Noname manuscript No.
(will be inserted by the editor)

An improvement in PCA application

PCA reduction simulation

Received: date / Accepted: date

Abstract This supplementary documentation includes a R code, [R Core Team(2017)] shown the effect of linear relationship between variables in multidimensional data and who the autovalue and autovector can be used to take off these variables.

Keywords Multivariate analysis · Dimensional reduction · PCA · Metabolomics ·

1 PCA reduction simulation

First load required libraries and codes

```
require(MASS)
```

```
## Loading required package: MASS
```

```
require(ggplot2)
```

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

```
require(ggbiplot)
```

```
## Loading required package: ggbiplot
```

```
## Loading required package: plyr
```

Supplementary documentation for article: “An improvement in PCA application” submited to publication on :

```
## Loading required package: scales

## Loading required package: grid
```

```
require(ggforce)
```

```
## Loading required package: ggforce
```

```
require(reshape)
```

```
## Loading required package: reshape

##
## Attaching package: 'reshape'

## The following objects are masked from 'package:plyr':
##
##     rename, round_any
```

Setting initial random state

```
set.seed(54321)
```

Procedure following the example published in stackexchange, [whuber (https://stats.stackexchange.com/users/919/wh
where:
$nVars$ = number of variables
rot = random rotation matrix
$n1$ = number of samples in group 1
$n2$ = number of samples in group 2
eps = Error SD should be small compared to the SDs
x = simulated data
y = rotated simulated data

```
nVars <- 5
rot <- qr.Q(qr(matrix(rnorm(nVars*nVars), nVars)))
sigma <- function(theta=0, lambda=c(1,1)) {
   cos.t <- cos(theta);
   sin.t <- sin(theta)
   a <- matrix(c(cos.t, sin.t, -sin.t, cos.t), ncol=2)
   t(a) %*% diag(lambda) %*% a
}

n1 <- 50
n2 <- 75
x <- rbind(mvrnorm(n1, c(-2,-1), sigma(0, c(1/2,1))),
           mvrnorm(n2, c(0,1), sigma(pi/3, c(1, 1/3))))
eps <- 0.25
x <- cbind(x, matrix(rnorm(dim(x)[1]*(nVars-2), sd=eps),
```

```
                      ncol=nVars-2))
y <- x %*% rot

colnames(y) <- paste(rep("X", 5), 1:5, sep='')
row.names(y) <- c(paste(rep("S", n1), 1:n1, sep=""),
                  paste(rep("R",n2), 1:n2, sep=""))
groups <- as.factor(c(rep("group1", n1), rep("group2",n2)))
```

Plotting the original data. The ellipse corresponde to the standard deviation of multinormal random samples.

```
xdf <- data.frame(x)
ggplot(data=xdf, aes(x=X1, y=X2, color=groups, shape=groups)) +
  geom_point() + theme(legend.direction="horizontal",
                       legend.position="top") +
  geom_ellipse(aes(x0=-2,y0=-1,b=1/2,a=1,angle=0),colour="red") +
  geom_ellipse(aes(x0=0,y0=1,b=1,a=1/3,angle=pi/3),
               colour="cyan") + coord_fixed()
```

Processing the newly simulated data using PCA. The reduced transformed data is plotted to compare with the previous graph.

```
pcaSim <- prcomp(y, center = TRUE, scale. = TRUE)

pcaSim.g <- ggbiplot(pcaSim, obs.scale=1, var.scale=1,
                     groups=groups, ellipse=TRUE, var.axes = FALSE) +
    scale_color_discrete(name="")  +
    theme(legend.direction="horizontal", legend.position="top")
print(pcaSim.g)
```

Additing variables with linear combination of previously created data:
$X6 = 1.0 \times X1 + 2.0 \times X2$
$X7 = -0.5 \times X3 + 0.25 \times X4$
$yLin$ is the previous data set including two new variables, which are a linear combination of other columns.

```
yLin <- y
yLin <- cbind(yLin, 1.0*y[,1] + 2.0*y[,2])
yLin <- cbind(yLin, -0.5*y[,3] + 0.25*y[,4])
colnames(yLin) <- paste(rep("X", 7), 1:7, sep='')
```

Processing, as before, this newly data frame.

```
pcaLin <- prcomp(yLin, center = TRUE, scale. = TRUE)

cat('\nEigenvalues for the initial simulated data :\n')
```

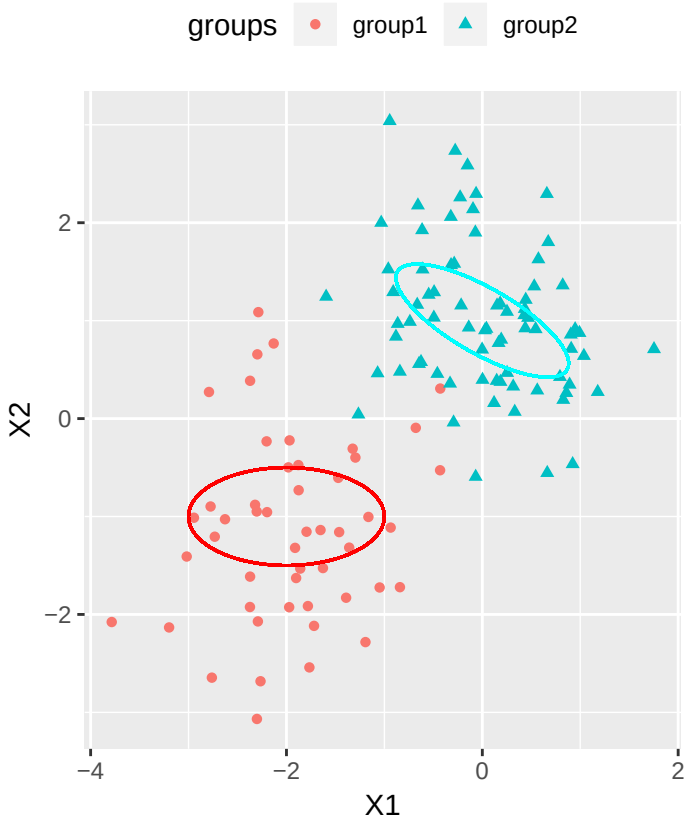


Fig. 1 Original simulated data before rotation. The ellipses shown the scattering of each sample.

```
##
## Eigenvalues for the initial simulated data :
```

```
cat(pcaSim$sdev^2)
```

```
## 3.337933 1.185353 0.2472813 0.1579312 0.07150173
```

```
cat('\nAfter linear combination of some columns:\n')
```

```
##
## After linear combination of some columns:
```

```
cat(pcaLin$sdev^2)
```

```
## 5.000921 1.473507 0.2518177 0.1823668 0.09138739 2.604345e-31 1.388881e-32
```

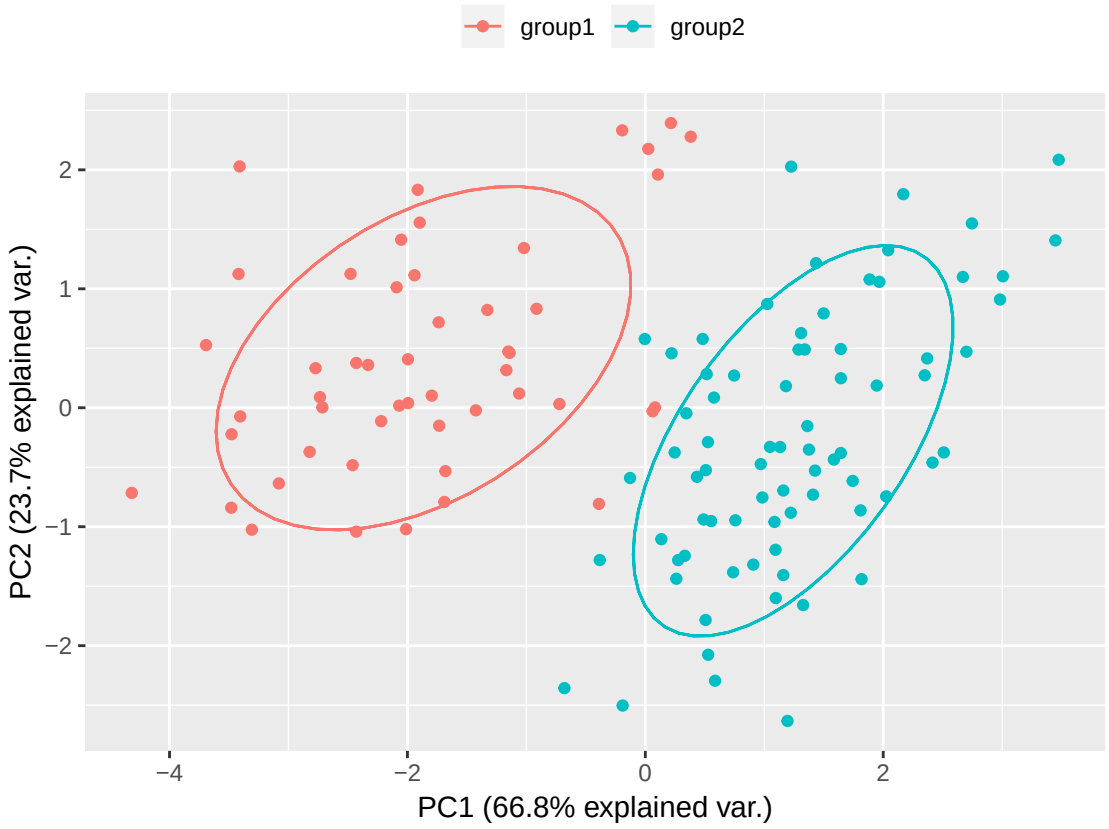


Fig. 2 Biplot of the simulated data y

```
pltdf <- data.frame(PC=c(1:5,1:7),
                    y=c(pcaSim$sdev^2/max(pcaSim$sdev^2),
                        pcaLin$sdev^2/max(pcaLin$sdev^2)),
                    clase=as.factor(c(rep('Original',5),
                                      rep('Linear vars.',7))))
pcaSim.sc <- ggplot(pltdf, aes(PC,y,color=clase)) + geom_line() +
  geom_point() + scale_color_discrete(name="") +
  theme(legend.direction="horizontal",legend.position="top") +
  xlab('principal component number') +
  ylab('proportion of explained variance') +
  scale_shape_manual("", values=c(19,15))
print(pcaSim.sc)
```

Searching by columns with linear combinations as some eigenvalues are quasi-zero. There are two approach: First, follow the book of Jolliffe, [Jolliffe(2002), page 27] and of Härdle and Simar, [Härdle and Simar(2012), page 284].

