K-means Clustering with Dataset PCA

Stephanie Cheng

2025-04-09

K-mean with PCA

Reduce dimensions and prepare data for clustering

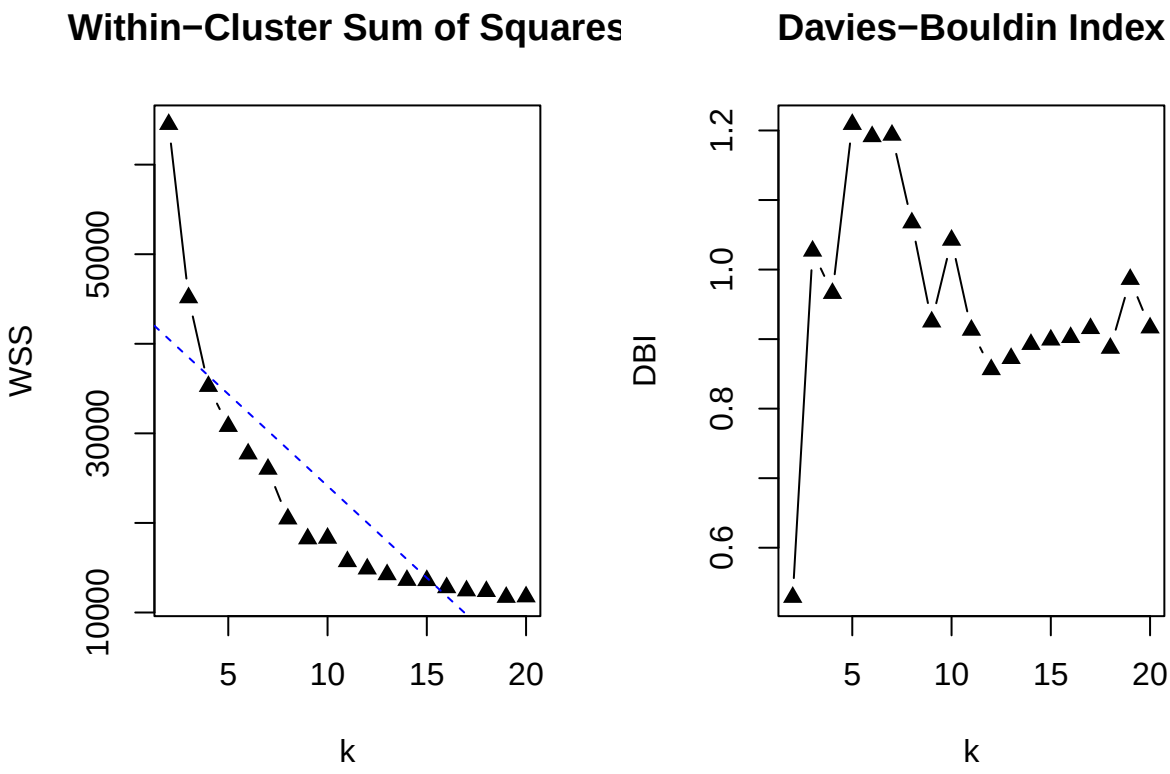
```
nyc_pca <- pca(new_york_city_data, "New York City")
```

```
##
## ===== PCA Summary for New York City =====
## Importance of components:
##                            PC1    PC2    PC3     PC4     PC5
## Standard deviation     1.6621 1.1450 0.7821 0.55057 0.10703
## Proportion of Variance 0.5525 0.2622 0.1223 0.06063 0.00229
## Cumulative Proportion  0.5525 0.8147 0.9371 0.99771 1.00000
## Contributing variable for each PC:
##                 PC1                 PC2                 PC3                 PC4
##       "TOTAL.UNITS"        "YEAR.BUILT"           "BOROUGH" "GROSS.SQUARE.FEET"
##                 PC5
##       "TOTAL.UNITS"
```

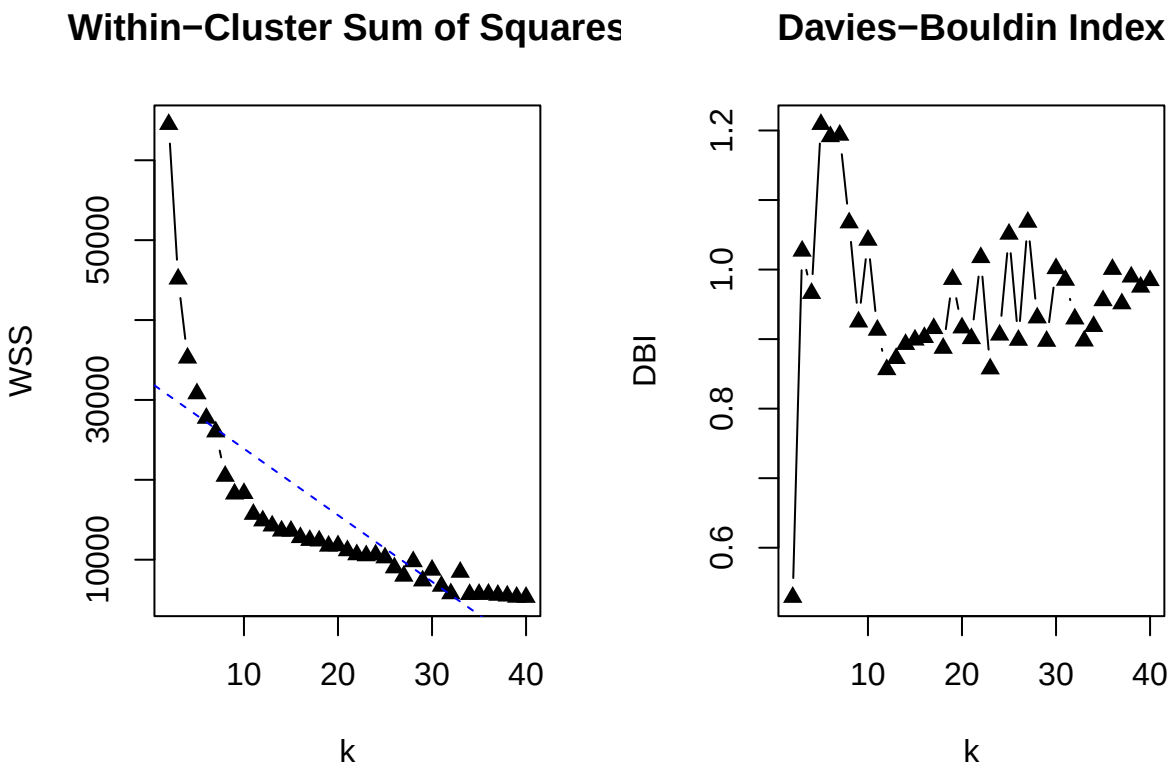
Calculate clustering evaluation with Davies Bouldin index & Within-cluster sum of squares. See the affect when K is increasing, then we can apply elbow method to avoid picking the best k within overfitting case.

```
nyc_k_stats_20 <- calculate_k_stats_PCA(nyc_pca, max_k = 20)
nyc_k_stats_40 <- calculate_k_stats_PCA(nyc_pca, max_k = 40)

# DBI & WSS plot
elbows_20 <- plot_kmeans(nyc_k_stats_20$errs, nyc_k_stats_20$DBI)
```



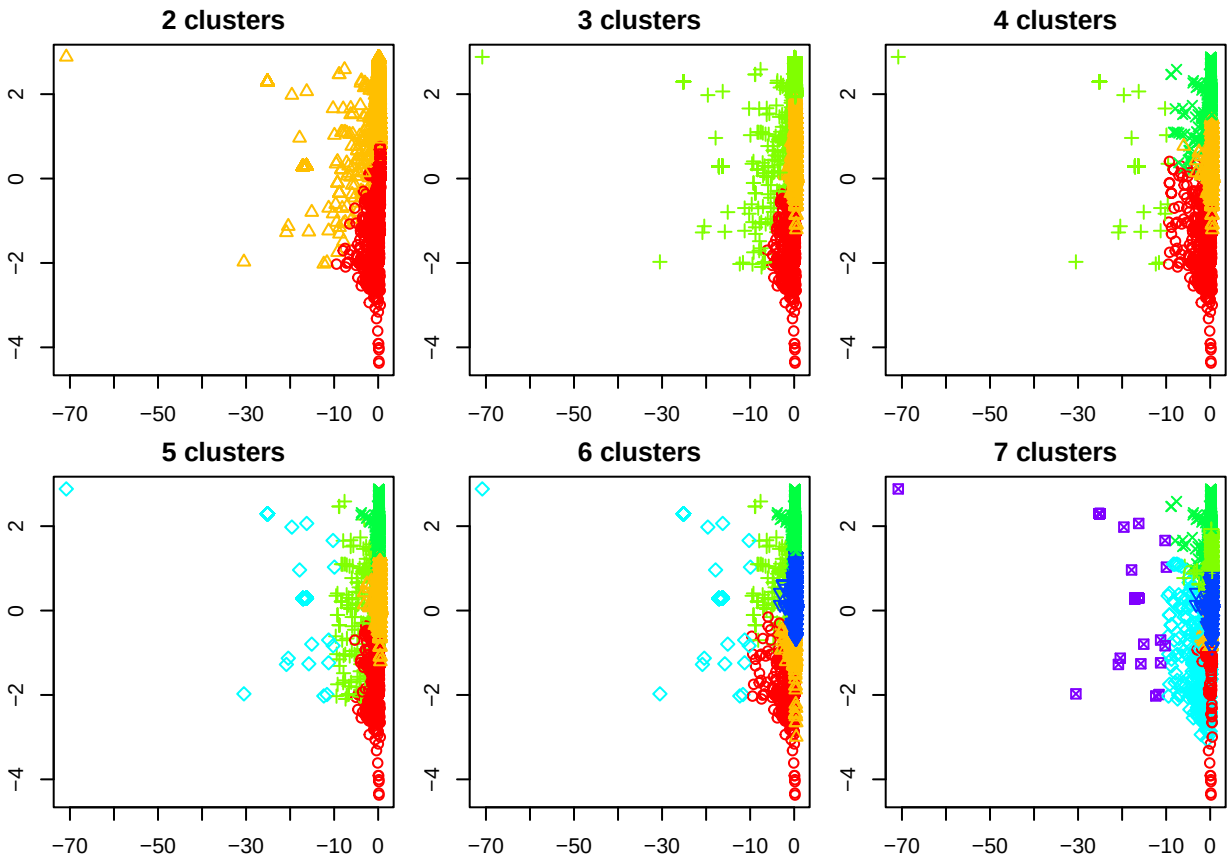

```
elbows_40 <- plot_kmeans(nyc_k_stats_40$errs, nyc_k_stats_40$DBI)
```




```
best_k <- 4
```

Plot all clusters from 2 to 7 as the best k clusters is within that range.

```
plot_clusters(nyc_k_stats_20$X.syn, min_k = 2, max_k = 7)
```



K-means on PCA as PCA gives a lower-dimensional variable that improves clustering quality

```
km <- kmeans(nyc_pca$x, centers = best_k, nstart = 25)
summarize_kmeans(km, "New York City")
```

```
##
## ===== K-means Model Performance Summary for New York City =====
## Total within-cluster sum of squares (WSS): 35251.74
##
## Cluster sizes:
## [1] 4534 8979 9712  148
##
## Cluster centers (in PCA space):
##    TOTAL.UNITS YEAR.BUILT    BOROUGH GROSS.SQUARE.FEET   TOTAL.UNITS
## 1   0.04775774  1.6588805  0.6265526      0.0004797091 -0.0022669754
## 2   0.32302248  0.2507256 -0.5846668     -0.0317504231 -0.0011815392
## 3  -0.03575967 -1.0168523  0.2619555      0.0259434145  0.0008871676
## 4 -18.71388200  0.6962166 -0.9133678      0.2091189580  0.0829144223
```

Interpret what the clusters mean with the original data

```
new_york_city_data$cluster <- km$cluster
aggregate(. ~ cluster, data = new_york_city_data, mean)
```

```
##   cluster  BOROUGH RESIDENTIAL.UNITS TOTAL.UNITS GROSS.SQUARE.FEET YEAR.BUILT
```

```
## 1       1 4.244817          2.838774    3.018306          4280.963   1993.093
## 2       2 4.162602          1.540149    1.642165          1975.099   1934.099
## 3       3 2.568678          3.458299    4.071664          6344.001   1924.844
## 4       4 1.547297        313.898649  319.216216        416353.034   2002.135
```

Export the clusters for Supervised learning

```
dir.create("after_cluster_dataset")
```

```
## Warning in dir.create("after_cluster_dataset"): 'after_cluster_dataset' already
## exists
```

```
write.csv(new_york_city_data,
          file = "after_cluster_dataset/cleaned_nyc_with_clusters.csv",
          row.names = FALSE)
```