Clustering and Dimensionality Reduction

2024-08-18

```r
library(ggplot2)
library(tidyverse)
```

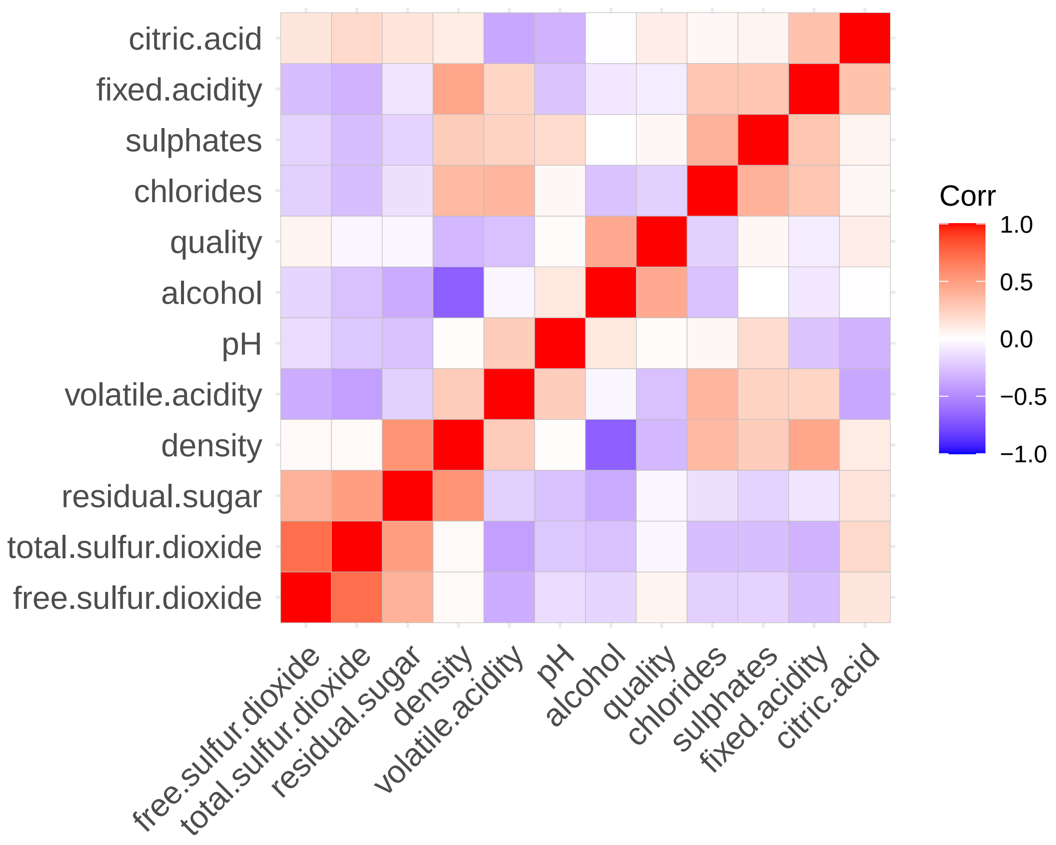
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(Rtsne)
library(ggcorrplot)
```

```r
# Read and make color binary
wines <- read.csv("/Users/krummelha/Downloads/wine.csv", header = TRUE)


# wine_color <- as.data.frame(wines$color)
# colnames(wine_color) <- "Color"
#
# wines <- wines[, !(names(wines) %in% c('color'))]


ggcorrplot::ggcorrplot(cor(wines[1:12]), hc.order = TRUE)
```



```r
# Normalize data,
Z <- scale(wines[1:12], center=TRUE, scale=FALSE)
```

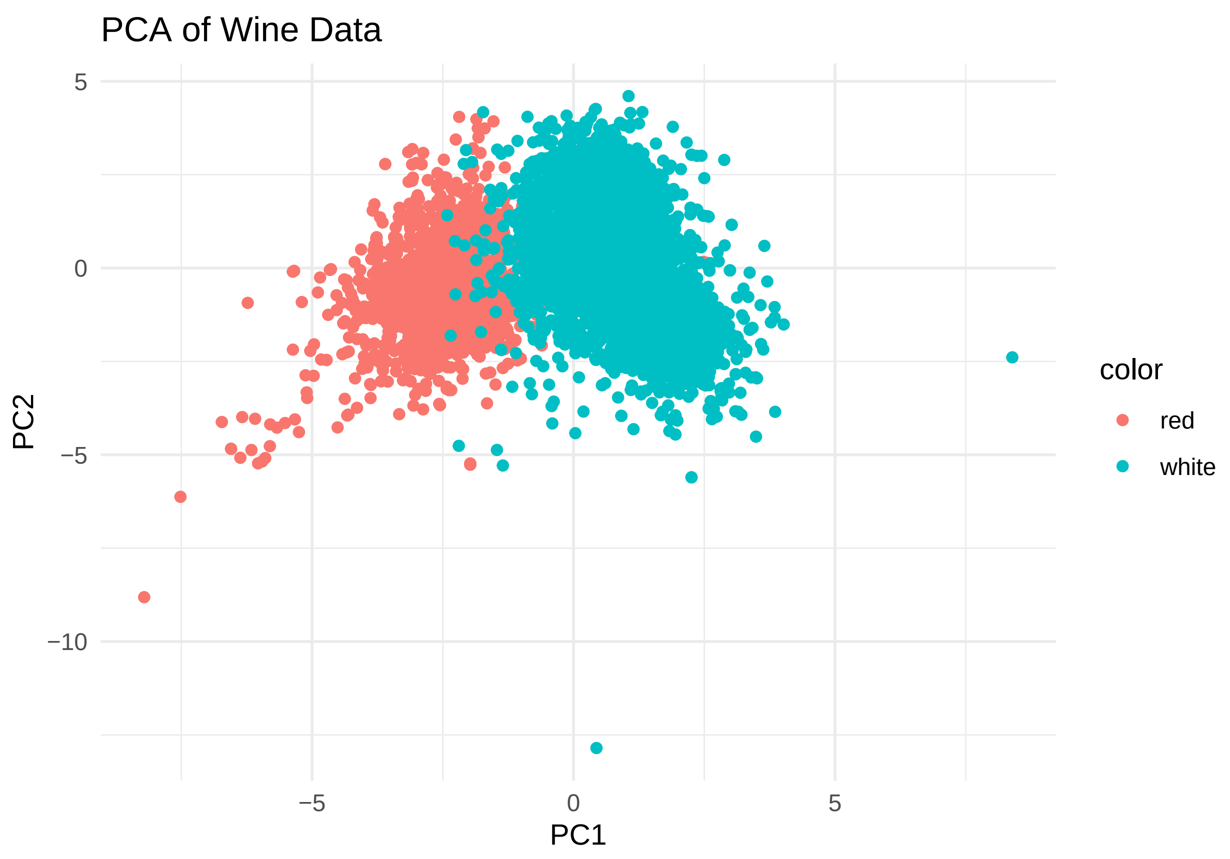
PCA Model

```r
pca <- prcomp(Z, center = TRUE, scale. = TRUE)
summary(pca)
```

```
## 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
```

```r
pc_data <- as.data.frame(pca$x[, 1:2])
pc_data$color <- wines$color
```

```r
# Plotting
ggplot(pc_data, aes(x = PC1, y = PC2, color = color)) +
  geom_point() +
  labs(title = "PCA of Wine Data", x = "PC1", y = "PC2") +
  theme_minimal()
```



PCA does a good job of clustering/separating between red and white wines. However the principal components found do not explain a large amount of the data. You can see from the summary that PC1 only explains variation of 25% of the data. It takes about 6-8 PCs to understand most of the data, and at that point it seems suboptimal, since the variables become exponentially difficult to interpret. The best option would be to utilize variables from the data instead in a different model.

t-SNE Model

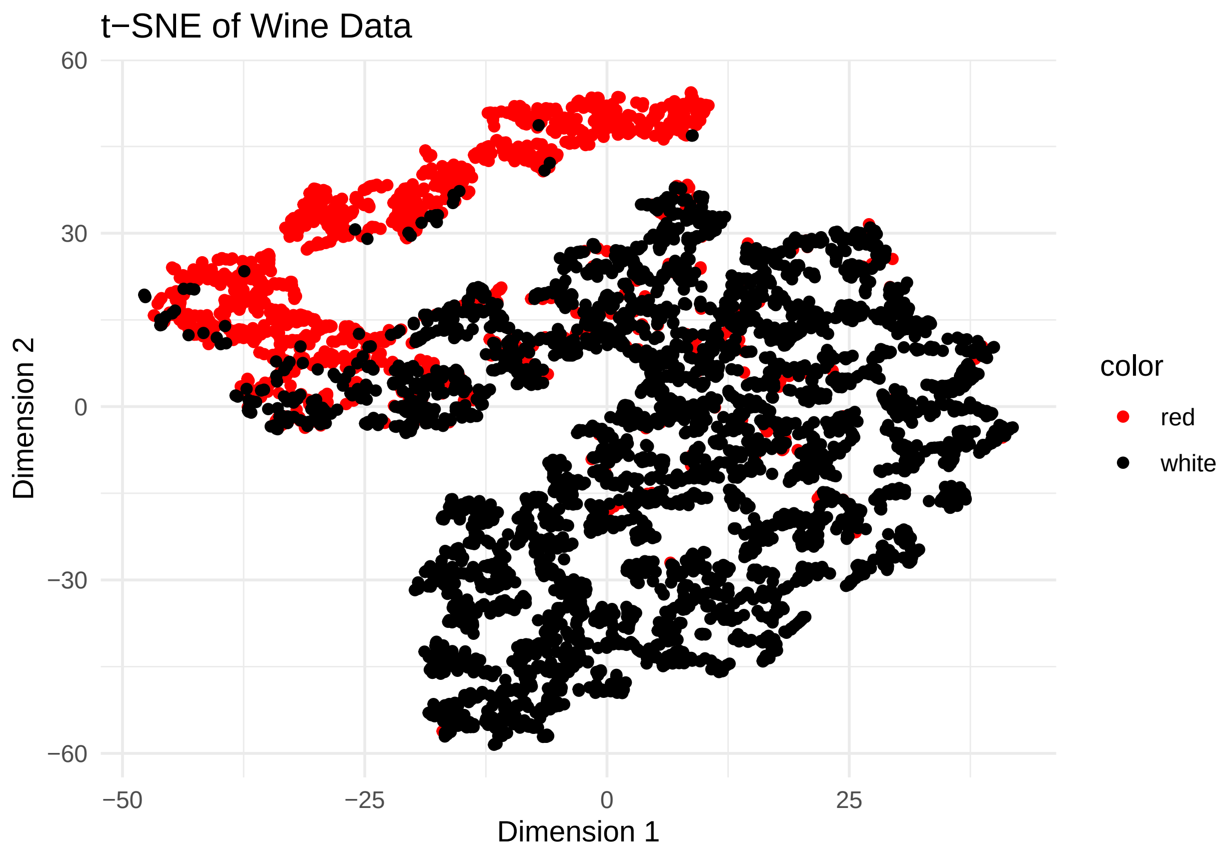
```r
wines_unique <- unique(wines)
A <- scale(wines_unique[1:12], center=TRUE, scale=FALSE)

tsne_results <- Rtsne(A, dims = 2, pca = TRUE, check_duplicates = FALSE, verbose = TRUE)
```

