Ένωση Πληροφορικών Ελλάδας

Εισαγωγή στην Αναλυτική Δεδομένων με τη γλώσσα **R**

Χάρης Γεωργίου (MSc, PhD)

https://www.epe.org.gr https://www.hiu.gr

Ένωση Πληροφορικών Ελλάδας

Στόχοι:

- Πρώτος “καθολικός” φορέας εκπροσώπησης πτυχιούχων Πληροφορικής.
- Αρμόδιος φορέας εκπροσώπησης επαγγελματιών Πληροφορικής.
- Αρμόδιος επιστημονικός “συμβουλευτικός” φορέας για το Δημόσιο.
- Αρωγός της Εθνικής Ψηφιακής Στρατηγικής & Παιδείας της χώρας.

Τομείς παρέμβασης

Ποιοι είναι οι κύριοι τομείς παρεμβάσεων της ΕΠΕ;

1. Εθνική Ψηφιακή Στρατηγική & Οικονομία
2. Εργασιακά (ΤΠΕ), Δημόσιος & ιδιωτικός τομέας
3. Παιδεία (Α′, Β′, Γ′)
4. Έρευνα & Τεχνολογία
5. Έργα & υπηρεσίες ΤΠΕ
6. Ασφάλεια συστημάτων & δεδομένων
7. Ανοικτά συστήματα & πρότυπα
8. Χρήση ΕΛ/ΛΑΚ
9. Πνευματικά δικαιώματα
10. Κώδικας Δεοντολογίας (ΤΠΕ)
11. Κοινωνική μέριμνα (ICT4D)

Harris Georgiou (MSc, PhD) – https://github.com/xgeorgio/info

- R&D: Associate post-doc researcher and lecturer with the University Athens (NKUA) and University of Piraeus (UniPi)
- Consultant in Medical Imaging, Machine Learning, Data Analytics, Signal Processing, Process Optimization, Dynamic Systems, Complexity & Emergent A.I., Game Theory
- HRTA member since 2009, LEAR / scientific advisor
- HRTA field operator (USAR, scuba diver)
- Wilderness first aid, paediatric (child/infant)
- Humanitarian aid & disaster relief in Ghana, Lesvos, Piraeus
- Support of unaccomp. minors, teacher in community schools
- Streetwork training, psychological first aid & victim support
- 2+4 books, 170+ scientific papers/articles (and 5 marathons)

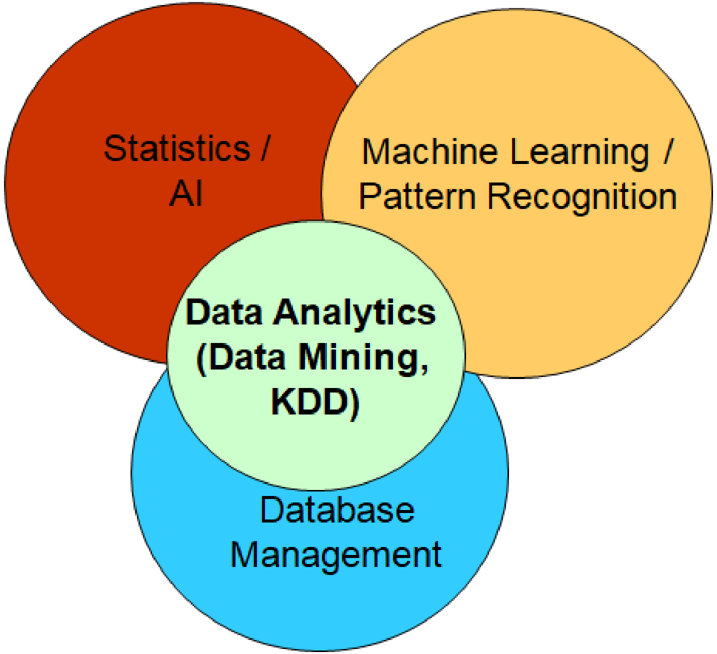
Επισκόπηση – Πηγές

- Περιεχόμενα:
  - Τι είναι η Μηχανική Μάθηση και η Αναλυτική Δεδομένων (ML/DA).
  - Η γλώσσα **R** ως εργαλείο-πλατφόρμα εφαρμογών ML/DA.
  - Βασικό συντακτικό, δυνατότητες, συναρτήσεις, διαγράμματα.
  - Κατηγορίες προβλημάτων ML/DA:
    - Classification, Regression, Clustering, Time Series Analysis.

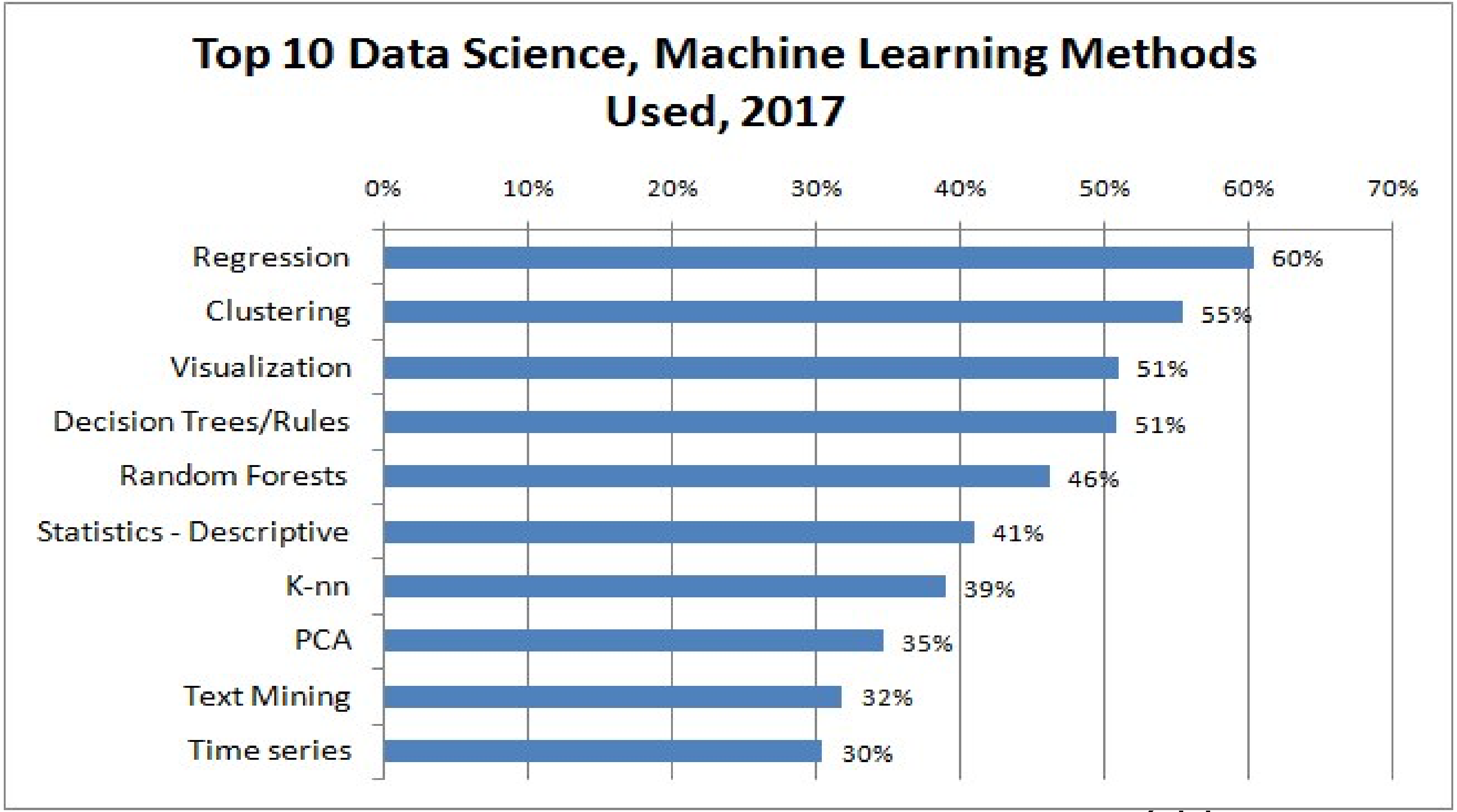
- Πηγές:
  - «Εργαστήριο Αναλυτικής Δεδομένων» – μάθημα ΠΜΣ Πανεπ. Πειραιά (σημειώσεις) 2017-2021.
  - Yanchang Zhao, "R and Data Mining: Examples and Case Studies" (2015) http://www.RDataMining.com
  - Norman Matloff, "The Art of R Programming: A Tour of Statistical Software Design" (1st Edition), No Starch Press, 2011.
  - Rui Miguel Forte, "Mastering Predictive Analytics with R", Packt Publishing, 2015.

Σχετικά επιστημονικά πεδία

- Στατιστική / *«Τεχνητή Νοημοσύνη»*, Μηχανική Μάθηση / Αναγνώριση Προτύπων, Διαχείριση Βάσεων Δεδομένων
- Οι παραδοσιακές τεχνικές επεξεργασίας δεδομένων που μας προσφέρουν αυτές οι επιστημονικές περιοχές μπορεί να είναι ανεφάρμοστες λόγω:
  - του μεγάλου όγκου,
  - των πολλών διαστάσεων,
  - της ετερογένειας των δεδομένων,
  - των απαιτήσεων επεξεργασίας,
  - ...

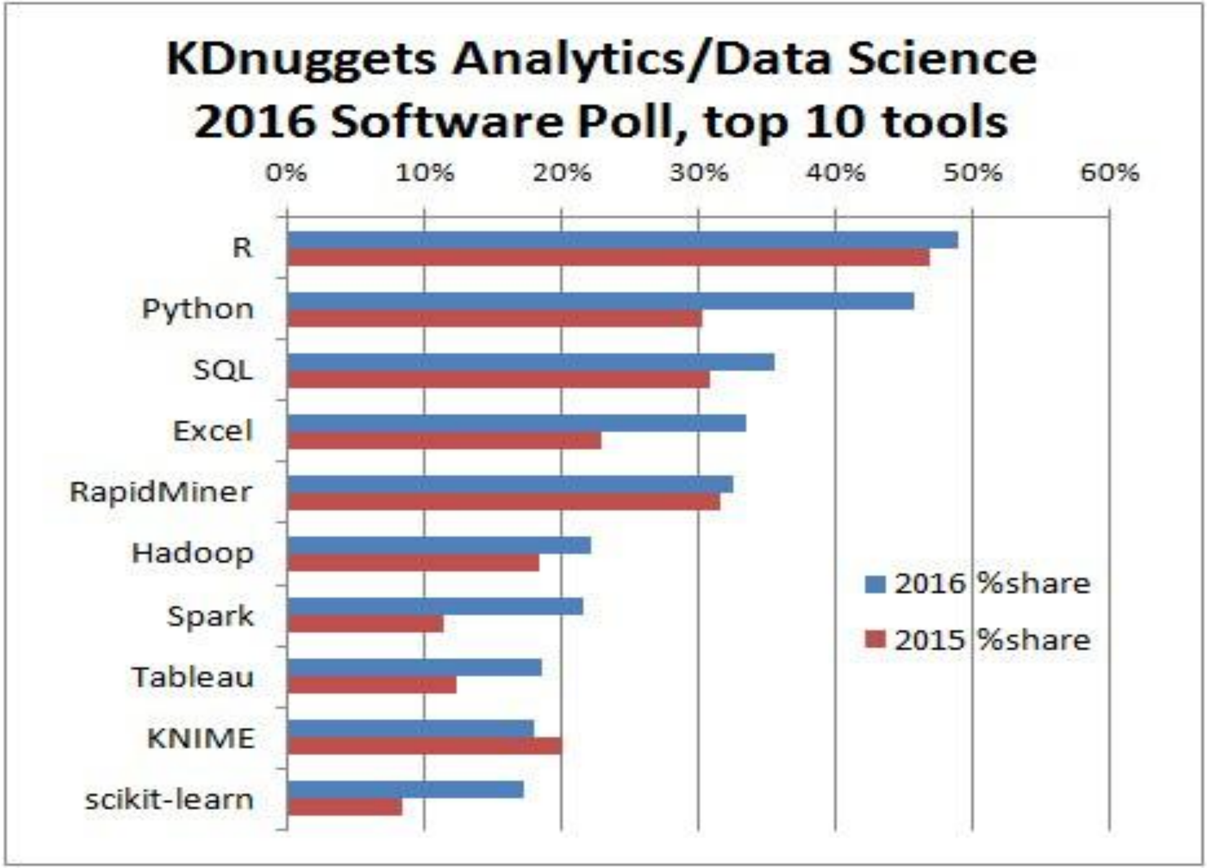


Με ποιες τεχνικές …

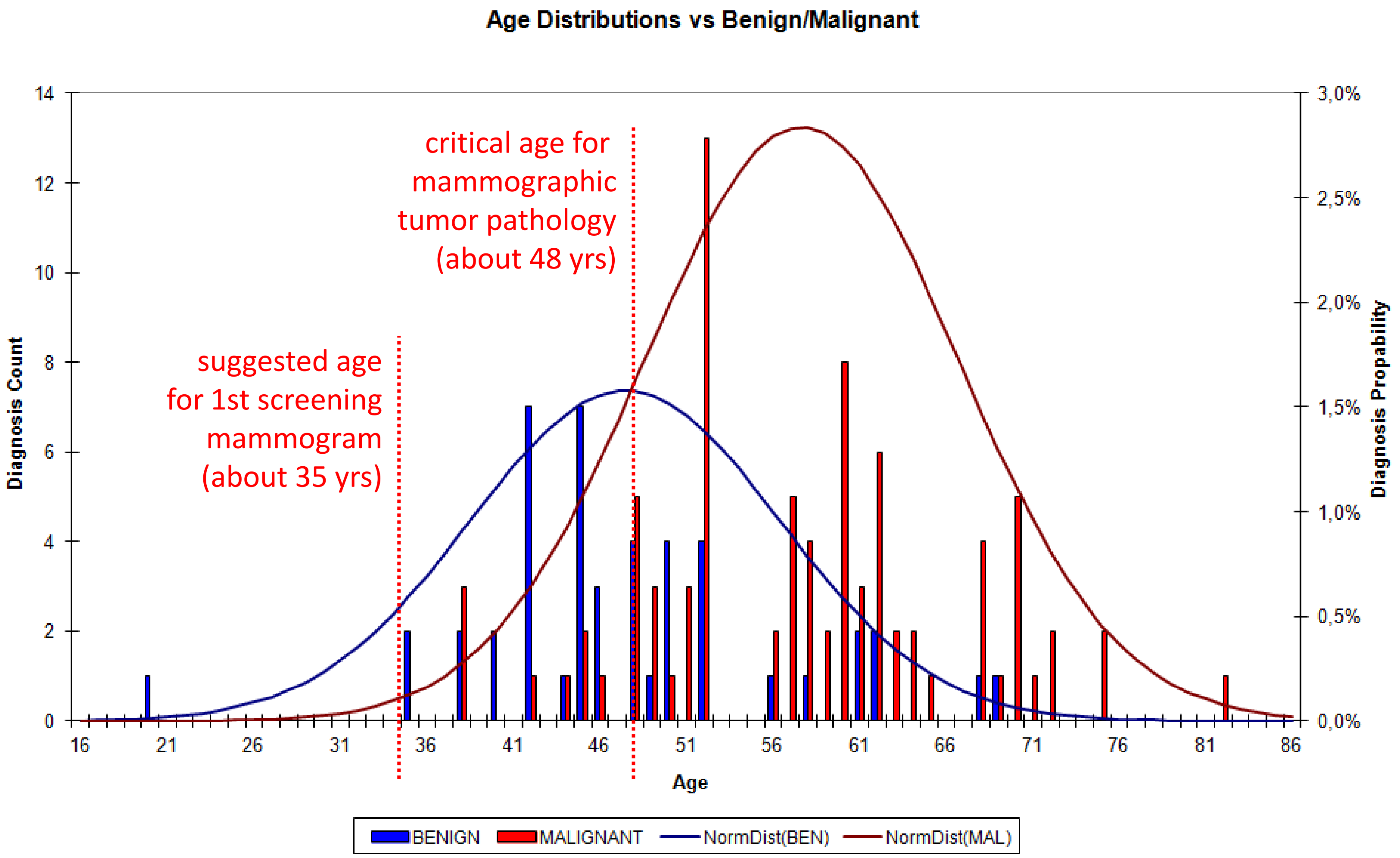


πηγή: kdnuggets.com

Με τι λογισμικό ...



πηγή: kdnuggets.com



Age Distributions vs Benign/Malignant
critical age for mammographic tumor pathology (about 48 yrs)
suggested age for 1st screening mammogram (about 35 yrs)
Diagnosis Count
Diagnosis Propability
Age
14
12
10
8
6
4
2
0
3,0%
2,5%
2,0%
1,5%
1,0%
0,5%
0,0%
16
21
26
31
36
41
46
51
56
61
66
71
76
81
86
BENIGN
MALIGNANT
NormDist(BEN)
NormDist(MAL)

Η γλώσσα R

- **R** can be regarded as a continuation of the “S” programming language, which was developed at Bell Labs (1993) by Rick Becker, John Chambers and Allan Wilks.
- It uses the “matrix manipulation” programming paradigm, i.e., algebraic operations in tabular data, mostly numeric but also supports composite structures.
- “Data frames” are usually such composite data structures, directly reflecting records and DB schemas from real-world applications.
- “Interactive” mode enables online script-based manipulation of data and iterative code prototyping, often without the need to “run” a complete program.
- GUI and visual tools support make data exploration extremely intuitive.
- Extensive support of state-of-the-art algorithms from Linear Algebra, Statistics, Machine Learning, Signal Processing, etc (CRAN repository = 20k official packages).

- Operators
- Datatypes
- Control Structures
    - Control Structures Examples
- Loops
    - Loops Examples
- Data Structures
    - Vectors
    - Matrices
    - Dataframes
- Functions
    - Functions Examples
- Apply Functions
    - Apply
    - Sapply
    - Examples of Both
- **Basic Plotting**
- How to clean the environment
- How to set the directory
- How to read CSV file

Operators

- Comparison Operators
  - == (equal)
  - != (not equal)
