```
import numpy as np
import pandas as pd

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
#import neccessary libraries
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df = pd.read_csv("/content/drive/MyDrive/Data_Analytics/BMW sales data (2010-2024) (1).csv")
df
```

	Model	Year	Region	Color	Fuel_Type	Transmission	Engine_Size_L	Mileage_KM	Price_USD	Sales_Volume	Sales_Classification
0	5 Series	2016	Asia	Red	Petrol	Manual	3.5	151748	98740	8300	High
1	i8	2013	North America	Red	Hybrid	Automatic	1.6	121671	79219	3428	Low
2	5 Series	2022	North America	Blue	Petrol	Automatic	4.5	10991	113265	6994	Low
3	X3	2024	Middle East	Blue	Petrol	Automatic	1.7	27255	60971	4047	Low
4	7 Series	2020	South America	Black	Diesel	Manual	2.1	122131	49898	3080	Low
...	...	...	...	...	...	...	...	...	...	...	...
49995	i3	2014	Asia	Red	Hybrid	Manual	4.6	151030	42932	8182	High
49996	i3	2023	Middle East	Silver	Electric	Manual	4.2	147396	48714	9816	High
49997	5 Series	2010	Middle East	Red	Petrol	Automatic	4.5	174939	46126	8280	High
49998	i3	2020	Asia	White	Electric	Automatic	3.8	3379	58566	9486	High
49999	X1	2020	North America	Blue	Diesel	Manual	3.3	171003	77492	1764	Low

50000 rows × 11 columns

```
df.nunique()
```

	0
Model	11
Year	15
Region	6
Color	6
Fuel_Type	4
Transmission	2
Engine_Size_L	36
Mileage_KM	44347
Price_USD	38246
Sales_Volume	9845
Sales_Classification	2

dtype: int64

```
df.describe()
```

	Year	Engine_Size_L	Mileage_KM	Price_USD	Sales_Volume
count	50000.000000	50000.000000	50000.000000	50000.000000	50000.000000
mean	2017.015700	3.247180	100307.203140	75034.600900	5067.514680
std	4.324459	1.009078	57941.509344	25998.248882	2856.767125
min	2010.000000	1.500000	3.000000	30000.000000	100.000000
25%	2013.000000	2.400000	50178.000000	52434.750000	2588.000000
50%	2017.000000	3.200000	100388.500000	75011.500000	5087.000000
75%	2021.000000	4.100000	150630.250000	97628.250000	7537.250000
max	2024.000000	5.000000	199996.000000	119998.000000	9999.000000

```
#categorizing columns
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Model                 50000 non-null  object 
 1   Year                  50000 non-null  int64  
 2   Region                50000 non-null  object 
 3   Color                 50000 non-null  object 
 4   Fuel_Type             50000 non-null  object 
 5   Transmission          50000 non-null  object 
 6   Engine_Size_L         50000 non-null  float64
 7   Mileage_KM            50000 non-null  int64  
 8   Price_USD             50000 non-null  int64  
 9   Sales_Volume          50000 non-null  int64  
 10  Sales_Classification  50000 non-null  object 
dtypes: float64(1), int64(4), object(6)
memory usage: 4.2+ MB
```

```
df["Model"].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 50000 entries, 0 to 49999
Series name: Model
Non-Null Count  Dtype
--------------  -----
50000 non-null  object
dtypes: object(1)
memory usage: 390.8+ KB
```

```
df['Model'].unique()
```

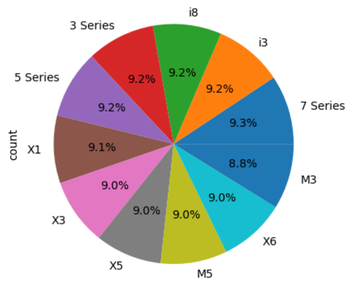
```
array(['5 Series', 'i8', 'X3', '7 Series', 'M5', '3 Series', 'X1', 'M3',
       'X5', 'i3', 'X6'], dtype=object)
```

```
df['Model'].isnull().sum()
```

```
np.int64(0)
```

