Final project - Stephani Ceballos

Description of the data

In this dataset we can see the daily count of COVID 19 cases, hospitalizations and deaths in New York City. In the variables we can observe the day, death count, hospitalizations and cases, there is also a probable number of the same issues, this is because there may be cases that were not counted due to lack of information, etc. There are also variables in which we can observe the same theme but obtained weekly.

All variables are numeric, discrete, only the date is a string, ordinal.

Important variables:

- 'CASE_COUNT': display the daily count of covid 19 cases in New York City. Is a numeric variable, discreet.
- 'HOSPITALIZED_COUNT': display the daily count of hospitalization due to covid 19 in New York City. Is a numeric variable, discreet.
- 'DEATH_COUNT': display the daily count of deaths due to covid 19 in New York City. Is a numeric variable, discreet.
- 'DATE_OF_INTEREST': display the date in which the count was taken in New York City. Is a string variable, ordinal.

In [17]:

```
df.head(2)
```

Out[17]:

	DATE_OF_INTEREST	CASE_COUNT	HOSPITALIZED_COUNT	DEATH_COUNT
0	02/29/2020	1	1	0
1	03/01/20	0	1	0

Interesting observations

- There was a huge wave of this pandemic situation in which, New York City reach more than 50,000 positive cases of Covid-19 per day.

In [7]:

```
df.loc[:, 'CASE_COUNT'].max()
```

Out[7]:

```
54767
```

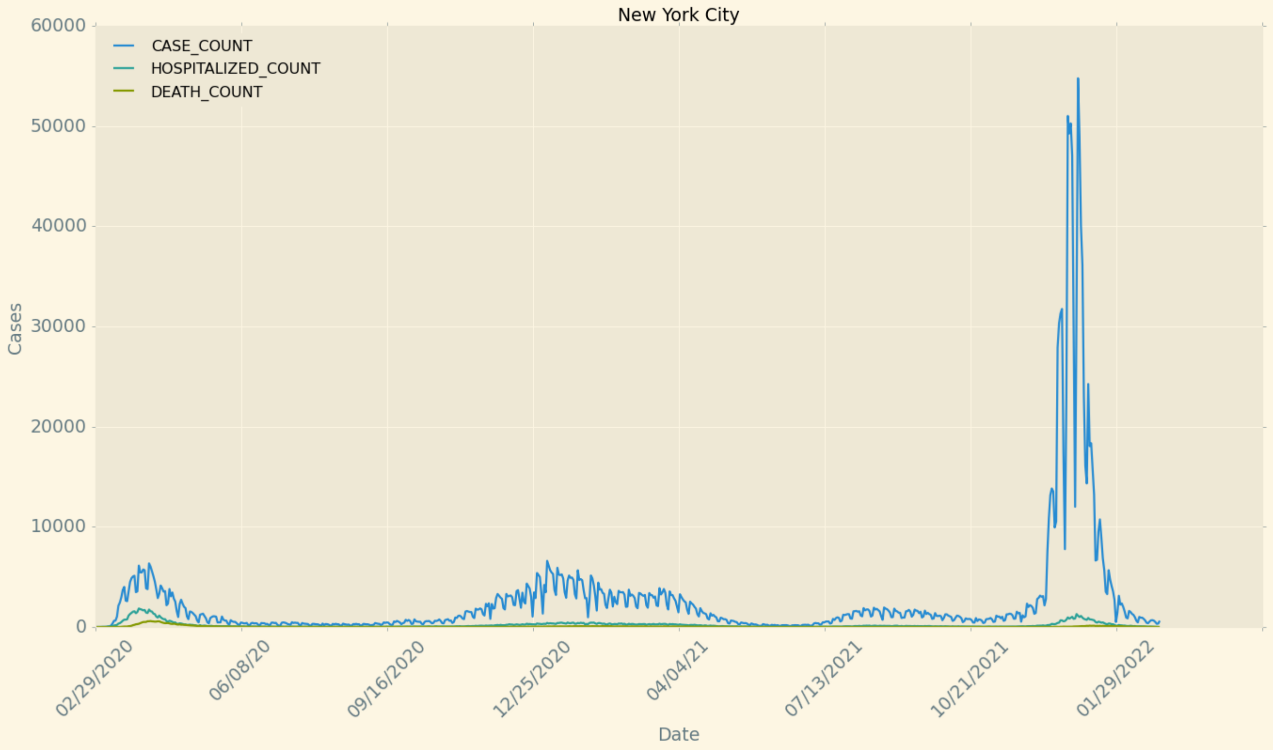
Conclusion: it was really a day in which the cases of covid 19 over passed the 50,000 cases