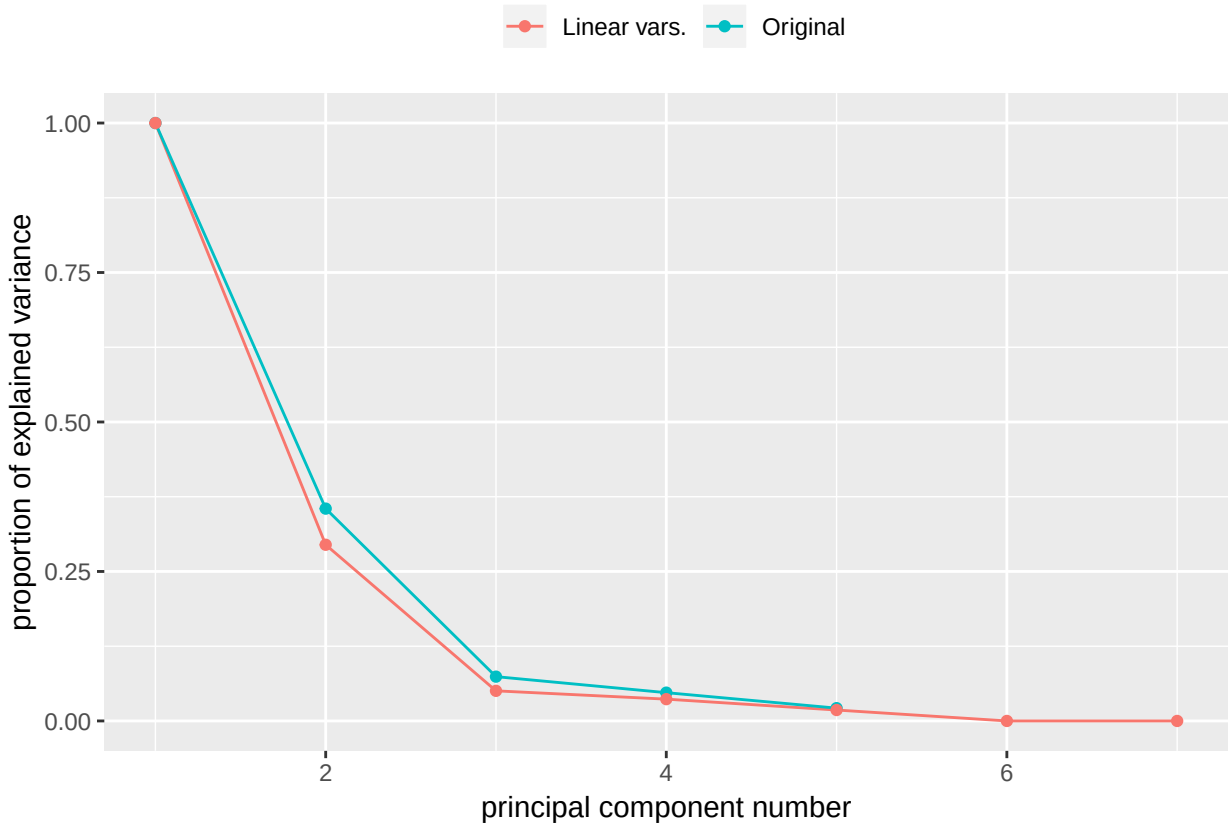


Fig. 3 Biplot of the simulated data including two columns with linear relationship.

There are two small eigenvalues (6th and 7th). Calculing the values of the eigenvectors divided by the standard deviation and the correlation between variable X_i and the normalized principal component (NPC) Z_j, as, the options "center" and "scale." are being used:

$$r_{X_i Z_j} = \sqrt{l_j} g_{R,ij},$$

where, g_{ij} are the componentes of the jth eigenvector, l_j is the jth eigenvalue.

```
cat('The coefficients in the corresponding PC are:\n')
```

```
## The coefficients in the corresponding PC are:
```

```
round(pcaLin$rotation[,6:7]/pcaLin$scale, digits = 2)
```

```
##      PC6   PC7
## X1 -0.24 -0.04
```

```
## X2 -0.48 -0.07
## X3 -0.17  1.18
## X4  0.09 -0.59
## X5  0.00  0.00
## X6  0.24  0.04
## X7 -0.35  2.35
```

```
cat('The variable with the highest coefficient, in absolute value,
    is "X7"; as previously established X7 <- -0.5 * X3 + 0.25 * X4\n')
```

```
## The variable with the highest coefficient, in absolute value,
##     is "X7"; as previously established X7 <- -0.5 * X3 + 0.25 * X4
```

```
cat('From Härle and Simar book, calculating the correlation
    between variables and the respective PC :\n')
```

```
## From Härle and Simar book, calculating the correlation
##     between variables and the respective PC :
```

```
CorYpY <- pcaLin$rotation[,6:7]*
  matrix(pcaLin$sdev[6:7], nrow(pcaLin$rotation), 2, byrow=TRUE)
round(apply(CorYpY, 2, function(x){return (x/max(abs(x)))}), digits=3)
```

```
##       PC6    PC7
## X1 -0.213 -0.031
## X2 -0.805 -0.117
## X3 -0.104  0.697
## X4  0.061 -0.408
## X5  0.000  0.000
## X6  1.000  0.145
## X7 -0.148  1.000
```

Removing the identified columns with highest coefficiente and recalculing the PCA.

```
pcaLin <- prcomp(yLin[,-7], center = TRUE, scale. = TRUE)
cat('\nThe new eigenvalues are :\n')
```

```
##
## The new eigenvalues are :
```

```
cat(pcaLin$sdev^2)
```

```
## 4.301644 1.195469 0.2518124 0.1617787 0.08929632 2.552429e-31
```

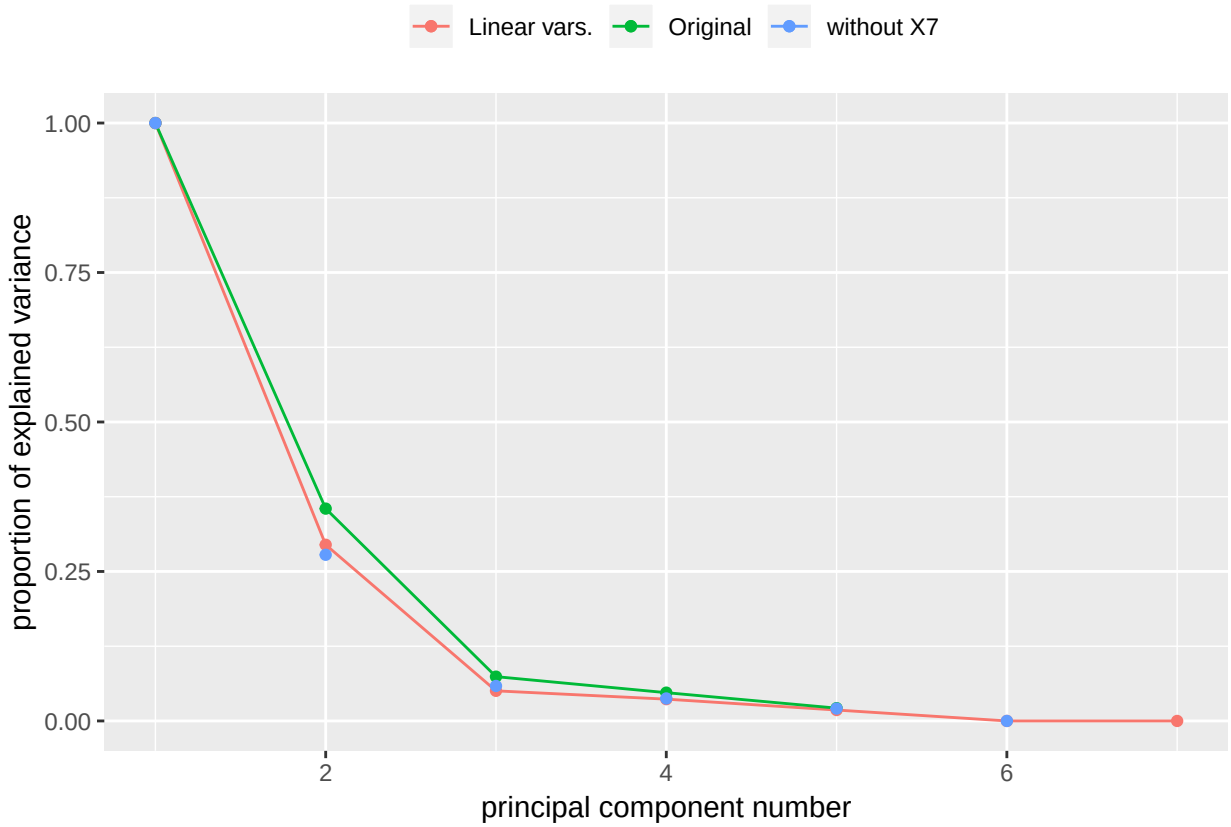


Fig. 4 Biplot of PCA without the variable X7.

```
pcaSim.sc <- pcaSim.sc + geom_point(data =
data.frame(x2=1:6,y2=pcaLin$sdev^2/max(pcaLin$sdev^2),
           clase=rep('without X7',6)),
aes(x=x2,y=y2,colour=clase))
print(pcaSim.sc)
```

```
cat('\nThe coefficients for the 6th PC are now:\n')
```

```
##
## The coefficients for the 6th PC are now:
```

```
round(pcaLin$rotation[,6]/pcaLin$scale, digits = 2)
```

```
##    X1    X2    X3    X4    X5    X6
##  0.24  0.49  0.00  0.00  0.00 -0.24
```

In this case, the linear relationship can be clearly observed. Remembering that $X6 = 1.0 \times X1 + 2.0 \times X2$ or $0 = -X6 + X1 + 2 \times X2$, which can be write as

$$1 \times X1 + 2 \times X2 + 0 \times X3 + 0 \times X4 + 0 \times X5 + (-1) \times X6 = 0$$

equal to the coefficients of the 6th PC, dividided by 0.24.

```
cat('The Correlation of the variables to this PC is\n')
```

```
## The Correlation of the variables to this PC is
```

```
CorYpY2 <- pcaLin$rotation[,6]*pcaLin$sdev[6]
round(CorYpY2/max(abs(CorYpY2)), digits=3)
```

```
##     X1     X2     X3     X4     X5     X6
##  0.213  0.805  0.000  0.000  0.000 -1.000
```

```
cat('Eleminating the 2th variable, variable with the greatest
    coefficient, and recalculating the PCA:')
```

```
## Eleminating the 2th variable, variable with the greatest
##     coefficient, and recalculating the PCA:
```

```
pcaLin <- prcomp(yLin[,-c(2,7)], center = TRUE, scale. = TRUE)
cat('\nThe new eigenvalues are now:\n')
```

```
##
## The new eigenvalues are now:
```

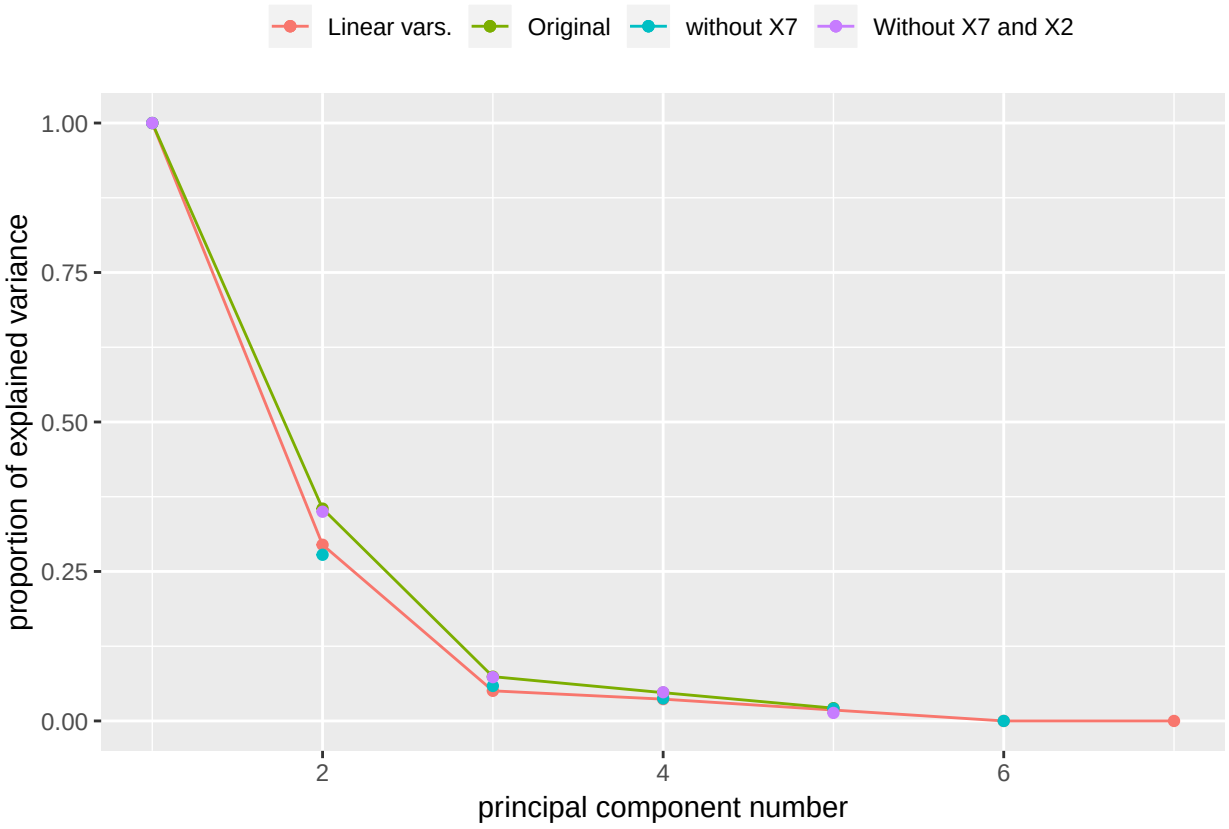
```
cat(pcaLin$sdev^2)
```

```
## 3.367867 1.178511 0.2466689 0.1617779 0.0451745
```

```
cat('\nNow the lowest eigenvalue is 5% of the highest, that is,
    there is no linear relationship between the remaining variables.')
```

```
##
## Now the lowest eigenvalue is 5% of the highest, that is,
##     there is no linear relationship between the remaining variables.
```

