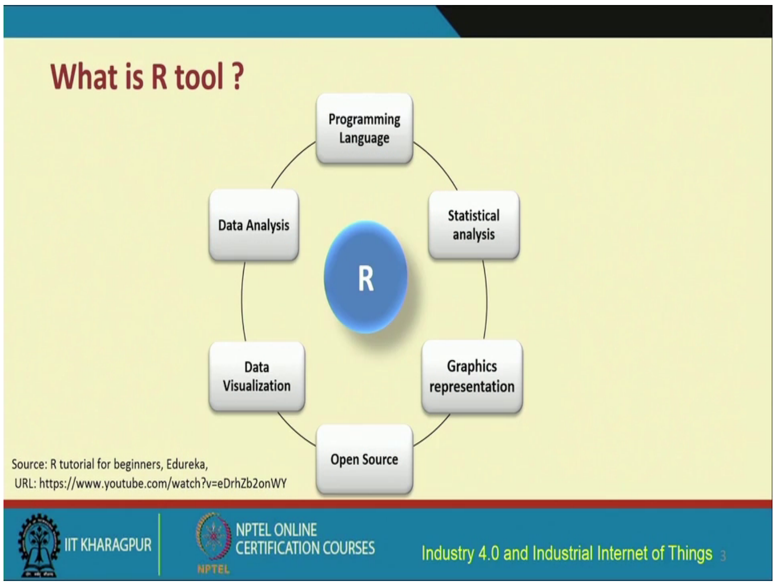
# Introduction to Industry 4.0 and Industrial Internet of Things

**Prof. Sudip Misra**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Kharagpur**

## Lecture - 42
## IIOT Analytics and Data Management: Tutorial for R & Julia Programming

In this particular lecture, I am going to show you two tools platforms in fact, which could be used for IIoT analytics. One of these is popularly known as the R, the R tool which offers it is a comprehensive tool offering different platforms you know has its own language for implementation and so on. So, R is basically a tool that is widely used for analytics. So R and another language which is known as Julia, Julia is also a very popular high level language which could be used for implementing these analytics solutions. So, both R and Julia are quite popular in the IIoT analytics scenarios and their implementation.

(Refer Slide Time: 01:23)

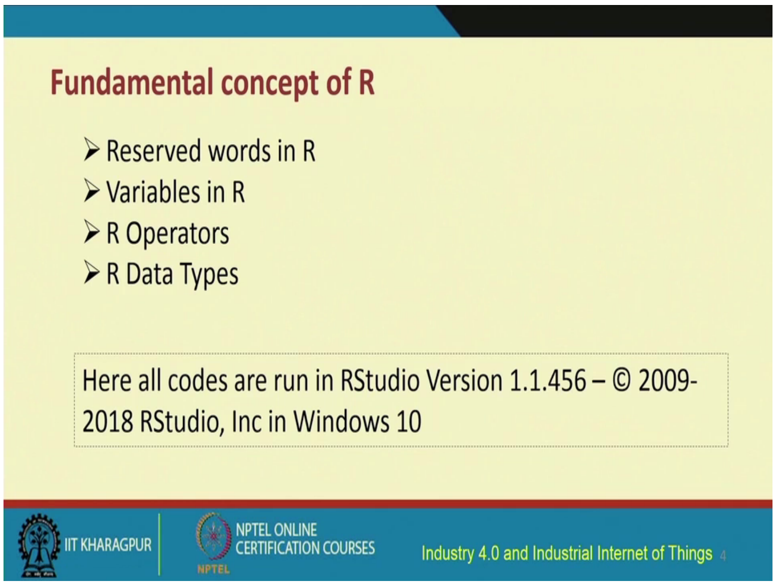


### What is R tool ?

Source: R tutorial for beginners, Edureka,
URL: https://www.youtube.com/watch?v=eDrhZb2onWY

So, let us first look at R, what is R and the R programming. So, R it is a tool; R is a tool which offers different features; different features of programming you know so it offers a programming language, it offers statistical analysis platform, it has graphical representation platform, data visualization, open source its R is open source and has different data analytics features. So, these are the different features of R.

(Refer Slide Time: 01:55)



## Fundamental concept of R

- Reserved words in R
- Variables in R
- R Operators
- R Data Types

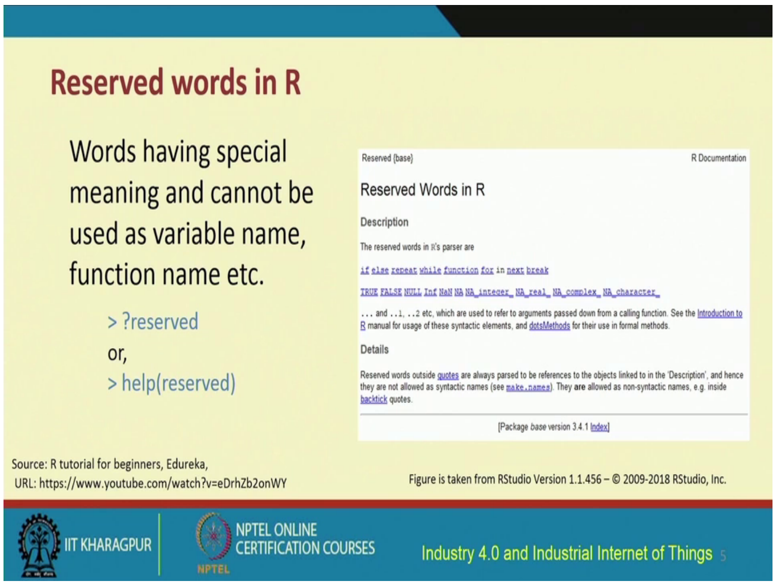
Here all codes are run in RStudio Version 1.1.456 – © 2009-2018 RStudio, Inc in Windows 10

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Industry 4.0 and Industrial Internet of Things 4

And R basically has different it is you know the programming environment in R supports different features. So, the fundamental concepts are like this that there are certain reserved words in R, their certain variables that can be defined in R, there are R operators, R data types and these I am going to show you shortly and what are these and you know is just a simple instance of all of these is what I am going to show you.

So, here basically everything will be shown in the R studio. R studio basically has support for this R programming platform. So, I am going to show you this thing in the R studio, but let us first proceed further and see that what are these reserved the words in R.

(Refer Slide Time: 02:39)



## Reserved words in R

Words having special meaning and cannot be used as variable name, function name etc.

> ?reserved

or,

> help(reserved)

Reserved {base} R Documentation

Reserved Words in R

Description

The reserved words in R's parser are

`if else repeat while function for in next break`

`TRUE FALSE NULL Inf NaN NA NA_integer_ NA_real_ NA_complex_ NA_character_`

`...` and `..1`, `..2` etc, which are used to refer to arguments passed down from a calling function. See the Introduction to R manual for usage of these syntactic elements, and dotsMethods for their use in formal methods.

Details

Reserved words outside quotes are always parsed to be references to the objects linked to in the 'Description', and hence they are not allowed as syntactic names (see `make.names`). They **are** allowed as non-syntactic names, e.g. inside backtick quotes.

[Package *base* version 3.4.1 Index]

Source: R tutorial for beginners, Edureka,
URL: https://www.youtube.com/watch?v=eDrhZb2onWY

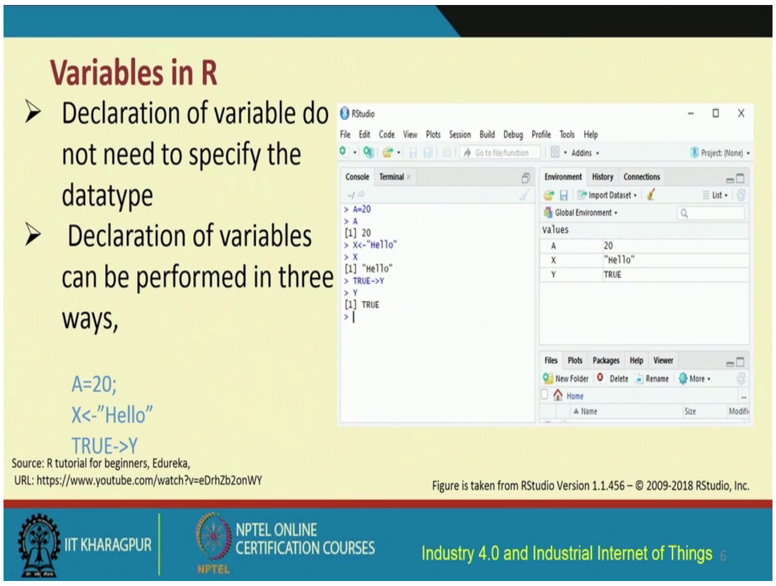
Figure is taken from RStudio Version 1.1.456 – © 2009-2018 RStudio, Inc.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Industry 4.0 and Industrial Internet of Things

So, these you know there is this reserved you know words which have some special meaning and which cannot be used as a variable name. For example, this reserved itself. So, if you basically type in the R studio if you type in question mark reserved or help within parenthesis reserved, this is what you are going to get you are going to get these reserved words that are supported in R.

