Required Library

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Load Data

In [ ]:
```
Car = pd.read_csv("Car_price.csv")
```

In [ ]:
```
#first 5 rows
Car.head()
```

Out[ ]:

| | Brand | Model Name | Model Variant | Car Type | Transmission | Fuel Type | Year | Kilometers |
|---|---|---|---|---|---|---|---|---|
| **0** | Mahindra | TUV300 | AX5 | SUV | Manual | CNG | 2017 | 164654 |
| **1** | Skoda | Rapid | Style | Sedan | Manual | Petrol | 2018 | 41351 |
| **2** | Maruti Suzuki | Alto | Z | Hatchback | Manual | Diesel | 2002 | 119090 |
| **3** | Hyundai | Grand i10 | Magna | Hatchback | Manual | Diesel | 2013 | 19979 |
| **4** | Mahindra | XUV500 | W8 | SUV | Manual | Petrol | 2011 | 130591 |

In [ ]:
```
#last 5 rows
Car.tail()
```

Out[ ]:

| | Brand | Model Name | Model Variant | Car Type | Transmission | Fuel Type | Year | Kilome |
|---|---|---|---|---|---|---|---|---|
| **140899** | Mahindra | Scorpio | W11 | SUV | Manual | Petrol | 2002 | 6 |
| **140900** | Hyundai | i10 | Era | Hatchback | Manual | Petrol | 2013 | 4 |
| **140901** | Honda | Jazz | V | Sedan | Automatic | Diesel | 2009 | 13 |
| **140902** | Honda | WR-V | E | SUV | Manual | Diesel | 2022 | 8 |
| **140903** | Chevrolet | Tavera | Base | MPV | Manual | Diesel | 2009 | 4 |

Required COLUMNS

In [ ]:
```
print("__List of columns__")
print()
Car.columns.tolist()
```

```
__List of columns__
```

Out[ ]:
```
['Brand',
 'Model Name',
 'Model Variant',
 'Car Type',
 'Transmission',
 'Fuel Type',
 'Year',
 'Kilometers',
 'Owner',
 'State',
 'Accidental',
 'Price']
```

In [ ]:
```
New_Car= Car[["Brand","Car Type","Transmission","Fuel Type","Year","Kilometers
"Owner","State","Accidental","Price"]]
```

In [ ]:
```
#shape
print("Shape of Car Data")
New_Car.shape
```

```
Shape of Car Data
```

Out[ ]:
```
(140904, 10)
```

In [ ]:
```
#Column and rows
col = New_Car.shape[1]
row = New_Car.shape[0]
print("Total Numbers of Columns: ", col)
print("Total Numbers of Rows: ", row)
```

```
Total Numbers of Columns:  10
Total Numbers of Rows:  140904
```

In [ ]:
```
#Information of Data
print("___Information of Data___")
print()
New_Car.info()
```

```
___Information of Data___

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 140904 entries, 0 to 140903
Data columns (total 10 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   Brand         140904 non-null  object
 1   Car Type      140904 non-null  object
 2   Transmission  140904 non-null  object
 3   Fuel Type     140904 non-null  object
 4   Year          140904 non-null  int64
 5   Kilometers    140904 non-null  int64
 6   Owner         140904 non-null  object
 7   State         140904 non-null  object
 8   Accidental    140904 non-null  object
 9   Price         140904 non-null  int64
dtypes: int64(3), object(7)
memory usage: 10.8+ MB
```

In [ ]:
```
#check Data type
New_Car.dtypes
```

Out[ ]:

| | 0 |
|---|---|
| **Brand** | object |
| **Car Type** | object |
| **Transmission** | object |
| **Fuel Type** | object |
| **Year** | int64 |
| **Kilometers** | int64 |
| **Owner** | object |
| **State** | object |
| **Accidental** | object |
| **Price** | int64 |

**dtype:** object

In [ ]:
```
#change Data type for year colum
New_Car.loc[:,"Year"] = New_Car["Year"].astype(str)
print(New_Car["Year"].dtypes)
```

```
object
```

Check Duplicates

In [ ]:
```
duplicate = New_Car.duplicated().sum()
print("Total Numbers of Duplicate: ", duplicate)
```

```
Total Numbers of Duplicate:  0
```

In [ ]:
```
#Check Null
print("__Sum of Null__")
print()
New_Car.isnull().sum()
```

```
__Sum of Null__
```

