

lab03 – R

R web site: <http://www.r-project.org/>

R reference card: <http://cran.r-project.org/doc/contrib/Short-refcard.pdf>

Quick R - <http://www.statmethods.net/index.html>

A brief Introduction to R - Examples of Expressions

You should try each of the following commands and type ?command in case you want to get a description of the method (e.g. ?log)

Simple expressions

3+5*6	
a <- 2+2	assigns expression's value to variable 'a'
3^(3+2)	
b <- 1:10	defines a sequence and assigns it to variable 'b'
b*3	
log(b)	
seq(1,15,2)	defines a sequence
?log	help on a function
help.search("clustering")	
objects()	lists existing objects
rm(obj1, obj2,...)	removes existing objects
str(obj)	displays the internal structure of an object
Menu "File; Change dir..."	change directory pointed by the prompt
dir()	displays directory contents
v <- c(1,2,3,4,5)	defines a vector
m <- matrix(c(1,2,3,4),2,2)	defines 2x2 matrix de 2x2
a <- array(1:8, c(2,2,2))	defines 2x2x2 array
m*2	
m[1,1]	
m[1,]	

The auto dataset (UCI machine learning repository)

source: <http://archive.ics.uci.edu/ml/machine-learning-databases/autos/>

(in the dataset the missing values are denoted by '?')

File/change dir	to the directory with the dataset
auto<-read.table("lab03-auto.txt",sep=";")	
auto[1:10,]	first 10 rows
auto <-read.table("lab03-auto.txt",sep=";", header = TRUE)	
summary(auto)	summary statistics
str(auto)	
hist(auto\$price)	Histogram (does not work because missing values are represented as '?'. See below to change this.)
hist(auto[,2:2])	
pairs(auto[,15:20])	scatters for all pairs of variables
plot(auto\$bore,auto\$stroke)	scatter 2 variables
plot(auto\$horsepower, auto\$price)	

missing values are imported as factors

In R, missing values are represented by the symbol NA (not available)

is.na(auto)	indicates which elements are missing
-------------	--------------------------------------

For the variables with missing values replace ? by NA

auto[auto\$normalized.losses=='?',"normalized.losses"]<-NA	
auto[auto\$num.of.doors=='?',"num.of.doors"]<-NA	
auto[auto\$bore=='?',"bore"]<-NA	
auto[auto\$stroke=='?',"stroke"]<-NA	
auto[auto\$horsepower=='?',"horsepower"]<-NA	
auto[auto\$peak.rpm=='?',"peak.rpm"]<-NA	
auto[auto\$price=='?',"price"]<-NA	
str(auto)	note that some variables were assumed as categorical due to the '?' character

Convert data types to numeric

auto\$normalized.losses <- as.numeric(auto\$normalized.losses)	
auto\$num.of.doors <- as.numeric(auto\$num.of.doors)	
auto\$bore <- as.numeric(auto\$bore)	
auto\$stroke <- as.numeric(auto\$stroke)	
auto\$horsepower <- as.numeric(auto\$horsepower)	
auto\$peak.rpm <- as.numeric(auto\$peak.rpm)	
auto\$price <- as.numeric(auto\$price)	

auto[!complete.cases(auto),]	rows having missing values
auto[complete.cases(auto),]	complete rows
auto_complete <- na.omit(auto)	
mean(auto\$price)	Average. Returns NA due to the missing values
mean(auto\$price, na.rm=T)	Average omitting the missing values
auto[is.na(auto\$price),'price']	selects rows with NA in price
auto[is.na(auto\$price),'price'] <- mean(auto\$price, na.rm=T)	Fill missing values with the average
auto[is.na(auto\$price),'price'] <- median(auto\$price, na.rm=T)	Fill missing values with the median

Using scripts in R

Menu "File; New script"	create the script
write down the following commands	
a <- runif(100,10,20)	random uniform distribution
hist(a)	plots an histogram
Menu "File;Save as..." test.R	save the script
source('teste.R')	run the script

Using scripts in R (using loops)

Writes to files five different histograms

```
for (i in 1:5){
  a <- runif(10*i, 10, 20)
  hist(a, main=paste("uniform n=", 10*i))
  savePlot(filename= paste("uniform n=", 10*i),type="jpeg")
}
```

Discretization and normalization

(here we will use some functions borrowed from the dprep R package, that is not compatible with the current version of R.

See files in <http://cran.r-project.org/src/contrib/Archive/dprep/> for the source code and manuals of the package.

Alternatively, you can install a previous version of R, 2.10, and run the package.

Previous versions are available here: <http://cran.r-project.org/bin/windows/base/old/>)

data <- read.table("exemplo.txt")	Load a short dataset:
-----------------------------------	-----------------------

Script for equal width discretization (from dprep R package)

```
#data: name of the dataset to be discretized
#varcon: vector containing the columns to be discretized.
# example desc.ew(data,1:1)
desc.ew <- function(data, varcon){
  p <- dim(data)[2]
  f <- p - 1
  ft <- rep(0, f)
  for(i in 1:length(varcon)) {
    ft[varcon[i]] <- 1
  }
  for(i in 1:f) {
    if(ft[i] > 0) {
      grupos <- nclass.scott(data[, i])
      data[, i] <- as.vector(cut(data[, i], grupos, labels=FALSE))
    }
  }
  data
}
```

Script for equal depth discretization (from dprep R package)

