International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Method Comparison of Two Electrolyte Testing Devices: A Data Analytics based on Python Approach

Dr P. Ragavendran

Manager Lab Operation

Abstract

In this paper, analyze the statistical data analysis using Python. In Phyton, user can able to run multiple analyses at the single time and data visualization, analytics are more productive. In this paper, 10 data's of sodium perform in two different devices and check correlation, t-test, chi-square test, oneway ANOVA in Python. Different proposed structures of Data Science and address the impact of statistics and data analysis.

Keywords: Data Visualization, Python, Correlation, t-Test, Chi-Square test

1. Introduction

Data analysis is the process of cleaning, transforming, and analyzing raw data to obtain usable, relevant information that can assist businesses in making educated decisions. By giving relevant insights and data, which are commonly presented in charts, photos, tables, and graphs, the technique helps to lessen the risks associated with decision-making [1].

Data Analytics are commonly performing in Excel. When compared with Excel, other statistical software's such as Phyton and R has more advanced techniques than Excel. For Example. Data analysis in Excel need to add on Analysis tool pack and perform the data analytics tools what it listed [2]. End user amend any data points, need to run these respective tools again and data visualization is limited in MS Excel. Excel provides a range of built-in statistical functions, but it may not offer the breadth and depth of statistical analysis capabilities available in dedicated statistical software or programming languages like Python or R [3].

Python is a general-purpose programming language that provides more flexibility and scalability compared to MS Excel. It can handle large datasets and perform complex calculations more efficiently. Python allows you to automate repetitive tasks, build custom workflows, and integrate with other tools and libraries for advanced data analysis. Python libraries such as Pandas and NumPy provide powerful tools for data manipulation and transformation. These libraries offer efficient data structures, such as DataFrames, that allow you to easily filter, aggregate, reshape, and join datasets. Python also supports regular expressions for complex text processing tasks, where in Excel primarily designed for manual data entry and analysis, lacking robust automation capabilities. It can be challenging to automate repetitive tasks or build complex data analysis workflows. Excel also has limitations in terms of reproducibility, making it harder to document and share analysis steps or ensures consistent results across different users or versions of the spreadsheet [4].

In this article, describe the data analysis and visualization of Electrolyte data in 2 testing devices in Python with various statistical tools.

2. Methods

2.1 Experimental Design

10 samples are performed in two different devices for Sodium analysis.
Total datasets is 20 (10 rows x 2 columns). Datas are listed below table.,

2.2 Data Analytics tools

Results obtained two difference device are calculated in

- Correlation
- t-test
- Chi square test
- One Way ANOVA

using Python in Google Colab.

3. Results and Discussion

Datas performed in two different devices has Microsoft excel format and this sheet uploaded into Google Colab.
Data Analysis in Google Colab explained in following steps.,

3.1 Create Python Notebook

Open Google chrome and then open Google Colab (https://colab.research.google.com)
Click New Notebook

3.2 Importing Library

Once create new notebook, and import library as pandas and numpy.

```
[ ]  import pandas as pd
     import numpy as np
```

Fig 1 – Import of Numpy and Pandas Library

Numpy and Pandas widely used for Data sciences, Machine Learning and Scientific Computing and they complement each other in many application. Numpy is used for numerical computing, Core Functionality, Array Operation, Integration with Low Level Launguages. Pandas is used for Data structure, alignment, cleaning and analysis [5].

3.3 Importing Dataset

Dataset in Excel / CSV / text form need to upload in Python form analysis.

Fig – 2 : Read data – Upload Data into Excel file.

3.4 Data Analysis

Purpose is to check the correlation of two devices performed in same sample. 10 rows and 2 columns selected and total data is 20. Data cleaning not required for this small size sample analysis. If large size data, data cleaning and outlier check need to execute.

In this analysis, following parameters are listed to check the effectiveness of correlation of two devices.

3.4.1 Correlation

In Correlation analysis in Python, 'seaborn' Library need to import for graphical presentation such as correlation graph, distribution plot.

Install seaborn library in Python as !pip install seaborn.

Seaborn is widely used in data analysis, exploratory data visualization, and generating publication-quality graphics. It complements the functionality of other data analysis libraries in the Python ecosystem, such as NumPy, Pandas, and SciPy. By leveraging Seaborn's capabilities, you can create visually engaging and informative plots to effectively communicate your data analysis results.

Then correlation plot and distribution plot performed for this dataset.

- Correlation Graph

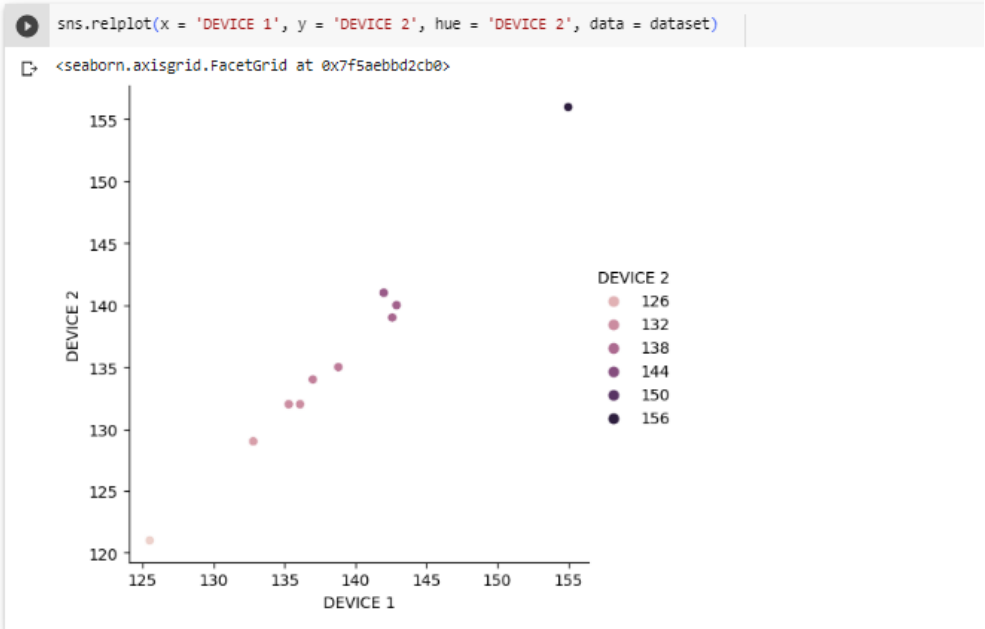

Fig 3 : Correlation co-efficient - Scatter Graph

Distribution Plot

```
sns.displot(dataset, x = 'DEVICE 1', y = 'DEVICE 2', legend = 'true')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f5b1f5a3eb0>
```

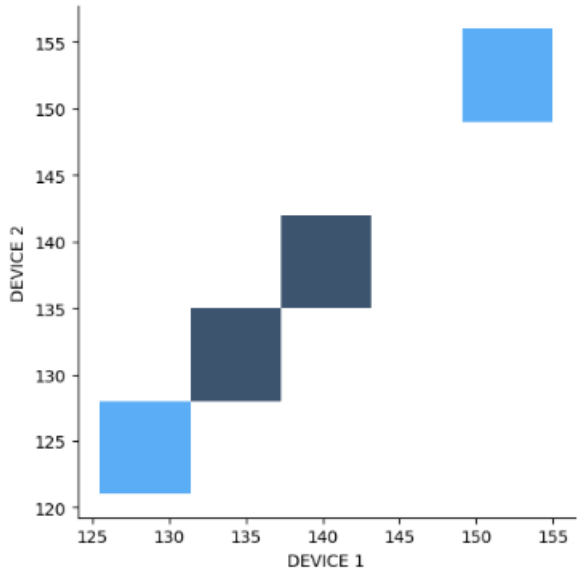

Fig 4 : Correlation Graph – Distribution Plot

Correlation

```
sns.regplot(x = 'DEVICE 1', y = 'DEVICE 2', data = dataset)
```

```
<Axes: xlabel='DEVICE 1', ylabel='DEVICE 2'>
```

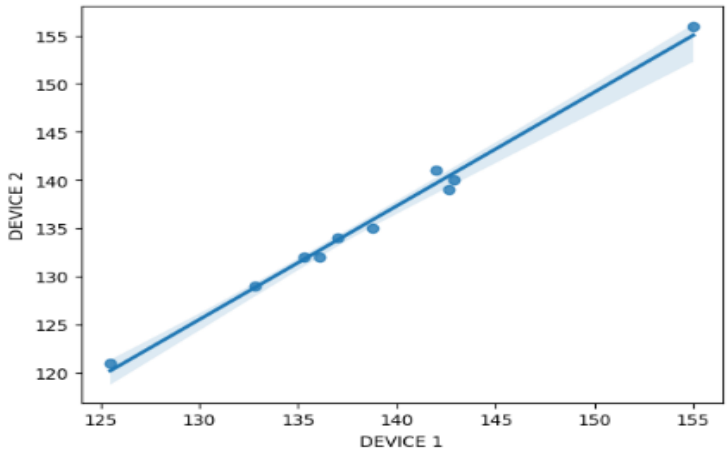

Fig 5 : Correlation – Regression Plot

Correlation calculation of two devices calculated in Python through DataFrame() function.

```
[ ]  data = dataset
     df = pd.DataFrame(data)
     print(df.corr())
```

```
            DEVICE 1  DEVICE 2
DEVICE 1    1.000000  0.995214
DEVICE 2    0.995214  1.000000
```

Fig 6 : Correlation Summary

Detailed correlation summary of slope, Intercept, r2, p-value are detailed as below image.,

```
import scipy.stats
x = dataset['DEVICE 1']
y = dataset['DEVICE 2']
result = scipy.stats.linregress(x,y)
print("SLOPE : ",result.slope)
print("Intercept :",result.intercept)
print("r Value :",result.rvalue)
print("p value :",result.pvalue)
print("Standard Error :",result.stderr)
if(result.pvalue<0.05):
  print("THERE IS NO SIGNIFICANT DIFFERENCE BETWEEN TWO DEVICES. BOTH RESULTS ARE CORRELATING")
else:
  print("THERE IS SIGNIFICANT DIFFERENCE BETWEEN TWO DEVICES")
```

```
SLOPE :  1.1818013343217202
Intercept : -28.134025203854776
r Value : 0.9952144071772019
p value : 2.2815228174075985e-09
Standard Error : 0.04102470684004908
THERE IS NO SIGNIFICANT DIFFERENCE BETWEEN TWO DEVICES. BOTH RESULTS ARE CORRELATING
```

Fig 7 : Correlation summary

In this statistical Analysis, Device 1 and 2 result correlate as 0.995214, observe > 0.9. It seems Values are equal in both the analyzers and No statistically difference between these two devices. Standard Error of these two data sets has 0.041 [6].

3.4.2 t-test

The t-test is a statistical hypothesis test used to determine if there is a significant difference between the means of two groups or samples. It is commonly employed when the sample size is small (typically, less than 30).

In t-test perform in Python, need to install pingouin Library as !pip install pingouin, import t, ttest_ind from scipy.stats [7]

```
import numpy as np
import pingouin as pg
import matplotlib.pyplot as plt
from scipy.stats import t
from scipy.stats import ttest_ind

data_group1 = dataset['DEVICE 1']
data_group2 = dataset['DEVICE 2']
result = pg.ttest(data_group1, data_group2, correction= True)
t_stat, p_val =ttest_ind(data_group1, data_group2)
print(result)

plt.hist(data_group1, alpha=0.5, label='DEVICE 1')
plt.hist(data_group2, alpha=0.5, label='DEVICE 2')
plt.legend(loc='upper right')

plt.axvline(np.mean(data_group1), color='blue', linestyle='--')
plt.axvline(np.mean(data_group2), color='red', linestyle='--')
plt.axvline(np.mean(data_group2) + t_stat*np.std(data_group2)/np.sqrt(len(data_group2)), color='red', linestyle='--')
plt.text(np.mean(data_group2) + t_stat*np.std(data_group2)/np.sqrt(len(data_group2)), 3.25, f"t = {round(t_stat, 2)}, p = {round(p_val, 2)}", rotation=90)
plt.show()
```

```
              T       dof alternative     p-val        CI95%   cohen-d  \
T-test  0.762894  17.493455   two-sided  0.455692  [-5.1, 10.9]  0.341176

         BF10     power
T-test  0.488  0.111678
```

Fig 8 : t-test Data Analysis

t-value is 0.762 interpretated that group-1 mean has higher than group-2 mean. P-value indicate that 0.455692, observe more than (>0.05), there is no statistical difference between group-1 and 2. Results are correlating [8]

