Coding Tips and Exploring Data

Amanda and Pilar

09/02/2020

Coding: The Script

Recall the difference between the Console and the Script.

The console is for "quick and dirty" coding, for trying out commands, etc. But most of the time you'll want to save your work. This is what the script is for.

In it, we write our code, and then we send it to the console for execution.

To execute our code, we put our cursor on the line and hit ctrl + Enter

Remember: whatever you write on the console IS NOT SAVED. What you write on the script is, but there may be some things you are not interested in saving!

Coding: Keeping it neat

Most important: COMMENT YOUR CODE AND KEEP IT NEAT

Comments are written with hashtags #. R will not run anything that comes after one.

Comment everything that you do. Your future self, your coauthors, and replicators will appreciate it.

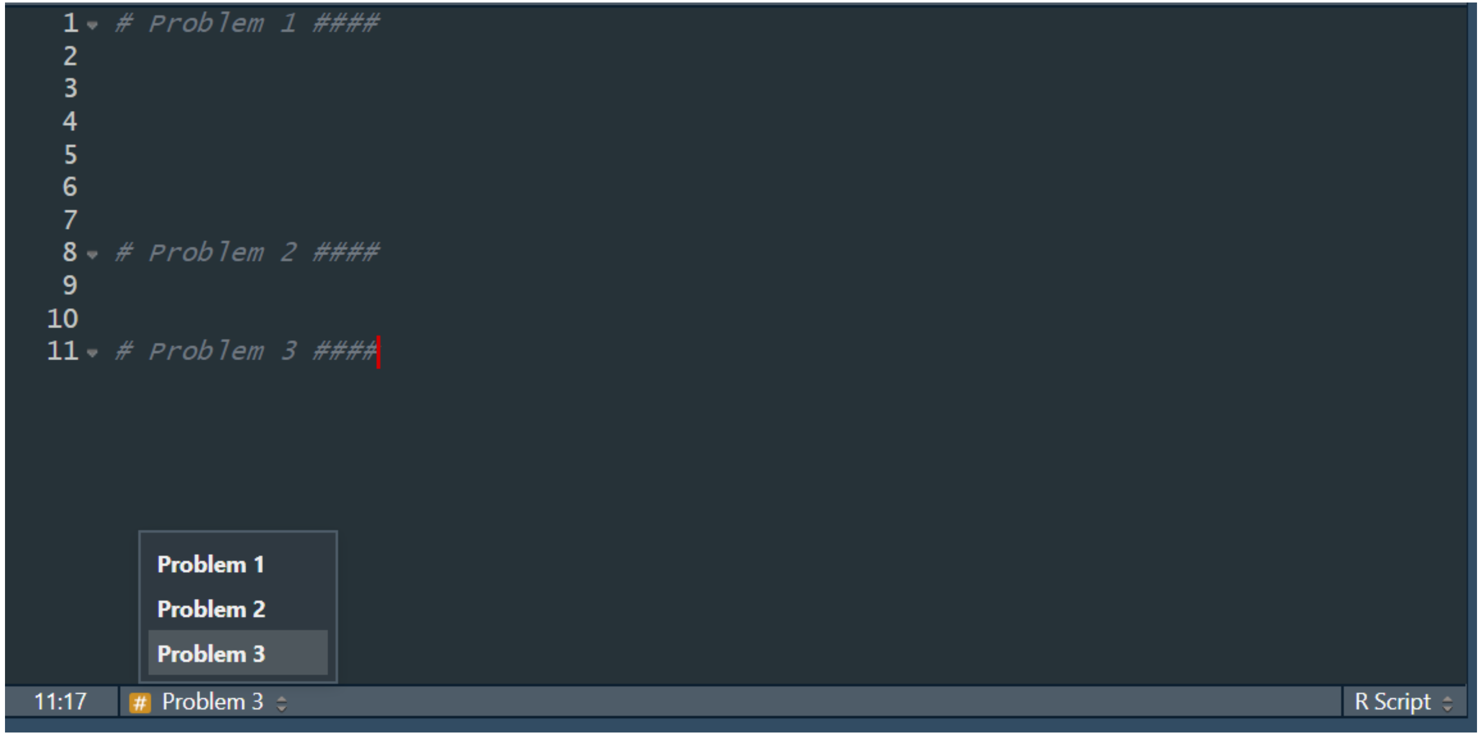
```r
# Create a vector with birthdates
bday <- c(1993,1991,1996,1989)

# Calculate mean
mean(bday)
```

```
## [1] 1992.25
```

