```
# Outlier: A data value that is numerically distant from a data set.
#Outliers imapct statistical measures such as mean, variance, standered devaition.
#utliers less affect the median and IQR.
```

```
import matplotlib.pyplot as plt
import pandas as pd
```

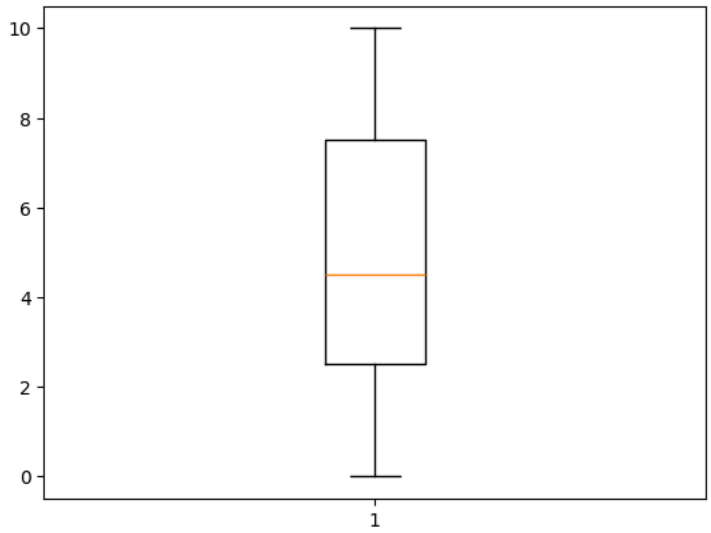
```
df = pd.DataFrame({"x":[1,3,4,5,7,9,0,10]})
```

```
plt.boxplot(df["x"])
plt.show
```

```
matplotlib.pyplot.show
def show(*args, **kwargs) -> None

/usr/local/lib/python3.11/dist-packages/matplotlib/pyplot.py
Display all open figures.

Parameters
----------
block : bool, optional
```

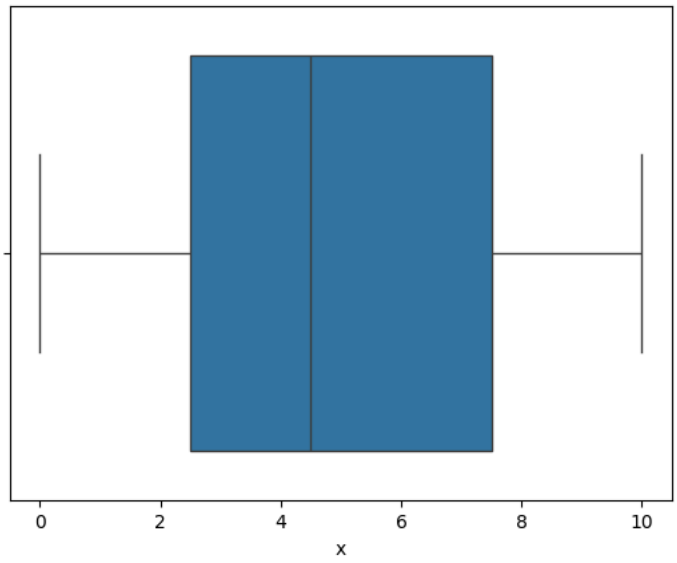



```
import seaborn as sns
sns.boxplot(x=df["x"])
plt.show
```

```
matplotlib.pyplot.show
def show(*args, **kwargs) -> None
```

/usr/local/lib/python3.11/dist-packages/matplotlib/pyplot.py
Display all open figures.

Parameters
----
block : bool, optional




```
# To extract outliers data
```

```
df[(df["x"]<4) | (df["x"]>9)]
```

	x
0	1
1	3
6	0
7	10

```
df = pd.DataFrame({"gender":["F","F","F","F","M","M","F"],"Marks":[30,40,32,44,56,62,77],
                   "no_of_assignments":[1,1,2,1,2,3,2]})
df
```

	gender	Marks	no_of_assignments
0	F	30	1
1	F	40	1
2	F	32	2
3	F	44	1
4	M	56	2
5	M	62	3
6	F	77	2

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
#Frequency Distribution
#Graphical representation of variable with corresponding frequency.
```

```
df["gender"].unique()
```

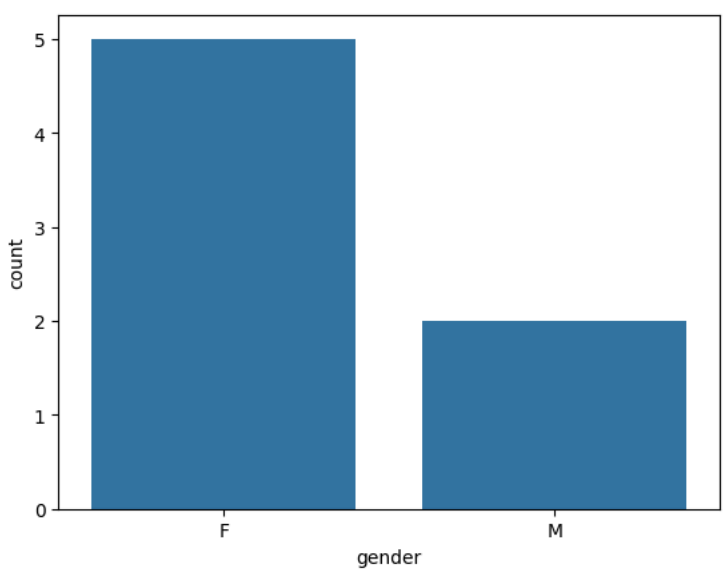
```
array(['F', 'M'], dtype=object)
```

```
df["gender"].value_counts()
```

gender	count
F	5
M	2

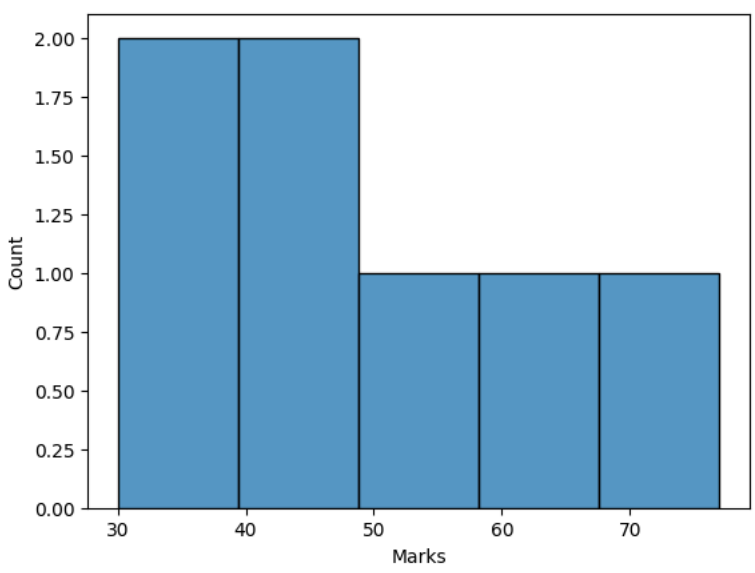
dtype: int64

```
sns.countplot(x=df["gender"])
plt.show()
```



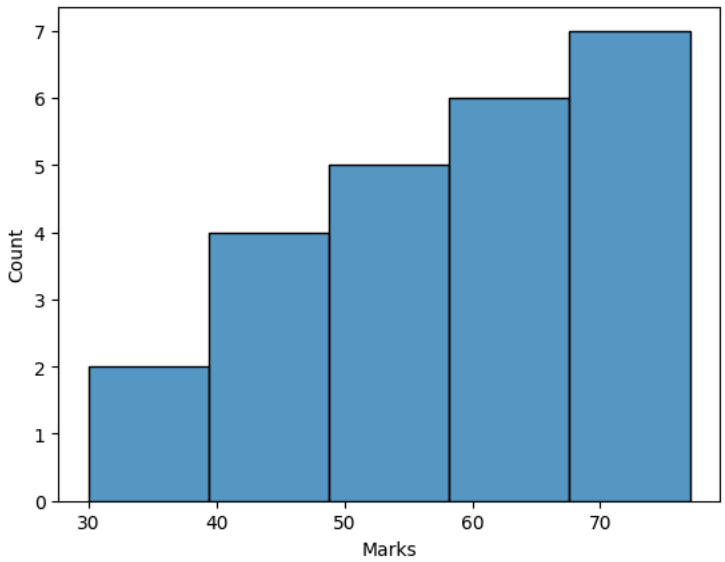

```
#continous Frequency Distribution:
#Graphical representation of continous variable with corresponding frequency.
```

```
sns.histplot(df['Marks'], bins=5, stat = 'count')
plt.show()
```




```
# Cumulative Frequency Distribution
```

```
sns.histplot(df['Marks'], bins=5, stat = 'count', cumulative=True)
plt.show()
```




```
#Probability
# chanve of occurances
# Sum of all possiblities = 1
```

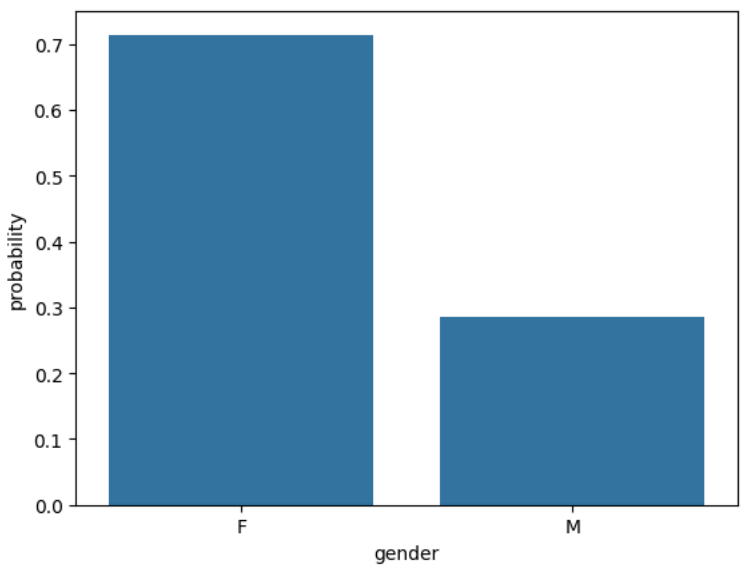
```
df["gender"].value_counts()/len(df)
```

gender	count
F	0.714286
M	0.285714

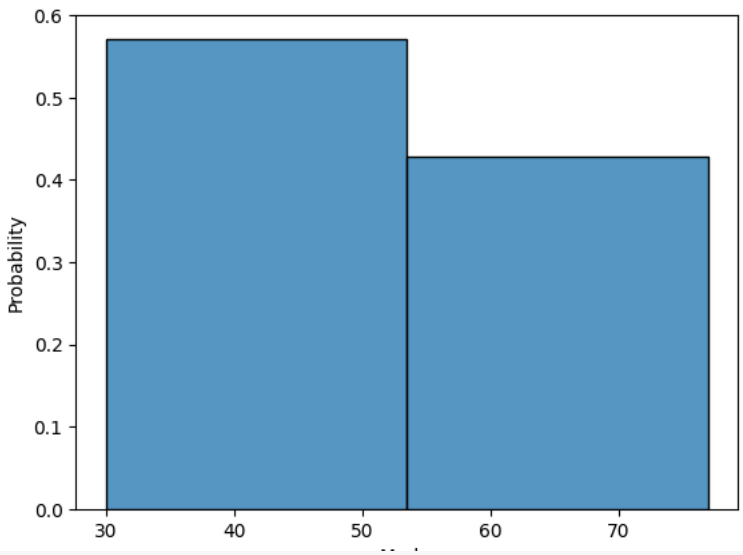
dtype: float64

```
#Probability distribution
```

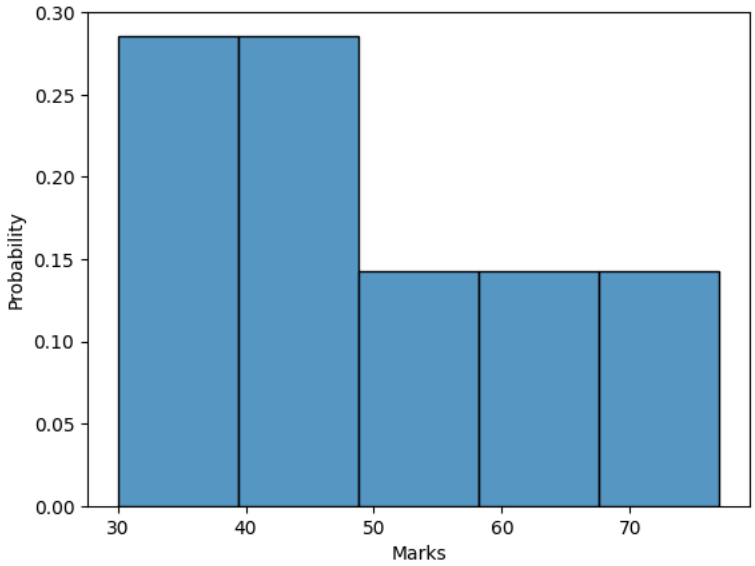
```
sns.countplot(x=df["gender"],stat="probability")
plt.show()
```




```
# Continous Probability Distribution.
sns.histplot(df["Marks"],bins=2, stat="probability")
plt.show()
```




```
#Continous Probability Distribution
sns.histplot(df['Marks'], bins=5, stat= "probability")
plt.show()
```




```
# Cumulative probability distribution.
sns.histplot(df['Marks'],bins=5, stat="probability", cumulative=True)
plt.show()
```