- The hugest wave of cases of covid 19 took place on march of 2020

In [37]:

```
df.plot(figsize = (20,10))
plt.xlabel('Date', fontsize = 17)
plt.xticks(fontsize = 17, rotation = 45)
plt.ylabel('Cases', fontsize = 17)
plt.yticks(fontsize = 17)
plt.title('New York City', fontsize = 17)
```

```
Out[37]: Text(0.5, 1.0, 'New York City')
```



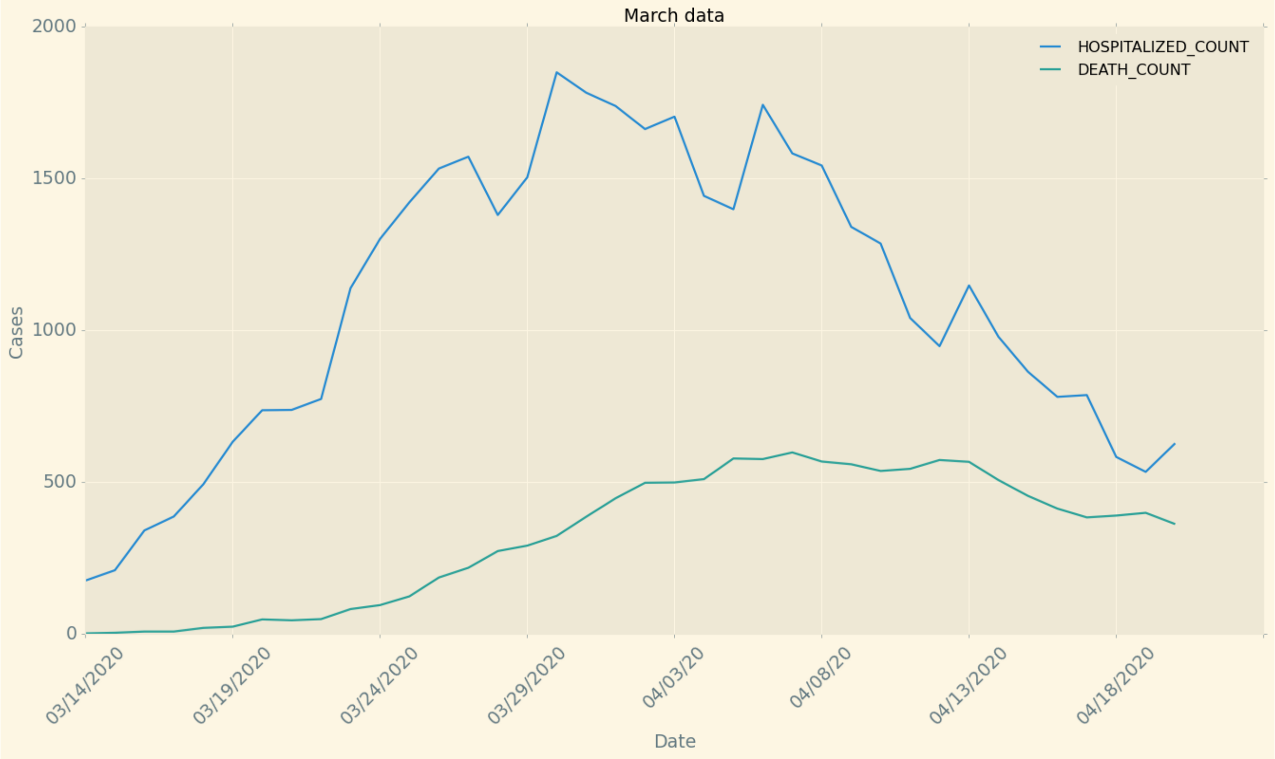
Conclusion: I graphed the database of the daily count and I realize that actually, the highest wave of covid 19 was on the end of the last year, it was greater even than how it was on march of 2020

