HW Marcus Martinez

Marcus Martinez

2024-08-05

5

Wine Color

We will examine different dimensionality-reducing methods and k-means clustering on the chemical properties to see if we can distinguish between white and red wine. We will cluster into two groups as we try to say if the wine is red or white. We will analyze it using k-means clustering. We selected two for our k since we are targeting two outcomes.

Set Up

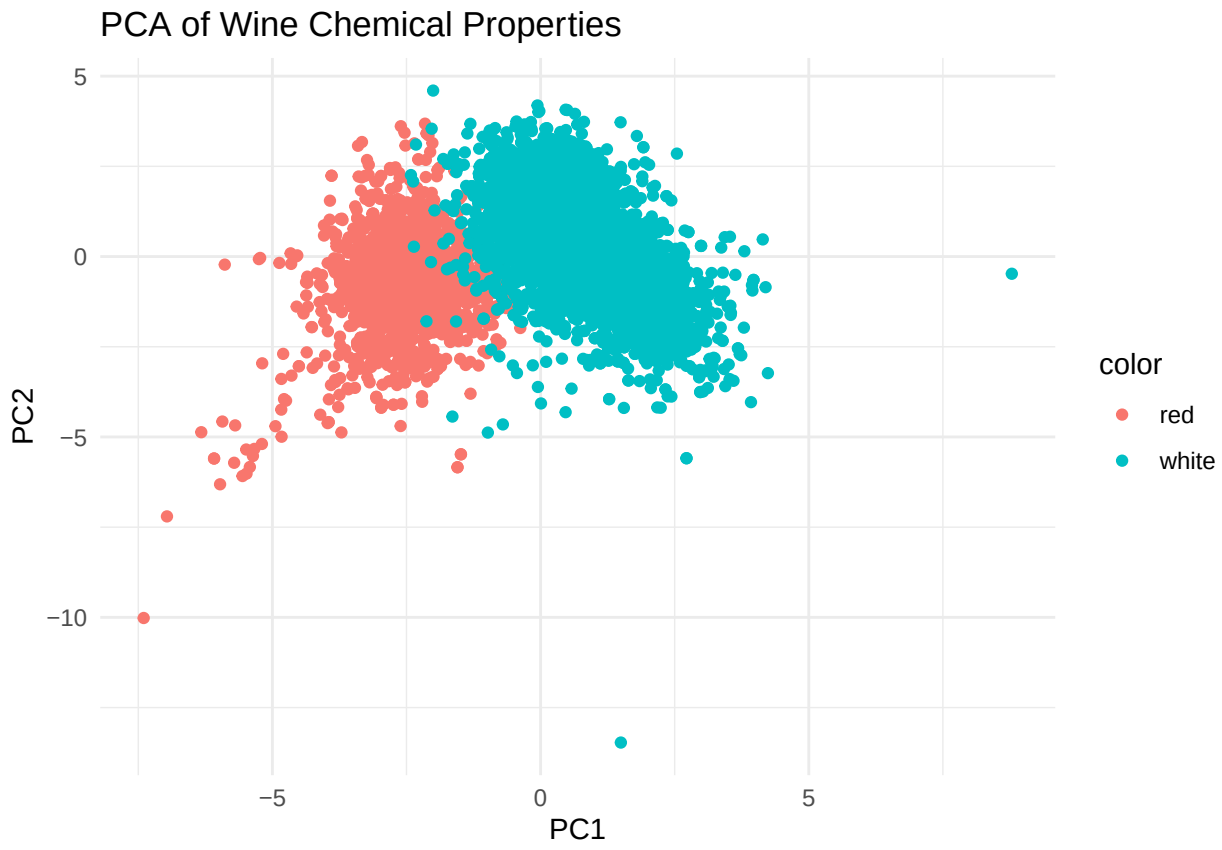
```
library(ggplot2)
library(Rtsne)
library(cluster)
wine_data <- read.csv("/Users/navyasinghal/Desktop/Summer/Intro to Machine Learning/Unsupervised Learni

# Select only the chemical properties for analysis
chemical_properties <- wine_data[, 1:11]
```

PCA

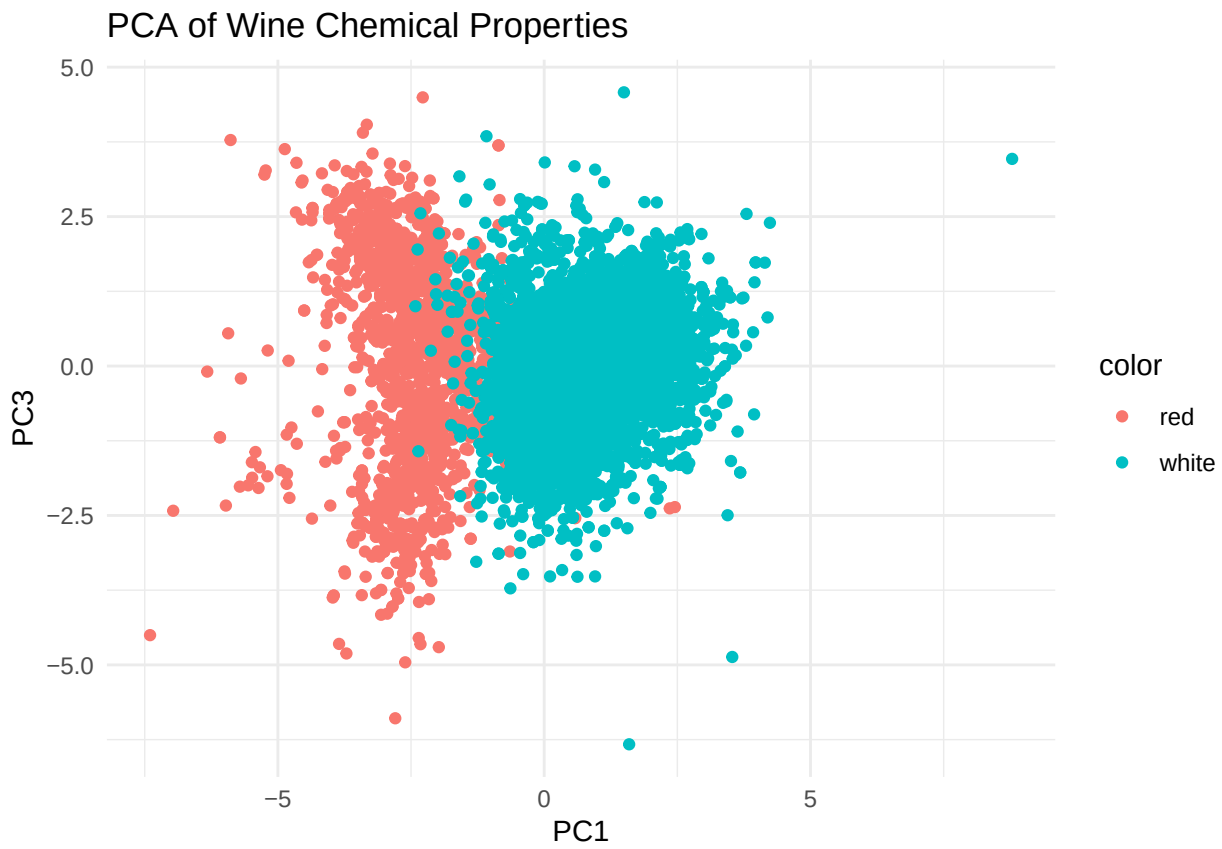
```
# Perform PCA
pca <- prcomp(chemical_properties, center = TRUE, scale. = TRUE)
summary(pca)
```

```
## 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
```

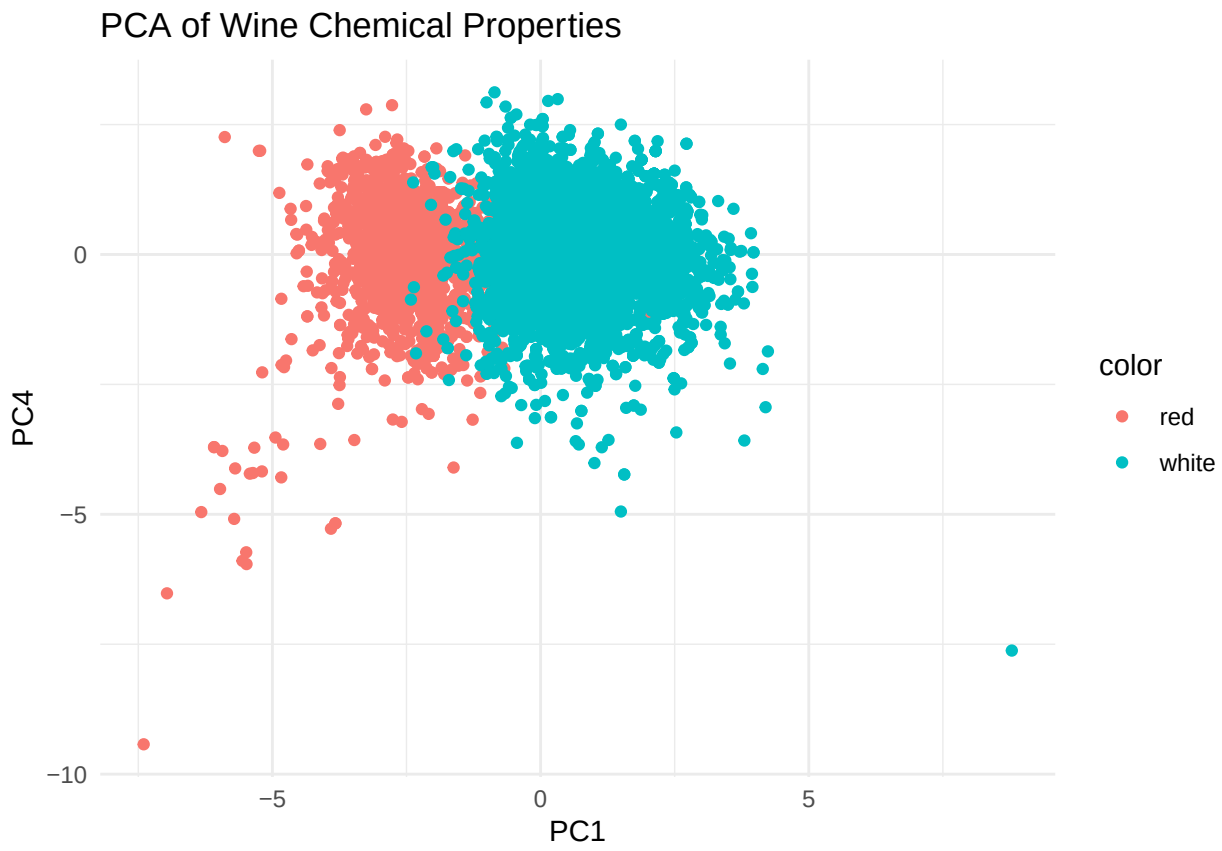
```r
pca_result <- data.frame(pca$x[, 1:11])
pca_result$color <- wine_data$color

# Plot PCA results: 1 and 2
ggplot(pca_result, aes(x = PC1, y = PC2, color = color)) +
  geom_point() +
  ggtitle("PCA of Wine Chemical Properties") +
  theme_minimal()
```



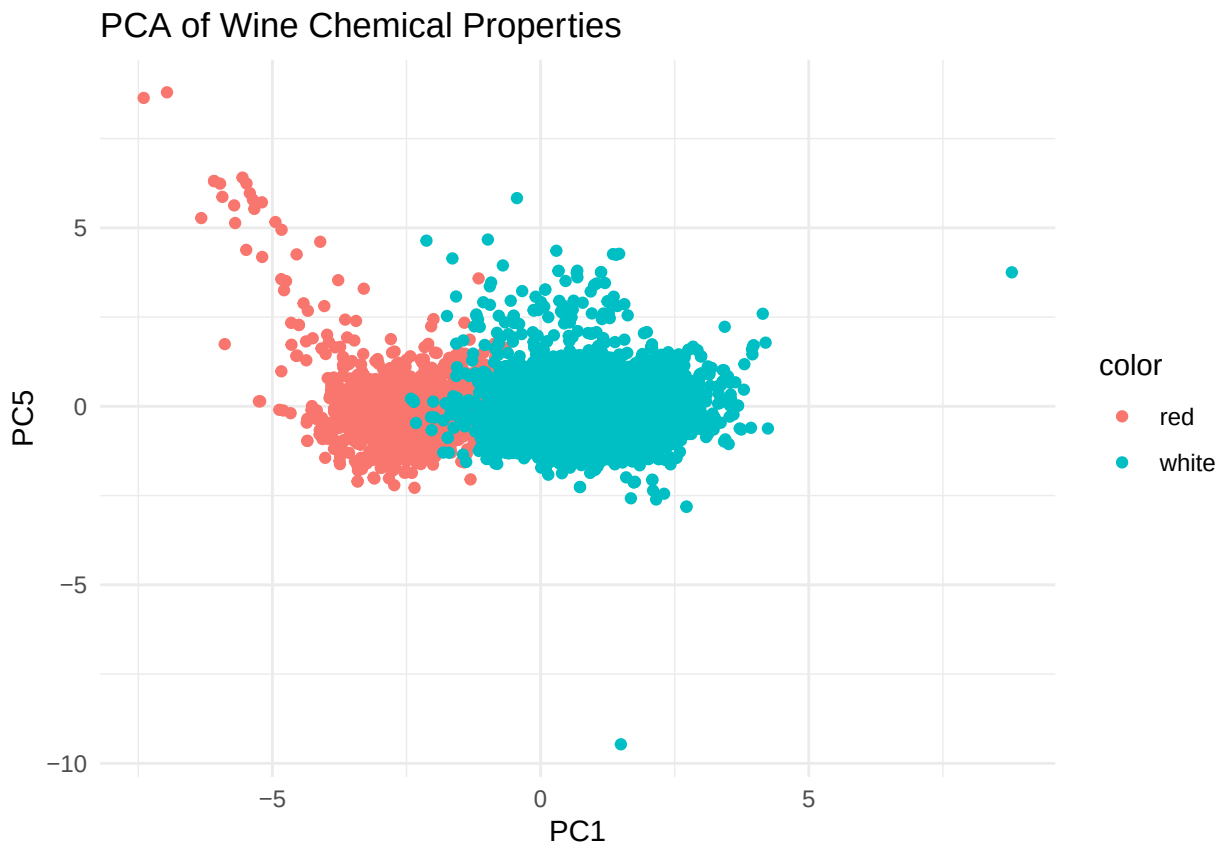

```r
# Plot PCA results: 1 and 3
ggplot(pca_result, aes(x = PC1, y = PC3, color = color)) +
  geom_point() +
  ggtitle("PCA of Wine Chemical Properties") +
  theme_minimal()
```



