

5.1.1. Stacked Plot

Create a stacked area plot to visualize the temperature variations for three different cities (City A, City B, and City C) across the months of the year. The temperature data is provided for each city in the editor.

Your task is to:

- Create a stacked area plot using the data.
- Label the x-axis as "Month", the y-axis as "Temperature", and provide the title "Temperature Variation" for the plot.
- Display the plot showing the temperature variation for each city throughout the months of the year.

Sample Test Cases

stackedpl...

```
1 import pandas as pd
2
3
4 # Data for Months and Temperature for three cities
5 data = {
6     'Month': ['January', 'February', 'March', 'April', 'May',
7               'June', 'July', 'August', 'September', 'October', 'November',
8               'December'],
9     'City_A_Temperature': [5, 7, 10, 13, 17, 20, 22, 21, 18, 12,
10                           8, 6],
11     'City_B_Temperature': [2, 3, 5, 6, 10, 14, 16, 17, 12, 9, 5,
12                           3],
13     'City_C_Temperature': [3, 4, 6, 8, 9, 12, 15, 14, 10, 7, 4, 2]
14 }
15
16 df = pd.DataFrame(data)
17 plt.stackplot(df["Month"],df["City_A_Temperature"],df["City_B_Temperature"],df["City_C_Temperature"])
18 plt.title("Temperature Variation")
19 plt.xlabel("Month")
20 plt.ylabel("Temperature")
21 plt.show()
```

Terminal Test cases

< Prev

Reset

Submit

Next >

5.2.1. Titanic Dataset00:25

Write a Python program to analyze and visualize data from the Titanic dataset based on the following instructions:

Dataset Information:

The dataset is stored in a CSV file named `titanic.csv` and has been loaded using the `pandas` library. It contains the following columns:

- `Pclass`: Passenger class (1 = First, 2 = Second, 3 = Third).
- `Gender`: Gender of the passenger (male/female).
- `Age`: Age of the passenger.
- `Survived`: Survival status (0 = Did not survive, 1 = Survived).
- `Fare`: Ticket fare paid by the passenger.

Visualization:

To represent these trends, you will create 5 visualizations using `Matplotlib`. The visualizations should be arranged in a 3x2 grid (3 rows and 2 columns).

Sample Test Cases

titanicDat...Submit

1import pandas as pd
2import matplotlib.pyplot as plt
3
4# Load the Titanic dataset from the CSV file
5df = pd.read_csv('titanic.csv')
6
7# Set up the figure for 5 subplots
8fig, axes = plt.subplots(3, 2, figsize=(12, 12))
9
10
11axes[0, 0].bar(df['Pclass'].value_counts().index, df['Pclass'].value_counts(), color='skyblue')
12axes[0, 0].set_title("Passenger Class Distribution")
13axes[0, 0].set_xlabel("Pclass")
14axes[0, 0].set_ylabel("Count")
15
16
17axes[0, 1].pie(df['Gender'].value_counts(), labels=df['Gender'].value_counts().index, autopct='%1.1f%%', colors=['lightblue', 'lightcoral'])
18axes[0, 1].set_title("Gender Distribution")
19
20

TerminalTest cases

5.2.1. Titanic Dataset00:25

Write a Python program to analyze and visualize data from the Titanic dataset based on the following instructions:

Dataset Information:

The dataset is stored in a CSV file named `titanic.csv` and has been loaded using the `pandas` library. It contains the following columns:

- `Pclass`: Passenger class (1 = First, 2 = Second, 3 = Third).
- `Gender`: Gender of the passenger (male/female).
- `Age`: Age of the passenger.
- `Survived`: Survival status (0 = Did not survive, 1 = Survived).
- `Fare`: Ticket fare paid by the passenger.

Visualization:

To represent these trends, you will create 5 visualizations using `Matplotlib`. The visualizations should be arranged in a 3x2 grid (3 rows and 2 columns).

Sample Test Cases+

titanicDat...Submit

19

20 `axes[1, 0].hist(df['Age'].dropna(), bins=8, color='lightgreen',`
`edgecolor='black')`

21 `axes[1, 0].set_title("Age Distribution")`

22 `axes[1, 0].set_xlabel("Age")`

23 `axes[1, 0].set_ylabel("Frequency")`

24

25 `axes[1,`
`1].bar(df['Survived'].value_counts().index, df['Survived'].value_co`
`unts(), color=['lightblue', 'lightcoral'])`

26 `axes[1, 1].set_title("Survival Count")`

27 `axes[1, 1].set_xlabel("Survived (0 = No, 1 = Yes)")`

28 `axes[1, 1].set_ylabel("Count")`

29

30 `axes[2, 0].scatter(df['Age'], df['Fare'],`
`color='orange', edgecolors='black')`

31 `axes[2, 0].set_title("Fare vs Age")`

32 `axes[2, 0].set_xlabel("Age")`

33 `axes[2, 0].set_ylabel("Fare")`

34

35 `plt.tight_layout()`

36 `plt.show()`

TerminalTest cases

5.2.2. Histogram of passenger information of Titanic01:33

Write a Python code to plot a histogram for the distribution of the 'Age' column from the Titanic dataset. The histogram should display the frequency of different age ranges with the following specifications:

1. Use **30 bins** for the histogram.
2. Set the **edge color** of the bars to **black (k)**.
3. Label the x-axis as **'Age'** and the y-axis as **'Frequency'**.
4. Add the title **"Age Distribution"** to the histogram.

The Titanic dataset contains columns as shown below,

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	ParCh	Ticket	Fare	Cabin	Embarked

Sample Test Cases

Histograma...Submit

Explorer

1import pandas as pd
2import matplotlib.pyplot as plt
3
4# Load the Titanic dataset
5data = pd.read_csv('Titanic-Dataset.csv')
6
7# Data Cleaning
8data['Age'].fillna(data['Age'].median(), inplace=True)
9data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10data.drop('Cabin', axis=1, inplace=True)
11
12# Convert categorical features to numeric
13data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16# Write your code here for Histogram
17plt.hist(data["Age"],bins=30,edgecolor="k")
18plt.xlabel("Age")
19plt.ylabel("Frequency")
20plt.title('Age Distribution')
21plt.show()

Debugger

TerminalTest cases

< PrevResetSubmitNext >

5.2.3. Bar plot of survival rate of passengers 08:34

Write a Python code to plot a bar chart that shows the count of passengers who survived and did not survive in the Titanic dataset. The chart should display the following specifications:

1. Use the **'Survived'** column to show the count of survivors (0 = Did not survive, 1 = Survived).
2. Set the chart type to **'bar'**.
3. Add the title **"Survival Count"** to the chart.
4. Label the x-axis as **'Survived'** and the y-axis as **'Count'**.

The Titanic dataset contains columns as shown below,

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	ParCh	Ticket	Fare	Cabin	Embarked

Sample Test Cases

BarPlotOf... Submit

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Bar Plot for Survival Rate
17 survival_counts=data["Survived"].value_counts()
18 survival_counts.plot(kind="bar")
19 plt.title("Survival Count")
20 plt.xlabel("Survived")
21 plt.ylabel("Count")
22 plt.show()
```

Terminal

Test cases

Plot for Survival by Gender

Write Python code to plot a stacked bar chart that shows the count of passengers who survived and did not survive, grouped by gender, in the Titanic dataset. The chart should follow the following specifications:

- Group the data by the **'Sex'** column, then use the **value_counts()** function to count the occurrences of survivors (0 = Did not survive, 1 = Survived) for each gender.
- Use a **stacked bar chart** to display the survival counts.
- Set the title **"Survival by Gender"** to the chart.
- Set the x-axis as **'Gender'** and the y-axis as **'Count'**.
- The legend should indicate **'Not Survived'** and **'Survived'**.

The dataset contains columns as shown below,

Survived	Pclass	Name	Sex	Age	SibSp	ParCh	Ticket	Fare	Cabin	Embarked

Test Cases

```
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Bar Plot for Survival by Gender
17
18 survival_by_gender=data.groupby("Sex")
19 ["Survived"].value_counts().unstack().fillna(0)
20 survival_by_gender.columns=["Not Survived", "Survived"]
21 survival_by_gender.index=["0", "1"]
22 survival_by_gender.plot(kind="bar", stacked=True)
23 plt.title("Survival by Gender")
24 plt.xlabel("Gender")
25 plt.ylabel("Count")
26 plt.legend(title=None)
27 plt.show()
```

5.2.5. Bar Plot for Survival by Pclass

Write a Python code to plot a stacked bar chart that shows the count of passengers who survived and did not survive, grouped by passenger class (**Pclass**), in the Titanic dataset. The chart should display the following specifications:

1. Group the data by the **Pclass** column and count the number of survivors (0 = Did not survive, 1 = Survived) for each class using **value_counts()**.
2. Use a **stacked bar chart** to display the survival counts.
3. Add the title "**Survival by Pclass**" to the chart.
4. Label the x-axis as '**Pclass**' and the y-axis as '**Count**'.
5. The legend should indicate '**Not Survived**' and '**Survived**'.

The Titanic dataset contains columns as shown below,

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	ParCh	Ticket	Fare	Cabin	Embarked
-------------	----------	--------	------	-----	-----	-------	-------	--------	------	-------	----------

Sample Test Cases

+

BarPlotOf...

```
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Bar Plot for Survival by Pclass
17 survival_by_class=data.groupby("Pclass")
18 ["Survived"].value_counts().unstack().fillna(0)
19 survival_by_class.columns=["Not Survived", "Survived"]
20 survival_by_class.plot(kind="bar",stacked=True)
21 plt.title("Survival by Pclass")
22 plt.xlabel("Pclass")
23 plt.ylabel("Count")
24 plt.legend(title=None)
25 plt.show()
26
```

Terminal Test cases

< Prev Reset Submit Next >

5.2.10. Scatter Plot for Age vs. Fare02:09

Write a Python code to plot a scatter plot showing the relationship between the 'Age' and 'Fare' columns in the Titanic dataset. The scatter plot should display the following specifications:

1. Use the **Age** column for the x-axis and the **Fare** column for the y-axis.
2. Set the title of the plot to "**Age vs. Fare**".
3. Label the x-axis as '**Age**' and the y-axis as '**Fare**'.

The Titanic dataset contains columns as shown below,

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	ParCh	Ticket	Fare	Cabin	Embarked

Sample Data:

Sample Test Cases+

AgeFareS...Submit

```
5 data = pd.read_csv('Titanic-Dataset.csv')
6
7 # Data Cleaning
8 data['Age'].fillna(data['Age'].median(), inplace=True)
9 data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
15
16 # Write your code here for Box Plot for Fare by Pclass
17 plt.figure(figsize=(6.4,4.8))
18 data.scatter(data["Age"],data["Fare"])
19 plt.title("Age vs. Fare")
20 plt.xlabel("Age")
21 plt.ylabel("Fare")
22 plt.show()
23
24
25
26
```

TerminalTest cases

< PrevResetSubmitNext >

5.2.10. Scatter Plot for Age vs. Fare

Write a Python code to plot a scatter plot showing the relationship between the 'Age' and 'Fare' columns in the Titanic dataset. The scatter plot should display the following specifications:

1. Use the **Age** column for the x-axis and the **Fare** column for the y-axis.

2. Set the title of the plot to **"Age vs. Fare"**.

3. Label the x-axis as **'Age'** and the y-axis as **'Fare'**.

The Titanic dataset contains columns as shown below,

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	ParCh	Ticket	Fare	Cabin	Embarked

Sample Data:

Sample Test Cases

AgeFareS...

data = pd.read_csv('Titanic-Dataset.csv')

Data Cleaning

data['Age'].fillna(data['Age'].median(), inplace=True)

data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

data.drop('Cabin', axis=1, inplace=True)

Convert categorical features to numeric

data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})

data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

Write your code here for Box Plot for Fare by Pclass

plt.figure(figsize=(6.4,4.8))

plt.scatter(data['Age'],data['Fare'])

plt.title('Age vs. Fare')

plt.xlabel('Age')

plt.ylabel('Fare')

plt.show()

Terminal

Test cases

5.2.11. Scatter Plot for Age vs. Fare by Survived

03:17

Write a Python code to plot a scatter plot showing the relationship between the 'Age' and 'Fare' columns in the Titanic dataset, with points color-coded by survival status. The scatter plot should display the following specifications:

1. Use the **Age** column for the x-axis and the **Fare** column for the y-axis.
2. Color the points based on the **Survived** column: **Red** for passengers who did not survive (**Survived = 0**). **Blue** for passengers who survived (**Survived = 1**).
3. Set the title of the plot to **"Age vs. Fare by Survival"**.
4. Label the x-axis as **'Age'** and the y-axis as **'Fare'**.

The Titanic dataset contains columns as shown below,

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	ParCh	Ticket	Fare	Cabin	Embarked

Sample Test Cases

+

AgeFareS...

Submit

1import pandas as pd

2import matplotlib.pyplot as plt

3

4# Load the Titanic dataset

5data = pd.read_csv('Titanic-Dataset.csv')

6

7# Data Cleaning

8data['Age'].fillna(data['Age'].median(), inplace=True)

9data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

10data.drop('Cabin', axis=1, inplace=True)

11

12# Convert categorical features to numeric

13data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})

14data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

15

16colors = data['Survived'].map({0: 'red', 1: 'blue'})

17plt.scatter(data['Age'], data['Fare'], c=colors)

18plt.title('Age vs. Fare by Survival')

19plt.xlabel('Age')

20plt.ylabel('Fare')

21plt.show()

22

Terminal

Test cases

Debugger