#data: name of the dataset to be discretized
#varcon: vector containing the columns to be discretized.
#k: number of intervals per column
#example: **disc.ef(data,1:1,3)**

```
disc.ef <- function(data, varcon, k){
  data = as.matrix(data)
  p <- dim(data)[2]
  f <- p - 1
  ft <- rep(0, f)
  for(i in 1:length(varcon)) {
    ft[varcon[i]] = 1
  }
  for(i in 1:f) {
    if(ft[i] > 0) {
      data[, i] <- disc2(as.vector(data[, i]), k)
    }
  }
  data
}
```

Auxiliar function for disc.ef

```
disc2 <- function(x, k) {
  n = length(x)
  #print(n)
  ciclo = ceiling(n/k)
  #print(ciclo)
  y = x
}
```

```

for(i in 1:(k-1)){
  y[order(x)[((i-1)*ciclo+1):(i*ciclo)]]=i
}
y[order(x)[((k-1)*ciclo+1):n]]=k
#print(x)
return(y)
}

```

Script for Holte 1R discretization (from dprep R package)

#data: the data matrix

#convar: vector of continuous variables

example: disc.1r(data,1:1,6)

```
disc.1r <-function (data, convar, binsize = 6){
```

```
  data=as.matrix(data)
```

```
  if (dim(data)[2]==1) {
```

```
    stop ("You need class labels for your data.")
```

```
  }
```

```
  ncol = dim(data)[2]
```

```
  nrow = dim(data)[1]
```

```
  class = ncol
```

```
  for (i in convar) {
```

```
    id = matrix(1:nrow, ,1)
```

```
    discdata = cbind(id,data[,i])
```

```
    discdata = cbind(discdata,data[,class])
```

```
    discdata = discdata[order(discdata[,2]),]
```

```
    discrete = rep(0,nrow)
```

```
    discdata = cbind(discdata, discrete)
```

```
    modelist = list()
```

```
    kk = 1
```

```
    cc = 1
```

```
    j = 1
```

```
    nclass = length(levels(factor(data[,class])))
```

```
    maxclass = max(data[,class])
```

```
    freqc=rep(0,maxclass)
```

```
    sw = 0
```

```
    while (j <= nrow) {
```

```
      if (kk <= binsize) {
```

```
        discdata[j,4] = cc
```

```
        freqc[discdata[j,3]] = freqc[discdata[j,3]] + 1
```

```
        if (j == nrow) {
```

```
          modclass = which(freqc==max(freqc))
```

```
          modelist= unlist(list(modelist,
```

```
            modclass[1]))
```

```
        }
```

```
        j = j + 1
```

```
        kk = kk + 1
```

```
      } else {
```

```
        if (sw == 0) {
```

```
          modclass = which(freqc==max(freqc))
```

```
          modelist = unlist(list(modelist,modclass[1]))
```

```
          sw = 1
```

```
        }
```

```
        if (discdata[j,3] == modclass[1]) {
```

```
          discdata[j,4] = cc
```

```
          j = j + 1
```

```
        }else {
```

```
          cc = cc + 1
```

```
          discdata[j,4] = cc
```

```
          kk = 2
```

```
          freqc = rep(0, maxclass)
```

```
          freqc[discdata[j,3]] =
```

```

        freqc[discdata[j,3]] + 1
        if (j == nrow) {
            modelist=unlist(list(modelist, discdata[j,3]))
        }
        j = j + 1
        sw = 0
    }
}
}
gg = 1
for (m in 2:length(modelist)) {
    if (modelist[m] == modelist[m - 1]) {
        discdata[discdata[,4]== m, 4] = gg
    } else {
        gg = gg + 1
        discdata[discdata[,4]==m, 4] = gg
    }
}
data[,i] = discdata[order(discdata[,1]),4]
}
return (data)
}

```

Script for min max normalization (from dprep R package)

note: the method assumes that there are at least two columns with data and one last column with the class.

```

data <- read.table("week 04 - 1R_exemplo.txt")
data_3_col <- matrix(c(data[1:10,1:1], data[11:20,1:1], data[1:10,2:2]), 10, 3)
mmnorm(data_3_col, minval=0, maxval=1)
#These operations transform the data into [minval, mxval].
#Usually minval=0 and maxval=1.
#store all attributes of the original data

mmnorm <-function (data, minval=0, maxval=1) {
    d=dim(data)
    c=class(data)
    cnames=colnames(data)

    #remove classes from dataset
    classes=data[,d[2]]
    data=data[, -d[2]]

    minvect=apply(data, 2, min)
    maxvect=apply(data, 2, max)
    rangevect=maxvect-minvect
    zdata=scale(data, center=minvect, scale=rangevect)

    #remove attributes added by the function scale and turn resulting
    #vector back into a matrix with original dimensions
    #attributes(zdata)=NULL
    #zdata=matrix(zdata, dim(data)[1], dim(data)[2])

    newminvect=rep(minval, d[2]-1)
    newmaxvect=rep(maxval, d[2]-1)
    newrangevect=newmaxvect-newminvect
    zdata2=scale(zdata, center=FALSE, scale=(1/newrangevect))
    zdata3=zdata2+newminvect
    zdata3=cbind(zdata3, classes)
}

```

```
    if (c=="data.frame") zdata3=as.data.frame(zdata3)
    colnames(zdata3)=cnames
    return(zdata3)
}
```