  - >= (greater than or equal)
  - <= (less than or equal)
- Logical Operators
  - & (and)
  - | (or)
  - ! (not)

Data types

- R has five basic or “atomic classes”
  - character
  - Numeric (real number)
  - integer
  - Complex
- The most basic object is a vector

Control Structures

An if statement operates on length-one logical vectors

- Syntax

```
if (TRUE) {
        statement_1
}  else    {
        statement_2
}
```

- Example

```
if (1==0) {
        print(1)
}  else    {
        print(2)
}
```

Vectorized ifelse

The ifelse operates on vectors

- Syntax

Ifelse(test, true_value, false_value)

- Example

X<- 1:10

Ifelse(x<5, x, 0)

1 2 3 4 0 0 0 0 0 0

Loops

The basic syntax a for loop in R is

- Syntax

```
For (value in vector) {
statements
}
```

- Example

```
v<-LETTERS[1:4]
for  ( i in v) {
    print(i)
}
```

Data Structures

- Vectors Examples

1. Creating a sequence from 5 to 13 :

```
V<-5:13
print(v)
```

2. Create vector with elements from 5 to 9 incrementing by 0.4:

```
print(seq(5, 9, by=0.4))
```

3. The logical and numeric values are converted to characters:

```
s <- c('apple', 'red', 5, TRUE)
print(s)
```

Data Structures

- **Matrices**

Matrices are the R objects in which the elements are arranged in a two-dimensional rectangular layout

- Syntax

Matrix(data, nrow, ncol, byrow, dimnanes)

- Example

Elements are arranged sequentially by row and the column and row names are defined :

```
rownames = c("row1", "row2", "row3", "row4")

colnames = c("col1", "col2", "col3")

P <- matrix(c(3:14), nrow = 4, byrow = TRUE, dimnames = list(rownames, colnames))
print(P)
```

Data Structures

Matrices: Indexing and Operations

- Access the element at 2nd column and 4th row:

```
print(P[4,2])
```

- Access only the 2nd row:

```
print(P[2,])
```

- Access only the 3rd column:

```
print(P[,3])
```

- Add the matrices:

```
result <- matrix1 + matrix2
```

Data Structures

- Creation of a data frame:

```
emp.data <- data.frame( emp_id = c (1:5),

emp_name = c("Rick","Dan","Michelle","Ryan","Gary"), salary =
  c(623.3,515.2,611.0,729.0,843.25), start_date =
  as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",
  "2015-03-27")), stringsAsFactors = FALSE )
```