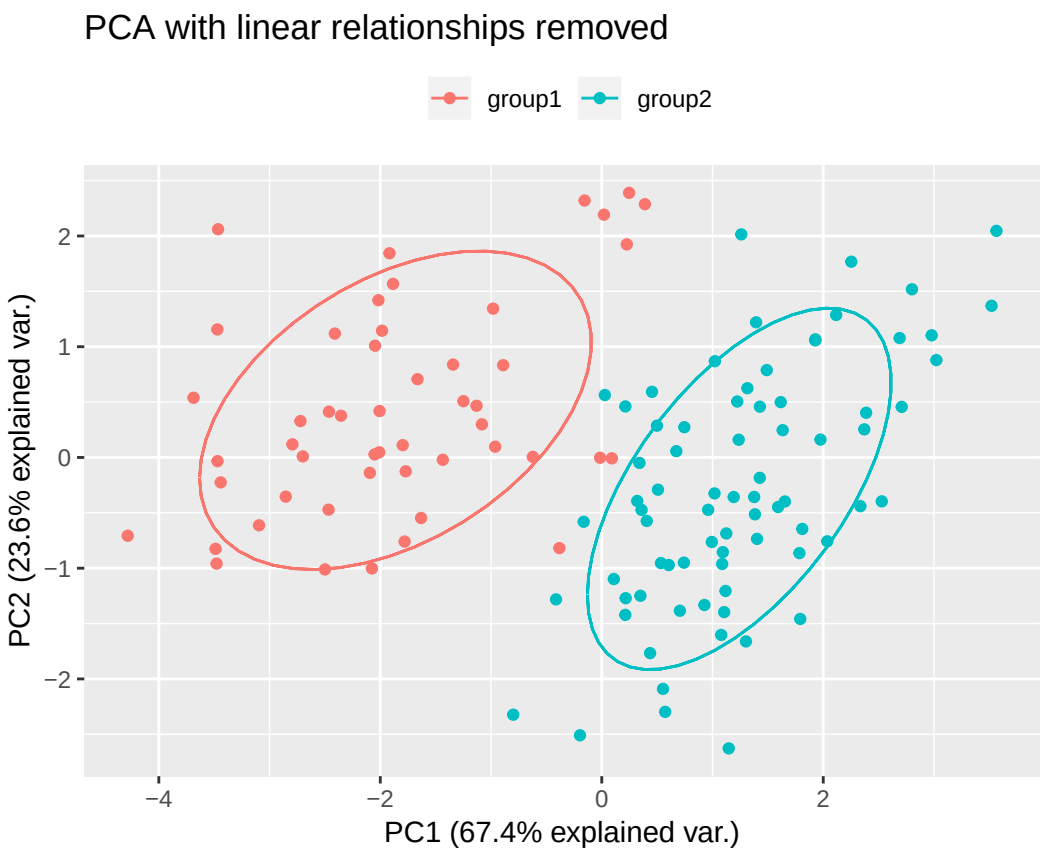
```
pcaSim.sc <- pcaSim.sc + geom_point(data =
data.frame(x2=1:5,y2=pcaLin$sdev^2/max(pcaLin$sdev^2),
           clase=rep('Without X7 and X2',5)),
aes(x=x2,y=y2, colour=clase))
print(pcaSim.sc)
```



Note that in the previous screeplot, the initial and the final eigenvalues coincide, despite the fact that a variable $X2$ was removed instead of $X6$.

Ploting the final PCA.

```
pcaSim.final.g <- ggbiplot(pcaLin, obs.scale=1, var.scale=1,
                           groups=groups,
                           ellipse=TRUE, var.axes = FALSE) +
                  scale_color_discrete(name="")  +
                  theme(legend.direction="horizontal",
                        legend.position="top")+
                  ggtitle('PCA with linear relationships removed')
print(pcaSim.final.g)
```



2 References

References

Härdle and Simar(2012). Härdle WK, Simar L (2012) Applied Multivariate Statistical Analysis. Springer, URL https://www.amazon.com/Applied-Multivariate-Statistical-Analysis-Wolfgang/dp/3642172288?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=3642172288

whuber (https://stats.stackexchange.com/users/919/whuber)(????). whuber (https://statsstackexchangecom/users/919/whuber) (????) Construct artificial slightly overlapping data for pca plot. Cross Validated, URL https://stats.stackexchange.com/q/35035, uRL:https://stats.stackexchange.com/q/35035 (version: 2012-08-24), https://stats.stackexchange.com/q/35035

Jolliffe(2002). Jolliffe I (2002) Principal component analysis. Springer Verlag, New York

R Core Team(2017). R Core Team (2017) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/