IMPORTANT

Course room

Monday, Tuesday, Wednesday:

– Génopode Building 2020

Thursday:

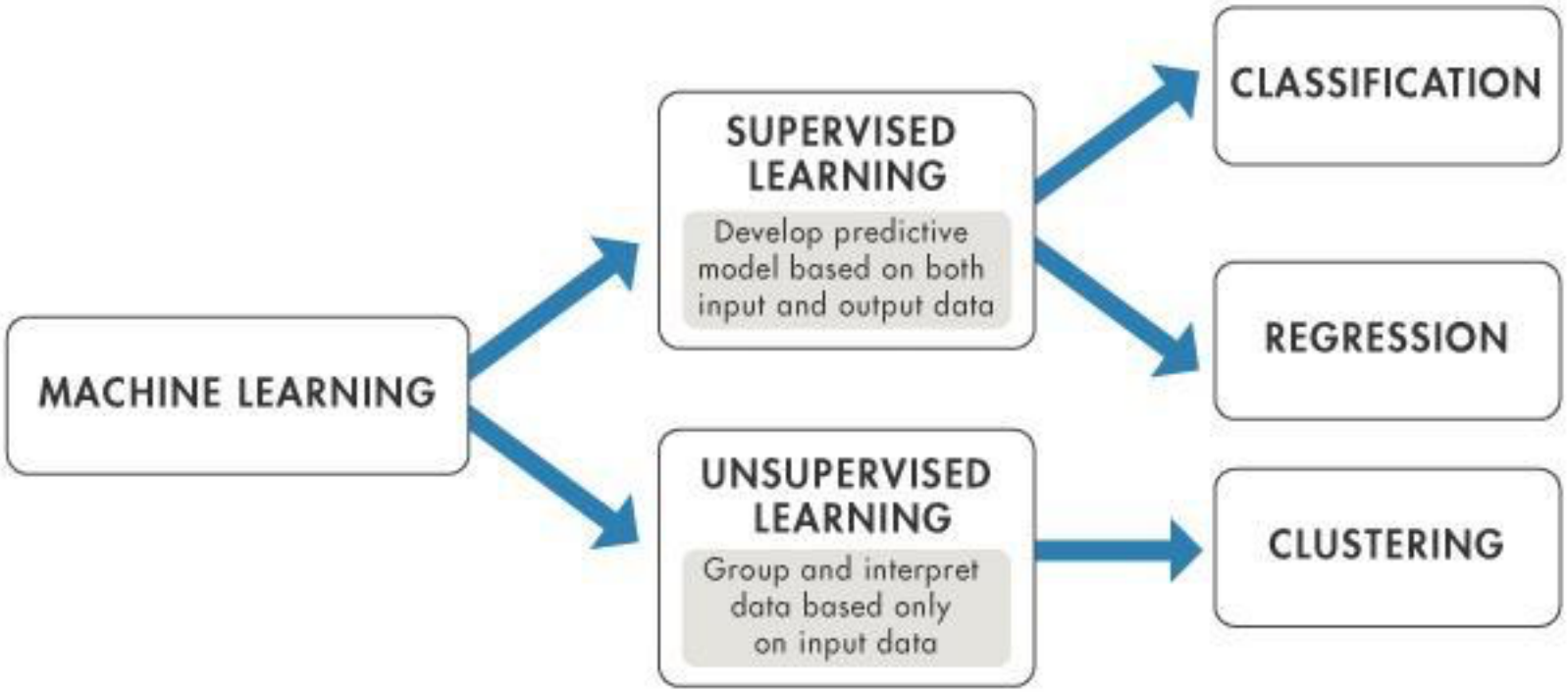
– Amphipôle Building 321

Course web-page

- Course page:
- https://edu.sib.swiss/course/view.php?id=344

- Login: smbd18
- Password: SIB-smbd18

Machine Learning

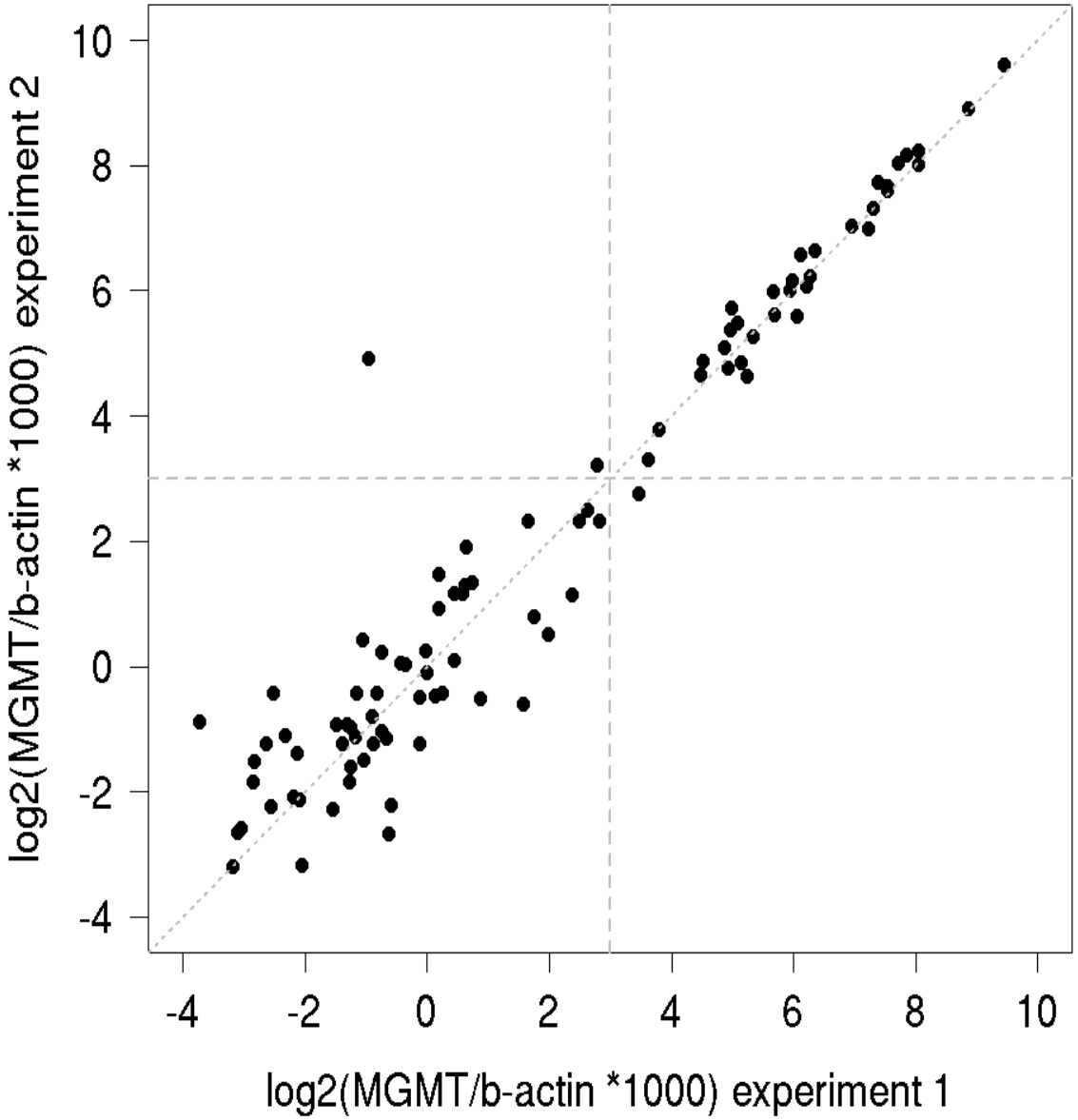


Credits: Rory Bunker, Fadi Thabtah

Regression & Correlation

Reproducibility of duplicate measurements.

Dotted line represents identity line (x = y). Dashed lines represent the cut-off between unmeth and methylated samples.



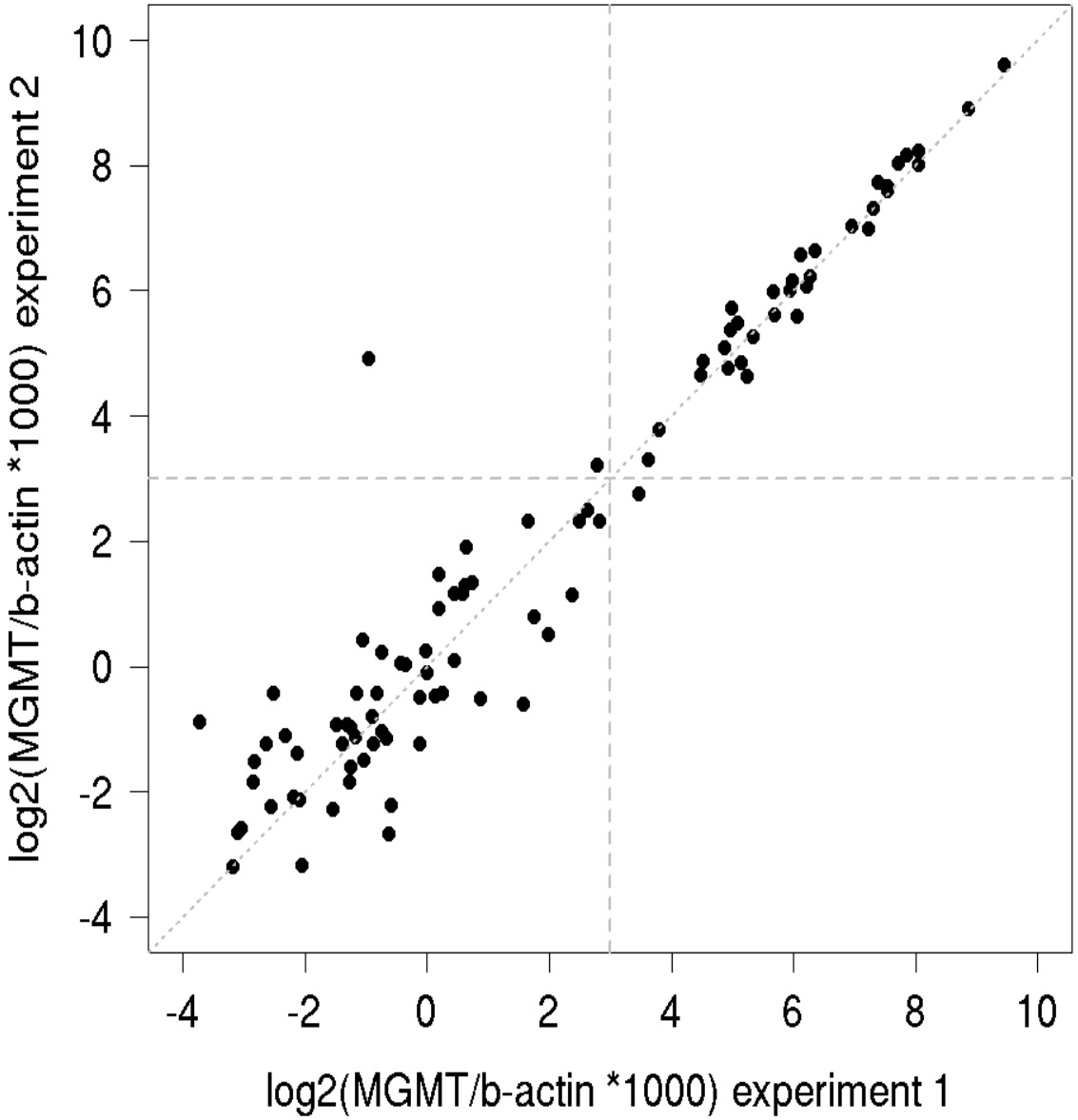
Credits: Eugenia Migliavacca

Reproducibility of duplicate measurements.

Dotted line represents identity line (x = y). Dashed lines represent the cut-off between unmeth and methylated samples.

Pearson correlation 0.996,

Spearman correlation 0.93, N=94.



Credits: Eugenia Migliavacca

Correlation r

Correlation r

is a measure of linear association

Correlation r

It indicates the strength of a linear relationship between two variables

The *correlation coefficient* r is defined as the average value of the product

(X in SUs)*(Y in SUs)

The *correlation coefficient* r is defined as the average value of the product

$$(X \text{ in SUs})*(Y \text{ in SUs})$$

where SU = standard units,

X in SUs = (X – mean(X))/SD(X),

Y in SUs = (Y – mean(Y))/SD(Y).

The *correlation coefficient* r is defined as the average value of the product

$$(X \text{ in SUs})*(Y \text{ in SUs})$$

where SU = standard units,

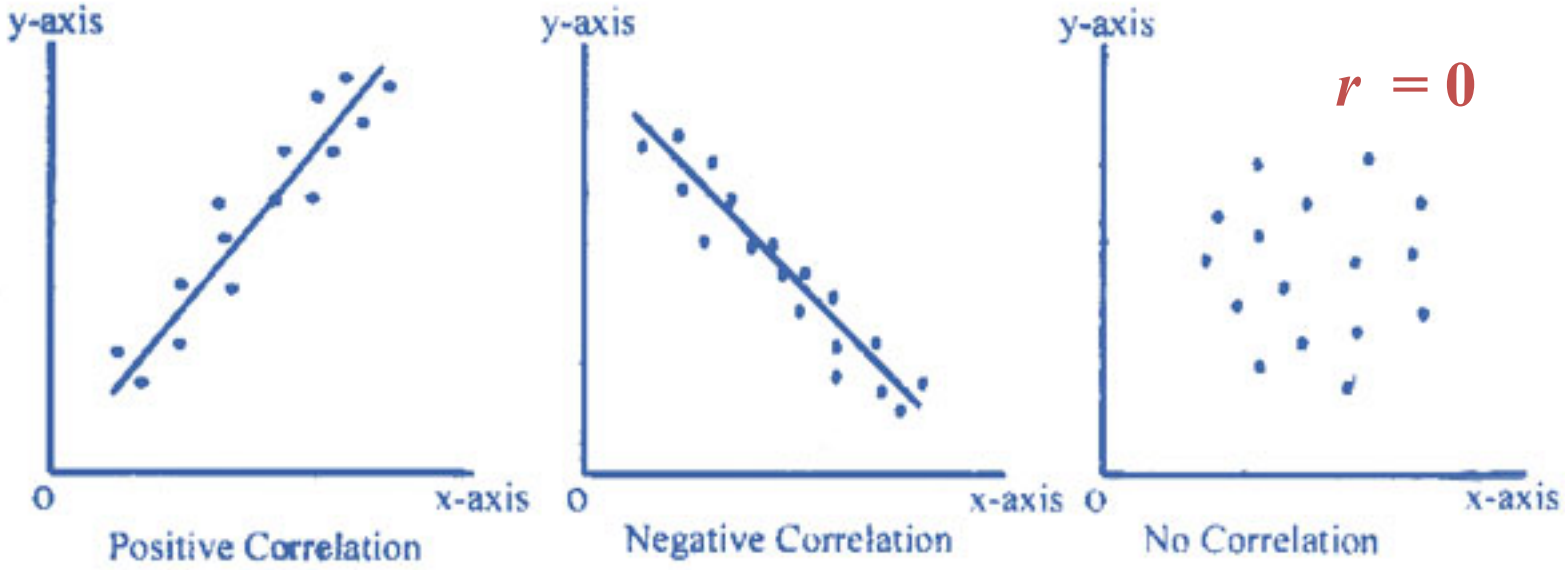
X in SUs = (X – mean(X))/SD(X),

Y in SUs = (Y – mean(Y))/SD(Y).

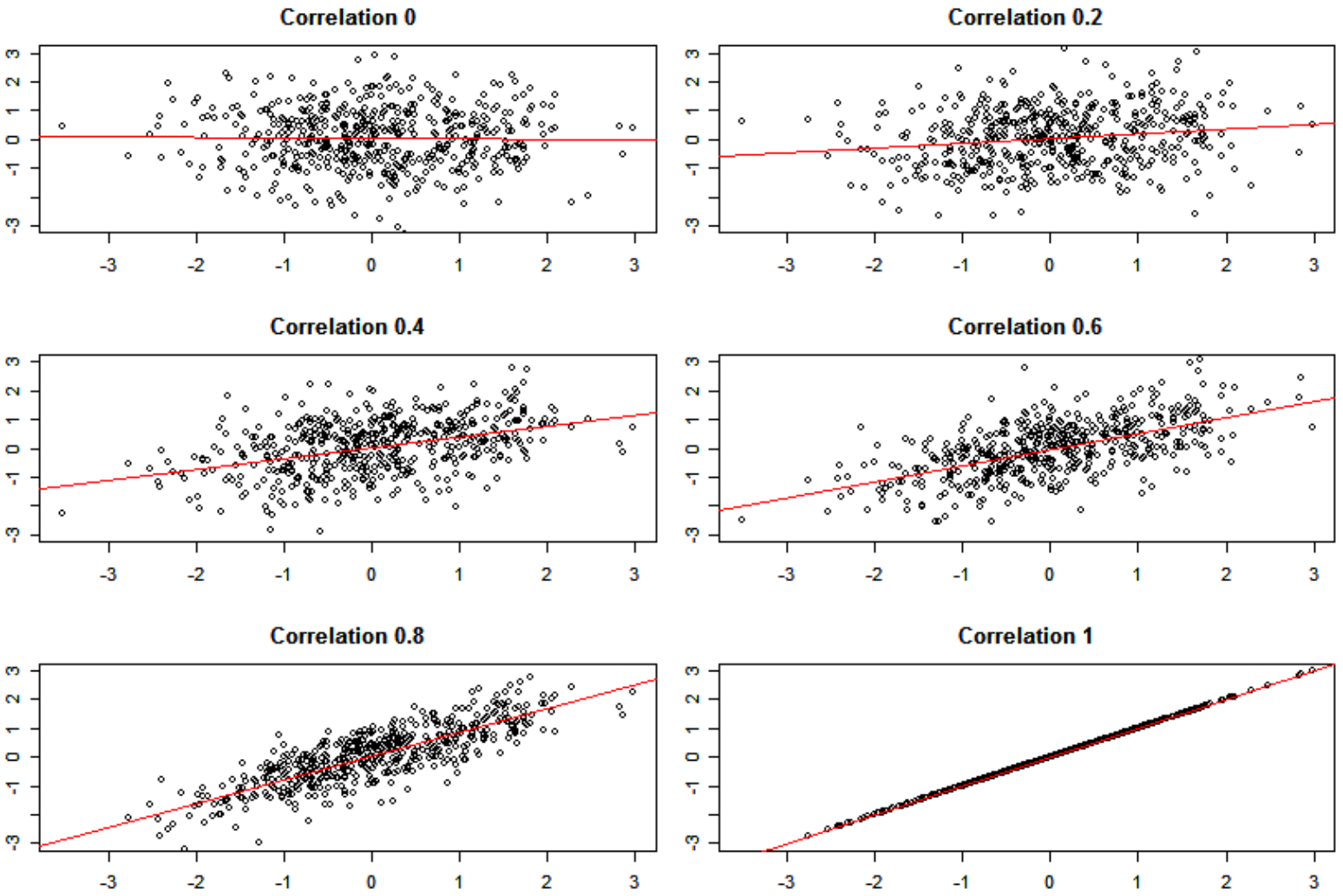
$$\mathbf{-1 \le r \le 1}$$

r* is a *unit-less quantity

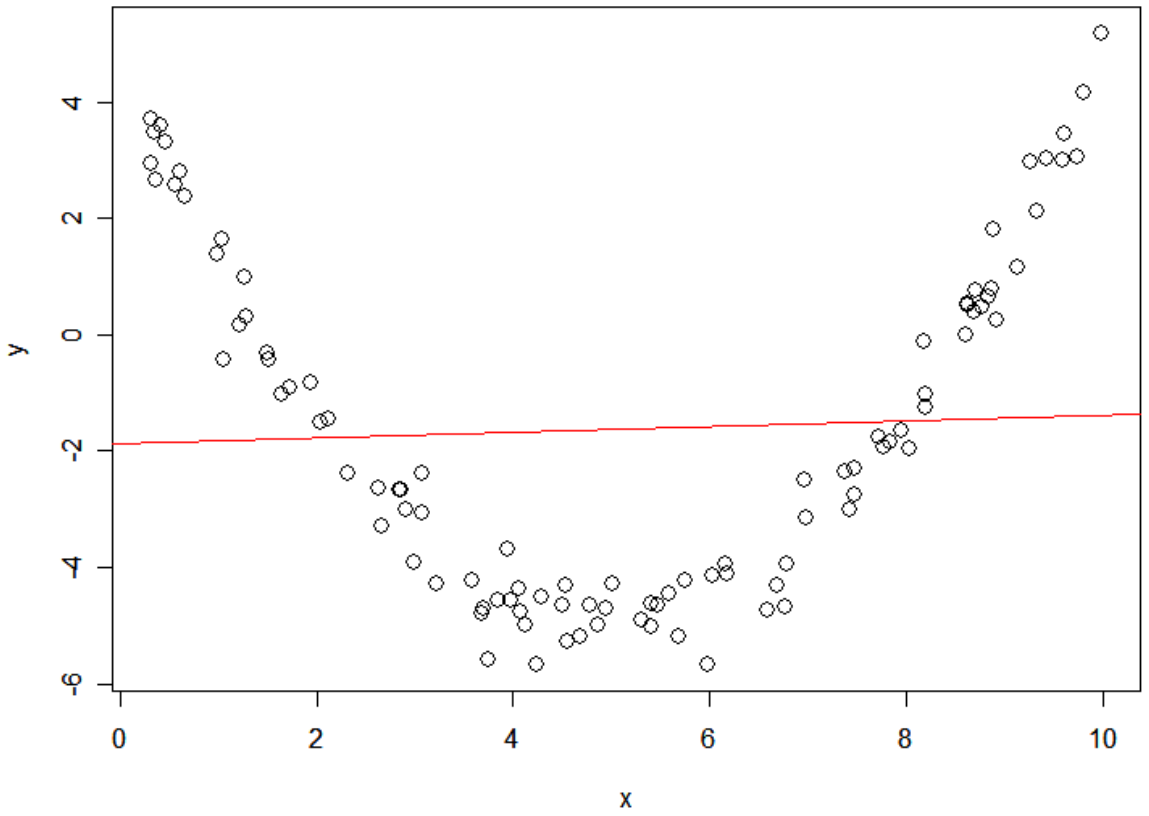
the closer r is to -1 or 1, the more tightly the points on the scatterplot are clustered around a line



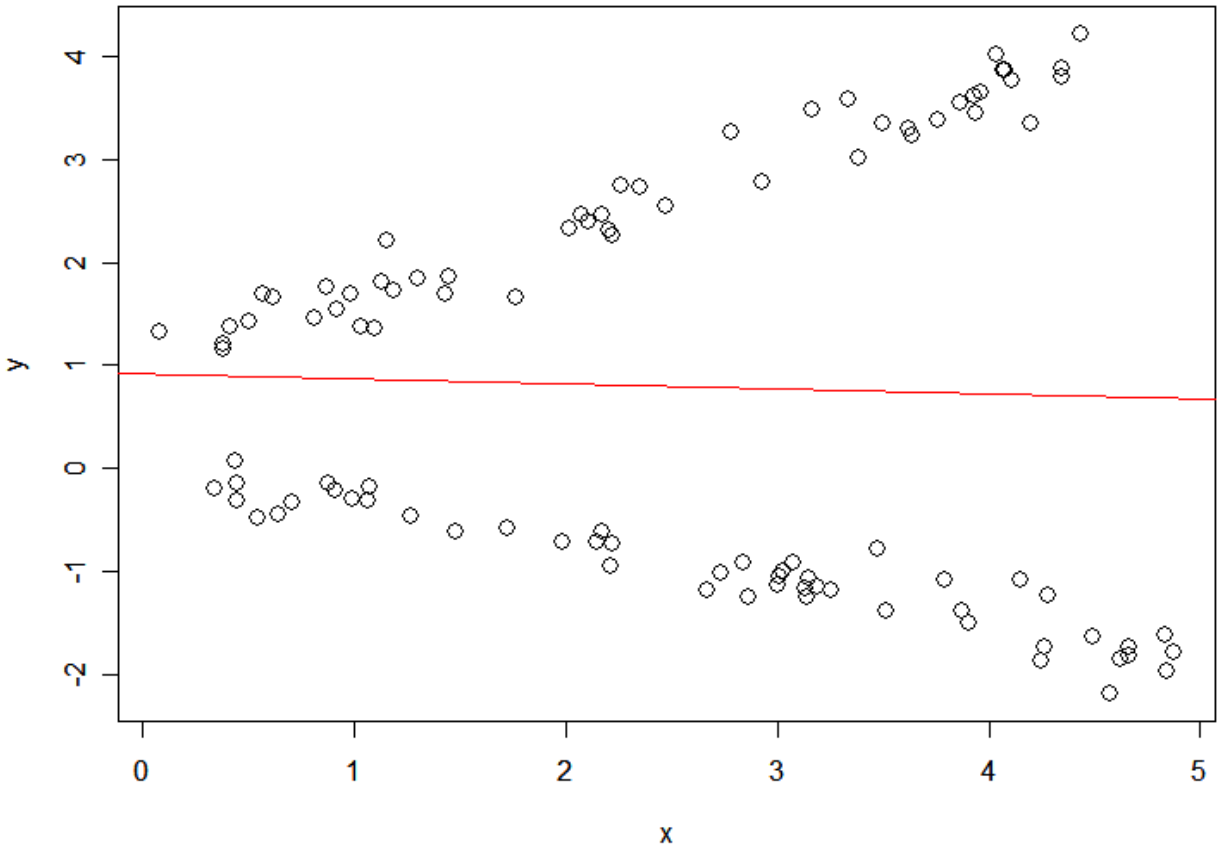
Examples of correlations between two variables



Correlation r = 0



Correlation r = 0



...and what r is not

- ***r is*** **a measure of *LINEAR ASSOCIATION***
- ***r*** **does *NOT* tell us if *Y* is a function of *X***
- ***r*** **does *NOT* tell us if *X causes Y***
- ***r*** **does *NOT* tell us if *Y causes X***
- ***r*** **does *NOT* tell us the slope of the line (except for its sign)**
- ***r*** **does *NOT* tell us what the scatterplot looks like (it is only a summary of the data)**

Correlation is NOT causation

- **You *cannot* infer that since *X* and *Y* are highly correlated (r close to –1 or 1) that *X* is *causing* a change in *Y***
- ***Y* could be causing *X***
- ***X* and *Y* could both be varying along with a third, possibly unknown factor (either causal or not; often *'time'*):**

Correlation does not imply causality !

Storks Deliver Babies ($p = 0.008$)

KEYWORDS:
Teaching;
Correlation;
Significance;
p-values.

Robert Matthews
Aston University, Birmingham, England.
e-mail: rajm@compuserve.com

Summary
This article shows that a highly statistically significant correlation exists between stork populations and human birth rates across Europe. While storks may not deliver babies, unthinking interpretation of correlation and *p*-values can certainly deliver unreliable conclusions.

Teaching Statistics. Volume 22, Number 2, Summer 2000

The NEW ENGLAND JOURNAL of MEDICINE

OCCASIONAL NOTES

Chocolate Consumption, Cognitive Function, and Nobel Laureates

Franz H. Messerli, M.D.

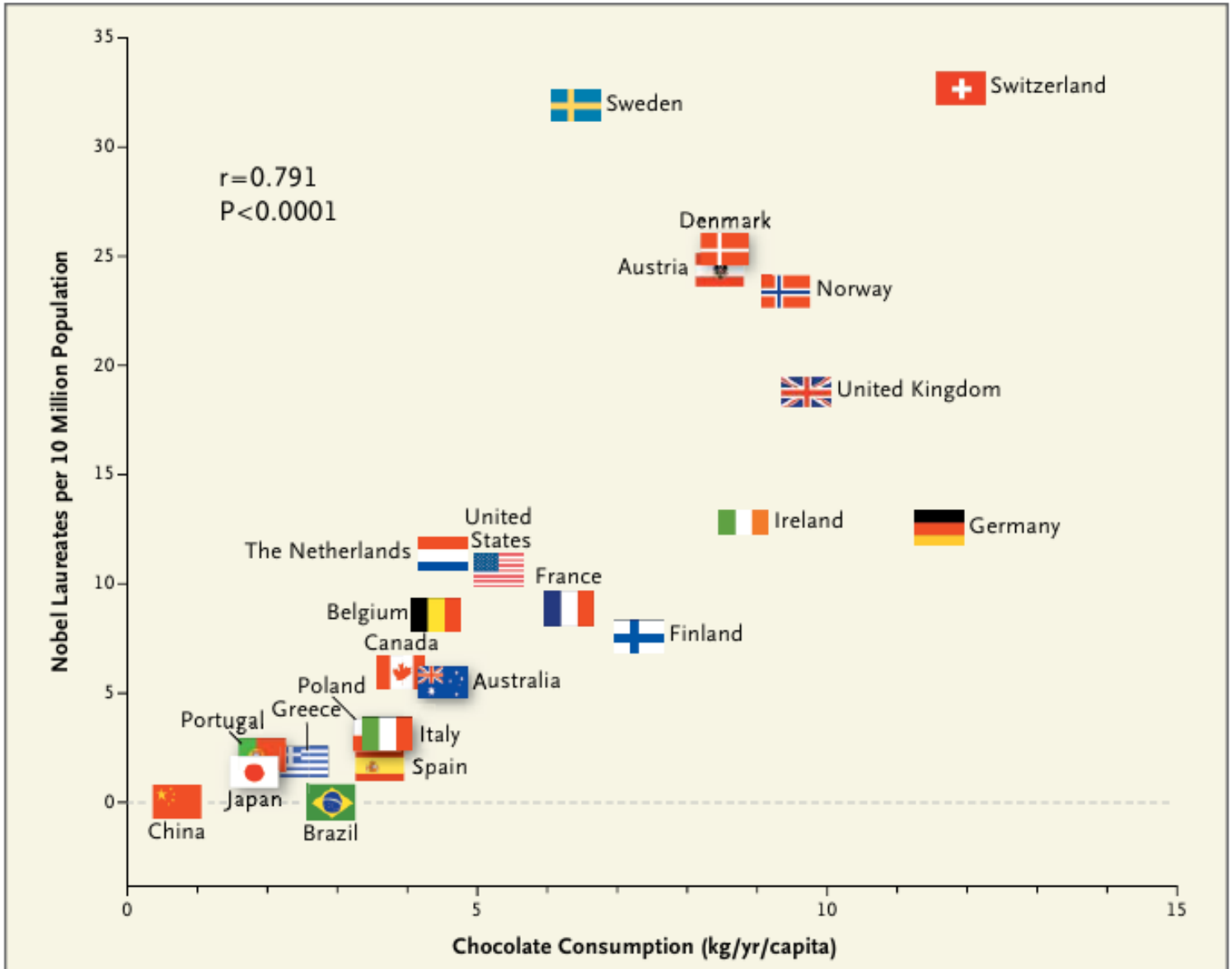


Figure 1. **Correlation between Countries' Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.**

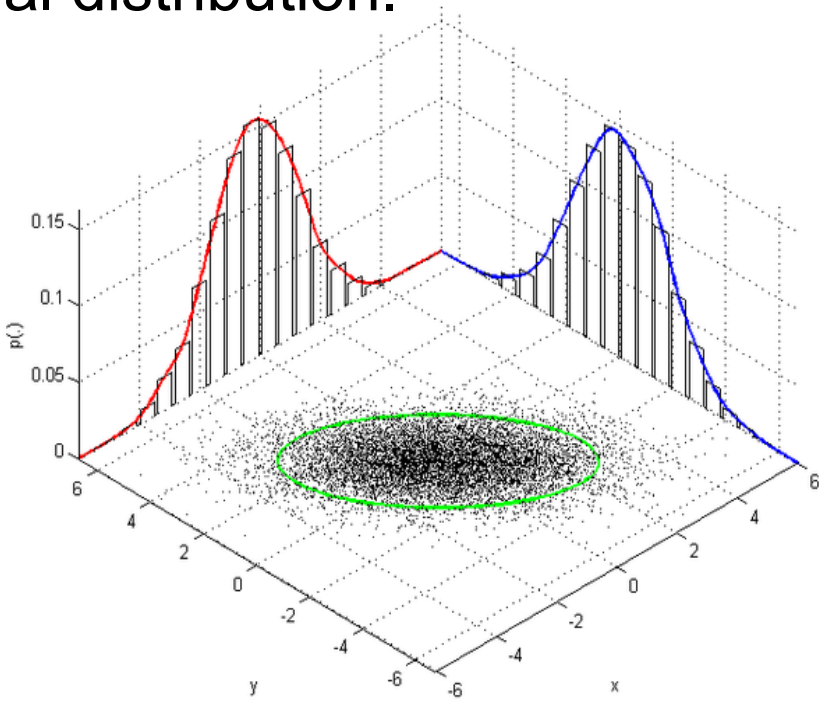
Pearson's correlation assumes that the data follows a bivariate normal distribution.

It will only assess whether there is a linear correlation in the data.

Other types of correlation (robust methods) are available: most commonly, the Spearman and Kendall correlations

Pearson, Spearman and Kendall correlations

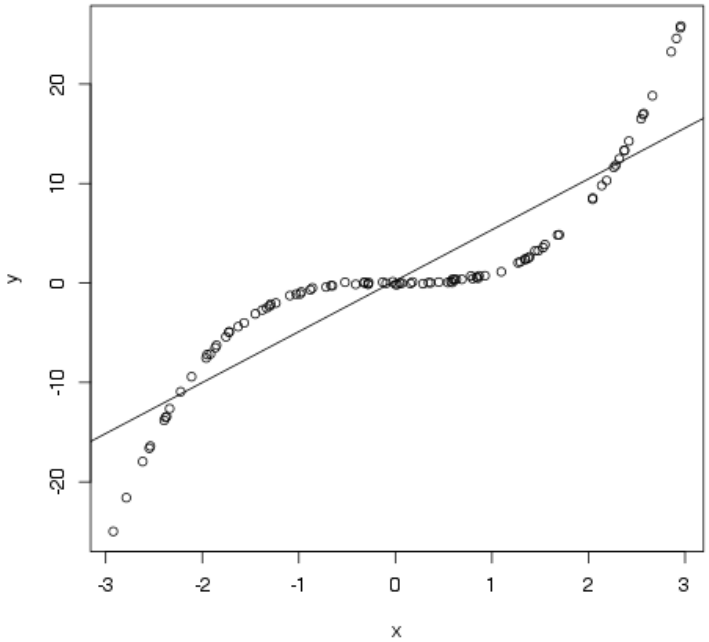
Pearson's correlation assumes that the data follows a bivariate normal distribution.



Measures the joint variability of two normalized variables.

Pearson, Spearman and Kendall correlations

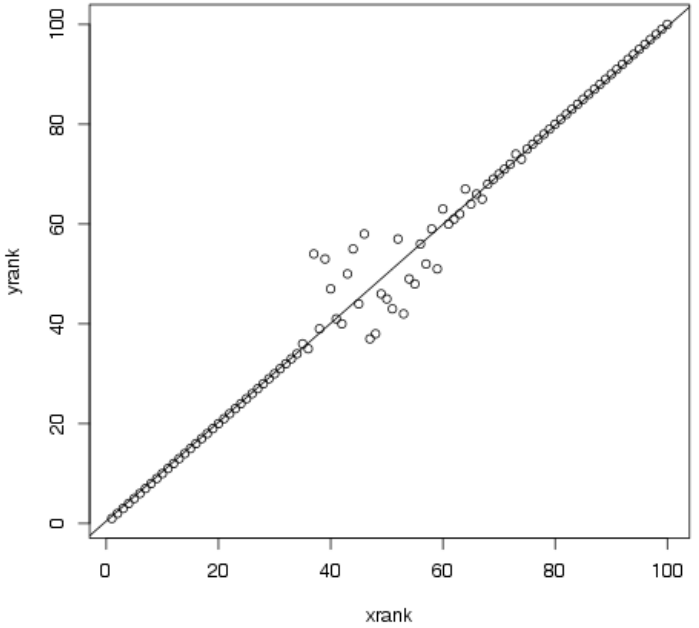
Spearman's correlation is based on the rank of values



Raw data

Pearson correlation: 0.90

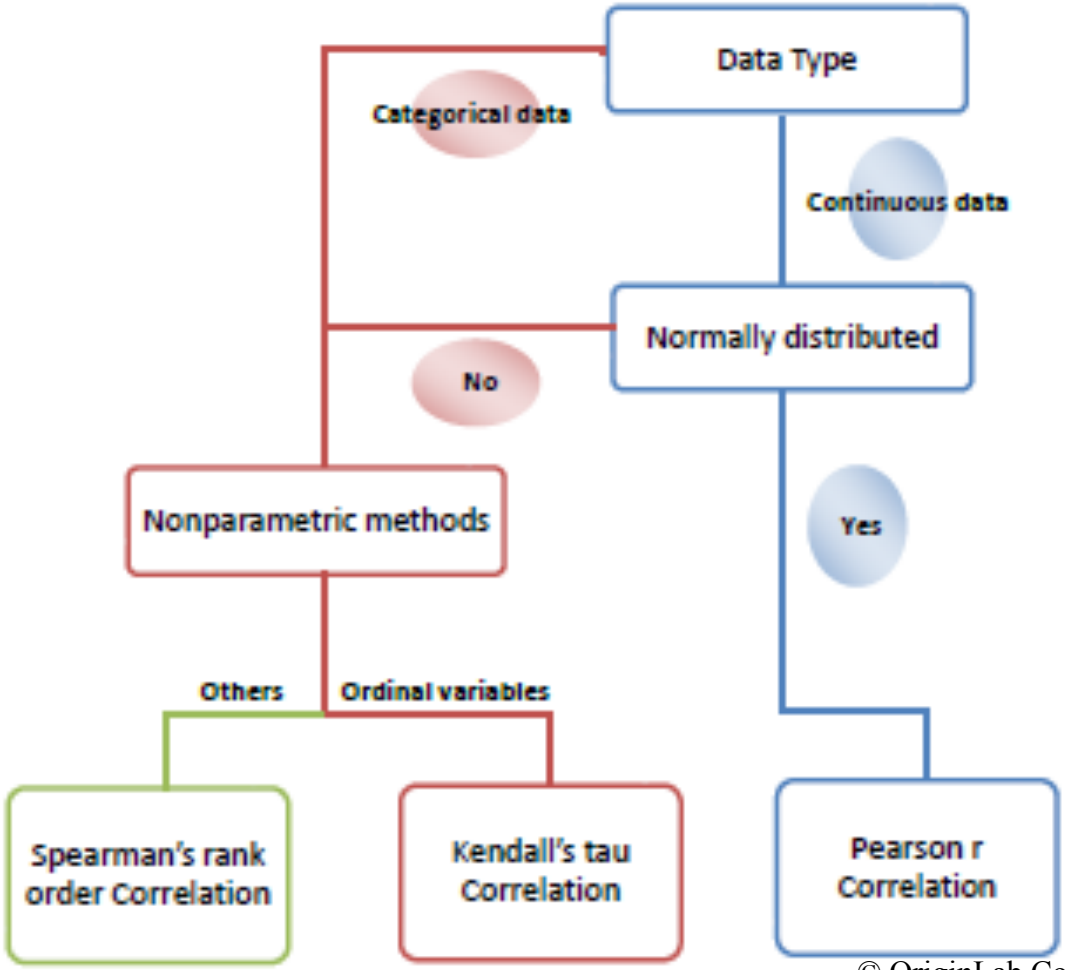
Spearman correlation: 0.99



Ranked data

Pearson correlation: 0.99

Correlation methods



In R:

```
>?cor
```

```
>cor(x,y)
```

Note, however, that if there are *missing values* (*NA*), then you will get an *error message*

In R:

```
>?cor

>cor(x,y)
```

Note, however, that if there are *missing values* (*NA*), then you will get an *error message*

Elementary statistical functions in R require

- *no* missing values, or
- explicit statement of what to do with *NA (na.rm=TRUE)*

```
> cor.test(x,y)
```

Testing whether a correlation is different from 0

```
> cor.test(x,y)

    Pearson's product-moment correlation

data:  x and y
t = 21.5241, df = 98, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8667723 0.9376171
sample estimates:
      cor
0.9085158
```

R Vs RevoScaleR

```
#Initialize some variables to specify the data sets.
inputFileClass <-
paste0("/media/sf_docVM/correlationregression/","class.csv")

#Import the  data.
class_data<- rxImport(inData = inputFileClass)
```

R

```
cor(  class_data[,3],
      class_data[,5])
```

RevoScaleR

```
rxCor(formula=~Height+Age,
      data = class_data[,4:5],
      reportProgress = 0)
```

Correlation measures the relationship

Correlation measures the relationship

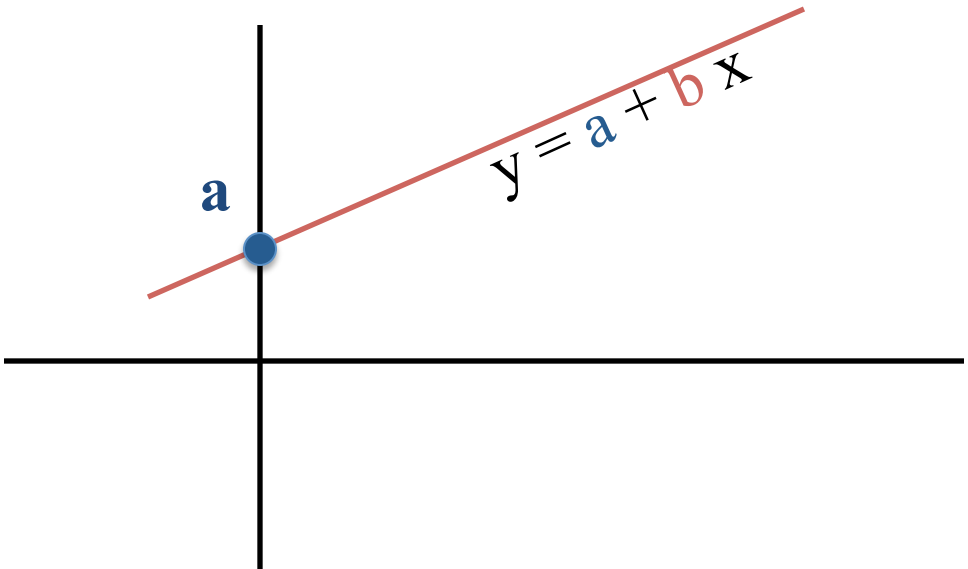
But it does not describe it

Description can be a line: linear Model

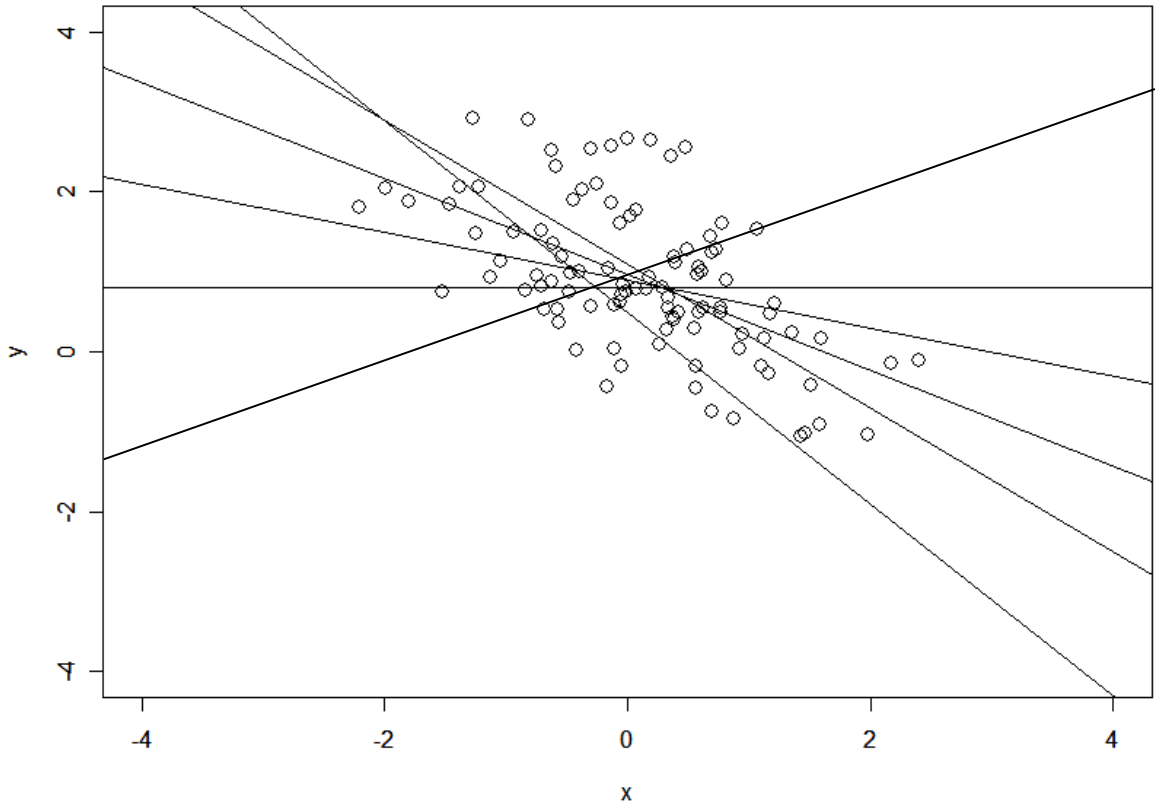
The equation for a line to predict y knowing x (in slope-intercept form) looks like

$$y = a + b\,x$$

where *a* is called the *intercept* and *b* is the *slope*.



What is the “best” line which fits this data ?
Can we use it to summarise the relation between x and y ?



y = 0.9 + 0.6x

y = 0.8 + 0x

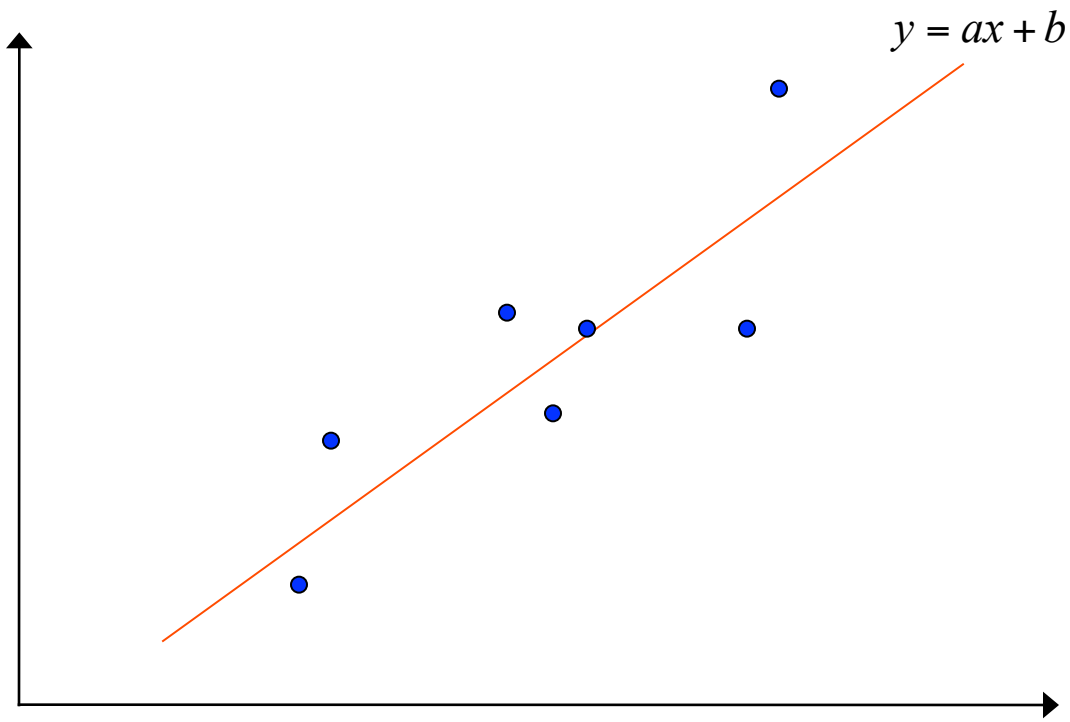
y = 0.9 – 0.3x

y = 1 – 0.6x

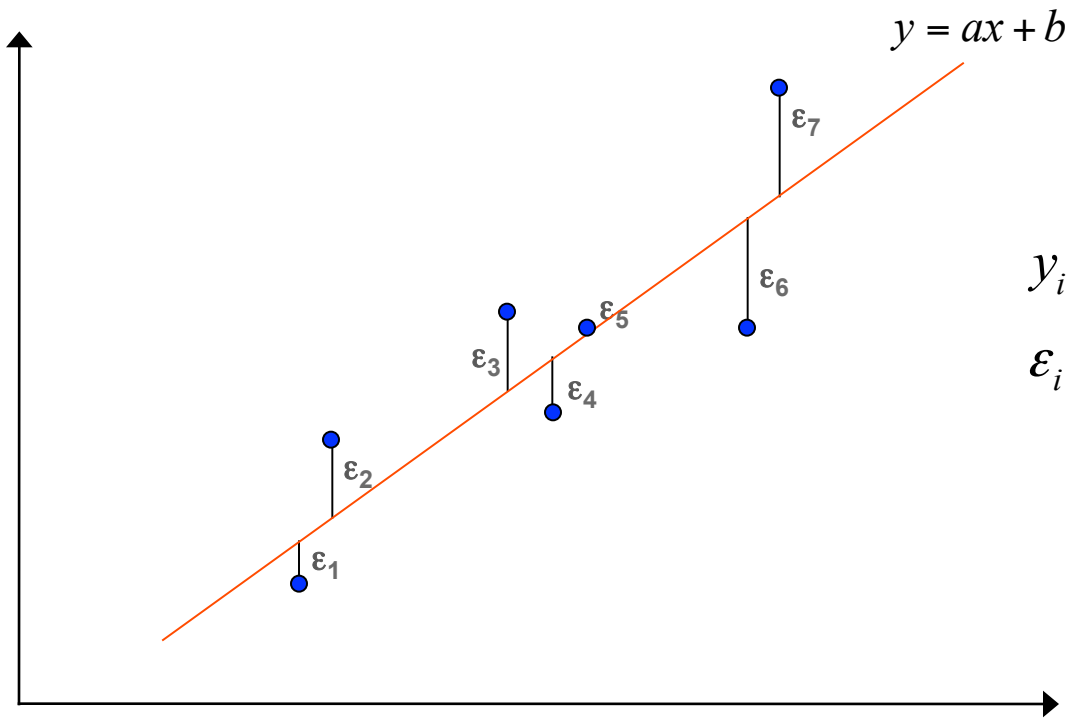
y = 1.1 – 0.9x

y = 0.5 – 1.2x

Least-square fitting



Least-square fitting



$$y_i = ax_i + b + \varepsilon_i$$

$$\varepsilon_i = y_i - (ax_i + b)$$

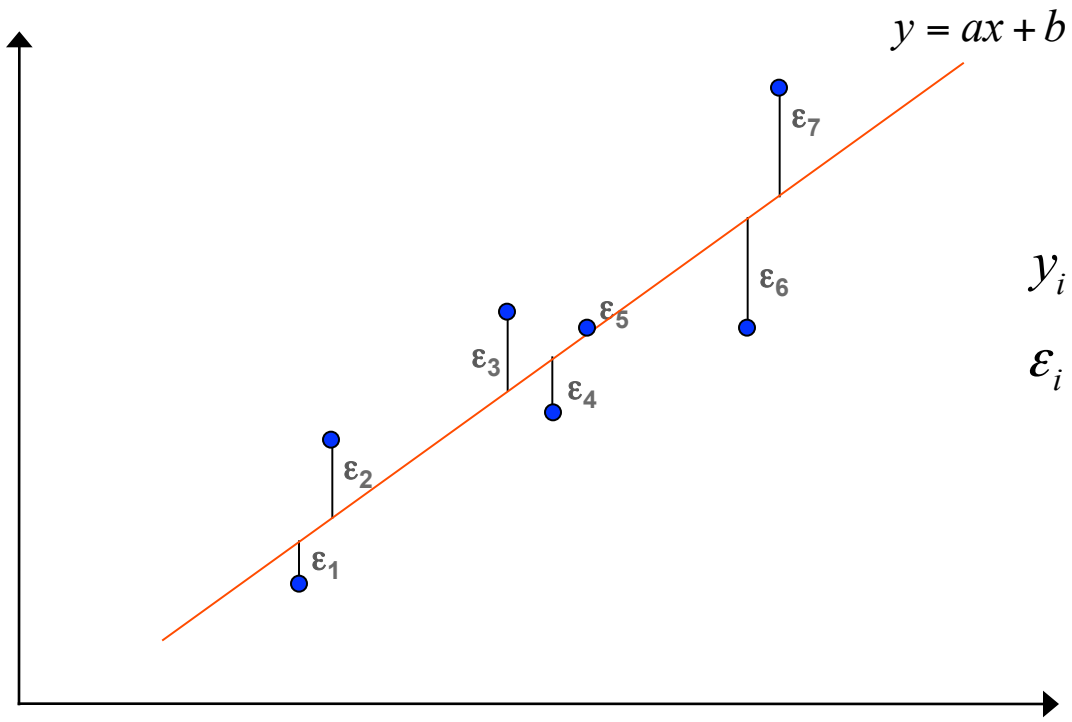
The least-squares procedure finds the straight line with the smallest sum of squares of vertical errors.

Regression line such that:

$$\sum_i \varepsilon_i^2 = \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \ldots$$

minimum

Least-square fitting

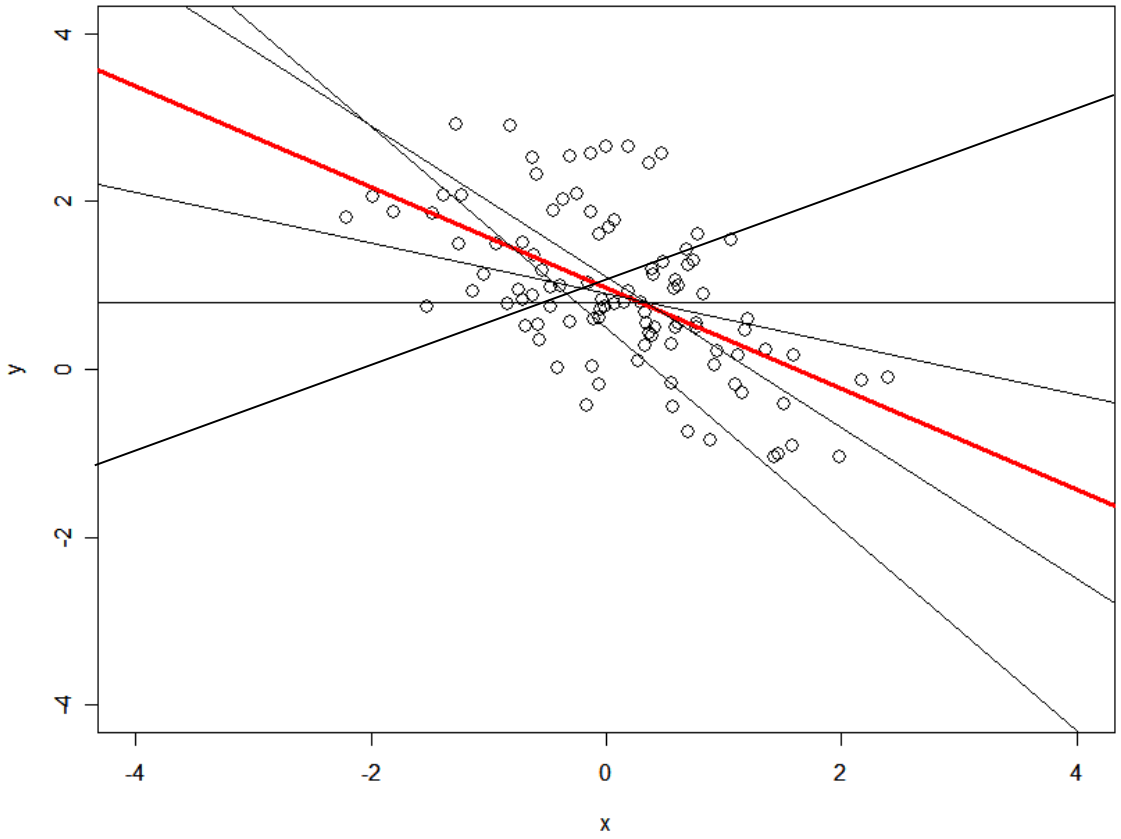


$$y_i = ax_i + b + \varepsilon_i$$

$$\varepsilon_i = y_i - (ax_i + b)$$

The least-squares procedure finds the straight line with the smallest sum of squares of vertical errors.

Over all possible straight lines, y= 1 - 0.6x is the "best" possible line according to this criterion.



y = 0.9 + 0.6x

y = 0.8 + 0x

y = 0.9 – 0.3x

y = 1 – 0.6x

y = 1.1 – 0.9x

y = 0.5 – 1.2x

Linearity in linear models

Linearity is about the model parameters

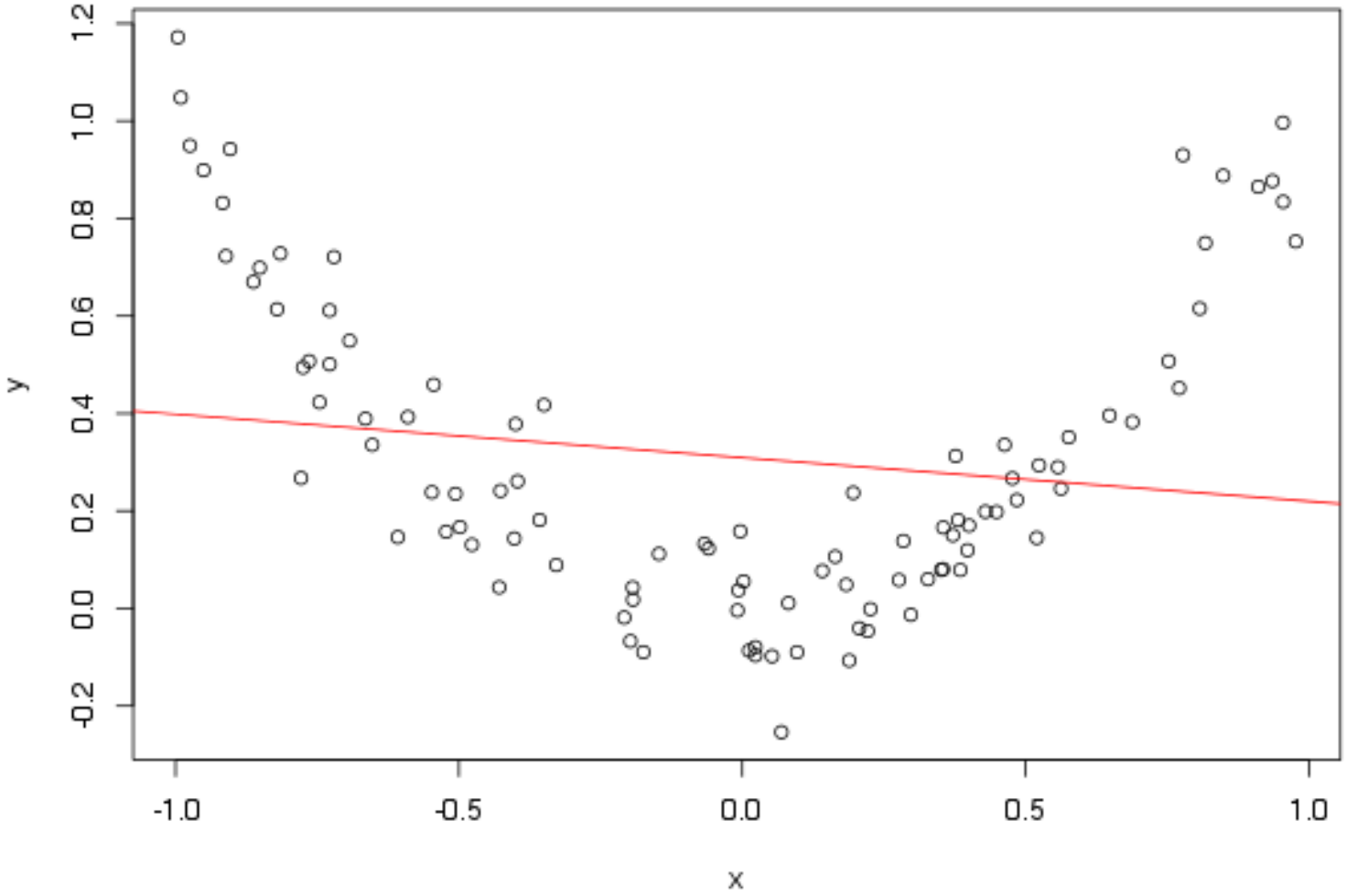
$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{p-1} X_{ip-1} + \varepsilon_i$$
$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \varepsilon_i$$
$$Y_i = \beta_0 + \beta_1 \log X_{i1} + \beta_2 X_{i2} + \varepsilon_i$$
$$Y_i = \beta \sin X_i + \varepsilon_i$$

Linear in βs

$$Y_i = \beta_0 + \log(\beta_1 X_{i1} + \beta_2 X_{i2}) + \beta_3 X_{i3} + \varepsilon_i$$
$$Y_i = \beta_0 + \beta_1 \exp(\beta_2 X_i + \beta_3) + \varepsilon_i$$

Not linear in βs

What if the data is not linear ?

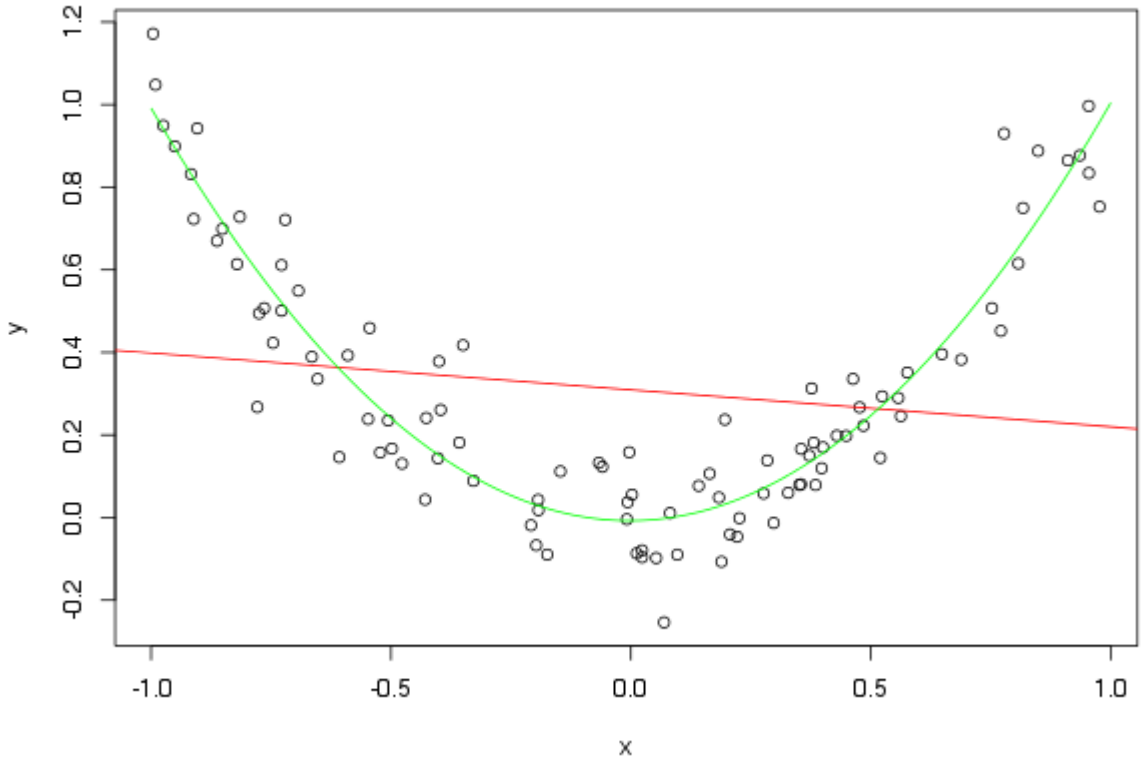


What if the data is not linear ?

Use a polynomial regression

$$y = b_0 + b_1 x + b_2 x^2$$

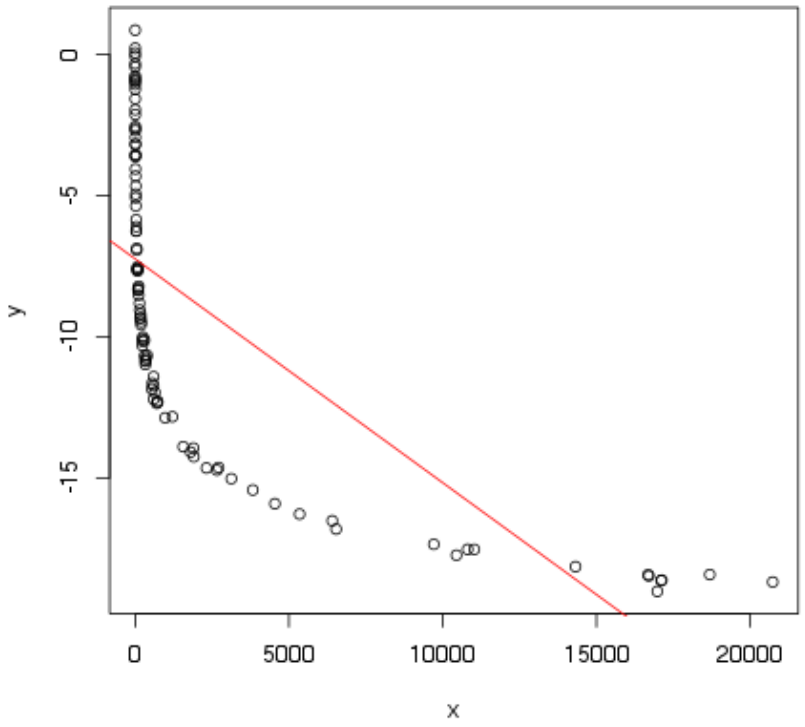
This is still linear for b_i; it is as if we had added a new variable.

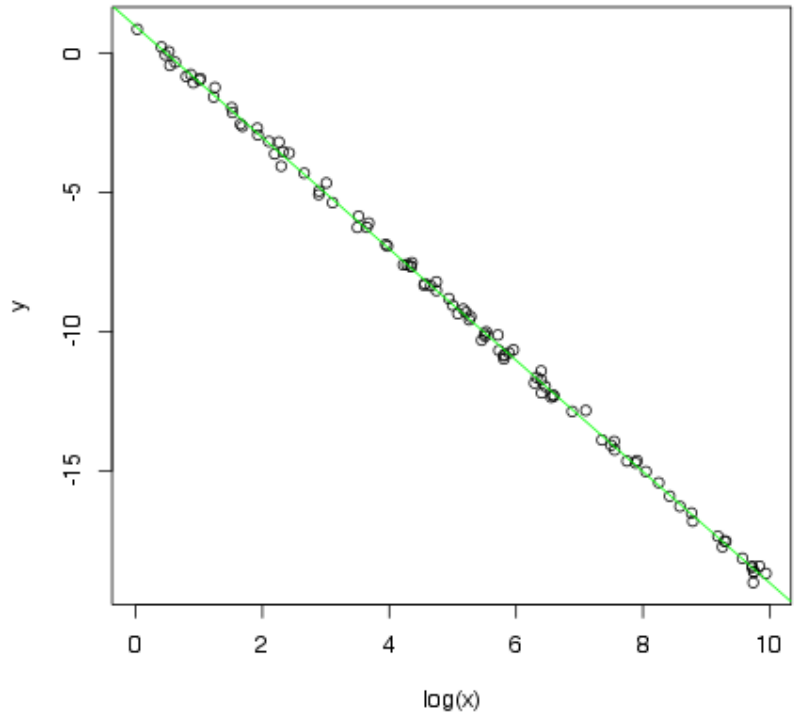


What if the data is not linear ?

Consider transforming the data (log)

$$\log(y) = a + b\,x$$





Linear models in matrix terms

$$Y_i = \beta_0 + \beta_1 X_{\mathbf{i}} + \varepsilon_i$$

is equivalent to

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ 1 & \vdots \\ 1 & X_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

or $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$$

is equivalent to

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & X_{12} \\ 1 & X_{21} & X_{22} \\ 1 & \vdots & \vdots \\ 1 & X_{n1} & X_{n2} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

or $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{p-1} X_{ip-1} + \varepsilon_i$$

is equivalent to

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1p-1} \\ 1 & X_{21} & X_{22} & \cdots & X_{2p-1} \\ 1 & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & \cdots & X_{np-1} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

or $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

Least-square estimation of regression coefficients

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{p-1} X_{ip-1} + \varepsilon_i$$

is equivalent to

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1p-1} \\ 1 & X_{21} & X_{22} & \cdots & X_{2p-1} \\ 1 & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & \cdots & X_{np-1} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

or $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

Least-square estimation of regression coefficients

$\mathbf{b} = (b_0 \cdots b_{p-1})'$ estimator of $\boldsymbol{\beta}$ is computed as follows:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

$$\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{Y} \qquad \text{where} \quad E\{\boldsymbol{\varepsilon}\} = \mathbf{0}$$

Least-square estimation of regression coefficients

$\mathbf{b} = (b_0 \cdots b_{p-1})'$ estimator of $\boldsymbol{\beta}$ is computed as follows:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

$$\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{Y} \qquad \text{where} \quad E\{\boldsymbol{\varepsilon}\} = \mathbf{0}$$

$$\boldsymbol{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

Computationally intensive

$$Y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3$$

in R:

```
yvar ~ xvar1 + xvar2 + xvar3
```

`~` **as "*described (or modeled) by*"**

$$Y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3$$

in R:

```
yvar ~ xvar1 + xvar2 + xvar3
```

`~` **as "*described (or modeled) by*"**

By default, an intercept is included in the model

To leave the intercept out:

```
yvar ~ -1 + xvar1 + xvar2 + xvar3
```

Generic form

`response ~ predictors`

predictors can be `numeric` **or** `factor`

R symbols to create formulas

`+` **to *add* more variables**

`-` **to *leave out* variables**

`:` **to introduce *interactions* between two terms**

`*` **to include *both interactions and the terms***

(`a*b` is the same as `a + b + a:b`)

`^n` ***adds all terms*** **including interactions up to order n**

`I()` **treats what's in () as a *mathematical expression***

A concrete example in R

Using the CLASS dataset, from the program SAS
(units have been modified from imperial to metric)

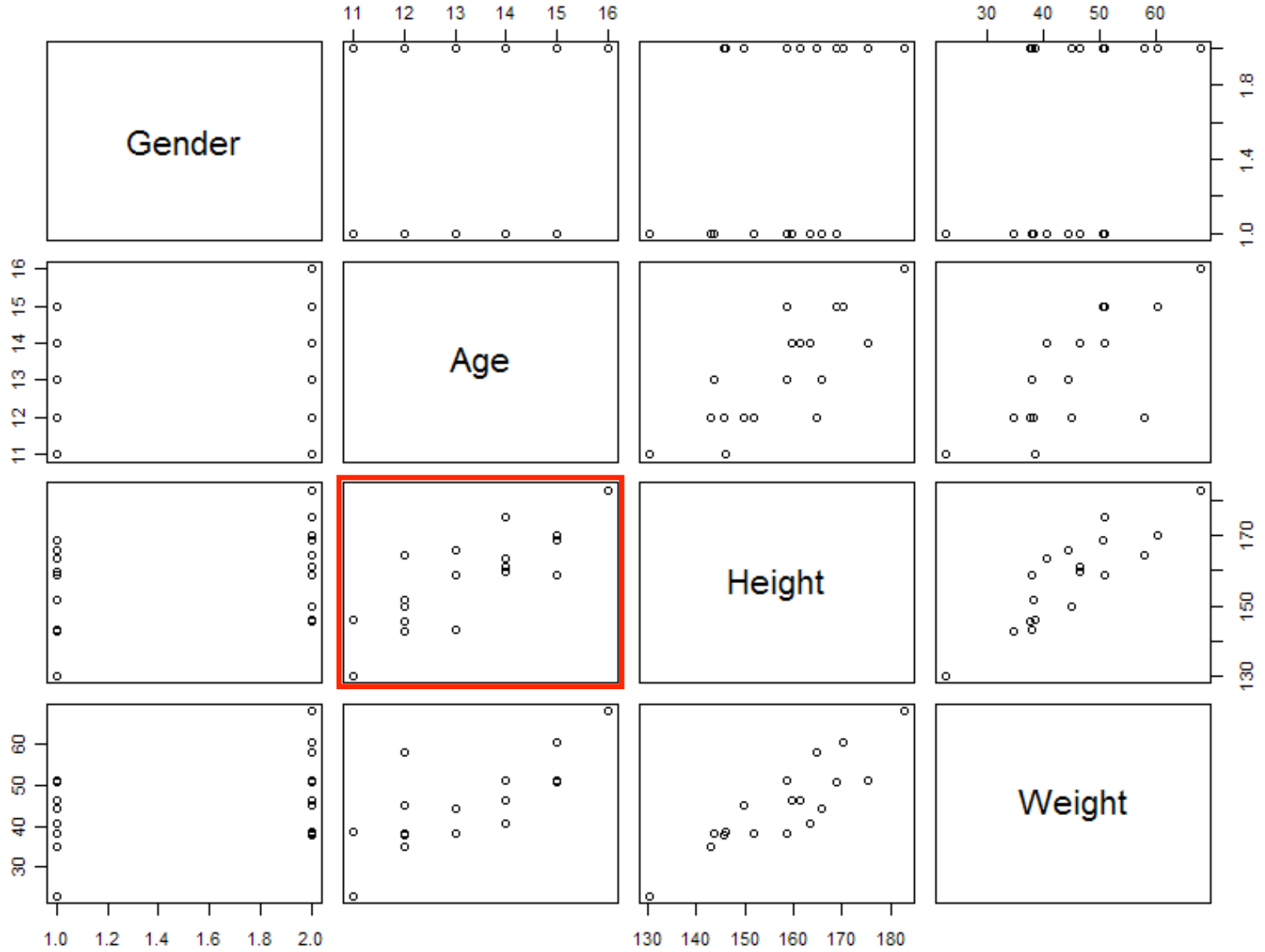
```
data <- read.table("http://lausanne.isb-sib.ch/~schutz/data/class.txt")
```

The CLASS dataset from SAS

```
> data
      Name Gender Age  Height  Weight
1    JOYCE      F  11 130.302 22.8765
2   THOMAS      M  11 146.050 38.5050
3    JAMES      M  12 145.542 37.5990
4     JANE      F  12 151.892 38.2785
5     JOHN      M  12 149.860 45.0735
6   LOUISE      F  12 143.002 34.8810
7   ROBERT      M  12 164.592 57.9840
8    ALICE      F  13 143.510 38.0520
9  BARBARA      F  13 165.862 44.3940
10 JEFFREY      M  13 158.750 38.0520
11   CAROL      F  14 159.512 46.4325
12   HENRY      M  14 161.290 46.4325
13  ALFRED      M  14 175.260 50.9625
14    JUDY      F  14 163.322 40.7700
15   JANET      F  15 158.750 50.9625
16    MARY      F  15 168.910 50.7360
17  RONALD      M  15 170.180 60.2490
18 WILLIAM      M  15 168.910 50.7360
19  PHILIP      M  16 182.880 67.9500
```

The CLASS dataset from SAS

```
> summary(data[,-1])

Gender      Age            Height          Weight
F: 9   Min.   :11.00   Min.   :130.3   Min.   :22.88
M:10   1st Qu.:12.00   1st Qu.:148.0   1st Qu.:38.17
       Median :13.00   Median :159.5   Median :45.07
       Mean   :13.32   Mean   :158.3   Mean   :45.31
       3rd Qu.:14.50   3rd Qu.:167.4   3rd Qu.:50.85
       Max.   :16.00   Max.   :182.9   Max.   :67.95


> pairs(data[,-1])
```

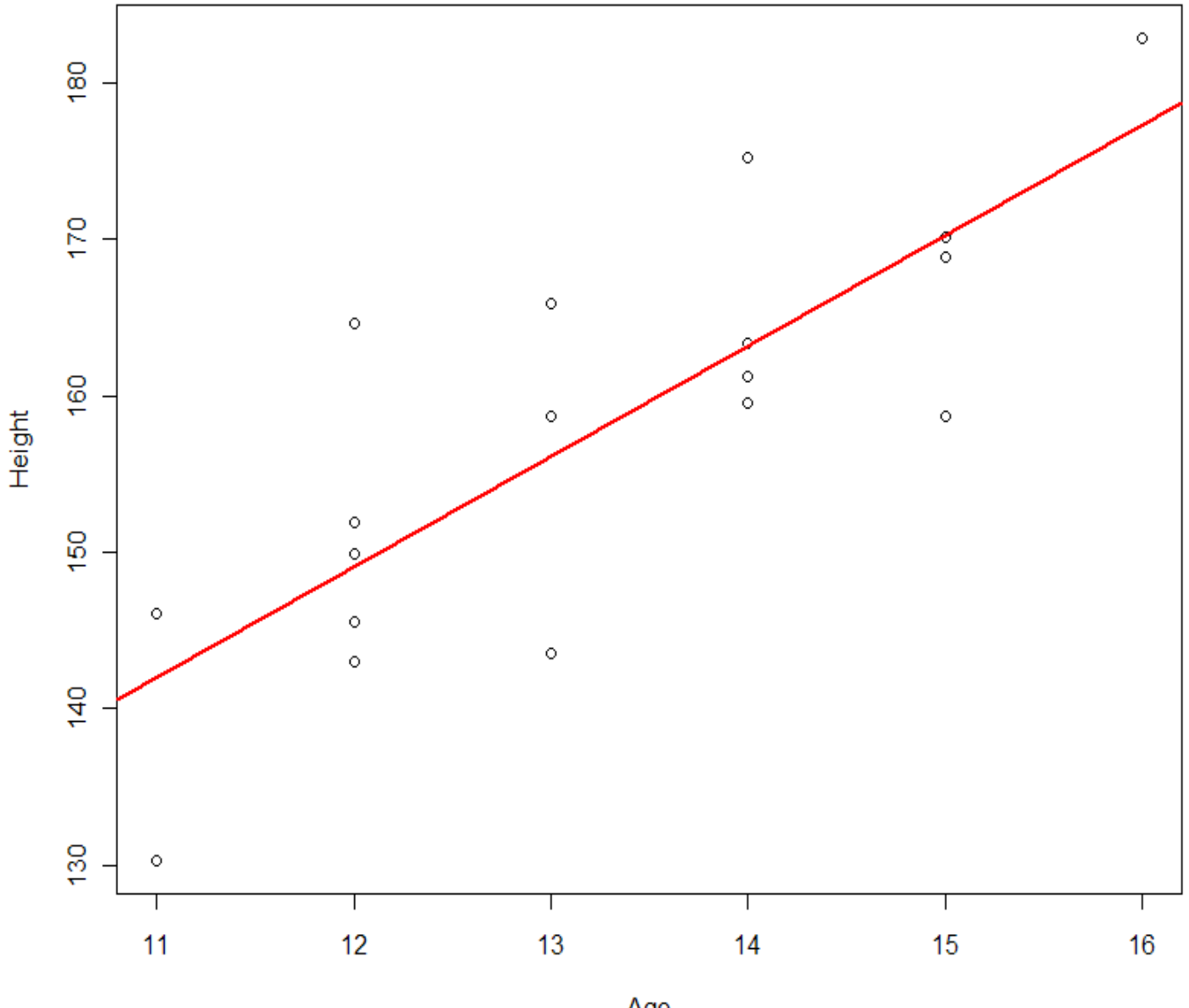

Gender
Age
Height
Weight
11 12 13 14 15 16
30 40 50 60
1.0 1.4 1.8
11 12 13 14 15 16
130 150 170
30 40 50 60
1.0 1.2 1.4 1.6 1.8 2.0
130 140 150 160 170 180

Fitting the linear model in R

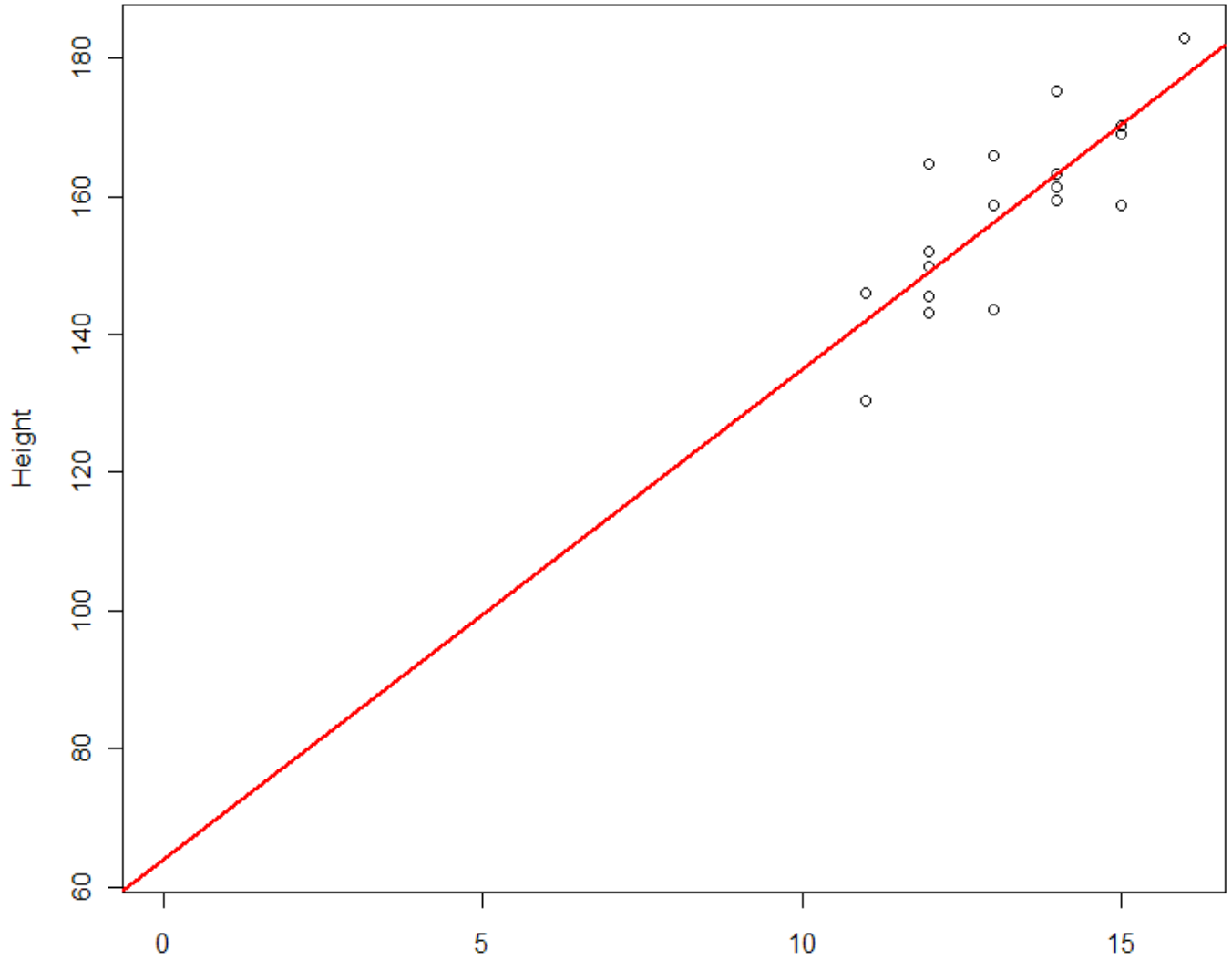
```
> model <- lm( Height ~ Age )
> model

Call:
lm(formula = Height ~ Age)

Coefficients:
(Intercept)          Age
      64.07         7.08
```

Model: Height = 64.07 + 7.08 x Age

```
> plot( Age, Height )
> abline(model, col="red", lwd=2)
```



```
>plot(Age, Height,
xlim=range(0,Age), ylim=range(coef(model)[1], Height))
>abline(model, col="red", lwd=2)
```



Example of summary results of the `lm` command in R

```
> summary( lm( Height ~ Age) )

Call:
lm(formula = Height ~ Age)

Residuals:
      Min        1Q    Median        3Q       Max
-12.59000  -3.57300  -0.07867   3.49000  15.57133

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
```

Example of summary results of the `lm` command in R

```
> summary( lm( Height ~ Age) )
```

Function call

```
Call:
lm(formula = Height ~ Age)

Residuals:
      Min        1Q    Median        3Q       Max
-12.59000  -3.57300  -0.07867   3.49000  15.57133

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
```

Example of summary results of the `lm` command in R

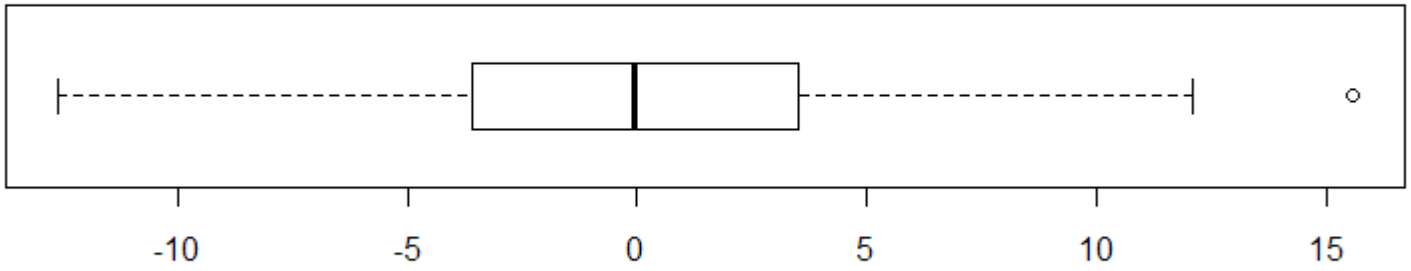
```
> summary( lm( Height ~ Age) )

Call:
lm(formula = Height ~ Age)

Residuals:
      Min        1Q    Median        3Q       Max
-12.59000  -3.57300  -0.07867   3.49000  15.57133

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
```

Five-number summary of the residuals (but no mean – why ?), equivalent to

```
> fivenum( residuals( model ) )
      8      11      17       4       7
-12.590  -3.573  -0.078   3.490  15.571
```

or, graphically, using a boxplot:

```
> boxplot( residuals ( model), horizontal=T)
```



Example of summary results of the `lm` command in R

```
> summary( lm( Height ~ Age) )

Call:
lm(formula = Height ~ Age)

Residuals:
      Min        1Q    Median        3Q       Max
-12.59000  -3.57300  -0.07867   3.49000  15.57133

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
```

Coefficients

These statistical tests tell us if the parameters are significantly different from 0.

****It is not interesting for the intercept, but usually interesting for the slope.**

Estimate and Std. Error are obtained from the matrices of the model.

T-value = Estimate / Std. Error

This assumes that the residuals follow a normal distribution !

Example of summary results of the `lm` command in R

```
> summary( lm( Height ~ Age) )

Call:
lm(formula = Height ~ Age)

Residuals:
      Min        1Q    Median        3Q       Max
-12.59000  -3.57300  -0.07867   3.49000  15.57133

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
```

RSE (Residual Standard Error) and degrees of freedom

The number of *degrees of freedom* indicates the number of independant pieces of data that are available to estimate the error

While we have 19 residuals here, they are not all independent: for example, the last one is constrained because the sum of all residuals must be 0.

The number of DF

total observations – number of parameters estimated

Two parameters are estimated (intercept + coefficient), so 19-2 = 17

Example of summary results of the `lm` command in R

```
> summary( lm( Height ~ Age) )

Call:
lm(formula = Height ~ Age)

Residuals:
      Min        1Q    Median        3Q       Max
-12.59000  -3.57300  -0.07867   3.49000  15.57133

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
```

RSE (Residual Standard Error) and degrees of freedom

The residual standard error is the standard deviation of the residuals (which we would usually like to be small)

It is not exactly equal to what the sd command would return:

```
> sd(residuals(model))
[1] 7.611075
> sqrt(sum(residuals(model)^2)/18)
[1] 7.611075
```

Here, we must divide by the number of degrees of freedom to get the same number:

```
> sqrt(sum(residuals(model)^2)/17)
[1] 7.831732
```

Example of summary results of the `lm` command in R

```
> summary( lm( Height ~ Age) )

Call:
lm(formula = Height ~ Age)

Residuals:
      Min        1Q    Median        3Q       Max
-12.59000  -3.57300  -0.07867   3.49000  15.57133

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
```

Multiple and adjusted R-squared

R^2 is the proportion of the total variance in the response data that is explained by the model

if R^2=1, the data fits perfectly on a straight line, and the model explains all the variance

Multiple and adjusted R-squared

R^2 is the proportion of the total variance in the response data that is explained by the model

if R^2=1, the data fits perfectly on a straight line, and the model explains all the variance

In the case of simple regression, it is equal to the square of the correlation coefficient between the two variables:

```
> summary(model)$r.squared
[1] 0.6584257
> cor(Age, Height)^2
[1] 0.6584257
```

Multiple and adjusted R-squared

R^2 is the proportion of the total variance in the response data that is explained by the model

if R^2=1, the data fits perfectly on a straight line, and the model explains all the variance

In the case of simple regression, it is equal to the square of the correlation coefficient between the two variables:

```
> summary(model)$r.squared
[1] 0.6584257
> cor(Age, Height)^2
[1] 0.6584257
```

The Adjusted R-squared is similar to R-squared, but it takes into account the number of variables in the model (we will come back to this later).

Example of summary results of the `lm` command in R

```
> summary( lm( Height ~ Age) )

Call:
lm(formula = Height ~ Age)

Residuals:
      Min        1Q    Median        3Q       Max
-12.59000  -3.57300  -0.07867   3.49000  15.57133

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
```

F-test for significance of regression

The F-statistic allows us to test if the whole regression (adding all variables *vs* having only the intercept in) is significant.

Note: With only one variable, it provides *exactly* the same result as the t-test for the significance of the coefficient of this variable.

Multiple regression:
assessing the effect of several variables *together*

R Vs RevoScaleR

```
#Initialize some variables to specify the data sets.
inputFileClass <-
paste0("/media/sf_docVM/correlationregression/","class.csv")

#Import the  data.
class_data<- rxImport(inData = inputFileClass)
```

R

```
#call lm
lm_class_basicR<-lm(
      formula= Height~Age,
      data =class_data[,-1])

#summary of lm output
summary(lm_class_basicR)
```

RevoScaleR

```
#call lm
lm_class<-rxLinMod(
      formula= Height~Age,
      data =class_data[,-1])

#summary of lm output
summary(lm_class)
```

Challenge

Investigate the correlation and the relationship between weight and age using R basic commands and RevoScaleR

Challenge: Solution

Investigate the correlation and the relationship between weight and height using R basic commands and RevoScaleR

R

```
#call lm
lm_class_basicR<-lm(
       formula= Height~Weight,
       data =class_data[,-1])

#summary of lm output
summary(lm_class_basicR)
```

RevoScaleR

```
#call lm
lm_class<-rxLinMod(
       formula= Height~Weight,
       data =class_data[,-1])

#summary of lm output
summary(lm_class)
```

What happens if both,
age and weight variables
were included in the same model ?

One multiple regression with two variables

```
Call:
lm(formula = Height ~ Age + Weight)

Residuals:
     Min       1Q   Median       3Q      Max
-9.20695 -3.30604 -0.04478  2.11432 10.41880

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 81.77355   12.90896   6.335 9.92e-06 ***
Age          3.11575    1.34668   2.314  0.03431 *
Weight       0.35064    0.08827   3.973  0.00109 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.728 on 16 degrees of freedom
Multiple R-squared: 0.828,      Adjusted R-squared: 0.8065
F-statistic: 38.52 on 2 and 16 DF,  p-value: 7.646e-07
```

This model allows us to determine the respective contribution of each variable separately.

Coefficients

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 81.77355   12.90896   6.335 9.92e-06 ***
Age          3.11575    1.34668   2.314  0.03431 *
Weight       0.35064    0.08827   3.973  0.00109 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

This is similar to the simple regression case.

Each test is conducted assuming that the tested parameter is the last one entering the model:

« If *weight* is already in the model, is the coefficient for *age* significantly different from 0 ? »

Two single regressions vs one multiple regression

```
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 ***
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 108.12816    6.80692  15.885 1.24e-11 ***
Weight        0.50194    0.06644   7.555 7.89e-07 ***
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  81.77355   12.90896   6.335 9.92e-06 ***
Age           3.11575    1.34668   2.314  0.03431 *
Weight        0.35064    0.08827   3.973  0.00109 **
```

While both age and weight seem significant by themselves, age is much less significant when weight is already included (see also the R^2).

It is likely that a lot of the information provided by the age is also provided by the weight, so that there may be little need to have both terms in the model.

Multiple and adjusted R-squared

```
Multiple R-squared: 0.828,      Adjusted R-squared: 0.8065
```

As before, R^2 is the proportion of the total variance in the response data that is explained by the model.

Adding a new variable in the model will always increase R^2, up to 1 when there the number of degrees of freedom is 0 (number of parameters to estimate = number of observations).

Multiple and adjusted R-squared

```
Multiple R-squared: 0.828,      Adjusted R-squared: 0.8065
```

The adjusted R-squared adjusts for the number of variables in the model, and does not necessarily increase when the number of variables increase; it can even be negative.

It is always equal or below R^2.

Example

```
y <- rnorm(10)
x1 <- rnorm(10); x2 <- rnorm(10); … ; x9 <-
rnorm(10)
summary(lm(y ~ x1)); summary(lm(y ~ x1+x2));
```

```
 1: Multiple R-squared: 0.1419,     Adjusted R-squared: 0.03464
 2: Multiple R-squared: 0.5173,     Adjusted R-squared: 0.3794
 3: Multiple R-squared: 0.557,      Adjusted R-squared: 0.3355
 4: Multiple R-squared: 0.5577,     Adjusted R-squared: 0.2039
 5: Multiple R-squared: 0.7953,     Adjusted R-squared: 0.5395
 6: Multiple R-squared: 0.8321,     Adjusted R-squared: 0.4962
 7: Multiple R-squared: 0.984,      Adjusted R-squared: 0.9281
 8: Multiple R-squared: 0.9851,     Adjusted R-squared: 0.866
 9: Multiple R-squared:     1,      Adjusted R-squared:   NaN
```

The last regression from the example

```
Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9)

Residuals:
ALL 10 residuals are 0: no residual degrees of freedom!

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.02693         NA      NA       NA
x1           0.53886         NA      NA       NA
x2          -0.52227         NA      NA       NA
x3           0.51881         NA      NA       NA
x4           0.74757         NA      NA       NA
x5           0.14394         NA      NA       NA
x6          -0.65387         NA      NA       NA
x7          -0.48271         NA      NA       NA
x8          -0.62487         NA      NA       NA
x9           0.23759         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:     1,      Adjusted R-squared:   NaN
F-statistic:   NaN on 9 and 0 DF,  p-value: NA
```

F-statistic for significance of regression

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 81.77355   12.90896   6.335 9.92e-06 ***
Age          3.11575    1.34668   2.314  0.03431 *
Weight       0.35064    0.08827   3.973  0.00109 **

F-statistic: 38.52 on 2 and 16 DF,  p-value: 7.646e-07
```

Again, the F-statistic allows us to test if the whole regression (adding all variables *vs* having only the intercept in) is significant.

If any of the tests for the individual variables is significant, the F-test will generally be significant as well.

However, even if no individual variable is significant (e.g. $p < 0.05$), the F-test can still be significant.

Categorical variables, dummy variables and contrasts

Categorical variables

We'd like to use categorical variables in a linear model, as in:

$$\textbf{Height} = b_0 + b_1\,\textbf{Age} + b_2\,\textbf{« Gender »} + \textbf{error}$$

Intuitively, we want to estimate a « Male » and a « Female » effect.

Categorical variables

We'd like to use categorical variables in a linear model, as in:

$$\textbf{Height} = b_0 + b_1\,\textbf{Age} + b_2\,\textbf{« Gender »} + \textbf{error}$$

Intuitively, we want to estimate a « Male » and a « Female » effect.

In practice, categorical variables (factors in R) are turned (by default, based on alphabetical order) into **dummy variables** of the form

$$\text{Gender} = \begin{cases} 0 \text{ if } \textbf{F}\text{emale} \\ 1 \text{ if } \textbf{M}\text{ale} \end{cases}$$

Example of summary results of the `lm` command in R

```
Call:
lm(formula = Height ~ Age + Gender)

Residuals:
    Min      1Q  Median      3Q     Max
-8.8462 -4.8523 -0.8102  3.3677 13.5058

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   62.291     14.957   4.165  0.00073 ***
Age            6.928      1.117   6.202 1.27e-05 ***
GenderM        7.204      3.251   2.216  0.04152 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.061 on 16 degrees of freedom
Multiple R-squared: 0.7387,     Adjusted R-squared: 0.706
F-statistic: 22.61 on 2 and 16 DF,  p-value: 2.176e-05
```

Example of summary results of the `lm` command in R

```
Call:
lm(formula = Height ~ Age + Gender)

Residuals:
    Min      1Q  Median      3Q     Max
-8.8462 -4.8523 -0.8102  3.3677 13.5058

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   62.291     14.957   4.165  0.00073 ***
Age            6.928      1.117   6.202 1.27e-05 ***
GenderM        7.204      3.251   2.216  0.04152 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.061 on 16 degrees of freedom
Multiple R-squared: 0.7387,     Adjusted R-squared: 0.706
F-statistic: 22.61 on 2 and 16 DF,  p-value: 2.176e-05
```

baseline for height among Female

Example of summary results of the `lm` command in R

```
Call:
lm(formula = Height ~ Age + Gender)

Residuals:
    Min      1Q  Median      3Q     Max
-8.8462 -4.8523 -0.8102  3.3677 13.5058

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    62.291     14.957   4.165  0.00073 ***
Age             6.928      1.117   6.202 1.27e-05 ***
GenderM         7.204      3.251   2.216  0.04152 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.061 on 16 degrees of freedom
Multiple R-squared: 0.7387,     Adjusted R-squared: 0.706
F-statistic: 22.61 on 2 and 16 DF,  p-value: 2.176e-05
```

baseline for height among Female

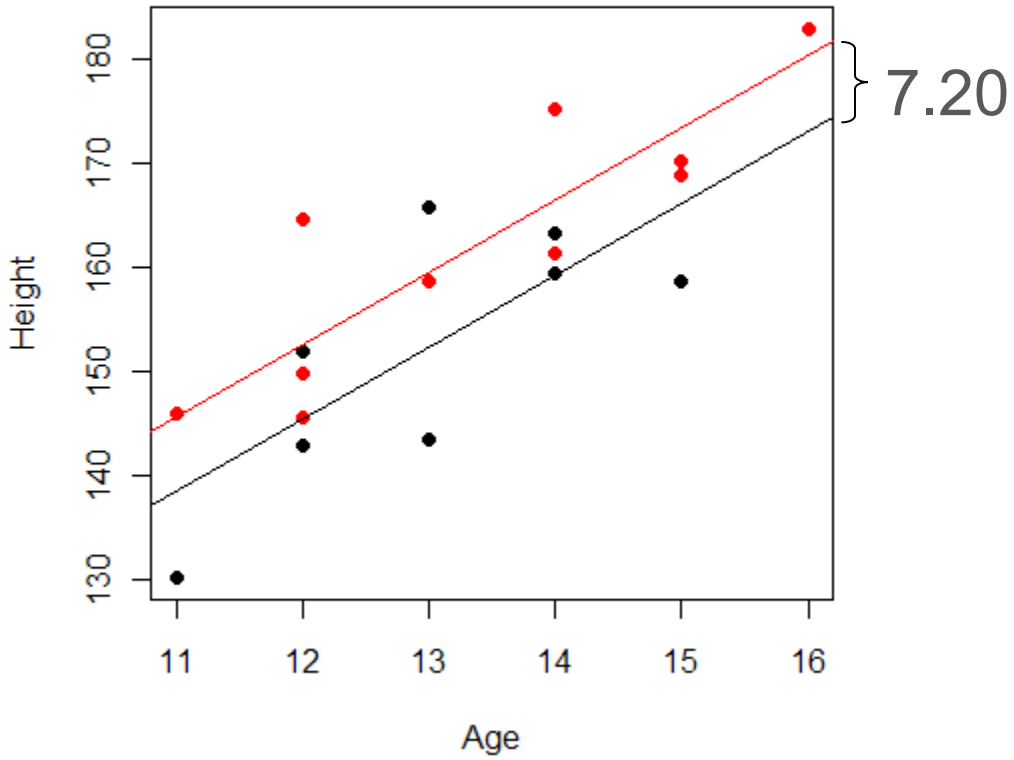
The factor GenderM corresponds to the difference in baseline for Males compared to females.

Graphical interpretation

The model specifies 2 straight lines, with the same slope but different y-intercepts:

For women: Height = 62.3 + 6.9 Age (in black)

For men: Height = 69.4 + 6.9 Age (in red)



We could also compute the difference in means between males and females directly:

```
> means <- tapply( data$Height, data$Gender, FUN=mean )
> means
       F        M
153.8958 162.3314
> diff(means)
       M
8.435622
```

This result is slightly different from the 7.20 cm difference found with the linear model.

Where does the difference come from ?

Interactions

So far, we have assumed a difference between the lines, but the same slope; that is, for both men and women, the effect of age is the same.

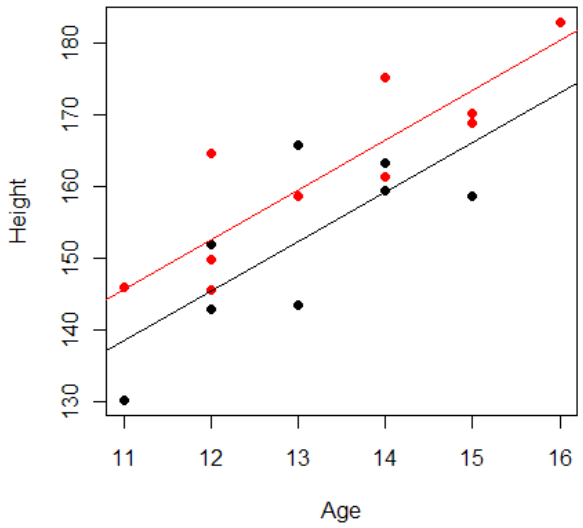
If this assumption is incorrect, it means that there is an *interaction* between the factors « age » and « gender », that is, the effect of age is different depending on the gender.

Interactions are modeled in R in the following way:

```
lm(formula = Height ~ Age + Gender + Age:Gender)
```

which is equivalent to

```
lm(formula = Height ~ Age * Gender)
```



Coefficients with an interaction

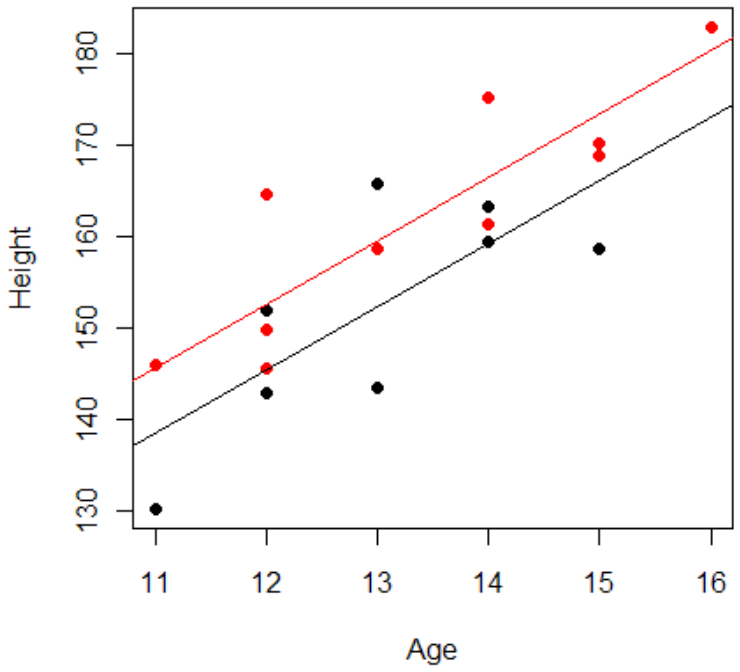
```
Call:
lm(formula = Height ~ Age * Gender)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  56.2610    24.4880   2.297  0.03640 *
Age           7.3841     1.8429   4.007  0.00114 **
GenderM      17.1304    31.5238   0.543  0.59483
Age:GenderM  -0.7468     2.3583  -0.317  0.75585
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The coefficients can be interpreted as follows:

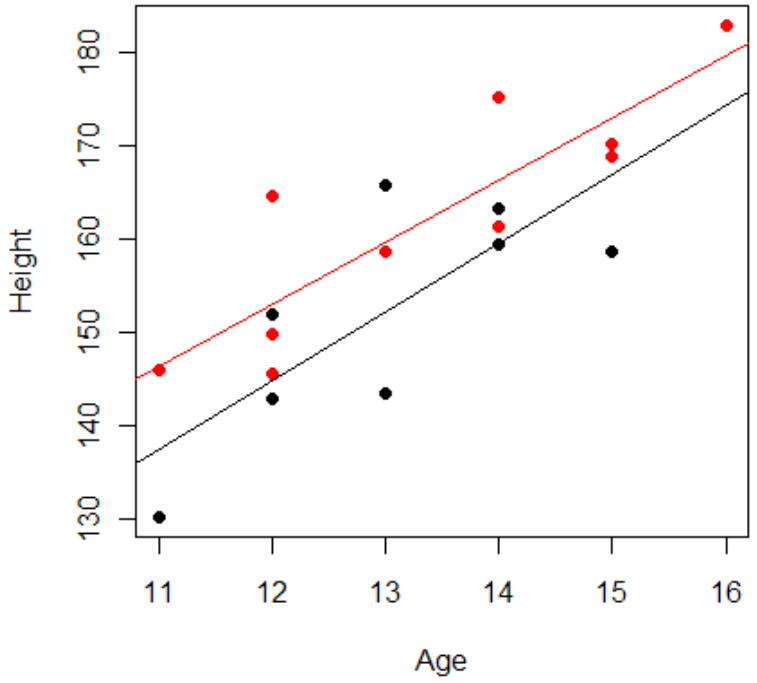
According to the model, the *height* is equal to

56.26 (the intercept)
plus 17.13, but only for males
plus 7.38 times the person’s age
minus 0.75 times the person’s age, but only for males.

Different slopes



No interaction



With interaction

What if Males were the baseline ?

```
Call:
lm(formula = Height ~ Age + Gender)

Residuals:
    Min      1Q  Median      3Q     Max
-8.8462 -4.8523 -0.8102  3.3677 13.5058

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   62.291     14.957   4.165  0.00073 ***
Age            6.928      1.117   6.202 1.27e-05 ***
GenderM        7.204      3.251   2.216  0.04152 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.061 on 16 degrees of freedom
Multiple R-squared: 0.7387,     Adjusted R-squared: 0.706
F-statistic: 22.61 on 2 and 16 DF,  p-value: 2.176e-05
```

```
Call:
lm(formula = Height ~ Age + Gender1)

Residuals:
    Min      1Q  Median      3Q     Max
-8.8462 -4.8523 -0.8102  3.3677 13.5058

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   69.495     15.135   4.592 0.000301 ***
Age            6.928      1.117   6.202 1.27e-05 ***
Gender1F      -7.204      3.251  -2.216 0.041517 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.061 on 16 degrees of freedom
Multiple R-squared: 0.7387,     Adjusted R-squared: 0.706
F-statistic: 22.61 on 2 and 16 DF,  p-value: 2.176e-05
```

The two models are exactly the same; only the way we look at the coefficient changes.

```
Gender1 <- relevel(Gender, ref="M")
```

R Vs RevoScaleR

R

```
# lm using basic R
lm_gender<-lm(
       formula=Height~Age+Gender,
       data =class_data)
summary(lm_gender)
```

RevoScaleR

```
# lm using RevoScaleR
# not working
>lm_class_gender<-rxLinMod(
       formula= Height~Age+Gender,
       data =class_data)
# working
>recodedDF2 <- rxFactors(inData
= class_data, sortLevels =
TRUE,factorInfo = c("Gender"))
rxGetVarInfo(recodedDF2)
>lm_class_gender<-rxLinMod(
       formula= Height~Age+Gender,
       data =recodedDF2)
>summary(lm_class_gender)
```

```
Call:
lm(formula = Height ~ Age + Gender, data = recodedDF2)

Residuals:
   Min     1Q Median     3Q    Max
-3.483 -1.910 -0.319  1.326  5.317

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  24.5241     5.8886   4.165 0.000731 ***
Age           2.7276     0.4398   6.202 1.27e-05 ***
GenderM       2.8362     1.2797   2.216 0.041517 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.78 on 16 degrees of freedom
Multiple R-squared:  0.7387,  Adjusted R-squared:  0.706
F-statistic: 22.61 on 2 and 16 DF,  p-value: 2.176e-05
```

```
Call:
rxLinMod(formula = Height ~ Age + Gender, data = recodedDF2)

Linear Regression Results for: Height ~ Age + Gender
Data: recodedDF2
Dependent variable(s): Height
Total independent variables: 4 (Including number dropped: 1)
Number of valid observations: 19
Number of missing observations: 0

Coefficients: (1 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  27.3603     5.9587   4.592 0.000301 ***
Age           2.7276     0.4398   6.202 1.27e-05 ***
Gender=F     -2.8362     1.2797  -2.216 0.041517 *
Gender=M     Dropped    Dropped Dropped  Dropped
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.78 on 16 degrees of freedom
Multiple R-squared: 0.7387
Adjusted R-squared: 0.706
F-statistic: 22.61 on 2 and 16 DF,  p-value: 2.176e-05
Condition number: 1.1301
```

Challenge: cheese dataset

As cheddar cheese matures, a variety of chemical processes take place. The taste of matured cheese is related to the concentration of several chemicals in the final product. In a study of cheddar cheese from the LaTrobe Valley of Victoria, Australia, samples of cheese were analyzed for their chemical composition and were subjected to taste tests. Overall taste scores were obtained by combining the scores from several tasters.

Case: Sample number
Taste: Subjective taste test score, obtained by combining the scores of several tasters
Acetic: concentration of acetic acid
H2S: concentration of hydrogen sulfide
Lactic: Concentration of lactic acid

EXERCISE
Which factor(s) influence the taste of cheese?

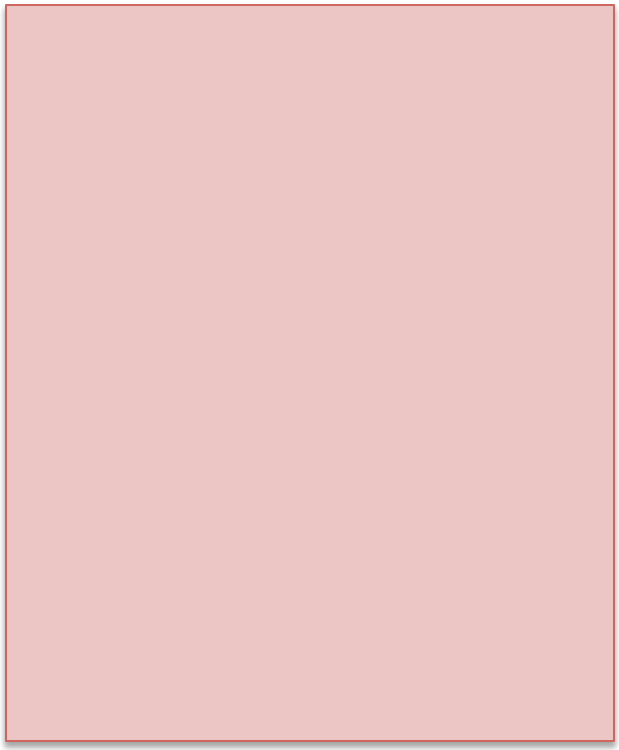
Learning using linear model

Learning using linear model

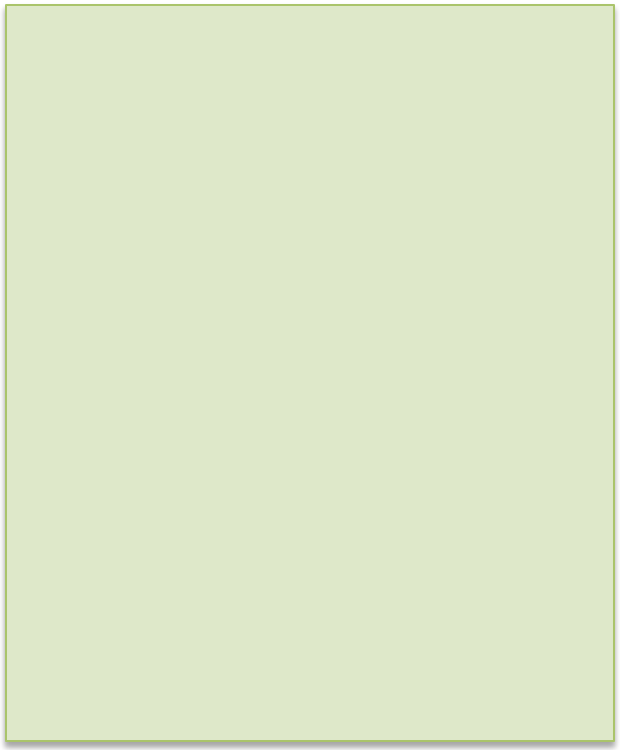
To Learn you need to train

Data

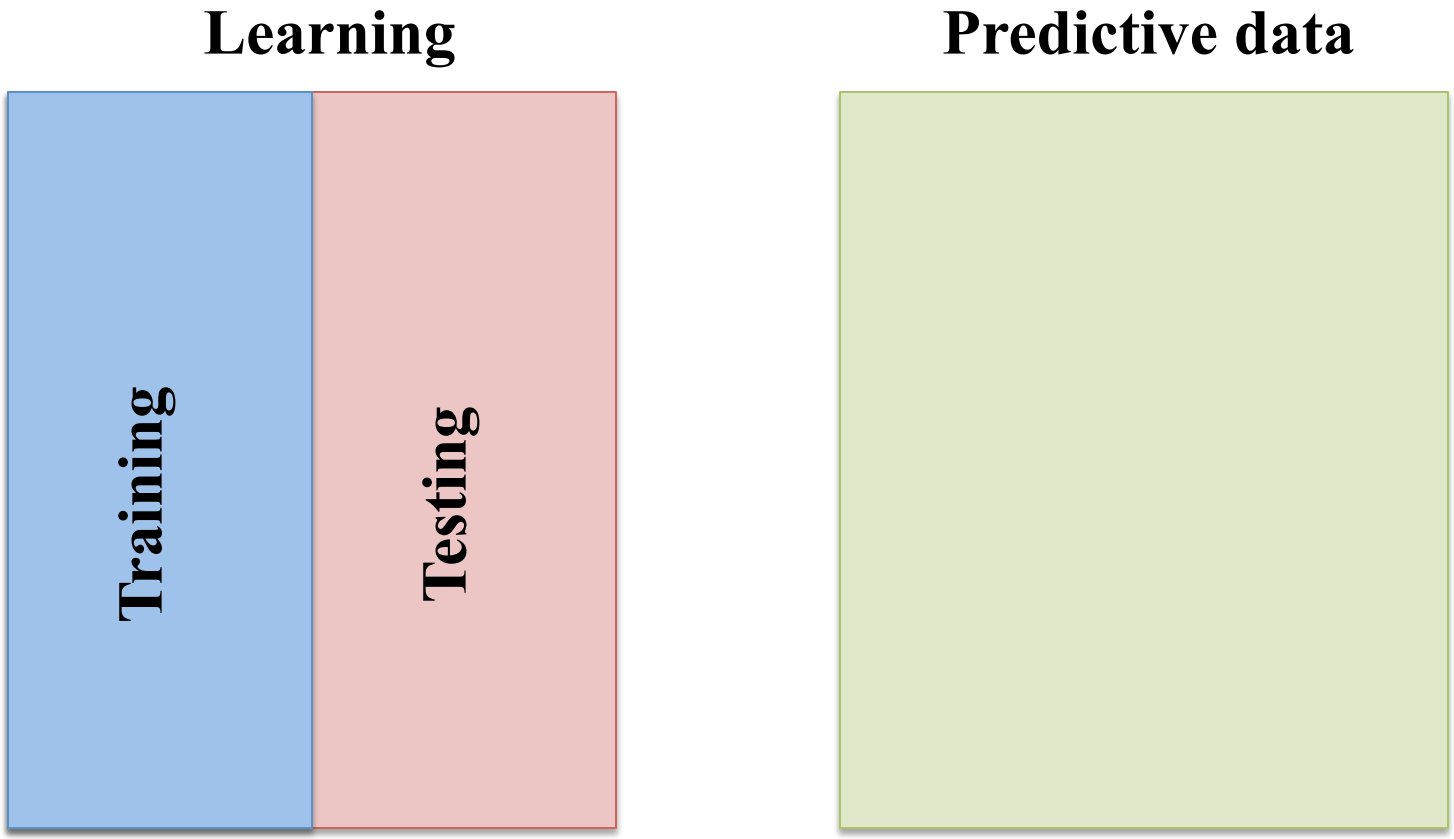
Learning

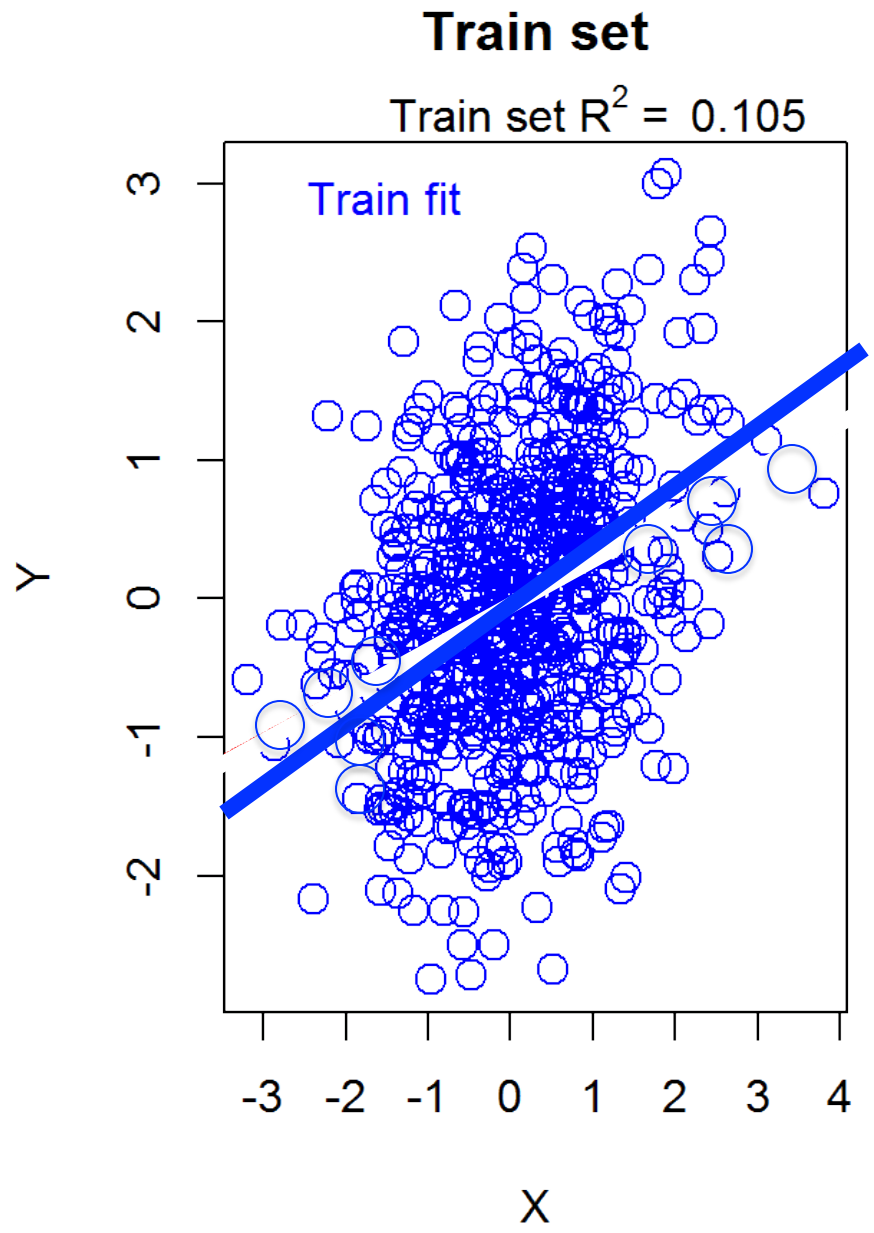


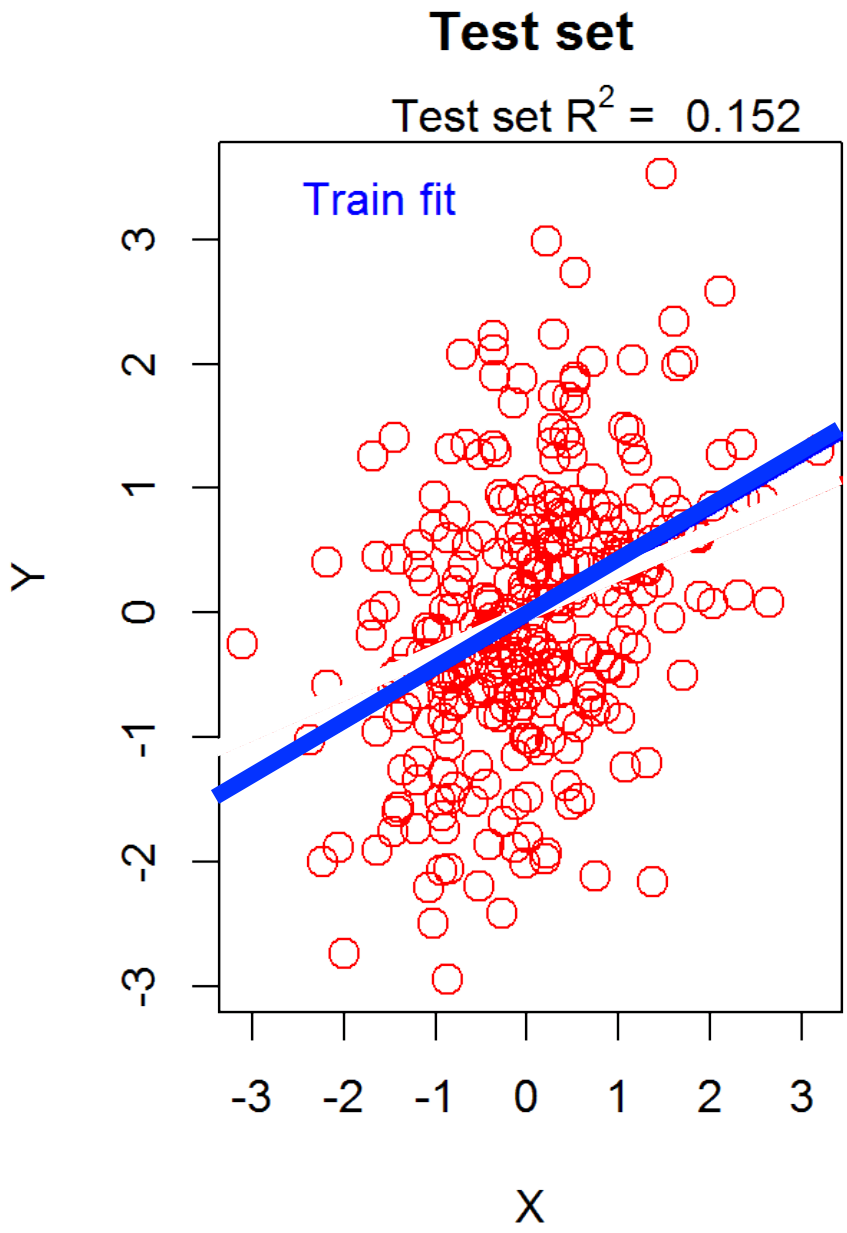
Predictive data

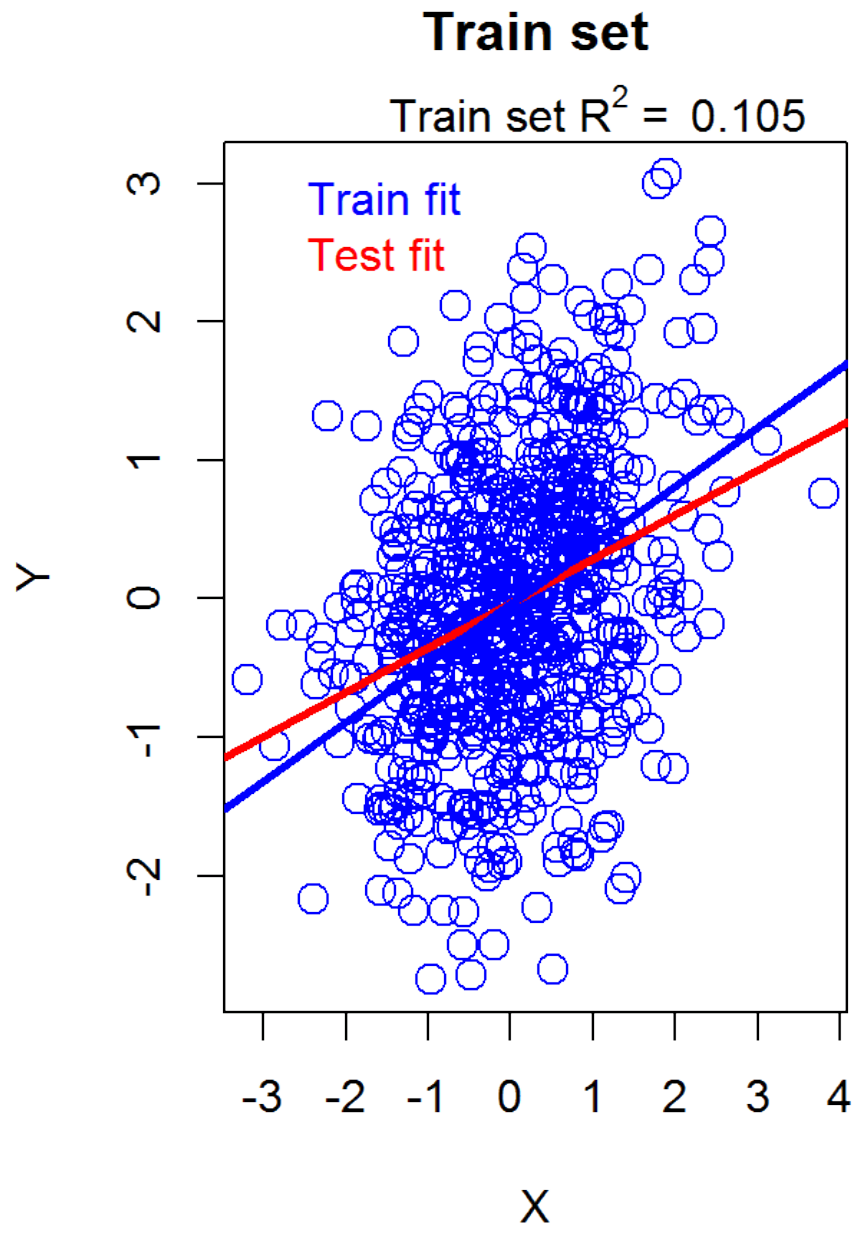


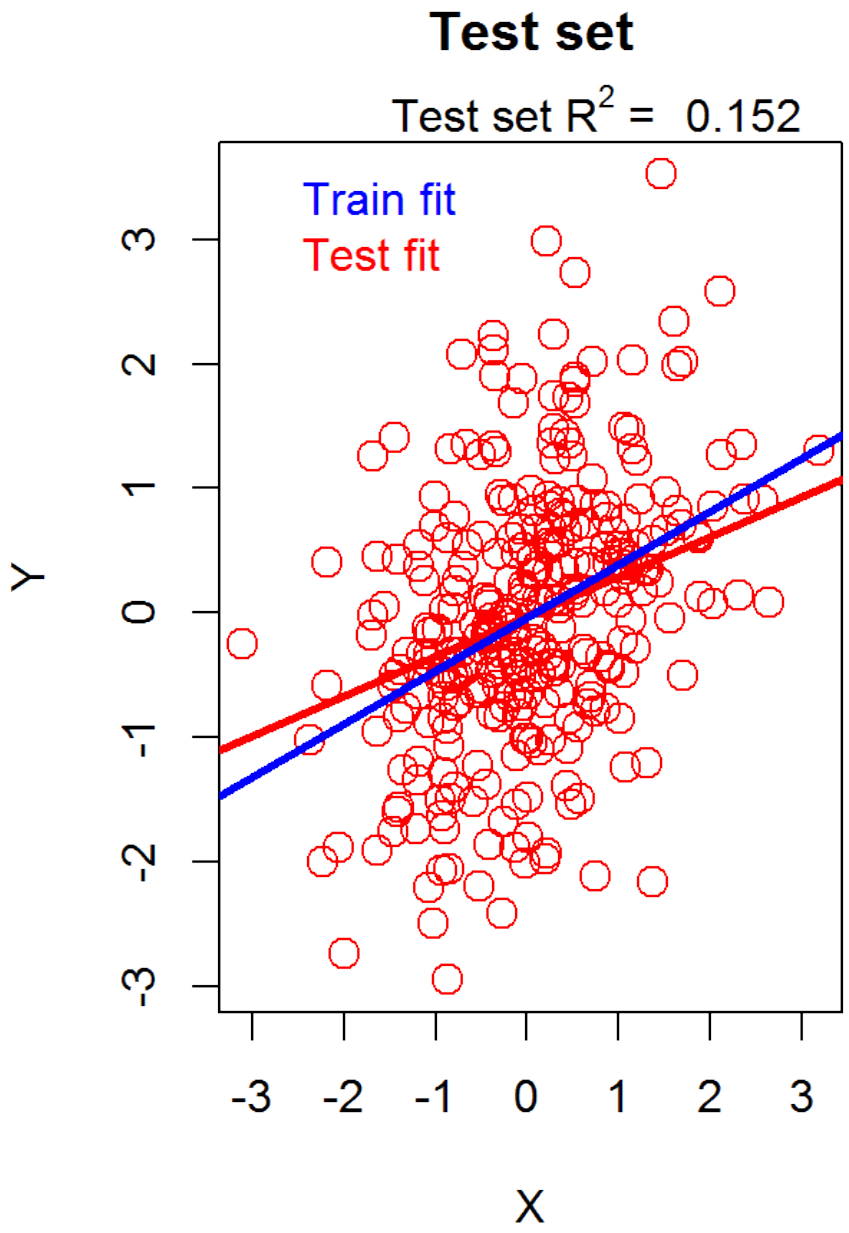
Data

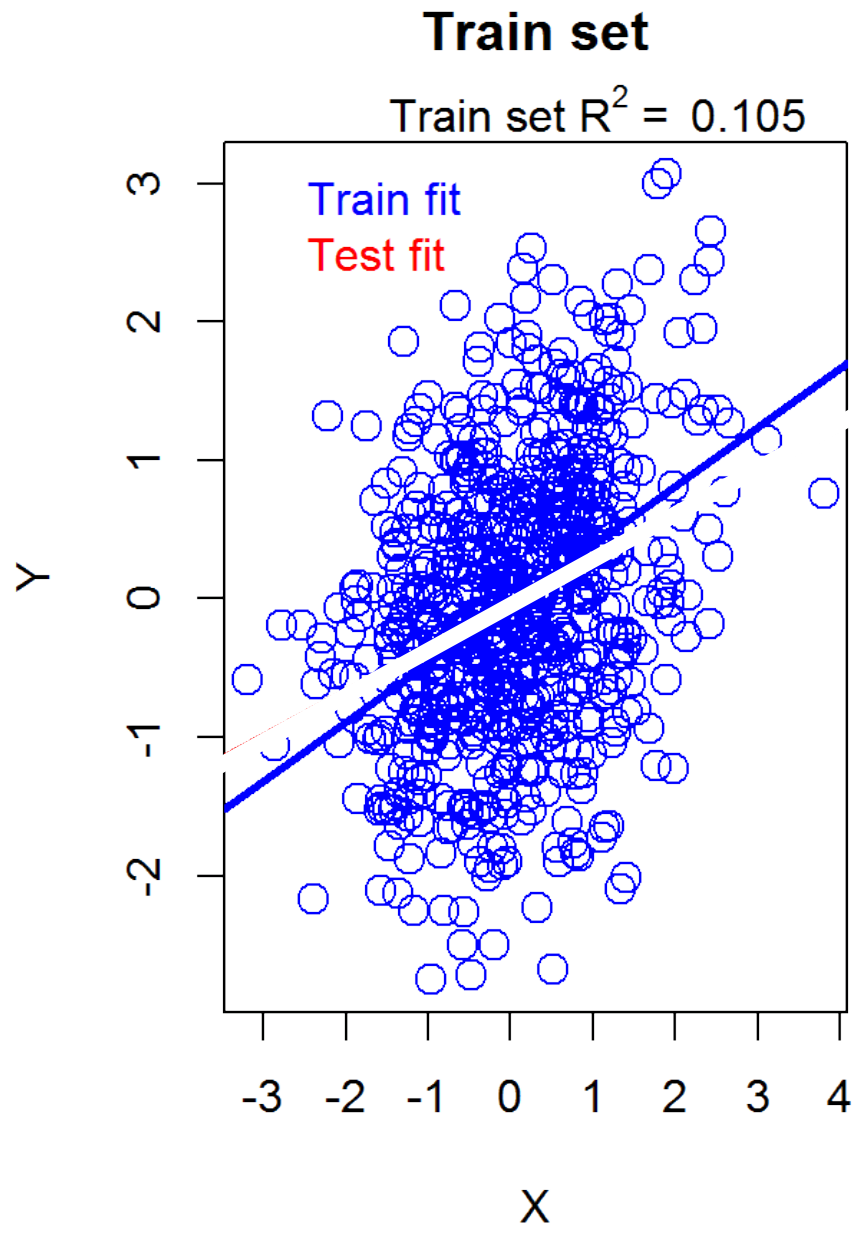


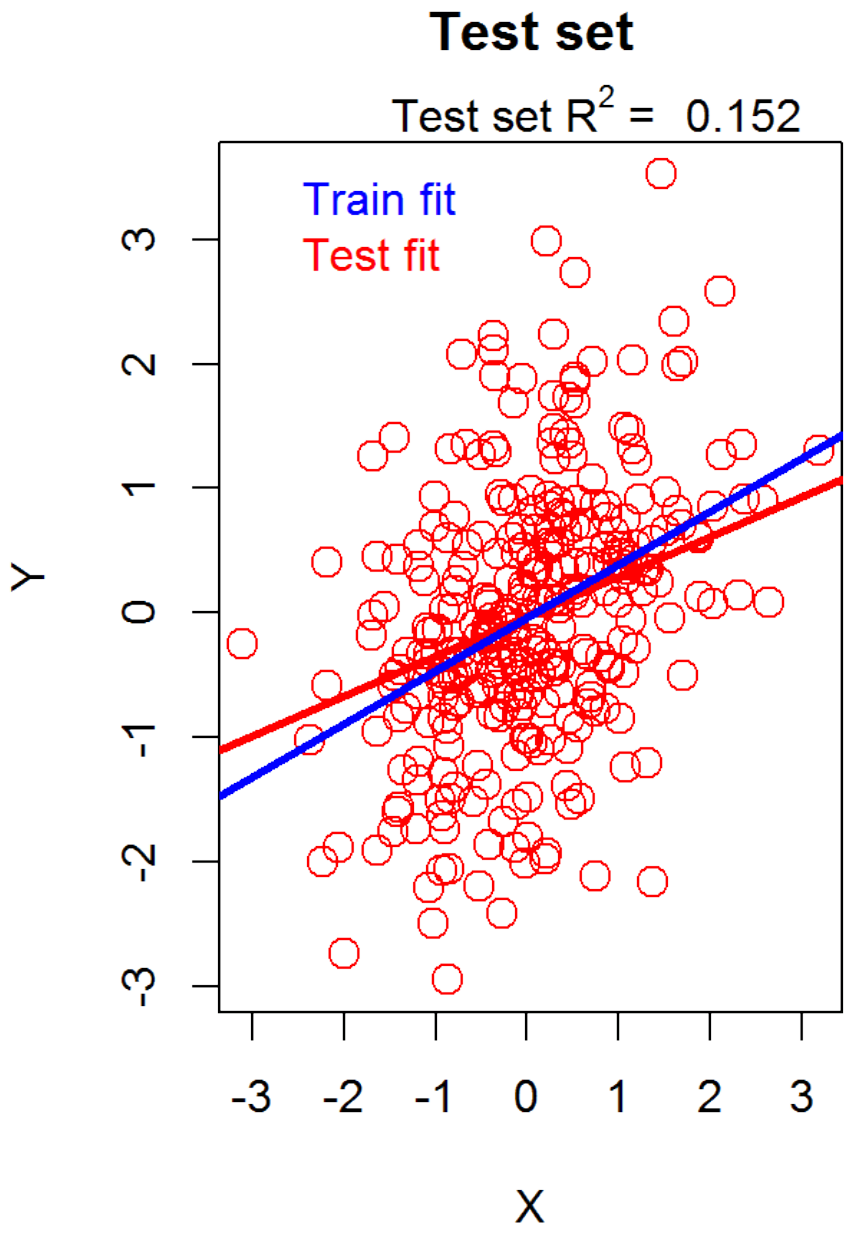

Train set
Train set R^2 = 0.105
Train fit
Y
X
-3 -2 -1 0 1 2 3 4
-2 -1 0 1 2 3

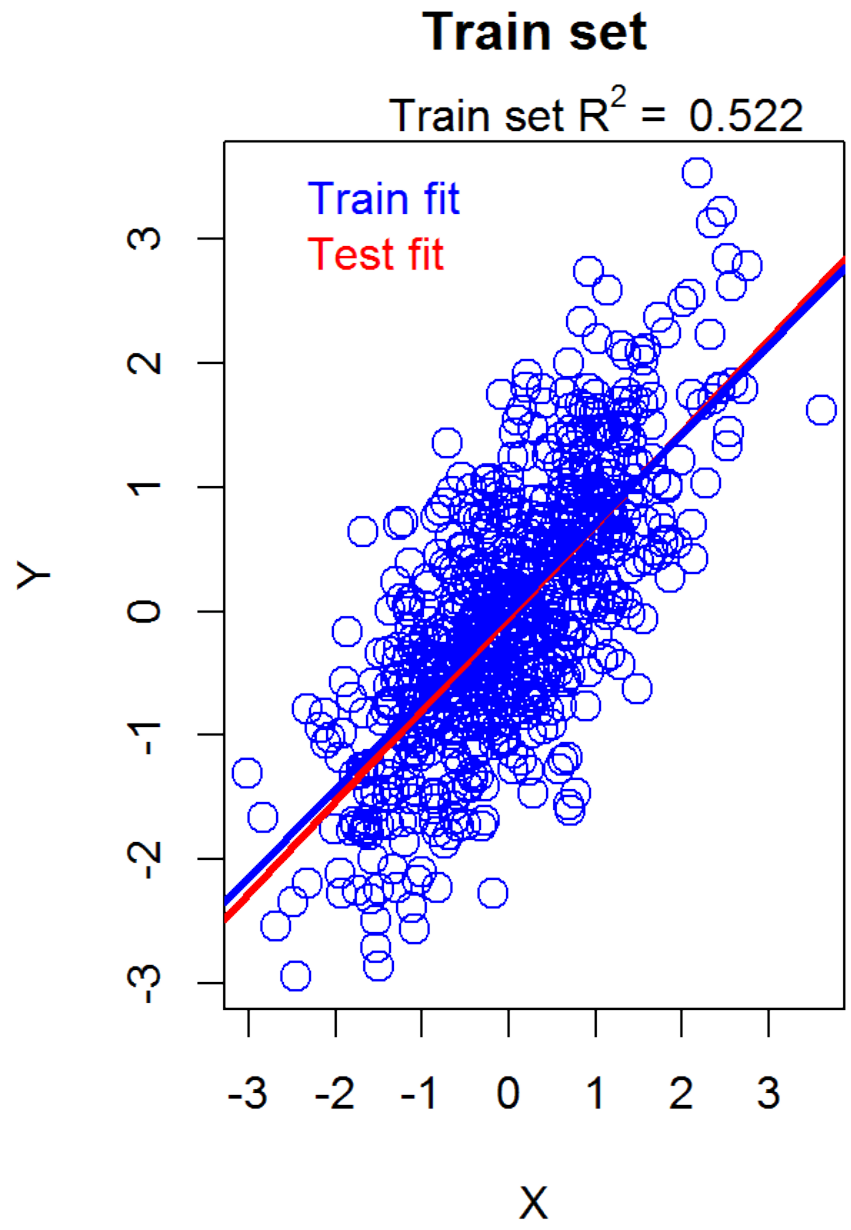

Test set
Test set R^2 = 0.152
Train fit
Y
X
-3 -2 -1 0 1 2 3
-3 -2 -1 0 1 2 3

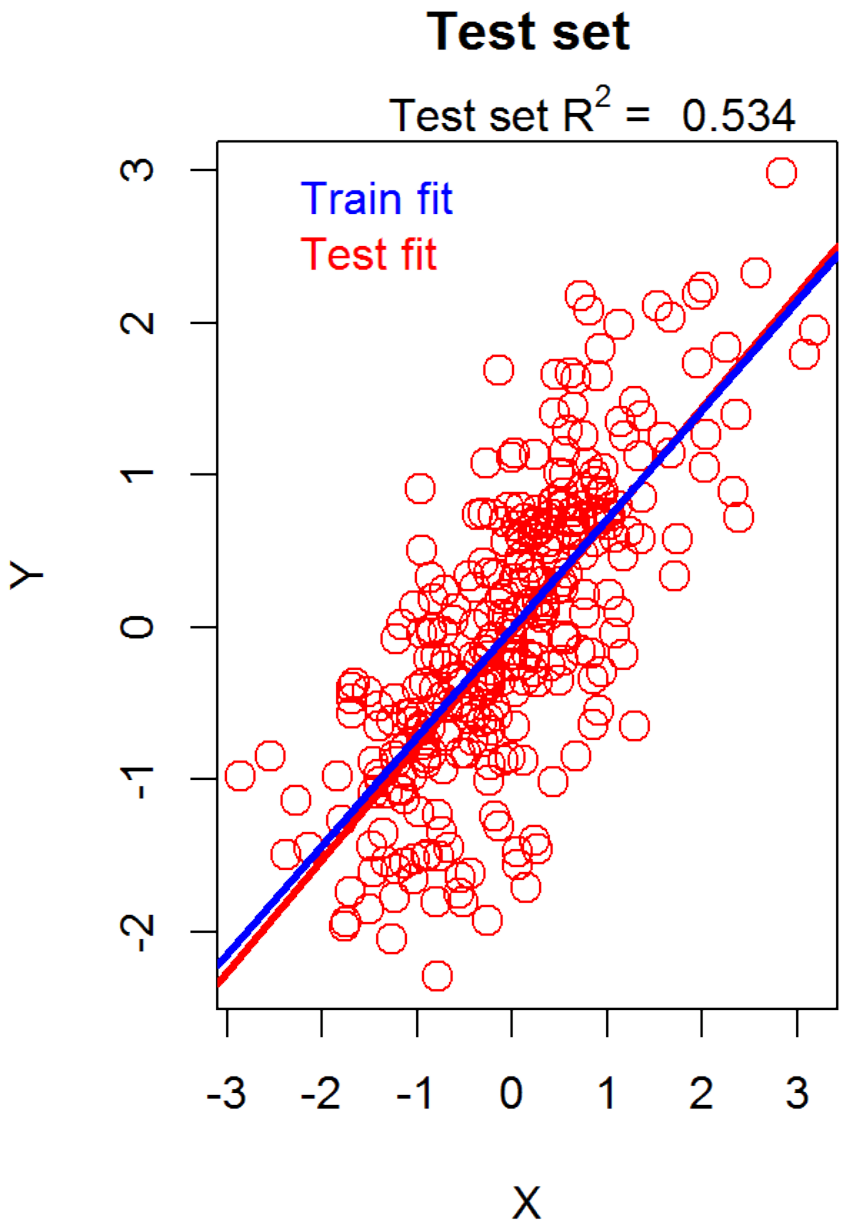

Train set
Train set R^2 = 0.105
Train fit
Test fit
Y
X
-3
-2
-1
0
1
2
3
4

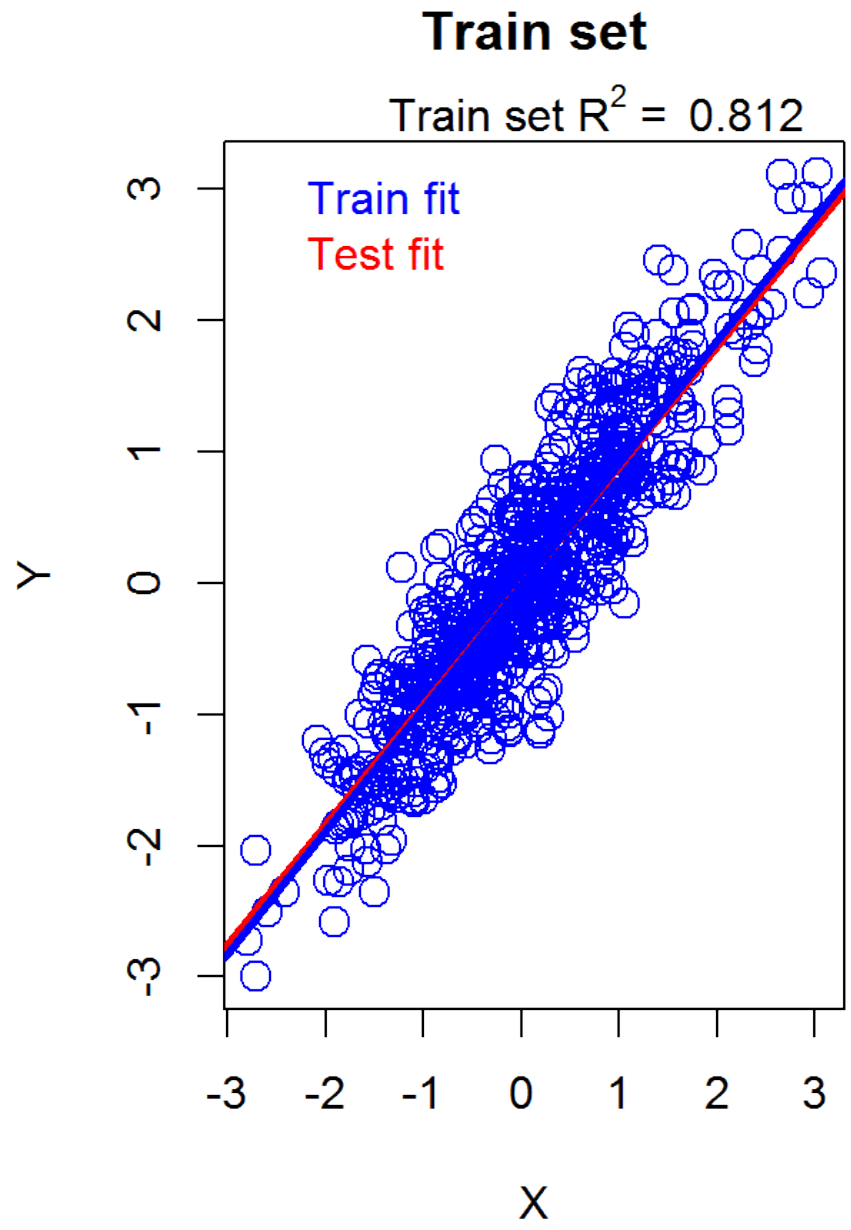

Test set
Test set R^2 = 0.152
Train fit
Test fit
Y
X
-3
-2
-1
0
1
2
3

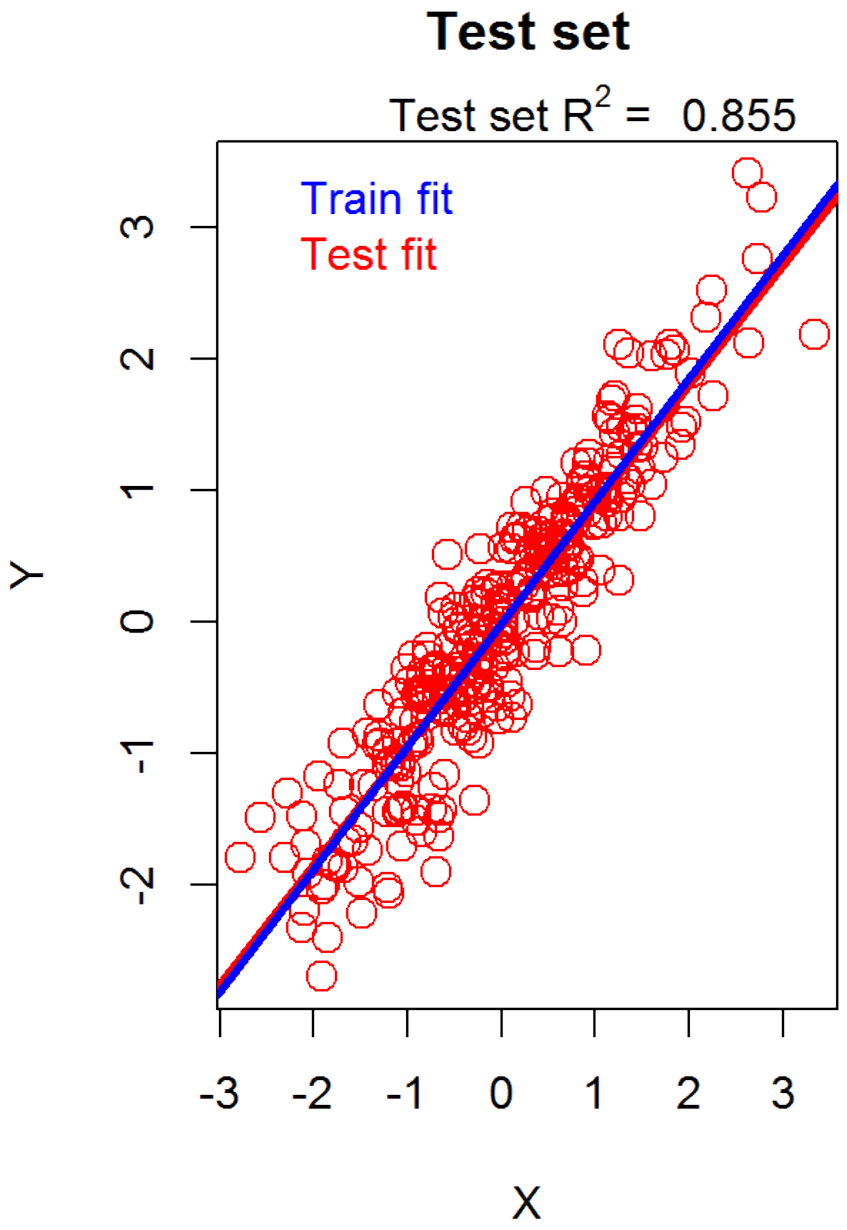

Train set
Train set R^2 = 0.105
Train fit
Test fit
Y
X
-3
-2
-1
0
1
2
3
4


Test set
Test set R^2 = 0.152
Train fit
Test fit
Y
X
-3
-2
-1
0
1
2
3


Train set
Train set R^2 = 0.522
Train fit
Test fit
Y
X
-3
-2
-1
0
1
2
3


Test set
Test set R^2 = 0.534
Train fit
Test fit
Y
X
-3
-2
-1
0
1
2
3


Train set
Train set R^2 = 0.812
Train fit
Test fit
Y
X
-3
-2
-1
0
1
2
3


Test set
Test set R^2 = 0.855
Train fit
Test fit
Y
X
-3
-2
-1
0
1
2
3

R Vs RevoScaleR

R

```
#Split data
>trainData=class[1:15,]
>testData=class[5:19,]

#To train, use:
>model <-
lm(formula = Height~Age+Gender,
data = trainData)

#To test and predict, use:
>predict(model, testData)
>cor(predict(model,
testData),testData$Height)
```

RevoScaleR

```
#Split data
>trainData=recodedDF2[1:15,]
>testData=recodedDF2[5:19,]

#To train, use:
>model<-rxLinMod
(formula= Height~Age+Gender,
data =trainData)

#To test and predict, use:
>rxpredict(modelObject=model,
data=testData)
```

Challenge

Investigate the correlation and the relationship between Weight and age by taking into account the gender using R basic commands and RevoScaleR

Challenge: Solution

Investigate the correlation and the relationship between weight and age using R basic commands and RevoScaleR

```
# lm using RevoScaleR on categorical variables
>recodedDF2 <- rxFactors(inData = class_data, sortLevels =
TRUE,factorInfo = c("Gender"))
>rxGetVarInfo(recodedDF2)
>lm_class_gender<-rxLinMod(
        formula= Weight~Age+Gender,
        data =recodedDF2)
>summary(lm_class_gender)

# basic R
>lm_class_gender_basic<-lm(
        formula=  Weight~Age+Gender,
        data =recodedDF2)
>summary(lm_class_gender_basic)
```

Challenge: Diabetes example

Explore the dataset using
summary statistics, regressions and correlations from RevoScaleR.

```
# Load the data and remove NAs
>data("PimaIndiansDiabetes2", package = "mlbench”)
```

Age ~ Height

Logistic regression

Discrete ~ continuous/discrete

Logistic regression

What is Logistic Regression?

Form of regression that allows the prediction of discrete variables by a mix of continuous and discrete predictors.

