REPORT ON CAR SALES DATA ANALYSIS

BIG DATA ANALYTICS

3325 SHARON PHILIP
3326 AKASH SINGH

INTRODUCTION

Analyzing car sales data is a crucial aspect of market research and strategic planning within the automotive industry. This process involves examining various metrics and trends related to vehicle purchases, including sales volumes, market share, consumer preferences, and regional variations.

Car sales data analysis serves as a powerful tool for decision-making and performance evaluation across the automotive sector. For manufacturers, it provides insights into which car models are resonating most with consumers, allowing for informed decisions on production planning, inventory management, and product development. Dealerships benefit from sales data analysis by understanding local market trends, identifying sales opportunities, and optimizing their inventory to meet customer demands effectively.

About the Data:

Dataset contain the sales data of different products in India. This data has been taken from the Kaggle dataset. Dataset contain information regarding the customer name,country , price, annual income ,gender ,model , and Number of orders per userId.

This analysis will be a exploratory data analysis and we will try to figure out some important insights from the data in order to make a data driven decision.

This dataset is taken from kaggle open source data
https://www.kaggle.com/code/kareemasg/car-sales-report

Dataset contain 11,251 rows and 16 columns. This is a big data but we can not generalize the concept for whole population but will try to figure out some strong insights that could help the users of this report to increase their sales.

Important questions to analyse:

1. Which gender exhibit the strongest affinity for buying car ?
2. Does transmission effect decisions for buying car ?
3. Which country present the most favorable advertising opportunities for company X to maximize car sales?
4. What car sales are most sought after by customers ?

We have used the Google Colab platform We have used the EDA (Exploratory Data Analysis)

algorithm for data analysis.

EDA Algorithm:

In the context of data analysis, EDA stands for Exploratory Data Analysis. It is an approach used to analyze datasets and understand their main characteristics, often before applying more complex statistical techniques or machine learning algorithms. EDA involves summarizing the main features of the data, often through visual methods, to gain insights and identify patterns or anomalies.

Key steps in EDA typically include:

1. Data Cleaning: Identifying and handling missing values, outliers, and inconsistencies in the dataset.

2. Summary Statistics: Calculating and examining basic descriptive statistics such as mean, median, mode, standard deviation, and range for numerical variables.

3. Data Visualization: Creating plots and charts to visualize the distribution, relationships, and patterns within the data. Common visualization techniques include histograms, box plots, scatter plots, and correlation matrices.

4. Dimensionality Reduction: Techniques such as PCA (Principal Component Analysis) or t-SNE (t-distributed Stochastic Neighbor Embedding) may be used to reduce the dimensionality of the dataset and visualize high-dimensional data in lower-dimensional space.

5. Hypothesis Testing: Performing statistical tests to validate assumptions or hypotheses about the data.

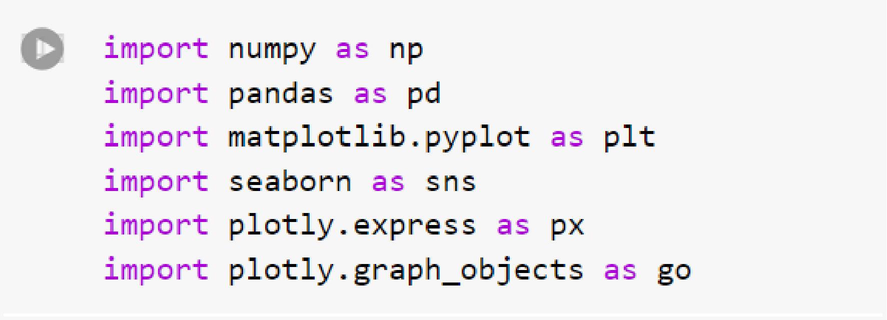
Overall, EDA helps analysts and data scientists understand the structure and characteristics of the data, identify potential issues or trends, and inform subsequent analysis or modeling decisions. It plays a crucial role in the data exploration process and is often the first step in any data analysis project.

PROCESSING THE DATA

Import required libraries

We will be using following libraries in order to clean and analyze the data.

- Pandas
- Numpy
- Matplotlib
- Seaborn
- Operating System (OS)



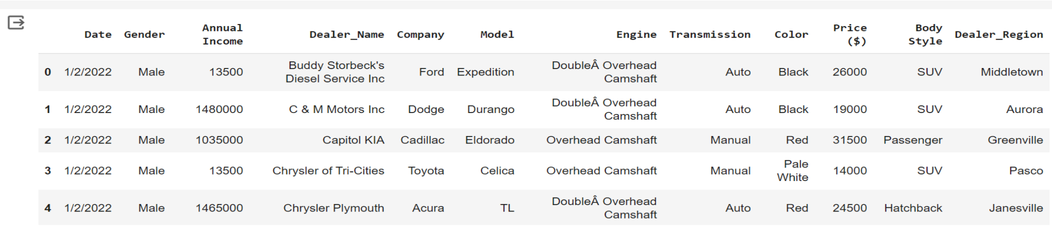
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
```

```
df=pd.read_csv('/content/Car Sales.xlsx - car_data (1).csv')
```

df.shape

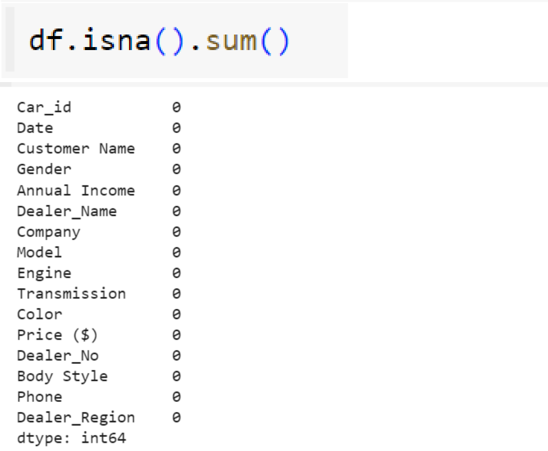
```
(23906, 12)
```

df.head()



	Date	Gender	Annual Income	Dealer_Name	Company	Model	Engine	Transmission	Color	Price ($)	Body Style	Dealer_Region
0	1/2/2022	Male	13500	Buddy Storbeck's Diesel Service Inc	Ford	Expedition	DoubleÂ Overhead Camshaft	Auto	Black	26000	SUV	Middletown
1	1/2/2022	Male	1480000	C & M Motors Inc	Dodge	Durango	DoubleÂ Overhead Camshaft	Auto	Black	19000	SUV	Aurora
2	1/2/2022	Male	1035000	Capitol KIA	Cadillac	Eldorado	Overhead Camshaft	Manual	Red	31500	Passenger	Greenville
3	1/2/2022	Male	13500	Chrysler of Tri-Cities	Toyota	Celica	Overhead Camshaft	Manual	Pale White	14000	SUV	Pasco
4	1/2/2022	Male	1465000	Chrysler Plymouth	Acura	TL	DoubleÂ Overhead Camshaft	Auto	Red	24500	Hatchback	Janesville

```
#drop unrelated/blank columns
df=df.drop(columns=['Car_id','Customer Name','Dealer_No
','Phone'],axis=1)
#check for null values
```



