**# 6\_Basic Instructions for R**

**Created by : Yogendra Patil**

**PhD, University of Alabama**

R package

<http://cran.r-project.org/>

R-studio

<http://www.rstudio.com/ide/download/>

R-studio keyboard shortcuts

<file:///C:/Program%20Files/RStudio/www/docs/keyboard.htm>

**Tutorial 2 Import Data**

How to import data from clipboard?

Create an excel spreadsheet, with variable names and observations.

Copy the whole observations (this gets saved in the cache!!).

Type in the following in R-studio Source (editor),

winners <- read.delim('clipboard') (Run) (See Workspace)

now type 🡪

winners (Ctrl+Enter) (See R-studio Console)

**Tutorial 3 Variables**

(Source/Editor)

Romeo <- 18 (use ‘<-‘ instead of ‘=’)

Juliet <- 42

Oscar <- 44

Sierra <- 17

Romeo (Ctrl + Enter and see the result in Console)

Oscar <- 38 (you can change the value)

Oscar

Juliet - Romeo

Romeo + Juliet + Oscar + Sierra

(Romeo + Juliet + Oscar + Sierra)/4

Average.Age <- (Romeo + Juliet + Oscar + Sierra)/4 (To store the result in a variable)

Average.Age

(Command Window/Console)

> source('~/.active-rstudio-document', echo=TRUE)

> Romeo <- 18

> Juliet <- 42

> Oscar <- 44

> Sierra <- 17

> Romeo

[1] 18

> Oscar <- 38

> Oscar

[1] 38

> Oscar

[1] 38

> Juliet - Romeo

[1] 24

> Romeo + Juliet + Oscar + Sierra

[1] 115

> (Romeo + Juliet + Oscar + Sierra)/4

[1] 28.75

> Average.Age <- (Romeo + Juliet + Oscar + Sierra)/4

> Average.Age

[1] 28.75

**Tutorial 4 Vectors**

c('Romeo','Juliet','Oscar','Sierra') % Create a vector with char variables  
% c() 🡪 actually is a **concatenate function**.

names <- c('Romeo','Juliet','Oscar','Sierra') % assign it to names

names (Ctrl + Enter)

> c('Romeo','Juliet','Oscar','Sierra')

[1] "Romeo" "Juliet" "Oscar" "Sierra"

> names <- c('Romeo','Juliet','Oscar','Sierra')

> names

[1] "Romeo" "Juliet" "Oscar" "Sierra"

c(1,2,7,2.5) % Create a vector with numerical variables

1:10 % create numbers from 1 to 10

7:14

-2:5

seq(from = 2, to = 20, by = 2) % create a sequence starting with 2 with an increment of % 2 upto 20.

rep(names,3) % repeat the char vector “names” in a row three times

> rep(names,3)

[1] "Romeo" "Juliet" "Oscar" "Sierra" "Romeo" "Juliet" "Oscar" "Sierra" "Romeo" "Juliet"

[11] "Oscar" "Sierra"

rep(names, each = 3) % repeat the char vector “names” in a row three times

> rep(names,each = 3)

[1] "Romeo" "Romeo" "Romeo" "Juliet" "Juliet" "Juliet" "Oscar" "Oscar" "Oscar" "Sierra"

[11] "Sierra" "Sierra"

rep(1:4,each = 3)

> rep(1:4,each = 3)

[1] 1 1 1 2 2 2 3 3 3 4 4 4

seq(1, 9, by = pi)

> seq(1, 9, by = pi)

[1] 1.000000 4.141593 7.283185

seq(0, 1, length.out = 11)

> seq(0, 1, length.out = 11)

[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

seq(0, 1, length.out = 13)

> seq(0, 1, length.out = 13)

[1] 0.00000000 0.08333333 0.16666667 0.25000000 0.33333333 0.41666667 0.50000000 0.58333333

[9] 0.66666667 0.75000000 0.83333333 0.91666667 1.00000000

**Tutorial 5 Getting Help**

?seq (Ctrl + Enter)

help(seq)

all R functions

<http://127.0.0.1:38804/help/library/base/html/00Index.html>

**Tutorial 6 Frozen Code**

**Tutorial 7+8 Vector Operations**

girls <- c('Juliet','Sierra')

boys<-c('Romeo','Oscar')

everyone <- c(girls,boys) % concatenate smaller vectors into one big!!

everyone

> girls <- c('Juliet','Sierra')

> boys<-c('Romeo','Oscar')

> everyone <- c(girls,boys)

> everyone

[1] "Juliet" "Sierra" "Romeo" "Oscar"

everyone[2] % show entry in the 2nd position

> everyone[2]

[1] "Sierra"

everyone[-2] % show all entries except the one in the 2nd position

> everyone[-2]

[1] "Juliet" "Romeo" "Oscar"

everyone[c(1,3)] % show entries in the 1st and 3rd position

> everyone[c(1,3)]

[1] "Juliet" "Romeo"

everyone[c(3,1)] % First show entry in the 3rd position and then in 1st position

> everyone[c(3,1)]

[1] "Romeo" "Juliet"

everyone[c(-1,-3)] % show all entries except the one in the 1st and 3rd position

> everyone[c(-1,-3)]

[1] "Sierra" "Oscar"

x <- c(2,4,6,8)

y <- c(7,8,3,1)

x + y

sum(x)

mean(y)

x \* 10

z <- y + 3

z

x \* y

> x <- c(2,4,6,8)

> y <- c(7,8,3,1)

> x + y

[1] 9 12 9 9

> sum(x)

[1] 20

> mean(y)

[1] 4.75

> x \* 10

[1] 20 40 60 80

> z <- y + 3

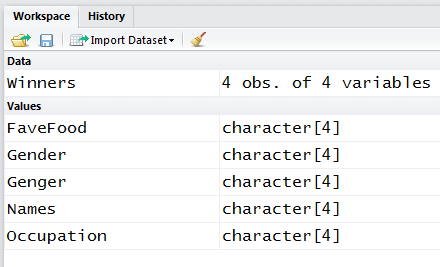
> z

[1] 10 11 6 4

> x \* y

[1] 14 32 18 8

**Tutorial 9 Making a Data Frame**



Names <- c('Romeo','Juliet','Oscar','Sierra')

Gender <- c('M','F','M','F')

Occupation <- c('Clown','Magician','Hobo','Unemployed')

FaveFood <- c('Tuna','Sausage','Melons','Kimchi')

Winners <- data.frame(Names,Gender,Occupation,FaveFood) % It’s just like structure % combining different variables together

Winners

> Winners

Names Gender Occupation FaveFood

1 Romeo M Clown Tuna

2 Juliet F Magician Sausage

3 Oscar M Hobo Melons

4 Sierra F Unemployed Kimchi

**Tutorial 10+11+12 More about Data Frames**

Type in the Console

?**mtcars** % **inbuilt data set** about cars % like Dr Adams course

head(mtcars) % get the first 6 entries in the data frame variable called ‘mtcars’

tail(mtcars) % get the last 6 entries in the data frame variable called ‘mtcars’

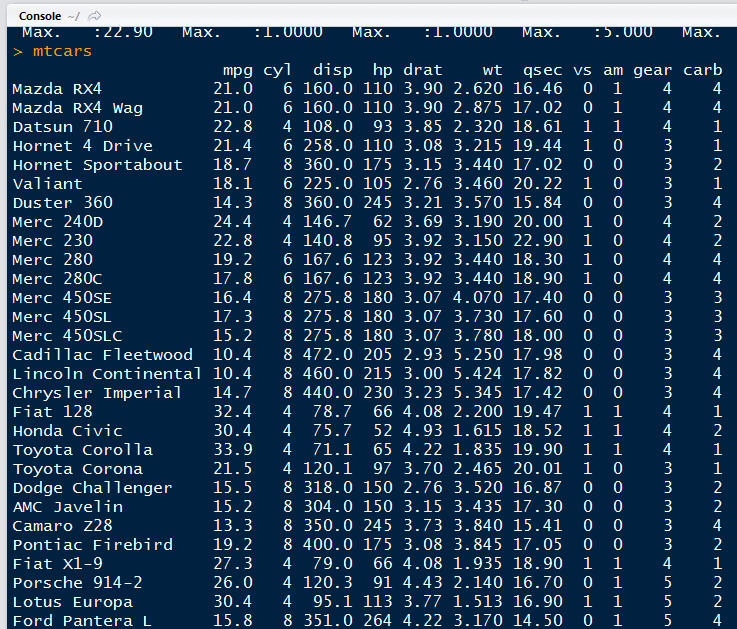
head(mtcars,10) % get the first 10 entries in the data frame variable called ‘mtcars’

str(mtcars) % get the **str**uctural information about the data frame variable called %‘mtcars’

summary(mtcars) % get the statistical summary of all the data frame variables

**Tutorial 11**

mtcars (Ctrl + Enter 🡪 C+E )



% say if you want to see the first row of the data

% (first entry = ROW, second entry = COLUMN)

mtcars[1,2] % get entry in the 1st row and 2nd column

> mtcars[1,2]

