4: Data Exploration

Environmental Data Analytics | Kateri Salk

Spring 2020

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 in 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] "/Users/emilymcnamara/Desktop/Env Data Analytics/Environmental_Data_Analytics_2020"
```

```
# Asks R to tell you what it has set as your working directory
```

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?

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

# This allows you to set a specific working directory as one of your choosing
# Can also go to "session" and "set working directory" and can choose a location
# Can comment to yourself that this script will be set to a certain file path and if it's not set to th
```

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)

# Allows you to see which packages you've loaded and where there may be some conflicts
# Have to write "message = FALSE" to ensure that document can be knitted so the warning messages aren't
```

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.

We're going to be using csv 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.

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

# MetaData file in Data folder includes specific info for the corresponding raw data files
# If have multiple tabs in an excel file, separate tabs and save each one as csv file to upload it

# Relative file path (friendly for users regardless of machine)
USGS.flow.data <- read.csv("./Data/Raw/USGS_Site02085000_Flow_Raw.csv")

# Because we know R is pointing to the project file that contains our data, we can have it search for a
# starting with "." means "go one folder in"
# Do ".." to mean go out when folder and then in another folder
# Even though working directory is sent to Project, Knit directory may not be, so have to set Knit dire
# Can also go directly to folder on desktop and copy the path name of the specific data file

# What happens if we don't assign a name to our imported dataset?
#read.csv("./Data/Raw/USGS_Site02085000_Flow_Raw.csv")
## If it's not named, R just reads the data and tells you what it said. It won't be called up in Enviro
```

```
# Another option is to choose with your browser
#read.csv(file.choose())
# pulls up finder to choose specific folder BUT not reproducible because others can't tell which folder

# 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: You find it under Data –> Metadata. It provides descriptions and clarity to what abbreviations and numeric codes mean. It also provides the source, so people can go directly to the data to navigate it through the main page. Shows date of access to, so if it was accessed on an earlier date, people can see if the data was revised more recently. Includes info on how to save the data in a way that is consistent and others can find it: Files are named according to the following naming convention: `databasename_datatype_details_stage.format`, where: (see meta data info at bottom of doc)

```
View(USGS.flow.data)
```

```
## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):
## running command ''/usr/bin/otool' -L '/Library/Frameworks/R.framework/Resources/
## modules/R_de.so'' had status 1
```

```
# Alternate option: click on data frame in Environment tab

class(USGS.flow.data)
```

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

```
# Tells you the class which is "data frame". This means its 2D and that it contains different modes "nu
colnames(USGS.flow.data)
```

```
##  [1] "agency_cd"              "site_no"                "datetime"
##  [4] "X165986_00060_00001"    "X165986_00060_00001_cd" "X165987_00060_00002"
##  [7] "X165987_00060_00002_cd" "X84936_00060_00003"     "X84936_00060_00003_cd"
## [10] "X84937_00065_00001"     "X84937_00065_00001_cd"  "X84938_00065_00002"
## [13] "X84938_00065_00002_cd"  "X84939_00065_00003"     "X84939_00065_00003_cd"
```

```
# Not very informative

# 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")
# Have to list all the column names, can't just list specific ones. Got these column names from Meta Da
# This only saves the renamed columns of dataframe in R, not in excel. Need to specifically code it to
# General coding rule: Don't surpass 80 characters in a single line. Can see character count on bottom
# If want to specify, write: colnames(USGS.flow.data)[] and put specific names in there. You can put "3

str(USGS.flow.data)
```

```
## 'data.frame':    33690 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/ 33690 levels "1/1/00","1/1/01",..: 2873 3896 4919 5198 5291 5
##  $ discharge.max            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ discharge.max.approval   : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ 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  39 39 39 39 39 39 39 39 39 39 ...
##  $ discharge.mean.approval  : Factor w/ 4 levels "","A","A:e","P": 2 2 2 2 2 2 2 2 2 2 ...
##  $ 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 ...
```

```
# Structure of a data frame. Tells us: you have a data frame and then the size of the data frame and gi
# shows some columns as N/A and some as just blanks. I.e. Discharge max is numeric so it's N/A, Dischar

dim(USGS.flow.data)
```

```
## [1] 33690    15
```

```
# Dimensions of the data frame. Can also see this info in the Environment tab and from structure.

length(USGS.flow.data)
```

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

```
# Length gives the number of columns NOT the number of rows

nrow(USGS.flow.data)
```

```
## [1] 33690
```

```
# Gives the number of rows

head(USGS.flow.data)
```

```
##   agency_cd site_no datetime discharge.max discharge.max.approval discharge.min
## 1      USGS 2085000  10/1/27            NA                                   NA
## 2      USGS 2085000  10/2/27            NA                                   NA
## 3      USGS 2085000  10/3/27            NA                                   NA
## 4      USGS 2085000  10/4/27            NA                                   NA
## 5      USGS 2085000  10/5/27            NA                                   NA
## 6      USGS 2085000  10/6/27            NA                                   NA
##   discharge.min.approval discharge.mean discharge.mean.approval gage.height.max
## 1                                     39                       A              NA
## 2                                     39                       A              NA
## 3                                     39                       A              NA
## 4                                     39                       A              NA
## 5                                     39                       A              NA
## 6                                     39                       A              NA
##   gage.height.max.approval gage.height.min gage.height.min.approval
## 1                                       NA
## 2                                       NA
```

```
## 3                                       NA
## 4                                       NA
## 5                                       NA
## 6                                       NA
##   gage.height.mean gage.height.mean.approval
## 1               NA
## 2               NA
## 3               NA
## 4               NA
## 5               NA
## 6               NA
```

```
# Gives idea of what the first 6 rows look like
head(USGS.flow.data, 10)
```

```
##    agency_cd site_no datetime discharge.max discharge.max.approval
## 1       USGS 2085000  10/1/27            NA
## 2       USGS 2085000  10/2/27            NA
## 3       USGS 2085000  10/3/27            NA
## 4       USGS 2085000  10/4/27            NA
## 5       USGS 2085000  10/5/27            NA
## 6       USGS 2085000  10/6/27            NA
## 7       USGS 2085000  10/7/27            NA
## 8       USGS 2085000  10/8/27            NA
## 9       USGS 2085000  10/9/27            NA
## 10      USGS 2085000 10/10/27            NA
##    discharge.min discharge.min.approval discharge.mean discharge.mean.approval
## 1             NA                                   39                         A
## 2             NA                                   39                         A
## 3             NA                                   39                         A
## 4             NA                                   39                         A
## 5             NA                                   39                         A
## 6             NA                                   39                         A
## 7             NA                                   39                         A
## 8             NA                                   39                         A
## 9             NA                                   39                         A
## 10            NA                                   39                         A
##    gage.height.max gage.height.max.approval gage.height.min
## 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.min.approval gage.height.mean gage.height.mean.approval
## 1                                        NA
## 2                                        NA
## 3                                        NA
## 4                                        NA
## 5                                        NA
## 6                                        NA
```

```
## 7                                       NA
## 8                                       NA
## 9                                       NA
## 10                                      NA
```

```
# Gives first 10 rows
tail(USGS.flow.data, 5)
```

```
##       agency_cd site_no datetime discharge.max discharge.max.approval
## 33686      USGS 2085000 12/22/19            NA
## 33687      USGS 2085000 12/23/19            NA
## 33688      USGS 2085000 12/24/19            NA
## 33689      USGS 2085000 12/25/19            NA
## 33690      USGS 2085000 12/26/19            NA
##       discharge.min discharge.min.approval discharge.mean
## 33686            NA                                 18.1
## 33687            NA                                 18.6
## 33688            NA                                 18.8
## 33689            NA                                 16.6
## 33690            NA                                 15.1
##       discharge.mean.approval gage.height.max gage.height.max.approval
## 33686                       P              NA
## 33687                       P              NA
## 33688                       P              NA
## 33689                       P              NA
## 33690                       P              NA
##       gage.height.min gage.height.min.approval gage.height.mean
## 33686              NA                                     1.93
## 33687              NA                                     1.94
## 33688              NA                                     1.95
## 33689              NA                                     1.91
## 33690              NA                                     1.88
##       gage.height.mean.approval
## 33686                         P
## 33687                         P
## 33688                         P
## 33689                         P
## 33690                         P
```

```
# Gives the last 5 rows
USGS.flow.data[30000:30005, c(3, 8, 14)]
```

```
##       datetime discharge.mean gage.height.mean
## 30000 11/18/09           27.5             1.72
## 30001 11/19/09           31.6             1.80
## 30002 11/20/09           37.1             1.88
## 30003 11/21/09           32.1             1.80
## 30004 11/22/09           23.7             1.66
## 30005 11/23/09          337.0             3.87
```

```
# Matrix subsetting: lines 30,000 to 30,000 and specific column and row by a specific amount

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

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

```
# Can ask for specific columns. I.e. specific date and time by just writing USGS.flow.data$datetime wit
# Adding "class" tells you which class it is
class(USGS.flow.data$discharge.mean)
```

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

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

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

```
summary(USGS.flow.data)
```

```
##  agency_cd       site_no            datetime     discharge.max
##  USGS:33690   Min.   :2085000   1/1/00 :    1   Min.   :   0.26
##               1st Qu.:2085000   1/1/01 :    1   1st Qu.:   7.23
##               Median :2085000   1/1/02 :    1   Median :  21.15
##               Mean   :2085000   1/1/03 :    1   Mean   :  88.15
##               3rd Qu.:2085000   1/1/04 :    1   3rd Qu.:  59.80
##               Max.   :2085000   1/1/05 :    1   Max.   :4730.00
##                                 (Other):33684   NA's   :28342
##  discharge.max.approval discharge.min     discharge.min.approval
##   :28342                Min.   :   0.09    :28342
##  A: 5347                1st Qu.:   4.38   A: 5347
##  P:    1                Median :  12.60   P:    1
##                         Mean   :  30.46
##                         3rd Qu.:  34.80
##                         Max.   :1460.00
##                         NA's   :28342
##  discharge.mean    discharge.mean.approval gage.height.max
##  Min.   :   0.02      : 5108               Min.   : 0.890
##  1st Qu.:   9.30   A  :28265               1st Qu.: 1.490
##  Median :  24.00   A:e:  294               Median : 1.830
##  Mean   :  59.48   P  :   23               Mean   : 2.124
##  3rd Qu.:  54.00                           3rd Qu.: 2.310
##  Max.   :4600.00                           Max.   :17.020
##  NA's   :5108                              NA's   :28229
##  gage.height.max.approval gage.height.min gage.height.min.approval
##   :28229                  Min.   :0.840    :28229
##  A: 5460                  1st Qu.:1.380   A: 5460
##  P:    1                  Median :1.650   P:    1
##                           Mean   :1.736
##                           3rd Qu.:2.030
##                           Max.   :9.190
##                           NA's   :28229
##  gage.height.mean gage.height.mean.approval
##  Min.   : 0.870    :24870
##  1st Qu.: 1.450   A: 8797
##  Median : 1.770   P:   23
##  Mean   : 1.952
##  3rd Qu.: 2.200
##  Max.   :15.040
##  NA's   :24870
```

```
# Structure can be more interprative. Factors won't appear as conveniently as numeric values
```

What happened to blank cells in the spreadsheet when they were imported into R? > Answer:

Adjusting Datasets

Removing NAs

Notice in our dataset that our discharge and gage height observations have many NAs, meaning no measurement was recorded for a specific day. In some cases, it might be in our best interest to remove NAs from a dataset. Removing NAs or not will depend on your research question.

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.02    9.30   24.00   59.48   54.00 4600.00    5108
```

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.870   1.450   1.770   1.952   2.200  15.040   24870
```

```
# Gives distribution and tells you how many N/As are in the data frame. Quite a few in this case
```

Question: What types of research questions might make it favorable to remove NAs from a dataset, and what types of research questions might make it favorable to retain NAs in the dataset?

> Answer: Helpful to retain for future reference. Removing it can help when running analysis on presence data instead of presence/absence. If want to compare discharge mean and height mean, only want to do it where values are present. Also, if want to run a correlation matrix, NAs screw it up so need to remove it.

```
USGS.flow.data.complete <- na.omit(USGS.flow.data)
# If making permenant changes to data frame, want to call it something else or else you have to rerun c
# na.omit only runs the complete datasets in your data frame
dim(USGS.flow.data)
```

```
## [1] 33690    15
```

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

```
## [1] 5342   15
```

```
# A lot fewer rows without NAs

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

```
## [1] 51.08613
```

```
sd(USGS.flow.data.complete$discharge.mean)
```

```
## [1] 137.2094
```

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

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.220    5.683   16.600   51.086   44.800 3270.000
```

```
# These are means, sd's, and summary of the complete dataset without the NAs
# Summary doesn't show sd, so need to run that separately
```

Formatting dates

R will often import dates as factors or characters rather than dates. To fix, this we need to tell R that it is looking at dates. We also need to specify the format the dates are in. By default, if you don't provide a format, R will attempt to use %Y-%m-%d or %Y/%m/%d as a default. Note: if you are working collaboratively

in an international setting, using a year-month-day format in spreadsheets is the least ambiguous of date formats. Make sure to check whether month-day-year or day-month-year is used in an ambiguously formatted spreadsheet.

Formatting of dates in R:

%d day as number (0-31) %m month (00-12, can be e.g., 01 or 1) %y 2-digit year %Y 4-digit year %a abbreviated weekday %A unabbreviated weekday %b abbreviated month %B unabbreviated month

In some cases when dates are provided as integers, you may need to provide an origin for your dates. Beware: the "origin" date for Excel (Windows), Excel (Mac), R, and MATLAB all have different origin dates. Google this if it comes up. Origin will be January 1st, 1970

```
help(as.Date)

# Adjust date formatting for today
# Write code for three different date formats.
# An example is provided to get you started.
# (code must be uncommented)
today <- Sys.Date()
# Whatever day your computer thinks it is will be the date that's called up. It shows up in the environ

format(today, format = "%B")
```

```
## [1] "January"
```

```
#Formats today as the month

format(today, format = "%d")
```

```
## [1] "21"
```

```
format(today, format = "%b")
```

```
## [1] "Jan"
```

```
format(today, format = "%y")
```

```
## [1] "20"
```

```
USGS.flow.data$datetime <- as.Date(USGS.flow.data$datetime, format = "%m/%d/%y")
# If want R to perceive date/time column as date, have to tell R which column, and which format exists
#BUT it says it's 2027 instead of 1997 because all the dates prior to 1969 are noted in the future
```

Note that for every date prior to 1969, R has assigned the date in the 2000s rather than the 1900s. This can be fixed with an `ifelse` statement inside a function. Run through the code below and write what is happening in the comment above each line.

```
#
USGS.flow.data$datetime <- format(USGS.flow.data$datetime, "%y%m%d")

#
create.early.dates <- (function(d) {
       paste0(ifelse(d > 181231,"19","20"),d)
       })
#
USGS.flow.data$datetime <- create.early.dates(USGS.flow.data$datetime)

#
USGS.flow.data$datetime <- as.Date(USGS.flow.data$datetime, format = "%Y%m%d")
```

Saving datasets

We just edited our raw dataset into a processed form. We may want to return to this processed dataset later, which will be easier to do if we save it as a spreadsheet.

```
write.csv(USGS.flow.data, file = "./Data/Processed/USGS_Site02085000_Flow_Processed.csv", row.names=FALS

# Want to store apart from raw data file. Have to make a folder in R Data folder named "Processed" in o
```

Tips and Tricks

Knitting

- 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. If your document is not knitting correctly, try switching between the document directory and project directory as a first troubleshooting option.

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 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.