



Data analysis using R

ANDRÉ PIMENTA – APIMENTA@DI.UMINHO.PT

CESAR ANALIDE - ANALIDE@DI.UMINHO.PT

PAULO NOVAIS – PJON@DI.UMINHO.PT

Agenda

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

R Programming language



The screenshot shows the RStudio interface with the following components:

- Code Editor:** Shows a script named `diamondPricing.R*` containing R code for data exploration and visualization.
- Console:** Displays the output of the R code, including summary statistics for the `carat` and `price` variables.
- History:** A list of previous R commands and their execution times.
- Plots:** A scatter plot titled "Diamond Pricing" showing Price vs. Carat, with points colored by clarity (I1, SI2, SI1, VS2, VS1, VVS2, VVS1, IF).

```
library(ggplot2)
View(diamonds)
summary(diamonds)
summary(diamonds$price)
qplot(carat, price, data=diamonds)
qplot(carat, price, data=diamonds, color=clarity,
      xlab="Carat", ylab="Price",
      main="Diamond Pricing") +
  coord_cartesian(xlim=c(0, 3.5)) +
  opts(plot.title=theme_text(size=23))
qplot(carat, binwidth=0.05, data=diamonds)
qplot(log(x), log(y)) + geom_smooth(method=loess)
qplot(x, y, xlab="Year", ylab="Users in Mi")
qplot(x, y) + geom_smooth(method="loess")
qplot(x, y)
```

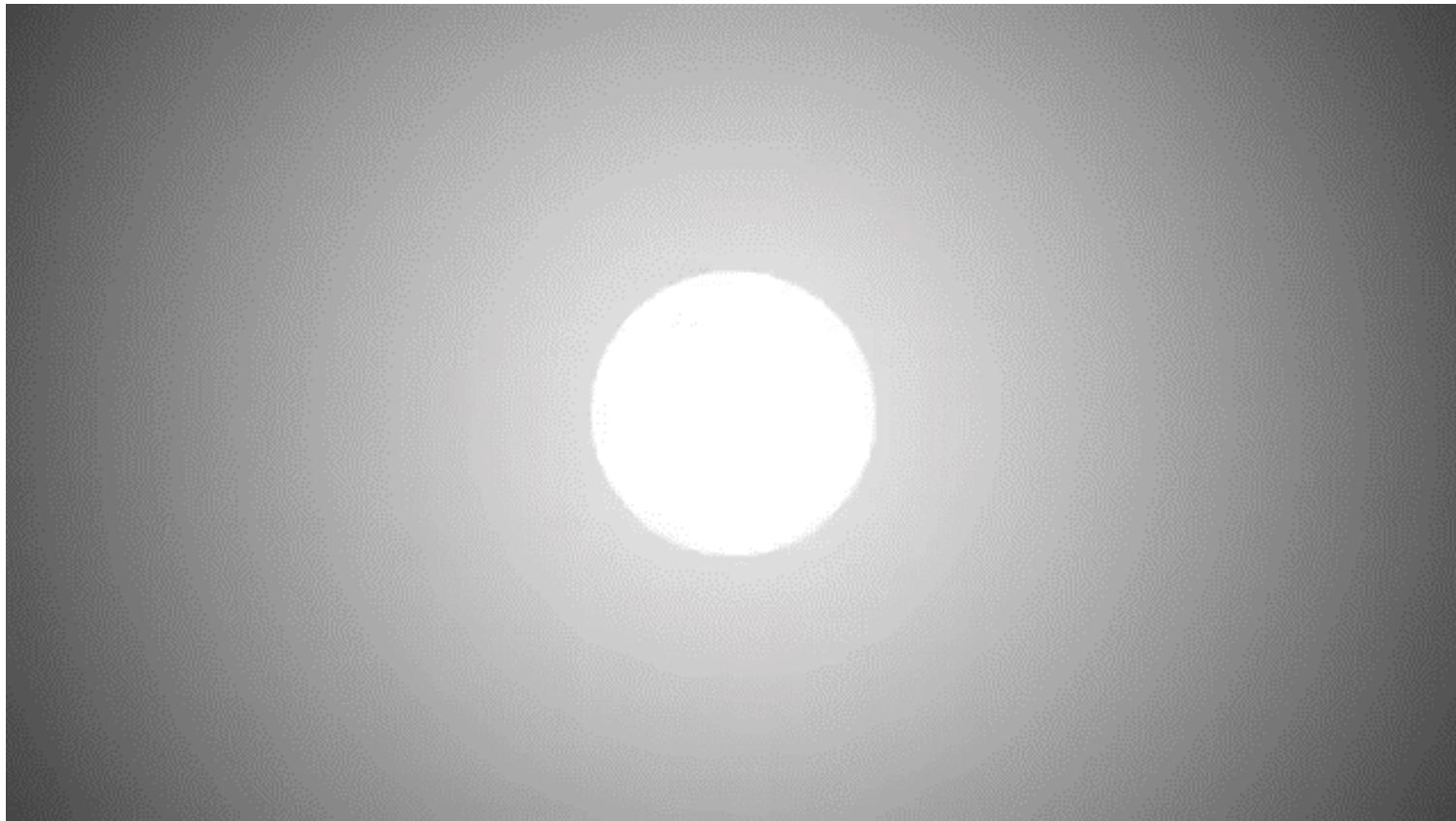
```
14:1 f (Top Level) R Script
Console ~ / 
x          y          z
Min. : 0.000 Min. : 0.000 Min. : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.910
Median : 5.700 Median : 5.710 Median : 3.530
Mean   : 5.731 Mean   : 5.735 Mean   : 3.539
3rd Qu.: 6.540 3rd Qu.: 6.540 3rd Qu.: 4.040
Max.  :10.740 Max.  :58.900 Max.  :31.800

> summary(diamonds$price)
    Min. 1st Qu. Median  Mean 3rd Qu.  Max.
      326     950    2401   3933    5324   18820
> qplot(carat, price, data=diamonds)
> qplot(carat, price, data=diamonds, color=clarity,
  +       xlab="Carat", ylab="Price", main="Diamond Pricing") +
  +       coord_cartesian(xlim=c(0, 3.5)) +
  +       opts(plot.title=theme_text(size=23))
>
```

