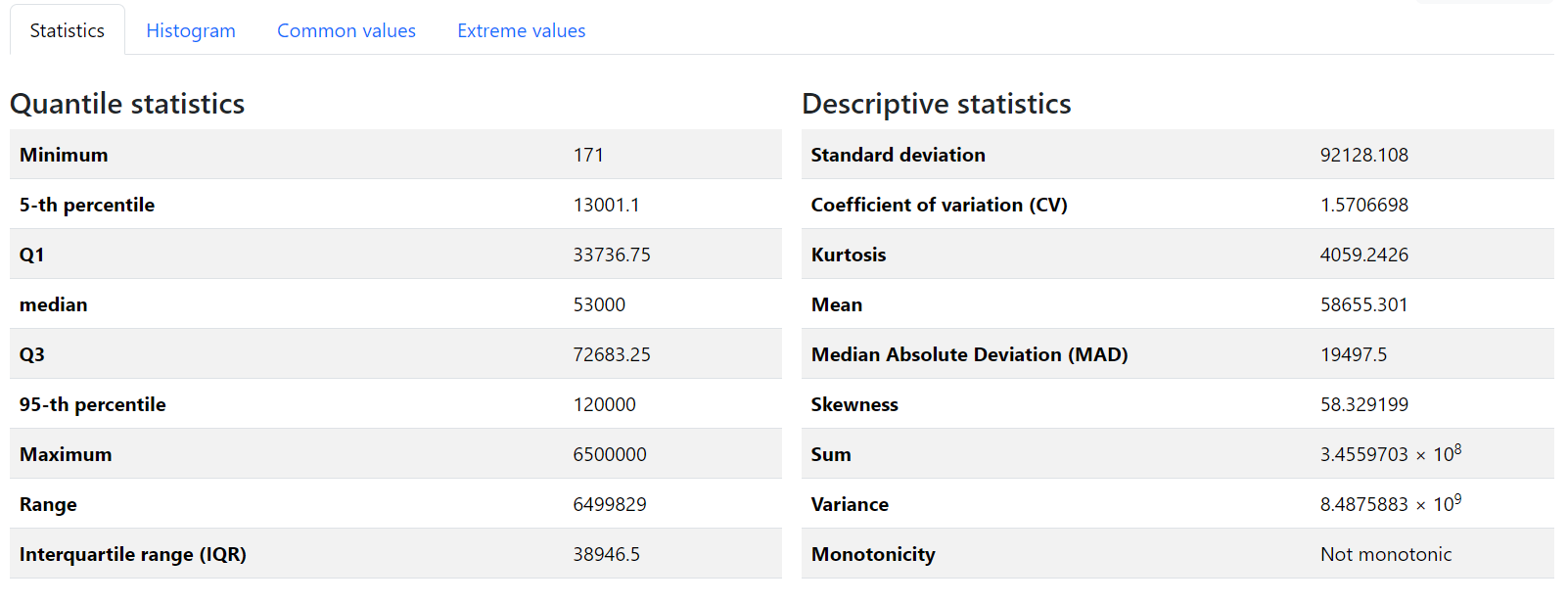
# Kilometers\_Driven



Our dataset's Kilometers\_Driven column shows significant variability and extreme outliers, which can heavily impact Our car price prediction model. Here’s an in-depth analysis:

### **1. Presence of Outliers**

* The **maximum value** (6,500,000 km) is extremely high compared to the **95th percentile** (120,000 km).
* The **range** (6,499,829 km) is enormous, suggesting potential outliers.
* **Kurtosis = 4059.24**, which is extremely high, indicating a heavy-tailed distribution with extreme outliers.

### **2. Skewness & Data Distribution**

* **Skewness = 58.33**, meaning the data is highly **right-skewed** (positively skewed). Most cars have lower Kilometers\_Driven, but a few cars have exceptionally high values.
* The **mean (58,655.3 km) is much higher than the median (53,000 km)**, confirming that the dataset is skewed.