Coding: Keeping it neat

You can also create sections of code by using four hashtags (or dashes) after your section title. This will make it easier to find your spot on the code (they will get long!).



```
# Problem 1 ####






# Problem 2 ####


# Problem 3 ####
```

Coding: Keeping it neat

ALWAYS use <− as your assignment mechanism, NOT =

ALWAYS use spaces between operators, like =, <−, ~

- USE: mod1 <- lm(formula = weight ~ height + age + gender)
- NOT: mod1<-lm(formula=weight~height+age+gender)

Tip: keyboard shortcut for <- includes these spaces. Option + - on a Mac, or Alt + - on Linux and Windows.

Coding: Loading libraries

Always a good idea to load your libraries at the top of your code.

```r
library(tidyverse)
library(car)
library(janitor)
```

If you load a library without installing it first, you will get this error:

```r
library(BioConductor)
```

```
## Error in library(BioConductor): there is no package called 'BioConductor'
```

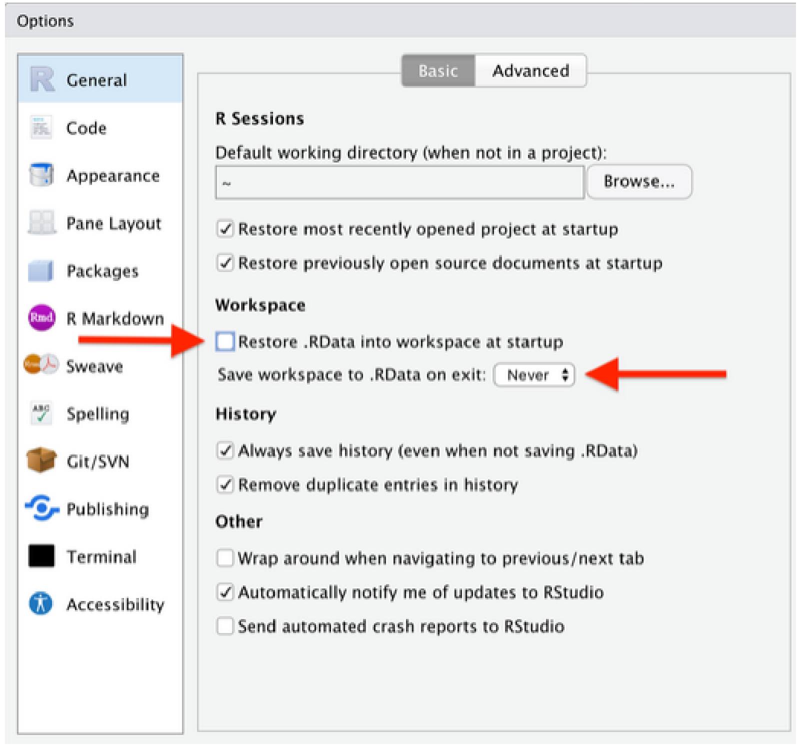
If you try to run a command from a package without loading it first, you will get this error:

```r
glimpse(gapmider)
```

```
## Error in glimpse(gapmider): object 'gapmider' not found
```

Coding: What to save

Always save your script. NEVER save your environment upon exit. It's usually a good idea to start each session from scratch.



R Sessions

Default working directory (when not in a project): ~

☑ Restore most recently opened project at startup

☑ Restore previously open source documents at startup

Workspace

□ Restore .RData into workspace at startup

Save workspace to .RData on exit: Never

History

☑ Always save history (even when not saving .RData)

☑ Remove duplicate entries in history

Other

□ Wrap around when navigating to previous/next tab

☑ Automatically notify me of updates to RStudio

□ Send automated crash reports to RStudio

Each time you open R you're environment will be empty. This means that the objects you created in your previous session will not be saved (but you can always re-create them from the script).

Coding: Other good practices

NEVER write code that, when shared, changes someone else’s machine

- USE: #install tidyverse
- NOT: install.packages("tidyverse")

Coding: Working Directory

Your working directory is the folder in your computer where the script is saved. We usually want to bring in other files into R (such as a .csv file that contains our data) or we may want to save things like plots.

