R package `plsdepot`
SIMPLS Canonical Analysis

Gaston Sanchez
www.gastonsanchez.com/plsdepot

1 Introduction

SIMPLS is a technique proposed by Sijmen de Jong (1993) as an alternative algorithm for PLS regression. In turn, SIMPLS Canonical Analysis (SIMPLS-CA) is a sister method of SIMPLS in which the blocks X and Y play a symmetric role. In other words, X are no longer predictors, and Y are no longer responses. Instead, they both can be considered as blocks of descriptors. The main idea behind SIMPLS-CA is to look for components $t_h = Xa_h$ and $u_h = Yb_h$ by maximizing the following expression:

$$cov(Xa_h, Yb_h)$$

under the following conditions

- normalized coefficients: $\|a_h\| = 1$ and $\|b_h\| = 1$
- orthogonal components: $t'_h(t_1, \dots, t_{h-1}) = 0$
- orthogonal components: $u'_h(u_1, \dots, u_{h-1}) = 0$

2 Data `linnerud`

For this demo we are going to use the data set `linnerud` that already comes in `plsdepot`. This data contains 6 variables measured on 20 individuals. The variables can be grouped in two blocks. One block X for three physical measurements, and another block Y for exercise outputs.

```
# load the package
library(plsdepot)

# load the data
data(linnerud)

# let's take a peek
head(linnerud)

##   Weight Waist Pulse Pulls Squats Jumps
## 1    191    36    50     5    162    60
## 2    189    37    52     2    110    60
```

```
## 3    193     38     58     12     101    101
## 4    162     35     62     12     105     37
## 5    189     35     46     13     155     58
## 6    182     36     56      4     101     42
```

3 Function `simplsca()`

`plsdepot` comes with the function `simplsca()` that performs SIMPLS-CA. This function has 3 arguments: `X`, `Y`, and `scaled`. `X`, as you may guess, is the data containing the predictors. This can be either a matrix or a data frame. `Y` is the data containg the responses, which can also be either a matrix or a data frame. `scaled` specifies whether to standardize the data (`TRUE` by default). Let's apply `simplsca()` on `linnerud`.

```
# apply simplsca
my_simca = simplsca(linnerud[, 1:3], linnerud[, 4:6])

# what's in my_simca?
my_simca

##
## SIMPLS Canonical Analysis
## ----------------------------------------
## $x.scores    X-scores (T-components)
## $x.wgs       X-weights
## $y.scores    Y-scores (U-components)
## $y.wgs       Y-weights
## $cor.xt      X,T correlations
## $cor.yu      Y,U correlations
## $cor.xu      X,U correlations
## $cor.yt      Y,T correlations
## $cor.tu      T,U correlations
## $R2X         explained variance of X by T
## $R2Y         explained variance of Y by T
## ----------------------------------------
##
```

What you get in `my_simca` is an object of class `"simplsca"`, and everytime you print an object of such class you get a display with the list of results.

3.1 T components

The first two elements in the list are `$x.scores` and `$x.wgs` which contains the extracted PLS components, and its associated weights (i.e. coefficients).

```
# check scores T
head(round(my_simca$x.scores, 3))
```

```
##       t1     t2
## 1 -0.643 -0.222
## 2 -0.770  0.053
## 3 -0.907  0.150
## 4  0.688  0.387
## 5 -0.487 -0.379
## 6 -0.229  0.033
```

```
# T-weights
my_simca$x.wgs
```

```
##             t1        t2
## Weight -0.5899 -0.714095
## Waist  -0.7713  0.700020
## Pulse   0.2389 -0.006399
```

Similarly, elements three and four give the scores `$y.scores` and its associated weights `$y.wgs`

```
# U components
head(round(my_simca$y.scores, 3))
```

```
##       u1     u2
## 1 -0.371  0.102
## 2 -1.340  0.579
## 3 -0.082  0.520
## 4 -0.355 -0.574
## 5  0.463 -0.530
## 6 -1.306  0.160
```

```
# U-weights
my_simca$y.wgs
```

```
##            u1      u2
## Pulls  0.6133 -0.4170
## Squats 0.7470 -0.2888
## Jumps  0.2567  0.8618
```

3.2 Correlations between variables and components

In order to check how the T and U components are associated with their own group of variables, we use `$cor.xt` and `$cor.yu`.

```
# correlations between X and T
my_simca$cor.xt
```

```
##             t1       t2
## Weight -0.9476 -0.28471
```

```
## Waist  -0.9620  0.22440
## Pulse   0.5108  0.02154

# correlations between Y and U
my_simca$cor.yu

##            u1        u2
## Pulls  0.8802 -0.275720
## Squats 0.9397 -0.003128
## Jumps  0.7407  0.667892
```

The first component t_1 is capturing well enough the information of `Weight` and `Waist`. Likewise, the correlations with u_1 indicate that it summarizes the information of the Y block.