### **3. Variability & Spread**

* **Standard deviation (92,128 km)** is very large, showing that Kilometers\_Driven values are widely spread.
* **IQR (38,946.5 km)** suggests that most data lies within a smaller range, but extreme values increase overall variation.
* **Coefficient of Variation (CV = 1.57)** indicates high relative variability.

### **4. Possible Data Preprocessing Steps for Price Prediction Model**

1. **Outlier Handling:**
   * Use **log transformation** (log1p(Kilometers\_Driven)) to reduce skewness.
   * Apply **winsorization** or **capping** (e.g., cap max values at the 95th percentile: 120,000 km).
   * Remove extreme outliers (e.g., values above 1,000,000 km may be unrealistic).
2. **Feature Transformation:**
   * Consider **binning Kilometers\_Driven** into categories (e.g., low, medium, high usage) to reduce the effect of extreme values.
3. **Scaling:**
   * Since Kilometers\_Driven has a high range, **Min-Max Scaling** or **Standardization (Z-score)** can help normalize the feature for regression models.

### **5. Impact on Car Price Prediction**

* Cars with extremely high mileage tend to have **lower prices**, but a few luxury or well-maintained cars might not follow this trend.
* Using **raw Kilometers\_Driven** in a model without transformation could lead to poor performance due to skewness.
* Feature engineering (like log transformation or categorical binning) can improve model accuracy.

# Mileage

Our dataset's Mileage column appears to have a more **normal and stable distribution** compared to Kilometers\_Driven. Here's an in-depth analysis:

### **1. Presence of Outliers**

* **Maximum (33.54 km/l) is not too extreme** compared to the **95th percentile (25.47 km/l)**.
* The **range (26.04 km/l)** is moderate, meaning there is no drastic spread.
* **Kurtosis = -0.27**, which is close to zero, indicating a relatively normal distribution with no heavy tails or extreme outliers.

### **2. Skewness & Data Distribution**

* **Skewness = 0.214**, which is **slightly right-skewed** but still close to a normal distribution.
* **Mean (18.32 km/l) and Median (18.2 km/l) are very close**, further supporting a near-normal distribution.

### **3. Variability & Spread**

* **Standard deviation (4.17 km/l)** indicates moderate variation.
* **IQR (5.8 km/l)** shows that the middle 50% of values are spread within this range, meaning Mileage values are fairly consistent.
* **Coefficient of Variation (CV = 0.23)** is relatively low, meaning Mileage has less variability relative to its mean.

### **4. Possible Data Preprocessing Steps for Price Prediction Model**

1. **Outlier Handling:**
   * Since there are no extreme outliers, no major action is needed.
   * However, capping at the **95th percentile (25.47 km/l)** may help stabilize the model.
2. **Feature Transformation:**
   * **No need for log transformation**, as the data is already close to normal.
   * **Binning Mileage** into categories like "Low", "Medium", and "High" could be useful for some models.
3. **Scaling:**
   * If using **distance-based models (e.g., KNN, Linear Regression, Neural Networks)**, **Standardization (Z-score)** can improve performance.

### **5. Impact on Car Price Prediction**

* **Mileage has a strong inverse relationship with car price**:
  + Higher mileage (better fuel efficiency) **usually** leads to a higher price, but exceptions exist (luxury cars may have lower mileage but higher prices).
* **Since the data is already well-behaved**, it will likely be one of the most useful features in the model.
* Including interactions like Mileage \* Engine Power can enhance predictive power.

# Engine

Our dataset's Engine column (likely representing engine displacement in cc) shows **moderate variability with some skewness**. Here's an in-depth analysis:

### **1. Presence of Outliers**

* **Maximum (5998 cc) is significantly higher than the 95th percentile (2982 cc)**, indicating potential outliers.
* **Range (5926 cc) is quite large**, suggesting a wide variation in engine sizes.
* **Kurtosis = 2.97**, which is slightly higher than 3, indicating a moderately heavy tail (some extreme values but not excessively so).