Predictors do not have to be

- normally distributed
- linearly related
- have equal variance in each group

Logistic regression is rarely taught because it requires a lot of computational power

Binary Logistic Regression Model

Y = Binary response, ex. Gender (male=1, female=0)

X = Quantitative predictor, ex. height

π = Proportion of success (1, yes, success, male) at any X

Binary Logistic Regression Model

Y = Binary response, ex. Gender (male=1, female=0)

X = Quantitative predictor, ex. height

π = Proportion of success (1, yes, success, male) at any X

Logit form

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X$$

Background

Logit is the natural log of an odds ratio; often called a log odds even though it really is a log odds ratio.

Logits are continuous

p = 0.50, then logit = 0

p = 0.70, then logit = 0.84

p = 0.30, then logit = -0.84

Binary Logistic Regression Model

Y = Binary response, ex. Gender (male=1, female=0)

X = Quantitative predictor, ex. height

π = Proportion of success (1, yes, success, male) at any X

Logit form

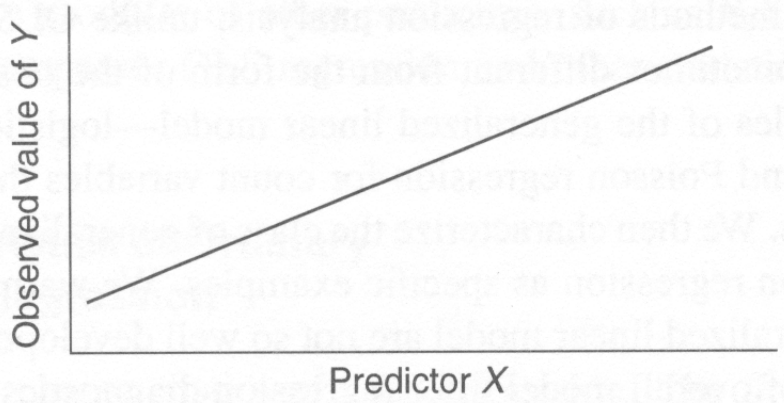
$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X$$

Probability form

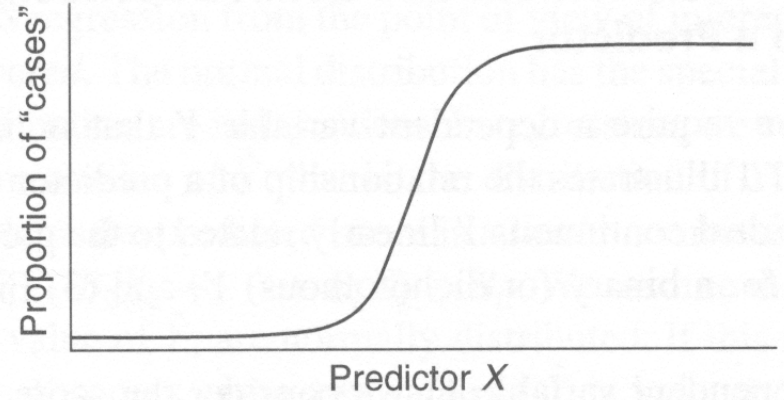
$$\pi = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

The logistic function

(A) For a continuous outcome variable Y, the numerical value of Y at each value of X.

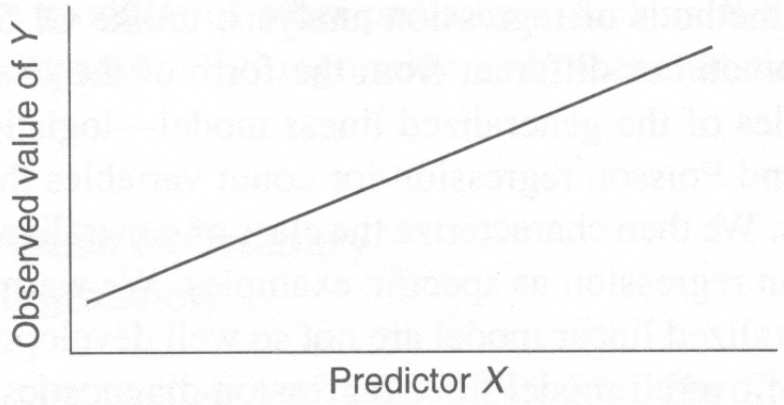


(B) For a binary outcome variable, the proportion of individuals who are "cases" (exhibit a particular outcome property) at each value of X.

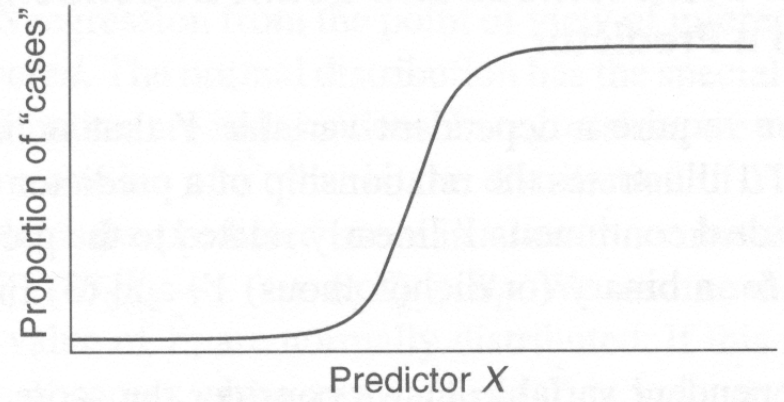


The logistic function

(A) For a continuous outcome variable Y, the numerical value of Y at each value of X.



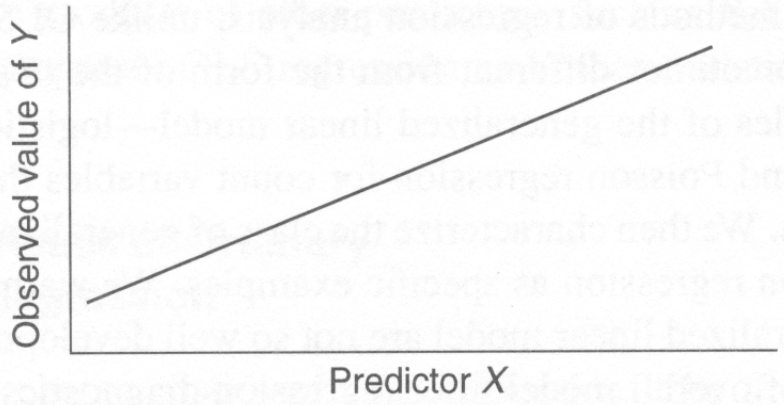
(B) For a binary outcome variable, the proportion of individuals who are "cases" (exhibit a particular outcome property) at each value of X.



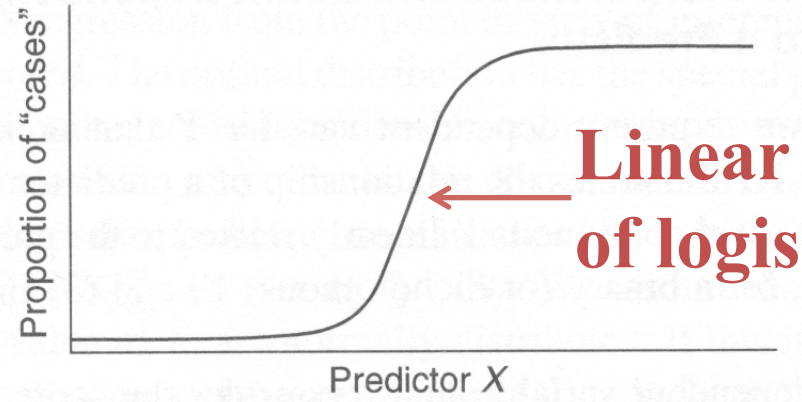
$$\pi = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

The logistic function

(A) For a continuous outcome variable Y, the numerical value of Y at each value of X.



(B) For a binary outcome variable, the proportion of individuals who are "cases" (exhibit a particular outcome property) at each value of X.



Linear part of logistic fit

Change in probability is not constant (linear) with constant changes in X

Assumptions

Linearity in the logit:

the regression equation should have a linear relationship with the logit form of the DV.

There is no assumption about the predictors being linearly related to each other.

Absence of multicollinearity

No outliers

R Vs RevoScaleR

R

```
>logitmodel_basic<-glm(
      Gender~Height,
      family=binomial,
      data=recordedDF)

>summary(logitmodel)
```

RevoScaleR

```
>logitmodel<-rxLogit(
      Gender~Height,
      data=recodedDF)

>summary(logitmodel)
```

Diabetes example

In R

```
# Load the data and remove NAs
>data("PimaIndiansDiabetes2", package = "mlbench")
>PimaIndiansDiabetes2 <- na.omit(PimaIndiansDiabetes2)

# run model
>logitmodel_R <- glm( diabetes ~glucose, data =
PimaIndiansDiabetes2, family = binomial)
>summary(logitmodel_R)
```

In R

```
Call:
glm(formula = diabetes ~ glucose, family = binomial, data =
PimaIndiansDiabetes2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1728  -0.7475  -0.4789   0.7153   2.3860

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.095521   0.629787  -9.679   <2e-16 ***
glucose      0.042421   0.004761   8.911   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 498.10  on 391  degrees of freedom
Residual deviance: 386.67  on 390  degrees of freedom
AIC: 390.67

Number of Fisher Scoring iterations: 4
```

In R

```
Call:
glm(formula = diabetes ~ glucose, family = binomial, data =
PimaIndiansDiabetes2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1728  -0.7475  -0.4789   0.7153   2.3860

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.095521   0.629787  -9.679   <2e-16 ***
glucose      0.042421   0.004761   8.911   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 498.10  on 391  degrees of freedom
Residual deviance: 386.67  on 390  degrees of freedom
AIC: 390.67

Number of Fisher Scoring iterations: 4
```

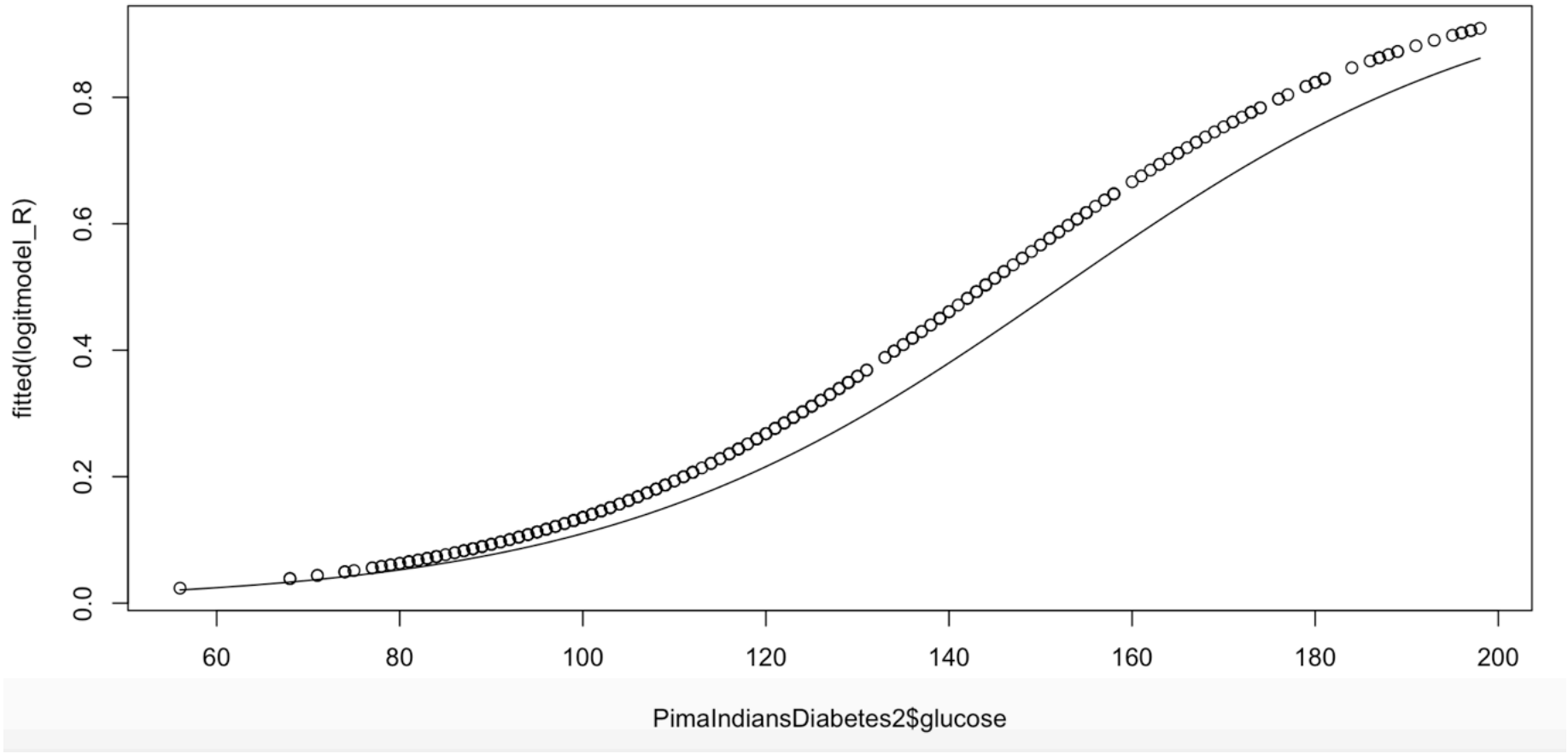
```
Call:
glm(formula = diabetes ~ glucose, family = binomial, data =
PimaIndiansDiabetes2)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.095521   0.629787  -9.679   <2e-16 ***
glucose      0.042421   0.004761   8.911   <2e-16 ***
---
```

$$\pi = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

Proportion of diabetic patients at the estimate glucose level

$$\hat{\pi} = \frac{e^{-6.09 + 0.04\,Ht}}{1 + e^{-6.09 + 0.04\,Ht}}$$

```
>plot(fitted(logitmodel_R)~PimaIndiansDiabete
s2$glucose)
>curve(exp(-6.09+0.04*x)/
(1+exp(-6.09+0.04*x)), add=TRUE)
```



Logistic regression and Odds

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X$$

where $odds = \dfrac{\pi}{1-\pi} = e^{\beta_0+\beta_1 X} = \dfrac{P(Yes)}{P(No)}$

The logistic model assumes a linear relationship between the *predictors* and *log(odds)*.

$$odds = \frac{\pi}{1-\pi} \Leftrightarrow \pi = \frac{odds}{1+odds}$$

The assumption underlying the logistic model is that this odds ratio does not depend on x

if $$odds = \frac{\pi}{1-\pi} = e^{\beta_0 + \beta_1 X}$$

Then the odd ratio (when x increases of 1) is $e^{\beta 1}$

Logistic Regression for TMS data

```
glm(formula = diabetes ~ glucose, family = binomial,
data = PimaIndiansDiabetes2)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.095521   0.629787  -9.679   <2e-16 ***
glucose      0.042421   0.004761   8.911   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’
0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be
1)
    Null deviance: 498.10  on 391  degrees of freedom
Residual deviance: 386.67  on 390  degrees of freedom
AIC: 390.67

Number of Fisher Scoring iterations: 4
```

Note: $e^{0.04} = 1.040811$ = odds ratio

Example: TMS for Migraines

Transcranial Magnetic Stimulation vs. Placebo

Pain Free?	TMS	Placebo
YES	39	22
NO	61	78
Total	**100**	**100**

Example: TMS for Migraines

Transcranial Magnetic Stimulation vs. Placebo

Pain Free?	TMS	Placebo
YES	39	22
NO	61	78
Total	**100**	**100**

$$odds_{TMS} = \frac{39/100}{61/100} = \frac{39}{61} = 0.639$$

$$\hat{\pi} = \frac{0.639}{1+0.639} = 0.39$$

Example: TMS for Migraines

Transcranial Magnetic Stimulation vs. Placebo

Pain Free?	TMS	Placebo
YES	39	22
NO	61	78
Total	**100**	**100**

$$odds_{TMS} = \frac{39/100}{61/100} = \frac{39}{61} = 0.639$$

$$odds_{Placebo} = \frac{22}{78} = 0.282$$

Example: TMS for Migraines

Transcranial Magnetic Stimulation vs. Placebo

Pain Free?	TMS	Placebo
YES	39	22
NO	61	78
Total	100	100

$$odds_{TMS} = \frac{39/100}{61/100} = \frac{39}{61} = 0.639$$

$$odds_{Placebo} = \frac{22}{78} = 0.282$$

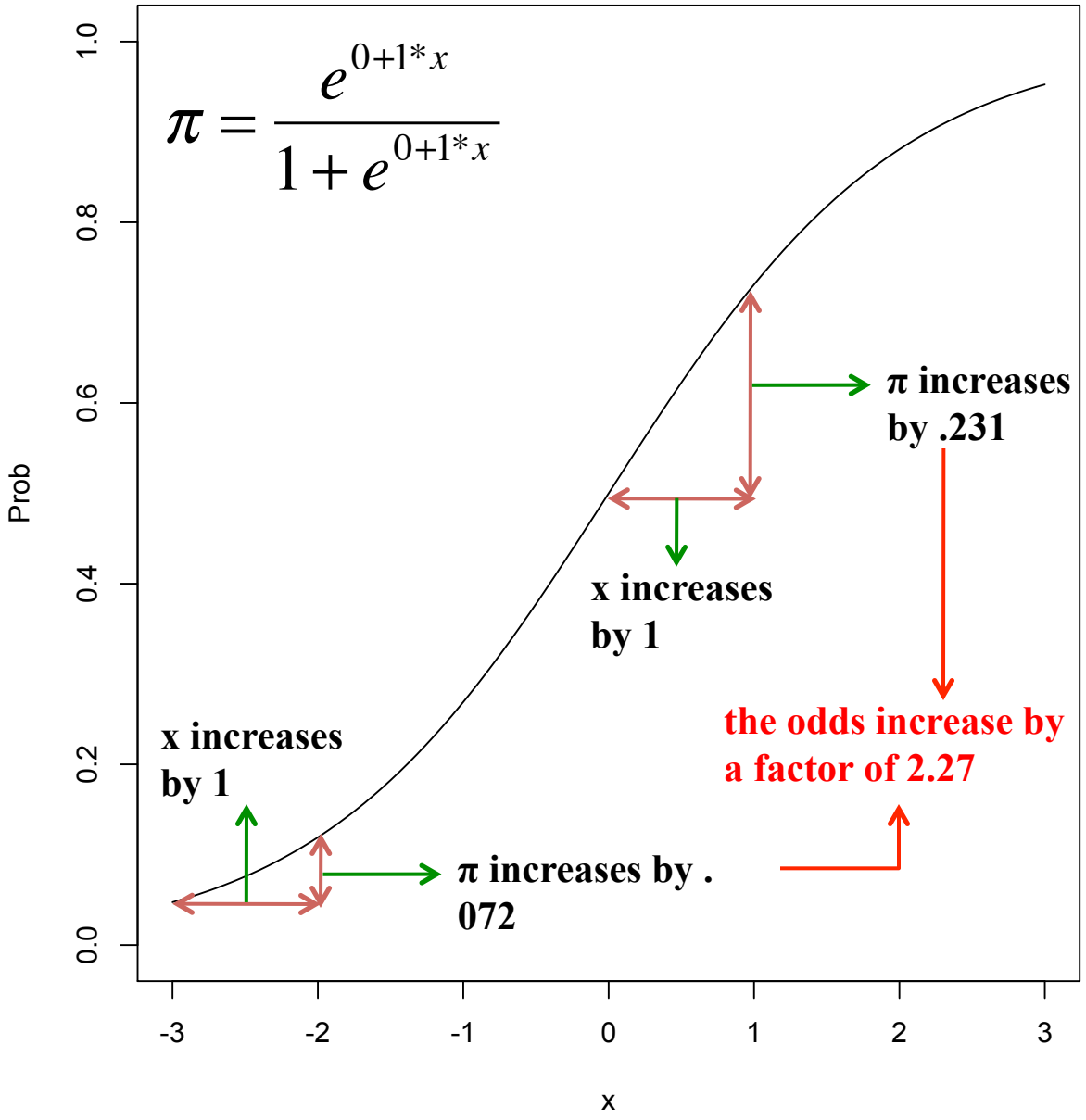
$$Odds\ ratio = \frac{0.639}{0.282} = 2.27$$

Odds are 2.27 times higher of getting relief using TMS than placebo

Logistic Regression for TMS data

```
>lmod=glm(cbind(Yes,No)~Group,family=binomial,data=TMS)

>summary(lmod)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.2657     0.2414  -5.243 1.58e-07 ***
GroupTMS      0.8184     0.3167   2.584  0.00977 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1

(Dispersion parameter for binomial family taken to be
1)

    Null deviance: 6.8854  on 1  degrees of freedom
Residual deviance: 0.0000  on 0  degrees of freedom
AIC: 13.701
```

Note: $e^{0.8184} = 2.27$ = odds ratio


$\pi = \frac{e^{0+1*x}}{1+e^{0+1*x}}$
Prob
0.0
0.2
0.4
0.6
0.8
1.0
-3
-2
-1
0
1
2
3
x
x increases by 1
π increases by .072
x increases by 1
π increases by .231
the odds increase by a factor of 2.27

Two forms of logistic data

1. “Bernoulli” logistic regression.

Example: 0/1

2. “Binomial counts” logistic regression.

Example: 0/1/2/..

Challenge

In R: use logistic regression to look at whether self-reported race/ethnicity predicts having a health care plan in BRFSS data.

…This is reasonably quick…

Challenge: solution

In R: use logistic regression to investigate whether self-reported race/ethnicity predicts having a health care plan in BRFSS data.

```
>brfss$has_plan <- brfss$hlthpln1 == 1
>summary(glm(has_plan ~ as.factor(x.race), data=brfss, family=binomial))
```

Challenge: Breast cancer Dataset

```
>data(BreastCancer, package="mlbench")
>bc$Class <- ifelse(bc$Class == "malignant", 1, 0)
>bc$Class <- factor(bc$Class, levels = c(0, 1))
>bc <- BreastCancer[complete.cases(BreastCancer), ]
str(bc)
>logit_out<-glm(Class ~ Cell.shape,
family="binomial", data = bc)
>summary(logit_out)
```

Use RevolScaleR commands to perform this logistic regression

Thank you for your attention