Name:Kaushik Kotian Roll no.:30 Div:D15B Batch:B

Experiment No.:4

AIM: Implementation of Statistical Hypothesis Test using Scipy and Sci-kit learn.

PROBLEM STATEMENT:

1. Perform the following Tests:
2. Pearson's Correlation Coefficient
3. Spearman's Rank Correlation
4. Kendall's Rank Correlation
5. Chi-Squared Test

THEORY:

1.Pearson's Correlation Coefficient

Theory: Pearson's Correlation Coefficient (r) measures the linear relationship between two continuous variables. It quantifies the degree to which a change in one variable corresponds to a change in another. The coefficient's value ranges from -1 to 1, where 1 indicates a perfect positive linear relationship, -1 indicates a perfect negative linear relationship, and 0 indicates no linear relationship.
Application: It is used when you want to assess the strength and direction of a linear relationship between two variables. It's important that the data meet the assumptions of normality, linearity, and homoscedasticity.

2.Spearman's Rank Correlation

Theory: Spearman's Rank Correlation Coefficient (ρ) is a non-parametric measure of rank correlation. It assesses how well the relationship between two variables can be described using a monotonic function. Unlike Pearson's, it does not assume linearity or normal distribution of the data.
Application: It's particularly useful when the data are ordinal or when the assumptions of Pearson's correlation are not met. It can also be used with continuous data that is not normally distributed.

3.Kendall's Rank Correlation

Theory: Kendall's Rank Correlation Coefficient (τ) measures the ordinal association between two quantities. It calculates the difference between the probability that the observed data are in the same order versus the probability that they are in different orders.

Application: Kendall's Tau is useful for small sample sizes or for data with many ties. It is less sensitive to errors and outliers compared to Pearson's and Spearman's correlations. Like Spearman's, it does not require the data to be linear or normally distributed.

4.Chi-Squared Test

Theory: The Chi-Squared Test assesses whether there is a significant association between two categorical variables. It compares the observed frequencies in each category against the expected frequencies if there were no association between the variables.

Application: It is used to test hypotheses about the independence of two variables in a contingency table. It's important for the expected frequencies to be sufficiently large (typically at least 5) in each cell of the contingency table to ensure the validity of the test.

SciPy

Purpose: SciPy is an open-source Python library used for scientific and technical computing. It builds on NumPy, providing a large number of higher-level functions that are useful in optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, and statistics.

Statistical Tests: Includes a wide range of probability distributions, statistical functions, and tests. For example, it provides functions for Pearson's Correlation Coefficient, Spearman's Rank Correlation, Kendall's Tau, and the Chi-Squared Test, which are essential for the statistical hypothesis testing mentioned in your problem statement.

Integration and Optimization: Offers modules for numerical integration and optimization functions.

Linear Algebra: Provides functions for linear algebra operations.

Interpolation: Includes tools for interpolating functions.

Scikit-learn

Purpose: Scikit-learn is an open-source machine learning library for Python. It is built on NumPy, SciPy, and Matplotlib. It provides simple and efficient tools for data mining and data analysis. It's accessible to everybody and reusable in various contexts.

Machine Learning Models: Offers a wide range of supervised and unsupervised learning algorithms. This includes classification, regression, clustering, and dimensionality reduction.

Model Selection and Evaluation: Provides tools for model selection, evaluation, and hyperparameter tuning, such as cross-validation and grid search.

Preprocessing: Includes data preprocessing tools like normalization, scaling, and handling of missing values.

IMPLEMENTATION:

1. **Loading dataset into Google collab.**

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files  CVD_cleaned.csv
• CVD_cleaned.csv(text/csv) - 32453765 bytes, last modified: 2/9/2024 - 100% done
Saving CVD_cleaned.csv to CVD_cleaned.csv
```

```
[2] import pandas as pd
    import matplotlib.pyplot as plt
    import numpy as np
    import seaborn as sb

    df=pd.read_csv('CVD_cleaned.csv')
```

```
[3] df.head()
```

pression	Diabetes	Arthritis	Sex	Age_Category	Height_(cm)	Weight_(kg)	BMI	Sm
No	No	Yes	Female	70-74	150.0	32.66	14.54	
No	Yes	No	Female	70-74	165.0	77.11	28.29	
No	Yes	No	Female	60-64	163.0	88.45	33.47	
No	Yes	No	Male	75-79	180.0	93.44	28.73	
No	No	No	Male	80+	191.0	88.45	24.37	

2.Pearson's Correlation Coefficient

Pandas dataframe.corr() is used to find the pairwise correlation of all columns in the Pandas Dataframe in Python. Any NaN values are automatically excluded. Any non-numeric data type or columns in the Dataframe, it is ignored.

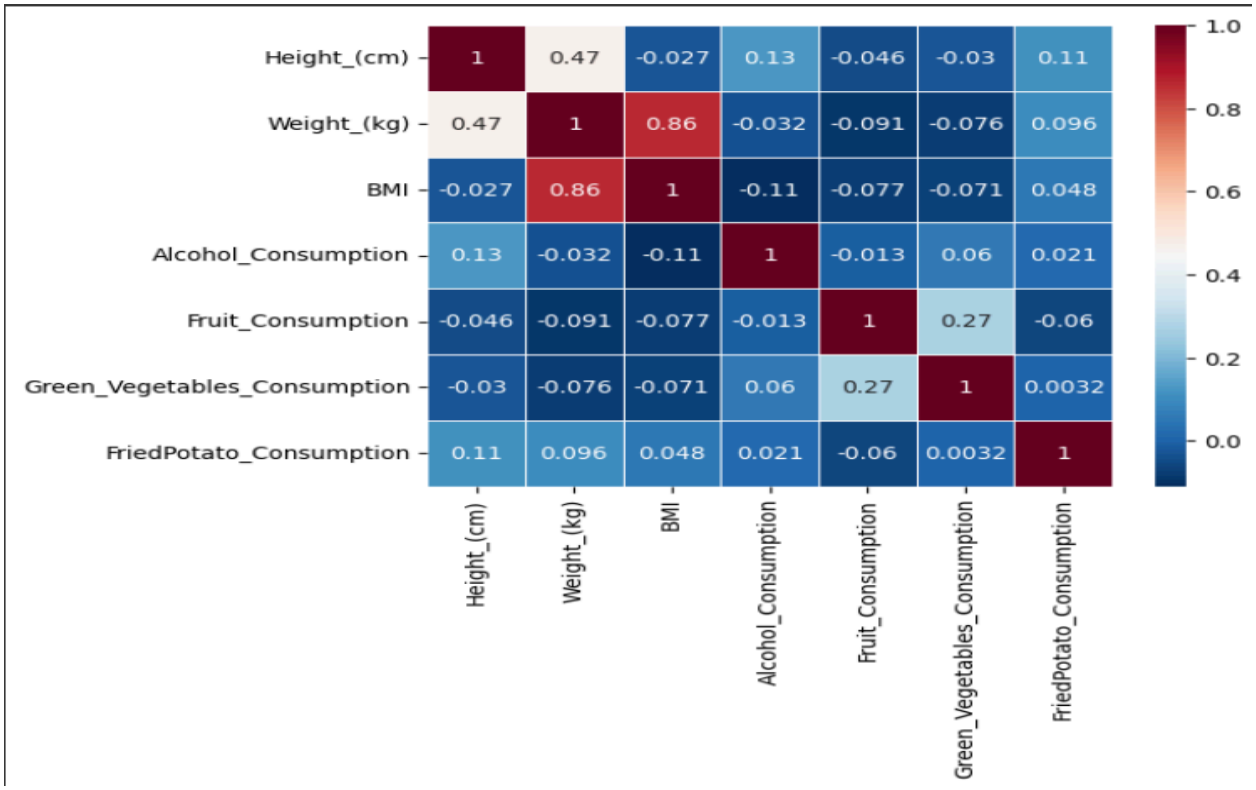
```
pearsoncorr=df.corr(method='pearson')
pearsoncorr
```

```
he default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False
```

_(kg)	BMI	Alcohol_Consumption	Fruit_Consumption	Green_Vegetables_Consumption	FriedPotato_Consumption
72186	-0.027408	0.128835	-0.045911	-0.030148	0.108795
00000	0.859699	-0.032373	-0.090612	-0.075904	0.096351
59699	1.000000	-0.108684	-0.076611	-0.070640	0.048366
32373	-0.108684	1.000000	-0.012562	0.060053	0.020543
90612	-0.076611	-0.012562	1.000000	0.270430	-0.060311
75904	-0.070640	0.060053	0.270430	1.000000	0.003180
96351	0.048366	0.020543	-0.060311	0.003180	1.000000

To make this look beautiful and easier to interpret, we have made a heat map after calculating the Pearson coefficient of correlation.

```
import seaborn as sb
sb.heatmap(pearsoncorr, xticklabels=pearsoncorr.columns,
           yticklabels=pearsoncorr.columns, cmap='RdBu_r',
           annot = True, linewidth=0.5)
```



3.Spearman's Rank Correlation

```
[10] from scipy.stats import spearmanr
     df['Height_(cm)'].corr(df['Weight_(kg)'], method ='spearman')
```

```
     0.5075130498631143
```

```
[11] from scipy.stats import spearmanr
     df['BMI'].corr(df['Weight_(kg)'], method ='spearman')
```

```
     0.8463610831356277
```

```
[13] from scipy.stats import spearmanr
     df['Alcohol_Consumption'].corr(df['BMI'], method ='spearman')
```

```
     -0.09538025062716514
```

4.Kendall's Rank Correlation

Pandas dataframe.corr() is used to find the pairwise correlation of all columns in the dataframe.

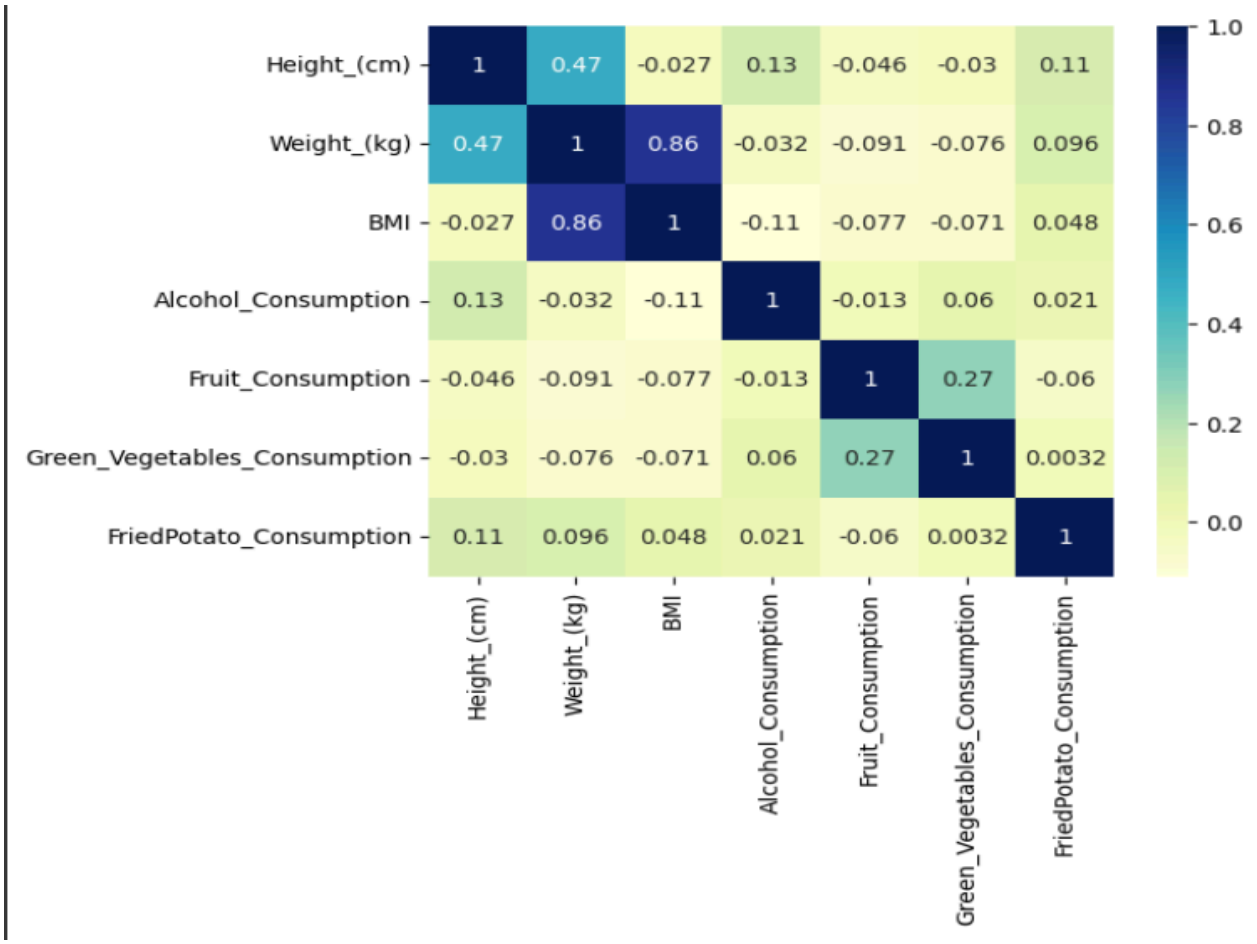
```
from scipy.stats.stats import kendalltau
corr = df.corr(method='kendall')
corr
```

```
-8d06292f6146>:1: DeprecationWarning: Please use `kendalltau` from the `scipy.stats` namespace, the `scipy.stats.stats` namespace is deprecated.
s.stats import kendalltau
-8d06292f6146>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False
method='kendall')
```

	Height_(cm)	Weight_(kg)	BMI	Alcohol_Consumption	Fruit_Consumption	Green_Vegetables_Consumption	FriedPotato_Consumption
cm)	1.000000	0.368892	0.010106	0.112344	-0.036793	-0.022748	0.123956
kg)	0.368892	1.000000	0.663507	-0.004053	-0.071476	-0.053788	0.117636
	0.010106	0.663507	1.000000	-0.069568	-0.063491	-0.050596	0.066076
umption	0.112344	-0.004053	-0.069568	1.000000	0.005458	0.077971	0.054649
mption	-0.036793	-0.071476	-0.063491	0.005458	1.000000	0.237000	-0.092833
Consumption	-0.022748	-0.053788	-0.050596	0.077971	0.237000	1.000000	-0.067444
nsumption	0.123956	0.117636	0.066076	0.054649	-0.092833	-0.067444	1.000000

Visualize using a Heat-map

```
[15] import seaborn as sb
     sb.heatmap(pearsoncorr, xticklabels=corr.columns.values,
                yticklabels=corr.columns.values, cmap='YlGnBu',
                annot = True)
```



5.Chi-Squared Test

SciPy's chi2_contingency() returns four values, $\chi 2$ value, p-value, degree of freedom and expected values.

```
from scipy.stats import chi2_contingency
contingency= pd.crosstab(df['Alcohol_Consumption'], df['FriedPotato_Consumption'])
c, p, dof, expected = chi2_contingency(contingency)
print(p)
```

```
0.0
```

CONCLUSION:

Through the implementation of statistical hypothesis tests using SciPy, including Pearson's, Spearman's, Kendall's correlations, and the Chi-Squared Test, we examined relationships between variables in our dataset. These tests enabled us to validate or refute our hypotheses, offering insights into variable associations and emphasizing the value of statistical analysis in data-driven decision-making.