### **2. Skewness & Data Distribution**

* **Skewness = 1.40**, meaning the data is **right-skewed** (positively skewed).
* **Mean (1625 cc) is greater than the Median (1493 cc)**, confirming that larger engine sizes pull the average upward.
* A few cars with very high engine displacement (above 4000 cc) might be causing this skewness.

### **3. Variability & Spread**

* **Standard deviation (600.8 cc)** is large, indicating significant variation in engine sizes.
* **IQR (786 cc)** suggests that the middle 50% of cars have engines between 1198 cc and 1984 cc.
* **CV = 0.37**, meaning engine displacement has moderate variability compared to its mean.

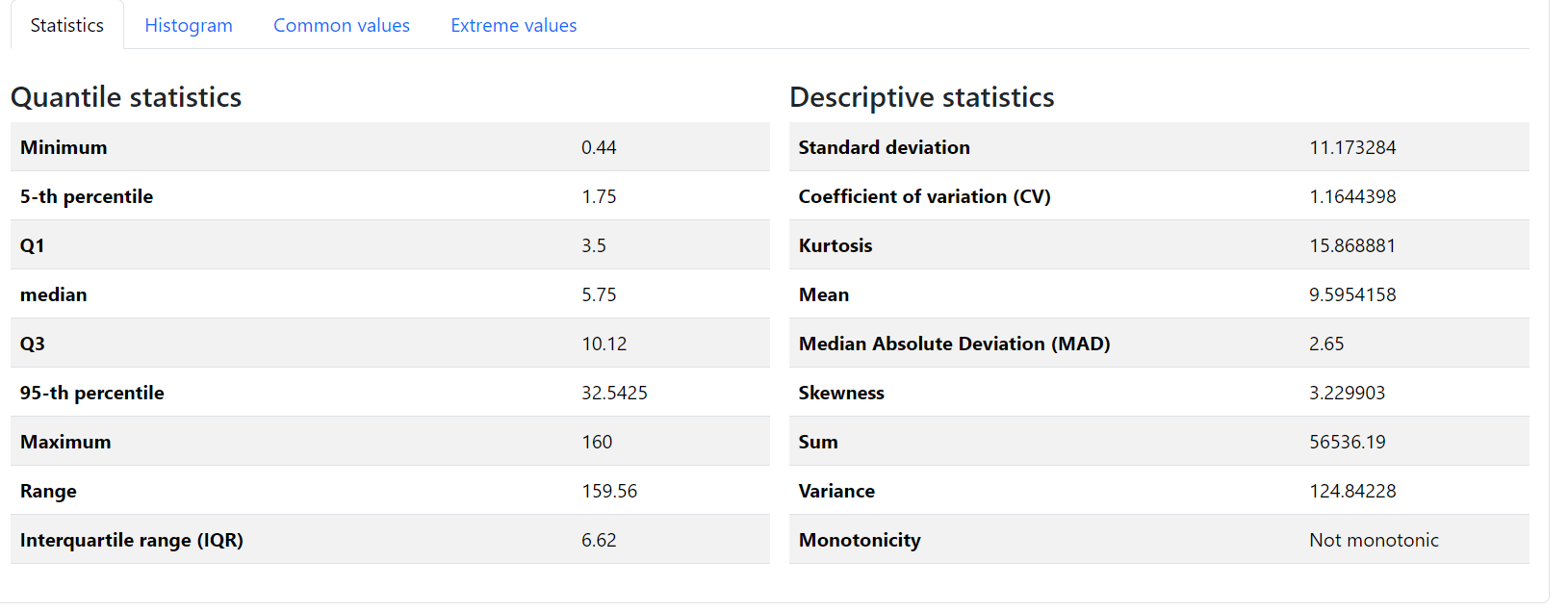
### **4. Possible Data Preprocessing Steps for Price Prediction Model**

1. **Outlier Handling:**
   * Consider **capping max values** at the **95th percentile (2982 cc)** to reduce the influence of extreme values.
   * Use **log transformation (log1p(Engine))** to make the distribution more normal.
2. **Feature Transformation:**
   * **Binning Engine Size** into categories (e.g., Small: <1200 cc, Medium: 1200-2000 cc, Large: >2000 cc) can help certain models.
3. **Scaling:**
   * If using distance-based models, apply **Standardization (Z-score)** for better performance.

### **5. Impact on Car Price Prediction**

* **Larger engines generally increase car prices**, but this relationship is **non-linear** (high-performance cars with large engines may have extreme prices).
* **Combining Engine with Mileage (Engine \* Mileage)** can help capture efficiency trade-offs.
* **Luxury brands often have larger engines, meaning Brand should be considered alongside Engine Size**.

# Price



Our dataset's Price column shows **extreme skewness and heavy-tailed distribution**, which can significantly impact model performance. Here's an in-depth analysis:

### **1. Presence of Outliers**

* **Maximum (160) is much higher than the 95th percentile (32.54)**, indicating extreme outliers.
* **Range (159.56) is enormous**, showing large variations in car prices.
* **Kurtosis = 15.87**, which is extremely high, indicating a heavy-tailed distribution with many extreme values.

### **2. Skewness & Data Distribution**

* **Skewness = 3.23**, meaning the data is **highly right-skewed**.
* **Mean (9.60) is much greater than the Median (5.75)**, confirming that a few high-priced cars are pulling the average up.

### **3. Variability & Spread**

* **Standard deviation (11.17)** is very high, indicating significant price variations.
* **IQR (6.62)** shows that the middle 50% of prices fall between **3.5 and 10.12**, but the tail extends far beyond that.
* **CV = 1.16**, indicating high variability in price relative to its mean.

### **4. Possible Data Preprocessing Steps for Price Prediction Model**

1. **Outlier Handling:**
   * Consider **capping prices** at the **95th percentile (32.54)** or using **log transformation (log1p(Price))** to normalize the distribution.
   * Check for possible **data entry errors** for extreme values.
2. **Feature Transformation:**
   * Since price is the **target variable**, use **log transformation** to stabilize variance and improve model performance.
   * **Binning prices** into categories (e.g., Low, Mid, High) may be useful for classification-based models.
3. **Scaling:**
   * If using regression models, applying **Min-Max Scaling** or **Standardization** can improve performance.

### **5. Impact on Car Price Prediction**

* Since price is the **dependent variable**, high skewness can make predictions inaccurate.
* A **log-transformed price (log1p(Price))** will help models capture relative price differences better.
* Luxury cars with high prices could be treated separately to avoid distortion in the model.

# Kilometers\_Driven\_log

Your Kilometers\_Driven\_log column (log-transformed version of Kilometers\_Driven) shows a **much-improved distribution** compared to the original Kilometers\_Driven column. Here's an in-depth analysis:

### **1. Presence of Outliers**

* **Maximum (15.69) is higher than the 95th percentile (11.70),** but the gap is far smaller than in the original column.
* **Range (10.55) is much smaller compared to the original Kilometers\_Driven column**, indicating effective compression of extreme values.
* **Kurtosis = 4.72**, slightly higher than 3, suggesting a moderately heavy-tailed distribution.

### **2. Skewness & Data Distribution**

* **Skewness = -1.29**, meaning the data is now **left-skewed** after the log transformation.
* **Mean (10.76) is slightly lower than the Median (10.88),** confirming slight left skewness.
* The transformation has significantly **reduced right-skewness from the original data**, making it more normal-like.

### **3. Variability & Spread**

* **Standard deviation (0.72)** is much smaller than in the original column, meaning values are now more tightly clustered.
* **IQR (0.77)** is also very small, confirming that most values are concentrated in a narrow range.
* **Coefficient of Variation (CV = 0.067)** is very low, meaning Kilometers\_Driven\_log is now much more stable than the original.