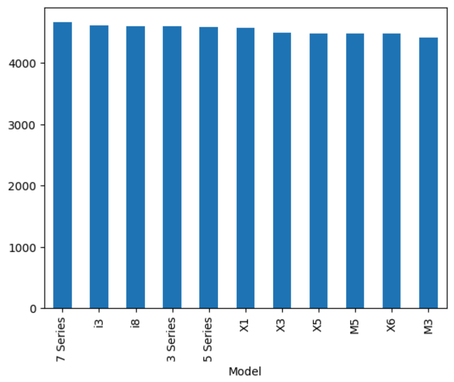
```
df['Model'].value_counts().plot(kind='pie',autopct='%0.1f%%')
```

```
<Axes: ylabel='count'>
```




```
df['Model'].value_counts().plot(kind='bar')
```

```
<Axes: xlabel='Model'>
```




```
df['Mileage_KM'].isnull().sum()
```

```
np.int64(0)
```

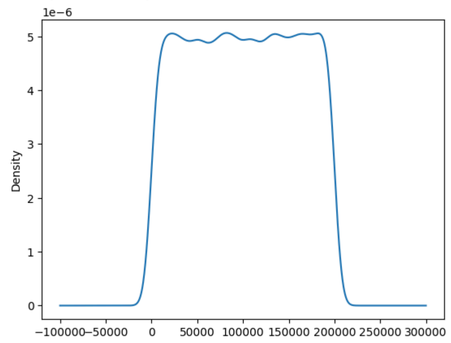
```
df['Mileage_KM'].describe()
```

	Mileage_KM
count	50000.000000
mean	100307.203140
std	57941.509344
min	3.000000
25%	50178.000000
50%	100388.500000
75%	150630.250000
max	199996.000000

dtype: float64

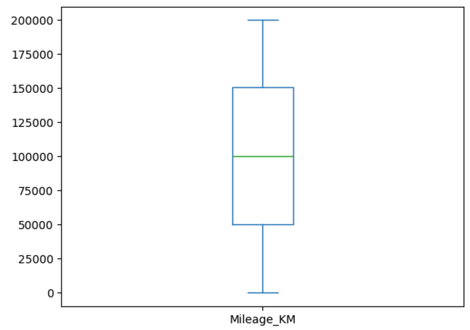
```
df['Mileage_KM'].plot(kind='kde')
```

```
<Axes: ylabel='Density'>
```




```
df['Mileage_KM'].plot(kind='box')
```

```
<Axes: >
```




```
df[df['Mileage_KM'] < 100]
```

	Model	Year	Region	Color	Fuel_Type	Transmission	Engine_Size_L	Mileage_KM	Price_USD	Sales_Volume	Sales_Classification
5291	i3	2010	Africa	White	Petrol	Manual	2.8	3	93933	5336	Low
6621	i3	2020	Africa	Grey	Electric	Automatic	1.8	48	39983	5511	Low
7023	X3	2012	Middle East	White	Diesel	Manual	2.6	58	39244	9531	High
7780	5 Series	2018	South America	Blue	Petrol	Automatic	4.5	21	55195	9860	High
8728	5 Series	2024	Europe	Silver	Diesel	Automatic	3.7	90	66518	5675	Low
12216	5 Series	2018	Asia	Black	Hybrid	Automatic	3.6	83	30514	8176	High
14312	X1	2022	North America	Grey	Diesel	Automatic	1.9	43	103557	5390	Low
14696	X6	2013	Africa	Silver	Petrol	Automatic	1.5	55	102652	5657	Low
15587	X1	2019	Asia	Blue	Petrol	Automatic	3.7	62	69136	4536	Low
15971	7 Series	2014	South America	Grey	Electric	Manual	4.6	63	91779	1345	Low
17180	X5	2017	Asia	Silver	Petrol	Automatic	3.4	29	65476	6454	Low
17858	7 Series	2012	Africa	Grey	Electric	Automatic	2.2	57	51618	584	Low
19201	i8	2014	Africa	Black	Petrol	Manual	2.9	95	95193	1446	Low
20924	3 Series	2016	Europe	Red	Petrol	Automatic	2.7	36	114661	5912	Low
23362	7 Series	2015	North America	Silver	Hybrid	Automatic	2.4	23	78427	348	Low
23453	M3	2016	South America	Silver	Petrol	Manual	2.9	69	55272	4524	Low
26337	M5	2018	Asia	Grey	Diesel	Automatic	4.0	65	32046	4690	Low
30063	X3	2020	South America	Grey	Electric	Automatic	1.7	86	77330	328	Low
30624	3 Series	2022	Asia	Red	Petrol	Automatic	4.4	64	107867	3897	Low
30922	X3	2015	South America	Blue	Diesel	Manual	2.6	68	36450	6269	Low
30987	X5	2021	South America	Black	Diesel	Manual	3.5	65	109305	6443	Low
34471	X5	2017	Asia	Grey	Diesel	Manual	3.8	82	72234	2453	Low
36659	3 Series	2016	North America	Silver	Petrol	Manual	2.9	92	63437	9556	High
43961	M5	2016	Asia	White	Diesel	Manual	4.6	42	48000	1706	Low
49317	i8	2016	North America	Blue	Diesel	Manual	2.2	57	106938	1248	Low

