Data Analysis with Python on csv file Data:-

admissions.csv:-

StudentID,Name,Source

1,John Doe,Social Media

2,Jane Smith,Google Search

3,Ali Khan,Banner Enquiry

4,Amy Adams,Reference by Friend

5,Tom Lee,Social Media

6,Linda Ray,Google Search

7,Harry Potter,Reference by Friend

8,Clark Kent,Banner Enquiry

9,Lois Lane,Social Media

10,Bruce Wayne,Google Search

Note:-

admission-data-analysis.py :-

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the CSV file
file_path = 'd:/python demo/admissions.csv'  # Make sure the file is in the same
directory
df = pd.read_csv(file_path)

# Show the first few records (optional)
print("Sample data:")
print(df.head())

# Count the number of admissions per source
source_counts = df['Source'].value_counts()

print("\nNumber of admission forms by source:")
```

```
print(source_counts)

# Plotting the data (optional)
plt.figure(figsize=(8, 5))
source_counts.plot(kind='bar', color='skyblue', edgecolor='black')
plt.title('Number of Admission Forms by Source')
plt.xlabel('Source')
plt.ylabel('Number of Forms')
plt.xticks(rotation=45)
plt.tight_layout()
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

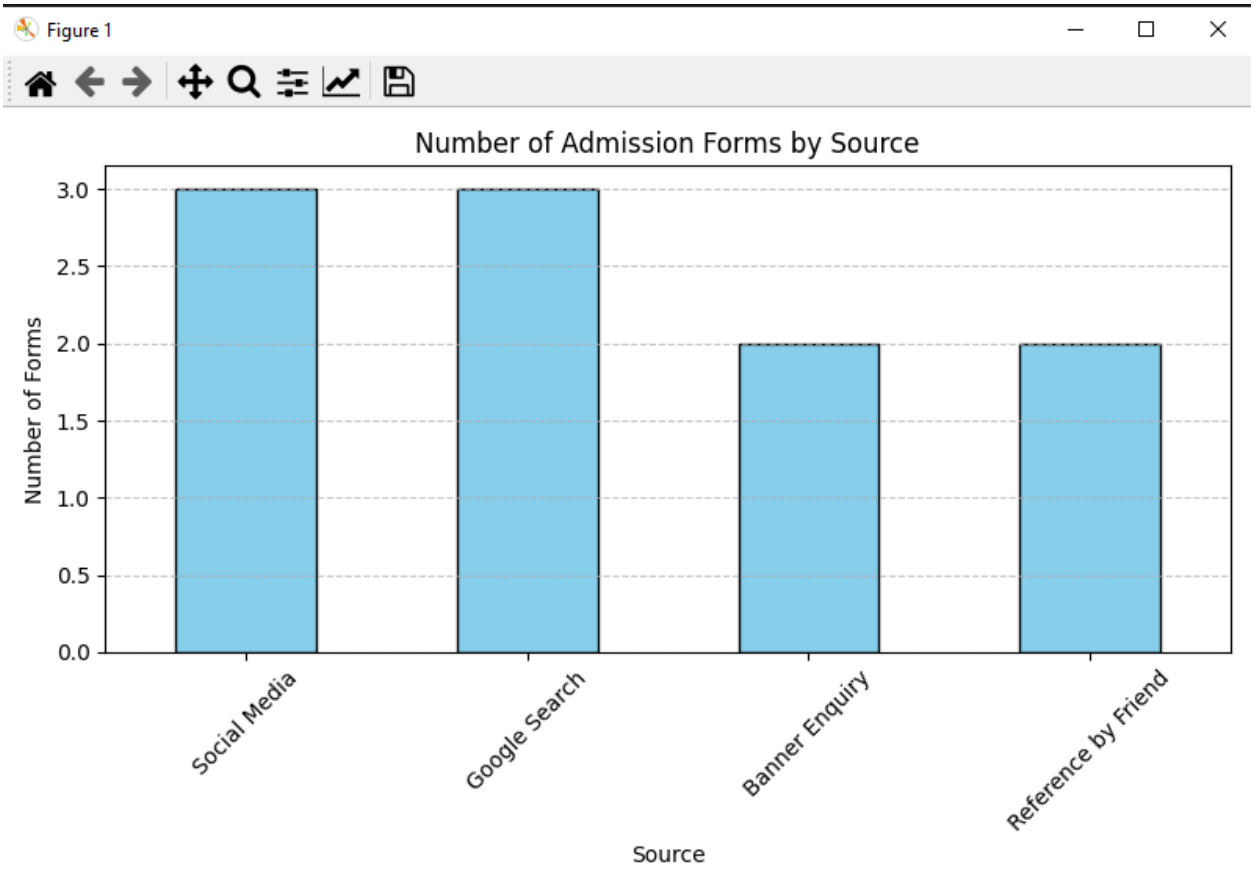
output :-

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS D:\python demo> & "C:/Users/Big Data/AppData/Local/Programs/Python/Python313/python.exe
Sample data:
   StudentID        Name              Source
0          1    John Doe        Social Media
1          2  Jane Smith       Google Search
2          3    Ali Khan      Banner Enquiry
3          4   Amy Adams  Reference by Friend
4          5     Tom Lee        Social Media

Number of admission forms by source:
Source
Social Media           3
Google Search          3
Banner Enquiry         2
Reference by Friend    2
Name: count, dtype: int64
```

Output of graph:-



Note:-

Here's a **line-by-line explanation** of your code:

1. Import Required Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
```

- `pandas` (aliased as `pd`) is a powerful library used for data manipulation and analysis.
- `matplotlib.pyplot` (aliased as `plt`) is a plotting library used to create static, animated, and interactive visualizations.

□ 2. Load the CSV File

```
file_path = 'd:/python demo/admissions.csv'
df = pd.read_csv(file_path)
```

- `file_path`: Stores the path to your CSV file containing admission data.
- `pd.read_csv(file_path)`: Reads the CSV file into a DataFrame called `df`, which acts like an Excel table in Python.

□ 3. Display Sample Data (Optional)

```
print("Sample data:")
print(df.head())
```

- `df.head()` displays the **first 5 rows** of the DataFrame to give you a quick look at the data.
- Useful for verifying that the file loaded correctly.

□ 4. Count Admissions by Source

```
source_counts = df['Source'].value_counts()
```

- `df['Source']`: Accesses the column named `"Source"` in the dataset.
- `.value_counts()`: Counts how many times each unique source (e.g., "Website", "Referral", etc.) appears.

□ The result: a **summary table** showing how many admission forms came from each source.

---

5. Print the Counts

```
print("\nNumber of admission forms by source:")
print(source_counts)
```

- Displays the count results in the console for review.

---

6. Prepare for Plotting

```
plt.figure(figsize=(8, 5))
```

- Creates a new figure (canvas) for the chart.
- `figsize=(8, 5)`: Sets the size of the chart to 8 inches wide and 5 inches tall.

---

7. Create a Bar Chart

```
source_counts.plot(kind='bar', color='skyblue', edgecolor='black')
```

- Plots a **bar chart** from the `source_counts` data.
- `kind='bar'`: Specifies that it should be a vertical bar chart.
- `color='skyblue'`: Fills the bars with a light blue color.
- `edgecolor='black'`: Adds a black outline around each bar.

---

8. Add Title and Labels

```
plt.title('Number of Admission Forms by Source')
plt.xlabel('Source')
plt.ylabel('Number of Forms')
```

- `plt.title()`: Adds a title at the top of the chart.
- `plt.xlabel()`: Labels the x-axis as `"Source"`.
- `plt.ylabel()`: Labels the y-axis as `"Number of Forms"`.

---

9. Rotate X-Axis Labels

```
plt.xticks(rotation=45)
```

- Rotates the labels on the x-axis by 45 degrees.
- Helps prevent text from overlapping if labels are long.

---

10. Improve Layout and Add Grid

```
plt.tight_layout()
plt.grid(axis='y', linestyle='--', alpha=0.7)
```

- `plt.tight_layout()`: Adjusts layout so that labels/titles fit well and don't get cut off.
- `plt.grid()`: Adds horizontal grid lines for better readability.
  - `axis='y'`: Adds grid lines only along the y-axis.
  - `linestyle='--'`: Makes the grid lines dashed.
  - `alpha=0.7`: Makes grid lines semi-transparent.

---

11. Show the Plot

```
plt.show()
```

- Displays the final bar chart in a new window or inline (if you're using Jupyter Notebook or similar).