ENVIRONMENTAL DATA ANALYTICS: M3 – DATA EXPLORATION

Spring 2024

Nicholas School of the Environment - Duke University

Part 1.1

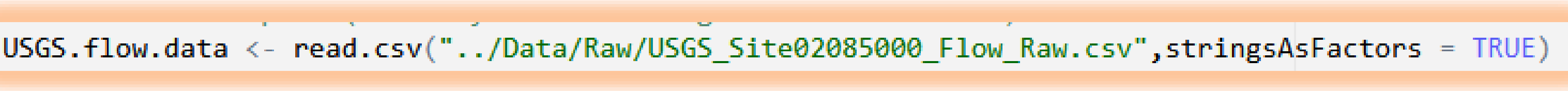
Q&A on Data Exploration

- Best practices in coding
  - Loading packages
  - Importing datasets
- Exploring data
  - Absolute vs relative paths
  - Missing data
  - Dates
  - Saving processed data

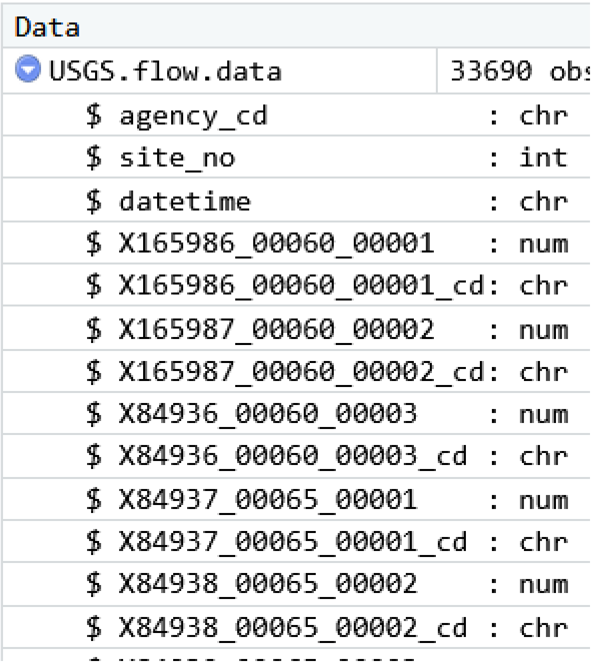
Q&A: **Importing datasets**

- □ **Include** `stringAsFactors = True` **when importing files**

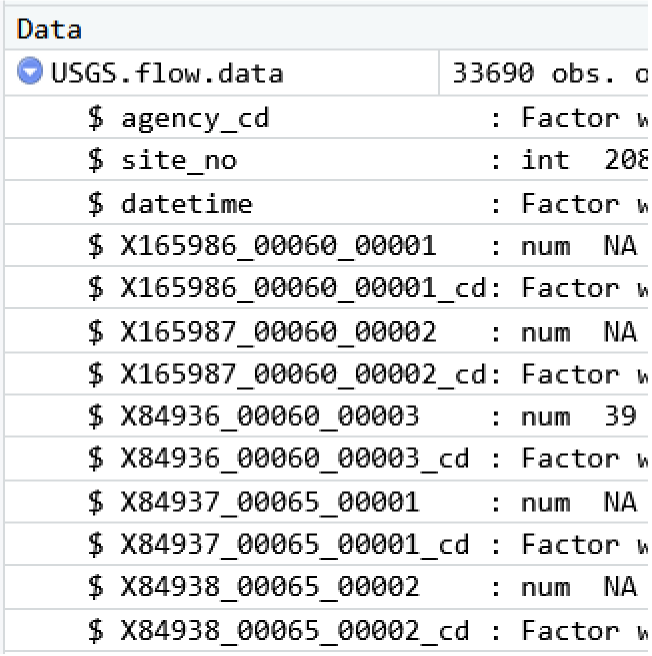
Line 100…



```
USGS.flow.data <- read.csv("../Data/Raw/USGS_Site02085000_Flow_Raw.csv",stringsAsFactors = TRUE)
```



```
Data
USGS.flow.data                  33690 obs
    $ agency_cd             : chr
    $ site_no               : int
    $ datetime              : chr
    $ X165986_00060_00001   : num
    $ X165986_00060_00001_cd: chr
    $ X165987_00060_00002   : num
    $ X165987_00060_00002_cd: chr
    $ X84936_00060_00003    : num
    $ X84936_00060_00003_cd : chr
    $ X84937_00065_00001    : num
    $ X84937_00065_00001_cd : chr
    $ X84938_00065_00002    : num
    $ X84938_00065_00002_cd : chr
```



```
Data
USGS.flow.data                  33690 obs. o
    $ agency_cd             : Factor w
    $ site_no               : int  208
    $ datetime              : Factor w
    $ X165986_00060_00001   : num  NA
    $ X165986_00060_00001_cd: Factor w
    $ X165987_00060_00002   : num  NA
    $ X165987_00060_00002_cd: Factor w
    $ X84936_00060_00003    : num  39
    $ X84936_00060_00003_cd : Factor w
    $ X84937_00065_00001    : num  NA
    $ X84937_00065_00001_cd : Factor w
    $ X84938_00065_00002    : num  NA
    $ X84938_00065_00002_cd : Factor w
```

Data types: What are Factors

- *Numeric* vs *character* columns

- *Factors*…
  - …are useful for analyzing/visualizing **categorical** data
  - …have *levels*
  - …can have *labels* too

Part 1.2

Q&A on Visual Data Exploration

Part 2

Review – Data Structures

Coding Challenges!

The “here” package

here() facilitates relative paths in your script

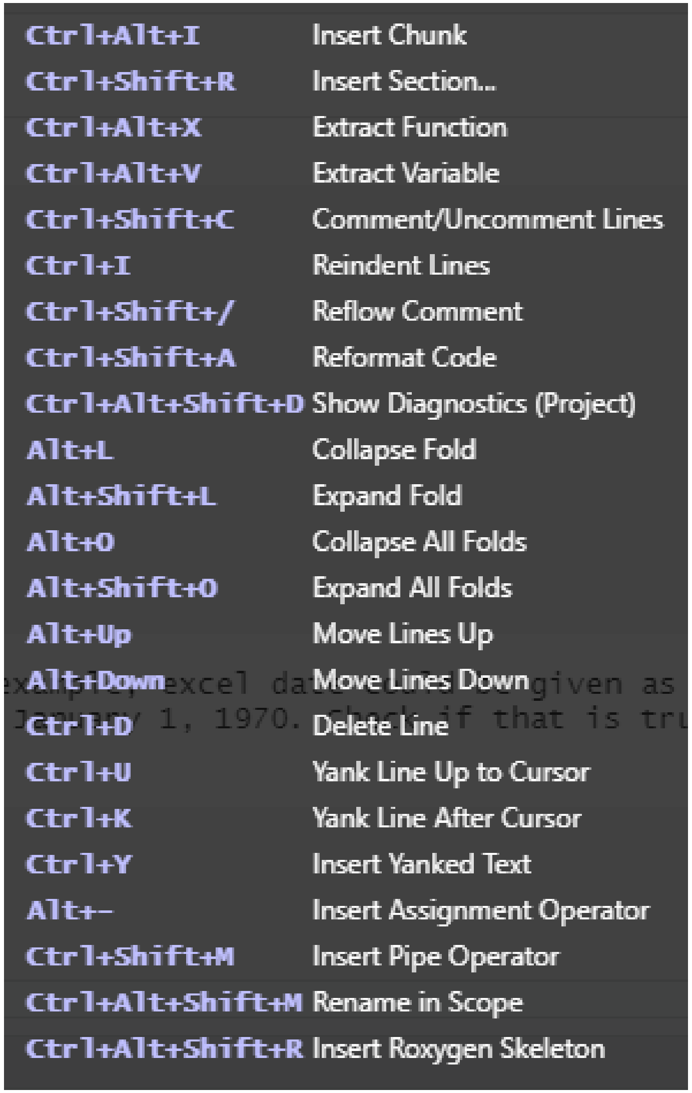
http://jenrichmond.rbind.io/post/where-is-here/

- **`here()`** –
  - points to the project’s “root” folder, i.e. the one containing the `.Rproj` file.
  - Is not affected by setwd()

- **`here('data', 'raw', 'my_file.csv')`**
  - Creates a path to `…/data/raw/my_file.csv`

Tips for the day – Rmd shortcuts

- □ Naming code chunks…
- □ Keyboard shortcuts:



Ctrl+Alt+I	Insert Chunk
Ctrl+Shift+R	Insert Section...
Ctrl+Alt+X	Extract Function
Ctrl+Alt+V	Extract Variable
Ctrl+Shift+C	Comment/Uncomment Lines
Ctrl+I	Reindent Lines
Ctrl+Shift+/	Reflow Comment
Ctrl+Shift+A	Reformat Code
Ctrl+Alt+Shift+D	Show Diagnostics (Project)
Alt+L	Collapse Fold
Alt+Shift+L	Expand Fold
Alt+O	Collapse All Folds
Alt+Shift+O	Expand All Folds
Alt+Up	Move Lines Up
Alt+Down	Move Lines Down
Ctrl+D	Delete Line
Ctrl+U	Yank Line Up to Cursor
Ctrl+K	Yank Line After Cursor
Ctrl+Y	Insert Yanked Text
Alt+-	Insert Assignment Operator
Ctrl+Shift+M	Insert Pipe Operator
Ctrl+Alt+Shift+M	Rename in Scope
Ctrl+Alt+Shift+R	Insert Roxygen Skeleton

