Main exercise_GonzaloDíazAmor

Main exercise: Given a soil microbiome dataset, to design and develop a model to determine the location of novel samples.

Take as input the file 1_taxa_counts.csv. Each cell is the abundance of one taxon in that sample.

Classify the samples without an assigned class in the 1_metadata.csv file. Additionally, you could also report the probability to belong to the predicted class.

Determine the most relevant taxa (i.e. otuids) to classify the samples

Data Lecture

We load the file '1_taxa_counts.csv' and '1_metadata.csv'. In the first file we can see that we have 717 rows and 201 variables and in the second file we have 200 rows and 2 variables.

```
## Rows: 717 Columns: 201
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (201): otuids, 11116.L29A088.1195382, 11116.L08A089.1198461, 11116.L20A0...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Rows: 200 Columns: 2
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): SampleID, env
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## [1] 717 201
```

```
## [1] 200   2
```

We transpose the first file and perform a description and cleaning of the data.

```
#ID
id1<-colnames(datos1)
#Transpose
datos1<-t(datos1)
datos1<-as.data.frame(datos1)

#in meta I have the id and the class
meta1<-as.data.frame(meta1)
#First row as name of the columns
colnames(datos1) <- datos1[1,]
datos1 <- datos1[-1, ]
datos1$id<-id1[2:201]
meta1$id<-meta1$SampleID
#left join of data from df1 and df2
df1<-merge(datos1, meta1, by.x = "id",by.y="SampleID")
df1$env<-as.factor(df1$env)

#15 are not classified
df1_lm<-df1[,-1]
table(df1$env)
```

```
##
##   Aurora Columbus   Ithaca  Lansing   Urbana
##       51        8       53       62       11
```

```
#which values has variance 0
colvar0<-apply(df1_lm,2,function(x) var(x,na.rm=T)==0)
```

```
## Warning in var(x, na.rm = T): NAs introducidos por coerción

## Warning in var(x, na.rm = T): NAs introducidos por coerción
```

```
#get the column names
print(paste("Names of the columns with all 0's or NA: ",names(df1_lm)[colvar0|is.na(colvar0)]))
```

```
##  [1] "Names of the columns with all 0's or NA:  585221"
##  [2] "Names of the columns with all 0's or NA:  250148"
##  [3] "Names of the columns with all 0's or NA:  878714"
##  [4] "Names of the columns with all 0's or NA:  225453"
##  [5] "Names of the columns with all 0's or NA:  854050"
##  [6] "Names of the columns with all 0's or NA:  584331"
##  [7] "Names of the columns with all 0's or NA:  216643"
##  [8] "Names of the columns with all 0's or NA:  539978"
##  [9] "Names of the columns with all 0's or NA:  606989"
## [10] "Names of the columns with all 0's or NA:  833317"
## [11] "Names of the columns with all 0's or NA:  242284"
## [12] "Names of the columns with all 0's or NA:  415661"
## [13] "Names of the columns with all 0's or NA:  216925"
## [14] "Names of the columns with all 0's or NA:  137818"
## [15] "Names of the columns with all 0's or NA:  769643"
## [16] "Names of the columns with all 0's or NA:  11428"
## [17] "Names of the columns with all 0's or NA:  238109"
## [18] "Names of the columns with all 0's or NA:  510316"
## [19] "Names of the columns with all 0's or NA:  883748"
## [20] "Names of the columns with all 0's or NA:  810679"
## [21] "Names of the columns with all 0's or NA:  env"
## [22] "Names of the columns with all 0's or NA:  id.y"
```

```
drop <- names(colvar0[colvar0==TRUE])[1:20]
df = df1_lm[,!(names(df1_lm) %in% drop)]
df<-subset(df,!is.na(df$env))
```

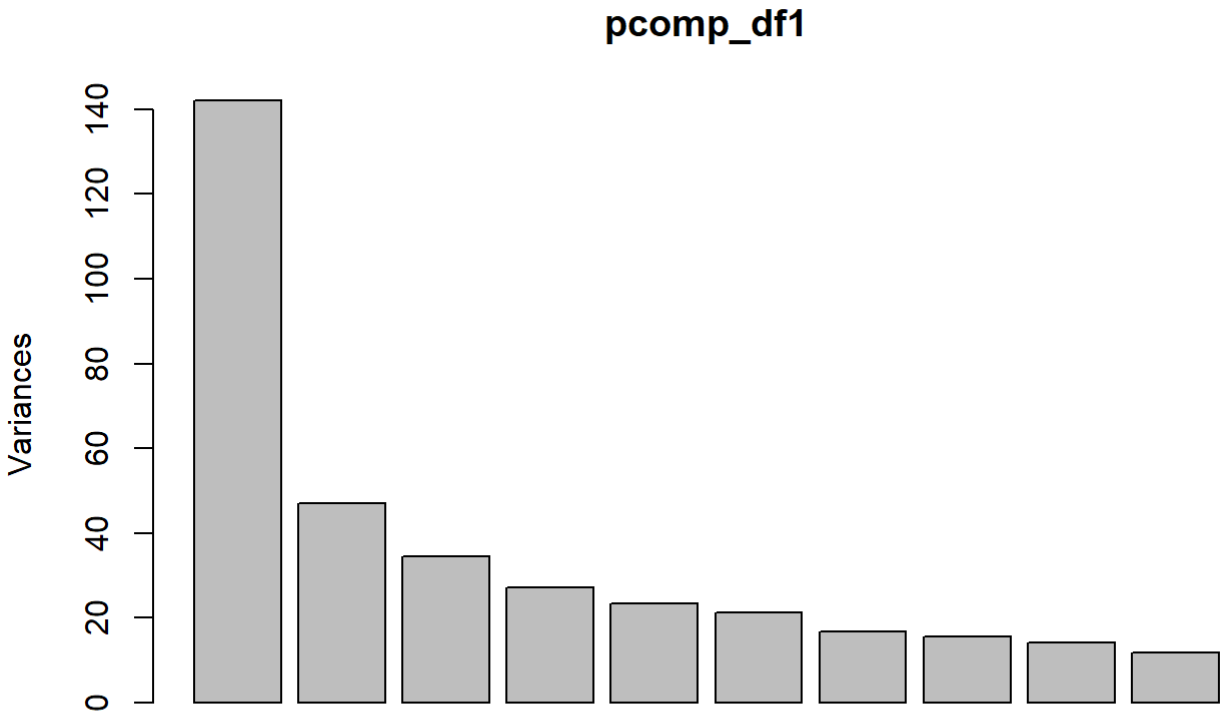
There are 15 rows without class and 20 columns with all 0's without include "env" and "id.y".

