A3-Componentes Principales

Roel De la Rosa - A01197595

9/10/2022

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.1.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'pillar'
```

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

PARTE A

```
x1 <- c(2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2, 1, 1.5, 1.1)
x2 <- c(2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9)
M<- data.frame(x1,x2)
M
```

```
##     x1  x2
## 1  2.5 2.4
## 2  0.5 0.7
## 3  2.2 2.9
## 4  1.9 2.2
## 5  3.1 3.0
## 6  2.3 2.7
## 7  2.0 1.6
## 8  1.0 1.1
## 9  1.5 1.6
## 10 1.1 0.9
```

```
m1 <- c(rep(mean(x1),10))
m2 <- c(rep(mean(x2),10))
Mc <- data.frame(x1-m1,x2-m2)
Mc
```

```
##    x1...m1 x2...m2
## 1     0.69    0.49
## 2    -1.31   -1.21
## 3     0.39    0.99
## 4     0.09    0.29
## 5     1.29    1.09
## 6     0.49    0.79
## 7     0.19   -0.31
## 8    -0.81   -0.81
## 9    -0.31   -0.31
## 10   -0.71   -1.01
```

```
Mcov <- cov(Mc)
ValP <- eigen(Mcov)$values
VecP <- eigen(Mcov)$vectors

T_VecP = t(VecP)
T_Mc = t(Mc)
CP = T_VecP%*%T_Mc
rownames(CP) = c('CP1','CP2')
t(CP)
```

```
##               CP1         CP2
##  [1,]  0.82797019 -0.17511531
##  [2,] -1.77758033  0.14285723
##  [3,]  0.99219749  0.38437499
##  [4,]  0.27421042  0.13041721
##  [5,]  1.67580142 -0.20949846
##  [6,]  0.91294910  0.17528244
##  [7,] -0.09910944 -0.34982470
##  [8,] -1.14457216  0.04641726
##  [9,] -0.43804614  0.01776463
## [10,] -1.22382056 -0.16267529
```

El resultado que tenemos son dos componentes principales y podemos observar como es que cada variable afecta a dicho componente principal.

PARTE B

```
cpa <- prcomp(M, scale = TRUE)
names(cpa)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```
print("desviaciones estándar: ")
```

```
## [1] "desviaciones estándar: "
```

```
cpa$sdev
```

```
## [1] 1.3877785 0.2721594
```

```
print("medias: ")
```

```
## [1] "medias: "
```

```
print("center y scale dan las medias y desv estándar previa estandarización: ")
```

```
## [1] "center y scale dan las medias y desv estándar previa estandarización: "
```

```
cpa$center
```

```
##   x1   x2
## 1.81 1.91
```

```
cpa$scale
```

```
##        x1        x2
## 0.7852105 0.8464960
```

```
print("Los coeficientes de la combinación lineal normalizada de componete")
```

```
## [1] "Los coeficientes de la combinación lineal normalizada de componete"
```

```
cpa$rotation
```

```
##           PC1        PC2
## x1 -0.7071068  0.7071068
## x2 -0.7071068 -0.7071068
```

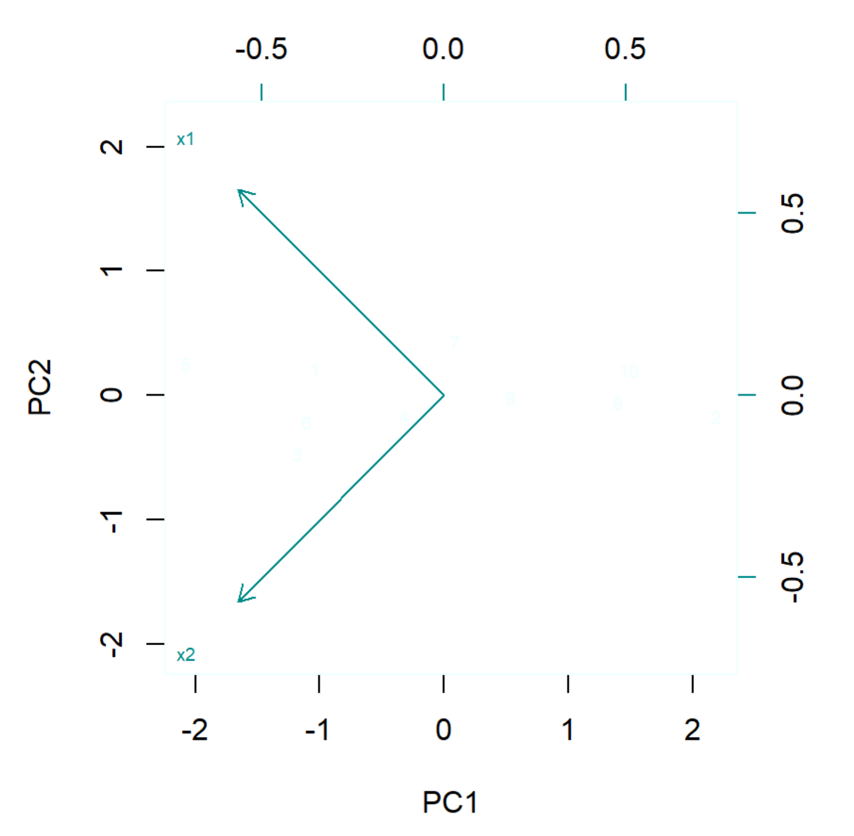
```
print("Los datos por sustituidos en la combinación lineal de vectores propios:")
```

```
## [1] "Los datos por sustituidos en la combinación lineal de vectores propios:"
```

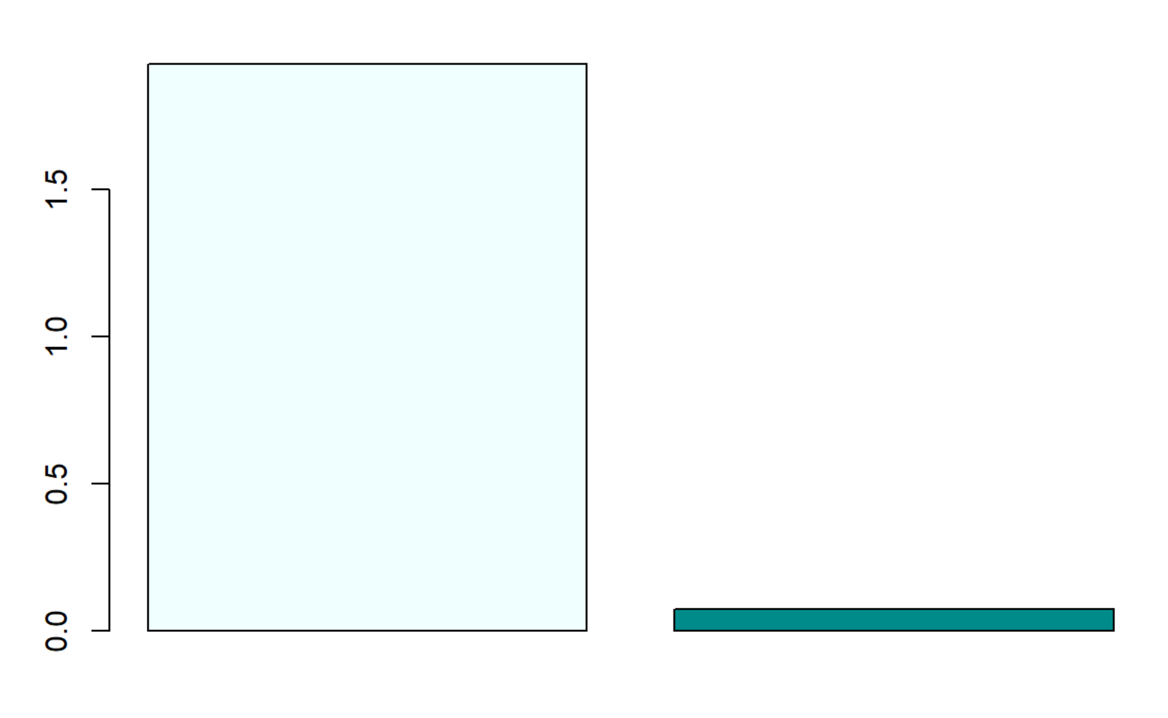
```
cpa$x
```

```
##               PC1         PC2
##  [1,] -1.03068029  0.21205314
##  [2,]  2.19045016 -0.16894230
##  [3,] -1.17818776 -0.47577321
##  [4,] -0.32329464 -0.16119898
##  [5,] -2.07219947  0.25117173
##  [6,] -1.10117414 -0.21865330
##  [7,]  0.08785251  0.43005447
##  [8,]  1.40605089 -0.05281009
##  [9,]  0.53811824 -0.02021127
## [10,]  1.48306451  0.20430982
```

