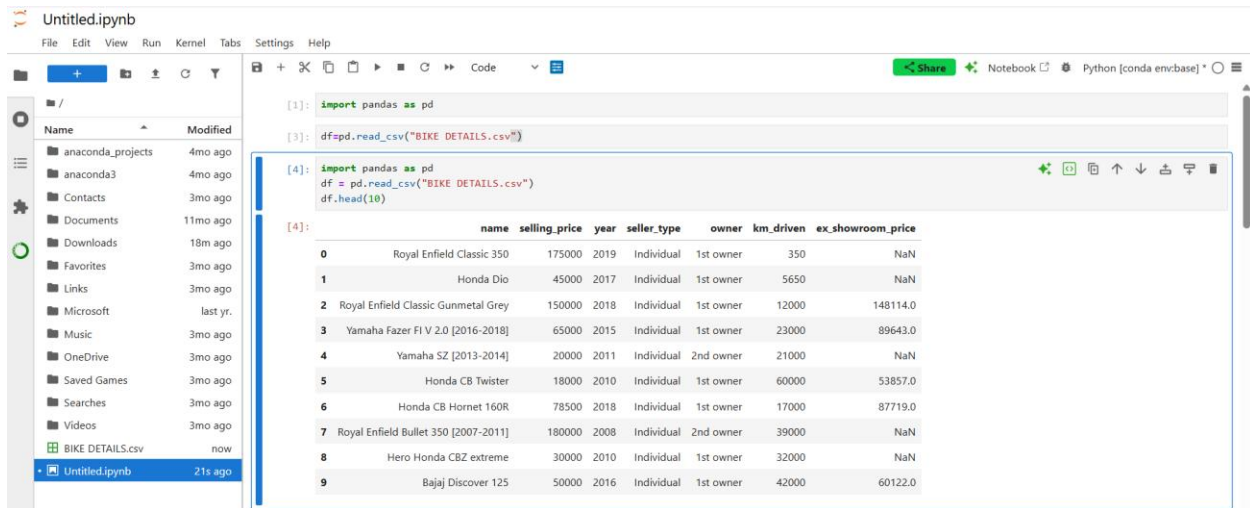


Assignment Code: DA-AG-009 EDA | Assignment

Yogita Hiwale

Question 1: Read the Bike Details dataset into a Pandas DataFrame and display its first 10 rows.



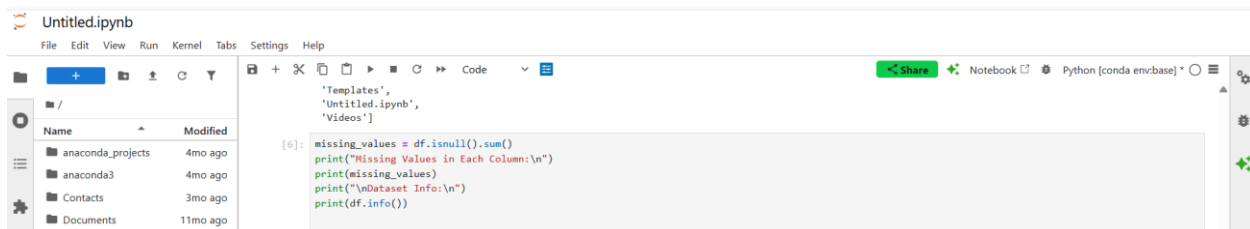
The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The file explorer lists various folders and files, including 'BIKE DETAILS.csv' and 'Untitled.ipynb'. The code editor contains the following Python code:

```
[1]: import pandas as pd
[2]: df=pd.read_csv("BIKE DETAILS.csv")
[4]: import pandas as pd
df = pd.read_csv("BIKE DETAILS.csv")
df.head(10)
```

The output of the code is a table with 10 rows and 8 columns: name, selling_price, year, seller_type, owner, km_driven, ex_showroom_price. The data is as follows:

	name	selling_price	year	seller_type	owner	km_driven	ex_showroom_price
0	Royal Enfield Classic 350	175000	2019	Individual	1st owner	350	NaN
1	Honda Dio	45000	2017	Individual	1st owner	5650	NaN
2	Royal Enfield Classic Gunmetal Grey	150000	2018	Individual	1st owner	12000	148114.0
3	Yamaha Fazer FI V 2.0 [2016-2018]	65000	2015	Individual	1st owner	23000	89643.0
4	Yamaha SZ [2013-2014]	20000	2011	Individual	2nd owner	21000	NaN
5	Honda CB Twister	18000	2010	Individual	1st owner	60000	53857.0
6	Honda CB Hornet 160R	78500	2018	Individual	1st owner	17000	87719.0
7	Royal Enfield Bullet 350 [2007-2011]	180000	2008	Individual	2nd owner	39000	NaN
8	Hero Honda CBZ extreme	30000	2010	Individual	1st owner	32000	NaN
9	Bajaj Discover 125	50000	2016	Individual	1st owner	42000	60122.0

Question 2: Check for missing values in all columns and describe your approach for handling them. (Include your Python code and output in the code box below.)



The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The file explorer lists various folders and files, including 'BIKE DETAILS.csv' and 'Untitled.ipynb'. The code editor contains the following Python code:

```
[6]: missing_values = df.isnull().sum()
print("Missing Values in Each Column:\n")
print(missing_values)
print("\nDataset Info:\n")
print(df.info())
```

Name

Modified

anaconda_projects

4mo ago

anaconda3

4mo ago

Contacts

3mo ago

Documents

11mo ago

Downloads

18mo ago

Favorites

3mo ago

Links

3mo ago

Microsoft

last yr.

Music

3mo ago

OneDrive

3mo ago

Saved Games

3mo ago

Searches

3mo ago

Videos

3mo ago

BIKE DETAILS.csv

now

Untitled.ipynb

now

Missing Values in Each Column:

name

0

selling_price

0

year

0

seller_type

0

owner

0

km_driven

0

ex_showroom_price

435

dtype: int64

Dataset Info:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1061 entries, 0 to 1060

Data columns (total 7 columns):

#

Column

Non-Null Count

Dtype

0

name

1061 non-null

object

1

selling_price

1061 non-null

int64

2

year

1061 non-null

int64

3

seller_type

1061 non-null

object

4

owner

1061 non-null

object

5

km_driven

1061 non-null

int64

6

ex_showroom_price

626 non-null

float64

dtypes: float64(1), int64(3), object(3)

memory usage: 58.2+ KB

None

Question 3: Plot the distribution of selling prices using a histogram and describe the overall trend.

(Include your Python code and output in the code box below.)

```
[7]: import matplotlib.pyplot as plt
|
| plt.figure(figsize=(8,5))
| plt.hist(df['selling_price'].dropna())
| plt.xlabel("Selling Price")
| plt.ylabel("Frequency")
| plt.title("Distribution of Selling Prices")
| plt.show()
```



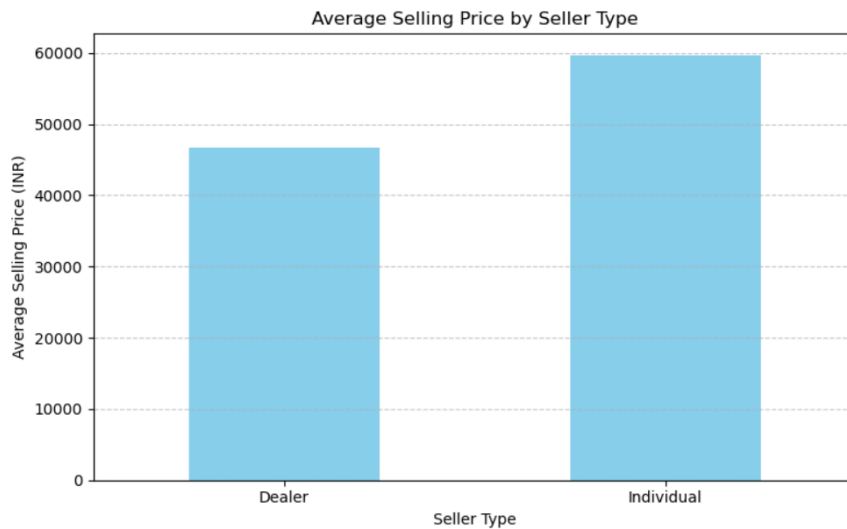
Question 4: Create a bar plot to visualize the average selling price for each seller_type and write one observation.

```
[8]: df = pd.read_csv("BIKE DETAILS.csv")

# NaN values hatao seller_type aur selling_price ke liye
df_clean = df.dropna(subset=["seller_type", "selling_price"])

# seller_type ke hisaab se average selling price nikalo
avg_price = df_clean.groupby("seller_type")["selling_price"].mean()

# Bar plot banao
plt.figure(figsize=(8, 5))
avg_price.plot(kind="bar", color="skyblue")
plt.title("Average Selling Price by Seller Type")
plt.xlabel("Seller Type")
plt.ylabel("Average Selling Price (INR)")
plt.xticks(rotations=0)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.tight_layout()
plt.show()
```



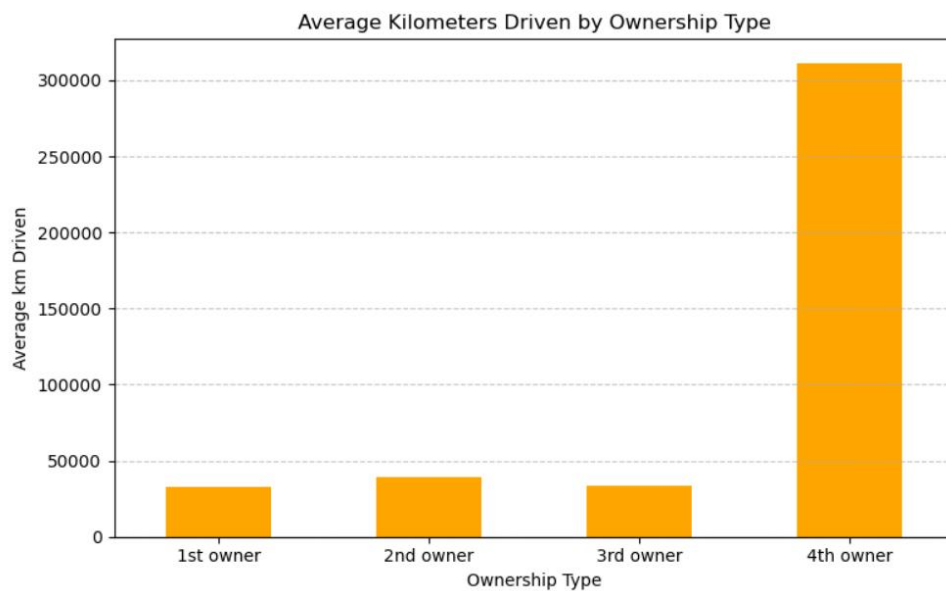
Question 5: Compute the average km_driven for each ownership type (1st owner, 2nd owner, etc.), and present the result as a bar plot.

```
[9]: df = pd.read_csv("BIKE DETAILS.csv")

# NaN values hatao owner aur km_driven ke liye
df_clean = df.dropna(subset=["owner", "km_driven"])

# owner ke hisaab se average km_driven nikaalo
avg_km = df_clean.groupby("owner")["km_driven"].mean()

# Bar plot banao
plt.figure(figsize=(8, 5))
avg_km.plot(kind="bar", color="orange")
plt.title("Average Kilometers Driven by Ownership Type")
plt.xlabel("Ownership Type")
plt.ylabel("Average km Driven")
plt.xticks(rotation=0)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.tight_layout()
plt.show()
```



