Assignment 4: PLINK, PCA, and Clustering

Habiba Tarek Ramadan Ali "ID: 21010445"

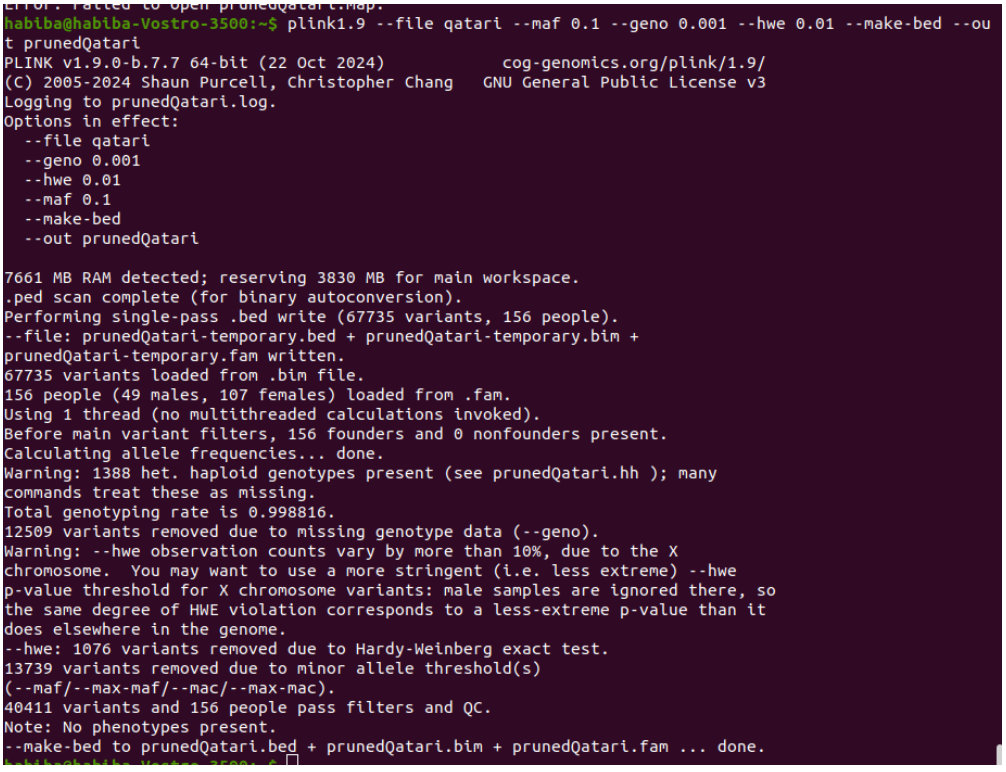
2025-03-17

Part 1: Principal Component Analysis Using PLINK

QC on Qatari dataset using PLINK

Total number of removed variants is 27324

Test	Threshold	Variants Removed
HWE	0.01	1076
MAF	0.1	13739
GENO	0.001	12509

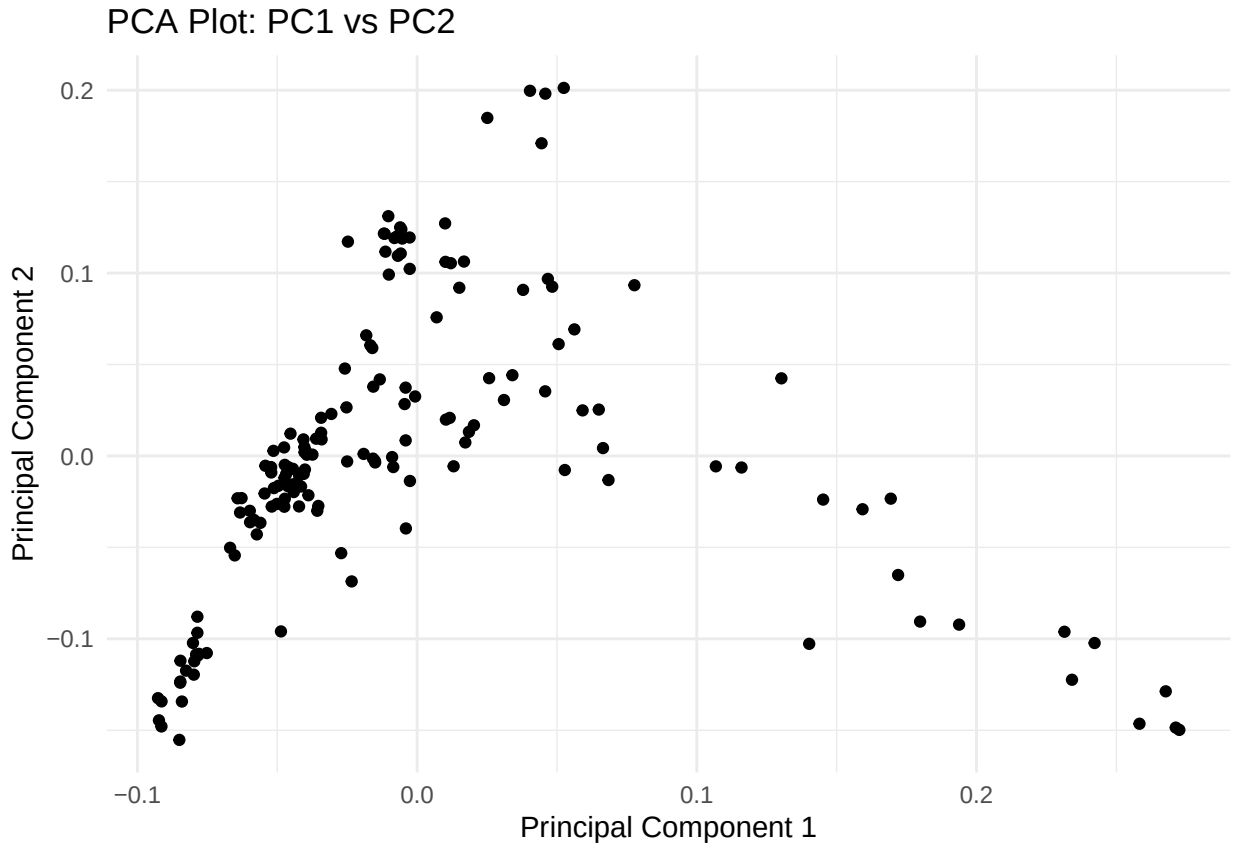


PCA

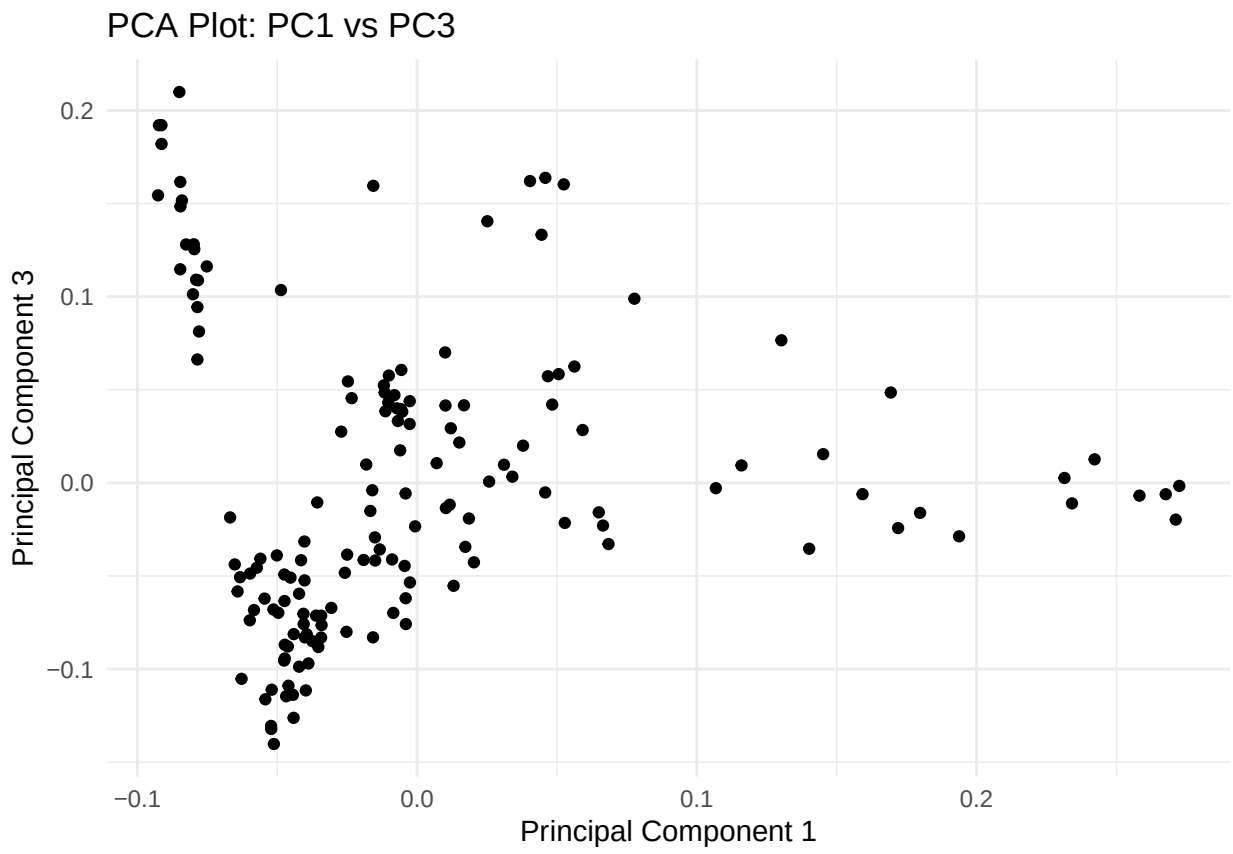
```
habiba@habiba-Vostro-3500:~$ plink1.9 --bfile prunedQatari --pca  --out qatari_pca
PLINK v1.9.0-b.7.7 64-bit (22 Oct 2024)           cog-genomics.org/plink/1.9/
(C) 2005-2024 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to qatari_pca.log.
Options in effect:
  --bfile prunedQatari
  --out qatari_pca
  --pca

7661 MB RAM detected; reserving 3830 MB for main workspace.
40411 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using up to 8 threads (change this with --threads).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1032 het. haploid genotypes present (see qatari_pca.hh ); many
commands treat these as missing.
Total genotyping rate is exactly 1.
40411 variants and 156 people pass filters and QC.
Note: No phenotypes present.
Excluding 1061 variants on non-autosomes from relationship matrix calc.
