Title: Extension of previously built application to perform given pre-processing tasks.

Objective/Aim: To perform

1. Correlation analysis - Chi-Square Test
2. Correlation analysis – Correlation coefficient (Pearson coefficient) & Covariance
3. Normalization using various techniques.

Introduction:

Correlation Analysis: is a statistical method used to measure the strength of the linear relationship between two variables and compute their association. Simply put - correlation analysis calculates the level of change in one variable due to the change in the other.

Normalization: In statistics and applications of statistics, normalization can have a range of meanings.[1] In the simplest cases, normalization of ratings means adjusting values measured on different scales to a notionally common scale, often prior to averaging. In more complicated cases, normalization may refer to more sophisticated adjustments where the intention is to bring the entire probability distributions of adjusted values into alignment.

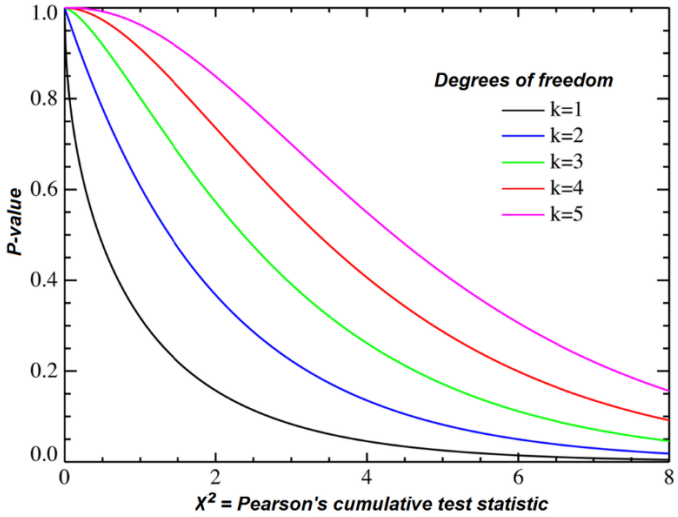
Theory /Block Diagrams:

Methods of correlation analysis:

1. Chi-square test: A chi-squared test, also written as $\chi 2$ test, is a statistical hypothesis test that is valid to perform when the test statistic is chi-squared distributed under the null hypothesis, specifically Pearson's chi-squared test and variants thereof. Pearson's chi-squared test is used to determine whether there is a statistically significant difference between the expected frequencies and the observed frequencies in one or more categories of a contingency table.

   Chi-squared Distribution:

Authors: 2018BTECS00050
2018BTECS00064



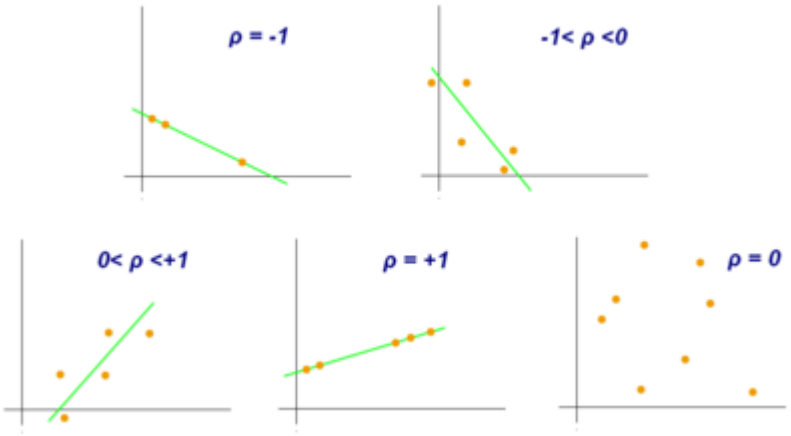
Formula:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

χ^2 = chi squared
O_i = observed value
E_i = expected value

2. Correlation coefficient (Pearson coefficient) & Covariance: In statistics, the Pearson correlation coefficient — also known as Pearson's r, the Pearson product-moment correlation coefficient, the bivariate correlation, or colloquially simply as the correlation coefficient — is a measure of linear correlation between two sets of data.

   Scatter diagrams with different values of correlation coefficient:



Methods of normalization:

Authors: 2018BTECS00050
2018BTECS00064

1. Min-max normalization: Min-max normalization is one of the most common ways to normalize data. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1.

   Formula:
   $$\frac{value - min}{max - min}$$
2. Z-score normalization: If a value is exactly equal to the mean of all the values of the feature, it will be normalized to 0. If it is below the mean, it will be a negative number, and if it is above the mean it will be a positive number.

   Formula (Basic): $z = (x - \mu) / \sigma$
3. Normalization by decimal scaling: Decimal scaling is a data normalization technique. In this technique, we move the decimal point of values of the attribute. This movement of decimal points totally depends on the maximum value among all values in the attribute.
   Formula: Normalized value of attribute= $(v^i / 10^j)$

Procedure/ Algorithm:

Chi-square Test:

```
def chi2test_with_x_y():
    table = [   x,y  ]
    stat, p, dof, expected = chi2_contingency(table)
    prob = 0.95
    critical = chi2.ppf(prob, dof)
    if abs(stat) >= critical:
        print("Dependent (reject H0)")
    else:
        print("Independent (fail to reject H0)")
    print(str(expected))
```

Pearson Coefficient:

```
def pearsoncoef_with_x_y():
    corr, _ = pearsonr(x, y)
    str=""
    str+='Pearsons correlation: %.3f' % corr
    if(corr<0):
        str+="\nNegative correlation exists"
    if(corr>0):
        str+="\nPositive correlation exists"
```


Authors: 2018BTECS00050
2018BTECS00064

```
    if(corr==0):
        str+="\nNo correlation exists"
    print(str)
```

Covariance:

```
def covariance_with_x_y():
    cov_mat = np.stack((x, y), axis = 0)
    print(str(cov_mat))
    cov_mat = np.stack((x, y), axis = 1)
    print(str(cov_mat))
```

Normalization:

1.Min-Max:

```
def min_max_norm_x():
    x_min_max_scaled = x.copy()
    x_min_max_scaled = (x_min_max_scaled[5] - x_min_max_scaled[0]) / (x_min_ma
x_scaled[5] - x_min_max_scaled[0])
    plt.scatter(x_min_max_scaled, x_min_max_scaled)
    plt.show()
    print(str(x_min_max_scaled))
```

2.Z-score:

```
def z_score_norm_x():
    x_z_scaled = np.array(x)
    x_z_scaled = (x_z_scaled[0] - x_z_scaled[3]) / x_z_scaled.std()
    plt.scatter(x_z_scaled, x_z_scaled)
    plt.show()
    print(str(x_z_scaled))
```

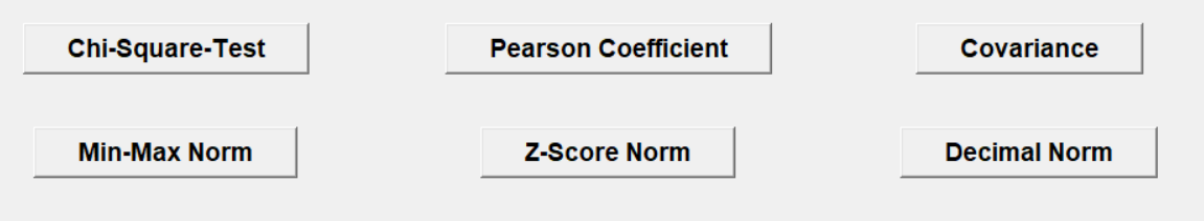
3.Decimal Scaling:

```
def dec_scale_norm_x():
    p = x[5]
    q = len(str(abs(p)))
    x_des_scaled = x[5]/10**q
    plt.scatter(x_des_scaled, x_des_scaled)
    plt.show()
    print(str(x_des_scaled))
```

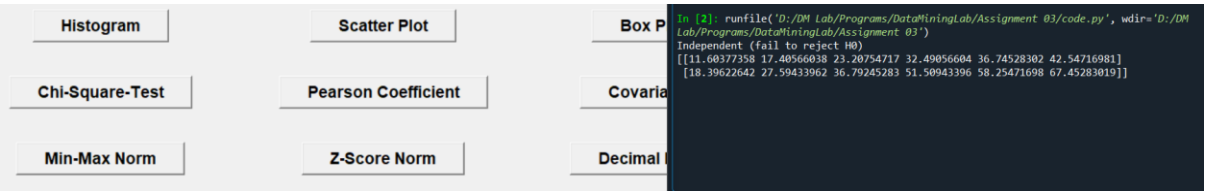

Authors: 2018BTECS00050
2018BTECS00064

Actual Experimentation/ simulation/ result/ Observation:

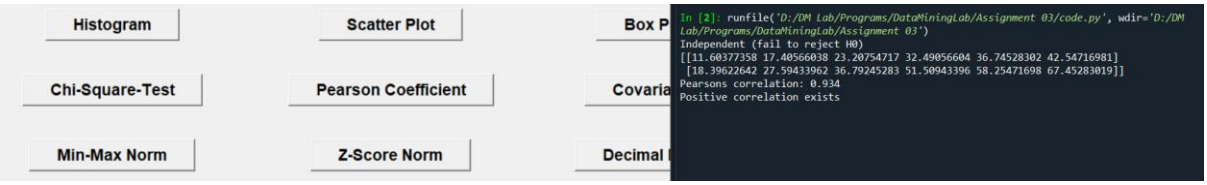
GUI:



