

Data Analysis in Software Engineering using R

Daniel Rodriguez and Javier Dolado

2021-10-10

Contents

Welcome

This **Data Analysis in Software Engineering (DASE)** book/notes will try teach you how to do data science with R in Software Engineering.

It is a work in progress.

Acknowledgments

Projects:

- PRESI: TIN2013-46928-C3
 - amuSE TIN2013-46928-C3-2-R
 - PERTEST TIN2013-46928-C3-1-R
- QARE: TIN2016-76956-C3
 - BadgePeople: TIN2016-76956-C3-3-R
 - TESTEAMOS: TIN2016-76956-C3-1-R
- Network SBSE (SEBASNet): TIN2015-71841-REDT
- TestBUS PID2019-105455GB-C32

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 United States License.

Part I

Introduction to the R Language

Chapter 1

Introduction to R

The goal of the first part of this book is to get you up to speed with the basics of **R** as quickly as possible.

1.1 Installation

Install the latest preview version for getting all features.

Follow the procedures according to your operating system.

- Linux: You need to have `blas` and `gfortran` installed on your Linux, for installing the `coin` package.
- `Rgraphviz` requires installation from `source("http://bioconductor.org/biocLite.R")`, then `biocLite("Rgraphviz")`.
- Uncomment the following lines for installing all missing packages (this will take some time):

```
# listofpackages <- c("arules", "arulesViz", "bookdown", "ggplot2", "vioplot", "UsingR", "fpc", "ri")
# newpackages <- listofpackages[!(listofpackages %in% installed.packages()[, "Package"])]
# if(length(newpackages)>0) install.packages(newpackages, dependencies = TRUE)
#
# # install from archive
# if (!is.element("rgp", installed.packages()[, 1]))
# { install.packages("https://cran.r-project.org/src/contrib/Archive/rgp/rgp_0.4-1.tar.gz",
#                     repos = NULL)
# }
## end of installing packages

# in Linux you may need to run several commands (in the terminal) and install additional libraries
# sudo R CMD javareconf
# sudo apt-get install build-essential
```

```
# sudo apt-get install libxml2-dev
# sudo apt-get install libpq
# sudo apt-get install libpq-dev
# sudo apt-get install -y libmariadb-client-lgpl-dev
# sudo apt-get install texlive-xetex
# sudo apt-get install r-cran-rmysql
```

1.2 R and RStudio

- R is a programming language for statistical computing and data analysis that supports a variety of programming styles. See R in Wikipedia
- R has multiple online resources and books.
- R coding style
- R-Bloggers
- Getting help in R
 - RStudio cheat sheet
 - Base R cheat sheet
 - Advanced R cheat sheet
 - Data Visualization cheat sheet
 - R Markdown cheatsheet
 - [R Markdown Basics] (http://rmarkdown.rstudio.com/authoring_basics.html)
 - Python with R and Reticulate Cheatsheet
 - caret
 - All cheatsheets and translations
 - `help(" ")` command
- R as a calculator. Console: It uses the command-line interface.
- This document is an RMarkdown document. See bookdown.org

Examples:

```
x <- c(1,2,3,4,5,6)      # Create ordered collection (vector)
y <- x^2                  # Square the elements of x
print(y)                  # print (vector) y

## [1]  1  4  9 16 25 36
mean(y)                   # Calculate average (arithmetic mean) of (vector) y; result is s

## [1] 15.16667
var(y)                    # Calculate sample variance
```

```

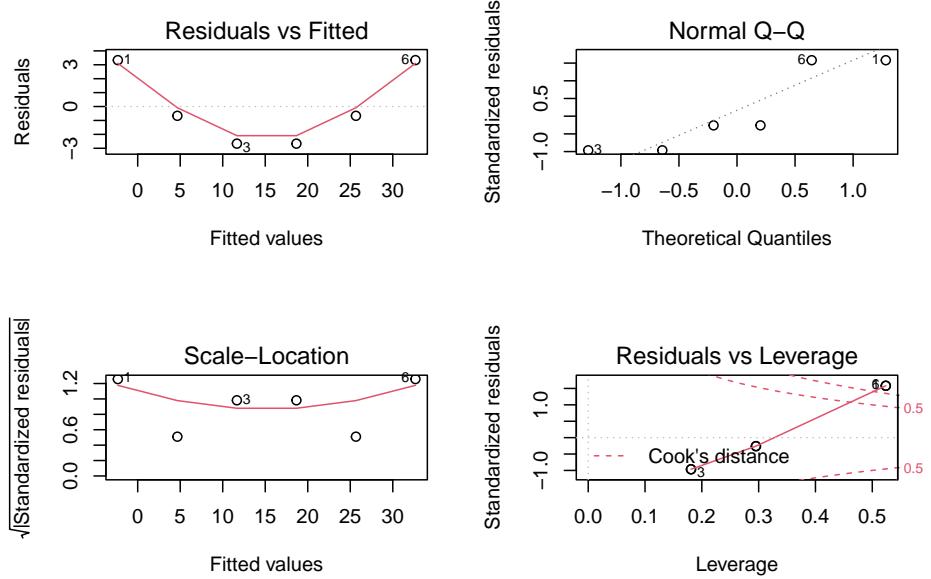
## [1] 178.9667
lm_1 <- lm(y ~ x)      # Fit a linear regression model "y = f(x)" or "y = B0 + (B1 * x)"
# store the results as lm_1
print(lm_1)            # Print the model from the (linear model object) lm_1

## 
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)          x
## -9.333        7.000
summary(lm_1)           # Compute and print statistics for the fit

## 
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    1     2     3     4     5     6 
## 3.3333 -0.6667 -2.6667 -2.6667 -0.6667 3.3333 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -9.3333    2.8441  -3.282 0.030453 *  
## x            7.0000    0.7303   9.585 0.000662 *** 
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.055 on 4 degrees of freedom
## Multiple R-squared:  0.9583, Adjusted R-squared:  0.9478 
## F-statistic: 91.88 on 1 and 4 DF,  p-value: 0.000662

# of the (linear model object) lm_1
par(mfrow=c(2, 2))      # Request 2x2 plot layout
plot(lm_1)               # Diagnostic plot of regression model

```



```

help(lm)
?lm
apropos("lm")

## [1] ".colMeans"      ".lm.fit"        "colMeans"       "confint.lm"
## [5] "contr.helmert"  "dummy.coef.lm"  "glm"           "glm.control"
## [9] "glm.fit"         "KalmanForecast" "KalmanLike"     "KalmanRun"
## [13] "KalmanSmooth"   "kappa.lm"        "lm"            "lm_1"
## [17] "lm.fit"          "lm.influence"   "lm.wfit"       "model.matrix.lm"
## [21] "nlm"             "nlminb"         "predict.glm"   "predict.lm"
## [25] "residuals.glm"  "residuals.lm"   "summary.glm"   "summary.lm"

example(lm)

##
## lm> require(graphics)
##
## lm> ## Annette Dobson (1990) "An Introduction to Generalized Linear Models".
## lm> ## Page 9: Plant Weight Data.
## lm> ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
##
## lm> trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
##
## lm> group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
##
## lm> weight <- c(ctl, trt)
##
## lm> lm.D9 <- lm(weight ~ group)

```

```

##  

## lm> lm.D90 <- lm(weight ~ group - 1) # omitting intercept  

##  

## lm> ## No test:  

## lm> ##D anova(lm.D9)  

## lm> ##D summary(lm.D90)  

## lm> ## End(No test)  

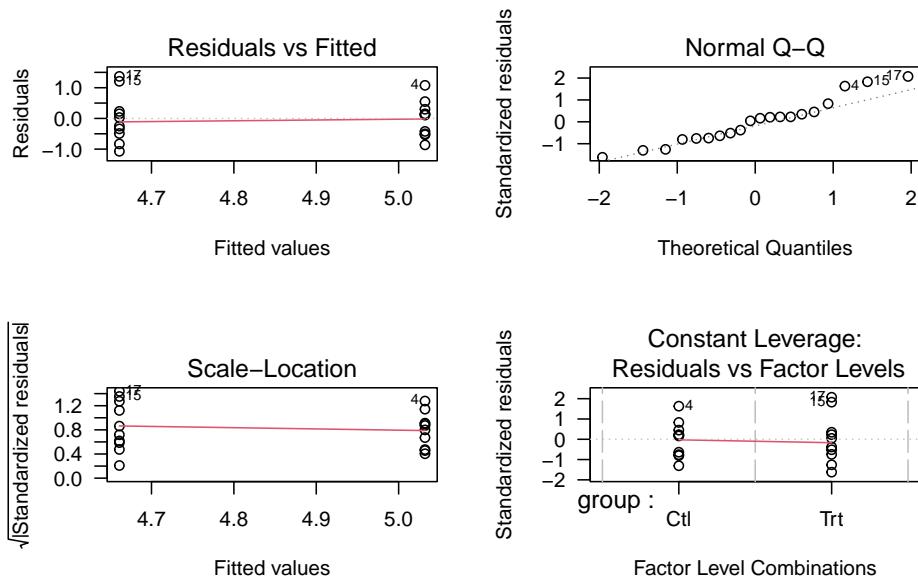
## lm> opar <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))  

##  

## lm> plot(lm.D9, las = 1)      # Residuals, Fitted, ...

```

lm(weight ~ group)



```

##  

## lm> par(opar)
##  

## lm> ## Don't show:  

## lm> ## model frame :
## lm> stopifnot(identical(lm(weight ~ group, method = "model.frame"),
## lm+                         model.frame(lm.D9)))
##  

## lm> ## End(Don't show)
## lm> ### less simple examples in "See Also" above
## lm>
## lm>
## lm>

```

- R script. # A file with R commands # comments `source("filewithcommands.R")`

- ```
sink("recordmycommands.lis") savehistory()
```
- From command line:
    - Rscript
    - Rscript file with `-e` (e.g. `Rscript -e 2+2`)
    - To exit R: `quit()`
  - Variables. R is case sensitive
 

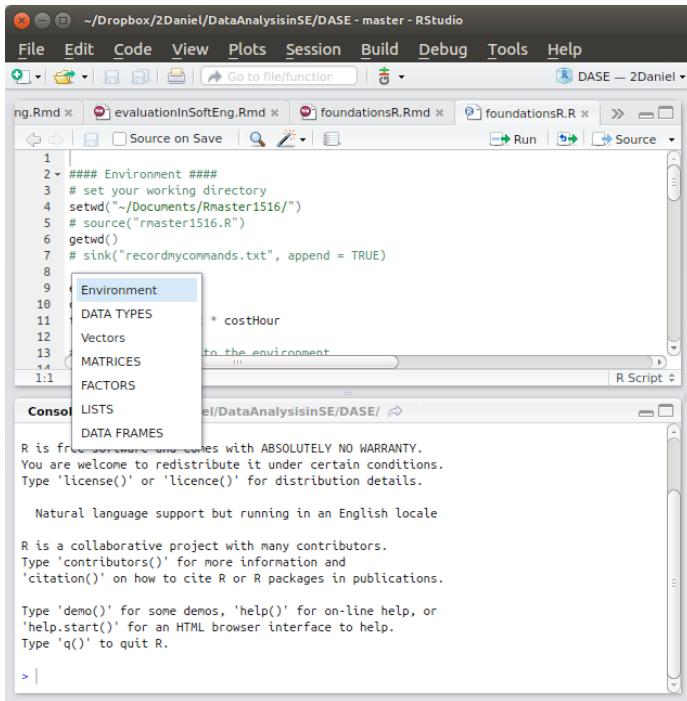
```
var1 <- 1:10
vAr1 <- 11:20
var1
```

```
[1] 1 2 3 4 5 6 7 8 9 10
vAr1
```

```
[1] 11 12 13 14 15 16 17 18 19 20
```
  - Operators
    - assign operator `<-`
    - sequence operator, for example: `mynums <- 0:20`
    - arithmetic operators: `+ - = / ^ %/% %%` (integer division) `%%` (modulus operator)
  - The workspace. Objects.
    - `ls()` `objects()` `ls.str()` lists and describes the objects
    - `rm(x)` delete a variable. E.g., `rm(totalCost)`
    - `s.str()`
    - `objects()`
    - `str()` The structure function provides information about the variable
  - RStudio, RCommander and RKWard are the well-known IDEs for R (more later).

---

  - Four `# ('#####')` create an *environment* in RStudio. An environment binds a set of names to a set of values. You can think of an environment as a bag of names.
    - Environment basics



## Working directories:

```
set your working directory
setwd("~/workingDir/")
getwd()

[1] "/home/drg/Projects/DASE"
record R commands:
sink("recordmycommands.txt", append = TRUE)
```

## 1.3 Basic Data Types

- class( )
  - logical: TRUE, FALSE
  - numeric, integer:
    - is.numeric( )
    - is.integer( )
  - character

Examples:

```
TRUE
```

```
[1] TRUE
class(TRUE)

[1] "logical"
FALSE

[1] FALSE
NA # missing
```

```
[1] NA
class(NA)

[1] "logical"
T

[1] TRUE
F

[1] FALSE
```

```
NaN
```

```
[1] NaN
class(NaN)

[1] "numeric"
numeric data type
2
```

```
[1] 2
class(2)
```

```
[1] "numeric"
2.5
```

```
[1] 2.5
2L # integer
```

```
[1] 2
class(2L)
```

```
[1] "integer"
```

```
is.numeric(2)
```

```
[1] TRUE
```

```
is.numeric(2L)
```

```
[1] TRUE
```

```
is.integer(2)
```

```
[1] FALSE
```

```
is.integer(2L)
```

```
[1] TRUE
```

```
is.numeric(NaN)
```

```
[1] TRUE
```

- data type coercion:

- `as.numeric()`
- `as.character()`

- `as.integer()`

Examples:

```
truenum <- as.numeric(TRUE)
```

```
truenum
```

```
[1] 1
```

```
class(truenum)
```

```
[1] "numeric"
```

```
falsenum <- as.numeric(FALSE)
```

```
falsenum
```

```
[1] 0
```

```
num2char <- as.character(55)
```

```
num2char
```

```
[1] "55"
```

```
char2num <- as.numeric("55.3")
```

```
char2int <- as.integer("55.3")
```

### 1.3.1 Mising values

- NA stands for Not Available, which is not a number as well. It applies to missing values.
- NaN means ‘Not a Number’

Examples:

```
NA + 1
[1] NA
mean(c(5,NA,7))

[1] NA
mean(c(5,NA,7), na.rm=TRUE) # some functions allow to remove NAs
[1] 6
```

---

## 1.4 Vectors

Examples:

```
phases <- c("reqs", "dev", "test1", "test2", "maint")
str(phases)

chr [1:5] "reqs" "dev" "test1" "test2" "maint"
is.vector(phases)

[1] TRUE
thevalues <- c(15, 60, 30, 35, 22)
names(thevalues) <- phases
str(thevalues)

Named num [1:5] 15 60 30 35 22
- attr(*, "names")= chr [1:5] "reqs" "dev" "test1" "test2" ...
thevalues

reqs dev test1 test2 maint
15 60 30 35 22
```

A single value is a vector! Example:

```
aphase <- 44
is.vector(aphase)

[1] TRUE
```

```
length(aphase)
[1] 1
length(thevalues)
[1] 5
```

### 1.4.1 Coercion for vectors

```
thevalues1 <- c(15, 60, "30", 35, 22)
class(thevalues1)
[1] "character"
thevalues1
[1] "15" "60" "30" "35" "22"
<- is equivalent to assign ()
assign("costs", c(50, 100, 30))
```

### 1.4.2 Vector arithmetic

The operation is carried out in all the elements of the vector. For example:

```
assign("costs", c(50, 100, 30))
costs/3
[1] 16.66667 33.33333 10.00000
costs - 5
[1] 45 95 25
costs <- costs - 5

incomes <- c(200, 800, 10)
earnings <- incomes - costs
sum(earnings)

[1] 845
R recycles values in vectors!
vector1 <- c(1,2,3)
vector2 <- c(10,11,12,13,14,15,16)
vector1 + vector2

Warning in vector1 + vector2: longer object length is not a multiple of shorter
object length
```

```

[1] 11 13 15 14 16 18 17

Subsetting vectors
Subsetting vectors []
phase1 <- phases[1]
phase1

[1] "reqs"
phase3 <- phases[3]
phase3

[1] "test1"
thevalues[phase1]

reqs
15
thevalues["reqs"]

reqs
15
testphases <- phases[c(3,4)]
thevalues[testphases]

test1 test2
30 35
Negative indexes

phases1 <- phases[-5]
phases1

[1] "reqs" "dev" "test1" "test2" "maint"
phases1

[1] "reqs" "dev" "test1" "test2"
#phases2 <- phases[-testphases] ## error in argument
phases2 <- phases[-c(3,4)]
phases2

[1] "reqs" "dev" "maint"
subset using logical vector

phases3 <- phases[c(FALSE, TRUE, TRUE, FALSE)] #recycled first value
phases3

```

```

[1] "dev" "test1"
selectionv <- c(FALSE, TRUE, TRUE, FALSE)
phases3 <- phases[selectionv]
phases3

[1] "dev" "test1"
selectionvec2 <- c(TRUE, FALSE)

thevalues2 <- thevalues[selectionvec2]
thevalues2

reqs test1 maint
15 30 22
Generating regular sequences with `:` and `seq`
aseqofvalues <- 1:20

aseqofvalues2 <- seq(from=-3, to=3, by=0.5)
aseqofvalues2

[1] -3.0 -2.5 -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0 2.5 3.0
aseqofvalues3 <- seq(0, 100, by=10)
aseqofvalues4 <- aseqofvalues3[c(2, 4, 6, 8)]
aseqofvalues4

[1] 10 30 50 70
aseqofvalues4 <- aseqofvalues3[-c(2, 4, 6, 8)]
aseqofvalues4

[1] 0 20 40 60 80 90 100
aseqofvalues3[c(1,2)] <- c(666,888)
aseqofvalues3

[1] 666 888 20 30 40 50 60 70 80 90 100
Logical values in vectors TRUE/FALSE

aseqofvalues3 > 50

[1] TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE
aseqofvalues5 <- aseqofvalues3[aseqofvalues3 > 50]
aseqofvalues5

```

```

[1] 666 888 60 70 80 90 100
aseqofvalues6 <- aseqofvalues3[!(aseqofvalues3 > 50)]
aseqofvalues6

[1] 20 30 40 50
Comparison functions

aseqofvalues7 <- aseqofvalues3[aseqofvalues3 == 50]
aseqofvalues7

[1] 50
aseqofvalues8 <- aseqofvalues3[aseqofvalues3 == 22]
aseqofvalues8

numeric(0)
aseqofvalues9 <- aseqofvalues3[aseqofvalues3 != 50]
aseqofvalues9

[1] 666 888 20 30 40 60 70 80 90 100
logicalcond <- aseqofvalues3 >= 50
aseqofvalues10 <- aseqofvalues3[logicalcond]
aseqofvalues10

[1] 666 888 50 60 70 80 90 100
Remove Missing Values (NAs)

aseqofvalues3[c(1,2)] <- c(NA,NA)
aseqofvalues3

[1] NA NA 20 30 40 50 60 70 80 90 100
aseqofvalues3 <- aseqofvalues3[!is.na(aseqofvalues3)]
aseqofvalues3

[1] 20 30 40 50 60 70 80 90 100

```

---

## 1.5 Arrays and Matrices

Matrices are actually long vectors.

```

mymat <- matrix(1:12, nrow =2)
mymat

```

```

[,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1 3 5 7 9 11
[2,] 2 4 6 8 10 12
mymat <- matrix(1:12, ncol =3)
mymat

[,1] [,2] [,3]
[1,] 1 5 9
[2,] 2 6 10
[3,] 3 7 11
[4,] 4 8 12
mymat <- matrix(1:12, nrow=2, byrow = TRUE)
mymat

[,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1 2 3 4 5 6
[2,] 7 8 9 10 11 12
mymat <- matrix(1:12, nrow=3, ncol=4)
mymat

[,1] [,2] [,3] [,4]
[1,] 1 4 7 10
[2,] 2 5 8 11
[3,] 3 6 9 12
mymat <- matrix(1:12, nrow=3, ncol=4, byrow=TRUE)
mymat

[,1] [,2] [,3] [,4]
[1,] 1 2 3 4
[2,] 5 6 7 8
[3,] 9 10 11 12
recycling
mymat <- matrix(1:5, nrow=3, ncol=4, byrow=TRUE)

Warning in matrix(1:5, nrow = 3, ncol = 4, byrow = TRUE): data length [5] is not
a sub-multiple or multiple of the number of rows [3]
mymat

[,1] [,2] [,3] [,4]
[1,] 1 2 3 4
[2,] 5 1 2 3
[3,] 4 5 1 2

```

```

rbind cbind

cbind(1:3, 1:3)

[,1] [,2]
[1,] 1 1
[2,] 2 2
[3,] 3 3

rbind(1:3, 1:3)

[,1] [,2] [,3]
[1,] 1 2 3
[2,] 1 2 3

mymat <- matrix(1)

mymat <- matrix(1:8, nrow=2, ncol=4, byrow=TRUE)
mymat

[,1] [,2] [,3] [,4]
[1,] 1 2 3 4
[2,] 5 6 7 8

rbind(mymat, 9:12)

[,1] [,2] [,3] [,4]
[1,] 1 2 3 4
[2,] 5 6 7 8
[3,] 9 10 11 12

mymat <- cbind(mymat, c(5,9))
mymat

[,1] [,2] [,3] [,4] [,5]
[1,] 1 2 3 4 5
[2,] 5 6 7 8 9

mymat <- matrix(1:8, byrow = TRUE, nrow=2)
mymat

[,1] [,2] [,3] [,4]
[1,] 1 2 3 4
[2,] 5 6 7 8

rownames(mymat) <- c("row1", "row2")
mymat

[,1] [,2] [,3] [,4]
row1 1 2 3 4

```

```

row2 5 6 7 8
colnames(mymat) <- c("col1", "col2", "col3", "col4")
mymat

col1 col2 col3 col4
row1 1 2 3 4
row2 5 6 7 8
mymat2 <- matrix(1:12, byrow=TRUE, nrow=3, dimnames=list(c("row1", "row2", "row3"),
 c("col1", "col2", "col3", "col4")))
mymat2

col1 col2 col3 col4
row1 1 2 3 4
row2 5 6 7 8
row3 9 10 11 12

Coercion in Arrays

matnum <- matrix(1:8, ncol = 2)
matnum

[,1] [,2]
[1,] 1 5
[2,] 2 6
[3,] 3 7
[4,] 4 8

matchar <- matrix(LETTERS[1:6], nrow = 4, ncol = 3)
matchar

[,1] [,2] [,3]
[1,] "A" "E" "C"
[2,] "B" "F" "D"
[3,] "C" "A" "E"
[4,] "D" "B" "F"

matchars <- cbind(matnum, matchar)
matchars

[,1] [,2] [,3] [,4] [,5]
[1,] "1" "5" "A" "E" "C"
[2,] "2" "6" "B" "F" "D"
[3,] "3" "7" "C" "A" "E"
[4,] "4" "8" "D" "B" "F"

Subsetting

mymat3 <- matrix(sample(-8:15, 12), nrow=3) #sample 12 numbers between -8 and 15

```

```
mymat3

[,1] [,2] [,3] [,4]
[1,] 5 13 -5 15
[2,] 3 -8 12 0
[3,] -3 8 4 -1
mymat3[2,3]

[1] 12
mymat3[1,4]

[1] 15
mymat3[3,]

[1] -3 8 4 -1
mymat3[,4]

[1] 15 0 -1
mymat3[5] # counts elements by column

[1] -8
mymat3[9]

[1] 4
Subsetting multiple elements

mymat3[2, c(1,3)]

[1] 3 12
mymat3[c(2,3), c(1,3,4)]

[,1] [,2] [,3]
[1,] 3 12 0
[2,] -3 4 -1
rownames(mymat3) <- c("r1", "r2", "r3")
colnames(mymat3) <- c("c1", "c2", "c3", "c4")
mymat3["r2", c("c1", "c3")]

c1 c3
3 12
Subset by logical vector
mymat3[c(FALSE, TRUE, FALSE),
```

```
c(TRUE, FALSE, TRUE, FALSE)]

c1 c3
3 12
mymat3[c(FALSE, TRUE, TRUE),
 c(TRUE, FALSE, TRUE, TRUE)]

c1 c3 c4
r2 3 12 0
r3 -3 4 -1
matrix arithmetic

row1 <- c(220, 137)
row2 <- c(345, 987)
row3 <- c(111, 777)

mymat4 <- rbind(row1, row2, row3)
rownames(mymat4) <- c("row_1", "row_2", "row_3")
colnames(mymat4) <- c("col_1", "col_2")
mymat4

col_1 col_2
row_1 220 137
row_2 345 987
row_3 111 777
mymat4/10

col_1 col_2
row_1 22.0 13.7
row_2 34.5 98.7
row_3 11.1 77.7
mymat4 -100

col_1 col_2
row_1 120 37
row_2 245 887
row_3 11 677
mymat5 <- rbind(c(50,50), c(10,10), c(100,100))
mymat5

[,1] [,2]
[1,] 50 50
[2,] 10 10
[3,] 100 100
```

```

mymat4 - mymat5

col_1 col_2
row_1 170 87
row_2 335 977
row_3 11 677
mymat4 * (mymat5/100)

col_1 col_2
row_1 110.0 68.5
row_2 34.5 98.7
row_3 111.0 777.0
index matrices

m1 <- array(1:20, dim=c(4,5))
m1

[,1] [,2] [,3] [,4] [,5]
[1,] 1 5 9 13 17
[2,] 2 6 10 14 18
[3,] 3 7 11 15 19
[4,] 4 8 12 16 20
index <- array(c(1:3, 3:1), dim=c(3,2))
index

[,1] [,2]
[1,] 1 3
[2,] 2 2
[3,] 3 1
#use the "index matrix" as the index for the other matrix
m1[index] <-0
m1

[,1] [,2] [,3] [,4] [,5]
[1,] 1 5 0 13 17
[2,] 2 0 10 14 18
[3,] 0 7 11 15 19
[4,] 4 8 12 16 20

```

---

## 1.6 Factors

- Factors are variables in R which take on a limited number of different values; such variables are often referred to as ‘categorical variables’ and

'enumerated type'.

