Réalisé par : Zyad Benchrifa

On va faire une analyse sur jeu de données obtenu à partir de mesures de billets de banque. Il se compose de 200 mesures. La première moitié de ces mesures proviennent de billets de banque authentiques. L'autre moitié provient de billets contrefaits. Les valeurs ont été mesurées en millimètres.

Importons maintenant l'ensemble de données à l'aide de la fonction read.csv(). J'ai déjà enregistré le fichier de données au format CSV (fichier délimité par des virgules) dans le répertoire de travail. L'argument file spécifie le nom du fichier avec l'extension CSV.

```
data = read.csv(file = "bank2.csv",
                header = TRUE)
head(data)
```

Résultat :

	note.type	Length	Left	Right	Bottom	Top	Diagonal
1	counterfeit	214.8	131.0	131.1	9.0	9.7	141.0
2	counterfeit	214.6	129.7	129.7	8.1	9.5	141.7
3	counterfeit	214.8	129.7	129.7	8.7	9.6	142.2
4	counterfeit	214.8	129.7	129.6	7.5	10.4	142.0
5	counterfeit	215.0	129.6	129.7	10.4	7.7	141.8
6	counterfeit	215.7	130.8	130.5	9.0	10.1	141.4

Première étape : La matrice des corrélations ; l'ACP essaie d'obtenir les caractéristiques avec une variance maximale et la variance est élevée pour les caractéristiques de magnitude élevée.

```
data.scaled <- scale(x = data[, -1],
                     center = TRUE,
                     scale = TRUE)
head(data.scaled)
```

La matrice :

	Length	Left	Right	Bottom	Top	Diagonal
[1,]	-0.2549435	2.433346	2.8299417	-0.2890067	-1.1837648	0.4482473
[2,]	-0.7860757	-1.167507	-0.6347880	-0.9120152	-1.4328473	1.0557460
[3,]	-0.2549435	-1.167507	-0.6347880	-0.4966762	-1.3083061	1.4896737
[4,]	-0.2549435	-1.167507	-0.8822687	-1.3273542	-0.3119759	1.3161027
[5,]	0.2761888	-1.444496	-0.6347880	0.6801176	-3.6745902	1.1425316
[6,]	2.1351516	1.879368	1.3450576	-0.2890067	-0.6855997	0.7953894

La deuxième étape est l'extraction des composants principaux. Le moyen le plus simple consiste à utiliser les fonctions prcomp() ou princomp() à l'aide du package stats.

Ici, je vais extraire les composants principaux en utilisant la fonction eigen() du package de base et la fonction prcomp() du package stats. J'utiliserai d'abord la fonction eigen() pour calculer les valeurs propres et les vecteurs propres, appliquez la fonction eigen à la matrice de covariance de l'ensemble de données pour obtenir les valeurs propres (eigenvalues) et les vecteurs propres (eigenvectors) et enfin utilisez la fonction print() pour voir les résultats.

```
e = eigen(cov(data[, -1]))
print(e)
```

Résultat :

```
eigen() decomposition
$values
[1] 3.00030487 0.93562052 0.24341371 0.19465874 0.08521185 0.03551468

$vectors
            [,1]        [,2]       [,3]       [,4]        [,5]        [,6]
[1,] -0.04377427  0.01070966 -0.3263165 -0.5616918  0.75257278  0.09809807
[2,]  0.11216159  0.07144697 -0.2589614 -0.4554588 -0.34680082 -0.76651197
[3,]  0.13919062  0.06628208 -0.3447327 -0.4153296 -0.53465173  0.63169678
[4,]  0.76830499 -0.56307225 -0.2180222  0.1861082  0.09996771 -0.02221711
[5,]  0.20176610  0.65928988 -0.5566857  0.4506985  0.10190229 -0.03485874
[6,] -0.57890193 -0.48854255 -0.5917628  0.2584483 -0.08445895 -0.04567946
```