Graphical visualization

1. How was the behavior of the cases of covid 19 on march 2020?

```
In [61]:
mar_data = df.iloc[14:52,1:]
mar_data.plot(figsize = (20,10))
plt.xlabel('Date', fontsize = 17)
plt.xticks(fontsize = 17, rotation = 45)
plt.ylabel('Cases', fontsize = 17)
plt.yticks(fontsize = 17)
plt.title('March data', fontsize = 17)
```

```
Out[61]: Text(0.5, 1.0, 'March data')
```



Conclusion: Actually, we can observe that the cases started incresing by the end of march 2020, but, they also started to decrease by the beginnig of april, so, the real increase of cases started by the end of the month. We made an hypothesis about that march of 2020 was the month with higher covid 19 cases, but with this graph we can reject the hypothesis, because, actually it started at the end of this month.

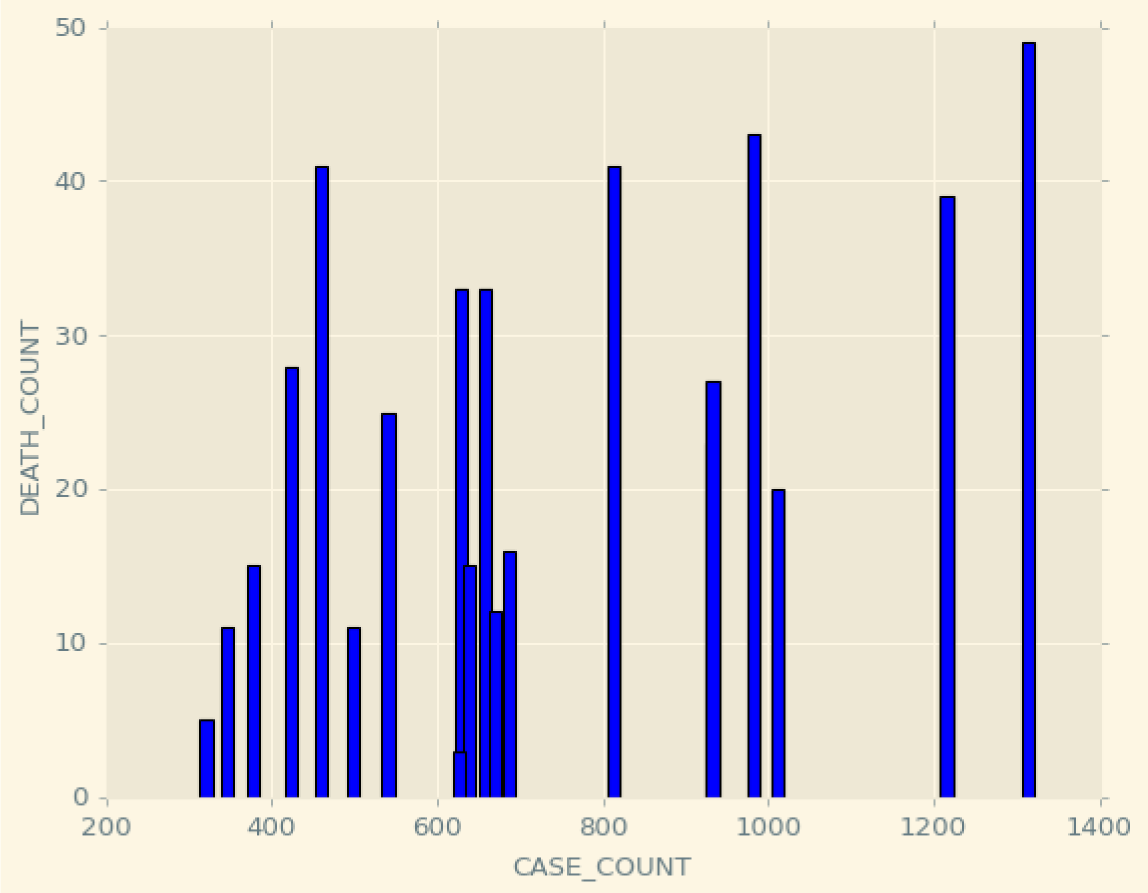
1. What's the relation between the cases of covid 19 and deaths due to covid 19?

