# Business Analytics and Big Data: Practice with R

Mehmet Gençer
Assoc.Prof., Organization Studies & Computer Engineering
mehmetgencer@yahoo.com
mehmet.gencer@ieu.edu.tr
https://mgencer.com

# The tool: R

- R is a free software environment for statistical computing and graphics: https://www.r-project.org/
- It provides a wide variety of statistical and graphical techniques.
- Free resources:
  https://cran.r-project.org/doc/manuals/R-intro.pdf

  http://www.rdatamining.com/docs

# R interaction

- When you run R you meet a console.
- R console can understand commands and variables
- A command has a name, and parameters given in paranthesis
  > data()
  > help(data)
- Some variables are already defined:
  > AirPassengers
  Some are defined by you, then used:
  > x <- rnorm(50,mean=10,sd=5)
  > mean(x)

# R: data

- Most common R variable type is “dataframe”, which corresponds to a spreadsheet.
- Accessing specific value of dataframe:

  >women[2,1]
- Accessing columns of dataframe:
  > women$weight
  or
  >women[2]
- Accessing rows:
  >women[2,]
- Or all
  >women[,]

- An R variable (or so called object) can have different data types:
  >class(ChickWeight$Diet)

  [1] "factor"
  > class(ChickWeight$weight)
  [1] "numeric"
- R can do matrix arithmetic too. But here we limit our discussion to data frames only.

# Exploring data - 1

- A first step in analysis is having a good look at data, i.e. exploration, to develop hypothesis about relations.
- > stem(faithful$eruptions)
- > hist(faithful$eruptions,breaks=20,prob=TRUE)
- An R plot/drawing is a living object, which can be manipulated with additions. For example the above plot is further improved with the below commands:
  > lines(density(faithful$eruptions, bw=0.1)) #add density lines
  > rug(faithful$eruptions)  #show actual data points

# Exploring Data - 2

- There are many functions for data exploration:
  > summary(faithful)
- And more for plotting:
  > plot(faithful$eruptions,faithful$waiting)

# R is modular

- R community is a source of numerous packages you may find useful for different analysis.
- Install packages as in the following example:
  > install.packages("scatterplot3d")
- Then load package anytime:
  > library(scatterplot3d)
- Then use the commands made available:
  >
  scatterplot3d(ChickWeight$weight,ChickWeight$Time,ChickWeight$Diet)

# Before analysis: Data loading, cleaning and preperation

- Import data:
  > mydata <-  read.csv(“data.csv”)
- Finding data for practice? e.g.:
  http://www.rdatamining.com/resources/data
  http://www.inside-r.org/howto/finding-data-internet
- The book datasets:
  >install.packages(“TH.data”)
  >bodyfat
- Conversion:
  > myTitanic <- as.data.frame(Titanic)
  > summary(myTitanic)

# Exploring relations in data

- > cor(iris[,1:4])
  > cor(bodyfat$age,bodyfat$hipcirc)
- Or visually:
  > plot(bodyfat$age,bodyfat$anthro3a)
- Explore multiple relations at once:
  > pairs(iris)

# Special purpose exploration libraries

- An example:
  >install.packages("lattice")
  >library(lattice)
  >help(volcano)
  >filled.contour(volcano, color=terrain.colors,
  asp=1,plot.axes=contour(volcano, add=T))
  > persp(volcano, theta=25, phi=30, expand=0.5, col="lightblue")
- Another example:
  >install.packages("MASS")
  > library(MASS)
  > parcoord(iris[1:4], col=iris$Species)
  > parallelplot(~iris[1:4] | Species, data=iris)

# Fundamental models: linear

- Used for building a model as follows, where y is the dependent variable:
  $y = c_0 + c_1 x_1 + c_2 x_2 + \cdots + c_k x_k$
- A simple example:
  > mymodel <- lm(faithful$waiting ~ faithful$eruptions)
  > summary(mymodel)
- Note the parameter relevance and model strength (i.e. $R^2$)
- Bias and variance tradeoff → clean the outliers!
  > boxplot(ChickWeight$weight)