On va utilisez la même fonction pour obtenir les valeurs propres et les vecteurs propres des données centrées-réduites (scaled). La fonction print() affichera les résultats des données centrées-réduites.

```
e.scaled = eigen(cov(data.scaled))
print(e.scaled)
```

Résultat :

```
eigen() decomposition
$values
[1] 2.9455582 1.2780838 0.8690326 0.4497687 0.2686769 0.1888799

$vectors
             [,1]        [,2]        [,3]       [,4]       [,5]        [,6]
[1,] -0.006987029  0.81549497 -0.01768066  0.5746173 -0.0587961 -0.03105698
[2,]  0.467758161  0.34196711  0.10338286 -0.3949225  0.6394961  0.29774768
[3,]  0.486678705  0.25245860  0.12347472 -0.4302783 -0.6140972 -0.34915294
[4,]  0.406758327 -0.26622878  0.58353831  0.4036735 -0.2154756  0.46235361
[5,]  0.367891118 -0.09148667 -0.78757147  0.1102267 -0.2198494  0.41896754
[6,] -0.493458317  0.27394074  0.11387536 -0.3919305 -0.3401601  0.63179849
```

On peut voir la différence entre les vecteurs propres et les valeurs propres des données centrées-réduites et non centrées-réduites. Si on regarde la composante $values à partir du résultat de la décomposition propre, la variation du premier composant PC1 dans les données centrées-réduites est réduite par rapport aux données non centrées-réduites. Cette réduction a entraîné une légère augmentation de la variation du reste des composants dans les données centrées-réduites.

On va maintenant calculer la proportion des écarts. Pour cela, on va attachez d'abord la composante de valeurs de la décomposition propre à partir des données centrées-réduites à l'aide du signe dollar (e.scaled$values) pour obtenir les variances. Comme la variation totale est de 6 (six composantes et chacune avec une variance unitaire), on va divisez donc chaque variance par la variation totale pour obtenir la proportion de chaque variance.

Appliquez la fonction cumsum() à la proportion de variance obtenue précédemment pour obtenir les proportions cumulées.

```
e.scaled$values
e.scaled$values/6
cumsum(e.scaled$values/6)
```

```
[1] 2.9455582 1.2780838 0.8690326 0.4497687 0.2686769 0.1888799
[1] 0.49092637 0.21301396 0.14483876 0.07496145 0.04477948 0.03147998
[1] 0.4909264 0.7039403 0.8487791 0.9237405 0.9685200 1.0000000
```

Les résultats montrent que le premier CP explique 49% de la variation. Les trois premiers CP expliquent une proportion cumulée de 85 % de la variation totale.

On Calcule maintenant les vecteurs des composantes principales. Pour cela, on va multiplier la matrice data.scaled avec la matrice de vecteurs propres centrées-réduites (e.scaled$vectors) en utilisant l'opérateur de multiplication de matrice %*%. La fonction Head imprimera les six premières lignes de la matrice des composants.

```
data.pc  = as.matrix(data.scaled) %*% e.scaled$vectors
head(data.pc)
```

Résultat :

```
           [,1]        [,2]       [,3]       [,4]         [,5]        [,6]
[1,]  1.7430272  1.64669605  1.4201973 -2.7479691  0.003293759 -0.60202200
[2,] -2.2686248 -0.53744461  0.5313151 -0.6573558 -0.158171742 -0.45654268
[3,] -2.2717009 -0.10740754  0.7156191 -0.3408384 -0.453880889  0.04532905
[4,] -2.2778385 -0.08743490 -0.6041176 -0.3918255 -0.282913485  0.05543875
[5,] -2.6255397  0.03909779  3.1883837  0.4240168 -0.277502895 -0.72026433
[6,]  0.7565089  3.08101359  0.7845117 -0.5980322  0.192757017  0.10529393
```

Calculons maintenant les mêmes composants principaux en utilisant le package de statistiques.

```
require(stats)
pc <- prcomp(x = data[, -1],
             center = TRUE,
             scale. = TRUE)
head(pc$x)
```