```
df.isna().sum()
```

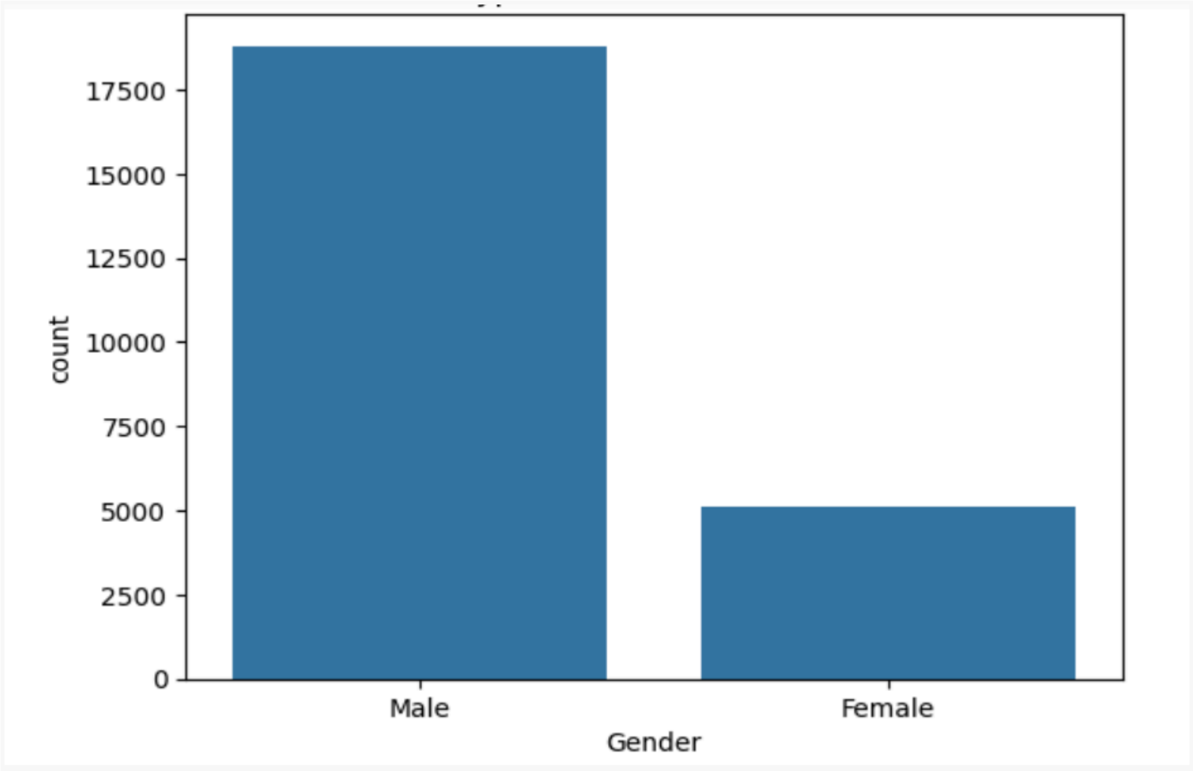
```
Car_id           0
Date             0
Customer Name    0
Gender           0
Annual Income    0
Dealer_Name      0
Company          0
Model            0
Engine           0
Transmission     0
Color            0
Price ($)        0
Dealer_No        0
Body Style       0
Phone            0
Dealer_Region    0
dtype: int64
```

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

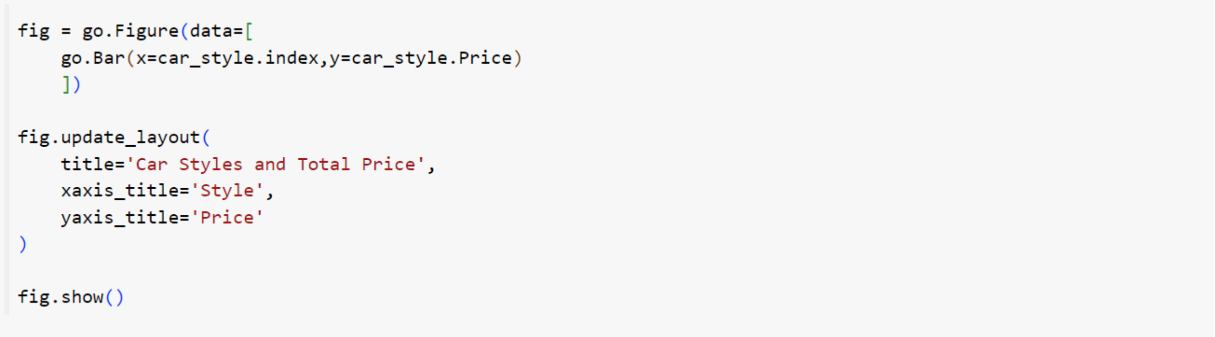
Data Analysis:

- **Gender:**

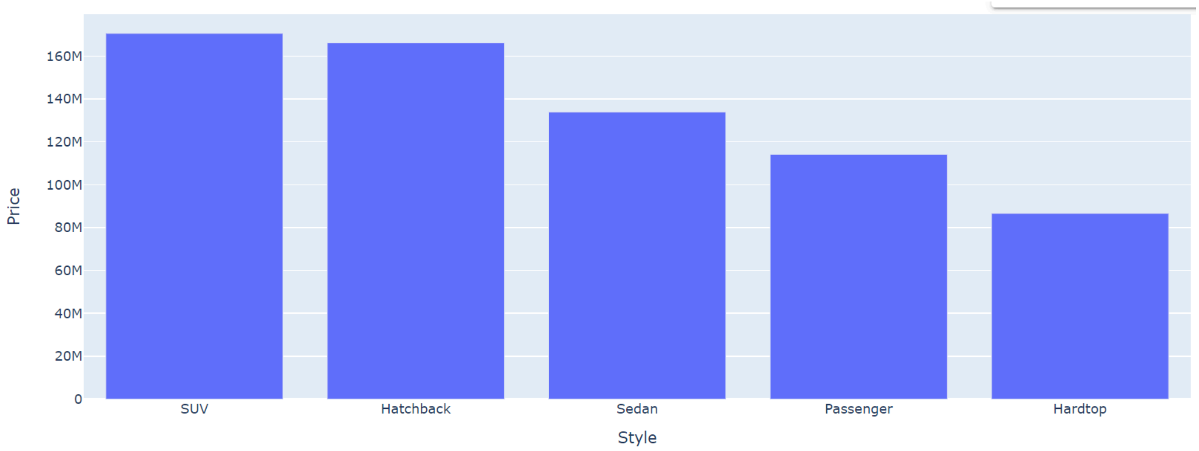
```
# plotting a bar chart for Gender and it's count
  sns.countplot(data=df,x='Gender')
  plt.title('Type of Gender and num')
```



# plotting a bar chart for cars styles and total price

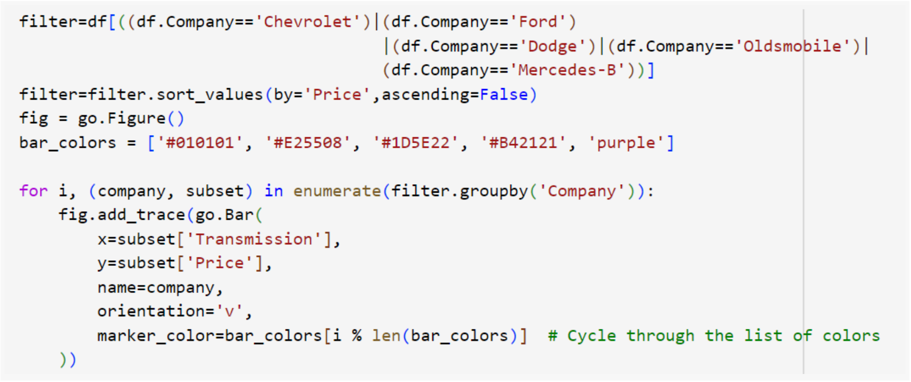


