Introduction to R: **Wrap Up**

Day 4, Part C

In this lecture

1. Writing an R script
2. Running an R script
3. Getting help

Writing an R script: Headers

```
#########################################################
## Author:      This is your name.
##
## Description: This is (briefly) what this code does.
##
## Notes:       This is anything a user should know when
##              running this code.
#########################################################
```

Writing an R script: Loading libraries and clearing the work space

Most R scripts start by loading any required libraries.

```
> library(reshape2)
> library(ggplot2)
```

Writing an R script: Loading libraries and clearing the work space

Most R scripts start by loading any required libraries.

```
> library(reshape2)
> library(ggplot2)
```

After this, you typically want to clear the work space. This ensures that you're running your code in a clean environment each time.

```
> rm(list = ls())
```

Writing an R script: Directories

It's often useful to define the directories for input and output files near the top of the code. This makes it easy to see where input files are coming from and where any output is saved.

```
> data_dir <- "J:/DATA/USA/BRFSS/"
> out_dir <- "J:/Project/us_counties/risk_factors/"
```

Writing an R script: First few lines

Put all together, the beginning of an R script often looks something like:

```
##########################################################
## Author:      This is your name.
##
## Description: This is (briefly) what this code does.
##
## Notes:       This is anything a user should know when
##              running this code.
##########################################################

library(reshape2)
library(ggplot2)
rm(list = ls())

data_dir <- "J:/DATA/USA/BRFSS/"
out_dir <- "J:/Project/us_counties/risk_factors/"
```

Writing an R script: Comments

Don't forget to comment!

Writing an R script: Comments

Don't forget to comment!

Use comments to:

- Label blocks of code. This will help you navigate your code later
- Explain why you're doing something (if it's not self-evident)
- Write yourself (and other users) notes about particularly tricky lines of code

Writing an R script: Comments

Don't forget to comment!

Use comments to:

- Label blocks of code. This will help you navigate your code later
- Explain why you're doing something (if it's not self-evident)
- Write yourself (and other users) notes about particularly tricky lines of code

You want to provide enough information so that your future self, or someone else, can quickly understand the structure and purpose of your code at a later date.

However, it is possible to provide too much information, making your code more cumbersome (e.g., writing out what each line of code does).

Writing an R script: Things to leave out

While you do want to *load* libraries in your scripts, you do not want to *install* them. Libraries only need to be installed once, and doing that each time you run a script is a waste of time.

GOOD:

```
> library(ggplot2)
```

BAD:

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

Writing an R script: Things to leave out

You *usually* also want to leave out lines of code that don't (permanently) modify anything and are primarily used for exploring your data while writing code. For example:

```
> View(data)
> names(data)
> head(data)
> str(data)
> table(data$variable)
```

You absolutely should use these and similar functions as you develop code, but since they have no permanent effect unless you assign the output to an object, they are not needed to reproduce an analysis.

Writing an R script: Styles

When writing R code you have lots of decisions to make that impact how your code *looks* but not how it *functions*.

For example:

- <- vs = for assignment
- single vs double quotes
- spacing between arguments, operators, etc.
- indentation
- white space
- comment style
- object/variable naming conventions

Writing an R script: Styles

You want to aim for readability and consistency.

Writing an R script: Styles

You want to aim for readability and consistency.

There are a number of different style guides that provide recommendations on these topics:

- Hadley Wickham (R demigod) style guide: http://adv-r.had.co.nz/Style.html
- Google R style guide: https://google.github.io/styleguide/Rguide.xml

Running an R script

We have been primarily running chunks of code directly from RStudio (`ctrl + enter`).

This is fine when developing code, but you should always test that the final version works in a new instance of R (`ctrl + shift + F10` in RStudio). This ensures, among other things, that you have properly specified the packages that are required.

Running an R script

"Production runs" should also be done in a new instance of R, and via the source command:

```
> source(file = "C:/Users/ngraetz/Documents/repos/r_training_penn/lectures/lecture_1a_r_basics.r",
+     echo = T)
```

Never, ever do a "production run" by running code block by block. It is incredibly easy to miss or repeat something, or modify something part way through, and that will make your research non-reproducible.

Getting help

Order matters!

Level 1:

1. Help files
2. Additional package documentation
3. Searching Google/Stack Overflow

Level 2:

4. Posting on Stack Overflow
5. Asking colleagues (or other bystanders)

Getting help: Help files

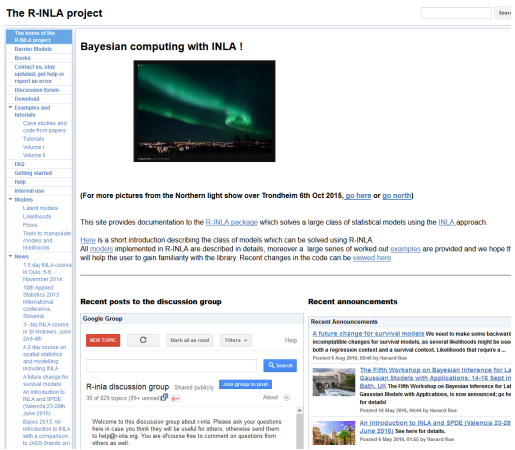
The first place you should look for help is a function's help file:

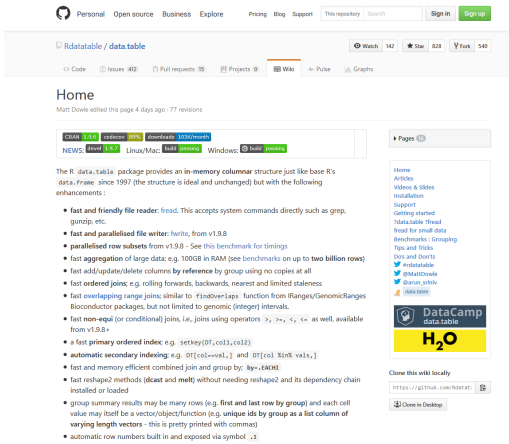
- Are you specifying all required arguments?
- Do defaults for arguments you're not specifying make sense?
- Are you providing values in the right format (e.g., class, structure)?
- How is the output of this function structured?

Pay particular attention to the examples at the bottom; these can always be run, and it's often useful to look at input and output in a case where the function is actually working as intended.

Getting help: Additional package documentation

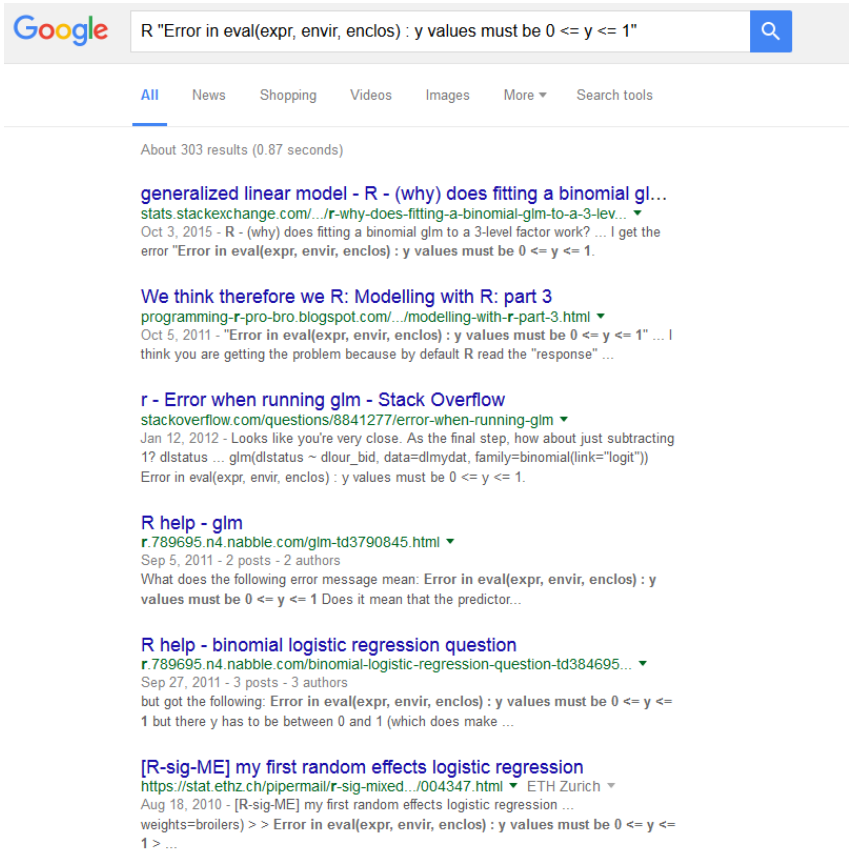
Some R packages have websites with dedicated Wiki's, FAQs, and/or discussion groups (often monitored by the developers). When available, these are a great resource.





Getting help: Google

It is often sufficient to simply cut and paste an error message into Google:



Getting help: Stack Overflow

Stack Overflow is a question and answer site for programming.

You will often find that others have already asked the same (or a sufficiently similar) question that you have. If not, you can post your own question.

Getting help: Stack Overflow

Stack Overflow is a question and answer site for programming.

You will often find that others have already asked the same (or a sufficiently similar) question that you have. If not, you can post your own question.

Some advice:

ALWAYS search before posting (a good rule in general).

If you are going to post, provide either:

- A minimal working example (MWE).
- An example (or clear description) of the output you hope to obtain.

Getting help: Colleagues

Code drop-in hours

Daily, 3:30-4:30

https://hub.ihme.washington.edu/display/ODT/Code+Drop-In+Hours

Getting help: Colleagues

Code drop-in hours

Daily, 3:30-4:30

https://hub.ihme.washington.edu/display/ODT/Code+Drop-In+Hours

R slack channel

`#research-r`

The End!!!