Day31_Descriptive_Statistics

June 28, 2025

Day 31 – Descriptive Statistics in Python

Introduction

Descriptive statistics is the process of summarizing and describing the main features of a dataset. It gives you a quick overview of the shape, center, and spread of the data before performing any deeper analysis or machine learning. It helps answer questions like: - What is the average value? - How spread out is the data? - Are there any outliers? - How symmetrical is the distribution?

1 Central Tendency

Central tendency is about finding the center or the typical value in a dataset. The three common measures are:

1.1 Mean (Average)

The mean is the arithmetic average of a dataset. It gives a central value but is sensitive to extreme values (outliers).

[1]:

```python
import numpy as np

data = [10, 20, 30, 40, 50]
mean_value = np.mean(data)
print("Mean:", mean_value)
```

```
Mean: 30.0
```

Formula (Manual Calculation):

```
Mean = (x  + x  + ... + x ) / n
Example: (10 + 20 + 30 + 40 + 50) / 5 = 150 / 5 = 30
```

1.2 Median (Middle Value)

The median gives a better idea of the center when the data has outliers or is skewed.

[2]:

```python
median_value = np.median(data)
print("Median:", median_value)
```

```
Median: 30.0
```

Steps to Calculate Manually:

1. Arrange data in ascending order
2. If number of values (n) is odd: median is the middle value
3. If n is even: median is the average of the two middle values

```
Example:
Data: [10, 20, 30, 40, 50] (n=5, odd) → Median = 30
Data: [10, 20, 30, 40] (n=4, even) → Median = (20+30)/2 = 25
```

1.3 Mode (Most Frequent Value)

Mode is useful for categorical or repeated data. A dataset can have: - No mode (no repeats) - One mode (unimodal) - Two modes (bimodal) - More than two modes (multimodal)

[12]:

```python
from scipy import stats
mode_result = stats.mode(data, keepdims=True)
print("Mode:", mode_result.mode[0])
```

```
Mode: 10
```

Steps to Calculate Manually: 1. Count the frequency of each value in the dataset 2. The value that appears the most times is the mode

```
Example:
Data: [10, 20, 20, 30, 40] → Mode = 20 (appears twice)
```

2 Measures of Asymmetry (Skewness) and Tailedness (Kurtosis)

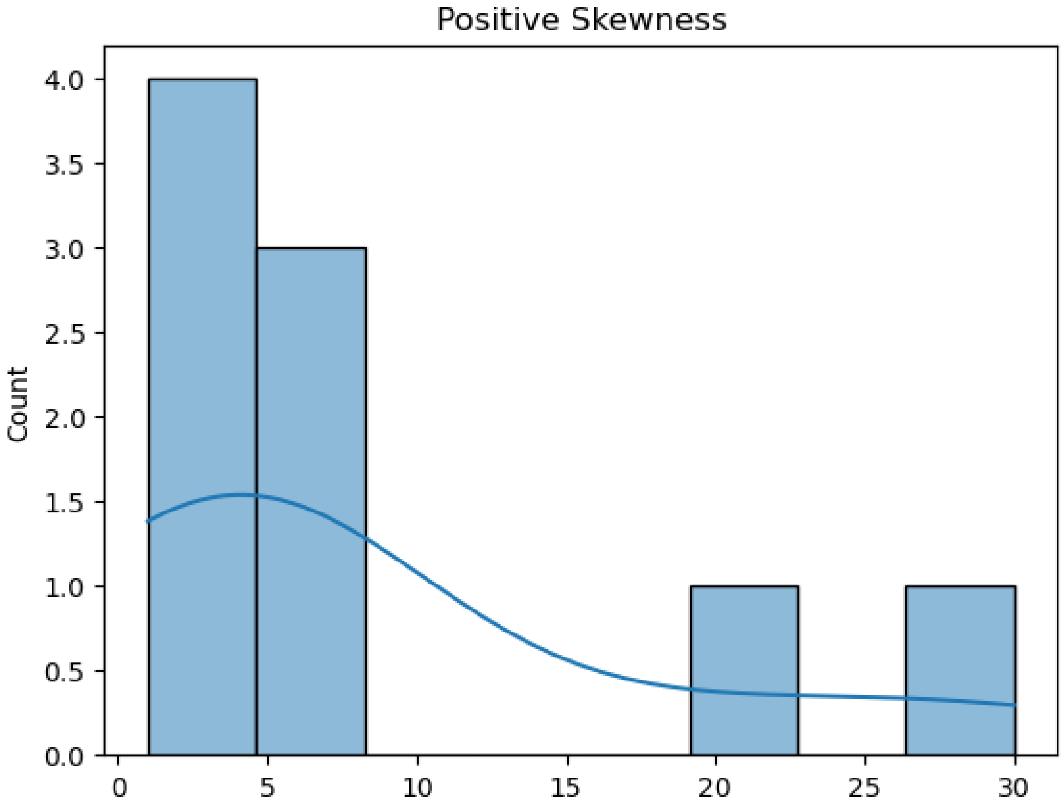
These measures help describe the shape of the data distribution.

2.1 Skewness

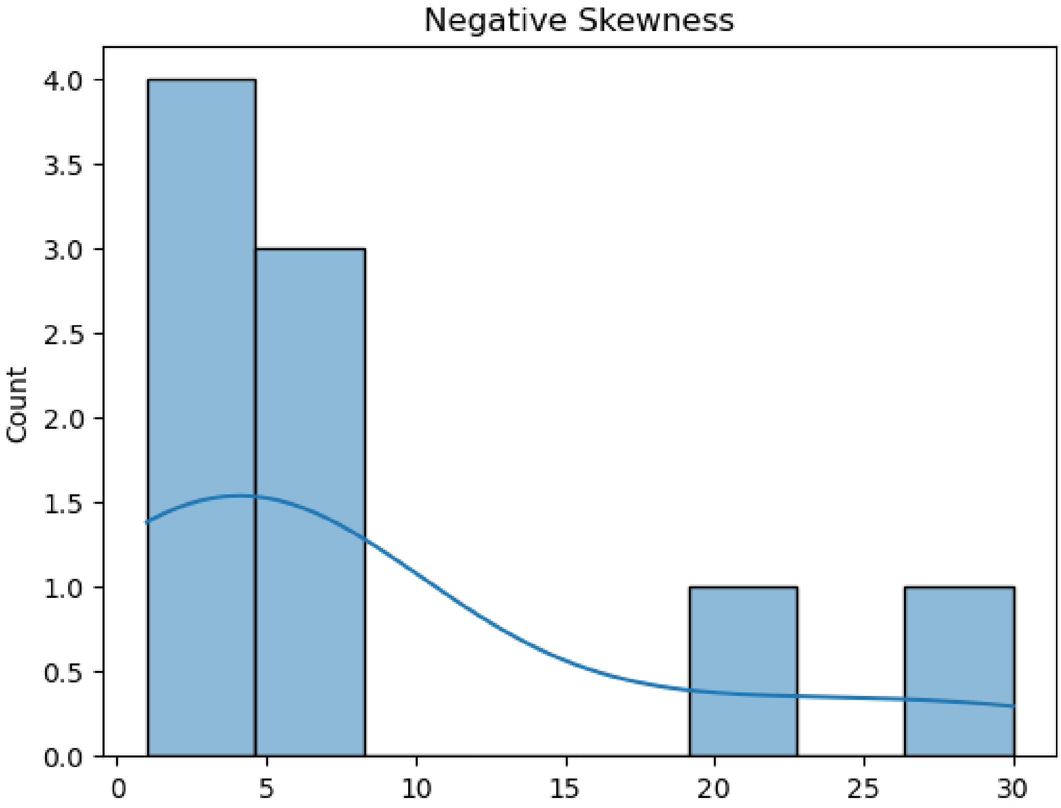
[13]:

```python
from scipy.stats import skew
import matplotlib.pyplot as plt
import seaborn as sns

# Positive skewed data
data_pos_skew = [1, 2, 3, 4, 5, 6, 7, 20, 30]
sns.histplot(data_pos_skew, kde=True)
plt.title("Positive Skewness")
plt.show()
print("Skewness (Positive):", skew(data_pos_skew))

# Negative skewed data
data_neg_skew = [30, 20, 7, 6, 5, 4, 3, 2, 1]
sns.histplot(data_neg_skew, kde=True)
plt.title("Negative Skewness")
plt.show()
print("Skewness (Negative):", skew(data_neg_skew))

# Symmetrical data
```

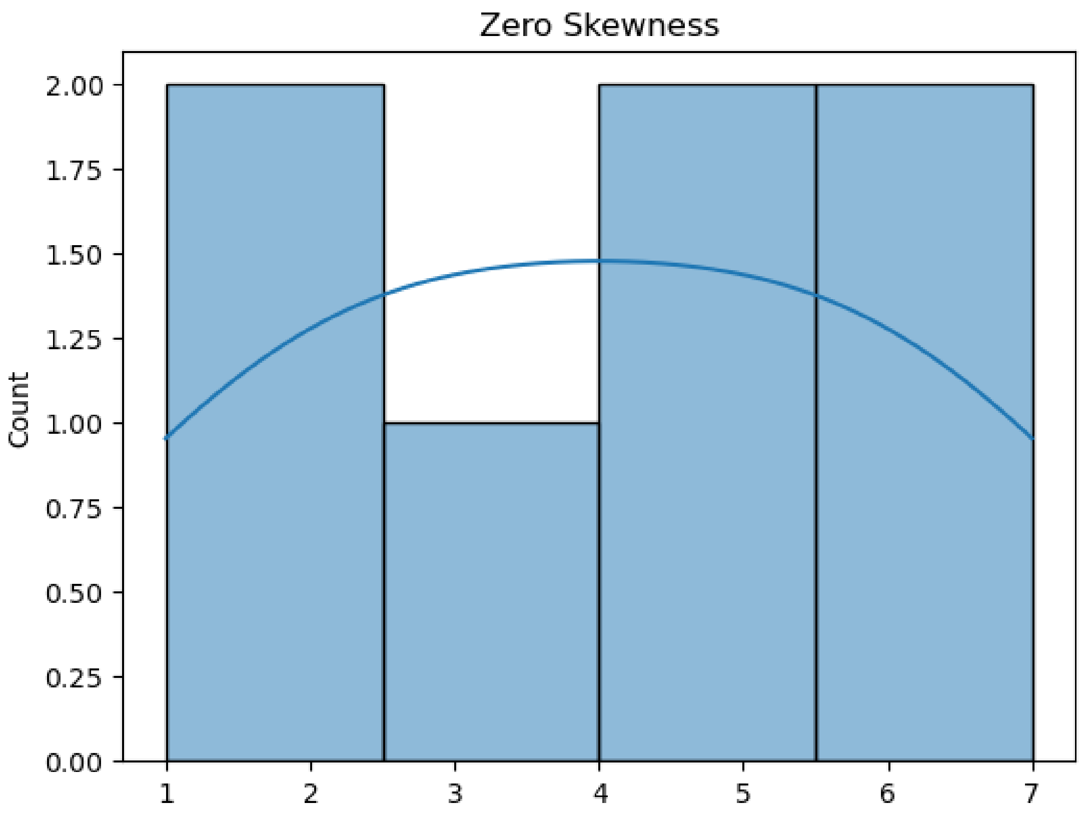
```python
data_sym = [1, 2, 3, 4, 5, 6, 7]
sns.histplot(data_sym, kde=True)
plt.title("Zero Skewness")
plt.show()
print("Skewness (Zero):", skew(data_sym))
```



```
Skewness (Positive): 1.429343971606444
```



```
Skewness (Negative): 1.429343971606444
```



```
Skewness (Zero): 0.0
```

Explanation:

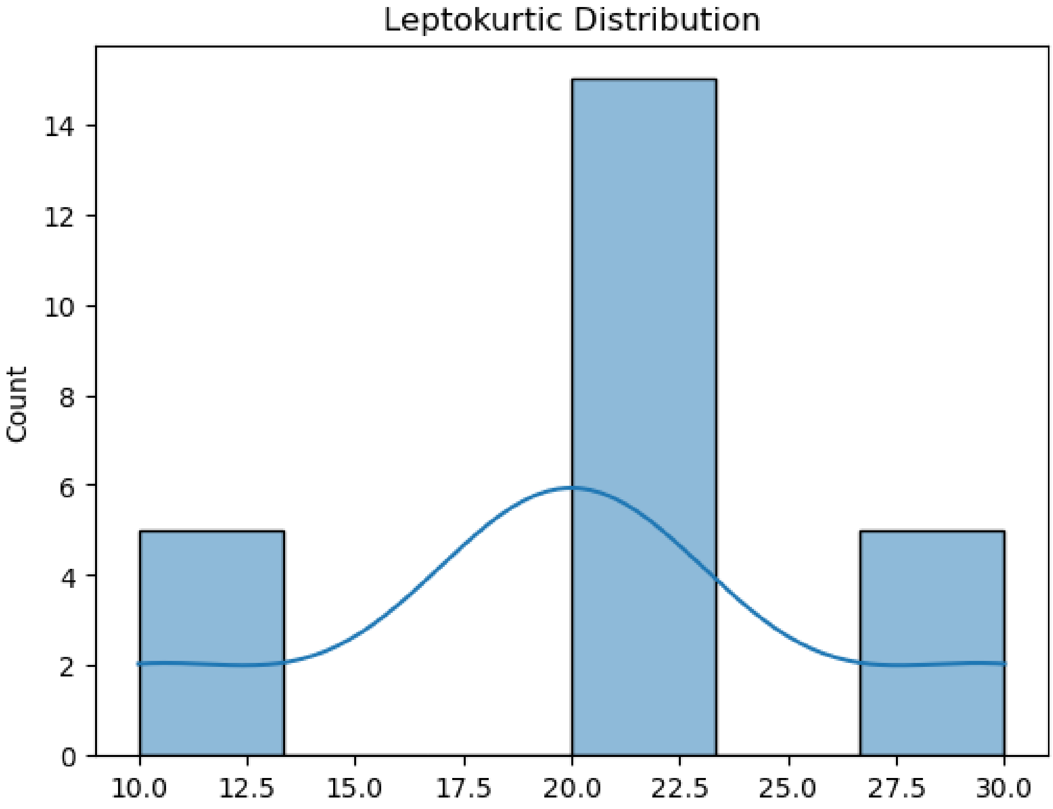
Skewness measures the asymmetry of the data distribution:

- **Positive skew** (right skew): tail is on the right; mean > median > mode
- **Negative skew** (left skew): tail is on the left; mean < median < mode
- **Zero skew**: symmetric data (bell curve); mean = median = mode

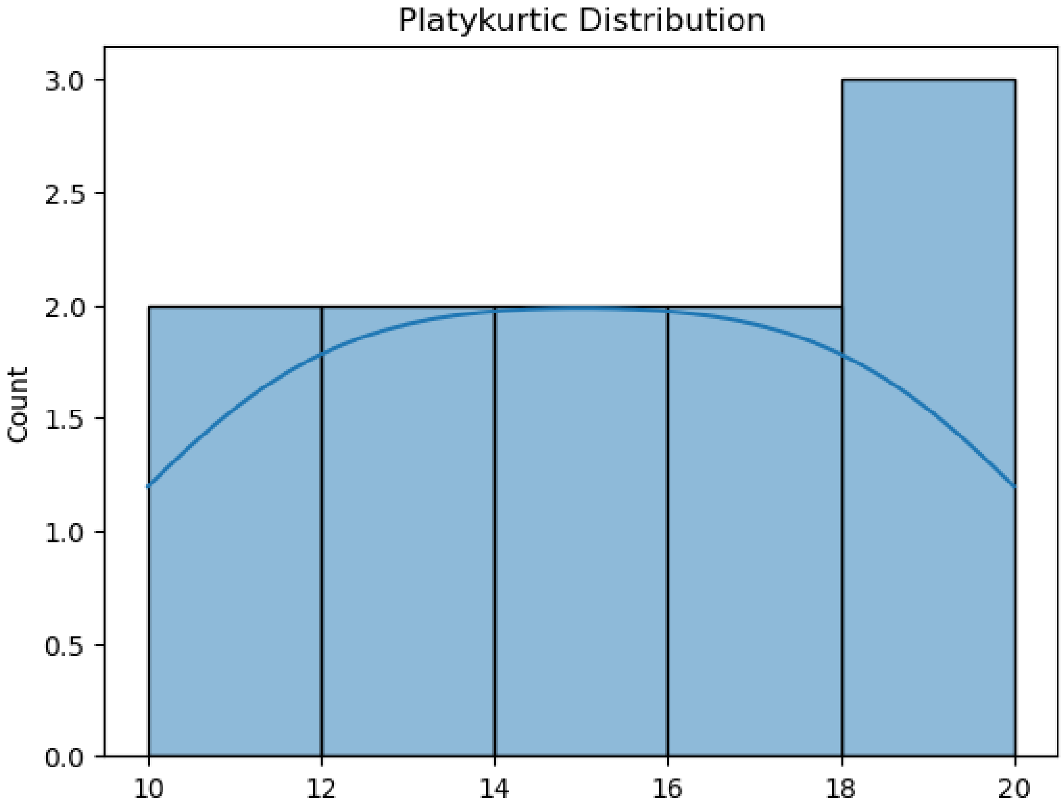
2.2 Kurtosis

[14]:
```
from scipy.stats import kurtosis

# Leptokurtic (high peak, heavy tails)
data_lepto = [10]*5 + [20]*15 + [30]*5
sns.histplot(data_lepto, kde=True)
plt.title("Leptokurtic Distribution")
plt.show()
print("Kurtosis (Leptokurtic):", kurtosis(data_lepto, fisher=False))

# Platykurtic (flat peak)
data_platy = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

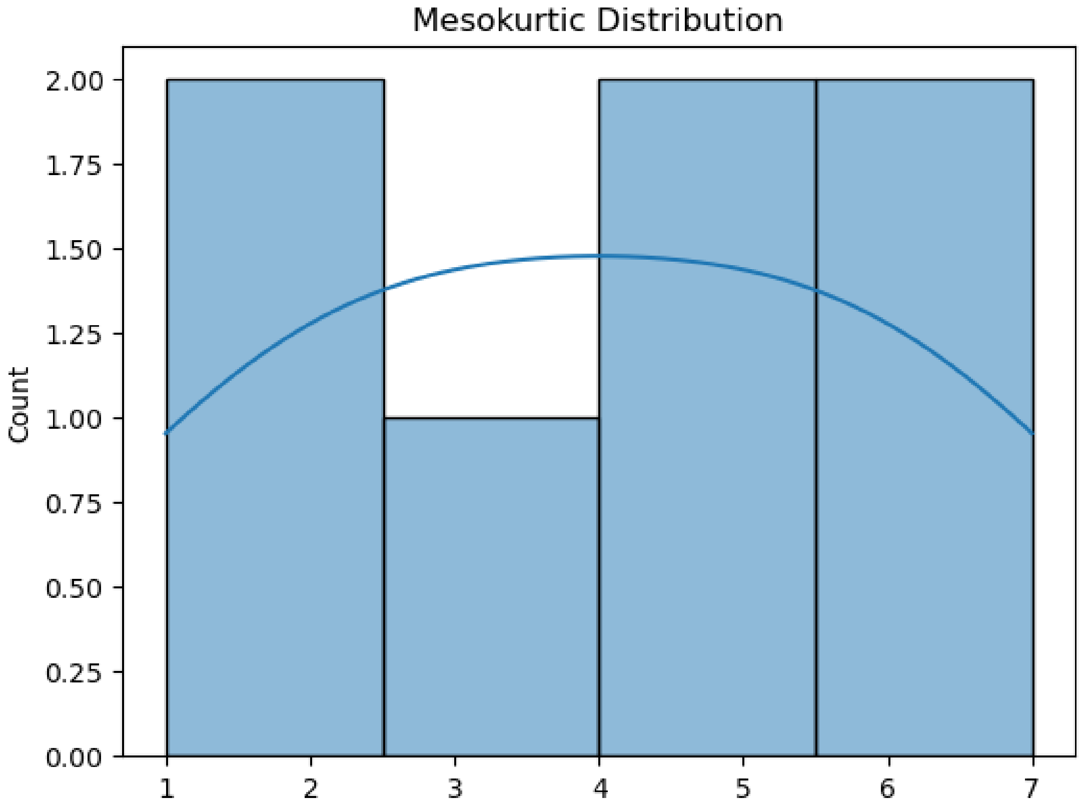
```
sns.histplot(data_platy, kde=True)
plt.title("Platykurtic Distribution")
plt.show()
print("Kurtosis (Platykurtic):", kurtosis(data_platy, fisher=False))

# Mesokurtic (normal shape)
data_meso = [1, 2, 3, 4, 5, 6, 7]
sns.histplot(data_meso, kde=True)
plt.title("Mesokurtic Distribution")
plt.show()
print("Kurtosis (Mesokurtic):", kurtosis(data_meso, fisher=False))
```



```
Kurtosis (Leptokurtic): 2.5
```



```
Kurtosis (Platykurtic): 1.78
```



```
Kurtosis (Mesokurtic): 1.75
```

Explanation:

Kurtosis tells about the heaviness of the tails (extremes) in a distribution:

- **Leptokurtic (kurtosis > 3):** sharp peak, heavy tails (more outliers)
- **Platykurtic (kurtosis < 3):** flat peak, light tails (fewer outliers)
- **Mesokurtic (kurtosis = 3):** normal bell-shaped curve

3 Measures of Variability (Spread)

These show how much the data varies from the center.

3.1 Variance

Variance is the average of the squared differences from the mean. It gives you a general idea of the spread.

Formula:
Variance = $\Sigma(x - \ )^2$ / n

Formula (Manual Calculation):
1. Find the mean: = (x + x + … + x) / n

2. Subtract the mean from each value and square it: (x -)^2
3. Add all squared values: Σ(x -)^2
4. Divide by number of items (n) for population variance

```
Example:
Data: [10, 20, 30]
Mean = (10+20+30)/3 = 20
Variance = [(10-20)² + (20-20)² + (30-20)²] / 3 = (100 + 0 + 100)/3 = 66.67
```

[6]:

```python
variance = np.var(data)
print("Variance:", variance)
```

```
Variance: 200.0
```

3.2 Standard Deviation

Standard deviation is the square root of variance. It tells you, on average, how far data points are from the mean.

Formula:
Standard Deviation = $\sqrt{\text{Variance}}$

From the above example: $\sqrt{66.67}$ 8.16

[7]:

```python
std_dev = np.std(data)
print("Standard Deviation:", std_dev)
```

```
Standard Deviation: 14.142135623730951
```

3.3 Coefficient of Variation

The coefficient of variation is useful for comparing the spread of data from different datasets regardless of their units.

[8]:

```python
cv = (std_dev / mean_value) * 100
print("Coefficient of Variation:", cv, "%")
```

```
Coefficient of Variation: 47.14045207910317 %
```

4 Covariance

Covariance measures the direction of the relationship between two variables:

- Positive value: both increase together
- Negative value: one increases, the other decreases
- Value close to zero: weak or no relationship

[9]:

```python
data_x = [1, 2, 3, 4, 5]
data_y = [2, 4, 6, 8, 10]
cov_matrix = np.cov(data_x, data_y)
print("Covariance Matrix:\n", cov_matrix)
```

```
Covariance Matrix:
 [[ 2.5  5. ]
 [ 5.  10. ]]
```

5 Summary with describe()

The `describe()` function provides summary statistics for each column:

- `count`: number of non-null values
- `mean`: average
- `std`: standard deviation
- `min`, 25%, 50%, 75%, `max`: percentiles (quartiles)

[11]:

```
import pandas as pd

df = pd.DataFrame({'Marks': data})
print(df.describe())
```

```
           Marks
count   5.000000
mean   30.000000
std    15.811388
min    10.000000
25%    20.000000
50%    30.000000
75%    40.000000
max    50.000000
```

6 Key Takeaway

Descriptive statistics give a snapshot of your dataset:

- Central tendency (mean, median, mode)
- Shape (skewness, kurtosis)
- Spread (variance, std dev, CV)
- Relationship (covariance)

These are essential before modeling or applying machine learning. Always explore and understand your data first!