Lab 4

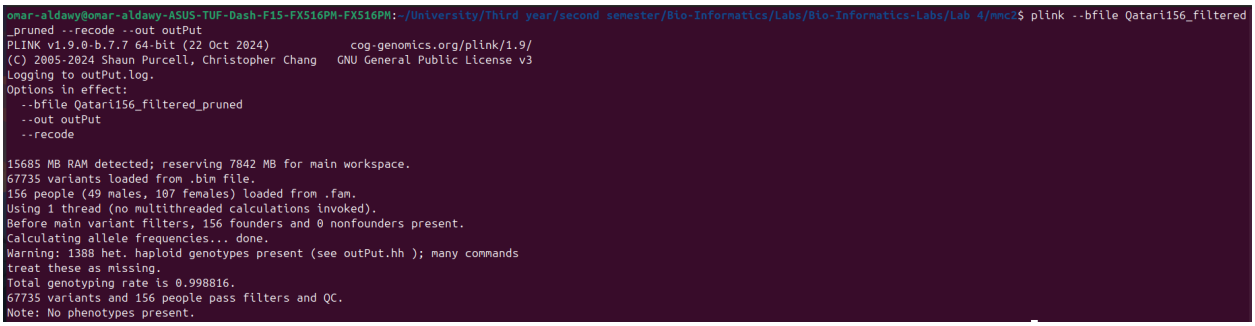
Omar Aldawy Ibrahim Aldawy 21010864

2025-03-18

Part 1: Principal Component Analysis Using PLINK

Converting files to Ped and Map format

```
plink --bfile your_input_filename --recode --out your_output_filename
```



```
omar-aldawy@omar-aldawy-ASUS-TUF-Dash-F15-FX516PM-FX516PM:~/University/Third year/second semester/Bio-Informatics/Labs/Bio-Informatics-Labs/Lab 4/mmc2$ plink --bfile Qatari156_filtered
_pruned --recode --out outPut
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 outPut.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --out outPut
  --recode

15685 MB RAM detected; reserving 7842 MB for main workspace.
67735 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1388 het. haploid genotypes present (see outPut.hh ); many commands
treat these as missing.
Total genotyping rate is 0.998816.
67735 variants and 156 people pass filters and QC.
Note: No phenotypes present.
```

Figure 1: creating ped and map files

Running Quality Control with thresholds {hwe: 0.01, maf:0.1, geno: 0.001}

```
plink --file your_dataset --hwe 0.01 --maf 0.1 --geno 0.001 --recode --out qc_filtered
```

- 12509 variants removed due to missing genotype data (–geno).
- 1076 variants removed due to Hardy-Weinberg exact test (–hwe).
- 13739 variants removed due to minor allele threshold(s) (–maf).
- Total variants removed = 27324.

Running PCA analysis

```
plink --file qc_filtered --pca --out pca_results
```

Loading the PCA results into R.

```r
# Load PCA results into R
pca_data <- read.table("mmc2/pca_results.eigenvec", header = FALSE)

# Rename columns
colnames(pca_data) <- c("FID", "IID", paste0("PC", 1:20))  # Adjust to include the first 20 PCs
```

```
omar-aldawy@omar-aldawy-ASUS-TUF-Dash-F15-FX516PM-FX516PM:~/University/Third year/second semester/Bio-Informatics/Labs/Bio-Informatics-Labs/Lab 4/mmc2$ plink --file outPut --hwe 0.01 -
-maf 0.1 --geno 0.001 --recode --out qc_filtered
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 qc_filtered.log.
Options in effect:
  --file outPut
  --geno 0.001
  --hwe 0.01
  --maf 0.1
  --out qc_filtered
  --recode

15685 MB RAM detected; reserving 7842 MB for main workspace.
.ped scan complete (for binary autoconversion).
Performing single-pass .bed write (67735 variants, 156 people).
--file: qc_filtered-temporary.bed + qc_filtered-temporary.bim +
qc_filtered-temporary.fam written.
67735 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1388 het. haploid genotypes present (see qc_filtered.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
12509 variants removed due to missing genotype data (--geno).
Warning: --hwe observation counts vary by more than 10%, due to the X
chromosome.  You may want to use a more stringent (i.e. less extreme) --hwe
p-value threshold for X chromosome variants: male samples are ignored there, so
the same degree of HWE violation corresponds to a less-extreme p-value than it
does elsewhere in the genome.
--hwe: 1076 variants removed due to Hardy-Weinberg exact test.
13739 variants removed due to minor allele threshold(s)
(--maf/--max-maf/--mac/--max-mac).
40411 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode ped to qc_filtered.ped + qc_filtered.map ... done.
```

Figure 2: filter data

```
omar-aldawy@omar-aldawy-ASUS-TUF-Dash-F15-FX516PM-FX516PM:~/University/Third year/second semester/Bio-Informatics/Labs/Bio-Informatics-Labs/Lab 4/mmc2$ plink --file qc_filtered --pca -
-out pca_results
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 pca_results.log.
Options in effect:
  --file qc_filtered
  --out pca_results
  --pca

15685 MB RAM detected; reserving 7842 MB for main workspace.
.ped scan complete (for binary autoconversion).
Performing single-pass .bed write (40411 variants, 156 people).
--file: pca_results-temporary.bed + pca_results-temporary.bim +
pca_results-temporary.fam written.
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 pca_results.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.
```

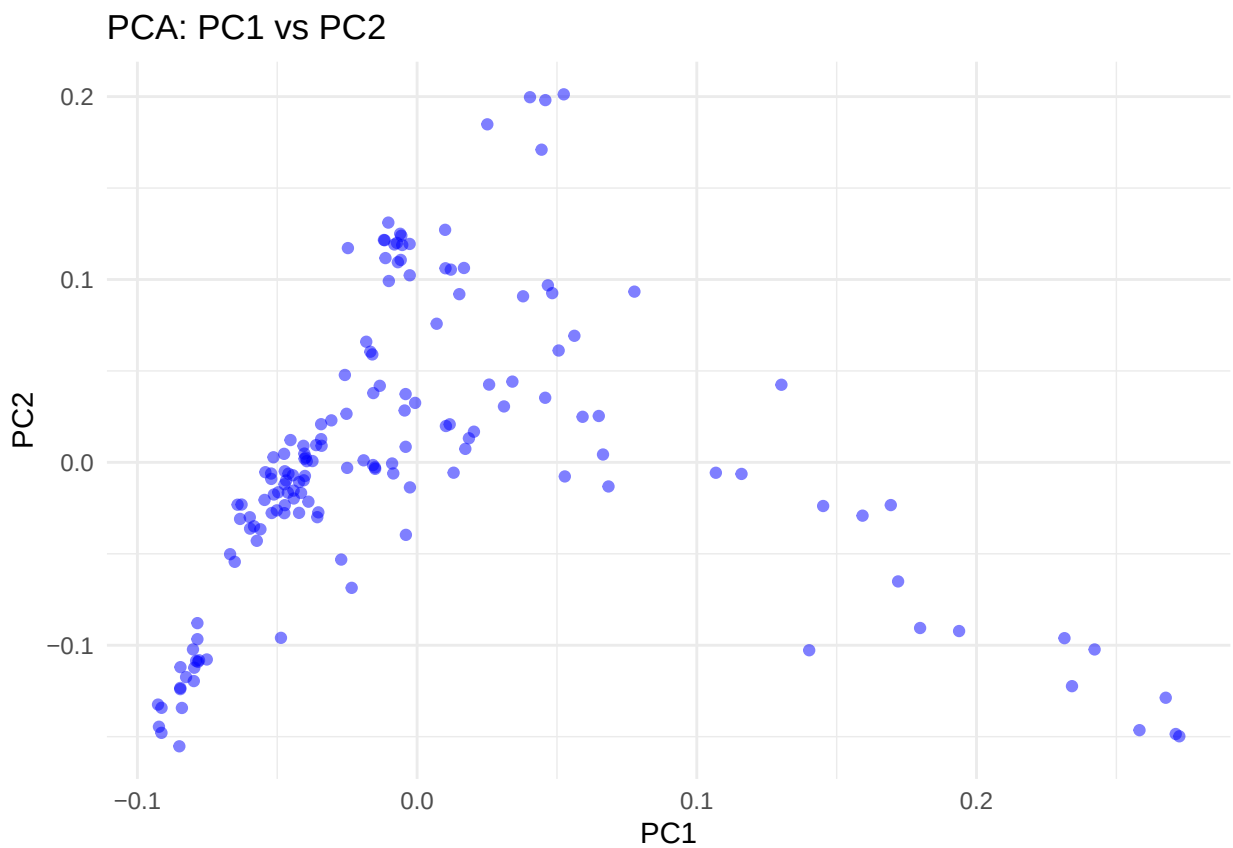
Figure 3: Principle Component Analysis

```r
# View the first few rows
head(pca_data)
```

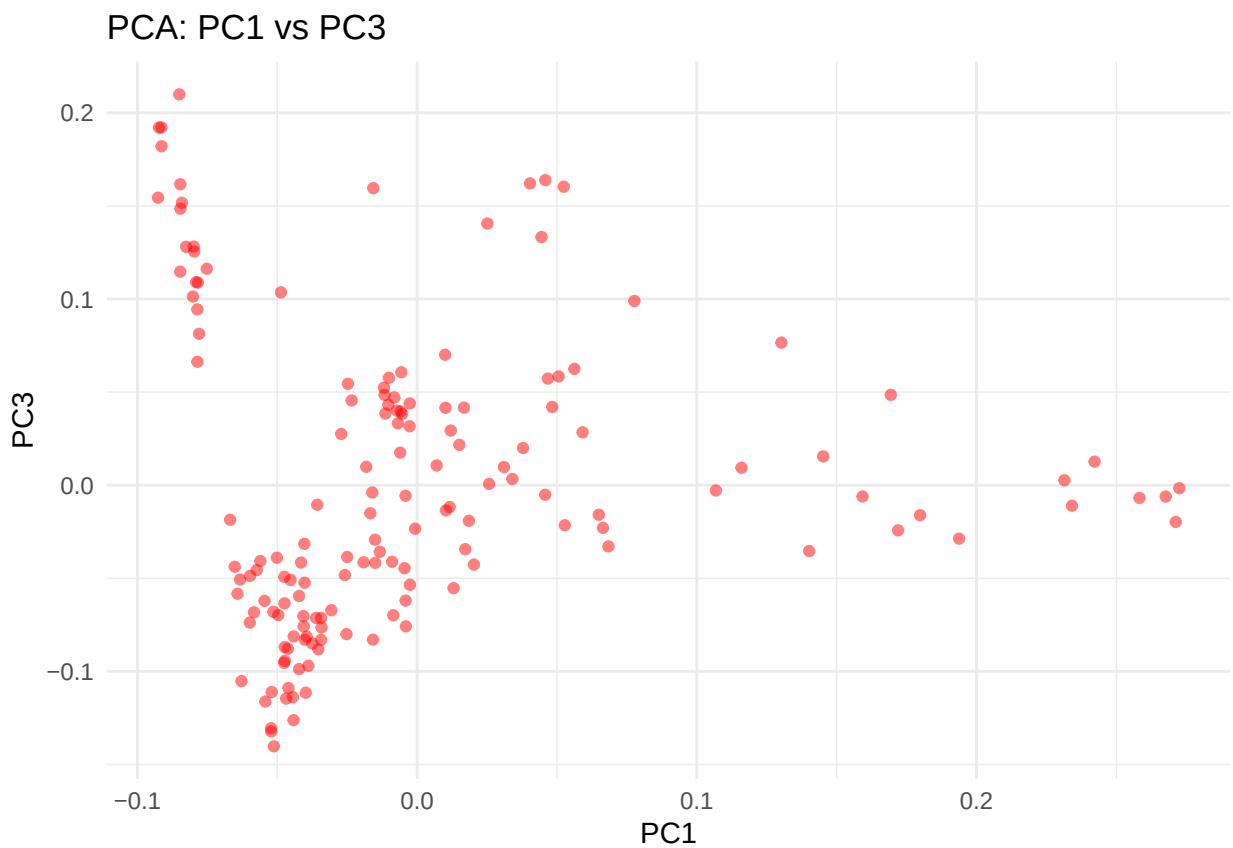
```
##        FID      IID        PC1          PC2          PC3         PC4        PC5
## 1  QBC-092  QBC-092  0.0257471  0.042520200  0.000671924 -0.00464458  0.0567232
## 2  QBC-256  QBC-256 -0.0394316  0.000642535 -0.081346600  0.00336033 -0.0417539
## 3  QBC-107  QBC-107 -0.0401049 -0.007438850 -0.082995400 -0.00482384 -0.0441360
## 4  QBC-171  QBC-171 -0.0156592  0.037883500  0.159538000 -0.00573798 -0.1284940
## 5 QPRC-110 QPRC-110 -0.0118682  0.121597000  0.052364500 -0.03498780  0.0928528
## 6  QBC-240  QBC-240  0.0562310  0.069204100  0.062476200 -0.03996550 -0.1202210
##            PC6         PC7        PC8        PC9       PC10       PC11
## 1 -0.048486600 -0.01393230  0.0781528  0.0268408 -0.0576382 -0.0534880
## 2 -0.005265190 -0.00426791  0.0015872 -0.0132695 -0.0113520 -0.0629116
## 3  0.036809600  0.03496910  0.0378915 -0.0742618  0.0906036 -0.0527715
## 4  0.012056700 -0.07296310 -0.0100284 -0.0304847  0.0949105 -0.0160013
## 5 -0.000847322  0.00554121  0.0413364 -0.0462024  0.0545213  0.0187307
## 6 -0.051613800 -0.06141850 -0.0734059  0.0955638 -0.0516559  0.0368023
##         PC12        PC13       PC14        PC15       PC16       PC17
## 1  0.0823719 -0.07114360 0.05323750  0.00297373 0.00254944 -0.0176990
## 2  0.0394701 -0.05954530 0.07098600  0.00811332 0.05649910  0.0291155
## 3 -0.0116274 -0.03030940 0.08332640 -0.01676970 0.00751046  0.0244089
## 4 -0.0589077 -0.05330160 0.00979758 -0.02945260 0.00606746  0.0600727
## 5 -0.0465393 -0.00735418 0.03413410 -0.05906570 0.00389287  0.0233314
## 6  0.0382373 -0.03715100 0.05112190  0.04009920 0.09371810  0.0175659
##         PC18       PC19        PC20
## 1  0.0170634 -0.0863075  0.09055250
## 2 -0.0189206 -0.0123068 -0.00542049
## 3  0.0717282 -0.0222190 -0.05674480
## 4  0.0482577 -0.0272777 -0.02243840
## 5  0.1386180 -0.0242056  0.00986568
## 6  0.0351497  0.0467357 -0.14841100
```

Create 2D scatter plots comparing PC1 vs PC2, PC1 vs PC3, and PC2 vs PC3.

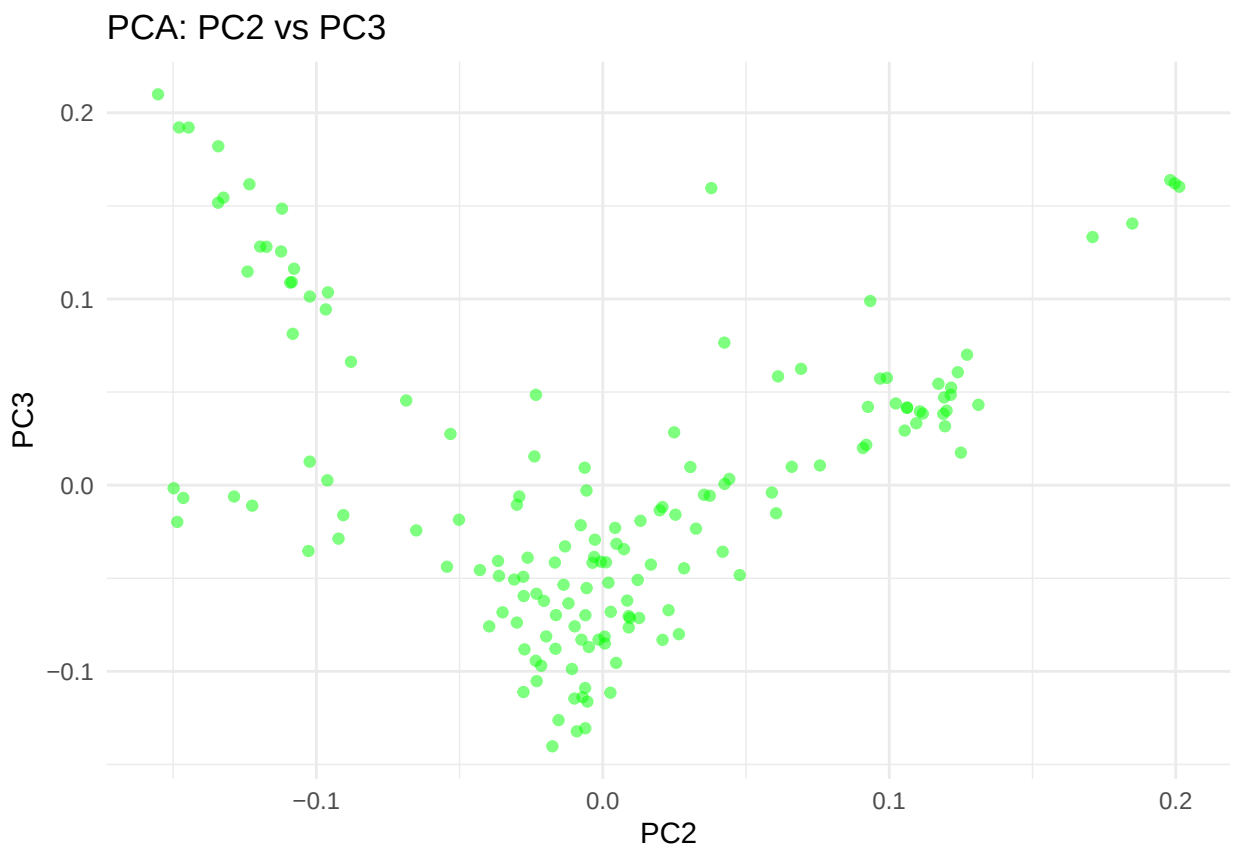
```r
# Load required package
library(ggplot2)

# PC1 vs PC2
ggplot(pca_data, aes(x = PC1, y = PC2)) +
  geom_point(alpha = 0.5, color = "blue") +
  labs(title = "PCA: PC1 vs PC2", x = "PC1", y = "PC2") +
  theme_minimal()
```




```r
# PC1 vs PC3
ggplot(pca_data, aes(x = PC1, y = PC3)) +
  geom_point(alpha = 0.5, color = "red") +
  labs(title = "PCA: PC1 vs PC3", x = "PC1", y = "PC3") +
  theme_minimal()
```

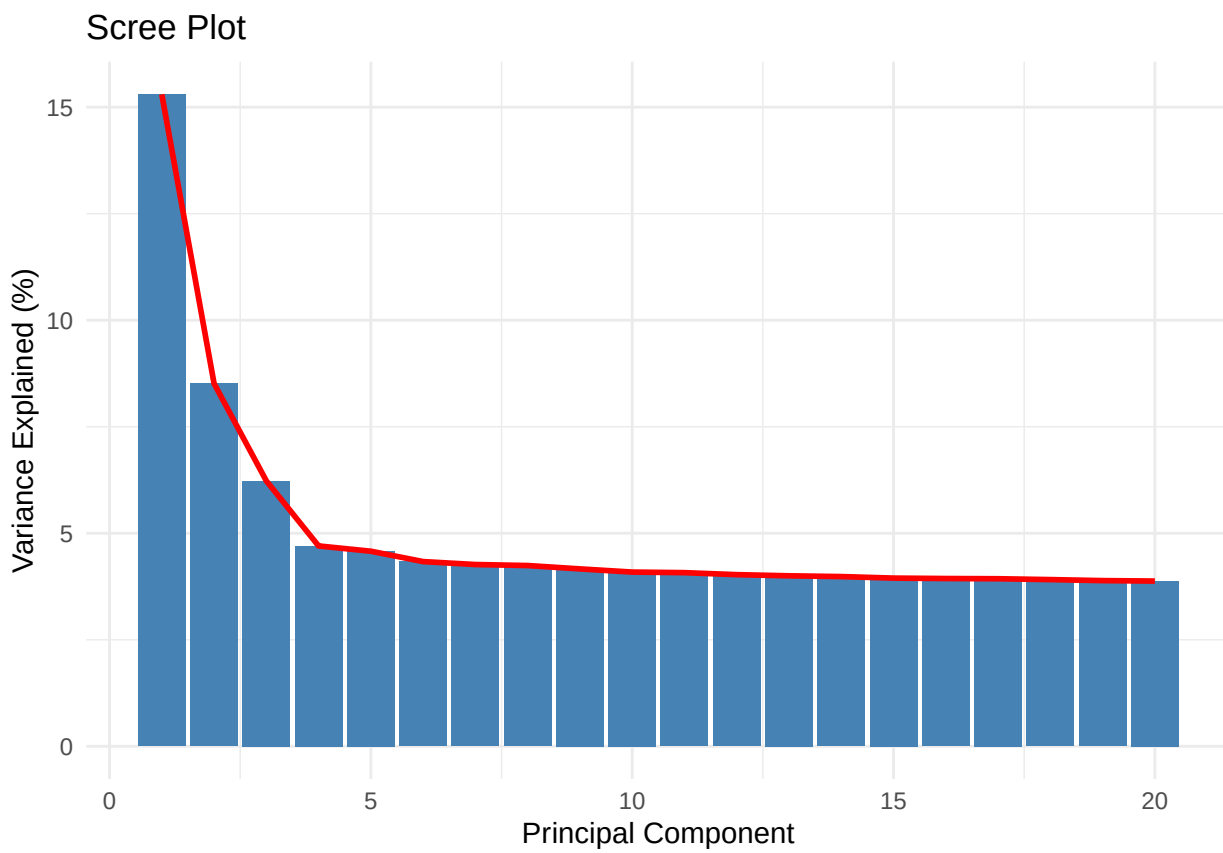



```r
# PC2 vs PC3
ggplot(pca_data, aes(x = PC2, y = PC3)) +
  geom_point(alpha = 0.5, color = "green") +
  labs(title = "PCA: PC2 vs PC3", x = "PC2", y = "PC3") +
  theme_minimal()
```



Creating a scree plot for the first 20 components.

```
# Load eigenvalues
eigenvalues <- read.table("mmc2/pca_results.eigenval", header = FALSE)

# Create a data frame with PC index and variance explained
scree_data <- data.frame(
  PC = 1:20,
  Variance = eigenvalues$V1[1:20] / sum(eigenvalues$V1) * 100  # Convert to percentage
)

# Plot scree plot
ggplot(scree_data, aes(x = PC, y = Variance)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_line(aes(group = 1), color = "red", linewidth = 1) +
  labs(title = "Scree Plot", x = "Principal Component", y = "Variance Explained (%)") +
  theme_minimal()
```



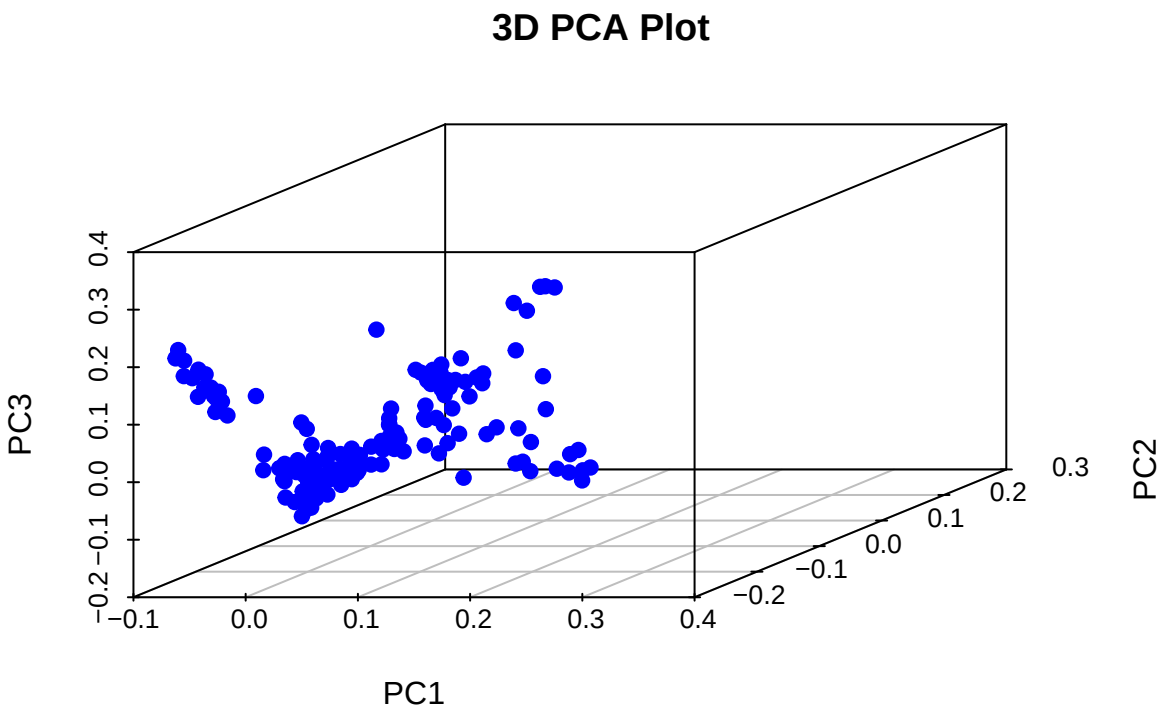
- Choosing first 5 components is enough as they explain the most variance.

Creating a 3D plot of the first three principal components.

```
# Install scatterplot3d package if not installed
if (!require("scatterplot3d")) install.packages("scatterplot3d", dependencies = TRUE)
```

```
## Loading required package: scatterplot3d
```

```
# Load the package
library(scatterplot3d)

# Create 3D scatter plot
scatterplot3d(pca_data$PC1, pca_data$PC2, pca_data$PC3,
              color = "blue", pch = 19,
              main = "3D PCA Plot",
              xlab = "PC1", ylab = "PC2", zlab = "PC3")
```



Part 2: Clustering in R

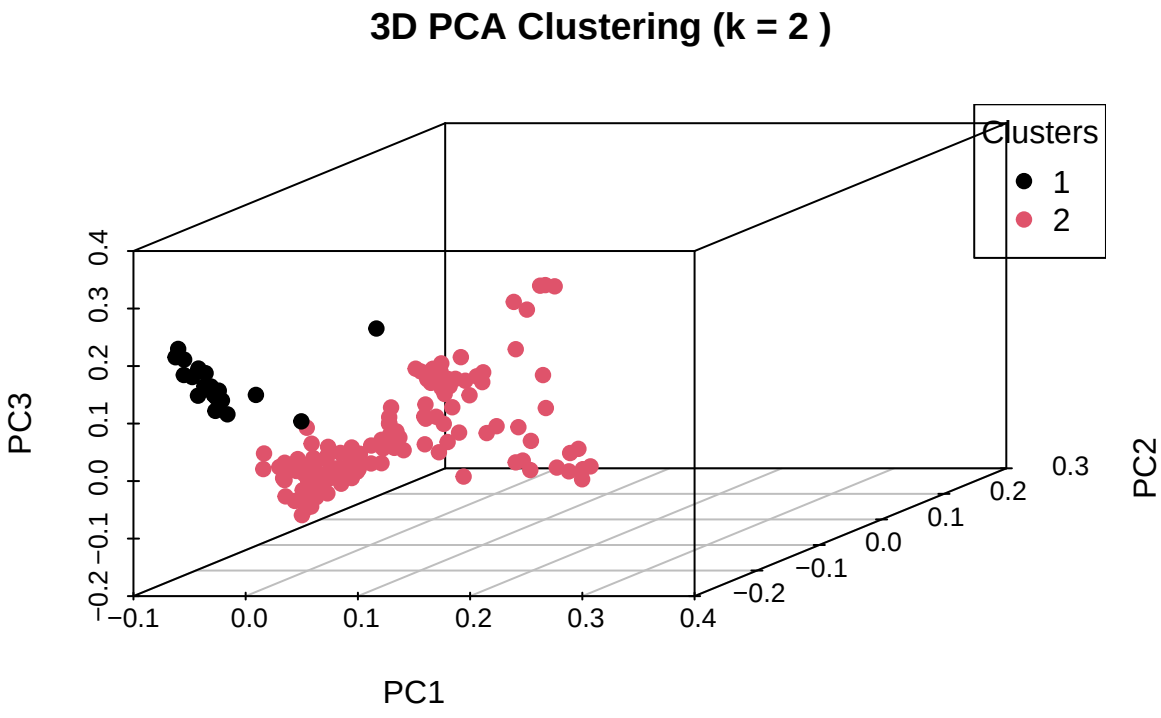
Choosing the first three principal components: PC1, PC2, PC3

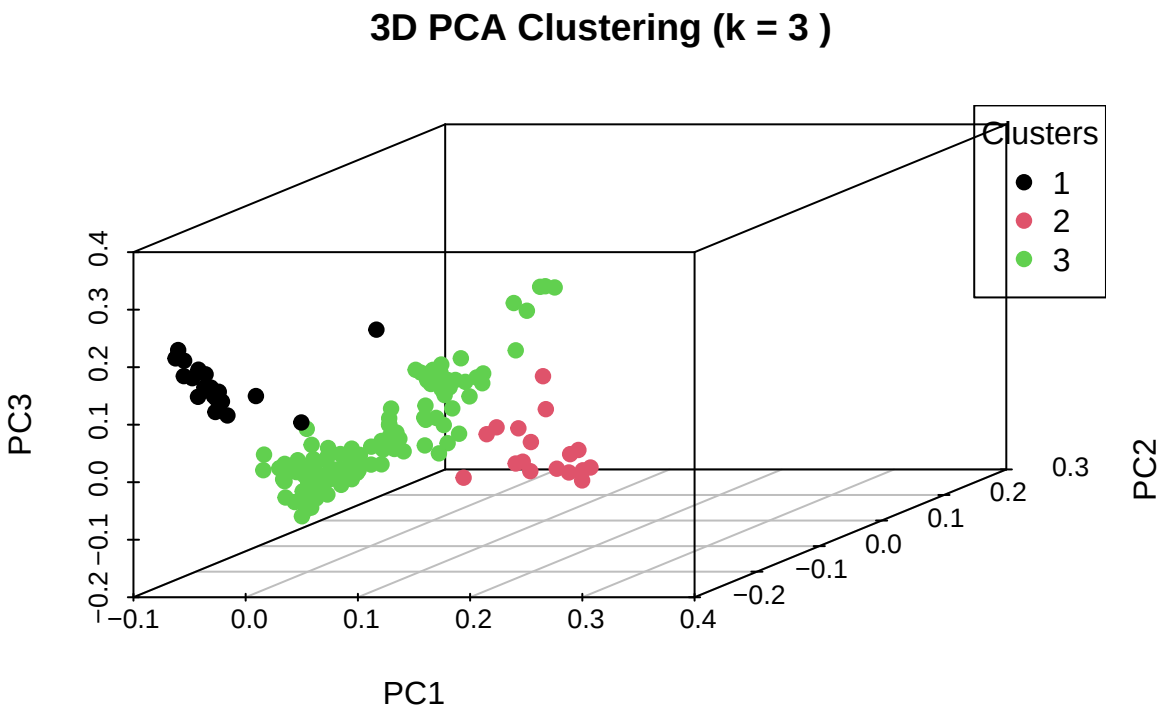
```
# Select only the first three principal components
pca_reduced <- pca_data[, c("PC1", "PC2", "PC3")]
```

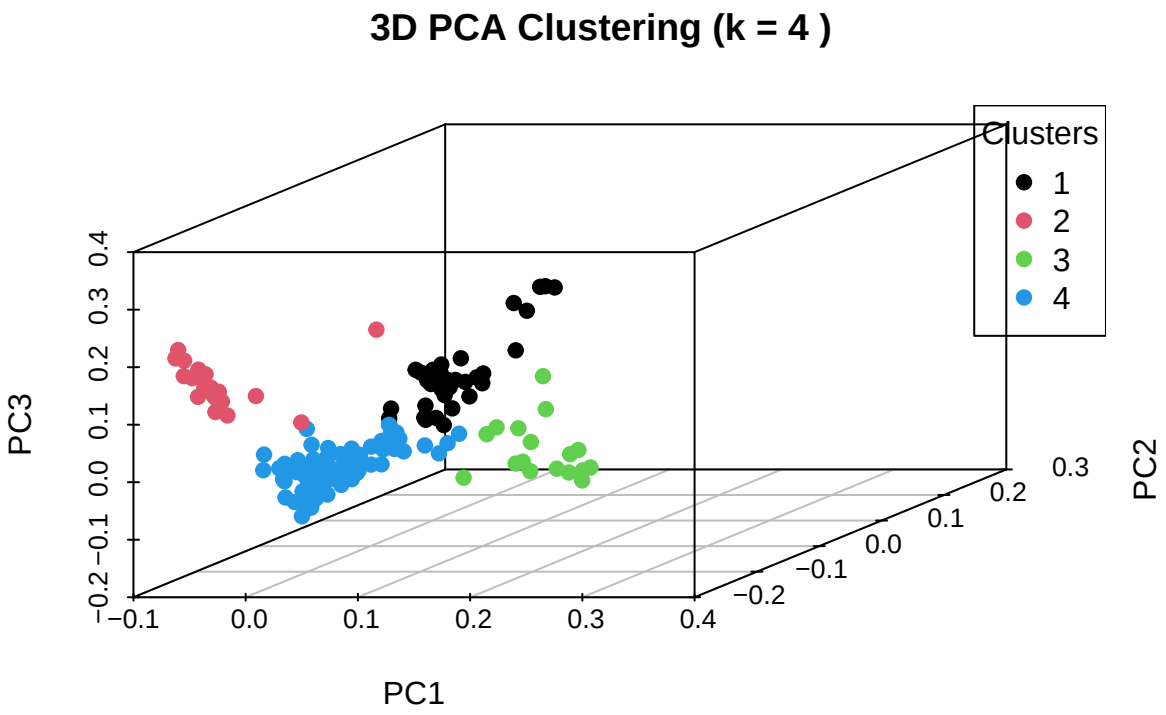
Use k-means clustering with different numbers of clusters.

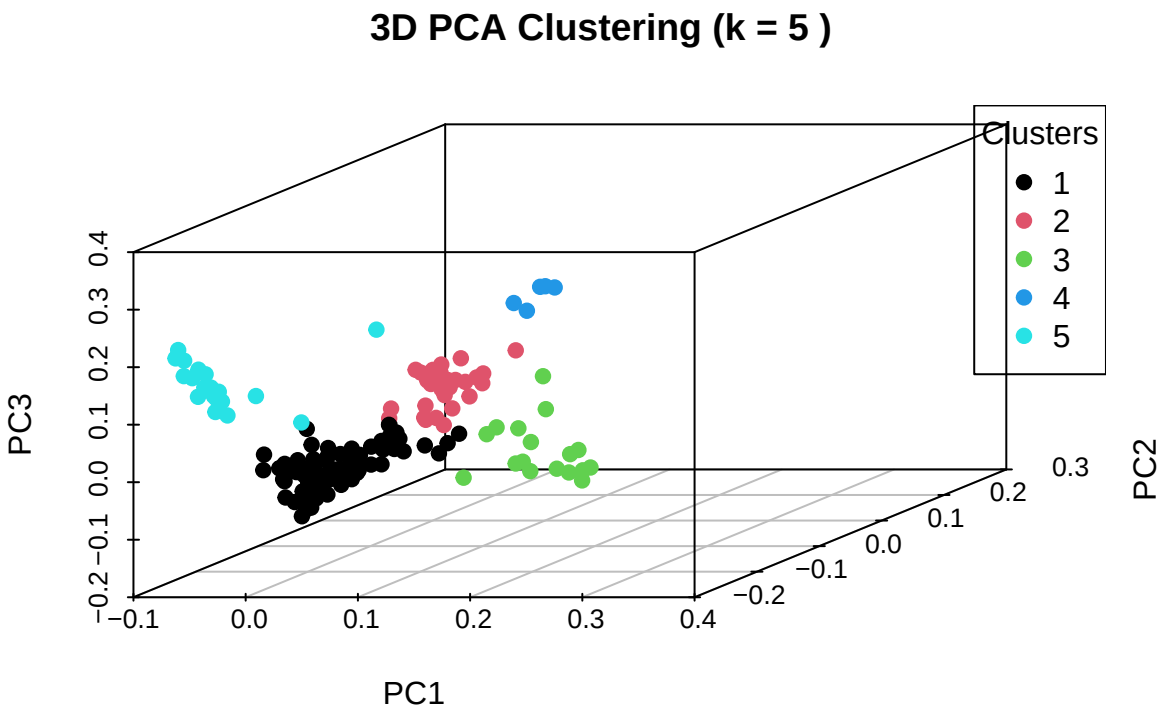
```
# Set seed for reproducibility
set.seed(42)

# Try different numbers of clusters
k_values <- c(2, 3, 4, 5, 7, 9)

# Loop through different k values
for (k in k_values) {
  # Perform k-means clustering
  kmeans_result <- kmeans(pca_reduced, centers = k, nstart = 25)

  # Add cluster labels to the dataset
  pca_reduced$Cluster <- as.factor(kmeans_result$cluster)

  # 3D visualization
```

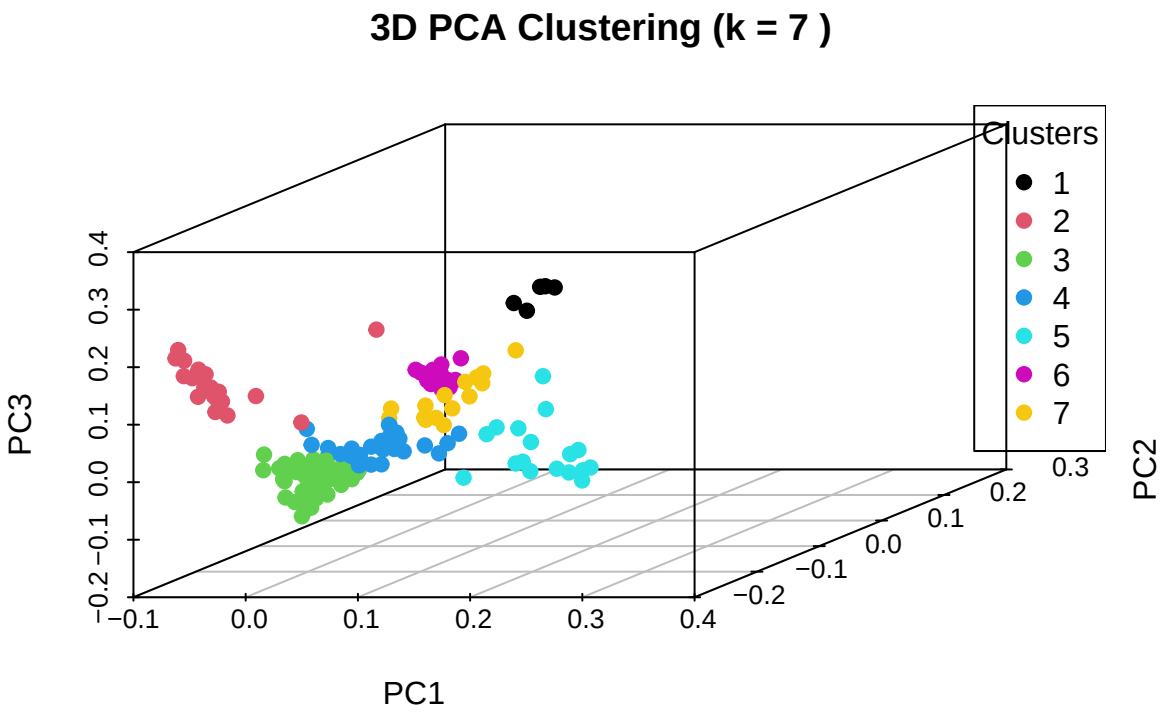
```r
  scatterplot3d(pca_reduced$PC1, pca_reduced$PC2, pca_reduced$PC3,
                color = as.numeric(pca_reduced$Cluster), pch = 19,
                main = paste("3D PCA Clustering (k =", k, ")"),
                xlab = "PC1", ylab = "PC2", zlab = "PC3")

  # Add legend
  legend("topright", legend = levels(pca_reduced$Cluster),
         col = 1:k, pch = 19, title = "Clusters")
}
```

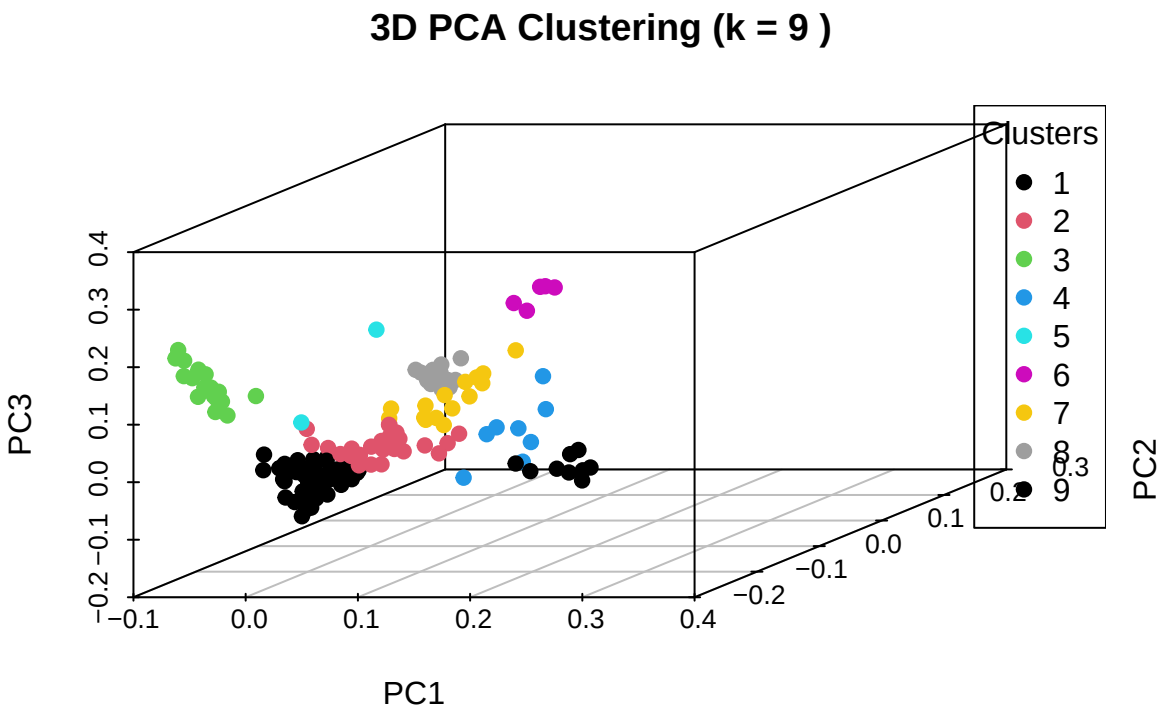



3D PCA Clustering (k = 3)
Clusters
1
2
3
PC3
PC1
PC2
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4


3D PCA Clustering (k = 4)
Clusters
1
2
3
4
PC3
PC1
PC2
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
−0.1
0.0
0.1
0.2
0.3
0.4
−0.2
−0.1
0.0
0.1
0.2
0.3


3D PCA Clustering (k = 5)
Clusters
1
2
3
4
5
PC3
PC1
PC2
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
−0.1
0.0
0.1
0.2
0.3
0.4
−0.2
−0.1
0.0
0.1
0.2
0.3


3D PCA Clustering (k = 7)
Clusters
1
2
3
4
5
6
7
PC3
PC1
PC2
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
−0.1
0.0
0.1
0.2
0.3
0.4
−0.2
−0.1
0.0
0.1
0.2
0.3



Determine the optimality of the number of clusters using Dunn's index.

```
# Load required libraries
if (!require("clValid")) install.packages("clValid", dependencies = TRUE)
```

```
## Loading required package: clValid
```

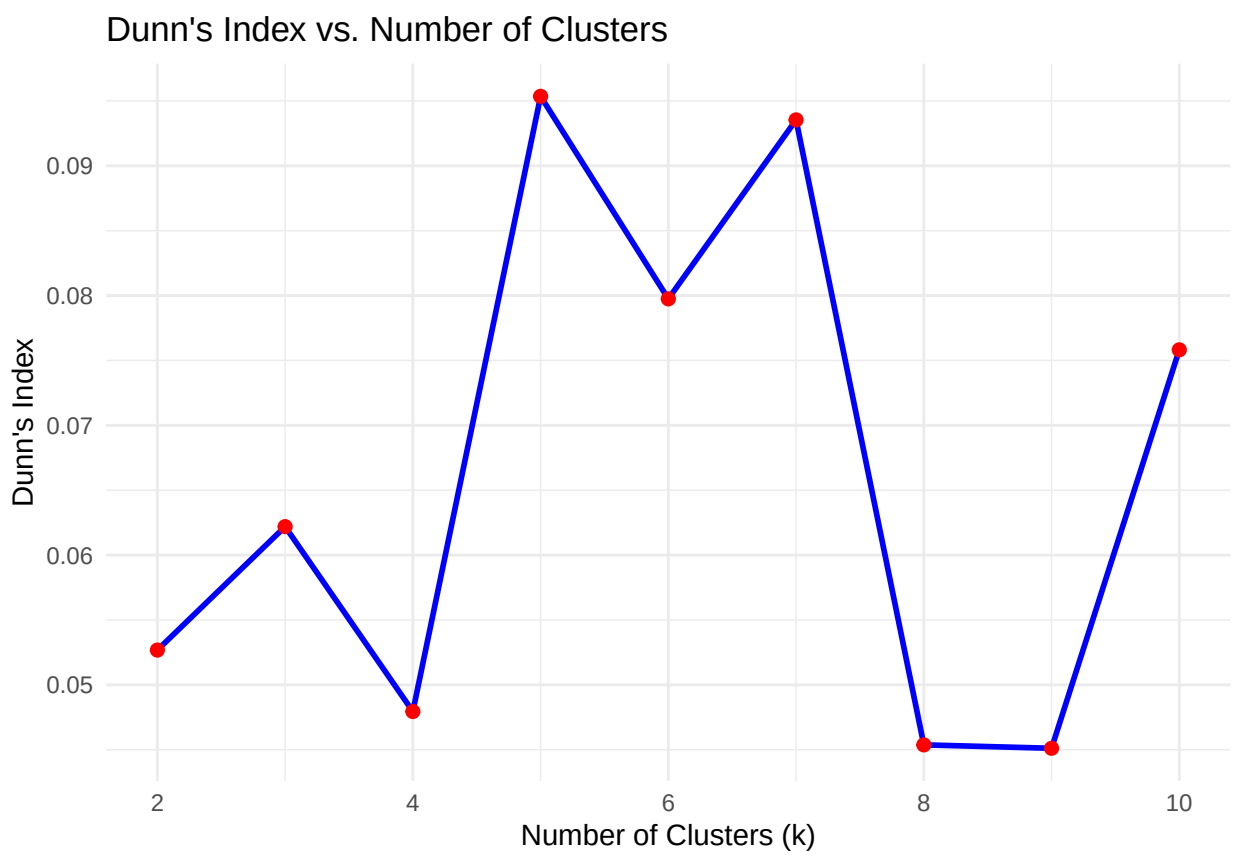
```
## Loading required package: cluster
```

```
if (!require("cluster")) install.packages("cluster", dependencies = TRUE)
library(clValid)
library(cluster)
```

```
# Select only the first three principal components
pca_reduced <- pca_data[, c("PC1", "PC2", "PC3")]

# Define range of k values to test
k_values <- 2:10  # Try from k=2 to k=10
dunn_index_values <- numeric(length(k_values))  # Store Dunn's index for each k

# Loop through different k values
for (i in seq_along(k_values)) {
  k <- k_values[i]

  # Perform k-means clustering
  kmeans_result <- kmeans(pca_reduced, centers = k, nstart = 25)
```

```r
  # Compute Dunn's index
  dunn_index_values[i] <- dunn(dist(pca_reduced), kmeans_result$cluster)
}
```

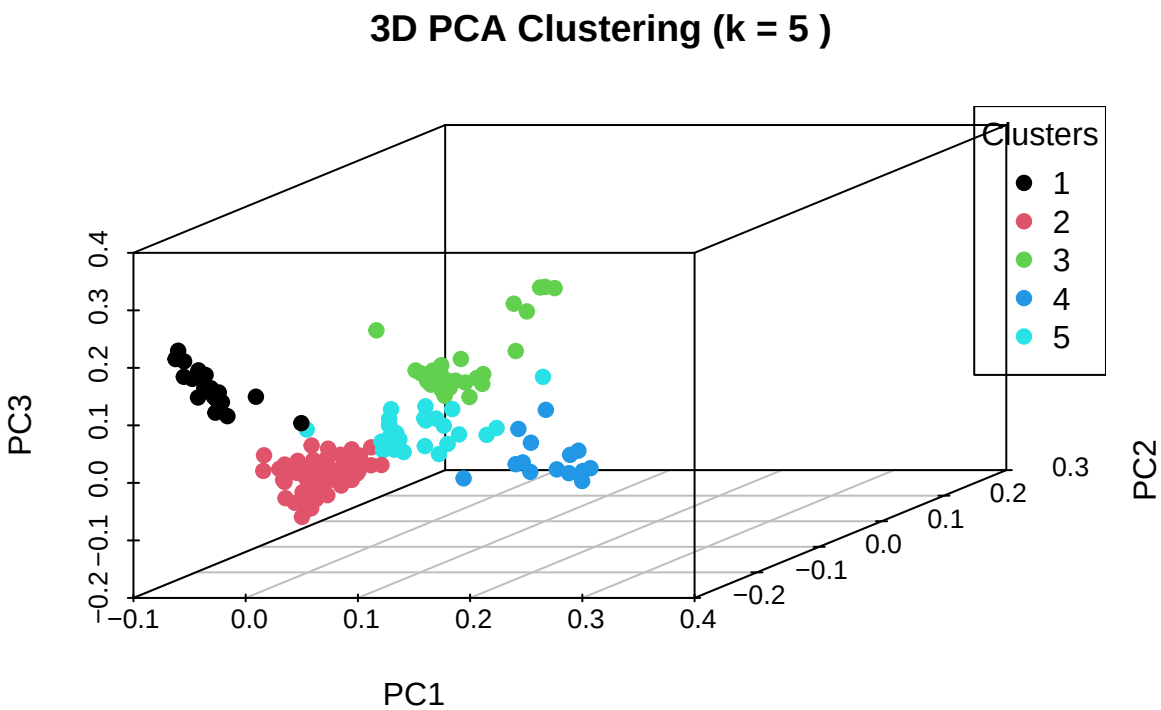
```r
# Create a data frame for plotting
dunn_data <- data.frame(k = k_values, Dunn_Index = dunn_index_values)

# Plot Dunn's Index vs. Number of Clusters
ggplot(dunn_data, aes(x = k, y = Dunn_Index)) +
  geom_line(color = "blue", linewidth = 1) +
  geom_point(color = "red", size = 2) +
  labs(title = "Dunn's Index vs. Number of Clusters",
       x = "Number of Clusters (k)", y = "Dunn's Index") +
  theme_minimal()
```



Perform k-means clustering with the optimal number of clusters.

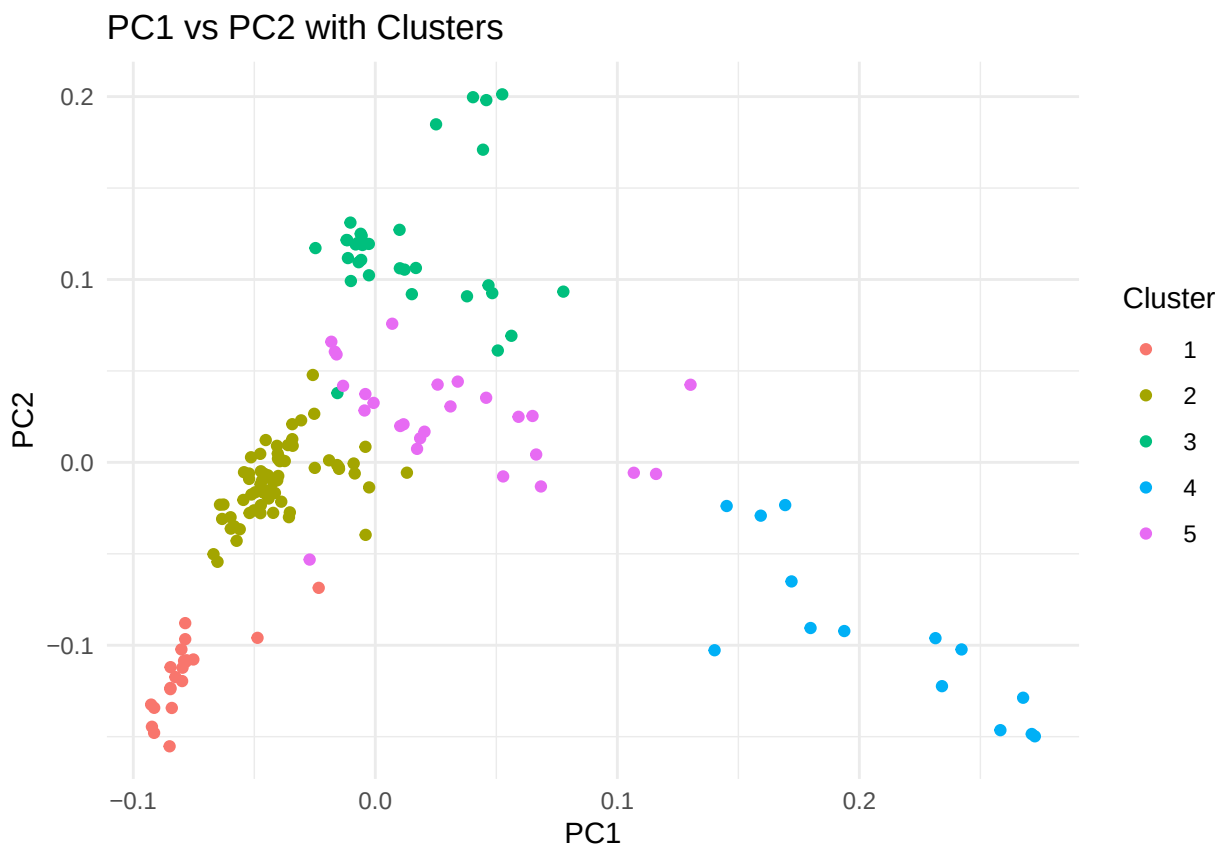
```r
set.seed(42)

# Perform k-means clustering with the optimal number of clusters
kmeans_result <- kmeans(pca_reduced, centers = 5, nstart = 25)

# Add cluster labels to the dataset
pca_reduced$Cluster <- as.factor(kmeans_result$cluster)

# 3D visualization
```

```r
scatterplot3d(pca_reduced$PC1, pca_reduced$PC2, pca_reduced$PC3,
              color = as.numeric(pca_reduced$Cluster), pch = 19,
              main = paste("3D PCA Clustering (k =", 5, ")"),
              xlab = "PC1", ylab = "PC2", zlab = "PC3")

# Add legend
legend("topright", legend = levels(pca_reduced$Cluster),
       col = 1:5, pch = 19, title = "Clusters")
```

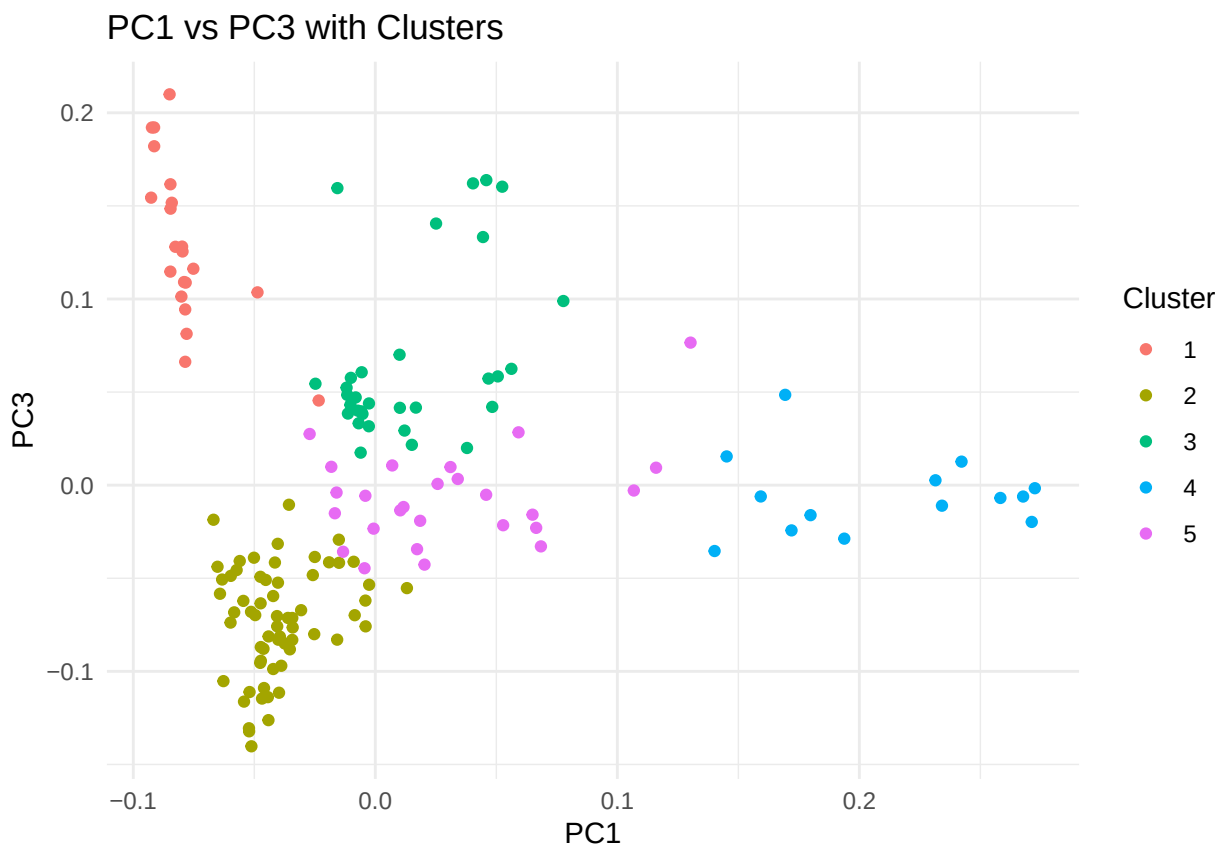


Visualizing the clusters corresponding to the subpopulations produced from each clustering on the PCA plots.

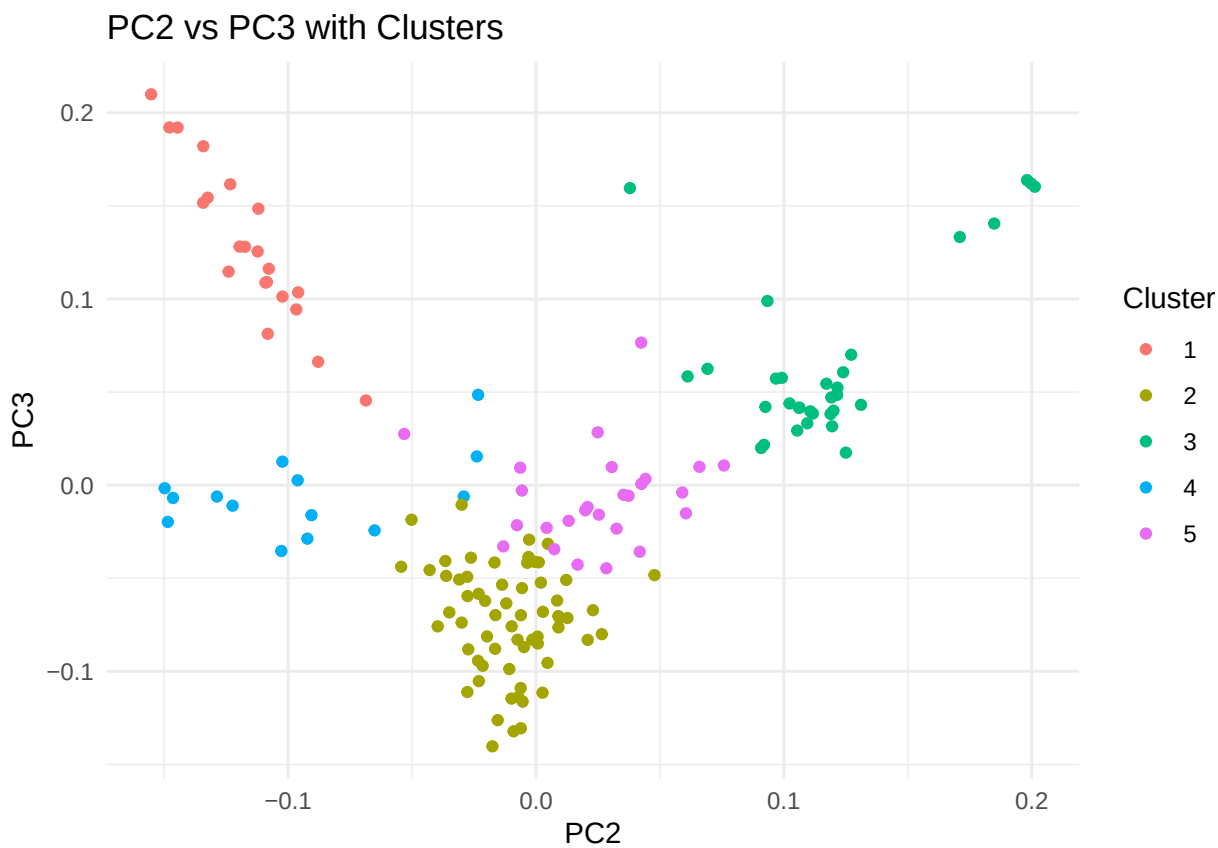
```r
# PC1 vs PC2 with Clusters
ggplot(pca_reduced, aes(x=PC1, y=PC2, color=Cluster)) +
  geom_point() + theme_minimal() +
  labs(title="PC1 vs PC2 with Clusters", x="PC1", y="PC2")
```




```r
# PC1 vs PC3 with Clusters
ggplot(pca_reduced, aes(x=PC1, y=PC3, color=Cluster)) +
  geom_point() + theme_minimal() +
  labs(title="PC1 vs PC3 with Clusters", x="PC1", y="PC3")
```

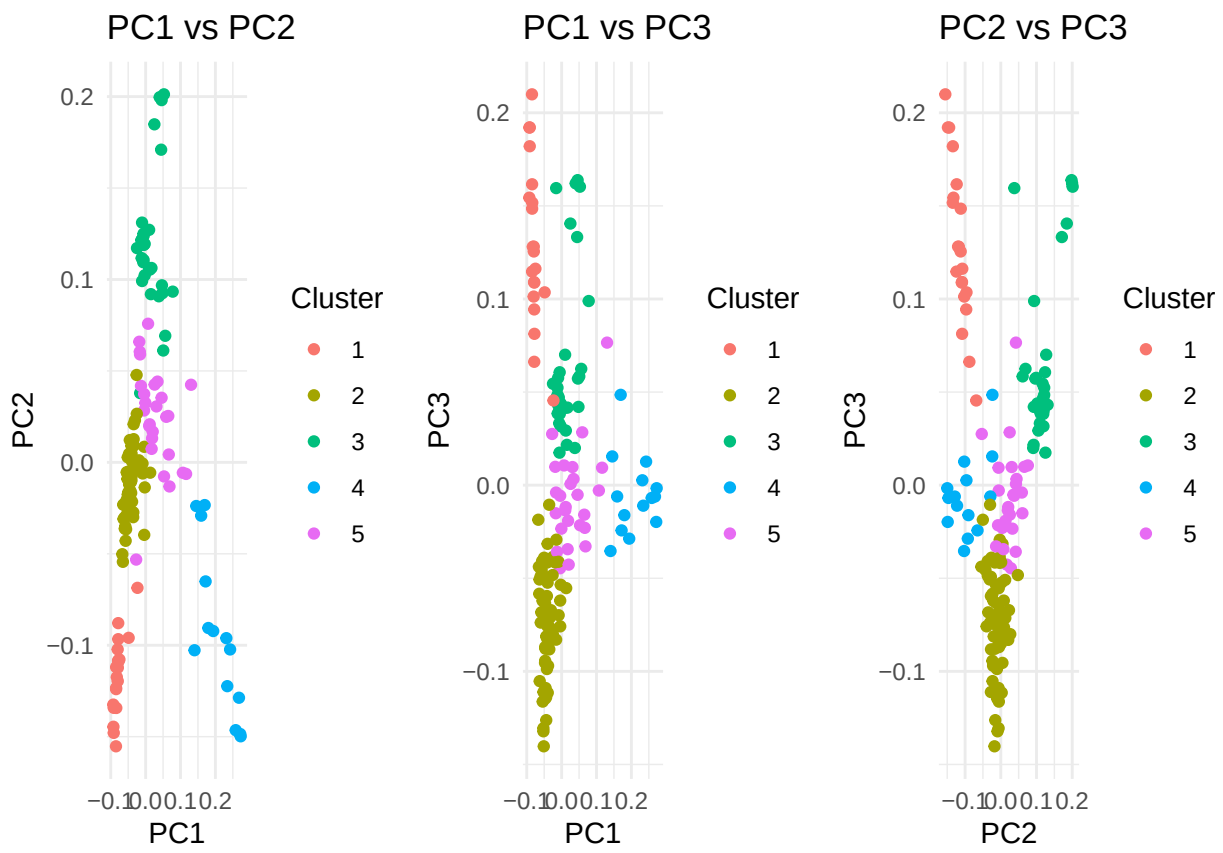



```r
# PC2 vs PC3 with Clusters
ggplot(pca_reduced, aes(x=PC2, y=PC3, color=Cluster)) +
  geom_point() + theme_minimal() +
  labs(title="PC2 vs PC3 with Clusters", x="PC2", y="PC3")
```



Creating a side-by-side comparison of the clusters formed by k-means.

```
library(gridExtra)

plot1 <- ggplot(pca_reduced, aes(x=PC1, y=PC2, color=Cluster)) +
  geom_point() + theme_minimal() + labs(title="PC1 vs PC2")

plot2 <- ggplot(pca_reduced, aes(x=PC1, y=PC3, color=Cluster)) +
  geom_point() + theme_minimal() + labs(title="PC1 vs PC3")

plot3 <- ggplot(pca_reduced, aes(x=PC2, y=PC3, color=Cluster)) +
  geom_point() + theme_minimal() + labs(title="PC2 vs PC3")

grid.arrange(plot1, plot2, plot3, ncol=3)
```


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