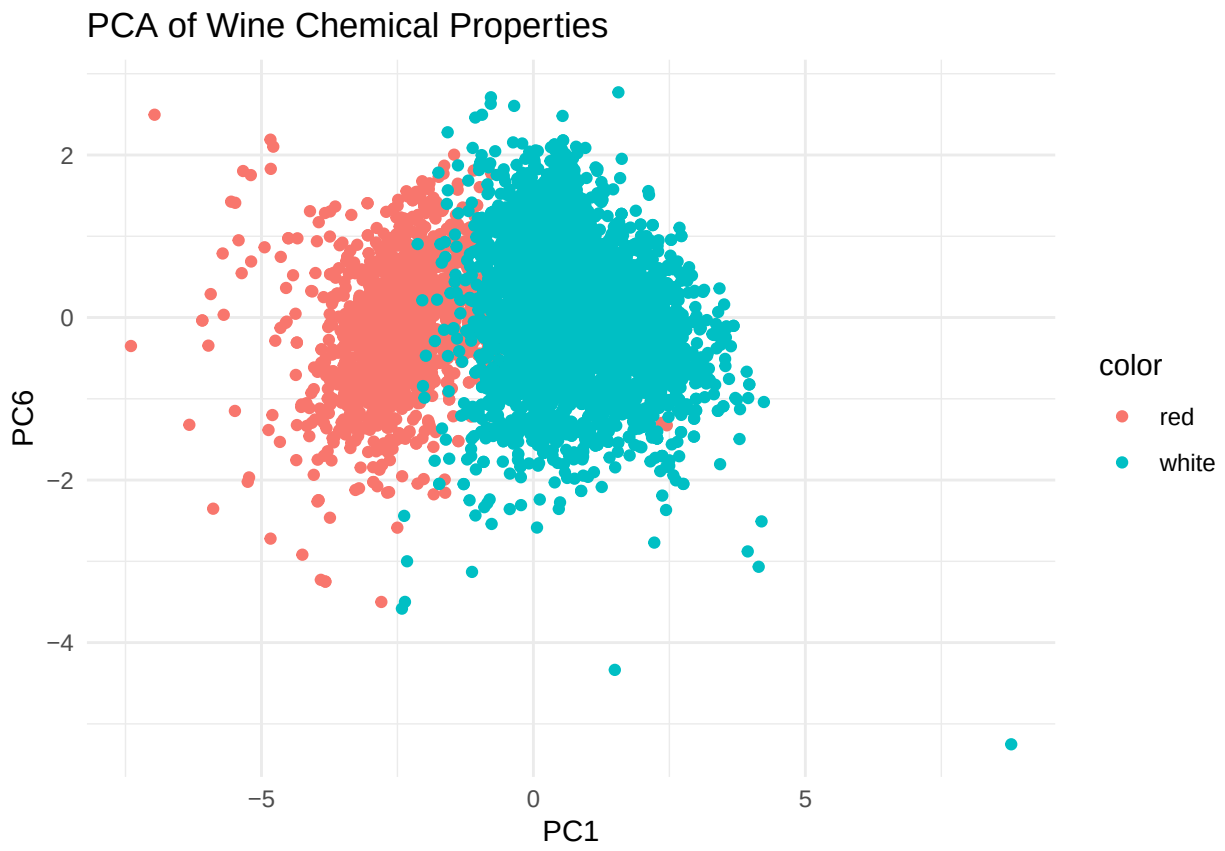

```r
# Plot PCA results: 1 and 4
ggplot(pca_result, aes(x = PC1, y = PC4, color = color)) +
  geom_point() +
  ggtitle("PCA of Wine Chemical Properties") +
  theme_minimal()
```



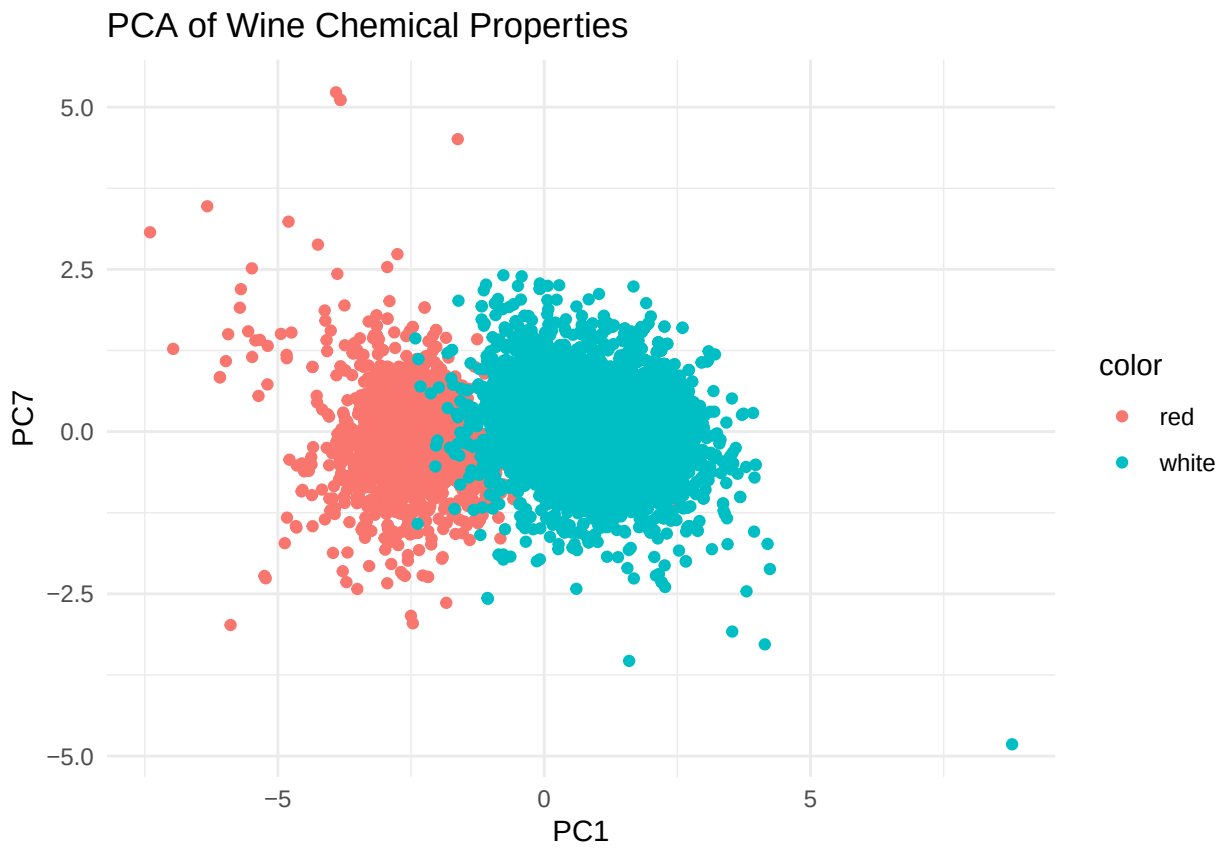

```r
# Plot PCA results: 1 and 5
ggplot(pca_result, aes(x = PC1, y = PC5, color = color)) +
  geom_point() +
  ggtitle("PCA of Wine Chemical Properties") +
  theme_minimal()
```



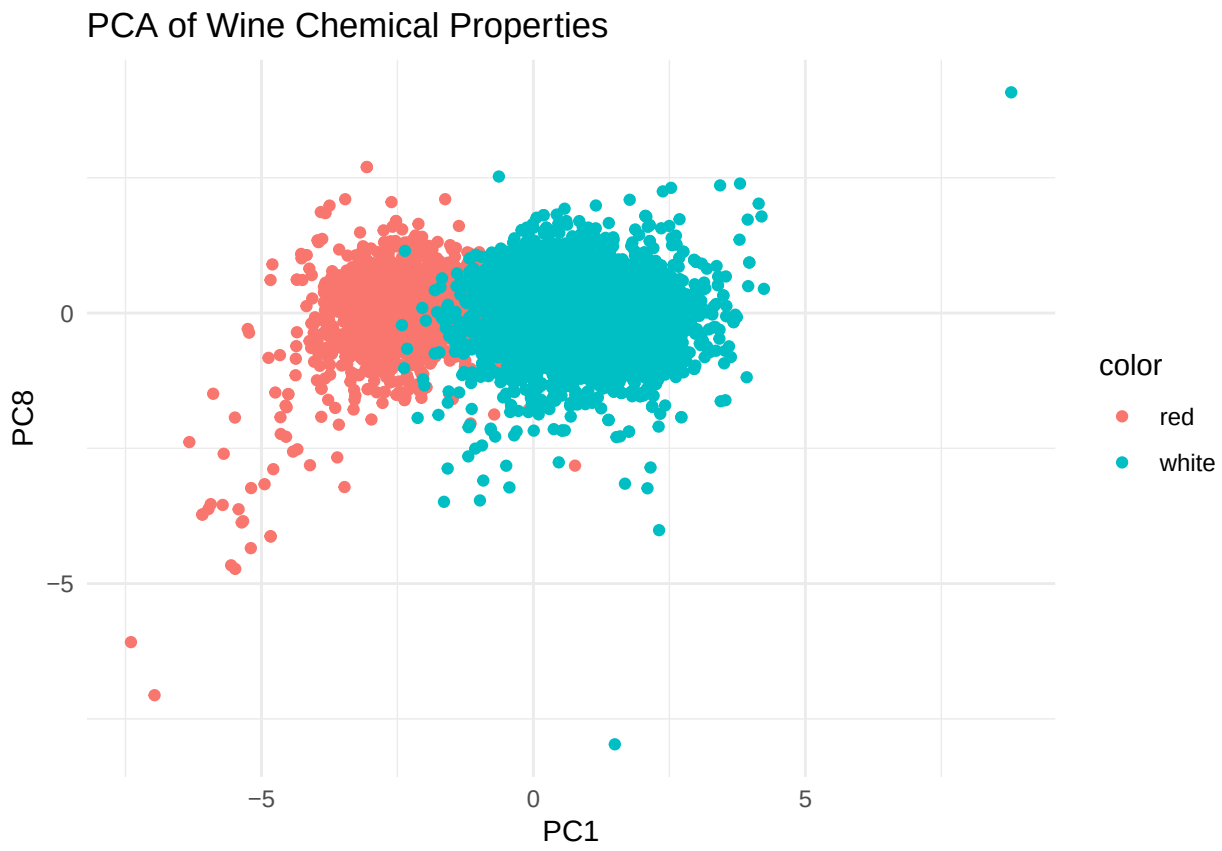

```r
# Plot PCA results: 1 and 6
ggplot(pca_result, aes(x = PC1, y = PC6, color = color)) +
  geom_point() +
  ggtitle("PCA of Wine Chemical Properties") +
  theme_minimal()
```



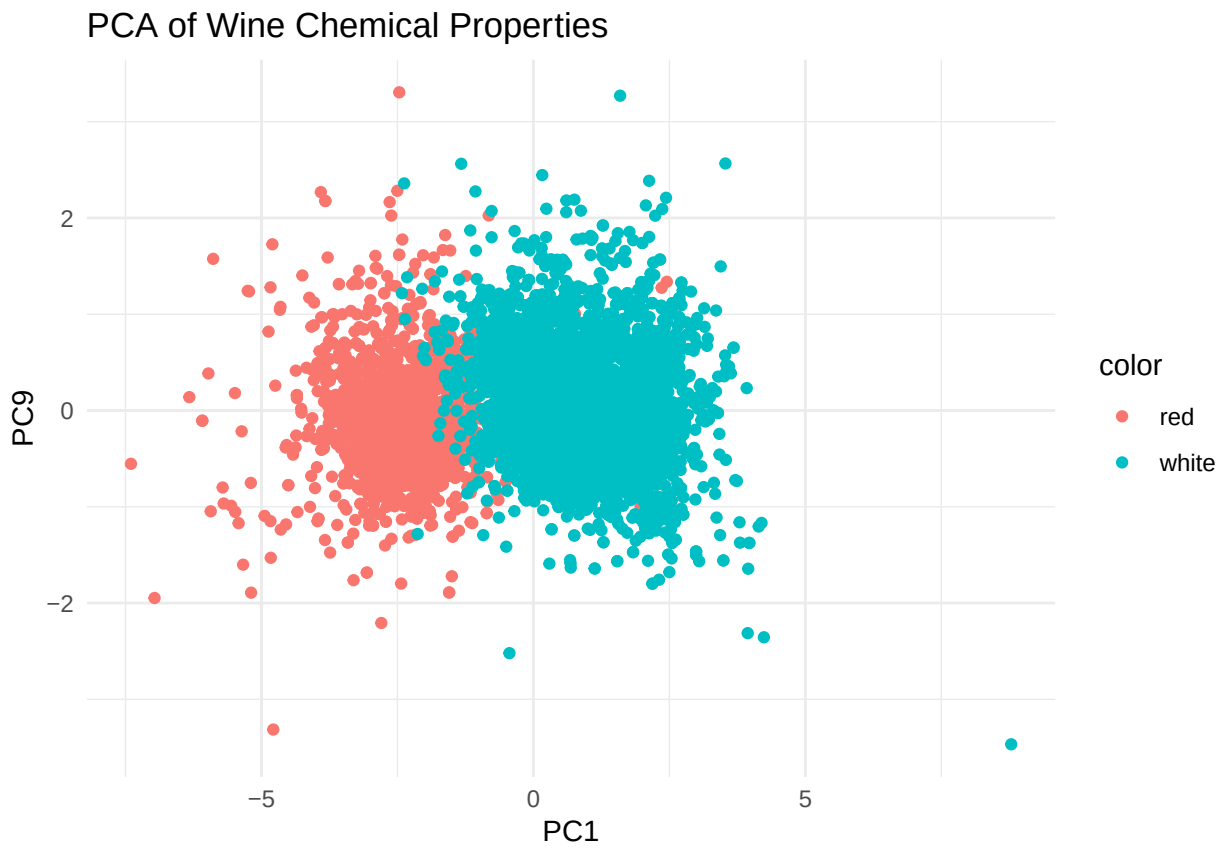

```r
# Plot PCA results: 1 and 7
ggplot(pca_result, aes(x = PC1, y = PC7, color = color)) +
  geom_point() +
  ggtitle("PCA of Wine Chemical Properties") +
  theme_minimal()
```



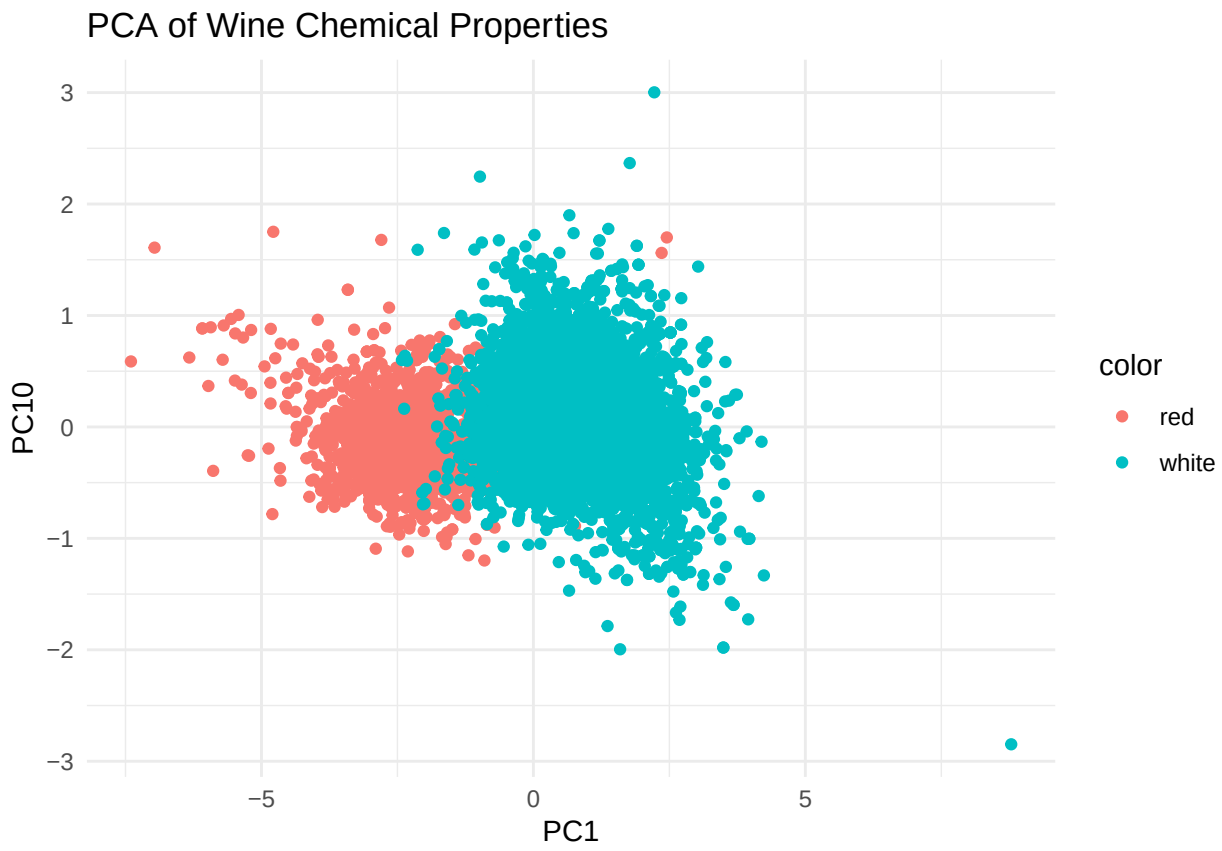

```r
# Plot PCA results: 1 and 8
ggplot(pca_result, aes(x = PC1, y = PC8, color = color)) +
  geom_point() +
  ggtitle("PCA of Wine Chemical Properties") +
  theme_minimal()
```



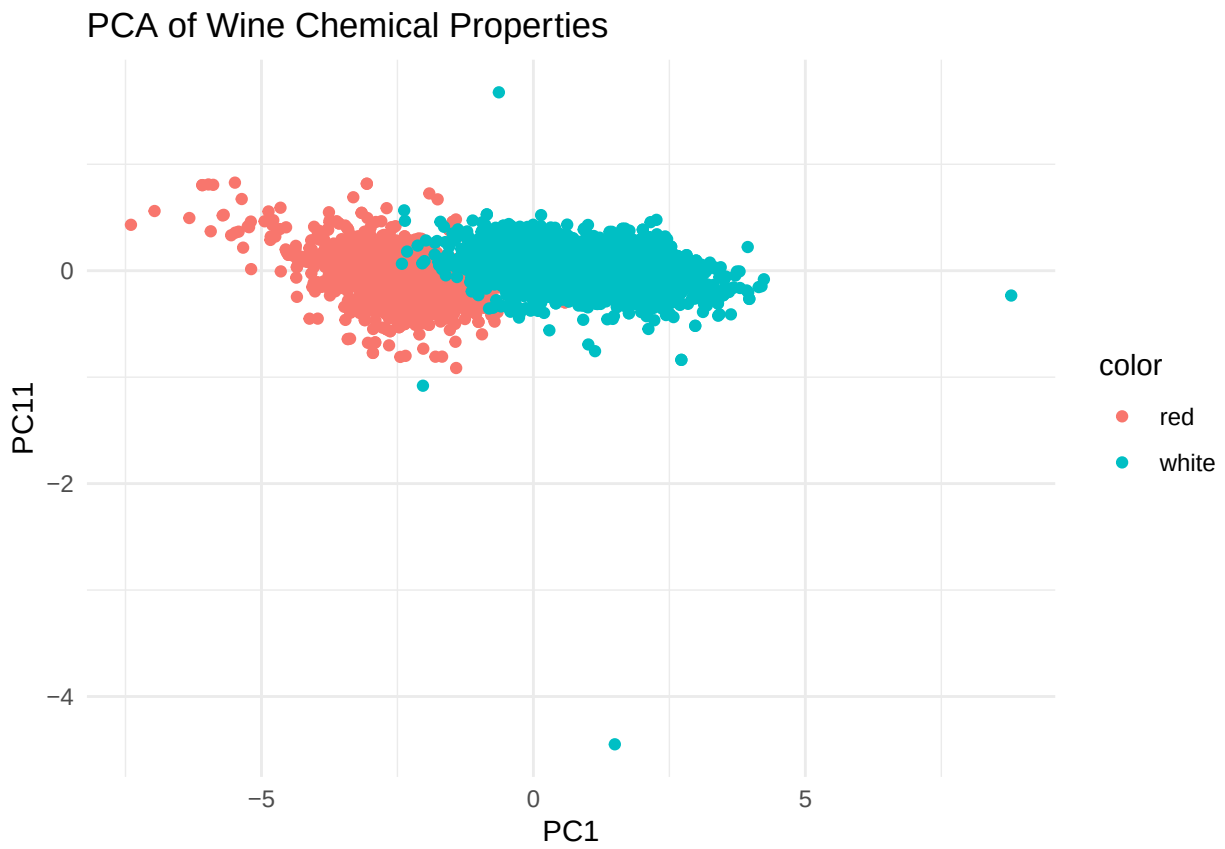

```r
# Plot PCA results: 1 and 9
ggplot(pca_result, aes(x = PC1, y = PC9, color = color)) +
  geom_point() +
  ggtitle("PCA of Wine Chemical Properties") +
  theme_minimal()
```




```r
# Plot PCA results: 1 and 10
ggplot(pca_result, aes(x = PC1, y = PC10, color = color)) +
  geom_point() +
  ggtitle("PCA of Wine Chemical Properties") +
  theme_minimal()
```




```r
# Plot PCA results: 1 and 11
ggplot(pca_result, aes(x = PC1, y = PC11, color = color)) +
  geom_point() +
  ggtitle("PCA of Wine Chemical Properties") +
  theme_minimal()
```



Given the summary of the different principal components above, we can say that this does a decent job. The first principal component does an excellent job separating red and white wines. It also captures 27.54% of the variability, and the first two capture 50.21%.

tSNE

```
# Remove duplicate rows from the chemical properties
chemical_properties_unique <- unique(chemical_properties)

# Find the row indices of the unique rows in the original data
unique_indices <- which(duplicated(chemical_properties) == FALSE)

# Perform t-SNE on the unique rows
tsne <- Rtsne(chemical_properties_unique, dims = 2, perplexity = 30, verbose = TRUE, max_iter = 500)
```

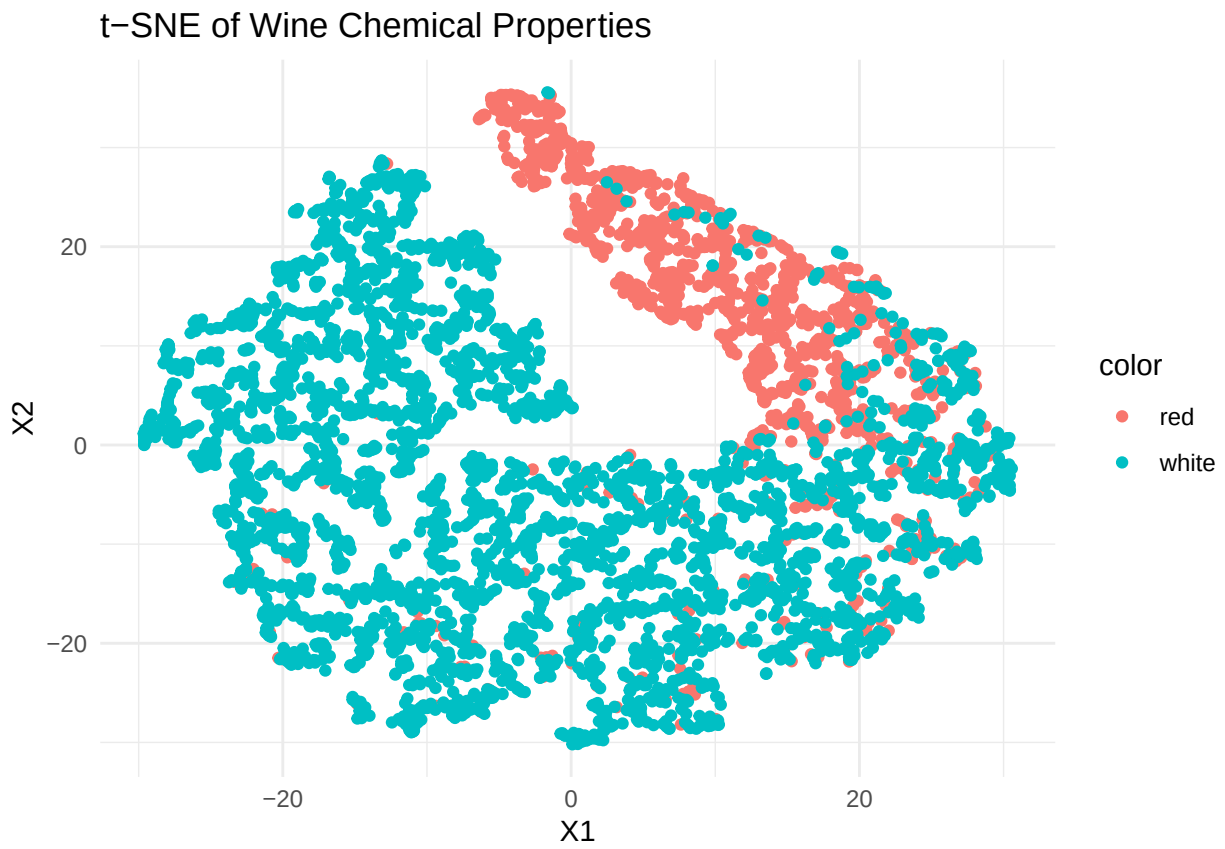
```
## Performing PCA
## Read the 5318 x 11 data matrix successfully!
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 0.20 seconds (sparsity = 0.020605)!
## Learning embedding...
## Iteration 50: error is 91.022945 (50 iterations in 0.38 seconds)
## Iteration 100: error is 72.663843 (50 iterations in 0.37 seconds)
```

```
## Iteration 150: error is 69.504792 (50 iterations in 0.37 seconds)
## Iteration 200: error is 68.316070 (50 iterations in 0.42 seconds)
## Iteration 250: error is 67.765489 (50 iterations in 0.40 seconds)
## Iteration 300: error is 2.048021 (50 iterations in 0.37 seconds)
## Iteration 350: error is 1.628113 (50 iterations in 0.37 seconds)
## Iteration 400: error is 1.407323 (50 iterations in 0.37 seconds)
## Iteration 450: error is 1.272880 (50 iterations in 0.38 seconds)
## Iteration 500: error is 1.183611 (50 iterations in 0.38 seconds)
## Fitting performed in 3.80 seconds.
```

```
tsne_result <- data.frame(tsne$Y)
tsne_result$color <- wine_data$color[unique_indices]
summary(tsne)
```

```
##                      Length Class  Mode
## N                        1  -none- numeric
## Y                    10636  -none- numeric
## costs                 5318  -none- numeric
## itercosts               10  -none- numeric
## origD                    1  -none- numeric
## perplexity               1  -none- numeric
## theta                    1  -none- numeric
## max_iter                 1  -none- numeric
## stop_lying_iter          1  -none- numeric
## mom_switch_iter          1  -none- numeric
## momentum                 1  -none- numeric
## final_momentum           1  -none- numeric
## eta                      1  -none- numeric
## exaggeration_factor      1  -none- numeric
```

```
# Plot t-SNE results
ggplot(tsne_result, aes(x = X1, y = X2, color = color)) +
  geom_point() +
  ggtitle("t-SNE of Wine Chemical Properties") +
  theme_minimal()
```



tSNE is another method for reducing the dimensionality of the data set. We use it to reduce the data to two components by setting the dims = 2. While the result is not as good as the one for PCA, it still does split the data relatively well. While there is some overlap in the data, there is a distinct grouping of the data between the red and white wines.

Hierarchical Clustering

```
wine_data <- read.csv("/Users/navyasinghal/Desktop/Summer/Intro to Machine Learning/Unsupervised Learni

# Drop the 'quality' and 'color' columns to work with chemical components only
chemical_data <- wine_data[, -which(names(wine_data) %in% c("quality", "color"))]

# Compute the distance matrix and perform hierarchical clustering
distance_matrix <- dist(chemical_data)
hclust_result <- hclust(distance_matrix, method = "ward.D2")

# Cut the dendrogram to obtain 2 clusters (assuming two colors: red and white)
wine_data$cluster <- cutree(hclust_result, k = 2)

# Convert the cluster assignments to a factor
wine_data$cluster <- as.factor(wine_data$cluster)

# Compare clusters with actual wine color
table(wine_data$color, wine_data$cluster)
```

```
##
##            1    2
##   red   1561   38
##   white 2203 2695
```

We will now perform hierarchical clustering to determine the wine color in the data. It seems to do an excellent job separating red wines into specific clusters, but it is unsuitable for white wines as they are evenly distributed.

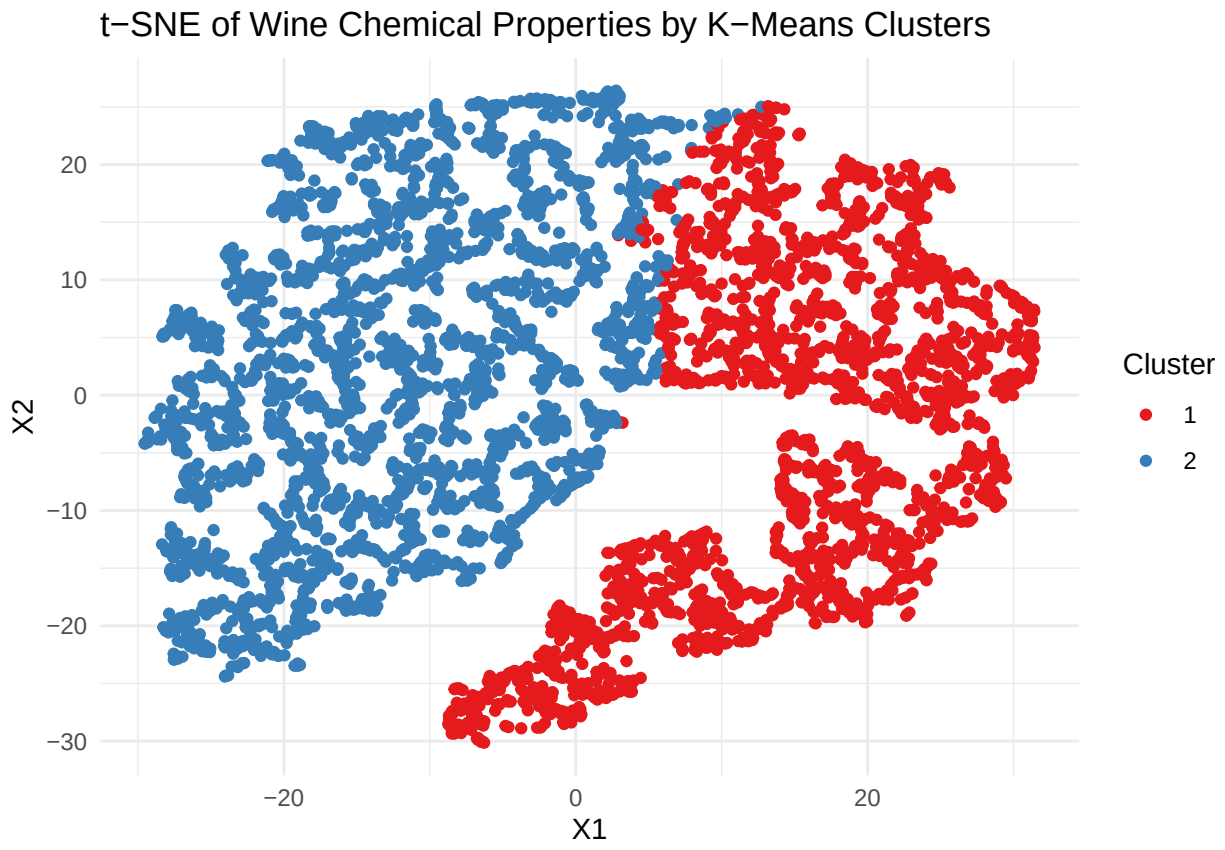
K-Means Clustering

```
# Load the dataset
wine_data <- read.csv("/Users/navyasinghal/Desktop/Summer/Intro to Machine Learning/Unsupervised Learni

# Select only the chemical properties for analysis
chemical_properties <- wine_data[, 1:11]

# Perform K-Means clustering on the original chemical properties (with duplicates)
set.seed(42)
kmeans_result <- kmeans(chemical_properties, centers = 2, nstart = 25)  # Assuming 2 clusters for red/w

# Add the cluster assignments to the original data
wine_data$Cluster <- as.factor(kmeans_result$cluster)

# Remove duplicate rows from the chemical properties for t-SNE
chemical_properties_unique <- unique(chemical_properties)

# Perform t-SNE on the unique chemical properties
tsne <- Rtsne(chemical_properties_unique, dims = 2, perplexity = 30, verbose = TRUE, max_iter = 500)
```

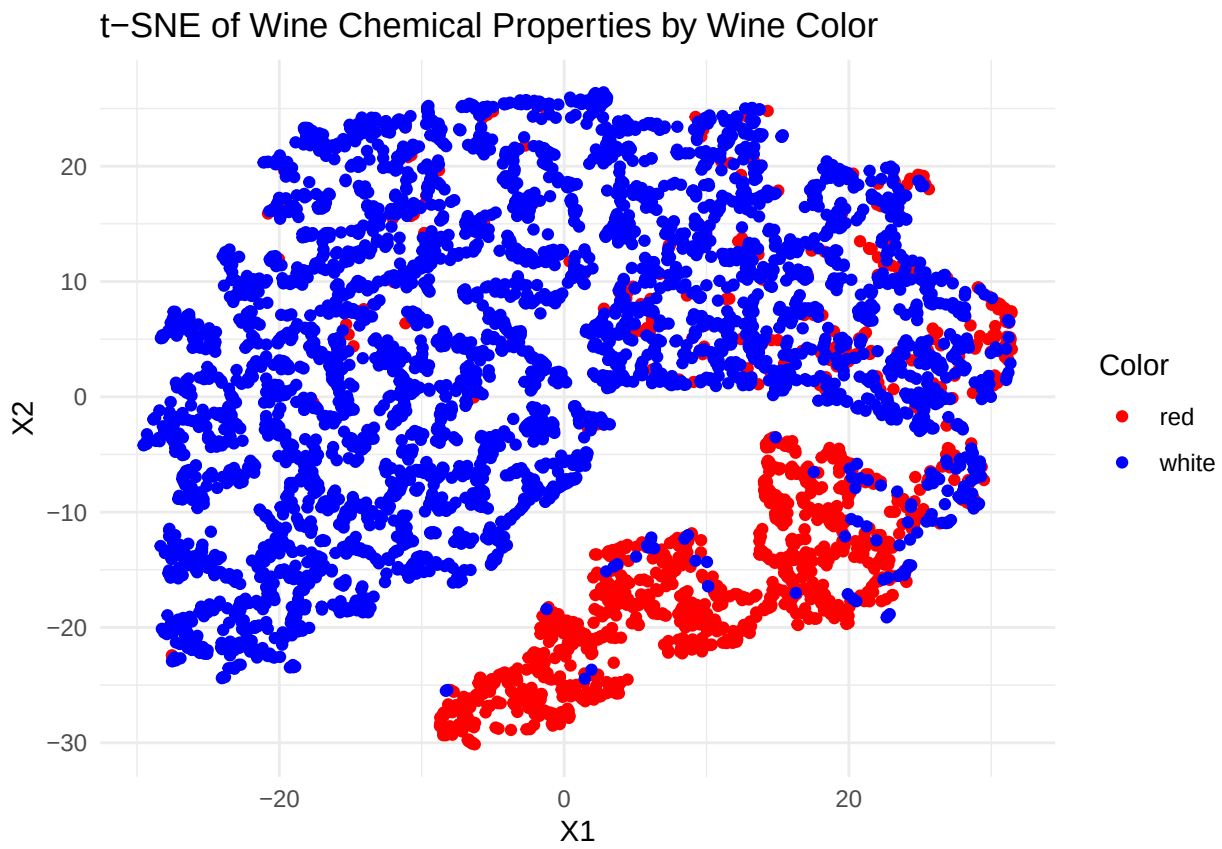
```
## Performing PCA
## Read the 5318 x 11 data matrix successfully!
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 0.20 seconds (sparsity = 0.020605)!
## Learning embedding...
## Iteration 50: error is 91.028589 (50 iterations in 0.42 seconds)
## Iteration 100: error is 72.754485 (50 iterations in 0.48 seconds)
## Iteration 150: error is 69.422538 (50 iterations in 0.42 seconds)
## Iteration 200: error is 68.329199 (50 iterations in 0.40 seconds)
## Iteration 250: error is 67.800171 (50 iterations in 0.39 seconds)
## Iteration 300: error is 2.059270 (50 iterations in 0.37 seconds)
## Iteration 350: error is 1.644757 (50 iterations in 0.36 seconds)
## Iteration 400: error is 1.420505 (50 iterations in 0.36 seconds)
## Iteration 450: error is 1.285843 (50 iterations in 0.36 seconds)
## Iteration 500: error is 1.195657 (50 iterations in 0.36 seconds)
## Fitting performed in 3.93 seconds.
```

```
tsne_result <- data.frame(tsne$Y)
```

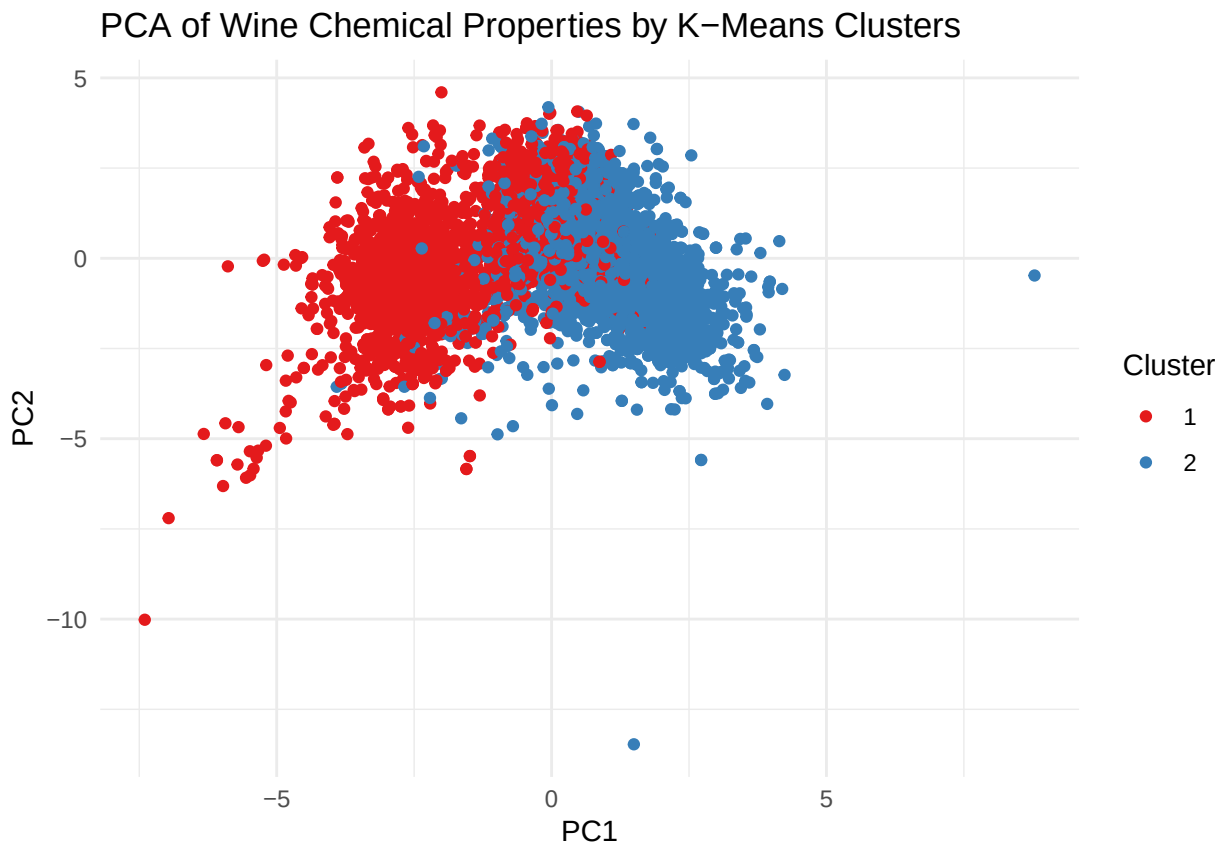
```r
# Map the clusters back to the t-SNE results using row indices
unique_indices <- which(duplicated(chemical_properties) == FALSE)
tsne_result$Cluster <- wine_data$Cluster[unique_indices]
tsne_result$Color <- wine_data$color[unique_indices]

# Plot t-SNE results, colored by K-Means clusters
ggplot(tsne_result, aes(x = X1, y = X2, color = Cluster)) +
  geom_point() +
  scale_color_brewer(palette = "Set1") +
  ggtitle("t-SNE of Wine Chemical Properties by K-Means Clusters") +
  theme_minimal()
```



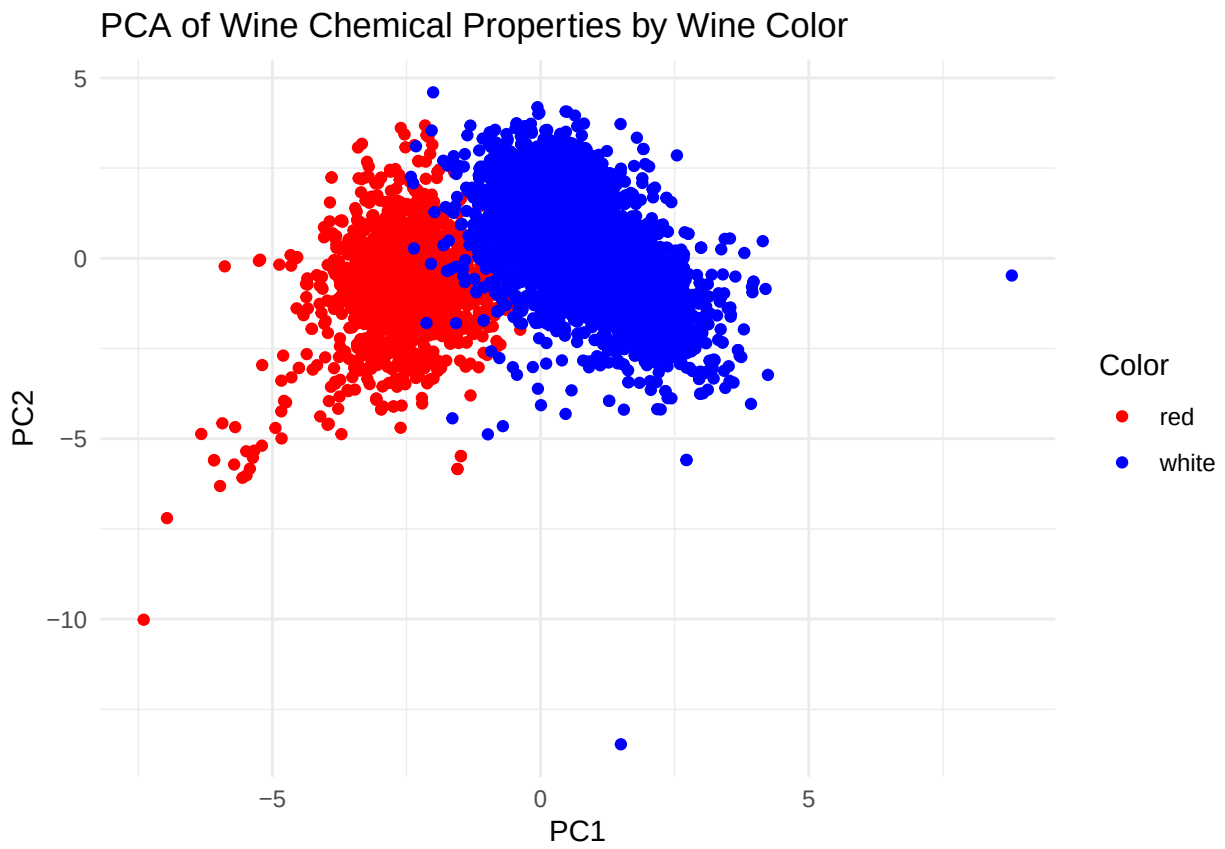

```r
# Plot t-SNE results, colored by actual wine color
ggplot(tsne_result, aes(x = X1, y = X2, color = Color)) +
  geom_point() +
  scale_color_manual(values = c("red" = "red", "white" = "blue")) +
  ggtitle("t-SNE of Wine Chemical Properties by Wine Color") +
  theme_minimal()
```




```r
# Perform PCA on the original chemical properties to visualize the clusters
pca <- prcomp(chemical_properties, center = TRUE, scale. = TRUE)
pca_result <- data.frame(pca$x[, 1:2])
pca_result$Cluster <- wine_data$Cluster
pca_result$Color <- wine_data$color

# Plot PCA results, colored by K-Means clusters
ggplot(pca_result, aes(x = PC1, y = PC2, color = Cluster)) +
  geom_point() +
  scale_color_brewer(palette = "Set1") +
  ggtitle("PCA of Wine Chemical Properties by K-Means Clusters") +
  theme_minimal()
```




```
# Plot PCA results, colored by actual wine color
ggplot(pca_result, aes(x = PC1, y = PC2, color = Color)) +
  geom_point() +
  scale_color_manual(values = c("red" = "red", "white" = "blue")) +
  ggtitle("PCA of Wine Chemical Properties by Wine Color") +
  theme_minimal()
```



We visualize how well we do with k-means clustering by mapping in 2D using the first two Principal Components and the two dimensions of tSNE.

Based on the graphs above, I would argue that PCA best separates the wine into colors. K-means clustering will distribute the data evenly and seems very similar to PCA, but it is much more incorrect when looking at the tSNE. tSNE seems like the second best option with clear white and red groups, but it overlaps most parts of each group, even if only a few points, while PCA only overlaps in the border region.

Now for the Quality

This analysis will be similar to the above exploration but will instead focus on the quality of the wine rather than the color. A significant difference will be the clustering groups. It will now be categorized into seven groups since there are seven quality measurements quality measurements, one for each quality out of 10. No wine was rated a quality of 1, 2, or 10.

Set Up

```
# Load necessary libraries
library(ggplot2)
library(Rtsne)
library(cluster)

# Load the dataset
wine_data <- read.csv("/Users/navyasinghal/Desktop/Summer/Intro to Machine Learning/Unsupervised Learni
```

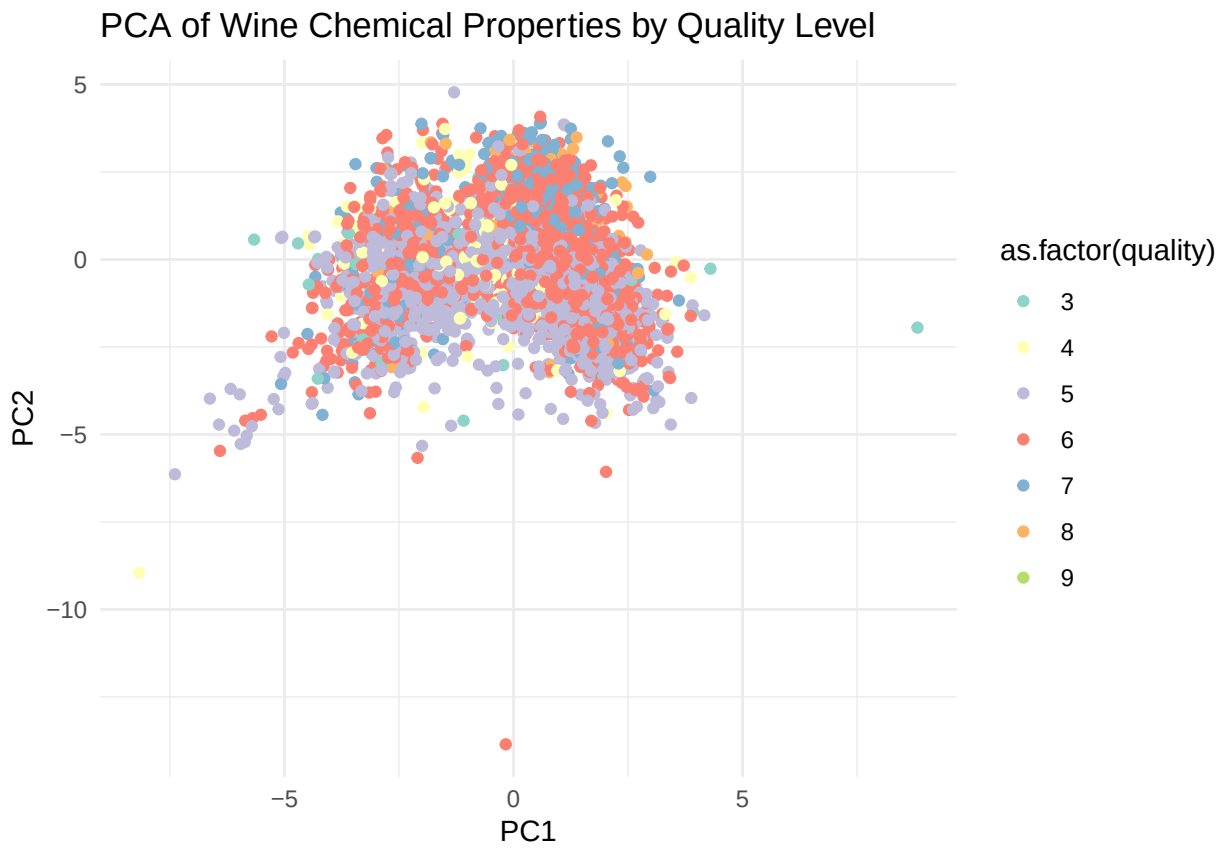
```r
# Select only the chemical properties for analysis
chemical_properties <- wine_data[, 1:11]

# Remove duplicate rows from the chemical properties
chemical_properties_unique <- unique(chemical_properties)

# Find the row indices of the unique rows in the original data
unique_indices <- which(duplicated(chemical_properties) == FALSE)
```

PCA

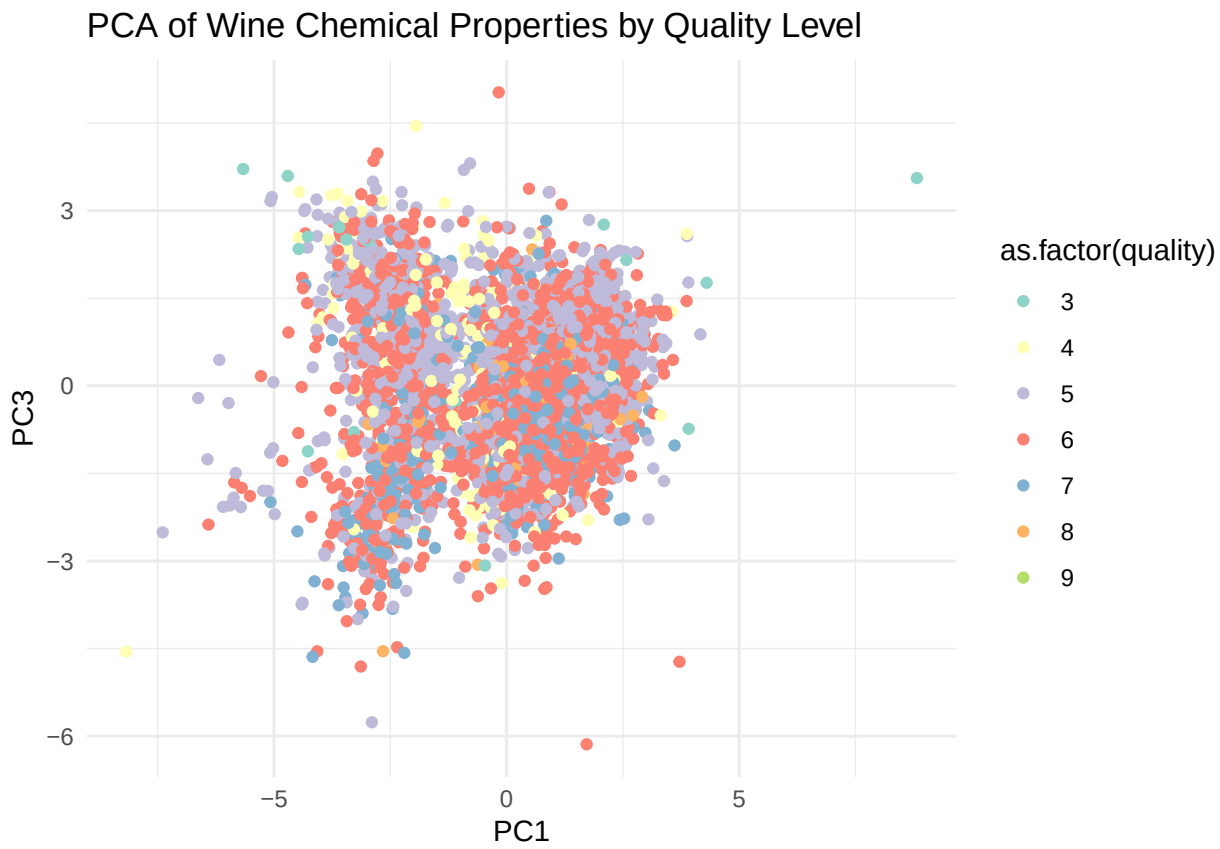
```r
# Perform PCA
pca <- prcomp(chemical_properties_unique, center = TRUE, scale. = TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5    PC6     PC7
## Standard deviation     1.7287 1.5732 1.2599 0.97648 0.86155 0.7919 0.72181
## Proportion of Variance 0.2717 0.2250 0.1443 0.08668 0.06748 0.0570 0.04736
## Cumulative Proportion  0.2717 0.4967 0.6410 0.72766 0.79514 0.8521 0.89951
##                           PC8     PC9    PC10    PC11
## Standard deviation     0.71251 0.58026 0.47532 0.18731
## Proportion of Variance 0.04615 0.03061 0.02054 0.00319
## Cumulative Proportion  0.94566 0.97627 0.99681 1.00000
```

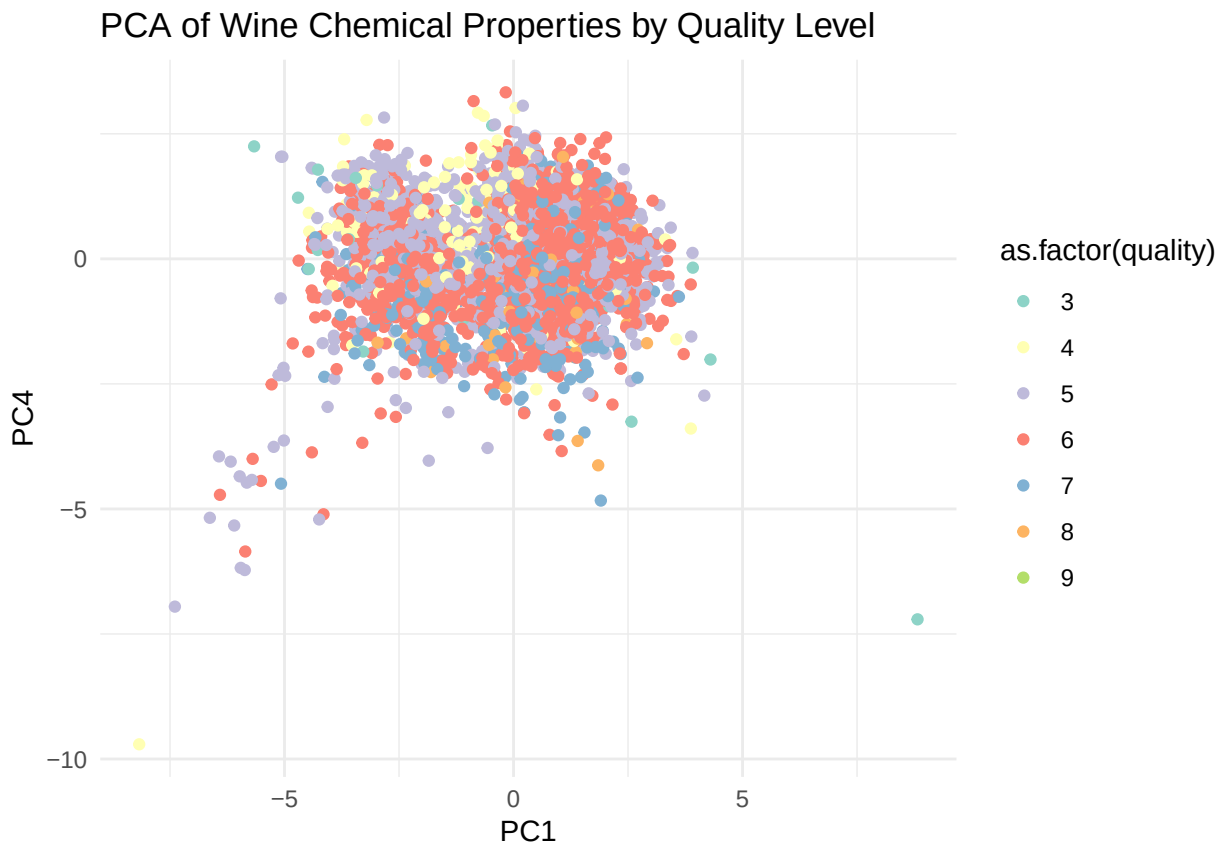
```r
pca_result <- data.frame(pca$x[, 1:11])
pca_result$quality <- wine_data$quality[unique_indices]

# Plot PCA results, colored by wine quality: 1 and 2
ggplot(pca_result, aes(x = PC1, y = PC2, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("PCA of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```



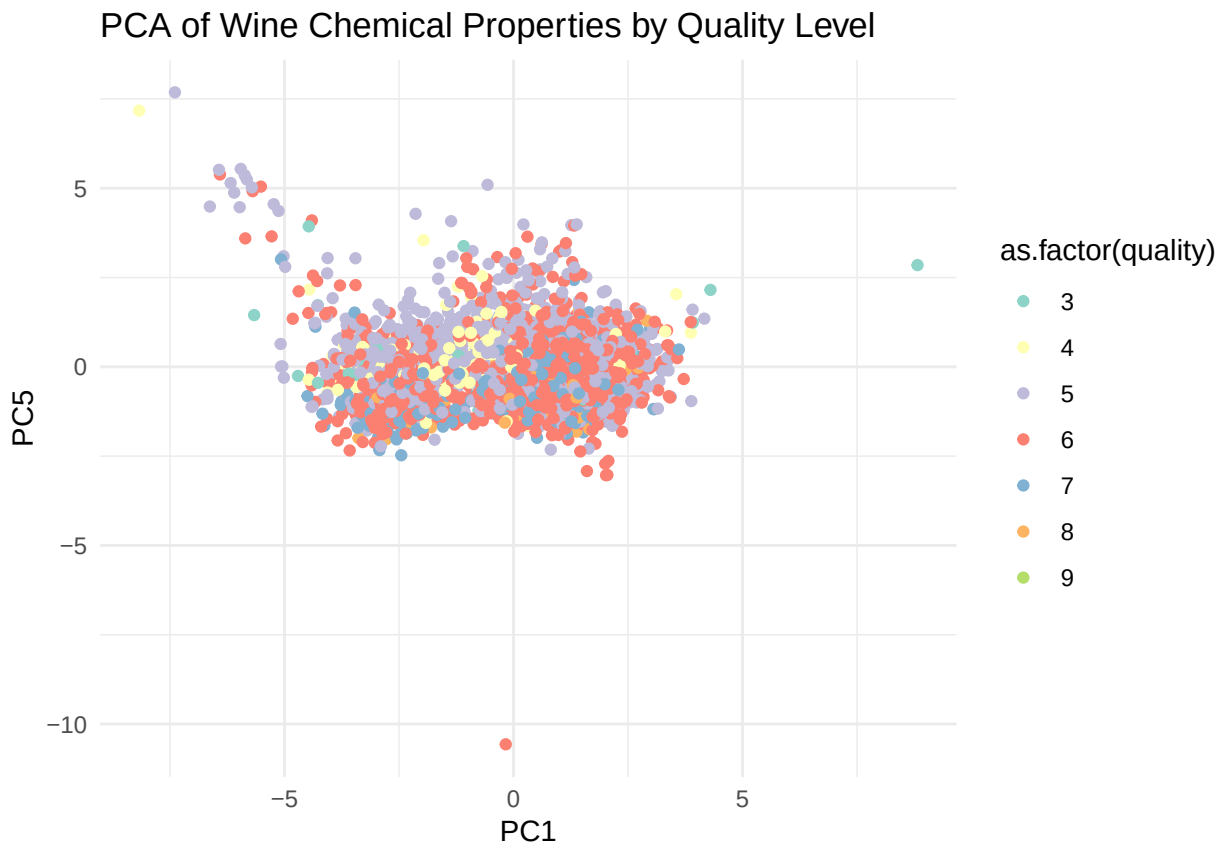

```r
# Plot PCA results, colored by wine quality: 1 and 3
ggplot(pca_result, aes(x = PC1, y = PC3, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("PCA of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```



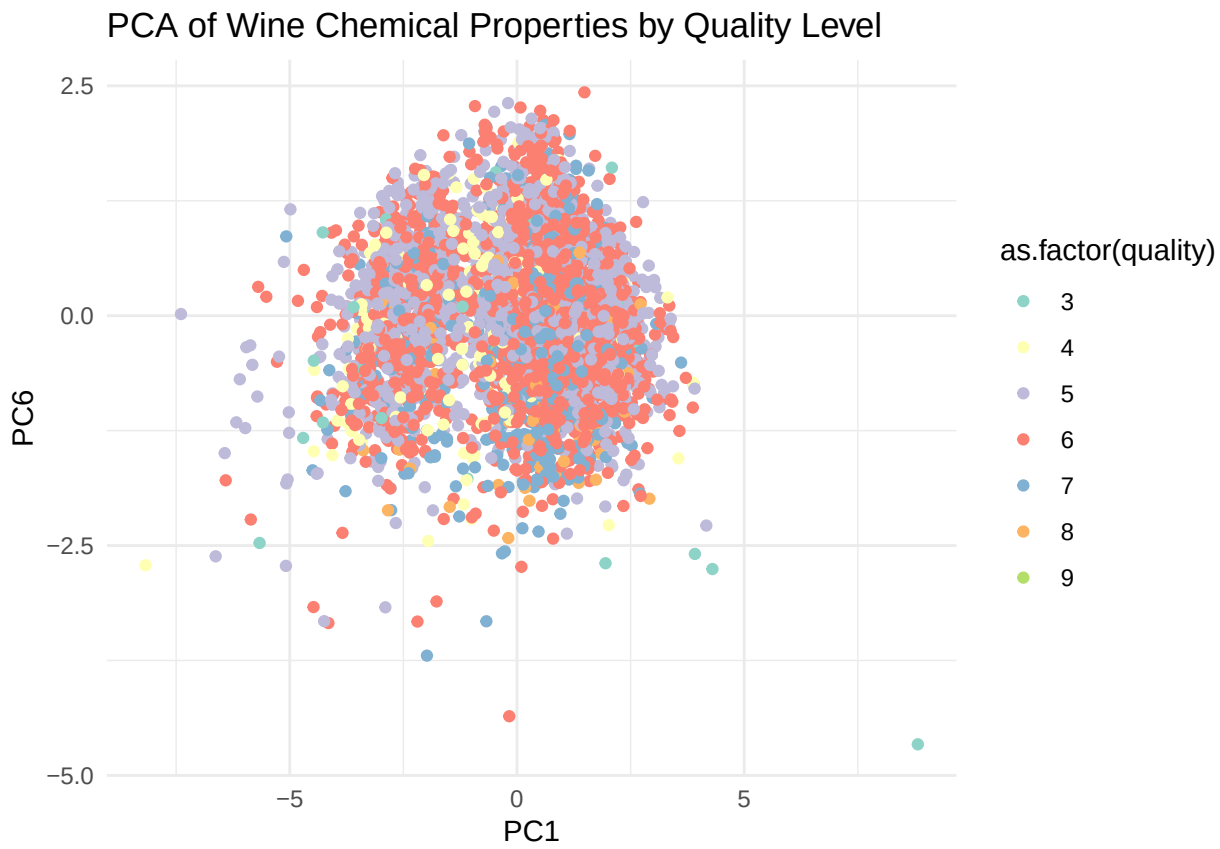

```r
# Plot PCA results, colored by wine quality: 1 and 4
ggplot(pca_result, aes(x = PC1, y = PC4, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("PCA of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```



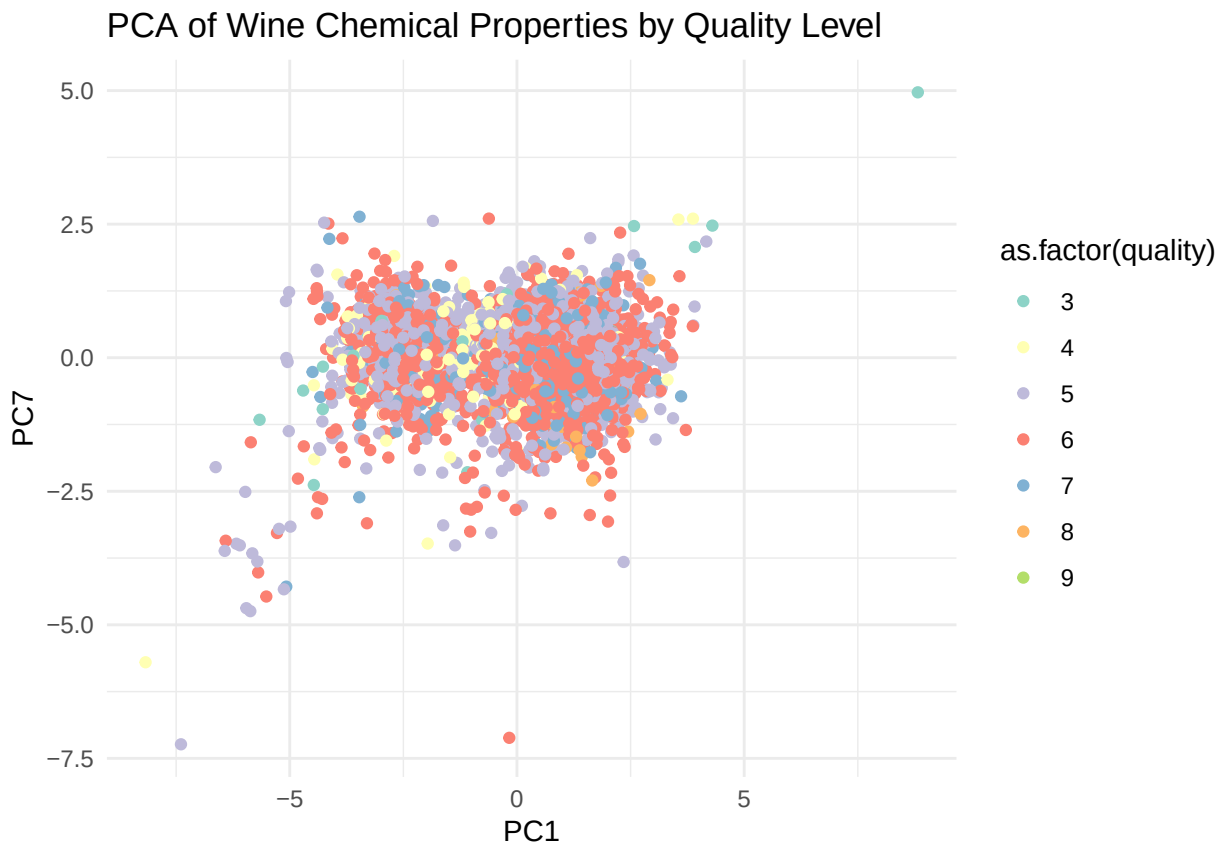

```r
# Plot PCA results, colored by wine quality: 1 and 5
ggplot(pca_result, aes(x = PC1, y = PC5, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("PCA of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```



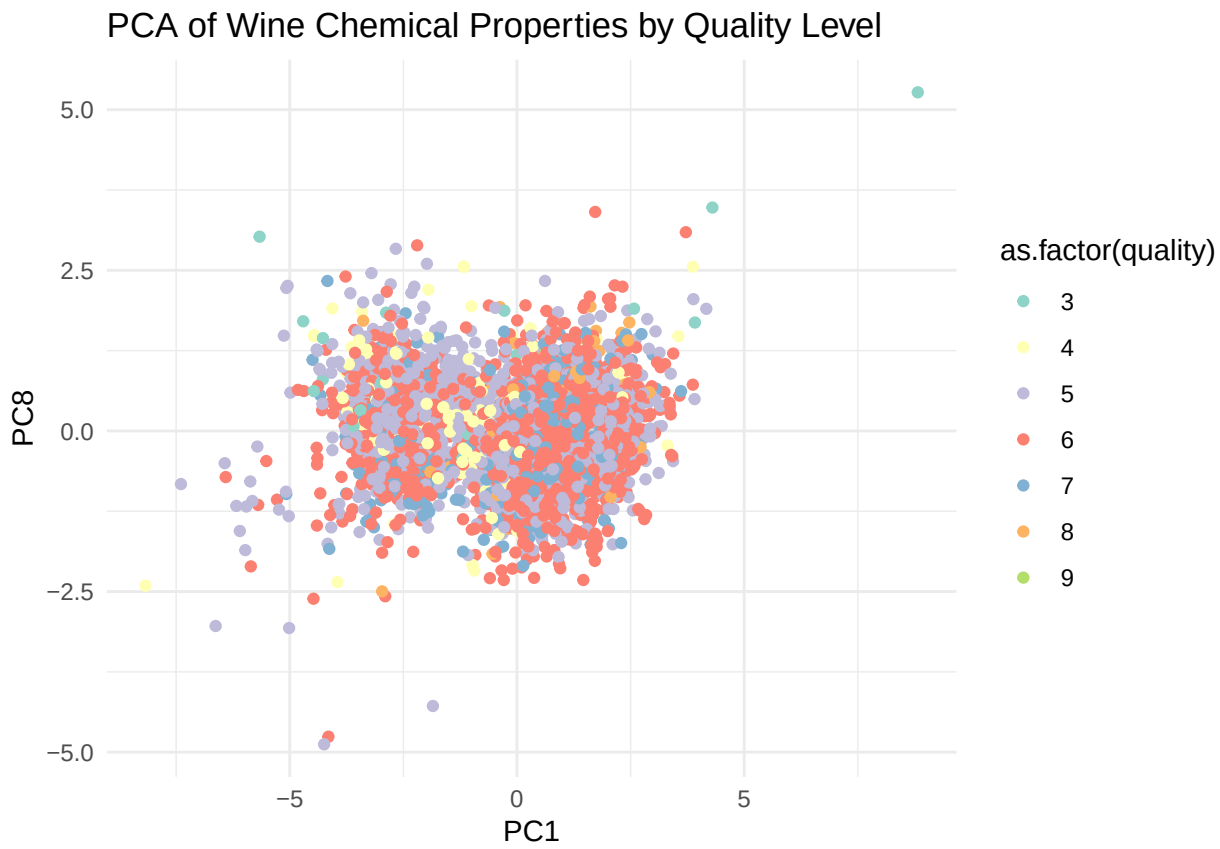

```r
# Plot PCA results, colored by wine quality: 1 and 6
ggplot(pca_result, aes(x = PC1, y = PC6, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("PCA of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```



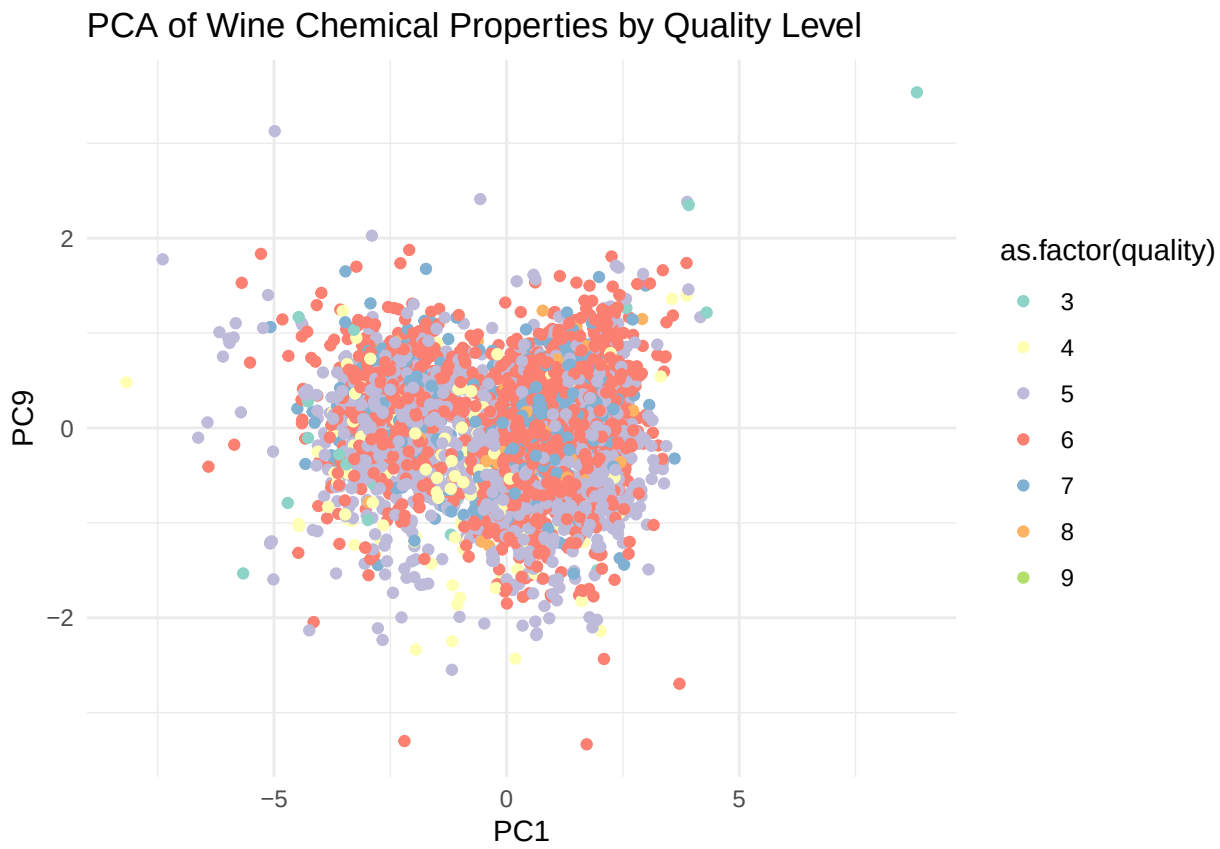

```
# Plot PCA results, colored by wine quality: 1 and 7
ggplot(pca_result, aes(x = PC1, y = PC7, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("PCA of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```



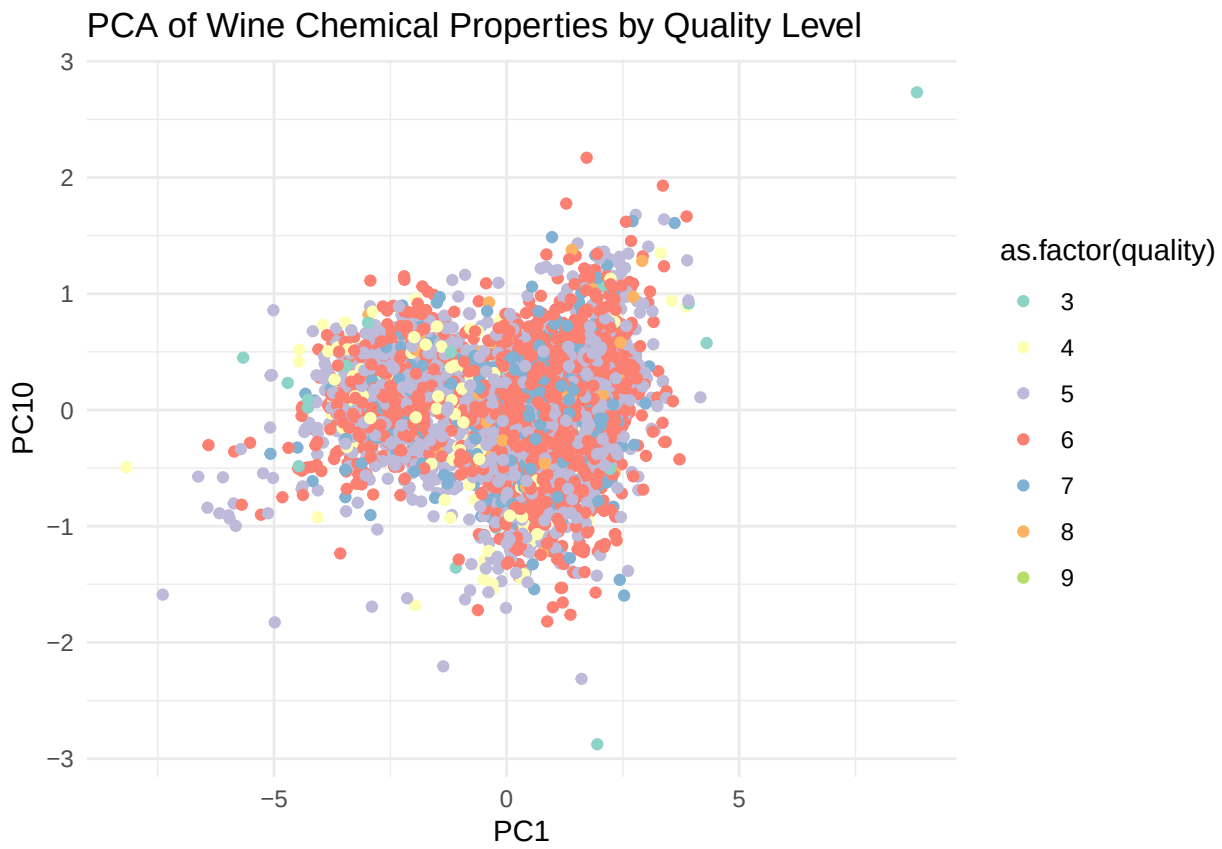

```r
# Plot PCA results, colored by wine quality: 1 and 8
ggplot(pca_result, aes(x = PC1, y = PC8, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("PCA of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```



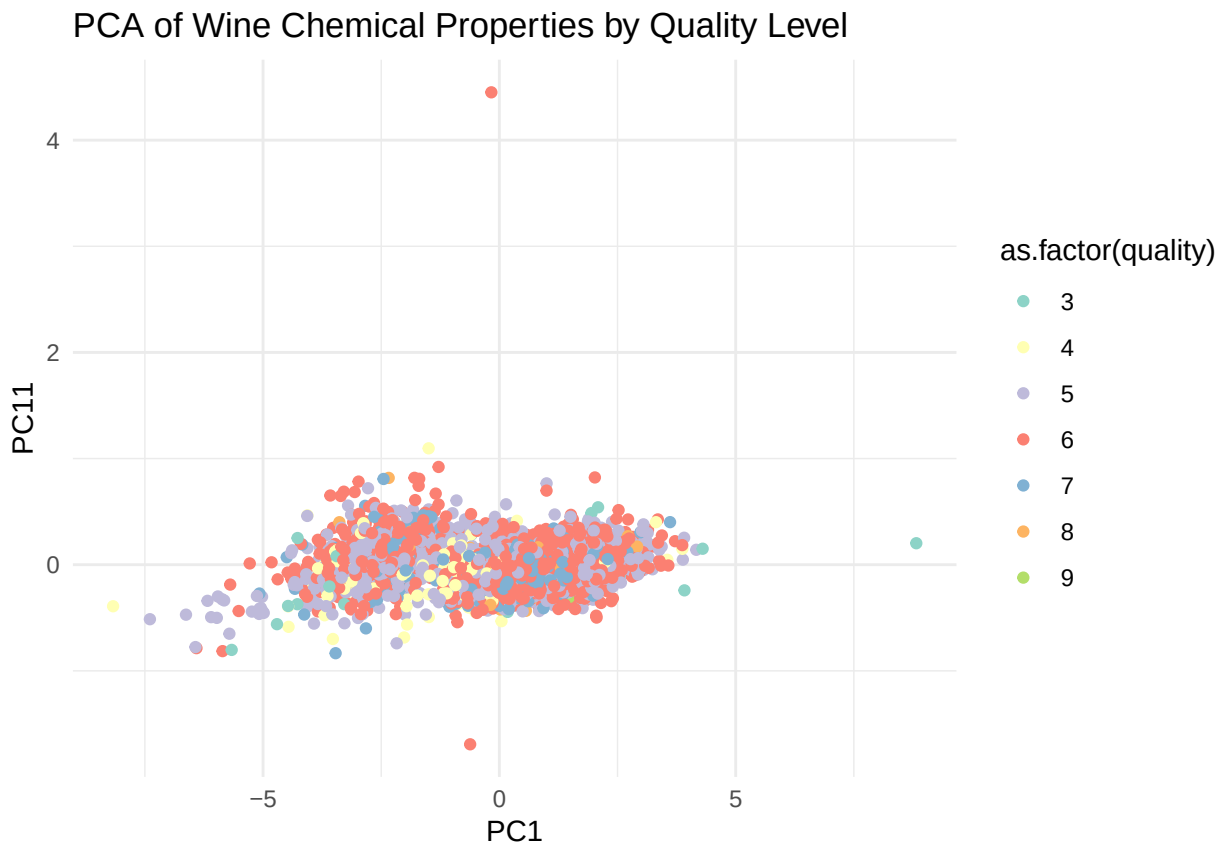

```r
# Plot PCA results, colored by wine quality: 1 and 9
ggplot(pca_result, aes(x = PC1, y = PC9, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("PCA of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```




```r
# Plot PCA results, colored by wine quality: 1 and 10
ggplot(pca_result, aes(x = PC1, y = PC10, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("PCA of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```




```r
# Plot PCA results, colored by wine quality: 1 and 11
ggplot(pca_result, aes(x = PC1, y = PC11, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("PCA of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```



The first two components do explain 49.67%, almost as well as the explanation for color, of the data's variability, but as you can see in the graphs above, this does not separate out into quality well. None of the principal components gives back defined quality groups.

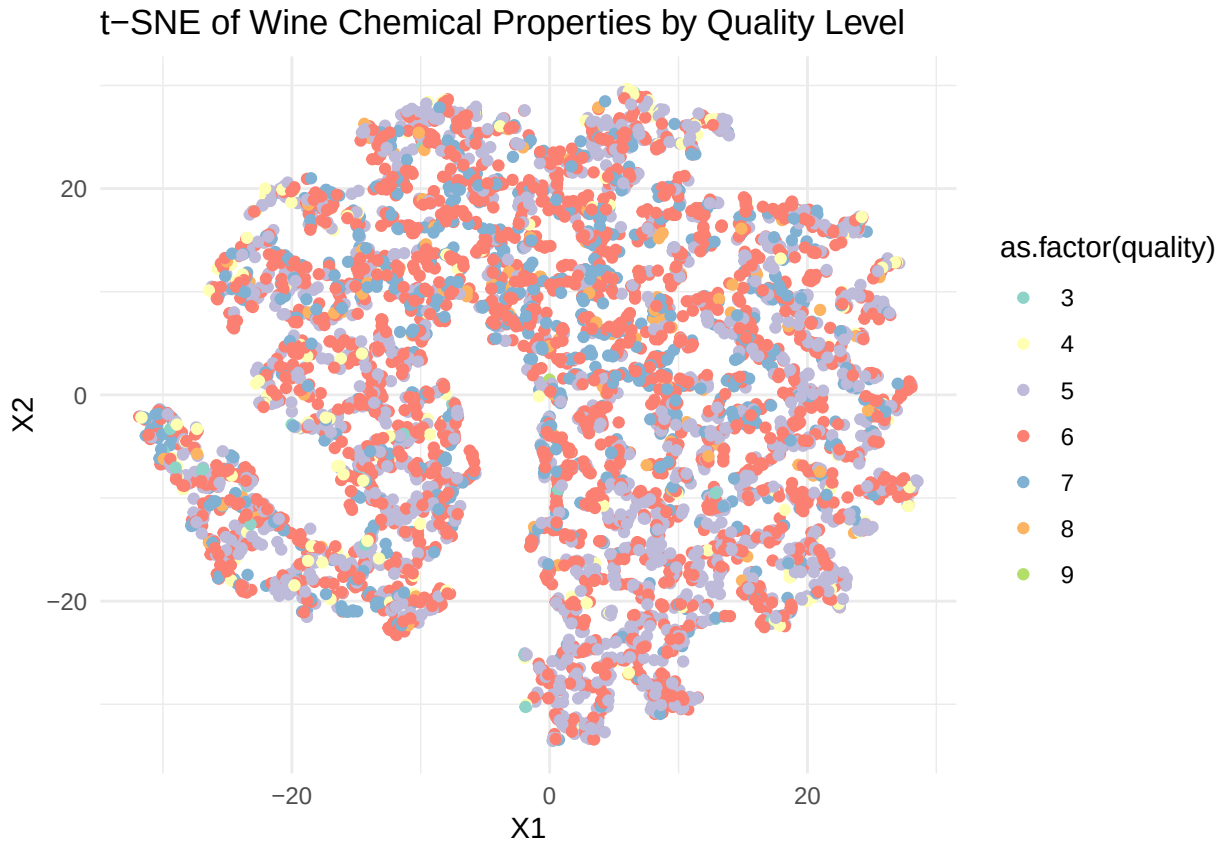
tSNE

```
# Remove duplicate rows from the chemical properties
chemical_properties_unique <- unique(chemical_properties)

# Find the row indices of the unique rows in the original data
unique_indices <- which(duplicated(chemical_properties) == FALSE)

# Perform t-SNE on the unique rows
tsne <- Rtsne(chemical_properties_unique, dims = 2, perplexity = 30, verbose = TRUE, max_iter = 500)
```

```
## Performing PCA
## Read the 5318 x 11 data matrix successfully!
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 0.21 seconds (sparsity = 0.020605)!
## Learning embedding...
## Iteration 50: error is 91.059187 (50 iterations in 0.40 seconds)
## Iteration 100: error is 71.807703 (50 iterations in 0.38 seconds)
```

```
## Iteration 150: error is 68.779084 (50 iterations in 0.38 seconds)
## Iteration 200: error is 67.897360 (50 iterations in 0.39 seconds)
## Iteration 250: error is 67.460341 (50 iterations in 0.40 seconds)
## Iteration 300: error is 2.017718 (50 iterations in 0.39 seconds)
## Iteration 350: error is 1.608499 (50 iterations in 0.36 seconds)
## Iteration 400: error is 1.391751 (50 iterations in 0.36 seconds)
## Iteration 450: error is 1.261399 (50 iterations in 0.37 seconds)
## Iteration 500: error is 1.176842 (50 iterations in 0.37 seconds)
## Fitting performed in 3.81 seconds.
```

```r
tsne_result <- data.frame(tsne$Y)
tsne_result$quality <- wine_data$quality[unique_indices]

# Plot t-SNE results, colored by wine quality
ggplot(tsne_result, aes(x = X1, y = X2, color = as.factor(quality))) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  ggtitle("t-SNE of Wine Chemical Properties by Quality Level") +
  theme_minimal()
```



As you can see above, t-SNE is also not a great way to separate wine quality using chemical properties. The data does not appear to show any distinct groups.

Hierarchical Clustering

```
# Load the dataset
wine_data <- read.csv("/Users/navyasinghal/Desktop/Summer/Intro to Machine Learning/Unsupervised Learni

# Drop the 'color' column to work with chemical components and 'quality'
chemical_data <- wine_data[, -which(names(wine_data) == "color")]

# Compute the distance matrix and perform hierarchical clustering
distance_matrix <- dist(chemical_data[, -which(names(wine_data) == "quality")])  # Exclude quality from
hclust_result <- hclust(distance_matrix, method = "ward.D2")

# Choose the number of clusters (you can experiment with different numbers)
num_clusters <- 7
wine_data$cluster <- cutree(hclust_result, k = num_clusters)

# Convert the cluster assignments to a factor
wine_data$cluster <- as.factor(wine_data$cluster)

# Compare clusters with actual wine quality
table(wine_data$quality, wine_data$cluster)
```

```
##
##       1   2   3   4   5   6   7
##   3   9   5   4   2   1   5   4
##   4  38  47  43  31  21   4  32
##   5 248 346 356 561 168  82 377
##   6 301 425 617 764 310  70 349
##   7 123 112 297 311 168  10  58
##   8  12  14  66  56  30   0  15
##   9   0   0   2   2   1   0   0
```

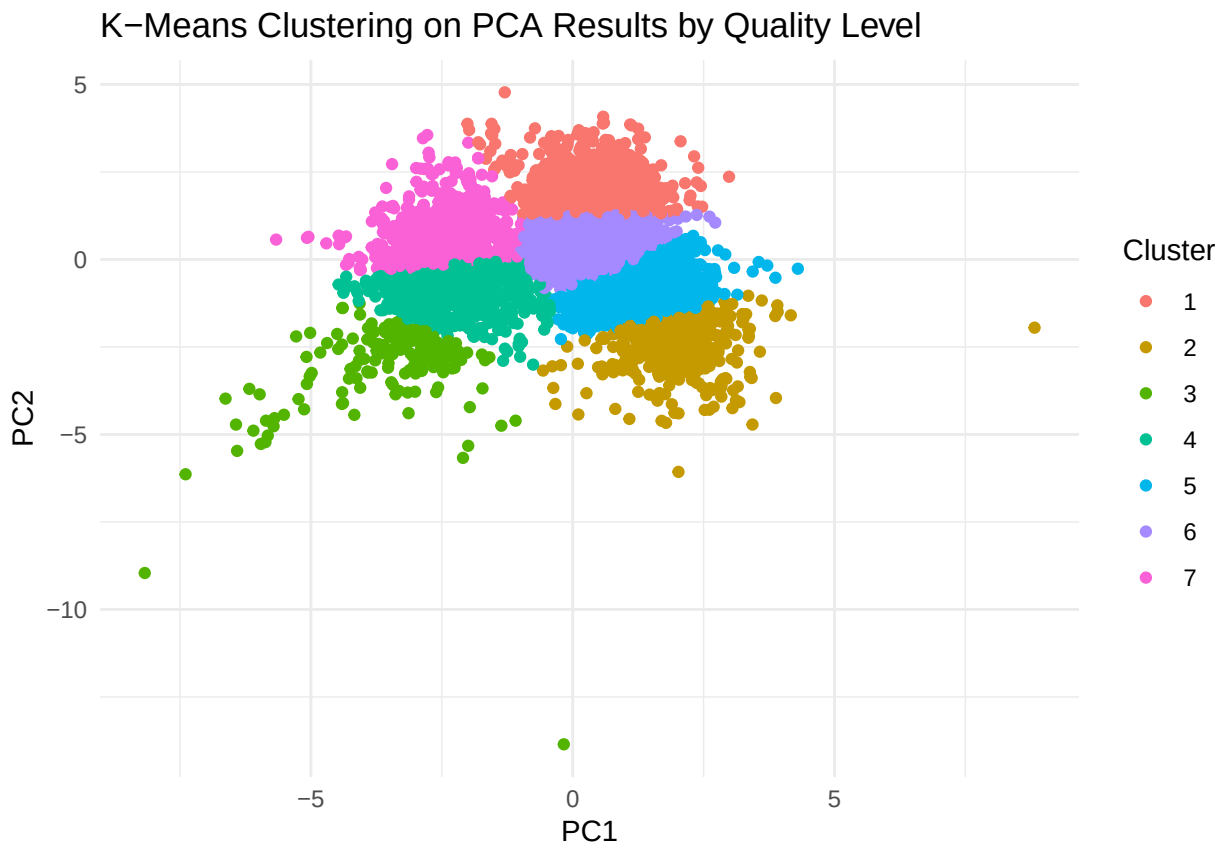
As the table above shows, each of the wine qualities was spaced out relatively equally among each cluster, which indicates that it does not do a good job of distinguishing the quality of wines based on chemical components.

K-Means Clustering

```
# Perform K-Means clustering on the PCA results
set.seed(42)
kmeans_result <- kmeans(pca$x[, 1:2], centers = length(unique(pca_result$quality)), nstart = 25)
```

```
## Warning: did not converge in 10 iterations
```

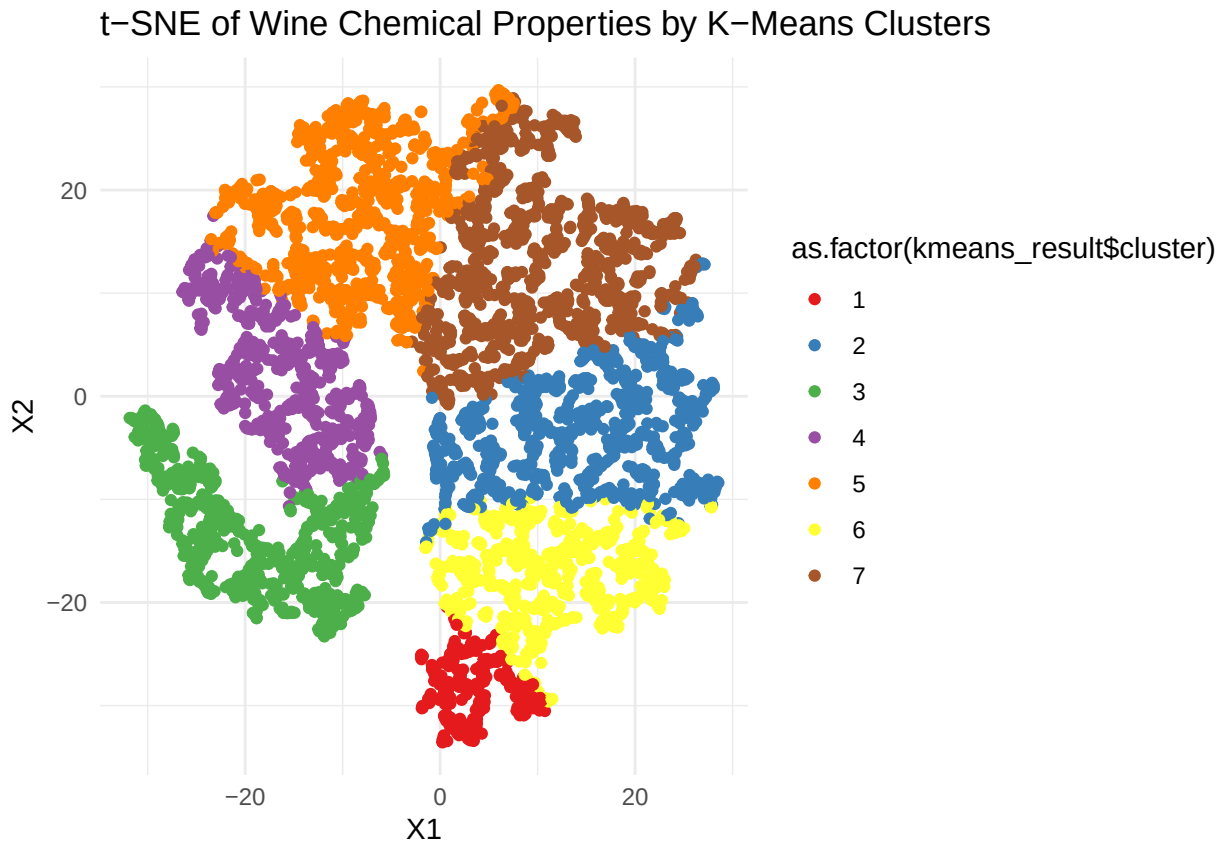
```
pca_result$Cluster <- as.factor(kmeans_result$cluster)

# Plot K-Means clustering results, colored by clusters
ggplot(pca_result, aes(x = PC1, y = PC2, color = Cluster)) +
  geom_point() +
  ggtitle("K-Means Clustering on PCA Results by Quality Level") +
  theme_minimal()
```




```
# Perform K-Means clustering on the original chemical properties
set.seed(42)
kmeans_result <- kmeans(chemical_properties_unique, centers = length(unique(wine_data$quality)), nstart
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 265900)
```

```
# Add the cluster assignments to the original data
chemical_properties_unique$Cluster <- as.factor(kmeans_result$cluster)

# Plot t-SNE results, colored by K-Means clusters
ggplot(tsne_result, aes(x = X1, y = X2, color = as.factor(kmeans_result$cluster))) +
  geom_point() +
  scale_color_brewer(palette = "Set1") +
  ggtitle("t-SNE of Wine Chemical Properties by K-Means Clusters") +
  theme_minimal()
```



Final Recommendation

Overall, it seems that none of the methods do a good job of separating the quality of the wine. This is probably because the quality of a wine is very subjective, and I am not sure how it is ranked here. K-means clustering looks like it does a good job, which is deceiving since it just spreads out into equal groups here. Which, when compared with quality, has no pattern. Overall, none of these methods seem to be a great way to categorize wine quality based on our given measurements. However, PCA and tSNE do an excellent job separating by wine color. PCA is better due to its more distinct separation, less overlap, and how well it explains the variability with the initial two principal components.