So, these reserved wards are like if else, repeat, while, function, for, then next, break, true, false etcetera. So, these are some of these different reserved words and their corresponding details are also available. So, if you just type in this in the R studio this reserve then you would be able to see this.

(Refer Slide Time: 03:39)



## Variables in R

- Declaration of variable do not need to specify the datatype
- Declaration of variables can be performed in three ways,

```
A=20;
X<-"Hello"
TRUE->Y
```

```
> A=20
> A
[1] 20
> X<-"Hello"
> X
[1] "Hello"
> TRUE->Y
> Y
[1] TRUE
>
```

| Values | |
|---|---|
| A | 20 |
| X | "Hello" |
| Y | TRUE |

Source: R tutorial for beginners, Edureka, URL: https://www.youtube.com/watch?v=eDrhZb2onWY

Figure is taken from RStudio Version 1.1.456 – © 2009-2018 RStudio, Inc.

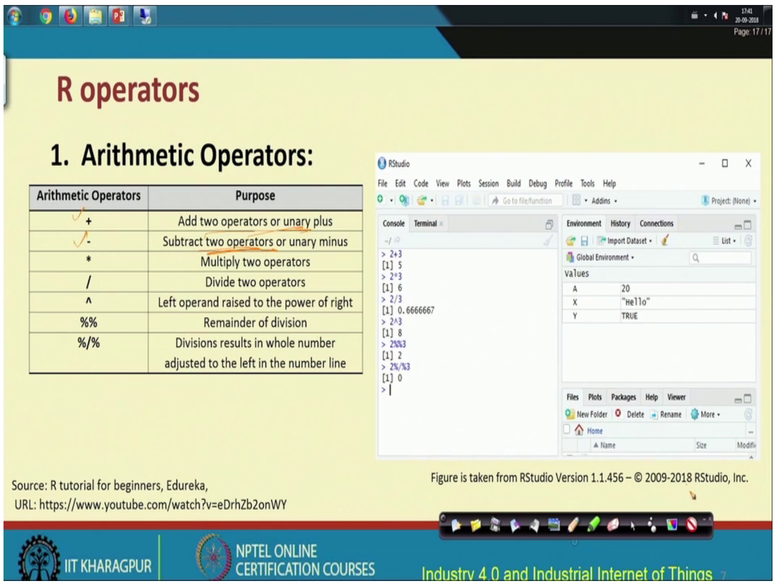
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Industry 4.0 and Industrial Internet of Things 6

Then what is a variable? Variables are basically like the variables in any programming languages, these are basically declarations that I that will have to be done and these will be assigned certain values right. So, these variables will have to be declared for example, these are some of these variables A is a variable, X is a variable, Y is a variable. So, these variables they could be assigned different values.

So, for example, A equal to 20, 20 is the value that is assigned to this variable A, then hello is a string which is assigned this to the value of this is a value that is assigned to this variable X and true is a value that is assigned to this variable Y so these are the different types of variables and how they can be used in the or they can be declared how these variables can be declared in the R studio platform. So, these are basically the essential features of this R programming language.

So, as you can see over here this is just as snap shot that is taken screenshot that is taken from the R studio and as you can see over here this is how you will make certain declarations A equal to 20 and now then if you want to print A you get this value 20. Similarly you declare this variable X to be hello and if you print X then you are going to get this value the result is going to be hello, then true Y basically if you print Y then you are going to get this value true because that is what this valuable this variable Y is going to have this value and also in this R studio platform which I am going to show you shortly you will be able to see these different there is a separate panel over which you can see all these global environment values of these different variables A, X and Y and their corresponding values shown.

(Refer Slide Time: 05:45)



## R operators

### 1. Arithmetic Operators:

| Arithmetic Operators | Purpose |
|---|---|
| + | Add two operators or unary plus |
| - | Subtract two operators or unary minus |
| * | Multiply two operators |
| / | Divide two operators |
| ^ | Left operand raised to the power of right |
| %% | Remainder of division |
| %/% | Divisions results in whole number adjusted to the left in the number line |

```
> 2+3
[1] 5
> 2*3
[1] 6
> 2/3
[1] 0.6666667
> 2^3
[1] 8
> 2%%3
[1] 2
> 2%/%3
[1] 0
>
```

| Values | |
|---|---|
| A | 20 |
| X | "Hello" |
| Y | TRUE |

Source: R tutorial for beginners, Edureka,
URL: https://www.youtube.com/watch?v=eDrhZb2onWY

Figure is taken from RStudio Version 1.1.456 – © 2009-2018 RStudio, Inc.

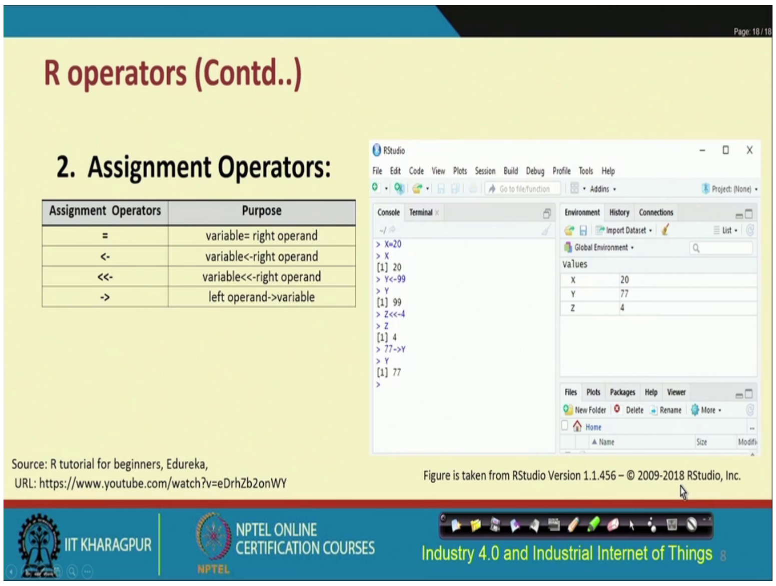
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal
Environment History Connections
Import Dataset
Global Environment
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home
Page: 17 / 17
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Industry 4.0 and Industrial Internet of Things

Next I am going to show you how different other things are done. For example, the use of different parameters sorry operators in R different operators such as arithmetic operators are possible operators such as different other different arithmetic operators such as the plus operator which will do the unary plus; unary plus, this minus will do the subtraction and then star is for multiplication, slash is for division then this cap is basically for power basically two to the power x will you know. So, you have to use this power with the cap then you have double percentage which is basically the remainder of a division and this percentage slash percentage basically this is the division resulting in a whole number adjusted to the left in the number line.

So, these are some of these different arithmetic operators that are supported in R and this is a screen shot that is taken from R studio as you can see over here it will show you the different arithmetic operators 2 plus 3 and then if you hit enter you will get this result 5 right so this is just a this is just a unary plus or rather not the unary plus, but the binary. So, here actually what you are doing is you are just adding these two different operands with the help of this plus operator and you are getting this result 5, similarly the star is for multiplications 2 into 3 so you get the result then this will be divided by 2 divided by

3 in floating point and this is 2 to the power 3 right. So, 2 cube will be 8 and so on. So, these are these different operators that are supported in R.

