Quantiles and Percentiles

1. Quantiles

Quantiles are statistical measures used to divide a set of numerical data into equal-sized groups, with each group containing an equal number of observations.

Quantiles are important measures of variability and can be used to: understand distribution of data, summarize and compare different datasets. They can also be used to identify outliers.

There are several types of quantiles used in statistical analysis, including:

```
a. Quartiles: Divide the data into four equal parts
    - Q1 (25th percentile)
    - Q2(50th percentile or median)
    - Q3 (75th percentile)


b. Deciles: Divide the data into ten equal parts.
    - D1 (10th percentile)
    - D2(20th percentile)
    - ... D9 (90th percentile).


c. Percentiles: Divide the data into 100 equal parts,
    - P1 (1st percentile)
    - P2(2nd percentile)
    - ..., P99 (99th percentile).


d. Quintiles: Divides the data into 5 equal parts
```

Things to remember while calculating these measures

1. Data should be sorted from low to high
2. You are basically finding the location of an observation
3. They are not actual values in the data
4. All other tiles can be easily derived from Percentiles

2. Percentile

A percentile is a statistical measure that represents the percentage of observations in a dataset that fall below a particular value.

`For example` : the 75th percentile is the value below which 75% of the observations in the dataset fall.

Formula to calculate the percentile value:

PL = $\frac{p}{100} \cdot (N + 1)$

where:

- PL = the desired percentile value location
- N = the total number of observations in the dataset
- p = the percentile rank (expressed as a percentage)

Given Data

[78, 82, 84, 88, 91, 93, 94, 96, 98, 99]

Step-by-Step Solution

1. **Sort the Data** (already sorted in this case):
   [78, 82, 84, 88, 91, 93, 94, 96, 98, 99]

2. **Calculate the Position of the 75th Percentile**:

   P = $\frac{p}{100} \cdot (N + 1)$ where N is the number of observations.

   P = $\frac{75}{100} \cdot (10 + 1)$

   P = $0.75 \cdot 11$ = 8.25

   The 75th percentile lies at the 8.25th position.

3. **Interpolate between the 8th and 9th Values**:

The 8th value in the sorted dataset is 96, and the 9th value is 98. We need to interpolate to find the exact 75th percentile.

The formula for interpolation is:

$$P = x_8 + 0.25 \cdot (x_9 - x_8)$$

Substitute the values:

$$P = 96 + 0.25 \cdot (98 - 96)$$

$$P = 96 + 0.25 \cdot 2$$

$$P = 96 + 0.5 = 96.5$$

Result

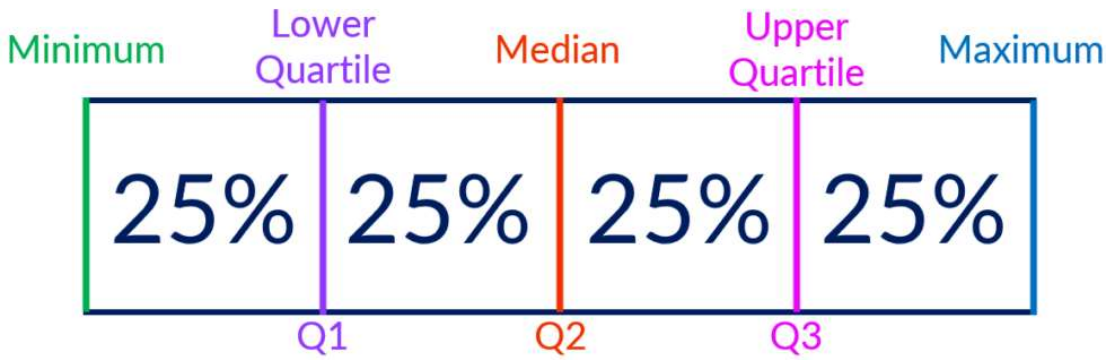
The **75th percentile** score is **96.5**.

5 Number Summary

The five-number summary is a descriptive statistic that provides a summary of a dataset. It consists of five values that divide the dataset into four equal parts, also known as quartiles.

The five-number summary includes the following values:

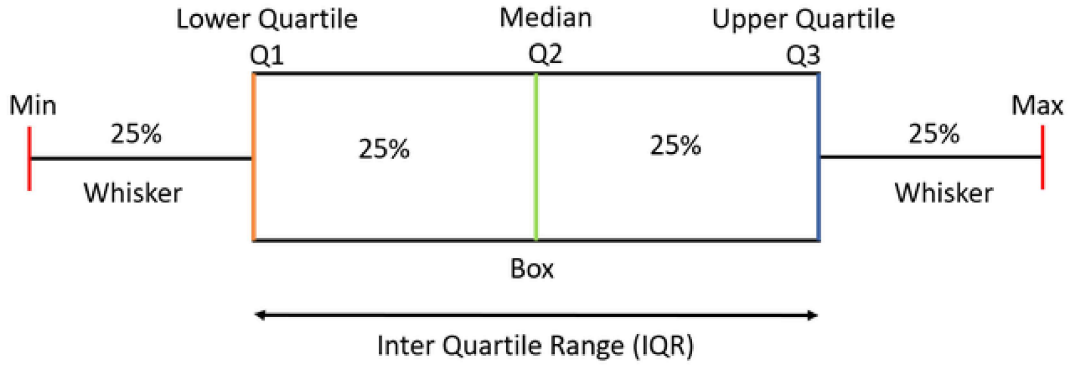
```
1. Minimum value: The smallest value in the dataset.
2. First quartile (Q1): The value that separates the lowest 25%
of the data from the rest of the dataset.
3. Median (Q2): The value that separates the lowest 50% from the
highest 50% of the data.
4. Third quartile (Q3): The value that separates the lowest 75%
of the data from the highest 25% of the data.
5. Maximum value: The largest value in the dataset.
```



The five-number summary is often represented visually using a box plot, which displays the range of the dataset, the median, and the quartiles.
The five-number summary is a useful way to quickly summarize the central tendency, variability, and distribution of a dataset.

Box Plot

- A box plot, also known as a box-and-whisker plot, is a graphical representation of a dataset that shows the distribution of the data.
- The box plot displays a summary of the data, including the minimum and maximum values, the first quartile (Q1), the median (Q2), and the third quartile (Q3).

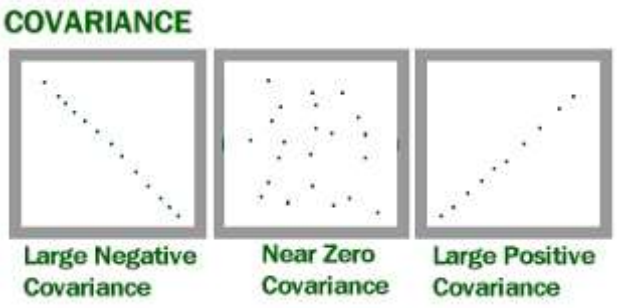


1. Box plot displayed the three quartiles Q1, Q2 and Q3 on a rectangular box, aligned either horizontally or vertically.
2. The lower whisker is the line from the first quartile to the smallest data point within (1.5 X IQR) from the first quartile.
   (min = Q1 - 1.5*IQR)
3. The upper whisker is the line from the third quartile to the larget data point within (1.5 X IQR) from the third quartile.
   (max = Q3 + 1.5*IQR)
4. A point beyond a whisker but less than 3*IQR from box edge is called outlier.
5. A point more than 3+IQR from the box edge is called extreme outlier.

Benifits of a Box Plot

- Easy way to see the distribution of data
- Tells about skewness of data
- Can identify outliers
- Compare 2 categories of data

Covariance



What problem does Covariance solve?

Covariance helps us understand how two variables change together. It shows whether an increase in one variable will likely result in an increase or decrease in the other variable. This helps in determining the relationship between two variables.

What is Covariance and how is it interpreted?

Covariance is a measure that tells us how two variables move together.

- If the covariance is positive, it means that when one variable goes up, the other also tends to go up.
- If the covariance is negative, it means that when one variable goes up, the other tends to go down.
- If the covariance is close to zero, it means there is no clear pattern in how the variables move together.

How is it calculated?

The formula for covariance between two variables (X) and (Y) in a sample is:

$$\text{Cov}(X, Y) = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

where:

- $X_i and Y_i$ are the individual data points.
- $\bar{X} and \bar{Y}$ are the means (averages) of X and Y.
- n is the number of data points.

For the covariance of a variable X with itself, which is actually the variance of X:

$$\text{Cov}(X, X) = \frac{\sum(X_i - \bar{X})^2}{N} = \text{Var}(X)$$

Disadvantages of using Covariance

Scale Dependent: Covariance is affected by the scale of the variables. Larger values will produce a larger covariance, making it difficult to compare across different datasets.

Not Easy to Interpret: The actual value of covariance does not provide a clear indication of the strength of the relationship between the variables.

No Standard Range: Covariance values can range from negative to positive infinity, making it hard to judge the degree of the relationship. Correlation is often preferred because it standardizes this relationship to a range between -1 and 1.

In [1]:
```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

import warnings
warnings.filterwarnings('ignore')
```

```
C:\Users\SVF\anaconda3\lib\site-packages\scipy\__init__.py:155: UserWarning: A NumPy version >=1.18.5 and <1.26.0 is required for this version of SciPy (detected version 1.26.4
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:
```
x = pd.Series([12,25,68,42,113])
y = pd.Series([11,29,58,121,100])
```

In [3]:
```
df = pd.DataFrame()

df['x'] = x
df['y'] = y

df.head(2)
```

Out[3]:

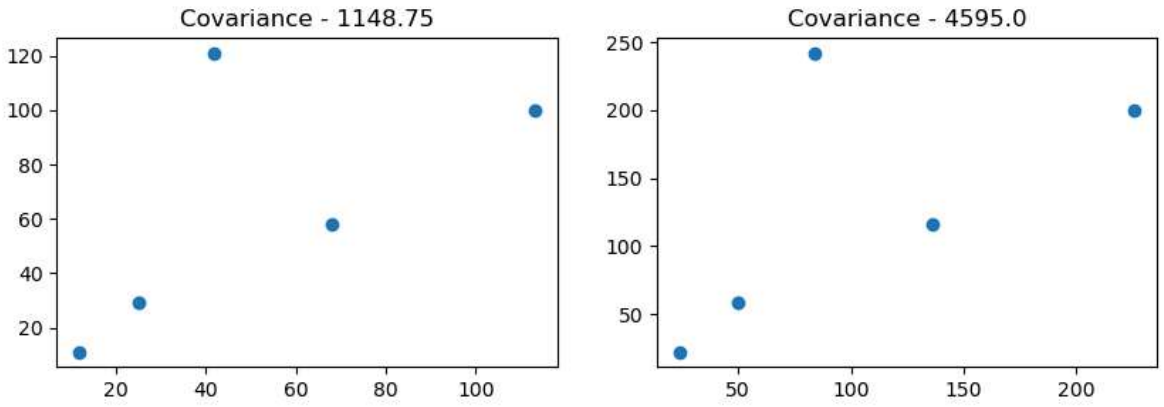
	x	y
0	12	11
1	25	29

In [4]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 3))

# Plot scatterplots on each axes
ax1.scatter(df['x'], df['y'])
ax2.scatter(df['x']*2, df['y']*2)

ax1.set_title("Covariance - " + str(np.cov(df['x'],df['y'])[0,1]))
ax2.set_title("Covariance - " + str(np.cov(df['x']*2,df['y']*2)[0,1]))
```

Out[4]:
```
Text(0.5, 1.0, 'Covariance - 4595.0')
```



In [5]:
```
# scale dependent
print(np.cov(df['x'],df['y'])[0,1])
print(np.cov(df['x']*2,df['y']*2)[0,1])
```

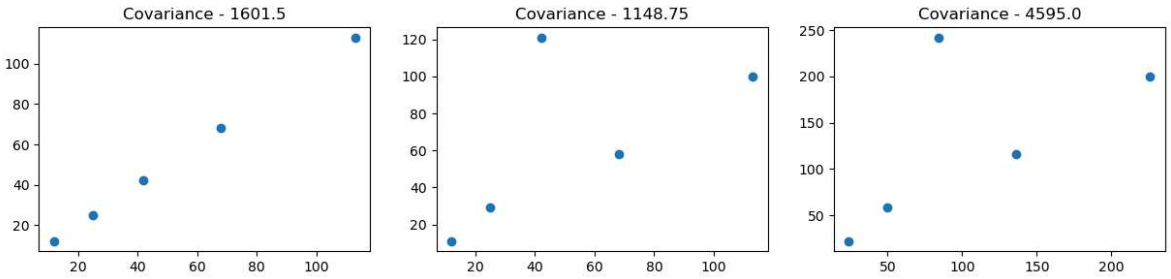
```
1148.75
4595.0
```

In [6]:
```
fig, ax = plt.subplots(1, 3, figsize=(15, 3))

# Plot scatterplots on each axes
ax[0].scatter(df['x'], df['x'])
ax[1].scatter(df['x'], df['y'])
ax[2].scatter(df['x']*2, df['y']*2)
```

```
ax[0].set_title("Covariance - " + str(np.cov(df['x'],df['x'])[0,1]))
ax[1].set_title("Covariance - " + str(np.cov(df['x'],df['y'])[0,1]))
ax[2].set_title("Covariance - " + str(np.cov(df['x']*2,df['y']*2)[0,1]))
```

```
Out[6]:  Text(0.5, 1.0, 'Covariance - 4595.0')
```

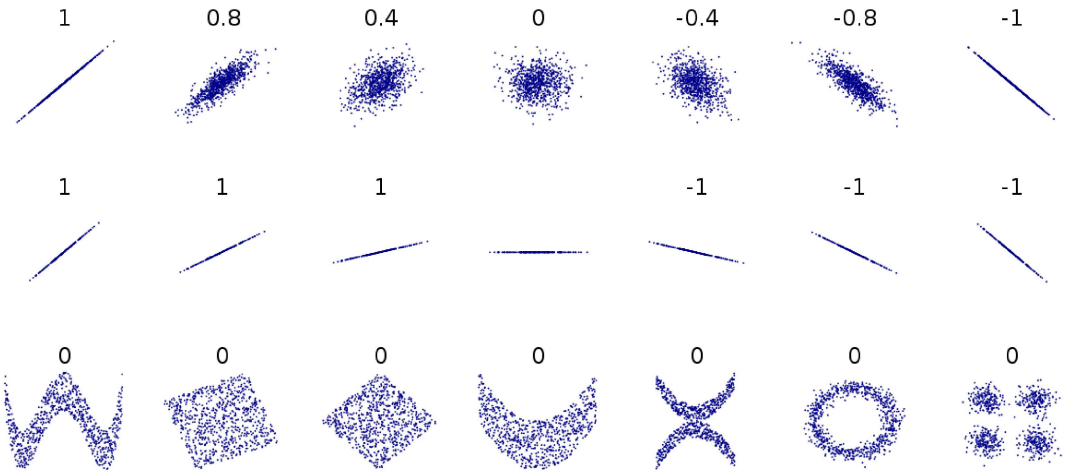


Correlation

Correlation helps us understand the strength and direction of the relationship between two variables. It tells us not only if the variables move together but also how strongly they are related, making it easier to compare different datasets.

What is Correlation?

Correlation is a statistical measure that describes how strongly two variables are related and whether they move in the same or opposite directions. It is standardized, meaning its values always range between -1 and 1.



- A correlation of 1 means the variables move together perfectly in the same direction.
- A correlation of -1 means the variables move together perfectly in opposite directions.
- A correlation of 0 means there is no linear relationship between the variables.

How is it calculated?

The formula for the correlation coefficient (r) between two variables X and Y is:

$$r = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y}$$

where:

- $\mathrm{Cov}(X, Y)$ is the covariance between X and Y.
- σ_X is the standard deviation of X.
- σ_Y is the standard deviation of Y.

Advantages

1. **Standardized Measure**: Correlation values are always between -1 and 1, making it easy to understand the strength and direction of the relationship.
2. **Simple Interpretation**: The correlation coefficient directly shows the strength and direction of the linear relationship.

Disadvantages

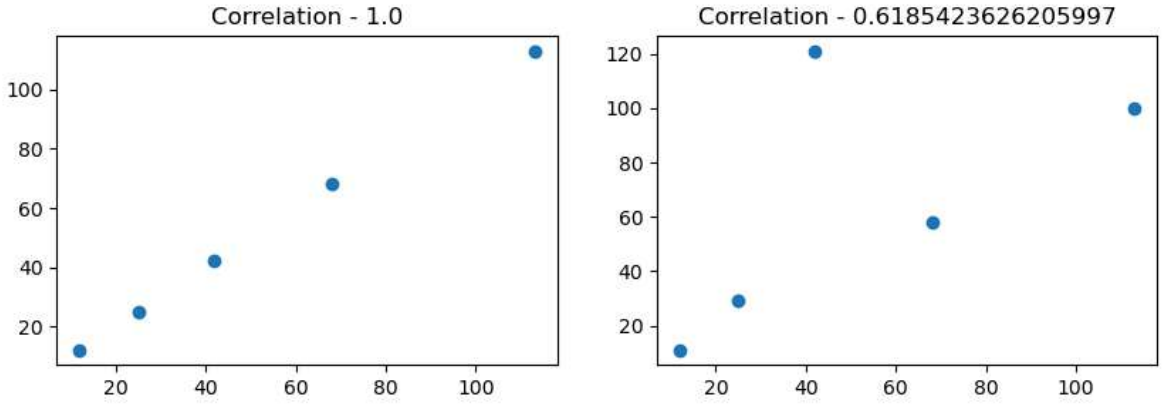
1. **Linear Relationships Only**: Correlation only measures linear relationships. If the relationship between variables is non-linear, correlation may not provide useful information.
2. **Sensitive to Outliers**: Correlation can be significantly affected by outliers, which can distort the measure of the relationship.
3. **Does Not Imply Causation**: A high correlation between two variables does not mean that one variable causes the other to change. It only indicates that they are related.

In [7]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 3))

# Plot scatterplots on each axes
ax1.scatter(df['x'], df['x'])
ax2.scatter(df['x'], df['y'])

ax1.set_title("Correlation - " + str(df['x'].corr(df['x'])))
ax2.set_title("Correlation - " + str((df['x']).corr(df['y'])))
```

Out[7]:
```
Text(0.5, 1.0, 'Correlation - 0.6185423626205997')
```



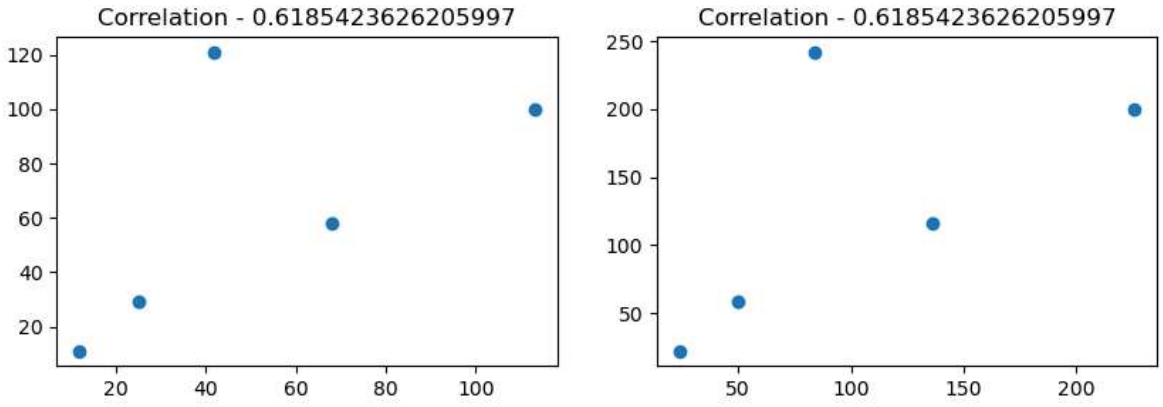
In [8]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 3))

# Plot scatterplots on each axes
```

```
ax1.scatter(df['x'], df['y'])
ax2.scatter(df['x']*2, df['y']*2)

ax1.set_title("Correlation - " + str(df['x'].corr(df['y'])))
ax2.set_title("Correlation - " + str((df['x']*2).corr(df['y']*2)))
```

Out[8]: Text(0.5, 1.0, 'Correlation - 0.6185423626205997')



Correlation Does Not Imply Causation

When we say that "correlation does not imply causation," we mean that even if two variables are correlated (i.e., they tend to move together), it doesn't mean that one variable causes the other to change. Correlation simply indicates a relationship between the two variables, but it doesn't explain why that relationship exists.

Example 1: Number of Firefighters and Damage from Fires

`Observation` : There is a high correlation between the number of firefighters at a fire and the amount of damage caused by the fire.
`Explanation` : This doesn't mean that more firefighters cause more damage. Instead, larger fires cause more damage and require more firefighters to respond. The size of the fire is the underlying cause.

Example 2: Sunglasses Sales and Ice Cream Sales

`Observation` : There is a correlation between sunglasses sales and ice cream sales.
`Explanation` : This doesn't mean that buying sunglasses causes people to buy ice cream. Instead, both are more frequently bought during sunny and warm weather. The weather is the underlying cause.