[1] 6

mtcars[1,] % get all entries in the first row

> mtcars[1,]

mpg cyl disp hp drat wt qsec vs am gear carb

Mazda RX4 21 6 160 110 3.9 2.62 16.46 0 1 4 4

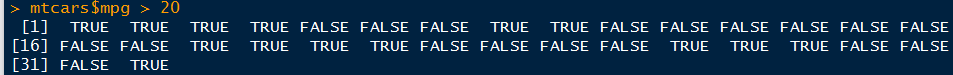
**Tutorial 12**

head(mtcars) % first 6 entries in ‘mtcars’

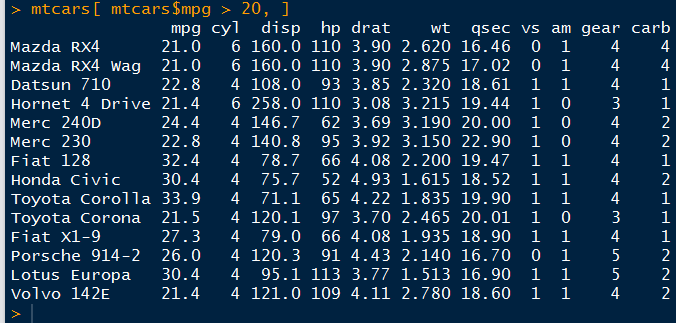
mtcars$mpg % get all the entries in the ‘**mpg**’ variables inside the ‘**mtcars**’



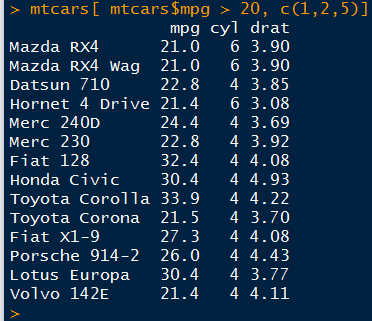
mtcars$mpg > 20 % Tell me **logically** which entries in ‘mpg’ is > 20



mtcars[ mtcars$mpg > 20, ] % Get the ‘rows’ that have ‘mpg’ > 20, **Remember**: all %columns here because [ROWS,COLUMNS] and “**SQUARE BRACKET**”



mtcars[ mtcars$mpg > 20, c(1,2,5)] % I want ‘rows’ that have ‘mpg’ > 20 and Columns % (1,2,5). REMEMBER : c is the vector concatenate function!!!



**Tutorial 13+14 Functions**

sum((1:10)^2)

> sum((1:10)^2)

[1] 385

sum\_sqr <- function(x) {sum(x^2)}

sum\_sqr(1:10)

> sum\_sqr <- function(x) {sum(x^2)}

> sum\_sqr(1:10)

[1] 385

**Tutorial 14**

% Functions with multiple Arguments

fun <- function(x,y) {

out <- x^2;

out <- out + y;

out % **R function can return only one value and it should be denoted at the end**

}

fun(4,2)

> fun(1,2)

[1] 3

> fun(4,2)

[1] 18

% How to join two strings together

**'%p%'** <- function(x,y) { **paste**( x,y, sep = '') }

first\_name**%p%**last\_name

> first\_name%p%last\_name

[1] "GordonDavis"

**Tutorial 15 Packages**

install.packages('plyr')

library('plyr')

head(mtcars)

ddply(mtcars,c('cyl'),summarise, max = max(disp), min = min('disp') )

> ddply(mtcars,c('cyl'),summarise, max = max(disp), min = min('disp') )

cyl max min

1 4 146.7 disp

2 6 258.0 disp

3 8 472.0 disp

**Tutorial 15 Package**

**Tutorial 16 Merging Data Frames**

df1 <- data.frame(

key = c('aaa','bbb','ccc'),

field1 = c(3,1,4)

)

df2 <- data.frame(

key = c('aaa','ccc','eee'),

field2 = c(2,1,7)

)

df1

df2

merge(df1,df2,by = c('key'))

> df1

key field1

1 aaa 3

2 bbb 1

3 ccc 4

> df2

key field2

1 aaa 2

2 ccc 1

3 eee 7

> merge(df1,df2)

key field1 field2

1 aaa 3 2

2 ccc 4 1

> merge(df1,df2,by = c('key'))

key field1 field2

1 aaa 3 2

2 ccc 4 1

merge(df1,df2,by = c('key'), all.x = TRUE)

> merge(df1,df2,by = c('key'), all.x = TRUE)

key field1 field2

1 aaa 3 2

2 bbb 1 NA

3 ccc 4 1

merge(df1,df2,by = c('key'), all = TRUE)

> merge(df1,df2,by = c('key'), all = TRUE)

key field1 field2

1 aaa 3 2

2 bbb 1 NA

3 ccc 4 1

4 eee NA 7

**Tutorial 17 Examples**

df1 <- da