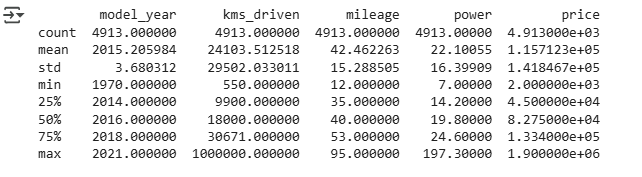
**Exploratory Data Analysis (EDA) - Project 2 - Internship**

We begin EDA by examining basic descriptive statistics for the numerical features using:

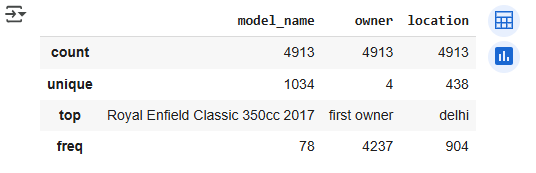
print(df.describe())



### **📌 Key Insights:**

1. model\_year ranges from **1970 to 2021** with a median around 2016.
2. kms\_driven shows high variability (max = 100,000 km), suggesting wide usage differences.
3. mileage ranges from **12 to 95 kmpl**, with an average of ~42.
4. power has an outlier max of 197.3 BHP (possibly a superbike).
5. price ranges widely from ₹2,000 to ₹19 lakhs (likely some outliers or luxury bikes).

df.describe(include='object')



### **📌 Key Insights:**

1. model\_name has 1,034 unique values; the most common is Royal Enfield Classic 350cc 2017 (78 listings).
2. owner is dominated by first owner entries (4,237 out of 4,913).
3. location includes 438 unique city/place entries — Delhi appears most frequently (904 times).

## **💰 Univariate Analysis – Price**

We explored the price column using histograms, boxplots, and categorical segmentation to understand how used bike prices are distributed.

### **📊 1. Distribution Plot**

We plotted a histogram to visualize the overall distribution of prices:

import matplotlib.pyplot as plt

import seaborn as sns

import matplotlib.ticker as ticker

plt.figure(figsize=(12, 6)) # Make plot wide enough

ax = sns.histplot(data=df, x='price', bins=50, kde=True, color='skyblue')

# Format X-axis with commas (₹ 10,000 instead of 10000)

ax.xaxis.set\_major\_formatter(ticker.FuncFormatter(lambda x, \_: f'{int(x):,}'))

plt.title('Distribution of Bike Prices')

plt.xlabel('Price (₹)')

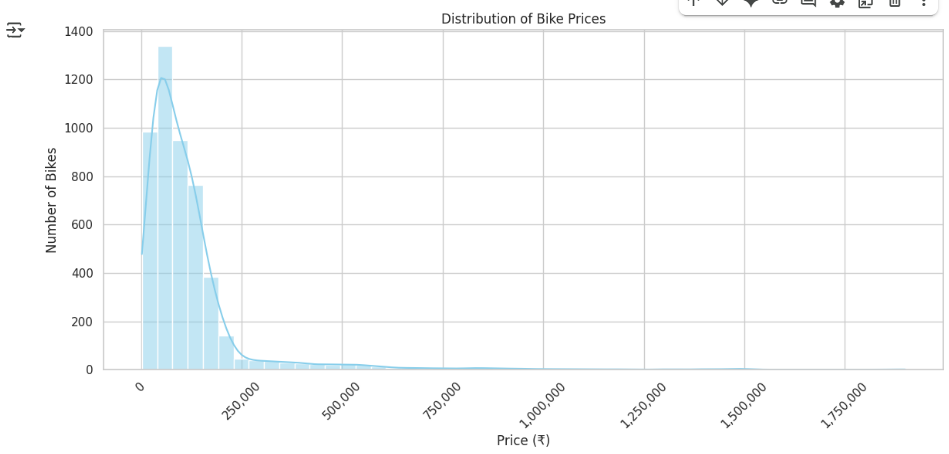
plt.ylabel('Number of Bikes')

plt.xticks(rotation=45) # Rotate for readability

plt.grid(True)

plt.tight\_layout()

plt.show()



#### **🔍 Observations:**

* The distribution is **right-skewed**, with most bikes priced in the lower range.
* Majority of bikes fall **under ₹2,50,000**, with fewer listings in the high-end range.

### **📊 2. Bar Plot of Price Segments (₹0 – ₹2.5L)**

To explore this further, we created price segments from ₹0 to ₹2.5L and visualized the frequency using a bar plot.

# Step 1: Define bins and labels

bins = [0, 50000, 100000, 150000, 200000, 250000]

labels = ['0–50K', '50K–1L', '1L–1.5L', '1.5L–2L', '2L–2.5L']

# Step 2: Create a new column with price segment

df['price\_segment'] = pd.cut(df['price'], bins=bins, labels=labels, include\_lowest=True)

# Step 3: Count bikes in each segment

segment\_counts = df['price\_segment'].value\_counts().sort\_index()

# Step 4: Plot

plt.figure(figsize=(10, 6))

sns.barplot(x=segment\_counts.index, y=segment\_counts.values, palette='pastel')

plt.title('Bike Count by Price Segment (₹0–2.5L)')

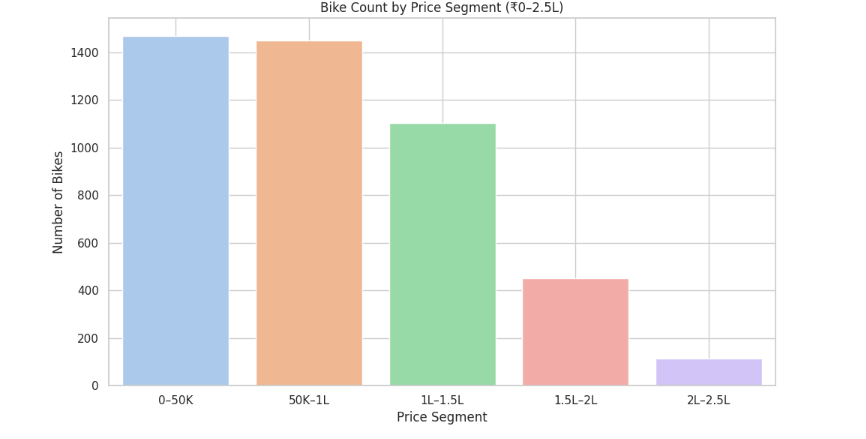
plt.xlabel('Price Segment')

plt.ylabel('Number of Bikes')

plt.grid(True)

plt.tight\_layout()

plt.show()



#### **🔍 Insights:**

* Most bikes are listed in the **₹0–₹50K** range.
* The second-highest volume is in the **₹50K–₹1L** segment.
* This confirms a **strong focus on budget-friendly used bikes** in the dataset.

✅ These insights help in defining pricing strategies, identifying high-volume segments, and selecting features for price prediction models

## ⚠️ **Price Outlier Detection and Handling**

### **🔍 Outlier Detection Using IQR**

To identify price outliers, we used the Interquartile Range (IQR) method:

Q1 = df['price'].quantile(0.25)

Q3 = df['price'].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

* Lower Bound: ₹ Q1 – 1.5 × IQR
* Upper Bound: ₹ Q3 + 1.5 × IQR
* Outliers Detected: 306 rows

### **🧠 Decision:**

Upon inspection, we found that many outliers were genuine bikes:

✅ Very old bikes (priced below ₹10,000)

✅ Premium bikes (priced above ₹3L)

✅ Niche or luxury models

Hence, dropping them would have led to loss of meaningful information.

