# Lab Part 1 – Manuel Eiweck

## Task 1)

My steps to cluster the data were the following:

- Load the data with pandas
- Drop the NaN values
- Scale the dataset
- Cluster the dataset with each recommended pair
- Plot each clustered result in a scatter plot

First, I wanted to cluster with each possible combination of nutrition pairs, however these would end up in 946 calculations at the end which would be way too much so I plotted each combination first without clustering, then choose a handful of combinations which had an interesting pattern / trend visible.

For the clustering part I used KMeans from the sklearn.cluster module. As I do not know the optimum number of clusters, I used the elbow Method this works by clustering with different cluster numbers (in my case from 1-15) then comparing its scores and pick the best one. The score is based on the distance of each point in the cluster to its assigned centroid.

Then I plotted each result in a scatter plot with different colors for each assigned cluster. The result can be seen in Figure 2.

For the final decision I would go for a visualization based one by comparing the resulting plots. I would prefer the Combination of Water and Energy as seen in Figure 2. Because there is a clear trend and correlation detectable. As the amount of Water decreases the amount of energy increases which makes sense as water has less energy then other ingredients. Another good combination could be Folate_DFE_(µg) and Folic_Acid_(µg)

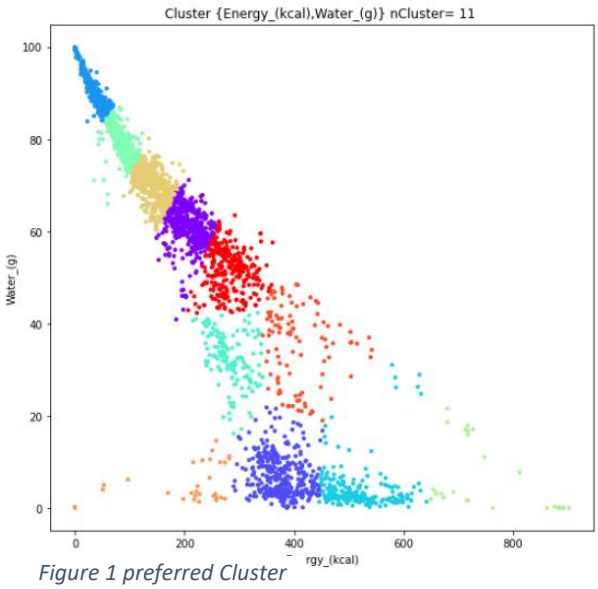


*Figure 1 preferred Cluster*

So in conclusion the selection of the number of clusters were easier to solve by automated and statistical analysis also the assignment of the clusters.

The selection of the nutrition pair is easier by visualization as we can detect patterns and trends and after the clustering how meaningful the assigned clusters are and if it even gives us any useful information

