```
# Measure of shape (3rd Business Moment)
```

```
# skewness
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df = pd.DataFrame({"x":[1,2,3,4,5]})
df
```

	x
0	1
1	2
2	3
3	4
4	5

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

```
df['x'].skew()
```

```
np.float64(0.0)
```

```
df = pd.DataFrame({"x":[0,1,2,4,5,6,7],
                   "y":[1,2,3,4,5,6,7],
                   "z":[2,3,4,33,44,55,6]})
df
```

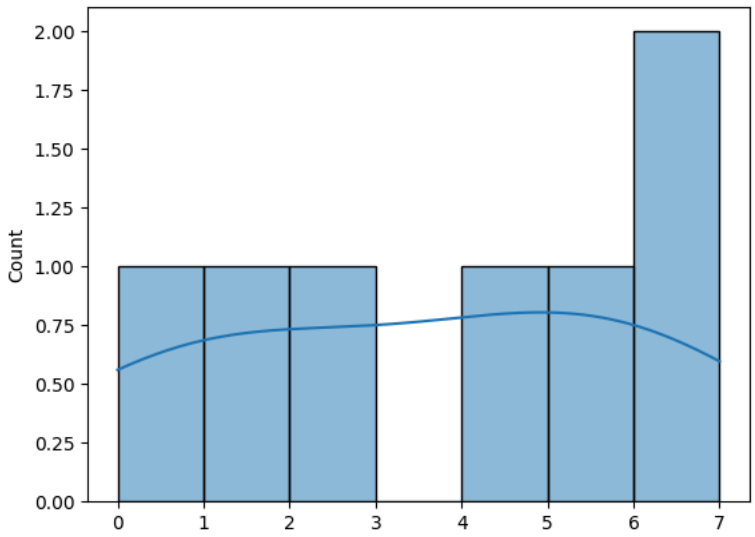
	x	y	z
0	0	1	2
1	1	2	3
2	2	3	4
3	4	4	33
4	5	5	44
5	6	6	55

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

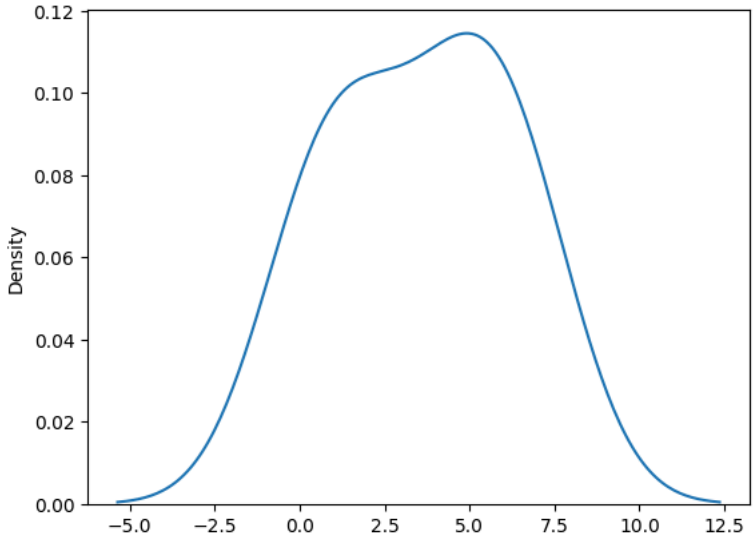
```
print("Mean of x:",df["x"].mean())
print("Median of x:",df["x"].median())
print("Skewness of x:",df["x"].skew())
```

```
Mean of x: 3.5714285714285716
Median of x: 4.0
Skewness of x: -0.11221840987727329
```

```
sns.histplot(df["x"],bins=7,kde=True)
plt.show()
```



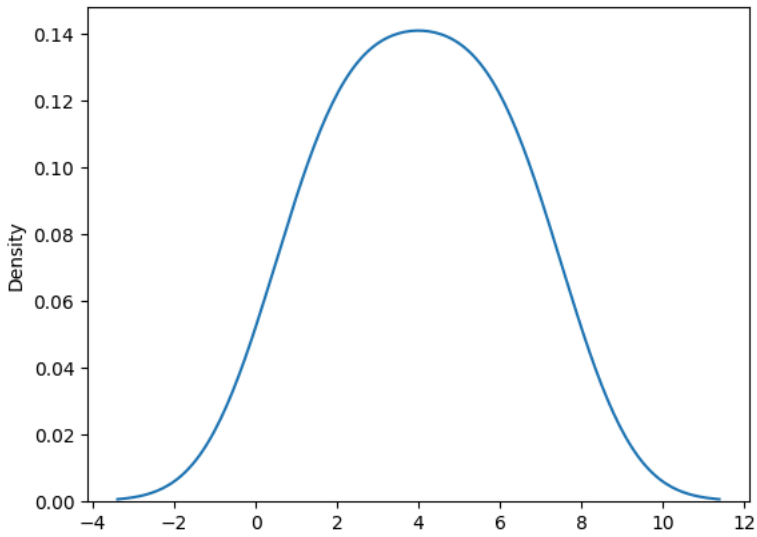

```
sns.kdeplot(df['x'])
plt.show()
```



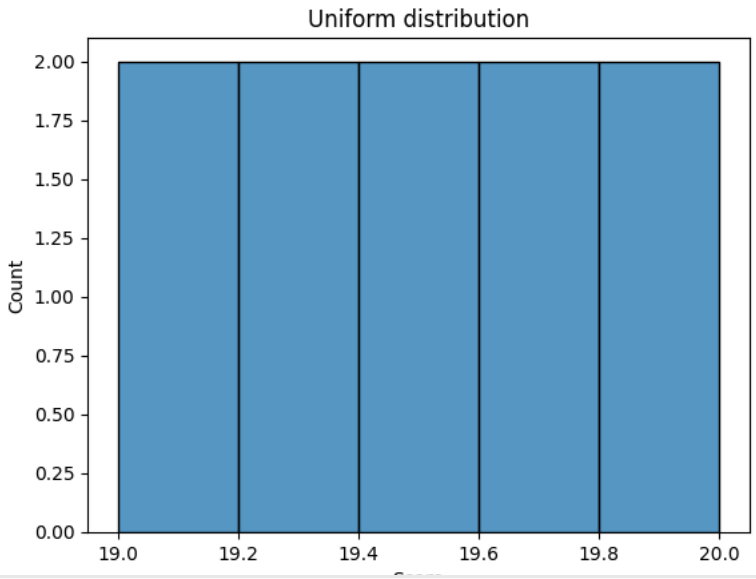

```
print ("mean of y:",df["y"].mean())
print ("median of y:",df["y"].median())
print ("skewness of y:",df["y"].skew())
```

```
mean of y: 4.0
median of y: 4.0
skewness of y: 0.0
```

```
sns.kdeplot(df['y'])
plt.show()
```




```
ds1 = pd.DataFrame({'Score':[19,19.1,19.2,19.3,19.4,19.5,19.6,19.7,19.8,20]})
sns.histplot(ds1['Score'])
plt.title('Uniform distribution')
plt.show()
```




```
# Calculate probability using Z-score
```

```
Zvalue = (49-60)/10
Zvalue
```

```
-1.1
```

```
from scipy import stats
stats.norm.cdf(Zvalue)
```

```
np.float64(0.13566606094638267)
```

Q. When measuring the heights of all students as a local university, it was found that it was normally distributed with a mean height of 5.5 feet and standard deviation of 0.5 feet. What proportion of students are between 5.81 feet to 6.3 feet?

```
z1 = (5.81-5.5)/0.5
z2 = (6.3-5.5)/0.5
p1 = stats.norm.cdf(z1)
p2 =stats.norm.cdf(z2)
print("probability of students less than 6.3:",p2)
```

```
probability of students less than 6.3: 0.945200708300442
```

```
# Central Limit Theorem:
```

```
# Calculate zscore using probability
```

```
from scipy import stats
stats.norm.ppf(0.99)
```

```
np.float64(2.3263478740408408)
```

```
#Bivariate & Multivariate Analysis
```

```
import pandas as pd
```

```
df = pd.DataFrame({"x":[11,22,13,24,30], "y":[10,9,8,7,6],"z":[18,19,21,22,40]})
df
```

	x	y	z
0	11	10	18
1	22	9	19
2	13	8	21
3	24	7	22

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

```
df.cov()
```

	x	y	z
x	62.50	-10.00	54.25
y	-10.00	2.50	-11.75

```
# Correlation
df.corr()
```

	x	y	z