



Visualization (II)

CSE 160

Summer 2026

Questions?



Announcements

- [Midterm Scores](#) available on Gradescope
 - Accepting Regrade Requests until Friday, July 31 at 11:59pm
- [Programming Practice 5](#) due Sunday, August 2 at 11:59 PM
- [Homework Assignment 4](#) due Monday, August 3 at 11:59 PM
- [Resubmission Cycle 2](#) due tonight July 29th at 11:59pm
 - Homework 1 and Homework 2 are eligible
 - Fill out the [Google Form!](#)

Things You Might Come Across

How would you implement the following in Python?

- Read in a CSV
- Draw a graph
- Get a random number
- Use a machine learning algorithm

Intro to Visualization (what's the big deal?)

```

ms.csv | spreadske_scores.csv X
spreadske_scores.csv > data
schedule_date,schedule_season,schedule_week,schedule_playoff,team_home,score_home
1/4/2026,2025,18,FALSE,Cincinnati Bengals,18,20,Cleveland Browns,CIN,-8,45.5,Payco
1/4/2026,2025,18,FALSE,Denver Broncos,19,3,Los Angeles Chargers,DEN,-14.5,37.5,Emp
1/4/2026,2025,18,FALSE,Houston Texans,38,30,Indianapolis Colts,HOU,-11,39,NRG Stadiu
1/4/2026,2025,18,FALSE,Jacksonville Jaguars,41,7,Tennessee Titans,JAX,-13.5,47.5,
1/4/2026,2025,18,FALSE,Las Vegas Raiders,14,12,Kansas City Chiefs,KC,-4.5,36,Alleg
1/4/2026,2025,18,FALSE,Los Angeles Rams,37,20,Arizona Cardinals,LAR,-13.5,47.5,Sol
1/4/2026,2025,18,FALSE,Minnesota Vikings,16,3,Green Bay Packers,MIN,-10.5,36.5,U.S.
1/4/2026,2025,18,FALSE,New England Patriots,38,10,Miami Dolphins,NE,-11.5,45.5,Gi
1/4/2026,2025,18,FALSE,New York Giants,34,17,Dallas Cowboys,DAL,-3.5,49,MetLife S
1/4/2026,2025,18,FALSE,Philadelphia Eagles,17,24,Washington Commanders,PHI,-4.5,39
1/4/2026,2025,18,FALSE,Pittsburgh Steelers,26,24,Baltimore Ravens,BAL,-3.5,40.5,A
1/10/2026,2025,Wildcard,TRUE,Carolina Panthers,31,34,Los Angeles Rams,LAR,-10.5,44
1/10/2026,2025,Wildcard,TRUE,Chicago Bears,31,27,Green Bay Packers,GB,-1,44.5,Sol
1/11/2026,2025,Wildcard,TRUE,Jacksonville Jaguars,24,27,Buffalo Bills,JAX,-1.5,51
1/11/2026,2025,Wildcard,TRUE,New England Patriots,16,3,Los Angeles Chargers,NE,-3
1/11/2026,2025,Wildcard,TRUE,Philadelphia Eagles,19,23,San Francisco 49ers,PHI,-6
1/12/2026,2025,Wildcard,TRUE,Pittsburgh Steelers,6,30,Houston Texans,HOU,-3,38.5,
1/17/2026,2025,Division,TRUE,Denver Broncos,33,30,Buffalo Bills,DEN,-1.5,46,Empow
1/17/2026,2025,Division,TRUE,Seattle Seahawks,41,6,San Francisco 49ers,SEA,-7.5,43
1/18/2026,2025,Division,TRUE,Chicago Bears,17,20,Los Angeles Rams,LAR,-3.5,48.5,S
1/18/2026,2025,Division,TRUE,New England Patriots,28,16,Houston Texans,NE,-3.5,41
1/25/2026,2025,Conference,TRUE,Denver Broncos,7,10,New England Patriots,NE,-4,43,
1/25/2026,2025,Conference,TRUE,Seattle Seahawks,31,27,Los Angeles Rams,SEA,-2.5,4
2/8/2026,2025,Superbowl,TRUE,New England Patriots,13,29,Seattle Seahawks,SEA,-5,4

```

Source: <https://www.kaggle.com/datasets/tobycrabtree/nfl-scores-and-betting-data?resource=download>

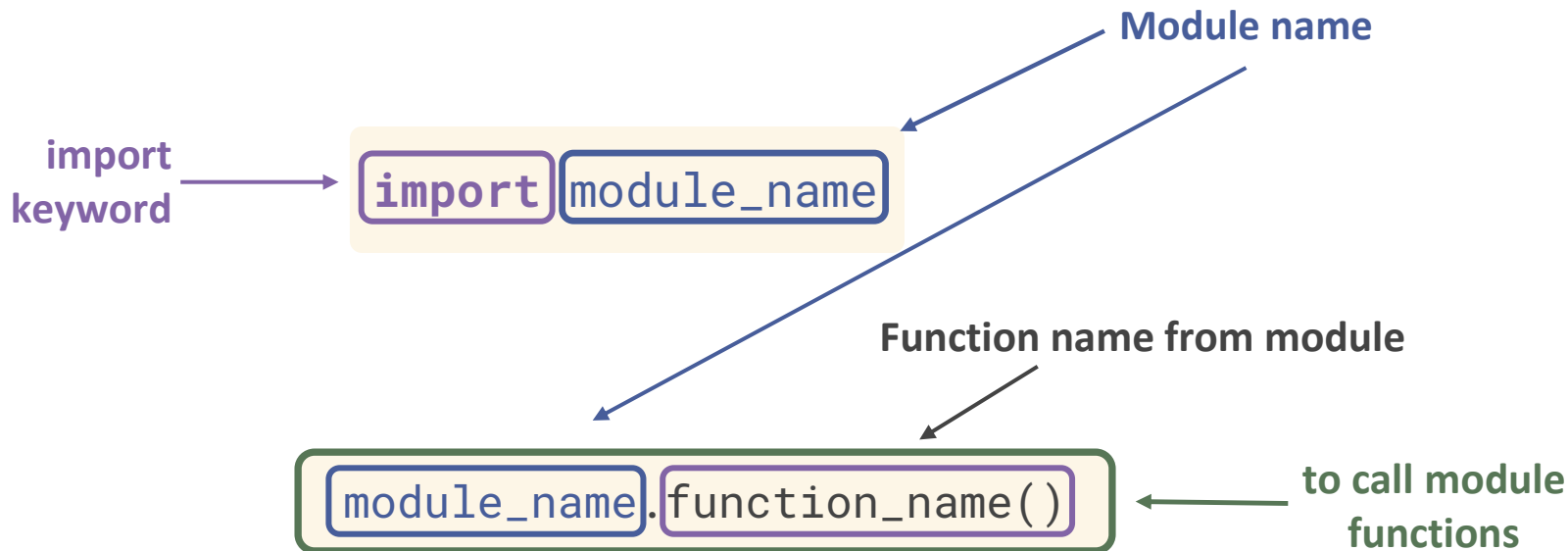
```

~/Downloads/archive/nfl_teams.csv
1 Month,Questions
2 "2008-07-01 00:00:00","4"
3 "2008-08-01 00:00:00","3748"
4 "2008-09-01 00:00:00","14033"
5 "2008-10-01 00:00:00","14578"
6 "2008-11-01 00:00:00","12715"
7 "2008-12-01 00:00:00","12080"
8 "2009-01-01 00:00:00","15833"
9 "2009-02-01 00:00:00","17580"
10 "2009-03-01 00:00:00","20482"
11 "2009-04-01 00:00:00","21336"
12 "2009-05-01 00:00:00","25812"
13 "2009-06-01 00:00:00","28325"
14 "2009-07-01 00:00:00","32481"
15 "2009-08-01 00:00:00","32773"
16 "2009-09-01 00:00:00","33053"
17 "2009-10-01 00:00:00","36329"
18 "2009-11-01 00:00:00","38514"
19 "2009-12-01 00:00:00","37792"
20 "2010-01-01 00:00:00","44936"
21 "2010-02-01 00:00:00","44810"
22 "2010-03-01 00:00:00","52234"
23 "2010-04-01 00:00:00","50083"
24 "2010-05-01 00:00:00","51996"
25 "2010-06-01 00:00:00","55747"
26 "2010-07-01 00:00:00","60846"
27 "2010-08-01 00:00:00","63657"
28 "2010-09-01 00:00:00","63227"