- Extraction of Specific columns:

```
result <- data.frame(emp.data$emp_name,emp.data$salary)
```

Functions

An R function is created by using the keyword **function**. The basic syntax of an R function definition is as follows :

```
function_name <- function(arg_1, arg_2, ...) {
 Function body
 return(return_value)
}
```

- Example

Creation of a function to print squares of numbers in sequence and calling this, supplying 6 as an argument:

```
new.function <- function(a) {
for(i in 1:a) { b <- i^2
 print(b) } }
new.function(6)
```

Apply Functions

I. Syntax of apply function

apply(X, MARGIN, FUN, ARGs)

- Arguments
- X: array, matrix or data.frame
- MARGIN: 1 for rows, 2 for columns
- FUN: one or more functions
- ARGs: possible arguments for functions

- Example

```
apply(iris[1:8,1:3], 1, mean)
```

Basic Graphics

1. Scatterplot
2. Boxplot
3. Histogram
4. Quantile-Quantile plot

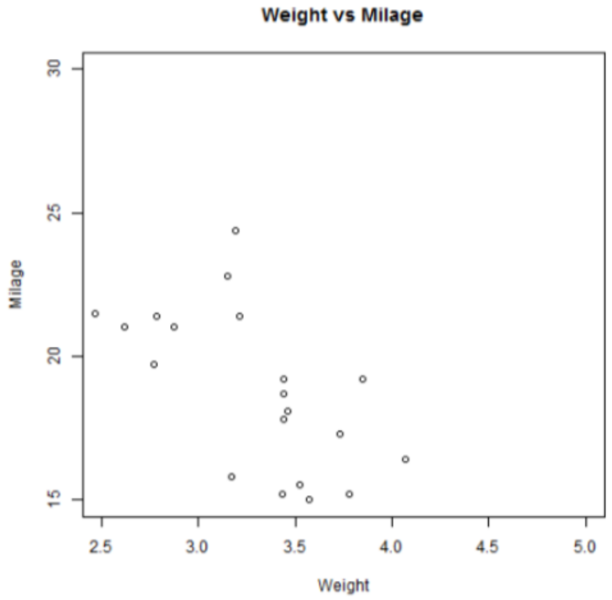
Scatterplot

The simple scatterplot is created using the **plot()** function.

- Syntax

plot(x, y, main, xlab, ylab, xlim, ylim, axes)

- **x** is the data set whose values are the horizontal coordinates.
- **y** is the data set whose values are the vertical coordinates.
- **main** is the tile of the graph.
- **xlab** is the label in the horizontal axis.
- **ylab** is the label in the vertical axis.
- **xlim** is the limits of the values of x used for plotting.
- **ylim** is the limits of the values of y used for plotting.
- **axes** indicates whether both axes should be drawn on the plot.



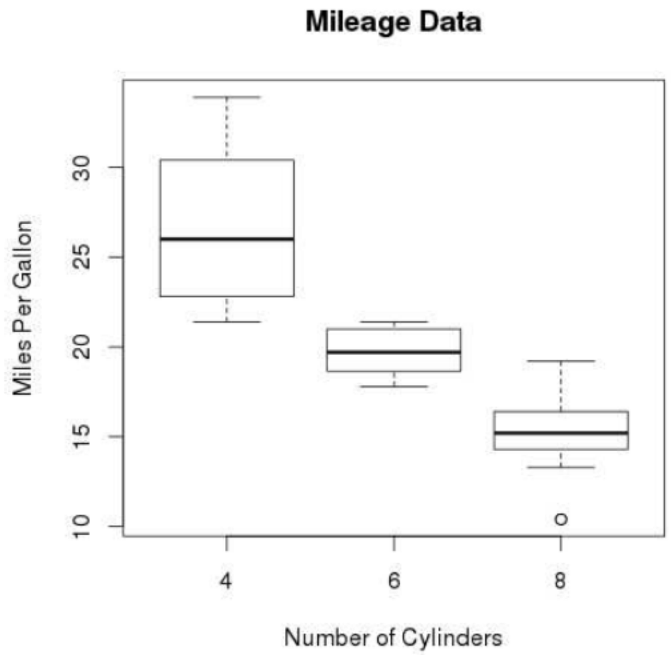
Boxplot

Boxplots are created in R by using the **boxplot()** function.

- Syntax

boxplot(x, data, names, main)

- **x** is a vector or a formula.
- **data** is the data frame.
- **names** are the group labels which will be printed
- **main** is used to give a title to the graph.



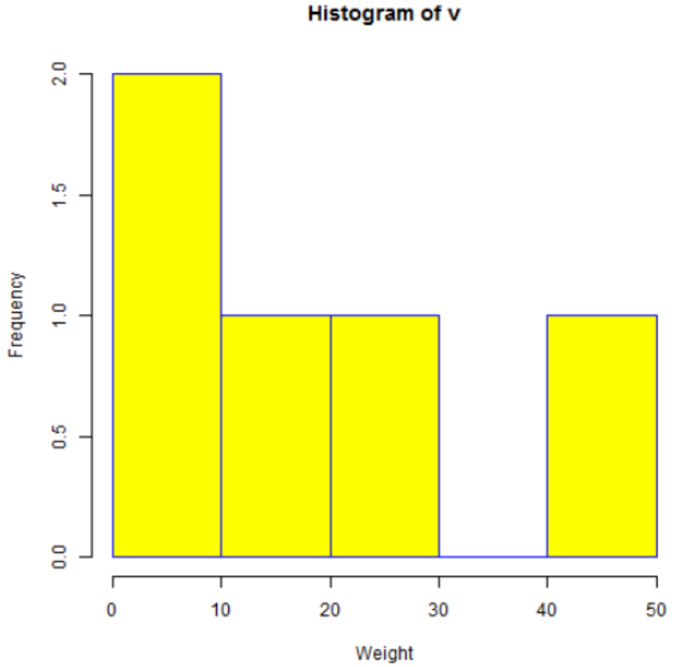
Histogram

R creates histogram using **hist()** function.

- Syntax

hist(v,main,xlab,xlim,ylim,breaks,col,border)

- **v** is a vector containing numeric values used in histogram.
- **main** indicates title of the chart.
- **col** is used to set color of the bars.
- **border** is used to set border color of each bar.
- **xlab** is used to give description of x-axis.
- **xlim** is used to specify the range of values on the x-axis.
- **ylim** is used to specify the range of values on the y-axis.
- **breaks** is used to mention the width of each bar.



How to clean the environment

- clean everything from previous runs

```
closeAllConnections()

rm(list=ls())
```

How to read CSV file

```
math_dataset=read.table("student-mat.csv",sep=";",header=TRUE)
```

Opens the file student-mat.csv

The delimiter between columns is ";“

Header= TRUE means that the first row of the CSV file will be used as a label for each column.

