Programming, Problem Solving, and Algorithms

CPSC203, 2019 W1

Announcements

Lab this week: web-data-viz pipeline

“Problem of the Day” continues!

Today:

An authentic scraping experience

Pandas

Beautiful Soup

Reads the html source into a data structure that's easy to query!

OLD

```
html = simple_get("https://www.billboard.com/charts/hot-100" + '/' + date)
mydivs = html.findAll("div", {"class": "chart-list-item"}) // all the data is here!!
```

tag attr value

```
for div in mydivs:
    s = Song(div.attrs['data-title'], div.attrs['data-artist'], int(div.attrs['data-rank']))
```

```
mylis = html.findAll("li", {"class": "chart-list__element"}) # all the data is here!!
```

top level tag for each song on list.

```
for li in mylis:
    # WHAT SHALL WE DO???
```

NEW

Digging Deeper

```
mylis = html.findAll("_____", {"class": "_______________________"}) # all the data is here!!

for li in mylis:
   s = Song(li.find("span", {"class": "chart-element__information__song"}).string,   title ✓
      li.find("span," {"class": "chart-element_information_artist"}).string
    int(li.find("span", {"class":"chart-element_rank_number"}).string)
   int(li.find("span", {"class": "chart-element_information_text--peak"}).string.split(" ")[0])
      li.find("span", {"class":                                                        )
                          ⋮
```

Last Week?

What data is given as “last week’s rank” for songs that are new to the chart?

```
try:
   s.last_week = int( li.find( “span”,
      {“class”: “chart-element__information__delta__text  text--last”}).string.split(“ “)[0])
except ValueError:
   pass
```

← tests to see if right side of assignment to s.last_week is an integer.

if it isn't then last_week default will be assigned.

Go get the updated scraper!

It’ a treasure hunt!!

1. Find the given code. updated billboard code
2. Remember the instructions for grabbing the given code.
3. Get set up in PyCharm.

Now, find the updates to the web scraping code…

Stop reviewing the code at line 100.

