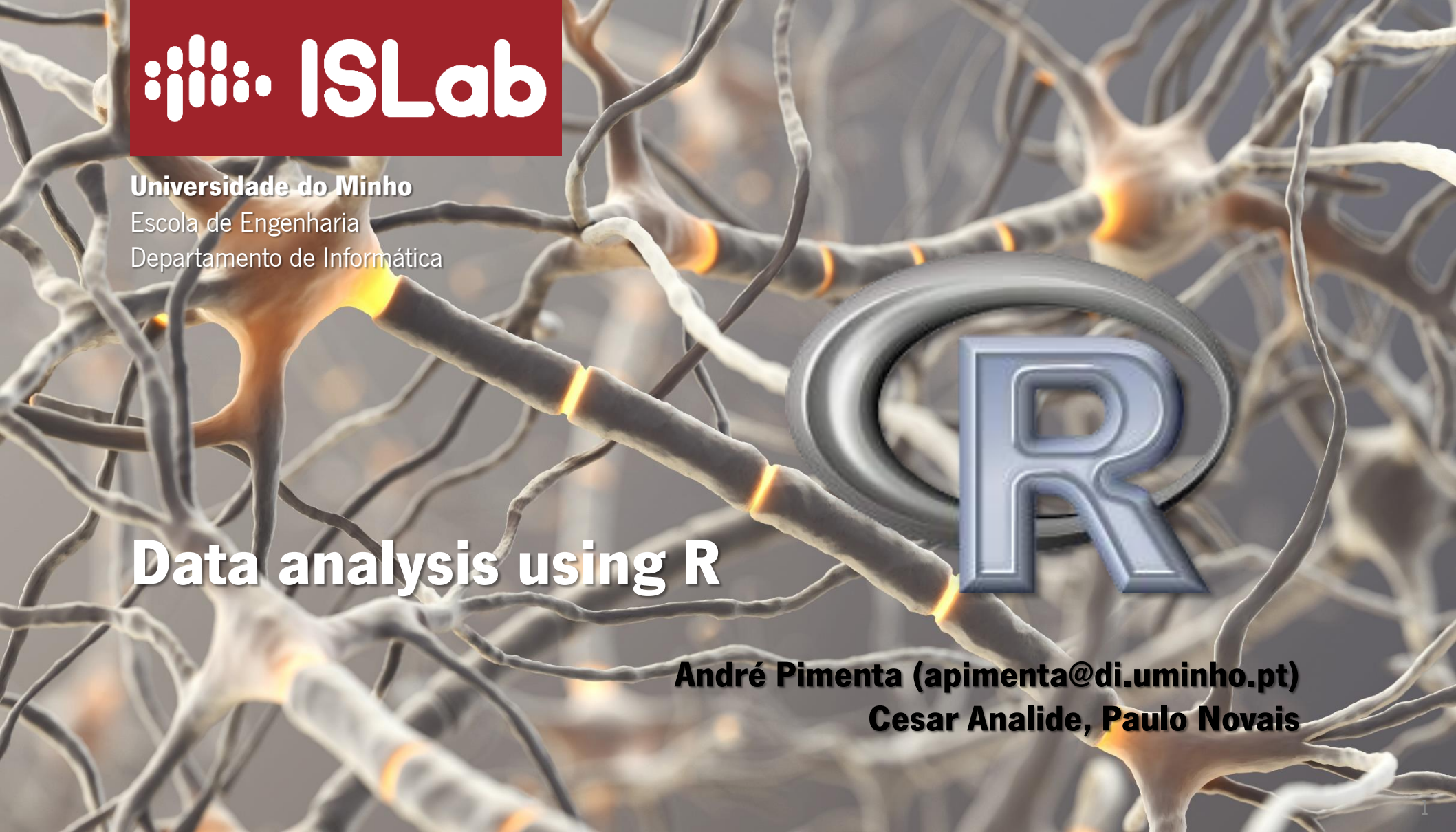


ISLab

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Data analysis using R

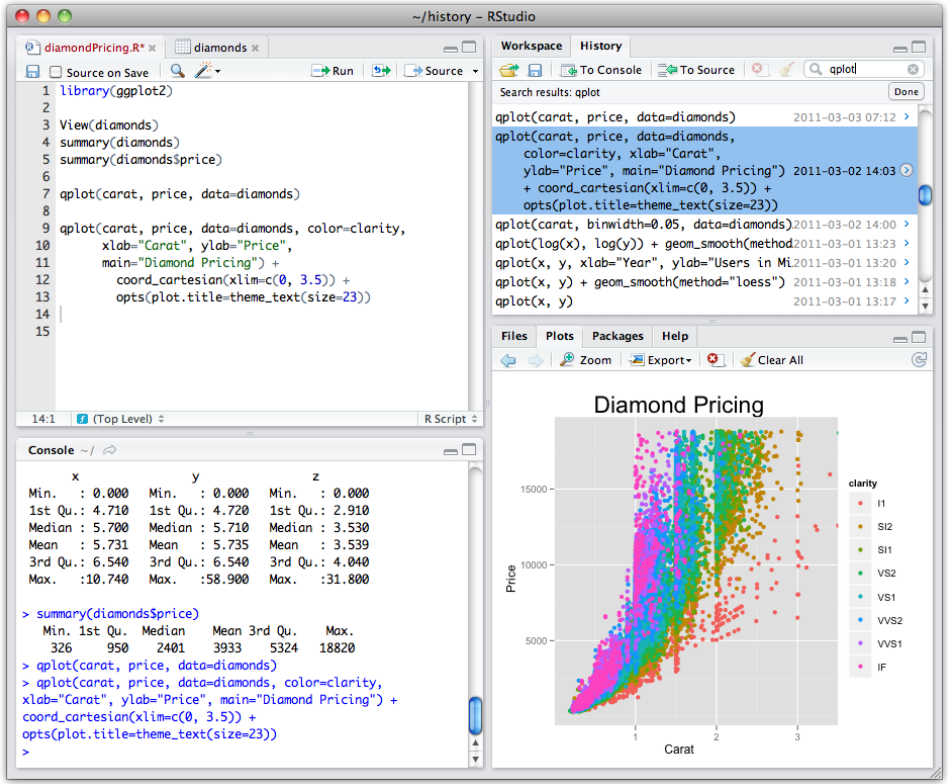
André Pimenta (apimenta@di.uminho.pt)
Cesar Analide, Paulo Novais

