Principal Component Analysis of 6 vertical temperature sensors

Ana Laura Diedrichs

May 3rd, 2014

PRINCIPAL COMPONENT ANALISIS

librería a utilizar. Constantes para cargar el archivo de datos

```
library("FactoMineR")

SEP <- ";"
FILE_NAME <- "120726-minimal.csv"
```

Cargamos el dataset a memoria. Realizamos un vistazo de su contenido. Contiene 6 variables.

```
data <- as.data.frame(read.csv(FILE_NAME,sep=SEP))
dataset <- data[,5:10]
head(dataset)
```

```
##    s_0 s_0_4 s_0_75 s_1_50  s_2  s_3
## 1 18.2  17.3   17.4   17.0 17.8 17.8
## 2 18.4  17.2   17.3   16.8 17.7 17.5
## 3 18.4  17.1   17.2   16.7 17.3 17.2
## 4 18.4  17.0   17.1   16.6 17.1 17.0
## 5 18.5  16.9   17.1   16.5 17.1 16.9
## 6 18.5  16.8   16.9   16.4 17.0 16.7
```

Renombramos las columnas para mayor claridad

```
names.col <- c("time","s_0","s_0_4","s_0_75","s_1_50","s_2","s_3")
data.table <- as.data.frame(cbind(paste(data[,2],data[,3],sep=" "),data[,5:10]))
colnames(data.table)<- names.col
```

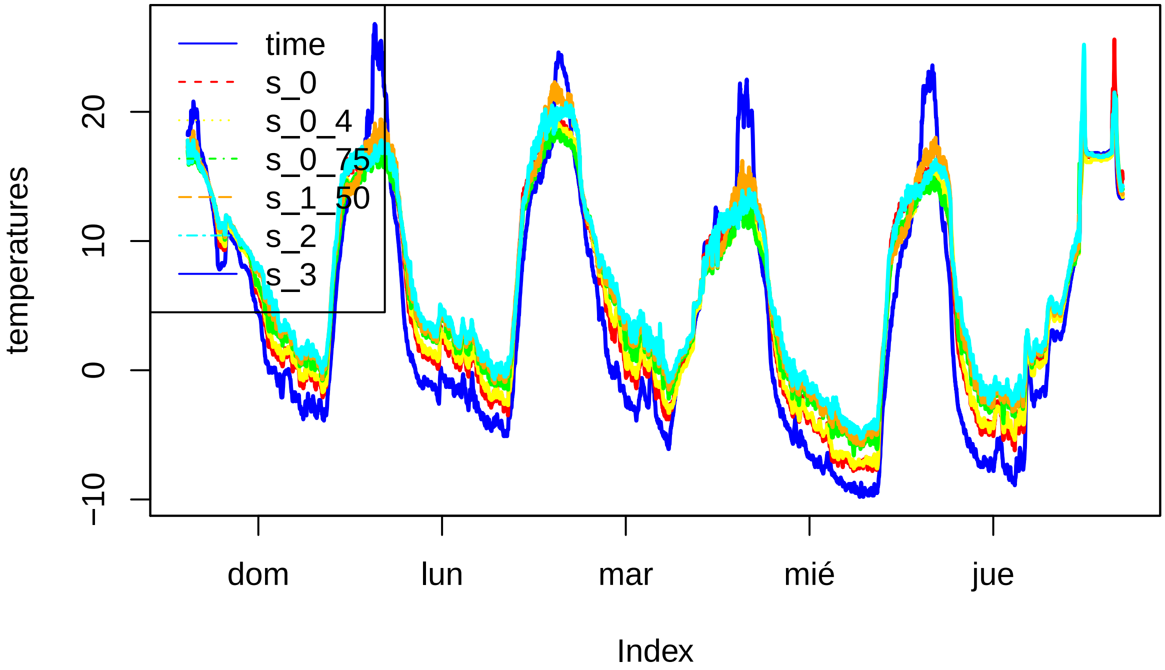
Crearemos un objeto timeSeries (xts) para manipular mejor los datos

```
library(lubridate)
t <- ymd_hms(data.table[,1])  #<--convierte string a date
library(xts)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
ts<- xts(x=data.table[,2:7],order.by=t)
my.color<- c("blue","red","yellow","green","orange","cyan")

plot(as.zoo(ts), plot.type="s", col=my.color, lty=1,lwd=2,ylab="temperatures")
legend(x="topleft", legend=names.col, col=my.color,lty=1:6)
```



```
#TODO verificar que la leyenda del gráfico corresponda con las variables
```

A continuación graficamos la serie temporal de los sensores por separado para mayor claridad

```
 require(graphics)
library(timeSeries)
```

```
## Loading required package: timeDate
##
## Attaching package: 'timeSeries'
##
## The following object is masked from 'package:zoo':
##
##     time<-
```

```
par(mfrow=c(1, 1))
#línea que marca el CERO
lines2 <- function(X, Y, type, xlab, ylab, col, pch, lty, lwd, cex) {
  lines(x=X, y=Y, col=col)
  abline(h=0, col = "brown", lwd=2)
}
plot(as.zoo(ts), plot.type="m", col = .colorwheelPalette(3),panel=lines2)
```

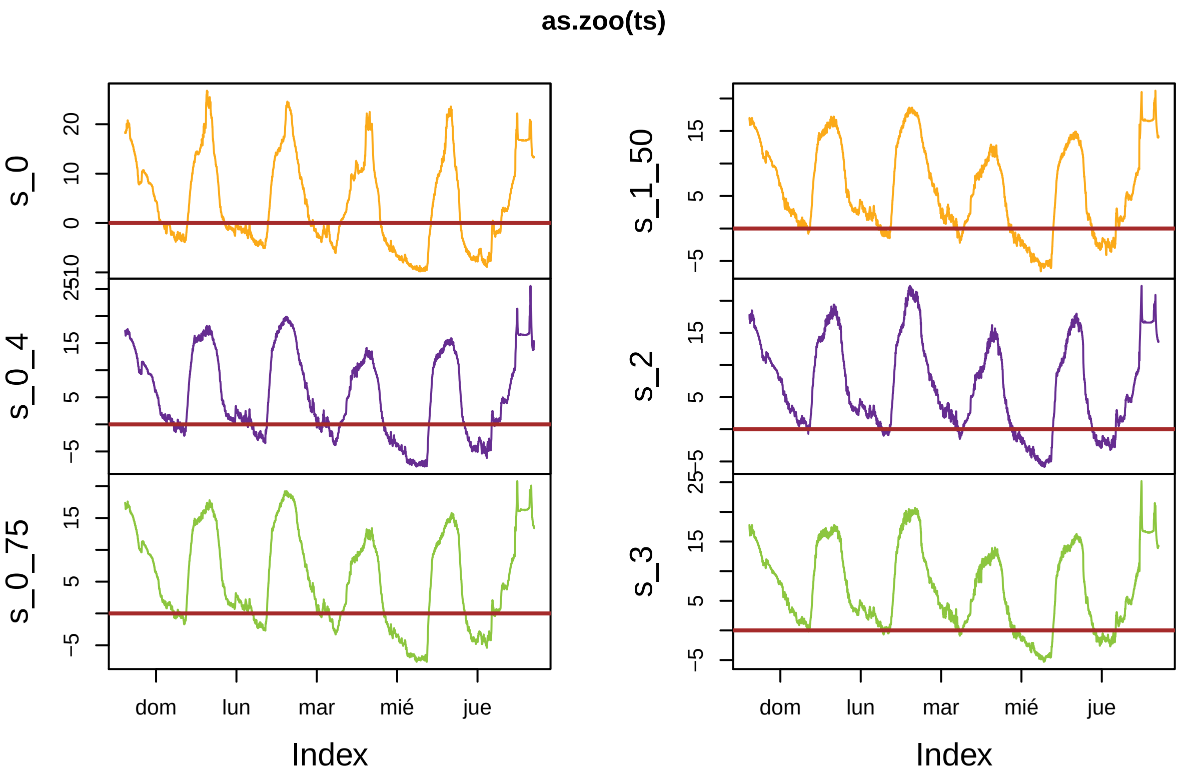
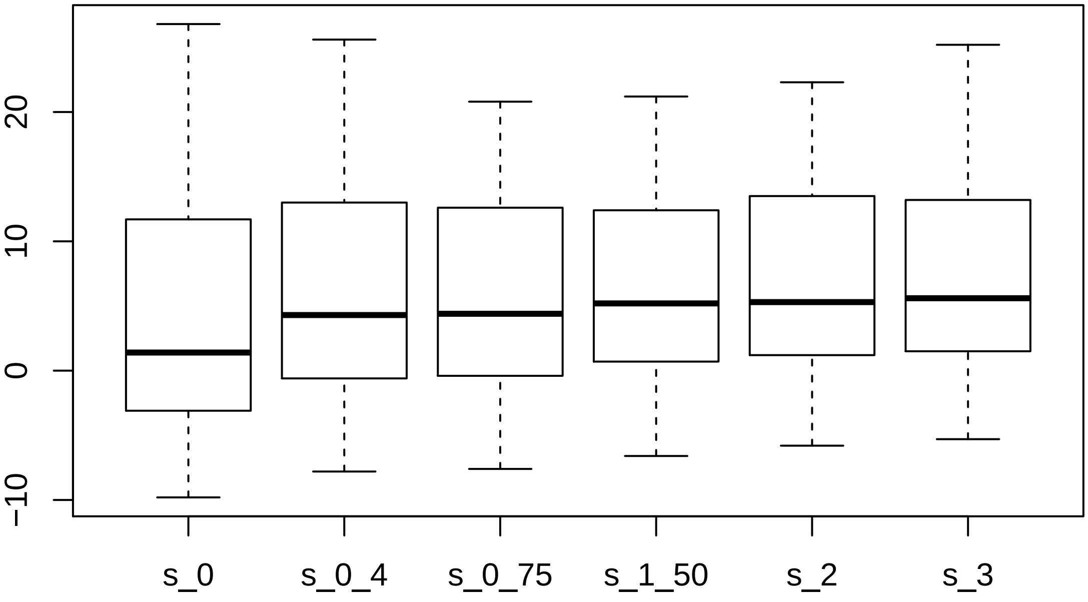


Gráfico de cajas para tener un vistazo de los datos

```
boxplot(data.table[,2:7])
```



metodo para analisis de componentes principales

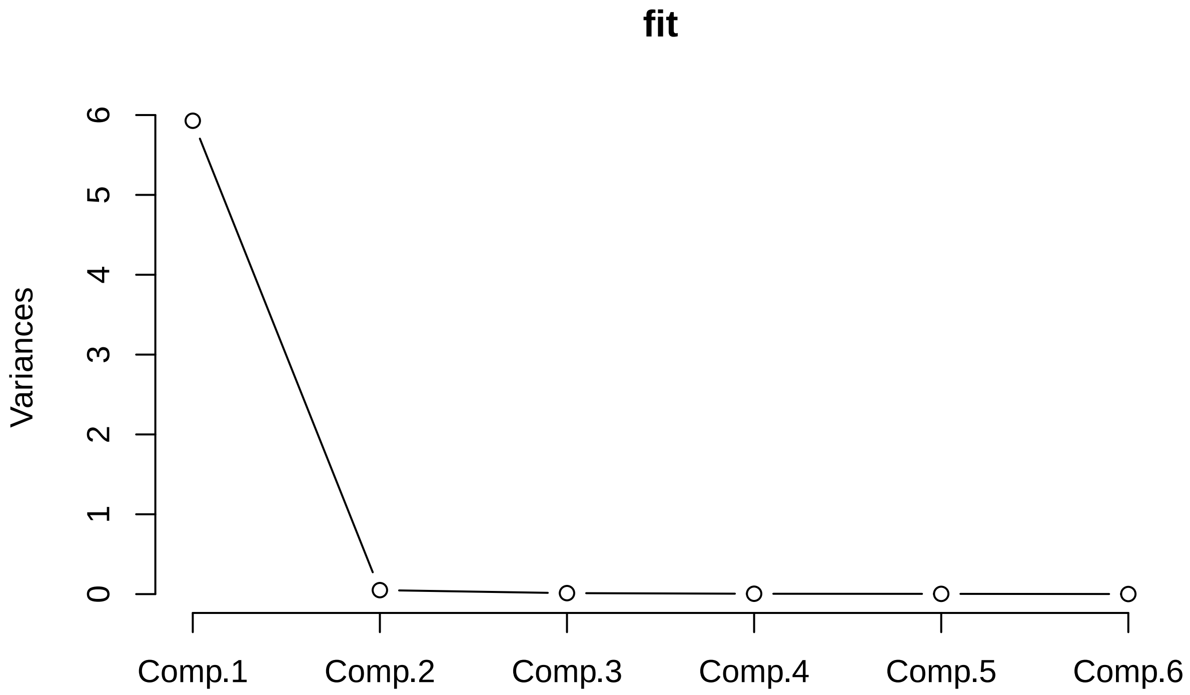
```
fit <- princomp(dataset, cor=TRUE)
# print variance accounted for
summary(fit)
```

```
## Importance of components:
##                           Comp.1      Comp.2      Comp.3      Comp.4
## Standard deviation     2.4348493 0.223474096 0.111403742 0.068724841
## Proportion of Variance 0.9880818 0.008323445 0.002068466 0.000787184
## Cumulative Proportion  0.9880818 0.996405290 0.998473756 0.999260940
##                              Comp.5       Comp.6
## Standard deviation     0.0569590739 0.0344967380
## Proportion of Variance 0.0005407227 0.0001983375
## Cumulative Proportion  0.9998016625 1.0000000000
```

```
# pc loadings
loadings(fit)
```

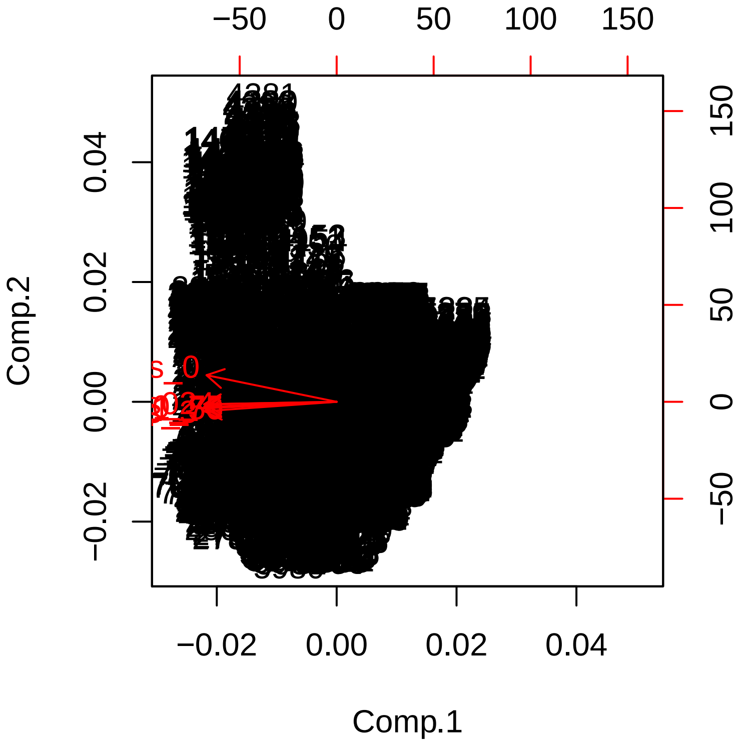
```
##
## Loadings:
##        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
## s_0    -0.402  0.897               -0.129
## s_0_4  -0.409         0.583  0.334  0.390  0.471
## s_0_75 -0.410 -0.175  0.270 -0.326  0.292 -0.733
## s_1_50 -0.410 -0.222  0.201 -0.385 -0.734  0.234
## s_2    -0.409        -0.626 -0.390  0.398  0.351
## s_3    -0.409 -0.317 -0.392  0.690 -0.220 -0.234
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.167  0.167  0.167  0.167  0.167  0.167
## Cumulative Var  0.167  0.333  0.500  0.667  0.833  1.000
```

```
plot(fit,type="lines") # scree plot
```

