

R #1 Activity - Tasks

1. Familiarize yourself with RStudio (help, history, workspace)
2. Install and update packages
3. Install and run the Rcmdr package
4. Familiarize yourself with Rcmdr and Help
5. Access and manipulate data
6. Look over Rcmdr metadata (Fox, 2005)
7. Explore making plots / doing stats with Rcmdr

R Studio Free-ware

Download R Studio: (<http://www.rstudio.com/>)

Welcome to RStudio - Open source
and enterprise-ready professional
software for R

Download RStudio

Discover Shiny

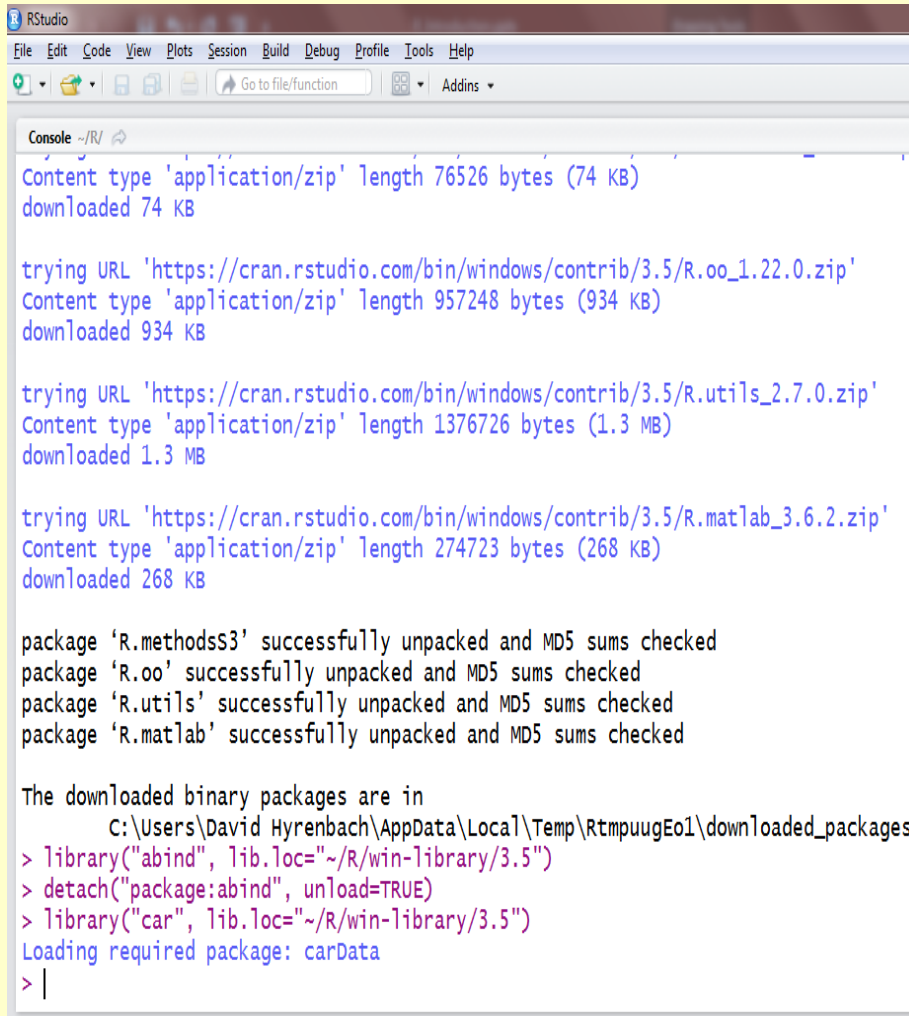


NOTE: You can run R Studio, without first running R.
R studio will automatically open and run R.

R Studio Overview

Explore the Multiple Windows

Console:



```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
Console ~R/
Content type 'application/zip' length 76526 bytes (74 KB)
downloaded 74 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/R.oo_1.22.0.zip'
Content type 'application/zip' length 957248 bytes (934 KB)
downloaded 934 KB

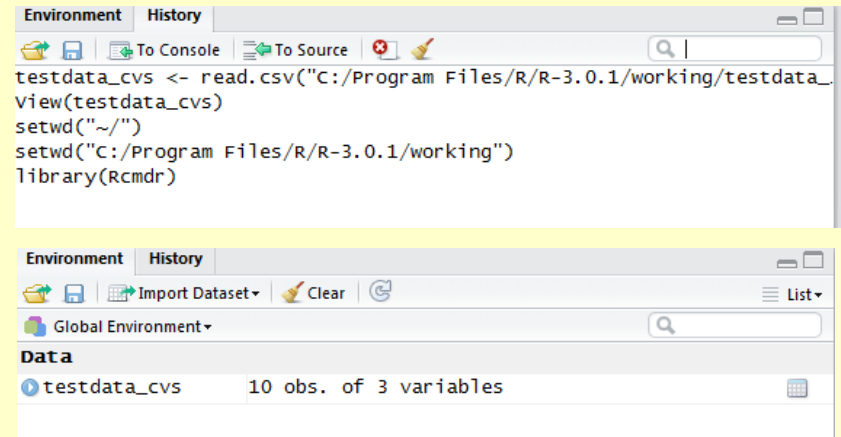
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/R.utils_2.7.0.zip'
Content type 'application/zip' length 1376726 bytes (1.3 MB)
downloaded 1.3 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/R.matlab_3.6.2.zip'
Content type 'application/zip' length 274723 bytes (268 KB)
downloaded 268 KB

package 'R.methodsS3' successfully unpacked and MD5 sums checked
package 'R.oo' successfully unpacked and MD5 sums checked
package 'R.utils' successfully unpacked and MD5 sums checked
package 'R.matlab' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\David Hyrenbach\AppData\Local\Temp\RtmpuugEo1\downloaded_packages
> library("abind", lib.loc=~R/win-library/3.5")
> detach("package:abind", unload=TRUE)
> library("car", lib.loc=~R/win-library/3.5")
Loading required package: carData
> |
```

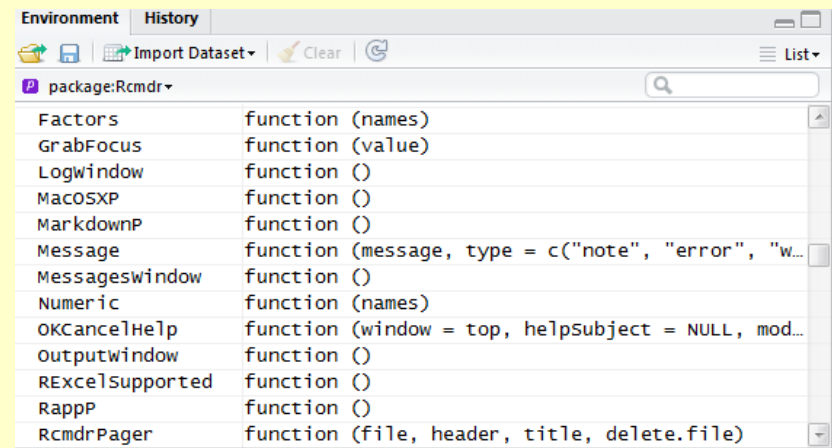
Environment / History:



```
Environment History
To Console To Source
testdata_cvs <- read.csv("C:/Program Files/R/R-3.0.1/working/testdata_
View(testdata_cvs)
setwd("~/")
setwd("C:/Program Files/R/R-3.0.1/working")
library(Rcmdr)

Environment History
Import Dataset Clear
Global Environment
Data
testdata_cvs 10 obs. of 3 variables
```

Files / Plots / Packages / Help



```
Environment History
Import Dataset Clear
package:Rcmdr
Factors function (names)
GrabFocus function (value)
Logwindow function ()
MacOSXP function ()
MarkdownP function ()
Message function (message, type = c("note", "error", "w...
Messageswindow function ()
Numeric function (names)
OKCancelHelp function (window = top, helpsubject = NULL, mod...
Outputwindow function ()
REXcelSupported function ()
RappP function ()
RcmdrPager function (file, header, title, delete.file)
```

