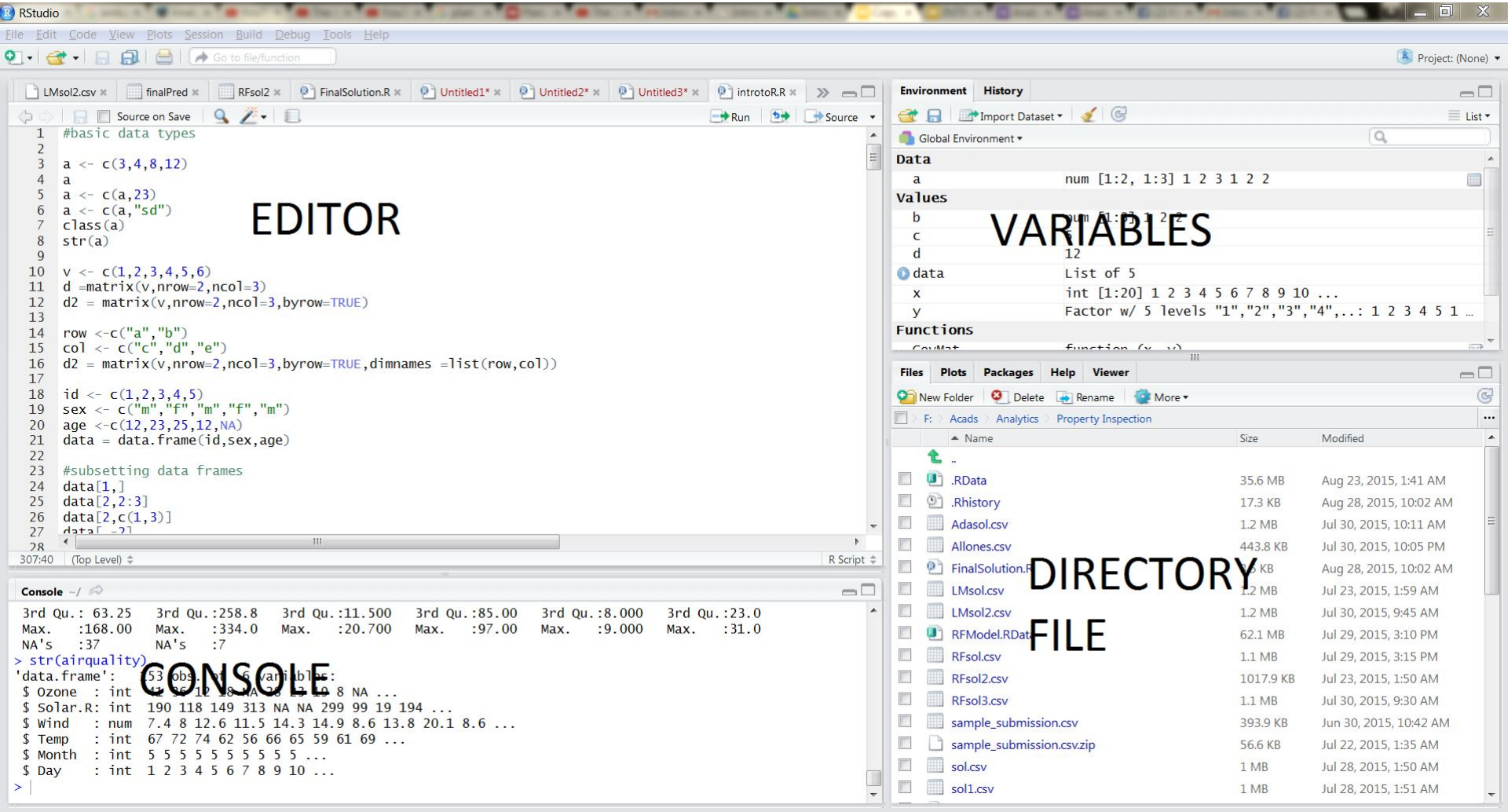
INTRODUCTION TO R

INTRODUCTION TO R



SETTING DIRECTORY

- **getwd()**

  lists the current directory

- **setwd()**

  sets the new directory

  >>setwd("C://users/guest/documents")

IMPORTING DATASETS

- Data is most likely to be in csv format
- ***read.csv()***

  >> read.csv("FlightData.csv")

- Other input functions include read.table() and read.xlsx()
- Use ***?read.csv()*** to check for other options while importing
- write into csv files using the function ***write.csv()***

  >> write.csv("mySubmission.csv",myDataFrame)

BASIC DATA INSPECTION

- **names(dataset)** - lists the variable names in the dataset
- **head(dataset, x)** - lists the top x elements of the data frame
- **tail(dataset,x)** - lists the bottom x elements of the data frame
- **summary(dataset)** - Basic summary of the each of the columns
- **str(dataset)** - Gives the structure of all the columns of the dataset
- **levels(variable)** - lists the levels of factor variable of the dataset

MISSING VALUES

- Missing values represented as NA in R
- **is.na()** - checks for missing values for a particular variable
- **na.omit(dataset)** - Returns a clean dataset with no missing values
- **na.rm()** - Available as an option in most built-in functions. Disregards rows with missing values while computing the function value

>> mean(data , na.rm = TRUE)

BASIC OPERATIONS

- *> 1 + 1 # Simple Arithmetic*
- O/P : 2
- *> 2 + 3 * 4 # Operator precedence*
- O/P : 14
- *> 3 ^ 2 # Exponentiation*
- O/P : 9
- *> exp(1) # Basic mathematical functions are*
- 2.718282
- *> sqrt(10)*
- 3.162278

BASIC BUILT IN FUNCTIONS

- **rep(a,x)** - Returns a vector of size x filled with a's
- **seq(x,y,step)** - Returns a vector with values starting from x till y with the step being the difference between consecutive terms
- **substr(x, start,stop)** - Extract segments defined by start and stop from characters
- **toupper()** - convert text to uppercase. Similarly tolower()
- **sub(pattern, replacement, x)** - Find pattern in x and replace it with replacement
- **subset(data,condition)** - returns the subset of the dataset which satisfies the condition

HELP AND EXAMPLES

- **Getting help**

  > help.start()

  >help(mean)

  ?mean

  ??mean

- **Any Examples**

  > example(mean)

DATA TYPES

- VECTORS
- MATRIX
- ARRAY
- DATA FRAMES
- LIST
- FACTORS

VECTORS

- The most basic data type
- Can contain only objects of the same class
- c() can be used to create vector of objects

```
> x <- c(0.5,0.6)          # Numeric Vector

> x<- c (TRUE, FALSE )     # Logical Vector

> x<- c (“a”, “b”, “c”)    # Character

> x <- 9:29                # Integer

> x <- c(0+2i , 2+4i)      # Complex
```

VECTORS

The vector() function is also used to create vectors

```
> x <- vector("numeric", length=10)

> x

[1] 0 0 0 0 0 0 0 0 0 0
```

EXPLICIT COERCION

Objects can be explicitly coerced from one class to another using the as.* functions, if available.

```
> x <- 0:6

> class(x)

[1] "integer"

> as.numeric(x)

[1] 0 1 2 3 4 5 6
```

EXPLICIT COERCION

as.logical(x)

[1] FALSE TRUE TRUE TRUE TRUE TRUE TRUE

> as.character(x)

[1] "0" "1" "2" "3" "4" "5" "6"

MATRICES

Matrices are vectors with a dimension attribute. The dimension attribute is itself an integer vector of length 2(nrow,ncol). Matrices are created using the ***matrix()*** function

```
> m <- matrix(1:6,nrow = 2, ncol = 3)

> m
```

	[,1]	[,2]	[,3]
[1,]	*1*	*3*	*5*
[2,]	*2*	*4*	*6*

CBIND() AND RBIND()

Matrices can be created by binding two vectors along the row or the column

> x <- 1:3 *> y <- 10:12*

> cbind(x,y)

```
     [,1] [,2]
[1,]    1   10
[2,]    2   11
[3,]    3   12
```

> rbind(x,y)

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]   10   11   12
```

LIST

A list is a collection of vectors

- mixed data types
- varying length objects are allowed

list() function can be used to create lists

> students <- c (“Mary”, “Bob” ,” John”)

> ages <- c(14, 15, 18)

> classroom <- list (students, age)

DATA FRAMES

A Data frame is similar to a matrix in the sense that it has rows and columns but differs from it in that the columns can be of different types.

> df1 <- data.frame(S = c ("Mary", "Raj", "Salman") ,test1=c(50,40,50))

> df1

	S	test1
1	Mary	50
2	Raj	40
3	Salman	50

SUBSETTING LISTS

- x<-list(foo=1:4,bar=0.6)

```
>x[1]
    $foo
    [1] 1 2 3 4
>x[[1]]
    [1] 1 2 3 4
>x$bar
    [1] 0.6
>x['bar']
  $bar [1] 0.6
```

CONTROL STRUCTURES

- **if-else**
- **for**
- **while**
- **switch**
- **ifelse**

for and **while** are rarely used as most of the functions like tapply(),lapply() etc. can get the job done and that too in much less time.

SUBSETTING

- There are number of operators that can be used to extract subsets of R objects.
- [always returns an object of the same class as the original; can be used to select more than one element (there is one exception).
- [[is used to extract elements of a list or a data frame; it can only be used to extract a single element and the class of the returned object will not necessarily be a list or data frame.
- $ is used to extract elements of a list or data frame by name; semantics are similar to that of [[.

SUBSETTING EXAMPLE

```
> x <- c("a", "b", "c", "c", "d", "a")

> x[1]

[1] "a"

> x[1:4]

[1] "a" "b" "c" "c"

> x[x > "a"]
```

SUBSETTING EXAMPLE

```
> x[x > "a"]

[1] "b" "c" "c" "d"

> u <- x > "a"

> u

 [1] FALSE TRUE TRUE TRUE TRUE FALSE

> x[u]

 [1] "b" "c" "c" "d"
```

SUBSETTING LISTS

```
> x <- list(foo = 1:4, bar = 0.6)

> x[1]

$foo

[1] 1 2 3 4

> x[[1]]

[1] 1 2 3 4
```

SUBSETTING LISTS (contd….)

```
> x$bar

[1] 0.6

> x[["bar"]]

[1] 0.6

 > x["bar"]

 $bar [1] 0.6
```

CONTROL STRUCTURES

- **if-else**
- **for**
- **while**
- **switch**
- **ifelse**

for and **while** are rarely used as most of the times functions like **lapply()** and **tapply()** etc. can get the job done and even that in much less time.

LAPPLY()

- When you want to apply a function to each element of a list/vector
- Always returns list as output. Input may or maynot be list.
- This is the workhorse of other apply functions. peel back their code and you will often find lapply underneath
- Access individual member of the output list using [[]] operator

```
> roundSepal.Length = lapply(iris$Sepal.Length,round)

> roundSepal.Length[[3]]
```

TAPPLY()

- Splits the datasets according to subsets and applies a function over the subset
- Subsetting the factor may or maynot be part of dataframe
- Subsetting must have same number of rows as that of the dataset
- Subsetting vector can be categorical as well as numerical

```
>tapply(airquality$Wind,airquality$Month,mean)
```

SPLIT()

- splits the dataset to subsets based on the factor or list of the factors

  > a->split(dataset,factor)

- The newly created subsets can be accessed by using $ operator.

  > a$subsetname

  > split(airquality,airquality$Month)

OTHER IMPORTANT FUNCTIONS

- **colMeans()**- means along the column
- **rowMeans()**- means along the row
- **table(x,y)** - creates confusion matrix for x and y vectors
- **prop.table(x)** - outputs the proportion of individual elements of the vector x
- **which()** - true indices of the logical object

USER DEFINED FUNCTIONS

- R allows user to define functions following a syntax as follows :

```
 newFunction <- function(arg1, arg2, ...)
{
 ...code…
 return (output)
}
```

- Objects defined in the function are local variables and can only be accessed from within the function.
- To use a function to modify global variables , use <<- assignment operator
- Scoping rules also play an important role in determining variable values.

BASIC PLOTTING FUNCTIONS

- plot()
- hist()
- boxplot()
- heatmap()
- barplot()

Advanced plotting can be done by using ggplot from ggplot2 library

EXERCISE

- What is the mean of ‘Sepal.Length’ for the species virginica?
- Write R code that returns a vector of the means of the variables 'Sepal.Length', 'Sepal.Width' and 'Petal.Width'
- How many samples of setosa species have Petal.Width > 0.2 and Petal.Length > 1.4?
- What is mean Sepal length for samples of Virginica species whose Petal width >2.1 and Sepal width > 2.8?
- Which sample belonging to the vernicolor species and having Petal length >4.4 and has the largest Petal width ?
- For the airquality dataset, find the mean for days with 3 days gap in between them. (Your final answer should have the 3 mean values)

EXERCISES

- Calculate the average miles per gallon(mpg) by number of cylinders in the car(cyl)
- What is the absolute difference between the average horsepower of 4-cylinder cars and the average horsepower of 8-cylinder cars

THANK YOU