

Exploratory Statistical Data Analysis With R Software (ESDAR)

Swayam Prabha

Lecture 3

Working with R Software

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Slides can be downloaded from
<http://home.iitk.ac.in/~shalab/sp>



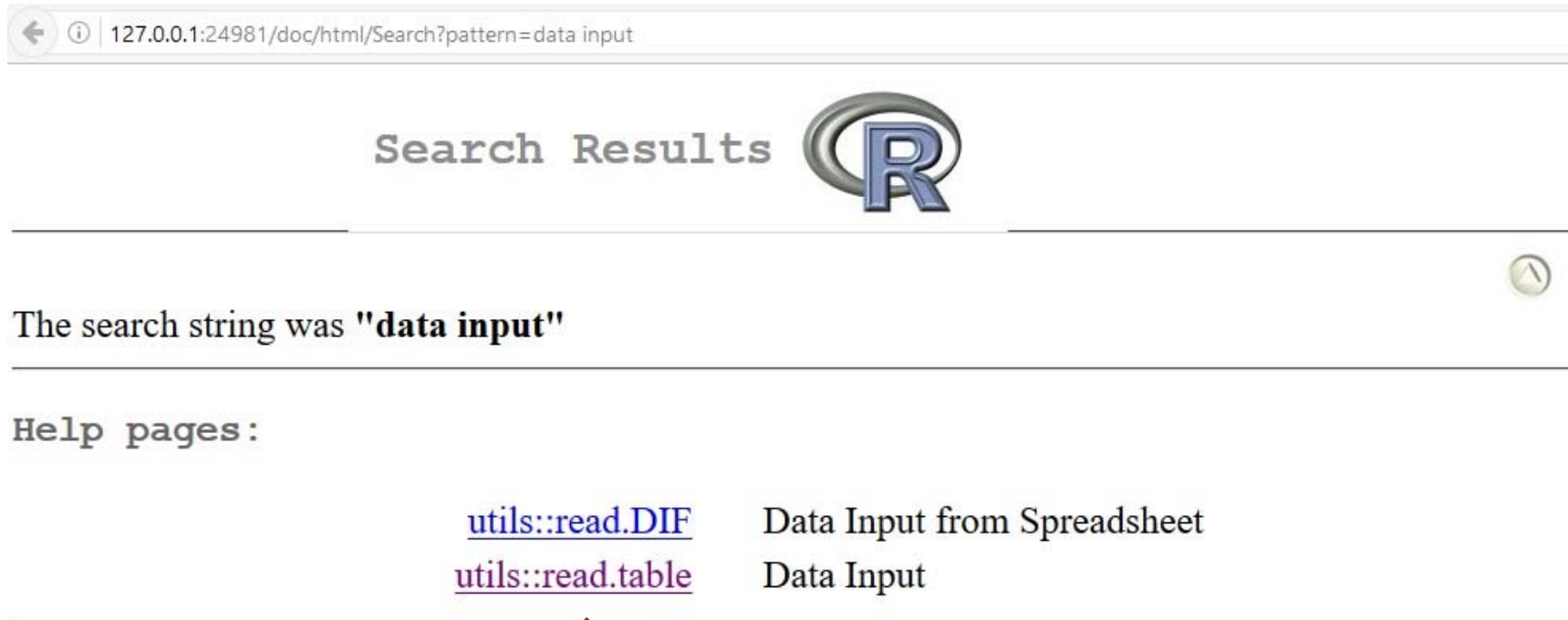
Getting Help in R

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

Then we get....

Then we get....



The screenshot shows a web browser window with the address bar containing "127.0.0.1:24981/doc/html/Search?pattern=data input". The page title is "Search Results" followed by the R logo. Below the title, it states "The search string was 'data input'". Underneath, there is a section titled "Help pages:" which lists two search results: "[utils::read.DIF](#) Data Input from Spreadsheet" and "[utils::read.table](#) Data Input". A red arrow points to the "[utils::read.table](#)" link.

Clicking over the link give required information

Getting Help in R

'`help()`' for on-line help,

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

 R Console

```
> help()
```

```
> help.start()
```

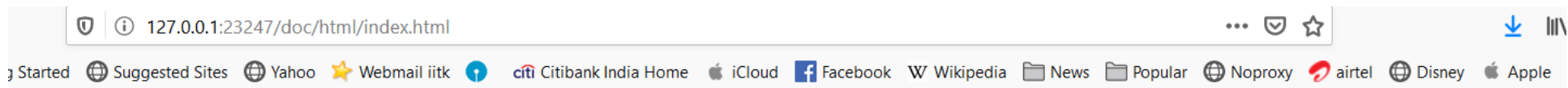
```
If nothing happens, you should open
```

```
'http://127.0.0.1:13077/doc/html/index.html' yourself
```

```
> |
```

Then we get....

Getting Help in R



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](#)

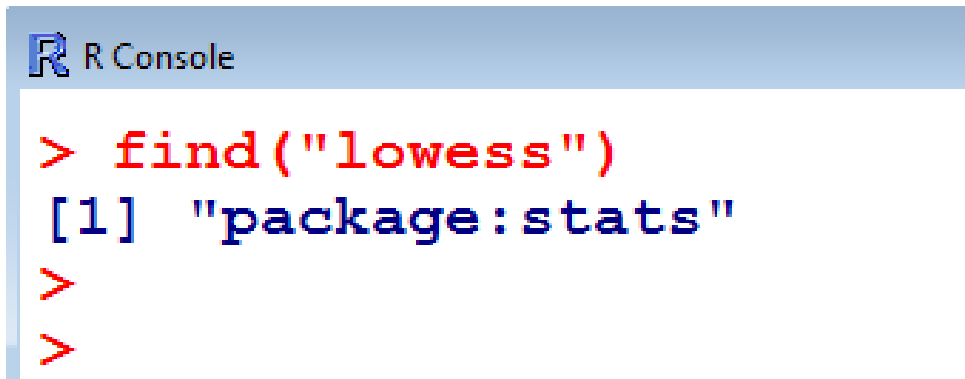
Getting Help in R

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

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

For example

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



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

Getting Help in R

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

`apropos("lm")` returns

R Console

```
> 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.glm"              "anova.lm"             "anova.lm"
[13] "anova.glm"              "anova.lm"             "anova.lm"
[16] "anova.glm"              "anova.lm"             "anova.lm"
[19] "anova.glm"              "anova.lm"             "anova.lm"
[22] "anova.glm"              "anova.lm"             "anova.lm"
[25] "anova.glm"              "anova.lm"             "anova.lm"
[28] "anova.glm"              "anova.lm"             "anova.lm"
[31] "anova.glm"              "anova.lm"             "anova.lm"
[34] "anova.glm"              "anova.lm"             "anova.lm"
[37] "anova.glm"              "anova.lm"             "anova.lm"
[40] "anova.glm"              "anova.lm"             "anova.lm"
[43] "anova.glm"              "anova.lm"             "anova.lm"
[46] "anova.glm"              "anova.lm"             "anova.lm"
[49] "anova.glm"              "anova.lm"             "anova.lm"
```

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.


```
> 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

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]

```
R Console
> 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

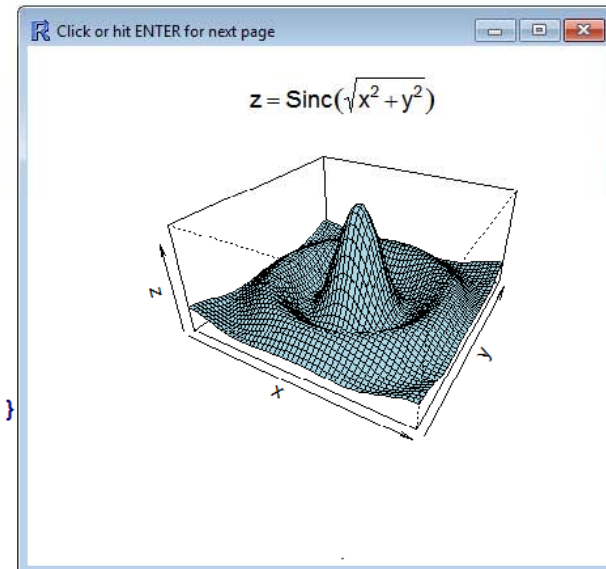
> rotsinc <- function(x,y)
+ {
+   sinc <- function(x) { y <- sin(x)/x ; y[is.na(y)] <- 1; y }
+   10 * sinc( sqrt(x^2+y^2) )
+ }

> 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

Demonstration of R Functions

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

`demo(graphics)`

```
R Console
> demo(graphics)
      demo(graphics)
      ---- ~~~~~

Type <Return> to start :

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

> require(grDevices); require(graphics)

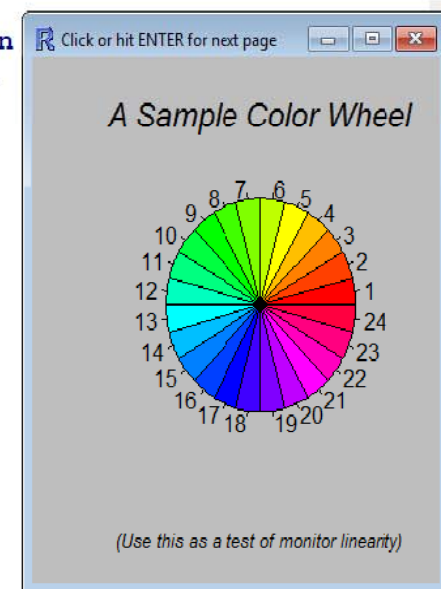
> ## A little color wheel.      This code just plots equally spaced hues in
> ## a pie chart.             If you have a cheap SVGA monitor (like me) you will
> ## probably find that numerically equispaced does not mean visually
> ## equispaced. On my display at home, these colors tend to cluster at
> ## 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".
/
```



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)
```

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.

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....

Contents of Libraries

```
R Console
> library(help=spatial)
> |

Documentation for package '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
Author:           Brian Ripley [aut, cre, cph], Roger Bivand [ctb],
          William Venables [cph]
```

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.

Installing Packages and Libraries

To install any package,

- run the R program,
- then on the command line, use the `install.packages` function to download the libraries we want.

Installing Packages and Libraries

The base R package contains some necessary libraries only.

Other libraries are required for advanced statistical work which are downloaded and installed as and when required.

Run the R program, then use the `install.packages` command to download the libraries.

Examples :

