Principal Components

Banknotes dataset

```
# banknote.r
#
library(mclust)     # dataset
df0 <- banknote
df0[95:106,]
```

```
##          Status Length  Left Right Bottom  Top Diagonal
## 95      genuine  214.7 129.6 129.5    8.3 10.0    142.0
## 96      genuine  215.6 129.9 129.9    9.0  9.5    141.7
## 97      genuine  215.0 130.4 130.3    9.1 10.2    141.1
## 98      genuine  214.4 129.7 129.5    8.0 10.3    141.2
## 99      genuine  215.1 130.0 129.8    9.1 10.2    141.5
## 100     genuine  214.7 130.0 129.4    7.8 10.0    141.2
## 101 counterfeit  214.4 130.1 130.3    9.7 11.7    139.8
## 102 counterfeit  214.9 130.5 130.2   11.0 11.5    139.5
## 103 counterfeit  214.9 130.3 130.1    8.7 11.7    140.2
## 104 counterfeit  215.0 130.4 130.6    9.9 10.9    140.3
## 105 counterfeit  214.7 130.2 130.3   11.8 10.9    139.7
## 106 counterfeit  215.0 130.2 130.2   10.6 10.7    139.9
```

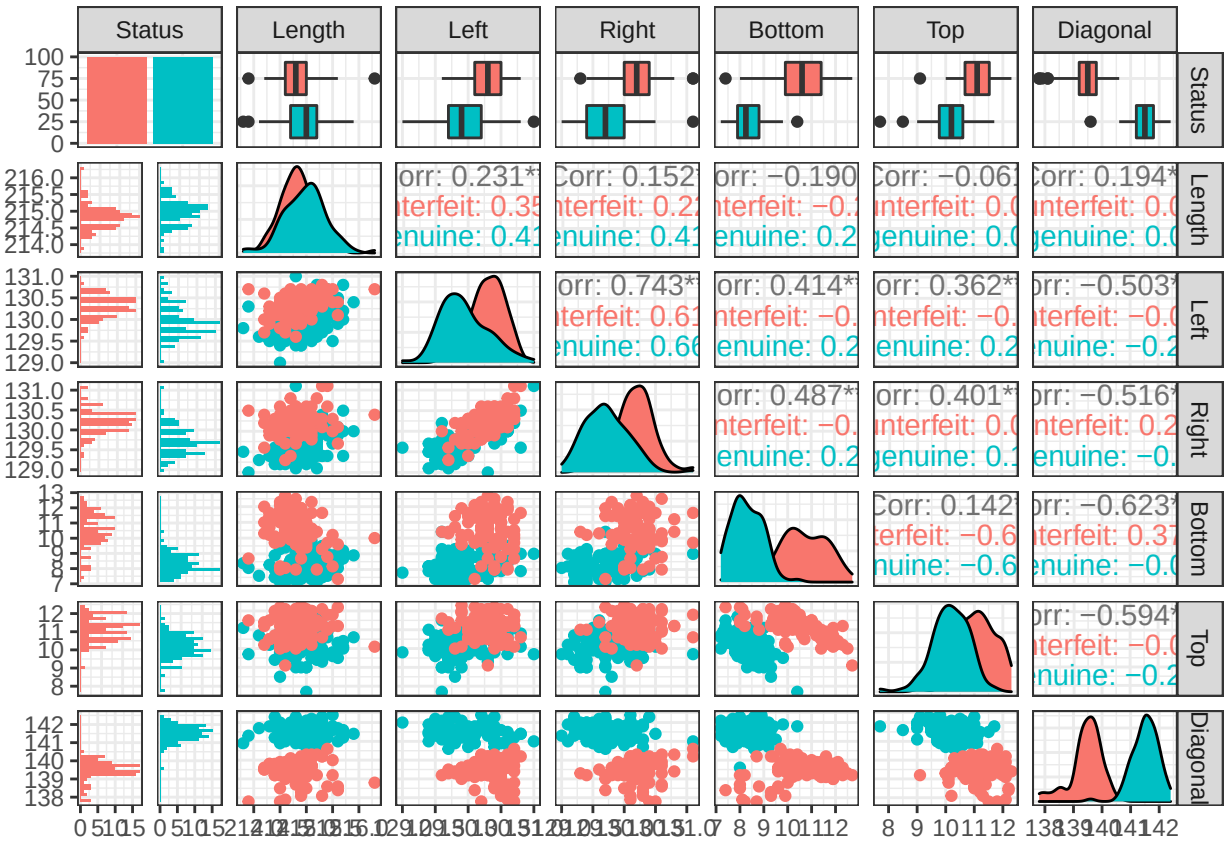
```
str(df0)
```

```
## 'data.frame':    200 obs. of  7 variables:
##  $ Status  : Factor w/ 2 levels "counterfeit",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Length  : num  215 215 215 215 215 ...
##  $ Left    : num  131 130 130 130 130 ...
##  $ Right   : num  131 130 130 130 130 ...
##  $ Bottom  : num  9 8.1 8.7 7.5 10.4 9 7.9 7.2 8.2 9.2 ...
##  $ Top     : num  9.7 9.5 9.6 10.4 7.7 10.1 9.6 10.7 11 10 ...
##  $ Diagonal: num  141 142 142 142 142 ...
```

```
summary(df0)
```

```
##          Status        Length           Left           Right
##  counterfeit:100   Min.   :213.8   Min.   :129.0   Min.   :129.0
##  genuine    :100   1st Qu.:214.6   1st Qu.:129.9   1st Qu.:129.7
##                    Median :214.9   Median :130.2   Median :130.0
##                    Mean   :214.9   Mean   :130.1   Mean   :130.0
##                    3rd Qu.:215.1   3rd Qu.:130.4   3rd Qu.:130.2
##                    Max.   :216.3   Max.   :131.0   Max.   :131.1
##      Bottom            Top           Diagonal
##  Min.   : 7.200   Min.   : 7.70   Min.   :137.8
##  1st Qu.: 8.200   1st Qu.:10.10   1st Qu.:139.5
##  Median : 9.100   Median :10.60   Median :140.4
##  Mean   : 9.418   Mean   :10.65   Mean   :140.5
##  3rd Qu.:10.600   3rd Qu.:11.20   3rd Qu.:141.5
##  Max.   :12.700   Max.   :12.30   Max.   :142.4
```

```
#
# plot ggpairs
#
#install.packages("GGally")
library(GGally)
ggpairs(df0, mapping = aes(col = Status)) + theme_bw()
```




```
#
# duplicate df then remove response Status
#
df <- df0
df$Status = NULL
pca = prcomp(df,center = TRUE, scale = TRUE)
str(pca)
```

```
## List of 5
##  $ sdev    : num [1:6] 1.716 1.131 0.932 0.671 0.518 ...
##  $ rotation: num [1:6, 1:6] 0.00699 -0.46776 -0.48668 -0.40676 -0.36789 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "Length" "Left" "Right" "Bottom" ...
##   .. ..$ : chr [1:6] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:6] 214.9 130.12 129.96 9.42 10.65 ...
##   ..- attr(*, "names")= chr [1:6] "Length" "Left" "Right" "Bottom" ...
##  $ scale   : Named num [1:6] 0.377 0.361 0.404 1.445 0.803 ...
##   ..- attr(*, "names")= chr [1:6] "Length" "Left" "Right" "Bottom" ...
##  $ x       : num [1:200, 1:6] -1.74 2.27 2.27 2.28 2.63 ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ : NULL
##   .. ..$ : chr [1:6] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

