```
from google.colab import files
uploaded = files.upload()
```

Choose Files Customer P…haviors.csv

Customer Purchasing Behaviors.csv(text/csv) - 7268 bytes, last modified: 4/2/2026 - 100% done

```
Saving Customer Purchasing Behaviors.csv to Customer Purchasing Behaviors.csv
```

```
#1
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#2
df = pd.read_csv("Customer Purchasing Behaviors.csv")
df
```

	user_id	age	annual_income	purchase_amount	loyalty_score	region	purchase_frequency
0	1	25	45000	200	4.5	North	12
1	2	34	55000	350	7.0	South	18
2	3	45	65000	500	8.0	West	22
3	4	22	30000	150	3.0	East	10
4	5	29	47000	220	4.8	North	13
...	...	...	...	...	...	...	...
233	234	40	60000	450	7.2	West	20
234	235	38	59000	430	6.9	North	20
235	236	54	74000	630	9.4	South	27
236	237	32	52000	360	5.8	West	18
237	238	31	51000	340	5.6	North	17

238 rows × 7 columns

Next steps: Generate code with df | New interactive sheet

```
#3
print(df.shape)
```

```
(238, 7)
```

```
#4
print(df.columns)
```

```
Index(['user_id', 'age', 'annual_income', 'purchase_amount', 'loyalty_score',
       'region', 'purchase_frequency'],
      dtype='object')
```

```
#5
print(df.head(5))
```

```
   user_id  age  annual_income  purchase_amount  loyalty_score region  \
0        1   25          45000              200            4.5  North
1        2   34          55000              350            7.0  South
2        3   45          65000              500            8.0   West
3        4   22          30000              150            3.0   East
4        5   29          47000              220            4.8  North

   purchase_frequency
0                  12
1                  18
2                  22
3                  10
4                  13
```

```
#6
print(df.dtypes)
```

```
user_id                int64
age                    int64
annual_income          int64
purchase_amount        int64
loyalty_score        float64
```

```
region                object
purchase_frequency     int64
dtype: object
```

```
#7
print(df.isnull().sum())
```

```
user_id              0
age                  0
annual_income        0
purchase_amount      0
loyalty_score        0
region               0
purchase_frequency   0
dtype: int64
```

```
#8
num_cols = df.select_dtypes(include=np.number).columns

for col in num_cols:
    df[col].fillna(df[col].mean(), inplace=True)
df
```

```
/tmp/ipython-input-2753514288.py:5: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]

  df[col].fillna(df[col].mean(), inplace=True)
```

	user_id	age	annual_income	purchase_amount	loyalty_score	region	purchase_frequency
0	1	25	45000	200	4.5	North	12
1	2	34	55000	350	7.0	South	18
2	3	45	65000	500	8.0	West	22
3	4	22	30000	150	3.0	East	10
4	5	29	47000	220	4.8	North	13
...	...	...	...	...	...	...	...
233	234	40	60000	450	7.2	West	20
234	235	38	59000	430	6.9	North	20
235	236	54	74000	630	9.4	South	27
236	237	32	52000	360	5.8	West	18
237	238	31	51000	340	5.6	North	17

238 rows × 7 columns

Next steps: Generate code with df | New interactive sheet

```
#9
cat_cols = df.select_dtypes(include='object').columns

for col in cat_cols:
    df[col].fillna(df[col].mode()[0], inplace=True)
df
```

```
/tmp/ipython-input-3043855207.py:5: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  df[col].fillna(df[col].mode()[0], inplace=True)
```

	user_id	age	annual_income	purchase_amount	loyalty_score	region	purchase_frequency
0	1	25	45000	200	4.5	North	12
1	2	34	55000	350	7.0	South	18
2	3	45	65000	500	8.0	West	22
3	4	22	30000	150	3.0	East	10
233	234	40	60000	450	7.2	West	20
234	235	38	59000	430	6.9	North	20
235	236	54	74000	630	9.4	South	27
236	237	32	52000	360	5.8	West	18
237	238	51	51000	340	5.6	North	17