# Linear models ...

- Examining the model fit
  > plot(faithful$eruptions, faithful$waiting)
  > abline(mymodel)
- And examine predictions:
  > predict(faithful)

# Preparing for model building

- In most model building methods we will use,  one needs a training set and a test set while building methods.
- A 30% to 70% separation of data sample is a good rule of thumb for doing this.
- > ind <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))
  > trainData <- iris[ind==1,]
  > testData <- iris[ind==2,]

# Bigdata methods: Decision trees

- Install method library
  > install.packages(“party”)
  > library(party)
- Apply:
  > myFormula <- Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
  > iris_ctree <- ctree(myFormula, data=trainData)
  > iris_ctree
- check the prediction
  > table(predict(iris_ctree), trainData$Species)
  > table(predict(iris_ctree,testData), testData$Species)
- Can you identify the false positives?

# ...continued...

- Alternatively, using the library “rpart”
- Get ready
  >library(rpart)
  >library(TH.data)
  > summary(bodyfat)
  > ind <- sample(2, nrow(bodyfat), replace=TRUE, prob=c(0.7, 0.3))
  > bodyfat.train <- bodyfat[ind==1,]
  > bodyfat.test <- bodyfat[ind==2,]
- Model:
  > myFormula <- DEXfat ~ age + waistcirc + hipcirc + elbowbreadth + kneebreadth
  > bodyfat_rpart <- rpart(myFormula, data = bodyfat.train, control = rpart.control(minsplit = 10))
- Examine
  > bodyfat_rpart
  > plot(bodyfat_rpart)
  > text(bodyfat_rpart, use.n=T)

# Clustering: k-means

- For the example remove the already existing clustering clues:
  > iris2 <- iris
  > iris2$Species <- NULL
  > kmeans.result <- kmeans(iris2, 3)
  > table(iris$Species, kmeans.result$cluster)
  > plot(iris2[c("Sepal.Length", "Sepal.Width")], col = kmeans.result$cluster)
  > # plot cluster centers
  > points(kmeans.result$centers[,c("Sepal.Length", "Sepal.Width")], col = 1:3, pch = 8, cex=2)

# Clustering: hierarchical

- Also an internal method
  ```
  > hc <- hclust(dist(iris), method="ave")
  > plot(hc, labels=iris$Species)
  # cut tree into 3 clusters
  > rect.hclust(hc, k=3)
  > groups <- cutree(hc, k=3)
  ```
-

# Time Series Analysis

- > plot(AirPassengers)
- Decompose the time series into trend and seasonal parts:
  > apts <- ts(AirPassengers, frequency=12)
  > f <- decompose(apts)
  > plot(f)

# Time series forecasting

- Using autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA)
  > 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
  >ts.plot(AirPassengers, fore$pred, U, L, col=c(1,2,4,4), lty = c(1,1,2,2))
  > legend("topleft", c("Actual", "Forecast", "Error Bounds (95% Confidence)"), col=c(1,2,4), lty=c(1,1,2))

# Association rules

- Get the example dataset:
  http://www.cs.toronto.edu/~delve/data/titanic/desc.html
- Load it:
  > getwd() |#check directory
  > titanic.raw <- read.csv("titanic.csv")
- > summary(titanic.raw)
  > library(arules)
  > # find association rules with default settings
  > rules.all <- apriori(titanic.raw)
  >library(arulesViz)
  >plot(rules.all, method="graph")

# More R? - 1

- R is a general computing platform, for conducting research, implementing algorithms.
- The data type of objects match this purpose.
- e.g. vector manipulations:
  > women$weight # a vector from dataframe
  > women$weight *2
  > sd(women$weight *2)
  > log(women$weight *2)
  > women$weight / women$height

# More R - 2

- One can filter data:
  > tallwomen <- women[women$height>65,]
- Create sequences:
  > 1:100
- Explore and use numerous functions for statistical distributions and tests:
  >stem(women$height)
  > shapiro.test(women$height)

# Even more R?

- See r-project.org
- See “An Introduction to R” and other free books at website.
- Check CRAN for numerous contributed packages
- See the R Data Mining website and free book: http://www.rdatamining.com/