### **4. Why This Transformation is Useful**

* The log transformation has **significantly reduced the impact of extreme values**, making Kilometers\_Driven\_log more suitable for modeling.
* Left-skewness might indicate that **some very low Kilometers\_Driven values exist**, but this is less problematic than right-skewness.
* This transformed feature is now better suited for **linear regression models**, as it reduces heteroscedasticity.

### **5. Impact on Car Price Prediction**

* **This log-transformed feature will likely improve the model’s predictive performance**, as it balances the distribution.
* If you use models like **Linear Regression, Ridge, or Lasso**, Kilometers\_Driven\_log should perform better than Kilometers\_Driven.
* If using **Tree-based models (Random Forest, XGBoost)**, the transformation may not be necessary, but keeping both versions can help in feature selection.

# price\_log

Your price\_log column (log-transformed version of Price) shows a **much-improved distribution** compared to the original Price column. Here's an in-depth analysis:

### **1. Presence of Outliers**

* **Maximum (5.08) is higher than the 95th percentile (3.48),** but extreme values have been significantly compressed.
* **Range (5.90) is much smaller compared to the original Price column,** indicating successful reduction of extreme variation.
* **Kurtosis = 0.17**, close to zero, indicating a **near-normal distribution** with minimal heavy tails.

### **2. Skewness & Data Distribution**

* **Skewness = 0.41**, which is only slightly right-skewed, meaning the transformation has **significantly reduced** the extreme skewness present in the original Price column (which had a skewness of 3.23).
* **Mean (1.84) and Median (1.75) are very close**, further supporting a **more balanced distribution**.

### **3. Variability & Spread**

* **Standard deviation (0.87) is now much lower** than in the original Price column, indicating more compact data.
* **IQR (1.06)** confirms that most values are tightly packed between **1.25 and 2.31**.
* **Coefficient of Variation (CV = 0.47)** is much lower than in the original Price, suggesting more stability.

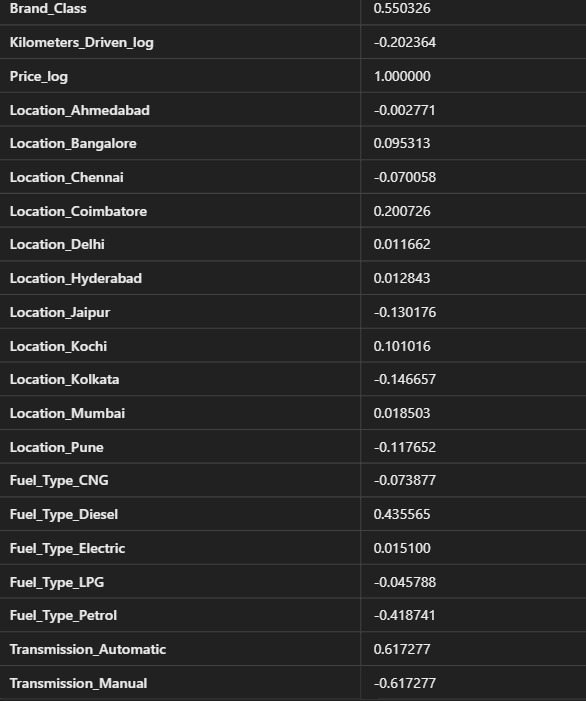
### **4. Why This Transformation is Useful**

* The log transformation has **effectively stabilized variance** and **reduced the effect of extreme high prices**.
* A near-normal distribution is ideal for **linear regression models**, as it improves predictive accuracy and reduces heteroscedasticity.
* The transformation ensures that **percentage changes in price** are more meaningful than absolute changes.