The new dataset is 699 variables and 185 observations

PCA

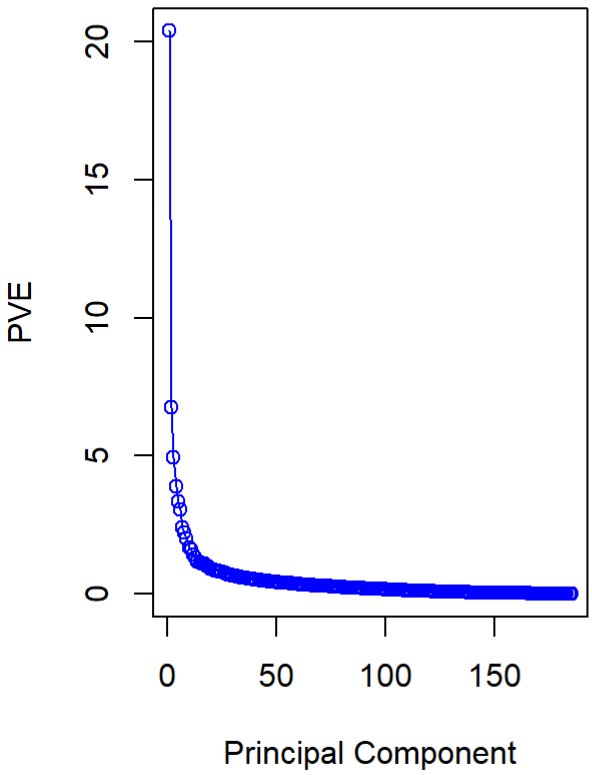
The principal components of a collection of points in a real coordinate space are a sequence of p unit vectors, where the i-th vector is the direction of a line that best fits the data while being orthogonal to the first $i-1$ vectors. Here, a best-fitting line is defined as one that minimizes the average squared distance from the points to the line. These directions constitute an orthonormal basis in which different individual dimensions of the data are linearly uncorrelated. Principal component analysis (PCA) is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest.

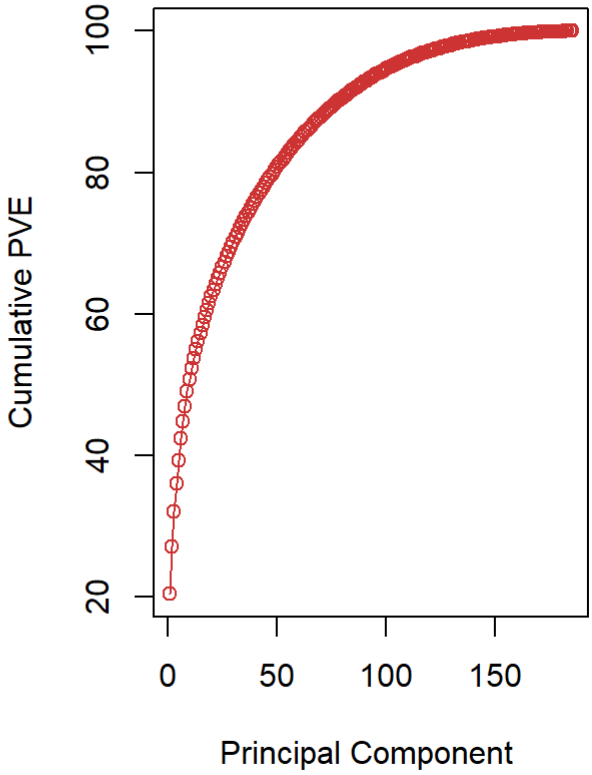
```
#PCA
df1_pca<-df
df1_pca$id.y<-NULL
df1_pca$env<-NULL

pcomp_df1 <- prcomp(df1_pca,scale=TRUE)
plot(pcomp_df1)
```



As we can see in the superior graph the first PCA has over 140 of variance which is high enought than 45 of the second PCA.

```
pve =100*pcomp_df1$sdev ^2/sum(pcomp_df1$sdev ^2)
par(mfrow=c(1,2))
plot(pve , type="o", ylab="PVE", xlab=" Principal Component ", col="blue")
plot(cumsum(pve), type="o", ylab="Cumulative PVE", xlab="Principal Component ", col="brown3")
```






```
summary(pcomp_df1)$importance
```

