20020015_is4116

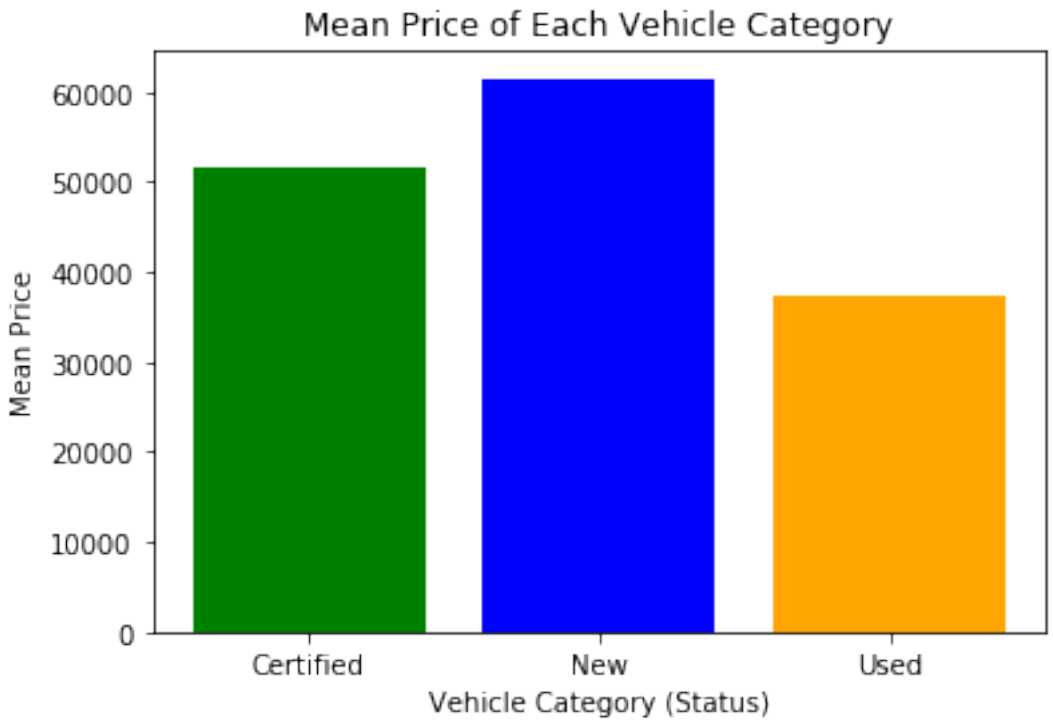
February 17, 2025

In [1]:
```
# Loading data
import pandas as pd
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
df.head()
```

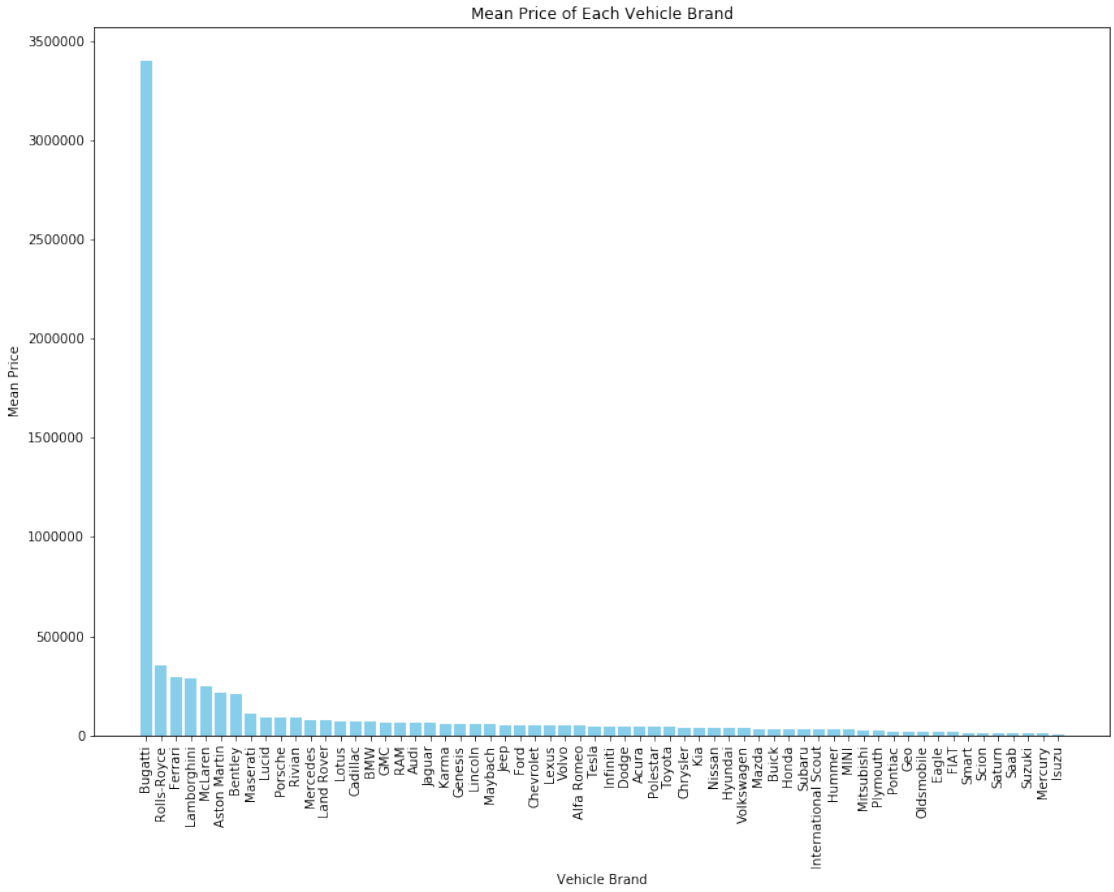
Out[1]:

	Brand	Model	Year	Status	Mileage	Dealer	Price
0	Hyundai	Ioniq	2024	New	0	Schomp Hyundai	1
1	Pontiac	Sunfire	2002	Used	153667	Streamline Auto Outlet	1500
2	Ford	F-150	2000	Used	217000	Chevrolet of Mandan	1699
3	Pontiac	G6	2009	Used	193401	Shea Buick GMC	1795
4	Chevrolet	HHR	2007	Used	201450	Dan Cummins Chrysler Dodge Jeep RAM of Paris	1900

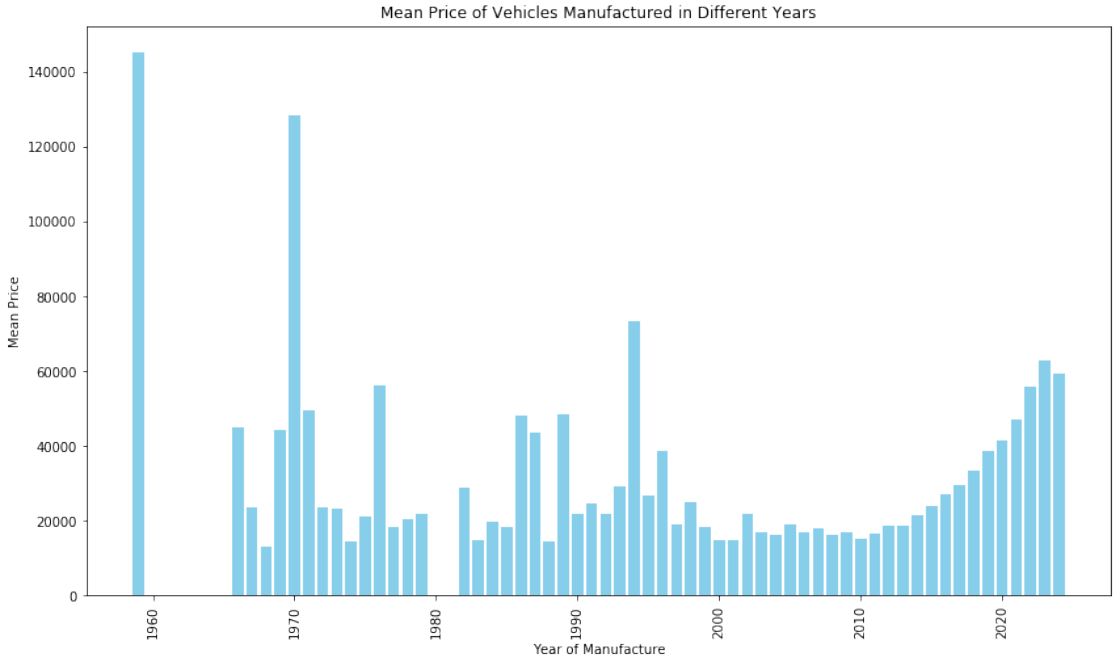
In [5]:
```
#Analyisis 01
import matplotlib.pyplot as plt
# Plotting with custom colors using matplotlib
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
mean_prices = df.groupby('Status')['Price'].mean()
# Bar plot
plt.bar(mean_prices.index, mean_prices, color=['green', 'blue', 'orange'])
# Adding titles and labels
plt.title('Mean Price of Each Vehicle Category')
plt.xlabel('Vehicle Category (Status)')
plt.ylabel('Mean Price')
# Show the plot
plt.show()
```



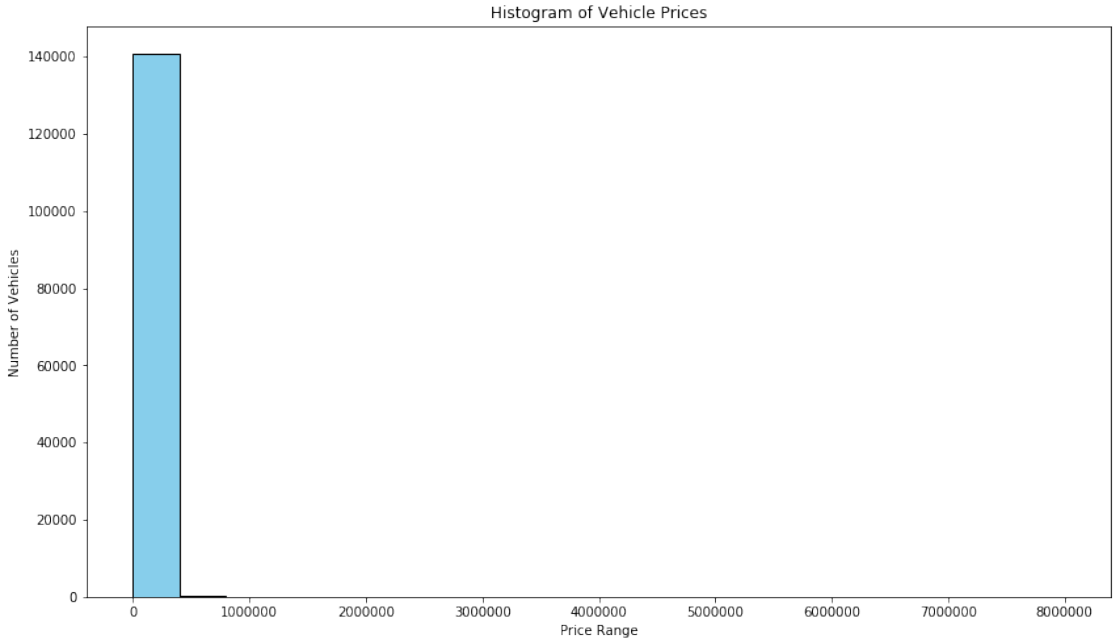
In [6]:
```
#Analyisis 02
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt  # Added import for matplotlib
# Load the dataset from a CSV file
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Grouping by 'Brand' and calculating the mean price for each brand
mean_brand_prices = df.groupby('Brand')['Price'].mean()
# Sort the mean prices in descending order
mean_brand_prices_sorted = mean_brand_prices.sort_values(ascending=False)
# Creating a larger figure for better readability
plt.figure(figsize=(14, 10))  # Increase width to 14 and height to 10
# Bar plot with custom colors for each brand
plt.bar(mean_brand_prices_sorted.index, mean_brand_prices_sorted, color='skyblue')
# Adding titles and labels
plt.title('Mean Price of Each Vehicle Brand')
plt.xlabel('Vehicle Brand')
plt.ylabel('Mean Price')
# Rotate x-axis labels for better readability if there are many brands
plt.xticks(rotation=90)
# Show the plot
plt.show()
```



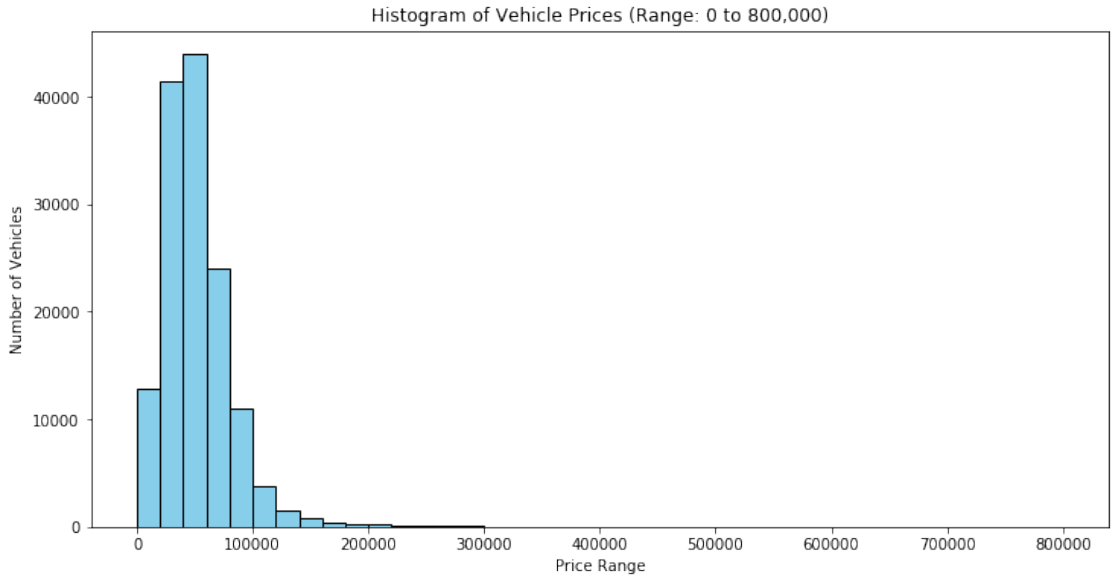

```
In [7]: #Analyisis 03
        import pandas as pd
        import matplotlib.pyplot as plt  # Added import for matplotlib
        # Load the dataset from a CSV file
        df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
        # Grouping by 'Year' and calculating the mean price
        mean_year_prices = df.groupby('Year')['Price'].mean()
        # Creating a larger figure for better readability
        plt.figure(figsize=(14, 8))  # Increase width to 12 and height to 6 (adjust as needed)
        # Bar plot with custom colors for each year
        plt.bar(mean_year_prices.index, mean_year_prices, color='skyblue')
        # Adding titles and labels
        plt.title('Mean Price of Vehicles Manufactured in Different Years')
        plt.xlabel('Year of Manufacture')
        plt.ylabel('Mean Price')
        # Rotate x-axis labels for better readability if there are many years
        plt.xticks(rotation=90)
        # Show the plot
        plt.show()
```




```
In [9]: #Analyisis 04_part_01
        import matplotlib.pyplot as plt
        import pandas as pd
        df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
        # Defining price bins, from 0 to 8078160 with intervals of 400000
        price_bins = range(0, 8078160, 400000)
        # Creating the histogram of 'Price' with the specified bins
        plt.figure(figsize=(14, 8))  # Increase figure size for readability
        plt.hist(df['Price'], bins=price_bins, color='skyblue', edgecolor='black')
        # Adding titles and labels
        plt.title('Histogram of Vehicle Prices')
        plt.xlabel('Price Range')
        plt.ylabel('Number of Vehicles')
        # Show the plot
        plt.show()
```

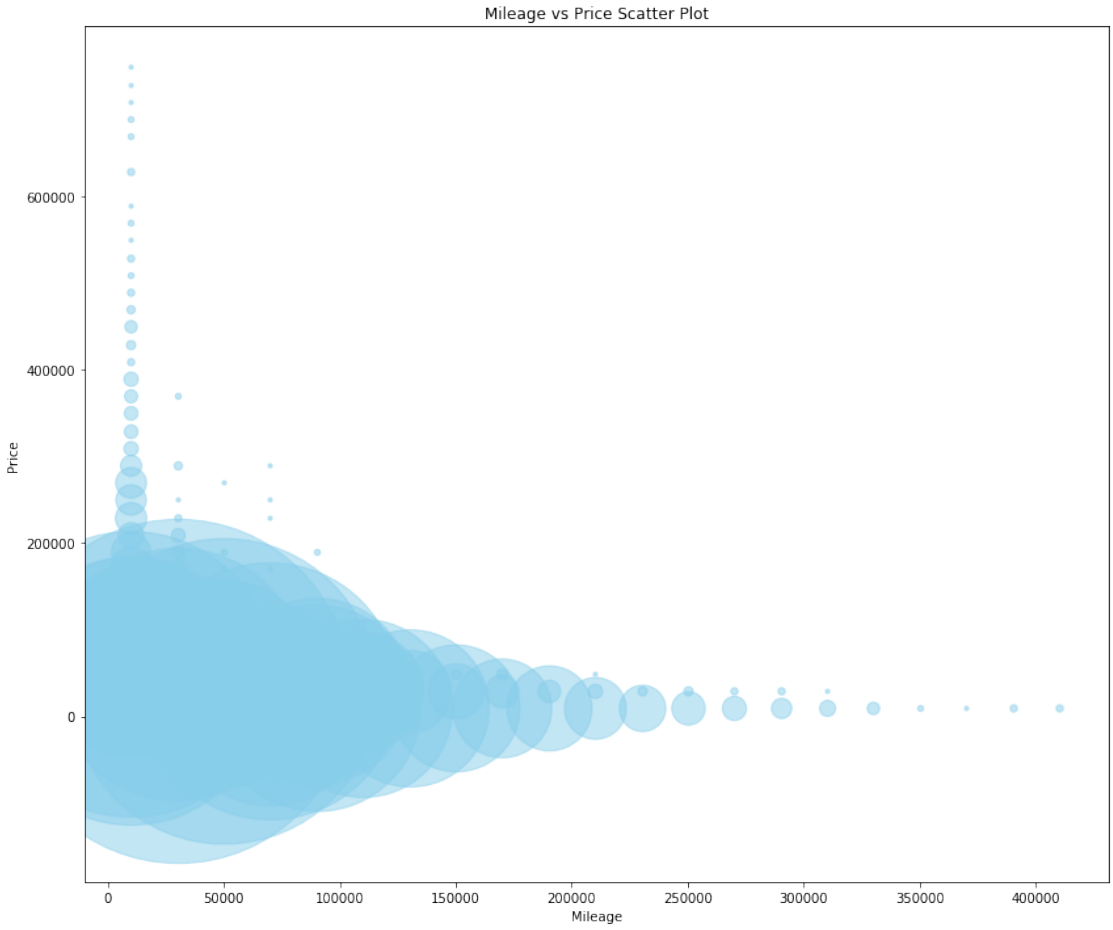



```
In [10]: #Analyisis 04_part_02
         import matplotlib.pyplot as plt
         import pandas as pd
         df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
         # Defining price bins for range 0 to 800000 with intervals of 20000
         price_bins_800k = range(0, 800001, 20000)
         # Creating the histogram for 'Price' in the range 0 to 800,000
         plt.figure(figsize=(12, 6))  # Increase figure size for readability
         plt.hist(df['Price'], bins=price_bins_800k, color='skyblue', edgecolor='black')
         # Adding titles and labels
         plt.title('Histogram of Vehicle Prices (Range: 0 to 800,000)')
         plt.xlabel('Price Range')
         plt.ylabel('Number of Vehicles')
         # Show the plot
         plt.show()
```



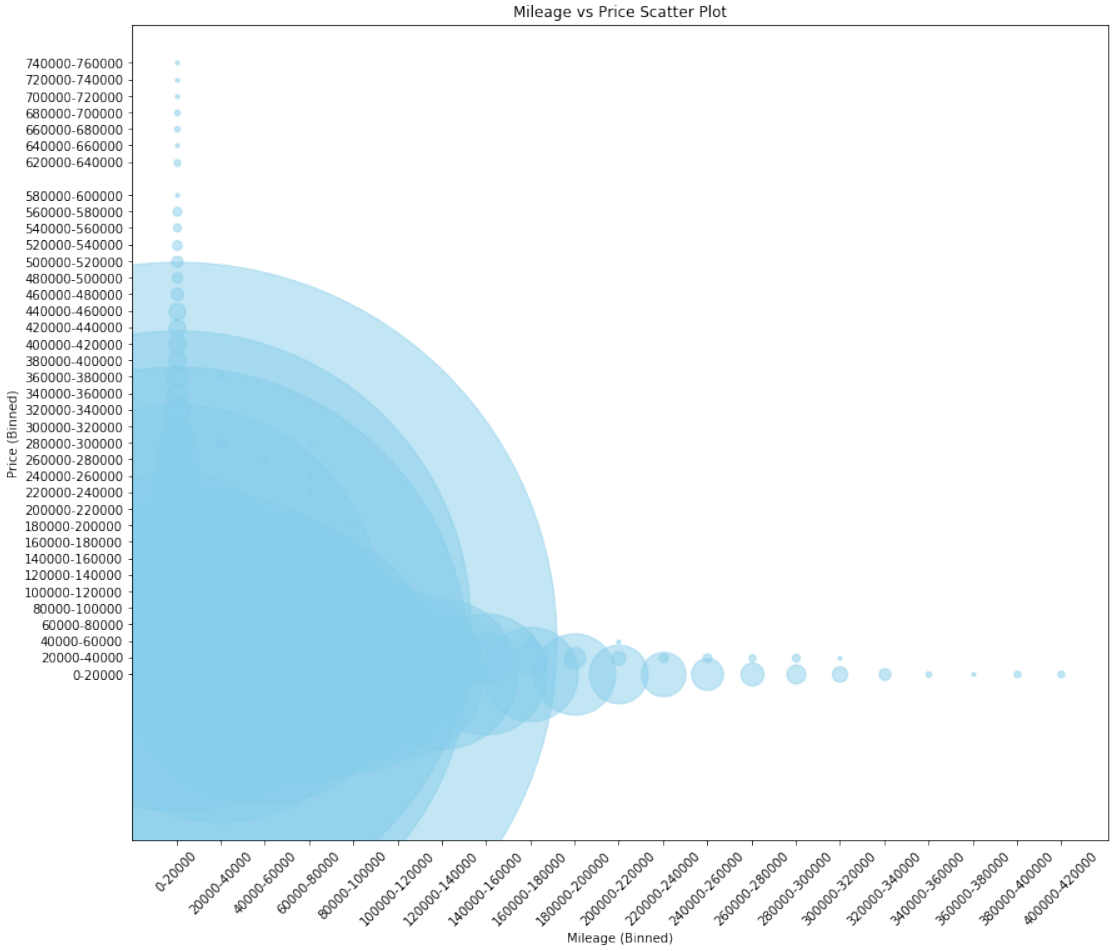

```
In [11]: #Analyisis 05_part_01
         import matplotlib.pyplot as plt
         import numpy as np
         df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
         # Define mileage bins (0 to 420000 with intervals of 20000)
         mileage_bins = range(0, 420001, 20000)
         # Define price bins (0 to 800000 with intervals of 20000)
         price_bins = range(0, 800001, 20000)
         # Bin the data into these ranges
         df['Mileage_bin'] = pd.cut(df['Mileage'], bins=mileage_bins)
         df['Price_bin'] = pd.cut(df['Price'], bins=price_bins)
         # Count the number of vehicles in each combination of mileage and price range
         vehicle_counts = pd.crosstab(df['Mileage_bin'], df['Price_bin'])
         # Create scatter plot
         fig, ax = plt.subplots(figsize=(14, 12))
         # Plotting the scatter plot using the vehicle counts as size
         # For visualization, we'll use the center of each bin as the x and y coordinates.
         for mileage_bin in vehicle_counts.index:
             for price_bin in vehicle_counts.columns:
                 # Getting the center of each bin
                 x = (mileage_bin.left + mileage_bin.right) / 2
                 y = (price_bin.left + price_bin.right) / 2
                 count = vehicle_counts.loc[mileage_bin, price_bin]
                 # Scatter plot with count as size
                 ax.scatter(x, y, s=count*10, color='skyblue', alpha=0.5)
         # Adding titles and labels
         plt.title('Mileage vs Price Scatter Plot')
         plt.xlabel('Mileage')
```

```
plt.ylabel('Price')
# Show the plot
plt.show()
```



In [12]:
```
#Analyisis 05_part_02
import matplotlib.pyplot as plt
import numpy as np
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Define mileage bins (0 to 420000 with intervals of 20000)
mileage_bins = range(0, 420001, 20000)
# Define price bins (0 to 800000 with intervals of 20000)
price_bins = range(0, 800001, 20000)
# Bin the data into these ranges
df['Mileage_bin'] = pd.cut(df['Mileage'], bins=mileage_bins, right=False)
df['Price_bin'] = pd.cut(df['Price'], bins=price_bins, right=False)
# Count the number of vehicles in each combination of mileage and price range
vehicle_counts = pd.crosstab(df['Mileage_bin'], df['Price_bin'])
# Create scatter plot
```

```
fig, ax = plt.subplots(figsize=(14, 12))
# Plotting the scatter plot using the vehicle counts as size
# We will convert the bins to categorical to avoid treating them as continuous values
for mileage_bin in vehicle_counts.index:
    for price_bin in vehicle_counts.columns:
        # Getting the center of each bin
        x = (mileage_bin.left + mileage_bin.right) / 2
        y = (price_bin.left + price_bin.right) / 2
        count = vehicle_counts.loc[mileage_bin, price_bin]
        # Scatter plot with count as size (scaled for visibility)
        ax.scatter(x, y, s=count*10, color='skyblue', alpha=0.5)
# Adding titles and labels
plt.title('Mileage vs Price Scatter Plot')
plt.xlabel('Mileage (Binned)')
plt.ylabel('Price (Binned)')
# Format the x and y ticks to show the actual bin labels
ax.set_xticks([(mileage_bin.left + mileage_bin.right) / 2 for mileage_bin in vehicle_
ax.set_xticklabels([f"{mileage_bin.left}-{mileage_bin.right}" for mileage_bin in vehi
ax.set_yticks([(price_bin.left + price_bin.right) / 2 for price_bin in vehicle_counts
ax.set_yticklabels([f"{price_bin.left}-{price_bin.right}" for price_bin in vehicle_co
# Show the plot
plt.show()
```



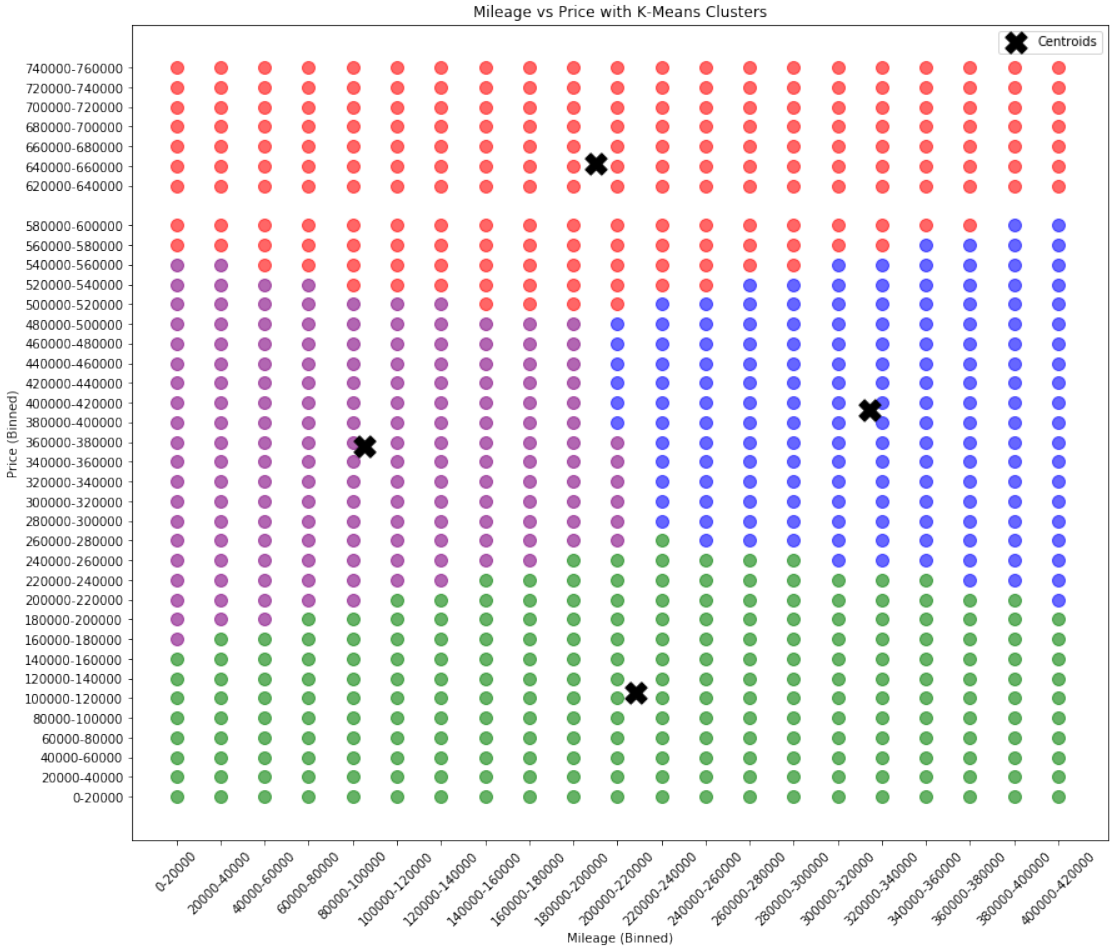
In [13]:
```
#Analyisis 06
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Calculate the correlation between Price and Mileage
correlation = df['Price'].corr(df['Mileage'])
# Print the correlation value
print(f"Correlation between Price and Mileage: {correlation}")
```

```
Correlation between Price and Mileage: -0.3387062919928825
```

In [16]:
```
#Analysis 07
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
```

```
# Define mileage bins (0 to 420000 with intervals of 20000)
mileage_bins = range(0, 420001, 20000)
# Define price bins (0 to 800000 with intervals of 20000)
price_bins = range(0, 800001, 20000)
# Bin the data into these ranges
df['Mileage_bin'] = pd.cut(df['Mileage'], bins=mileage_bins, right=False)
df['Price_bin'] = pd.cut(df['Price'], bins=price_bins, right=False)
# Count the number of vehicles in each combination of mileage and price range
vehicle_counts = pd.crosstab(df['Mileage_bin'], df['Price_bin'])
# Prepare data for clustering
X = np.array([[mileage_bin.mid, price_bin.mid]
              for mileage_bin in vehicle_counts.index
              for price_bin in vehicle_counts.columns])
# Apply KMeans clustering
kmeans = KMeans(n_clusters=4)  # You can adjust the number of clusters
kmeans.fit(X)
# Get cluster labels
labels = kmeans.labels_
# Create scatter plot
fig, ax = plt.subplots(figsize=(14, 12))
# Plot each cluster with different colors
for i in range(len(X)):
    ax.scatter(X[i][0], X[i][1], s=100, c=['blue', 'green', 'red', 'purple'][labels[i]
# Plot the centroids of the clusters
centroids = kmeans.cluster_centers_
ax.scatter(centroids[:, 0], centroids[:, 1], s=300, c='black', marker='X', label='Cent
# Adding titles and labels
plt.title('Mileage vs Price with K-Means Clusters')
plt.xlabel('Mileage (Binned)')
plt.ylabel('Price (Binned)')
# Format the x and y ticks to show the actual bin labels
ax.set_xticks([(mileage_bin.left + mileage_bin.right) / 2 for mileage_bin in vehicle_c
ax.set_xticklabels([f"{mileage_bin.left}-{mileage_bin.right}" for mileage_bin in vehic

ax.set_yticks([(price_bin.left + price_bin.right) / 2 for price_bin in vehicle_counts
ax.set_yticklabels([f"{price_bin.left}-{price_bin.right}" for price_bin in vehicle_cou
# Show the plot
plt.legend()
plt.show()
```



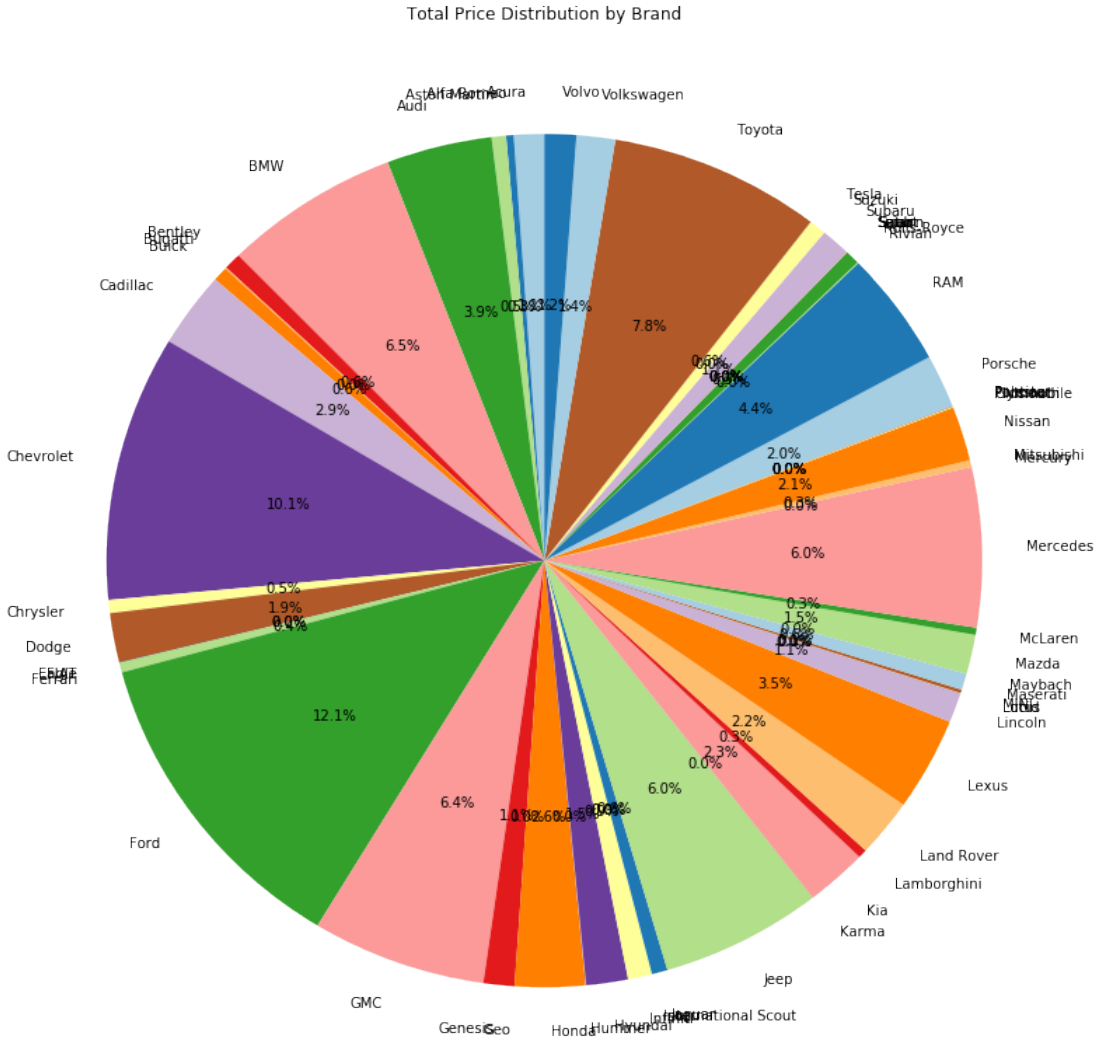
In [19]:
```
#Analysis 08_part_01
#Only 1 Bugatti is present in dataset. That is displayed in this graph, with its one
import plotly.express as px
import pandas as pd
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Filter data for Bugatti brand
bugatti_df = df[df['Brand'] == 'Bugatti']
# Group by 'Dealer' and calculate the mean price
dealer_mean_price = bugatti_df.groupby('Dealer')['Price'].mean().reset_index()
# Sort by mean price and take the top 20 dealers
top_dealers = dealer_mean_price.sort_values(by='Price', ascending=False).head(20)
# Create a bar plot using Plotly Express
fig = px.bar(top_dealers, x='Dealer', y='Price', title="Top 20 Dealers - Mean Price o
             labels={"Dealer": "Dealer", "Price": "Mean Price"},
             color='Price', color_continuous_scale='Viridis')
# Show the plot
fig.show()
```

In [20]:
```
#Analysis 08_part_02
import plotly.express as px
import pandas as pd
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Filter data for Ferrari brand
ferrari_df = df[df['Brand'] == 'Ferrari']
# Group by 'Dealer' and calculate the mean price
dealer_mean_price_ferrari = ferrari_df.groupby('Dealer')['Price'].mean().reset_index()
# Sort by mean price and take the top 20 dealers
top_dealers_ferrari = dealer_mean_price_ferrari.sort_values(by='Price', ascending=Fals
# Create a bar plot using Plotly Express
fig = px.bar(top_dealers_ferrari, x='Dealer', y='Price', title="Top 20 Dealers - Mean
             labels={"Dealer": "Dealer", "Price": "Mean Price"},
             color='Price', color_continuous_scale='Viridis')
# Update the layout to set the dimensions
fig.update_layout(
    width=1000,  # Width in pixels (14)
    height=800,  # Height in pixels (12)
)
# Show the plot
fig.show()
```

In [21]:
```
#Analysis 08_part_03
import plotly.express as px
import pandas as pd
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Filter data for Rolls-Royce brand
rolls_royce_df = df[df['Brand'] == 'Rolls-Royce']
# Group by 'Dealer' and calculate the mean price
dealer_mean_price_rolls_royce = rolls_royce_df.groupby('Dealer')['Price'].mean().reset
# Sort by mean price and take the top 20 dealers
top_dealers_rolls_royce = dealer_mean_price_rolls_royce.sort_values(by='Price', ascend
# Create a bar plot using Plotly Express
fig = px.bar(top_dealers_rolls_royce, x='Dealer', y='Price', title="Top 20 Dealers - R
             labels={"Dealer": "Dealer", "Price": "Mean Price"},
             color='Price', color_continuous_scale='Viridis')
# Update the layout to set the dimensions
fig.update_layout(
    width=1000,  # Width in pixels (14)
    height=1500,  # Height in pixels (12)
)
# Show the plot
fig.show()
```

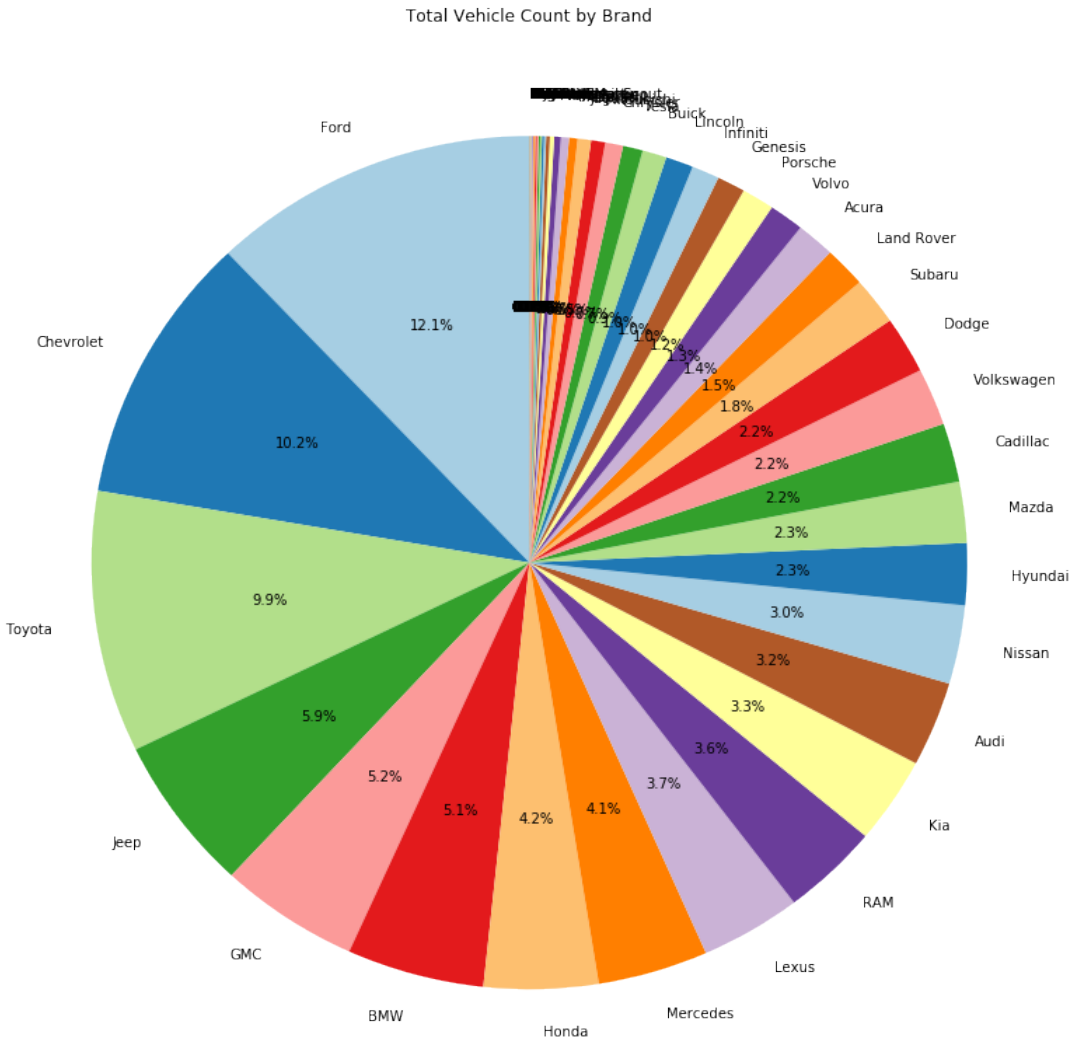
In [22]:
```
#Analysis 09_part_01
import matplotlib.pyplot as plt
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Group the data by 'Brand' and calculate the total price for each brand
```

```
total_price_per_brand = df.groupby('Brand')['Price'].sum()
# Set the figure size for the pie chart
plt.figure(figsize=(14, 14))
# Create a pie chart
plt.pie(total_price_per_brand, labels=total_price_per_brand.index, autopct='%1.1f%%',
# Adding a title
plt.title('Total Price Distribution by Brand')
# Show the plot
plt.show()
```



In [23]:
```
#Analysis 09_part_02
import matplotlib.pyplot as plt
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
```

```
# Group the data by 'Brand' and calculate the total count of vehicles for each brand
vehicle_count_per_brand = df['Brand'].value_counts()
# Set the figure size for the pie chart
plt.figure(figsize=(14, 14))
# Create a pie chart for the vehicle counts by brand
plt.pie(vehicle_count_per_brand, labels=vehicle_count_per_brand.index, autopct='%1.1f%
# Adding a title
plt.title('Total Vehicle Count by Brand')
# Show the plot
plt.show()
```



In [24]:
```
#Analysis 10
import plotly.express as px
```

```
import pandas as pd
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Group by 'Brand' and calculate the total price for each brand
brand_total_price = df.groupby('Brand')['Price'].sum().reset_index()
# Find the brand with the highest total price
top_brand = brand_total_price.sort_values(by='Price', ascending=False).iloc[0]['Brand
# Filter data for the top brand
top_brand_df = df[df['Brand'] == top_brand]
# Group by 'Dealer' and calculate the mean price for the top brand
dealer_mean_price_top_brand = top_brand_df.groupby('Dealer')['Price'].mean().reset_inc
# Sort by mean price and take the top 20 dealers
top_dealers_top_brand = dealer_mean_price_top_brand.sort_values(by='Price', ascending=
# Create a bar plot using Plotly Express
fig = px.bar(top_dealers_top_brand, x='Dealer', y='Price', title=f"Top 20 Dealers - Me
             labels={"Dealer": "Dealer", "Price": "Mean Price"},
             color='Price', color_continuous_scale='Viridis')
# Update the layout to set the dimensions
fig.update_layout(
    width=1000,  # Width in pixels (14)
    height=800,  # Height in pixels (12)
)
# Show the plot
fig.show()
```

In [25]:
```
#Analysis 11
import plotly.express as px
import pandas as pd
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Group by 'Brand' and calculate the total price for each brand
brand_total_price = df.groupby('Brand')['Price'].sum().reset_index()
# Find the brand with the highest total price
top_brand = brand_total_price.sort_values(by='Price', ascending=False).iloc[0]['Brand
# Filter data for the top brand
top_brand_df = df[df['Brand'] == top_brand]
# Group by 'Model' and calculate the mean price for the top brand
model_mean_price = top_brand_df.groupby('Model')['Price'].mean().reset_index()
# Sort by mean price and take the top 20 models
top_20_models = model_mean_price.sort_values(by='Price', ascending=False).head(20)
# Create a bar plot using Plotly Express
fig = px.bar(top_20_models, x='Model', y='Price', title=f"Top 20 Models - Mean Price
             labels={"Model": "Model", "Price": "Mean Price"},
             color='Price', color_continuous_scale='Viridis')
# Update the layout to set the dimensions
fig.update_layout(
    width=1000,  # Width in pixels (14)
    height=800,  # Height in pixels (12)
)
# Show the plot
```

```
fig.show()
```

In [26]:
```
#Analysis 12
import pandas as pd
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Group by 'Brand' and count the number of vehicles for each brand
brand_vehicle_count = df.groupby('Brand').size().reset_index(name='Vehicle Count')
# Find the brand with the highest number of vehicles
top_brand_by_vehicles = brand_vehicle_count.sort_values(by='Vehicle Count', ascending=
# Filter data for the top brand
top_brand_df = df[df['Brand'] == top_brand_by_vehicles]
# Calculate the mean price for the top brand
mean_price_top_brand = top_brand_df['Price'].mean()
# Output the results
print(f"The brand with the highest number of vehicles is {top_brand_by_vehicles}.")
print(f"The mean price of vehicles from {top_brand_by_vehicles} is {mean_price_top_bra
```

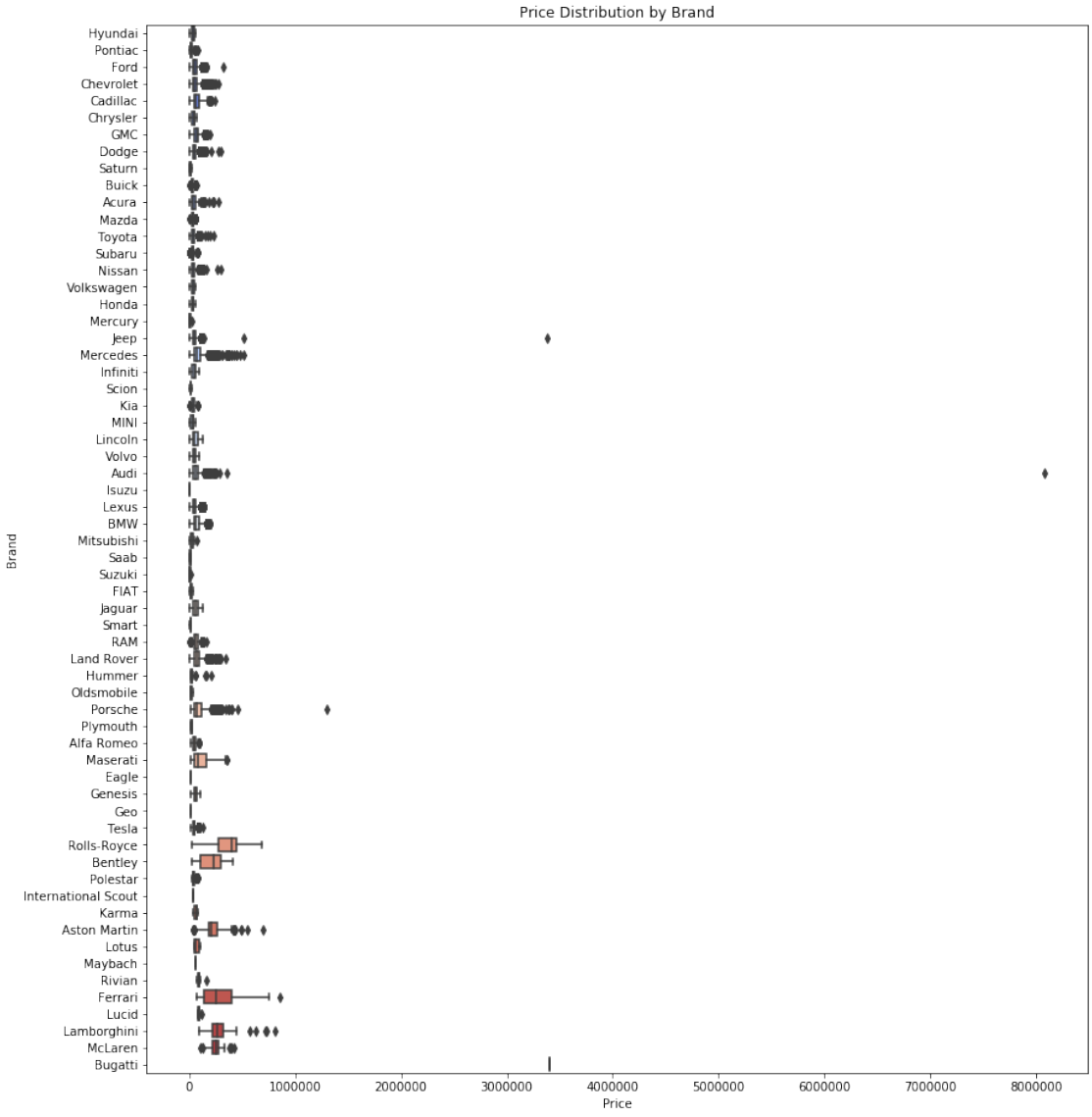
```
The brand with the highest number of vehicles is Ford.
The mean price of vehicles from Ford is 52774.63.
```

In [27]:
```
#Analysis 13
import pandas as pd
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Group by 'Year' and count the number of vehicles for each year
year_vehicle_count = df.groupby('Year').size().reset_index(name='Vehicle Count')
# Find the year with the highest number of vehicles
top_year_by_vehicles = year_vehicle_count.sort_values(by='Vehicle Count', ascending=Fa
top_year_vehicle_count = year_vehicle_count.sort_values(by='Vehicle Count', ascending=
# Filter data for the top year
top_year_df = df[df['Year'] == top_year_by_vehicles]
# Calculate the mean price for the top year
mean_price_top_year = top_year_df['Price'].mean()
# Output the results including the number of vehicles
print(f"The year with the highest number of vehicles is {top_year_by_vehicles}, with
print(f"The mean price of vehicles from the year {top_year_by_vehicles} is {mean_price
```

```
The year with the highest number of vehicles is 2023, with 53523 vehicles.
The mean price of vehicles from the year 2023 is 62838.26.
```

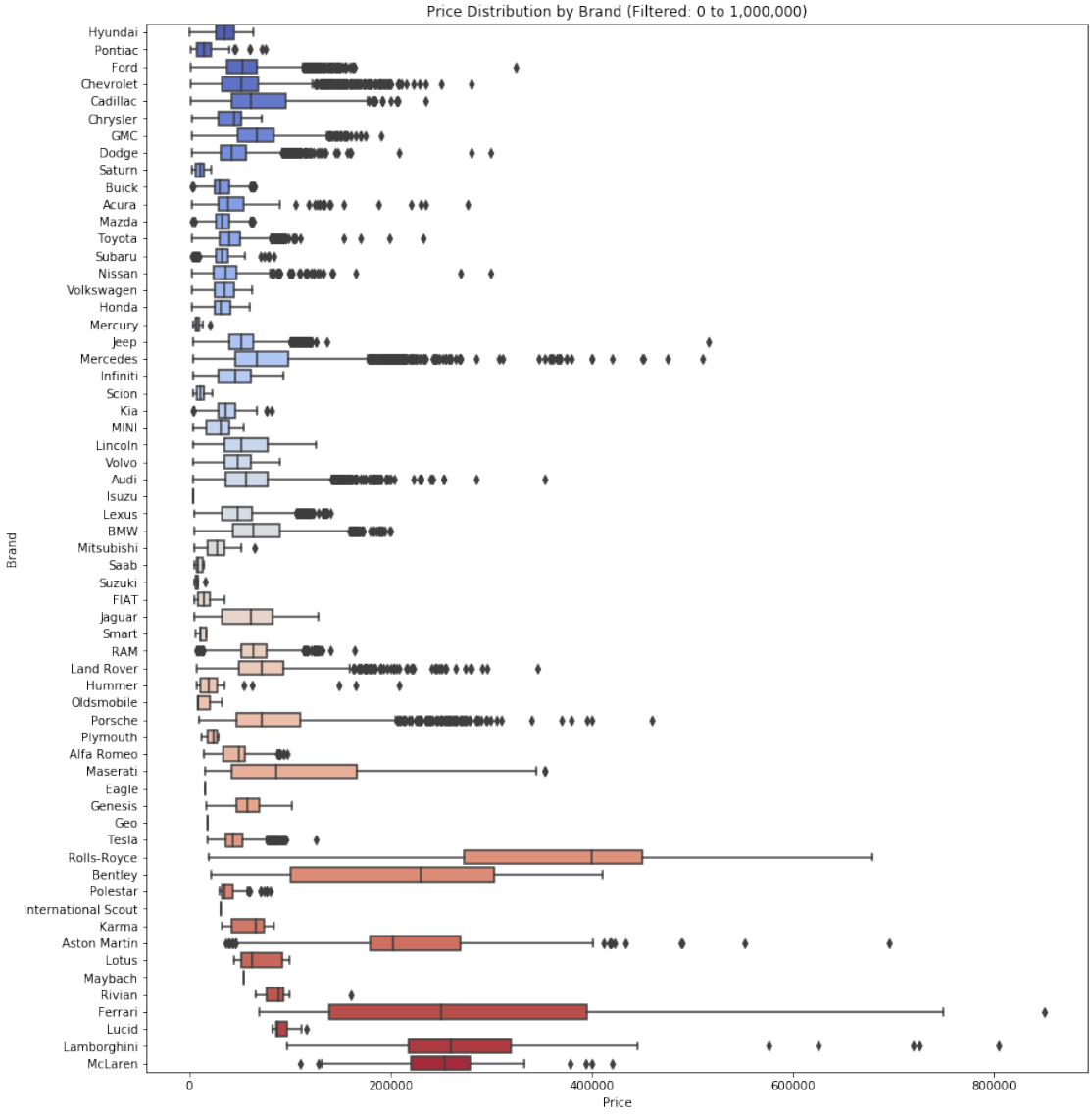
In [28]:
```
#Analysis 14_part_01
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Set the figure size
plt.figure(figsize=(14, 16))
# Create a boxplot for Price across different Brands, with brands on the y-axis
sns.boxplot(x='Price', y='Brand', data=df, palette='coolwarm')
```

```
# Adding titles and labels
plt.title('Price Distribution by Brand')
plt.xlabel('Price')
plt.ylabel('Brand')
# Show the plot
plt.show()
```



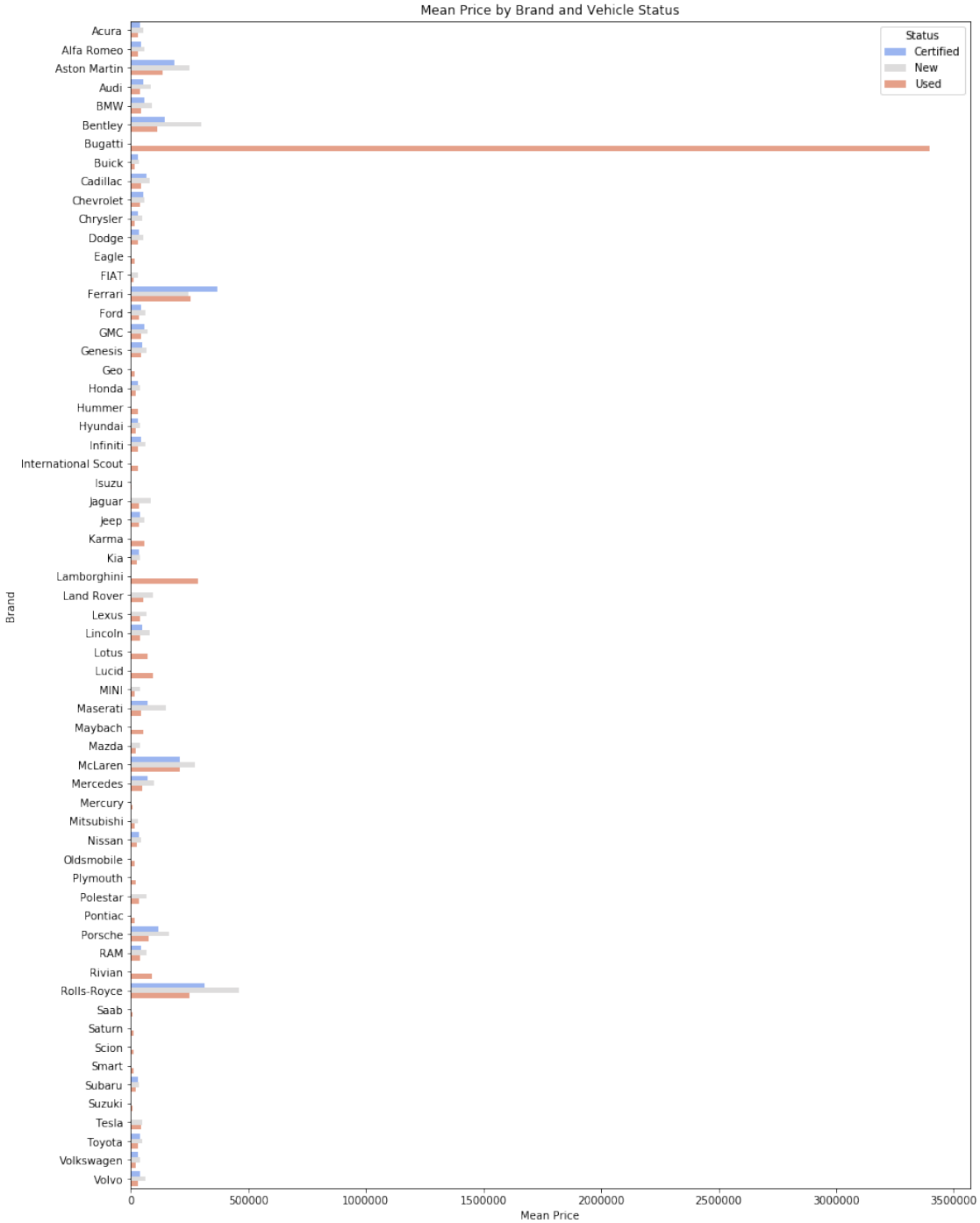
In [29]:
```
#Analysis 14_part_02
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Filter the data to only include prices within the range 0 to 1,000,000
df_filtered = df[(df['Price'] >= 0) & (df['Price'] <= 1000000)]
```

```
# Set the figure size
plt.figure(figsize=(14, 16))
# Create a boxplot for Price across different Brands, with brands on the y-axis
sns.boxplot(x='Price', y='Brand', data=df_filtered, palette='coolwarm')
# Adding titles and labels
plt.title('Price Distribution by Brand (Filtered: 0 to 1,000,000)')
plt.xlabel('Price')
plt.ylabel('Brand')
# Show the plot
plt.show()
```



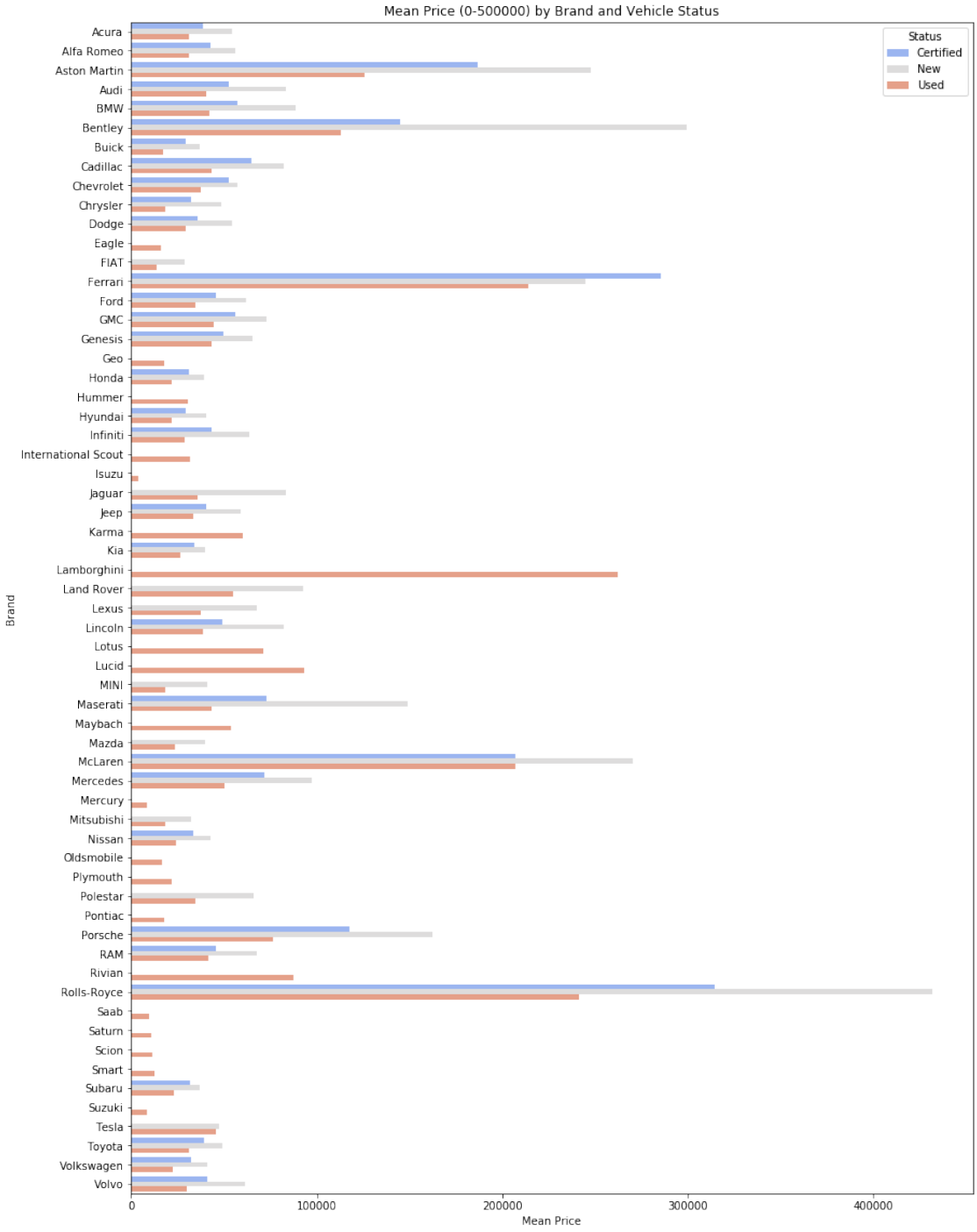
In [37]:
```
#Analysis 15_part_01
import seaborn as sns
```

```
import matplotlib.pyplot as plt
df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
# Grouping by Brand and Status to calculate the mean price
mean_price_by_status = df.groupby(['Brand', 'Status'])['Price'].mean().reset_index()
# Set the figure size for the bar chart
plt.figure(figsize=(14, 20))
# Create a side-by-side bar plot with Brand on Y-axis and Price on X-axis
sns.barplot(y='Brand', x='Price', hue='Status', data=mean_price_by_status, palette='c
# Adding titles and labels
plt.title('Mean Price by Brand and Vehicle Status')
plt.ylabel('Brand')
plt.xlabel('Mean Price')
# Show the plot
plt.show()
```




```
In [38]: #Analysis 15_part_02
         import seaborn as sns
         import matplotlib.pyplot as plt
         df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
         # Filter the data to include only vehicles with a price between 0 and 500000
```

```
df_filtered = df[(df['Price'] >= 0) & (df['Price'] <= 500000)]
# Grouping by Brand and Status to calculate the mean price
mean_price_by_status = df_filtered.groupby(['Brand', 'Status'])['Price'].mean().reset_
# Set the figure size for the bar chart
plt.figure(figsize=(14, 20))
# Create a side-by-side bar plot with Brand on Y-axis and Price on X-axis
sns.barplot(y='Brand', x='Price', hue='Status', data=mean_price_by_status, palette='co
# Adding titles and labels
plt.title('Mean Price (0-500000) by Brand and Vehicle Status')
plt.ylabel('Brand')
plt.xlabel('Mean Price')
# Show the plot
plt.show()
```




```
In [39]: #Analysis 05 (Using Plotly)
         import plotly.express as px
         import pandas as pd
         # Load the CSV data into a DataFrame (replace 'your_file.csv' with the actual file pa
         df = pd.read_csv('car_sale_usa.csv', encoding='utf-16')
```

```
# Create a scatter plot using Plotly Express
fig = px.scatter(df, x='Mileage', y='Price', title="Mileage vs Price",
                 labels={"Mileage": "Mileage", "Price": "Price"},
                 color='Mileage',  # You can also color by another column
                 hover_data=['Mileage', 'Price'])  # Display more information on hove
# Show the plot
fig.show()
```

In [ ]: