# project3-1

June 19, 2024

[6]:
```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import statistics as st
import warnings

# Ignore warnings
warnings.filterwarnings("ignore")

# Set pandas option to display all columns
pd.set_option("display.max_columns", None)

# Specify the file path
file_path = "C:\\Users\\subbiah\\OneDrive\\Desktop\\BIKE.csv"

# Read the CSV file into a DataFrame
df = pd.read_csv(file_path)
print('Dataset dimension: ', df.shape)

# check the attributes in the dataset
print('Attributes in the dataset: ', df.columns.values)

# Display the first few rows of the DataFrame
print(df.head(10))
```

```
Dataset dimension:  (308, 10)
Attributes in the dataset:  ['S.no' 'Bike_company' 'Bike_model'
'Manufactured_year' 'Engine_warranty'
 'Engine_type' 'Fuel_type' 'CC(Cubic capacity)' 'Fuel_Capacity' 'Price']
   S.no   Bike_company     Bike_model  Manufactured_year  Engine_warranty  \
0     1          Bajaj    Avenger 220               2020              5.0
1     2            TVS     Apache RTR               2020              5.0
2     3           Hero        Passion               2020              5.0
3     4          Honda      Activa 3G               2020              5.0
4     5         Suzuki         Access               2020              5.0
5     6  Royal Enfield  Royal Enfield               2019              5.0
6     7         Yamaha        Fascino               2018              5.0
```

```
7     8            KTM        KTM RC                2017              5.0
8     9       Mahindra          Gusto               2016              5.0
9    10       Kawasaki          Ninja               2011              5.0

  Engine_type Fuel_type CC(Cubic capacity) Fuel_Capacity   Price
0      Single    Petrol              220CC     10 Litres  113000
1      V-twin    Petrol              120CC     11 Litres   70000
2       Boxer    Petrol              140CC     12 Litres   85000
3      Single    Petrol              150CC     13 Litres   90000
4      V-twin    Petrol              350CC     14 Litres   65000
5       Boxer    Petrol              350CC     15 Litres  180000
6      Single    Petrol              150CC     16 Litres   65000
7      V-twin    Petrol              300CC     17 Litres  150000
8       Boxer    Petrol              100CC     18 Litres   89000
9      Single    Petrol              170CC     19 Litres   99999
```

[9]:
```
df.isnull().sum()
```

[9]:
```
S.no                 0
Bike_company         0
Bike_model           0
Manufactured_year    0
Engine_warranty      2
Engine_type          0
Fuel_type            0
CC(Cubic capacity)   0
Fuel_Capacity        1
Price                0
dtype: int64
```

[10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 308 entries, 0 to 307
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   S.no                308 non-null    int64
 1   Bike_company        308 non-null    object
 2   Bike_model          308 non-null    object
 3   Manufactured_year   308 non-null    int64
 4   Engine_warranty     306 non-null    float64
 5   Engine_type         308 non-null    object
 6   Fuel_type           308 non-null    object
 7   CC(Cubic capacity)  308 non-null    object
 8   Fuel_Capacity       307 non-null    object
 9   Price               308 non-null    int64
dtypes: float64(1), int64(3), object(6)
```

```
memory usage: 24.2+ KB
```

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

[11]:

| | S.no | Manufactured_year | Engine_warranty | Price |
|---|---|---|---|---|
| count | 308.000000 | 308.000000 | 306.000000 | 3.080000e+02 |
| mean | 154.496753 | 2010.847403 | 6.032680 | 2.548416e+05 |
| std | 89.059088 | 104.747343 | 3.719542 | 2.950174e+05 |
| min | 1.000000 | 202.000000 | 2.000000 | 5.000000e+04 |
| 25% | 77.750000 | 2017.000000 | 4.000000 | 7.500000e+04 |
| 50% | 154.500000 | 2019.000000 | 5.000000 | 1.045000e+05 |
| 75% | 231.250000 | 2020.000000 | 8.000000 | 3.412500e+05 |
| max | 308.000000 | 2050.000000 | 50.000000 | 1.779990e+06 |

[12]:
```
df['Engine_warranty'].fillna(df['Engine_warranty'].mean(), inplace=True)
```

[13]:
```
df['Fuel_Capacity'].fillna(df['Fuel_Capacity'].mode()[0], inplace=True)
```

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

[14]:
```
S.no                  0
Bike_company          0
Bike_model            0
Manufactured_year     0
Engine_warranty       0
Engine_type           0
Fuel_type             0
CC(Cubic capacity)    0
Fuel_Capacity         0
Price                 0
dtype: int64
```

[15]:
```
df.describe()
```

[15]:

| | S.no | Manufactured_year | Engine_warranty | Price |
|---|---|---|---|---|
| count | 308.000000 | 308.000000 | 308.000000 | 3.080000e+02 |
| mean | 154.496753 | 2010.847403 | 6.032680 | 2.548416e+05 |
| std | 89.059088 | 104.747343 | 3.707407 | 2.950174e+05 |
| min | 1.000000 | 202.000000 | 2.000000 | 5.000000e+04 |
| 25% | 77.750000 | 2017.000000 | 4.000000 | 7.500000e+04 |
| 50% | 154.500000 | 2019.000000 | 5.000000 | 1.045000e+05 |
| 75% | 231.250000 | 2020.000000 | 8.000000 | 3.412500e+05 |
| max | 308.000000 | 2050.000000 | 50.000000 | 1.779990e+06 |

[16]:
```
df.duplicated().sum()
```

[16]:
```
0
```

[17]:
```
df.shape
```

[17]:
```
(308, 10)
```

[18]:
```
df.count()
```

[18]:
```
S.no                  308
Bike_company          308
Bike_model            308
Manufactured_year     308
Engine_warranty       308
Engine_type           308
Fuel_type             308
CC(Cubic capacity)    308
Fuel_Capacity         308
Price                 308
dtype: int64
```

[19]:
```
df.head()
```

[19]:

| | S.no | Bike_company | Bike_model | Manufactured_year | Engine_warranty | Engine_type | Fuel_type | CC(Cubic capacity) | Fuel_Capacity | Price |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | Bajaj | Avenger 220 | 2020 | 5.0 | Single | Petrol | 220CC | 10 Litres | 113000 |
| 1 | 2 | TVS | Apache RTR | 2020 | 5.0 | V-twin | Petrol | 120CC | 11 Litres | 70000 |
| 2 | 3 | Hero | Passion | 2020 | 5.0 | Boxer | Petrol | 140CC | 12 Litres | 85000 |
| 3 | 4 | Honda | Activa 3G | 2020 | 5.0 | Single | Petrol | 150CC | 13 Litres | 90000 |
| 4 | 5 | Suzuki | Access | 2020 | 5.0 | V-twin | Petrol | 350CC | 14 Litres | 65000 |

[20]:
```
df.drop('S.no', axis=1, inplace=True)
```

[21]:
```
df.head()
```

[21]:

| | Bike_company | Bike_model | Manufactured_year | Engine_warranty | Engine_type | Fuel_type | CC(Cubic capacity) | Fuel_Capacity | Price |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Bajaj | Avenger 220 | 2020 | 5.0 | Single | Petrol | 220CC | 10 Litres | 113000 |
| 1 | TVS | Apache RTR | 2020 | 5.0 | V-twin | | | | |
| 2 | Hero | Passion | 2020 | 5.0 | Boxer | | | | |
| 3 | Honda | Activa 3G | 2020 | 5.0 | Single | | | | |
| 4 | Suzuki | Access | 2020 | 5.0 | V-twin | | | | |

```
1    Petrol              120CC     11 Litres   70000
2    Petrol              140CC     12 Litres   85000
3    Petrol              150CC     13 Litres   90000
4    Petrol              350CC     14 Litres   65000
```

[22]:
```
numerical_columns=df.columns[df.dtypes !='object']
categorical_columns = df.columns[df.dtypes =='object']
print("numerical_columns",numerical_columns)
print("categorical_columns",categorical_columns)
```

