Stat 437 Project 2

Nam Jun Lee (11606459)

Contents

1 Introduction

1.1 Using Methods

This project describes the results for each two datasets using (1) Principal Component Analysis, (2) Sparse Principal Component Analysis.

(1) PCA: It is a method of literally finding the main component of distributed data by reducing high-dimensional data to low-dimensional data.

(2) SPCA: It is a method of changing a typical PCA with the aim of improving PC interpret ability from the perspective of dominant features along the direction by generating sparse loading vectors with most entries zero.

1.2 Data Information

(1) The data files are "labels.csv" that contains the cancer type for each sample, and "data.csv" that contains the "gene expression profile" (i.e., expression measurements of a set of genes) for each sample. Here each sample is for a subject and is stored in a row of "data.csv". In fact, the data set contains the gene expression profiles for 801 subjects, each with a cancer type, where each gene expression profile contains the gene expressions for the same set of 20531 genes. The cancer types are: BRCA, KIRC, COAD, LUAD and PRAD. In both files "labels.csv" and "data.csv", each row name records which sample a label or observation is for.

(2) The Spam column of "SPAM.csv" shows the actual status of each email and contains the measurements of features. Most of the attributes indicate whether a particular word or character was frequently occuring in the e-mail. There are a total of 4601 rows and 57 columns, each represented as a function row in the .csv file, and the feature can be considered a "predictor". In addition, the first 1813 rows of the dataset (i.e., observations) are for spam emails, and the rest are for non-spam emails.

1.3 Two Tasks

In the case of **task A**, each `PCA` and `SPCA` analysis are performed for the gene profile dataset related to cancer type. Prior to the analysis, only the necessary information is preprocessed. And follow three confirmation procedures. (1) Using the dataset, check whether each feature accounts for a significant portion of the variation in gene expression. (2) Through each `PCA` and `SPCA` analysis, it is confirmed whether there is a specific gene expression pattern for cancer types. (3) For each cancer type, it is confirmed whether the five main components obtained through analysis explain the main proportion of the volatility of the dataset. After that, after each `PCA` and `SPCA` are implemented to derive detailed results, the results of the two analysis techniques are compared and interpreted.

In the case of **task B**, `PCA` analysis is performed on the spam dataset. Prior to the analysis, only necessary information is preprocessed (Highly correlated features with missing values). And after setting up a training set and a test set in these datasets, the results are interpreted by performing a `PCA` analysis technique using educational data.

2 Methods & Results

2.1 Task A. Analysis of gene expression data

2.1.1 Data processing

Please use `set.seed(123)` for random sampling via the command `sample`.

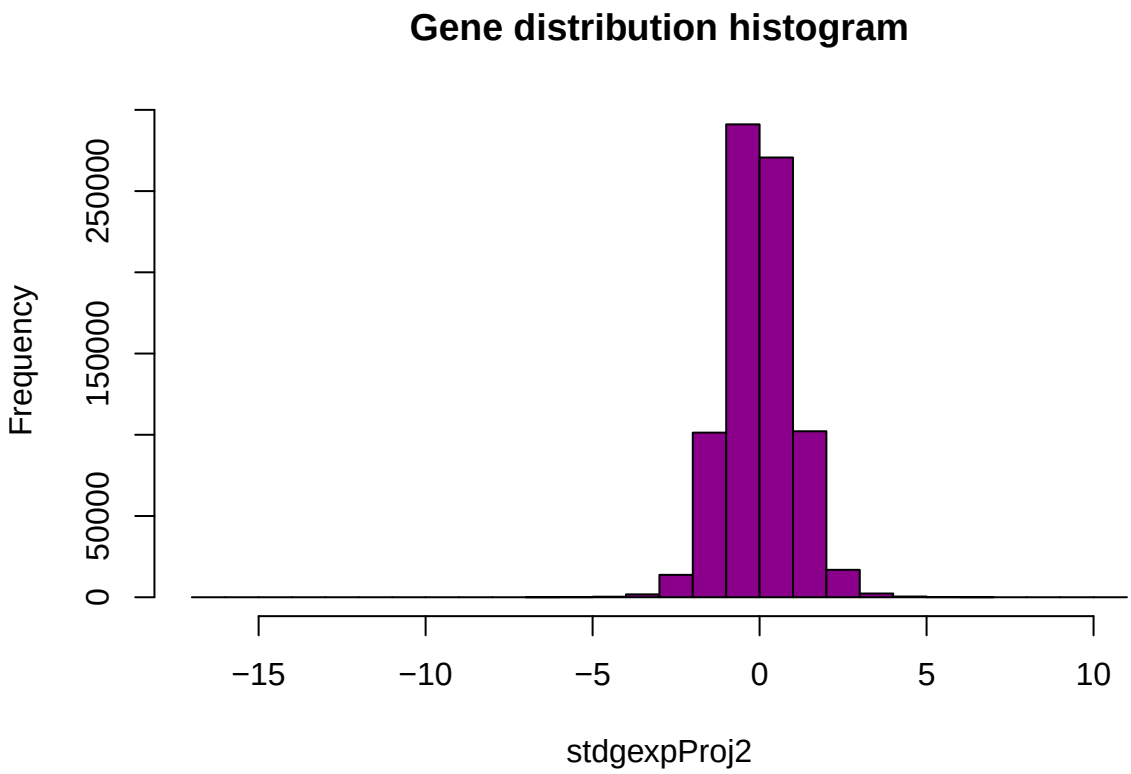
- Filter out genes (from "data.csv") whose expressions are zero for at least 300 subjects, and save the filtered data as R object "gexp2".
- Use the command `sample` to randomly select 1000 genes and their expressions from "gexp2", and save the resulting data as R object "gexp3".
- Use the command `scale` to standardize the gene expressions for each gene in "gexp3". Save the standardized data as R object "stdgexpProj2".

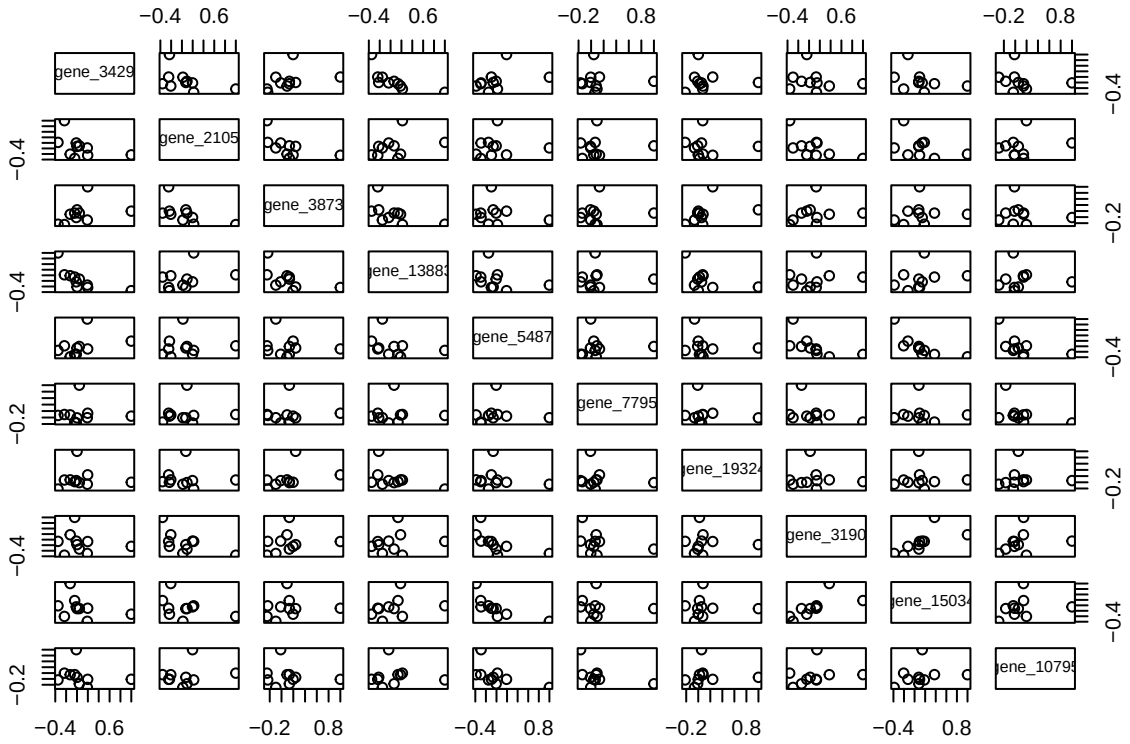
```
## [1]  801 1000
```

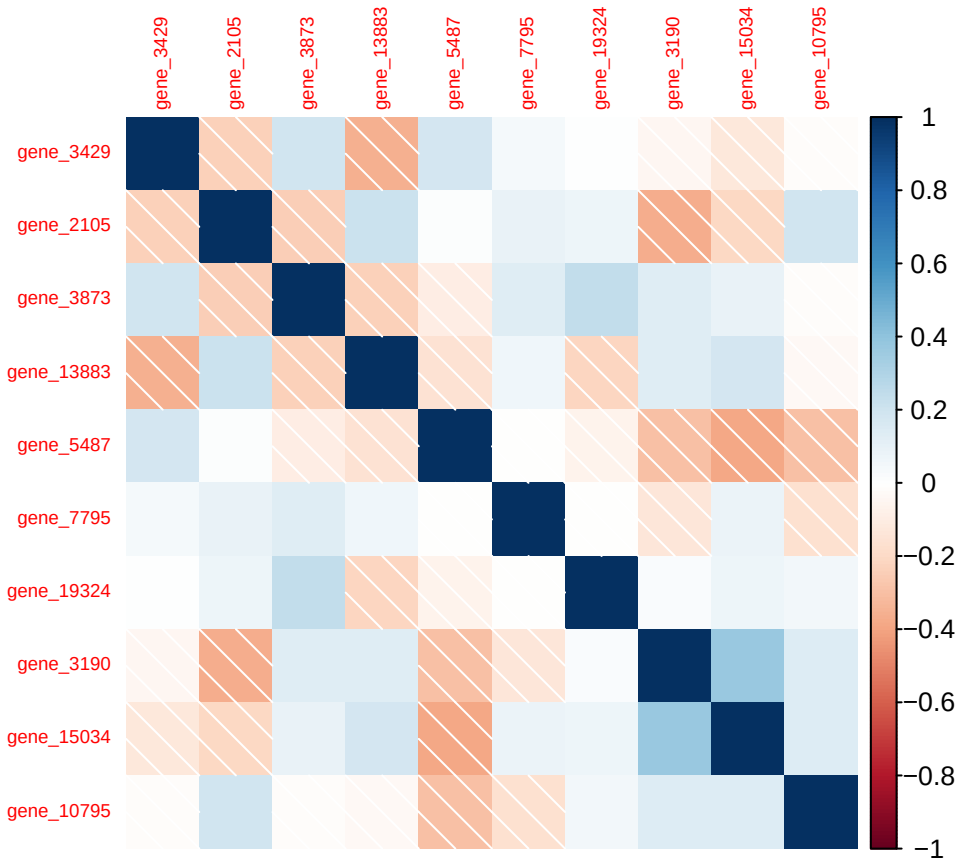
As a result of checking the rows and columns of the preprocessed data, it can be seen that the preprocessing is well done with 1000 columns and 801 rows.

2.1.2 1.a

Are there genes for which linear combinations of their expressions explain a significant proportion of the variation of gene expressions in the data set? Note that each gene corresponds to a feature, and a principal component based on data version is a linear combination of the expression measurements for several genes.


Gene distribution histogram
Frequency
0
50000
150000
250000
−15
−10
−5
0
5
10
stdgexpProj2

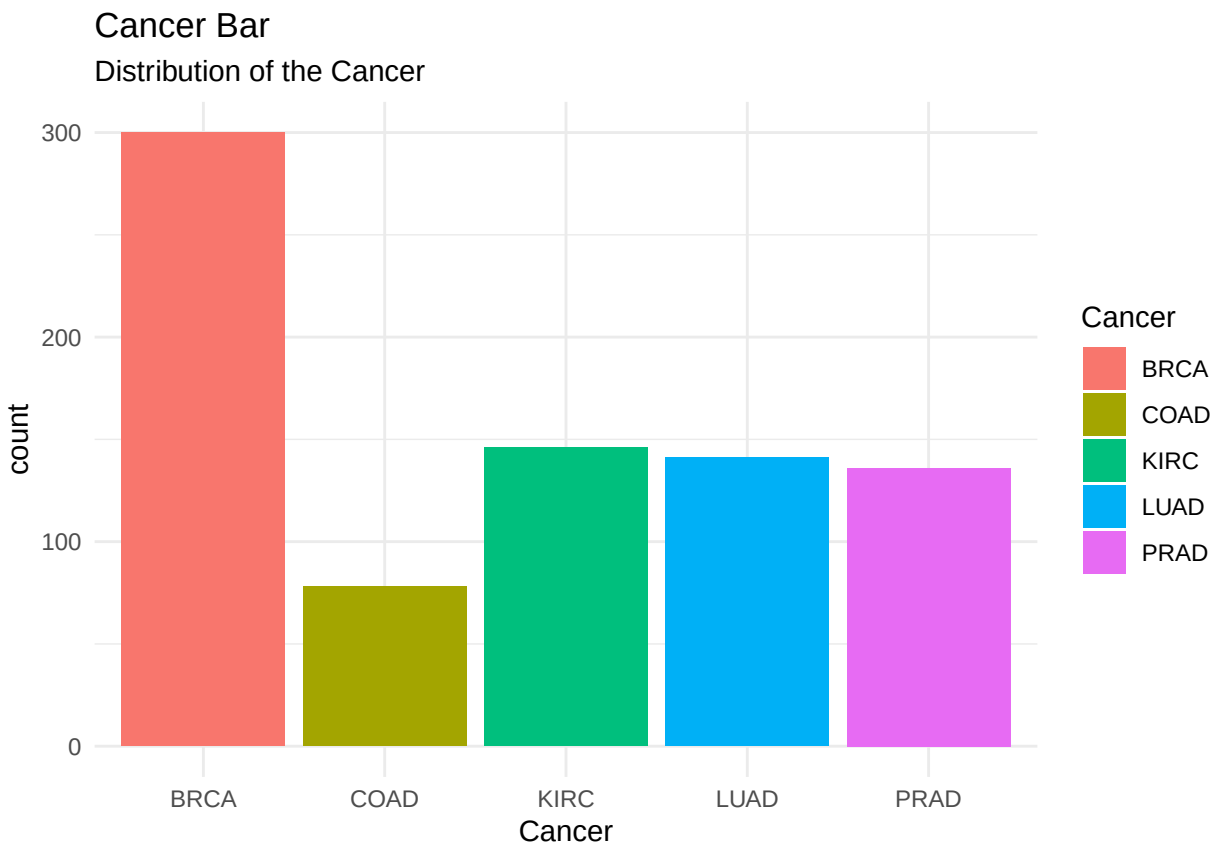

gene_3429
gene_2105
gene_3873
gene_5487
gene_7795
gene_3190
−0.4 0.6
−0.2 0.8
−0.4
−0.2

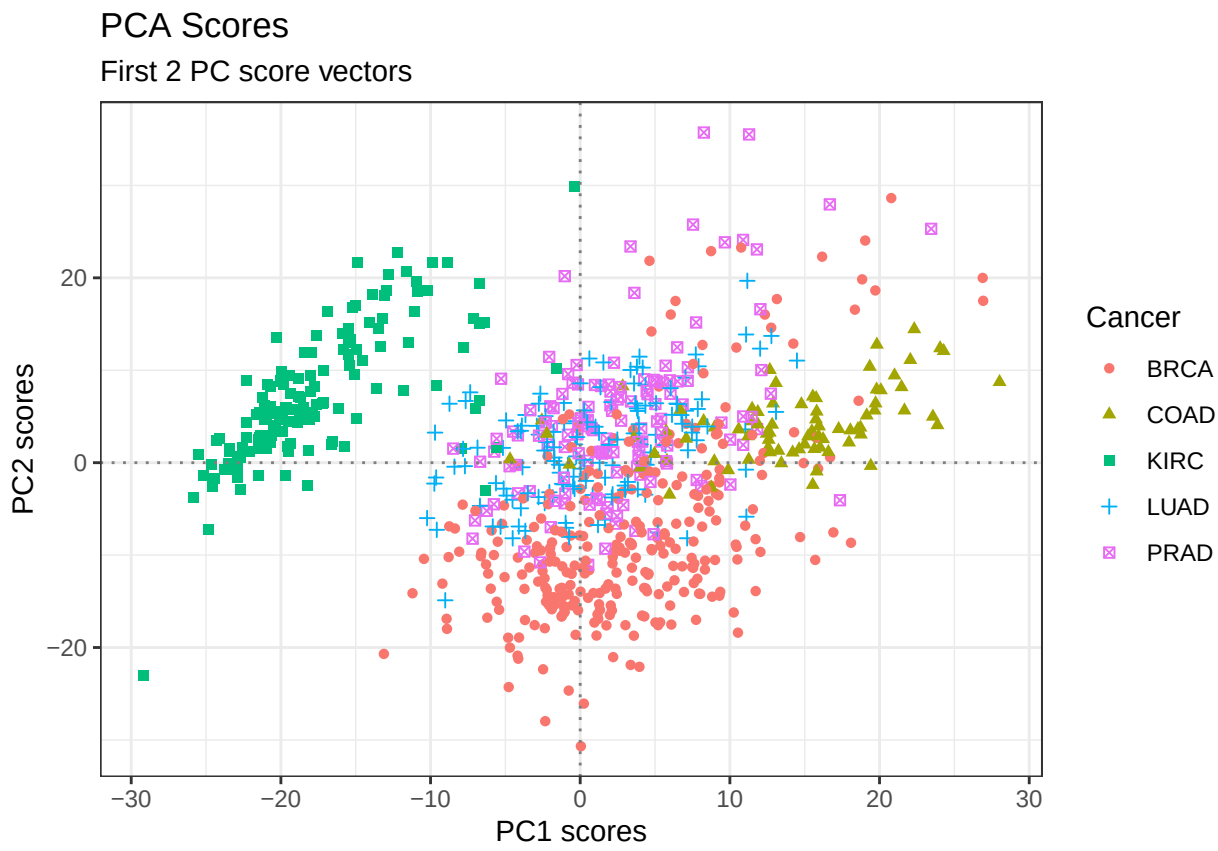


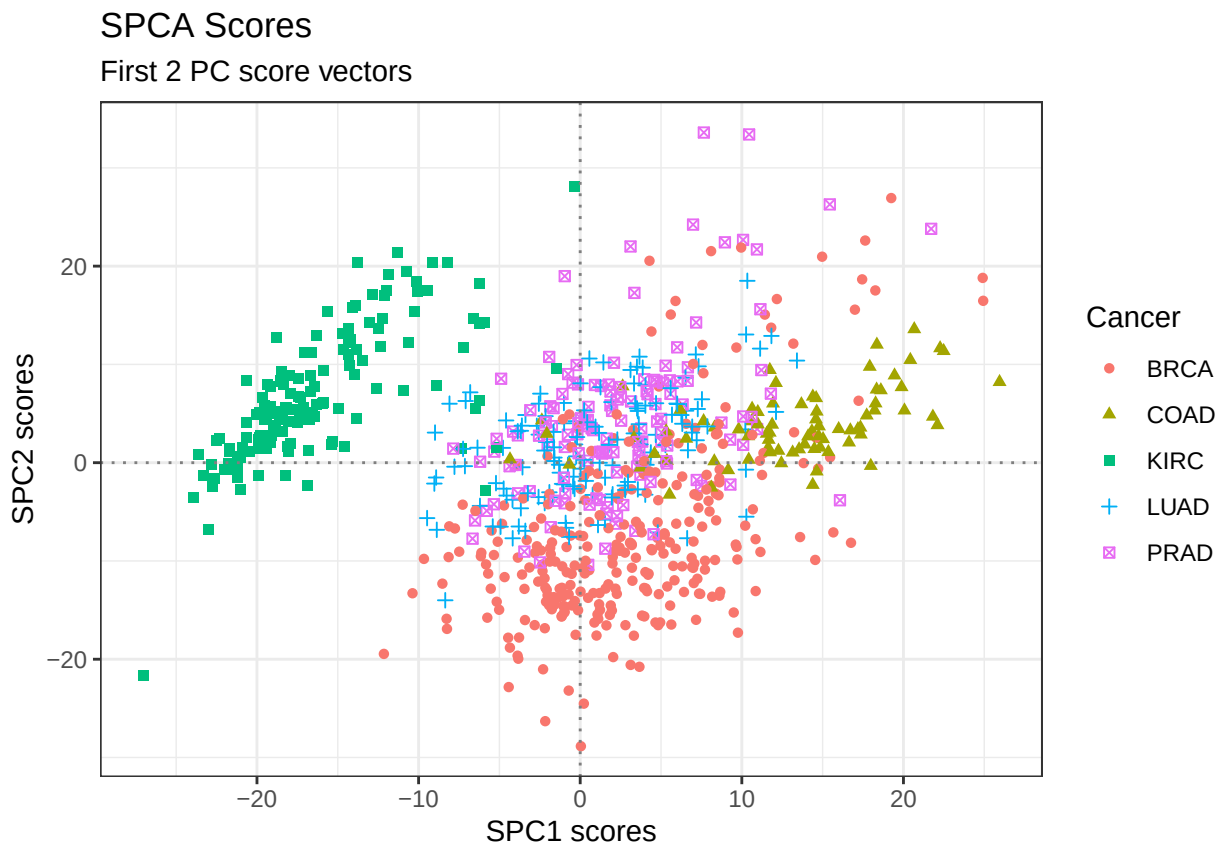
As a result of checking the variation in gene expression through the histogram, it can be confirmed that it follows a normal distribution. In addition, the correlation graph of genes shows that there is little correlation between each gene expression feature, and that **there is no specific gene that explains a significant portion of the linear combination of gene expression variations**.

