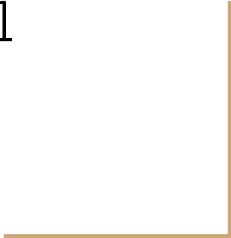


Programming, Problem Solving, and Algorithms

CPSC203, 2019 W1



Announcements

- 1) if POTD takes longer than 20,
pls link.
- 2) course website: [cheeren.github.io/
cpsc208test](https://cheeren.github.io/cpsc208test)

Lab this week: web-data-viz pipeline

“Problem of the Day” continues!

Today:

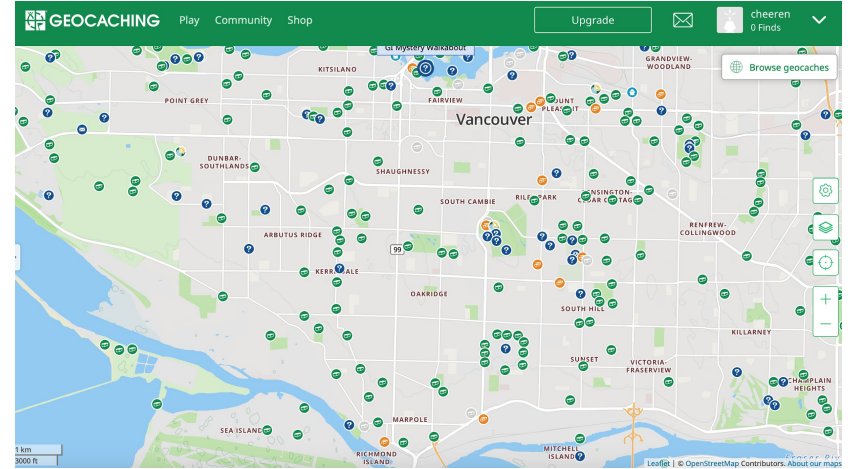
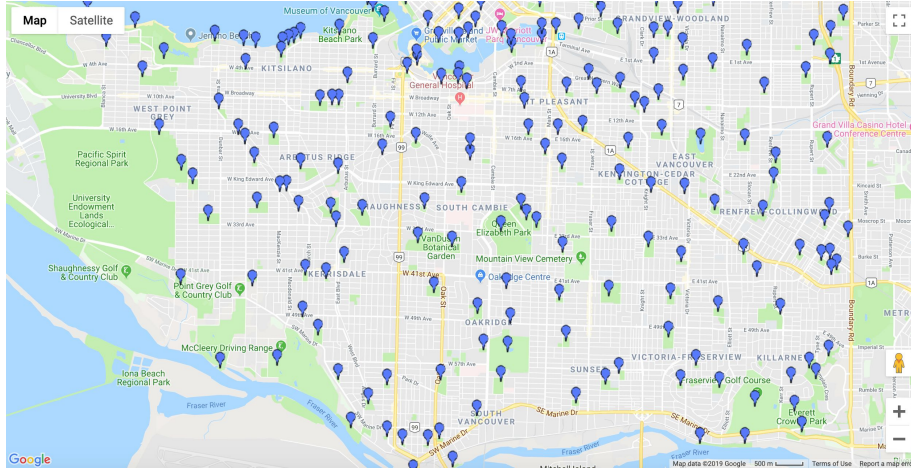
What's your favorite source of data?

Intro to scraping

Pandas

Information from data...

https://vanmapp1.vancouver.ca/gmaps/covmap.htm?map=parks_areas

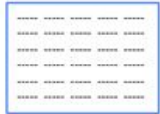


<https://www.geocaching.com/play/map?lat=49.23710338135142&lng=-123.1318473815918&zoom=13&asc=true&sort=distance&st=vancouver%2C+British+Columbia>

103 to 203



CPSC103++ Data Flow:



Diverse data
sources

```
13 if pivot != right:
14     array[right], array[pivot] = array[pivot], array[right]
15     return partition_right(array, left,
16                             right, pivot)
17
18 def partition_right(array, left, right,
19                     pivot = array[right]):
20     mid = left
```

```
13 if pivot != right:
14     array[right], array[pivot] = array[pivot], array[right]
15     return partition_right(array, left,
16                             right, pivot)
17
18 def partition_right(array, left, right,
19                     pivot = array[right]):
20     mid = left
```

data synthesis

```
13 if pivot != right:
14     array[right], array[pivot] = array[pivot], array[right]
15     return partition_right(array, left,
16                             right, pivot)
17
18 def partition_right(array, left, right,
19                     pivot = array[right]):
20     mid = left
```

Python problem
solution using
simple data types
and elementary list
iteration.



