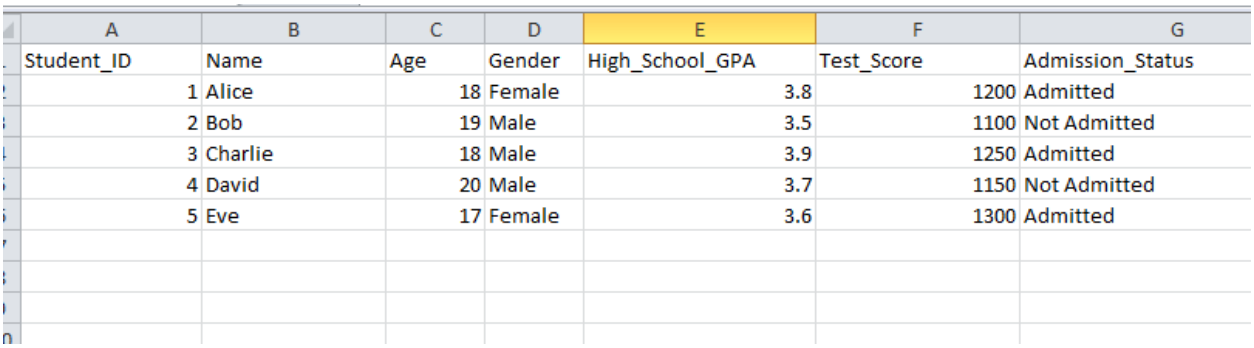
Create one csv file and save it as student_admissions.csv



	A	B	C	D	E	F	G
	Student_ID	Name	Age	Gender	High_School_GPA	Test_Score	Admission_Status
2	1	Alice	18	Female	3.8	1200	Admitted
3	2	Bob	19	Male	3.5	1100	Not Admitted
4	3	Charlie	18	Male	3.9	1250	Admitted
5	4	David	20	Male	3.7	1150	Not Admitted
6	5	Eve	17	Female	3.6	1300	Admitted
7							
8							
9							
10							

