Live Notes from Software Carpentry 2016-04-16-UCSB

Please download
https://github.com/remi-daigle/2016-04-15-UCSB/blob/gh-pages/2016-04-15-UCSB.zip

Shell notes:
http://remi-daigle.github.io/2016-04-15-UCSB/shell/

Computing Wisdom

Never use spaces between names because programs read them as breaks between arguments

Hadley Quotes

- Write for your future self

Shell

[Lesson material by Remi]

`cd` .. brings you back one subdirectory
`cd ~/` home directory

If you are lost in directories, use `pwd` (present working directory)

USE TAB in BASH to quickly get the directory name (saves typing the full name, TAB matches to unique characters)

`rm` = remove
`ls` = list files

If you want to use textedit.app on a mac in BASH, you must do this call:

```
open -a TextEdit file.txt
```

```
Or if you want to use the default text editor
open -t file.txt
```

Rm *.txt deletes all the .txt files within this folder.
But, first use 'echo' which will tell you what that command is going to do

Wc = word count. Lists the word count for a file in columns. Columns represent line count, word count, and character count, in order

Wc *pdb = shows word count for all pdb files, with the same column order as above

But what if we only want one of those data colums?
wc -l *pdb (wc space dash 'letter l" space star pdb
This is saying count the lines in each of the pdb files within this folder. (l signifies line i guess?). The result shows in the terminal

```
169-231-117-248:molecules sara-katherinecoxon$ wc -l *pdb
      20 cubane.pdb
      12 ethane.pdb
       9 methane.pdb
      30 octane.pdb
```

```
 21 pentane.pdb
 15 propane.pdb
107 total
```

If we want to write that info to a file, ie a .txt file, type wc -l *pdb >linenumber.txt

So same command as above, but the '>' tells it that it wants the output of the command to go somewhere besides the terminal window. In this case, to create and put it into a file named linenumber.txt

NEVER use rm / or you delete your whole HD!!!

If you want a new computer to to simplify your life, type rm -rf /

Echo is better than ls because you can list certain files *.extension

Wc is wordcount - works on any plain text file including .csv files

6
for filename in *.dat; do head -n 2 $filename | tail -n 1; done > class.txt

Then to make your own program, paste in text editor the above and save that file as .sh

Then go back and run out via bash .sh

Now to get tricky, go back to .sh, delete the write out to class.text and replace with $1 (this will tell it not to write out but just do it)

Then back out to BASH prompt and run it

Bash .sh then your file name .txt

POWER UP!

A flag is a -letter and tell it what to do within that function sort -n is number sort -a alphabetical so they are arguments.

This is a different hat then r.

Piping is like chaining operations together, both in command line and in R **|**

```
command --help
```

For loops use **;**
$ assigns variable
`test=1` (assigns a value of 1 to variable test)
`echo $test` (returns the variable value)
.sh is a shell script
`bash` to run shell script and append new file with type to write output
`bash script.sh something.txt`

R and RStudio

Lesson materials by Julie

RStudio IDE Cheatsheet (it's also in your cheatsheets folder you downloaded!)

variable name assignment (snake_case, dot.notation, camelCase)

Questions

If you use plot(cars) without a call to specific columns, does it work if there are more than two columns?
For a data frame with more that 2 columns, it's a good idea to specify which. For example:
plot(x=lifeExp,y=gdpPercap,data=gapminder)
Or:
plot(x= gapminder$lifeExp,y=gapminder$gdpPercap)

What do you have to do line before abline?
No, you can insert the coefficients manually, or acquire the coefficients with other functions such as lm()

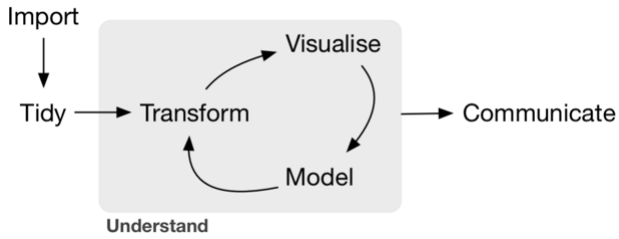
When using the Github repository to create a webpage, can media files such as .jpeg be stored natively on the Github server, or do you always have to point to another server where the content is stored offsite?

Git & Github

Ben's lesson material: http://remi-daigle.github.io/2016-04-15-UCSB/git/
Reproducible science =

- script your workflow: import, tidy, transform, visualize, model, communicate *using R/RStudio*

- version your code, text and data *using git.*
- store and share your code *using GitHub*
- communicate creating reports or webpages *using Rmarkdown*.
    - What this means is embedding your R code within plain text to create your thesis!! With all the figures embedded using your R code! Never copy/paste figures into a Word doc again!



- Always a good practice to create your repository with an associated readme file.
- ownername.github.io/repositoryname to view the website associated with the repository

You can interact with git and github through the command line for even more control! Yay the command line is useful!!