Out[ ]:

| | 0 |
|---|---|
| **Brand** | 0 |
| **Car Type** | 0 |
| **Transmission** | 0 |
| **Fuel Type** | 0 |
| **Year** | 0 |
| **Kilometers** | 0 |
| **Owner** | 0 |
| **State** | 0 |
| **Accidental** | 0 |
| **Price** | 0 |

**dtype:** int64

In [ ]:
```
#Describe of Columns
print("___Description of Data___")
print()
New_Car.describe()
```

```
___Description of Data___
```

Out[ ]:

| | Kilometers | Price |
|---|---|---|
| **count** | 140904.000000 | 1.409040e+05 |
| **mean** | 95024.595987 | 7.617872e+05 |
| **std** | 49133.157878 | 4.438578e+05 |
| **min** | 10000.000000 | 5.005500e+04 |
| **25%** | 52421.000000 | 4.116420e+05 |
| **50%** | 94973.500000 | 6.828030e+05 |
| **75%** | 137618.000000 | 1.034178e+06 |
| **max** | 179998.000000 | 2.744280e+06 |

In [ ]:
```
New_Car.describe(include= "object")
```

Out[ ]:

| | Brand | Car Type | Transmission | Fuel Type | Year | Owner | State | A |
|---|---|---|---|---|---|---|---|---|
| **count** | 140904 | 140904 | 140904 | 140904 | 140904 | 140904 | 140904 | |
| **unique** | 18 | 5 | 2 | 5 | 24 | 3 | 27 | |
| **top** | Maruti Suzuki | Hatchback | Manual | Petrol | 2020 | 1st | Maharashtra | |
| **freq** | 54030 | 55890 | 119793 | 79187 | 14608 | 75429 | 24613 | |

Exploraty Data Analysis and Visualization

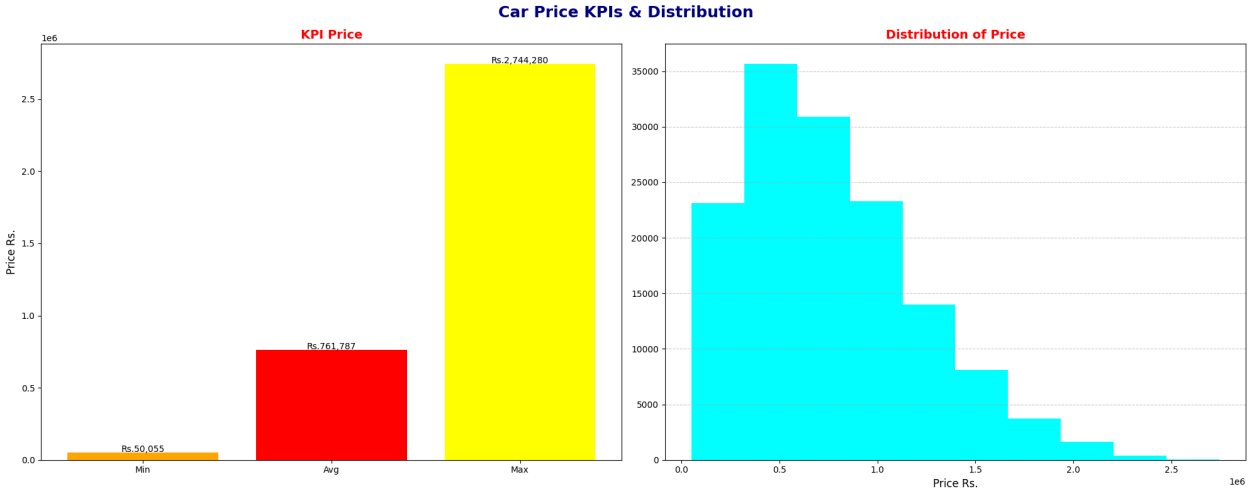
KPI required

Price

In [ ]:
```
print("____KPI for Price____\n")

Total_Profit = np.sum(New_Car["Price"])
Avg_Price= np.mean(New_Car["Price"])
Max_Price = np.max(New_Car["Price"])
Min_Price= np.min(New_Car["Price"])
Percentiles = New_Car["Price"].quantile([0.25,0.50,0.75])
print(f"Total Profit = Rs. {Total_Profit}")
print(f"Average Price = Rs. {Avg_Price:,.3f}")
print(f"Maximum Price = Rs. {Max_Price}")
print(f"Minimum Price = Rs. {Min_Price}")
print("Percentiles:")
print(f"0.25 Rs. {Percentiles[0.25]:,.0f}")
print(f"0.50 Rs. {Percentiles[0.50]:,.0f}")
print(f"0.75 Rs. {Percentiles[0.75]:,.0f}")
```

```
____KPI for Price____

Total Profit = Rs. 107338859947
Average Price = Rs. 761,787.174
Maximum Price = Rs. 2744280
Minimum Price = Rs. 50055
Percentiles:
0.25 Rs. 411,642
0.50 Rs. 682,803
0.75 Rs. 1,034,178
```

In [ ]:
```
fig ,axs= plt.subplots(1,2, figsize= (20,8))

#KPIs for Price
#____Bar chart for Price____
x = ["Min", "Avg", "Max"]
y= [Min_Price,Avg_Price,Max_Price]
c = ["Orange", "Red", "Yellow"]
axs[0].bar(x,y, color=c)
axs[0].set_title("KPI Price", fontsize=14, fontweight="bold",color= "Red")
axs[0].set_ylabel("Price Rs.",fontsize=12)

#show values for bar chart
for i,v in enumerate(y):
    axs[0].text(i, v+5000, f"Rs.{v:,.0f}",ha = "center", fontsize= 10, color=


#___Histogram for price distribution____
axs[1].hist(New_Car["Price"],bins=10, color= "Aqua")
axs[1].set_title("Distribution of Price",fontsize=14, fontweight="bold",color
axs [1]. set_xlabel("Price Rs.", fontsize= 12)
axs[1]. grid(axis ="y", linestyle="--",alpha=0.6)
fig.suptitle("Car Price KPIs & Distribution", fontsize=18, fontweight="bold",
plt.tight_layout()
plt.show()
```

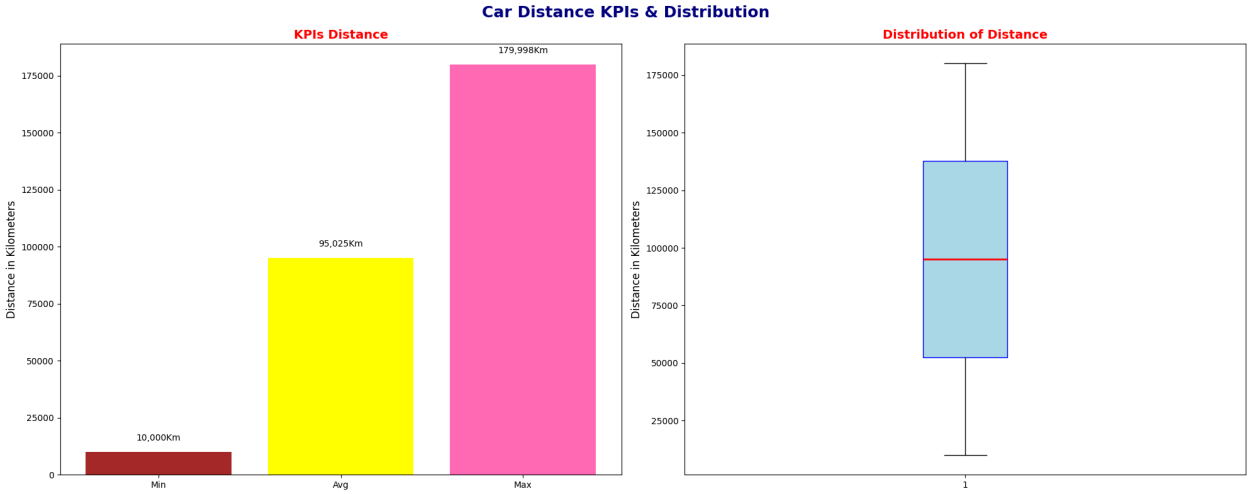



Kilometers

In [ ]:
```
print("____KPI for Kilometres____\n")

Total_Distance = np.sum(New_Car["Kilometers"])
Avg= New_Car["Kilometers"].mean()
Max_Distance = np.max(New_Car["Kilometers"])
Min_Distance = np.min(New_Car["Kilometers"])
Percentiles = New_Car["Kilometers"].quantile([0.25,0.50,0.75])
print(f"Total Distance = {Total_Distance} Km")
print(f"Average Distance  = {Avg:.3f} Km")
print(f"Maximum Distance = {Max_Distance} Km")
print(f"Minimum Distance =  {Min_Distance} Km")
print("Percentiles:")
print(f"25%  {Percentiles[0.25]:,.0f} Km")
print(f"50%  {Percentiles[0.50]:,.0f} Km")
print(f"75% {Percentiles[0.75]:,.0f} Km")
```

```
____KPI for Kilometres____

Total Distance = 13389345673 Km
Average Distance  = 95024.596 Km
Maximum Distance = 179998 Km
Minimum Distance =  10000 Km
Percentiles:
25%  52,421 Km
50%  94,974 Km
75% 137,618 Km
```