Data Structures

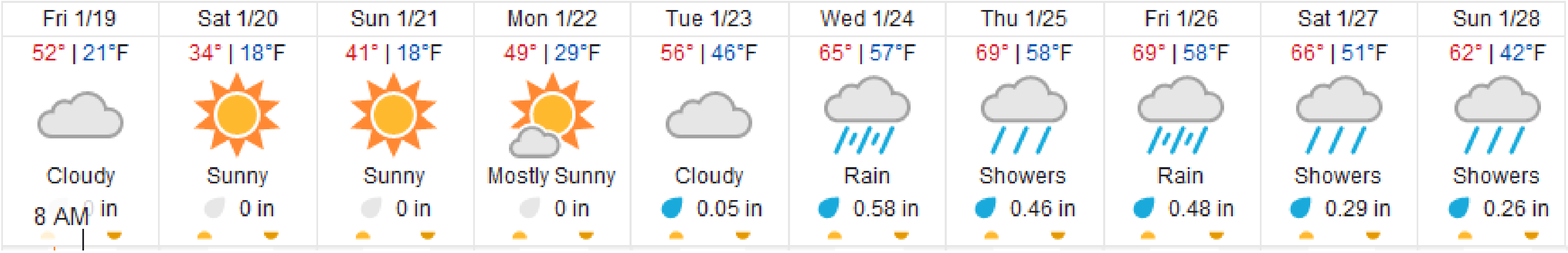
- Vector
- Matrix
- Array
- List
- Data Frame

- What they can hold
- How to construct
- Number of dimensions
- How to extract elements

Coding Challenge #1

- □ Find a ten-day forecast of temperatures (Fahrenheit) for Durham, North Carolina. **Create two vectors**, one representing the high temperature on each of the ten days and one representing the low

https://www.wunderground.com/forecast/us/nc/durham



Fri 1/19	Sat 1/20	Sun 1/21	Mon 1/22	Tue 1/23	Wed 1/24	Thu 1/25	Fri 1/26	Sat 1/27	Sun 1/28
52° | 21°F	34° | 18°F	41° | 18°F	49° | 29°F	56° | 46°F	65° | 57°F	69° | 58°F	69° | 58°F	66° | 51°F	62° | 42°F
Cloudy	Sunny	Sunny	Mostly Sunny	Cloudy	Rain	Showers	Rain	Showers	Showers
8 AM 0 in	0 in	0 in	0 in	0.05 in	0.58 in	0.46 in	0.48 in	0.29 in	0.26 in

Coding Challenge #2 & #3

- Now, create two additional vectors that include the ten-day forecast for the high and low temperatures in Celsius. *Use a function to create the two new vectors from your existing ones in Fahrenheit.*

$$\boxed{(°\mathbf{F} - 32) \times 5/9 = °\mathbf{C}}$$

- *Combine your four vectors into a data frame with informative column names*

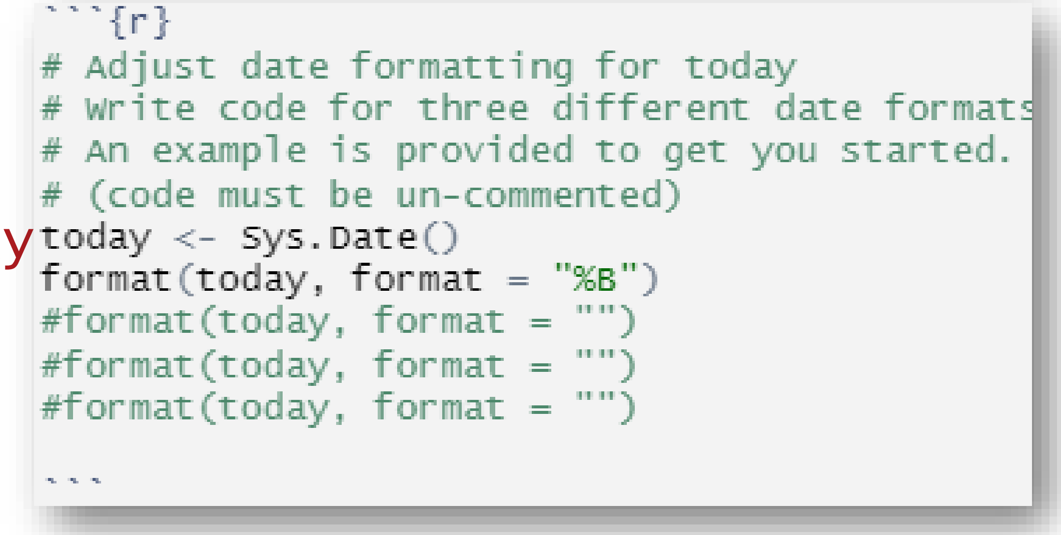
Coding Challenge #4

- Use the common functions `summary` and `sd` to obtain basic data summaries of the ten-day forecast. How would you call these functions differently for the entire data frame vs. a single column?

Coding Challenge #5

- Date formats:

%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



~~~
```{r}
# Adjust date formatting for today
# Write code for three different date formats
# An example is provided to get you started.
# (code must be un-commented)
today <- Sys.Date()
format(today, format = "%B")
#format(today, format = "")
#format(today, format = "")
#format(today, format = "")

```
~~~

The “lubridate” package

- More powerful than `as.date()`
- `ymd()`… `ydm()`… `mdy()`…
- `fast_strptime()` & `parse_dateTime2()`
  - parses character dates into date obj
  - Has a “`cutoff_2000`” feature (to help with Y2K issue)