ML_Problem5_Clustering_DimRed

2024-08-13

Writeup

For problem #5, we ran PCA, T-sne, and K-Means. T-Sne was by far the worst of the 3 models, while the PCA and K-Means turned out good. Based on the plot looking at quality, we believe that the K-means test seems to be the most reliable.

Loading Neccesary Libraries

```
library(readr)
library(ggplot2)
library(ggfortify)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(Rtsne)
```

Loading Data

Process & Scale the Data

```
wine_data_scaled <- scale(wine_data[, -ncol(wine_data)])
head(wine_data_scaled)
```

```
##      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## [1,]     0.1424623        2.1886645   -2.192664     -0.7447208 0.5699140
## [2,]     0.4510010        3.2819823   -2.192664     -0.5975941 1.1978825
## [3,]     0.4510010        2.5531038   -1.917405     -0.6606484 1.0266184
## [4,]     3.0735801       -0.3624106    1.660957     -0.7447208 0.5413699
## [5,]     0.1424623        2.1886645   -2.192664     -0.7447208 0.5699140
## [6,]     0.1424623        1.9457049   -2.192664     -0.7657389 0.5413699
##      free.sulfur.dioxide total.sulfur.dioxide   density         pH sulphates
## [1,]          -1.1000552           -1.4462472 1.0349132  1.8129500 0.1930819
## [2,]          -0.3112961           -0.8624022 0.7014323 -0.1150642 0.9995017
## [3,]          -0.8746955           -1.0924018 0.7681285  0.2580999 0.7978967
## [4,]          -0.7620156           -0.9862481 1.1016093 -0.3638402 0.3274852
## [5,]          -1.1000552           -1.4462472 1.0349132  1.8129500 0.1930819
## [6,]          -0.9873753           -1.3400936 1.0349132  1.8129500 0.1930819
##         alcohol    quality
## [1,] -0.9153937 -0.9371575
## [2,] -0.5800235 -0.9371575
## [3,] -0.5800235 -0.9371575
## [4,] -0.5800235  0.2079830
## [5,] -0.9153937 -0.9371575
## [6,] -0.9153937 -0.9371575
```

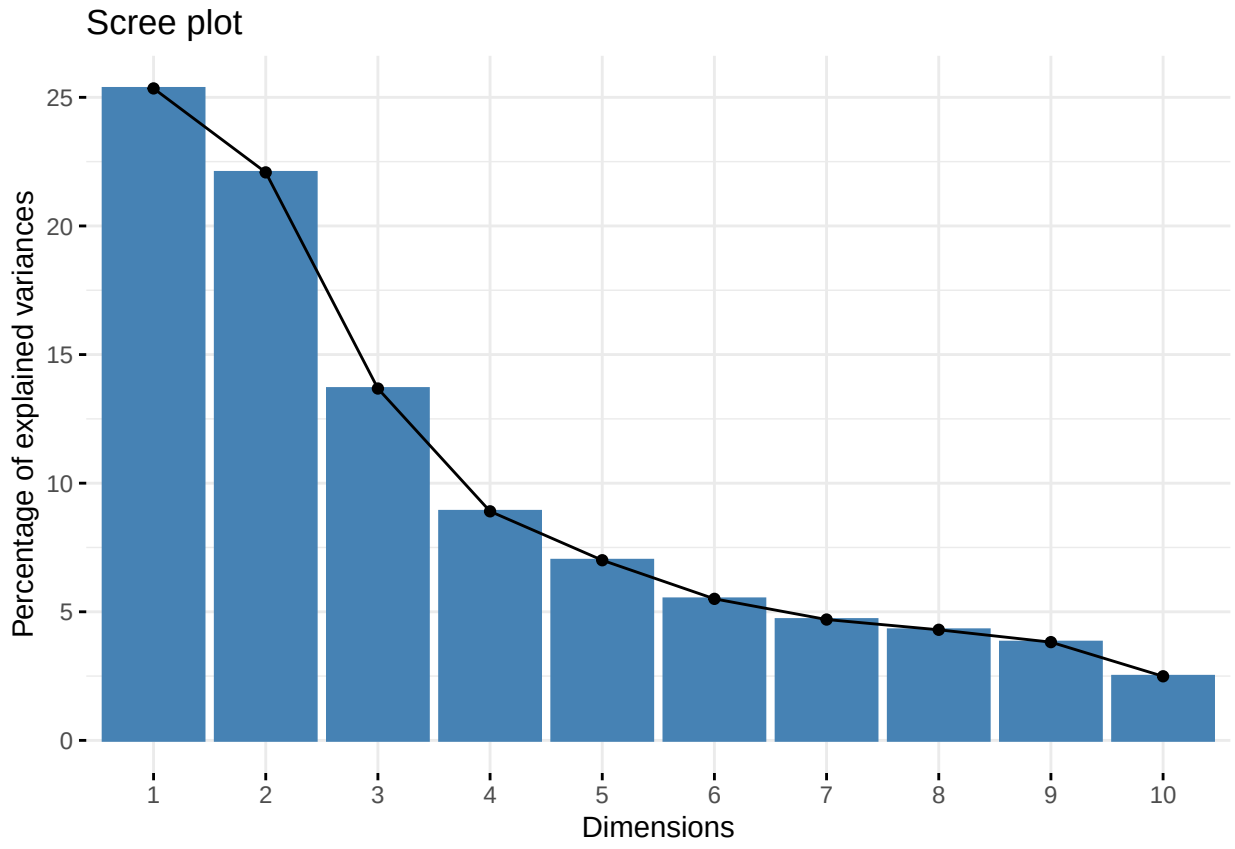
Perform the PCA

```
pca_result <- prcomp(wine_data_scaled, center = TRUE, scale. = TRUE)
summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.7440 1.6278 1.2812 1.03374 0.91679 0.81265 0.75088
## Proportion of Variance 0.2535 0.2208 0.1368 0.08905 0.07004 0.05503 0.04699
## Cumulative Proportion  0.2535 0.4743 0.6111 0.70013 0.77017 0.82520 0.87219
##                           PC8    PC9    PC10    PC11    PC12
## Standard deviation     0.7183 0.6770 0.54682 0.47706 0.18107
## Proportion of Variance 0.0430 0.0382 0.02492 0.01897 0.00273
## Cumulative Proportion  0.9152 0.9534 0.97830 0.99727 1.00000
```

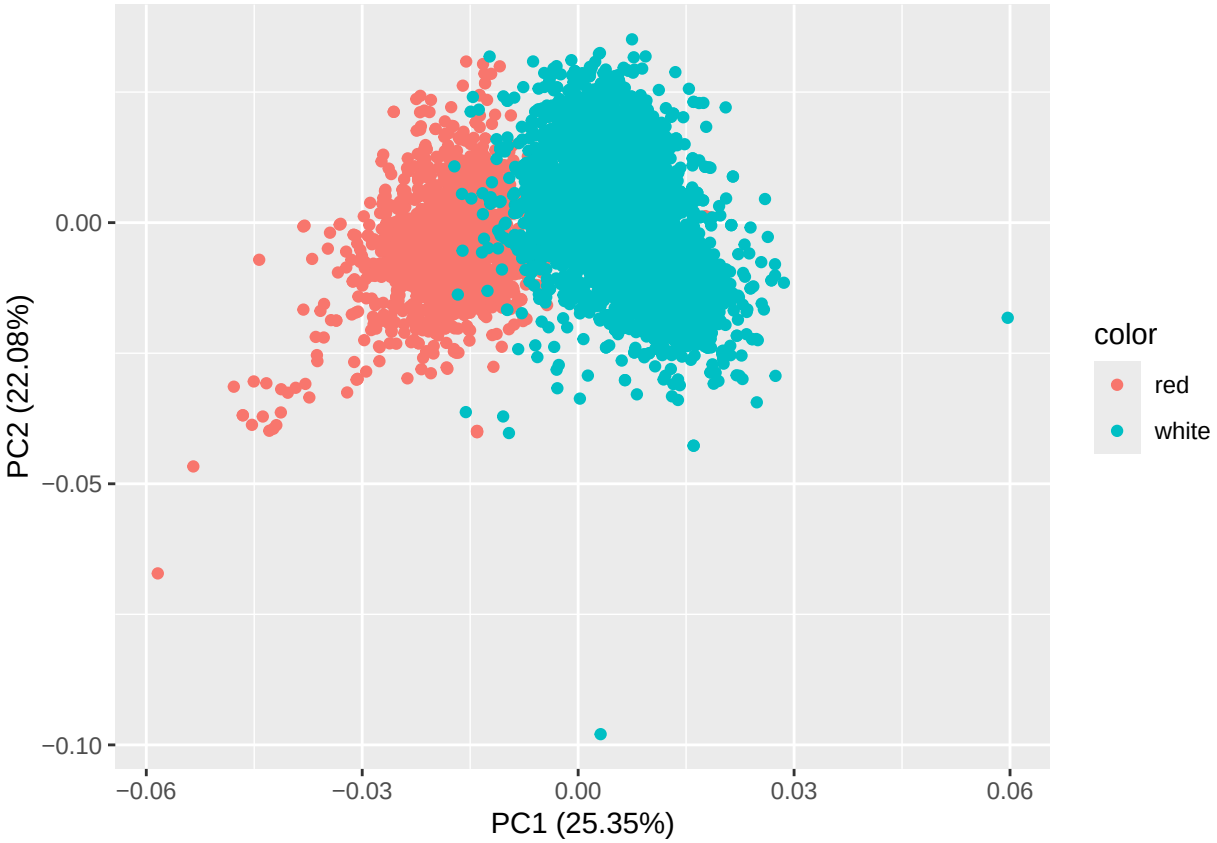
Visaulize using Scree Plot

```
fviz_eig(pca_result)
```



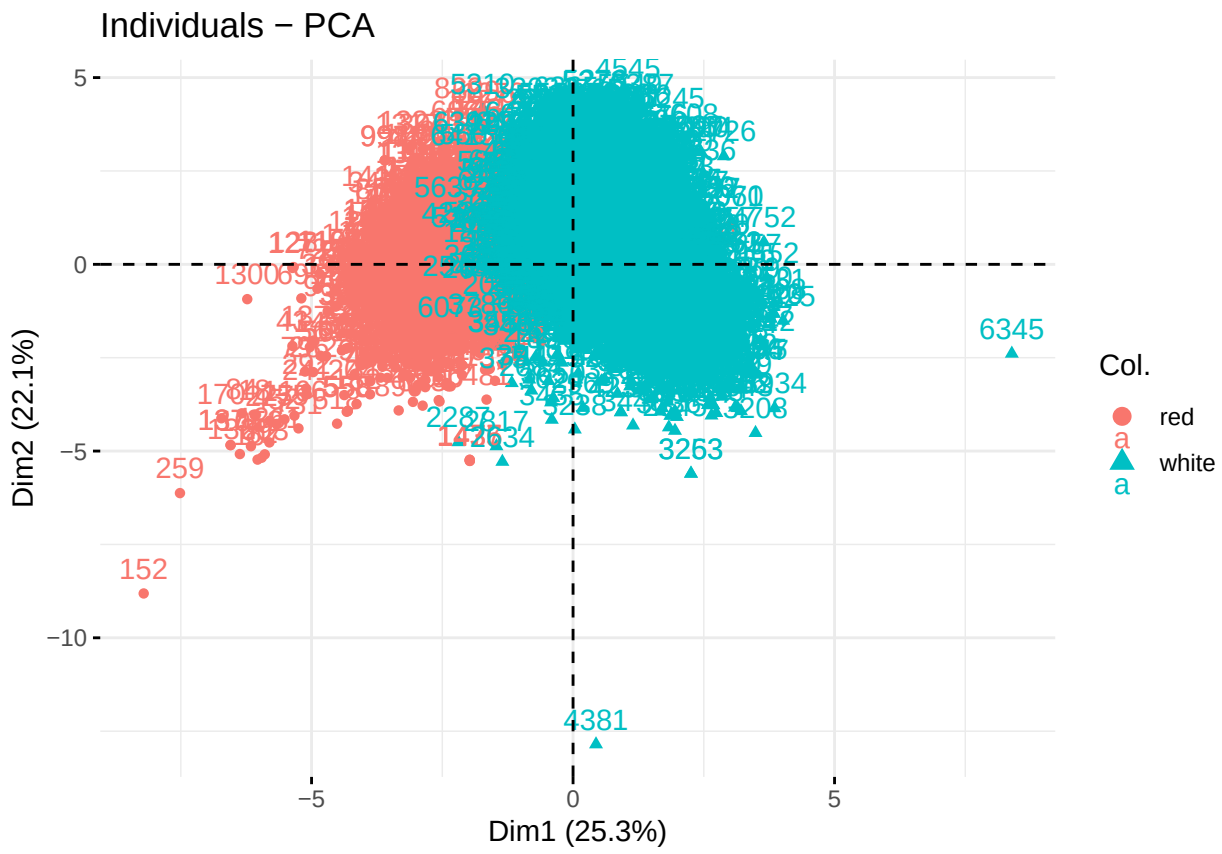
Visualize using PCA Biplot

```
autoplot(pca_result, data = wine_data, colour = 'color')
```



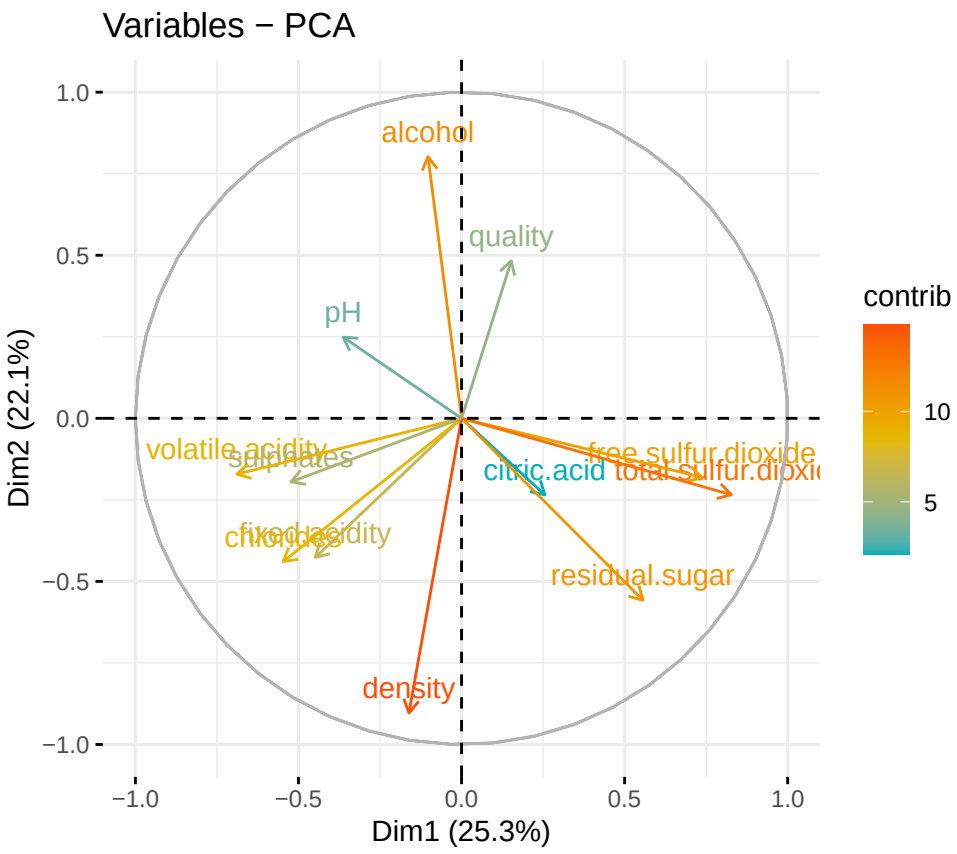
Plot Individuals

```
fviz_pca_ind(pca_result, col.ind = wine_data$color)
```



Visaulize Variables Plot

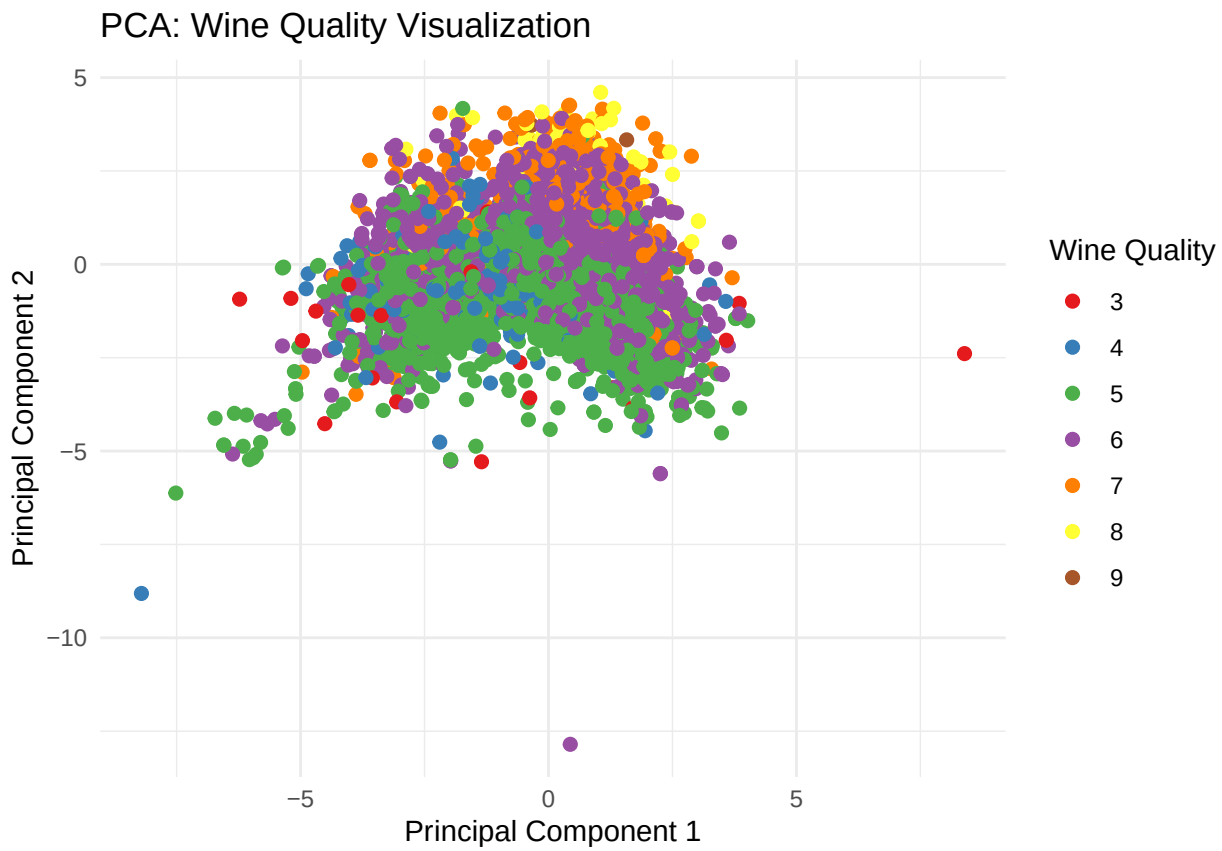
```
fviz_pca_var(pca_result, col.var = "contrib", gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"))
```