```
#
pca
```

```
## Standard deviations (1, .., p=6):
## [1] 1.7162629 1.1305237 0.9322192 0.6706480 0.5183405 0.4346031
##
## Rotation (n x k) = (6 x 6):
##                   PC1         PC2         PC3        PC4         PC5         PC6
## Length    0.006987029 -0.81549497  0.01768066  0.5746173 -0.0587961  0.03105698
## Left     -0.467758161 -0.34196711 -0.10338286 -0.3949225  0.6394961 -0.29774768
## Right    -0.486678705 -0.25245860 -0.12347472 -0.4302783 -0.6140972  0.34915294
## Bottom   -0.406758327  0.26622878 -0.58353831  0.4036735 -0.2154756 -0.46235361
## Top      -0.367891118  0.09148667  0.78757147  0.1102267 -0.2198494 -0.41896754
## Diagonal  0.493458317 -0.27394074 -0.11387536 -0.3919305 -0.3401601 -0.63179849
```

```
#
# eigenvalues (square-root)
#
pca$sdev^2
```

```
## [1] 2.9455582 1.2780838 0.8690326 0.4497687 0.2686769 0.1888799
```

```
#
# eigenvectors (loading vectors)
#
pca$rotation
```

```
##                   PC1         PC2         PC3        PC4         PC5         PC6
## Length    0.006987029 -0.81549497  0.01768066  0.5746173 -0.0587961  0.03105698
## Left     -0.467758161 -0.34196711 -0.10338286 -0.3949225  0.6394961 -0.29774768
## Right    -0.486678705 -0.25245860 -0.12347472 -0.4302783 -0.6140972  0.34915294
## Bottom   -0.406758327  0.26622878 -0.58353831  0.4036735 -0.2154756 -0.46235361
## Top      -0.367891118  0.09148667  0.78757147  0.1102267 -0.2198494 -0.41896754
## Diagonal  0.493458317 -0.27394074 -0.11387536 -0.3919305 -0.3401601 -0.63179849
```

```
#
# transformed data (scores vectors)
#
pca$x[1:16,]
```

```
##              PC1         PC2        PC3         PC4          PC5         PC6
##  [1,] -1.7430272 -1.64669605 -1.4201973 -2.74796911  0.003293759  0.60202200
##  [2,]  2.2686248  0.53744461 -0.5313151 -0.65735578 -0.158171742  0.45654268
##  [3,]  2.2717009  0.10740754 -0.7156191 -0.34083839 -0.453880889 -0.04532905
##  [4,]  2.2778385  0.08743490  0.6041176 -0.39182554 -0.282913485 -0.05543875
##  [5,]  2.6255397 -0.03909779 -3.1883837  0.42401683 -0.277502895  0.72026433
##  [6,] -0.7565089 -3.08101359 -0.7845117 -0.59803217  0.192757017 -0.10529393
##  [7,]  2.5121235 -1.22391424 -0.2430283  0.92666684 -0.620993957  0.76238069
##  [8,]  2.7021533  1.13199022  1.1859845 -0.25489648  0.245874500 -0.23937010
##  [9,]  2.0331111  0.31369320  0.9797961  0.29444051 -1.221076857 -0.19564482
## [10,] -0.3169163 -1.30244988 -0.7420246 -0.43024244  0.070445196  0.38252580
## [11,] -0.2568427 -1.82641661  1.3465350 -0.78169056 -0.541458564 -0.68333341
## [12,]  2.5260508 -0.37212246  0.6730065  0.28633636 -0.800276738 -0.09160207
```

```
## [13,]  0.3072104 -1.55883089  0.5979640 -0.61394121  1.611035676 -0.93739972
## [14,]  1.7416620  0.40667411  1.0081498 -0.42434228 -0.497446935 -0.13768982
## [15,]  1.5785992 -0.68420723  0.8616917 -0.08046661 -0.207778188 -0.27229681
## [16,]  2.0942280  0.72779860 -2.0509052 -0.79377044  0.006931890  0.64478237
```

```
#
# variance ratios
#
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.7163 1.1305 0.9322 0.67065 0.51834 0.43460
## Proportion of Variance 0.4909 0.2130 0.1448 0.07496 0.04478 0.03148
## Cumulative Proportion  0.4909 0.7039 0.8488 0.92374 0.96852 1.00000
```

```
#
str(summary(pca))
```

```
## List of 6
##  $ sdev      : num [1:6] 1.716 1.131 0.932 0.671 0.518 ...
##  $ rotation  : num [1:6, 1:6] 0.00699 -0.46776 -0.48668 -0.40676 -0.36789 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "Length" "Left" "Right" "Bottom" ...
##   .. ..$ : chr [1:6] "PC1" "PC2" "PC3" "PC4" ...
##  $ center    : Named num [1:6] 214.9 130.12 129.96 9.42 10.65 ...
##   ..- attr(*, "names")= chr [1:6] "Length" "Left" "Right" "Bottom" ...
##  $ scale     : Named num [1:6] 0.377 0.361 0.404 1.445 0.803 ...
##   ..- attr(*, "names")= chr [1:6] "Length" "Left" "Right" "Bottom" ...
##  $ x         : num [1:200, 1:6] -1.74 2.27 2.27 2.28 2.63 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:6] "PC1" "PC2" "PC3" "PC4" ...
##  $ importance: num [1:3, 1:6] 1.716 0.491 0.491 1.131 0.213 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "Standard deviation" "Proportion of Variance" "Cumulative Proportion"
##   .. ..$ : chr [1:6] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "summary.prcomp"
```

```
#
summary(pca)$importance
```

```
##                            PC1      PC2       PC3      PC4       PC5       PC6
## Standard deviation     1.716263 1.130524 0.9322192 0.670648 0.5183405 0.4346031
## Proportion of Variance 0.490930 0.213010 0.1448400 0.074960 0.0447800 0.0314800
## Cumulative Proportion  0.490930 0.703940 0.8487800 0.923740 0.9685200 1.0000000
```

```
#
d2 = data.frame(summary(pca)$importance)
names(d2)
```

```
## [1] "PC1" "PC2" "PC3" "PC4" "PC5" "PC6"
```

```
rownames(d2)
```

```
## [1] "Standard deviation"     "Proportion of Variance" "Cumulative Proportion"
```

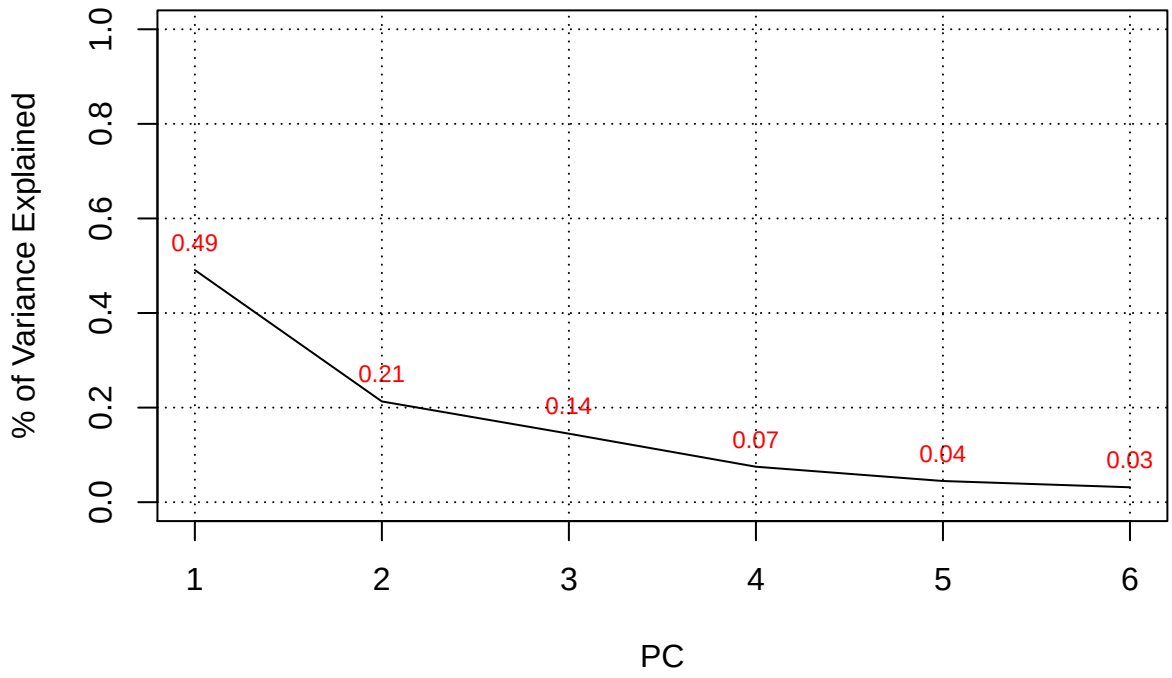
```
#
#
# PLOTTING PCA RESULTs
#
# proportion of variance explained (PVE) by each PC
#
aux = pca$sdev^2
PVE = aux/sum(aux)
PVE
```

```
## [1] 0.49092637 0.21301396 0.14483876 0.07496145 0.04477948 0.03147998
```

```
#
PVE_labels = round(PVE,2)
PVE_labels
```

```
## [1] 0.49 0.21 0.14 0.07 0.04 0.03
```

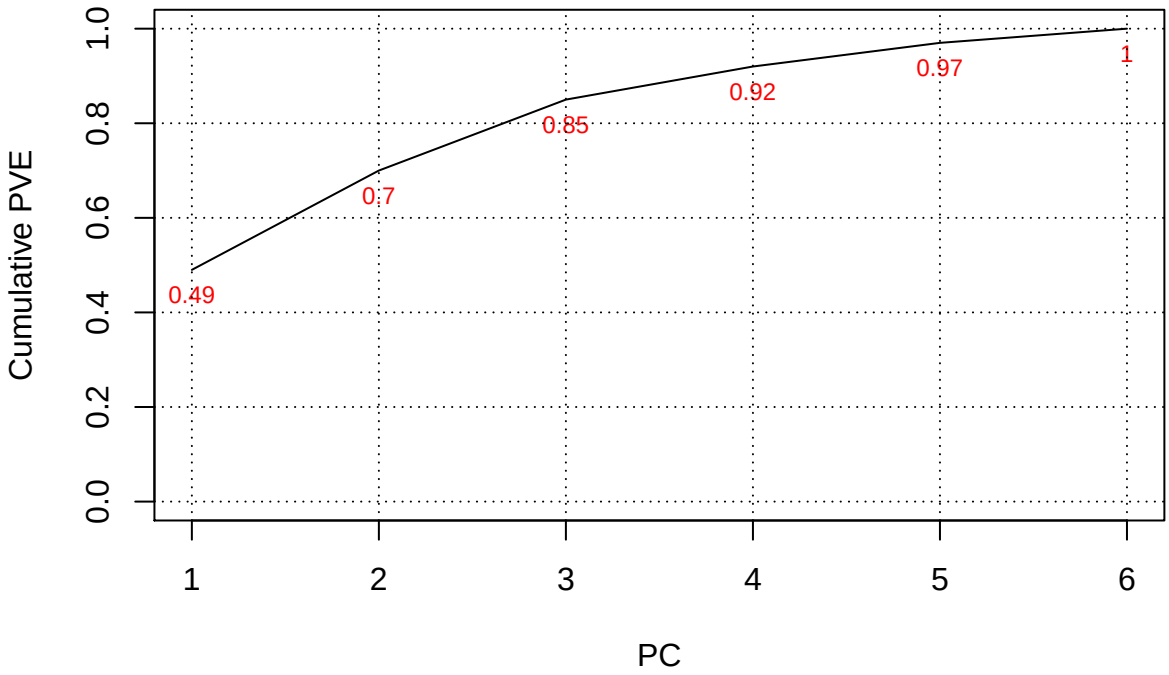
```
plot(PVE,xlab="PC",ylab="% of Variance Explained",type = "l",ylim=c(0,1))
text(PVE,labels=PVE_labels,cex=0.75,pos=3,offset=0.5,col="red")
grid(col="black")
```




