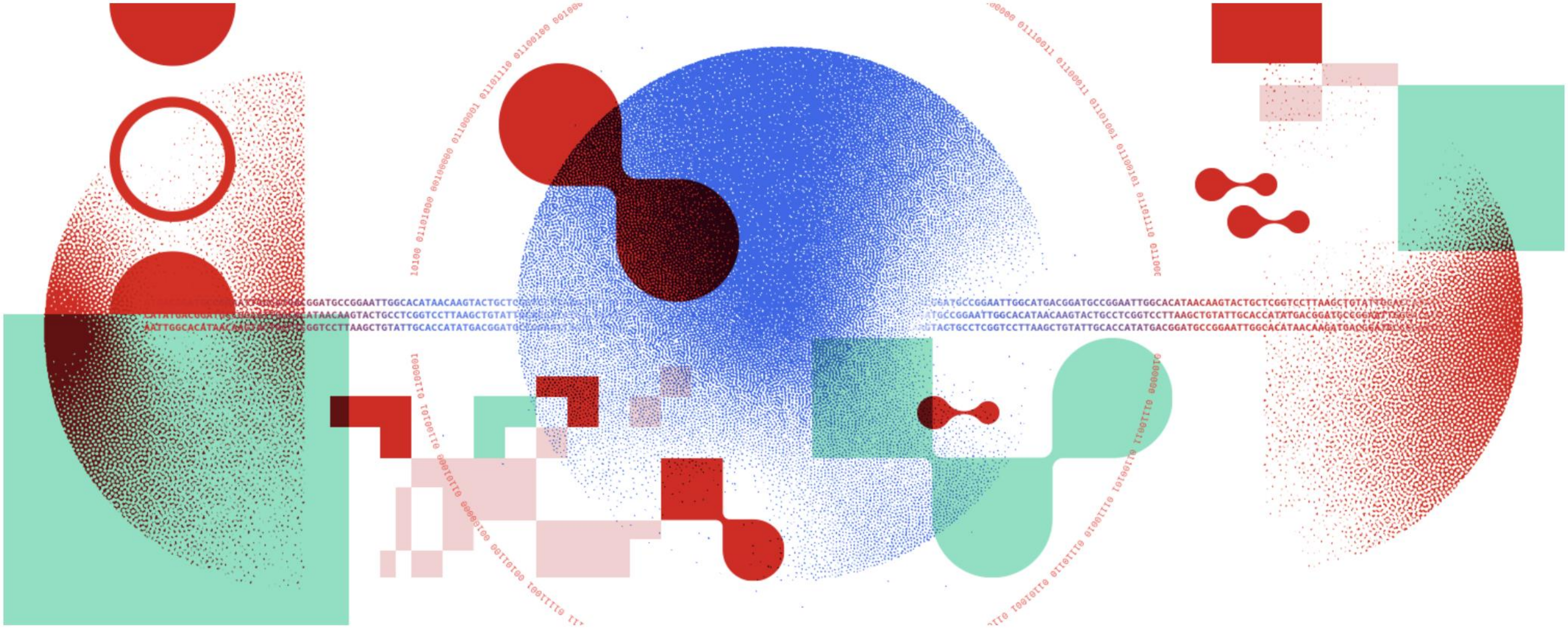


Introduction to statistics

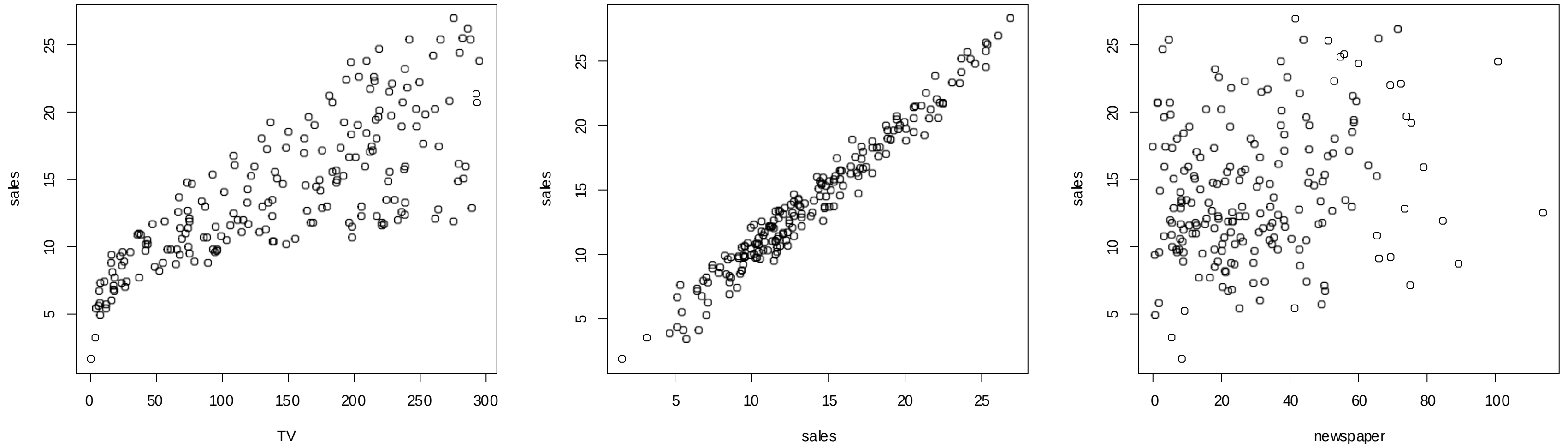
Lausanne, January 2025

Joao Lourenço and Rachel Marcone

Correlation and Simple Regression

Day 3:
Correlation and Regression

Scatterplot

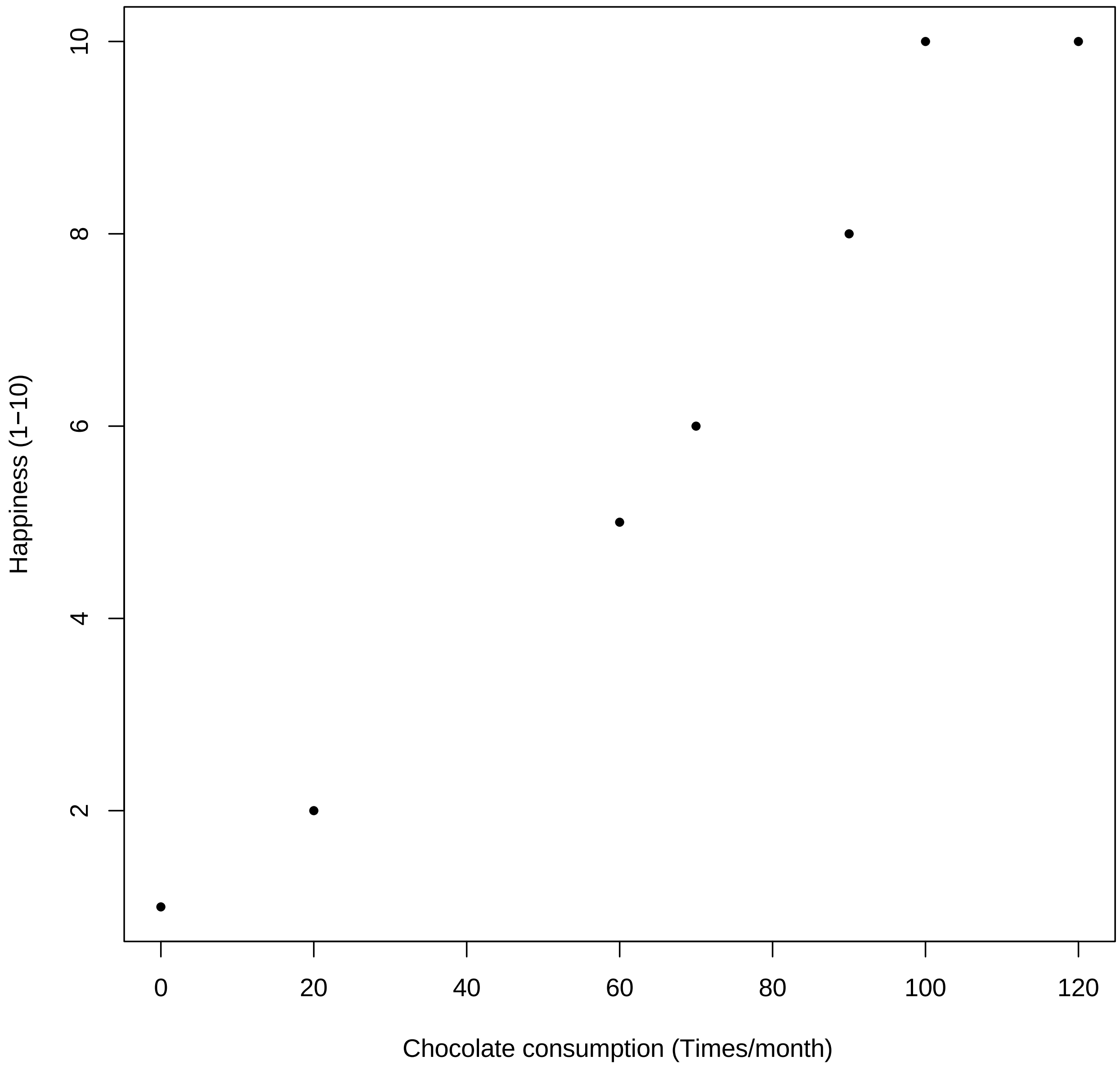


We are often interested in the statistical dependence between two variables, aka “correlation”

