Assignment2

December 14, 2025

[6]:
```
# DATASET LOADING...
```

[9]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[13]:
```
df = pd.read_csv('Car Sales.xlsx - car_data.csv')
df
```

[13]:

	Car_id	Date	Customer Name	Gender	Annual Income	\
0	C_CND_000001	1/2/2022	Geraldine	Male	13500	
1	C_CND_000002	1/2/2022	Gia	Male	1480000	
2	C_CND_000003	1/2/2022	Gianna	Male	1035000	
3	C_CND_000004	1/2/2022	Giselle	Male	13500	
4	C_CND_000005	1/2/2022	Grace	Male	1465000	
…	…	…	…	…	…	
23901	C_CND_023902	12/31/2023	Martin	Male	13500	
23902	C_CND_023903	12/31/2023	Jimmy	Female	900000	
23903	C_CND_023904	12/31/2023	Emma	Male	705000	
23904	C_CND_023905	12/31/2023	Victoire	Male	13500	
23905	C_CND_023906	12/31/2023	Donovan	Male	1225000	

	Dealer_Name	Company	Model	\
0	Buddy Storbeck's Diesel Service Inc	Ford	Expedition	
1	C & M Motors Inc	Dodge	Durango	
2	Capitol KIA	Cadillac	Eldorado	
3	Chrysler of Tri-Cities	Toyota	Celica	
4	Chrysler Plymouth	Acura	TL	
…	…	…	…	
23901	C & M Motors Inc	Plymouth	Voyager	
23902	Ryder Truck Rental and Leasing	Chevrolet	Prizm	
23903	Chrysler of Tri-Cities	BMW	328i	
23904	Chrysler Plymouth	Chevrolet	Metro	
23905	Pars Auto Sales	Lexus	ES300	

	Engine	Transmission	Color	Price ($)	\
0	DoubleÃ Overhead Camshaft	Auto	Black	26000	

```
1      DoubleÂ Overhead Camshaft         Auto       Black      19000
2              Overhead Camshaft       Manual         Red      31500
3              Overhead Camshaft       Manual  Pale White      14000
4      DoubleÂ Overhead Camshaft         Auto         Red      24500
…                            …          …         …        …
23901          Overhead Camshaft       Manual         Red      12000
23902  DoubleÂ Overhead Camshaft         Auto       Black      16000
23903          Overhead Camshaft       Manual         Red      21000
23904  DoubleÂ Overhead Camshaft         Auto       Black      31000
23905  DoubleÂ Overhead Camshaft         Auto  Pale White      27500

       Dealer_No  Body Style    Phone Dealer_Region
0      06457-3834        SUV  8264678    Middletown
1      60504-7114        SUV  6848189        Aurora
2      38701-8047  Passenger  7298798    Greenville
3      99301-3882        SUV  6257557         Pasco
4      53546-9427  Hatchback  7081483    Janesville
…           …        …      …              …
23901  60504-7114  Passenger  8583598         Pasco
23902  06457-3834    Hardtop  7914229    Middletown
23903  99301-3882      Sedan  7659127    Scottsdale
23904  53546-9427  Passenger  6030764        Austin
23905  38701-8047    Hardtop  7020564    Middletown

[23906 rows x 16 columns]
```

[ ]:
```
# FIRST 5 ROWS OF THE DATASET...
```

[11]:
```
df.head()
```

[11]:
```
         Car_id      Date Customer Name Gender  Annual Income  \
0  C_CND_000001  1/2/2022     Geraldine   Male          13500
1  C_CND_000002  1/2/2022           Gia   Male        1480000
2  C_CND_000003  1/2/2022        Gianna   Male        1035000
3  C_CND_000004  1/2/2022       Giselle   Male          13500
4  C_CND_000005  1/2/2022         Grace   Male        1465000

                          Dealer_Name   Company       Model  \
0  Buddy Storbeck's Diesel Service Inc      Ford  Expedition
1                   C & M Motors Inc     Dodge     Durango
2                       Capitol KIA  Cadillac    Eldorado
3              Chrysler of Tri-Cities    Toyota      Celica
4                   Chrysler Plymouth     Acura          TL

                     Engine Transmission       Color  Price ($)  Dealer_No   \
0  DoubleÂ Overhead Camshaft         Auto       Black      26000  06457-3834
1  DoubleÂ Overhead Camshaft         Auto       Black      19000  60504-7114
```

```
2          Overhead Camshaft       Manual         Red      31500  38701-8047
3          Overhead Camshaft       Manual  Pale White      14000  99301-3882
4  DoubleÃ Overhead Camshaft         Auto         Red      24500  53546-9427

  Body Style    Phone Dealer_Region
0        SUV  8264678    Middletown
1        SUV  6848189        Aurora
2  Passenger  7298798    Greenville
3        SUV  6257557         Pasco
4  Hatchback  7081483    Janesville
```