```
#
# Cumulative proportion of variance explained (CPVE) by each PC
#
CPVE = cumsum(PVE)
CPVE
```

```
## [1] 0.4909264 0.7039403 0.8487791 0.9237405 0.9685200 1.0000000
```

```
CPVE = round(CPVE,2)
plot(CPVE,xlab="PC",ylab="Cumulative PVE",type = "l",ylim=c(0,1))
text(CPVE,labels=CPVE,cex=0.75,pos=1,offset=0.5,col="red")
grid(col="black")
```




```
#
#install.packages("factoextra")
library(factoextra)
#
# get more from pca
#
pcaDat = get_pca(pca)
#
str(pcaDat)
```

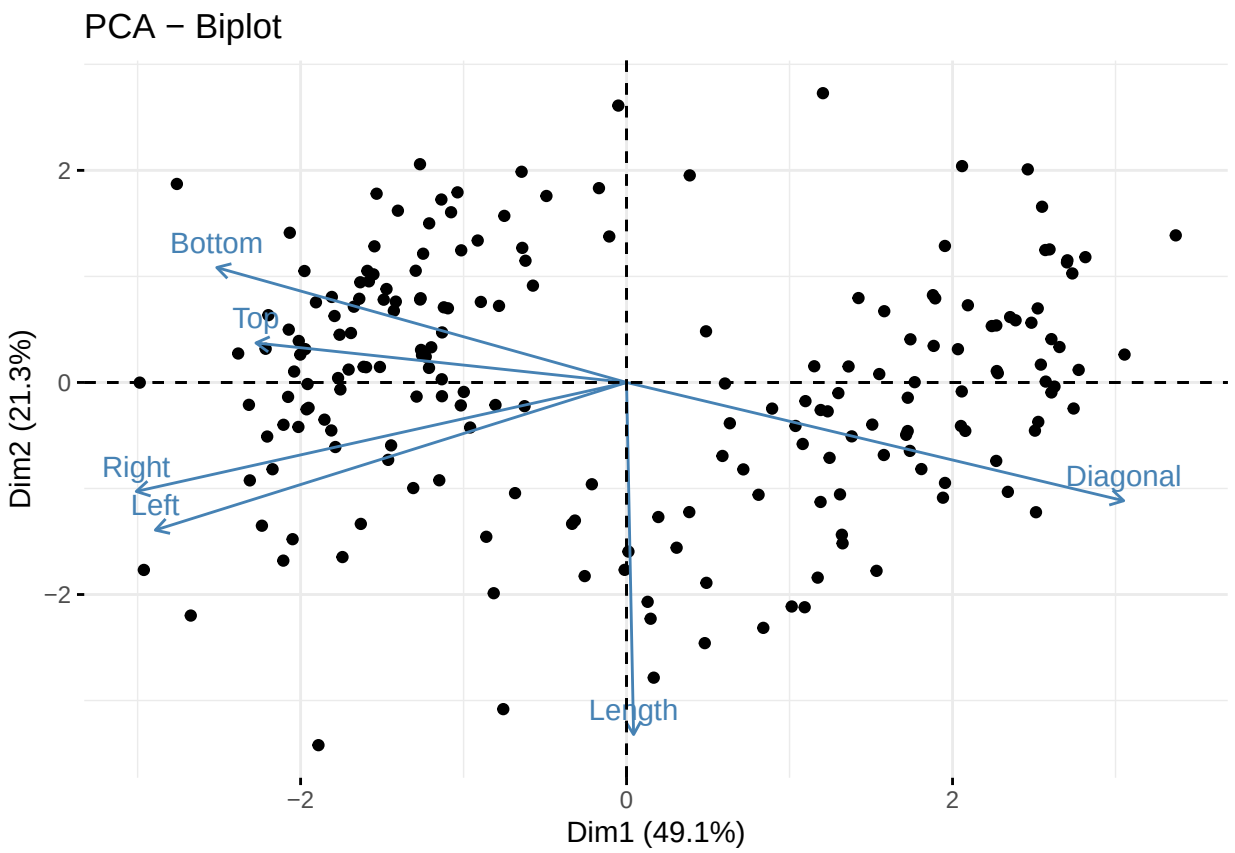
```
## List of 4
##  $ coord  : num [1:6, 1:6] 0.012 -0.803 -0.835 -0.698 -0.631 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "Length" "Left" "Right" "Bottom" ...
##   .. ..$ : chr [1:6] "Dim.1" "Dim.2" "Dim.3" "Dim.4" ...
##  $ cor    : num [1:6, 1:6] 0.012 -0.803 -0.835 -0.698 -0.631 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "Length" "Left" "Right" "Bottom" ...
##   .. ..$ : chr [1:6] "Dim.1" "Dim.2" "Dim.3" "Dim.4" ...
##  $ cos2   : num [1:6, 1:6] 0.000144 0.644481 0.697674 0.487349 0.398663 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "Length" "Left" "Right" "Bottom" ...
##   .. ..$ : chr [1:6] "Dim.1" "Dim.2" "Dim.3" "Dim.4" ...
##  $ contrib: num [1:6, 1:6] 0.00488 21.87977 23.68562 16.54523 13.53439 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "Length" "Left" "Right" "Bottom" ...
##   .. ..$ : chr [1:6] "Dim.1" "Dim.2" "Dim.3" "Dim.4" ...
##  - attr(*, "class")= chr [1:2] "factoextra" "pca_var"
```