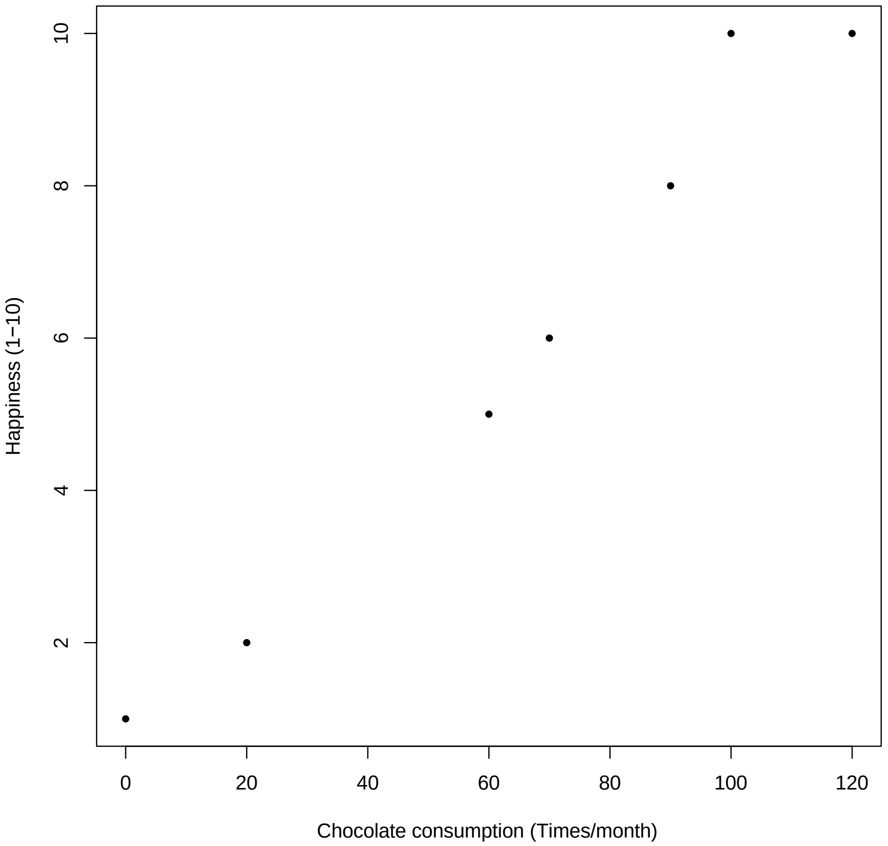
Pearson correlation

- Is a measure of linear association
- Pearson correlation coefficient (r) indicates the strength of a linear relationship between two variables
- Pearson correlation coefficient (r) is defined as cov(X,Y)/sd(X)*sd(Y) which corresponds to a sort of average value of the product

  (*X* in SUs)*(*Y* in SUs)
- where SU = standard units
- *X* in SUs = (X – mean(X))/SD(X)
- *Y* in SUs = (Y – mean(Y))/SD(Y)



10
8
6
4
2
Happiness (1−10)
0
20
40
60
80
100
120
Chocolate consumption (Times/month)



Chocolate consumption	Happiness
70	6
60	5
0	1
90	8
20	2
100	10
120	10

Pearson correlation

Average of (*X* in SUs)*(*Y* in SUs)

- where SU = standard units
- *X* in SUs = (X – mean(X))/SD(X)
- *Y* in SUs = (Y – mean(Y))/SD(Y)
- X=(70,60,0,90,20,100,120), mean(Y) = 65.71429, SD(Y) = 43.14979
- Xin SUs = (0.09932178, -0.13242904, -1.52293392, 0.56282341, -1.05943229, 0.79457422, 1.25807585)
- Y= (6,5,1,8,2,10,10), mean(X) = 6, SD(X)= 3.605551
- Y in SUs = (0.0000000 ,-0.2773501, -1.3867505, 0.5547002, -1.1094004, 1.1094004, 1.1094004)
- Average of (*X* in SUs)*(*Y* in SUs) = 5.913401/6 = 0.9855668

Pearson correlation-Guide for interpretation

Evans, J. D. (1996) (Straightforward statistics for the behavioral sciences.) suggests for the absolute value of r:

.00-.19 “very weak”
.20-.39 “weak”
.40-.59 “moderate”
.60-.79 “strong”
.80-1.0 “very strong”

Pearson correlation

$$-1 \leq r \leq 1$$

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

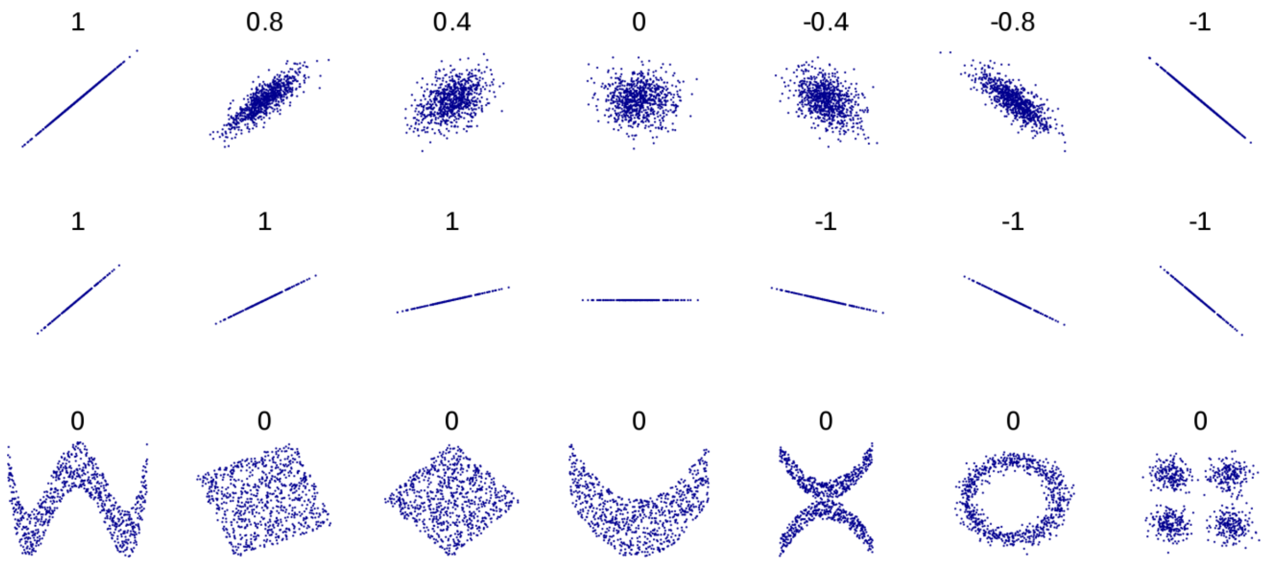


Image source: Wikipedia

To recap …

- ***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), *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 variable (either causal or not)