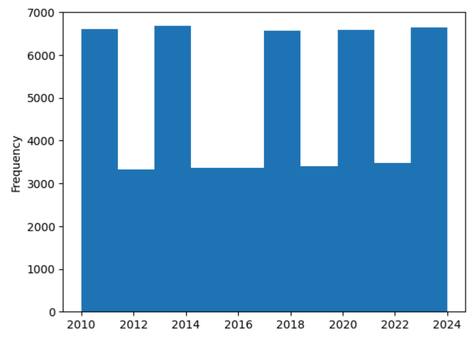
```
df['Year'].describe()
```

	Year
count	50000.000000
mean	2017.015700
std	4.324459
min	2010.000000
25%	2013.000000
50%	2017.000000
75%	2021.000000
max	2024.000000

dtype: float64

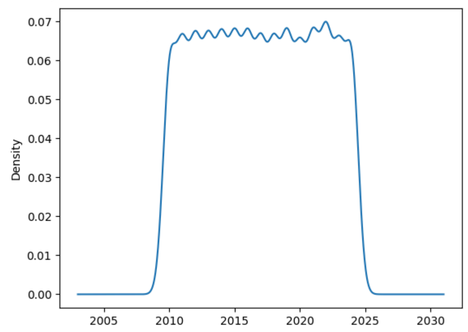
```
df['Year'].plot(kind='hist')
```

```
<Axes: ylabel='Frequency'>
```




```
df['Year'].plot(kind='kde')
```

```
<Axes: ylabel='Density'>
```




```
df['Color'].head()
```

	Color
0	Red
1	Red
2	Blue
3	Blue
4	Black

dtype: object

```
df['Color'].describe()
```

	Color
count	50000
unique	6
top	Red
freq	8463

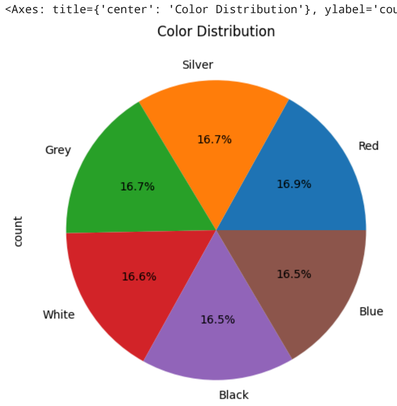
dtype: object

```
df['Color'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 50000 entries, 0 to 49999
Series name: Color
Non-Null Count  Dtype
--------------  -----
50000 non-null  object
dtypes: object(1)
memory usage: 390.8+ KB
```

```
df['Color'].value_counts().plot(kind='pie', autopct='%1.1f%%', figsize=(6,6), title='Color Distribution')
```

```
<Axes: title={'center': 'Color Distribution'}, ylabel='count'>
```




```
df['Fuel_Type'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 50000 entries, 0 to 49999
Series name: Fuel_Type
Non-Null Count  Dtype
--------------  -----
50000 non-null  object
dtypes: object(1)
memory usage: 390.8+ KB
```

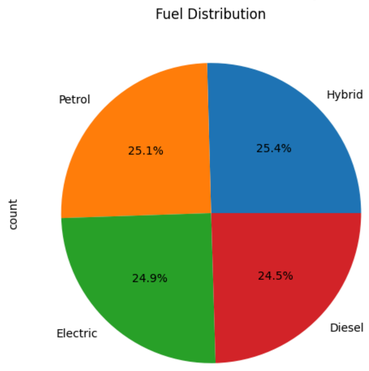
```
df['Fuel_Type'].describe()
```

	Fuel_Type
count	50000
unique	4
top	Hybrid
freq	12716

dtype: object

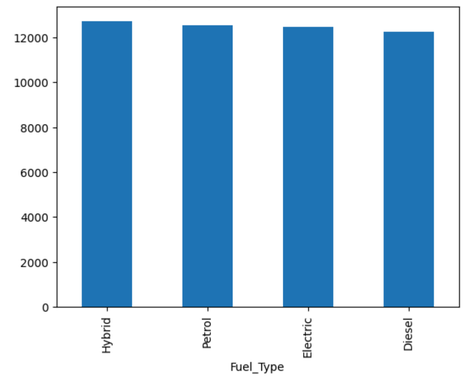
```
df['Fuel_Type'].value_counts().plot(kind='pie',autopct='%1.1f%%', figsize=(6,6), title='Fuel Distribution')
```

```
<Axes: title={'center': 'Fuel Distribution'}, ylabel='count'>
```

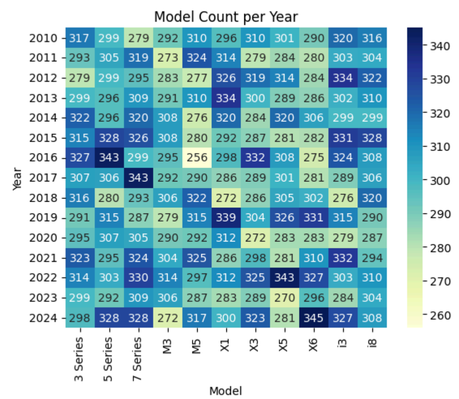



```
df['Fuel_Type'].value_counts().plot(kind='bar')
```

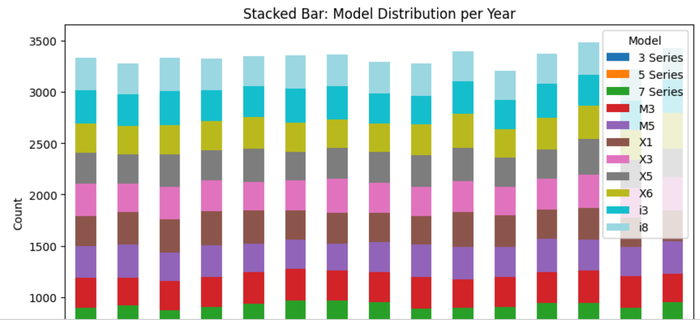
```
<Axes: xlabel='Fuel_Type'>
```



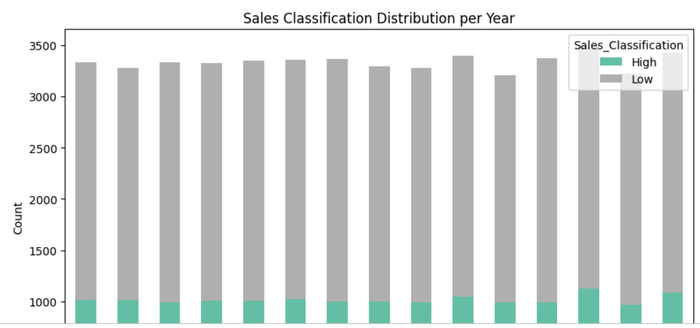

```
sns.heatmap(df.groupby(['Year','Model']).size().unstack(), cmap='YlGnBu', annot=True, fmt='d'); plt.title('Model Count per Year'); plt.show()
```




```
df.groupby(['Year','Model']).size().unstack().plot(kind='bar', stacked=True, figsize=(10,6), colormap='tab20'); plt.title('Stacked Bar: Model Distribution per Year'); plt.ylabel('Count'); plt.show()
```