Chi-Square:

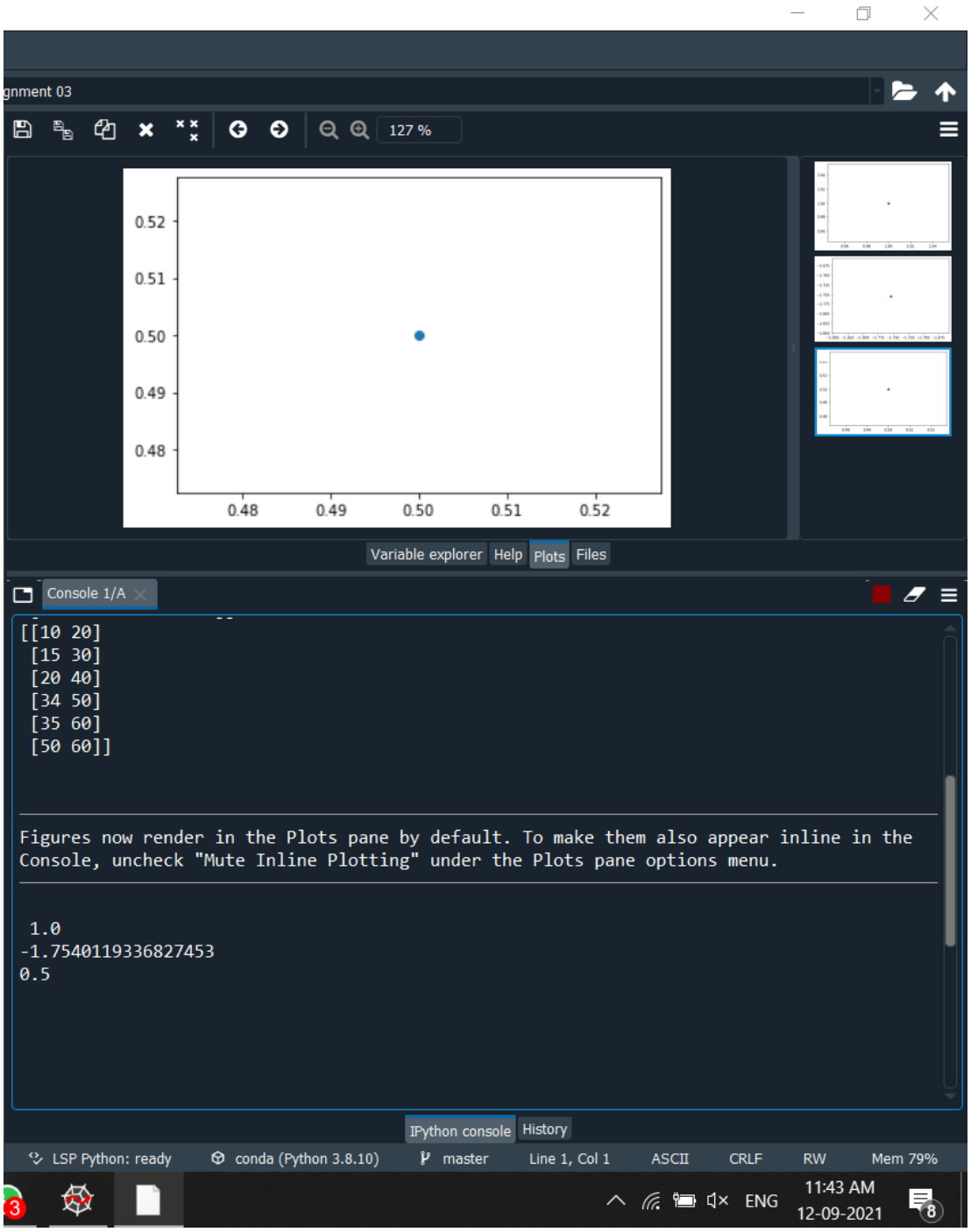


Pearson's coefficient:



Normalization:

Authors: 2018BTECS00050
2018BTECS00064



Conclusion:

From this assignment we could implement the theoretical knowledge of

1. Chi-square test
2. Pearson's coefficient
3. Normalization by:
    a. Min-Max

Authors: 2018BTECS00050
2018BTECS00064

b. Z-score
c. Decimal scaling

References:

1. www.statisticshowto.com
2. www.codecademy.com
3. en.wikipedia.org

Authors: 2018BTECS00050
2018BTECS00064