

1 INSTALLING R AND R COMMANDER (Windows)

1.1 Installing R (Current version is 3.4.0 as of 2017-05-05)

NOTE: The instructions below are for Windows installation. To install R for Mac OS X, visit <https://cran.r-project.org/bin/macosx/>.

- Uninstall earlier versions of R and delete all R library folders under Program Files (if applicable)
- Close all other programs
- Go to

- <http://www.r-project.org/>

- Click

- download R

- Among **CRAN mirrors**, for me, the 0-Cloud link below works well

- <https://cloud.r-project.org/>

- Click

- Download R for Windows

- Click

- install R for the first time

- Click

- Download R 3.4.0 for Windows (62 megabytes, 32/64 bit)

and save the file (which will go to your Downloads folder)

- IMPORTANT: Install the program to C: drive, not to the default folder.
- When installing, unselect 32-bit Files (if you have Windows 7 or above) and choose

- Yes (customized startup)

and then choose

- SDI (separate windows)

This above step is important for running Rcmdr smoothly.

Next,

- HTML help
- Start R from the icon on the desktop.
- **You will now need to update packages.**
- Choose
 - Packages > Update packages

and select

- 0-Cloud [<https>] (Or, any other site you prefer).

Follow the instructions to update packages

1.2 Installing R Commander (Current version is 2.3-2 as of 2017-05-05)

NOTE: **The instructions below are for Windows installation.** To install R Commander for Mac OS X, visit <http://socserv.mcmaster.ca/jfoxx/Misc/Rcmdr/installation-notes.html>.

- Exit R (if it is open) and start R¹
- The easiest way to install the Rcmdr package is via the command

- `install.packages("Rcmdr")`

This will unpack about 30 or so packages.

- When you *first* load the Rcmdr package with the command

- `library(Rcmdr)`

it will offer to download and install missing dependencies (with a **terrible noise**); allow it to do so. (It will, by default, install packages from CRAN.)

- Exit Commander and R.

¹Note: If you wish to load the R Commander automatically when R starts up, you can add the following to the Rprofile.site file in R's 'etc' directory: (Use this with care as it may not work on some computers.)

```
local({
  old <- getOption("defaultPackages")
  options(defaultPackages = c(old, "Rcmdr"))
})
```

- Next time you start R, just choose
 - Packages > Load Package > Rcmdr.
 - Or, you can still enter `library(Rcmdr)` to start Rcmdr

This will start the R Commander window and you can start using it now.

- Periodically you should choose
 - Packages > Update Packages.
- Additional help is available here:
 - <http://socserv.mcmaster.ca/jfoxf/Misc/Rcmdr/>

2 INSTALLING OTHER USEFUL PACKAGES (Windows) — Optional

2.1 “Using R” by Verzani

- Install `UsingR` from R by typing
 - `install.packages("UsingR",dependencies=TRUE)`
- or from R,
 - Packages > Install Package(s)...
- Once installed, you can load it from R by
 - `library(UsingR)`,
- or from Rcmdr by
 - Tools > Load Package(s)...

This package is useful for plotting confidence and prediction bands, and providing predictions by, e.g., from the `Table3.1Sales-Advertising.csv` file:

```
simple.lm(Dataset$ADVT, Dataset$SALES, show.residuals=TRUE, show.ci=TRUE, pred=c(9,10,11))2
```

²Usage

```
simple.lm(x, y, show.residuals=FALSE, show.ci=FALSE, conf.level=0.95, pred=)
```

Arguments

x The predictor variable

y The response variable

show.residuals set to TRUE to plot residuals

show.ci set to TRUE to plot confidence intervals

conf.level if show.ci=TRUE will plot these CI's at this level

pred values of the x-variable for prediction, in the form `pred=c(a,b,c)`

2.2 corrplot (Correlation Plot)

- Install `corrplot` package from R first by

- Packages > Install Package(s)...

- After installing `corrplot` for the first time from R, load it from R by

- Packages > Load package...

or by

- `library(corrplot)`

or from Rcmdr by

- Tools > Load Package(s)...

- Then generate the `corrmatrix` using Rcmdr by Statistics > Summaries > Correlation matrix...

- Basically, we do this:

- Rcmdr produces a command `cor(Some R commands)`. Write it as,

- `M <- cor(Some R commands) # Just call it M now`

- `corrplot(M, method = "ellipse")`

- Also possible are the commands,

- `corrplot(M, method = "number")`

- `corrplot(M, order = "FPC",method="ellipse") # This orders them, nice!`