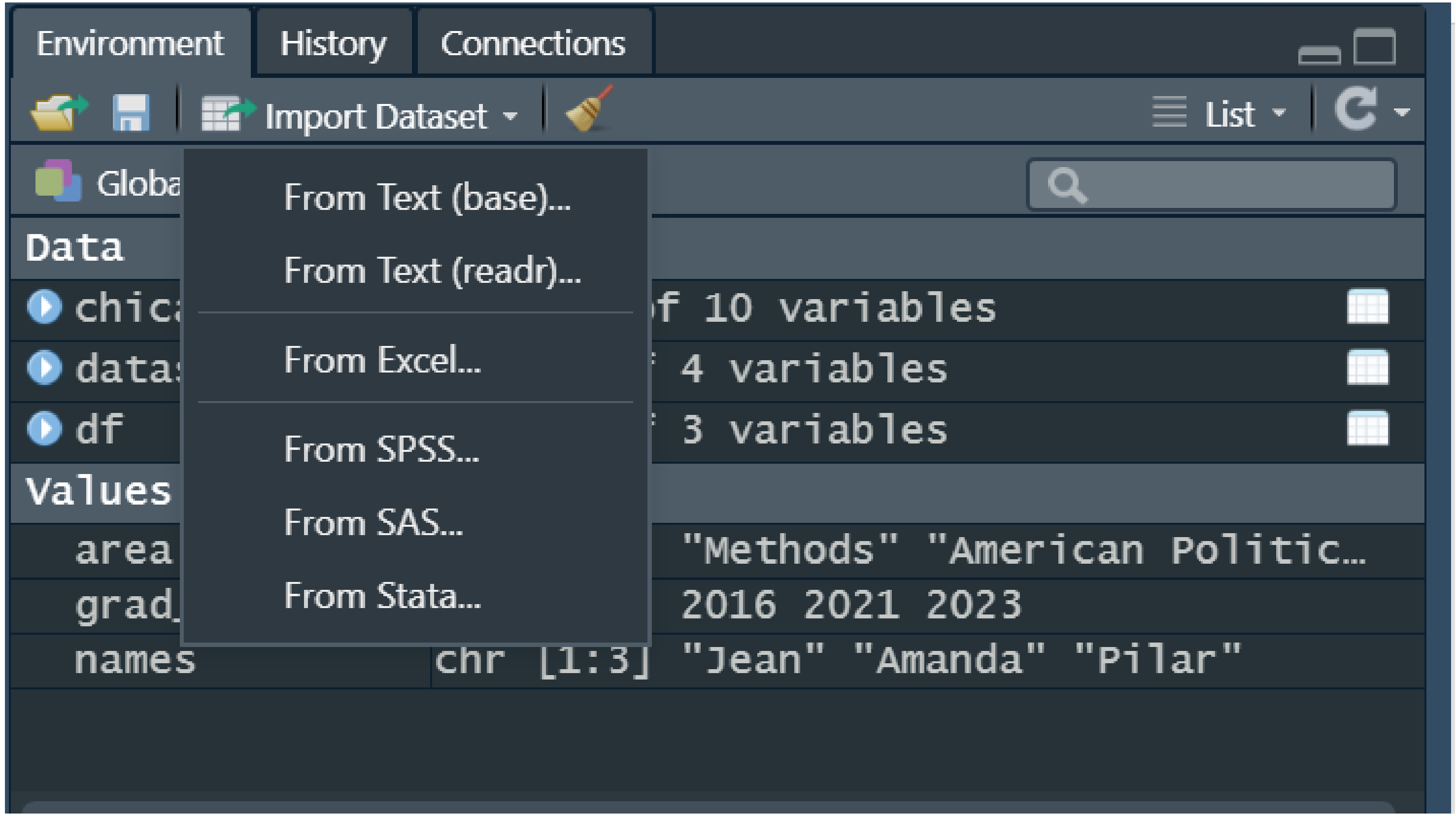
We want all of this to be in the same folder, so we'll tell R where this folder is. You will then be able to see these files under the "Files" pane, or with dir().

We do this by using the command setwd(""). You will copy-paste your folder's path in these quotations.

```
setwd("C:/Users/pilim/Dropbox/Northwestern/TA/Math Camp/Session 2")
```

Windows users: you will have to change the backslash \ for the forward slash /

Reading in Data: The easy way



```
Data
chica              f 10 variables
data:              4 variables
df                 3 variables
Values
area               "Methods" "American Politic…
grad_              2016 2021 2023
names        chr [1:3] "Jean" "Amanda" "Pilar"
```

Reading in Data: Most efficient way

Once you are comfortable with setwd() and how to type paths, this is the most efficient option. You can run a script that has a line to import the datasets.

