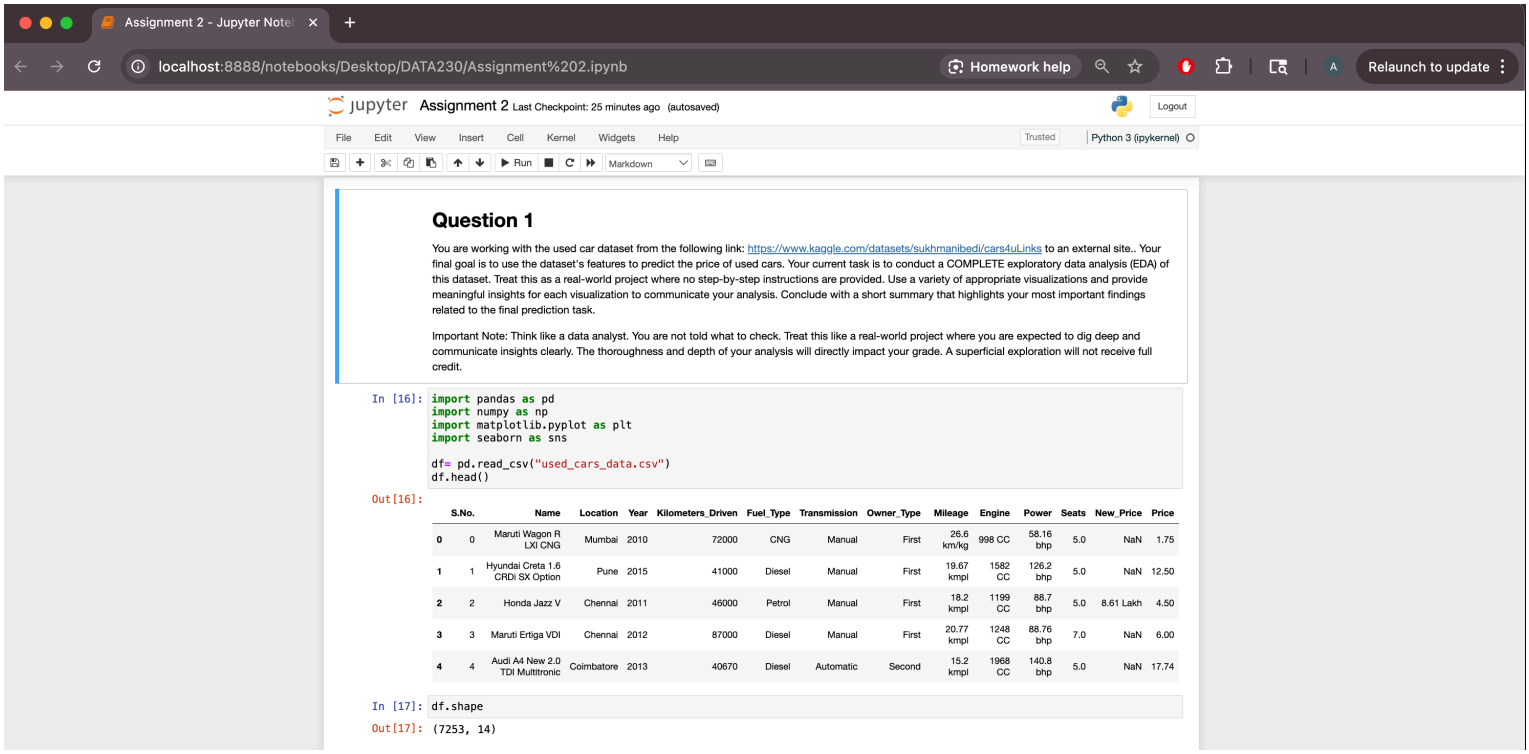
**Ameya Satish Khond** **SJSU ID:019136845**

DATA 230 - Assignment 2

1. You are working with the used car dataset from the following link: https://www.kaggle.com/datasets/sukhmanibedi/cars4uLinks to an external site.. Your final goal is to use the dataset's features to predict the price of used cars. Your current task is to conduct a COMPLETE exploratory data analysis (EDA) of this dataset. Treat this as a real-world project where no step-by-step instructions are provided. Use a variety of appropriate visualizations and provide meaningful insights for each visualization to communicate your analysis. Conclude with a short summary that highlights your most important findings related to the final prediction task.

Important Note: Think like a data analyst. You are not told what to check. Treat this like a real-world project where you are expected to dig deep and communicate insights clearly. The thoroughness and depth of your analysis will directly impact your grade. A superficial exploration will not receive full credit.



Assignment 2 - Jupyter Notebook

localhost:8888/notebooks/Desktop/DATA230/Assignment%202.ipynb

jupyter **Assignment 2** Last Checkpoint: 26 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Markdown

In [18]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7253 entries, 0 to 7252
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   S.No.              7253 non-null   int64  
 1   Name               7253 non-null   object 
 2   Location           7253 non-null   object 
 3   Year               7253 non-null   int64  
 4   Kilometers_Driven  7253 non-null   int64  
 5   Fuel_Type          7253 non-null   object 
 6   Transmission       7253 non-null   object 
 7   Owner_Type         7253 non-null   object 
 8   Mileage            7251 non-null   object 
 9   Engine             7207 non-null   object 
 10  Power              7207 non-null   object 
 11  Seats              7200 non-null   float64
 12  New_Price          1006 non-null   object 
 13  Price              6019 non-null   float64
dtypes: float64(2), int64(3), object(9)
memory usage: 793.4+ KB
```

In [19]:
```
# Feature Engineering

# 1. Car_Age calculation

current_year = 2024
df['Car_Age'] = current_year - df['Year']

# 2. Brand extraction using the Pandas .str
df['Brand'] = df['Name'].str.split(' ').str[0]
```

In [20]:
```
df.drop(columns=['S.No.', 'Name', 'Year', 'New_Price'], inplace=True)
```

In [21]:
```
cols = ['Price', 'Kilometers_Driven', 'Mileage', 'Engine', 'Power', 'Seats']
print(f"\nMissing Value Count After Drop:")
print(df[cols].isnull().sum())
```

```
Missing Value Count After Drop:
Price                1234
Kilometers_Driven       0
Mileage                 2
Engine                 46
Power                  46
Seats                  53
dtype: int64
```

Assignment 2 - Jupyter Notebook

localhost:8888/notebooks/Desktop/DATA230/Assignment%202.ipynb

jupyter **Assignment 2** Last Checkpoint: 27 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Markdown

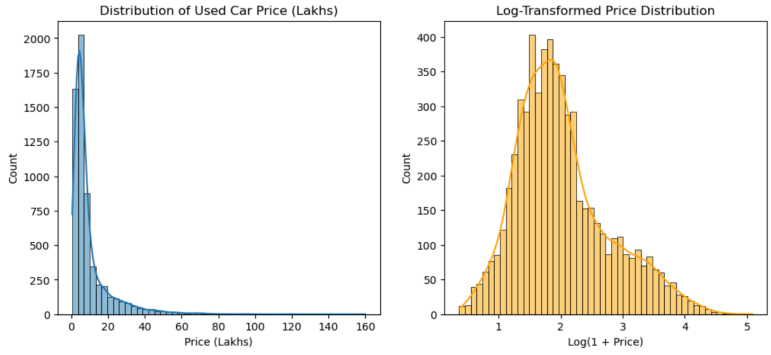
In [22]:
```
# Univariate Analysis (Target Variable)

plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.histplot(df['Price'], kde=True, bins=50)
plt.title('Distribution of Used Car Price (Lakhs)')
plt.xlabel('Price (Lakhs)')

# Log transformation for better visualization of skewed data

plt.subplot(1, 2, 2)
sns.histplot(np.log1p(df['Price']), kde=True, bins=50, color='orange')
plt.title('Log-Transformed Price Distribution')
plt.xlabel('Log(1 + Price)')
```

Out[22]:
```
Text(0.5, 0, 'Log(1 + Price)')
```



localhost:8888/notebooks/Desktop/DATA230/Assignment%202.ipynb

jupyter Assignment 2 Last Checkpoint: 42 minutes ago (autosaved)

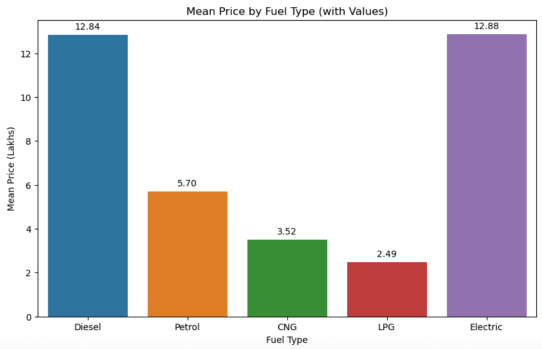
File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3 (ipykernel)

In [75]:
```
# Price Distribution by Fuel Type

fuel_order = ['Diesel', 'Petrol', 'CNG', 'LPG', 'Electric']
plt.figure(figsize=(10, 6))
ax = sns.barplot(
    x='Fuel_Type',
    y='Price',
    data=df,
    order=fuel_order,
    errorbar=None
)
plt.title('Mean Price by Fuel Type (with Values)')
plt.xlabel('Fuel Type')
plt.ylabel('Mean Price (Lakhs)')

for p in ax.patches:
    ax.annotate(
        format(p.get_height(), '.2f'),
        (p.get_x() + p.get_width() / 2., p.get_height()),
        ha = 'center',
        va = 'center',
        xytext = (0, 9),
        textcoords = 'offset points'
    )
```




Assignment 2 - Jupyter Notel

localhost:8888/notebooks/Desktop/DATA230/Assignment%202.ipynb

jupyter Assignment 2 Last Checkpoint: 28 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

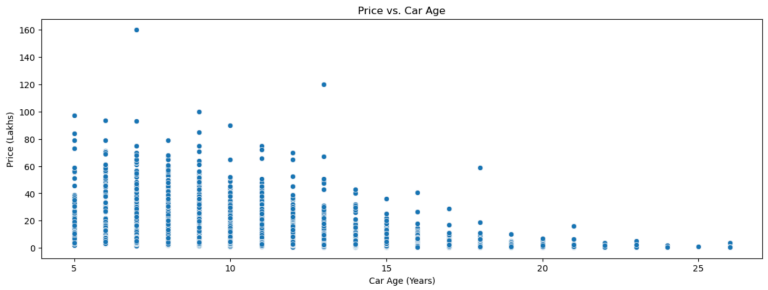
Trusted Python 3 (ipykernel)

In [23]:
```
# Bivariate Analysis (Categorical Features vs. Price)

# A. Car Age vs. Price

plt.figure(figsize=(15, 5))
sns.scatterplot(x='Car_Age', y='Price', data=df)
plt.title('Price vs. Car Age')
plt.xlabel('Car Age (Years)')
plt.ylabel('Price (Lakhs)')
```

Out[23]: Text(0, 0.5, 'Price (Lakhs)')



Assignment 2 - Jupyter Notebook

localhost:8888/notebooks/Desktop/DATA230/Assignment%202.ipynb

jupyter **Assignment 2** Last Checkpoint: 28 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

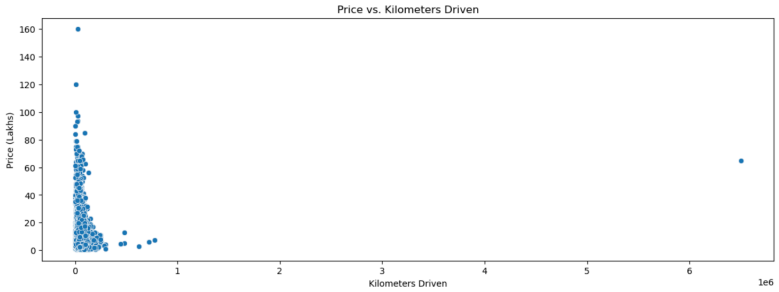
Trusted | Python 3 (ipykernel)

Run | Markdown

In [24]:
```
# B. Kilometers_Driven vs. Price

plt.figure(figsize=(15, 5))
sns.scatterplot(x='Kilometers_Driven', y='Price', data=df)
plt.title('Price vs. Kilometers Driven')
plt.xlabel('Kilometers Driven')
plt.ylabel('Price (Lakhs)')
```

Out[24]: Text(0, 0.5, 'Price (Lakhs)')




localhost:8888/notebooks/Desktop/DATA230/Assignment%202.ipynb

jupyter **Assignment 2** Last Checkpoint: 44 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

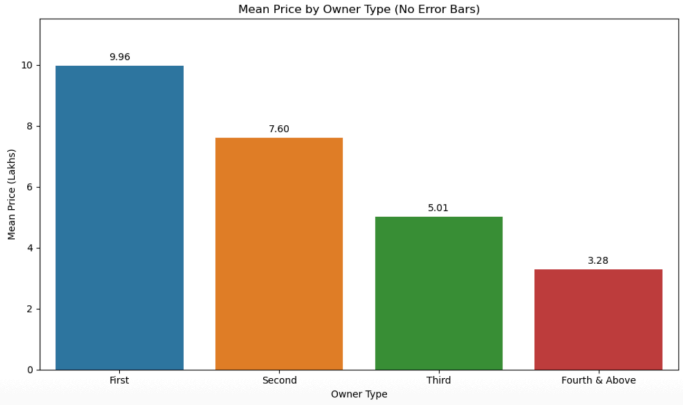
Trusted | Python 3 (ipykernel)

Run | Code

In [78]:
```
# C. Price vs. Owner_Type

plt.figure(figsize=(10, 6))
ax = sns.barplot(
    x='Owner_Type',
    y='Price',
    data=df,
    order=['First', 'Second', 'Third', 'Fourth & Above'],
    errorbar=None
)
plt.title('Mean Price by Owner Type (No Error Bars)')
plt.xlabel('Owner Type')
plt.ylabel('Mean Price (Lakhs)')
plt.ylim(0, ax.get_ylim()[1] * 1.1)
plt.tight_layout()

for p in ax.patches:
    ax.annotate(
        format(p.get_height(), '.2f'),
        (p.get_x() + p.get_width() / 2., p.get_height()),
        ha = 'center',
        va = 'center',
        xytext = (0, 9),
        textcoords = 'offset points'
    )
```



Assignment 2 - Jupyter Notebook

localhost:8888/notebooks/Desktop/DATA230/Assignment%202.ipynb

jupyter **Assignment 2** Last Checkpoint: 29 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

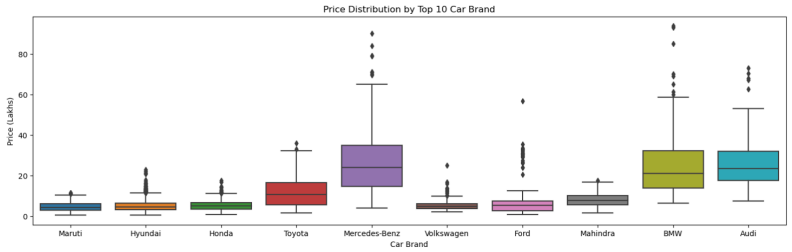
Trusted | Python 3 (ipykernel)

Run | Markdown

In [25]:
```
# C. Categorical Features vs. Price (Top 10 Brands)

top_brands = df['Brand'].value_counts().nlargest(10).index
plt.figure(figsize=(18, 5))
sns.boxplot(x='Brand', y='Price', data=df[df['Brand'].isin(top_brands)], order=top_brands)
plt.title('Price Distribution by Top 10 Car Brand')
plt.xlabel('Car Brand')
plt.ylabel('Price (Lakhs)')
```

Out[25]: Text(0, 0.5, 'Price (Lakhs)')




localhost:8888/notebooks/Desktop/DATA230/Assignment%202.ipynb

jupyter **Assignment 2** Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

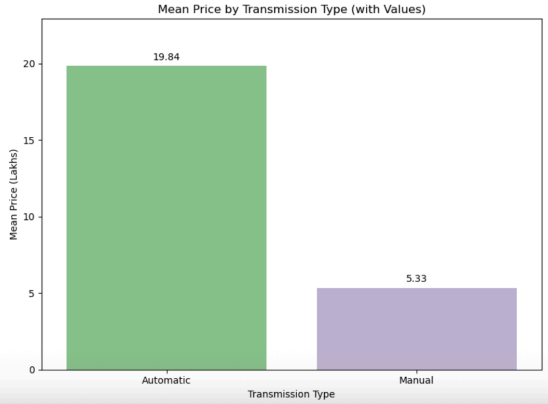
Run | Code

In [80]:
```
# E. Create the Bar Plot (Mean Price vs. Transmission Type)

plt.figure(figsize=(8, 6))
ax = sns.barplot(
    x='Transmission',
    y='Price',
    data=df,
    order=['Automatic', 'Manual'],
    palette='Accent',
    errorbar=None
)

plt.title('Mean Price by Transmission Type (with Values)')
plt.xlabel('Transmission Type')
plt.ylabel('Mean Price (Lakhs)')
plt.ylim(0, ax.get_ylim()[1] * 1.1)
plt.tight_layout()

for p in ax.patches:
    ax.annotate(
        format(p.get_height(), '.2f'),
        (p.get_x() + p.get_width() / 2., p.get_height()),
        ha = 'center',
        va = 'center',
        xytext = (0, 9),
        textcoords = 'offset points'
    )
```



localhost:8888/notebooks/Desktop/DATA230/Assignment%202.ipynb

jupyter Assignment 2 Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3 (ipykernel)

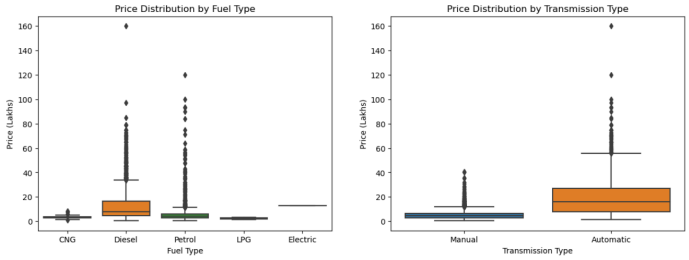
Run Code

In [81]:
```
#  F. Fuel Type and Transmission vs. Price

plt.figure(figsize=(15, 5))
plt.subplot(1, 2, 1)
sns.boxplot(x='Fuel_Type', y='Price', data=df)
plt.title('Price Distribution by Fuel Type')
plt.xlabel('Fuel Type')
plt.ylabel('Price (Lakhs)')


plt.subplot(1, 2, 2)
sns.boxplot(x='Transmission', y='Price', data=df)
plt.title('Price Distribution by Transmission Type')
plt.xlabel('Transmission Type')
plt.ylabel('Price (Lakhs)')
```

Out[81]: Text(0, 0.5, 'Price (Lakhs)')




localhost:8888/notebooks/Desktop/DATA230/Assignment%202.ipynb

jupyter Assignment 2 Last Checkpoint: an hour ago (autosaved)

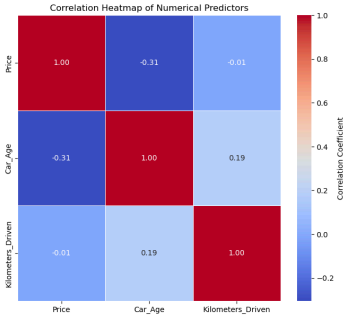
In [82]:
```
numerical_cols = ['Price', 'Car_Age', 'Kilometers_Driven']
df_corr = df[numerical_cols]

corr_matrix = df_corr.corr()

plt.figure(figsize=(8, 7))
sns.heatmap(
    corr_matrix,
    annot=True,
    cmap='coolwarm',
    fmt=".2f",
    linewidths=.5,
    cbar_kws={'label': 'Correlation Coefficient'}
)

plt.title('Correlation Heatmap of Numerical Predictors')
```

Out[82]: Text(0.5, 1.0, 'Correlation Heatmap of Numerical Predictors')




## Summary:

1. The Price distribution is heavily right-skewed (as seen in the raw histogram).Log-transformation of Price is used for better visualization of skewed data.
2. Brand is a highly effective categorical feature, clearly separating luxury brands (e.g., BMW, Audi) from economy brands (e.g., Maruti, Hyundai).
3. Transmission (Automatic vs. Manual) also shows a significant difference in price distribution, making it an important categorical factor.
4. An extreme outlier exists in the (Kilometers Driven) column and must be removed or capped before model training to prevent inaccurate predictions.
5. Diesel cars show a noticeably higher median and mean price distribution compared to Petrol cars, making this a valuable predictor.
6. Automatic cars (approx. 20 Lakhs average) are significantly more expensive than Manual cars (approx. 5.5 Lakhs average).

**Summary:**

1. The Price distribution is heavily right-skewed (as seen in the raw histogram).Log-transformation of Price is used for better visualization of skewed data.
2. Brand is a highly effective categorical feature, clearly separating luxury brands (e.g., BMW, Audi) from economy brands (e.g., Maruti, Hyundai).
3. Transmission (Automatic vs. Manual) also shows a significant difference in price distribution, making it an important categorical factor.
4. An extreme outlier exists in the (Kilometers Driven) column and must be removed or capped before model training to prevent inaccurate predictions.
5. Diesel cars show a noticeably higher median and mean price distribution compared to Petrol cars, making this a valuable predictor.
6. Automatic cars (approx. 20 Lakhs average) are significantly more expensive than Manual cars (approx. 5.5  Lakhs average).

2. You are working with population dataset (population.csv) from the following link: https://github.com/datasets/population/blob/main/data/population.csvLinks to an external site.. Your task is to use Plotly to create one visualization that represents the entire dataset clearly and completely. Your grade will depend on how effectively your visualization communicates the full information contained in the data.

**Question 2**

You are working with population dataset (population.csv) from the following link:
https://github.com/datasets/population/blob/main/data/population.csvLinks to an external site.. Your task is to use Plotly to create one visualization that represents the entire dataset clearly and completely. Your grade will depend on how effectively your visualization communicates the full information contained in the data.

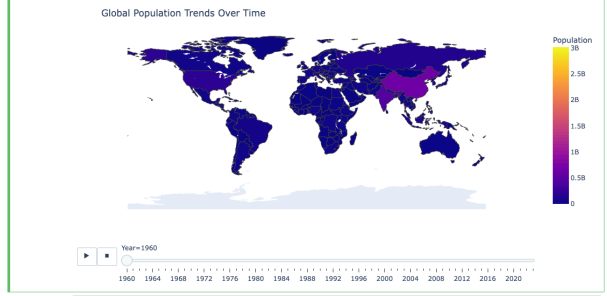
In [30]:
```
import pandas as pd
import plotly.express as px

df = pd.read_csv('population.csv')

df.rename(columns={
    'Country Name': 'Country',
    'Country Code': 'Code',
    'Year': 'Year',
    'Value': 'Population'
}, inplace=True)

fig = px.choropleth(
    df,
    locations="Code",
    color="Population",
    hover_name="Country",
    animation_frame="Year",
    color_continuous_scale="Plasma",
    title="Global Population Trends Over Time"
)

fig.update_layout(
    geo=dict(showframe=False, showcoastlines=False),
    margin=dict(l=0, r=0, t=50, b=0)
)
fig.show()
```




In [ ]: