**LYON SEBASTIAN WIBOWO**

**PRE-SCREENING ASSIGNMENT**

Note: Every code and visualization were run using Jupyter Notebook

Make sure python file and dataset are in the same directory

Provide both .ipynb files for Jupyter Notebook and .py files for Python

**QUESTION 1**

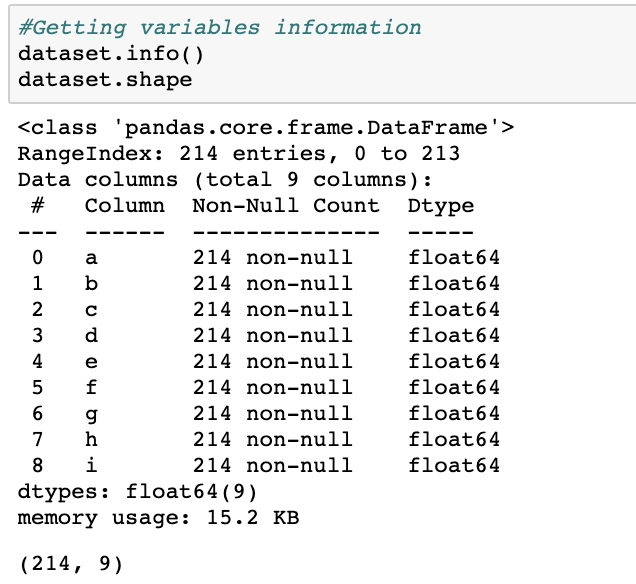
***Please assist the consultant in the area of statistical analysis by doing this;***

* A descriptive analysis of the additives (columns named as “a” to “i”), which must include summaries of findings (parametric/non-parametric).
* Correlation and ANOVA, if applicable, is a must.
* A graphical analysis of the additives, including a distribution study.
* A clustering test of your choice (unsupervised learning), to determine the distinctive number of formulations present in the dataset.



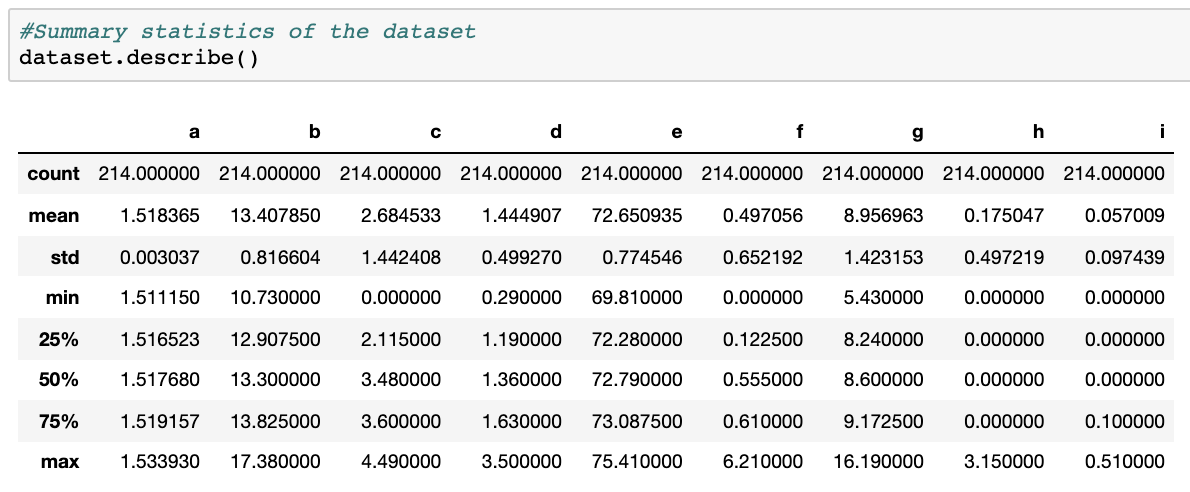
*Figure 1. Importing necessary libraries and dataset*

We start off by importing the necessary libraries that will be used in further processing, as well as import the dataset.



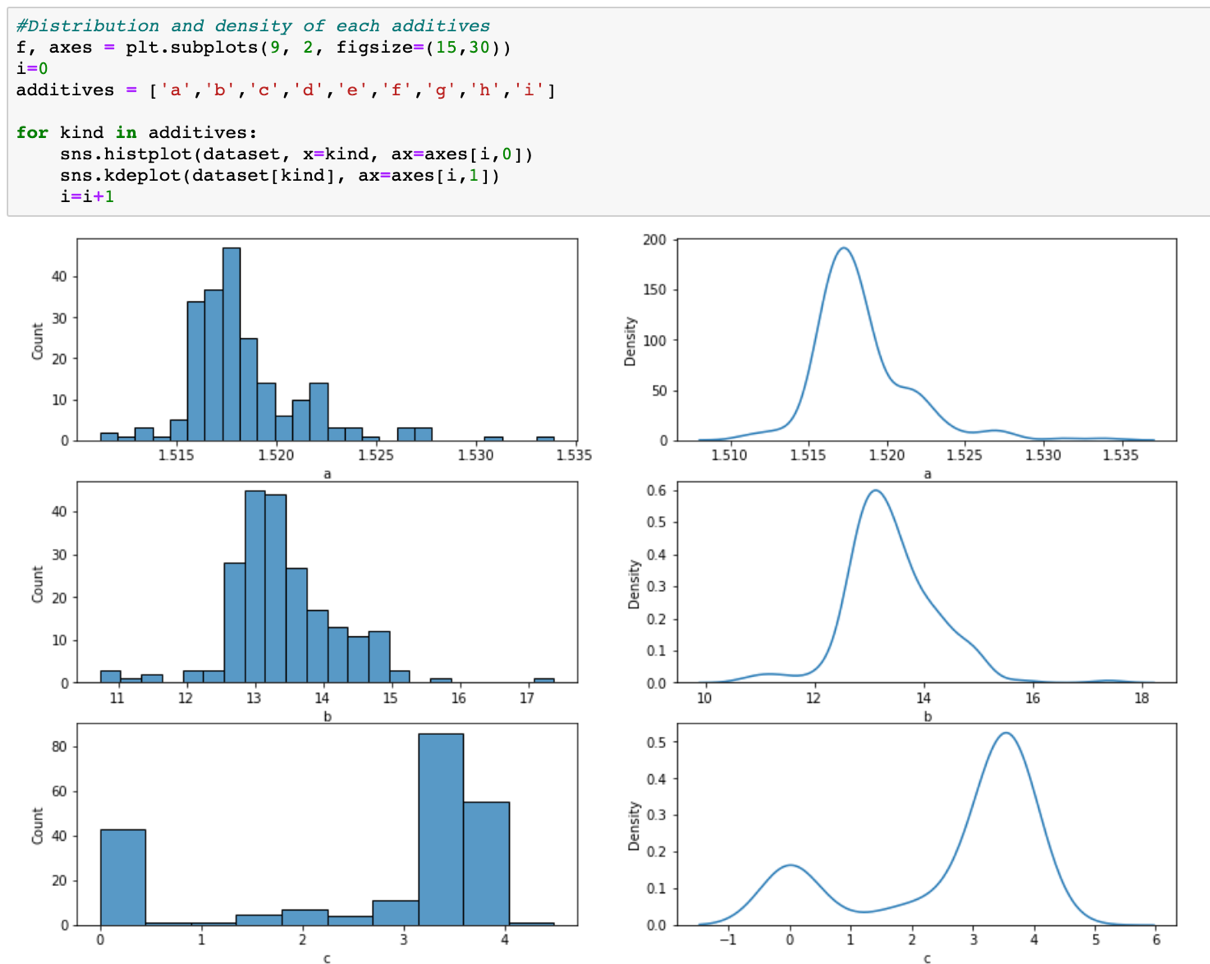
*Figure 2. Dataset variable information*

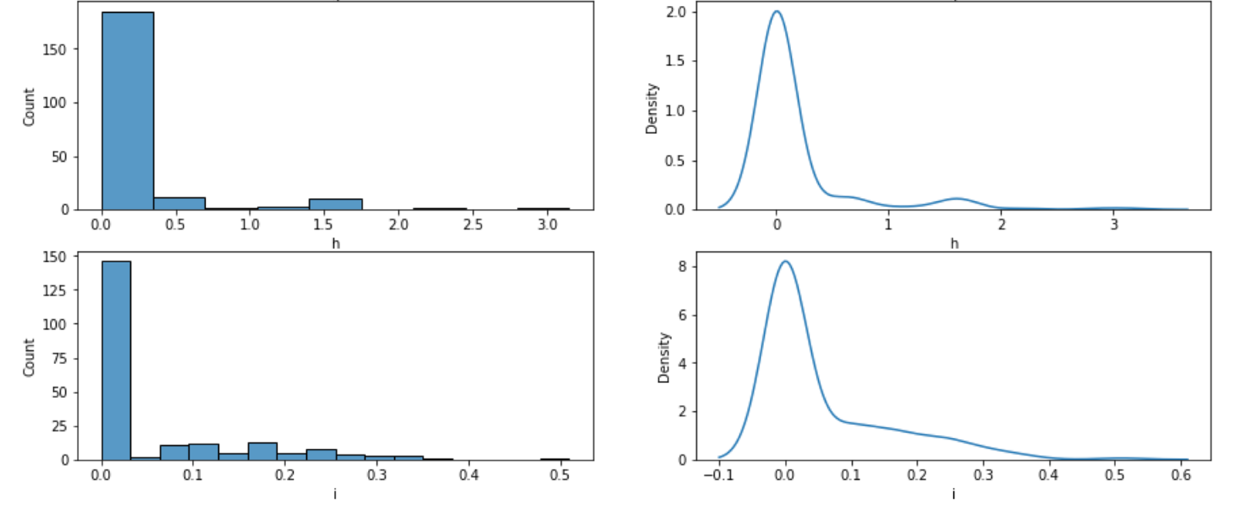
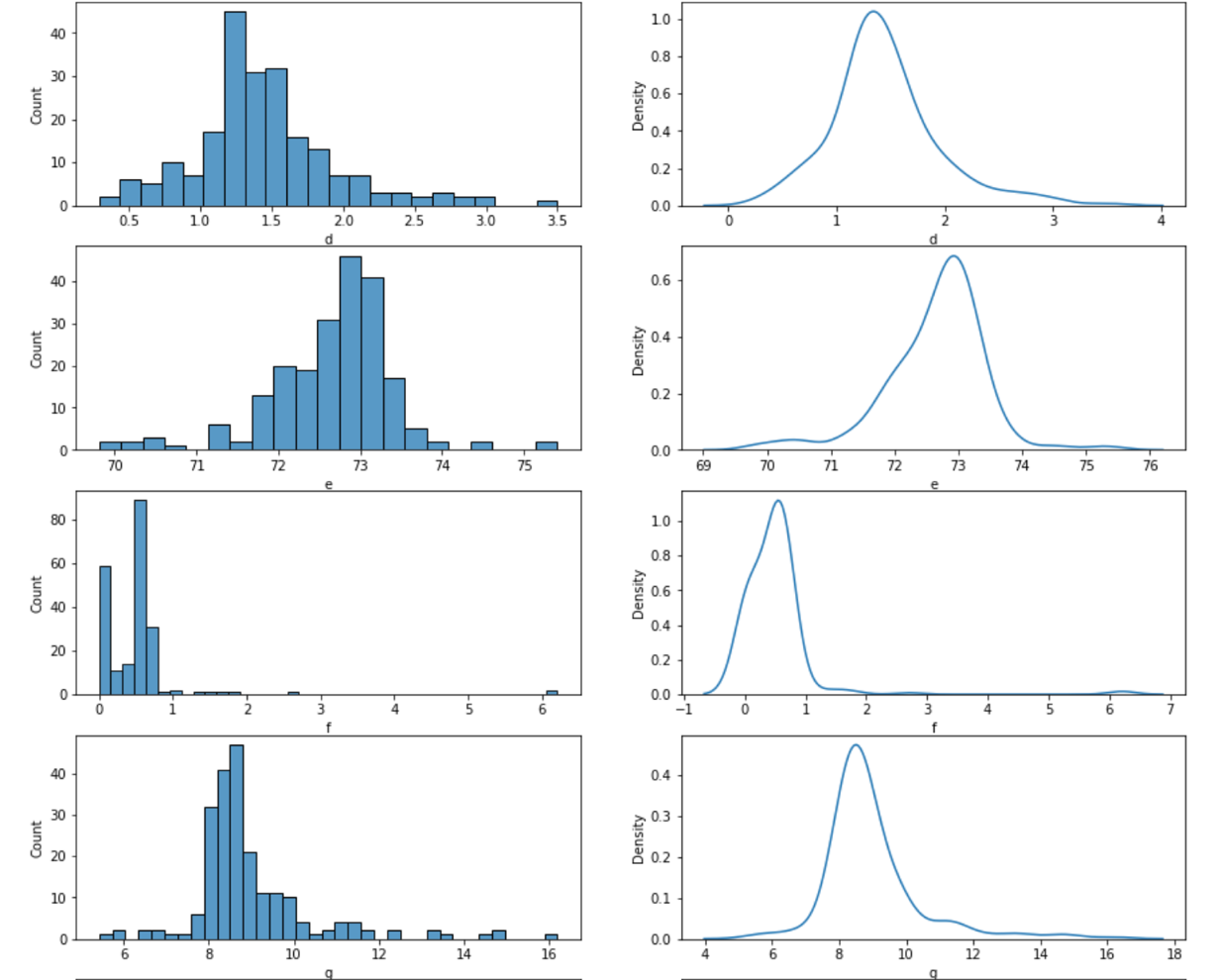
From Figure 2, we were able to retrieve a description of the imported dataset. There is a total of 214 observations/rows and 9 columns in the dataset altogether. All of the columns have a float data type and no NULL values are found.



*Figure 3. Summary statistics of dataset*

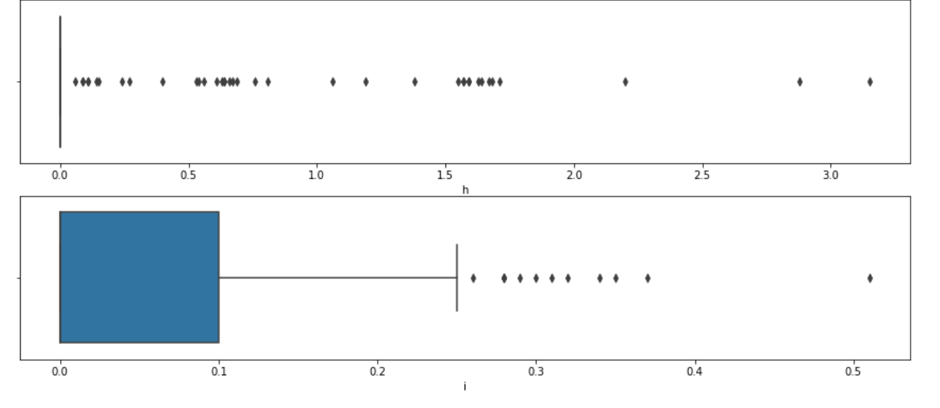
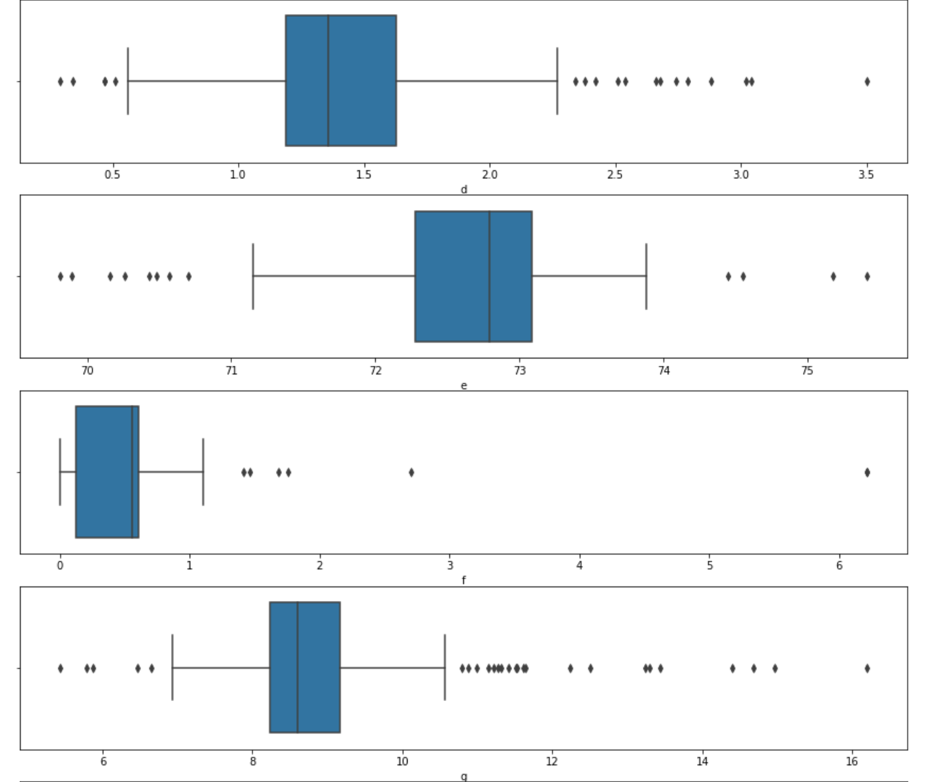
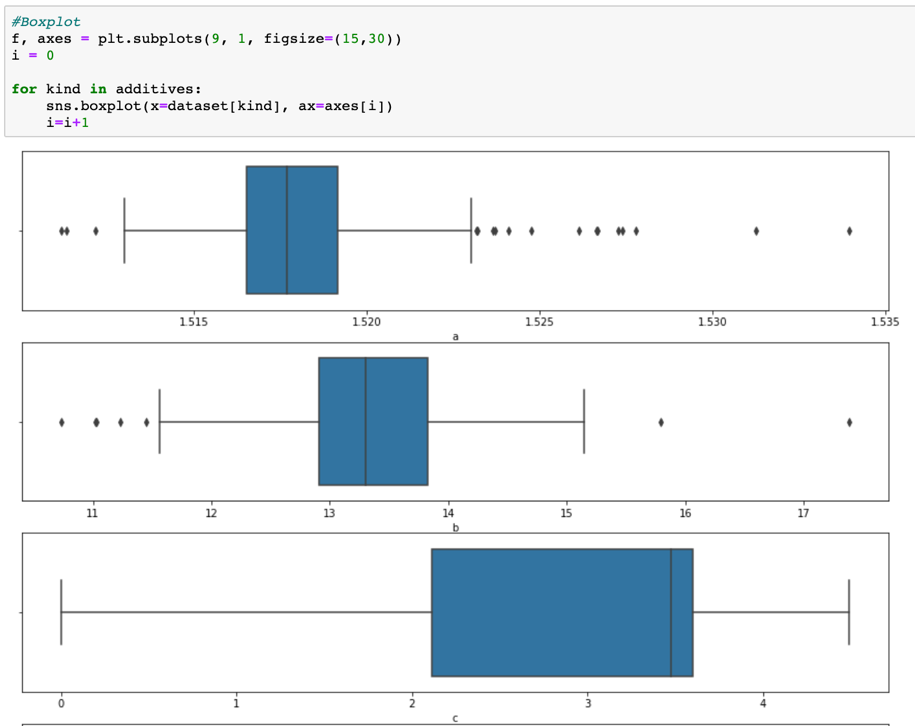
From the summary statistics in Figure 3, we know that most of the columns have mean that are very close to the median (50% percentile). The variance of data within each column are also very small, except for column c and g. Both of these findings suggest that the data in most of the columns are concentrated around the mean and have a lean curve distribution. This can be further proven through the use of histogram and density plots.





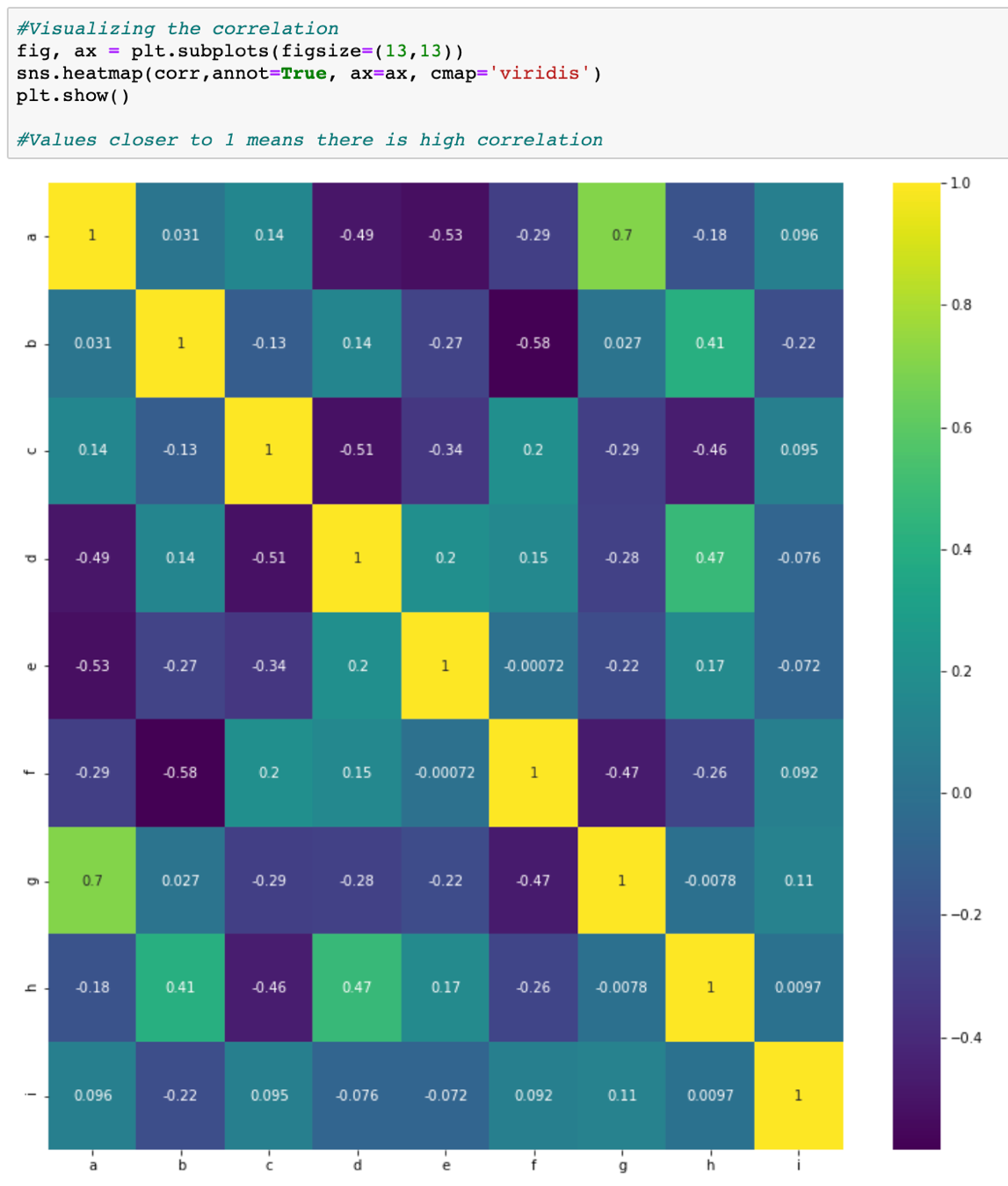
*Figure 4. Histogram and density plot*

From Figure 4, we can see that additive ‘c’ is a bi-model, which means there are two distinct peaks/groups within the data. Most of the column can also be considered as normally distributed, except for the column f, g, h, and i, which are right-skewed. Another thing that can be pointed out is that c, f, h, and i are columns inflated with 0.



*Figure 5. Boxplot*

From Figure 5, we can see boxplots of the data. The boxplots show that there are outliers present in each column. However, due to the outliers not being too extreme and has a value that is close to the whiskers, we can let them be as they could be valid data.



*Figure 6. Pearson Correlation Visualization*

For correlation testing, as we don’t have a target variable, ANOVA testing is not possible. However, we can still conduct coefficient correlation as the dataset contains continuous data. The type of coefficient correlation we will perform is Pearson’s. Figure 6 is a heatmap that maps out the correlation between the variables, with values closer to 1 showing positive correlation and values closer to -1 showing negative correlation.

From the heatmap, we can infer that:

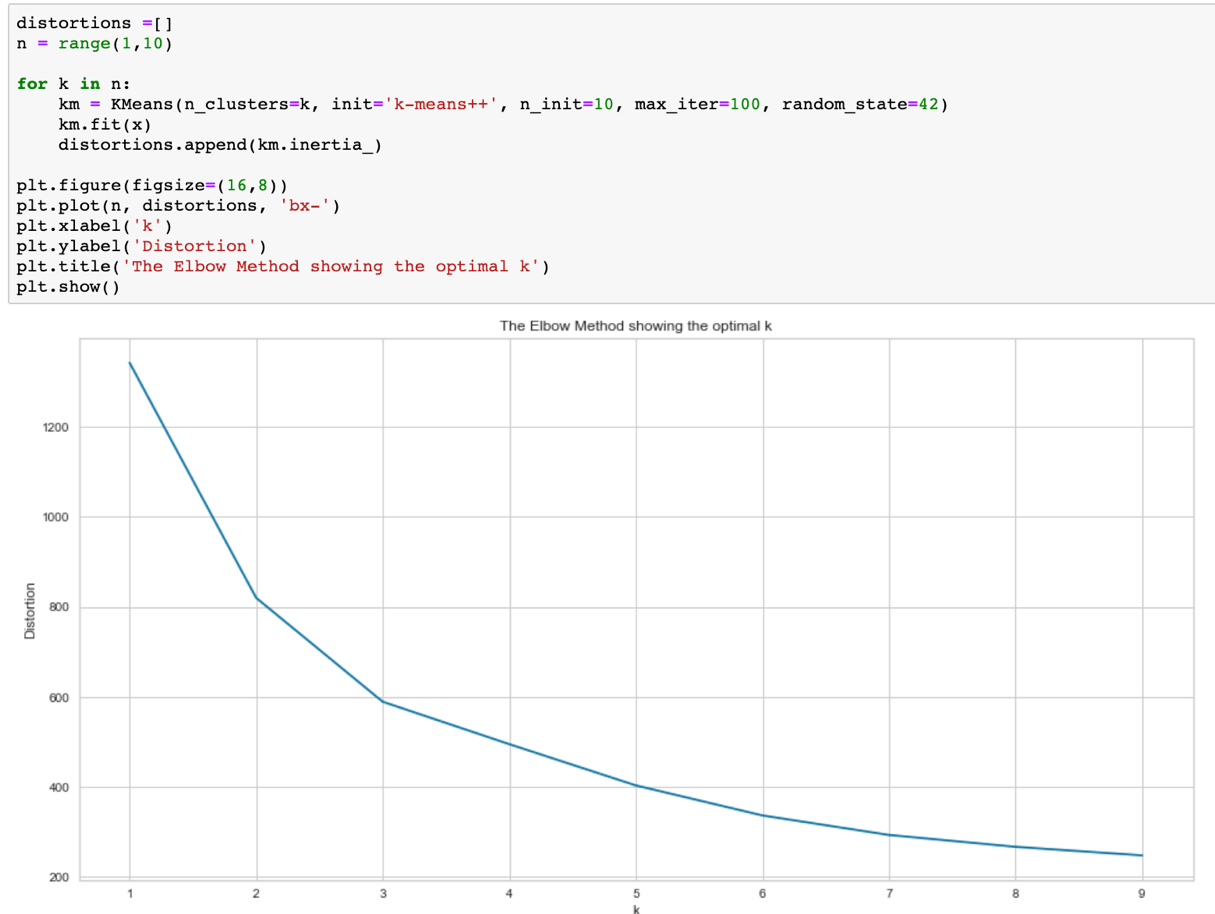
* Additive ‘a’ and ‘g’ have strong correlation
* Additive ‘a’ and ‘e’ have a low correlation
* Additive ‘b’ and ‘f’ have a low correlation
* Additive ‘c’ and ‘d’ have a low correlation

Variables such as additive ‘a’ and ‘g’ which have strong correlation usually means that they have similar data and very small variance between their data. This can cause Multicollinearity, where two variables have high intercorrelation and can undermine the significance of one of the variables. Thus, if this issue happens, we can drop one of the desired variables. However, as we are conducting clustering later on, which is an unsupervised classification model, we don’t need to drop variables, as clustering models don’t really depend correlation values.



