Application of Big Data in Social Science

week 15

Heidi Hyeseung Choi
Fall 2022
HYSIS
heidichoi@hanyang.ac.kr

Announcements

~~9.1~~	~~1~~	~~Introduction~~
~~9.8~~	~~2~~	~~Web Scraping~~
~~9.15~~	~~3~~	
~~9.22~~	~~4~~	~~Natural Language Processing~~
~~9.29~~	~~5~~	
~~10.6~~	~~6~~	~~Text Analysis~~ ~~(recorded lecture on week 7)~~
~~10.13~~	~~7~~	
~~10.20~~	~~8~~	~~Mid term exam (as school schedule)~~
~~10.27~~	~~9~~	~~Midterm review & word cloud~~
~~11.3~~	~~10~~	~~Social Network Analysis~~
~~11.10~~	~~11~~	~~COVID-19 T-T~~
~~11.17~~	~~12~~	~~Machine Learning: Supervised Learning~~
~~11.24~~	~~13~~	~~Supervised Learning~~
~~12.1~~	~~14~~	~~Machine Learning: Unsupervised Learning~~
12.8	**15**	**Data Visualization**
12.15	16	Final Exam

Homework assignment 2

- Due 22nd December.

You have two options to choose from:

1. Devising a marketing strategy.
2. Visualization.
   - Made a little change in here.

Homework assignment 2: working with raw data

Option 1: Devising a marketing strategy

- You are at a marketing team and you want to do a target marketing
- Using raw data given, you are to process data and using analysis that we have done during the class, try to come up with target marketing.
- For each target group, you can write a short paragraph describing their consuming behavior, and how you would conduct target marketing for each group.
- Within data, there are a lot of info available so whichever data you choose, it is up to you.
- Hand in your 1)jupyter notebook file and 2)a word file with figures and written descriptions. (up to 4 pages including everything would be fine)

Homework assignment 2: working with raw data (choose one!)

Option 2. Visualization:

Datasets: 1. USDA ERS - County-level Data Sets 2.: World bank database

1. For US data population.
    1. State level choropleth map.
    2. COUNTY level choropleth map for ONE state of your choice.
2. World data population.
    1. Use world bank database to download.
    2. Manipulate data. (should include ISO-3)

- Hand in your 1) jupyter notebook file and 2) word file with figures in them.
- # of figures, etc does not matter as long as you are able to get the figures out.

Final exam: Dec 15th 1pm – 2:30pm

- Total of 37 questions:
  - lms: 22 questions x 2.5 points= 55 points
  - coding: 15 questions x 3 points = 45 points.
  - might be too many, so I might adjust a little

- Mostly about theoretical aspects
  - eg) What is modularity?
  - eg) what is true about unsupervised machine learning?

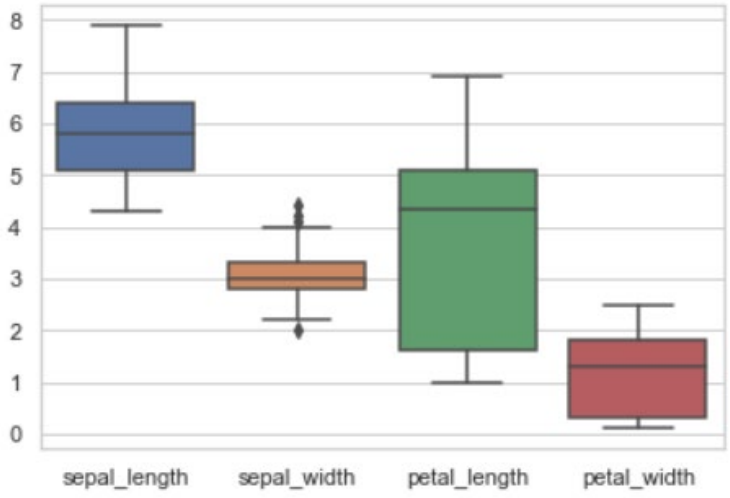
- Coding questions:
  - will give you documentation and ask you to code it
  - ask about simple grammar that we learned
  - interpreting the code.
    - What would be the likely output of the following code?
    - Which library would you most likely to import to do the following?
    - What would this line of code do?

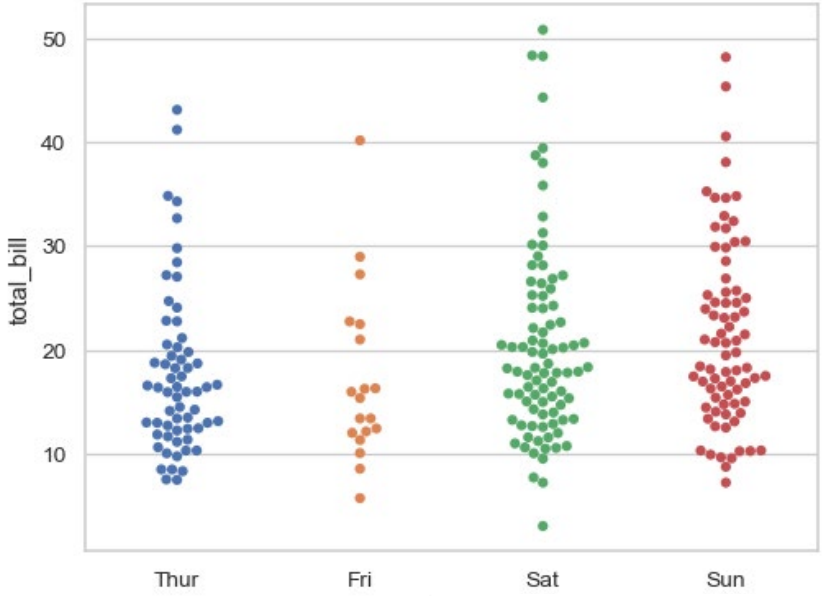
Scope of the exam:
What we learned from week 10

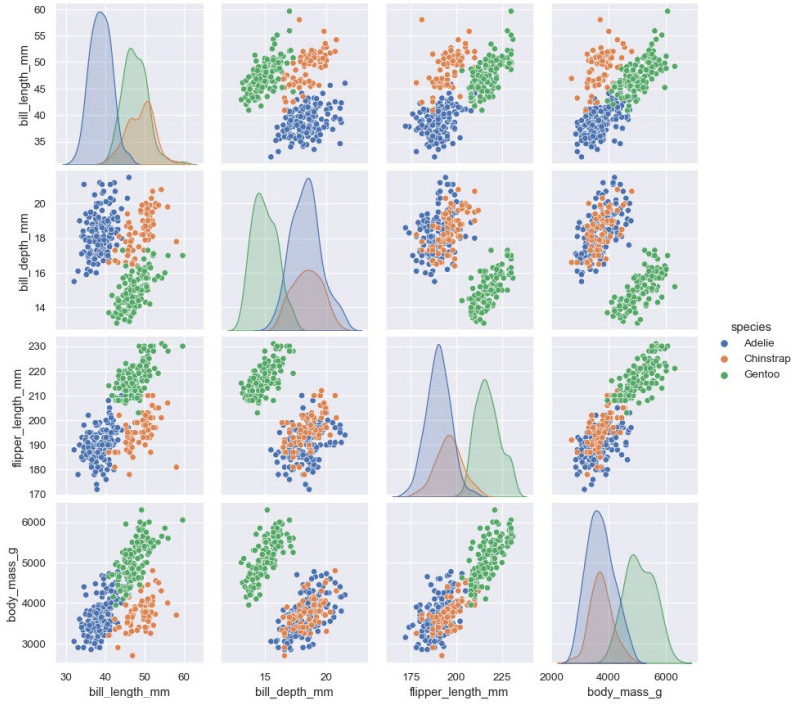
11.3	10	Social Network Analysis
11.10	11	COVID-19 T-T
11.17	12	Machine Learning: Supervised Learning
11.24	13	Supervised Learning
12.1	14	Machine Learning: Unsupervised Learning
12.8	15	Data Visualization

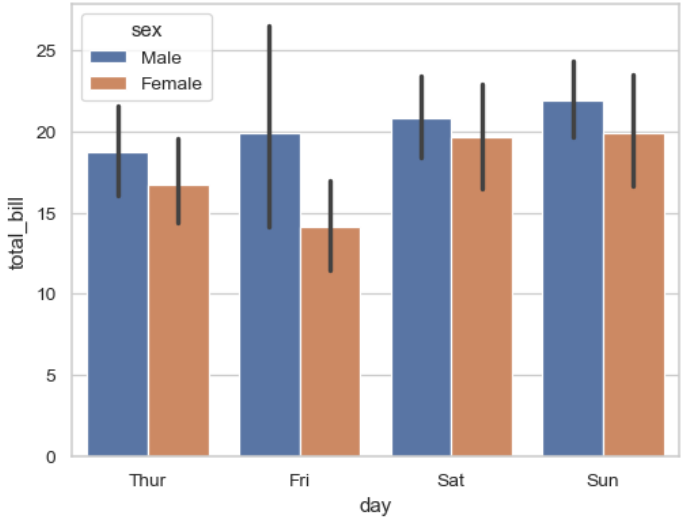
Plot basics

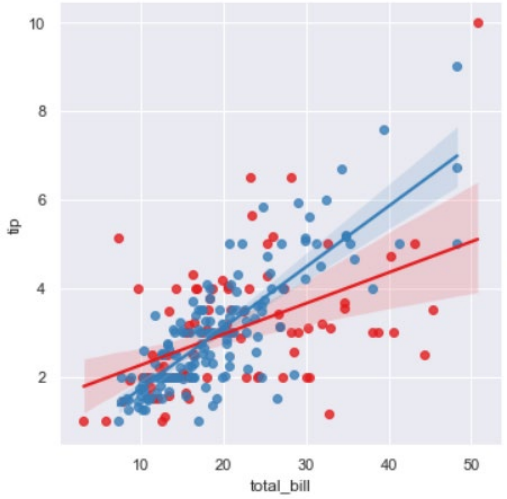
Different types of plots

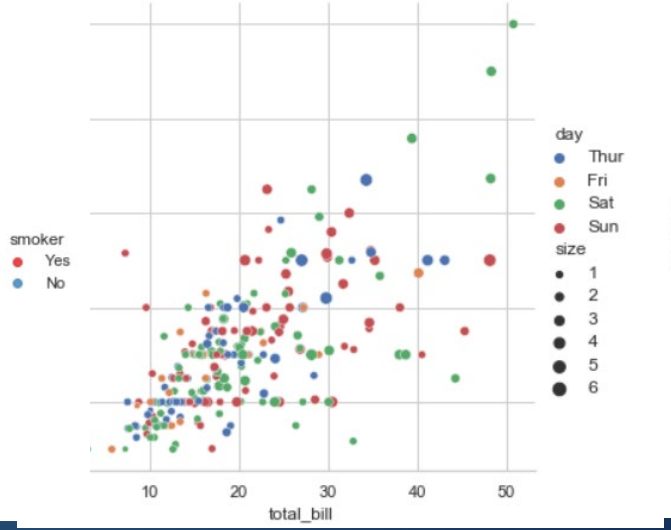


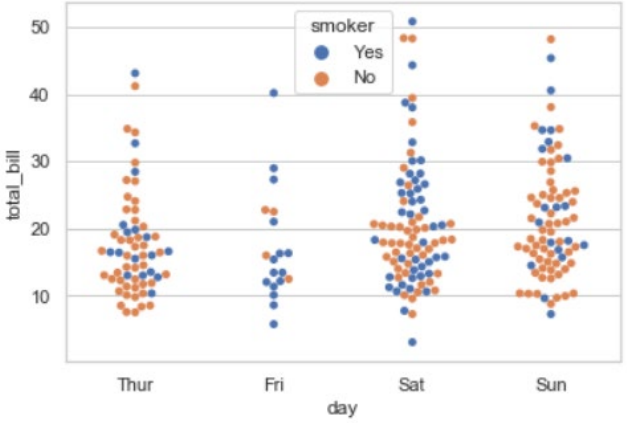












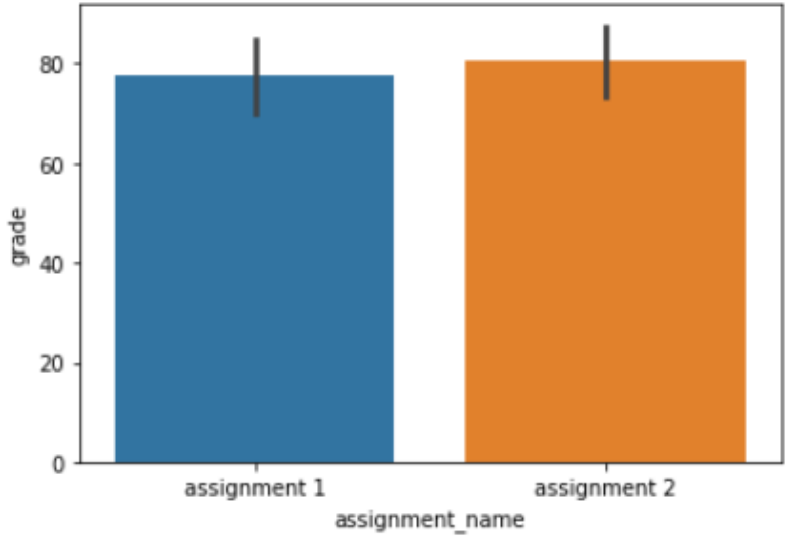
seaborn: barplot

the barplot() function operates on a full dataset and applies a function to obtain the estimate (taking the mean by default).

```
>>> sns.barplot(data = student, x = "assignment_name", y ="grade")
```

- it computes a ci(confidence interval) around the estimate and it is plotted using error bars.
- This indicates based on our data, 95% of similar situations would have an outcome within this range.