```
numerical_columns Index(['Manufactured_year', 'Engine_warranty', 'Price'],
dtype='object')
categorical_columns Index(['Bike_company', 'Bike_model', 'Engine_type',
'Fuel_type',
       'CC(Cubic capacity)', 'Fuel_Capacity'],
      dtype='object')
```

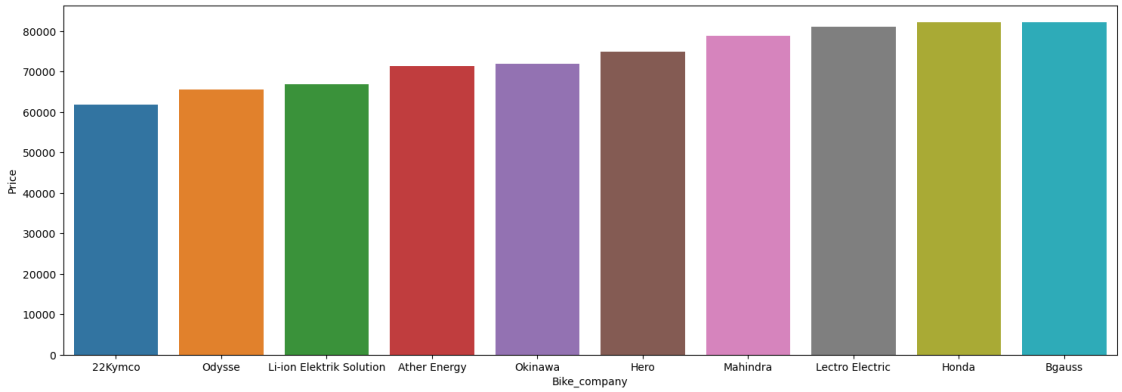
[23]:
```
top_ten_expensive_company = df.groupby("Bike_company")["Price"].mean().
↪sort_values(ascending=True).reset_index().head(10)
top_ten_expensive_company
```

[23]:

| | Bike_company | Price |
|---|---|---|
| 0 | 22Kymco | 61785.000000 |
| 1 | Odysse | 65516.500000 |
| 2 | Li-ion Elektrik Solution | 66790.909091 |
| 3 | Ather Energy | 71377.777778 |
| 4 | Okinawa | 71892.777778 |
| 5 | Hero | 74944.157895 |
| 6 | Mahindra | 78799.352941 |
| 7 | Lectro Electric | 81169.230769 |
| 8 | Honda | 82155.875000 |
| 9 | Bgauss | 82291.000000 |

[24]:
```
plt.figure(figsize=(18,6))
sns.barplot(x="Bike_company",y="Price",data=top_ten_expensive_company)
```

[24]:
```
