**Software Carpentry Saturday October 3**

**R**

<https://etherpad-mozilla.org/swc-iplant-2015-10-141>

<https://www.dropbox.com/s/zr0vit95f7ed0mc/shell-history.txt>

<http://naupaka.github.io/2015-10-03-ua-iplant/>

<http://software-carpentry.org/>

<https://atmo.iplantcollaborative.org/application#instances/f2aac81c-0b7d-460f-864b-2433e78e6047>

R script: *Software\_carpentry\_October\_Rscript.R*

Note1: R doesn’t care about indentation and spaces!

Note2: to comment, put an ‘#’ at the beginning of the line.

Note3: no multiline comment on R, BUT you can do it all at once by selecting several lines and then **ctrl-shift-C**.

Give a name to a data file on R:

>>> data1 **<- read.csv(“**file\_name.csv**”)**

There are 2 ways to set up directory:

* manually: from Files in the down right window
* *Session* > *Set Working Directory* > *Choose Directory*

Note: *Run* runs everything, whereas cmd-Enter runs just the line I’m on. *Run* is something you do at the end when you wrote your whole program.

Get mean, median, min, max:

>>> **summary(**data1**)**

Print string:

**“**hello world**”**

[1] “hello world”

Get directory, equivalent of *pwd* in R:

>>> **getwd()**

Change directory:

>>> **setwd()**

note: avoid doing that though because it’s nicer to have all files in the same directory. Otherwise you have to push different directories in github…

Plot years (x) vs population (y):

>>> **plot(**data1**$x,**data1**$y)**

or human readable:

>>> **plot(x=**data1**$x, y=**data1**$y)**

Functions doc:

>>> **help(**plot**)**

Titles: if you put the axes label in *title* command, it will put both titles on top of each other. If you want the *xlab* one, you need to write all in the same line:

>>> plot(x=data1$year,y=data1$lifeExp,

main="Afghanistan life Expectancy",

xlab="year",

ylab="life expectancy")

Note: running line by line, you’ll see a ‘+’ , a way for R to tell you it’s not done.

same as:

>>> plot(x=data1$year,y=data1$lifeExp,main="Afghanistan life Expectancy", xlab="year", ylab="life expectancy")

Note: this second form is less nice because the line is too long, so annoying to read.

Regression line:

>>> **abline(lm(**y**~**x**))**

Note1: ~is alt-N

Note2: don’t forget to define x and y.

>>> x **<-** data1$year

Note: you can also set names for each variable (axes label, title, data) to use a short cut for the rest of the program (see Rscript).

Save a plot from the script:

>>> **pdf(file=** “myplot.pdf”**)**

>>> **plot** …

>>> **dev.off()**

Note1: make sure you’re in the right directory with *getwd()*

Note2: it won’t appear on my mac automatically in the folder. Refresh it!