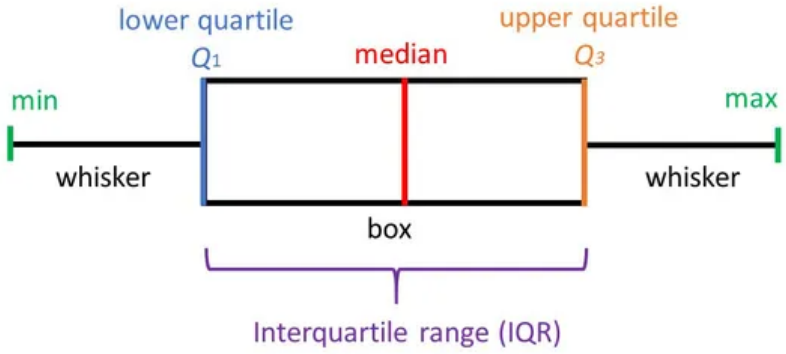
```
>>> sns.barplot(data = student,
>>>    x = "assignment_name",
>>>    y ="grade", ci = "sd")
```

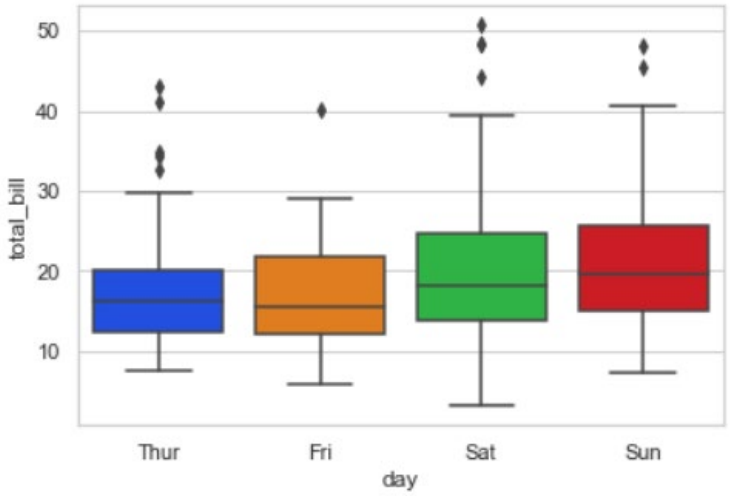


boxplots

- Boxplot shows distribution and skewness of data by looking at data quartiles and average. This kind of plot shows the three quartile values of the distribution along with extreme values.
- The “whiskers” extend to points that lie within 1.5 IQRs of the lower and upper quartile, and then observations that fall outside this range are displayed independently called “outliers”.
- This means that each value in the boxplot corresponds to an actual observation in the data.

```
>>> sns.boxplot(x = ‘column’, y=‘column’, data=data)
```

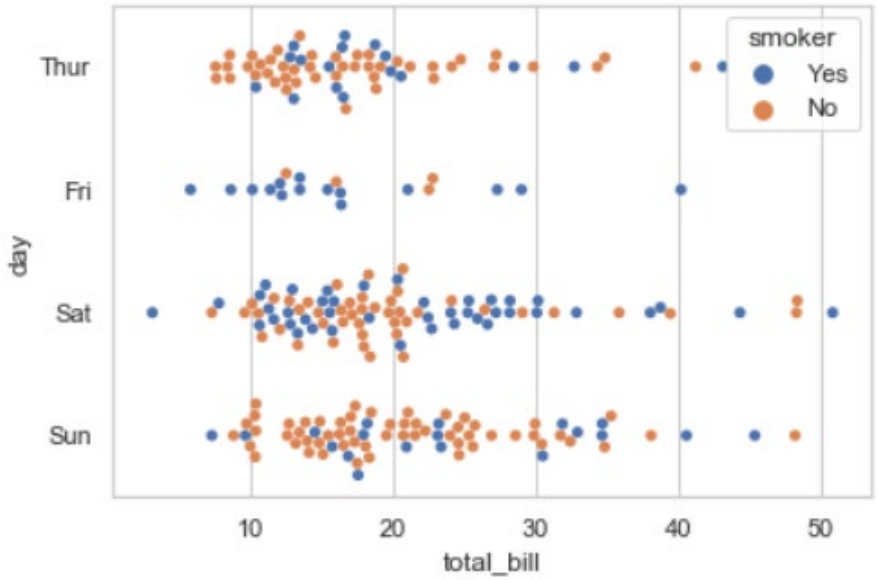


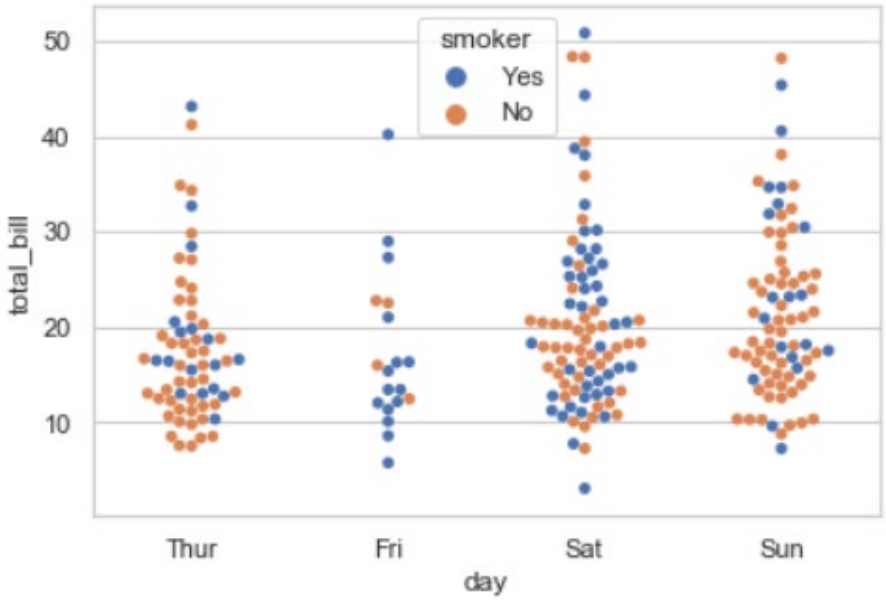


swarmplot

- To complement a boxplot or violin plot, swarmplot show all observations along with some representation of the underlying distribution.
- Draws a categorical scatterplot with non-overlapping points.

```
>>>sns.swarmplot(x =“column”, y=“column”, data = df)
```