R Studio Overview

User-friendly Console

Console ~/ ↻

```
R version 3.4.0 (2017-04-21) -- "You Stupid Darkness"  
Copyright (C) 2017 The R Foundation for Statistical Computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

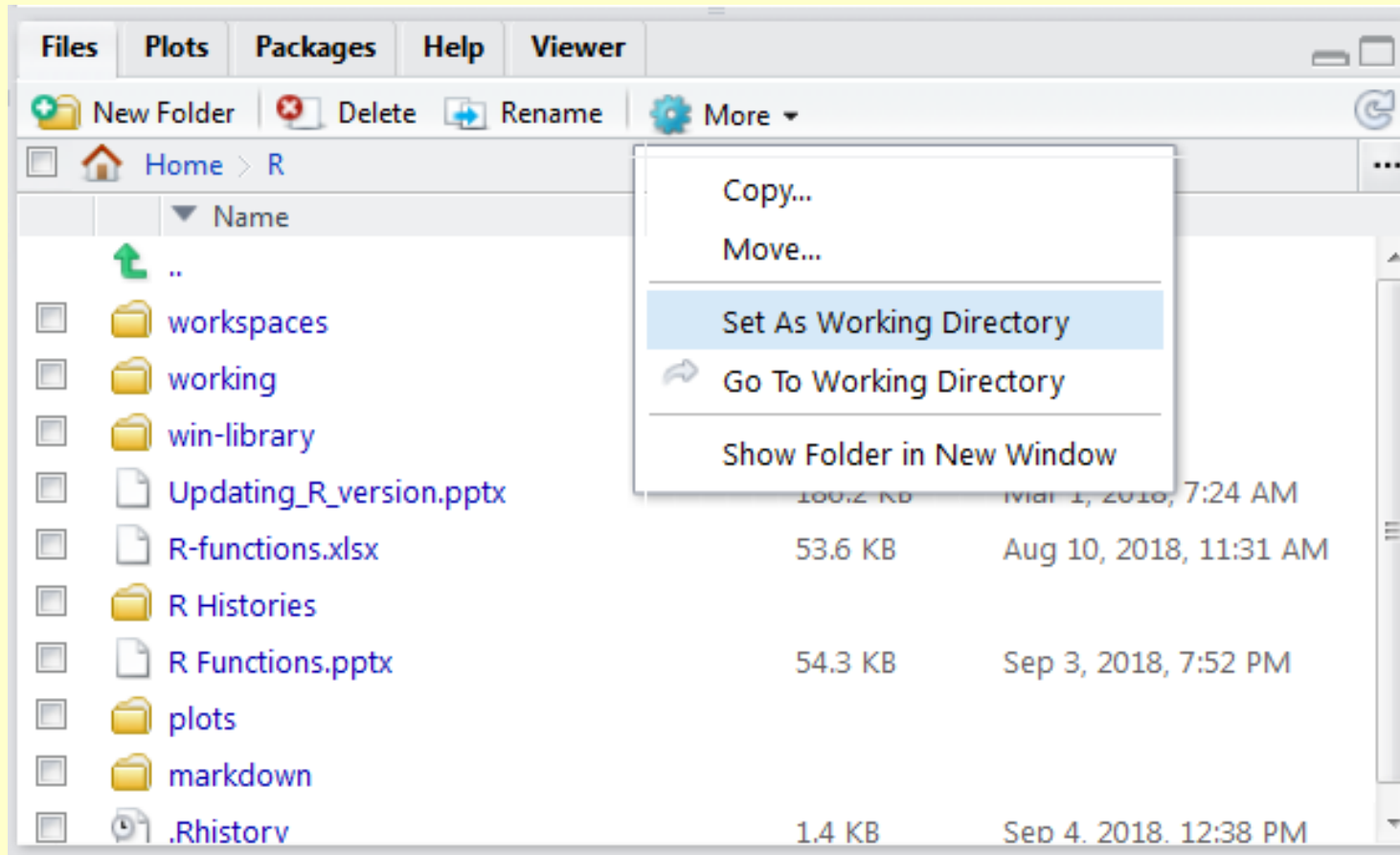
```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.
```