t-test distribution graph detailed in below image.,

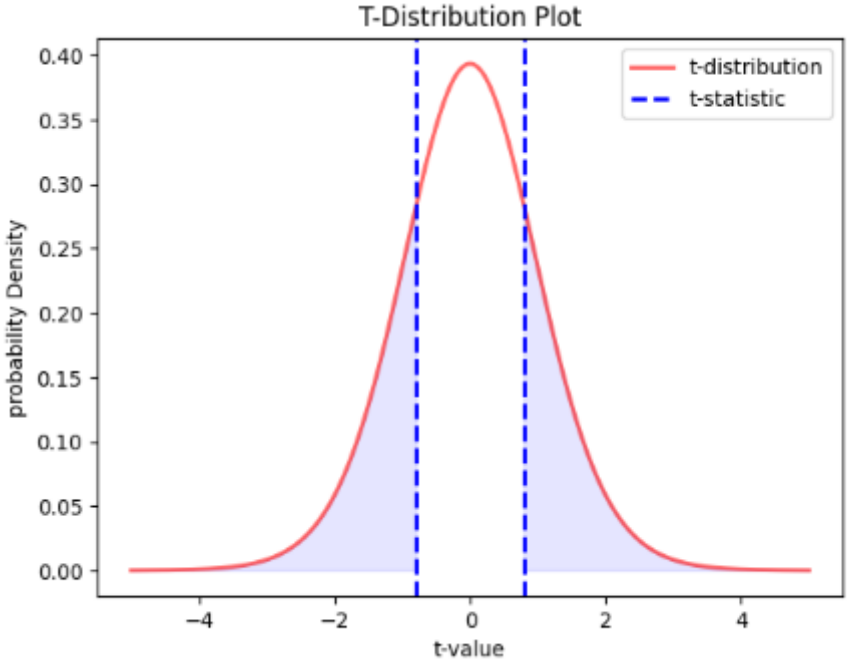

Fig 9 : t-test Distribution Graph

3.4.3 Chi Square Test

The chi-square test is a statistical hypothesis test used to determine if there is a significant association or difference between two categorical variables. It is particularly useful when dealing with categorical data and is often employed to analyze data in fields such as social sciences, market research, and biology.

Chi square details are listed as below.,

Fig 10 : Chi –Square test

As per chi-square test is 0.99999, it seems that both datasets are perfectly correlated. There is no statistical difference obtained in two datasets.

3.4.4 One Way ANOVA

One-way ANOVA (Analysis of Variance) is a statistical test used to determine if there are any statistically significant differences between the means of two or more groups. It is designed to compare the means of multiple groups simultaneously, rather than comparing pairs of groups as in t-tests. Perform one way ANOVA in Python need to import f_oneway from scipy.stats

One way ANOVA results in Python detailed as below.,

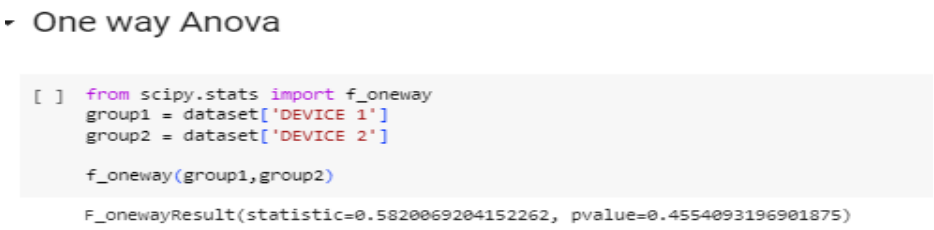

Fig 11 – Oneway ANOVA results

F-statistic of 0.5820 does not provide strong evidence to reject the null hypothesis. This suggests that there are no significant differences among the group means.

The p-value of 0.4554 is larger than the commonly used significance level of 0.05. This indicates that the observed differences in the group means are not statistically significant at the chosen significance level. In other words, there is not enough evidence to conclude that the means of the groups differ significantly from each other [9].

In summary, based on the provided results, there is no significant difference among the group means being compared in this one-way ANOVA analysis.

4. Conclusion

Same dataset performed different statistical approach in Python to get.,

- Detailed conclusion of users
- Time is less
- Data's are produce with lot of statistical and graphical presentation.

In Python, in this same code need to check different type of datasets without addition of program.

In this overall findings, Python has good Data analytics tools check data's in all aspects.

5. Conflict of Interest Statement

Authors declare that don't have no conflict of interest.

6. References

1. Elgendy N and Elragal A (2014). Conference Paper in Lecture Notes in Computer Science, page No : 217 – 227.
2. Elliott AC, Hynan LS, Reisch JS, Smith JP (2006). Preparing Data for Analysis Using Microsoft Excel. Journal of Investigative Medicine, Vol 54 No 6, Page No : 334-341.
3. McKinney W (2013). Python for Data Analysis. Book Chapter 1- Essential Python Libraries, Page No : 3-7
4. Acharjya DP, Ahmed KP (2016). A Survey on Big Data Analytics: Challenges, Open

Research Issues and Tools. International Journal of Advanced Computer Science and Applications, Vol 7, No 2, page No: 511 – 518

5. R. Wille (2005). Formal concept analysis as mathematical theory of concept and concept hierarchies, Lecture Notes in Artificial Intelligence, page No:1-33.
6. Harwalkar D, Gahana SP, Pilakkal T, Sreeganesan TG (2020). Analytical Study Of Correlation Between Demand And Renewable Energy Forecasting Using Data Mining/Analytics. Journal of Emerging Technologies and Innovative Research. Vol 7, No 10, page No:72-78.
7. Vallat R (2018). Pingouin: statistics in Python. The Journal of Open Source Software.
8. Rouder JN, Speckman PL, Sun D, Morey RD, Iverson, G (2009). Bayesian t tests for accepting and rejecting the null hypothesis. Psychon. Bull. Rev., Vol 16, No 2, 225–237.
9. McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), Proceedings of the 9th python in science conference (pp. 51–56).