Relationship matrix calculation complete.
--pca: Results saved to qatari_pca.eigenval and qatari_pca.eigenvec .
habiba@habiba-Vostro-3500:~$
```

```r
# Load PCA Eigenvectors (Remove first two columns)
qatari_eigenVec <- read.table("/home/habiba/qatari_pca.eigenvec", header=

# Keep only columns from 3rd column
qatari_eigenVec <- qatari_eigenVec[, 3:ncol(qatari_eigenVec)]

# Rename first few PCs (adjust as needed)
colnames(qatari_eigenVec)[1:3] <- c("PC1", "PC2", "PC3")

# Load PCA Eigenvalues
qatari_eigenVal <- read.table("/home/habiba/qatari_pca.eigenval", header=

colnames(qatari_eigenVal) <- c("Eigenvalue")

# Print results
# print(qatari_eigenVec)
# print(qatari_eigenVal)
```

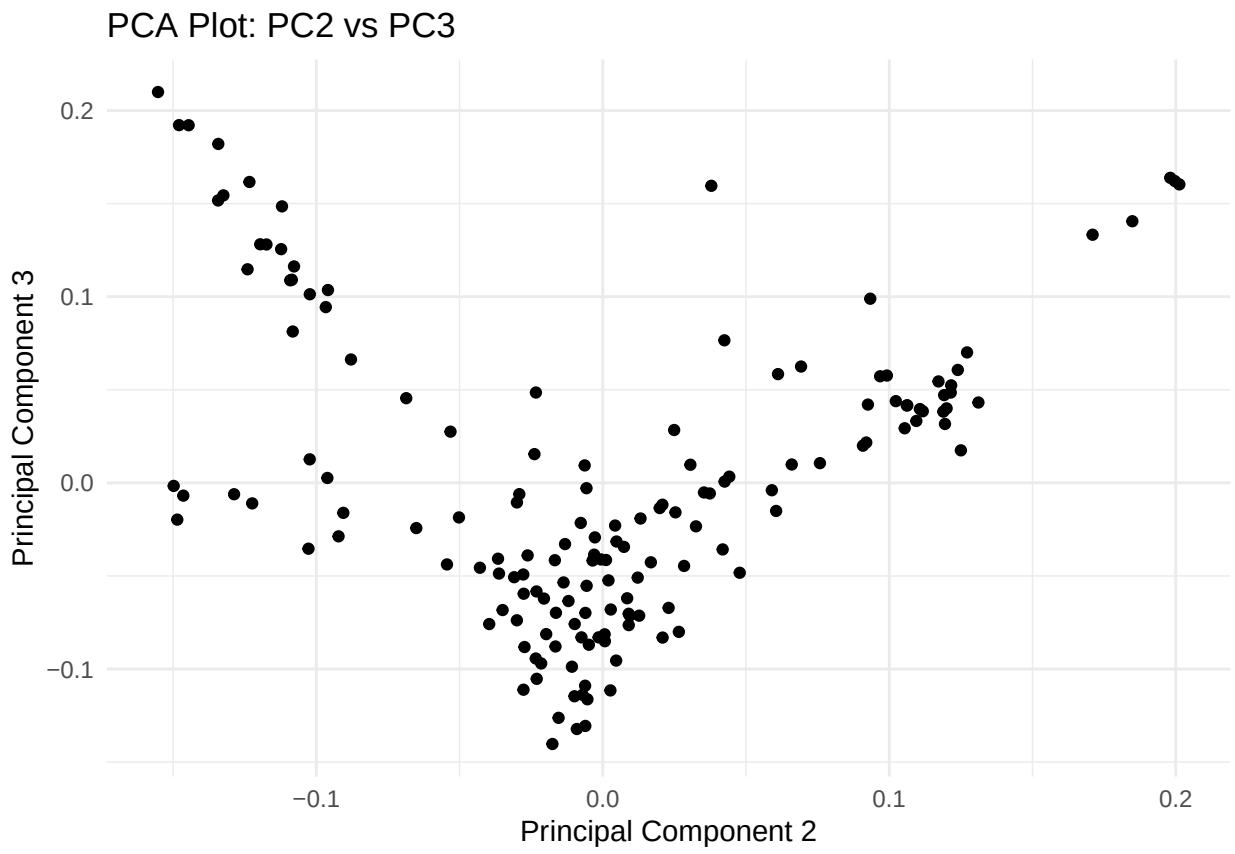
```r
# Load necessary libraries
library(ggplot2)

# Create scatter plots for PC1 vs PC2, PC1 vs PC3, and PC2 vs PC3
ggplot(qatari_eigenVec, aes(x=PC1, y=PC2)) +
  geom_point(aes()) +  # Color points based on PC3
  theme_minimal() +
  labs(title="PCA Plot: PC1 vs PC2", x="Principal Component 1", y="Princi
```



```
ggplot(qatari_eigenVec, aes(x=PC1, y=PC3)) +
  geom_point(aes()) +
  theme_minimal() +
  labs(title="PCA Plot: PC1 vs PC3", x="Principal Component 1", y="Princi
```