Adding quality labels to PCA

```
pca_data <- as.data.frame(pca_result$x)
pca_data$Quality <- wine_data$quality  # Replace 'quality' with the actual column name for wine quality

# Plot the first two principal components
ggplot(pca_data, aes(x = PC1, y = PC2, color = as.factor(Quality))) +
  geom_point(size = 2) +
  scale_color_brewer(palette = "Set1") +
  theme_minimal() +
  labs(title = "PCA: Wine Quality Visualization",
       x = "Principal Component 1",
       y = "Principal Component 2",
       color = "Wine Quality")
```




```
loadings <- pca_result$rotation[, 1:2]  # For the first two principal components

# Print loadings
print(loadings)
```

```
##                              PC1        PC2
## fixed.acidity        -0.25692873 -0.2618431
## volatile.acidity     -0.39493118 -0.1051983
## citric.acid           0.14646061 -0.1440935
## residual.sugar        0.31890519 -0.3425850
## chlorides            -0.31344994 -0.2697701
## free.sulfur.dioxide   0.42269137 -0.1111788
## total.sulfur.dioxide  0.47441968 -0.1439475
## density              -0.09243753 -0.5549205
## pH                   -0.20806957  0.1529219
## sulphates            -0.29985192 -0.1196342
## alcohol              -0.05892408  0.4927275
## quality               0.08747571  0.2966009
```

Run tSNE

```
# Combine scaled data with the target variable
wine_data_tsne <- wine_data_scaled
```

```r
wine_data_clean <- wine_data[!duplicated(wine_data_tsne), ]
wine_data_tsne_unique <- wine_data_clean[, -ncol(wine_data_clean)]  # Exclude target variable
target_variable <- wine_data_clean$color  # Store target variable separately
```

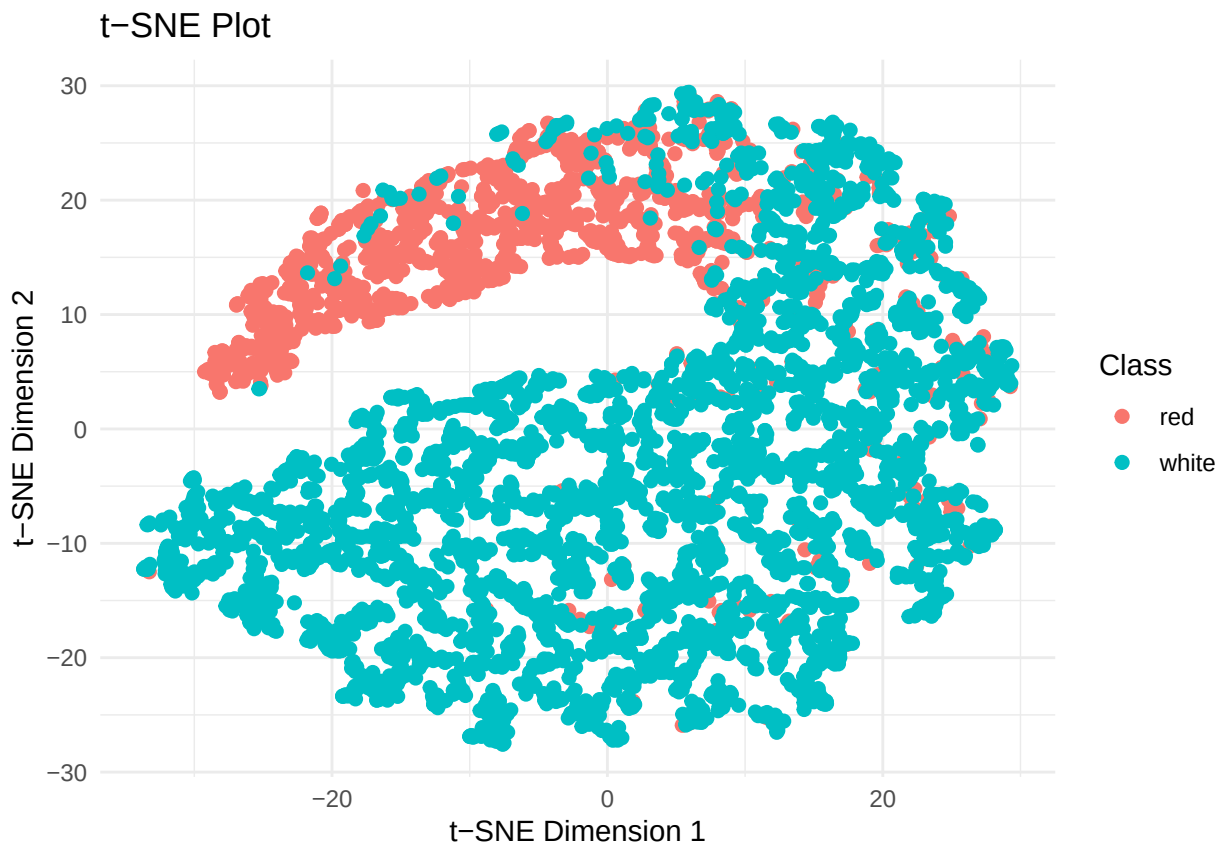
```r
set.seed(22)  # For reproducibility
tsne_result <- Rtsne(wine_data_tsne_unique, dims = 2, perplexity = 30, verbose = TRUE, max_iter = 500)
```

```
## Performing PCA
## Read the 5318 x 12 data matrix successfully!
## OpenMP is working. 1 threads.
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 0.83 seconds (sparsity = 0.020625)!
## Learning embedding...
## Iteration 50: error is 90.831002 (50 iterations in 0.52 seconds)
## Iteration 100: error is 72.474480 (50 iterations in 0.47 seconds)
## Iteration 150: error is 69.574016 (50 iterations in 0.44 seconds)
## Iteration 200: error is 68.104705 (50 iterations in 0.43 seconds)
## Iteration 250: error is 67.352846 (50 iterations in 0.42 seconds)
## Iteration 300: error is 2.042336 (50 iterations in 0.40 seconds)
## Iteration 350: error is 1.626538 (50 iterations in 0.41 seconds)
## Iteration 400: error is 1.404730 (50 iterations in 0.45 seconds)
## Iteration 450: error is 1.269878 (50 iterations in 0.44 seconds)
## Iteration 500: error is 1.182186 (50 iterations in 0.45 seconds)
## Fitting performed in 4.43 seconds.
```

Plot tSNE

```r
tsne_data <- as.data.frame(tsne_result$Y)
colnames(tsne_data) <- c("Dim1", "Dim2")
tsne_data$Class <- target_variable  # Add the target variable back to the t-SNE result

# Visualize the t-SNE plot
ggplot(tsne_data, aes(x = Dim1, y = Dim2, color = Class)) +
  geom_point(size = 2) +
  theme_minimal() +
  labs(title = "t-SNE Plot", x = "t-SNE Dimension 1", y = "t-SNE Dimension 2")
```



Testing how well tSNE performed for quality

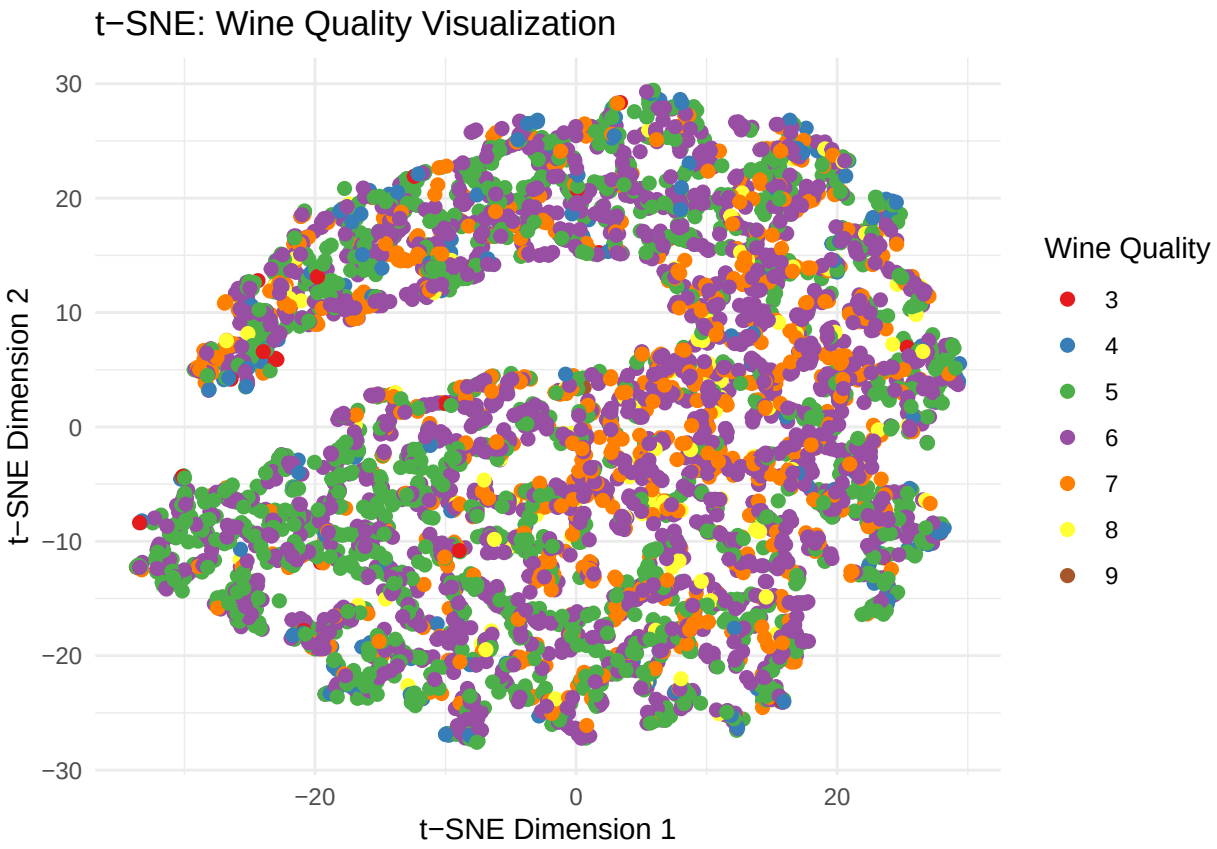
```
wine_data_clean <- wine_data[!duplicated(wine_data_tsne), ]

# Confirm the dimensions match
dim(wine_data_clean)  # Should match the dimensions of wine_data_tsne_unique and tsne_data
```

```
## [1] 5318   13
```

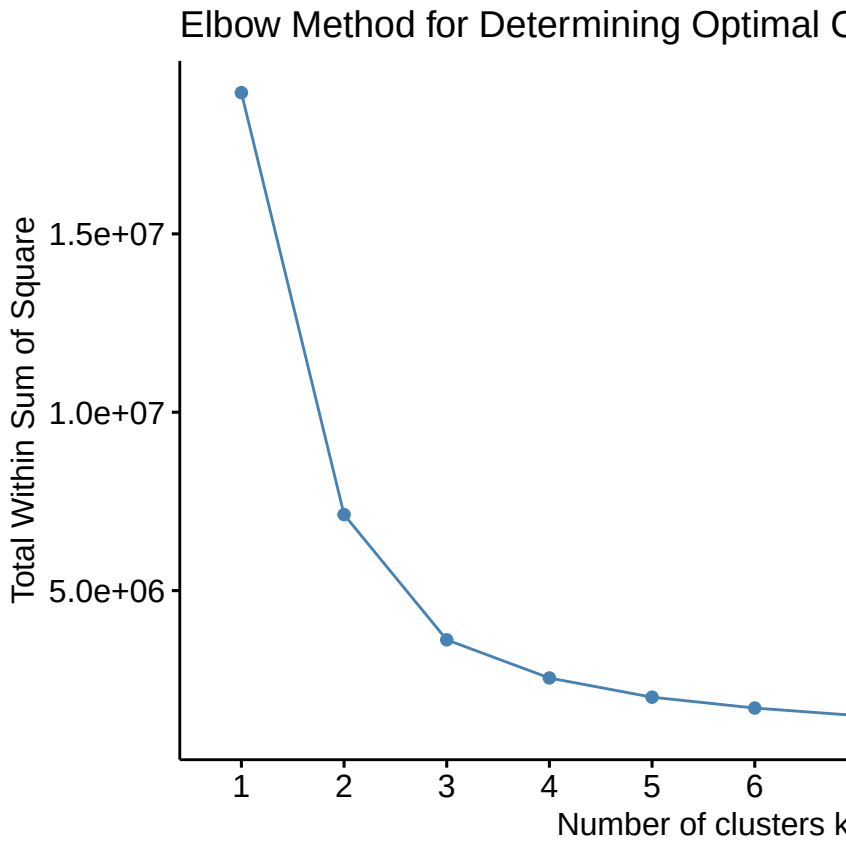
```
# Now add the Quality column from the cleaned data to tsne_data
tsne_data$Quality <- wine_data_clean$quality

ggplot(tsne_data, aes(x = Dim1, y = Dim2, color = as.factor(Quality))) +
  geom_point(size = 2) +
  scale_color_brewer(palette = "Set1") +
  theme_minimal() +
  labs(title = "t-SNE: Wine Quality Visualization",
       x = "t-SNE Dimension 1",
       y = "t-SNE Dimension 2",
       color = "Wine Quality")
```



Running a K-Means Analysis

```
wine_data_kmeans <- wine_data_tsne_unique
set.seed(22)
fviz_nbclust(wine_data_kmeans, kmeans, method = "wss") +
  labs(title = "Elbow Method for Determining Optimal Clusters")
```

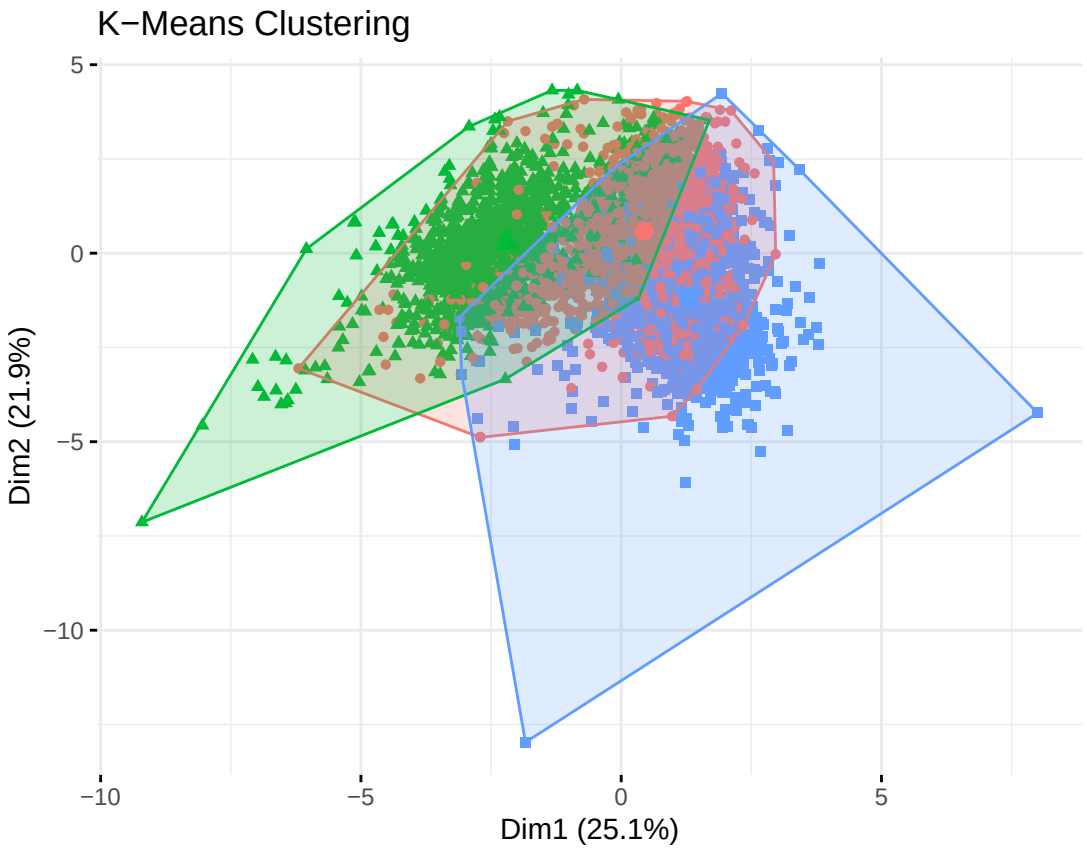


Determining the best nummber of Clusters

Running the K-Means Algorithm

```
set.seed(22)
kmeans_result <- kmeans(wine_data_kmeans, centers = 3, nstart = 100)
```

```
wine_data_kmeans$cluster <- as.factor(kmeans_result$cluster)
wine_data_kmeans_numeric <- wine_data_kmeans[, -which(names(wine_data_kmeans) == "cluster")]
# Visualize clusters using PCA
fviz_cluster(kmeans_result, data = wine_data_kmeans_numeric,
             geom = "point", ellipse.type = "convex",
             ggtheme = theme_minimal()) +
  labs(title = "K-Means Clustering")
```



Visualizing the Clusters

Visualizing K-Means performance on Quality

```
wine_data_numeric <- wine_data_kmeans[, sapply(wine_data_kmeans, is.numeric)]

# Check the structure to ensure only numeric columns remain
str(wine_data_numeric)
```

```
## tibble [5,318 x 12] (S3: tbl_df/tbl/data.frame)
##  $ fixed.acidity       : num [1:5318] 7.4 7.8 7.8 11.2 7.4 7.9 7.3 7.8 7.5 6.7 ...
##  $ volatile.acidity    : num [1:5318] 0.7 0.88 0.76 0.28 0.66 0.6 0.65 0.58 0.5 0.58 ...
##  $ citric.acid         : num [1:5318] 0 0 0.04 0.56 0 0.06 0 0.02 0.36 0.08 ...
##  $ residual.sugar      : num [1:5318] 1.9 2.6 2.3 1.9 1.8 1.6 1.2 2 6.1 1.8 ...
##  $ chlorides           : num [1:5318] 0.076 0.098 0.092 0.075 0.075 0.069 0.065 0.073 0.071 0.097 ...
##  $ free.sulfur.dioxide : num [1:5318] 11 25 15 17 13 15 15 9 17 15 ...
##  $ total.sulfur.dioxide: num [1:5318] 34 67 54 60 40 59 21 18 102 65 ...
##  $ density             : num [1:5318] 0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num [1:5318] 3.51 3.2 3.26 3.16 3.51 3.3 3.39 3.36 3.35 3.28 ...
##  $ sulphates           : num [1:5318] 0.56 0.68 0.65 0.58 0.56 0.46 0.47 0.57 0.8 0.54 ...
##  $ alcohol             : num [1:5318] 9.4 9.8 9.8 9.8 9.4 9.4 10 9.5 10.5 9.2 ...
##  $ quality             : num [1:5318] 5 5 5 6 5 5 7 7 5 5 ...
```

```
pca_result <- prcomp(wine_data_numeric, center = TRUE, scale. = TRUE)
```

```r
# Add the first two principal components to the original data frame
wine_data_kmeans$PC1 <- pca_result$x[, 1]
wine_data_kmeans$PC2 <- pca_result$x[, 2]

ggplot(wine_data_kmeans, aes(x = PC1, y = PC2, color = as.factor(quality), shape = cluster)) +
  geom_point(size = 2) +
  scale_color_brewer(palette = "Set1") +
  theme_minimal() +
  labs(title = "K-Means Clustering with Wine Quality",
       x = "Principal Component 1",
       y = "Principal Component 2",
       color = "Wine Quality",
       shape = "Cluster")
```