```
# car_sales analysis

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv("/content/Car_sales.csv")
df
```

	Manufacturer	Model	Sales in thousands	4-year resale value	Vehicle type	Price in thousands	Engine size	Horsepower	Wheelbase	Width	Length	Curb weight
0	Acura	Integra	16.919	16.36	Passenger	21.5	1.8	140	101.2	67.3	172.4	2.639
1	Acura	TL	39.384	19.875	Passenger	28.4	3.2	225	108.1	70.3	192.9	3.517
2	Acura	CL	NaN	NaN	Passenger	.	3.2	225	106.9	70.6	192	3.47
3	Acura	RL	8.588	29.725	Passenger	42	3.5	210	114.6	71.4	196.6	3.85
4	Audi	A4	20.397	22.255	Passenger	23.99	1.8	150	102.6	68.2	178	2.998
...	...	...	...	...	...	...	...	...	...	...	...	...
152	Volvo	V40	3.545	.	Passenger	24.4	1.9	160	100.5	67.6	176.6	3.042
153	Volvo	S70	15.245	.	Passenger	27.5	2.4	168	104.9	69.3	185.9	3.208
154	Volvo	V70	17.531	.	Passenger	28.8	2.4	168	104.9	69.3	186.2	3.259
155	Volvo	C70	3.493	.	Passenger	45.5	2.3	236	104.9	71.5	185.7	3.601
156	Volvo	S80	18.969	.	Passenger	36	2.9	201	109.9	72.1	189.8	3.6

157 rows × 15 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.head()
```

	Manufacturer	Model	Sales in thousands	4-year resale value	Vehicle type	Price in thousands	Engine size	Horsepower	Wheelbase	Width	Length	Curb weight	c
0	Acura	Integra	16.919	16.36	Passenger	21.5	1.8	140	101.2	67.3	172.4	2.639	
1	Acura	TL	39.384	19.875	Passenger	28.4	3.2	225	108.1	70.3	192.9	3.517	
2	Acura	CL	NaN	NaN	Passenger	.	3.2	225	106.9	70.6	192	3.47	
3	Acura	RL	8.588	29.725	Passenger	42	3.5	210	114.6	71.4	196.6	3.85	
4	Audi	A4	20.397	22.255	Passenger	23.99	1.8	150	102.6	68.2	178	2.998	

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.tail()
```

	Manufacturer	Model	Sales in thousands	4-year resale value	Vehicle type	Price in thousands	Engine size	Horsepower	Wheelbase	Width	Length	Curb weight
152	Volvo	V40	3.545	.	Passenger	24.4	1.9	160	100.5	67.6	176.6	3.042
153	Volvo	S70	15.245	.	Passenger	27.5	2.4	168	104.9	69.3	185.9	3.208
154	Volvo	V70	17.531	.	Passenger	28.8	2.4	168	104.9	69.3	186.2	3.259
155	Volvo	C70	3.493	.	Passenger	45.5	2.3	236	104.9	71.5	185.7	3.601
156	Volvo	S80	18.969	.	Passenger	36	2.9	201	109.9	72.1	189.8	3.6

```
# info about data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 157 entries, 0 to 156
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Manufacturer         157 non-null    object 
 1   Model                157 non-null    object 
 2   Sales in thousands   156 non-null    float64
 3   4-year resale value  156 non-null    object 
 4   Vehicle type         157 non-null    object 
 5   Price in thousands   157 non-null    object 
 6   Engine size          157 non-null    object 
 7   Horsepower           157 non-null    object 
 8   Wheelbase            157 non-null    object 
 9   Width                157 non-null    object 
 10  Length               157 non-null    object 
 11  Curb weight          157 non-null    object 
```

```
 12  Fuel capacity        157 non-null    object
 13  Fuel efficiency      157 non-null    object
 14  Latest Launch        157 non-null    object
dtypes: float64(1), object(14)
memory usage: 18.5+ KB
```

```
# cheacking null values
df.isnull().sum()
```

	0
Manufacturer	0
Model	0
Sales in thousands	1
4-year resale value	1
Vehicle type	0
Price in thousands	0
Engine size	0
Horsepower	0
Wheelbase	0
Width	0
Length	0
Curb weight	0
Fuel capacity	0
Fuel efficiency	0
Latest Launch	0

dtype: int64

```
# drop the dublicates values
df.dropna(inplace=True)
df
```

	Manufacturer	Model	Sales in thousands	4-year resale value	Vehicle type	Price in thousands	Engine size	Horsepower	Wheelbase	Width	Length	Curb weight
0	Acura	Integra	16.919	16.36	Passenger	21.5	1.8	140	101.2	67.3	172.4	2.639
1	Acura	TL	39.384	19.875	Passenger	28.4	3.2	225	108.1	70.3	192.9	3.517
3	Acura	RL	8.588	29.725	Passenger	42	3.5	210	114.6	71.4	196.6	3.85
4	Audi	A4	20.397	22.255	Passenger	23.99	1.8	150	102.6	68.2	178	2.998
5	Audi	A6	18.780	23.555	Passenger	33.95	2.8	200	108.7	76.1	192	3.561
...	...	...	...	...	...	...	...	...	...	...	...	...
152	Volvo	V40	3.545	.	Passenger	24.4	1.9	160	100.5	67.6	176.6	3.042
153	Volvo	S70	15.245	.	Passenger	27.5	2.4	168	104.9	69.3	185.9	3.208
154	Volvo	V70	17.531	.	Passenger	28.8	2.4	168	104.9	69.3	186.2	3.259
155	Volvo	C70	3.493	.	Passenger	45.5	2.3	236	104.9	71.5	185.7	3.601
156	Volvo	S80	18.969	.	Passenger	36	2.9	201	109.9	72.1	189.8	3.6

156 rows × 15 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.isnull().sum()
```

	0
Manufacturer	0
Model	0
Sales in thousands	0
4-year resale value	0
Vehicle type	0
Price in thousands	0
Engine size	0
Horsepower	0
Wheelbase	0
Width	0
Length	0
Curb weight	0
Fuel capacity	0
Fuel efficiency	0
Latest Launch	0

dtype: int64

```
df.describe() # all the mathematical,statistical function like mean,meadian,mode and min max.
```

	Sales in thousands
count	156.000000
mean	53.247333
std	68.176559
min	0.110000
25%	14.212750
50%	30.073000
75%	68.069750
max	540.561000

```
# all the columns in the dataset
df.columns
```

```
Index(['Manufacturer', 'Model', 'Sales in thousands', '4-year resale value',
       'Vehicle type', 'Price in thousands', 'Engine size', 'Horsepower',
       'Wheelbase', 'Width', 'Length', 'Curb weight', 'Fuel capacity',
       'Fuel efficiency', 'Latest Launch'],
      dtype='object')
```

```
df['Manufacturer'].value_counts()
```

Manufacturer	count
Dodge	11
Ford	11
Toyota	9
Mercedes-Benz	9
Chevrolet	9
Mitsubishi	7
Nissan	7
Chrysler	7
Oldsmobile	6
Pontiac	6
Lexus	6
Mercury	6
Volkswagen	6
Volvo	6
Cadillac	5
Honda	5
Saturn	5
Plymouth	4
Buick	4
Porsche	3
Acura	3

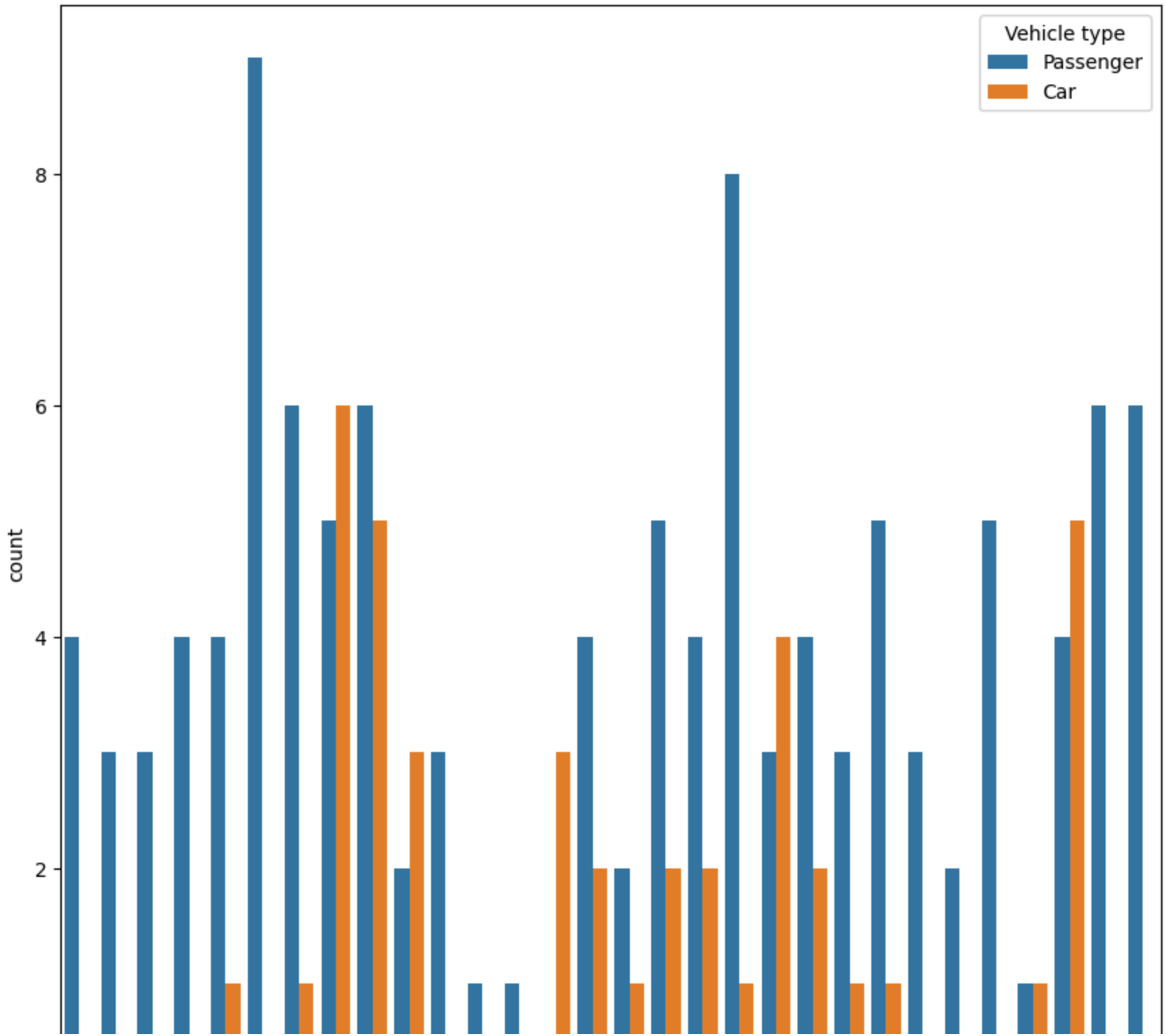
Audi	3
Jeep	3
Hyundai	3
BMW	3
Lincoln	3
Saab	2
Subaru	2
Jaguar	1
Infiniti	1

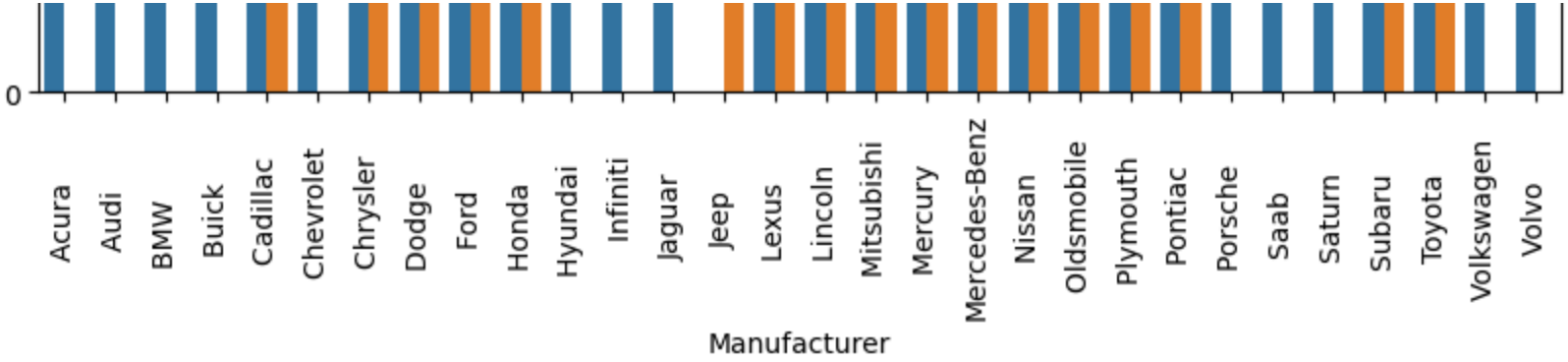
dtype: int64

Start coding or generate with AI.

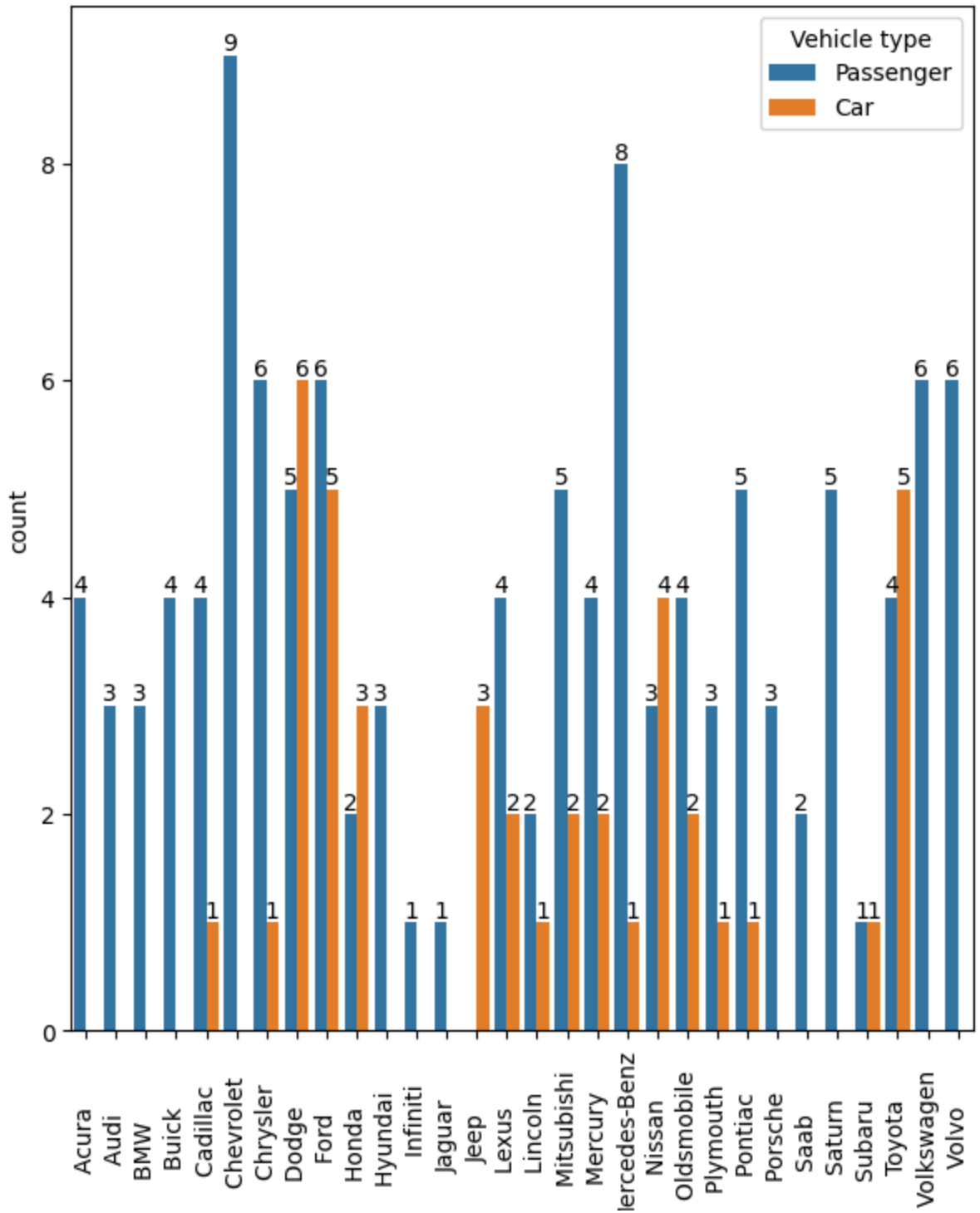
```
#ploting graphs
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 10))  # Set the figure size
ax = sns.countplot(x='Manufacturer', data=df,hue='Vehicle type')  # Plot the count plot
plt.xticks(rotation=90)  # Rotate x-axis labels
plt.show()
for viewvalues in ax.containers:
  ax.bar_label(viewvalues)
```


Vehicle type
Passenger
Car
count
8
6
4
2




```
plt.figure(figsize=(7, 8))  # Set the figure size
ax = sns.countplot(x='Manufacturer', data=df,hue='Vehicle type')  # Plot the count plot
plt.xticks(rotation=90)  # Rotate x-axis labels
for viewvalues in ax.containers:
    ax.bar_label(viewvalues)
```

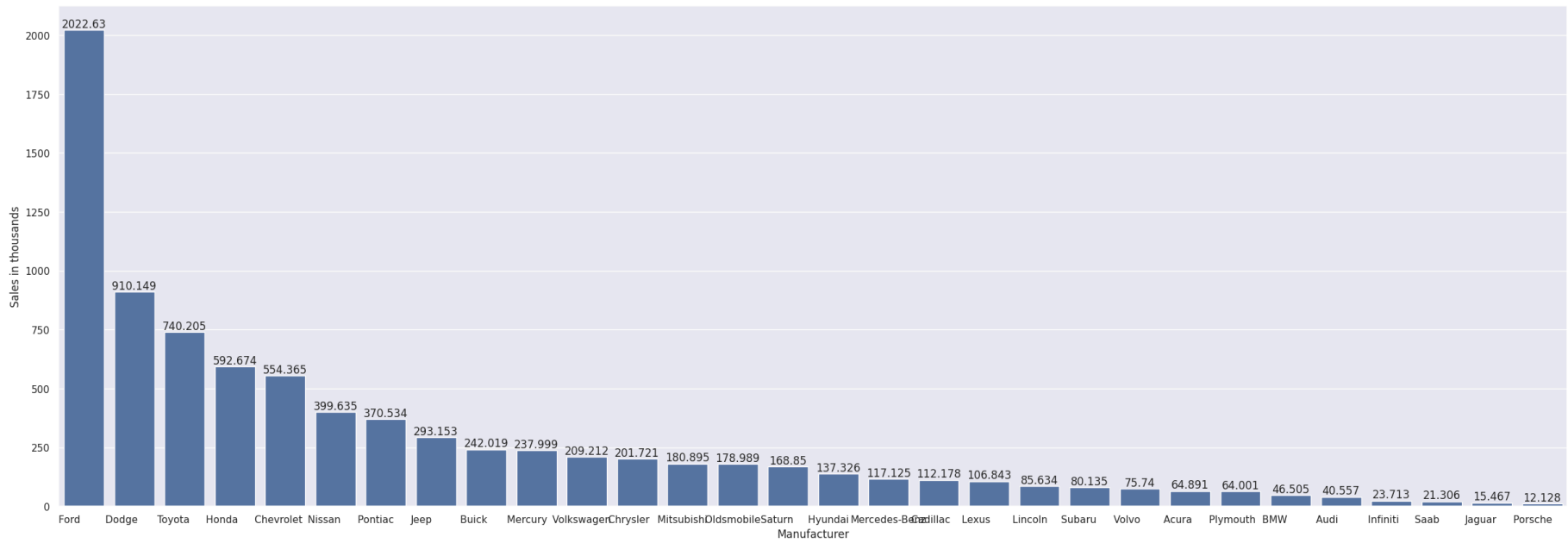

Vehicle type
Passenger
Car
count
0
2
4
6
8
Acura
Audi
BMW
Buick
Cadillac
Chevrolet
Chrysler
Dodge
Ford
Honda
Hyundai
Infiniti
Jaguar
Jeep
Lexus
Lincoln
Mitsubishi
Mercury
Mercedes-Benz
Nissan
Oldsmobile
Plymouth
Pontiac
Porsche
Saab
Saturn
Subaru
Toyota
Volkswagen
Volvo

Manufacturer

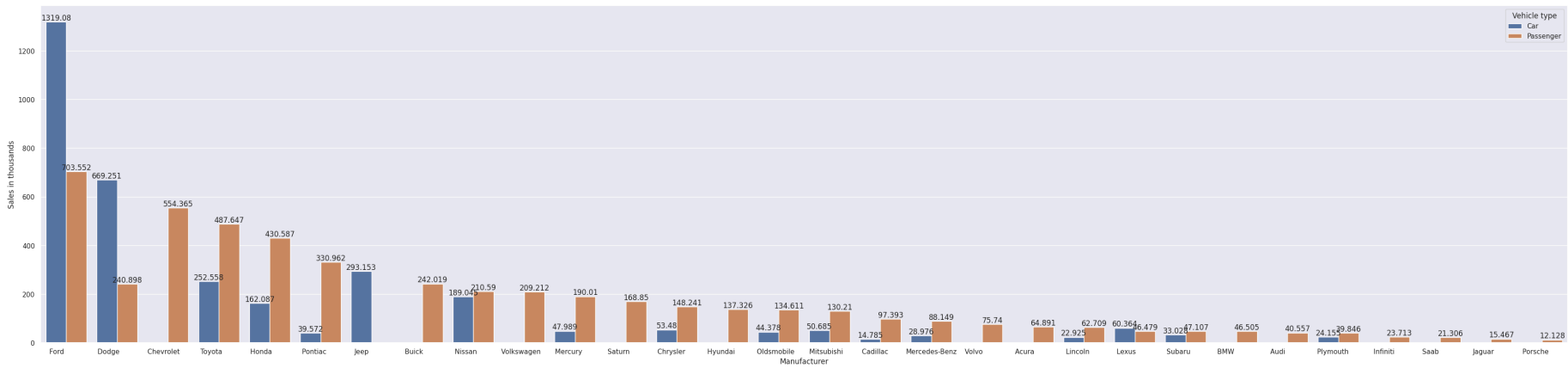
Start coding or generate with AI.

max sales generated by Chevrolet and Mercedes-Benz

```
#
sales_rev=df.groupby(['Manufacturer'], as_index=False)['Sales in thousands'].sum().sort_values(b
sales_rev
sns.set(rc={'figure.figsize': (30,10)})
ax=sns.barplot(x='Manufacturer',y='Sales in thousands',data=sales_rev)
for labels in ax.containers:
  ax.bar_label(labels)
```



```
# sales revenue using vechile type
sales_rev=df.groupby(['Manufacturer','Vehicle type'], as_index=False)['Sales in thousands'].sum(
sales_rev
sns.set(rc={'figure.figsize': (45,10)})
ax=sns.barplot(x='Manufacturer',y='Sales in thousands',data=sales_rev,hue='Vehicle type')
for labels in ax.containers:
  ax.bar_label(labels)
```




```
df.columns
```

```
Index(['Manufacturer', 'Model', 'Sales in thousands', '4-year resale value',
       'Vehicle type', 'Price in thousands', 'Engine size', 'Horsepower',
       'Wheelbase', 'Width', 'Length', 'Curb weight', 'Fuel capacity',
       'Fuel efficiency', 'Latest Launch'],
      dtype='object')
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Sample data processing
sales_rev = df.groupby(['Manufacturer', 'Horsepower'], as_index=False)['Sales in thousands'].su
sales_rev = sales_rev.sort_values(by='Sales in thousands', ascending=False)

# Set figure size and aesthetics
plt.figure(figsize=(15, 8))
sns.set(style="whitegrid")  # Optional styles: "darkgrid", "ticks", etc.

# Create bar plot
ax = sns.barplot(x='Manufacturer', y='Sales in thousands', data=sales_rev, hue='Horsepower', pa

# Add title and labels
plt.title("Sales by Manufacturer and Horsepower", fontsize=16, weight='bold')
plt.xlabel("Manufacturer", fontsize=14)
plt.ylabel("Sales in Thousands", fontsize=14)

# Rotate x-axis labels and add data labels
plt.xticks(rotation=45)
for container in ax.containers:
    ax.bar_label(container, fmt='%.1f')

# Add a legend and customize grid
plt.legend(title="Horsepower")
plt.grid(visible=True, linestyle="--", alpha=0.7)

# Show the plot
plt.show()
```