```
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.
```

R Studio Overview

Setting and Going to your Working Directory

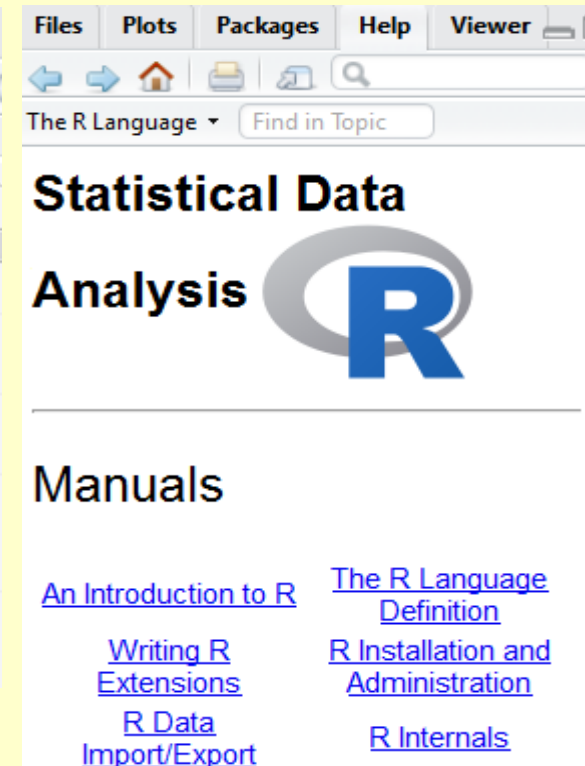
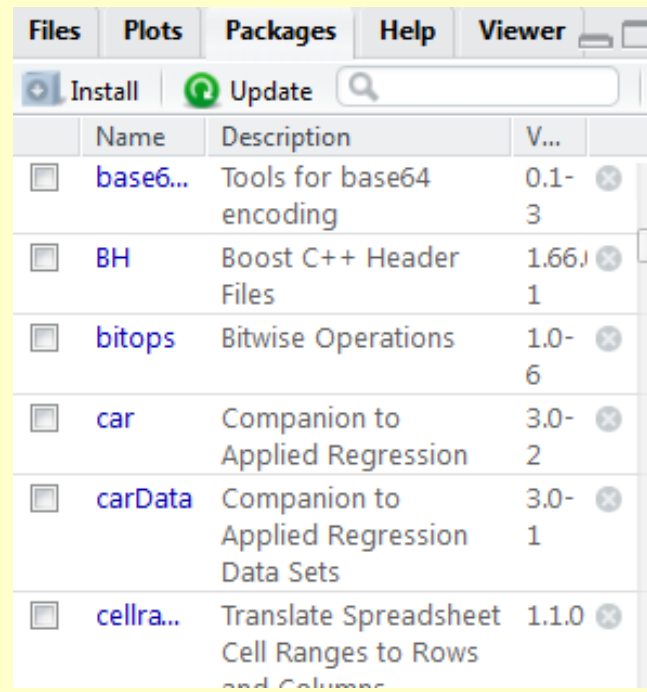
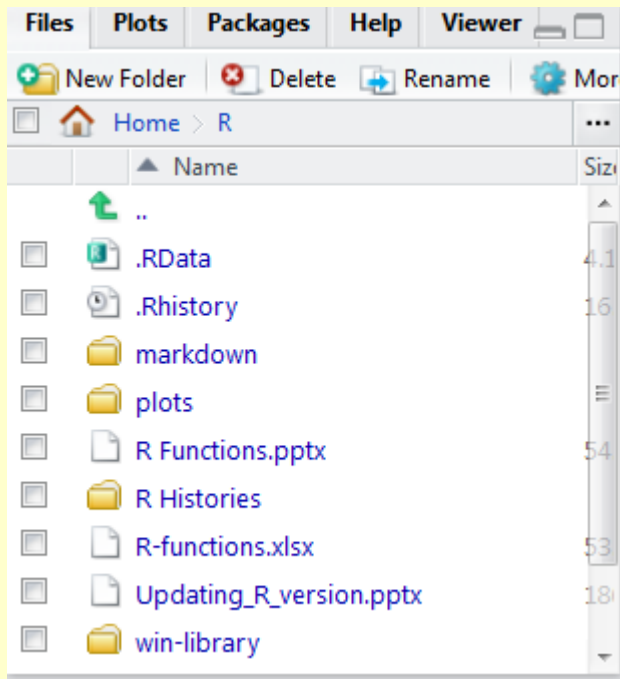
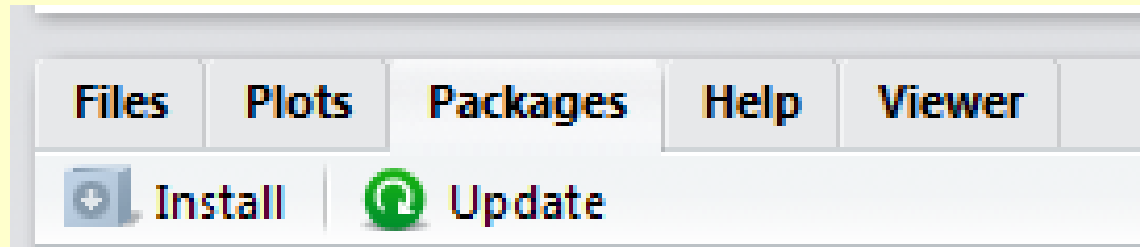


Set WD
by navigating
to a folder

Go to
your WD

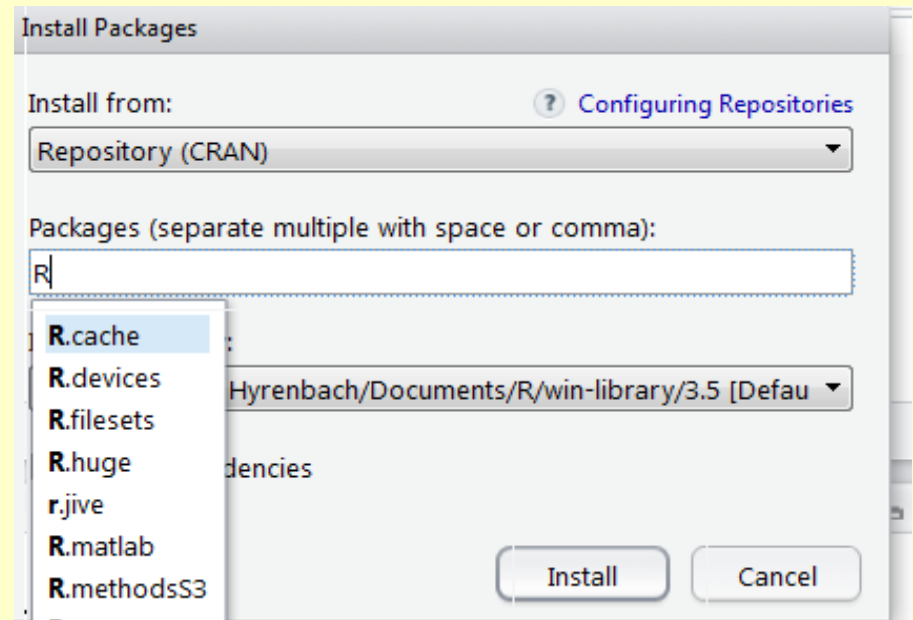
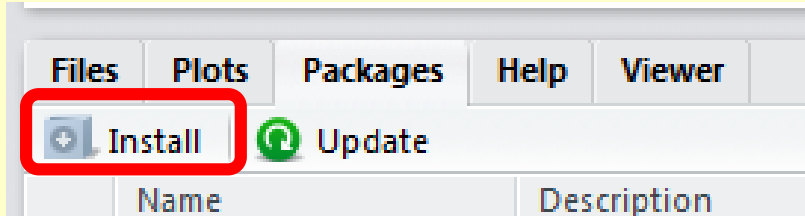
R Studio Overview

Accessing Files / Plots / Packages / Help

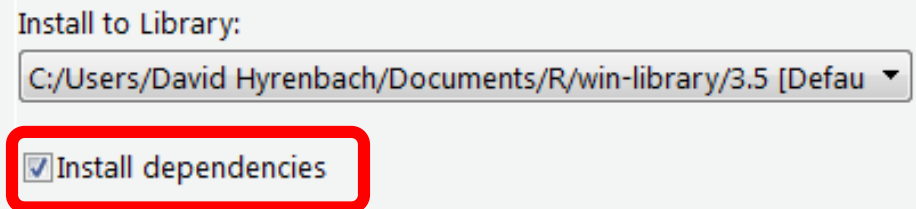


Install Packages

First time: Use install to download into library

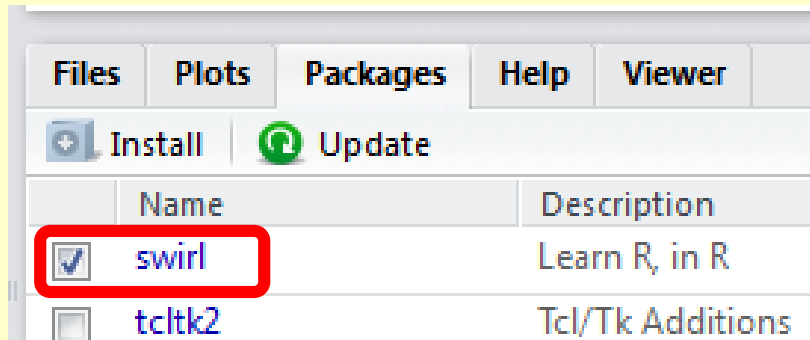


Install dependencies to add all other needed packages



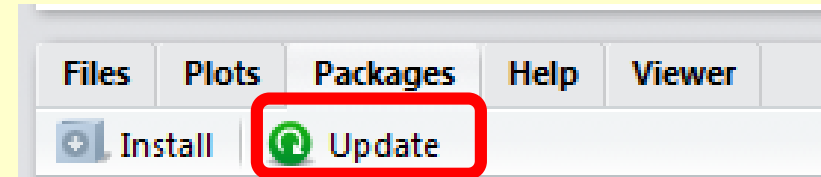
Install Packages

Other times: Just click box to activate package



Updating Packages

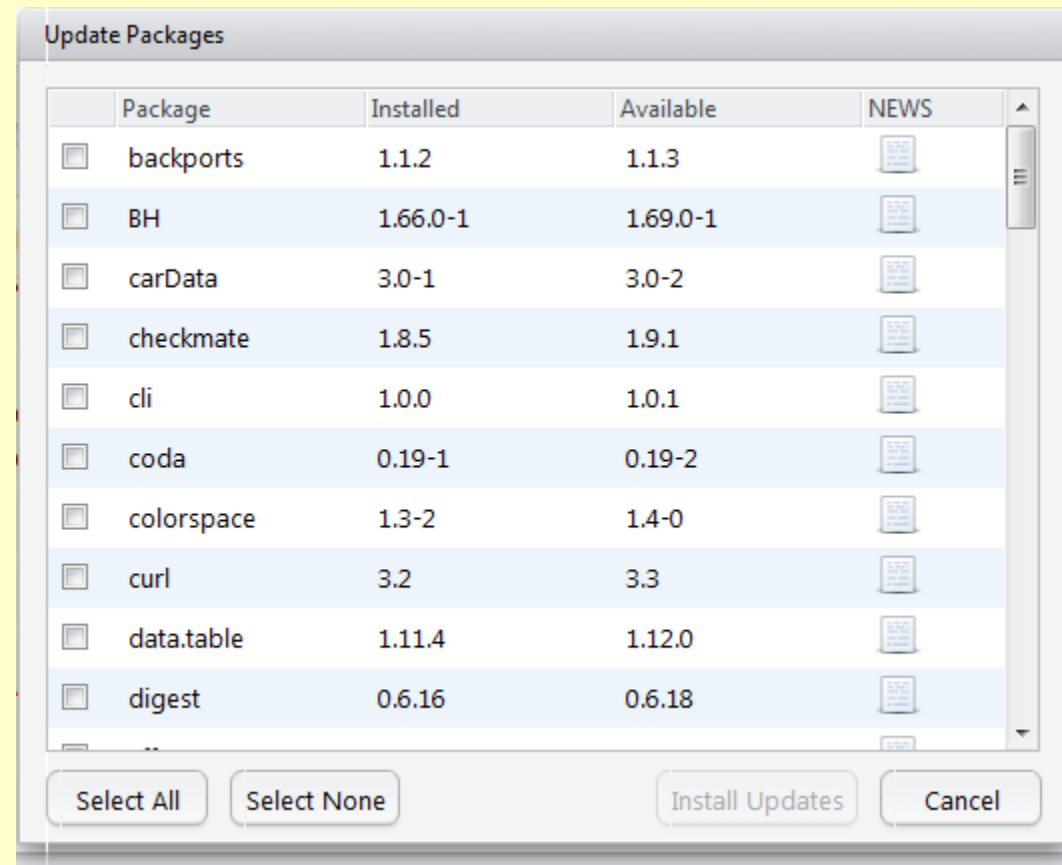
Update Packages



Click the packages
you want to update

Lists packages:
installed vs available

And link to NEWS



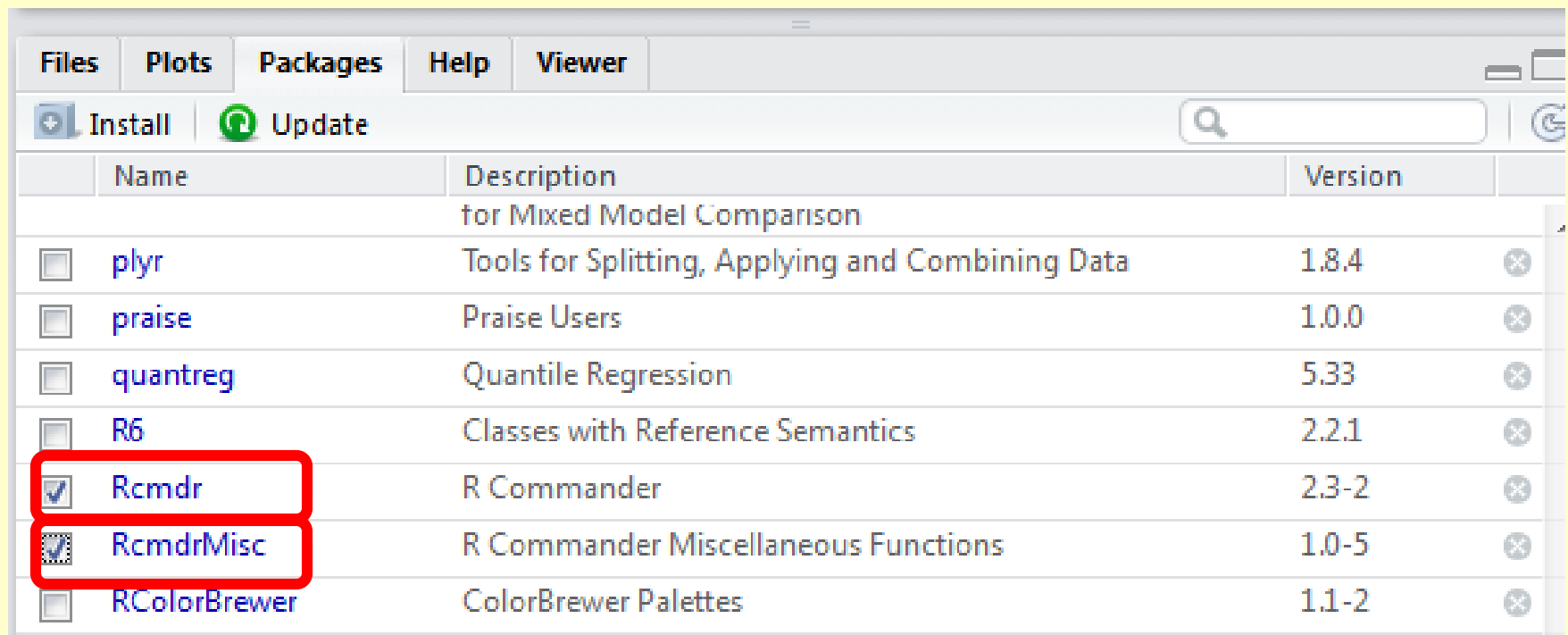
Installing Rcmdr Package

1. Install Rcmdr package
2. Load Rcmdr package
3. A new GUI window will pop-up; which will add functionality to R via menu-driven functions

NOTE: You can install packages using the R console or RStudio. I show you how to do the installation both ways, but you will only need to do it once.

HINT: I suggest using RStudio.

Installing "Rcmdr" Package in R Studio



The screenshot shows the R Studio interface with the 'Packages' pane open. The 'Install' button is highlighted. A table lists several packages, with 'Rcmdr' and 'RcmdrMisc' selected (checked) and highlighted with a red box. The table has columns for Name, Description, and Version.

	Name	Description	Version	
		for Mixed Model Comparison		
<input type="checkbox"/>	plyr	Tools for Splitting, Applying and Combining Data	1.8.4	⊗
<input type="checkbox"/>	praise	Praise Users	1.0.0	⊗
<input type="checkbox"/>	quantreg	Quantile Regression	5.33	⊗
<input type="checkbox"/>	R6	Classes with Reference Semantics	2.2.1	⊗
<input checked="" type="checkbox"/>	Rcmdr	R Commander	2.3-2	⊗
<input checked="" type="checkbox"/>	RcmdrMisc	R Commander Miscellaneous Functions	1.0-5	⊗
<input type="checkbox"/>	RColorBrewer	ColorBrewer Palettes	1.1-2	⊗

```
> library("Rcmdr", lib.loc="~/R/win-library/3.4")
```

```
Loading required package: splines
```

```
Loading required package: RcmdrMisc
```

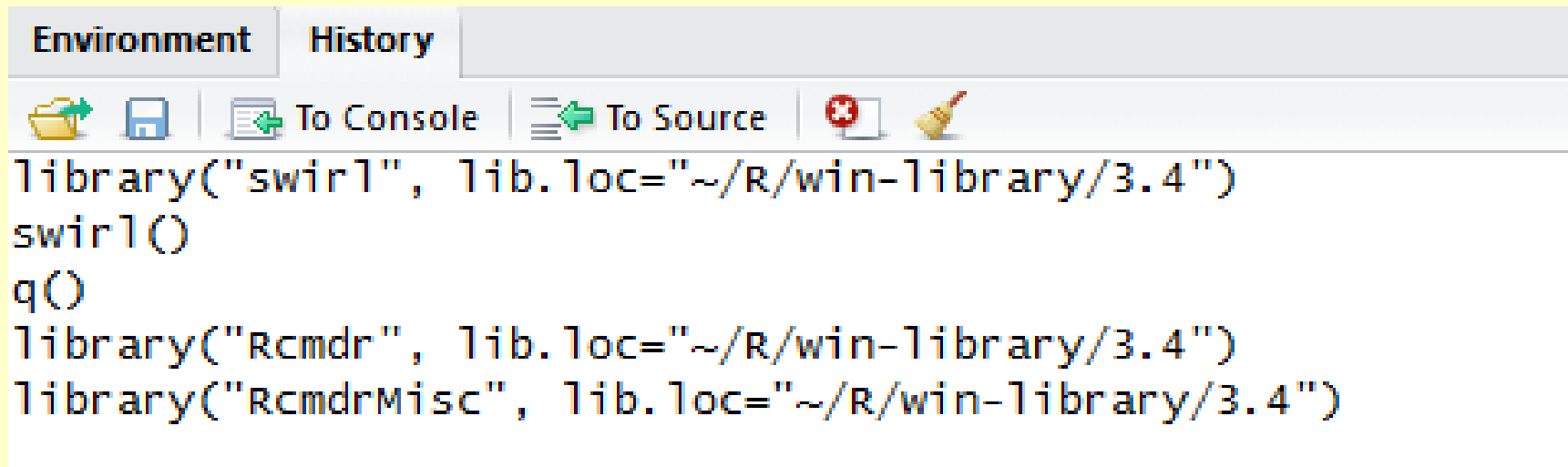
```
Loading required package: car
```

```
Loading required package: sandwich
```

```
RcmdrMsg: [1] NOTE: R Commander Version 2.3-2: Mon Sep 04 13:08:46 2017
```

```
Rcmdr version 2.3-2
```

Check the History Window



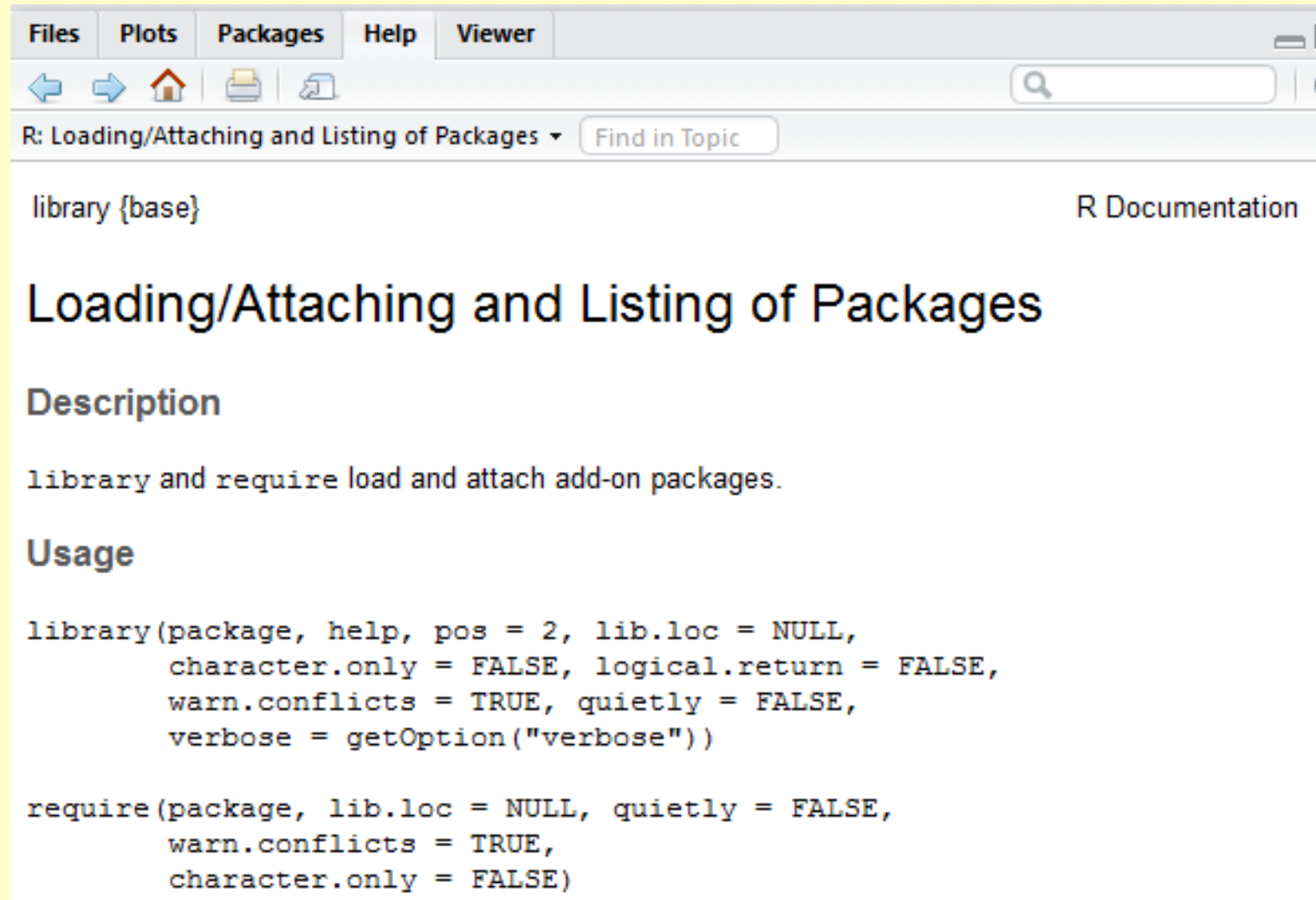
The screenshot shows the RStudio History window. The window has two tabs: "Environment" and "History". The "History" tab is active. Below the tabs is a toolbar with icons for "To Console", "To Source", and a "Clear" button. The history list contains the following commands:

```
library("swirl", lib.loc="~/R/win-library/3.4")
swirl()
q()
library("Rcmdr", lib.loc="~/R/win-library/3.4")
library("RcmdrMisc", lib.loc="~/R/win-library/3.4")
```

> library("Rcmdr", lib.loc="~/R/win-library/3.4")

Check the Function Library

> `help(library)`



The screenshot shows the R help window for the `library` function. The window title is "R: Loading/Attaching and Listing of Packages". The main heading is "Loading/Attaching and Listing of Packages". The description states: "library and require load and attach add-on packages." The usage section shows the following code:

```
library(package, help, pos = 2, lib.loc = NULL,  
        character.only = FALSE, logical.return = FALSE,  
        warn.conflicts = TRUE, quietly = FALSE,  
        verbose = getOption("verbose"))  
  
require(package, lib.loc = NULL, quietly = FALSE,  
        warn.conflicts = TRUE,  
        character.only = FALSE)
```

<http://www.rcommander.com/>

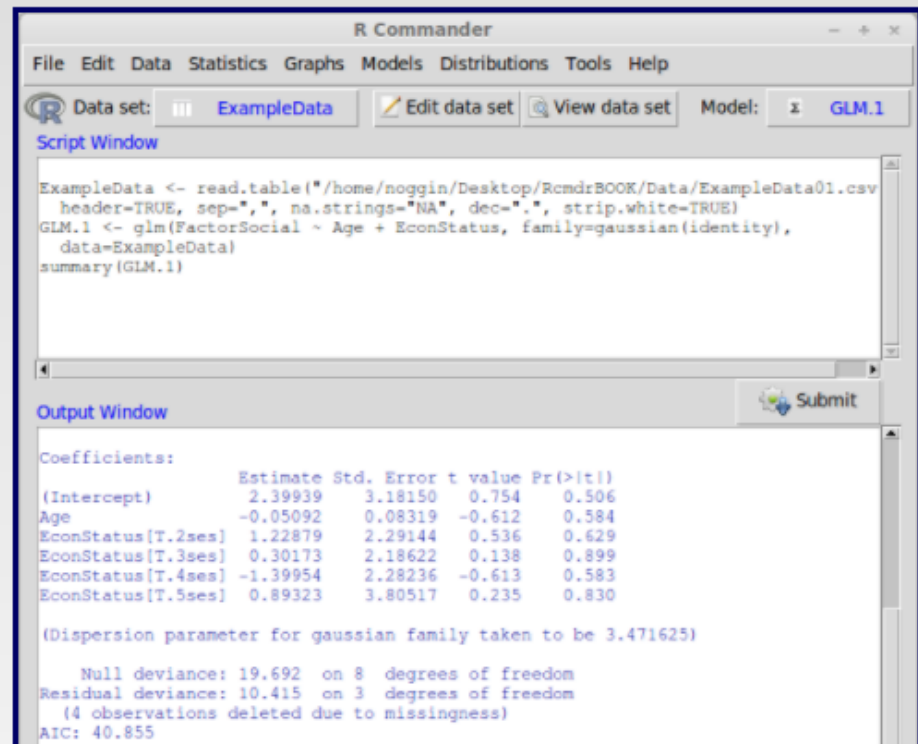
<http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/>

Rcommander a graphical interface for R

R commander (Rcmdr)

R provides a powerful and comprehensive system for analysing data and when used in conjunction with the R-commander (a graphical user interface, commonly known as Rcmdr) it also provides one that is easy and intuitive to use. Basically, R provides the engine that carries out the analyses and Rcmdr provides a convenient way for users to input commands. The Rcmdr program enables analysts to access a selection of commonly-used R commands using a simple interface that should be familiar to most computer users. It also serves the important role of helping users to implement R commands and develop their knowledge and expertise in using the command line --- an important skill for those wishing to exploit the full power of the program.

Information about installing R can be found on the web at the R homepage <http://www.r-project.org/> which provides lots of information about the R project and also directs users to one of the CRAN sites (the Comprehensive R Archive Network) that have been set up on many servers across the world in order for users to download the software. CRAN provides all files necessary to install R on a number of different computing platforms (Linux, MacOS X and Windows) along with detailed information about installation and also offers manuals and contributed documentation in a number of languages and for a number of specific disciplines.



The screenshot shows the R Commander application window. The menu bar includes File, Edit, Data, Statistics, Graphs, Models, Distributions, Tools, and Help. The interface shows a 'Data set:' dropdown set to 'ExampleData', an 'Edit data set' button, a 'View data set' button, and a 'Model:' dropdown set to 'GLM.1'. The 'Script Window' contains the following R code:

```
ExampleData <- read.table("/home/noggin/Desktop/RcmdrBOOK/Data/ExampleData01.csv",
  header=TRUE, sep=";", na.strings="NA", dec=".", strip.white=TRUE)
GLM.1 <- glm(FactorSocial ~ Age + EconStatus, family=gaussian(identity),
  data=ExampleData)
summary(GLM.1)
```

The 'Output Window' displays the following results:

Coefficients:	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.39939	3.18150	0.754	0.506
Age	-0.05092	0.08319	-0.612	0.584
EconStatus[T.2ses]	1.22879	2.29144	0.536	0.629
EconStatus[T.3ses]	0.30173	2.18622	0.138	0.899
EconStatus[T.4ses]	-1.39954	2.28236	-0.613	0.583
EconStatus[T.5ses]	0.89323	3.80517	0.235	0.830

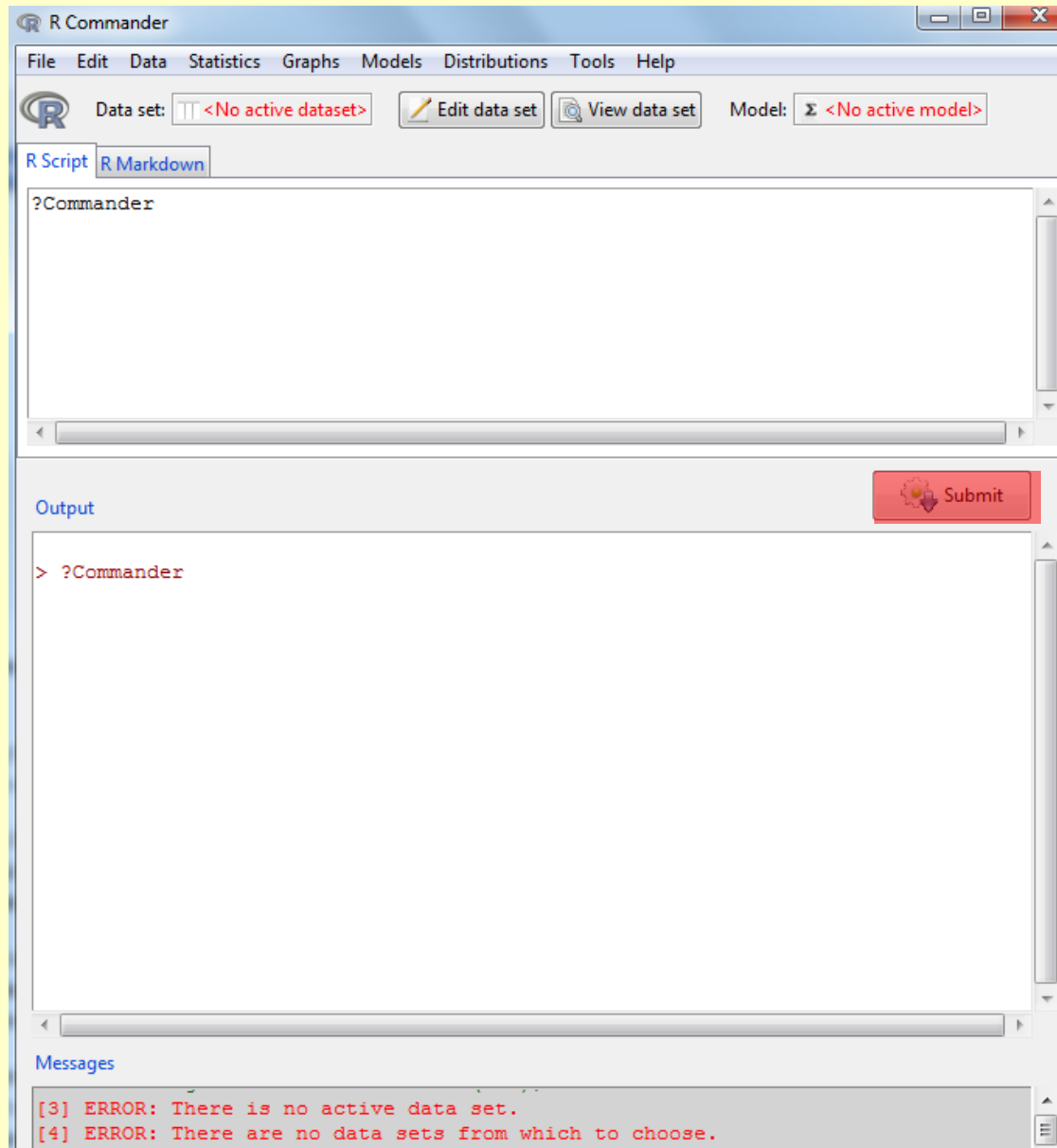
(Dispersion parameter for gaussian family taken to be 3.471625)

Null deviance: 19.692 on 8 degrees of freedom
Residual deviance: 10.415 on 3 degrees of freedom
(4 observations deleted due to missingness)
AIC: 40.855

Running Rcmdr Package

Console Window

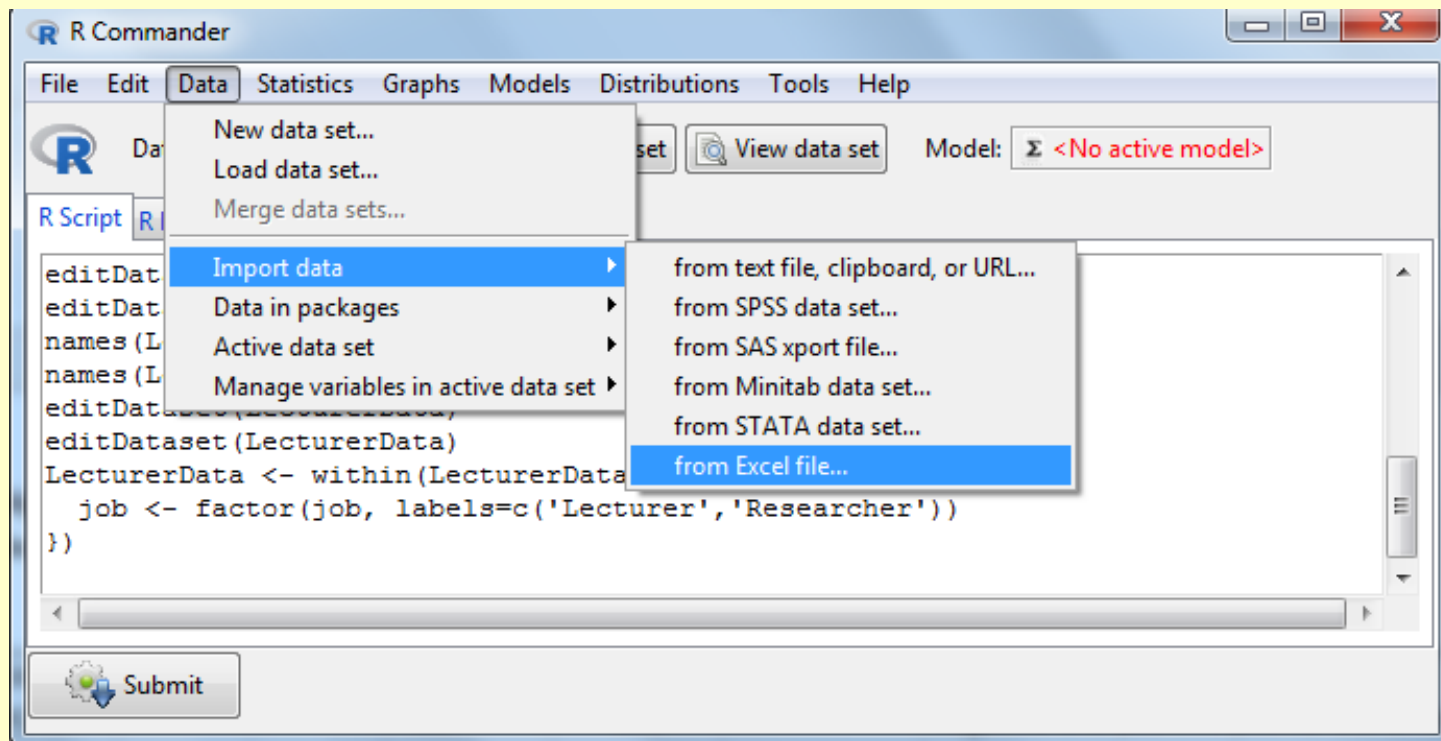
Output Window



Submit Button ("ENTER")

Importing Data into Rcmdr

Note: Excel files ("xlsx", "csv" and ".txt") and SPSS files ("dat") can be easily imported into Rcmdr



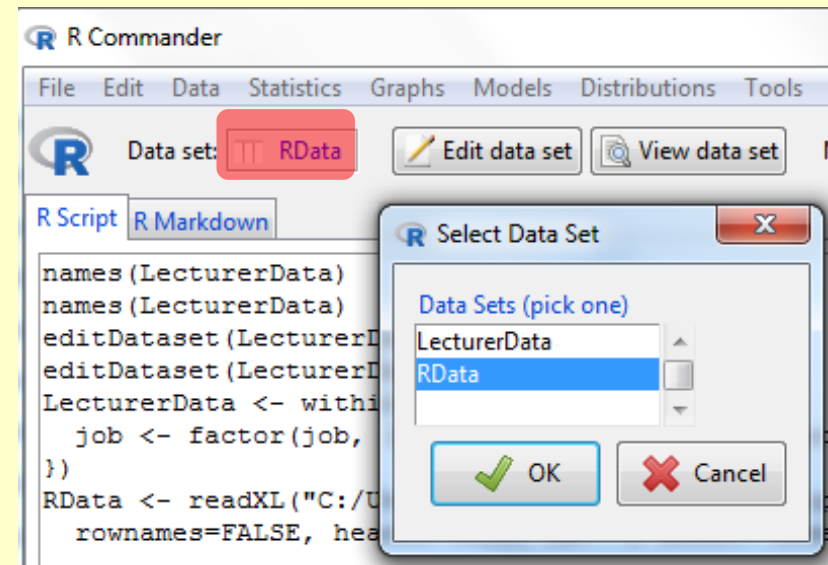
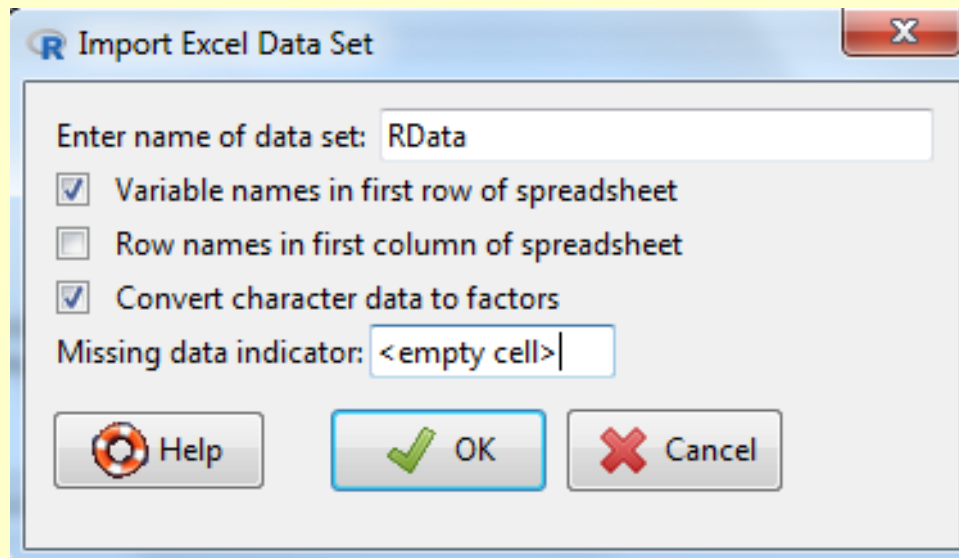
Data / Import Data / from Excel file...

Importing Data into Rcmdr

Import file **RData1.xlsx**

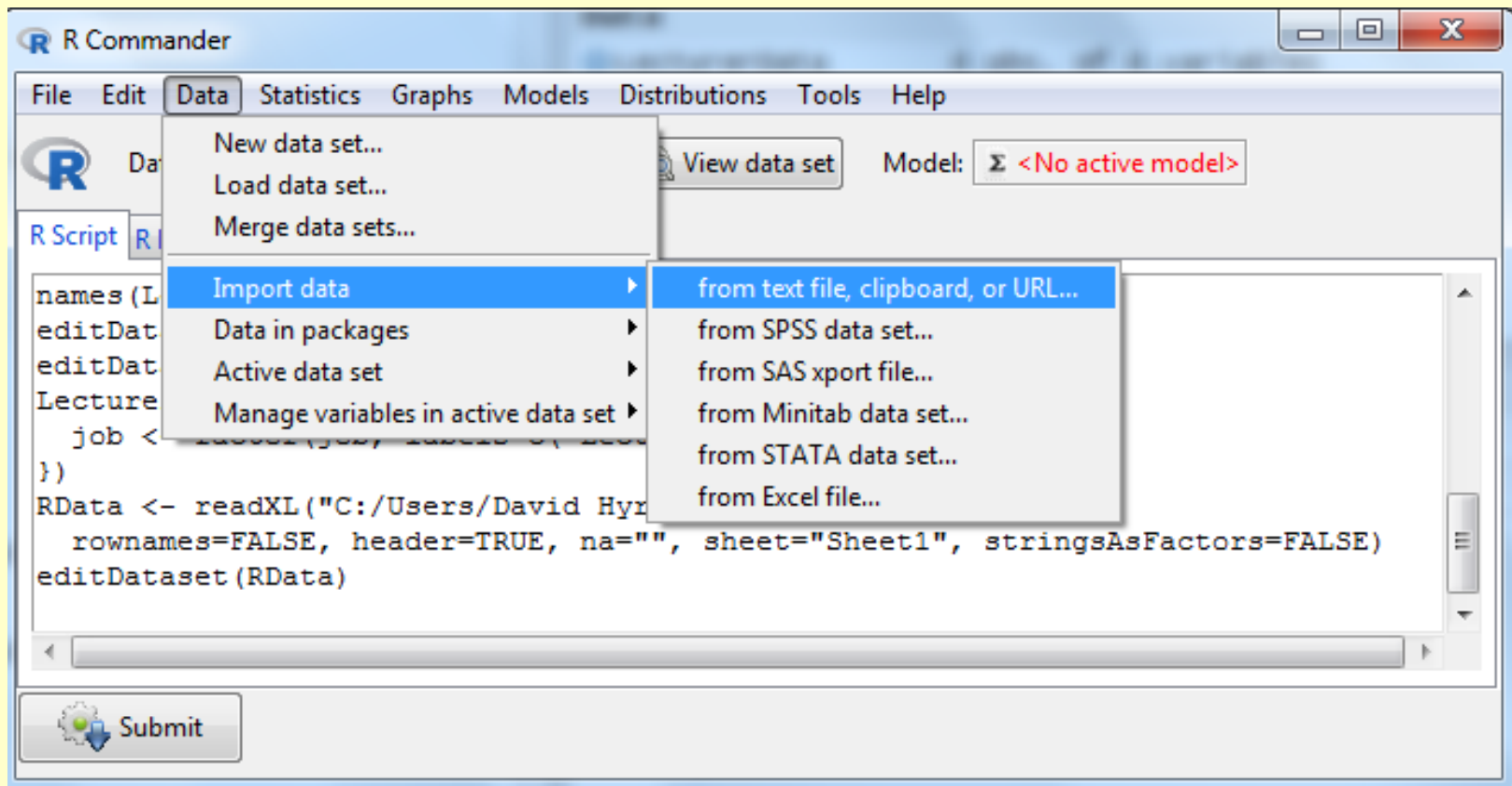
Enter the following information:

dataset name,
whether headers are present,
whether you want to convert character data into factors,
and whether there are any missing data (empty cells)



Importing Data into Rcmdr

Note: Importing ".txt" files from a variety of sources.



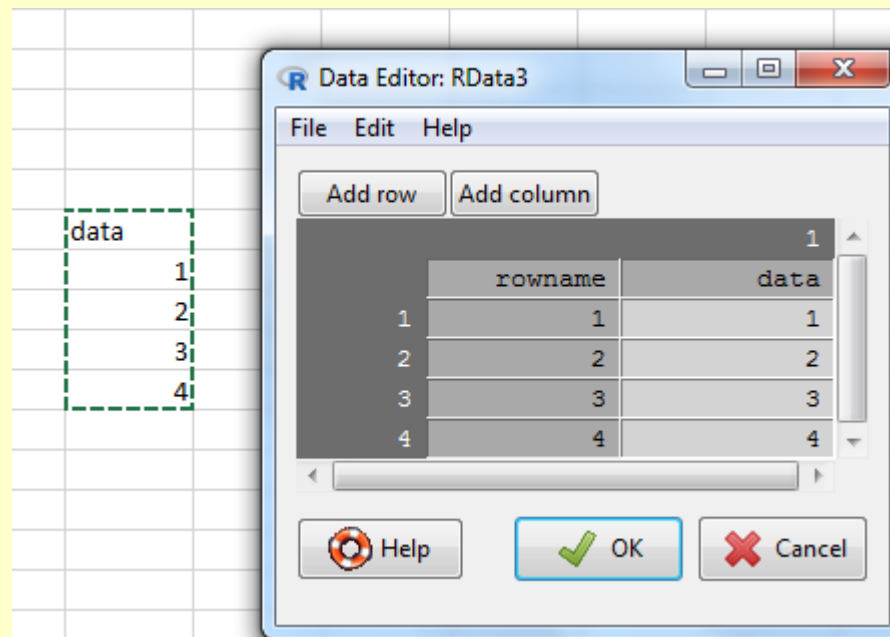
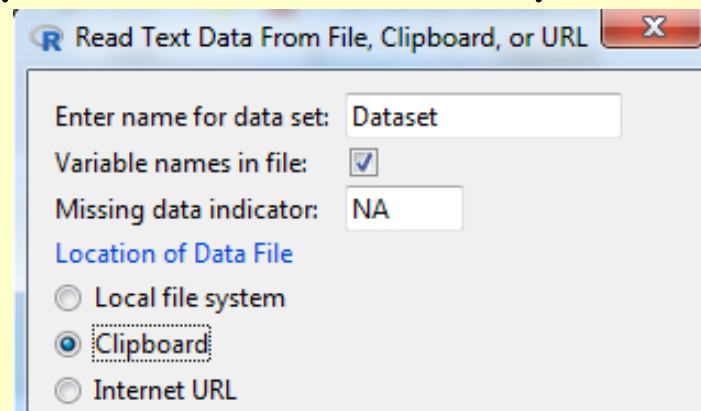
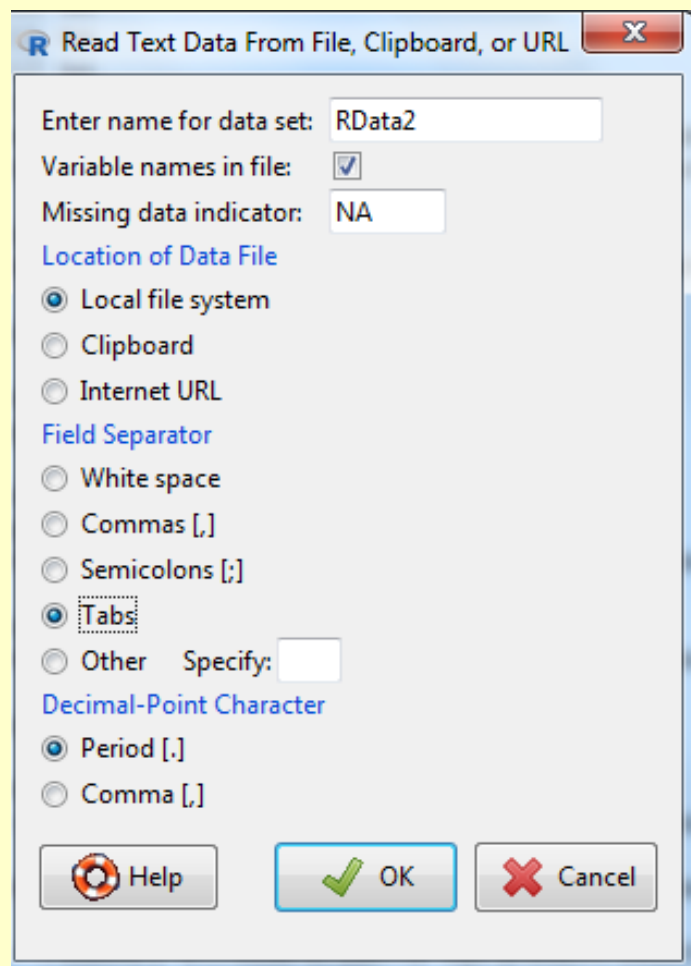
Data / Import Data/ from text file, clipboard, or URL...

Importing Data into Rcmdr

Import file **RData2.txt**

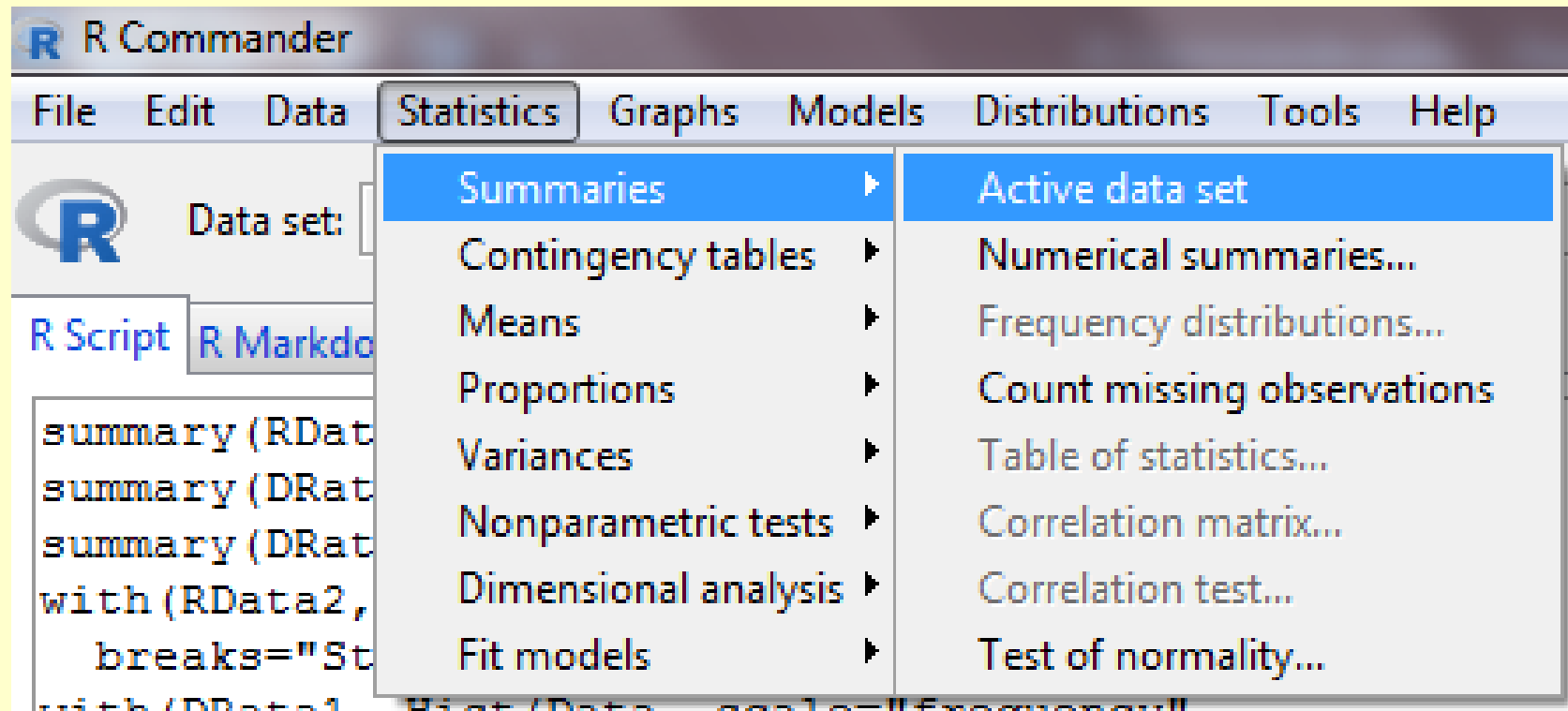
Import data from clipboard

Enter metadata:



Summarizing Data with Rcmdr

- Summarize datasets using:
Statistics / Summaries / Active data set



NOTE: you need to select the active dataset and perform these calculations one at a time

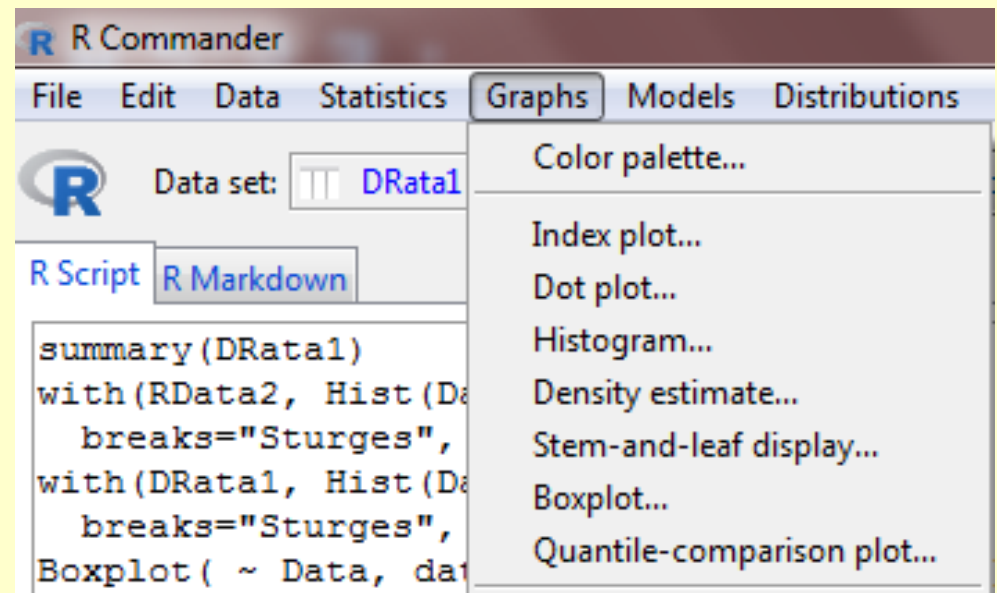
Summarizing Data with Rcmdr

- Create plots using:

Graphs /

Histograms

Box Plots



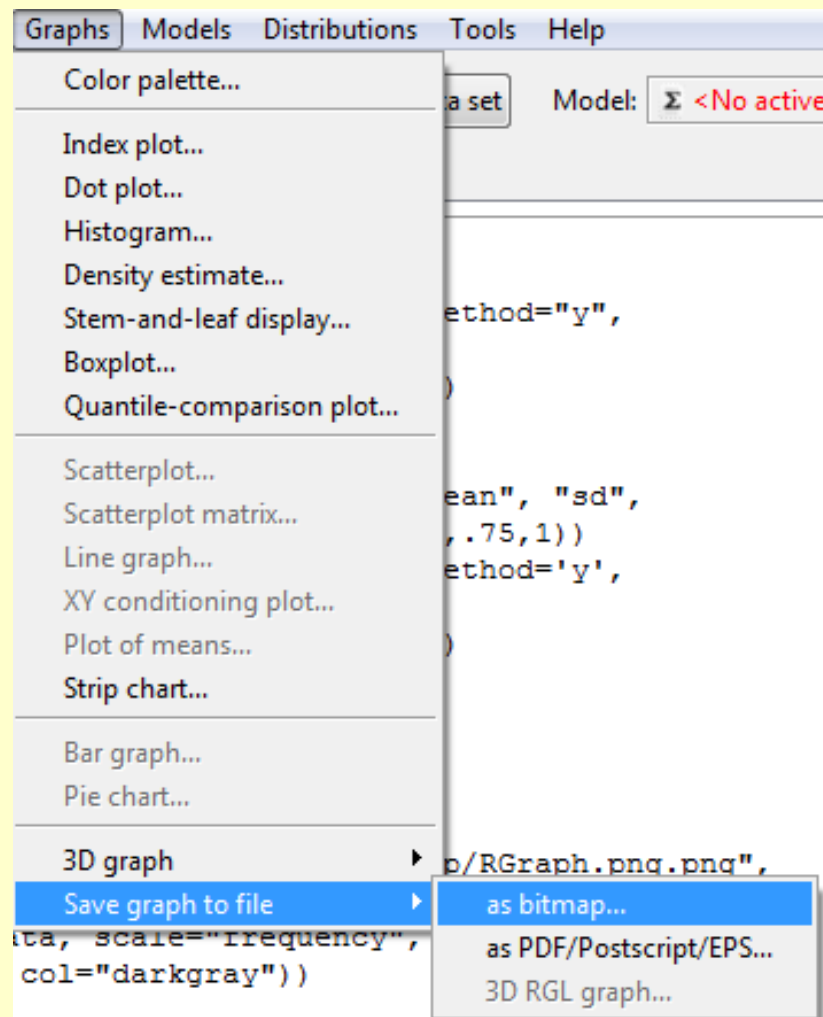
NOTE: you need to select the active dataset and perform these calculations one at a time

HINT: What types of plots can RCmdr make?
(Explore the other available plots)

Saving Plots Made with Rcmdr

- After you create a plot, can save it using:

Graphs /
Save graph to file /
as bitmap
as PDF / PS / EPS

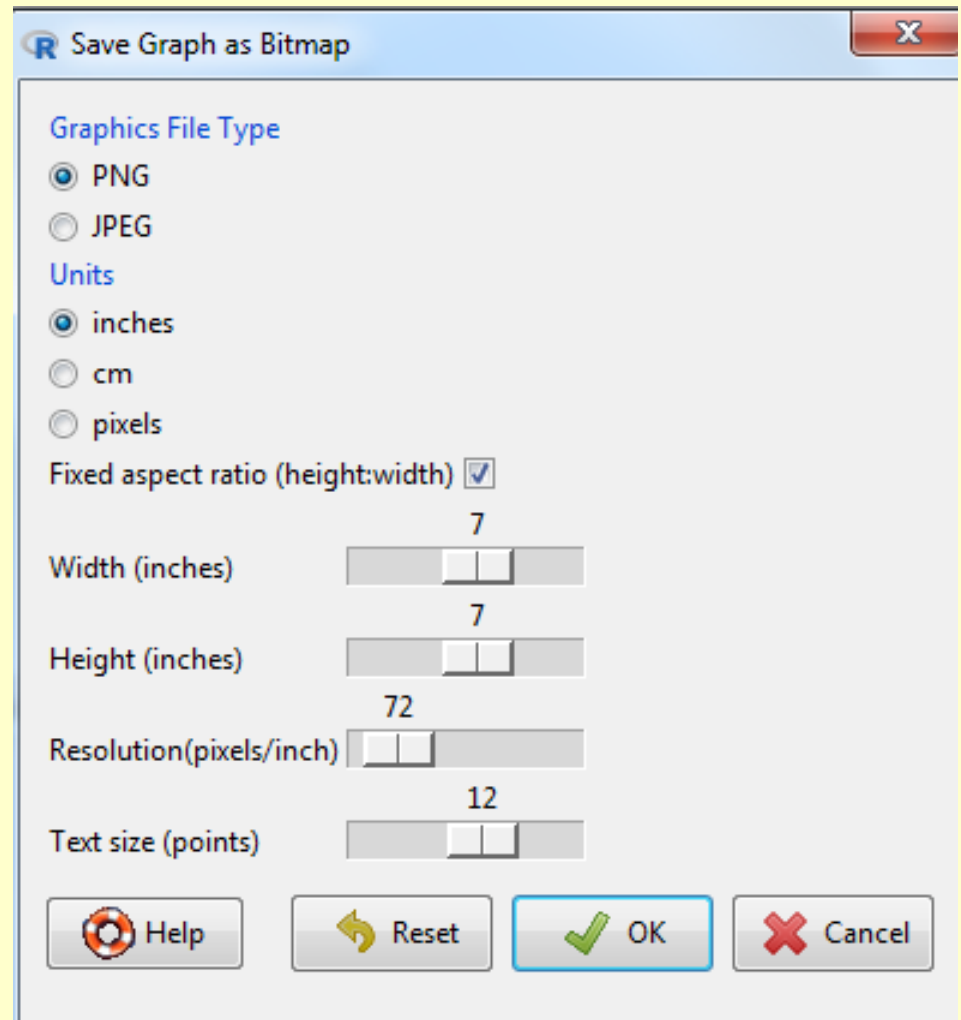


Saving Plots Made with Rcmdr

NOTE: Bitmap is JPEG or PNG format files

You can select:

- Picture size
- Font size



Credits & Acknowledgements

This tutorial is a product of OpenIntro and is released under a Creative Commons Attribution-ShareAlike 3.0 Unported (<http://creativecommons.org/licenses/by-sa/3.0>).

This lab was adapted for MARS 4910 by David Hyrenbach, from a lab for OpenIntro by Andrew Bray and Mine Çetinkaya-Rundel from a lab written by Mark Hansen of UCLA Statistics.



Types of Variables in R

Numeric

Numbers (e.g. 7, 0.5) **NOTE: NA (no data)**

Coding variables / factors

Uses numbers to represent different of data
(e.g. Gender: 1 = male, 2 = female)

Date

Dates (e.g. 21-06-1973, 06-21-73, 21-Jun-1973)

String

Letters (e.g. 'Andy', 'Mary')



Creating a Numeric Variable

Numeric variables are the easiest to create:

