Recap of Unemployment Rates Internal Project

Fall 2022: Michael Yip, Susan Yang

This Semester’s Timeline

Week Plan

Week 1: 09/19 - 09/30

- Introduction to project
- Brief review of Python

Week 2: 10/03 - 10/10

- Introduction to **NumPy, Pandas**

Week 3: 10/10 - 10/14

- Visualization using **Matplotlib**

Week 4: 10/22 - 10/30

- Brief Intro to **Seaborn**
- Basic **API call to retrieve data**

Week 5: 10/31 - 11/7

- **Regression: Linear & Logistic**
- **K-means**

Week 6: 11/8 - 11/15

- **Support Vector Machines**

NO MEETING WEEK

- Enjoy your break!

Week 8: 11/28 - 12/2

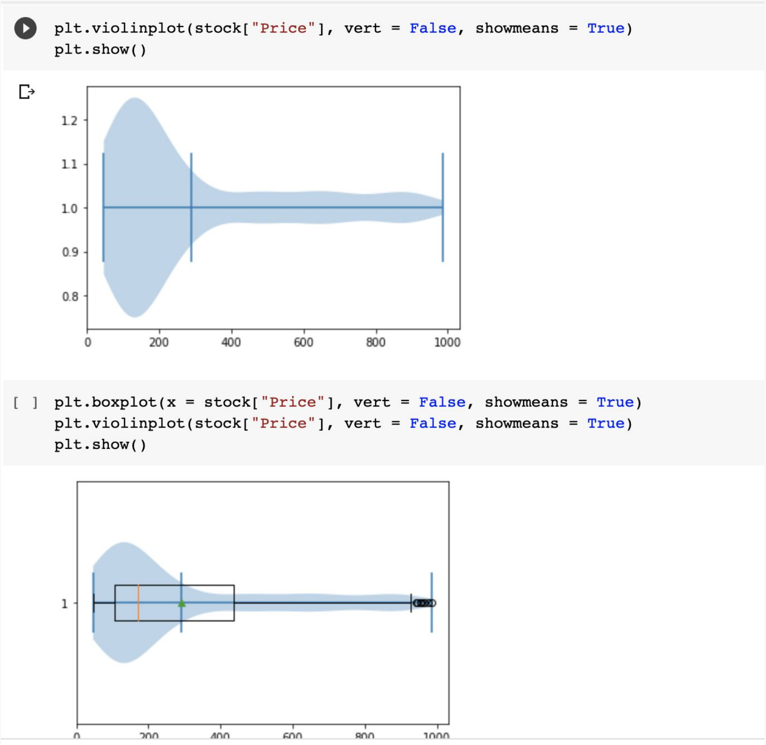
- Design your own final project!
- Prepare for final deliverable

Week 9: **12/2**

- **Final Presentation**

Lab highlights

Week 3 Matplotlib

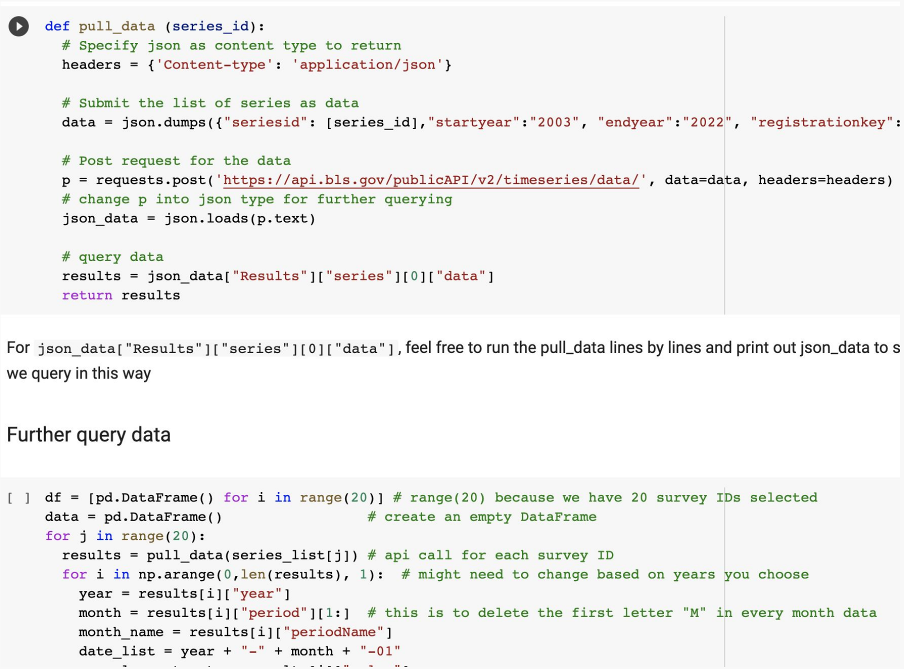


```
plt.violinplot(stock["Price"], vert = False, showmeans = True)
plt.show()
```

```
plt.boxplot(x = stock["Price"], vert = False, showmeans = True)
plt.violinplot(stock["Price"], vert = False, showmeans = True)
plt.show()
```