```
fig = go.Figure(data=[
    go.Bar(x=car_style.index,y=car_style.Price)
    ])

fig.update_layout(
    title='Car Styles and Total Price',
    xaxis_title='Style',
    yaxis_title='Price'
)

fig.show()
```

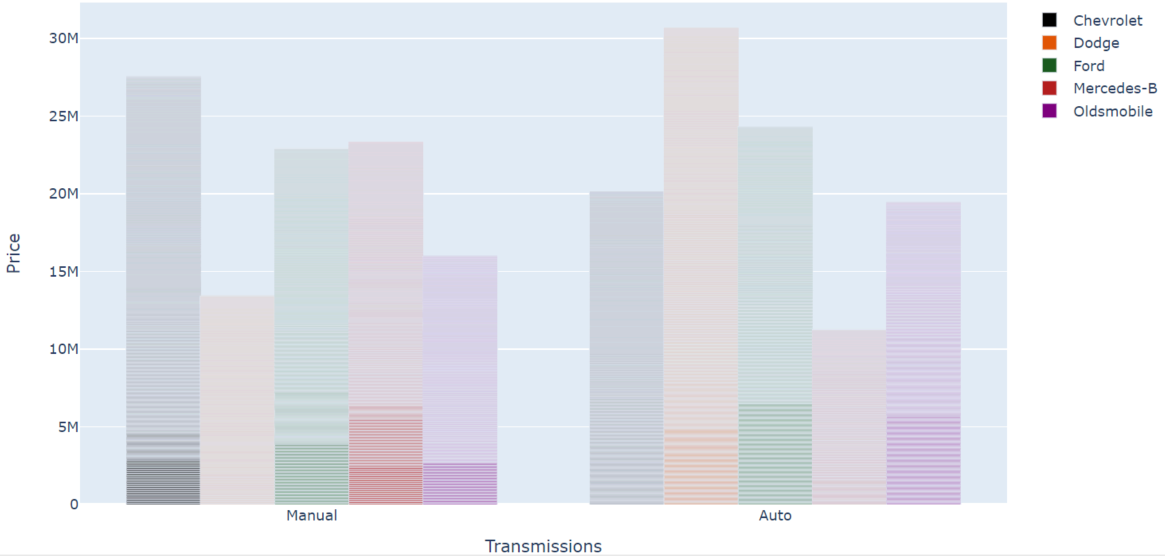


From above graphs we can see that most of the sales are of SUVs

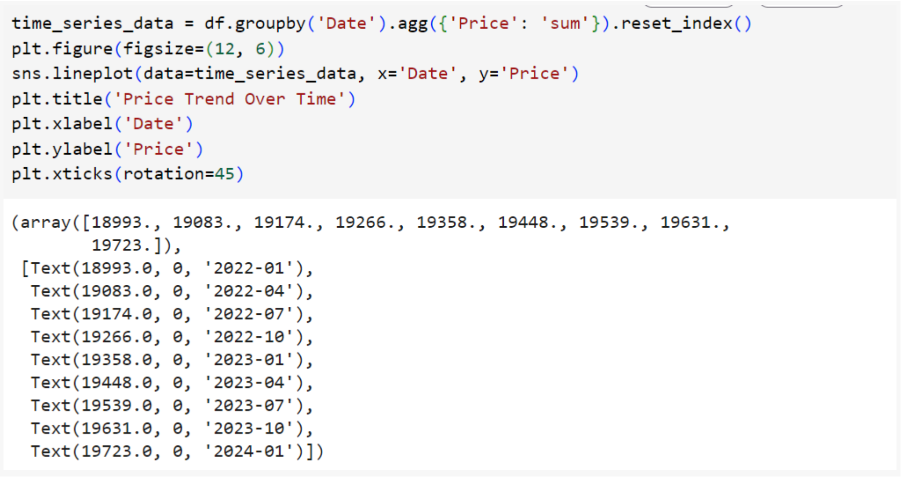
TRANSMISSION:



```
filter=df[((df.Company=='Chevrolet')|(df.Company=='Ford')
                                     |(df.Company=='Dodge')|(df.Company=='Oldsmobile')|
                                     (df.Company=='Mercedes-B'))]
filter=filter.sort_values(by='Price',ascending=False)
fig = go.Figure()
bar_colors = ['#010101', '#E25508', '#1D5E22', '#B42121', 'purple']