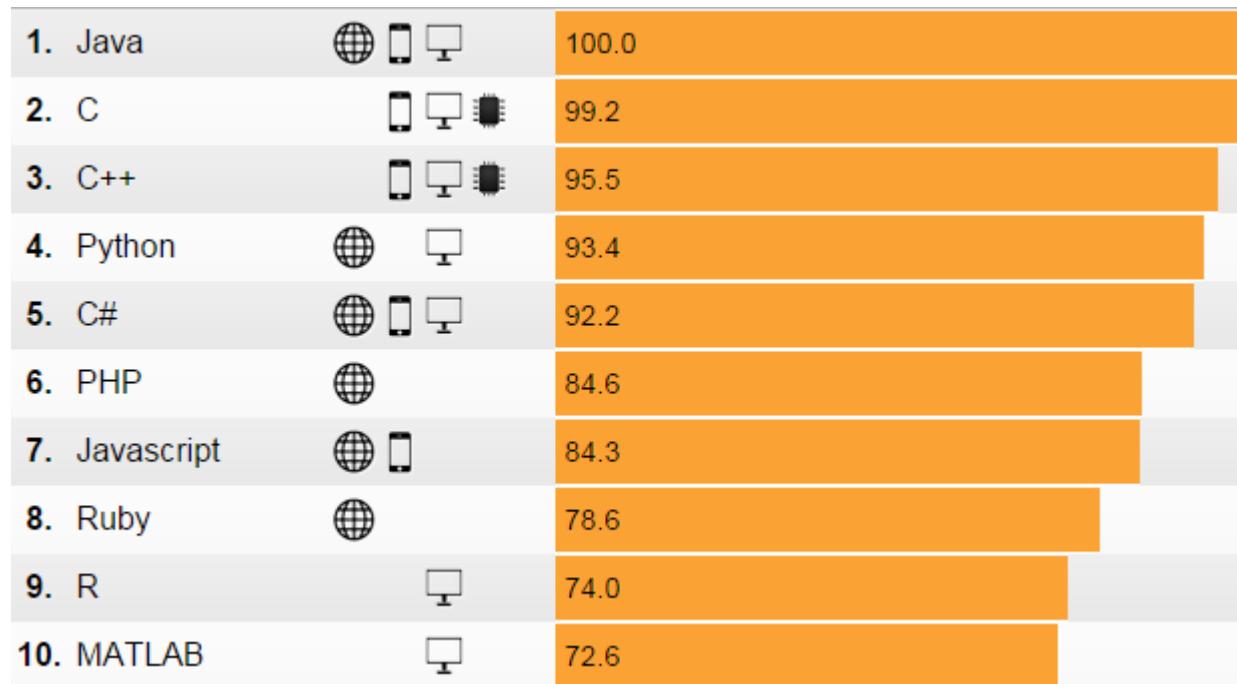
What is R ?

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

What is R ?



