Descriptive Analysis

In [24]:
```
import pandas as pd
```

In [33]:
```
import pandas as pd

df = pd.read_csv(r"C:\Users\un\Desktop\Case Study\Data_set2.csv")

print(df)
```

```
          date   price  production  exchange_rate  fuel_price  \
0     1/1/2000   26.94    1781.200         73.150        13.2
1     2/1/2000   25.00    1781.200         73.475        16.2
2     3/1/2000   23.09    1781.200         73.750        16.2
3     4/1/2000   22.13    1781.200         74.300        16.2
4     5/1/2000   21.71    1077.600         74.735        16.2
..         ...     ...         ...            ...         ...
271   8/1/2022  239.24    1461.675        357.880       430.0
272   9/1/2022  228.44    1461.675        365.500       430.0
273  10/1/2022  224.78    1931.200        363.000       430.0
274  11/1/2022  222.28    1931.200        368.500       430.0
275  12/1/2022  218.20    1931.200        367.500       420.0

    Poduction Cost (Rs/Hr) Tax rate
0                  12,500      15%
1                  13,200      15%
2                  14,000      15%
3                  11,500      15%
4                  12,000      15%
..                    ...      ...
271                45,000      15%
272                46,500      15%
273                48,000      15%
274                49,500      15%
275                51,000      15%

[276 rows x 7 columns]
```

In [36]:
```
#Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [37]:
```
#Ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

#Exploratory data analysis

In [39]:
```
df.shape #View dimensions of dataset
```

Out[39]:
```
(276, 7)
```

In [40]:
```
df.head() #Preview the dataset
```

Out[40]:

	date	price	production	exchange_rate	fuel_price	Poduction Cost (Rs/Hr)	Tax rate
0	1/1/2000	26.94	1781.2	73.150	13.2	12,500	15%
1	2/1/2000	25.00	1781.2	73.475	16.2	13,200	15%
2	3/1/2000	23.09	1781.2	73.750	16.2	14,000	15%
3	4/1/2000	22.13	1781.2	74.300	16.2	11,500	15%
4	5/1/2000	21.71	1077.6	74.735	16.2	12,000	15%

In [41]:
```
df.info() #View summary of dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 276 entries, 0 to 275
Data columns (total 7 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   date                    276 non-null    object 
 1   price                   276 non-null    float64
 2   production              276 non-null    float64
 3   exchange_rate           276 non-null    float64
 4   fuel_price              276 non-null    float64
 5   Poduction Cost (Rs/Hr)  276 non-null    object 
 6   Tax rate                276 non-null    object 
dtypes: float64(4), object(3)
memory usage: 15.2+ KB
```

In [42]:
```
#Check for missing values
df.isnull().sum()
```

Out[42]:
```
date                      0
price                     0
production                0
exchange_rate             0
fuel_price                0
Poduction Cost (Rs/Hr)    0
Tax rate                  0
dtype: int64
```

In [43]:
```
#Summary statistics of numerical columns
df.describe()
```

Out[43]:

	price	production	exchange_rate	fuel_price
count	276.000000	276.000000	276.000000	276.000000
mean	68.467842	1925.655851	134.871667	88.548913
std	41.680099	625.223689	53.498613	68.945093
min	20.610000	909.320000	73.150000	13.200000
25%	33.495000	1473.830000	102.578750	50.000000
50%	63.075000	1898.000000	114.750000	84.000000
75%	88.627500	2384.000000	152.047500	104.000000
max	250.070000	3196.750000	368.500000	460.000000

In [44]:
```
#Summary statistics of character columns
df.describe(include=['object'])
```

Out[44]:

	date	Poduction Cost (Rs/Hr)	Tax rate
count	276	276	276
unique	276	176	4
top	1/1/2000	22,500	15%
freq	1	6	210

In [45]:
```
#Summary statistics of all the columns
df.describe(include='all')
```

Out[45]:

	date	price	production	exchange_rate	fuel_price	Poduction Cost (Rs/Hr)	Tax rate
count	276	276.000000	276.000000	276.000000	276.000000	276	276
unique	276	NaN	NaN	NaN	NaN	176	4
top	1/1/2000	NaN	NaN	NaN	NaN	22,500	15%
freq	1	NaN	NaN	NaN	NaN	6	210
mean	NaN	68.467842	1925.655851	134.871667	88.548913	NaN	NaN
std	NaN	41.680099	625.223689	53.498613	68.945093	NaN	NaN
min	NaN	20.610000	909.320000	73.150000	13.200000	NaN	NaN
25%	NaN	33.495000	1473.830000	102.578750	50.000000	NaN	NaN
50%	NaN	63.075000	1898.000000	114.750000	84.000000	NaN	NaN
75%	NaN	88.627500	2384.000000	152.047500	104.000000	NaN	NaN
max	NaN	250.070000	3196.750000	368.500000	460.000000	NaN	NaN

In [46]:
```
#Mean
mean = df['price'].mean()
print(mean)
```

```
68.46784210934781
```

In [47]:
```
#Median
median = df['price'].median()
print(median)
```

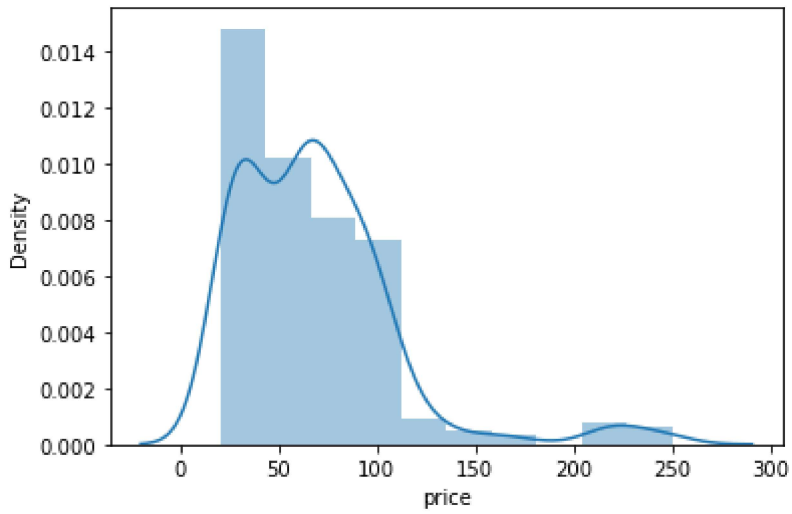
```
63.075
```

In [48]:
```
#Mode
mode = df['price'].mode()
print(mode)
```

```
0    38.37
1    63.58
2    66.37
3    98.00
Name: price, dtype: float64
```

In [49]:
```
#Plot the distribution
data = df['price']
sns.distplot(data, bins=10, hist=True, kde=True, label = 'price')
```

Out[49]: <AxesSubplot:xlabel='price', ylabel='Density'>



In [50]:
```
#Minimum value
df['price'].min()
```

Out[50]: 20.61

In [52]:
```
#Maximum value
df['price'].max()
```

Out[52]: 250.07

In [53]:
```
#Range
df['price'].max() - df['price'].min()
```

Out[53]: 229.45999999999998

In [54]:
```
#Variance
df['price'].var()
```

Out[54]: 1737.2306822566084

In [55]:
```
#Standard deviation
df['price'].std()
```

Out[55]: 41.68009935516719

In [56]:
```
#Median (Q2 or 50th percentile)
Q2 = df['price'].quantile(0.5)
Q2
```

Out[56]: 63.075

In [57]:
```
#Q3 or 75th percentile
Q3 = df['price'].quantile(0.75)
Q3
```

Out[57]: 88.6275

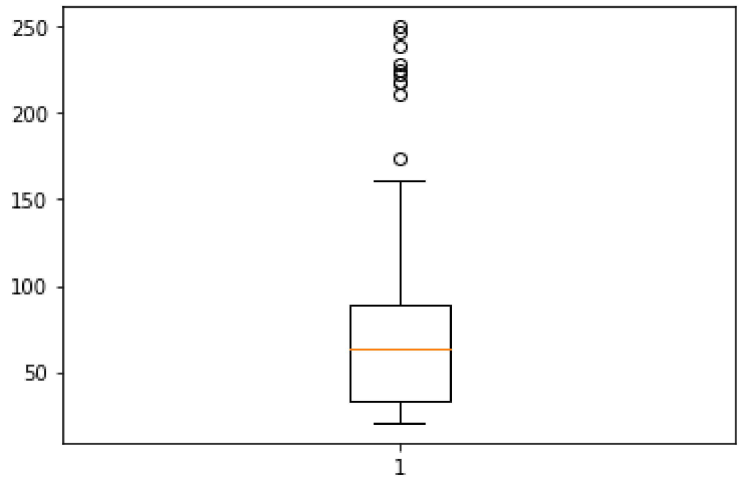
In [58]:
```
#Q1 or 25th percentile
Q1 = df['price'].quantile(0.25)
Q1
```

Out[58]: 33.495000000000005

In [59]:
```
#Interquartile Range
IQR = Q3  - Q1
IQR
```

Out[59]: 55.13249999999999

In [60]:
```
plt.boxplot(df['price'])
plt.show()
```



In [61]:
```
#Skewness
df['price'].skew()
```

Out[61]: 1.9223021275259535

In [62]:
```
#Kurtosis
df['price'].kurt()
```

Out[62]: 5.337302994401137

In [ ]:
```

```