

CS106A

Juliette Woodrow

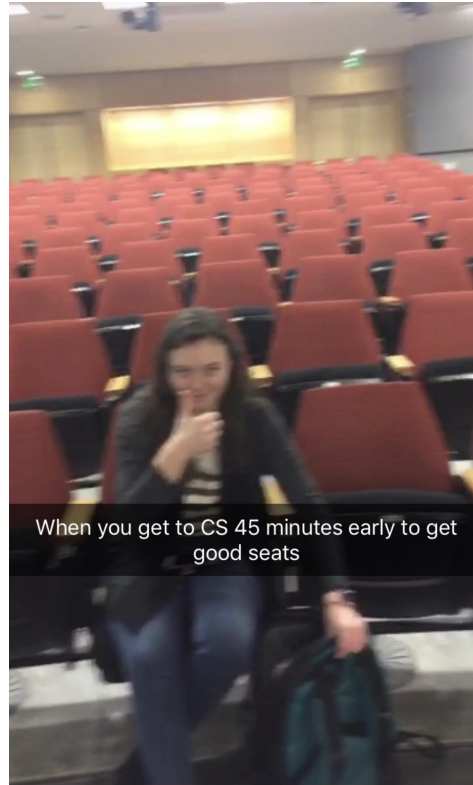
Housekeeping Part 1

- Happy Friday!
- Congratulations on making it through week 9
- Entries for the Graphics/Image contest are due **tonight** at 11:59pm

Housekeeping Part 2

- Final exam is Friday, June 3rd 8:30-11:30am
- Location of the final exam is by first letter of your last name:
 - A-H Bishop Auditorium (in Lathrop)
 - I-Z CEMEX Auditorium (in Knight Management Center)
- Final exam covers the whole class
- It is open book/note but only printed materials as you cannot access the internet or files on your computer (similar to the midterm)
- You will use BlueBook software to take the final exam
- A practice final exam and solutions will be posted on the website later tonight
- OAE students should receive emails with exam logistics by class time on Weds

A little bit about me...



Let's Jump Right In

Overview

1. Advanced (cooler) way to make lists :)
2. Tools for developing and analyzing data
3. Answer questions about data with your 106a skills

List Comprehensions

Problem: getting a list of squares

Problem: getting a list of squares

- Imagine you have a list of numbers, and you want a list of those same numbers squared

Problem: getting a list of squares

- Imagine you have a list of numbers, and you want a list of those same numbers squared

[4, 6, 7, 8] → [16, 36, 49, 64]

Problem: getting a list of squares

- Imagine you have a list of numbers, and you want a list of those same numbers squared

`[4, 6, 7, 8] → [16, 36, 49, 64]`

- How would you produce this output list?

Problem: getting a list of squares - Attempt #1

[4, 6, 7, 8] → [16, 36, 49, 64]

Problem: getting a list of squares - Attempt #1

[4, 6, 7, 8] → [16, 36, 49, 64]

```
def get_squared(num_lst):  
    squares = []  
    for num in num_lst:  
        squares.append(num**2)  
    return squares
```

Problem: getting a list of squares - Attempt #2

Problem: getting a list of squares - Attempt #2

```
num_lst = [4, 6, 7, 8]
```

Problem: getting a list of squares - Attempt #2

```
num_lst = [4, 6, 7, 8]
```

```
squared_lst = [num ** 2 for num in num_lst]
```


Problem: getting a list of squares - Attempt #2

```
num_lst = [4, 6, 7, 8]
```

```
squared_lst = [num ** 2 for num in num_lst]
```



this is a list comprehension!

List Comprehensions

```
[num ** 2 for num in num_lst]
```

List Comprehensions

```
[num ** 2 for num in num_lst]
```

- **Definition:** one way to make a new list based on the values of an existing list

List Comprehensions

```
[num ** 2 for num in num_lst]
```

- **Definition:** one way to make a new list based on the values of an existing list
- **Three Key Parts:**

List Comprehensions

```
---  
[num ** 2 for num in num_lst]
```



expression

- **Definition:** one way to make a new list based on the values of an existing list
- **Three Key Parts:**
 - Expression