In [ ]:
```
fig ,axs= plt.subplots(1,2, figsize= (20,8))

#KPIs for Price
#____Bar chart for Price____
x = ["Min", "Avg", "Max"]
y = [Min_Distance,Avg,Max_Distance]
c = ["Brown", "yellow", "Hotpink"]
axs[0].bar(x,y, color = c)
axs[0].set_title("KPIs Distance", fontsize=14, fontweight="bold",color= "Red")
axs[0].set_ylabel("Distance in Kilometers",fontsize=12)

#show values for bar chart
for i,v in enumerate(y):
    axs[0].text(i, v+5000, f"{v:,.0f}Km",ha = "center", fontsize= 10, color= "

#___Histogram for price distribution____
axs[1].boxplot(New_Car["Kilometers"],patch_artist=True, boxprops= dict(facecol
axs[1].set_title("Distribution of Distance",fontsize=14, fontweight="bold",col
axs[1].set_ylabel("Distance in Kilometers", fontsize=12)
fig.suptitle("Car Distance KPIs & Distribution", fontsize=18, fontweight="bold
plt.tight_layout()
plt.show()
```




1. Which state contributes the most and less to total sales?

In [ ]:
```
State= New_Car.groupby("State")["Price"]. sum().reset_index()
Highest_State = State.sort_values(by= "Price", ascending=False)[0:5]
Highest_State
```

Out[ ]:

| | State | Price |
|---|---|---|
| **15** | Maharashtra | 19469266696 |
| **5** | Delhi | 16858388878 |
| **12** | Karnataka | 14943331497 |
| **22** | Tamil Nadu | 10758335410 |
| **7** | Gujarat | 9755059732 |

In [ ]:
```
Lowest_State= State.sort_values(by= "Price", ascending=True)[0:5]
Lowest_State
```

Out[ ]:

| | State | Price |
|---|---|---|
| **19** | Puducherry | 190810835 |
| **18** | Other UTs | 201642499 |
| **25** | Uttarakhand | 242054908 |
| **10** | Jammu & Kashmir | 268168937 |
| **9** | Himachal Pradesh | 282015173 |

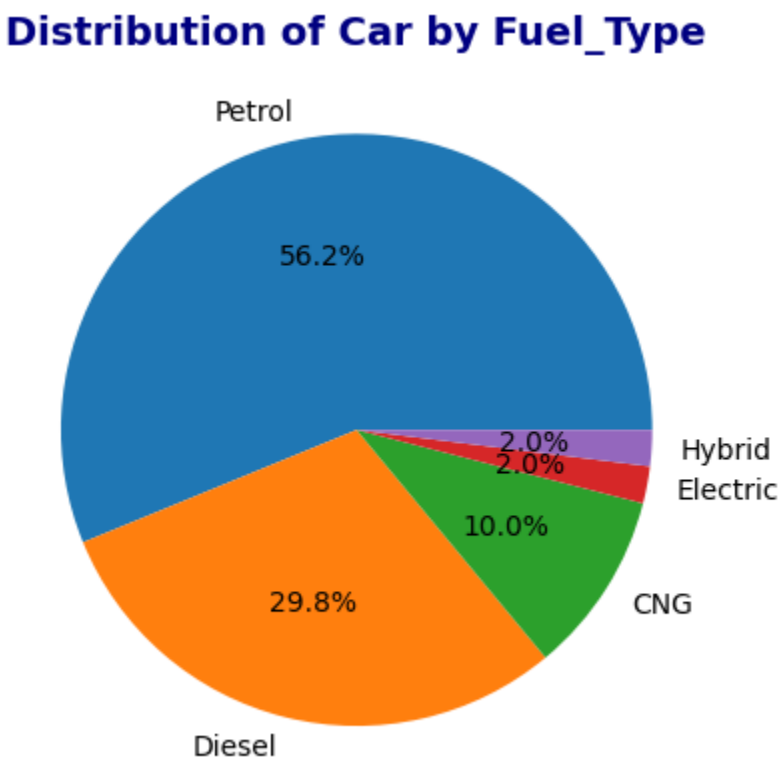
Result/outcome

The total car sales across all states amount to Rs. 107,338,859,947. Maharashtra

contributed the highest, accounting for 18.14% of total sales, while Puducherry contributed the lowest, with only 0.18% of total sales. This indicates that Maharashtra is the leading market for car sales, whereas Puducherry represents a very small portion of the overall market.

2. Which fuel type is most common among the cars?"

In [ ]:
```
x=New_Car["Fuel Type"].value_counts()
y= New_Car["Fuel Type"]. value_counts().keys()
plt.pie(x, labels=y,autopct="%0.1f%%")
plt.title("Distribution of Car by Fuel_Type", fontsize=14, fontweight='bold',c
plt.show()
```

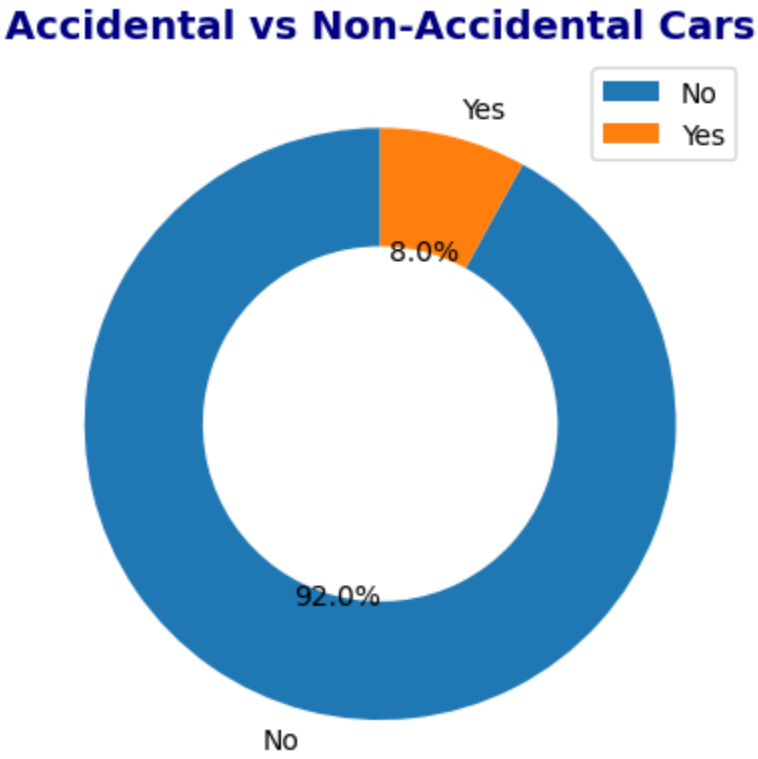



Result/Outcome

The dataset shows the distribution of cars by fuel type. Petrol is the most common fuel type, making up 56.2% of all cars, followed by Diesel at 29.8%, CNG at 10%, and Hybrid & Electric cars at only 2.0%. This indicates that Petrol cars dominate the market, while Hybrid and Electric cars are still a very small fraction of total sales.

3. How many cars have been involved in accidents compared to non-accidental cars?

In [ ]:
```
x = New_Car["Accidental"].value_counts()
y = New_Car["Accidental"]. value_counts().keys()
plt.pie(x, labels = y, autopct="%0.1f%%",startangle= 90,wedgeprops= {"width":0
plt.legend()
plt.title("Accidental vs Non-Accidental Cars", fontsize=14, fontweight='bold',
plt.show()
```




Outcome/Result

The analysis shows that 8% of cars are accidental while 92% are non-accidental, indicating that most vehicles sold are in good condition and buyers prefer non-accidental cars.

4. Find the top 5 Revenue Cars

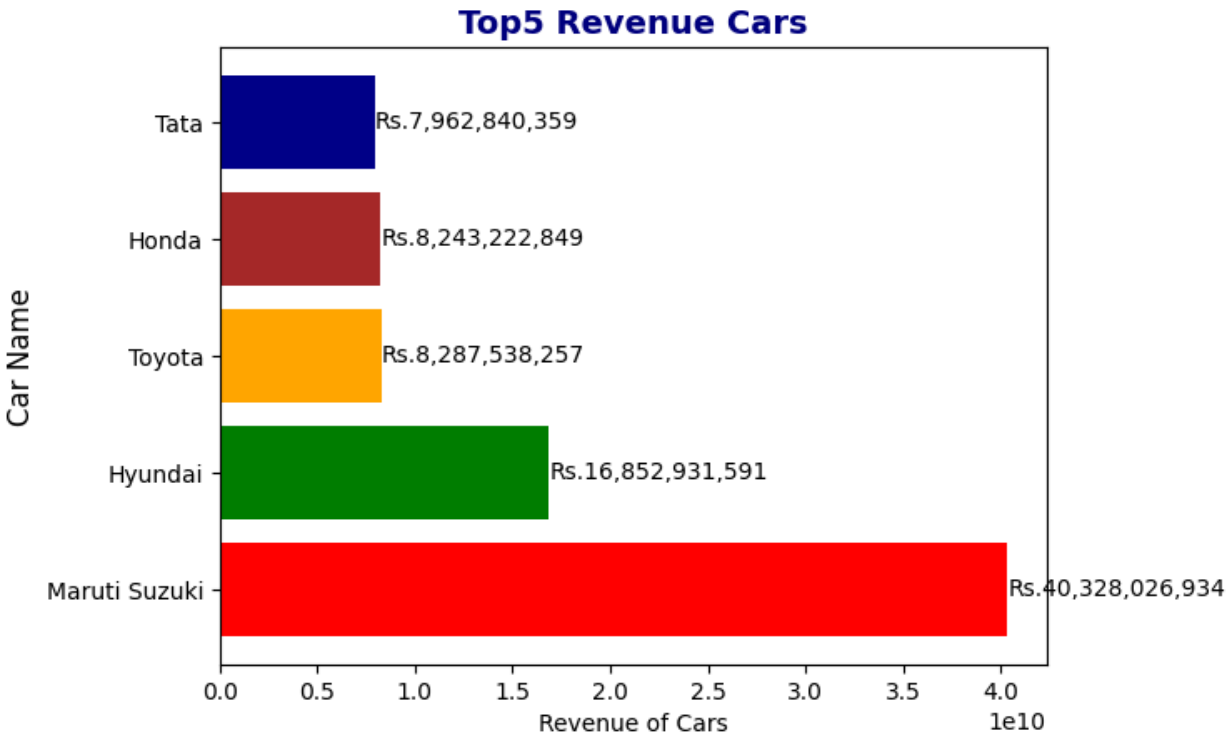
In [ ]:
```
Brand= New_Car.groupby("Brand")["Price"].sum().reset_index()
Top5_Revenue_Car= Brand.sort_values(by= "Price", ascending=False)[0:5]
```

In [ ]:
```
Top5_Revenue_Car
```

Out[ ]:

| | Brand | Price |
|---|---|---|
| **10** | Maruti Suzuki | 40328026934 |
| **5** | Hyundai | 16852931591 |
| **16** | Toyota | 8287538257 |
| **4** | Honda | 8243222849 |
| **15** | Tata | 7962840359 |

In [ ]:
```
plt.barh(Top5_Revenue_Car["Brand"], Top5_Revenue_Car["Price"],color = ["Red","
plt.title("Top5 Revenue Cars", fontsize= 14, fontweight= "bold", color= "navy"
plt.ylabel("Car Name",fontsize=12)
plt.xlabel("Revenue of Cars")
for i, v in enumerate(Top5_Revenue_Car["Price"]):
    plt.text(v + 5000, i, f"Rs.{v:,.0f}", va='center', fontsize=10, color='bla
plt.show()
```




Result/outcome

Maruti generated the highest revenue, contributing 37.58% of total sales, indicating strong market dominance.

5. Find the top 5 sold Car

In [ ]:
```
Top5_sold_Car = New_Car["Brand"].value_counts().reset_index()[0:5]
```