[ ]:
```
# LAST 5 ROWS OF THE DATASET...
```

[12]:
```
df.tail()
```

[12]:
```
             Car_id        Date Customer Name  Gender  Annual Income  \
23901  C_CND_023902  12/31/2023        Martin    Male          13500
23902  C_CND_023903  12/31/2023         Jimmy  Female         900000
23903  C_CND_023904  12/31/2023          Emma    Male         705000
23904  C_CND_023905  12/31/2023      Victoire    Male          13500
23905  C_CND_023906  12/31/2023       Donovan    Male        1225000

                         Dealer_Name    Company    Model  \
23901                C & M Motors Inc   Plymouth  Voyager
23902  Ryder Truck Rental and Leasing  Chevrolet    Prizm
23903          Chrysler of Tri-Cities        BMW     328i
23904               Chrysler Plymouth  Chevrolet    Metro
23905                 Pars Auto Sales      Lexus    ES300

                         Engine Transmission       Color  Price ($)  \
23901          Overhead Camshaft       Manual         Red      12000
23902  DoubleÃ Overhead Camshaft         Auto       Black      16000
23903          Overhead Camshaft       Manual         Red      21000
23904  DoubleÃ Overhead Camshaft         Auto       Black      31000
23905  DoubleÃ Overhead Camshaft         Auto  Pale White      27500

       Dealer_No  Body Style    Phone Dealer_Region
23901  60504-7114  Passenger  8583598         Pasco
23902  06457-3834    Hardtop  7914229    Middletown
23903  99301-3882      Sedan  7659127    Scottsdale
23904  53546-9427  Passenger  6030764        Austin
23905  38701-8047    Hardtop  7020564    Middletown
```

[ ]:
```
# SHAPE OF DATASET...
```

[12]:
```
df.shape
```

[12]:
```
(23906, 16)
```

[ ]:
```
# COLUMN NAMES...
```

[13]:
```
df.columns
```

[13]:
```
Index(['Car_id', 'Date', 'Customer Name', 'Gender', 'Annual Income',
       'Dealer_Name', 'Company', 'Model', 'Engine', 'Transmission', 'Color',
       'Price ($)', 'Dealer_No ', 'Body Style', 'Phone', 'Dealer_Region'],
      dtype='object')
```

[ ]:
```
# DATA TYPES USING INFO()...
```

[14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23906 entries, 0 to 23905
Data columns (total 16 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Car_id         23906 non-null  object
 1   Date           23906 non-null  object
 2   Customer Name  23905 non-null  object
 3   Gender         23906 non-null  object
 4   Annual Income  23906 non-null  int64 
 5   Dealer_Name    23906 non-null  object
 6   Company        23906 non-null  object
 7   Model          23906 non-null  object
 8   Engine         23906 non-null  object
 9   Transmission   23906 non-null  object
 10  Color          23906 non-null  object
 11  Price ($)      23906 non-null  int64 
 12  Dealer_No      23906 non-null  object
 13  Body Style     23906 non-null  object
 14  Phone          23906 non-null  int64 
 15  Dealer_Region  23906 non-null  object
dtypes: int64(3), object(13)
memory usage: 2.9+ MB
```

[ ]:
```
# DATA CLEANING AND PREPROCESSING...
```

[15]:
```
df.isnull().sum()
```

[15]:
```
Car_id           0
Date             0
Customer Name    1
Gender           0
Annual Income    0
```