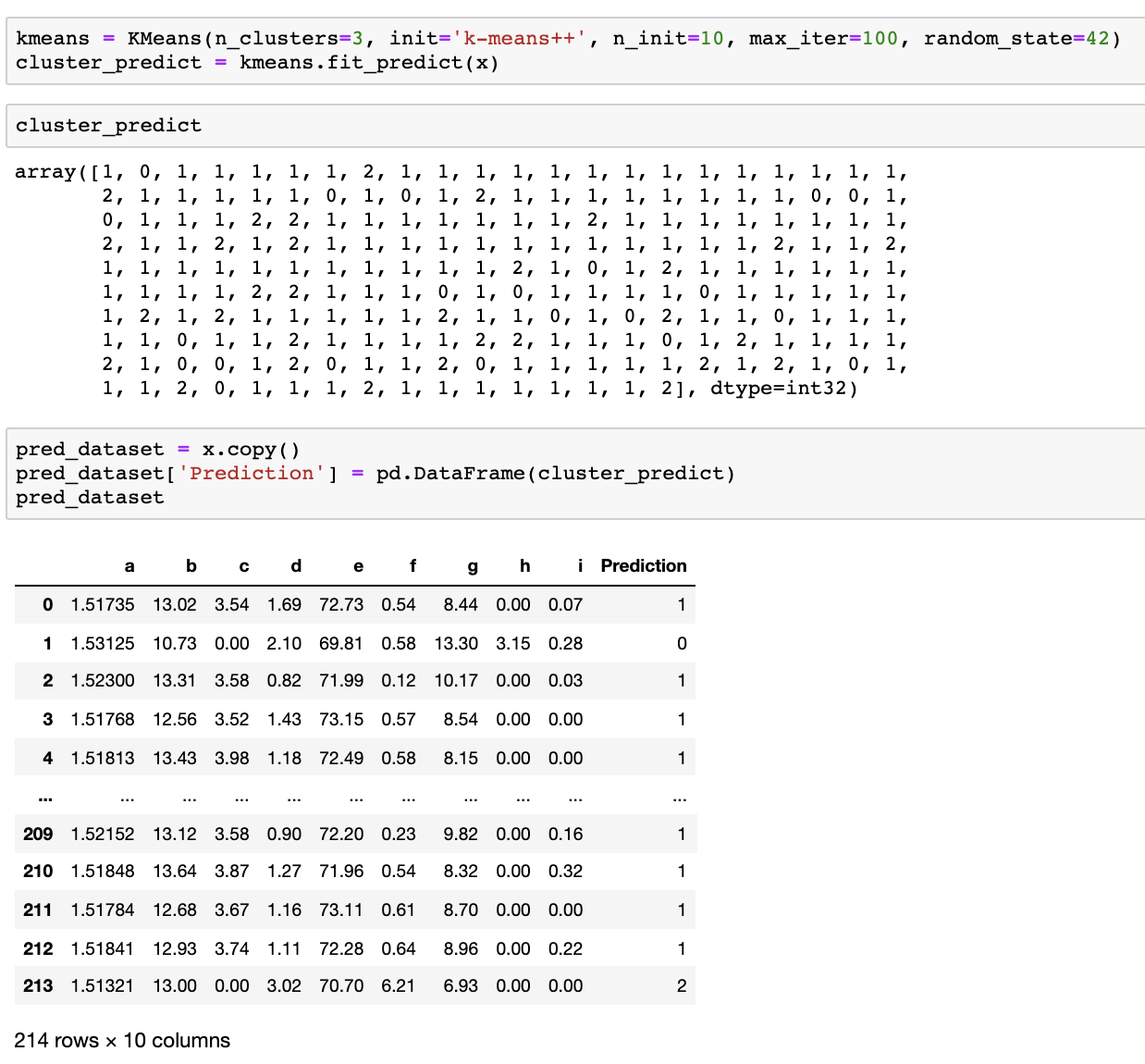
*Figure 7. K-Means Testing*

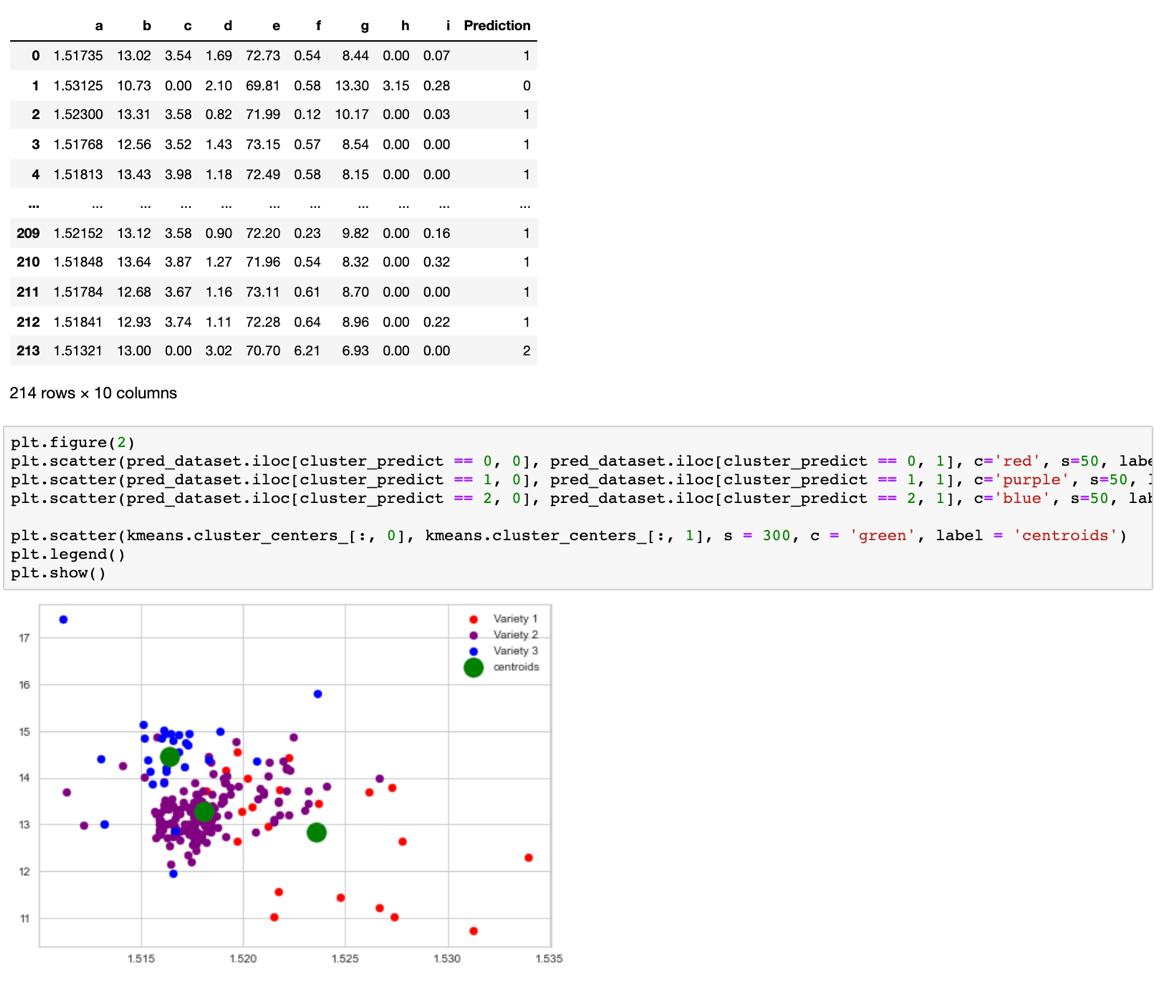
Before we build the K-Means model, we can visualize the silhouette score and the content of each ‘k’ cluster. As seen in Figure 7, we have visualized 2, 3, 4, and 5 k-means cluster. The best ‘k’ is then determined to be 3, as it has an acceptable amount of data within each cluster, as well as having a silhouette score of around 0.5-0.6. The k-means with 4 cluster is also very similar to the k-means with 3 cluster. However, the k-means with 4 cluster has one cluster that contain very little data (colored in blue), which makes it less desirable. Thus, the acceptable ‘k’ is 3. We can further prove this by using the elbow method seen in Figure 8.



*Figure 8. Elbow method*

Once we have determined the right ‘k’, we can then begin building the K-Means model and visualize the clusters created.



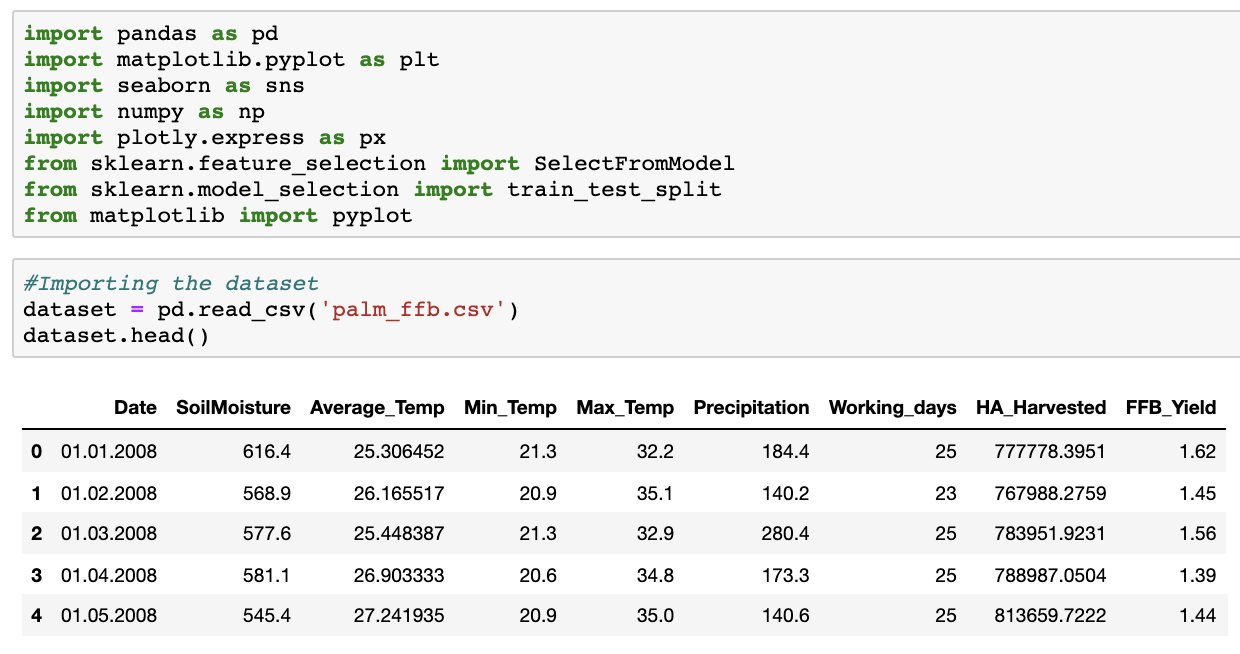


*Figure 9. K-Means Model Building and Visualization*

We can see from Figure 9 that each cluster are separated evenly and each centroid has some distance between them. Thus, we can say that between the mix of additives, there are about **3 variations of formulations** present.

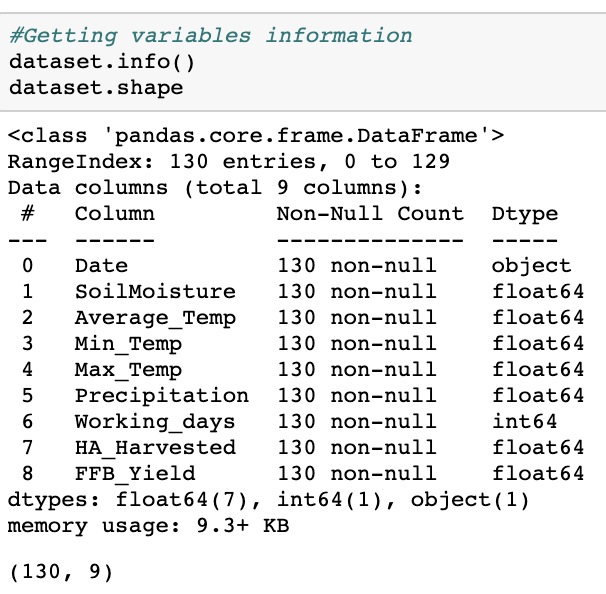
**QUESTION 2**

### A team of plantation planners are concerned about the yield of oil palm trees, which seems to fluctuate. They have collected a set of data and needed help in analysing on how external factors influence fresh fruit bunch (FFB) yield. Some experts are of opinion that the flowering of oil palm tree determines the FFB yield, and are linked to the external factors.



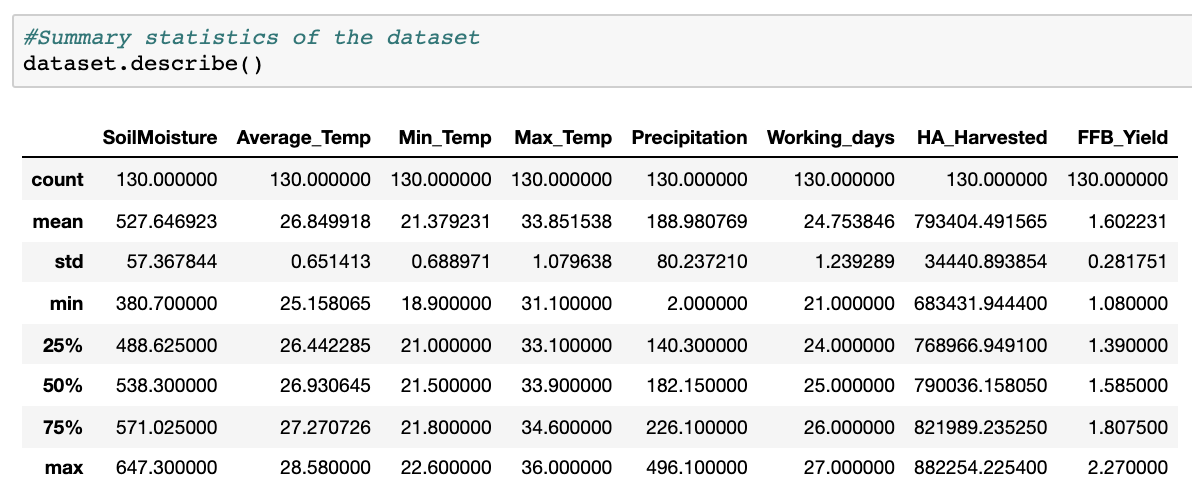
*Figure 10. Import libraries and palm dataset*

Same as Question 1, we will import the libraries that are going to be used later on along with the palm dataset.



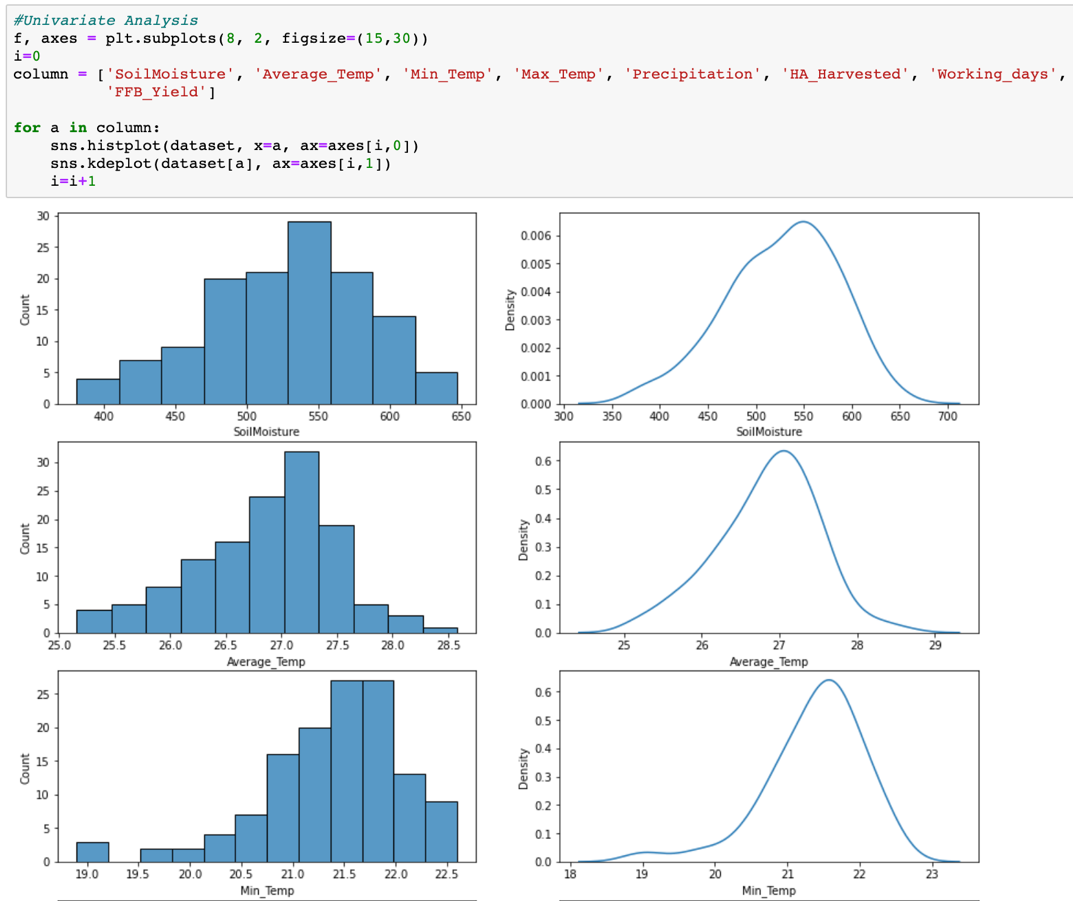
*Figure 11. Palm dataset variable information*

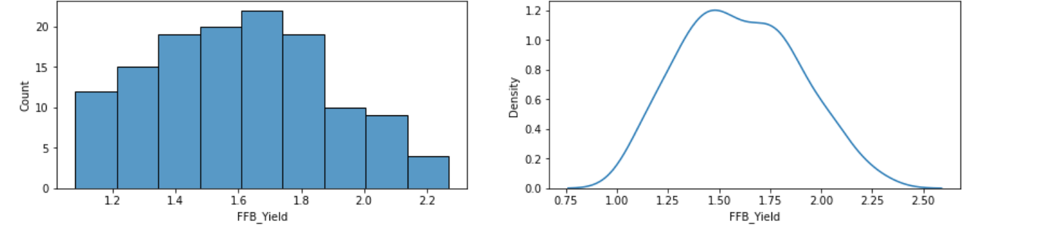
The palm dataset has a total of 130 observations/row and 9 columns. There are 3 data types in the dataset, which are int, float, and object or nominal data. There is no data with NULL value in the dataset.



*Figure 12. Palm dataset summary statistics*

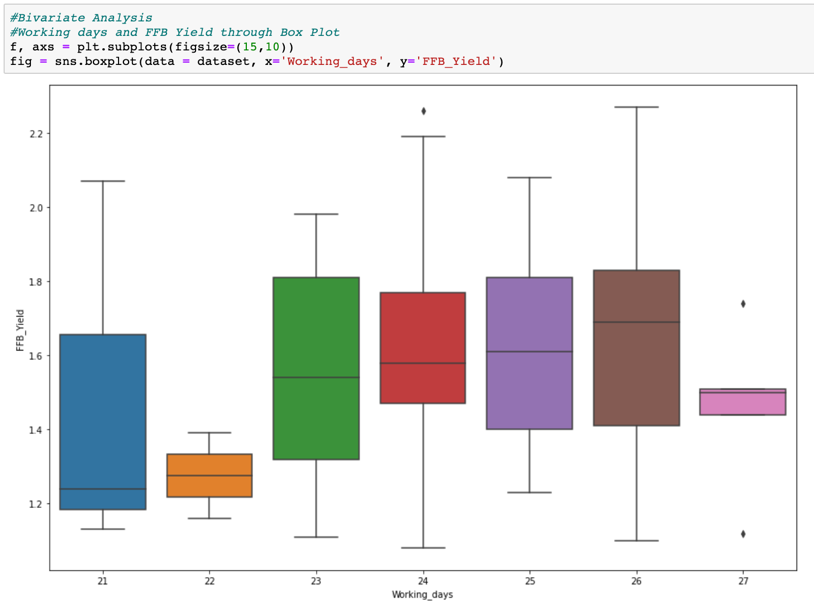
From the summary statistics, we can see that the mean and median are very close to each other, pointing to an almost symmetrical distribution. This can be proven using histograms and density plot. The standard deviation and variance are also very small, excluding soil\_moisture, precipitation, and HA\_Harvested.





*Figure 11. Palm dataset histogram and density plot*

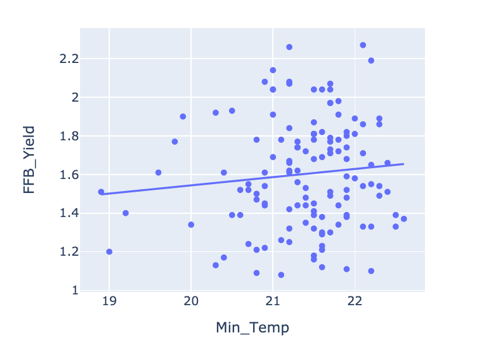
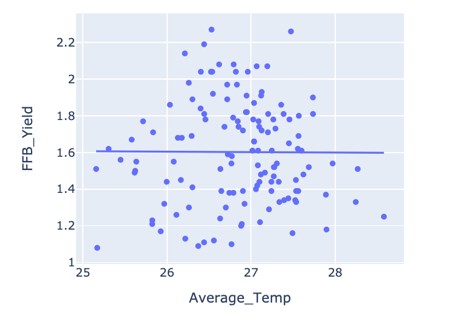
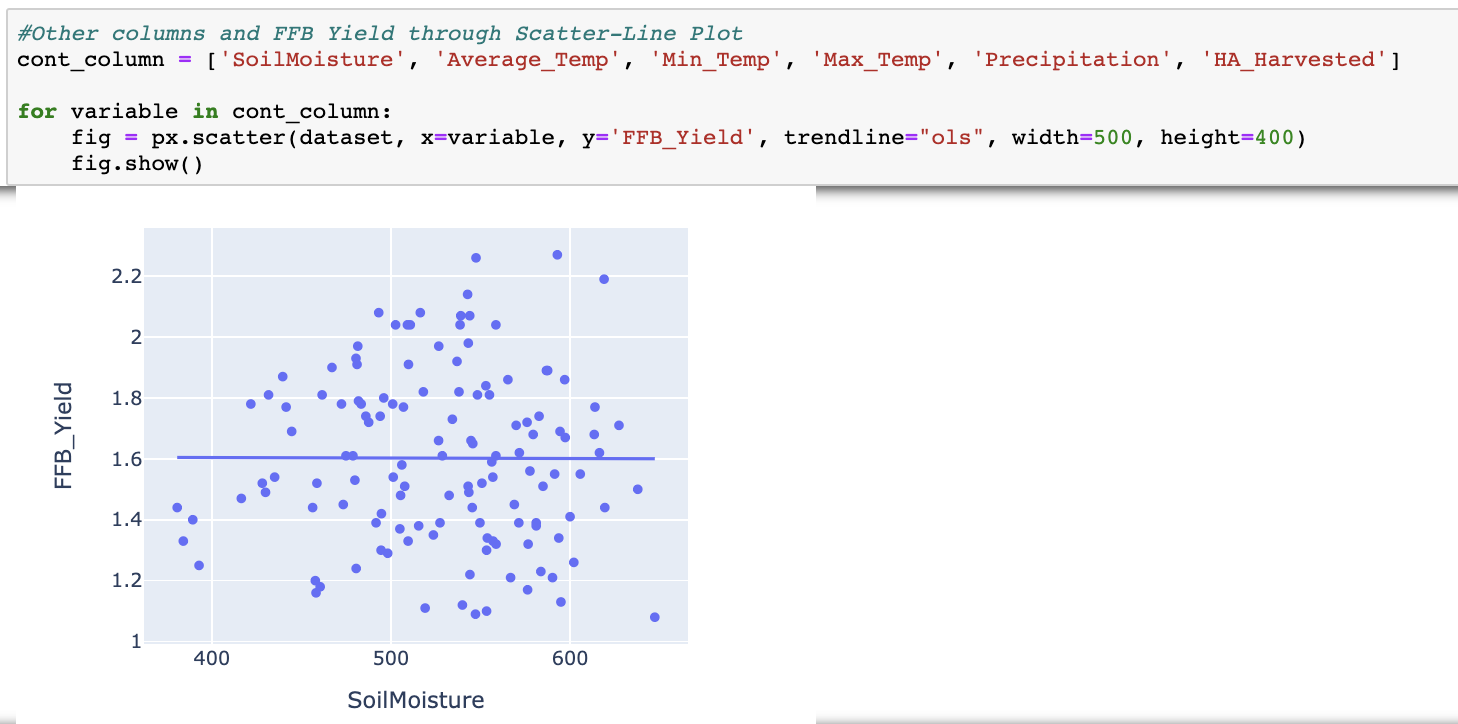
From the histogram and density plot, we can see that all the variables in the palm dataset are almost normally distributed and there is no issue of skewness or kurtosis.

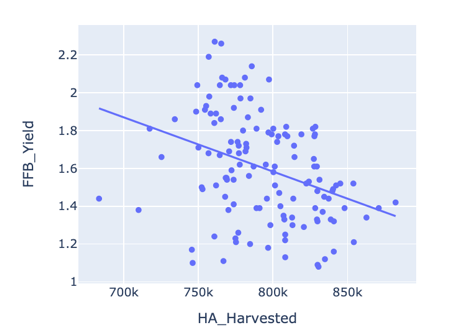
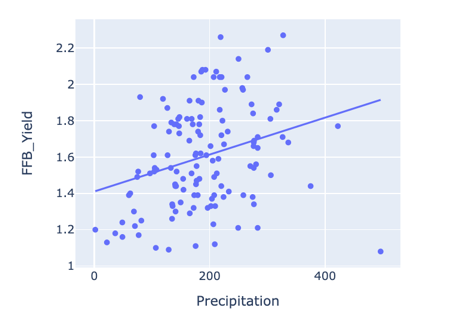


*Figure 12. Working\_days plot against FFB\_Yield*

Besides plotting single variables, we can also plot two variables against each other, such as Figure 12 for example, where we plot working\_days against FFB\_Yield. From Figure 12, we can infer that as working days increase, FFB\_Yield tends to increase. This is apparent as the median of the box plot increase with increasing working days, except for 27. This can be explained because there may not be enough data within the dataset, as proven by the low count of 27 working days in the histogram in Figure 11.

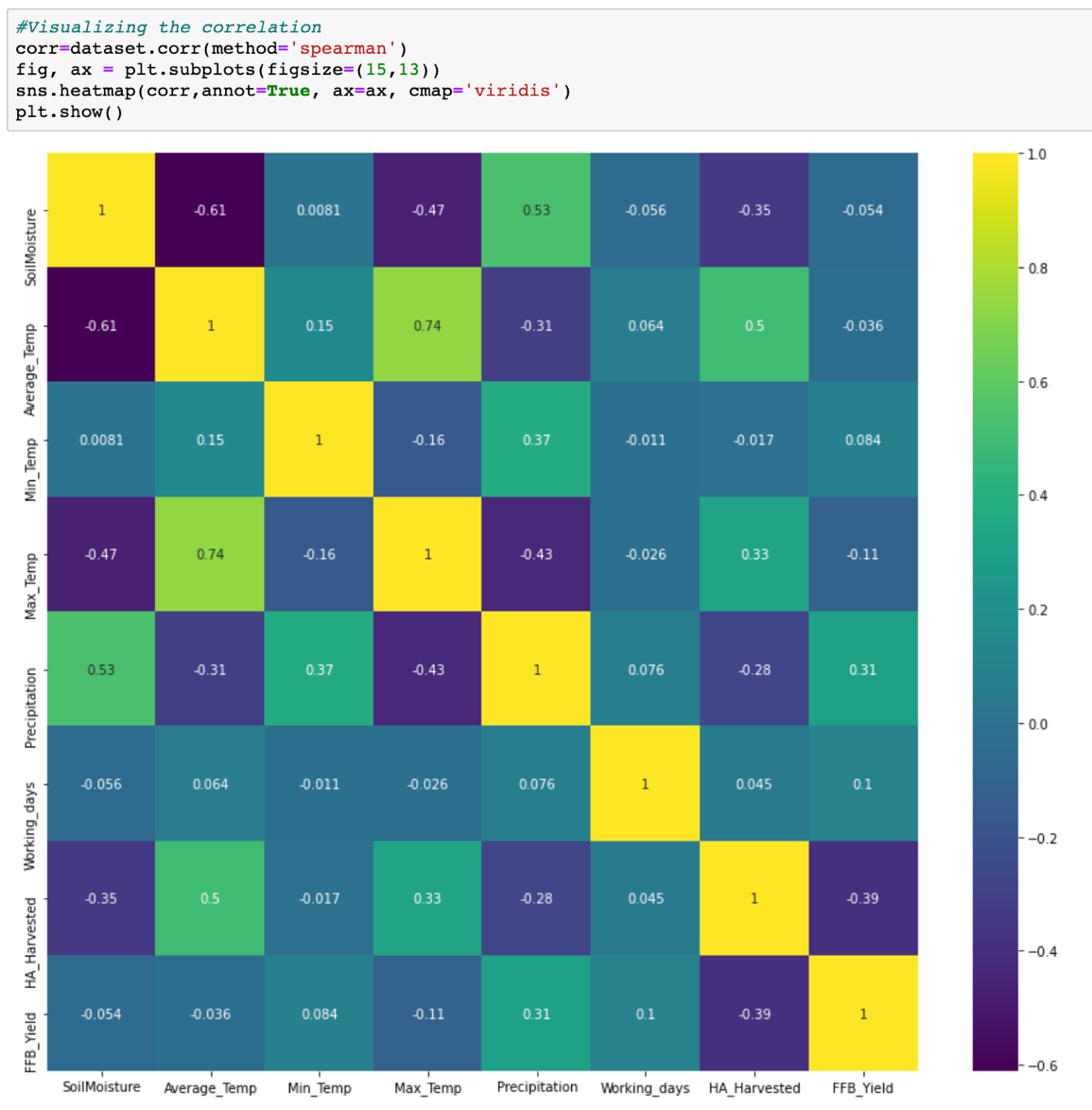
We can also visualize other variables against FFB\_Yield to analyze their correlation through scatter-line plots.





*Figure 13. Scatter-Line Plot Correlation*

From Figure 13, we can have some idea regarding the correlation of each variable with FFB\_Yield. SoilMoisture and Average\_Temp has no clear correlation with FFB\_Yield. Min\_Temp has a slight positive correlation with FFB\_Yield, while Max\_Temp has a slight negative correlation. Precipitation has a positive correlation with FFB\_Yield, whereas HA\_Harvested has a negative correlation. To further cement the correlation, we can use Pearson’s correlation coefficient and visualize it into a heat map, as seen in Figure 14.



*Figure 14. Pearson’s Correlation Heat Map*

From Figure 14, we can see that the top variable that has significant impact on FFB\_Yield are Precipitation, HA\_Harvested, Working\_days, and Min\_Temp. By using Pearson’s Correlation Coefficient, we can then determine that these factors are what impact FFB\_Yield the most and should be closely monitored. However, we can also perform other feature importance method to validate if these 4 variables are truly the best variable to monitor.



*Figure 15. Embedded Feature Importance Method*

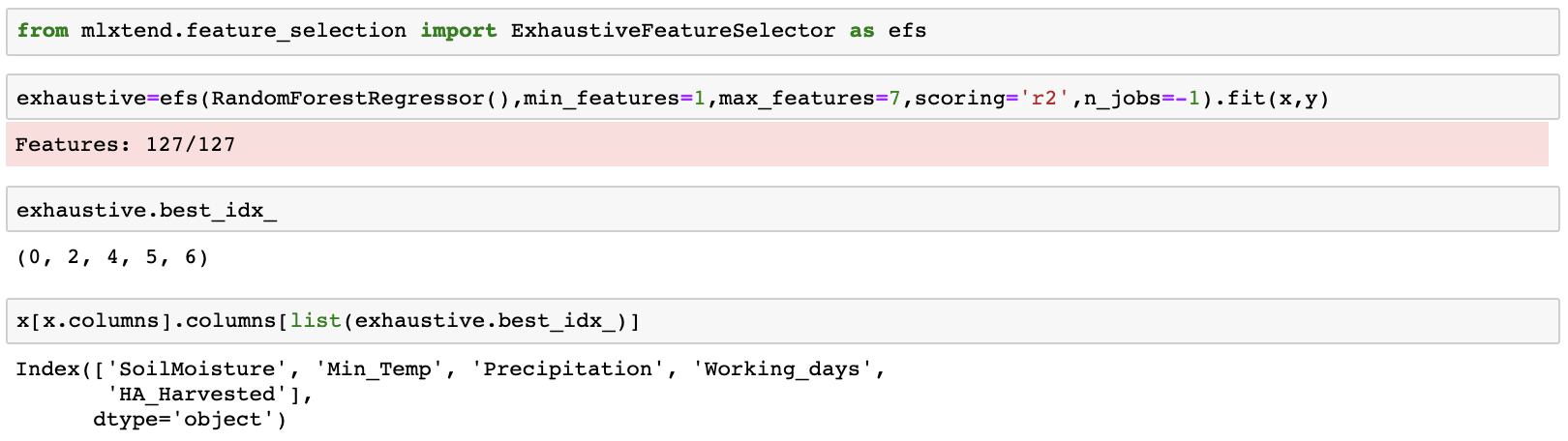
The Embedded Feature Importance method uses an in-built function from a regressor or classifier to select important features. In the Figure 15 example we used the RandomForestRegressor() to perform embedded feature importance. The result of the model is that Precipitation and HA\_Harvested are the best feature to use.



*Figure 16. Wrapper Feature Importance Method*

The Wrapper Feature Importance method uses iteration to seek the best combination of features that can be used for the model specified. Popular wrapper methods are forward selection and backward selection. In Figure 16, we perform both forward and backward selections. Forward selection is a greedy algorithm that starts with null variable and begins adding one variable to the model at a time. Whereas backward selection starts with all variable and begins removing one variable from the model at a time. Both of these methods aim to find the best combination for the model. The result of these methods, as seen in Figure 16, are SoilMoisture, Precipitation, Working\_days, and HA\_Harvested for Forward Selection, and SoilMoisture, Min\_Temp, Precipitation, and HA\_Harvested for Backward Selection.

Another wrapper method that can be used is an Exhaustive Feature Selection where every combination of variables possible are tested against the model to choose the best combination. Exhaustive selection is the one of the best selection methods, however it takes a lot of CPU power as it needs to test out every combination.



*Figure 17. Exhaustive Feature Selection*

After testing each combination possible, the exhaustive feature selection has selected 5 best features to be used which are SoilMoisture, Min\_Temp, Precipitation, Working\_days, and HA\_Harvested.

In conclusion, from the 4 methods used to select important features, we can determine that **HA\_Harvested**, **Precipitation**, and **Working\_days** definitely affect FFB\_Yield. Other external factors that may impact FFB\_Yield include **Min\_temp** and **SoilMoisture**.

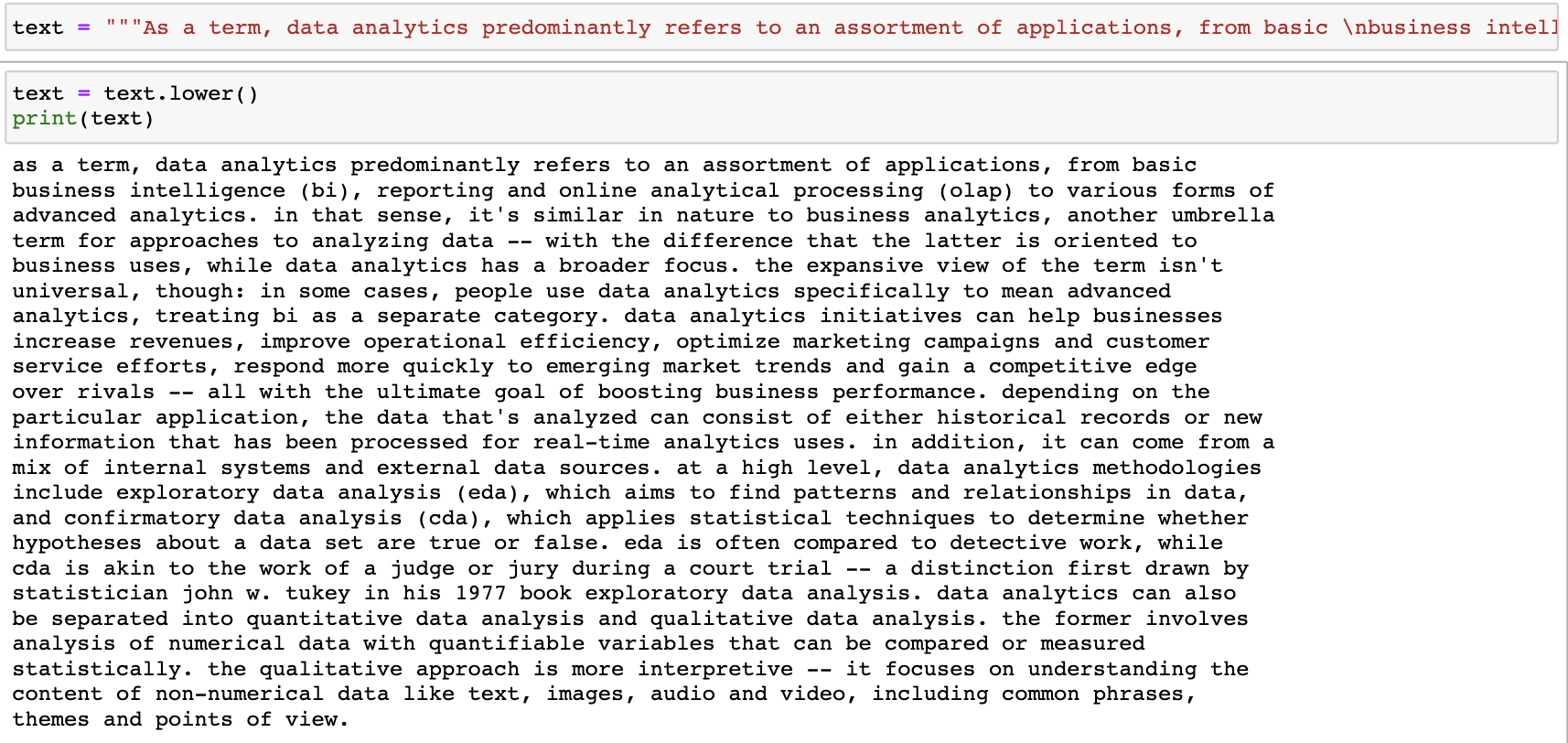
**QUESTION 3**

Feed the following paragraph into your favourite data analytics tool, and answer the following;

a. What is the probability of the word “data” occurring in each line ?

b. What is the distribution of distinct word counts across all the lines ?