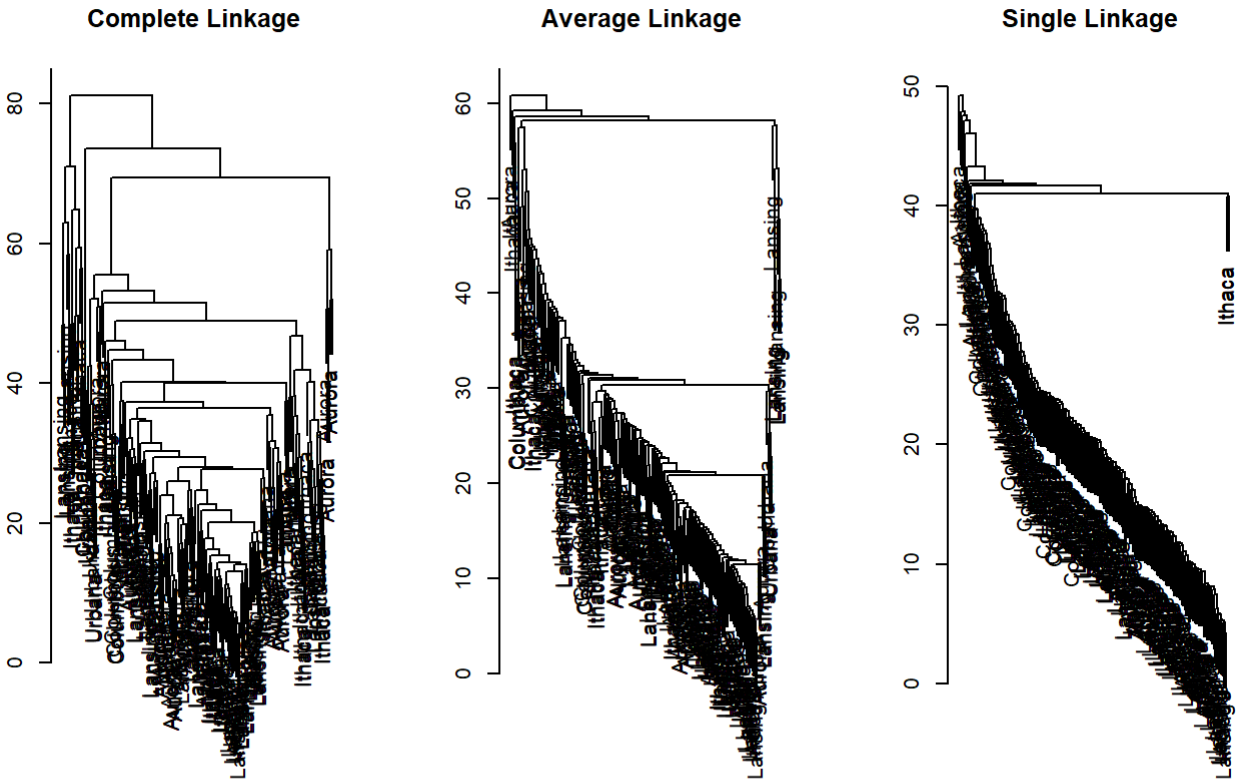
```
##                            PC1      PC2      PC3      PC4      PC5     PC6
## Standard deviation     11.9213 6.855215 5.871867 5.205465 4.838242 4.60918
## Proportion of Variance  0.2039 0.067420 0.049470 0.038880 0.033580 0.03048
## Cumulative Proportion   0.2039 0.271320 0.320790 0.359670 0.393250 0.42373
##                             PC7      PC8      PC9     PC10     PC11     PC12
## Standard deviation     4.083015 3.934171 3.748861 3.425472 3.343282 3.152525
## Proportion of Variance 0.023920 0.022210 0.020160 0.016830 0.016040 0.014260
## Cumulative Proportion  0.447650 0.469850 0.490020 0.506850 0.522890 0.537150
##                            PC13     PC14     PC15     PC16     PC17     PC18
## Standard deviation     3.017083 2.844599 2.828063 2.782379 2.749005 2.686104
## Proportion of Variance 0.013060 0.011610 0.011470 0.011110 0.010840 0.010350
## Cumulative Proportion  0.550210 0.561820 0.573290 0.584400 0.595240 0.605590
##                            PC19     PC20     PC21     PC22     PC23     PC24
## Standard deviation     2.601961 2.526335 2.476316 2.434242 2.404747 2.366561
## Proportion of Variance 0.009710 0.009160 0.008800 0.008500 0.008300 0.008040
## Cumulative Proportion  0.615310 0.624460 0.633260 0.641760 0.650060 0.658100
##                           PC25     PC26     PC27     PC28     PC29     PC30
## Standard deviation     2.35991 2.325056 2.247756 2.208968 2.180239 2.168386
## Proportion of Variance 0.00799 0.007760 0.007250 0.007000 0.006820 0.006750
## Cumulative Proportion  0.66609 0.673840 0.681090 0.688090 0.694910 0.701660
##                            PC31    PC32    PC33     PC34    PC35     PC36
## Standard deviation     2.136916 2.11259 2.08975 2.044103 2.02657 1.994723
## Proportion of Variance 0.006550 0.00640 0.00627 0.005990 0.00589 0.005710
## Cumulative Proportion  0.708210 0.71461 0.72088 0.726870 0.73276 0.738470
##                            PC37     PC38     PC39     PC40     PC41     PC42
## Standard deviation     1.981053 1.966602 1.928426 1.900827 1.889201 1.870792
## Proportion of Variance 0.005630 0.005550 0.005340 0.005180 0.005120 0.005020
## Cumulative Proportion  0.744100 0.749650 0.754990 0.760170 0.765290 0.770310
##                            PC43     PC44     PC45     PC46     PC47     PC48
## Standard deviation     1.854052 1.832923 1.816518 1.798231 1.783257 1.773043
## Proportion of Variance 0.004930 0.004820 0.004730 0.004640 0.004560 0.004510
## Cumulative Proportion  0.775250 0.780070 0.784800 0.789440 0.794000 0.798510
##                            PC49     PC50     PC51    PC52     PC53     PC54
## Standard deviation     1.759242 1.730471 1.703485 1.69563 1.688592 1.663989
## Proportion of Variance 0.004440 0.004300 0.004160 0.00413 0.004090 0.003970
## Cumulative Proportion  0.802950 0.807250 0.811410 0.81554 0.819630 0.823600
##                            PC55     PC56     PC57     PC58     PC59     PC60
## Standard deviation     1.659101 1.651802 1.637723 1.617242 1.610175 1.595747
## Proportion of Variance 0.003950 0.003910 0.003850 0.003750 0.003720 0.003650
## Cumulative Proportion  0.827550 0.831460 0.835310 0.839060 0.842780 0.846440
##                           PC61     PC62     PC63     PC64     PC65     PC66
## Standard deviation     1.56143 1.548547 1.534727 1.527381 1.514624 1.506151
## Proportion of Variance 0.00350 0.003440 0.003380 0.003350 0.003290 0.003250
## Cumulative Proportion  0.84994 0.853380 0.856760 0.860100 0.863390 0.866650
##                            PC67    PC68     PC69     PC70     PC71     PC72
## Standard deviation     1.472016 1.46315 1.457958 1.440859 1.437114 1.420857
## Proportion of Variance 0.003110 0.00307 0.003050 0.002980 0.002960 0.002900
## Cumulative Proportion  0.869760 0.87283 0.875880 0.878860 0.881820 0.884720
##                            PC73    PC74     PC75     PC76    PC77    PC78
## Standard deviation     1.404235 1.39581 1.387639 1.362685 1.34180 1.33493
## Proportion of Variance 0.002830 0.00280 0.002760 0.002660 0.00258 0.00256
## Cumulative Proportion  0.887550 0.89034 0.893100 0.895770 0.89835 0.90091
##                            PC79    PC80     PC81     PC82     PC83     PC84
## Standard deviation     1.317936 1.30086 1.286506 1.285198 1.274143 1.262469
## Proportion of Variance 0.002490 0.00243 0.002370 0.002370 0.002330 0.002290
## Cumulative Proportion  0.903400 0.90583 0.908200 0.910570 0.912900 0.915190
##                            PC85     PC86     PC87     PC88    PC89     PC90
## Standard deviation     1.250026 1.236775 1.230902 1.218035 1.20040 1.189845
## Proportion of Variance 0.002240 0.002190 0.002170 0.002130 0.00207 0.002030
```

```