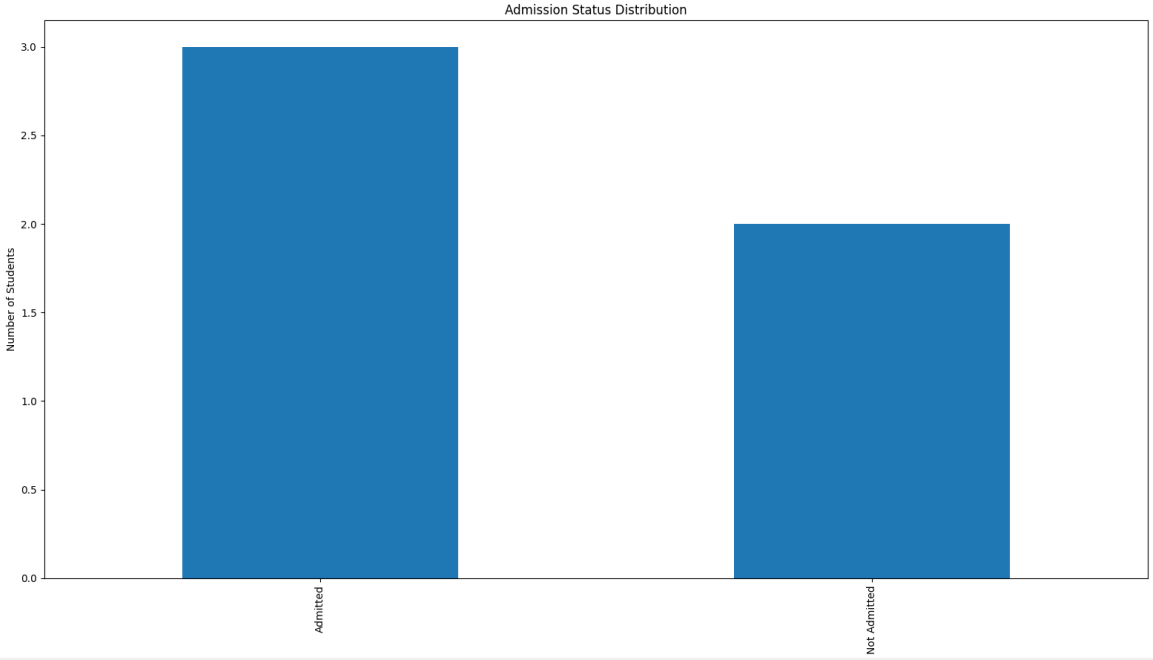
Create file readcsv.py:-

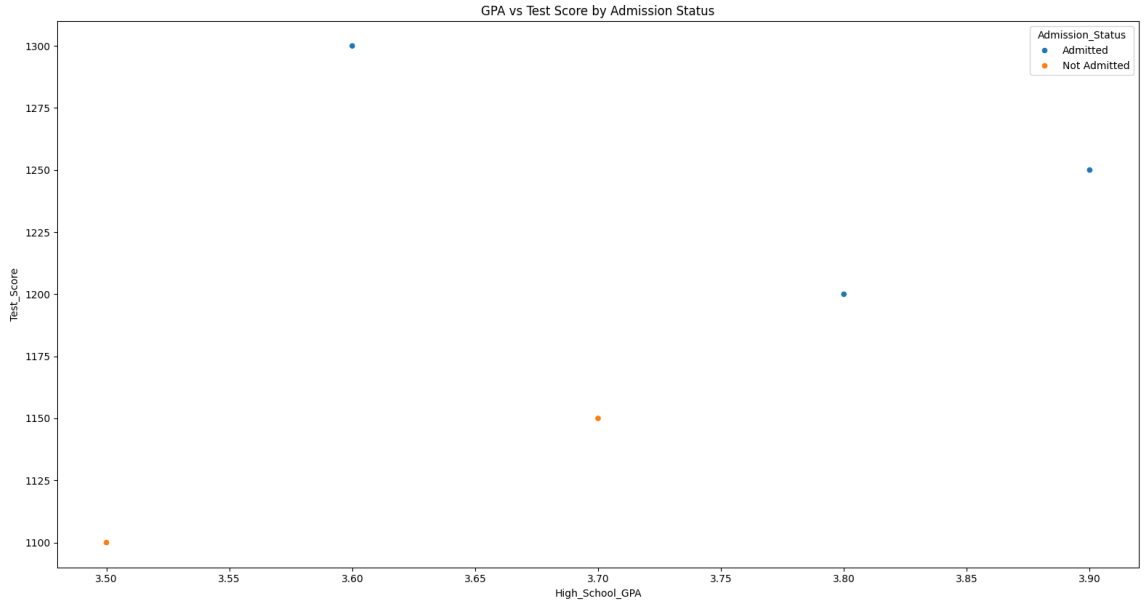
```
import pandas as pd

# Load the dataset (assuming a CSV file)
data = pd.read_csv("student_admissions.csv")

# Display the first few rows of the dataset
print(data.head())

# Summary statistics of numerical columns
print(data.describe())

# Distribution of admission status (e.g., how many were admitted vs. not
admitted)
admission_count = data['Admission_Status'].value_counts()
print(admission_count)

single_column = data['Name']
print(single_column)

filtered_df_4 = data[(data['Age'] >= 18) & (data['Age'] <= 20)]
print("filter data are:")
print(filtered_df_4)
# Plotting the distribution of Admission Status
import matplotlib.pyplot as plt
admission_count.plot(kind='bar', title="Admission Status Distribution")
plt.ylabel('Number of Students')
plt.show()

# Correlation between GPA and Test Score
import seaborn as sns
sns.scatterplot(data=data, x='High_School_GPA', y='Test_Score',
hue='Admission_Status')
plt.title('GPA vs Test Score by Admission Status')
plt.show()
```

OutPut:-

```
PS D:\python demo> & "C:/Users/Big Data/AppData/Local/Programs/Python/Python311/python.exe" "d:/
   Student_ID     Name  Age  Gender  High_School_GPA  Test_Score Admission_Status
0           1    Alice   18  Female              3.8        1200         Admitted
1           2      Bob   19    Male              3.5        1100     Not Admitted
2           3  Charlie   18    Male              3.9        1250         Admitted
3           4    David   20    Male              3.7        1150     Not Admitted
4           5      Eve   17  Female              3.6        1300         Admitted
       Student_ID        Age  High_School_GPA   Test_Score
count    5.000000   5.000000         5.000000     5.000000
mean     3.000000  18.400000         3.700000  1200.000000
std      1.581139   1.140175         0.158114    79.056942
min      1.000000  17.000000         3.500000  1100.000000
25%      2.000000  18.000000         3.600000  1150.000000
50%      3.000000  18.000000         3.700000  1200.000000
75%      4.000000  19.000000         3.800000  1250.000000
max      5.000000  20.000000         3.900000  1300.000000
Admission_Status
Admitted        3
Not Admitted    2
Name: count, dtype: int64
0      Alice
1        Bob
2    Charlie
3      David
4        Eve
```

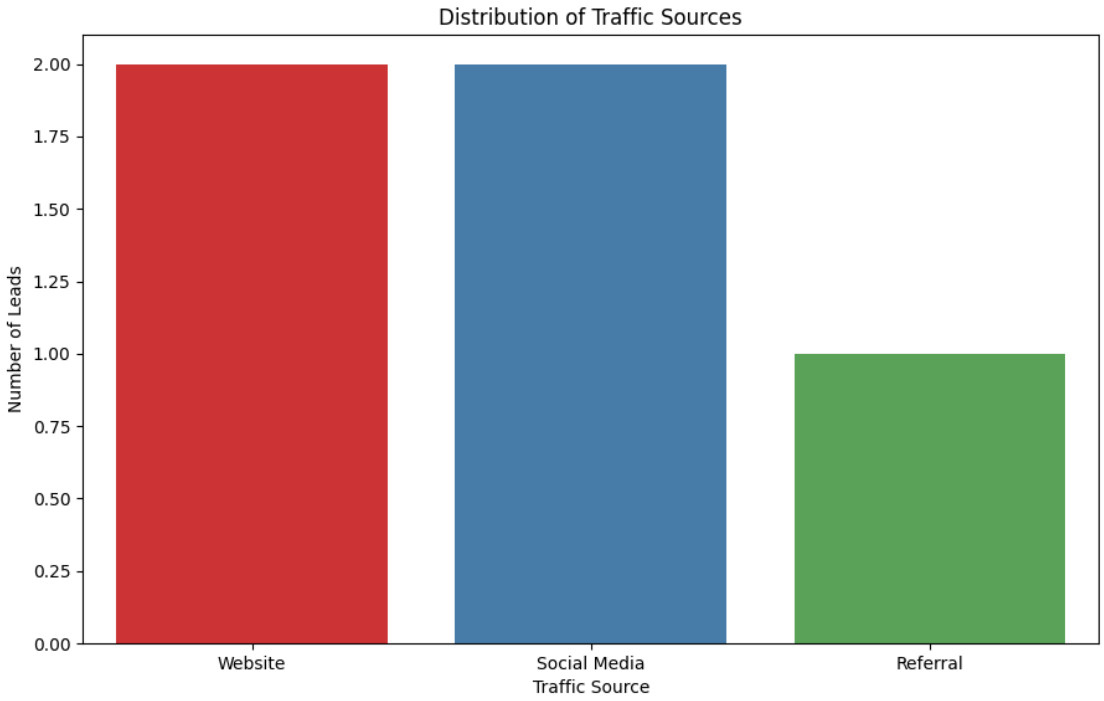
```
Name: Name, dtype: object
filter data are:
   Student_ID     Name  Age  Gender  High_School_GPA  Test_Score Admission_Status
0           1    Alice   18  Female              3.8        1200         Admitted
1           2      Bob   19    Male              3.5        1100     Not Admitted
2           3  Charlie   18    Male              3.9        1250         Admitted
3           4    David   20    Male              3.7        1150     Not Admitted
```


Admission Status Distribution
Number of Students
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Admitted
Not Admitted


GPA vs Test Score by Admission Status
Admission_Status
Admitted
Not Admitted
Test_Score
1300
1275
1250
1225
1200
1175
1150
1125
1100
3.50
3.55
3.60
3.65
3.70
3.75
3.80
3.85
3.90
High_School_GPA

lead.csv file:-

Lead_ID	Name	Lead Enquiry	Under Follow-up	Converted Admission	Course Name	Traffic Source
1	Alice	Yes	Yes	No	Data Science	Website
2	Bob	Yes	No	Yes	Web Development	Social Media
3	Charlie	Yes	Yes	Yes	Cyber Security	Referral
4	David	Yes	No	No	Machine Learning	Website
5	Eve	Yes	Yes	Yes	Data Science	Social Media



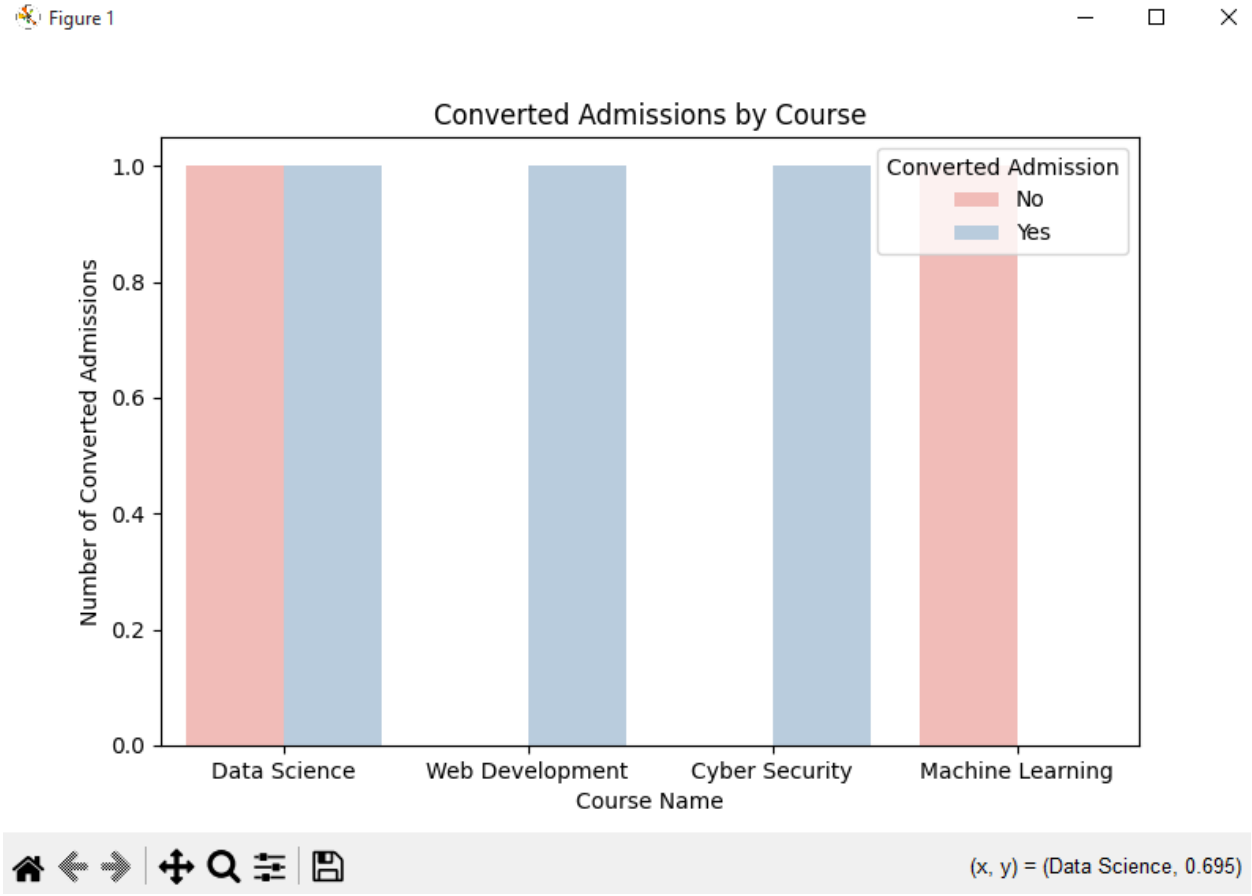


Figure 1
Converted Admissions by Course
Converted Admission
No
Yes
Number of Converted Admissions
1.0
0.8
0.6
0.4
0.2
0.0
Data Science
Web Development
Cyber Security
Machine Learning
Course Name
(x, y) = (Data Science, 0.695)