3.3 Cross-Correlations

We aso have the cross-correlations between the extracted components and the variables of the other blocks.

```
# correlations between X and T
my_simca$cor.xt

##             t1       t2
## Weight -0.9476 -0.28471
## Waist  -0.9620  0.22440
## Pulse   0.5108  0.02154

# correlations between Y and U
my_simca$cor.yu

##            u1        u2
## Pulls  0.8802 -0.275720
## Squats 0.9397 -0.003128
## Jumps  0.7407  0.667892

# correlations between T and U
my_simca$cor.tu

##            t1         t2         u1         u2
## t1  1.000e+00  6.222e-18  5.536e-01 -2.869e-01
## t2  6.222e-18  1.000e+00 -2.630e-01  3.948e-01
## u1  5.536e-01 -2.630e-01  1.000e+00  1.164e-16
## u2 -2.869e-01  3.948e-01  1.164e-16  1.000e+00
```

3.4 Explained Variance

Besides the correlation among the data blocks and the extracted components, ther last results are the proportion of explained variance. They allow us to to assess how well the components explain the variability in their wach block of variables.

```
# explained variance of X by T
my_simca$R2XT

##            t1     t2
## Weight 0.8980 0.9791
## Waist  0.9255 0.9758
## Pulse  0.2609 0.2613

# explained variance of Y by T
my_simca$R2YT

##            t1      t2
## Pulls  0.2363 0.32833
## Squats 0.3506 0.42960
## Jumps  0.0414 0.04715

# explained variance of Y by U
my_simca$R2YU

##            u1     u2
## Pulls  0.7747 0.8507
## Squats 0.8830 0.8830
## Jumps  0.5487 0.9948

# explained variance of X by U
my_simca$R2XU

##             u1      u2
## Weight 0.21597 0.24121
## Waist  0.36926 0.50174
## Pulse  0.03542 0.05538
```

4 Plotting "simplsca" objects

An accessory function is the `plot()` method that allows us to get some graphics of the basic results. Basically, we can plot either the variables and the observations on the specified components. The variables are plotted inside a circle of correlations. In turn, the observations are plotted using a scatter-plot.

4.1 Plotting variables

The default output when using `plot()` is a graphic showing the correlations of the variables with the first two components. This plot can be regarded as a radar. The closer a variable appears on the perimiter of the circle, the better it is represented. In addition, if two variables are highly correlated, they will appear near each other. If two variables are negatively correlated, they will tend to appear in opposite extremes. If two variables are uncorrelated, they will be orothogonal to each other.

```r
# default plot
plot(my_simca)
```

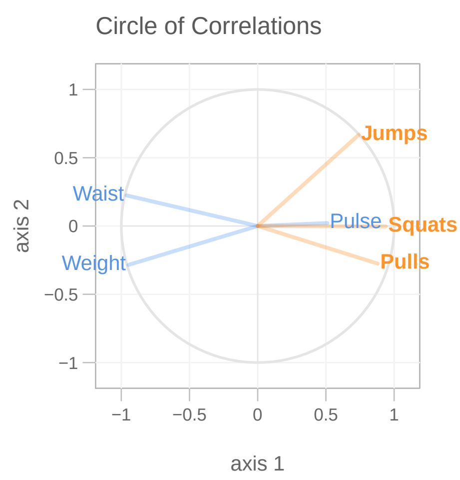

Figure 1: Circle of correlations (axes 1-2)

4.2 Plotting observations

The alternative output when using `plot()` is to show a scatter-plot of the observations on the specified components.

```r
# default plot
plot(my_simca, what = "observations", show.names = TRUE)
```

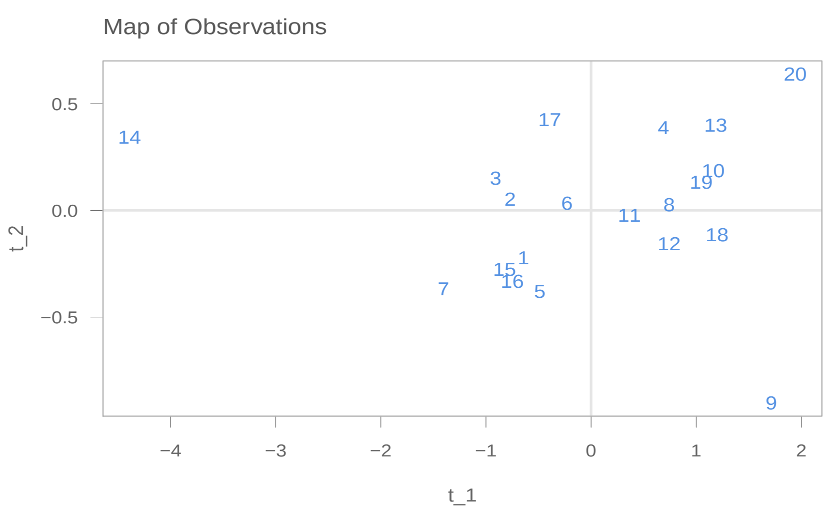

Figure 2: Plot of observations (comps 1-2)

References

de Jong S. (1993) SIMPLS: An Alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18: 251–263.

Tenenhaus M. (1998) *La Regression PLS. Theorie et Pratique*. Paris: Editions TECHNIP.