# Lab5: PCA and Clustering

Maria Bassem Emil

2024-03-28

ID: 20011141

# Part 1: Principal Component Analysis Using PLINK (35 Points)

## Task 1.1: QC and PCA (15 Points)

1. Run Minor Allele Frequency count on your dataset using PLINK and the flag –freq. Provide a screenshot of the head of the file. Explain the output

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --freq --out qatari_freq
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to qatari_freq.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --freq
  --out qatari_freq

4429 MB RAM detected; reserving 2214 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 qatari_freq.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
--freq: Allele frequencies (founders only) written to qatari_freq.frq .
```

Runing Minor Allele Frequency count

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ head qatari_freq.frq
 CHR          SNP   A1   A2          MAF  NCHROBS
   1   rs10907175    C    A      0.08974      312
   1    rs7519837    T    C       0.4387      310
   1   rs10907187    A    G       0.2596      312
   1    rs6603803    G    A       0.3141      312
   1    rs6688000    A    G       0.1346      312
   1    rs7513222    A    G       0.3013      312
   1    rs3128309    A    G      0.05449      312
   1   rs12084736    T    C       0.1763      312
   1   rs12045693    A    C       0.2564      312
```

Head of freq file

In the provided output:

- **CHR**: Chromosome number or identifier. In our case it is 1.
- **SNP**: SNP identifier.
- **A1**: Allele 1 (minor allele).
- **A2**: Allele 2 (major allele).
- **MAF**: Minor allele frequency.
- **NCHROBS**: Number of allele observations used to calculate the frequency.

For example, if we took the first SNP, we find that the SNP id is rs10907175 on chromosome 1. This SNP has alleles C (minor) and A (major). The minor allele frequency (MAF) is 0.08974. Allele C (minor) was observed 312 times (156 samples * 2 alleles), that is this SNP was observed in all the samples.

### 2. Run QC on your dataset using PLINK

Try the following flags separately –maf –geno and –hwe filters. Try and report different levels and thresholds focusing on the number of variants removed. Add screenshots of the log files output at the end of each trial (3 max with meaningful values for each flag).

***In –maf, as the value increases, the number of samples remove will increase.***

- –maf 0.05

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --maf 0.05 --recode --out filtered_maf_0.05
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to filtered_maf_0.05.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --maf 0.05
  --out filtered_maf_0.05
  --recode

4429 MB RAM detected; reserving 2214 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 filtered_maf_0.05.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
0 variants removed due to minor allele threshold(s)
(--maf/--max-maf/--mac/--max-mac).
67735 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode ped to filtered_maf_0.05.ped + filtered_maf_0.05.map ... done.
```

- –maf 0.075

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --maf 0.075 --recode --out filtered_maf_0.075
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to filtered_maf_0.075.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --maf 0.075
  --out filtered_maf_0.075
  --recode

4429 MB RAM detected; reserving 2214 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 filtered_maf_0.075.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
9517 variants removed due to minor allele threshold(s)
(--maf/--max-maf/--mac/--max-mac).
58218 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode ped to filtered_maf_0.075.ped + filtered_maf_0.075.map ... done.
```

- –maf 0.1

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --maf 0.1 --recode --out filtered_maf_0.1
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to filtered_maf_0.1.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --maf 0.1
  --out filtered_maf_0.1
  --recode

4429 MB RAM detected; reserving 2214 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 filtered_maf_0.1.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
16606 variants removed due to minor allele threshold(s)
(--maf/--max-maf/--mac/--max-mac).
51129 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode ped to filtered_maf_0.1.ped + filtered_maf_0.1.map ... done.
```

***In –geno, as the threshold decreases, the number of samples remove will increase.***

- –geno 0.001

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --geno 0.001 --recode --out filtered_geno_0.001
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to filtered_geno_0.001.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --geno 0.001
  --out filtered_geno_0.001
  --recode

4429 MB RAM detected; reserving 2214 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 filtered_geno_0.001.hh );
many commands treat these as missing.
Total genotyping rate is 0.998816.
12509 variants removed due to missing genotype data (--geno).
55226 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode ped to filtered_geno_0.001.ped + filtered_geno_0.001.map ... done.
```

- –geno 0.00003

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --geno 0.00003 --recode --out filtered_geno_0.00003
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to filtered_geno_0.00003.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --geno 0.00003
  --out filtered_geno_0.00003
  --recode

4429 MB RAM detected; reserving 2214 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 filtered_geno_0.00003.hh );
many commands treat these as missing.
Total genotyping rate is 0.998816.
12509 variants removed due to missing genotype data (--geno).
55226 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode ped to filtered_geno_0.00003.ped + filtered_geno_0.00003.map ... done.
```

- –geno 0.05

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --geno 0.05 --recode --out filtered_geno_0.05
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to filtered_geno_0.05.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --geno 0.05
  --out filtered_geno_0.05
  --recode

4429 MB RAM detected; reserving 2214 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 filtered_geno_0.05.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
0 variants removed due to missing genotype data (--geno).
67735 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode ped to filtered_geno_0.05.ped + filtered_geno_0.05.map ... done.
```

***In –hwe, as the threshold decreases, the number of samples remove will increase.***

- –hwe 1e-6 (low hwe threshold) –> high stringent

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --hwe 1e-6 --recode --out filtered_hwe_1e-6
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to filtered_hwe_1e-6.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --hwe 1e-6
  --out filtered_hwe_1e-6
  --recode

4429 MB RAM detected; reserving 2214 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 filtered_hwe_1e-6.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
Warning: --hwe observation counts vary by more than 10%, due to the X
chromosome.  You may want to use a less stringent --hwe p-value threshold for X
chromosome variants.
--hwe: 0 variants removed due to Hardy-Weinberg exact test.
67735 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode ped to filtered_hwe_1e-6.ped + filtered_hwe_1e-6.map ... done.
```

- –hwe 1e-4 (moderate hwe threshold) –> moderate stringent

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --hwe 1e-4 --recode --out filtered_hwe_1e-4
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to filtered_hwe_1e-4.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --hwe 1e-4
  --out filtered_hwe_1e-4
  --recode

4429 MB RAM detected; reserving 2214 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 filtered_hwe_1e-4.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
Warning: --hwe observation counts vary by more than 10%, due to the X
chromosome.  You may want to use a less stringent --hwe p-value threshold for X
chromosome variants.
--hwe: 0 variants removed due to Hardy-Weinberg exact test.
67735 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode ped to filtered_hwe_1e-4.ped + filtered_hwe_1e-4.map ... done.
```

- –hwe 0.001 (high hwe threshold) –> less stringent

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --hwe 0.001 --recode --out filtered_hwe_0.001
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to filtered_hwe_0.001.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --hwe 0.001
  --out filtered_hwe_0.001
  --recode

4429 MB RAM detected; reserving 2214 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 filtered_hwe_0.001.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
Warning: --hwe observation counts vary by more than 10%, due to the X
chromosome.  You may want to use a less stringent --hwe p-value threshold for X
chromosome variants.
--hwe: 0 variants removed due to Hardy-Weinberg exact test.
67735 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode ped to filtered_hwe_0.001.ped + filtered_hwe_0.001.map ... done.
```

Run the final version of your QC using all the flags combined and report the final number of variants. Use the following thresholds (hwe: 0.01, maf: 0.1, geno: 0.001)

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --hwe 0.01 --maf 0.1 --geno 0.001 --recode --out combined_qc
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to combined_qc.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --geno 0.001
  --hwe 0.01
  --maf 0.1
  --out combined_qc
  --recode

4429 MB RAM detected; reserving 2214 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 combined_qc.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 less stringent --hwe p-value threshold for X
chromosome variants.
--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 combined_qc.ped + combined_qc.map ... done.
```

Run PCA on your dataset using the PLINK and –pca flag. You should recode the data to be in the format ped/map

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --file combined_qc --pca --recode --out pca_qc_results
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to pca_qc_results.log.
Options in effect:
  --file combined_qc
  --out pca_qc_results
  --pca
  --recode

