

Final Project - Beer

February 9, 2019

1 Final Project - Beer

```
In [1]: import plotly
        plotly.tools.set_credentials_file(username='rshaver', api_key='2flKjiFtkaIJm3WeXCtj')
```

```
In [2]: import pandas as pd
        import numpy as np

        import plotly.plotly as py
        import plotly.tools as tls
        plotly.offline.init_notebook_mode(connected=True)
```

```
In [6]: from ipywidgets import IntSlider

        slider = IntSlider(value = 50)
        slider
```

```
IntSlider(value=50)
```

```
In [7]: from ipywidgets import HTML

        text = HTML("The slider's value is {}".format(slider.value))
        text
```

```
HTML(value="The slider's value is 50")
```

```
In [8]: def update_html(change):
        text.value = "The slider's value is {}".format(slider.value)
        slider.observe(update_html, 'value')
```

```
In [9]: df = pd.read_csv("brewery_count_by_state_1984_dec2018.csv", header=0)
        df.fillna(0, inplace=True)
        df
```

```
Out[9]:
```

	Unnamed: 0	1984	1985	1986	1987	1988	1989	1990	1991	1992	\
0	AK	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	4.0	

1	AL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	AR	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	AZ	0.0	0.0	0.0	0.0	0.0	6.0	8.0	8.0	7.0
4	CA	9.0	12.0	15.0	23.0	39.0	50.0	67.0	76.0	93.0
5	CO	0.0	0.0	0.0	0.0	3.0	7.0	11.0	22.0	27.0
6	CT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	DC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	DE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	FL	0.0	0.0	0.0	0.0	3.0	6.0	7.0	13.0	17.0
10	GA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	HI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	IA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	5.0
13	ID	0.0	0.0	0.0	3.0	3.0	3.0	3.0	5.0	8.0
14	IL	0.0	0.0	0.0	0.0	4.0	5.0	6.0	9.0	12.0
15	IN	0.0	0.0	0.0	0.0	0.0	4.0	4.0	5.0	6.0
16	KS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	KY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	LA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	3.0
19	MA	0.0	0.0	0.0	3.0	3.0	5.0	5.0	6.0	9.0
20	MD	0.0	0.0	0.0	0.0	0.0	3.0	3.0	4.0	5.0
21	ME	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	5.0
22	MI	0.0	0.0	0.0	0.0	3.0	3.0	3.0	3.0	4.0
23	MN	0.0	0.0	3.0	4.0	5.0	7.0	7.0	8.0	8.0
24	MO	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	4.0
25	MS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26	MT	0.0	0.0	0.0	0.0	0.0	0.0	3.0	5.0	7.0
27	NC	0.0	0.0	0.0	0.0	0.0	6.0	8.0	11.0	12.0
28	ND	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
29	NE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0
30	NH	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
31	NJ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
32	NM	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	5.0
33	NV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
34	NY	0.0	0.0	3.0	4.0	5.0	6.0	8.0	13.0	16.0
35	OH	0.0	0.0	0.0	0.0	3.0	5.0	6.0	9.0	12.0
36	OK	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
37	OR	0.0	3.0	5.0	8.0	13.0	15.0	21.0	26.0	30.0
38	PA	4.0	4.0	5.0	6.0	6.0	8.0	11.0	12.0	12.0
39	RI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
40	SC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
41	SD	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
42	TN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
43	TX	4.0	4.0	4.0	4.0	5.0	7.0	8.0	7.0	8.0
44	UT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	4.0
45	VA	0.0	0.0	0.0	0.0	0.0	0.0	4.0	5.0	7.0
46	VT	0.0	0.0	0.0	0.0	0.0	3.0	4.0	6.0	6.0
47	WA	4.0	4.0	4.0	4.0	7.0	8.0	10.0	13.0	15.0
48	WI	5.0	6.0	7.0	10.0	12.0	14.0	17.0	18.0	19.0

49	WV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	WY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
51	Total	26.0	33.0	46.0	69.0	114.0	171.0	224.0	306.0	381.0

	...	2009	2010	2011	2012	2013	2014	2015	2016	2017	\
0	...	18	20	23	25	25	28	35	36	45	
1	...	5	7	9	15	23	25	30	37	52	
2	...	5	5	7	13	18	23	29	34	44	
3	...	34	37	45	55	59	77	91	110	130	
4	...	340	358	401	472	528	654	788	927	1,106	
5	...	116	129	149	185	234	300	352	386	448	
6	...	19	21	20	24	39	50	59	76	103	
7	...	3	4	6	5	11	12	13	13	13	
8	...	11	11	12	13	13	15	21	25	33	
9	...	52	60	71	89	113	158	205	264	338	
10	...	22	26	25	31	37	48	54	69	102	
11	...	11	11	12	15	17	20	26	23	28	
12	...	28	27	35	46	54	60	71	94	115	
13	...	25	25	33	39	42	51	57	67	76	
14	...	60	62	68	95	123	164	210	244	291	
15	...	39	43	59	72	95	116	151	163	213	
16	...	17	17	21	21	25	27	37	47	53	
17	...	15	14	21	25	25	38	48	60	73	
18	...	11	12	12	12	15	17	25	34	43	
19	...	44	48	52	65	82	98	124	146	189	
20	...	26	24	26	40	43	55	73	88	116	
21	...	42	44	49	53	60	71	84	102	131	
22	...	104	111	131	160	195	256	316	379	452	
23	...	39	42	56	70	75	113	142	165	214	
24	...	47	51	55	60	63	77	90	116	145	
25	...	0	3	0	4	7	10	14	14	16	
26	...	31	31	35	41	49	62	74	79	98	
27	...	52	61	71	94	125	155	207	260	330	
28	...	3	3	3	7	8	10	11	15	22	
29	...	17	16	18	20	25	35	39	47	53	
30	...	17	19	21	25	35	46	63	73	88	
31	...	22	24	31	29	37	49	71	96	123	
32	...	21	26	30	38	48	60	71	86	110	
33	...	18	20	20	23	26	34	39	44	46	
34	...	84	100	123	150	193	255	329	394	471	
35	...	66	75	81	99	114	151	187	236	324	
36	...	10	10	10	13	17	18	21	26	43	
37	...	108	120	148	180	220	244	281	304	347	
38	...	96	118	146	162	183	233	278	333	411	
39	...	5	5	6	9	12	14	15	17	27	
40	...	15	16	17	23	30	37	51	59	84	
41	...	7	8	9	12	12	15	19	21	28	
42	...	23	26	31	45	51	69	88	101	120	

43	...	47	59	84	107	128	170	220	266	333
44	...	18	18	18	22	24	27	29	34	39
45	...	42	44	54	67	85	117	155	209	287
46	...	27	32	34	39	45	51	66	73	84
47	...	138	157	188	230	266	314	383	424	499
48	...	108	119	126	146	146	168	189	217	261
49	...	8	8	8	8	9	13	17	24	27
50	...	13	16	15	19	25	28	32	33	39
51	...	2,129	2,343	2,725	3,312	3,934	4,938	6,080	7,190	8,863

2018

0	51
1	55
2	53
3	146
4	1,236
5	500
6	124
7	13
8	38
9	386
10	121
11	38
12	125
13	87
14	338
15	234
16	64
17	86
18	47
19	230
20	141
21	165
22	510
23	239
24	168
25	19
26	108
27	387
28	26
29	60
30	106
31	146
32	124
33	50
34	532
35	377
36	55

```

37      386
38      472
39      33
40      94
41      35
42     140
43     387
44      46
45     328
46      90
47     540
48     303
49      32
50      44
51  10,115

```

```
[52 rows x 36 columns]
```

```

In [10]: # Converts the data frame above into a CSV file
df.to_csv(r'/Users/xavieryelnatz/Desktop/Python for DS (UCSD)/Final Project - Beer\br

```

1.1 Example of Choropleth Map

```

In [11]: df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/2011_us_ag

for col in df.columns:
    df[col] = df[col].astype(str)

scl = [[0.0, 'rgb(242,240,247)'],[0.2, 'rgb(218,218,235)'],[0.4, 'rgb(188,189,220)'],
       [0.6, 'rgb(158,154,200)'],[0.8, 'rgb(117,107,177)'],[1.0, 'rgb(84,39,143)']

df['text'] = df['state'] + '<br>' +\
    'Beef '+df['beef']+' Dairy '+df['dairy']+'<br>'+\
    'Fruits '+df['total fruits']+' Veggies ' + df['total veggies']+'<br>'+\
    'Wheat '+df['wheat']+' Corn '+df['corn']

data = [ dict(
    type='choropleth',
    colorscale = scl,
    autocolorscale = False,
    locations = df['code'],
    z = df['total exports'].astype(float),
    locationmode = 'USA-states',
    text = df['text'],
    marker = dict(
        line = dict (
            color = 'rgb(255,255,255)',
            width = 2

```

```

    ) ),
    colorbar = dict(
        title = "Millions USD")
    ]

layout = dict(
    title = '2011 US Agriculture Exports by State<br>(Hover for breakdown)',
    geo = dict(
        scope='usa',
        projection=dict( type='albers usa' ),
        showlakes = True,
        lakecolor = 'rgb(255, 255, 255)'),
    )

fig = dict( data=data, layout=layout )
py.iplot( fig, filename='d3-cloropleth-map' )

```

High five! You successfully sent some data to your account on plotly. View your plot in your browser.

Out[11]: <plotly.tools.PlotlyDisplay object>

1.2 Things to Do:

- Find out how to map / scale the US states, and overlay data onto each
- Find a method of visualizing this data over time (years)
- Convert Excel format to one which Jupyter can use (Plot.ly)

1.3 Plot.ly

- <https://plot.ly/python/ipython-notebook-tutorial/>
- <https://plot.ly/python/choropleth-maps/>
- <https://www.youtube.com/watch?v=hA39KSTb3dY>
- <https://plot.ly/python/#animations>
- <https://plot.ly/python/gapminder-example/>
- <https://plot.ly/python/bubble-maps/>

1.4 Folium / Leaflet

- <https://www.youtube.com/watch?v=4RnU5qKTfYY> - <https://www.youtube.com/watch?v=xN2N-p33V1k>

1.5 Widgets

- <https://www.youtube.com/watch?v=1ndo6C1KWjI>
- <https://www.youtube.com/watch?v=i40d8-Hu4vM>
- ipywidgets (core UI controls / sliders)
- bqplot (2d plotting)
- pythreejs, ipyvolume (3d plotting)
- ipyleaflet (maps)

1.6 CSV Analysis / ML

- <https://www.youtube.com/watch?v=-0NwrcZOKhQ>
- [https://www.youtube.com/watch?v=Q73ADVZCq\\$U](https://www.youtube.com/watch?v=Q73ADVZCq$U)
- <https://www.youtube.com/watch?v=OBPjFnyxoCc>
- https://www.youtube.com/watch?v=zJ4RK6jtYCU&list=PLbD3QT5__Llz88nB-B-Kp5s118DOkAHr1

1.7 Resources

- <https://plot.ly/python/choropleth-maps/>
- <https://www.kaggle.com/rdoume/beerreviews>
- <https://www.kaggle.com/ehallmar/beers-breweries-and-beer-reviews>
- <http://beer.tany.kim/>
- <https://untappd.com/api/docs>
- https://www.reddit.com/r/Untappd/comments/41i45t/mass_data_export/
- <https://www.kaggle.com/nickhould/craft-cans>
- <https://data.world/datafiniti/breweries-brew-pubs-in-the-usa>
- https://www.reddit.com/r/datasets/comments/6i0v3g/craft_beer_dataset/
- <https://github.com/nickhould/craft-beers-dataset>
- <https://catalog.data.gov/dataset?tags=beer>
- <https://catalog.data.gov/dataset/yearly-statistical-beer-data-by-state-2007-2016>
- <https://catalog.data.gov/dataset/beer-production-and-operations-reports>
- <https://catalog.data.gov/dataset/brewery-count-by-state-1984-march-31-2017>
- <https://catalog.data.gov/dataset/brewery-count-by-state-1984-march-31-2018>
- <https://www.ftb.gov/foia/fri.shtml>