```
import warnings
warnings.filterwarnings('ignore')


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

from sklearn.preprocessing import StandardScaler #Z-score variables

from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture

from sklearn.metrics import silhouette_score, silhouette_samples

%matplotlib inline
```

```
bk = pd.read_csv("https://raw.githubusercontent.com/cmparlettpelleriti/CPSC392ParlettPelleriti/master/Data/burger-king-items.txt",
                sep='\t')

bk.head()
```

	Item	Serving.size	Calories	Fat.Cal	Protein(g)	Fat(g)	Sat.Fat(g)	Trans.fat(g)	Chol(mg)	Sodium(mg)
0	Hamburger	109.0	260	90.0	13	10.0	4.0	0.0	35	490
1	Cheeseburger	121.0	300	130.0	16	14.0	6.0	0.0	45	710
2	Double_Hamburger	146.0	360	160.0	22	18.0	8.0	0.0	70	520
3	Double_Cheeseburger	171.0	450	230.0	26	26.0	12.0	1.0	95	960
4	Buck_Double	158.0	410	200.0	24	22.0	10.0	0.5	85	740

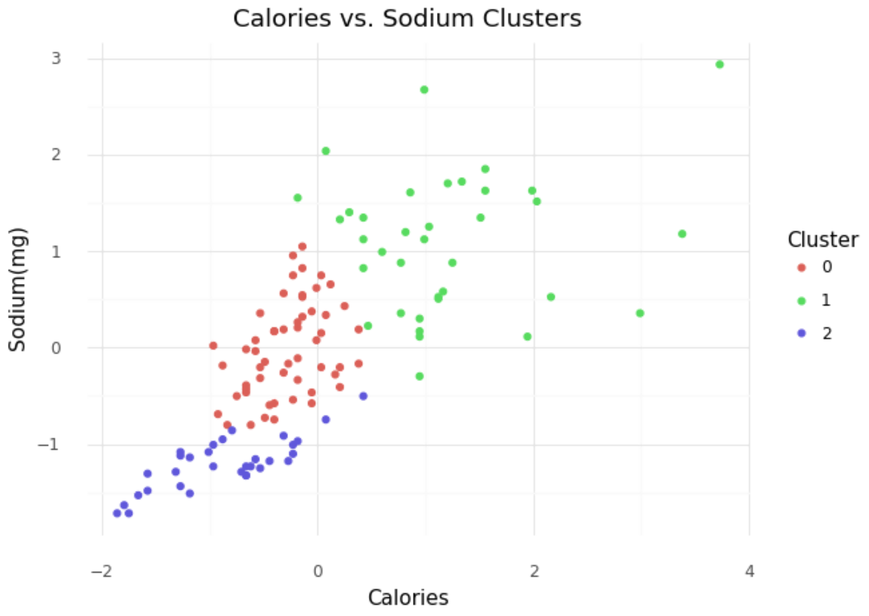
```
# single GMM
X = bk[["Calories", "Sodium(mg)"]]

z = StandardScaler()
X[["Calories", "Sodium(mg)"]] = z.fit_transform(X)

gmm = GaussianMixture(n_components = 3)
gmm.fit(X)

X["cluster"] = gmm.predict(X)

(ggplot(X, aes(x = "Calories", y = "Sodium(mg)", color = "factor(cluster)")) + geom_point() +
 theme_minimal() + labs(title = "Calories vs. Sodium Clusters") +
 scale_color_discrete(name = "Cluster"))
```



```
<ggplot: (8731392970599)>
```

Multiple Dimensions

```
# sugar protein fat
features = ["Sugar(g)", "Protein(g)", "Fat(g)"]

X = bk[features]

z = StandardScaler()
X[features] = z.fit_transform(X[features])
```

```
# model
gmm = GaussianMixture(n_components = 3)
gmm.fit(X[features])
```

```
▾        GaussianMixture
GaussianMixture(n_components=3)
```

```
# predictions
cluster = gmm.predict(X[features])
cluster
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 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,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1])
```

```
# sil
silhouette_score(X[features], cluster)
```

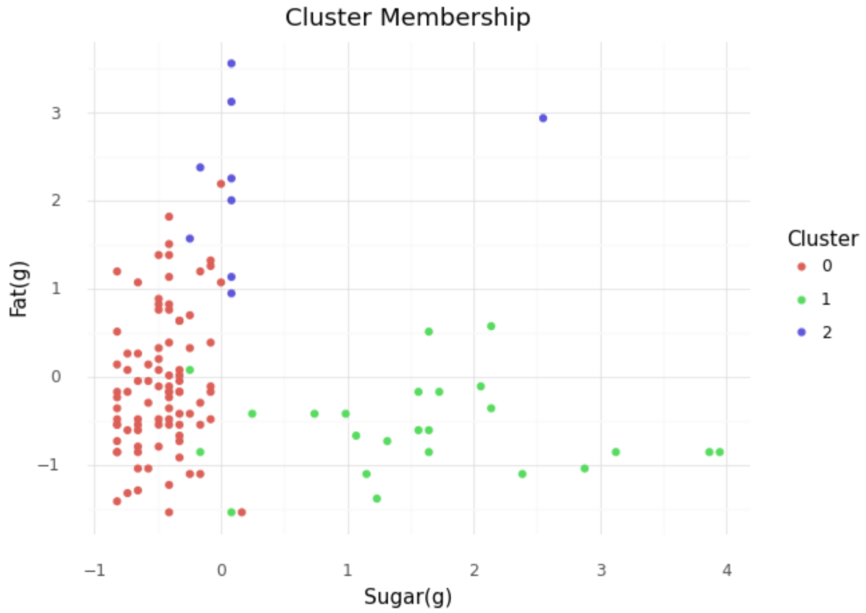
```
0.41269446652563374
```

```
# probabilities
gmm.predict_proba(X[features])
```

```
       [3.47702831e-017, 1.00000000e+000, 4.95858193e-016],
       [5.71207524e-033, 1.00000000e+000, 3.49702092e-086],
       [5.27520202e-020, 1.00000000e+000, 3.81787443e-087],
       [1.04101950e-054, 1.00000000e+000, 8.42596148e-236],
       [5.12977701e-073, 1.00000000e+000, 8.68560032e-310],
       [1.64241012e-030, 1.00000000e+000, 2.01447666e-059],
       [4.77551251e-024, 1.00000000e+000, 5.27405603e-060],
       [9.28935335e-001, 7.10646648e-002, 2.56688884e-085],
       [2.05639557e-018, 1.00000000e+000, 1.86891823e-165],
       [3.44131978e-082, 1.00000000e+000, 0.00000000e+000],
       [5.68247232e-122, 1.00000000e+000, 0.00000000e+000],
       [1.19723118e-117, 1.00000000e+000, 0.00000000e+000]])
```

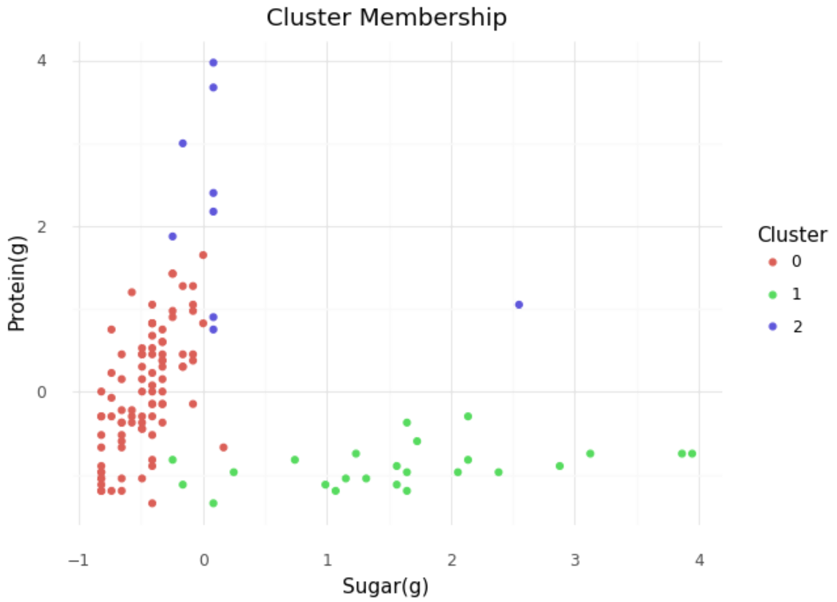
```
# graph
X["cluster"] = cluster

(ggplot(X, aes(x = "Sugar(g)", y = "Fat(g)", color = "factor(cluster)")) + geom_point() +
theme_minimal() + labs(title = "Cluster Membership") + scale_color_discrete(name ="Cluster"))
```



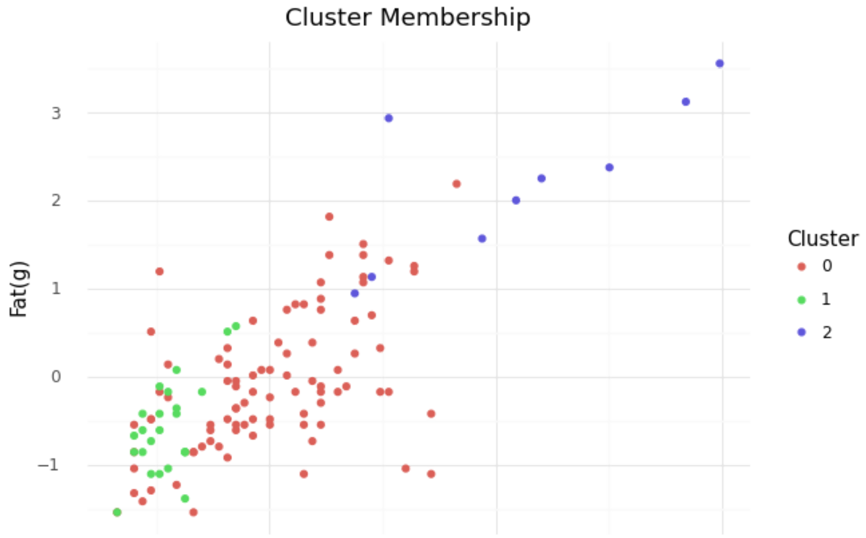
```
<ggplot: (8731328348599)>
```

```
(ggplot(X, aes(x = "Sugar(g)", y = "Protein(g)", color = "factor(cluster)")) + geom_point() +
theme_minimal() + labs(title = "Cluster Membership") + scale_color_discrete(name ="Cluster"))
```



```
<ggplot: (8731322565029)>
```

```
(ggplot(X, aes(x = "Protein(g)", y = "Fat(g)", color = "factor(cluster)")) + geom_point() +
theme_minimal() + labs(title = "Cluster Membership") + scale_color_discrete(name ="Cluster"))
```



```
<ggplot: (8731322502211)>
```

✓ 1s completed at 00:41