Agenda

- What is R?
- Introduction to R
- Introduction to Machine Learning and Data Mining
- Exercises

Data analysis using R

R Programing language



Data analysis using R

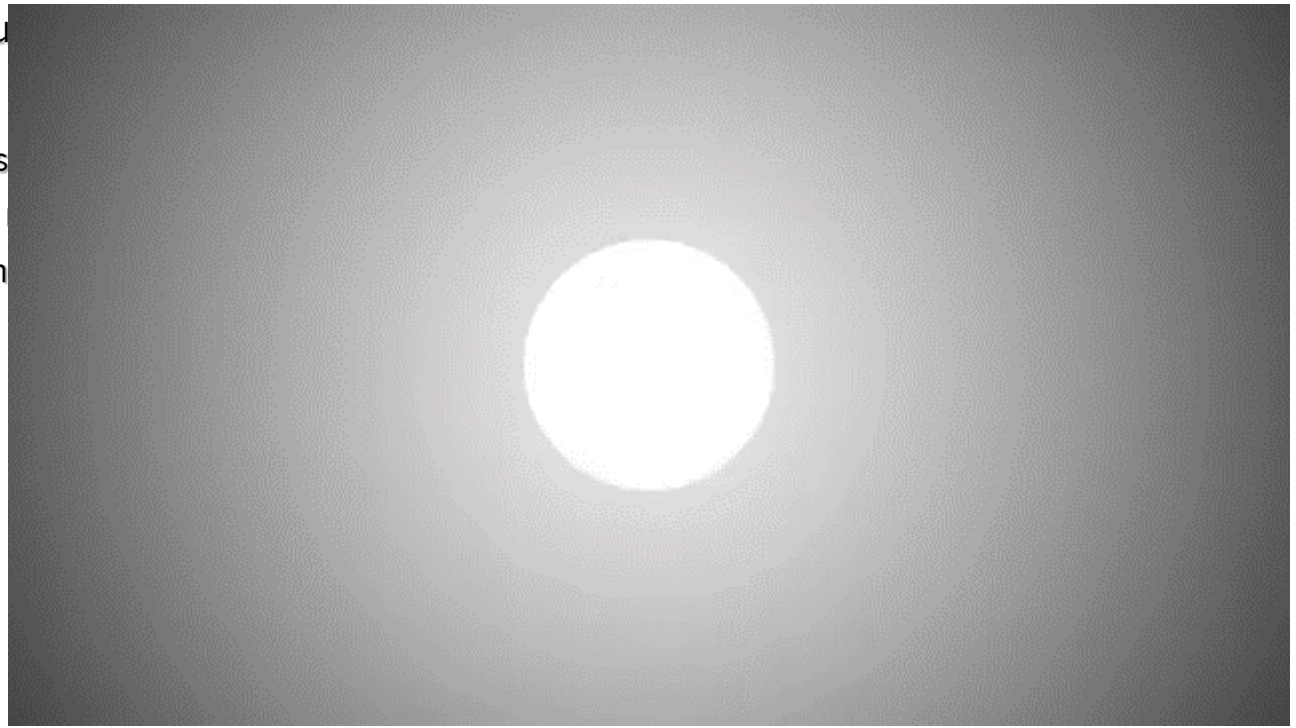
What is R?

- Scripting programming language;
- Open Source;
- A tool for Data Scientists:
  - Statistical analysis;
  - Data mining;
  - Machine Learning.

Data analysis using R

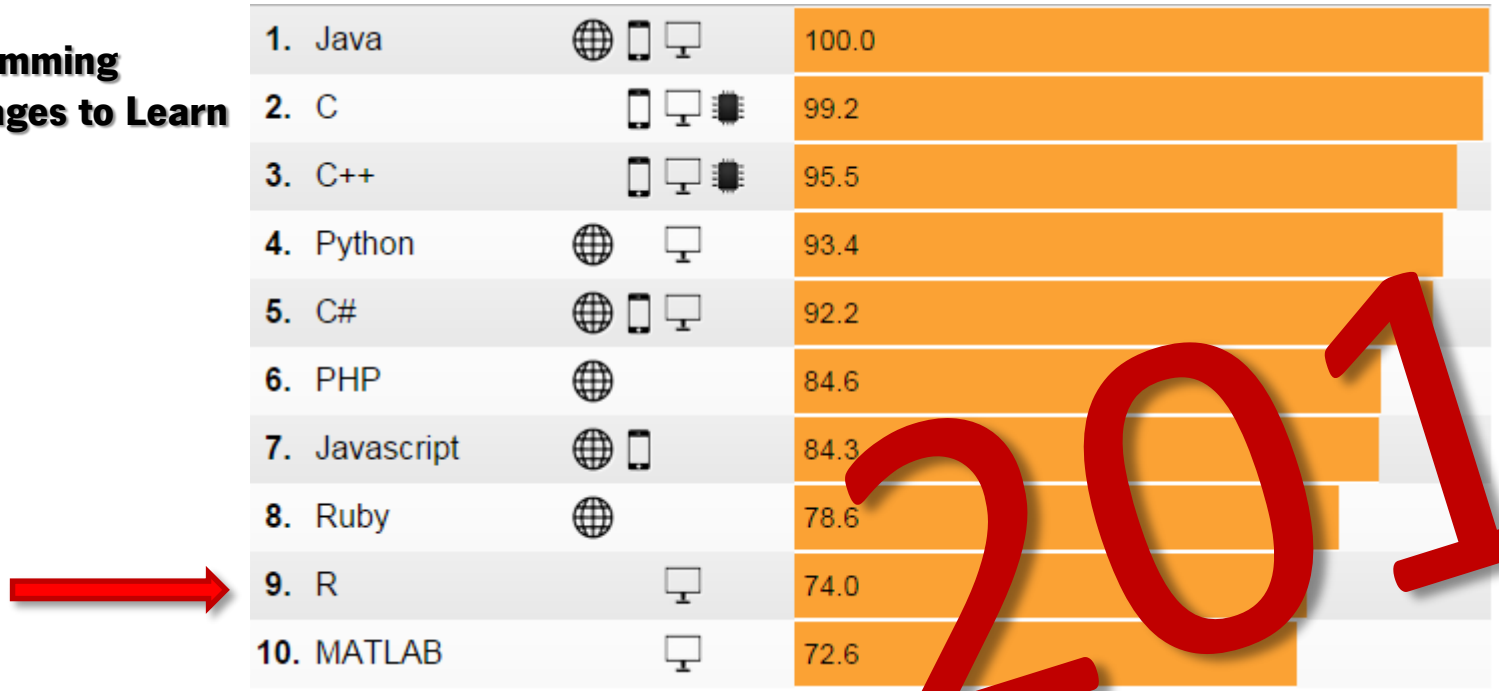
What is R?

- Scripting programming language;
- Open Sou
- A tool for
  - Statis
  - Data
  - Mach



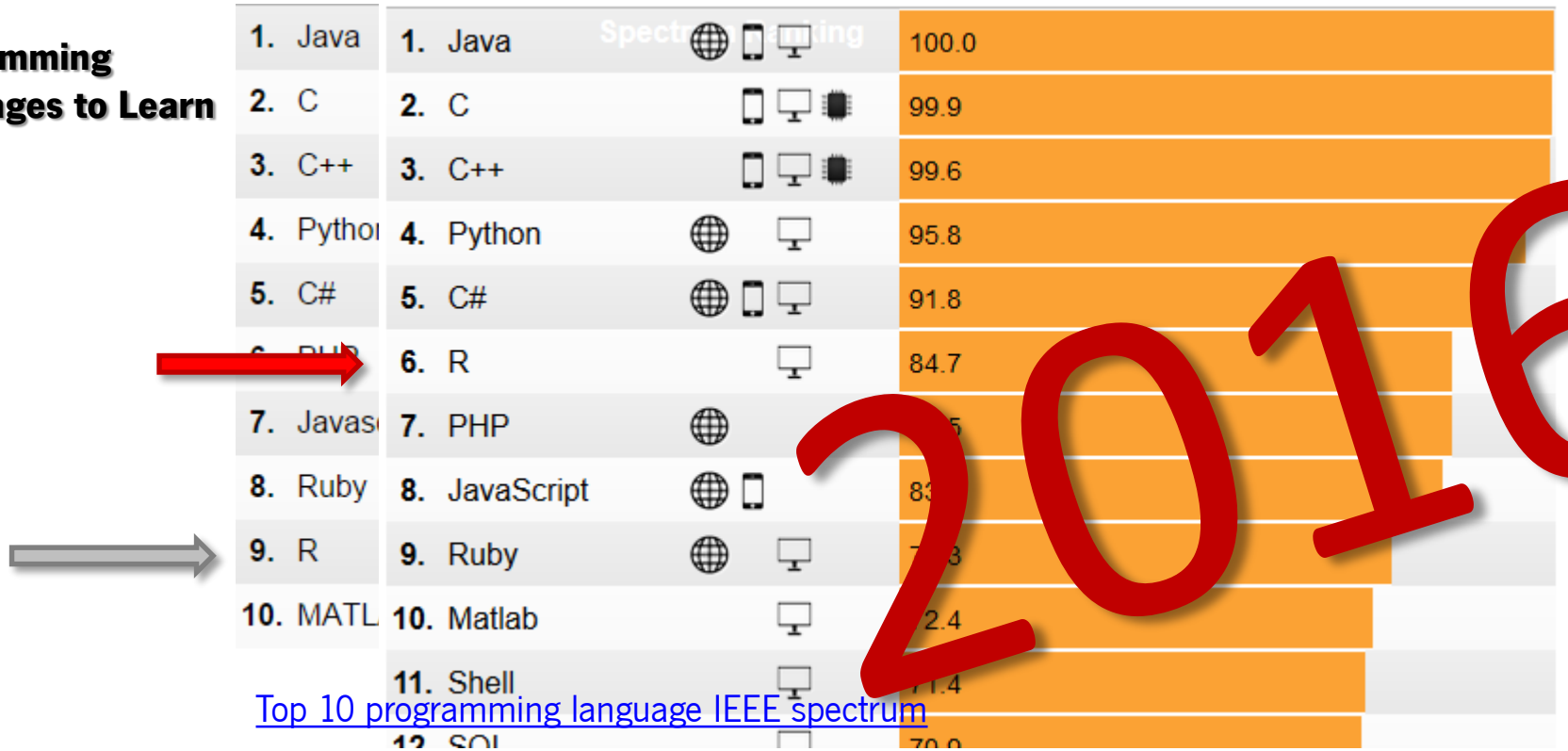
Data analysis using R

Top 10 Programming Languages to Learn



[Top 10 programming language IEEE spectrum]

Top 10 Programming Languages to Learn



[Top 10 programming language IEEE spectrum]

Top 10 Programming Languages to Learn

[Top 10 programming language IEEE spectrum]

