STAT40150 Multivariate Analysis Assignment 1

K.Saketh Sai Nigam 22201204

2023-03-23

QUESTION

- **This assignment is based on the milk spectra data set (Milk_MIR_Traits_data_2023.csv) available on Brightspace. Background to these mid-infrared (MIR) spectral data extracted from milk samples, and their associated protein and technological traits, is provided in the recorded invited lecture from adjunct Assoc. Prof. Sinead McParland (see Brightspace), and in the pdf slidedeck SMcPar- land_LectureSlides_2023.pdf on Brightspace. Download the dataset Milk_MIR_Traits_data_2023.csv from Brightspace.**
- **The initial columns in the dataset contain details (i.e. covariates) of the cows which produced the milk samples and the protein and technological traits of the milk samples measured in the laboratory. The data in the final 531 columns are the MIR spectra, with the first row of these columns detailing the wavenumber (measured in cm−1). The spectral values in the dataset are the absorbance values (the log10 of the reciprocal of the transmittance value). The water region has been removed.**

==

1. Load the data set into R. Use the set.seed function in R to set the seed to your student number. Randomly generate a number between 1 and n (where n is the number of rows in the dataset), and delete that observation/row from the dataset. Ensure that you include the code used in this step in the R code you submit with your assignment so that your work can be reproduced. [0 marks]

==

```
#You may employ the set.seed() function to set the seed using your student number as shown below:
set.seed(22201204)

# retrieve the data file and save it as a "MilkDataFrame" data frame.
MilkDataFrame <- read.csv("/Users/saketh/Desktop/MS SEM 2/Multivariant/Ass1/Milk_MIR_Traits_data_2023.csv")
cat("Original Dimensions of MilkDataFrame: ",dim(MilkDataFrame))
```

```
## Original Dimensions of MilkDataFrame:  431 582
```

```
# a randomized row index is produced.
MilkDataFrameRandomRow <- sample(nrow(MilkDataFrame), 1)
cat("The Randomly Generated Row in the MilkDataFrame: ",MilkDataFrameRandomRow)
```

```
## The Randomly Generated Row in the MilkDataFrame:  43
```

```
# The data frame is subset to eliminate the randomly chosen row.
MilkDataFrame <- MilkDataFrame[-MilkDataFrameRandomRow, ]
cat("Updated Dimensions of the MilkDataFrame after removing the",MilkDataFrameRandomRow,"row is: ",dim(MilkDataFr
ame))
```

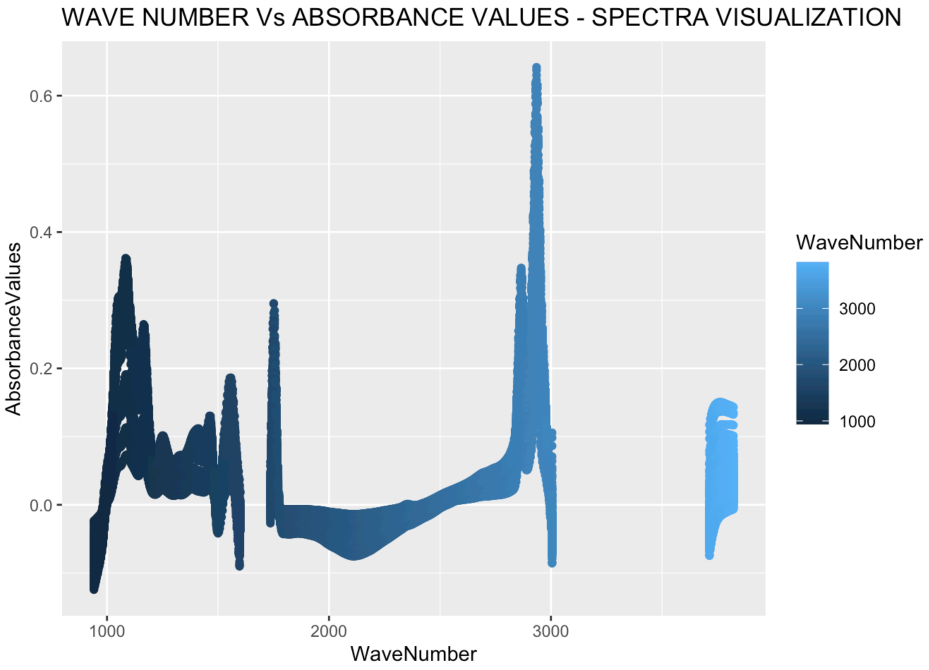
```
## Updated Dimensions of the MilkDataFrame after removing the 43 row is:  430 582
```

==

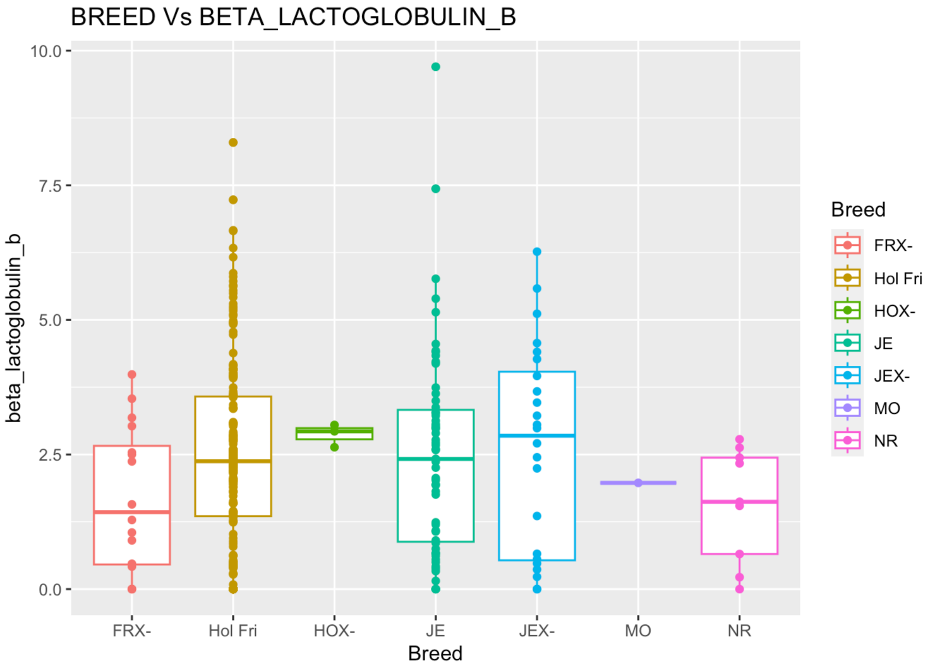
2. The milk protein β Lactoglobulin B is used in the production of protein drinks. Remove from the dataset any record/observation which has a missing/NA value for β Lactoglobulin B. Then, visualise the spectra and the protein trait β Lactoglobulin B using (separate) suitable plots. Comment on the plots. Remove any observations with β Lactoglobulin B outside of 3 standard deviations from the mean of the trait. [10 marks]

==

```
#Removing the Missing/NA Values in the MilkDataFrame
MilkDataFrame <- MilkDataFrame[complete.cases(MilkDataFrame$beta_lactoglobulin_b), ]

#Creating the Spectral Dataset
MilkDataFrameSpectral <- MilkDataFrame[, 52:ncol(MilkDataFrame)]
#Assigning Spectral Dataset to a Variable
MilkDataFrameSpectralColumnNames = colnames(MilkDataFrameSpectral)

#Loading Library required for Reshapping the Dataset
library(tidyr)
#pivot the data frame into a long format
MDFSEdited = MilkDataFrameSpectral %>% pivot_longer(cols=c(MilkDataFrameSpectralColumnNames),
                   names_to='WaveNumber',
                   values_to='AbsorbanceValues')
MDFSEdited$WaveNumber <- as.numeric(sub("^X", "", MDFSEdited$WaveNumber))

#Loading Library required for Plotting the Dataset
library("ggplot2")
# Plotting with ggplot2 package
ggplot(MDFSEdited,aes(WaveNumber, AbsorbanceValues, color= WaveNumber)) +
  ggtitle("WAVE NUMBER Vs ABSORBANCE VALUES - SPECTRA VISUALIZATION") +
  geom_point()
```



```
#Assigning the Trait of the dataframe in the variable
betalactoglobulinb <- MilkDataFrame$beta_lactoglobulin_b

# Plotting with ggplot2 package
ggplot(MilkDataFrame,aes(x=Breed,y=beta_lactoglobulin_b, color=Breed)) +
  geom_boxplot() +
 ggtitle("BREED Vs BETA_LACTOGLOBULIN_B") +
   geom_point()
```



```
#Calculating Mean for Betalactoglobulinb
Meanofbetalactoglobulinb <- mean(betalactoglobulinb)
cat("Mean of the betalactoglobulinb Trait values: ",Meanofbetalactoglobulinb)
```

```
## Mean of the betalactoglobulinb Trait values:  2.432206
```

```
#Calculating Standard Deviation for Betalactoglobulinb
SDofbetalactoglobulinb <- sd(betalactoglobulinb)
cat("Standard Deviation of the betalactoglobulinb Trait values: ",SDofbetalactoglobulinb)
```

```
## Standard Deviation of the betalactoglobulinb Trait values:  1.749949
```

```
#Calculating Range by Upper and Lower values
LowerCutOff = Meanofbetalactoglobulinb - 3*(SDofbetalactoglobulinb)
UpperCutOff = Meanofbetalactoglobulinb + 3*(SDofbetalactoglobulinb)

#Removing any observations with β Lactoglobulin B outside of 3 standard deviations from the mean of the trait
MilkDataFrame <- MilkDataFrame[(betalactoglobulinb >= LowerCutOff) && (betalactoglobulinb <= UpperCutOff),]
cat("The Dimensions of the new Formed Milk Dataset: ",dim(MilkDataFrame))
```

```
## The Dimensions of the new Formed Milk Dataset:  305 582
```

==

PLOT INFERENCE:-

(WAVE NUMBERS Vs ABSORBANCE VALUES) PLOT INTERPRETATION:-

Hence, it is evident from the query that the very first row of wavelengths in the query ranges from 941 to 3822, and the values are the Spectra Values, which reflect how so much energy is absorbed by the milk at each particular wavelength. Hence, it is evident from the Figure that the lowest absorbance value is at 941 (cm-1) wavelength and greater in around 2996. (cm-1)

BOXPLOT INTERPRETATION:-

A box plot is made using points for the beta lactoglobulin b variable of the MilkDataFrame dataset, which is grouped by the Breed variable. More specifically, the beta lactoglobulin b variable is represented on the y-axis of the figure, and Breed is represented on the x-axis. The box plot, which displays the beta lactoglobulin b variable’s distribution for each Breed, is made using the geom boxplot() method.The Range is very low in the “HOX” Breed and High in “JEX”

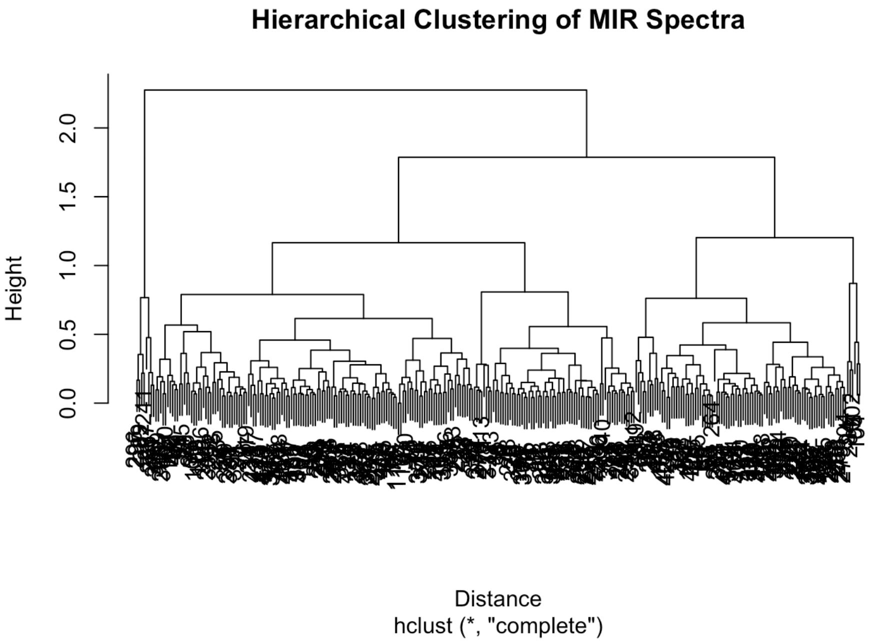
==

3. Use hierarchical clustering and k-means clustering to determine if there are clusters of similar MIR spectra in the data. Motivate any decisions you make. Compare the hierarchical clustering and k-means clustering solutions. Comment on/explore any clustering structure you uncover, considering the data generating context. [25 marks]

==

HIERARCHICAL CLUSTERING

```
#Calculate Distance using euclidean method for each cluster
Distance <- dist(((MilkDataFrameSpectral)),method = "euclidean")
#Do Hierarchical Clustering by 'hclust()'
HierarchicalClustering <- hclust(Distance)
#Plotting the dendrogram structure in Hierarchical Clustering
plot(HierarchicalClustering, main = "Hierarchical Clustering of MIR Spectra")
```

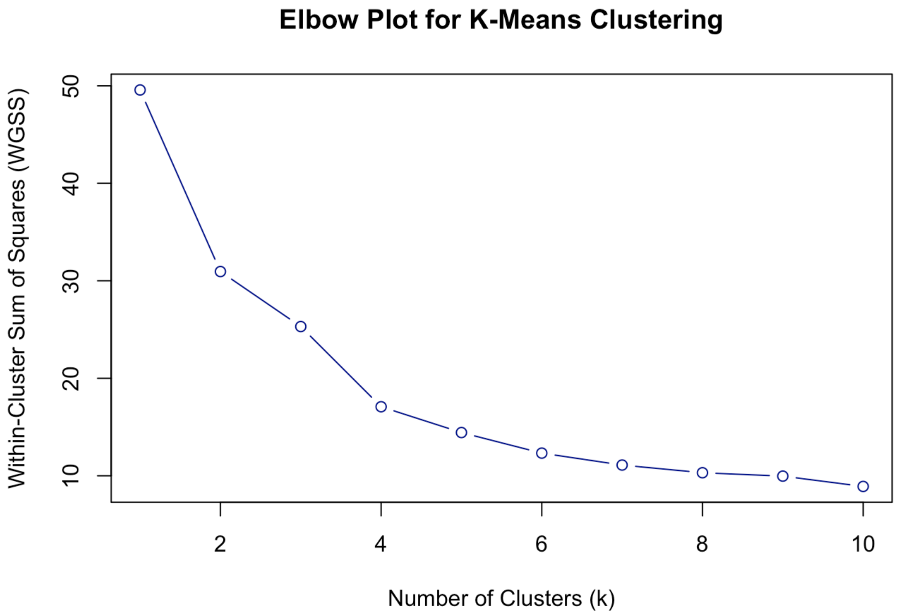


K-MEANS CLUSTERING

```
#Within-Group Sum of Squares is done to Do the K means Clustering
SVar = MilkDataFrameSpectral
WGSS = rep(0,10)
n = nrow(SVar)
WGSS[1] = (n-1) * sum(apply(SVar, 2, var))

for (k in 2:10) {
  WGSS[k] = sum(kmeans(SVar, centers = k)$withinss)
}
```

WGSS Vs K-Values Plot:-

```
#Optimal calculation of K value using plot between WGSS and K- values
plot(1:10, WGSS, type = "b", col="darkblue",xlab = "Number of Clusters (k)", ylab = "Within-Cluster Sum of Square
s (WGSS)", main = "Elbow Plot for K-Means Clustering")
```



The "elbow" of the plot, which is where the decline in WGSS starts to level off, can be used to calculate the ideal value of K. According to the plot, the ideal value of K in this situation is most likely to fall between 2 and 4 and the value is 3

```
#K-means clustering with 3 clusters
KmeansClustering = kmeans(SVar, centers = 3)
#Cutting Hierarchical clustering with 3 clusters
CutHierarchialClustering = cutree(HierarchicalClustering, k=3)
#Table formation
CombinationalTable = table(CutHierarchialClustering, KmeansClustering$cluster)

#Comparing the 2 clusters using "e1071" Package
library(e1071)
#Comparing TWO Clusters
ClassAggForm = classAgreement(CombinationalTable)
cat("The Comparision value: ",ClassAggForm$crand)
```

```
## The Comparision value:  0.5417111
```

While a number near to 0 shows little agreement beyond chance and a negative value implies disagreement, a value close to 1 suggests substantial agreement between the two clustering algorithms.So, from output we can tell that accuracy is of 54% have the same values for the both clustering

==

4. Apply principal components analysis to the spectral data, motivating any decisions you make in the process. Plot the cumulative proportion of the variance explained by the first 10 principal components. How many principal components do you think are required to represent the spectral data? Explain your answer. [10 marks]

==

NAMES OF THE CATEGORIES OF THE PRINCIPAL COMPONENT ANALYSIS DATASET:-

```
#Performing PCA for given DataSet and Assigining to Variable
PCAOfDataset = prcomp(SVar)
#The Names of the labels of the PCAOfDataset
PCANames = c(names(PCAOfDataset))
#Printing the Names
for(i in 1:length(PCANames))
{
  cat("\nThe Names of the PCA Category", i,":", PCANames[i])
}
```

```
## 
## The Names of the PCA Category 1 : sdev 
## The Names of the PCA Category 2 : rotation 
## The Names of the PCA Category 3 : center 
## The Names of the PCA Category 4 : scale 
## The Names of the PCA Category 5 : x
```

STANDARD DEVIATION:-

```
# Printing the standard deviation
cat("The Standard Deviation is: ",head(PCAOfDataset$sdev))
```

```
## The Standard Deviation is:  0.3212475 0.1754362 0.1446442 0.08290299 0.02129296 0.01591997
```

SUMMARY OF THE PCAOfDataset:-

```
#The Summary of the PCAofDataset
summary(PCAOfDataset)
```

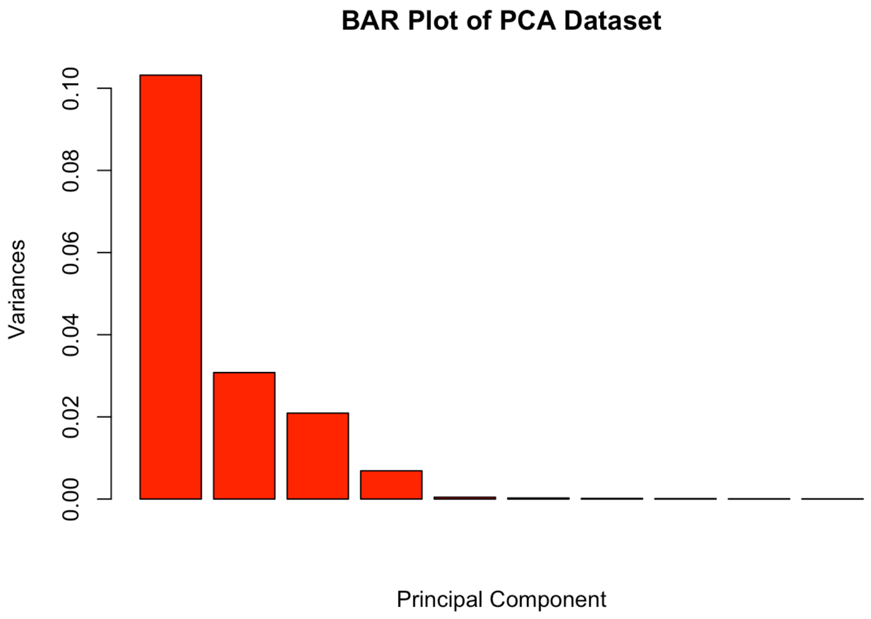
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     0.3212 0.1754 0.1446 0.08290 0.02129 0.01592 0.01258
## Proportion of Variance 0.6329 0.1887 0.1283 0.04215 0.00278 0.00155 0.00097
## Cumulative Proportion  0.6329 0.8216 0.9499 0.99204 0.99482 0.99638 0.99735
##                            PC8      PC9     PC10     PC11    PC12     PC13
## Standard deviation     0.01055 0.008307 0.007265 0.005484 0.00475 0.004606
## Proportion of Variance 0.00068 0.000420 0.000320 0.000180 0.00014 0.000130
## Cumulative Proportion  0.99803 0.998450 0.998780 0.998960 0.99910 0.999230
##                            PC14     PC15     PC16     PC17     PC18     PC19
## Standard deviation     0.004202 0.003913 0.003551 0.003084 0.002861 0.002659
## Proportion of Variance 0.000110 0.000090 0.000080 0.000060 0.000050 0.000040
## Cumulative Proportion  0.999340 0.999430 0.999510 0.999570 0.999620 0.999660
##                            PC20     PC21     PC22     PC23     PC24     PC25
## Standard deviation     0.002453 0.002362 0.002199 0.002086 0.001961 0.001805
## Proportion of Variance 0.000040 0.000030 0.000030 0.000030 0.000020 0.000020
## Cumulative Proportion  0.999700 0.999730 0.999760 0.999790 0.999810 0.999830
##                            PC26     PC27     PC28     PC29    PC30     PC31
## Standard deviation     0.001698 0.001519 0.001373 0.001351 0.00131 0.001223
## Proportion of Variance 0.000020 0.000010 0.000010 0.000010 0.00001 0.000010
## Cumulative Proportion  0.999850 0.999860 0.999880 0.999890 0.99990 0.999910
##                            PC32     PC33     PC34      PC35      PC36      PC37
## Standard deviation     0.001163 0.001158 0.001023 0.0009674 0.0009241 0.0008977
## Proportion of Variance 0.000010 0.000010 0.000010 0.0000100 0.0000100 0.0000000
## Cumulative Proportion  0.999910 0.999920 0.999930 0.9999300 0.9999400 0.9999500
##                            PC38      PC39      PC40      PC41      PC42
## Standard deviation     0.000859 0.0007999 0.0007654 0.0007454 0.0007353
## Proportion of Variance 0.000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  0.999950 0.9999500 0.9999600 0.9999600 0.9999600
##                             PC43      PC44      PC45      PC46      PC47
## Standard deviation     0.0006771 0.0006432 0.0005834 0.0005716 0.0005345
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  0.9999700 0.9999700 0.9999700 0.9999700 0.9999700
##                             PC48      PC49      PC50      PC51      PC52
## Standard deviation     0.0005248 0.0005157 0.0004974 0.0004725 0.0004396
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  0.9999800 0.9999800 0.9999800 0.9999800 0.9999800
##                             PC53      PC54      PC55      PC56      PC57
## Standard deviation     0.0004224 0.0004135 0.0004101 0.0003889 0.0003828
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  0.9999800 0.9999800 0.9999900 0.9999900 0.9999900
##                             PC58      PC59      PC60      PC61      PC62
## Standard deviation     0.0003651 0.0003523 0.0003326 0.0003246 0.0003108
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  0.9999900 0.9999900 0.9999900 0.9999900 0.9999900
##                             PC63      PC64      PC65      PC66      PC67
## Standard deviation     0.0002975 0.0002788 0.0002769 0.0002642 0.0002558
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  0.9999900 0.9999900 0.9999900 0.9999900 0.9999900
##                             PC68      PC69      PC70      PC71      PC72
## Standard deviation     0.0002515 0.0002464 0.0002359 0.0002322 0.0002248
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  0.9999900 0.9999900 0.9999900 0.9999900 0.9999900
##                             PC73      PC74      PC75      PC76      PC77
## Standard deviation     0.0002144 0.0002135 0.0002034 0.0001973 0.0001949
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
##                             PC78     PC79      PC80     PC81      PC82
## Standard deviation     0.0001926 0.000187 0.0001841 0.000182 0.0001745
## Proportion of Variance 0.0000000 0.000000 0.0000000 0.000000 0.0000000
## Cumulative Proportion  1.0000000 1.000000 1.0000000 1.000000 1.0000000
##                             PC83      PC84      PC85      PC86      PC87
## Standard deviation     0.0001672 0.0001613 0.0001589 0.0001521 0.0001493
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
##                             PC88      PC89      PC90      PC91      PC92
## Standard deviation     0.0001415 0.0001383 0.0001358 0.0001312 0.0001236
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
##                             PC93      PC94      PC95      PC96      PC97
## Standard deviation     0.0001196 0.0001181 0.0001148 0.0001101 0.0001048
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
##                             PC98      PC99     PC100    PC101     PC102
## Standard deviation     0.0001038 9.948e-05 9.329e-05 8.94e-05 8.794e-05
## Proportion of Variance 0.0000000 0.000e+00 0.000e+00 0.00e+00 0.000e+00
## Cumulative Proportion  1.0000000 1.000e+00 1.000e+00 1.00e+00 1.000e+00
##                            PC103    PC104     PC105     PC106     PC107
## Standard deviation     8.625e-05 8.44e-05 7.793e-05 7.695e-05 7.595e-05
## Proportion of Variance 0.000e+00 0.00e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.00e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC108     PC109    PC110     PC111     PC112
## Standard deviation     7.271e-05 6.965e-05 6.81e-05 6.715e-05 6.405e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.00e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.00e+00 1.000e+00 1.000e+00
##                            PC113     PC114     PC115     PC116     PC117
## Standard deviation     6.335e-05 6.169e-05 5.909e-05 5.853e-05 5.728e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC118     PC119     PC120     PC121     PC122
## Standard deviation     5.585e-05 5.268e-05 5.264e-05 5.194e-05 5.084e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC123     PC124     PC125     PC126     PC127
## Standard deviation     4.967e-05 4.714e-05 4.696e-05 4.399e-05 4.368e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC128     PC129     PC130     PC131     PC132
## Standard deviation     4.304e-05 4.095e-05 4.047e-05 3.953e-05 3.779e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC133     PC134     PC135     PC136     PC137
## Standard deviation     3.682e-05 3.613e-05 3.547e-05 3.425e-05 3.392e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC138     PC139     PC140     PC141     PC142
## Standard deviation     3.337e-05 3.208e-05 3.171e-05 3.045e-05 2.822e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC143     PC144     PC145     PC146     PC147
## Standard deviation     2.763e-05 2.701e-05 2.626e-05 2.485e-05 2.438e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC148     PC149     PC150     PC151     PC152
## Standard deviation     2.355e-05 2.268e-05 2.204e-05 2.103e-05 2.069e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC153     PC154     PC155     PC156     PC157
## Standard deviation     1.965e-05 1.811e-05 1.728e-05 1.677e-05 1.615e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC158     PC159     PC160     PC161     PC162
## Standard deviation     1.511e-05 1.459e-05 1.377e-05 1.256e-05 1.129e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC163     PC164     PC165     PC166     PC167
## Standard deviation     1.068e-05 1.041e-05 9.886e-06 9.586e-06 8.968e-06
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                           PC168    PC169     PC170     PC171     PC172
## Standard deviation     8.27e-06 8.01e-06 7.691e-06 7.291e-06 6.712e-06
## Proportion of Variance 0.00e+00 0.00e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.00e+00 1.00e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC173     PC174    PC175     PC176     PC177
## Standard deviation     6.531e-06 5.785e-06 5.12e-06 4.997e-06 4.691e-06
## Proportion of Variance 0.000e+00 0.000e+00 0.00e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.00e+00 1.000e+00 1.000e+00
##                           PC178     PC179     PC180    PC181     PC182    PC183
## Standard deviation     4.63e-06 4.515e-06 4.076e-06 3.65e-06 3.408e-06 3.24e-06
## Proportion of Variance 0.00e+00 0.000e+00 0.000e+00 0.00e+00 0.000e+00 0.00e+00
## Cumulative Proportion  1.00e+00 1.000e+00 1.000e+00 1.00e+00 1.000e+00 1.00e+00
##                            PC184     PC185     PC186     PC187     PC188
## Standard deviation     2.983e-06 2.731e-06 2.577e-06 2.281e-06 2.143e-06
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC189    PC190     PC191     PC192    PC193    PC194
## Standard deviation     1.739e-06 1.63e-06 1.555e-06 1.488e-06 1.34e-06 1.28e-06
## Proportion of Variance 0.000e+00 0.00e+00 0.000e+00 0.000e+00 0.00e+00 0.00e+00
## Cumulative Proportion  1.000e+00 1.00e+00 1.000e+00 1.000e+00 1.00e+00 1.00e+00
##                            PC195     PC196     PC197     PC198     PC199
## Standard deviation     1.263e-06 1.175e-06 1.159e-06 9.114e-07 8.825e-07
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC200     PC201     PC202     PC203     PC204
## Standard deviation     8.275e-07 7.706e-07 7.068e-07 6.286e-07 5.447e-07
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC205     PC206     PC207     PC208     PC209
## Standard deviation     5.327e-07 4.714e-07 4.657e-07 4.339e-07 3.938e-07
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC210     PC211     PC212   PC213     PC214
## Standard deviation     3.638e-07 3.369e-07 3.073e-07 2.9e-07 2.771e-07
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.0e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.0e+00 1.000e+00
##                            PC215     PC216     PC217     PC218     PC219
## Standard deviation     2.629e-07 2.261e-07 2.056e-07 1.933e-07 1.804e-07
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC220     PC221     PC222     PC223     PC224
## Standard deviation     1.684e-07 1.453e-07 1.365e-07 1.249e-07 1.209e-07
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC225     PC226     PC227     PC228     PC229
## Standard deviation     1.091e-07 1.052e-07 1.006e-07 8.771e-08 8.387e-08
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC230     PC231     PC232     PC233     PC234
## Standard deviation     8.096e-08 7.064e-08 6.155e-08 5.847e-08 5.112e-08
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC235    PC236     PC237     PC238     PC239
## Standard deviation     4.848e-08 4.34e-08 4.111e-08 3.812e-08 3.624e-08
## Proportion of Variance 0.000e+00 0.00e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.00e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC240     PC241     PC242     PC243     PC244
## Standard deviation     3.373e-08 3.227e-08 3.042e-08 2.777e-08 2.759e-08
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC245     PC246     PC247     PC248     PC249
## Standard deviation     2.639e-08 2.427e-08 2.312e-08 2.047e-08 1.974e-08
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC250     PC251     PC252     PC253     PC254
## Standard deviation     1.888e-08 1.829e-08 1.663e-08 1.549e-08 1.472e-08
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC255     PC256     PC257     PC258    PC259
## Standard deviation     1.418e-08 1.282e-08 1.222e-08 1.213e-08 1.15e-08
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.00e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.00e+00
##                            PC260     PC261     PC262    PC263     PC264
## Standard deviation     1.087e-08 1.075e-08 9.559e-09 9.28e-09 8.989e-09
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.00e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.00e+00 1.000e+00
##                            PC265     PC266    PC267     PC268     PC269
## Standard deviation     8.819e-09 7.773e-09 7.33e-09 7.079e-09 6.712e-09
## Proportion of Variance 0.000e+00 0.000e+00 0.00e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.00e+00 1.000e+00 1.000e+00
##                            PC270     PC271     PC272     PC273     PC274
## Standard deviation     6.332e-09 6.054e-09 5.881e-09 5.527e-09 5.261e-09
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC275     PC276     PC277     PC278     PC279
## Standard deviation     4.764e-09 4.616e-09 4.153e-09 4.135e-09 3.913e-09
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC280     PC281     PC282     PC283     PC284
## Standard deviation     3.786e-09 3.641e-09 3.357e-09 3.344e-09 3.098e-09
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC285     PC286    PC287     PC288     PC289   PC290
## Standard deviation     2.907e-09 2.768e-09 2.55e-09 2.397e-09 2.151e-09 2.1e-09
## Proportion of Variance 0.000e+00 0.000e+00 0.00e+00 0.000e+00 0.000e+00 0.0e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.00e+00 1.000e+00 1.000e+00 1.0e+00
##                            PC291     PC292     PC293     PC294     PC295
## Standard deviation     2.003e-09 1.887e-09 1.764e-09 1.655e-09 1.611e-09
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC296     PC297     PC298     PC299     PC300
## Standard deviation     1.492e-09 1.308e-09 1.255e-09 1.124e-09 9.962e-10
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC301    PC302     PC303     PC304     PC305
## Standard deviation     9.839e-10 9.29e-10 6.812e-10 5.568e-17 3.053e-17
## Proportion of Variance 0.000e+00 0.00e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.00e+00 1.000e+00 1.000e+00 1.000e+00
```

The Summary contains the following data:

- *The standard deviation of each principal component (PC)*
- *The proportion of variance explained by each PC*
- *The cumulative proportion of variance explained by each PC*

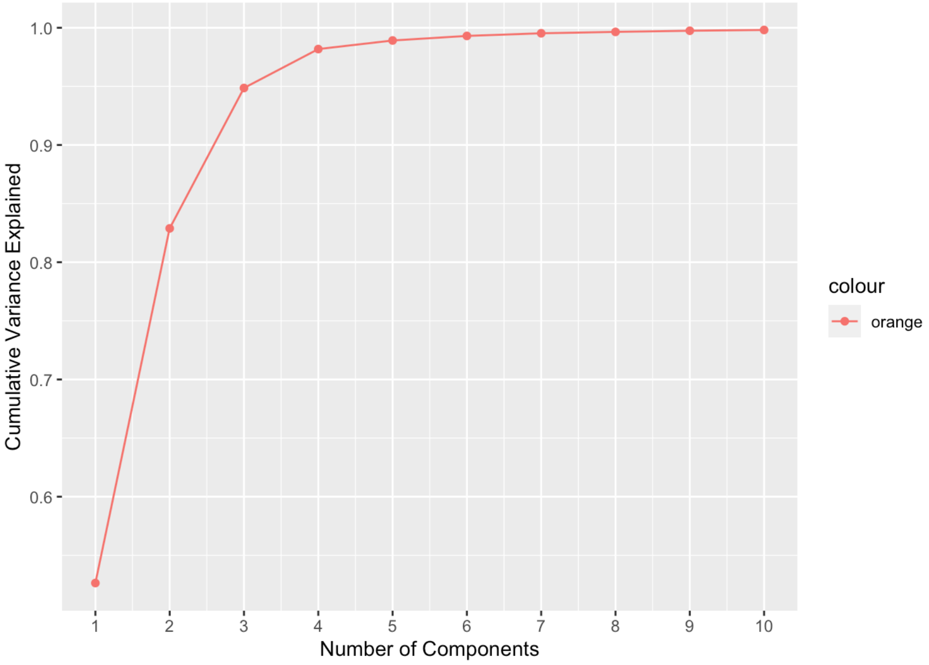
BarPlot of Principal Component Vs Variances:

```
#Plotting Variance with the Principal Component Variables
plot(PCAOfDataset,col = "red",xlab = "Principal Component",main = "BAR Plot of PCA Dataset")
```



The bar plot shown above shows the variance values in relation to each principal component. The variance decreases as we raise the principal components, reaching zero at the fourth component and remaining at zero at larger numbers.

```
#Calculating Standard Deviation Matrix
StandardDeviationMatrix = apply(SVar, 2, sd)
#Standardize the columns of the Standard Deviation Matrix
Dstandard = sweep(SVar, 2, StandardDeviationMatrix, "/")
#Performing PCA for given DataSet and Assigining to Variable
PCAOfDatasetstandard = prcomp(Dstandard)
#Performing PCA for given DataSet, scaling it and Assigining to Variable
PCAOfDatasetstandard <- prcomp(SVar, scale.=T)
#Calculating Variance
VariableValue <- PCAOfDatasetstandard$sdev[1:10]^2 / sum(PCAOfDatasetstandard$sdev^2)
#Calculating Cummulative Variance
CummulativeVariance <- cumsum(VariableValue)
#setting cummulative Variance and variance in the Dataframe
VarianceDataFrame <- data.frame( numOfcomponents = 1:length(VariableValue), cummulativeVarianceProportion = Cummu
lativeVariance)
#Plotting to tell the how many best components are required based on the cummulative variance
ggplot(VarianceDataFrame, aes(x=numOfcomponents, y=cummulativeVarianceProportion, color = "orange")) +
  geom_line() +
  geom_point() +
  scale_x_continuous(breaks = seq(1, length(VariableValue), by = 1)) +
  labs(x="Number of Components", y="Cumulative Variance Explained")
```



So, it is evident from the graphic that we need 4 principal components in order to achieve the best cumulative variance of almost 95%. As the number of components is increased, the cumulative variance remains constant.

===

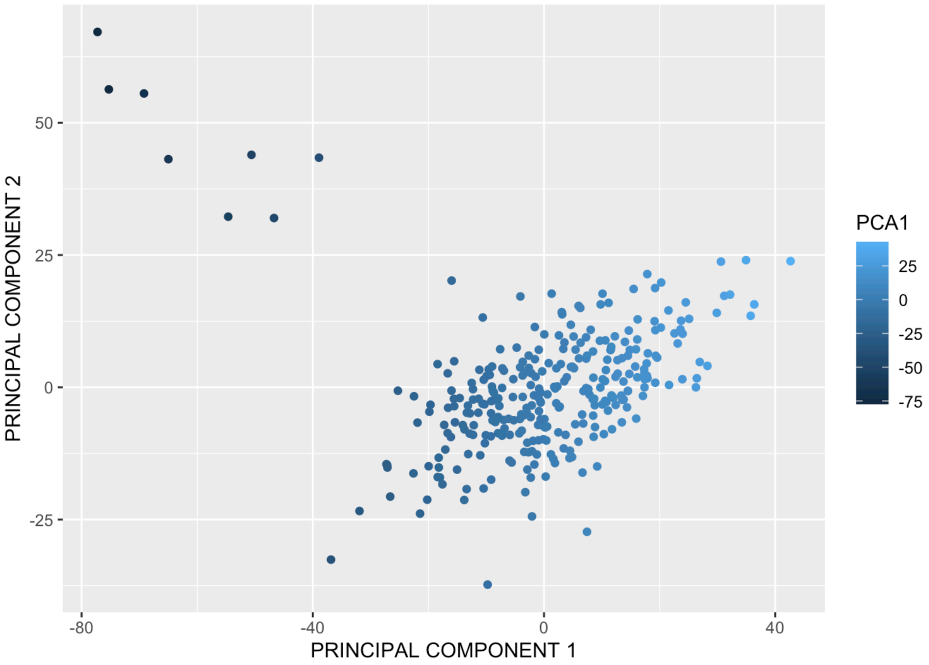
5. Derive the principal component scores for the milk samples from first principles (i.e., you should not use an inbuilt function such as predict(. . .)). Plot the principal component scores for the milk samples. Comment on any structure you observe. [15 marks]

===

```
#Scaling of the DataSet
Standardiseddata = scale(SVar)
#Covariance Matrix of the above data
CovarianceMatrix = cov(Standardiseddata)
#Calculating the Eigen values of the Covariance Matrix
EigenValuesOfData = eigen(CovarianceMatrix)
#Calculating the Proportion of the variance
ProportionOfVariance = EigenValuesOfData$values / sum(EigenValuesOfData$values)
#Storing the data of vectors in a variable
PCAComponents <- EigenValuesOfData$vectors[, 1:4]
#Calculating PCA score
PCAScore <- Standardiseddata %*% PCAComponents
head(PCAScore)
```

```
##           [,1]        [,2]      [,3]      [,4]
## 1 -9.562733   3.3933685 -1.358185 1.8037476
## 2 -8.872213  -0.6482674 -5.038227 0.1183923
## 3 -3.854645  -8.1926793 -2.550892 1.4931299
## 4 -2.285472   0.4160081  1.113284 1.7659712
## 5 -1.668417 -14.5899250 -5.620147 4.5564727
## 6 -7.199986   3.5836695 -1.860449 0.4525771
```

```
#Plotting the PCA 1 with PCA 2
df <- data.frame(PCAScore[, 1], PCAScore[, 2])
#Plot the first two principal components
ggplot(df, aes(x = PCAScore[, 1], y = PCAScore[, 2],color = df$PCAScore...1.)) +
  geom_point() +
  labs(x = "PRINCIPAL COMPONENT 1", y = "PRINCIPAL COMPONENT 2",color="PCA1")
```



Principal component analysis (PCA) is used to scale the data, compute the covariance matrix, and then identify the eigenvalues and eigenvectors of the covariance matrix on a dataset (SVar). The proportion of variation that each basic component contributes to being explained is also estimated. In the output, the first principal component's values are used to color the points in the range of values from -75 to 25, and the result provides a scatter plot of the first two principal components. The majority of the scatter points fall between -40 and 40.

==

6. Interest lies in predicting the β Lactoglobulin B trait based on the MIR spectra. Principal components regression (PCR) is one approach to doing so for such n < p data. Research the principal components regression method and how it works e.g., see An Introduction to Statistical Learning with Applications in R by James et al. (2021), The Elements of Statistical Learning by Hastie et al. (2017), and/or the peer-reviewed journal article The pls Package: Principal Component and Partial Least Squares Regression in R by Mevik and Wehrens (2007).In your own words, write a maximum 1 page synopsis of the PCR method. Your synopsis should (i) explain the method's purpose, (ii) provide a general description of how the method works, (iii) detail any choices that need to be made when using the method and (iv) outline the advantages and disadvantages of the method. [30 marks]

==

(i) explain the method's purpose

==

When there are more predictors than observations, Principal Component Regression (PCR) is used to model the relationship between a collection of predictor variables and a response variable. PCR accomplishes this by decreasing the number of dimensions in the data while maintaining a high level of variation in the predictor factors.For dealing with high-dimensional data, such as the MIR spectra data in the provided dataset, where there are numerous predictor variables (531 MIR spectra variables) and few observations, PCR is especially helpful. When modeling highly dimensional data, problems with overfitting and multicollinearity can occur. The reduction in dimensionality achieved using PCR helps to solve these problems.In PCR, the predictor variables are broken down into a series of principal components, which are orthogonal to one another and a linear combination of the original variables. These principle components serve as predictors in a multivariate linear regression model by capturing the most significant patterns in the data. Cross-validation is used to decide how many primary components to include in the regression model. The main goal of PCR is to provide a more simplified model that is simpler to understand and more general to new data. Several fields where high-dimensional data is prevalent, like genetics, chemometrics, and finance, can benefit from the usage of PCR.

(ii) provide a general description of how the method works

==

The two-step procedure of Principal Component Regression (PCR) entails:

- **Principal Component Analysis (PCA):** To minimize the dimensionality of the data, PCA is first applied to the predictor variables (MIR spectra in the provided dataset). The original variables are converted into a set of main components through PCA, which eliminates any redundant data and captures the most significant patterns in the data. The original variables are combined linearly into the primary components, which are orthogonal to one another. The first five components often account for the majority of the variation in the data since the principle components are ranked according to how much variance in the data they capture.

- **Regression:** In the second phase, a multiple linear regression model is constructed with the target response variable (in this example, the Lactoglobulin B characteristic) as the outcome variable and the principal components as predictors. Using cross-validation approaches, such as leave-one-out cross-validation, the model's primary component count is computed. Measures like the coefficient of determination or the root mean square error (RMSE) are used to assess the model's performance (R-squared).

With PCR, a type of dimensionality reduction, high-dimensional data can be made simpler while still preserving the key patterns. PCR can handle instances when there are more predictors than observations and can assist in resolving problems with multicollinearity and overfitting that can occur in high-dimensional data by employing the principal components as predictors in a regression model.

(iii) detail any choices that need to be made when using the method

==

While utilizing Principal Component Regression (PCR), various decisions must be made:

- **Principal component count:** When employing PCR, it is crucial to decide how many principal components should be included in the regression model. An underfit model can be produced by having too few main components, and an overfit model can be produced by including too many principal components. Cross-validation methods, such as leave-one-out cross-validation or k-fold cross-validation, are frequently used to establish the ideal number of principal components.

- **Scaling of variables:** While utilizing PCR, it's crucial to consider the predictor variables' scaling. Scaling makes guarantee that the main component analysis is not dominated by variables with higher variances. To scale the variables, standardization or normalization is frequently utilized.

- **Response variable selection:** The important response variable needs to be chosen with attention. It must have a significant relationship with the predictor variables and be a continuous variable.

- **Outlier identification and correction:** The outcomes of PCR can be significantly impacted by data outliers. Thus, it's crucial to identify outliers and handle them correctly. Outlier detection methods, such Mahalanobis distance or Cook's distance, and outlier treatment methods, like removal or imputation, can be used to do this.

- **Results interpretation:** Because the primary components are frequently tricky to understand, it might be difficult to interpret the results of PCR. Determining the loadings of the principal components and interpreting them in light of the initial predictor variables is crucial.

Careful consideration must be given to these options to guarantee that the PCR model is suitable for the available data and produces accurate results.

(iv) outline the advantages and disadvantages of the method

==

Principal Component Regression (PCR) has several benefits, including:

- **Dimensionality reduction:** By locating the most crucial patterns in the data and eliminating any extraneous information, PCR helps to reduce the dimensionality of high-dimensional data. Building models and interpreting the outcomes are made simpler as a result.

- **Regression models:** Its performance can be improved by minimizing multicollinearity and overfitting problems by reducing the dimensionality of the data, which can be achieved by PCR.

- **Simple to implement:** PCR may be carried out using common statistical tools and is comparatively simple to execute.

- **Flexibility:** PCR can be applied with any kind of linear regression model and a large variety of data formats.

Principal Component Regression (PCR) has several drawbacks.

- **Interpretation:** It might be tricky to interpret the main components produced by PCA, which can make it difficult to evaluate the regression model's results.

- **Overfitting:** When there are too many primary components in a model, overfitting can happen, which makes the model less generalizable to new data.

- **Information loss:** As PCA concentrates primarily on the most significant patterns in the data, it may cause some information to be lost.

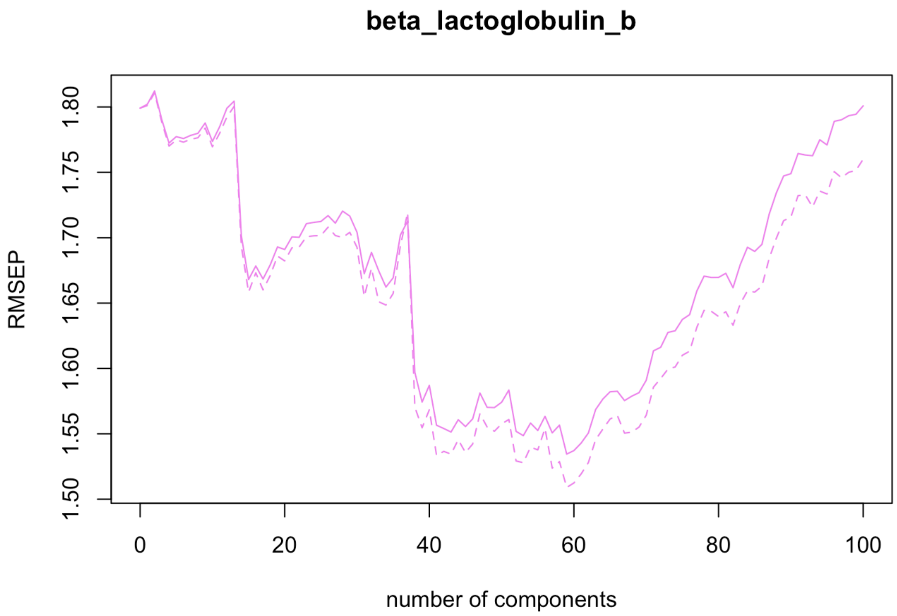
- **Pre-processing requirements:** Data must be pre-processed before PCR can be performed. This pre-processing may involve scaling, centering, and other processes that increase the complexity of the analysis.

==

7. Use the function pcr in the pls R package to use PCR to predict the β Lactoglobulin B levels from the spectra for a test set, where the test set is one third of the data. Motivate any decisions you make. [25 marks]

==

```
#Package is used for library of functions for performing Partial Least Squares (PLS) and Principal Component Regr
ession (PCR)
library (pls)
#Assign to a Varibles
PrinCompRegressDataSet=data.frame(MilkDataFrame[,"beta_lactoglobulin_b"],MilkDataFrame[,-c(1:51)])
colnames(PrinCompRegressDataSet)[1]="beta_lactoglobulin_b"
DataSize <- floor(nrow(PrinCompRegressDataSet)*0.75)
SizeVal <- floor(nrow(PrinCompRegressDataSet)*0.25)+1
TrDataIdx=sample(1:nrow(PrinCompRegressDataSet), size = DataSize)
TestIdx = setdiff(1:nrow(PrinCompRegressDataSet), TrDataIdx)
#Fitting Principal Component Regression Model
PriCompRegFit <- pcr (beta_lactoglobulin_b ~., data = PrinCompRegressDataSet , subset = TrDataIdx ,scale = TRUE ,
ncomp = 100,  validation = "CV")
#Visualize performance of the regression in the plot
validationplot (PriCompRegFit , val.type = "RMSE",col ="violet")
```



```
#Generate Predicted Values
PriCompRegPred <- predict (PriCompRegFit , PrinCompRegressDataSet[TestIdx , -c(1)], ncomp =4 )
#Calculate Mean
mean ((PriCompRegPred -PrinCompRegressDataSet[TestIdx ,1] )^2)
```

```
## [1] 2.25043
```

The Mean after calculating, it is given by 2.25043

==

8. Seven milk proteins, one of which is β Lactoglobulin B, are important for the production of cheese and whey (see invited lecture slides). Here, for some records/observations the β Lactoglobulin B values are exactly 0, while there are non-zero values for the other milk proteins for the same records. Often records with such strange measurements are deleted, arguably losing information. Here, rather than delete these observations, the β Lactoglobulin B values of 0 could be treated as 'missing at random'. Often such missing values are imputed using e.g., the mean of the observed β Lactoglobulin B values. In the multivariate setting, matrix completion methods can be used to impute such missing at random values. (Note that matrix completion approaches are often used to power recommender systems such as Netflix.). One matrix completion method uses principal components analysis as detailed in section 12.3 in An Introduction to Statistical Learning with Applications in R by James et al. (2021). Read this section to understand how the method works. Write your own code to impute the β Lactoglobulin B values that are 0 using principal components analysis on the seven milk proteins data. You must use the function prcomp or eigen in your solution. Comment on the results you obtain. [30 marks]

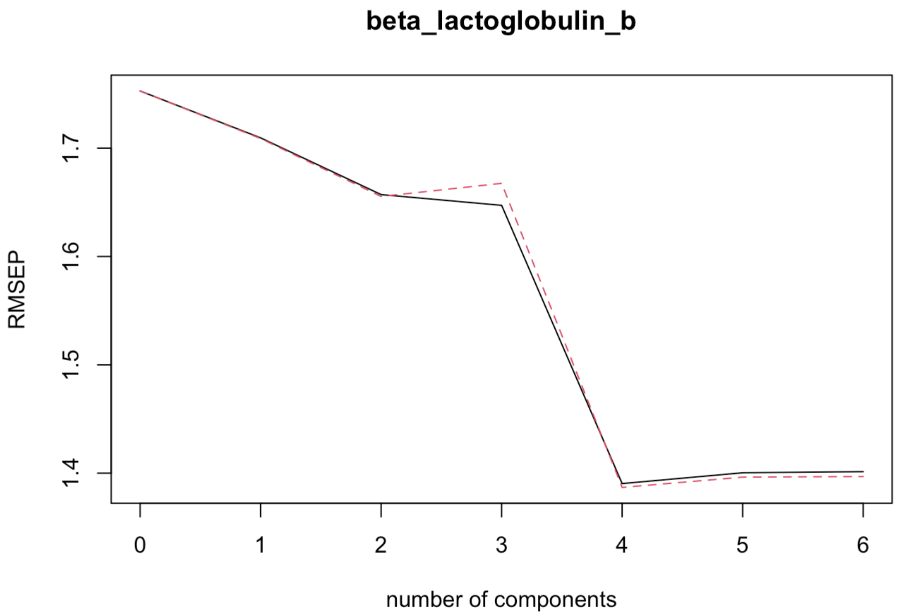
==

```
#Extract data and create Data Frame
ProData=MilkDataFrame[,7:13]
#Calculating Standard Deviation
StandDev = apply(ProData, 2, sd)
#dividing each element by its corresponding standard deviation
ProDataStandardValue = sweep(ProData, 2, StandDev, "/")
#perform principal component analysis
PrincCompAnalyProtien=prcomp(ProDataStandardValue)
#Printing summary
summary(PrincCompAnalyProtien)
```

```
## Importance of components:
##                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.921 1.1312 0.9498 0.69120 0.58290 0.45168 0.32852
## Proportion of Variance 0.527 0.1828 0.1289 0.06825 0.04854 0.02915 0.01542
## Cumulative Proportion  0.527 0.7098 0.8387 0.90690 0.95544 0.98458 1.00000
```

The principal component analysis (PCA) of the protein data in the MilkDataFrame is described in the summary. It displays the proportion of variance explained by each principal component (PC) as well as its standard deviation. The first PC, followed by the second (18.28%) and third (12.89%) PCs, accounts for 52.7% of the variance in the data. How much of the overall variation in the data is explained by each succeeding PC is indicated by the cumulative proportion of variance. The variance in this situation is completely explained by the first seven PCs.

```
#Validation plot using CV
PrinCompRegModelFitX <- pcr (beta_lactoglobulin_b ~., data = ProData ,validation = "CV")
validationplot (PrinCompRegModelFitX , val.type = "RMSE")
```



The figure shows that the mean squared error is lowest when there are six basic components. The largest amount of variance, up to 98 percent of variance, is accounted for by 6 primary components working together. Thus, the matrix completion method is chosen to have 6 main components.

```
#Fits SVD Model
SVDFitModel<-function (Z, B ) {
 Object <- prcomp(Z)
 Result= Object$x%*%Object$rotation
  return (Result)
}

#substituting the 0 values to na
ProData[ProData == 0] <- NA
# changing all the Na values to column mean
CXhat <- ProData
SxBar <- colMeans (ProData , na.rm = TRUE)
CXhat[is.na(CXhat)] <- SxBar[7]

# Setting the initall values
ThrsholdValue <- 1e-7
RelativeErrorValue <- 1
IterationValue <- 0
MissOrNotValue <- is.na(ProData)
# Calculating the MSE with data which has mean values replaced
MsSoldValue <- mean (( scale (ProData , SxBar , FALSE)[!MissOrNotValue])^2)
MsSoValue <- mean (ProData[!MissOrNotValue]^2)

while (RelativeErrorValue > ThrsholdValue) {
IterationValue <- IterationValue + 1
# Step 2(a)
#Updating the value
XAppValue <- SVDFitModel (CXhat , B = 6)
# Step 2(b)
# Updating the values with predicted scores
CXhat[MissOrNotValue] <- XAppValue[MissOrNotValue]
 # Step 2(c)
# Calculating the relative error
MSSValue <- mean (((ProData - XAppValue)[!MissOrNotValue])^2)
RelativeErrorValue <- (MsSoldValue -MSSValue ) / MsSoValue
# setting the current values as old
MsSoldValue <- MSSValue }
```

In order to impute the missing values in beta-lactoglobulin B, we used a matrix completion method. The approach tries to reduce the relative error (mssold-mss/ms0) first, and then imputes the missing values using the values with the lowest relative error (below a certain threshold). To fill in the gaps in the data, we generated score vectors using prcomp, where the scores are nothing more than the projection of additional points onto the principal axis. According to the results, the relative error has decreased to a level below which new values are assigned to the missing data.

==

9. Using PCR, predict the β Lactoglobulin B values from the MIR spectra for a test set where the training set contains:

(a) all records with an observed, non-zero value of β Lactoglobulin B.

(b) all records but where 0 values of β Lactoglobulin B are imputed using the observed mean.

(c) all records but where 0 values of β Lactoglobulin B values are imputed using principal components analysis.

Comment on what you observe. [30 marks]

==

9(a)

```
#data division into training and test data
MilkDataFrameWithNonZeroTrain=data.frame(beta_lactoglobulin_b=MilkDataFrame[TrDataIdx,"beta_lactoglobulin_b"],MilkDataFrame[TrDataIdx,-c(1:51)])
MilkDataFrameWithNonZeroTest=data.frame(beta_lactoglobulin_b=MilkDataFrame[TestIdx,"beta_lactoglobulin_b"],MilkDataFrame[TestIdx,-c(1:51)])
TrainOfX=MilkDataFrameWithNonZeroTrain[,-c(1)]
TrainOfY=MilkDataFrameWithNonZeroTrain[,1]
TestOfX=MilkDataFrameWithNonZeroTest[,-c(1)]
TestOfY=MilkDataFrameWithNonZeroTest[,1]
```

```
#fits a principal component regression (PCR) model
PrinCompRegFitNonZero<- pcr(beta_lactoglobulin_b ~., data = MilkDataFrameWithNonZeroTrain  ,
scale = TRUE ,  ncomp = 42)
#generate predicted values
PrinCompRegPredNonZero <- predict (PrinCompRegFitNonZero , TestOfX, ncomp =40 )
#Calculate Mean Value
mean ((PrinCompRegPredNonZero -TestOfY )^2)
```

```
## [1] 2.357818
```

The beta globulin levels are predicted after the model is fitted to a dataset comprising some observations of beta globulin levels as 0. The validation plot, which was previously used, yields the optimal number of main components and has the lowest RMSE. 42 fundamental components were found. - The model is fitted using 44 major components to compare the expected and actual betalactoglobulin levels. The mean squared error in this instance was 2.357818, which is quite high.

==

9(b)

```
#Assigning Variables
MilkDataFrameMean=MilkDataFrame
ValueOfMean <- mean (MilkDataFrameMean$beta_lactoglobulin_b )
MilkDataFrameMean[MilkDataFrameMean == 0] <- ValueOfMean
#data division into training and test data
ValueOfMeanTrain=data.frame(beta_lactoglobulin_b=MilkDataFrameMean[TrDataIdx,"beta_lactoglobulin_b"],MilkDataFram
eMean[TrDataIdx,-c(1:51)])
ValueOfMeanTest=data.frame(beta_lactoglobulin_b=MilkDataFrameMean[TestIdx,"beta_lactoglobulin_b"],MilkDataFrameMe
an[TestIdx,-c(1:51)])
TestOfX=ValueOfMeanTest[,-c(1)]
TestOfY=ValueOfMeanTest[,1]
```

```
#fits a principal component regression (PCR) model
PrinCompRegFitMean<- pcr (beta_lactoglobulin_b ~., data = MilkDataFrameWithNonZeroTrain  ,
scale = TRUE ,  ncomp = 42)
#generate predicted values
PrinCompRegPredMean <- predict (PrinCompRegFitMean , TestOfX, ncomp =40 )
#Calculate Mean Value
mean ((PrinCompRegPredMean -TestOfY )^2)
```

```
## [1] 1.928521
```

The model is fitted, and the beta globulin levels are predicted, when mean values are used to replace some observations of beta globulin levels as 0. The validation plot, which was previously used, yields the optimal number of main components and has the lowest RMSE. 42 fundamental components were found. The model is fitted using 42 major components to compare the expected and actual betalactoglobulin levels. - The mean squared error in this instance was 1.928521, which is considerably less than the MSE of the data that included 0 values.

==

9(c)

```
ValueOfy=CXhat$beta_lactoglobulin_b
#data division into training and test data
MilkDataFramePCATrain=data.frame(beta_lactoglobulin_b=ValueOfy[TrDataIdx],MilkDataFrameMean[TrDataIdx,-c(1:51)])
ValueOfMeanTest=data.frame(beta_lactoglobulin_b=ValueOfy[TestIdx],MilkDataFrameMean[TestIdx,-c(1:51)])
TestOfX=ValueOfMeanTest[,-c(1)]
TestOfY=ValueOfMeanTest[,1]
```

```
#fits a principal component regression (PCR) model
PrinCompRegFitPCR<- pcr (beta_lactoglobulin_b ~., data = ValueOfMeanTest  ,
scale = TRUE ,  ncomp = 42)
#generate predicted values
PrinCompRegPredPCR <- predict (PrinCompRegFitPCR , TestOfX, ncomp =40 )
#Calculate Mean Value
mean ((PrinCompRegPredPCR -TestOfY )^2)
```

```
## [1] 1.248093
```

Using a dataset with certain observations of beta globulin levels that were imputed as 0, the matrix completion approach is used to forecast beta globulin b values. - The validation plot, which was previously used, yields the optimal number of main components and has the lowest RMSE. 42 fundamental components were found. - The model is fitted with 42 essential components to compare the expected and actual beta lactoglobulin b values. - In this case, the dataset's mean squared error was 1.248093, which was less than the mean squared errors of the datasets with the mean substituted for the 0 values and the data set with 0 values.

==

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.