Data Science with Python Course 2 - Week 1

Dr. Ha Nguyen
Ambyint - Canada

Week 1 - Agenda

1. Instructors, Students, Syllabus, Policy
2. Scikit-Learn
3. Datasets
4. Data Visualization

Week 1 - Agenda

1. Instructors, Students, Syllabus, Policy
2. Scikit-Learn
3. Datasets
4. Data Visualization

Instructors - Students

1. Students
    a. Introduction
    b. Expectation
2. Instructors
    a. Introduction

Syllabus - Policy

1. Course Syllabus
2. Homeworks
    a. 7 homeworks: due 1 hour before the next class
3. Submission: Github/Gitlab/Bitbucket

Programming Tools

Data Analysts vs Engineers vs Scientists

Data Analysts deliver value to their companies by taking data, using it to answer questions, and communicating the results to help make business decisions. Common tasks done by data analysts include data cleaning, performing analysis and creating data visualizations.

Data Engineers build and optimize the systems that allow data scientists and analysts to perform their work. Every company depends on its data to be accurate and accessible to individuals who need to work with it. The data engineer ensures that any data is properly received, transformed, stored, and made accessible to other users.

https://www.dataquest.io/blog/data-analyst-data-scientist-data-engineer/

Data Analysts vs Engineers vs Scientists

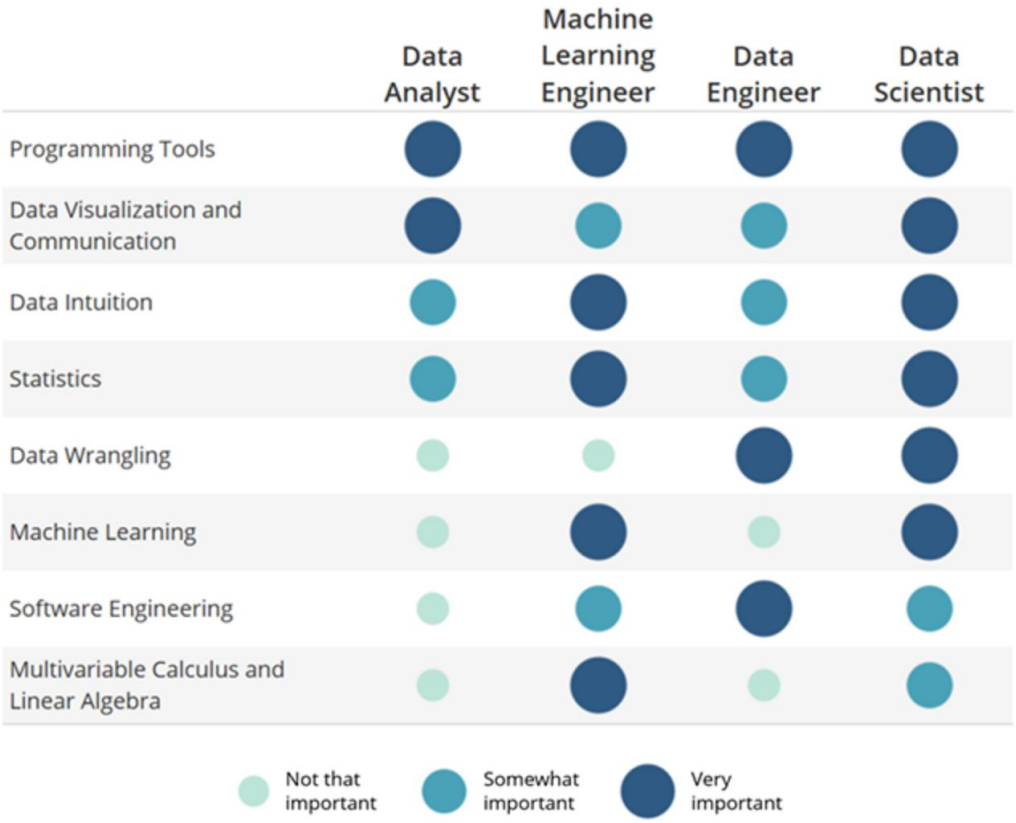
The **Data Scientist** is an individual that can provide immense value by tackling more open-ended questions and leveraging their knowledge of advanced statistics and algorithms. If the analyst focuses on understanding data from the past and present perspectives, then the scientist focuses on producing reliable predictions for the future.

The following are examples of work performed by **data scientists**:

- Evaluating statistical models to determine the validity of analyses.
- Using machine learning to build better predictive algorithms.
- Testing and continuously improving the accuracy of machine learning models.
- Building data visualizations to summarize the conclusion of an advanced analysis.

https://www.dataquest.io/blog/data-analyst-data-scientist-data-engineer/

Data Scientist



How do Data Science skills map to Data Science jobs?

Big Data - What it means to you?

https://youtu.be/-Gj93L2Qa6c

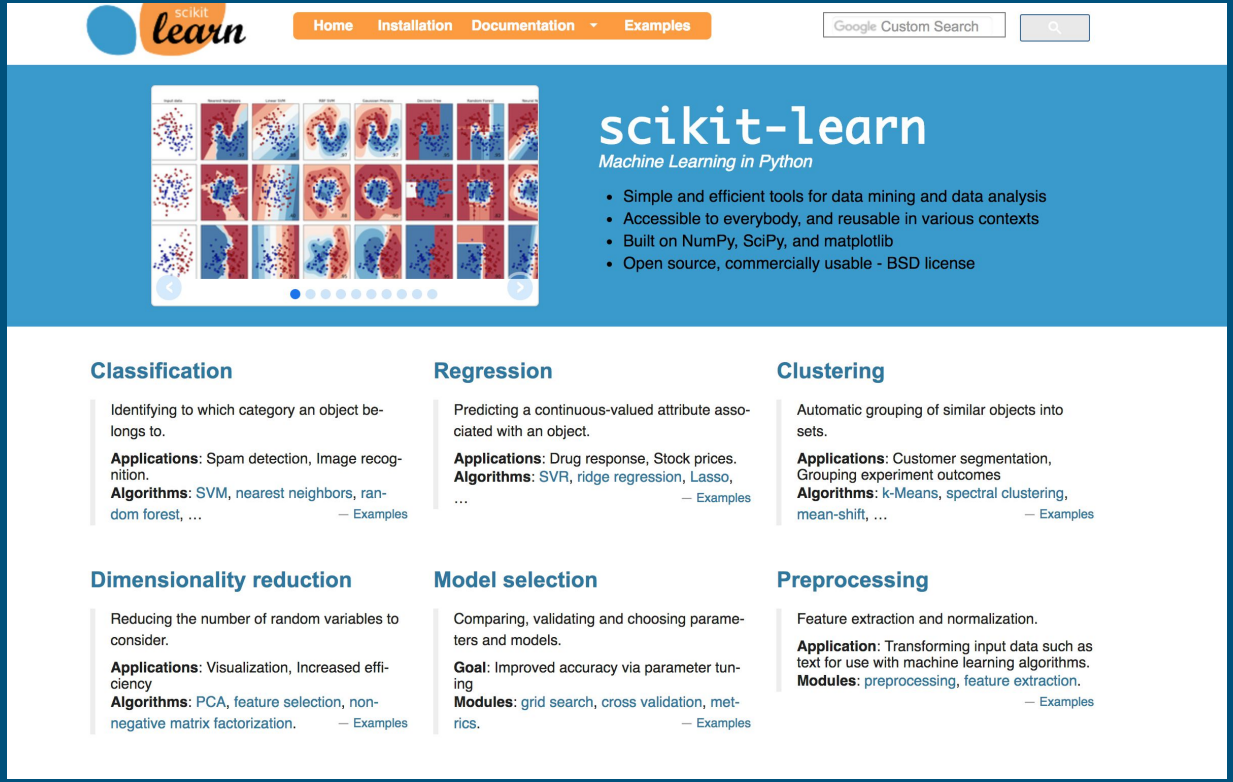
Big Data - Data Science - Data Analytics?

https://youtu.be/yR2wWQYiVKM

Week 1 - Agenda

1. Instructors, Students, Syllabus, Policy
2. Scikit-Learn
3. Datasets
4. Data Visualization

Scikit-Learn



Week 1 - Agenda

1. Instructors, Students, Syllabus, Policy
2. Scikit-Learn
3. Datasets
4. Data Visualization

Datasets

https://www.kaggle.com/datasets

https://www.dataquest.io/blog/free-datasets-for-projects/

https://www.quandl.com/

https://www.analyticsvidhya.com/blog/2018/05/24-ultimate-data-science-projects-to-boost-your-knowledge-and-skills/

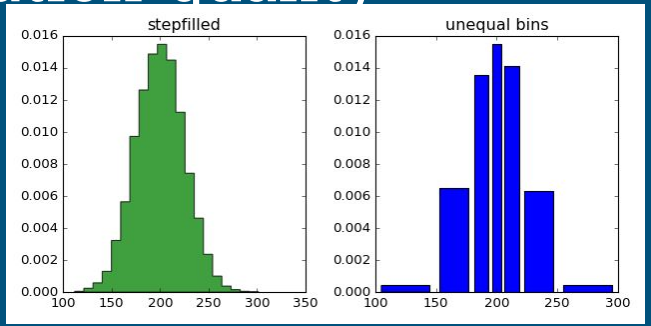
etc ...

Week 1 - Agenda

1. Instructors, Students, Syllabus, Policy
2. Scikit-Learn
3. Datasets
4. Data Visualization

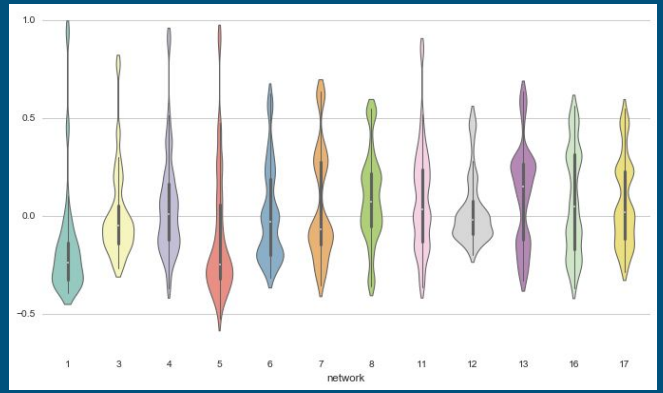
matplotlib

1. Initial release: 2003
2. Python data visualization tool
3. Designed to closely ensemble MATLAB
4. Many other tools built on top of it: pandas, Seaborn
5. Not very useful for creating publication-quality charts quickly and easily



