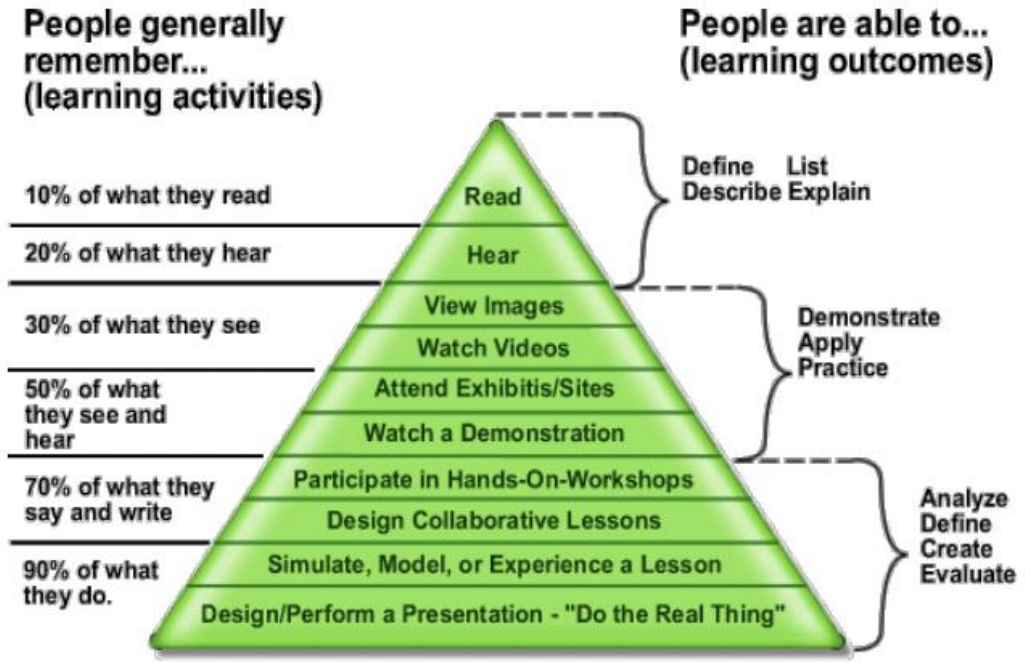
Session 10:
Exploratory Data Analysis

▶Presented by : Dr. Huynh Vo Trung Dung

What is Data Visualization?

Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from



Source: Wikipedia

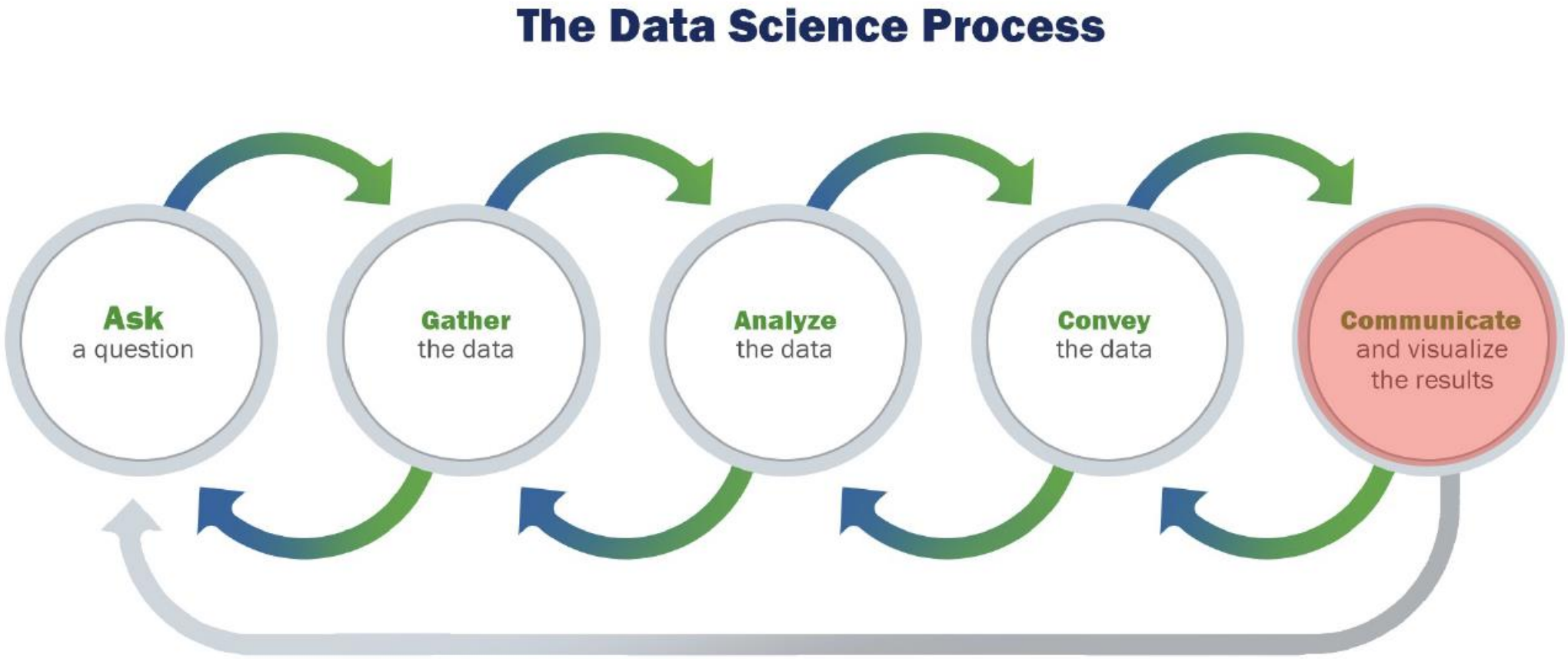
What is Data Visualization?

Data visualization translates complex data sets into visual formats that are easier for the human brain to comprehend.

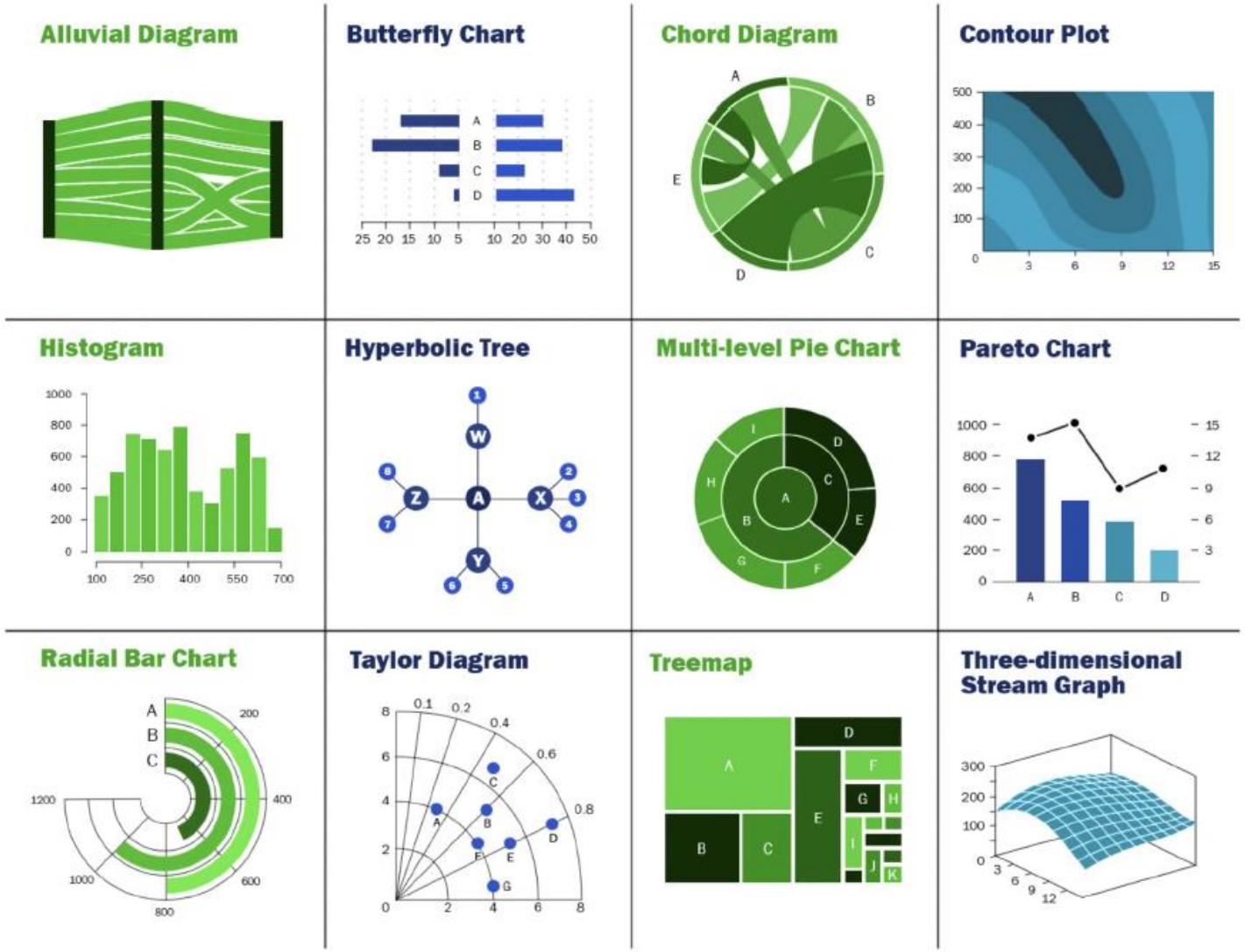
Area (feet^2)	Sale Price ($)
2140	460k
1416	232k
1534	315k
852	178k
500	135k
...	...

What is Data Visualization?

Data visualization translates complex data sets into visual formats that are easier for the human brain to comprehend.



What is Data Visualization?



Data visualizations can take on several different forms and can use a variety of different data

Library


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

Matplotlib offers extensive customization but demands more code

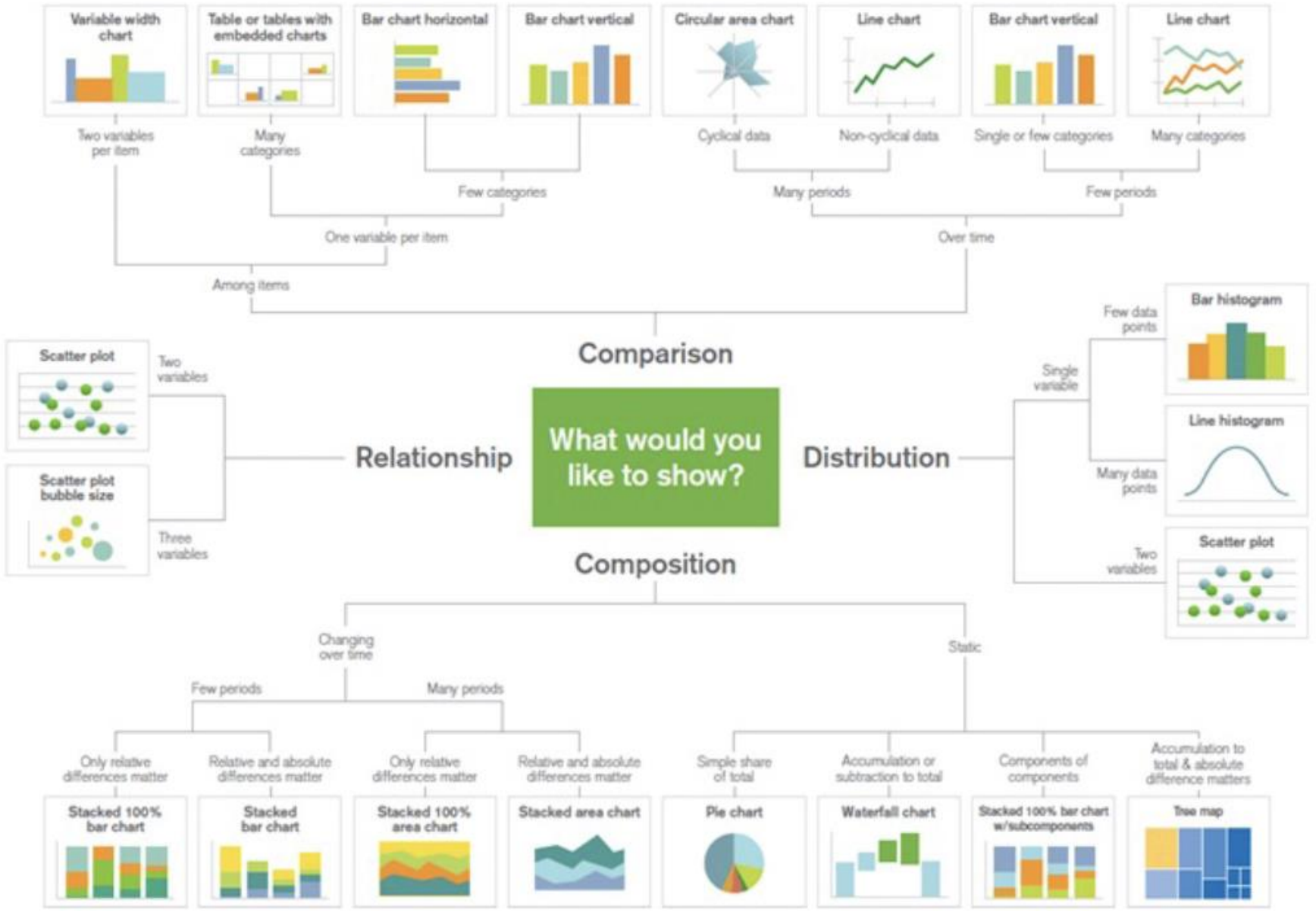
```
import seaborn as sns
```

Seaborn simplifies statistical plots with built-in themes

```
import plotly.express as px
```

Plotly excels at creating dynamic and interactive visualizations

Data Visualization with Python



Electricity Transformer Dataset (ETT Dataset)

Ex: Electricity Transformer Dataset

How to Understand This Dataset

	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

Field	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
Description	The recorded **date**	**H**igh **U**se**F**ul Load	**H**igh **U**se**L**ess Load	Middle **U**se**F**ul Load	Middle **U**se**L**ess Load	Low **U**se**F**ul Load	Low **U**se**L**ess Load	**O**il Temperature (target)

Ex: Electricity Transformer Dataset (ETT Dataset)

FPT Education
FPT SCHOOL OF BUSINESS & TECHNOLOGY

Prepare and load dataset

```
!curl -L -o ETTh1.csv https://raw.githubusercontent.com/zhouhaoyi/ETDataset/main/ETT-small/ETTh1.csv
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2528k  100 2528k    0     0  3567k      0 --:--:-- --:--:-- --:--:-- 3571k
```

```
[ ]  import plotly.express as px
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import numpy as np
```

```
[ ]  df = pd.read_csv("/content/ETTh1.csv", parse_dates=['date'])
```

Ex: Electricity Transformer Dataset

Prepare and load dataset

```
df.describe()
```

	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
count	17420.000000	17420.000000	17420.000000	17420.000000	17420.000000	17420.000000	17420.000000
mean	7.375141	2.242242	4.300239	0.881568	3.066062	0.856932	13.324672
std	7.067744	2.042342	6.826978	1.809293	1.164506	0.599552	8.566946
min	-22.705999	-4.756000	-25.087999	-5.934000	-1.188000	-1.371000	-4.080000
25%	5.827000	0.737000	3.296000	-0.284000	2.315000	0.670000	6.964000
50%	8.774000	2.210000	5.970000	0.959000	2.833000	0.975000	11.396000
75%	11.788000	3.684000	8.635000	2.203000	3.625000	1.218000	18.079000
max	23.643999	10.114000	17.341000	7.747000	8.498000	3.046000	46.007000

The number of not-empty values

The average (mean) value

The standard deviation

The minimum value

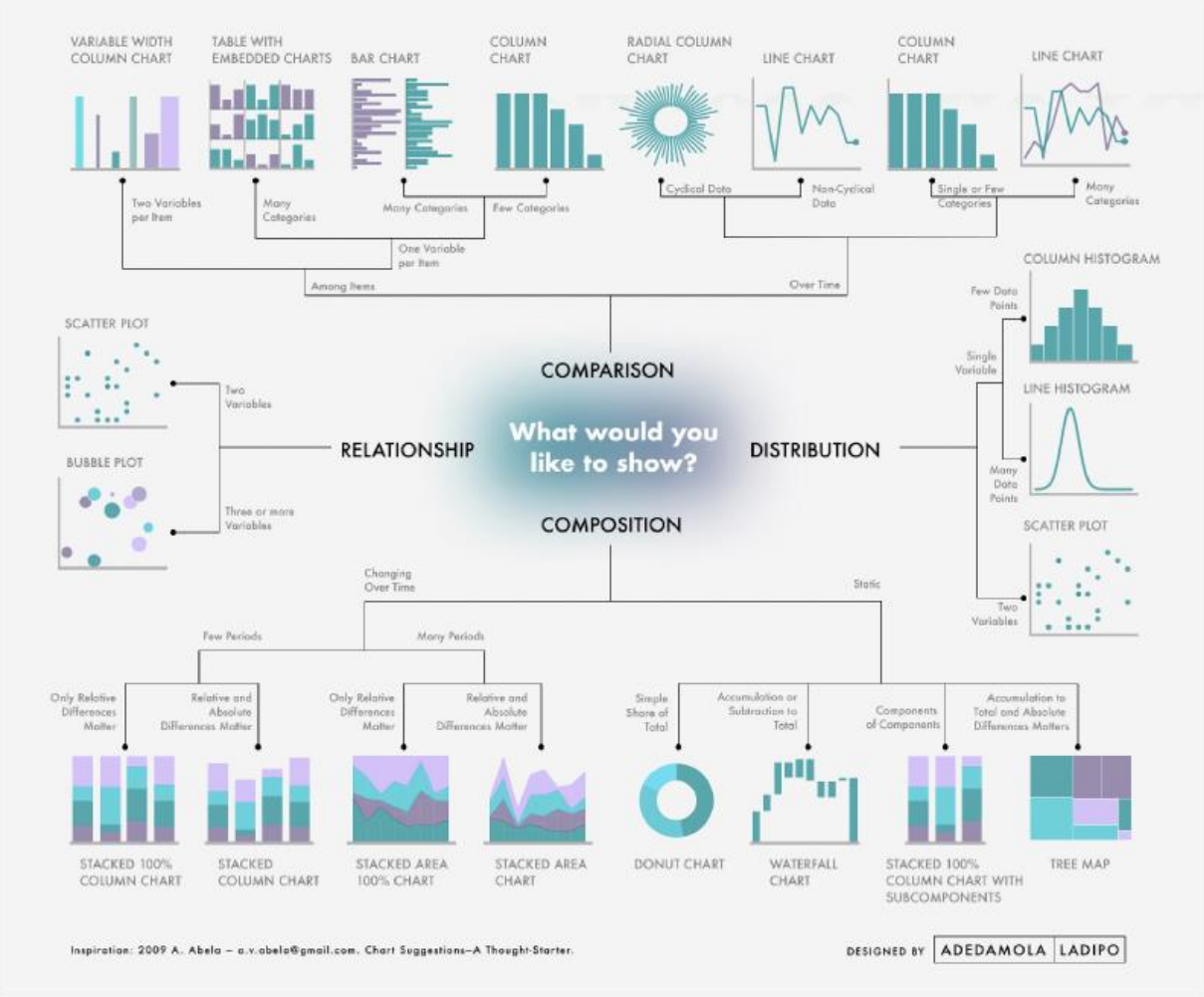
The 25% percentile*

The 50% percentile*

The 75% percentile*

The maximum value

Choose the right chart type



- Data visualization is an essential tool for any data-driven organization, allowing users to make sense of complex data sets and communicate insights to stakeholders effectively.
- However, with so many chart types to choose from, it can be challenging to decide which one to use for a given set of data.

Choose the right chart type

- Step 1: Determine the Type of Data
- Step 2: Identify the Relationship Between Variables
- Step 3: Determine the Purpose of Visualization
- Step 4: Identify the Audience
- Step 5: Select the Appropriate Chart Type

Choose the right chart type

❑Step 1: Determine the Type of Data

Data can be categorized into one of four types: quantitative, categorical, temporal, or spatial.

Quantitative data

data refers to numerical values

Categorical data

refers to non-numerical values

Temporal data

Refers to time-based data

Spatial data

refers to location-based data

Electricity Transformer Dataset

	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

Choose the right chart type

❑Step 2: Identify the Relationship Between Variables

Identify the relationship between the variables you want to represent in your visualization.
Do you want to show a comparison, a distribution, or a relationship?

	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

Electricity Transformer Dataset

- A comparison chart is useful for showing the differences between two or more data points, such as a bar chart or a column chart.
- A distribution chart is useful for showing how data is spread out, such as a histogram or a box plot.
- A relationship chart is useful for showing how two or more variables are related, such as a scatter plot or a bubble chart.

Choose the right chart type

❑Step 3: Determine the Purpose of Visualization

What message do you want to convey through your data visualization?
Do you want to show a trend, a comparison, or a distribution?

	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

Electricity Transformer Dataset

- if you want to show a trend over time, a line chart or an area chart might be more appropriate.
- If you want to compare data points, a bar chart or a column chart might be a better choice.
- If you want to show a distribution, a histogram or a box plot might be more useful.

Choose the right chart type

❑Step 4: Identify the Audience

It's important to consider the audience for whom you are creating the visualization.
Will they understand complex charts or require a more straightforward representation?

	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

Electricity Transformer Dataset

- If your audience is expert, you might be able to use more complex charts, such as heat maps.
- If your audience is less familiar with data visualization, simpler charts like pie charts or bar charts might be more effective.

Choose the right chart type

❑Step 5: Select the Appropriate Chart Type

Remember, no chart type is a one-size-fits-all solution, and sometimes, multiple chart types might work better to communicate your message effectively. Therefore, it's essential to experiment with different chart types to find the most appropriate one for your data.

Electricity Transformer Dataset

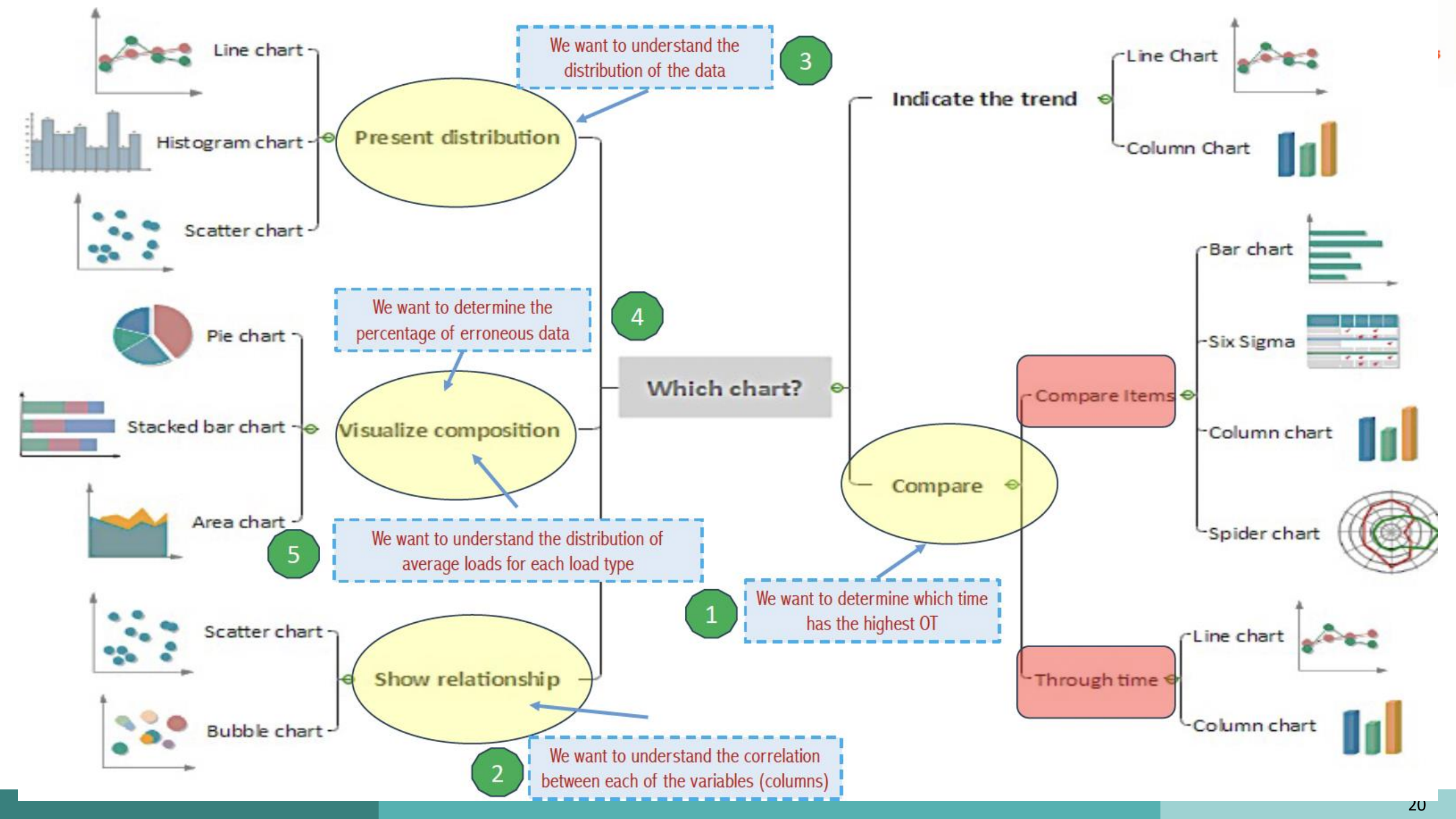
	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

Choose the right chart type

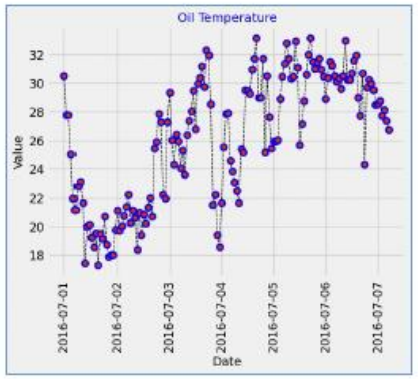
Electricity Transformer Dataset

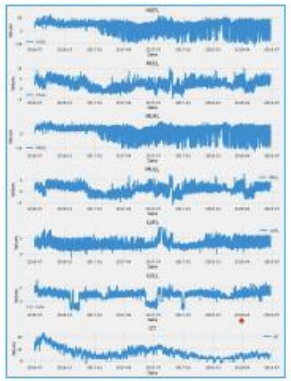
	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

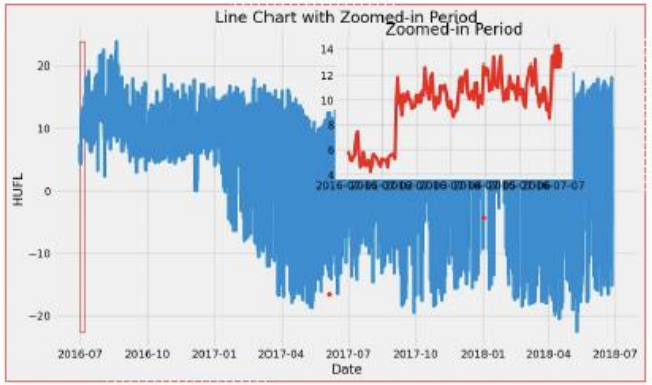
1. We want to determine which time has the highest OT
2. We want to understand the correlation between each of the variables (columns)
3. We want to understand the distribution of the data
4. We want to determine the percentage of erroneous data
5. We want to understand the distribution of average loads for each load type (High Useful Load - HUFL, High Useless Load - HULL, Middle Useful Load - MUFL, Middle Useless Load - MULL, Low Useful Load - LUFL, Low Useless Load - LULL)

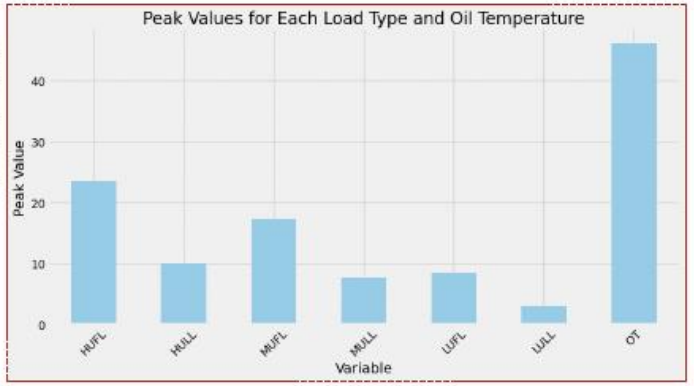

Which chart?
Present distribution
We want to understand the distribution of the data
3
Line chart
Histogram chart
Scatter chart
Visualize composition
We want to determine the percentage of erroneous data
4
Pie chart
Stacked bar chart
Area chart
5
We want to understand the distribution of average loads for each load type
Show relationship
We want to understand the correlation between each of the variables (columns)
2
Scatter chart
Bubble chart
Compare
We want to determine which time has the highest OT
1
Indicate the trend
Line Chart
Column Chart
Compare Items
Bar chart
Six Sigma
Column chart
Spider chart
Through time
Line chart
Column chart

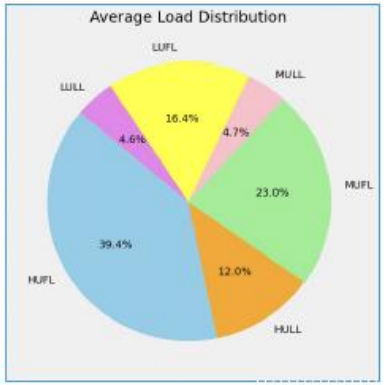
Data Visualization for ETT Dataset

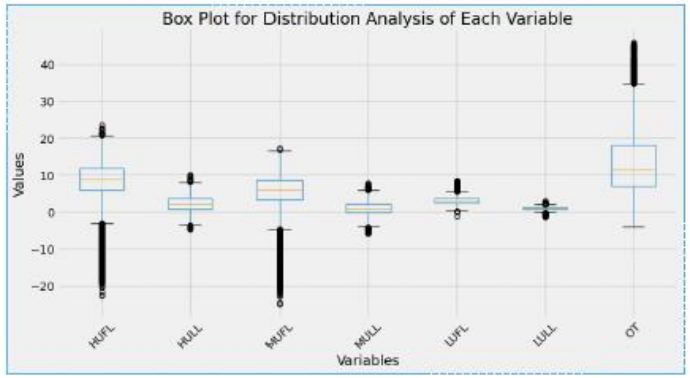


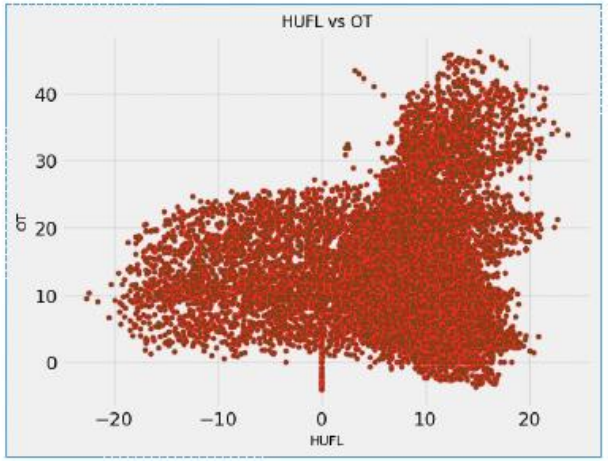


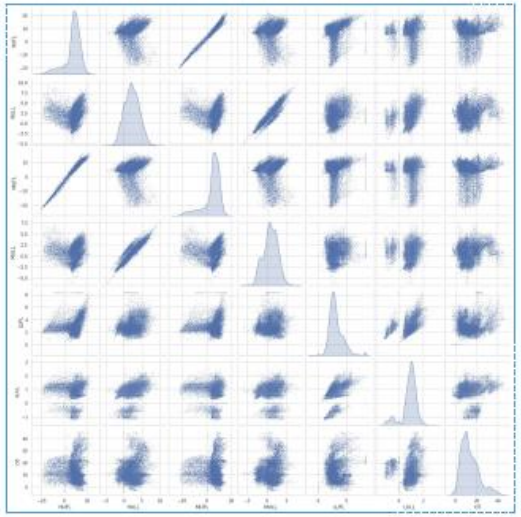












Line Chart

ETT Dataset: Line Chart

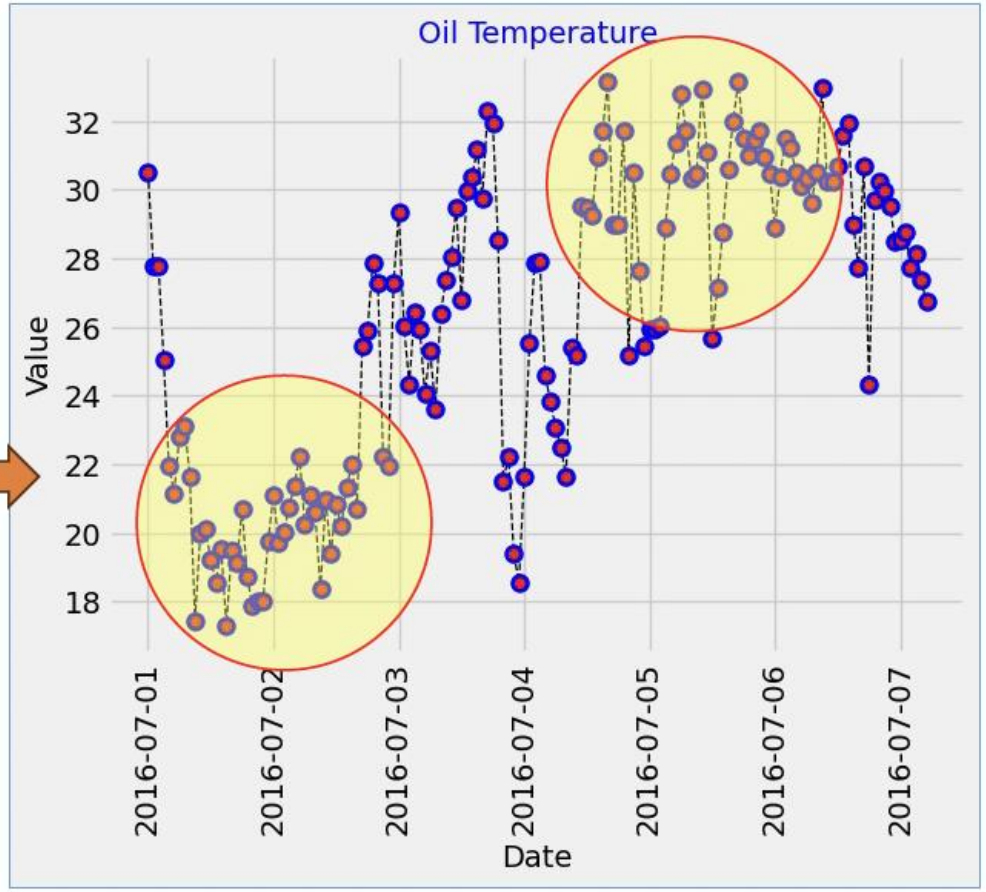
It is one of the most common charts that is used to observe a single or multiple variables with the change of another variable. Basically, it is used in trend analysis and time series analysis

Single Plot Chart

Display the Oil Temperature Of The First 100 samples

```
plt.plot(df.date[0:150], df.OT[0:150], marker = 'o', color = 'black',
         linewidth = 0.9, linestyle = '--',
         markeredgecolor = 'blue',
         markeredgewidth = '2.0',
         markerfacecolor = 'red', markersize = 7.0)
plt.title('Oil Temperature', color = 'Blue', size = 14)
plt.xlabel('Date', size = 14)
plt.ylabel('Value', size = 14)
plt.style.use('fivethirtyeight')
plt.grid(True)
plt.xticks(rotation = 90)
plt.show()
```

Matplotlib Library



ETT Dataset: Line Chart

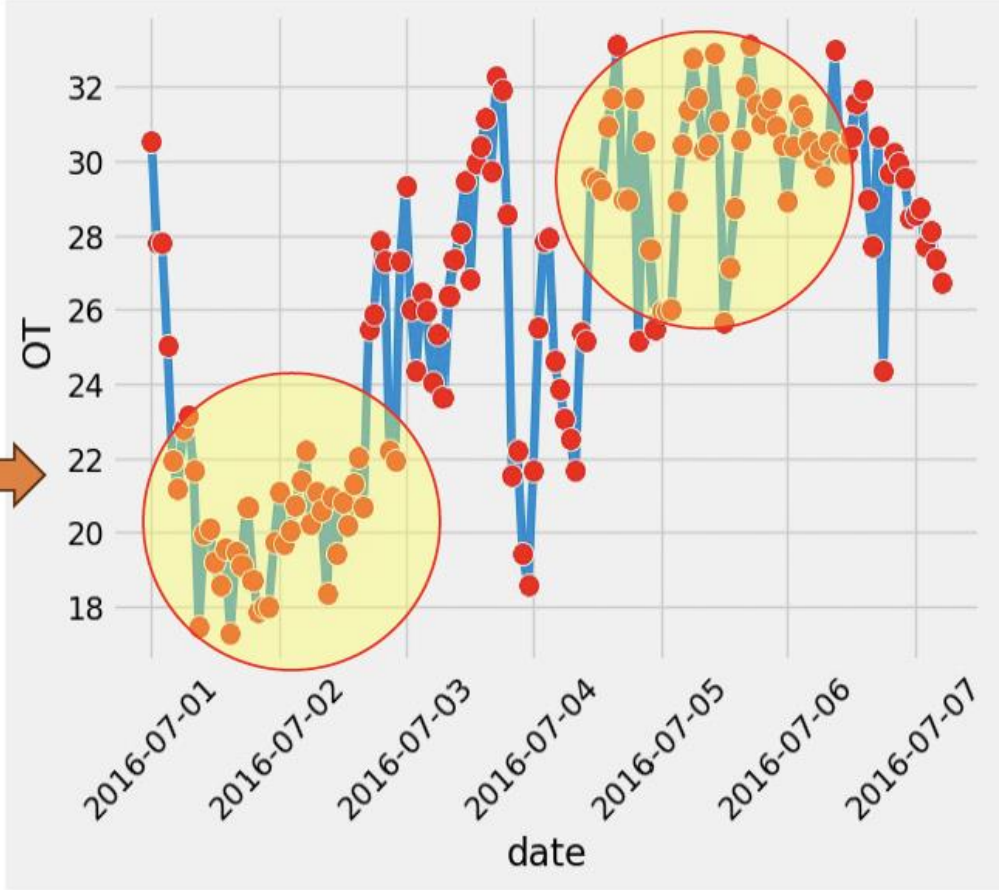
It is one of the most common charts that is used to observe a single or multiple variables with the change of another variable. Basically, it is used in trend analysis and time series analysis

Single Plot Chart

Display the Oil Temperature Of The First 150 samples

```
#using seaborn
sns.lineplot(data=df, x=df.date[0:150], y=df.OT[0:150], marker='o',
             markersize=10, markerfacecolor='red')
plt.xticks(rotation=45)
```

Seaborn Library



ETT Dataset: Line Chart

It is one of the most common charts that is used to observe a single or multiple variables with the change of another variable. Basically, it is used in trend analysis and time series analysis

Single Plot Chart

Display the Oil Temperature Of The First 150 samples

```
#using plotly
import plotly.express as px
fig = px.line(df, x=df.date[0:150], y=df.OT[0:150],
              title='Oil Temperature', markers=True)
fig.update_layout(
    xaxis_title="Date", yaxis_title="Value"
)
fig.show()
```

Plotly Library

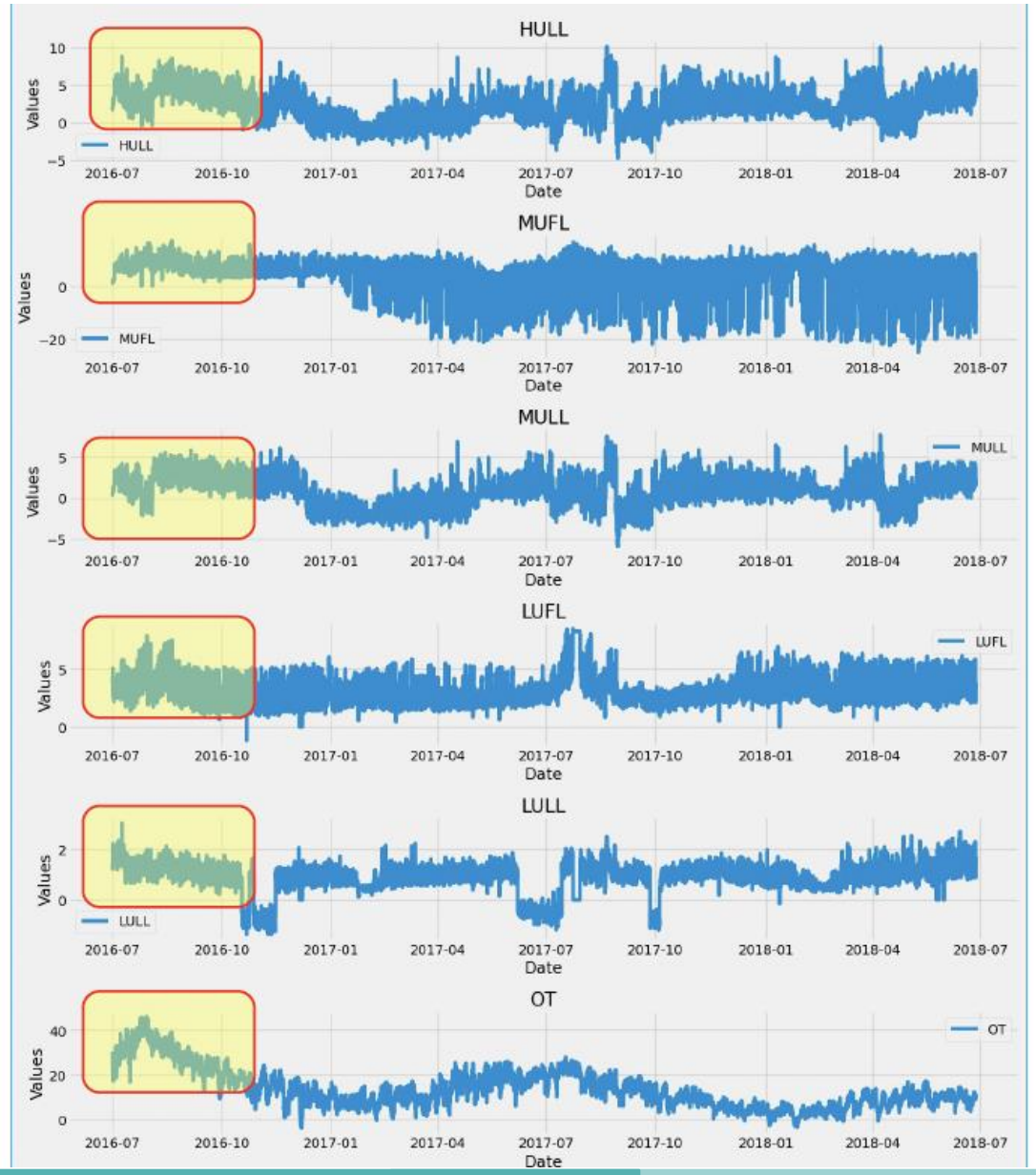
ETT Dataset: Line Chart

Multiple Plot Line Chart

Display the HUFL, HULL, MUFL, MULL,... of all samples

```
# Setting the figure size and layout
fig, axes = plt.subplots(nrows=7, ncols=1, figsize=(15, 20))

# Plotting line charts for each variable except 'date'
for i, column in enumerate(df.columns[1:]):
    axes[i].plot(df['date'], df[column], label=column)
    axes[i].set_title(column)
    axes[i].set_xlabel('Date')
    axes[i].set_ylabel('Values')
    axes[i].legend()

# Adjust layout for better spacing
plt.tight_layout()

# Display the plots
plt.show()
```

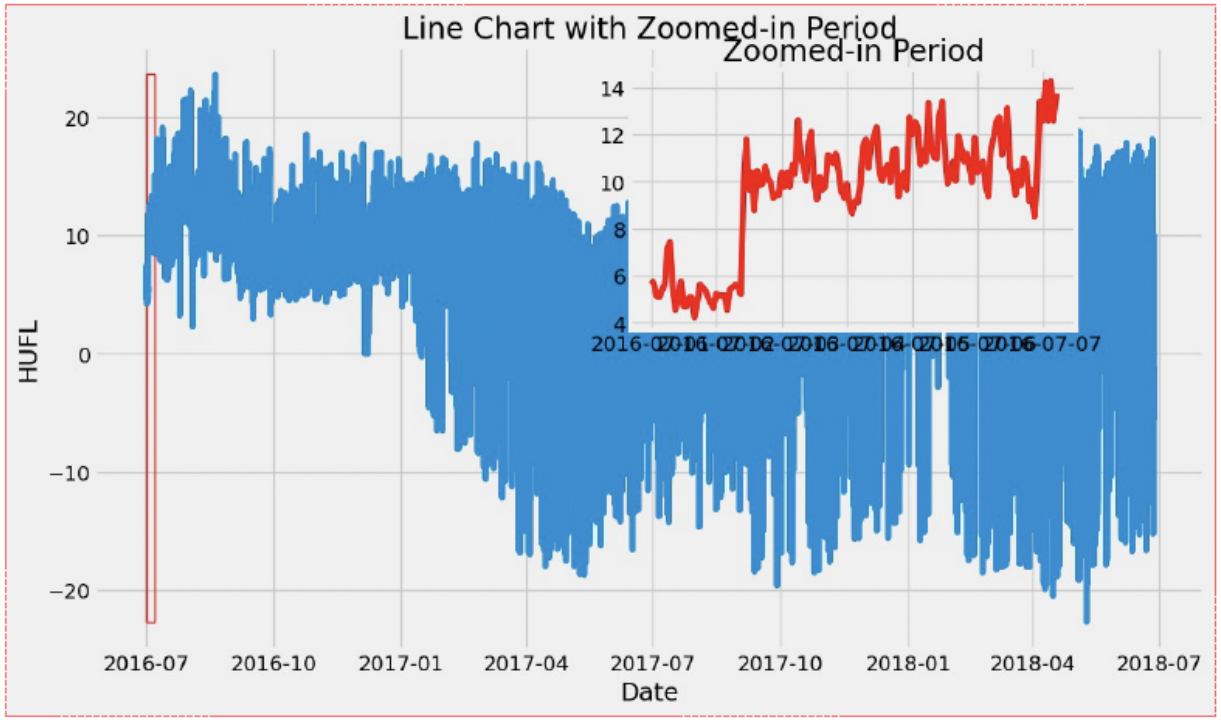
Matplotlib Library



ETT Dataset: Line Chart

Line Chart With Zoomed-in-Period

```
#  selecting a time period to zoom in on
start = 0
end =  150
# Creating the main line chart with a highlighted square
fig, ax = plt.subplots(figsize=(12, 7))
# Plotting the main line chart for a selected variable (example: 'HUFL')
ax.plot(df['date'], df['HUFL'], label='HUFL')
# Adding a rectangle to highlight the zoomed-in area
rect = plt.Rectangle((df['date'][start], min(df['HUFL'])),
                     df['date'][end] - df['date'][start],
                     max(df['HUFL']) - min(df['HUFL']),
                     linewidth=1, edgecolor='r', facecolor='none')
ax.add_patch(rect)
# Creating an inset (zoomed-in) plot within the main plot
ax_inset = fig.add_axes([0.5, 0.5, 0.35, 0.35])  # [left, bottom, width, height]
ax_inset.plot(df['date'][start:end], df['HUFL'][start:end], color='r')
ax_inset.set_title('Zoomed-in Period')
# Setting labels and title for the main plot
ax.set_xlabel('Date')
ax.set_ylabel('HUFL')
ax.set_title('Line Chart with Zoomed-in Period')
# Display the plot
plt.show()
```

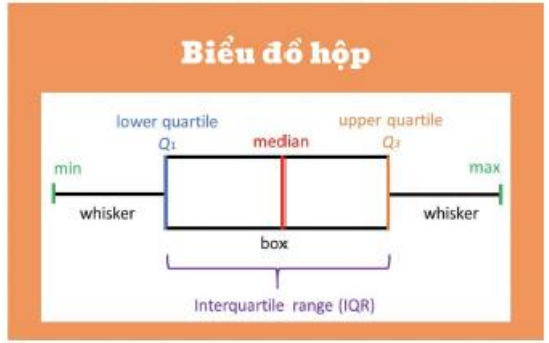


Matplotlib Library

Box Plots

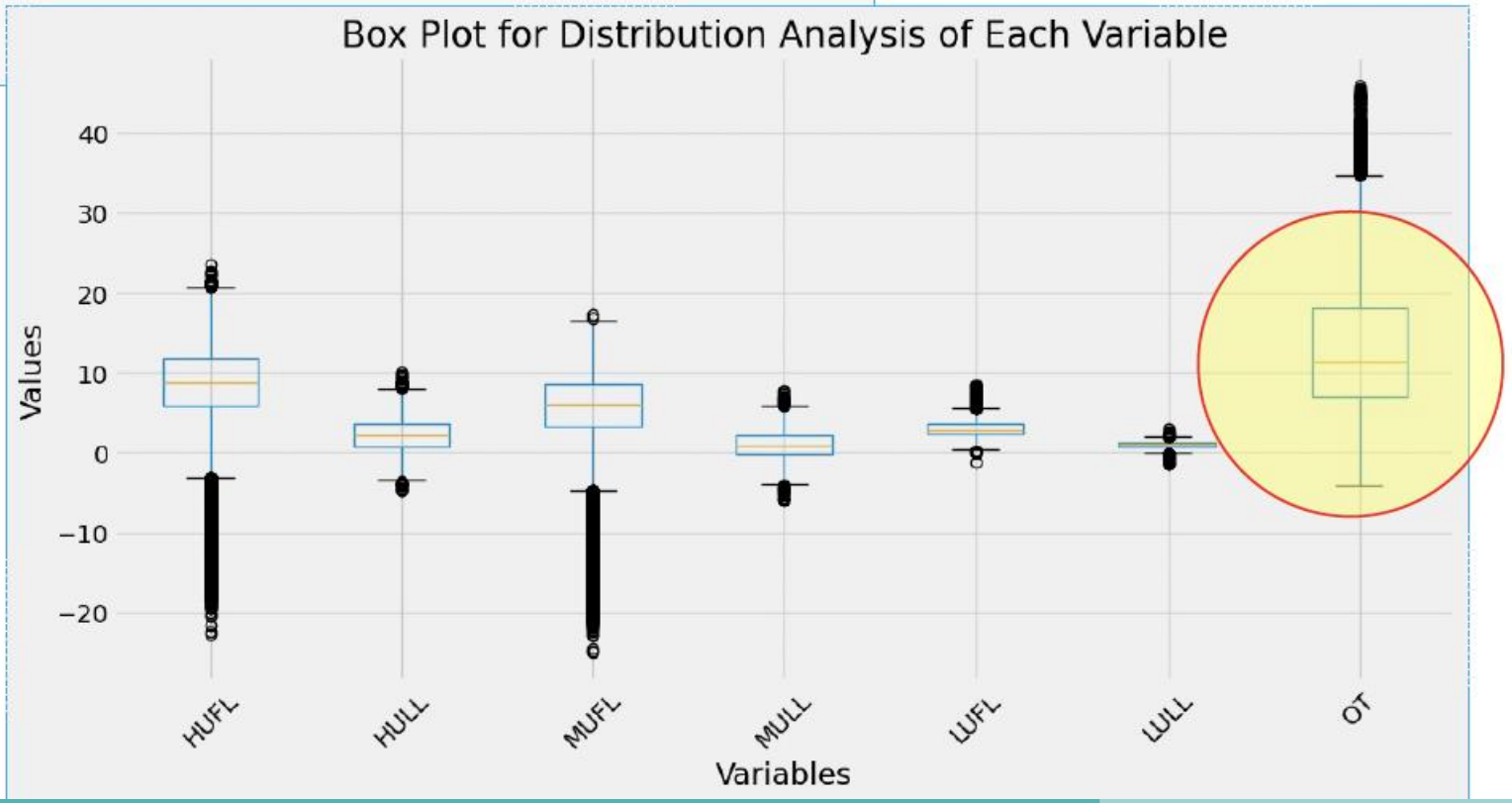
ETT Dataset: Box Plots

```
# Creating Box Plots for Distribution Analysis of each variable
plt.figure(figsize=(12, 6))
df.boxplot(column=['HUFL', 'HULL', 'MUFL', 'MULL', 'LUFL', 'LULL', 'OT'])
plt.xlabel('Variables')
plt.ylabel('Values')
plt.title('Box Plot for Distribution Analysis of Each Variable')
plt.xticks(rotation=45)
plt.show()
```

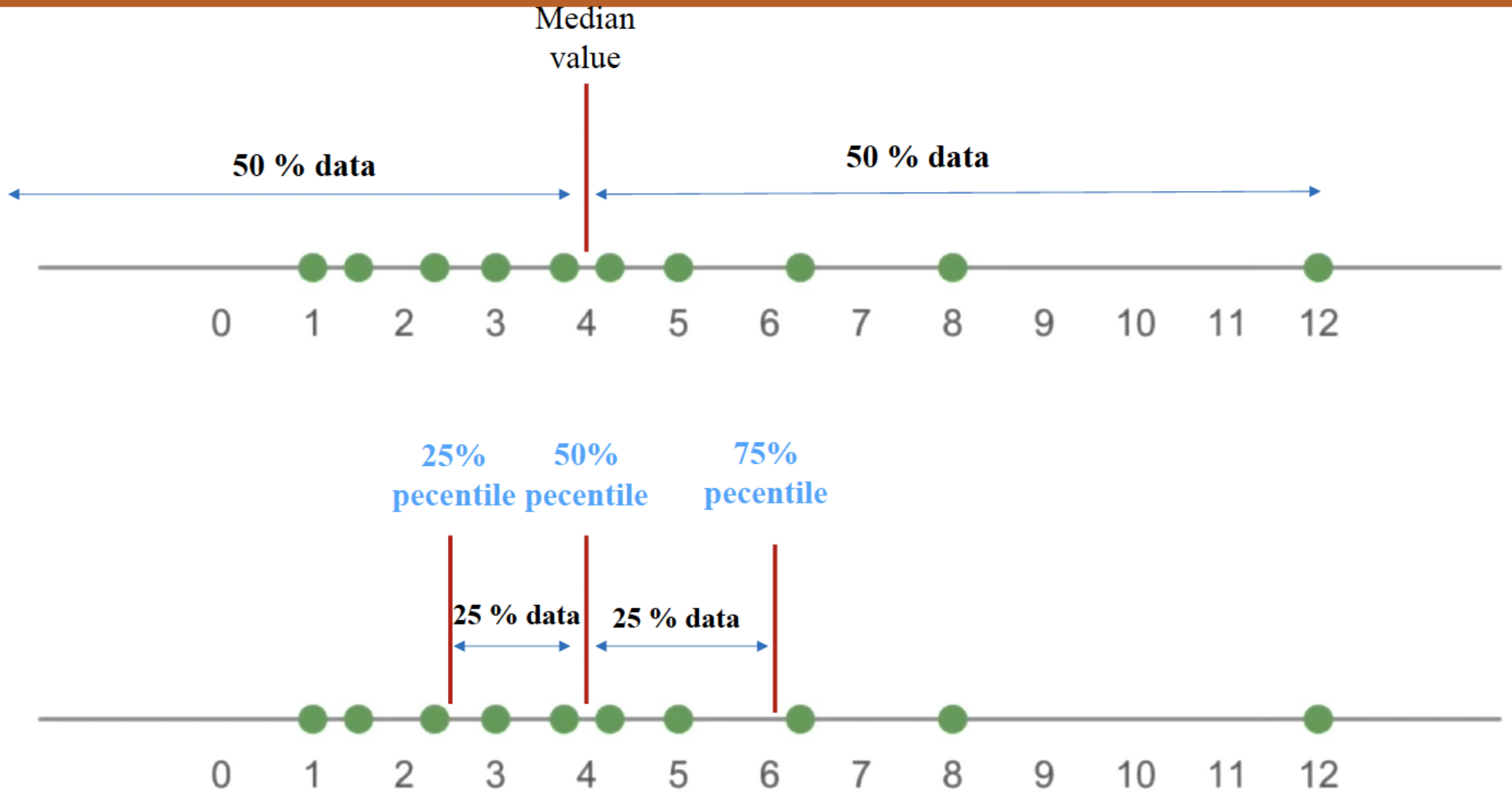


Matplotlib Library

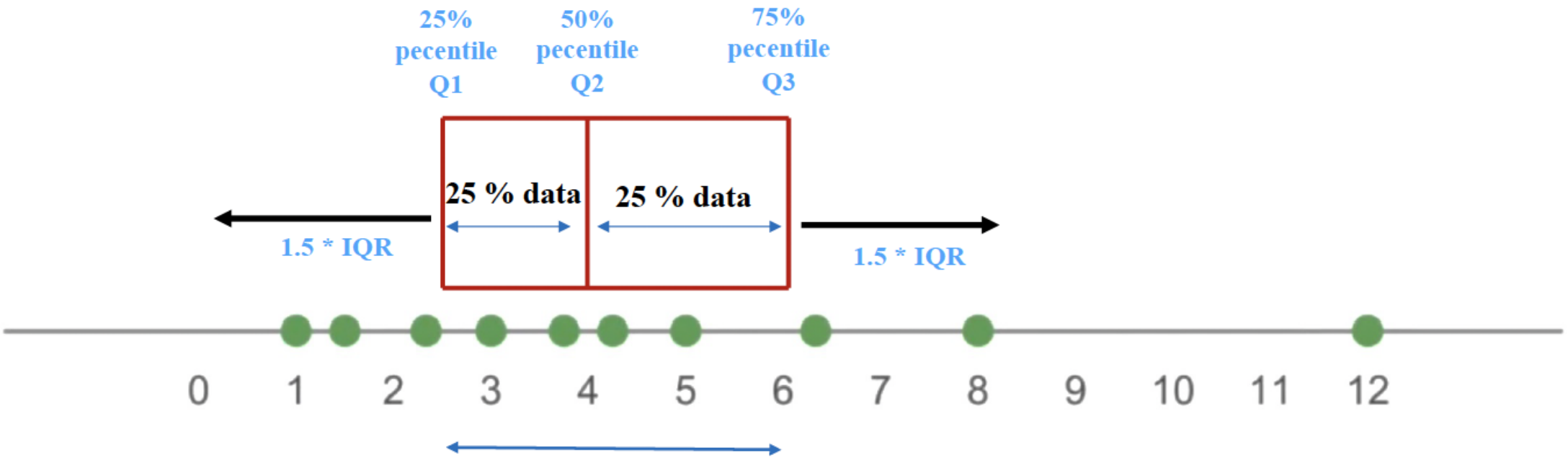
- Box and whisker plots portray the distribution of your data, outliers, and the median.
- Determine the existence of outliers within the dataset



ETT Dataset: Box Plots

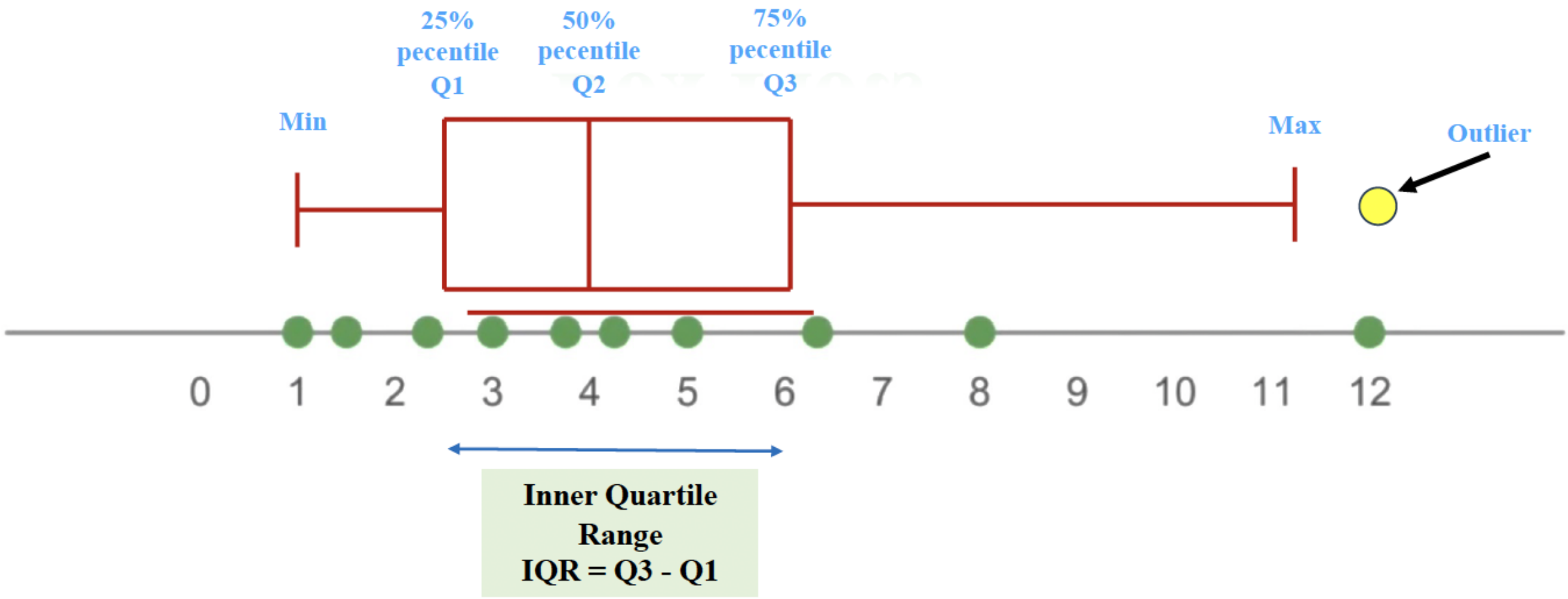


ETT Dataset: Box Plots



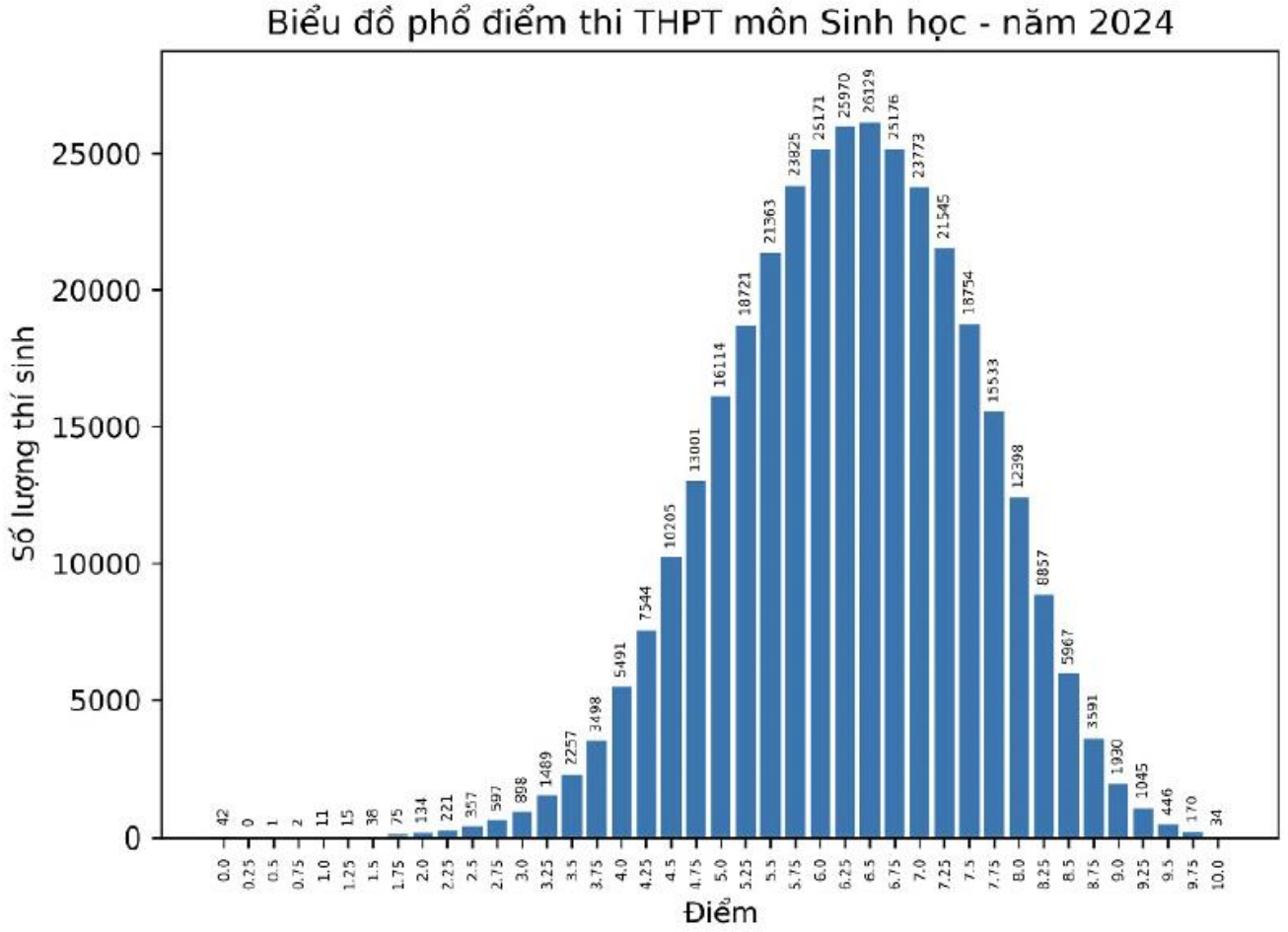
Why John Tukey set 1.5 IQR to detect outliers instead of 1 or 2?

ETT Dataset: Box Plots



ETT Dataset: Box Plots

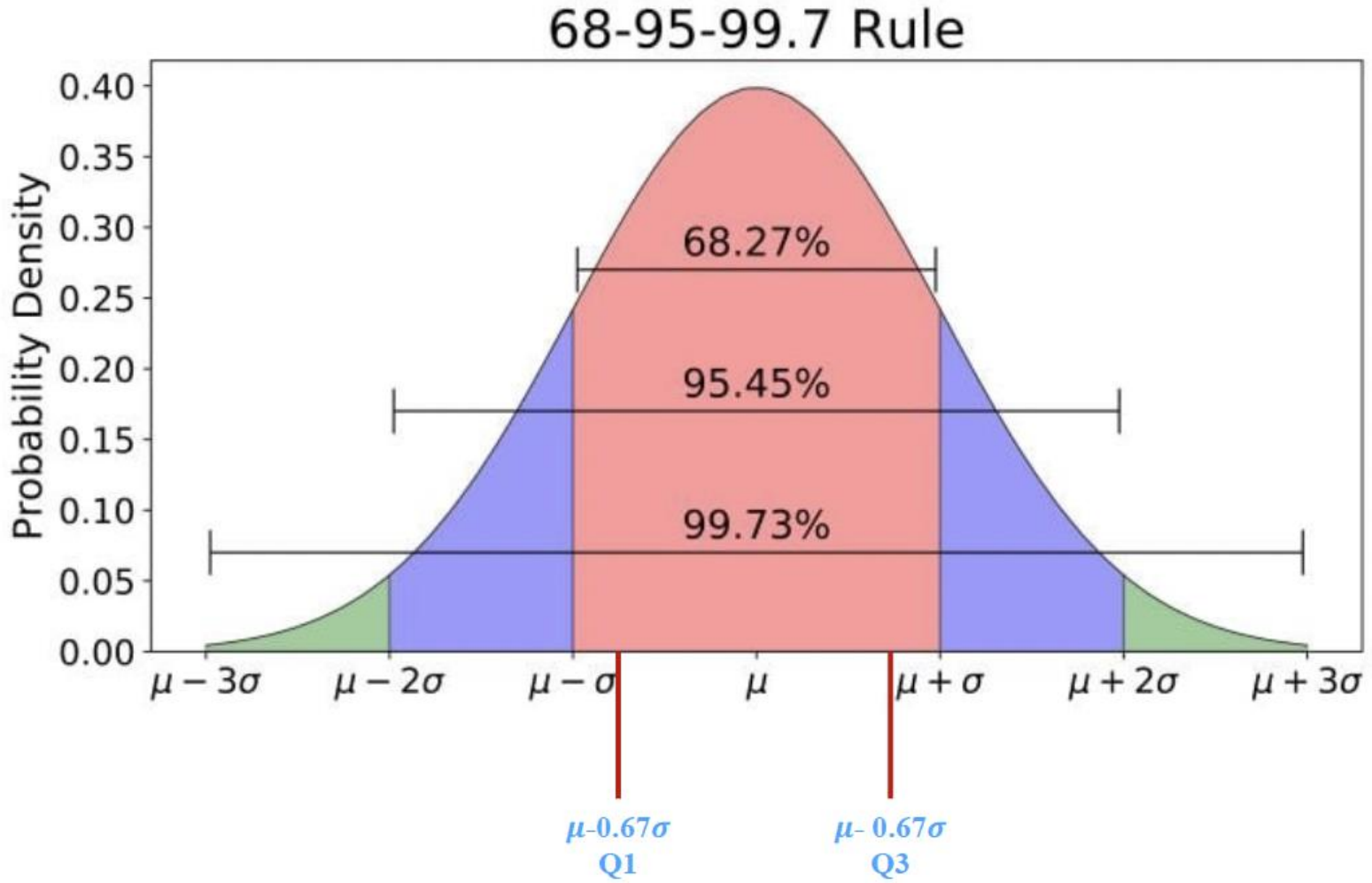
	2023		2024	
Tổng số HS	324,625		342,388	
ĐTB	6.39		6.28	
Trung vị	6.5		6.25	
Số HS có điểm <=1	36	0.011 %	56	0.016 %
Số HS có điểm dưới (<5)	33,754	10.398 %	45,876	13.399 %
Điểm HS đạt được nhiều nhất	6.5		6.5	
Số điểm 0	19		42	
Số điểm 10	135		34	



12

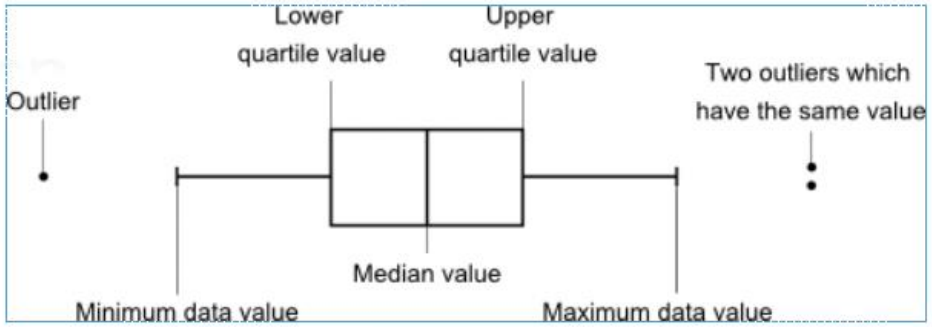
ETT Dataset: Box Plots

0.27 percent of the data lies outside three standard deviations (>3σ) of the mean (μ), taking both sides into account, the little red region in the figure



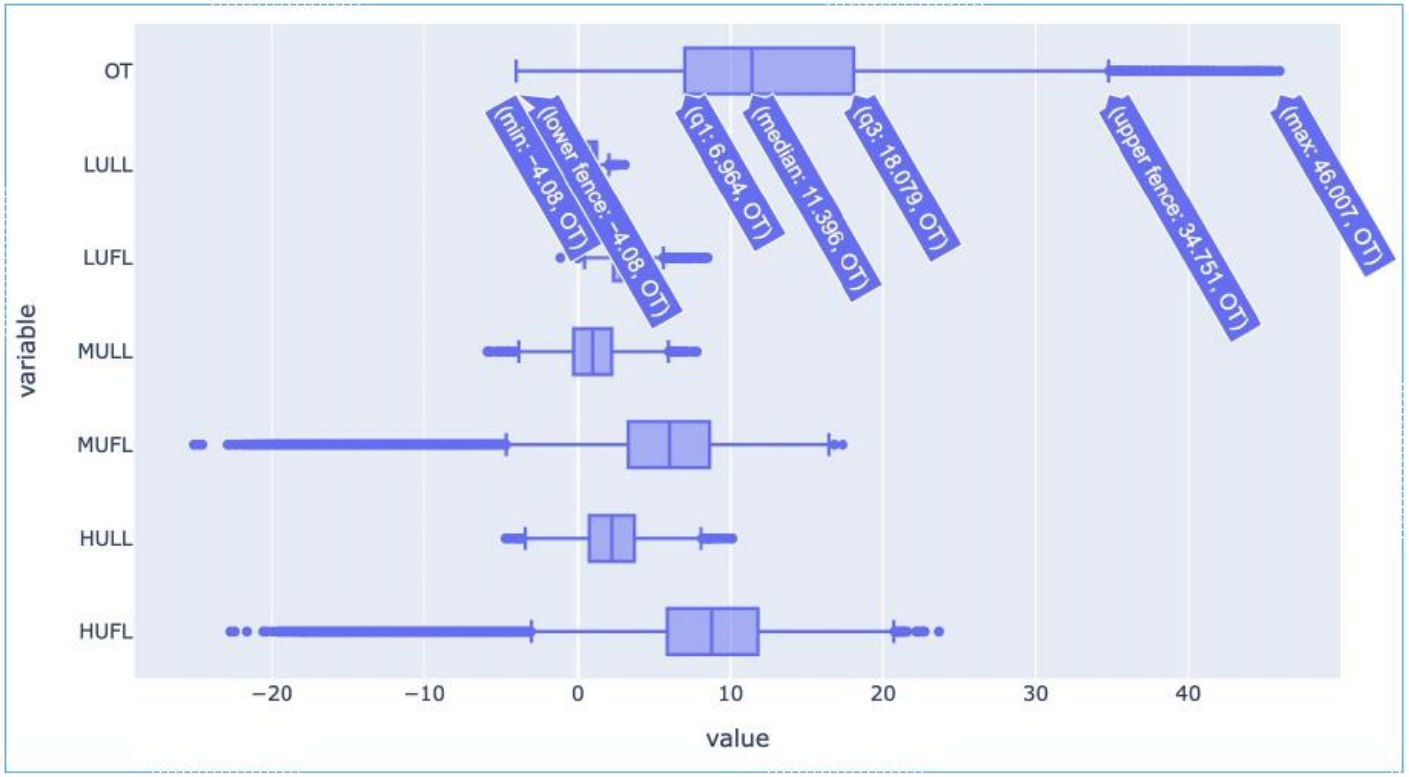
ETT Dataset: Box Plots

```
import plotly.express as px
# df = px.data.tips()
fig = px.box(df, x=['HUFL', 'HULL', 'MUFL', 'MULL', 'LUFL', 'LULL', 'OT'])
fig.show()
```



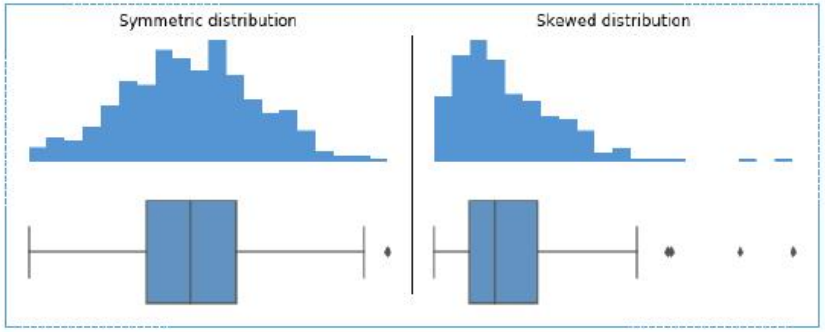
Plotly Library

- Box and whisker plots portray the distribution of your data, outliers, and the median.
- Determine the existence of outliers within the dataset



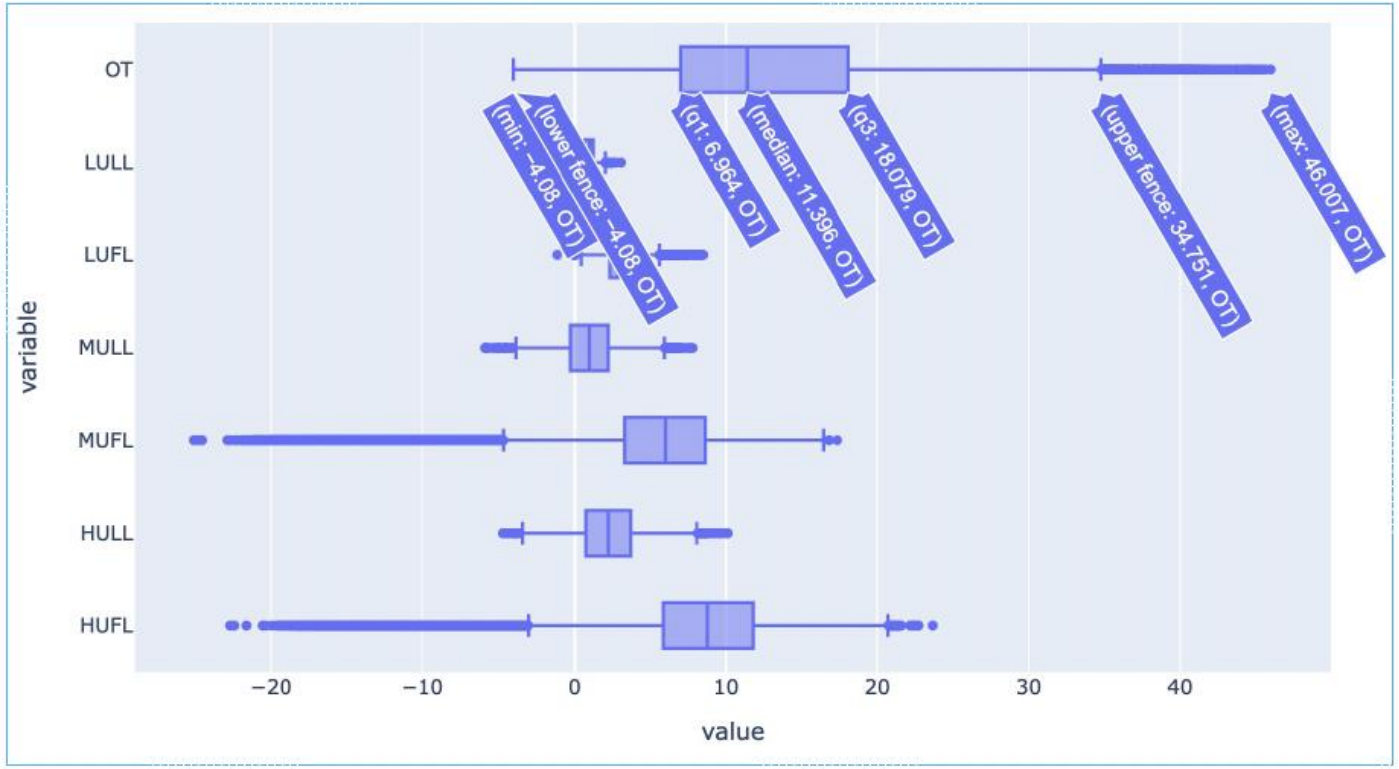
ETT Dataset: Box Plots

```
import plotly.express as px
# df = px.data.tips()
fig = px.box(df, x=['HUFL', 'HULL', 'MUFL', 'MULL', 'LUFL', 'LULL', 'OT'])
fig.show()
```



Plotly Library

- Box and whisker plots portray the distribution of your data, outliers, and the median.
- Determine the existence of outliers within the dataset

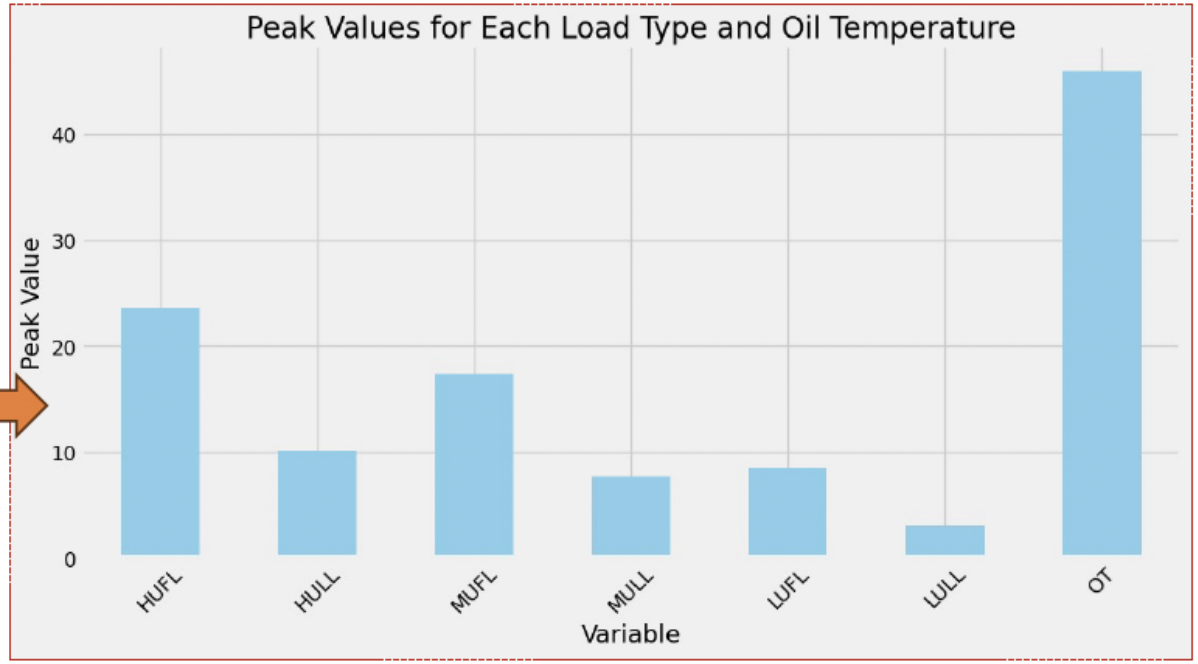


ETT Dataset: Bar Chart

A bar chart is also one of the most used charts. It uses bars to show the change in the value of a particular variable wrt the other. This type of chart is generally used discrete or categorical data. A bar chart can be horizontal or vertical.

```python
# Identifying peak values and their corresponding times for each column
peak_values = df.max()
peak_times = df.idxmax()

# Creating a DataFrame to display peak values and their times
peak_analysis = pd.DataFrame({'Peak Value': peak_values, 'Peak Time': peak_times})

# Dropping the 'date' column as it's not a load or temperature type
peak_analysis = peak_analysis.drop('date')

# Plotting the peak values for visual representation
plt.figure(figsize=(12, 6))
peak_analysis['Peak Value'].plot(kind='bar', color='skyblue')
plt.title('Peak Values for Each Load Type and Oil Temperature')
plt.xlabel('Variable')
plt.ylabel('Peak Value')
plt.xticks(rotation=45)
plt.show()
```

Matplotlib Library

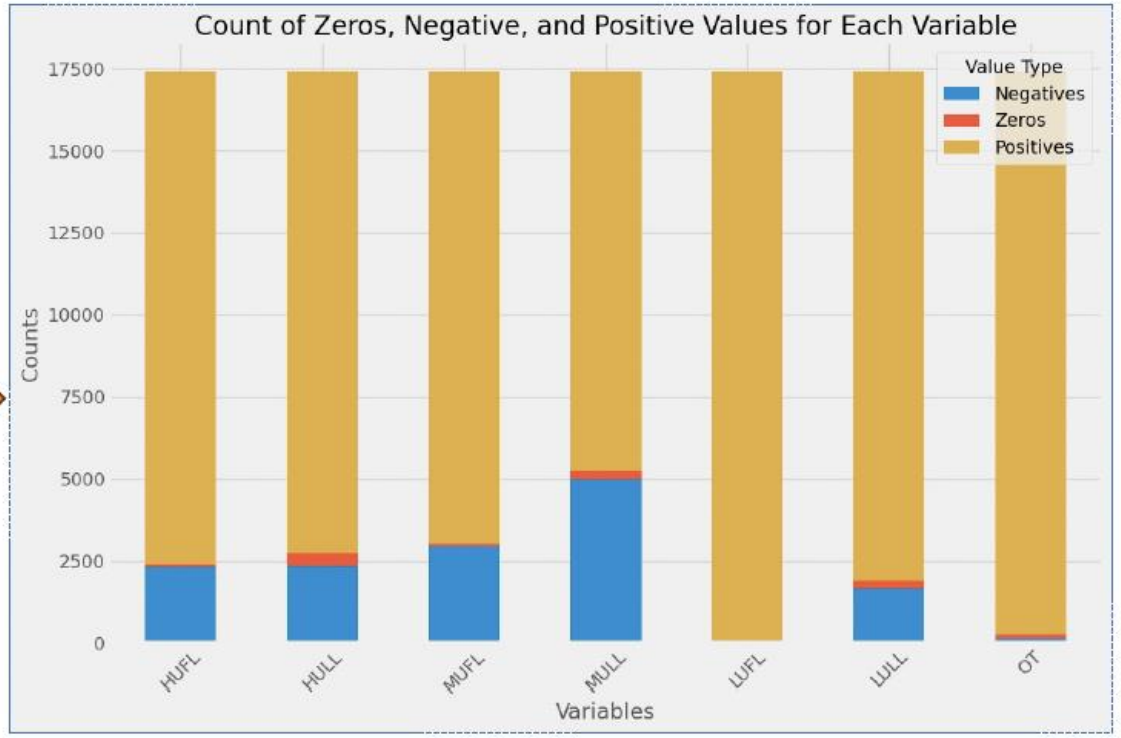


ETT Dataset: Bar Chart

A bar chart is also one of the most used charts. It uses bars to show the change in the value of a particular variable wrt the other. This type of chart is generally used discrete or categorical data. A bar chart can be horizontal or vertical.

```
# Preparing the data for the stacked bar chart to count zeros, negatives, and p
counts = {column: [
                    np.sum(df[column] < 0),
                    np.sum(df[column] == 0),
                    np.sum(df[column] > 0)]
          for column in df.columns[1:]}

# Creating a DataFrame from the counts dictionary
counts_df = pd.DataFrame(counts, index=['Negatives', 'Zeros', 'Positives']):T

# Plotting the stacked bar chart
ax = counts_df.plot(kind='bar', stacked=True, figsize=(12, 8))
ax.set_title('Count of Zeros, Negative, and Positive Values for Each Variable')
ax.set_xlabel('Variables')
ax.set_ylabel('Counts')
plt.xticks(rotation=45)
plt.legend(title='Value Type')
plt.tight_layout()
plt.show()
```

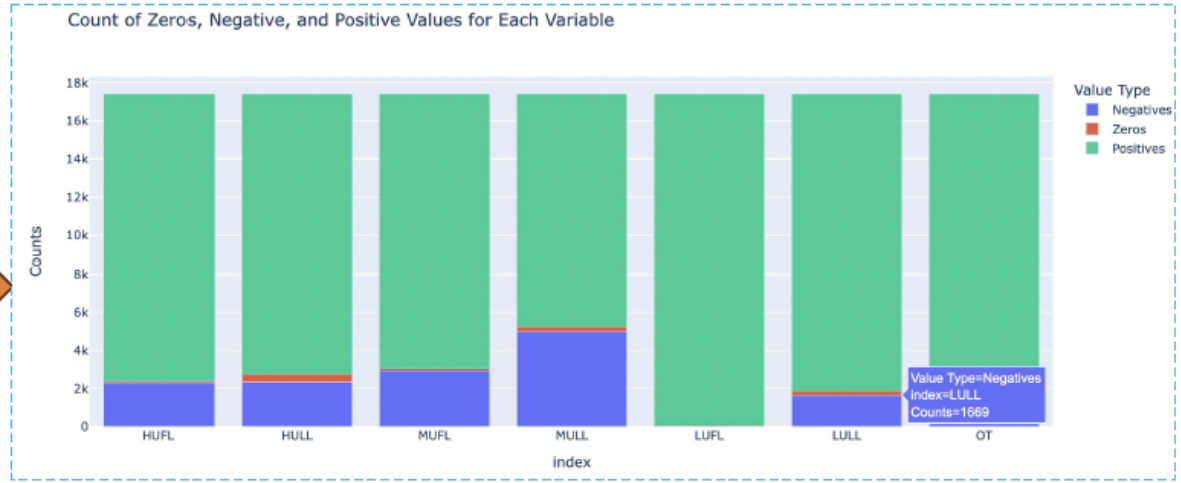
Matplotlib Library



ETT Dataset: Bar Chart

A bar chart is also one of the most used charts. It uses bars to show the change in the value of a particular variable wrt the other. This type of chart is generally used discrete or categorical data. A bar chart can be horizontal or vertical.

```
# Preparing the data for the stacked bar chart to count zeros, negatives, and p
counts = {column: [
                    np.sum(df[column] < 0),
                    np.sum(df[column] == 0),
                    np.sum(df[column] > 0)]
          for column in df.columns[1:]}

# Creating a DataFrame from the counts dictionary
counts_df = pd.DataFrame(counts, index=['Negatives', 'Zeros', 'Positives']).T

# Plotting the stacked bar chart
ax = counts_df.plot(kind='bar', stacked=True, figsize=(12, 8))
ax.set_title('Count of Zeros, Negative, and Positive Values for Each Variable')
ax.set_xlabel('Variables')
ax.set_ylabel('Counts')
plt.xticks(rotation=45)
plt.legend(title='Value Type')
plt.tight_layout()
plt.show()
```

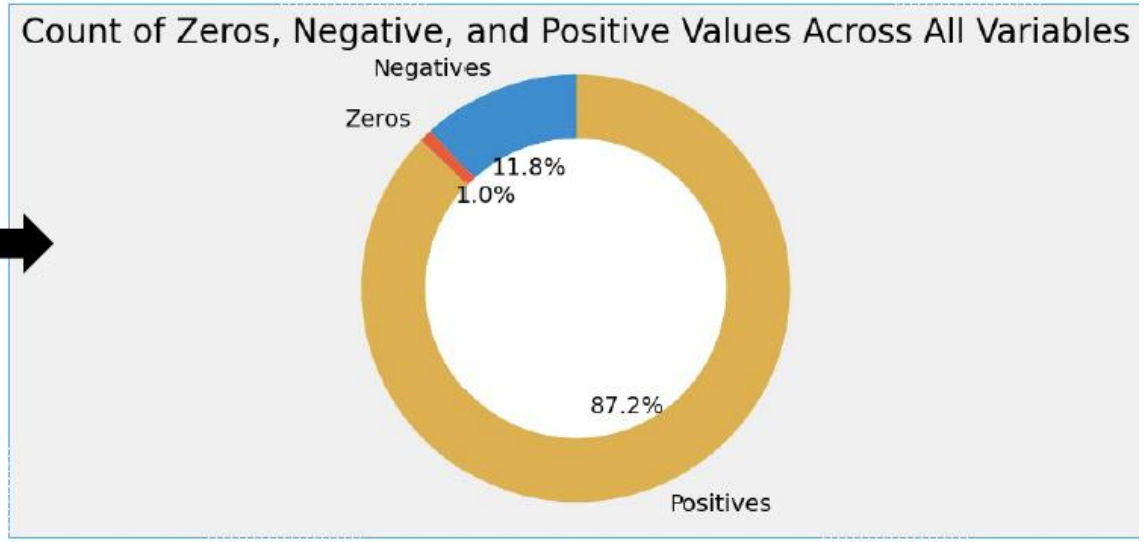


Plotly Library

Donut Chart

ETT Dataset: Donut Chart

```python
# Donut chart
# Preparing the data for the stacked bar chart to count zeros, negatives, and positives for each variable
counts = {column: [
                    np.sum(df[column] < 0),
                    np.sum(df[column] == 0),
                    np.sum(df[column] > 0)]
          for column in df.columns[1:]}

# Creating a DataFrame from the counts dictionary
counts_df = pd.DataFrame(counts, index=['Negatives', 'Zeros', 'Positives']).T

# Preparing data for the donut chart
counts_for_donut = counts_df.sum()

# Plotting the donut chart
fig, ax = plt.subplots()
ax.pie(counts_for_donut, labels=counts_for_donut.index, autopct='%1.1f%%', startangle=90, wedgeprops=dict(width=0.3))

# Draw a circle at the center to create a donut hole
centre_circle = plt.Circle((0,0),0.70,fc='white')
fig.gca().add_artist(centre_circle)

# Equal aspect ratio ensures that pie is drawn as a circle.
ax.axis('equal')

plt.title('Count of Zeros, Negative, and Positive Values Across All Variables')
plt.show()
```



Matplotlib Library

Correlation Chart

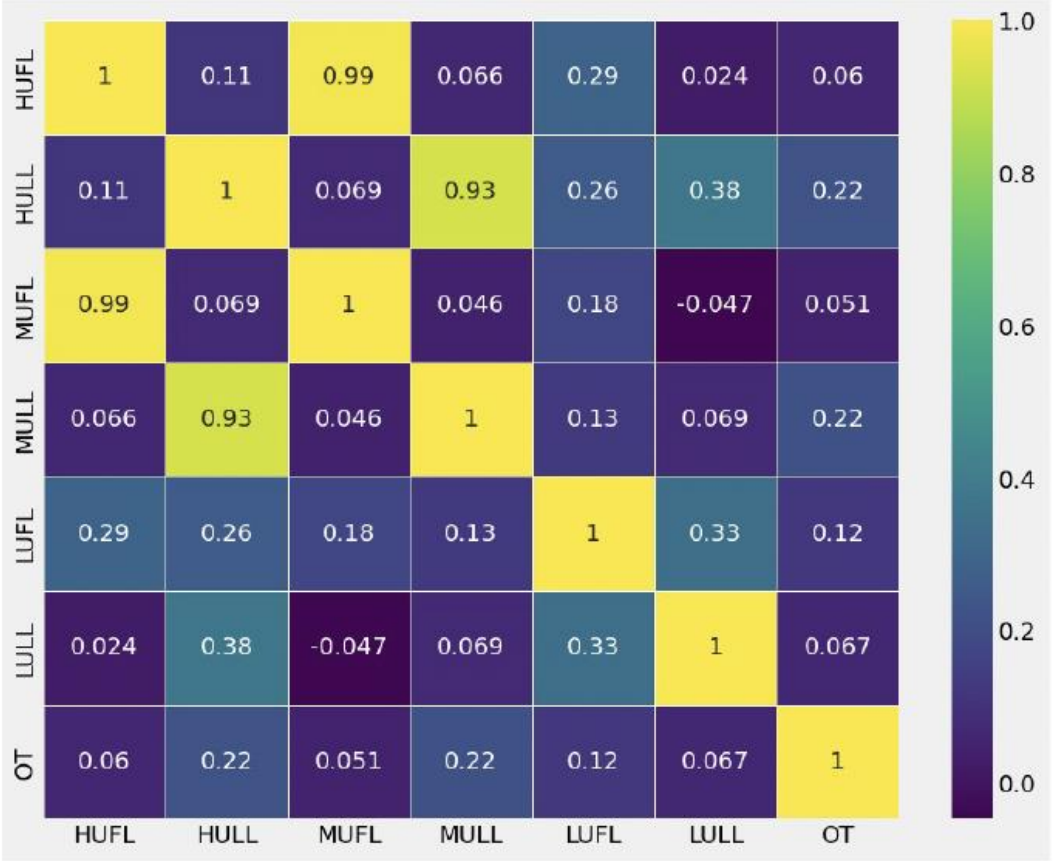
ETT Dataset: Correlation Chart

A heatmap is basically a two-dimensional plot which is mainly used to analyze the correlation between the different fields in a dataset. A heatmap is divided in square boxes that are colored with a specific color representing the correlation between the two fields it corresponds to

	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

```
corr_matrix = df.corr(numeric_only=True)
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='viridis', linewidths=0.5)
```

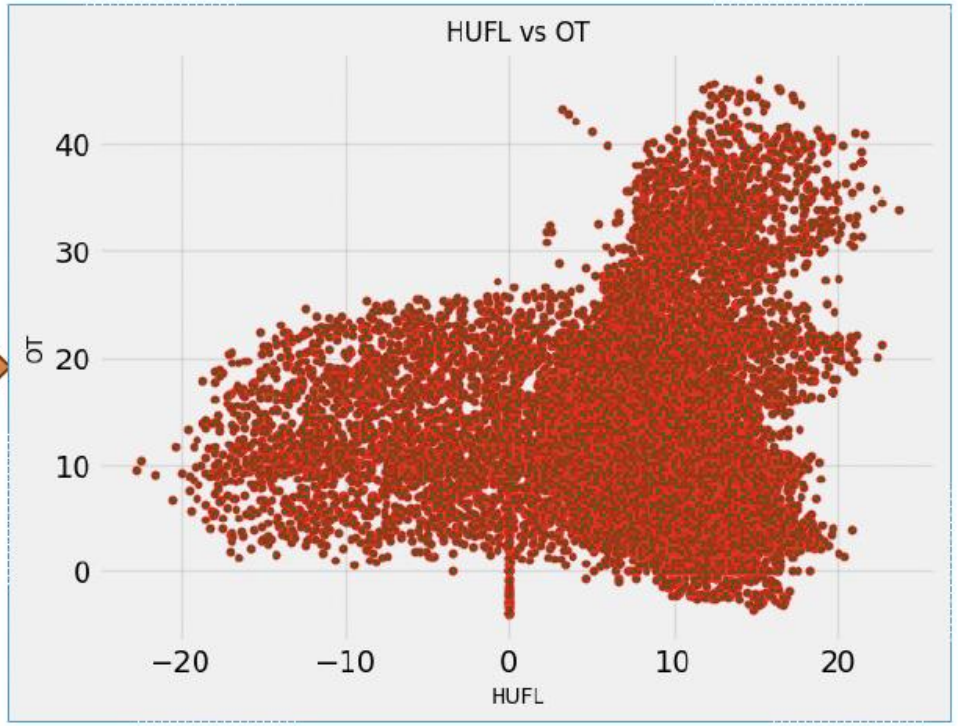
Seaborn Library



ETT Dataset: Correlation Chart

In a scatter plot, the data is scattered between the two axes. These type of plots are basically used to study the correlation between the two variables. If one variable increases or decreases with other, they are positively correlated. If one variable increases as the other decreases and vice versa, they are negatively correlated. If there is no such relation, they aren't related. It is one of the most common charts to observe the relation between two variables

```
plt.scatter(df.HUFL, df.OT, marker = "o",
            color = "green", linewidths = 1, edgecolors = "red", s = 10)
plt.style.use('fivethirtyeight')
plt.xlabel("HUFL", size = 10, color = "black")
plt.ylabel("OT", size = 10, color = "black")
plt.title("HUFL vs OT", size =12, color = "black")
plt.xticks(color = "black")
plt.yticks(color = "black")
plt.grid(color = "grey", alpha = 0.2)
plt.show()
```



Matplotlib Library

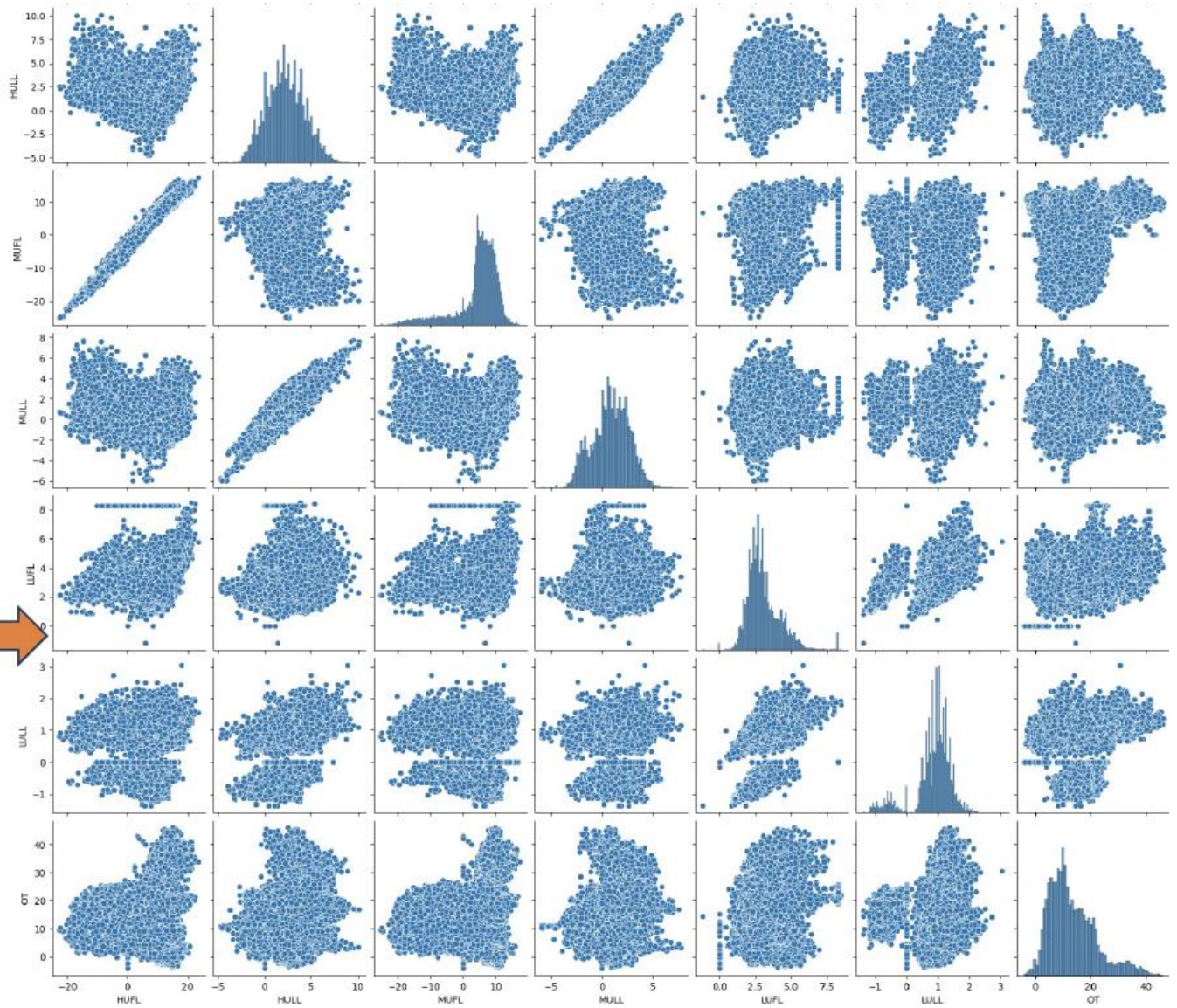
ETT Dataset: Pairplot

A pairplot, as the name suggests, plots pairwise bivariate distributions to show relation of each variable with all other variables in a dataset. It is very useful for exploring the data and analyzing relations between different variables. Since it creates a bivariate plot of each variable with others, the total number of plots created are n^2 where n is the number of variables

	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

```
sns.pairplot(df)
```

Seaborn Library

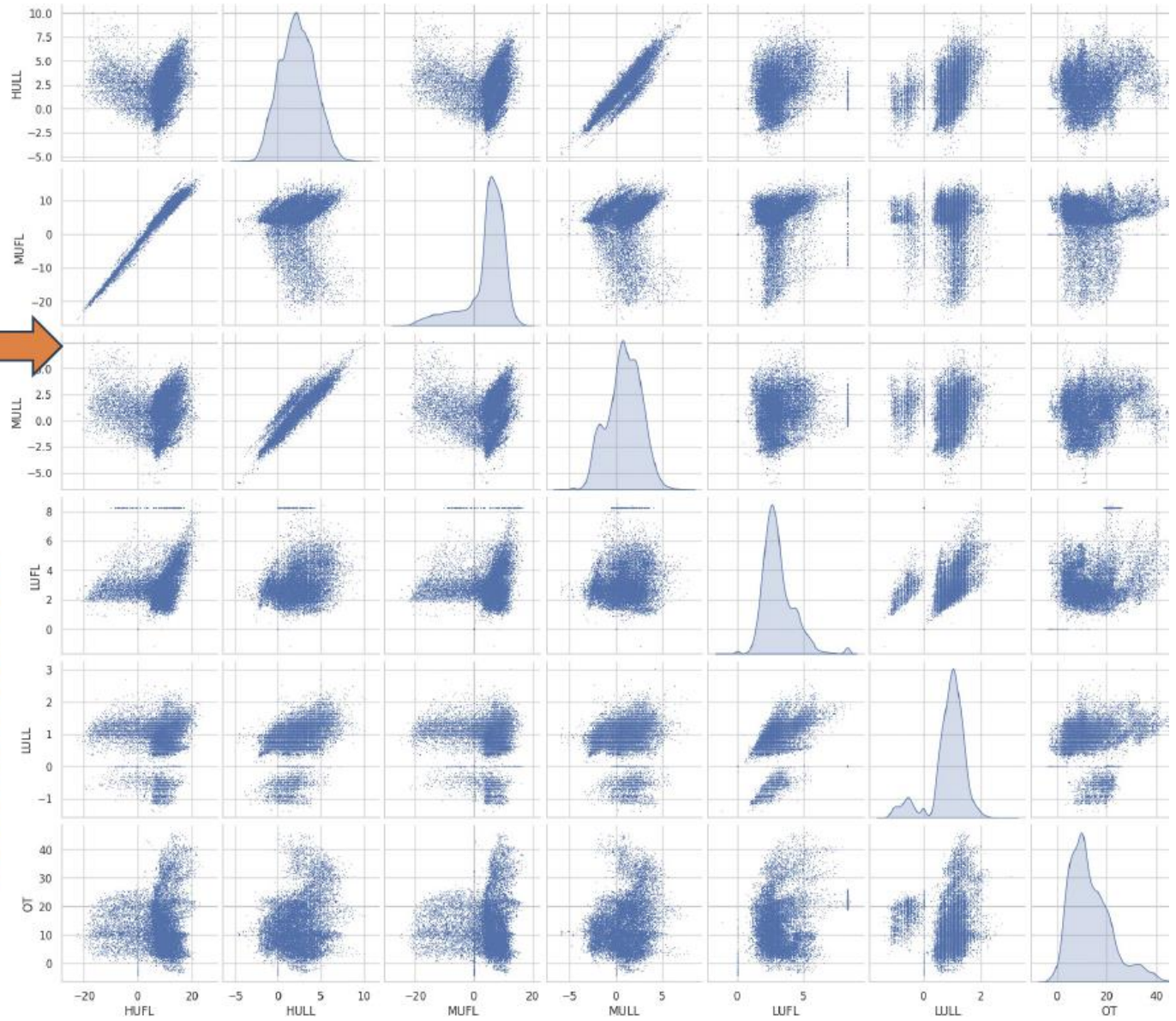


ETT Dataset: Pairplot

	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

```python
# Setting the aesthetic style of the plots
sns.set(style="whitegrid")

# Creating the pair plot with custom settings
pair_plot = sns.pairplot(df,
                         diag_kind="kde", # Kernel density estimate for diagonal
                         markers="o",     # Custom marker style
                         plot_kws=dict(linewidth=0, s=0.9), # line width and size
                         diag_kws=dict(fill=True)) # Shade for KDE plots

# Enhancing aesthetics with despine
sns.despine()
```

Seaborn Library



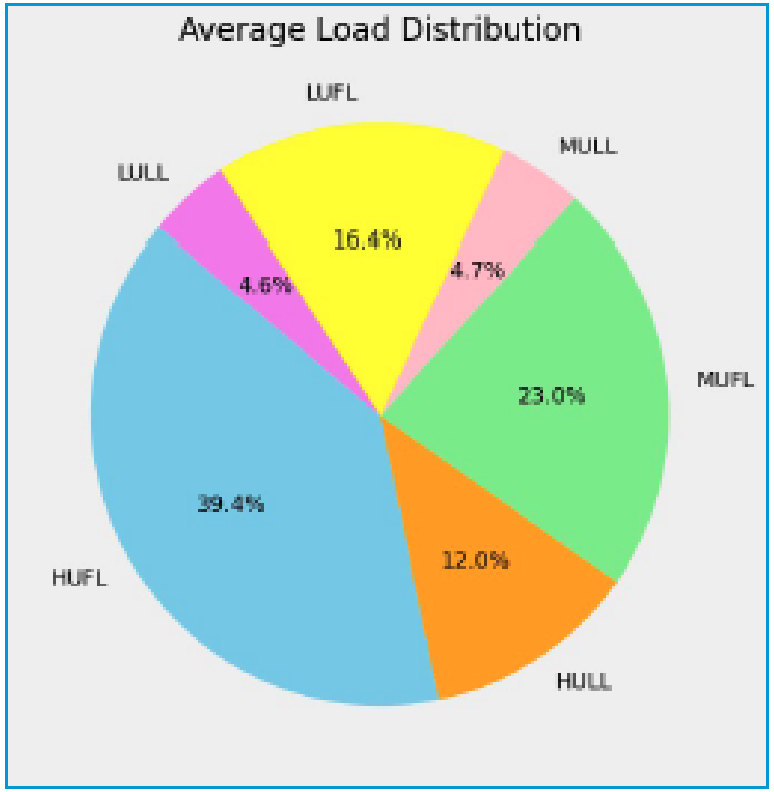
Pie Chart

ETT Dataset: Pie Chart

The pie chart showing the distribution of average loads for each load type (High Useful Load - HUFL, High Useless Load - HULL, Middle Useful Load - MUFL, Middle Useless Load - MULL, Low Useful Load - LUFL, Low Useless Load - LULL). Each segment of the pie represents the average proportion of a specific load type relative to the total average load. This visualization helps in understanding the relative contribution of each load type to the overall load in the dataset

```
# Calculating the average of each load type for the pie chart
average_loads = df[['HUFL', 'HULL', 'MUFL', 'MULL', 'LUFL', 'LULL']].mean()

# Setting up the figure for the pie chart
plt.figure(figsize=(8, 8))

# Creating the pie chart
plt.pie(average_loads, labels=average_loads.index, autopct='%1.1f%%', startangle=140, colors=colors)

# Adding a title
plt.title('Average Load Distribution')

# Display the plot
plt.show()
```

Matplotlib Library



Student Performance Dataset

Student Performance Dataset

How to Understand This Dataset

StudentsPerformance

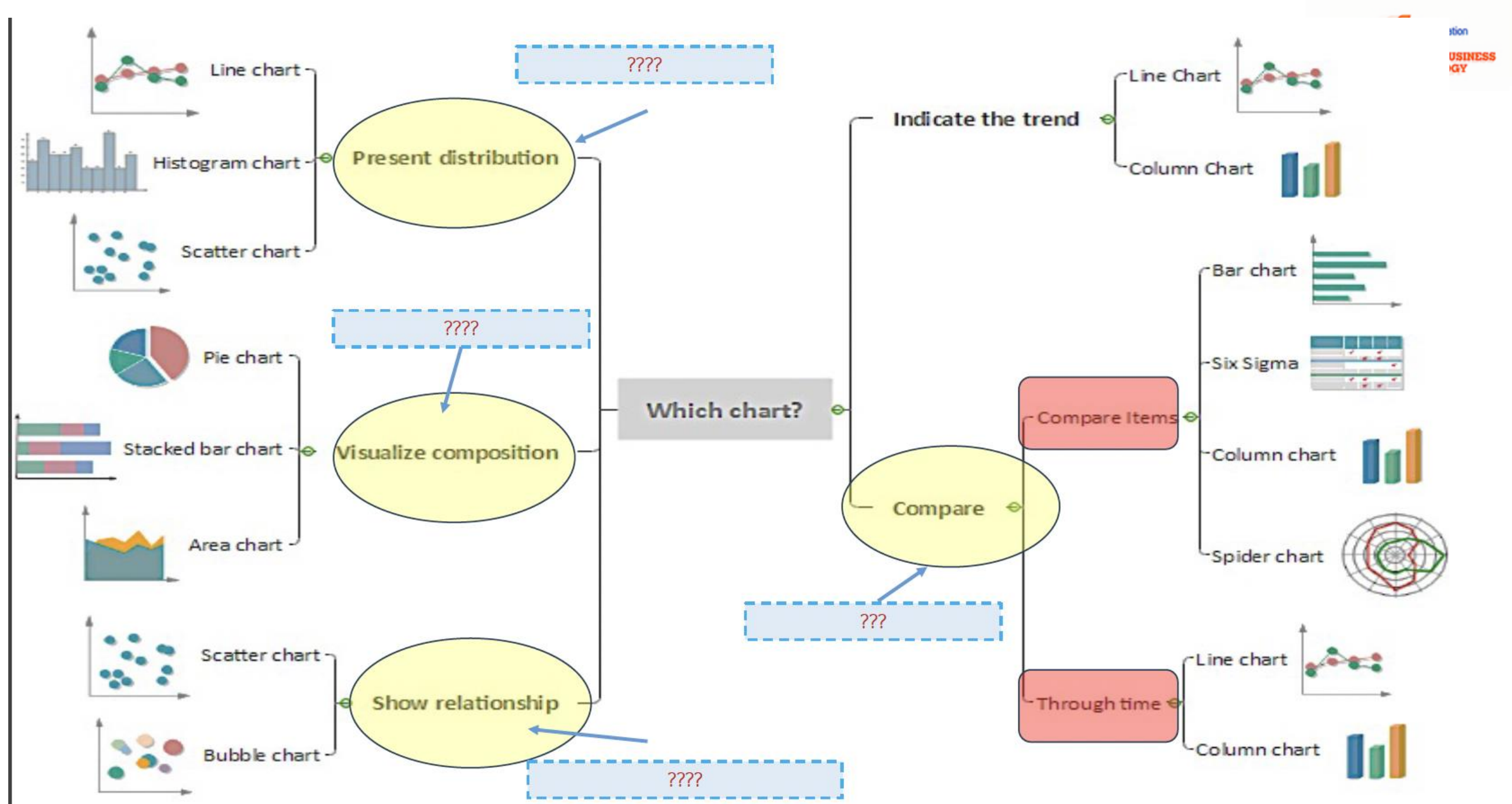
gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
female	group B	bachelor's degree	standard	none	72	72	74
female	group C	some college	standard	completed	69	90	88
female	group B	master's degree	standard	none	90	95	93
male	group A	associate's degree	free/reduced	none	47	57	44
male	group C	some college	standard	none	76	78	75
female	group B	associate's degree	standard	none	71	83	78
female	group B	some college	standard	completed	88	95	92
male	group B	some college	free/reduced	none	40	43	39

Student Performance Dataset

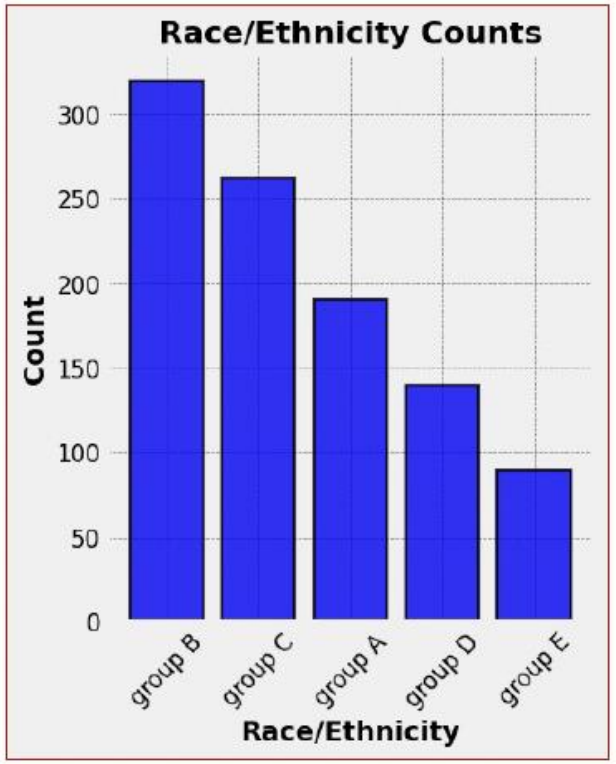
Prepare and load dataset

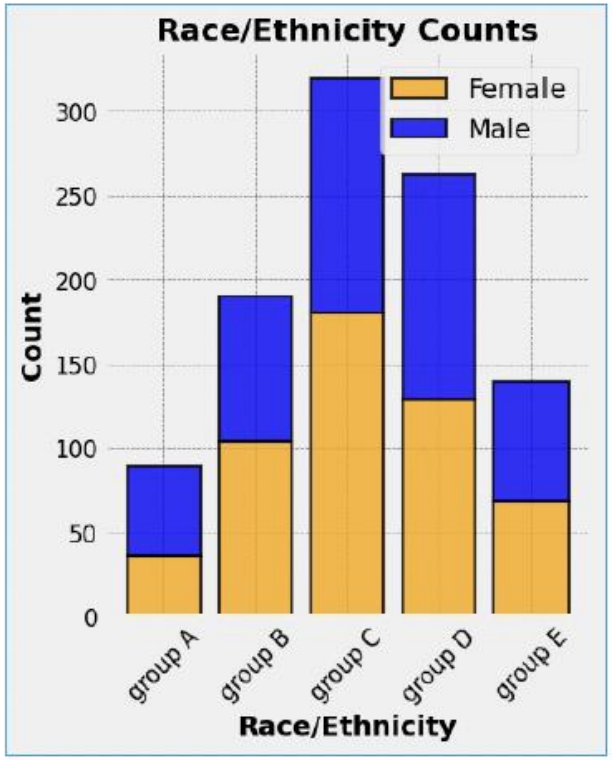
```
# Student Performance Dataset
df_st = pd.read_csv(r"/content/StudentsPerformance.csv")
df_st.describe()
```

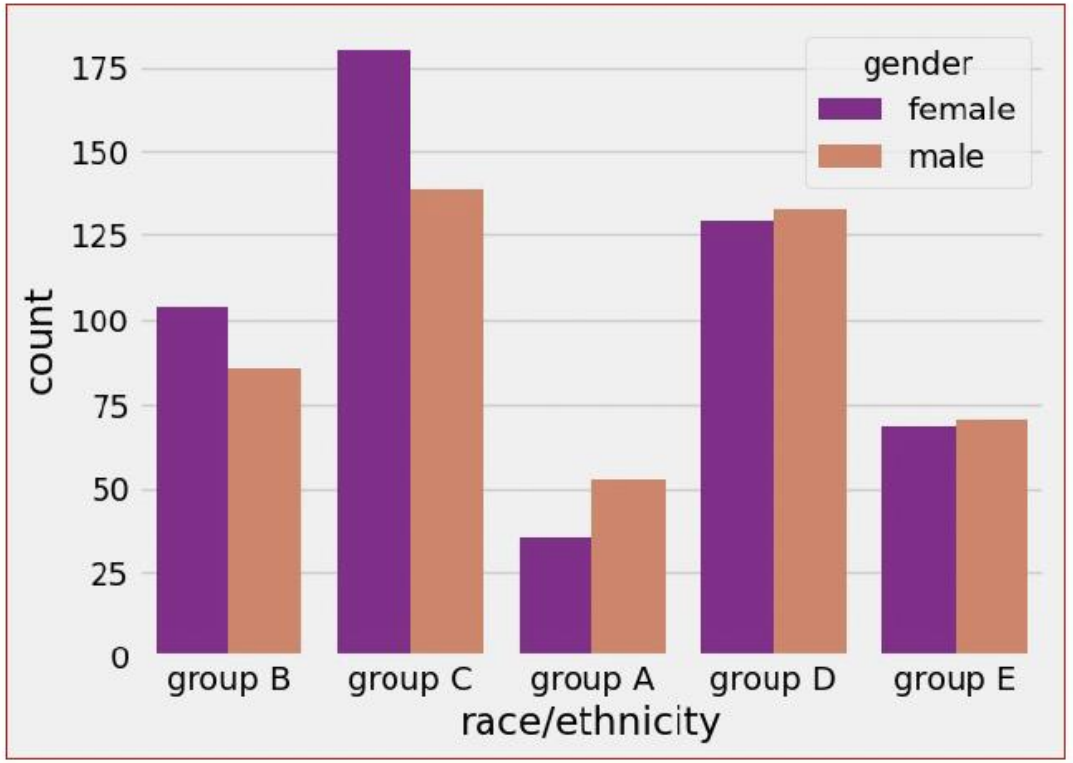
	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000


Which chart?
Present distribution
????
Line chart
Histogram chart
Scatter chart
Visualize composition
????
Pie chart
Stacked bar chart
Area chart
Show relationship
????
Scatter chart
Bubble chart
Compare
???
Indicate the trend
Line Chart
Column Chart
Compare Items
Bar chart
Six Sigma
Column chart
Spider chart
Through time
Line chart
Column chart

Student Performance Dataset







Student Performance Dataset: Bar Chart

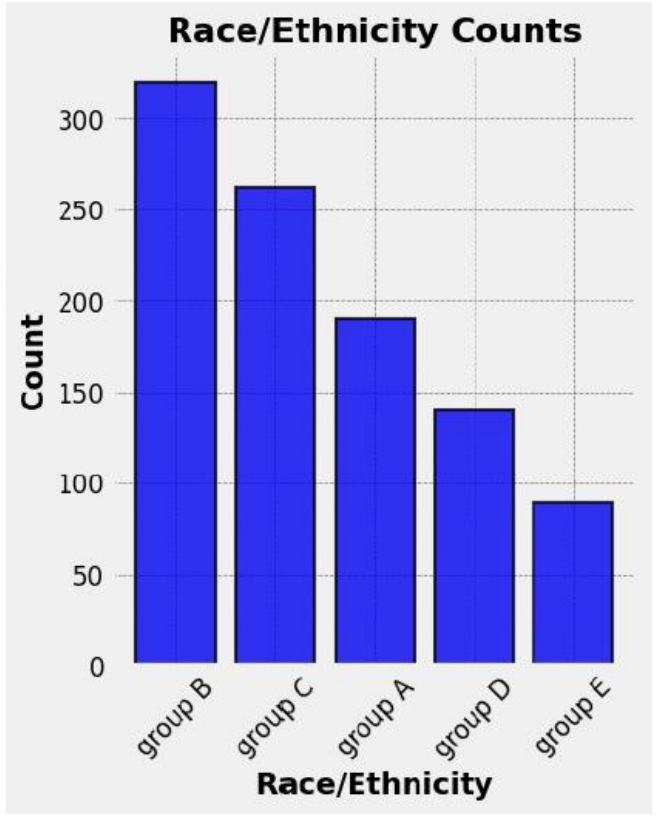
A bar chart is also one of the most used charts. It uses bars to show the change in the value of a particular variable wrt the other. This type of chart is generally used discrete or categorical data. A bar chart can be horizontal or vertical.

StudentsPerformance

gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
female	group B	bachelor's degree	standard	none	72	72	74
female	group C	some college	standard	completed	69	90	88
female	group B	master's degree	standard	none	90	95	93
male	group A	associate's degree	free/reduced	none	47	57	44
male	group C	some college	standard	none	76	78	75
female	group B	associate's degree	standard	none	71	83	78
female	group B	some college	standard	completed	88	95	92
male	group B	some college	free/reduced	none	40	43	39

Kaggle Dataset

Matplotlib Library



Student Performance Dataset: Bar Chart

A bar chart is also one of the most used charts. It uses bars to show the change in the value of a particular variable wrt the other. This type of chart is generally used discrete or categorical data. A bar chart can be horizontal or vertical.

```
[ ]  # Student Performance Data from Kaggle
     fig = plt.figure(figsize=(4, 5))
     plt.style.use('fivethirtyeight')
     plt.bar(
         x=df_st["race/ethnicity"].unique(),
         height=df_st["race/ethnicity"].value_counts().to_list(),
         color='blue',          # Set the color of the bars
         edgecolor='black',     # Set the color of the bar edges
         linewidth=1.5,         # Set the width of the bar edges
         alpha=0.8              # Set the transparency of the bars
     )

     plt.xlabel('Race/Ethnicity', fontsize=14, fontweight='bold')  # Set the x-axis label with font size and style
     plt.ylabel('Count', fontsize=14, fontweight='bold')           # Set the y-axis label with font size and style
     plt.title('Race/Ethnicity Counts', fontsize=16, fontweight='bold')  # Set the chart title with font size and style
     plt.xticks(fontsize=12, rotation = 45)    # Set the font size of the x-axis tick labels
     plt.yticks(fontsize=12)    # Set the font size of the y-axis tick labels
     plt.grid(True, linestyle='--', linewidth=0.5, alpha=0.5, color = "black")  # Display gridlines with a dashed style and reduced opacity

     plt.show()
```

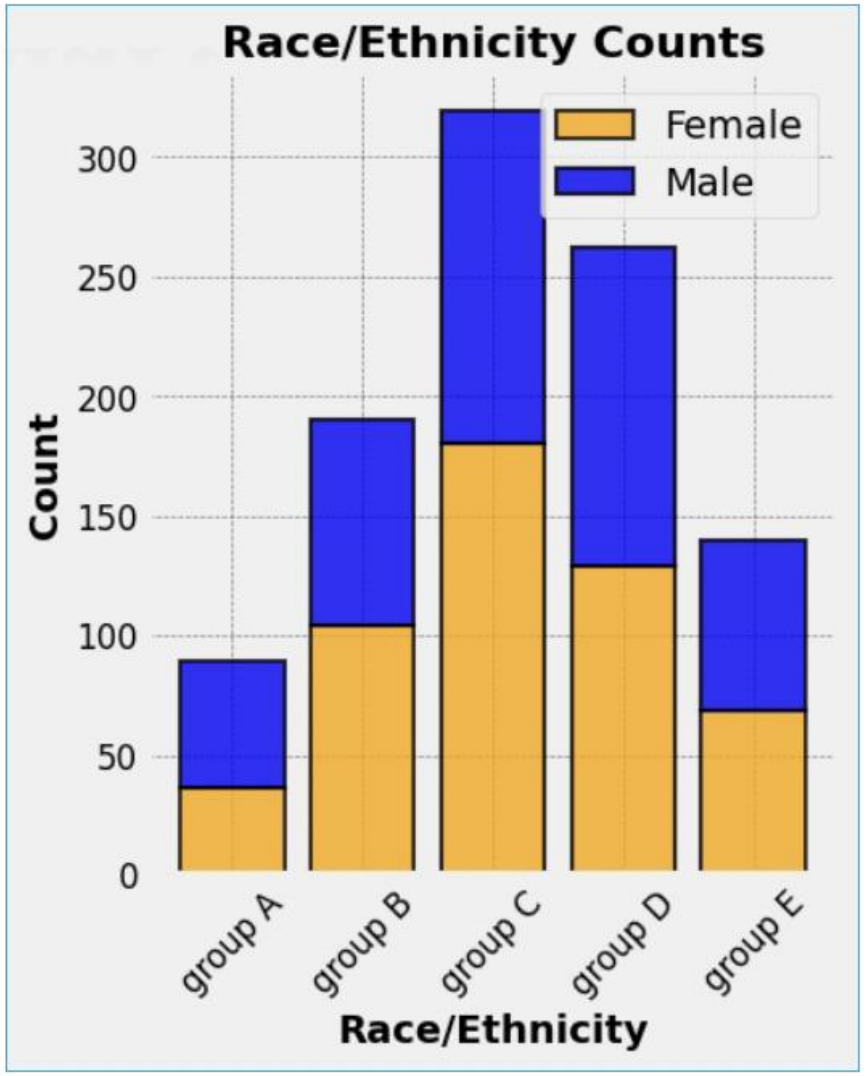
Student Performance Dataset: Bar Chart

StudentsPerformance

gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
female	group B	bachelor's degree	standard	none	72	72	74
female	group C	some college	standard	completed	69	90	88
female	group B	master's degree	standard	none	90	95	93
male	group A	associate's degree	free/reduced	none	47	57	44
male	group C	some college	standard	none	76	78	75
female	group B	associate's degree	standard	none	71	83	78
female	group B	some college	standard	completed	88	95	92
male	group B	some college	free/reduced	none	40	43	39

gender	female	male
race/ethnicity		
group A	36	53
group B	104	86
group C	180	139
group D	129	133
group E	69	71

Matplotlib Library



Student Performance Dataset: Bar Chart

```
fig = plt.figure(figsize=(4, 5))
plt.style.use('fivethirtyeight')
plt.bar(
    x=grouped_data.index,
    height= grouped_data["female"],
    label = "Female",
    color='orange',       # Set the color of the bars
    edgecolor='black',    # Set the color of the bar edges
    linewidth=1.5,        # Set the width of the bar edges
    alpha=0.8             # Set the transparency of the bars
)

plt.bar(
    x=grouped_data.index,
    height= grouped_data["male"],
    bottom = grouped_data["female"],
    label = "Male",
    color='blue',       # Set the color of the bars
    edgecolor='black',    # Set the color of the bar edges
    linewidth=1.5,        # Set the width of the bar edges
    alpha=0.8             # Set the transparency of the bars
)

plt.xlabel('Race/Ethnicity', fontsize=14, fontweight='bold')  # Set the x-axis label with font size and style
plt.ylabel('Count', fontsize=14, fontweight='bold')           # Set the y-axis label with font size and style
plt.title('Race/Ethnicity Counts', fontsize=16, fontweight='bold')  # Set the chart title with font size and style
plt.xticks(fontsize=12, rotation = 45)    # Set the font size of the x-axis tick labels
plt.yticks(fontsize=12)    # Set the font size of the y-axis tick labels
plt.grid(True, linestyle='--', linewidth=0.5, alpha=0.5, color = "black")# Display gridlines with a dashed style and reduced opacity
plt.legend()
```

Student Performance Dataset: Count Plot

A CountPlot is basically used to show the count of the number of observations of a categorical variable in a dataset. This type of plot can only be used with categorical variables.

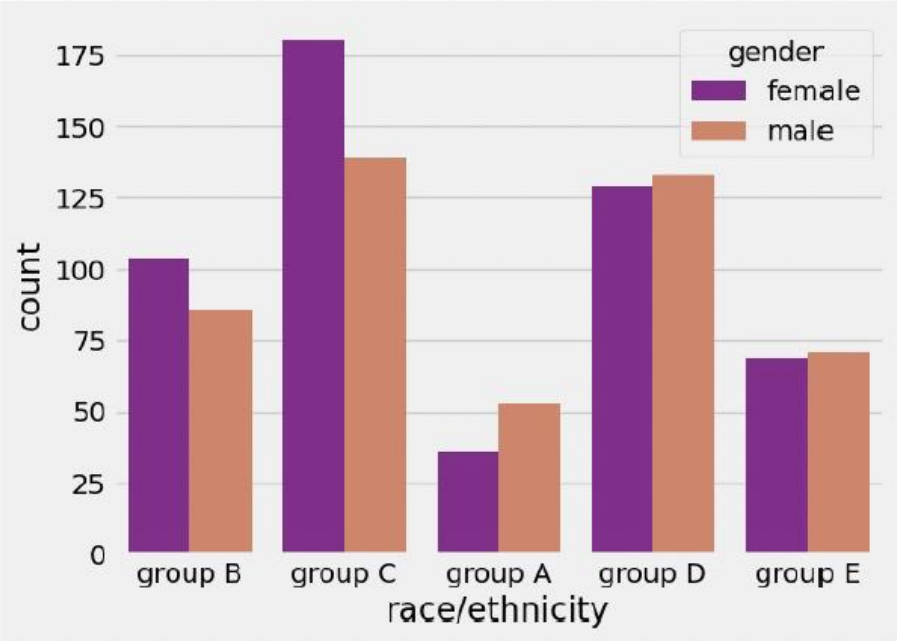
StudentsPerformance

gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
female	group B	bachelor's degree	standard	none	72	72	74
female	group C	some college	standard	completed	69	90	88
female	group B	master's degree	standard	none	90	95	93
male	group A	associate's degree	free/reduced	none	47	57	44
male	group C	some college	standard	none	76	78	75
female	group B	associate's degree	standard	none	71	83	78
female	group B	some college	standard	completed	88	95	92
male	group B	some college	free/reduced	none	40	43	39

Student Performance Kaggle Dataset

```
# Seaborn
sns.countplot(data = df_st, x = "race/ethnicity", hue = "gender",
              palette = "plasma")
```

```
<Axes: xlabel='race/ethnicity', ylabel='count'>
```



Seaborn Library

Iris Dataset

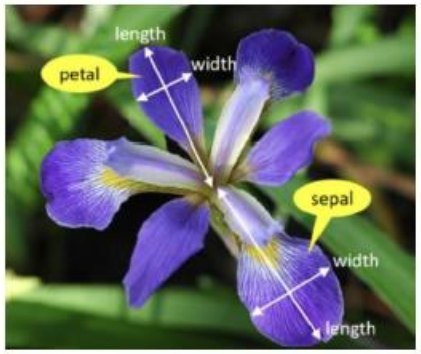
Iris Dataset

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa

Iris Versicolor

Iris Setosa

Iris Virginica



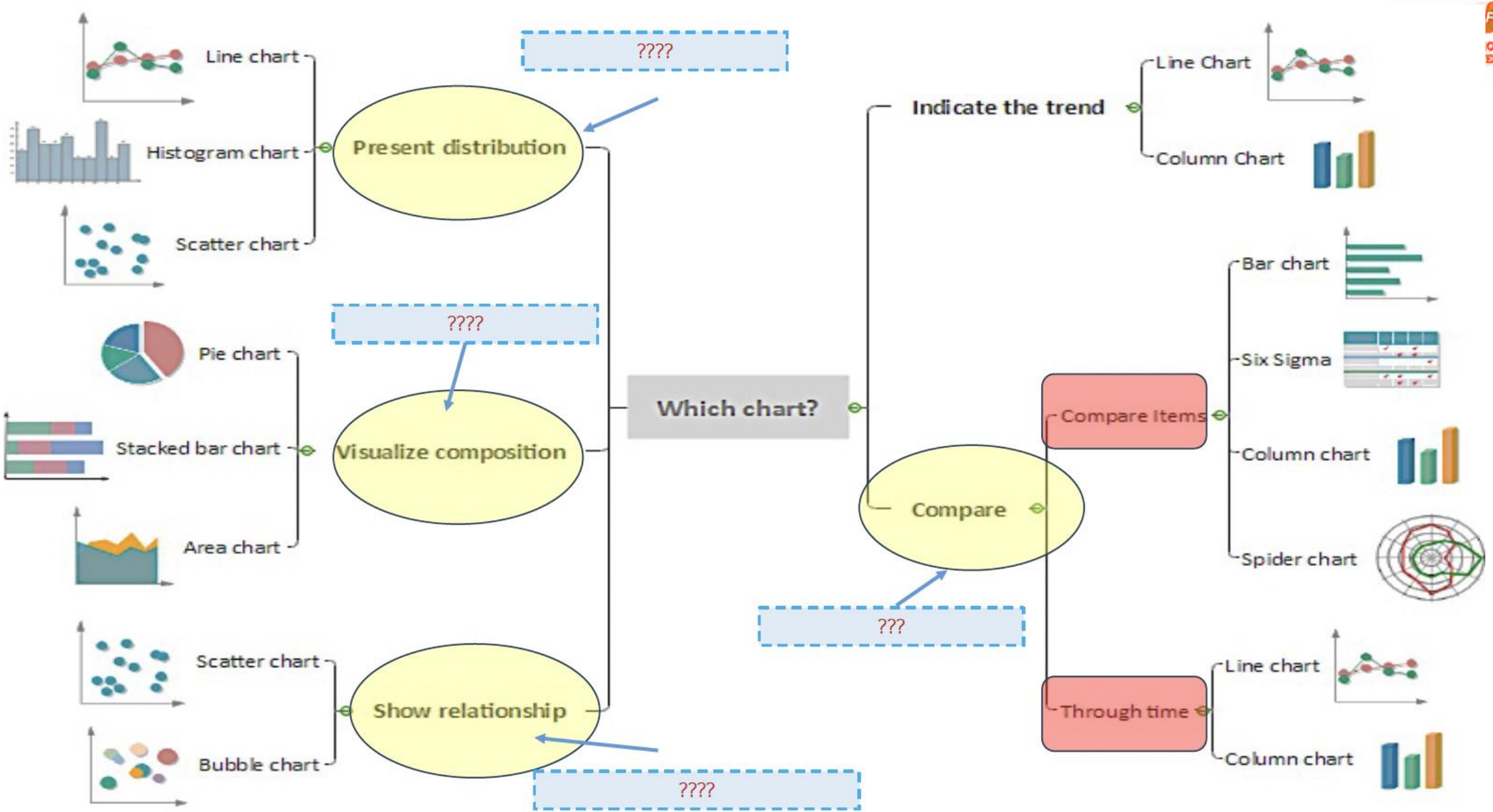
How to Understand This Dataset

Iris Dataset

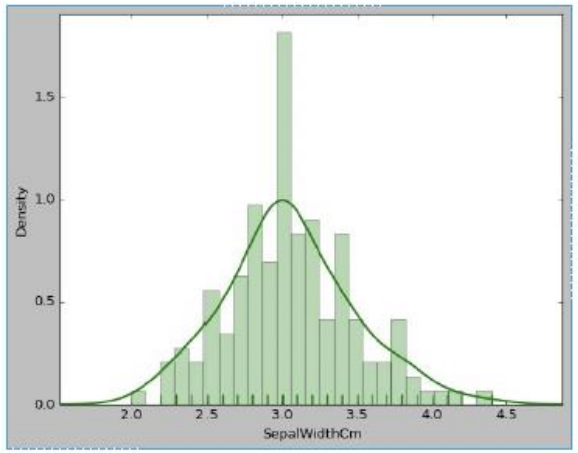
Prepare and load dataset

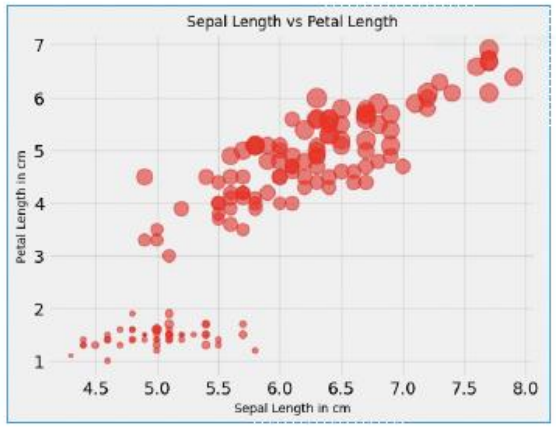
```
# Iris Dataset
# Student Performance Dataset
iris = pd.read_csv(r"/content/Iris.csv")
iris.describe()
```

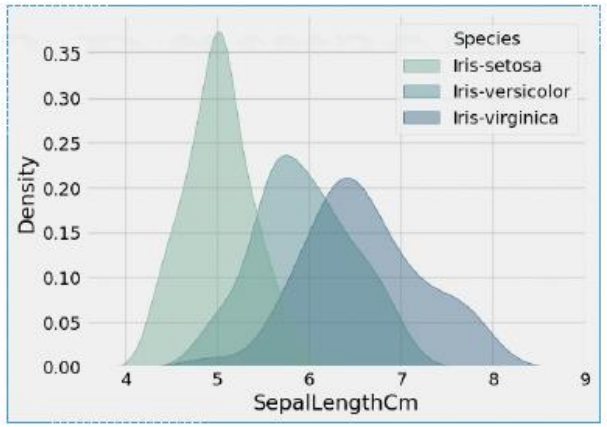
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

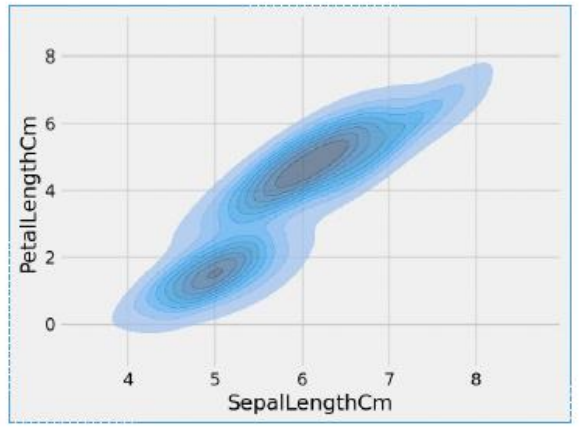

Which chart?
Present distribution
????
Line chart
Histogram chart
Scatter chart
Visualize composition
????
Pie chart
Stacked bar chart
Area chart
Show relationship
Scatter chart
Bubble chart
????
Compare
???
Indicate the trend
Line Chart
Column Chart
Compare Items
Bar chart
Six Sigma
Column chart
Spider chart
Through time
Line chart
Column chart

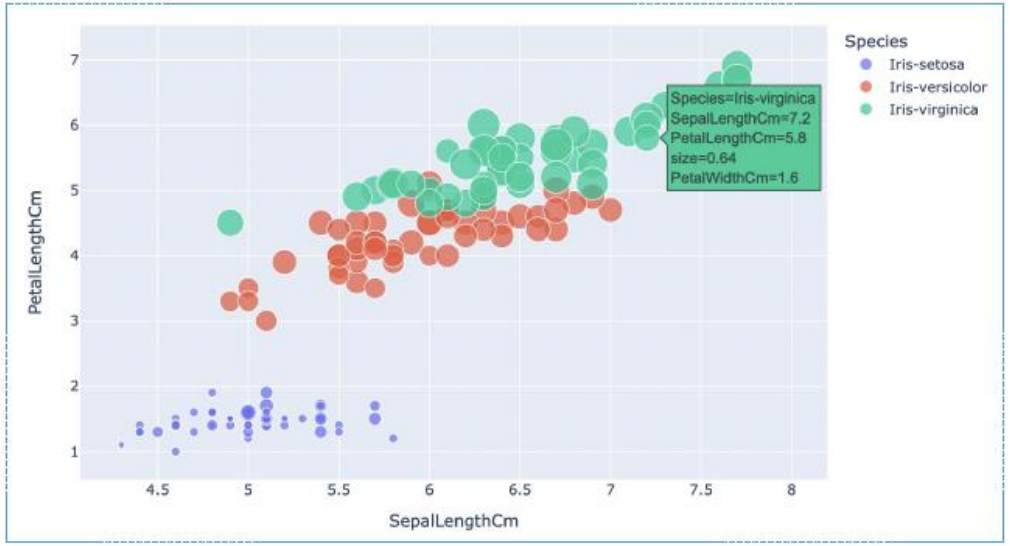
Iris Dataset

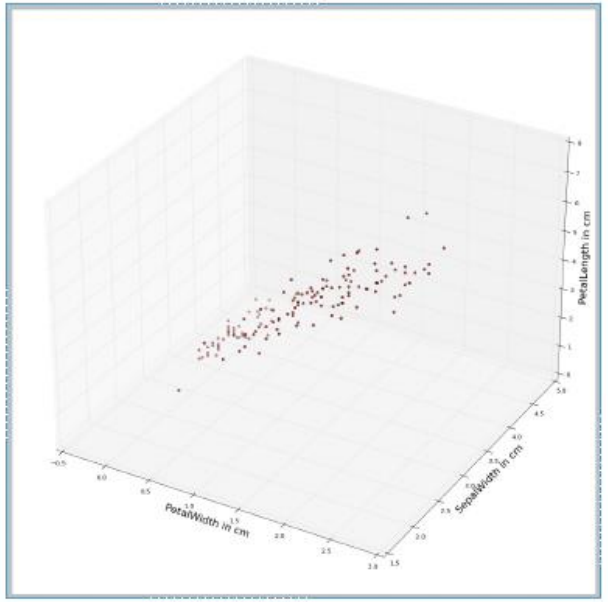


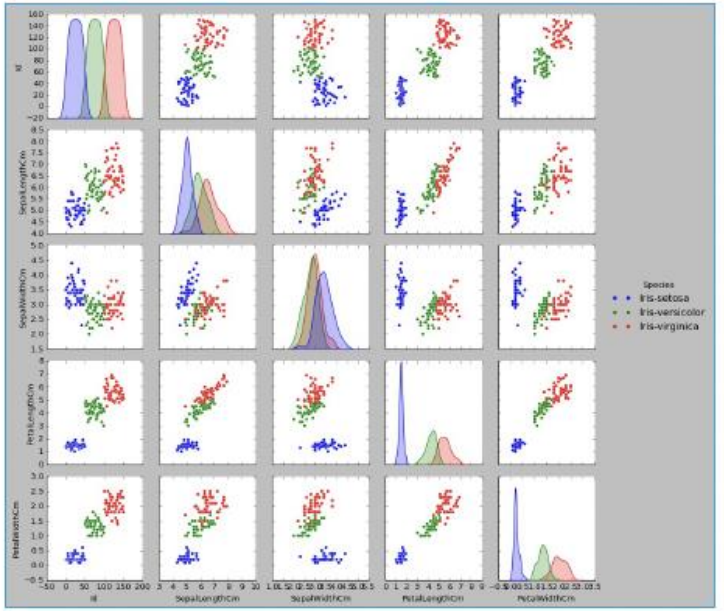












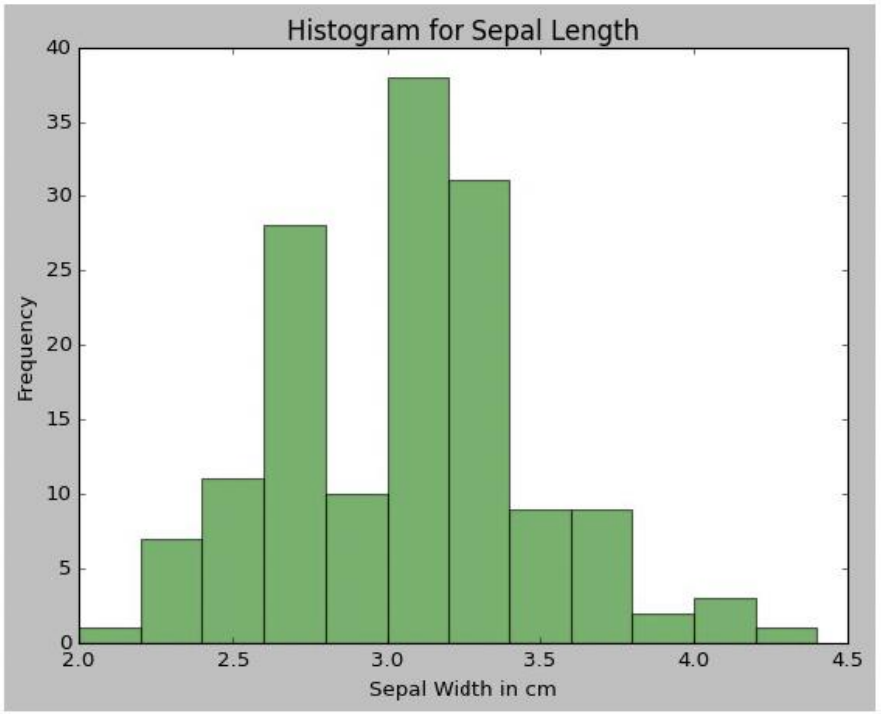
Iris Dataset: Histogram

This type of chart is used to observe the frequency distribution of a variable. The variable is divided into different intervals and the length of bars of the histogram show the frequency of the variable for a particular interval. Histograms can be used to observe discrete as well as continuous data. It is a great way to see how the values are being distributed and how skewed is the dataset.

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa

Iris Dataset

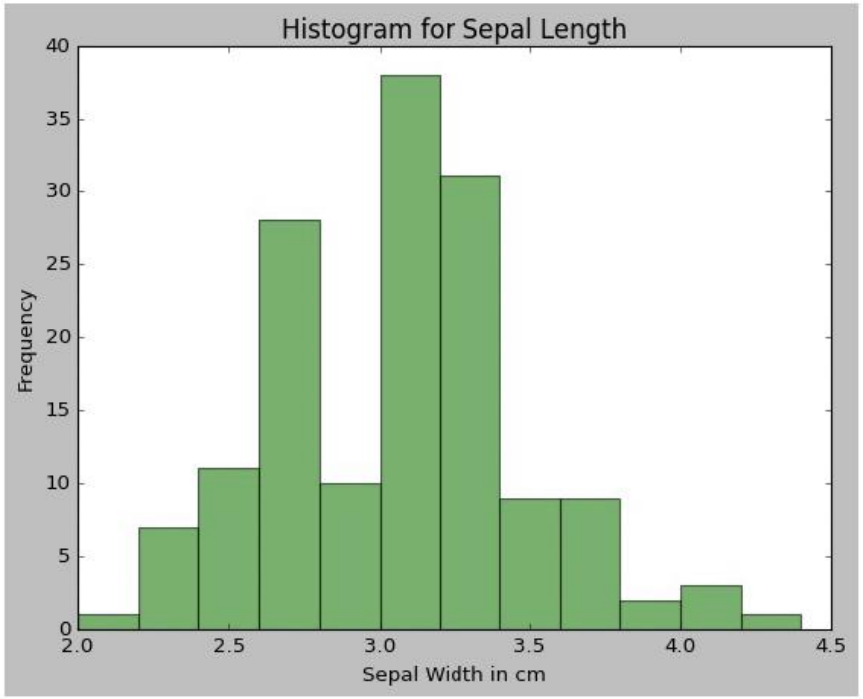
Matplotlib Library



Iris Dataset: Histogram

This type of chart is used to observe the frequency distribution of a variable. The variable is divided into different intervals and the length of bars of the histogram show the frequency of the variable for a particular interval. Histograms can be used to observe discrete as well as continuous data. It is a great way to see how the values are eing distributed and how skewed is the dataset.

```
iris = pd.read_csv("/content/Iris.csv")

# If bins of specific width are needed
bin_width = 0.2
bins = int((iris.SepalWidthCm.max()-iris.SepalWidthCm.min())/bin_width)
plt.style.use('classic')
plt.hist(iris.SepalWidthCm, bins = bins, color = "green", alpha = 0.6,
         orientation = "vertical", rwidth = 1)
plt.xlabel("Sepal Width in cm", size = 12, color = "black")
plt.ylabel("Frequency", size = 12, color = "black")
plt.title("Histogram for Sepal Length", size =15, color = "black")
plt.xticks(color = "black")
plt.yticks(color = "black")
plt.show()
```



Iris Dataset

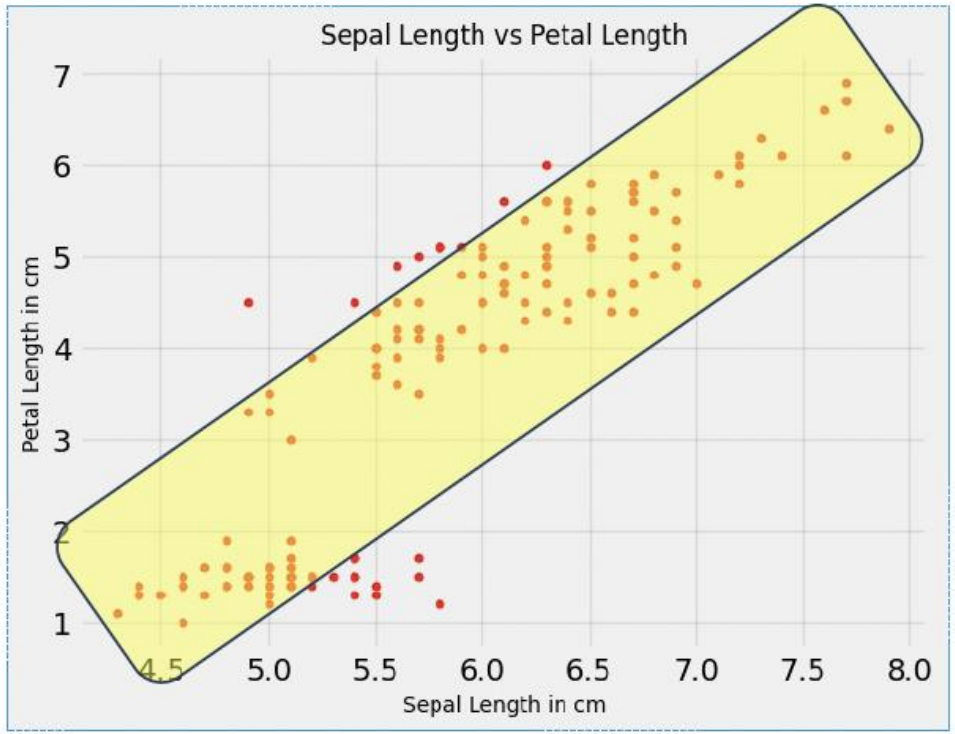
Matplotlib Library

Iris Dataset: Scatter

In a scatter plot, the data is scattered between the two axes. These type of plots are basically used to study the correlation between the two variables. If one variable increases or decreases with other, they are positively correlated. If one variable increases as the other decreases and vice versa, they are negatively correlated. If there is no such relation, they aren't related. It is one of the most common charts to observe the relation between two variables

```
# Downloading iris dataset

iris = pd.read_csv("/content/Iris.csv")
iris.describe()
```

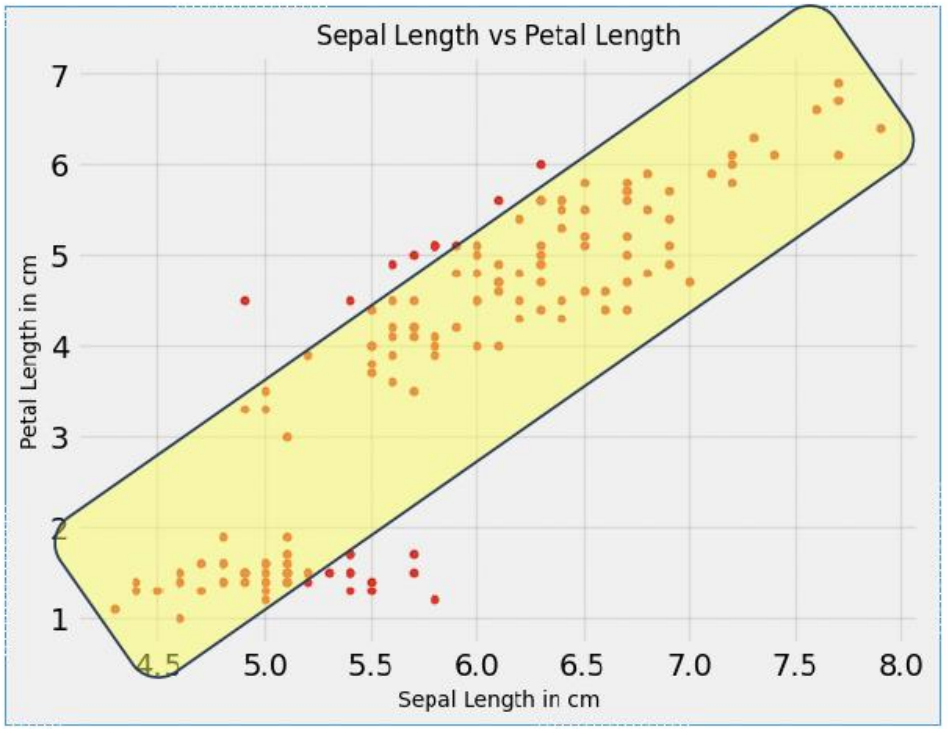
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000



Iris Dataset: Scatter

In a scatter plot, the data is scattered between the two axes. These type of plots are basically used to study the correlation between the two variables. If one variable increases or decreases with other, they are positively correlated. If one variable increases as the other decreases and vice versa, they are negatively correlated. If there is no such relation, they aren't related. It is one of the most common charts to observe the relation between two variables

```
# Using matplotlib

plt.scatter(iris.SepalLengthCm, iris.PetalLengthCm, marker = "o",
            color = "red", linewidths = 1, edgecolors = "red", s = 10)
plt.style.use('fivethirtyeight')
plt.xlabel("Sepal Length in cm", size = 10, color = "black")
plt.ylabel("Petal Length in cm", size = 10, color = "black")
plt.title("Sepal Length vs Petal Length", size =12, color = "black")
plt.xticks(color = "black")
plt.yticks(color = "black")
plt.grid(color = "grey", alpha = 0.2)
plt.show()
```



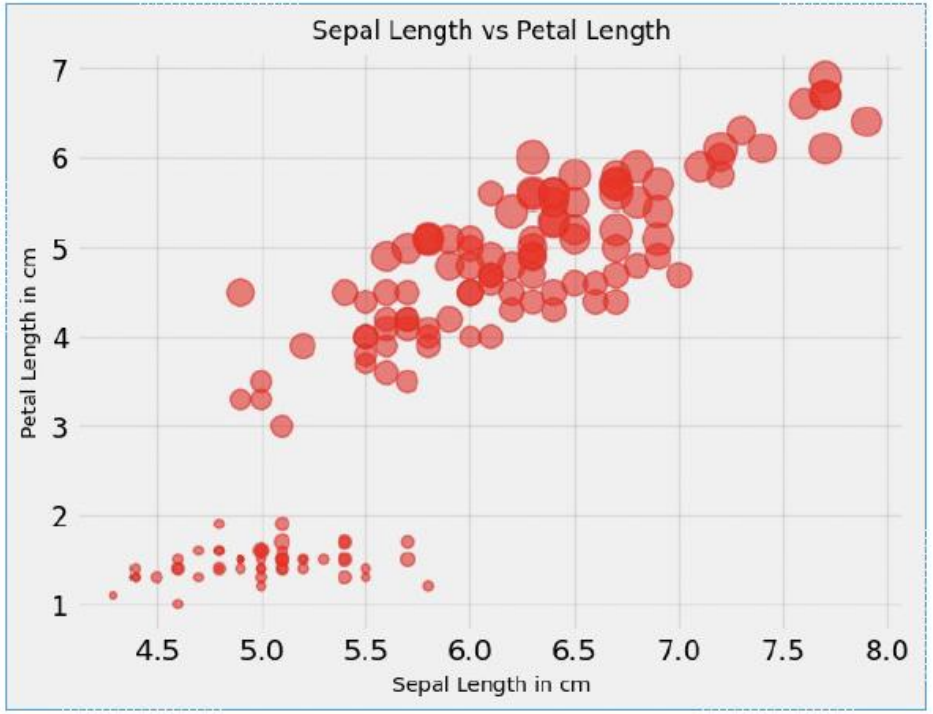
Iris Dataset: Bubble Chart

A bubble chart is similar to a scatter plot but is used when we need to observe the relation between three variables. In this chart, the size of the bubble corresponds to the third variable.

```
# Downloading iris dataset

iris = pd.read_csv("/content/Iris.csv")
iris.describe()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

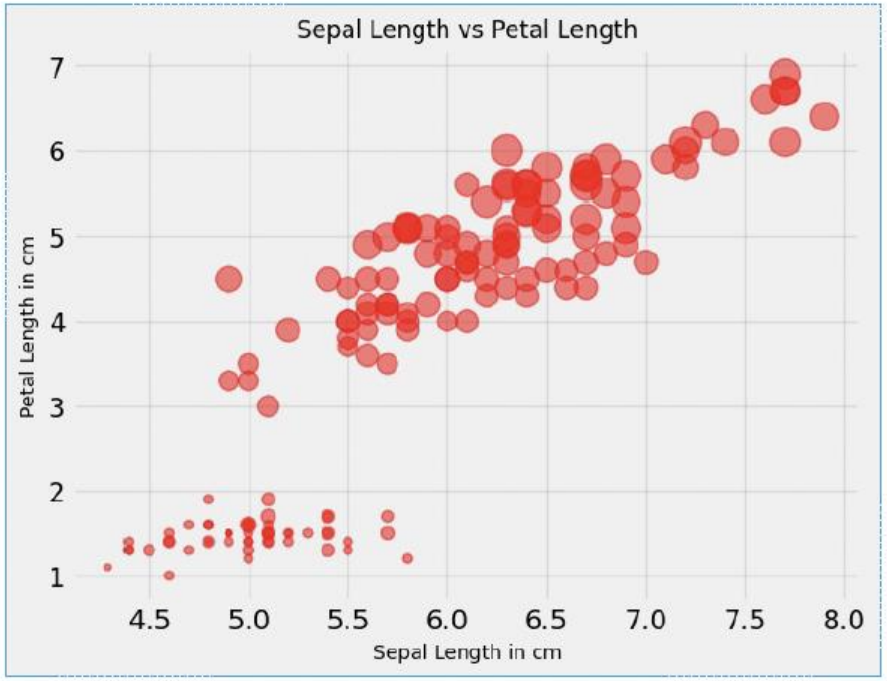
Matplotlib Library



Iris Dataset: Bubble Chart

A bubble chart is similar to a scatter plot but is used when we need to observe the relation between three variables. In this chart, the size of the bubble corresponds to the third variable.

```
from sklearn.preprocessing import MinMaxScaler

sizes = iris['PetalWidthCm']/iris['PetalWidthCm'].max()

# Here, the size of the bubble will show the Petal Width

plt.figure()
plt.scatter(iris.SepalLengthCm, iris.PetalLengthCm, marker = "o",
            color = "red", linewidths = 1, edgecolors = "red",
            s= sizes*250, alpha = 0.5)
plt.style.use('fivethirtyeight')
plt.xlabel("Sepal Length in cm", size = 10, color = "black")
plt.ylabel("Petal Length in cm", size = 10, color = "black")
plt.title("Sepal Length vs Petal Length", size =12, color = "black")
plt.xticks(color = "black")
plt.yticks(color = "black")
plt.grid(color = "grey", alpha = 0.2)
plt.show()
```



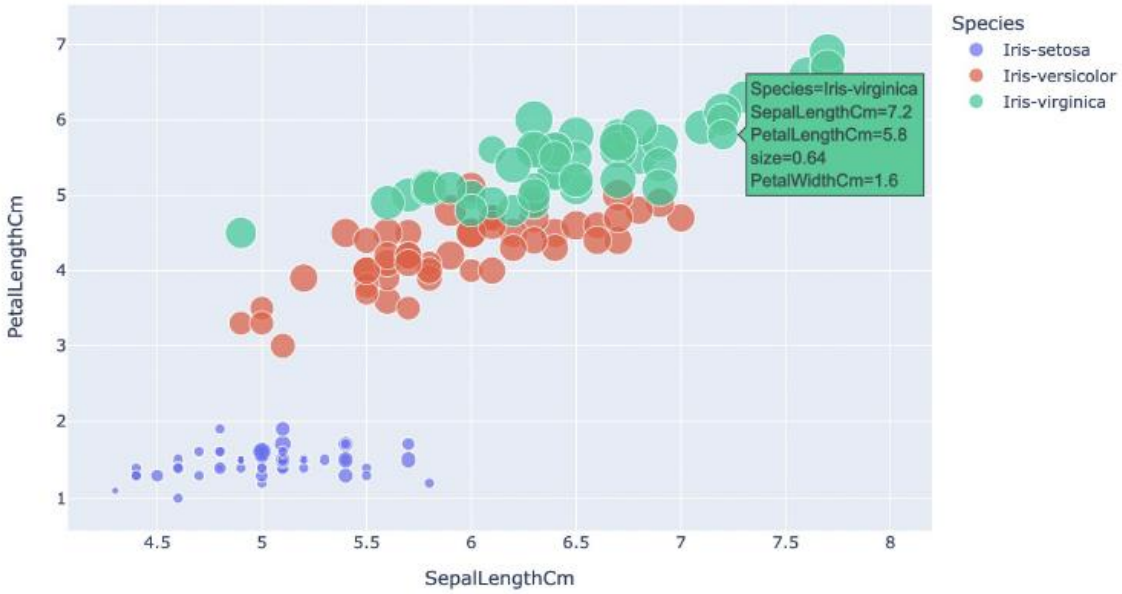
Matplotlib Library

Iris Dataset: Bubble Chart

A bubble chart is similar to a scatter plot but is used when we need to observe the relation between three variables. In this chart, the size of the bubble corresponds to the third variable.

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa

Iris Dataset




```
import plotly.express as px
fig = px.scatter(iris, x="SepalLengthCm", y="PetalLengthCm", color="Species",
                 size=sizes, hover_data=['PetalWidthCm'])
fig.show()
```

Plotly Library

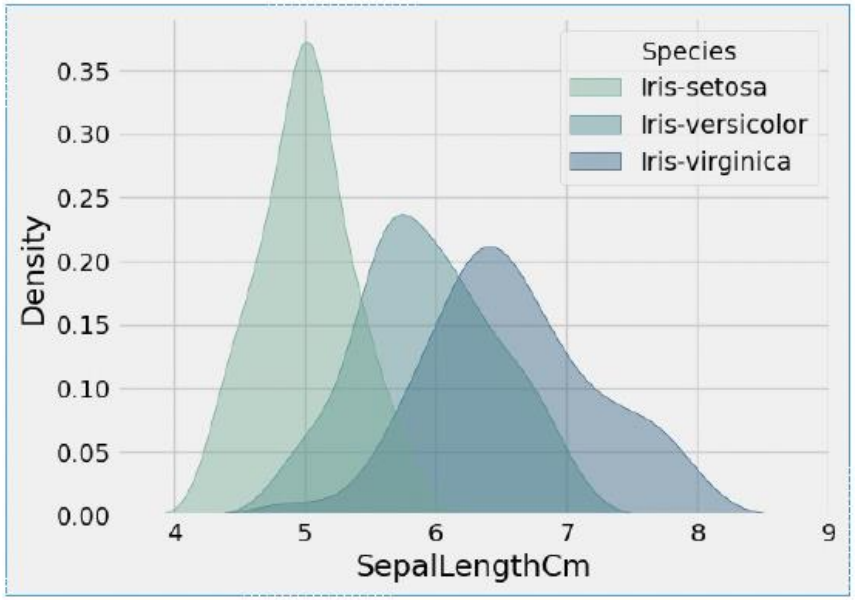
Iris Dataset: KDE Chart

A KDE Plot or Kernel Density Estimate plot is used to visualize the probability density of a continuous variable. We can plot probability density at different values of the variable or even the probability density against different values of some other variable. The latter one is known as a 2-dimensional KDE Plot which helps us to analyze the relation between two variables.

```
# Downloading iris dataset

iris = pd.read_csv("/content/Iris.csv")
iris.describe()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

Seaborn Library




```
sns.kdeplot(x=iris["SepalLengthCm"], hue = iris["Species"],
            linewidth = 0.5, fill = True, multiple = "layer", cbar = True,
            palette = "crest", alpha = 0.4)
```

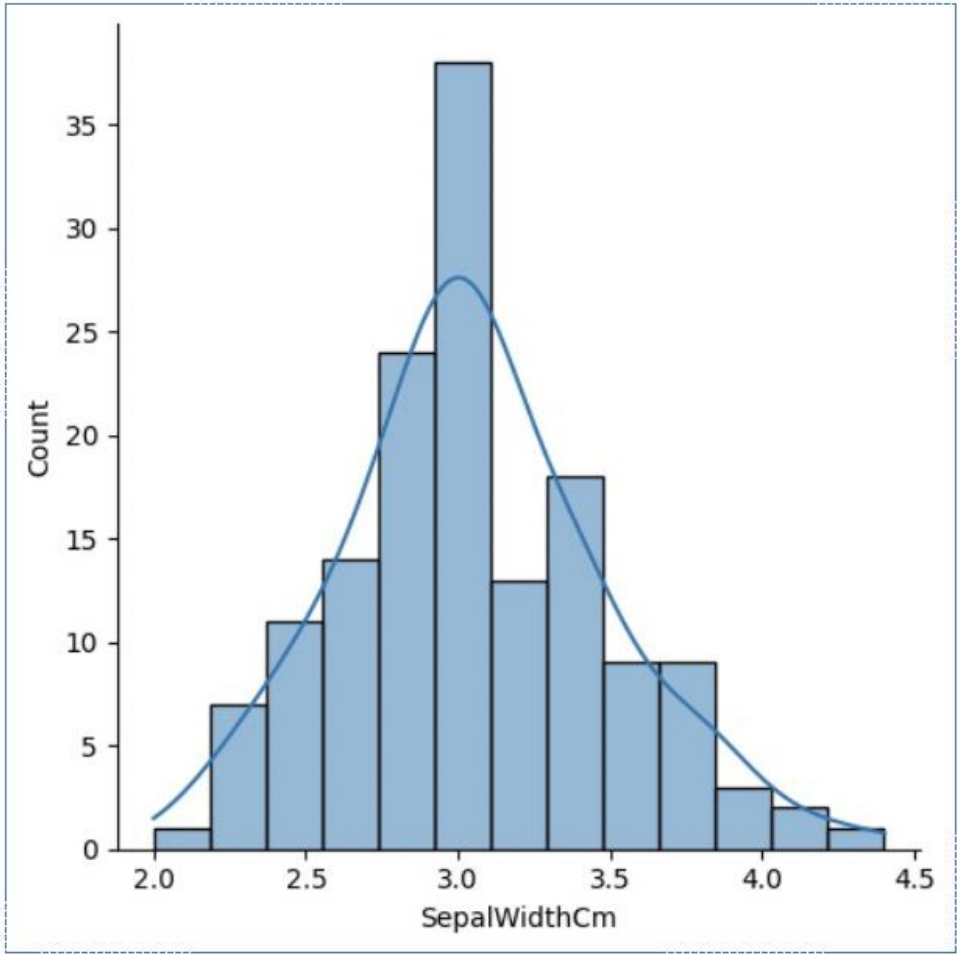
Iris Dataset: DisPlot Chart

A distplot is used to plot a histogram along with the kernel density estimate of a particular variable. Such a plot is used with discrete or continuous data. This function basically combines a histogram and kdeplot.

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa

Iris Dataset

Seaborn Library



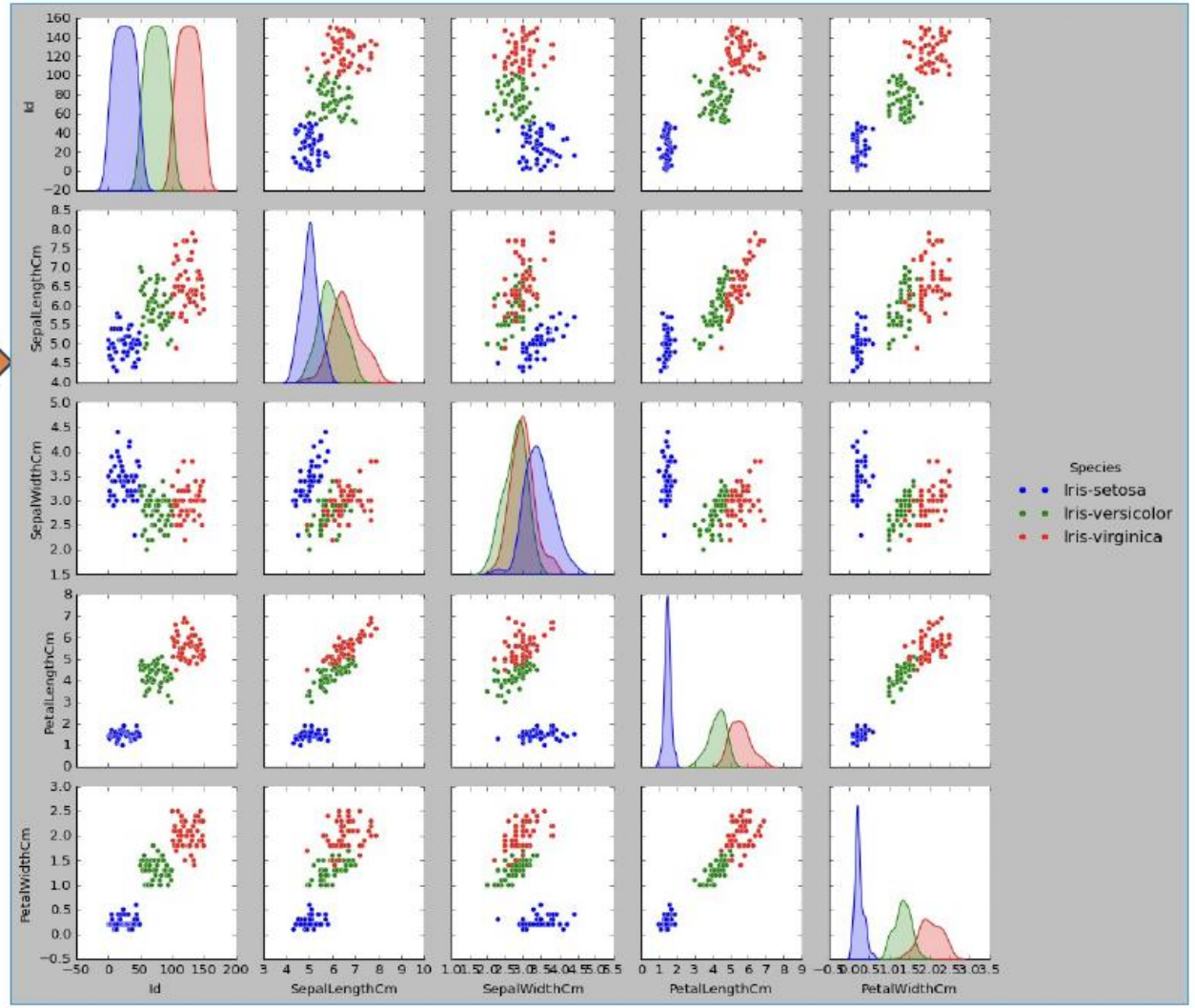
Iris Dataset: PariPlot Chart

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa

Iris Dataset

```
sns.pairplot(data=iris, hue = "Species", corner = False,
             diag_kws = {"linewidth":1, "fill":True},
             plot_kws = dict(marker = "o"))
```

Seaborn Library



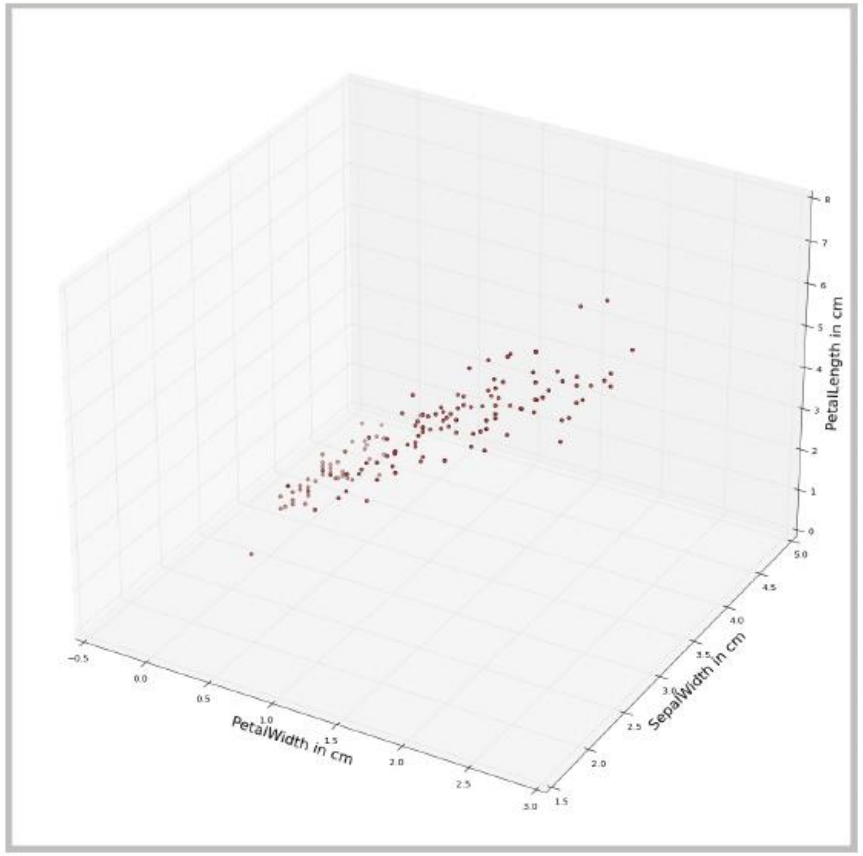
Iris Dataset: 3D Chart

A three dimensional plot is used when we need to observe relation between three numerical variables. To observe relation between two numeric and one categorical variable, we can use a 2-D plot with 'hue' parameter. But for more than 3 numeric variables, a 3-D plot is needed. We can plot scatter plots, surface plots, contour plots, etc. to show the variation.

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa

Iris Dataset

```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize = (20,20))
plt.style.use("classic")
ax = fig.add_subplot(111, projection='3d')
ax.scatter(iris["PetalWidthCm"], iris["SepalWidthCm"],iris["PetalLengthCm"], c = "red")
ax.set_xlabel("PetalWidth in cm", fontsize = 20)
ax.set_ylabel("SepalWidth in cm", fontsize = 20)
ax.set_zlabel("PetalLength in cm", fontsize = 20)
```



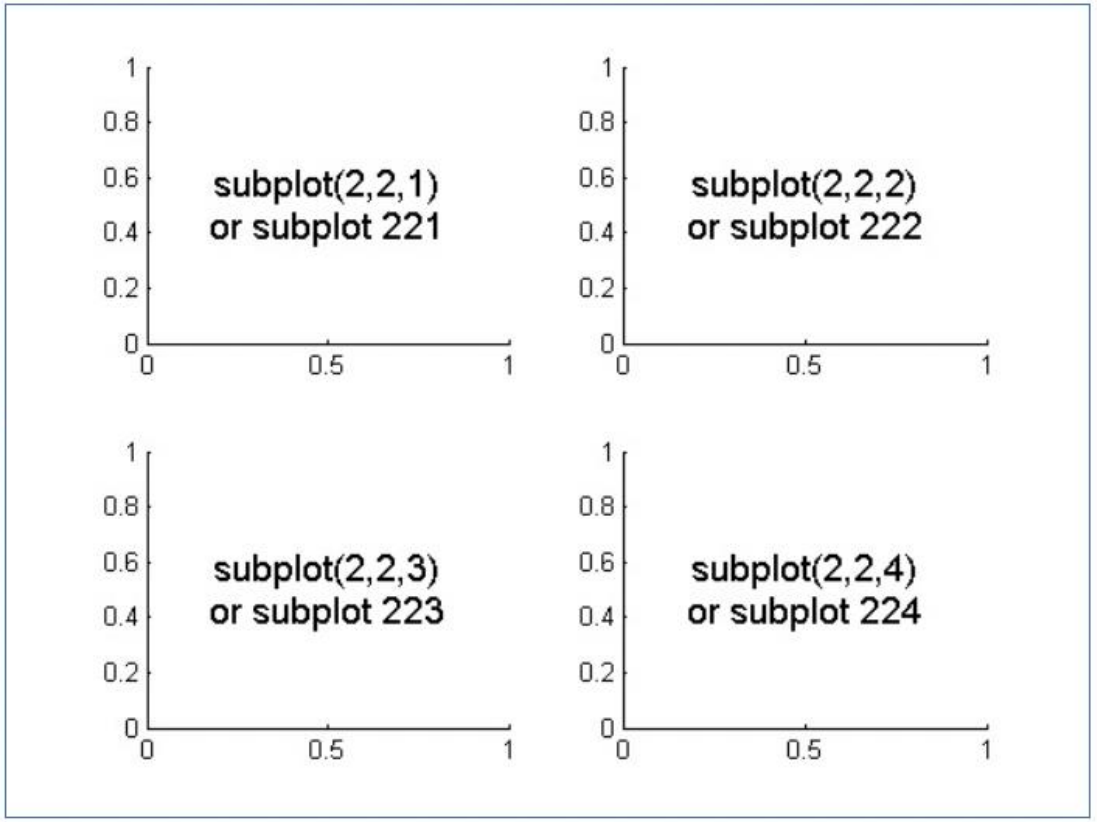
Iris Dataset: 3D Chart

A three dimensional plot is used when we need to observe relation between three numerical variables. To observe relation between two numeric and one categorical variable, we can use a 2-D plot with 'hue' parameter. But for more than 3 numeric variables, a 3-D plot is needed. We can plot scatter plots, surface plots, contour plots, etc. to show the variation.

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa

Iris Dataset

```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize = (20,20))
plt.style.use("classic")
ax = fig.add_subplot(111, projection='3d')
ax.scatter(iris["PetalWidthCm"], iris["SepalWidthCm"],iris["PetalLengthCm"], c = "red")
ax.set_xlabel("PetalWidth in cm", fontsize = 20)
ax.set_ylabel("SepalWidth in cm", fontsize = 20)
ax.set_zlabel("PetalLength in cm", fontsize = 20)
```



Other Charts

Word Cloud

This type of visualization gives us information about the context of the data. The size of different words represents the frequency or importance of those words. Such visualizations are quite important in analyzing data from social networking websites and also exploring data before performing Natural Language Processing.

```
import pandas as pd

df_fs = pd.read_csv('/content/five_star_reviews.csv')

df_fs.head()
```

	Rating	Review
0	5	You have to improve the camera performance, an...
1	5	It keeps hanging up completely phone(S10+) and...
2	5	Perfect for virtual conversation. We suggest t...
3	5	A very decent experience i've gained using the...
4	5	This app is very good as everyone knows. Here ...

5 ★ review for the app

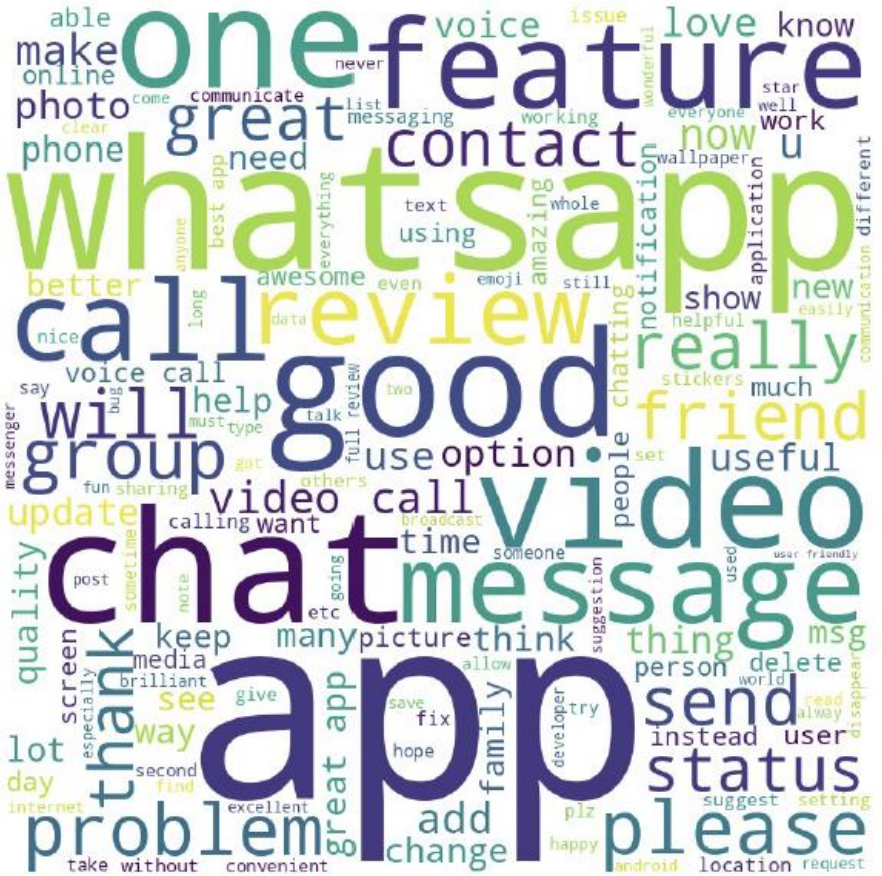
What is the main concern of customers?

Rating	Review
5	You have to improve the camera performance, and when we send photos through WhatsApp can't get helpful. If there's any other problem that i spot I will inform you Thank you.
5	It keeps hanging up completely phone(S10+) and after some time my phone starts working and then I a mostly if the,phone isn't hanged up and I'm on call none of us can hear each other, not Full Review
5	Perfect for virtual conversation. We suggest to add more features and secrets! And one more, please statement/media in a same group or private conversation. Sorry for my bad grammar. WhatsaFull Revie
5	A very decent experience i've gained using the app for nearly 7 years. To be honest it is one of the best comes handy. Just one issue plz fix the custom wallpaper. Otherwise everythinFull Review

Word Cloud

This type of visualization gives us information about the context of the data. The size of different words represents the frequency or importance of those words. Such visualizations are quite important in analyzing data from social networking websites and also exploring data before performing Natural Language Processing.

Rating	Review
5	You have to improve the camera performance, and when we send photos through WhatsApp can't get helpful. If there's any other problem that i spot I will inform you Thank you.
5	It keeps hanging up completely phone(S10+) and after some time my phone starts working and then I a mostly if the,phone isn't hanged up and I'm on call none of us can hear each other, not Full Review
5	Perfect for virtual conversation. We suggest to add more features and secrets!😆 And one more, please statement/media in a same group or private conversation. Sorry for my bad grammar. WhatsaFull Revie
5	A very decent experience i've gained using the app for nearly 7 years. To be honest it is one of the best comes handy. Just one issue plz fix the custom wallpaper. Otherwise everythinFull Review



5 ★ review for the app

What is the main concern of customers?

Word Cloud

This type of visualization gives us information about the context of the data. The size of different words represents the frequency or importance of those words. Such visualizations are quite important in analyzing data from social networking websites and also exploring data before performing Natural Language Processing.

```
import pandas as pd

df_os = pd.read_csv('/content/one_star_reviews.csv')

df_os.head()
```

	Rating	Review
0	1	Latest update broke the photo taking function....
1	1	It was one of my favourite app. Easy and secur...
2	1	WhatsApp has a major glitch or bug that is hor...
3	1	It was a good app for communication but it is ...
4	1	A almost a year passed and the bug I reported ...

1 ★ review for the app

What is the main concern of customers?

Rating	Review
1	Latest update broke the photo taking function. I do not use the Phone's Auto Rotate. have to press the rotate button that appears at the Shortcuts (Extreme rightFull Revie
1	It was one of my favourite app. Easy and secure one. But now i am facing the probler voice, i have to keep open the whatsapp during the call. Second is that there is Full R
1	WhatsApp has a major glitch or bug that is horrible for user. For eg, whenever you list how much you try to open the lock screen it will be always black, black, blacFull Revi
1	It was a good app for communication but it is not good since last few months when it messages that I had broadcasted to my recipients were no longer been seen or read
1	A almost a year passed and the bug I reported in march is still in the app. I don't get i rotates the picture(s) automatically. It's really annoying and u reallyFull Review

Word Cloud

This type of visualization gives us information about the context of the data. The size of different words represents the frequency or importance of those words. Such visualizations are quite important in analyzing data from social networking websites and also exploring data before performing Natural Language Processing.

Rating	Review
1	Latest update broke the photo taking function. I do not use the Phone's Auto Rotate. have to press the rotate button that appears at the Shortcuts (Extreme rightFull Revie
1	It was one of my favourite app. Easy and secure one. But now i am facing the probler voice, i have to keep open the whatsapp during the call. Second is that there is Full R
1	WhatsApp has a major glitch or bug that is horrible for user. For eg, whenever you list how much you try to open the lock screen it will be always black, black, blacFull Revi
1	It was a good app for communication but it is not good since last few months when it messages that I had broadcasted to my recipients were no longer been seen or read
1	A almost a year passed and the bug I reported in march is still in the app. I don't get i rotates the picture(s) automatically. It's really annoying and u reallyFull Review

1 ★ review for the app

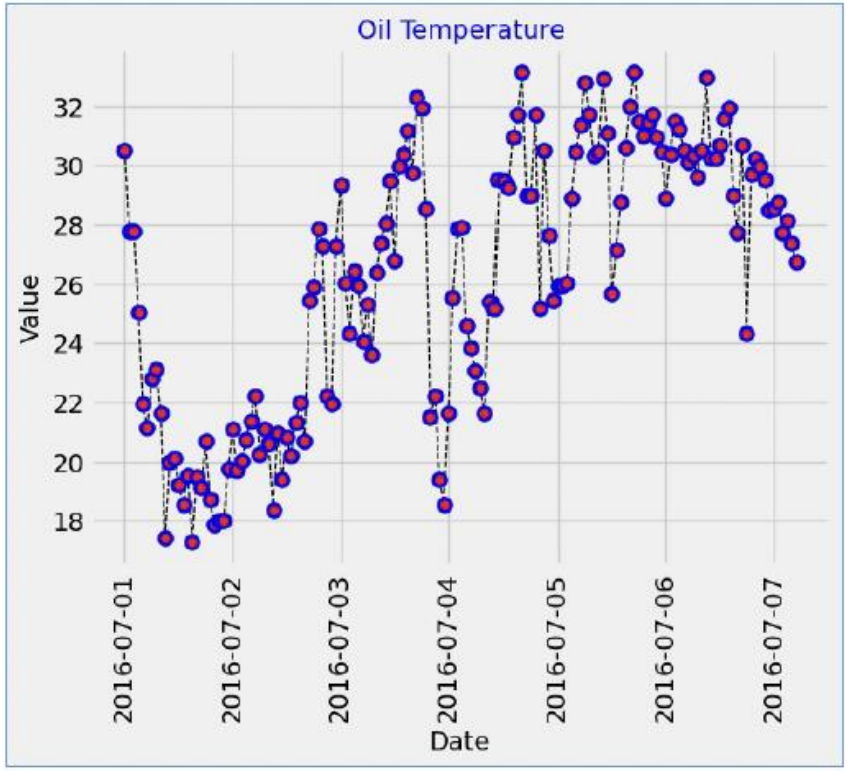
What is the main concern of customers?

Tips for Right Charts

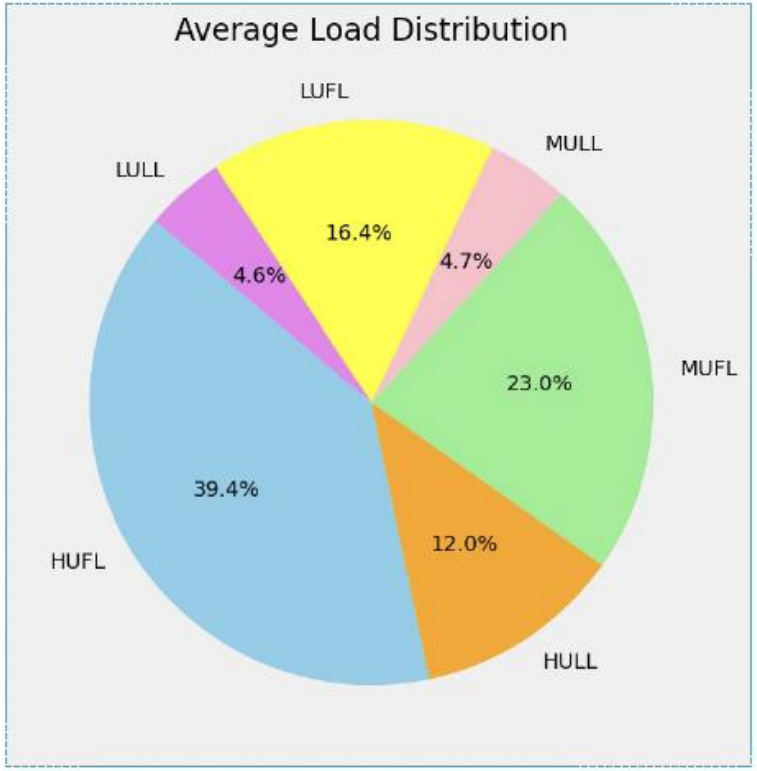
Tips for Right Charts

FPT Education
FPT SCHOOL OF BUSINESS & TECHNOLOGY

Type 1: showing change over time

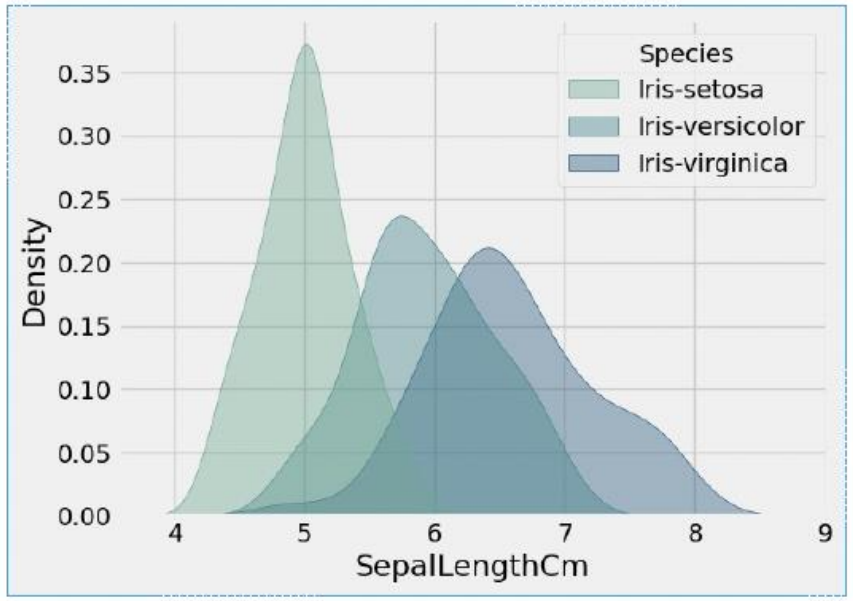


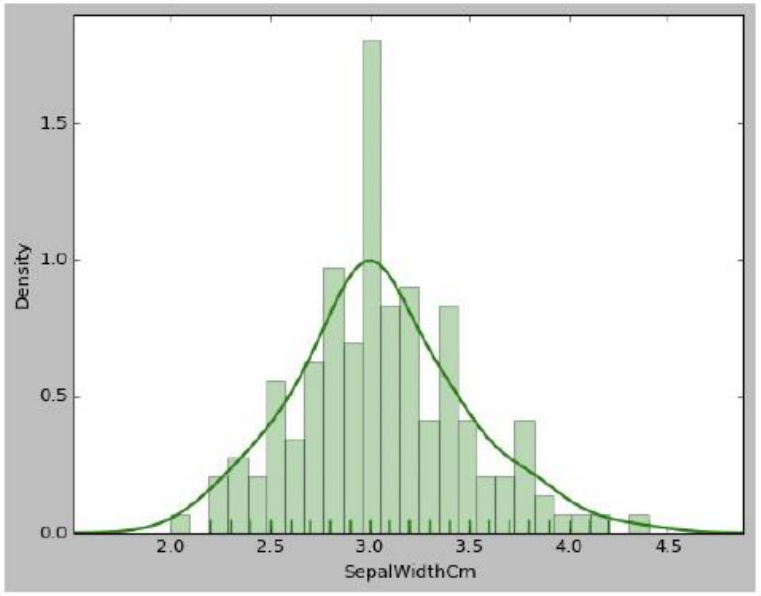
Type 2: showing a part-to-whole composition

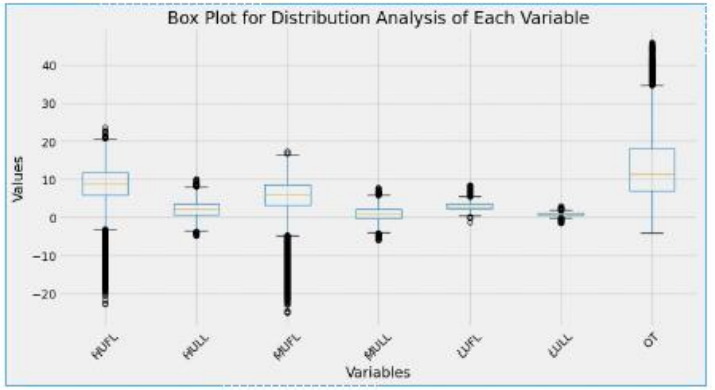


Tips for Right Charts

Type 3: How Data Distributed

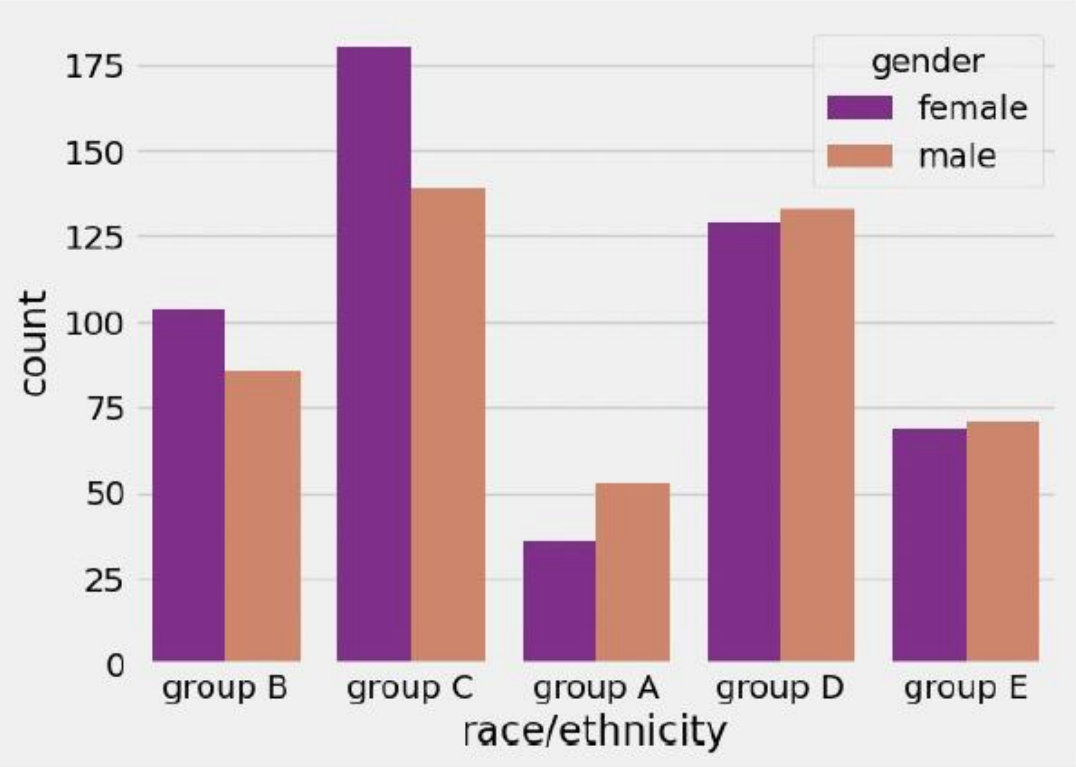


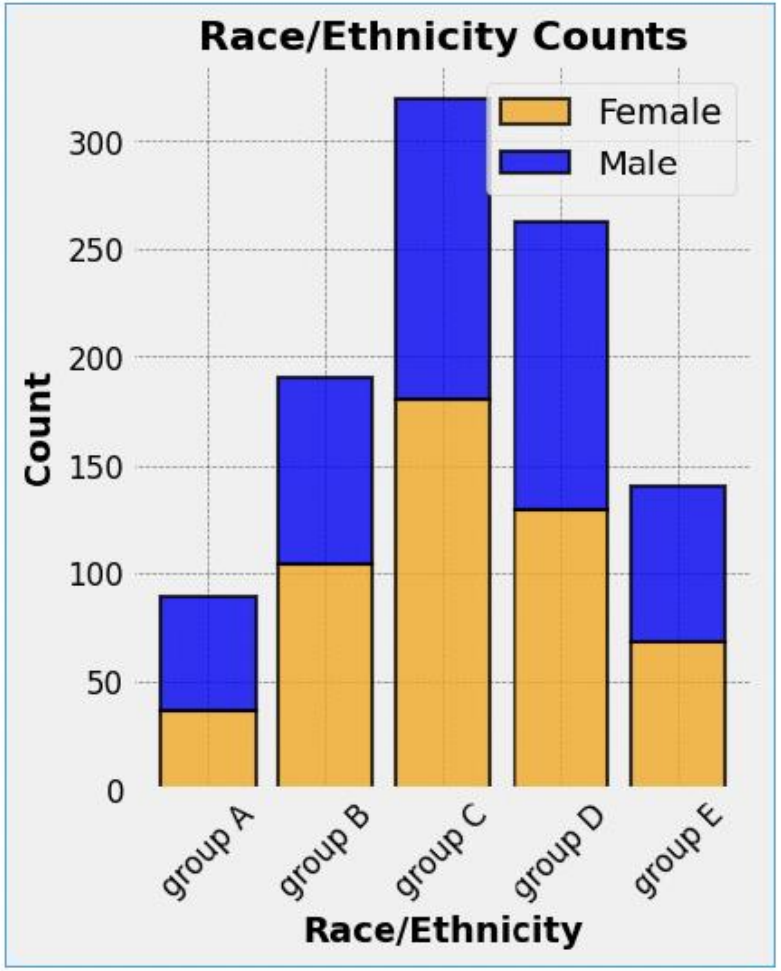




Tips for Right Charts

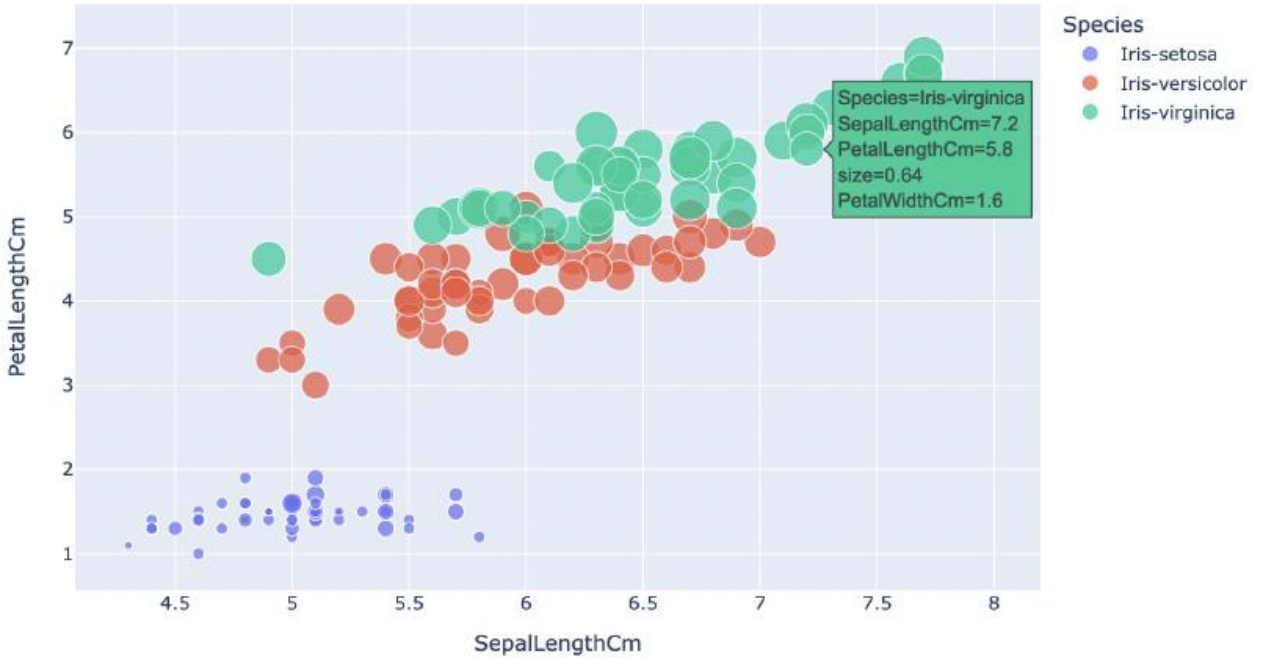
Type 4: Comparing Values Between Groups

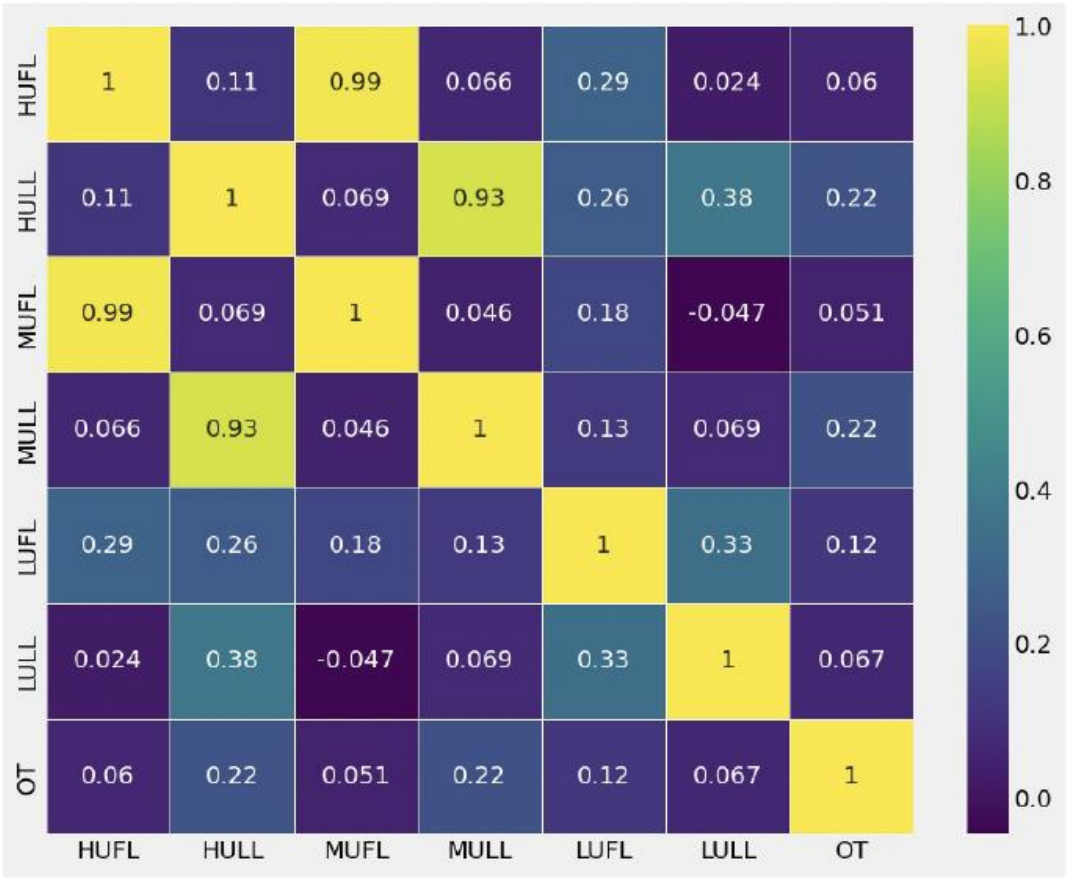




Tips for Right Charts

Type 5: Observing relationships between variables

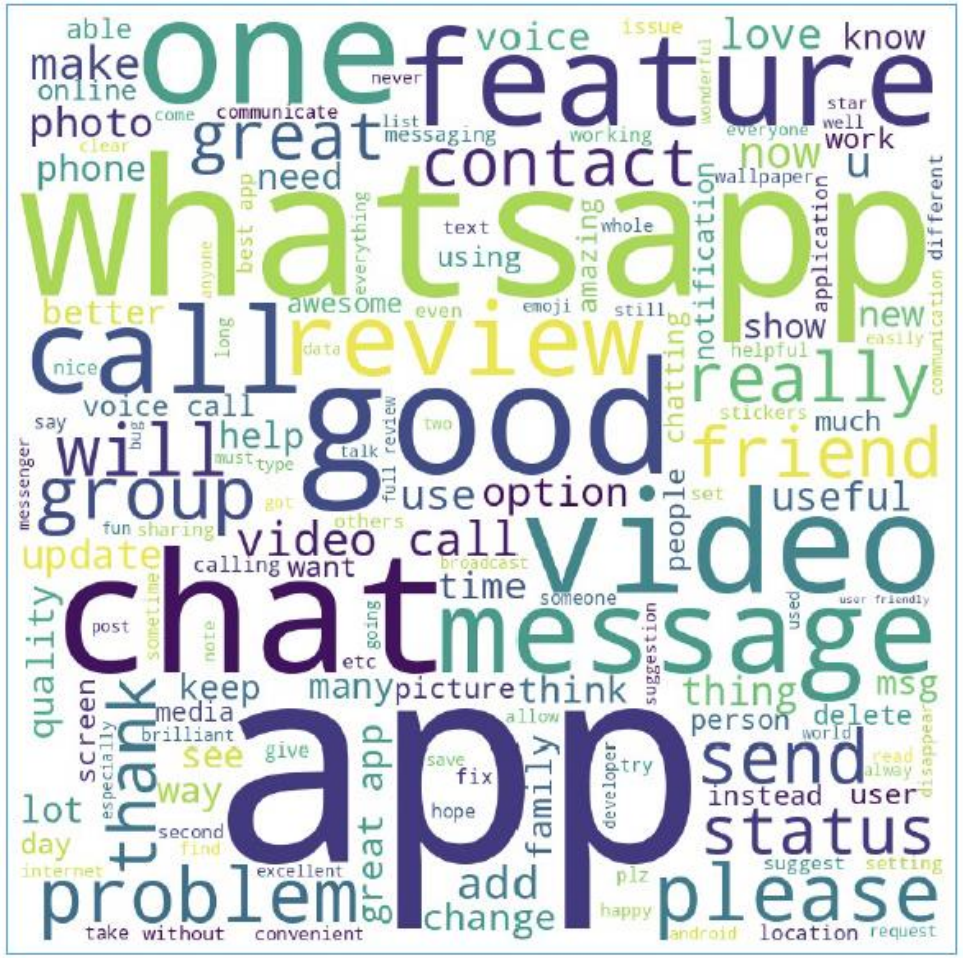


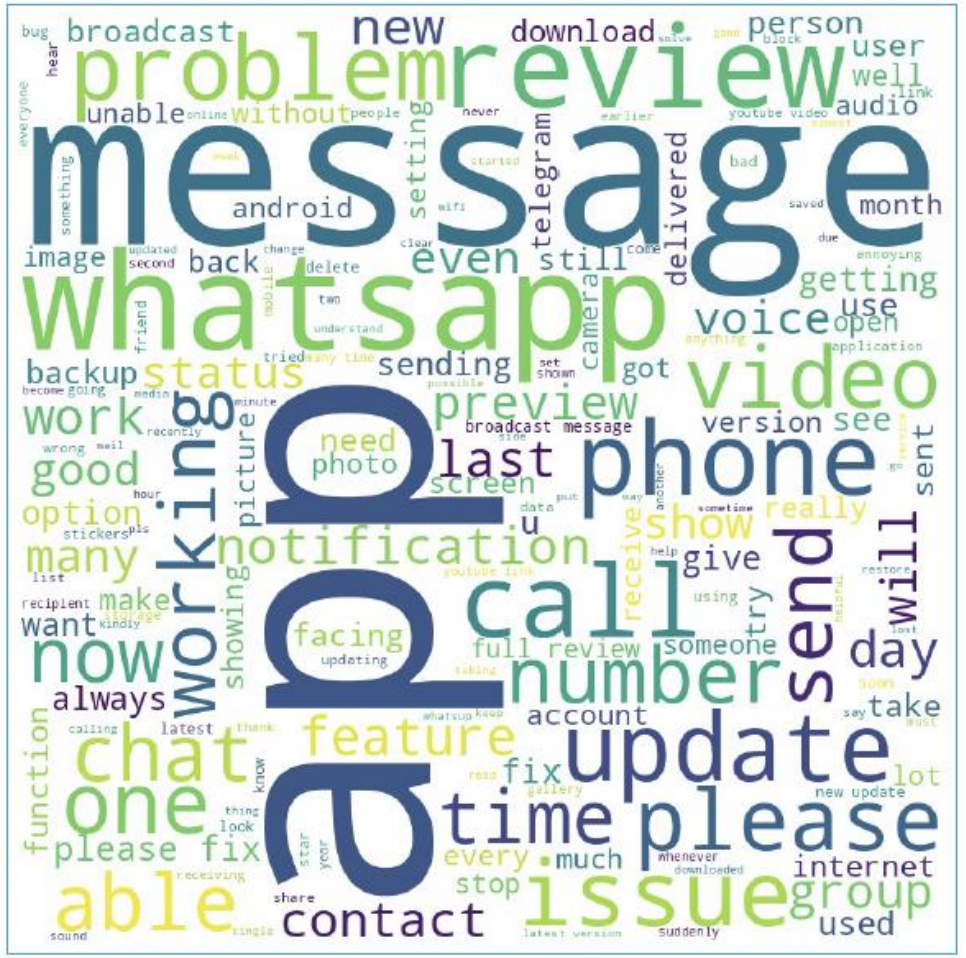


Tips for Right Charts

FPT Education
FPT SCHOOL OF BUSINESS & TECHNOLOGY

Type :6 Dominant Feature Presentation





Practice: Uber Trips

1. Displot
a. Density vs. Day
b. Density vs. Hour
c. Density vs. Weekday

2. Heatmap: Weekday vs. Hour

3. Scatter: Lat vs. Lon (Legend: Uber Trips)