(Refer Slide Time: 07:49)



Page: 18 / 18

## R operators (Contd..)

### 2. Assignment Operators:

| Assignment Operators | Purpose |
|---|---|
| = | variable= right operand |
| <- | variable<-right operand |
| <<- | variable<<-right operand |
| -> | left operand->variable |

```
> X=20
> X
[1] 20
> Y<-99
> Y
[1] 99
> Z<<-4
> Z
[1] 4
> 77->Y
> Y
[1] 77
>
```

Values

| | |
|---|---|
| X | 20 |
| Y | 77 |
| Z | 4 |

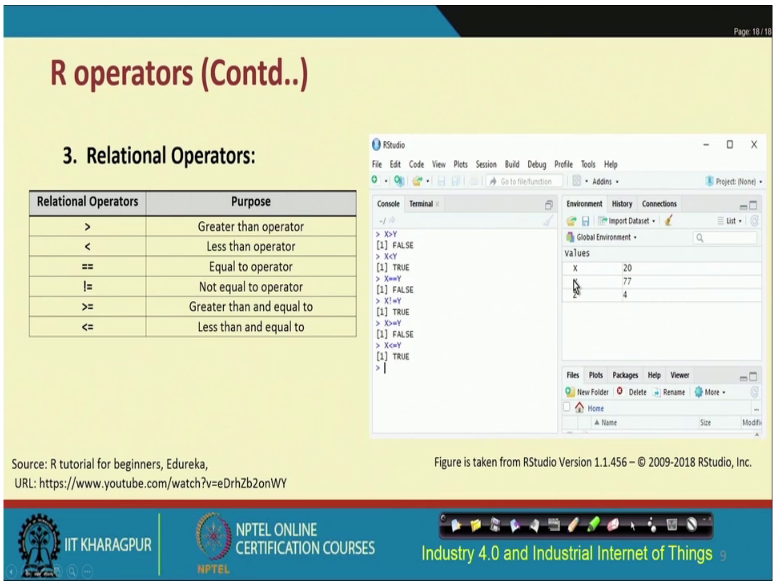
Source: R tutorial for beginners, Edureka,
URL: https://www.youtube.com/watch?v=eDrhZb2onWY

Figure is taken from RStudio Version 1.1.456 – © 2009-2018 RStudio, Inc.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES NPTEL Industry 4.0 and Industrial Internet of Things 8

There are other operators, assignment operators. Operators like equal to, then you have this less than signed hyphen, then you have this double less than hyphen and hyphen greater than and their corresponding purposes are also mentioned over here. So, this is just an example likewise like the previous slides I have even given you an example of all of these different other assignment operators like the equal to etcetera. So, X equal to ill just give you an example one of these X equal to 20 basically assigns this value 20 to X and then if you print the value of X you see that you get the result as 20 and these the use of these different other operators are also given over here and there examples given over here so you may go through and try to understand how they work.

(Refer Slide Time: 08:47)



## R operators (Contd..)

### 3. Relational Operators:

| Relational Operators | Purpose |
|---|---|
| > | Greater than operator |
| < | Less than operator |
| == | Equal to operator |
| != | Not equal to operator |
| >= | Greater than and equal to |
| <= | Less than and equal to |

```
> X>Y
[1] FALSE
> X<Y
[1] TRUE
> X==Y
[1] FALSE
> X!=Y
[1] TRUE
> X>=Y
[1] FALSE
> X<=Y
[1] TRUE
>
```

```
Values
X    20
Y    77
Z    4
```

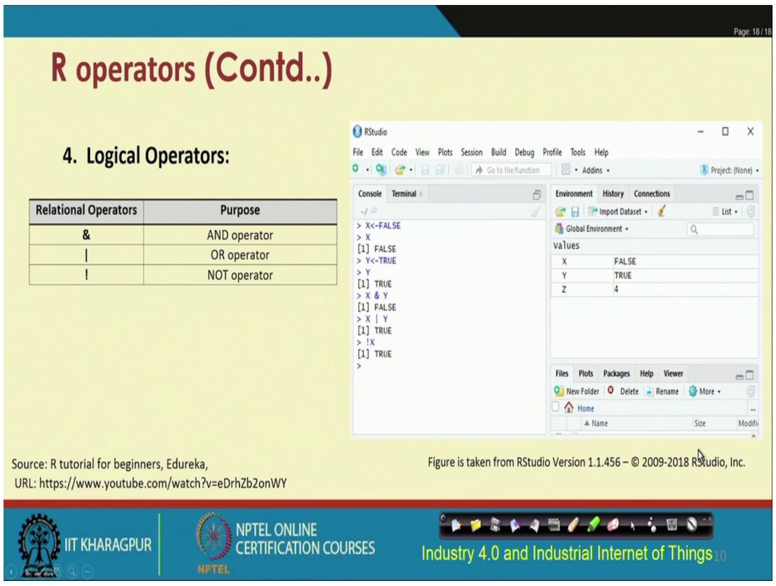
Source: R tutorial for beginners, Edureka,
URL: https://www.youtube.com/watch?v=eDrhZb2onWY

Figure is taken from RStudio Version 1.1.456 – © 2009-2018 RStudio, Inc.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Industry 4.0 and Industrial Internet of Things 9

A relational operators are also possible. Relational operators like less than, greater than, double equal to, this is not equal to and then you have greater than equal to, less than equal to and so on and their corresponding use and examples are also given over here. So, X greater than Y, if X, Y and Z are configured to have these different values corresponding values as shown over here in this variable window so then if you write X greater than Y, obviously, X is not greater than Y, so you will get the value false. Similarly X less than Y true, X double equal to Y false and so on.

(Refer Slide Time: 09:25)



## R operators (Contd..)

### 4. Logical Operators:

| Relational Operators | Purpose |
|---|---|
| & | AND operator |
| \| | OR operator |
| ! | NOT operator |

```
> X<-FALSE
> X
[1] FALSE
> Y<-TRUE
> Y
[1] TRUE
> X & Y
[1] FALSE
> X | Y
[1] TRUE
> !X
[1] TRUE
>
```

```
Values
X    FALSE
Y    TRUE
Z    4
```

Source: R tutorial for beginners, Edureka,
URL: https://www.youtube.com/watch?v=eDrhZb2onWY

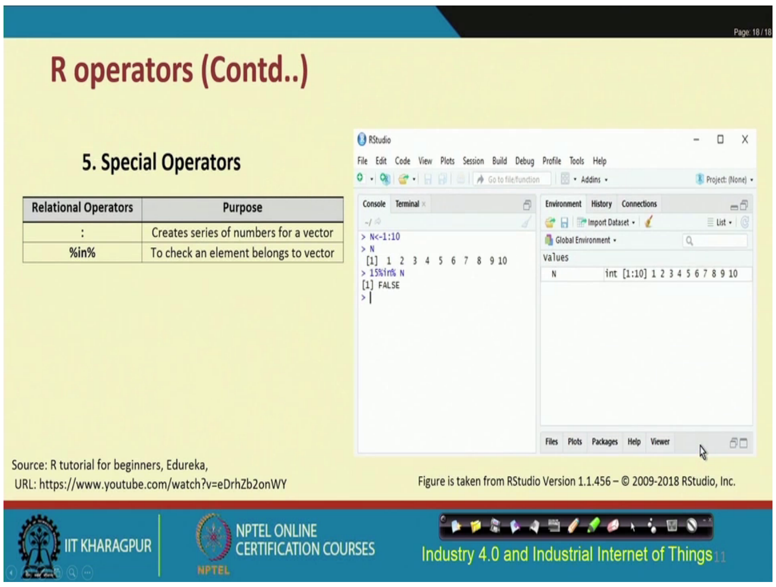
Figure is taken from RStudio Version 1.1.456 – © 2009-2018 RStudio, Inc.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Industry 4.0 and Industrial Internet of Things 10

So, then you have this logical operators like ampersand, then you have this vertical bar and this exclamation sign and there corresponding meanings are also given over ampersand is AND, then this is OR and this is NOT.

So, if these values are preconfigured to have these variables are preconfigured to have these values like this X false, Y true and Z 4 then, all of these different operators if you use them in this manner you are going to get these results that are shown over here like in the previous slides with the other operators.

(Refer Slide Time: 09:59)



Page: 18 / 18

## R operators (Contd..)

### 5. Special Operators

| Relational Operators | Purpose |
|---|---|
| : | Creates series of numbers for a vector |
| %in% | To check an element belongs to vector |

```
> N<-1:10
> N
 [1]  1  2  3  4  5  6  7  8  9 10
> 15%in% N
[1] FALSE
> |
```

Values

```
N              int [1:10] 1 2 3 4 5 6 7 8 9 10
```

Source: R tutorial for beginners, Edureka,
URL: https://www.youtube.com/watch?v=eDrhZb2onWY

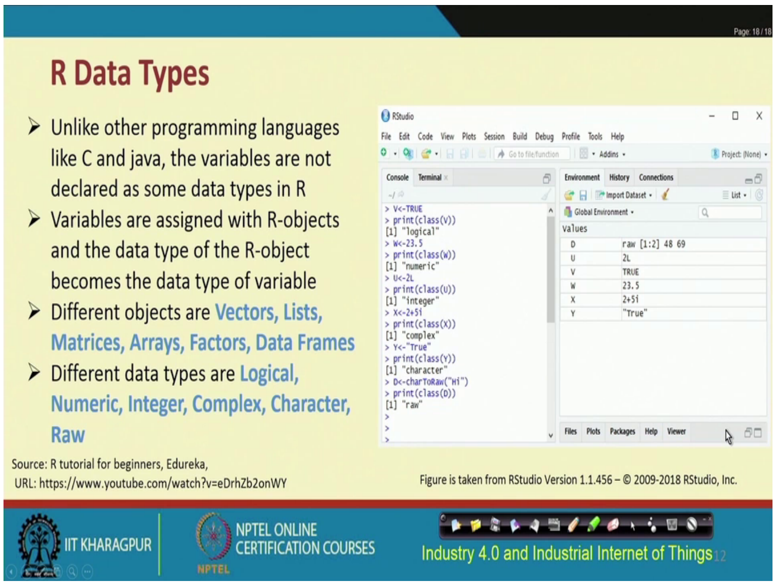
Figure is taken from RStudio Version 1.1.456 – © 2009-2018 RStudio, Inc.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NPTEL | Industry 4.0 and Industrial Internet of Things 11

There are certain special operators, this colon basically creates a series of numbers for a vector, then percentage in percent will help to check an element belonging to a vector or not so for instance if you have this N which is a set of 10 different integers like this defined like this, then if you print N; then if you print N then what you get is basically this particular list 1 through 10 right and this other operator also and its corresponding example is shown over here.

So, there are different data types that are supported, but before we understand these data types, we need to understand that there are different something called objects, R objects that are supported by R.

(Refer Slide Time: 10:37)



## R Data Types

- Unlike other programming languages like C and java, the variables are not declared as some data types in R
- Variables are assigned with R-objects and the data type of the R-object becomes the data type of variable
- Different objects are Vectors, Lists, Matrices, Arrays, Factors, Data Frames
- Different data types are Logical, Numeric, Integer, Complex, Character, Raw

```
> V<-TRUE
> print(class(V))
[1] "logical"
> W<-23.5
> print(class(W))
[1] "numeric"
> U<-2L
> print(class(U))
[1] "integer"
> X<-2+5i
> print(class(X))
[1] "complex"
> Y<-"True"
> print(class(Y))
[1] "character"
> D<-charToRaw("Hi")
> print(class(D))
[1] "raw"
>
>
>
```

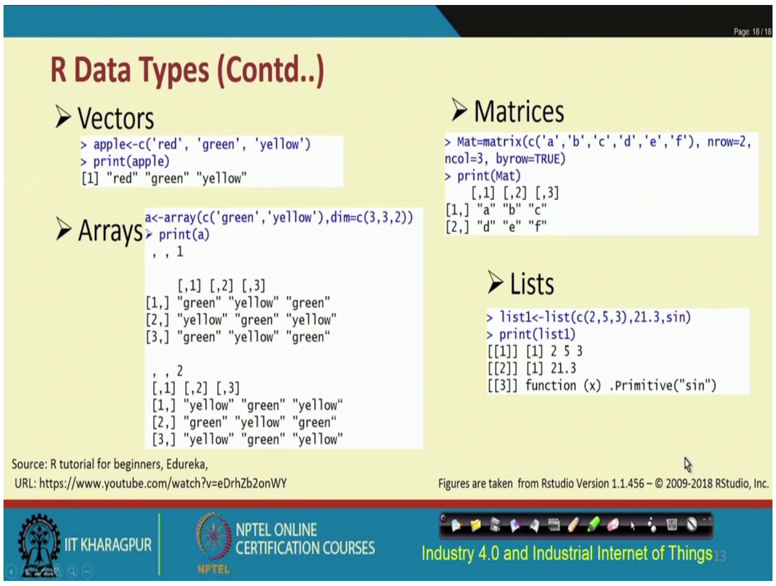
```
Values
 D    raw [1:2] 48 69
 U    2L
 V    TRUE
 W    23.5
 X    2+5i
 Y    "True"
```

Source: R tutorial for beginners, Edureka, URL: https://www.youtube.com/watch?v=eDrhZb2onWY

Figure is taken from RStudio Version 1.1.456 – © 2009-2018 RStudio, Inc.

So, these are these different R objects that are supported vectors, lists, matrices, arrays, factors, data frames and so on and so forth. So, these different; these different objects and the corresponding data types that are supported are the logical, numeric, integer, complex, character, raw etcetera. So, these together are used in order to build different complex programs that could be executed in this R studio platform. So, so this is just a screenshot once again these are these different; these different parameters and the different data types that are used. So, you have V which is true then if you print class V then you get logical and similarly for these different other ones also it is given over here.

(Refer Slide Time: 11:47)



## R Data Types (Contd..)

- Vectors

```
> apple<-c('red', 'green', 'yellow')
> print(apple)
[1] "red" "green" "yellow"
```

- Arrays

```
a<-array(c('green','yellow'),dim=c(3,3,2))
> print(a)
, , 1

     [,1]     [,2]     [,3]
[1,] "green"  "yellow" "green"
[2,] "yellow" "green"  "yellow"
[3,] "green"  "yellow" "green"

, , 2
     [,1]     [,2]     [,3]
[1,] "yellow" "green"  "yellow"
[2,] "green"  "yellow" "green"
[3,] "yellow" "green"  "yellow"
```

- Matrices

```
> Mat=matrix(c('a','b','c','d','e','f'), nrow=2,
ncol=3, byrow=TRUE)
> print(Mat)
     [,1] [,2] [,3]
[1,] "a"  "b"  "c"
[2,] "d"  "e"  "f"
```

- Lists

```
> list1<-list(c(2,5,3),21.3,sin)
> print(list1)
[[1]] [1] 2 5 3
[[2]] [1] 21.3
[[3]] function (x) .Primitive("sin")
```

Source: R tutorial for beginners, Edureka, URL: https://www.youtube.com/watch?v=eDrhZb2onWY

Figures are taken from Rstudio Version 1.1.456 – © 2009-2018 RStudio, Inc.

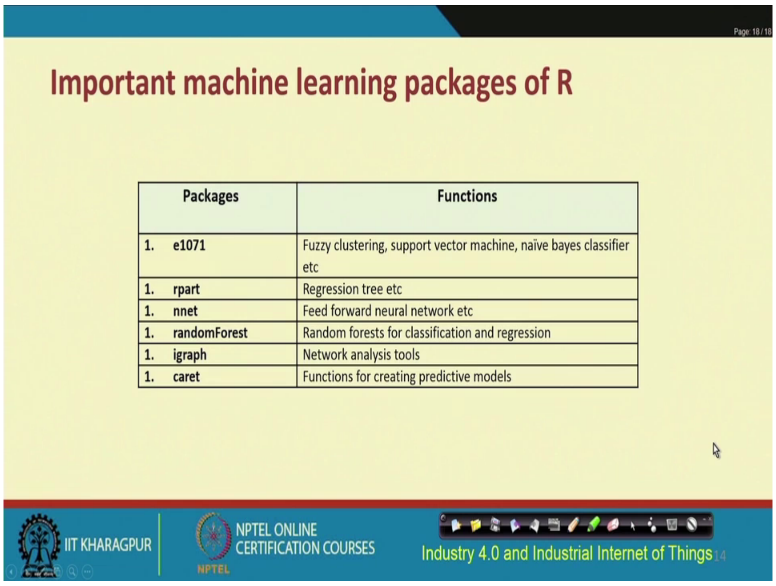
So, what are these different R objects? So, one is this vector. So, vector is basically like a one dimensional array. So, you can have something like this you can define a vector like this apple less than hyphen c within parenthesis you can give these values red, green, yellow and then if you print apple, then what you get are these different values of this particular vector red, green and yellow. So, this is just a one dimensional; one dimensional array and then you have matrices matrix can be defined in this manner. So, mat equal to matrix then you define this particular matrix so here you are saying that this is going to be the matrix where n row; that means, the number of rows is going to be 2, number of columns equal to 3 and by row true so; that means, that you are going to build this matrix by row.

So, you know so this is just a this matrix that we are going to build this matrix will have 2 rows, these are; these are the 2 different rows and it is going to have 3 columns like this and if you print this matrix then you get a, b, c, d, e, f; a, b, c, d, e, f like this. Then you have this arrays, arrays basically help you to get the multi dimensional; multi dimensional kind of array so it is like a 3D, 4D kind of array it can be built with the help of this arrays. So, this is how to do it a less than hyphen array c green, yellow dimension equal to c 3, 3, 2 print a.

