Project 2

GENERAL INSTRUCTIONS:

this is NOT a group project

- **CLEARLY mark where you are answering each question (all written questions must be answered in Markdown cells, NOT as comments in code cells)**
- **Show all code necessary for the analysis, but remove superfluous code**

DONUTS

Using the dataset *krispykreme.csv*,

- **a) make 3 scatterplots using ggplot to show:**
  - `Sodium_100g` **vs** `Total_Fat_100g`
  - `Sodium_100g` **vs.** `Sugar_100g`
  - `Sugar_100g` **vs** `Total_Fat_100g`

- **b) Using the scatterplots from part a as well as the donuts dataset, thouroughly discuss which clustering method (KMeans, Gaussian Mixture Models (EM), Hierarchical Clustering, or DBSCAN) you think would be best for this data and WHY. Be sure to include discussions of assumptions each algorithm does/does not make, and what types of data they are good/bad for (mention each of the 4 algorithms at least once). (*IN A MARKDOWN CELL*)**

Please note that for this assignment, "It's easier to code" or "it's comuptationally efficient" does not count as a valid reason. The reasons should be based on the algorithms/data.

(Please use "" to make any mention of one of the algorithms bold in your discussion. For example "I think **DBSCAN** is the best algorithm ever!" will make the word "DBSCAN" bold in a Markdown cell).**

- **c) Implement the algorithm you think will work best here using the 3 variables `Sodium_100g`, `Total_Fat_100g` and `Sugar_100g`, and describe how you chose any hyperparameters (such as distance, # of clusters, min_samples, eps, linkage...etc). Make sure to z-score your variables. (*IN A MARKDOWN CELL*)**

- **d) Thouroughly discuss the performance of your clustering model.**
  - **which metric did you use to asses your model? (*IN A MARKDOWN CELL*)**
  - **how did your model perform? (*IN A MARKDOWN CELL*)**
  - **remake the 3 graphs from part a, but color by cluster assignment. Describe what characterizes each cluster, and give an example of a label for that cluster (e.g. "these donuts are low fat, and low sugar so I would call these healthy donuts") (*IN A MARKDOWN CELL*)**

- **e) Choose ONE other of the `_100g` variables from the data set to add to your clustering model to improve it.**
  - **explain why you chose this variable. (*IN A MARKDOWN CELL*)**
  - **make a new model, identical to the model in part c, but also including your new variable.**
  - **did this variable improve the fit of your clustering model? How can you tell? (*IN A MARKDOWN CELL*)**

Note: The columns with `_100g` at the end represent the amount of that nutrient per 100 grams of the food. For example, `Total_Fat` tells you the total amount of fat in that food, whereas `Total_Fat_100g` tells you how much fat there is per 100 grams of that food.

In [1]:

```
# import necessary packages
import warnings
```

```
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
from plotnine import *

from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import NearestNeighbors

from sklearn.cluster import DBSCAN


from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
import scipy.cluster.hierarchy as sch
from matplotlib import pyplot as plt
from sklearn.cluster import AgglomerativeClustering

from sklearn.metrics import silhouette_score

%matplotlib inline
```

In [2]:

```
doughnuts = pd.read_csv("https://raw.githubusercontent.com/cmparlettpelleriti/CPSC392ParlettPelleriti/master/Data/KrispyKreme.csv")
doughnuts.head()
```

Out[2]:

	Restaurant_Item_Name	restaurant	Restaurant_ID	Item_Name	Item_Description	Food_Category	Serving_Size	Serving_Size
0	Krispy Kreme Apple Fritter	Krispy Kreme	49	Apple Fritter	Apple Fritter, Doughnuts	Baked Goods	100	
1	Krispy Kreme Chocolate Iced Cake Doughnut	Krispy Kreme	49	Chocolate Iced Cake Doughnut	Chocolate Iced Cake Doughnut, Doughnuts	Baked Goods	71	
2	Krispy Kreme Chocolate Iced Custard Filled Dou...	Krispy Kreme	49	Chocolate Iced Custard Filled Doughnut	Chocolate Iced Custard Filled Doughnut, Doughnuts	Baked Goods	85	
3	Krispy Kreme Chocolate Iced Glazed Doughnut	Krispy Kreme	49	Chocolate Iced Glazed Doughnut	Chocolate Iced Glazed Doughnut, Doughnuts	Baked Goods	63	
4	Krispy Kreme Chocolate Iced Glazed Cruller Dou...	Krispy Kreme	49	Chocolate Iced Glazed Cruller Doughnut	Chocolate Iced Glazed Cruller Doughnut, Doughnuts	Baked Goods	70	