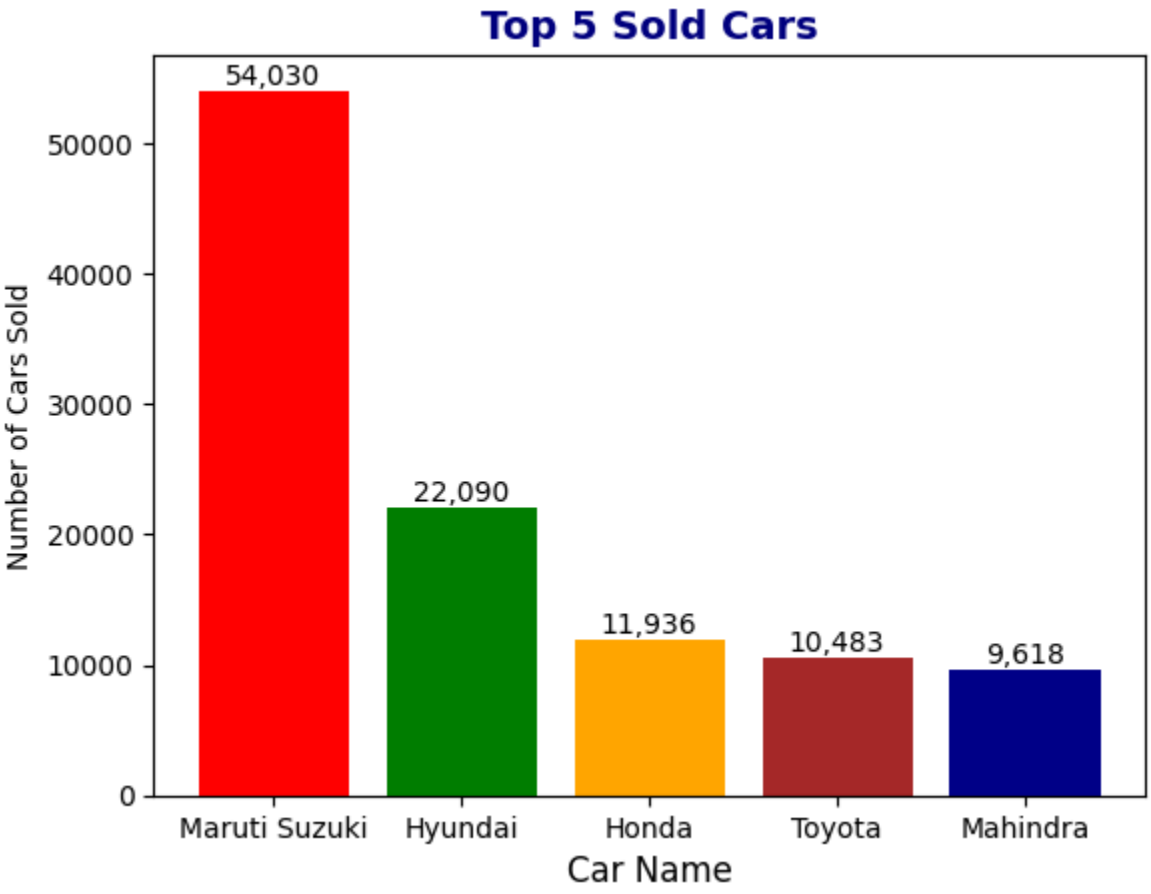
```
Top5_sold_Car
```

Out[ ]:

| | Brand | count |
|---|---|---|
| **0** | Maruti Suzuki | 54030 |
| **1** | Hyundai | 22090 |
| **2** | Honda | 11936 |
| **3** | Toyota | 10483 |
| **4** | Mahindra | 9618 |

In [ ]:

```
plt.bar(Top5_sold_Car["Brand"], Top5_sold_Car["count"], color = ["Red","Green"
plt.title("Top 5 Sold Cars", fontsize= 14, fontweight= "bold", color= "navy")
plt.xlabel("Car Name",fontsize=12)
plt.ylabel("Number of Cars Sold")
for i, v in enumerate(Top5_sold_Car["count"]):
    plt.text(i, v + 500, f"{v:,}", ha='center', fontsize=10, color='black')
plt.show()
```




Result/outcome

Maruti Suzuki is the most sold car brand, accounting for 49.96% of the top 5 sold

cars.

6.Top and Bottom 10 Years Comparison for Revenue and Units Sold.

a. Which years contributed the highest and lowest total revenue from car sales?

b. Which years recorded the highest and lowest number of cars sold?

In [ ]:
```
#Total sold by year
Car_Sold_Year = New_Car["Year"].value_counts().reset_index()
Top10_Sold_year = Car_Sold_Year.head(10)
Buttom10_Sold_year = Car_Sold_Year.tail(10)
```

