

Project Completion and Certification

This project demonstrates the successful completion of a Udemy course(*Python Project for Basics Data Analysis*) focused on analyzing a car sales dataset. By completing this project, a certificate was earned from Udemy, signifying the acquisition of skills in data analysis techniques using Python. The project involved exploring sales trends, identifying popular car models, and uncovering factors that influence car purchasing decisions.

Goals for analyzing this data?

Some common goals with car sales data might be:

Finding the most popular car models or brands.

Analyzing sales trends over time. Comparing sales performance across different vehicle types. Identifying factors that influence sales (like price, fuel efficiency, etc.).

{1}Importing Libraries

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

{2} Loading The Dataset

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

	Manufacturer	Model	Sales in thousands	4-year resale value	Vehicle type	Price in thousands	Engine size	Horsepower	Wheelbase	Width	Length
0	Acura	Integra	16.919	16.36	Passenger	21.5	1.8	140	101.2	67.3	172.
1	Acura	TL	39.384	19.875	Passenger	28.4	3.2	225	108.1	70.3	192.
2	Acura	CL	NaN	NaN	Passenger	.	3.2	225	106.9	70.6	19

Next steps:

[Generate code with df](#)



[View recommended plots](#)

{3}Exploratory Data Analysis & Data Cleaning

```
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
```

```
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')
```

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

```
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
```

This will drop rows with any missing values

```
df.dropna(inplace=True)
pd.isnull(df).sum()
```

```
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
```

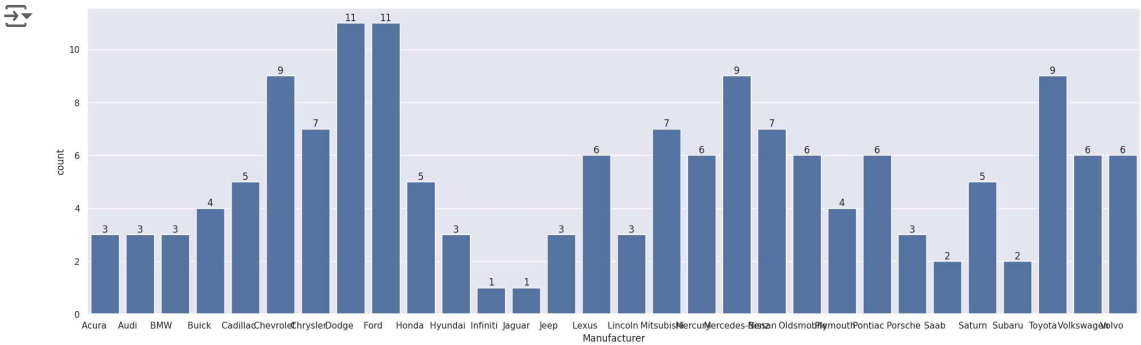
Double-click (or enter) to edit

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

```
Manufacturer
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
Name: count, dtype: int64
```

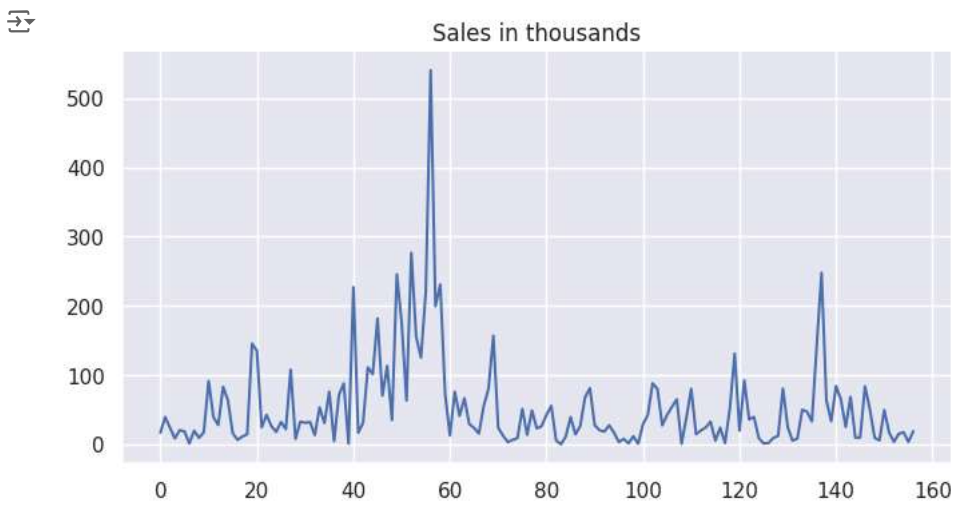
▼ Suitable Visualization

```
ax=sns.countplot(x="Manufacturer",data=df)
for bar in ax.containers:
    ax.bar_label(bar)
```



Maximum sale by dodge and Ford This bar chart illustrates the frequency distribution of car manufacturers within the dataset. Each bar represents a unique manufacturer, with the height corresponding to the number of cars associated with that brand

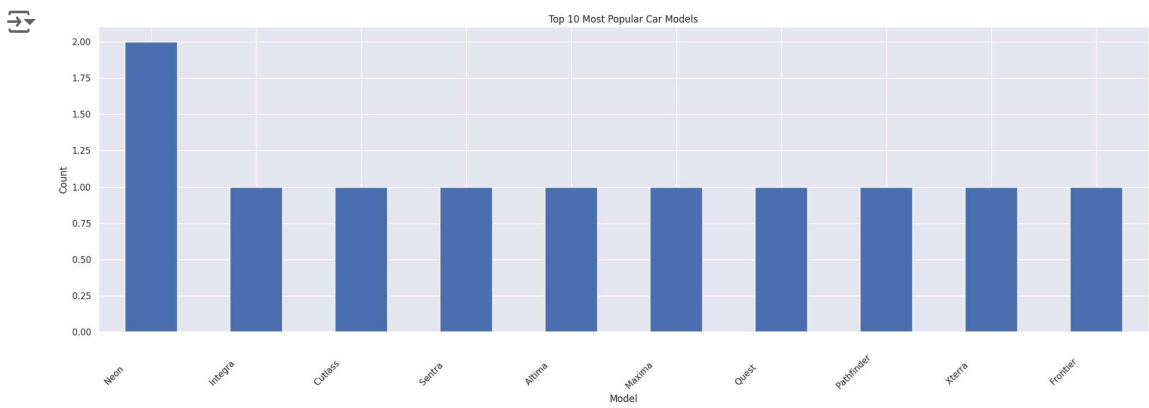
```
from matplotlib import pyplot as plt
df['Sales in thousands'].plot(kind='line', figsize=(8, 4), title='Sales in thousands')
plt.gca().spines[['top', 'right']].set_visible(False)
```



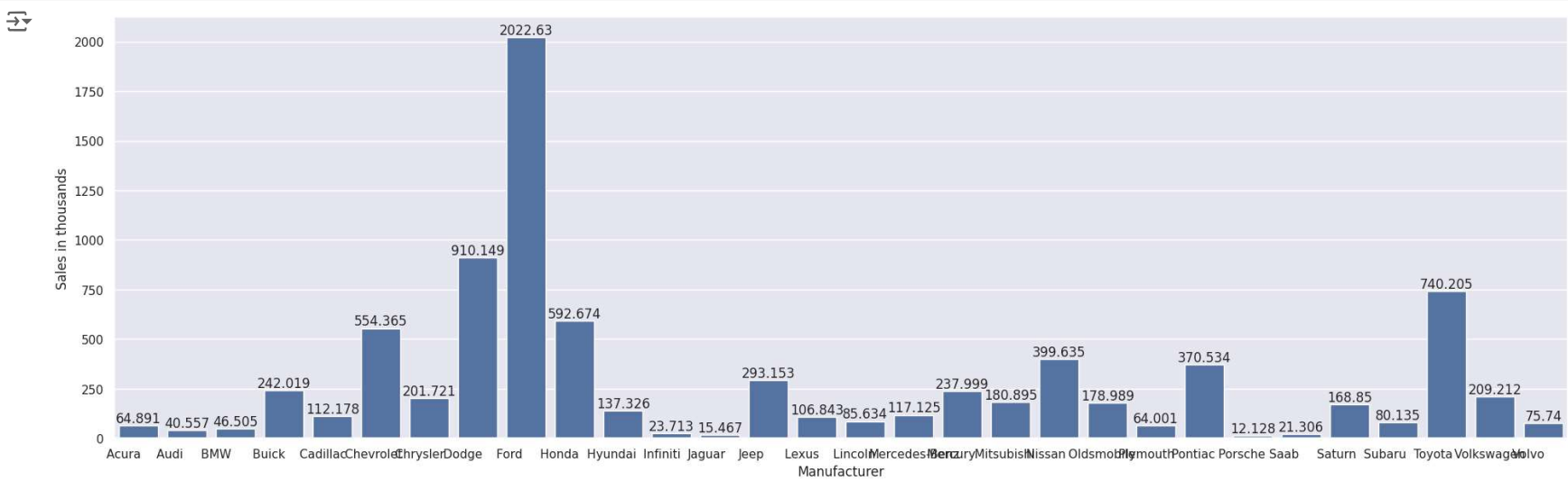
A simple line plot that visualizes the trend of car sales over time or some other index (depending on how your data is structured). The x-axis likely represents time periods or some other sequential data, while the y-axis shows the corresponding sales figures in thousands.

4. Top 10 Most Popular Models:

```
top_models = df["Model"].value_counts().head(10)
top_models.plot(kind="bar")
plt.title("Top 10 Most Popular Car Models")
plt.xlabel("Model")
plt.ylabel("Count")
plt.xticks(rotation=45, ha='right')
plt.show()
```



```
gen_graph= df.groupby(['Manufacturer'], as_index=False)['Sales in thousands'].sum()
ax=sns.barplot(x="Manufacturer",y="Sales in thousands",data=gen_graph)
for bar in ax.containers:
    ax.bar_label(bar)
```

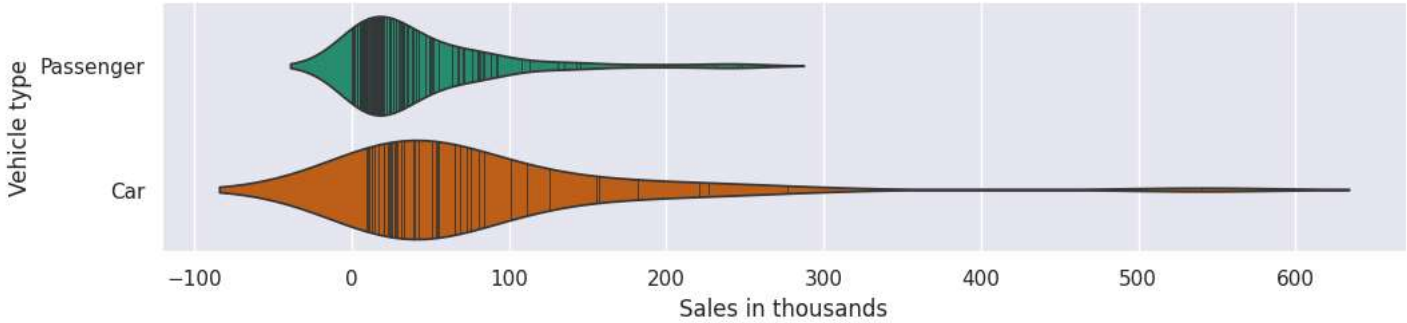


This visualization will help you understand the total sales for each car manufacturer.

```
from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(df['Vehicle type'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(df, x='Sales in thousands', y='Vehicle type', inner='stick', palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

```
<ipython-input-51-cbaf612c41e6>:5: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

sns.violinplot(df, x='Sales in thousands', y='Vehicle type', inner='stick', palette='Dark2')
```



A violin plot showing the distribution of car sales across different vehicle types. It allows you to compare sales patterns, identify outliers, and spot potential trends in the data. This visualization helps understand how sales figures vary across different categories of vehicles.