SIM ID: 10263704

NAME: Subathra Sundarbabu

DATE: 3 Jan 2025

CM 2015 Programming With Data Mid Term

How do people's choice of local versus global brands in India relate to global

1 An introduction to the research space:

The used car marking in india is growing rapidly, with people choosing between different local and global brands. Their choices are oftern influenced by many factors like cost, brand, fuel type and many more. Local brands are often seen as affordable and practical while global brands offer advanced features with a premium image. This research will explore how the preference for local versus global car brands in India's used car market relate to global car production trends. It will give us insights into consumer choices in india reflect on global car manufacturing patters.

1.1 Aim & Objectives:

The aim is to understand how people in india choose between local and global car brands. The main objective is to analyse the factors influencing brand preferences, compare the popularity of local brands with global brands and explore how these choices relate to global car production trends. This report will also explore how car prices, different car features and brand reputation impact consumer decisions, and provide insights into the overall trends in the indian used car market.

1.2 Acquire a dataset:

For this research, I will use 2 datasets:

1. **Used car in India**: This data is from Kaggle. It includes information about car brands, models, age, mileage, and prices.
2. **Global Car Production By Region**: This dataset was web scraped from a website and shows the total number of car production for each country.

These datasets are good for this research because they provide detailed information on India's used car market and global car production, allowing for a comparison of India's car preferences and global production trends.

1.3 Utilise the dataset:

The datasets will be used to explore brand preferences in the Indian used car market and compare them to global car production trends. The used car dataset will help analyze the popularity of local brands and global brands.

The global car production dataset will provide insights into production trends. This is allow us to see how India's brand preferences align with global manufacturing patterns.

I have combined both dataset. Combining both datasets will offer a clearer understanding of consumer behavior in India.

1.4 Communicate ideas and concepts clearly:

The goal is to explain how people in India choose between local and global car brands and how this connects to global car production trends. The analysis will be presented clearly, using charts to make the data easy to understand. This approach ensures that the research is easy to follow and provides useful insights about the link between car buying habits in India and global car production trends.

1.5 Summary of the area of research:

This research focuses on understanding how people in India choose between local and global brands in the used car market. It explores the factors that influence these choices. The research also looks at how these preferences align with global car production trends, comparing car manufacturing data from different countries. By analyzing these patterns, the study aims to provide insights into consumer behavior in India and its connection to global production trends.

2 Project background

2.1 Why the field is of interest/relevant

This field is important because it examines the balance between local and global automotive brands in India's used car market, which is a sector that is growing rapidly. Understanding these preferences can provide valuable insights for manufacturers and market analysts.

2.2 Previous exploration of the topic

While there are studies on global car production and Indian market trends separately, the connection between Indian consumer preferences and global car production trends has not been fully explored, making this research unique.

2.3 Scope of Work

The project will analyze Indian used car preferences based on brand and their relation to global production trends. It will not cover unrelated factors like regional taxes or export-import policies.

2.4 Steps in data processing pipeline

The research involves cleaning and preparing the datasets, performing exploratory data analysis, creating visualizations to identify trends, and comparing brand preferences in India to global production data.

2.5 How I will evaluate my aims and objectives

The evaluation will focus on analyzing trends and patterns in the data to see if they support the research question, ensuring that the objectives are addressed and the insights align with the aims.

3 Ethics of use of data have been considered

3.1 Data Provenace and licensing

The dataset comes from open sources, Kaggle and web-scraped global car production statistics. These sources are publicly accessible and licensed for research and analysis. Proper attribution is given to the original sources to respect intellectual property rights and maintain transparency.

3.2 Intellectual Property and Attribution

The analysis has the potential to generate new insights and intellectual property by uncovering trends in car brand preferences and their alignment with global production. All findings will be appropriately attributed to the dataset creators, ensuring ethical usage and acknowledgment.

3.3 Ethical Implications of data use

Careful consideration is given to the implications of using this data. The analysis avoids promoting harmful assumptions or conclusions and ensures the research findings do not discriminate any groups. All interpretations will be context-sensitive and justified.

3.4 Data Processing and Accessibility

The data is accessible and processed directly in the notebook. Clear documentation of processing steps ensures traceability, enabling others to understand the workflow and outcomes.

3.5 Bias Considerations

The potential for bias in the dataset has been reviewed. If the data disproportionately represents certain brands or countries, this bias will be acknowledged and addressed in the analysis to provide balanced insights. This ensures the findings are fair.

4 Dataset 1: Kaggle Dataset (Used Car)

4.1 Import Libraries

In [1]:
```
# Load the libraries: To start of with this analyses, we need the pandas library to
import pandas as pd
```

4.2 Load Dataset

In [2]:
```
# Load the kaggle dataset
used_car_df = pd.read_csv("used_car_dataset.csv")
```

To view the first 5 rows, I have used the head function. This provides me with the overview of the different columns and the data stored.

In [3]:
```
# To view the first 5 row
used_car_df.head(5)
```

Out[3]:

	Brand	model	Year	Age	kmDriven	Transmission	Owner	FuelType	PostedDa
0	Honda	City	2001	23	98,000 km	Manual	second	Petrol	Nov-2
1	Toyota	Innova	2009	15	190000.0 km	Manual	second	Diesel	Jul-2
2	Volkswagen	VentoTest	2010	14	77,246 km	Manual	first	Diesel	Nov-2
3	Maruti Suzuki	Swift	2017	7	83,500 km	Manual	second	Diesel	Nov-2
4	Maruti Suzuki	Baleno	2019	5	45,000 km	Automatic	first	Petrol	Nov-2

4.3 Relevancy of data and justified use of data source

4.3.1 Origin of the dataset

The dataset comes from Kaggle and contains information on used cars in the Indian market. It includes details such as car brand, model, year, price, and other features, with 9,582 entries in total. This dataset was collected from various online sources of car listings in India, providing a comprehensive view of the used car market.

4.3.2 Why this data source is appropriate for the research question posed

It provides detailed information on car brands, models, prices, and other features in the Indian market. These variables are essential for analyzing brand preferences in India, specifically comparing local brands like Maruti Suzuki with global brands such as Volkswagen and Toyota.

4.3.3 Clearly identifiable case for working with this specific type of data

Brand, Model, and AskPrice columns, are directly linked to understanding consumer preferences in India, making it suitable for this research.

4.3.4 Format of data is suitable for analysis

Ussed car dataset is in csv format, making them easy to load and analyse using python. It have numerical and categorical data that supports statistical and comparative analysis.

4.3.5 Consideration of at least two other datasets

Car sales data from differenet dealerships could offer more insights into consumer demand but might be geographically limited. A car registration dataset could provide real-time data but may lack details on car features or brand preferences.

I have used the count function to count the total number values in each column. From this, we can get an overview of how many total number of rows and values are there.

In [4]:
```
# Counting non-null values in each column
used_car_df.count()
```

Out[4]:
```
Brand           9582
model           9582
Year            9582
Age             9582
kmDriven        9535
Transmission    9582
Owner           9582
FuelType        9582
PostedDate      9582
AdditionInfo    9582
AskPrice        9582
dtype: int64
```

Using the info function, I have print the information about the DataFrame. It provides a summary of the DataFrame, showing the data types, number of non-null entries, and memory usage.

In [5]:
```
# To print information about the DataFrame
used_car_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9582 entries, 0 to 9581
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Brand         9582 non-null   object
 1   model         9582 non-null   object
 2   Year          9582 non-null   int64 
 3   Age           9582 non-null   int64 
 4   kmDriven      9535 non-null   object
 5   Transmission  9582 non-null   object
 6   Owner         9582 non-null   object
 7   FuelType      9582 non-null   object
 8   PostedDate    9582 non-null   object
 9   AdditionInfo  9582 non-null   object
 10  AskPrice      9582 non-null   object
dtypes: int64(2), object(9)
memory usage: 823.6+ KB
```

Based on the output of the info function, we can see that the data type for all columns are object except Age which is int64. Moreover, all the columns have 9582 non-null values except kmDriven column which have 9535 non-null values. To get a clear view of the total number of null values I have also used the isnull function.

In [6]:
```
# To find the total null values in each column
used_car_df.isnull().sum()
```

Out[6]:
```
Brand           0
model           0
Year            0
Age             0
kmDriven       47
Transmission    0
Owner           0
FuelType        0
PostedDate      0
AdditionInfo    0
AskPrice        0
dtype: int64
```

The output shows that only the kmDriven column has 47 missing values, while all other columns have none. We need to address these missing values to get a cleaner dataset for our analysis.

4.4 Data Cleaning

4.4.1 Cleaning kmDriven Column

I need to convert the datatype of kmDriven column from object to float, then fill in the missing values with median. Since the dataset is large, filling the missing values with the median is a best option as it reduces the impact of outliers in the kmDriven column.

In [7]:
```
# Convert kmDriven column to numeric, removing the km and comma
used_car_df['kmDriven'] = used_car_df['kmDriven'].str.replace(' km', '', regex=True
```

I have used the describe function to generate summary statistics for the numerical columns in the DataFrame, including count, mean, standard deviation, min, and max values.

In [8]:
```
# Check the distribution for the different columns
used_car_df.describe()
```

Out[8]:

	Year	Age	kmDriven
count	9582.000000	9582.000000	9535.000000
mean	2016.361094	7.638906	70605.891453
std	4.087226	4.087226	56308.596299
min	1986.000000	0.000000	0.000000
25%	2014.000000	5.000000	43000.000000
50%	2017.000000	7.000000	65000.000000
75%	2019.000000	10.000000	86000.000000
max	2024.000000	38.000000	980002.000000

Based from the describe output, we can see that:

1. Most cars in the dataset are between 5-10 years old and have driven 43,000-86,000 km, based on the 25th and 75th percentiles for age and km driven.
2. Newer cars tend to have lower mileage, while older cars have higher mileage, as seen in the distribution of the Age and kmDriven columns.
3. The dataset mainly consists of cars from 2014-2019, reflecting the median and range of years, indicating a balance of older and newer vehicles.

Next, I have imported the warnings library to avoid any warning messages that might appear when running the code, ensuring the output remains clean and focused.

In [9]:
```
import warnings
warnings.filterwarnings("ignore")
```

As I mentioned earlier, I have used the fillna function to fill the null values in the kmDriven column with the median value, ensuring that the data is clean.

In [10]:
```
# Fill missing values with the median
used_car_df['kmDriven'].fillna(used_car_df['kmDriven'].median(), inplace=True)
```

To check all the null values are treated properly in the kmDriven column, I have used the isnull function again.

In [11]:
```
# Verify there is no missing values
used_car_df.isnull().sum()
```

Out[11]:
```
Brand           0
model           0
Year            0
Age             0
kmDriven        0
Transmission    0
Owner           0
FuelType        0
PostedDate      0
AdditionInfo    0
AskPrice        0
dtype: int64
```

4.4.2 Cleaning AskPrice Column

AskPrice column needs to be cleaned because it contains non-numeric values, price symbol. Cleaning the column will ensure that all values are in a consistent, numerical format for accurate analysis.

In [12]:
```
used_car_df.head()
```

Out[12]:

	Brand	model	Year	Age	kmDriven	Transmission	Owner	FuelType	PostedDa
0	Honda	City	2001	23	98000.0	Manual	second	Petrol	Nov-2
1	Toyota	Innova	2009	15	190000.0	Manual	second	Diesel	Jul-2
2	Volkswagen	VentoTest	2010	14	77246.0	Manual	first	Diesel	Nov-2
3	Maruti Suzuki	Swift	2017	7	83500.0	Manual	second	Diesel	Nov-2
4	Maruti Suzuki	Baleno	2019	5	45000.0	Automatic	first	Petrol	Nov-2

As mentioned earlier, the AskPrice column has the price symbol and we need to remove that to analyse the dataset further. Hence, we need to convert it to a numeric format for analysis.

In [13]:
```
# Clean and convert 'AskPrice' to numeric
used_car_df['AskPrice'] = used_car_df['AskPrice'].str.replace('₹', '', regex=True).
```

In [14]:
```
# Verify the cleaning process is done
used_car_df.head()
```

Out[14]:

	Brand	model	Year	Age	kmDriven	Transmission	Owner	FuelType	PostedDa
0	Honda	City	2001	23	98000.0	Manual	second	Petrol	Nov-2
1	Toyota	Innova	2009	15	190000.0	Manual	second	Diesel	Jul-2
2	Volkswagen	VentoTest	2010	14	77246.0	Manual	first	Diesel	Nov-2
3	Maruti Suzuki	Swift	2017	7	83500.0	Manual	second	Diesel	Nov-2
4	Maruti Suzuki	Baleno	2019	5	45000.0	Automatic	first	Petrol	Nov-2

4.5 Dropping Columns

I need to remove the PostedDate and AdditionInfo columns because as it does not contribute any value to my analysis.

In [15]:
```
# Dropping columns that are not relevant
used_car_df.drop(['PostedDate', 'AdditionInfo'], axis=1, inplace=True)
used_car_df.head()
```

Out[15]:

	Brand	model	Year	Age	kmDriven	Transmission	Owner	FuelType	AskPrice
0	Honda	City	2001	23	98000.0	Manual	second	Petrol	195000.0
1	Toyota	Innova	2009	15	190000.0	Manual	second	Diesel	375000.0
2	Volkswagen	VentoTest	2010	14	77246.0	Manual	first	Diesel	184999.0
3	Maruti Suzuki	Swift	2017	7	83500.0	Manual	second	Diesel	565000.0
4	Maruti Suzuki	Baleno	2019	5	45000.0	Automatic	first	Petrol	685000.0

4.6 Lambda Function

To categorize the brands into local and global I have used the unique function to find out the different brands.

In [16]:
```
unique_brands = used_car_df['Brand'].unique()
unique_brands
```

Out[16]:
```
array(['Honda', 'Toyota', 'Volkswagen', 'Maruti Suzuki', 'BMW', 'Ford',
       'Kia', 'Mercedes-Benz', 'Hyundai', 'Audi', 'Renault', 'MG',
       'Volvo', 'Skoda', 'Tata', 'Mahindra', 'Mini', 'Land Rover', 'Jeep',
       'Chevrolet', 'Jaguar', 'Fiat', 'Aston Martin', 'Porsche', 'Nissan',
       'Force', 'Mitsubishi', 'Lexus', 'Isuzu', 'Datsun', 'Ambassador',
       'Rolls-Royce', 'ICML', 'Bajaj', 'Opel', 'Ashok', 'Bentley',
       'Ssangyong', 'Maserati'], dtype=object)
```

I have used lambda function to categorize the brands into Local and Global based on my list of unique brand values, which can later use for visualization and analysis.

In [17]:
```
local_brands = ["Maruti Suzuki", "Tata", "Mahindra"]
used_car_df['BrandCategory'] = used_car_df['Brand'].apply(lambda x: 'Local' if x in
used_car_df.head()
```

Out[17]:

	Brand	model	Year	Age	kmDriven	Transmission	Owner	FuelType	AskPrice
0	Honda	City	2001	23	98000.0	Manual	second	Petrol	195000.0
1	Toyota	Innova	2009	15	190000.0	Manual	second	Diesel	375000.0
2	Volkswagen	VentoTest	2010	14	77246.0	Manual	first	Diesel	184999.0
3	Maruti Suzuki	Swift	2017	7	83500.0	Manual	second	Diesel	565000.0
4	Maruti Suzuki	Baleno	2019	5	45000.0	Automatic	first	Petrol	685000.0

4.6 Data Exploration and Visualization

4.6.1 Import Libraries

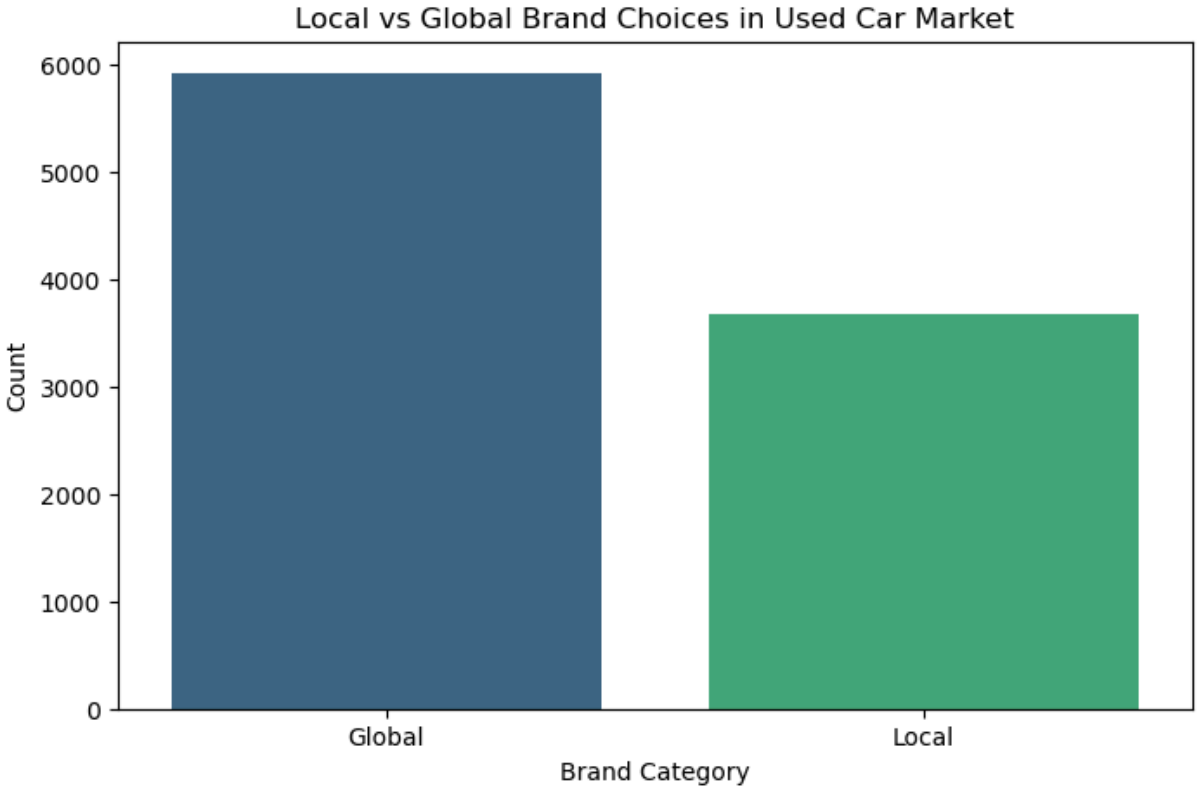
I have imported the required library which is matplotlib and seaborn to start with my visualizations.

In [18]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

4.6.2 Count of Local vs Global Brands

I created a bar chart to understand the distribution of brand preferences in the Indian used car market. This helps to identify how the market leans towards local brands or global brands, which is needed for analyzing how these preferences relate to global car production trends.

In [19]:
```
plt.figure(figsize=(8, 5))
sns.countplot(data=used_car_df, x='BrandCategory', palette='viridis')
plt.title('Local vs Global Brand Choices in Used Car Market')
plt.xlabel('Brand Category')
plt.ylabel('Count')
plt.show()
```



Insights:

- The count plot shows a clear dominance of global brands in the used car market in India, with over 5000 listings for global brands compared to around 3000 for local

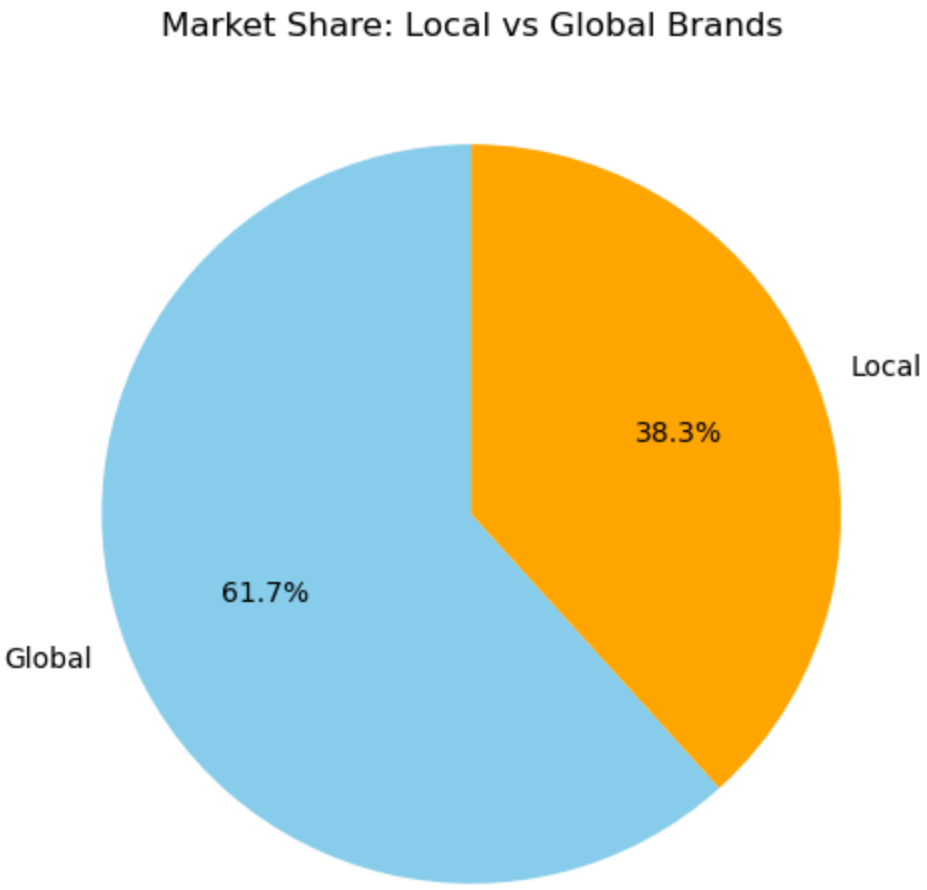
brands.

- This indicates that global brands are more prevalent in the used car market.

4.6.3 Market Share of Local vs Global Brands

I created pie chart to visually represent the proportion of the used car market by each brand category. This helps for a clear comparison of how dominant local brands are in India versus global brands, which directly ties into understanding consumer preferences and their alignment with global production trends.

In [20]:
```
brand_counts = used_car_df['BrandCategory'].value_counts()
plt.figure(figsize=(6, 6))
brand_counts.plot(kind='pie', autopct='%1.1f%%', colors=['skyblue', 'orange'], star
plt.title('Market Share: Local vs Global Brands')
plt.ylabel('')
plt.show()
```



Insights:

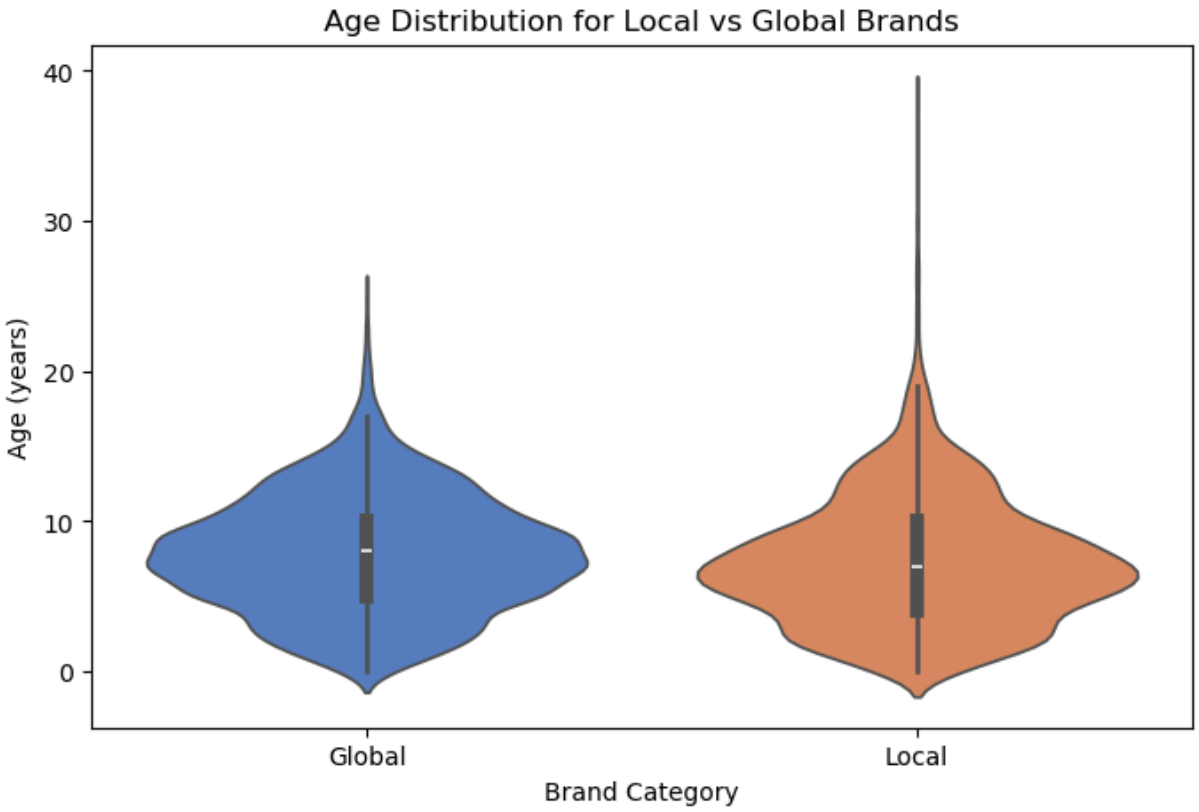
- The market share distribution shows that global brands has a larger portion of 61.7% of the used car listings, while local brands represent 38.3%.

- This suggests that consumers tend to lean more towards global brands when purchasing used cars, possibly due to perceived quality or brand reputation.

4.6.4 Distribution of car age by brand category

Violin plot was created to understand how the age of cars varies between local and global brands in the Indian used car market. This analysis helps explore whether local or global brands tend to have older or newer cars available, which is important for understanding consumer preferences and brand value retention over time.

In [21]:
```
plt.figure(figsize=(8, 5))
sns.violinplot(data=used_car_df, x='BrandCategory', y='Age', palette='muted')
plt.title('Age Distribution for Local vs Global Brands')
plt.xlabel('Brand Category')
plt.ylabel('Age (years)')
plt.show()
```



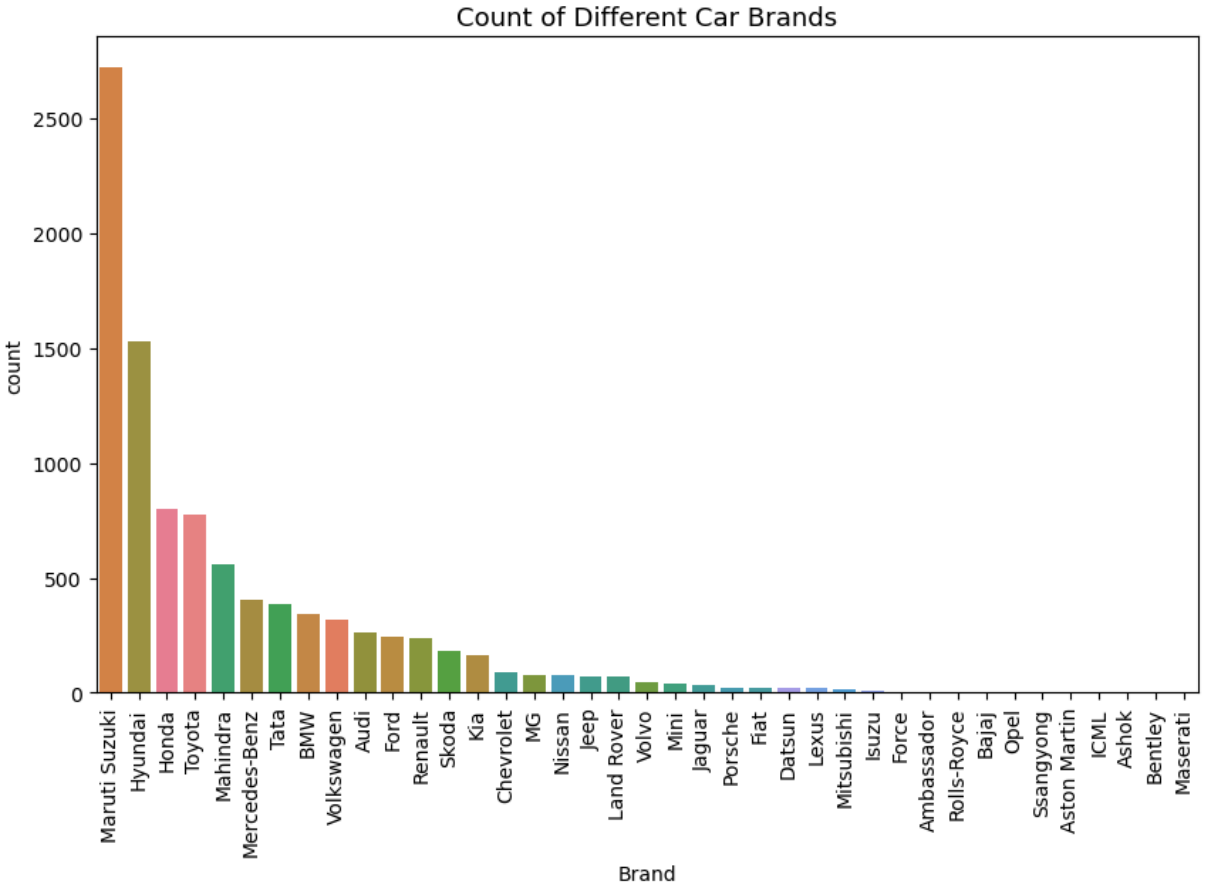
Insights:

- Local brand cars have a wider range in age, including some very old cars, as shown by the maximum age around 38 years.
- Global brand cars tend to be slightly newer on average, with most cars between 5 and 10 years old.
- Both local and global brands have a similar spread of ages, but the global brands show slightly fewer cars on the older end of the spectrum.

4.6.5 Count of Car Brands

The count of car brands was visualized to identify the most commonly listed car brands in the Indian used car market. This helps to understand the market dominance of specific brands and how they may influence consumer choices between local and global options, which is key to answering the research question about brand preferences.

In [22]:
```
plt.figure(figsize = (10,6))
brands = used_car_df['Brand'].value_counts().index
sns.countplot(x="Brand", data=used_car_df, order=brands, hue="Brand")
plt.title("Count of Different Car Brands", fontsize=13)
plt.xticks(rotation=90)
plt.show()
```



Insights:

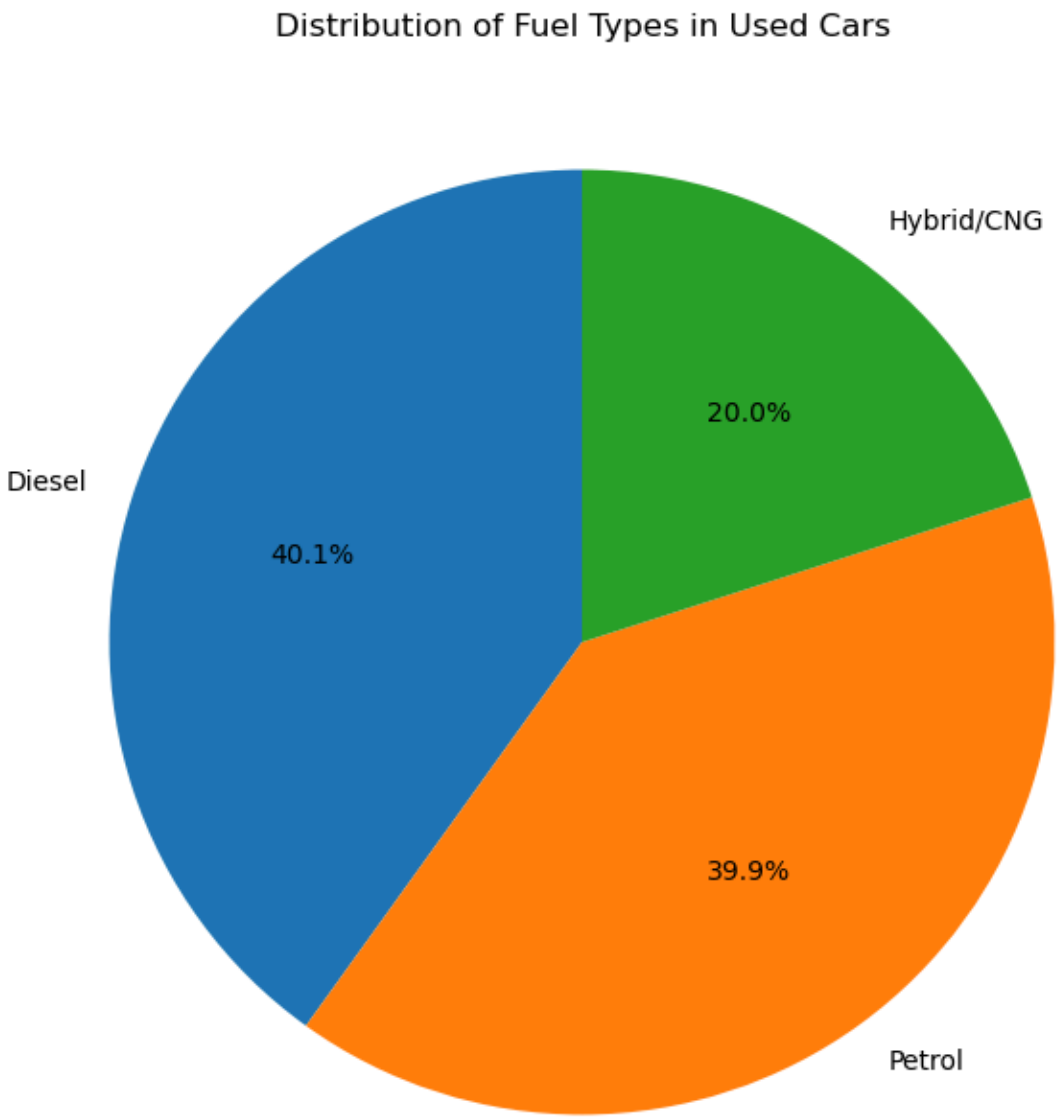
- Maruti Suzuki has the highest number of cars, with a total of 2720. This means Maruti Suzuki is very popular in the Indian used car market.
- Maserati has the lowest number of cars, showing it is a rare and less common brand.

4.6.6 Fuel Type

The pie chart was created to show the distribution of car fuel types in the used car market. This helps to understand the preferences for different fuel types, which can be important for

analyzing how fuel preferences might differ between local and global brands, and how they may influence consumer decisions in the market.

In [23]:
```
# Pie chart for Fuel Type distribution
fuel_type_counts = used_car_df['FuelType'].value_counts()
plt.figure(figsize=(8, 8))
fuel_type_counts.plot(kind='pie', autopct='%1.1f%%', startangle=90)
plt.title('Distribution of Fuel Types in Used Cars')
plt.ylabel('')
plt.show()
```



Insights:

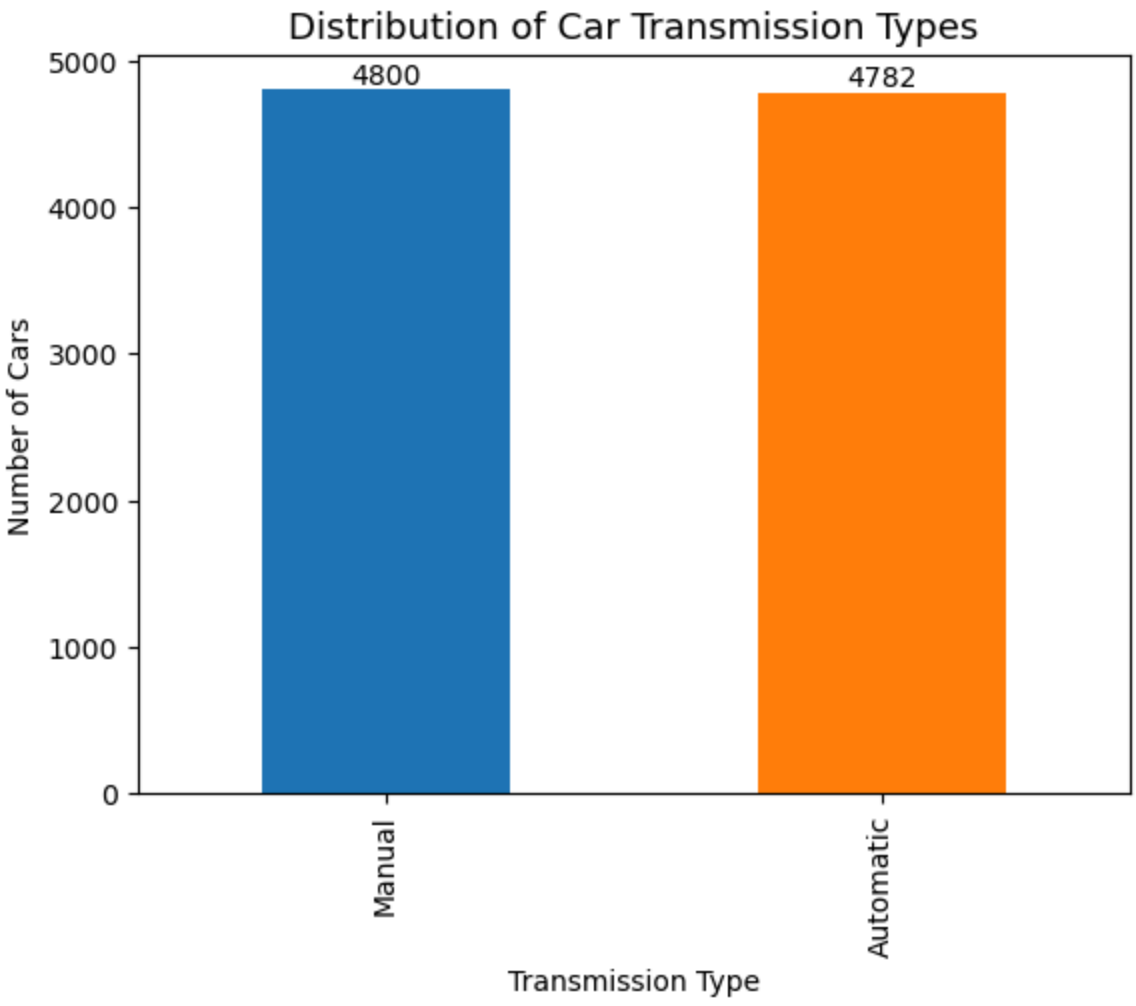
- Diesel cars make up 40.1% of the market, indicating a significant preference for diesel vehicles in the used car market.

- Petrol cars account for 39.9%, showing a nearly equal preference to diesel cars, with a slight difference.
- Hybrid cars make up 20%, reflecting a growing interest in alternative fuel options, but still a smaller segment compared to petrol and diesel.
- Diesel and petrol cars dominate the market, with almost equal shares, while hybrid cars still represent a niche but growing preference.

4.6.7 Distribution of car transmission types

The bar chart was created to analyze the preference for manual vs. automatic transmissions in the used car market. It provides insights into consumer preferences and how transmission type might influence the choice of brand in the Indian market.

In [24]:
```
transmission_count = used_car_df['Transmission'].value_counts()
bars = transmission_count.plot(kind='bar', color=['#1f77b4', '#ff7f0e'])
plt.title('Distribution of Car Transmission Types', fontsize=13)
plt.xlabel('Transmission Type')
plt.ylabel('Number of Cars')
bars.bar_label(bars.containers[0], fontsize=10)
plt.show()
```



Insights:

- There are 4,800 manual cars. Manual transmissions are often chosen for their lower cost and better fuel efficiency, though they are less common now.
- With 4,782 automatic cars, automatic transmissions are preferred for convenience, especially in city driving.
- The counts for both types are nearly equal, indicating that both manual and automatic transmissions are popular in the used car market. This suggests a balanced demand for both transmission types.

5 Dataset 2: Web Scraped Dataset (car production by country)

5.1 Import Libraries

I have imported the requests and BeautifulSoup libraries to fetch and parse the HTML content from web pages for web scraping.

In [25]:
```
import requests
from bs4 import BeautifulSoup
```

5.2 Web Scraping Dataset

I have used the requests.get method to fetch the HTML content of the Wikipedia page containing the list of countries by vehicle exports.

In [26]:
```
url = 'https://www.datapandas.org/ranking/car-production-by-country'
response = requests.get(url)
```

I have checked if the webpage was successfully fetched by verifying the response status code.

In [27]:
```
# To check if the request was successful
if response.status_code == 200:
    print("Successfully fetched the webpage.")
else:
    print("Failed to fetch the webpage.")
    exit()
```

```
Successfully fetched the webpage.
```

I parsed the webpage content using BeautifulSoup to extract the data.

In [28]:
```
# Parse the page content with BeautifulSoup
soup = BeautifulSoup(response.text, 'html.parser')
```

I found the table and its respective id containing the car production data on the webpage using BeautifulSoup.

In [29]:
```
# Find the table on the page
table = soup.find('table', {'id': 'full_data_table'})
```

I extracted the headers from the table to get the column names using BeautifulSoup.

In [30]:
```
# Extract the table rows
headers = []
for th in table.find_all('th'):
    headers.append(th.text.strip())
```

I extracted the table rows excluding the header to gather the data from each row using BeautifulSoup.

In [31]:
```
# Extract the table rows
rows = []
for tr in table.find_all('tr')[1:]:  # Skip the header row
    cols = tr.find_all('td')
    row = [col.text.strip() for col in cols]
    rows.append(row)
```

I converted the extracted table into a DataFrame and displayed the first few rows using df.head().

In [32]:
```
# Convert the table into a DataFrame
df = pd.DataFrame(rows, columns=headers)
df.head()
```

Out[32]:

	Region ↕	Car Production ↕
0	Argentina	536.9K
1	Austria	107.5K
2	Belgium	276.6K
3	Brazil	2.4M
4	Canada	1.2M

5.3 Relevancy of data and justified use of data source

5.3.1 Origin of Dataset

The dataset is a web-scraped table of global car production data, which includes production numbers for different countries. This dataset was gathered from open sources that track global car manufacturing trends and production statistics.

5.3.2 Why this data source is appropriate for the research question posed:

The global car production dataset provides insights into production trends in various countries, allowing for a comparison between Indian consumer preferences and global manufacturing patterns.

5.3.3 Clearly identifiable case for working with this specific type of data

The global car production dataset helps compare production trends across countries, offering context to Indian preferences.

5.3.4 Format of data is suitable for analysis

Even though the table of the car production by country is web scraped from a website, it is stored as a pandas dataframe to continue with the analysis. I have also exported the dataframe to a csv. If the web scraping method takes long time or is unable to load the table the exported csv file can be used for the analysis.

I have used the to_csv function to convert the dataframe to a csv file. I have did this to make sure my analysis continues even if my web scraping part is not working on other laptops.

In [33]:
```
# Save the DataFrame to a CSV file
df.to_csv('car_production_by_country.csv', index=False)
```

Uncomment this line to load the dataset if the web scrapping part is not working.

In [34]:
```
# Load the dataset
# df = pd.read_csv("car_production_by_country.csv")
```

I used df.info to display information about the DataFrame, showing that it contains 36 rows and 2 columns, both of which are of object data type.

In [35]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 2 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Region ↕          36 non-null     object
 1   Car Production ↕  36 non-null     object
dtypes: object(2)
memory usage: 708.0+ bytes
```

I have checked if there is any null values present in the dataset using the isnull function.

In [36]:
```
df.isnull().sum()
```

Out[36]:
```
Region ↕            0
Car Production ↕    0
dtype: int64
```

5.4 Data Cleaning

The thousands and millions are represented as K and M in the car production column. This should be converted to numeric for to continue with the analysis. I have created a function to execute the process of converting it to numeric.

In [37]:
```
# Function to convert car production values
def convert_values(value):
    if 'K' in value:
        return int(float(value.replace('K', '').strip()) * 1000)
    elif 'M' in value:
        return int(float(value.replace('M', '').strip()) * 1000000)
    else:
        return float(value.strip())

df['Car Production ↕'] = df['Car Production ↕'].apply(convert_values)
```

In [38]:
```
df.head()
```

Out[38]:

	Region ↕	Car Production ↕
0	Argentina	536900
1	Austria	107500
2	Belgium	276600
3	Brazil	2400000
4	Canada	1200000

Once, all the values have been converted to numeric, I have sorted the dataframe in desecending order and reset the index to find out the top most car production country.

In [39]:
```
df = df.sort_values(by='Car Production ↕', ascending=False)
df = df.reset_index(drop=True)
df.head()
```

Out[39]:

	Region ↕	Car Production ↕
0	China	27000000
1	United States	10100000
2	Japan	7800000
3	India	5500000
4	South Korea	3800000

I have used the info function to ensure the car production data type have changed correctly.

In [40]:
```
df.info()
```

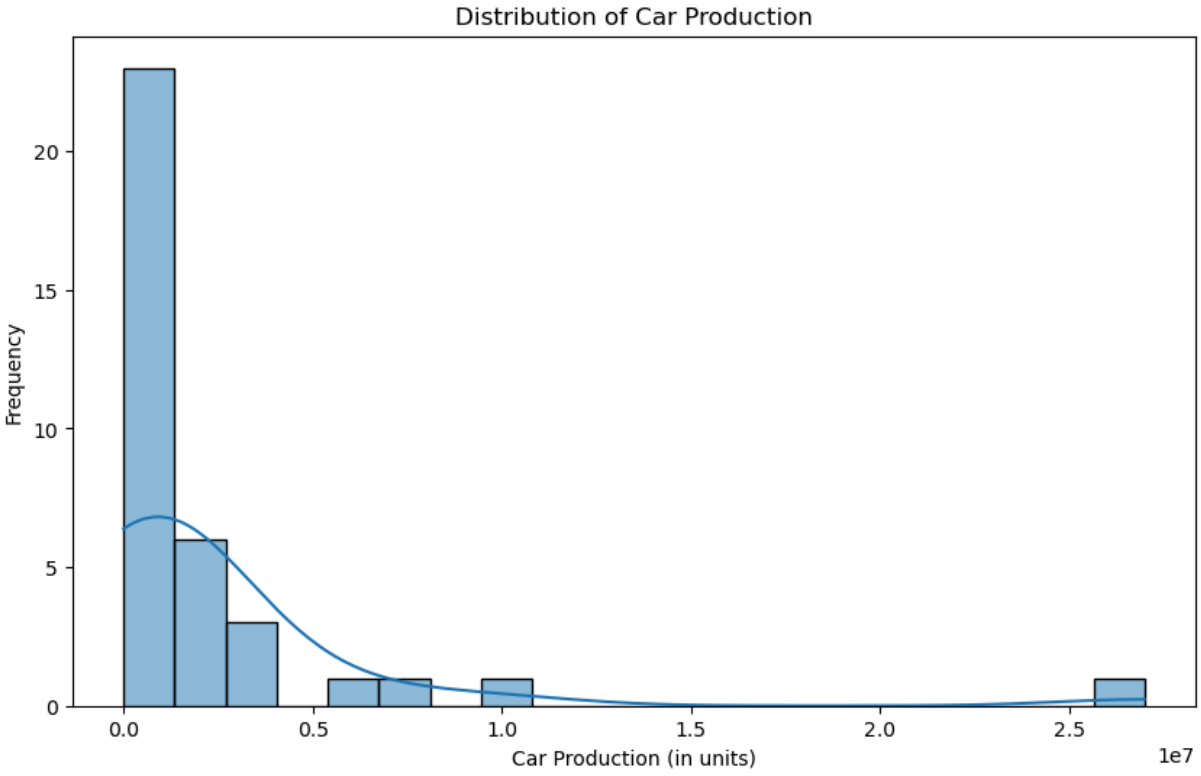
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 2 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Region ↕          36 non-null     object
 1   Car Production ↕  36 non-null     int64 
dtypes: int64(1), object(1)
memory usage: 708.0+ bytes
```

5.5 Data Exploration and Visualization

5.5.1 Distribution of Car Production

This histogram shows the distribution of car production values across all regions. It helps us understand the spread of production levels and identify any skewness in the data.

In [41]:
```
# Plotting the distribution of car production
plt.figure(figsize=(10, 6))
sns.histplot(df['Car Production ↕'], kde=True, bins=20)
plt.title('Distribution of Car Production')
plt.xlabel('Car Production (in units)')
plt.ylabel('Frequency')
plt.show()
```



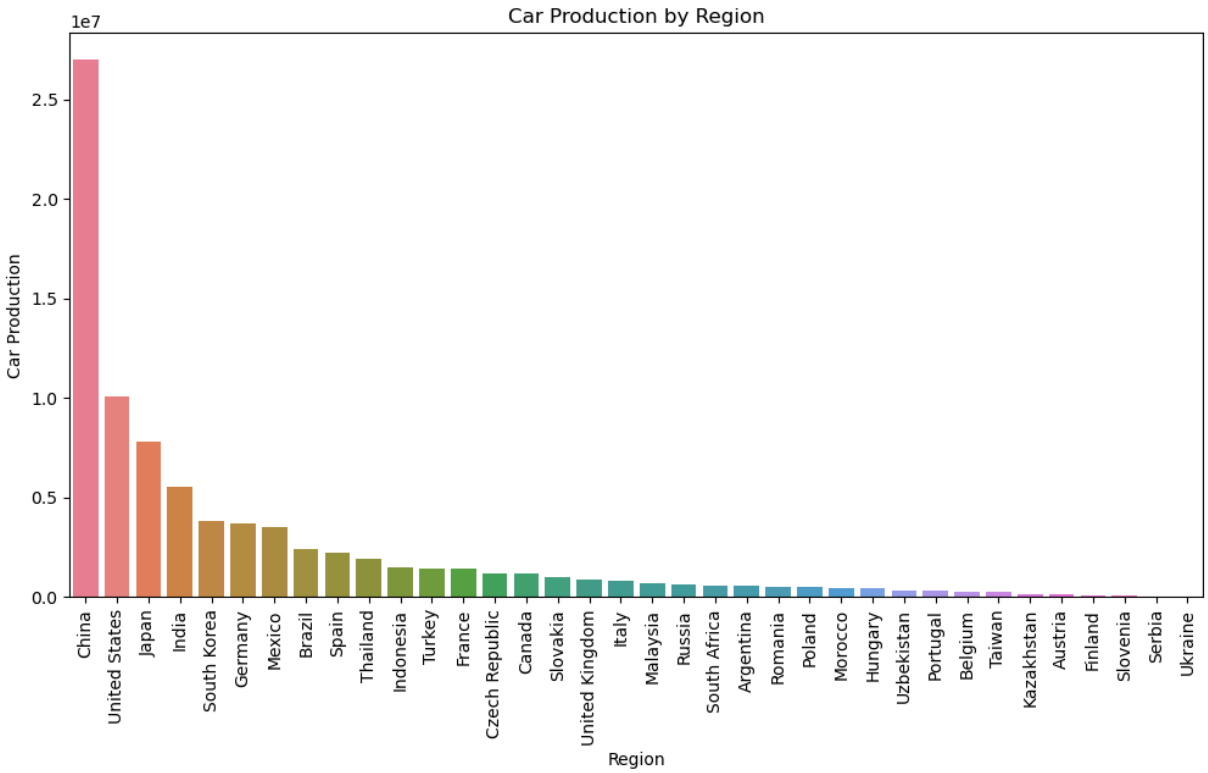
Insights:

- The histogram shows a right-skewed distribution, with most countries having lower car production values.
- Most regions produce fewer cars, with a large portion having production values below 1.5 million.
- The distribution exhibits exponential decay, where fewer regions have higher production levels.

5.5.2 Car Production by Region

Bar plot showing car production values across regions. This visualization helps identify the highest and lowest car production countries across the region.

In [42]:
```
plt.figure(figsize=(12, 6))
sns.barplot(x='Region ↕', y='Car Production ↕', data=df, palette='husl')
plt.title('Car Production by Region')
plt.xlabel('Region')
plt.ylabel('Car Production')
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
plt.show()
```



Insights:

- China, United States, and Japan are the top car producers.
- Ukraine is the lowest car producers among all the countries.
- India, South Korea, Germany, and Mexico are strong moderate producers, making 2-4 million cars.
- Many small countries produce fewer than 1 million cars.

6 Combined Dataset for Further Exploration

6.1 Combining dataset based on Brand & Country

I need to combine the dataset to explore further and also to get an answer for my research question. I am combining the dataset based on the brand and country. I am mapping it to the respective values. Firstly, I have find the unique values in the brand columns.

In [43]:
```
unique_brands = used_car_df['Brand'].unique()
unique_brands
```

Out[43]:
```
array(['Honda', 'Toyota', 'Volkswagen', 'Maruti Suzuki', 'BMW', 'Ford',
       'Kia', 'Mercedes-Benz', 'Hyundai', 'Audi', 'Renault', 'MG',
       'Volvo', 'Skoda', 'Tata', 'Mahindra', 'Mini', 'Land Rover', 'Jeep',
       'Chevrolet', 'Jaguar', 'Fiat', 'Aston Martin', 'Porsche', 'Nissan',
       'Force', 'Mitsubishi', 'Lexus', 'Isuzu', 'Datsun', 'Ambassador',
       'Rolls-Royce', 'ICML', 'Bajaj', 'Opel', 'Ashok', 'Bentley',
       'Ssangyong', 'Maserati'], dtype=object)
```

Next, I have created a dictionary to manually map the brand with the countries.

In [44]:
```
brand_to_country = {
    'Honda': 'Japan','Toyota': 'Japan','Volkswagen': 'Germany','Maruti Suzuki': 'In
    'Ford': 'USA','Kia': 'South Korea','Mercedes-Benz': 'Germany', 'Hyundai': 'Sout
    'Renault': 'France','MG': 'China','Volvo': 'Sweden','Skoda': 'Czech Republic','
    'Mini': 'Germany','Land Rover': 'UK','Jeep': 'USA','Chevrolet': 'USA','Jaguar':
    'Aston Martin': 'UK','Porsche': 'Germany', 'Nissan': 'Japan','Force': 'India','
    'Lexus': 'Japan','Isuzu': 'Japan','Datsun': 'Japan','Ambassador': 'India','Roll
    'Bajaj': 'India','Opel': 'Germany','Ashok': 'India','Bentley': 'UK','Ssangyong'
}
```

I have used the map function to map the country based on the brand column.

In [45]:
```
# Mapping the country and the brand
used_car_df['Region ↕'] = used_car_df['Brand'].map(brand_to_country)
used_car_df.head()
```

Out[45]:

	Brand	model	Year	Age	kmDriven	Transmission	Owner	FuelType	AskPrice
0	Honda	City	2001	23	98000.0	Manual	second	Petrol	195000.0
1	Toyota	Innova	2009	15	190000.0	Manual	second	Diesel	375000.0
2	Volkswagen	VentoTest	2010	14	77246.0	Manual	first	Diesel	184999.0
3	Maruti Suzuki	Swift	2017	7	83500.0	Manual	second	Diesel	565000.0
4	Maruti Suzuki	Baleno	2019	5	45000.0	Automatic	first	Petrol	685000.0

I need to create the car production column as well. So, I have created a map function of region to car production and map the value based on the region column. At the end, I need to add the region and car production columns and each region will have its respective car production value mapped.

In [46]:
```
# Mapping the region and car production
Region_to_car_production = dict(zip(df['Region ↕'], df['Car Production ↕']))
used_car_df['Car Production ↕'] = used_car_df['Region ↕'].map(Region_to_car_product
```

In [47]:
```
used_car_df.head()
```

Out[47]:

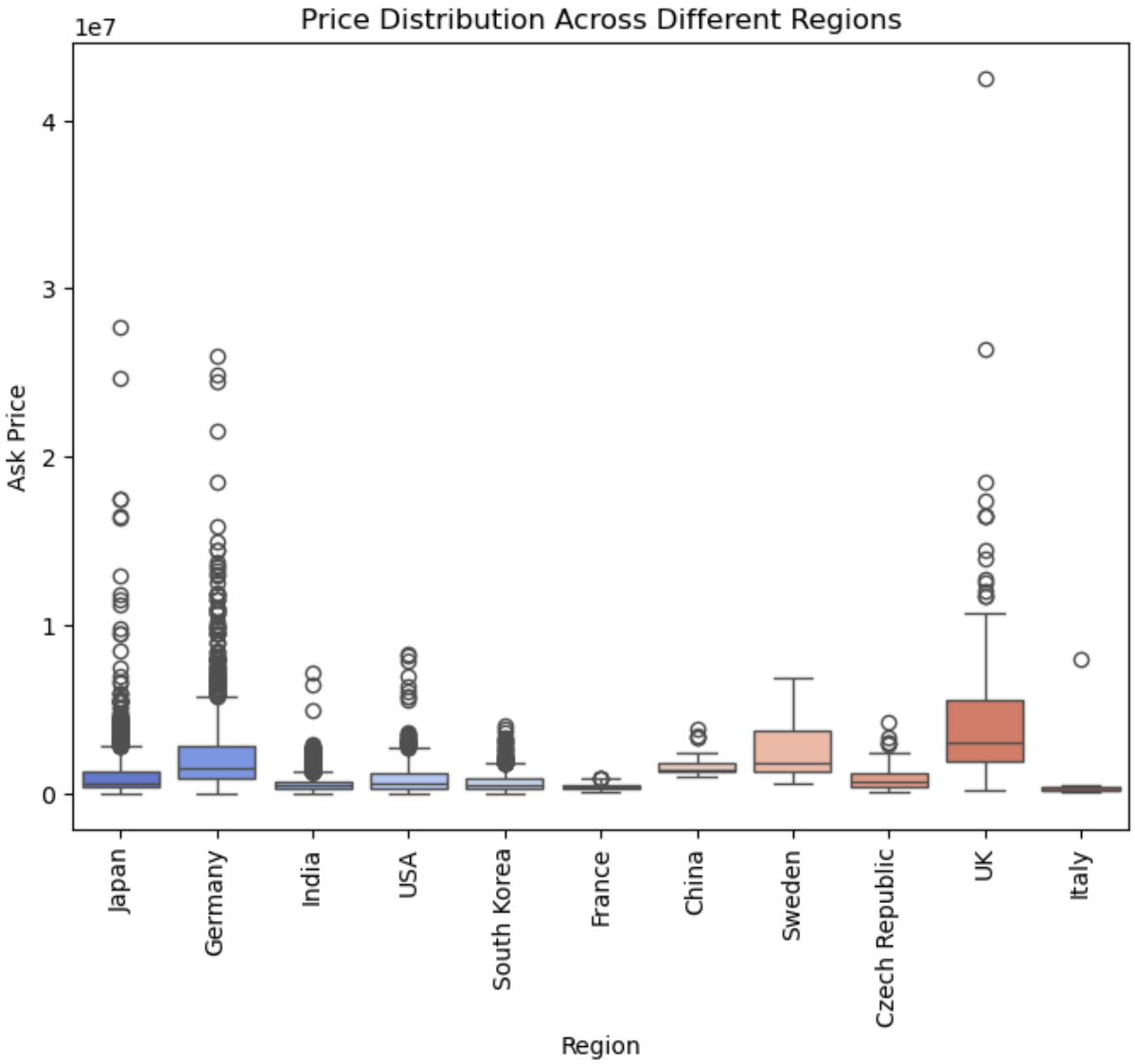
	Brand	model	Year	Age	kmDriven	Transmission	Owner	FuelType	AskPrice
0	Honda	City	2001	23	98000.0	Manual	second	Petrol	195000.0
1	Toyota	Innova	2009	15	190000.0	Manual	second	Diesel	375000.0
2	Volkswagen	VentoTest	2010	14	77246.0	Manual	first	Diesel	184999.0
3	Maruti Suzuki	Swift	2017	7	83500.0	Manual	second	Diesel	565000.0
4	Maruti Suzuki	Baleno	2019	5	45000.0	Automatic	first	Petrol	685000.0

6.2 Data Exploration and Visualization

6.2.1 Price Distribution for Different region

I created boxplot to show how car prices vary across different regions. It helps to compare local and global brands and see how car production in each region affects pricing, which shows about how regional production trends influence local versus global brand choices.

In [48]:
```
# Box plot for AskPrice by region
plt.figure(figsize=(8, 6))
sns.boxplot(data=used_car_df, x='Region ↕', y='AskPrice', palette='coolwarm')
plt.title('Price Distribution Across Different Regions')
plt.ylabel('Ask Price')
plt.xlabel('Region')
plt.xticks(rotation=90)
plt.show()
```



Insights:

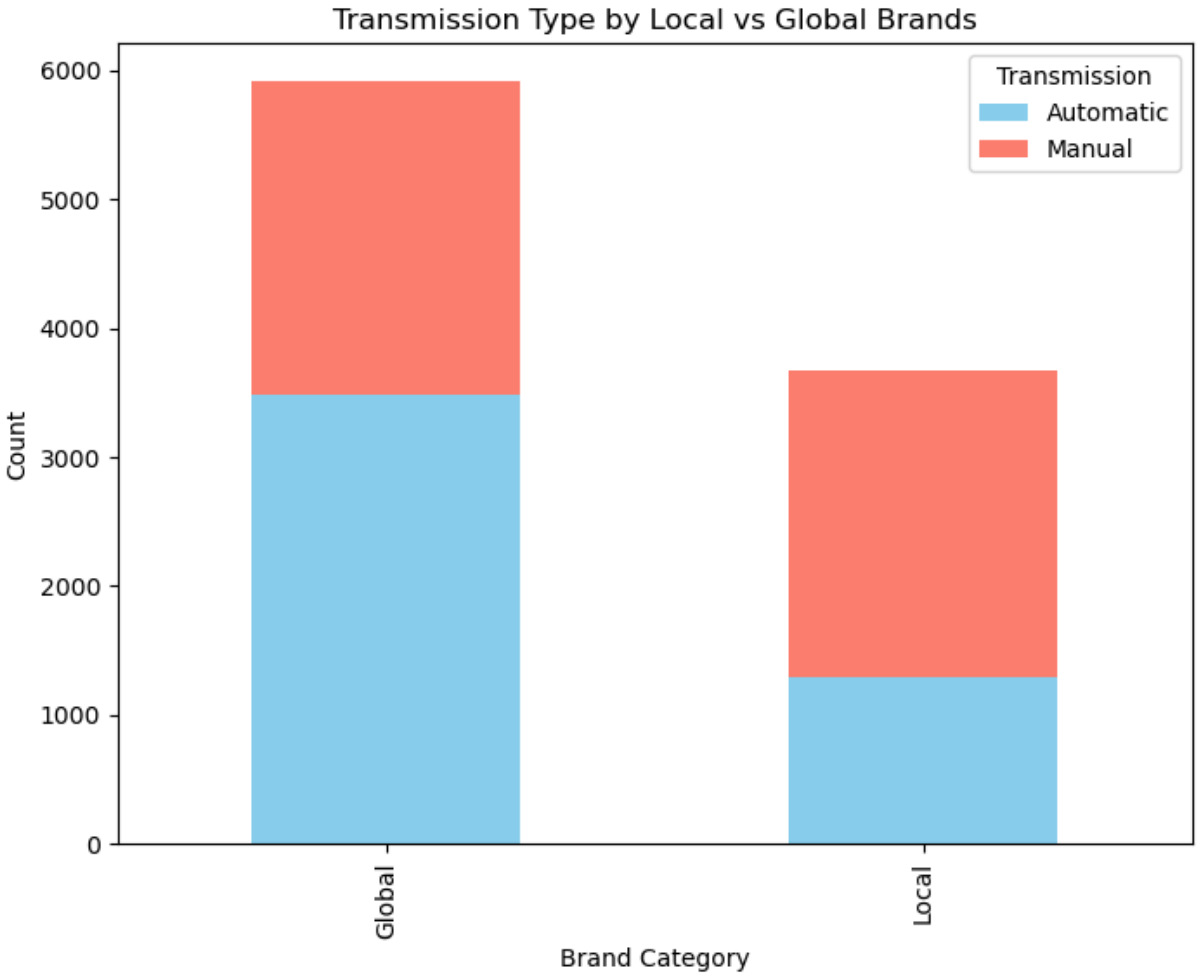
- There is a wide variation in car prices across regions, with some regions like Germany, Japan, and the UK showing extremely high-value cars, while others like France and south korea have more concentrated price distributions around the mid-range.
- Some regions, like Sweden and the UK, show a skew towards high-value cars, whereas countries like France and India have more affordable pricing in the lower quartiles.
- The distribution of prices also reveals outliers in various regions, indicating the presence of luxury and high-end vehicles alongside more standard options.

6.2.2 Stacked bar chart for transmission type by brand category

I created stacked bar chart to visualize the distribution of car transmissions across local and global brands. This chart helps to explore how the transmission types differ between local and global brands, providing insight into how these factors influence consumer preferences.

In [49]:
```
# Stacked bar chart for transmission type by brand category
transmission_count = used_car_df.groupby(['BrandCategory', 'Transmission']).size().
print(transmission_count)
transmission_count.plot(kind='bar', stacked=True, figsize=(8, 6), color=['skyblue',
plt.title('Transmission Type by Local vs Global Brands')
plt.ylabel('Count')
plt.xlabel('Brand Category')
plt.show()
```

```
Transmission    Automatic  Manual
BrandCategory
Global               3483    2430
Local                1299    2370
```



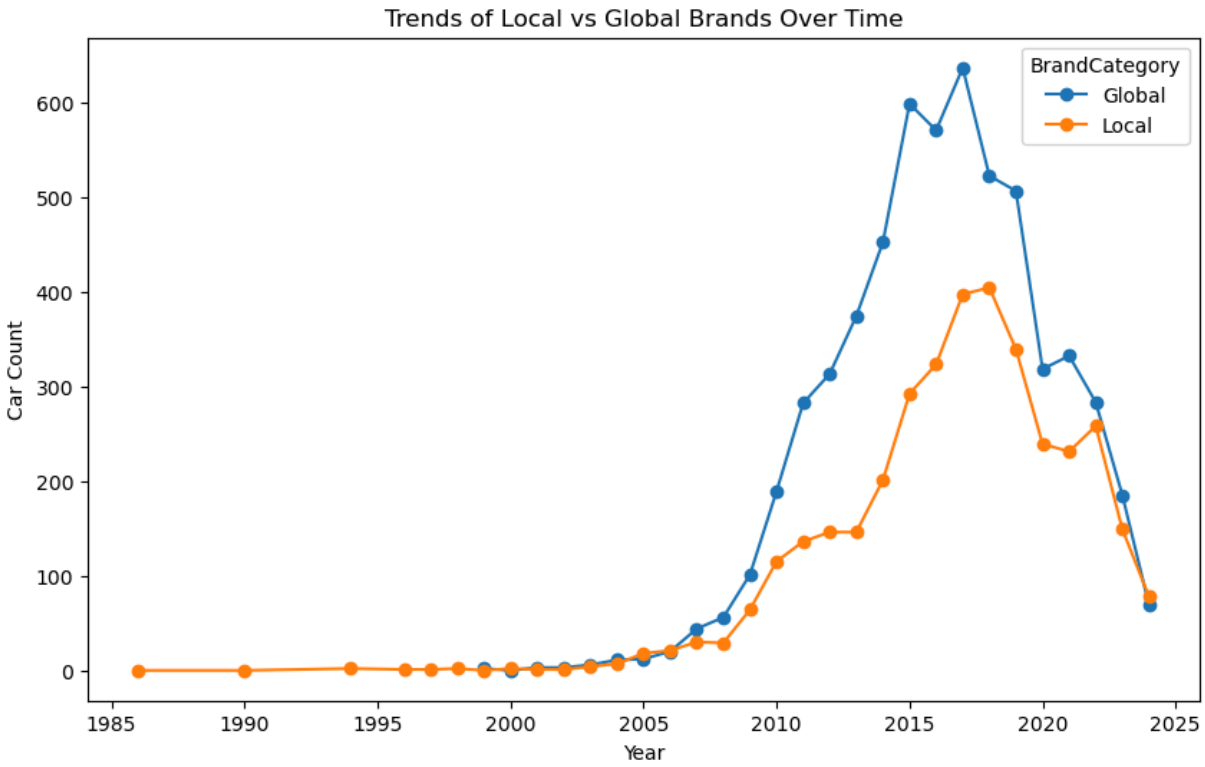
Insights:

- Global brands have a higher proportion of automatic transmissions compared to local brands, indicating a preference for convenience features in cars produced globally.
- Local brands, have a more balanced distribution between manual and automatic transmissions, with a slightly higher number of manual transmission cars.
- Global brands may cater to a different segment of the market that prefers automatic transmissions, while local brands offer more manual options, possibly due to cost considerations or different consumer preferences in India.

6.2.3 Line plot for local vs global brands over years

I created a line plot to track the number of local and global car brands over the years. This helps to see the trend of how local and global brands have changed in the Indian market.

In [50]:
```
# Line plot for local vs global brands over years
brand_year = used_car_df.groupby(['Year', 'BrandCategory']).size().unstack().reset_
brand_year.plot(x='Year', kind='line', stacked=False, figsize=(10, 6), marker='o')
plt.title('Trends of Local vs Global Brands Over Time')
plt.ylabel('Car Count')
plt.xlabel('Year')
plt.show()
```



Insights:

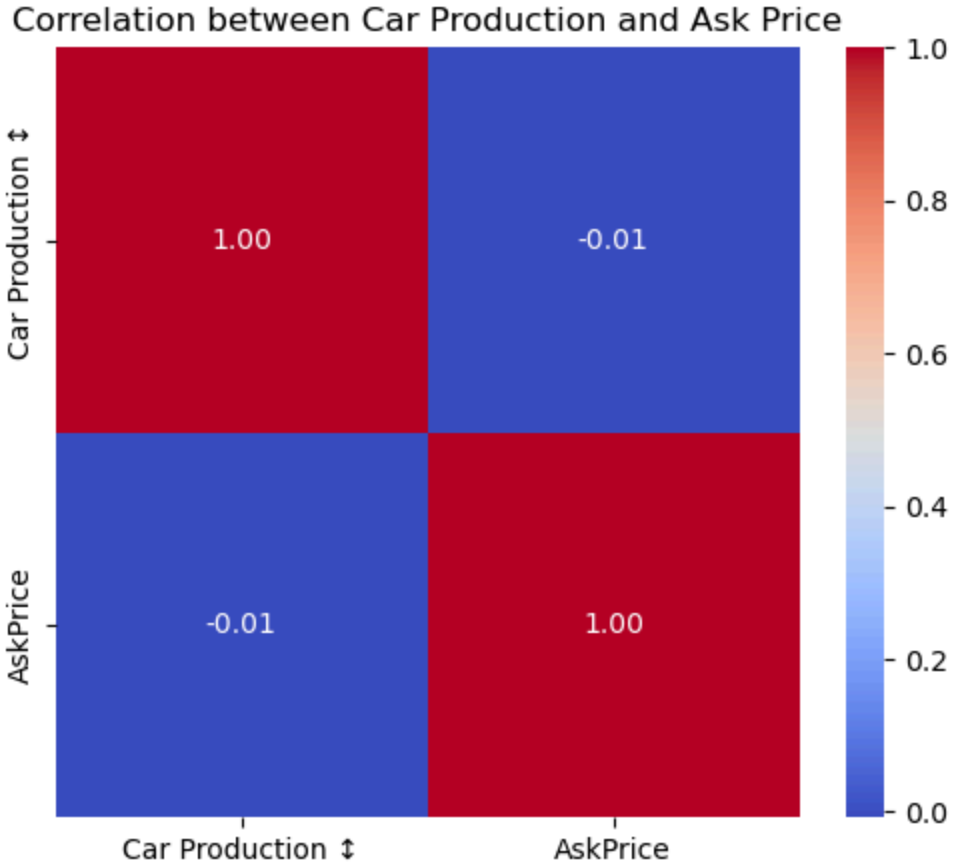
- Global brands saw a gradual increase in presence starting from 2000, peaking between 2013 and 2017.
- Local brands has a sharp rise in the number of vehicles from 2014 onwards.
- Both global and local brands experienced a decline in 2020 and 2021, likely due to the COVID-19 pandemic's impact.
- Local brands maintained a consistent and strong market presence, especially after 2015, despite fluctuations in global brand numbers.

6.2.4 Correlation between Car Production and AskPrice

I analyzed the correlation between car production and ask price to see if global car production trends influence the pricing of used cars in India. This helps to understand how

production volumes of local and global brands might affect used car prices in the Indian market.

In [51]:
```
corr = used_car_df[['Car Production ↕', 'AskPrice']].corr()
plt.figure(figsize=(6, 5))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation between Car Production and Ask Price')
plt.show()
```



Insights:

- The correlation value of -0.01 suggests there is a very weak negative correlation between car production and the ask price of cars. This means that as car production increases slightly, the ask price tends to decrease.
- Hence, there is no significant relationship between car production and the ask price, as indicated by the near-zero correlation. This means that the global production of cars does not directly influence the pricing of used cars in India in your dataset.
- The very weak negative correlation could tell that, at least in this dataset, the production of cars may not significantly impact the used car prices.

7 Conclusion

To conclude, in India's used car market, people tend to prefer local brands like Maruti Suzuki because they are more affordable, while global brands like Toyota and Volkswagen attract buyers looking for higher-end features. The global car production trends show that countries like Japan and Germany produce more cars, which makes these global brands more available in India. This shows how the choices of local and global brands in India are linked to global car production, with affordability and brand prestige influencing buyer decisions.

8 Saving the Libraries in requirements file

In [52]:
```
pip freeze > requirements.txt
```

```
Note: you may need to restart the kernel to use updated packages.
```

9 References

DataPandas, 2024, Car production by country. Available at: https://www.datapandas.org/ranking/car-production-by-country (Accessed: 9 December 2024)

Kumar, M., 2024, Used Car Dataset. Available at: https://www.kaggle.com/datasets/mohitkumar282/used-car-dataset (Accessed: 10 December 2024)

List of countries by vehicle exports, 2024. Available at: https://en.wikipedia.org/wiki/List_of_countries_by_vehicle_exports (Accessed: 15 December 2024)

Chen, X., Gu, S., Deng, X. and Huang, L., 2022. Used Car Prices in India: What about Future?. Available at: https://www.researchgate.net/publication/363370759_Used_Car_Prices_in_India_What_about_Fut (Accessed: 15 December 2024)

Pavlinek, P., 2022. Transition of the automotive industry towards electric vehicle production in the east European integrated periphery. https://www.researchgate.net/publication/365001279_Transition_of_the_automotive_industry_tov (Accessed: 15 December 2024)