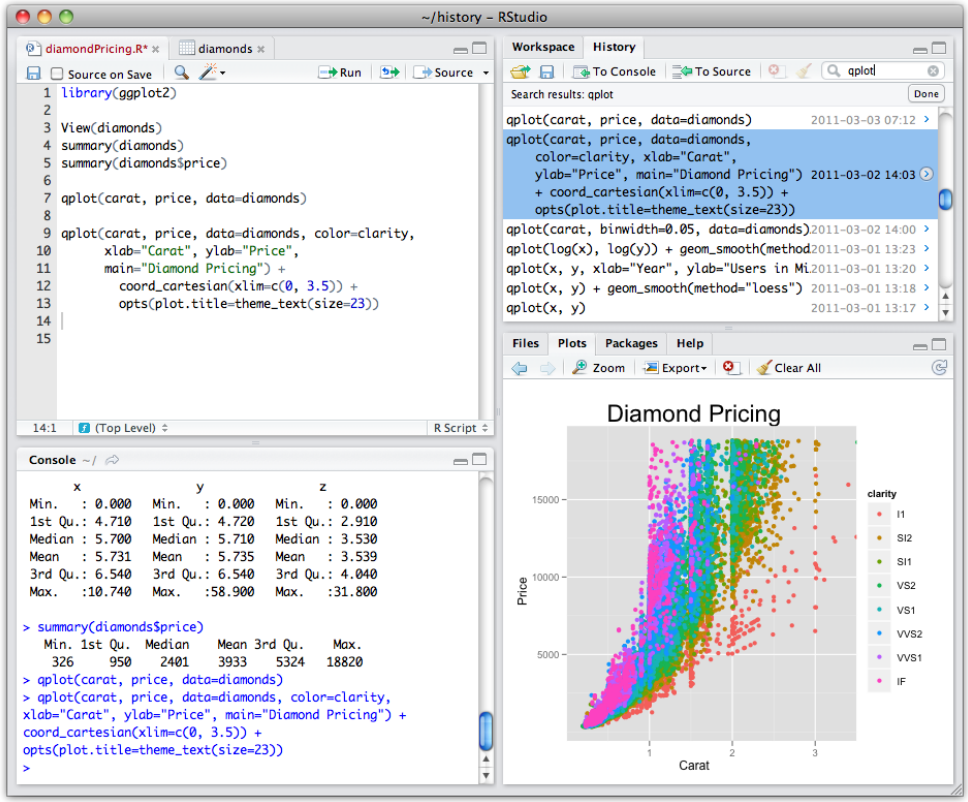
Installation

- Download R:
  - Windows -> http://cran.r-project.org/bin/windows/base/
  - Mac OS X -> http://cran.r-project.org/bin/macosx/
  - Linux (Ubuntu) -> http://cran.r-project.org/bin/linux/ubuntu/README

- Download RStudio:
  - http://www.rstudio.com/products/rstudio/download/

Data analysis using R

Starting with R



Assignment operators

- Assignment operator : **<-**
  - assigns a value to an R-object

```
> x <- 3          # assigns value 3 to object x
> x
[1] 3

> x = 3           # Possible, but may give unexpected results
```

Arithmetic operators

- +, -, /, *, ^

```
> x + x          # addition
[1] 6

> x/2            # division
[1] 1.5

> x^2            # exponentiation
[1] 9
```

Relational operators

- Comparison of values:

```
> x == 3                # x equal to 3
[1] TRUE

> x != 3                # x not equal to 3
[1] FALSE

> x < 3                 # x smaller than 3
[1] FALSE

> x >= 3                # x greater or equal to 3
[1] TRUE
```

Data formats:

vectors (1)

- **c(. . .)** concatenates numbers into a numeric vector:

```
> a <- c(3, 4 ,9)        # vector a
> a
[1] 3 4 9

> class(a)               # what class of vector?
[1] "numeric“

> length(a)              # how many elements?
[1] 3

> a[2]                   # what is 2nd element of a?
[1] 4
```

Data formats: vectors (2)

- **c(. . .)** also concatenates characters into a character vector:

```
> b <- c("cat", "dog")
> b
[1] "cat" "dog“

> class(b)
[1] "character“

> length(b)
[1] 2

> b[2]
[1] "dog"
```

Exercise 1

- Run the following commands:

```
> a <- c(3, 4 ,9)
> b <- c("cat", "dog")
> a+3
> b+3
> a*3
> a*a
> a==4
> (a==4)*a
> a>4
> a[a>4]
```

Functions

- R-functions have the following structure:

  result <- functionname(arg1,arg2, . . .)

  - **result** stores the outcome of the function
  - **arg1, arg2, . . .** are the arguments of the function
  - Some arguments are mandatory, others not (those with default values)

  > a <- c(3,4,9) # for example

  > l <- length(a) # for example

- To open help page type:

  > ?functionname

  > ?c # for example

  > ?length # for example

Exercise 2

- Use functions **length, mean, sum, var** to obtain for vector **a**:
  - a) The number of elements
  - b) The mean
  - c) The sum
  - d) The variance

- Use **?length, ?mean, ?sum, ?var** to see help page;

Generate vectors (1)

- R has several functions to generate vectors:
  - **seq( )** yields a sequence of numbers:

    seq(from = 1, to = 1, by = ..., length.out = NULL)

Exercise 3

- Check help page:

```
> ?seq                    # for help page
```

- Run these commands and see if you understand them:

```
> seq(from=1, to=5)
> seq(5)
> 1:5
> seq(1, 5, by=2)
> seq(1, 5, length.out=9)
```

Generate vectors (2)

- **rep( )** repeats numbers and/or vectors:
  - **rep(x, times=1, each=1)**
    - **x** is a number or vector
    - **times** is the number of replications of **x** (default = 1)
    - **each** is the number of replications of the element of **x** (default = 1)

Exercise 4

- Check help page:

  > ?rep # for help page

- Run the following commands and see if you understand:

  > rep(1, times=2)

  > rep(1:4, times=2)

  > rep(1:4, each=2)

  > rep(1:4, times=2, each=2)

  > rep(1:4, 1:4)

Data analysis using R

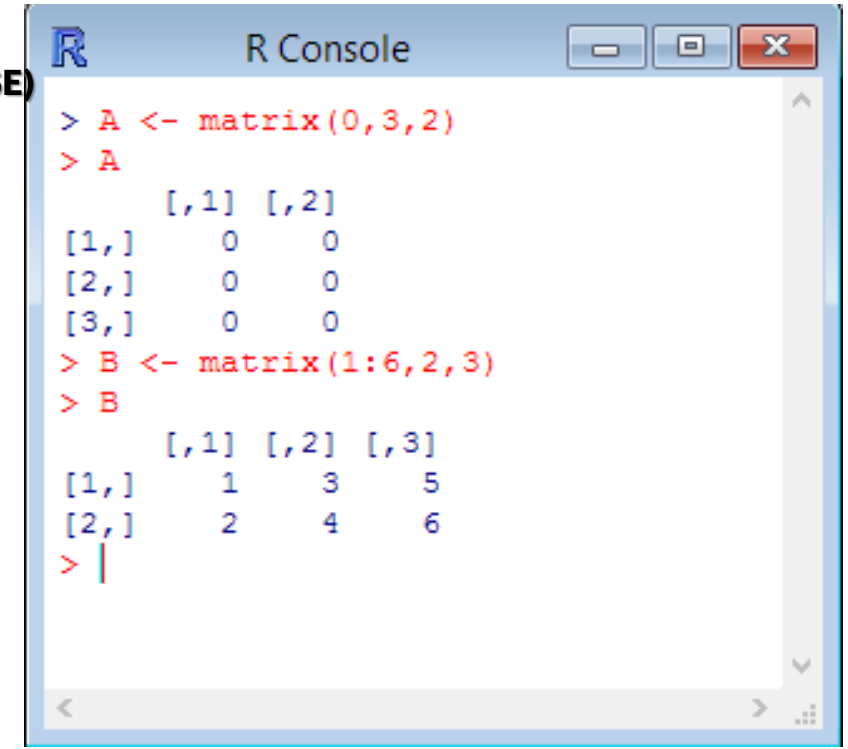
Data formats:

matrix

- A matrix is a 2-dimensional array:

  matrix(x, nrow=1, ncol=1, byrow=FALSE)

  - **x** can be a number or a vector
  - **nrow** and **ncol** are dimensions
  - Default is filled by column

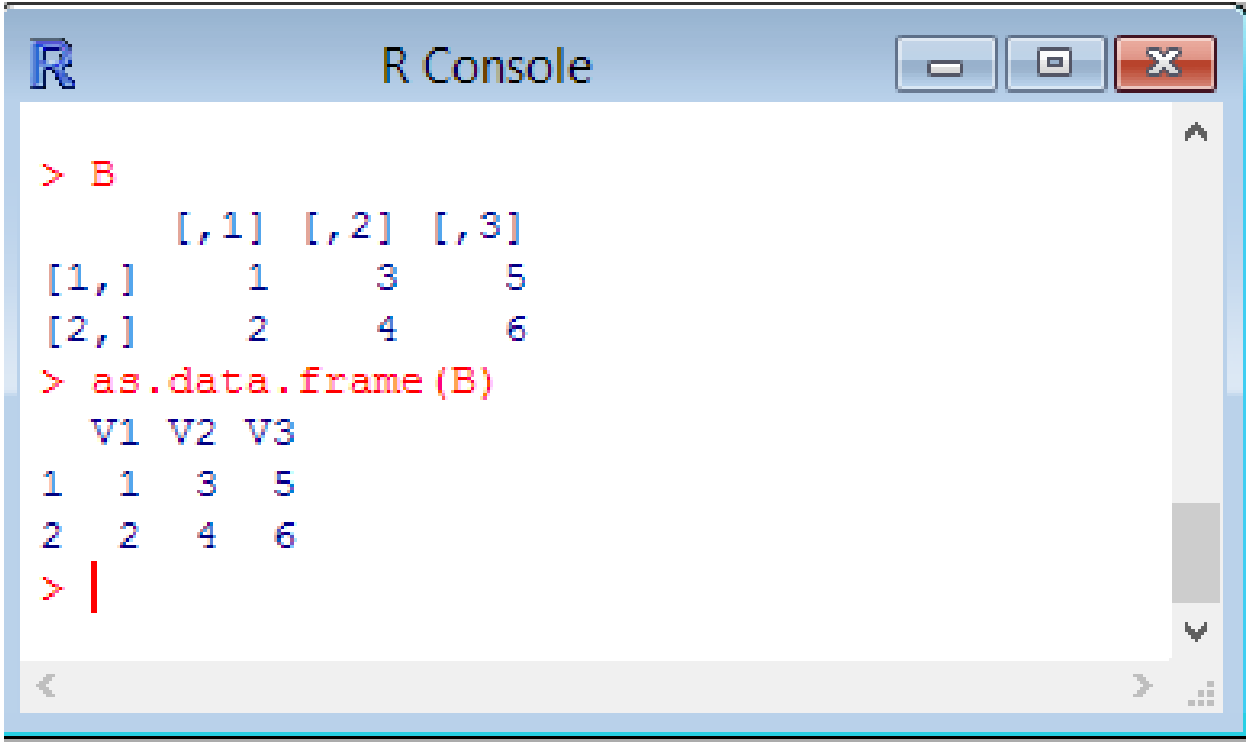


Matrix computations

R Console

```
> B
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> B[1,] # 1st row of B
[1] 1 3 5
> B[2,3] # element in 2nd row, 3rd column
[1] 6
> 2*B+10 # applies to each element of B
     [,1] [,2] [,3]
[1,]   12   16   20
[2,]   14   18   22
> B[,1]<-0 # set 1st column to zero
> B
     [,1] [,2] [,3]
[1,]    0    3    5
[2,]    0    4    6
>
```

Data frames



Data analysis using R

Data frames examples

- R contains many built-in data sets:
  - For an overview, type:

    > data( )

R data sets

```
Data sets in package 'datasets':

AirPassengers
                Monthly Airline Passenger
                Numbers 1949-1960
BJsales         Sales Data with Leading
                Indicator
BJsales.lead (BJsales)
                Sales Data with Leading
                Indicator
BOD             Biochemical Oxygen Demand
CO2             Carbon Dioxide Uptake in
                Grass Plants
ChickWeight     Weight versus age of chicks
                on different diets
```

  - We will look at **chickwts** (further down the list)

    > dataset$variable # extracts a variable from a dataset

    > chickwts # shows the dataset

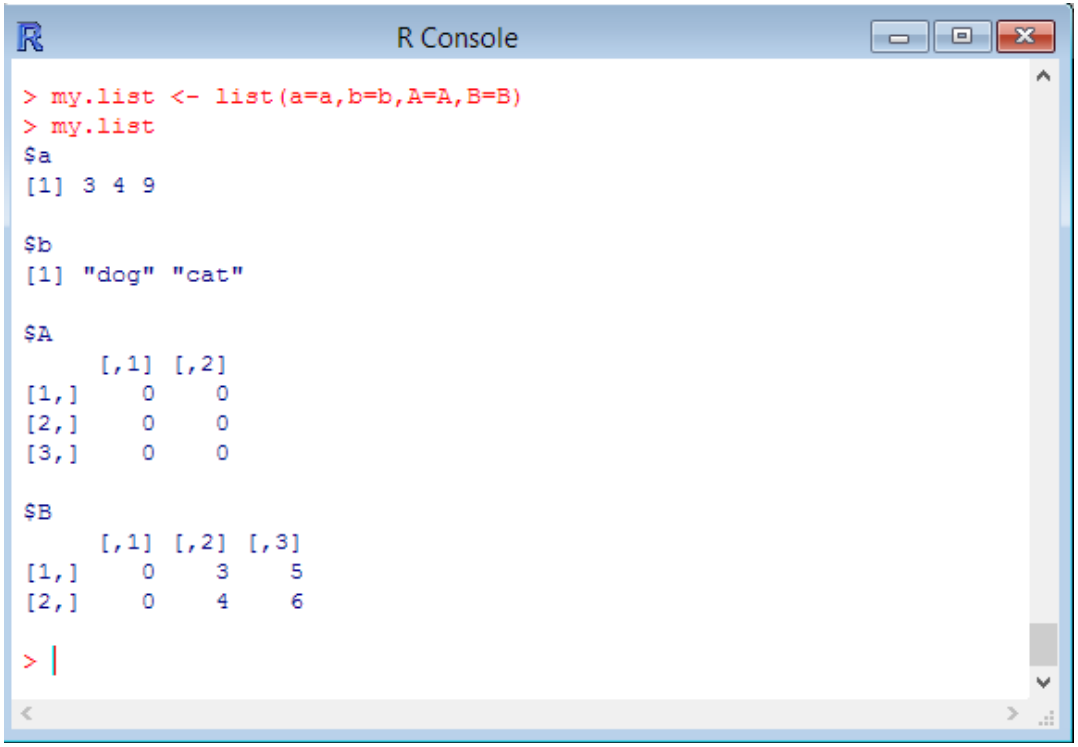
    > chickwts$feed # extracts the variable **feed**

Data analysis using R

Data formats:

lists

- Lists are used to store all kinds of R-objects:
  - Vectors;
  - Matrices;
  - Formulas;
  - etc.



Data analysis using R

Summarizing data

- **summary( )** is a function to summarize R objects (including data frames)

```
> summary(chickwts)                    # provides a summary of the variables
     weight           feed
 Min.   :108.0   casein   :12
 1st Qu.:204.5   horsebean:10
 Median :258.0   linseed  :12
 Mean   :261.3   meatmeal :11
 3rd Qu.:323.5   soybean  :14
 Max.   :423.0   sunflower:12
```

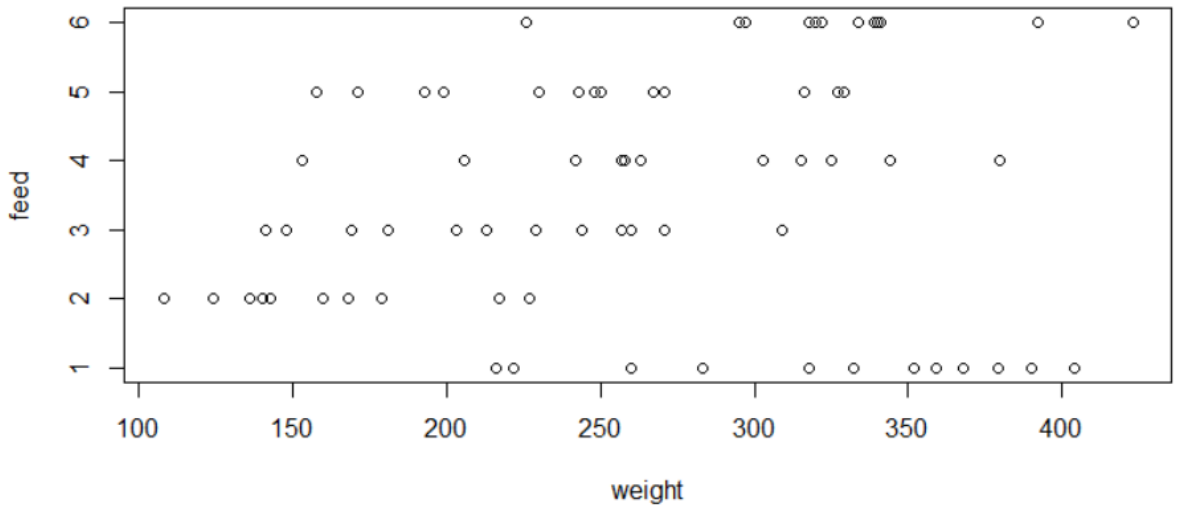
- Note that R recognizes the class of the variables and summarizes them correctly

Plotting data

- plot()is the basic R-function for making plots:
  - You can plot a data frame:

> plot(chickwts)

- Result is scatter plot
- **feed** treated as numerical



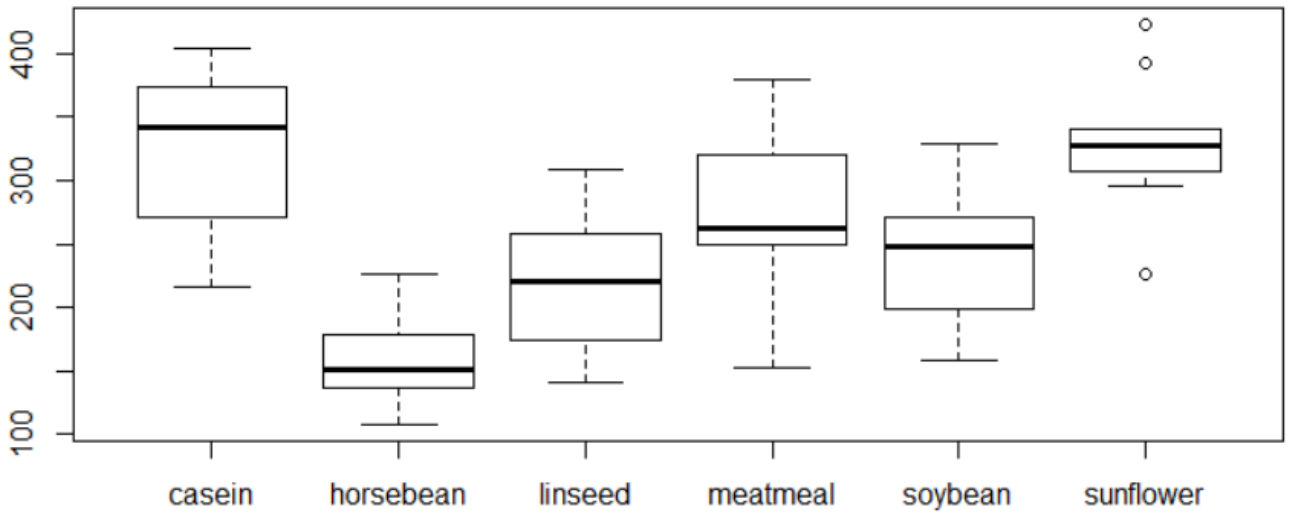
Formulas

- Formulas are used to specify statistical models;
- The formula operator is the **~** sign:
  - **y ~ x** # y as a function of x
  - **y ~ x + z** # y as a function of x and z
  - **y ~ x*z** # y as a function of x, z and xz

Exercise 6

- For the data set **chickwts**:
  - Plot **weight** as a function of **feed**

```
> plot(chickwts$feed,chickwts$weight)
```



Machine Learning

Machine Learning

Techniques

- **Classification**:
  - predict class from observations
- **Clustering**:
  - group observations into “meaningful” groups
- **Regression** (Prediction):
  - predict values from observations

Classification

- Classify a document into a predefined category;
  - documents can be text, images...
- Some examples are:
  - Naive Bayes Classifier, KNN, SVM;
- Example:
  - Features: Humidity, Temperature, Season;
  - Classifies if it rains or not;

Clustering

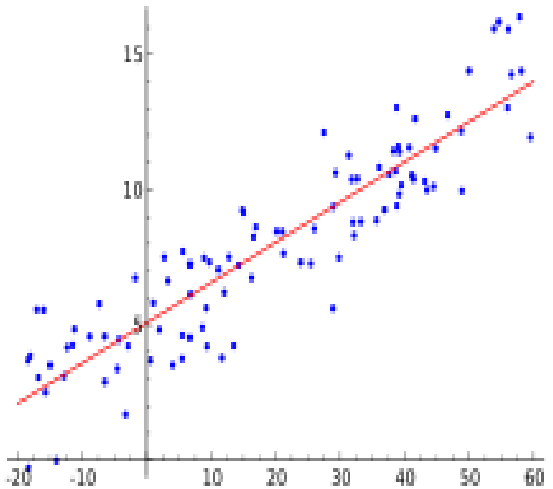
- The task of grouping a set of objects in such a way that
  - objects in the same group (called a **cluster**) are more similar to each other;
  - objects in other groups are different from each other;
- Objects are not predefined;
- For example, these keywords:
  - "man's shoe"
  - "women's shoe"
  - "women's t-shirt"
  - "man's t-shirt"

can be clustered into 2 categories "shoe" and "t-shirt" or "man" and "women"

- Popular clustering algorithms are
  - K-means clustering;
  - Hierarchical clustering;

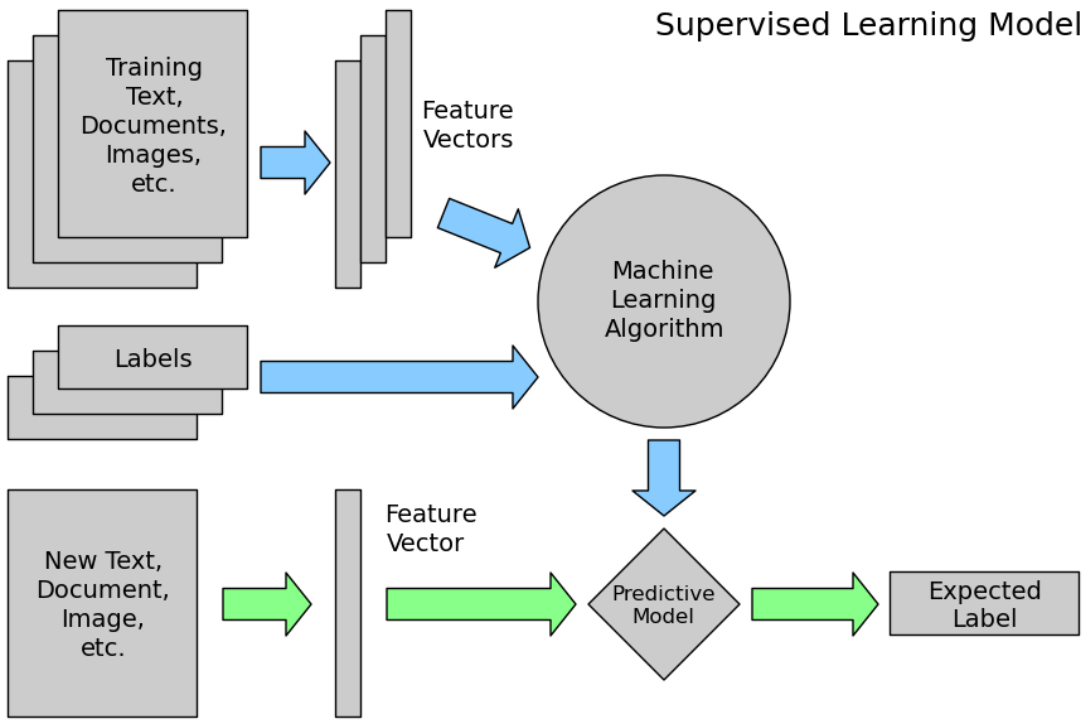
Regression

- A measure of the relation between the mean value of one variable (e.g. output) and corresponding values of other variables (e.g. time and cost);
- **Regression analysis** is a statistical process for estimating the relationships among variables;
- Regression means to **predict** the output value using **training** data;
- Some examples are:
  - Logistic Regression (binary regression),
  - Artificial Neural Networks.



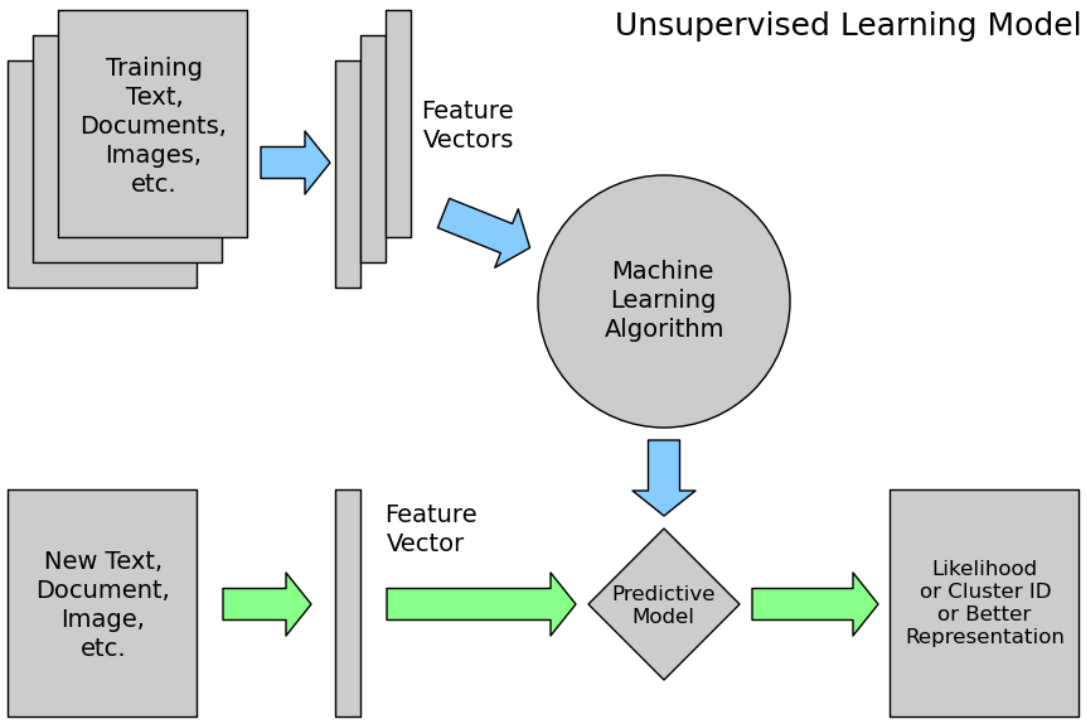
Machine Learning

Supervised Learning



Machine Learning

Unsupervised Learning



Machine Learning

Application areas

- Spam e-mail Detection;
- Machine Translation (Language Translation);
- Image Search (Similarity);
- Clustering (KMeans): Amazon Recommendations;
- Classification: Google News;
- Text Summarization: Google News;
- Rating a Review/Comment: Yelp;
- Fraud Detection: Credit card providers;
- Decision Making: Bank/Insurance sectors;
- Sentiment/Mood Analysis;
- Speech Understanding: Siri's iPhone;
- Face Detection: Facebook's Photo tagging;

http://how-old.net/

Exercise A

- Regression using Artificial neural networks (ANN);
- Problem: Credit scoring:
  - Selecting the correct independent variables (e.g. income, age, gender);
  - Variables:
    - clientId,
    - icome,
    - age,
    - loan,
    - LTI (the Loan To Yearly income ratio),
    - default10yr
  - Creditworthiness = f(income, age, gender, ...)
  - Whether or not a default will occur within 10 years?

Exercise B

- Use infert dataset from default datasets: Infertility after Spontaneous and Induced Abortion
  - 248 observations and 8 variables:
    - education,
    - age,
    - parity,
    - induced,
    - case
    - spontaneous,
    - stratum,
    - pooled.stratum
  - Formula: case~age+parity+induced+spontaneous

```
> trainset <- dataset[1:240, ] ## extract a set to train the NN
> testset <- dataset[70:90, ] ## select the test set
```

Contactos

- Universidade do Minho
- Escola de Engenharia
- Departamento de Informática
- http://islab.di.uminho.pt
- DI-3.22