Introduction to Data Visualization and Web Scraping

https://github.com/kaopanboonyuen/GISTDA2023

Outline

- Part 0: Basic Python and Pandas
- Part 1: Data Visualization using Google Data Studio
- Part 2: Web Scraping in Python With BeautifulSoup and Twitter Scraping

All python codes/notebooks/slides will be posted here:

https://github.com/kaopanboonyuen/GISTDA2023

Part 0: Basic Python and Pandas

Module 0: Basic Python and Pandas

- Python Recap: [CO Open in Colab]
- Pandas: [CO Open in Colab]

Part 1: Data Visualization using Google Data Studio
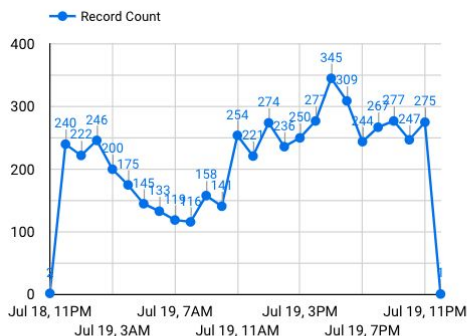
Module 1: Google Data Studio (Looker Studio): https://lookerstudio.google.com/

- Disaster Tweets (Data Set):

  https://github.com/kaopanboonyuen/GISTDA2023/raw/main/dataset/visualize/disaster_text.csv
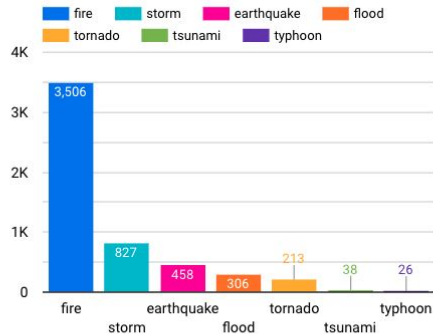
- Med Resource (Data Set):

  https://github.com/kaopanboonyuen/GISTDA2023/raw/main/dataset/visualize/med_resources_text.csv
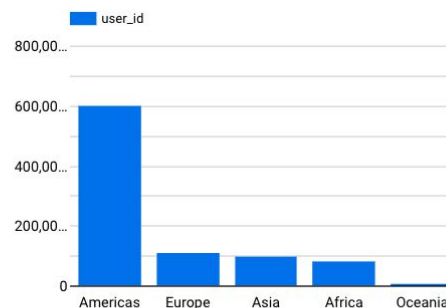
Disastor Monitoring Dashboard

Tweet by Datetime
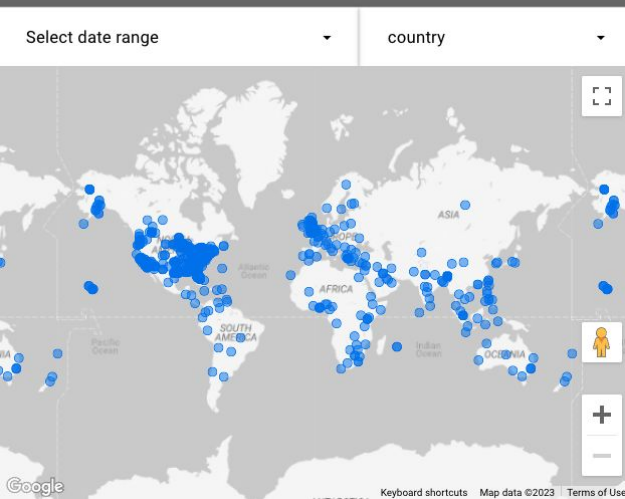

Record Count

Tweet by DisasterType

fire   storm   earthquake   flood   tornado   tsunami   typhoon

fire	3,506
storm	827
earthquake	458
flood	306
tornado	213
tsunami	38
typhoon	26

Tweet by Region

user_id

Tweet by Location

Select date range ▼   country ▼

Most Retweeted Tweets

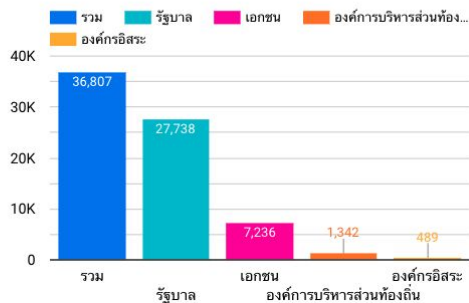|	clean_text	retweet_count ▼
1.	ghana 🇬🇭🇬🇭🇬🇭 you should be proud of the shatt...	2,202
2.	join hands for helping the flood victims of assam a...	1,354
3.	heavy police and fire presence at 4400 holden in we...	1,027
4.	amazing video shot of the tornado east of carmang...	982
5.	m5.3 earthquake σεισμός strikes 23 km nw of athe...	905
6.	excited to finally announce that i am joining the fire...	647
7.	finally after all the stop and starts delays unforesee...	590
8.	11 37pm fireworks then tear gas deployed puertoric...	434
9.	m5.1 earthquake σεισμός strikes 23 km nw of athe...	372
10.	if unsealepstein is "the happening" wouldn't that be ...	243
11.	m5.3 earthquake σεισμός strikes 23 km nw of athe...	180
12.	found our biggest offset yet for the fault that cause...	169

1 - 100 / 5055

Thailand Physician Resource
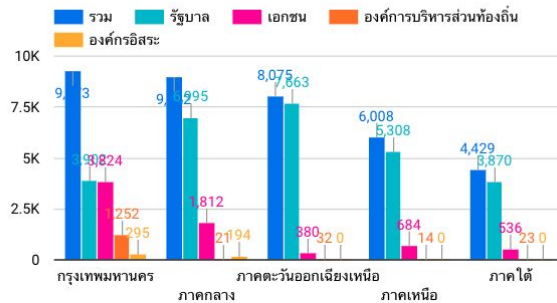
#Total Physicians

จำนวนแพทย์
36,807

จำนวนสถานพยาบาล
1,271

จำนวนเตียง
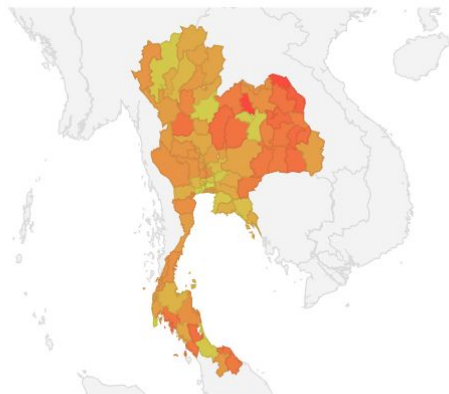**149,641**

#Physician by Department

Legend: รวม | รัฐบาล | เอกชน | องค์การบริหารส่วนท้อง... | องค์กรอิสระ

- รวม: 36,807
- รัฐบาล: 27,738
- เอกชน: 7,236
- องค์การบริหารส่วนท้องถิ่น: 1,342
- องค์กรอิสระ: 489

#Physician by Region

Legend: รวม | รัฐบาล | เอกชน | องค์การบริหารส่วนท้องถิ่น | องค์กรอิสระ

- กรุงเทพมหานคร: 9,?3 | 3,?? | 3,824 | 1,252 | 295
- ภาคกลาง: 9,?295 | 1,812 | 21?94
- ภาคตะวันออกเฉียงเหนือ: 8,075 | 7,863 | 380 | 32 0
- ภาคเหนือ: 6,008 | 5,308 | 684 | 14 0
- ภาคใต้: 4,429 | 3,870 | 536 | 23 0

#Population per Physician

ภาค ▼   จังหวัด ▼

601 ▨ 4,965

#Highest Population Per Physician

จังหวัด	สัดส่วนประชากรต่อแพทย์ ❶ ❷ ▼
5. กาฬสินธุ์	3,731
6. เพชรบูรณ์	3,719
7. ศรีสะเกษ	3,697
8. พัทลุง	3,669
9. อำนาจเจริญ	3,666

1 - 77 / 77 ‹ ›

#Lowest Population Per Physician

จังหวัด	สัดส่วนประชากรต่อแพทย์ ▲
1. กรุงเทพมหานคร	601
2. ภูเก็ต	901
3. สมุทรสาคร	1,069
4. พิษณุโลก	1,086

1 - 77 / 77 ‹ ›

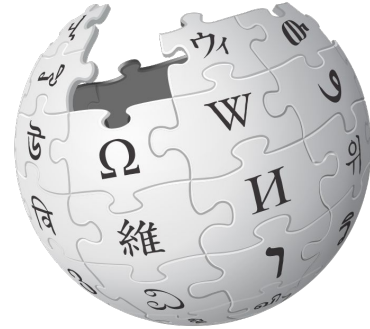# Part 2: Web Scraping in Python With BeautifulSoup and Twitter Scraping

Module 2: Web Scraping and Twitter Scraping

- Web Scraping: **CO** Open in Colab
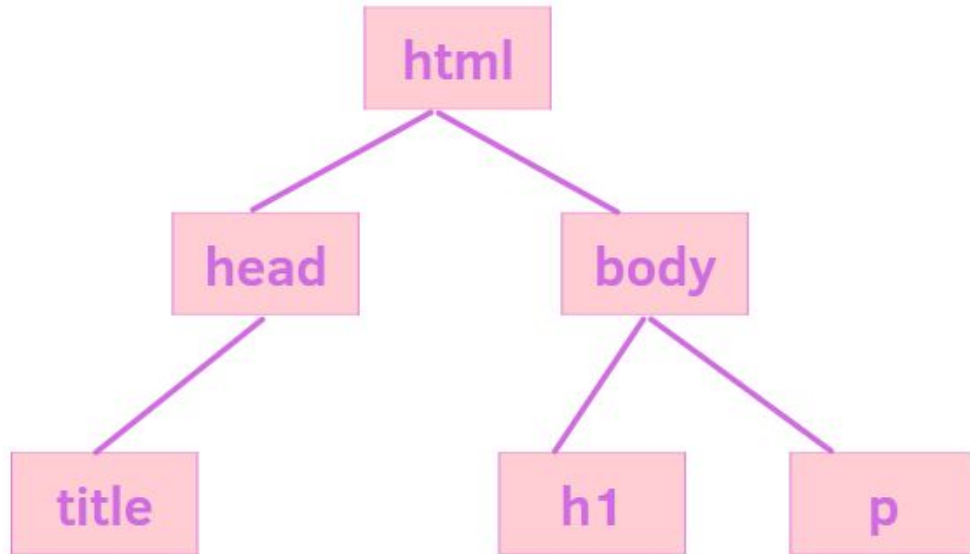- Twitter Scraping: **CO** Open in Colab

Seaborn: Statistical Data Visualization

Seaborn helps to visualize the statistical relationships, To understand how variables in a dataset are related to one another and how that relationship is dependent on other variables, we perform statistical analysis.

This Statistical analysis helps to visualize the trends and identify various patterns in the dataset.

- Line Plot
- Scatter Plot
- Box plot
- Point plot
- Count plot
- Violin plot
- Swarm plot
- Bar plot
- KDE Plot

https://www.geeksforgeeks.org/data-visualization-with-python-seaborn/

```
1 !pip install -q seaborn
```

```
1 # load the csv
2 data = pd.read_csv("https://github.com/kaopanboonyuen/GISTDA2023/raw/main/dataset/visualize/nba.csv")
3
4 # show first 5 column
5 data.head()
```

|	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6796117.0
2	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0

Line plot:

Lineplot Is the most popular plot to draw a relationship between x and y with the possibility of several semantic groupings.

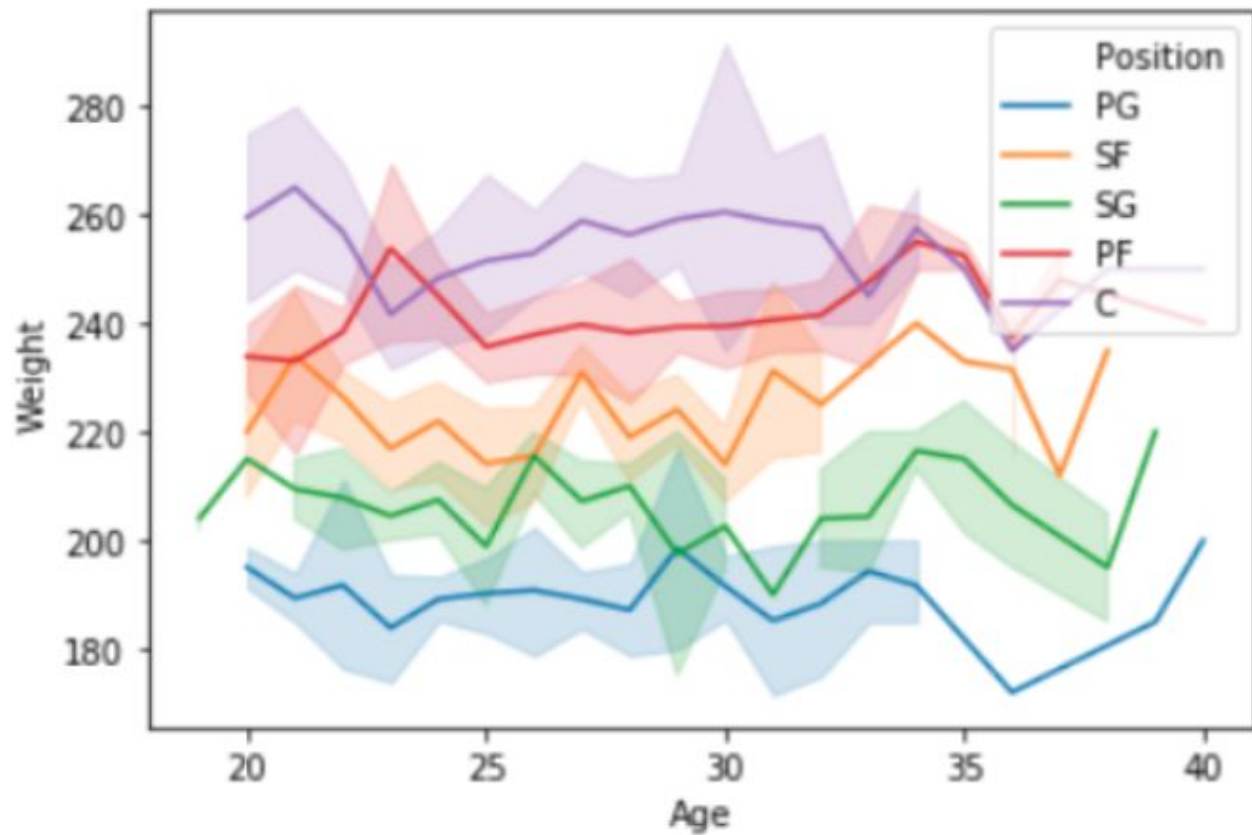> **Syntax :** *sns.lineplot(x=None, y=None)*
>
> **Parameters:**
>
> **x, y:** *Input data variables; must be numeric. Can pass data directly or reference columns in data.*

Output:
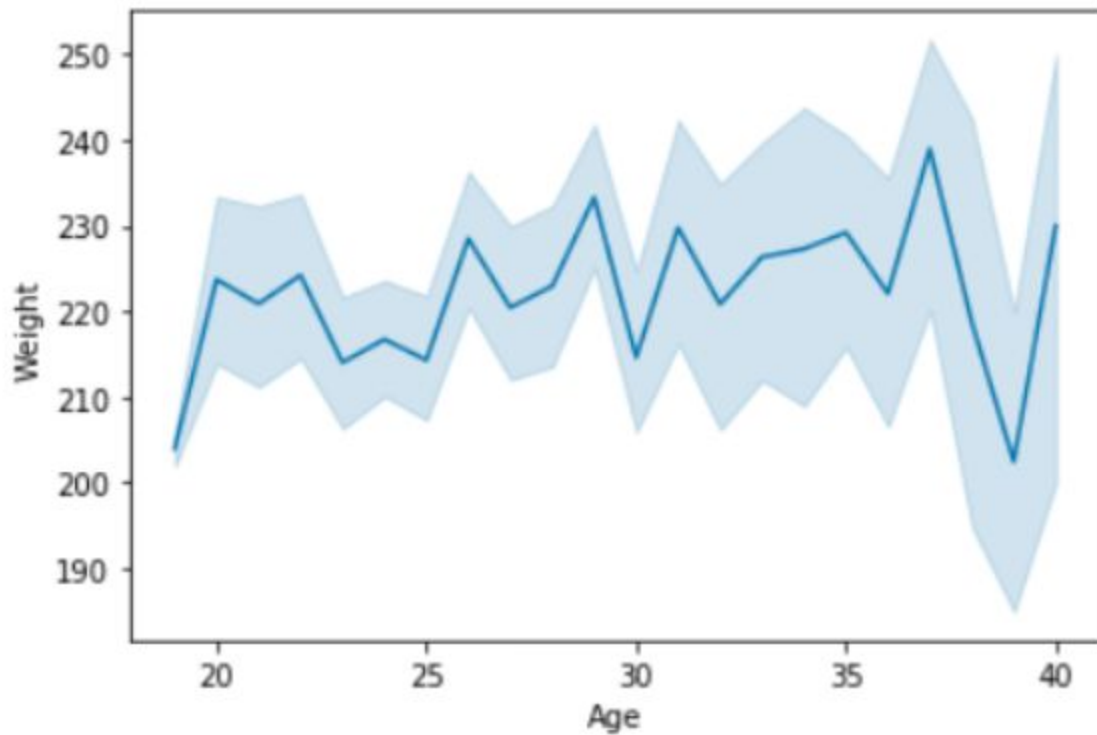
Output:

Scatter Plot:

[Scatterplot](#) Can be used with several semantic groupings which can help to understand well in a graph against continuous/categorical data. It can draw a two-dimensional graph.
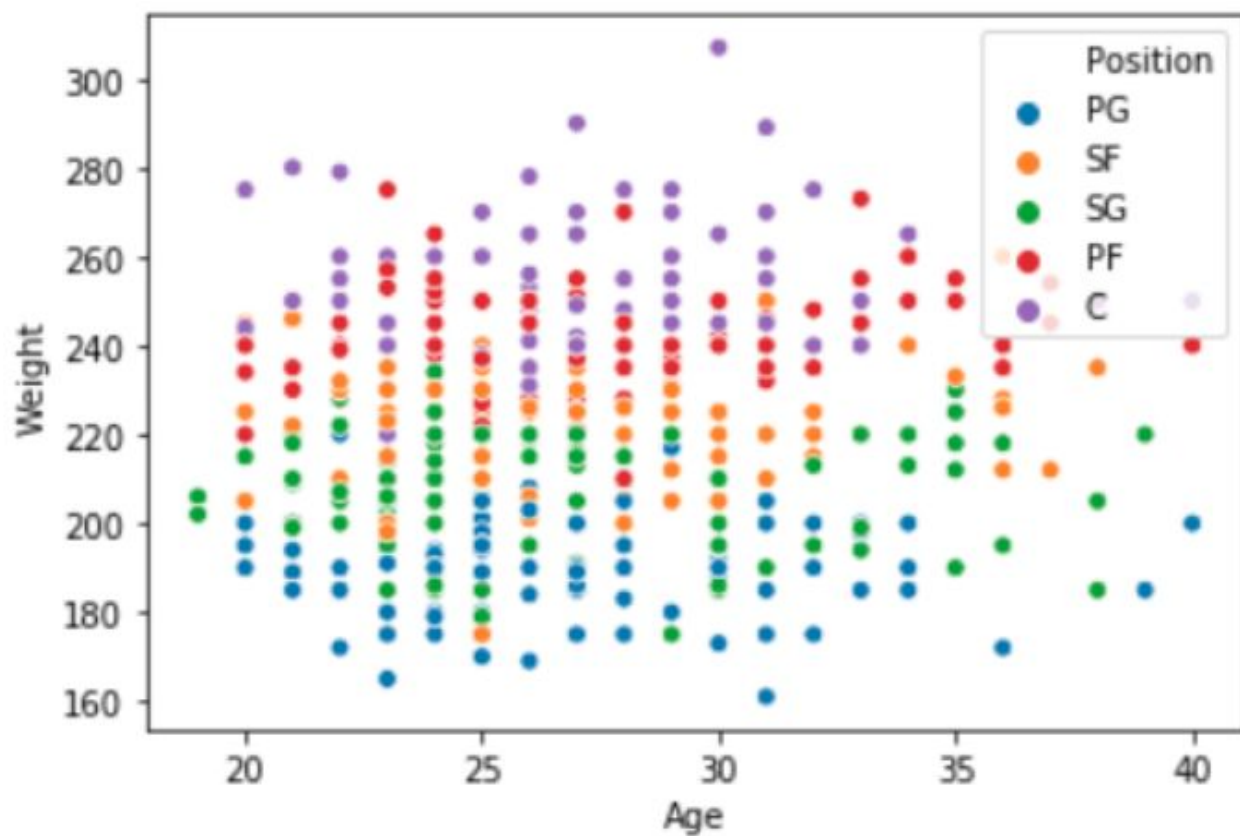
> ***Syntax:*** *seaborn.scatterplot(x=None, y=None)*
>
> ***Parameters:***
>
> ***x, y:*** *Input data variables that should be numeric.*
>
> ***Returns:*** *This method returns the Axes object with the plot drawn onto it.*

Output:

A box plot (or box-and-whisker plot) s is the visual representation of the depicting groups of numerical data through their quartiles against continuous/categorical data.

A box plot consists of 5 things.

- Minimum
- First Quartile or 25%
- Median (Second Quartile) or 50%
- Third Quartile or 75%
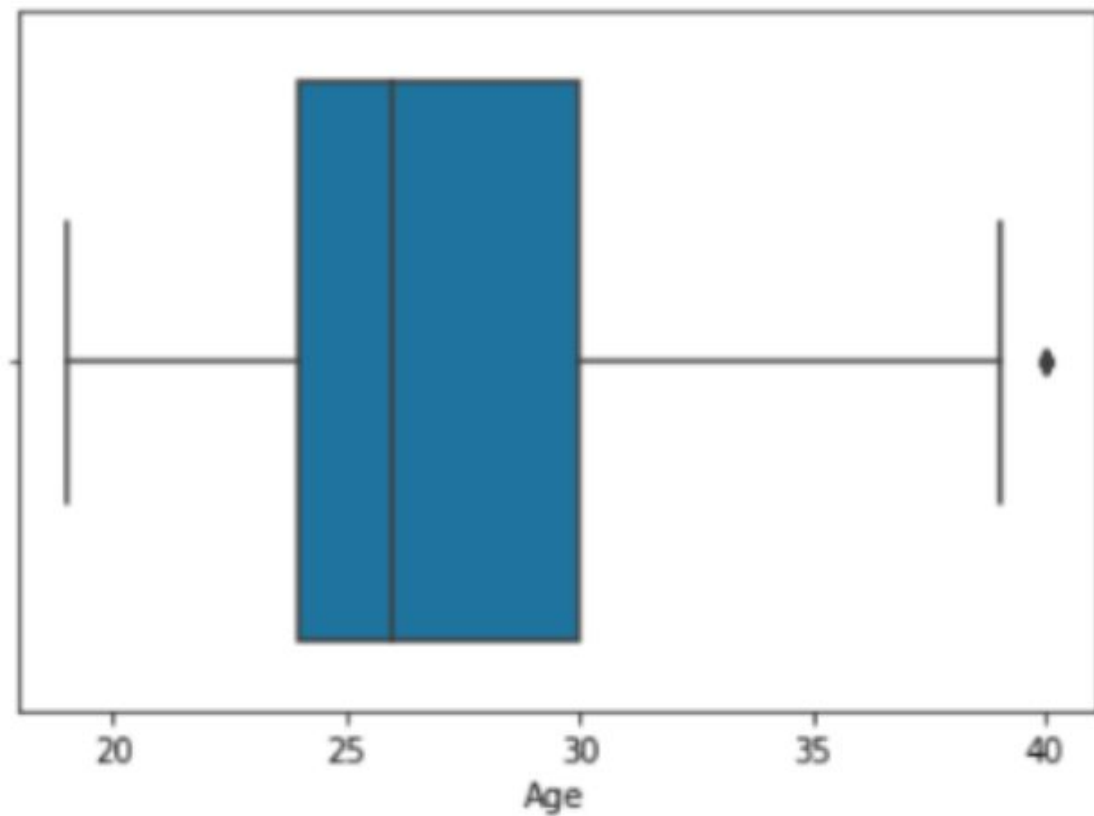- Maximum

Syntax:

seaborn.boxplot(x=None, y=None, hue=None, data=None)

Parameters:

- *x, y, hue:* *Inputs for plotting long-form data.*
- *data:* *Dataset for plotting. If x and y are absent, this is interpreted as wide-form.*

Returns: *It returns the Axes object with the plot drawn onto it.*
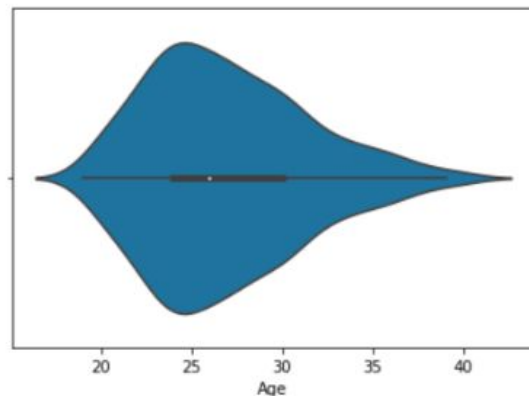
Output:

Violin Plot:

A violin plot is similar to a boxplot. It shows several quantitative data across one or more categorical variables such that those distributions can be compared.

Syntax: *seaborn.violinplot(x=None, y=None, hue=None, data=None)*

Parameters:

- **x, y, hue:** *Inputs for plotting long-form data.*
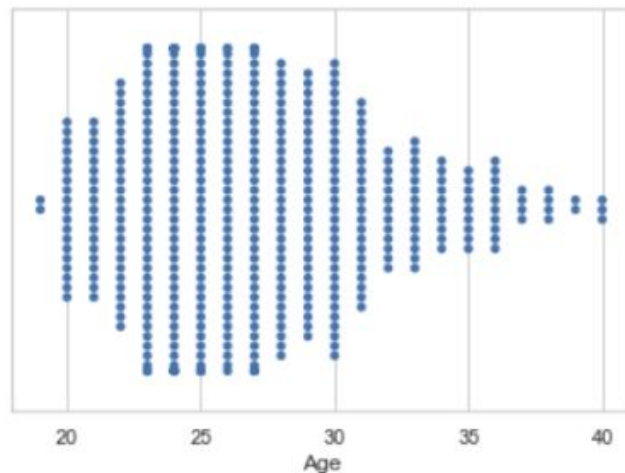- **data:** *Dataset for plotting.*

Output:

Swarm plot:

A swarm plot is similar to a strip plot, We can draw a swarm plot with non-overlapping points against categorical data.

Syntax: *seaborn.swarmplot(x=None, y=None, hue=None, data=None)*

Parameters:

- **x, y, hue:** *Inputs for plotting long-form data.*
- **data:** *Dataset for plotting.*

Output:

Bar plot:

[Barplot](#) represents an estimate of central tendency for a numeric variable with the height of each rectangle and provides some indication of the uncertainty around that estimate using error bars.
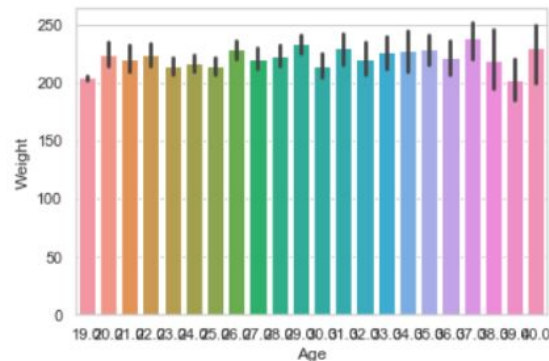
Syntax : *seaborn.barplot(x=None, y=None, hue=None, data=None)*

Parameters :

- **x, y :** *This parameter take names of variables in data or vector data, Inputs for plotting long-form data.*
- **hue :** *(optional) This parameter take column name for colour encoding.*
- **data :** *(optional) This parameter take DataFrame, array, or list of arrays, Dataset for plotting. If x and y are absent, this is interpreted as wide-form. Otherwise it is expected to be long-form.*

Returns : *Returns the Axes object with the plot drawn onto it.*

Output:

Point plot:

Point plot used to show point estimates and confidence intervals using scatter plot glyphs. A point plot represents an estimate of central tendency for a numeric variable by the position of scatter plot points and provides some indication of the uncertainty around that estimate using error bars.
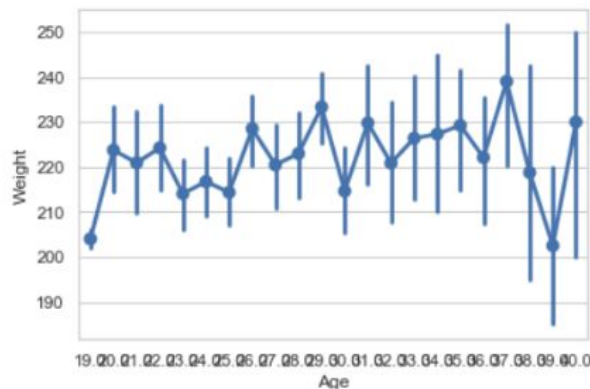
Syntax: *seaborn.pointplot(x=None, y=None, hue=None, data=None)*

Parameters:

- **x, y:** *Inputs for plotting long-form data.*
- **hue:** *(optional) column name for color encoding.*
- **data:** *dataframe as a Dataset for plotting.*

*Return:* *The Axes object with the plot drawn onto it.*

Output:

Count plot:

Count plot used to Show the counts of observations in each categorical bin using bars.

Syntax : *seaborn.countplot(x=None, y=None, hue=None, data=None)*

Parameters :

- **x, y:** *This parameter take names of variables in data or vector data, optional, Inputs for plotting long-form data.*
- **hue :** *(optional) This parameter take column name for color encoding.*
- **data :** *(optional) This parameter take DataFrame, array, or list of arrays, Dataset for plotting. If x and y are absent, this is interpreted as wide-form. Otherwise, it is expected to be long-form.*

Returns: *Returns the Axes object with the plot drawn onto it.*
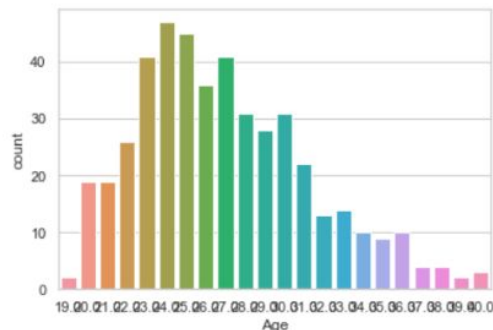
KDE Plot:

[KDE Plot](#) described as **Kernel Density Estimate** is used for visualizing the Probability Density of a continuous variable. It depicts the probability density at different values in a continuous variable. We can also plot a single graph for multiple samples which helps in more efficient data visualization.

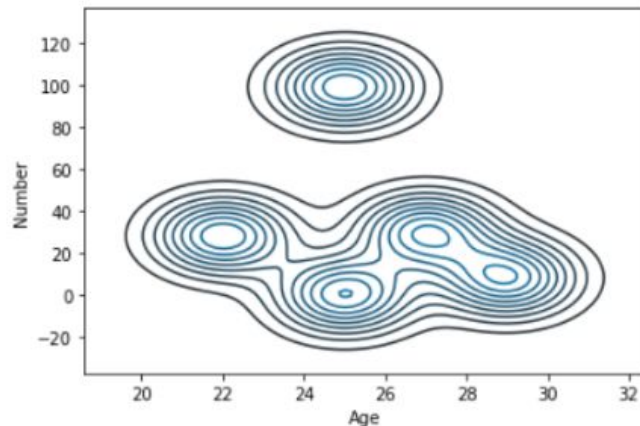**Syntax:** *seaborn.kdeplot(x=None, *, y=None, vertical=False, palette=None, **kwargs)*

Parameters:

x, y : *vectors or keys in data*

vertical : *boolean (True or False)*

data : *pandas.DataFrame, numpy.ndarray, mapping, or sequence*

Output:

Pair Plot