hot-wheels-sales-analysis-2025-1

November 17, 2025

0.0.1 HOT WHEELS SALES ANALYSIS 2025

[2]:
```
pwd
```

[2]:
```
'C:\\Users\\Admin\\PY-Project 2025'
```

[3]:
```
cd ..
```

```
C:\Users\Admin
```

[4]:
```
cd Downloads
```

```
C:\Users\Admin\Downloads
```

[5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style='whitegrid')
plt.rcParams['figure.figsize'] = (10,6)
```

[6]:
```
df = pd.read_csv("hotwheels_dataset.csv")
print("Rows, Columns:", df.shape)
display(df.head(101))
df.info()
```

```
Rows, Columns: (1600, 11)

     Car_ID              Car_Name  Model_Year           Series Country_Sold  \
0    1430.0  Lamborghini Aventador      1968.0  Limited Edition        India
1    1192.0                 BMW M3      1968.0  Limited Edition        Japan
2    1295.0             Mustang GT      1968.0  Limited Edition        Japan
3      63.0                 BMW M3      1968.0  Limited Edition           UK
4    1512.0  Lamborghini Aventador      1968.0  Limited Edition          USA
..      …                      …          …               …            …
96    768.0            Corvette Z06      1972.0  Limited Edition          USA
97    905.0  Lamborghini Aventador      1972.0  Limited Edition          USA
98    167.0  Lamborghini Aventador      1972.0  Limited Edition          USA
```

```
99      NaN                 BMW M3      1972.0  Limited Edition        Japan
100   697.0                 BMW M3      1972.0         Mainline      Germany

     Units_Sold  Price_USD Age_Group_Buyers Is_Banned Manufacturing_Country  \
0      121585.0        NaN              NaN        No              Thailand
1      477293.0     165.66           05-Dec        No             Indonesia
2        1430.0      92.90           05-Dec        No             Indonesia
3      353832.0     108.95              25+        No              Malaysia
4      475327.0     146.24            13-18        No              Malaysia
..          …        …               …       …                   …
96      20360.0     247.66            13-18        No                   NaN
97     354828.0     157.12           05-Dec        No              Thailand
98     128678.0     225.03            13-18        No             Indonesia
99     156816.0     240.78              25+       NaN              Malaysia
100    484781.0     149.35            13-18        No              Thailand

     Rating
0       8.8
1       7.1
2       6.6
3       6.8
4       6.7
..      …
96      8.0
97      8.7
98      5.8
99      7.6
100     8.4

[101 rows x 11 columns]

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1600 entries, 0 to 1599
Data columns (total 11 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Car_ID                 1520 non-null   float64
 1   Car_Name               1520 non-null   object 
 2   Model_Year             1520 non-null   float64
 3   Series                 1520 non-null   object 
 4   Country_Sold           1520 non-null   object 
 5   Units_Sold             1520 non-null   float64
 6   Price_USD              1520 non-null   float64
 7   Age_Group_Buyers       1520 non-null   object 
 8   Is_Banned              1520 non-null   object 
 9   Manufacturing_Country  1520 non-null   object 
 10  Rating                 1520 non-null   float64
dtypes: float64(5), object(6)
```

```
memory usage: 137.6+ KB
```

[17]:

```
df.columns
```

[17]:

```
Index(['Car_ID', 'Car_Name', 'Model_Year', 'Series', 'Country_Sold',
       'Units_Sold', 'Price_USD', 'Age_Group_Buyers', 'Is_Banned',
       'Manufacturing_Country', 'Rating'],
      dtype='object')
```

[ ]:

```
# show missing counts
print("Missing values per column:\n", df.isnull().sum())

# remove exact duplicate rows
df.drop_duplicates(inplace=True)
print("After drop_duplicates:", df.shape)
```

0.0.2 4. Clean important columns (drop only truly broken rows; fill others sensibly)

[ ]:

```
# Keep rows that have essential values (Car_Name, Model_Year, Units_Sold)
df = df.dropna(subset=['Car_Name','Model_Year','Units_Sold']).copy()

# Convert Model_Year to integer (safe cast)
df['Model_Year'] = df['Model_Year'].astype(int)

# Fill common columns sensibly
df['Price_USD'] = df['Price_USD'].fillna(df['Price_USD'].median())          #␣
 ↪numeric
df['Rating'] = df['Rating'].fillna(df['Rating'].median())
df['Age_Group_Buyers'] = df['Age_Group_Buyers'].fillna('Unknown')
df['Is_Banned'] = df['Is_Banned'].fillna('No')
df['Manufacturing_Country'] = df['Manufacturing_Country'].fillna('Unknown')
df['Series'] = df['Series'].fillna('Unknown')

# Verify
print("After cleaning:", df.shape)
print(df.isnull().sum())
```

0.0.3 5 Top 10 models by units sold

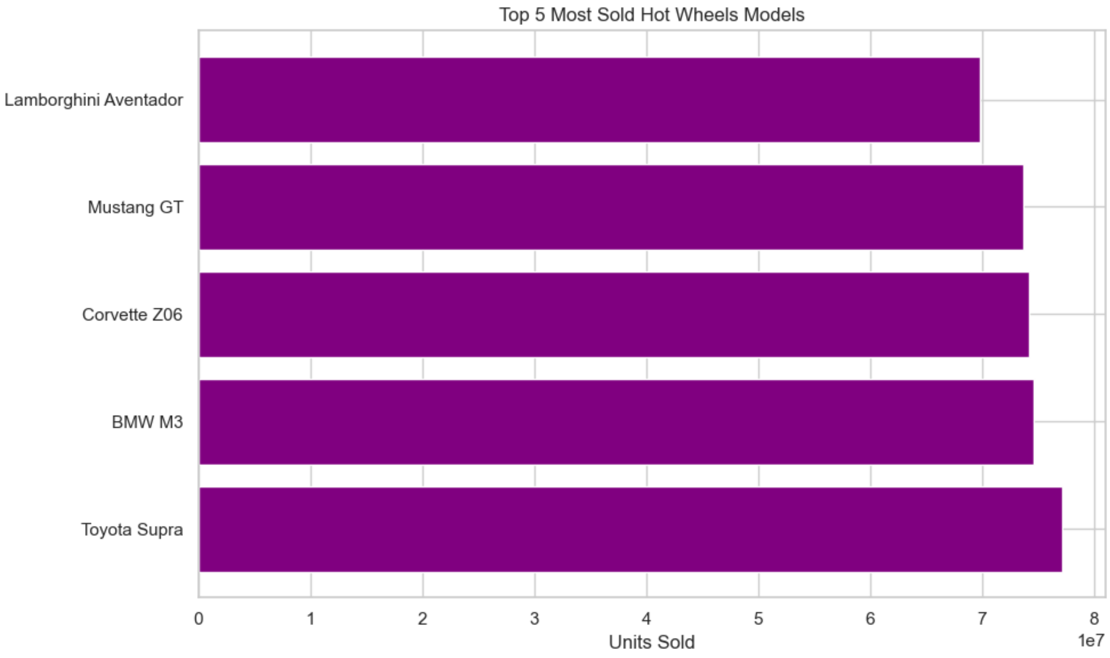
[7]:

```
top_models = df.groupby('Car_Name')['Units_Sold'].sum().nlargest(10)

# Total sales per year
yearly_sales = df.groupby('Model_Year')['Units_Sold'].sum()

# Top manufacturing countries
mfg_country_sales = df.groupby('Manufacturing_Country')['Units_Sold'].sum().
 ↪nlargest(10)
```

```
# Banned cars summary
banned_summary = df[df['Is_Banned'] == 'Yes'].groupby('Car_Name')['Units_Sold'].
↪sum()
```

0.0.4 6. TOP 5 MOST SOLD HOT WHEELS MODELS

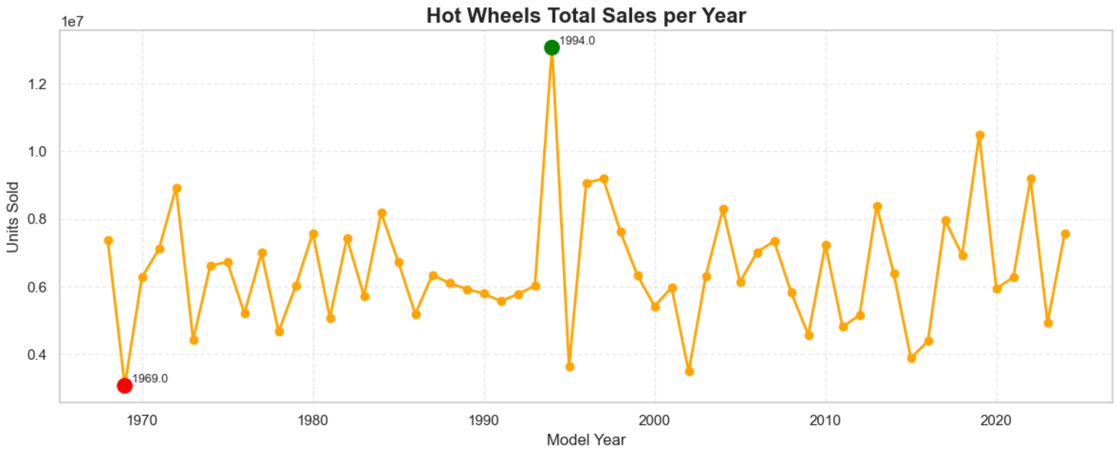
[8]:
```
#6
top_models = df.groupby('Car_Name')['Units_Sold'].sum().nlargest(10)
plt.barh(top_models.index, top_models.values, color='purple')
plt.title('Top 5 Most Sold Hot Wheels Models')
plt.xlabel('Units Sold')
plt.tight_layout()
plt.show()
```



0.0.5 7.TOTAL SALES PER YEAR

[9]:
```
plt.figure(figsize=(12,5))
plt.plot(yearly_sales.index, yearly_sales.values, marker='o', color='orange',␣
↪linewidth=2)
plt.title('Hot Wheels Total Sales per Year', fontsize=16, fontweight='bold')
plt.xlabel('Model Year'); plt.ylabel('Units Sold'); plt.grid(True,␣
↪linestyle='--', alpha=0.4)

# highlight
```

```
max_year = yearly_sales.idxmax()
min_year = yearly_sales.idxmin()
plt.scatter([max_year, min_year], [yearly_sales[max_year],␣
 ↪yearly_sales[min_year]], color=['green','red'], s=120, zorder=5)
plt.text(max_year, yearly_sales[max_year], f'  {max_year}', va='bottom',␣
 ↪fontsize=9)
plt.text(min_year, yearly_sales[min_year], f'  {min_year}', va='bottom',␣
 ↪fontsize=9)

plt.tight_layout()
plt.show()
```



[10]:

```
#8
CLEAN_PATH = "hotwheels_dataset_cleaned.csv"
df.to_csv(CLEAN_PATH, index=False)
print("Saved cleaned dataset to", CLEAN_PATH)
```

```
Saved cleaned dataset to hotwheels_dataset_cleaned.csv
```

[18]:

```
assert df['Model_Year'].dtype == int
assert df[['Car_Name','Units_Sold']].notnull().all().all()
```

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
Cell In[18], line 1
----> 1 assert df['Model_Year'].dtype == int
      2 assert df[['Car_Name','Units_Sold']].notnull().all().all()

AssertionError:
```

[19]:

```
series_counts = df['Series'].value_counts()
```

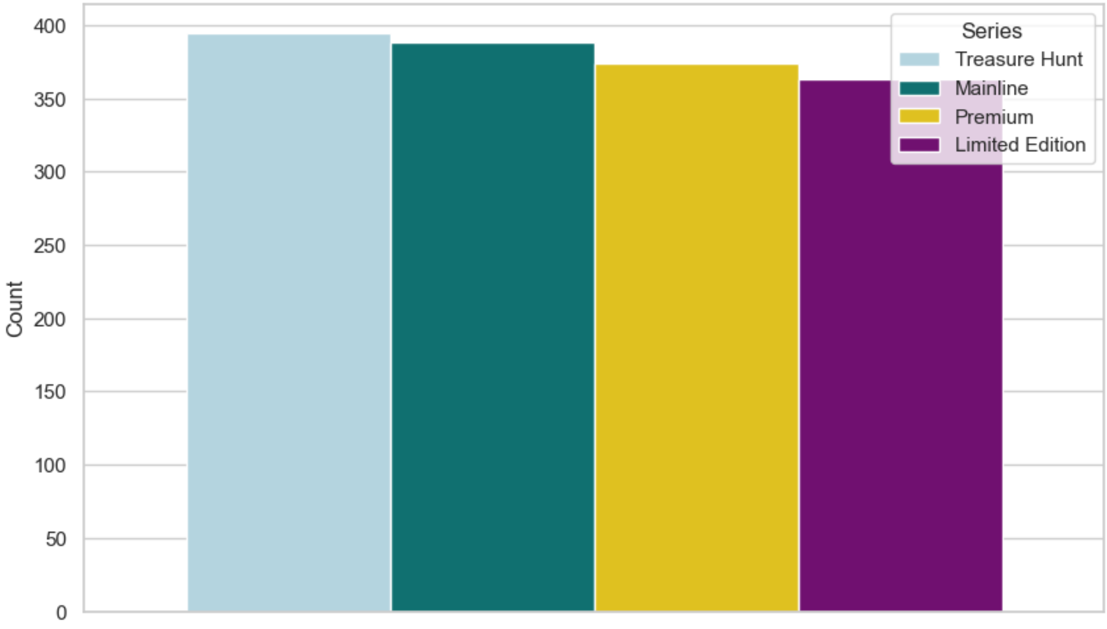
[13]:

```
df_series = df['Series'].value_counts().reset_index()
df_series.columns = ['Series', 'Count']
```

0.0.6 THE 4 SERIES

[16]:

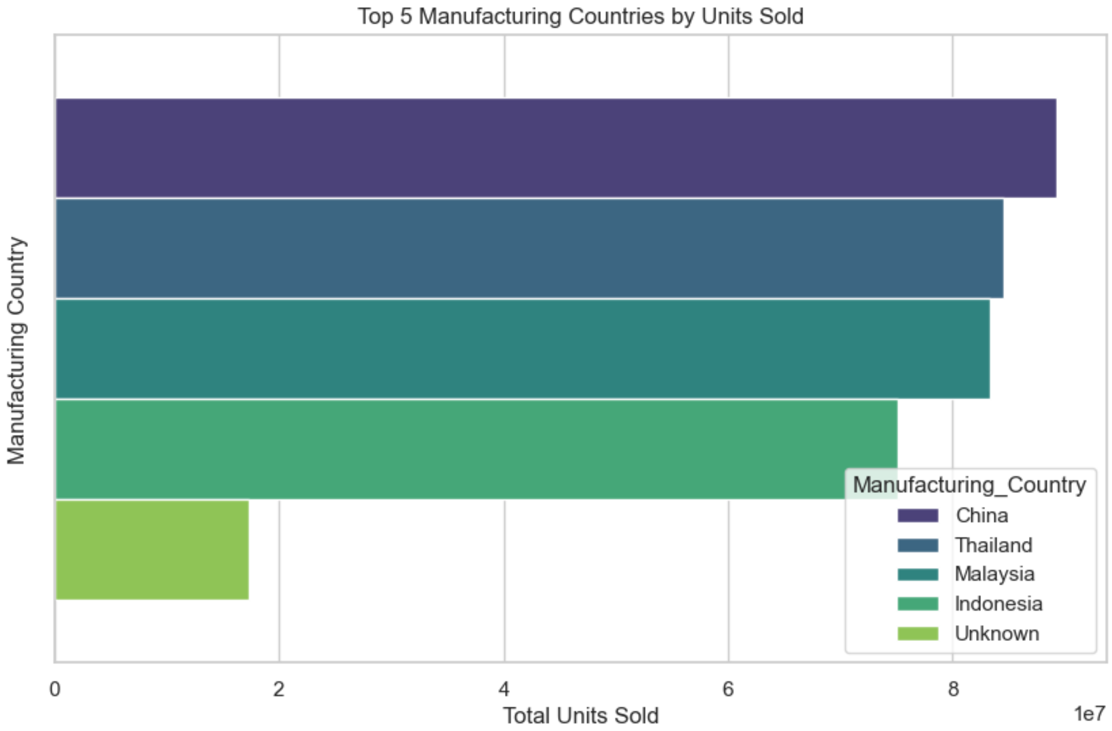
```
sns.barplot(data=df_series, hue="Series", y="Count",
            palette=["lightblue","teal","gold","purple"])
plt.show()
```



0.0.7 Top 5 Manufacturing Countries by Units Sold

[37]:

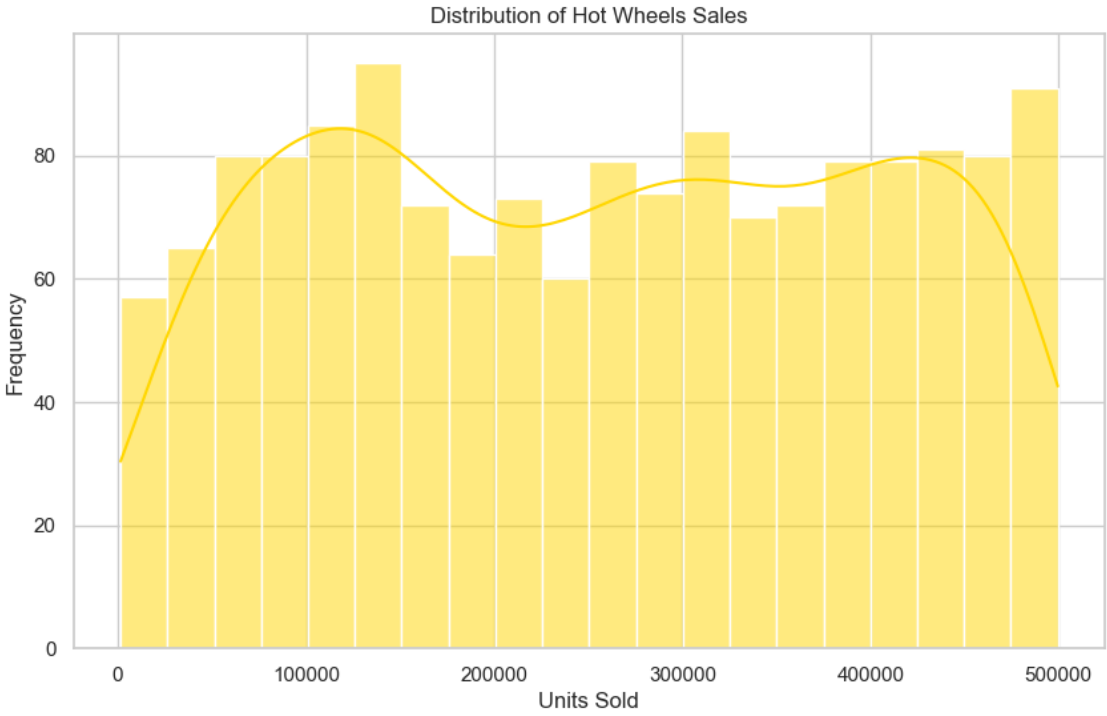
```
mfg_country_sales = df.groupby('Manufacturing_Country')['Units_Sold'].sum().
  ↪sort_values(ascending=False).head(10)

# Plot bar chart
sns.barplot(x=mfg_country_sales.values, hue=mfg_country_sales.index,␣
  ↪palette='viridis')
plt.title('Top 5 Manufacturing Countries by Units Sold')
plt.xlabel('Total Units Sold')
plt.ylabel('Manufacturing Country')
plt.show()
```



0.0.8 Histogram — Distribution of Units Sold

[20]:

```
sns.histplot(df['Units_Sold'], bins=20, kde=True, color='gold')
plt.title('Distribution of Hot Wheels Sales')
plt.xlabel('Units Sold')
plt.ylabel('Frequency')
plt.show()
```

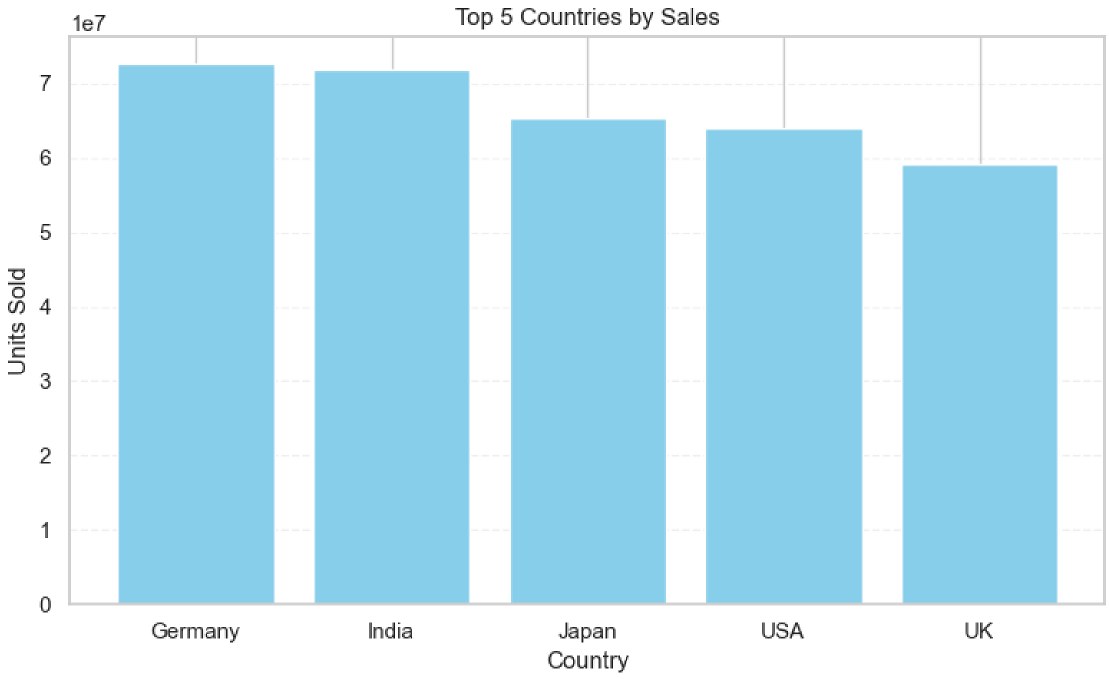


[39]:
```
banned = df[df['Is_Banned'] == 'Yes']
banned.groupby('Car_Name')['Units_Sold'].sum().nlargest(10)
```

[39]:
```
Car_Name
Toyota Supra             3801332.0
Mustang GT               3246861.0
BMW M3                   2373522.0
Corvette Z06             2323931.0
Lamborghini Aventador    1528605.0
Name: Units_Sold, dtype: float64
```

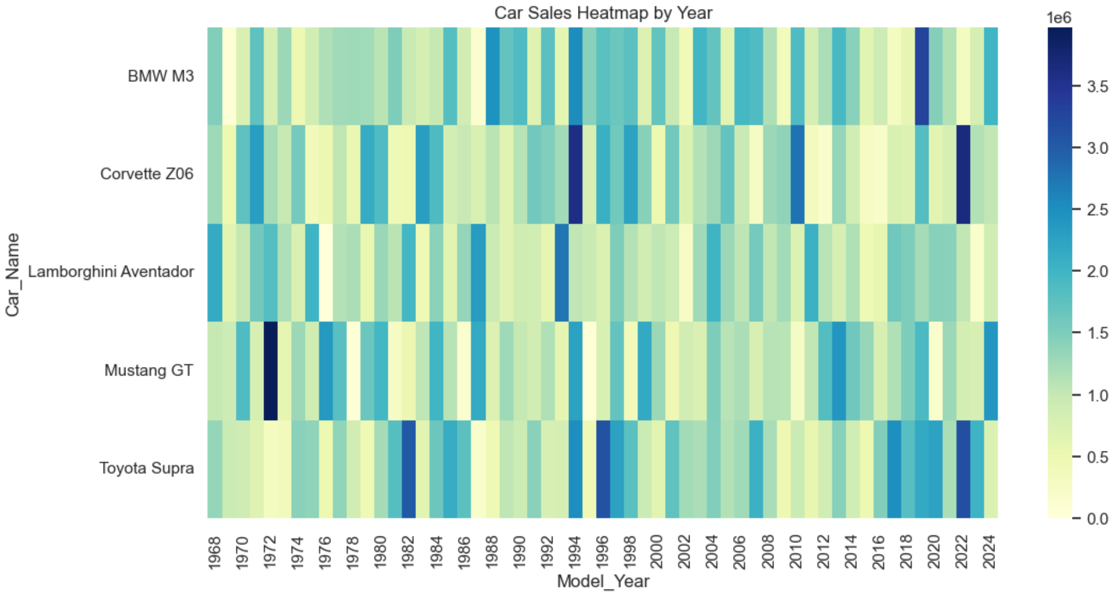
0.0.9 TOP 5 COUNTRIES BY SALES

[40]:
```
country_sales = df.groupby('Country_Sold')['Units_Sold'].sum().nlargest(5)

plt.figure(figsize=(8,5))
plt.bar(country_sales.index, country_sales.values, color="skyblue")

plt.title("Top 5 Countries by Sales")
plt.xlabel("Country")
plt.ylabel("Units Sold")
plt.grid(axis='y', linestyle='--', alpha=0.3)

plt.tight_layout()
plt.show()
```



0.0.10 CAR SALES WITH HEATMAP BY YEAR

[22]:
```
car_year_sales = df.pivot_table(
    values='Units_Sold',
    index='Car_Name',
    columns='Model_Year',
    aggfunc='sum',
    fill_value=0
)

plt.figure(figsize=(12,6))
sns.heatmap(car_year_sales, cmap="YlGnBu")
plt.title("Car Sales Heatmap by Year")
plt.show()
```



0.0.11 Summary: Hot Wheels Sales Analysis

This project analyzes a dataset of 1,600 Hot Wheels car entries to understand sales patterns, model popularity, manufacturing trends, and global market performance. After cleaning missing and duplicate values, the data was explored through several visualizations. The analysis identifies the top-selling car models, the strongest manufacturing countries, and the top buyer markets. Yearly sales trends reveal fluctuations in demand, while age-group and series distributions show consumer preferences. Additional insights include sales distribution patterns, banned car statistics, and a heatmap showing long-term model performance across multiple years.

Overall, the project provides a complete end-to-end understanding of how Hot Wheels perform globally — across models, years, countries, and manufacturing origins.

[ ]: