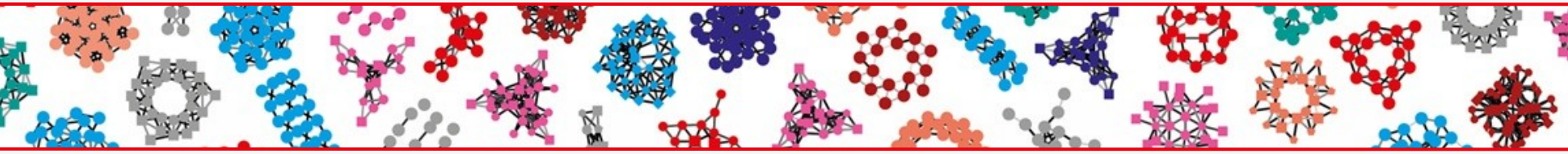
SIB
Swiss Institute of Bioinformatics

First steps with R in Life Sciences: Statistics

February, 2022 - Streamed

Wandrille Duchemin

-- with slides from Diana Marek, Wandrille Duchemin, Leonore Wigger



07 Starting with statistics in R

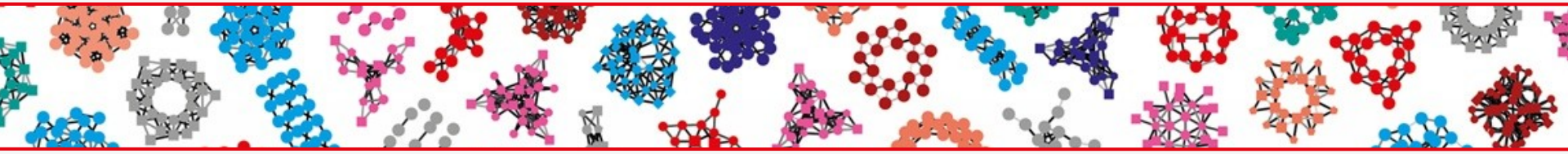
Getting started

From the data you have collected, you will have one or more question(s) (hypotheses to test).

- Make informative pictures to reveal relationships between variables.
- Decide on statistical model.
- Assess the assumptions underlying your modelling before final decision (results might be unreliable if assumptions are violated).
- Translate statistical model into R language, run statistical test

Covered in this lecture:

- T-test
- Correlation
- Simple linear regression



Hypothesis testing and linear modelling in R

Statistical hypothesis testing

- Two hypotheses in competition :
  - H0: the NULL hypothesis (usually the most conservative)
  - H1: the alternative hypothesis (usually the one we are actually interested in)

Example:

H0: « There is no difference in weight between two given strains of mice »

H1: « The average weight in KO mice is different from that in WT mice »

Statistical test:

- Calculate test statistic,
- Calculate associated p-value,
- Check if p-value is small enough to reject H0, according to pre-defined significance level.

Statistical hypothesis testing

- **Test statistic**:

  Variable calculated from sample data. Measures the degree of agreement between the sample of data and the null hypothesis. Example: t statistic in the t-test.

- **p-value**:

  Probability of observing a result (and test statistic) at least as extreme as the one obtained from the analyzed data, assuming the null hypothesis is true.

- **significance level (alpha level)**:

  Decision threshold for the p-value below which we reject the null hypothesis (conventionally, 0.05 or 0.01). It is also the probability of mistakenly rejecting the null hypothesis.

Being careful with interpretations

1. statistical significance does NOT imply biological significance,
biological significance does NOT imply statistical significance.

2. The absence of a statistically significant difference
does NOT imply a an *equivalence (i.e. that groups are the same)*.

T-test

Goal:

- **Compare a continuous measure between two groups: Is the difference between the two group means statistically significant?**

Assumptions:

- Observations are independent
- ~~Values are normally distributed within in each group~~
    - Values's **means** are normally distributed within each group
- ~~(Same variance in each group)~~

  R uses Welch's t-test, which does not assume equal variance

Example data set: sleep

Student's sleep data: shows the effect of two soporific drugs on 10 patients:
hours of sleep gained with the drug compared to control condition without drug

```
>data(sleep)
>head(sleep)
 extra group ID
1   0.7     1  1
2  -1.6     1  2
3  -0.2     1  3
4  -1.2     1  4
5  -0.1     1  5
6   3.4     1  6

>summary(sleep)
extra        group       ID
 Min.   :-1.600   1:10   1      :2
 1st Qu.:-0.025   2:10   2      :2
 Median : 0.950          3      :2
 Mean   : 1.540          4      :2
 3rd Qu.: 3.400          5      :2
 Max.   : 5.500          6      :2
                       (Other):8
```

Cushny, A. R. and Peebles, A. R. (1905) The action of optical isomers: II hyoscines. The Journal of Physiology 32, 501–510.

Student (1908) The probable error of the mean. Biometrika, 6, 20.

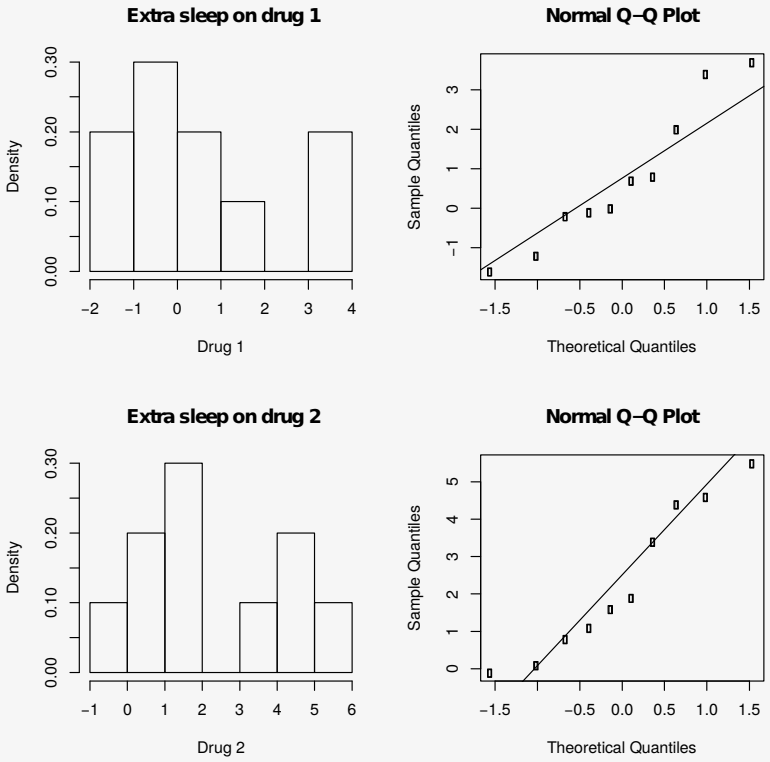
Check normality of the data with plots

>data(sleep) # Data which shows the effect of two soporific drugs (increase in hours of sleep compared to control) on patients.

Using histograms (hist()) and QQ-Plots (qqnorm(), qqline()), we can visually assess the normality of the data.

```
>par(mfrow=c(2,2))
>hist(sleep$extra[sleep$group==1],
freq=FALSE, xlab="Drug 1",
main=" Extra sleep on drug 1")
>qqnorm(sleep$extra[sleep$group==1])
>qqline(sleep$extra[sleep$group==1])
```

```
>hist(sleep$extra[sleep$group==2],
freq=FALSE, >xlab="Drug 2",
main=" Extra sleep on drug 2")
>qqnorm(sleep$extra[sleep$group==2])
>qqline(sleep$extra[sleep$group==2])
```



Recommendations for assessing normality

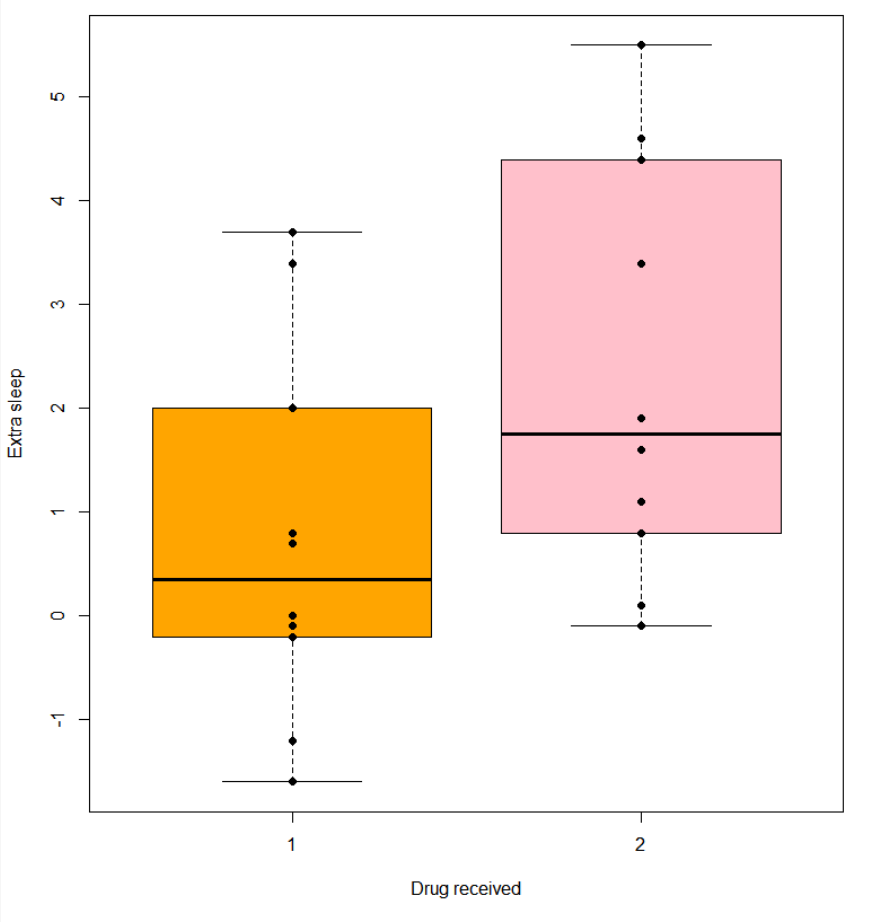
T-test is somewhat robust to non-normal data. No need to be too strict about normality requirement.

QQ-Plot: If you only do one type of assessment, use this!

Histograms: Better for larger data sets. Distributions hard to asses for small data sets.

Visualize group differences with boxplot()

```
>boxplot(extra ~ group, data=sleep, col=c("orange", "pink"),
ylab="Extra sleep", xlab="Drug received")
>points(extra ~ group, data = sleep, col="black",pch = 19)
```



Are the two means significantly different?

Function t.test()

```
>t.test(sleep$extra[sleep$group==1],
        sleep$extra[sleep$group==2])
>t.test(extra ~ group, data=sleep) #equivalent to the above
```

Two-sided **Welch two-sample t-test** (modified t-test, does not assume equal sample variances)

Welch Two Sample t-test

data: extra by group
t = -1.8608, df = 17.776, p-value = **0.07939**
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832 0.2054832
sample estimates:
mean in group 1 mean in group 2
           0.75   2.33

No significant difference between group means at alpha level 0.05

T-test object

- t.test() and other tests return an R object that can be assigned to a variable. This object is a list.
- View the names of the list's slots using names().
- Access the elements of a list using the $ or the [[ ]] operators.

```
> test.res <- t.test(sleep$extra[sleep$group==1],
                     sleep$extra[sleep$group==2])

> names(test.res)
[1] "statistic"   "parameter"   "p.value"     "conf.int"
[5] "estimate"    "null.value"  "alternative" "method"
[9] "data.name"

> test.res[["statistic"]]   #or: test.res$statistic
 t
-1.860813

> test.res[["p.value"]]     #or: test.res$p.value
[1] 0.07939
```

Paired data

Sleep data set has two measurements per person (ID): one for each drug.

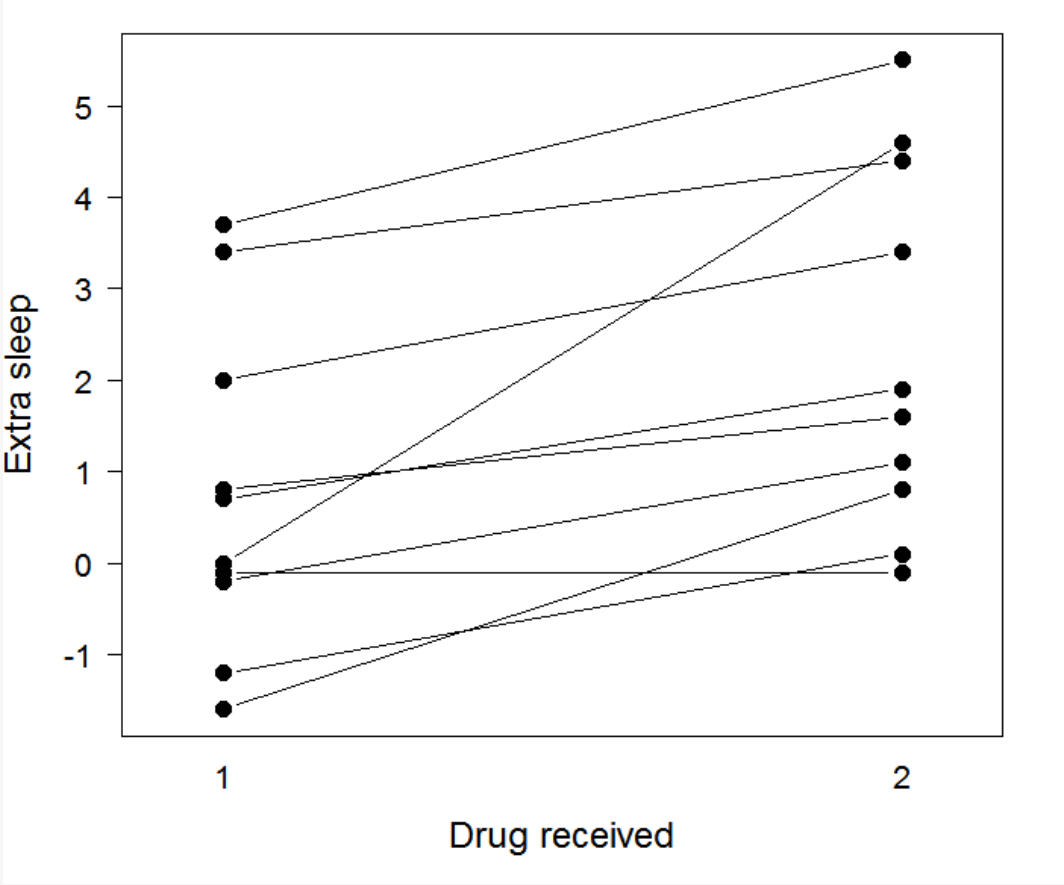
A paired t-test would be more appropriate than an unpaired t-test.

Normality assumption:

- The mean of the **differences** between pairs are normally distributed.

Paired data representation

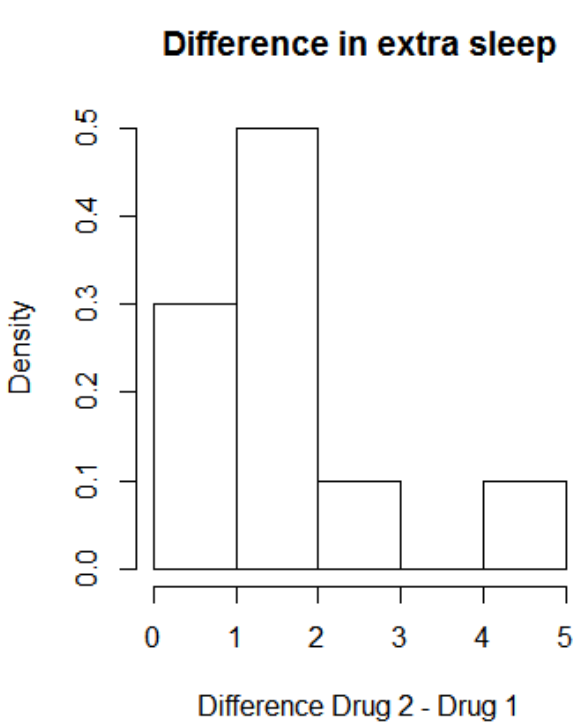
```
>interaction.plot(response=sleep$extra, x.factor=sleep$group,
 trace.factor=sleep$ID, legend=FALSE, type="b", lty=1, pch=16,
 xlab="Drug received", ylab="Extra sleep")
```

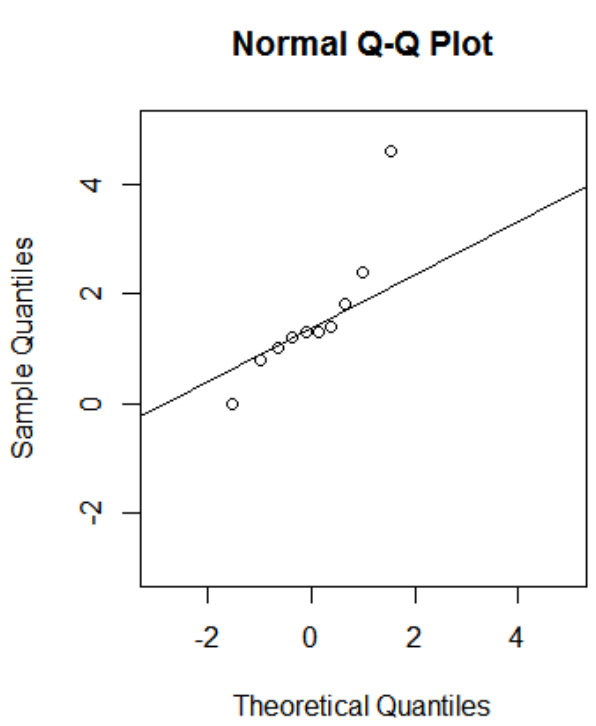


Is the difference between the two treatments significant?