Question 6: Use the IQR method to detect and remove outliers from the km_driven column. Show before-and-after summary statistics.

```
[10]:
df = pd.read_csv("BIKE DETAILS.csv")

df_clean = df.dropna(subset=["km_driven"])

print("Before removing outliers:")
print(df_clean["km_driven"].describe())

Q1 = df_clean["km_driven"].quantile(0.25)
Q3 = df_clean["km_driven"].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

df_no_outliers = df_clean[(df_clean["km_driven"] >= lower_bound) & (df_clean["km_driven"] <= upper_bound)]

print("\nAfter removing outliers:")
print(df_no_outliers["km_driven"].describe())
```

```
Before removing outliers:
count      1061.000000
mean       34359.833176
std        51623.152702
min         350.000000
25%        13500.000000
50%        25000.000000
75%        43000.000000
max        88000.000000
Name: km_driven, dtype: float64

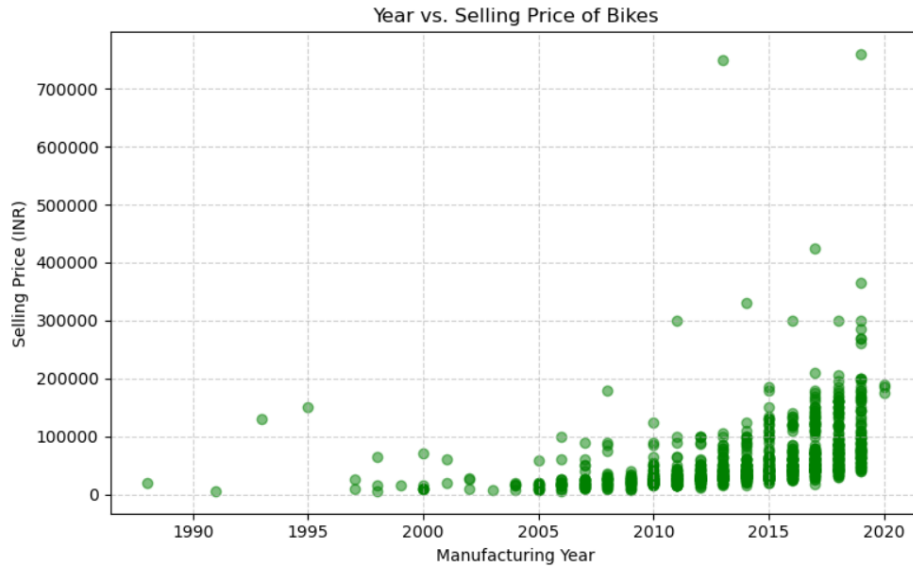
After removing outliers:
count      1022.000000
mean       28203.415851
std        19552.083583
min         350.000000
25%        13000.000000
50%        24000.000000
75%        40000.000000
max        86000.000000
Name: km_driven, dtype: float64
```

Question 7: Create a scatter plot of year vs. selling_price to explore the relationship between a bike's age and its price.

```
[11]:
df = pd.read_csv("BIKE DETAILS.csv")

df_clean = df.dropna(subset=["year", "selling_price"])

plt.figure(figsize=(8, 5))
plt.scatter(df_clean["year"], df_clean["selling_price"], alpha=0.5, color="green")
plt.title("Year vs. Selling Price of Bikes")
plt.xlabel("Manufacturing Year")
plt.ylabel("Selling Price (INR)")
plt.grid(True, linestyle="--", alpha=0.6)
plt.tight_layout()
plt.show()
```



Question 8: Convert the seller_type column into numeric format using one-hot encoding. Display the first 5 rows of the resulting DataFrame.

```
[12]: df = pd.read_csv("BIKE DETAILS.csv")
df_encoded = pd.get_dummies(df, columns=["seller_type"])
print(df_encoded.head())
```

	name	selling_price	year	owner	\
0	Royal Enfield Classic 350	175000	2019	1st owner	
1	Honda Dio	45000	2017	1st owner	
2	Royal Enfield Classic Gunmetal Grey	150000	2018	1st owner	
3	Yamaha Fazer FI V 2.0 [2016-2018]	65000	2015	1st owner	
4	Yamaha SZ [2013-2014]	20000	2011	2nd owner	

	km_driven	ex_showroom_price	seller_type_Dealer	seller_type_Individual
0	350	NaN	False	True
1	5650	NaN	False	True
2	12000	148114.0	False	True
3	23000	89643.0	False	True
4	21000	NaN	False	True

Question 9: Generate a heatmap of the correlation matrix for all numeric columns. What correlations stand out the most?

```
[14]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

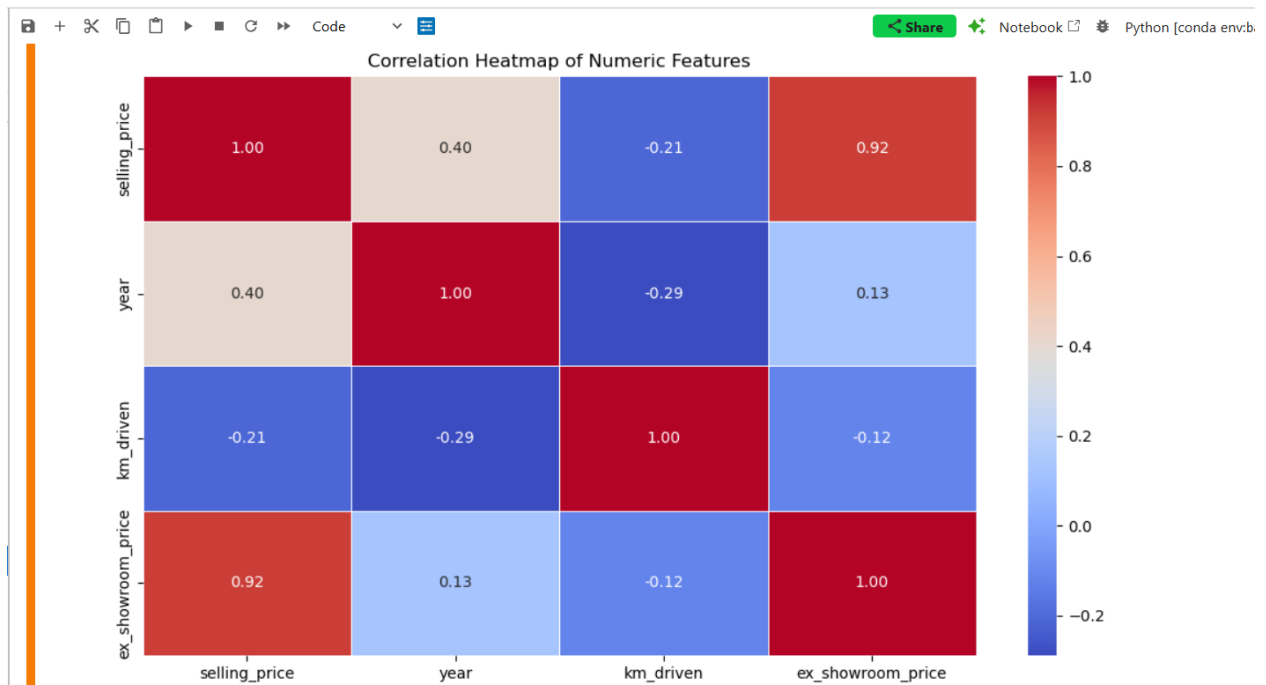
df = pd.read_csv("BIKE DETAILS.csv")

numeric_df = df.select_dtypes(include=["number"])

corr_matrix = numeric_df.corr()

|
plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Correlation Heatmap of Numeric Features")
plt.tight_layout()
plt.show()
```

Correlation Heatmap of Numeric Features



Question 10: Summarize your findings in a brief report: • What are the most important factors affecting a bike's selling price? • Mention any data cleaning or feature engineering you performed.

Key Factors Affecting a Bike's Selling Price

- **Year of Manufacture:** Newer bikes tend to have higher selling prices due to lower depreciation.
- **Kilometers Driven:** Bikes with fewer kilometers driven usually sell for more, indicating less wear and tear.
- **Ownership History:** First-owner bikes are priced higher than second or third-owner bikes, as buyers prefer fewer previous owners.
- **Ex-Showroom Price:** There is a strong positive correlation between the original price and the resale value.
- **Seller Type:** Most listings are from individual sellers, which influences pricing trends in the dataset.

Data Cleaning and Feature Engineering

- **Missing Value Handling:** Removed rows with missing values in critical columns like `selling_price`, `km_driven`, and `owner`.
- **Outlier Removal:** Applied the IQR method to the `km_driven` column to eliminate extreme values.
- **One-Hot Encoding:** Converted the `seller_type` column into numeric format using one-hot encoding for modeling.
- **Correlation Analysis:** Generated a heatmap to identify strong relationships between numeric features.