

```
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

from sklearn.metrics import silhouette_score

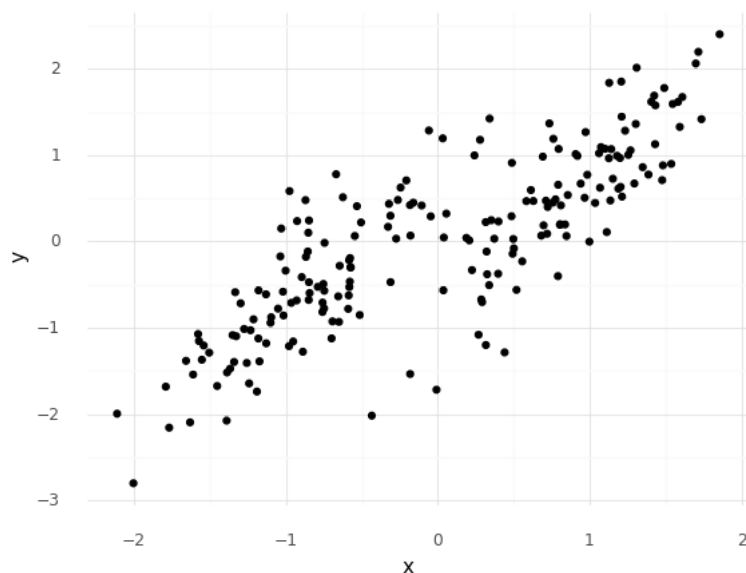
%matplotlib inline
```

```
d1 = pd.read_csv("https://raw.githubusercontent.com/cmparlettPelleriti/CPSC392ParlettPelleriti/master/Data/KMEM1.csv")
d1.head()
```

	x	y
0	1.206087	1.847686
1	0.351919	0.245691
2	1.201688	0.630774
3	0.733710	1.364328
4	1.150745	0.724892

```
z = StandardScaler()
d1[["x", "y"]] = z.fit_transform(d1)

ggplot(d1, aes(x = "x", y = "y")) + geom_point() + theme_minimal()
```



<ggplot: (8776873643888)>

```
# dbscan and plot
db1 = DBSCAN(eps = 0.5, min_samples = 25).fit(d1)
print(len(db1.labels_))
```

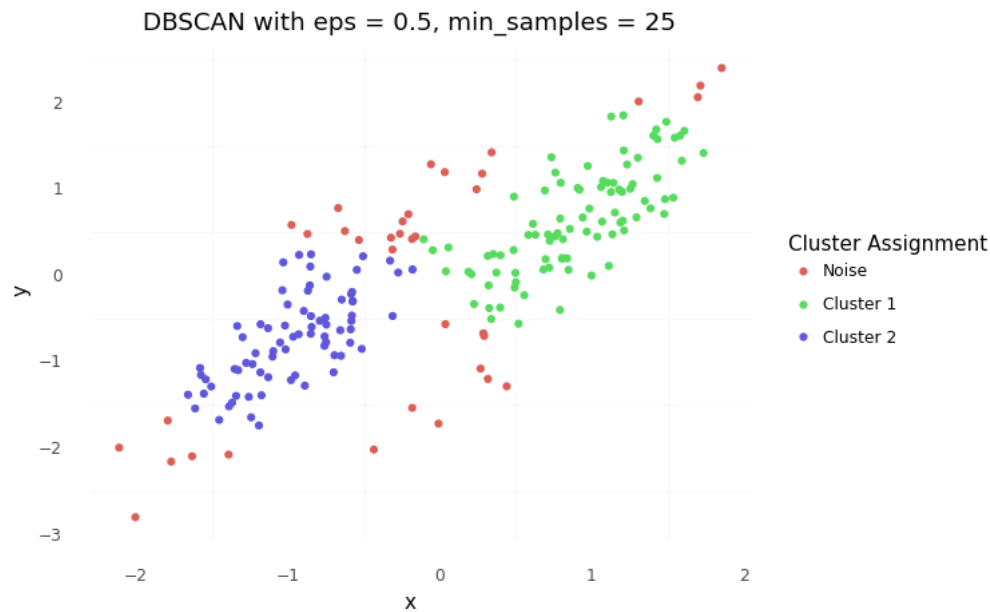
200

```
labsList = ["Noise"]
labsList = labsList + ["Cluster " + str(i) for i in range(1, len(set(db1.labels_)))]

d1["assignments"] = db1.labels_

(ggplot(d1, aes(x = "x", y = "y", color = "factor(assignments)")) +
```

```
geom_point() +
theme_minimal() +
scale_color_discrete(name = "Cluster Assignment",
                      labels = labsList) +
theme(panel_grid_major = element_blank()) +
labs(title = "DBSCAN with eps = 0.5, min_samples = 25")
```



<ggplot: (8776873506871)>

```
print(d1["assignments"])
```

```
0      0
1      0
2      0
3      0
4      0
..
195    1
196    1
197    1
198    1
199   -1
Name: assignments, Length: 200, dtype: int64
```

```
# grab only clustered data points
d1_clustered = d1.loc[(d1.assignments >= 0)]

silhouette_score(d1_clustered[["x","y"]], d1_clustered["assignments"])

0.6169609560667532
```

```
# overall
silhouette_score(d1[["x","y"]], d1["assignments"])

0.39760284702143056
```