```

Source:
<https://data.stackexchange.com/stackoverflow/query/1926661#result>
 Sets

Anatomy of Calling a Function from a Module

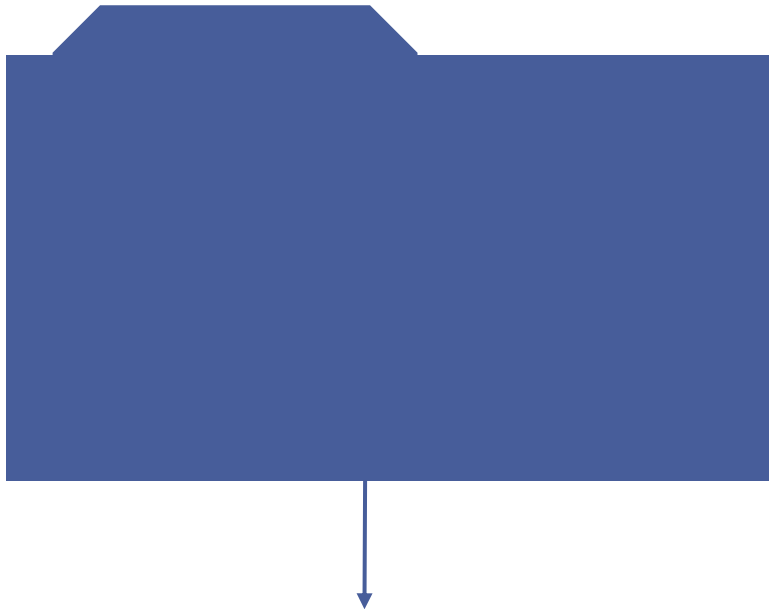


Matplotlib

- Matplotlib is a Python package with functions that can be used to produce graphs
- Matplotlib works well when you have two lists (of the same size)
 - List 1 acts as the x (independent) variable
 - List 2 acts as the y (dependent) variable
- To use the functions we have to import the module!

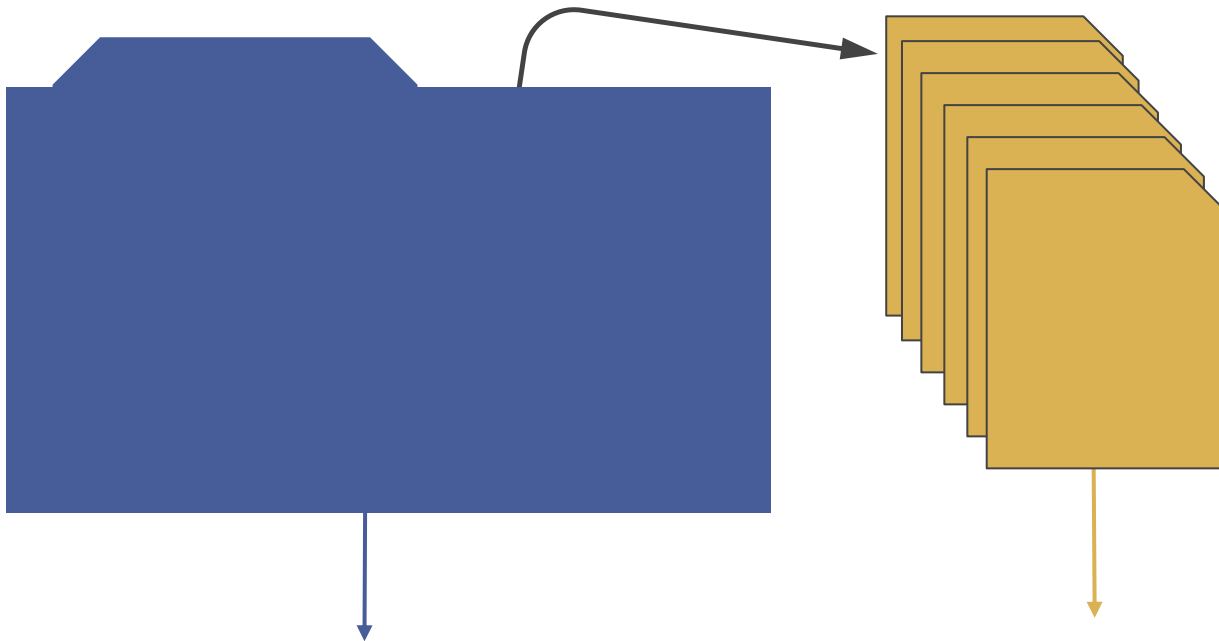


Packages vs Modules vs Functions



Package: a folder that contains multiple modules

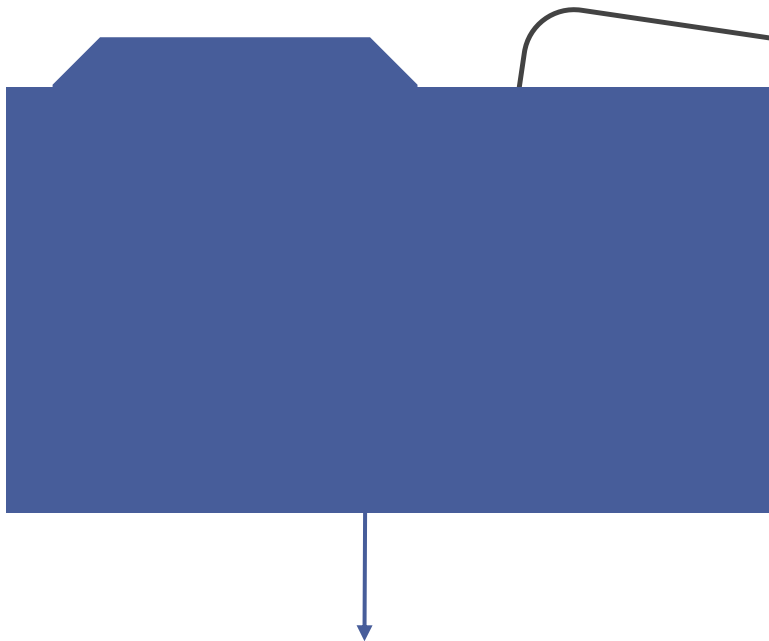
Packages vs Modules vs Functions



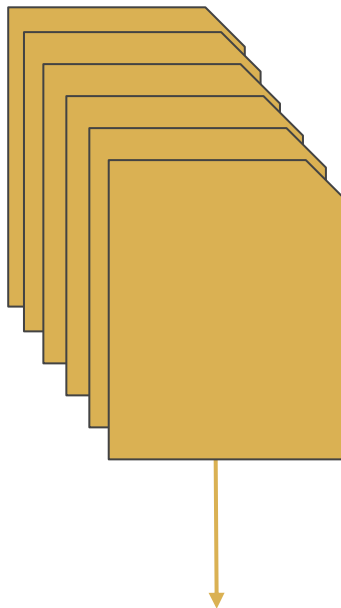
Package: a folder that contains multiple modules

Module: a .py file

Packages vs Modules vs Functions



Package: a folder that contains multiple modules



Module: a .py file



Function: compute calculations

Matplotlib Functions

```
plt.plot(x_data, y_data)
```

- Pass in two lists of data, with the x values before the y values
- Values at the same index are corresponding to one another
- Connects points in the order they appear in your list - so x_data should usually be in ascending order

```
plt.title("Title")
```

- Pass in a string to title the graph

```
plt.xlabel("X Label")
```

- Pass in a string to label the x axis

```
plt.ylabel("Y Label")
```

- Pass in a string to label the y axis

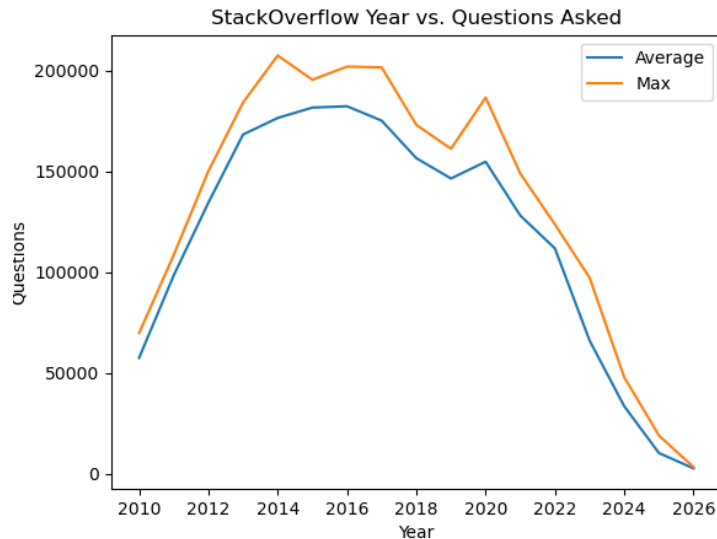
Matplotlib Functions - Data Labels & Legends

```
plt.plot(x_data, y_data,  
label = "Your Label")
```

- Pass in two lists and a string label

```
plt.legend()
```

- Adds a legend to your graph; you can also modify its location



Matplotlib Functions

```
plt.savefig("your_filename")
```

- Pass in a string for the filename
- Will save the figures into a .png file on your computer

```
plt.clf()
```

- Will reset your plot so you are not unintentionally adding new data points/lines each time you run your code

Matplotlib Functions

```
plt.axhline(val)  
plt.axvline(val)
```

- Draws axis lines at the specified value
- `.axhline()` is for a horizontal line
- `.axvline()` is for a vertical line

```
plt.xlim(low, high)  
plt.ylim(low, high)
```

- These functions limit the range of the axes
- Both numbers are inclusive (unlike `range`)

NFL

Example: NFL

```
schedule_season,schedule_date,team_home.....  
2018,2019-01-06,Chicago Bears.....  
2022,2023-01-08,Atlanta Falcons.....  
1981,1981-10-11,San Francisco 49ers.....  
2014,2014-10-12,Oakland Raiders.....  
2012,2012-10-14,Arizona Cardinals.....
```

games_data.csv

```
def load_games(csv_file):  
    ...  
    with open(csv...  
        ...  
        ...
```

parse_data.py

```
def get_avgs_and_seasons(games):  
    ...  
    season_totals = {}  
    season_counts = {}  
    ...
```

plot_nfl.py

Example 2: NFL

```
schedule_season,schedule_date,team_home.....  
2018,2019-01-06,Chicago Bears.....  
2022,2023-01-08,Atlanta Falcons.....  
1981,1981-10-11,San Francisco 49ers.....  
2014,2014-10-12,Oakland Raiders.....  
2012,2012-10-14,Arizona Cardinals.....
```

games_data.csv

```
def load_games(csv_file):  
    ...  
    with open(csv...  
        ...  
        ...
```

parse_data.py

CSV file containing the
data

```
def get_avgs_and_seasons(games) :  
    ...  
    season_totals = {}  
    season_counts = {}  
    ...
```

plot_nfl.py

Example 2: NFL

```
schedule_season,schedule_date,team_home.....  
2018,2019-01-06,Chicago Bears.....  
2022,2023-01-08,Atlanta Falcons.....  
1981,1981-10-11,San Francisco 49ers.....  
2014,2014-10-12,Oakland Raiders.....  
2012,2012-10-14,Arizona Cardinals.....
```

games_data.csv

```
def load_games(csv_file):  
    ...  
    with open(csv...  
        ...  
        ...
```

parse_data.py

```
def get_avgs_and_seasons(games) :  
    ...  
    season_totals = {}  
    season_counts = {}  
    ...
```

plot_nfl.py

Python file that contains a function `load_games`, which parses csv data and returns a dictionary representation of the data

Example 2: NFL

```
schedule_season,schedule_date,team_home.....  
2018,2019-01-06,Chicago Bears.....  
2022,2023-01-08,Atlanta Falcons.....  
1981,1981-10-11,San Francisco 49ers.....  
2014,2014-10-12,Oakland Raiders.....  
2012,2012-10-14,Arizona Cardinals.....
```

games_data.csv

```
def load_games(csv_file):  
    ...  
    with open(csv...  
        ...  
        ...
```

parse_data.py



```
def get_avgs_and_seasons(games):  
    ...  
    season_totals = {}  
    season_counts = {}  
    ...
```

plot_nfl.py

A file containing a function that parses dictionary data into two lists we can use to graph! Also the place we'll be putting graphing code

Example: NFL

```
schedule_season,schedule_date...  
2018,2019-01-06,Chicago Bears.....  
2022,2023-01-08,Atlanta Falcons.....  
1981,1981-10-11,San Francisco 49ers.....  
2014,2014-10-12,Oakland Raiders.....  
2012,2012-10-14,Arizona Cardinals.....
```

games_data.csv

```
def load_games(csv_file):  
    ...  
    with open(csv...  
        ...  
        ...
```

parse_data.py

```
def get_avgs_and_seasons(games):  
    ...  
    season_totals = {}  
    season_counts = {}  
    ...
```

plot_nfl.py

Your Task

Within the **plot_nfl.py** file, write code that calls the previously defined functions and plots the data returned from **get_avgs_and_seasons** on a graph (with a title and axis labels)

Example: NFL - Answers

```
import matplotlib.pyplot as plt
from parse_data import load_games
```

```
games = load_games("games_data.csv")
seasons, avg_totals = get_avgs_and_seasons(games)
```

```
plt.plot(seasons, avg_totals)
```

```
plt.title("Average Total Points per Season")
plt.xlabel("Season")
plt.ylabel("Average Total Points")
plt.savefig("nfl")
plt.clf()
```

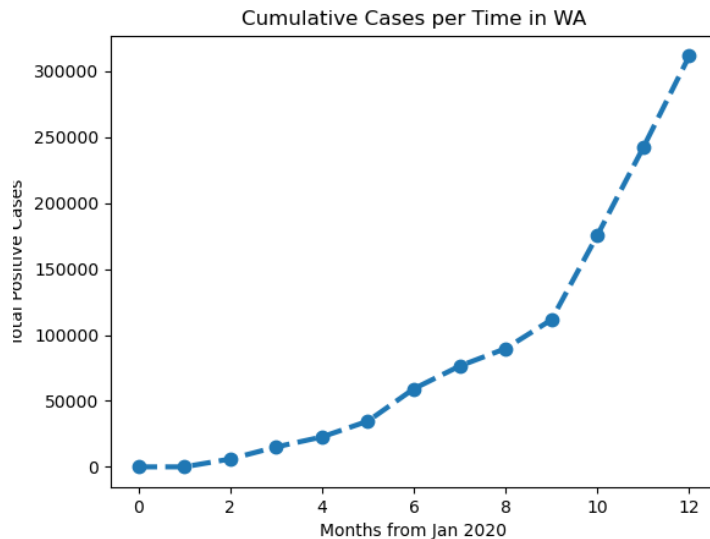
Covid Cases

Practice 1 - Cumulative Cases in WA

Write code that will produce the following graph, a visualization of cumulative positive cases in the state of Washington. Include the correct title and axes titles.

x-axis = "Months from Jan 2020"

y-axis = "Total Positive Cases"



Practice 2 - Percentage of Cases per State

Suppose you are given a csv file called `state_populations.csv` containing state acronyms and their total reported population in 2020 according to the U.S. census. Write a program that reads in this file and plots the percentage of positive cases each state has with respect to their total population across the first 12 months. Your program should also save the generated figure.

Example

California reported 7429 cases in month 0 and has a population of 39,538,223 so you should plot the value $7429/39,538,223 = 0.000187$ (you may round to 4 decimal places using `round(num, 4)` where `num` is the value you want to round).