5 rows × 32 columns

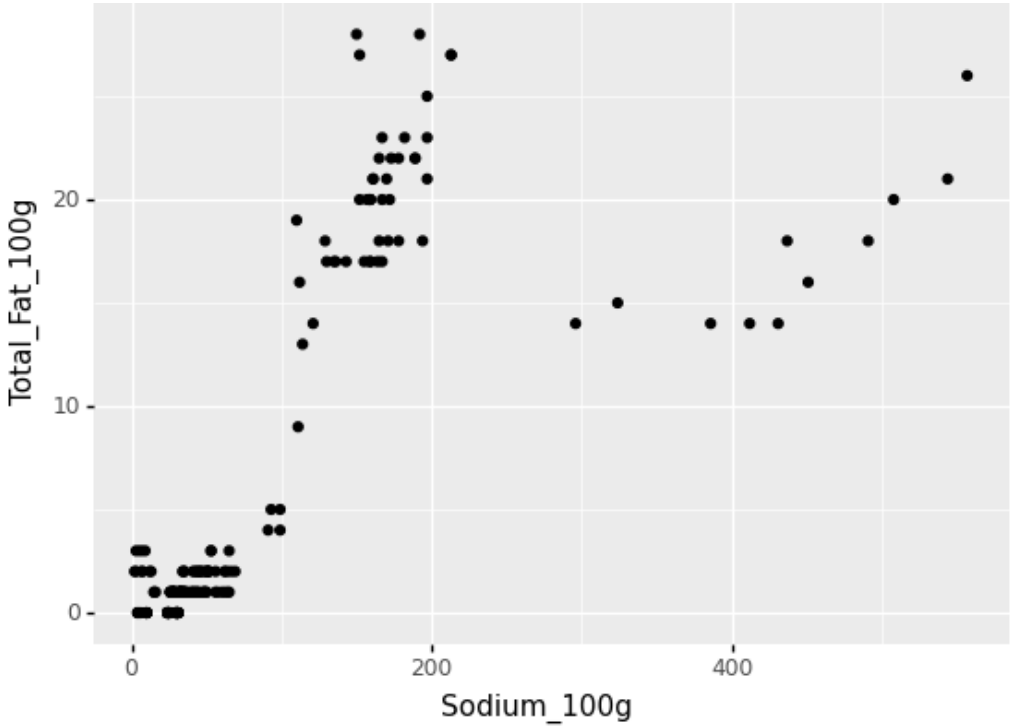
In [3]:

```
#a
features = ["Sodium_100g","Total_Fat_100g", "Sugar_100g"]
X = doughnuts[features]

z = StandardScaler()
z = z.fit_transform(X)
```

In [5]:

```
(ggplot(X, aes("Sodium_100g", "Total_Fat_100g")) + geom_point())
```

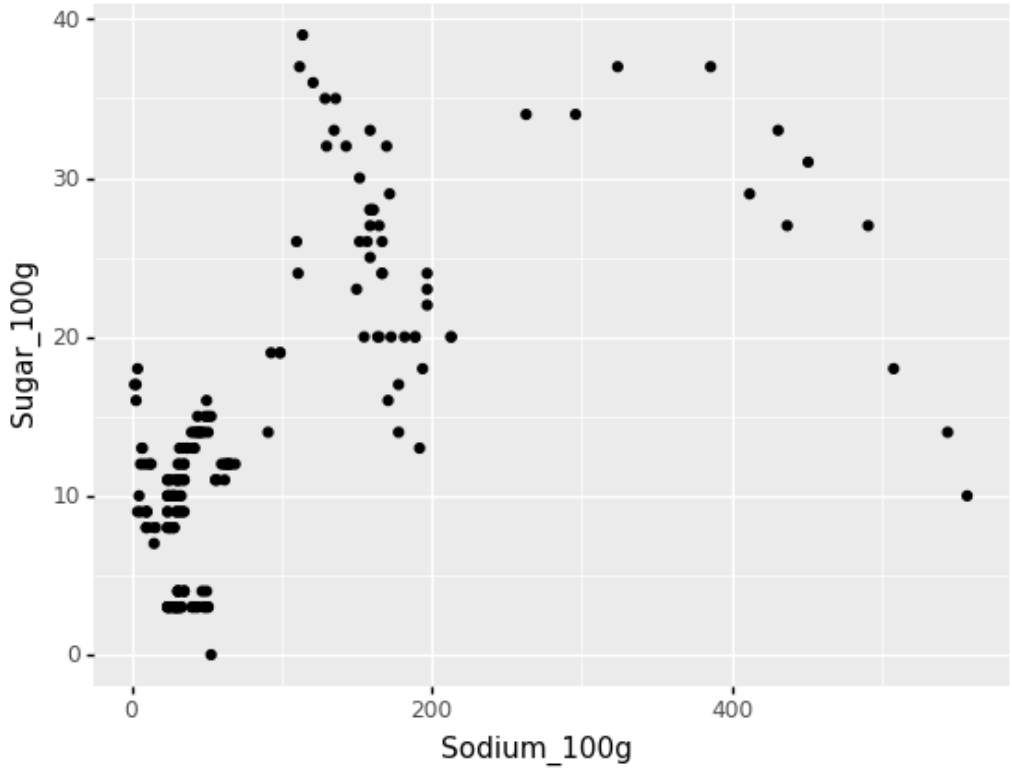


Out[5]:

```
<ggplot: (8787717355813)>
```

In [6]:

```
(ggplot(X, aes("Sodium_100g", "Sugar_100g")) + geom_point())
```

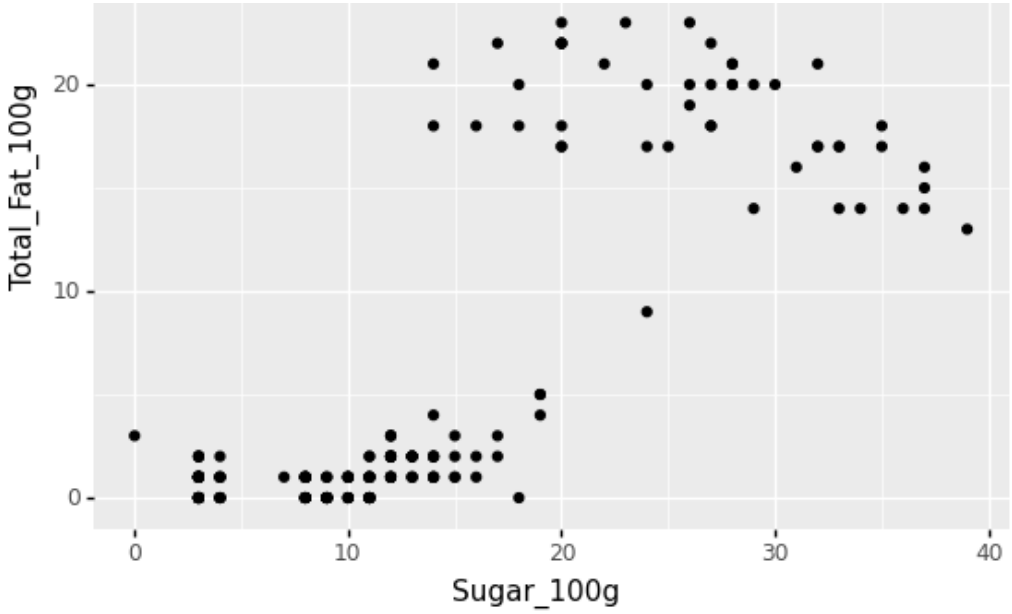


Out[6]:

```
<ggplot: (8787717656055)>
```

In [7]:

```
(ggplot(X, aes("Sugar_100g", "Total_Fat_100g")) + geom_point())
```



Out[7]:

<ggplot: (8787717690170)>

b) Using the scatterplots from part a as well as the donuts dataset, thouroughly discuss which clustering method (KMeans, Gaussian Mixture Models (EM), Hierarchical Clustering, or DBSCAN) you think would be best for this data and WHY. Be sure to include discussions of assumptions each algorithm does/does not make, and what types of data they are good/bad for (mention each of the 4 algorithms at least once). (IN A MARKDOWN CELL)

Kmeans is not the right algorithm for this data set because this algorithm is a hard assignment and all the variance is the same. This algorithm assumes the spherical assumption, but we do not want that since there is alot of variance througout the data set. **DBSCAN** is not the correct algorithm because it fails to identify cluster if the density varies and if the data set is sparse. As seen by the three graphs above, the data points are far apart from each other and there density varies, so DBSCAN would not do a good job of clustering the data paoints. The clusters will have and the graphs will show low cohesion and separation. We want high cohesion and separation. High cohesion shows that the data points in a cluster are close to each other. High separation shows that the clusters are far away from each other. Finally, **Hierarchical Clustering** is not the correct algorithm because the distance between the data points are very different, none of the linkage clustering methods work very well with with data. Single Linkage does not work because is sensitive to noise and outleirs and cannot group clusters properly if there is any noise between the clusters. As seen in the three graphs above there are some noise data points between the clusters, so they would not be clustered correctly. Complete and Average linkage would not work well with this data set because these two linkage techniques tend to break large clusters and are biased towards globular clusters. In all three graphs, you can tell that there is globular structure, so these two method would not work.

The **Gaussian Mixture Models (EM)** is the correct algorithm for this dataset because this algorithm is a soft assignment and the data points in the graphs variance is different. Hard assignment is when each data point gets put into a separate cluster. Soft assignment is when the algorith gives a probability of that data point to be in those clusters is assigned. As seen in the three graphs above, there are no spherical clusters, there are more elliptical shaped clusters. EM estimates the variance is so instead of spherical clusters it allows us to have other shapes of clusters.

In [17]:

```
# c Implement the algorithm you think will work best here using the 3 variables Sodium_10
0g, Total_Fat_100g and Sugar_100g,
features2 = ["Sodium_100g","Total_Fat_100g", "Sugar_100g"]
X2 = doughnuts[features2]
z2 = StandardScaler()
z2 = z2.fit_transform(X)
EM = GaussianMixture(n_components = 3)
EM.fit(X2)
cluster = EM.predict(X2)
X2["cluster"] = cluster
silhouette_score(X2, cluster)
```

Out[17]:

Out[17]:

```
0.7952703020805632
```

c) Describe how you chose any hyperparameters (such as distance, # of clusters, min_samples, eps, linkage...etc). Make sure to z-score your variables. (IN A MARKDOWN CELL)

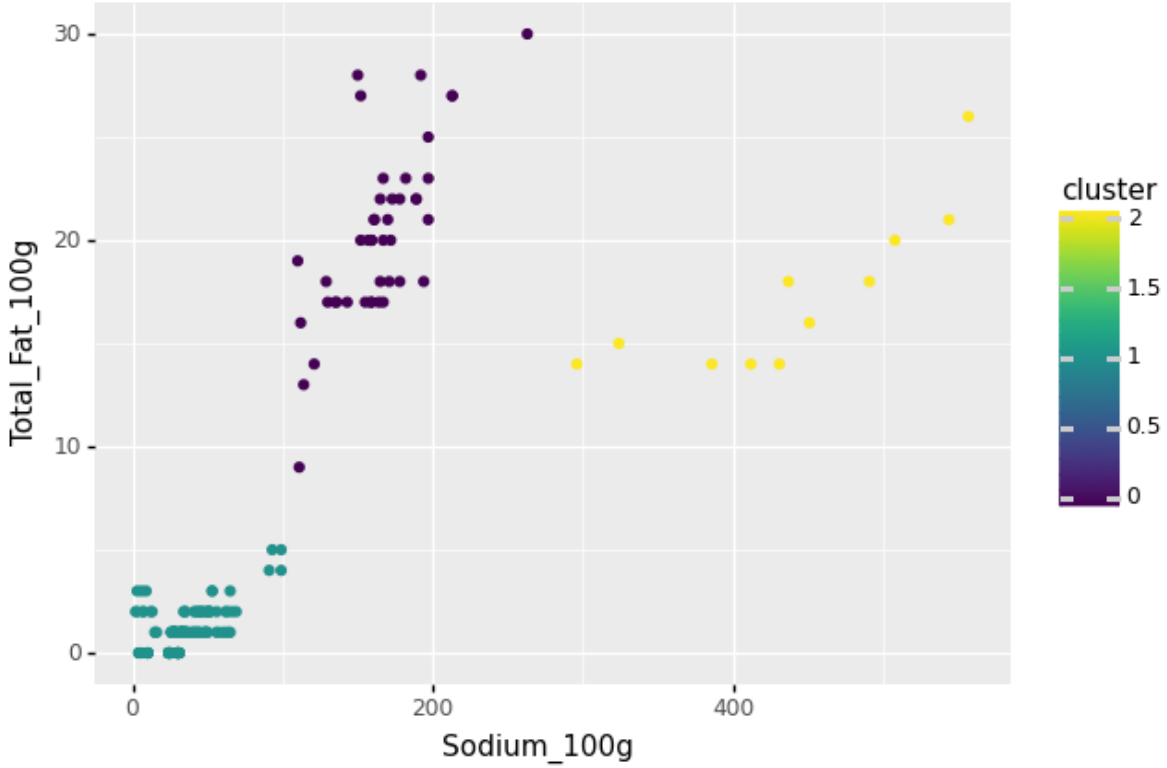
I chose the number of clusters by changing the number of "n_components". With two clusters, the silhouette score was .74. With four clusters, the silhouette score was .78. With three clusters, the silhouette score is .79. .79 is the higest silhouette score out of three, so that is why I chose three as the number of clusters. I knew that it had to be a low number since as the number of clusters increases, the average variance decreases. We cluster the data to find similarities among the categories. The smaller number of the clusters is better to identify similarities. It is also better the number of clusters to be low to avoid data leakage.

d) Which metric did you use to asses your model? How did your model perform? (IN A MARKDOWN CELL)

I used the silhouette score metric to asses my model. We want a high silhouette score. To have a high silohouette score, we need to have high cohesion and separation. Having high cohesion means the data points in the clusters are close to each other. Having high separation means that the clusters are far from each other. When a silhouette score is postive/above zero and close to one, that means that there is high cohesion and separation. This means that the data points in each cluster are very close to each other (high cohesion) and the clusters are far from each other (high separation). This is what we want. When a silhouette score is negative/below zero and close to zero, that means there is low cohesion and separation.This means that the data points in each cluster are far from each other (low cohesion) and the clusters are very close to each other (low separation). The silhouette score for this model is .79, it is positive and close to one. This means the model has high cohesion and separation.

In [13]:

```
#d) Remake the 3 graphs from part a, but color by cluster assignment.
(ggplot(X, aes("Sodium_100g", "Total_Fat_100g", color = "cluster")) + geom_point())
```

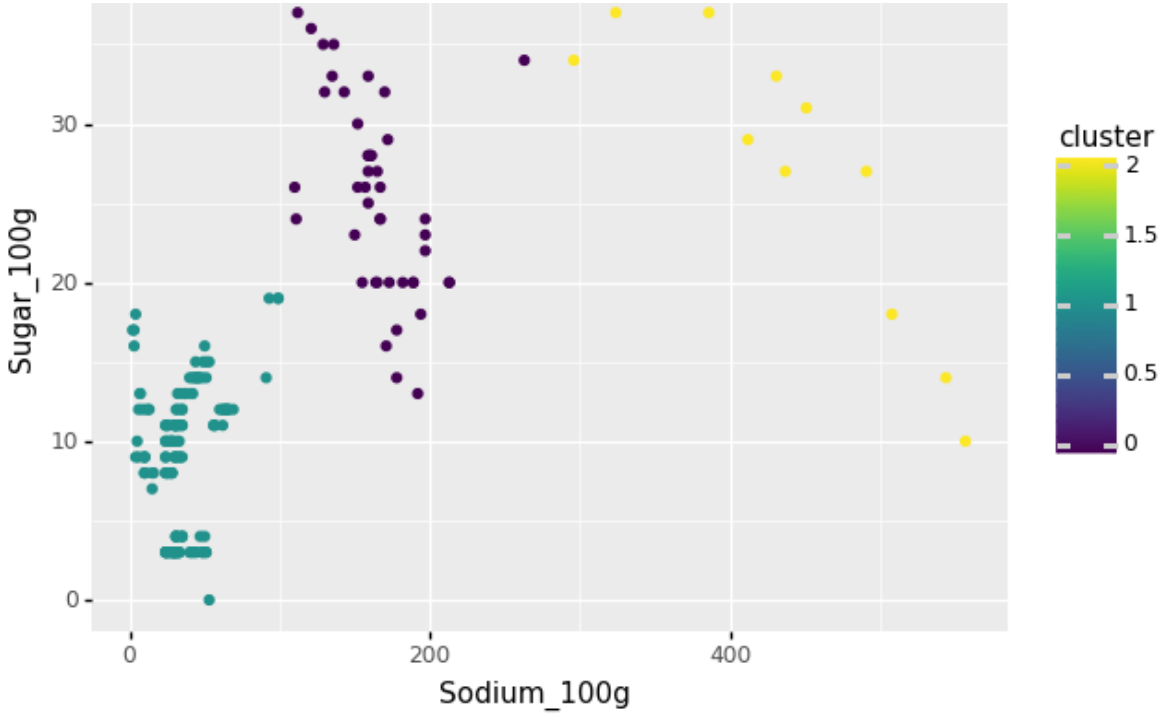


Out[13]:

```
<ggplot: (8774112939517)>
```

In [14]:

```
(ggplot(X, aes("Sodium_100g", "Sugar_100g", color = "cluster")) + geom_point())
```

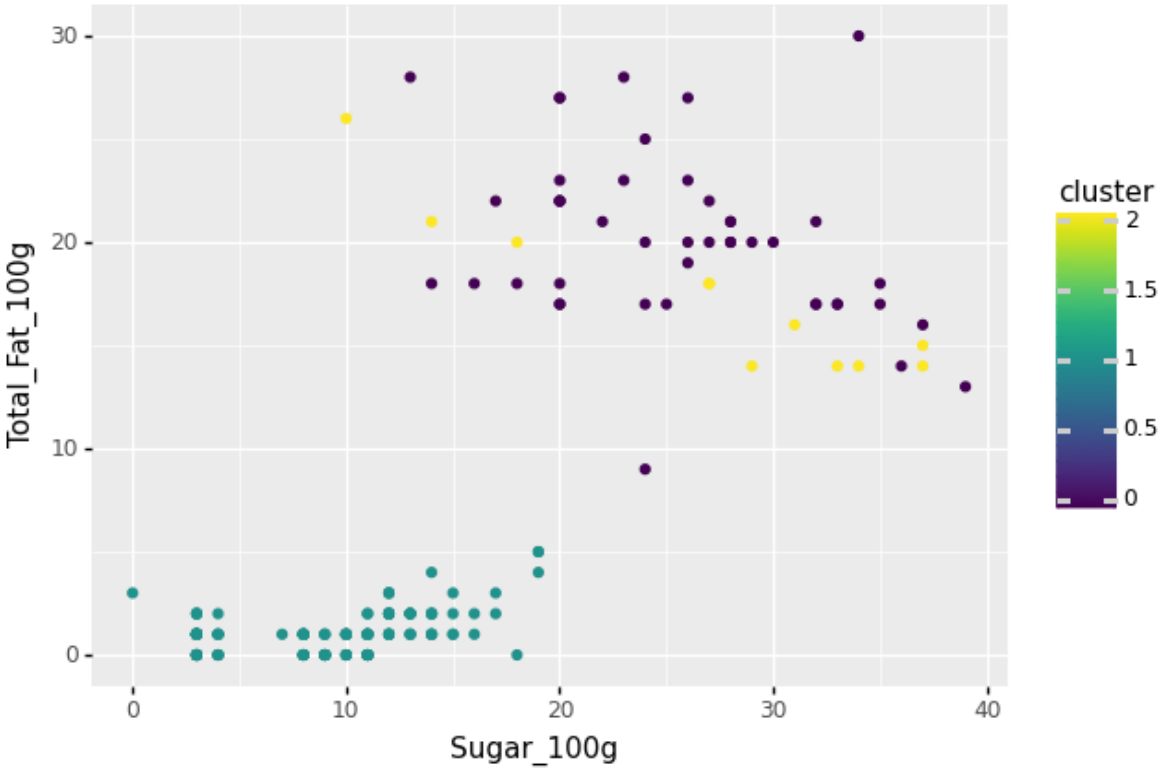


Out[14]:

<ggplot: (8774107312656)>

In [15]:

```
(ggplot(X, aes("Sugar_100g", "Total_Fat_100g", color = "cluster")) + geom_point())
```



Out[15]:

<ggplot: (8774114052618)>

d) Describe what characterizes each cluster, and give an example of a label for that cluster (e.g. "these donuts are low fat, and low sugar so I would call these healthy donuts") (IN A MARKDOWN CELL)

Sodium_100g vs Total Fat_100g (first graph)

- **The blue cluster's data points are in the low sodium and fat area, so these doughnuts are healthy**
- **The purple cluster's data points are in the low sodium and high fat area, so these doughnuts are not the healthiest**
- **The yellow cluster's data points are in the high sodium and fat area, so these doughnuts are not healthy at**

- **The yellow cluster's data points are in the high sodium and fat area, so these doughnuts are not healthy at all**

Sodium_100_g vs Sugar_100g (second graph)

- **The blue cluster's data points are in the low sodium and low-mid level of sugar area, so these doughnuts are not the healthiest**
- **The purple cluster's data points are in the low sodium and high sugar area, so these doughnuts are not healthy**
- **The yellow cluster's data points are in the high sodium and sugar area, so these doughnuts are the least healthiest doughnuts**

Sugar_100g vs Total_Fat_100g (third graph)

- **The blue cluster's data points are in the low sugar and fat area, so these doughnuts are healthy**
- **The purple and yellow clusters's data points are mixed with each other. The two clusters show high sugar and fat, so the doughnuts in both these clusters are extremely unhealthy.**

In [20]:

```
#e) make a new model, identical to the model in part c, but also including your new varia
ble.
features2 = ["Sodium_100g","Total_Fat_100g", "Sugar_100g", "Cholesterol_100g"]
X = doughnuts[features2]
z = StandardScaler()
z = z.fit_transform(X)
EM = GaussianMixture(n_components = 3)
EM.fit(X)
cluster2 = EM.predict(X)
X["cluster2"] = cluster2
silhouette_score(X, cluster2)
```

Out[20]:

```
0.7927805296764683
```

e) Explain why you chose this variable. Did this variable improve the fit of your clustering model? How can you tell? (IN A MARKDOWN CELL)

I chose this variable because cholestrol is another health related variable. I wanted to see the "healthiness" of doughnuts. No, adding a new variable did not improve the fit of my clustering model. In part C, the silohouette score is .795 while in part E, the silohoutte score is .792. When a silhouette score is postive/above zero and close to one, that means that there is high cohesion and separation. This means that the data points in each cluster are very close to each other and the clusters are far from each other. This is what we want. The silhouette score for this model is .792, it is positive and close to one. This means the model has high cohesion and separation. However this variable did not improved the fit of the model since the silhouette score dropped from .795 to .792, which is a .003 decrease.