```
friends<-c(5,2,0,4,1,10,12,15,12, 17)
```

```
alcohol<-c(10,15,20,5,30,25,20,16,17,18)
```

```
income<-c(20000,40000,35000,22000, 50000,  
5000, 100, 3000, 10000, 10)
```

```
neurotic<-c(10,17,14,13,21,7,13,9,14,13)
```



Creating a Coding Variable

Imagine we had 5 students and 5 lecturers in a sample and we wanted to create a coding variable called **job**.

Enter the data: `job<-c(1,1,1,1,1,2,2,2,2,2)`

Or:

```
job<-c(rep(1, 5),rep(2, 5))
```

Then convert **job** to a factor:

```
job<-factor(job, levels = c(1:2),  
labels = c("Lecturer", "Student"))
```



Creating a Date Variable

Use the *as.Date()* function:

```
DoB<-as.Date(c("1977-07-03", "1969-05-24",  
"1973-06-21", "1970-07-16", "1949-10-10", "1983-  
11-05", "1987-10-08", "1989-09-16", "1973-05-20",  
"1984-11-12"))
```

Each date has been entered as a text string (in quotations) in the appropriate format (yyyy-mm-dd).

By enclosing these data in the *as.Date()* function, these strings are converted to date objects.



Creating a String Variable

We use the `c()` function and list all values in quotations so that R knows that it is string data.

As such, we can create a variable called **name** as follows:

```
name<-c("Ben", "Martin", "Andy", "Paul", "Graham",  
"Carina", "Karina", "Doug", "Mark", "Zoe")
```



Creating a Dataframe

We can bind various variables together into a dataframe:

```
lecturerData<-data.frame(Name, DoB, job, friends,  
alcohol, income, neurotic)
```

lecturerData

	name	<u>birth_date</u>	job	friends	alcohol	income	neurotic
1	Ben	1977-07-03	Lecturer	5	10	20000	10
2	Martin	1969-05-24	Lecturer	2	15	40000	17
3	Andy	1973-06-21	Lecturer	0	20	35000	14
4	Paul	1970-07-16	Lecturer	4	5	22000	13
5	Graham	1949-10-10	Lecturer	1	30	50000	21
6	Carina	1983-11-05	Student	10	25	5000	7
7	Karina	1987-10-08	Student	12	20	100	13
8	Doug	1989-09-16	Student	15	16	3000	9
9	Mark	1973-05-20	Student	12	17	10000	14
10	Zoe	1984-11-12	Student	17	18	10	13