Introduction to R Software
Prof. Shalabh
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

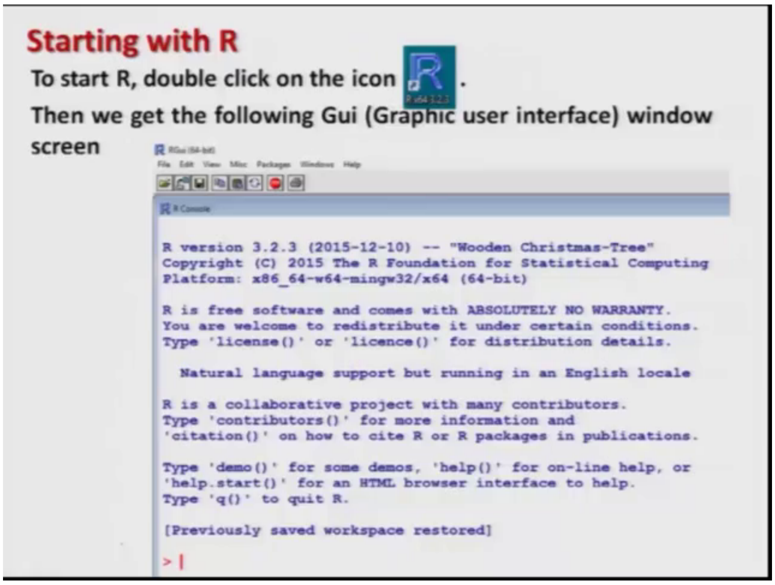
Lecture – 02
Help, Demonstration, Examples, Packages and Libraries

Welcome to the next lecture on introduction to R software. You may kindly recall that in the last lecture we have understood what is R, why should we use R and how to install R and R studio software on our computer. So, now, we are ready with the our software. So, the phoenix aspect is that that once we start with R we would like to take the help for example, whenever we are writing a program and we want to do something we do not know how to get it done. So, we have to seek some help. So, there are 2 different ways to get help in the R software, some are inbuilt which are internal help sources and some are external help sources I guess Google and so on.

So, we are going to discuss about here that how to take help for the R software and then there are some other aspects like and how to guess the demonstration of a particular syntax, how to get a for examples on a particular syntax, how to use packages and libraries? So, all these topics we are going to discuss in this lecture. So, and now we try to learn how to start R and what are the different aspects, but before that let me tell you how we are going to do. I have prepared some slides on which I will be giving you the details how to execute a program and side by side I will try to demonstrate that how to use a particular syntax for a particular command over the R.

So, I will be switching over different windows, one windows for the R software and say another window for our slides.

(Refer Slide Time: 02:08)



Starting with R

To start R, double click on the icon .

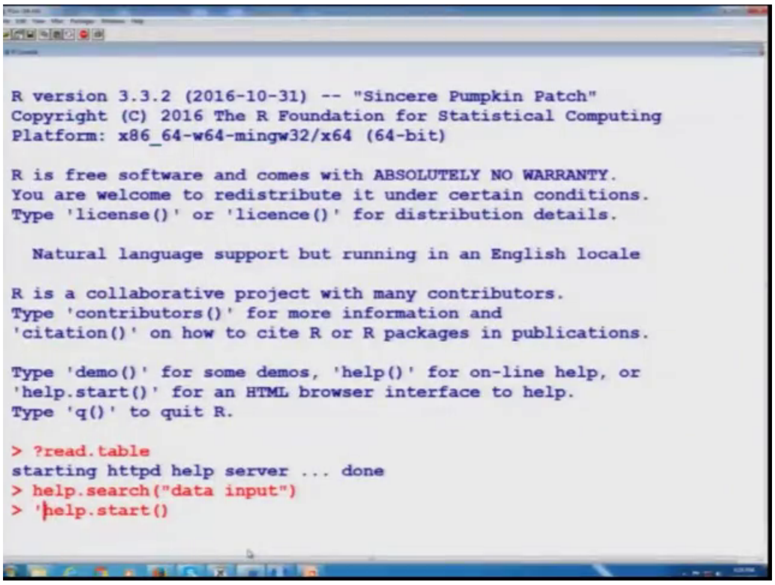
Then we get the following Gui (Graphic user interface) window screen

```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 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.

  Natural language support but running in an English locale

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.

[Previously saved workspace restored]

> |
```

So, you can see here on the desktop of your computer that there is going to be a icon of here R like this. So, you need to double click on this icon right. So, let us try to do it here so now, I have taken a screenshot of the R Gui and this is pasted over here and now let us try to see what is there in this window. If you try to read it here for the first 2 paragraph it is trying to give you the details about the R software and in the last 2 paragraph it is trying to give you a different options for example, how to get a citation here and how to obtain the demonstration of a particular function, how to get here a help and how to quit the R program and so on.

So, if you try to see, if you try to read it you will get information about this, but now after this once we have started the program over here, now our objective is that how to obtain the help in the R. So, first of all what we can do you can see in this window. So, what you can do you can just double click over here and if you try sit here we will get here something like this help.

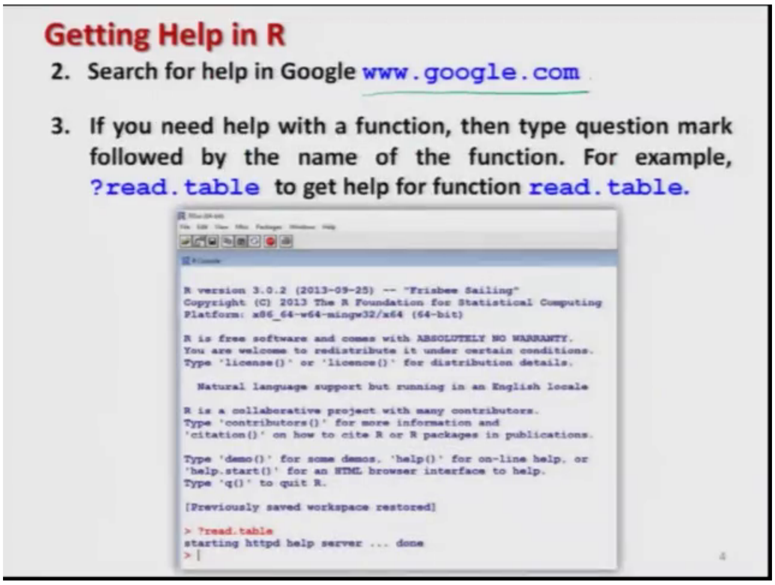
(Refer Slide Time: 03:30)



```
R version 3.3.2 (2016-10-31) -- "Sincere Pumpkin Patch"
Copyright (C) 2016 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.

  Natural language support but running in an English locale

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.

> ?read.table
starting httpd help server ... done
> help.search("data input")
> 'help.start()
```

And then you can see that there are different types of thing and if you try to click over here you will get here more things right, on the internet and then you can for example, if you want to know about the feedback or what is R you can just click over here and it will give you each and everything. So, now, what I have done, I have just copied and pasted this thing over on my slide and because to give you here and here and idea I am trying to explain you here for example, you can see here there are some FAQ on R, FAQ on the R for windows some manuals in the pdf format, different types of here R function help for in the html format on the internet and so on. So, all these things are there and so the first approach to take the help on any function is that you please try to go through the help menu and try to seek help.

(Refer Slide Time: 04:34)



Getting Help in R

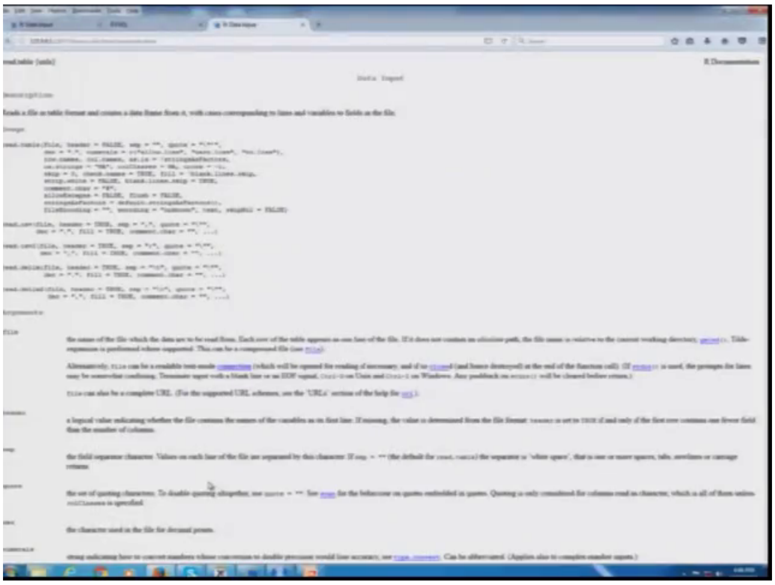
2. Search for help in Google www.google.com
3. If you need help with a function, then type question mark followed by the name of the function. For example, ?read.table to get help for function read.table.

```
R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 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.

  Natural language support but running in an English locale

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.

[Previously saved workspace restored]

> ?read.table
starting httpd help server ... done
> |
```

The next option is that we can go to Google dot com and there are different resources which are available at a different sites including the support from the website of the R software. So, we can just look into different types of example, different types of syntax and they will also try to extend their help to us. If you try to see here that suppose I want to take a help on help on some function say called as read dot table, well I am using here some new names like as function or read table right. This things should not to bother you because later on you are going to learn all this things, but my objective here is simply to show you that suppose you know that there is command read dot table and so obtain the help on that function.