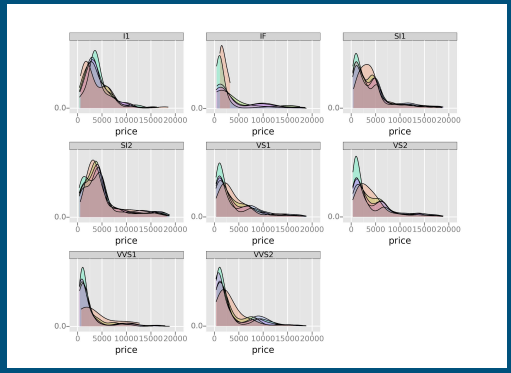
seaborn

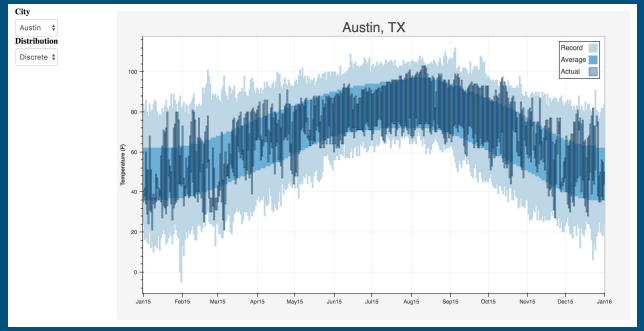
1. Built on top of matplotlib: “Or, as Michael Waskom says in the “Introduction to Seaborn”: “If matplotlib “tries to make easy things easy and hard things possible”, seaborn tries to make a well-defined set of hard things easy too.”
2. Harnesses the power of matplotlib to create beautiful charts in a few lines of code

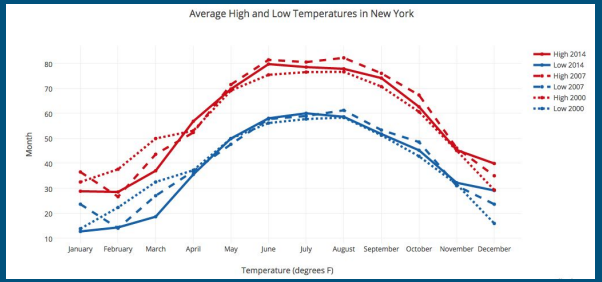


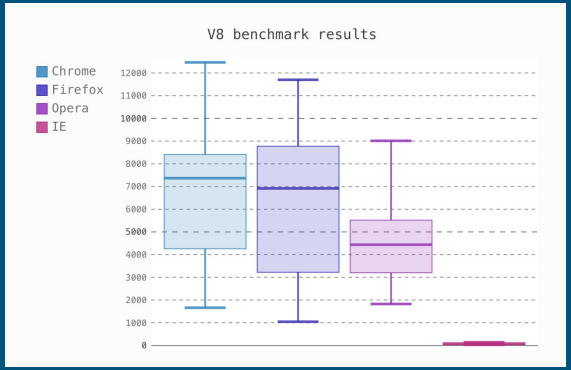
Other visualization tools

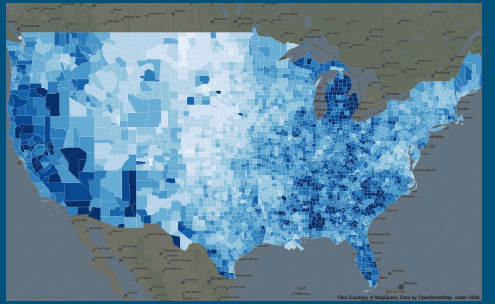
1. ggplot
2. Plotly
3. Geoplotlib
4. Bokeh
5. pygal
6. Gleam
7. missingno
8. Leather







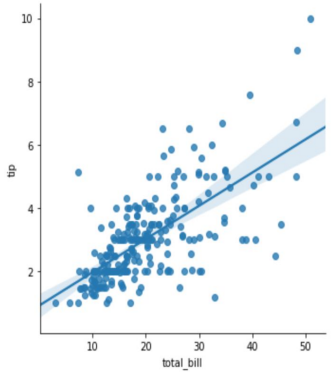




seaborn

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

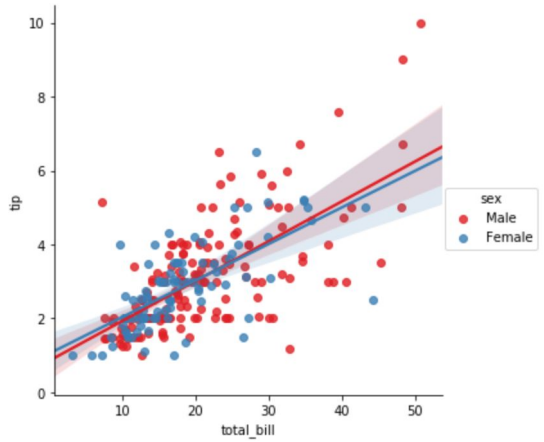
In [4]:
```
# Using lmplot
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#pd_tips = pd.read_csv('seaborn-data/tips.csv')
tips = sns.load_dataset('tips')
sns.lmplot(x= 'total_bill', y='tip', data=pd_tips)
plt.show()
```



Seaborn - grouping factor (same plot)

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

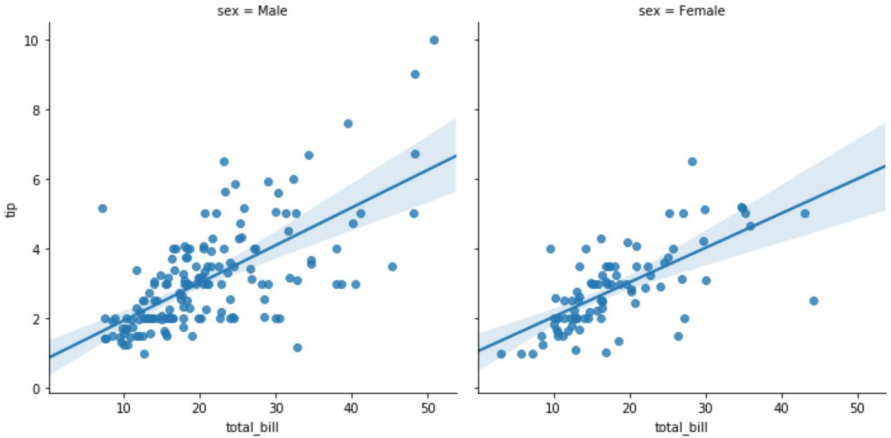
In [9]:
```
# grouping factor
sns.lmplot(x='total_bill', y='tip', data=tips, hue='sex', palette='Set1')
plt.show()
```



Seaborn - grouping factor (subplot)

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

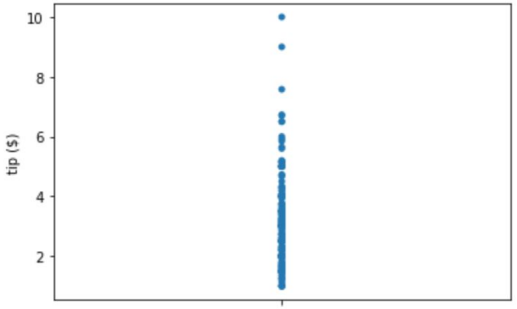
```
In [10]: # grouping factor (different plot)
         sns.lmplot(x='total_bill', y='tip', data=tips, col='sex')
         plt.show()
```



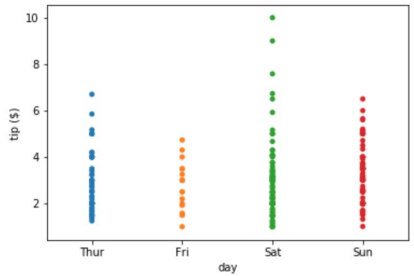
Seaborn - visualize univariate distributions

- Strip plots
- Swarm plots
- Violin plots

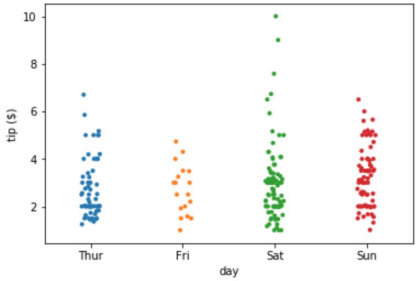
In [12]:
```
# strip plot
sns.stripplot(y= 'tip', data=tips)
plt.ylabel('tip ($)')
plt.show()
```



In [14]:
```
# grouping with strip plot
sns.stripplot(x='day', y='tip', data=tips)
plt.ylabel('tip ($)')
plt.show()
```



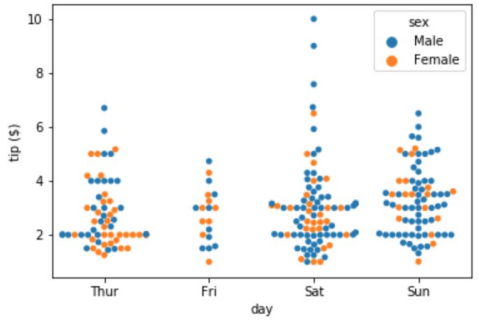
In [16]:
```
# spread out strip plot
sns.stripplot(x='day', y='tip', data=tips, size=4, jitter=True)
plt.ylabel('tip ($)')
plt.show()
```



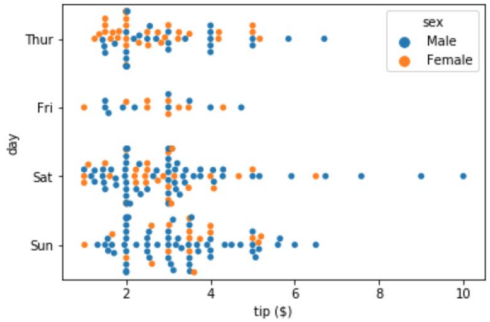
Seaborn - visualize univariate distributions

- Strip plots
- Swarm plots
- Violin plots

In [18]:
```
# swarm plot - grouping
sns.swarmplot(x='day', y='tip', data=tips, hue='sex')
plt.ylabel('tip ($)')
plt.show()
```



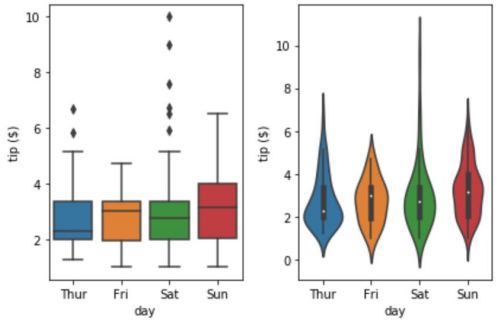
In [19]:
```
# swarm plot - grouping - orientation
sns.swarmplot(x='tip', y='day', data=tips, hue='sex', orient='h')
plt.xlabel('tip ($)')
plt.show()
```



Seaborn - visualize univariate distributions

- Strip plots
- Swarm plots
- Violin plots

In [20]:
```
# violin plots
plt.subplot(1,2,1)
sns.boxplot(x='day', y='tip', data=tips)
plt.ylabel('tip ($)')
plt.subplot(1,2,2)
sns.violinplot(x='day', y='tip', data=tips)
plt.ylabel('tip ($)')
plt.tight_layout()
plt.show()
```

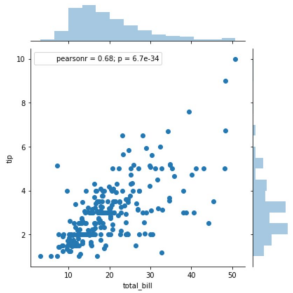


Seaborn - visualize multivariate distributions

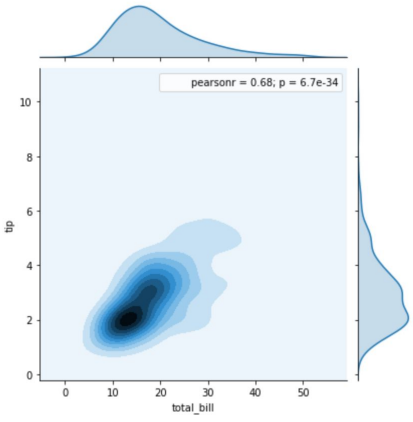
- Joint plots
- Pair plots
- Heat maps

```
In [21]: # visualize multivariate distributions
         # joint plots
         sns.jointplot(x= 'total_bill', y= 'tip', data=tips)
         plt.show()
```

```
/anaconda3/envs/cv-with-python-1/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning: The 'normed'
kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
/anaconda3/envs/cv-with-python-1/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning: The 'normed'
kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



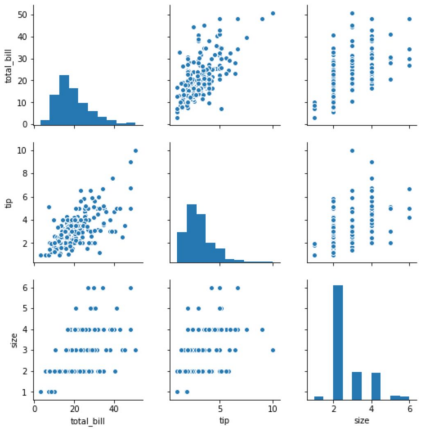

```
In [22]: # using kde
         sns.jointplot(x='total_bill', y= 'tip', data=tips, kind='kde')
         plt.show()
```



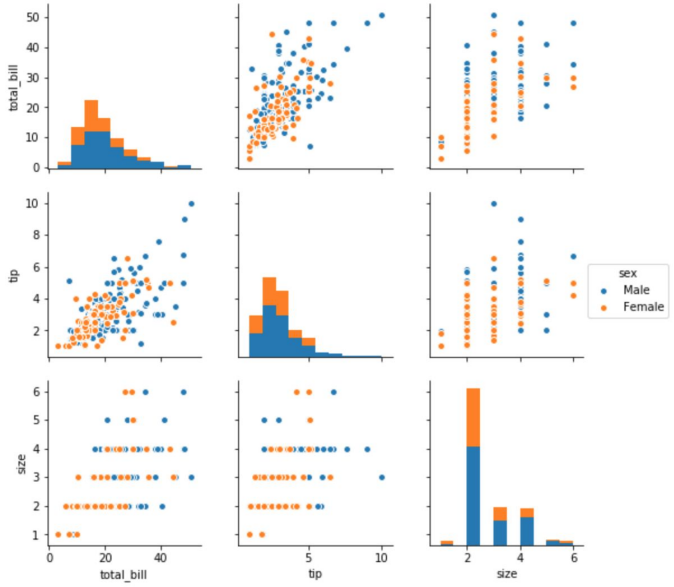
Seaborn - visualize multivariate distributions

- Joint plots
- Pair plots
- Heat maps

```
In [23]: # pair plot
sns.pairplot(tips)
plt.show()
```



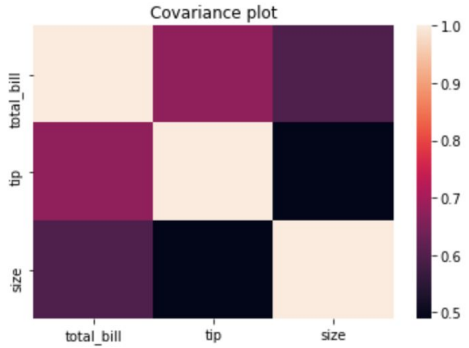

```
In [24]: # pair plot
sns.pairplot(tips, hue ='sex')
plt.show()
```



Seaborn - visualize multivariate distributions

- Joint plots
- Pair plots
- Heat maps

```
In [26]: # plot correlation using heatmap
         covariance = tips.corr()

In [27]: print(covariance)

                   total_bill       tip      size
total_bill           1.000000  0.675734  0.598315
tip                  0.675734  1.000000  0.489299
size                 0.598315  0.489299  1.000000

In [28]: sns.heatmap(covariance)
         plt.title('Covariance plot')
         plt.show()
```

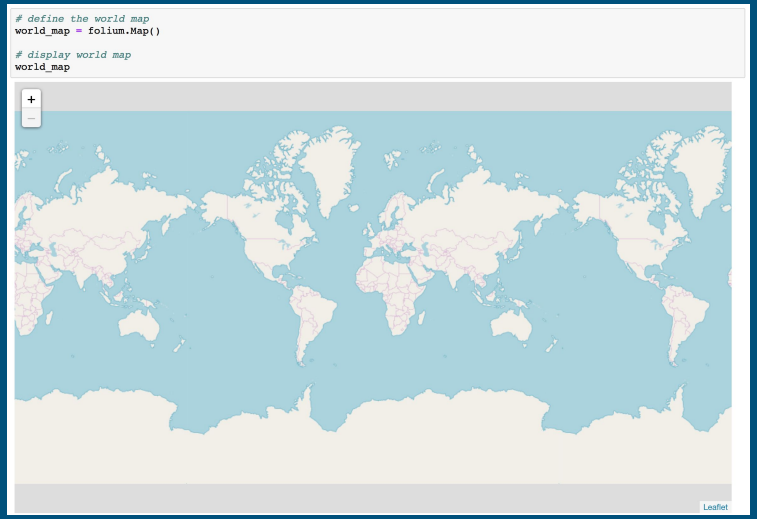


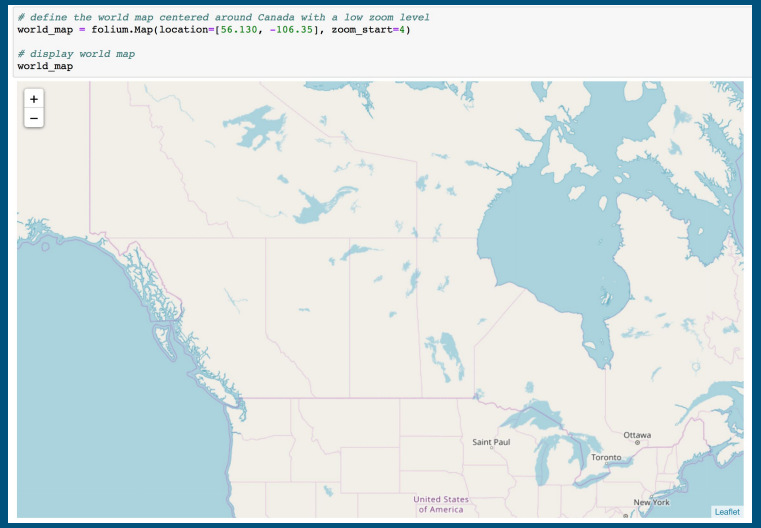
Creating Maps & Visualizing Geospatial Data

1. Visualization Tools
    a. Folium
    b. Maps & Markers
    c. Choropleth Maps

Folium

- Folium is a powerful data visualization library in Python that was built primarily to help people visualize geospatial data.

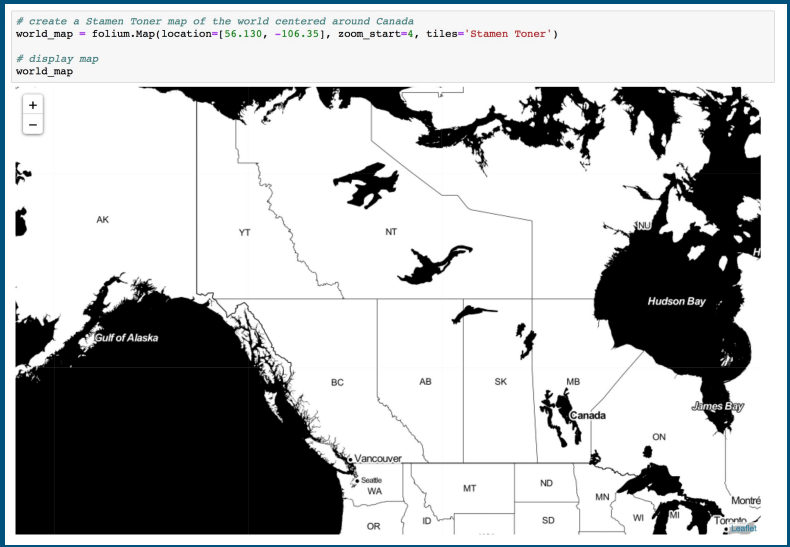




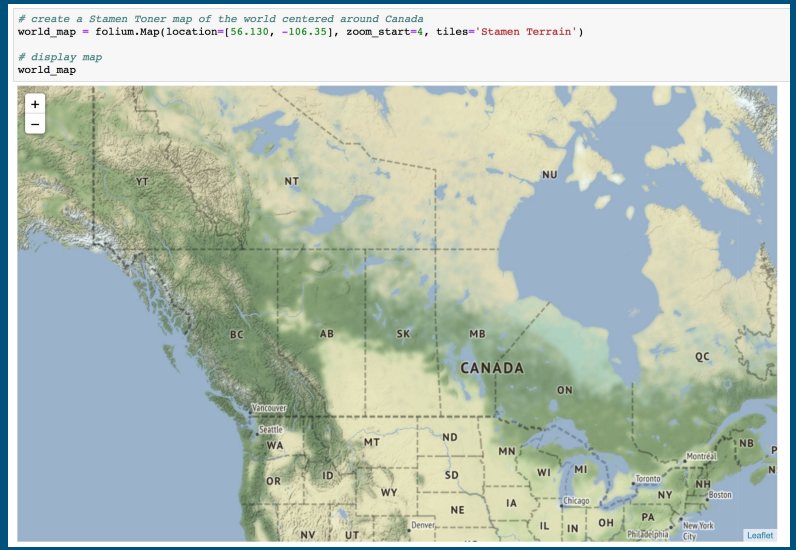
Folium

- Folium is a powerful data visualization library in Python that was built primarily to help people visualize geospatial data.

```
# create a Stamen Toner map of the world centered around Canada
world_map = folium.Map(location=[56.130, -106.35], zoom_start=4, tiles='Stamen Toner')

# display map
world_map
```



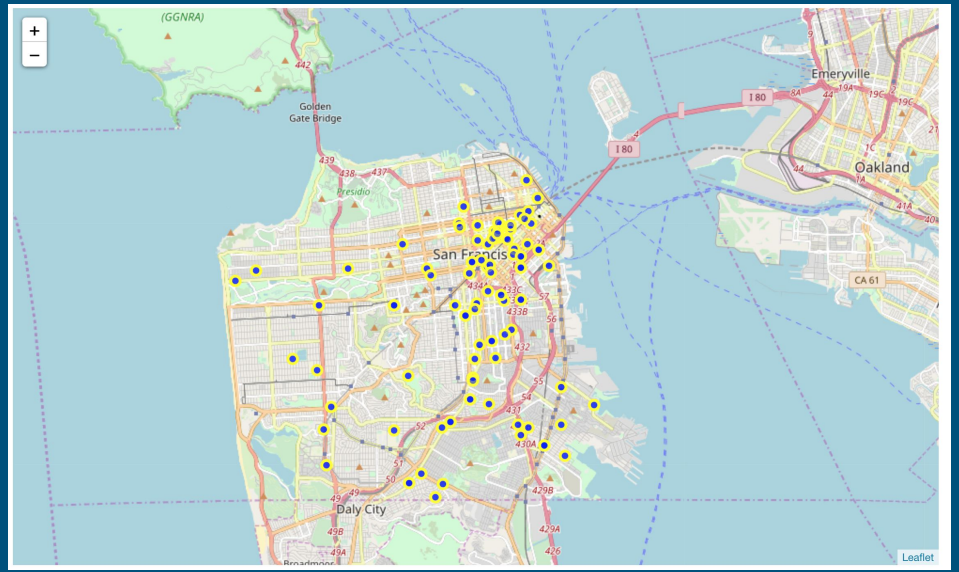

```
# create a Stamen Toner map of the world centered around Canada
world_map = folium.Map(location=[56.130, -106.35], zoom_start=4, tiles='Stamen Terrain')

# display map
world_map
```



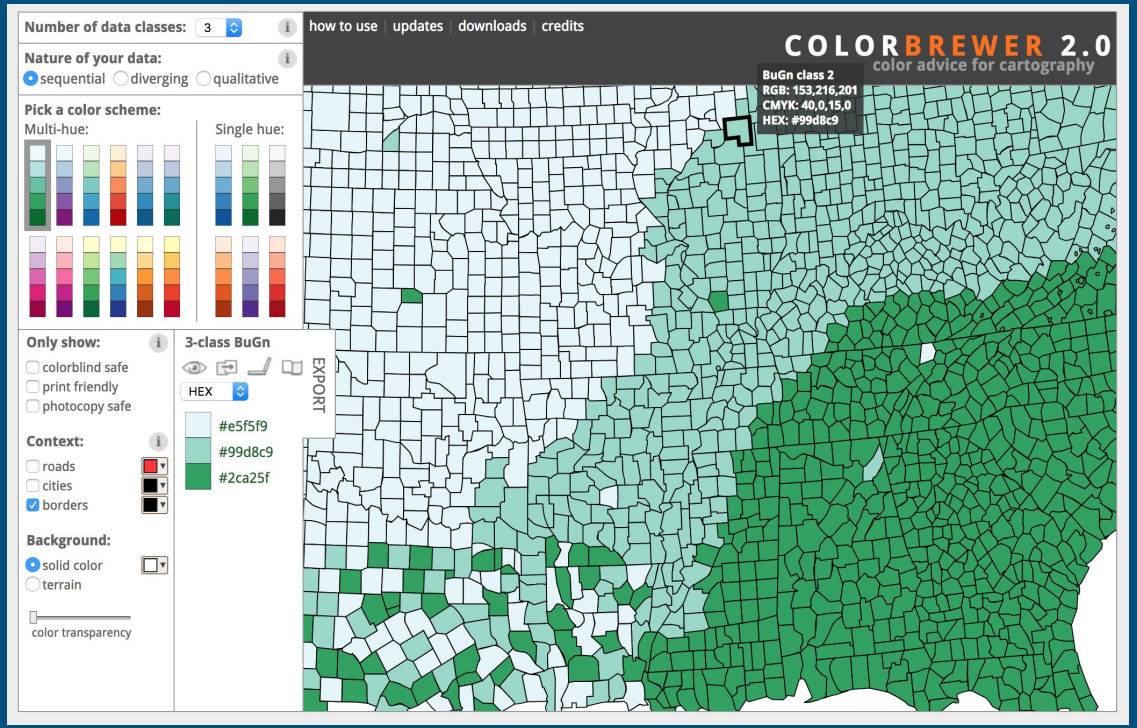
Maps & Markers

Work with the Folium library and learn how to superimpose markers on top of a map for interesting visualizations.

```
# San Francisco latitude and longitude values
latitude = 37.77
longitude = -122.42

# create map and display it
sanfran_map = folium.Map(location=[latitude, longitude], zoom_start=12)

# display the map of San Francisco
sanfran_map
```

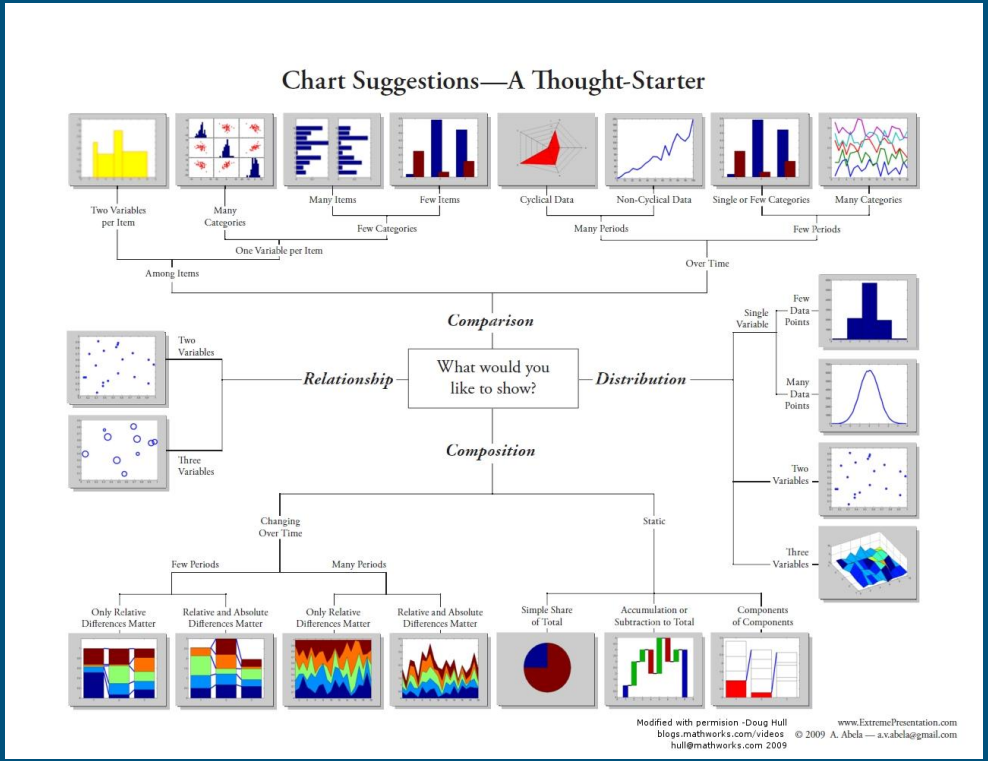


Color Choice



http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=

Chart Selection



https://eazybi.com/blog/data_visualization_and_chart_types/

Google Colab

https://colab.research.google.com

Next Week

1. Statistical Techniques in Data Science

Reference

1. Introduction to Data Visualization with Python (DataCamp)
2. https://www.statsmodels.org/stable/index.html
3. https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9
4. https://altair-viz.github.io/
5. https://bokeh.pydata.org/en/latest/docs/dev_guide/bokehjs.html

Thank you