```
## Performing PCA
## Read the 5320 x 12 data matrix successfully!
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 0.24 seconds (sparsity = 0.020617)!
## Learning embedding...
## Iteration 50: error is 90.985787 (50 iterations in 0.45 seconds)
## Iteration 100: error is 73.245858 (50 iterations in 0.47 seconds)
## Iteration 150: error is 70.013464 (50 iterations in 0.51 seconds)
## Iteration 200: error is 68.850984 (50 iterations in 0.48 seconds)
## Iteration 250: error is 68.238907 (50 iterations in 0.48 seconds)
## Iteration 300: error is 2.066139 (50 iterations in 0.44 seconds)
## Iteration 350: error is 1.662714 (50 iterations in 0.42 seconds)
## Iteration 400: error is 1.442646 (50 iterations in 0.42 seconds)
## Iteration 450: error is 1.307364 (50 iterations in 0.42 seconds)
## Iteration 500: error is 1.218850 (50 iterations in 0.43 seconds)
## Iteration 550: error is 1.157362 (50 iterations in 0.44 seconds)
## Iteration 600: error is 1.114982 (50 iterations in 0.43 seconds)
## Iteration 650: error is 1.086639 (50 iterations in 0.46 seconds)
## Iteration 700: error is 1.067830 (50 iterations in 0.43 seconds)
## Iteration 750: error is 1.053616 (50 iterations in 0.43 seconds)
## Iteration 800: error is 1.042447 (50 iterations in 0.43 seconds)
## Iteration 850: error is 1.032553 (50 iterations in 0.44 seconds)
## Iteration 900: error is 1.023529 (50 iterations in 0.44 seconds)
## Iteration 950: error is 1.015847 (50 iterations in 0.44 seconds)
## Iteration 1000: error is 1.009684 (50 iterations in 0.44 seconds)
## Fitting performed in 8.89 seconds.
```

```r
# Create a data frame with the t-SNE results
tsne_data <- as.data.frame(tsne_results$Y)
colnames(tsne_data) <- c("Dim1", "Dim2")
tsne_data$color <- wines_unique$color

ggplot(tsne_data, aes(x = Dim1, y = Dim2, color = color)) +
  geom_point() +
  labs(title = "t-SNE of Wine Data", x = "Dimension 1", y = "Dimension 2") +
  theme_minimal() +
  scale_color_manual(values = c("red", "black"))
```