<Axes: xlabel='Bike_company', ylabel='Price'>
```



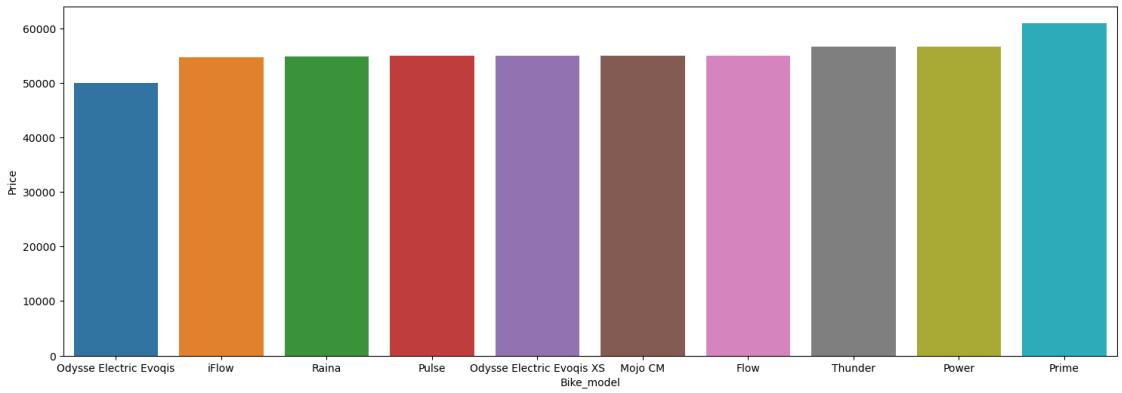

[25]:
```
top_ten_expensive_bike_model = df.groupby("Bike_model")["Price"].mean().
↪sort_values(ascending=True).reset_index().head(10)
top_ten_expensive_bike_model
```

[25]:

| | Bike_model | Price |
|---|---|---|
| 0 | Odysse Electric Evoqis | 50000.0 |
| 1 | iFlow | 54700.0 |
| 2 | Raina | 54800.0 |
| 3 | Pulse | 55000.0 |
| 4 | Odysse Electric Evoqis XS | 55000.0 |
| 5 | Mojo CM | 55000.0 |
| 6 | Flow | 55000.0 |
| 7 | Thunder | 56700.0 |
| 8 | Power | 56700.0 |
| 9 | Prime | 61000.0 |

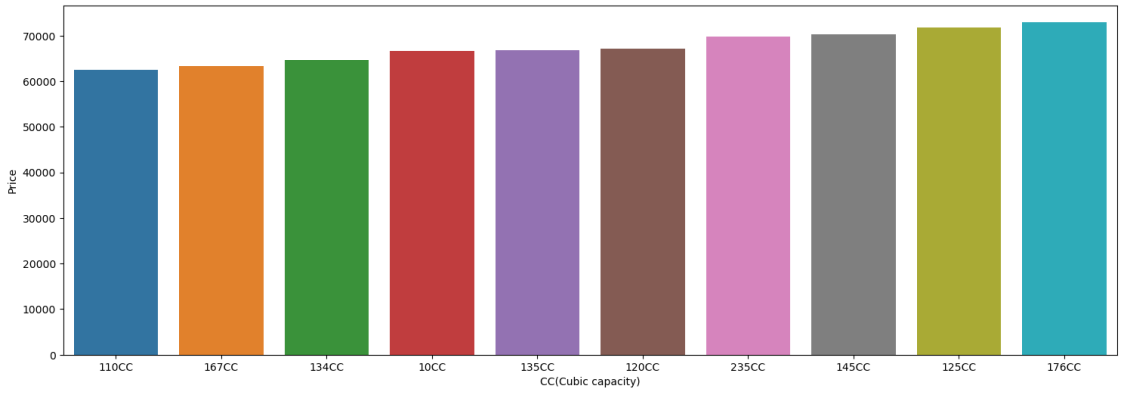
[26]:
```
plt.figure(figsize=(18,6))
sns.barplot(x="Bike_model",y="Price",data=top_ten_expensive_bike_model)
```

[26]: <Axes: xlabel='Bike_model', ylabel='Price'>

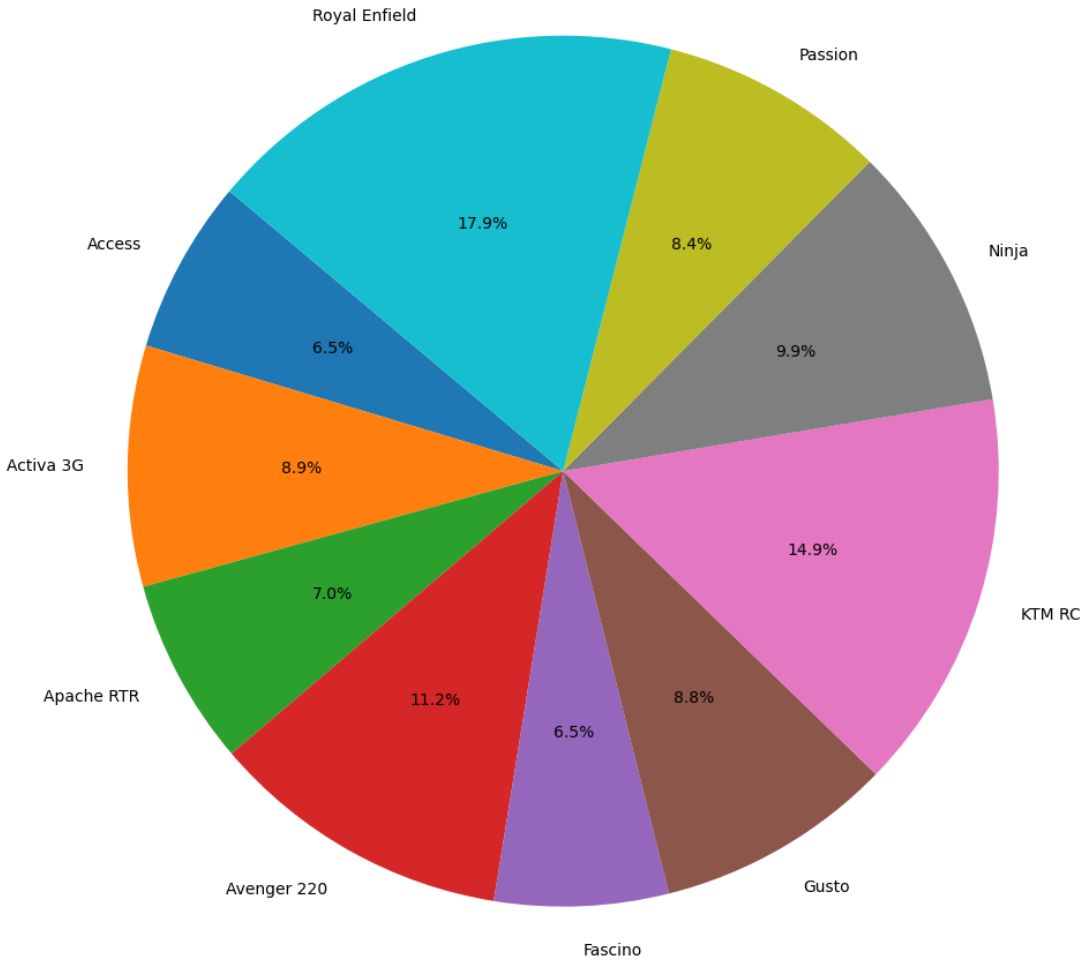


[28]:
```
cc_with_price = df.groupby("CC(Cubic capacity)")["Price"].mean().
 ↪sort_values(ascending=True).reset_index().head(10)