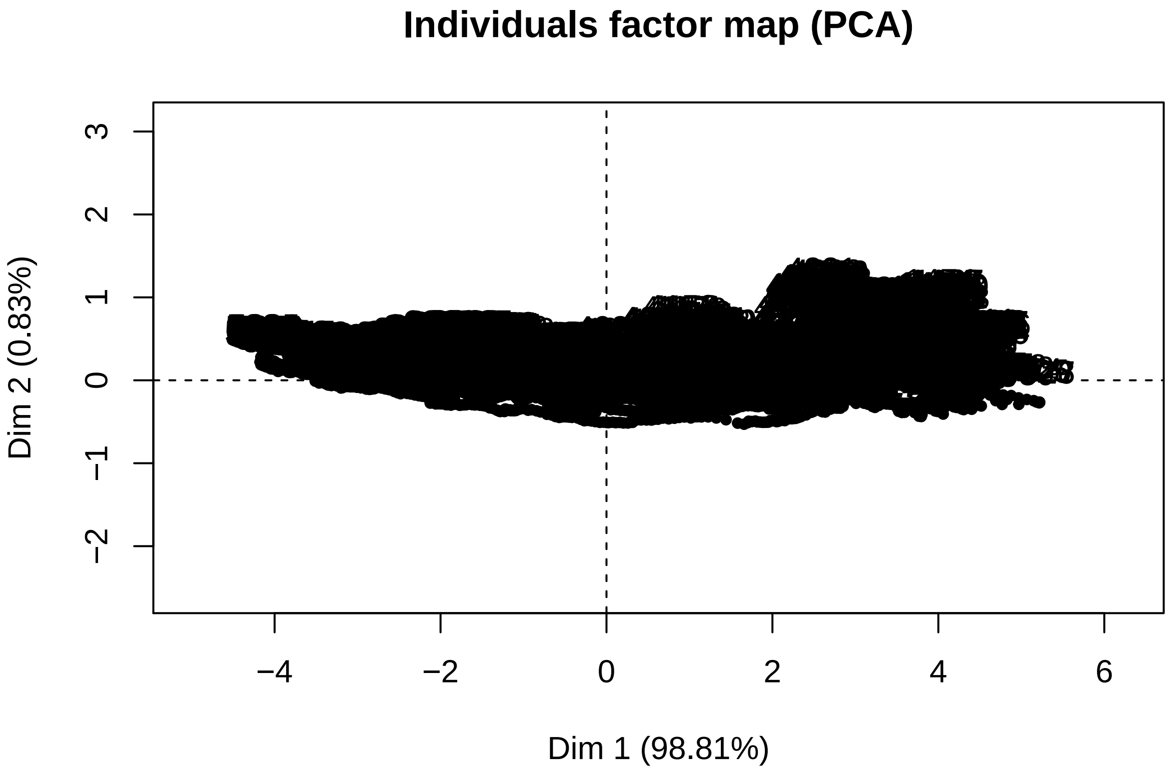


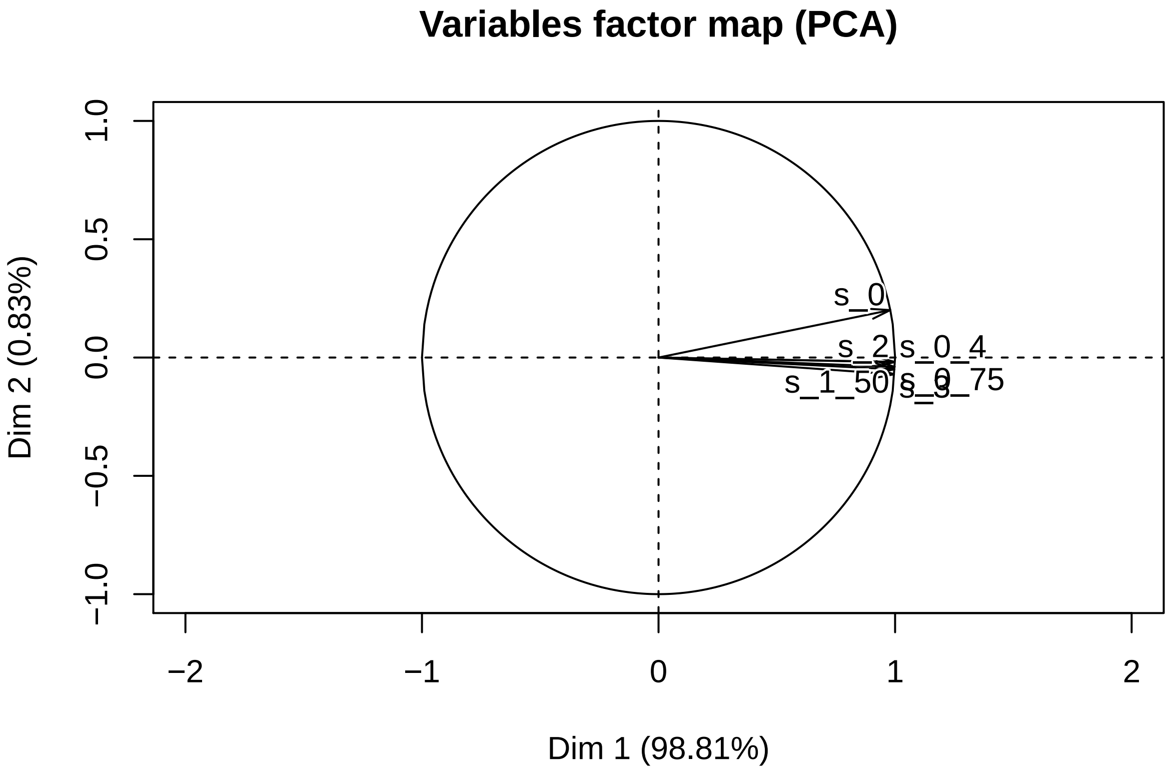
fit$scores # the principal components

```
biplot(fit)
```



```
# PCA Variable Factor Map
result <- PCA(dataset) # graphs generated automatically
```




Variables factor map (PCA)
1.0
0.5
0.0
−0.5
−1.0
Dim 2 (0.83%)
s_0
s_2
s_0_4
s_1_50
s_0_75
s_3
−2
−1
0
1
2
Dim 1 (98.81%)