Check normality of the differences between pairs

```
>difference = sleep$extra[sleep$group==1]-sleep$extra[sleep$group==2]

>hist(difference, freq=FALSE, xlab="Difference Drug 2 - Drug 1",
main="Difference in extra sleep", col="white")

>qqnorm(difference, xlim=c(-5, 2), ylim=c(-5, 2))
>qqline(difference)
```





Most points are close to the qqline but there is an outlier

Paired t-test - P5S

- Use a paired t-test when the data contains two measures for the same subject/entity.

```
>t.test(sleep$extra[sleep$group==1],
        sleep$extra[sleep$group==2], paired=TRUE)
# paired values must be at the same position in the two vectors
# do not use formula notation (extra~sleep) for paired t-test
```

```
Paired t-test

data:  sleep$extra[sleep$group == 1] and
sleep$extra[sleep$group == 2]
t = -4.0621, df = 9, p-value = 0.002833
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 -2.4598858 -0.7001142
sample estimates:
mean of the differences
                  -1.58
```

The difference between the two treatments is significant at alpha level 0.05

Non-parametric alternatives to the t-test

- When the data deviates strongly from normality, a non-parametric test can be used in place of a t-test.
- Non-parametric tests do not assume any particular distribution of the data.

Instead of t-test (without pairing), use Mann-Whitney U test.

```
>wilcox.test(sleep$extra[sleep$group==1],
      sleep$extra[sleep$group==2])

>wilcox.test(extra~group, data=sleep)  # equivalent
```

Instead of paired t-test, use Wilcoxon Signed Rank test.

```
>wilcox.test(sleep$extra[sleep$group==1],
      sleep$extra[sleep$group==2], paired=TRUE)
```

These two tests have different names but are both implemented in the R function wilcox.test.

Function wilcox.test()

For the sleep data, a paired test is appropriate.

```
>wilcox.test(sleep$extra[sleep$group==1],
        sleep$extra[sleep$group==2], paired=TRUE)
```

Wilcoxon signed rank test with continuity correction

data: sleep$extra[sleep$group == 1] and sleep$extra[sleep$group == 2]
V = 0, p-value = **0.009091**
alternative hypothesis: true location shift is not equal to 0

- The conclusion is the same as it was for the paired t-test.
- The p-value is a little higher wilcox.test: 0.009091
- t.test: 0.002833

The difference between the two treatments is significant at alpha level 0.05

Function wilcox.test(): warning messages about p-value computation

- wilcox.test() implements two ways to compute p-values: exact and by approximation
- The method can be selected with parameter **exact=TRUE** or **exact=FALSE**
- The default is "exact" if sample size < 50 *and* there are no ties in the data. Otherwise it is by normal approximation.

If warning messages saying "cannot compute exact p-value" are displayed, then computation of exact p-value failed and a normal approximation was performed.

Function wilcox.test(): warning messages about p-value computation

- wilcox.test() implements two ways to compute p-values: exact and by approximation
- The method can be selected with parameter **exact=TRUE** or **exact=FALSE**
- The default is "exact" if sample size < 50 *and* there are no ties in the data. Otherwise it is by normal approximation.

If warning messages saying "cannot compute exact p-value" are displayed, then computation of exact p-value failed and a normal approximation was performed.

```
Warning messages:
1: In wilcox.test.default(sleep$extra[sleep$group == 1], sleep$extra[sleep$group ==  :
  cannot compute exact p-value with ties
2: In wilcox.test.default(sleep$extra[sleep$group == 1], sleep$extra[sleep$group ==  :
  cannot compute exact p-value with zeroes
```

These warnings don't mean that there is an error in the result. An (approximated) p-value is still provided and can be reported.

Let’s practice - 10

Come back to the mice data-set stored in the "mice_data" data frame.

1) Considering WT mice weight and KO mice weight separately, check the assumption of normality graphically.

2) Make an appropriate plot to visualize the mouse weights grouped by genotype.

3) Perform a test to see whether the mouse weight is different between the two genotypes.

*4) Repeat step **1** to **3** for the diet variable.*

In a nutshell

- R can help you to make a graphical representation of your hypothesis and to test it using the right model based on your data (check the assumptions).
- R offers a wide range of functions for simple hypotheses testing such as:
  - t.test(): Student's t-test
  - wilcox.test(): Whitney Mann U and Wilcoxon Signed Rank tests (non-parametric)

  Further examples not covered in this course:
  - var.test() : F test for equality of variances
  - fisher.test() : Fisher's exact test
  - chisq.test() : Chi-squared contingency tables tests and goodness-of-fit tests
  - ks.test() : Kolmogorov-Smirnov test (non parametric)
  - ...

Bivariate linear correlation

- **Goal: Quantify the strength of a linear correlation between two continuous variables**

- cor() computes a correlation between two variables.

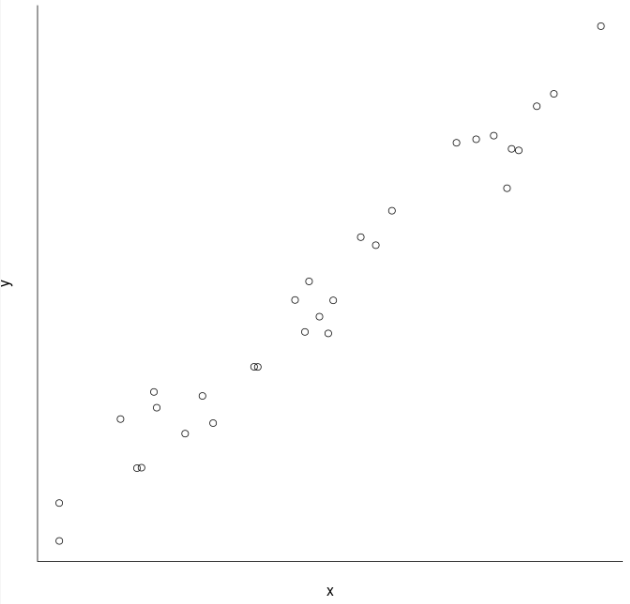
  Default: method="pearson" (linear correlation)

  Other options: method="spearman", method="kendall"

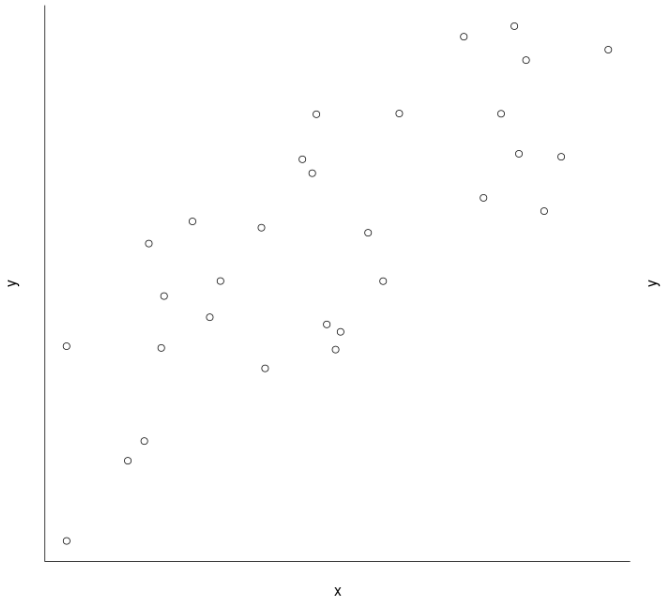
  (rank-based correlations)

- cor.test() computes a correlation and performs a corresponding statistical test to obtain a p-value (for Pearson correlation: p-value from linear regression, same as lm())

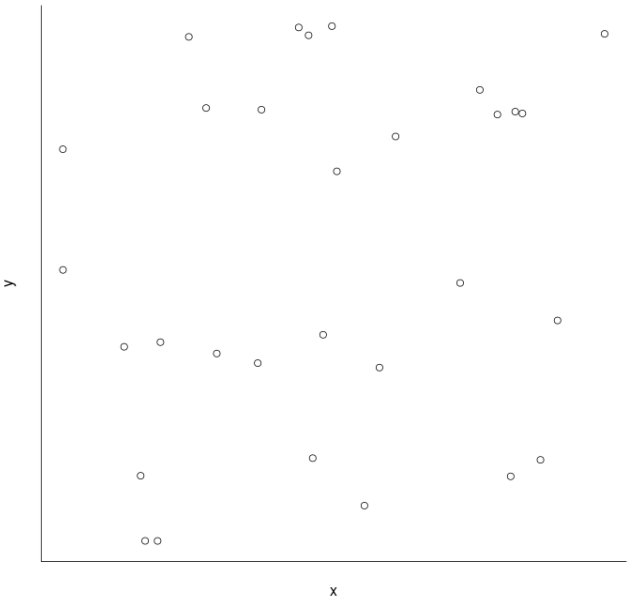
Scatter plots and correlation strength



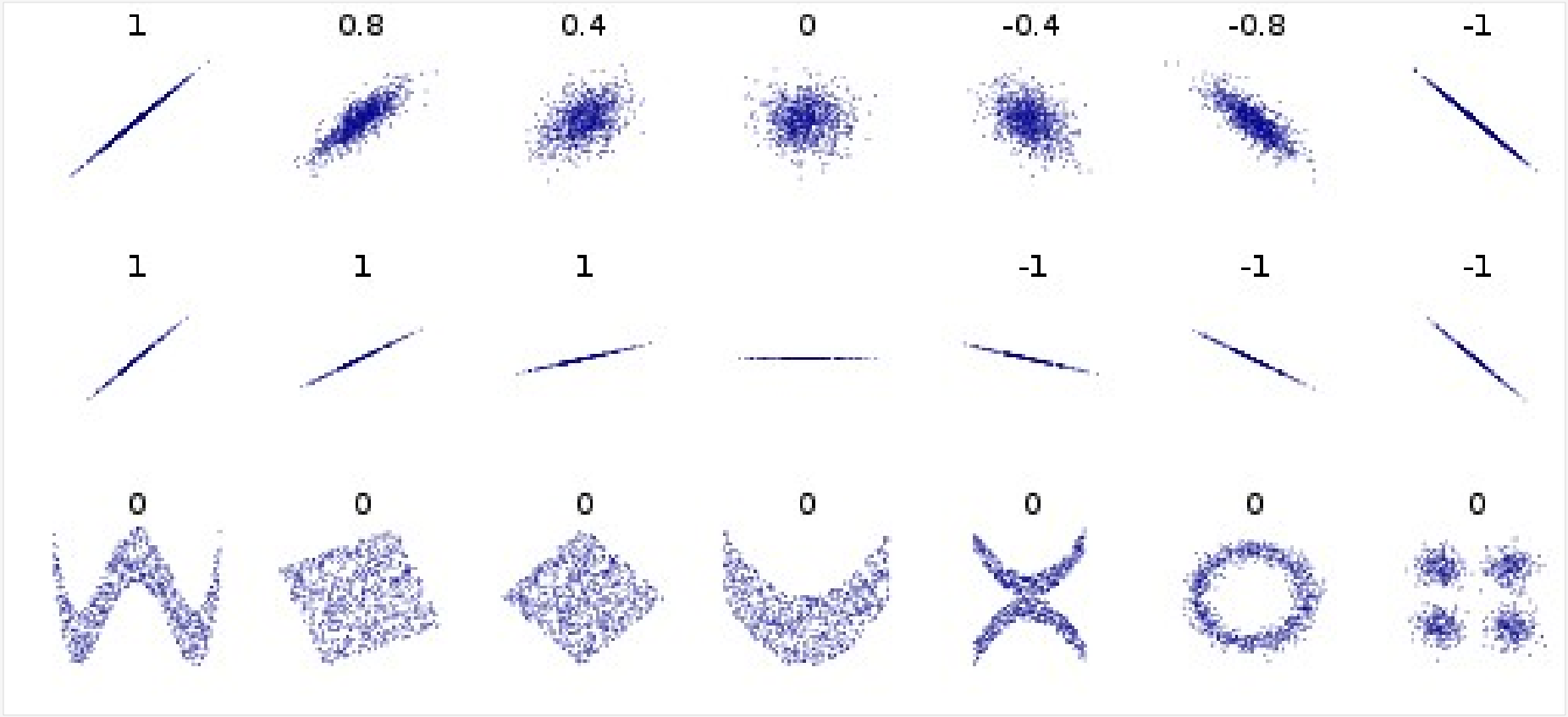
Strong linear correlation: points are close to a straight line



Medium-strong linear correlation: points more or less follow a straight line



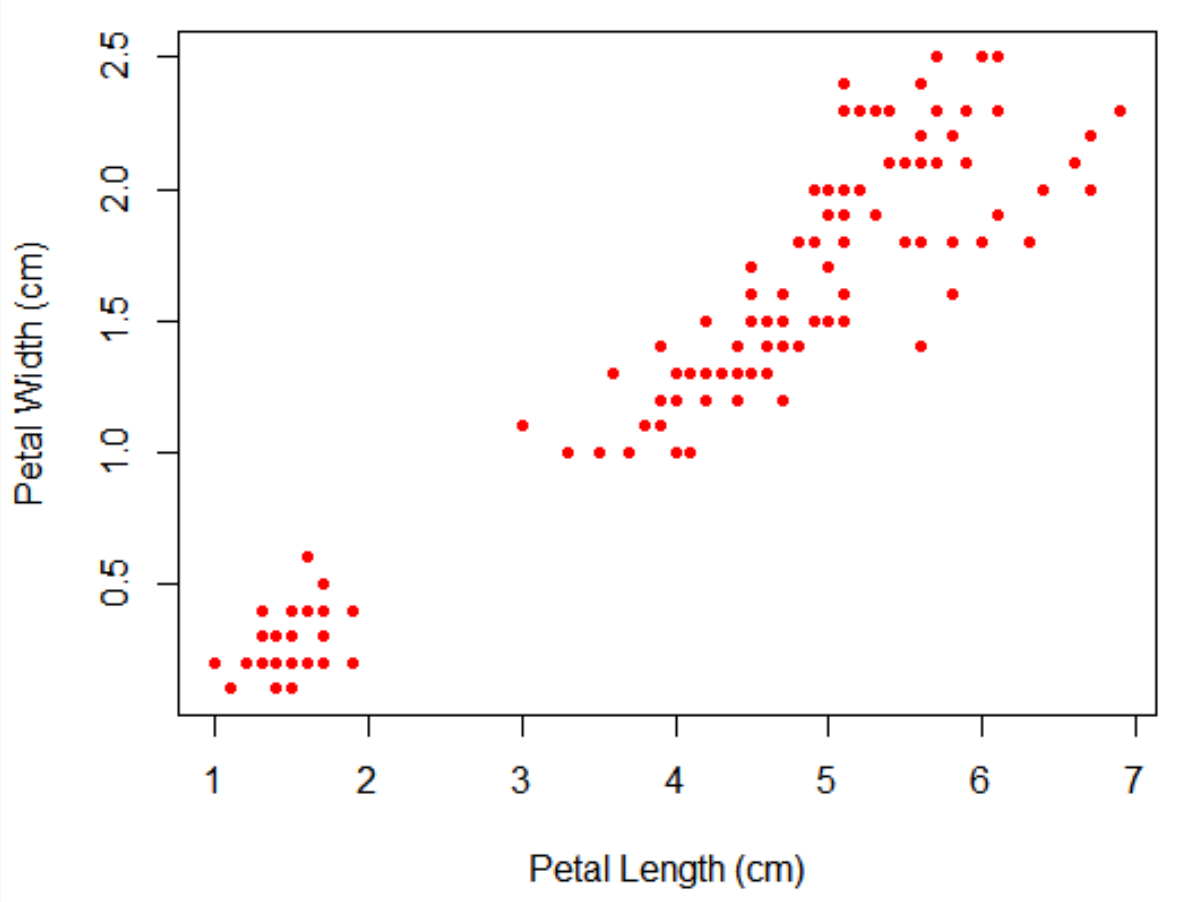
No correlation: Points have random pattern



Several sets of (x, y) points, with the Pearson correlation coefficient of x and y for each set. Note that the correlation reflects the noisiness and direction of a linear relationship (top row), but not the slope of that relationship (middle), nor many aspects of nonlinear relationships (bottom).

Scatter plot

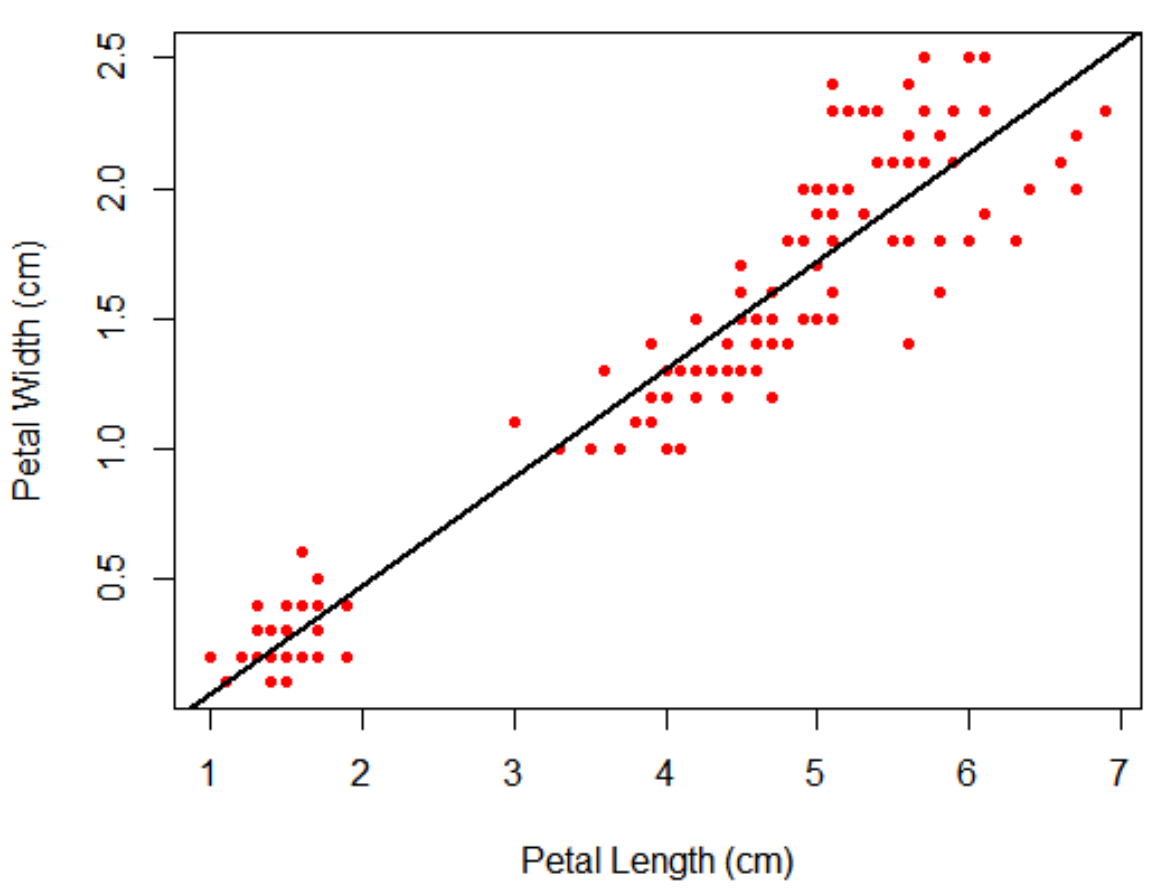
```
> plot(iris$Petal.Length, iris$Petal.Width,
       col="red", pch=20,
       xlab="Petal Length", ylab="Petal Width")
```



Does a significant linear correlation exist between sepal length and width?

Scatter plot

```
> plot(iris$Petal.Length, iris$Petal.Width,
      col="red", pch=20,
      xlab="Petal Length", ylab="Petal Width")
> abline(lm(iris$Petal.Width~iris$Petal.Length),
      col="black", lwd=2)
```



Does a significant linear correlation exist between sepal length and width?

Visual assessment: Points are close to trend line

Linear correlation

```
>cor(iris$Petal.Length, iris$Petal.Width, method="pearson")
[1] 0.9628654

>cor.test(iris$Petal.Length, iris$Petal.Width,
method="pearson")

        Pearson's product-moment correlation

data:  iris$Petal.Length and iris$Petal.Width
t = 43.387, df = 148, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9490525 0.9729853
sample estimates:
      cor
0.9628654
```

We can reject the null hypothesis that there is no association

Linear regression

Goal: Determine the extent to which there is a linear relationship between an "outcome" variable (dependent variable) and one more "explanatory" variables (independent variables, predictor variables).

Can a significant part of the variability in the outcome be predicted/explained by the independent variables?

Outcome variable: continuous (e.g. weight, heart rate, blood sugar)

Explanatory variables: continuous OR categorical (e.g. gender)

In R, the linear regression model is specified by a model formula of the form:

outcome ~ explanatory variables

Simple linear regression

- A simple regression model (one explanatory variable) is specified by
  - y = a + b*x+ err

a: Intercept

b: coefficient of explanatory var., x: explanatory var.

err: error term (=residuals)

Assumptions :

- Homoscedasticity : independence between residual variance and variables
- Linearity + absence of linear relationship between variables
- independence of the observations.
- Residuals centered around predicted value (mean=0)
- + normality of the **residual's mean**
  - → only used to assess parameters confidence interval

- *Otherwise : try log-transform (for heteroskedasticity) or non-parametric methods if the assumptions are not met.*

Summary of the data

```
>class_data <- read.csv("class.csv")
#dataset* of 19 students' measurements
>summary(class_data)
```

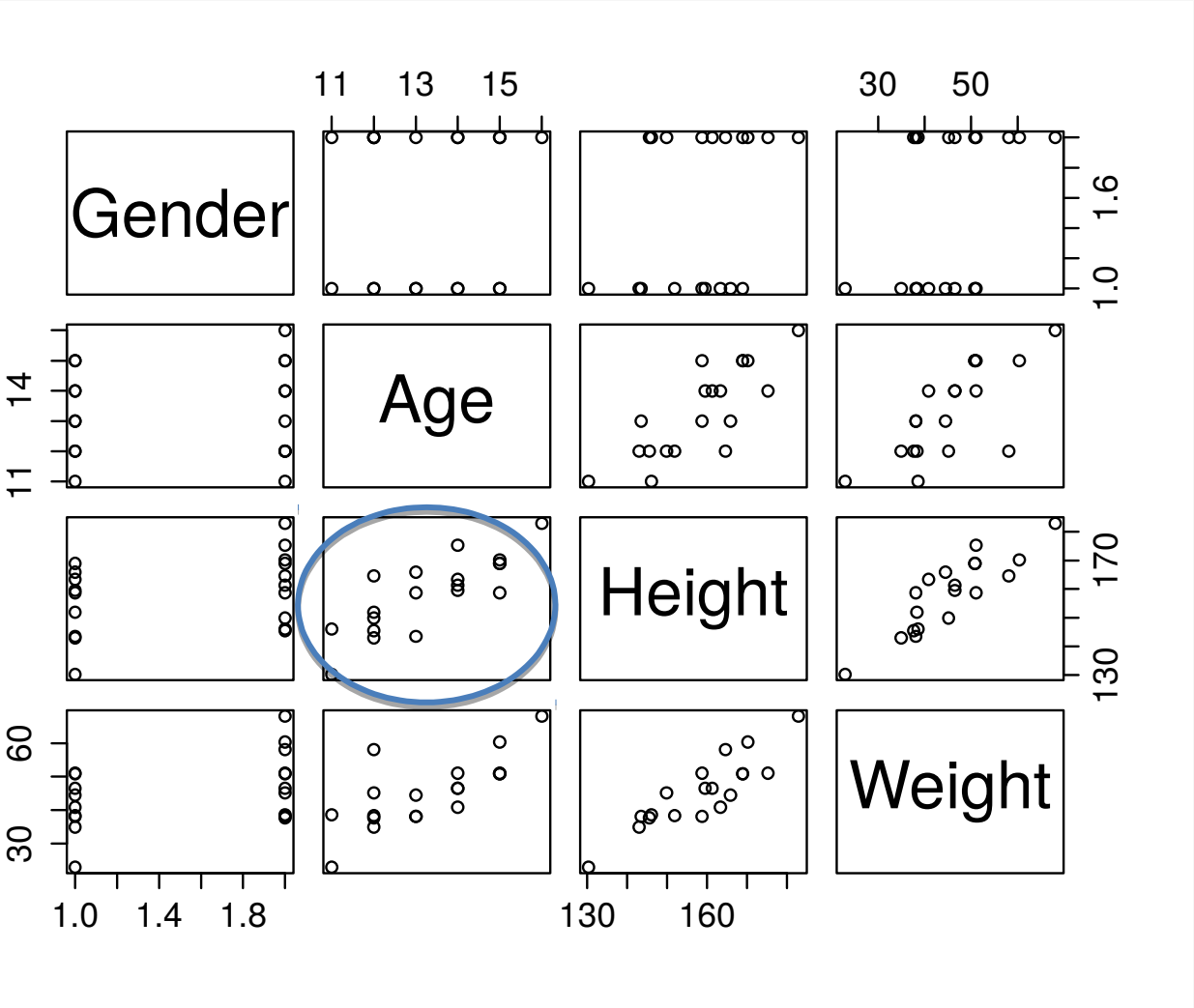
```
   Gender       Age             Height          Weight     
    F: 9   Min.   :11.00   Min.   :130.3   Min.   :22.91  
    M:10   1st Qu.:12.00   1st Qu.:148.0   1st Qu.:38.22  
           Median :13.00   Median :159.5   Median :45.13  
           Mean   :13.32   Mean   :158.3   Mean   :45.37  
           3rd Qu.:14.50   3rd Qu.:167.4   3rd Qu.:50.92  
           Max.   :16.00   Max.   :182.9   Max.   :68.04  
```

*CLASS dataset, from the program SAS (names removed and units have been modified from imperial to metric)

Representation of the data

`>pairs(class_data)`

Height~a+b1*Age+ err



The lm() Function

- **lm()** : fitting a linear model.
- Creates an R object which contains the regression result and can be stored or printed. Just printing the result provides only the regression coefficients.
- The **summary()** and **plot()** functions can be used to provide more information, including diagnostic plots.