4429 MB RAM detected; reserving 2214 MB for main workspace.
.ped scan complete (for binary autoconversion).
Performing single-pass .bed write (40411 variants, 156 people).
--file: pca_qc_results-temporary.bed + pca_qc_results-temporary.bim +
pca_qc_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_qc_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.
--pca: Results saved to pca_qc_results.eigenval and pca_qc_results.eigenvec .
--recode ped to pca_qc_results.ped + pca_qc_results.map ... done.
```

## Task 1.2: PCA Visualization (20 Points)

Explore Egienvectors and Eigenvalues. For Ubuntu/MacOS/Windows WSL users, use the awk, vi, and head commands to view the eigenvalues and eigenvectors.

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ head pca_qc_results.eigenval
4.43856
2.46985
1.80692
1.36404
1.32812
1.25756
1.23744
1.23056
1.2077
1.1861
```

```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ awk '{print$1,$2,$3,$4}' pca_qc_results.eigenvec > pcs
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ head pcs
QBC-092 QBC-092 0.0257471 0.0425202
QBC-256 QBC-256 -0.0394316 0.000642535
QBC-107 QBC-107 -0.0401049 -0.00743885
QBC-171 QBC-171 -0.0156592 0.0378835
QPRC-110 QPRC-110 -0.0118682 0.121597
QBC-240 QBC-240 0.056231 0.0692041
QPRC-019 QPRC-019 -0.092289 -0.144578
QBC-183 QBC-183 -0.0572974 -0.0428937
QBC-086 QBC-086 -0.00267536 0.119443
QPRC-039 QPRC-039 0.0340692 0.0441574
```

Libraries and Imports

```
library(ggplot2)
# install.packages("scatterplot3d")
library(scatterplot3d)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#install.packages("clValid")
library(clValid)
```

```
## Warning: package 'clValid' was built under R version 4.3.3
```

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

```
# install.packages("dendextend")
library(dendextend)
```

```
## Warning: package 'dendextend' was built under R version 4.3.3
```

```
##
## ---------------------
## Welcome to dendextend version 1.17.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
# install.packages("gridExtra")
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.3.3
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

Load the PCA results into R.

```
# Load PCA results
eigen_vec <- read.table("D:\\Third Year Computer\\Term 2\\Bio\\Labs\\Lab 5\\pca_qc_results.eigenvec", header=FALS
E)

# Set column names
colnames(eigen_vec) <- c("Family ID", "Sample ID", paste0("PC", 1:(ncol(eigen_vec) - 2)))
head(eigen_vec)
```

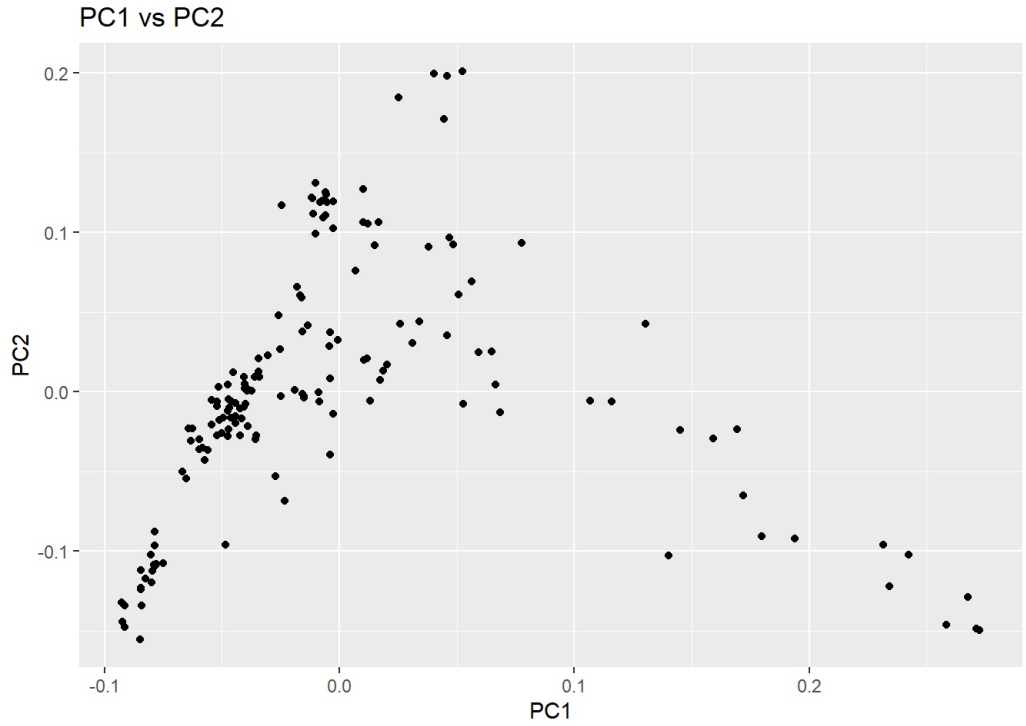
```
##   Family ID Sample ID        PC1          PC2          PC3         PC4
## 1   QBC-092   QBC-092  0.0257471  0.042520200  0.000671924 -0.00464458
## 2   QBC-256   QBC-256 -0.0394316  0.000642535 -0.081346600  0.00336033
## 3   QBC-107   QBC-107 -0.0401049 -0.007438850 -0.082995400 -0.00482384
##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

```
eigen_values <- read.table("D:\\Third Year Computer\\Term 2\\Bio\\Labs\\Lab 5\\pca_qc_results.eigenval", header=F
ALSE)
head(eigen_values)
```

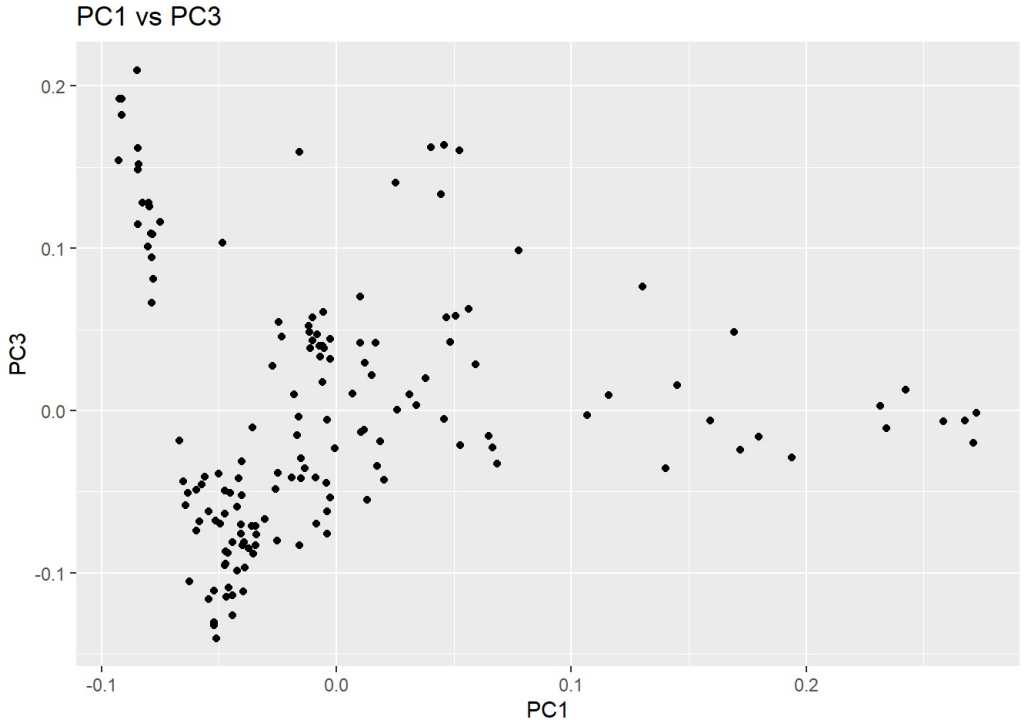
```
##        V1
## 1 4.43856
## 2 2.46985
## 3 1.80692
## 4 1.36404
## 5 1.32812
## 6 1.25756
```

Create 2D scatter plots comparing PC1 vs PC2, PC1 vs PC3, and PC2 vs PC3 using ggplot2.

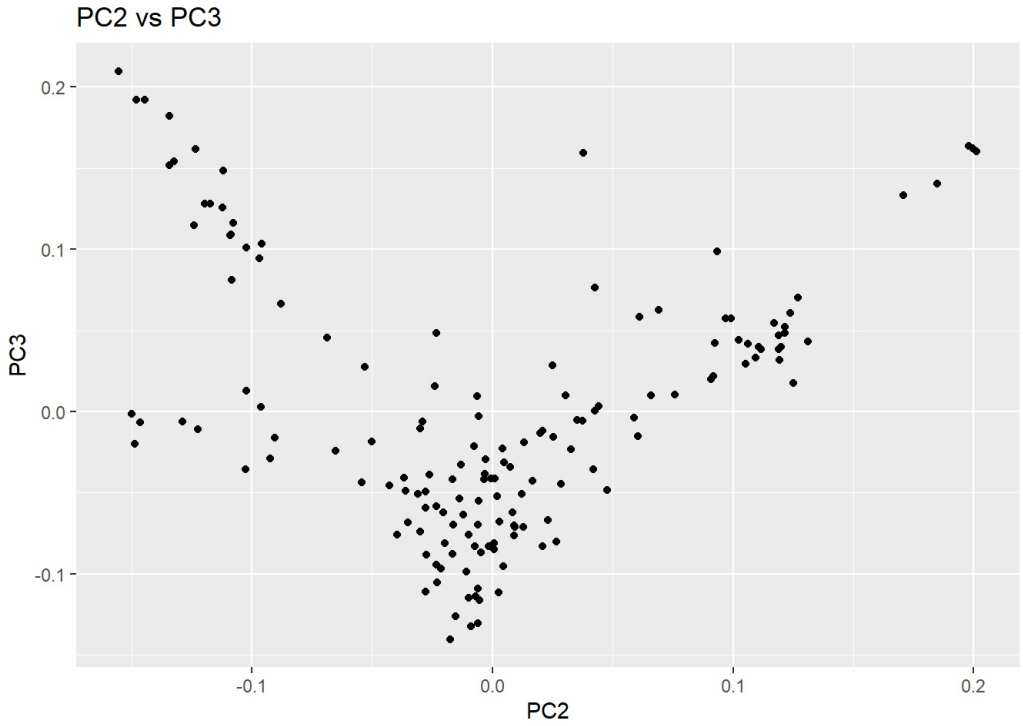
```
# Create scatter plot: PC1 vs PC2
ggplot(eigen_vec, aes(x = PC1, y = PC2)) +
  geom_point() +
  labs(title = "PC1 vs PC2")
```



```
# Create scatter plot: PC1 vs PC3
ggplot(eigen_vec, aes(x = PC1, y = PC3)) +
  geom_point() +
  labs(title = "PC1 vs PC3")
```



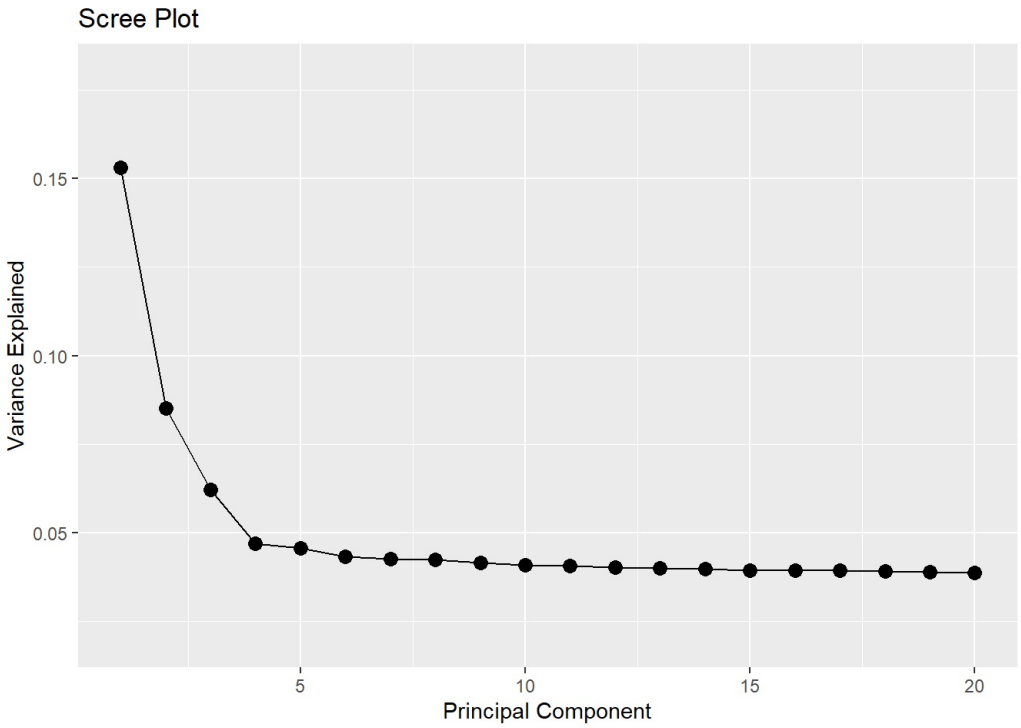

```
# Create scatter plot: PC2 vs PC3
ggplot(eigen_vec, aes(x = PC2, y = PC3)) +
  geom_point() +
  labs(title = "PC2 vs PC3")
```




Create a scree plot for the first 20 components with the explained variance.

```
# Calculate total sum of eigenvalues
total_variance <- sum(eigen_values$V1)
# Explained variance
explained_variance <- eigen_values$V1 / total_variance
variance_20 <- explained_variance[1:20]

# Scree plot
# instead of geom_line() we could use geom_col() according to geeks for geeks
qplot(c(1:20), variance_20) +
 geom_line() +
 geom_point(size=3)+
 xlab("Principal Component") +
 ylab("Variance Explained") +
 ggtitle("Scree Plot") +
 ylim(0.02, 0.18)
```

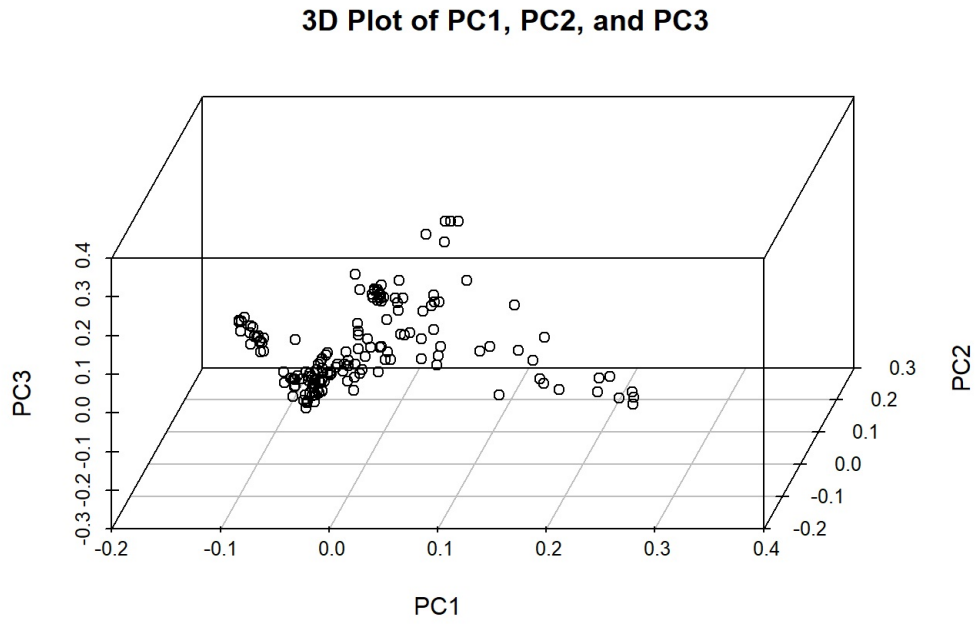
```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




Install and use the scatterplot3d package for 3D plots.

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

```
scatterplot3d(eigen_vec$PC1, eigen_vec$PC2, eigen_vec$PC3,
              main="3D Plot of PC1, PC2, and PC3",
              xlab="PC1", ylab="PC2", zlab="PC3",
              angle=75)
```




# Part 2: Clustering in R (40 Points)

## Task 2.1: Perform Clustering (10 Points)

Reduce the dimensionality of the dataset by only choosing the first three pricipal components PC1, PC2, PC3

```
# Select the first three principal components
eigen_vec_3 <- eigen_vec %>%
  select(PC1, PC2, PC3)
head(eigen_vec_3)
```

```
##          PC1          PC2          PC3
## 1  0.0257471  0.042520200  0.000671924
## 2 -0.0394316  0.000642535 -0.081346600
## 3 -0.0401049 -0.007438850 -0.082995400
## 4 -0.0156592  0.037883500  0.159538000
## 5 -0.0118682  0.121597000  0.052364500
## 6  0.0562310  0.069204100  0.062476200
```

We will perform clustering techniques to find the clusters that correspond to different subpopulations in this population

- Use k-means clustering with kmeans function

- Try different number of clusters (k)

- Determine the optimality of the number of clusters using Dunn's index or Xie Beni's index. HINT: for Dunn's index use the dunn function. For Xie Beni's index use the fclustIndex function

```
# Perform k-means clustering with different values of k
k_values <- 3:12
# dunn
dunn_indices <- numeric(length(k_values))

for (i in seq_along(k_values)) {
  k <- k_values[i]
  # Perform k-means clustering
  # nstart > 1 is recommended
  km.res <- kmeans(eigen_vec_3, centers = k, nstart = 25)

  Dist <- dist(eigen_vec_3, method="euclidean")
  # Calculate Dunn's index
  dunn_indices[i] <- dunn(Dist, km.res$cluster)


  # Print results
  cat("For k =", k, ", Dunn's Index:", dunn_indices[i], "\n")
}
```

```
## For k = 3 , Dunn's Index: 0.03427011
## For k = 4 , Dunn's Index: 0.04794632
## For k = 5 , Dunn's Index: 0.09535199
## For k = 6 , Dunn's Index: 0.07974664
## For k = 7 , Dunn's Index: 0.03447875
## For k = 8 , Dunn's Index: 0.04126353
## For k = 9 , Dunn's Index: 0.07626233
## For k = 10 , Dunn's Index: 0.06649092
## For k = 11 , Dunn's Index: 0.07581813
## For k = 12 , Dunn's Index: 0.02513609
```

```
# Dunn's index has a value between zero and infinity, and should be maximized
best_k_dunn <- k_values[which.max(dunn_indices)]
best_dunn <- max(dunn_indices)

# Print the best k and its corresponding indices
cat("Best k according to Dunn's index:", best_k_dunn, "\n")
```

```
## Best k according to Dunn's index: 5
```

```
km_final <- kmeans(eigen_vec_3, centers = best_k_dunn, nstart = 25)
```

## Task 2.2: Perform Hierarchical Clustering (10 Points)

Try hierarchical clustering methods, once with single linkage and another with average linkage clustering.

Use the hclust function. Note that you need to specify the method as either “single” for single linkage or “average” for average linkage.

```
Dist <- dist(eigen_vec_3, method="euclidean")

hclust_single <- hclust(Dist, method = 'single')
hclust_avg <- hclust(Dist, method = 'average')
```

Try different number of clusters

Determine the optimality of the number of clusters using Dunn’s index or Xie Beni’s index for the average linkage clustering

```
num_clusters <- 2:10
# Store Dunn's index or Xie Beni's index for each number of clusters
dunn_hier_avg_indices <- numeric(length(num_clusters))

for (i in seq_along(num_clusters)) {
  k <- num_clusters[i]

    # Cut the dendrogram to get clusters
  clusters_avg <- cutree(hclust_avg, k)

  # Dunn's index
  dunn_hier_avg_indices[i] <- dunn(Dist, clusters_avg)

  cat("For k =", k, ", Dunn's Average Index:", dunn_hier_avg_indices[i], "\n")
}
```

```
## For k = 2 , Dunn's Average Index: 0.1195777
## For k = 3 , Dunn's Average Index: 0.1195777
## For k = 4 , Dunn's Average Index: 0.1055171
## For k = 5 , Dunn's Average Index: 0.1312304
## For k = 6 , Dunn's Average Index: 0.1312304
## For k = 7 , Dunn's Average Index: 0.1312304
## For k = 8 , Dunn's Average Index: 0.1244844
## For k = 9 , Dunn's Average Index: 0.1244844
## For k = 10 , Dunn's Average Index: 0.1244844
```

```
# Dunn's index has a value between zero and infinity, and should be maximized
best_hier_k_dunn <- k_values[which.max(dunn_hier_avg_indices)]
best_hier_dunn <- max(dunn_hier_avg_indices)

# Print the best k and its corresponding indices
cat("Best k according to Dunn's index:", best_hier_k_dunn, "\n")
```

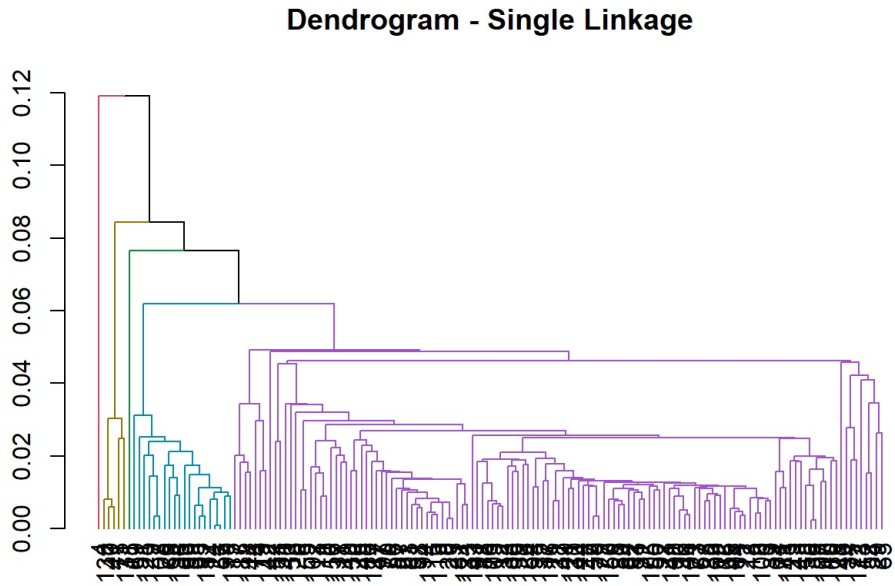
```
## Best k according to Dunn's index: 6
```

Determine the optimality of the number of clusters using common sense (expert eye) for single linkage clustering.

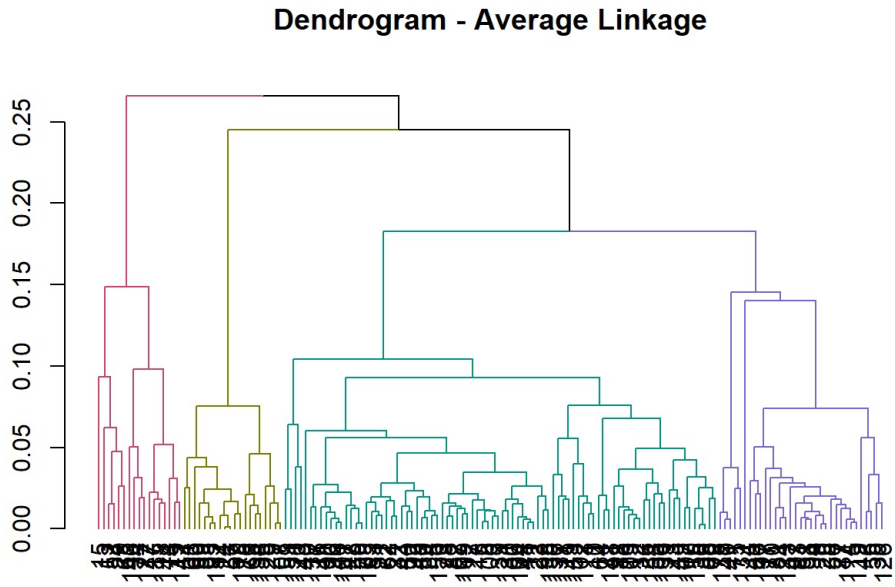
I drew the dendrogram plot from the single linkage clustering. Then I counted each branch as one cluster, thus I obtained k = 5 as the best k (num of clusters)

Plot the dendrograms using the plot function

```
suppressPackageStartupMessages(library(dendextend))
single_dend_obj <- as.dendrogram(hclust_single)
single_col_dend <- color_branches(single_dend_obj, h = 0.05)
plot(single_col_dend, main = "Dendrogram - Single Linkage")
```