```
table(tsne_data$color)/table(wines$color)
```

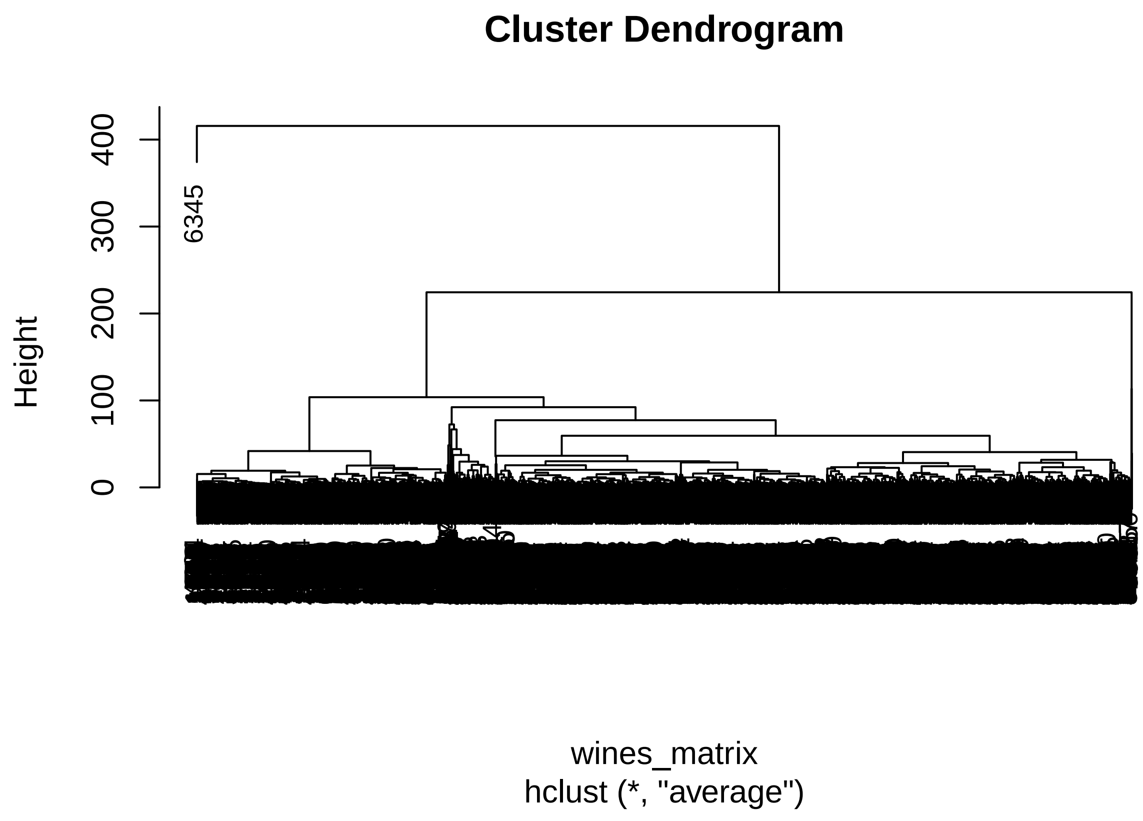
```
##
##       red     white
## 0.8499062 0.8086974
```

tSNE does not do a great job of separating reds vs white wines, and visually does not perform as well as other methods explored in this analysis. Since there are few variables that demonstrate obvious non-linear realtionships, it is suboptimal to use tSNE for this dataset.

Clustering Models

Hierarchichal

```
# Hierarchal
# Pairwise distance matrix using the dist function, use it in hclust, dendogram
wines_matrix = dist(Z, method='euclidean')
hier_wines = hclust(wines_matrix, method='average')
plot(hier_wines, cex=0.8)
```

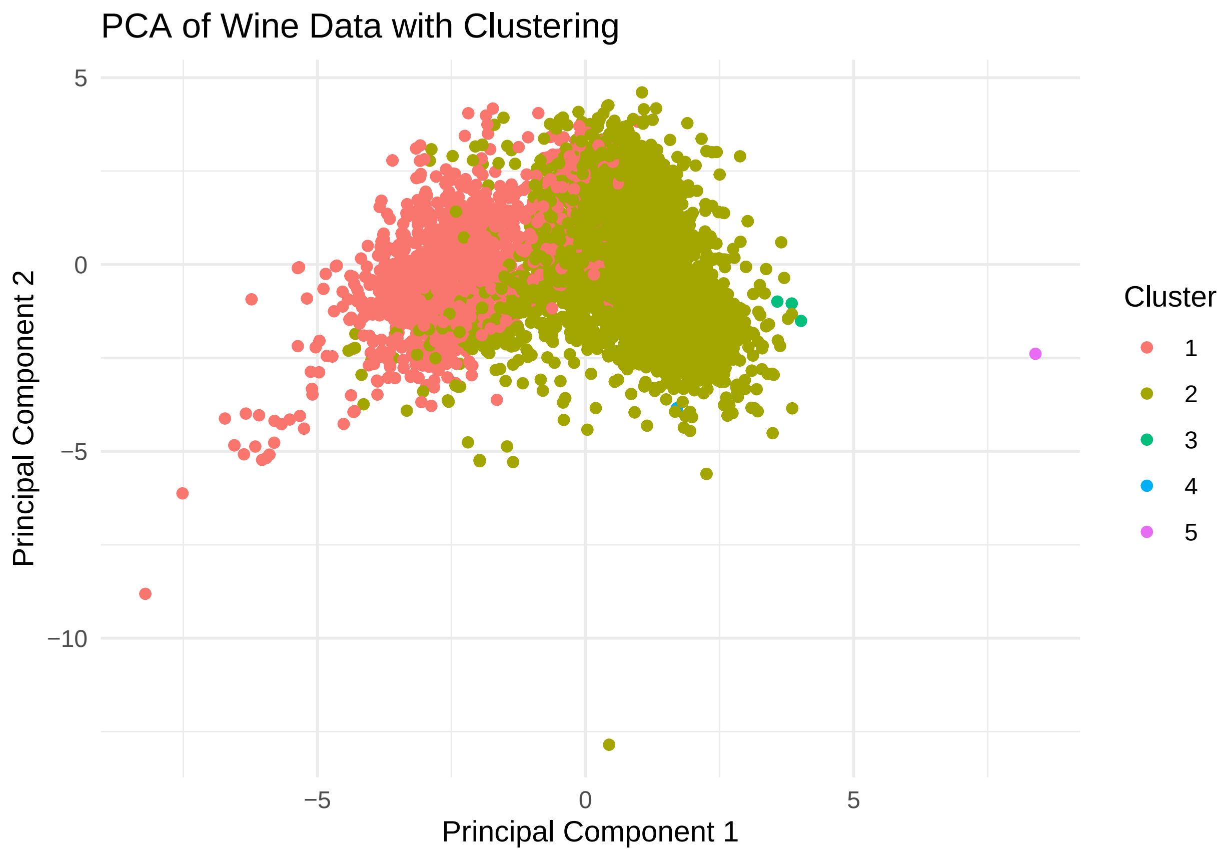


```
# Cut the tree into 5 clusters
wines_cluster = cutree(hier_wines, k=5)
summary(factor(wines_cluster))
```

```
##    1    2    3    4    5
## 1741 4750    3    2    1
```

```
# Results
pc_data$cluster <- as.factor(wines_cluster)

ggplot(pc_data, aes(x = PC1, y = PC2, color = cluster)) +
  geom_point() +
  labs(title = "PCA of Wine Data with Clustering",
       x = "Principal Component 1",
       y = "Principal Component 2",
       color = "Cluster") +
  theme_minimal()
```



```
table(pc_data$cluster)
```

```
##
##    1    2    3    4    5
## 1741 4750    3    2    1
```

```
table(wines$color)
```

```
##
##   red white
##  1599  4898
```

Percentage correct for reds and whites. Binned clusters 3,4,5 to the whites

```
cat('Reds:', 1741 / 1599, '\n')
```

```
## Reds: 1.088806
```

```
cat('Whites:', 4756 / 4898, '\n')
```

```
## Whites: 0.9710086
```

Overall, the hierarchical model did a decent job at predicting the clusters accurately for both the red and white wine categories. However, the model over predicted Red wines by about 8%, while it predicted 97% of the white wines correctly. Thus, clustering performs the best of all the models explored here because it offered the most accurate visual clustering relationships and overall results.

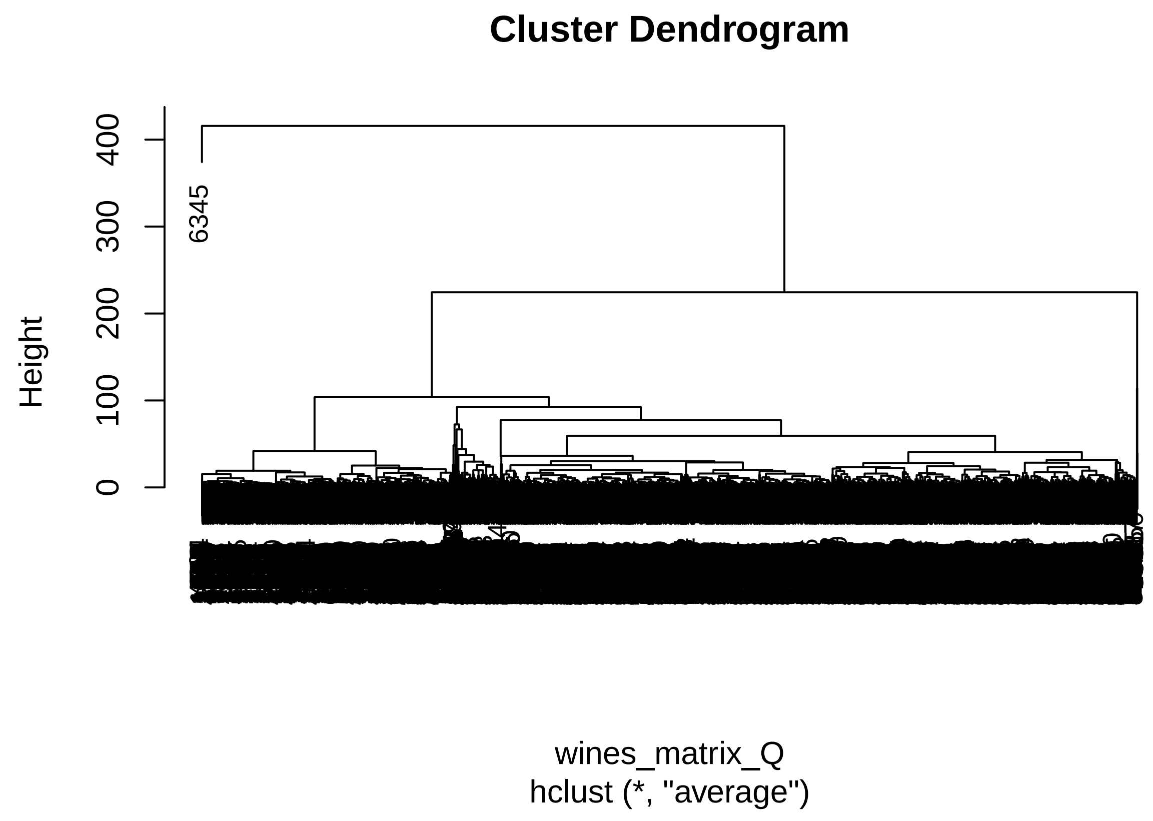
```
# Trying for quality...Make PCA for visualization, and trying wines quality cluster
Q <- scale(wines[1:11], center=TRUE, scale=FALSE)

pcaQ <- prcomp(Q, center = TRUE, scale. = TRUE)
summary(pcaQ)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.7407 1.5792 1.2475 0.98517 0.84845 0.77930 0.72330
## Proportion of Variance 0.2754 0.2267 0.1415 0.08823 0.06544 0.05521 0.04756
## Cumulative Proportion  0.2754 0.5021 0.6436 0.73187 0.79732 0.85253 0.90009
##                            PC8     PC9   PC10    PC11
## Standard deviation     0.70817 0.58054 0.4772 0.18119
## Proportion of Variance 0.04559 0.03064 0.0207 0.00298
## Cumulative Proportion  0.94568 0.97632 0.9970 1.00000
```

```
pc_data_Q <- as.data.frame(pcaQ$x[, 1:2])
pc_data_Q$quality <- wines$quality
```

```
wines_matrix_Q = dist(Z, method='euclidean')
hier_wines_Q = hclust(wines_matrix_Q, method='average')
plot(hier_wines_Q, cex=0.8)
```



```r
wines_cluster_Q = cutree(hier_wines_Q, k=7)
summary(factor(wines_cluster_Q))
```

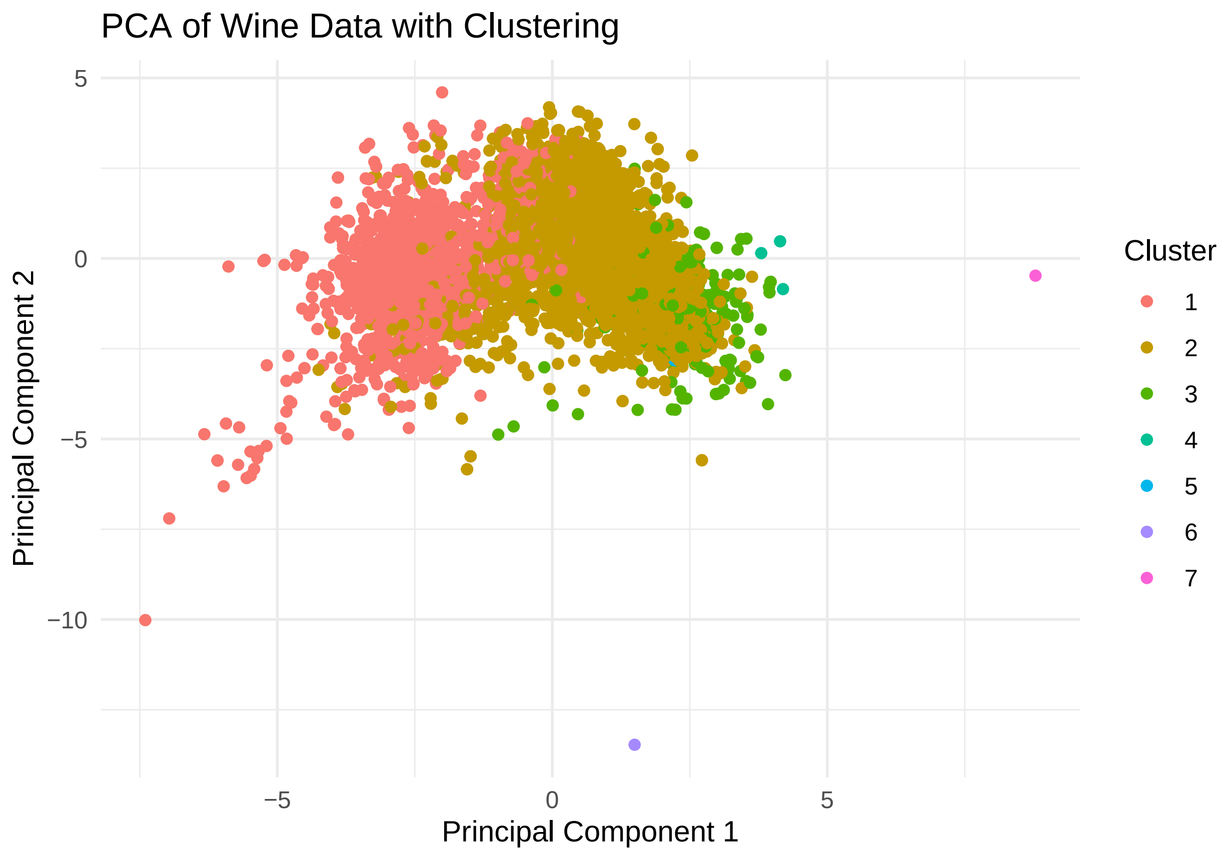
```
##    1    2    3    4    5    6    7
## 1741 4417  332    3    2    1    1
```

```r
table(wines$quality)
```

```
##
##    3    4    5    6    7    8    9
##   30  216 2138 2836 1079  193    5
```

```r
pc_data_Q$cluster <- as.factor(wines_cluster_Q)

ggplot(pc_data_Q, aes(x = PC1, y = PC2, color = cluster)) +
  geom_point() +
  labs(title = "PCA of Wine Data with Clustering",
       x = "Principal Component 1",
       y = "Principal Component 2",
       color = "Cluster") +
  theme_minimal()
```



The clustering model did not do a good job of determining wine quality. As seen in the plot, this model continues to separate the data based on whether a wine is red or white rather than its quality. Thus, though this model did a great job classifying whether a wine is red or white, another model would be a better option to determine wine quality given the constraints of the features given.