- Many other functions can be applied to the regression objects:
  - **residuals()** extracts a vector containing the residuals (error)
  - **coef()** extracts the regression coefficients
  - **anova()** produces the corresponding ANOVA table (not covered

Simple linear regression

```
> model_height_age <- lm(Height~Age, data=class_data)
> model_height_age

Call:
lm(formula = Height ~ Age, data = class_data)

Coefficients:
(Intercept)          Age
     64.069        7.079
```

Simple linear regression

```
> model_height_age<-lm(Height~Age, data=class_data)
> model_height_age
```

Height~a+b1*Age+ err

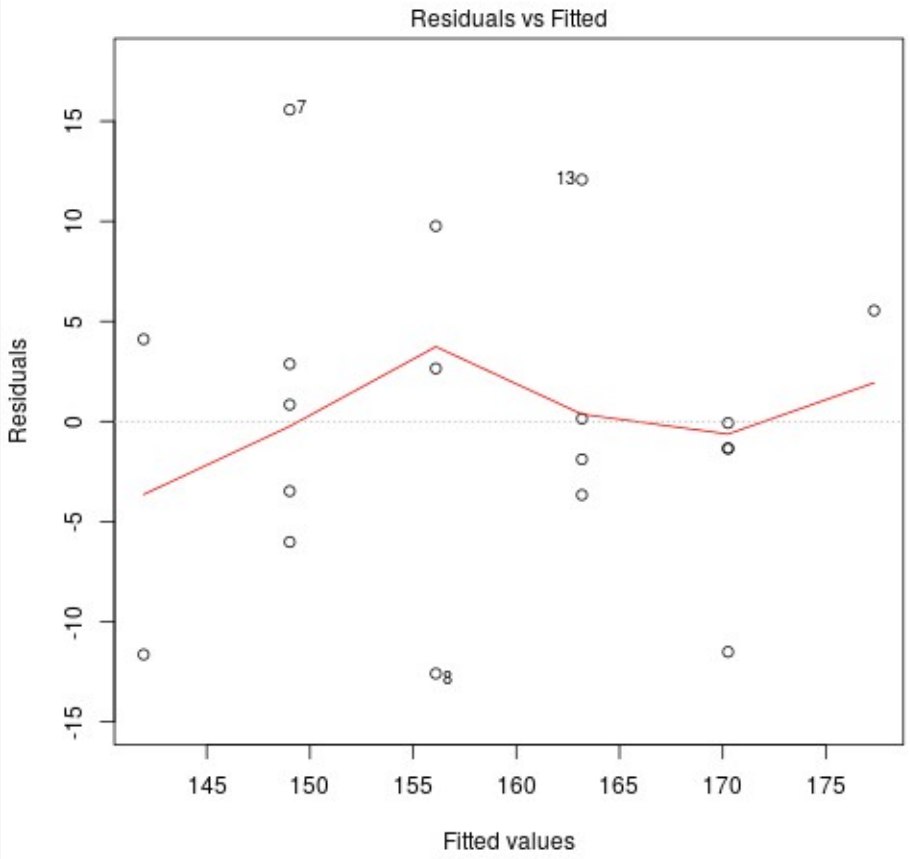
```
Call:
lm(formula = Height ~ Age, data = class_data)

Coefficients:
(Intercept)          Age
     64.069        7.079
```

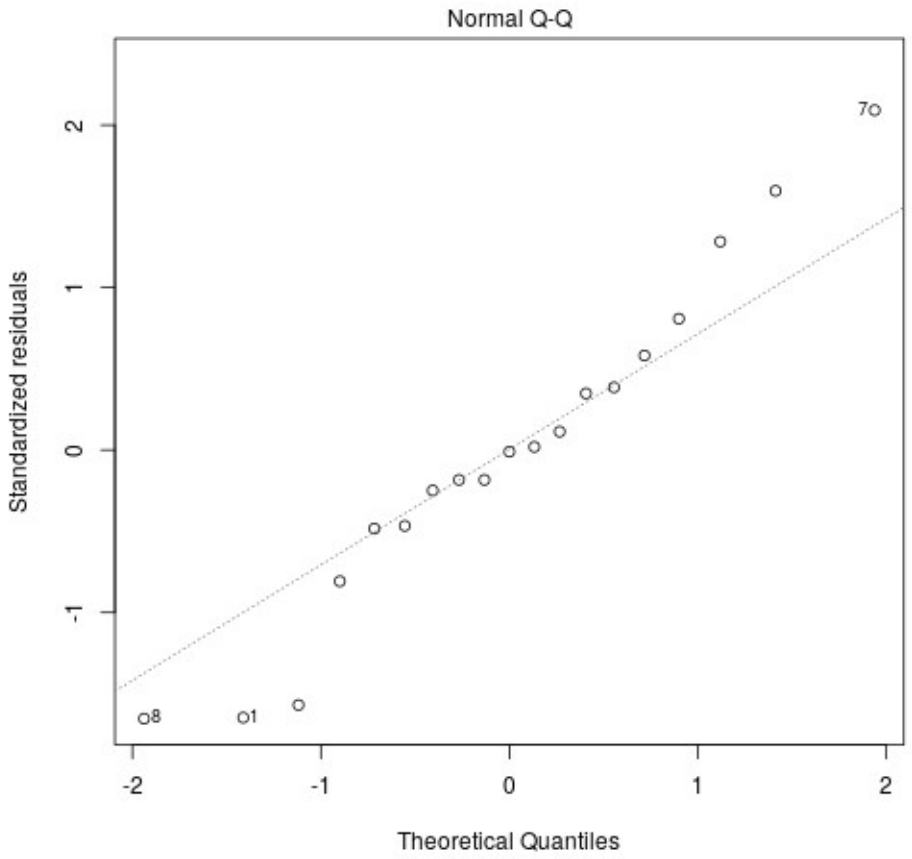
Model: Height = 64.07 + 7.08 x Age

Check model assumptions

The output of lm() already contains some diagnostic plots:

```
> par(mfrow=c(1,2))
> plot(model_height_age, which=1)
> plot(model_height_age, which=2)
```

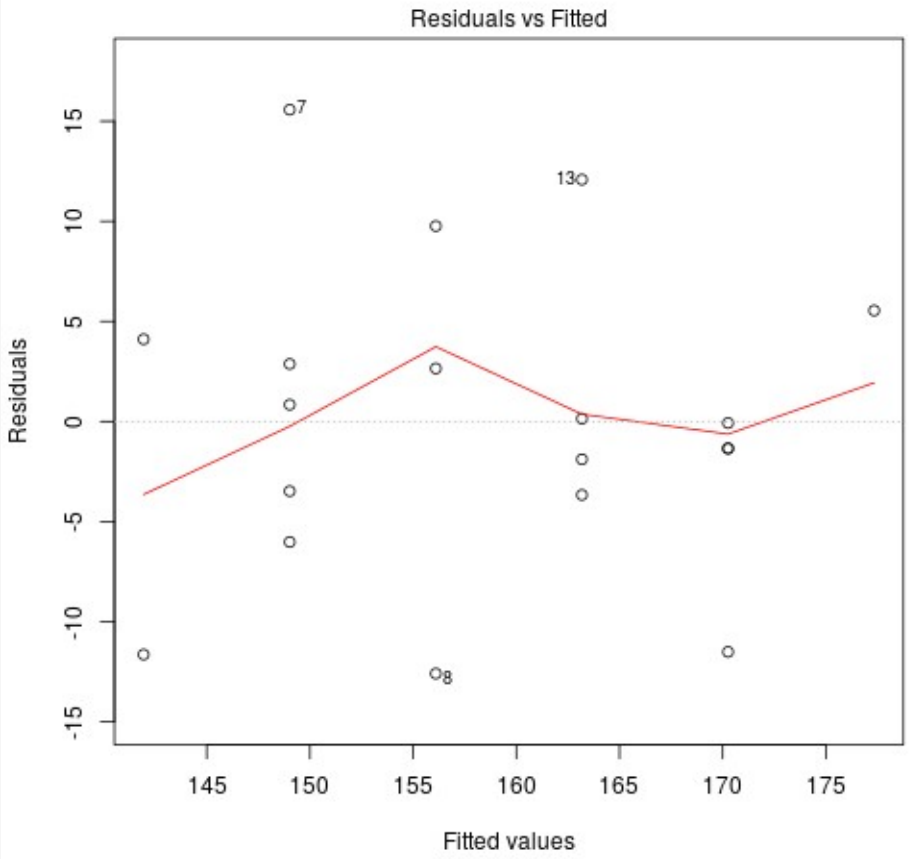


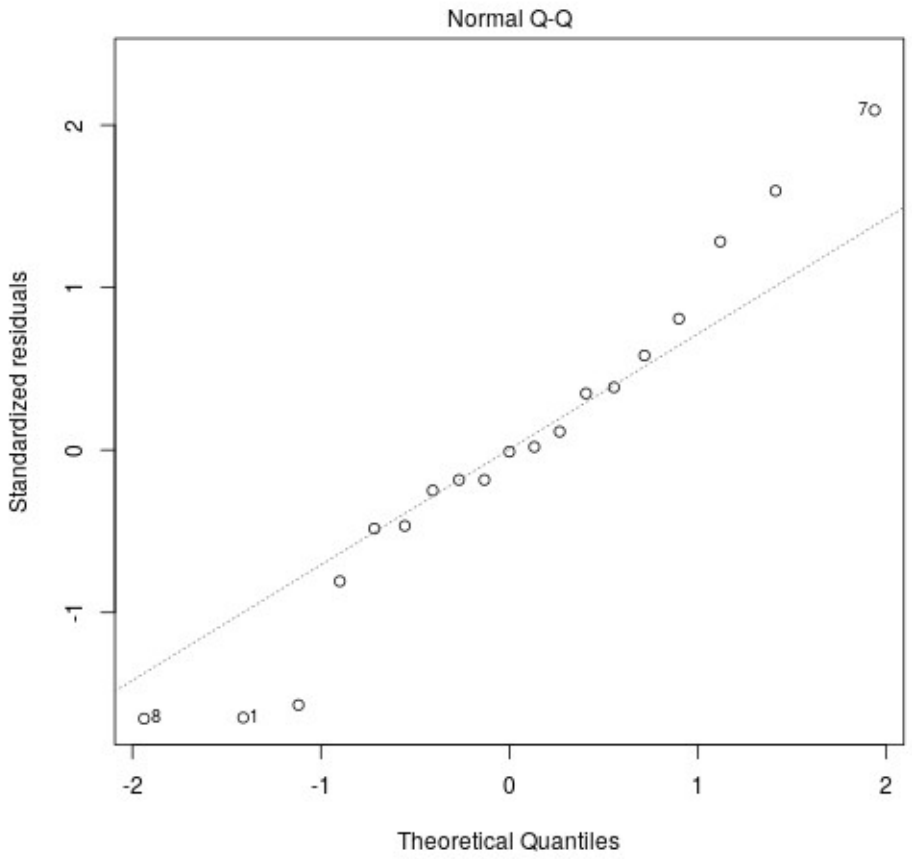


Check model assumptions

Left plot : homoscedasticity (variance or residual equal along axis)
+ mean of residuals at 0

Right plot : normality of residuals

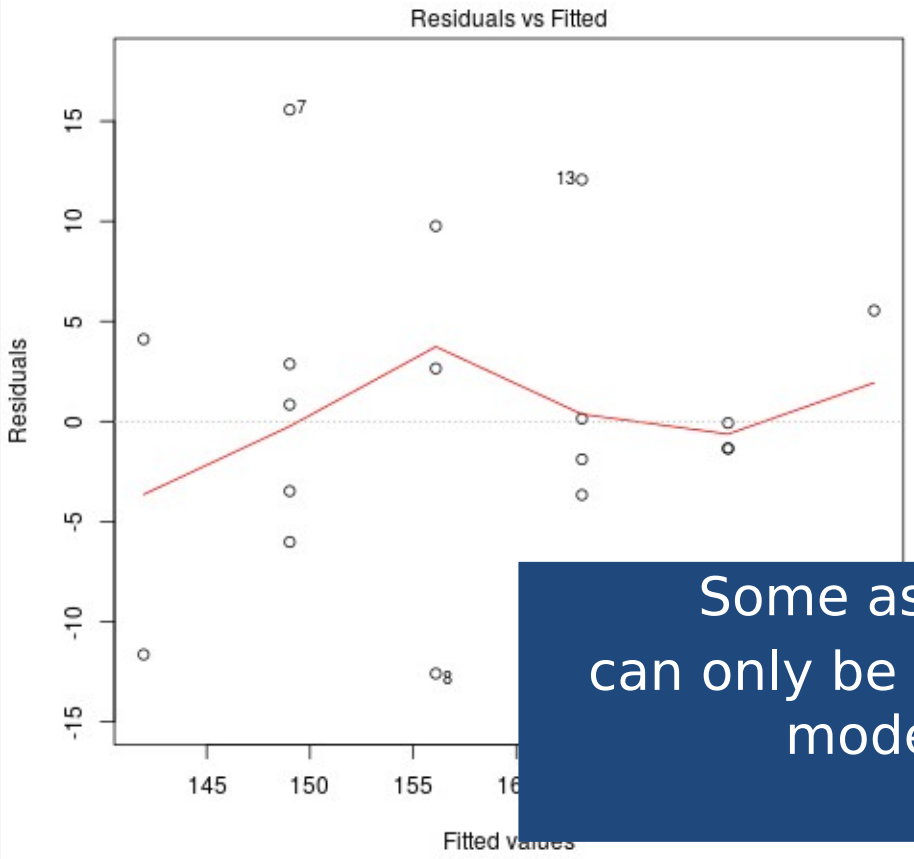


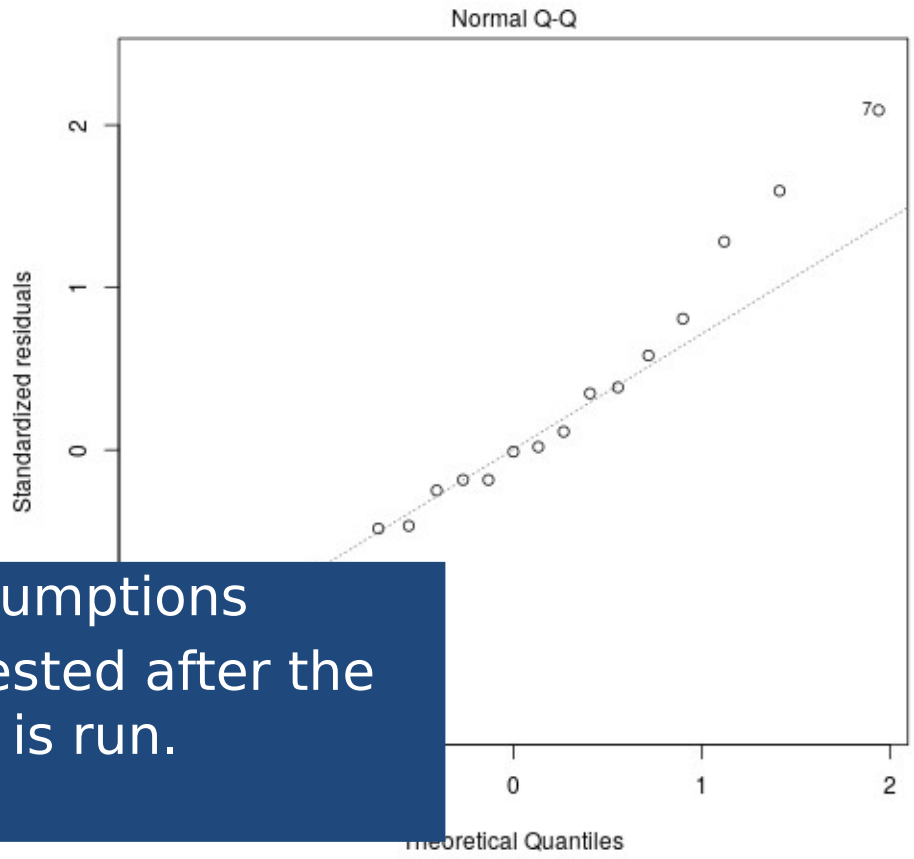


Check model assumptions

Left plot : homoscedasticity (variance or residual equal along axis)
+ mean of residuals at 0

Right plot : normality of residuals

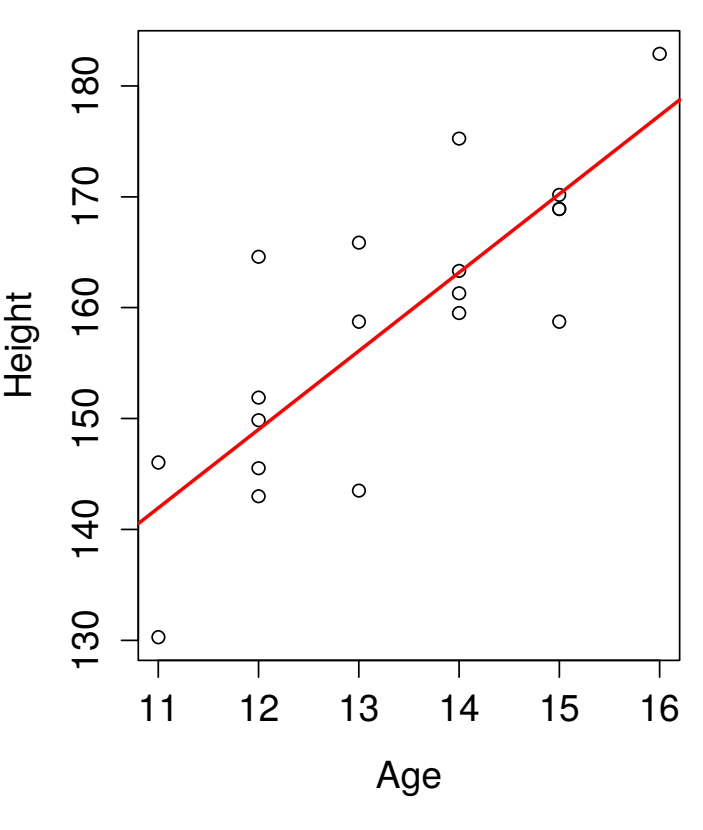




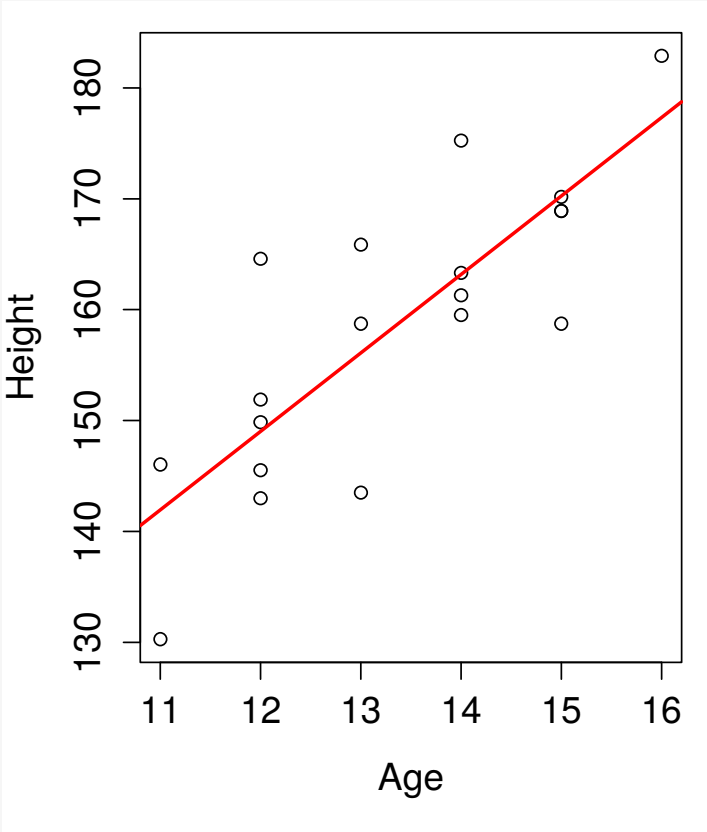
Some assumptions
can only be tested after the
model is run.

Representation of the fit

>plot(Age~Height,data=class_data)
>abline(model_height_age, col="red", lwd=2)



Functions to extract data from lm object (I): coefficients



coefficients: y-intercept and slope of the regression line

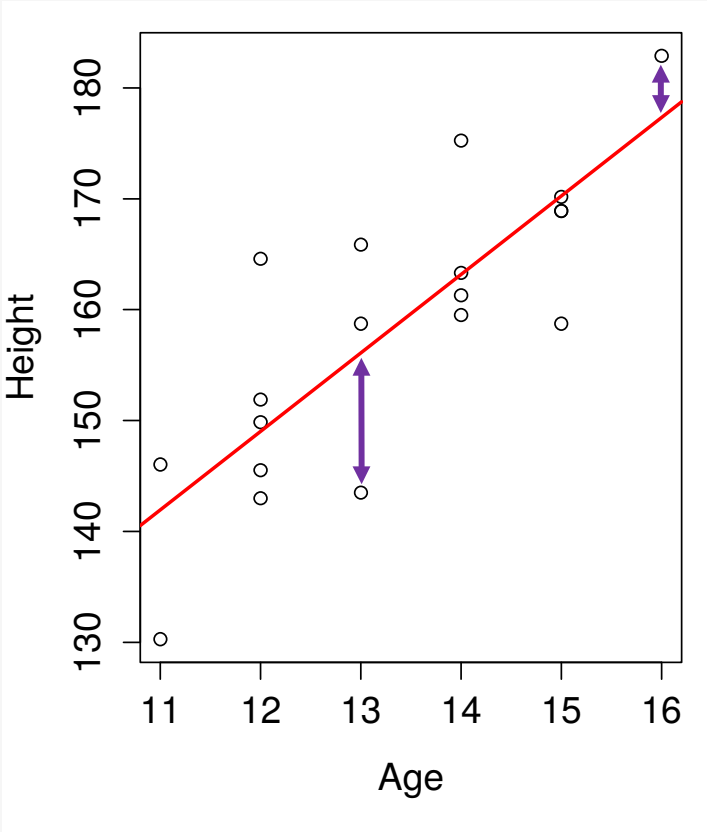
```
#get the coefficients as vector
>coef(model_height_age)
```

```
(Intercept)        Age
  64.068667   7.079333
```

y-intercept of the line

slope of the line

Functions to extract data from lm object (II): 2) residuals **- P5S**



residuals: vertical distances of data points from the regression line

```
# get the residuals as vector
>residuals(model_height_age)
```

```
        1         2         3         4         5
-11.6393   4.1087  -3.4787   2.8713   0.8393
        6         7         8         9        10
 -6.0187  15.5713 -12.5900   9.7620   2.6500
       11        12        13        14        15
 -3.6673  -1.8893  12.0807   0.1427 -11.5087
       16        17        18        19
 -1.3487  -0.0787  -1.3487   5.5420
```

```
>summary(model_height_age)
```

Height~a+b1*Age+ err

```
Call:
lm(formula = Height ~ Age, data = class_data)

Residuals:
     Min       1Q   Median       3Q      Max
-12.5900  -3.5730  -0.0787   3.4900  15.5713

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   64.069     16.565   3.868  0.00124 **
Age            7.079      1.237   5.724 2.48e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.832 on 17 degrees of freedom
Multiple R-squared:  0.6584,    Adjusted R-squared:  0.6383
F-statistic: 32.77 on 1 and 17 DF,  p-value: 2.48e-05
```

Error: Difference between the observed and the fitted points (line)

Significance of the parameters : Is Age different from 0? YES

R^2: Fraction of the variance explained by the model

F-test: Is the model significant compared to a model with just the intercept? YES

In a nutshell

R offers different ways to model your hypotheses. Choose one suited to your types of variables and your research question. Covered in this course:

- Comparing two group means

  t.test()

- Testing linear correlation between continuous variables:

  cor(), cor.test()

- Building simple linear models between a continuous variable and a continuous or categorical variable.

  lm()

Summary - Overall analysis workflow

1. Specify your biological question and your experimental design very clearly, then collect your data.
2. Save your data into a csv format in a dedicated folder.
3. Start up RStudio , create an R project, open a new script file and save it where you save your data. Don't forget to annotate it and save it regularly.
4. Import your data into R. Check everything in your data. Make sure it is what you expect it to be.
5. Explore your data, first with R's plotting functions. Make an hypothesis. Try to guess the answer that your statistical test should give you.
6. Perform your test to confirm your answer.
7. Communicate your findings.
8. Make sure your files (data, scripts, figures, reports) are well organised in your folder.

More to explore...

- **R manuals:** http://cran.r-project.org/manuals.html
- **Datacamp free tutorials:** https://www.datacamp.com/courses/free-introduction-to-r
- **STHDA (Statistical Tools for High Throughput Data Analysis) free tutorials:** http://www.sthda.com/english/
- **Stackoverflow** documentation, resources and user forum: http://stackoverflow.com/tags/r/info
- **Rseek -** search engine on numerous online R resources: http://www.rseek.org

Credits and Acknowledgments

- Content and slides developed by:
  Diana Marek, Geoffrey Fucile, Alex Smith, Linda Dib, Leonore Wigger, Wandrille Duchemin
- Content inspired by material from:
  - Owen L. PETCHEY and “Getting started with R” book
  - Daniel WEGMANN and Frédéric SCHÜTZ
  - Robert STOJNIĆ and Ian ROBERTS
  - Jenny DRNEVICH

Thank you for your attention

http://www.sib.swiss/training

Any questions? Contact training@sib.swiss

Let’s practice - 11

The data set "Pima" comes from a study on diabetes in women of Pima Indian heritage. We are using a subset (Pima.tr).

1) Load the package MASS using library(). (You may need to install it first). Load the dataset Pima.tr using data(). Use ? to get an idea which variables it contains.

2) Hypothesis: Blood glucose level (glu) is associated with diastolic blood pressure (bp). Run a linear model to test the hypothesis.

3) Visualize the fit with a scatter plot and a trend line.

4) Check assumptions of the model (homoscedasticity, mean of residual at 0, normality of the residuals) graphically.