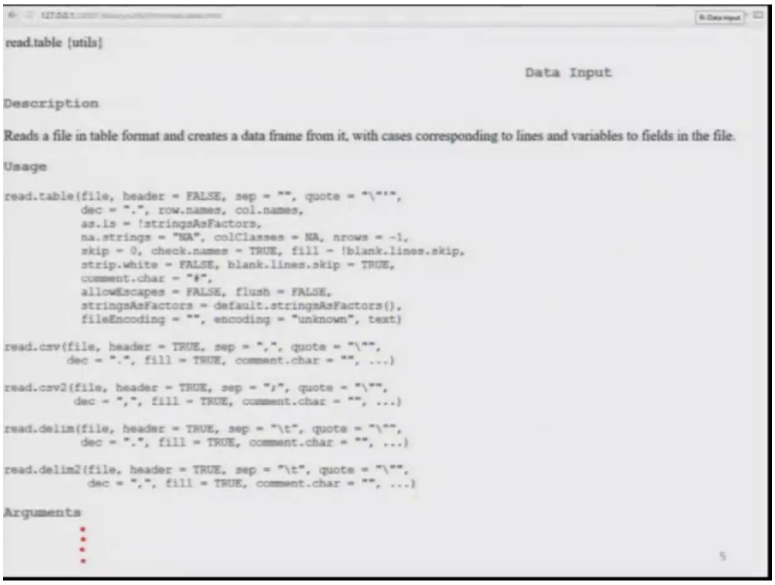
So, suppose I want to obtain the some help on the read dot table function, what I have to do here I simply have to write say question mark and after this I have 2 type here the functioning for example, here I have done here question mark followed by read dot table and we try to type this read dot table on the command line for example, I will simply try to copy and paste over here. So, that we save sub time and I tried to paste it over here.

(Refer Slide Time: 06:04)



And you can see here that this comes on an internet sites automatically and this is here the address of the site over here and then you can see it from here to here it is trying to give you the entire details about the function read dot table right. I miss simply is calling it just to show you that these things available that these things are available to you and I have taken here the screenshot of this thing in the following slides for example, here we are getting the same slide over here.

(Refer Slide Time: 06:43)



read.table {utils}

Data Input

Description

Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

Usage

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", row.names, col.names,
           as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
          dec = ",", fill = TRUE, comment.char = "", ...)

read.delim(file, header = TRUE, sep = "\t", quote = "\"",
           dec = ".", fill = TRUE, comment.char = "", ...)

read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
            dec = ",", fill = TRUE, comment.char = "", ...)
```

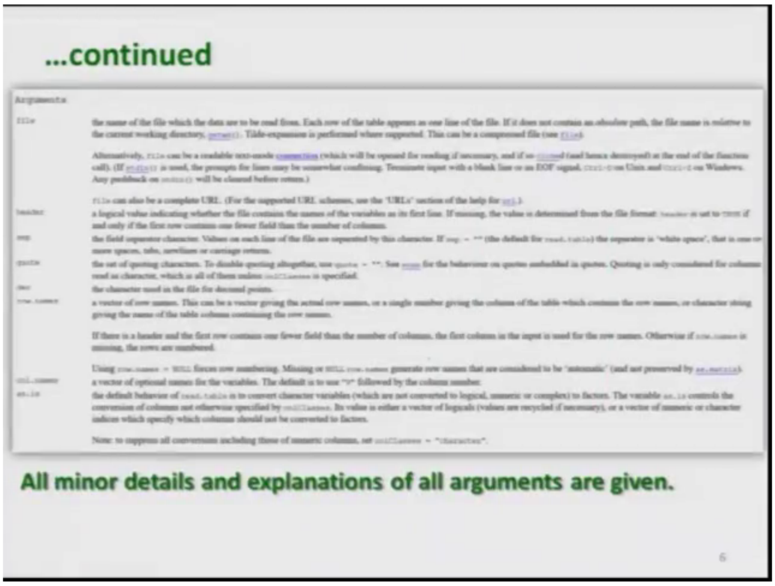
Arguments

⋮

5

And then you can see here we get here the slide like this one here and simply for example, if you try to see here it is trying to give you the complete format about the command read dot table and then how to read a comma separated file and so on.

(Refer Slide Time: 07:01)



...continued

Arguments

file — the name of the file which the data are to be read from. Each row of the table appears as one line of the file. If it does not contain an absolute path, the file name is relative to the current working directory, getwd(). Tilde-expansion is performed where supported. This can be a compressed file (see file).

Alternatively, file can be a readable text-mode connection (which will be opened for reading if necessary, and if so closed (and hence destroyed) at the end of the function call). (If stdin() is used, the prompts for lines may be somewhat confusing. Terminate input with a blank line or an EOF signal, Ctrl-D on Unix and Ctrl-Z on Windows. Any pushback on stdin() will be cleared before return.)

file can also be a complete URL. (For the supported URL schemes, see the 'URLs' section of the help for url.)

header — a logical value indicating whether the file contains the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns.

sep — the field separator character. Values on each line of the file are separated by this character. If sep = "" (the default for read.table) the separator is 'white space', that is one or more spaces, tabs, newlines or carriage returns.

quote — the set of quoting characters. To disable quoting altogether, use quote = "". See scan for the behaviour on quotes embedded in quotes. Quoting is only considered for columns read as character, which is all of them unless colClasses is specified.

dec — the character used in the file for decimal points.

row.names — a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row names.

If there is a header and the first row contains one fewer field than the number of columns, the first column in the input is used for the row names. Otherwise if row.names is missing, the rows are numbered.

Using row.names = NULL forces row numbering. Missing or NULL row.names generate row names that are considered to be 'automatic' (and not preserved by as.matrix).

col.names — a vector of optional names for the variables. The default is to use "V" followed by the column number.

as.is — the default behavior of read.table is to convert character variables (which are not converted to logical, numeric or complex) to factors. The variable as.is controls the conversion of columns not otherwise specified by colClasses. Its value is either a vector of logicals (values are recycled if necessary), or a vector of numeric or character indices which specify which columns should not be converted to factors.

Note: to suppress all conversions including those of numeric columns, set colClasses = "character".

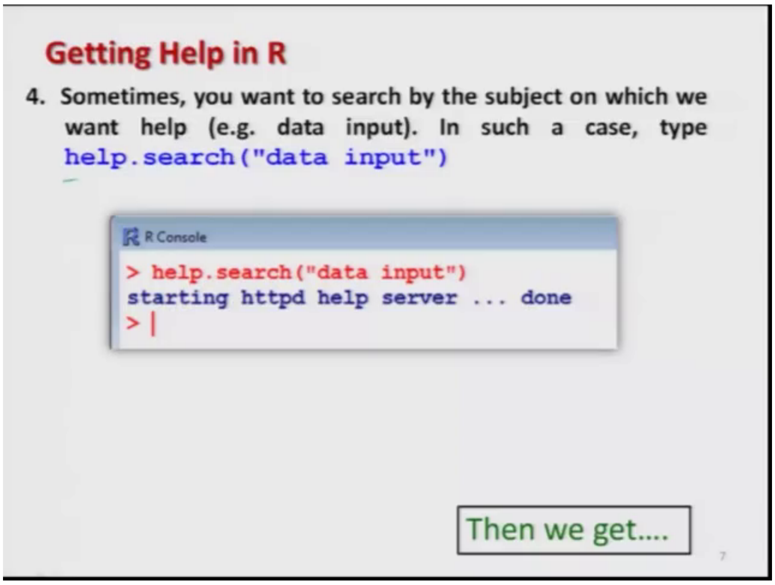
All minor details and explanations of all arguments are given.

6

And you will see that there are different types of arguments which are given here that how to give here the filename, how to handle the header part how to handle the separator parts and so on.

So, this gives you the most authentic help on say R software on the function read dot table, well I have taken one example to show you similarly as you to see further and as soon as you are you are introduced to a function you can take the help on this thing also. So, it gives you the complete exhaustive help, next approaches that I can use here a function help dot search.

(Refer Slide Time: 07:38)



Getting Help in R

4. Sometimes, you want to search by the subject on which we want help (e.g. data input). In such a case, type `help.search("data input")`

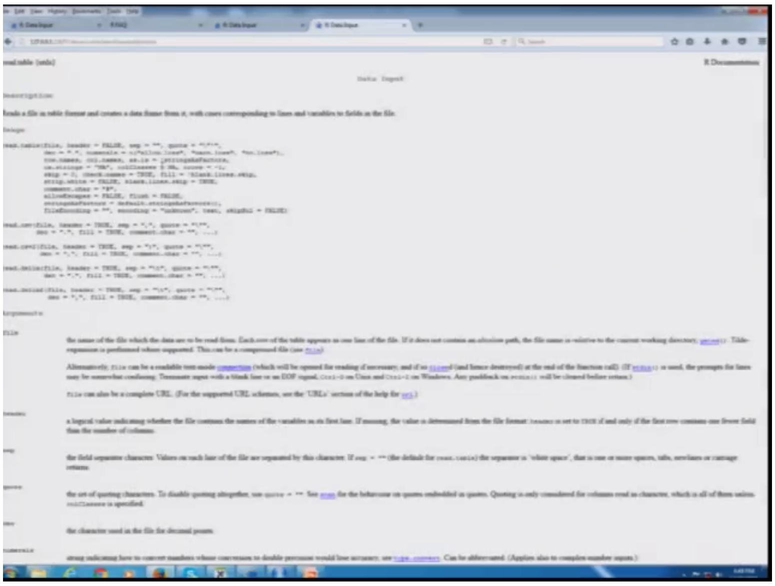
R Console

