DataFest Prep

Part 2 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)

Teams

If your team from last time is still in the room, great!

If they're not, partner with someone who's new.

If you're new, partner with someone whose team deserted them.

Getting reacquainted

1. Find that project directory/folder in which you stored the data and your program(s).

   (...or create one if you're new)

2. Find that text document you used to store notes about what you learn
   - (maybe it's a plain text file)
   - (maybe it's a Word doc)
   - (maybe it's on Google Docs)

3. Open your tool of choice (Python, R, Excel) and make sure you can still run your code

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?
```

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

In Python:

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

Then in an interactive console:

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

In R:

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

Then in an interactive console:

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

Identify the parts

Look at the “parts” of the data set and identify what each one consists of.

(In our case, “parts” means the columns of our sole DataFrame.)

Reorganize as appropriate:

- Are there some we just flat don’t need? Get rid of them.
- Is there some obvious stuff to clean up? Go ahead and do it.
- Are there some that have fundamentally different types of data than others? Maybe partition into separate DataFrames.

You have 4 minutes. Go.

What I did

```
import pandas as pd

cpsc = pd.read_csv("CPSCFeedback.csv")

# First row has question description, which we’ll condense and put in a
# column name for analysis. Second row has outright crap.
cpsc = cpsc.drop(cpsc.index[0:2])

# Two kinds of data: coded (multiple choice) and free-text. Put into two
# different DataFrames since we’ll treat them very differently.
coded = cpsc[[’Q5’,’Q6’,’Q7’,’Q13’,’Q10’,’Q13.1’,’Q12’]]
coded.columns = [’Why’,’Styles’,’Collaborate’,’Honor’,’Time’,’Level’,
    ’Programming’]
text = cpsc[[’Q5_3_TEXT’,’Q6_4_TEXT’,’Q11’]]
text.columns = [’Why’,’Styles’,’Classes’]

# Don’t need the original DataFrame now.
del cpsc
```

Examine each part (in isolation)

For each part, get the lay of the land.

- For a numeric variable, summary stats and histogram.
- For a categorical variable, 1-d contingency table.
- For text data, histogram of lengths.

You have 4 minutes. Go.

What I did

```
for col in coded:
    print(col.upper() + ":")
    print(coded[col].value_counts())
    print("\n")

for col in text:
    counts = text[col].str.len()
    counts.hist()
    plt.title(col)
```

Clean and transform (as necessary)

Are there values that need to be recoded, binned, separated, or combined in some way?

You have 8 minutes. Go.

What I did

```
replacements = {
    'The love of programing and the subject':'love',
    'To get a good job and make money':'money',
    'Periodic quizzes instead of a midterm':'quizzes',
    'Incremental labs that connect to larger projects':'incremental',
    'Experience Points':'XP',
    ...
}

for old, new in replacements.items():
    coded.replace(old, new, regex=True, inplace=True)
```

...to be continued...