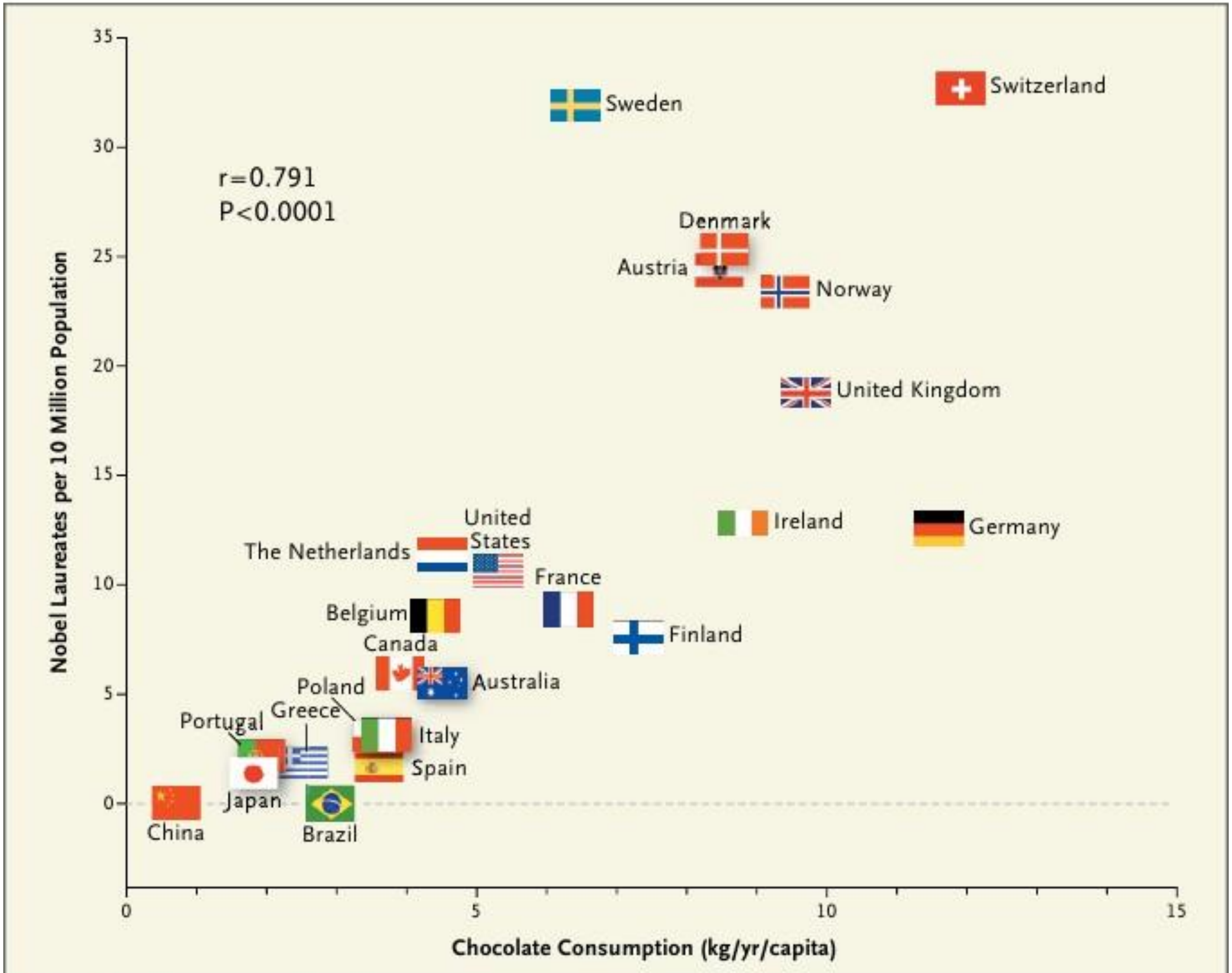
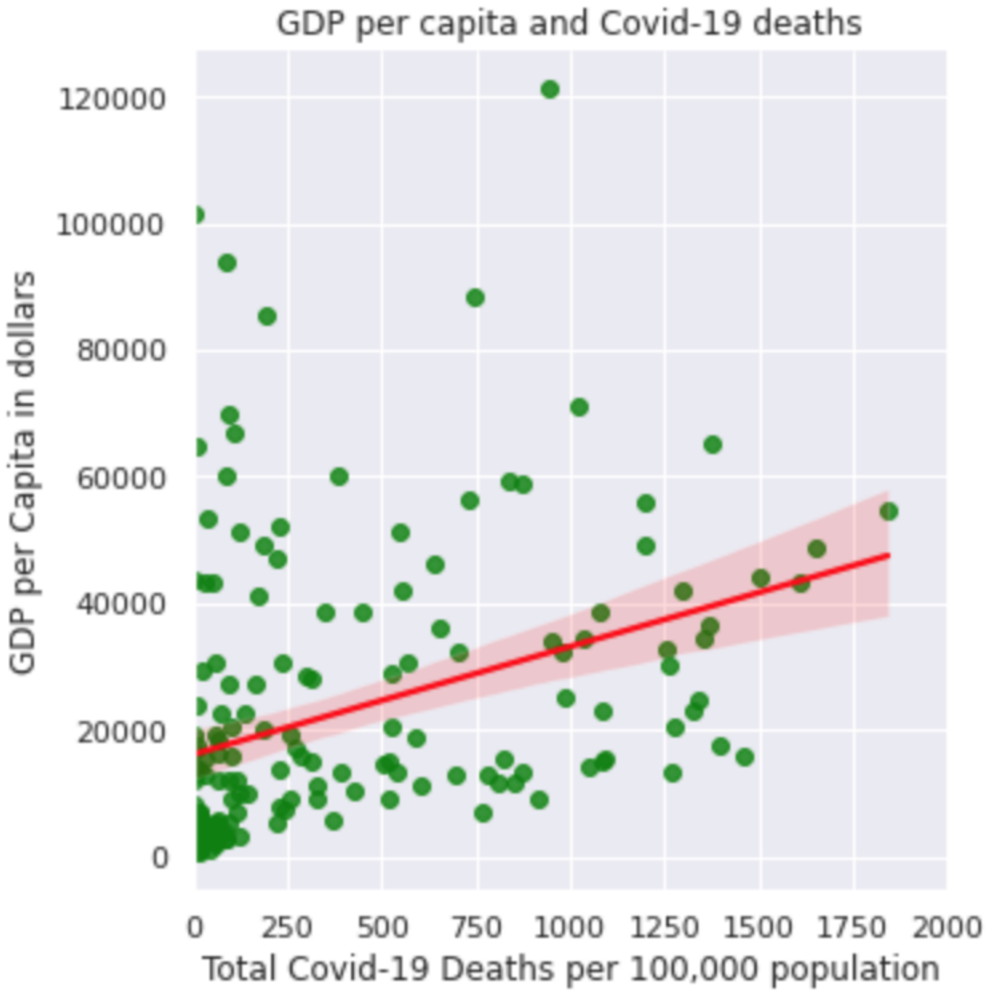


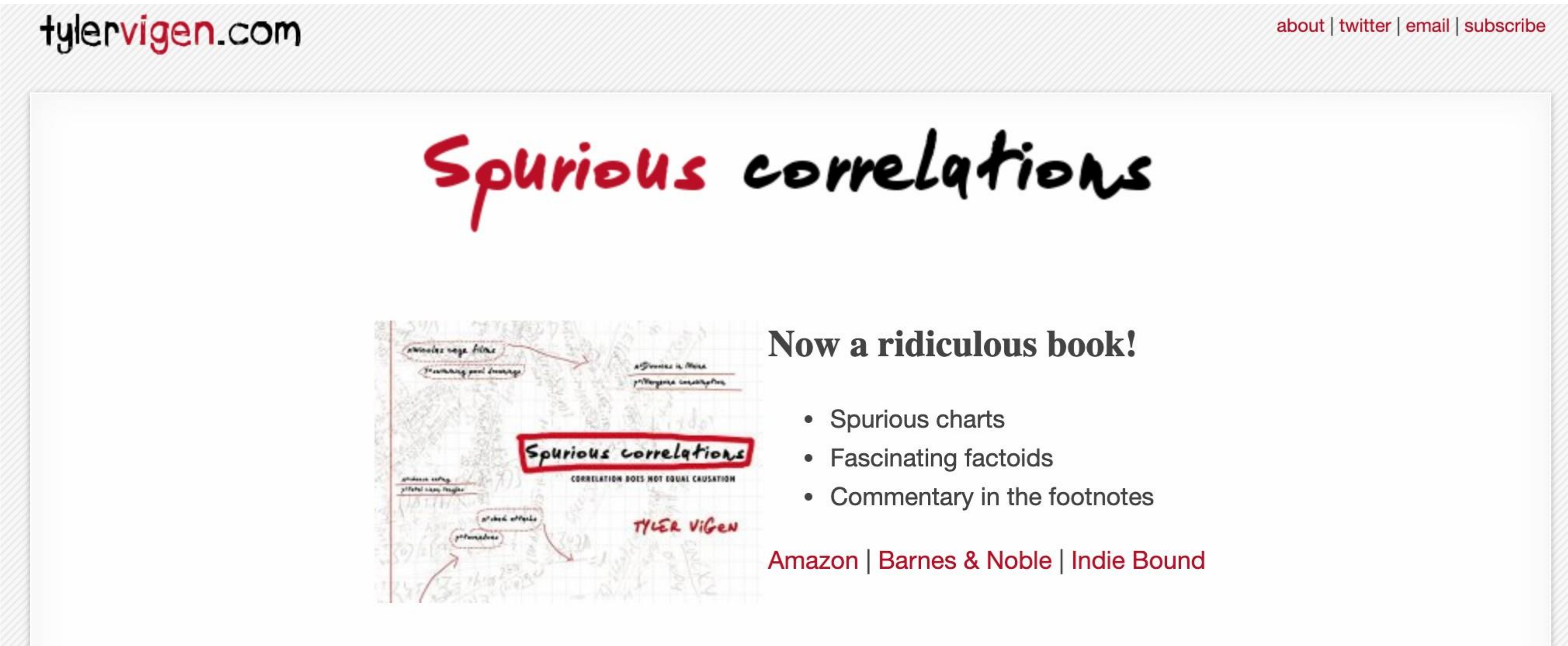
Figure 1. Correlation between Countries' Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.