There are several commands to read files from outside R. The command you use depends on the type of data file. Below, one of the most common ones:

```
chicago <- read.csv("data/Census Data_selected_2008-2012.csv")
```

If I had not set my working directory, I would write the whole path to the file:

```
chicago <- read.csv("C:/Users/pilim/Dropbox/Northwestern/TA/Math Camp/Session 2/data/Census Data_
```

PRACTICE

1) Download the Chicago dataset from Canvas (both the .csv and the .dta)

2) Upload the csv version to R.

Reading in Data

Besides the Base R commands for importing data, many packages offer alternatives. See "readr" and "haven", both part of the Tidyverse collection.

"haven" is useful for importing Stata files, which you will often encounter

PRACTICE: Install and load the package "haven". Use it to read in the .dta version of the Chicago dataset.

```r
library(haven)

chicago_stat <- read_dta("data/Census Data_selected_2008-2012.dta")
```

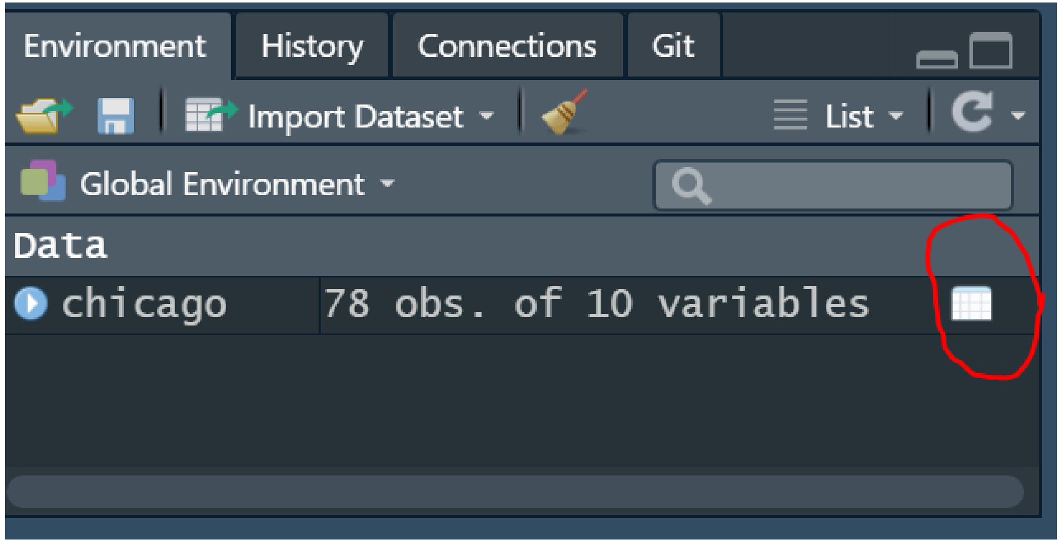
Exploring data

There are several initial steps you can do to get a sense of what the dataset looks like, what variables it contains, how it's structured, etc.

1) The first easy way is to use the command View(), which opens up the dataset in a new window.

```
View(chicago)
```

Alternatively, you can click the on the dataset icon next to the name of the dataset in the Environment window.



Data

chicago 78 obs. of 10 variables

Exploring data

2) Look at the dimensions of the dataset:

```
dim(chicago)
```

```
## [1] 78 10
```

3) Look at the names of the variables:

```
names(chicago)
```

```
##  [1] "Community.Area.Number"
##  [2] "COMMUNITY.AREA.NAME"
##  [3] "PERCENT.OF.HOUSING.CROWDED"
##  [4] "PERCENT.HOUSEHOLDS.BELOW.POVERTY"
##  [5] "PERCENT.AGED.16..UNEMPLOYED"
##  [6] "PERCENT.AGED.25..WITHOUT.HIGH.SCHOOL.DIPLOMA"
##  [7] "PERCENT.AGED.UNDER.18.OR.OVER.64"
##  [8] "PER.CAPITA.INCOME"
##  [9] "HARDSHIP.INDEX"
## [10] "ZONE"
```

PRACTICE: Now try the same with chicago_stat. What's different?

Exploring the Data

4) Look at the firs rows of the dataset. (Figure out how to look at the last rows too).

```
head(chicago)
```

```
##   Community.Area.Number COMMUNITY.AREA.NAME PERCENT.OF.HOUSING.CROWDED
## 1                     1         Rogers Park                        7.7
## 2                     2          West Ridge                        7.8
## 3                     3              Uptown                        3.8
## 4                     4      Lincoln Square                        3.4
## 5                     5        North Center                        0.3
## 6                     6           Lake View                        1.1
##   PERCENT.HOUSEHOLDS.BELOW.POVERTY PERCENT.AGED.16..UNEMPLOYED
## 1                             23.6                         8.7
## 2                             17.2                         8.8
## 3                             24.0                         8.9
## 4                             10.9                         8.2
## 5                              7.5                         5.2
## 6                             11.4                         4.7
##   PERCENT.AGED.25..WITHOUT.HIGH.SCHOOL.DIPLOMA PERCENT.AGED.UNDER.18.OR.OVER.64
## 1                                         18.2                             27.5
## 2                                         20.8                             38.5
## 3                                         11.8                             22.2
```

Exploring Data

4) Summarize the variables

```
summary(chicago)
```

```
##  Community.Area.Number COMMUNITY.AREA.NAME PERCENT.OF.HOUSING.CROWDED
##  Min.   : 1            Length:78           Min.   : 0.300
##  1st Qu.:20            Class :character    1st Qu.: 2.325
##  Median :39            Mode  :character    Median : 3.850
##  Mean   :39                                Mean   : 4.921
##  3rd Qu.:58                                3rd Qu.: 6.800
##  Max.   :77                                Max.   :15.800
##  NA's   :1
##  PERCENT.HOUSEHOLDS.BELOW.POVERTY PERCENT.AGED.16..UNEMPLOYED
##  Min.   : 3.30                    Min.   : 4.70
##  1st Qu.:13.35                    1st Qu.: 9.20
##  Median :19.05                    Median :13.85
##  Mean   :21.74                    Mean   :15.34
##  3rd Qu.:29.15                    3rd Qu.:20.00
##  Max.   :56.50                    Max.   :35.90
##
##  PERCENT.AGED.25..WITHOUT.HIGH.SCHOOL.DIPLOMA PERCENT.AGED.UNDER.18.OR.OVER.64
##  Min.   : 2.50                                Min.   :13.50
```

PRACTICE

1) Find the variable that has the neighborhood names and print it out (i.e. show the neighborhood names).

2) Get the descriptive statistics for percent of households below poverty.

3) What can you say about Rogers Park's level of poverty in comparison to the rest of the city?

4) Which neighborhood has the highest poverty rate? What zone is it located in?

5) Create a new variable that indicates whether the poverty rate is above or below the average poverty rate of the city.

6) Use the command 'plot' to make a scatter plot between poverty rates and percent without HS diploma.
(We'll learn how to make prettier plots in the last session)

More skills ! 😃

Find, download and upload the data for the latest wave of the World Value Survey.

```r
wvs <- read_dta("data/WVS_Cross-National_Wave_7_stata_v1_4.dta")
```

Also find and download the codebook for the dataset.

Tables

One-way tables are useful to see the values the variable takes on and how it is distributed.

Let's explore attitudes towards women by looking at some of the questions in the WVS (Q29-Q33). For example:

```
table(wvs$Q30)
```

```
##
##     1     2     3     4
##  6232 10785 31423 20078
```

Tables

Two way tables allow us to see how two variables are related to each other.

Do men and women both think that men make better political leaders?

```
table(wvs$Q29, wvs$Q260)
```

```
## 
##           1     2
##   1  5941  4602
##   2  9363  8663
##   3 12611 14874
##   4  4333  7353
```

Tables

Although these tables are informative, it's better to see these numbers as proportions.

Among men, what percentage "Agree strongly" that men make better poltiical leaders? What is this percentage among women?

```
prop.table(table(wvs$Q29, wvs$Q260),2)
```

```
##
##              1         2
##   1 0.1842285 0.1296630
##   2 0.2903436 0.2440832
##   3 0.3910630 0.4190804
##   4 0.1343649 0.2071734
```

What is that 2 doing there? What happens if we change it for a 1?

PRACTICE

1) Create a two-way table between sex and question Q30 (University is more important for a boy than for a girl)

2) How do opinions about women differ between religious and non-religious people? Look for question Q6 in the codebook. Pick a question (Q29-Q33) and cross it with Q6.