Smt. Indira Gandhi College of Engineering
Computer Engineering Department
Ghansoli – Navi Mumbai
Academic Year 2023-24 (Even Sem)

Student Name: Khyati Garude **Roll No.:** 13 **Class:** BE **Sem:** VIII

Course Name: Applied Data Science Lab

Course Code: CSL8023

Experiment No. 03

Experiment Title: Descriptive Statistics Multivariate : Computation Of Karl Pearson Coefficient Of Correlation On Multivariate Data

Date of Performance	Date of Submission	Marks (10)					Sign / Remark
		A	B	C	D	E	
		2	3	2	2	1	
25/1/24	1/2/24	2	3	2	2	0	20/3/24
		Total Marks					
		09					

A: Prerequisite Knowledge
B: Implementation
C: Oral
D: Content
E: Punctuality & Discipline

Khyati Garuda 13
BE COMPS
ADS

Date:_____

DATE	EXPERIMENT-3	SIGN
1/2/24	Descriptive statistics Multivariate : Computation of Karl Pearson coefficient of correlation on Multivariate Data	20/3/24

AIM: Descriptive Statistics Multivariate - Computation of Karl Pearson coefficient of correlation on Multivariate data.

THEORY:

Correlation is a statistical measure that expresses the extent to which two variables are linearly related (meaning they change together at a constant rate).

It is a common tool for describing simple relationships without making a statement about cause and effect.

The sample correlation coefficient , r, quantifies the strength of the relationship.

Correlations are also tested for statistical significance.

Date:_____

→ The Karl Pearson's correlation coefficient assesses linear relationships between pairs of variables in multivariate data.

→ It results in a correlation matrix, where each cell contains the correlation coefficient between two variables, offering a comprehensive view of relationships among all variables.

→ Values range from -1 to 1, indicating perfect negative to perfect positive linear relationships, with 0 indicating no linear relationship.

→ Karl Pearson's correlation assumes linearity between variables and may not capture non linear associations.

→ It helps assess the strength and significance of associations among variables. It is useful for identifying multicollinearity in regression analysis.

→ It is sensitive to outliers and may not capture complex or nonlinear relationships effectively.

Date:_____

The formula for Karl Pearson correlation coefficient between two variables x and y is given by -

$$r = \frac{\Sigma dx\, dy}{\sqrt{\Sigma d_x^2 \, \Sigma d^2 y}} \quad ; \text{ for ungrouped data}$$

$$r = \frac{\Sigma fdxdy - \frac{\Sigma fdx \cdot \Sigma fdy}{N}}{\sqrt{\Sigma fdx^2 - \frac{(\Sigma fdx)^2}{N}} \cdot \sqrt{\Sigma fdy^2 - \frac{(\Sigma fdy)^2}{N}}} \quad ; \text{ for grouped data}$$

ALGORITHM:

→ For non-grouped data:

(i) Import the pandas library as pd.
(ii) Load the data from the CSV file located at '/content/Sleep_Efficiency.csv' into a pandas DataFrame named 'data'.
(iii) Drop the 'ID' column from the DataFrame 'data'.
(iv) Extract the columns "sleep duration" and "sleep efficiency" from the DataFrame and assign them to variables 'sleep_duration' and 'sleep_efficiency' respectively.
(v) Use the corr() method on 'sleep_duration' with 'sleep_efficiency' as the parameter to compute the Karl Pearson coefficient of correlation.
(vi) Use the corr() method on the DataFrame 'data' to compute the correlation coefficient matrix for all columns.

Date:_____

(vii) Print the Karl Pearson coefficient of correlation and the coefficient matrix to the console.

→ For Grouped data:

(i) Import pandas as pd and numpy as np.
(ii) Define a dictionary 'data' containing age ranges, sleep efficiency ranges, and corresponding frequencies.
(iii) Split the range strings and convert them into numerical values for age and sleep efficiency.
(iv) Create a function 'midpoint' to calculate the midpoint of a given range tuple.
(v) Apply the midpoint function to age and sleep efficiency range to get their midpoints.
(vi) Construct a DataFrame 'df' using the calculated midpoints and frequencies.
(vii) Create dictionaries to store correlation matrices and average correlations for each attribute.
(viii) Iterate over columns in the Data frame, compute the correlation matrix for each attribute, and store them.
(ix) Calculate the mean for age and sleep efficiency.
(x) Calculate the Karl Pearson coefficient using the formula.
(xi) Print the computed Karl Pearson coefficient.
(xii) Iterate over stored correlation matrices and print them.
(xiii) Print average correlations for each attribute along with interpretations based on their magnitude.

Date:_____

WORKING:

→ Calculation of Karl Pearson Coefficient of correlation between X (sleep duration) and Y (sleep efficiency)-

	X	Y	dx $(x-\bar{x})$	dy $(y-\bar{y})$	dx^2	dy^2	$dxdy$
	6	0.88	-1.35	0.142	1.8225	0.02016	-0.1917
	7	0.66	-0.35	-0.078	0.1225	0.00608	0.0273
	8	0.89	0.65	0.152	0.4225	0.02310	0.0988
	6	0.51	-1.35	-0.228	1.8225	0.05198	0.3078
	8	0.76	0.65	0.022	0.4225	0.00048	0.0143
	7.5	0.9	0.15	0.168	0.0225	0.02822	0.0252
	6	0.54	-1.35	-0.198	1.8225	0.03920	0.2673
	10	0.9	2.65	0.168	7.0225	0.02822	0.4452
	6	0.79	-1.35	0.052	1.8225	0.00270	-0.0702
	9	0.55	1.65	-0.188	2.7225	0.03534	-0.3102
$\Sigma=$	73.5	7.38			18.025	0.23548	0.6138
					(Σdx^2)	(Σdy^2)	$(\Sigma dxdy)$

$$\bar{X} = \frac{73.5}{10} = 7.35$$

$$\bar{Y} = \frac{7.38}{10} = 0.738$$

$$r = \frac{\Sigma dx\,dy}{\sqrt{\Sigma d^2x\,\Sigma d^2y}} = \frac{0.6138}{\sqrt{18.025 \times 0.23548}}$$

$$r = 0.2979$$

Date:_____

→ Karl Pearson coefficient of correlation for grouped data –

Age (X)	Sleep efficiency (Y)	Frequency (f)
10 - 20	0.5 - 0.6	5
20 - 30	0.6 - 0.7	9
30 - 40	0.7 - 0.8	12
40 - 50	0.8 - 0.85	7
50 - 60	0.85 - 0.9	15
60 - 70	0.9 - 1.0	7

$$r = \frac{\Sigma fdxdy - \frac{\Sigma fdx \cdot \Sigma fdy}{N}}{\sqrt{\Sigma fdx^2 - \frac{(\Sigma fdx)^2}{N}} \cdot \sqrt{\Sigma fdy^2 - \frac{(\Sigma fdy)^2}{N}}}$$

$r = 0.9919$

CONCLUSION:

Using Karl Pearson's coefficient of correlation for multivariate data offers a simple way to quantify linear relationships between variables. However, it assumes linearity, it is sensitive to outliers, and doesn't capture non-linear associations or non-normal distributions well.

20/3/24

```
import pandas as pd

# Load the data from the CSV file
file_path = '/content/Sleep_Efficiency.csv'
data = pd.read_csv(file_path)

# Exclude the 'ID' column
data = data.drop(columns=['ID'])

# Extract the columns "Sleep duration" and "Sleep efficiency"
sleep_duration = data['Sleep duration']
sleep_efficiency = data['Sleep efficiency']

# Calculate Karl Pearson coefficient of correlation
correlation_coefficient = sleep_duration.corr(sleep_efficiency)

# Create a coefficient matrix
coefficient_matrix = data.corr()

# Print the results
print("Karl Pearson Coefficient of Correlation:", correlation_coefficient)
print("\nCoefficient Matrix:")
print(coefficient_matrix)
```

```
Karl Pearson Coefficient of Correlation: -0.027466558164158498

Coefficient Matrix:
                            Age  Sleep duration  Sleep efficiency  \
Age                    1.000000       -0.062462          0.098357
Sleep duration        -0.062462        1.000000         -0.027467
Sleep efficiency       0.098357       -0.027467          1.000000
REM sleep percentage   0.042091       -0.015940          0.062362
Deep sleep percentage  0.021730       -0.037304          0.787335
Light sleep percentage -0.031905       0.041804         -0.819204
Awakenings             -0.017789       0.004939         -0.564979
Caffeine consumption   -0.171460       -0.014802         0.065082
Alcohol consumption    0.047188        -0.046243         -0.389624
Exercise frequency     0.072308        -0.068272         0.259563

                       REM sleep percentage  Deep sleep percentage  \
Age                                0.042091               0.021730
Sleep duration                    -0.015940              -0.037304
Sleep efficiency                   0.062362               0.787335
REM sleep percentage               1.000000              -0.208159
Deep sleep percentage             -0.208159               1.000000
Light sleep percentage            -0.017462              -0.974311
Awakenings                        -0.025332              -0.308267
Caffeine consumption               0.060037               0.001742
```

Alcohol consumption	-0.053258	-0.361731
Exercise frequency	0.031768	0.179102

	Light sleep percentage	Awakenings
Age	-0.031905	-0.017789
Sleep duration	0.041804	0.004939
Sleep efficiency	-0.819204	-0.564979
REM sleep percentage	-0.017462	-0.025332
Deep sleep percentage	-0.974311	-0.308267
Light sleep percentage	1.000000	0.321218
Awakenings	0.321218	1.000000
Caffeine consumption	-0.015593	-0.108615
Alcohol consumption	0.380571	0.206090
Exercise frequency	-0.190191	-0.219578

	Caffeine consumption	Alcohol consumption
Age	-0.171460	0.047188
Sleep duration	-0.014802	-0.046243
Sleep efficiency	0.065082	-0.389624
REM sleep percentage	0.060037	-0.053258
Deep sleep percentage	0.001742	-0.361731
Light sleep percentage	-0.015593	0.380571
Awakenings	-0.108615	0.206090
Caffeine consumption	1.000000	-0.123308
Alcohol consumption	-0.123308	1.000000
Exercise frequency	-0.068224	0.006934

	Exercise frequency
Age	0.072308
Sleep duration	-0.068272
Sleep efficiency	0.259563
REM sleep percentage	0.031768
Deep sleep percentage	0.179102
Light sleep percentage	-0.190191

```
port pandas as pd

Load the data from the CSV file
le_path = '/content/Sleep_Efficiency.csv'
ta = pd.read_csv(file_path)

Exclude the 'ID' column
ta = data.drop(columns=['ID'])

Extract the first 10 rows of the columns "Sleep duration" and "Sleep efficier
eep_duration = data['Sleep duration'].iloc[:10]
eep_efficiency = data['Sleep efficiency'].iloc[:10]

Calculate Karl Pearson coefficient of correlation
rrelation_coefficient = sleep_duration.corr(sleep_efficiency)
```

```
# Create a coefficient matrix for the first 10 rows
coefficient_matrix = data.iloc[:10].corr()

# Print the results
print("Karl Pearson Coefficient of Correlation (First 10 rows):", correlation_C
print("\nCoefficient Matrix (First 10 rows):")
print(coefficient_matrix)
```

	Age	Sleep duration	Sleep efficiency	\
Age	1.000000	-0.136458	0.439861	
Sleep duration	-0.136458	1.000000	0.292217	
Sleep efficiency	0.439861	0.292217	1.000000	
REM sleep percentage	-0.045240	0.005491	0.041018	
Deep sleep percentage	0.260468	0.191487	0.902765	
Light sleep percentage	-0.248553	-0.192110	-0.909367	
Awakenings	-0.310508	-0.136532	-0.769122	
Caffeine consumption	-0.232544	-0.210819	-0.208751	
Alcohol consumption	0.323376	-0.247879	-0.488986	
Exercise frequency	0.852037	0.186901	0.571650	

	REM sleep percentage	Deep sleep percentage	\
Age	-0.045240	0.260468	
Sleep duration	0.005491	0.191487	
Sleep efficiency	0.041018	0.902765	
REM sleep percentage	1.000000	-0.106246	
Deep sleep percentage	-0.106246	1.000000	
Light sleep percentage	-0.136265	-0.970588	
Awakenings	-0.030596	-0.616723	
Caffeine consumption	0.735691	-0.370196	
Alcohol consumption	-0.030755	-0.511961	
Exercise frequency	-0.060581	0.440082	

	Light sleep percentage	Awakenings	\
Age	-0.248553	-0.310508	
Sleep duration	-0.192110	-0.136532	
Sleep efficiency	-0.909367	-0.769122	
REM sleep percentage	-0.136265	-0.030596	
Deep sleep percentage	-0.970588	-0.616723	
Light sleep percentage	1.000000	0.621856	
Awakenings	0.621856	1.000000	
Caffeine consumption	0.184906	-0.072548	
Alcohol consumption	0.517519	0.476138	
Exercise frequency	-0.423791	-0.432405	

	Caffeine consumption	Alcohol consumption	\
Age	-0.232544	0.323376	
Sleep duration	-0.210819	-0.247879	
Sleep efficiency	-0.208751	-0.488986	
REM sleep percentage	0.735691	-0.030755	
Deep sleep percentage	-0.370196	-0.511961	
Light sleep percentage	0.184906	0.517519	
Awakenings	-0.072548	0.476138	

```
Caffeine consumption          1.000000        0.013710
Alcohol consumption           0.013710        1.000000
Exercise frequency           -0.387298        0.105183

                           Exercise frequency
Age                              0.852037
Sleep duration                   0.186901
Sleep efficiency                 0.571650
REM sleep percentage            -0.060581
Deep sleep percentage            0.440082
Light sleep percentage          -0.423791
Awakenings                      -0.432405
Caffeine consumption            -0.387298
Alcohol consumption              0.105183
Exercise frequency               1.000000
```

```
import pandas as pd
import numpy as np

# Given data
data = {
    'Age': ['10-20', '20-30', '30-40', '40-50', '50-60', '60-70'],
    'Sleep_Efficiency': ['0.5-0.6', '0.6-0.7', '0.7-0.8', '0.8-0.85', '0.85-0.
    'Frequency': [5, 9, 12, 7, 15, 7],
}

# Convert ranges to numerical values
age_values = [(int(age.split('-')[0]), int(age.split('-')[1])) for age in data
sleep_efficiency_values = [(float(se.split('-')[0]), float(se.split('-')[1]))

# Function to convert range to midpoint
def midpoint(range_tuple):
    return (range_tuple[0] + range_tuple[1]) / 2

# Convert ranges to midpoints
age_midpoints = [midpoint(range_tuple) for range_tuple in age_values]
sleep_efficiency_midpoints = [midpoint(range_tuple) for range_tuple in sleep_e

# Create DataFrame from midpoints and frequency
df = pd.DataFrame({'Age': age_midpoints, 'Sleep_Efficiency': sleep_efficiency_

# Initialize dictionary to store correlation matrices and average correlations
correlation_matrices = {}
average_correlations = {}

# Calculate correlation matrix for each attribute
for column in df.columns:
    correlation_matrix = df.corr()[[column]]
    correlation_matrices[column] = correlation_matrix
```

```
    # Calculate average correlation for the attribute
    avg_corr = correlation_matrix[column].mean()
    average_correlations[column] = avg_corr

# Compute mean for Age and Sleep Efficiency
mean_age = np.mean(df['Age'])
mean_sleep_efficiency = np.mean(df['Sleep_Efficiency'])

# Compute Karl Pearson coefficient
numerator = ((df['Age'] - mean_age) * (df['Sleep_Efficiency'] - mean_sleep_eff
denominator_age = np.sqrt(((df['Age'] - mean_age) ** 2 * df['Frequency']).sum(
denominator_sleep_efficiency = np.sqrt(((df['Sleep_Efficiency'] - mean_sleep_e

karl_pearson_coefficient = numerator / (denominator_age * denominator_sleep_ef

print("Karl Pearson Coefficient:", karl_pearson_coefficient)
# Print correlation matrices for each attribute
for attribute, matrix in correlation_matrices.items():
    print(f"\nCorrelation Matrix for {attribute}:")
    print(matrix)

# Print average correlations along with interpretations
print("\nAverage Correlations and Interpretations:")
for attribute, avg_corr in average_correlations.items():
    if avg_corr < 0.3:
        interpretation = "Low correlation"
    elif avg_corr < 0.7:
        interpretation = "Moderate correlation"
    else:
        interpretation = "High correlation"
    print(f"{attribute}: Average Correlation = {avg_corr:.2f} ({interpretation
```

20/3/24

```
Karl Pearson Coefficient: 0.9919666297130671

Correlation Matrix for Age:
                       Age
Age                1.000000
Sleep_Efficiency   0.992914
Frequency          0.331344

Correlation Matrix for Sleep_Efficiency:
                  Sleep_Efficiency
Age                       0.992914
Sleep_Efficiency          1.000000
Frequency                 0.358038

Correlation Matrix for Frequency:
                  Frequency
Age                0.331344
Sleep_Efficiency   0.358038
Frequency          1.000000

Average Correlations and Interpretations:
Age: Average Correlation = 0.77 (High correlation)
Sleep_Efficiency: Average Correlation = 0.78 (High correlation)
Frequency: Average Correlation = 0.56 (Moderate correlation)
```

20/3/24