Source: https://www.nejm.org/doi/full/10.1056/NEJMon1211064



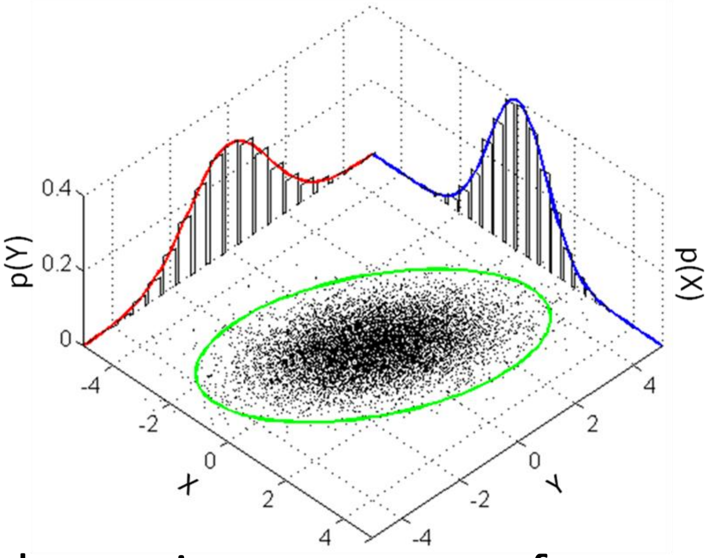
https://towardsdatascience.com/coronavirus-correlations-5f49e5bb9710

Correlation is not causation



Assumptions of Pearson correlation

- The only assumption of Pearson correlation is that the data follows a bivariate normal distribution

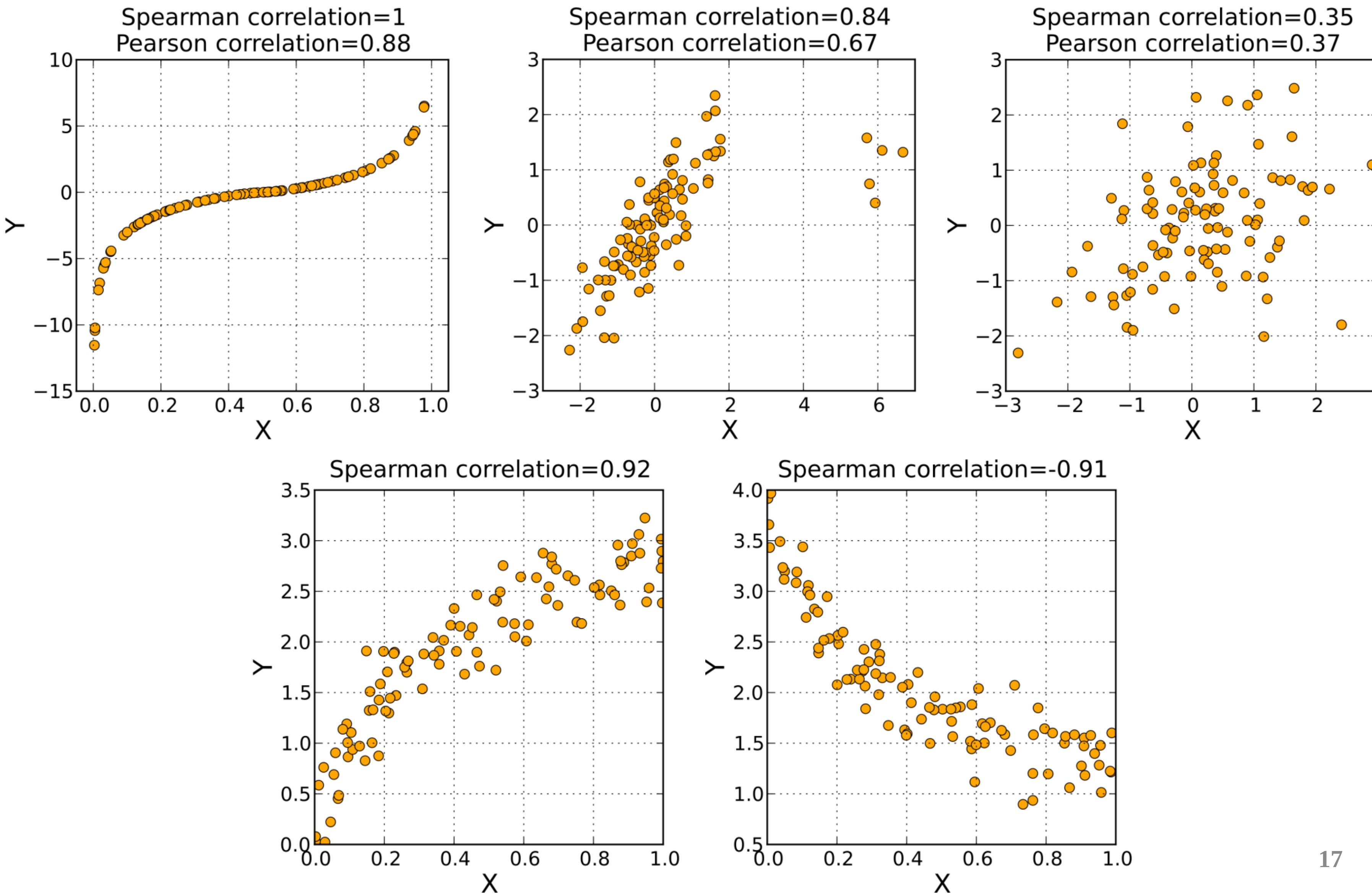


- When this assumption is not met, alternative measures of association between two variables should be used
  - Spearman rank correlation
  - Kendal rank correlation

Spearman (rank) correlation

- A nonparametric measure of rank correlation
- The Spearman correlation coefficient (denoted by the Greek letter rho) is defined as the Pearson correlation coefficient between the rank variables
  - also a unit-less value varying between -1 and +1
- It tells us how well the relationship between two variables can be described using a monotonic function
  - increase/decrease in one variable is associated with increase/decrease in the other variable
  - Not necessarily linear association!

Spearman correlation



In R:

```
>?cor

>?cor.test
```

```
>cor(x,y)

>cor.test(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)

    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
```

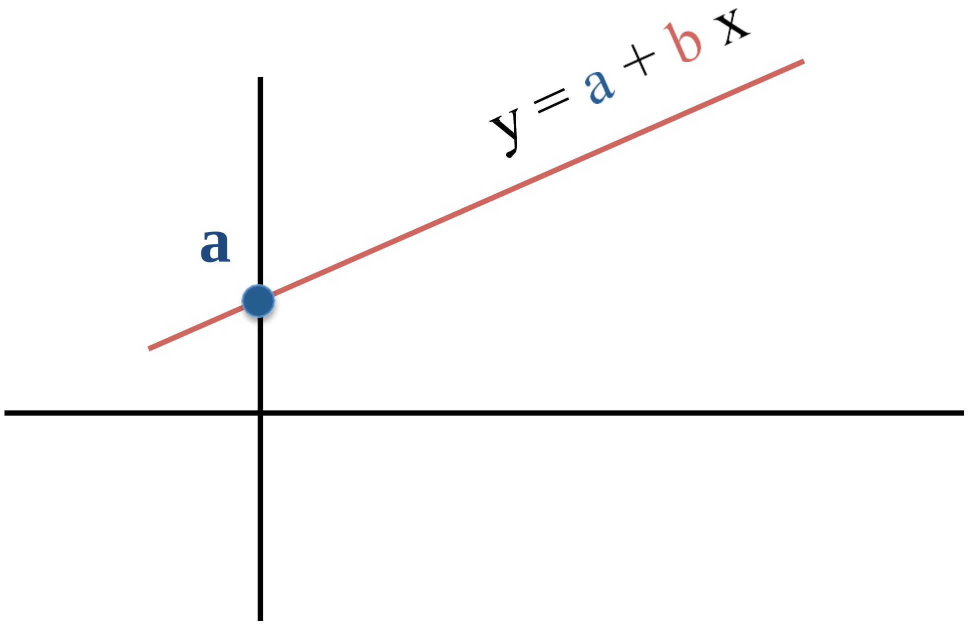
- **Correlation** describes the association between variables, but does not describe it

- Often it is useful to obtain a mathematical model that describes the association between variables, hence **regression**

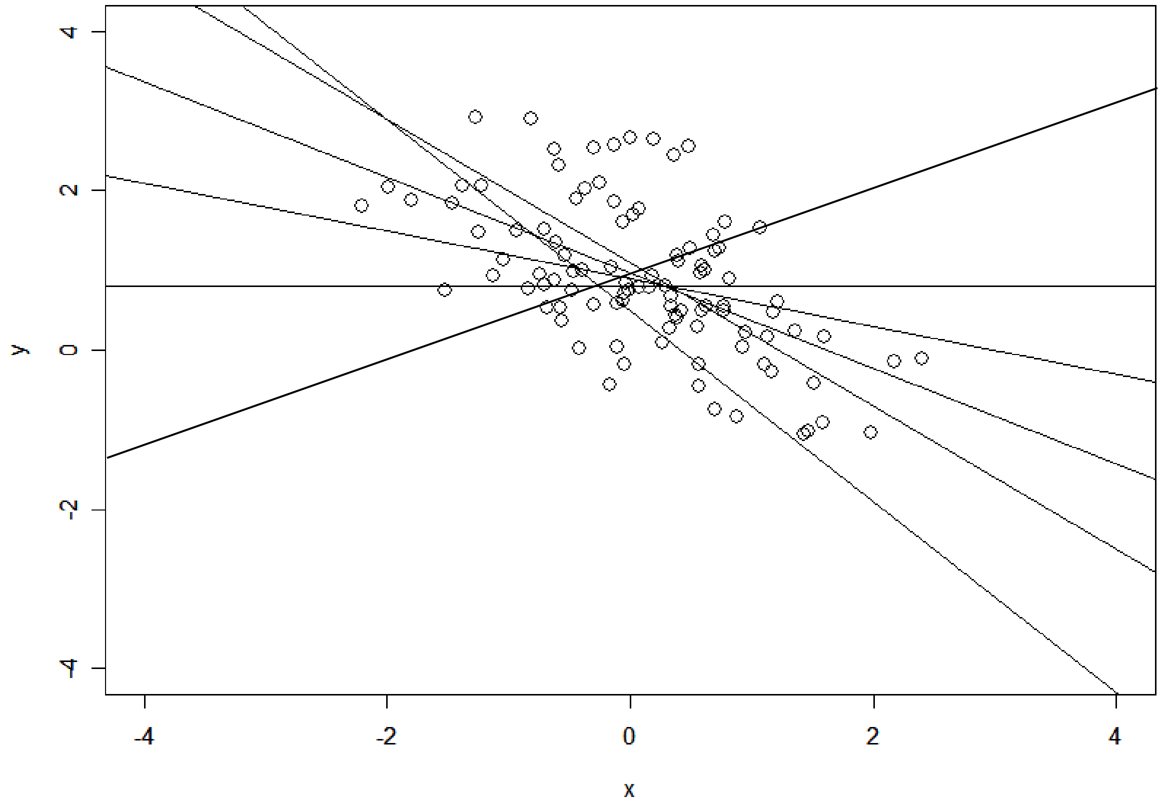
The equation for a line that can be used to predict y knowing x (in slope-intercept form) looks like

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

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



What is the “best” line that fits this data ? → need a criteria
Can we use it to summarize the relation between x and y ?



y = 0.9 + 0.6x

y = 0.8 + 0x

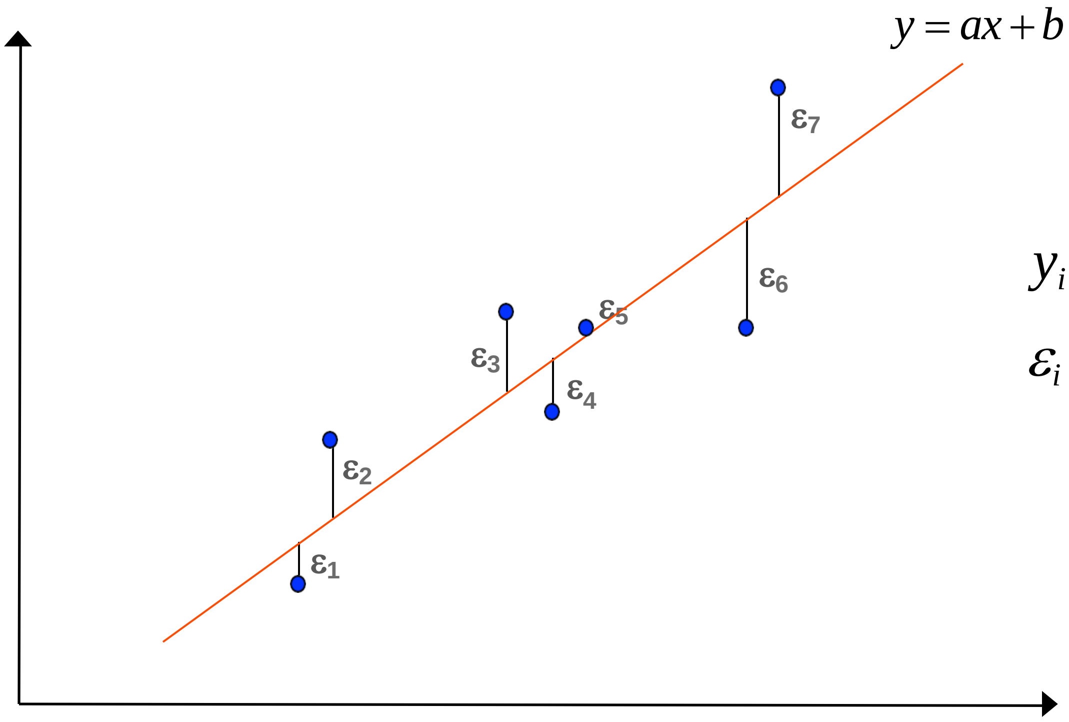
y = 0.9 – 0.3x

y = 1 – 0.6x

y = 1.1 – 0.9x

y = 0.5 – 1.2x

Least-squares approach to fit a line



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

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

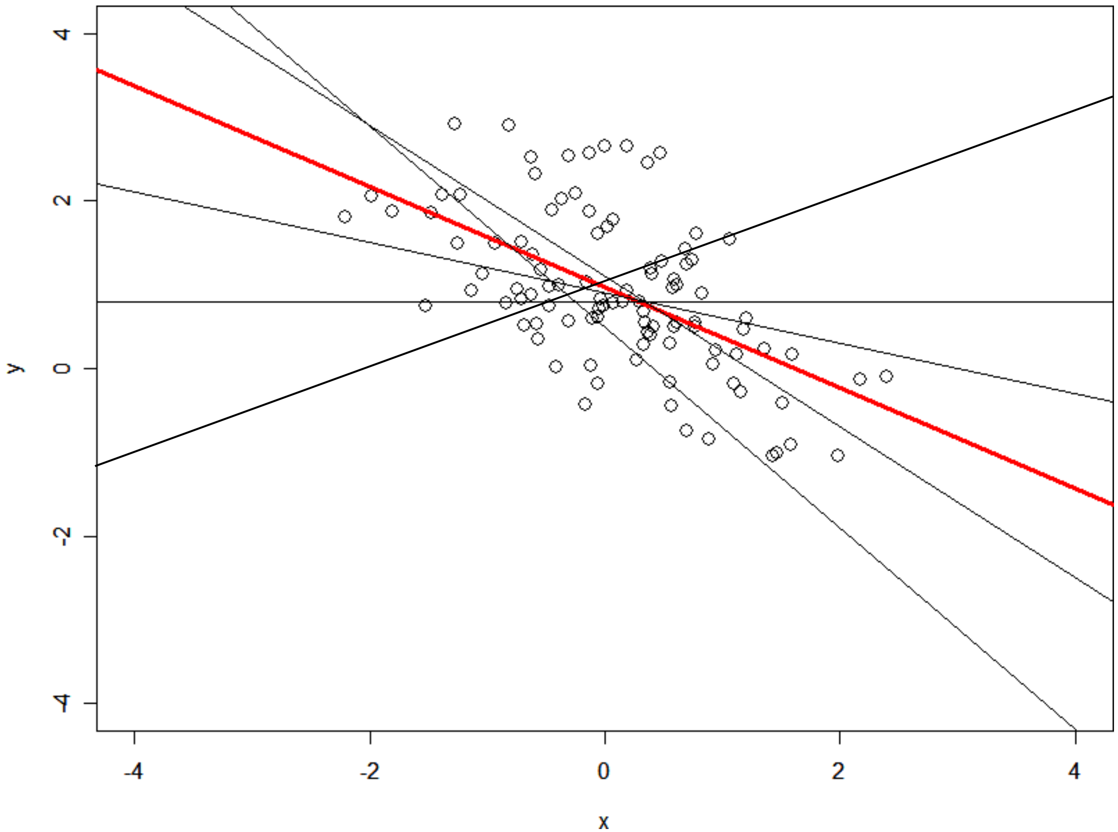
The least-squares procedure finds the straight line with the smallest sum of squares of vertical errors.

Finds a regression line such that $\sum_i \varepsilon_i^2 = \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \ldots$ is minimum.

Over all possible straight lines,

y= 1 - 0.6x is the “best” possible line

according to least-squares criterion



y = 0.9 + 0.6x

y = 0.8 + 0x

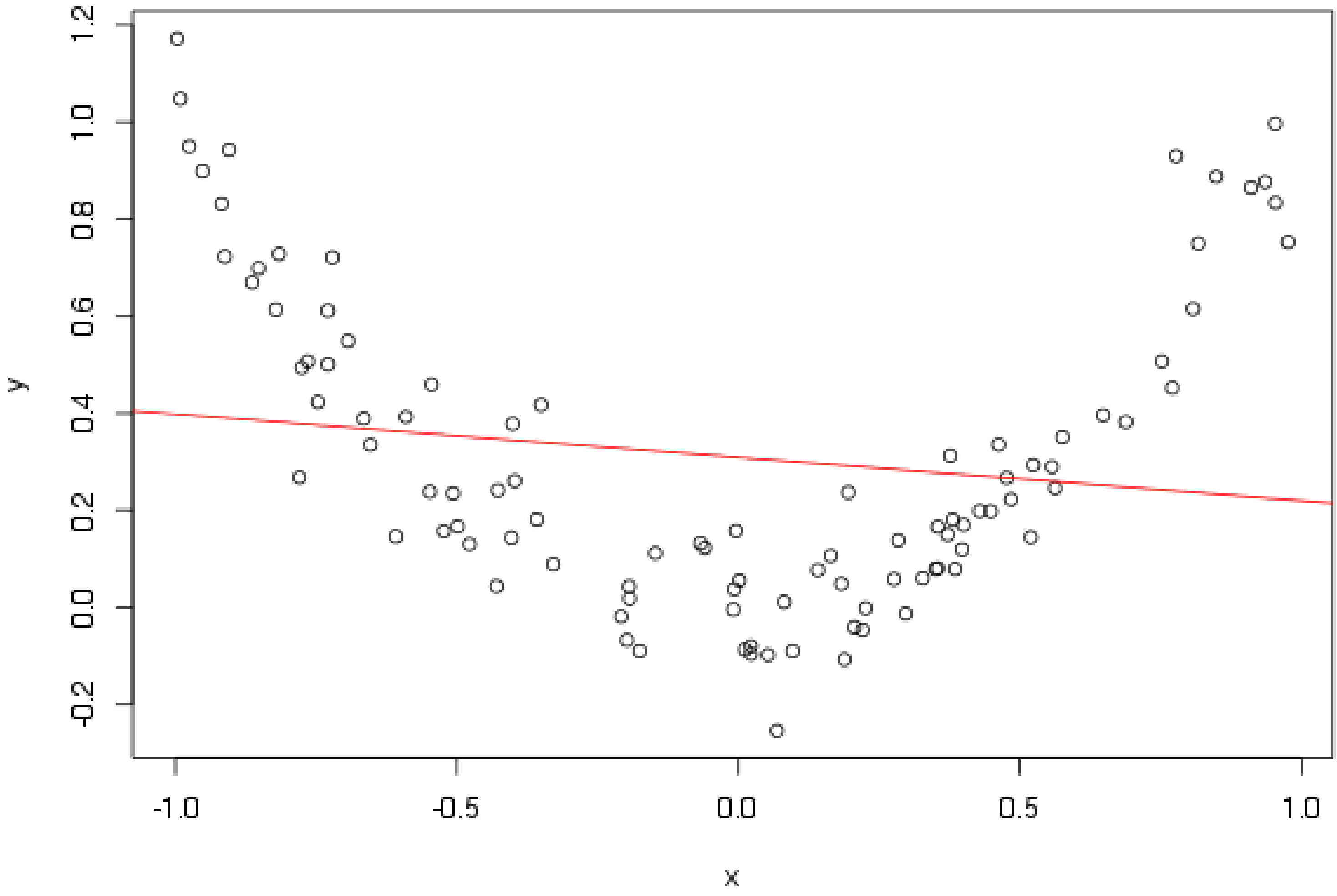
y = 0.9 – 0.3x

y = 1 – 0.6x

y = 1.1 – 0.9x

y = 0.5 – 1.2x

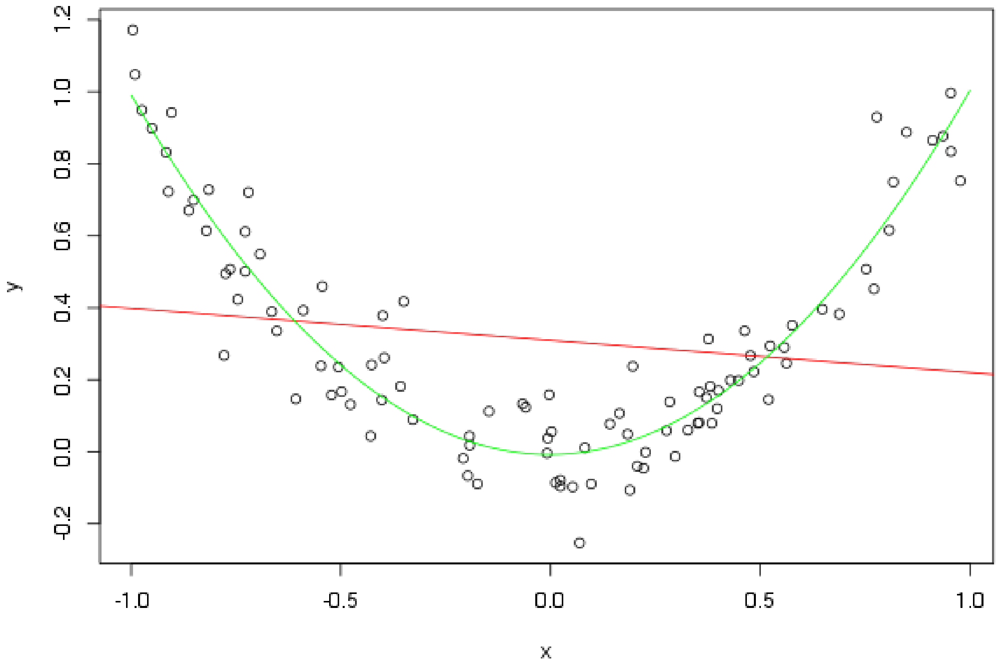
What if the association is not linear ?



What if the data is not linear ?

Use a polynomial regression

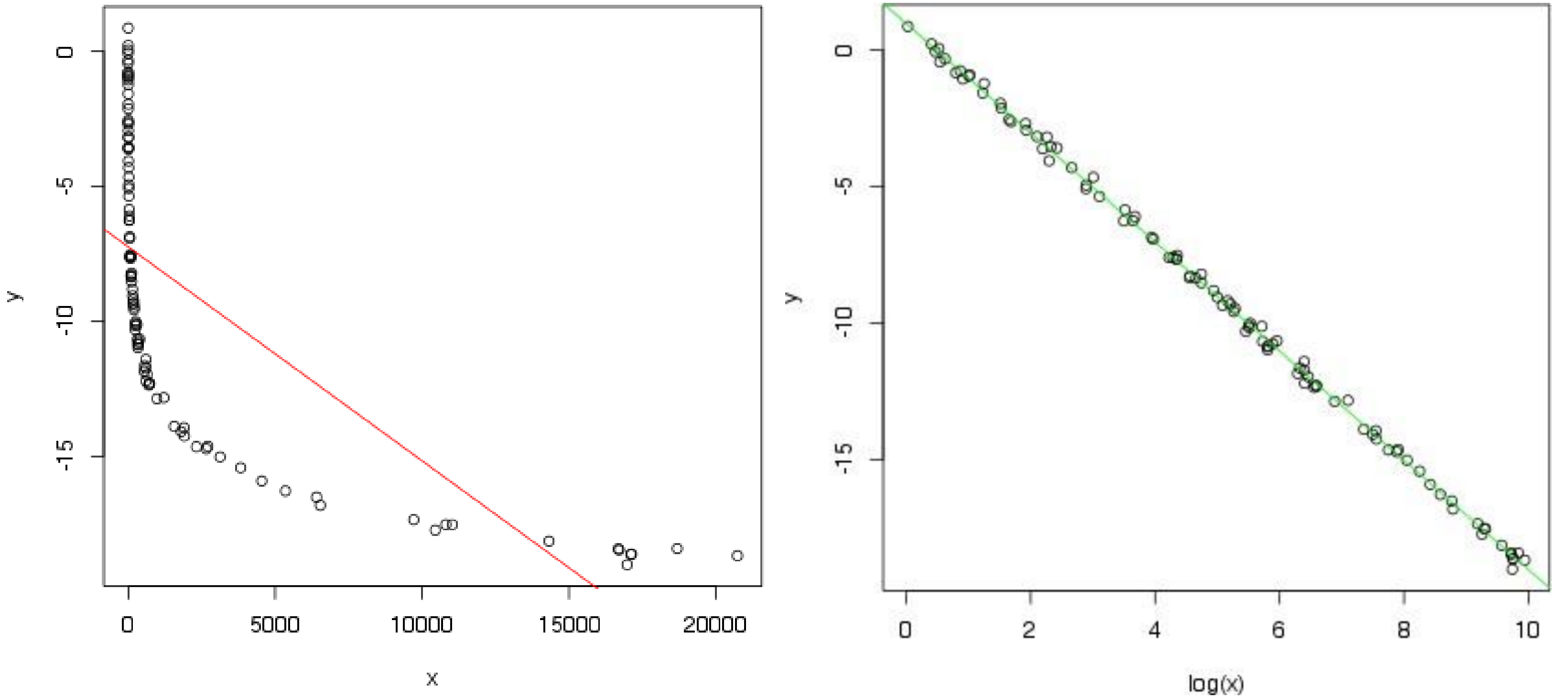
$$\mathbf{y = b_0 + b_1\,x + b_2\,x^2}$$



What if the association is not linear ?

Consider transforming the data (log)

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



Linear models in matrix form

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

is equivalent to

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

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

Linear models in matrix form

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

is equivalent to

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

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

Linear models in matrix form

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

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

Linear models in matrix form

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

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

Least-square estimation of regression coefficients

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

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

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

Least-square estimation of regression coefficients

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

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

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

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

Computationally intensive

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

in R:

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

read "~" as "described (or modeled) by"

By default, an intercept is included in the model

To leave the intercept out:

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

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

in R:

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

read "~" as "described (or modeled) by"

By default, an intercept is included in the model

To leave the intercept out:

```
yvar ~ -1 + xvar1 + xvar2 + xvar3
yvar ~ 0 + xvar1 + xvar2 + xvar3
```

More on model formulas

Generic form

`response ~ predictors`

predictors can be `numeric` or `categorical`

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*

Let’s walk through an example in R

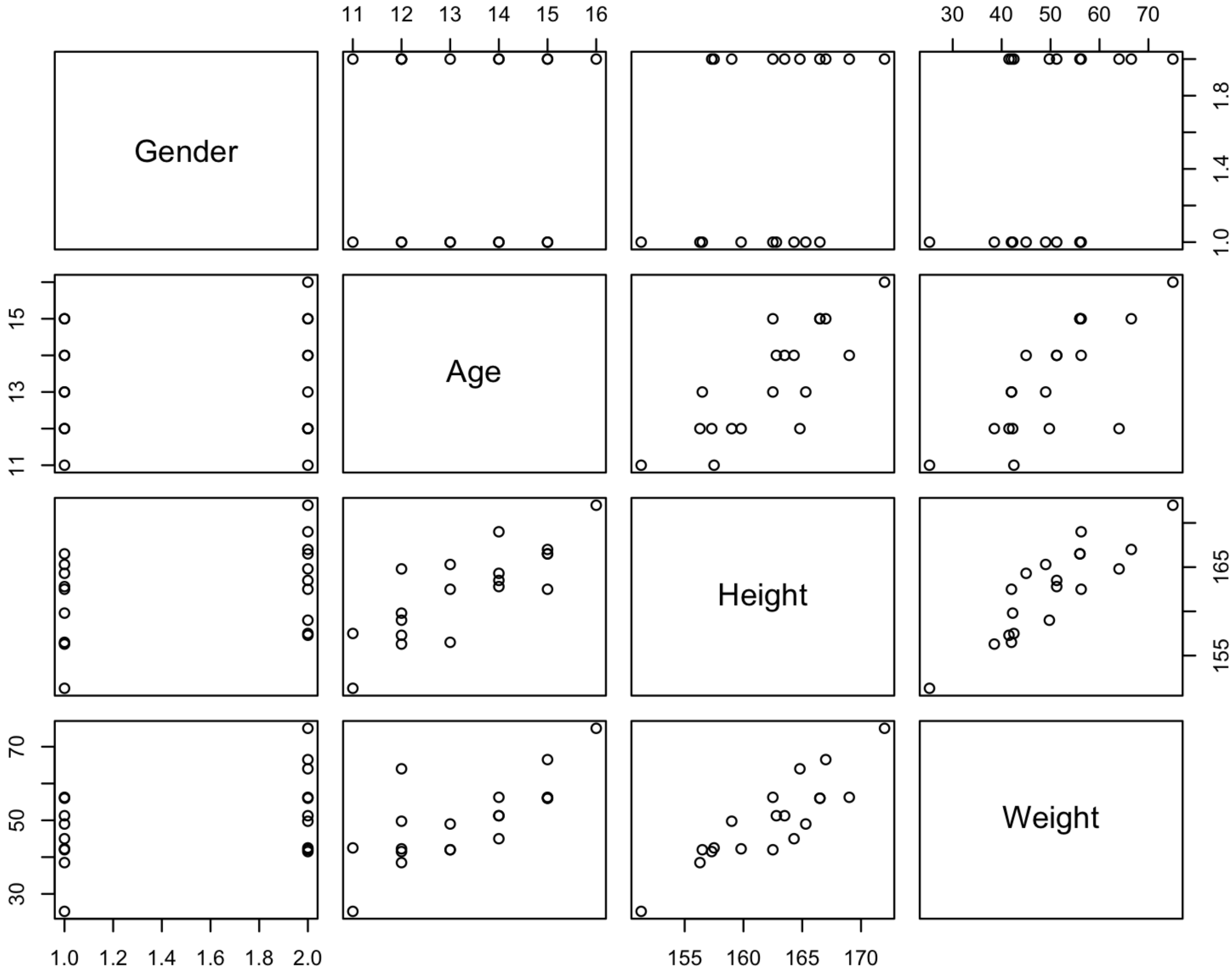
Inspired by the CLASS dataset, from the program SAS (units have been modified from imperial to metric)

The CLASS dataset

```
> class
      Name Gender Age Height Weight
1    JOYCE      F  11  151.3  25.25
2   THOMAS      M  11  157.5  42.50
3    JAMES      M  12  157.3  41.50
4     JANE      F  12  159.8  42.25
5     JOHN      M  12  159.0  49.75
6   LOUISE      F  12  156.3  38.50
7   ROBERT      M  12  164.8  64.00
8    ALICE      F  13  156.5  42.00
9  BARBARA      F  13  165.3  49.00
10 JEFFREY      M  13  162.5  42.00
11   CAROL      F  14  162.8  51.25
12   HENRY      M  14  163.5  51.25
13  ALFRED      M  14  169.0  56.25
14    JUDY      F  14  164.3  45.00
15   JANET      F  15  162.5  56.25
16    MARY      F  15  166.5  56.00
17  RONALD      M  15  167.0  66.50
18 WILLIAM      M  15  166.5  56.00
19  PHILIP      M  16  172.0  75.00
```

The CLASS dataset

```
> summary(class)
     Name              Gender               Age            Height     
 Length:19          Length:19          Min.   :11.00   Min.   :151.3  
 Class :character   Class :character   1st Qu.:12.00   1st Qu.:158.2  
 Mode  :character   Mode  :character   Median :13.00   Median :162.8  
                                       Mean   :13.32   Mean   :162.3  
                                       3rd Qu.:14.50   3rd Qu.:165.9  
                                       Max.   :16.00   Max.   :172.0  
     Weight     
 Min.   :25.25  
 1st Qu.:42.12  
 Median :49.75  
 Mean   :50.01  
 3rd Qu.:56.12  
 Max.   :75.00  
> pairs(class[,-1])
```



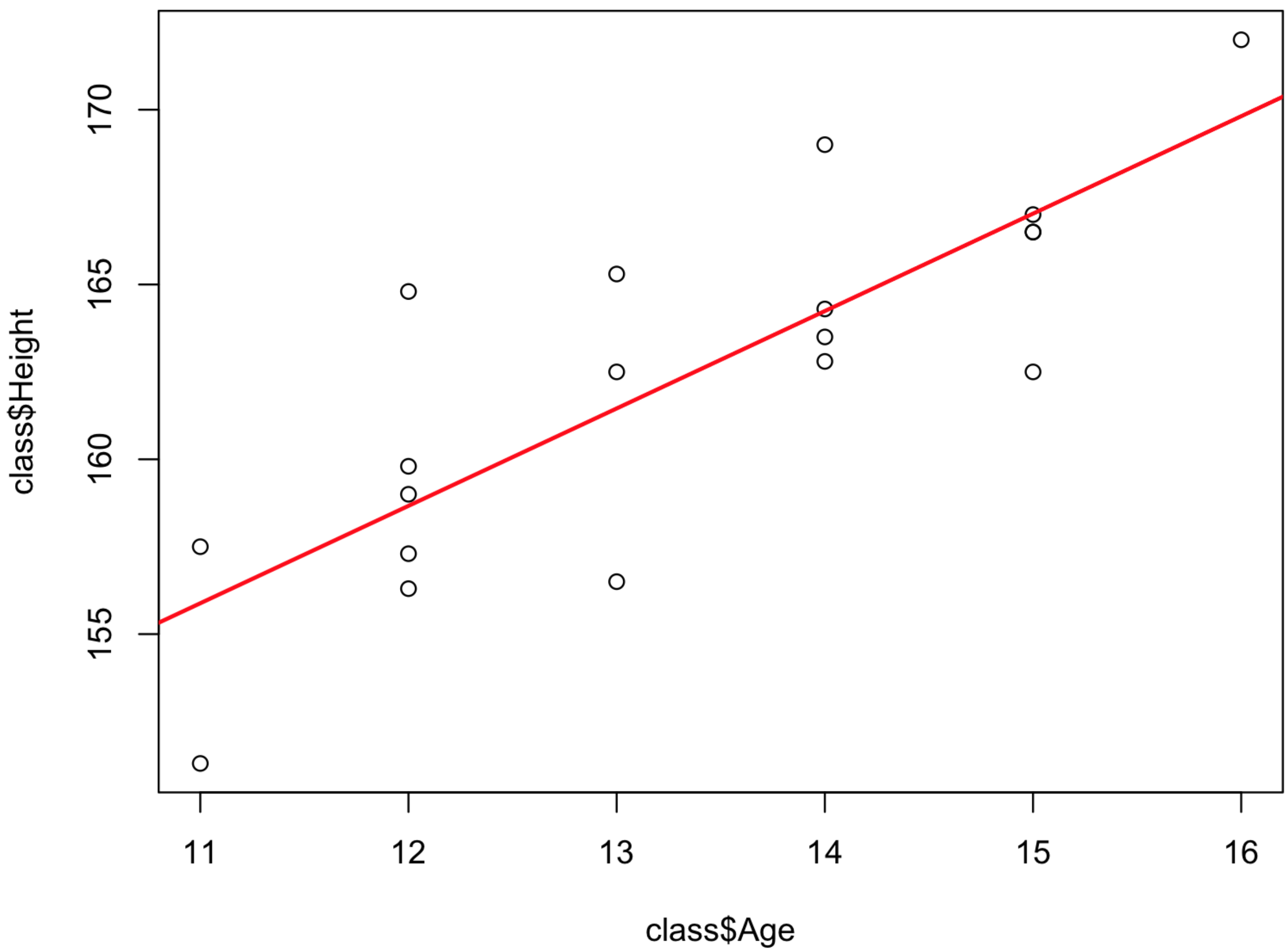
Gender
Age
Height
Weight
11 12 13 14 15 16
30 40 50 60 70
1.0 1.4 1.8
11 13 15
155 165
30 50 70
1.0 1.2 1.4 1.6 1.8 2.0
155 160 165 170