```
> help.search("data input")
starting httpd help server ... done
> |
```

Then we get....

So, the format of this command is the following that I will try to type help dot search and suppose I want to know about how to input the data, right and suppose I do not know that what are the different ways of data input and so on, but I have some way idea that I want to import the data. So, in very simple language I am trying to write down help dot search and inside the bracket with this double quotes I am trying to write data input.

(Refer Slide Time: 08:30)



read.table {utils}

R Documentation

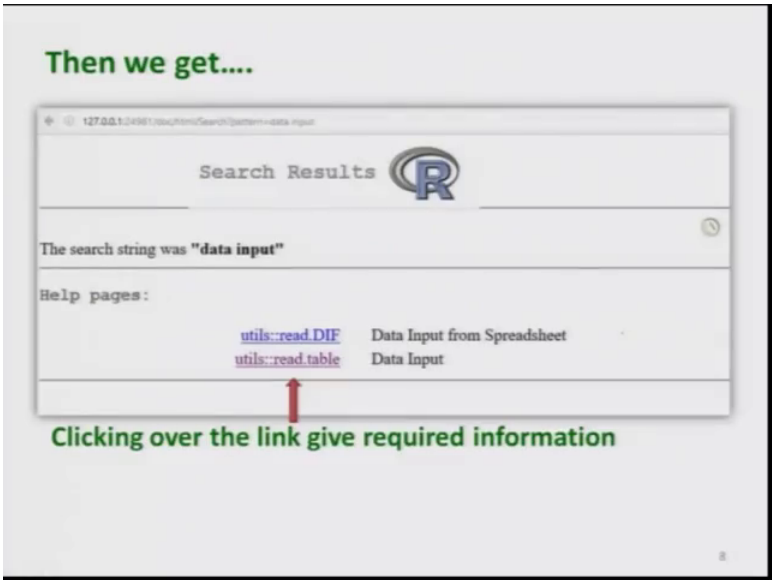
Data Input

Description

Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

So, now try us try to do it here, copy this command into out R you see, now this is again going to the website of the R software and it is trying to give you here that there are 2 different ways to input the data of one is read dot d f and say and that is read or table and suppose if I want to know about the read dot table it will give me the all the details over here right. So, if you try to see here I have taken here with the screenshot of the first slide and in the next slide I have taken the screenshot of the second thing.

(Refer Slide Time: 08:58)



Then we get....

Search Results

The search string was **"data input"**

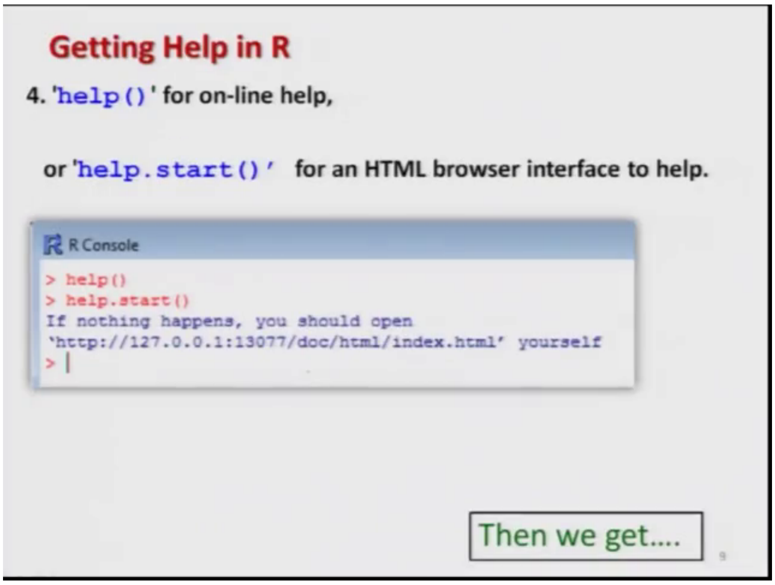
Help pages:

utils::read.DIF	Data Input from Spreadsheet
utils::read.table	Data Input

Clicking over the link give required information

So, here if you either you try to click over here then you will get the details of the utility which is contained and read dot DIF, that is trying to give you that how to input the data from a spreadsheet and similarly if you want to input the details there is another utility what is called as read dot table and it is trying to give you that how to input the data and as soon as you try to click over any of the link you will get more details about the these utilities that how to get the data right.

(Refer Slide Time: 09:30)



Getting Help in R

4. 'help()' for on-line help,

or 'help.start()' for an HTML browser interface to help.

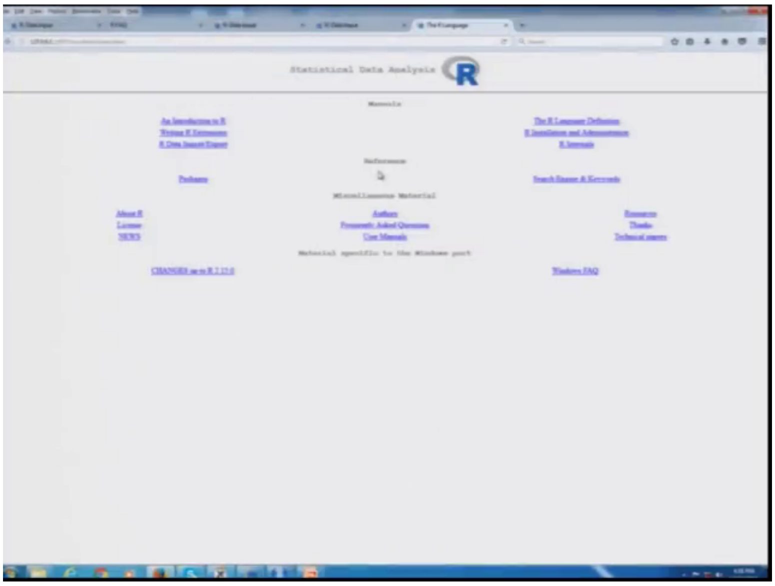
```
> help()
> help.start()
If nothing happens, you should open
'http://127.0.0.1:13077/doc/html/index.html' yourself
> 
```

Then we get....

Another ways that I can type here something like help and say and these 2 brackets or other approaches is that I can write down here, help dot start and inside this bracket and this will bring us to the website of the R software let us try to do it here.

So, suppose I tried to type here help dot start.

(Refer Slide Time: 10:13)



Statistical Data Analysis

Manuals

An Introduction to R
Writing R Extensions
R Data Import/Export

The R Language Definition
R Installation and Administration
R Internals

Reference

Packages

Search Engine & Keywords

Miscellaneous Material

About R
License
NEWS

Authors
Frequently Asked Questions
User Manuals

Resources
Thanks
Technical papers

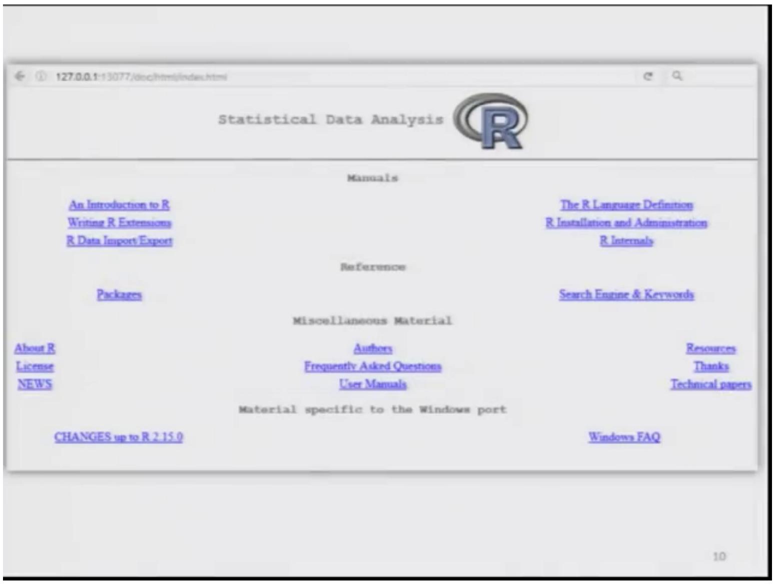
Material specific to the Windows port

CHANGES up to R 2.15.0

Windows FAQ

You can see here that this brings us to set this type of site and here again we have different types of things. So, I have again screenshot of the this site which is here.

(Refer Slide Time: 10:25)



127.0.0.1:13077/doc/html/index.html

Statistical Data Analysis

Manuals

An Introduction to R
Writing R Extensions
R Data Import/Export

The R Language Definition
R Installation and Administration
R Internals

Reference

Packages

Search Engine & Keywords

Miscellaneous Material

About R
License
NEWS

Authors
Frequently Asked Questions
User Manuals

Resources
Thanks
Technical papers

Material specific to the Windows port

CHANGES up to R 2.15.0

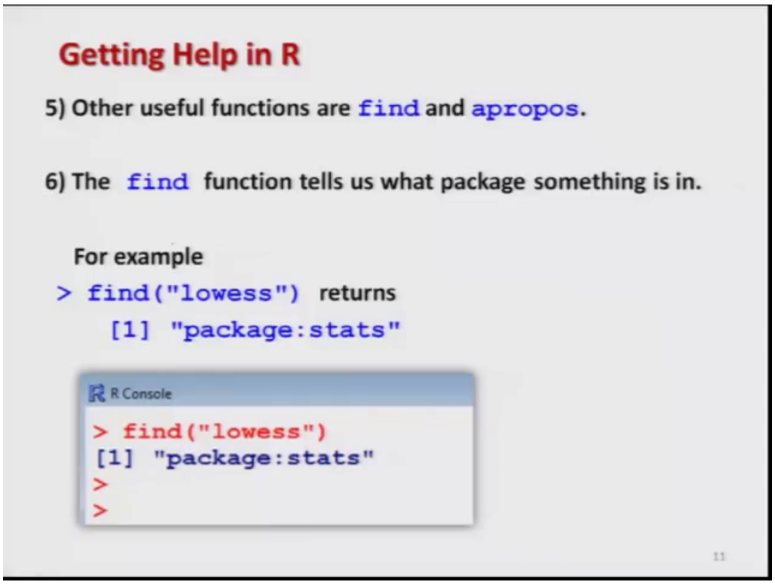
Windows FAQ

10

And now you can read it actually more clearly that it was trying to give you an introduction to R and then how to write R extension say import and export of the data and there are many many things. So, you simply have to click over the link and then you have to read it and they will give you the complete idea about the software right. One thing again I would like to make it clear that this course is at an elementary level where my objective is to give you some help. So, that you can start with the R software and in any programming language if you try to learn it more you have to do it yourself.

So, this course is intending to help you to start with the R program and to understand the basic fundamentals, right. So, you have to just keep and these things in mind.

(Refer Slide Time: 11:19)



Getting Help in R

5) Other useful functions are `find` and `apropos`.

6) The `find` function tells us what package something is in.

For example

`> find("lowess")` returns

`[1] "package:stats"`

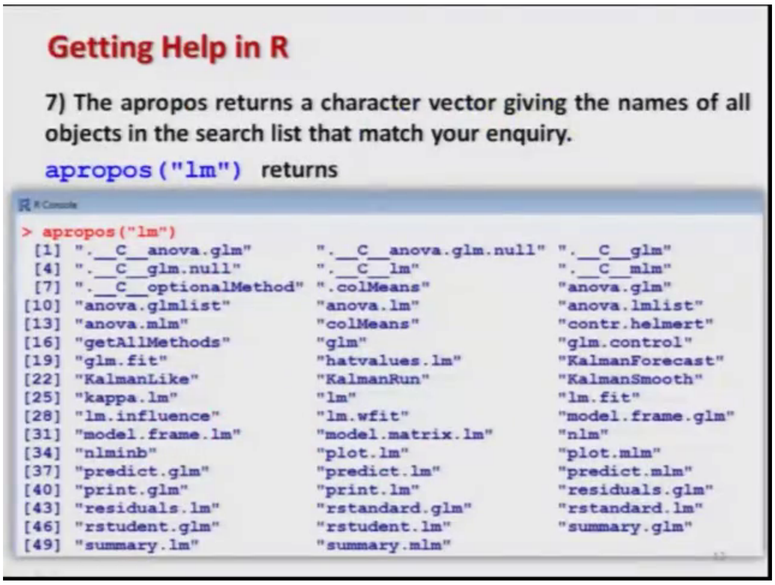
R Console

```
> find("lowess")
[1] "package:stats"
>
>
```

So, now the next approach to get some help us to use two commands first is find and another is apropos right. this function here find, this tells us what is there inside a package for example, if I say that there is a package which is called as lowess l o w e s s lowess right and suppose I want to know what is in this lowess. So, now, if I try to type this find lowess inside the bracket and contain inside the double quotes you will see here what do I get right. It is trying to give us the information that it is a package that is contained in statistics what is it denoted here as a considered as a s t a t s.

So, again I have taken the the screenshot of this thing and you can see here that it is trying to give you the information that this this lowess package is a package which is contained and the inside the package call s t a t s stats right ok.

(Refer Slide Time: 12:53)



Getting Help in R

7) The apropos returns a character vector giving the names of all objects in the search list that match your enquiry.

`apropos("lm")` returns

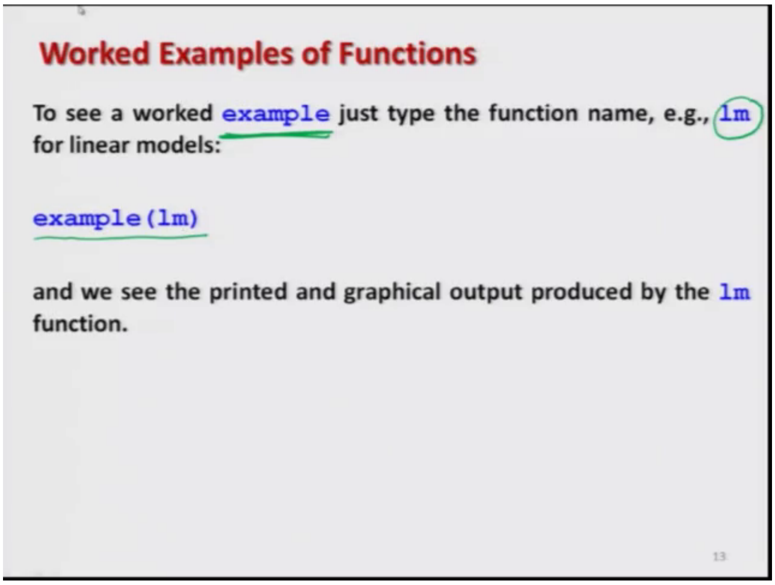
```
> apropos("lm")
 [1] ".__C__anova.glm"        ".__C__anova.glm.null" ".__C__glm"
 [4] ".__C__glm.null"         ".__C__lm"             ".__C__mlm"
 [7] ".__C__optionalMethod"   ".colMeans"            "anova.glm"
[10] "anova.glmlist"          "anova.lm"             "anova.lmlist"
[13] "anova.mlm"              "colMeans"             "contr.helmert"
[16] "getAllMethods"          "glm"                  "glm.control"
[19] "glm.fit"                "hatvalues.lm"         "KalmanForecast"
[22] "KalmanLike"             "KalmanRun"            "KalmanSmooth"
[25] "kappa.lm"               "lm"                   "lm.fit"
[28] "lm.influence"           "lm.wfit"              "model.frame.glm"
[31] "model.frame.lm"         "model.matrix.lm"      "nlm"
[34] "nlminb"                 "plot.lm"              "plot.mlm"
[37] "predict.glm"            "predict.lm"           "predict.mlm"
[40] "print.glm"              "print.lm"             "residuals.glm"
[43] "residuals.lm"           "rstandard.glm"        "rstandard.lm"
[46] "rstudent.glm"           "rstudent.lm"          "summary.glm"
[49] "summary.lm"             "summary.mlm"
```

Now, there is another what we called as apropos, apropos is a command that will show it what is wear for example, suppose I have some idea of say some word say lm you will find that l m is the command for say linear models, but anyway I have some brief idea that there is some command which I remember that contains two letters lm. So, I will try to type here apropos here like this and inside the bracket within the double quotes I will try to write down here lm and when I try to do so, I get here a screenshot like this one well.

Now, you can have faith on me that this is screenshots are exactly the replica what are we going to get when I try to execute this commands, well just to gain your confidence I will try to run this apropos l m here once again you can see here that it is trying to give us us this type of information and I have simply taken the screenshot over here and so that I can explain you it here more easily. So, you can see here that that I was trying to find out the 2 letters l m and we can see here it is lm is here right, if you try to say l m is here say l m is here, l m is here, l m is here, l m is here, l m is here. So, in and there are so many function in which 2 letters l m R common.

So, now from here you can choose that which function you wanted to know right ok.

(Refer Slide Time: 14:53)



Worked Examples of Functions

To see a worked `example` just type the function name, e.g., `lm` for linear models:

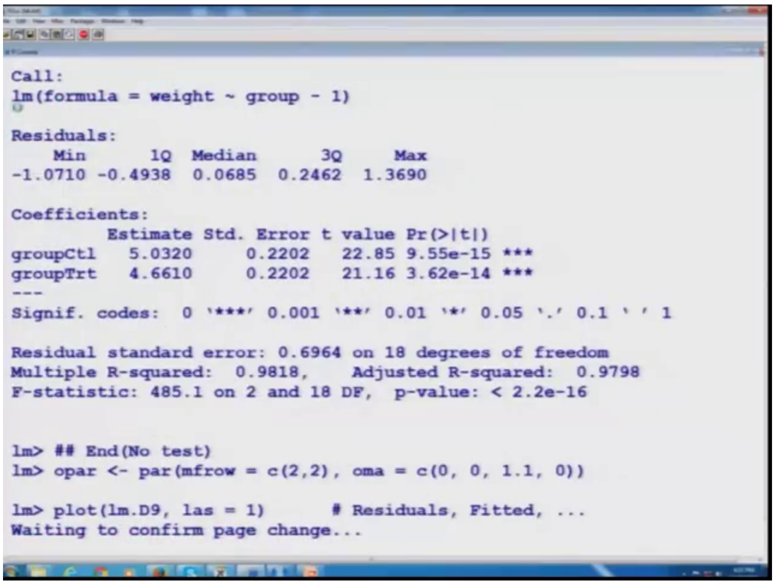
```
example(lm)
```

and we see the printed and graphical output produced by the `lm` function.

13

So, now after this once you have some idea about the function you would like to know who to use it, well their syntax and commands are given in the R help, but definitely we would like to have some examples. So, that the things are more clear to us, this R software helps us and and it provides us some example which has builted inside the software. So, if you want to know any example of any particular command or say function you have to use the syntax here say example and suppose I want to have some example about a l m, what I have to do? If we have to do I simply have to write here example and lm inside the brackets right and soul try to do it here and we will see here what do we get here.

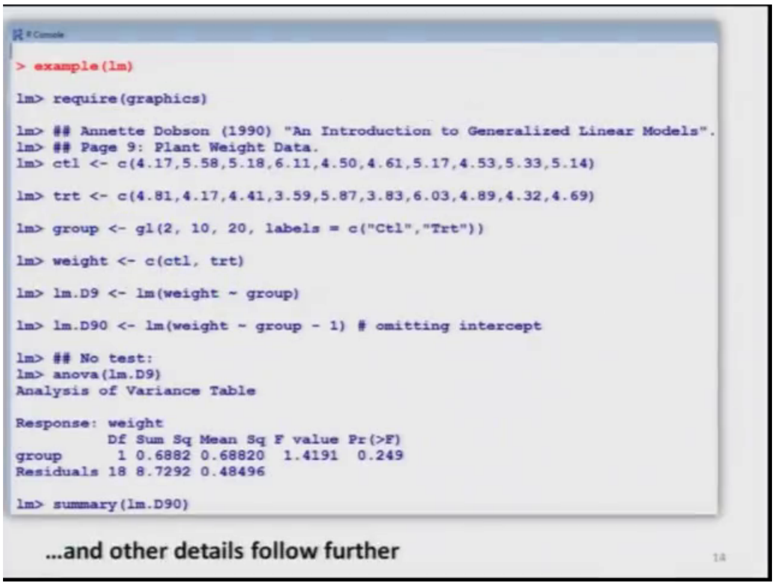
(Refer Slide Time: 15:49)