plt.figure(figsize=(18,6))
sns.barplot(x="CC(Cubic capacity)",y="Price",data=cc_with_price)
```

[28]:
```
<Axes: xlabel='CC(Cubic capacity)', ylabel='Price'>
```

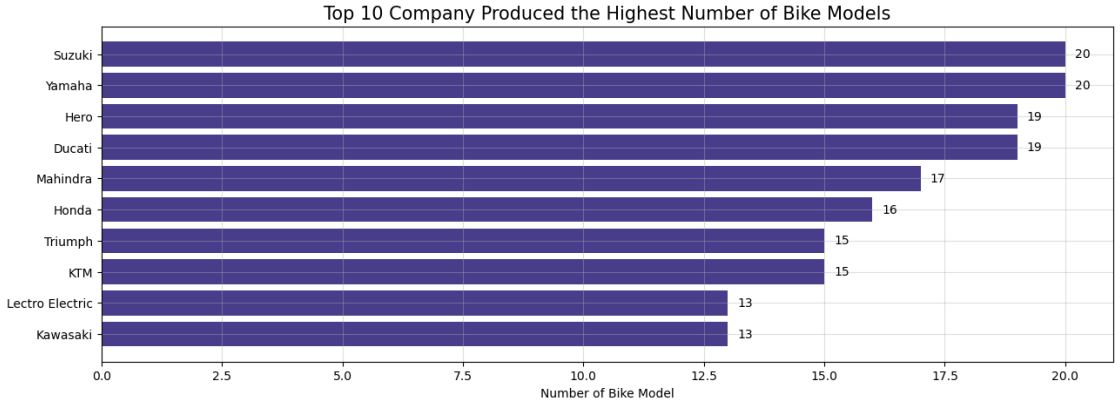



[34]:
```
# Assuming the columns are 'Bike_model' and 'Fuel_Capacity'
# We will aggregate the data to get the total fuel capacity for each Bike_model␣
 ↪for the first 10 entries
bike_fuel_capacity_first_10 = df.head(10).groupby('Bike_model')['Price'].sum()

# Plotting the data as a pie chart
plt.figure(figsize=(10, 10))  # Increase the figure size for better readability
plt.pie(bike_fuel_capacity_first_10, labels=bike_fuel_capacity_first_10.index,␣
 ↪autopct='%1.1f%%', startangle=140)

# Adding a title to the plot
plt.title('Distribution of Bike Models by Total Fuel Capacity (First 10␣
 ↪Entries)', fontsize=16)

# Display the plot
plt.tight_layout()  # Adjusts plot to ensure everything fits without overlapping
plt.show()
```

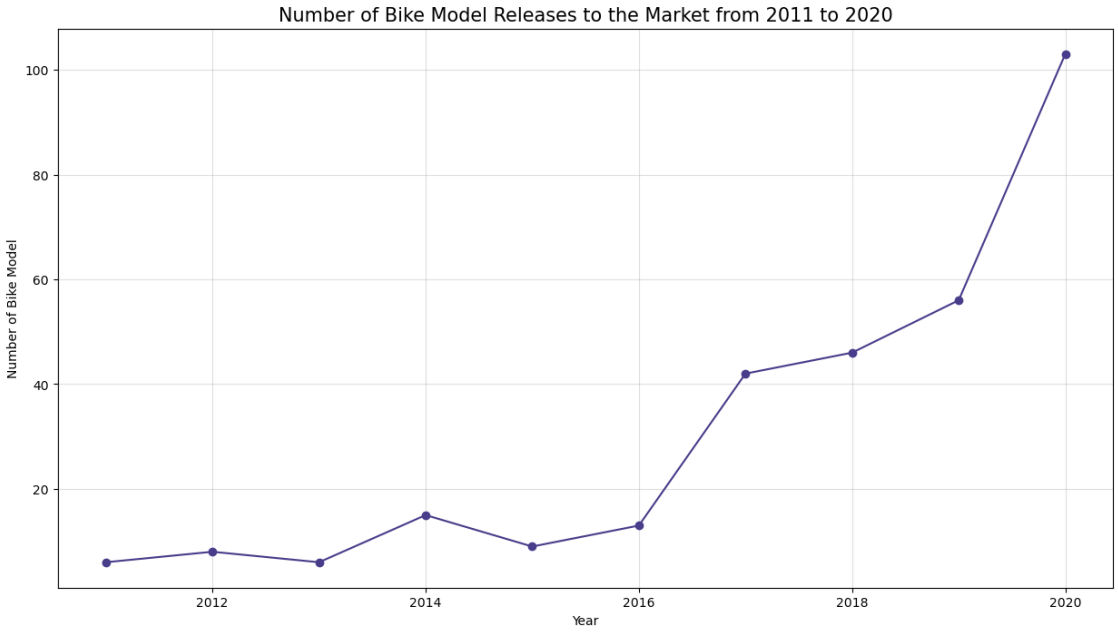



[35]:
```
top10_company = df.Bike_company.value_counts()[:10][::-1]

plt.figure(figsize=(15,5))
plt.barh(y=top10_company.index, width=top10_company.values,␣
 ↪color='darkslateblue')
plt.xlabel('Number of Bike Model')
plt.title('Top 10 Company Produced the Highest Number of Bike Models',␣
 ↪fontsize=15)

for index, values in enumerate(top10_company):
    plt.text(values+0.2, index, str(values), va='center')
```

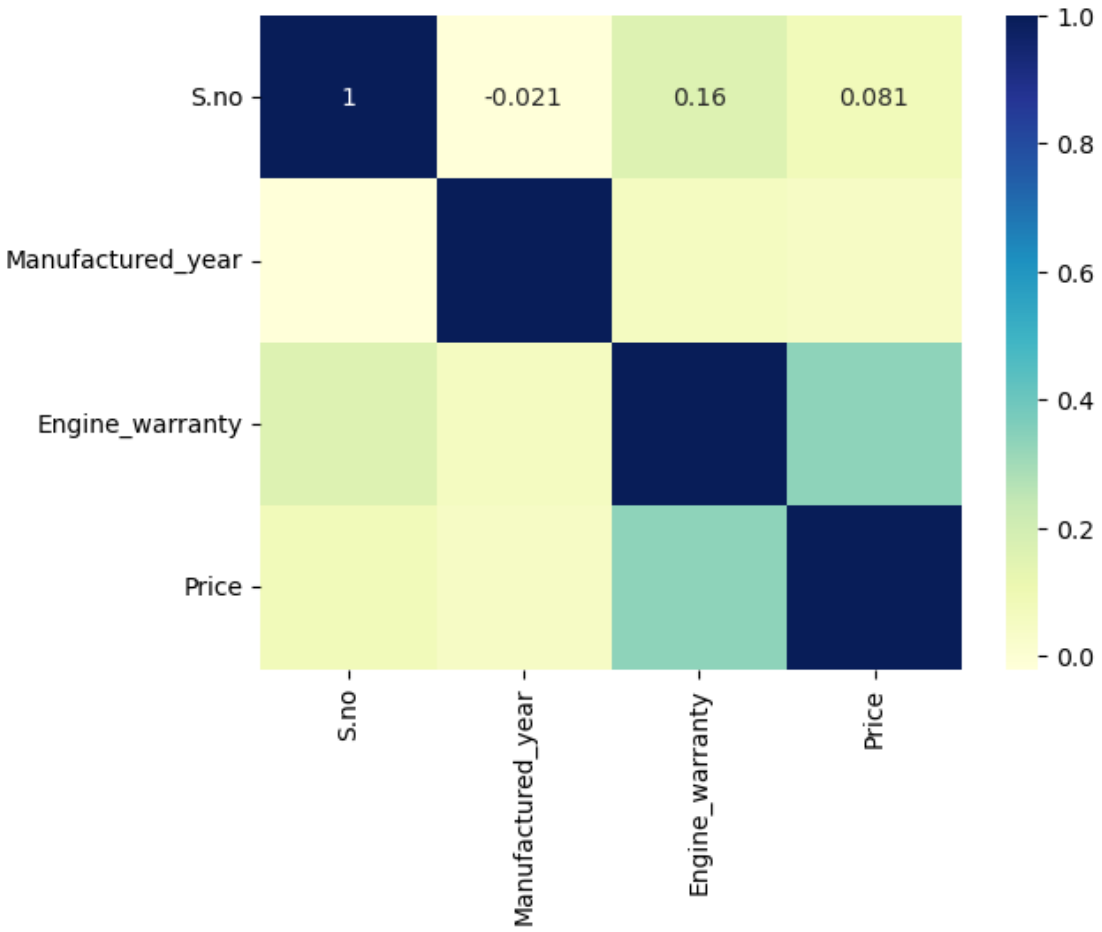
```
plt.grid(alpha=0.4)
plt.show()
```



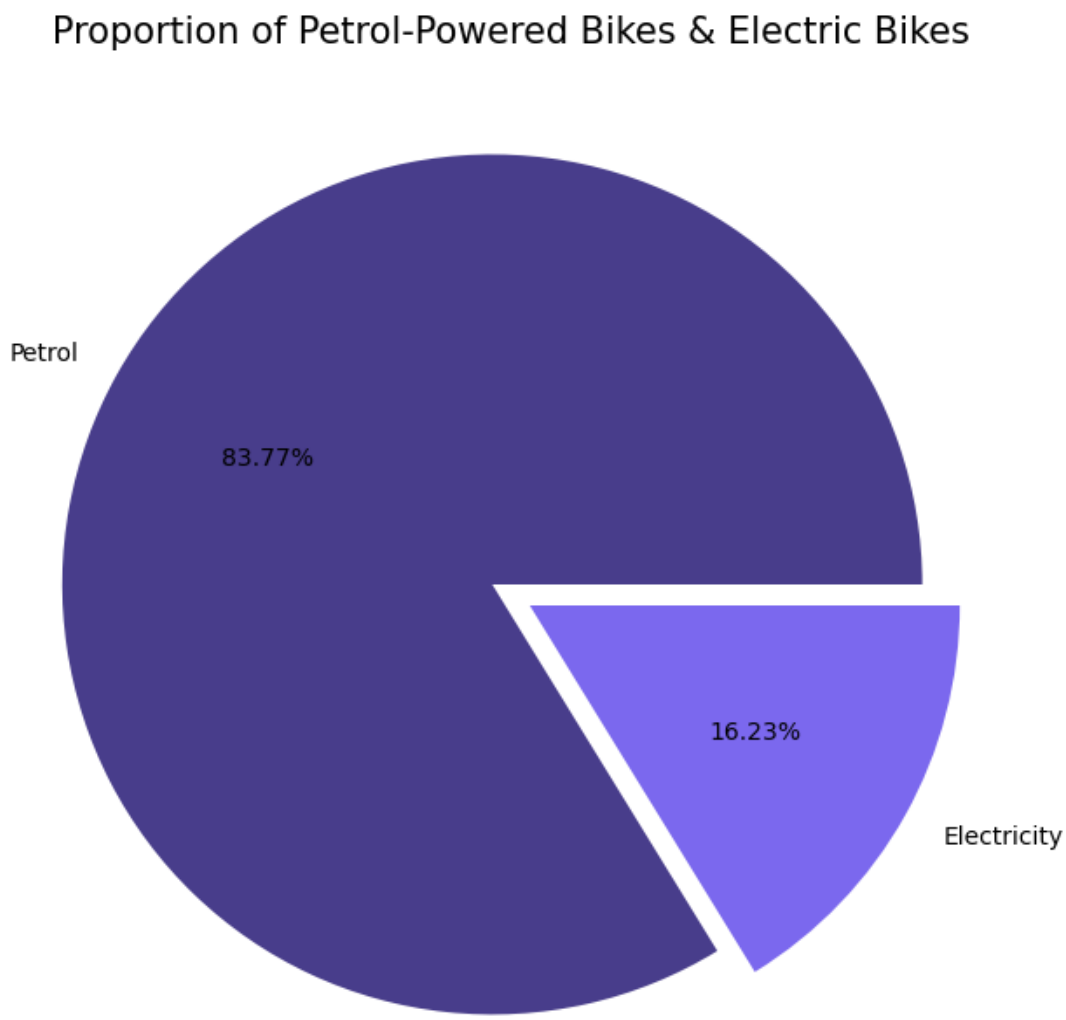

[36]:
```
man_year = df['Manufactured_year'].value_counts()
# there are 4 incorrect data: [1840, 202, 2050, 1790]
# exclude these data from the analysis
man_year_fil = man_year[:10].sort_index()

plt.figure(figsize=(15,8))
plt.plot(man_year_fil.index, man_year_fil.values, color='darkslateblue',␣
 ↪marker='o')
plt.xlabel('Year')
plt.ylabel('Number of Bike Model')
plt.title('Number of Bike Model Releases to the Market from 2011 to 2020',␣
 ↪fontsize=15)
plt.grid(alpha=0.4)
plt.show()
```



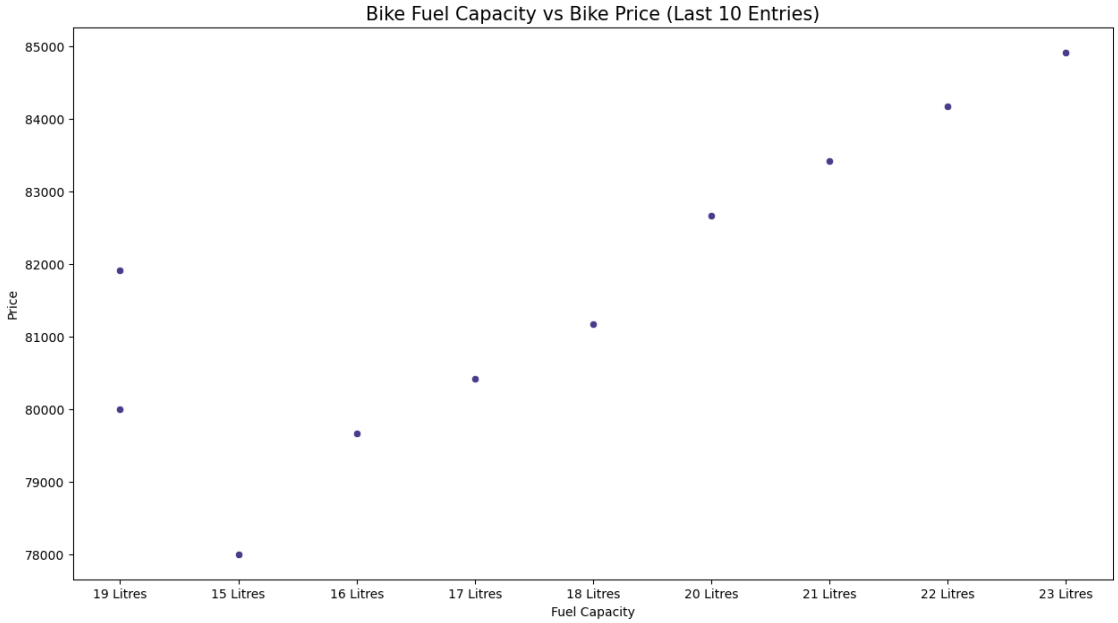

[37]:
```
sns.heatmap(df.corr(numeric_only=True),cmap='YlGnBu',annot=True)
plt.show()
```




[38]:
```
fueltype = df['Fuel_type'].value_counts()

plt.figure(figsize=(8,8))
plt.pie(fueltype.values, labels=fueltype.index, autopct='%.2f%%',␣
 ↪colors=['darkslateblue','mediumslateblue'], explode=([0.05]*len(fueltype.
 ↪index)))
plt.title('Proportion of Petrol-Powered Bikes & Electric Bikes', fontsize=15)
plt.show()
```




[47]:
```
# Assuming the columns are 'Fuel_Capacity' and 'Price'
# Plotting the data as a scatter plot for the last 10 entries
plt.figure(figsize=(15, 8))
sns.scatterplot(data=df.tail(10), x='Fuel_Capacity', y='Price',␣
↪color='darkslateblue')
plt.xlabel('Fuel Capacity')
plt.ylabel('Price')
plt.title('Bike Fuel Capacity vs Bike Price (Last 10 Entries)', fontsize=15)
plt.show()
```


Bike Fuel Capacity vs Bike Price (Last 10 Entries)
Price
85000
84000
83000
82000
81000
80000
79000
78000
19 Litres
15 Litres
16 Litres
17 Litres
18 Litres
20 Litres
21 Litres
22 Litres
23 Litres
Fuel Capacity

[ ]: