PROJECT 2

> Thomas Moore
>
> CPSC 392
>
> Professor Parlett
>
> 4/29/21

Donut Nutrition Plots and analysis using different methods

A)

making 3 scatterplots using ggplot to show:

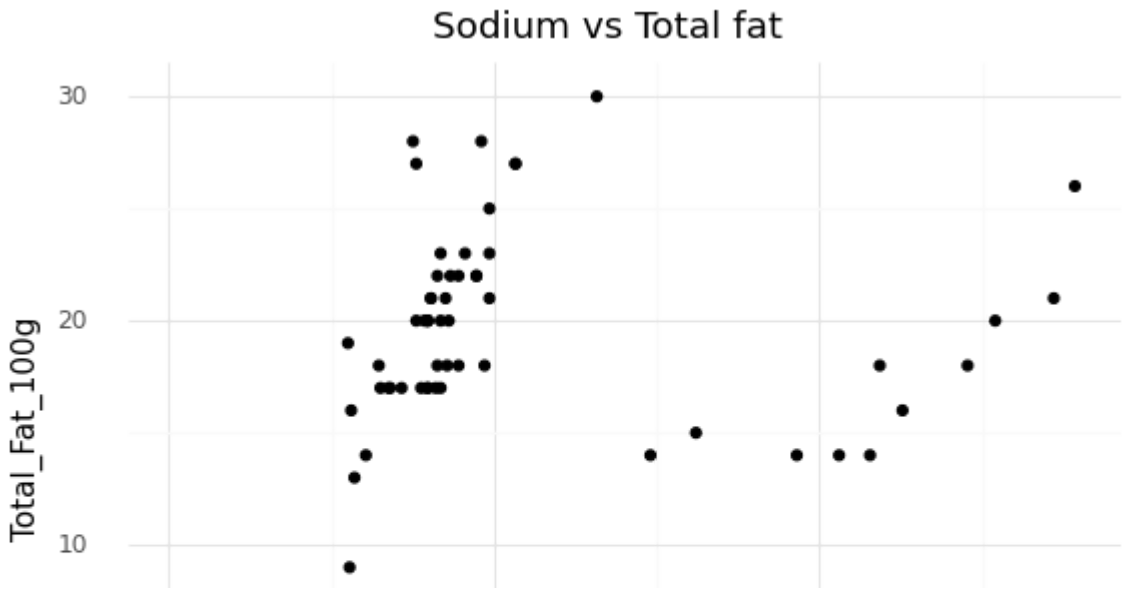
- Sodium_100g vs Total_Fat_100g
- Sodium_100g vs. Sugar_100g
- Sugar_100g vs Total_Fat_100g

```
# importing packages

# import necessary packages
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
from plotnine import *
from sklearn import *
# Linear Regression Model
from sklearn.linear_model import LinearRegression
# Logistic Regression Model
```

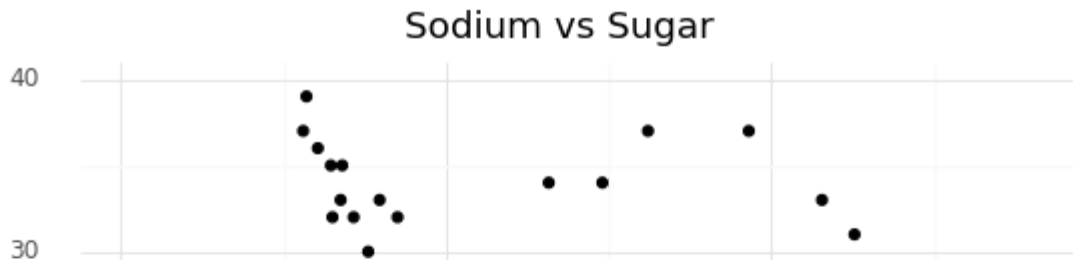
```
from sklearn.linear_model import LogisticRegression
#Z-score variables
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, plot_confusion_matrix
# simple TT split cv
from sklearn.model_selection import train_test_split
# k-fold cv
from sklearn.model_selection import KFold
#LOO cv
from sklearn.model_selection import LeaveOneOut
# cross validation metrics
from sklearn.model_selection import cross_val_score
# cross validation metrics
from sklearn.model_selection import cross_val_predict
# model eval
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
from sklearn.metrics import silhouette_score
from sklearn.neighbors import NearestNeighbors
from sklearn.cluster import DBSCAN
from sklearn.cluster import AgglomerativeClustering
import scipy.cluster.hierarchy as sch
from matplotlib import pyplot as plt
%matplotlib inline

# a)
# reading in file
data = pd.read_csv("https://raw.githubusercontent.com/cmparlettpelleriti/CPSC392ParlettPelleriti/master/Data/KrispyKreme.csv

data.head()
```

	Restaurant_Item_Name	restaurant	Restaurant_ID	Item_Name	Item_Description	I
0	Krispy Kreme Apple Fritter	Krispy Kreme	49	Apple Fritter	Apple Fritter, Doughnuts	
1	Krispy Kreme Chocolate Iced Cake Doughnut	Krispy Kreme	49	Chocolate Iced Cake Doughnut	Chocolate Iced Cake Doughnut, Doughnuts	
2	Krispy Kreme Chocolate Iced Custard Filled Dou...	Krispy Kreme	49	Chocolate Iced Custard Filled Doughnut	Chocolate Iced Custard Filled Doughnut, Doughnuts	
3	Krispy Kreme Chocolate Iced Glazed Doughnut	Krispy Kreme	49	Chocolate Iced Glazed Doughnut	Chocolate Iced Glazed Doughnut, Doughnuts	
4	Krispy Kreme Chocolate Iced Glazed Cruller	Krispy Kreme	49	Chocolate Iced Glazed	Chocolate Iced Glazed Cruller Doughnut	

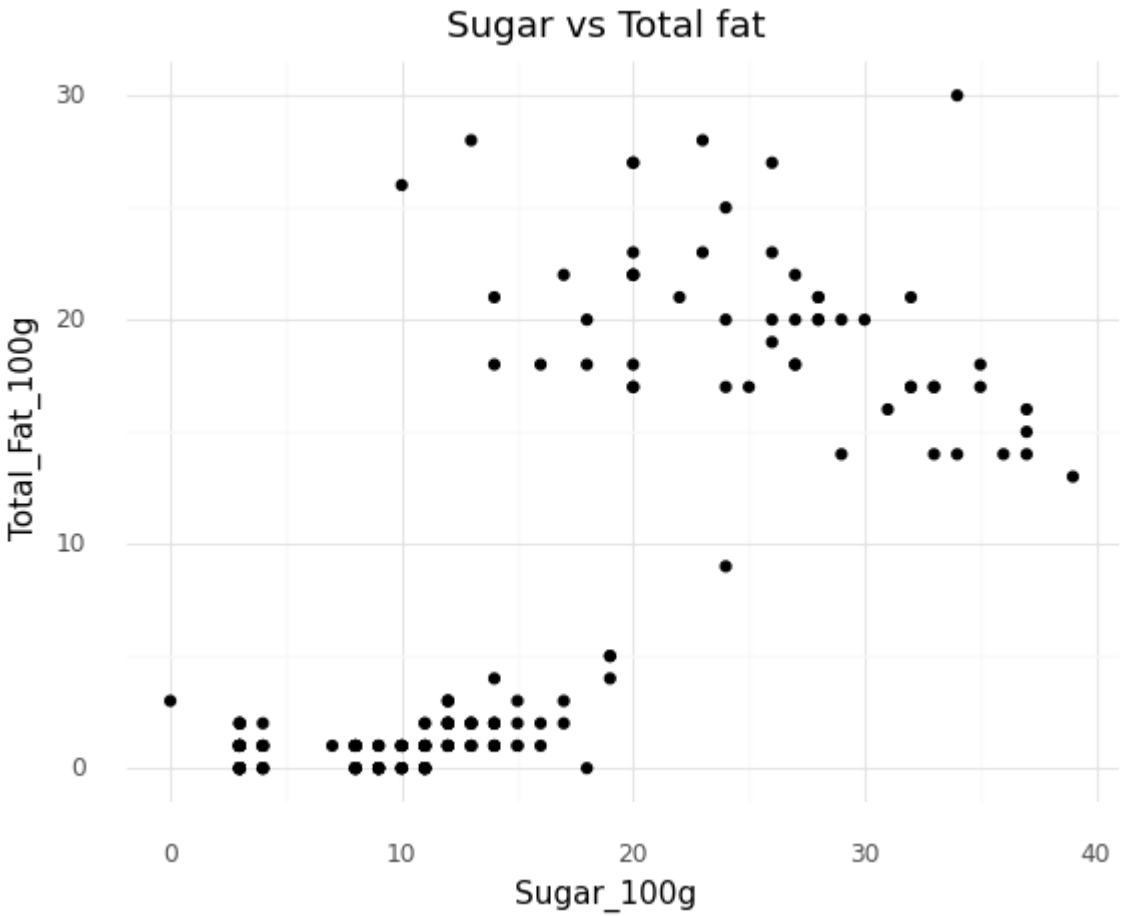
```
# making the scatter plots for A
(ggplot(data, aes(x = "Sodium_100g", y = "Total_Fat_100g"))+geom_point()+ggtitle("Sodium vs Total fat")+
 theme_minimal())
```




```
(ggplot(data, aes(x = "Sodium_100g", y = "Sugar_100g"))+geom_point()+ggtitle("Sodium vs Sugar")+
 theme_minimal())
```




```
(ggplot(data, aes(x = "Sugar_100g", y = "Total_Fat_100g"))+geom_point()+ggtitle("Sugar vs Total fat")+
 theme_minimal())
```




```
<ggplot: (8740423388153)>
```

B)

Looking at the scatter plots above, I think that a **Gaussian Mixture Model** or **Hierarchial Clustering model** will be **most accurate** with this data. This is because I notice pretty clear ovular clustering in all 3 graphs above.

I also notice there is little to no noise at a glance, which makes it much easier to choose a simpler model that doesn't take noise into account. Also the data points seem to have different amounts of variance at different positions on the graph for all 3 visuals. (Best to worst prediction: Hierarchial Clustering, GMM, DBSCAN, Kmeans)

Gaussian Mixture Models gives a normal distribution curve that provides a symmetric and unimodal visual of the data. Often a bell shape curve with most the data in the center, while as the data gets further from the center the data becomes less likely. This is a soft assignment (probability of being in a cluster) which differs from k-means hard assignment.

The Pros of Gaussian Mixture Models are that it can estimate what the variance is between the data which allows us to make assumptions of clusters that are not spherical necissarily (good for data with non-spherical clusters). This allows for more accurate clustering since it is not bound so strictly to a clustering pattern. This is probablistic assignment.

Cons of Gaussian Mixture Models are that it is a bit more complicated than k-means and also we are still assuming data points are in a cluster since we dont take noise into account (bad for data with lots of noise)

K means will not be good for this data set since the data is not circular. K means will make the clustering too specific and lead to more loosely packed cluster (which is worse in terms of accuracy)

K means algorithm assigns each data point to each cluster which in hard assignment format.

The Pros of k-means are that it is iterative process which allows for intuitive clustering. The Cons of k-means are assumption of circular clusters which can lead to a inaccurate assumptions of cluster patterns since the variance between points is assumed to be the same, which is not the case with our donut data (there is different levels of variation between points).

I think **Hierarchial Clustering** will be good for this data set because it allows us to take clusters and merge them together in a hierarchy. We start with making specific clusters at the bottom of the hierarchy and go all the way up increasing in cluster

classification broadness. Although this method is slow in runtime, our data set is small enough to not have any big issues. The flexability of number of clusters and clear adaptable relationships we can set up between clusters makes this very applicable to this data set. One problem is that you cannot unmerge clusters.

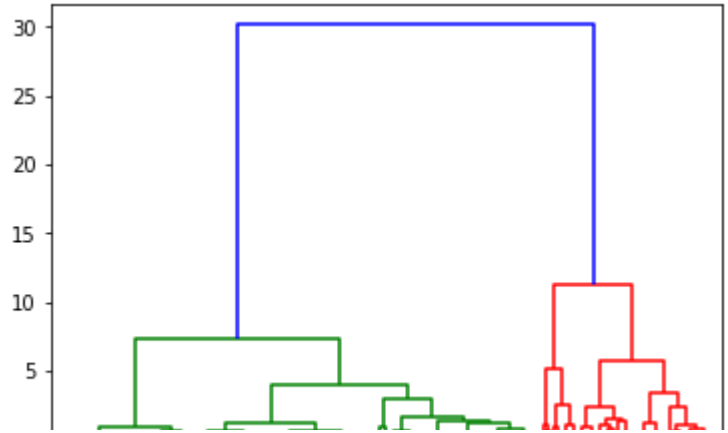
Lastly, **DBSCAN** would be good for this model but would honestly be over-built for this data set. Most the data is already in clear clusters and doesn't have much noise from a glance (which dbscan is good at filtering datasets with noise). DBSCAN would be good for this data in that it accurately defines oddly shaped clusters (non spherical) which reduces cluster variance.

DBSCAN is not effective when we have lots of predictors in our data set, or we have overlapping data that could be in more than one cluster (which I notice that this could be an issue with our data since there isnt much empty space between point, which could lead to inaccurate categorization).

Also it is not good for data sets with high variance in density (which we see in our graphs above). There is a lot of variance in the density of points for all 3 graphs, making dbscan not my top option for accuracy for this particular data set.

C)

```
# implementing the Hierarchial Clustering using all 3 variables
features = ["Sodium_100g", "Sugar_100g", "Total_Fat_100g"]
x = data[features]
z = StandardScaler()

x[features] = z.fit_transform(x) # z scoring

# ward allows for different variance clustering, distcance metric = ward
hac = AgglomerativeClustering(affinity = "euclidean", linkage = "ward")
hac.fit(x)

dendrogram = sch.dendrogram(sch.linkage(x, method = 'ward'))
```




```
hac = AgglomerativeClustering(n_clusters = 8, # gave me the best score
                              affinity = "euclidean",
                              linkage= "ward")
hac.fit(x)
membership = hac.labels_
membership
```

```
array([1, 3, 2, 5, 3, 1, 1, 5, 1, 2, 1, 1, 3, 3, 3, 1, 1, 5, 2, 3, 1, 3,
       1, 2, 1, 3, 5, 5, 5, 2, 1, 1, 6, 2, 1, 1, 6, 7, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
       7, 7, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       0, 0, 0, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 1, 4, 4, 4, 4, 4, 4,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 4, 4, 4, 0, 1, 1, 1, 1, 1, 5, 1, 7, 7, 7, 7, 7, 7,
       7, 1, 6, 1, 1, 5, 7, 7, 7, 1, 1, 1, 5, 2, 0, 0, 0, 0, 0, 0, 7, 7,
       7, 7, 7, 7, 4, 4, 4])
```

```
print("silhouette score: ")
silhouette_score(x,membership)
```

```
silhouette score:
0.8213204168016336
```

D)

Hierarchial Clustering analysis:

I tried several values for n_clusters and found that 3 produced the highest value for silhouette score. I tried 3, 7, **8**, 10, 12 and got values of 0.7331275230772667 , 0.7690558036164844, **0.8213204168016336**, 0.7858993750616131, 0.7777415850036854 respectively. (although the values change every time I run it, the numbers stay consistant around these values)

For a cluster value of **8** I got a silhouette score of 0.8213204168016336, which was the most accurate score out of all of the cluster values I tried. This surprised me because when I was looking at the graphs with a naked eye, I expected to see 3 with the highest score since I noticed 3 rough groups of data points, but I guess for a different model (such as GMM) it will be closer to that value. It is vary interesting to see how the algorithm can pick up clustering groups with much greater efficiency since the naked eye cannot be anywhere near as accurate when determining clustering based off data density variance.

Since when I noticed variance among density of points when initially looking at the graphs, I used affinity = "ward" which allows me to have more accurate clustering with points that have different variance by minimizing the variance of the merged clusters.

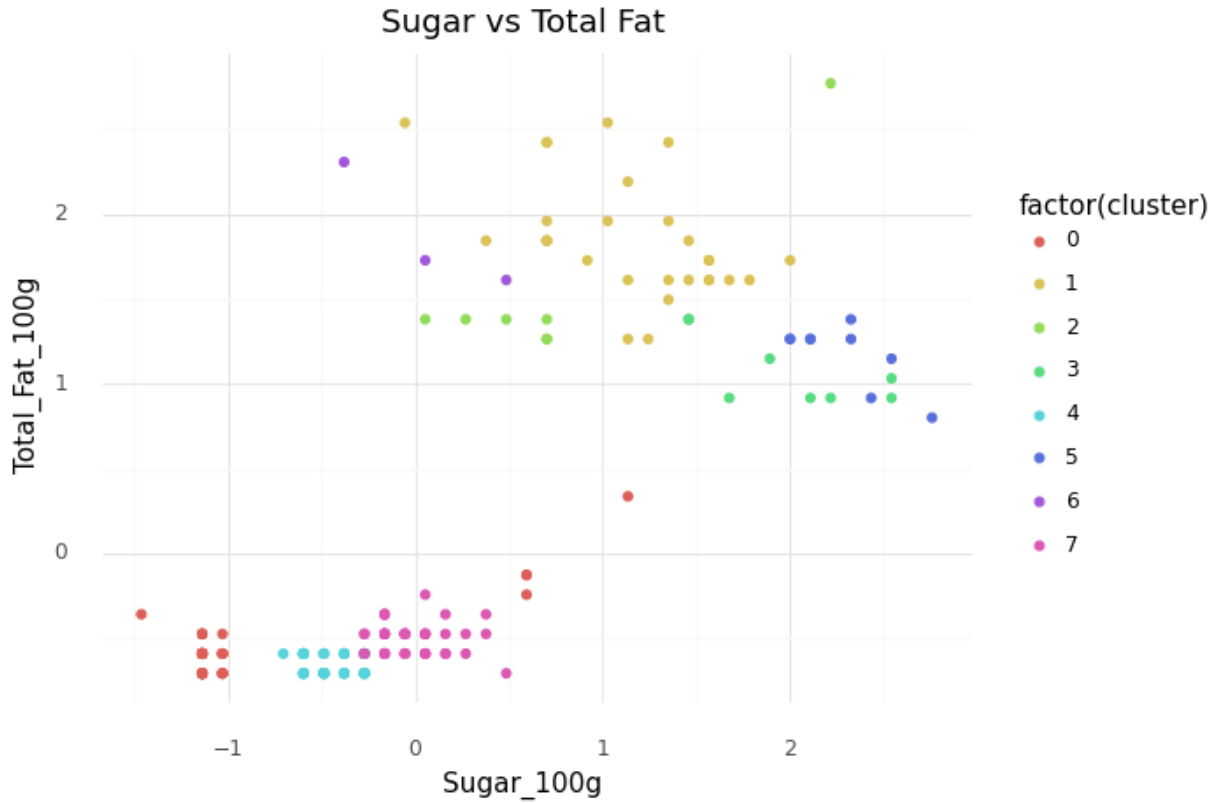
If my linkage is ward, only euclidean is accepted. Affinity = euclidean is associated with my linkage and is the only value that works when you use ward. Linkage being my distance metric between data points for clustering purposes.

Since my clustering model had a silhouette score of 0.8213204168016336 which tells me that the model did a good job of categorizing data in our set with the clusters set to 8. Most of the inaccuracy of my model would probably come from data that could go into one or more cluster group.

Re-making 3 graphs from part A with color clustering.

- describe what characteristics are in each cluster, give an example of a label for the cluster (e.g. "these donuts are low fat, and low sugar so I would call these healthy donuts") (IN A MARKDOWN CELL)

```
# re-making Sugar vs Total Fat
x["cluster"] = membership
(ggplot(x, aes(x = "Sugar_100g", y = "Total_Fat_100g", color = "factor(cluster)"))+ geom_point() +
 theme_minimal() + ggtitle("Sugar vs Total Fat"))
```




```
<ggplot: (8740423240477)>
```

Sugar vs Total Fat cluster analysis

- Anything at or around 0 on the x or y axis represents normal levels of sugar or fat compared to the rest of the values in the data set.
- Density of the donut is not a factor when analyzing fat/sugar content of the donut since the mass is constant. If something is more dense it will just be a smaller volume sample.
- Whenever I reference sugar and total fat, it is per 100 grams of the given donut.

CLUSTER 0 (orange):

This cluster is in the bottom left and is **densely packed** with all points having **very similar variance** with the exception of the one point on the left that is slightly further away. This cluster has **Lower than average** (around 0.5 stdv below mean) **total fat content** and **very low sugar levels** (less than 1 stdv below mean)

These points represent donuts with very little sugar (counterintuitive, ik) and very little fat compared to the rest of the donuts in this data set.

- This could be a group of bagels!

CLUSTER 1 (gold/yellow):

This cluster has much **higher variance** than cluster 0 which means the points are much more spread out. They all hover very high on the y axis, representing **high fat content**. They have slightly different lateral variance which indicates these donuts have different amounts of sugar than the others in the same cluster, but for the most part are **higher amounts** (around 1 stdv from the normal amount) **of sugar** amongst donuts in this data set.

These points could represent donuts that have icing (increasing the sugar content) and fried for a long period of time (increasing the fat).

- This group would most likely contain boston cream donuts or another filled donuts due to the amount of sugar and fat.

CLUSTER 2 (light green):

This cluster has **higher variance** than cluster 0, but less variance than cluster 1. It takes on a flatter, and longer elyptical shape with not much value fluctuation along the y axis (total fat). This cluster has **Slightly higher than average fat and sugar levels.**

The cluster lands in pretty similar values for both x and y, meaning it has similar amounts of fat and sugar content (these values are on the same scale since we z scored earlier when building the model)

- These points could represent a donut that was fried for a very specific amount of time, and it may or may not have icing or powdered sugar, depending on the value along the x- axis (sugar content).

CLUSTER 3 (teal):

This cluster has a similar shape to cluster 2, flatter and **medium amount of variance** (low variance looks like cluster 0, high variance looks like cluster 1)

This cluster has **High sugar** (2 stdv above mean) and **higher than average fat levels** (about 1 stdv above mean)

- These donuts could be glazed donuts!

CLUSTER 4 (light blue):

These points have **very low variance** and the cluster is located **very low** on the y-axis (**fat content**) and also **very low** along the x-axis(**sugar content**) but not as low as cluster 0.

These donuts are most likely fried for a very short amount of time (because of the low fat content) and also have a little bit more sugar than group 0.

- This cluster could be a group of very lightly powdered donuts!

CLUSTER 5 (dark blue):

These points have **medium variance** and have very similar horizontal and vertical variance. The cluster is located very far along the x-axis, indicating **high sugar content** in this cluster compared to the rest of the clusters in the data set. The cluster is in the **middle to slightly above average for fat content** (1 stdv above mean).

These donuts were fried for an average amount of time and have extra icing on them.

- This is a group of glazed and sprinkled donuts!

CLUSTER 6 (purple):

These points have **high variance** but the cluster is located high on the y-axis, meaning each donut has **lots of fat** comnparatively. The cluster is in the middle of the x-axis indicating **normal levels of sugar**.

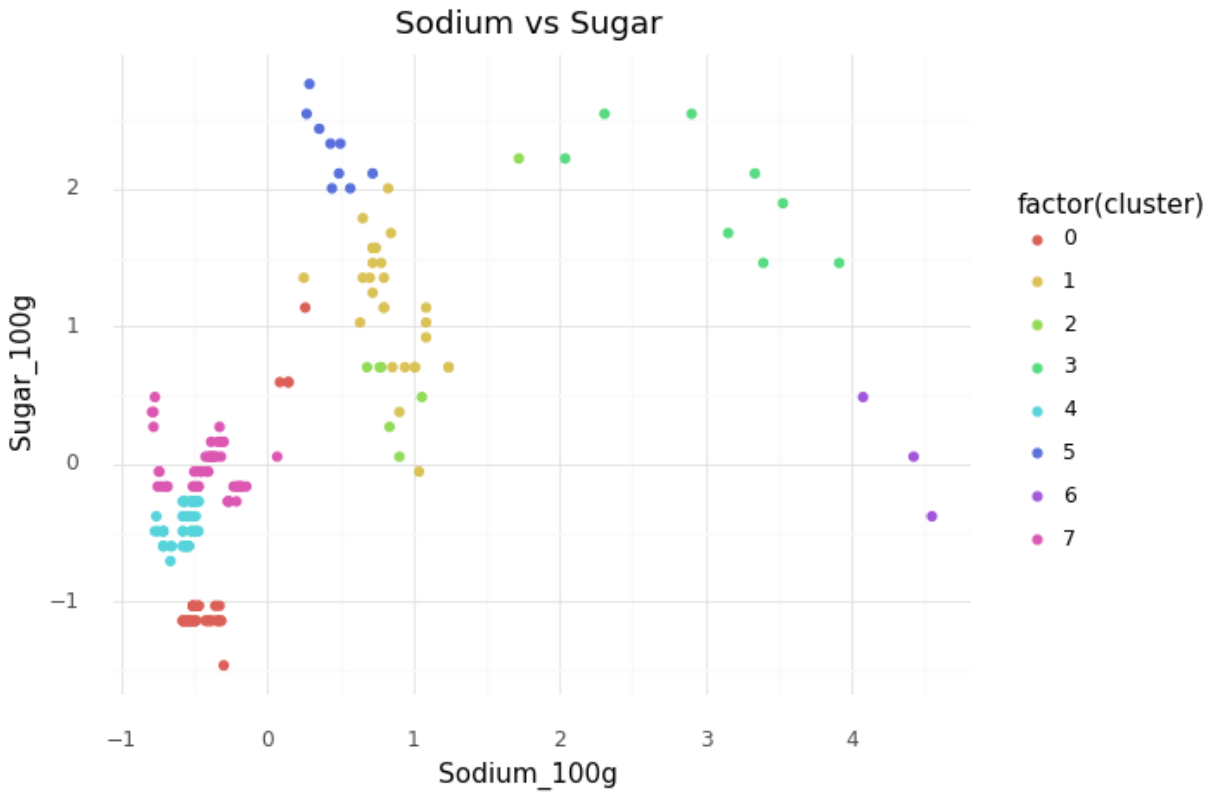
- These high fat, normal sugar donuts could be glazed old fasioned donuts!

CLUSTER 7 (pink):

These points have **very low variance** and are located in the **normal levels of sugar**, but **slightly below average levels of fat**.

- These donuts could be cinnamon sugar donuts that were not fried for long!

```
# re-making Sodium vs Sugar
x["cluster"] = membership
(ggplot(x, aes(x = "Sodium_100g", y = "Sugar_100g", color = "factor(cluster)"))+ geom_point() +
 theme_minimal() + ggtitle("Sodium vs Sugar"))
```




```
<ggplot: (8740423351145)>
```

Sodium vs Sugar graph analysis

- anything at or around 0 on the x or y axis represents normal levels of sugar or sodium compared to the rest of the values in the data set. Less than 0 indicates less than average sodium/sugar levels and more than 0 means more than average sodium/sugar levels.

CLUSTER 0 (orange):

This cluster is in the bottom left and is densely packed with all points having very similar variance with the exception of the one point below that is slightly further away and a couple that are higher in sugar but have similar sodium levels.

These points represent donuts with very little sugar and less sodium compared to the rest of the donuts in this data set.

Low soduim donut that might be glazed or not glazed! (based on position on y axis)

CLUSTER 1 (gold/yellow): This cluster has much higher variance than cluster 0. They all hover very high on the y axis, representing high sugar content. The cluster also sits around 1 on the x-axis representing higher than average sodium levels compared to the rest of the donuts in this data set.

These are donuts that ARE glazed and are not low sodium!

CLUSTER 2 (light green): These points have low variance and the cluster is located low on the y-axis indicating below average sugar levels and it is located low on the x-axis which means it has lower amounts of sodium compared to the average donut in this data set.

This cluster represents donuts that are slightly healthier than the average donut in this data set.

CLUSTER 3 (teal): These points have medium variance and the cluster is located high on the y-axis indicating high sugar levels and it is located high on the x-axis which means it has lots of sodium compared to the average donut in this data set.

This is an unhealthy donut with lots of sodium and sugar.

CLUSTER 4 (light blue): These points have low variance and the cluster is located low on the y-axis indicating low sugar levels and it is located low on the x-axis which means it has low amounts of sodium compared to the average donut in this data set.

This is a healthier donut than the average donut in this data set. Could be a low sodium and unglazed donut cluster.

CLUSTER 5 (dark blue): These points have low to medium variance and the cluster is located very high on the y-axis indicating high sugar levels and it is located higher than average on the x-axis which means it has a little more sodium than the average donut in this data set.

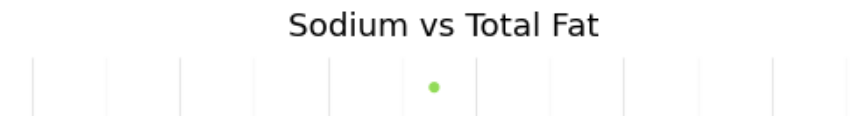
This is a glazed donut with more than average sodium levels.

CLUSTER 6 (purple): These points have medium variance and the cluster is located in the middle of the y-axis indicating normal sugar levels and it is located very high on the x-axis which means it has lots of sodium compared to the average donut in this data set.

This cluster represents unhealthy donuts that the chef accidentally spilled salt into the dough.

CLUSTER 7 (pink): These points have low variance and the cluster is located in the middle of the y-axis (around 0) indicating normal sugar levels and it is located low on the x-axis which means it has lower amounts of sodium compared to the average donut in this data set.

This is a cluster of your typical glazed donut.



Sodium vs Total Fat graph analysis

- anything at or around 0 on the x or y axis represents normal levels of sodium or total fat compared to the rest of the values in the data set. Less than 0 indicates less than average sodium/fat levels and more than 0 means more than average sodium/fat levels.

.

CLUSTER 0 (orange):

These points have **low variance** and the cluster is located below the middle of the y-axis (around 0) indicating **slightly less than normal fat levels** and it is located low to mid (around 0) on the x-axis which means it has **lower to normal amounts of sodium** compared to the average donut in this data set.

- This is a cluster of your typical glazed donuts with slightly less than average levels of total fat and sodium levels.

CLUSTER 1 (gold/yellow): These points have **medium variance** and the cluster is located high on the the y-axis (above 0, and almost at 2 standard deviations above the mean) indicating **more than normal fat levels** and it is located mid to high (slightly above 0) on the x-axis which means it has **slightly higher amounts of sodium** compared to the average donut in this data set.

- This is a cluster of donuts fried for longer than normal and dough made with more salt.

CLUSTER 2 (light green): These points have **low variance** and the cluster is located higher than average on the the y-axis (above 0, around 1 standard deviations above the mean) indicating **more than normal fat levels** and it is located higher (around 1) on the x-axis which means it has **slightly higher amounts of sodium** compared to the average donut in this data set.

- This is a cluster of donuts with slightly more than average levels of total fat and sodium levels.

CLUSTER 3 (teal): These points have **medium to high variance** and the cluster is located higher than average on the the y-axis (above 0, around 1 standard deviations above the mean) indicating **more than normal fat levels** and it is located higher (around 3 (standard deviations above mean)) on the x-axis which means it has **high amounts of sodium** compared to the average donut in this data set.

- This cluster represents donuts fried for a long time in sodium rich oil.

CLUSTER 4 (light blue): These points have **low variance** and the cluster is located lower than average on the the y-axis (around 0.8 standard deviations below the mean) indicating **less than normal fat levels** and it is located lower (less than 0, close to -1) on the x-axis which means it has **low amounts of sodium** compared to the average donut in this data set.

- This cluster represents donuts fried for a short period of time in oil with low sodium content.

CLUSTER 5 (dark blue): These points have **low to medium variance** and the cluster is located higher than average on the the y-axis (around 1 standard deviations above the mean) indicating **more than normal fat levels** and it is located in the middle (slightly above 0) on the x-axis which means it has **normal** to **slightly higher** than normal **amounts of sodium** compared to the average donut in this data set.

- This cluster represents donuts fried for a slightly longer than normal in oil with normal sodium content.

CLUSTER 6 (purple): These points have **medium to high variance** and the cluster is located higher on the y-axis (around 2 standard deviations above the mean) indicating **high fat levels** and it is located in the high (around 4 standard deviations of the mean) on the x-axis which means it has **very high amounts of sodium** compared to the average donut in this data set.

- This cluster represents donuts fried for longer than normal in sodium rich oil.

CLUSTER 7 (pink): These points has **low variance** and the cluster is located low on the the y-axis (around 0.5 standard deviations below the mean) indicating **lower fat levels** and it is located low (around 0.5 standard deviations below the mean) on the x-axis which means it has **lower amounts of sodium** compared to the average donut in this data set.

- This cluster represents donuts fried for a short period of time in oil with low sodium content.