```
Call:
lm(formula = weight ~ group - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0710 -0.4938  0.0685  0.2462  1.3690

Coefficients:
         Estimate Std. Error t value Pr(>|t|)
groupCtl   5.0320     0.2202   22.85 9.55e-15 ***
groupTrt   4.6610     0.2202   21.16 3.62e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6964 on 18 degrees of freedom
Multiple R-squared:  0.9818,    Adjusted R-squared:  0.9798
F-statistic: 485.1 on 2 and 18 DF,  p-value: < 2.2e-16


lm> ## End(No test)
lm> opar <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))

lm> plot(lm.D9, las = 1)      # Residuals, Fitted, ...
Waiting to confirm page change...
```

So, now you can see here that it is trying to give you here some examples and so on. So, suppose now I try to copy and paste this l m over here. So, you can see here that it is trying to give us here all the information about the l m and now if I try to change the page here I will get here more information. So, I have just copied and pasted all this information in my slide over here so that you have a better idea.

(Refer Slide Time: 16:23)



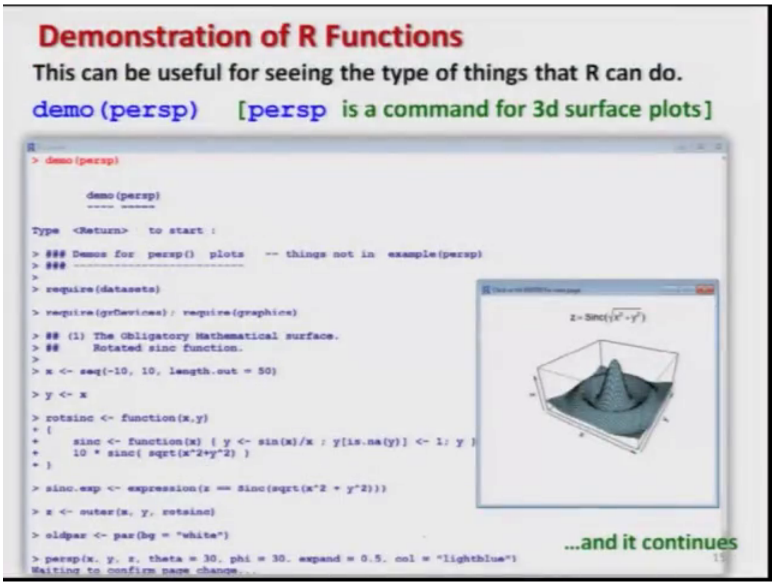
```
> example(lm)

lm> require(graphics)

lm> ## Annette Dobson (1990) "An Introduction to Generalized Linear Models".
lm> ## Page 9: Plant Weight Data.
lm> ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)

lm> trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)

lm> group <- gl(2, 10, 20, labels = c("Ctl","Trt"))

lm> weight <- c(ctl, trt)

lm> lm.D9 <- lm(weight ~ group)

lm> lm.D90 <- lm(weight ~ group - 1) # omitting intercept

lm> ## No test:
lm> anova(lm.D9)
Analysis of Variance Table

Response: weight
          Df Sum Sq Mean Sq F value Pr(>F)
group      1 0.6882 0.68820  1.4191  0.249
Residuals 18 8.7292 0.48496

lm> summary(lm.D90)
```

...and other details follow further

14

So, this is the first slide which we have got here right and if you try to see here it is trying to give you all the details that is trying to take an example from this book and page number this, there is some data and on which they have implemented the package l m and then they are trying to give all the details right and then there all some detail over here right.

(Refer Slide Time: 16:52)



Demonstration of R Functions

This can be useful for seeing the type of things that R can do.

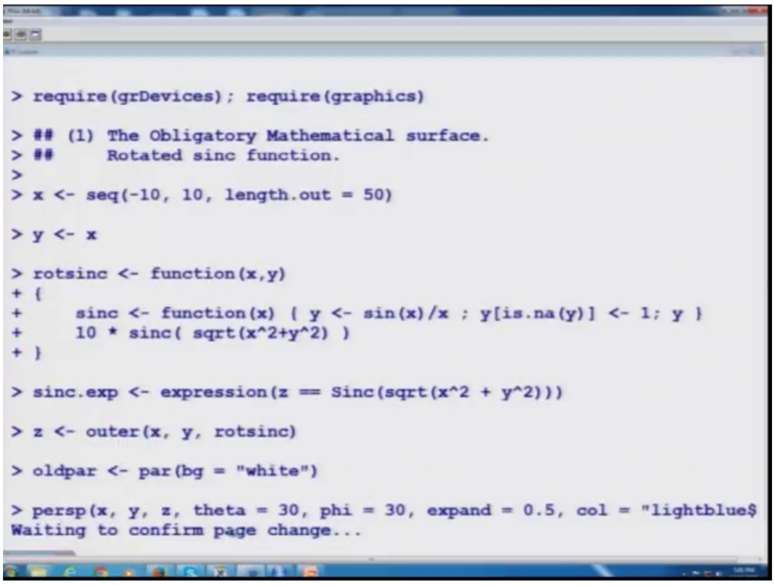
demo (persp)   [persp is a command for 3d surface plots]

