CCT College Dublin

Assessment Cover Page

To be provided separately as a word doc for students to include with every submission

Module Title:	*Programming for DA* *Statistics for Data Analytics* *Machine Learning for Data Analysis* *Data Preparation & Visualisation*
Assessment Title:	*Individual*
Lecturer Name:	*Marina Iantorno/John O'Sullivan* *Sam Weiss* *Muhammad Iqbal* *David McQuaid*
Student Full Name:	Dovlet Reyimov
Student Number:	2022341
Assessment Due Date:	06.01.2023
Date of Submission:	06.01.2023

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Contents

References

www.fao.org : Food and Agrriculture Organization of the United Nations.

McCornack, R. L. (1965). Extended tables of the Wilcoxon matched pair signed rank statistic. *Journal of the American Statistical Association, 60*(311), 864–871. https://doi.org/10.2307/2283253.

T-test with Python (pythonfordatascience.org).

https://labs.la.utexas.edu/gilden/files/2016/05/Statistics-Text.pdf.

https://thedatascientist.com/.

Sjoukje A. Osinga, Dilli Paudel, Spiros A. Mouzakitis, Ioannis N. Athanasiadis (2022): “Big data in agriculture: Between opportunity and solution” , https://doi.org/10.1016/j.agsy.2021.103298,(https://www.sciencedirect.com/science/article/pii/S0308521X21002511).

Andy Field’ın Discovering Statistics Using SPSS (Sage, 2005) adlı eserinin 8., 9. ve 10

McCornack, R. L. (1965). Extended tables of the Wilcoxon matched pair signed rank statistic

Sampling Distributions and Hypothesis Testing 85

www.investopedia.com

https://medium.com/analytics-vidhya/seaborn-cheat-sheet

www.qualtrics.com

www.activestate.com/resources/datasheets/

Head First Statistics: A Brain-Friendly Guide

Practical Statistics for Data Scientists

Introduction to Probability

Introduction to Machine Learning with Python: A Guide for Data Scientists

Python Machine Learning By Example

Pattern recognition and machine learning

Python for data analysis

İmport Library

Seaborn Library. Seaborn is a statistical Python data visualization library based on the Matplotlib library. Seaborn offers users a high-level interface for making statistical visualizations. Using the Seaborn Library's Functions A dataset-based API (Application Programming Interface) for examining the relationships between multiple variables in detail. Support for observing and performing statistical operations on categorical variables Supports multi-plot grids to create complex visualizations Univariate and bivariate visualizations to compare between subsets Wide color gamut that makes visualization clearer Seaborn and Matplotlib Library Seaborn offers different solutions to the user with its color palette, interfaces and graphic varieties, so I visualized my data by using these libraries. When working with Pandas Library, Seaborn functions work on Pandas DataFrames while Matplotlib does not accept DataFrames.

Import Datas.

FOASTAT_data:

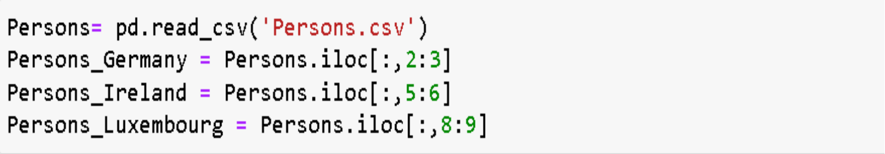
Land_data:

Ireland_data:

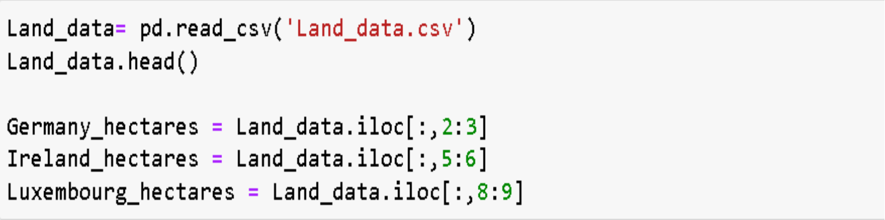
Luxembourg_data:

Germany_data:

I called Jupyter notebook and continued by explaining them. I explained these data with diagrams.

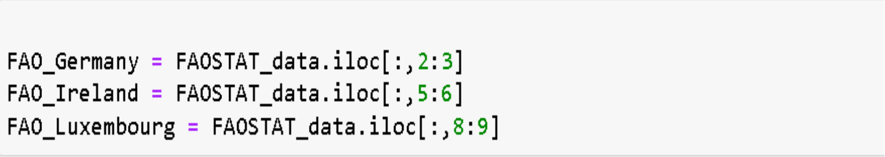


```
Persons= pd.read_csv('Persons.csv')
Persons_Germany = Persons.iloc[:,2:3]
Persons_Ireland = Persons.iloc[:,5:6]
Persons_Luxembourg = Persons.iloc[:,8:9]
```

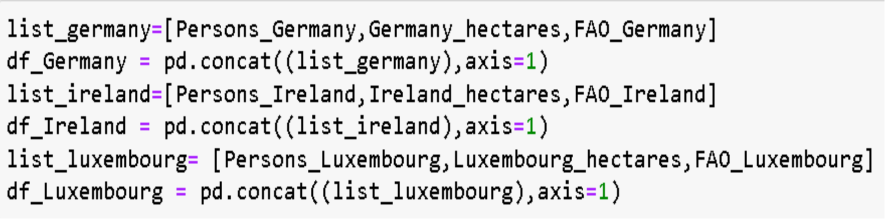


```
Land_data= pd.read_csv('Land_data.csv')
Land_data.head()

Germany_hectares = Land_data.iloc[:,2:3]
Ireland_hectares = Land_data.iloc[:,5:6]
Luxembourg_hectares = Land_data.iloc[:,8:9]
```



```
FAO_Germany = FAOSTAT_data.iloc[:,2:3]
FAO_Ireland = FAOSTAT_data.iloc[:,5:6]
FAO_Luxembourg = FAOSTAT_data.iloc[:,8:9]
```



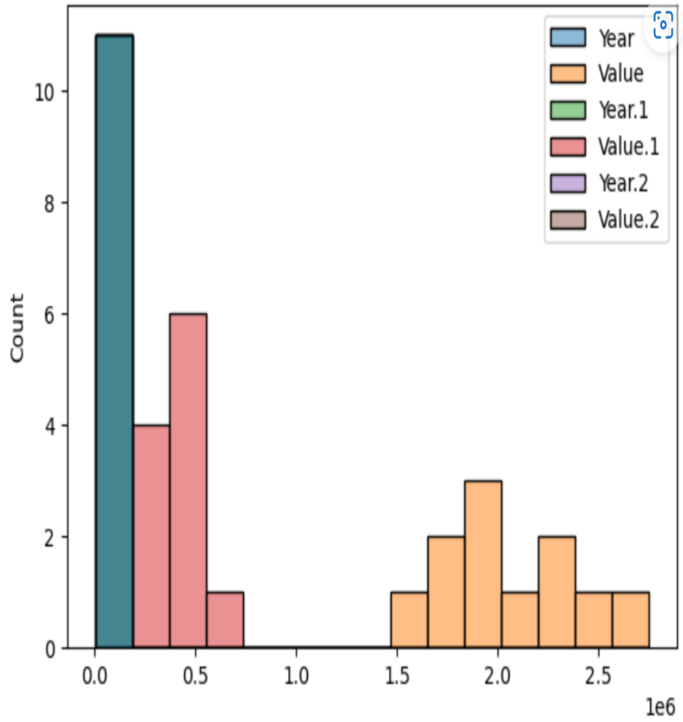
```
list_germany=[Persons_Germany,Germany_hectares,FAO_Germany]
df_Germany = pd.concat((list_germany),axis=1)
list_ireland=[Persons_Ireland,Ireland_hectares,FAO_Ireland]
df_Ireland = pd.concat((list_ireland),axis=1)
list_luxembourg= [Persons_Luxembourg,Luxembourg_hectares,FAO_Luxembourg]
df_Luxembourg = pd.concat((list_luxembourg),axis=1)
```

Histogram Calculations

Histplot

I plotted a univariate or bivariate histogram to show the distributions of the datasets.The histogram is a classic visualization tool that represents the distribution of one or more variables by counting the number of observations that fall into separate boxes.

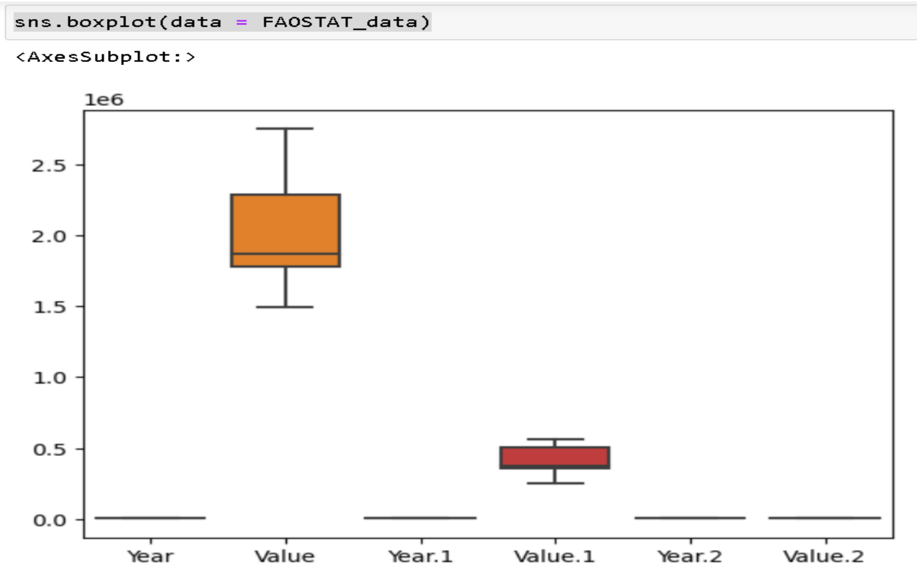
This function can normalize the statistic calculated at each bin to estimate the frequency, density or probability mass and add a smooth curve obtained using a kernel density estimation like the one below.



Seaborn.boxplot

I drew a box plot to show the distributions by category.

A boxplot (or box and whisker plot) shows the distribution of quantitative data across variables or levels of a categorical variable in a way that facilitates comparisons. The box shows the quartiles of the dataset, while the horizontal lines expand to show the rest of the distribution, except for the points identified as "outliers" using a method that is a function of interquartile range. The Pandas library plays a pretty important role. I visualized the combat stats of all Pokémon using a boxplot.

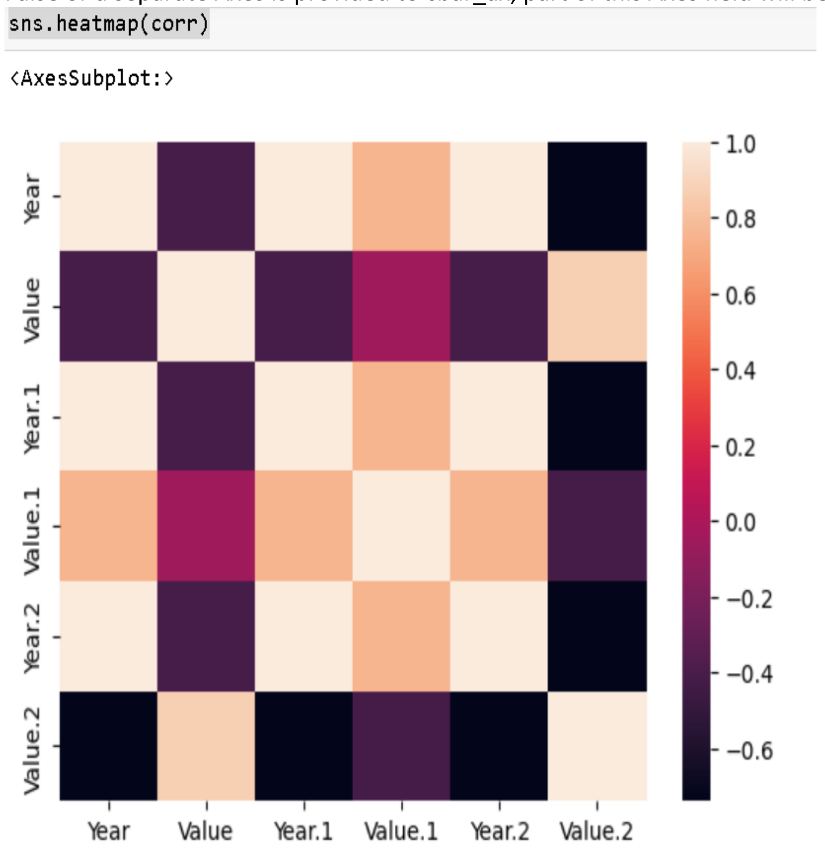


```
sns.boxplot(data = FAOSTAT_data)
```

```
<AxesSubplot:>
```

Heat Map

I visualized heatmaps in matrix style FAO_data. I plotted rectangular data as a color-coded matrix. This is an Axis level function and will plot the heatmap to the currently active Axes if none is supplied to the ax argument. Unless cbar is False or a separate Axes is provided to cbar_ax, part of this Axes field will be taken and used to draw a colormap.



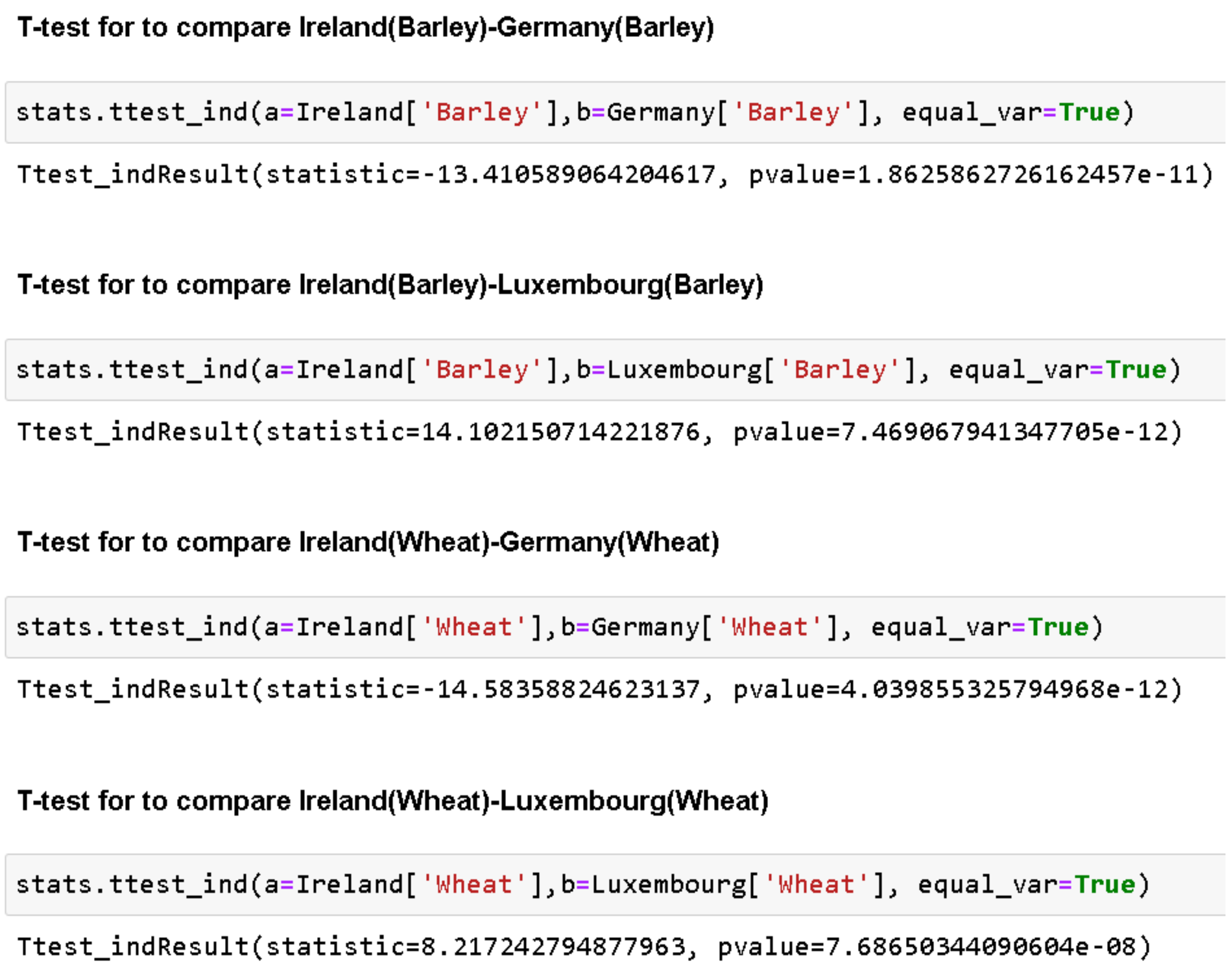
```
sns.heatmap(corr)
```

```
<AxesSubplot:>
```

I applied Histogram Calculations in all my datasets. I did not specify all of them in the report, by visualizing them because I am aware that I should not exceed 3000 words.

T-Test

A t-test is an inferential statistic I used to determine if there was a significant difference between the means of the two groups and how they were related. I used t-tests because the datasets follow a normal distribution and have unknown variances, such as the dataset recorded by flipping a coin 100 times. The t test is a test used for hypothesis testing in statistics and I used the t statistic, t distribution values and degrees of freedom to determine statistical significance. KEY PREDICTIONS A t test is an inferential statistic that is used to determine if there is a statistically significant difference between the means of two variables. T-test is a test used for hypothesis testing in statistics. Computing a t-test requires three baseline data values, including the difference between the mean values from each data set, the standard deviation of each group, and the number of data values. T-tests can be dependent or independent.



T-test for to compare Ireland(Barley)-Germany(Barley)

```
stats.ttest_ind(a=Ireland['Barley'],b=Germany['Barley'], equal_var=True)
```

```
Ttest_indResult(statistic=-13.410589064204617, pvalue=1.8625862726162457e-11)
```

T-test for to compare Ireland(Barley)-Luxembourg(Barley)

```
stats.ttest_ind(a=Ireland['Barley'],b=Luxembourg['Barley'], equal_var=True)
```

```
Ttest_indResult(statistic=14.102150714221876, pvalue=7.469067941347705e-12)
```

T-test for to compare Ireland(Wheat)-Germany(Wheat)

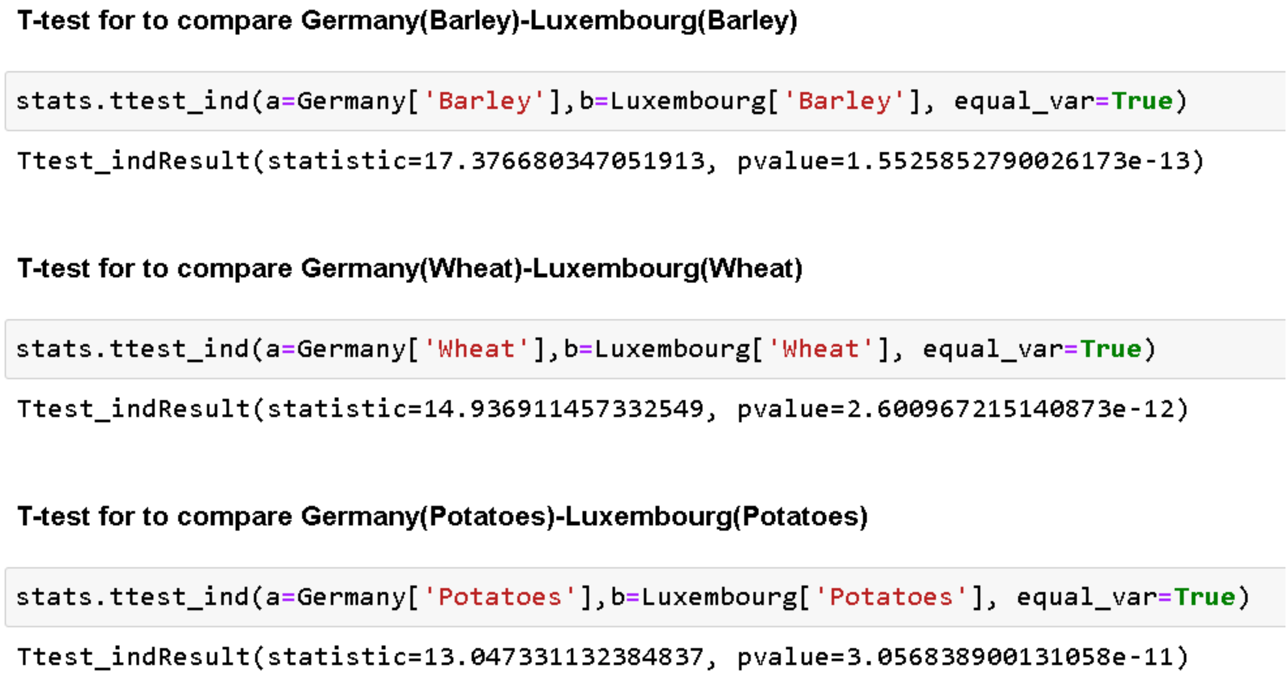
```
stats.ttest_ind(a=Ireland['Wheat'],b=Germany['Wheat'], equal_var=True)
```

```
Ttest_indResult(statistic=-14.58358824623137, pvalue=4.039855325794968e-12)
```

T-test for to compare Ireland(Wheat)-Luxembourg(Wheat)

```
stats.ttest_ind(a=Ireland['Wheat'],b=Luxembourg['Wheat'], equal_var=True)
```

```
Ttest_indResult(statistic=8.217242794877963, pvalue=7.68650344090604e-08)
```



T-test for to compare Germany(Barley)-Luxembourg(Barley)

```
stats.ttest_ind(a=Germany['Barley'],b=Luxembourg['Barley'], equal_var=True)
```

```
Ttest_indResult(statistic=17.376680347051913, pvalue=1.5525852790026173e-13)
```

T-test for to compare Germany(Wheat)-Luxembourg(Wheat)

```
stats.ttest_ind(a=Germany['Wheat'],b=Luxembourg['Wheat'], equal_var=True)
```

```
Ttest_indResult(statistic=14.936911457332549, pvalue=2.600967215140873e-12)
```

T-test for to compare Germany(Potatoes)-Luxembourg(Potatoes)

```
stats.ttest_ind(a=Germany['Potatoes'],b=Luxembourg['Potatoes'], equal_var=True)
```

```
Ttest_indResult(statistic=13.047331132384837, pvalue=3.056838900131058e-11)
```

I ran a T-Test across my Datasets as a T-test is the final statistical measure to identify differences between two tools that may or may not be related. The test uses randomly selected samples from two categories or groups. I obtained the results with a statistical method where the samples were randomly selected and there was no perfect normal distribution.

T-Test Description A t-test examines a set of data collected (My datasets) from two similar or different groups to determine the probability that the result will differ from what is usually obtained. The accuracy of the test depends on several factors, including the distribution models used and variables that affect the samples collected. Based on the parameters, I tested and got a T-value as statistical inference of the probability that the usual outcome was by chance. I compared my datasets and found that he could randomly select agricultural products between these countries and come to a standard conclusion.

The final T-test interpretation was obtained in one of two ways: a null hypothesis indicates that the difference between the means is zero and both tools are shown as equal. An alternative hypothesis implies that the difference between the means is nonzero. This hypothesis rejects the null hypothesis, indicating that the data set is fairly accurate and not by chance.

assumptions The test runs on a set of assumptions such as: The measurement scale used for such hypothesis testing follows a series of continuous or sequential patterns. Calculated parameters and variants affecting samples and surrounding groups are based on standard evaluation. The tests are based on completely random sampling. Reliability is often questioned, as no individuality is preserved in the samples. When the data was plotted according to the T-test distribution, I followed a normal distribution.

Wilcoxon Test

In the Wilcoxon test, which can refer to either the rank sum test or the signed rank test version, I compared my dataset to two paired groups and got a non-parametric statistical test result.

The tests basically calculate the difference between sets of pairs and I have determined whether these differences are statistically significantly different from each other.

In both versions of the model, I assumed that the pairs in the data came from the dependent variables, that is, they followed the same variable or over time or place. It formed the basis for hypothesis testing of non-parametric statistics used for population data without numerical values.