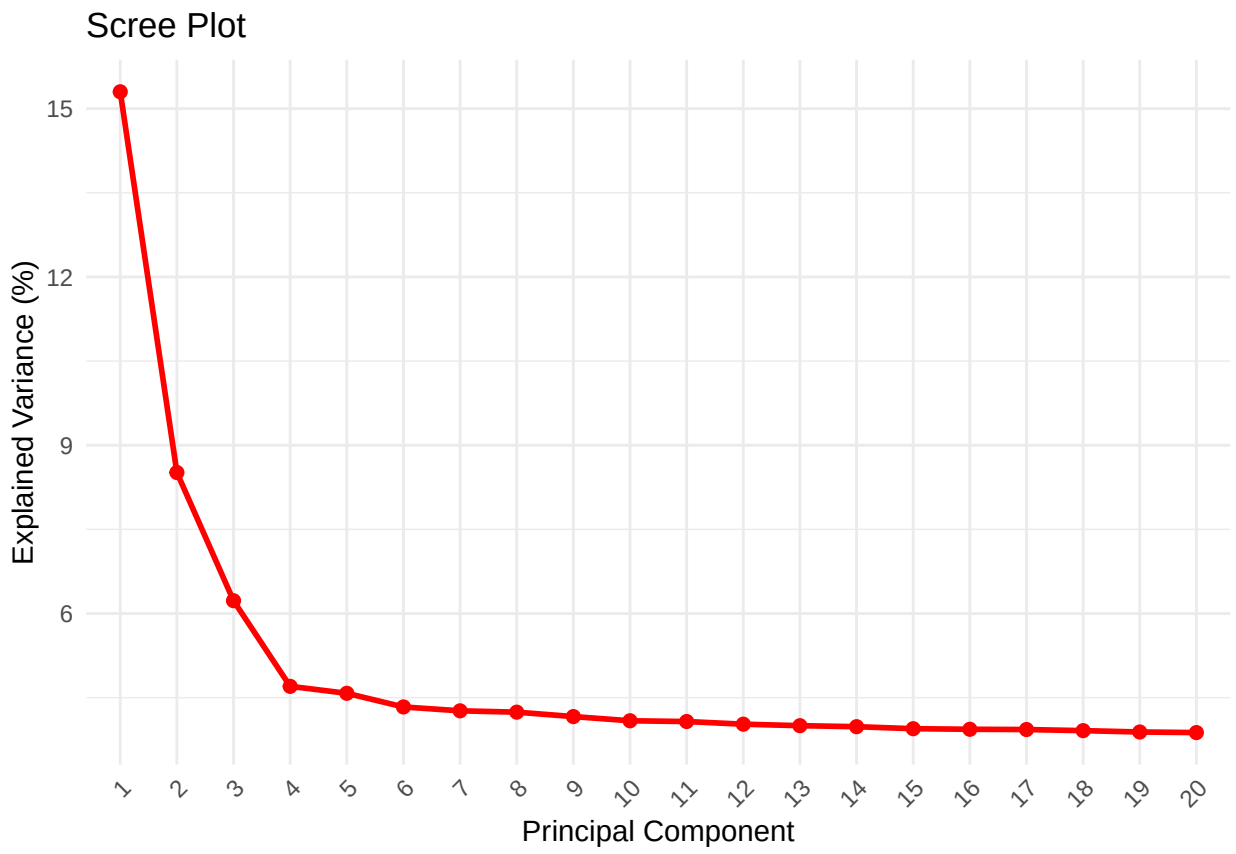


```r
ggplot(qatari_eigenVec, aes(x=PC2, y=PC3)) +
  geom_point(aes()) +
  theme_minimal() +
  labs(title="PCA Plot: PC2 vs PC3", x="Principal Component 2", y="Princi
```



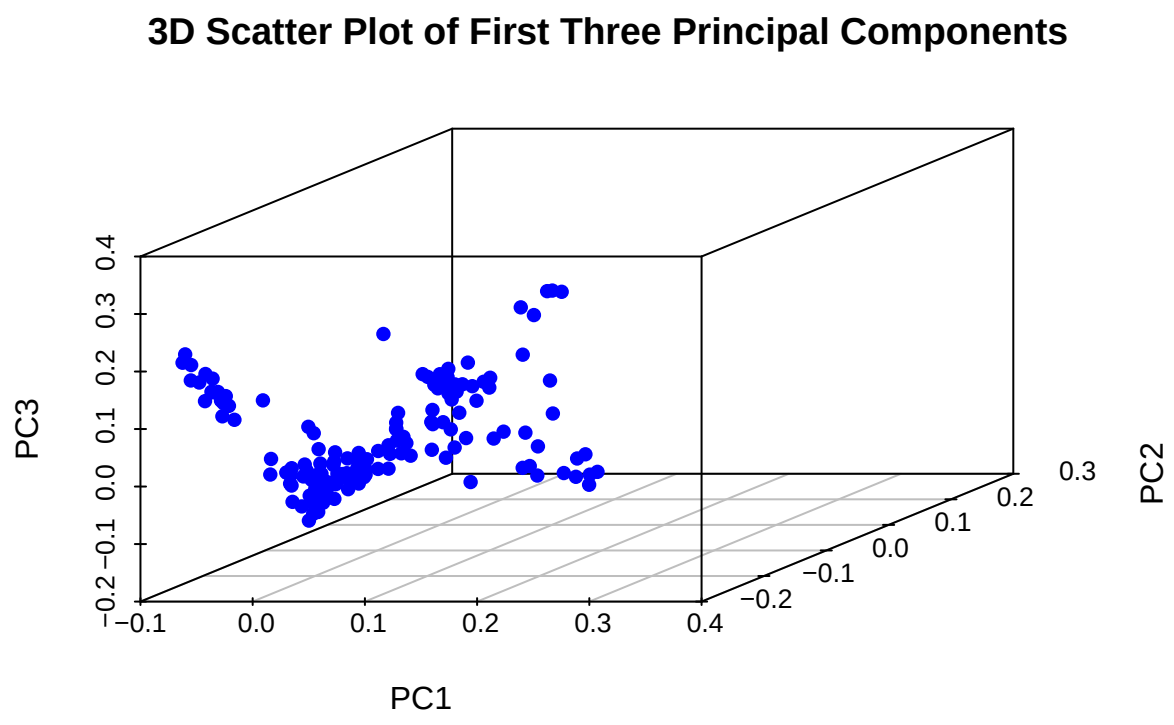
```r
library(ggplot2)

# Compute explained variance
total_variance <- sum(qatari_eigenVal$Eigenvalue)
qatari_eigenVal$ExplainedVariance <- (qatari_eigenVal$Eigenvalue / total_

# Add PC numbers for plotting
qatari_eigenVal$PC <- seq_len(nrow(qatari_eigenVal))  # Ensure sequence s

# Scree plot for the first 20 principal components
ggplot(qatari_eigenVal[1:20, ], aes(x = factor(PC), y = ExplainedVariance
  geom_line(aes(group = 1), color = "red", size = 1) +  # Add a line for
  geom_point(color = "red", size = 2) +  # Highlight points
  labs(title = "Scree Plot", x = "Principal Component", y = "Explained Va
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-ax
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning w
## generated.
```


Scree Plot
Explained Variance (%)
15
12
9
6
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Principal Component

```r
#install.packages("scatterplot3d", INSTALL_opts = c('--no-lock'))
#install.packages("cluster")
#install.packages("ggplot2")

library(cluster)    # For Dunn's index
library(ggplot2)          # 2D visualization
library(scatterplot3d)    # 3D visualization
```

```r
# 3D scatter plot
scatterplot3d(qatari_eigenVec[, 1], qatari_eigenVec[, 2], qatari_eigenVec
              xlab = "PC1",
              ylab = "PC2",
              zlab = "PC3",
              main = "3D Scatter Plot of First Three Principal Components
              pch = 16,  # Solid circles
              color = "blue")
```



Part 2: Clustering in R

Task 2.1: Perform Clustering

```r
# Load necessary libraries
library(fpc)              # Cluster evaluation metrics
library(gridExtra)        # Arrange multiple plots
library(grid)             # Work with graphical objects

reducedVectors <- qatari_eigenVec[, 1:3]
```

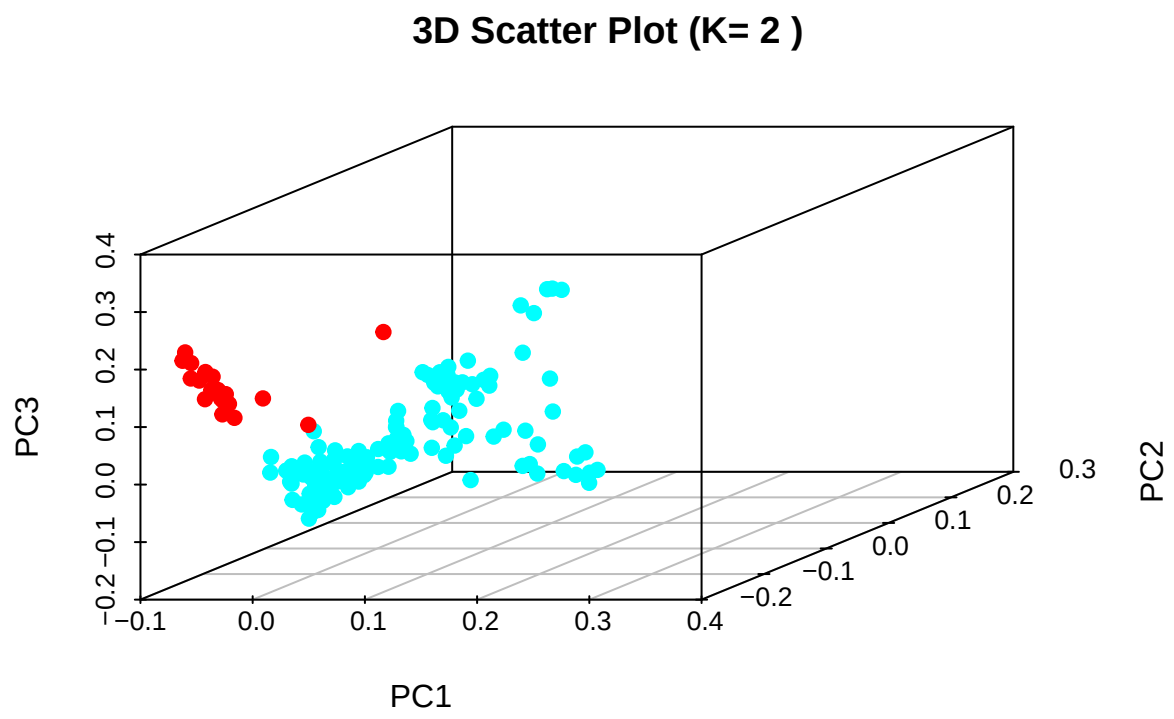
```r
# Function to compute Dunn's Index
# The ratio between the min inter-cluster distance to the max intra clus
# The higher the Dunn index the better the clustering
compute_dunn_index <- function(data, clusters) {
  dist_matrix <- dist(data)
  dunn_index <- cluster.stats(dist_matrix, clusters)$dunn
  return(dunn_index)
  #return(dunn(dist(data), kmeans_result$clusters)) msh sh3'ala
}

plot_2d_clusters <- function(data, clusters, k) {
  data$Cluster <- as.factor(clusters)  # Convert cluster labels to facto

  plot_pc1_pc2 <- ggplot(data, aes(x = PC1, y = PC2, color = Cluster)) +
    geom_point(size = 2) +
    labs(title = paste("K =", k, "- PC1 vs PC2"), x = "PC1", y = "PC2") +
    theme_minimal()

  plot_pc1_pc3 <- ggplot(data, aes(x = PC1, y = PC3, color = Cluster)) +
    geom_point(size = 2) +
    labs(title = paste("K =", k, "- PC1 vs PC3"), x = "PC1", y = "PC3") +
    theme_minimal()

  plot_pc2_pc3 <- ggplot(data, aes(x = PC2, y = PC3, color = Cluster)) +
    geom_point(size = 2) +
    labs(title = paste("K =", k, "- PC2 vs PC3"), x = "PC2", y = "PC3") +
```

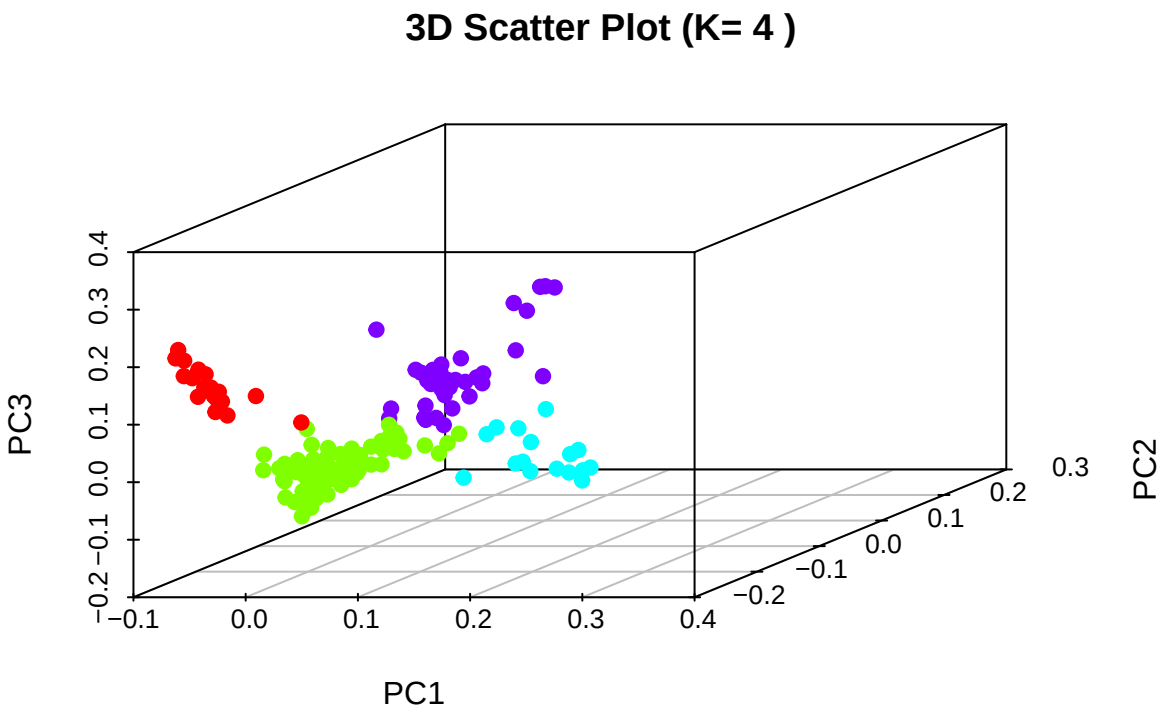
```
    theme_minimal()

  # Arrange the three plots side by side
    return(grid.arrange(plot_pc1_pc2, plot_pc1_pc3, plot_pc2_pc3, ncol =
  }
```

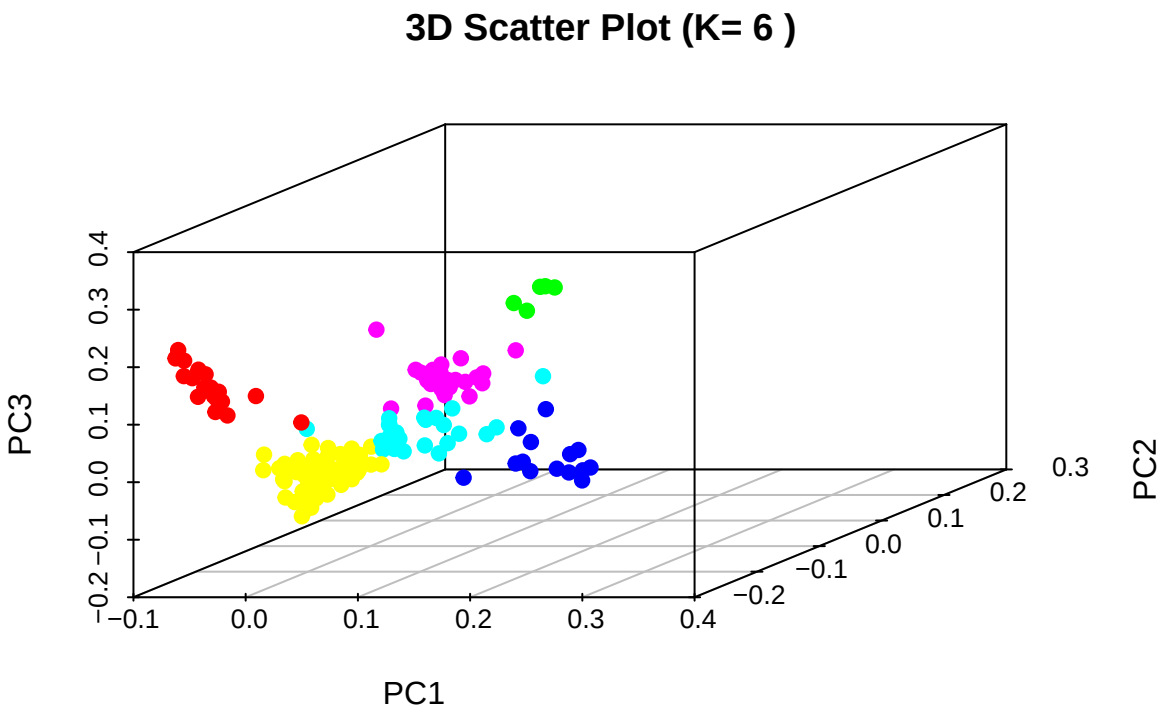
```
# Define cluster sizes
cluster_sizes <- c(2, 4, 6, 8, 10)

# Initialize storage for results
kmeans_results <- list()
dunn_values <- data.frame(Clusters = integer(), Dunn = numeric())
plots_3d <- list()
plots_2d <- list()

# Run clustering and generate plots
for (k in cluster_sizes) {
  set.seed(123) #ensures reproducibility

  # Perform K-means clustering
  kmeans_result <- kmeans(reducedVectors, centers = k, nstart = 25) #run
  kmeans_results[[as.character(k)]] <- kmeans_result #store the results t
  clusters <- kmeans_result$cluster # assignment of each point

  # Compute Dunn's Index
  dunn_index <- compute_dunn_index(reducedVectors, clusters)
  dunn_values[k/2, ] <- c(k, dunn_index)
}
```

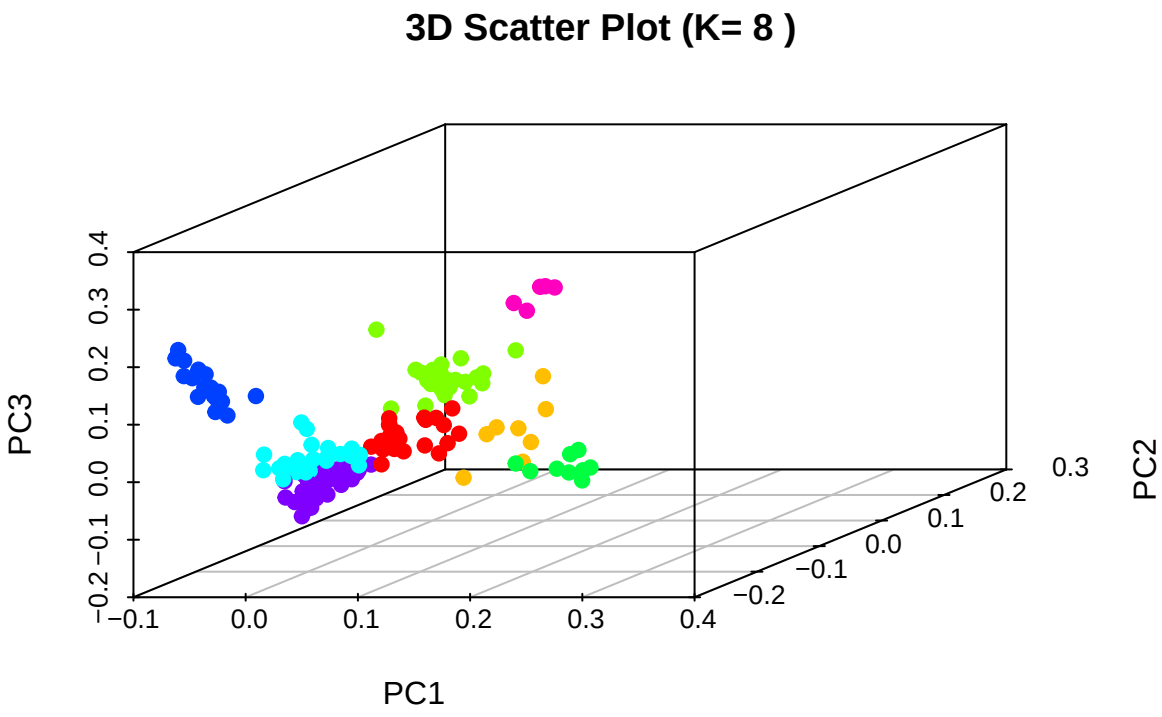
```
# Function to generate a 3D scatter plot
for (k in cluster_sizes) {
  clusters <- kmeans_results[[as.character(k)]]$cluster
  cluster_colors <- rainbow(length(unique(clusters)))[clusters]

  scatterplot3d(reducedVectors$PC1, reducedVectors$PC2, reducedVectors$PC
                color = cluster_colors, pch = 19,
```

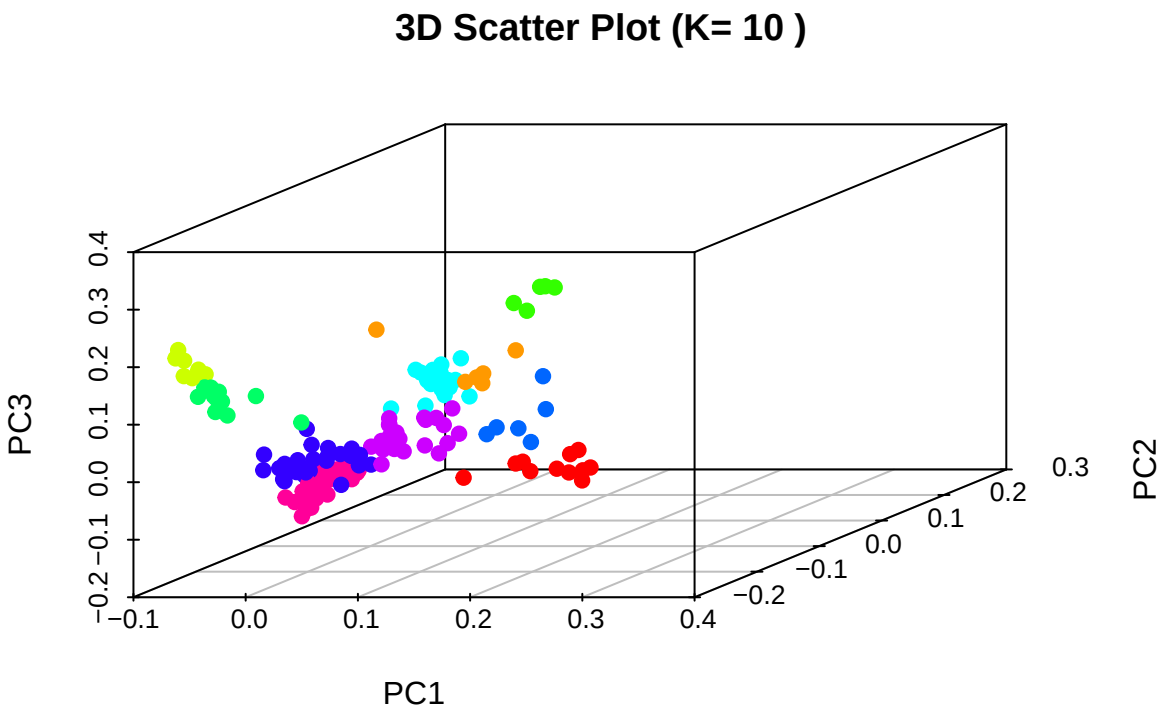
```
                main = paste("3D Scatter Plot (K=", k, ")"),
                xlab = "PC1", ylab = "PC2", zlab = "PC3")

}
```



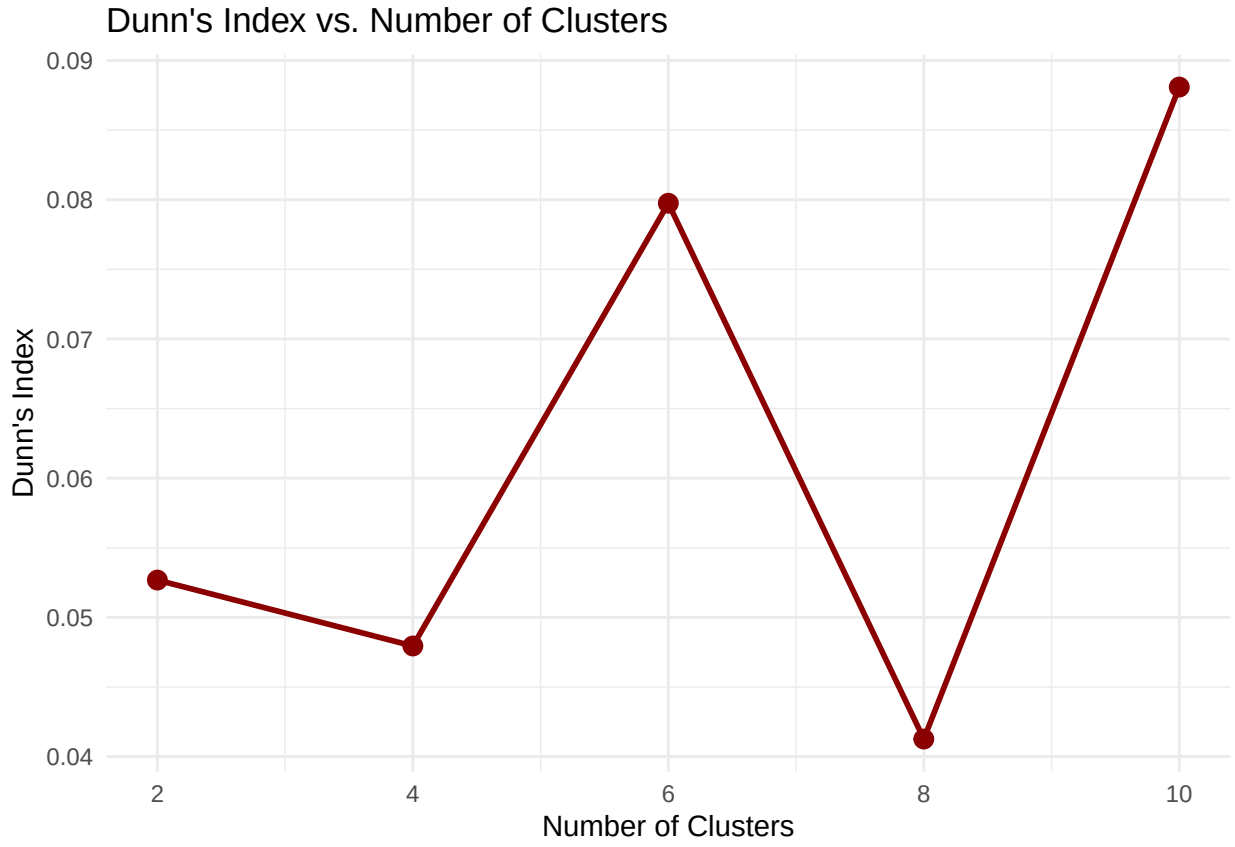

3D Scatter Plot (K= 4)
PC3
PC1
PC2
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4


3D Scatter Plot (K= 6)
PC3
PC1
PC2
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4


3D Scatter Plot (K= 8)
PC3
PC1
PC2
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4

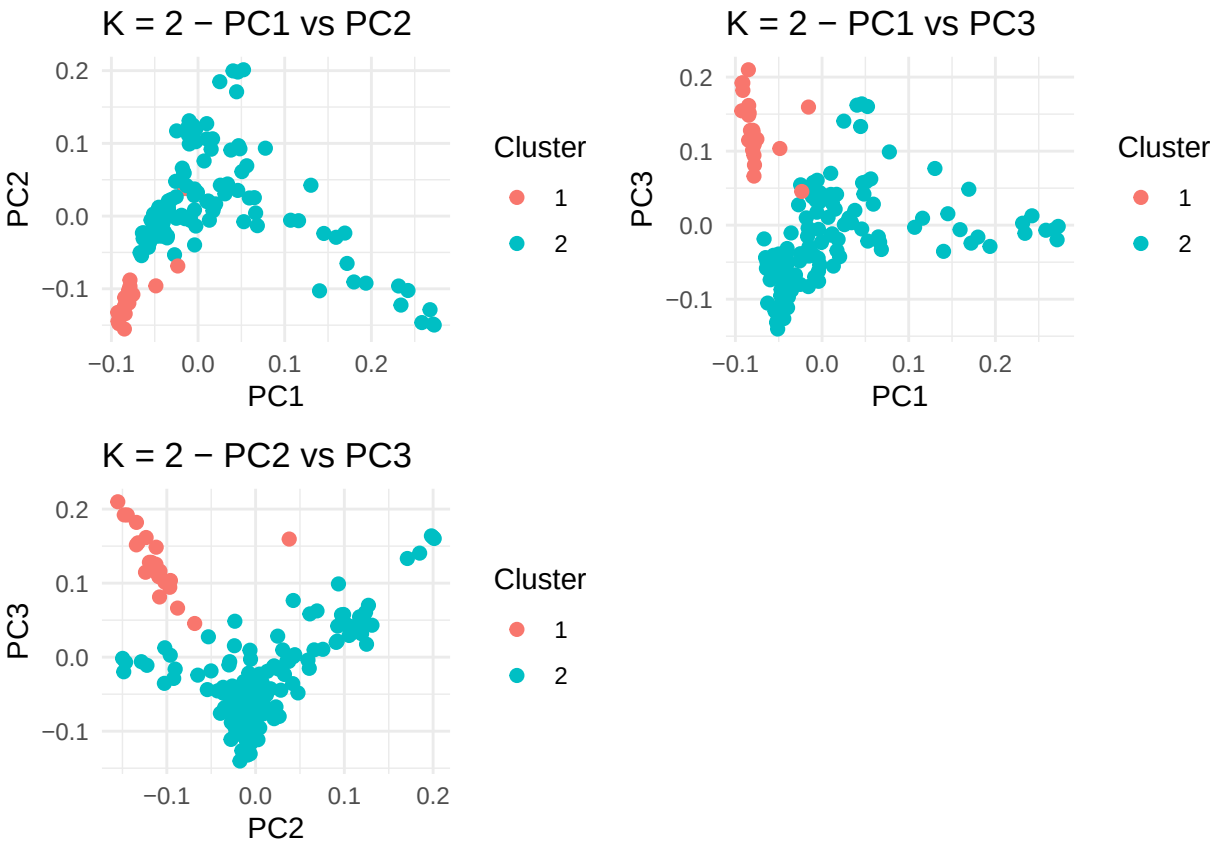

3D Scatter Plot (K= 10)
PC3
PC1
PC2
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4