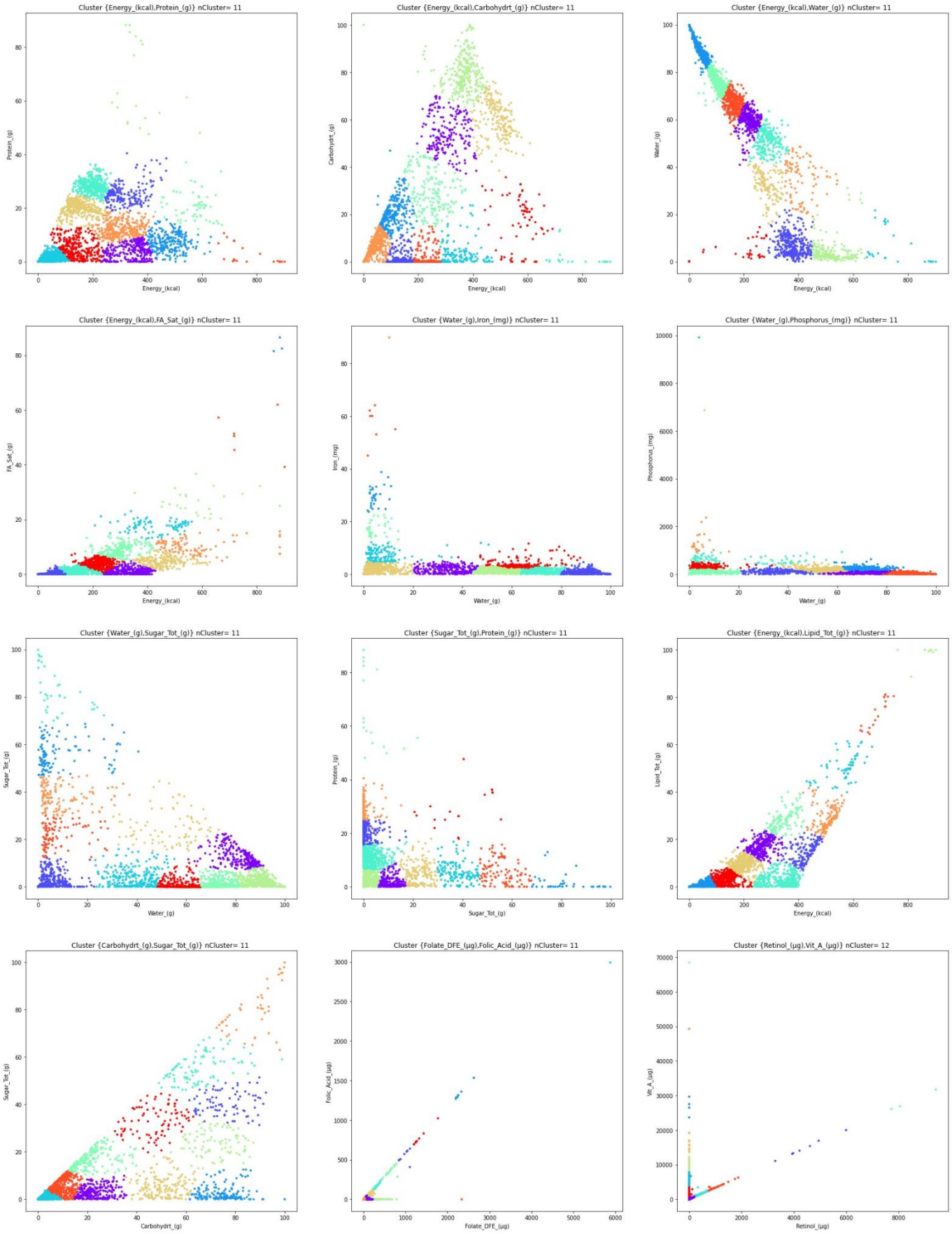


*Figure 2 clusters for the recommended pairs and custom chosen pairs*

# Task 2)

Found Correlations:

- Folate_DFE_(µg) Folate_Tot_(µg) 0.98
- Folate_DFE_(µg) Folic_Acid_µg) 0.94
- Energy_(kcal) Water_(g) -0.91
- FA_Mono_(g) Lipid_Tot_(g) 0.89
- Folic_Acid_(µg) Folate_Tot_(µg) 0.86

Steps:

- load data with pandas
- drop NaN values
- calculate a correlation matrix with pandas corr()

With the raw calculation matrix data I generated a correlation heatmap matrix visualization with seaborn ‘seaborn.heatmap(correlationData,annot=True)’ see Figure 4. There we can read the strongest positive and negative correlations. However as the matrix is relatively huge I also ordered the matrix by its values using ‘unstack’ and ‘sort_values’ see Figure 3.

| | | |
|---|---|---|
| Water_(g) | Energy_(kcal) | -0.913302 |
| Energy_(kcal) | Water_(g) | -0.913302 |
| Carbohydrt_(g) | Water_(g) | -0.775625 |
| Water_(g) | Carbohydrt_(g) | -0.775625 |
| Choline_Tot_ (mg) | Cholestrl_(mg) | 0.786464 |
| Sodium_(mg) | Ash_(g) | 0.829513 |
| Ash_(g) | Sodium_(mg) | 0.829513 |
| Folate_Tot_(µg) | Folic_Acid_(µg) | 0.863999 |
| Folic_Acid_(µg) | Folate_Tot_(µg) | 0.863999 |
| Lipid_Tot_(g) | FA_Mono_(g) | 0.886816 |
| FA_Mono_(g) | Lipid_Tot_(g) | 0.886816 |
| Vit_A_(µg) | Beta_Carot_(µg) | 0.892891 |
| Beta_Carot_(µg) | Vit_A_(µg) | 0.892891 |
| Folate_DFE_(µg) | Folic_Acid_(µg) | 0.943640 |
| Folic_Acid_(µg) | Folate_DFE_(µg) | 0.943640 |
| Folate_DFE_(µg) | Folate_Tot_(µg) | 0.981942 |
| Folate_Tot_(µg) | Folate_DFE_(µg) | 0.981942 |

*Figure 3 correlations ordered by strength*

On huge datasets the correlation matrix can get overly complex and large therefore it is not possible anymore to read data from it. In this case the correlation can only be extracted using a statistical approach by ordered them.

Another completely visualization-based approach would be to display a parallel plot for every possible nutrition pair. Like the scatter plot for each possible pair in task 1. However, this method is scaling way worse as we would have to look at 946 plots in our case. Here some correlations will be clearly missed.

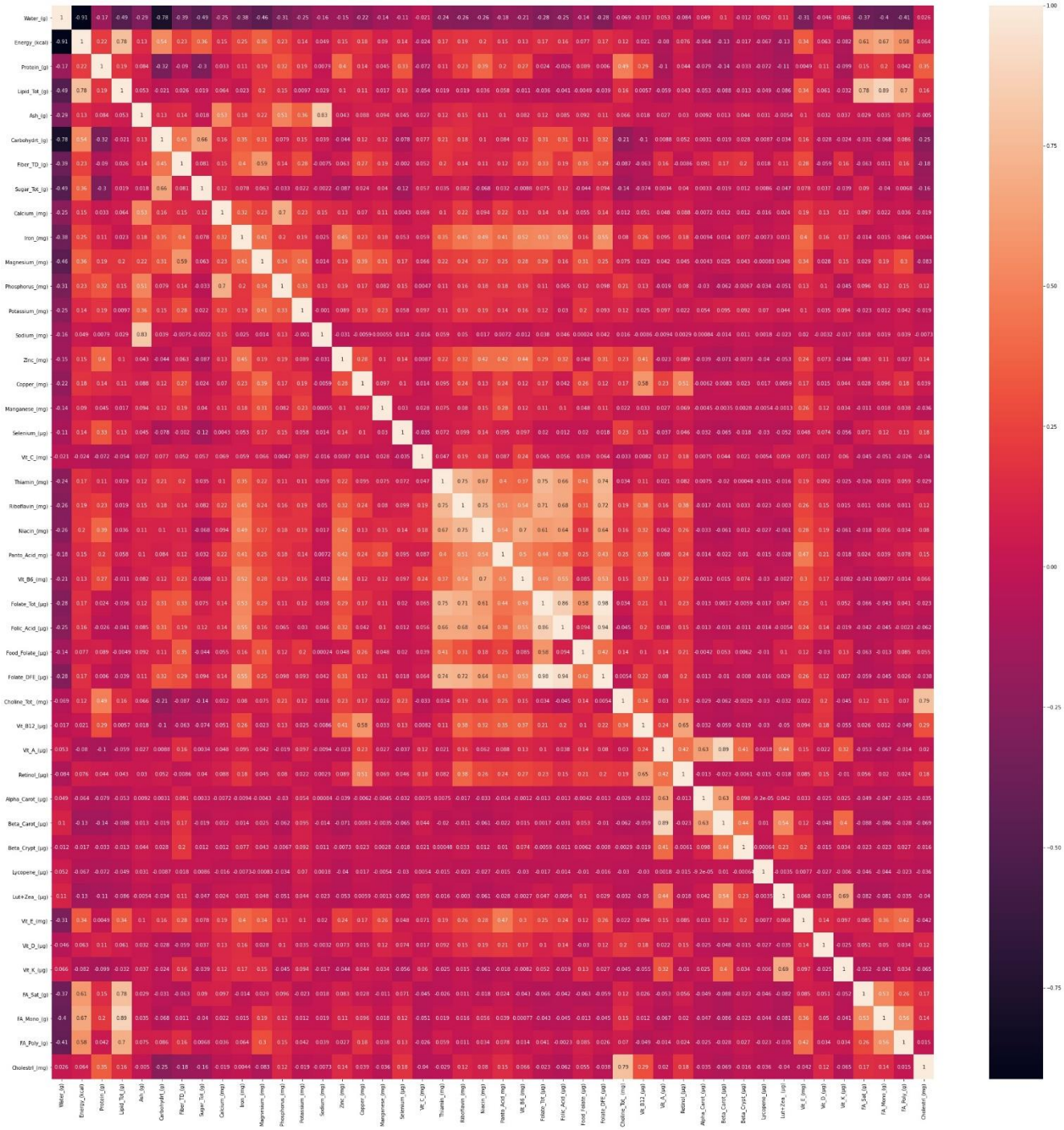

*Figure 4 correlation matrix visualization*

## Task 3)

After the usual loading, the data steps I started by figuring out a way to access my data for each group. I came across the pandas dataframe groupBy method which allows me to access each attribute for every group in our case keyword.

To test it out I want to get the mean of every attribute for each group to compare it.
This can be achieved with this command:
dataset[headerWithValuesAndKeyword].groupby('Keyword').mean()
This provides me an output seen in Figure 5.

| Keyword | Water_(g) | Energy_(kcal) | Protein_(g) | Lipid_Tot_(g) | Ash_(g) | Carbo |
|---|---|---|---|---|---|---|
| ALCOHOLIC BEVERAGE | 72.013684 | 165.157895 | 0.522105 | 1.083684 | 0.163158 | |
| ANIMAL FAT | 16.843333 | 717.333333 | 0.730000 | 80.173333 | 1.273333 | |
| BABYFOOD | 65.359463 | 167.228188 | 4.335839 | 6.695302 | 1.174295 | 2 |
| BEEF | 63.115168 | 201.747204 | 25.275101 | 10.818188 | 1.153602 | |
| BEVERAGES | 72.795641 | 105.166667 | 2.311368 | 0.885214 | 1.054487 | |

*Figure 5 mean for each attribute grouped by keyword*

The problem here is that the amount of values is too large to compare, so I decided to create a custom heatmap like the one I used in task2. However, the normal seaborn heatmap calculates the colors for the whole plot, which would look wrong in my case as I want a color scale per each row of Attribute (water, energy,…) . I achieved this by combining the heatmap plot with plt.subplot, where I plot a separated heatmap with a single row and combine them into a single image without gaps and the labels only on the bottom. See Figure 8 (The resulting plot is quite large, so it was a problem to fit it onto an A4 page)

After that I decided to compare the groups: BEEF, VEGTABLES and SWEETS I choose them because they have the largest amount of data values and therefore should provide a reasonable amount of information.

As I do not know anything about the attributes and do not want to make wrong assumption, I cannot focus on specific attributes, so I must look at each attribute on its own. A radar chart however would not do me a favor as it would have too much data per angle. So, I decided for separated scatter charts of each 946 nutrition-pair combination where I color the dots accordingly to the groups. (BEEF=red,SWEETS=blue,VEGETABLES=green). See Figure 6 for some selected plots, we can see a difference in “Zinc_(mg)” as only BEEF seems to have way higher numbers than the other two groups. This information is seeming true as the statistical analysis shows us that the mean of in “Zinc_(mg)” is 6.1 for BEEF and ~0.86 for VEGETABLES and SWEETS.

Figure 7 shows us a separated heatmap for means of the selected group, this is a good combination of statistical analysis with visualization.

For the question: “Was it easier to do the comparison with statistical analysis only, or by employing visualization?” I would say it heavily depends on what we want to achieve from the data. In our case there was no defined goal so an visualization based approach was more helpful than only numbers. I

think that the combination of both as seen in the heatmap is a good way to go. Especially when looking for patterns or trends scatter plots are useful even when we have a huge number of them detecting significant differences in groups was still possible.

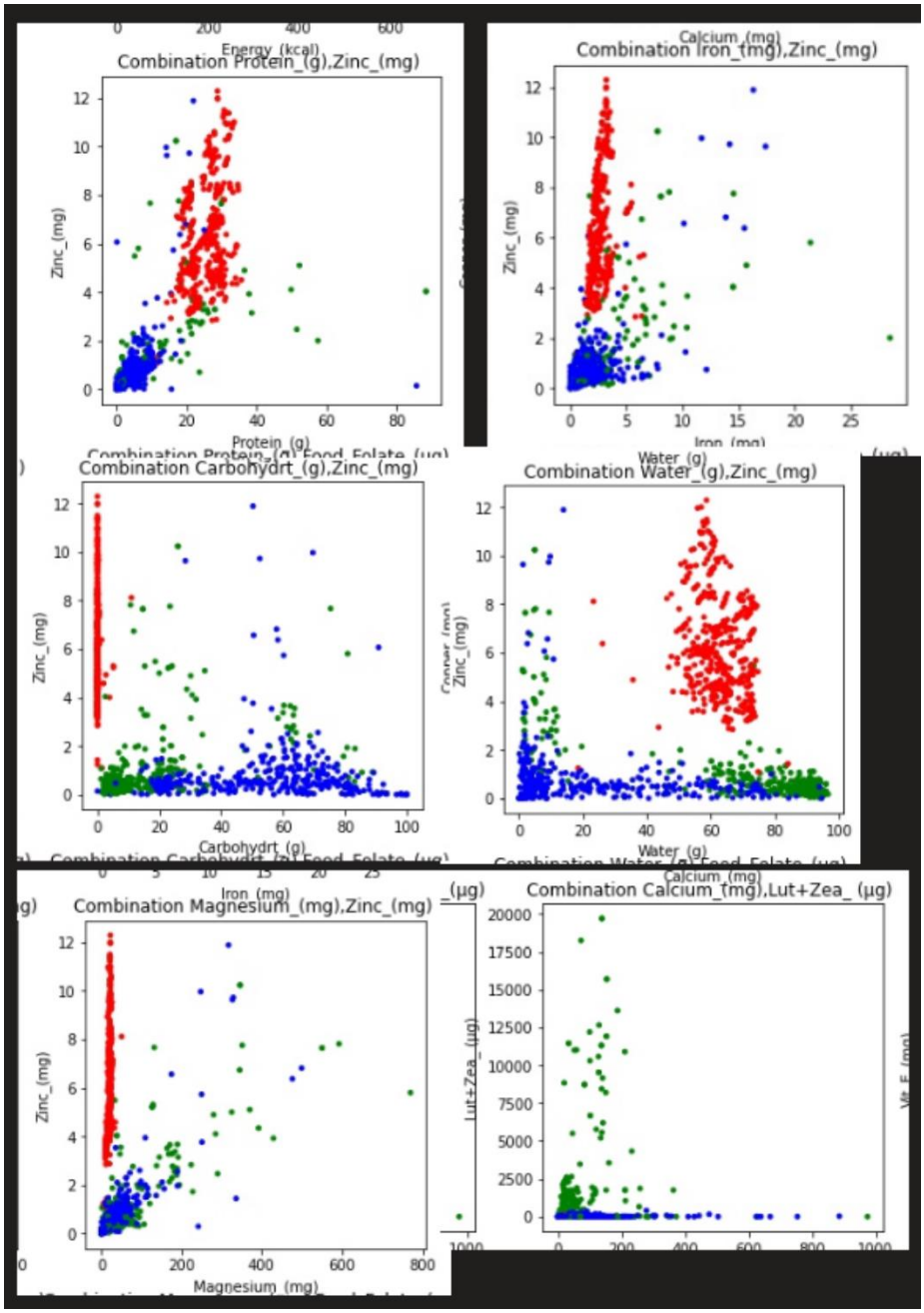


*Figure 6 selected scatter plots for BEEF=red, SWEETS=blue and VEGETABLES=green*

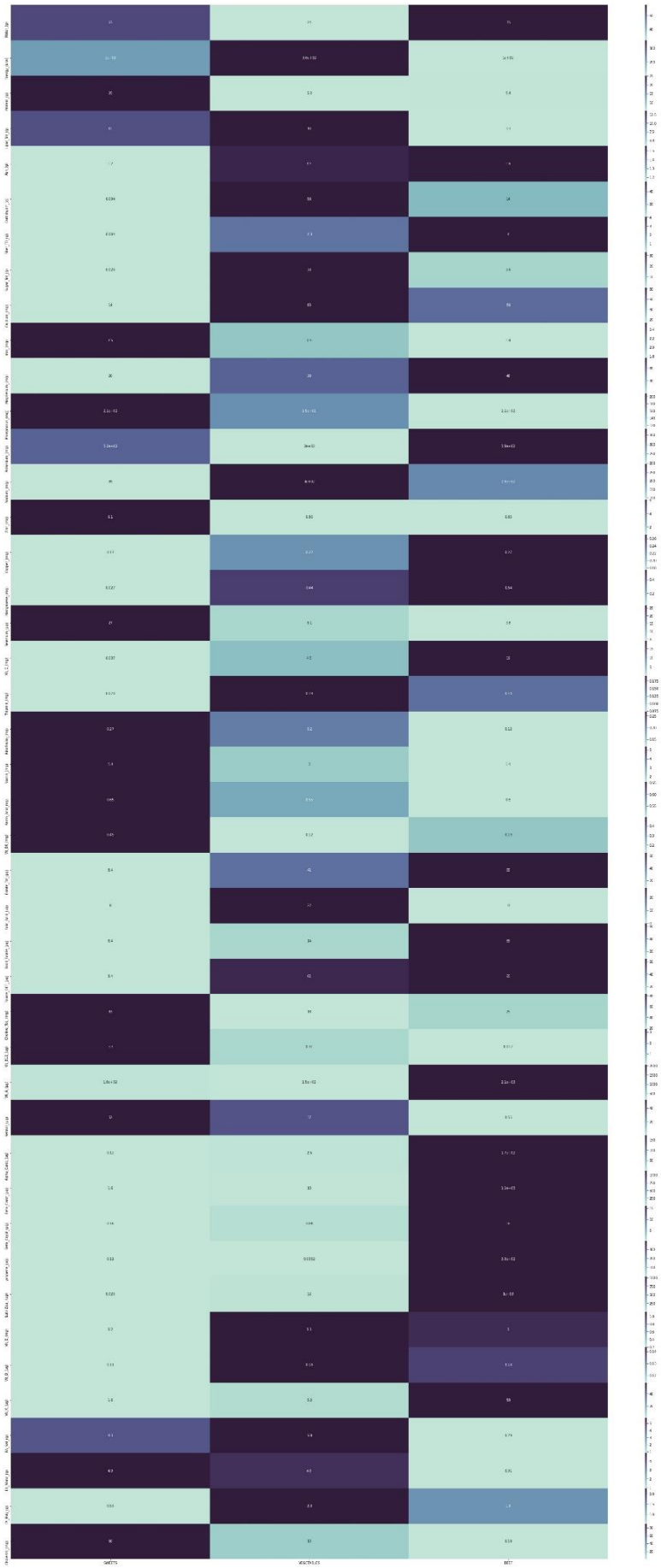

*Figure 7 heatmap of the means for the BEEF, SWEETS and VEGETABLES*

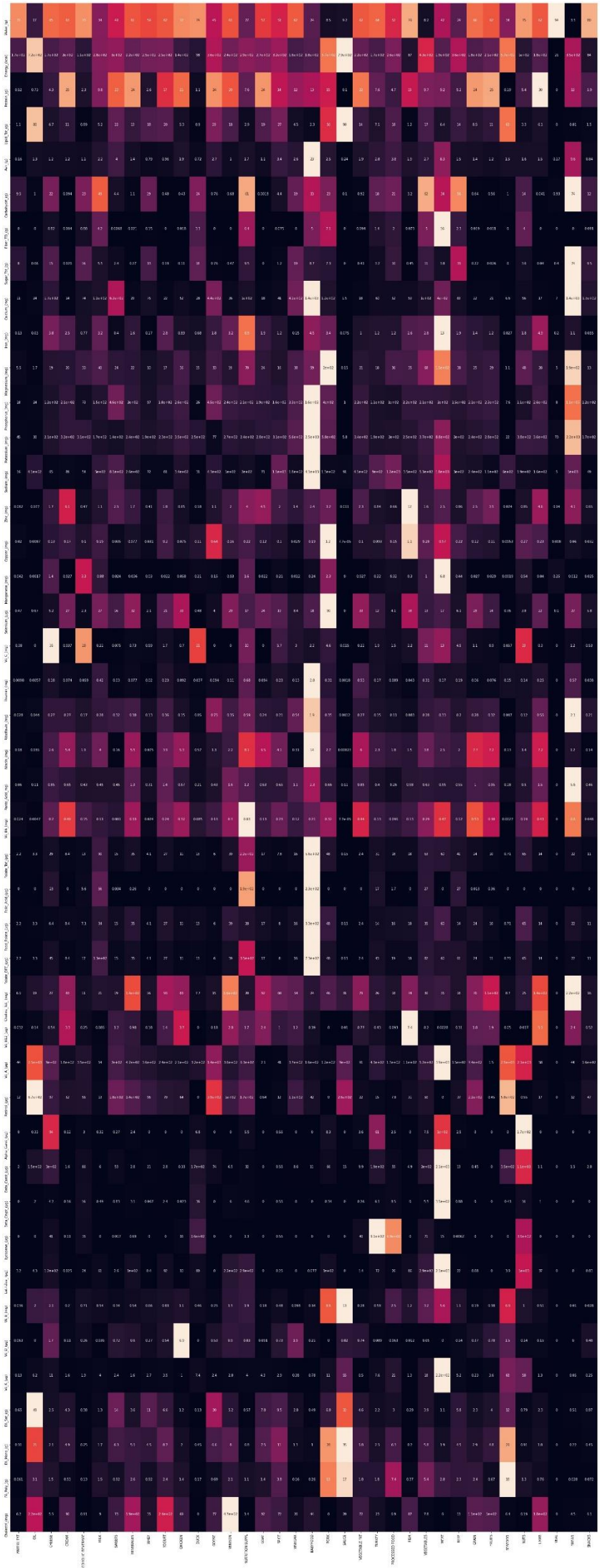

*Figure 10 customized heatmap visualization of the means, each row represents a different attribute, each column a different keyword group*