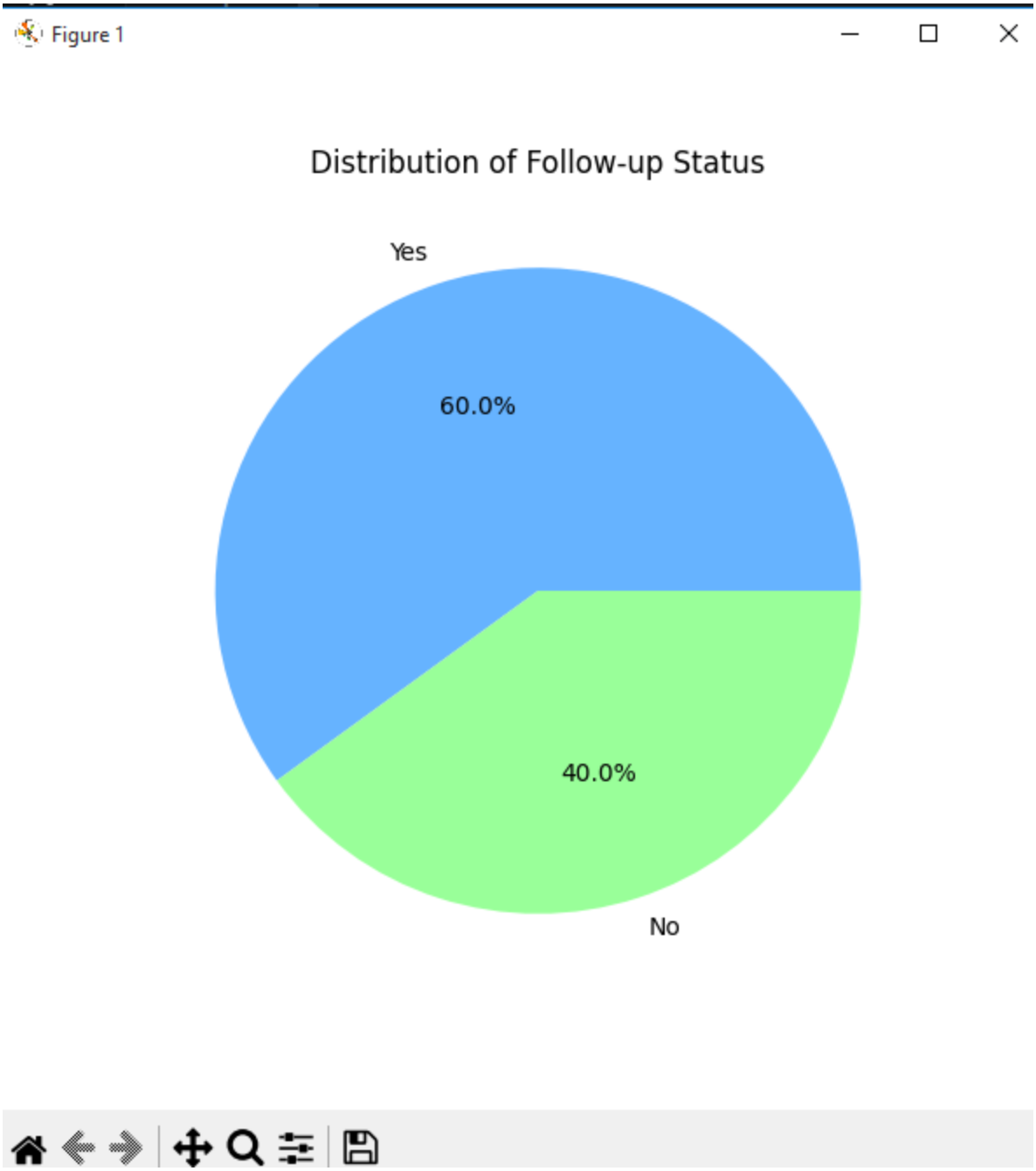


Figure 1
Distribution of Follow-up Status
Yes
60.0%
40.0%
No

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

# Sample data similar to the CSV file format
data = pd.read_csv('lead.csv')

# Create a DataFrame from the data
df = pd.DataFrame(data)

# Display the full DataFrame
print("Full DataFrame:")
print(df)

# Example 1: Filter leads that are 'Under Follow-up' and show their 'Course Name'
and 'Traffic Source'
under_follow_up = df[df['Under Follow-up'] == 'Yes'][['Name', 'Course Name',
'Traffic Source']]
print("\nLeads Under Follow-up (with Course and Traffic Source):")
print(under_follow_up)

# Example 2: Filter leads that have been 'Converted Admission' and show their
'Course Name' and 'Traffic Source'
converted_admission = df[df['Converted Admission'] == 'Yes'][['Name', 'Course
Name', 'Traffic Source']]
print("\nConverted Admissions (with Course and Traffic Source):")
print(converted_admission)

# Example 3: Filter leads that came from 'Social Media' and show their 'Name' and
'Course Name'
social_media_leads = df[df['Traffic Source'] == 'Social Media'][['Name', 'Course
Name']]
print("\nLeads from Social Media (with Course Name):")
print(social_media_leads)

# Example 4: Filter leads who are under follow-up and have an interest in 'Data
Science'
data_science_follow_up = df[(df['Under Follow-up'] == 'Yes') & (df['Course Name']
== 'Data Science')]
print("\nData Science Leads Under Follow-up:")
print(data_science_follow_up)

# Plotting the distribution of traffic sources
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='Traffic Source', palette='Set1')
plt.title('Distribution of Traffic Sources')
plt.xlabel('Traffic Source')
```

```
plt.ylabel('Number of Leads')
plt.show()

# Plotting the number of converted admissions by course
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='Course Name', hue='Converted Admission',
palette='Pastel1')
plt.title('Converted Admissions by Course')
plt.xlabel('Course Name')
plt.ylabel('Number of Converted Admissions')
plt.show()

# Pie chart showing the distribution of follow-up status
follow_up_status = df['Under Follow-up'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(follow_up_status, labels=follow_up_status.index, autopct='%1.1f%%',
colors=['#66b3ff', '#99ff99'])
plt.title('Distribution of Follow-up Status')
plt.show()

# Pie chart showing the distribution of converted admissions
conversion_status = df['Converted Admission'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(conversion_status, labels=conversion_status.index, autopct='%1.1f%%',
colors=['#ff9999', '#66ff66'])
plt.title('Distribution of Converted Admissions')
plt.show()
```

output:-

```
PS D:\python demo> & "C:/Users/Big Data/AppData/Local/Programs/Python/Python311/python.exe" "d:/python de

Full DataFrame:
   Lead_ID     Name Lead Enquiry Under Follow-up Converted Admission       Course Name Traffic Source
0        1    Alice          Yes             Yes                  No      Data Science        Website
1        2      Bob          Yes              No                 Yes   Web Development   Social Media
2        3  Charlie          Yes             Yes                 Yes    Cyber Security       Referral
3        4    David          Yes              No                  No  Machine Learning        Website
4        5      Eve          Yes             Yes                 Yes      Data Science   Social Media

Leads Under Follow-up (with Course and Traffic Source):
      Name     Course Name Traffic Source
0    Alice    Data Science        Website
2  Charlie  Cyber Security       Referral
4      Eve    Data Science   Social Media

Converted Admissions (with Course and Traffic Source):
      Name      Course Name Traffic Source
1      Bob  Web Development   Social Media
2  Charlie   Cyber Security       Referral
4      Eve     Data Science   Social Media
```