DataFest Prep

Part 1 of n

A data exploration checklist

- ✓ Get acquainted (with your data)
- ✓ Get curious (about your data)
- ✓ Get it loaded
- ✓ Identify the parts
- ✓ Examine each part (in isolation)
- ✓ Clean and transform (as necessary)
- ✓ Summarize each part (briefly)
- ✓ Formulate questions (for the next phase)

Form a team

Look around for other people with your same data analysis tool

How many on a team?
(Dunno...2? 3? 4?)

How many laptops?
(Dunno...each with their own? All look at one screen?)

Getting set up

1. Create a project directory/folder in which to store the data and your program(s)
   - If you know how to do it, and have time (probably not today), create and connect to a shared github/bitbucket repo
2. Open your tool of choice (Python, R, Excel) and make sure it's "pointed to" that directory/folder
3. Open a text document to store notes about what you learn
   - A plain text file in your github/bitbucket repo would be good if you're using that
   - Google Docs would be good if you're not

Taking notes

You want to keep **a record of what you've learned** because believe me it will otherwise get lost in the madness

- Concisely state conclusions you have drawn, large and small, and how you know they're true

You want to keep **a running list of to-dos** (things you've thought of but haven't had a chance to explore yet)

- You'll think of things far faster than you can actually do them. Jot stuff down as soon as you have an idea so you can come back to it.

Taking notes

```
Stuff we've learned
-------------------
Dataset has National League batting stats for 7 seasons
Way more hitters for 2012-2016 than previous (missing data?)
Batting avg and home runs look pretty normal (exploratory_plots.py)
Stolen bases look bimodal; lots of zeros (exploratory_plots.py)
Mean batting avg: .283 (summary_stats.py)  a bit higher than expected


To dos:
-------
Batting average vs. home runs -- related? (and how)?
Confirm/deny: better teams have higher overall batting average?
    Find overall records for the teams online (maybe just one year)
    Compile team batting averages from individual players (requires loop)
Has batting avg been trending up over time?
```

Today's data set

`cs.umw.edu/~stephen/CPSCFeedback.csv`

(Download it now)

Get acquainted (with your data)

Run your eyeballs over this data set. Get an idea of what it contains.

Use a text editor, or Excel, or a browser, or Python, or R, or anything you want.

Talk out loud with your partner(s) as you look through it.

You have 3 minutes. Go.

Get acquainted (with your data)

?

Get curious (about your data)

Ask yourselves what could be meaningfully learned from this data set.

Brainstorm and write down **at least three semi-concrete questions** that come to mind about it.

You have 3 minutes. Go.

Get curious (about your data)

?

Get it loaded

Start a `.py` or `.R` file, and in it put the code to load the data set from your filesystem into Python/R.

(You'll want to do this in a script rather than ad hoc because it'll take some trial and error, and you'll want it repeatable.)

If the data set is a single CSV (like this one), your goal is a single Pandas or R DataFrame.

Glance at the **column names**, the **number of rows**, and the **head** and the **tail** of the DataFrame to sanity check it all worked.

You have 3 minutes. Go.

In Python:

```
import pandas as pd
cpsc = pd.read_csv("CPSCFeedback.csv")
```

Then in an interactive console:

```
>>> cpsc.columns
>>> len(cpsc)
>>> cpsc.head()
>>> cpsc.tail()
```

In R:

```
cpsc <- read.csv("CPSCFeedback.csv",header=TRUE)
```

Then in an interactive console:

```
> colnames(cpsc)
> nrow(cpsc)
> head(cpsc)
> tail(cpsc)
```

Get it loaded

Other useful quick-and-dirty-summary commands:

- Python: `.info()`, `.describe()`
- R: `str()`, `summary()`