So, if you print that then you get this is this one dimension, this is the second dimension, so this is this first dimension and then this is the second dimension as shown over here and this corresponding values green, yellow, green, yellow, green, yellow, green, yellow, green and so on. So, like this, this is one layer and this is the second layer you could have the third one also, fourth one also so you can build multiple layers of these matrices and you are going to get a multi dimensional array which is over here known as the array in R and then you have this list.

List basically can be built in this manner it is so; so basically the previous arrays, matrices we considered are of homogeneous type and in the case of list you are going to have heterogeneous values. So, you can have a floating point along with integer and so on. So, you are going to have all of these built in one unit. So, this is known as the list in R.

(Refer Slide Time: 14:15)



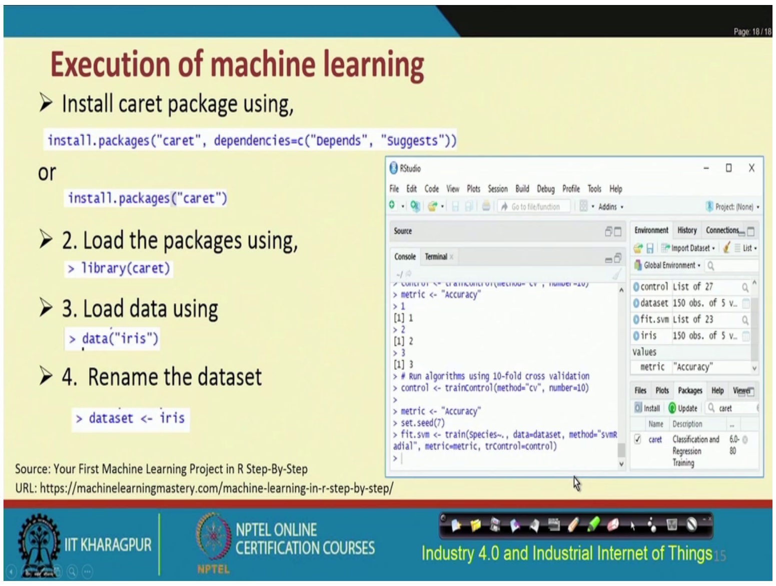
## Important machine learning packages of R

| Packages | Functions |
|---|---|
| 1. e1071 | Fuzzy clustering, support vector machine, naïve bayes classifier etc |
| 1. rpart | Regression tree etc |
| 1. nnet | Feed forward neural network etc |
| 1. randomForest | Random forests for classification and regression |
| 1. igraph | Network analysis tools |
| 1. caret | Functions for creating predictive models |

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Industry 4.0 and Industrial Internet of Things 14

So, there are different machine learning packages that are also supported by R the different packages for let us say random forest, there is a package which; which basically implements this is a library which basically implements this random forest of machine learning, then you have different other things like igraph which is for network analysis, caret for having different functions for creating predictive models, e1071 is a package or a library which supports fuzzy clustering svm, then nice Naive Bayes classifier etcetera, rpart is for building regression trees, nnet is for feed forward neural network like this there are many other different packages that are supported for R.

(Refer Slide Time: 15:03)



## Execution of machine learning

- Install caret package using,

```
install.packages("caret", dependencies=c("Depends", "Suggests"))
```

or

```
install.packages("caret")
```

- 2. Load the packages using,

```
> library(caret)
```

- 3. Load data using

```
> data("iris")
```

- 4. Rename the dataset

```
> dataset <- iris
```

```
> metric <- "Accuracy"
> 1
[1] 1
> 2
[1] 2
> 3
[1] 3
> # Run algorithms using 10-fold cross validation
> control <- trainControl(method="cv", number=10)
>
> metric <- "Accuracy"
> set.seed(7)
> fit.svm <- train(Species~., data=dataset, method="svmRadial", metric=metric, trControl=control)
>
```

```
control  List of 27
dataset  150 obs. of 5 v...
fit.svm  List of 23
iris     150 obs. of 5 v...
Values
 metric  "Accuracy"
```

| | Name | Description | |
|---|---|---|---|
| ✓ | caret | Classification and Regression Training | 6.0-80 |

Source: Your First Machine Learning Project in R Step-By-Step
URL: https://machinelearningmastery.com/machine-learning-in-r-step-by-step/

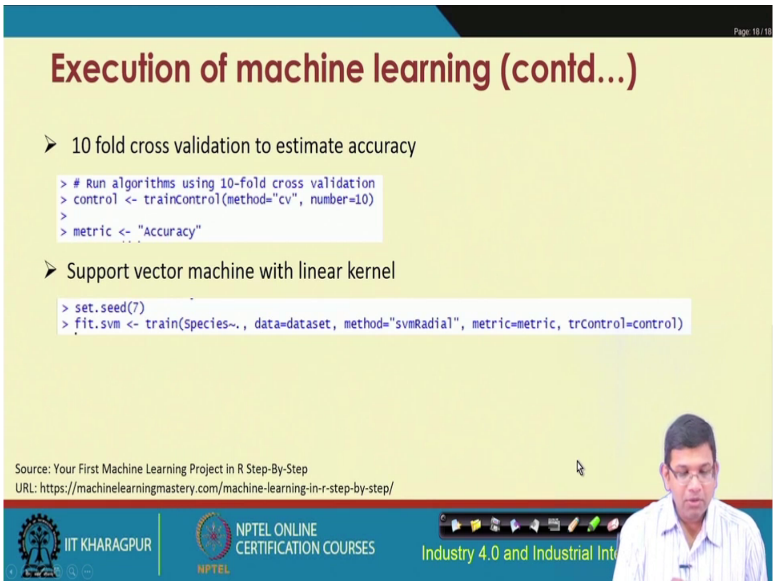
IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Industry 4.0 and Industrial Internet of Things 15

So, how do you execute these in for machine learning implementations. So, again this is a snapshot of this R studio. So, first of all I will I am just going to give you an example. So, here we have you know this installation of the caret package we will be using this caret package so we are we have to install it first. So, this installation is done using this particular command, install dot packages caret and then you have certain other things or you can use install dot packages caret this will also install this caret package or this library which has which has different functions already which are already implemented.

Then to load that particular package this is this command that you are going to use library carried loading the data using this package for that loading the data basically you can do with the help of this particular command data iris. Iris is the data set and for example, over here as you can see this iris data set and the corresponding values that are over here, the features are given over here these are these different other data sets like the data set, iris, the control etcetera. So, the corresponding values that are given over here are shown in this particular window.

So, data iris will basically load the data from this iris dataset and rename the data set can be done. So, you know data set less than hyphen iris will basically rename this particular data set to data set and as you can see over here so this data set and iris will both have the same contents.

(Refer Slide Time: 16:51)



Page: 18/18

## Execution of machine learning (contd...)

- 10 fold cross validation to estimate accuracy

```
> # Run algorithms using 10-fold cross validation
> control <- trainControl(method="cv", number=10)
>
> metric <- "Accuracy"
```

- Support vector machine with linear kernel

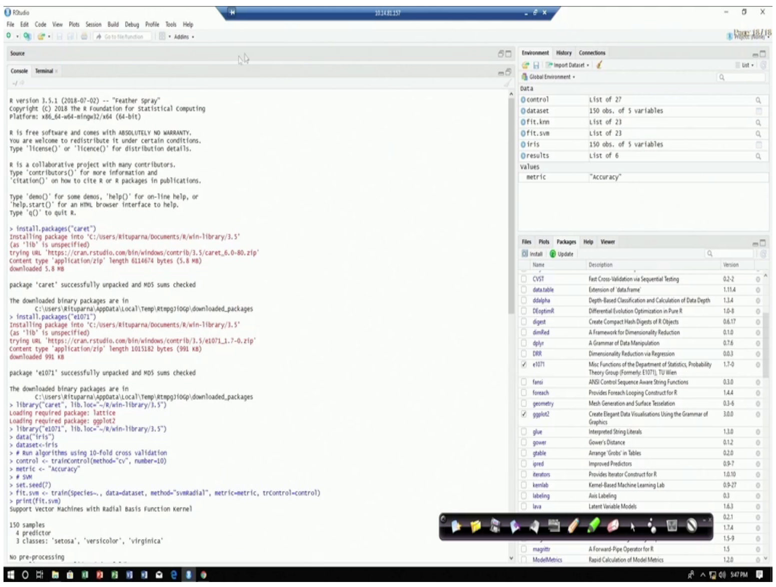
```
> set.seed(7)
> fit.svm <- train(Species~., data=dataset, method="svmRadial", metric=metric, trControl=control)
```

Source: Your First Machine Learning Project in R Step-By-Step
URL: https://machinelearningmastery.com/machine-learning-in-r-step-by-step/

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Industry 4.0 and Industrial Int

So, then you have different other comments for cross validation 10-fold cross validation for estimation of accuracy these are actually different things that you can do with the help of you know machine learning algorithms such as support fixed vector machines and so on. So, for cross validation you know this is this is the command that you use control train control method equal to cv, number equal to 10 and the metric is accuracy. For use of support vector machine you know you first set the seed, so this is the random number seed, the seed you set first and then basically to invoke svm on a particular data set, data equal data set and the methods that will be used are also specified you use this particular comment. So, fit dot svm, this svm comment will be executing svm on this particular data set; data set the name of which is data set.

(Refer Slide Time: 17:49)


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Environment History Connections
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer
Install
Update
5:47 PM

```
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("caret")
Installing package into 'C:/Users/Rituparna/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/caret_6.0-80.zip'
Content type 'application/zip' length 6114674 bytes (5.8 MB)
downloaded 5.8 MB

package 'caret' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Rituparna\AppData\Local\Temp\RtmpgJiOGp\downloaded_packages
> install.packages("e1071")
Installing package into 'C:/Users/Rituparna/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/e1071_1.7-0.zip'
Content type 'application/zip' length 1015182 bytes (991 KB)
downloaded 991 KB

package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Rituparna\AppData\Local\Temp\RtmpgJiOGp\downloaded_packages
> library("caret", lib.loc="~/R/win-library/3.5")
Loading required package: lattice
Loading required package: ggplot2
> library("e1071", lib.loc="~/R/win-library/3.5")
> data("iris")
> dataset<-iris
> # Run algorithms using 10-fold cross validation
> control <- trainControl(method="cv", number=10)
> metric <- "Accuracy"
> # SVM
> set.seed(7)
> fit.svm <- train(Species~., data=dataset, method="svmRadial", metric=metric, trControl=control)
> print(fit.svm)
Support Vector Machines with Radial Basis Function Kernel

150 samples
  4 predictor
  3 classes: 'setosa', 'versicolor', 'virginica'

No pre-processing
```

| Data | |
|---|---|
| control | List of 27 |
| dataset | 150 obs. of 5 variables |
| fit.knn | List of 23 |
| fit.svm | List of 23 |
| iris | 150 obs. of 5 variables |
| results | List of 6 |
| **Values** | |
| metric | "Accuracy" |

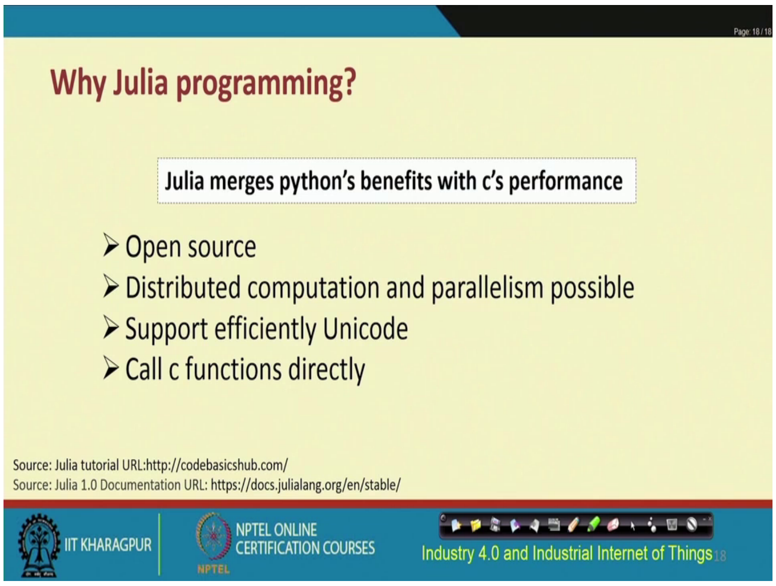
| | Name | Description | Version |
|---|---|---|---|
| ☐ | CVST | Fast Cross-Validation via Sequential Testing | 0.2-2 |
| ☐ | data.table | Extension of 'data.frame' | 1.11.4 |
| ☐ | ddalpha | Depth-Based Classification and Calculation of Data Depth | 1.3.4 |
| ☐ | DEoptimR | Differential Evolution Optimization in Pure R | 1.0-8 |
| ☐ | digest | Create Compact Hash Digests of R Objects | 0.6.17 |
| ☐ | dimRed | A Framework for Dimensionality Reduction | 0.1.0 |
| ☐ | dplyr | A Grammar of Data Manipulation | 0.7.6 |
| ☐ | DRR | Dimensionality Reduction via Regression | 0.0.3 |
| ✓ | e1071 | Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien | 1.7-0 |
| ☐ | fansi | ANSI Control Sequence Aware String Functions | 0.3.0 |
| ☐ | foreach | Provides Foreach Looping Construct for R | 1.4.4 |
| ☐ | geometry | Mesh Generation and Surface Tesselation | 0.3-6 |
| ✓ | ggplot2 | Create Elegant Data Visualisations Using the Grammar of Graphics | 3.0.0 |
| ☐ | glue | Interpreted String Literals | 1.3.0 |
| ☐ | gower | Gower's Distance | 0.1.2 |
| ☐ | gtable | Arrange 'Grobs' in Tables | 0.2.0 |
| ☐ | ipred | Improved Predictors | 0.9-7 |
| ☐ | iterators | Provides Iterator Construct for R | 1.0.10 |
| ☐ | kernlab | Kernel-Based Machine Learning Lab | 0.9-27 |
| ☐ | labeling | Axis Labeling | 0.3 |
| ☐ | lava | Latent Variable Models | 1.6.3 |
| | [illegible] | [illegible] | 0.2.1 |
| | [illegible] | [illegible] | 1.7.4 |
| | [illegible] | [illegible] | 1.5-9 |
| ☐ | magrittr | A Forward-Pipe Operator for R | 1.5 |
| ☐ | ModelMetrics | Rapid Calculation of Model Metrics | 1.2.0 |

So, this is basically this R studio environment platform that you can see over here all these different data sets and their corresponding features are given over here in this particular panel and for installation of these different libraries you have to basically from this particular check list you will have to check the libraries that you want to include.

So, as I was telling you for installation, this command will install the caret package. So, install dot packages caret will install this caret package over here through this particular comment. So, then install dot packages e1071 will install this package e1071 right. So, this is this other package and as you cans see over here e1071 which has miscellaneous functions of the department of statistics probability theory etcetera from TU Wien that basically will be installed and also then if you have to want to include these libraries then these are the comments to be used.

So, library caret this will basically help you in loading these libraries, then library ten e1071 will also load this library then data iris this data iris will basically import this particular data set and this is going to be renamed to this data set data set and then using svm you execute the data that you have in that data set in this particular manner. So, this will be executed as svm will be executed on this data set which has the name data let us. So, what we have got is a glimpse of the R platform which is widely used for analytics.

(Refer Slide Time: 19:47)



Page: 18 / 18

## Why Julia programming?

Julia merges python's benefits with c's performance

- Open source
- Distributed computation and parallelism possible
- Support efficiently Unicode
- Call c functions directly

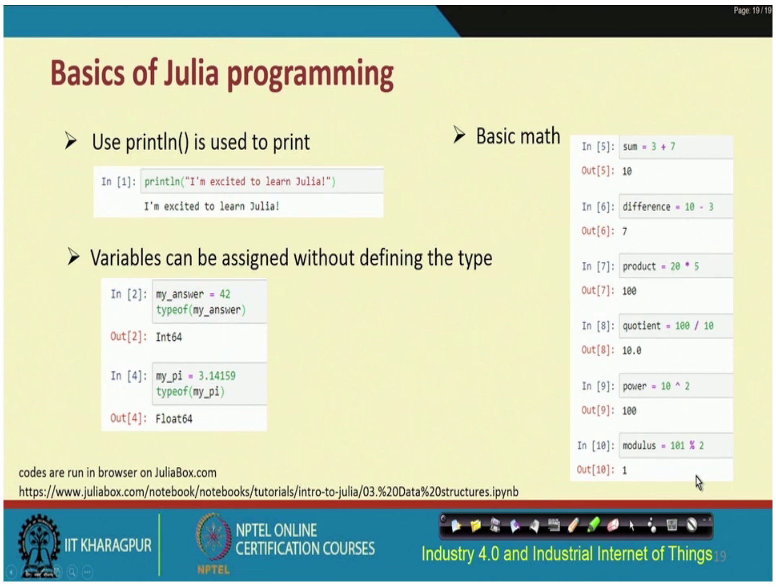
Source: Julia tutorial URL:http://codebasicshub.com/
Source: Julia 1.0 Documentation URL: https://docs.julialang.org/en/stable/

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Industry 4.0 and Industrial Internet of Things 18

Let us now have a glimpse at the Julia programming language which is quite popular programming language used for analytics programming high level programming of analytics and Julia is a popular programming language that builds on the benefits that are obtained from python and the performance that is achieved with c language. So, it is a combination of you can think of conceptually as a combination of benefits from python and c.

So, Julia is a open source programming language and it supports distributed computation and parallelism and there are different c like called function calls that are supported by Julia.

(Refer Slide Time: 20:17)



Page: 19 / 19

## Basics of Julia programming

- Use println() is used to print

```
In [1]: println("I'm excited to learn Julia!")
        I'm excited to learn Julia!
```

- Variables can be assigned without defining the type

```
In [2]: my_answer = 42
        typeof(my_answer)
Out[2]: Int64

In [4]: my_pi = 3.14159
        typeof(my_pi)
Out[4]: Float64
```

- Basic math

```
In [5]: sum = 3 + 7
Out[5]: 10

In [6]: difference = 10 - 3
Out[6]: 7

In [7]: product = 20 * 5
Out[7]: 100

In [8]: quotient = 100 / 10
Out[8]: 10.0

In [9]: power = 10 ^ 2
Out[9]: 100

In [10]: modulus = 101 % 2
Out[10]: 1
```

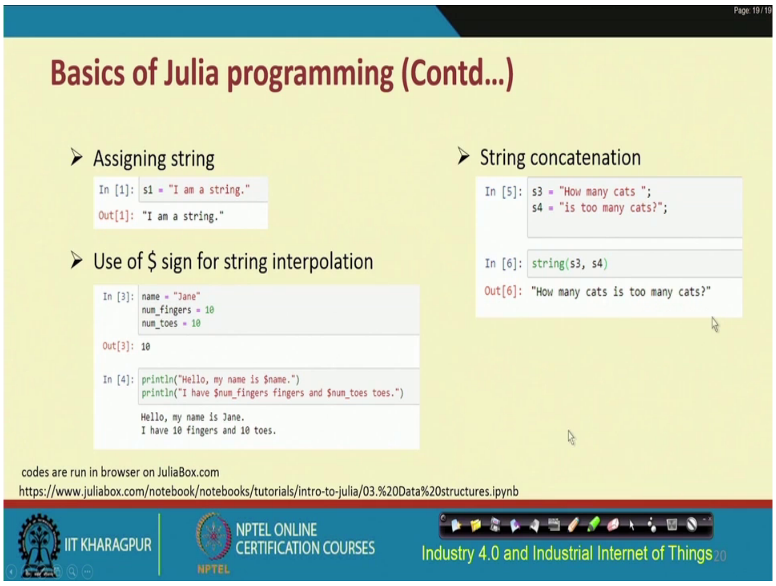
codes are run in browser on JuliaBox.com
https://www.juliabox.com/notebook/notebooks/tutorials/intro-to-julia/03.%20Data%20structures.ipynb

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Industry 4.0 and Industrial Internet of Things 19

So, you know these are some of these different common programming you know programming features that are there one is this println. So, this println is for printing right so it is used for printing and for example, if you type in println I am excited to learn Julia then after that if you hit enter you will get I am excited to learn Julia so this is the use of println and then for different other variables you can define these different variables in this manner for example, this particular variable my answer equal to 42 type of my answer and then if you execute this the output will be in 64. So, type of; type of this variable is integer. So, you get integer 64 which is the type of this my answer. Then my pi equal to 3.14159 and then if you type in type of my pi you get this 14.64. So, these are the different outputs of this comment type of. Type of is a basically comment for getting that type of a particular variable and this assignment of values to variables can be done in this manner as shown in Julia.

So, the basic math functions are supported sum equal to 3 plus 7 will basically do the sum of two different integers which are given over here so 3 plus 7; 10. Difference equal to 10 minus 3 will give you 7, product equal 20 times 5 will give you 100, quotient equal to 100 slash 10 will give you 10 dot 0. So, this basically will be in the floating point and then you have power equal to 10 to the power 2 so this cap is basically for the power. So, 10 to the power 2 will give you 100 and then modulus 101 percentage 2 will give you this mod value of 1.

(Refer Slide Time: 22:31)



Page: 19 / 19

## Basics of Julia programming (Contd...)

- Assigning string

```
In [1]: s1 = "I am a string."
Out[1]: "I am a string."
```

- Use of $ sign for string interpolation

```
In [3]: name = "Jane"
        num_fingers = 10
        num_toes = 10
Out[3]: 10

In [4]: println("Hello, my name is $name.")
        println("I have $num_fingers fingers and $num_toes toes.")

        Hello, my name is Jane.
        I have 10 fingers and 10 toes.
```

- String concatenation

```
In [5]: s3 = "How many cats ";
        s4 = "is too many cats?";

In [6]: string(s3, s4)
Out[6]: "How many cats is too many cats?"
```

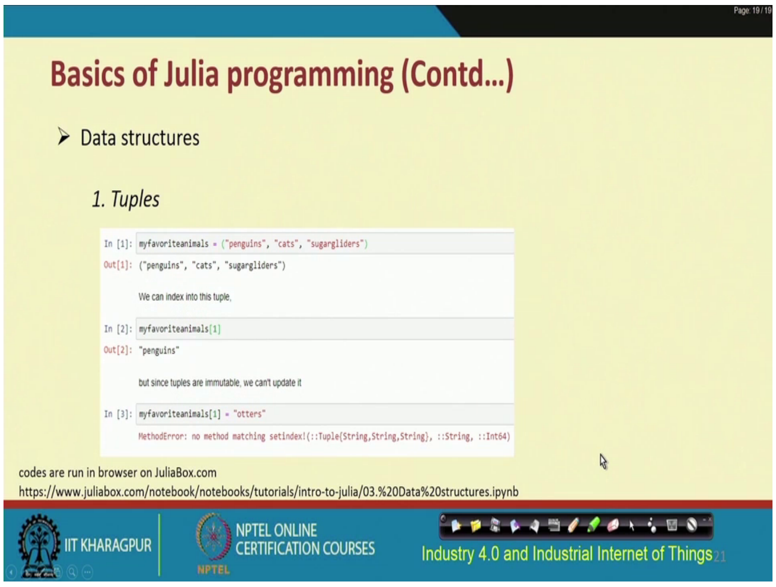
codes are run in browser on JuliaBox.com
https://www.juliabox.com/notebook/notebooks/tutorials/intro-to-julia/03.%20Data%20structures.ipynb

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NPTEL | Industry 4.0 and Industrial Internet of Things 20

So, these are the different other operations that are supported assignment of strings. So, for example, you can have a string defined in this manner s1 equal to within quotation mark I am a string, this will basically create this string s1 and then if you execute then you get this output I am a string.

So, dollar sign is used for string interpolation. So, name equal to Jane, number of fingers equal to 10, number of toes is equal to 10 then the output will be 10. Now if you type in println hello my name is dollar name then the value of name which is Jane is going to be printed in this manner. So, similarly printing I have dollar number of figures fingers and dollar number of toes, then you will get the corresponding values the dollar basical dollar number figures will give you the corresponding value that is stored in this particular variable. So, dollar number toes will give you this particular value that is stored for this particular variable which is 10. So, I have 10 fingers and 10 toes is what is going to be printed if you execute this particular statement. Then you have for string concatenation you can use something like this s3 equal to how many cats is a string s4 is another string is too many cats, then string s3 comma s4 will concatenate the strings, s3 and s4 concatenate means like these two strings are going to be joined together. So, you are going to get this joint string which is how many cats is too many cats.

(Refer Slide Time: 24:11)



## Basics of Julia programming (Contd...)

- Data structures

*1. Tuples*

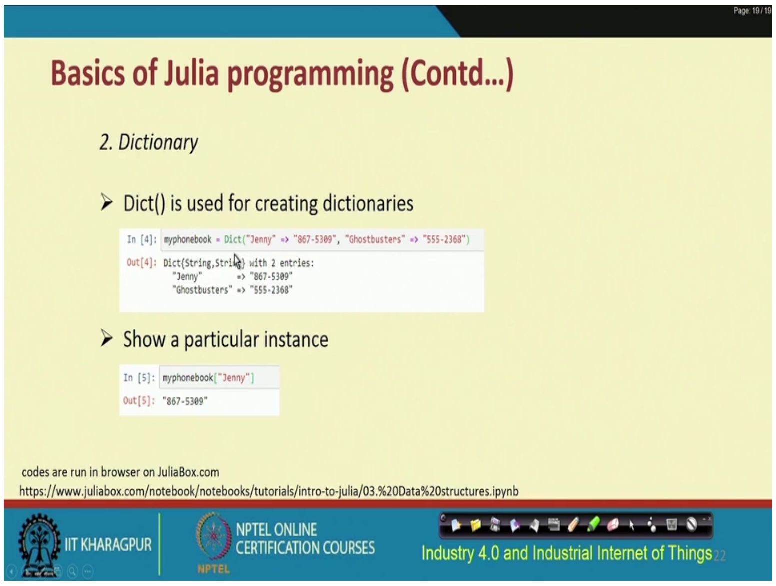
```
In [1]: myfavoriteanimals = ("penguins", "cats", "sugargliders")
Out[1]: ("penguins", "cats", "sugargliders")

        We can index into this tuple,

In [2]: myfavoriteanimals[1]
Out[2]: "penguins"

        but since tuples are immutable, we can't update it

In [3]: myfavoriteanimals[1] = "otters"
        MethodError: no method matching setindex!(::Tuple{String,String,String}, ::String, ::Int64)
```

codes are run in browser on JuliaBox.com
https://www.juliabox.com/notebook/notebooks/tutorials/intro-to-julia/03.%20Data%20structures.ipynb

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Industry 4.0 and Industrial Internet of Things 21

There are different data structures that are supported in Julia. So, these are the ones that are shown over here. So, tuples basically. So, this is how you can create these tuples. So, my favourite animals equal to penguins, cats, sugar gliders this will create this particular tuple and so the output will be is so if you execute this thing so output is going to be penguins, cats, sugar gliders. So, like this you can built different other types of tuples as well.

(Refer Slide Time: 24:45)



## Basics of Julia programming (Contd...)

*2. Dictionary*

- Dict() is used for creating dictionaries

```
In [4]: myphonebook = Dict("Jenny" => "867-5309", "Ghostbusters" => "555-2368")
Out[4]: Dict{String,String} with 2 entries:
          "Jenny"        => "867-5309"
          "Ghostbusters" => "555-2368"
```

- Show a particular instance

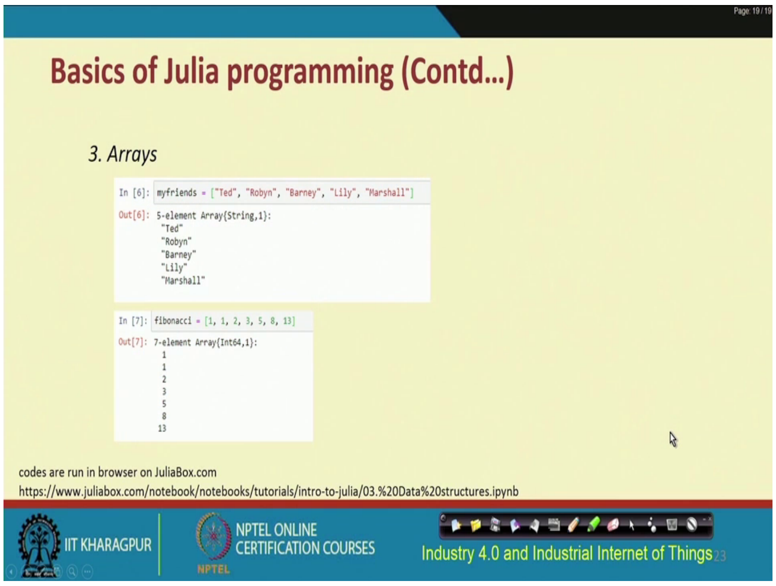
```
In [5]: myphonebook["Jenny"]
Out[5]: "867-5309"
```

codes are run in browser on JuliaBox.com
https://www.juliabox.com/notebook/notebooks/tutorials/intro-to-julia/03.%20Data%20structures.ipynb

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Industry 4.0 and Industrial Internet of Things 22

To create a dictionary this is the comment to be used. So, my phonebook equal to dict is this particular comment will create this dictionary. So, dict then you have all these different values that are going to be stored against these different variables. So, this will create this particular dictionary named my phonebook. So, then if you; if you print this particular my phonebook dictionary, then this is how it is going to look like. If you want to show a particular instance this is the way to do it.

(Refer Slide Time: 25:21)



Page: 19 / 19

## Basics of Julia programming (Contd...)

*3. Arrays*

```
In [6]: myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]

Out[6]: 5-element Array{String,1}:
         "Ted"
         "Robyn"
         "Barney"
         "Lily"
         "Marshall"
```

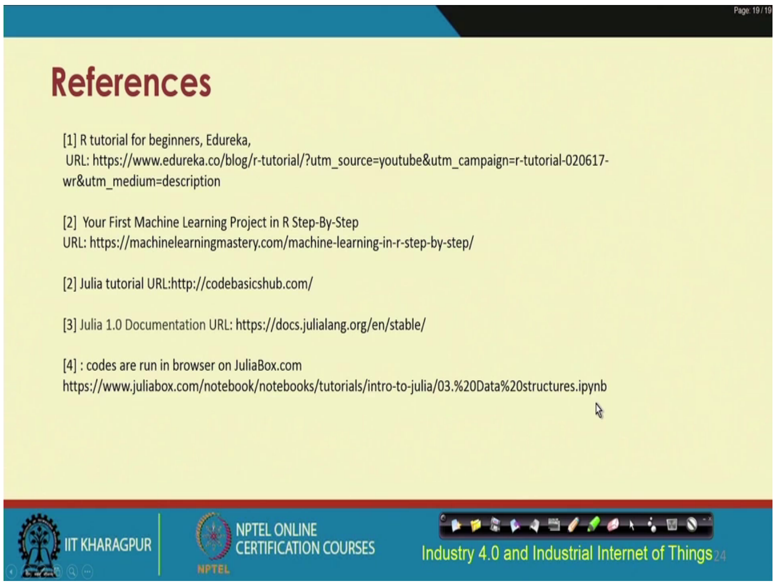
```
In [7]: fibonacci = [1, 1, 2, 3, 5, 8, 13]

Out[7]: 7-element Array{Int64,1}:
          1
          1
          2
          3
          5
          8
         13
```

codes are run in browser on JuliaBox.com
https://www.juliabox.com/notebook/notebooks/tutorials/intro-to-julia/03.%20Data%20structures.ipynb

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Industry 4.0 and Industrial Internet of Things 23

Arrays to build arrays in Julia you can do it very easily with the help of putting all these array items within square brackets. So, this will create this 5 element array my friends having these elements Ted, Robyn, Barney, Lily and Marshall. So, this one will create a seven element Fibonacci array having these different elements over here.

(Refer Slide Time: 25:53)



Page: 19 / 19

## References

[1] R tutorial for beginners, Edureka,
URL: https://www.edureka.co/blog/r-tutorial/?utm_source=youtube&utm_campaign=r-tutorial-020617-wr&utm_medium=description

[2] Your First Machine Learning Project in R Step-By-Step
URL: https://machinelearningmastery.com/machine-learning-in-r-step-by-step/

[2] Julia tutorial URL:http://codebasicshub.com/

[3] Julia 1.0 Documentation URL: https://docs.julialang.org/en/stable/

[4] : codes are run in browser on JuliaBox.com
https://www.juliabox.com/notebook/notebooks/tutorials/intro-to-julia/03.%20Data%20structures.ipynb

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Industry 4.0 and Industrial Internet of Things 24

So, with this we come to an end of Julia. These are these different references that what I have done is basically, I have given you a very high level expository view of what is R, what is Julia, how is this R studio platform like, what are these basic operators that are there, what are the different functions that are supported in Julia and so on. So, these are the only very basic half an hour kind of glimpse of what is there. With this particular lecture you are not expected to become an expert of R or Julia; however, if your job requires you or if you are indeed interested to have a deeper understanding of R and Julia you have to build your knowledge further and you have to take different tutorials that are there for R and Julia and you have to become master of these languages if you are required to, but from a course perspective for IIoT an industry 4.0, I think this particular knowledge that you have been provided through this particular lecture this will be sufficient for you at least to start with. So, these are some of these different references if you are interested you can go through these different references for in the future.

Thank you.