```
install.packages("ggplot2") : installs package ggplot2
```

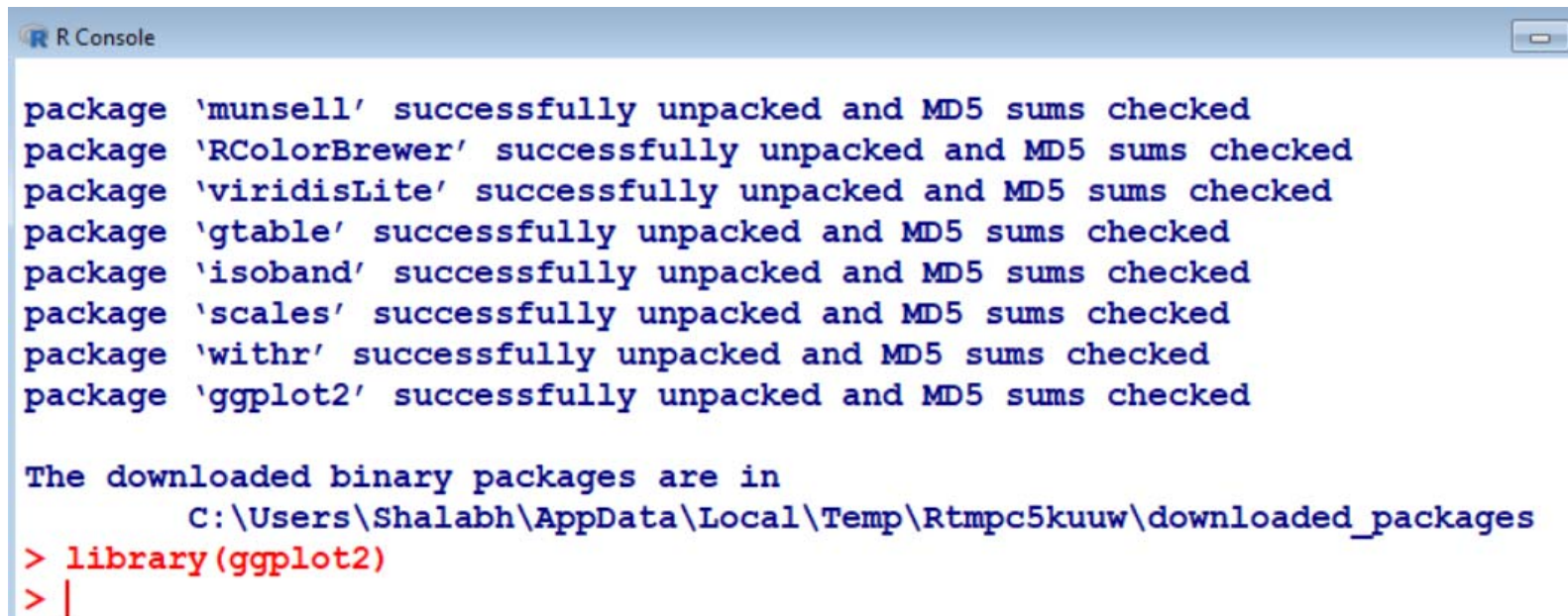
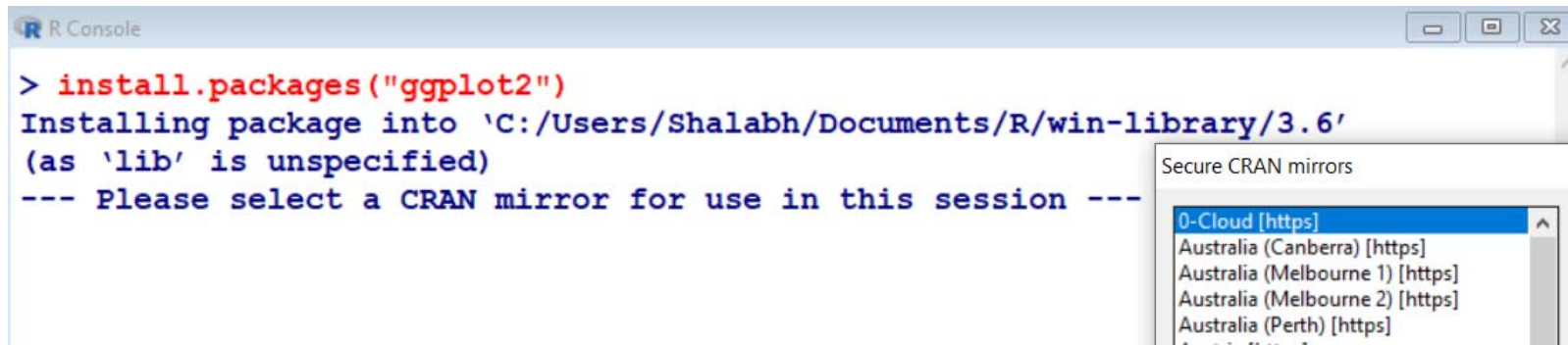
```
install.packages("agricolae") : installs package agricolae
```

```
install.packages("DoE.base") : installs package DoE.base
```

Installing Packages and Libraries

Example

```
install.packages("ggplot2")
```



Cleaning up the Windows

We assign names to variables when analyzing any data.

It is good practice to remove the variable names given to any data frame at the end each session in R.

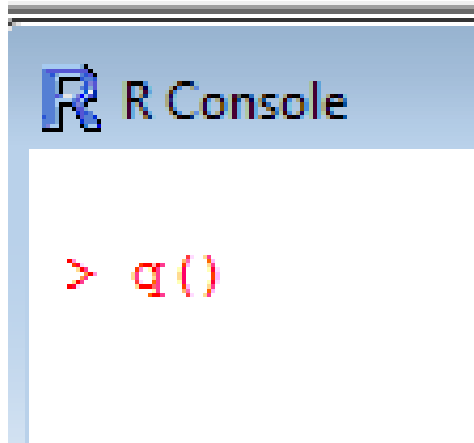
`rm()` command removes variable names

For example,

`rm(x, y, z)` removes the variables `x`, `y` and `z`.

How to quit in R

Type `q()` to quit R.



```
R Console  
> q()
```