E)

```
# checking other _100g variables
data.count(0) #counts non na columns

     Restaurant_Item_Name      205
     restaurant                205
```

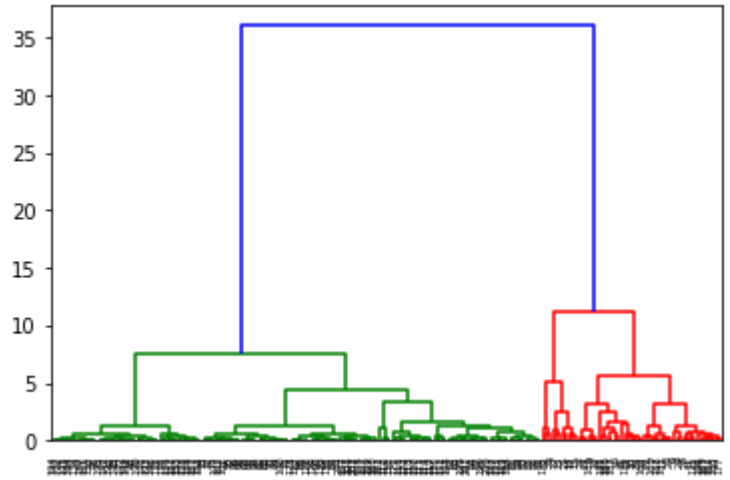
```
Restaurant_ID             205
Item_Name                 205
Item_Description          205
Food_Category             205
Serving_Size              205
Serving_Size_text           0
Serving_Size_Unit         205
Serving_Size_household      7
Calories                  205
Total_Fat                 205
Saturated_Fat             205
Trans_Fat                 205
Cholesterol               205
Sodium                    205
Potassium                  39
Carbohydrates             205
Protein                   205
Sugar                     205
Dietary_Fiber             174
Calories_100g             205
Total_Fat_100g            205
Saturated_Fat_100g        205
Trans_Fat_100g            205
Cholesterol_100g          205
Sodium_100g               205
Potassium_100g             39
Carbohydrates_100g        205
Protein_100g              205
Sugar_100g                205
Dietary_Fiber_100g        174
dtype: int64
```

I picked the variable **Calories_100g** to include in my model because it is a value that is commonly looked at by consumers when it comes to consumer nutrition. It also has the same count value as my previous predictors which makes it very applicable in terms of number of data values.

It will be interesting to see if my Heirarchial model can categorize another predictor that is a bit more general of a variable, meaning it isn't a specific as Total_Fat100g, Sodium_100g, and Sugar_100g are. These previous predictors I used measure one value (fat,

sodium, and sugar content) whereas Calories_100g takes all nutritional values (like fat, sodium, sugar, protein, saturated fat,

```
# adding in calories_100g
features = ["Sodium_100g", "Sugar_100g", "Total_Fat_100g", "Calories_100g"]
x = data[features]
z = StandardScaler()

x[features] = z.fit_transform(x) # z scoring

# ward allows for different variance clustering, distcance metric = ward
hac = AgglomerativeClustering(affinity = "euclidean", linkage = "ward")
hac.fit(x)

dendrogram = sch.dendrogram(sch.linkage(x, method = 'ward'))
```




```
hac = AgglomerativeClustering(n_clusters = 8, # gave me the best score
                              affinity = "euclidean",
                              linkage= "ward")
hac.fit(x)
membership = hac.labels_
membership
```

```
array([0, 4, 1, 0, 4, 1, 0, 0, 0, 1, 1, 0, 4, 4, 4, 1, 0, 0, 1, 4, 0, 4,
       0, 1, 0, 4, 0, 0, 0, 1, 1, 1, 5, 1, 1, 1, 5, 2, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
       6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
       7, 7, 7, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 6, 6, 6, 6, 6, 6,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 6, 6, 6, 7, 1, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2,
       2, 0, 5, 1, 1, 0, 2, 2, 2, 1, 1, 1, 0, 1, 3, 3, 3, 3, 3, 3, 2, 2,
       2, 2, 2, 2, 6, 6, 6])
```

```
print("silhouette score: ")
silhouette_score(x,membership)
```

```
silhouette score:
0.5147938480463081
```

Analysis of model with the new predictor:

With the addition of the predictor "Calories_100g", my *exact* model became a lot **less accurate** in terms of silhouette score (dropping from about 0.821 to 0.515). This was with the exact same amount of clusters I used before and all the same.

I messed around changing the amount of clusters with my new model to see if I could replicate the accuracy of my first model in this new model, and the highest silhouette score I recorded was 0.7565603870763944 with 3 numbers of clusters.

This makes sense that the score is lower (for both my exact same model and my one with n_clusters = 3) since I am adding more parameters, my model will naturally become less accurate (with this model set-up) in clustering since there is more variables to satisfy a data point (this increases probability of data points overlapping and falling into multiple clusters which reduces score).