Rmarkdown

- ****text** for bold** ////// *_text_ for italic* //// **#text for main header** ////
- **## ### #### ###### ###### for subsection headers**

Data wrangling

Visualization

Google charts
https://developers.google.com/chart/

http://rmarkdown.rstudio.com/

ggplot2 catalogue:

http://shiny.stat.ubc.ca/r-graph-catalog/

Eco-data-science

Visit our web page: http://eco-data-science.github.io/
Sign up for eco-data-science e-mails and such

Questions

- Why are we not using github for questions?
    - Because for now this is easier? :) Yup.
    - Yes, this is much faster. And we don't want to have to introduce Github Issues just yet.
- How is rmarkdown different from html?
    - Markdown can be rendered *as* html; so you can write in plain text and it will be 'knitted' together with all the html tags you'd need. Basically you can make webpages without knowing html
        - But if you know HTML tags, you can use them within Markdown and they'll be just fine. E.g. if you want to make text bold, you can use Markdown (**bold text** or __bold text__) or HTML (bold text...)
    - Markdown is much easier to read/write, Rmarkdown is the R=flavoured version where you can insert chunks of working R code in it Awesome!!
- Why did we delete the master branch?
    - Because we wanted to avoid confusion, creating a different branch essentially creates 2 version of you project. We deleted the master to avoid 'accidentally' working in that branch. We need to use a branch called 'gh-pages' to be able to have a website on github for a project.
- Can we put images and other media files in Github repository?
    - YES. There are file size limits though. See Github Help:
        - What is my disk quota?
        - Working with large files
- Talking about images in markdown, what exclamation point are we talking about?
    - The syntax for images is Note the ! before the []; this is how markdown knows it's an image and not just a link.
    - See Mastering Markdown · GitHub Guides for more.
- What if we actually want to display a # or some symbol, on the page?

    - \#
    - The \ in this case is the 'escape character', it's used to escape the normal function of the special formatting characters like _ or #
- How do you change the font?
    - There are no 'built in' ways to change the font using pure markdown, but you can use html/css to change the font
        - See HTML CSS - W3schools.com
- Why did the branch we created have to be named 'gh-pages?' What types of name formats work?
    - You 'can' name your branches anything you want, but 'gh-pages' is a special name that will tell github to generate a web page at username.github.io/reponame from the content in the repo. I don't know of any other such special names
    - The gh-pages stands for Github Pages. See GithubPages.com, and User, Organization, and Project Pages - Github Help for details

Type of GitHub Pages site	Pages default domain & host location on GitHub	Publishing branch
User Pages site	`username.github.io`	master
Organization Pages site	`orgname.github.io`	master
Project Pages site owned by a **user account**	`username.github.io/projectname`	gh-pages
Project Pages site owned by an **organization**	`orgname.github.io/projectname`	gh-pages

- How do I make my own .gitignore file for my repository in RStudio?
    - Short answer: create a text file, name it .gitignore put the file names you want to ignore
    - Long answer: using the github desktop app, right click on files you would like to ignore, and click on ignore
    - Also see Hadley's advice on how best to setup Git and GitHub with RStudio
- Is the github desktop app basically an ftp client?
    - No. Think of it like Dropbox, which allows you to have things on your local computer that sync online. But, unlike Dropbox, there is way more control of how those things are updated to 'the cloud' online. It tracks WHO did it (github username) and WHAT they did (line by line differencing). Plus, you have to put a human-readable 'commit message' so that you and everyone else can read a short description of what you did.
    - No idea?
- Rmarkdown outputs advanced question: can you have 2 outputs? Short answer: yes
    - Sounds cool. Having the pdf would look different though?

    - Or are you assuming it is the render of the hmtl view byt file printed as .pdf?
- If we develop webpages using MD in GitHub or Rstudio, is it portable to making a website in a non-GitHub URL or do we have to convert it to HTML before doing so?
    - Short answer: yes!
    - Long answer: the html is always totally portable, md is not necessarily portable, it needs 'something' to convert the md to html, GitHub uses Jekyll, which can also be used on other websites/servers. We also use Rstudio to convert from Rmd (not portable) to html (portable) or md (portable if you have access to jekyll on the server.
- Chris: can you pass any object from left to the right of the then %>% operator, besides dataframes?
    - YES, it will pass any type of object. The important thing to remember is that the first argument of the receiving function on the right has to be appropriately handled as whatever the object is. Since dplyr functions are oriented around data frames, they always return a dataframe object as an input and an output.
- What's the difference between the "Mutate" function and the "Transform" function?
    - Not much! They will produce the same numbers, but mutate will also make your resulting dataframe as a tbl_df as opposed to transform which will return a normal dataframe (see code below)

```
library(dplyr)
library(gapminder)

gap_mutate <- gapminder %>%
   mutate(gdp=pop*gdpPercap)

gap_transform <- gapminder %>%
   transform(gdp=pop*gdpPercap)

# all the values in them are equal
all.equal(gap_mutate,gap_transform)
## [1] TRUE

# but gap_mutate and gap_transform are not totally identical
identical(gap_mutate,gap_transform)
## [1] FALSE

# because their structure is different
str(gap_mutate)
## Classes 'tbl_df', 'tbl' and 'data.frame':    1704 obs. of  7 variables:
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ pop      : int  8425333 9240934 10267083 11537966 13079460 14880372 12881816
```

```
13867957 16317921 22227415 ...
##  $ gdpPercap: num  779 821 853 836 740 ...
##  $ gdp      : num  6.57e+09 7.59e+09 8.76e+09 9.65e+09 9.68e+09 …

str(gap_transform)
## 'data.frame':    1704 obs. of  7 variables:
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ pop      : int  8425333 9240934 10267083 11537966 13079460 14880372 12881816
13867957 16317921 22227415 ...
##  $ gdpPercap: num  779 821 853 836 740 ...
##  $ gdp      : num  6.57e+09 7.59e+09 8.76e+09 9.65e+09 9.68e+09 ...
```

- What's the URL for Ben's informatics site from Bren with all of the HTML widget showcases?
    - http://ucsb-bren.github.io/env-info/wk06_widgets.html
-