### **✅ Final Action:**

Instead of removing them, we created a new column price\_outlier with binary values:

0 → Not an outlier

1 → Statistical outlier based on IQR

df['price\_outlier'] = ((df['price'] < lower\_bound) | (df['price'] > upper\_bound)).astype(int)

This allows us to:

Keep all records intact

Perform conditional analysis or filtering when needed

Improve fairness and transparency in modeling

### **📊 Visual Example :**

We also plotted a scatterplot of price vs. kms\_driven to visualize these outliers:

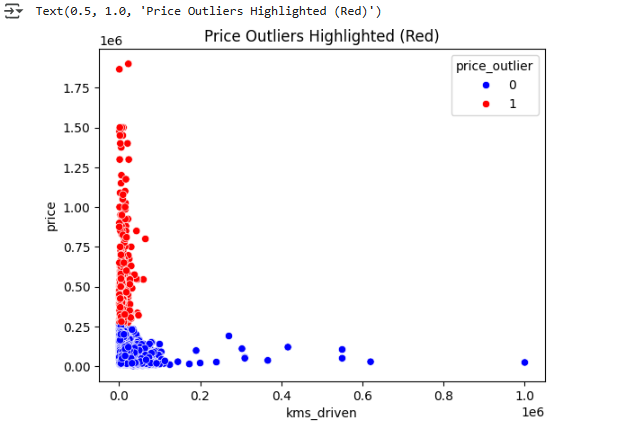
🔴 Red: Outliers

🔵 Blue: Normal values

import seaborn as sns

sns.scatterplot(data=df, x='kms\_driven', y='price', hue='price\_outlier', palette=['blue', 'red'])

plt.title("Price Outliers Highlighted (Red)")



### **📈 4. Skewness Check**

df['price'].skew()



The skewness value is **positive**, confirming the long tail towards higher prices. A log transformation may be considered during modeling.

**👤 5. Price by Owner Type**

We compared prices across different owner categories:

import seaborn as sns

import matplotlib.pyplot as plt

owner\_order = ['first owner', 'second owner', 'third owner', 'fourth owner or more']

sns.boxplot(

x='owner',

y='price',

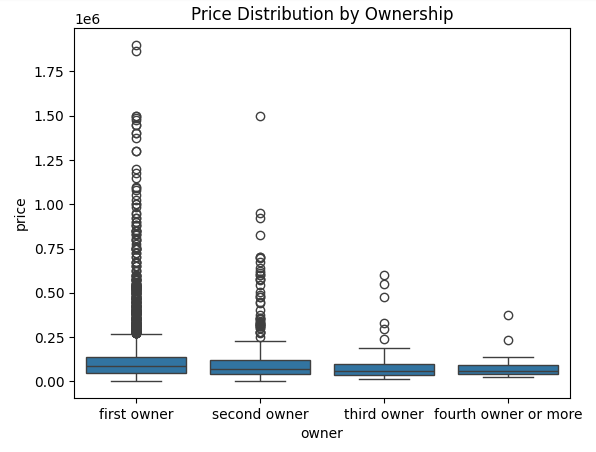
data=df,

order=owner\_order

)

plt.title("Price Distribution by Ownership")

plt.show()



**Insight:**

**First-owner** bikes are generally priced higher and are clustered to be most in number followed by second and third owners while **fourth-owner** bikes are few and at the lower end of the price range.

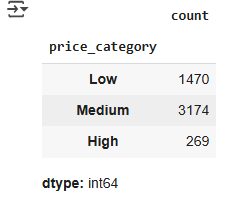
## **6. 📊 Price Segmentation Analysis**

To better understand how bike prices are distributed, we categorized the bikes into three price segments:

df['price\_category'] = pd.cut(df['price'], bins=[0, 50000, 300000, df['price'].max()], labels=['Low', 'Medium', 'High'])

As we can see  
**Low** = 0-50,000  
**Medium** = 50,000-300,000  
**High** = Above 300,000

df['price\_category'].value\_counts().sort\_index()

  
We can see that the Mediumcategory has the highest number of bikes while the Highcategory has the lowest number of bikes.

## **✅ Summary**

* The used bike market is heavily concentrated below ₹2.5L
* Budget bikes (₹0–₹50K) dominate the listings
* First-owner bikes command the highest number.
* We have seen positive skewness and the chart is left-skewed.
* Bikes between 50,000-30,0000 are the highest.