```
biplot(x = cpa, scale = 0, cex =0.6, col = c("azure", "darkcyan"))
```




```
barplot(cpa$sdev^2, col = c("azure", "darkcyan"))
```

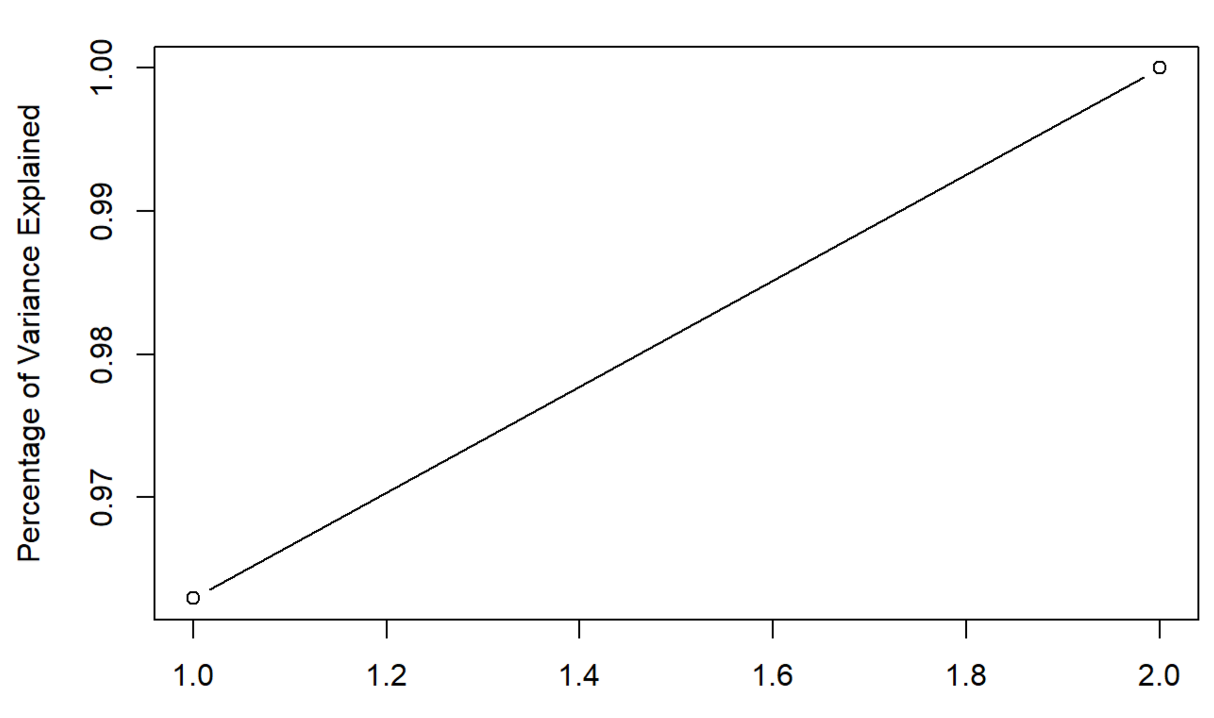



```
summary(cpa)
```

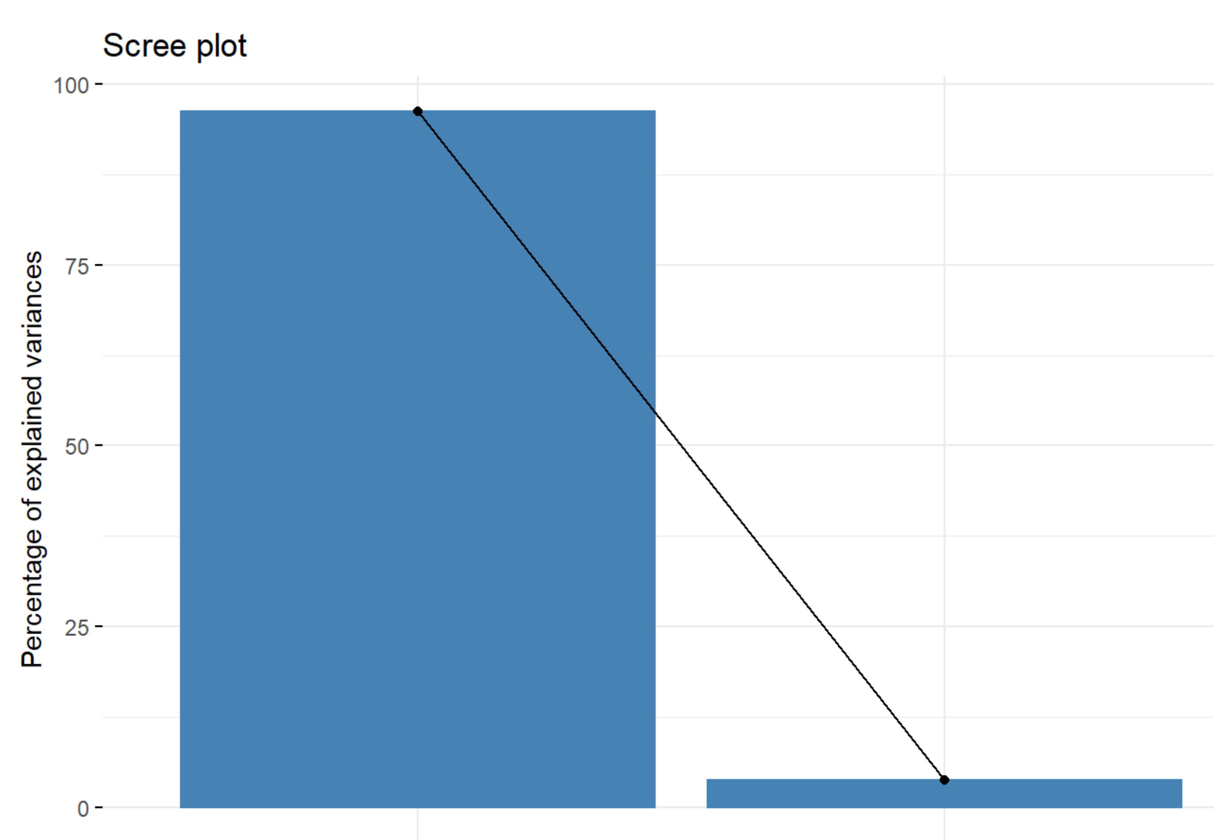
```
## Importance of components:
##                          PC1     PC2
## Standard deviation     1.388 0.27216
## Proportion of Variance 0.963 0.03704
## Cumulative Proportion  0.963 1.00000
```

```
variance = cpa$sdev^2 / sum(cpa$sdev^2)

plot(cumsum(variance), xlab = "Principal Component",
            ylab = "Percentage of Variance Explained",
            type = "b")
```

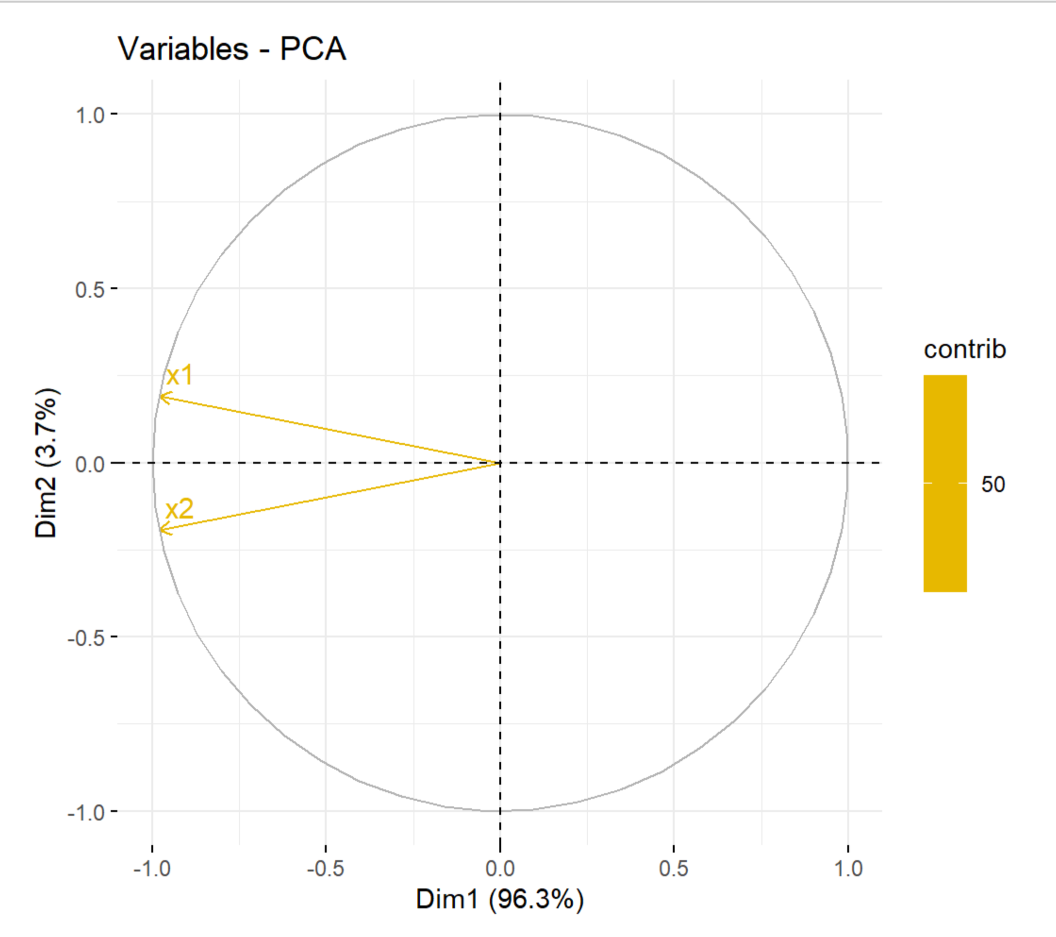



```
fviz_eig(cpa)
```



En estas dos gráficas podemos observar como es que cada componente principal comprende la variabilidad de los datos

```
fviz_pca_var(cpa,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```



Aquí podemos observar como es que cada vector se encuentra con respecto a las dimensiones de los componentes principales.