## Cumulative Proportion  0.917430 0.919620 0.921800 0.923930 0.92599 0.928030
##                           PC91     PC92     PC93     PC94     PC95     PC96
## Standard deviation     1.17977 1.170422 1.162728 1.144916 1.129018 1.113158
## Proportion of Variance 0.00200 0.001970 0.001940 0.001880 0.001830 0.001780
## Cumulative Proportion  0.93002 0.931990 0.933930 0.935810 0.937640 0.939410
##                            PC97     PC98     PC99    PC100    PC101   PC102
## Standard deviation     1.109697 1.094579 1.089928 1.076198 1.056961 1.04233
## Proportion of Variance 0.001770 0.001720 0.001700 0.001660 0.001600 0.00156
## Cumulative Proportion  0.941180 0.942900 0.944600 0.946270 0.947870 0.94943
##                           PC103    PC104    PC105     PC106     PC107     PC108
## Standard deviation     1.020528 1.015863 1.004566 0.9922937 0.9837686 0.9662917
## Proportion of Variance 0.001490 0.001480 0.001450 0.0014100 0.0013900 0.0013400
## Cumulative Proportion  0.950920 0.952400 0.953850 0.9552600 0.9566500 0.9579900
##                            PC109    PC110     PC111     PC112     PC113
## Standard deviation     0.9529694 0.942307 0.9410536 0.9352481 0.9216263
## Proportion of Variance 0.0013000 0.001270 0.0012700 0.0012500 0.0012200
## Cumulative Proportion  0.9592900 0.960570 0.9618400 0.9630900 0.9643100
##                            PC114    PC115     PC116     PC117     PC118
## Standard deviation     0.8997591 0.891585 0.8780537 0.8741776 0.8609495
## Proportion of Variance 0.0011600 0.001140 0.0011100 0.0011000 0.0010600
## Cumulative Proportion  0.9654700 0.966610 0.9677200 0.9688200 0.9698800
##                            PC119     PC120     PC121     PC122     PC123
## Standard deviation     0.8531988 0.8464616 0.8286931 0.8209009 0.8057399
## Proportion of Variance 0.0010400 0.0010300 0.0009900 0.0009700 0.0009300
## Cumulative Proportion  0.9709200 0.9719500 0.9729400 0.9739000 0.9748400
##                            PC124     PC125     PC126     PC127     PC128
## Standard deviation     0.8036054 0.7994014 0.7768123 0.7726253 0.7611496
## Proportion of Variance 0.0009300 0.0009200 0.0008700 0.0008600 0.0008300
## Cumulative Proportion  0.9757600 0.9766800 0.9775500 0.9784000 0.9792300
##                            PC129     PC130     PC131     PC132     PC133
## Standard deviation     0.7470861 0.7393305 0.7211996 0.7143897 0.7107709
## Proportion of Variance 0.0008000 0.0007800 0.0007500 0.0007300 0.0007200
## Cumulative Proportion  0.9800300 0.9808200 0.9815600 0.9823000 0.9830200
##                            PC134     PC135     PC136     PC137    PC138
## Standard deviation     0.6932518 0.6825711 0.6698482 0.6612481 0.653427
## Proportion of Variance 0.0006900 0.0006700 0.0006400 0.0006300 0.000610
## Cumulative Proportion  0.9837100 0.9843800 0.9850200 0.9856500 0.986260
##                            PC139    PC140     PC141    PC142     PC143
## Standard deviation     0.6493191 0.640344 0.6161139 0.614658 0.5991341
## Proportion of Variance 0.0006000 0.000590 0.0005400 0.000540 0.0005200
## Cumulative Proportion  0.9868700 0.987460 0.9880000 0.988540 0.9890600
##                            PC144     PC145     PC146     PC147     PC148
## Standard deviation     0.5967304 0.5933978 0.5735227 0.5682614 0.5603311
## Proportion of Variance 0.0005100 0.0005100 0.0004700 0.0004600 0.0004500
## Cumulative Proportion  0.9895700 0.9900700 0.9905500 0.9910100 0.9914600
##                            PC149     PC150     PC151     PC152     PC153
## Standard deviation     0.5514611 0.5464679 0.5373955 0.5274298 0.5165098
## Proportion of Variance 0.0004400 0.0004300 0.0004100 0.0004000 0.0003800
## Cumulative Proportion  0.9919000 0.9923200 0.9927400 0.9931400 0.9935200
##                            PC154    PC155     PC156    PC157     PC158
## Standard deviation     0.5066744 0.499945 0.4982422 0.493679 0.4916226
## Proportion of Variance 0.0003700 0.000360 0.0003600 0.000350 0.0003500
## Cumulative Proportion  0.9938900 0.994250 0.9946000 0.994950 0.9953000
##                            PC159     PC160    PC161     PC162     PC163
## Standard deviation     0.4777354 0.4672432 0.454339 0.4402223 0.4335932
## Proportion of Variance 0.0003300 0.0003100 0.000300 0.0002800 0.0002700
## Cumulative Proportion  0.9956300 0.9959400 0.996240 0.9965200 0.9967800
##                            PC164     PC165     PC166    PC167     PC168
## Standard deviation     0.4307146 0.4191513 0.4119625 0.402779 0.3933564
## Proportion of Variance 0.0002700 0.0002500 0.0002400 0.000230 0.0002200
```

```
## Cumulative Proportion  0.9970500 0.9973000 0.9975500 0.997780 0.9980000
##                            PC169     PC170     PC171     PC172     PC173
## Standard deviation     0.3844501 0.3751348 0.3631886 0.3463084 0.3406943
## Proportion of Variance 0.0002100 0.0002000 0.0001900 0.0001700 0.0001700
## Cumulative Proportion  0.9982100 0.9984200 0.9986000 0.9987800 0.9989400
##                            PC174    PC175     PC176     PC177     PC178   PC179
## Standard deviation     0.3240695 0.316381 0.3085325 0.2800945 0.2696879 0.26348
## Proportion of Variance 0.0001500 0.000140 0.0001400 0.0001100 0.0001000 0.00010
## Cumulative Proportion  0.9990900 0.999240 0.9993700 0.9994900 0.9995900 0.99969
##                            PC180    PC181     PC182     PC183      PC184
## Standard deviation     0.2464692 0.227411 0.2270257 0.2067362 0.09484909
## Proportion of Variance 0.0000900 0.000070 0.0000700 0.0000600 0.00001000
## Cumulative Proportion  0.9997800 0.999850 0.9999300 0.9999900 1.00000000
##                              PC185
## Standard deviation     5.29412e-15
## Proportion of Variance 0.00000e+00
## Cumulative Proportion  1.00000e+00
```

With the first two PCA we have about 27% of variance explained. In case WE want to have about 80% of variance explained we need to take about 48 PCA and about 78 for the 90% of variance explained.

```
sd.data=scale(df1_pca)
par(mfrow=c(1,3))
data.dist=dist(sd.data)
plot(hclust(data.dist), labels=df$env , main="Complete Linkage ", xlab="", sub="",ylab="")
plot(hclust(data.dist , method ="average"), labels=df$env , main="Average Linkage ", xlab="", sub="",yl
ab="")
plot(hclust(data.dist , method ="single"), labels=df$env , main="Single Linkage ", xlab="", sub="",ylab
="")
```



```
hc.out=hclust(dist(sd.data))
hc.clusters =cutree (hc.out ,5)
table(hc.clusters ,df$env)
```

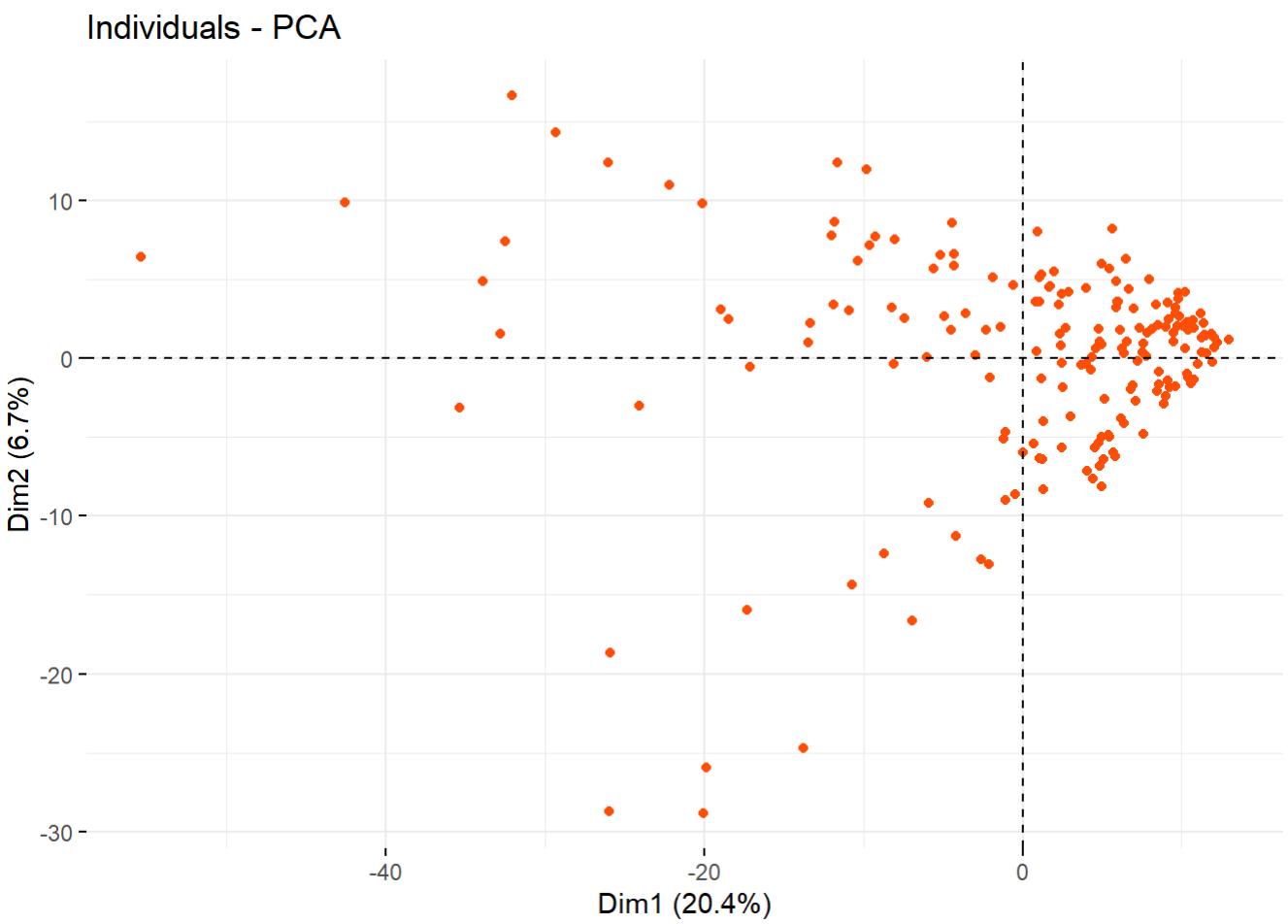
```
##
## hc.clusters Aurora Columbus Ithaca Lansing Urbana
##           1     46        6     45      55     11
##           2      5        0      0       0      0
##           3      0        2      0       0      0
##           4      0        0      8       1      0
##           5      0        0      0       6      0
```

```
km.out=kmeans(sd.data , 5, nstart =20)
km.clusters =km.out$cluster
table(km.clusters ,hc.clusters)
```

```
##            hc.clusters
## km.clusters   1   2   3   4   5
##           1  25   0   0   6   0
##           2   0   0   0   3   6
##           3   5   5   0   0   0
##           4   4   0   2   0   0
##           5 129   0   0   0   0
```