```
> demo(persp)


        demo(persp)
        ---- ------

Type  <Return>   to start :

> ### Demos for  persp()  plots   -- things not in  example(persp)
> ### -------------------------
>
> require(datasets)

> require(grDevices); require(graphics)

> ## (1) The Obligatory Mathematical surface.
> ##     Rotated sinc function.
>
> x <- seq(-10, 10, length.out = 50)

> y <- x

> rotsinc <- function(x,y)
+ {
+     sinc <- function(x) { y <- sin(x)/x ; y[is.na(y)] <- 1; y }
+     10 * sinc( sqrt(x^2+y^2) )
+ }

> sinc.exp <- expression(z == Sinc(sqrt(x^2 + y^2)))

> z <- outer(x, y, rotsinc)

> oldpar <- par(bg = "white")

> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue")
Waiting to confirm page change...
```

...and it continues

So, after this let us try to see that in case if I want to have a demonstration of a function then how to get it done right. So, in order to get a demonstration of a function, the contracts is demo and inside the bracket I have to write down the name of the function on which I need the demonstration. So, for example here I am trying to take a demonstration on the function p e r s p actually this persp is a function that creates the perspective plots which are 3 dimensional surface plots right. So, let us try to do it on the R directly and then after this I will try to show you that I have copied and pasted the screenshots over here. So, now, if I tried to copy and paste over here demo persp you can see it is trying to show you here something like this, its information and after this it is trying to give you here something here more right.

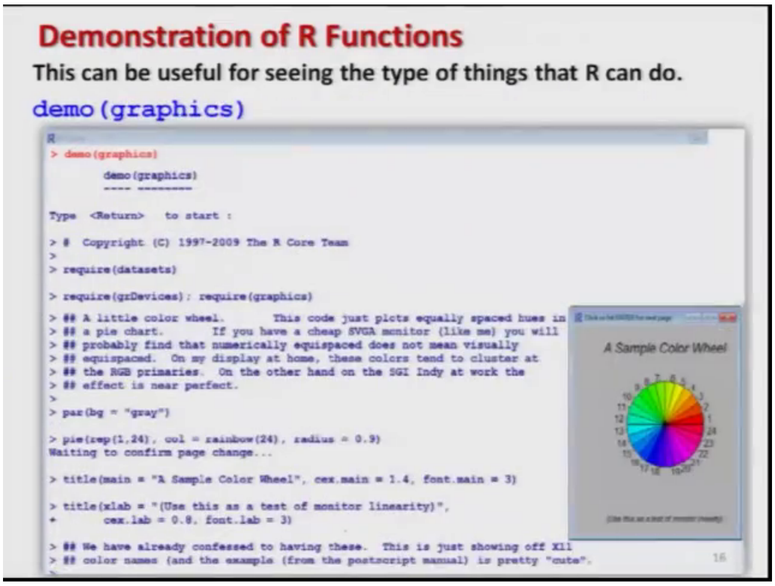
(Refer Slide Time: 17:55)



```
> require(grDevices); require(graphics)

> ## (1) The Obligatory Mathematical surface.
> ##     Rotated sinc function.
>
> x <- seq(-10, 10, length.out = 50)

> y <- x

> rotsinc <- function(x,y)
+ {
+     sinc <- function(x) { y <- sin(x)/x ; y[is.na(y)] <- 1; y }
+     10 * sinc( sqrt(x^2+y^2) )
+ }

> sinc.exp <- expression(z == Sinc(sqrt(x^2 + y^2)))

> z <- outer(x, y, rotsinc)

> oldpar <- par(bg = "white")

> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue$
Waiting to confirm page change...
```

So, I have just copied and pasted its information and it will it will also give you a lot here plot here something like this which I have made here, like this over here and these are the details which we are which are available over there. So, you can see here that it is trying to give you all the details.

(Refer Slide Time: 18:28)



Demonstration of R Functions

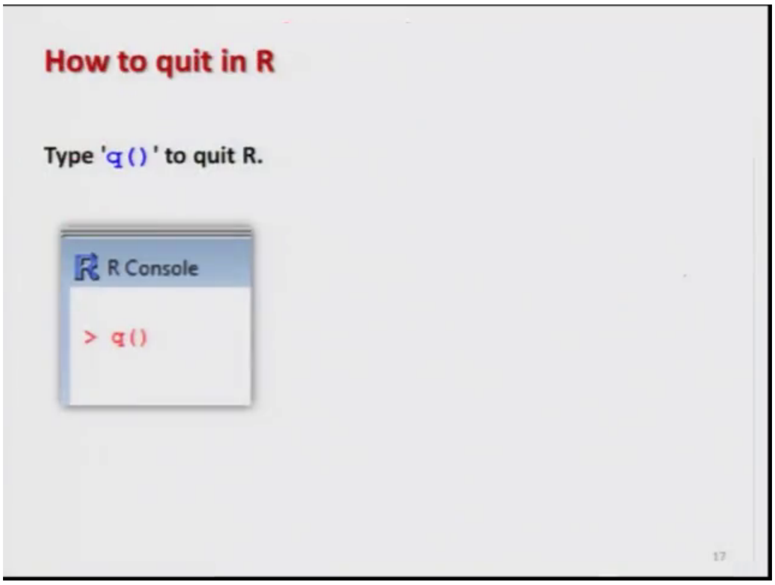
This can be useful for seeing the type of things that R can do.

demo(graphics)

```
> demo(graphics)

        demo(graphics)
        ---- ---------

Type  <Return>   to start :

> #  Copyright (C) 1997-2009 The R Core Team
>
> require(datasets)

> require(grDevices); require(graphics)

> ## A little color wheel.       This code just plots equally spaced hues in
> ## a pie chart.       If you have a cheap SVGA monitor (like me) you will
> ## probably find that numerically equispaced does not mean visually
> ## equispaced.  On my display at home, these colors tend to cluster at
> ## the RGB primaries.  On the other hand on the SGI Indy at work the
> ## effect is near perfect.
>
> par(bg = "gray")

> pie(rep(1,24), col = rainbow(24), radius = 0.9)
Waiting to confirm page change...

> title(main = "A Sample Color Wheel", cex.main = 1.4, font.main = 3)

> title(xlab = "(Use this as a test of monitor linearity)",
+       cex.lab = 0.8, font.lab = 3)

> ## We have already confessed to having these.  This is just showing off X11
> ## color names (and the example (from the postscript manual) is pretty "cute".
```

Similarly, if I want to have another function to be demonstrated, so I have to simply use here the demo and inside the bracket graphics and once you try to do it yourself you will see that you will get here this type of a screen and that is containing here all possible details. And also it will try to give you here a graphic like this one which is available over there, at this moment it does not make much sense to go through with the details what are given on the slide, but my modest objective is to show you that how you can get a demonstration of a particular function which you want to use right.

(Refer Slide Time: 19:05)



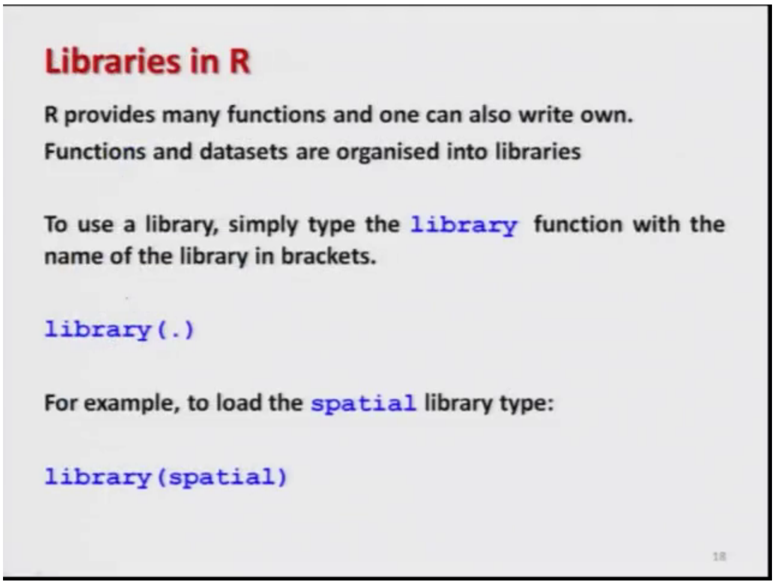
How to quit in R

Type 'q()' to quit R.