We’ll use pandas for data analysis, so we should learn how to use it…

Pandas and data frames

```
import pandas as pd
```

Imports the pandas library. We will almost always use an abbreviation…

Instead of saying `pandas.read_csv(‘file.csv’)`

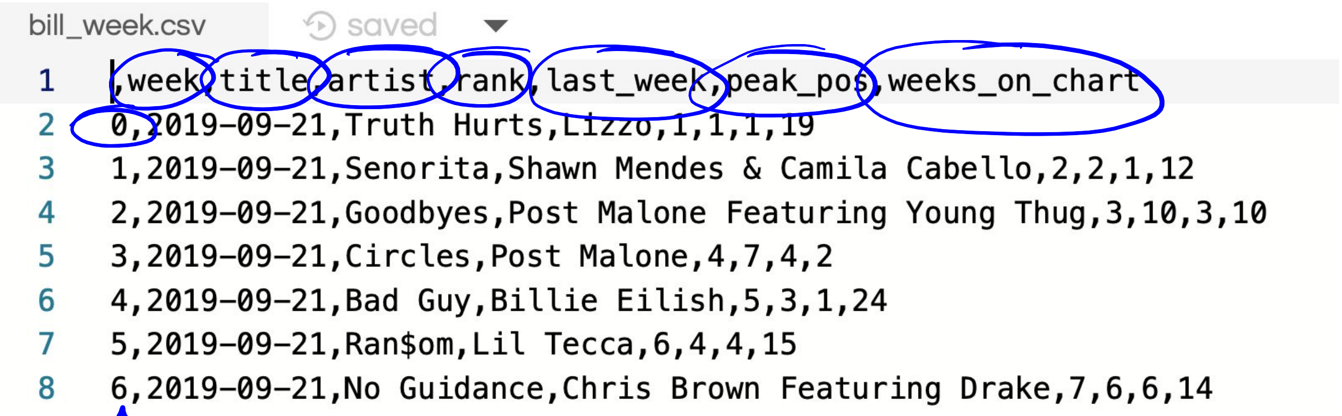
we can say `pd.read_csv('file.csv')`

This function returns a DataFrame containing the data from `file.csv`

CSV files

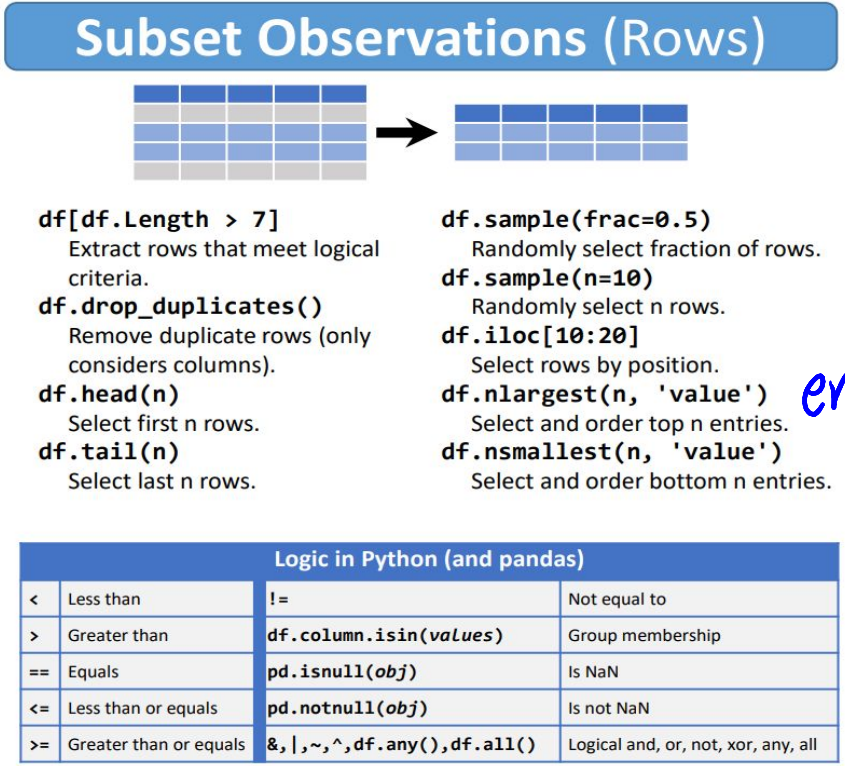
To implement `df = pd.read_csv(‘file.csv’)`

`file.csv` must have field names in row 1, and data beginning in row 2.



```
,week,title,artist,rank,last_week,peak_pos,weeks_on_chart
0,2019-09-21,Truth Hurts,Lizzo,1,1,1,19
1,2019-09-21,Senorita,Shawn Mendes & Camila Cabello,2,2,1,12
2,2019-09-21,Goodbyes,Post Malone Featuring Young Thug,3,10,3,10
3,2019-09-21,Circles,Post Malone,4,7,4,2
4,2019-09-21,Bad Guy,Billie Eilish,5,3,1,24
5,2019-09-21,Ran$om,Lil Tecca,6,4,4,15
6,2019-09-21,No Guidance,Chris Brown Featuring Drake,7,6,6,14
```

Selecting Rows



Subset Observations (Rows)

`df[df.Length > 7]`
Extract rows that meet logical criteria.

`df.drop_duplicates()`
Remove duplicate rows (only considers columns).

`df.head(n)`
Select first n rows.

`df.tail(n)`
Select last n rows.

`df.sample(frac=0.5)`
Randomly select fraction of rows.

`df.sample(n=10)`
Randomly select n rows.

`df.iloc[10:20]`
Select rows by position.

`df.nlargest(n, 'value')`
Select and order top n entries.

`df.nsmallest(n, 'value')`
Select and order bottom n entries.

Logic in Python (and pandas)

<	Less than	!=	Not equal to
>	Greater than	df.column.isin(*values*)	Group membership
==	Equals	pd.isnull(*obj*)	Is NaN
<=	Less than or equals	pd.notnull(*obj*)	Is not NaN
>=	Greater than or equals	&,|,~,^,df.any(),df.all()	Logical and, or, not, xor, any, all

```
df.nlargest(10,‘last_week’)
```

Returns top 10 hits from last week.

```
endurance_df =
df[ df[‘weeks_on_chart’] > 10 ]
```

Returns all songs that have been on the charts for more than 10 weeks.

df.weeks_on_chart

Adding a column

```
df[‘gradient’] = df[‘last_week’] - df[‘rank’]
```

Adds a column to the DataFrame containing the difference for every row.

What does this do?

```
df[ df[‘weeks_on_chart’] > 10 ].count()[‘title’]
```

Some challenges…

Given last week's chart,

1) How many new songs were there?
2) What's the average peak?
3) Among those who were on the list for more than 10wk, what's the average peak? (is it very different than the previous answer?)
4) Which song changed the most? Was it rising or falling?
5) Write and answer your own question:

   ____

POTD #7 Thu

https://github.students.cs.ubc.ca/cpsc203-2019w-t1/potd07

Describe any snags you run into:

1. Line ___: ______________________________
2. Line ___: ______________________________
3. Line ___: ______________________________
4. Line ___: ______________________________
5. Line ___: ______________________________

ToDo for next class...

POTD: Continue every weekday! Submit to repo.

Reading: TLACS Ch 10 & 12 (lists and dictionaries)

References:

https://www.dataschool.io/best-python-pandas-resources/

https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

https://www.crummy.com/software/BeautifulSoup/bs4/doc/