```
Dealer_Name       0
Company           0
Model             0
Engine            0
Transmission      0
Color             0
Price ($)         0
Dealer_No         0
Body Style        0
Phone             0
Dealer_Region     0
dtype: int64
```

[1]:
```
# CHECK AND REMOVE DUPLICATE RECORDS...
```

[14]:
```
df.duplicated().sum()
```

[14]:
```
np.int64(0)
```

[15]:
```
df.drop_duplicates(inplace=True)
```

[16]:
```
# Convert date columns to datetime...
```

[17]:
```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
```

[18]:
```
# Convert numerical columns to int/float
```

[19]:
```
df['Annual Income'] = df['Annual Income'].astype(int)
df['Price ($)'] = df['Price ($)'].astype(float)
```

[20]:
```
#Fix casing issues (upper/lower/title case)...
```

[21]:
```
text_cols = ['Customer Name', 'Dealer_Name', 'Company', 'Model',
             'Engine', 'Transmission', 'Color', 'Body Style', 'Dealer_Region']

df[text_cols] = df[text_cols].apply(lambda x: x.str.strip().str.title())
```

[22]:
```
#Summary statistics using describe()
```

[23]:
```
df.describe()
```

[23]:
```
                                Date  Annual Income     Price ($)  \
count                          23906   2.390600e+04  23906.000000
mean   2023-03-01 14:28:10.822387456   8.308403e+05  28090.247846
min              2022-01-02 00:00:00   1.008000e+04   1200.000000
25%              2022-09-20 00:00:00   3.860000e+05  18001.000000
50%              2023-03-13 00:00:00   7.350000e+05  23000.000000
75%              2023-09-08 00:00:00   1.175750e+06  34000.000000
```

```
max              2023-12-31 00:00:00   1.120000e+07  85800.000000
std                              NaN   7.200064e+05  14788.687608

              Phone
count  2.390600e+04
mean   7.497741e+06
min    6.000101e+06
25%    6.746495e+06
50%    7.496198e+06
75%    8.248146e+06
max    8.999579e+06
std    8.674920e+05
```

[24]:
```
# Value counts for categorical columns...
```

[25]:
```
df['Body Style'].value_counts()
```

[25]:
```
Body Style
Suv          6374
Hatchback    6128
Sedan        4488
Passenger    3945
Hardtop      2971
Name: count, dtype: int64
```

[26]:
```
# Group-by analysis (e.g., average, total, count)...
```

[27]:
```
df.groupby('Company')['Price ($)'].mean().sort_values(ascending=False)
```

[27]:
```
Company
Cadillac      40972.093558
Saab          36516.338095
Lexus         34024.567332
Buick         33634.362187
Oldsmobile    31894.250225
Lincoln       31407.036585
Saturn        31092.609215
Toyota        29513.120721
Plymouth      29404.980551
Pontiac       29358.300251
Infiniti      29318.153846
Ford          29263.682156
Mercury       28535.163616
Honda         28082.959040
Subaru        27931.340741
Volvo         27788.593156
Nissan        27047.511287
```

```
Mercedes-B    26944.842802
Mitsubishi    26673.818324
Dodge         26406.341113
Chevrolet     26198.606377
Chrysler      26019.529464
Volkswagen    25568.552888
Jaguar        25138.194444
Bmw           25090.622785
Acura         24758.561684
Audi          22851.790598
Porsche       22674.894737
Jeep          21057.338843
Hyundai       19386.234848
Name: Price ($), dtype: float64
```

[28]:
```
# Identify top or bottom performing categories
```

[29]:
```
df.groupby('Dealer_Region')['Price ($)'].sum().sort_values(ascending=False)
```

[29]:
```
Dealer_Region
Austin        117192531.0
Janesville    106351234.0
Scottsdale     95969374.0
Aurora         88687382.0
Greenville     88149602.0
Pasco          88040714.0
Middletown     87134628.0
Name: Price ($), dtype: float64
```

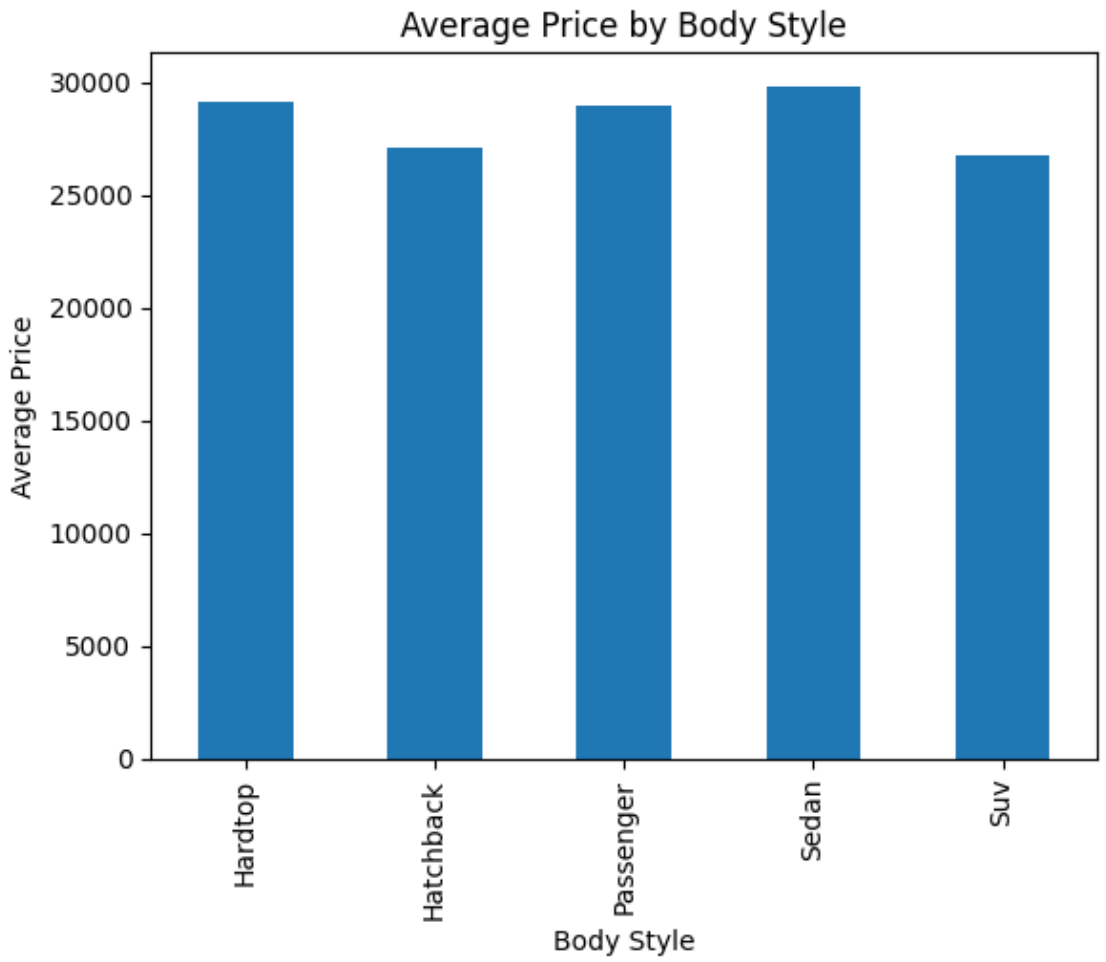
[30]:
```
# Correlation analysis between numerical columns
```

[31]:
```
df[['Annual Income', 'Price ($)']].corr()
```

[31]:
```
                Annual Income  Price ($)
Annual Income        1.000000   0.012065
Price ($)            0.012065   1.000000
```

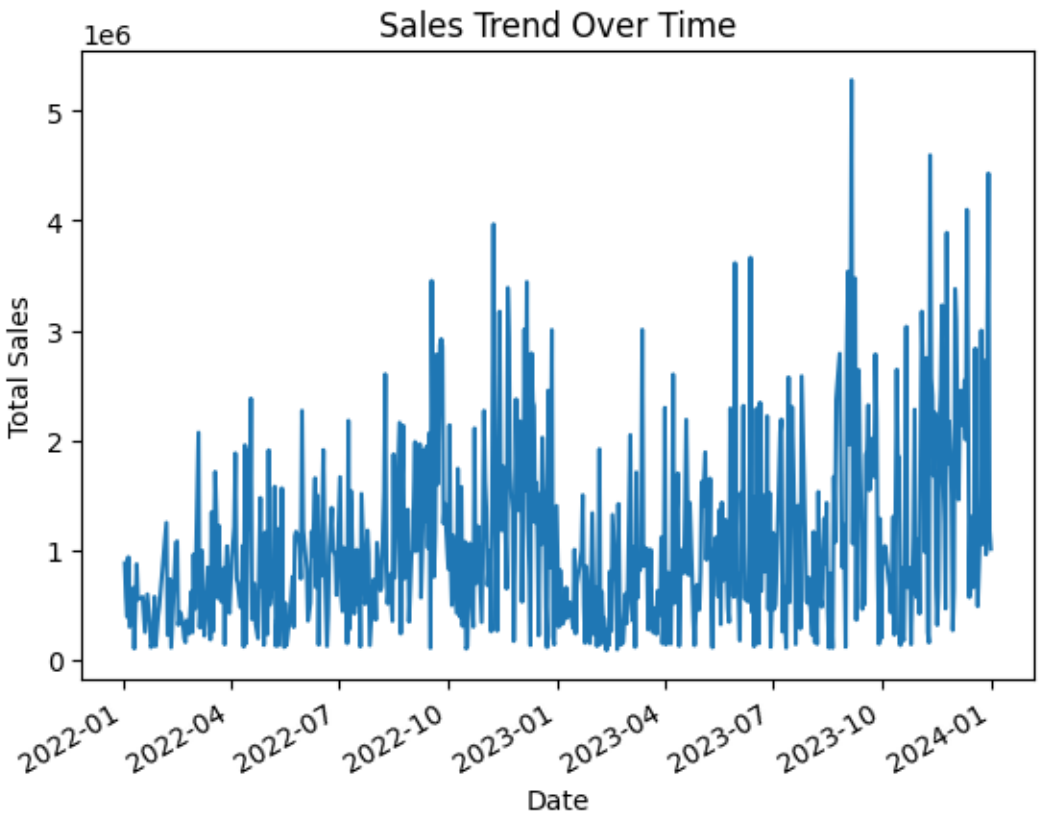
[32]:
```
#Bar Chart - Category-wise comparison...
```

[33]:
```
df.groupby('Body Style')['Price ($)'].mean().plot(kind='bar', title='Average␣
  ↪Price by Body Style')
plt.xlabel('Body Style')
plt.ylabel('Average Price')
plt.show()
```



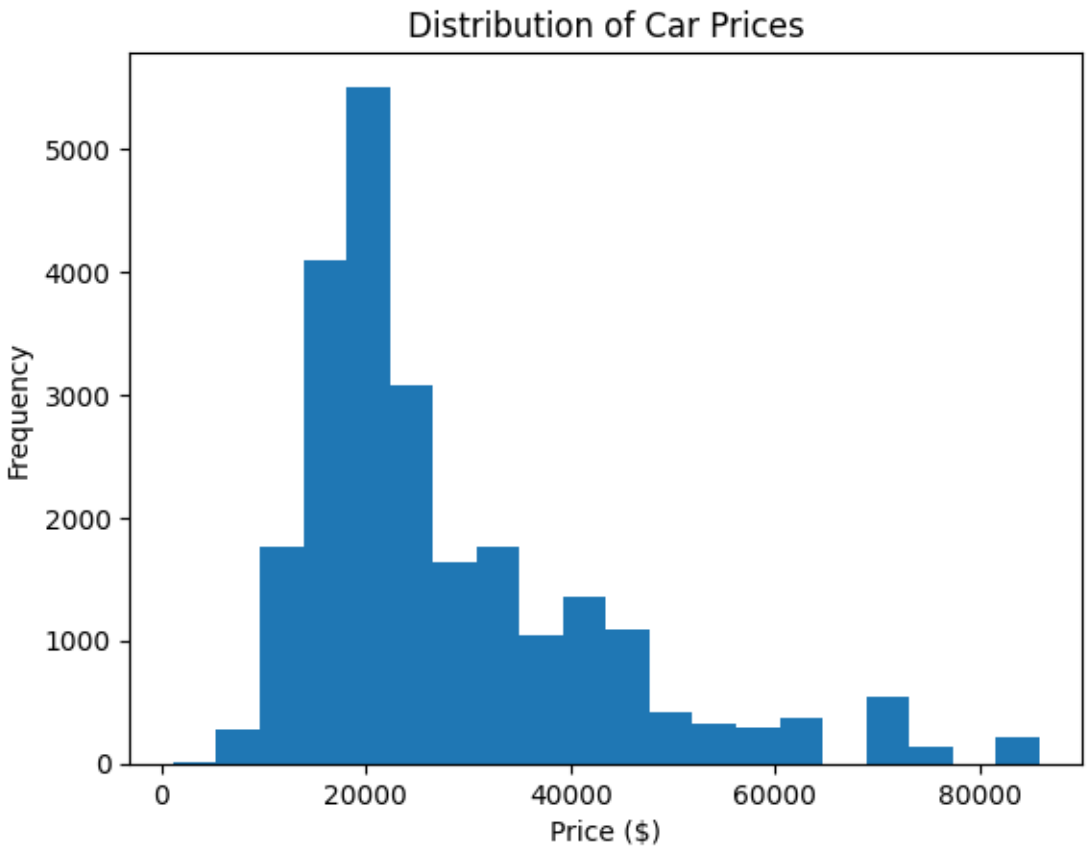
[34]:
```
# Line Chart - Trend analysis
```

[35]:
```
df.groupby('Date')['Price ($)'].sum().plot(title='Sales Trend Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.show()
```



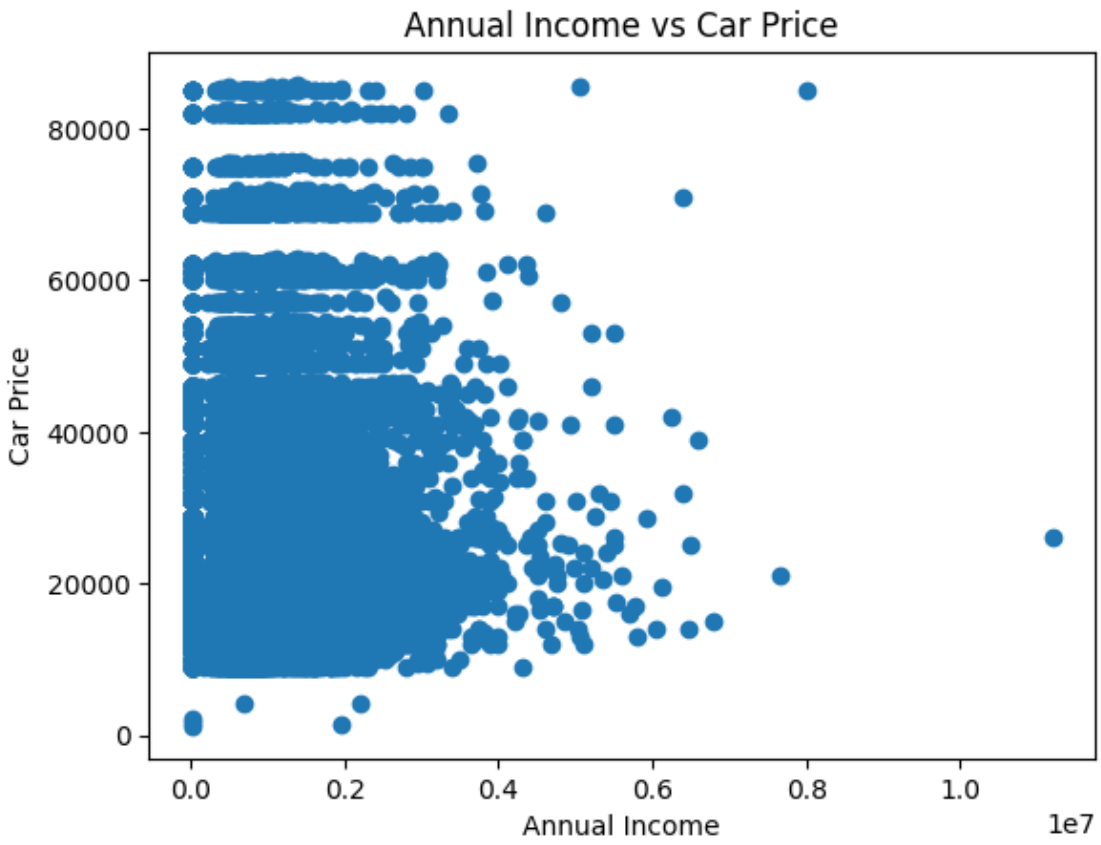
[36]:
```
# Histogram - Distribution of a numerical column
```

[37]:
```
plt.hist(df['Price ($)'], bins=20)
plt.title('Distribution of Car Prices')
plt.xlabel('Price ($)')
plt.ylabel('Frequency')
plt.show()
```



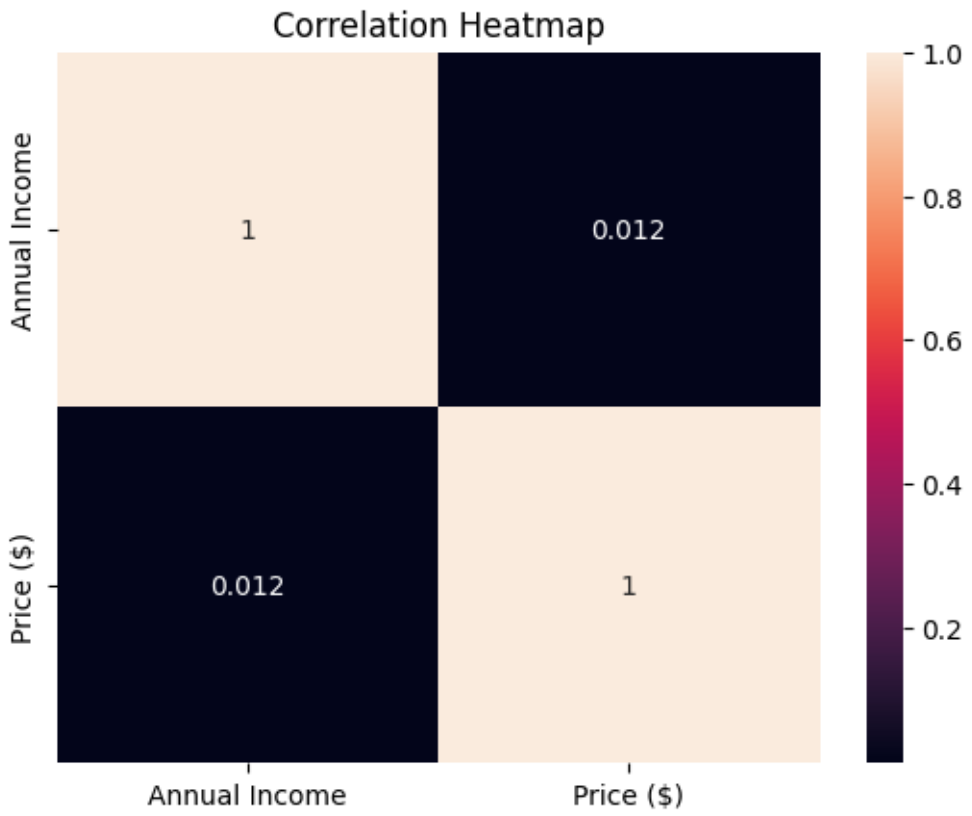
[38]:
```
# Scatter Plot - Relationship between two variables
```

[39]:
```
plt.scatter(df['Annual Income'], df['Price ($)'])
plt.title('Annual Income vs Car Price')
plt.xlabel('Annual Income')
plt.ylabel('Car Price')
plt.show()
```



[40]:
```
# Heatmap - Correlation between numerical columns
```

[32]:
```
import seaborn as sns
sns.heatmap(df[['Annual Income', 'Price ($)']].corr(), annot=True)
plt.title('Correlation Heatmap')
plt.show()
```



[ ]:
```
Insight1:-SUVs dominate the market in terms of sales volume, indicating strong␣
↪consumer preference for larger and utility-focused vehicles.

Insight2:-Customers with higher annual income tend to purchase higher-priced␣
↪vehicles, showing a clear relationship between purchasing power and car␣
↪pricing.

Insight3:-Luxury brands such as Cadillac and Acura command significantly higher␣
↪average prices, contributing more revenue per sale despite lower volume.

Insight4:-Sales trends over time show consistent demand with noticeable peaks,␣
↪suggesting possible seasonal buying behavior.

Insight5:-Some dealer regions contribute disproportionately to total sales␣
↪revenue, highlighting opportunities for region-focused marketing and␣
↪inventory planning.
```