Week 8 Data visualisation¶

Name: Cheong Win Yan

Matric number: 23005189/1

- Data visualisation is a graphical representation of your information or data.
- It is similar to Excel but in Python.

Importance of Data Visualization

- Making complex data understandable
- Enhancing decision making
- Storytelling with Data

In Python, data visualization is mostly performed using 3 libraries:

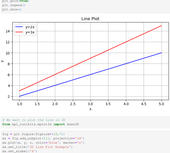
1. matplotlib
2. seaborn
3. plotly

1. Line Plot

In [1]:
```
# Import required libraries
import matplotlib.pyplot as plt
```

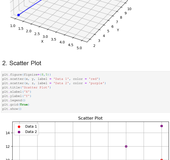
In [2]:
```
# Define some data to plot
x = [1, 2, 3, 4, 5]
y = [1, 3, 5, 7, 9]  # y = 2x
```

In [3]:
```
# Plot a graph
plt.figure(figsize = (6, 4))
plt.plot(x, y, color='blue')
plt.title('Graph of x vs y')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.show()
```



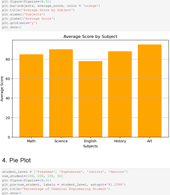
In [4]:
```
# Define another data
y = [1, 2, 3, 4, 5]  # y = x
```

In [5]:
```
# Plot two lines on the same graph
plt.figure(figsize = (6, 4))
plt.plot(x, y, color='blue', label='y=2x')
plt.plot(x, y, color='red', label='y=x')
plt.title('Line Plot')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.legend()
plt.show()
```



In [6]:
```
# Be sure to plot the plot inside
%matplotlib inline
```

In [7]:
```
fig = plt.figure(figsize=(5,5))
ax = fig.add_subplot(111, projection='3d')
ax.plot(x, y, color='blue', marker='o')
ax.set_title('3D Line Plot Example')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
plt.show()
```



2. Scatter Plot

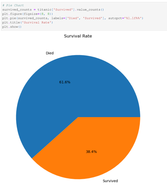
In [8]:
```
plt.figure(figsize=(4,3))
plt.scatter(x, y, label= 'Data 1', color = 'red')
plt.scatter(x, y+1, label= 'Data 2', color = 'blue')
plt.title('Scatter Plot')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.grid(True)
plt.show()
```

3. Bar Plot

In [9]:
```
#assessment_scores_dict
subjects = ['Math', 'Science', 'English', 'History', 'Art']
average_scores = [85, 90, 78, 88, 95]
plt.figure(figsize=(6,4))
plt.bar(subjects, average_scores, color = 'orange')
plt.title('Average Score by Subject')
plt.xlabel('Subjects')
plt.ylabel('Average Score')
plt.grid(axis='y')
plt.show()
```

4. Pie Plot

In [10]:
```
student_level = ['Freshmen', 'Sophomores', 'Juniors', 'Seniors']
num_students=[25, 20, 15, 30]
plt.figure(figsize=(6,4))
plt.pie(num_students, labels = student_level, autopct='%1.1f%%')
plt.title('Percentage of Chemical Engineering Student')
plt.show()
```



Exercise: Data Visualization on Titanic Dataset

Objective: Using the Titanic dataset, explore the data and create various visualizations to gain insights. At the end, please answer the following questions:

1. Did majority of the passengers survived?
2. Is there any relationship between Sex and survival?

Instructions:

1. Load the Dataset:

- Load the titanic.csv into your Jupyter Notebook environment
- Display first few rows using head()

2. Visualize Data

- Bar Plot: To show the number of passengers in each class (Pclass).
- Pie Chart: Show the proportion of passengers who survived vs. did not survive (Survived column).
- Line Plot: Plot the age distribution of passengers. Use Age on the x-axis and the count on the y-axis.
- Scatter Plot: Visualize the relationship between Fare and Age, using different colors for survivors and non-survivors.

3. Summarize the Results

- Answer the 2 questions stated in the Objective

In [11]:
```
import pandas as pd
```

In [12]:
```
df = pd.DataFrame()
```

In [13]:
```
# Load the dataset
titanic = pd.read_csv('titanic.csv')
```

In [14]:
```
# Display the first 5 rows
titanic.head()
```

Out[14]:

	PassengerId	Survived	Pclass	Name	Sex	Age	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	8.0500

In [15]:
```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   Fare         891 non-null    float64
dtypes: float64(2), int64(3), object(2)
memory usage: 48.9+ KB
```

In [16]:
```
titanic.describe()
```

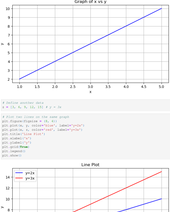
Out[16]:

	PassengerId	Survived	Pclass	Age	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	32.204208
std	257.353842	0.486592	0.836071	14.526497	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	14.454200
75%	668.500000	1.000000	3.000000	35.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	512.329200

The dataset contains:

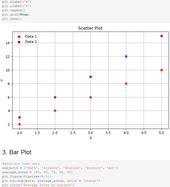
- PassengerID: A number system to indicate a passenger
- Survived: 0 = Died, 1 = Survived
- Pclass: Indicator of the passenger's social status (1 = Upper, 2 = Middle, 3 = Lower)
- Name: Passenger's name
- Sex: Passenger's gender
- Age: Passenger's age
- Fare: Ticket price

In [17]:
```
# Bar plot
plt.figure(figsize=(5,5))
pclass_counts = titanic['Pclass'].value_counts().sort_index()
plt.bar(pclass_counts.index, pclass_counts.values, color='blue')
plt.title('Number of Passengers in Each Class')
plt.xlabel('Passenger Class')
plt.ylabel('Count')
plt.grid(axis='y')
plt.show()
```



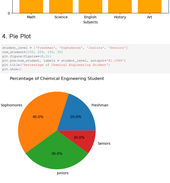
In the Titanic ship, majority are from Class 3 (Lower), followed by Class 1 (Upper), and Class 2 (Middle).

In [18]:
```
# Pie chart
survived_counts = titanic['Survived'].value_counts()
plt.figure(figsize=(5, 5))
plt.pie(survived_counts, labels=['Died', 'Survived'], autopct='%1.1f%%')
plt.title('Survival Rate')
plt.show()
```



Most of the people on the Titanic ship did NOT survived (61.9%). Only 38.4% of the passengers survived.

In [19]:
```
# Line Plot
plt.figure(figsize=(5, 3))
plt.plot(titanic['Age'].value_counts().sort_index(), plt.title('Line')
plt.title('Age Distribution Of Passengers')
plt.xlabel('Age')
plt.ylabel('Count')
plt.grid()
plt.show()
```



Most of the passengers are around 15-35 years old with majority being 29 years old. The youngest being below 1 year old and oldest being 80 years old.

In [20]:
```
import seaborn as sns
```

In [21]:
```
# Scatter plot
plt.figure(figsize=(5, 5))
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=titanic, palette=['red', 'green'])
plt.title('Fare vs Age')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.grid(True)
plt.show()
```

Most people would've thought that fare doesn't really contributes to survival. However, we can see that there is more green (Survived) concentrated at the upper side of y-axis (More expensive fare) while red (Died) are more concentrated at the lower side of y-axis (Cheaper fare).

In [22]:
```
# Plot the relationship between Fare and Survived
plt.figure(figsize=(5, 5))
sns.boxplot(x='Survived', y='Fare', data=titanic)
plt.title('Survival Based on Fare')
plt.xlabel('Survived')
plt.ylabel('Fare')
plt.grid(True)
plt.show()
```


This boxplot just proves it.

Summary of findings:

- Survival Rate: Majority of the passengers did not survive.
- Fare and Survival Relationship: Passengers who paid higher fares (e.g., first-class passengers) had better chances of survival. Likely due to prioritization during the evacuation.