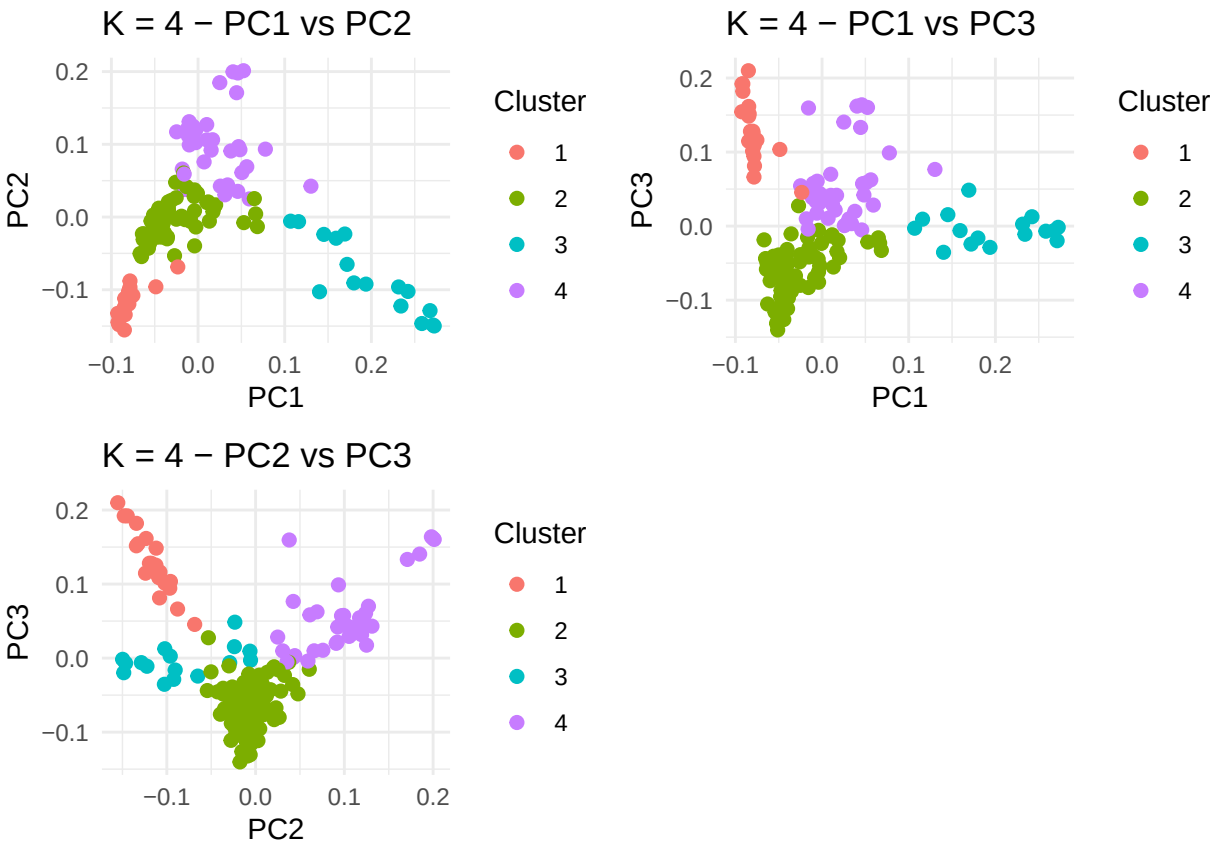
```r
ggplot(dunn_values, aes(x = Clusters, y = Dunn)) +
  geom_line(color = "darkred", size = 1) +
  geom_point(size = 3, color = "darkred") +
  labs(title = "Dunn's Index vs. Number of Clusters",
       x = "Number of Clusters",
       y = "Dunn's Index") +
  theme_minimal()
```

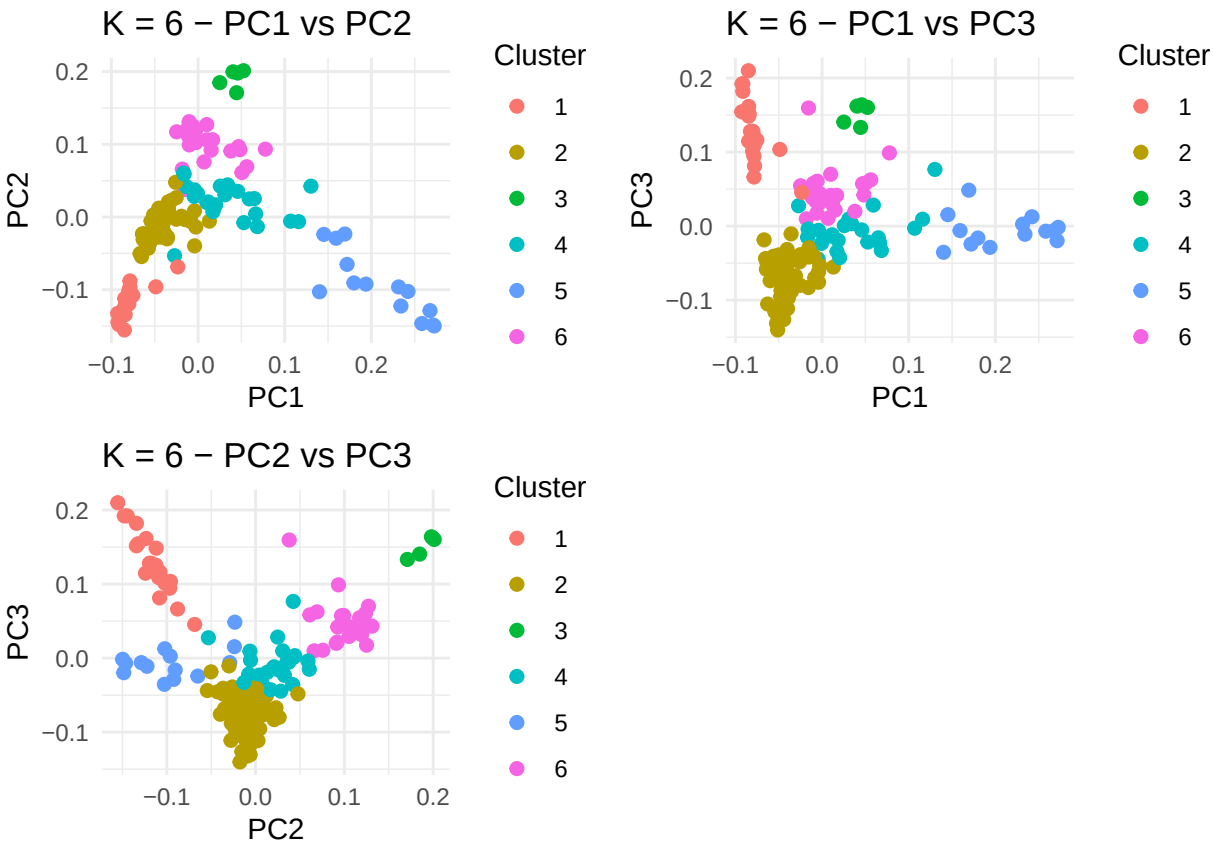


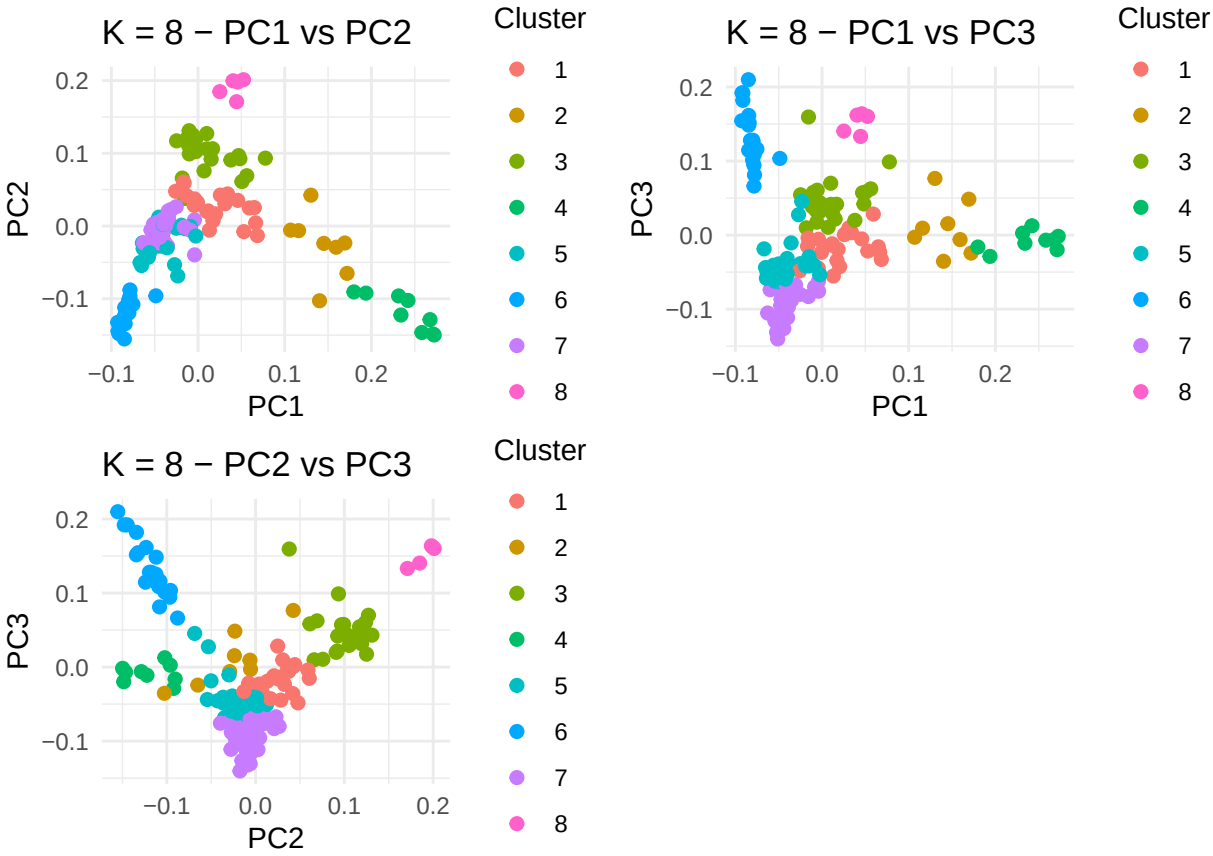
Task 2.2: Visualize Clusters

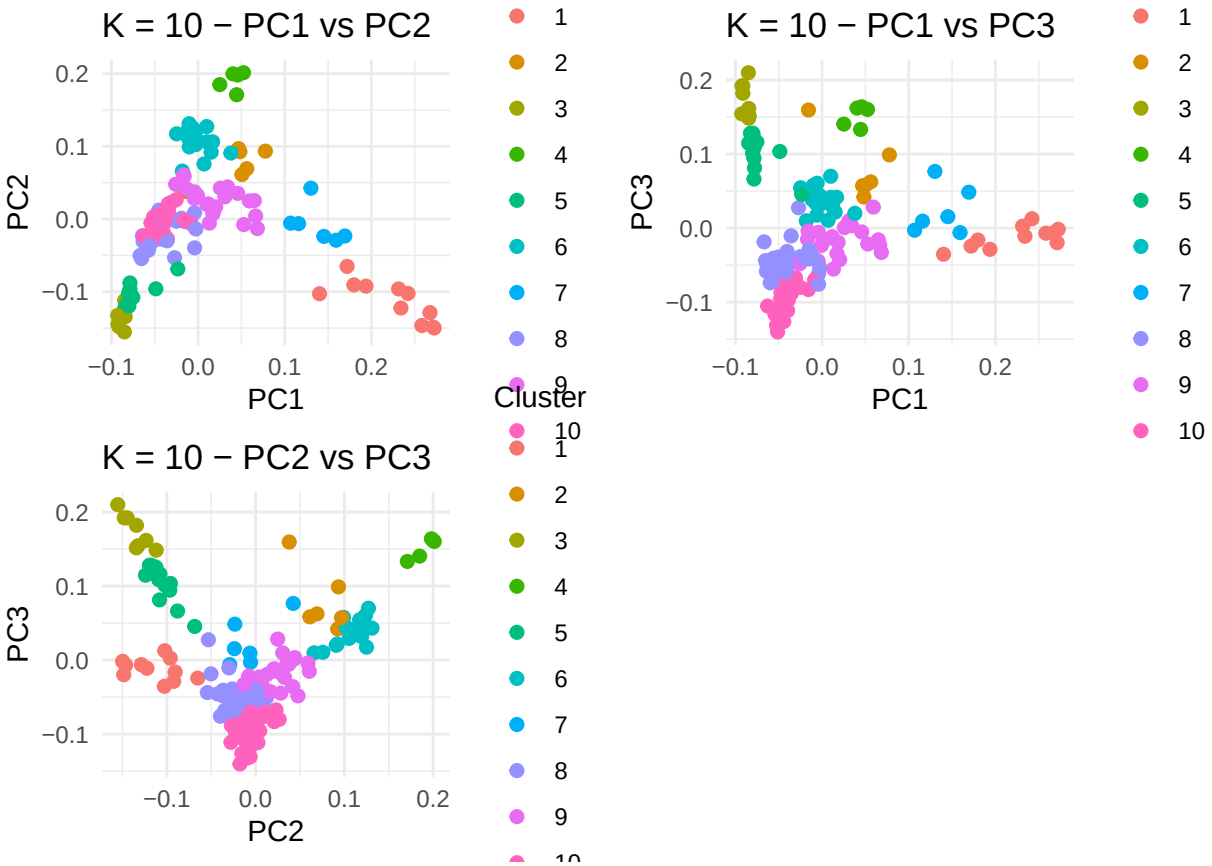
```
# Display 2D plots for each cluster size
for (k in cluster_sizes) {
  plots_2d[[as.character(k)]] <- plot_2d_clusters(qatari_eigenVec,kmeans_
}
```




K = 4 − PC1 vs PC2
PC2
PC1
0.2
0.1
0.0
−0.1
−0.1
0.0
0.1
0.2
Cluster
1
2
3
4
K = 4 − PC1 vs PC3
PC3
PC1
Cluster
1
2
3
4
K = 4 − PC2 vs PC3
PC3
PC2
Cluster
1
2
3
4


K = 6 − PC1 vs PC2
PC2
PC1
0.2
0.1
0.0
−0.1
−0.1
0.0
0.1
0.2
Cluster
1
2
3
4
5
6
K = 6 − PC1 vs PC3
PC3
PC1
Cluster
1
2
3
4
5
6
K = 6 − PC2 vs PC3
PC3
PC2
Cluster
1
2
3
4
5
6


K = 8 − PC1 vs PC2
PC2
PC1
0.2
0.1
0.0
−0.1
−0.1
0.0
0.1
0.2
Cluster
1
2
3
4
5
6
7
8
K = 8 − PC1 vs PC3
PC3
PC1
Cluster
K = 8 − PC2 vs PC3
PC3
PC2
Cluster


K = 10 − PC1 vs PC2
PC2
PC1
K = 10 − PC1 vs PC3
PC3
PC1
K = 10 − PC2 vs PC3
PC3
PC2
Cluster
1
2
3
4
5
6
7
8
9
10
0.2
0.1
0.0
−0.1