for data analysis

In [48]:
```
import pandas as pd
import numpy as np
```

for data visualization

In [51]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [52]:
```
df= pd.read_csv("Users/Hp/AppData/Roaming/Microsoft/Windows/Start Menu/Programmes/Anaconda3 (64-bit)/Data_Visualization_with_Python_12-main/vw.csv")
display(df.shape)
print('number of rows',df.shape[0])
print('number of columns',df.shape[1])
display(df)
print(df)
```

```
(15157, 9)
number of rows 15157
number of columns 9
```

	model	year	price	transmission	mileage	fuelType	mpg	engineSize
0	T-Roc	2019	25000	Automatic	13904	Diesel	49.6	2.0
1	T-Roc	2019	26963	Automatic	4562	Diesel	49.6	2.0
2	T-Roc	2019	25000	Manual	7414	Diesel	50.4	2.0
3	T-Roc	2019	26462	Automatic	4825	Petrol	32.5	2.0
4	T-Roc	2019	22995	Semi-Auto	6500	Petrol	39.8	1.5
...	...	...	...	...	...	...	...	...
15152	Eos	2012	5990	Manual	74000	Diesel	58.9	2.0
15153	Fox	2008	1799	Manual	88102	Petrol	46.3	1.2
15154	Fox	2009	1590	Manual	70000	Petrol	41.0	1.4
15155	Fox	2006	1250	Manual	82764	Petrol	46.3	1.2
15156	Fox	2007	2295	Manual	74000	Petrol	46.3	1.2

15157 rows × 9 columns

```
      model  year  price transmission  mileage fuelType  mpg  engineSize
0     T-Roc  2019  25000    Automatic    13904   Diesel 49.6         2.0
1     T-Roc  2019  26963    Automatic     4562   Diesel 49.6         2.0
2     T-Roc  2019  25000       Manual     7414   Diesel 50.4         2.0
3     T-Roc  2019  26462    Automatic     4825   Petrol 32.5         2.0
4     T-Roc  2019  22995    Semi-Auto     6500   Petrol 39.8         1.5
...     ...   ...    ...          ...      ...      ...  ...         ...
15152   Eos  2012   5990       Manual    74000   Diesel 58.9         2.0
15153   Fox  2008   1799       Manual    88102   Petrol 46.3         1.2
15154   Fox  2009   1590       Manual    70000   Petrol 41.0         1.4
15155   Fox  2006   1250       Manual    82764   Petrol 46.3         1.2
15156   Fox  2007   2295       Manual    74000   Petrol 46.3         1.2

[15157 rows x 9 columns]
```

In [53]:
```
display(df.head())
```

	model	year	price	transmission	mileage	fuelType	mpg	engineSize
0	T-Roc	2019	25000	Automatic	13904	Diesel	49.6	2.0
1	T-Roc	2019	26963	Automatic	4562	Diesel	49.6	2.0
2	T-Roc	2019	25000	Manual	7414	Diesel	50.4	2.0
3	T-Roc	2019	26462	Automatic	4825	Petrol	32.5	2.0
4	T-Roc	2019	22995	Semi-Auto	6500	Petrol	39.8	1.5

value counts()

In [54]:
```
display(df["transmission"].value_counts())
```

```
Manual       9417
Semi-Auto    3780
Automatic    1960
Name: transmission, dtype: int64
```

Pie-Chart

In [55]:
```
import matplotlib.pyplot as plt
df= pd.read_csv("vw.csv")
df["transmission"].value_counts().plot(kind="pie",autopct='%1.2f%%',startangle=90)
plt.title('Percentage of Transmission of Cars')
plt.ylabel('')
plt.show()
```

Creating DataFrame

In [56]:
```
df=pd.DataFrame(df["transmission"].value_counts())
display(df.head)
```

	transmission
Manual	9417
Semi-Auto	3780
Automatic	1960

In [57]:
```
df.reset_index(inplace=True)
df.rename(columns={"index":"transmission type",
                   "transmission":"number of cars"})
display(df.head)
```

	transmission type	number of cars
0	Manual	9417
1	Semi-Auto	3780
2	Automatic	1960

In [58]:
```
df["percentage of cars"]=(df["number of cars"]/df.shape[0])*100
display(df.head,round(2))
```

	transmission type	number of cars	percentage of cars
0	Manual	9417	62.13
1	Semi-Auto	3780	24.94
2	Automatic	1960	12.93

Bar-plot

In [59]:
```
sns.barplot(x="transmission type",y="percentage of cars",data=df,ax=None,hue=None,color="green")
plt.title("Percentage of Transmission of Cars")
plt.show()
```

Comparison

The pie-chart gives a clear indication that manual type holds greater percentage alone than the the other two types combined. If needed, percentages can also be shown as annotations in the bar-chart but the pie-chart shows this comparison even without annotations. On the other hand, x and y labels in the bar-graph presents a more detailed picture than the pie-chart all labels were not used in pie-charts.

So, for a more detailed picture, bars can be helpful while pie-charts are applicable to make visualizations in an easier.faster way

In [ ]:
```
```

In [60]:
```
def linear_eqn(x, m, c):
    y=np.arange(1.11,1)
    y=m*x+c
    y=m*x+c

    df=pd= pd.DataFrame()
    df=pd= pd.DataFrame()
    df=pd["x" ] = x
    df=pd["y" ] = m*x + c

    display (df=pd)
```

In [62]:
```
df= pd.read_csv("Users/Hp/AppData/Roaming/Microsoft/Windows/Start Menu/Programmes/Anaconda3 (64-bit)/Data_Visualization_with_Python_12-main/vw.csv")
display(df.head(10))
```

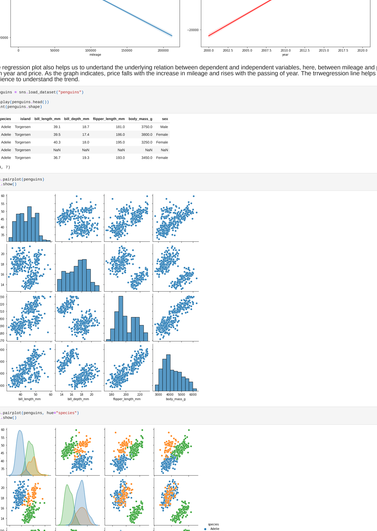
	model	year	price	transmission	mileage	fuelType	mpg	engineSize
0	T-Roc	2019	25000	Automatic	13904	Diesel	49.6	2.0
1	T-Roc	2019	26963	Automatic	4562	Diesel	49.6	2.0
2	T-Roc	2019	25000	Manual	7414	Diesel	50.4	2.0
3	T-Roc	2019	26462	Automatic	4825	Petrol	32.5	2.0
4	T-Roc	2019	22995	Semi-Auto	6500	Petrol	39.8	1.5
5	T-Roc	2020	31995	Manual	10	Petrol	42.2	1.5
6	T-Roc	2020	27995	Manual	10	Petrol	42.2	1.5
7	T-Roc	2020	28495	Semi-Auto	10	Petrol	32.5	2.0
8	T-Roc	2019	21995	Manual	10	Petrol	44.1	1.0
9	T-Roc	2019	23395	Manual	10	Petrol	42.2	1.5

In [63]:
```
plt.figure(figsize=(16,14))
```

In [64]:
```
plt.subplot(2, 1, 1)
sns.scatterplot(x="year", y="price", data=df)


plt.subplot(2, 2, 2)
sns.scatterplot(x="engineSize", y="price", data=df)

plt.tight_layout()
plt.show()
```



In the price Vs. year scatter plot, it is following a certain trend as the price rises wuth the passing of years. But the plot doesn't show any distinct relationship between engineSize and price. Price sometimes rises, sometimes stays constant, even sometimes falls as the increase in engine size.

In [66]:
```
plt.figure(figsize=(16,16))

plt.subplot(2,1, 1)
sns.regplot(x="mileage", y="price", data=df,scatter_kws={"color":"orange", "edgecolor":"white"})


plt.subplot(2, 2, 2)
sns.regplot(x="year",y="price", data=df, line_kws={"color":"red"},scatter_kws={"edgecolor":"white"})


plt.tight_layout()
plt.show()
```


The regression plot also helps us to understand the underlying relation between dependent and independent variables, here, between mileage and price and then year and price. As the graph indicates, price falls with the increase in mileage and rises with the passing of year. The trwegression line helps the audience to understand the trend.

In [67]:
```
penguins = sns.load_dataset("penguins")

display(penguins.head())
print(penguins.shape)
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female

```
(344, 7)
```

In [68]:
```
sns.pairplot(penguins)
plt.show()
```


In [69]:
```
sns.pairplot(penguins, hue="species")
plt.show()
```


Pair-plot helps to visualize multiple inputs in a single plot. This also helps to compare those inputs among themselves. For example, here, it is shown that in some cases, Chinstrap and Adélie show same characteristics as blue and orange dots are overlapping while the green dots are showing distant characters. In most cases, relation between x and y axis is positive. Adding a categorical column as hue was important as different color points indicate different inputs.

In [ ]:
```
```