R Programming, Week 4

Denis Vrdoljak

Matrixes

Matrixes

```
m <- matrix(x, nrow = 3, ncol = 3)
```
Create a matrix from x.

`m[2, ]` - Select a row

`m[ , 1]` - Select a column

`m[2, 3]` - Select an element

`t(m)`
Transpose

`m %*% n`
Matrix Multiplication

`solve(m, n)`
Find x in: m * x = n

Create a Matrix

- # matrix() function takes the data and the number of rows (nrow) and makes a matrix by filling down each column from the left to the right

- > matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9), nrow = 3)
- [,1] [,2] [,3]
- [1,] 1 4 7
- [2,] 2 5 8
- [3,] 3 6 9

- > matrix(1:8, ncol = 2)
- [,1] [,2]
- [1,] 1 5
- [2,] 2 6
- [3,] 3 7
- [4,] 4 8

Manipulate a Matrix

- > matrix1
- [,1] [,2] [,3]
- [1,] 1 4 7
- [2,] 2 5 8
- [3,] 3 6 9
-

- > matrix1[1, 3]
- [1] 7

- > matrix1[2,]
- [1] 2 5 8

- # the result is returned as a mathematical vector

Change a Matrix

- > m = matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9), nrow = 3)
- > m
- [,1] [,2] [,3]
- [1,] 1 4 7
- [2,] 2 5 8
- [3,] 3 6 9

- # To change use operator “=”
- > m[,2] = 1
- > m
- [,1] [,2] [,3]
- [1,] 1 1 7
- [2,] 2 1 8
- [3,] 3 1 9

- # algebra
- > m + 2
- [,1] [,2] [,3]
- [1,] 3 3 9
- [2,] 4 3 10
- [3,] 5 3 11

- # To remove a column use operator “-”
- > m[,-2]
- [,1] [,2]
- [1,] 1 7
- [2,] 2 8
- [3,] 3 9

Data Frames

Data Frames

```
df <- data.frame(x = 1:3, y = c('a', 'b', 'c'))
```

A special case of a list where all elements are the same length.

x	y
1	a
2	b
3	c

List subsetting

`df$x` 

`df[[2]]` 

Understanding a data frame

`View(df)` See the full data frame.

`head(df)` See the first 6 rows.

Matrix subsetting

`df[ , 2]` 

`df[2, ]` 

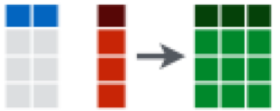
`df[2, 2]` 

`nrow(df)` Number of rows.

`ncol(df)` Number of columns.

`dim(df)` Number of columns and rows.

`cbind` - Bind columns.



`rbind` - Bind rows.

Data Frames

Data frame is used for storing data tables

> n = c(2, -3, 5)

> n[1] 2 -3 5

>

> b=c(TRUE, FALSE, TRUE)

> b[1] TRUE FALSE TRUE

>

> s = c("aa", "bb", "cc")

> s[1] "aa" "bb" "cc”

Create Data Frame from the List

```
> ll <- list (list(1:3), list(4:6), list(7:9))
> df <- as.data.frame(ll)
> df
 X1.3 X4.6 X7.9  # header contains the column names
1    1    4    7
2    2    5    8
3    3    6    9
```

Retrieving Data

```
#first row, second column
> df[1,2]
[1] 4

#number of rows
> nrow(df)
[1] 3

#number of columns
> ncol(df)
[1] 3
```

```
#preview – show top few rows
> head(df)
  X1.3 X4.6 X7.9
1    4    7
2    2    5    8
3    3    6    9
```

Data Frame Column Vector

```
# retrieve the second column vector
> df[[2]]
[1] 4 5 6

# retrieve the column vector by its name
> df[["X1.3"]]
[1] 1 2 3
# or
> df $X1.3
[1] 1 2 3
```

Data Frame Column Slice

```
# slice second column
> df[2]
  X4.6
1    4
2    5
3    6
```

```
# slice column by name
> df["X1.3"]
  X1.3
1    1
2    2
3    3
```

```
#slice two columns
> df[c("X1.3","X4.6")]
  X1.3   X4.6
1    1      4
2    2      5
3    3      6
```

Data Frame Row Slice

```
#numeric indexing
> df[3,]
  X1.3  X4.6  X7.9
3    3     6    9

> df[c(2,3),]
  X1.3  X4.6  X7.9
2    2     5    8
3    3     6    9
```

```
# name indexing
> df["1",]
  X1.3  X4.6  X7.9
1    1     4    7

> df[c("2","3"),]
  X1.3  X4.6  X7.9
2    2     5    8
3    3     6    9
```

Data Input

Environment

ls()	List all variable in environment
rm(x)	Remove x from the environment
rm(list=ls())	Remove all variables from environment

Example
> ls()
[1] “a”
[2] “b”
[3] “x”

>rm(b)
> ls()
[1] “a”
[3] “x”

getwd()

```
> getwd()              # get current working directory

> setwd("<new path>")   # set working directory
 > getwd()
[1] "/Users/mario/Downloads/final”
> setwd("/Users/mario/Documents/Santa Clara")

> getwd()[
1] "/Users/mario/Documents/Santa Clara"
```