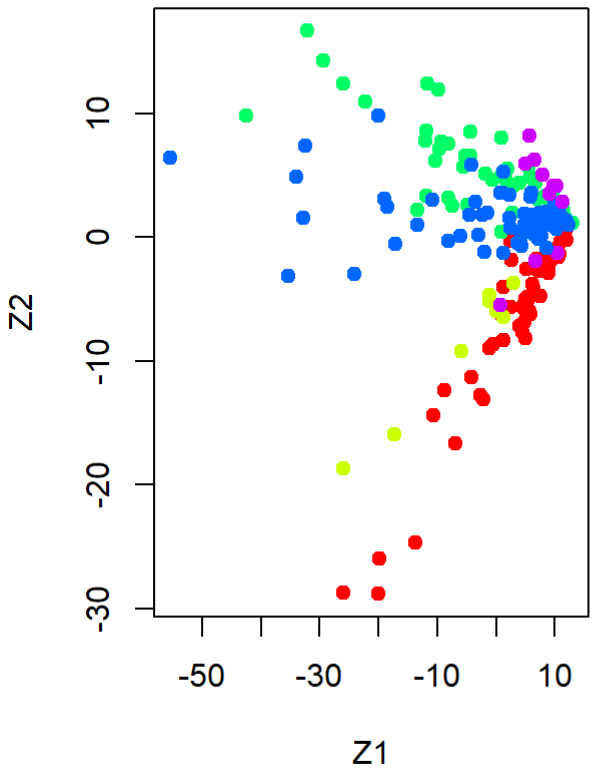
The plots of dendograms above is with different types of linkage to group the samples. As we can see it is a huge amount of data and it is hard to achieve some information

```
fviz_pca_ind(pcomp_df1, geom.ind = "point",
             col.ind = "#FC4E07",
             axes = c(1,2),
             pointsize = 1.5)
```



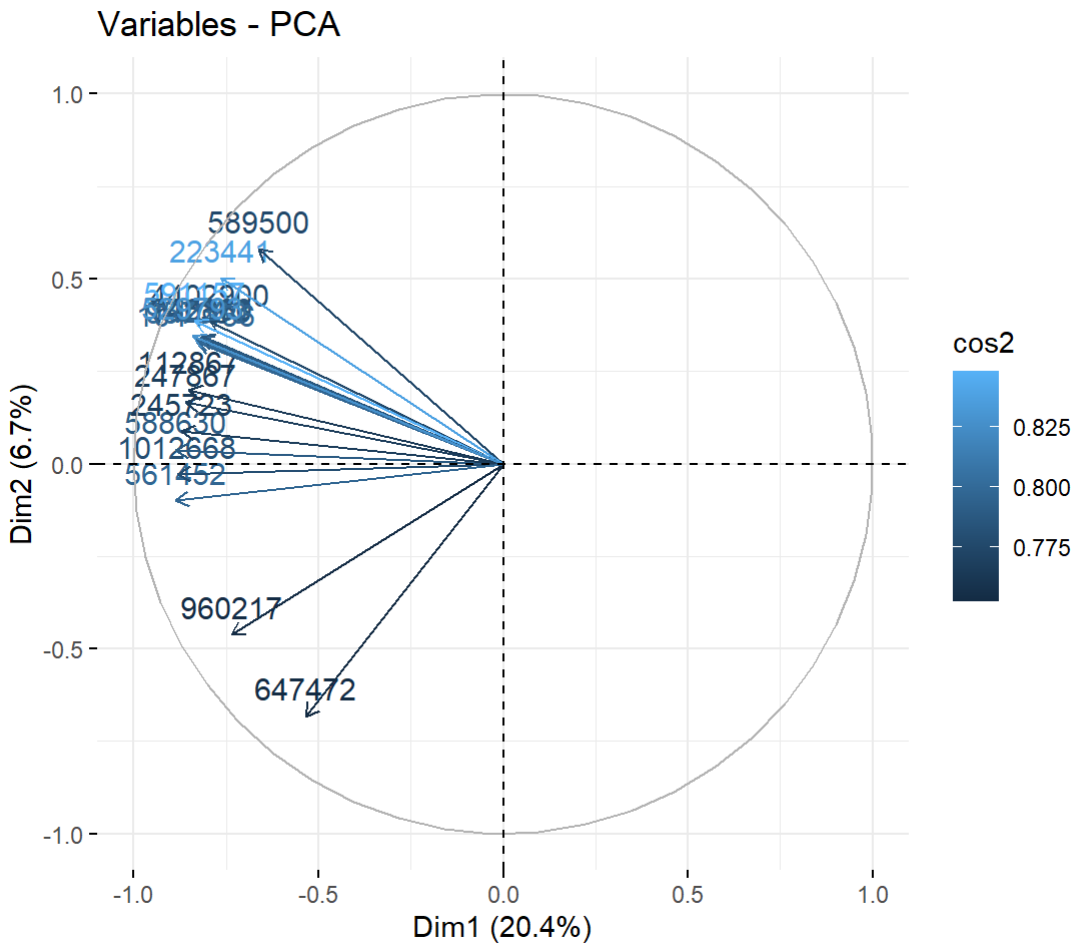
This representation we can see is the first PCA and the points of each row. More than half of the points are over the 0 value of x-axis which is the PCA1 and are very disperse in the left side.

```
colores <- function(vec){
  # la función rainbow() devuelve un vector que contiene el número de colores distintos
  col <- rainbow(length(unique(vec)))
  return(col[as.numeric(as.factor(vec))])
}

par(mfrow = c(1,2))
# Observaciones sobre PC1 y PC2
plot(pcomp_df1$x[,1:2], col = colores(df$env),
     pch = 19,
     xlab = "Z1",
     ylab = "Z2")
```



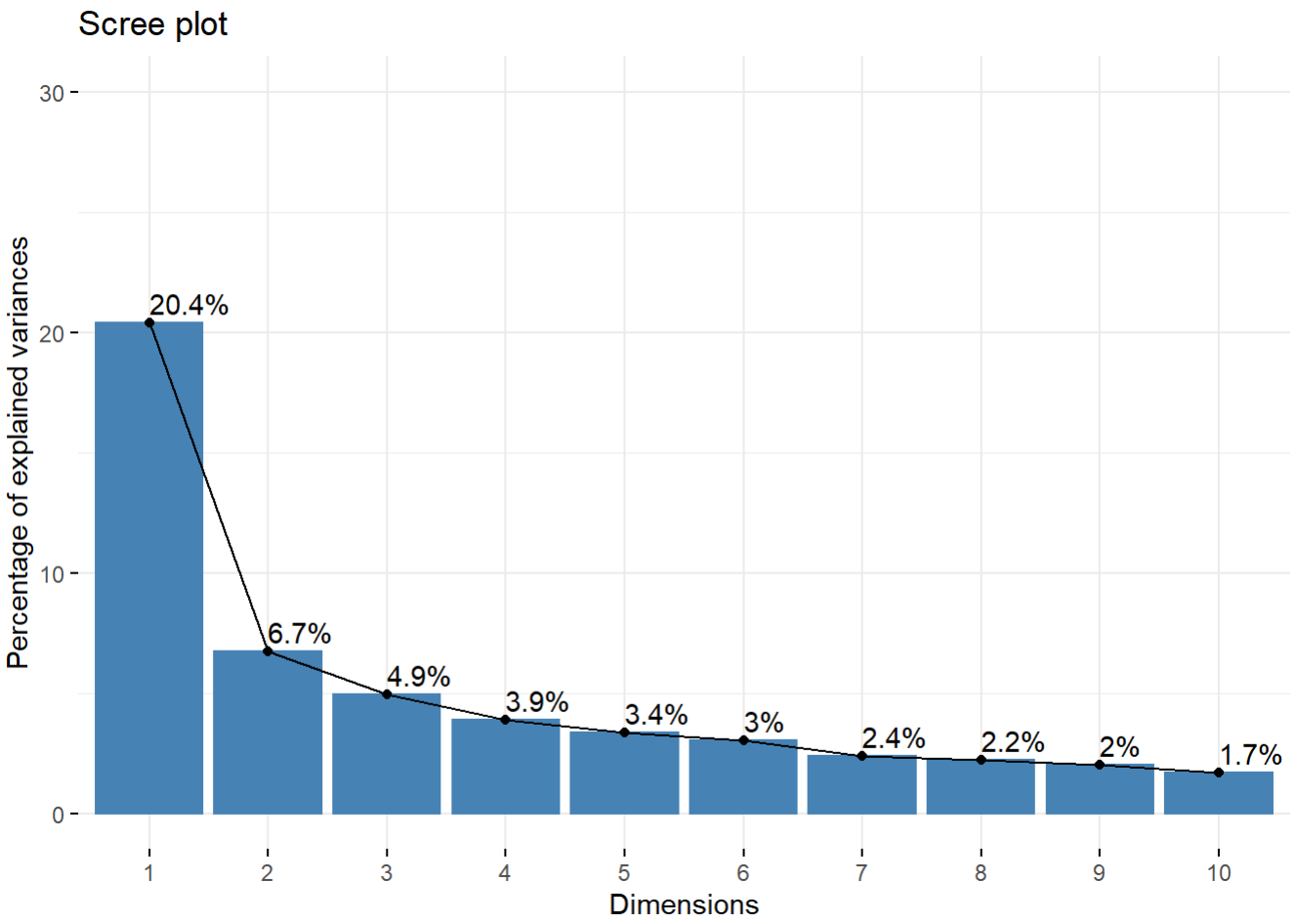
In this representation we can see the plot of the two firsts PCAs with the class of each row. The points of different classes are crossed over all the graph.

```
fviz_pca_var(pcomp_df1,col.var = "cos2", select.var = list(cos2 = 0.75))
```



Here we can see the variables which are over 0.75 of influence in the first PCA with negative values all of them

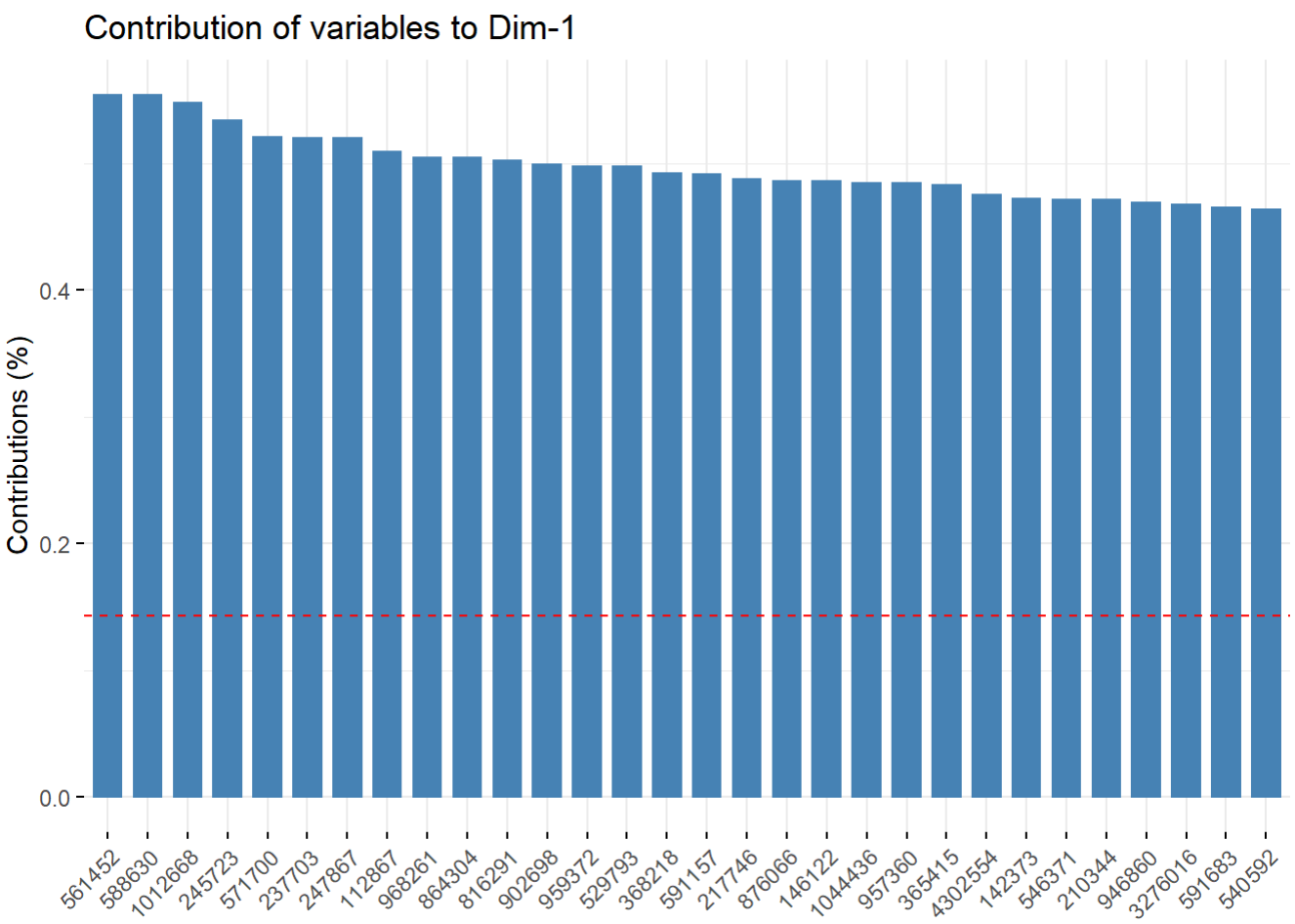
```
#Eleccion de componentes principales
fviz_screeplot(pcomp_df1, addlabels = TRUE, ylim = c(0, 30))
```



In this screeplot we can choose the dimensions of the PCA by the elbow rule which is a concordance of the minimum number of PCA and the maximum of variance explained.

We have seen the choose of two PCA or dimensions due to be the easier way to visualize.

```
fviz_contrib(pcomp_df1, choice = "var", axes = 1, top = 30)
```



Up here we can see the 30 variables which most influence are over the PCA1

Neural Network

For the classification of the classes of this problem the best approach was made using neural network with an arquitecture of 697 neurons of input, 400 neurons in the next layer, 200 in the next hidden layer and finally 5 neurons for the classification of the 5 class.

```
#creating indices
df$id.y<-NULL
trainIndex <- createDataPartition(df$env,p=0.8,list=FALSE)

#splitting data into training/testing data using the trainIndex object
df1_pca_var_train <- df[trainIndex,] #training data (80% of data)

df1_pca_var_test <- df[-trainIndex,] #testing data (20% of data)



newdata <- one_hot(as.data.table(df1_pca_var_train))
newdata_test<-one_hot(as.data.table(df1_pca_var_test))
#Scale data

#xo = apply(o,MARGIN = 2, FUN = range01)

#newdata[, 1:697] <- data.frame(lapply(newdata[, 1:697], scl))

colnames(newdata)<-paste("V",colnames(newdata),sep="")
colnames(newdata_test)<-paste("V",colnames(newdata_test),sep="")
n <- names(newdata)

f <- as.formula(paste("Venv_Aurora+Venv_Lansing+Venv_Ithaca+Venv_Columbus+Venv_Urbana~", paste(n[!n %in
% c("Venv_Aurora","Venv_Lansing","Venv_Ithaca","Venv_Columbus","Venv_Urbana")], collapse = "+")))



#Entrenamos la red neuronal
nn <- neuralnet( f,
                 data = newdata,
                 hidden = c(697,400,200, 5),
                 stepmax=1e6,
                 act.fct = "logistic",
                 linear.output = FALSE,
                 lifesign = "minimal")
```

```
## hidden: 697, 400, 200, 5    thresh: 0.01    rep: 1/1    steps:     116   error: 0.00057  time: 50.32
secs
```

```
#plot(nn)
# Compute predictions
pr.nn <- compute(nn, newdata_test[, 1:697])
# Extract results
pr.nn_ <- pr.nn$net.result
p_asignacion<-pr.nn$net.result
# Accuracy (training set)
original_values <- max.col(newdata_test[, 698:702])
pr.nn_2 <- max.col(pr.nn_)


confusionMatrix(as.factor(pr.nn_2),as.factor(original_values))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 1 2 3 4 5
##          1 8 0 0 1 1
##          2 1 0 3 9 0
##          3 0 0 7 2 0
##          4 1 1 0 0 0
##          5 0 0 0 0 1
##
## Overall Statistics
##
##                Accuracy : 0.4571
##                  95% CI : (0.2883, 0.6335)
##     No Information Rate : 0.3429
##     P-Value [Acc > NIR] : 0.1077
##
##                   Kappa : 0.3323
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.8000  0.00000   0.7000  0.00000  0.50000
## Specificity            0.9200  0.61765   0.9200  0.91304  1.00000
## Pos Pred Value         0.8000  0.00000   0.7778  0.00000  1.00000
## Neg Pred Value         0.9200  0.95455   0.8846  0.63636  0.97059
## Prevalence             0.2857  0.02857   0.2857  0.34286  0.05714
## Detection Rate         0.2286  0.00000   0.2000  0.00000  0.02857
## Detection Prevalence   0.2857  0.37143   0.2571  0.05714  0.02857
## Balanced Accuracy      0.8600  0.30882   0.8100  0.45652  0.75000
```

```
table(pr.nn_2)
```

```
## pr.nn_2
##  1  2  3  4  5
## 10 13  9  2  1
```

```
table(original_values)
```

```
## original_values
##  1  2  3  4  5
## 10  1 10 12  2
```

As we can see in the table above this classification is not as good as we desire because the accuracy it is about 0.5.

Conclusion

The use of PCA was useful for us because we can see the influence of 30 variables in the PCA1 which explained above 20% of variance of the problem. Inside

Neural Network is used with the original variables for the classification of the classes due to the predictive variables. We can say that we can improve our result if this type of input would be scaled and the use of PCA.