ashishsa_hw2_p2

Load the dataset.

```
df1 <- read.csv("Ch10Ex11.csv", header = FALSE)
head(df1)
```

```
##            V1         V2         V3         V4         V5         V6
## 1 -0.96193340  0.4418028 -0.9750051  1.4175040  0.8188148  0.3162937
## 2 -0.29252570 -1.1392670  0.1958370 -1.2811210 -0.2514393  2.5119970
## 3  0.25878820 -0.9728448  0.5884858 -0.8002581 -1.8203980 -2.0589240
## 4 -1.15213200 -2.2131680 -0.8615249  0.6309253  0.9517719 -1.1657240
## 5  0.19578280  0.5933059  0.2829921  0.2471472  1.9786680 -0.8710180
## 6  0.03012394 -0.6910143 -0.4034258 -0.7298590 -0.3640986  1.1253490
##            V7          V8          V9        V10        V11        V12
## 1 -0.02496682 -0.06396600  0.03149702 -0.3503106 -0.7227299 -0.2819547
## 2 -0.92220620  0.05954277 -1.40964500 -0.6567122 -0.1157652  0.8259783
## 3 -0.06476437  1.59212400 -0.17311700 -0.1210874 -0.1875790 -1.5001630
## 4 -0.39155860  1.06361900 -0.35000900 -1.4890580 -0.2432189 -0.4330340
## 5 -0.98971500 -1.03225300 -1.10965400 -0.3851423  1.6509570 -1.7449090
## 6 -1.40404100 -0.80613040 -1.23792400  0.5776018 -0.2720642  2.1765620
##           V13         V14        V15        V16        V17        V18
## 1  1.33751500  0.70197980  1.0076160 -0.4653828  0.6385951  0.2867807
## 2  0.34644960 -0.56954860 -0.1315365  0.6902290 -0.9090382  1.3026420
## 3 -1.22873700  0.85598900  1.2498550 -0.8980815  0.8702058 -0.2252529
## 4 -0.03879128 -0.05789677 -1.3977620 -0.1561871 -2.7359820  0.7756169
## 5 -0.37888530 -0.67982610 -2.1315840 -0.2301718  0.4661243 -1.8004490
## 6  1.43640700 -1.02578100  0.2981582 -0.5559659  0.2046529 -1.1916480
##          V19         V20        V21        V22        V23        V24
## 1 -0.2270782 -0.22004520 -1.2425730 -0.1085056 -1.8642620 -0.5005122
## 2 -1.6726950 -0.52550400  0.7979700 -0.6897930  0.8995305  0.4285812
## 3  0.4502892  0.55144040  0.1462943  0.1297400  1.3042290 -1.6619080
## 4  0.6141562  2.01919400  1.0811390 -1.0766180 -0.2434181  0.5134822
## 5  0.6262904 -0.09772305 -0.2997108 -0.5295591 -2.0235670 -0.5108402
## 6  0.2350916  0.67096470  0.1307988  1.0689940  1.2309870  1.1344690
##           V25         V26        V27        V28         V29         V30
## 1 -1.32500800  1.06341100 -0.2963712 -0.1216457  0.08516605  0.62417640
## 2 -0.67611410 -0.53409490 -1.7325070 -1.6034470 -1.08362000  0.03342185
## 3 -1.63037600 -0.07742528  1.3061820  0.7926002  1.55946500 -0.68851160
## 4 -0.51285780  2.55167600 -2.3143010 -1.2764700 -1.22927100  1.43439600
## 5  0.04600274  1.26803000 -0.7439868  0.2231319  0.85846280  0.27472610
## 6  0.55636800 -0.35876640  1.0798650 -0.2064905 -0.00616453  0.16425470
##          V31          V32         V33        V34        V35        V36
## 1 -0.5095915 -0.216725500 -0.05550597 -0.4844491 -0.5215811  1.9491350
## 2  1.7007080  0.007289556  0.09906234  0.5638533 -0.2572752 -0.5817805
## 3 -0.6154720  0.009999363  0.94581000 -0.3185212 -0.1178895  0.6213662
## 4 -0.2842774  0.198945600 -0.09183320  0.3496279 -0.2989097  1.5136960
## 5 -0.6929984 -0.845707200 -0.17749680 -0.1664908  1.4831550 -1.6879460
## 6  1.1567370  0.241774500  0.08863952  0.1829540  0.9426771 -0.2096004
```

```
##           V37        V38         V39        V40
## 1  1.32433500  0.4681471  1.06110000  1.6559700
## 2 -0.16988710 -0.5423036  0.31293890 -1.2843770
## 3 -0.07076396  0.4016818 -0.01622713 -0.5265532
## 4  0.67118470  0.0108553 -1.04368900  1.6252750
## 5 -0.14142960  0.2007785 -0.67594210  2.2206110
## 6  0.53626210 -1.1852260 -0.42274760  0.6243603
```

```
nrow(df1)
```

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

```
ncol(df1)
```

```
## [1] 40
```

We observe that there is 1000 Rows and 40 Columns. Also we observe that as this is a Genomic Dataset and the general structure as observed from head tells us that the general method of distance like Eucledian Distance or Manhatten Distance is unsuitable for calculating the Uncertanity in the dataset since the dataset contains Negative Values. So we use the Correlation Coefficient to calculate the Uncertanity between the data points.

```
distance1 <- dist(cor(df1))
```

We now plot the dendrogram by using various types of linkages ad compare the results of Single, Complete, Average and Centroid Based Linkage Methods and check the results for each of these methods.

We use the ggplot package to plot the Dendrogram.

```
library(philentropy)
```

```
## Warning: package 'philentropy' was built under R version 3.6.3
```

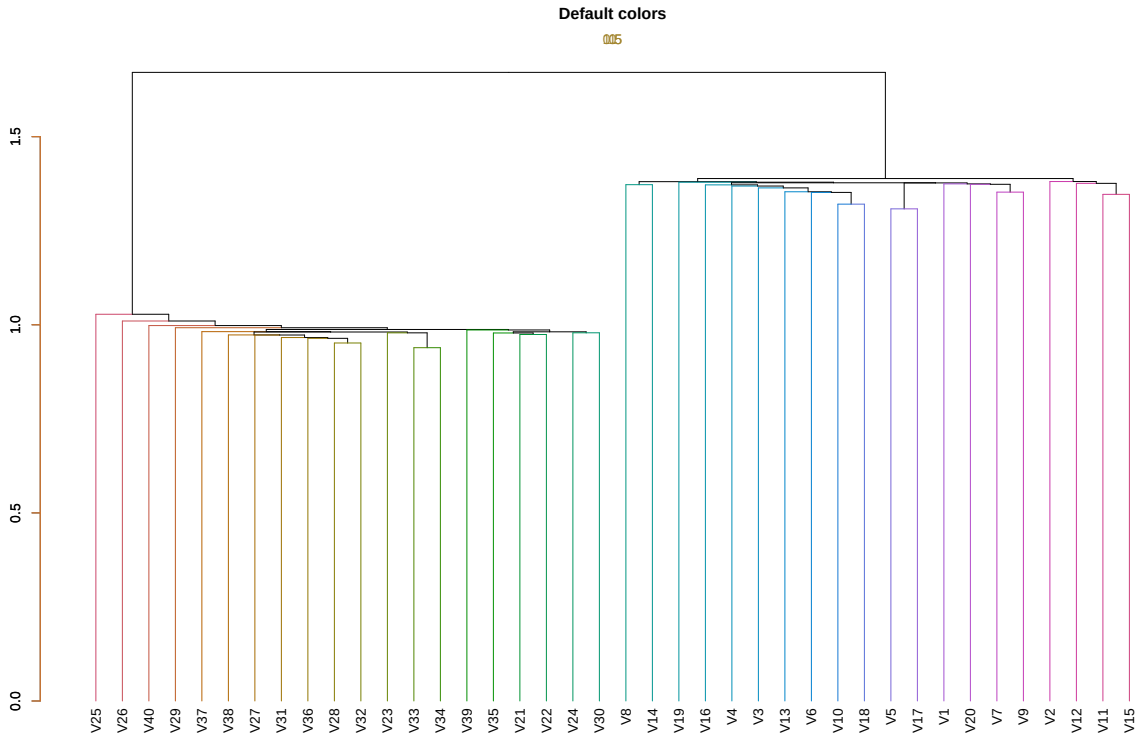
```
library(ggplot2)
library(ggdendro)
```

```
## Warning: package 'ggdendro' was built under R version 3.6.3
```

```
library(dplyr)
library(dendextend)
```

```
## Warning: package 'dendextend' was built under R version 3.6.3
```

```
hclust1 <- hclust(distance1,method="single")
dend1 <- as.dendrogram(hclust1)
dend1 %>% set("branches_k_color") %>%
  plot(main = "Default colors") %>%
  axis(side = 2,col = "#F38630",labels = TRUE) %>%
  mtext(col = "#A38630")
```

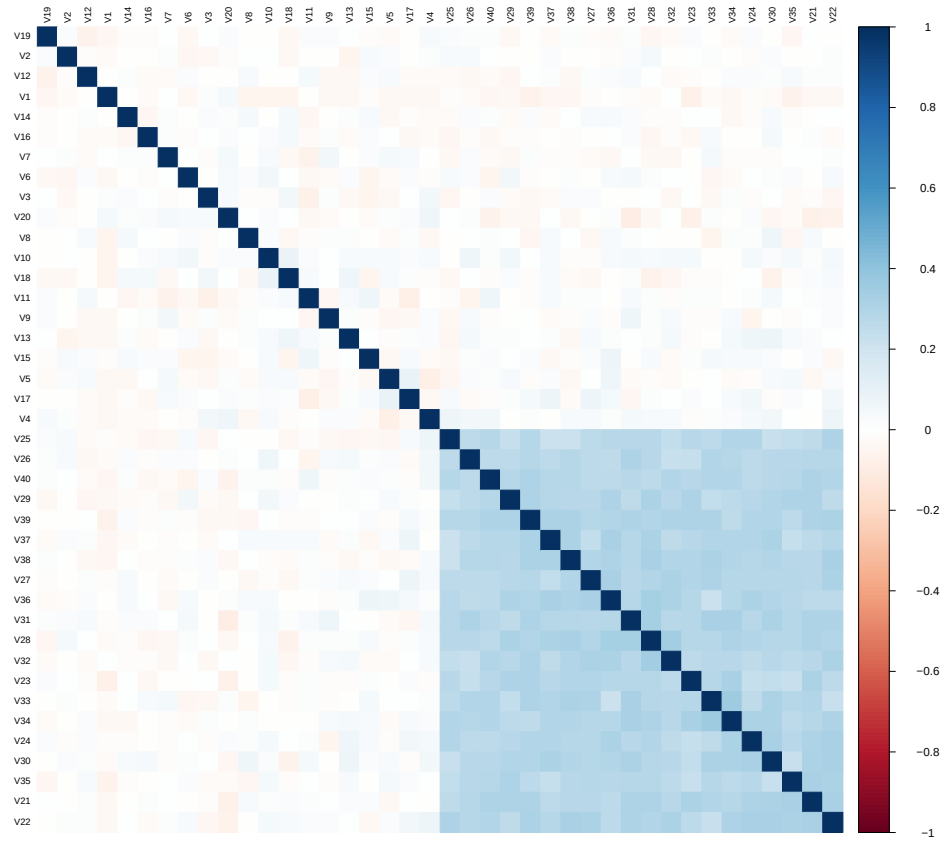



```
library(corrplot)
```

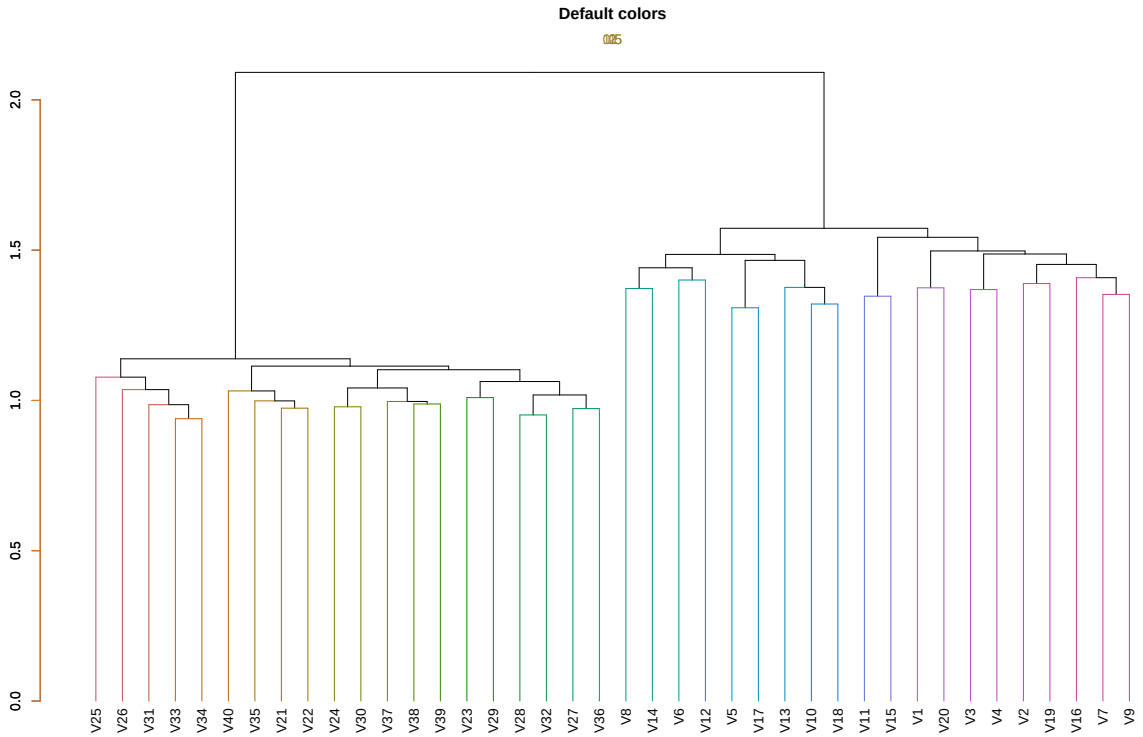
```
## Warning: package 'corrplot' was built under R version 3.6.3
```

```
## corrplot 0.84 loaded
```

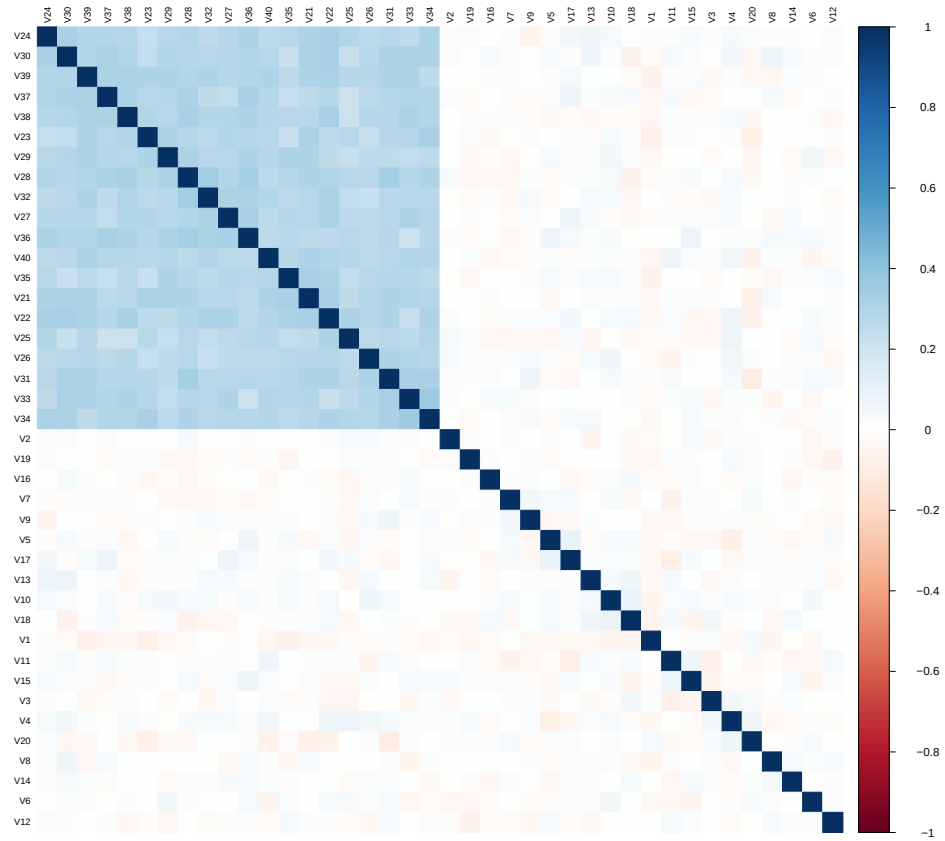
```
corrplot(cor(df1),method='color',order="hclust",hclust.method = 'single',tl.col = 'black', tl.cex = 0.7
```



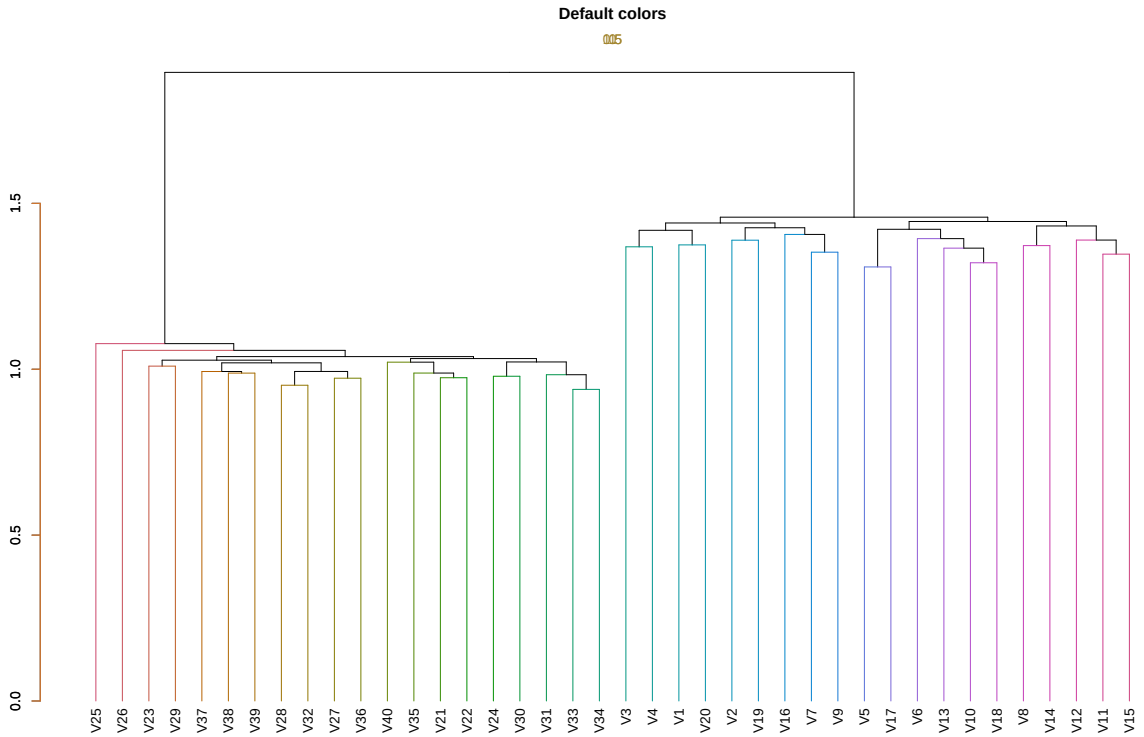

```
hclust2 <- hclust(distance1,method="complete")
dend2 <- as.dendrogram(hclust2)
dend2 %>% set("branches_k_color") %>%
  plot(main = "Default colors") %>%
  axis(side = 2,col = "#F38630",labels = TRUE) %>%
  mtext(col = "#A38630")
```



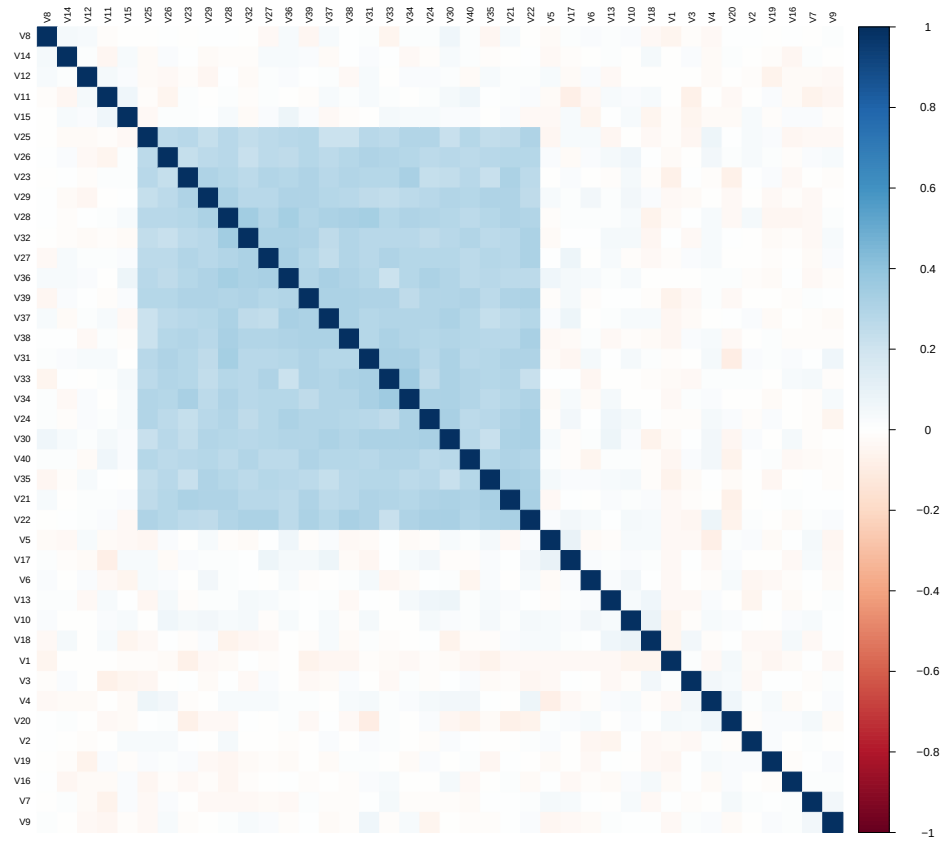

```
corrplot(cor(df1),method='color',order="hclust",hclust.method = 'complete',tl.col = 'black', tl.cex = 0
```



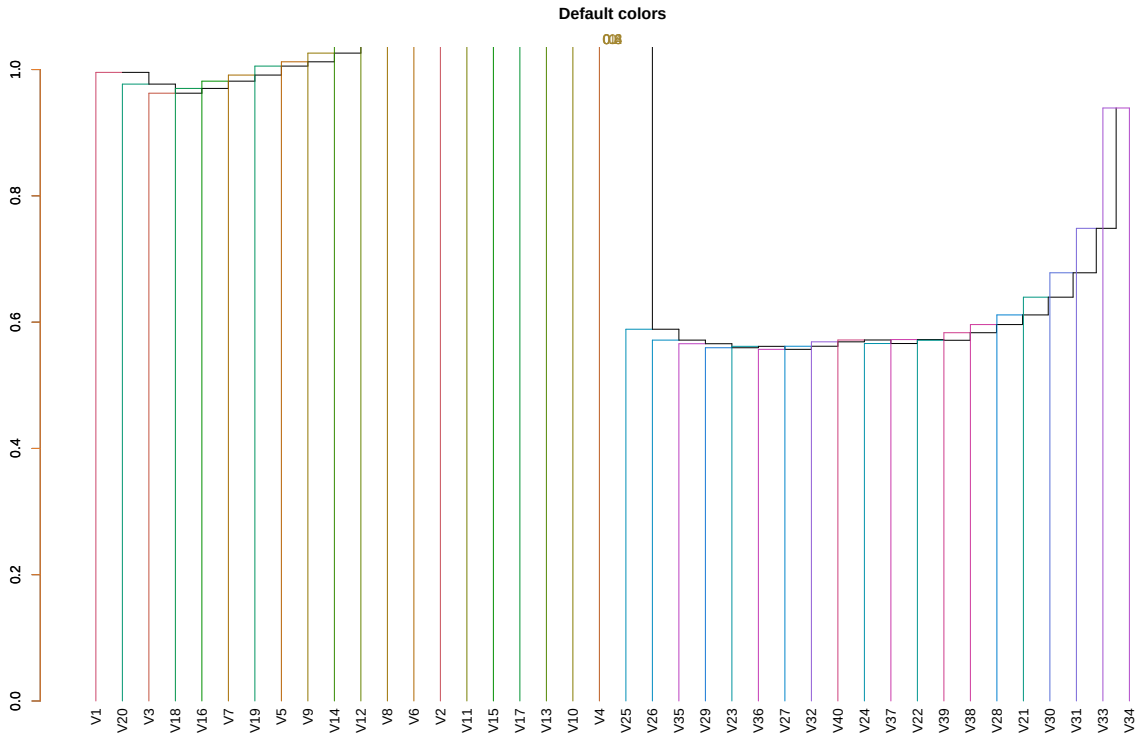

```
hclust3 <- hclust(distance1,method="average")
dend3 <- as.dendrogram(hclust3)
dend3 %>% set("branches_k_color") %>%
  plot(main = "Default colors") %>%
  axis(side = 2,col = "#F38630",labels = TRUE) %>%
  mtext(col = "#A38630")
```



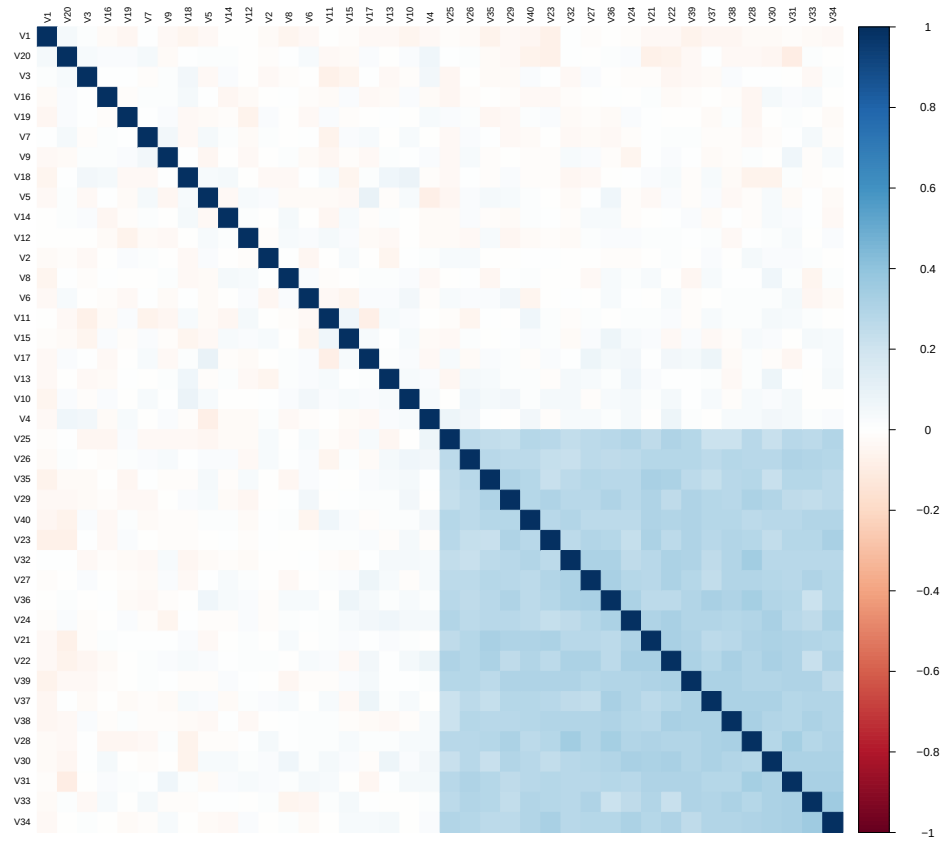

```
corrplot(cor(df1),method='color',order="hclust",hclust.method = 'average',tl.col = 'black', tl.cex = 0.
```




```
hclust4 <- hclust(distance1,method="centroid")
dend4 <- as.dendrogram(hclust4)
dend4 %>% set("branches_k_color") %>%
  plot(main = "Default colors") %>%
  axis(side = 2,col = "#F38630",labels = TRUE) %>%
  mtext(col = "#A38630")
```




```
corrplot(cor(df1),method='color',order="hclust",hclust.method = 'centroid',tl.col = 'black', tl.cex = 0
```



From the Visual Observation we observe that the Dataset is divided into 2 groups. So we divide the dataset into two groups.

```
cutree1 <- cutree(dend1,2)
cutree1
```

```
##  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
```

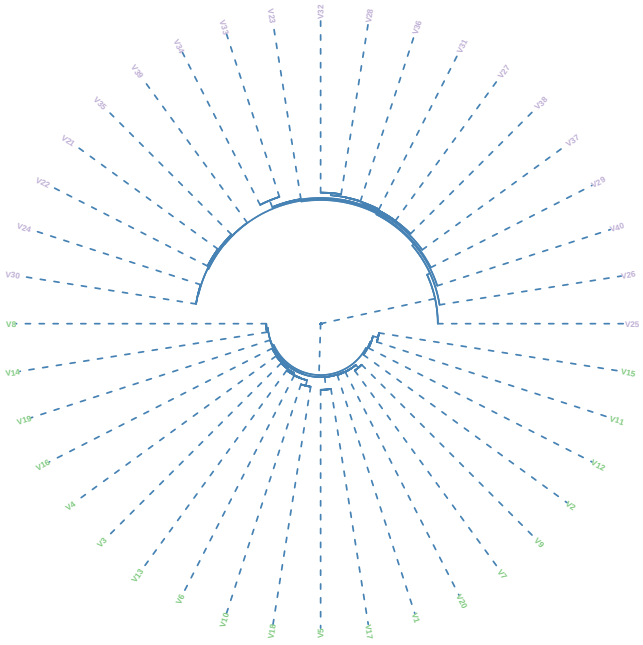
```
library(RColorBrewer)
library(ape)
```

```
## Warning: package 'ape' was built under R version 3.6.3
```

```
##
## Attaching package: 'ape'
```

```
## The following objects are masked from 'package:dendextend':
##
##     ladderize, rotate
```

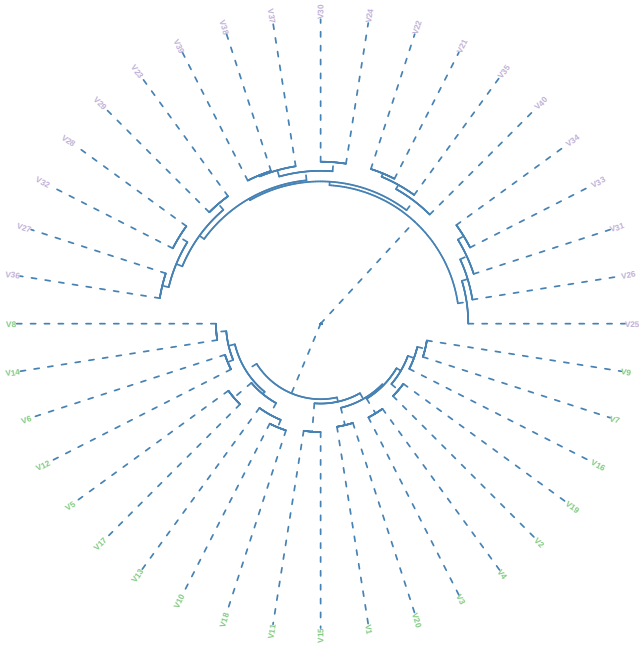
```
plot(as.phylo(hclust1), type = "fan", cex = 0.6,
    tip.color = brewer.pal(3, 'Accent')[cutree1],
     font = 2,
     edge.color = 'steelblue', edge.width = 2, edge.lty = 2)
```



```
cutree2 <- cutree(dend2,2)
cutree2
```

```
##  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
```

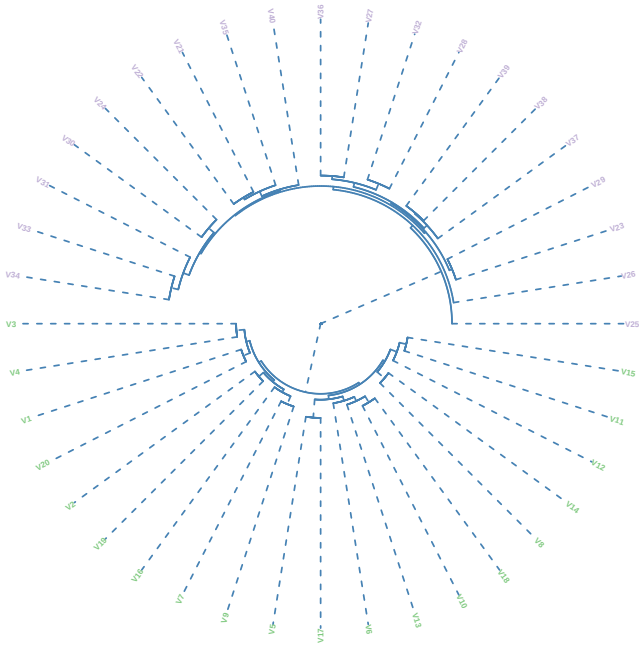
```
plot(as.phylo(hclust2), type = "fan", cex = 0.6,
    tip.color = brewer.pal(3, 'Accent')[cutree2],
      font = 2,
     edge.color = 'steelblue', edge.width = 2, edge.lty = 2)
```



```
cutree3 <- cutree(dend3,2)
cutree3
```

```
##  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
```

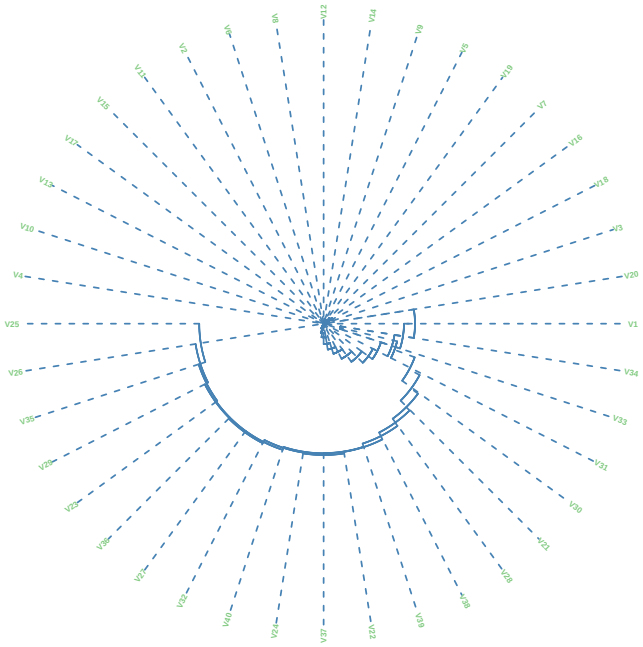
```
plot(as.phylo(hclust3), type = "fan", cex = 0.6,
    tip.color = brewer.pal(3, 'Accent')[cutree3],
      font = 2,
     edge.color = 'steelblue', edge.width = 2, edge.lty = 2)
```



```
cutree4 <- cutree(dend4,2)
cutree4
```

```
##  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
```

```
plot(as.phylo(hclust4), type = "fan", cex = 0.6,
    tip.color = brewer.pal(3, 'Accent')[cutree4],
      font = 2,
     edge.color = 'steelblue', edge.width = 2, edge.lty = 2)
```



We observe from the Fan plot and Also from the summary of each of the cutree of the dendrograms of various types of linkage methods used that the objects get classified based on the type of the linkage methods used such as Single, Complete,Centroid and Average. Incase of Average Linkage we observe a dramatic result where in all the data points get clustered in to the same cluster. While in case of other clusters there is a partition of the Principle space into 2 clusters but the size of each of these clusters vary.

In order to check which variables differ the most between Healthy and Diseased Patient we perform:

1)Principle Component Analysis on the dataset

2)K-Means Clustering

```
library(factoextra)
```

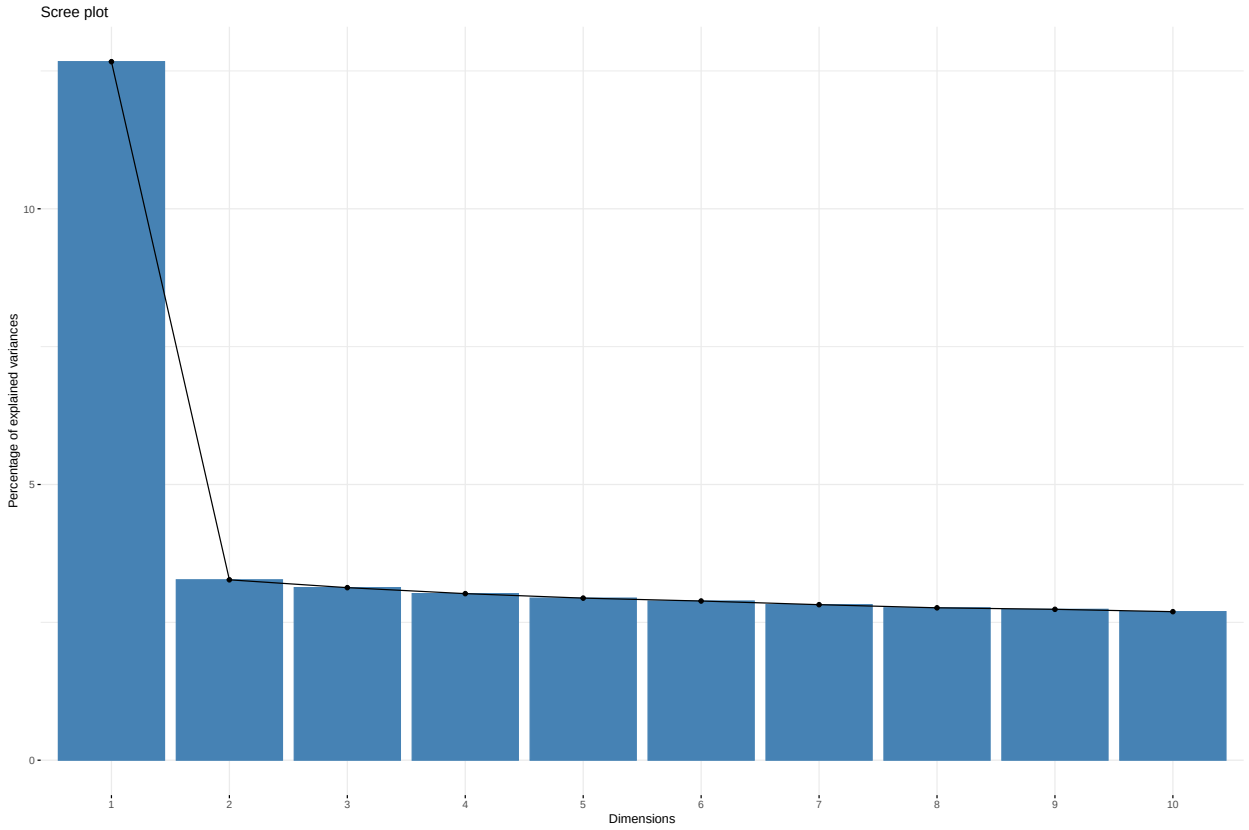
```
## Warning: package 'factoextra' was built under R version 3.6.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
pcomp_df <- prcomp(t(df1))
```

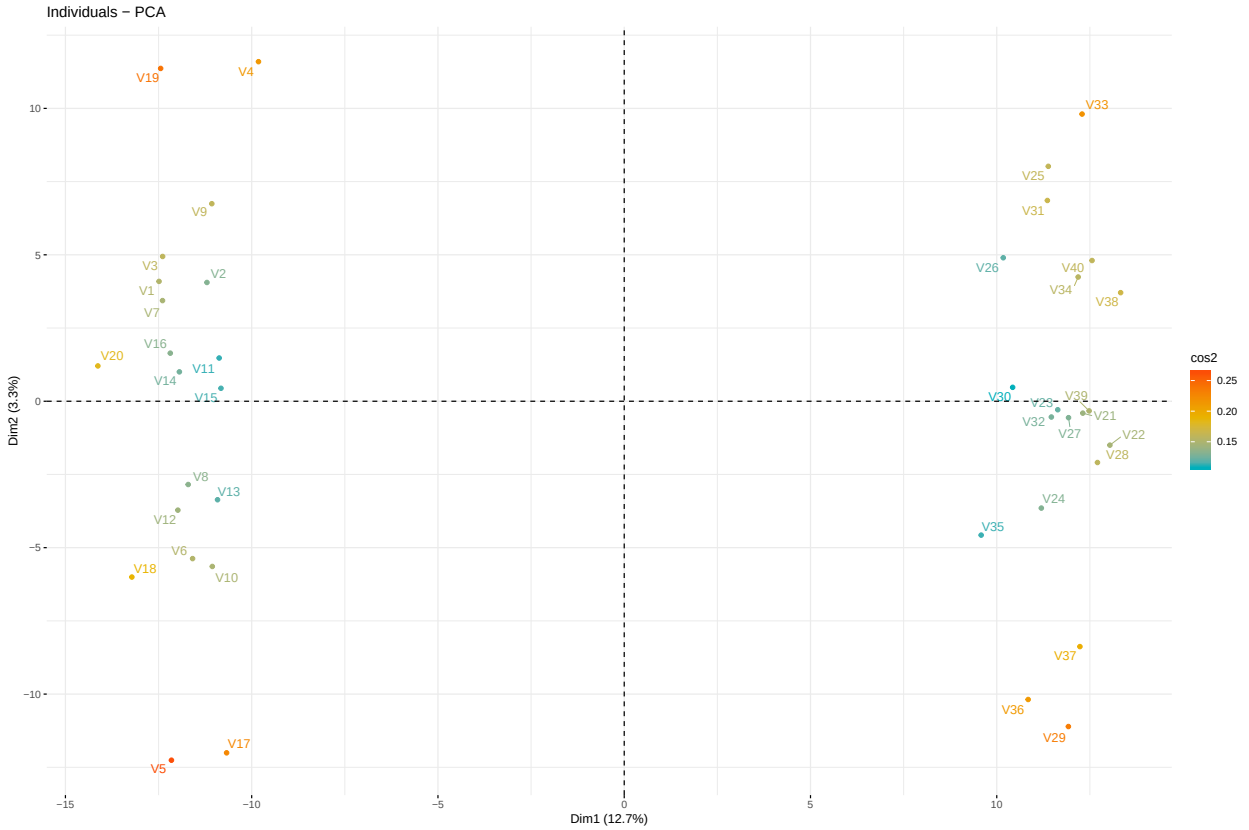
We observe the percentage of variance by each principal component.

```
fviz_eig(pcomp_df)
```



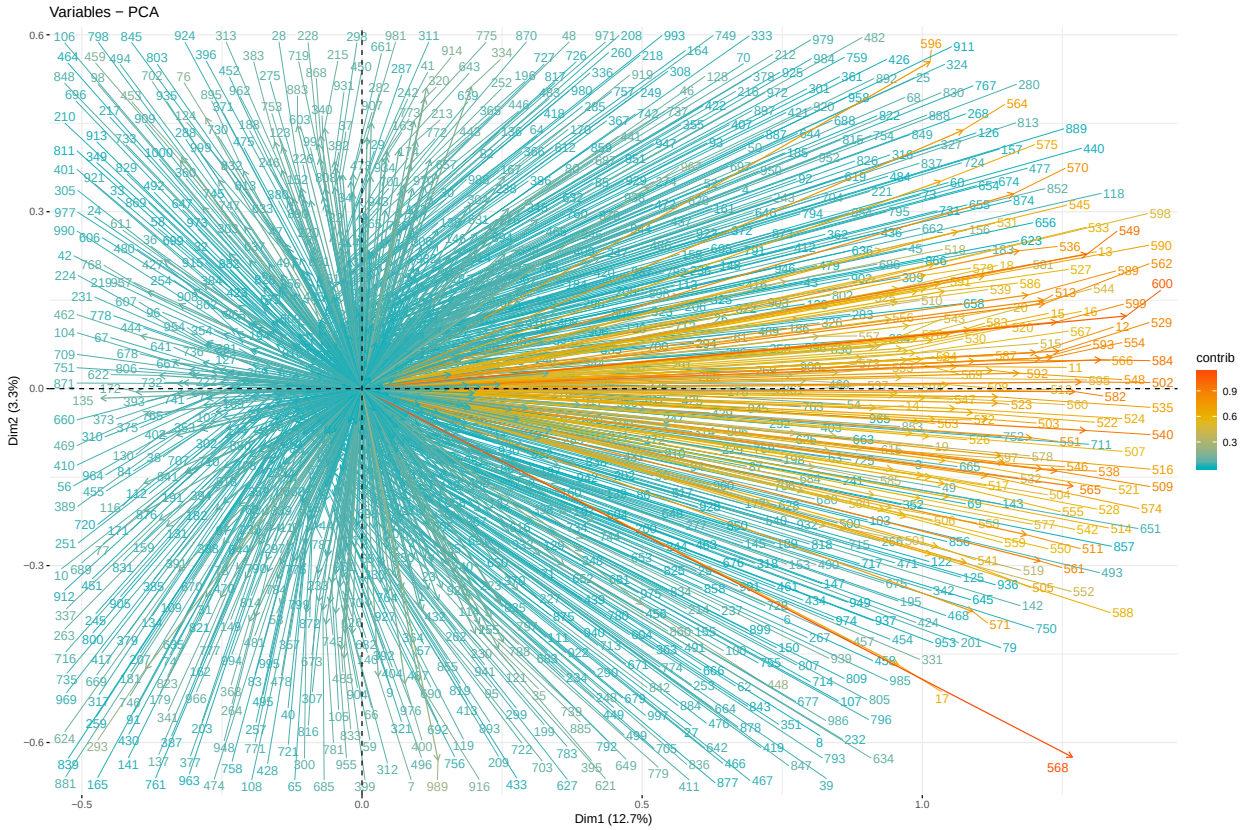
We Also map how the various Genes are mapped across the componenet space and this helps us visualize the clusters.

```
fviz_pca_ind(pcomp_df,
             col.ind = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE
             )
```



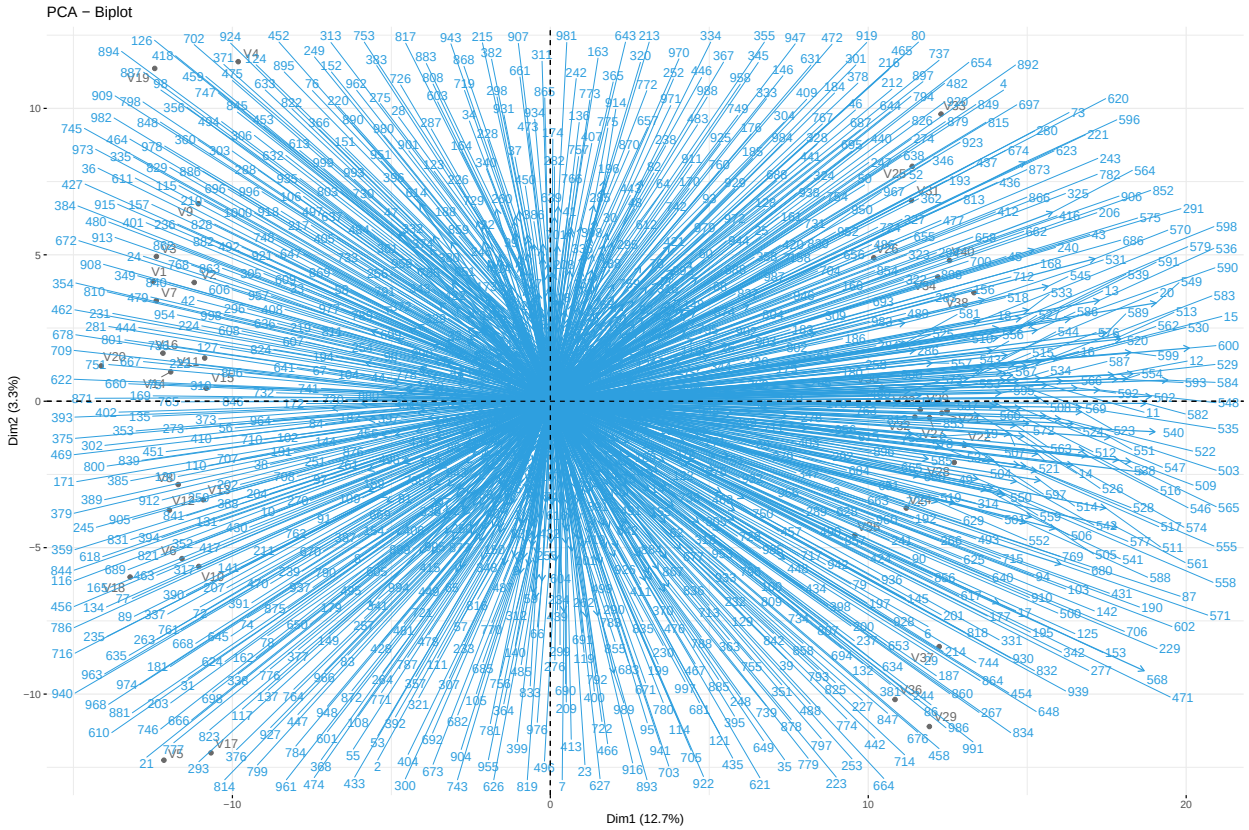
This plot helps us visualize how positively correlated points are mapped to the same side of the component space and the negatively correlated points are mapped to the opposite side of the component space.

```
fviz_pca_var(pcomp_df,
             col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE
             )
```



Biplot of the Genes

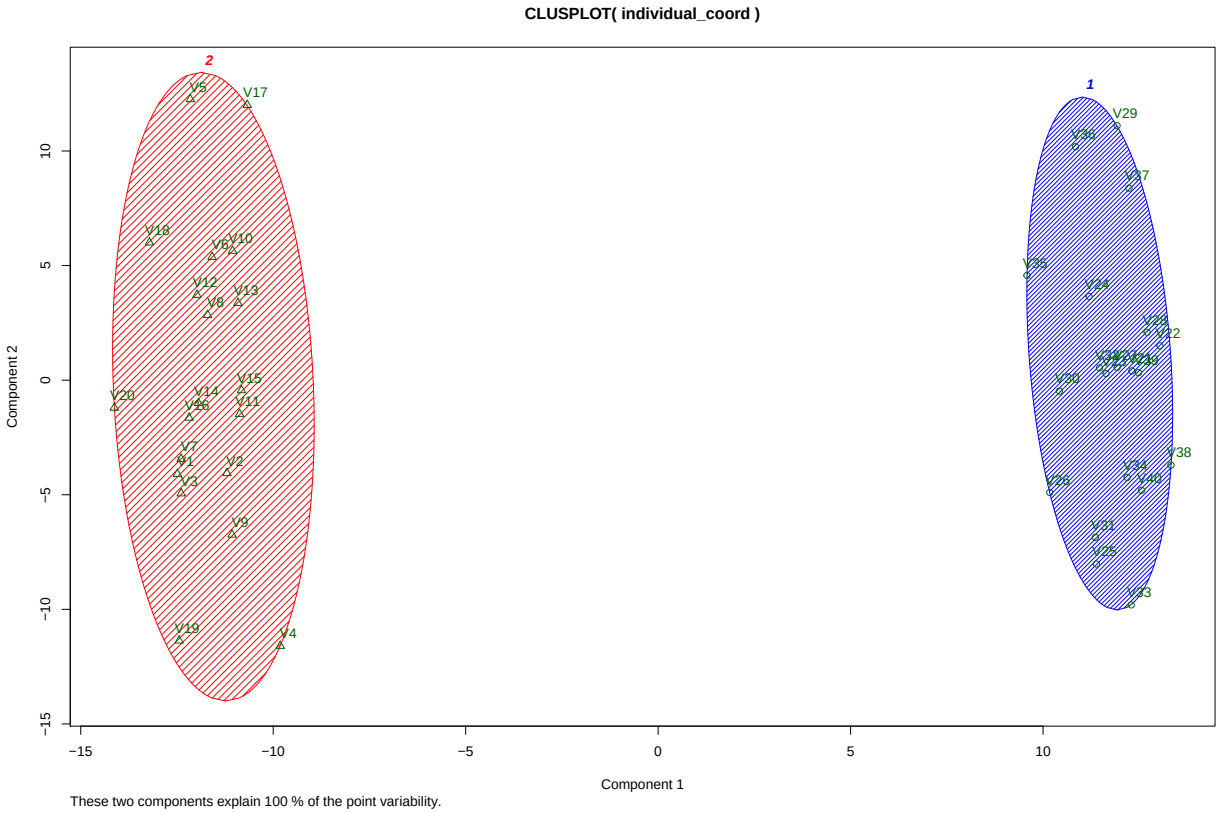
```
fviz_pca_biplot(pcomp_df, repel = TRUE,
                col.var = "#2E9FDF",
                col.ind = "#696969"
                )
```



‘ After observing the PCA Results in these visualization we use K-Means Clustering to divide each of the 2 principle components into its corresponding cluster.

```
individual_coord <- pcomp_df$x[, 1:2]
k2 <- kmeans(individual_coord, centers = 2,nstart=10)
```

```
library(cluster)
clusplot(individual_coord, k2$cluster, color=TRUE, shade=TRUE,
   labels=2, lines=0)
```



These two components explain 100 % of the point variability.

```
total <- apply(pcomp_df$x, 1, sum)
top <- order(abs(total), decreasing = TRUE)
top[1:10]
```

```
##  [1] 23 18 29 15 21 26 17 12 10 39
```

These are the Genes that differ the most accross the two groups.