```
#
#
#
#
```

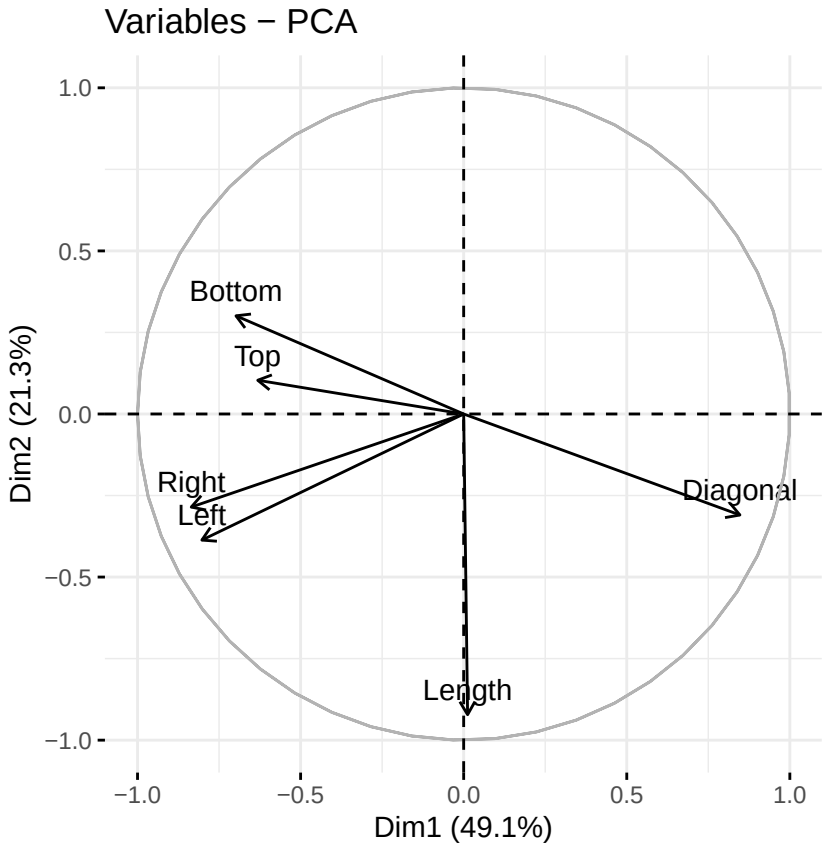
```
# correlation between transformed variables PCs and original variables Xs
#
pcaDat$coord
```