Fitting the linear model in R

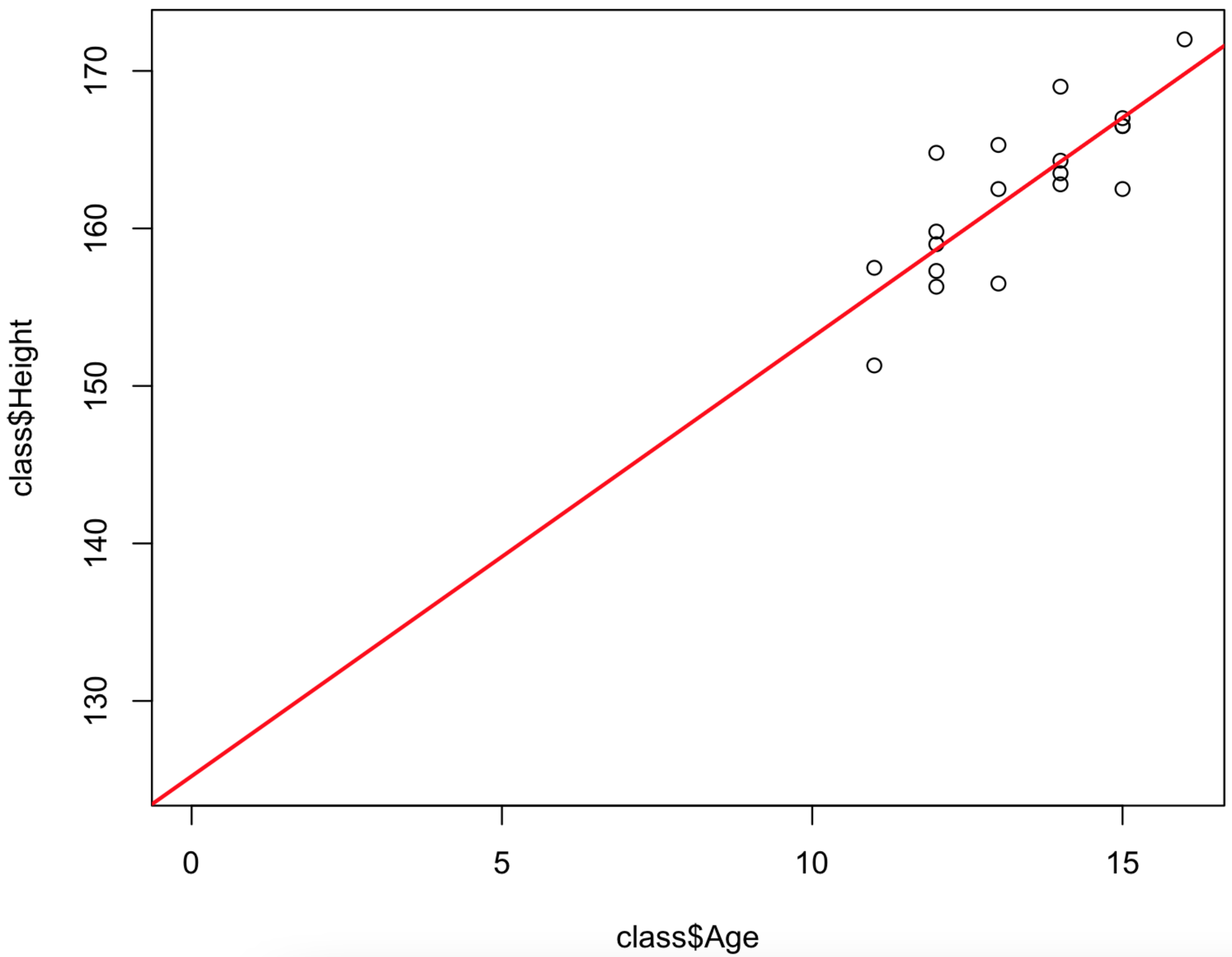
```
> lm( Height ~ Age, data=class)

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

Coefficients:
(Intercept)          Age
    125.224        2.787

> model <- lm( Height ~ Age, data=class)
> model

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

Coefficients:
(Intercept)          Age
    125.224        2.787
```

Height = 125.224 + 2.787x Age

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



```
> plot(class$Age, class$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(model)

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

Residuals:
   Min     1Q Median     3Q    Max
-4.957 -1.407 -0.031  1.374  6.130

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 125.2239     6.5217  19.201 5.82e-13 ***
Age           2.7871     0.4869   5.724 2.48e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.083 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(model)
```
Function call

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

Residuals:
   Min     1Q Median     3Q    Max 
-4.957 -1.407 -0.031  1.374  6.130 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 125.2239     6.5217  19.201 5.82e-13 ***
Age           2.7871     0.4869   5.724 2.48e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.083 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(model)

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

Residuals:
   Min      1Q Median      3Q     Max 
-4.957 -1.407 -0.031  1.374  6.130 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 125.2239     6.5217  19.201 5.82e-13 ***
Age           2.7871     0.4869   5.724 2.48e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.083 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
```

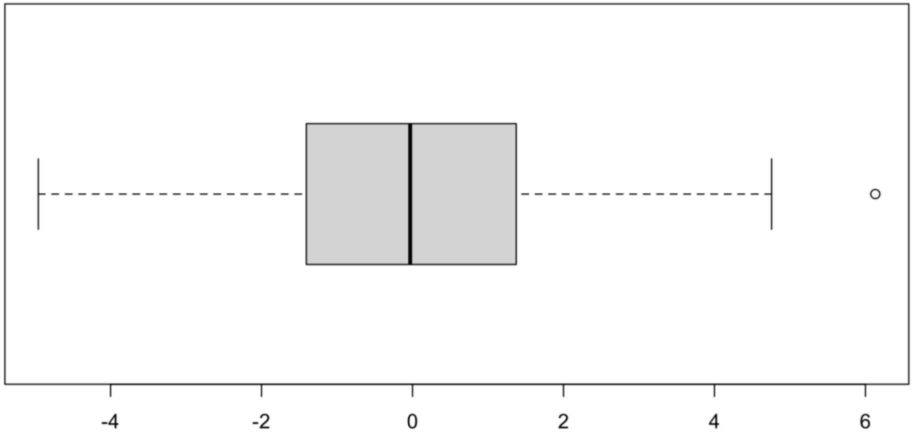
Five-number summary of the residuals equivalent to

```
> fivenum( residuals( model ) )
```

```
          8          11          17           4           7
-4.95669291 -1.40669291 -0.03097113  1.37401575  6.13044619
```

or, graphically, using a boxplot:

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



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

```
> summary(model)

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

Residuals:
   Min     1Q Median     3Q    Max 
-4.957 -1.407 -0.031  1.374  6.130 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 125.2239     6.5217  19.201 5.82e-13 ***
Age           2.7871     0.4869   5.724 2.48e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.083 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 used for hypothesis testing

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(model)

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

Residuals:
   Min     1Q Median     3Q    Max 
-4.957 -1.407 -0.031  1.374  6.130 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 125.2239     6.5217  19.201 5.82e-13 ***
Age           2.7871     0.4869   5.724 2.48e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.083 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

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]  2.996486
sqrt(sum(residuals(model)^2)/18)
[1]  2.996486
```

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

```
> sqrt(sum(residuals(model)^2)/17)
[1]  3.083359
```

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

```
> summary(model)

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

Residuals:
   Min     1Q Median     3Q    Max 
-4.957 -1.407 -0.031  1.374  6.130 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 125.2239     6.5217  19.201 5.82e-13 ***
Age           2.7871     0.4869   5.724 2.48e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.083 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

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)$r.squared
[1] 0.6584257
> cor(class$Age,class$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)$r.squared
[1] 0.6584257
> cor(class$Age,class$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(model)

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

Residuals:
   Min     1Q Median     3Q    Max 
-4.957 -1.407 -0.031  1.374  6.130 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 125.2239     6.5217  19.201 5.82e-13 ***
Age           2.7871     0.4869   5.724 2.48e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.083 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.

It calculates the F value which is given by the variation explained by our model divided by the variation that remains.

Mathematically : $\frac{SS(mean) - SS(fit)/(pfit - pmean)}{SS(fit)/(n - pfit)}$

Pfit= number of parameters in the fit (2 parameters)
Pmean = number of parameters in the mean line (1 parameter)
Note: With only one variable, it provides *exactly* the same result as the t-test for the significance of the coefficient of this variable.

Challenge

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