2.1.3 1.b

Ideally, a type of cancer should have its "signature", i.e., a pattern in the gene expressions that is specific to this cancer type. From the "labels.csv", you will know which expression measurements belong to which cancer type. Identify the signature of each cancer type (if any) and visualize it. For this, you need to be creative and should try both PCA and Sparse PCA.


Cancer Bar
Distribution of the Cancer
300
200
100
0
count
BRCA
COAD
KIRC
LUAD
PRAD
Cancer
Cancer
BRCA
COAD
KIRC
LUAD
PRAD


PCA Scores
First 2 PC score vectors
PC2 scores
PC1 scores
20
0
−20
−30
−20
−10
0
10
20
30
Cancer
BRCA
COAD
KIRC
LUAD
PRAD



First, checking the frequency of gene expressions contained in each cancer type, it can be seen that `BRCA` is the most common and `COAD` is the least distributed.
As a result of performing each PCA and SPCA to find a gene expression pattern specific to each cancer type, it can be seen that the characteristic of `KIRC` is that the negative score of PC1 is strong and the score of PC2 is slightly higher than medium. The characteristic of `COAD` is that PC1 has a strong positive score and PC2 has a moderate score. The characteristic of `BRCA` is that the score of PC1 is slightly higher than the middle and the score of PC2 is slightly lower than the middle. Most of the remaining `LUAD` and `PRAD` are moderately distributed in the center and have no strong scores on either axis. In addition, it can be seen that `KIRC` and `COAD` were the best separated, and the remaining three classes were not well separated.

2.1.4 1.c

There are 5 cancer types. Would 5 principal components, obtained either from PCA or Sparse PCA, explain a dominant proportion of variability in the data set, and serve as the signatures of the 5 cancer types? Note that the same set of genes were measured for each cancer type.

```
## [1] 11.50546 21.52057 30.02800 35.67928 40.01737
```

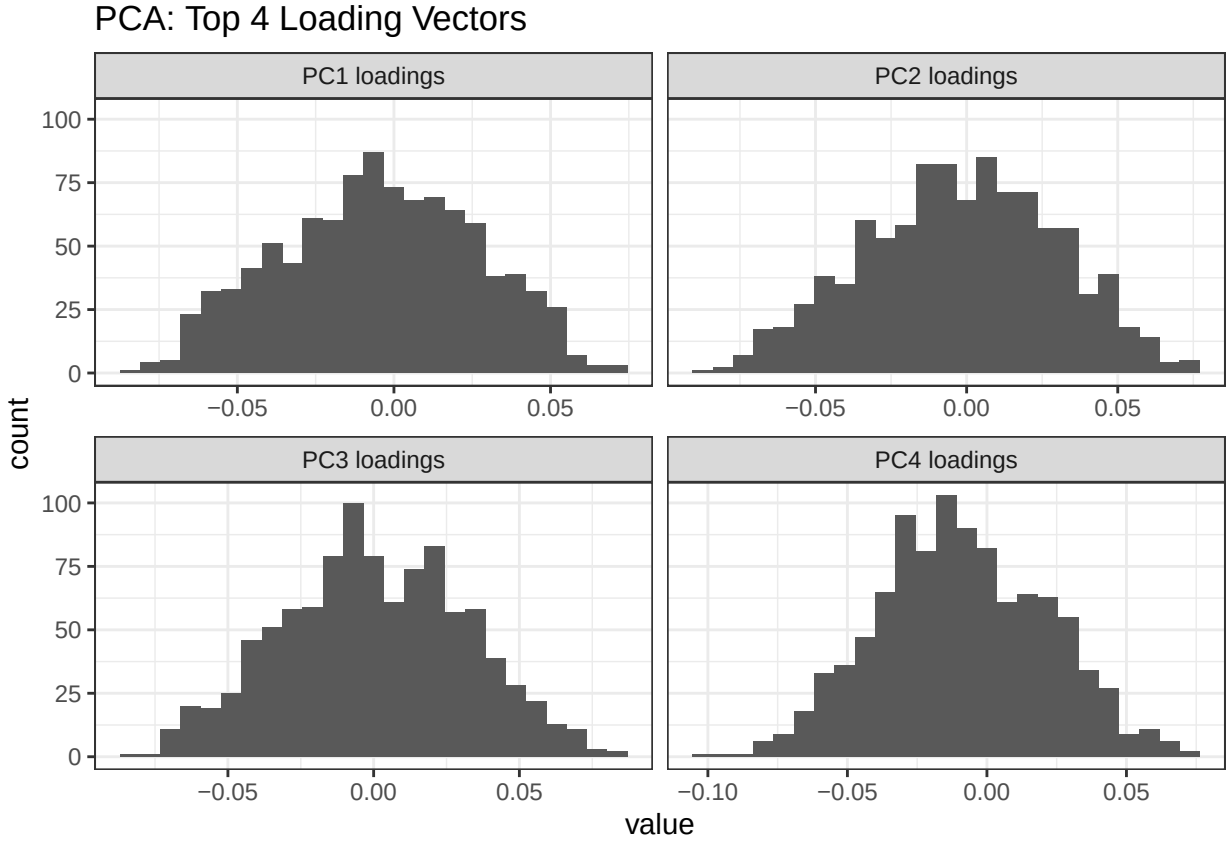
```
## [1] 0.09866376 0.18723858 0.26277867 0.31282473 0.35144122
```

To determine if the five main components obtained from `PCA` or `SPCA` account for the main comparison of data set volatility, the cumulative fluctuation rate described by `PCA` is **40.17 %** and SPCA

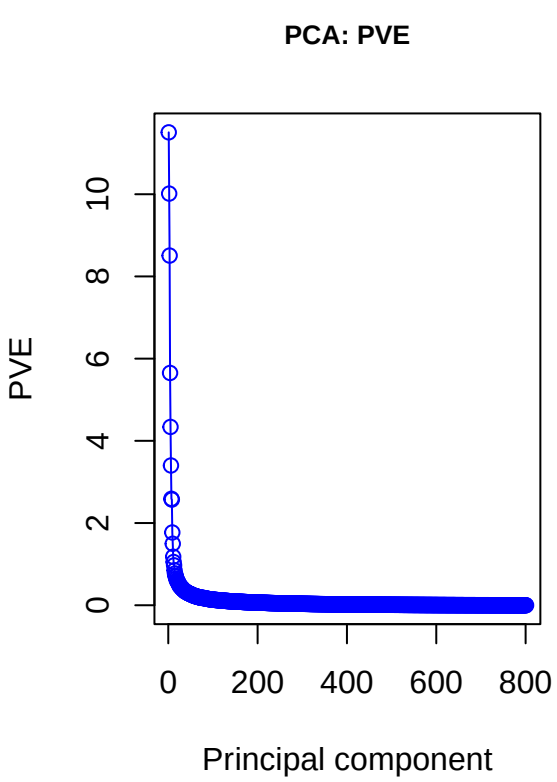
adjusted CPVE is **0.35144122**, this means cumulative fluctuation rate described by `SPCA` is **35.14 %**, **which does not explain the signature role of the five cancer types**.

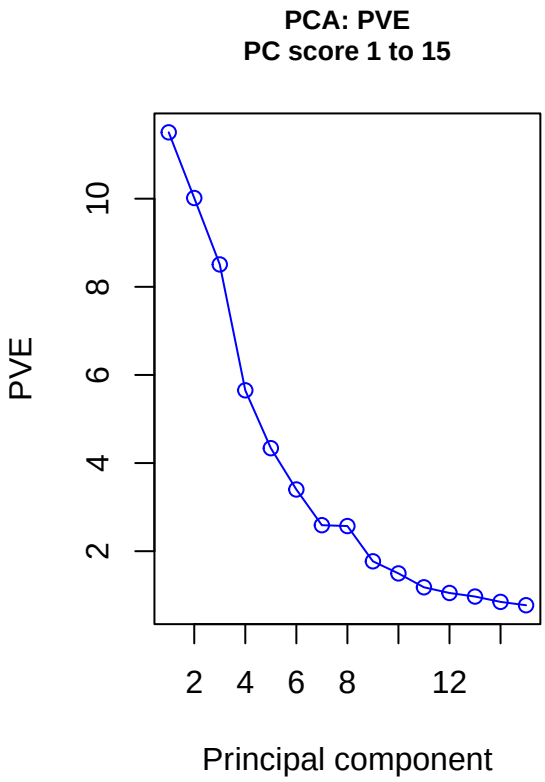
2.1.5 2.a

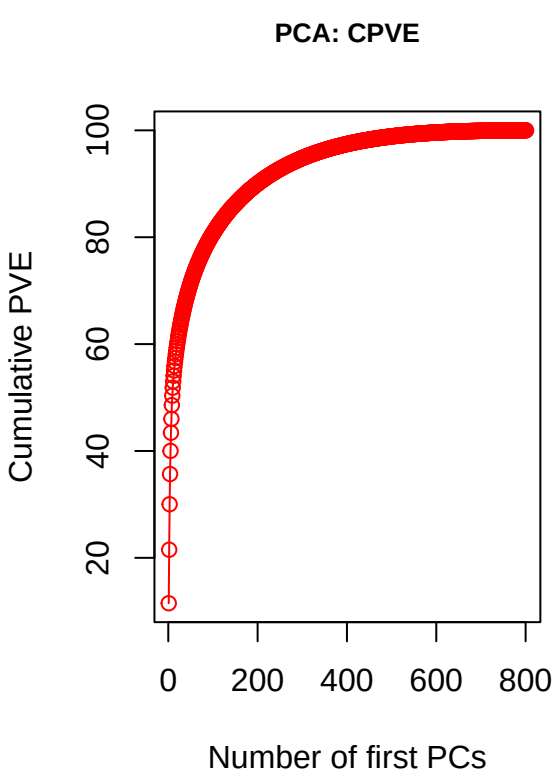
Apply PCA, determine the number of principal components, provide visualizations of low-dimensional structures, and report your findings. Note that you need to use "labels.csv" for the task of discoverying patterns such as if different cancer types have distinct transformed gene expressions (that are represented by principal components). For PCA or Sparse PCA, low-dimensional structures are usually represented by the linear space spanned by some principal components.

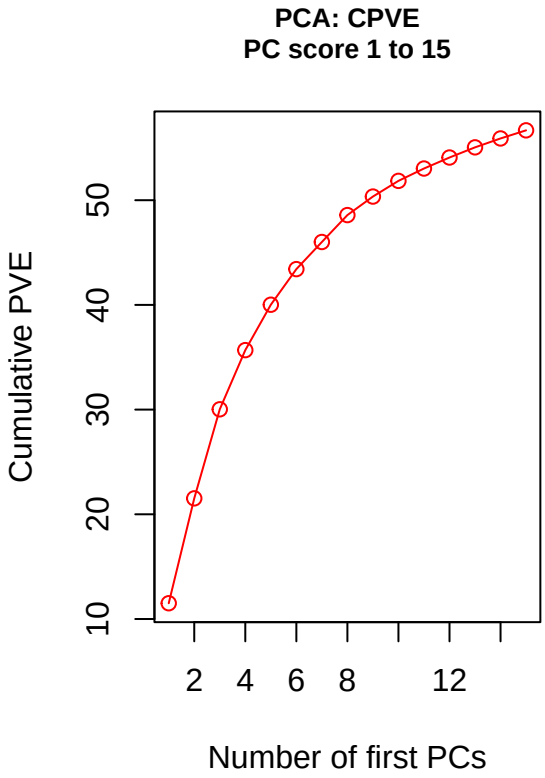


Through the top four loading vectors of PCA, each load vector has many nonzero entries and does not appear to have a dominant feature in any direction.

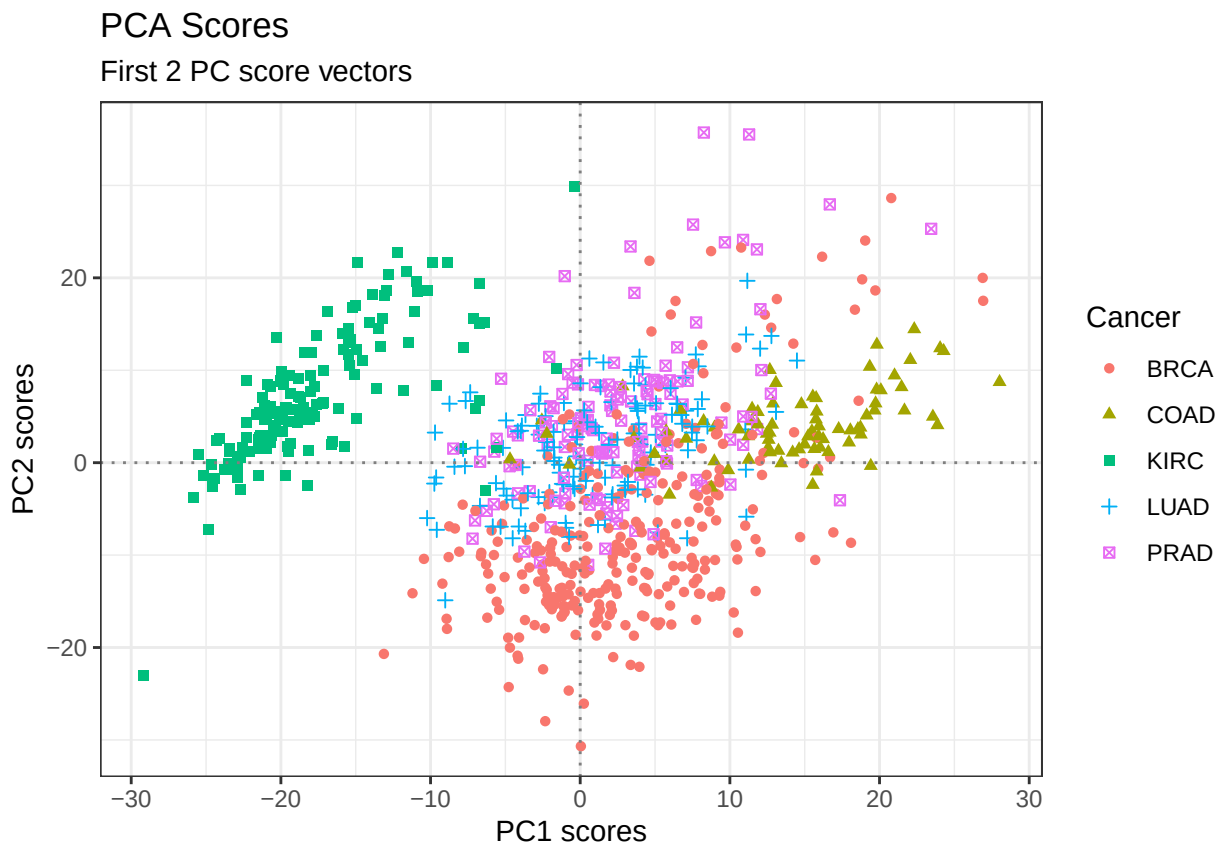

PCA: PVE
PVE
0
2
4
6
8
10
0
200
400
600
800
Principal component


PCA: PVE
PC score 1 to 15
PVE
2
4
6
8
10
2
4
6
8
12
Principal component

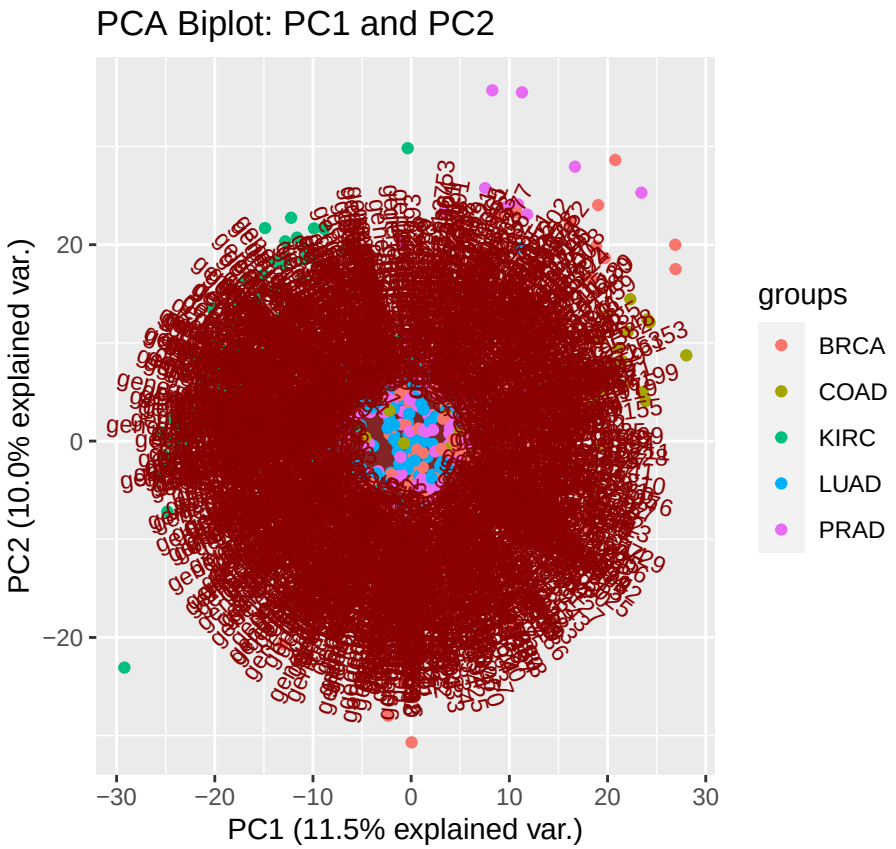




When checking the ratio of variance and the cumulative variance ratio of PCA, the ratio of variance cannot be accurately identified due to 801 many features. Therefore, as a result of examining the top 15 variance ratios to check more accurately, it is better to use up to seven main components because the slope changes rapidly at seven points. It can be seen that the variance of data can be confirmed from a relatively small number of components.



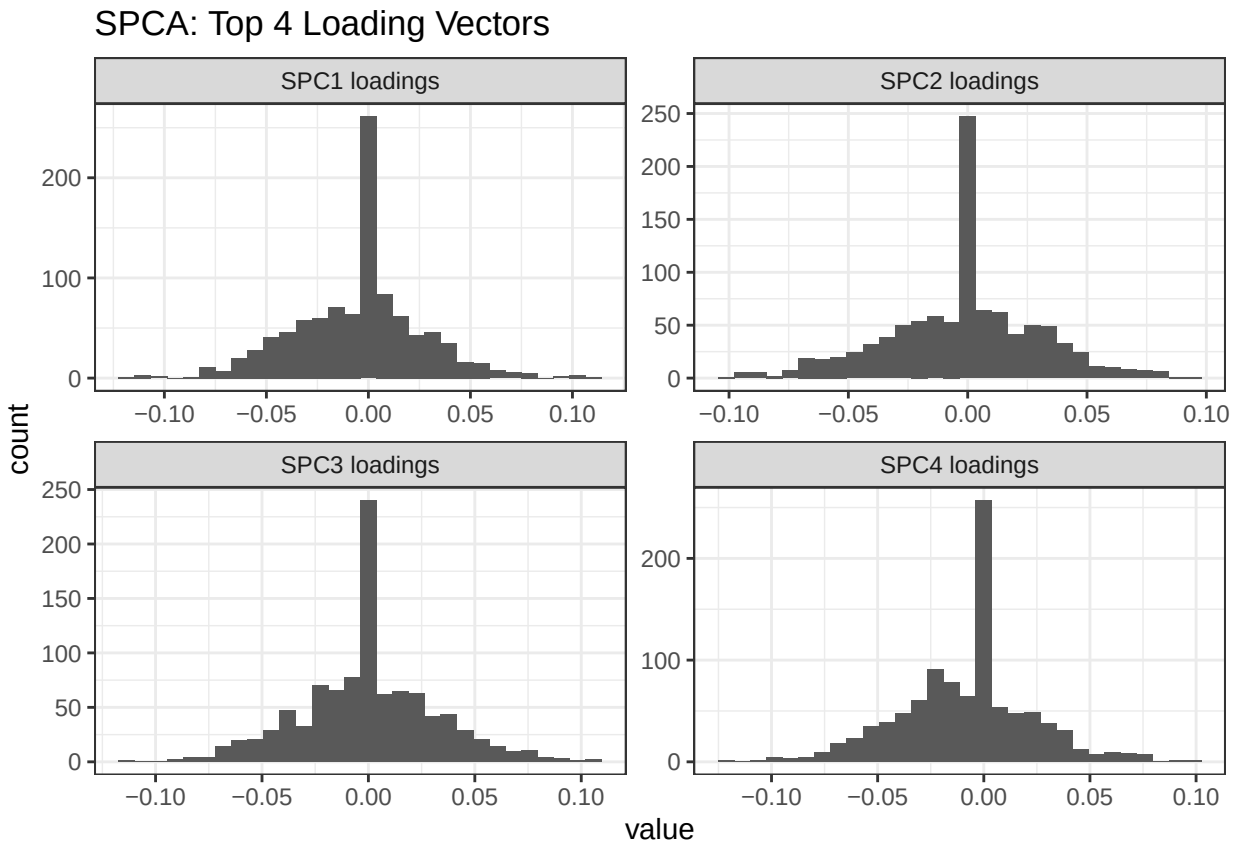
Visualization using the first two PC score vectors shows that `KIRC` and `COAD` are relatively well classified, and the remaining three classes are quite overlapping, making it difficult to distinguish a particular signature.



As a result of visualization to check the biplot of the data, the exact biplot cannot be confirmed due to too many components.

2.1.6 2.b

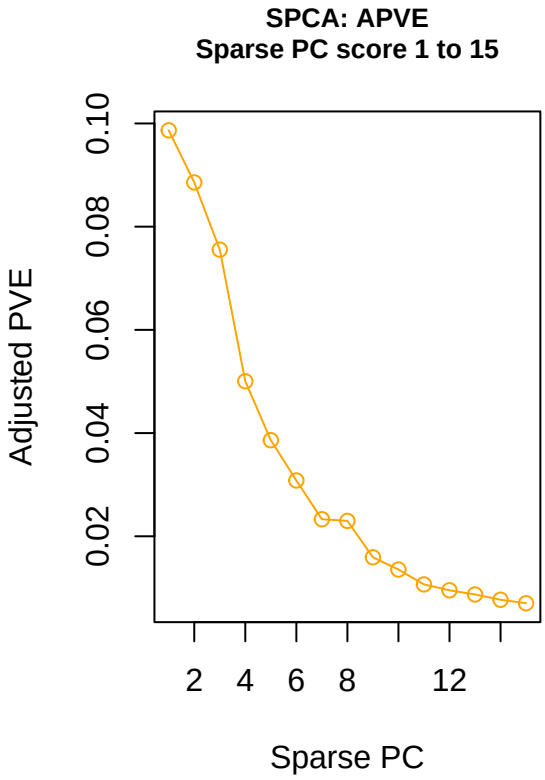
Apply Sparse PCA, provide visualizations of low-dimensional structures, and report your findings. Note that you need to use "labels.csv" for the task of discovering patterns. Your laptop may not have sufficient computational power to implement Sparse PCA with many principal components. So, please pick a value for the sparsity controlling parameter and a value for the number of principal components to be computed that suit your computational capabilities.

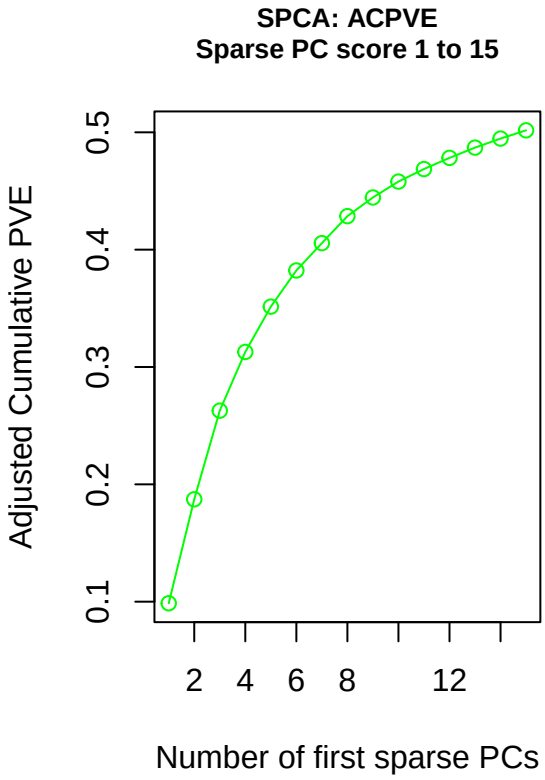



```
## [1] 824
```

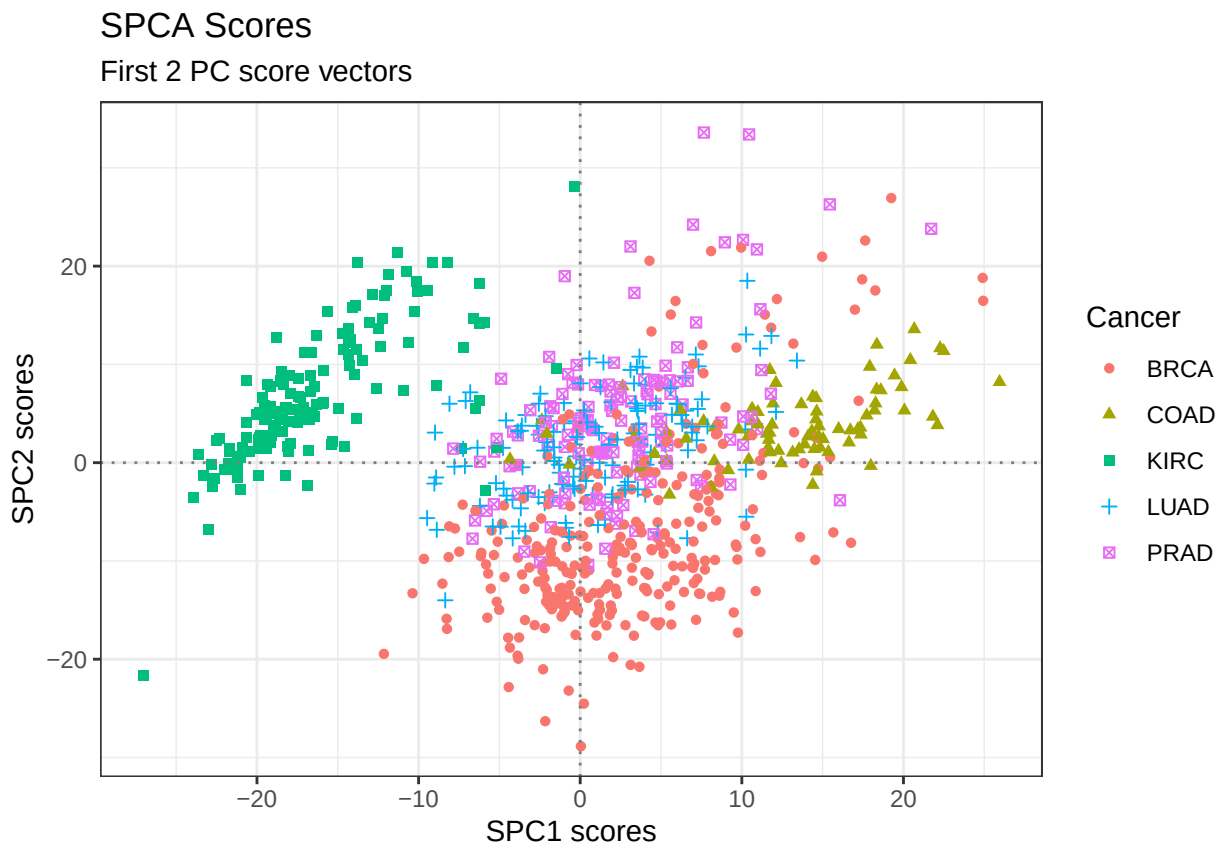
```
## [1] 42
```

Through the top four loading vectors of SPCA, each load vector has nonzero entries and does appear to have a large dominant features in direction. So when I checked the first loading vector, I found that there are **824** nonzero entries in the first loading vector and **42** large dominant features.

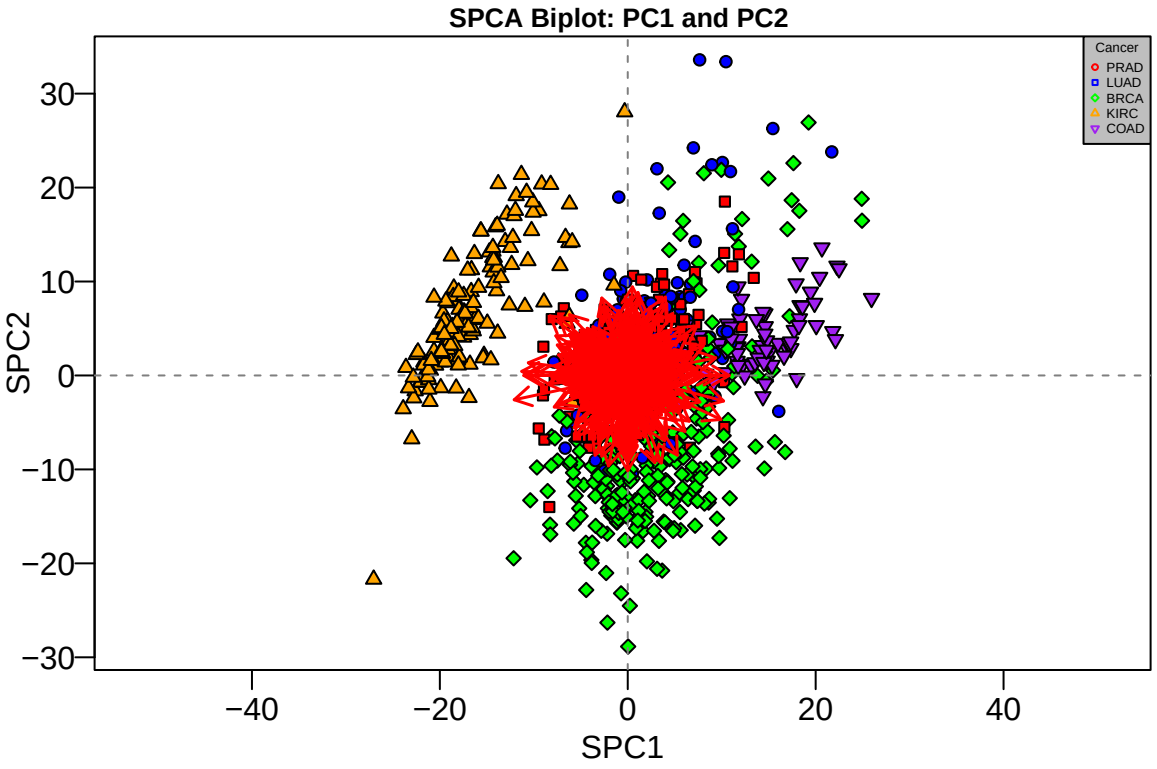




After setting the sparsity control parameter values of SPCA to 15, and checking the adjusted variance ratio and the adjusted cumulative variance ratio, it is recommended to use up to 7 main components because the slope changes rapidly at 7 points.



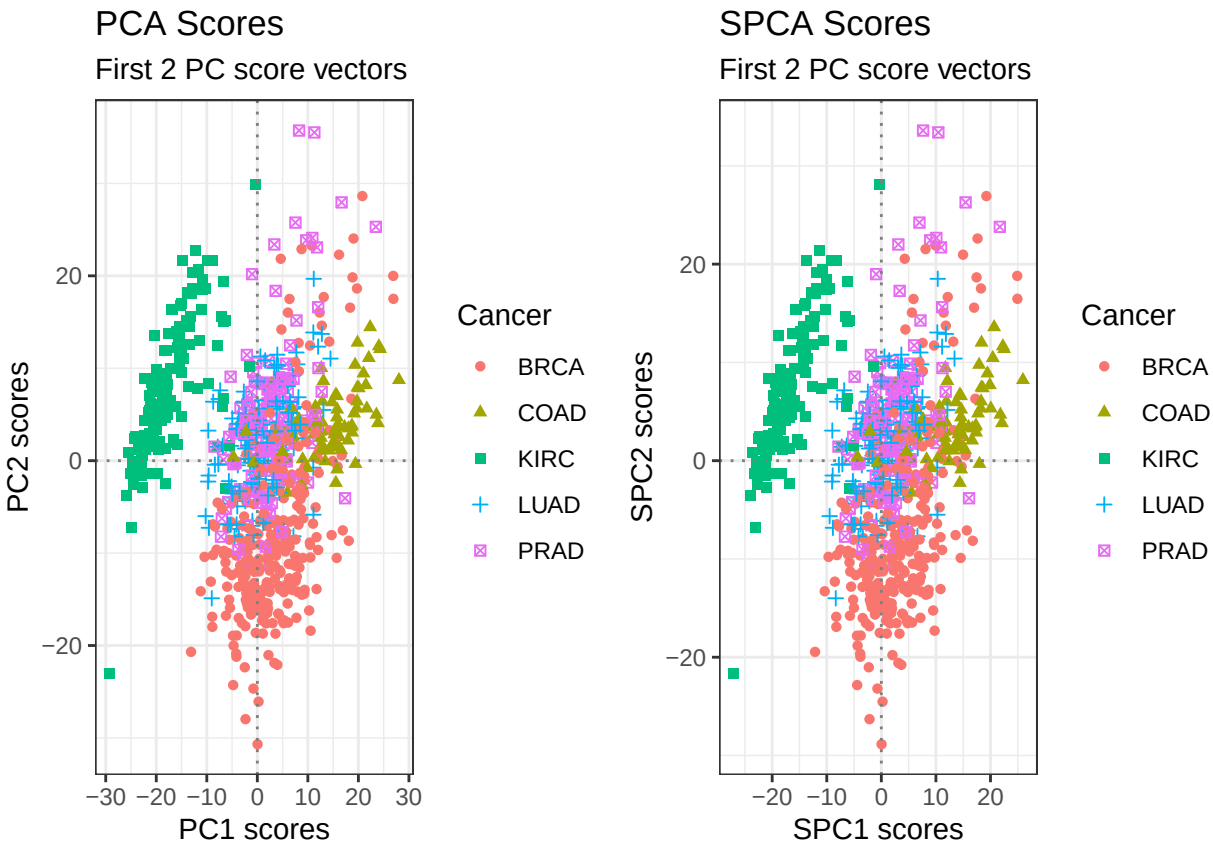
Visualization using SPCA's first two PC score vectors shows that `KIRC` and `COAD` are relatively well classified and the other three classes are quite overlapping, making it difficult to distinguish a particular signature.



As a result of visualization to check the demarcation of data through SPCA, there are too many components to check the exact demarcation.

2.1.7 2.c

Do PCA and Sparse PCA reveal different low-dimensional structures for the gene expressions for different cancer types?



Comparing the graphs of PCA and Sparse PCA through the first two pc score vectors, it can be seen that since the two graphs have very similar values and structures, **they reveal the same low-dimensional structure for gene expression for different cancer types**.

2.2 Task B: Analysis of SPAM emails data set

2.2.1 Data processing

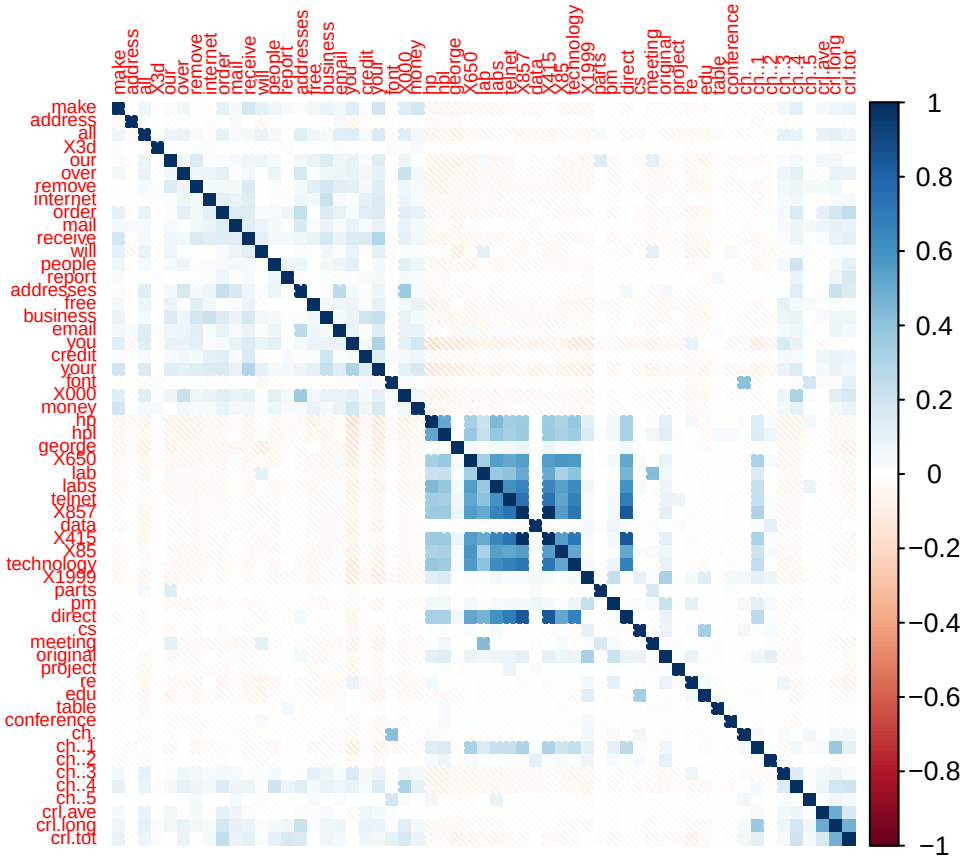
- Remove rows that have missing values. For a .csv file, usually a blank cell is treated as a missing value.

```
## [1] 4601   58
```

After checking the rows and columns of preprocessed data from which unnecessary rows `testid` and missing values have been removed, it can be seen that they have 4601 columns and 58 rows.

- Check for highly correlated features using the absolute value of sample correlation. For example, "crl.ave" (average length of uninterrupted sequences of capital letters), "crl.long" (length of longest uninterrupted sequence of capital letters) and "crl.tot" (total number of capital letters in the e-mail) may be highly correlated. Whether you choose to remove some highly correlated features from subsequent analysis or not, you need to provide a justification for your choice.

Note that each feature is stored in a column of the original data set and each observation in a row. You will analyze the processed data set.



```
##      row col
## X415  34  32
## X857  32  34
```

```
## [1] 4601   56
```

Prior to the analysis, features with high correlation between features can affect the model result, so we checked features with an absolute value of 90% or more of the correlation, and found that **x415** and **x857** have high correlation with each other. After removing these two features and preprocessing the data again, it can be confirmed that the data preprocessing was well done with 4601 rows and 56 rows.

2.2.2 3.a

Use `set.seed(123)` wherever the command `sample` is used or cross-validation is implemented, randomly select without replacement 300 observations from the data set and save them as training set "train.RData", and then randomly select without replacement 100 observations from the remaining observations and save them as "test.RData". You need to check if the training set contains observations from both classes; otherwise, no model can be trained.

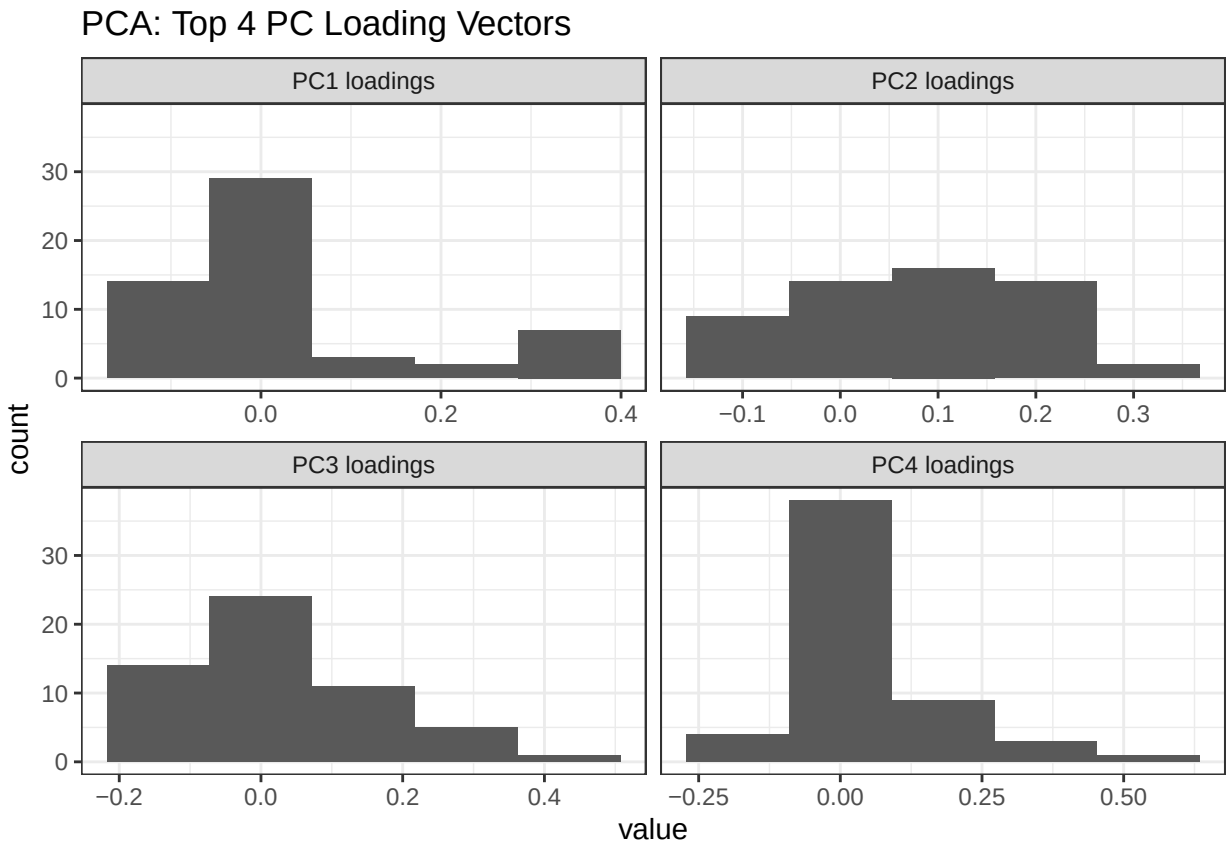
```
##
## FALSE  TRUE
##   184   116
```

```
##
## FALSE  TRUE
##    56    44
```

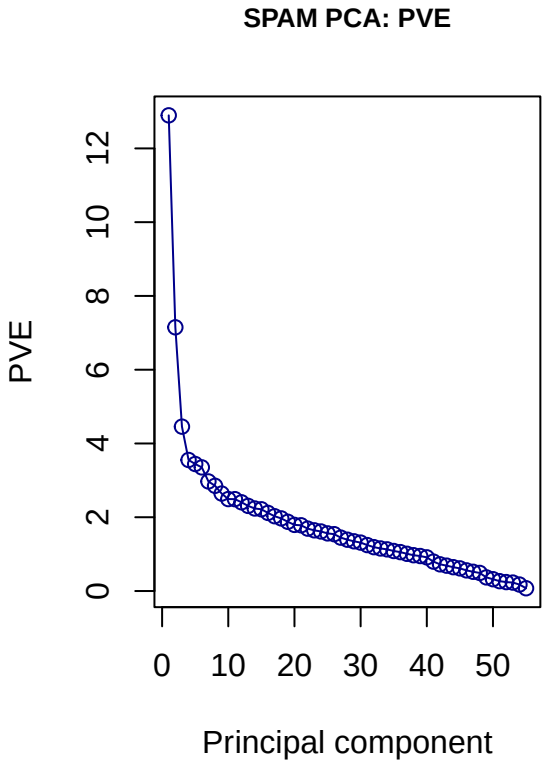
After randomly storing 300 observations in a train set and storing the remaining 100 observations in a randomized test set, it can see that the training set has 116 spam, 184 non-spams, and the test set has 44 spam and 56 non-spams.