Normalize function

```
#function that does min max normalization
normalize_min_max <- function(element,old_min,old_max,new_min, new_max ){
  v=((element-old_min)/(old_max-old_min))*(new_max-new_min)+new_min
}
```

```
#normalize grades according to min max normalization
new_min = 0
new_max = 100
old_min= min(grades_math)
old_max= max(grades_math)
```

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

Normalize function

```
grades_normalized= apply(grades_math,1, function(x)
   normalize_min_max(x,old_min,old_max,new_min,new_max))
```

```
#insert the normalized grades in the dataset

math_dataset$grades_normalized <- c(grades_normalized)
```

```
#compute mean, median values

mean_value <- mean(grades_math$G3)

median_value <- median(grades_math$G3)
```

minkowski distance

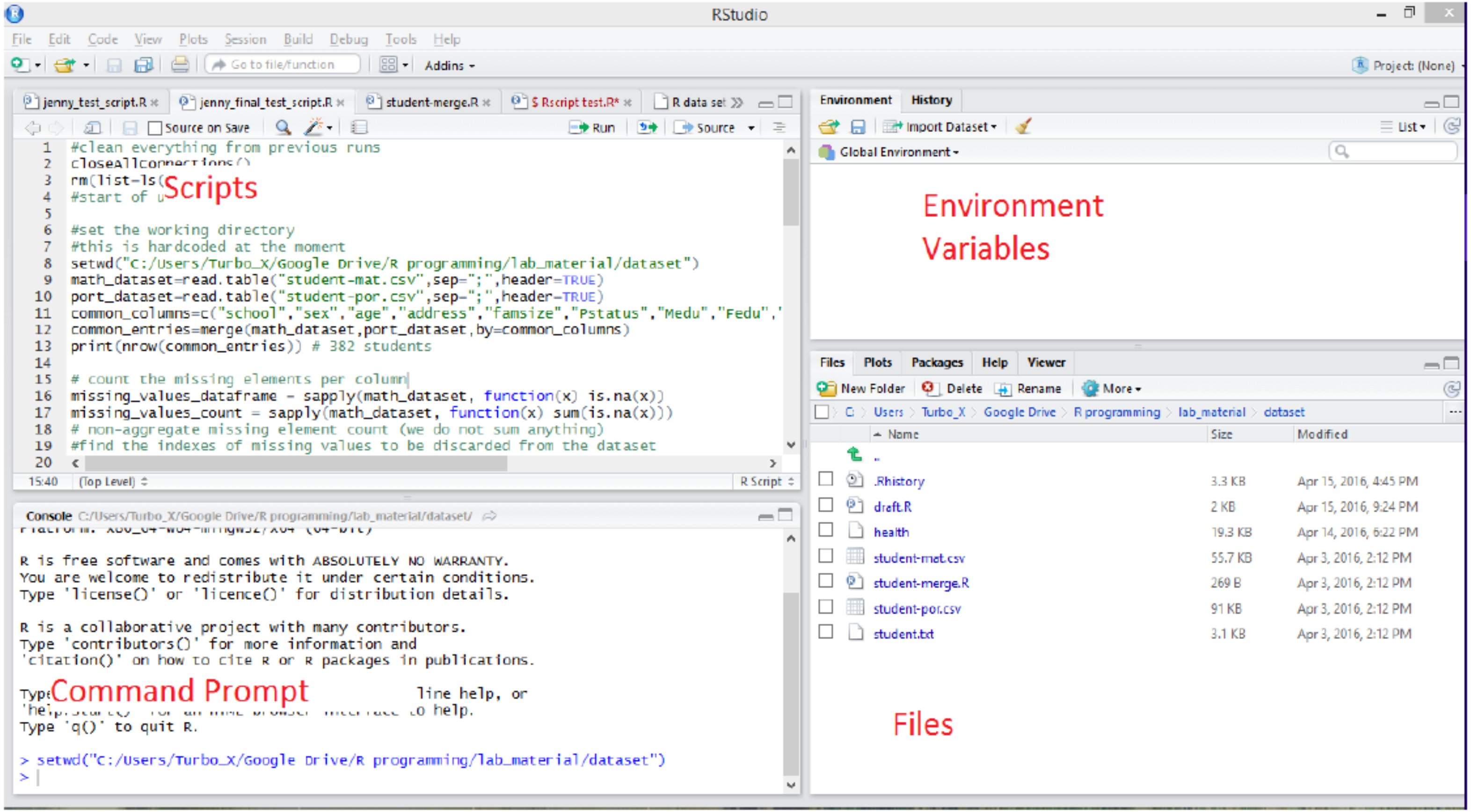
- The **Minkowski distance** is a metric in a normed vector space which can be considered as a generalization of both the Euclidean distance and the Manhattan distance.
- The Minkowski distance of order *p* between two points

is defined as:

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

```
#minkowski distance

stats::dist(students.matrix,method = "minkowski")
```

Getting Started with R studio



Classification:

- Decision trees: *rpart*, *party*
- Random forest: *randomForest*, *party*
- SVM: *e1071*, *kernlab*
- Neural networks: *nnet*, *neuralnet*, *RSNNS*
- Performance evaluation: *ROCR*

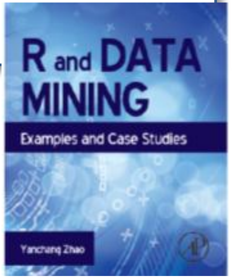
Clustering:

- *k*-means: *kmeans()*, *kmeansruns()*
- *k*-medoids: *pam()*, *pamk()*
- Hierarchical clustering: *hclust()*, *agnes()*, *diana()*
- DBSCAN: *fpc*
- BIRCH: *birch*
- Cluster validation: packages *clv*, *clValid*, *NbClust*

Time series analysis:

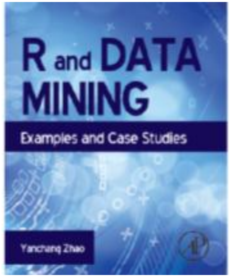
- Time series decomposition: *decomp()*, *decompose()*, *arima()*, *stl()*
- Time series forecasting: *forecast*
- Time Series Clustering: *TSclust*
- Dynamic Time Warping (DTW): *dtw*

- Regression: to predict a continuous value, such as the volume of rain
- Classification: to predict a categorical class label, such as weather: rainy, sunny, cloudy or snowy



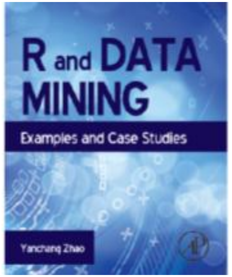
The Iris Dataset

```
# iris data
str(iris)

## 'data.frame': 150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1..
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1..
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0..
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",...

# split into training and test datasets
set.seed(1234)
ind <- sample(2, nrow(iris), replace=T, prob=c(0.7, 0.3))
iris.train <- iris[ind==1, ]
iris.test <- iris[ind==2, ]
```