We added some more methods of **beautifying** the graphs, dealing with **overplotting**, **labels overlapping**, and log&sqrt plot, etc. We also introduced more plots types including **Stack plot and Stem plot**.

Week 4 API calls



```
def pull_data (series_id):
  # Specify json as content type to return
  headers = {'Content-type': 'application/json'}

  # Submit the list of series as data
  data = json.dumps({"seriesid": [series_id],"startyear":"2003", "endyear":"2022", "registrationkey":

  # Post request for the data
  p = requests.post('https://api.bls.gov/publicAPI/v2/timeseries/data/', data=data, headers=headers)
  # change p into json type for further querying
  json_data = json.loads(p.text)

  # query data
  results = json_data["Results"]["series"][0]["data"]
  return results
```

For `json_data["Results"]["series"][0]["data"]`, feel free to run the pull_data lines by lines and print out json_data to s
we query in this way

Further query data

```
df = [pd.DataFrame() for i in range(20)] # range(20) because we have 20 survey IDs selected
data = pd.DataFrame()                    # create an empty DataFrame
for j in range(20):
  results = pull_data(series_list[j]) # api call for each survey ID
  for i in np.arange(0,len(results), 1):  # might need to change based on years you choose
    year = results[i]["year"]
    month = results[i]["period"][1:]  # this is to delete the first letter "M" in every month data
    month_name = results[i]["periodName"]
    date_list = year + "-" + month + "-01"
```

We talked about what **API** is, **why** retrieving data from API, and **how to query** a retrieved **JSON** file. With sample code for US Bureau of Labor Statistics and another example for Youtube API provided.

Week 6 Logistic Regression

Properties of a Logistic Function

The logistic function is a type of **sigmoid**, a

ı properties.

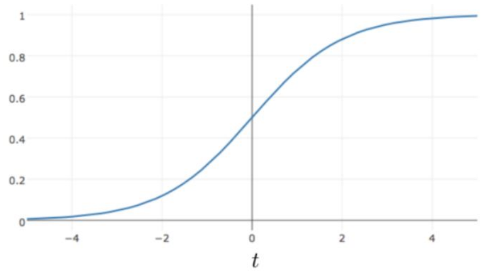
n 0 and 1, nc

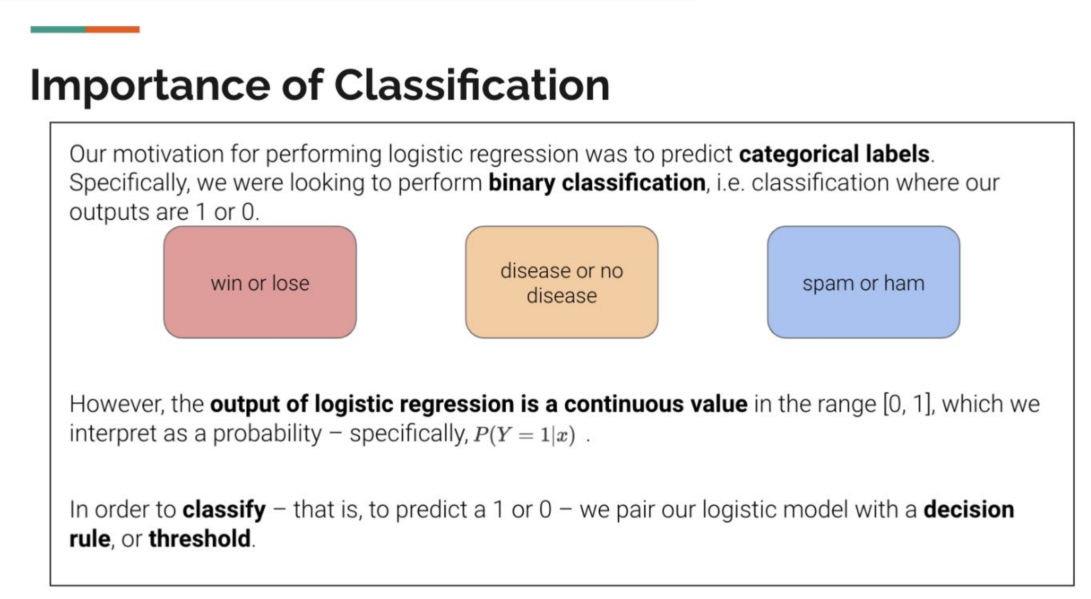
; linear

obabilities.

‹ of this as

babilities.





Importance of Classification

Our motivation for performing logistic regression was to predict **categorical labels**. Specifically, we were looking to perform **binary classification**, i.e. classification where our outputs are 1 or 0.

win or lose

disease or no disease

spam or ham

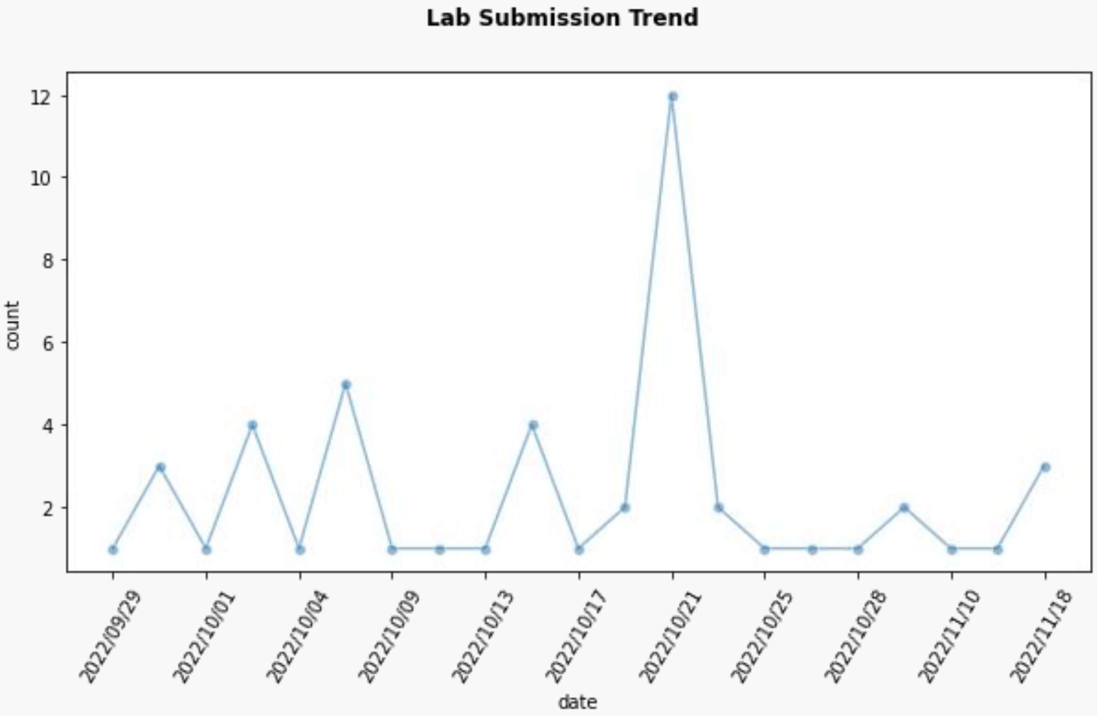
However, the **output of logistic regression is a continuous value** in the range [0, 1], which we interpret as a probability – specifically, $P(Y = 1|x)$.

In order to **classify** – that is, to predict a 1 or 0 – we pair our logistic model with a **decision rule**, or **threshold**.

Feedback for members

Thank you for the great job!

Submission and Completeness



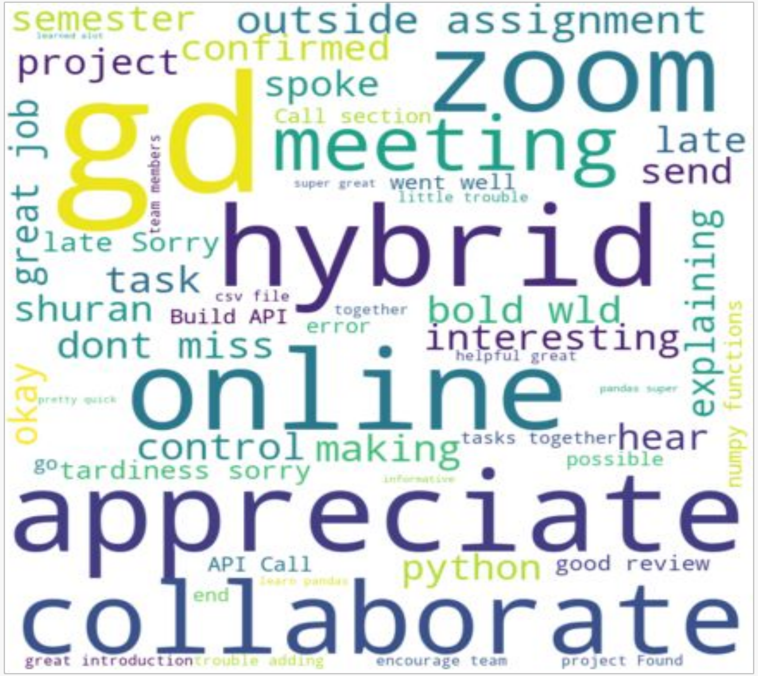
In total we received about 50 submissions. Most people finished labs on NumPy, Pandas, Matplotlib, and Seaborn.

Pretty satisfying results!

Submission and Completeness

Members Feedback andAttitude

Only 23 Feedbacks collected and mostly on first two weeks



The feedback from first two weeks are mostly **positive with meaningful feedback/advices**, members are happy to have a **zoom option**. Though we still receive few abstention of final projects.

Work from members

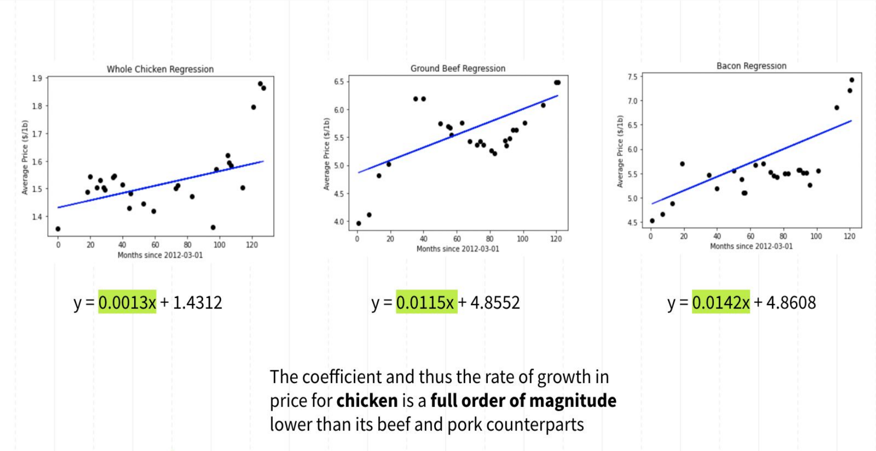
Final Group Presentation

Group	Order	Member
1	**1**	Janelle Correa, Jiayin Lin, Jennifer Ly
2	**2**	Daniel Hwang, William Lee, Bianca Rein Del Rosario
3	**3**	Nadine Ratinho, Meng-Han Wu
4	**5**	Arjun Balaji, Nick Erwin Zhiyang Chia
5	**4**	Yesenia Morales, Sinead McCaffery

We have five groups and all groups presented in the end!!!

Big shout out to all of our members! Goodjob👍

Some highlights

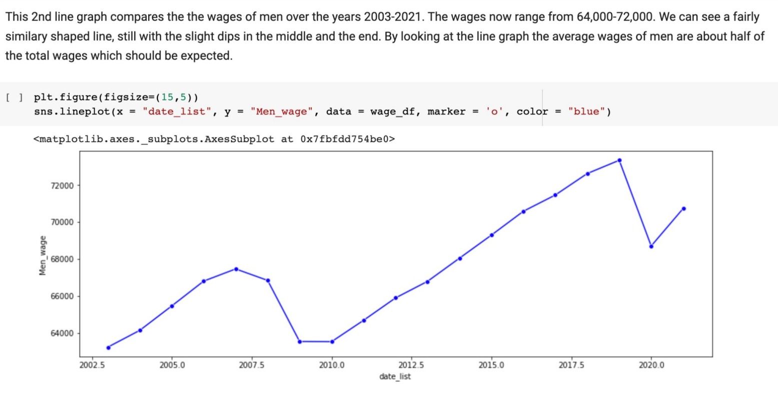


y = 0.0013x + 1.4312

y = 0.0115x + 4.8552

y = 0.0142x + 4.8608

The coefficient and thus the rate of growth in price for **chicken** is a **full order of magnitude** lower than its beef and pork counterparts