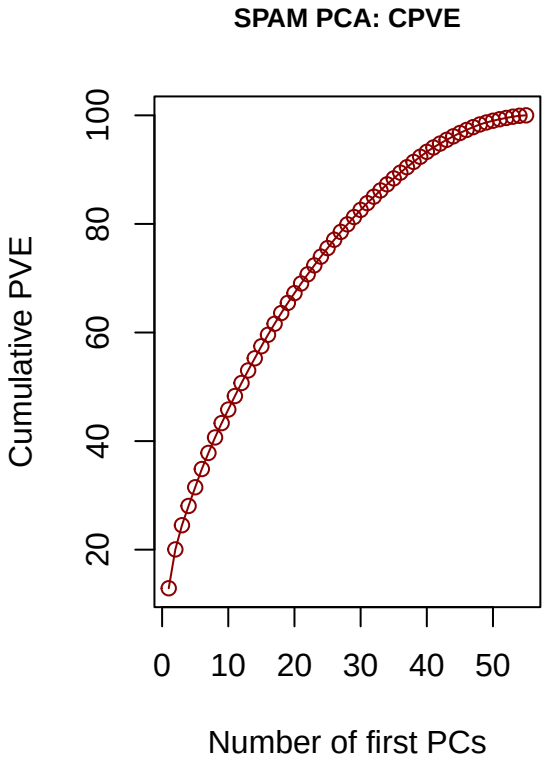
2.2.3 3.b

Apply PCA to the training data "train.RData" and see if you find any pattern that can be used to approximately tell a spam email from a non-spam email.

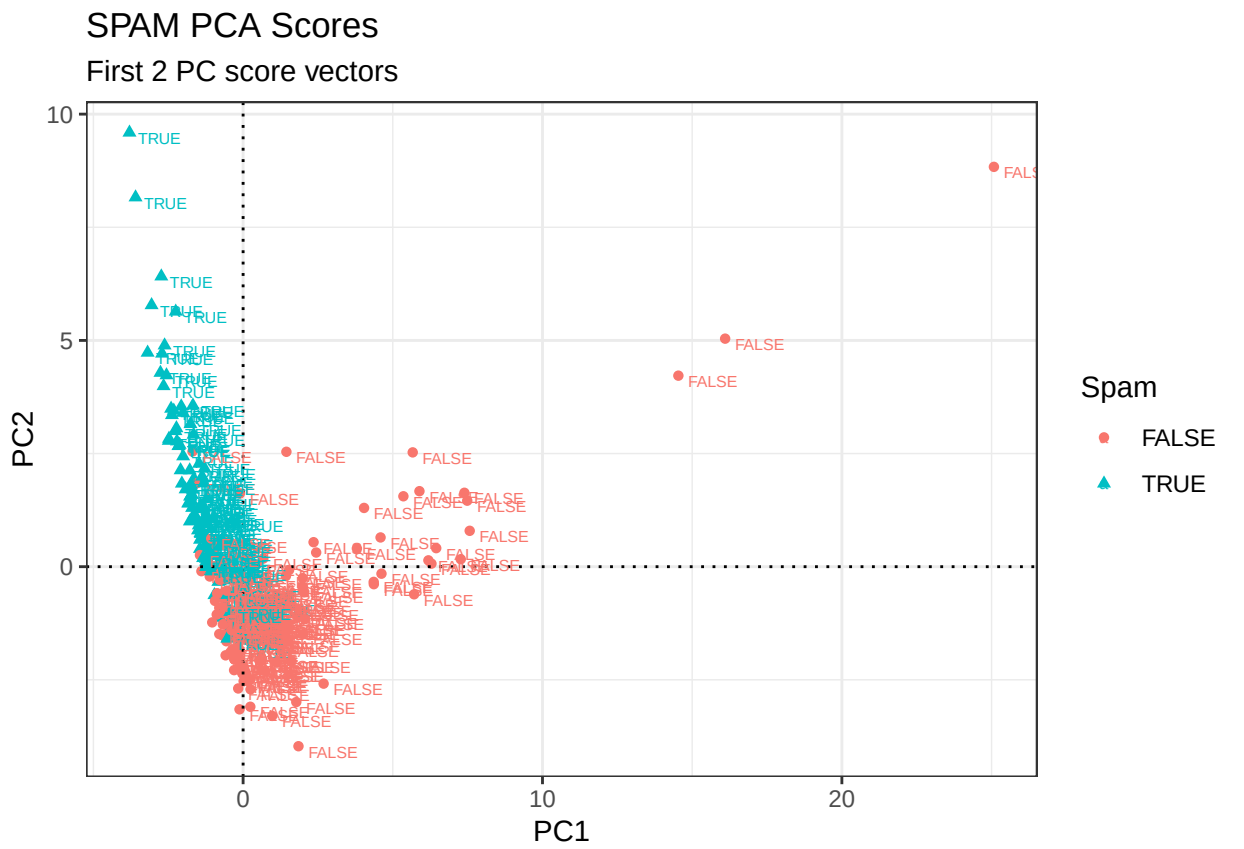


With the top four PCA load vectors of training data, each load vector has many nonzero entries, and there seems to be little dominant function in either direction.

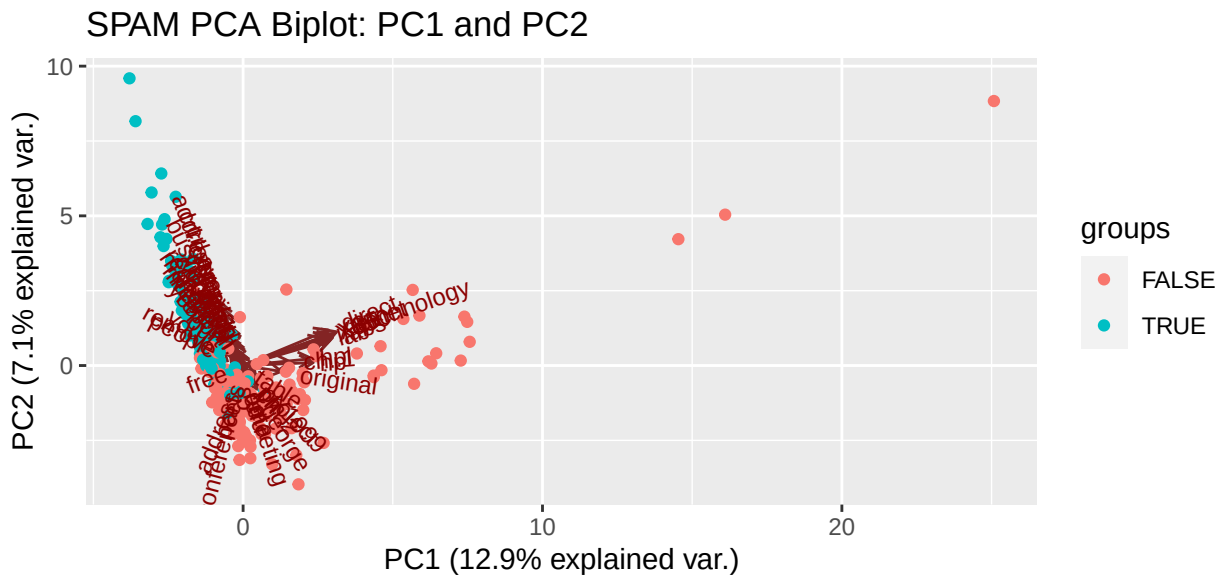




As a result of checking the variance ratio and cumulative variance ratio of the PCA in the training set, it is judged that it is better to use up to four main components because the slope changes rapidly at four points.



As a result of visualizing and finding patterns using the first two PC score vectors, it can be seen that the characteristic of `spam` is that the score of PC1 is strong negative and the score of PC2 is higher than the middle. The characteristic of `non-spam` can be seen that the score of PC1 is strong positive and the score of PC2 is slightly lower than the middle. In addition, it can be seen that the two classes are relatively well separated.



As a result of checking the biplot in the spam dataset, “direct”, “technology” has a strong positive load on PC1, and “free” has a weak negative load on PC1. Also, although you can’t see it clearly, it can be seen that “business” and “money” have a strong positive load on PC2, while “addresses” and “conference” have a weak negative load.

3 Discussion

In analyzing genetic data and spam data, it would be good to check accuracy and AUC by adding linear discriminant analysis, duplicate analysis, and distance-based classification analysis. In addition, when checking biplot using genetic data, it is worth considering checking biplot by reducing the number of ingredients because too many ingredients could not produce accurate results.

4 Appendix

4.1 Task A

4.1.1 Data preprocessing

```
# import csv file
dt <- read.csv("data.csv", row.names = 1, header = TRUE)
# filter out expressions are zero for at least 300 subjects
gexp2 <- dt[, (colSums(dt == 0, na.rm = TRUE) <= 300), drop=TRUE]
# select randomly 1000 genes and expressions from gexp2
set.seed(123)
gexp3 <- sample(gexp2, 1000, replace = FALSE)
# standardize the gene expression
stdgexpProj2 <- scale(gexp3, center = TRUE, scale = TRUE)
# show number of columns and rows
dim(stdgexpProj2)
```

4.1.2 1.a

```
# histogram
hist(stdgexpProj2, main="Gene distribution histogram",
     col="darkmagenta")
# correlations among features
corMat <- cor(stdgexpProj2[,c(1:10)])
pairs(corMat)
corrplot(corMat, method="shade",
         tl.cex=0.57)
```