### **5. Impact on Car Price Prediction**

* price\_log is a much **better target variable** than raw Price for models like **Linear Regression, Ridge, Lasso, and Neural Networks**.
* If using **Tree-based models (Random Forest, XGBoost, LightGBM)**, keeping both Price and price\_log might be beneficial to see which works best.
* This transformation allows the model to **better capture relative price differences** rather than being overly influenced by high-end luxury cars.

# 🔹 Key Observations



This correlation table shows how each feature relates to **Price\_log**, which is the log-transformed price of the car. Let's analyze the key insights:

### **1️⃣ Strongly Positively Correlated Features (High Impact on Price)**

These features have a strong positive correlation with Price\_log (closer to **1**), meaning **higher values of these features lead to higher car prices**:

* **Power (0.767)** 🔥 → More powerful cars are priced higher.
* **Engine (0.687)** 🚗 → Larger engines generally indicate more premium cars.
* **Transmission\_Automatic (0.617)** ⚙ → Automatic cars are more expensive than manual ones.
* **Fuel\_Type\_Diesel (0.436)** ⛽ → Diesel cars tend to have a higher price.
* **Brand\_Class (0.550)** 🏷 → More premium brands tend to have higher prices.

### **2️⃣ Strongly Negatively Correlated Features (Lower Values → Higher Price)**

These features have a **strong negative correlation** with Price\_log, meaning **higher values in these features lower the car price**:

* **Ageofcar (-0.499)** 📉 → Older cars have lower prices.
* **Mileage (-0.307)** ⛽ → Higher mileage cars are often cheaper, likely due to wear and tear.
* **Kilometers\_Driven\_log (-0.202)** 🚘 → More driven cars tend to have lower resale values.
* **Owner\_Type (-0.198)** 👤 → Cars with more previous owners are cheaper.

### **3️⃣ Fuel Type & Transmission Effects**

* **Fuel\_Type\_Diesel (+0.436)** 🛢 → Diesel cars have **higher prices**.
* **Fuel\_Type\_Petrol (-0.419)** ⛽ → Petrol cars have **lower prices**.
* **Transmission\_Automatic (+0.617)** ⚙ → Automatic cars are much **higher in price**.
* **Transmission\_Manual (-0.617)** → Manual cars are **cheaper**.

### **4️⃣ Location-Based Effects**

* **Coimbatore (+0.201), Kochi (+0.101), Bangalore (+0.095)** → Cars in these cities are priced higher.
* **Jaipur (-0.130), Kolkata (-0.147), Pune (-0.118)** → Cars in these cities tend to have lower prices.

## 

## **📌 Final Thoughts**

* **Price is strongly driven by Power, Engine, Transmission Type, and Brand Class.**
* **Age of the car, mileage, and owner count reduce prices.**
* **Automatic and diesel cars are more expensive than manual and petrol cars.**
* **City-wise variations exist in car pricing.**
* **The log transformation (Price\_log) helped normalize price-related correlations.**

# 

# dropping **Location columns**

is a reasonable decision based on your correlation analysis. Here's why:

### **📌 Why Dropping Location is a Good Choice?**

1. **Low Correlation with Price\_log**
   * The correlation values of location-based features range between **-0.15 to +0.20**, which is quite weak.
   * This means location does **not significantly** impact price prediction.
2. **Reduces Model Complexity**
   * One-hot encoding **Location** creates multiple new columns, increasing dimensionality.
   * More features can lead to overfitting and higher computational costs without adding much predictive power.
3. **Generalization Improvement**
   * By removing location data, the model learns to generalize better across different cities.
   * A model trained without location dependence can work for **new unseen locations** without issues.

### **🔹 When Should You Keep Location?**

* If **specific cities influence pricing**, like high-demand areas.
* If price trends differ **significantly** between locations (e.g., metro vs. small cities).
* If you plan on training separate models for different locations.

### **📌 Final Decision**

✅ **Drop Location columns** for a more general, efficient model.  
 ❌ If you suspect location **does impact price**, you could test it with feature importance analysis later.