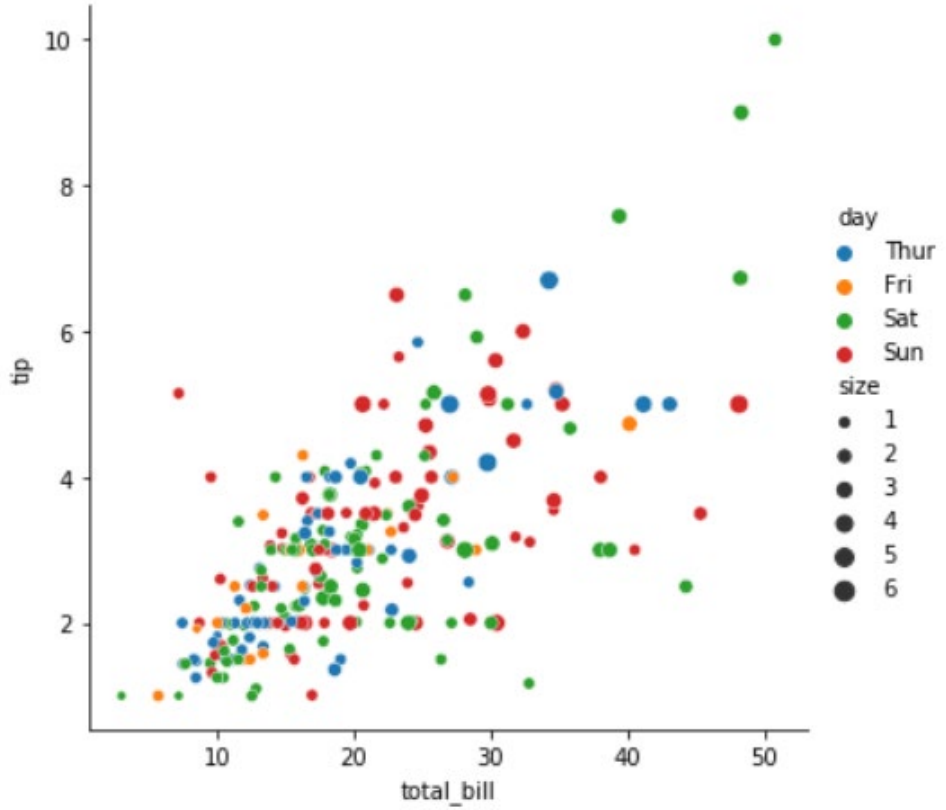


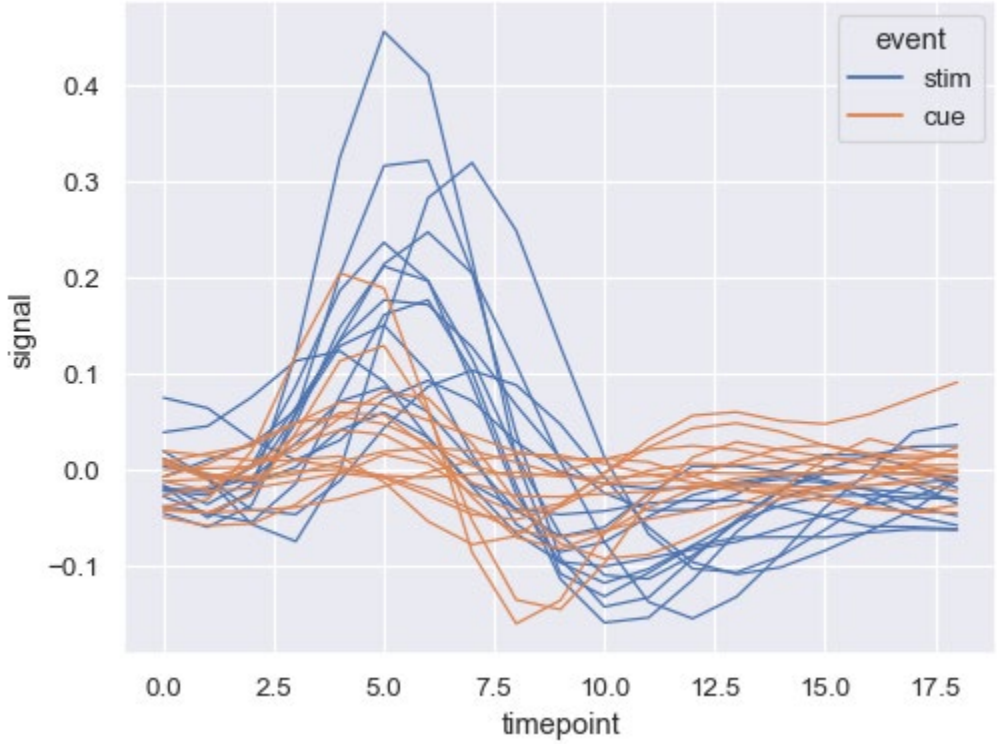


Scatterplot and lineplot

- To show relationship between two variables.

```
>>> sns.scatterplot (x =“column”, y=“column”, data = df)
>>> sns.relplot(x =“column”, y=“column”, data = df, kind=“line”)
```

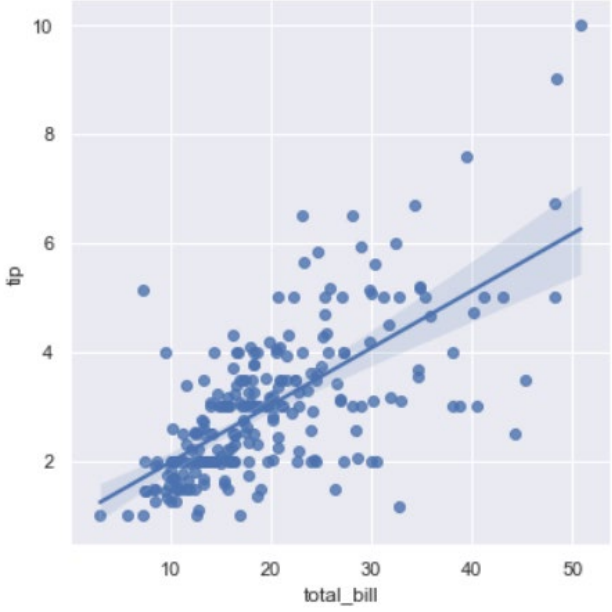


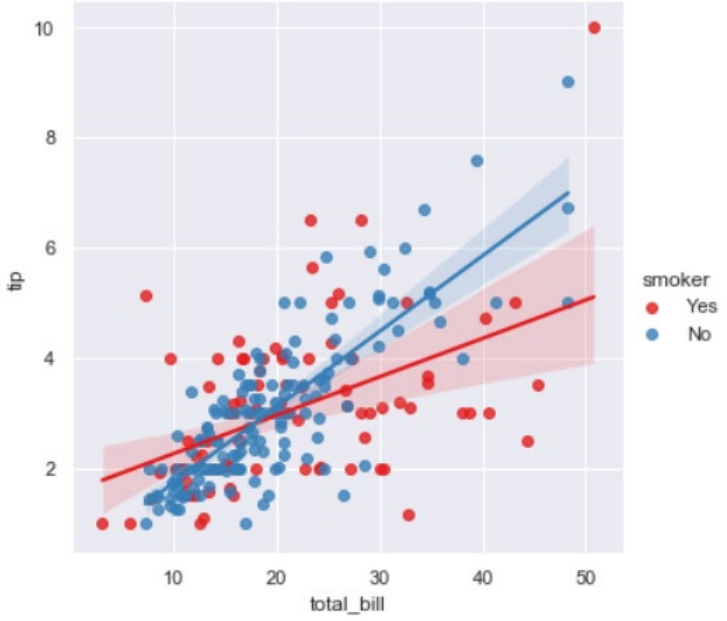


lmplot: linear model

- lmplot draws a linear line across scatter plot to find the best available fit.
- This allows use to predict the best fit, thus we would be able to make generalized prediction.

```
>>> sns.lmplot(x =“column”, y=“column”, data = df)
```



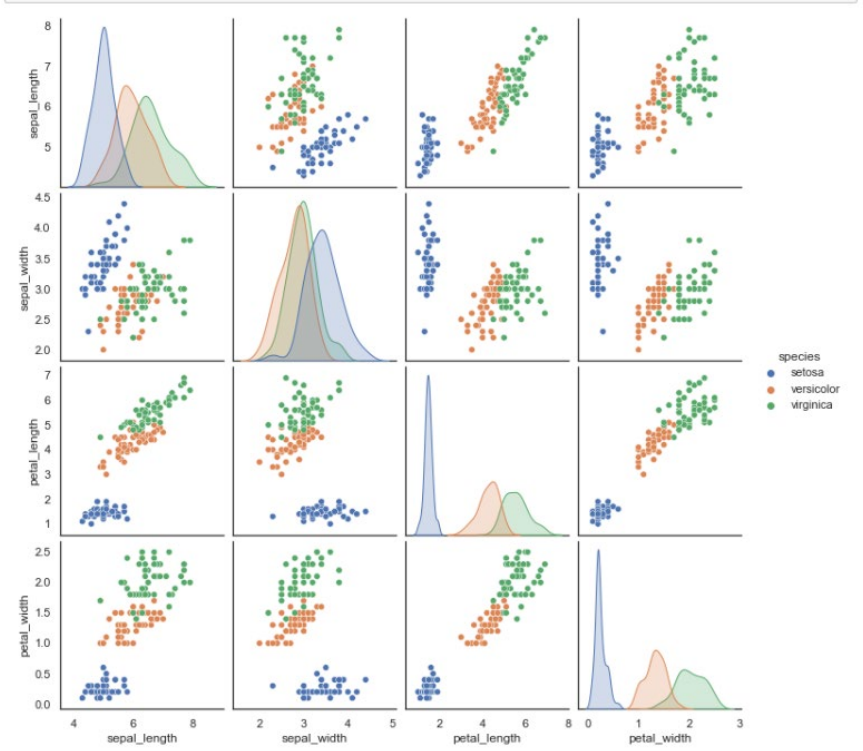


pairplot

- A **pairplot** plot a pairwise relationships in a dataset.
- The **pairplot** function creates a grid of Axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column.

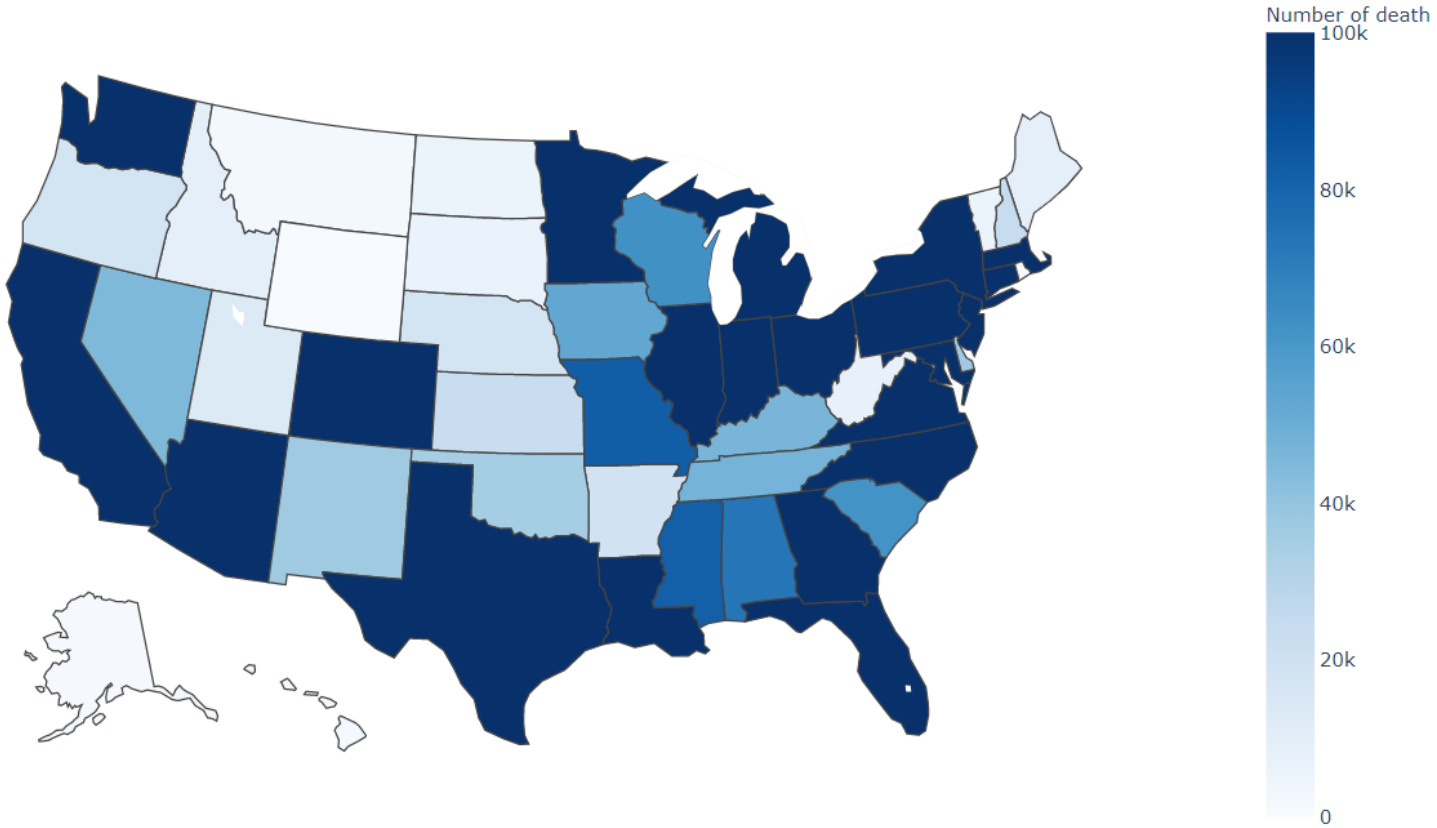
```
>>>sns.pairplot(data)
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica



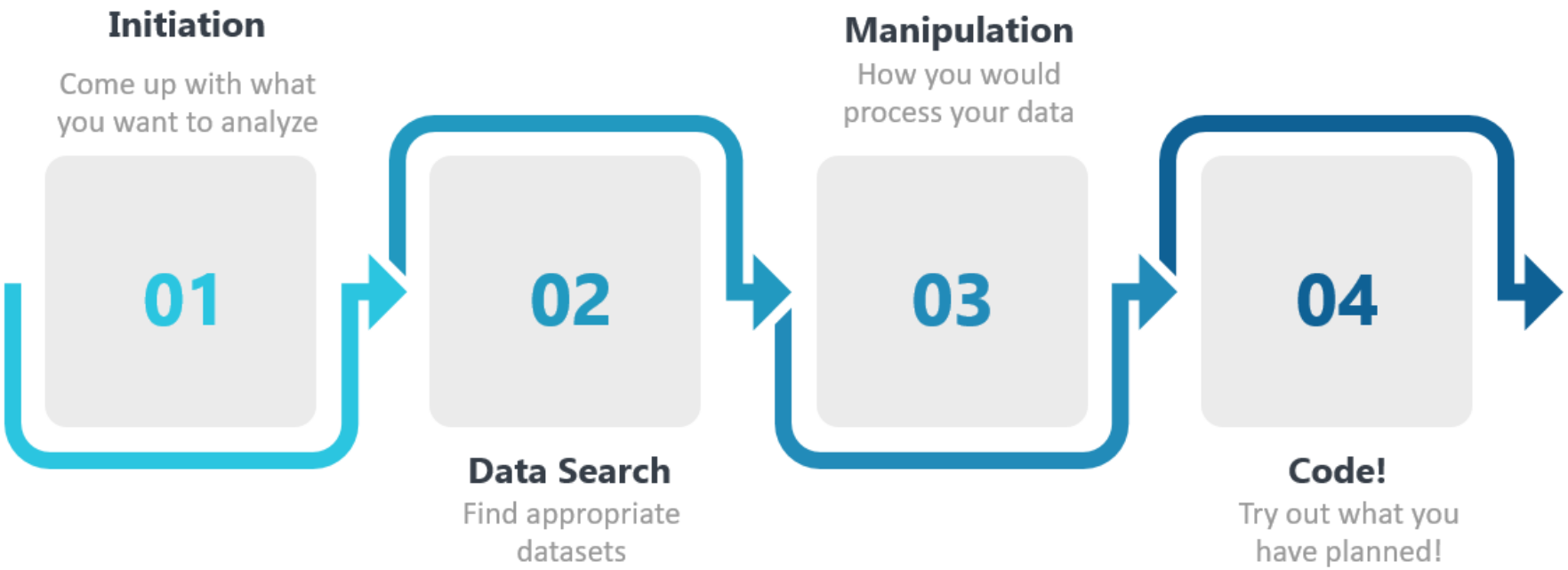
Data visualization!

- Using real data, we will be plotting choropleth today.



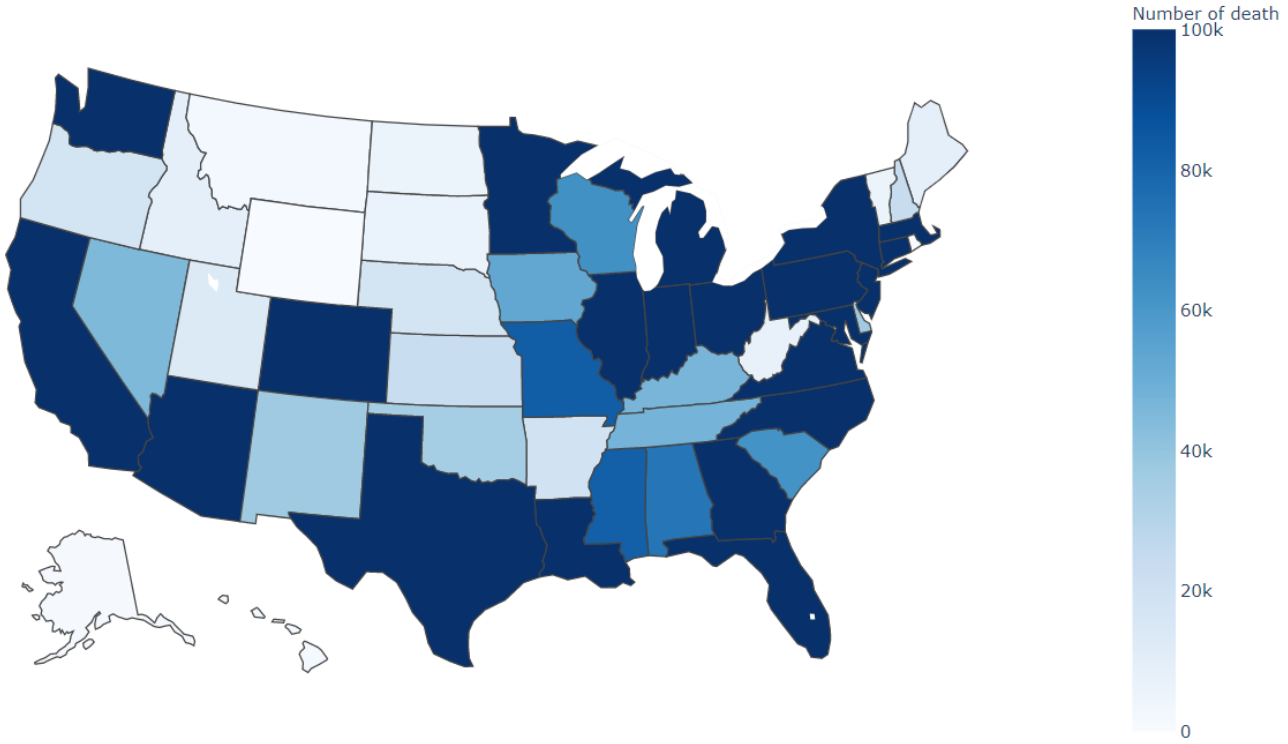
Handling real data.

- Data is not really readily available for you.
- You would need to manipulate data so that you can analyze them.
  - Think of how you want to process your data.
  - Google them out (stackoverflow etc).
- Write down the order that you want to process your data.



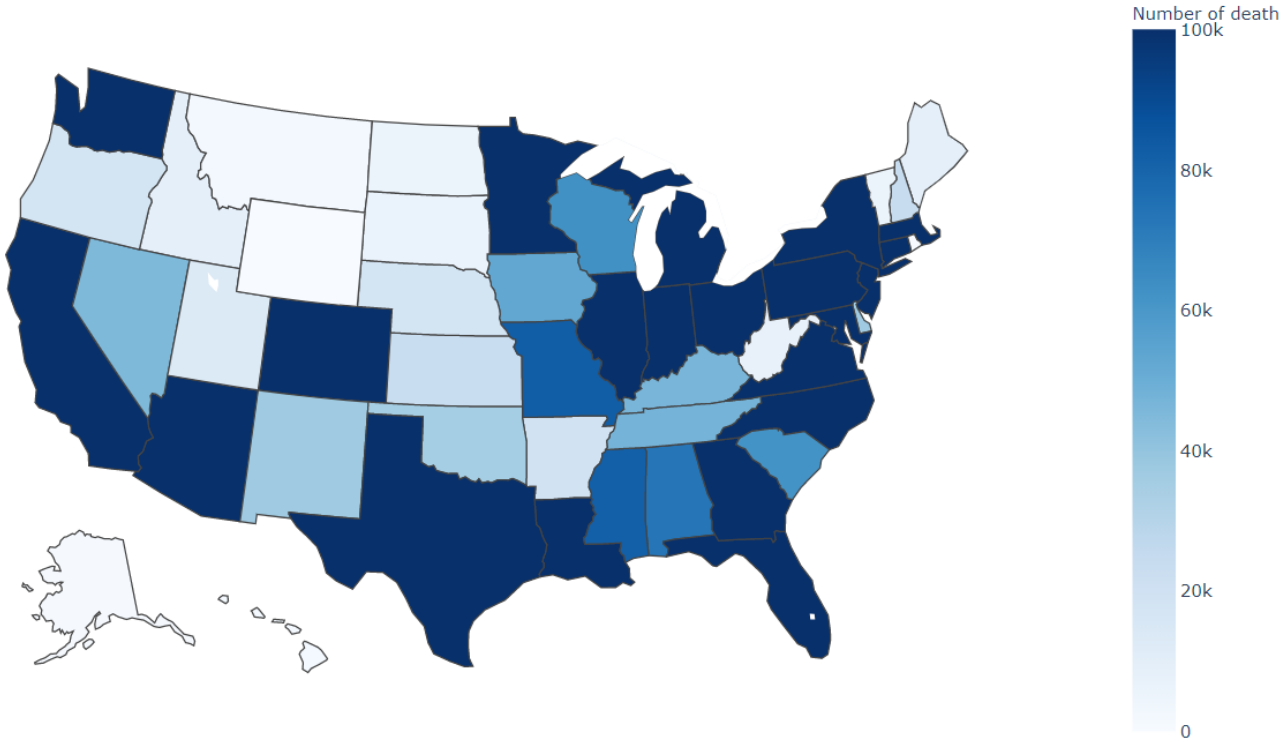
For our case..

- I want to visualize like this below using COVID-19 datasets.
- What is this map called?!?



For our case..

- Choropleth map!
- County level? State level?



For our case..

- Look for available data.
- Our data.

Data 1: usa_county data

	A	B	C	D	E
1	county	Province_State	Date	Confirmed	Deaths
2	Autauga	Alabama	1/22/20	0	0
3	Autauga	Alabama	1/23/20	0	0
4	Autauga	Alabama	1/24/20	0	0
5	Autauga	Alabama	1/25/20	0	0
6	Autauga	Alabama	1/26/20	0	0
7	Autauga	Alabama	1/27/20	0	0
8	Autauga	Alabama	1/28/20	0	0
9	Autauga	Alabama	1/29/20	0	0
10	Autauga	Alabama	1/30/20	0	0
11	Autauga	Alabama	1/31/20	0	0

Data 2: state_code data

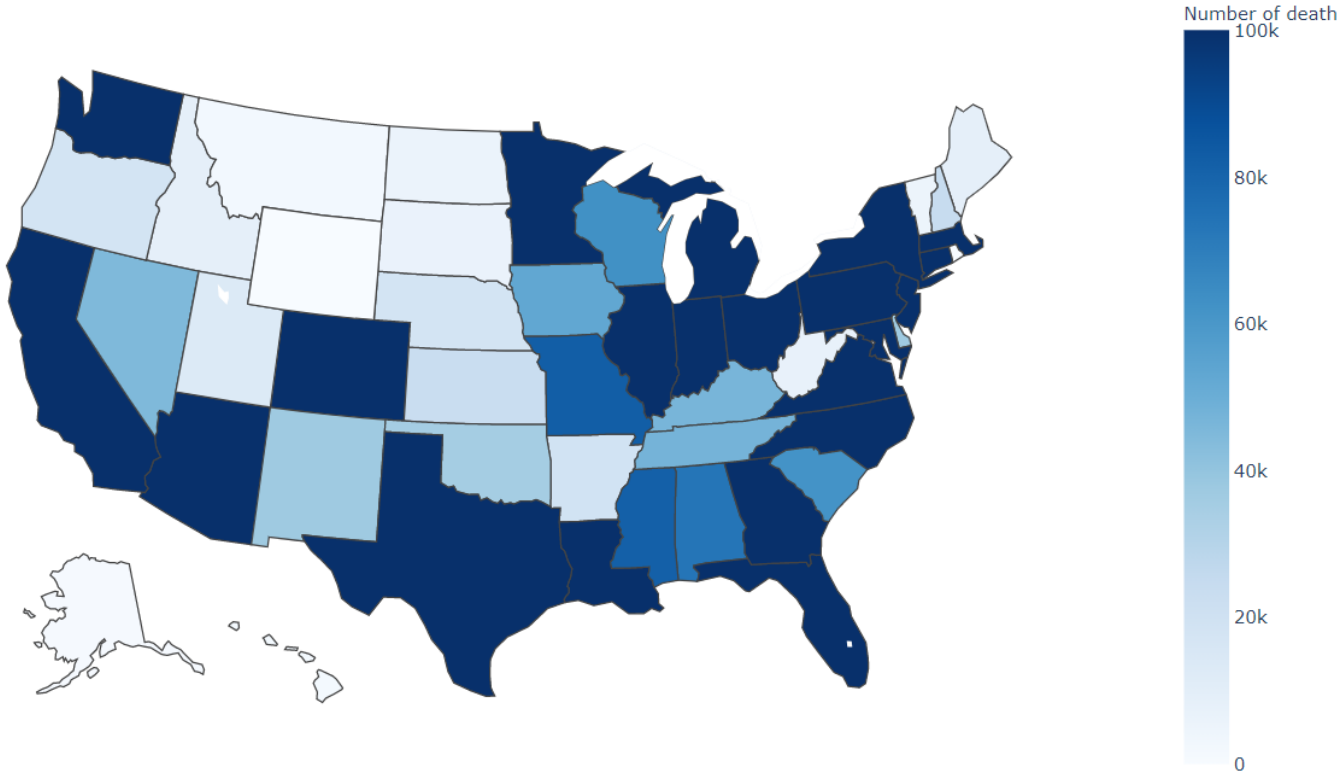
	A	B
	state	code
	Alabama	AL
	Alaska	AK
	Arizona	AZ
	Arkansas	AR
	California	CA
	Colorado	CO
	Connecticut	CT
	Delaware	DE
0	Florida	FL
1	Georgia	GA
2	Hawaii	HI
3	Idaho	ID
4	Illinois	IL

Data 3: FIPS data

A	B	C
FIPS	county	State
1001	Autauga	AL
1003	Baldwin	AL
1005	Barbour	AL
1007	Bibb	AL
1009	Blount	AL
1011	Bullock	AL
1013	Butler	AL
1015	Calhoun	AL
1017	Chambers	AL
1019	Cherokee	AL
1021	Chilton	AL
1023	Choctaw	AL
1025	Clarke	AL
1027	Clay	AL
1029	Cleburne	AL
1031	Coffee	AL
1033	Colbert	AL
1035	Conecuh	AL
1037	Coosa	AL
1039	Covington	AL
1041	Crenshaw	AL

Visualization

- Choropleth Map!
  - Map that allows you to differentiate regions, countries etc.
  - and visualize!
- Use Plotly to visualize!



df.groupby()

- groupby() function of pandas is used when you want to do some calculations on the groups.

```
>>> deaths= df.groupby(by='Province_State').max()
```

	Province_State	county	Deaths
0	Alabama	Autauga	803
1	Alabama	Baldwin	882
2	Alabama	Barbour	137
3	Alabama	Bibb	95
4	Alabama	Blount	72
...	...	...	...
3146	Wyoming	Sweetwater	0
3147	Wyoming	Teton	0
3148	Wyoming	Uinta	0
3149	Wyoming	Washakie	0
3150	Wyoming	Weston	0

3151 rows × 3 columns

```
>>> deaths= df.groupby(by='Province_State').agg({'Deaths':sum})
```

	county	Province_State	Date	Confirmed	Deaths
0	Autauga	Alabama	1/22/20	0	0
1	Autauga	Alabama	1/23/20	0	0
2	Autauga	Alabama	1/24/20	0	0
3	Autauga	Alabama	1/25/20	0	0
4	Autauga	Alabama	1/26/20	0	0
...	...	...	...	...	...
592383	Weston	Wyoming	7/23/20	4	0
592384	Weston	Wyoming	7/24/20	4	0
592385	Weston	Wyoming	7/25/20	4	0
592386	Weston	Wyoming	7/26/20	4	0
592387	Weston	Wyoming	7/27/20	5	0

	Province_State	Deaths	Confirmed
0	Alabama	73332	2843036
1	Alaska	1244	85667
2	Arizona	127359	5272291
3	Arkansas	19048	1364980
4	California	481219	17618410

DataFrame.groupby(*by=None, axis=0, level=None, as_index=True, sort=True, group_keys=True, squeeze=NoDefault.no_default, observed=False, dropna=True*)

4 ways to combine data sets: merge!

df1

x1	x2
A	1
B	2
C	3

+

df2

x1	x3
A	T
B	T
D	F

=

x1	x2	x3
A	1	T
B	2	T
C	3	NaN

pd.merge(df1, df2,
how = 'left', on = 'x1')

✓ Join matching rows from df2 to df1

x1	x2	x3
A	1	T
B	2	T

pd.merge(df1, df2,
how = 'inner', on = 'x1')

✓ Only retain rows existing in both sets

x1	x2	x3
A	1	T
B	2	T
D	NaN	F

pd.merge(df1, df2,
how = 'right', on = 'x1')

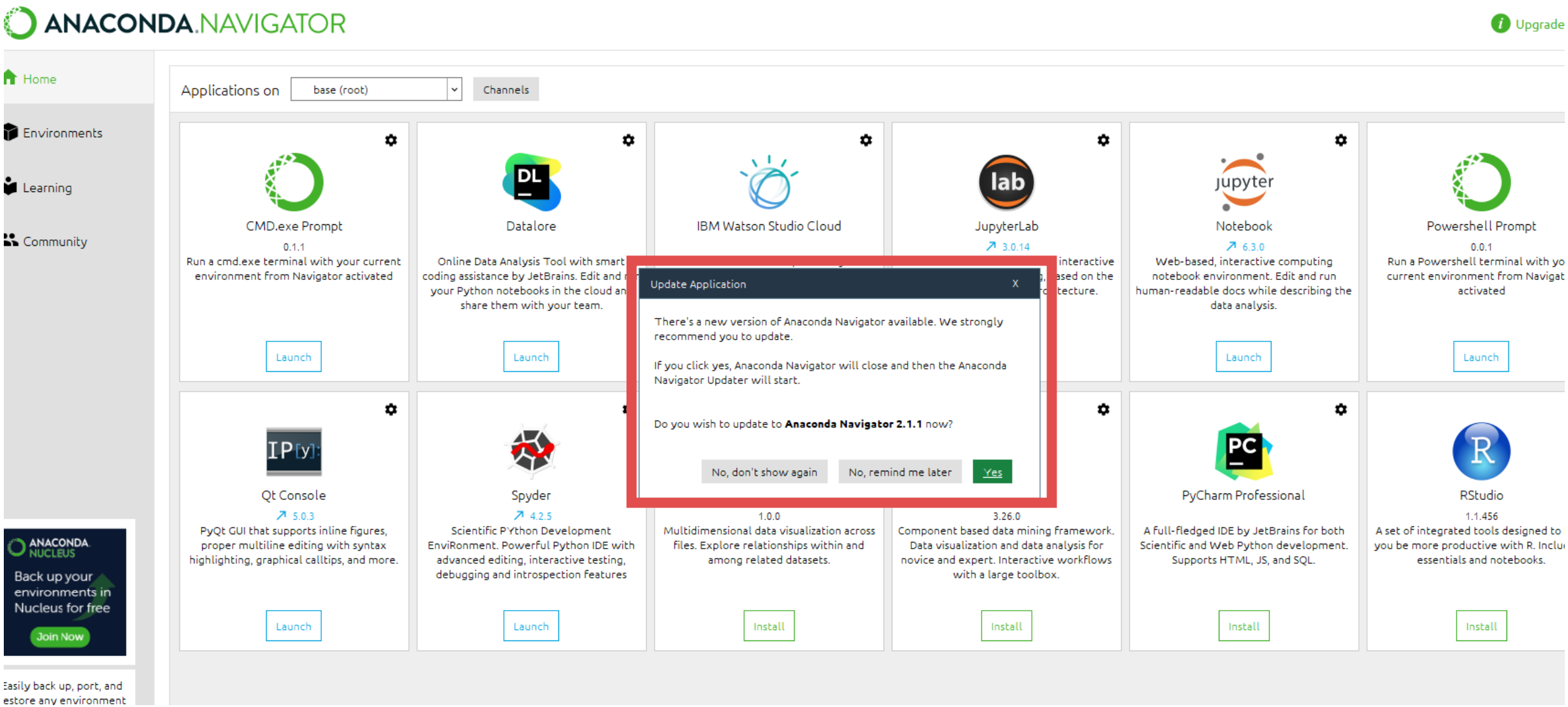
✓ Join matching rows from df1 to df2

x1	x2	x3
A	1	T
B	2	T
C	3	NaN
D	NaN	F

pd.merge(df1, df2,
how = 'outer', on = 'x1')

✓ Retain all values, all rows

Update anaconda time to time!



data status!!

county_df

	Province_State	county	Deaths	Confirmed
0	Alabama	Autauga	803	35257
1	Alabama	Baldwin	882	64254
2	Alabama	Barbour	137	21840
3	Alabama	Bibb	95	12080
4	Alabama	Blount	72	15676
...	...	...	...	...
3146	Wyoming	Sweetwater	0	6606
3147	Wyoming	Teton	0	13368
3148	Wyoming	Uinta	0	8800
3149	Wyoming	Washakie	0	3008
3150	Wyoming	Weston	0	110

3151 rows × 4 columns

fips

	FIPS	county	State
0	01001	Autauga	AL
1	01003	Baldwin	AL
2	01005	Barbour	AL
3	01007	Bibb	AL
4	01009	Blount	AL
...	...	...	...
3227	72151	Yabucoa	PR
3228	72153	Yauco	PR
3229	78010	St. Croix	VI
3230	78020	St. John	VI
3231	78030	St. Thomas	VI

3232 rows × 3 columns

state_code

	state	code
0	Alabama	AL
1	Alaska	AK
2	Arizona	AZ
3	Arkansas	AR
4	California	CA
5	Colorado	CO
6	Connecticut	CT
7	Delaware	DE
8	Florida	FL
9	Georgia	GA
10	Hawaii	HI
11	Idaho	ID
12	Illinois	IL

Let’s install plotly library and let’s get started

- `pip install plotly`

Please do not use explorer for visualization.

If you are, then set your default browser to chrome!

https://stackoverflow.com/questions/47772157/how-to-change-the-default-browser-used-by-jupyter-notebook-in-windows

Let’s launch our Jupyter notebook