```
> q()
```

The next thing is this that if you are using R and if you want to quit from r. So, how to quit?

(Refer Slide Time: 19:33)



Libraries in R

R provides many functions and one can also write own.
Functions and datasets are organised into libraries

To use a library, simply type the library function with the name of the library in brackets.

```
library(.)
```

For example, to load the spatial library type:

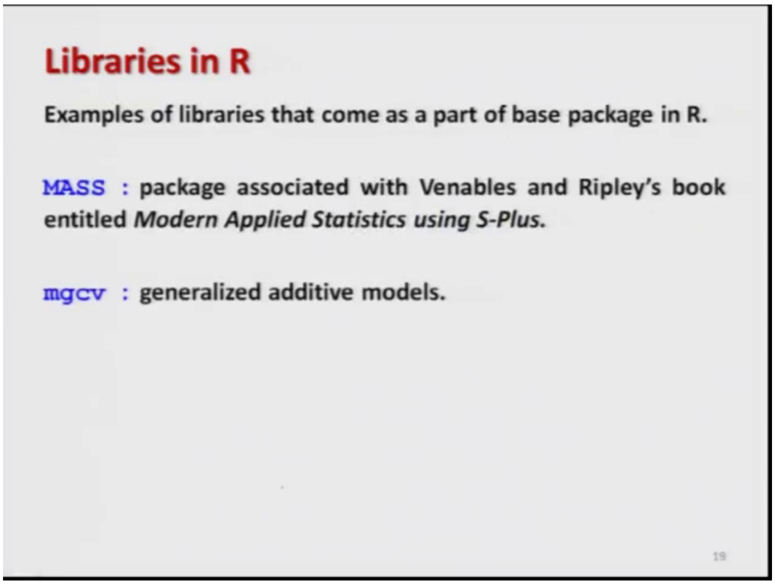
```
library(spatial)
```

In order to quit you simply have to write here q and this this starting and closing brackets. So, if you try to see over here I have given here a screenshot also. So, if you try to write q and the and these 2 brackets then you can easily quit the R program also. The next aspect on which I would like to concentrate is the libraries in R, the first question comes, what is a library? So, if you try to see in simple words the literal meaning of the library is that a place where there is a collection of many books right. So simple, so similarly this libraries are also the collection of several types of commands to execute different types of task and R provides many many functions and out of which there are 2 types of function one function which are built in inside the R software and say another types of software which you can create.

Beside those things ah when you try to install the R software you will see that there are several options given, one options is to that when you are starting the R for the first time then you are actually downloading the base software. So, in the base software they have given most of the say this common commands which are useful for a commands user and beside that if you want to use any specialized function then they have given the commands for that, for that specialized task inside a library for example, in case if I want to fit the times series model. So, there will be a library for the time series, suppose if I want to fit a fit a special data model so there will be another library that is dealing with the spatial data.

So, in order to use those functions we have to download the libraries separately and we need to install them, then only I can execute those things. So, in case if you want to use a library you have to type the function here library and inside the bracket you have to give the name of the library which you want to use for example, if I want to do any statistical data analysis for say spatial. So, I have to use here a library s p a t i a l say spatial and I will try to use here the commands library say spatial and once I try to do it over here this library will be uploaded and then I can use it.

(Refer Slide Time: 22:03)



Libraries in R

Examples of libraries that come as a part of base package in R.

MASS : package associated with Venables and Ripley's book entitled *Modern Applied Statistics using S-Plus.*

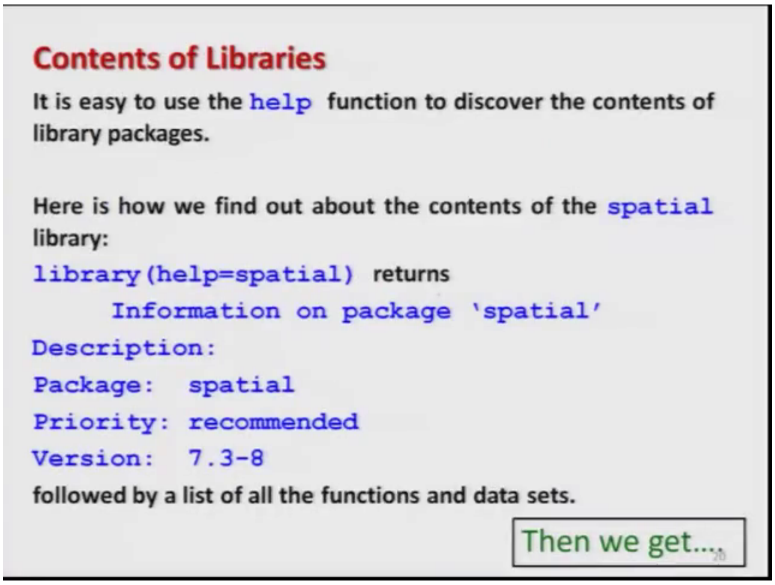
mgcv : generalized additive models.

19

Some built in libraries we wish comes as a part of the base package in R they are something like here one is MASS and say another is mgcv and please remember one thing this is capital m capital a and capital s s. This mass package actually this contains various type of data set and so the tools which are related to a book modern applied statistics using s plus which was written by venables and Ripley's at least that was the book about the software s plus in which they have use different types of data set different types of command and this library contains all those commands over here.

Similarly, there is another library mgcv. So, this library contains about the details about the generalized additive model. So, if you want to use some generalized additive models, then you have to first load this library and only after that you can use it and your order to load a library simply right library and inside the brackets the name of the library right.

(Refer Slide Time: 23:21)



Contents of Libraries

It is easy to use the `help` function to discover the contents of library packages.

Here is how we find out about the contents of the `spatial` library:

`library(help=spatial)` returns

```
     Information on package 'spatial'
Description:
Package:  spatial
Priority: recommended
Version:  7.3-8
```

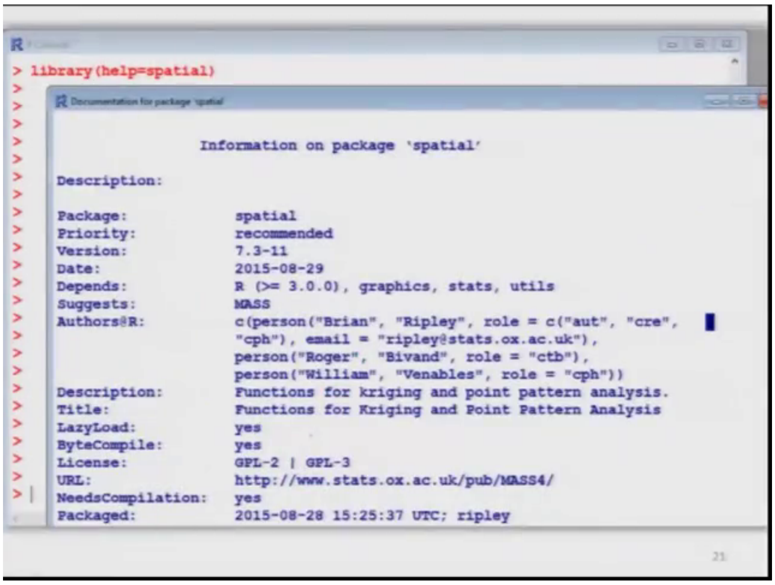
followed by a list of all the functions and data sets.

Then we get...

So, once you load a library means; obviously, you would like to know what are the contents of that library because as such you have no idea. So, I can use the help function to discover the contents of a library package for example, earlier I had just install the package see here a spatial.

Now, I want to know that what is there in this say spatial package and I need to know the help. So, I will try to write down here the library to help equal to the package name. So, this is help equal to say spatial and once I try to once I try to execute it I will get here this type of screenshot, right mean the package name is spatial priority this is good this is the version and after this, this will give you all the details about this package.

(Refer Slide Time: 24:30)



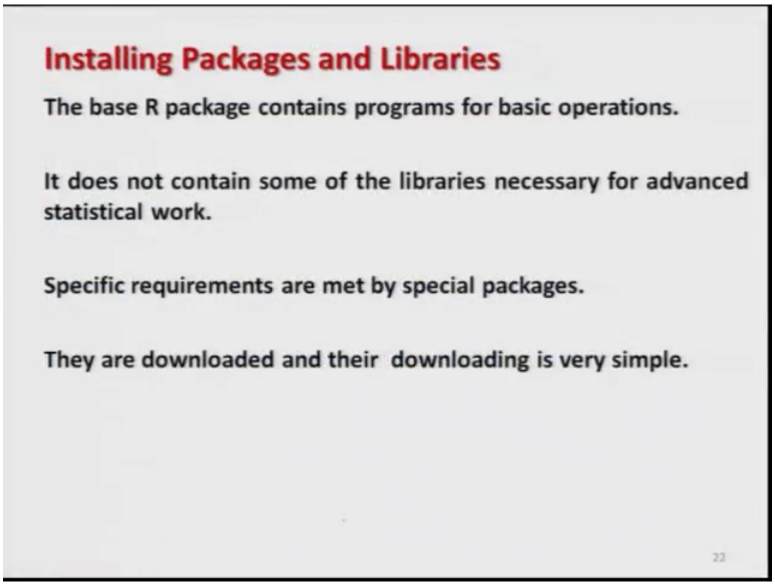
```
> library(help=spatial)