In [98]:

```
CASE_COUNT = df['CASE_COUNT'].tail(20)
DEATH_COUNT = df['DEATH_COUNT'].tail(20)

plt.xlabel('CASE_COUNT')
plt.ylabel('DEATH_COUNT')
plt.bar(CASE_COUNT, DEATH_COUNT, color='blue', width=15)

plt.show()
```



Conclusion: We can observe that actually, the relation between deaths and cases are very high, and this is because every certain amount of positive cases of covid 19, that would be related with a possible death due to covid 19.

Possible questions

1. What's the average of cases, deaths and hospitalizations due to covid 19?

In [13]:

```
df[['CASE_COUNT','DEATH_COUNT','HOSPITALIZED_COUNT']].describe()
```

Out[13]:

	CASE_COUNT	DEATH_COUNT	HOSPITALIZED_COUNT
count	731.000000	731.000000	731.000000
mean	2652.082079	46.898769	201.128591
std	6033.148680	90.812335	294.289132
min	0.000000	0.000000	1.000000
25%	412.500000	7.000000	42.000000
50%	1057.000000	14.000000	84.000000
75%	2790.000000	52.000000	261.500000
max	54767.000000	598.000000	1850.000000

Conclusion: the average of cases of covid 19 count daily in New York City is 2652.082079, of deaths due to covid 19 is 46.898769 and of hospitalizations is 201.128591

1. How can we observe the amount of deaths that are caused by the different amount of positive cases of covid 19?

In [103…

```
pd.pivot_table(df,
 index = ['CASE_COUNT'],
 values = ['DEATH_COUNT'],
 margins = True
 )
```

Out[103…

	DEATH_COUNT
CASE_COUNT	
0	0.000000
1	0.000000
3	0.000000
5	0.000000
7	0.000000
...	...
49265	39.000000
50266	44.000000
51008	35.000000
54767	66.000000
All	46.898769

654 rows × 1 columns

Conclusion: actually, we can observe how each day that reach a certain quantity of positve cases a amount of deaths due to covid 19 per day just in New York City, we can observe and prove how the medical assistance there is pretty good, because even when they reach a high amount of cases, they stayed on a very low amount of deaths.

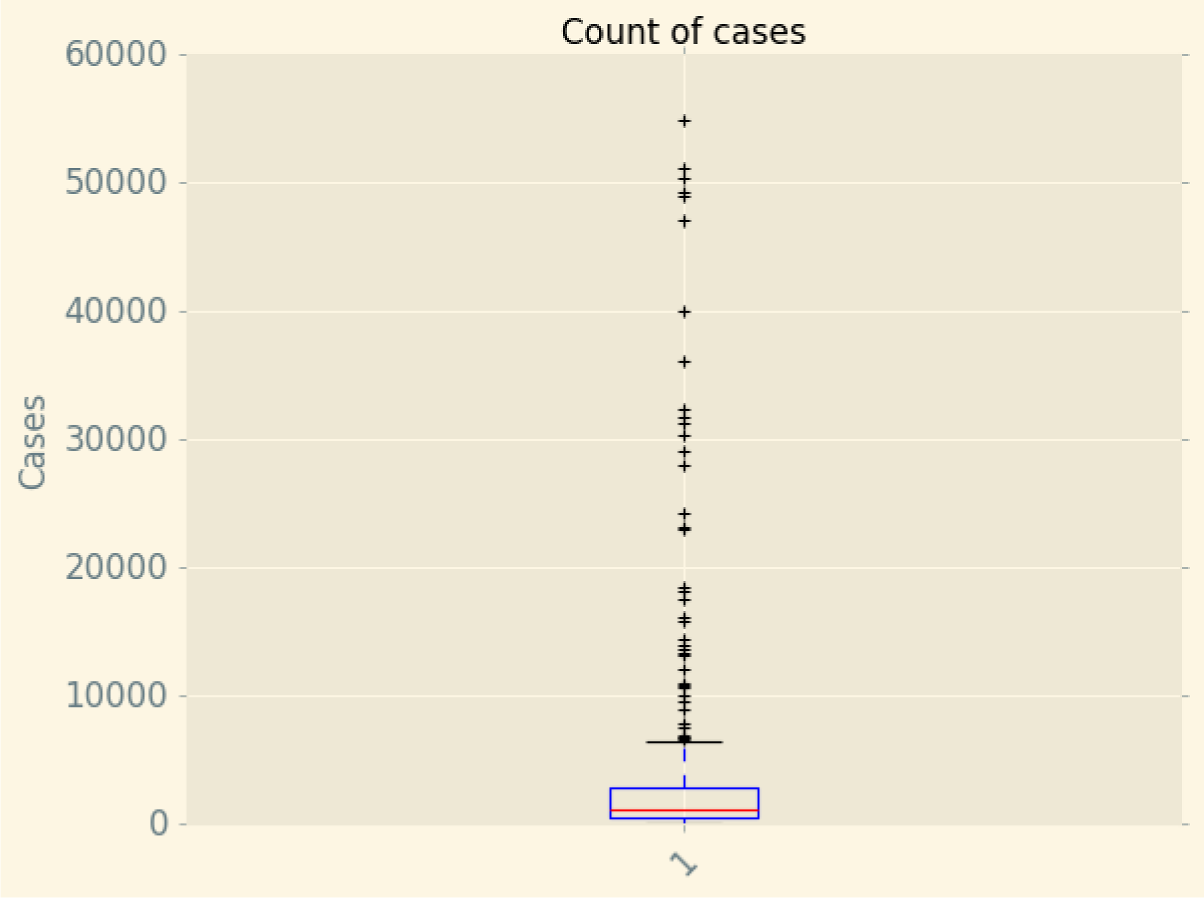
1. Is there any atypical value within the amount of cases per day?

In [113…

```
x = df.loc[:, 'CASE_COUNT']
plt.xticks(fontsize = 15, rotation = 45)
plt.ylabel('Cases', fontsize = 15)
plt.yticks(fontsize = 15)
plt.title('Count of cases', fontsize = 15)
plt.boxplot(x)
```

Out[113…

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f09f76b8710>,
  <matplotlib.lines.Line2D at 0x7f09f76c7d10>],
 'caps': [<matplotlib.lines.Line2D at 0x7f09f76f8550>,
  <matplotlib.lines.Line2D at 0x7f09f76cf7d0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f09f76c72d0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f09f76cfd50>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f09f76d82d0>],
 'means': []}
```



```
In [114…  df['CASE_COUNT'].describe()
```

```
Out[114…  count      731.000000
          mean      2652.082079
          std       6033.148680
          min          0.000000
          25%        412.500000
          50%       1057.000000
          75%       2790.000000
          max      54767.000000
          Name: CASE_COUNT, dtype: float64
```

Conclusion: we can observe that, by the 2790 and so on, the values obtained were atypical values, so, the 'normal' was from 412.5 to 2790 cases of covid 19 per day on New York City.

Problems I faced

- The first problem that I faced was that my dataset was very huge, so I had to drop many columns that just didn't add important value.
- Another problem was that I wasn't able to upload the whole data because of the amount of information.

Next steps to take

1. I will make shorter the dataset
2. I will look for more possible analysis for my data

In [88]:

```
import pandas as pd
import numpy as np
```

In [6]:

```
df = pd.read_csv("datasets/COVID-19_Daily_Counts_of_Cases__Hospitalizations__and
df.head(2)
```

Out[6]:

	DATE_OF_INTEREST	CASE_COUNT	probable_case_count	HOSPITALIZED_COUNT	DEATH_COUI
0	02/29/2020	1	0	1	
1	03/01/20	0	0	1	

2 rows × 62 columns

In [16]:

```
df.drop(['probable_case_count','CASE_COUNT_7DAY_AVG', 'all_case_count_7day_avg',
        'DEATH_COUNT_7DAY_AVG', 'all_death_count_7day_avg', 'BX_CASE_COUNT',
        'bx_probable_case_count', 'BX_HOSPITALIZED_COUNT', 'BX_DEATH_COUNT',
        'bx_probable_death_count', 'BX_CASE_COUNT_7DAY_AVG',
        'bx_all_case_count_7day_avg', 'BX_HOSPITALIZED_COUNT_7DAY_AVG',
        'BX_DEATH_COUNT_7DAY_AVG', 'bx_all_death_count_7day_avg',
        'BK_CASE_COUNT', 'bk_probable_case_count', 'BK_HOSPITALIZED_COUNT',
        'BK_DEATH_COUNT', 'bk_probable_death_count', 'BK_CASE_COUNT_7DAY_AVG',
        'bk_all_case_count_7day_avg', 'BK_HOSPITALIZED_COUNT_7DAY_AVG',
        'BK_DEATH_COUNT_7DAY_AVG', 'bk_all_death_count_7day_avg',
        'MN_CASE_COUNT', 'mn_probable_case_count', 'MN_HOSPITALIZED_COUNT',
        'MN_DEATH_COUNT', 'mn_probable_death_count', 'MN_CASE_COUNT_7DAY_AVG',
        'mn_all_case_count_7day_avg', 'MN_HOSPITALIZED_COUNT_7DAY_AVG',
        'MN_DEATH_COUNT_7DAY_AVG', 'mn_all_death_count_7day_avg',
        'QN_CASE_COUNT', 'qn_probable_case_count', 'QN_HOSPITALIZED_COUNT',
        'QN_DEATH_COUNT', 'qn_probable_death_count', 'QN_CASE_COUNT_7DAY_AVG',
        'qn_all_case_count_7day_avg', 'QN_HOSPITALIZED_COUNT_7DAY_AVG',
        'QN_DEATH_COUNT_7DAY_AVG', 'qn_all_death_count_7day_avg',
        'SI_CASE_COUNT', 'si_probable_case_count', 'SI_HOSPITALIZED_COUNT',
        'SI_DEATH_COUNT', 'si_probable_death_count', 'SI_CASE_COUNT_7DAY_AVG',
        'si_all_case_count_7day_avg', 'SI_HOSPITALIZED_COUNT_7DAY_AVG',
        'SI_DEATH_COUNT_7DAY_AVG', 'si_all_death_count_7day_avg', 'INCOMPLETE','D
         axis = 1,
         inplace = True
        )
```

In [18]:

```
mpl.style.use('classic')
```

In [19]:

```
print(plt.style.available)
```

```
['Solarize_Light2', '_classic_test_patch', 'bmh', 'classic', 'dark_background',
'fast', 'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn', 'seaborn-bright',
'seaborn-colorblind', 'seaborn-dark', 'seaborn-dark-palette', 'seaborn-darkgri
d', 'seaborn-deep', 'seaborn-muted', 'seaborn-notebook', 'seaborn-paper', 'seabo
rn-pastel', 'seaborn-poster', 'seaborn-talk', 'seaborn-ticks', 'seaborn-white',
'seaborn-whitegrid', 'tableau-colorblind10']
```

In [21]:

```
import matplotlib as mpl
import matplotlib.pyplot as plt
```

In [27]:

```
df.columns = list(map(str, df.columns))
df.columns
#df.loc([:, 'DEATH_COUNT'],:'630').plot
```

Out[27]:

```
Index(['DATE_OF_INTEREST', 'CASE_COUNT', 'HOSPITALIZED_COUNT', 'DEATH_COUNT'], d
type='object')
```

In [28]:

```
df.set_index('DATE_OF_INTEREST', inplace = True)
df.head()
```

Out[28]:

	CASE_COUNT	HOSPITALIZED_COUNT	DEATH_COUNT
DATE_OF_INTEREST			
02/29/2020	1	1	0
03/01/20	0	1	0
03/02/20	0	2	0
03/03/20	1	7	0
03/04/20	5	2	0

In [32]:

```
#df.columns = list(map(str, df.columns))
df.columns
```

Out[32]:

```
Index(['CASE_COUNT', 'HOSPITALIZED_COUNT', 'DEATH_COUNT'], dtype='object')
```

In [33]:

```
import matplotlib as mpl
import matplotlib.pyplot as plt
```

In [34]:

```
mpl.style.use('Solarize_Light2')
print(plt.style.available)
```

```
['Solarize_Light2', '_classic_test_patch', 'bmh', 'classic', 'dark_background',
'fast', 'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn', 'seaborn-bright',
'seaborn-colorblind', 'seaborn-dark', 'seaborn-dark-palette', 'seaborn-darkgri
d', 'seaborn-deep', 'seaborn-muted', 'seaborn-notebook', 'seaborn-paper', 'seabo
rn-pastel', 'seaborn-poster', 'seaborn-talk', 'seaborn-ticks', 'seaborn-white',
'seaborn-whitegrid', 'tableau-colorblind10']
```

In [49]:

```
pip install seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.11.2-py3-none-any.whl (292 kB)
     |████████████████████████████████| 292 kB 4.8 MB/s
Requirement already satisfied: pandas>=0.23 in /srv/conda/envs/notebook/lib/pyth
on3.7/site-packages (from seaborn) (1.0.4)
Requirement already satisfied: scipy>=1.0 in /srv/conda/envs/notebook/lib/python
3.7/site-packages (from seaborn) (1.4.1)
```

```
Requirement already satisfied: numpy>=1.15 in /srv/conda/envs/notebook/lib/pytho
n3.7/site-packages (from seaborn) (1.18.5)
Requirement already satisfied: matplotlib>=2.2 in /srv/conda/envs/notebook/lib/p
ython3.7/site-packages (from seaborn) (3.2.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /srv/
conda/envs/notebook/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn)
(2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in /srv/conda/envs/notebook/
lib/python3.7/site-packages (from matplotlib>=2.2->seaborn) (2.8.1)
Requirement already satisfied: cycler>=0.10 in /srv/conda/envs/notebook/lib/pyth
on3.7/site-packages (from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /srv/conda/envs/notebook/li
b/python3.7/site-packages (from matplotlib>=2.2->seaborn) (1.2.0)
Requirement already satisfied: pytz>=2017.2 in /srv/conda/envs/notebook/lib/pyth
on3.7/site-packages (from pandas>=0.23->seaborn) (2020.1)
Requirement already satisfied: six in /srv/conda/envs/notebook/lib/python3.7/sit
e-packages (from cycler>=0.10->matplotlib>=2.2->seaborn) (1.16.0)
Installing collected packages: seaborn
Successfully installed seaborn-0.11.2
Note: you may need to restart the kernel to use updated packages.
```

In [50]:

```python
import seaborn as sns
```