Erica Zhou (*ezhou*)
25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

1

See `wrangler-synsets.py`.

Erica Zhou (*ezhou*) 25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

2

118294 unique words are in the dataset.

```
awk -F ',' '{print $1}' synsets-clean.txt | sort -u | wc -l
```

Erica Zhou (*ezhou*) 25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

3

The following countries have won the World Cup the given # of times:

```
5 BRA
4 ITA
3 GER
2 ARG
2 URU
1 FRA
1 ENG
1 ESP
```

```
egrep '[[:upper:]],1,' data/worldcup-clean.csv | cut -c1-3 | uniq -c
```

Wrangler Transforms:

1. Split repeatedly on `'|-'` into rows
2. Delete row 1
3. Delete empty rows
4. Split data repeatedly on `'|'`
5. Set split1 name to 1
6. Set split2 name to 2
7. Set split3 name to 3
8. Set split4 name to 4
9. Drop split5
10. Fold 1, 2, 3, 4 using header as a key
11. Delete rows where value = `'0 '`
12. Split value repeatedly on `','` into rows

Erica Zhou (*ezhou*) 25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

4

Synsets:

```
cat data/synsets.txt |                          # open file
awk -F ',’                                      # split on commas
’{split($2,words," ");                          # split words (2nd entry) by " "
split($3,defs,";");                             # split defs (3rd entry) by ";"
for (word in words)                             # loop through words
    for (def in defs)                           # loop through definitions
        print a[word],"," b[def];}’             # print word and definition
```

World Cup:

```
tr -d ’\n’ < data/worldcup-semiclean.txt |      # remove newlines in file
tr ’-’ ’\n’ |                                   # create lines on "-" (1 per country)
awk -F ’|’                                      # split on pipe
’NR > 2 {                                       # don’t look at first two lines (header)
for (i = 2; i < 6; i++) {                       # loop through top 4 places (1 is country)
    split($i,place,",");                        # create list of years of given placement
    for (yr in place)                           # loop through years
        if (place[yr] != 0)                     # check if year is 0 (= no placements)
            print $1,",",i-1,",",place[yr];     # print desired info if valid year
    }
}’
```

Erica Zhou (*ezhou*) 25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

5

Data Wrangler had a clearer UI, so it was easier to see at every step what particular transformations were being made and what the resulting table would look like. Additionally, because changes were made through the UI, the syntax was less cluttered because, for example, only delimiters needed to be entered versus the entire command with escape sequences. However, Data Wrangler was definitely more difficult in terms of figuring out exactly what specific commands did if data wasn't in the right format, or if issues prevented the commands from executing correctly. For example, there were a few times my transforms looked okay in the preview but resulted in empty tables, and since the program blackboxed the actual transformation, it was hard to figure out what went wrong.

Command line tools definitely have a steeper learning curve because it is harder to immediately see what's going on. Without the UI, it is more difficult to parse through plain text both when writing the commands (especially with brackets, escape commands, etc. cluttering the command) and reading the responses. However, since the code operates at a lower level, there is a lot more granularity in terms of seeing more intermediate, incremental results, and it is much easier to debug in the smaller increments. Additionally, since the command line operates at a lower level and without having to run through the UI, it overall ran faster, and it was much easier to make, test, and undo changes.

Erica Zhou (*ezhou*) 25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

6

Cleaning script:

```
# create usable format
(echo "Date|Artist|Song|Album|Label|Show|DJ" &&            # add column names
tr -d '\n' < data/wmbr.txt |                               # delete newlines
sed "s/Date:/|/g" | tr '|' '\n' |                          # make newlines by date
sed -e "s/Artist:/|/" -e "s/Song:/|/" -e "s/Album:/|/"     # change all labels to pipe
    -e "s/Label:/|/" -e "s/Show:/|/" -e "s/DJ:/|/")
> data/wmbr-clean.txt                                      # write to new file

# fix typos/casing & add featured artists to 'Artist' column
cat data/wmbr-clean.txt |                                  # open initially cleaned file
sed -e "s/Ellish/Eilish/"                                  # fix Billie Eilish typo & casing
    -e "s/billie eilish/Billie Eilish/"
    -e "s/Rosalia/ROSALIA/"                                # make ROSALIA casing consistent
    -e "s/James Black/James Blake/"                        # fix James Black typo
awk -F '|' 'BEGIN{OFS="|"}                                 # account for delimiter
{gsub(/and/,",",$2); gsub(/\&/,",",$2);                    # make "," consistent artist delimiter
if (match($3,"\(feat. .*\)") > 0)                          # check if there is a featured artist
    $2=$2 "," substr($3, RSTART+7, RLENGTH-8)}             # add featured artist to artists ($2)
{print}'                                                   # print entire line
> data/wmbr-cleaner.txt                                    # write to new file
```

Erica Zhou (*ezhou*) 25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

7

[**'amiina'**, **'Brendan Little'**, **'Gary War'**, **'lindefelt'**, **'Peter Galperin'**, **'Thingy'**, **'White Hills'**] have played or recorded live at WMBR.

```
df = pd.read_csv("data/wmbr-cleaner.txt", delimiter = "|")
df = df[['Artist','Song','Album']]
live_songs = df[df['Song'].str.contains('live',case=False) | df['Album'].str.contains('live',case=False
return sorted(live_songs['Artist'].unique(),key=lambda v: v.upper())
```

Erica Zhou (*ezhou*) 25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

8

DJ	Song
DJ Lipika	3
Brian Sennett	2
L-Train	2
Sara Achour	2
Lisa	1
TJ Connelly	1

are the DJs who have played songs from the Stranger Things soundtrack.

```
df = pd.read_csv("data/wmbr-cleaner.txt", delimiter = "|")
stranger_things = df[df['Album'].str.contains('stranger things',case=False)]
return stranger_things[['DJ','Song']].groupby('DJ').count().sort_values('Song',ascending=False)
```

Erica Zhou (*ezhou*) 25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

9

```
          Song
Year
2019  0.189655
2018  0.224490
2017  0.266667
```

are the yearly proportions of Billie Ellish songs.

```
df = pd.read_csv("data/wmbr-cleaner.txt", delimiter = "|")
df['Year'] = df['Date'].str[-4:]
df = df[df['Year'].isin(['2017','2018','2019'])]
df['Billie'] = df['Artist'].str.contains('billie eilish',case=False)
billie_songs = df[df['Billie']][['Song','Year']].groupby('Year').count()
all_songs = df[['Song','Year']].groupby('Year').count()
return (billie_songs/all_songs).sort_values('Year',ascending=False)
```

Erica Zhou (*ezhou*) 25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

10

index	Song
0 Juice	9
2 Soulmate	3
1 Tempo (feat. Missy Elliott)	3
5 Boys	2
3 Crybaby	2
4 Cuz I Love You	2
6 Good As Hell	2
8 Humanize	2
7 Lingerie	2
9 Water Me	2
11 Fitness	1
16 Jerome	1
12 Juice - Breakbot Mix	1
13 Like A Girl	1
14 Torn Apart, Pt. II (Bastille vs. GRADES vs. Li...	1
15 Truth Hurts	1
10 Worship	1

are the Lizzo songs (and counts) played on WMBR for the years she appeared on talk shows.

```
lizzo = pd.read_json('data/lizzo_appearances.json')
lizzo_years = lizzo[lizzo['Title'].str.contains('show',case=False)]['Year'].unique()

df = pd.read_csv('data/wmbr-cleaner.txt', delimiter = '|')
df['Year'] = df['Date'].str[-4:]
df = df[df['Year'].isin(lizzo_years)]
lizzo_songs = df[df['Artist'].str.contains('lizzo',case=False) |
    df['Song'].str.contains('lizzo',case=False)]
return lizzo_songs['Song'].value_counts().reset_index().sort_values(['Song','index'],
    ascending=[False,True])
```

Erica Zhou (*ezhou*) 25 September 2019
Repo: *https://github.com/ericazhou7/6.s080-labs*
Commit Hash: *61e0274550b2fbb943c38a1fcce0b8afb64f231b*

6.S080 Lab 2

11

Walk It Talk It by **Migos** was the most danceable track by a 2018 Spotify Top 100 artist that was also played on WMBR.

```
wmbr = pd.read_csv('data/wmbr-cleaner.txt',delimiter='|')
wmbr['Artist'] = wmbr['Artist'].str.split(',')
wmbr['Artist'] = wmbr['Artist'].apply(lambda x: [val.strip() for val in x])
wmbr['Artist'] = wmbr['Artist'].apply(lambda x: set(x))
wmbr_artists = set.union(*wmbr['Artist'])

top2018 = pd.read_csv('data/top2018.csv')
top2018_wmbr = top2018[top2018['artists'].isin(wmbr_artists)]
return top2018_wmbr.sort_values('danceability',ascending=False)[['name','artists','danceability']]
```