



JIGSAW ACADEMY  
THE ONLINE SCHOOL OF ANALYTICS

##-----Introduction to R-----#

```
setwd("path to data in your system")
```

```
#Agenda:Variables,vector,selection of element of vectors,entering data into  
dataframe
```

```
#Use # for writing a comment
```

```
p<-5
```

```
q<-p+5
```

```
q
```

```
#Can do arithmetic operations
```

```
pi
```

```
sqrt(25)
```

```
2^2+5
```

```
p<-5
```

```
# Items in R are stored as objects
```

```
#Setting variables
```

```
#Assignment of values to a variable "<-"
```

```
x<-5
```

```
# other forms of assignment operator
```

```
fun=6
```

```
fun<-6
```

```
#Over writing a variable
```

```
x<-"Jigsaw"
```

```
#deleting the variables
```

```
rm(x)
```

```
x
```

```
rm(foo,fun)
```

#Programatically figuring out objects

ls()

data()

?iris

# How to create a vector?

# It consists of either numbers,strings, or logical values but not all of them together.

# It contains only 1 type of class

# Type of variables : Class of the variables (Integer,character,logical)

# c is the combining operator

x<-5 #it is a numeric vector of 1 element

x<-c('1','2','-5','6') # numeric vector of 4 elements

x

a<-3

a1<-c(a,a\*4,-7\*a)

a1

A<-c(1,2,3,NA)

A

class(A)

string<-c("1","2","2","3","4")

#class gives the data type of vector

class(string)

B<-c('a','b','c',NA)

B

class(B)

sp<-c(TRUE,FALSE,TRUE) #class is

```
class(sp)
```

```
a1<-c(1,1,1)
```

```
a2<-c(2,2,2)
```

```
a1+a2
```

```
#Vectors shouldn't have mixed type of data
```

```
p<-c(1,2,"g")
```

```
class(p)
```

```
p<-c(TRUE,FALSE,"G")
```

```
p<-c(TRUE,FALSE,3)
```

```
class(c(TRUE,FALSE,1))
```

```
x<-c(1,2,3,4)
```

```
class(x)
```

```
x1<-c(1.2,2.4,3.5,4.5)
```

```
class(x1)
```

```
#Vector,working with vectors
```

```
# Vector : Most Simplest structure in R .Only one data type
```

```
num<-c("a","b","c","d","e")
```

```
num
```

```
num[1]
```

```
num[4]
```

```
num[-1]
```

```
num[1:2]
```

```
#assigning names
```

```
names(num)<-c("x1","x2","x3","x4","x5")
```

```
names(num)[4]
```

```
num["x4"]
```

```
num[c("x1","x2")]
```

```
length(num)
```

```
#dataframes
```

```
#Each column can be a different Data types.
```

```
#Consider the following vectors :
```

```
product=c("Bag","shoes","belt","belt")
```

```
total_price=c(500,1000,150,10000)
```

```
color=c("Blue","red","red","blue")
```

```
quantity=c(5,2,3,4)
```

```
product_details <-
```

```
data.frame(product,total_price,color,quantity,stringsAsFactors=FALSE)
```

```
product_details
```

```
class(product_details)
```

```
str(product_details)
```

```
product_details <- data.frame(product,total_price,color,quantity)
```

```
product_details[,2]
```

```
product_details[, "total_price"]
```

```
product_details[2,]
```

```
product_details[2,2]
```

```
head(product_details,2)
tail(product_details,2)
```

```
factors
# data structure to store categorical variables.
gender<- c(1,2,1,2,1,2,1,2)
gender <- factor(gender, levels = c(1,2), labels = c("Male","Female"))
table(gender)
```

```
x <- c("yes","no","no","yes","no")
y <- as.factor(x)
class(y)
table(y)
```

```
##Importing tabular data##
```

```
#Checks needed to ensure that data is imported correctly
#1. Delimiter in the file
#2. How missing values are populated in the data
```

```
import1<-read.table("F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Intro to
R\\sample2.csv",sep="," ,header = TRUE)
```

```
summary(import1) #Focus on the missing values
str(import1) #Are column data types correct
```

```
import2<-read.table("F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Intro to
R\\sample1.txt")#Why error
```

```
import2<-read.table("F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Intro to
R\\sample1.txt",sep="\t")
```

```
import2#What is wrong?
```

```
import2<-read.table("F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Intro to  
R\\sample1.txt",sep="\t",header=TRUE)
```

```
import2
```

```
summary(import2)
```

```
str(import2)
```

```
import2<-read.table("F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Intro to  
R\\sample1.txt",sep="\t",header=TRUE,na.strings = c(NA,"Missing",""))
```

```
summary(import2)
```

```
str(import2)
```

```
class(import1)
```

```
class(import2)
```

##Data import is considered successfull: 1.Delimiters are identified correctly,  
2.Missing values in the data are recognized as missing (NA)

#Getting tables from web pages

#Many web pages contain tabular data as comma separated or tab separated

#UCI Abalone data set

```
web2<-read.csv("https://archive.ics.uci.edu/ml/machine-learning-  
databases/abalone/abalone.data",header=FALSE)
```

```
web2
```

#Lists : Recursive vectors. Can handle different data types

```
my.list <- list( name = c("Robert", "Emma"), age = c (65, 54,43),retired = c (TRUE,  
FALSE))
```

```
my.list
```

```
my.list$age
```

```
my.list["age"]
```

```
class(my.list$age)  
class(my.list["age"])
```

```
my.list[["age"]][2]
```

```
my.list[[3]]  
my.list [[3]][2]
```

```
list(c("Robert", "Emma"), age = c (65, 54),  
     retired = c (TRUE, FALSE))
```

#Why study lists, results of machine learning algorithms/statistical modelling algorithms is a list object

```
model1<-lm(data1$MYCT~data1$MMIN)
```

```
mode(model1)
```

```
model1$coefficients  
model1$residuals #etc
```

#Importing non tabular data will also result in the creation of list

#Sometimes data is not available in tabular formats: csv, sql servers#

#Particularly when working with API's

#Glassdoor API, <https://www.glassdoor.co.in/developer/index.htm>

```
install.packages("jsonlite")
```

```
library(jsonlite)
```

```
web1<-
```

```
fromJSON("http://api.glassdoor.com/api/api.htm?v=1&format=json&t.p=47699  
&t.k=g9GdVHIQ1eM&action=employers&q=pharmaceuticals&userip=192.168.4  
3.42&useragent=Mozilla/%2F4.0")
```

```
class(web1)
```

```
str(web1)
```

#One can save the web page locally and still read in the tables

```
web4<-readHTMLTable("F:\\Work\\Jigsaw Academy\\Corporate  
Trainings\\Intro to R\\Historical Consumer Price Index (CPI).html")
```

```
web4
```

```
class(web4)
```

#If analysis on this data has to be done then it should be converted to a dataframe

```
web4<-as.data.frame(web4)
```

```
class(web4)
```

```
head(web4)#What is wrong?
```

```
write.csv(web4,"F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Intro to  
R\\cpi.csv",row.names = F)
```

```
web5<-read.csv("F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Intro to  
R\\cpi.csv")
```

```
head(web5)#Need to skip the first line
```

```
web5<-read.csv("F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Intro to  
R\\cpi.csv",skip=1)
```

```
head(web5)
```

#Loading packages: some functionalities are not available under the hood.

```
library(ggplot2)
```

```
install.packages("arules")
```





JIGSAW ACADEMY  
THE ONLINE SCHOOL OF ANALYTICS

```
library(arules)
data(AdultUCI)
Adult<-AdultUCI
Adult
?AdultUCI
save.image()
```