```
avg_dend_obj <- as.dendrogram(hclust_avg)
avg_col_dend <- color_branches(avg_dend_obj, h = 0.15)
plot(avg_col_dend, main = "Dendrogram - Average Linkage")
```



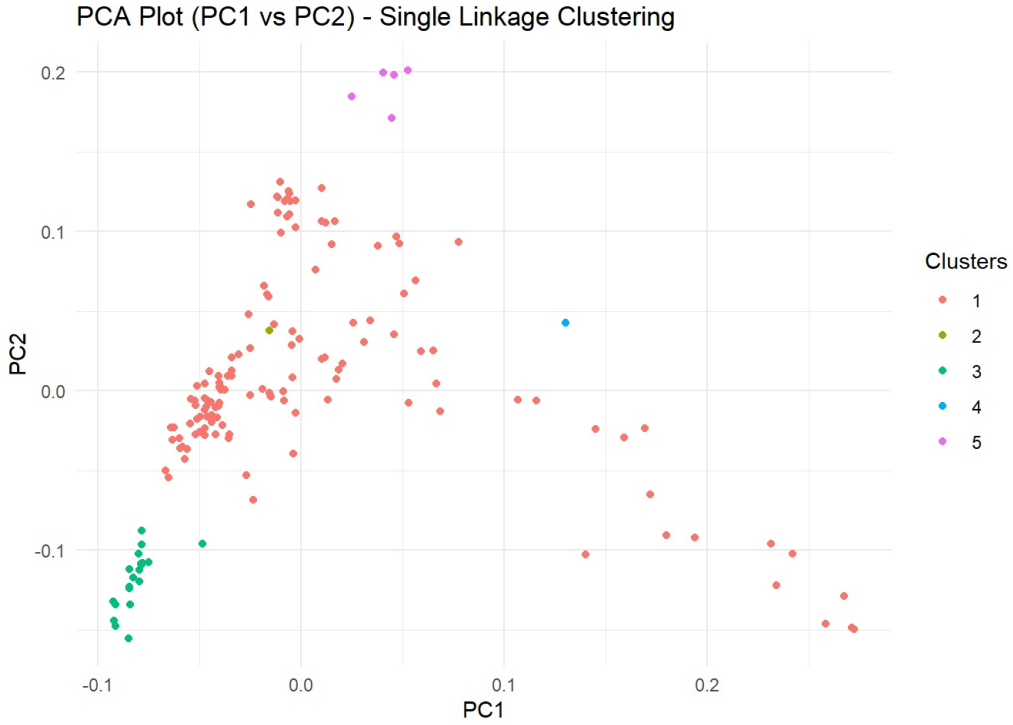

# Task 2.3: Visualize Clusters (20 Points)

```
clusters_kmeans <- km_final$cluster
clusters_single <- cutree(hclust_single, k = 5)
clusters_avg <- cutree(hclust_avg, best_hier_k_dunn)

eig_combined <- cbind(eigen_vec_3, clusters_single, clusters_avg, clusters_kmeans)
head(eig_combined)
```

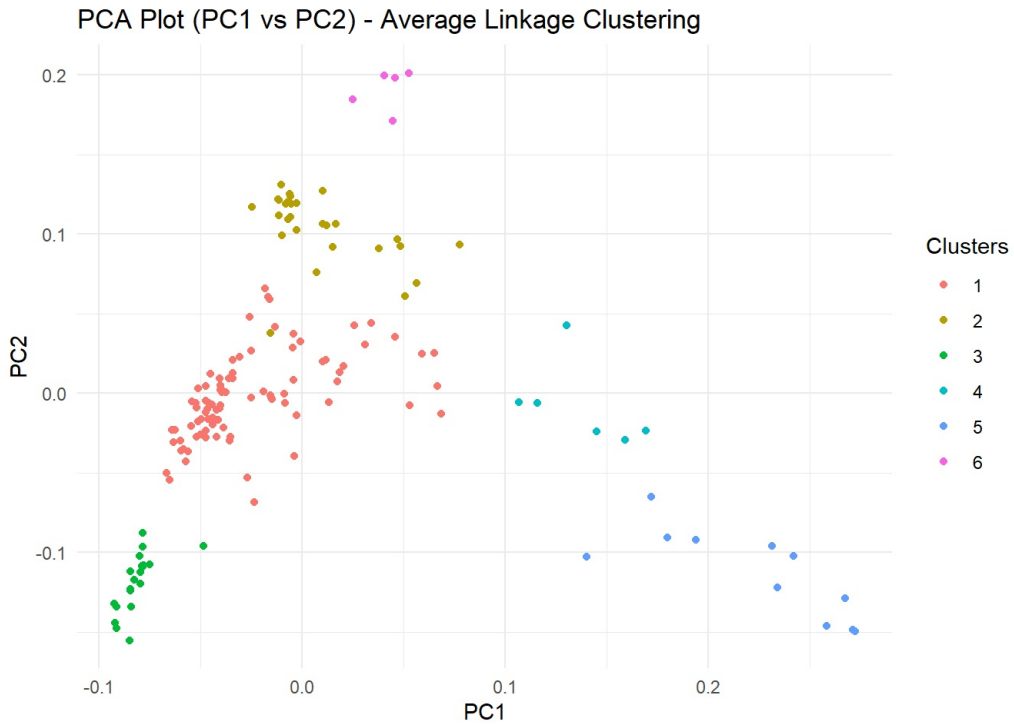
```
##           PC1          PC2          PC3 clusters_single clusters_avg
## 1  0.0257471  0.042520200  0.000671924               1            1
## 2 -0.0394316  0.000642535 -0.081346600               1            1
## 3 -0.0401049 -0.007438850 -0.082995400               1            1
## 4 -0.0156592  0.037883500  0.159538000               2            2
## 5 -0.0118682  0.121597000  0.052364500               1            2
## 6  0.0562310  0.069204100  0.062476200               1            2
##   clusters_kmeans
## 1               4
## 2               3
## 3               3
## 4               5
## 5               5
## 6               5
```

Visualize the clusters correposponding to the subpopulations that were produced from each clustering on the pca plots (PC1 vs PC2, PC1 vs PC3 and PC2 vs PC3) using ggplot2 and do not forget to color them.

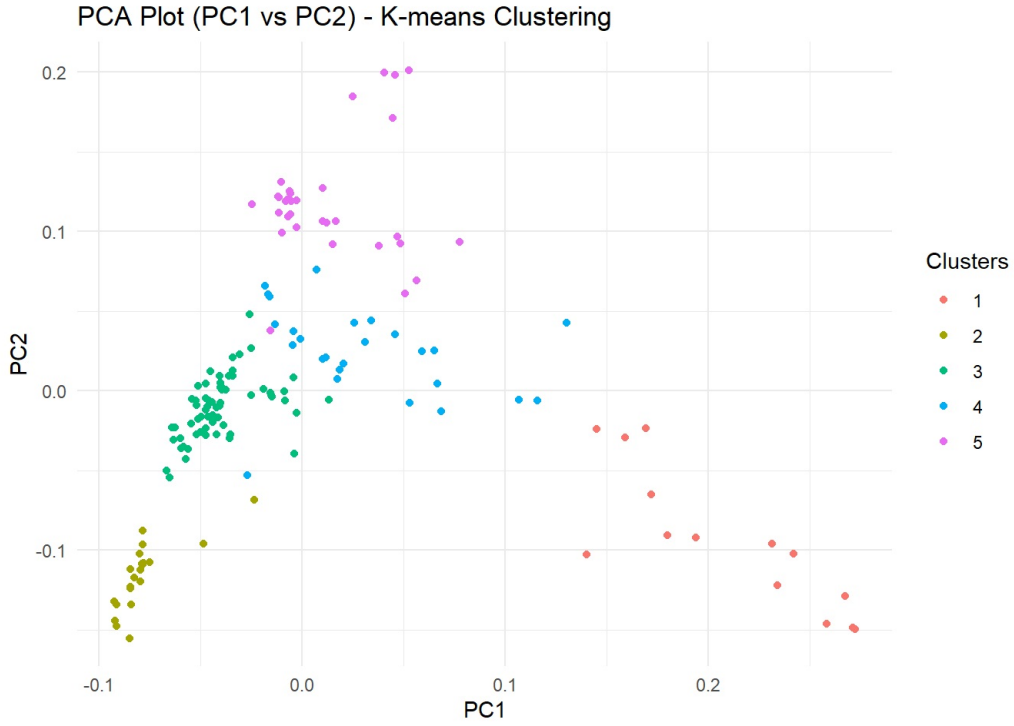
```
# Plot PCs plots with clusters colored
# PC1 vs PC2
ggplot(eig_combined, aes(x = PC1, y = PC2, color = factor(clusters_single))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC2) - Single Linkage Clustering", color = "Clusters") +
  theme_minimal()
```



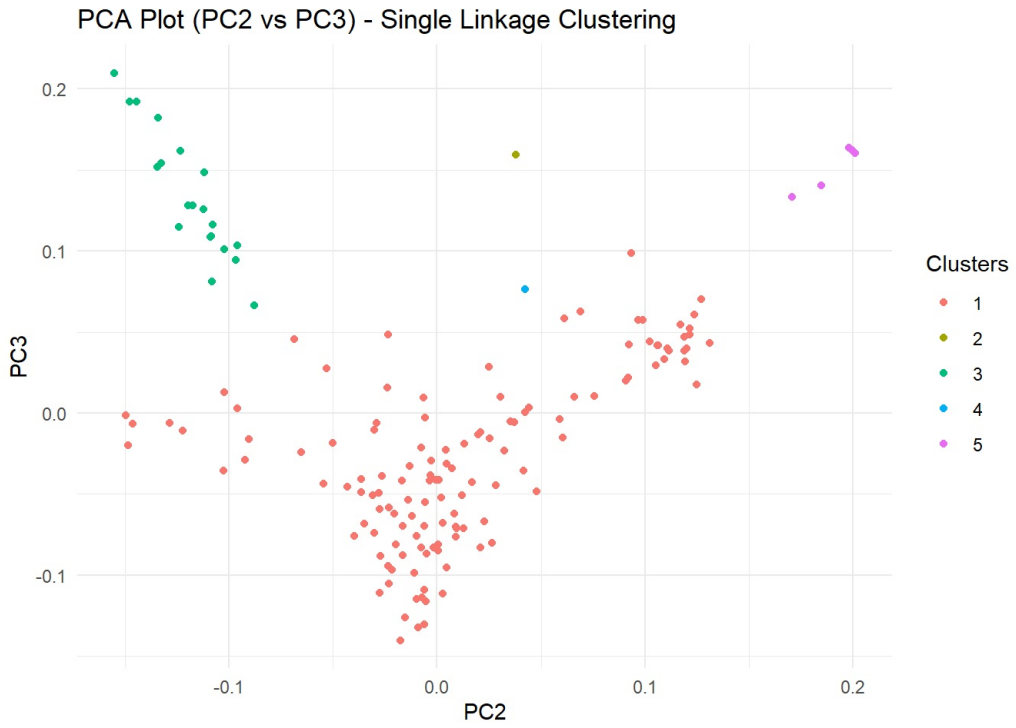

```
ggplot(eig_combined, aes(x = PC1, y = PC2, color = factor(clusters_avg))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC2) - Average Linkage Clustering", color = "Clusters") +
  theme_minimal()
```



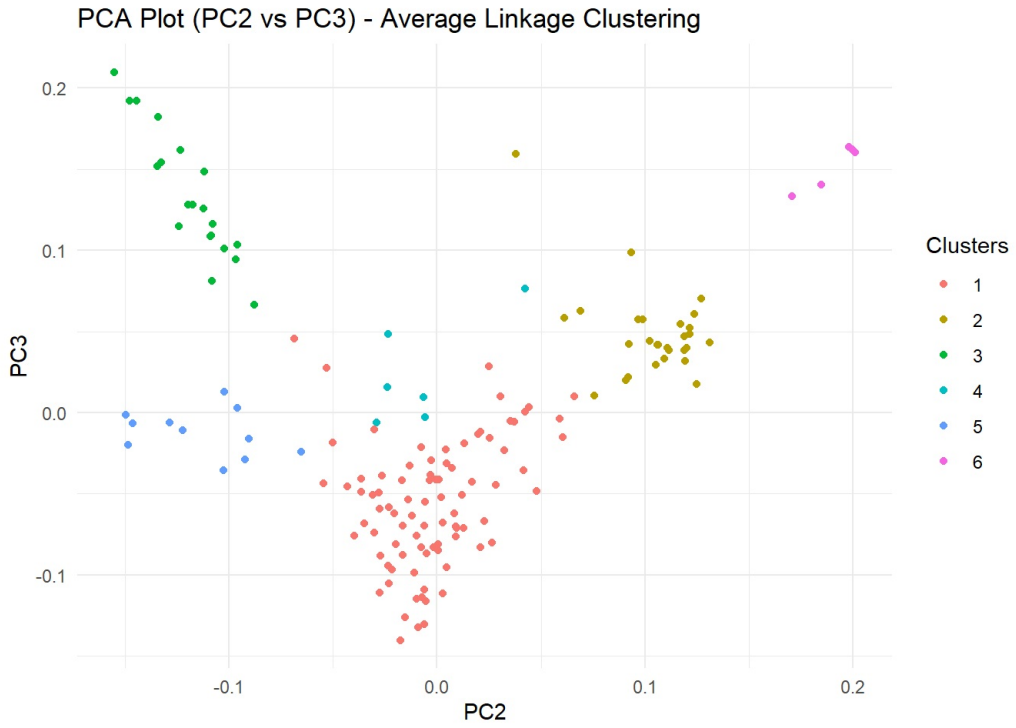

```
ggplot(eig_combined, aes(x = PC1, y = PC2, color = factor(clusters_kmeans))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC2) - K-means Clustering", color = "Clusters") +
  theme_minimal()
```



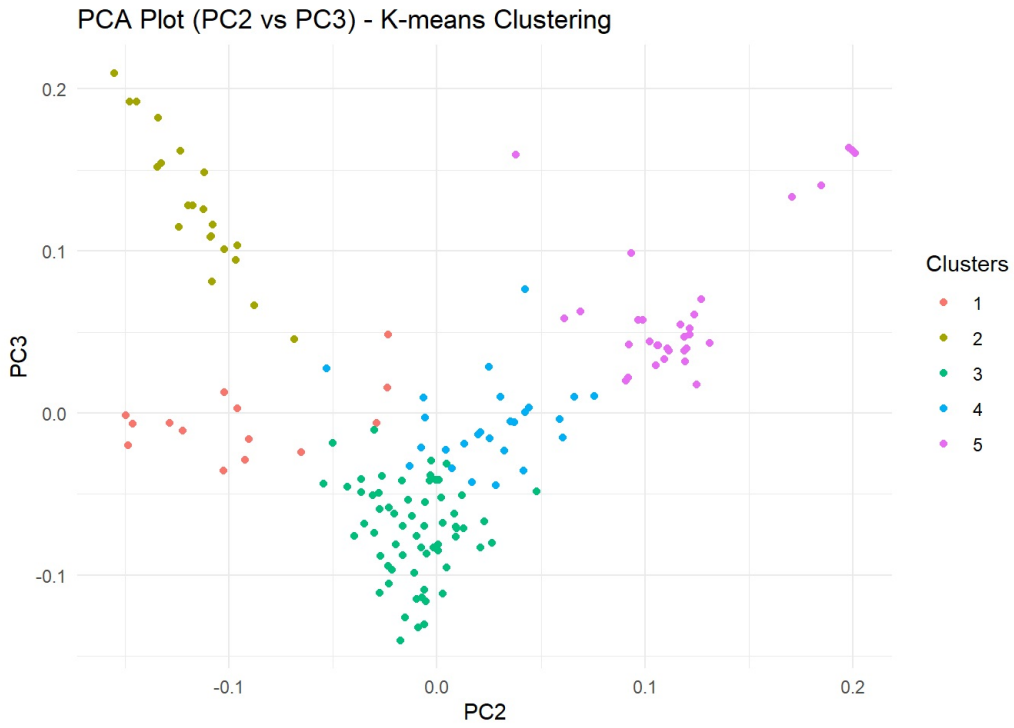

```
# PC2 vs PC3
ggplot(eig_combined, aes(x = PC2, y = PC3, color = factor(clusters_single))) +
  geom_point() +
  labs(title = "PCA Plot (PC2 vs PC3) - Single Linkage Clustering", color = "Clusters") +
  theme_minimal()
```



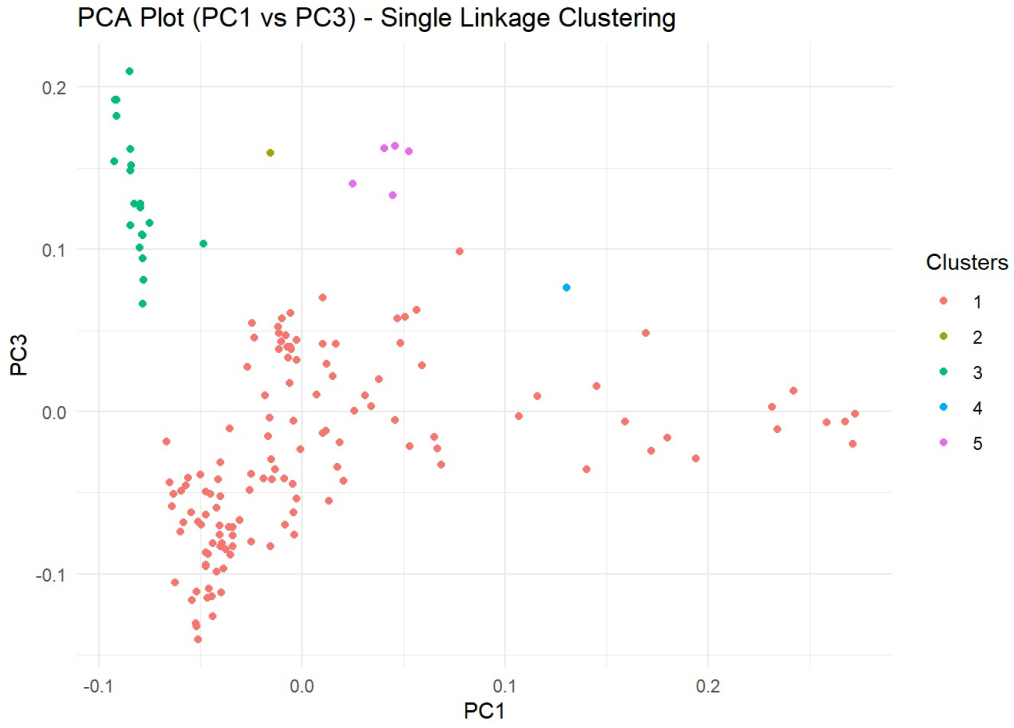

```
ggplot(eig_combined, aes(x = PC2, y = PC3, color = factor(clusters_avg))) +
  geom_point() +
  labs(title = "PCA Plot (PC2 vs PC3) - Average Linkage Clustering", color = "Clusters") +
  theme_minimal()
```



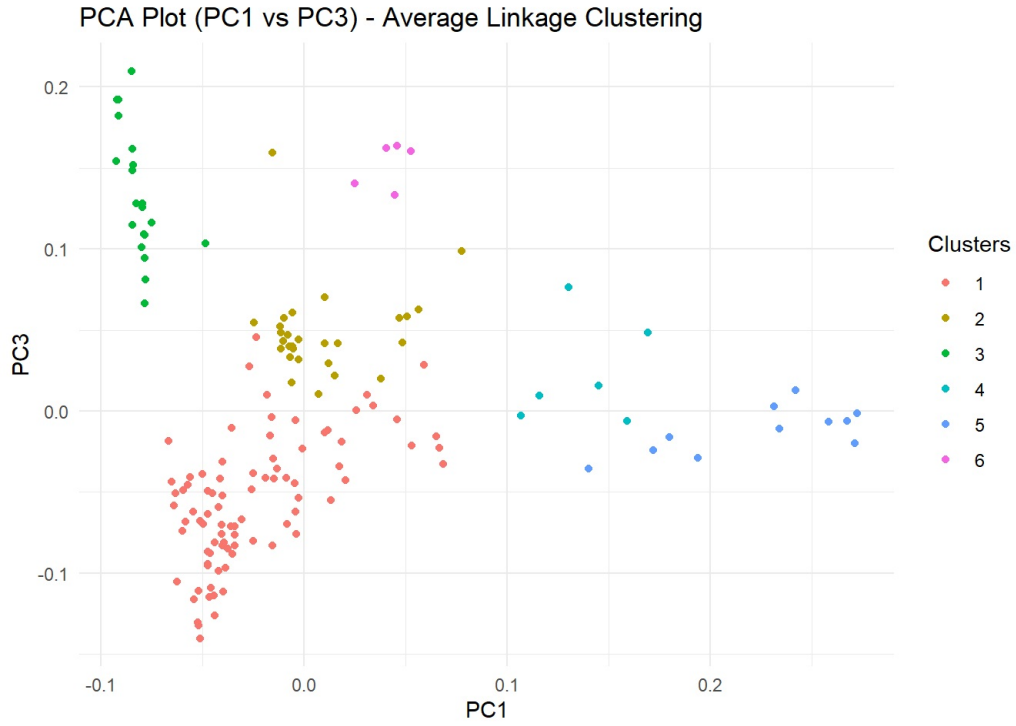

```
ggplot(eig_combined, aes(x = PC2, y = PC3, color = factor(clusters_kmeans))) +
  geom_point() +
  labs(title = "PCA Plot (PC2 vs PC3) - K-means Clustering", color = "Clusters") +
  theme_minimal()
```



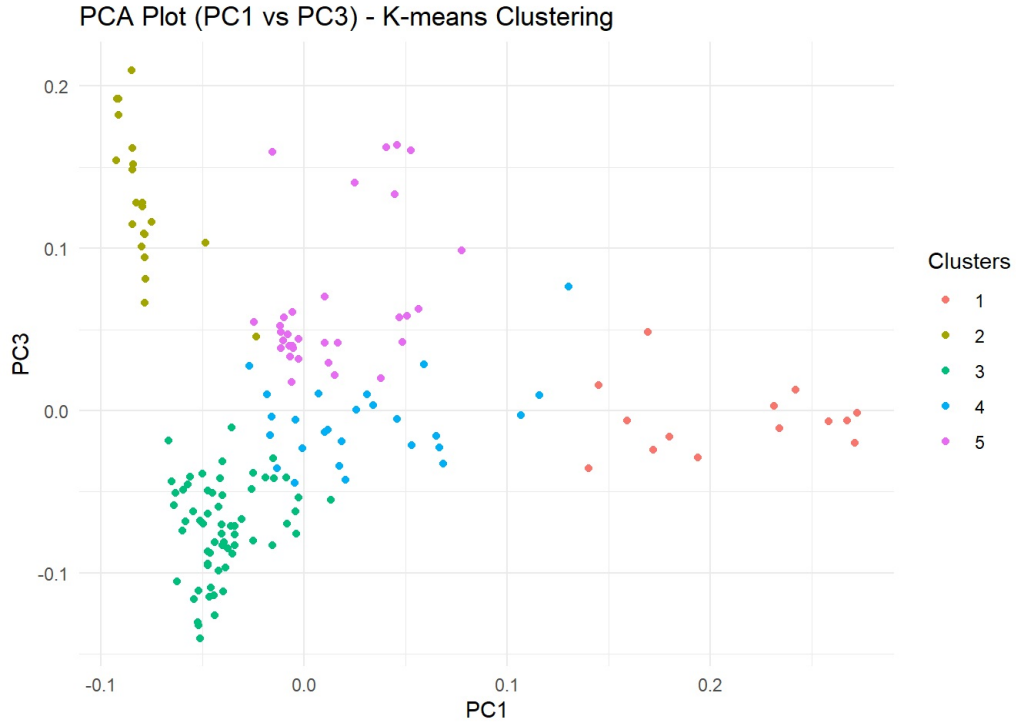

```
# PC1 vs PC3
ggplot(eig_combined, aes(x = PC1, y = PC3, color = factor(clusters_single))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC3) - Single Linkage Clustering", color = "Clusters") +
  theme_minimal()
```




```
ggplot(eig_combined, aes(x = PC1, y = PC3, color = factor(clusters_avg))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC3) - Average Linkage Clustering", color = "Clusters") +
  theme_minimal()
```

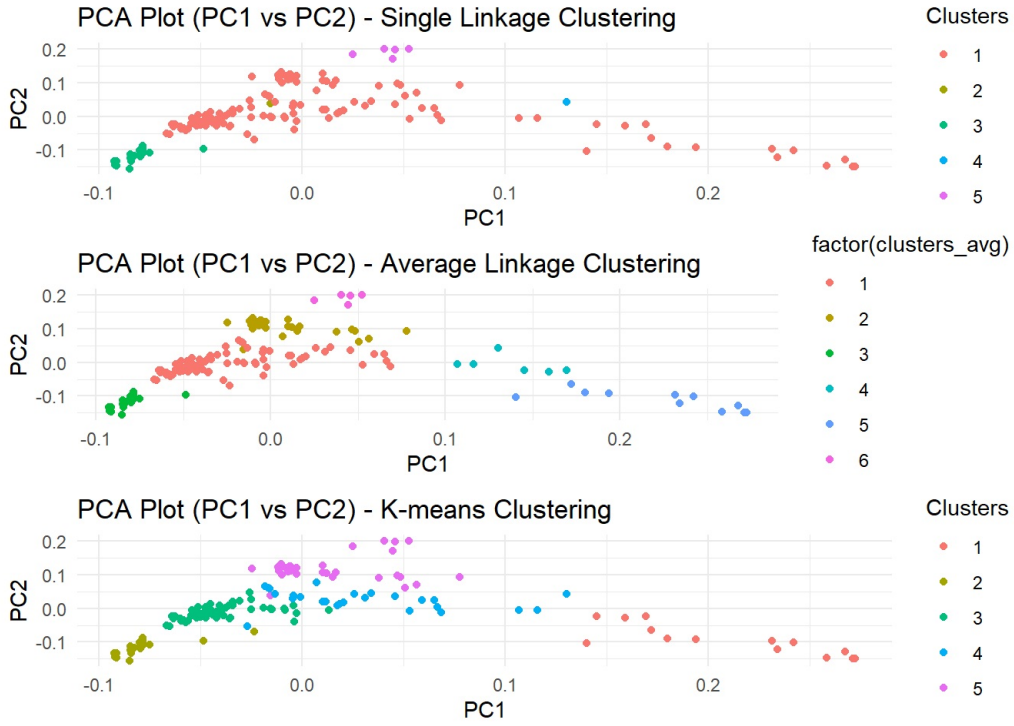



```
ggplot(eig_combined, aes(x = PC1, y = PC3, color = factor(clusters_kmeans))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC3) - K-means Clustering", color = "Clusters") +
  theme_minimal()
```

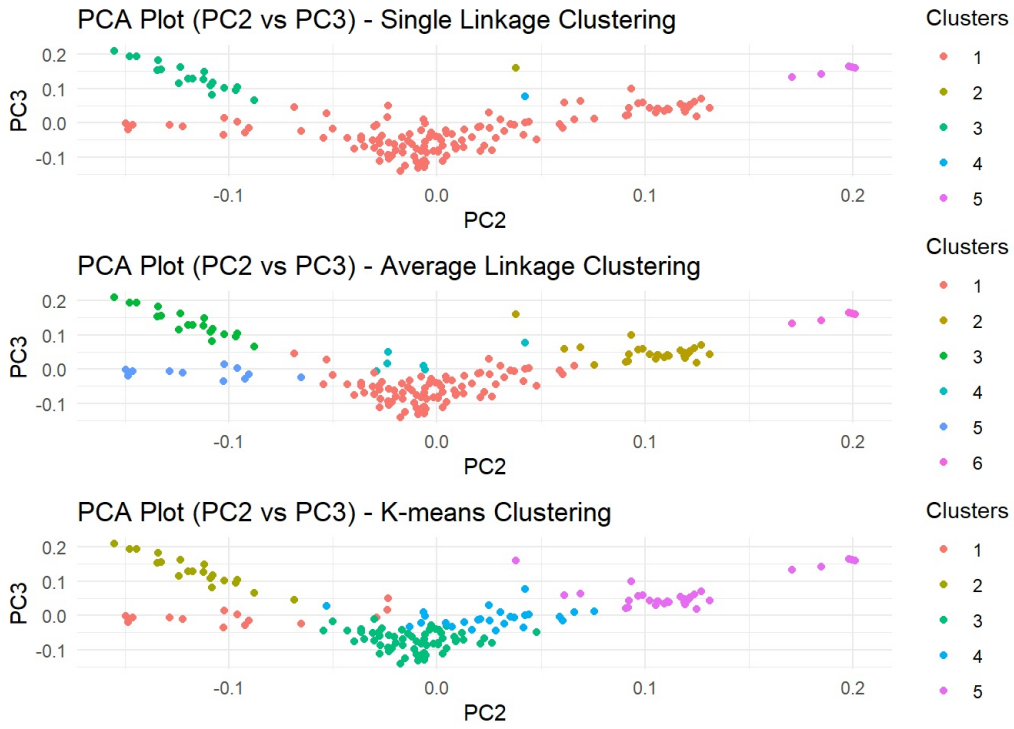



Create a side-by-side comparison of the clusters formed by k-means, single linkage, and average linkage methods. Use gridExtra's grid.arrange function to plot multiple clustering results side by side for easy comparison. Make sure each plot uses different colors for each cluster to aid in comparison

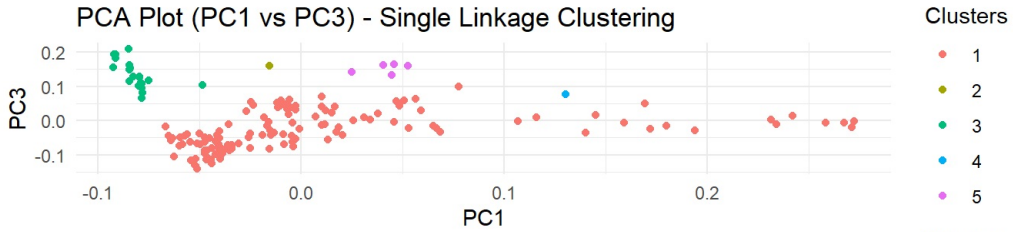
```r
# Plot PCs plots with clusters colored
# PC1 vs PC2
pc1_2_single <- ggplot(eig_combined, aes(x = PC1, y = PC2, color = factor(clusters_single))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC2) - Single Linkage Clustering", color = "Clusters") +
  theme_minimal()

pc1_2_avg <- ggplot(eig_combined, aes(x = PC1, y = PC2, color = factor(clusters_avg))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC2) - Average Linkage Clustering") +
  theme_minimal()

pc1_2_kmeans <- ggplot(eig_combined, aes(x = PC1, y = PC2, color = factor(clusters_kmeans))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC2) - K-means Clustering", color = "Clusters") +
  theme_minimal()

# PC2 vs PC3
pc2_3_single <- ggplot(eig_combined, aes(x = PC2, y = PC3, color = factor(clusters_single))) +
  geom_point() +
  labs(title = "PCA Plot (PC2 vs PC3) - Single Linkage Clustering", color = "Clusters") +
  theme_minimal()

pc2_3_avg <- ggplot(eig_combined, aes(x = PC2, y = PC3, color = factor(clusters_avg))) +
  geom_point() +
  labs(title = "PCA Plot (PC2 vs PC3) - Average Linkage Clustering", color = "Clusters") +
  theme_minimal()

pc2_3_kmeans <- ggplot(eig_combined, aes(x = PC2, y = PC3, color = factor(clusters_kmeans))) +
  geom_point() +
  labs(title = "PCA Plot (PC2 vs PC3) - K-means Clustering", color = "Clusters") +
  theme_minimal()

# PC1 vs PC3
pc1_3_single <- ggplot(eig_combined, aes(x = PC1, y = PC3, color = factor(clusters_single))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC3) - Single Linkage Clustering", color = "Clusters") +
  theme_minimal()

pc1_3_avg <- ggplot(eig_combined, aes(x = PC1, y = PC3, color = factor(clusters_avg))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC3) - Average Linkage Clustering", color = "Clusters") +
  theme_minimal()

pc1_3_kmeans <- ggplot(eig_combined, aes(x = PC1, y = PC3, color = factor(clusters_kmeans))) +
  geom_point() +
  labs(title = "PCA Plot (PC1 vs PC3) - K-means Clustering", color = "Clusters") +
  theme_minimal()

# Arrange plots side by side for comparison
grid.arrange(pc1_2_single, pc1_2_avg, pc1_2_kmeans, nrow = 3, ncol = 1)
```

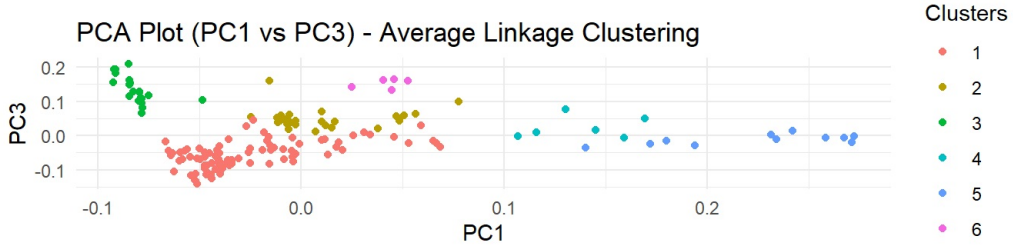



```
grid.arrange(pc2_3_single, pc2_3_avg, pc2_3_kmeans, nrow = 3, ncol = 1)
```




```
grid.arrange(pc1_3_single, pc1_3_avg, pc1_3_kmeans, nrow = 3, ncol = 1)
```





## Comments on comparison:

For PC1 vs PC2, the hierarichal clustering using average clustering produced the best result as it produced cohesive clusters with distinguishable boundaries.

For PC2 vs PC3, I think k-means did great as it splited the data into groups of related points. The points are somehow near to each other.

For PC1 vs PC3, the hierarichal clustering using average clustering was able to cluster the points into somehow good and coherent clusters.

## Write your comments on how the data quality control affected the dataset

Quality control on data is crucial as it helps in multiple things making the quality of clustering and analysis more valuable.

For example, quality control helps in removing of outliers. This enhances the robustness of the datasetas outliers can skew statistical analysis and clustering results, leading to erroneous interpretations.

Another benefit is the normalization and standardization of the data ensuring that all variables contribute equally to the clustering process, preventing variables with larger scales from dominating the analysis so that the clustering algorithm can effectively capture the underlying patterns in the data without being influenced by variable magnitudes.

There are many more benefits, such as handling the missing values through imputation or though deciding to remove them from the data.

Overall, data quality control measures contribute significantly to the reliability, accuracy, and interpretability of clustering results. By addressing data quality issues, we can ensure that the analysis will yield meaningful insights.