Résultat :

```
            PC1         PC2        PC3        PC4          PC5         PC6
[1,] -1.7430272 -1.64669605 -1.4201973 -2.7479691  0.003293759  0.60202200
[2,]  2.2686248  0.53744461 -0.5313151 -0.6573558 -0.158171742  0.45654268
[3,]  2.2717009  0.10740754 -0.7156191 -0.3408384 -0.453880889 -0.04532905
[4,]  2.2778385  0.08743490  0.6041176 -0.3918255 -0.282913485 -0.05543875
[5,]  2.6255397 -0.03909779 -3.1883837  0.4240168 -0.277502895  0.72026433
[6,] -0.7565089 -3.08101359 -0.7845117 -0.5980322  0.192757017 -0.10529393
```

On utilise la fonction summary() pour imprimer les mesures des composants liées à la variance.

```
summary(pc)
```

```
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
```

On peut voir que le premier CP explique 49% de la variation. Les trois premiers CP expliquent une proportion cumulée de 84.8 % de la variation totale.

Maintenant, affichons la variance expliquée par les PC en utilisant la fonction plot() et la fonction screeplot().

Code :

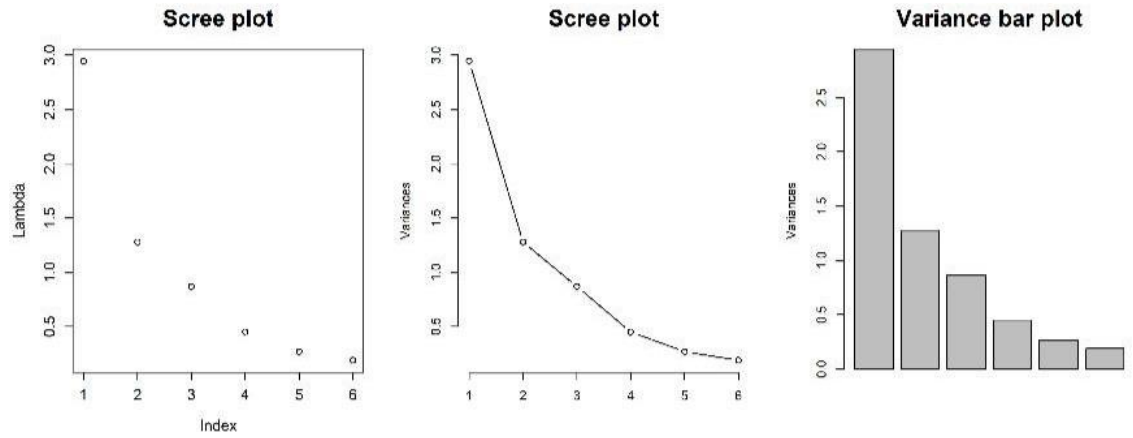
```
par(mfrow=c(1, 3))

plot(e.scaled$values, xlab = "Index", ylab = "Lambda",
     main = "Scree plot",
     cex.lab = 1.2, cex.axis = 1.2, cex.main = 1.8)

screeplot(x= pc, type = "line", main = "Scree plot", cex.main = 1.8)

plot(pc, main = "Variance bar plot", cex.main = 1.8)
```

Résultat :



La majeure partie de la variation est due aux deux premières composantes principales (70 %).

Corrélation des variables d'origine avec les CP

```
mean.dev <- scale(data[, -1],
                  center = TRUE,
                  scale = FALSE)
head(mean.dev)
```

```
     Length    Left   Right  Bottom     Top Diagonal
[1,] -0.096  0.8785  1.1435 -0.4175 -0.9505   0.5165
[2,] -0.296 -0.4215 -0.2565 -1.3175 -1.1505   1.2165
[3,] -0.096 -0.4215 -0.2565 -0.7175 -1.0505   1.7165
[4,] -0.096 -0.4215 -0.3565 -1.9175 -0.2505   1.5165
[5,]  0.104 -0.5215 -0.2565  0.9825 -2.9505   1.3165
[6,]  0.804  0.6785  0.5435 -0.4175 -0.5505   0.9165
```

Calculons maintenant les CP pour les données non centrées-réduites. Multipliez les écarts moyens Mean.dev avec les e$vectors en utilisant l'opérateur pipe de multiplication matricielle. La fonction head() imprimera les six premières lignes des composants principaux.

```
mean.dev.pc <- mean.dev %*% e$vectors
head(mean.dev.pc)
```

```
           [,1]       [,2]        [,3]        [,4]        [,5]        [,6]
[1,] -0.5496481 -0.5063730 -0.27586457 -1.19372812 -1.17050345  0.05836252
[2,] -2.0186292 -0.6612636  0.50199746  0.01544486 -0.29113718  0.14582450
[3,] -1.8356755 -1.1753073 -0.04562915  0.18906546 -0.11268124  0.12578824
[4,] -2.4943672  0.1188916 -0.07652521  0.31613769 -0.08076370  0.07052799
[5,] -0.7013228 -3.1947667  0.83877387 -0.52104953  0.08262773  0.26879336
[6,] -0.8458460 -0.4824940 -0.77029691 -1.07530245 -0.09605941 -0.11128017
```

Pour voir la corrélation entre les CP et les variables d'origine, on va appliquer la fonction cor(). On commence par combiner les CP avec les variables de données à l'aide de la fonction cbind() avant d'appliquer la fonction de corrélation. La fonction d'impression pour l'objet r.cor résultera en une matrice de corrélation des PC et des variables de données en combinaison.

```
r.cor <- cor(cbind(mean.dev.pc, data[, -1]))
head(r.cor)
```

Résultat :

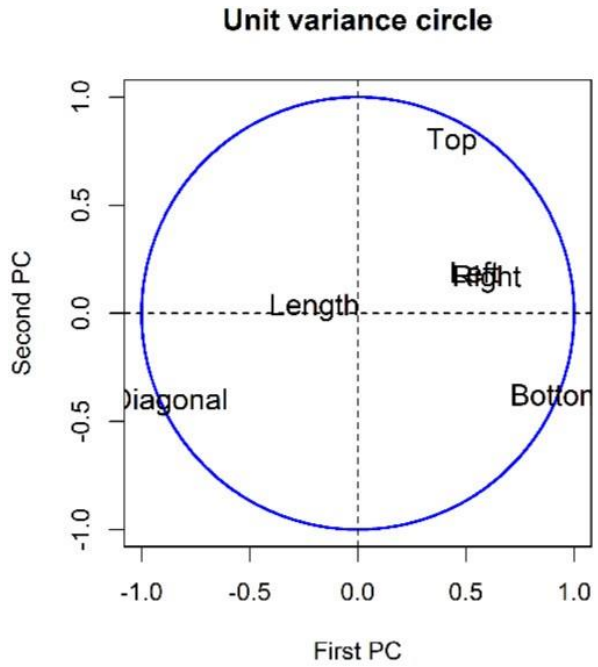
```
              1             2             3             4             5             6      Length       Left       Right
1  1.000000e+00 -7.014633e-18 -4.469949e-16 -7.976627e-16 -1.020254e-15 -6.521526e-16 -0.20136050  0.5381321  0.5966697
2 -7.014633e-18  1.000000e+00  1.983991e-16 -5.252503e-16 -1.038670e-16  5.101457e-16  0.02751049  0.1914237  0.1586673
3 -4.469949e-16  1.983991e-16  1.000000e+00  4.464784e-16 -1.835115e-16 -1.808638e-15 -0.42754733 -0.3538910 -0.4209169
4 -7.976627e-16 -5.252503e-16  4.464784e-16  1.000000e+00 -1.130514e-15 -1.357142e-15 -0.65812392 -0.5566063 -0.4534936
5 -1.020254e-15 -1.038670e-16 -1.835115e-16 -1.130514e-15  1.000000e+00 -2.570084e-15  0.58340637 -0.2804091 -0.3862445
6 -6.521526e-16  5.101457e-16 -1.808638e-15 -1.357142e-15 -2.570084e-15  1.000000e+00  0.04909498 -0.4001151  0.2946144
       Bottom         Top     Diagonal
1  0.921229423  0.435255426 -0.870231992
2 -0.377020918  0.794217722 -0.410109295
3 -0.074460283 -0.342054932 -0.253377217
4  0.056839988  0.247648882  0.098959622
5  0.020200457  0.037046504 -0.021396513
6 -0.002898297 -0.008181424 -0.007470889
```

Choisissons maintenant la matrice de corrélation des variables pour les deux premiers CP importants.

```
r1 = r.cor[7:12, 1:2]
print(r1)
```

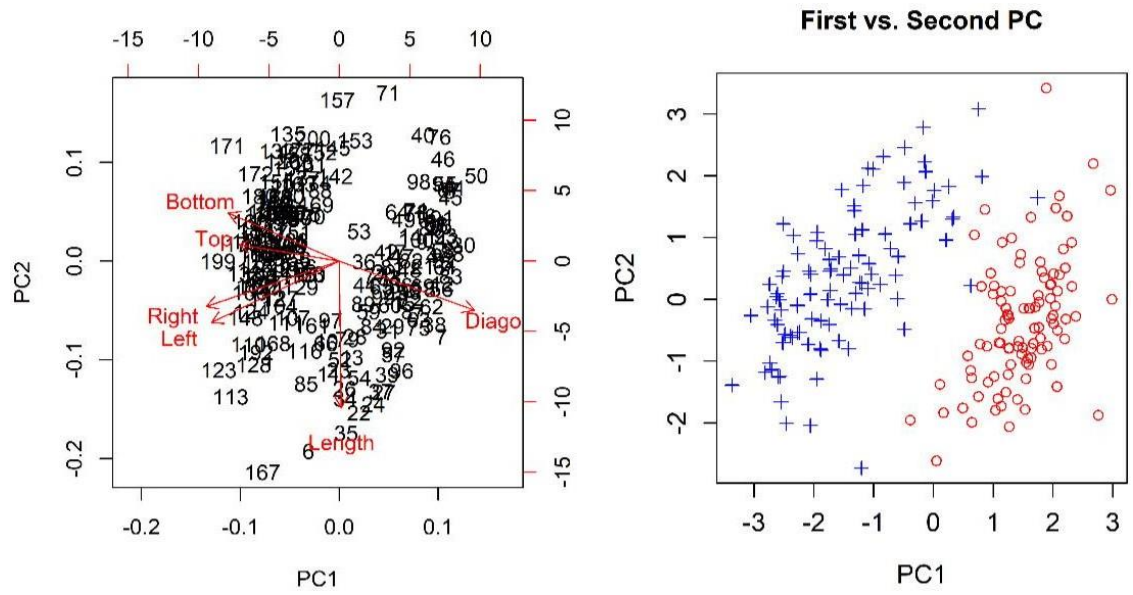
	1	2
Length	-0.2013605	0.02751049
Left	0.5381321	0.19142371
Right	0.5966697	0.15866725
Bottom	0.9212294	-0.37702092
Top	0.4352554	0.79421772
Diagonal	-0.8702320	-0.41010930

Affichage de la corrélation des variables d'origine avec les PC



La figure montre que les variables hautes, bas et diagonale correspondent à des corrélations proches de la périphérie du cercle. Ces variables sont bien expliquées par les deux premiers CP. CP1 est essentiellement la différence entre la variable du cadre inférieur et la diagonale. Le deuxième CP est mieux décrit par la différence entre la variable du cadre supérieur et la somme des variables du cadre inférieur et de la diagonale. Le pourcentage de variance de longueur, variable gauche et droite expliquée par les deux premiers CP est relativement faible.

Tracer les composants principaux



Si nous regardons simultanément les deux graphiques, cela montre que les billets de banque authentiques sont à peu près caractérisés par de grandes valeurs de variable inférieure et de plus petites valeurs de variable diagonale. Les billets de banque contrefaits montrent des valeurs plus élevées de la variable supérieure et des valeurs plus petites de la variable diagonale.