Matplotlib bar or
line graph or other
summative output
illustrating results
of computation.



```
13 if pivot != right:
14     array[right], array[pivot] = array[pivot], array[right]
15     return partition_right(array, left,
16                             right, pivot)
17
18 def partition_right(array, left, right,
19                     pivot = array[right]):
20     mid = left
```

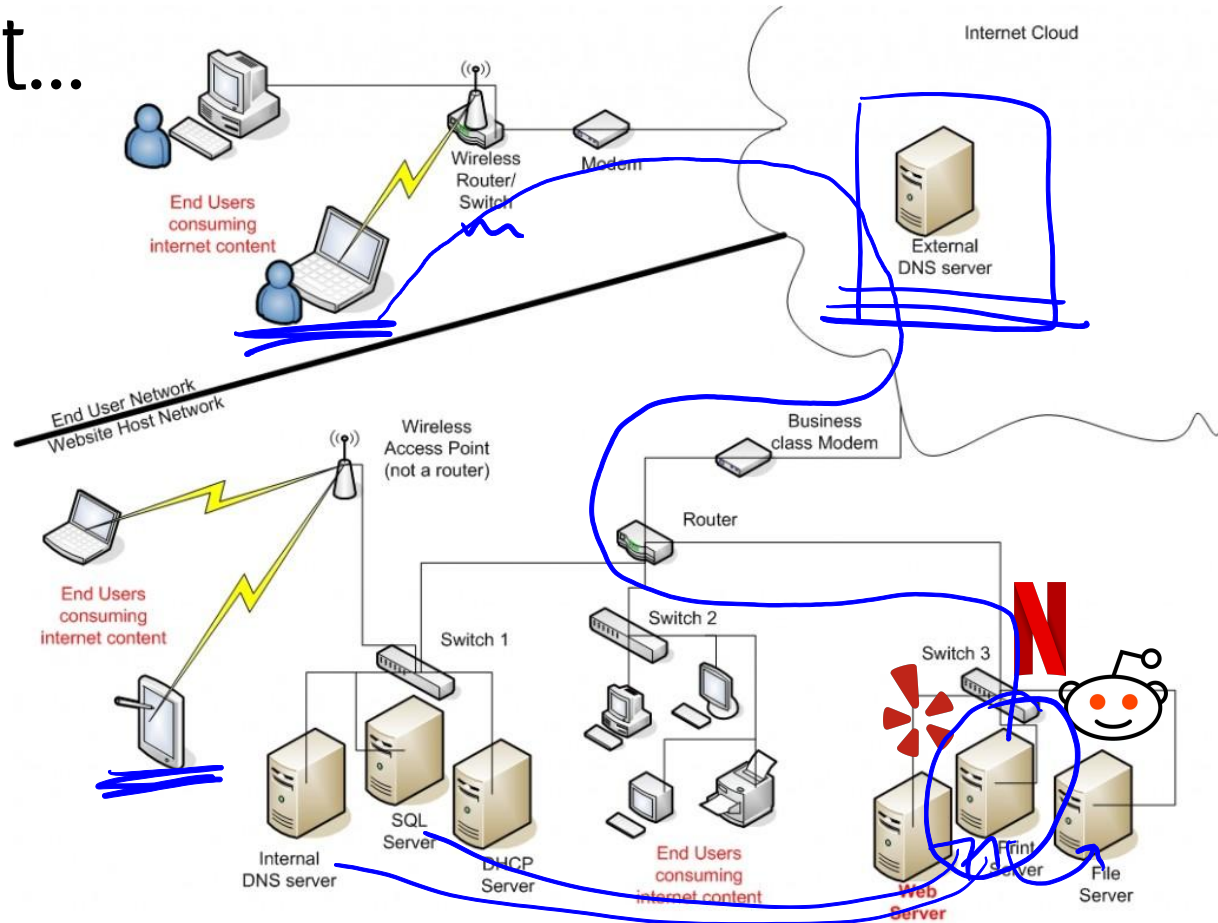
Analysis and data
assembly

mpg	hp	wt
19.200	123.000	3.325
20.091	146.000	3.217
1.065	12.120	0.173
2.173	24.720	0.351
30.124	4706.007	0.952
6.027	68.563	0.978
0.200	0.407	0.304



Diverse outputs

The internet...



Billboard Hot 100...

Navigate to <https://www.billboard.com/charts/hot-100>

What happens to the URL if you load a past week? _____

What happens to the page if you substitute a different date into the URL?

