5: Data Exploration

Environmental Data Analytics | Sarah Ko - Notes

Spring 2019

LESSON OBJECTIVES

1. Set up a data analysis session in RStudio
2. Import and explore datasets in R
3. Apply data exploration skills to a real-world example dataset

BEST PRACTICES FOR R

In many situations in data analytics, you may be expected to work from multiple computers or share projects among multiple users. A few general best practices will avoid common pitfalls related to collaborative work.

Set your working directory

A session in RStudio will always function by mapping to a specific folder in your computer, called the *working directory*. All navigation between folders and files will happen relative to this working directory. When you open an R project, your working directory will automatically set to the folder that holds the project file. If you open an R script or RMarkdown document directly by double-clicking the file, your working directory will automatically set to the folder that holds that file. It is a good idea to note with a comment at the top of your file which working directory you intend the user to designate.

In this course, we will always open the R project file for the course, and additional navigation of the working directory will happen from that folder. To check your working directory, use the following R command:

```
# Working directory should be set to the parent folder for the Environmental Data Analytics Course, i.e

getwd()
```

```
## [1] "C:/Users/Sarah/Documents/Duke/Year 2/Spring 2019/Data Analytics/Environmental_Data_Analytics"
```

What is the output that results?

If your working directory is not set to the folder you want, you have several options. The first is to directly code your working directory. You may do this by defining an absolute file path (below). What are the pitfalls of using an absolute file path?

In a collaborative setting, might want to note what the working directory should be set as

```
# Absolute file path is commented out
#setwd("/Users/katerisalk/Documents/Duke/Courses/Environmental_Data_Analytics")
```

You may change your working directory without coding by going to the Session menu in RStudio and navigating to the Set Working Directory tab. From there, you may select from a series of options to reset your working directory.

Another option is to use the R package `here`. We will not be using this option in class, but it is growing quite popular among R users. A more detailed description and rationale can be found here: https://github.com/jennybc/here_here.

Load your packages

At the top of your R scripts, you should load any packages that need to be used for that R script. A common issue that arises is that packages will be loaded in the middle of the code, making it difficult to run specific chunks of code without scrolling to make sure all necessary packages are loaded. For example, the tidyverse package is one that we will use regularly in class.

At the same time, you should also load your theme if you are doing any data visualization with ggplot. More on this later.

```
# Load package
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.2

## -- Attaching packages ----------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0

## Warning: package 'ggplot2' was built under R version 3.5.2

## Warning: package 'tibble' was built under R version 3.5.2

## Warning: package 'tidyr' was built under R version 3.5.2

## Warning: package 'readr' was built under R version 3.5.2

## Warning: package 'dplyr' was built under R version 3.5.2

## -- Conflicts -------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#the advantage is that 'library' tells you any errors, 'require' does not

#best practice to load all packages right at the top
```

Import your datasets

Datasets can be imported into R. Good data practices dictate that raw data (from yourself or others) should not be changed and re-saved within the spreadsheet, but rather the data should be changed with reproducible techniques and saved as a new file. Note: data should be saved in nonproprietary formats, namely .csv or .txt files rather than .xls or .xlsx files.

To read in a data file, you may specify a file path with an *absolute* or a *relative* file path. As above with your working directory, it is a better practice to use a relative directory. To navigate a relative file path, use `./` followed by the tab key to navigate forward in the folder structure, and use `../` followed by the tab key to navigate back out of the folder structure. For example, this lesson is located in the "Lessons" folder, and we need to navigate into the "Data" folder. After clicking the correct folder, use `/` and press tab again to continue the process.

You may also import datasets from the Files tab, but this is not recommended since this is not reproducible.

Note: In the Knit menu in the Editor, you will need to specify whether your knit directory should be the document directory or the project directory. For class today, we will need the directory to be set to the project directory so that we can access relative file paths correctly. However, if you are knitting your document into a PDF, it may be desirable to set your knit directory back to the document directory so that the RMarkdown file and the PDF are saved in the same place.

```
# Absolute file path (not recommended)
#read.csv("/Users/katerisalk/Documents/Duke/Courses/Environmental_Data_Analytics/Data/Raw/USGS_Site0208

# Relative file path (friendly for users regardless of machine)
```

```r
USGS.flow.data <- read.csv("./Data/Raw/USGS_Site02085000_Flow_Raw.csv")
#sko: this goes down a level from the working direction, then down to data, raw, and file

# What happens if we don't assign a name to our imported dataset?
#read.csv("./Data/Raw/USGS_Site02085000_Flow_Raw.csv")

# Another option is to choose with your browser
# read.csv(file.choose())
#sko notes: this is not reproducible

# To import .txt files, use read.table rather than read.csv
#read.table()
```

EXPLORE YOUR DATASET

Take a moment to read through the README file associated with the USGS dataset on discharge at the Eno River. Where can you find this file? How does the placement and information found in this file relate to the best practices for reproducible data analysis? > ANSWER:

```r
View(USGS.flow.data)
# Alternate option: click on data frame in Environment tab

class(USGS.flow.data)
```

```
## [1] "data.frame"
```

```r
colnames(USGS.flow.data)
```

```
##  [1] "agency_cd"                "site_no"
##  [3] "datetime"                 "X165986_00060_00001"
##  [5] "X165986_00060_00001_cd" "X165987_00060_00002"
##  [7] "X165987_00060_00002_cd" "X84936_00060_00003"
##  [9] "X84936_00060_00003_cd"  "X84937_00065_00001"
## [11] "X84937_00065_00001_cd"  "X84938_00065_00002"
## [13] "X84938_00065_00002_cd"  "X84939_00065_00003"
## [15] "X84939_00065_00003_cd"
```

```r
# Rename columns
colnames(USGS.flow.data) <- c("agency_cd", "site_no", "datetime",
                              "discharge.max", "discharge.max.approval",
                              "discharge.min", "discharge.min.approval",
                              "discharge.mean", "discharge.mean.approval",
                              "gage.height.max", "gage.height.max.approval",
                              "gage.height.min", "gage.height.min.approval",
                              "gage.height.mean", "gage.height.mean.approval")
#tibbles give R an advantage over python
#tibbles are faster than dataframes
str(USGS.flow.data)
```

```
## 'data.frame':    33216 obs. of  15 variables:
##  $ agency_cd                : Factor w/ 1 level "USGS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ site_no                  : int  2085000 2085000 2085000 2085000 2085000 2085000 2085000 2085000 20
##  $ datetime                 : Factor w/ 33216 levels "1/1/00","1/1/01",..: 20 1021 2022 2295 2386 24
##  $ discharge.max            : num  74 61 56 54 48 47 44 41 44 57 ...
##  $ discharge.max.approval   : Factor w/ 4 levels "","A","A:e","P": 2 2 2 2 2 2 2 2 2 2 ...
##  $ discharge.min            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ discharge.min.approval   : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ discharge.mean           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ discharge.mean.approval  : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gage.height.max          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gage.height.max.approval : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gage.height.min          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gage.height.min.approval : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gage.height.mean         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gage.height.mean.approval: Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
```

```r
dim(USGS.flow.data)
```

```
## [1] 33216    15
```

```r
length(USGS.flow.data)
```

```
## [1] 15
```

```r
head(USGS.flow.data)
```

```
##   agency_cd site_no datetime discharge.max discharge.max.approval
## 1      USGS 2085000   1/1/28            74                      A
## 2      USGS 2085000   1/2/28            61                      A
## 3      USGS 2085000   1/3/28            56                      A
## 4      USGS 2085000   1/4/28            54                      A
## 5      USGS 2085000   1/5/28            48                      A
## 6      USGS 2085000   1/6/28            47                      A
##   discharge.min discharge.min.approval discharge.mean
## 1            NA                                    NA
## 2            NA                                    NA
## 3            NA                                    NA
## 4            NA                                    NA
## 5            NA                                    NA
## 6            NA                                    NA
##   discharge.mean.approval gage.height.max gage.height.max.approval
## 1                                      NA
## 2                                      NA
## 3                                      NA
## 4                                      NA
## 5                                      NA
## 6                                      NA
##   gage.height.min gage.height.min.approval gage.height.mean
## 1              NA                                        NA
## 2              NA                                        NA
## 3              NA                                        NA
## 4              NA                                        NA
## 5              NA                                        NA
## 6              NA                                        NA
##   gage.height.mean.approval
## 1
## 2
## 3
## 4
## 5
## 6
```

```r
head(USGS.flow.data, 10)
```

```
##    agency_cd site_no datetime discharge.max discharge.max.approval
## 1       USGS 2085000   1/1/28            74                      A
## 2       USGS 2085000   1/2/28            61                      A
## 3       USGS 2085000   1/3/28            56                      A
## 4       USGS 2085000   1/4/28            54                      A
## 5       USGS 2085000   1/5/28            48                      A
## 6       USGS 2085000   1/6/28            47                      A
## 7       USGS 2085000   1/7/28            44                      A
## 8       USGS 2085000   1/8/28            41                      A
## 9       USGS 2085000   1/9/28            44                      A
## 10      USGS 2085000  1/10/28            57                      A
##    discharge.min discharge.min.approval discharge.mean
## 1             NA                                    NA
## 2             NA                                    NA
## 3             NA                                    NA
## 4             NA                                    NA
## 5             NA                                    NA
## 6             NA                                    NA
## 7             NA                                    NA
## 8             NA                                    NA
## 9             NA                                    NA
## 10            NA                                    NA
##    discharge.mean.approval gage.height.max gage.height.max.approval
## 1                                       NA
## 2                                       NA
## 3                                       NA
## 4                                       NA
## 5                                       NA
## 6                                       NA
## 7                                       NA
## 8                                       NA
## 9                                       NA
## 10                                      NA
##    gage.height.min gage.height.min.approval gage.height.mean
## 1               NA                                        NA
## 2               NA                                        NA
## 3               NA                                        NA
## 4               NA                                        NA
## 5               NA                                        NA
## 6               NA                                        NA
## 7               NA                                        NA
## 8               NA                                        NA
## 9               NA                                        NA
## 10              NA                                        NA
##    gage.height.mean.approval
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
```

```r
tail(USGS.flow.data, 5)
```

```
##       agency_cd site_no datetime discharge.max discharge.max.approval
## 33212      USGS 2085000  12/5/18          76.7                      P
## 33213      USGS 2085000  12/6/18          68.9                      P
## 33214      USGS 2085000  12/7/18          65.2                      P
## 33215      USGS 2085000  12/8/18          64.0                      P
## 33216      USGS 2085000  12/9/18         149.0                      P
##       discharge.min discharge.min.approval discharge.mean
## 33212          68.9                      P           73.7
## 33213          62.8                      P           66.2
## 33214          60.4                      P           63.2
## 33215          60.4                      P           61.5
## 33216          60.4                      P           91.6
##       discharge.mean.approval gage.height.max gage.height.max.approval
## 33212                       P            2.55                        P
## 33213                       P            2.49                        P
## 33214                       P            2.46                        P
## 33215                       P            2.45                        P
## 33216                       P            2.97                        P
##       gage.height.min gage.height.min.approval gage.height.mean
## 33212            2.49                        P             2.53
## 33213            2.44                        P             2.47
## 33214            2.42                        P             2.44
## 33215            2.42                        P             2.43
## 33216            2.42                        P             2.64
##       gage.height.mean.approval
## 33212                         P
## 33213                         P
## 33214                         P
## 33215                         P
## 33216                         P
```

```r
USGS.flow.data[30000:30005, c(3, 8, 14)]
```

```
##       datetime discharge.mean gage.height.mean
## 30000  2/18/10           63.4             2.15
## 30001  2/19/10           56.9             2.08
## 30002  2/20/10           53.1             2.03
## 30003  2/21/10           50.4             1.99
## 30004  2/22/10           60.5             2.11
## 30005  2/23/10           80.5             2.34
```

```r
#give rows 30000 to 30005, columns 3, 8, 14

class(USGS.flow.data$datetime)
```

```
## [1] "factor"
```

```r
class(USGS.flow.data$discharge.mean)
```

```
## [1] "numeric"
```

```
class(USGS.flow.data$gage.height.mean)
```

```
## [1] "numeric"
```

```
#pay attention to the class - it will import and sort data differently this way

summary(USGS.flow.data)
```

```
##  agency_cd       site_no           datetime     discharge.max    
##  USGS:33216   Min.   :2085000   1/1/00 :     1   Min.   :   0.02  
##               1st Qu.:2085000   1/1/01 :     1   1st Qu.:   9.80  
##               Median :2085000   1/1/02 :     1   Median :  25.00  
##               Mean   :2085000   1/1/03 :     1   Mean   :  64.66  
##               3rd Qu.:2085000   1/1/04 :     1   3rd Qu.:  55.00  
##               Max.   :2085000   1/1/05 :     1   Max.   :4730.00  
##                                 (Other):33210   NA's   :5113     
##  discharge.max.approval discharge.min     discharge.min.approval
##     : 5113              Min.   :   0.09     :24777              
##  A   :27699              1st Qu.:   1.90   A: 8311              
##  A:e:  276               Median :   3.62   P:  128              
##  P  :  128               Mean   :  16.82                        
##                          3rd Qu.:  16.50                        
##                          Max.   :1200.00                        
##                          NA's   :24777                          
##  discharge.mean     discharge.mean.approval gage.height.max 
##  Min.   :   0.220    :28049                 Min.   : 0.890  
##  1st Qu.:   5.005   A: 5039                 1st Qu.: 1.470  
##  Median :  15.200   P:  128                 Median : 1.800  
##  Mean   :  44.598                           Mean   : 2.062  
##  3rd Qu.:  40.600                           3rd Qu.: 2.250  
##  Max.   :3270.000                           Max.   :17.000  
##  NA's   :28049                              NA's   :28052   
##  gage.height.max.approval gage.height.min gage.height.min.approval
##   :28052                  Min.   :0.840    :28171                 
##  A: 5038                  1st Qu.:1.380   A: 4919                 
##  P:  126                  Median :1.640   P:  126                 
##                           Mean   :1.707                           
##                           3rd Qu.:2.000                           
##                           Max.   :7.930                           
##                           NA's   :28171                           
##  gage.height.mean gage.height.mean.approval
##  Min.   : 0.870    :28171                  
##  1st Qu.: 1.430   A: 4919                  
##  Median : 1.720   P:  126                  
##  Mean   : 1.856                            
##  3rd Qu.: 2.120                            
##  Max.   :14.470                            
##  NA's   :28171
```

```
summary(USGS.flow.data$discharge.mean)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
##    0.220    5.005   15.200   44.598   40.600 3270.000    28049
```

```
summary(USGS.flow.data$gage.height.mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.870   1.430   1.720   1.856   2.120  14.470   28171 
```

What happened to blank cells in the spreadsheet when they were imported into R? > ANSWER: some are NA and some are blank. depends on the class

TIPS AND TRICKS: SPREADSHEETS

*Files should be saved as .csv or .txt for easy import into R. Note that complex formatting, including formulas in Excel, are not saved when spreadsheets are converted to comma separated or text formats (i.e., values alone are saved).

*The first row is reserved for column headers.

*A second, secondary row for column headers (e.g., units) should not be used if data are being imported into R. Incorporate units into the first row column headers if necessary.

*Short names are preferred for column headers, to the extent they are informative. Additional information can be stored in comments within R scripts and/or in README files.

*Spaces in column names will be replaced with a `.` when imported into R. When designing spreadsheets, avoid spaces in column headers.

*Avoid symbols in column headers. This can cause issues when importing into R.