                Information on package 'spatial'

Description:

Package:            spatial
Priority:           recommended
Version:            7.3-11
Date:               2015-08-29
Depends:            R (>= 3.0.0), graphics, stats, utils
Suggests:           MASS
Authors@R:          c(person("Brian", "Ripley", role = c("aut", "cre",
                    "cph"), email = "ripley@stats.ox.ac.uk"),
                    person("Roger", "Bivand", role = "ctb"),
                    person("William", "Venables", role = "cph"))
Description:        Functions for kriging and point pattern analysis.
Title:              Functions for Kriging and Point Pattern Analysis
LazyLoad:           yes
ByteCompile:        yes
License:            GPL-2 | GPL-3
URL:                http://www.stats.ox.ac.uk/pub/MASS4/
NeedsCompilation:   yes
Packaged:           2015-08-28 15:25:37 UTC; ripley
```

21

For sample here you can see this is the screenshot of the library help equal to spatial, it is trying to give you here the about the package priority version the date and it was incorporated in the R software and then suggestions and who has attained who are the authors of this package and simple in here that what type of things it is going to do and you can see here that there are different types of information about the details about the package see here is spatial.

So, similarly if you want to know about any other package you can take this help also.

(Refer Slide Time: 25:16)



Installing Packages and Libraries

The base R package contains programs for basic operations.

It does not contain some of the libraries necessary for advanced statistical work.

Specific requirements are met by special packages.

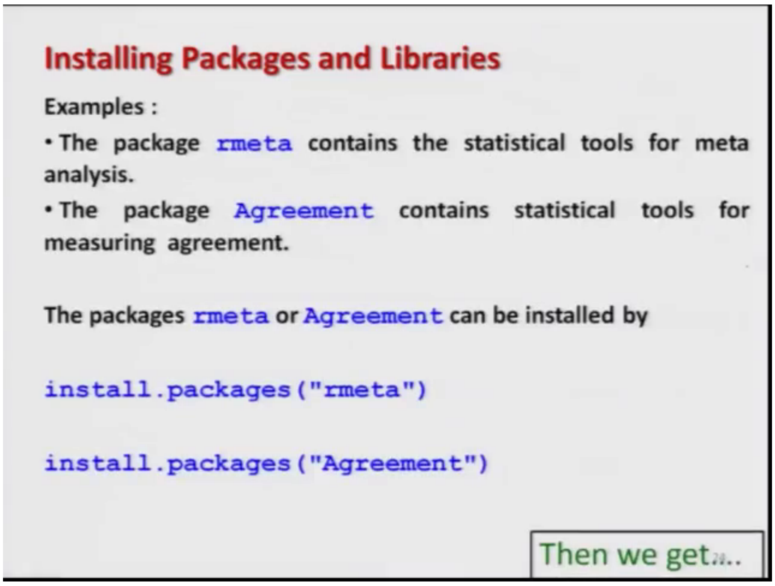
They are downloaded and their downloading is very simple.

22

Now, before you try to use this library or say packages we need to install them, there are some packages which comes as a part of the base package of R and there are certain packages which need to be installed externally right and this packages have different types of qualities. They are used for different types of task and the R package does not contain all this packages, we have to download it from the website of the R software. So, here we try to learn here how we are going to do right. So, first thing is this that we have to do downloading of the software, but believe me this downloading is very simple.

So, now if you want to install any package first we first establish that run the R program and then use the command install dot packages and as soon as you say install dot packages, then the package we will be downloaded and it will be installed.

(Refer Slide Time: 26:30)



Installing Packages and Libraries

Examples :

- The package `rmeta` contains the statistical tools for meta analysis.
- The package `Agreement` contains statistical tools for measuring agreement.

The packages `rmeta` or `Agreement` can be installed by

```
install.packages("rmeta")

install.packages("Agreement")
```

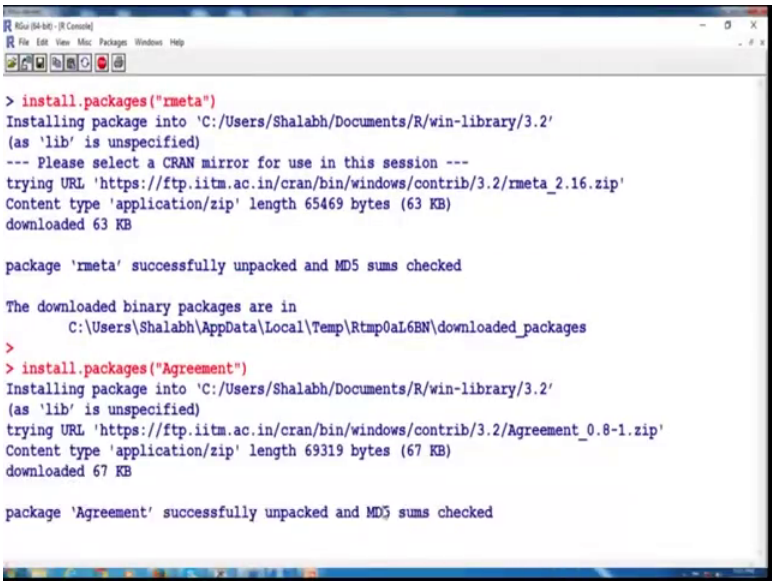
Then we get...

So, let us try to take some example and we try to understand how these things are happening for example, suppose I want to use here a package R meta, R meta is a software which is used in statistics for meta analysis of the data and this package does not come as a part of the base software right. Similarly, there is another package called as agreement and this package contains the statistical tools for measuring the agreement.

This package also does not come as a part of the bay software. So, we need to install these two software, one is r meta and say another is agreement right. So, now, we have these two packages r meta and agreement that we want to install. So, now, you can see this is pretty simple, you simply have to write down install dot packages and inside these two brackets you have to simply write R meta inside the double quotes and similarly if you want to installs the package agreement then you have to simply type here install package and inside the bracket, these two bracket you have to write down this agreement inside the double quotes and once I try to do it here I get here a screenshot like this one. So, let us try to see and verify the screenshot first.

For example if I want to install here this package over here.

(Refer Slide Time: 28:23)



RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

```
> install.packages("rmeta")
Installing package into 'C:/Users/Shalabh/Documents/R/win-library/3.2'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
trying URL 'https://ftp.iitm.ac.in/cran/bin/windows/contrib/3.2/rmeta_2.16.zip'
Content type 'application/zip' length 65469 bytes (63 KB)
downloaded 63 KB

package 'rmeta' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Shalabh\AppData\Local\Temp\Rtmp0aL6BN\downloaded_packages
>
> install.packages("Agreement")
Installing package into 'C:/Users/Shalabh/Documents/R/win-library/3.2'
(as 'lib' is unspecified)
trying URL 'https://ftp.iitm.ac.in/cran/bin/windows/contrib/3.2/Agreement_0.8-1.zip'
Content type 'application/zip' length 69319 bytes (67 KB)
downloaded 67 KB

package 'Agreement' successfully unpacked and MD5 sums checked
```

So, now, if I try to copy and paste this command over here you see I get here something like this and it is asking from where you want to download this package is now available in Algeria, Australia, and different countries. So, if I say I want to download it from say India and I gave here the name India here and then you can see here with this package installed over here right. Similarly if I want to install here this another package say agreement, what I have to do here I simply have to type here install dot packages and then you will see here that this package has been installed here right ok.

So, now I have given these things on my slides also and you can see here that is the same slide which we had obtain right and all these say packages they will be downloaded in a directory over there and this directory is given over here where the packages have been downloaded. So, that is not a very big deal and this process is automatic right. So, you can see here that is the slide that we have got after installing the agreement package this is trying to give you different types of information the size of those packages and everything right ok.

So, after this we stop here and now we have seen in this lecture that whenever we want to execute a particular type of task there are 2 options, the task can be completed using the packages which are contained inside the base package or they can be executed by using additional package. So, in case if I want to use additional package that has to be downloaded from the website of the R software, but for that you need not to go to website again and again you simple need have to an internet connection and if you try to give the command install dot packages.

Then it will automatically install the package into your software, but remember one thing that whenever you are going to use the package you first need to upload it and all those aspects we will try to look into the look into in the next lectures. So, try to play with this commands try to type it yourself in your computer and try to see what do you get the more you play with the software you will get better acquainted and then and then you will enjoy the learning more in the forthcoming lectures till then goodbye.