4.1.3 1.b

```
# create stdgexpProj2.1 data frame
stdgexpProj2.1 <- as.data.frame(stdgexpProj2)
# import csv file
lb <- read.csv("labels.csv", row.names = 1, header = TRUE)
# set labels in data frame
stdgexpProj2.1$Cancer <- lb[,1]
stdgexpProj2.1 <- stdgexpProj2.1[, c(1001, 1:1000)]
# show distribution of Cancer types plot
ggplot(stdgexpProj2.1, aes(Cancer, fill=Cancer)) +
   geom_bar() +
  ggtitle('Cancer Bar', subtitle = "Distribution of the Cancer") +
  theme_minimal()
```

```
# PCA for standardized data
pcaDt = prcomp(stdgexpProj2)
# first 15 sparse PCs are obtained by setting tuning parameters
spcaDt = elasticnet::spca(stdgexpProj2, K=15,
                          para=rep(1e-6,15),
                          type=c("predictor"),
                          sparse=c("penalty"),
                          lambda=1e-6, max.iter = 200,
                          eps.conv=1e-3)
# store first 2 score vectors in a data.frame: PCA
scoresMat = as.data.frame(pcaDt$x[,1:2])
# add labels
scoresMat$Cancer = lb[,1]
# create pca plot: first 2 pc
pcaPt <- ggplot(scoresMat, aes(PC1, PC2,
                               color = Cancer, shape = Cancer)) +
   geom_point() + theme_bw() +
   xlab("PC1 scores") +
   ylab("PC2 scores") +
   geom_hline(yintercept = 0, linetype="dotted", color = 'gray50') +
   geom_vline(xintercept = 0, linetype="dotted", color = 'gray50')  +
  ggtitle("PCA Scores", subtitle = "First 2 PC score vectors")
pcaPt
# get first 2 loading vectors and score vectors
sLVTwo = spcaDt$loadings[,1:2]
sSVTwo = stdgexpProj2%*%sLVTwo
# store first 2 score vectors in a data.frame: SPCA
scoresMat1 = as.data.frame(sSVTwo)
# add labels
scoresMat1$Cancer = lb[,1]
# create spca plot: first 2 spc
pcaPt1 <- ggplot(scoresMat1, aes(PC1, PC2,
                               color = Cancer, shape = Cancer)) +
   geom_point() + theme_bw() +
   xlab("SPC1 scores") +
   ylab("SPC2 scores") +
   geom_hline(yintercept = 0, linetype="dotted", color = 'gray50') +
   geom_vline(xintercept = 0, linetype="dotted", color = 'gray50') +
  ggtitle("SPCA Scores", subtitle = "First 2 PC score vectors")
pcaPt1
```

4.1.4 1.c

```
# compute PVE: PCA
pve = 100*pcaDt$sdev^2/sum(pcaDt$sdev^2)
# compute APVE: SPCA
```

```
pve1 = spcaDt$pev
# PCA CPVE
cumsum(pve[c(1,2,3,4,5)])
# SPCA ACPVE
cumsum(pve1[c(1,2,3,4,5)])
```

4.1.5 2.a

```
# pc loading vectors: first 4 PC
fourLD1 <- pcaDt$rotation[,1:4]
# change column names of fourLD1
colnames(fourLD1) = paste(colnames(fourLD1), "loadings", sep= " ")
# melt the matrix fourLD1 for plotting
dstack1 <- melt(fourLD1)
# First 4 PC loading vectors plot
lvplot1 = ggplot(dstack1, aes(x= value)) +
   geom_histogram(bins=25) +
   facet_wrap(~Var2, scales = "free_x") + theme_bw() +
  ggtitle("PCA: Top 4 Loading Vectors")
lvplot1
# proportion of variance explained
# PCA plot of PVE and CPVE
par(mfrow=c(1,2))
plot(1:length(pve), pve, type="o",
     ylab="PVE",
     col="blue", xlab = "Principal component")
title("PCA: PVE", cex.main=0.8)
plot(1:length(pve[1:15]), pve[1:15],
     type="o", ylab="PVE",
     col='blue', xlab = "Principal component")
title("PCA: PVE\nPC score 1 to 15",
      cex.main=0.8, cex.sub=0.5)
par(mfrow=c(1,2))
plot(cumsum(pve), type="o", col="red", ylab="Cumulative PVE",
     xlab="Number of first PCs")
title("PCA: CPVE", cex.main=0.8)
plot(cumsum(pve[1:15]), type="o",
     ylab="Cumulative PVE", col="red",
     xlab="Number of first PCs")
title("PCA: CPVE\nPC score 1 to 15",
      cex.main=0.8, cex.sub=0.5)
# show pca plot: first 2 PC
pcaPt
# pca biplot: first 2PC
biPCA = ggbiplot(pcaDt, choices = 1:2, obs.scale = 1, var.scale = 1,
```

```
                groups = stdgexpProj2.1$Cancer) +
  ggtitle("PCA Biplot: PC1 and PC2 ")
biPCA
```

4.1.6 2.b

```
# adjusted percentage of explained variance
fourLD2 = spcaDt$loadings[,1:4]
# change column names
colnames(fourLD2) = paste("SPC", 1:4, sep="")
colnames(fourLD2) = paste(colnames(fourLD2), "loadings", sep=" ")
# melt matrix fourLD2 for plotting
fdStack <- melt(fourLD2)
# first 4 loading vectors plot
histSpcaPt <- ggplot(fdStack, aes(x= value)) +
   geom_histogram(bins=30) +
   facet_wrap(~Var2, scales="free") +
   theme_bw() +
  ggtitle("SPCA: Top 4 Loading Vectors")
histSpcaPt
# dominant features
# first sparse loading vector
LV1 <- fourLD2[,1]
# all dominant features as those with nonzero loadings
df1 <- which(abs(LV1) > 0)
length(df1)
# dominant features as those with large loadings
df2 <- which(abs(LV1) >
               quantile(abs(LV1[abs(LV1)>0]),
                        probs = c(0.95)))
length(df2)
# plot adjusted PVE and adjusted CPVE; 15 sparse PCs
par(mfrow=c(1,2))
plot(pve1, type="o", ylab="Adjusted PVE", xlab="Sparse PC", col="orange")
title("SPCA: APVE\nSparse PC score 1 to 15",
      cex.main=0.8, cex.sub=0.5)
plot(cumsum(pve1), type="o", ylab="Adjusted Cumulative PVE",
     xlab="Number of first sparse PCs", col="green")
title("SPCA: ACPVE\nSparse PC score 1 to 15",
      cex.main=0.8, cex.sub=0.5)
# show spca plot
pcaPt1
# create color and shape scheme
pch.group = rep(21, nrow(sSVTwo))
pch.group[lb == "PRAD"] = 21;
pch.group[lb == "LUAD"] = 22;
```

```
pch.group[lb == "BRCA"] = 23;
pch.group[lb == "KIRC"] = 24;
pch.group[lb == "COAD"] = 25;
col.group = rep("blue", nrow(sSVTwo))
col.group[lb == "PRAD"] ="blue"
col.group[lb == "LUAD"] ="red"
col.group[lb == "BRCA"] ="green"
col.group[lb == "KIRC"] ="orange"
col.group[lb == "COAD"] ="purple"
# show spca biplot
par(mar=c(5,4,1,1), mgp=c(1.5,0.5,0))
plot(sSVTwo[,1], sSVTwo[,2], xlab="SPC1", ylab="SPC2",
     col="black", pch=pch.group, bg = col.group,
     las=1, asp=1, cex=0.8)
legend("topright", legend=c("PRAD", "LUAD", "BRCA", "KIRC", "COAD"),
       col=c("red", "blue", "green", "orange", "purple"),
       pch = c(21,22,23,24,25),
       bg="gray", title = "Cancer",
       cex = 0.4)
abline(v=0, lty=2, col="grey50")
abline(h=0, lty=2, col="grey50")
l.x=100*sLVTwo[,1]; l.y=100*sLVTwo[,2]
arrows(x0=0, x1=l.x, y0=0, y1=l.y,
col="red",length=.1, lwd=1.5)
title("SPCA Biplot: PC1 and PC2 ", cex.main=0.8)
```

4.1.7 2.c

```
# PCA and Sparse PCA plots on same rows for compare
grid.arrange(pcaPt, pcaPt1, nrow=1)
```

4.2 Task B

4.2.1 Data preprocessing

```
# import spam csv file
spam <- read.csv("SPAM.csv", stringsAsFactors = FALSE, header = TRUE)
# remove testid columns
spam <- subset(spam, select = -c(2))
# change spam variable type
spam$spam <- as.factor(spam$spam)
# check missing values and remove it
spam <- na.omit(spam)
# show number of columns and rows
```

```r
dim(spam)
# correlations among features
corMatFeature <- cor(spam[,c(-1)])
corrplot(corMatFeature, method="shade",
         tl.cex=0.57)
# check correlation greater than 0.9 in absolute value
which(abs(corMatFeature) > 0.9 & abs(corMatFeature) != 1, arr.ind = T)
# remove highly correlated features
spam.new <- subset(spam, select = -c(33,35))
# recheck the number of columns and rows
dim(spam.new)
```

4.2.2 3.a

```r
set.seed(123)
# randomly select without replacement 300 obs from data set and set train
train.Rdata <- spam.new[sample(nrow(spam.new), 300, replace = F),]
# randomly select without replacement 100 obs from data set and set test
test.Rdata <- spam.new[sample(nrow(spam.new), 100, replace = F),]
# show train table
table(train.Rdata$spam)
# show test table
table(test.Rdata$spam)
```

4.2.3 3.b

```r
# standardize train set
train.R <- scale(train.Rdata[,c(-1)], center = TRUE, scale = TRUE)
# PCA for standardized data
pcaSpam <- prcomp(train.R)
# store first 4 PC loading vectors in a matrix
pc4Spam <- pcaSpam$rotation[,1:4]
# change column names
colnames(pc4Spam) = paste(colnames(pc4Spam), "loadings",sep=" ")
# melt the pc4Spam for plotting
pc4SpamSt <- melt(pc4Spam)
# histogram for each 4 pc loading vectors
lvSpamPlot <- ggplot(pc4SpamSt, aes(x = value)) +
  geom_histogram(bins=5) +
  facet_wrap(~Var2, scales="free_x") +
  theme_bw() +
  ggtitle("PCA: Top 4 PC Loading Vectors")
lvSpamPlot
# compute PVE for each PC
```

```
spamPve = 100 * pcaSpam$sdev^2/ sum(pcaSpam$sdev^2)
# Plot PVE and CPVE
par(mfrow=c(1,2))
plot(1:length(spamPve), spamPve, type="o", col="darkblue",
     ylab ="PVE",
     xlab="Principal component")
title("SPAM PCA: PVE", cex.main=0.8)
plot(cumsum(spamPve), type="o", col="darkred",
     ylab="Cumulative PVE",
     xlab="Number of first PCs")
title("SPAM PCA: CPVE", cex.main=0.8)
# store first 2 score vectors in a data.frame: SPCA
scoresSpam <- as.data.frame(pcaSpam$x[,1:2])
# set labels in data frame
scoresSpam$Spam <- train.Rdata$spam
# spam pca plot: first 2 PC
psvSpam <- ggplot(scoresSpam, aes(PC1, PC2,
                                  color = Spam,
                                  shape=Spam)) +
  geom_point() +
  theme_bw() +
  geom_hline(yintercept = 0, linetype="dotted") +
  geom_vline(xintercept=0, linetype="dotted") +
  geom_text(aes(label=Spam), hjust=-0.2,vjust=1, size=2) +
  ggtitle("SPAM PCA Scores", subtitle = "First 2 PC score vectors")
psvSpam
# show biplot of spam data
biPCA11 = ggbiplot(pcaSpam, choices = 1:2, obs.scale = 1,
                   var.scale = 1, groups = train.Rdata$spam) +
  ggtitle("SPAM PCA Biplot: PC1 and PC2 ")
biPCA11
```