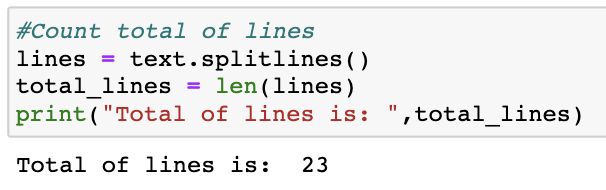
c. What is the probability of the word “analytics” occurring after the word “data” ?



*Figure 18. Importing text as it is*

Before answering the questions, we need to import the text into Jupyter Notebook as it is given.

1. Probability of the word “data” occurring in each line

****

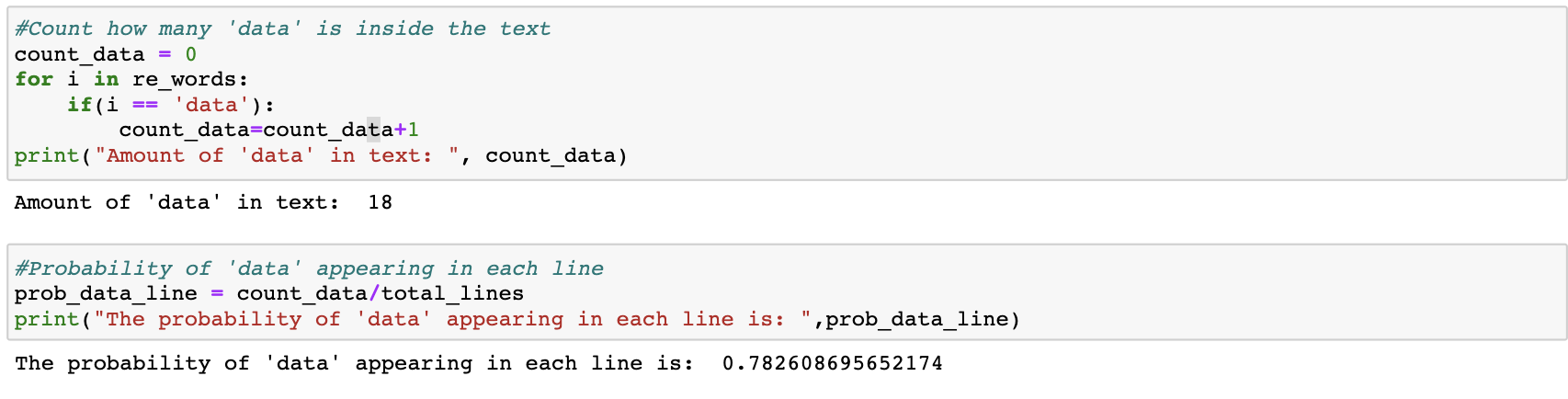
*Figure 19. Splitting the text into lines*

After importing the text, it is important to count the total of lines there are. The imported text has a total of 23 lines.



*Figure 20. Word tokenization*

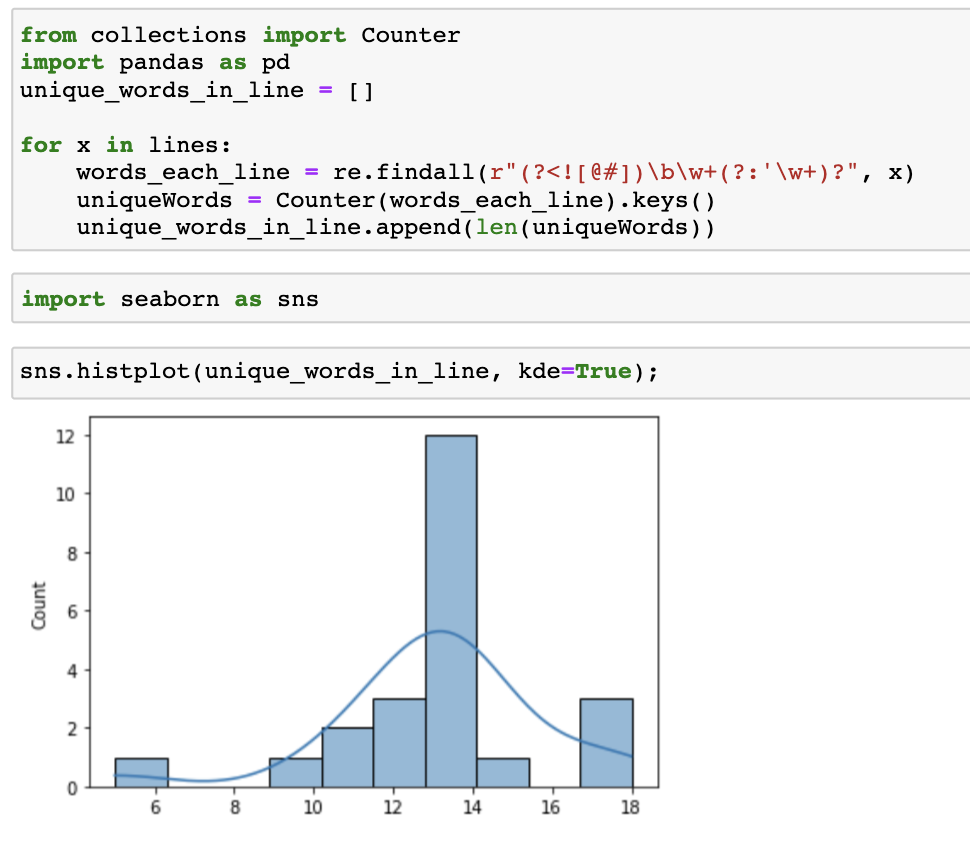
We can then split sentences from the text into singular words. In Figure 20, we use regular expression to split the words because we can then remove any symbols and punctuations, however still keeping apostrophes on words like “it’s”. The total of words available in the text is 317.



*Figure 21. Probability of data in each line*

After getting all of the words in the text, we can then find the frequency of data in the text, which is 18. Then, we can find the probability of ‘data’ appearing in each line by dividing the amount of ‘data’ against the total lines available, in this case 18/23, which would result in 0.782. Thus, the probability of data appearing in each line is **0.782** or **78,2%.**

1. Distribution of distinct word counts across all the lines



*Figure 22. Unique word distribution across all lines*

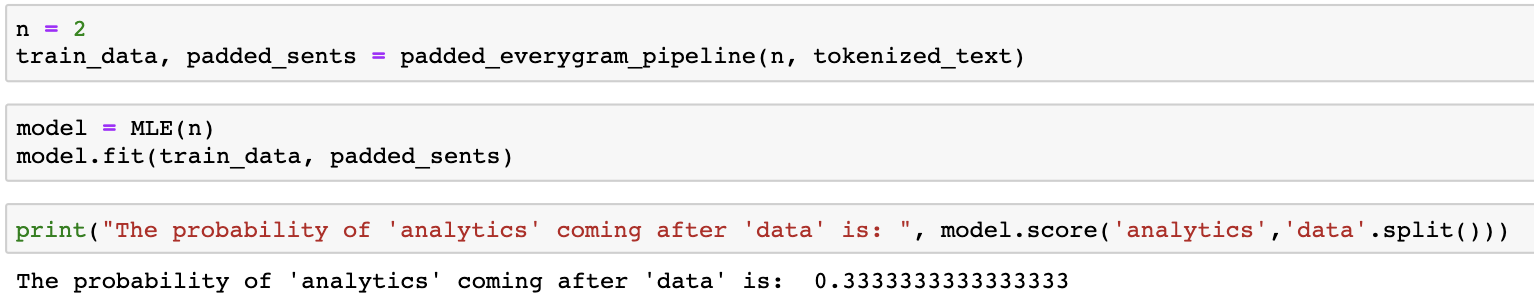
To find how every unique distributes across all line, we need to find the unique word in each line and plot them to a histogram-density plot. From Figure 22, we can see that unique words across the line are almost distributed normally across all line to an extent, where there is certain line with small amount of unique words. This is because of the last line of the text which only contain 5 words.

1. Probability of the word “analytics” occurring after the word “data”



*Figure 23. Re-tokenize words*

To find the probability of ‘analytics’ occurring after ‘data’, we can use bi-gram probability using the NLTK library. However, we need to retokenize the text, because symbols are very important in finding n-grams probability.



*Figure 24. Probability of ‘analytics’ after ‘data’*

After tokenizing the words, we can then define n=2 (because we want to do bigram probability). Then, we can feed the tokenized\_text and n into a n-gram pipeline provided by NLTK to get the training data and padded sentences. These can then be used to train the bi-gram model to get the probability. After the model has been trained, we then get the probility of ‘analytics’ after ‘data’ in the text, which is **0.3333** or **33,3%.**