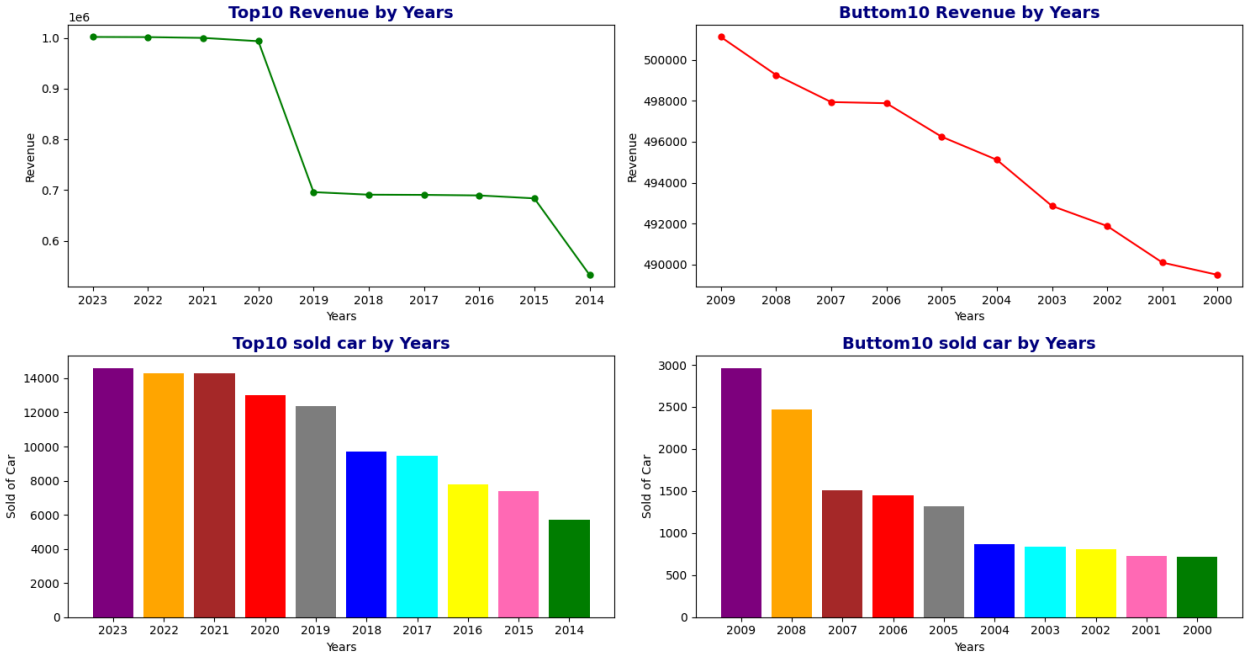
In [ ]:
```
#Sorting year by sold car
sort_sold_year =Car_Sold_Year.sort_values("Year", ascending=False)
Top10_sold_year= sort_sold_year.head(10)
Buttom10_sold_year= sort_sold_year.tail(10)
```

In [ ]:
```
#Total Revenue by Year
Revenue_year = New_Car.groupby("Year")["Price"].mean(). reset_index()
Revenue_year = Revenue_year.sort_values(by= "Price", ascending= False)
Top10_Revenue_year = Revenue_year.head(10)
Buttom10_Revenue_year = Revenue_year.tail(10)
```

In [ ]:
```
#sorting_year_revenue
sort_year = Revenue_year.sort_values("Year", ascending=False)
Top10_sort_year = sort_year.head(10)
Buttom10_sort_year= sort_year.tail(10)
```

In [ ]:
```
plt.figure(figsize=(15,8))

#Top10 revenue Car by year
plt.subplot(2,2,1)
a= Top10_sort_year["Year"]
b = Top10_Revenue_year["Price"]
plt.plot(a,b,marker= "o", markersize=5,color = "green")
plt.xlabel("Years")
plt.ylabel("Revenue")
plt.title("Top10 Revenue by Years", fontsize=14, fontweight="bold", color= "na

#Buttom10 Revenue Car by year
plt.subplot(2,2,2)
c= Buttom10_sort_year["Year"]
d = Buttom10_Revenue_year["Price"]
plt.plot(c,d,marker= "o",markersize=5,color= "red")
plt.xlabel("Years")
plt.ylabel("Revenue")
plt.title("Buttom10 Revenue by Years", fontweight= "bold", fontsize=14, color=

#Top10 sold Car in years
plt.subplot(2,2,3)
```

```
e = Top10_sold_year["Year"]
f = Top10_Sold_year["count"]
plt.bar(e,f, color=["Purple","Orange","Brown","Red","Gray","Blue","Aqua","yell
plt.xlabel("Years")
plt.ylabel("Sold of Car")
plt.title("Top10 sold car by Years", fontsize=14, fontweight="bold", color="na

#Buttom5 sold Car in year
plt.subplot(2,2,4)
g = Buttom10_sold_year["Year"]
h = Buttom10_Sold_year["count"]
plt.bar(g,h, color=["Purple","Orange","Brown","Red","Gray","Blue","Aqua","yell
plt.xlabel("Years")
plt.ylabel("Sold of Car")
plt.title("Buttom10 sold car by Years", fontsize=14, fontweight="bold", color=
plt.tight_layout()
plt.show()
```




Outcome/Result

Top 10 years show the highest revenue and car sales, indicating strong market demand. Bottom 10 years reflect low performance due to reduced sales or lower car prices. Some years show high revenue despite fewer sales — meaning premium cars boosted earnings.