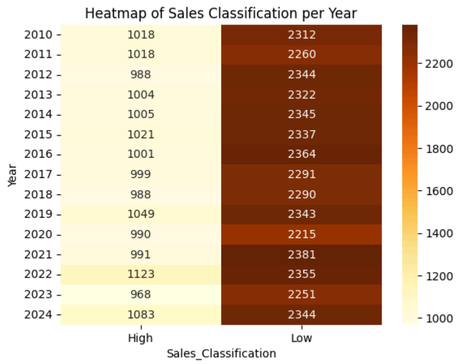



```
df.groupby(['Year','Sales_Classification']).size().unstack().plot(
    kind='bar', stacked=True, figsize=(10,6), colormap='Set2'
)
plt.title('Sales Classification Distribution per Year')
plt.ylabel('Count')
plt.show()
```



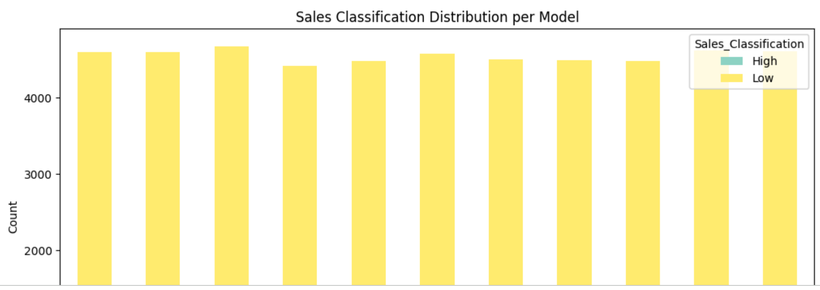

```
import seaborn as sns
sns.heatmap(df.groupby(['Year','Sales_Classification']).size().unstack(),
            annot=True, fmt='d', cmap='YlOrBr')
```

```
plt.title('Heatmap of Sales Classification per Year')
plt.show()
```

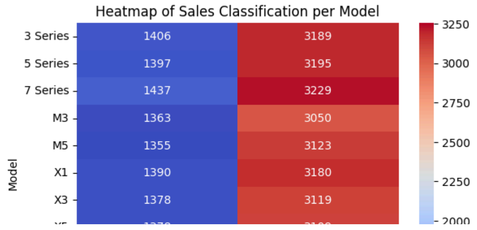



```
import squarify
counts = df.groupby(['Year','Sales_Classification']).size().reset_index(name='count')
squarify.plot(sizes=counts['count'], label=counts['Sales_Classification']+'-'+counts['Year'].astype(str), alpha=.8)
plt.axis('off')
plt.title('Treemap of Sales Classification by Year')
plt.show()
```


```
df.groupby(['Model','Sales_Classification']).size().unstack().plot(
    kind='bar', stacked=True, figsize=(12,6), colormap='Set3'
)
plt.title('Sales Classification Distribution per Model')
plt.ylabel('Count')
plt.show()
```




```
import seaborn as sns
sns.heatmap(df.groupby(['Model','Sales_Classification']).size().unstack(),
            annot=True, fmt='d', cmap='coolwarm')
plt.title('Heatmap of Sales Classification per Model')
plt.show()
```


Heatmap of Sales Classification per Model
Model
3 Series
5 Series
7 Series
M3
M5
X1
X3
1406
3189
1397
3195
1437
3229
1363
3050
1355
3123
1390
3180
1378
3119
3250
3000
2750
2500
2250
2000