Top 10 Programming Languages to Learn



Top 10 programming language IEEE spectrum

Installation

Windows -> <http://cran.r-project.org/bin/windows/base/>

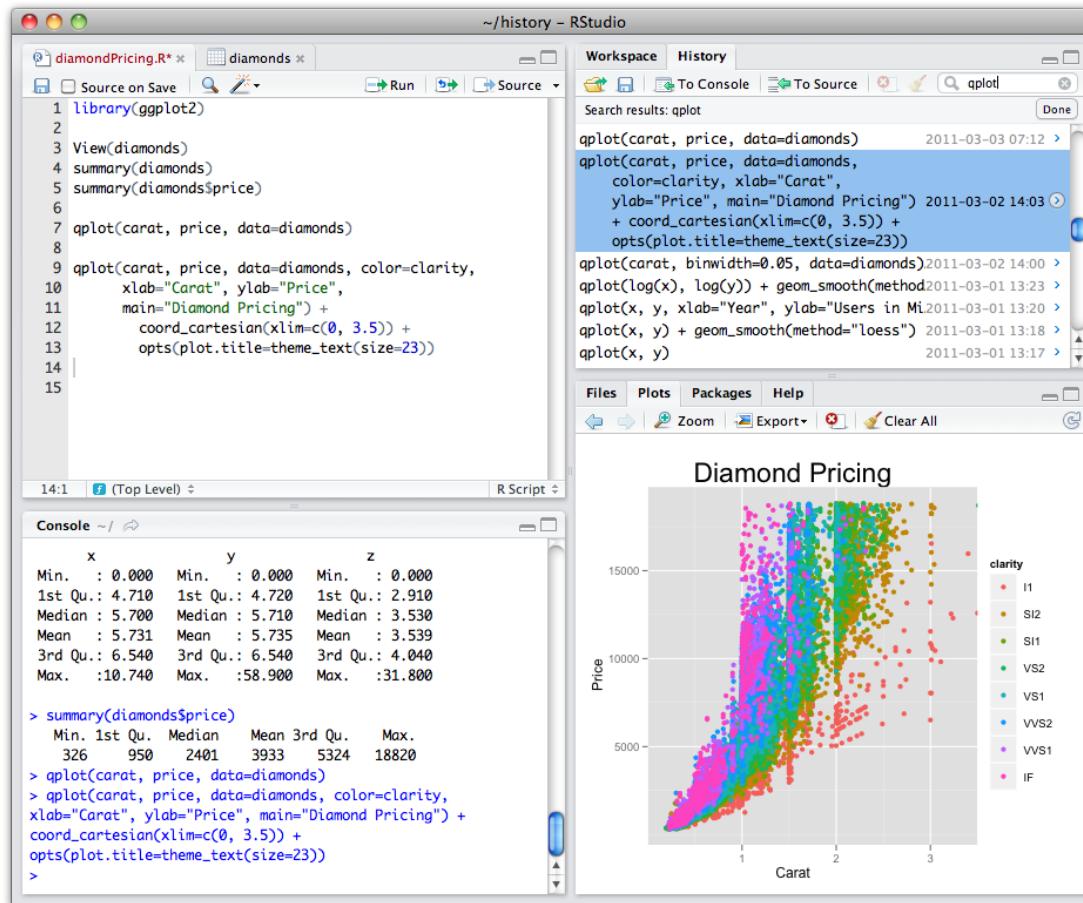
Mac OS X -> <http://cran.r-project.org/bin/macosx/>

Linux (Ubuntu) -> <http://cran.r-project.org/bin/linux/ubuntu/README>

Download Rstudio ->

<http://www.rstudio.com/products/rstudio/download/>

Starting with R



Assignment operator

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`

Logical operators

Tests of (in)equality

```
> 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)

To open help page type:

> ?functionname (help or “?”)

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

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)

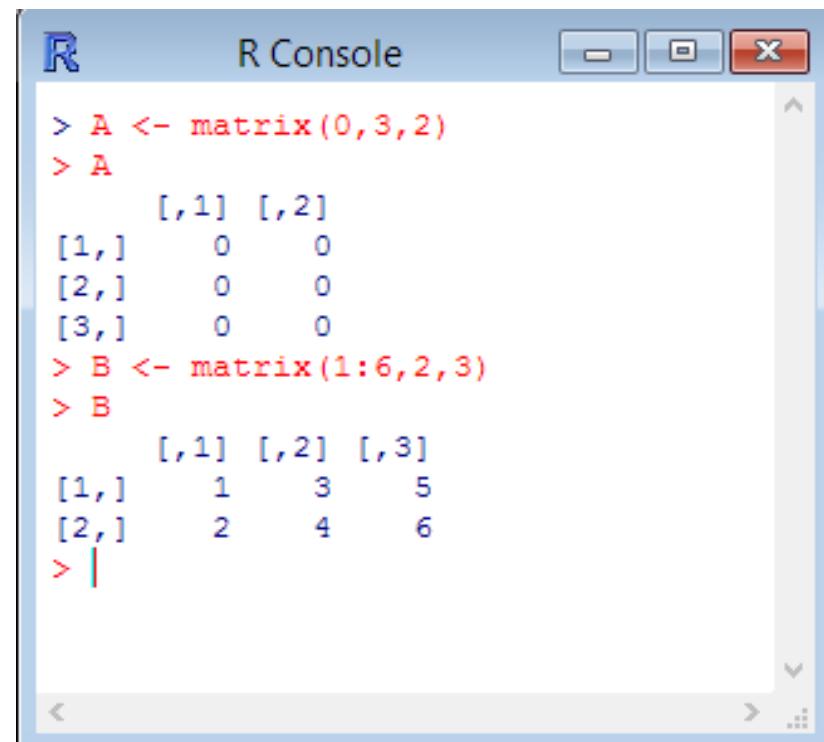
> rep(1:4, 1:4)

Data formats: matrix

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

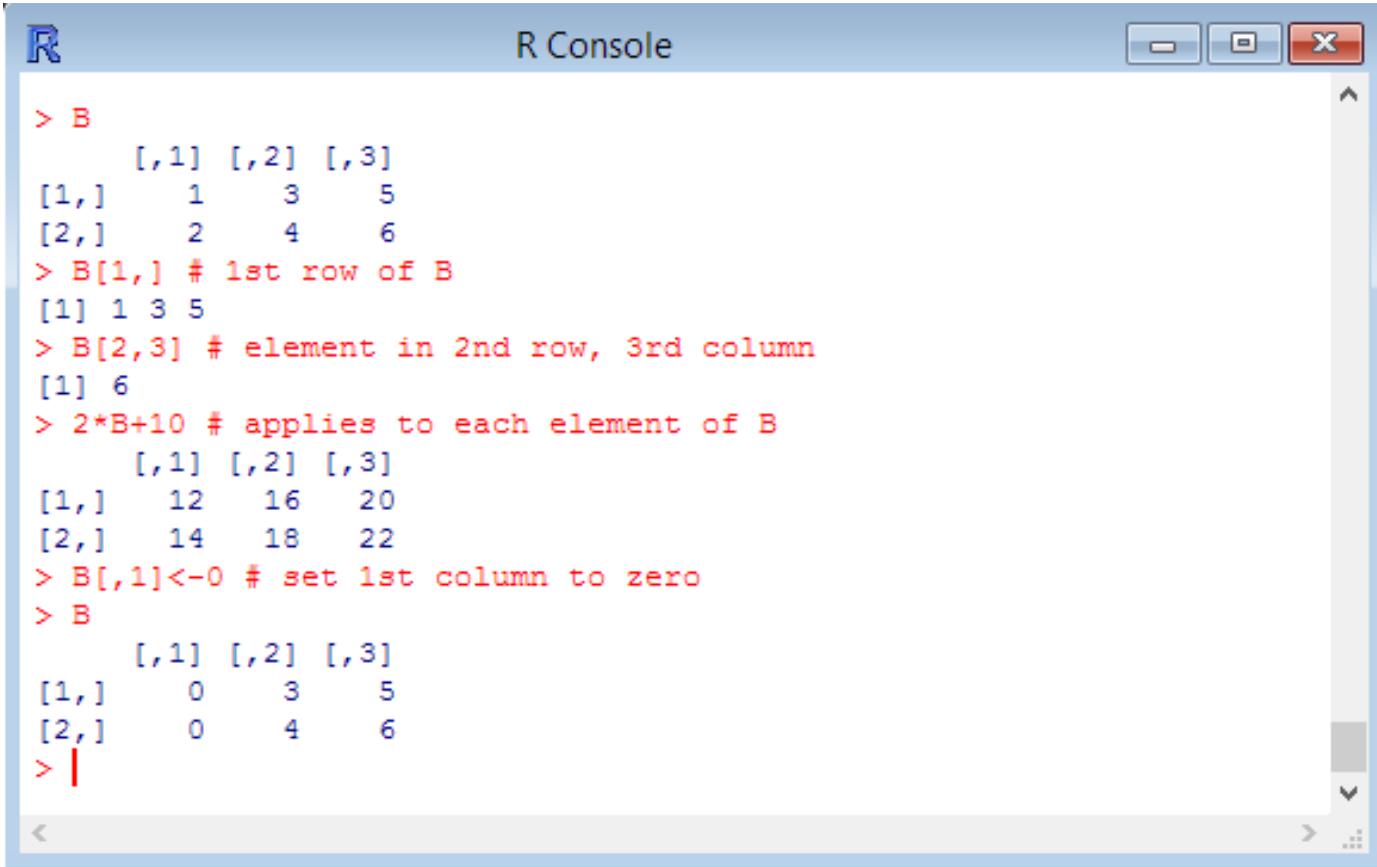


The screenshot shows the R console window with the title "R Console". It displays two matrix definitions and their outputs:

```
> A <- matrix(0,3,2)
> A
     [,1] [,2]
[1,]    0    0
[2,]    0    0
[3,]    0    0
> B <- matrix(1:6,2,3)
> B
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> |
```

The first matrix A is a 3x2 matrix filled with zeros. The second matrix B is a 2x3 matrix filled with integers from 1 to 6.

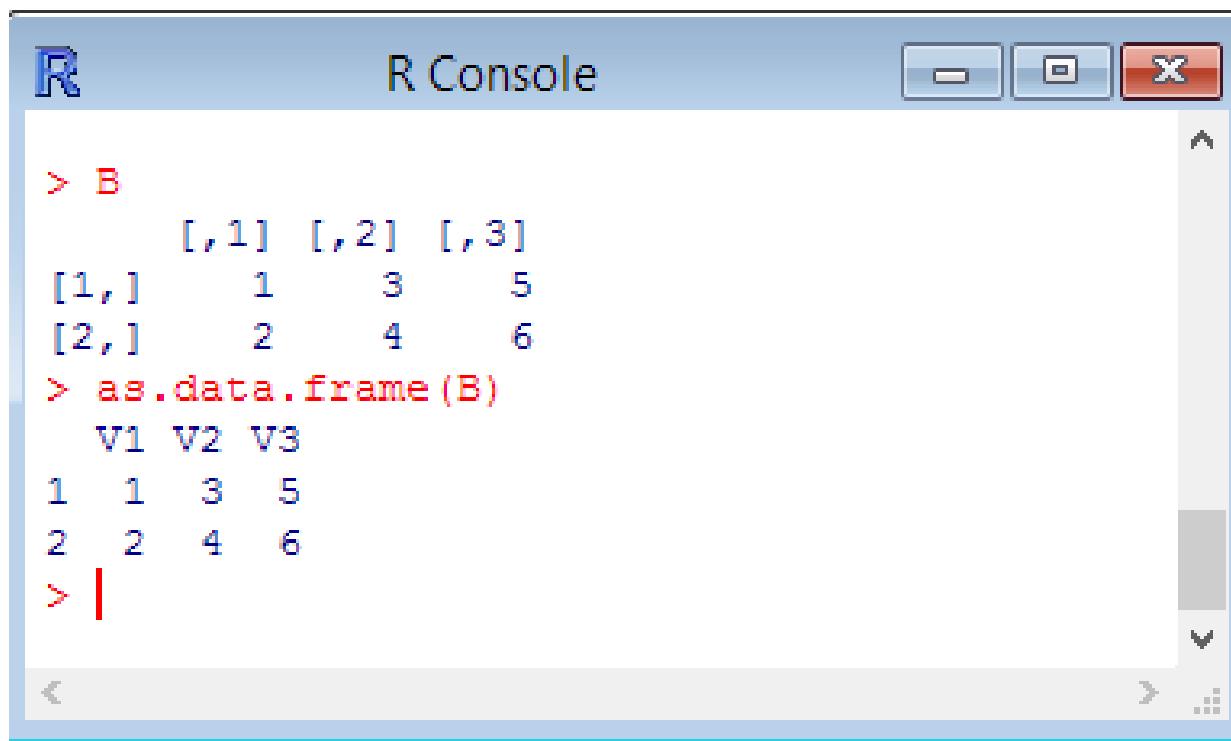
Matrix computations



The screenshot shows the R console interface with the title bar "R Console". The console window displays the following R session:

```
> 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



The screenshot shows the R Console window with the title "R Console". The console displays the following R session:

```
> B
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> as.data.frame(B)
   V1 V2 V3
1  1  3  5
2  2  4  6
> |
```

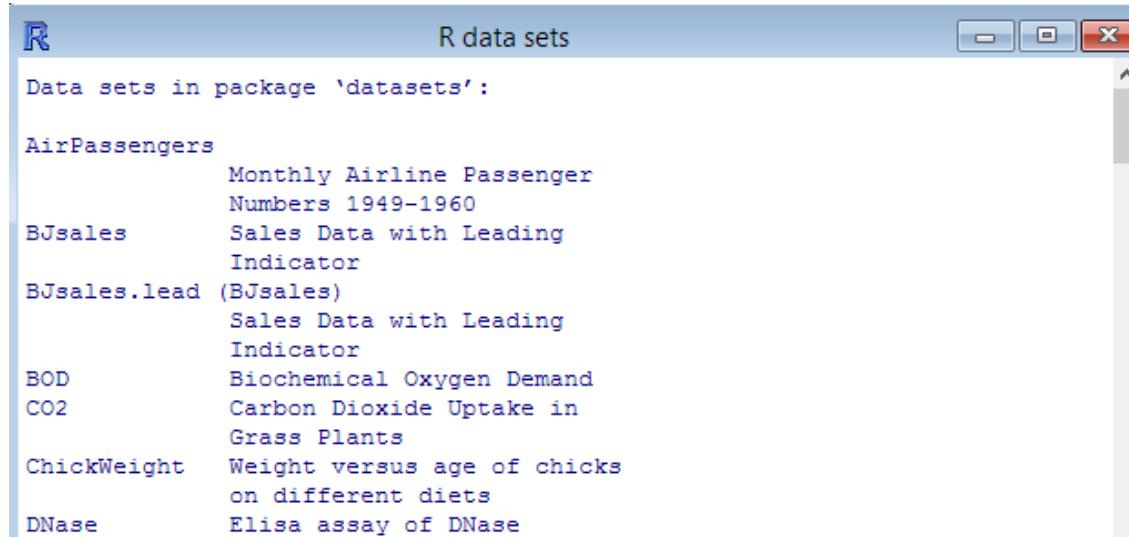
The output shows a matrix B being converted into a data frame. The resulting data frame has columns V1, V2, and V3, with rows corresponding to the matrix rows.

Data frame examples

R contains many build-in data sets

- For an overview, type **> data()**
- We will look at **chickwts** (further down the list)

> dataset\$variable # extracts the variable

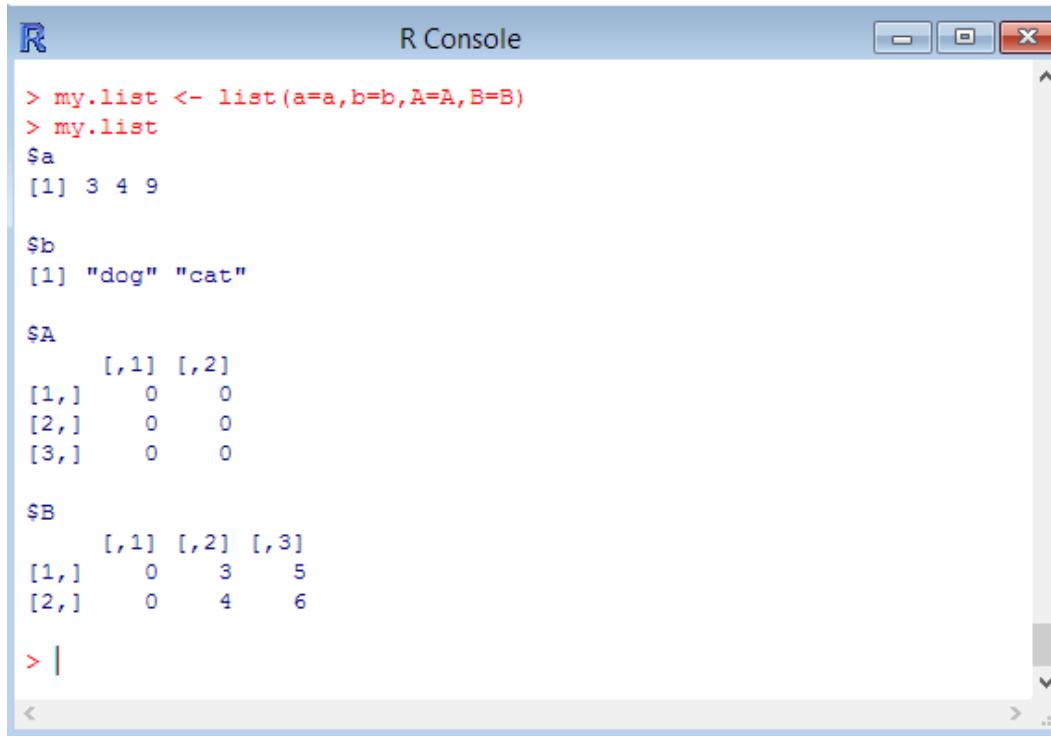


```
R data sets
Data sets in package 'datasets':
AirPassengers Monthly Airline Passenger
                 Numbers 1949-1960
BJsales        Sales Data with Leading
                  Indicator
BJsales.lead (BJsales)
                  Sales Data with Leading
                  Indicator
BOD           Biochemical Oxygen Demand
CO2           Carbon Dioxide Uptake in
                  Grass Plants
ChickWeight   Weight versus age of chicks
                  on different diets
DNase         Elisa assay of DNase
```

Data formats: lists

Lists are used to store all kinds of R-objects

- Vectors, matrices, formulas, ect.



The screenshot shows the R Console window with the title "R Console". The console displays the following R code and its output:

```
> my.list <- list(a=a,b=b,A=A,B=B)
> my.list
$a
[1] 3 4 9

$b
[1] "dog" "cat"

$A
     [,1] [,2]
[1,]     0     0
[2,]     0     0
[3,]     0     0

$B
     [,1] [,2] [,3]
[1,]     0     3     5
[2,]     0     4     6

> |
```

The list `my.list` contains four elements: `a` (a vector of integers [3, 4, 9]), `b` (a vector of strings ["dog", "cat"]), `A` (a 3x2 matrix of zeros), and `B` (a 2x3 matrix with values [0, 3, 5] and [0, 4, 6]).

Summarizing data

summary() is function to summarize R objects (inc. 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

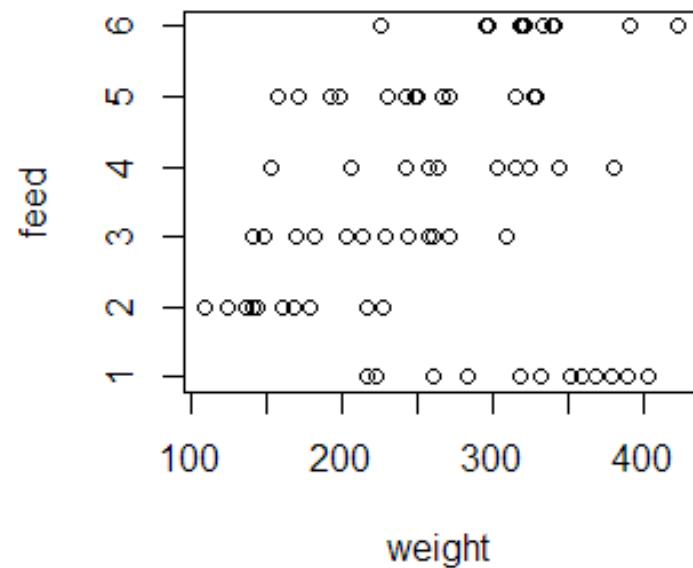
Plotting data

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 operator is the \sim sign

y ~ x # y as a function of x

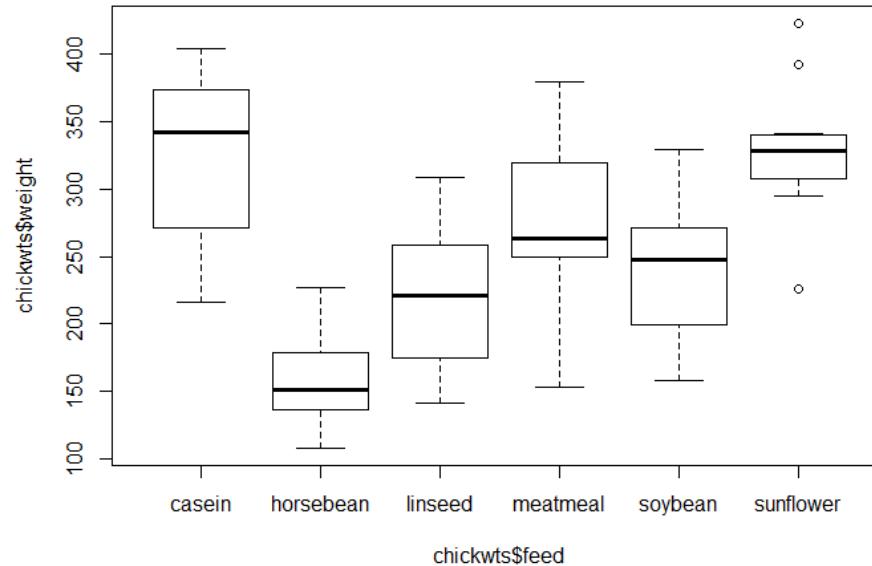
y ~ x + z # y as a function of x and z

y ~ x*z # y as a function of x, z and xz

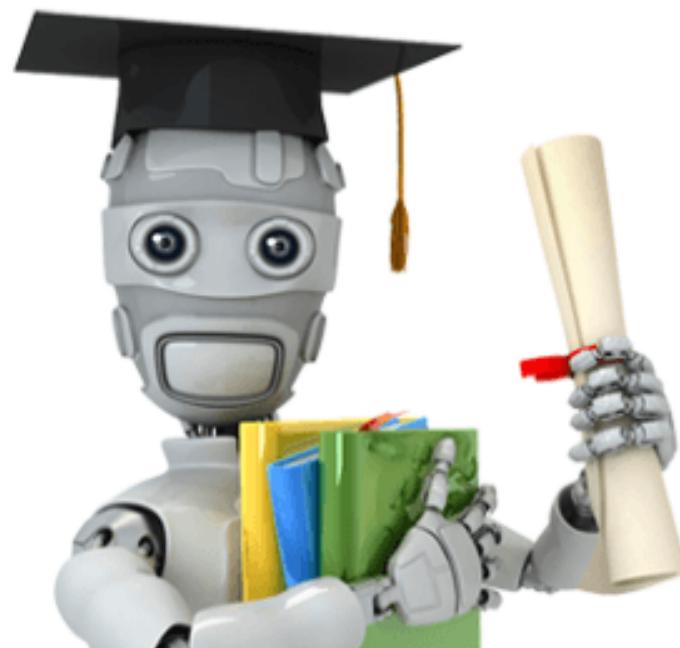
Exercise 6

For the data set **chickwts**

- Plot weight as a function feed



Machine Learning



Techniques

- classification: predict class from observations
- clustering: group observations into “meaningful” groups
- regression (prediction): predict value from observations

Classification

- classify a document into a predefined category.
- documents can be text, images
- Some examples: Naive Bayes Classifier, KNN, SVM .

Example:

- Features: Humidity ,Temperature ,Season
- Classifies if it rains or not

Clustering

clustering is 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 are not predefined

For e.g. these keywords

- “man’s shoe”
- “women’s shoe”
- “women’s t-shirt”
- “man’s t-shirt”
- can be cluster into 2 categories “shoe” and “t-shirt” or “man” and “women”

Popular ones are **K-means clustering** and **Hierarchical clustering**

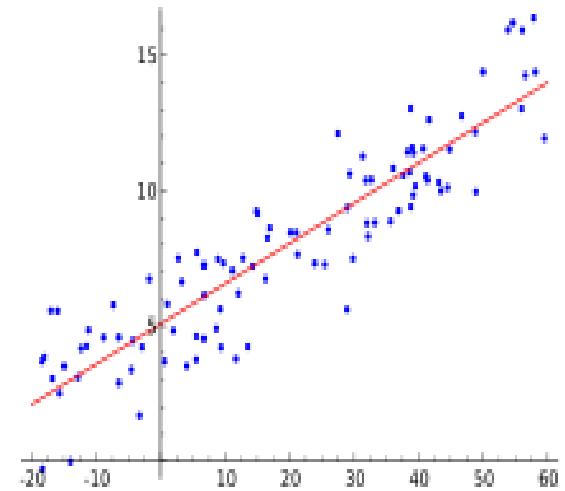
Regression

is 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).

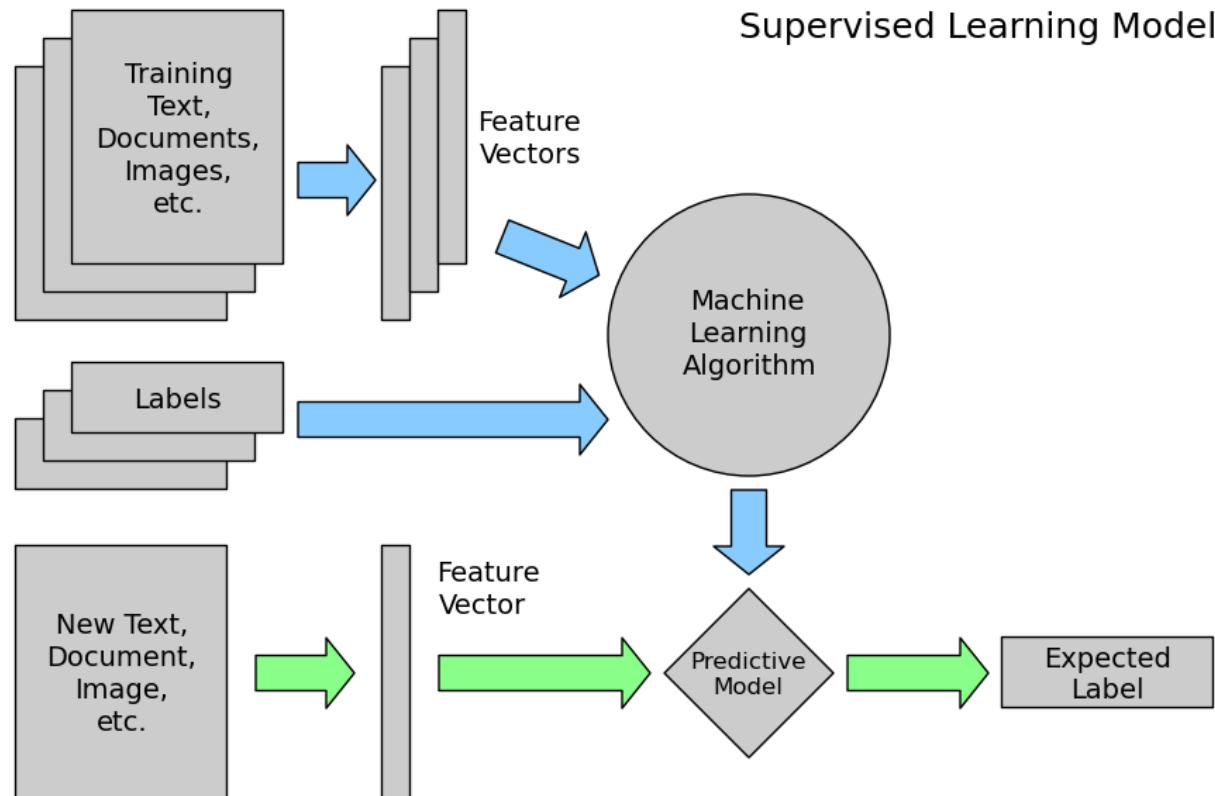
regression analysis is a statistical process for estimating the relationships among variables.

Regression means to **predict** the output value using training data.

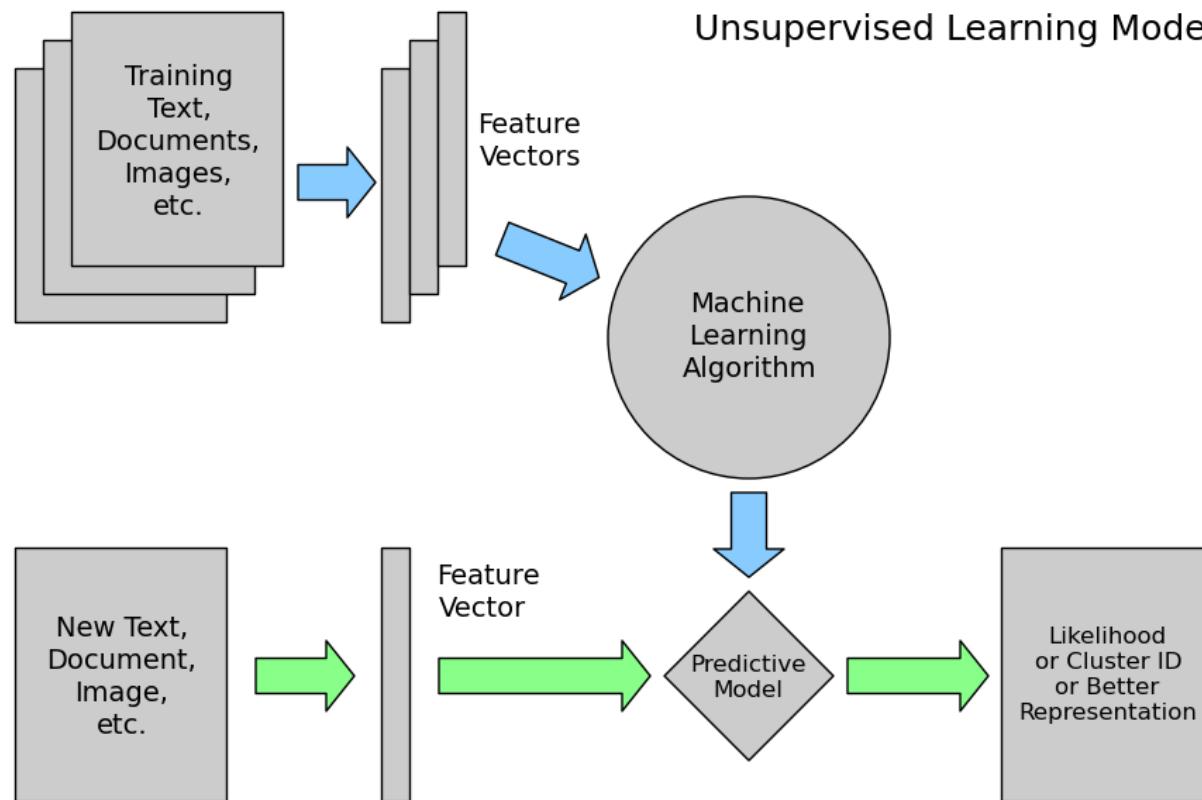
Some examples: Logistic regression (binary regression), artificial neural networks.



Supervised Learning



Unsupervised Learning



Use-Cases

- Spam Email Detection
- Machine Translation (Language Translation)
- Image Search (Similarity)
- Clustering (KMeans) : Amazon Recommendations
- Classification : Google News
- Text Summarization - Google News
- Rating a Review/Comment: Yelp

continued...

Use-Cases (contd.)

- Fraud detection : Credit card Providers
- Decision Making : e.g. Bank/Insurance sector
- Sentiment Analysis
- Speech Understanding – iPhone with Siri
- Face Detection – Facebook's Photo tagging

Example (Age Detection) -><http://how-old.net/>

R and Data Mining (example1)

Text Mining

- Find words related to "#word" on Twitter
- Find relationships between words

R and Machine Learning

Regression using Artificial neural networks (ANN)

Problem: Credit scoring

- selecting the correct independent variables (e.g. income, age, gender)
- Variables (clientId, income, age, loan, LTI- the loan to yearly income ratio, default10yr)
- $\text{creditworthiness} = f(\text{income}, \text{age}, \text{gender}, \dots)$
- whether or not a default will occur within 10 years?

Exercise

Use infert dataset from default datasets

- Description: infert Infertility after Spontaneous and Induced Abortion
- 248 observations and 8 variables
- Variables: "education" "age" "parity" "induced" "case" "spontaneou
s" "stratum" "pooled.stratum"

Formula: case~age+parity+induced+spontaneous

extract a set to train the NN

```
trainset <- dataset[1:240, ]
```

select the test set

```
testset <- dataset[70:90, ]
```