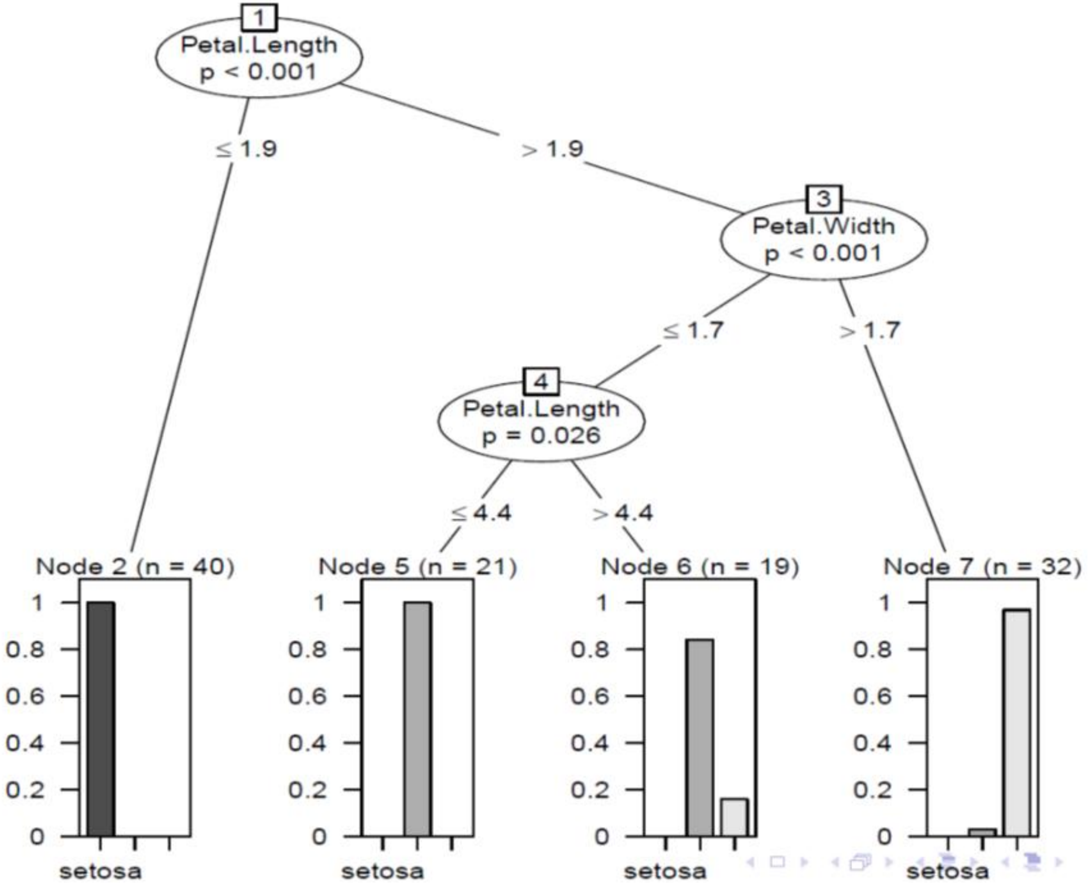


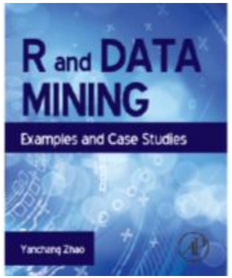
Build a Decision Tree

```r
# build a decision tree
library(party)
iris.formula <- Species ~ Sepal.Length + Sepal.Width +
                          Petal.Length + Petal.Width
iris.ctree <- ctree(iris.formula, data=iris.train)
```



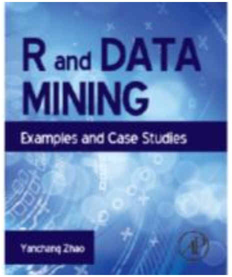
```
plot(iris.ctree)
```





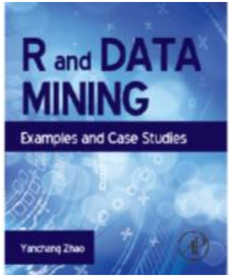
Prediction

```r
# predict on test data
pred <- predict(iris.ctree, newdata = iris.test)
# check prediction result
table(pred, iris.test$Species)

##
## pred         setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0         12         2
##   virginica       0          0        14
```

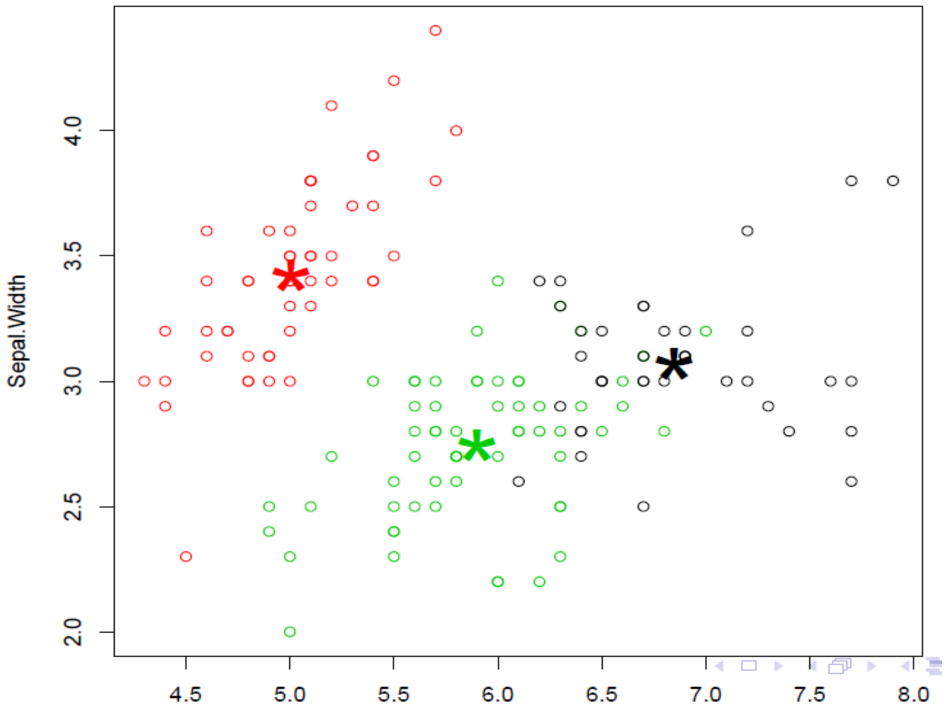


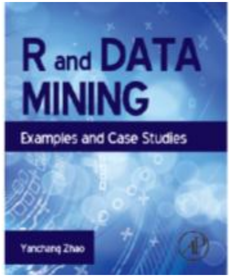
k-means Clustering

```
set.seed(8953)
iris2 <- iris
# remove class IDs
iris2$Species <- NULL
# k-means clustering
iris.kmeans <- kmeans(iris2, 3)
# check result
table(iris$Species, iris.kmeans$cluster)

##
##               1  2  3
##   setosa      0 50  0
##   versicolor  2  0 48
##   virginica  36  0 14
```



```r
# plot clusters and their centers
plot(iris2[c("Sepal.Length", "Sepal.Width")], col=iris.kmeans$cluster)
points(iris.kmeans$centers[, c("Sepal.Length", "Sepal.Width")],
       col=1:3, pch="*", cex=5)
```

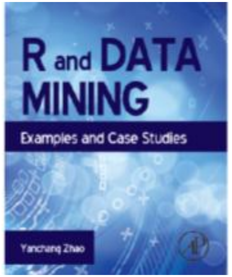




Linear Regression

```
## correlation between CPI and year / quarter
cor(year, cpi)

## [1] 0.9096316

cor(quarter, cpi)

## [1] 0.3738028

## build a linear regression model with function lm()
fit <- lm(cpi ~ year + quarter)
fit

##
## Call:
## lm(formula = cpi ~ year + quarter)
##
## Coefficients:
## (Intercept)         year      quarter
##   -7644.488        3.888        1.167
```

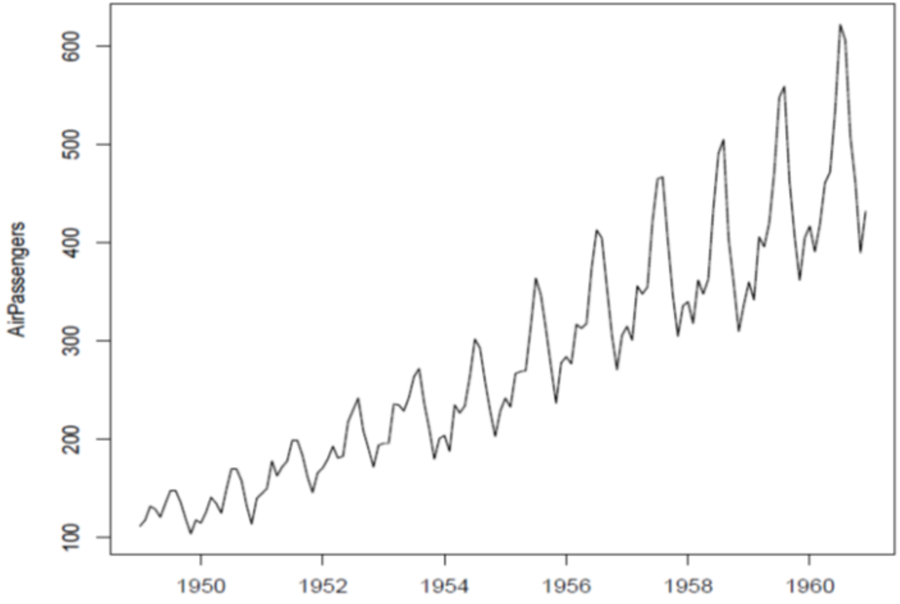
$$cpi = c_0 + c_1 * year + c_2 * quarter,$$

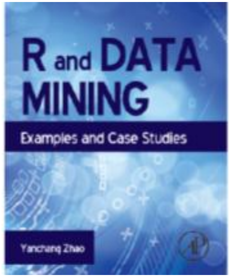


Data AirPassengers

Data `AirPassengers`: monthly totals of Box Jenkins international airline passengers, 1949 to 1960. It has 144(=12×12) values.

```
plot(AirPassengers)
```

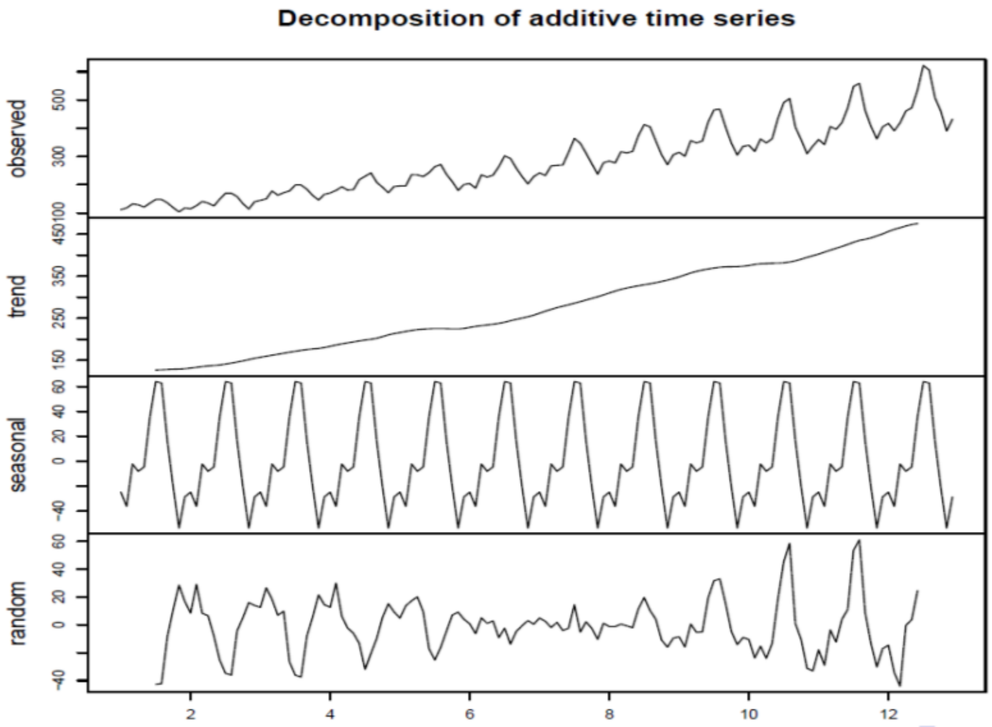


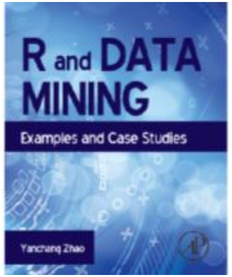


Decomposition

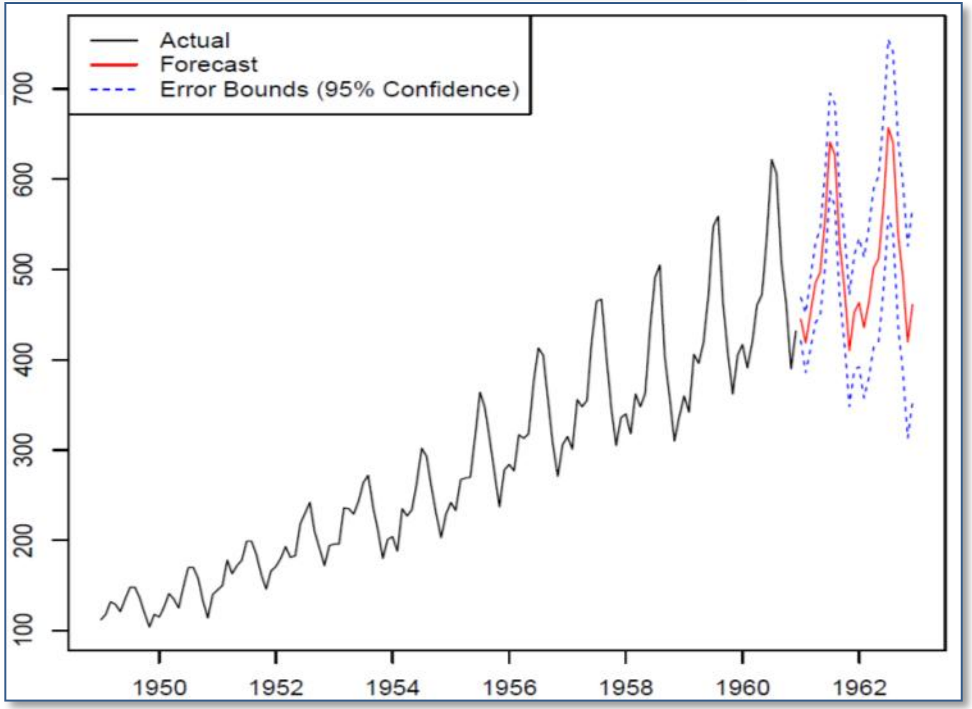
```r
apts <- ts(AirPassengers, frequency = 12)
f <- decompose(apts)
```

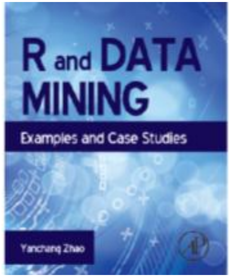
```r
plot(f)
```





```
# build an ARIMA model
fit <- arima(AirPassengers, order = c(1, 0, 0), list(order = c(2,
    1, 0), period = 12))
fore <- predict(fit, n.ahead = 24)
# error bounds at 95% confidence level
U <- fore$pred + 2 * fore$se
L <- fore$pred - 2 * fore$se
```



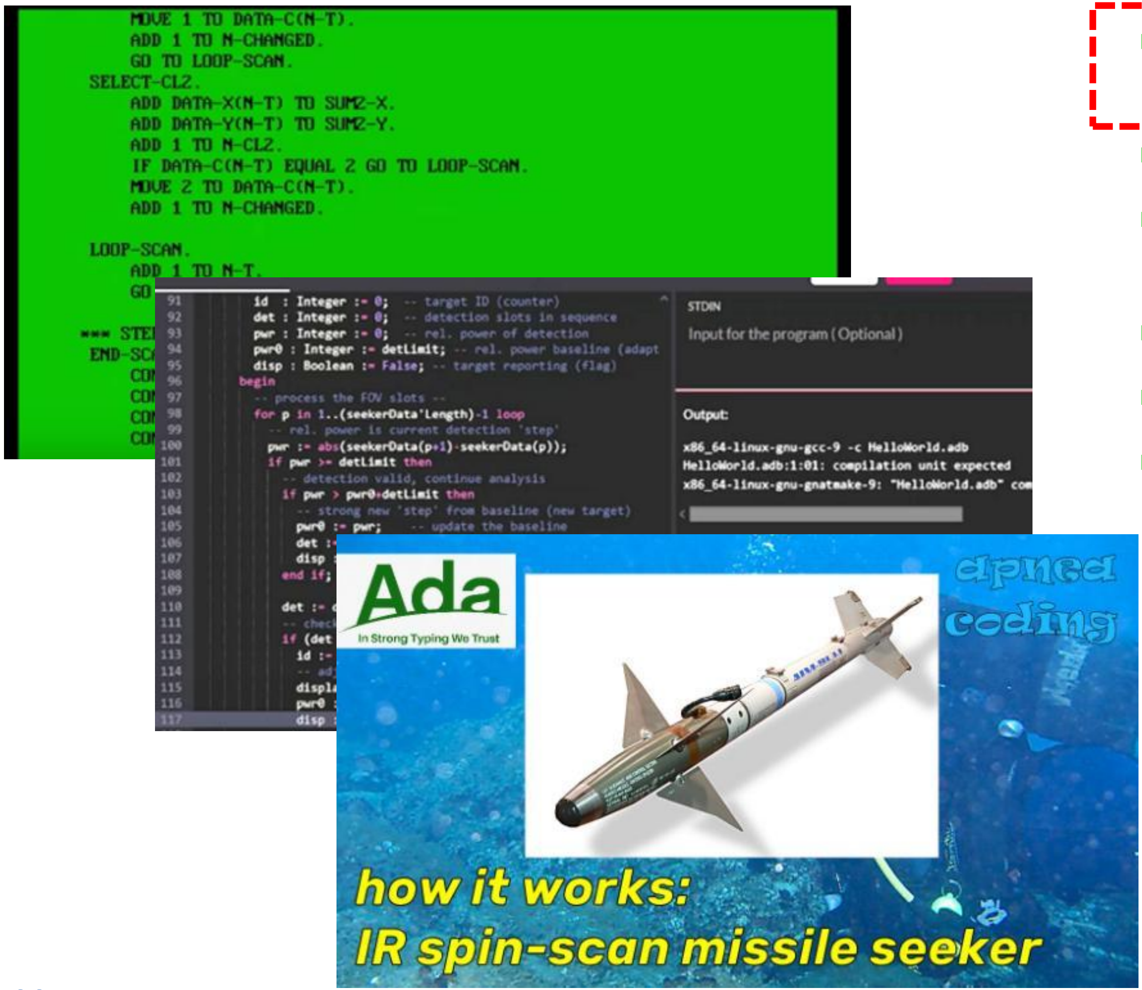


Σύνοψη

- Περιεχόμενα:
  - Τι είναι η Μηχανική Μάθηση και η Αναλυτική Δεδομένων (ML/DA)
  - Η γλώσσα **R** ως εργαλείο-πλατφόρμα εφαρμογών ML/DA
  - Βασικό συντακτικό, δυνατότητες, συναρτήσεις, διαγράμματα
  - Κατηγορίες προβλημάτων ML/DA:
    - Classification, Regression, Clustering, Time Series Analysis

- Πηγές:

  - «Εργαστήριο Αναλυτικής Δεδομένων» – μάθημα ΠΜΣ Πανεπ. Πειραιά (σημειώσεις) 2017-2021.
  - Yanchang Zhao, “R and Data Mining: Examples and Case Studies” (2015) http://www.RDataMining.com
  - Norman Matloff, “The Art of R Programming: A Tour of Statistical Software Design” (1st Edition), No Starch Press, 2011.
  - Rui Miguel Forte, “Mastering Predictive Analytics with R”, Packt Publishing, 2015.



- **Hamming (7,4) error correction codes in R**
- Kmeans clustering in COBOL
- Bi-directional Associative Memory (BAM) in Arduino/C
- Linear Regression in SQL, Matlab
- k-nearest-neighbor Classifier in SQL
- ...

YouTube:

@ApneaCoding

https://www.youtube.com/@apneacoding

Github:

@xgeorgio

https://github.com/xgeorgio

Ερωτήσεις

Χάρης Γεωργίου (MSc,PhD)
https://www.linkedin.com/in/xgeorgio/
https://twitter.com/xgeorgio_gr