```
##                Dim.1      Dim.2       Dim.3       Dim.4      Dim.5       Dim.6
## Length    0.01199158 -0.9219364  0.01648225  0.38536590 -0.0304764  0.01349746
## Left     -0.80279596 -0.3866019 -0.09637548 -0.26485400  0.3314768 -0.12940207
## Right    -0.83526859 -0.2854104 -0.11510550 -0.28856524 -0.3183114  0.15174296
## Bottom   -0.69810421  0.3009779 -0.54398559  0.27072283 -0.1116898 -0.20094033
## Top      -0.63139786  0.1034278  0.73418921  0.07392333 -0.1139569 -0.18208461
## Diagonal  0.84690418 -0.3096965 -0.10615680 -0.26284740 -0.1763188 -0.27458160
```

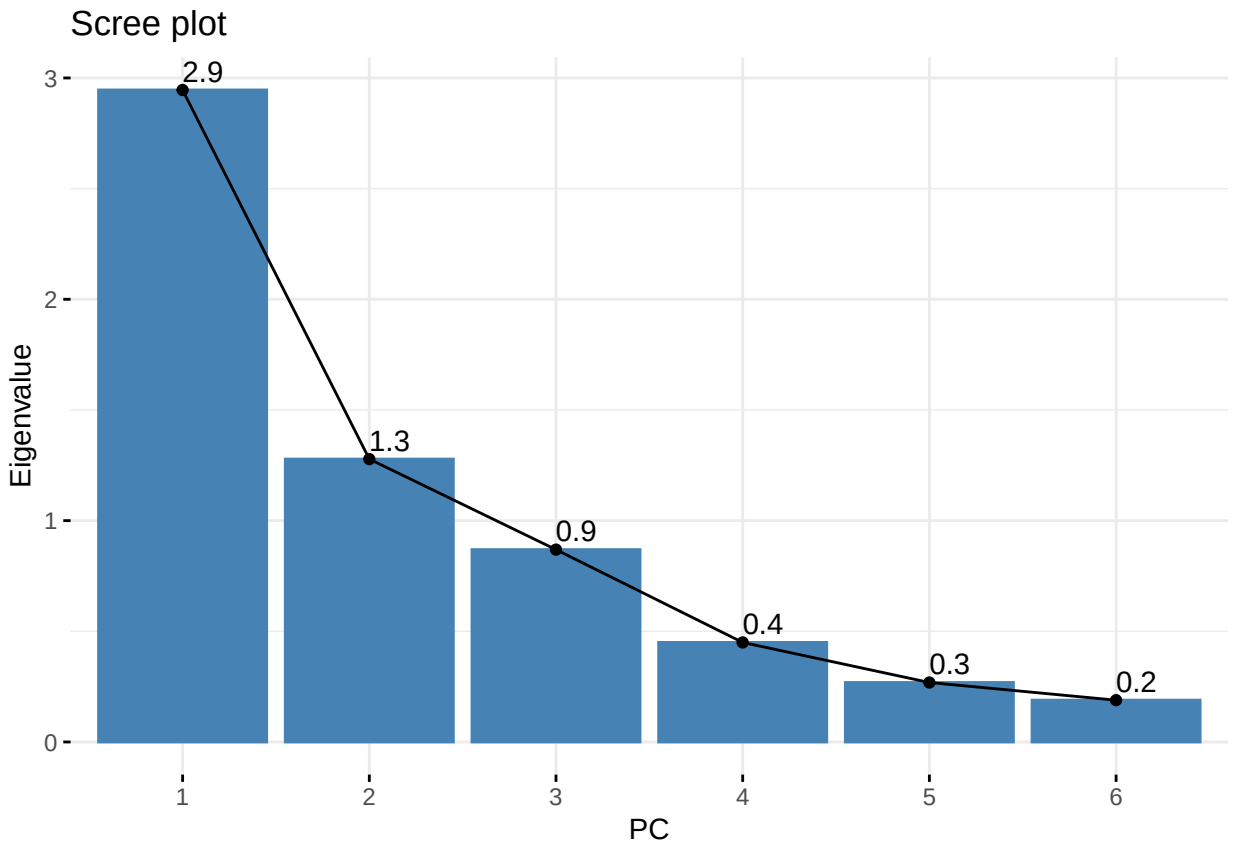
```
#
# biplot (label = 'var' to label vars but not each point)
#
fviz_pca_biplot(pca, label = "var")
```



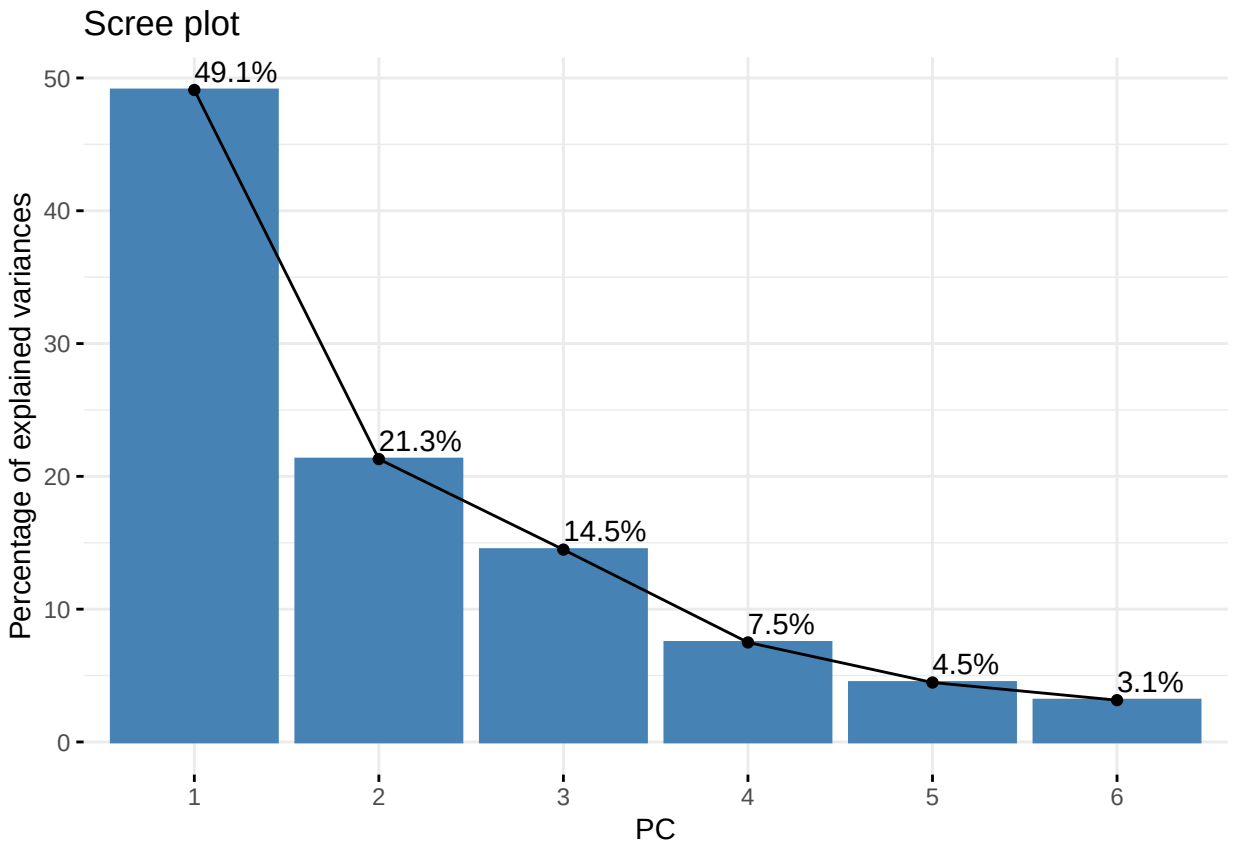

```
#
# variable loading plot
#
fviz_pca_var(pca)
```



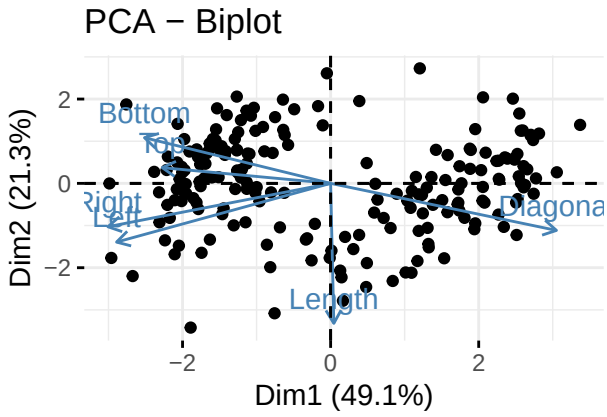

```
#
# eigenvalues plot
#
fviz_screeplot(pca, addlabels = TRUE, choice = "eigenvalue",xlab="PC")
```

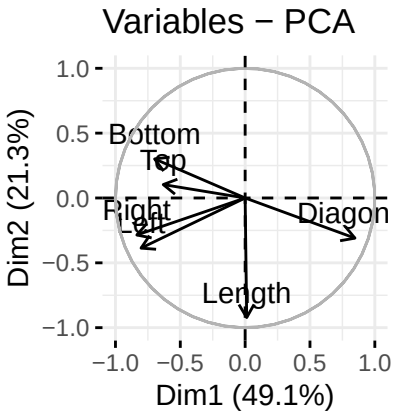


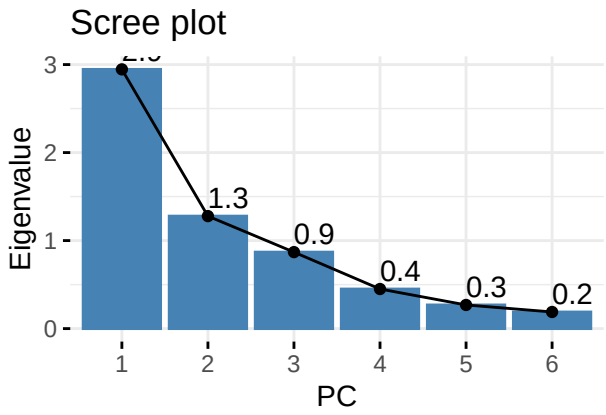

```
#
# explained variance plot
#
fviz_screeplot(pca, addlabels = TRUE, choice = "variance",xlab="PC")
```

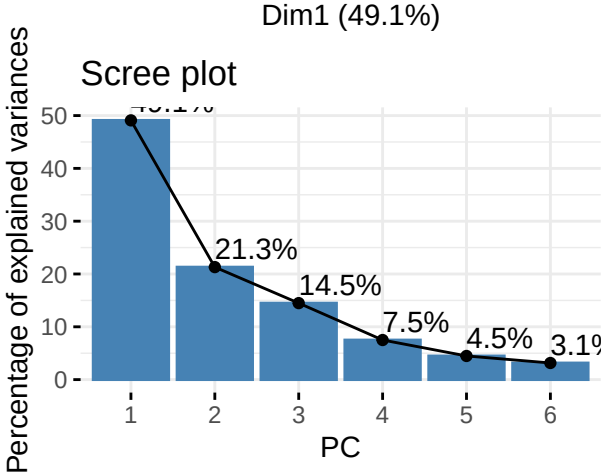



```
#
# 4in1 plot
#
p1 = fviz_pca_biplot(pca, label = "var")
p2 = fviz_pca_var(pca)
p3 = fviz_screeplot(pca, addlabels = TRUE, choice = "eigenvalue",xlab="PC")
p4 = fviz_screeplot(pca, addlabels = TRUE, choice = "variance",xlab="PC")
gridExtra::grid.arrange(p1,p2,p3,p4)
```

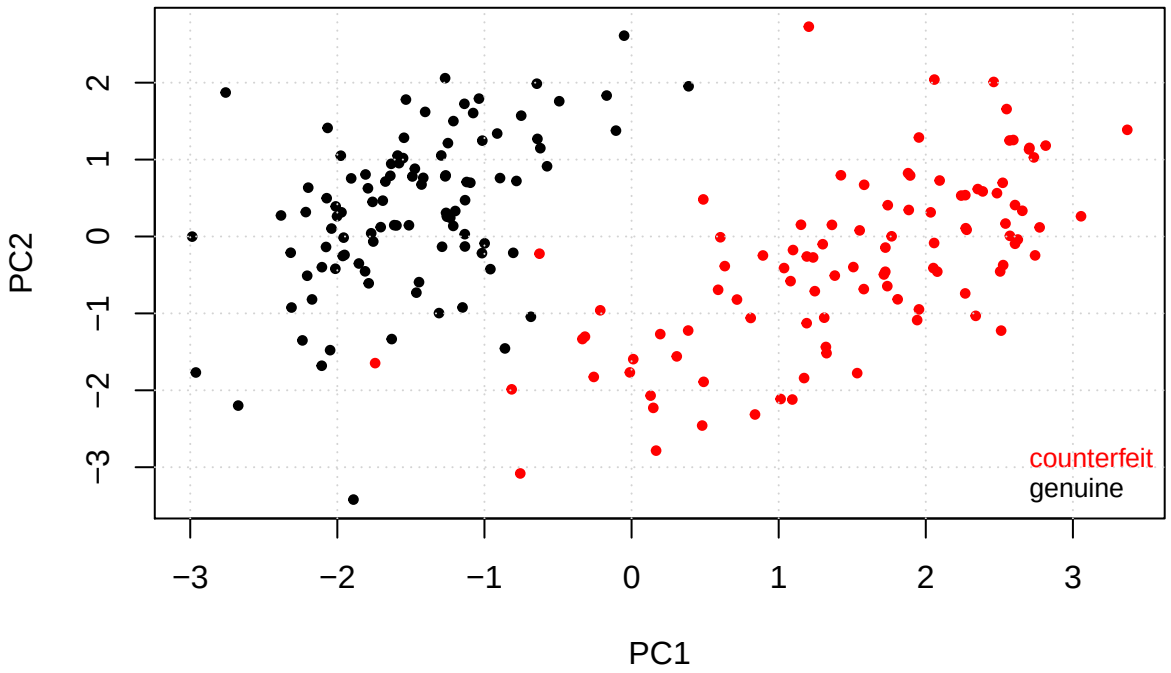




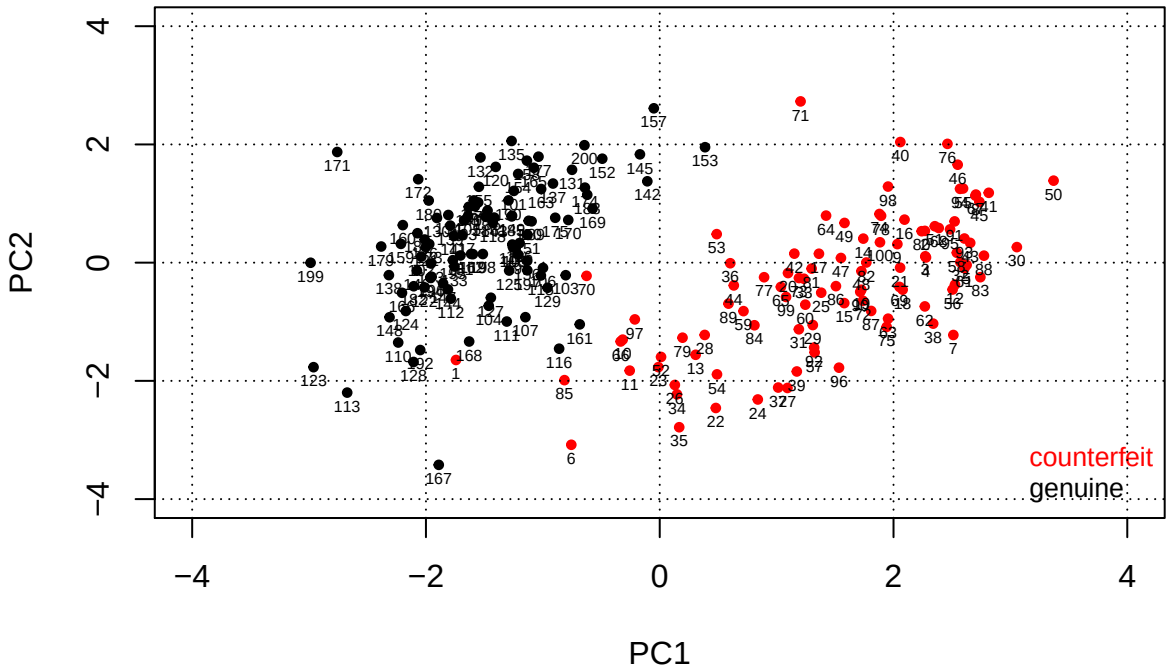





```
# or
# ggpubr::ggarrange(p1,p2,p3,p4)
#
# plot PC scores
#
PC1 = pca$x[,1]
PC2 = pca$x[,2]
color = df0$Status
legends = levels(df0$Status)
#
plot(PC2~PC1,pch=19,cex=0.6,col=color)
legend("bottomright",legend = legends,col = c("red","black"),bty = "n",
       pt.cex = 1,cex = 0.8,text.col = c("red","black"),inset = c(0,0))
grid()
```




```
#
# add point labels
#
bounds = c(-4,4)
plot(PC2~PC1,pch=19,cex=0.6,col=color,xlim=bounds,ylim=bounds)
text(PC2~PC1,labels=rownames(df),cex=0.5,pos=1,offset=0.25)
legend("bottomright",legend = legends,col = c("red","black"),bty = "n",
       pt.cex = 1,cex = 0.8,text.col = c("red","black"),inset = c(0,0))
grid(col="black")
```



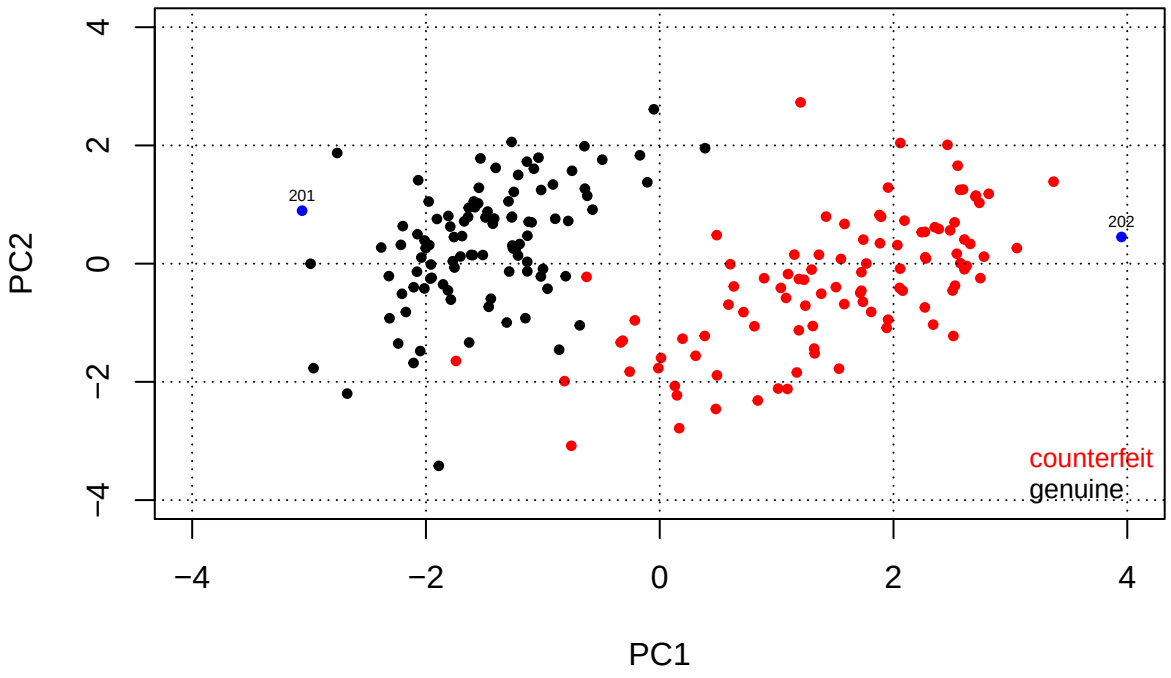
```
#
# classify two new banknotes
#
row1 = c(214,131,131,9,10,138)
row2 = c(215,130,128,8.5,9.5,142)
matrix = rbind(row1,row2)
newval = data.frame(matrix,row.names = c(201,202))
names(newval) = c("Length","Left","Right","Bottom","Top","Diagonal")
newval
```

```
##     Length Left Right Bottom  Top Diagonal
## 201    214  131   131    9.0 10.0      138
## 202    215  130   128    8.5  9.5      142
```

```
#
# find PC values from newval Xs
#
predict(pca, newval)
```

```
##           PC1       PC2        PC3       PC4      PC5       PC6
## 201 -3.059632 0.8957296 -0.8364652 -2.800674 1.083673  1.938025
## 202  3.950743 0.4515414 -0.2701876  1.444860 2.746157 -1.519346
```

```
df2 = predict(pca, newval)
#
# add new points on PC plot
#
bounds = c(-4,4)
plot(PC2~PC1,pch=19,cex=0.6,col=color,xlim=bounds,ylim=bounds)
points(PC2~PC1,df2,pch=19,cex=0.6,col="blue")
text(PC2~PC1,df2,labels=rownames(df2),cex=0.5,pos=3,offset=0.25)
legend("bottomright",legend = legends,col = c("red","black"),bty = "n",
       pt.cex = 1,cex = 0.8,text.col = c("red","black"),inset = c(0,0))
grid(col="black")
```



```
#
```