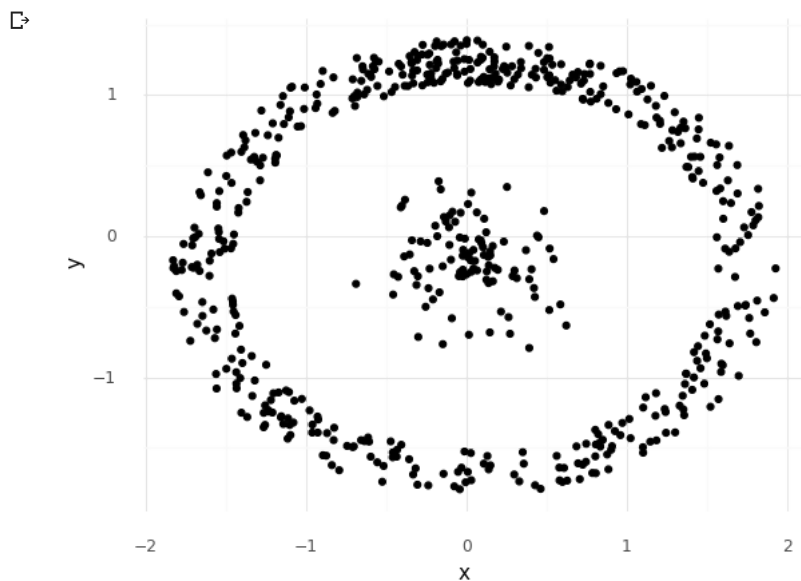
#### Dataset 4

```
d4 = pd.read_csv("https://raw.githubusercontent.com/cmparlettPelleriti/CPSC392ParlettPelleriti/master/Data/KMEM4.csv")
d4.head()
```

	x	y
0	-0.006848	0.395527
1	0.314820	-0.289261
2	0.171705	1.078077
3	-1.203661	1.325926
4	-0.179379	-0.036615

```
z = StandardScaler()
d4[["x","y"]] = z.fit_transform(d4)

ggplot(d4, aes(x = "x", y = "y")) + geom_point() + theme_minimal()
```



<ggplot: (8776873511997)>

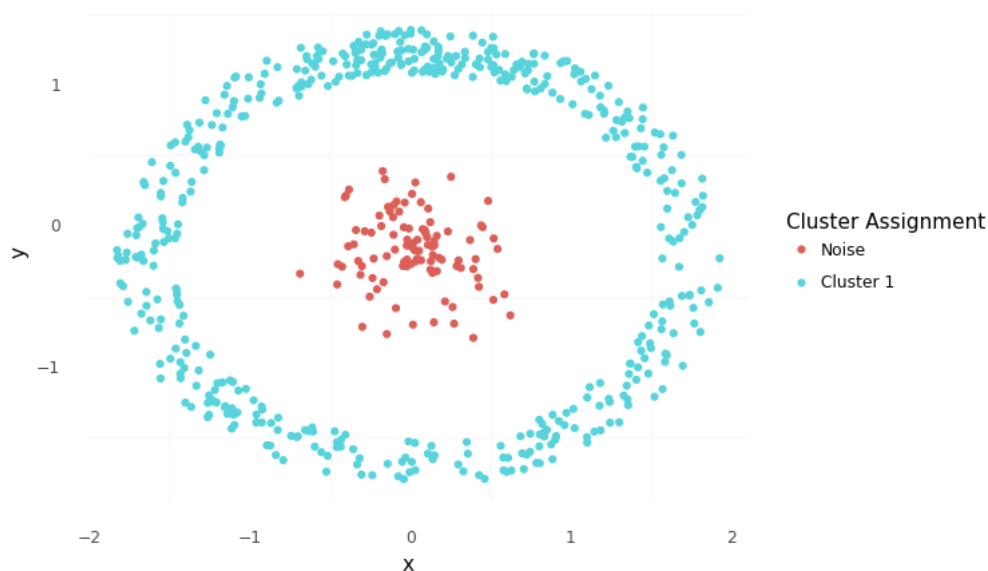
```
db4 = DBSCAN(eps = 0.5, min_samples = 3).fit(d4)

labsList = ["Noise"]
labsList = labsList + ["Cluster " + str(i) for i in range(1,len(set(db4.labels_)))]

d4["assignments"] = db4.labels_

(ggplot(d4, aes(x = "x", y = "y", color = "factor(assignments)")) +
 geom_point() +
 theme_minimal() +
 scale_color_discrete(name = "Cluster Assignment",
                      labels = labsList) +
 theme(panel_grid_major = element_blank()) +
 labs(title = "DBSCAN with eps = 0.5, min_samples = 3"))
```

DBSCAN with eps = 0.5, min\_samples = 3



<ggplot: (8776870473713)>

Compare

```
from sklearn.cluster import AgglomerativeClustering
```

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

hac = AgglomerativeClustering(2)
km = KMeans(2)
gm = GaussianMixture(2)

hac.fit(d4[["x", "y"]])
km.fit(d4[["x", "y"]])
gm.fit(d4[["x", "y"]])

d4["hac"] = hac.labels_
d4["km"] = km.labels_
d4["gm"] = gm.predict(d4[["x", "y"]])

print(ggplot(d4, aes(x = "x", y = "y",
                    color = "factor(hac)")) +
      geom_point() +
      theme_minimal() +
      scale_color_discrete(name = "Cluster Assignment") +
      theme(panel_grid_major = element_blank()) +
      labs(title = "Hierarchical Agglomerative"))

print(ggplot(d4, aes(x = "x", y = "y",
                    color = "factor(km)")) +
      geom_point() +
      theme_minimal() +
      scale_color_discrete(name = "Cluster Assignment") +
      theme(panel_grid_major = element_blank()) +
      labs(title = "KMeans"))

print(ggplot(d4, aes(x = "x", y = "y",
                    color = "factor(gm)")) +
      geom_point() +
      theme_minimal() +
      scale_color_discrete(name = "Cluster Assignment") +
      theme(panel_grid_major = element_blank()) +
      labs(title = "Gaussian Mixtures"))
```