Table File

```
# read data frame from the text file
> mydata=read.table(”indata.txt")
> mydata
   V1 V2 V3 V4
1 100 a1 b1 c1
2 200 a2 b2 c2
3 300 a3 b3 c3
4 400 a4 b4 c4
```

CSV File

```
# read CSV file -> mydata
> mydata=read.csv("datafile.csv")
> mydata
  X100 a1 b1 c1
1  200 a2 b2 c2
2  300 a3 b3 c3
```

Reading and Writing Data

Input	Ouput	Description
`df <- read.table('file.txt')`	`write.table(df, 'file.txt')`	Read and write a delimited text file.
`df <- read.csv('file.csv')`	`write.csv(df, 'file.csv')`	Read and write a comma separated value file. This is a special case of read.table/ write.table.
`load('file.RData')`	`save(df, file = 'file.Rdata')`	Read and write an R data file, a file type special for R.

Conditions								
	`a == b`	Are equal	`a > b`	Greater than	`a >= b`	Greater than or equal to	`is.na(a)`	Is missing
	`a != b`	Not equal	`a < b`	Less than	`a <= b`	Less than or equal to	`is.null(a)`	Is null

Excel File

```
# read.xl function (requires Perl runtime)
> library(gdata)                # load gdata package
> help(read.xls)                # documentation
> mydata = read.xls("mydata.xls")  # read from Excel file (first sheet)


# loadWorkbook function
> library(XLConnect)            # load XLConnect package
> wk = loadWorkbook("mydata.xls")
> df = readWorksheet(wk, sheet="Sheet1")
```

Minitab File

```
# for Minitab  Portable Worksheet format use read.mtp function
# function read.mtp
> library(foreign)                # load the foreign package
> help(read.mtp)                    # documentation
> mydata = read.mtp("mydata.mtp")  # read from .mtp file
```

SPSS File

```
# for SPSS format use function read.spss
> library(foreign)              # load the foreign package
> help(read.spss)                 # documentation
> mydata = read.spss("myfile", to.data.frame=TRUE)
```

Missing and Messy Dats

Finding Missing Values

```
# Finding missing values
> x1 <- c(1, 4, 3, NA, 7)
➢x2 <- c("a", "B", NA, "NA")

> x1
[1]  1  4  3 NA  7
x2
[1] "a"  "B"  NA   "NA"
```

```
# is NA?
is.na(x1)
[1] FALSE FALSE FALSE  TRUE FALSE
is.na(x2)
[1] FALSE FALSE  TRUE FALSE
# is NOT NA?
!is.na(x2)
[1]  TRUE  TRUE FALSE  TRUE
```

Missing Values in Dataframe

```
> ff=read.table("fillna.txt")
> ff
V1 V2 V3 V4
1 100  1  1  1
2 200  2  2  2
3 300  3  3  3
4 400  4  4  4
5 500 NA  5 NA
6 600 NA  6  6
```

```
> is.na(ff)
   V1    V2    V3    V4
[1,] FALSE FALSE FALSE FALSE
[2,] FALSE FALSE FALSE FALSE
[3,] FALSE FALSE FALSE FALSE
[4,] FALSE FALSE FALSE FALSE
[5,] FALSE  TRUE FALSE  TRUE
[6,] FALSE  TRUE FALSE FALSE
```

Cleaning Missing Values

```
> na.omit(ff)
V1 V2 V3 V4
1 100  1  1  1
2 200  2  2  2
3 300  3  3  3
4 400  4  4  4
```

```
> na.exclude(ff)
V1 V2 V3 V4
1 100  1  1  1
2 200  2  2  2
3 300  3  3  3
4 400  4  4  4
> na.fail(ff)
Error in na.fail.default(ff) :
missing values in object
```

```
> na.pass(ff)
V1 V2 V3 V4
1 100  1  1  1
2 200  2  2  2
3 300  3  3  3
4 400  4  4  4
5 500 NA  5 NA
6 600 NA  6  6
```

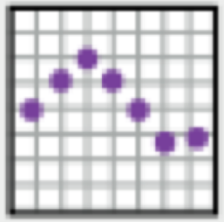
Replace NA with 0

```
> ff
V1 V2 V3 V4
1 100  1  1  1
2 200  2  2  2
3 300  3  3  3
4 400  4  4  4
5 500 NA  5 NA
6 600 NA  6  6
```

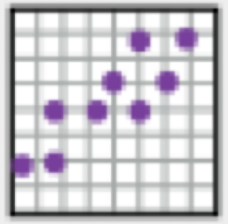
```
> ff[is.na(ff)] <- 0
> ff
V1 V2 V3 V4
1 100  1  1  1
2 200  2  2  2
3 300  3  3  3
4 400  4  4  4
5 500  0  5  0
6 600  0  6  6
```

Data Visualization and Graphics

Plotting



`plot(x)`
Values of x in order.



`plot(x, y)`
Values of x against y.

`hist(x)`
Histogram of x.

plot()
ggplot()
Interactive Plot

Thank You