

To estimate the mean, variance, standard deviation, and coefficient of variation (CV) for the given grouped frequency distribution, we follow these steps:

1. Calculate the Midpoint (x_i) of each class:

$$x_i = \frac{\text{Lower Bound} + \text{Upper Bound}}{2}$$

2. Calculate the Mean (\bar{x}) using the formula:

$$\bar{x} = \frac{\sum f_i x_i}{\sum f_i}$$

3. Calculate Variance (σ^2) using:

$$\sigma^2 = \frac{\sum f_i (x_i - \bar{x})^2}{\sum f_i}$$

4. Calculate Standard Deviation (σ) as:

$$\sigma = \sqrt{\sigma^2}$$

5. Calculate Coefficient of Variation (CV) as:

$$CV = \left(\frac{\sigma}{\bar{x}} \right) \times 100\%$$

I will now compute these values.



I will now compute these values.

Here are the estimated values for the given grouped frequency distribution:

- Mean (\bar{x}) = 2.24
- Variance (σ^2) = 0.424
- Standard Deviation (σ) = 0.651
- Coefficient of Variation (CV) = 29.10%

Here is the frequency distribution table with midpoints and the necessary calculations:

Class Interval	Frequency (f)	Midpoint (x)	$f \times x$	$(x - \bar{x})^2$	$f \times (x - \bar{x})^2$
1.2 - 1.6	220	1.4	220×1.4	$(1.4 - 2.24)^2$	$220 \times (1.4 - 2.24)^2$
1.6 - 2.0	150	1.8	150×1.8	$(1.8 - 2.24)^2$	$150 \times (1.8 - 2.24)^2$
2.0 - 2.4	90	2.2	90×2.2	$(2.2 - 2.24)^2$	$90 \times (2.2 - 2.24)^2$
2.4 - 2.8	110	2.6	110×2.6	$(2.6 - 2.24)^2$	$110 \times (2.6 - 2.24)^2$
2.8 - 3.2	280	3.0	280×3.0	$(3.0 - 2.24)^2$	$280 \times (3.0 - 2.24)^2$

Where:

- Midpoint (x) = $\frac{\text{Lower Bound} + \text{Upper Bound}}{2}$
- $f \times x$ = Frequency \times Midpoint
- Variance Calculation ($f \times (x - \bar{x})^2$)



```
import numpy as np
```

```
# Given class intervals and frequencies
classes = [(1.2, 1.6), (1.6, 2.0), (2.0, 2.4), (2.4, 2.8), (2.8, 3.2)]
frequencies = [220, 150, 90, 110, 280]
```

```
# Calculate midpoints
midpoints = [(c[0] + c[1]) / 2 for c in classes]
```

```
# Calculate Mean
total_freq = sum(frequencies)
mean = sum(f * x for f, x in zip(frequencies, midpoints)) / total_freq
```

```
# Calculate Variance
variance = sum(f * (x - mean) ** 2 for f, x in zip(frequencies, midpoints)) / total_freq
```

```
# Standard Deviation
std_dev = np.sqrt(variance)
```

```
# Coefficient of Variation (CV)
cv = (std_dev / mean) * 100
```

```
mean, variance, std_dev, cv
```

Step 1: Calculate the Correlation Coefficient (Pearson's r)

The correlation coefficient is given by:

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{(n \sum x^2 - (\sum x)^2)(n \sum y^2 - (\sum y)^2)}}$$

where:

- x and y are the data points
- n is the number of data points
- $\sum xy$ is the sum of the product of x and y values
- $\sum x$ and $\sum y$ are the sum of x and y values
- $\sum x^2$ and $\sum y^2$ are the sum of squared x and y values

You can also use Python's `numpy` function `np.corrcoef(x, y)[0,1]` to compute r .

Step 2: Compute the Linear Regression Model ($y = mx + b$)

A linear regression model follows the equation:

$$y = mx + b$$

where:

- m (slope) is given by:

$$m = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2}$$

- b (intercept) is given by:

$$b = \frac{\sum y - m \sum x}{n}$$

Alternatively, you can use Python's `scipy.stats.linregress(x, y)` function to compute the slope and intercept.

Here is the Python code and results for calculating the correlation coefficient and linear regression model:

Python Code:

```
import numpy as np
import scipy.stats as stats

def calculate_correlation_and_regression(x, y):
    # Compute correlation coefficient
    r = np.corrcoef(x, y)[0, 1]

    # Compute linear regression (slope and intercept)
    slope, intercept, _, _, _ = stats.linregress(x, y)

    # Regression equation
    regression_eq = f"y = {slope:.2f}x + {intercept:.2f}"

    return r, regression_eq

# Given datasets
dataset_1_x = np.array([1, 2, 3, 4, 5, 6, 7, 8])
dataset_1_y = np.array([30, 45, 51, 57, 60, 65, 70, 71])

dataset_2_x = np.array([100, 120, 140, 160, 170])
dataset_2_y = np.array([75, 90, 115, 140, 155])

# Calculate for both datasets
result_1 = calculate_correlation_and_regression(dataset_1_x, dataset_1_y)
result_2 = calculate_correlation_and_regression(dataset_2_x, dataset_2_y)

print(result_1)
print(result_2)
```

Results:

Dataset i:

- Correlation Coefficient: **0.9629** (Strong positive correlation)
- Linear Regression Model: **y = 5.44x + 31.64**

Dataset ii:

- Correlation Coefficient: **0.9946** (Very strong positive correlation)
- Linear Regression Model: **y = 1.16x - 44.88**

```
# Re-import necessary libraries since execution state was reset
```

```
import numpy as np
import scipy.stats as stats
```

```
def calculate_correlation_and_regression(x, y):
    # Compute correlation coefficient
    r = np.corrcoef(x, y)[0, 1]

    # Compute linear regression (slope and intercept)
    slope, intercept, _, _, _ = stats.linregress(x, y)

    # Regression equation
    regression_eq = f'y = {slope:.2f}x + {intercept:.2f}'

    return r, regression_eq

# Given datasets
dataset_1_x = np.array([1, 2, 3, 4, 5, 6, 7, 8])
dataset_1_y = np.array([30, 45, 51, 57, 60, 65, 70, 71])

dataset_2_x = np.array([100, 120, 140, 160, 170])
dataset_2_y = np.array([75, 90, 115, 140, 155])

# Calculate for both datasets
result_1 = calculate_correlation_and_regression(dataset_1_x, dataset_1_y)
result_2 = calculate_correlation_and_regression(dataset_2_x, dataset_2_y)

result_1, result_2
```

It looks like there is a calculation error in your formula for the slope (m) in Excel. Let's break it down and correct it.

Correct Formulas for Linear Regression

For a dataset with n points (x_i, y_i) , the formulas are:

Slope (m) Calculation:

$$m = \frac{n \sum(x_i y_i) - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

Intercept (c) Calculation:

$$c = \frac{\sum y_i - m \sum x_i}{n}$$

Correlation Coefficient (r) Calculation:

$$r = \frac{n \sum(x_i y_i) - \sum x_i \sum y_i}{\sqrt{(n \sum x_i^2 - (\sum x_i)^2)(n \sum y_i^2 - (\sum y_i)^2)}}$$

Fixing Your Excel Formulas

Fixing mmm Calculation:

Your formula:

$$[5*83150 - 680*575] / [5*98500 - (690^2)]$$

Correction:

Change "680" to "690" (since the sum of $x_{ix} - x_{ix}$ is 690, not 680)

$$= (5 * 83150 - 690 * 575) / (5 * 98500 - 690^2)$$

Fixing ccc Calculation:

Your formula should be:

$$= (575 - (m * 690)) / 5$$

Replace "m" with the correct cell reference.

Fixing Correlation Coefficient (rrr) Calculation:

Your formula:

$$[n * \sum xi * yi - \sum xi * \sum yi] / \sqrt{[(n * \sum xi * xi - (\sum xi)^2) * (n * \sum yi * yi - (\sum yi)^2)]}$$

Correction:

$$= (5 * 83150 - 690 * 575) / \sqrt{(5 * 98500 - 690^2) * (5 * 70575 - 575^2)}$$

Final Corrected Excel Formulas:

- **Slope (m):**

$$=(5 * 83150 - 690 * 575) / (5 * 98500 - 690^2)$$

- **Intercept (c):**

$$=(575 - (m * 690)) / 5$$

(Replace **m** with the correct cell reference)

- **Correlation Coefficient (r):**

$$=(5 * 83150 - 690 * 575) / \sqrt{(5 * 98500 - 690^2) * (5 * 70575 - 575^2)}$$

Correct Formulas for Linear Regression

For n data points (x_i, y_i) , the formulas are:

Slope (m) Calculation:

$$m = \frac{n \sum (x_i y_i) - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

Intercept (c) Calculation:

$$c = \frac{\sum y_i - m \sum x_i}{n}$$

Correlation Coefficient (r) Calculation:

$$r = \frac{n \sum (x_i y_i) - \sum x_i \sum y_i}{\sqrt{(n \sum x_i^2 - (\sum x_i)^2)(n \sum y_i^2 - (\sum y_i)^2)}}$$

Fixing Your Excel Formulas

1 Fixing mmm Calculation:

Your current formula:

$$[n * \sum xi * yi - \sum xi * \sum yi] / [n * \sum xi * xi - (\sum xi)^2]$$

Correct Excel Formula:

$$= (8 * 11193 - 318 * 279) / (8 * 12764 - 318^2)$$

2 Fixing ccc Calculation:

Your current formula:

$$(\sum yi - m * \sum xi) / n$$

Correct Excel Formula:

$$= (279 - (m * 318)) / 8$$

(Replace **m** with the correct cell reference.)

3 Fixing Correlation Coefficient (rrr) Calculation:

Your current formula:

$$[n * \sum xi * yi - \sum xi * \sum yi] / \sqrt{[(n * \sum xi * xi - (\sum xi)^2) * (n * \sum yi * yi - (\sum yi)^2)]}$$

Correct Excel Formula:

$$= (8 * 11193 - 318 * 279) / \text{SQRT}((8 * 12764 - 318^2) * (8 * 9857 - 279^2))$$

Final Corrected Excel Formulas:

- Slope (mmm)**

$$= (8 * 11193 - 318 * 279) / (8 * 12764 - 318^2)$$

- Intercept (ccc)**

= (279 - (m * 318)) / 8

- **Correlation Coefficient (rrr)**

= (8 * 11193 - 318 * 279) / SQRT((8 * 12764 - 318^2) * (8 * 9857 - 279^2))