Write one question you would like to ask of this data: _____

Anatomy of html...

```
<!DOCTYPE html>
```

tags in red

```
<html><head><title>The Dormouse's story</title></head>
```

```
<body><p class="title"><b>The Dormouse's story</b></p>
```

```
<p class="story">Once upon a time there were two little sisters.
```

```
Their names were <a href="http://example.com/elsie" class="sister"
```

```
id="link1">Elsie</a>, and <a href="http://example.com/lacie"
```

```
class="sister" id="link2">Lacie</a> and they lived at the bottom of a  
well.</p>
```

```
</body>
```

```
</html>
```

value of
an attribute.

attributes
attrs

Billboard Hot 100... page source

```
<div class="chart-list-item piano-content-overlay__gated-item" data-rank="49" data-artist="Taylor Swift" data-title="Lover" data-has-content="true"> <div
class="chart-list-item__first-row chart-list-item__cursor-pointer"> <div class="chart-list-item__position chart-list-item__position--centered"> <div
class="chart-list-item__rank "> 49 </div> <div
class="chart-list-item__award"> </div> </div> <div
class="chart-list-item__image-wrapper"> <div class="chart-list-item__trend-icon"> </div>
```

```
</div>
```

```
<div class="chart-list-item__text-wrapper"> <div class="chart-list-item__text "> <div class="chart-list-item__title">
<span class="chart-list-item__title-text">
Lover
```



```
Lover
</span> </div>
<div class="chart-list-item__artist">
<a href="/music/taylor-swift">
Taylor Swift
</a>
</div>
<div class="chart-list-item__lyrics">
<a href="https://www.billboard.com/articles/news/lyrics/7950218/ready-for-it-taylor-swift-lyrics">
<span class="hidden-mobile show-expanded-mobile-inline">Song </span>Lyrics
</a></div></div></div>
<div class="chart-list-item__chevron-wrapper"><i class="fa fa-chevron-down"></i></div></div>
<div class="chart-list-item__extra-info"><div class="chart-list-item__extra-info-shadow"></div>
<div class="chart-list-item__stats">
<div class="chart-list-item__stats-cell basic-user chart-list-item__stats-cell--first-cell"> <div class="chart-list-item__stats-icon fa fa-arrow-up
fa-rotate-45"></div>
<div class="chart-list-item__last-week">23</div>
LAST WEEK </div>
<div class="chart-list-item__stats-cell basic-user"> <div class="chart-list-item__stats-icon fa fa-arrow-up fa-rotate-45"></div>
<div class="chart-list-item__last-week">10</div>
TWO WEEKS AGO</div>
<div class="chart-list-item__stats-cell basic-user"> <div class="chart-list-item__stats-icon fa fa-line-chart"></div>
<div class="chart-list-item__weeks-at-one">10</div>
PEAK POSITION </div>
<div class="chart-list-item__stats-cell basic-user chart-list-item__stats-cell--no-border-right"><div class="chart-list-item__stats-icon fa fa-clock-o"></div>
<div class="chart-list-item__weeks-on-chart">4</div>
WEEKS ON CHART</div></div></div></div>
```

Beautiful Soup

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

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
python:

```
html = simple_get("https://www.billboard.com/charts/hot-100" + '/' + date)
```

```
mydivs = html.findAll("div", {"class": "chart-list-item"}) // all the data is here!!
```

```
for div in mydivs:
```

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

attr.
string that is
data-title's
value

→ class Song that we made.

Pandas and data frames

```
import pandas
```

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

Instead of saying **pandas.read_csv('file.csv')**

we can say

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.

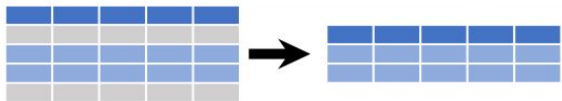
bill_week.csv

saved ▼

```
1 |,week,title,artist,rank,last_week,peak_pos,weeks_on_chart
2 0,2019-09-21,Truth Hurts,Lizzo,1,1,1,19
3 1,2019-09-21,Senorita,Shawn Mendes & Camila Cabello,2,2,1,12
4 2,2019-09-21,Goodbyes,Post Malone Featuring Young Thug,3,10,3,10
5 3,2019-09-21,Circles,Post Malone,4,7,4,2
6 4,2019-09-21,Bad Guy,Billie Eilish,5,3,1,24
7 5,2019-09-21,Ransom,Lil Tecca,6,4,4,15
8 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.

df.nlargest(10, 'last_week')

Returns top 10 hits from last week.

df[df['weeks_on_chart'] > 10]

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

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

Adding a column

```
df['gradient'] = df['last_week'] - df['rank']
```

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

```
df[ df['weeks_on_chart'] > 10 ]
```

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

POTD #6 Tue

<https://github.students.cs.ubc.ca/cpsc203-2019w-t1/potd06>

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:

TLACS Ch 17

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

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