Lab Review: Loading & Examining Data in R

Nathan Brouwer

2017-09-11

Review outline

- Intro
- Fisher's Iris data
- Loading data: iris
- The easy way: from base R w/ data(iris)
- Loading packages in base R: MASS
- command: library(MASS)
- Loading packages from CRAN: cowplot

This lab is based generally on Chapters 2 & 3, Whitlock and Schulter, 2nd Ed

Example data for Lab: Fisher's Irises

- Dataset made popular by R.A. Fisher
- Frequently used to explain/test stats procedures
- See ?iris for more details
- See also https://en.wikipedia.org/wiki/Iris_flower_data_set for more info.

Loading data into R the easy way: pre-made data in an R "Package"

- Getting data into R (or SAS, or ArcGIS…) can be a pain
- R comes with many datasets that are pre-loaded into it
- There are also many stat. techniques that can easily be added to R
- These are contained in "packages"

Load data that is already in the "base" distribution of R

Fisher's iris data comes automatically with R. You can load it into R's memory using the command "data()"

```
#Load the iris data
data(iris)
```

Look at the iris data

We'll look at the iris data using some commands like ls(), dim(), and names().

You can check that it was loaded using the **ls()** command ("list").

```
ls()
```

You can get info about the nature of the daframe using commands like **dim()**

```
dim(iris)
```

This tells us that the iris data is essentially a spreadhshett that has 150 rows and 5 columns.

We can get the column names with names()

```r
names(iris)
```

- Note that the first letter of each word is capitalized.

- What are the implications of this?

The top of the data and the bottom of the data can be checked with head() and tail() commands

```r
#top of dataframe
head(iris)

#bottom of dataframe
tail(iris)
```

Another common R command is **is()**, which tells you what something is in R land.

```r
is(iris)
```

- R might spew a lot of things out at you when you use **is()**
- usually the 1st item is most important.

- Here, it tells us that the "object" called "iris" in your workspace is 1st and foremost a "data.frame"
- A dataframe is essentially a spreadsheet of data loaded into R.

You can get basic info about the data themseleves using commands like **summary()**.

```r
summary(iris)
```

- Which variables are numeric?
- Which variables are categories/groups (aka "factors")?

If you wanted info on just 1 column, you would tell R to isolate that column like this, using a dollar sign ($).

```r
summary(iris$Sepal.Width)
```

That is, that name of the dataframe, a dollar sign ($), and the name of the column.

What happens when you don't capitalize something? Try these intentioanl mistakes (but remove the "#" from in front of each one):

```r
#all lower case
# summary(iris$sepal.width) # this won't work

#just "s" in "sepal" lower case
# summary(iris$sepal.Width)  #this won't work either

#or what if you capitalize "i" in "Iris"?
# summary(Iris$Sepal.Width) #won't work either
```

The 1st two error messages are not very informative; the 3rd one ("Error in summary(Iris$Sepal.Width) : object 'Iris' not found") does make a little sense.

Load data that is in another R package part 1

Packages that come w/ R

- Many scientists develop software for R, and they often include datasets to demonstrate how the software works.
- Some of this software, called a “package” comes with R already and just needs to be loaded.
- This is done with the **library()** command.
- The **MASS** package comes with R when you download it and has many useful functions and interesting datasets.

```
#Load the MASS package
library(MASS)
```

MASS contains a dataset called called “mammals”

```
data(mammals)
```

You can confirm that the mammals data is in your workspace using **ls()**

```
ls()
```

You should now have the “iris”" and the “mammals”" data in your R “workspace.”"

What is in the mammals dataset? Datasets actually usually have useful help files. Access help using the **?** function.

The help screen will pop up. It tells us that mammals is

> “A data frame with average brain and body weights for 62 species of land mammals.”

Since this is someone else’s data, the authors of the MASS package need to provide proper citation. At the bottom we can see that these data come from the paper:

> Allison, T. and Cicchetti, D. V. (1976) Sleep in mammals: ecological and constitutional correlates. Science 194: 732-734.

We can learn about the mammals data usig the usual commands

```
dim(mammals)
names(mammals)
head(mammals)
tail(mammals)
summary(mammals)
```

Load data that is in another R package part 2

Packages from the CRAN website

Most packages don’t come with R when you download it but are stored in a central site called CRAN. We’ll load the **cowplot** package.

Loading packages using R-Studio

RStudio makes it easy to find and load packages. Follow these instructions. A visual walkthrough is also avaialbel on D2L

- In the panel of RStudio that has the tabs “Plots”, “Packages”,“Help”, “Viewer” click on “Packages”"
- On the next line it says “Install” and “Update”. Click on “Install”
- A window will pop up. In the white field in the middle of the window under “Packages” type the name of the package you want.
- RStudio will automatically bring up potential packages as you type.
- Finish typing “cowplot” or click on the name.
- Click on the “Install” button.
- In the source vieewr some misc. test should show up. Most of the time this works. If it doesn’t, talk to the professor!

If an R packages doesn’t load properly, it could be for several reasons.
* 1st, your internet connection might be having problems.
* 2nd, the website where the package is stored might be down for mainteance.
* 3rd, the version of are you are using is probably newer than the version of R used to make the package. This is a real pain - aks for help from an expert R user if think you have this problem.

Loading packages directly using code

You can also use the **install.packages()** command to try to load the package. I already have the **doBy** packaged downloaded to my computer so I have “commented out” the code with a “#”. To run the code, remove the “#”. If you already followed the instructions above you don’t need to run the code.

```
#install.packages("cowplot")
```

Troubleshooting package downloads

What if you tell R to install a package you already have downloaded?

If you already have the package downloaded to your computer then a window will pop up asking you if you want to restart your computer. Normally this isn’t necessary; just click “no”. You might see a “warning” message pop up in the console such as “Warning in install.packages: package ‘doBy’ is in use and will not be installed”. This isn’t a problem for basic R work. If you are doing serious work (e.g. for a publication) you should restart R.

What if I can’t get a package I need loaded?

- Talk to someone who is good w/R (eg, your professor)
- Google something like “how to install R package” for general info
- Google something like “problem loading R package”
- Copy and paste any error message you might be getting into Google and see if anyone has written about this problem

See above for reasons why a package might not load properly the 1st time you try.

Finding R help with Google

There’s lots of info about R on the web, and if you have a problem, then someone else has probably had it before and perhaps written somethign about it.

The website **stackoverflow.com** has lots of info about R. However, many people who use it are hard-core programmers, who can come accross as jerks sometimes when they answer questions if you don’t follow the rules and protocols of stackoverflow.