- Factors in R are stored as a vector of integer values with a corresponding set of character values to use when the factor is displayed.
- The function `factor` is used to encode a vector as a factor

```

personnel <- c("Analyst1", "ManagerL2", "Analyst1", "Analyst2",
 "Boss", "ManagerL1", "ManagerL2", "Programmer1",
 "Programmer2", "Programmer3", "Designer1", "Designer2",
 "OtherStaff") # staff in a company

personnel_factors <- factor(personnel)
personnel_factors #sorted alphabetically

[1] Analyst1 ManagerL2 Analyst1 Analyst2 Boss ManagerL1
[7] ManagerL2 Programmer1 Programmer2 Programmer3 Designer1 Designer2
[13] OtherStaff
11 Levels: Analyst1 Analyst2 Boss Designer1 Designer2 ManagerL1 ... Programmer3
str(personnel_factors)

Factor w/ 11 levels "Analyst1","Analyst2",...: 1 7 1 2 3 6 7 9 10 11 ...
personnel2 <- factor(personnel,
 levels = c("Boss", "ManagerL1", "ManagerL2",
 "Analyst1", "Analyst2", "Designer1",
 "Designer2", "Programmer1", "Programmer2",
 "Programmer3", "OtherStaff"))
#do not duplicate the same factors
personnel2

[1] Analyst1 ManagerL2 Analyst1 Analyst2 Boss ManagerL1
[7] ManagerL2 Programmer1 Programmer2 Programmer3 Designer1 Designer2
[13] OtherStaff
11 Levels: Boss ManagerL1 ManagerL2 Analyst1 Analyst2 Designer1 ... OtherStaff
str(personnel2)

Factor w/ 11 levels "Boss","ManagerL1",...: 4 3 4 5 1 2 3 8 9 10 ...
a factor's levels will always be character values.

levels(personnel2) <- c("B", "M1", "M2", "A1", "A2",
 "D1", "D2", "P1", "P2", "P3", "OS")
personnel2

[1] A1 M2 A1 A2 B M1 M2 P1 P2 P3 D1 D2 OS
Levels: B M1 M2 A1 A2 D1 D2 P1 P2 P3 OS

personnel3 <- factor(personnel,
 levels = c("Boss", "ManagerL1", "ManagerL2",

```

```

 "Analyst1", "Analyst2", "Designer1",
 "Designer2", "Programmer1", "Programmer2",
 "Programmer3", "OtherStaff"),
c("B", "M1", "M2", "A1", "A2", "D1", "D2",
 "P1", "P2", "P3", "OS"))
personnel3

[1] A1 M2 A1 A2 B M1 M2 P1 P2 P3 D1 D2 OS
Levels: B M1 M2 A1 A2 D1 D2 P1 P2 P3 OS
Nominal versus ordinal, ordered factors
personnel3[1] < personnel3[2] # error, factors not ordered

Warning in Ops.factor(personnel3[1], personnel3[2]): '<' not meaningful for
factors

[1] NA
tshirts <- c("M", "L", "S", "S", "L", "M", "L", "M")

tshirt_factor <- factor(tshirts, ordered = TRUE,
 levels = c("S", "M", "L"))
tshirt_factor

[1] M L S S L M L M
Levels: S < M < L
tshirt_factor[1] < tshirt_factor[2]

[1] TRUE

```

---

## 1.7 Lists

Lists are the R objects which contain elements of different types: numbers, strings, vectors and other lists. A list can also contain a matrix or a function as one of their elements. A list is created using `list()` function.

Operators for subsetting lists include: - ‘[’ returns a list - ‘[[’ returns the list element - ‘\$’ returns the content of that element in the list

```

c("R good times", 190, 5)

[1] "R good times" "190" "5"
song <- list("R good times", 190, 5)
is.list(song)

[1] TRUE

```

```
str(song)

List of 3
$: chr "R good times"
$: num 190
$: num 5
names(song) <- c("title", "duration", "track")
song

$title
[1] "R good times"
##
$duration
[1] 190
##
$track
[1] 5
song$title

[1] "R good times"
song2 <- list(title="Good Friends",
 duration = 125,
 track = 2,
 rank = 6)

song3 <- list(title="Many Friends",
 duration = 125,
 track= 2,
 rank = 1,
 similar2 = song2)

song[1]

$title
[1] "R good times"
song$title

[1] "R good times"
str(song[1])

List of 1
$ title: chr "R good times"
```

```
song[[1]]

[1] "R good times"
str(song[[1]])

chr "R good times"
song2[3]

$track
[1] 2
song3[5] # a list

$similar2
$similar2$title
[1] "Good Friends"

$similar2$duration
[1] 125

$similar2$track
[1] 2

$similar2$rank
[1] 6
str(song3[5])

List of 1
$similar2:List of 4
..$ title : chr "Good Friends"
..$ duration: num 125
..$ track : num 2
..$ rank : num 6
song3[[5]]

$title
[1] "Good Friends"

$duration
[1] 125

$track
[1] 2

$rank
```

```
[1] 6
song3$similar2

$title
[1] "Good Friends"
##
$duration
[1] 125
##
$track
[1] 2
##
$rank
[1] 6
song[c(1,3)]

$title
[1] "R good times"
##
$track
[1] 5
str(song[c(1,3)])

List of 2
$ title: chr "R good times"
$ track: num 5
result <- song[c(1,3)]
result[1]

$title
[1] "R good times"
result[[1]]

[1] "R good times"
str(result)

List of 2
$ title: chr "R good times"
$ track: num 5
result$title

[1] "R good times"
```

```

result$track

[1] 5
access with [[to content
song3[[5]][[1]]

[1] "Good Friends"
song3$similar2[[1]]

[1] "Good Friends"
Subsets
subset by names
song[c("title", "track")]

$title
[1] "R good times"
##
$track
[1] 5
song3["similar2"]

$similar2
$similar2$title
[1] "Good Friends"
##
$similar2$duration
[1] 125
##
$similar2$track
[1] 2
##
$similar2$rank
[1] 6
resultsimilar <- song3["similar2"]
str(resultsimilar)

List of 1
$ similar2:List of 4
..$ title : chr "Good Friends"
..$ duration: num 125
..$ track : num 2
..$ rank : num 6
resultsimilar1 <- song3[["similar2"]]
str(resultsimilar1)

```

```

List of 4
$ title : chr "Good Friends"
$ duration: num 125
$ track : num 2
$ rank : num 6
resultsimilar1$title

[1] "Good Friends"
subset by logicals
song[c(TRUE, FALSE, TRUE, FALSE)]

$title
[1] "R good times"
##
$track
[1] 5
result3 <- song[c(TRUE, FALSE, TRUE, FALSE)] # is a list of two elements

extending the list
shared <- c("Hillary", "Mari", "Mikel", "Patty")

song3$shared <- shared
str(song3)

List of 6
$ title : chr "Many Friends"
$ duration: num 125
$ track : num 2
$ rank : num 1
$ similar2:List of 4
..$ title : chr "Good Friends"
..$ duration: num 125
..$ track : num 2
..$ rank : num 6
$ shared : chr [1:4] "Hillary" "Mari" "Mikel" "Patty"
cities <- list("Bilbao", "New York", "Tartu")
song3[[["cities"]]] <- cities
str(song3)

List of 7
$ title : chr "Many Friends"
$ duration: num 125
$ track : num 2
$ rank : num 1
$ similar2:List of 4

```

```

..$ title : chr "Good Friends"
..$ duration: num 125
..$ track : num 2
..$ rank : num 6
$ shared : chr [1:4] "Hillary" "Mari" "Mikel" "Patty"
$ cities :List of 3
..$: chr "Bilbao"
..$: chr "New York"
..$: chr "Tartu"

```

---

## 1.8 Data frames

A data frame is the data structure most often used for data analyses. A data frame is a list of equal-length vectors. Each element of the list can be thought of as a column and the length of each element of the list is the number of rows. As a result, data frames can store different classes of objects in each column (i.e. numeric, character, factor).

The `tidyverse` package provides a version of the data frame called `tibble`

```

thenames <- c("Ane", "Mike", "Laura", "Viktoria", "Martin")
ages <- c(44, 20, 33, 15, 65)
employee <- c(FALSE, FALSE, TRUE, TRUE, FALSE)

mydataframe <- data.frame(thenames, ages, employee)
mydataframe

```

```

thenames ages employee
1 Ane 44 FALSE
2 Mike 20 FALSE
3 Laura 33 TRUE
4 Viktoria 15 TRUE
5 Martin 65 FALSE

```

```
names(mydataframe) <- c("FirstName", "Age", "Employee")
```

```
str(mydataframe)
```

```

'data.frame': 5 obs. of 3 variables:
$ FirstName: chr "Ane" "Mike" "Laura" "Viktoria" ...
$ Age : num 44 20 33 15 65
$ Employee : logi FALSE FALSE TRUE TRUE FALSE

```

*#strings are not factors!*

```

mydataframe <- data.frame(thenames, ages, employee,
 stringsAsFactors=FALSE)

```

```

names(mydataframe) <- c("FirstName", "Age", "Employee")
str(mydataframe)

'data.frame': 5 obs. of 3 variables:
$ FirstName: chr "Ane" "Mike" "Laura" "Viktoria" ...
$ Age : num 44 20 33 15 65
$ Employee : logi FALSE FALSE TRUE TRUE FALSE
subset data frame

mydataframe[4,2]

[1] 15
mydataframe[4, "Age"]

[1] 15
mydataframe[, "FirstName"]

[1] "Ane" "Mike" "Laura" "Viktoria" "Martin"
mydataframe[c(2,5), c("Age", "Employee")]

Age Employee
2 20 FALSE
5 65 FALSE
matfromframe <- as.matrix(mydataframe[c(2,5), c("Age", "Employee")])
str(matfromframe)

num [1:2, 1:2] 20 65 0 0
- attr(*, "dimnames")=List of 2
..$: chr [1:2] "2" "5"
..$: chr [1:2] "Age" "Employee"
mydataframe[3]

Employee
1 FALSE
2 FALSE
3 TRUE
4 TRUE
5 FALSE
convert to vector
mydf0 <- mydataframe[3] #data.frame
str(mydf0)

'data.frame': 5 obs. of 1 variable:
$ Employee: logi FALSE FALSE TRUE TRUE FALSE

```

```

myvec <- mydataframe[[3]] #vector
str(myvec)

logi [1:5] FALSE FALSE TRUE TRUE FALSE
mydf0asvec <- as.vector(mydataframe[3]) # but it doesn't work . Use []
str(mydf0asvec)

'data.frame': 5 obs. of 1 variable:
$ Employee: logi FALSE FALSE TRUE TRUE FALSE
mydf0asvec <- as.vector(mydataframe[[3]])
str(mydf0asvec)

logi [1:5] FALSE FALSE TRUE TRUE FALSE
add column
height <- c(166, 165, 158, 176, 199)
weight <- c(66, 77, 99, 88, 109)
mydataframe$height <- height
mydataframe[["weight"]] <- weight
mydataframe

FirstName Age Employee height weight
1 Ane 44 FALSE 166 66
2 Mike 20 FALSE 165 77
3 Laura 33 TRUE 158 99
4 Viktoria 15 TRUE 176 88
5 Martin 65 FALSE 199 109

add a column

birthplace <- c("Tallinn", "London", "Donostia", "Paris", "New York")

mydataframe <- cbind(mydataframe, birthplace)
mydataframe

FirstName Age Employee height weight birthplace
1 Ane 44 FALSE 166 66 Tallinn
2 Mike 20 FALSE 165 77 London
3 Laura 33 TRUE 158 99 Donostia
4 Viktoria 15 TRUE 176 88 Paris
5 Martin 65 FALSE 199 109 New York

add a row

anton <- data.frame(FirstName = "Anton", Age = 77, Employee=TRUE, height= 170, weight =
mydataframe <- rbind (mydataframe, anton)
mydataframe

```

```

FirstName Age Employee height weight birthplace
1 Ane 44 FALSE 166 66 Tallinn
2 Mike 20 FALSE 165 77 London
3 Laura 33 TRUE 158 99 Donostia
4 Viktoria 15 TRUE 176 88 Paris
5 Martin 65 FALSE 199 109 New York
6 Anton 77 TRUE 170 65 Amsterdam

sorting

mydataframeSorted <- mydataframe[order(mydataframe$Age, decreasing = TRUE),] #all columns
mydataframeSorted

FirstName Age Employee height weight birthplace
6 Anton 77 TRUE 170 65 Amsterdam
5 Martin 65 FALSE 199 109 New York
1 Ane 44 FALSE 166 66 Tallinn
3 Laura 33 TRUE 158 99 Donostia
2 Mike 20 FALSE 165 77 London
4 Viktoria 15 TRUE 176 88 Paris

mydataframeSorted2 <- mydataframe[order(mydataframe$Age, decreasing = TRUE), c(1,2,6)]
mydataframeSorted2

FirstName Age birthplace
6 Anton 77 Amsterdam
5 Martin 65 New York
1 Ane 44 Tallinn
3 Laura 33 Donostia
2 Mike 20 London
4 Viktoria 15 Paris

```

## 1.9 R Functional Functions

R is a functional language and there are some special functions provided:  
`apply()`, `lapply()`, `sapply()`, `tapply()`, `mapply()`, `vapply()`

One of its main strengths lies in the use of the functions `apply()` (and all its variations) on lists, matrices, data frames or other data structures. The `tidyverse` package provides the `purrr` package for functional programming. The topic of functional programming lies beyond the purpose of this introduction.

Most of the commands that we use in our scripts are functions applied to data.

## 1.10 Environments

An environment is a place where R stores variables that is where R binds a set of names to a set of object values. An environment is something like a bag or a list of names. Every name in an environment is unique.

The top level environment `R_GlobalEnv` is created when we start up R and is the global environment. Every environment has parent environment. When we define a function, a new environment is created.

### 1.10.1 Global variables, local variables and programming scope

Global variables are those variables which exists throughout the execution of a program. Local variables are those variables which exist only within a certain part of a program like a function. The super-assignment operator, `«-`, is used to make assignments to global variables or to make assignments in the parent environment.

```
variables and functions in the current environment
ls()
```

```
[1] "ages" "anton" "aphase"
[4] "aseqofvalues" "aseqofvalues10" "aseqofvalues2"
[7] "aseqofvalues3" "aseqofvalues4" "aseqofvalues5"
[10] "aseqofvalues6" "aseqofvalues7" "aseqofvalues8"
[13] "aseqofvalues9" "birthplace" "char2int"
[16] "char2num" "cities" "costs"
[19] "ctl" "earnings" "employee"
[22] "falsenum" "group" "height"
[25] "incomes" "index" "lm_1"
[28] "lm.D9" "lm.D90" "logicalcond"
[31] "m1" "matchar" "matchars"
[34] "matfromframe" "matnum" "mydataframe"
[37] "mydataframeSorted" "mydataframeSorted2" "mydf0"
[40] "mydf0asvec" "mymat" "mymat2"
[43] "mymat3" "mymat4" "mymat5"
[46] "myvec" "num2char" "opar"
[49] "personnel" "personnel_factors" "personnel2"
[52] "personnel3" "phase1" "phase3"
[55] "phases" "phases1" "phases2"
[58] "phases3" "result" "result3"
[61] "resultsimilar" "resultsimilar1" "row1"
[64] "row2" "row3" "selectionv"
[67] "selectionvec2" "shared" "song"
[70] "song2" "song3" "testphases"
[73] "thenames" "thevalues" "thevalues1"
```

```

[76] "thevalues2" "trt" "truenum"
[79] "tshirt_factor" "tshirts" "var1"
[82] "vAr1" "vector1" "vector2"
[85] "weight" "x" "y"
to get the current environment
environment()

<environment: R_GlobalEnv>
the base environment is the environment of the base package
baseenv()

<environment: base>
list of environments
search()

[1] ".GlobalEnv" "package:stats" "package:graphics"
[4] "package:grDevices" "package:utils" "package:datasets"
[7] "package:methods" "Autoloads" "package:base"

functions and environments
in this example we do not return any value

myfunction <- function() {
 myvar_a<- 50
 myfunctioninside <- function() {
 myvar_a <- 100
 # myvar_a <<- 100
 print(myvar_a)
 }
 myfunctioninside()
 print(myvar_a)
 # myvar_a <<- 100
}

myvar_a <- 10
myfunction()

[1] 100
[1] 50
print(myvar_a)

[1] 10
create environment
my_env <- new.env()
my_env

```

```

<environment: 0x55d7d29f4c80>
ls(my_env)

character(0)
character(0)

character(0)
assign("myvar_a", 700, envir=my_env)
my_env$mytext = " a text"
ls(my_env)

[1] "mytext" "myvar_a"
myvar_a

[1] 10
my_env$myvar_a

[1] 700
parent.env(my_env)

<environment: R_GlobalEnv>
get('myvar_a', envir=my_env)

[1] 700

```

## 1.11 Reading Data

R is capable of reading most formats including CSV, MS Excel formats (xlsx, etc.) as well as other statistical packages (e.g. SAS, SPSS, etc.) and data mining tools such as ARFF (Weka's format).

R also provides has two native data formats, Rdata and Rds. These formats are used when leaving or starting an R session so that R objects can be stored or retrieved to continue in the same state). While Rdata is used to save multiple R objects, Rds is used to save a single R object.

```
load("data.rdata") # It is needed to be in the same directory (setwd())
```

To read CSV (Comma Separated Values) files in R

```
Import the data and look at the first six rows
f <- read.csv("data.csv")
head(f)
```

To read ARFF files, we can use the `foreign` library.

```

library(foreign)
isbsg <- read.arff("datasets/effortEstimation/isbsg10teaser.arff")

mydataISBSG <- isbsg[, c("FS", "N_effort")]
str(mydataISBSG)

'data.frame': 37 obs. of 2 variables:
$ FS : num 225 599 333 748 158 427 461 257 115 116 ...
$ N_effort: num 1856 10960 5661 1518 3670 ...

```

---

## 1.12 Plots

There are several graphic packages that are recommended, in particular `ggplot`. However, there is some basic support in the R base for graphics. The following Figure ?? shows a simple plot.

```
plot(mydataISBSG$FS, mydataISBSG$N_effort)
```

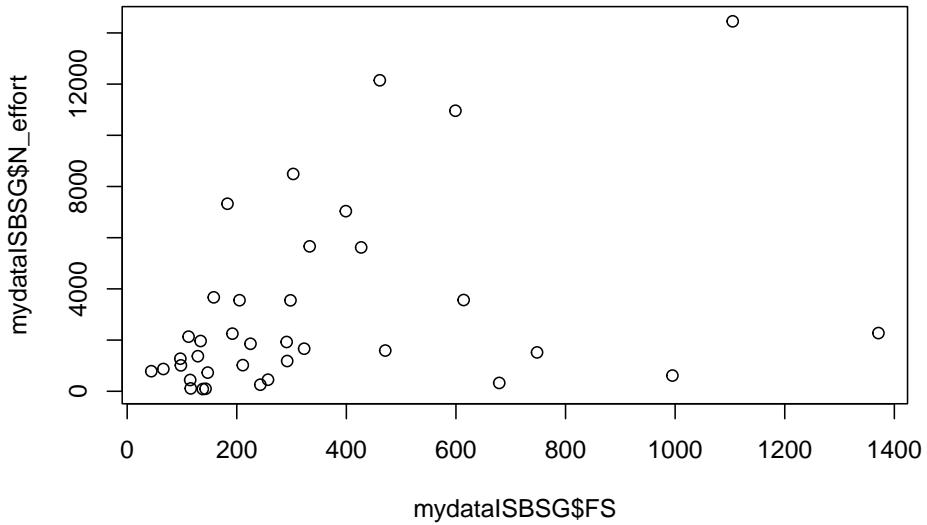


Figure 1.1: Simple plot

---

## 1.13 Control flow in R

R provides most common control flow structures found in most languages

```

if
x <- 6
if (x >= 5) {
 "x is greater than or equals 5"
} else {
 "x is smaller than 5"
}

[1] "x is greater than or equals 5"

ifelse
library(foreign)
kc1 <- read.arff("datasets/defectPred/D1/KC1.arff")
kc1$Defective <- ifelse(kc1$Defective == "Y", 1, 0)
head(kc1, 1)

LOC_BLANK BRANCH_COUNT LOC_CODE_AND_COMMENT LOC_COMMENTS
1 0 1 0 0
CYCLOMATIC_COMPLEXITY DESIGN_COMPLEXITY ESSENTIAL_COMPLEXITY LOC_EXECUTABLE
1 1 1 1 3
HALSTEAD_CONTENT HALSTEAD_DIFFICULTY HALSTEAD EFFORT HALSTEAD_ERROR_EST
1 11.58 2.67 82.35 0.01
HALSTEAD_LENGTH HALSTEAD_LEVEL HALSTEAD_PROG_TIME HALSTEAD_VOLUME
1 11 0.38 4.57 30.88
NUM_OPERANDS NUM_OPERATORS NUM_UNIQUE_OPERANDS NUM_UNIQUE_OPERATORS LOC_TOTAL
1 4 7 3 4 5
Defective
1 0

for loops
for(x in 1:5){
 print(x)
}

```

## 1.14 Built-in Datasets

R comes with some built-in datasets ready to use Description of datasets <http://www.sthda.com/english/wiki/r-built-in-data-sets>

```
data() #list of datasets already available
```

Then, to load a dataset is as follows.

```
load the mtcars Motor Trend Car Road Tests
data("mtcars")
```

And another example.

```
Monthly Airline Passenger Numbers 1949-1960
Time series object ts() convert a vector to a time series
data("AirPassengers")
str(AirPassengers)
plot(AirPassengers)
```

## 1.15 Other tools with R

### 1.15.1 Rattle

There is graphical interface, Rattle, that allow us to perform some data mining tasks with R (?).

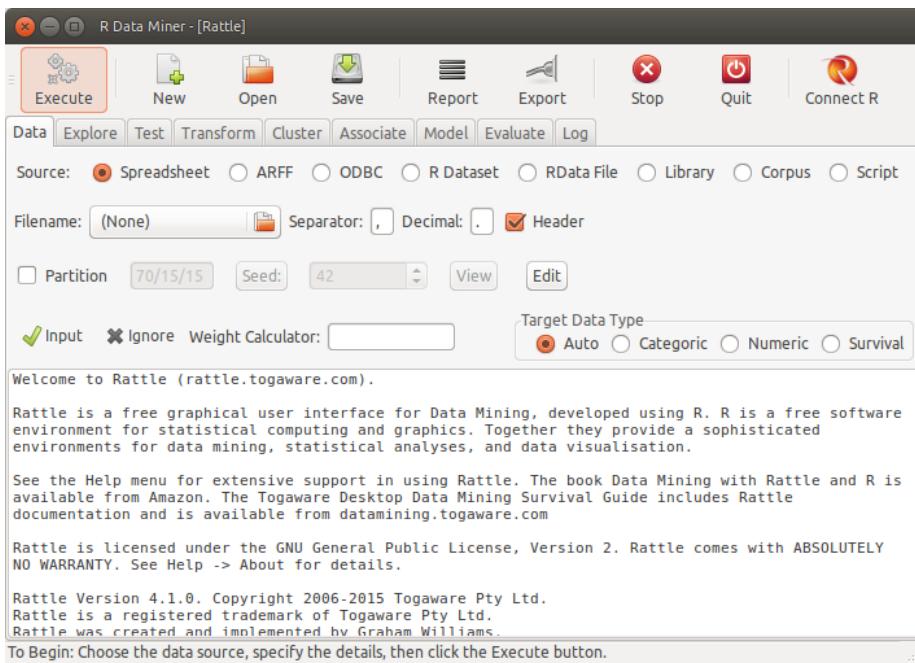


Figure 1.2: Rattle: GUI for Data mining with R

### 1.15.2 Jamovi

GUI for statistical analysis in R. It allow us to export the actual R code. Its Website is: <https://www.jamovi.org/>

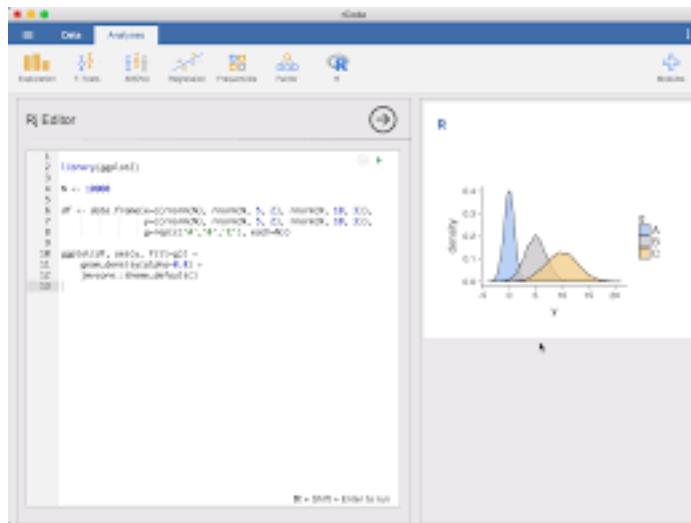


Figure 1.3: Jamovi

### 1.15.3 JASP

There is another GUI for statistics, JASP, but it is not so easy at the moment to export the R code. <https://jasp-stats.org/>

## **Part II**

# **Introduction to Data Mining**



We will deal with extracting information from data, either for estimation, defect prediction, planning, etc.

We will provide an overview of data analysis using different techniques.



## Chapter 2

# What is Data Mining / Knowledge Discovery in Databases (KDD)

The non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (?)

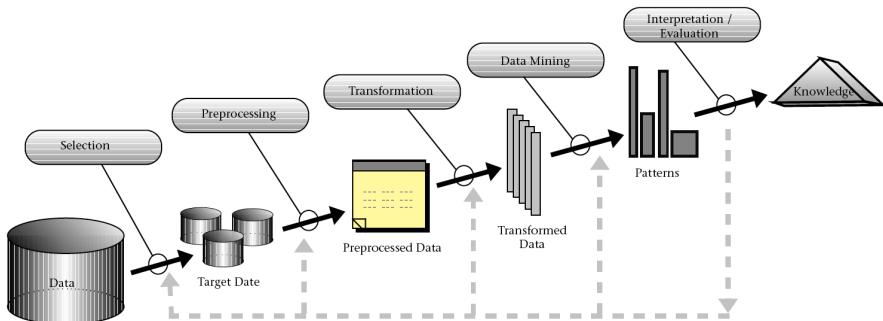


Figure 2.1: KDD Process

The Cross Industry Process for Data Mining (CRISP-DM) also provides a common and well-developed framework for delivering data mining projects identifying six steps (?):

1. Problem Understanding
2. Data Understanding

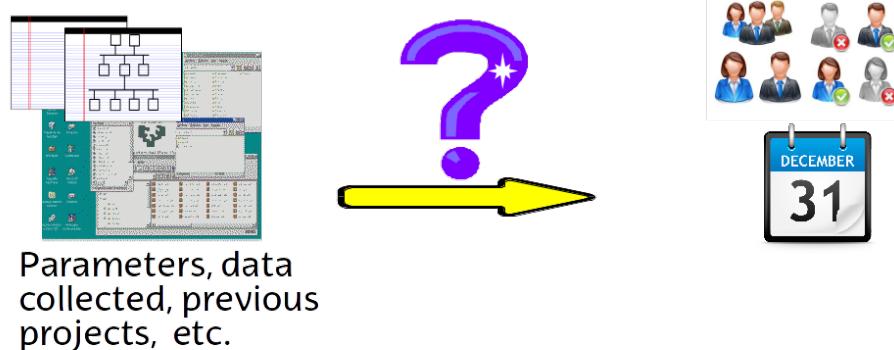
3. Data Preparation
4. Modeling
5. Evaluation
6. Deployment



Figure 2.2: CRISP-DM (Wikipedia)

## 2.1 The Aim of Data Analysis and Statistical Learning

- The aim of any data analysis is to **understand the data**
- and to build models for making predictions and estimating future events based on past data
- and to make statistical inferences from our data.
- We may want to test different hypothesis on the data
- We want to generate conclusions about the population where our sample data comes from
- Most probably we are interested in building a model for quality, time, defects or effort prediction



- We want to find a function  $f()$ , that given  $X_1, X_2, \dots$  computes  $Y = f(X_1, X_2, \dots, X_n)$

## 2.2 Data Science

Data science (DS) is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structured and unstructured data. Data science is related to data mining, machine learning and big data.

We may say that the term DS embraces all terms related to data analysis that previously were under different disciplines.

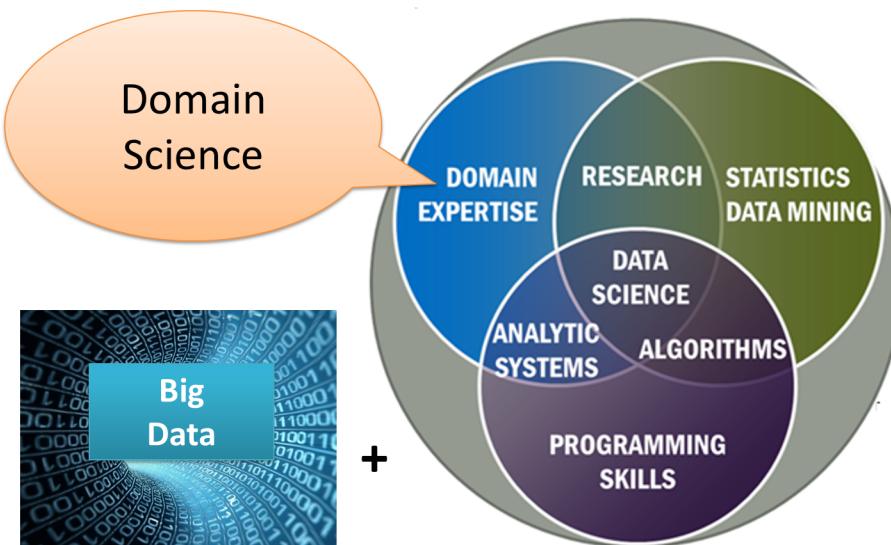


Figure 2.3: Wikipedia Data Science

## 2.3 Some References

- W.N. Venables, D.M. Smith and the R Core Team, *An Introduction to R*

Generic books about statistics:

- John Verzani, *simpleR - Using R for Introductory Statistics*
- Peter Dalgaard, *Introductory Statistics with R*, 2nd Edt., Springer, 2008
- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer, 2013
- Geoff Cumming, *Understanding the New Statistics: Effect Sizes, Confidence Intervals, and Meta-Analysis*, Routledge, New York, 2012

## 2.4 Data Mining and Data Science with R

- R for Data Science
- Practical Data Science with R \*R for Everyone: Advanced Analytics and Graphics
  - Graham Williams, *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery*, Springer 2011  
Also the author maintains a Web site: <http://rattle.togaware.com/>
  - Luis Torgo, *Data Mining with R: Learning with Case Studies*, Chapman and Hall/CRC, 2010
  - <http://www.rdatamining.com/>

## 2.5 Data Mining with Weka

Weka is another popular framework written in Java that can be used and extended with other languages and frameworks. The authors of Weka also have a popular book:

- Ian Witten, Eibe Frank, Mark Hall, Christopher J. Pal, Data Mining: Practical Machine Learning Tools and Techniques (4th Edt), Morgan Kaufmann, 2016, ISBN: 978-0128042915

## 2.6 R Markdown

Cheatsheet link for the markdown documents <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

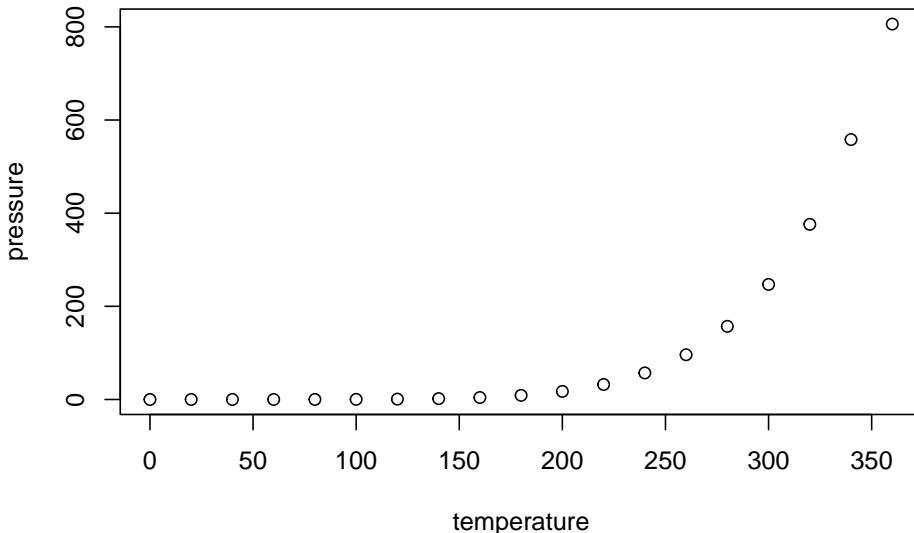
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
speed dist
Min. : 4.0 Min. : 2.00
1st Qu.:12.0 1st Qu.: 26.00
Median :15.0 Median : 36.00
Mean :15.4 Mean : 42.98
3rd Qu.:19.0 3rd Qu.: 56.00
Max. :25.0 Max. :120.00
```

## 2.7 Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

## 2.8 References

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>  
<https://rmarkdown.rstudio.com/lesson-15.html>

---

output:  
html\_document  
pdf\_document:  
  de-  
fault

---

##  
R  
and  
Python

output:  
html\_document  
pdf\_document:  
de-  
fault  
R  
and  
Python  
can  
in-  
ter-  
act  
to-  
gether  
via  
the  
*retic-*  
*ulate*  
pack-  
age.  
The  
doc-  
u-  
men-  
ta-  
tion  
for  
the  
**reticulate**  
pack-  
age  
can  
be  
found  
here:  
http  
s:  
//rs  
tudi  
o.cgi  
thub  
.io/  
reti  
cula  
te/  
/

---

output:  
html\_document  
pdf\_document:  
  de-  
  fault

---



\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
Instructions  
for  
con-  
fig-  
ur-  
ing  
the  
sys-  
tem  
can  
be  
found  
at  
RStu-  
dio  
site:  
http  
s:  
//su  
ppor  
t.rs  
tudi  
o.co  
m/  
hc/e  
n-us  
/art  
icle  
s/36  
0023  
6544  
74-  
Inst  
alli  
ng-  
and-  
Co  
nfig  
urin  
g-Py  
th  
on-  
wi  
th-  
RS  
tudi  
o.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
Or  
we  
can  
cre-  
ate  
our  
envi-  
ron-  
ment  
fol-  
low-  
ing -  
[R  
and  
Python  
– a  
happy  
union  
with  
retic-  
    u-  
    late  
    we-  
    bi-  
    nar]ht  
    tps:  
    //  
    ww  
w.yo  
utub  
e.co  
m/  
watc  
h?v  
=8  
WE-  
EU  
5k97  
Q&t  
=2  
7s

```

output:
html_document
pdf_document:
 de-
 fault

r
install.packages("reticulate")
Note
that
the
reticulate
pack-
age
needs
Python
>=
2.7
and
for
NumPy
re-
quires
NumPy
>=
1.6.
Us-
ing
Python
with
RMark-
down
and
RStu-
dio
The
R
build-
in
dataset
will
be
used
later
```

```

| output:
| html_document
| pdf_document:
| de-
| fault
| _____
| r
library("reticulate")

use_virtualenv("myenv")
data("mtcars")
python
print("Hello
Python!")

Hello
Python!
```

---

output:

html\_document

pdf\_document:

de-

fault

---

“‘python

dat-

ac-

trs

= {

‘CHN’:

{‘COUN-

TRY’:

‘China’,

‘POP’:

1\_398.72,

‘AREA’:

9\_596.96,

‘GDP’:

12\_234.78,

‘CONT’:

‘Asia’},

‘IND’:

{‘COUN-

TRY’:

‘In-

dia’,

‘POP’:

1\_351.16,

‘AREA’:

3\_287.26,

‘GDP’:

2\_575.67,

‘CONT’:

‘Asia’,

‘IND\_DAY’:

‘1947-

08-

15’},

‘USA’:

{‘COUN-

TRY’:

‘US’,

‘POP’:

329.74,

‘AREA’:

9\_833.52,

‘GDP’:

19\_485.39,

‘CONT’:

‘N.America’,

‘IND\_DAY’:

‘1776-

07-

04’},

‘IDN’

```

output:
html_document
pdf_document:
 de-
 fault

columns
 =
('COUN-
TRY',
'POP',
'AREA',
'GDP',
'CONT',
'IND_DAY')
```
``python
import
port
pan-
das
as
pd
im-
port
seaborn
as
sns
#ubuntu
#sudo
apt-
get
in-
stall
-y
python3-
seaborn
im-
port
mat-
plotlib.pyplot
as
plt
```

```

output:
html_document
pdf_document:
de-
fault


---


tips
=
sns.load_dataset("tips")
mylist
=
[“youtube”,
‘linkedin’,
‘1lit-
tle-
coder’]
sns.scatterplot(x=tips[‘total_bill’],
y =
tips[‘tip’],
hue=tips[‘day’])
plt.show()
```


```

```

python
fmri
=
sns.load_dataset("fmri")
r
f1
<-
subset(py$fmri,
region
==
"parietal")
python
import
matplotlib
as
mpl
sns.lmplot("timepoint","signal",
data=r.f1)


```

```

python
mpl.pyplot.show()

```

```

output:
html_document
pdf_document:
 de-
 fault


python
sns.lmplot("mpg",
"cyl",
data=r.mtcars)

python
mpl.pyplot.show()

python
import
pandas
as
pd
df
=
pd.DataFrame(data=datactrs,
index=columns).T
df
```

---

output:

html\_document

pdf\_document:

de-

fault

---

##

COUNTRY

POP

AREA

GDP

CONT

IND\_DAY

##

CHN

China

1398.72

9596.96

12234.78

Asia

NaN

##

IND

India

1351.16

3287.26

2575.67

Asia

1947-08-15

##

USA

US

329.74

9833.52

19485.39

N.America

1776-07-04

##

IDN

Indonesia

268.07

1910.93

1015.54

Asia

1945-08-17

##

BRA

Brazil

210.32

8515.77

2055.51

S.America

1822-09-07

##

PAK

Pakistan

205.71

```

output:
html_document
pdf_document:
 de-
 fault

python
df.to_csv('datasets/data_countries.csv')
We
can
read
the
dataset
from
python
python
df1
 =
pd.read_csv("datasets/other/data_countries.csv",
index_col=0)
Use
R to
read
and
write
data
from
 a
pack-
age
 r
library("nycflights13")
write.csv(flights,
"datasets/other/flights.csv")
Use
python
 to
read
 the
dataset
 and
 pro-
cess
 the
data
```

```

output:
html_document
pdf_document:
 de-
 fault

python
import
pandas
flights
 =
pandas.read_csv("datasets/other/flights.csv")
flights
 =
flights[flights['dest']
==="ORD"]
flights
 =
flights[['carrier',
'dep_delay',
'arr_delay']]
flights
 =
flights.dropna()
print(flights.head())
```

---

```
output:
html_document
pdf_document:
 de-
 fault
 ##
 carrier
 dep_delay
 arr_delay
 ##
 5
 UA
-4.0
12.0
 ##
 9
 AA
-2.0
8.0
 ##
 25
 MQ
8.0
32.0
 ##
 38
 AA
-1.0
14.0
 ##
 57
 AA
-4.0
4.0
 Use
Python
 for
plot-
ting
```

```

output:
html_document
pdf_document:
 de-
fault

python
import
matplotlib.pyplot
 as
 plt
import
numpy
 as
 np
t =
np.arange(0.0,
2.0,
0.01)
s =
1 +
np.sin(2*np.pi*t)
plt.plot(t,s)
plt.xlabel('time
(s)')
plt.ylabel('voltage
(mV)')
plt.grid(True)
plt.savefig("test.png")
plt.show()
```



Use  
R  
for  
plot-  
ting  
Python  
ob-  
jects:



```

output:
html_document
pdf_document:
 de-
 fault

Note
that
the
echo
 =
FALSE
 pa-
 ram-
 eter
 was
added
 to
 the
code
chunk
 to
 pre-
 vent
print-
 ing
 of
 the
 R
code
that
gen-
 er-
ated
 the
plot.
##
```

Ref-

er-

ences

output:  
html\_document  
pdf\_document:  
  de-  
  fault

-  
<https://rstudio.com/resources/webinars/rstudio-a-single-home-for-r-and-python/>  
- R  
  in-  
  ter-  
  face  
  to  
Python  
  - 3  
Wild-  
Caught  
  R  
  and  
Python  
  Ap-  
  pli-  
  ca-  
  tions

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
-  
RStu-  
dio  
+  
Python,  
Vi-  
sual  
Mark-  
down  
Edi-  
tor -  
RStu-  
dio  
Lat-  
est  
Up-  
date  
-  
[Ar-  
rays  
in R  
and  
  in  
Python]ht  
  tps:  
  //rs  
  tudi  
  o.cgi  
  thub  
  .io/  
  reti  
  cula  
  te/a  
  rtic  
  les/  
  arra  
  ys.h  
  tml

output:  
html\_document  
pdf\_document:  
  de-  
  fault

-  
http  
s://  
ww  
w.r-  
blog  
gers  
.com  
/202  
1/02  
/pyt  
  ho  
  ns-  
  pa  
  nd  
  as-  
  vs-  
  rs-  
dply  
r-wh  
  ich-  
    is-  
    the-  
best-  
  da  
  ta-  
anal  
ysis-  
libr  
ary/  
  ?u  
  tm  
\_sou  
rce=  
  feed  
burn  
  er  
  &u  
  tm  
  \_m  
ediu  
  m=  
emai  
l&u  
  tm  
  \_c  
  am  
paig  
  n=  
  Fe  
  \_d

output:  
html\_document  
pdf\_document:  
  de-  
  fault

#  
(PART)  
Data  
Sources  
and  
Met-  
rics  
and  
Stan-  
dards  
  in  
Software  
  En-  
  gi-  
neer-  
ing  
  De-  
fect  
Pre-  
dic-  
tion  
  {-}  
  #  
Data  
Sources  
  in  
Software  
  En-  
  gi-  
neer-  
ing

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault

We  
clas-  
sify  
this  
trail  
in  
the  
fol-  
low-  
ing  
cate-  
gories:

\*

*Source*  
code  
can  
be  
stud-  
ied  
to  
mea-  
sure  
its  
prop-  
er-  
ties,  
such  
as  
size  
or  
com-  
plex-  
ity.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_\*

*Source*  
*Code*  
*Man-*  
  *age-*  
  *ment*  
  *Sys-*  
  *tems*  
(SCM)  
make  
  it  
  pos-  
  sible  
  to  
store  
  all  
  the  
changes  
that  
  the  
dif-  
fer-  
ent  
source  
code  
files  
  un-  
dergo  
  dur-  
  ing  
  the  
project.

Also,  
SCM  
  sys-  
tems  
  al-  
  low  
  for  
work  
  to  
  be  
done  
  in  
  par-  
  allel  
  by  
dif-  
fer-  
ent  
  de-

\_\_\_\_\_

output:

html\_document

pdf\_document:

de-

fault

\_\_\_\_\_

\* Is-

sue

or

Bug

track-

ing

sys-

tems

(ITS).

Bugs,

de-

fects

and

user

re-

quests

are

man-

aged

in

ISTS,

where

users

and

de-

vel-

op-

ers

can

fill

tick-

ets

with

a de-

scrip-

tion

of a

de-

fect

found,

or a

de-

sired

new

func-

tion-

ality.

All

the

changes

to

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
\*

*Mes-*  
*sages*  
  be-  
  tween  
  de-  
  vel-  
  op-  
  ers  
  and  
  users.

In  
the  
case  
  of  
free/open  
source  
soft-  
ware,  
the  
projects  
  are  
open  
  to  
the  
world,  
and  
the  
mes-  
sages  
  are  
archived  
  in  
the  
form  
  of  
mail-  
ing  
lists  
and  
so-  
cial  
net-  
works  
which  
  can  
also  
  be  
mined  
  for

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
\*

*Meta-  
data  
about  
the  
projects.*

As  
well  
as  
the  
low  
level  
in-  
for-  
ma-  
tion  
of  
the  
soft-  
ware  
pro-  
cesses,

we  
can  
also  
find  
meta-  
data  
about  
the  
soft-  
ware  
projects  
which

can  
be  
use-  
ful  
for  
re-  
search.

This  
meta-  
data  
may  
in-  
clude  
intended-  
audience,  
pro-

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault



\*

*Us-  
age  
data.*

There  
are  
statis-  
tics  
about  
soft-  
ware  
down-  
loads,  
logs  
from  
servers,  
soft-  
ware

re-  
views,  
etc.

Types  
of  
in-  
for-  
ma-  
tion  
stored  
in  
the  
repos-  
ito-  
ries:

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_\*

Meta-  
information  
about  
the  
project  
it-  
self  
and  
the  
peo-  
ple  
that  
par-  
tici-  
pated.

+

Low-  
level  
in-  
for-  
ma-  
tion  
\*

Mail-  
ing  
Lists  
(ML)

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\*  
  
Bug  
Track-  
ing  
Sys-  
tems  
(BTS)  
or  
Project  
Tracker  
Sys-  
tem  
(PTS)  
\*  
  
Soft-  
ware  
Con-  
figu-  
ra-  
tion  
Man-  
age-  
ment  
Sys-  
tems  
(SCM)

output:  
html\_document  
pdf\_document:  
de-  
fault  
+  
Pro-  
cessed  
in-  
for-  
ma-  
tion.  
For  
ex-  
am-  
ple  
project  
man-  
age-  
ment  
in-  
for-  
ma-  
tion  
about  
the  
ef-  
fort  
esti-  
ma-  
tion  
and  
cost  
of  
the  
project.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_\*

Whether  
the  
repos-  
itory  
is  
pub-  
lic  
or  
not

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
\*

Sin-  
gle  
project  
vs. mul-  
ti-  
pro-  
jects.  
Whether  
the  
repos-  
itory  
con-  
tains  
in-  
for-  
ma-  
tion  
of a  
sin-  
gle  
project  
with  
mul-  
ti-  
ples  
ver-  
sions  
or  
mul-  
ti-  
ples  
projects  
and/or  
ver-  
sions.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
\*  
Type  
of  
con-  
tent,  
open  
source  
or  
in-  
dus-  
trial  
projects  
\*  
For-  
mat  
in  
which  
the  
in-  
for-  
ma-  
tion  
is  
stored  
and  
for-  
mats  
or  
tech-  
nolo-  
gies  
for  
ac-  
cess-  
ing  
the  
in-  
for-  
ma-  
tion:

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
+  
Text.  
It  
can  
be  
just  
plain  
text,  
CSV  
(Comma  
Sep-  
a-  
rated  
Val-  
ues)  
files,  
Attribute-  
Relation  
File  
For-  
mat  
(ARFF)  
or  
its  
vari-  
ants  
+  
Through  
databases.  
Down-  
load-  
ing  
dumps  
of  
the  
database.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
+  
  Re-  
  mote  
    ac-  
    cess  
    such  
    as  
  APIs  
    of  
  Web  
    ser-  
    vices  
    or  
  REST  
    #  
  Repos-  
    ito-  
    ries  
There  
  is a  
  num-  
  ber  
  of  
  open  
    re-  
    search  
  repos-  
    ito-  
    ries  
    in  
  Soft-  
  ware  
    En-  
    gi-  
  neer-  
    ing.  
Among  
them:

output:  
html\_document  
pdf\_document:  
de-  
fault  
+  
Zen-  
odo.  
It is  
be-  
com-  
ing  
a  
pop-  
ular  
site  
for  
pub-  
lish-  
ing  
datasets  
asso-  
ci-  
ated  
with  
pa-  
pers.  
It  
pro-  
vides  
DOIs  
for  
ref-  
er-  
enc-  
ing  
data  
and  
code:  
http  
s://  
zeno  
do.o  
rg/

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
+  
Spinel-  
  lis  
  main-  
  tais  
  a cu-  
  rated  
  repos-  
  itory  
  on  
Github:  
http  
  s:  
  //gi  
  thub  
  .com  
  /dsp  
  inel  
  lis/  
  awes  
  ome-  
  msr

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
+  
PROMISE  
(Pre-  
dic-  
tOr  
Mod-  
els  
In  
Soft-  
ware  
En-  
gi-  
neer-  
ing).  
There  
is a  
con-  
fer-  
ence  
with  
this  
name  
(Promise  
Con-  
fer-  
ence)

\_\_\_\_\_

output:

html\_document

pdf\_document:

de-

fault

\_\_\_\_\_

Some

pop-

ular

datasets

used

as

bench-

mark-

ing

in

may

pa-

per

can

still

be

found

on:

http:

//pr

omis

e.si

te.u

otta

wa.c

a/

SE

Re

posi

tory

/dat

aset

s-pa

ge.h

tml

The

is

some

well-

known

is-

sues

wit

the

NASA

datasets

and

the

source

code

..

output:  
html\_document  
pdf\_document:  
  de-  
  fault

+  
FLOSS-  
Mole  
  (?)  
http:  
  //fl  
  os  
smol  
e.or  
  g/  
  +  
FLOSS-  
Met-  
rics  
  (?)  
http:  
  //fl  
  os  
smet  
rics  
.org  
  /  
  +  
Qual-  
itas  
Cor-  
pus  
(QC)  
  (?)  
http:  
  //qu  
  alit  
asco  
rpus  
.com  
  /

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
+  
Sourcerer  
Project  
  (?):  
http:  
  //so  
urce  
  rer.  
  ics.  
  uci.  
edu/  
  +  
Ulti-  
mate  
  De-  
  bian  
Database  
(UDD)  
  (?)  
http:  
  //ud  
d.de  
bian  
.org  
  /  
  +  
Source-  
Forge  
  Re-  
  search  
Data  
Archive  
(SRDA)  
  (?)  
http:  
  //ze  
rlot  
.cse  
.nd.  
edu/

output:  
html\_document  
pdf\_document:  
de-  
fault

+  
Software-  
artifact  
In-  
fras-  
truc-  
ture  
Repos-  
itory  
(SIR)  
[ht  
tp:  
//sir.  
unl.  
edu]  
+  
Open-  
Hub:  
http  
s://  
ww  
w.op  
en  
hub.  
net/  
Not  
openly  
avail-  
able  
(and  
mainly  
for  
ef-  
fort  
esti-  
ma-  
tion):

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
+  
The  
  In-  
  ter-  
  na-  
  tional  
  Soft-  
  ware  
  Bench-  
  mark-  
    ing  
  Stan-  
  dards  
  Group  
    (IS-  
    BSG)  
http:  
  //  
  ww  
  w.is  
  bsg.  
  org/

Some  
  pa-  
  pers  
  and  
  pub-  
  lica-  
  tions/theses  
  that  
  have  
  been  
  used  
    in  
    the  
  liter-  
    a-  
  ture:

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
+  
He-  
lix  
Data  
Set  
(?):  
http:  
  //  
  ww  
  w.ic  
  t.sw  
  in.e  
  du.a  
  u/re  
  sear  
  ch/p  
  roje  
  cts/  
  heli  
  x/  
  +  
Bug  
Pre-  
dic-  
tion  
Dataset  
(BPD)  
  (?,  
  ?):  
http:  
//bu  
g.inf.  
  usi.  
  ch/

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
+  
Eclipse  
Bug  
Data  
(EBD)  
  (?,  
  ?):  
http:  
  //  
  ww  
  w.st  
  .cs.  
uni-  
saar  
land  
.de/  
soft  
evo/  
bug-  
data  
/ecl  
ipse  
  /  
  #  
Open  
Tools/Dashboards  
  to  
  ex-  
tract  
data  
Process  
  to  
  ex-  
tract  
data:  
\_\_\_\_\_

output:  
html\_document  
pdf\_document:  
de-  
fault  
Within  
the  
open  
source  
com-  
mu-  
nity,  
sev-  
eral  
toolk-  
its  
al-  
low  
us  
to  
ex-  
tract  
data  
that  
can  
be  
used  
to  
ex-  
plore  
projects:  
Metrics  
Gri-  
moire  
http:  
//  
metr  
icsg  
rimo  
ire.  
gith  
ub.i  
o/  


\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
SonarQube  
http://  
ww  
w.so  
narq  
ube.  
org/  
  
CKJM  
(OO  
Met-  
rics  
tool)  
http://gr  
omit  
.iiar.  
pwr.  
wroc  
.pl/  
p\_i  
nf/c  
kj  
m/  
Collects  
a  
large  
num-  
ber  
of  
object-  
oriented  
met-  
rics  
from  
code.  
##  
Is-  
sues

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
There  
are  
prob-  
lems  
such  
as  
dif-  
fer-  
ent  
tools  
re-  
port  
dif-  
fer-  
ent  
val-  
ues  
for  
the  
same  
met-  
ric  
(?)  
It is  
well-  
know  
that  
the  
NASA  
datasets  
have  
some  
prob-  
lems:

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
  \_\_\_\_\_  
  +  
  (?)  
The  
mis-  
use  
  of  
  the  
NASA  
met-  
rics  
data  
pro-  
gram  
data  
sets  
  for  
  au-  
  to-  
mated  
soft-  
ware  
  de-  
  fект  
pre-  
dic-  
tion

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
+  
(?)  
Data  
Qual-  
ity:  
Some  
Com-  
ments  
  on  
  the  
NASA  
Soft-  
ware  
  De-  
  fect  
Datasets

##  
Ef-  
fort  
Esti-  
ma-  
tion  
Data  
  in  
Soft-  
ware  
  En-  
  gi-  
neer-  
  ing

output:

html\_document

pdf\_document:

de-  
fault

It is  
worth  
high-  
light-  
ing  
the  
case  
of  
soft-  
ware

ef-  
fort  
esti-  
ma-  
tion  
datasets  
with  
their

pe-  
cu-  
liari-  
ties.

First,  
most  
ef-  
fort  
esti-  
ma-  
tion  
datasets  
used  
in  
the  
liter-

a-  
ture  
are  
scat-  
tered  
through

re-  
search

pa-  
pers

with

the

ex-

cep-

tion

of a

f...

output:  
html\_document  
pdf\_document:  
de-  
fault

---

Second,  
their  
size  
is  
very  
small  
with  
the  
ex-  
cep-  
tion  
of  
IS-  
BSG  
repos-  
itory  
dis-  
cussed  
pre-  
vi-  
ously  
which  
a  
small  
sam-  
ple  
is  
avail-  
able  
through  
PROMISE  
and  
the  
China  
dataset  
with  
499  
in-  
stances.

output:

html\_document

pdf\_document:

de-

fault

Third,

some

can

be

quite

old

in a

con-

text

and

time

that

is

not

ap-

pli-

ca-

ble

to

cur-

rent

de-

vel-

op-

ment

envi-

ron-

ments.

The

au-

thors

noted

that

the

old-

est

datasets

(CO-

COMO,

De-

shar-

nais,

Ke-

merer

and

Al-

brecht

and

Gaffney)

tend

to

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault

However,

soft-  
ware

ef-  
fort  
and  
cost  
esti-  
ma-  
tion  
still

re-  
main  
one  
of  
the  
main  
chal-  
lenges

in  
soft-  
ware  
engi-  
neer-  
ing  
and  
have

at-

tracted

a  
great  
deal  
of  
in-  
ter-  
est  
by

many  
re-  
searchers

(?).

For  
ex-  
am-  
ple,  
there  
are  
con-  
tinu-  
ous  
and

output:  
html\_document  
pdf\_document:  
de-  
fault  
Next  
Ta-  
ble  
??  
(fol-  
low-  
ing  
Mair  
et al  
(?) )  
shows  
the  
most  
open  
cost/effort  
datasets  
avail-  
able  
in  
the  
liter-  
a-  
ture  
with  
their  
main  
ref-  
er-  
ence.  
Table:  
Ef-  
fort  
Esti-  
ma-  
tion  
Dataset  
from  
arti-  
cles



\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
[<sup>1</sup>]:  
Do-  
nated  
through  
PROMISE.  
[<sup>2</sup>]:  
Only  
a  
sub-  
set  
of  
the  
data  
in  
the  
pa-  
per,  
the  
com-  
plete  
dataset  
is  
do-  
nated  
through  
PROMISE

#  
(PART)  
Ex-  
ploratory  
and  
De-  
scrip-  
tive  
Data  
anal-  
ysis  
{-}

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
#  
Ex-  
ploratory  
Data  
Anal-  
ysis  
##  
De-  
scrip-  
tive  
statis-  
tics  
The  
first  
task  
to  
do  
with  
any  
dataset  
is to  
char-  
acter-  
ize  
it in  
terms  
of  
sum-  
mary  
statis-  
tics  
and  
graph-  
ics.

\_\_\_\_\_

output:

html\_document

pdf\_document:

de-

fault

\_\_\_\_\_

Displaying

in-

for-

ma-

tion

graph-

i-

cally

will

help

us

to

iden-

tify

the

main

char-

ac-

teris-

tics

of

the

data.

To

de-

scribe

a

dis-

tri-

bu-

tion

we

of-

ten

want

to

know

where

it is

cen-

tered

and

and

what

the

spread

is

(mean,

me-

dian,

median,

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
##  
Ba-  
sic  
Plots  
\*  
*His-*  
*togram*  
  de-  
  fines  
  a se-  
  quence  
    of  
  breaks  
  and  
  then  
  counts  
    the  
  num-  
    ber  
    of  
  ob-  
  ser-  
    va-  
  tions  
    in  
    the  
  bins  
  formed  
    by  
    the  
  breaks.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_\*

*Box-*  
*plot*  
used  
  to  
  sum-  
  ma-  
  rize  
data  
  suc-  
  cinctly,  
  quickly  
  dis-  
  play-  
  ing  
  if  
  the  
data  
  is  
sym-  
met-  
ric  
  or  
  has  
sus-  
pected  
  out-  
  liers.



\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_\*

*Q-Q*  
*plot*  
is  
used  
to  
de-  
ter-  
mine  
if  
the  
data  
is  
close  
to  
be-  
ing  
nor-  
mally  
dis-  
tributed.

The  
quan-  
tiles  
of  
the  
stan-  
dard  
nor-  
mal  
dis-  
tri-  
bu-  
tion  
is  
rep-  
re-  
sented  
by a  
straight  
line.

The  
nor-  
mal-  
ity  
of  
the  
data  
can  
be  
\_\_\_\_\_

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
  
\*  
*Scat-  
ter-  
plot*  
pro-  
vides  
a  
graph-  
ical  
view  
of  
the  
rela-  
tion-  
ship  
be-  
tween  
two  
sets  
of  
num-  
bers:  
one  
nu-  
mer-  
ical  
vari-  
able  
against  
an-  
other.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
\*

*Ker-*  
*nel*  
*Den-*  
*sity*  
plot  
visu-  
al-  
izes  
the  
un-  
der-  
ly-  
ing  
dis-  
tri-  
bu-  
tion  
of a  
vari-  
able.  
Ker-  
nel  
den-  
sity  
esti-  
ma-  
tion  
is a  
non-  
parametric  
method

of  
esti-  
mat-  
ing  
the  
prob-  
abil-  
ity  
den-  
sity  
func-  
tion  
of  
con-  
tinu-  
ous  
ran-  
dom  
uni-

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\* *Vi-*  
*olin*  
*plot*  
is a  
com-  
bina-  
tion  
of a  
box-  
plot  
and  
a  
ker-  
nel  
den-  
sity  
plot.  
##  
Nor-  
mal-  
ity

output:

html\_document

pdf\_document:

de-

fault

—

\* A

nor-

mal

dis-

tri-

bu-

tion

is an

ar-

range-

ment

of a

data

set

in

which

most

val-

ues

clus-

ter

in

the

mid-

dle

of

the

range

\* A

graph-

ical

rep-

re-

sen-

ta-

tion

of a

nor-

mal

dis-

tri-

bu-

tion

is

sometime

sometimes

called

a

*bell*

*curve*

*h*

```

output:
html_document
pdf_document:
 de-
 fault

r #
Area
within
 2SD
 of
 the
mean
par(mfrow=c(1,2))
plot(function(x)
dnorm(x,
mean
 =
 0,
sd
 =
 1),
xlim
 =
c(-3,
 3),
main
 =
"SD
 1",
xlab
 =
"x",
ylab
 =
"",
cex
 =
 2)
segments(-2,
 0,
 -2,
0.4)
segments(2,
 0,
 2,
0.4)
 #
Area
within
 4SD
 of
 the
mean
plot(function(x)
dnorm(x,
mean
```

```

output:
html_document
pdf_document:
 de-
 fault

[] []
- if
we
sam-
ple
from
this
pop-
ula-
tion
we
get
“an-
other
pop-
ula-
tion”:
“‘r
#tidy
uses
the
pack-
age
for-
matR
to
for-
mat
the
code
```

```

output:
html_document
pdf_document:
de-
fault

sample.means
<-
rep(NA,
1000)
for
(i in
1:1000)
{
sam-
ple.40
<-
rnorm(40,
mean
=
60,
sd =
4)
#rnorm
gen-
er-
ates
ran-
dom
num-
bers
from
nor-
mal
dis-
tri-
bu-
tion
sam-
ple.means[i]
<-
mean(sample.40)
}
means40
<-
mean(sample.means)
sd40
<-
sd(sample.means)
means40
```

```

```
_____
output:
html_document
pdf_document:
  de-
fault
_____
## [1] 60.0144
r
sd40
## [1] 0.6592934
```

output:
html_document
pdf_document:
de-
fault

-
These
sam-
ple
means
are
an-
other
“pop-
ula-
tion”.
The
sam-
pling
dis-
tri-
bu-
tion
of
the
sam-
ple
mean
is
nor-
mally
dis-
tributed
mean-
ing
that
the
“mean
of a
rep-
re-
sen-
ta-
tive
sam-
ple
pro-
vides
an
esti-
mate
of
the
un-
known

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
r
hist(sample.means)

##  
Us-  
ing  
a  
run-  
ning  
Ex-  
am-  
ple  
to  
visu-  
alise  
the  
dif-  
fer-  
ent  
plots  
As a  
run-  
ning  
ex-  
am-  
ple  
we  
do  
next:  
  1.  
Set  
the  
path  
  to  
  to  
the  
file
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  2.
Read
  the
  Tele-
com1
dataset
  and
  print
  out
  the
  sum-
  mary
  statis-
  tics
  with
  the
  com-
  mand
  summary
    r
options(digits=3)
telecom1
  <-
read.table("./datasets/effortEstimation/Telecom1.csv",
sep=",",header=TRUE,
stringsAsFactors=FALSE,
  dec
  =
  ".")
#read
data
summary(telecom1)
```

output:

html_document

pdf_document:

de-

fault

##

size

effort

EstTotal

##

Min.

:

3.0

Min.

:

24

Min.

:

30

##

1st

Qu.:

37.2

1st

Qu.:

119

1st

Qu.:142

##

Median

:

68.5

Median

:

222

Median

:289

##

Mean

:100.3

Mean

:

284

Mean

:320

##

3rd

Qu.:164.0

3rd

Qu.:

352

3rd

Qu.:472

##

Max.

:884.0

output:
html_document
pdf_document:
de-
fault
_____*

We
see
that
this
dataset
has
three
vari-
ables
(or
pa-
ram-
e-
ters)
and
few
data
points
(18)
+

size:
the
inde-
pen-
dent
vari-
able
+ *ef-*
fort:

the
de-
pen-
dent
vari-
able
+

Est-
To-
tal:
the
esti-
mates
com-
ing
from
an
esti-
ma-
tion
method

```



---


output:
html_document
pdf_document:
de-
fault


---


“‘r
par(mfrow=c(1,2))
#n
fig-
ures
per
row
size_telecom1
<-
telecom1sizeefforttelecom1 <
-telecom1effort
hist(size_telecom1,
col="blue",
xlab='size',
ylab
=
'Prob-
abil-
ity',
main
=
'His-
togram
of
project
Size')
lines(density(size_telecom1,
na.rm
= T,
from
= 0,
to =
max(size_telecom1)))
plot(density(size_telecom1))
```

r
hist(effort_telecom1,
col="blue")
plot(density(effort_telecom1))

```

```

output:
html_document
pdf_document:
 de-
 fault

[|] [x]

r
boxplot(size_telecom1)
boxplot(effort_telecom1)
[|] [|]

r #
violin
plots
for
those
two
variables
library(vioplot)
vioplot(size_telecom1,
names
=
'')
title("Violin
Plot
of
Project
Size")
vioplot(effort_telecom1,
names
=
'')
title("Violin
Plot
of
Project
Effort")
[|] [|]
```

```

output:
html_document
pdf_document:
 de-
fault

r
par(mfrow=c(1,1))
qqnorm(size_telecom1,
main="Q-Q
Plot
 of
'size'")
qqline(size_telecom1,
col=2,
lwd=2,
lty=2)
#draws
 a
line
through
the
first
and
third
quartiles

r
qqnorm(effort_telecom1,
main="Q-Q
Plot
 of
'effort'")
qqline(effort_telecom1)

```

---

output:  
html\_document  
pdf\_document:  
  de-  
  fault  
  \*

We  
can  
ob-  
serve  
the  
non-  
normality  
of  
the  
data.  
\*

We  
may  
look  
the  
pos-  
sible  
rela-  
tion-  
ship  
be-  
tween  
size  
and  
ef-  
fort  
with  
a  
scat-  
ter  
plot  
r  
plot(size\_telecom1,  
effort\_telecom1)



```

output:
html_document
pdf_document:
 de-
 fault

####
Ex-
am-
ple
with
the
China
dataset
r
library(foreign)
china
<-
read.arff("./datasets/effortEstimation/china.arff")
china_size
<-
china$AFP
summary(china_size)
##
Min.
1st
Qu.
Median
Mean
3rd
Qu.
Max.
##
9
100
215
487
438
17518
r
china_effort
<-
china$Effort
summary(china_effort)

```

```

output:
html_document
pdf_document:
 de-
 fault
 ##
 Min.
 1st
 Qu.
 Median
 Mean
 3rd
 Qu.
 Max.
 ##
 26
 704
 1829
 3921
 3826
 54620
 r
 par(mfrow=c(1,2))
 hist(china_size,
 col="blue",
 xlab="Adjusted
Function
Points",
 main="Distribution
of
AFP")
 hist(china_effort,
 col="blue",xlab="Effort",
 main="Distribution
of
Effort")
 L L
 r
 boxplot(china_size)
 boxplot(china_effort)
 I I
```

```

output:
html_document
pdf_document:
 de-
 fault

r
qqnorm(china_size)
qqline(china_size)
qqnorm(china_effort)
qqline(china_effort)
□□
* We
 ob-
 serve
 the
 non-
 normality
 of
 the
 data.
#####
Nor-
mal-
ity.
Gal-
ton
data
```

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
It is  
the  
data  
based  
on  
the  
fa-  
mous  
1885  
Fran-  
cis  
Gal-  
ton's  
study  
about  
the  
rela-  
tion-  
ship  
be-  
tween  
the  
heights  
of  
adult  
chil-  
dren  
and  
the  
heights  
of  
their  
par-  
ents.  
¶¶¶#  
Nor-  
mal-  
iza-  
tion

output:  
html\_document  
pdf\_document:  
de-  
fault

Take  
logs  
in  
both  
inde-  
pen-  
dent  
vari-  
ables.

For  
 ex-  
 am-  
 ple,  
 with  
 the  
*China*  
 dataset.



\* If  
 the  
*log*  
 trans-  
 for-  
 ma-  
 tion  
 is  
 used,  
 then  
 the  
 esti-  
 ma-  
 tion  
 equa-  
 tion  
 is:

$$y = e^{b_0 + b_1 \log(x)}$$

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
##  
Cor-  
rela-  
tion

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault

*Correlation*

is a  
sta-  
tisti-  
cal  
rela-  
tion-  
ship  
be-  
tween  
two  
sets  
of  
data.  
With  
the  
whole  
dataset

we  
may  
check  
for  
the  
lin-  
ear  
Cor-  
rela-  
tion  
of  
the  
vari-  
ables

we  
are  
in-  
ter-  
ested  
in.

```

output:
html_document
pdf_document:
 de-
 fault

 As
 an
 ex-
 am-
 ple
 with
 the
 China
 dataset
 r
 par(mfrow=c(1,1))
 plot(china_size, china_effort)
 +
 r
 cor(china_size, china_effort)
 ##
 [1]
 0.685
 r
 cor.test(china_size, china_effort)
```

```

output:
html_document
pdf_document:
 de-
fault

Pearson's
product-moment
correlation

data:
china_size
and
china_effort

t =
21,
df
=
497,
p-value
<2e-16

alternative
hypothesis:
true
correlation
is
not
equal
to
0

95
percent
confidence
interval:

0.635
0.729

sample
estimates:

cor

0.685
```

```

output:
html_document
pdf_document:
 de-
 fault

r
cor(china_size, china_effort,
method="spearman")

[1]
0.649
r
cor(china_size, china_effort,
method="kendall")

[1]
0.468
```

output:

html\_document

pdf\_document:

de-  
fault

##

Con-  
fi-  
dence

In-  
ter-  
vals.

Boot-  
strap  
\*

Un-  
til  
now  
we  
have  
gen-  
er-  
ated  
point  
esti-  
mates

\* A  
*con-*  
*fi-*  
*dence*  
*in-*  
*ter-*  
*val*  
(CI)

is an  
in-  
ter-  
val  
esti-  
mate  
of a  
pop-  
ula-  
tion  
pa-  
ram-  
eter.

The  
pa-  
ram-  
eter  
can  
be  
the  
mean,

```

output:
html_document
pdf_document:
 de-
 fault

 An
 ex-
 am-
 ple
 from
 Ugarte
 et al.
 (?)
 r
set.seed(10)
norsim(sims
 =
100,
n =
36,
mu
 =
100,
sigma
 =
18,
conf.level
 =
0.95)
```



\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
\*

The  
range  
  de-  
  fined  
    by  
    the  
    con-  
    fi-  
    dence  
    in-  
    ter-  
    val  
    will  
    vary  
    with  
    each  
    sam-  
    ple,  
    be-  
    cause  
    the  
    sam-  
    ple  
    size  
    will  
    vary  
    each  
    time  
    and  
    the  
    stan-  
    dard  
    devi-

  a-  
  tion  
  will  
  vary  
  too.  
  \*

95%  
con-  
  fi-  
  dence  
    in-  
    ter-  
    val:  
    it is  
    the  
    prob-  
    abil-

output:  
html\_document  
pdf\_document:  
de-  
fault  
##  
Non-  
para-  
met-  
ric  
Boot-  
strap  
\*  
For  
com-  
put-  
ing  
CIs  
the  
im-  
por-  
tant  
thing  
is to  
know  
the  
as-  
sump-  
tions  
that  
are  
made  
to  
“know”  
the  
dis-  
tri-  
bu-  
tion  
of  
the  
statis-  
tic.  
\*

There  
is a  
way  
to  
com-  
pute  
con-  
fi-  
dence  
in-  
ter-  
---1-

output:  
html\_document  
pdf\_document:  
de-  
fault



- An  
ex-  
am-  
ple  
of  
boot-  
strap  
CI  
can  
be  
found  
in  
Chap-  
ter  
??,  
“Eval-  
ua-  
tion  
in  
Soft-  
ware  
En-  
gi-  
neer-  
ing”

#  
Classical  
Hy-  
poth-  
esis  
Test-  
ing

\_\_\_\_\_

output:

html\_document

pdf\_document:

de-

fault

\_\_\_\_\_

- By

“clas-

si-

cal”

we

mean

the

stan-

dard

“fre-

quen-

tist”

ap-

proach

to

hy-

poth-

esis

test-

ing.

The

“fre-

quen-

tist”

ap-

proach

to

prob-

abil-

ity

sees

it as

the

fre-

quency

of

events

in

the

long

run.

We

re-

peat

ex-

peri-

ments

over

and

over

and

output:  
html\_document  
pdf\_document:  
de-  
fault

-  
The  
clas-  
sical  
ap-  
proach

is  
usu-  
ally  
called  
**null**  
**hy-**  
**poth-**  
**esis**  
**sig-**  
**nifi-**  
**cance**  
**test-**

**ing**  
(NHST)

be-  
cause  
the  
pro-  
cess  
starts

by  
set-  
ting  
a  
null  
**hy-**  
**poth-**  
**esis**  
 $H_0$

which  
is  
the  
op-  
po-  
site  
about  
what  
we  
think  
is  
true.

---

output:  
html\_document  
pdf\_document:  
  de-  
  fault

---

The  
ra-  
tio-  
nale  
  of  
  the  
pro-  
cess  
  is  
that  
  the  
sta-  
tisti-  
  cal  
  hy-  
poth-  
  esis  
should  
  be  
*falsi-*  
  *fi-*  
  *able*,  
that  
  is,  
we  
can  
find  
evi-  
dence  
that  
  the  
  hy-  
poth-  
  esis  
  is  
not  
true.  
We  
try  
to  
find  
evi-  
dence  
against  
  the  
  null  
  hy-  
poth-  
  esis  
  *is*

output:  
html\_document  
pdf\_document:  
de-  
fault

-  
Usu-  
ally,  
the  
null  
hy-  
poth-  
esis  
is de-  
scribed

as  
the  
situ-  
a-  
tion  
of  
“no  
ef-  
fect”  
and  
the  
al-  
ter-  
na-  
tive  
hy-  
poth-  
esis  
de-  
scribes  
the  
ef-  
fect  
that  
we  
are  
look-  
ing  
for.

output:

html\_document

pdf\_document:

de-

fault

- Af-

ter

col-

lect-

ing

data,

tak-

ing

an

ac-

tual

sam-

ple,

we

mea-

sure

the

dis-

tance

of

our

pa-

ram-

eter

of

in-

ter-

est

from

the

hy-

poth-

e-

sized

pop-

ula-

tion

pa-

ram-

eter,

and

use

the

facts

of

the

sam-

pling

dis-

tri-

bu-

output:  
html\_document  
pdf\_document:  
de-  
fault

- If  
the  
prob-  
abil-  
ity  
of  
our  
sam-  
ple,  
given  
the  
null  
hy-  
poth-  
esis  
is  
high,  
this  
pro-  
vides  
evi-  
dence  
that  
the  
null  
hy-  
poth-  
esis  
is  
true.  
Con-  
versely,

if  
the  
prob-  
abil-  
ity  
of  
the  
sam-  
ple  
is  
low  
(given  
the  
hy-  
poth-  
esis),  
this  
is  
--:

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_

-  
The  
goal  
of  
the  
test  
is to  
de-  
ter-  
mine  
if  
the  
null  
hy-  
poth-  
esis  
can  
be  
re-  
jected.

A  
sta-  
tisti-  
cal  
test  
can  
ei-  
ther  
re-  
ject  
or  
fail  
to  
re-  
ject  
a  
null  
hy-  
poth-  
esis,  
but  
never  
prove  
it  
true.

output:  
html\_document  
pdf\_document:  
de-  
fault

---

- We  
can  
make  
two  
types  
of  
er-  
rors:  
false  
posi-  
tive  
(Type  
I)  
and  
false  
neg-  
a-  
tive  
(Type  
II)

Type  
I  
and  
Type  
II er-  
rors



Two-  
tailed  
NHST



One-  
tailed  
NHST



```

output:
html_document
pdf_document:
 de-
 fault

 - ele-
 men-
 tary
 ex-
 am-
 ple
 r
data
 =
c(52.7,
 53.9,
 41.7,
 71.5,
 47.6,
 55.1,
 62.2,
 56.5,
 33.4,
 61.8,
 54.3,
 50.0,
 45.3,
 63.4,
 53.9,
 65.5,
 66.6,
 70.0,
 52.4,
 38.6,
 46.1,
 44.4,
 60.7,
 56.4);
t.test(data,
mu=50,
alternative
 =
 'greater')
```

```

output:
html_document
pdf_document:
 de-
fault

One
Sample
t-test

data:
data

t =
2,
df =
=
23,
p-value
=
0.02

alternative
hypothesis:
true
mean
is
greater
than
50

95
percent
confidence
interval:

50.9
Inf

sample
estimates:

mean
of
x

54.3
```

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_

-  
Keep-  
ing  
this  
sim-  
ple,  
we  
could  
start  
hy-  
poth-  
esis  
test-  
ing  
about  
one  
sam-  
ple  
me-  
dian  
with  
the  
wilcoxon  
test  
for  
non-  
normal  
dis-  
tri-  
bu-  
tions.

```

output:
html_document
pdf_document:
 de-
fault

-
“ae”
is
the
ab-
so-
lute
er-
ror
in
the
China
Test
data
r
median(ae)
[1] 867
r
mean(ae)
[1] 1867
r
wilcox.test(ae,
mu=800,
alternative
=
'greater')
#change
the
values
of
mu
and
see
the
results
```

```

output:
html_document
pdf_document:
 de-
 fault

 ##
 ##
Wilcoxon
signed
rank
test
with
continuity
correction
 ##
 ##
data:
 ae
 ##
 V =
8990,
p-value
 =
8e-04
 ##
alternative
hypothesis:
true
location
 is
greater
than
800
```

output:  
html\_document  
pdf\_document:  
de-  
fault

-  
Quick  
in-  
tro-  
duc-  
tion  
at  
http  
s://  
psyc  
hsta  
ttwo  
rksh  
op.w  
ordp  
ress  
.com  
/201  
4/08  
/06/  
less  
on-  
9-hy  
poth  
esis-  
test  
ing/

output:  
html\_document  
pdf\_document:  
de-  
fault  
##  
p-  
values  
- p-  
value:  
the  
p-  
value  
of a  
sta-  
tisti-  
cal  
test  
is  
the  
prob-  
abil-  
ity,  
com-  
puted  
as-  
sum-  
ing  
that  
 $H_0$   
is  
true,  
that  
the  
test  
statis-  
tic  
would  
take  
a  
value  
as  
ex-  
treme  
or  
more  
ex-  
treme  
than  
that  
actu-  
ally  
ob-  
served.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_;

#  
(PART)  
Pre-  
pro-  
cess-  
ing  
{-}  
#  
Pre-  
pro-  
cess-  
ing

output:  
html\_document  
pdf\_document:  
  de-  
  fault

---

Following  
the  
data  
min-  
ing  
pro-  
cess,  
we  
de-  
scribe  
what  
  is  
meant  
  by  
pre-  
pro-  
cess-  
ing,  
clas-  
sical  
  su-  
  per-  
  vised  
mod-  
  els,  
  un-  
  su-  
  per-  
  vised  
mod-  
  els  
and  
eval-  
  ua-  
  tion  
  in  
the  
con-  
text  
  of  
soft-  
ware  
engi-  
neer-  
  ing  
with  
  ex-  
  am-  
  ples

output:

html\_document

pdf\_document:

de-

fault

This

task

is

prob-

ably

the

hard-

est

and

where

most

of ef-

fort

is

spend

in

the

data

min-

ing

pro-

cess.