238 rows × 7 columns

Next ith df New interactive sheet

```
#10
df.isnull().sum()
```

	0
user_id	0
age	0
annual_income	0
purchase_amount	0
loyalty_score	0
region	0
purchase_frequency	0

dtype: int64

```
#11
df[num_cols].mean()
```

	0
user_id	119.500000
age	38.676471
annual_income	57407.563025
purchase_amount	425.630252
loyalty_score	6.794118
purchase_frequency	19.798319

dtype: float64

```
#12
df[num_cols].median()
```

	0
user_id	119.5
age	39.0
annual_income	59000.0
purchase_amount	440.0
loyalty_score	7.0
purchase_frequency	20.0

dtype: float64

```
#13
df[num_cols].std()
```

	0
user_id	68.848868
age	9.351118
annual_income	11403.875717
purchase_amount	140.052062
loyalty_score	1.899047
purchase_frequency	4.562884

dtype: float64

```
#14
df[num_cols].min()
df[num_cols].max()
```

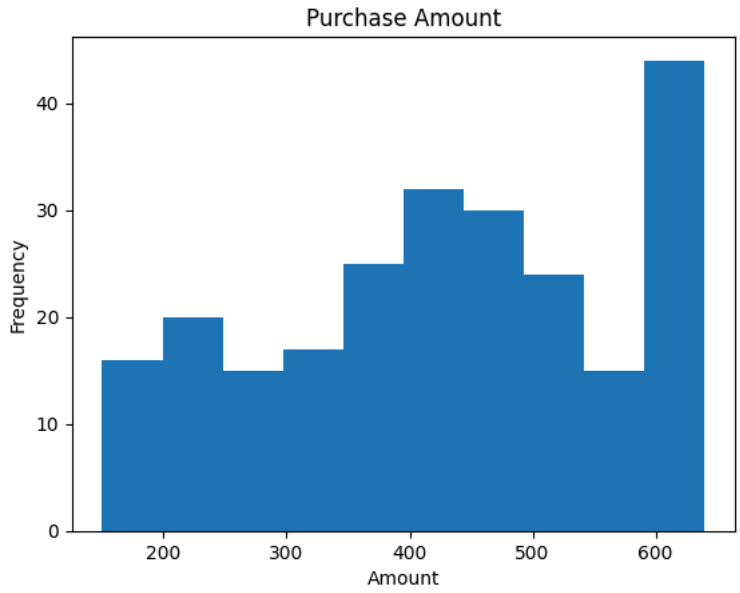
	0
user_id	238.0
age	55.0
annual_income	75000.0
purchase_amount	640.0
loyalty_score	9.5
purchase_frequency	28.0

dtype: float64

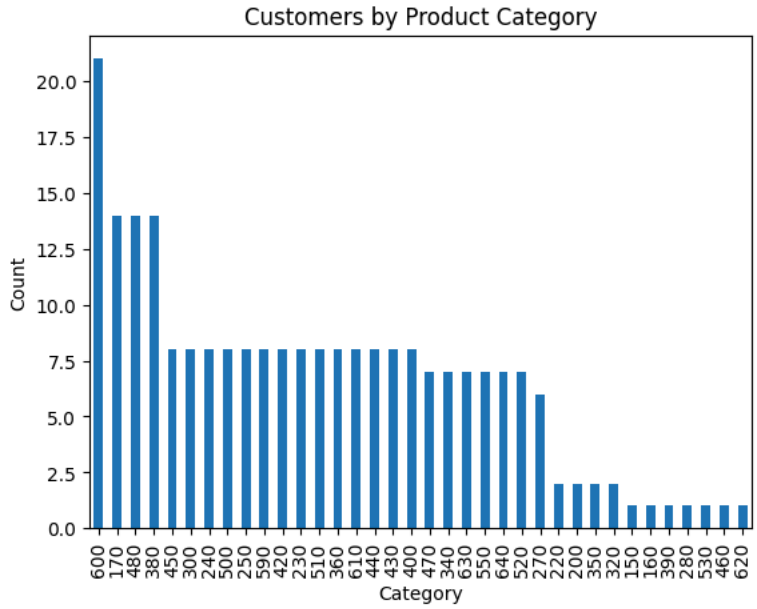
```
#15
df.describe()
```

	user_id	age	annual_income	purchase_amount	loyalty_score	purchase_frequency
count	238.000000	238.000000	238.000000	238.000000	238.000000	238.000000
mean	119.500000	38.676471	57407.563025	425.630252	6.794118	19.798319
std	68.848868	9.351118	11403.875717	140.052062	1.899047	4.562884
min	1.000000	22.000000	30000.000000	150.000000	3.000000	10.000000
25%	60.250000	31.000000	50000.000000	320.000000	5.500000	17.000000
50%	119.500000	39.000000	59000.000000	440.000000	7.000000	20.000000
75%	178.750000	46.750000	66750.000000	527.500000	8.275000	23.000000
max	238.000000	55.000000	75000.000000	640.000000	9.500000	28.000000

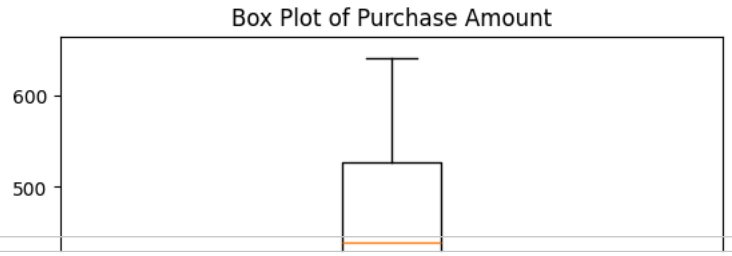
```
#16
plt.figure()
plt.hist(df['purchase_amount'])
plt.title("Purchase Amount")
plt.xlabel("Amount")
plt.ylabel("Frequency")
plt.show()
```



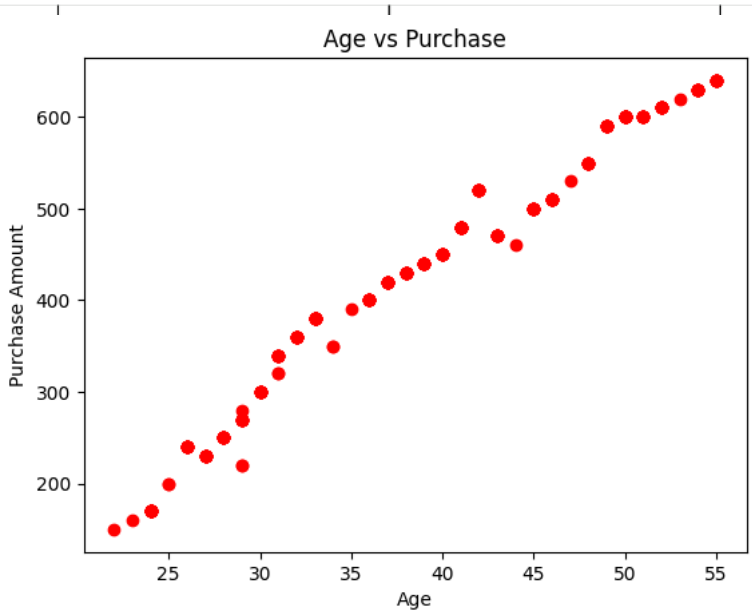

```
#17
plt.figure()
df["purchase_amount"].value_counts().plot(kind='bar')
plt.title("Customers by Product Category")
plt.xlabel("Category")
plt.ylabel("Count")
plt.show()
```



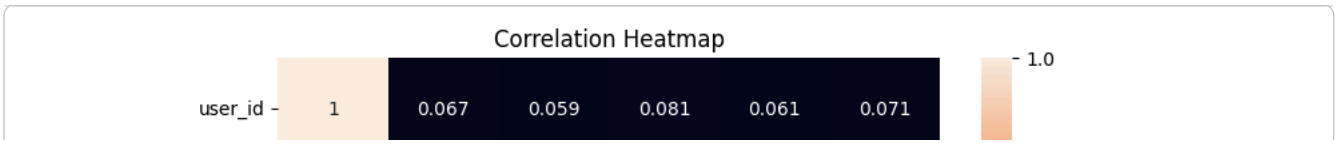

```
#18
plt.figure()
plt.boxplot(df["purchase_amount"])
plt.title("Box Plot of Purchase Amount")
plt.show()
```




```
#19
plt.figure()
plt.scatter(df["age"], df["purchase_amount"],color="red")
plt.xlabel("Age")
plt.ylabel("Purchase Amount")
plt.title("Age vs Purchase")
plt.show()
```




```
#20
plt.figure(figsize=(8,6))
sns.heatmap(df[num_cols].corr(), annot=True)
plt.title("Correlation Heatmap")
plt.show()
```


Correlation Heatmap
user_id
1
0.067
0.059
0.081
0.061
0.071
1.0