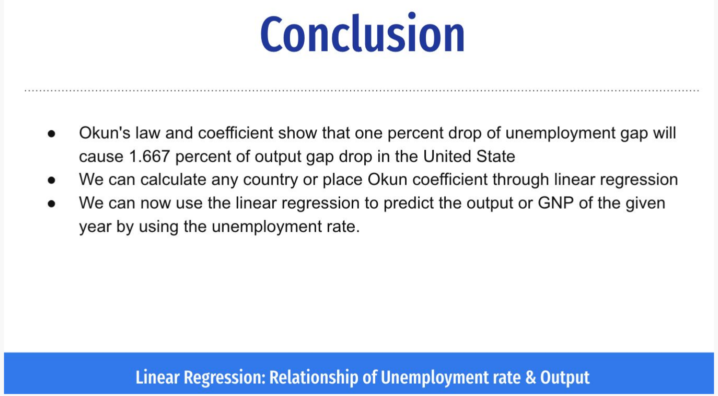


This 2nd line graph compares the the wages of men over the years 2003-2021. The wages now range from 64,000-72,000. We can see a fairly similary shaped line, still with the slight dips in the middle and the end. By looking at the line graph the average wages of men are about half of the total wages which should be expected.

```
[ ]  plt.figure(figsize=(15,5))
     sns.lineplot(x = "date_list", y = "Men_wage", data = wage_df, marker = 'o', color = "blue")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbfdd754be0>
```

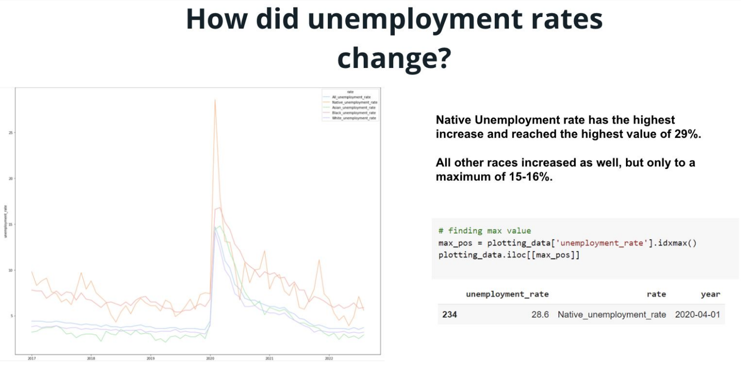
Men_wage
64000
66000
68000
70000
72000
2002.5
2005.0
2007.5
2010.0
2012.5
2015.0
2017.5
2020.0
date_list



Conclusion

- Okun's law and coefficient show that one percent drop of unemployment gap will cause 1.667 percent of output gap drop in the United State
- We can calculate any country or place Okun coefficient through linear regression
- We can now use the linear regression to predict the output or GNP of the given year by using the unemployment rate.

Linear Regression: Relationship of Unemployment rate & Output



How did unemployment rates change?

Native Unemployment rate has the highest increase and reached the highest value of 29%.

All other races increased as well, but only to a maximum of 15-16%.

```
# finding max value
max_pos = plotting_data['unemployment_rate'].idxmax()
plotting_data.iloc[[max_pos]]
```

	unemployment_rate	rate	year
234	28.6	Native_unemployment_rate	2020-04-01

Presentation links:

Group 1: https://colab.research.google.com/drive/1xGPuXtW79-hj9qkM2WK5aiRvKJXfPQOp?usp=sharing
https://docs.google.com/presentation/d/1UH7jM27FaUsvM1bpj_Ei2BjEG38LVp8d5JB6FbnWfQw/edit#slide=id.g1a6389126ac_0_49

Group 2:
https://docs.google.com/presentation/d/1ab8ikVQYLpH-qQ796Qx90M9Fp1LrFeme60BeRZC7kmY/edit?usp=sharing

Group 3:
https://docs.google.com/presentation/d/1l4j_MBctDbKnmeuGwVULnpSnkRdLC1Tb9W7KeZRJ_4Y/edit#slide=id.g8ea72f4a77_6_125

Group 4:
https://docs.google.com/presentation/d/1Z1mZnMecSwL5UE84okfjCS8qhrnEfShRM_86pCeqbXU/edit?usp=sharing

Group 5: https://colab.research.google.com/drive/17yxPA3tHjzYzGNnXx7Bl1sAgRfBu6hu5?usp=sharing

Presentation from project member: Jiayin Lin

https://docs.google.com/presentation/d/1UH7jM27FaUsvM1bpj_Ei2BjEG38LVp8d5JB6FbnWfQw/edit#slide=id.g1a6389126ac_0_49

Thanks for coming!
Hope you enjoyed our semester.