List Comprehensions

```
num ** 2 for num in num_lst]
```

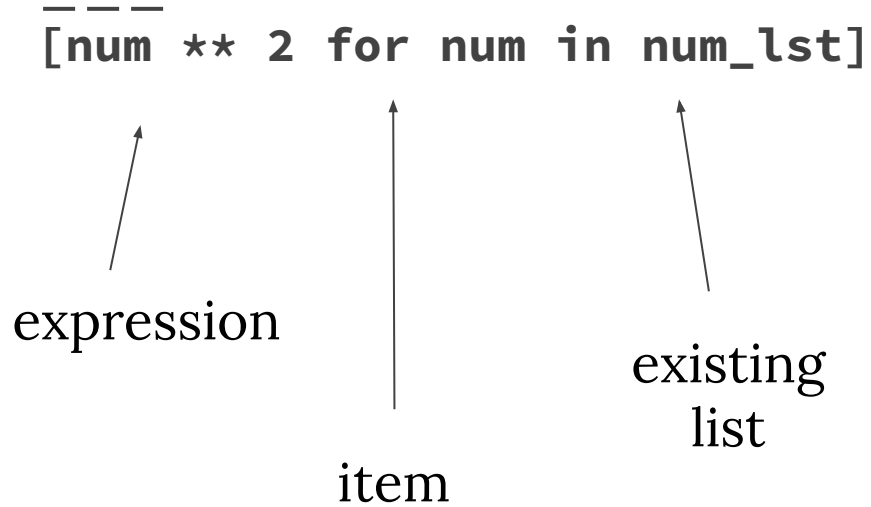
expression



item

- **Definition:** one way to make a new list based on the values of an existing list
- **Three Key Parts:**
 - Expression
 - Item

List Comprehensions



- **Definition:** one way to make a new list based on the values of an existing list
- **Three Key Parts:**
 - Expression
 - Item From Existing List
 - Existing List

Hey, we already know some of that syntax!

```
     
[num ** 2 for num in num_lst]
```


Hey, we already know some of that syntax!

```
— — —  
[num ** 2 for num in num_lst]
```

- [] → that makes it a list

Hey, we already know some of that syntax!

```
___  
[num ** 2 for num in num_lst]
```

- [] → that makes it a list
- for num in num_list → that's just a for each loop

Hey, we already know some of that syntax!

```
---  
[num ** 2 for num in num_lst]
```

- [] → that makes it a list
- **for num in num_list** → that's just a for each loop
- **n**2** → This is how we square a number

Let's try it out!

Let's try it out!

- You have a list of strings with random casing and you want a list of strings that are all lowercase

Let's try it out!

- You have a list of strings with random casing and you want a list of strings that are all lowercase

["Hi", "mOm", "aNd", "DAD"] → ["hi", "mom", "and", "dad"]

Let's try it out!

- You have a list of strings with random casing and you want a list of strings that are all lowercase

`["Hi", "mOm", "aNd", "DAD"]` → `["hi", "mom", "and", "dad"]`

- How can we use a list comprehension to do this?

Problem: getting a list of lowercase strings

```
random_case = ["Hi", "mOm", "aNd", "DAD"]
```


Problem: getting a list of lowercase strings

```
random_case = ["Hi", "mOm", "aNd", "DAD"]
```

```
all_lower = [s.lower() for s in random_case]
```

```
print(all_lower)
```

```
# would print ["hi", "mom", "and", "dad"]
```

Problem: converting temperature to fahrenheit

Problem: converting temperature to fahrenheit

- List of temperatures in degrees celsius
france_temps_c = [13, 14, 15, 16, 8, 9, 12]

Problem: converting temperature to fahrenheit

- List of temperatures in degrees celsius
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
- Want a list of temperatures in degrees fahrenheit

Problem: converting temperature to fahrenheit

Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

```
°C(9/5) + 32 = °F
```

Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

$^{\circ}\text{C} \times (9/5) + 32 = ^{\circ}\text{F}$

```
france_temps_f = [ ]
```

Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

$^{\circ}\text{C} \times (9/5) + 32 = ^{\circ}\text{F}$

```
france_temps_f = [ for t in france_temps_c]
```


Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

$^{\circ}\text{C} \times (9/5) + 32 = ^{\circ}\text{F}$

```
france_temps_f = [t*(9/5) + 32 for t in france_temps_c]
```

Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

$^{\circ}\text{C} \times (9/5) + 32 = ^{\circ}\text{F}$

```
france_temps_f = [t*(9/5) + 32 for t in france_temps_c]
```

```
print(france_temps_f)
```

```
# would print [55.4, 57.2, 59.0, 46.4, 48.2, 53.6, 46.4]
```

Problem: converting temperature to fahrenheit

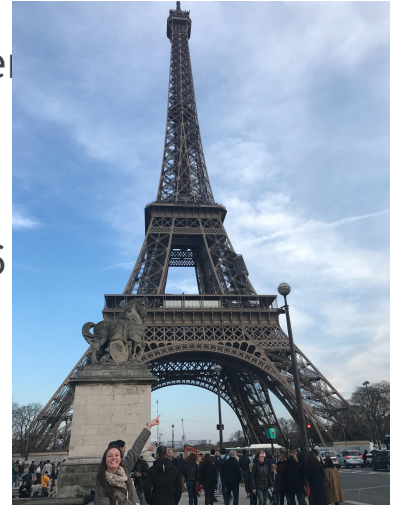
```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

```
°C(9/5) + 32 = °F
```

```
france_temps_f = [t*(9/5) + 32 for t in france_temps_c]
```

```
print(france_temps_f)
```

```
# would print [55.4, 57.2, 59.0, 60.8, 46.4, 48.2, 53.6]
```



Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

```
°C(9/5) + 32 = °F
```

```
france_temps_f = [t*(9/5) + 32 for t in france_temps_c]
```

```
print(france_temps_f)
```

```
# would print [55.4, 57.2, 59.0, 60.8, 46.4, 48.2, 53.6]
```



Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

$^{\circ}\text{C} \times (9/5) + 32 = ^{\circ}\text{F}$

```
france_temps_f = [t*(9/5) + 32 for t in france_temps_c]
```

Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

$^{\circ}\text{C} \times (9/5) + 32 = ^{\circ}\text{F}$

```
france_temps_f = [t*(9/5) + 32 for t in france_temps_c]
```

- Can we decompose this?

Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

$^{\circ}\text{C} \times (9/5) + 32 = ^{\circ}\text{F}$

```
france_temps_f = [t*(9/5) + 32 for t in france_temps_c]
```

- Can we decompose this? Yes !!

Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

$^{\circ}\text{C} \times (9/5) + 32 = ^{\circ}\text{F}$

```
france_temps_f = [t*(9/5) + 32 for t in france_temps_c]
```

- Can we decompose this? Yes !!

```
def make_fahrenheit(c):
```

```
    return c * (9/5) + 32
```


Problem: converting temperature to fahrenheit

```
france_temps_c = [13, 14, 15, 16, 8, 9, 12]
```

$^{\circ}\text{C} \times (9/5) + 32 = ^{\circ}\text{F}$

```
france_temps_f = [make_fahrenheit(t) for t in france_temps_c]
```

- Can we decompose this? Yes !!

```
def make_fahrenheit(c):
```

```
    return c * (9/5) + 32
```

Now you try...

Now you try...

- Given a list of numbers called `nums`, make a new list that has all of the numbers in `nums` with 10 added to each one

```
nums = [3, 5, 7, 9, 11, 13]
```

Now you try...

- Given a list of numbers called `nums`, make a new list that has all of the numbers in `nums` with 10 added to each one

```
nums = [3, 5, 7, 9, 11, 13]
```

```
new_nums = [10 + num for num in nums]
```

Now you try...

- Given a list of numbers called `nums`, make a new list that has the absolute value of `num - 3` for each `num`

```
nums = [3, 5, 7, 9, 11, 13]
```

Now you try...

- Given a list of numbers called `nums`, make a new list that has the absolute value of `num - 3` for each `num`

```
nums = [3, 5, 7, 9, 11, 13]
```

```
new_nums = [abs(num - 3) for num in nums]
```

Now you try...

- Given a list of strings representing numbers, create a new list of the float version of each string

```
str_nums = ['3.14', '1.59', '2.65', '35.8']
```

Now you try...

- Given a list of strings representing numbers, create a new list of the float version of each string

```
str_nums = ['3.14', '1.59', '2.65', '35.8']
```

```
float_nums = [float(str) for str in str_nums]
```



Why List Comprehensions?

- They make me feel cool 😎
- They are more concise
- They are *Pythonic*

Why List Comprehensions?

- They make me feel cool 😎
- They are more concise
- They are *Pythonic*

What does it mean to
by *Pythonic*?



What tools do we have to develop and analyze data?

Let's Analyze Some Data

- You have the power and tools to investigate all kinds of datasets

Let's Analyze Some Data

- You have the power and tools to investigate all kinds of datasets
- I found a cool dataset with UFO sightings
- [National UFO Reporting Center](#)



NBC News

Top defense official says government received about 400 reports of UFOs

The report released in June 2021 said that the U.S. government couldn't explain 143 of the 144 cases of unexplained aerial phenomena (UAP)...

1 week ago





NBC News

Top defense official says government received about 400 reports of UFOs

The report released in June 2021 said that the U.S. government couldn't explain 143 of the 144 cases of unexplained aerial phenomena (UAP)...

1 week ago





Top defense official says government received about 400 reports of UFOs

The report released in June 2021 said that the U.S. government couldn't explain 143 of the 144 cases of unexplained aerial phenomena (UAP)...

1 week ago



Congress Holds First U.F.O. Hearing in Half a Century

At House Hearing, Videos of Unexplained Aerial Sightings and a Push for ...
"This time, other U.S. Navy assets also observed unmanned aerial...

1 week ago





Let's look at the data we have

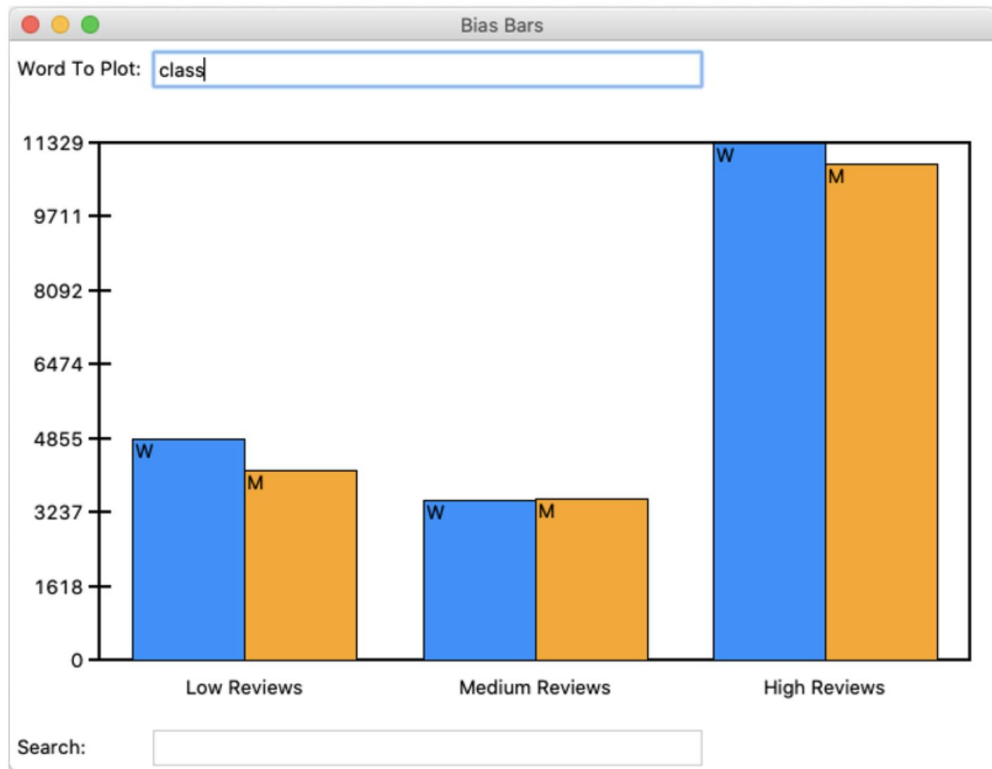
- Dataset of UFO sightings with:
 - reliable city and state_id descriptions
 - unreliable latitude and longitude
 - Datetime of sighting
 - Shape
 - Duration
 - Comments describing the event

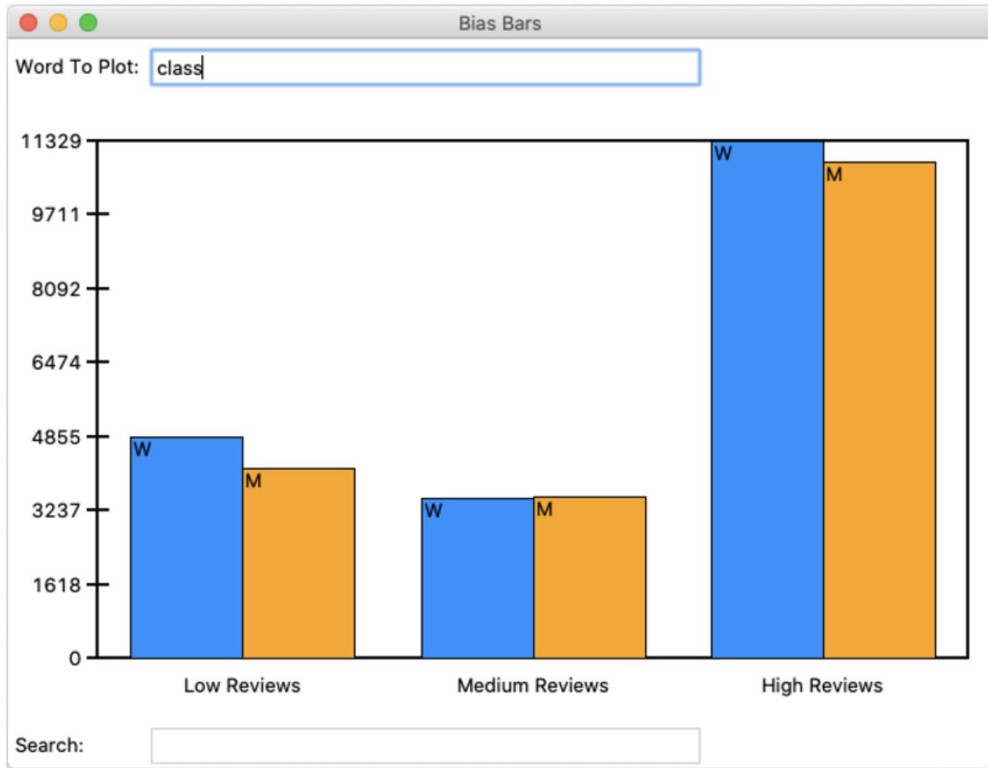
Example lines in the file

`"10/30/2013 21:20,palo alto,ca,us,disk,120,1-2
minutes,"Disk-shaped object with red/blue/green lights
flowing left to right that appeared to stop and become a
horizontal bar of light.",11/11/2013,37.4419444,-122.1419444"`

`"10/31/1974 07:00,rockford,il,us,light,5,5 seconds,"Bright
white light very fast no sound or sonic boom impossible
maneuvers too fast for
aircraft",11/26/2003,42.2711111,-89.0938889"`

How Can We Visualize Data?





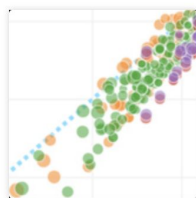
You are all already experts on this!
But...
there is another way

Using Plotly

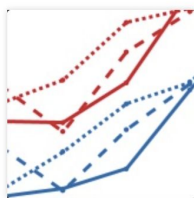
Using Plotly

- An open source library in python to create plots
- [Tons of documentation here. Feel free to explore.](#)

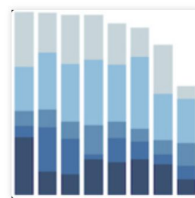
Basic Charts



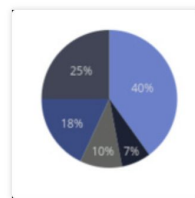
Scatter Plots



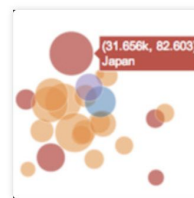
Line Charts



Bar Charts



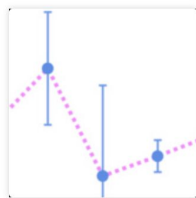
Pie Charts



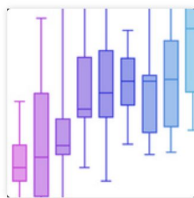
Bubble Charts

[More Basic Charts »](#)

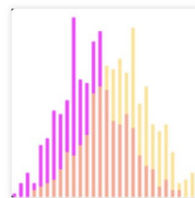
Statistical Charts



Error Bars



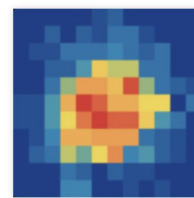
Box Plots



Histograms



Distplots



2D Histograms

[More Statistical Charts »](#)

Using Plotly

- Doesn't come with Python but you can easily install it
- Go to your terminal in pycharm:
- To install
 - **py -m pip install plotly** #(use python3 instead of py on Mac)

Using Plotly

import the module so we can use it

```
import plotly.express as px
```

use a list of x vals and a list of y vals

```
fig = px.scatter(x=[0, 1, 2, 3, 4], y=[0, 1, 4, 9, 16])
```

display the plot

```
fig.show()
```

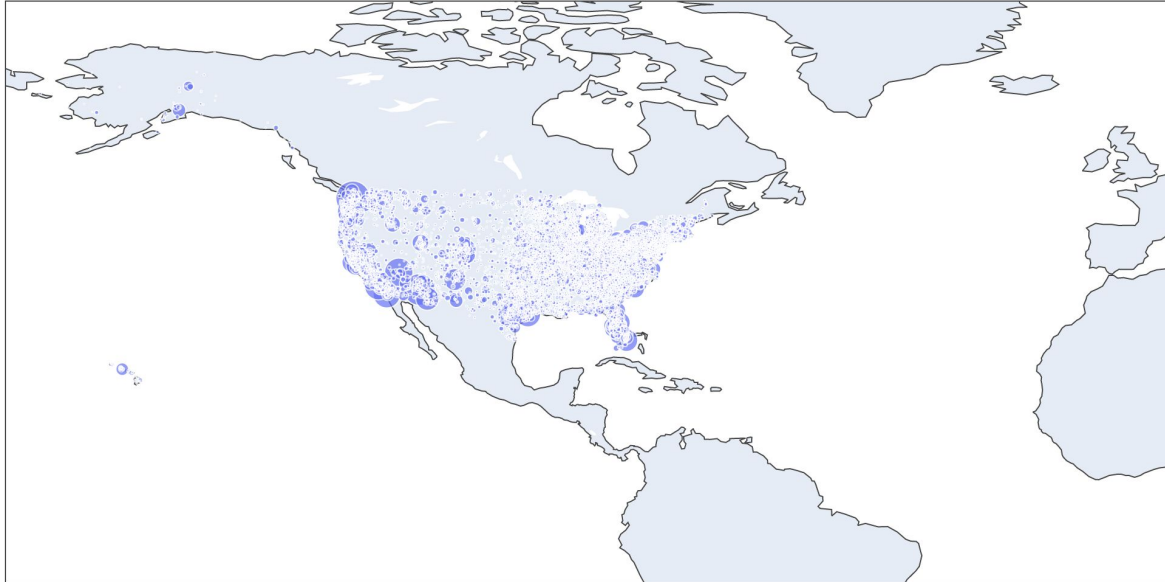
Let's try it out

Using Plotly

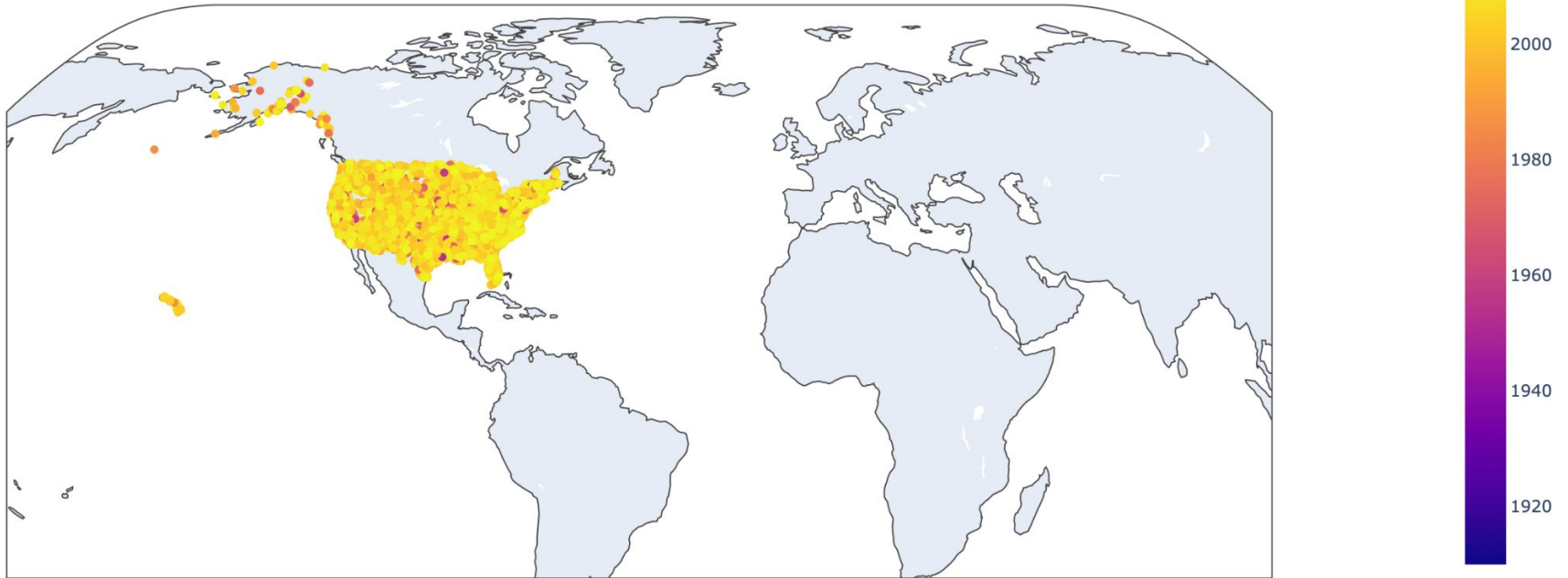
- You can make all kinds of graphs (even interactive ones)
- Basics: scatter, line, bar, pie, bubble
- Advanced: contour plots, heat maps, 3D Charts
- The world is your oyster

Using Plotly Today

- We are going to plot UFO sightings onto a map



Using Plotly Today



Using Plotly Today

- We are going to make a scatter geo plot

```
fig = px.scatter_geo(geo_dict,  
                    lat=geo_dict['latitude'],  
                    lon=geo_dict['longitude'])
```

```
fig.show()
```


Using Plotly Today

- We are going to make a scatter geo plot

You can think of this dictionary as a spreadsheet where the keys are the columns

```
fig = px.scatter_geo(geo_dict,  
                    lat=geo_dict['latitude'],  
                    lon=geo_dict['longitude'])
```

```
fig.show()
```

Using Plotly Today

- We are going to make a scatter geo plot

You can think of this dictionary as a spreadsheet where the keys are the columns

```
fig = px.scatter_geo(geo_dict,  
                    lat=geo_dict['latitude'],  
                    lon=geo_dict['longitude'],  
                    color=geo_dict['year'],  
                    size=geo_dict['num_sightings'])  
  
fig.show()
```

Recall the data we have

- Dataset of UFO sightings with:
 - reliable city and state_id descriptions
 - unreliable latitude and longitude
 - Datetime of sighting
 - Shape
 - Duration
 - Comments describing the event

Another dataset of US Cities to Lat/Long

- Since the lat/long data from the UFO dataset is unreliable, we have another dataset
- This dataset has city, state_id, latitude, longitude, lots more
- One line in the file looks like:

**Palo Alto,Palo Alto,CA,California,06085,Santa
Clara,37.3905,-122.1468,67973,1089,shape,FALSE,TRUE,America/
Los_Angelos,3,94028 94022 94303 94301 94306 94304,1840020333**

Another dataset of US Cities to Lat/Long

- We are going to use this to build a dictionary that looks like:

```
{  
    (city, state_id): (lat, lon)  
}
```

Another dataset of US Cities to Lat/Long

Using this example:

```
Palo Alto,Palo Alto,CA,California,06085,Santa  
Clara,37.3905,-122.1468,67973,1089,shape,FALSE,TRUE,America/  
Los_Angeles,3,94028 94022 94303 94301 94306 94304,1840020333
```

It would look like:

```
{  
    ('Palo Alto', 'CA'): (37.3905, -122.1468)  
}
```

Overview

- Once we have the dict with (city, state_id): (lat, lon),
 - 1) we can read through the UFO sightings
 - 2) parse each line to get the city, state_id, and year
 - 3) plot these sightings

Let's Checkout the Code