It is

quite

typi-

cal

to

trans-

form

the

data,

for

ex-

am-

ple,

find-

ing

in-

con-

sis-

ten-

cies,

nor-

mal-

is-

ing,

im-

put-

ing

miss-

ing

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
Typically,  
pre-  
processing  
con-  
sist  
of  
the  
fol-  
low-  
ing  
tasks  
(sub-  
pro-  
cesses):

\_\_\_\_\_  
output:

html\_document

pdf\_document:

de-

fault

\_\_\_\_\_

+

Data

clean-

ing

(con-

sis-

tency,

noise

de-

tec-

tion,

out-

liers)

\_\_\_\_\_

+

Data

inte-

gra-

tion

\_\_\_\_\_

+

Data

trans-

for-

ma-

tion

(nor-

maliza-

sation,

dis-

creti-

sation)

and

deriva-

tion

of

new

at-

tributes

from

ex-

ist-

ing

ones

(e.g.,

pop-

ula-

tion

den-

sity

from

```

|output:
|html_document
|pdf_document:
| de-
| fault
|_____
|## Data
```

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault

\_\_\_\_\_  
*Consistent*

data  
are  
se-  
man-  
ti-  
cally  
cor-  
rect  
based

on  
real-  
world  
knowl-  
edge  
of  
the  
prob-  
lem,  
i.e.,  
no  
con-  
straints

are  
vio-  
lated  
and  
data  
that  
can  
be  
used  
for  
in-  
duc-  
ing  
mod-  
els  
and  
anal-  
ysis.

For  
ex-  
am-  
ple,  
the  
LoC  
or  
ef-  
fort  
is

```

| output:
| html_document
| pdf_document:
| de-
| fault
| ##
| Miss-
| ing
| val-
| ues
```

\_\_\_\_\_  
output:

html\_document

pdf\_document:

de-

fault

\_\_\_\_\_

*Missing*

*val-*

*ues*

*will*

*have*

*a*

*neg-*

*a-*

*tive*

*ef-*

*fect*

*when*

*analysing*

*the*

*data*

*or*

*learn-*

*ing*

*mod-*

*els.*

*The*

*re-*

*sults*

*can*

*be*

*bi-*

*ased*

*when*

*com-*

*pared*

*with*

*the*

*mod-*

*els*

*in-*

*duced*

*from*

*the*

*com-*

*plete*

*data,*

*the*

*re-*

*sults*

*can*

*be*

*harder*

*to*

*anal-*

*yse,*

*z*

\_\_\_\_\_

output:

html\_document

pdf\_document:

de-

fault

\_\_\_\_\_

Missing

data

is

typi-

cally

clas-

si-

fied

into:

\*

MCAR

(Miss-

ing

Com-

pletely

at

Ran-

dom)

or

MAR

(Miss-

ing

At

Ran-

dom)

where

there

is no

rea-

son

for

those

miss-

ing

val-

ues

and

we

can

as-

sume

that

the

dis-

tri-

bu-

tion

could

fol-

low

the

output:  
html\_document  
pdf\_document:  
de-  
fault

---

*Imputation*

con-  
sists  
in  
re-  
plac-  
ing  
miss-  
ing  
val-  
ues  
for  
esti-  
mates  
of  
those  
miss-  
ing  
val-  
ues.

Many  
algo-  
rithms  
do  
can-  
not  
han-  
dle  
miss-  
ing  
val-  
ues  
and  
there-  
fore,  
im-  
pu-  
ta-  
tion  
meth-  
ods  
are  
needed.

We  
can  
use  
sim-  
ple  
ap-  
proaches

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
\*  
EM  
(Expectation-  
Maximisation)  
\*  
Distance-  
based  
+  $k$ -  
NN  
( $k$ -  
Nearest  
Neigh-  
bours)  
+  
Clus-  
ter-  
ing

output:

html\_document

pdf\_document:

de-

fault

—

In

R, a

miss-

ing

value

is

rep-

re-

sented

with

NA

and

the

ana-

lyst

must

de-

cide

what

to

do

with

miss-

ing

data.

The

sim-

plest

ap-

proach

is to

leave

out

in-

stances

(ig-

nore

miss-

ing

-IM-

)

with

with

miss-

ing

data.

This

func-

tion-

ality

is

output:  
html\_document  
pdf\_document:  
  de-  
  fault  
The  
**mice**  
  R  
  pack-  
  age.  
MICE  
(Mul-  
  ti-  
  vari-  
  ate  
  Im-  
  pu-  
  ta-  
  tion  
  via  
Chained  
Equa-  
tions)  
  as-  
sumes  
  that  
  data  
  are  
  miss-  
  ing  
  at  
  ran-  
  dom.  
Other  
  pack-  
  ages  
  in-  
  clude  
  **Amelia**,  
  **missForest**,  
  **Hmisc**  
  and  
  **mi**.  
##  
Noise

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault

\_\_\_\_\_  
Imperfections

of  
the  
real-  
world  
data  
that  
in-  
flu-  
ences  
neg-  
a-  
tively  
in  
the  
in-  
duced  
ma-  
chine  
learn-  
ing  
mod-  
els.  
Ap-  
proaches

to  
deal  
with  
noisy  
data  
in-  
clude:  
\* Ro-  
bust  
learn-

ers  
ca-  
pa-  
ble  
of  
han-  
dling  
noisy  
data  
(e.g.,  
C4.5  
through  
prun-  
ing  
strate-  
gies)

\_\_\_\_\_

output:

html\_document

pdf\_document:

de-

fault

\_\_\_\_\_

Types

of

noise

data:

\*

Class

Noise

(aka

la-

bel

noise).

+

There

can

be

con-

tra-

dic-

tory

cases

(all

at-

tributes

have

the

same

value

ex-

cept

the

class)

+

Mis-

clas-

sifi-

ca-

tions.

The

class

at-

tribute

is

not

la-

beled

with

the

true

la-

bel

(golden)

output:  
html\_document  
pdf\_document:  
de-  
fault  
##  
Out-  
liers  
There  
is a  
large  
amount  
of  
liter-  
ature  
re-  
lated  
to  
out-  
lier  
de-  
tec-  
tion,  
and  
fur-  
ther-  
more  
sev-  
eral  
defi-  
ni-  
tions  
of  
out-  
lier  
ex-  
ist.  
“r  
li-  
brary(DMwR2)  
li-  
brary(foreign)

```

output:
html_document
pdf_document:
de-
fault

kc1
<-
read.arff("./datasets/defectPred/D1/KC1.arff")
```
The
LOF
algo-
rithm
(lofactor),
given
a
data
set
it
pro-
duces
a
vec-
tor
of
local
out-
lier
fac-
tors
for
each
case.
r
kc1num
<-
kc1[,1:21]
outlier.scores
<-
lofactor(kc1num,
k=5)
plot(density(na.omit(outlier.scores)))

```

```

_____
output:
html_document
pdf_document:
  de-
  fault
_____
r
outliers
<-
order(outlier.scores,
decreasing=T) [1:5]
print(outliers)
  ##
[1]
  1
  6
 14
 31
 33
Another
sim-
ple
method
  of
Hiri-
doglou
  and
Berth-
elot
  for
posit-
ive
ob-
ser-
va-
tions.
  ##
Feature
  selec-
tion

```

output:
html_document
pdf_document:
 de-
 fault

Feature
 Se-
 lec-
 tion
 (FS)
aims
 at
iden-
tify-
ing
the
most
rele-
vant
 at-
tributes
from
 a
dataset.
It is
 im-
por-
tant
 in
dif-
fer-
ent
ways:

output:
html_document
pdf_document:
de-
fault
* A
re-
duced
vol-
ume
of
data
al-
lows
dif-
fer-
ent
data
min-
ing
or
search-
ing
tech-
niques
to
be
ap-
plied.

output:
html_document
pdf_document:
de-
fault
* Ir-
rele-
vant
and
re-
dundant
at-
tributes
can
gen-
er-
ate
less
ac-
cu-
rate
and
more
com-
plex
mod-
els.
Fur-
ther-
more,
data
min-
ing
algo-
rithms
can
be
exe-
cuted
faster.

output:
html_document
pdf_document:
de-
fault
* It
avoids
the
col-
lec-
tion
of
data
for
those
irrel-
e-
vant
and
re-
dundant
at-
tributes
in
the
fu-
ture.

output:

html_document

pdf_document:

de-

fault

—

The

prob-

lem

of

FS

re-

ceived

a

thor-

ough

treat-

ment

in

pat-

tern

recog-

ni-

tion

and

ma-

chine

learn-

ing.

Most

of

the

FS

algo-

rithms

tackle

the

task

as a

search

prob-

lem,

where

each

state

in

the

search

spec-

ifies

a

dis-

tinct

sub-

set

of

the

output:
html_document
pdf_document:
de-
fault

There
are
two
ma-
jor
ap-
proaches
in
FS
from
the
method's
out-
put
point
of
view:
*
Fea-
ture
sub-
set
se-
lec-
tion
(FSS)

output:
html_document
pdf_document:
 de-
 fault
 *
Fea-
ture
rank-
ing
 in
which
 at-
tributes
 are
ranked
as a
list
 of
features
which
 are
 or-
dered
 ac-
cord-
 ing
 to
eval-
 ua-
tion
mea-
sures
 (a
sub-
 set
 of
fea-
tures
is of-
ten
 se-
lected
from
 the
top
 of
the
rank-
 ing
list).

output:
html_document
pdf_document:
de-
fault
FFS
algo-
rithms
de-
signed
with
dif-
fer-
ent
eval-
ua-
tion
cri-
teria
broadly
fall
into
two
cate-
gories:

output:
html_document
pdf_document:
 de-
 fault

*
The
filter
model
 re-
 lies
 on
 gen-
 eral
 char-
 ac-
 teris-
 tics
 of
 the
 data
 to
 eval-
 uate
 and
 se-
 lect
 fea-
 ture
 sub-
 sets
 with-
 out
 in-
 volv-
 ing
 any
 data
 min-
 ing
 algo-
 rithm.

output:
html_document
pdf_document:
de-
fault

*

The
wrap-
per
model
re-
quires
one
pre-
de-
ter-
mined
min-
ing
algo-
rithm
and
uses
its
per-
for-
mance
as
the
eval-
ua-
tion
cri-
te-
rion.

It
searches
for
fea-
tures
bet-
ter
suited
to
the
min-
ing
algo-
rithm
aim-
ing
to
im-
prove
min-
ing

output:
html_document
pdf_document:
de-
fault
Feature
sub-
set
algo-
rithms
search
through
can-
di-
date
fea-
ture
sub-
sets
guide
by a
cer-
tain
eval-
ua-
tion
mea-
sure
(?)
which
cap-
tures
the
good-
ness
of
each
sub-
set.
An
opti-
mal
(or
near
opti-
mal)
sub-
set
is se-
lected
when
the
search
stops.

output:
html_document
pdf_document:
de-
fault

Some
ex-
ist-
ing
eval-
ua-
tion
mea-
sures
that
have
been
shown
ef-
fec-
tive
in
re-
mov-
ing
both
irrel-

e-
vant
and
re-
dundant
fea-
tures
in-
clude
the
con-
sis-

tency
mea-
sure
(?),
the
cor-
rela-
tion
mea-
sure
(?)
and
the
esti-
mated

output:

html_document

pdf_document:

de-

fault

—

+

Con-

sis-

tency

mea-

sure

at-

tempts

to

find

a

min-

i-

mum

num-

ber

of

fea-

tures

that

sepa-

rate

classes

as

con-

sis-

tently

as

the

full

set

of

fea-

tures

can.

An

in-

con-

sis-

tency

is de-

fined

as

to

in-

stances

hav-

ing

the

same

for

output:
html_document
pdf_document:
 de-
 fault

+
Cor-
rela-
tion
mea-
sure
eval-
 u-
 ates
 the
 good-
 ness
 of
 fea-
 ture
 sub-
 sets
 based
 on
 the
 hy-
 poth-
 esis
 that
 good
 fea-
 ture
 sub-
 sets
 con-
 tain
 fea-
 tures
 highly
 cor-
 re-
 lated
 to
 the
 class,
 yet
 un-
 cor-
 re-
 lated
 to
 each
 other.

output:

html_document

pdf_document:

de-

fault

—

+

Wrapper-

based

at-

tribute

se-

lec-

tion

uses

the

tar-

get

learn-

ing

algo-

rithm

to

esti-

mate

the

worth

of

at-

tribute

sub-

sets.

The

fea-

ture

sub-

set

se-

lec-

tion

algo-

rithm

con-

ducts

a

search

for a

good

sub-

set

us-

ing

the

in-

duc-

tion

al-

output:

html_document

pdf_document:

de-

fault

Langley

(?)

notes

that

fea-

ture

se-

lec-

tion

algo-

rithms

that

search

through

the

space

of

fea-

ture

sub-

sets

must

ad-

dress

four

main

is-

sues:

(i)

the

start-

ing

point

of

the

search,

(ii)

the

orga-

niza-

tion

of

the

search,

(iii)

the

eval-

ua-

tion

of

f...

output:

html_document

pdf_document:

de-
fault

It is
im-

prac-
tical

to
look

at
all

pos-
sible

fea-
ture

sub-
sets,

even
with

a
small

num-
ber

of
at-

tributes.

Fea-
ture

se-
lec-

tion
algo-

rithms
usu-

ally
pro-

ceed
greed-

ily
and

are
be

clas-
si-

fied
into

those
that

add
fea-

tures
to

an

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
####
FSe-
  lec-
  tor
pack-
  age
in R
The
FSe-
  lec-
  tor
pack-
  age
in R
  im-
  ple-
  ments
many
algo-
rithms
avail-
able
  in
Weka
  “‘r
  li-
brary(FSelector)
  li-
brary(foreign)
cm1
  <-
read.arff("./datasets/defectPred/D1/CM1.arff")
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
cm1RFWeights
  <-
  ran-
  dom.forest.importance(Defective
    ~.,
    cm1)
    cut-
    off.biggest.diff(cm1RFWeights)
    ""
  ##
  [1]
"LOC_COMMENTS"
"NUM_UNIQUE_OPERATORS"
Using
  the
  In-
  for-
  ma-
  tion
  Gain
  mea-
  sure
  as
rank-
ing:
  r
cm1GRWeights
  <-
  gain.ratio(Defective
    ~
    .,
    cm1)
cm1GRWeights
```

output:
 html_document
 pdf_document:
 de-
 fault

 ##
 attr_importance
 ##
 LOC_BLANK
 0.0000
 ##
 BRANCH_COUNT
 0.0000
 ##
 CALL_PAIRS
 0.0000
 ##
 LOC_CODE_AND_COMMENT
 0.0000
 ##
 LOC_COMMENTS
 0.0754
 ##
 CONDITION_COUNT
 0.0000
 ##
 CYCLOMATIC_COMPLEXITY
 0.0000
 ##
 CYCLOMATIC_DENSITY
 0.0000
 ##
 DECISION_COUNT
 0.0000
 ##
 DECISION_DENSITY
 0.0000
 ##
 DESIGN_COMPLEXITY
 0.0000
 ##
 DESIGN_DENSITY
 0.0000
 ##
 EDGE_COUNT
 0.0000
 ##
 ESSENTIAL_COMPLEXITY
 0.0000
 ##
 ESSENTIAL_DENSITY
 0.0000
 ##
 LOC_EXECUTABLE
 0.0888
 ##

```
_____
output:
html_document
pdf_document:
de-
fault
_____
r
cutoff.biggest.diff(cm1GRWeights)
##
[1]
"LOC_EXECUTABLE"
"LOC_TOTAL"
"LOC_COMMENTS"
##
[4]
"HALSTEAD_CONTENT"
"PERCENT_COMMENTS"
"NUM_UNIQUE_OPERATORS"
##
[7]
"NUM_UNIQUE_OPERANDS"
"NUMBER_OF_LINES"
"HALSTEAD_VOLUME"
##
[10]
"NUM_OPERATORS"
"HALSTEAD_ERROR_EST"
"HALSTEAD_LENGTH"
##
[13]
"HALSTEAD EFFORT"
"HALSTEAD_PROG_TIME"
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
r #
After
assigning
weights,
we
can
select
the
statisticaclly
significant
ones
cm1X2Weights
<-
chi.squared(Defective
~
.,
cm1)
cutoff.biggest.diff(cm1X2Weights)
```

```
_____
output:
html_document
pdf_document:
de-
fault
_____
## [1] "LOC_EXECUTABLE"
"LOC_COMMENTS"
"LOC_TOTAL"
## [4] "NUM_UNIQUE_OPERATORS"
"NUM_UNIQUE_OPERANDS"
"NUMBER_OF_LINES"
## [7] "HALSTEAD_VOLUME"
"NUM_OPERATORS"
"HALSTEAD_ERROR_EST"
## [10] "HALSTEAD_CONTENT"
"HALSTEAD EFFORT"
"HALSTEAD_PROG_TIME"
## [13] "HALSTEAD_LENGTH"
"PERCENT COMMENTS"
Using
CFS
at-
tribute
se-
lec-
tion
``r
li-
brary(FSelector)
li-
brary(foreign)
cm1
<-
read.arff("./datasets/defectPred/D1/CM1.arff")
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
result
<-
cfs(Defective
  ~.,
cm1)
f <-
as.simple.formula(result,
  "De-
  fec-
  tive")
f ""
##  

Defective
~  

LOC_COMMENTS
+
LOC_EXECUTABLE
+
HALSTEAD_CONTENT
+
##  

NUM_UNIQUE_OPERATORS
+
PERCENT_COMMENTS
##
<environment:
0x55d7e72edbe0>
```

output:
html_document
pdf_document:
de-
fault
Other
pack-
ages
for
Fea-
ture
se-
lec-
tion
in R
in-
clude
FSelectorRccp
which
re-
implments
the
FSlec-
tor
with-
out
WEKA
de-
pen-
den-
cies.

output:
html_document
pdf_document:
 de-
 fault

Another
pop-
ular
pack-
age
 is
Boruta,
which
 is
based
 on
 se-
 lec-
 tion
based
 on
Random
For-
est.

 In-
stance
 se-
 lec-
 tion

output:
html_document
pdf_document:
 de-
 fault

Removal
 of
 sam-
 ples
(com-
 ple-
 men-
 tary
 to
 the
 re-
 moval
 of
 at-
 tributes)
 in
 or-
 der
 to
 scale
 down
 the
 dataset
prior
 to
learning
 a
model
 so
 that
 there
 is
 (al-
 most)
 no
per-
 for-
mance
loss.

output:

html_document

pdf_document:

de-

fault

There

are

two

types

of

pro-

cesses:

*

Pro-

to-

type

Se-

lec-

tion

(PS)

(?)

when

the

sub-

set

is

used

with

a

dis-

tance

based

method

(kNN)

output:
html_document
pdf_document:
de-
fault
_____*

Train-
ing
Set
Se-
lec-
tion
(TSS)
(?)
in
which
an
ac-
tual
model
is
learned.

output:
html_document
pdf_document:
de-
fault

It is
also
a
search
prob-
lem
as
with
fea-
ture
se-
lec-
tion.
Gar-
cia
et al.
(?)
pro-
vide
a
com-
pre-
hen-
sive
overview
of
the
topic.

Dis-
cretiza-
tion

output:
html_document
pdf_document:
de-
fault
This
pro-
cess
trans-
forms
con-
tinu-
ous
at-
tributes
into
dis-
crete
ones,
by
asso-
ciat-
ing
cate-
gori-
cal
val-
ues
to
in-
ter-
vals
and
thus
trans-
form-
ing
quan-
tita-
tive
data
into
qual-
ita-
tive
data.

output:
html_document
pdf_document:
 de-
 fault

Cor-
rela-
tion
Co-
effi-
cient
and
Co-
vari-
ance
for
Nu-
meric
Data

output:
html_document
pdf_document:

Two
random
variables
 x
and
 y
are
called
inde-
pen-
dent
if
the
prob-
abil-
ity
dis-
tri-
bu-
tion
of
one
vari-
able
is
not
af-
fected
by
the
pres-
ence
of
an-
other.
 $\tilde{\chi}^2 = \frac{1}{d} \sum_{k=1}^n \frac{(O_k - E_k)^2}{E_k}$

```
_____
output:
html_document
pdf_document:
  de-
fault
_____
r
chisq.test(kc1$LOC_BLANK,kc1$BRANCH_TOTAL)
  ##
  ##
Chi-squared
test
  for
given
probabilities
  ##
  ##
data:
kc1$LOC_BLANK
  ##
X-squared
  =
17705,
df
  =
2095,
p-value
<2e-16
  r
chisq.test(kc1$DESIGN_COMPLEXITY,kc1$CYCLOMATIC_COMPLEXITY)
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
    ##
    ##
Pearson's
Chi-squared
test
  ##
  ##
data:
kc1$DESIGN_COMPLEXITY
  and
kc1$CYCLOMATIC_COMPLEXITY
  ##
X-squared
  =
25101,
  df
  =
696,
p-value
<2e-16
  ##
Nor-
mal-
iza-
tion
####
Min-
Max
Nor-
mal-
iza-
tion

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
r
library(caret)
preObj
<-
preProcess(kc1[, -22],
method=c("center",
"scale"))
#####
Z-
score
nor-
mal-
iza-
tion
TBD
##
Trans-
for-
ma-
tions
#####
Lin-
ear
Trans-
for-
ma-
tions
and
Quadratic
Trans
for-
ma-
tions
TBD
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
#####
Box-
  cox
trans-
  for-
  ma-
  tion
TBD
#####
Nom-
  inal
  to
  Bi-
  nary
trans-
  for-
  ma-
  tions
TBD
  ##
  Pre-
  pro-
  cess-
  ing
  in R
#####
The
dplyr
pack-
  age
```

output:
html_document
pdf_document:
de-
fault

The
dplyr
pack-
age
cre-
ated
by
Hadley
Wick-
ham.
Some
func-
tions
are
simi-
lar
to
SQL
syn-
tax.
key
func-
tions
in
dplyr
in-
clude:

output:
html_document
pdf_document:
 de-
 fault
 +
 se-
lect:
 se-
lect
columns
from
 a
dataframe
+ fil-
ter:
 se-
lect
rows
from
 a
dataframe
+
sum-
ma-
rize:
 al-
lows
 us
 to
 do
sum-
mary
stats
based
upon
 the
grouped
vari-
able
+
group_by:
group
by a
fac-
tor
vari-
able
+ ar-
range:
 or-
der
 the
dataset
+
join-

output:
html_document
pdf_document:
de-
fault

Tutorial:

http

s:

//gi

thub

.com

/jus

tm

arkh

am

/dpl

yr-

tuto

rial

Examples

r

library(dplyr)

Describe

the

dataframe:

r

str(kc1)

```
_____
output:
html_document
pdf_document:
  de-
  fault
  ##
'data.frame':
2096
obs.
  of
  22
variables:
  ##
  $
LOC_BLANK
  :
num
0 0
0 0
2 0
0 0
0 2
...
  ##
  $
BRANCH_COUNT
  :
num
1 1
1 1
1 1
1 1
1 1
1 1
...
  ##
  $
LOC_CODE_AND_COMMENT
  :
num
0 0
0 0
0 0
0 0
0 0
...
  ##
  $
LOC_COMMENTS
  :
num
0 0
0 0
0 0
0 0
0 0
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
tbl_df
cre-
ates
  a
  “lo-
  cal
data
frame”
as a
wrap-
per
for
bet-
ter
print-
ing
r
kc1_tbl
<-
tbl_df(kc1)
#deprecated
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  ##

Warning:
`tbl_df()` 
  was
deprecated
  in
dplyr
1.0.0.
  ##

Please
  use
`tibble::as_tibble()`
instead.
  ##

This
warning
  is
displayed
once
every
  8
hours.
  ##

Call
`lifecycle::last_lifecycle_warnings()`
  to
  see
where
this
warning
  was
generated.

  r
kc1_tbl
  <-
tibble(kc1)
Filter:
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
r #
Filter
rows:
  use
  comma
  or
  &
  to
represent
AND
condition
filter(kc1_tbl,
Defective
  ==
  "Y"
  &
LOC_BLANK
  !=
  0)
```

```
_____
output:
html_document
pdf_document:
de-
fault
_____
## 
# A
tibble:
251
  x
  22
  ##
LOC_BLANK
BRANCH_COUNT
LOC_CODE_AND_COMMENT
LOC_COMMENTS
CYCLOMATIC_COMPLEXI-
  ##
<dbl>
<dbl>
<dbl>
<dbl>
<dbl>
  ##
  1
  6
  21
  0
  10
  11
  ##
  2
  5
  15
  0
  2
  8
  ##
  3
  2
  5
  0
  0
  3
  ##
  4
  4
  5
  0
  2
  3
  ##
  5
  2
  11
  0
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
Another
oper-
ator
  is
%in%.
Select:
  r
select(kc1_tbl,
contains("LOC"),
Defective)
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  ##
# A
tibble:
2,096
x 6
##
LOC_BLANK
LOC_CODE_AND_COMME~
LOC_COMMENTS
LOC_EXECUTABLE
LOC_TOTAL
Defective
  ##
<dbl>
<dbl>
<dbl>
<dbl>
<dbl>
<dbl>
<fct>
  ##
  1
  0
  0
  0
  3
  5 N
  ##
  2
  0
  0
  0
  1
  3 N
  ##
  3
  0
  0
  0
  1
  3 N
  ##
  4
  0
  0
  0
  1
  3 N
  ##
  5
  2
  0
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
Now,
kc1_tbl
con-
tains("LOC"),
  De-
  fec-
  tive
Filter
and
Se-
lect
to-
gether:
  r #
nesting
method
filter(select(kc1_tbl,
contains("LOC"),
Defective),
Defective
!=0)
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  ##
# A
tibble:
2,096
x 6
##
LOC_BLANK
LOC_CODE_AND_COMME~
LOC_COMMENTS
LOC_EXECUTABLE
LOC_TOTAL
Defective
  ##
<dbl>
<dbl>
<dbl>
<dbl>
<dbl>
<dbl>
<fct>
  ##
  1
  0
  0
  0
  3
  5 N
  ##
  2
  0
  0
  0
  1
  3 N
  ##
  3
  0
  0
  0
  1
  3 N
  ##
  4
  0
  0
  0
  1
  3 N
  ##
  5
  2
  0
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
It is
eas-
ier
usign
the
chain-
ing
method:
  r #
  chaining
method
kc1_tbl
%>%
select(contains("LOC"),
Defective)
%>%
filter(Defective
!=0)
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  ##
# A
tibble:
2,096
  x 6
  ##
LOC_BLANK
LOC_CODE_AND_COMME~
LOC_COMMENTS
LOC_EXECUTABLE
LOC_TOTAL
Defective
  ##
<dbl>
<dbl>
<dbl>
<dbl>
<dbl>
<dbl>
<fct>
  ##
  1
  0
  0
  0
  3
  5 N
  ##
  2
  0
  0
  0
  1
  3 N
  ##
  3
  0
  0
  0
  1
  3 N
  ##
  4
  0
  0
  0
  1
  3 N
  ##
  5
  2
  0
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
Arrange
  as-
  cend-
    ing
  r #
kc1_tbl
%>%
select(LOC_TOTAL,
Defective)
%>%
arrange(LOC_TOTAL)
```

```
_____
output:
html_document
pdf_document:
de-
fault
_____
## 
# A
tibble:
2,096
x 2
##
LOC_TOTAL
Defective
##
<dbl>
<fct>
##
1
1 N
##
2
1 N
##
3
1 N
##
4
1 N
##
5
1 N
##
6
1 N
##
7
1 N
##
8
1 N
##
9
1 N
##
10
1 N
##
#
...
with
2,086
more
rows
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
Arrange
de-
scend-
ing:
r
kc1_tbl
%>%
select(LOC_TOTAL,
Defective)
%>%
arrange(desc(LOC_TOTAL))
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  ##
# A
tibble:
2,096
  x 2
    ##
LOC_TOTAL
Defective
  ##
<dbl>
<fct>
  ##
  1
288
  Y
  ##
  2
286
  Y
  ##
  3
283
  N
  ##
  4
220
  Y
  ##
  5
217
  Y
  ##
  6
210
  N
  ##
  7
205
  Y
  ##
  8
184
  Y
  ##
  9
179
  Y
  ##
  10
176
  Y
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
Mutate:
  r
  kc1_tbl
  %>%
  filter(Defective
    ==
    "Y")
  %>%
  select(NUM_OPERANDS,
NUM_OPERATORS,
Defective)
  %>%
  mutate(HalsteadLength
    =
    NUM_OPERANDS
    +
    NUM_OPERATORS)
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
    ##
  # A
tibble:
  325
  x 4
  ##
  NUM_OPERANDS
  NUM_OPERATORS
  Defective
  HalsteadLength
    ##
  <dbl>
  <dbl>
  <fct>
  <dbl>
    ##
  1
  64
  107
  Y
  171
  ##
  2
  52
  89
  Y
  141
  ##
  3
  17
  41
  Y
  58
  ##
  4
  41
  74
  Y
  115
  ##
  5
  54
  95
  Y
  149
  ##
  6
  75
  156
  Y
  821
```

```
_____
output:
html_document
pdf_document:
  de-
fault
_____
summarise:
  Re-
duce
vari-
ables
  to
  val-
ues
r #
Create
  a
table
grouped
  by
Defective,
  and
then
summarise
each
group
  by
taking
the
mean
  of
loc
kc1_tbl
%>%
group_by(Defective)
%>%
summarise(avg_loc
  =
mean(LOC_TOTAL,
na.rm=TRUE))
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  ##
# A
tibble:
  2 x
  2
  ##
Defective
avg_loc
  ##
<fct>
<dbl>
  ##
  1 N
15.9
  ##
  2 Y
44.7
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
r #
Create
  a
  table
grouped
  by
Defective,
  and
then
summarise
each
group
  by
taking
the
mean
  of
loc
kc1_tbl
%>%
group_by(Defective)
%>%
summarise_each(funs(mean,
min,
max),
BRANCH_COUNT,
LOC_TOTAL)
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  ##

Warning:
`summarise_each_()` `~
was
deprecated
  in
dplyr
0.7.0.
  ##

Please
use
`across()` `~
instead.
  ##

This
warning
  is
displayed
once
every
  8
hours.
  ##

Call
`lifecycle::last_lifecycle_warnings()`
  to
  see
where
this
warning
  was
generated.
```

output:
 html_document
 pdf_document:
 de-
 fault

 ##
 Warning:
 `fun`(``
 was
 deprecated
 in
 dplyr
 0.8.0.
 ##
 Please
 use
 a
 list
 of
 either
 functions
 or
 lambdas:
 ##
 ##
 #
 Simple
 named
 list:
 ##
 list(mean
 =
 mean,
 median
 =
 median)
 ##
 ##
 #
 Auto
 named
 with
 `tibble::lst()`:
 ##
 tibble::lst(mean,
 median)
 ##
 ##
 #
 Using
 lambdas
 ##
 list(~
 mean(.,
 trim

```
_____
output:
html_document
pdf_document:
de-
fault
_____
## 
# A
tibble:
  2 x
    7
    ##
Defective
BRANCH_COUNT_mean
LOC_TOTAL_mean
BRANCH_COUNT_min
LOC_TOTAL_min
    ##
<fct>
<dbl>
<dbl>
<dbl>
<dbl>
    ##
  1 N
3.68
15.9
    1
    1
    ##
  2 Y
10.1
44.7
    1
    2
    ##
    #
...
with
  2
more
variables:
BRANCH_COUNT_max
<dbl>,
LOC_TOTAL_max
<dbl>
```

output:
html_document
pdf_document:
de-
fault

It
seems
than
the
num-
ber
of
De-
fec-
tive
mod-
ules
is
larger
than
the
Non-
Defective
ones.

We
can
count
them
with:
r #
n()
or
tally
kc1_tbl
%>%
group_by(Defective)
%>%
tally()

```
_____
output:
html_document
pdf_document:
  de-
  fault
  ##
# A
tibble:
  2 x
  2
  ##
Defective
  n
  ##
<fct>
<int>
  ##
  1 N
1771
  ##
  2 Y
  325
  It
seems
that
it's
an
im-
bal-
anced
dataset...
  r #
randomly
sample
  a
fixed
number
  of
rows,
without
replacement
kc1_tbl
%>%
sample_n(2)
```

output:
 html_document
 pdf_document:
 de-
 fault

 ##
 # A
 tibble:
 2 x
 22
 ##
 LOC_BLANK
 BRANCH_COUNT
 LOC_CODE_AND_COMMENT
 LOC_COMMENTS
 CYCLOMATIC_COMPLEXITY
 ##
 <dbl>
 <dbl>
 <dbl>
 <dbl>
 <dbl>
 ##
 1
 0
 3
 0
 0
 2
 ##
 2
 0
 1
 0
 0
 1
 ##
 #
 ...
 with
 17
 more
 variables:
 DESIGN_COMPLEXITY
 <dbl>,
 ##
 #
 ESSENTIAL_COMPLEXITY
 <dbl>,
 LOC_EXECUTABLE
 <dbl>,
 HALSTEAD_CONTENT
 <dbl>,
 ##
 #
 HALSTEAD_DIFFICULTY

```
_____
output:
html_document
pdf_document:
de-
fault
_____
r #
randomly
sample
a
fraction
of
rows,
with
replacement
kc1_tbl
%>%
sample_frac(0.05,
replace=TRUE)
```

```
_____
output:
html_document
pdf_document:
  de-
fault
_____
## 
# A
tibble:
  105
    x
    22
    ##
  LOC_BLANK
  BRANCH_COUNT
  LOC_CODE_AND_COMMENT
  LOC_COMMENTS
  CYCLOMATIC_COMPLEXI-
    ##
  <dbl>
  <dbl>
  <dbl>
  <dbl>
  <dbl>
    ##
    1
    1
    3
    0
    0
    2
    ##
    2
    0
    1
    0
    0
    1
    ##
    3
    0
    1
    0
    0
    1
    ##
    4
    2
    5
    0
    0
    3
    ##
    5
    2
    7
    0
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
r #
Better
formatting
adapted
  to
  the
screen
width
glimpse(kc1_tbl)
```

output:
 html_document
 pdf_document:
 de-
 fault

##

Rows:
 2,096

##

Columns:
 22

##

\$

LOC_BLANK

<dbl>

0,
 0,
 0,
 0,
 2,
 0,
 0,
 0,
 0,
 2,
 2,
 0,
 2,
 1,
 2,
 2,
 ~

##

\$

BRANCH_COUNT

<dbl>

1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 5,
 ~

##

\$

output:
html_document
pdf_document:
de-
fault

Other
li-
braries
and
tricks

output:
html_document
pdf_document:
de-
fault

The
lubridate
pack-
age
con-
tains
a
num-
ber
of
func-
tions
facil-
itat-
ing
the
con-
ver-
sion
of
text
to
POSIX
dates.

As
an
ex-
am-
ple,
con-
sider
the
fol-
low-
ing
code.

We
may
use
this,
for
ex-
am-
ple,
with
time
se-
ries.

output:
html_document
pdf_document:
 de-
 fault

For
 ex-
 am-
 ple
http
 s://
cran
.r-pr
ojec
t.or
g/do
c/co
ntri
b/de
 _J
onge
 +V
 an
_der
 _L
 oo-
Intr
oduc
tion
 _t
o_d
 at
a_c
lean
 in
g_w
 it
h_R
.pdf

```

_____
output:
html_document
pdf_document:
  de-
fault
_____
r
library(lubridate)
dates
<-
c("15/02/2013",
  "15
  Feb
13",
  "It
happened
on
15
02
'13")
dmy(dates)
## [1]
"2013-02-15"
"2013-02-15"
"2013-02-15"

#
(PART)
Su-
per-
vised
Mod-
els
{-}
#
Su-
per-
vised
Clas-
sifi-
ca-
tion

```

output:

html_document

pdf_document:

de-

fault

—

A

clas-

sifi-

ca-

tion

prob-

lem

can

be

de-

fined

as

the

in-

duc-

tion,

from

a

dataset

D ,

of a

clas-

sifi-

ca-

tion

func-

tion

ψ

that,

given

the

at-

tribute

vec-

tor

of

an

in-

stance/example,

re-

turns

a

class

c . A

re-

gres-

sion

prob-

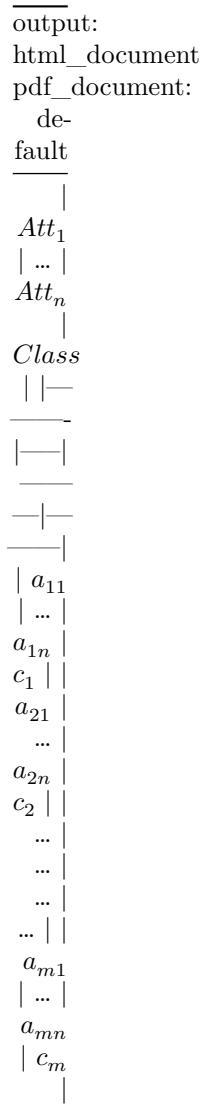
lem,

on

the

output:
html_document
pdf_document:
de-
fault

Dataset,
 D ,
is
typi-
cally
com-
posed
of n
at-
tributes
and
a
class
at-
tribute
 C .



output:

html_document

pdf_document:

de-

fault

Columns

are

usu-

ally

called

at-

tributes

or

fea-

tures.

Typ-

i-

cally,

there

is a

class

at-

tribute,

which

can

be

nu-

meric

or

dis-

crete.

When

the

class

is

nu-

meric,

it is

a re-

gres-

sion

prob-

lem.

With

dis-

crete

val-

ues,

we

can

talk

about

bi-

nary

clas-

-if

output:
html_document
pdf_document:
de-
fault
Once
we
learn
a
model,
new
in-
stances
are
clas-
si-
fied.
As
shown
in
the
next
fig-
ure.



output:
html_document
pdf_document:
de-
fault

We
have
mul-
tiple
types
of
mod-
els
such
as
clas-
sifi-
ca-
tion
trees,
rules,
neu-
ral
net-
works,
and
prob-
a-
bilis-
tic
clas-
si-
fiers
that
can
be
used
to
clas-
sify
in-
stances.

output:
html_document
pdf_document:
de-
fault

Fernandez
et al
pro-
vide
an
ex-
ten-
sive
com-
pari-
son
of
176
clas-
si-
fiers
us-
ing
the
UCI
dataset
(?).

output:
html_document
pdf_document:
de-
fault

We
will
show
the
use
of
dif-
fer-
ent
clas-
sifi-
ca-
tion
tech-
niques

in
the
prob-
lem
of
de-
fect
pre-
dic-
tion
as
run-
ning
ex-
am-
ple.

In
this
ex-
am-
ple, the
dif-
fer-
ent
datasets
are
com-
posed
of
clas-
sical
met-
rics
(Hal-
stead

output:
html_document
pdf_document:
 de-
 fault
 ##
Clas-
 sifi-
 ca-
 tion
Trees
There
 are
 sev-
 eral
 pack-
 ages
 for
 in-
 duc-
 ing
 clas-
 sifi-
 ca-
 tion
 trees,
 for
 ex-
 am-
 ple
 with
 the
 party
 pack-
 age
 (re-
 cur-
 sive
 par-
 ti-
 tion-
 ing):

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  ``'r
  li-
  brary(foreign)
  #
  To
  load
  arff
  file
  li-
  brary(party)
  #
  Build
  a
  deci-
  sion
  tree
  li-
  brary(caret)
  jm1
  <-
read.arff("./datasets/defectPred/D1/JM1.arff")
str(jm1)
```
```

```

output:
html_document
pdf_document:
 de-
 fault
 ##
'data.frame':
9593
obs.
 of
 22
variables:
 ##
 $
LOC_BLANK
 :
 num
 447
 37
 11
 106
 101
 67
 105
 18
 39
 143
 ...
 ##
 $
BRANCH_COUNT
 :
 num
 826
 29
 405
 240
 464
 187
 344
 47
 163
 67
 ...
 ##
 $
LOC_CODE_AND_COMMENT
 :
 num
 12
 8 0
 7
 11
 4 9
 0 1
 7
```

```

output:
html_document
pdf_document:
 de-
 fault
 “‘r
 #
Strat-
ified
par-
ti-
tion
(train-
ing
and
test
sets)
set.seed(1234)
in-
Train
<-
cre-
ate-
Dat-
a-
Parti-
tion(y=jm1$Defective,p=.60,list=FALSE)
jm1.train
<-
jm1[inTrain,]
jm1.test
<-
jm1[-
inTrain,]
```

```

output:
html_document
pdf_document:
 de-
 fault

jm1.formula
<-
jm1$Defective
 ~ .
 #
 for-
 mula
 ap-
 proach:
 de-
 fect
 as
 de-
 pen-
 dent
 vari-
 able
 and
 the
 rest
 as
indep-
 dent
 vari-
 ables
jm1.ctree
<-
ctree(jm1.formula,
data
 =
jm1.train)
```

```

output:
html_document
pdf_document:
 de-
 fault

#
pre-
dict
on
test
data
pred
<-
pre-
dict(jm1.ctree,
new-
data
=
jm1.test)
#
check
pre-
dic-
tion
re-
sult
ta-
ble(pred,
jm1.test$Defective)
```
##
##
pred
  N
  Y
##
  N
168
  11
##
  Y
2965
  692
  r
plot(jm1.ctree)

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____

Using
the
C50
pack-
age,
there
are
two
ways,
spec-
ifly-
ing
train
and
test-
ing
r
library(C50)
require(utils)
#
c50t
<-
C5.0(jm1.train[,-ncol(jm1.train)],
jm1.train[,ncol(jm1.train)])
c50t
<-
C5.0(Defective
~
.,
jm1.train)
summary(c50t)
plot(c50t)
c50tPred
<-
predict(c50t,
jm1.train)
#
table(c50tPred,
jm1.train$Defective)
```

```

_____
output:
html_document
pdf_document:
  de-
  fault
_____
Using
the
'rpart'
pack-
age
r #
Using
the
'rpart'
package
library(rpart)
jm1.rpart
<-
rpart(Defective
  ~
  .,
  data=jm1.train,
  parms
  =
  list(prior
    =
    c(.65,.35),
  split
  =
  "information"))
  #
  par(mfrow
  =
  c(1,2),
  xpd
  =
  NA)
plot(jm1.rpart)
text(jm1.rpart,
use.n
  =
  TRUE)
  _____
  r
jm1.rpart

```

output:
html_document
pdf_document:
de-
fault

n=
5757

node),
split,
n,
loss,
yval,
(yprob)

*
denotes
terminal
node

1)
root
5757
2010.0
N
(0.650
0.350)

2)
LOC_TOTAL<
38.5
4172
969.0
N
(0.751
0.249)
*

3)
LOC_TOTAL>=38.5
1585
825.0
Y
(0.441
0.559)

6)
LOC_TOTAL<
87.5
1027
540.0
N
(0.582

```
_____
output:
html_document
pdf_document:
  de-
fault
_____
r
library(rpart.plot)
#
asRules(jm1.rpart)
#
fancyRpartPlot(jm1.rpart)
###
Rules
C5
Rules
r
library(C50)
c50r
<-
C5.0(jm1.train[,-ncol(jm1.train)],
jm1.train[,ncol(jm1.train)],
rules
=
TRUE)
summary(c50r)
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
    ##
    ##
Call:
  ##
  C5.0.default(x
    =
jm1.train[,,
-ncol(jm1.train)],,
y =
  ##
jm1.train[,,
ncol(jm1.train)],,
rules
  =
TRUE)
  ##
  ##
  ##
C5.0
[Release
2.07
GPL
Edition]
Sun
Oct
10
13:35:54
2021
  ##
-----
  ##
  ##
Class
specified
  by
attribute
`outcome'
  ##
  ##
Read
5757
cases
(22
attributes)
from
undefined.data
  ##
  ##
Rules:
  ##
  ##
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
r #
c50rPred
<-
predict(c50r,
jm1.train)
#
table(c50rPred,
jm1.train$Defective)
##  
Distanced-
based
Meth-
ods
```

output:
html_document
pdf_document:
de-
fault

In
this
case,
there
is no
model
as
such.
Given
a
new
in-
stance
to
clas-
sify,
this
ap-
proach
finds
the
clos-
est
 k -
neighbours
to
the
given
in-
stance.

output:
 html_document
 pdf_document:
 de-
 fault


(Source:
 Wikipedia

```
-  

http  

s://  

en.w  

ikip  

edia  

.org  

/wik  

i/K-  

near  

es  

t_n  

eigh  

bors  

_alg  

orit  

hm)  

“‘r  

li-  

brary(class)  

m1  

<-  

knn(train=jm1.train[,-  

22],  

test=jm1.test[,-  

22],  

cl=jm1.train[22],  

k=3)  

table(jm1.test[22],m1)  

““
```

output:
html_document
pdf_document:
 de-
 fault
 ##
 m1
 ##
 N
 Y
 ##
 N
2851
282
 ##
 Y
554
149
 ##
Neu-
 ral
Net-
works
 ##
 Support
 Vector
 Machine

output:
html_document
pdf_document:
 de-
 fault



(Source:
wikipedia

http
s://
en.w
ikip
edia
.org
/wik
i/Su
ppor
t_v
ecto
r_m
achi
ne)

Prob-
a-
bilis-
tic
Meth-
ods

Naive
Bayes

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
Probabilistic
graph-
ical
model
  as-
  sign-
  ing a
  prob-
  abil-
  ity
  to
  each
  pos-
  sible
  out-
  come
 $p(C_k, x_1, \dots, x_n)$ 

Using
the
klaR
pack-
age
with
caret:
  r
library(caret)
library(klaR)
  ##
Loading
required
package:
MASS
  ##
  ##
Attaching
package:
'MASS'
```

```



---


output:
html_document
pdf_document:
  de-
fault


---


## 
The
following
object
  is
masked
from
'package:dplyr':
  ##
  ##
select
  ##
  The
following
object
  is
masked
from
'package:sm':
  ##
  ##
muscle
  r
model
  <-
NaiveBayes(Defective
  ~
  .,
data
  =
jm1.train)
predictions
  <-
predict(model,
jm1.test[,-22])
confusionMatrix(predictions$class,
jm1.test$Defective)

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  ##
Confusion
Matrix
  and
Statistics
  ##
  ##
Reference
  ##
Prediction
  N
  Y
  ##
  N
2963
554
  ##
  Y
170
149
  ##
  ##
Accuracy
  :
0.811
  ##
95%
CI
  :
(0.799,
0.824)
  ##
No
Information
Rate
  :
0.817
  ##
P-Value
[Acc
  >
NIR]
  :
0.815
  ##
  ##
Kappa
  :
0.2
  ##
  ##
```

```

_____
output:
html_document
pdf_document:
  de-
  fault
_____
Using
the
e1071
pack-
age:
  “r
    li-
brary
(e1071)
n1
<-
naiveBayes(jm1.train$Defective
  ~.,
  data=jm1.train)
#
Show
first
3 re-
sults
  us-
  ing
‘class’
head(predict(n1,jm1.test,
type
  =
c(“class”)),3)
#
class
  by
  de-
fault
  “
  ##
[1]
Y Y
  Y
  ##
Levels:
  N Y

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
r #
Show
first
  3
results
using
'raw'
head(predict(n1,jm1.test,
type
  =
c("raw")),3)
## N Y
## [1,]
8.6e-50
  1
## [2,]
0.0e+00
  1
## [3,]
0.0e+00
  1
```

output:
html_document
pdf_document:
de-
fault

There
are
other
vari-
ants
such
as
TAN
and
KDB
that
do
not
as-
sume
the
inde-
pend-
ence
con-
di-
tion
al-
lowin
us
more
com-
plex
struc-
tures.



output:
html_document
pdf_document:
 de-
 fault

 A
com-
 pre-
 hen-
 sice
com-
 pari-
 son
 of
 ##
Lin-
 ear
Dis-
 crim-
 i-
 nant
Anal-
 ysis
(LDA)

output:

html_document

pdf_document:

de-
fault

One
clas-
sical

ap-
proach
to
clas-
sifi-
ca-
tion

is
Lin-
ear
Dis-
crim-
i-
nant
Anal-
ysis
(LDA),

a
gen-
eral-
iza-
tion
of

Fisher's
lin-
ear
dis-
crim-
i-
nant,
as a
method

used
to
find

a
lin-
ear
com-
bina-
tion
of
fea-
tures
to
sepa-
rate
true-

```
_____
|output:
|html_document
|pdf_document:
|  de-
|  fault
|_____
|r
ldaModel
  <-
  train
  (Defective
  ~
  ,
  data=jm1.train,
  method="lda",
  preProc=c("center","scale"))
ldaModel
```

output:
 html_document
 pdf_document:
 de-
 fault

 ##
 Linear
 Discriminant
 Analysis
 ##
 ##
 5757
 samples
 ##
 21
 predictor
 ##
 2
 classes:
 'N',
 'Y'
 ##
 ##
 Pre-processing:
 centered
 (21),
 scaled
 (21)
 ##
 Resampling:
 Bootstrapped
 (25
 reps)
 ##
 Summary
 of
 sample
 sizes:
 5757,
 5757,
 5757,
 5757,
 5757,
 5757,
 5757,
 5757,
 ...
 ##
 Resampling
 results:
 ##
 ##
 Accuracy
 Kappa
 ##
 0.82
 0.164

output:

html_document

pdf_document:

de-
fault

We
can
ob-
serve
that
we
are
train-
ing
our
model

us-
ing

Defective

~ .
as a
for-
mula
were

Defective

is
the
class
vari-
able
sepa-
red
by ~
and
the
. .

means

the
rest
of
the
vari-
ables.

Also,

we
are
us-
ing
a
filter
for
the
train-
ing
data
to

output:
html_document
pdf_document:
de-
fault

Also,
as
stated
in
the
doc-
u-
men-
ta-
tion
about
the
train
method
: >
http:
//to
pepo
.git
hub.
io/c
aret
/tra
inin
g.ht
ml

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  “‘r
  ctrl
  <-
train-
Con-
trol(method
  =
  “re-
  peat-
  edcv”,repeats=3)
ldaModel
  <-
train
  (De-
  fec-
  tive
  ~.,
data=jm1.train,
method=“lda”,
  tr-
  Con-
  trol=ctrl,
  pre-
  Proc=c(“center”,“scale”))
ldaModel
  ““
```

output:

html_document

pdf_document:

de-

fault

##

Linear

Discriminant

Analysis

##

##

5757

samples

##

21

predictor

##

2

classes:

'N',

'Y'

##

##

Pre-processing:

centered

(21),

scaled

(21)

##

Resampling:

Cross-Validated

(10

fold,

repeated

3

times)

##

Summary

of

sample

sizes:

5181,

5182,

5181,

5182,

5180,

5181,

...

##

Resampling

results:

##

##

Accuracy

Kappa

##

output:
html_document
pdf_document:
de-
fault

Instead
of
ac-
cu-
racy
we
can
acti-
vate
other
met-
rics
us-
ing
summaryFunction=twoClassSummary
such
as
ROC,
sensitivity
and
specificity.
To
do
so,
we
also
need
to
specify
classProbs=TRUE.

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  “‘r
  ctrl
  <-
train-
Con-
trol(method
  =
  “re-
  peat-
  edcv”,repeats=3,
  classProbs=TRUE,
  sum-
  ma-
  ry-
Func-
tion=twoClassSummary)
ldaModel3xcv10
  <-
train
  (De-
  fec-
  tive
  ~.,
  data=jm1.train,
  method=“lda”,
  tr-
  Con-
  trol=ctrl,
  pre-
  Proc=c(“center”,“scale”))
ldaModel3xcv10
  ““
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  ##
Linear
Discriminant
Analysis
  ##
  ##
5757
samples
  ##
  21
predictor
  ##
  2
classes:
'N',
'Y'
  ##
  ##
Pre-processing:
centered
(21),
scaled
(21)
  ##
Resampling:
Cross-Validated
(10
fold,
repeated
  3
times)
  ##
Summary
  of
sample
sizes:
5181,
5181,
5181,
5182,
5182,
5181,
...
  ##
Resampling
results:
  ##
  ##
ROC
Sens
Spec
```

```

output:
html_document
pdf_document:
  de-
  fault
Most
meth-
ods
have
  pa-
  ram-
eters
  that
  need
    to
    be
  optimised
  and
  that
    is
  one
  of
  the
  “r
  pls-
Fit3x10cv
  <-
train
(De-
fec-
tive
  ~.,
data=jm1.train,
method="pls",
  tr-
Con-
trol=trainControl(classProbs=TRUE),
met-
ric="ROC",
  pre-
Proc=c("center","scale"))
plsFit3x10cv
  ""

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  ##
Partial
Least
Squares
  ##
  ##
5757
samples
  ##
  21
predictor
  ##
  2
classes:
'N',
'Y'
  ##
  ##
Pre-processing:
centered
(21),
scaled
(21)
  ##
Resampling:
Bootstrapped
(25
reps)
  ##
Summary
  of
sample
sizes:
5757,
5757,
5757,
5757,
5757,
5757,
5757,
  ...
  ##
Resampling
results
across
tuning
parameters:
  ##
  ##
ncomp
Accuracy
Kappa
```

output:
html_document
pdf_document:
de-
fault

r
plot(plsFit3x10cv)



The
pa-
ram-
eter
tuneLength
al-
low
us
to
spec-
ify
the
num-
ber
val-
ues
per
pa-
ram-
eter
to
con-
sider.

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  ``r
  pls-
Fit3x10cv
  <-
train
  (De-
  fec-
  tive
  ~.,
  data=jm1.train,
  method="pls",
  tr-
  Con-
  trol=ctrl,
  met-
  ric="ROC",
  tune-
  Length=5,
  pre-
  Proc=c("center","scale"))
  plsFit3x10cv
  ``
```

output:
 html_document
 pdf_document:
 de-
 fault

 ##
 Partial
 Least
 Squares
 ##
 ##
 5757
 samples
 ##
 21
 predictor
 ##
 2
 classes:
 'N',
 'Y'
 ##
 ##
 Pre-processing:
 centered
 (21),
 scaled
 (21)
 ##
 Resampling:
 Cross-Validated
 (10
 fold,
 repeated
 3
 times)
 ##
 Summary
 of
 sample
 sizes:
 5181,
 5182,
 5181,
 5182,
 5181,
 5182,
 ...
 ##
 Resampling
 results
 across
 tuning
 parameters:
 ##
 ##

```
_____
output:
html_document
pdf_document:
de-
fault
_____
r
plot(plsFit3x10cv)

Finally
to
pre-
dict
new
cases,
caret
will
use
the
best
class-
fier
ob-
tained
for
pre-
dic-
tion.
r
plsProbs
<-
predict(plsFit3x10cv,
newdata
=
jm1.test,
type
=
"prob")
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
r
plsClasses
<-
predict(plsFit3x10cv,
newdata
=
jm1.test,
type
=
"raw")
confusionMatrix(data=plsClasses,jm1.test$Defective)
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  ##
Confusion
Matrix
  and
Statistics
  ##
  ##
Reference
  ##
Prediction
  N
  Y
  ##
  N
3094
652
  ##
  Y
39
51
  ##
  ##
Accuracy
  :
0.82
  ##
95%
CI
  :
(0.807,
0.832)
  ##
No
Information
Rate
  :
0.817
  ##
P-Value
[Acc
  >
NIR]
  :
0.317
  ##
  ##
Kappa
  :
0.091
  ##
  ##
```

output:
html_document
pdf_document:
 de-
 fault

Pre-
dict-
ing
the
num-
ber
of
de-
fects
(nu-
mer-
ical
class)
From
 the
Bug
Pre-
dic-
tion
Repos-
itory
(BPR)
http:
//bu
g.inf.
 usi.
ch/d
ownl
oad.
php

output:
html_document
pdf_document:
 de-
 fault

Some
datasets
con-
tain
CK
and
other
 11
object-
oriented
met-
rics
for
the
last
ver-
sion
 of
 the
 sys-
 tem
 plus
cate-
 go-
 rized
(with
sever-
 ity
 and
 pri-
 or-
 ity)
post-
release
 de-
 fects.
 Us-
 ing
such
dataset:

```

output:
html_document
pdf_document:
de-
fault


---


“‘r
jdt
<-
read.csv(“./datasets/defectPred/BPD/single-
version-
ck-
oo-
EclipseJDTCore.csv”,
sep=“;”)
#
We
just
use
the
num-
ber
of
bugs,
so
we
re-
moved
oth-
ers
jdtclassname <
-NULLjdtNonTrivialBugs
<-
NULL
jdtmajorBugs <
-NULLjdtminorBugs
<-
NULL
jdtcriticalBugs <
-NULLjdthighPriorityBugs
<-
NULL
jdt$X
<-
NULL

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  #
Caret
  li-
  brary(caret)
  #
Split
data
into
train-
ing
and
test
datasets
set.seed(1)
  in-
Train
  <-
  cre-
  ate-
Dat-
  a-
Parti-
tion(y=jdt$bugs,p=.8,list=FALSE)
jdt.train
  <-
jdt[inTrain,]
jdt.test
  <-
jdt[-
inTrain,]
```
```

```

output:
html_document
pdf_document:
 de-
fault

r
ctrl
<-
trainControl(method
 =
"repeatedcv",repeats=3)
glmModel
<-
train
(bugs
 ~
.,
data=jdt.train,
method="glm",
trControl=ctrl,
preProc=c("center","scale"))
glmModel
```

```

output:
html_document
pdf_document:
 de-
 fault
 ##
Generalized
Linear
Model
 ##
 ##
 798
samples
 ##
 17
predictor
 ##
 ##
Pre-processing:
centered
(17),
scaled
(17)
 ##
Resampling:
Cross-Validated
(10
fold,
repeated
 3
times)
 ##
Summary
 of
sample
sizes:
719,
718,
718,
718,
718,
718,
718,
 ...
 ##
Resampling
results:
 ##
 ##
RMSE
Rsquared
MAE
 ##
0.936
0.273
0.442
```

```

output:
html_document
pdf_document:
 de-
 fault

Others
such
 as
Elas-
 tic-
net:
 r
glmnetModel
 <-
train
(bugs
 ~
 .,
data=jdt.train,
method="glmnet",
trControl=ctrl,
preProc=c("center","scale"))
glmnetModel
```

```

output:
html_document
pdf_document:
 de-
 fault

 ##

glmnet
 ##
 ##
 798
samples
 ##
 17
predictor
 ##
 ##
Pre-processing:
centered
(17),
scaled
(17)
 ##
Resampling:
Cross-Validated
(10
fold,
repeated
 3
times)
 ##
Summary
 of
sample
sizes:
718,
718,
718,
718,
718,
718,
718,
 ...
 ##
Resampling
results
across
tuning
parameters:
 ##
 ##
alpha
lambda
RMSE
Rsquared
MAE
 ##
 0_10
```

---

output:  
 html\_document  
 pdf\_document:  
 de-  
 fault  


---

 ##  
 Bi-  
 nary  
 Lo-  
 gis-  
 tic  
 Re-  
 gres-  
 sion  
 (BLR)  
 Binary  
 Lo-  
 gis-  
 tic  
 Re-  
 gres-  
 sion  
 (BLR)  
 can  
 mod-  
 els  
 fault-  
 proneness  
 as  
 fol-  
 lows  

$$fp(X) = \frac{e^{logit()}}{1+e^{logit(X)}}$$
  
 where  
 the  
 sim-  
 plest  
 form  
 for  
 logit  
 is:  

$$logit(X) = c_0 + c_1 X$$

```

output:
html_document
pdf_document:
de-
fault
“‘r
jdt
<-
read.csv(“./datasets/defectPred/BPD/single-
version-
ck-
oo-
EclipseJDTCore.csv”,
sep=“;”)
#
Caret
li-
brary(caret)
#
Con-
vert
the
re-
sponse
vari-
able
into
a
boolean
vari-
able
(0/1)
jdtbugs[jdtbugs>=1]<-
1
.cbo
<-
jd tcbobugs <
-jdtbugs
```

```

output:
html_document
pdf_document:
 de-
 fault

#
Split
data
into
train-
ing
and
test
datasets
jdt2
 =
data.frame(cbo,
bugs)
 in-
Train
 <-
 cre-
 ate-
Dat-
 a-
Parti-
tion(y=jdt2$bugs,p=.8,list=FALSE)
 jdt-
Train
 <-
jdt2[inTrain,]
jdtTest
 <-
jdt2[-
inTrain,]
 ``
```

output:  
html\_document  
pdf\_document:  
de-  
fault  
BLR  
mod-  
els  
fault-  
proneness  
are  
as  
fol-  
lows  
 $fp(X) = \frac{e^{logit()}}{1+e^{logit(X)}}$   
where  
the  
sim-  
plest  
form  
for  
logit  
is  
 $logit(X) = c_0 + c_1 X$

```

output:
html_document
pdf_document:
 de-
fault

“‘r
#
logit
 re-
gres-
sion
 #
glm-
Logit
 <-
train
(bugs
 ~.,
data=jdt.train,
method="glm",
fam-
ily=binomial(link
 =
logit))
glmLogit
 <-
glm
(bugs
 ~.,
data=jdtTrain,
fam-
ily=binomial(link
 =
logit))
sum-
mary(glmLogit)
“‘
```

```

output:
html_document
pdf_document:
 de-
 fault
 ##
 ##
Call:
 ##
 glm(formula
 =
bugs
 ~
 ,
family
 =
binomial(link
 =
logit),
data
 =
jdtTrain)
 ##
 ##
 Deviance
 Residuals:
 ##
 Min
 1Q
 Median
 3Q
 Max
 ##
-3.654
-0.591
-0.515
-0.471
2.150
 ##
 ##
Coefficients:
 ##
 Estimate
 Std.
 Error
 z
 value
Pr(>|z|)
 ##
 (Intercept)
-2.20649
0.13900
-15.87
<2e-16

 ##
```

```

output:
html_document
pdf_document:
 de-
 fault

Predict
 a
 sin-
 gle
point:
 r
newData
 =
 data.frame(cbo
 =
 3)
predict(glmLogit,
newData,
type
 =
 "response")

 1

0.117
```

```

output:
html_document
pdf_document:
 de-
 fault

Draw
the
re-
sults,
mod-
ified
from:
http:
 //
ww
w.sh
izuk
alab
.com
/too
lkit
s/pl
otti
ng-
logi
stic-
regr
essi
on-
in-r
“‘r
re-
sults
<-
pre-
dict(glmLogit,
jdtTest,
type
 =
“re-
sponse”)
range(jdtTrain$cbo)
““
```

```

output:
html_document
pdf_document:
de-
fault

[1] 0
156
r
range(results)
[1] 0.0992
0.9993
r
plot(jdt2$cbo,jdt2$bugs)
curve(predict(glmLogit,
data.frame(cbo=x),
type
=
"response"),add=TRUE)
[1]
r #
points(jdtTrain$cbo,fitted(glmLogit))
Another
type
of
graph:
r
library(popbio)
##
Attaching
package:
'popbio'
```

```

output:
html_document
pdf_document:
 de-
 fault
 ##
 The
 following
 object
 is
 masked
 from
 'package:caret':
 ##
 ##
 sensitivity
 r
logi.hist.plot(jdt2$cbo,jdt2$bugs,boxp=FALSE,type="hist",col="gray"
[1] "red"
 ##
 The
 caret
 pack-
 age
```

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault

There  
are  
hun-  
dreds  
of  
pack-  
ages  
to  
per-  
form  
clas-  
sifi-  
ca-  
tion  
task  
in R,  
but  
many  
of  
those  
can  
be  
used  
throughout  
the  
'caret'  
pack-  
age  
which  
helps  
with  
many  
of  
the  
data  
min-  
ing  
pro-  
cess  
task  
as  
de-  
scribed  
next.

output:  
html\_document  
pdf\_document:  
  de-  
  fault  
  The  
  caret  
  pack-  
  ageht  
  tp:  
  //to  
  pepo  
  .git  
  hub.  
  io/c  
  aret  
  /  
  pro-  
  vides  
  a  
  uni-  
  fied  
  in-  
  ter-  
  face  
  for  
  mod-  
  eling  
  and  
  pre-  
  dic-  
  tion  
  with  
  around  
  150  
  dif-  
  fer-  
  ent  
  mod-  
  els  
  with  
  tools  
  for:

output:  
html\_document  
pdf\_document:  
  de-  
  fault  
  +  
  data  
  split-  
  ting  
  +  
  pre-  
  processing  
  +  
  fea-  
  ture  
  se-  
  lec-  
  tion  
  +  
model  
  tun-  
  ing  
  us-  
  ing  
  re-  
  sam-  
  pling  
  +  
  vari-  
  able  
  im-  
  por-  
  tance  
  esti-  
  ma-  
  tion,  
  etc.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
Website:  
http:  
//ca  
  ret.  
  r-fo  
  rge.  
  r-pr  
  ojec  
t.org  
JSS  
Pa-  
per:  
  ww  
  w.js  
  tats  
  oft.  
  org/  
  v28/  
  i05/  
pape  
  r  
Book:  
  Ap-  
  plied  
  Pre-  
  dic-  
  tive  
  Mod-  
  eling

#  
Re-  
gres-  
sion  
{#re-  
gres-  
sion}

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
##  
Lin-  
ear  
Re-  
gres-  
sion  
mod-  
eling

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_

-  
*Lin-*  
*ear*  
*Re-*  
*gres-*  
*sion*  
is  
one  
of  
the  
old-  
est  
and  
most  
known  
pre-  
dic-  
tive  
meth-  
ods.  
As  
its  
name  
says,  
the  
idea  
is to  
try  
to  
fit a  
lin-  
ear  
equa-  
tion  
be-  
tween  
a de-  
pen-  
dent  
vari-  
able  
and  
an  
inde-  
pen-  
dent,  
or  
ex-  
plana-  
tory,

output:  
html\_document  
pdf\_document:  
  de-  
  fault

-  
*Mul-*  
*tiple*  
*lin-*  
*ear*  
*re-*  
*gres-*  
*sion*  
uses  
2 or  
more  
inde-  
pend-  
ent  
vari-  
ables  
for  
build-  
ing a  
model.

See  
http  
://  
ww  
w.wi  
kipe  
dia.  
org/  
wiki  
/Lin  
ea  
r\_r  
egre  
ssio  
n.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault

-  
First  
pro-  
posed  
many  
years  
ago  
but  
still  
very  
use-  
ful...



output:  
html\_document  
pdf\_document:  
de-  
fault

The  
equa-  
tion  
takes  
the  
form  
 $\hat{y} =$   
 $b_0 +$   
 $b_1 * x$

The  
method  
used  
to  
choose  
the  
val-  
ues  
 $b_0$   
and  
 $b_1$  is  
to  
min-  
i-  
mize  
the  
sum  
of  
the  
squares  
of  
the  
resid-  
ual  
er-  
rors.

```

output:
html_document
pdf_document:
 de-
 fault
#####
Re-
gres-
sion:
Gal-
ton
Data
Not
re-
lated
to
Soft-
ware
En-
gi-
neer-
ing
but
...
r
library(UsingR)
data(galton)
par(mfrow=c(1,2))
hist(galton$child,col="blue",breaks=100)
hist(galton$parent,col="blue",breaks=100)

r
plot(galton$parent,galton$child,pch=1,col="blue",
cex=0.4)
lm1
<-
lm(galton$child
~
galton$parent)
lines(galton$parent,lm1$fitted,col="red",lwd=3)
plot(galton$parent,lm1$residuals,col="blue",pch=1,
cex=0.4)
abline(c(0,0),col="red",lwd=3)


```

```

output:
html_document
pdf_document:
 de-
 fault

r
qqnorm(galton$child)

#####
Sim-
ple
Lin-
ear
Re-
gres-
sion
```

—  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
—

Given  
two  
vari-  
ables  
   $Y$   
  (re-  
  sponse)  
and  
   $X$   
  (pre-  
  dic-  
  tor),  
the  
  as-  
sump-  
tion  
  is  
that  
there  
is an  
  ap-  
prox-  
  i-  
mate  
  ( $\approx$ )  
lin-  
ear  
rela-  
tion  
  be-  
tween  
those  
vari-  
ables.  
—

The  
math-  
e-  
mat-  
ical  
model  
  of  
the  
ob-  
served  
data  
is de-  
scribed  
  as  
  ( $f_{\mu}$ )

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
- the  
pa-  
ram-  
eter  
 $\beta_0$  is  
named  
the  
*in-*  
*ter-*  
*cept*  
and  
 $\beta_1$  is  
the  
*slope*

-  
Each  
ob-  
ser-  
va-  
tion  
can  
be  
mod-  
eled  
as

output:  
html\_document  
pdf\_document:  
de-  
fault

$$\overline{y_i} = \beta_0 + \beta_1 x_i + \epsilon_i; \epsilon_i \sim N(0, \sigma^2)$$

-  $\epsilon_i$  is the error -  
This means that the variable  $y$  is normally distributed:

$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$$

---

output:  
html\_document  
pdf\_document:  
de-  
fault

---

The  
*pre-*  
*dic-*  
*tions*  
or  
*esti-*  
*ma-*  
*tions*  
of  
this  
model  
are  
ob-  
tained  
by a  
lin-  
ear  
equa-  
tion  
of  
the  
form  
 $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$ ,  
that

is,  
each  
new  
pre-  
dic-  
tion  
is  
com-  
puted  
with  
 $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ .

---

The  
ac-  
tual  
pa-  
ram-  
eters  
 $\beta_0$   
and  
 $\beta_1$

```

output:
html_document
pdf_document:
 de-
 fault

###
```

Least  
Squares

---

output:  
html\_document  
pdf\_document:  
de-  
fault

---

-  
One  
of  
the  
most  
used  
meth-  
ods  
for  
com-  
put-  
ing  
 $\hat{\beta}_0$   
and  
 $\hat{\beta}_1$  is  
the  
cri-  
te-  
rion  
of  
“least  
squares”  
min-  
i-  
miza-  
tion.

---

-  
The  
data  
is  
com-  
posed  
of  $n$   
pairs  
of  
ob-  
ser-  
va-  
tions  
 $(x_i, y_i)$

---

-  
Given  
an  
ob-  
ser-  
va-  
tion  
 $y_i$   
and  
its

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
###  
Lin-  
ear  
re-  
gres-  
sion  
in R  
The  
fol-  
low-  
ing  
are  
the  
ba-  
sic  
com-  
mands  
in  
R:

---

output:  
html\_document  
pdf\_document:  
de-  
fault

---

The  
ba-  
sic  
func-  
tion  
is  
**lm()**,  
that  
re-  
turns  
an  
ob-  
ject  
with  
the  
model.

-  
Other  
com-  
mands:  
**summary**  
prints  
out  
in-  
for-  
ma-  
tion  
about  
the  
re-  
gres-  
sion,  
**coef**  
gives  
the  
coef-  
fi-  
cients  
for  
the  
lin-  
ear  
model,  
**fitted**  
gives  
the  
pre-  
dictd  
value  
of ..

---

output:  
html\_document  
pdf\_document:  
  de-  
  fault  

---

##  
Linear  
Regression  
Diagnos-  
tics

output:  
html\_document  
pdf\_document:  
  de-  
  fault

-  
Sev-  
eral  
plots  
help  
  to  
evaluate  
  the  
suit-  
abil-  
  ity  
  of  
the  
lin-  
ear  
  re-  
gres-  
sion  
  +  
*Resid-*  
*uals*  
  *vs*  
  *fit-*  
  *ted:*  
The  
resid-  
uals  
should  
  be  
  ran-  
domly  
  dis-  
tributed  
around  
  the  
hori-  
zon-  
tal  
line  
rep-  
  re-  
sent-  
ing a  
resid-  
  ual  
  er-  
ror  
  of  
zero;  
that

```

output:
html_document
pdf_document:
 de-
 fault
#####
Sim-
ula-
tion
 ex-
 am-
 ple
#####
Sim-
 u-
late
 a
dataset
```

---

```

output:
html_document
pdf_document:
 de-
fault

“‘r
set.seed(3456)
#
equa-
tion
is y
=
-6.6
+
0.13
x +e
#
range
x
100,400
a <-
-6.6
b <-
0.13
num_obs
<-
60
xmin
<-
100
xmax
<-
400
x <-
sam-
ple(seq(from=xmin,
to =
xmax,
by
=1),
size=
num_obs,
re-
place=FALSE)

```

```

output:
html_document
pdf_document:
 de-
 fault

sderor
<- 9
#
sigma
 for
 the
 er-
 ror
term
 in
 the
model
e <-
rnorm(num_obs,
 0,
sder-
ror)
y <-
 a +
 b *
x +
 e
newlm
 <-
lm(y~x)
sum-
mary(newlm)
```
```

output:
 html_document
 pdf_document:
 de-
 fault

 ##
 Call:
 ##
 lm(formula
 = y
 ~
 x)
 ##
 ##
 Residuals:
 ##
 Min
 1Q
 Median
 3Q
 Max
 ##
 -26.518
 -5.645
 0.363
 5.695
 18.392
 ##
 ##
 Coefficients:
 ##
 Estimate
 Std.
 Error
 t
 value
 Pr(>|t|)
 ##
 (Intercept)
 -7.9060
 3.3922
 -2.33
 0.023
 *
 ##
 x
 0.1331
 0.0132
 10.05
 2.6e-14

 ##

 ##

```
_____
output:
html_document
pdf_document:
de-
fault
“‘r
cfa1
<-
coef(newlm)[1]
cfb2
<-
coef(newlm)[2]
plot(x,y,
xlab=“x
axis”,
ylab=
“y
axis”,
xlim
=
c(xmin,
xmax),
ylim
=
c(0,60),
sub
=
“Line
in
black
is
the
ac-
tual
model”)
ti-
tle(main
=
paste(“Line
in
blue
is
the
Re-
gres-
sion
Line
for”,,
num_obs,
”
points.”))
```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
abline(a
      =
      cfa1,
      b =
      cfb2,
      col=
      "blue",
      lwd=3)
abline(a
      =
      a,
      b =
      b,
      col=
      "black",
      lwd=1)
#orig-
inal
line
```

#####
Sub-
set a
 set
 of
points
from
 the
same
sam-
ple
```

```

output:
html_document
pdf_document:
 de-
 fault

 ““r
 #
 sam-
 ple
 from
 the
 same
 x to
 com-
 pare
 least
 squares
 lines
 #
 change
 the
 de-
 nom-
 ina-
 tor
 in
 newsam-
 ple
 to
 see
 how
 the
 least
 square
 lines
 changes
 ac-
 cord-
 ingly.
 newsam-
 ple
 <-
 as.integer(num_obs/8)
 #
 num-
 ber
 of
 pairs
 x,y
```

```

output:
html_document
pdf_document:
de-
fault
idxs_x1
<-
sam-
ple(1:num_obs,
size
=
newsam-
ple,
re-
place
=
FALSE)
#sam-
ple
in-
dexes
x1
<-
x[idxs_x1]
e1
<-
e[idxs_x1]
y1
<- a
+ b
* x1
+ e1
xy_obs
<-
data.frame(x1,
y1)
names(xy_obs)
<-
c("x_obs",
"y_obs")

```

```

output:
html_document
pdf_document:
 de-
 fault

newlm1
<-
lm(y1~x1)
sum-
mary(newlm1)
```
```

output:
 html_document
 pdf_document:
 de-
 fault

 ##
 Call:
 ##
 lm(formula
 =
 y1
 ~
 x1)
 ##
 ##
 Residuals:
 ##
 1
 2
 3
 4
 5
 6
 7
 ##
 3.968
 -8.537
 3.141
 -8.723
 7.294
 -0.235
 3.092
 ##
 ##
 Coefficients:
 ##
 Estimate
 Std.
 Error
 t
 value
 Pr(>|t|)
 ##
 (Intercept)
 2.9107
 7.7166
 0.38
 0.722
 ##
 x1
 0.0913
 0.0328
 2.79
 0.039

```
_____
output:
html_document
pdf_document:
de-
fault
_____
``r
cfa21
<-
coef(newlm1)[1]
cfb22
<-
coef(newlm1)[2]
plot(x1,y1,
xlab="x
axis",
ylab=
"y
axis",
xlim
=
c(xmin,
xmax),
ylim
=
c(0,60))
title(main
=
paste("New
line
in
red
with",
newsam-
ple, "
points
in
sam-
ple"))

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
abline(a
      = a,
      b =
      b,
      col=
      "black",
      lwd=1)
      #
      True
      line
abline(a
      =
      cfa1,
      b =
      cfb2,
      col=
      "blue",
      lwd=1)
#sam-
      ple
abline(a
      =
      cfa21,
      b =
      cfb22,
      col=
      "red",
      lwd=2)
#new
      line
      ``
```



output:
html_document
pdf_document:
 de-
 fault

Com-
pute
 a
con-
 fi-
dence
 in-
 ter-
 val
 on
 the
orig-
inal
sam-
 ple
 re-
gres-
sion
line

```

_____
output:
html_document
pdf_document:
  de-
  fault
_____
“‘r
newx
<-
seq(xmin,
xmax)
ypré-
dicted
<-
pre-
dict(newlm,
new-
data=data.frame(x=newx),
in-
ter-
val=
“con-
fi-
dence”, 
level=
0.90,
se =
TRUE)
plot(x,y,
xlab=“x
axis”, 
ylab=
“y
axis”, 
xlim
=
c(xmin,
xmax),
ylim
=
c(0,60))
#
points(x1,
fit-
ted(newlm1))
abline(newlm)

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
lines(newx,ypredictedfit[,2],col =
"red",lty =
2)lines(newx,ypredictedfit[,3],col="red",lty=2)
```

```

```

output:
html_document
pdf_document:
 de-
fault

r #
Plot
the
residuals
or
errors
ypredicted_x
<-
predict(newlm,
newdata=data.frame(x=x))
plot(x,y,
xlab="x
axis",
ylab=
"y
axis",
xlim
=
c(xmin,
xmax),
ylim
=
c(0,60),
sub
=
"",
pch=19,
cex=0.75)
title(main
=
paste("Residuals
or
errors",
num_obs,
"
points."))
abline(newlm)
segments(x,
y,
x,
ypredicted_x)

```

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
  
#####  
Take  
an-  
other  
sam-  
ple  
from  
the  
model  
and  
ex-  
plore

---

```

output:
html_document
pdf_document:
 de-
 fault

 “‘r
 #
equa-
tion
is y
=
-6.6
+
0.13
x +e
#
range
 x
100,400
num_obs
<-
 35
xmin
<-
 100
xmax
<-
 400
 x3
<-
sam-
ple(seq(from=xmin,
to =
xmax,
 by
=1),
size=
num_obs,
 re-
place=FALSE)
sder-
 ror
<-
 14
#
sigma
 for
the
er-
ror
term
 in
the
model
 e3
 -

```

```

output:
html_document
pdf_document:
 de-
 fault

 y3
 <- a
 + b
 * x3
 + e3
newlm3
 <-
 lm(y3~x3)
 sum-
 mary(newlm3)
 ``
```

---

output:

html\_document

pdf\_document:

de-

fault

---

##

##

Call:

##

lm(formula

=

y3

~

x3)

##

##

Residuals:

##

Min

1Q

Median

3Q

Max

##

-40.87

-9.20

-2.28

12.08

47.17

##

##

Coefficients:

##

Estimate

Std.

Error

t

value

Pr(>|t|)

##

(Intercept)

-0.9284

8.7458

-0.11

0.9161

##

x3

0.1193

0.0345

3.45

0.0015

\*\*

##

---

##

Signif. codes:

```

output:
html_document
pdf_document:
 de-
 fault

 “‘r
cfa31
 <-
 coef(newlm3)[1]
cfb32
 <-
 coef(newlm3)[2]
plot(x3,y3,
 xlab=“x
 axis”,
 ylab=
 “y
 axis”,
 xlim
 =
 c(xmin,
 xmax),
 ylim
 =
 c(0,60))
 ti-
 tle(main
 =
 paste(“Line
 in
 red
 is
 the
 Re-
 gres-
 sion
 Line
 for”,,
 num_obs,
 ”
 points.”))
abline(a
 =
 cfa31,
 b =
 cfb32,
 col=
 “red”,
 lwd=3)
abline(a
 = a,
 b =
 b,
 col=
 “black”,
 lwd=2)
```

```

output:
html_document
pdf_document:
 de-
 fault

con-
 fi-
dence
 in-
 ter-
vals
 for
 the
new
sam-
 ple
newx
 <-
seq(xmin,
xmax)
ypre-
dicted
 <-
 pre-
dict(newlm3,
new-
data=data.frame(x3=newx),
 in-
ter-
val=
“con-
 fi-
dence”,
level=
0.90,
se =
TRUE)
lines(newx,ypredictedfit[,2], col =
"red", lty =
2, lwd =
2)lines(newx, ypredictedfit[,3], col = "red", lty = 2,
lwd = 2)
```


```

output:
html_document
pdf_document:
de-
fault

Di-
ag-
nos-
tics
fro
as-
sess-
ing
the
re-
gres-
sion
line

output:
html_document
pdf_document:
de-
fault

Resid-
ual
Stan-
dard
Er-
ror -
It
gives
us
an
idea
of
the
typi-
cal
or
aver-
age
er-
ror
of
the
model.
It is
the
esti-
mated
stan-
dard
devi-
a-
tion
of
the
resid-
uals.

output:
html_document
pdf_document:
 de-
 fault

 R^2
statis-
tic -
This
 is
 the
pro-
por-
tion
 of
vari-
abil-
 ity
 in
 the
data
 that
is ex-
plained
 by
 the
model.
Best
val-
ues
 are
those
close
to 1.

Mul-
tiple
Lin-
ear
Re-
gres-
sion

output:
html_document
pdf_document:
de-
fault

Par-
tial
Least
Squares
- If
sev-
eral
pre-
dic-
tors
are
highly
cor-
re-
lated,
the
least
squares
ap-
proach
has
high
vari-
abil-
ity. -
PLS
finds
lin-
ear
com-
bi-
na-
tions
of
the
pre-
dic-
tors,
that
are
called
com-
po-
nents
or
la-
tent
vari-
ables.

output:
html_document
pdf_document:
 de-
 fault

Lin-
ear
 re-
 gres-
 sion
 in
Software
 Ef-
 fort
 esti-
 ma-
 tion

output:

html_document

pdf_document:

de-

fault

Fitting

a

lin-

ear

model

to

log-

log -

the

pre-

dic-

tive

power

equa-

tion

is

$y =$

$e^{b_0} *$

x^{b_1} ,

ig-

nor-

ing

the

bias

cor-

rec-

tions.

Note:

de-

pend-

ing

how

the

er-

ror

term

be-

haves

we

could

try

an-

other

gen-

eral

lin-

ear

model

(GLM)

or

other

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
r
library(foreign)
china
<-
read.arff("./datasets/effortEstimation/china.arff")
china_size
<-
china$AFP
summary(china_size)
##
Min.
 1st
Qu.
Median
Mean
 3rd
Qu.
Max.
##
  9
 100
 215
 487
 438
17518
r
china_effort
<-
china$Effort
summary(china_effort)
```

```

output:
html_document
pdf_document:
  de-
fault
  ##
Min.
  1st
  Qu.
Median
Mean
  3rd
  Qu.
Max.
  ##
  26
  704
  1829
  3921
  3826
  54620
  r
par(mfrow=c(1,2))
hist(china_size,
col="blue",
xlab="Adjusted
Function
Points",
main="Distribution
of
AFP")
hist(china_effort,
col="blue",xlab="Effort",
main="Distribution
of
Effort")
  L L
  r
boxplot(china_size)
boxplot(china_effort)
  I I

```

```
_____
output:
html_document
pdf_document:
de-
fault
_____
r
qqnorm(china_size)
qqline(china_size)
qqnorm(china_effort)
qqline(china_effort)
[|] [|]
Applying
the
log
func-
tion
(it
computes
nat-
ural
loga-
rithms,
base
e)
[|] [|]
[|] [|]
r
linmodel_logchina
<-
lm(logchina_effort
~
logchina_size)
par(mfrow=c(1,1))
plot(logchina_size,
logchina_effort)
abline(linmodel_logchina,
lwd=3,
col=3)
[|]
```

```
_____
output:
html_document
pdf_document:
  de-
fault
_____
r
par(mfrow=c(1,2))
plot(linmodel_logchina,
  ask
  =
FALSE)

r
linmodel_logchina
  ##
  ##
Call:
  ##
lm(formula
  =
logchina_effort
  ~
logchina_size)
  ##
  ##
Coefficients:
  ##
  (Intercept)
logchina_size
  ##
3.301
0.768
  ##
Ref-
er-
ences
```

—
output:
html_document
pdf_document:
 de-
 fault
—

The
New
Statis-
tics
with
R,
Andy
Hec-
tor,
2015
- An
In-
tro-
duc-
tion
to R,
W.N.
Ven-
ables
and
D.M.
Smith
and
the
R
De-
vel-
op-
ment
Core
Team

—
Prac-
tical
Data
Sci-
ence
with
R,
Nina
Zumel
and
John
Mount
- G.
James
et al,
An
In-
tro-

output:
html_document
pdf_document:
 de-
 fault

(PART)
 Un-
 su-
 per-
 vised
 Mod-
 els
 {-}
 #
 Un-
 su-
 per-
 vised
 or
 De-
 scrip-
 tive
 mod-
 eling

output:

html_document

pdf_document:

de-

fault

From

the

de-

scrip-

tive

(un-

su-

per-

vised)

point

of

view,

pat-

terns

are

found

to

pre-

dict

fu-

ture

be-

haviour

or

esti-

mate.

This

in-

clude

asso-

cia-

tion

rules,

clus-

ter-

ing,

or

tree

clus-

ter-

ing

which

aim

at

group-

ing

to-

gether

ob-

jects

(e.g.

```

output:
html_document
pdf_document:
  default
  |
  Att1
  |
  Attn
  | |-
  →
  | |-
  |
  | |
  a11
  ...
  a1n
  | a21
  | ...
  a2n
  | ...
  ...
  ... | |
  am1
  | ...
  amn
  |
  ##
Clus-
ter-
ing
“‘r
li-
brary(foreign)
  li-
brary(fpc)
    kc1
    <-
read.arff("./datasets/defectPred/D1/KC1.arff")

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
  _____
  #
  Split
  into
  train-
  ing
  and
  test
  datasets
  set.seed(1)
  ind
  <-
  sam-
  ple(2,
  nrow(kc1),
  re-
  place
  =
  TRUE,
  prob
  =
  c(0.7,
  0.3))
  kc1.train
  <-
  kc1[ind==1,
  ]
  kc1.test
  <-
  kc1[ind==2,
  ]
  #
  No
  class
  kc1.train$Defective
  <-
  NULL
```

```

output:
html_document
pdf_document:
de-
fault


---


ds
<-
db-
scan(kc1.train,
eps
=
0.42,
MinPts
= 5)
kc1.kmeans
<-
kmeans(kc1.train,
2) "
#####
k-
Means
r
library(reshape,
quietly=TRUE)
library(graphics)
kc1kmeans
<-
kmeans(sapply(na.omit(kc1.train),
rescaler,
"range"),
10)
#plot(kc1kmeans,
col
=
kc1kmeans$cluster)
#points(kc1kmeans$centers,
col
=
1:5,
pch
=
8)

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
## As-
soci-
a-
tion
rules
“‘r
li-
brary(arules)
# x
<-
as.numeric(kc1$LOC_TOTAL)
#
str(x)
#
sum-
mary(x)
#
hist(x,
breaks=30,
main=“LoC
To-
tal”)
#
xDisc
<-
dis-
cretize(x,
cate-
gories=5)
#
ta-
ble(xDisc)
```

```

_____
output:
html_document
pdf_document:
  de-
  fault
_____
for(i
    in
1:21)
kc1[,i]
  <-
  dis-
cretize(kc1[,i],
method
  =
  “in-
  ter-
  val”,,
breaks
= 5)
rules
  <-
apri-
ori(kc1,
  pa-
ram-
eter
  =
list(minlen=3,
supp=0.05,
conf=0.35),
  ap-
pear-
ance
  =
list(rhs=c(“Defective=Y”),
  de-
fault=“lhs”),,
con-
trol
  =
list(verbose=F))

```

```
_____
output:
html_document
pdf_document:
  de-
  fault
_____
#rules
<-
apri-
ori(kc1,
  #
  pa-
  ram-
  eter
  =
list(minlen=2,
  supp=0.05,
  conf=0.3),
  #
  ap-
  pear-
  ance
  =
list(rhs=c("Defective=Y",
  "De-
  fec-
  tive=N"),
  #
  de-
  fault="lhs"))
inspect(rules)
```
```

---

```

output:
html_document
pdf_document:
de-
fault

###
lhs
rhs
support
confidence
coverage
lift
count
##
[1]
{HALSTEAD_CONTENT=[38.6,77.2],
##
HALSTEAD_LEVEL=[0,0.4]}
=>
{Defective=Y}
0.0539
0.370
0.146
2.39
113
##
[2]
{LOC_CODE_AND_COMMENT=[0,2.4],
##
HALSTEAD_CONTENT=[38.6,77.2]}
=>
{Defective=Y}
0.0525
0.377
0.139
2.43
110
##
[3]
{LOC_CODE_AND_COMMENT=[0,2.4],
##
HALSTEAD_CONTENT=[38.6,77.2],
##
HALSTEAD_LEVEL=[0,0.4]}
=>
{Defective=Y}
0.0515
0.374
0.138
2.41
108

```

```

|output:
|html_document
|pdf_document:
| de-
| fault
|_____
|r
library(arulesViz)
plot(rules)

```

```
#(PART)
Eval-
ua-
tion
{-}
#
Eval-
ua-
tion
of
Mod-
els
```

output:  
html\_document  
pdf\_document:  
de-  
fault  
Once  
we  
ob-  
tain  
the  
model  
with  
the  
train-  
ing  
data,  
we  
need  
to  
eval-  
uate  
it  
with  
some  
new  
data  
(test-  
ing  
data).

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
>  
**No**  
**Free**  
**Lunch**  
**the-**  
**o-**  
**rem**  
> In  
the  
ab-  
sence  
of  
any  
knowl-  
edge  
about  
the  
pre-  
dic-  
tion  
prob-  
lem,  
no  
model  
>  
can  
be  
said  
to  
be  
uni-  
formly  
bet-  
ter  
than  
any  
other

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
##  
Build-  
  ing  
  and  
  Vali-  
  dat-  
  ing a  
Model

output:

html\_document

pdf\_document:

de-

fault

—

We

can-

not

use

the

the

same

data

for

train-

ing

and

test-

ing

(it is

like

eval-

uat-

ing

a

stu-

dent

with

the

exer-

cises

pre-

vi-

ously

solved

in

class,

the

stu-

dent's

marks

will

be

“op-

ti-

mistic”

and

we

do

not

know

about

stu-

dent

ca-

—

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault

Therefore,  
we  
should,  
at a  
min-  
i-  
mum,  
di-  
vide  
the  
dataset  
into  
*train-*  
*ing*  
and  
*test-*  
*ing*,  
learn  
the  
model  
with  
the  
train-  
ing  
data  
and  
test  
it  
with  
the  
rest  
of  
data  
as  
ex-  
plained  
next.

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
###  
Hold-  
out  
ap-  
proach

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault

### **Holdout**

#### **ap- proach**

con-  
sists  
of  
di-  
vid-  
ing  
the  
dataset  
into  
*train-*  
*ing*  
(typ-  
i-  
cally  
ap-  
prox.  
2/3  
of  
the  
data)  
and  
*test-*  
*ing*  
(ap-  
prox  
1/3  
of  
the  
data).  
+

Prob-  
lems:  
Data  
can  
be  
skewed,  
miss-  
ing  
classes,  
etc.

if  
ran-  
domly  
di-  
vided.  
Strat-  
ifica-  
tion

output:  
html\_document  
pdf\_document:  
de-  
fault  
Holdout  
esti-  
mate  
can  
be  
made  
more  
reli-  
able  
by  
re-  
peat-  
ing  
the  
pro-  
cess  
with  
dif-  
fer-  
ent  
sub-  
sam-  
ples  
(re-  
peated  
hold-  
out  
method).

output:  
html\_document  
pdf\_document:  
de-  
fault  
The  
er-  
ror  
rates  
on  
the  
dif-  
fer-  
ent  
iter-  
a-  
tions  
are  
aver-  
aged  
(over-  
all  
er-  
ror  
rate).

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_

-  
Usu-  
ally,  
part  
of  
the  
data  
points  
are  
used  
for  
build-  
ing  
the  
model  
and  
the  
re-  
main-  
ing  
points  
are  
used  
for  
vali-  
dat-  
ing  
the  
model.  
There  
are  
sev-  
eral  
ap-  
proaches  
to  
this  
pro-  
cess.

-  
*Vali-*  
*da-*  
*tion*  
*Set*  
*ap-*  
*proach:*  
it is  
the  
sim-  
plest  
method

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
de-  
fault  
\_\_\_\_\_  
\_\_\_\_\_  
###  
Cross  
Vali-  
da-  
tion  
(CV)

output:

html\_document

pdf\_document:

de-

fault

—

k-

fold

Cross-

Validation

in-

volves

ran-

domly

di-

vid-

ing

the

set

of

ob-

ser-

va-

tions

into

k

groups,

or

folds,

of

ap-

prox-

i-

mately

equal

size.

One

fold

is

treated

as a

vali-

da-

tion

set

and

the

method

is

trained

on

the

re-

main-

ing

$k - 1$

fold-

output:  
html\_document  
pdf\_document:  
de-  
fault  
+  
1st  
step:  
split  
dataset  
 $(D)$   
into  
 $k$   
sub-  
sets  
of  
ap-  
prox-  
i-  
mately  
equal  
size  
 $C_1, \dots, C_k$

output:  
html\_document  
pdf\_document:  
de-  
fault  
+  
2nd  
step:  
we  
con-  
struct  
a  
dataset  
 $D_i =$   
 $D -$   
 $C_i$   
used  
for  
train-  
ing  
and  
test  
the  
ac-  
cu-  
racy  
of  
the  
clas-  
sifier  
 $D_i$   
on  
 $C_i$   
sub-  
set  
for  
test-  
ing

\_\_\_\_\_  
output:  
html\_document  
pdf\_document:  
  de-  
  fault  
\_\_\_\_\_  
Having  
done  
this  
for  
all  $k$   
  we  
  esti-  
  mate  
  the  
  ac-  
  cu-  
  racy  
  of  
  the  
method  
  by  
aver-  
  ag-  
  ing  
  the  
  ac-  
  cu-  
  racy  
over  
the  
   $k$   
cross-  
validation  
  tri-  
  als  
\_\_\_\_\_  
###  
Leave-  
One-  
Out  
Cross-  
Validation  
(LOO-  
CV)

output:  
html\_document  
pdf\_document:  
  de-  
  fault

-  
*Leave-*  
*One-*  
*Out*  
*Cross-*  
*Validation*  
(LOO-  
CV):  
This  
is a  
spe-  
cial  
case  
of  
CV.  
In-  
stead  
of  
cre-  
at-  
ing  
two  
sub-  
sets  
for  
train-  
ing  
and  
test-  
ing,  
a  
sin-  
gle  
ob-  
ser-  
va-  
tion  
is  
used  
for  
the  
vali-  
da-  
tion  
set,  
and  
the  
re-  
main-  
ing  
-h

---

output:  
html\_document  
pdf\_document:  
  de-  
  fault

---

output: html\_document: default pdf\_document: default — ## Evaluation of Classification Models

The confusion matrix (which can be extended to multiclass problems) is a table that presents the results of a classification algorithm. The following table shows the possible outcomes for binary classification problems:

|                | <i>ActPos</i> | <i>ActNeg</i> |
|----------------|---------------|---------------|
| <i>PredPos</i> | <i>TP</i>     | <i>FP</i>     |
| <i>PredNeg</i> | <i>FN</i>     | <i>TN</i>     |

where *True Positives (TP)* and *True Negatives (TN)* are respectively the number of positive and negative instances correctly classified, *False Positives (FP)* is the number of negative instances misclassified as positive (also called Type I errors), and *False Negatives (FN)* is the number of positive instances misclassified as negative (Type II errors).

- Confusion Matrix in Wikipedia

From the confusion matrix, we can calculate:

- *True positive rate, or recall* ( $TP_r = \text{recall} = TP/TP + FN$ ) is the proportion of positive cases correctly classified as belonging to the positive class.
- *False negative rate* ( $FN_r = FN/TP + FN$ ) is the proportion of positive cases misclassified as belonging to the negative class.
- *False positive rate* ( $FP_r = FP/FP + TN$ ) is the proportion of negative cases misclassified as belonging to the positive class.
- *True negative rate* ( $TN_r = TN/FP + TN$ ) is the proportion of negative cases correctly classified as belonging to the negative class.

There is a trade-off between  $FP_r$  and  $FN_r$  as the objective is minimize both metrics (or conversely, maximize the true negative and positive rates). It is possible to combine both metrics into a single figure, predictive *accuracy*:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

to measure performance of classifiers (or the complementary value, the *error rate* which is defined as  $1 - \text{accuracy}$ )

- Precision, fraction of relevant instances among the retrieved instances,

$$\frac{TP}{TP + FP}$$

- Recall\$ (*sensitivity* probability of detection,  $PD$ ) is the fraction of relevant instances that have been retrieved over total relevant instances,  $\frac{TP}{TP+FN}$
- *f-measure* is the harmonic mean of precision and recall,  $2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$
- G-mean:  $\sqrt{PD \times Precision}$
- G-mean2:  $\sqrt{PD \times Specificity}$
- J coefficient,  $j-coeff = sensitivity + specificity - 1 = PD - PF$
- A suitable and interesting performance metric for binary classification when data are imbalanced is the Matthew's Correlation Coefficient ( $MCC$ )~?:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

$MCC$  can also be calculated from the confusion matrix. Its range goes from -1 to +1; the closer to one the better as it indicates perfect prediction whereas a value of 0 means that classification is not better than random prediction and negative values mean that predictions are worst than random.

### 2.8.1 Prediction in probabilistic classifiers

A probabilistic classifier estimates the probability of each of the possible class values given the attribute values of the instance  $P(c|x)$ . Then, given a new instance,  $x$ , the class value with the highest a posteriori probability will be assigned to that new instance (the *winner takes all* approach):

$$\psi(x) = \text{argmax}_c(P(c|x))$$

## 2.9 Other Metrics used in Software Engineering with Classification

In the domain of defect prediction and when two classes are considered, it is also customary to refer to the *probability of detection*, ( $pd$ ) which corresponds to the True Positive rate ( $TP_{rate}$  or *Sensitivity*) as a measure of the goodness of the model, and *probability of false alarm* ( $pf$ ) as performance measures~?.

The objective is to find which techniques that maximise  $pd$  and minimise  $pf$ . As stated by Menzies et al., the balance between these two measures depends on the project characteristics (e.g. real-time systems vs. information management systems) it is formulated as the Euclidean distance from the sweet spot  $pf = 0$  and  $pd = 1$  to a pair of  $(pf, pd)$ .

$$\text{balance} = 1 - \frac{\sqrt{(0 - pf^2) + (1 - pd^2)}}{\sqrt{2}}$$

It is normalized by the maximum possible distance across the ROC square  $(\sqrt{2}, 2)$ , subtracted this value from 1, and expressed it as a percentage.

## 2.10 Graphical Evaluation

### 2.10.1 Receiver Operating Characteristic (ROC)

The *Receiver Operating Characteristic (ROC)*(?) curve which provides a graphical visualisation of the results.

The Area Under the ROC Curve (AUC) also provides a quality measure between positive and negative rates with a single value.

A simple way to approximate the AUC is with the following equation:  $AUC = \frac{1+TP_r-FP_r}{2}$

### 2.10.2 Precision-Recall Curve (PRC)

Similarly to ROC, another widely used evaluation technique is the Precision-Recall Curve (PRC), which depicts a trade off between precision and recall and can also be summarised into a single value as the Area Under the Precision-Recall Curve (AUPRC)~?.

%AUPCR is more accurate than the ROC for testing performances when dealing with imbalanced datasets as well as optimising ROC values does not necessarily optimises AUPR values, i.e., a good classifier in AUC space may not be so good in PRC space. %The weighted average uses weights proportional to class frequencies in the data. %The weighted average is computed by weighting the measure of class (TP rate, precision, recall ...) by the proportion of instances there are in that class. Computing the average can be sometimes be misleading.

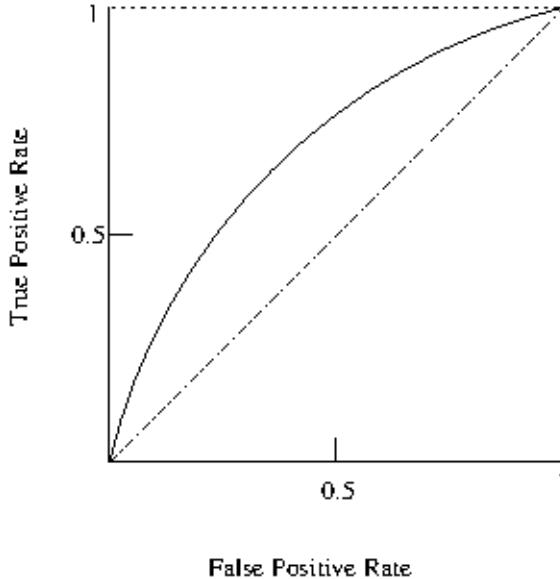


Figure 2.4: Receiver Operating Characteristic

For instance, if class 1 has 100 instances and you achieve a recall of 30%, and class 2 has 1 instance and you achieve recall of 100% (you predicted the only instance correctly), then when taking the average (65%) you will inflate the recall score because of the one instance you predicted correctly. Taking the weighted average will give you 30.7%, which is much more realistic measure of the performance of the classifier.

## 2.11 Numeric Prediction Evaluation

In the case of defect prediction, it matters the difference between the predicted value and the actual value. Common performance metrics used for numeric prediction are as follows, where  $\hat{y}_n$  represents the predicted value and  $y_n$  the actual one.

Mean Square Error (*MSE*)

$$MSE = \frac{(\hat{y}_1 - y_1)^2 + \dots + (\hat{y}_n - y_n)^2}{n} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

Root mean-squared error (*RMSE*)

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}}$$

Mean Absolute Error (*MAE*)

$$MAE = \frac{|\hat{y}_1 - y_1| + \dots + |\hat{y}_n - y_n|}{n} = \sqrt{\frac{\sum_{t=1}^n |\hat{y}_t - y_t|}{n}}$$