Advanced Statistics

Dr. Syed Faisal Bukhari

Associate Professor

Department of Data Science

Faculty of Computing and Information Technology

University of the Punjab

Textbooks

- **Probability & Statistics for Engineers & Scientists**, Ninth Edition, Ronald E. Walpole, Raymond H. Myer

- **Elementary Statistics: Picturing the World,** 6th Edition, Ron Larson and Betsy Farber

- **Elementary Statistics**, 13th Edition, Mario F. Triola

Reference books

- **Probability and Statistical Inference, Ninth Edition,** Robert V. Hogg, Elliot A. Tanis, Dale L. Zimmerman

- **Probability Demystified**, Allan G. Bluman

- **Practical Statistics for Data Scientists: 50 Essential Concepts,** Peter Bruce and Andrew Bruce

- **Schaum's Outline of Probability,** Second Edition, Seymour Lipschutz, Marc Lipson

- **Python for Probability, Statistics, and Machine Learning,** José Unpingco

References

- Probability & Statistics for Engineers & Scientists, Ninth edition, Ronald E. Walpole, Raymond H. Myer
- Elementary Statistics, Tenth Edition, Mario F. Triola

These notes contain material from the above resources.

Correlation

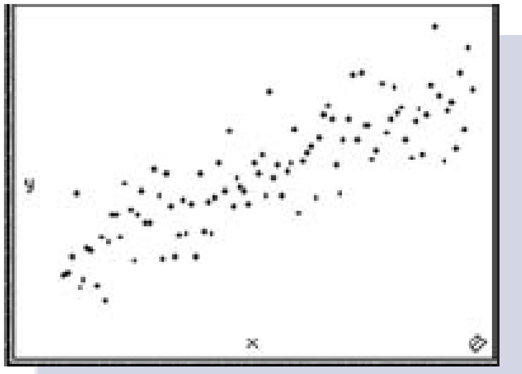
A **correlation** exists between **two variables** when **one** of them is **related** to the other in some way.

Exploring the Data

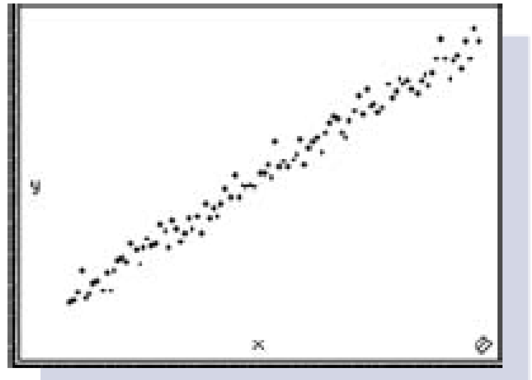
We can often see a **relationship between two variables** by constructing a **scatterplot**. When we examine a **scatterplot**, we should study the **overall pattern** of the plotted points. If there is a pattern, we should note its **direction**.

- ❑ An **uphill direction** suggests that as **one variable increases**, the **other also increases**.
- ❑ A **downhill direction** suggests that as **one variable increases**, the **other decreases**.
- ❑ We should look for **outliers**, which **are points that lie** very **far away** from all of the **other points**.

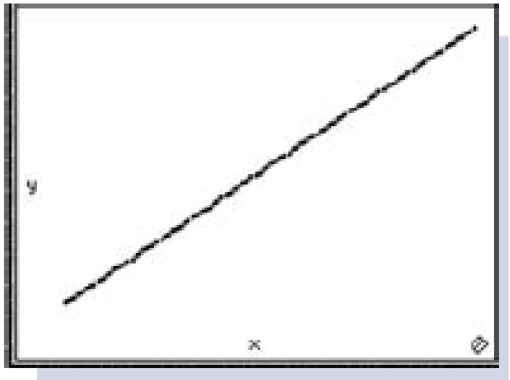
Scatter plots



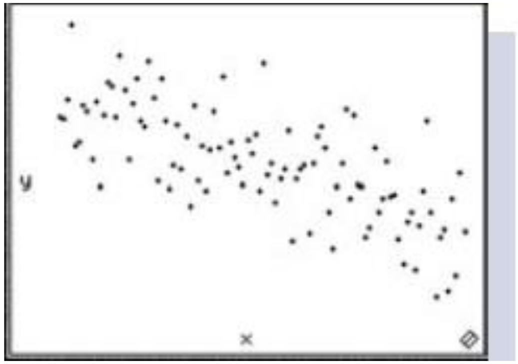
Positive correlation:
***r* = 0.851**



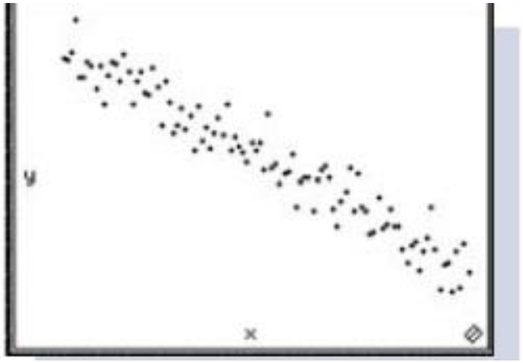
Positive correlation:
***r* = 0.991**



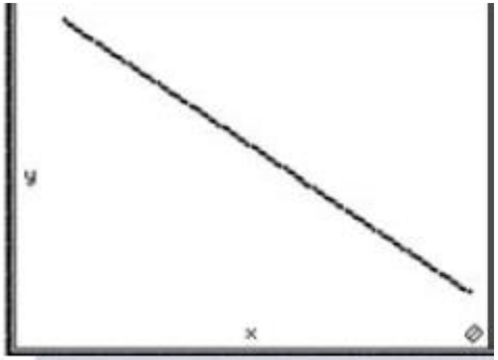
Perfect positive correlation:
***r* = 1**



Negative correlation:
***r* = – 0.702**

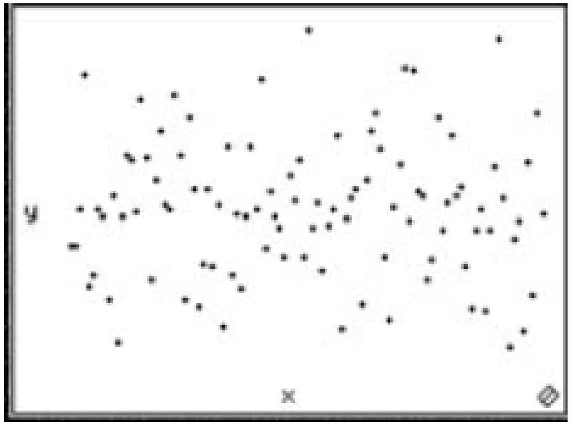


Negative correlation:
***r* = – 0.965**

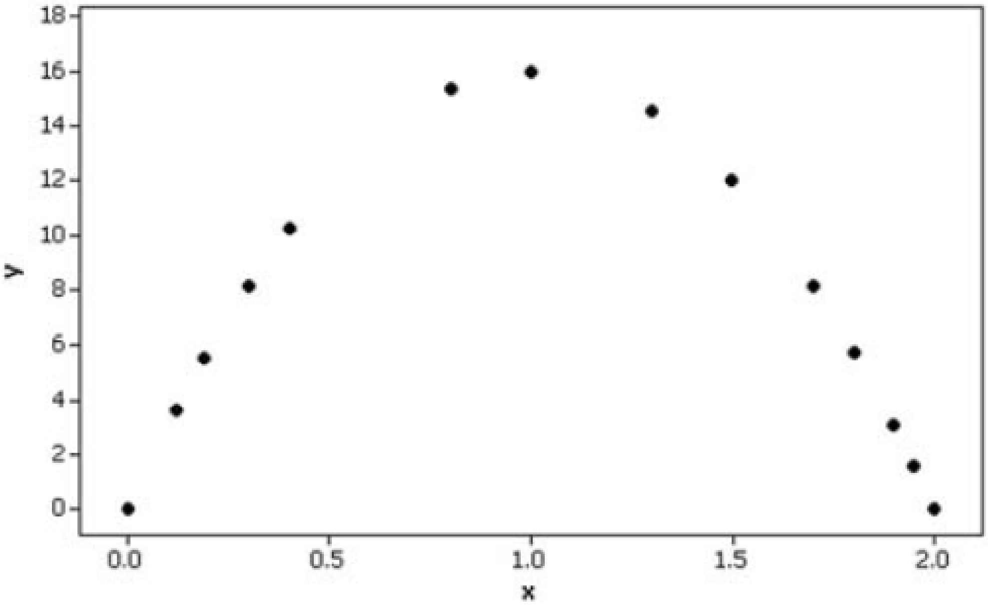


Perfect negative correlation:
***r* = –1**

Scatter plots



No correlation: *r* = 0



Nonlinear relationship: *r* = – 0.087

Palm reading

Palm reading

- Some **people believe** that the **length of their palm's lifeline** can be used to **predict longevity**.

- In a letter published in the **Journal of the American Medical Association**, authors M. E. Wilson and L. E. Mather **refuted that belief** with a **study of cadavers**.

- **Ages at death** were recorded, along with the **lengths** of **palm lifelines**. The authors concluded that **there is no significant correlation between age at death** and **length of lifeline**. Palmistry lost, hands down.

Requirements

Given any **collection of sample paired data**, the linear **correlation coefficient *r*** can always be computed, but the following requirements should be satisfied when **testing hypotheses** or making other **inferences about *r***.

1. The sample of **paired (*x*, *y*)** data is a ***random* sample** of **independent quantitative data**.
2. **Visual examination** of the **scatterplot** must confirm that the points approximate a **straight-line pattern**.
3. **Any outliers** must be removed if they are known to **be errors**. The effects of any other outliers should be considered by **calculating *r* with** and **without the outliers** included.

- ***Note*: Requirements 2 and 3** above are simplified attempts at checking this formal requirement:
- The **pairs of (*x*, *y*)** data must have **a bivariate normal distribution.** (This assumption basically requires that for any **fixed value of *x***, the **corresponding values of *y*** have a distribution that is **bell-shaped**, and for any fixed value of *y*, the values of *x* have a **distribution that is bell-shaped**.)
- This requirement is usually difficult to check, so for now, we will use Requirements 2 and 3 as listed above.

Notation for the Linear Correlation Coefficient

- **n**: represents the **number of pairs** of data present.

- **r**: represents the **linear correlation coefficient** for a **sample**.

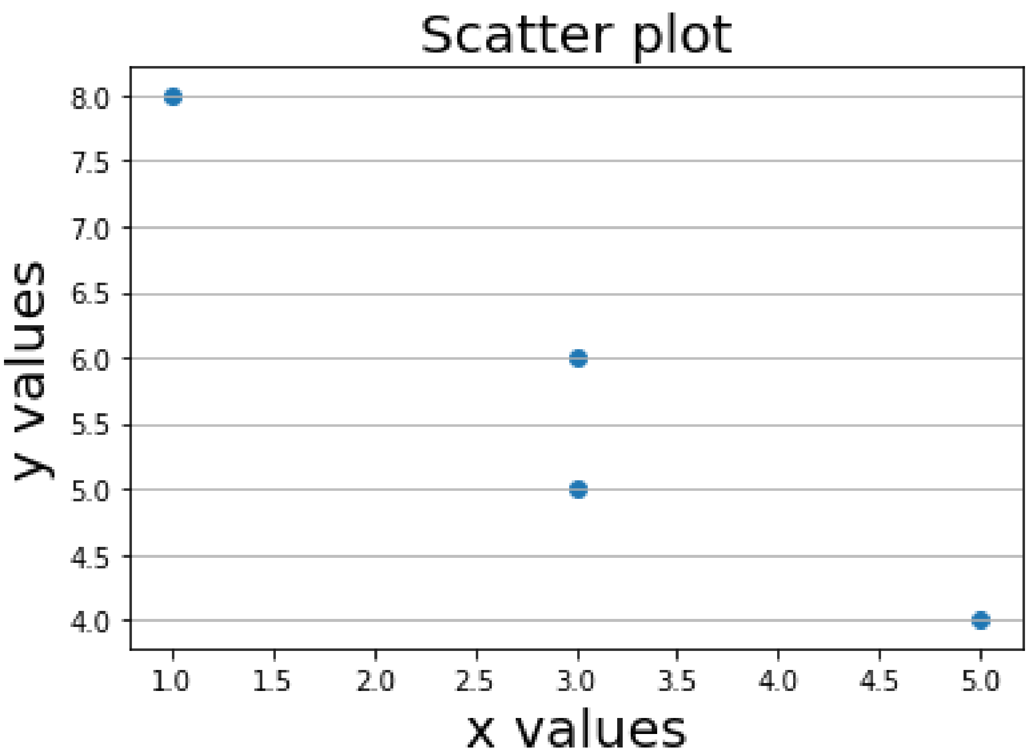
- **ρ**: Greek letter **rho** used to represent the **linear correlation coefficient** for a **population**.

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

Example Calculating *r* Using the simple random sample of data given in the table, find the value of the **linear correlation coefficient *r***.

x	3	1	3	5
y	5	8	6	4

REQUIREMENT The data are a **simple random sample**. The accompanying **Python-generated scatterplot** shows **a pattern of points** that does appear to be a **straight-line pattern.** There are no outliers. We can proceed with the calculation of the linear correlation coefficient ***r*****.**



x	y	xy	x^2	y^2
3	5	15	9	25
1	8	8	1	64
3	6	18	9	36
5	4	20	25	16
$\sum x = 12$	$\sum y = 23$	$\sum xy = 61$	$\sum x^2 = 44$	$\sum y^2 = 141$

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

$$r = \frac{4(61) - (12)(23)}{\sqrt{4(44) - (12)^2}\sqrt{4(141) - (23)^2}}$$

$$r = \frac{-32}{\sqrt{32}\sqrt{35}} = -0.956$$

- These calculations get quite messy with larger data sets, so it's **fortunate** that the **linear correlation coefficient** can be **found automatically** with many different **calculators** and **computer programs**

Interpreting the Linear Correlation Coefficient

- We need to interpret a calculated **value of *r***, such as the value of **–0.956** found in the preceding example.

- The value of *r* must always fall between **-1 and +1** inclusive.

- If ***r* is close to 0**, we conclude that **there is no linear correlation** between *x* and *y*, but if *r* is close **-1 to or +1** we conclude that there is a **linear correlation between *x* and *y***.

Properties of the Linear Correlation Coefficient *r*

1. The value of r is always between **-1** and **+1** inclusive. That is, $-1 \le r \le +1$

2. The value of r does not change if all values of either variable are **converted** to a **different scale**.

3. The value of **r** is **not affected** by the choice of **x** or **y**. Interchange all x- and y-values and the value of r will not change.

4. r measures the strength of a **linear relationship**. It is **not designed** to measure the **strength of a relationship** that is **not linear**.

Hypothesis Test for Correlation

Assume: **r = 0.926**, **n = 8**

1. **We state our hypothesis as**:

H_0: $\rho = 0$ (There is no linear correlation.)

H_1: $\rho \neq 0$ (There is a linear correlation.)

2. **The level of significance is set** $\alpha = 0.05$.

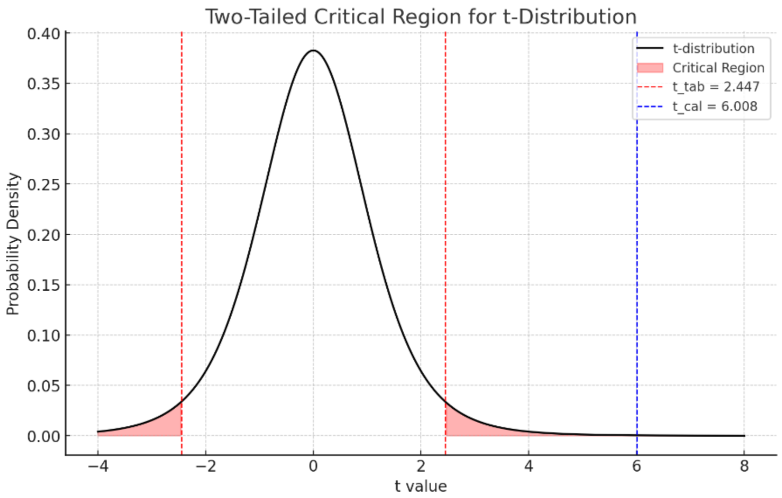
3. **Test statistic to be used is** $t_{cal} = \dfrac{r}{\sqrt{\dfrac{1 - r^2}{n-2}}}$

4. **Calculations:**

$$t_{cal} = \frac{0.926}{\sqrt{\dfrac{1 - (0.926)^2}{8-2}}} = 6.008$$

5. Critical region:

$|t_{cal}| > t_{tab}$, where $t_{tab} = t_{(\alpha/2,\, n-2)}$



$t_{tab} = t_{(0.0250,\, 6)} = 2.447$

$6.008 > 2.447$ (True)

6. Conclusion: Since t_{cal} is greater than the t_{tab}, so we reject H_o.

- There is **sufficient evidence** to support the claim of a linear correlation.

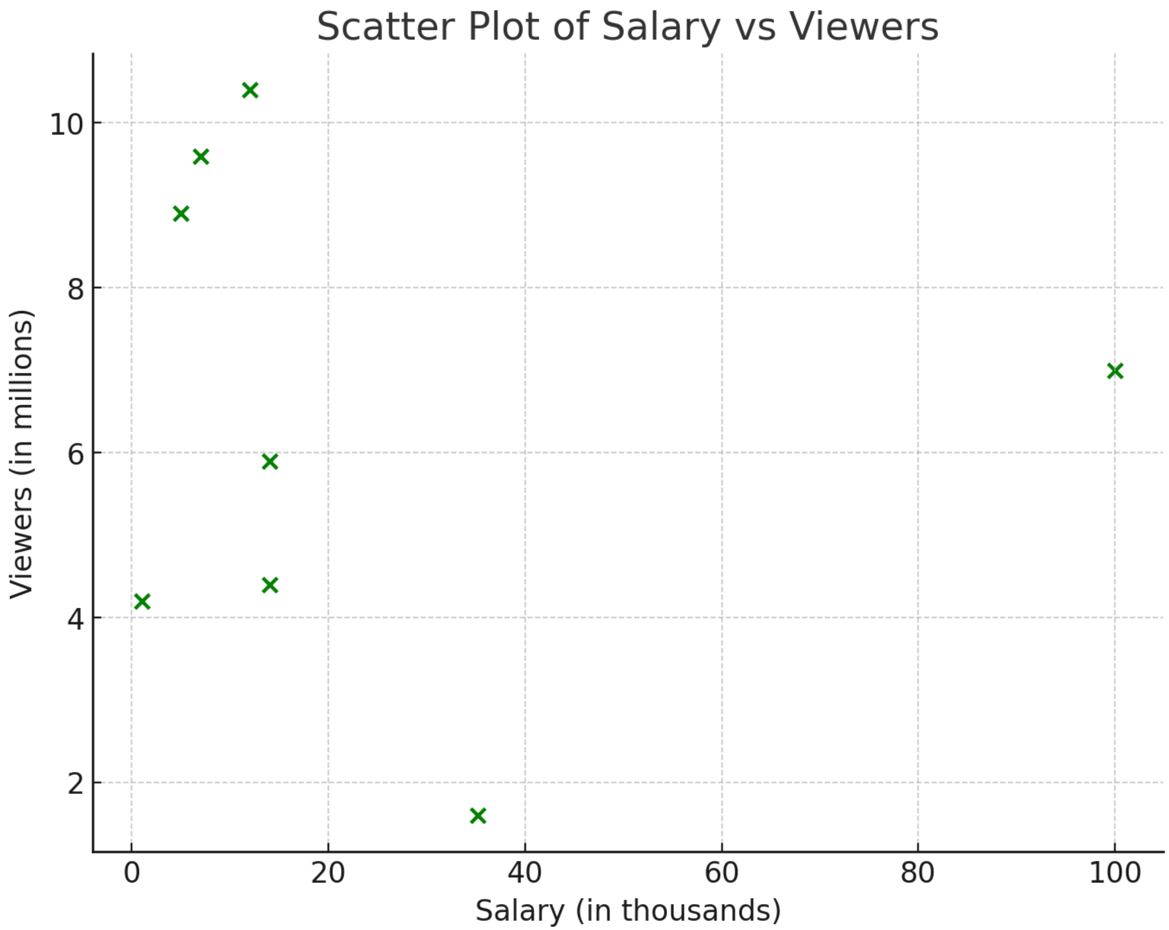
```python
from scipy.stats import t
import numpy as np
# Given values
r = 0.926
n = 8
# Hypotheses:
# H0: ρ = 0 (No linear correlation)
# H1: ρ ≠ 0 (Linear correlation exists)
# Calculate the t-statistic for testing

t_stat = r * np.sqrt((n - 2) / (1 - r**2))

# Degrees of freedom
df = n - 2
```

```python
# Calculate the p-value (two-tailed)
p_value = 2 * (1 - t.cdf(abs(t_stat), df))

# Output the results
print("t-statistic:", t_stat)
print("Degrees of freedom:", df)
print("p-value:", p_value)

# Conclusion
alpha = 0.05  # Significance level
if p_value < alpha:
    print("Reject H0: There is a significant linear correlation.")
else:
    print("Fail to reject H0: There is no significant linear correlation.")
```

Examples: Applications of correlation

Buying a TV Audience The *New York Post* published the **annual salaries (in millions)** and the **number of viewers (in millions),** with results given below for **Oprah Winfrey**, **David Letterman, Jay Leno, Kelsey Grammer, Barbara Walters, Dan Rather, James Gandolfini, and Susan Lucci**, repsectively. Is there **a correlation between salary and number of viewers**? Implement it in Python.

Salary	100	14	14	35.2	12	7	5	1
Viewers	7	4.4	5.9	1.6	10.4	9.6	8.9	4.2


Scatter Plot of Salary vs Viewers
Viewers (in millions)
10
8
6
4
2
0
20
40
60
80
100
Salary (in thousands)

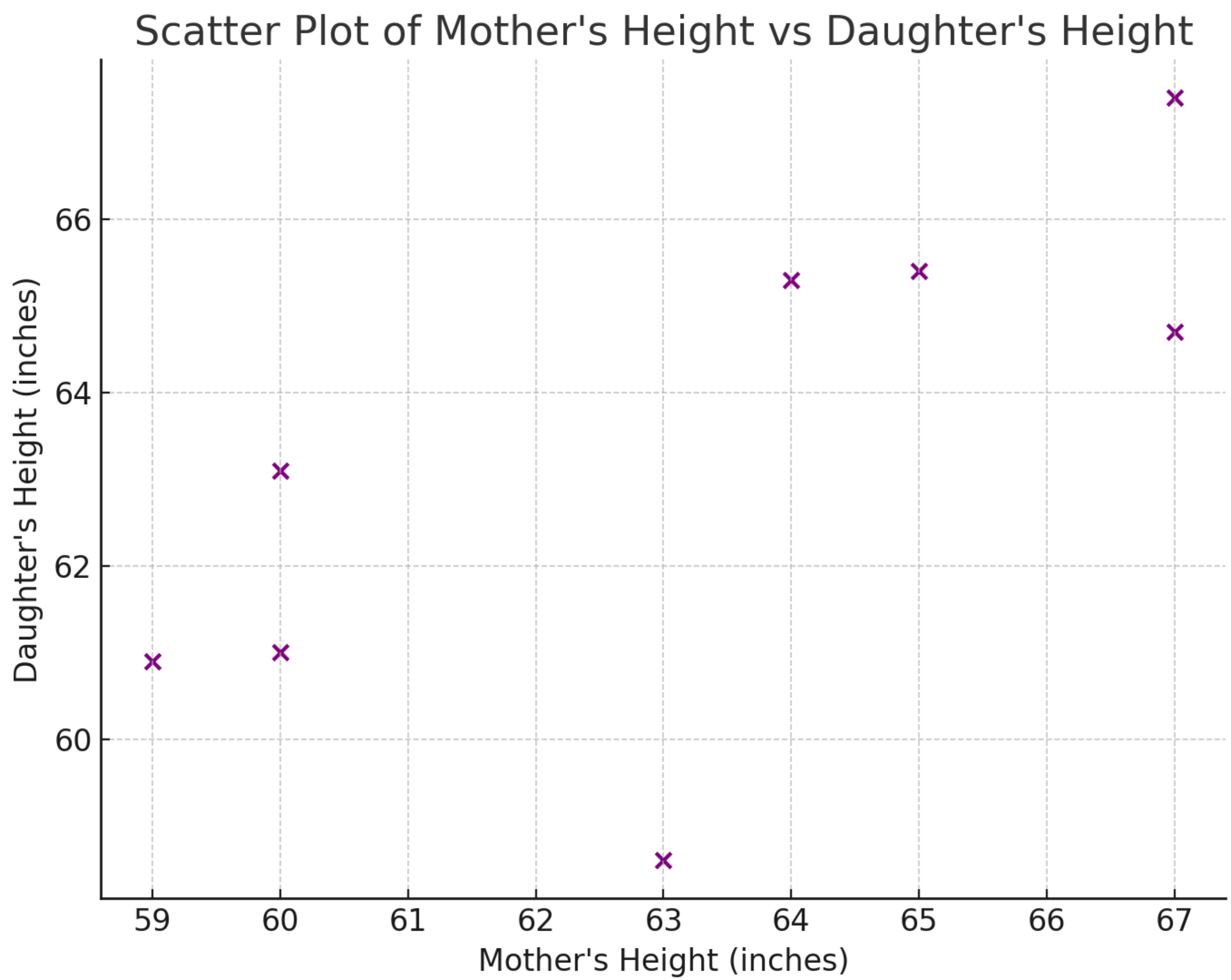
```python
# Import Python package
import numpy as np

# Statistical functions (scipy.stats)
from scipy.stats import pearsonr

# Define the data
salary = np.array([100, 14, 14, 35.2, 12, 7, 5, 1])
viewers = np.array([7, 4.4, 5.9, 1.6, 10.4, 9.6,
8.9, 4.2])

# Calculate the correlation coefficient and p-value
correlation_coefficient, p_value = pearsonr(salary,
viewers)

# Print correlation coefficient
print("Correlation Coefficient (r):",
correlation_coefficient)

# Print p-value if required
# print("P-value:", p_value)
```

The **correlation coefficient $r = -0.118$** indicates a very weak negative correlation between salary and viewers, suggesting no meaningful linear relationship between the two variables

Examples: Applications of correlation

Parent Child Heights Listed below are **heights (in inches)** of **mothers** and **heights (in inches)** of their daughters (based on data from the National Health Examination Survey). Does there appear to be a **linear correlation between mother's heights** and the **heights of their daughters**? Use Python.

Mother's height	63	67	64	60	65	67	59	60
Daughter's height	58.6	64.7	65.3	61.0	65.4	67.4	60.9	63.1



Scatter Plot of Mother's Height vs Daughter's Height
Daughter's Height (inches)
Mother's Height (inches)
66
64
62
60
59
60
61
62
63
64
65
66
67

```python
# Import Python package
import numpy as np

# Define the data
mother_height = np.array([63, 67, 64, 60, 65, 67, 
59, 60])
daughter_height = np.array([58.6, 64.7, 65.3, 61.0, 
65.4, 67.4, 60.9, 63.1])

# Calculate the correlation coefficient using NumPy
correlation_coefficient = np.corrcoef(mother_height, 
daughter_height)[0, 1]

# Print correlation coefficient
print("Correlation Coefficient (r):", 
correlation_coefficient)
```

The correlation coefficient **r = 0.693** indicates a moderate positive correlation between mothers' and daughters' heights, suggesting that as one increases, the other tends to increase as well