for i, (company, subset) in enumerate(filter.groupby('Company')):
    fig.add_trace(go.Bar(
        x=subset['Transmission'],
        y=subset['Price'],
        name=company,
        orientation='v',
        marker_color=bar_colors[i % len(bar_colors)]  # Cycle through the list of colors
    ))
```

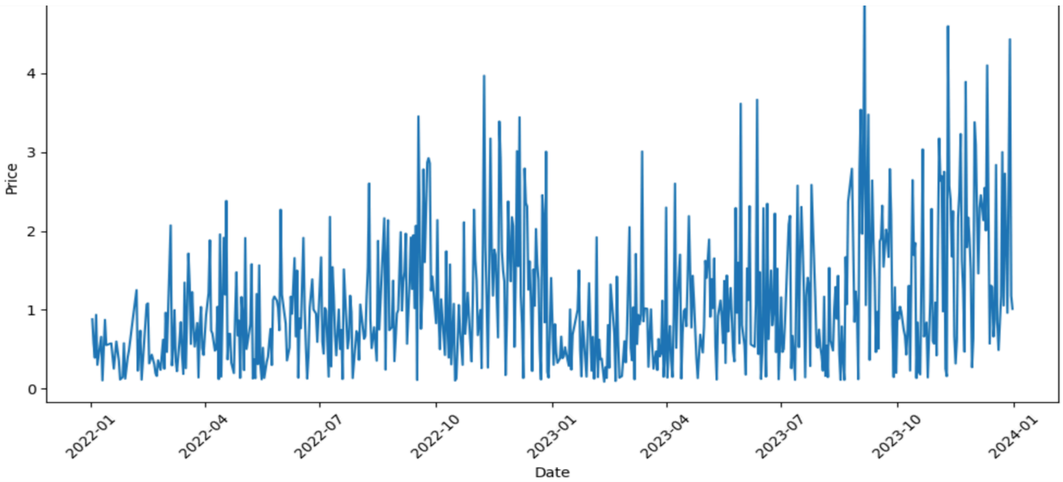


# price trend over time

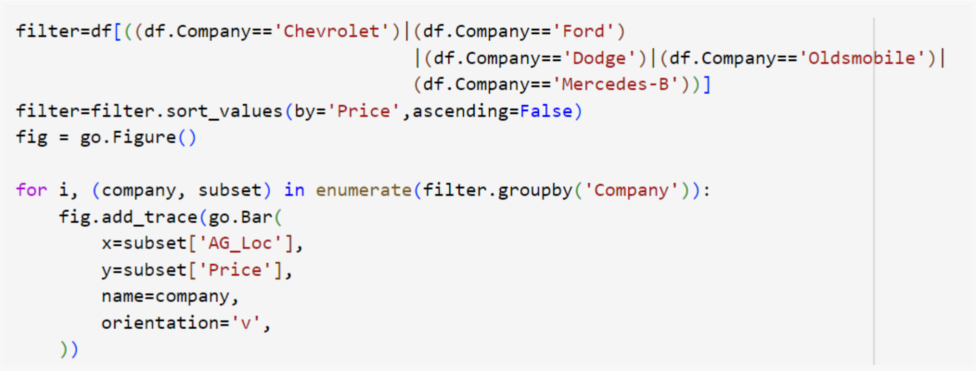


```
time_series_data = df.groupby('Date').agg({'Price': 'sum'}).reset_index()
plt.figure(figsize=(12, 6))
sns.lineplot(data=time_series_data, x='Date', y='Price')
plt.title('Price Trend Over Time')
plt.xlabel('Date')
plt.ylabel('Price')
plt.xticks(rotation=45)
```

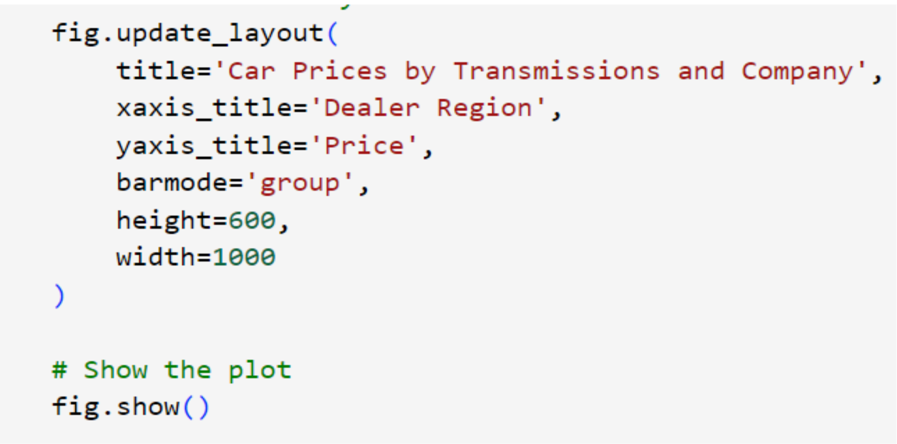
```
(array([18993., 19083., 19174., 19266., 19358., 19448., 19539., 19631.,
        19723.]),
 [Text(18993.0, 0, '2022-01'),
  Text(19083.0, 0, '2022-04'),
  Text(19174.0, 0, '2022-07'),
  Text(19266.0, 0, '2022-10'),
  Text(19358.0, 0, '2023-01'),
  Text(19448.0, 0, '2023-04'),
  Text(19539.0, 0, '2023-07'),
  Text(19631.0, 0, '2023-10'),
  Text(19723.0, 0, '2024-01')])
```



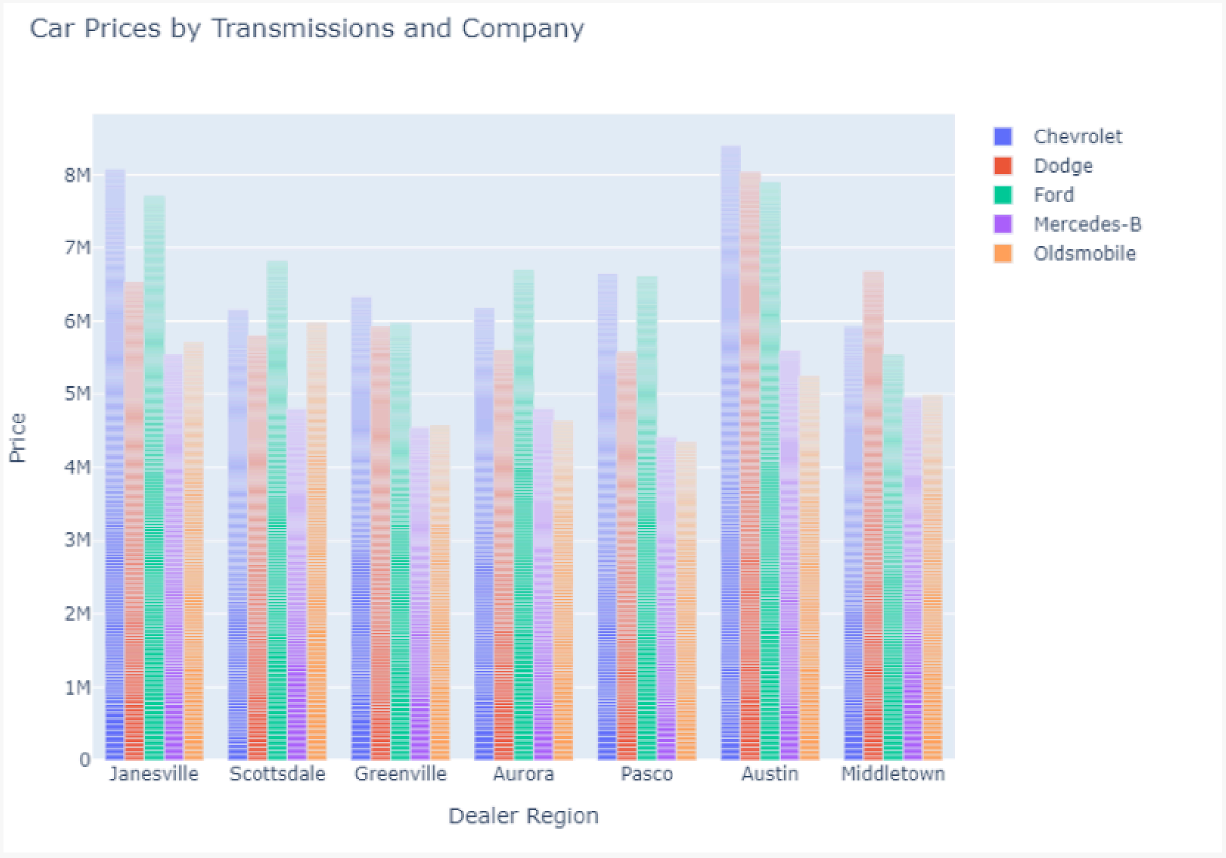
•DEALER REGION :



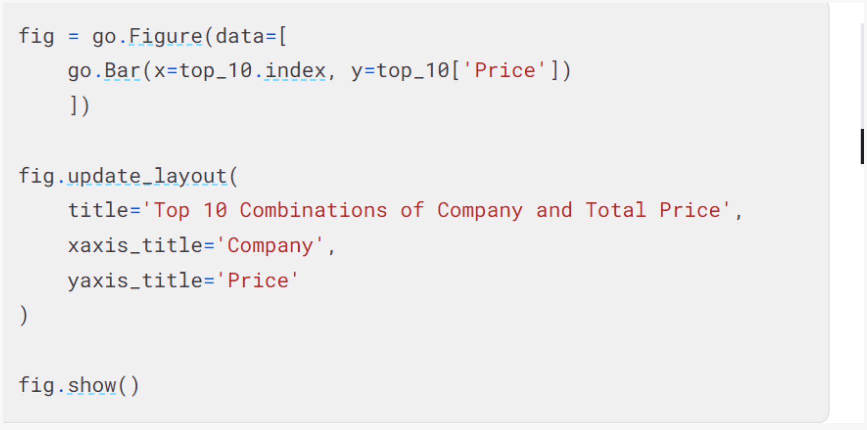
```
filter=df[((df.Company=='Chevrolet')|(df.Company=='Ford')
                                      |(df.Company=='Dodge')|(df.Company=='Oldsmobile')|
                                      (df.Company=='Mercedes-B'))]
filter=filter.sort_values(by='Price',ascending=False)
fig = go.Figure()

for i, (company, subset) in enumerate(filter.groupby('Company')):
    fig.add_trace(go.Bar(
        x=subset['AG_Loc'],
        y=subset['Price'],
        name=company,
        orientation='v',
    ))
```



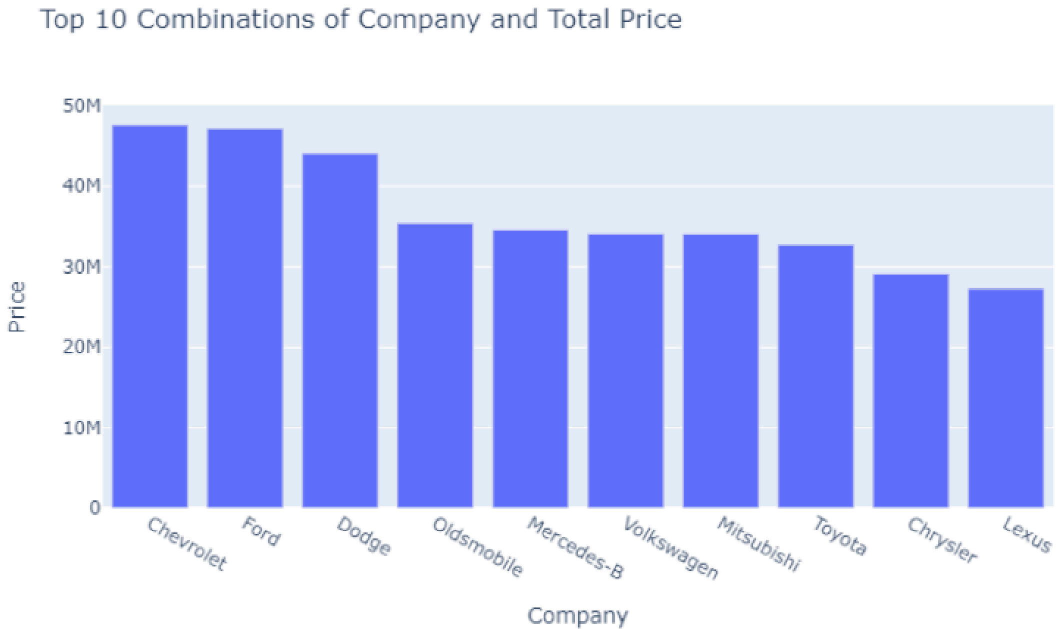
```
fig.update_layout(
    title='Car Prices by Transmissions and Company',
    xaxis_title='Dealer Region',
    yaxis_title='Price',
    barmode='group',
    height=600,
    width=1000
)

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



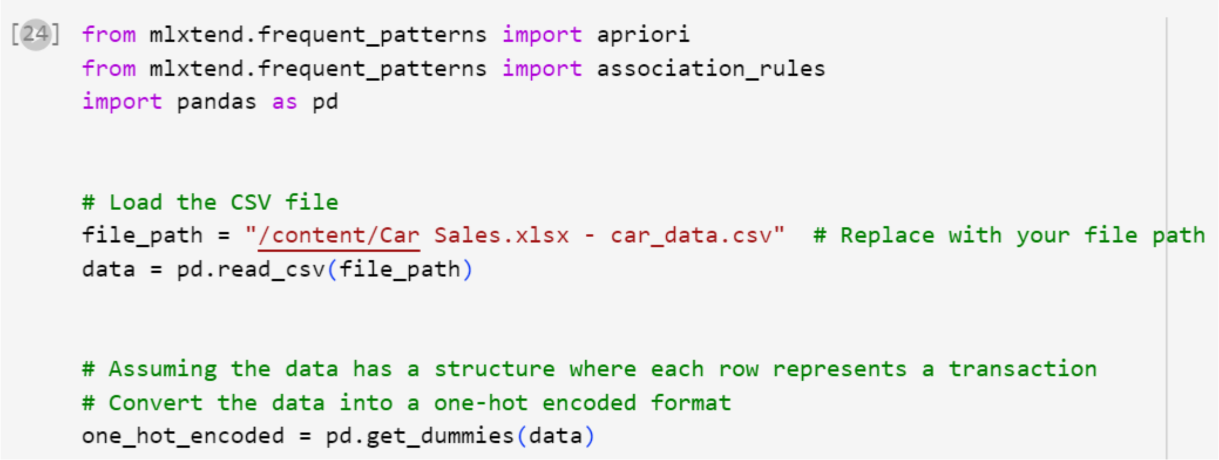
# combination of company and total price



```
fig = go.Figure(data=[
    go.Bar(x=top_10.index, y=top_10['Price'])
    ])

fig.update_layout(
    title='Top 10 Combinations of Company and Total Price',
    xaxis_title='Company',
    yaxis_title='Price'
)

fig.show()
```

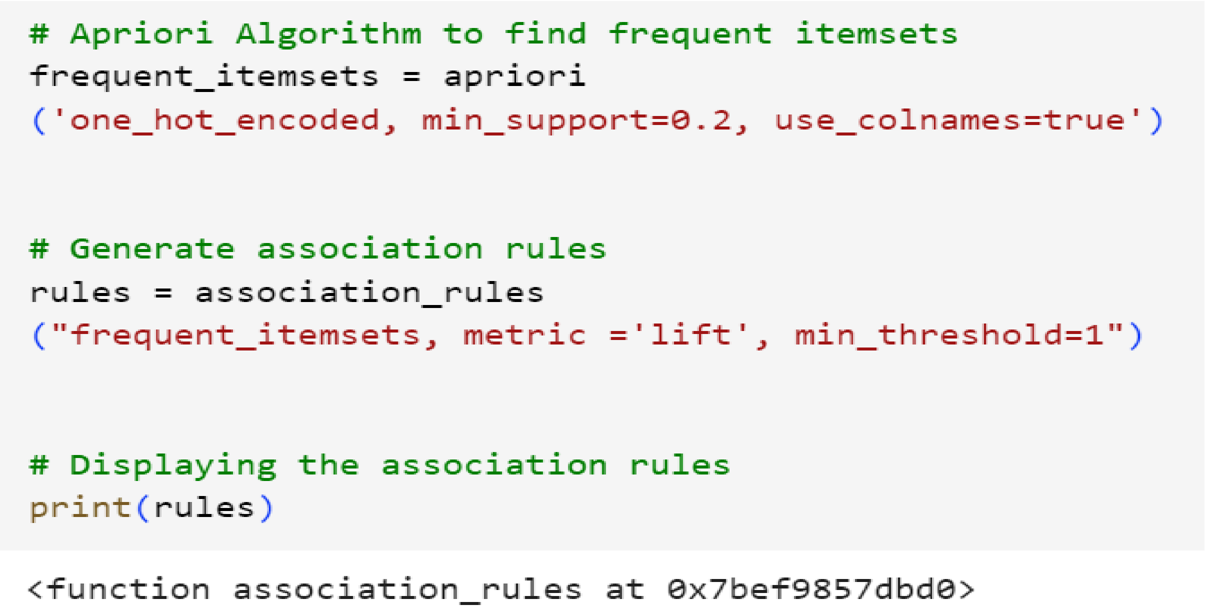


- **APRIORI ALGORITHM**



[24]

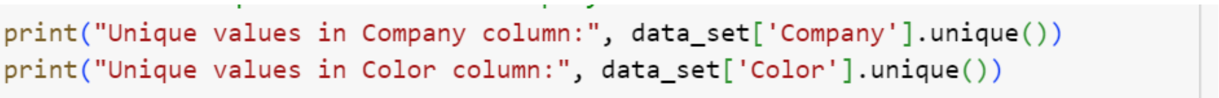
```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import pandas as pd


# Load the CSV file
file_path = "/content/Car Sales.xlsx - car_data.csv"  # Replace with your file path
data = pd.read_csv(file_path)


# Assuming the data has a structure where each row represents a transaction
# Convert the data into a one-hot encoded format
one_hot_encoded = pd.get_dummies(data)
```



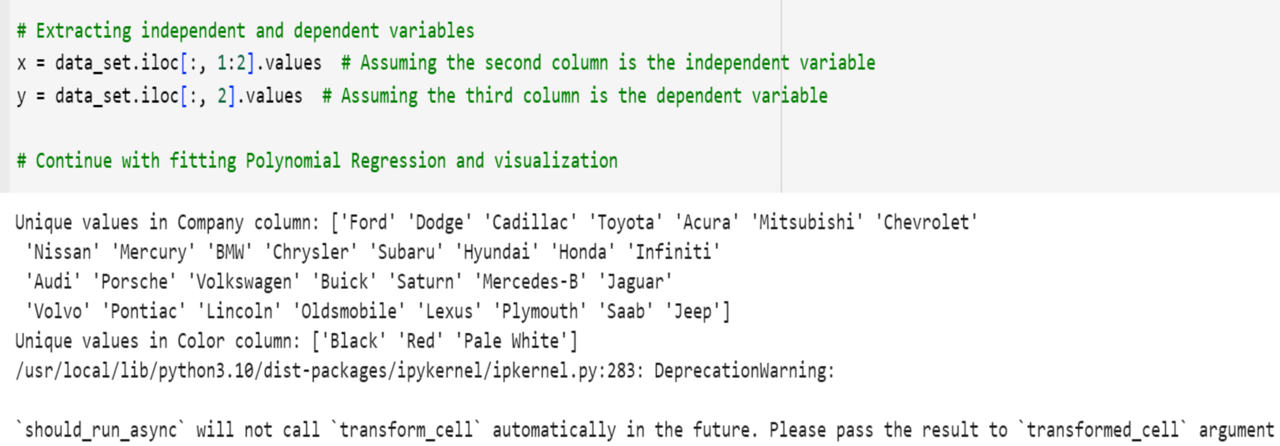
```
# Apriori Algorithm to find frequent itemsets
frequent_itemsets = apriori
('one_hot_encoded, min_support=0.2, use_colnames=true')


# Generate association rules
rules = association_rules
("frequent_itemsets, metric ='lift', min_threshold=1")


# Displaying the association rules
print(rules)
```

```
<function association_rules at 0x7bef9857dbd0>
```

- **POLYNOMIAL ALGORITHM**



```
print("Unique values in Company column:", data_set['Company'].unique())
print("Unique values in Color column:", data_set['Color'].unique())
```



```
# Extracting independent and dependent variables
x = data_set.iloc[:, 1:2].values  # Assuming the second column is the independent variable
y = data_set.iloc[:, 2].values  # Assuming the third column is the dependent variable

# Continue with fitting Polynomial Regression and visualization
```

```
Unique values in Company column: ['Ford' 'Dodge' 'Cadillac' 'Toyota' 'Acura' 'Mitsubishi' 'Chevrolet'
 'Nissan' 'Mercury' 'BMW' 'Chrysler' 'Subaru' 'Hyundai' 'Honda' 'Infiniti'
 'Audi' 'Porsche' 'Volkswagen' 'Buick' 'Saturn' 'Mercedes-B' 'Jaguar'
 'Volvo' 'Pontiac' 'Lincoln' 'Oldsmobile' 'Lexus' 'Plymouth' 'Saab' 'Jeep']
Unique values in Color column: ['Black' 'Red' 'Pale White']
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

`should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argument
```

CONCLUSION

car sales data analysis is indispensable for understanding and navigating the complexities of the automotive industry. By examining metrics such as sales volumes, market share, consumer preferences, and regional variations, stakeholders gain valuable insights that inform critical decision-making processes.

_THANK YOU!!!