Nonparametric distributions have no parameters and cannot be described by an equation like parametric distributions. It assumes that it comes from two matching or dependent populations.

The data is also assumed to be continuous rather than discrete. Since it is a non-parametric test, it does not require a certain probability distribution of the dependent variable in the analysis.

Wilcon Signed-Rank Distribution

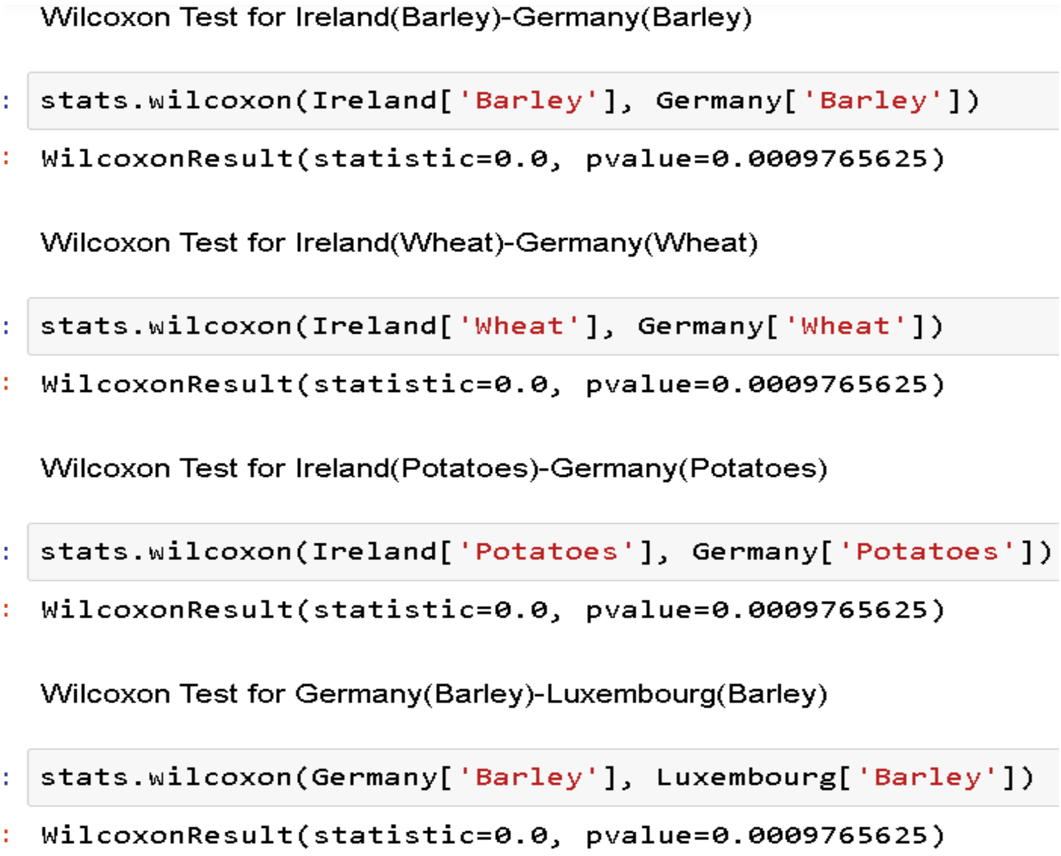
The formula above is derived and adapted from a great site: https://www.real-statistics.com/non-parametric-tests/wilcoxon-signed-ranks-test/wilcoxon-signed-ranks-exact-test/, and a more formal source can be McCornack (1965, p. 864).

The Distribution in Steps
If we have n datapoints, the ranks can go from 1 to n . Some of these were positive differences, some were negative. In total there are $2n$2n different ways these ranks could be split between the positive and negative differences.

In each of these possible splits, we can then sum the ranks for the positive and the ranks for the negative differences separately.

To get all the different possible permutations that we can have of each of the ranks distribution, we can use a 0 for a negative difference, and 1 for a positive. Then we can use Python's *itertools* library, and use the **product** function.Now for a small function to create all those possible combinations:
We only have 0 and 1 as input, and *n* possible options:

So a (0, 0, 0, 1, 0, 1) would indicate that ranks 1, 2, 3, and 5 were from a negative difference, and ranks 4 and 6 to a positive difference.
This should give the $2n$2n different permutations:



Wilcoxon Test for Ireland(Barley)-Germany(Barley)

```
stats.wilcoxon(Ireland['Barley'], Germany['Barley'])
```

```
WilcoxonResult(statistic=0.0, pvalue=0.0009765625)
```

Wilcoxon Test for Ireland(Wheat)-Germany(Wheat)

```
stats.wilcoxon(Ireland['Wheat'], Germany['Wheat'])
```

```
WilcoxonResult(statistic=0.0, pvalue=0.0009765625)
```

Wilcoxon Test for Ireland(Potatoes)-Germany(Potatoes)

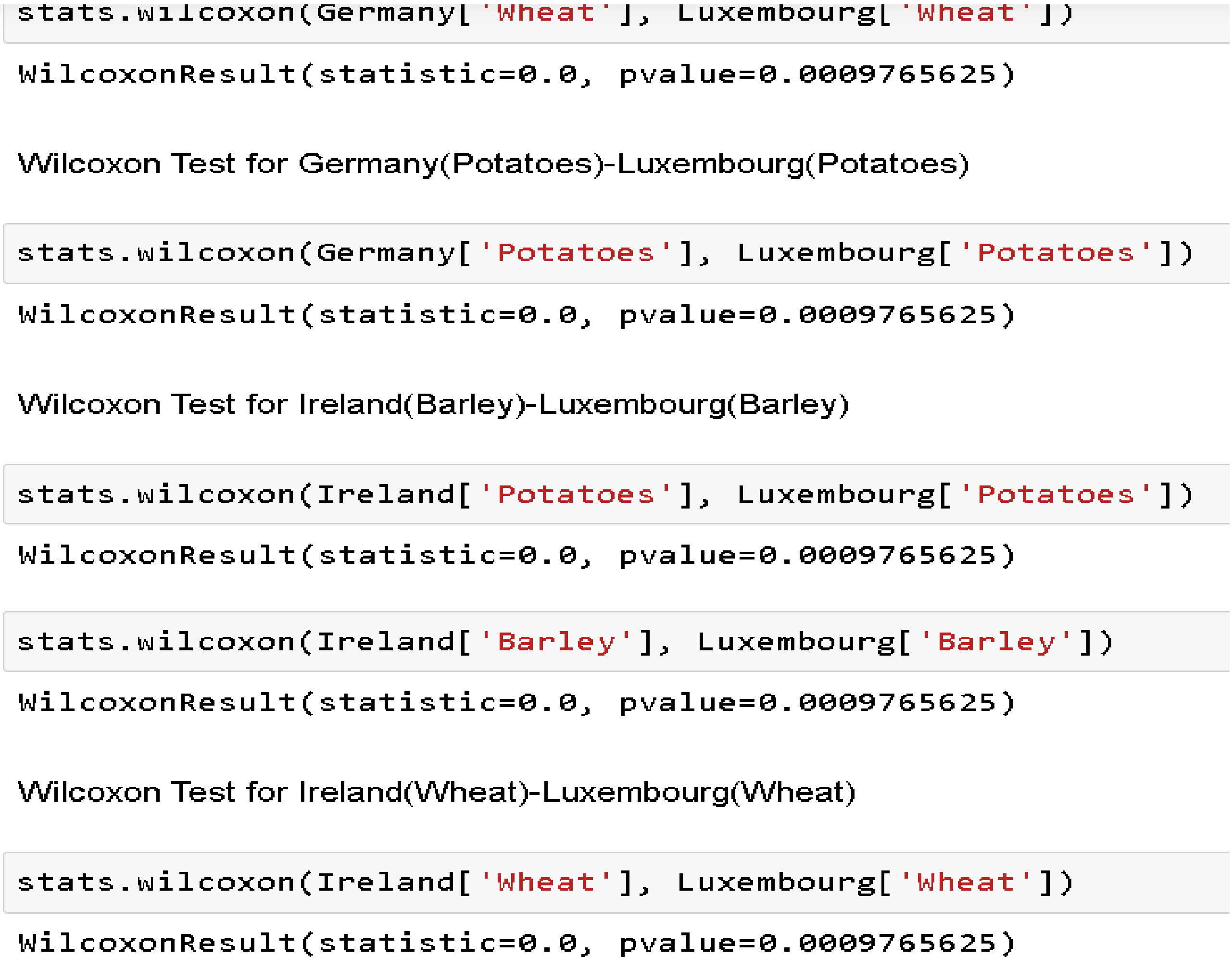
```
stats.wilcoxon(Ireland['Potatoes'], Germany['Potatoes'])
```

```
WilcoxonResult(statistic=0.0, pvalue=0.0009765625)
```

Wilcoxon Test for Germany(Barley)-Luxembourg(Barley)

```
stats.wilcoxon(Germany['Barley'], Luxembourg['Barley'])
```

```
WilcoxonResult(statistic=0.0, pvalue=0.0009765625)
```



```
stats.wilcoxon(Germany['Wheat'], Luxembourg['Wheat'])
```

```
WilcoxonResult(statistic=0.0, pvalue=0.0009765625)
```

Wilcoxon Test for Germany(Potatoes)-Luxembourg(Potatoes)

```
stats.wilcoxon(Germany['Potatoes'], Luxembourg['Potatoes'])
```

```
WilcoxonResult(statistic=0.0, pvalue=0.0009765625)
```

Wilcoxon Test for Ireland(Barley)-Luxembourg(Barley)

```
stats.wilcoxon(Ireland['Potatoes'], Luxembourg['Potatoes'])
```

```
WilcoxonResult(statistic=0.0, pvalue=0.0009765625)
```

```
stats.wilcoxon(Ireland['Barley'], Luxembourg['Barley'])
```

```
WilcoxonResult(statistic=0.0, pvalue=0.0009765625)
```

Wilcoxon Test for Ireland(Wheat)-Luxembourg(Wheat)

```
stats.wilcoxon(Ireland['Wheat'], Luxembourg['Wheat'])
```

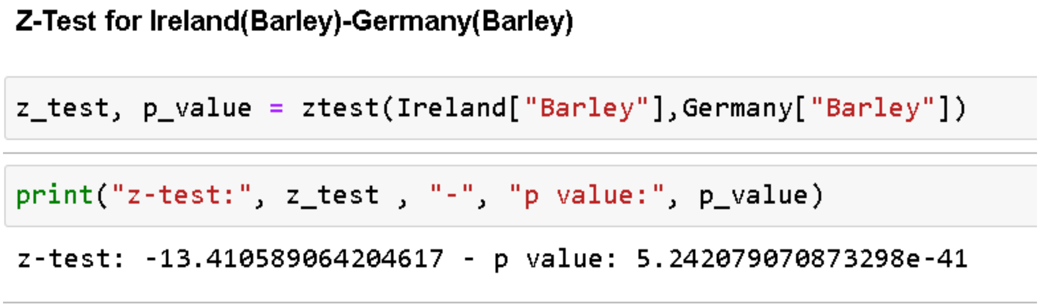
```
WilcoxonResult(statistic=0.0, pvalue=0.0009765625)
```

Z-Test

A z-score or z-statistic is a number that represents how many standard deviations the score from a z-test is above or below the mean population, and I applied this test on my dataset ka. I have found that there are some deviations. Essentially, I got a numerical measure that describes the relationship of a value to the mean of a set of values. A z-score of 0 indicates that the data point's score is the same as the mean score. I did not get such a result. A z-score of 1.0 indicates a value that is one standard deviation from the mean.

Z-scores can be positive or negative, but I've gotten positive numbers; A positive value indicates that the score is above average and a negative score indicates that it is below average. The dataset assumes that all samples are the same in size, indicating that the sample distribution approaches a normal distribution (also known as a "bell curve") as the sample size increases. and regardless of the population distribution pattern. Sample sizes greater than or equal to 30 are considered sufficient for the CLT to accurately estimate the characteristics of a population.

The fidelity of the z-test is dependent on CLT retention. Underline A z-test is used in hypothesis testing to evaluate whether a finding or relationship is statistically significant. Specifically, it tests whether two vehicles are the same (the null hypothesis). A z-test can only be used if the population standard deviation is known and the sample size is 30 data points or larger. Otherwise, a t-test should be used.

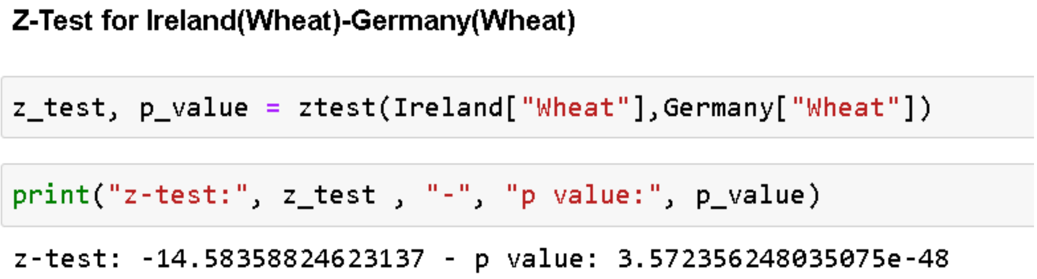


Z-Test for Ireland(Barley)-Germany(Barley)

```
z_test, p_value = ztest(Ireland["Barley"],Germany["Barley"])
```

```
print("z-test:", z_test , "-", "p value:", p_value)
```

```
z-test: -13.410589064204617 - p value: 5.242079070873298e-41
```

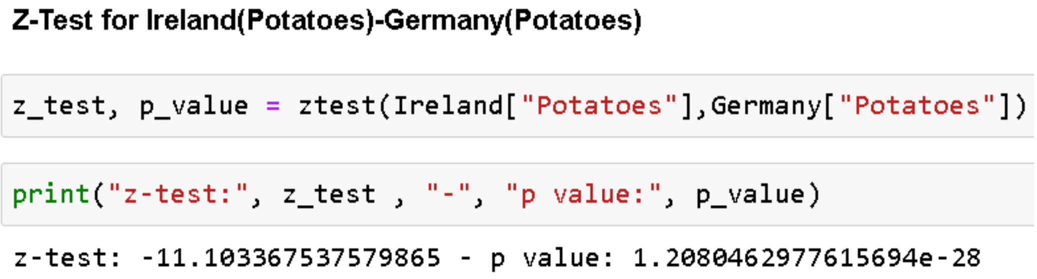


Z-Test for Ireland(Wheat)-Germany(Wheat)

```
z_test, p_value = ztest(Ireland["Wheat"],Germany["Wheat"])
```

```
print("z-test:", z_test , "-", "p value:", p_value)
```

```
z-test: -14.58358824623137 - p value: 3.572356248035075e-48
```

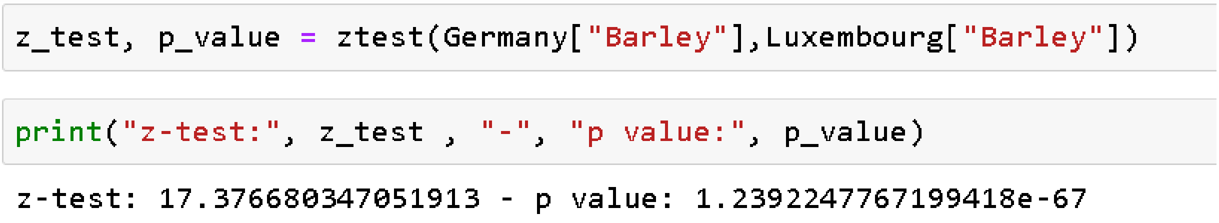


Z-Test for Ireland(Potatoes)-Germany(Potatoes)

```
z_test, p_value = ztest(Ireland["Potatoes"],Germany["Potatoes"])
```

```
print("z-test:", z_test , "-", "p value:", p_value)
```

```
z-test: -11.103367537579865 - p value: 1.2080462977615694e-28
```

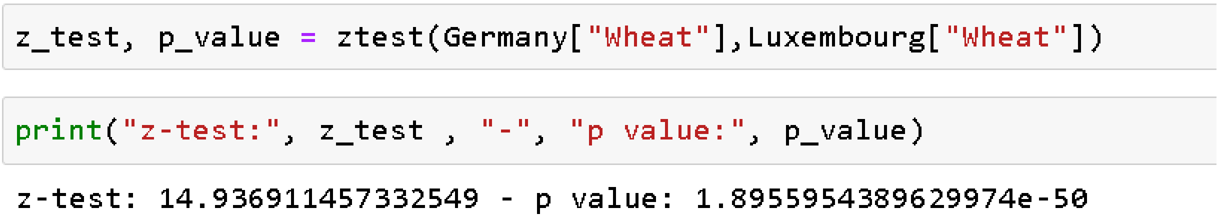


```
z_test, p_value = ztest(Germany["Barley"],Luxembourg["Barley"])
```

```
print("z-test:", z_test , "-", "p value:", p_value)
```

```
z-test: 17.376680347051913 - p value: 1.2392247767199418e-67
```

Z-Test for Germany(Wheat)-Luxembourg(Wheat)

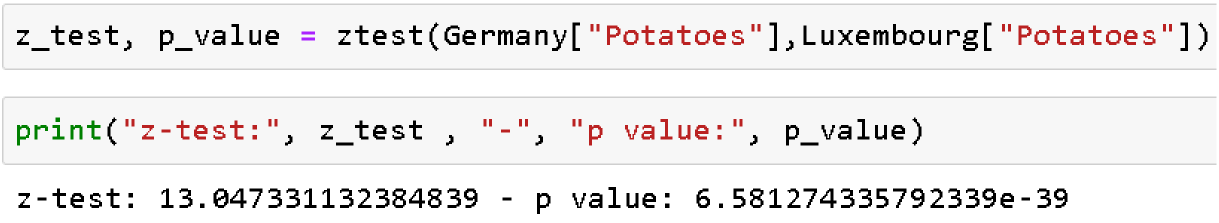


```
z_test, p_value = ztest(Germany["Wheat"],Luxembourg["Wheat"])
```

```
print("z-test:", z_test , "-", "p value:", p_value)
```

```
z-test: 14.936911457332549 - p value: 1.8955954389629974e-50
```

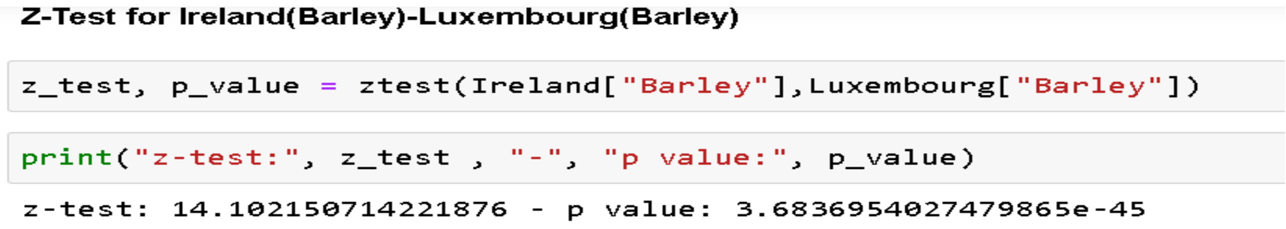
Z-Test for Germany(Potatoes)-Luxembourg(Potatoes)



```
z_test, p_value = ztest(Germany["Potatoes"],Luxembourg["Potatoes"])
```

```
print("z-test:", z_test , "-", "p value:", p_value)
```

```
z-test: 13.047331132384839 - p value: 6.581274335792339e-39
```

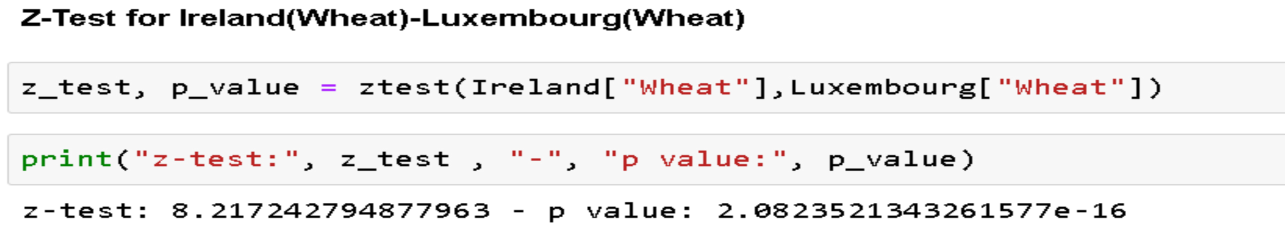


Z-Test for Ireland(Barley)-Luxembourg(Barley)

```
z_test, p_value = ztest(Ireland["Barley"],Luxembourg["Barley"])
```

```
print("z-test:", z_test , "-", "p value:", p_value)
```

```
z-test: 14.102150714221876 - p value: 3.6836954027479865e-45
```

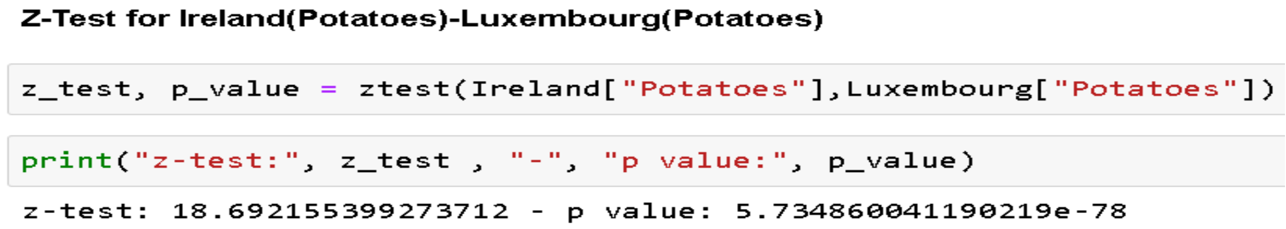


Z-Test for Ireland(Wheat)-Luxembourg(Wheat)

```
z_test, p_value = ztest(Ireland["Wheat"],Luxembourg["Wheat"])
```

```
print("z-test:", z_test , "-", "p value:", p_value)
```

```
z-test: 8.217242794877963 - p value: 2.0823521343261577e-16
```



Z-Test for Ireland(Potatoes)-Luxembourg(Potatoes)

```
z_test, p_value = ztest(Ireland["Potatoes"],Luxembourg["Potatoes"])
```

```
print("z-test:", z_test , "-", "p value:", p_value)
```

```
z-test: 18.692155399273712 - p value: 5.734860041190219e-78
```

Chi Square Test

ANOVA

I wanted to test a particular hypothesis on my dataset. As a marketer, I thought I could use Analysis of Variance (ANOVA). To help me understand how your different groups responded, I used ANOVA with a null hypothesis to test that different groups' means were equal. If there is a statistically significant result, it means that the two populations are not equal (or different). Like other types of statistical tests, in ANOVA I compared my dataset with the means of different groups and showed whether there were any statistical differences between the means. ANOVA is classified as a multi-objective test statistic. This means it can't tell you which particular groups are statistically significantly different from each other, it can only tell you that at least two groups are. I have tested whether the main ANOVA research question is whether the sampling means are from different populations.

Anova for Ireland-Germany (Barley,Barley)

```
Statistics=179.844, p=0.000
There is a significant difference between the median values (H0 is rejected)
```

Anova for Ireland-Germany (Wheat,Wheat)

```
Statistics=212.681, p=0.000
There is a significant difference between the median values (H0 is rejected)
```

Anova for Ireland-Germany (Potatoes,Potatoes)

```
Statistics=123.285, p=0.000
There is a significant difference between the median values (H0 is rejected)
```

Anova for Ireland-Luxembourg (Barley,Barley)

```
Statistics=198.871, p=0.000
There is a significant difference between the median values (H0 is rejected)
```

Anova for Ireland-Luxembourg (Wheat,Wheat)

```
Statistics=67.523, p=0.000
There is a significant difference between the median values (H0 is rejected)
```

Anova for Ireland-Luxembourg (Potatoes,Potatoes)

```
Statistics=349.397, p=0.000
There is a significant difference between the median values (H0 is rejected)
```

Anova for Germany-Luxembourg(Barley,Barley)

```
Statistics=301.949, p=0.000
There is a significant difference between the median values (H0 is rejected)
```

Anova for Germany-Luxembourg (Wheat,Wheat)

```
Statistics=223.111, p=0.000
There is a significant difference between the median values (H0 is rejected)
```

Anova for Germany-Luxembourg(Potatoes,Potatoes)

```
Statistics=170.233, p=0.000
There is a significant difference between the median values (H0 is rejected)
```

Decision Tree

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome.

The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.
The output is meaningful, but looks like absolute garbage. Luckily, we can make it beautiful with a *heatmap* from the **Seaborn** library.

This is cool too. I even changed the color to something more appealing with the cmap attribute… but what if I want to see both count and percentage at once? What if I want to see a label also? Luckily the seaborn heatmap has the ability to accept text labels for the *annot* field.

o… what if I put it all in one function and include options to show or not show certain parameters, and also go through some other marine-based options like showing a colormap or a colorbar? What if I add some summary stats like Accuracy, Precision, Recall and F-Score to the view? That would be incredibly convenient. With these considerations in my dataset, I created a function that does just that.
Use the confusion_matrix method from sklearn.metrics to compute the confusion matrix
The result is an array in which positions are the same as the quadrant we saw in the past
With data from the confusion matrix, you can interpret the results by looking at the classification report.
I used the confusion matrix to evaluate the performance of a machine learning classification algorithm. Is confusion matrix better than accuracy? The confusion matrix provides more information about a model's performance than classification accuracy, as it shows the number of correctly and incorrectly classified samples. Confusion matrices show the accuracy of the estimation of the classes. When trying to predict a number output, as in the case of a continuous output of a regression model, I used a confusion matrix. I've used the confusion matrix to evaluate the accuracy of a machine learning model (e.g. Classification) that tries to predict classes.

Linear Regression

Linear regression can be thought of as finding the straight line that best fits a set of scattered data points. So I started implementing it on my own dataset to find this line.

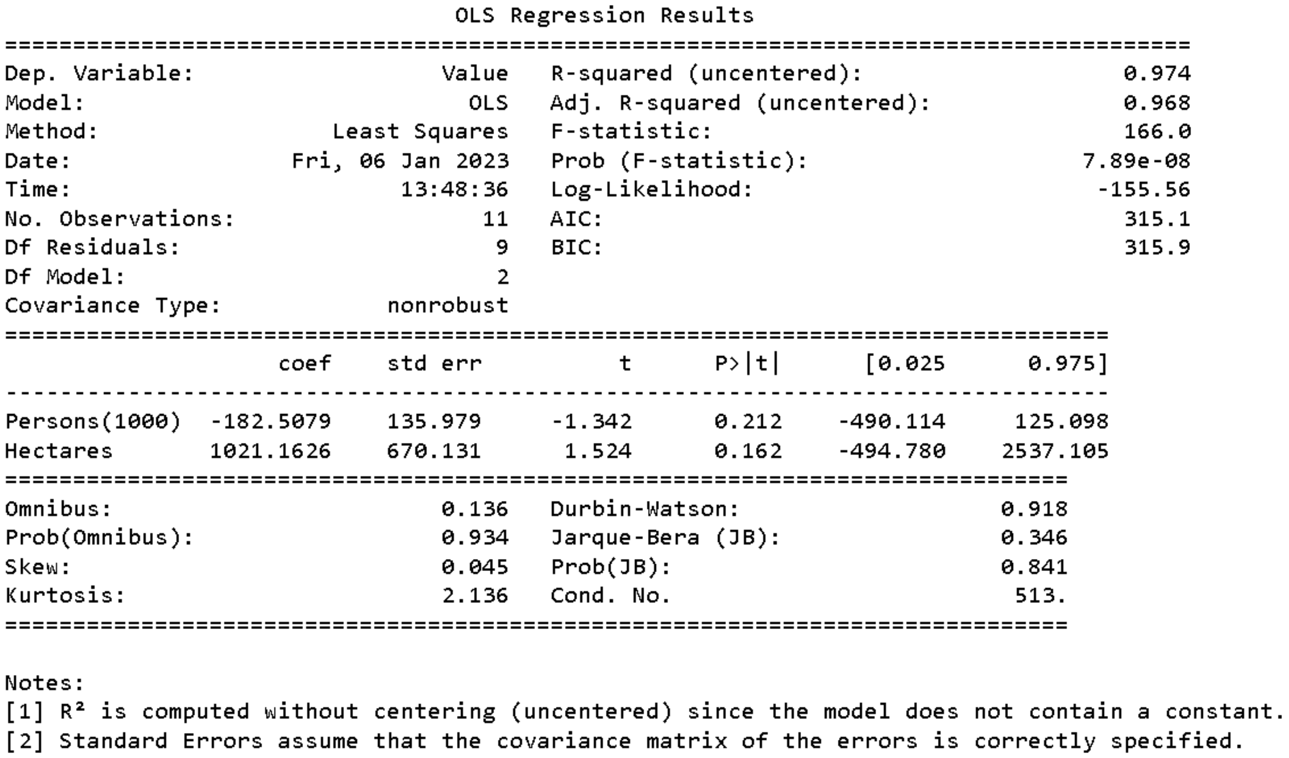
You can then project this line to predict new data points. Linear regression is a fundamental machine learning algorithm because of its relatively simple and essential features. Linear Regression Concepts A basic understanding of statistical mathematics is key to understanding linear regression and provides a good foundation in machine learning concepts.

Linear Regression Class Definition

A scikit-learn linear regression script begins by importing the *LinearRegression* class:
Although the class is not visible in the script, it contains default parameters that do the heavy lifting for simple least squares linear regression:
Calculate the intercept for the model. If set to False, no intercept will be used in the calculation.

Regression looks for relationships between variables. In general, in regression analysis, I considered some of the facts of your interest and made a number of observations. Each observation has two or more properties. Based on the assumption that at least one of the features is dependent on the others, I tried to establish a relationship between them. In other words, I needed to find a function that maps some properties or variables to others well enough.

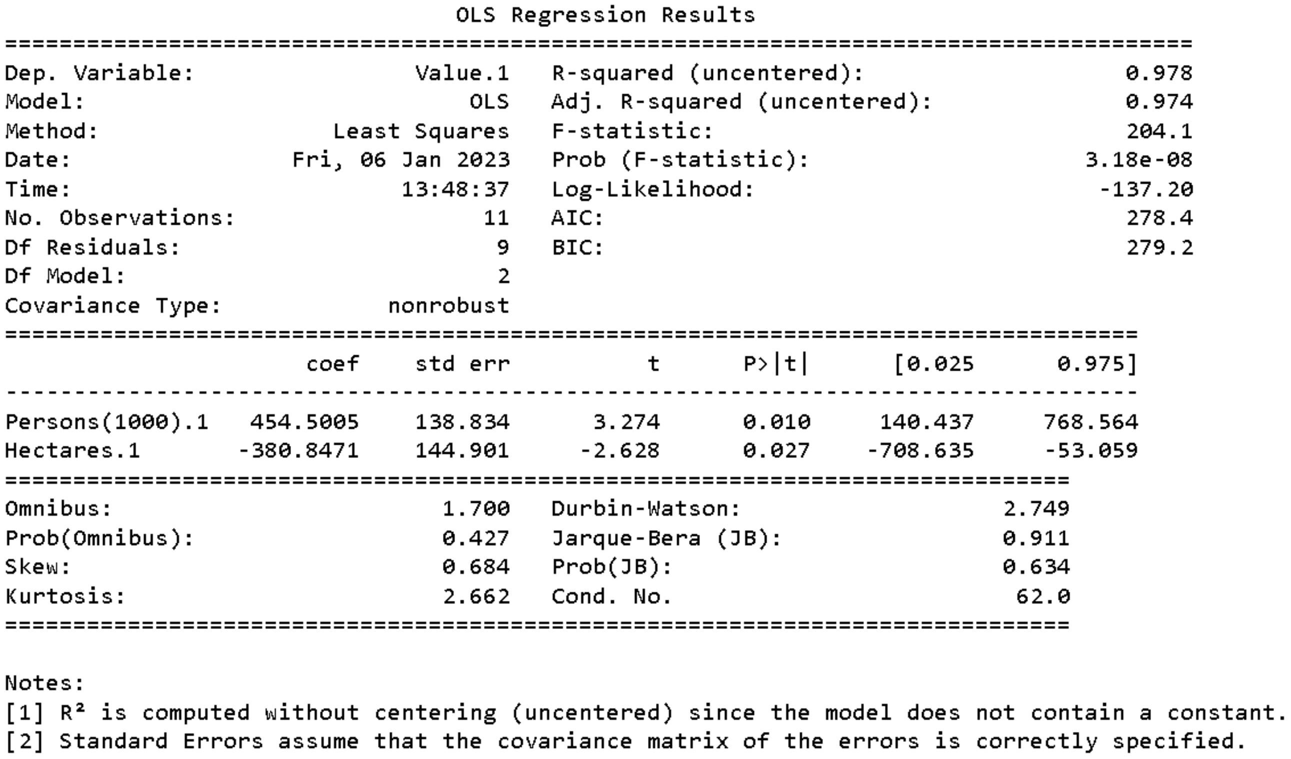
Linear Regression for Germany



```
                            OLS Regression Results
=======================================================================================
Dep. Variable:                  Value   R-squared (uncentered):                   0.974
Model:                            OLS   Adj. R-squared (uncentered):              0.968
Method:                 Least Squares   F-statistic:                              166.0
Date:                Fri, 06 Jan 2023   Prob (F-statistic):                    7.89e-08
Time:                        13:48:36   Log-Likelihood:                         -155.56
No. Observations:                  11   AIC:                                      315.1
Df Residuals:                       9   BIC:                                      315.9
Df Model:                           2
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Persons(1000)  -182.5079    135.979     -1.342      0.212    -490.114     125.098
Hectares       1021.1626    670.131      1.524      0.162    -494.780    2537.105
==============================================================================
Omnibus:                        0.136   Durbin-Watson:                   0.918
Prob(Omnibus):                  0.934   Jarque-Bera (JB):                0.346
Skew:                           0.045   Prob(JB):                        0.841
Kurtosis:                       2.136   Cond. No.                         513.
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

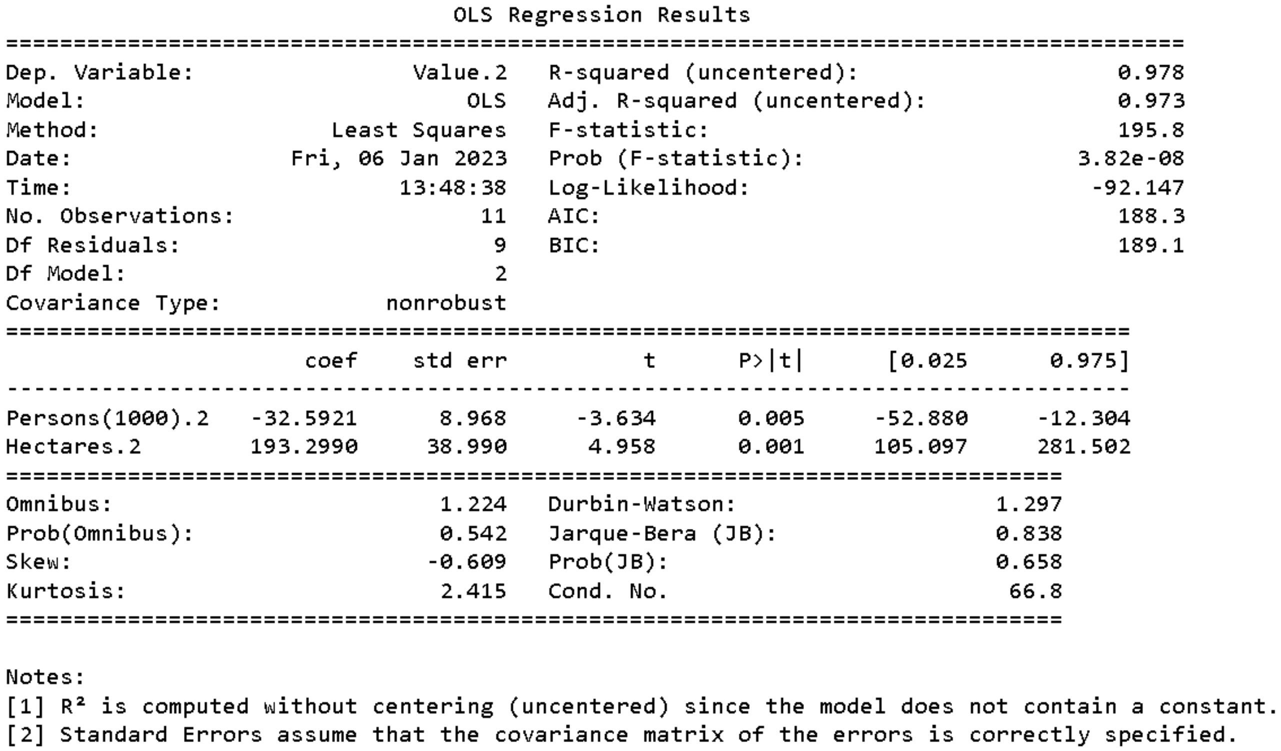
Linear Regresion for Ireland



```
                            OLS Regression Results
=======================================================================================
Dep. Variable:                Value.1   R-squared (uncentered):                   0.978
Model:                            OLS   Adj. R-squared (uncentered):              0.974
Method:                 Least Squares   F-statistic:                              204.1
Date:                Fri, 06 Jan 2023   Prob (F-statistic):                    3.18e-08
Time:                        13:48:37   Log-Likelihood:                         -137.20
No. Observations:                  11   AIC:                                      278.4
Df Residuals:                       9   BIC:                                      279.2
Df Model:                           2
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Persons(1000).1   454.5005    138.834      3.274      0.010     140.437     768.564
Hectares.1       -380.8471    144.901     -2.628      0.027    -708.635     -53.059
==============================================================================
Omnibus:                        1.700   Durbin-Watson:                   2.749
Prob(Omnibus):                  0.427   Jarque-Bera (JB):                0.911
Skew:                           0.684   Prob(JB):                        0.634
Kurtosis:                       2.662   Cond. No.                         62.0
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Linear Regresion for Luxembourg



```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:                Value.2   R-squared (uncentered):                   0.978
Model:                            OLS   Adj. R-squared (uncentered):              0.973
Method:                 Least Squares   F-statistic:                              195.8
Date:                Fri, 06 Jan 2023   Prob (F-statistic):                    3.82e-08
Time:                        13:48:38   Log-Likelihood:                         -92.147
No. Observations:                  11   AIC:                                      188.3
Df Residuals:                       9   BIC:                                      189.1
Df Model:                           2
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Persons(1000).2   -32.5921      8.968     -3.634      0.005     -52.880     -12.304
Hectares.2       193.2990     38.990      4.958      0.001     105.097     281.502
==============================================================================
Omnibus:                        1.224   Durbin-Watson:                   1.297
Prob(Omnibus):                  0.542   Jarque-Bera (JB):                0.838
Skew:                          -0.609   Prob(JB):                        0.658
Kurtosis:                       2.415   Cond. No.                         66.8
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Summary.

5 datasets were created and I got these datasets from the FAO data site. This data set is located in Germany, Luxembourg, Ireland and includes Barley, Wheat and Potato productions. I tried to establish a contradiction between these products and examined them on a product basis. I also looked at the production amounts between countries. I applied Algorithms and Tests between these data sets. FAOSTAT data I created a data set for these countries between the years 2010-2020 as Potato, Wheat, Barley for Agriculture. and in addition to this, I created a data set of the total area of hectares allocated by these countries for the cultivation of Potato, Barley, Wheat. I combined these 5 datasets under a single dataset and analyzed them in a single way. For this dataset I created, I wrote algorithms and applied tests. In this way, I found out how much labor force all these European countries have achieved to grow these products. I checked how successful they were. The reason I chose Luxembourg is that it is the 2nd country with the least production in the European Union. Malta produces the least, but I could not create a clear enough data set about Malta. I couldn't find a reliable data. That's why I chose Luxembourg. Ultimately, I chose Ireland. And I examined the production amounts of these countries for the last 10 years. I compared the amount of Irish production in Bolece. I found the position of Ireland in growing these products and I compared them.

Likewise, I compared Luxembourg and Germany. As a result, I have seen more clearly how many hectares of land and how much labor force they have grown these Potato, Wheat, Barley products.