R package `plsdepot`
Tucker's Inter-Battery Factor Analysis

Gaston Sanchez
www.gastonsanchez.com/plsdepot

1 Tucker's Inter-Battery

Tucker's Inter-Battery Factor Analysis is a method for analyzing two data tables (i.e. two blocks of variables) usually denoted by X and Y. The basic idea behind the method is to look for components $t_h = Xa_h$ and $u_h = Yb_h$ in such a way that:

1. they explain well their own block of variables, and
2. they are as much correlated as possible with each other.

In other words, t_h will be a good representant of X, and u_h will be a good representant of Y. At the same time, t_h will be as much correlated as possible with u_h.

So, how do we find such components t_h and u_h?. The way to find the components is to simultaneously maximize $var(Xa_h)$, $var(Yb_h)$ and $cor(Xa_h, Yb_h)$, which can be done by maximizing the following expression

$$cov(Xa_h, Yb_h) = cor(Xa_h, Yb_h)\sqrt{var(Xa_h)}\sqrt{var(Yb_h)}$$

In his book *La Regression PLS*, Michel Tenenhaus gives us the best interpretation of Tucker's Inter-Battery method: you can think of it as a comprise between a canonical correlation analysis of blocks X and Y, and principal components analysis of each block. In `plsdepot` you can find the function `interbat()` for performing Tucker's Inter-Battery Method.

2 Data `linnerud`

For this demo we are going to use the dataset `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 (`Weight`, `Wasit`, and `Pulse`), and another block Y for exercise outputs (`Pulls`, `Squats`, and `Jumps`).

```
# load the package
library(plsdepot)

# load the data
data(linnerud)
```

```r
# 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
```

2.1 Exploratory Analysis

Our initial exploratory analysis begins by getting some summary statistics of the variables

```r
summary(linnerud)
```

```
##      Weight        Waist          Pulse          Pulls           Squats
##  Min.   :138   Min.   :31.0   Min.   :46.0   Min.   : 1.00   Min.   : 50
##  1st Qu.:161   1st Qu.:33.0   1st Qu.:51.5   1st Qu.: 4.75   1st Qu.:101
##  Median :176   Median :35.0   Median :55.0   Median :11.50   Median :122
##  Mean   :179   Mean   :35.4   Mean   :56.1   Mean   : 9.45   Mean   :146
##  3rd Qu.:192   3rd Qu.:37.0   3rd Qu.:60.5   3rd Qu.:13.25   3rd Qu.:210
##  Max.   :247   Max.   :46.0   Max.   :74.0   Max.   :17.00   Max.   :251
##      Jumps
##  Min.   : 25.0
##  1st Qu.: 39.5
##  Median : 54.0
##  Mean   : 70.3
##  3rd Qu.: 85.2
##  Max.   :250.0
```

We can also calculate the correlations among the quantitative variables. In this case there are only six variables which makes it relatively easy to visually inspect the correlations directly from the matrix of correlations.

```r
# matrix of correlations
cor(linnerud)
```

```
##         Weight   Waist    Pulse   Pulls  Squats    Jumps
## Weight  1.0000  0.8702 -0.36576 -0.3897 -0.4931 -0.22630
## Waist   0.8702  1.0000 -0.35289 -0.5522 -0.6456 -0.19150
## Pulse  -0.3658 -0.3529  1.00000  0.1506  0.2250  0.03493
## Pulls  -0.3897 -0.5522  0.15065  1.0000  0.6957  0.49576
## Squats -0.4931 -0.6456  0.22504  0.6957  1.0000  0.66921
## Jumps  -0.2263 -0.1915  0.03493  0.4958  0.6692  1.00000
```

3 Function `interbat()`

To have a better and deeper understanding of the relationships between the physical measurements and the exercise variables, we can apply Tucker's Inter-Battery method with the function `interbat()`. This function has 3 arguments: `Data`, `comps`, and `scaled`. `Data`, as you may guess, is the data to be analyzed. This can be either a matrix or a data frame. `comps` is the number of components to be calculated. `scaled` specifies whether to standardize the data (`TRUE` by default).

```
# apply interbat
my_inter = interbat(linnerud[, 1:3], linnerud[, 4:6])

# what's in my_inter?
my_inter

##
## Tucker's Inter-Battery Method
## ----------------------------------------------
## $values       eigenvalues
## $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 U
## $com.xu       communality of X with U
## $com.yt       communality of Y with T
## $statistic    statistic Phi
## ----------------------------------------------
##
```

3.1 Eigenvalues

The first element in the list of results are the eigenvalues contained in `$values`

```
# check eigenvalues
my_inter$values

## [1] 1.272426 0.005657 0.001106
```

3.2 Components

The second element in the list of results is `$x.scores` (i.e. the t_h components) which contains the X components. We can inspect the scores of the first observations like this:

```r
# check scores th
head(round(my_inter$x.scores, 3))

##       t1     t2     t3
## 1 -0.643 -0.075 -0.764
## 2 -0.770 -0.155 -0.366
## 3 -0.907  0.201  0.453
## 4  0.688 -0.097  0.809
## 5 -0.487 -0.244 -1.363
## 6 -0.229  0.015  0.039
```

Likewise, we can check the Y scores (i.e. the u_h components)

```r
# check scores uh
head(round(my_inter$y.scores, 3))

##       u1     u2     u3
## 1 -0.371 -0.054 -0.823
## 2 -1.340  0.196 -0.717
## 3 -0.082  0.585  0.866
## 4 -0.355 -0.629  0.744
## 5  0.463 -0.399  0.397
## 6 -1.306 -0.201 -0.359
```

3.3 Factor Structure

In order to start examining the factor structure of the blocks, we inspect the correlations of the physical measurements X and their components t_h.

```r
# correlations X-T
my_inter$cor.xt

##             t1     t2       t3
## Weight -0.9476 0.4049 -0.19478
## Waist  -0.9620 0.1169 -0.07474
## Pulse   0.5108 0.6088  0.95034
```

We also inspect the correlations of the exercise measurements Y and their components u_h.

```r
# correlations Y-U
my_inter$cor.yu

##            u1     u2       u3
## Pulls  0.8802 0.1946  0.61617
## Squats 0.9397 0.4257 -0.13368
## Jumps  0.7407 0.9420  0.01581
```

The next step for checking the factor structure is to inspect the cross-correlations of the physical measurements X with the exercise components u_h, as well as the exercise measurements Y with the physical components t_h.

```
# correlations X-U
my_inter$cor.xu

##             u1       u2       u3
## Weight -0.4647 -0.07254  0.01414
## Waist  -0.6077  0.04249 -0.02679
## Pulse   0.1882 -0.04195 -0.05158

# correlations Y-T
my_inter$cor.yt

##            t1       t2        t3
## Pulls  0.4862  0.03028 -0.030384
## Squats 0.5921  0.02202  0.025832
## Jumps  0.2035 -0.13643 -0.002575
```

3.4 Explained Variance

Besides the correlation among the data blocks and the extracted components, another important result is the proportion of explained variance. Here the purpose is to assess how well each group of components explain their own block.

```
# explained variance of X by T
my_inter$R2X

##            t1     t2     t3
## Weight 0.8980 1.0620 1.0999
## Waist  0.9255 0.9391 0.9447
## Pulse  0.2609 0.6316 1.5347
```

```
# explained variance of Y by U
my_inter$R2Y

##            u1     u2    u3
## Pulls  0.7747 0.8125 1.192
## Squats 0.8830 1.0642 1.082
## Jumps  0.5487 1.4361 1.436
```

3.5 Inter-group Communalities

Because the components t_h and u_h are extracted in such a way that they are as much correlated as possible, it is also interesting to check how well each group of components may explain the other block.

```
# communality of X with U
my_inter$com.xu

##             u1      u2      u3
## Weight 0.21597 0.22123 0.22143
## Waist  0.36926 0.37107 0.37178
## Pulse  0.03542 0.03718 0.03984
```

```
# communality of Y with T
my_inter$com.yt

##            t1      t2      t3
## Pulls  0.2363 0.23726 0.23819
## Squats 0.3506 0.35108 0.35175
## Jumps  0.0414 0.06001 0.06002
```

3.6 Statistic Φ

The last result is the Φ statistic that is supposed to provide information about the minimum number m of components to be kept. The Φ statistic for each component follows a *chi-square* distribution with $(p-m)(q-m)$ degrees of freedom (p number of variables in X, q number of variables in Y).

```
# phi statistic
my_inter$statistic

##       phi df  p.value
## 1 25.5838  9 0.002389
## 2  0.5842  4 0.964800
## 3  0.1034  1 0.747824
```

4 Plotting "interbat" 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 principal axes (associated to 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_inter)
```

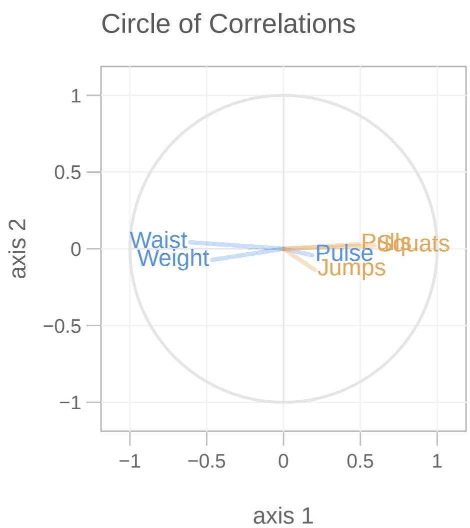

Figure 1: Circle of correlations (axes 1-2)

```r
# another plot using components 1 and 3
plot(my_inter, comps = c(1, 3))
```

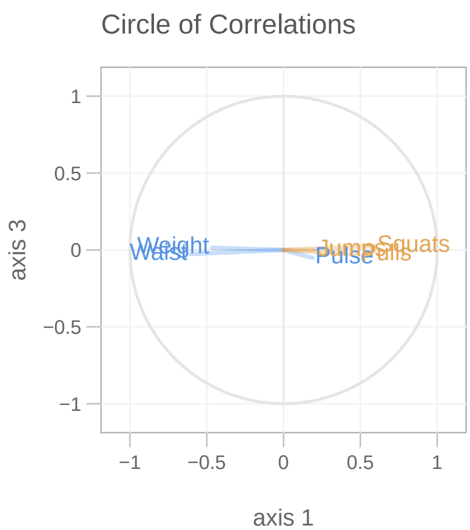

Figure 2: Circle of correlations (axes 1-3)

4.2 Plotting observations

The alternative output when using `plot()` is to show a scatter-plot of the observations on the specified components. This is done by specifying the argument `what="observations"`. Moreover, if we wish to show the labels of the observations, we have to set the paramter `show.names=TRUE`

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

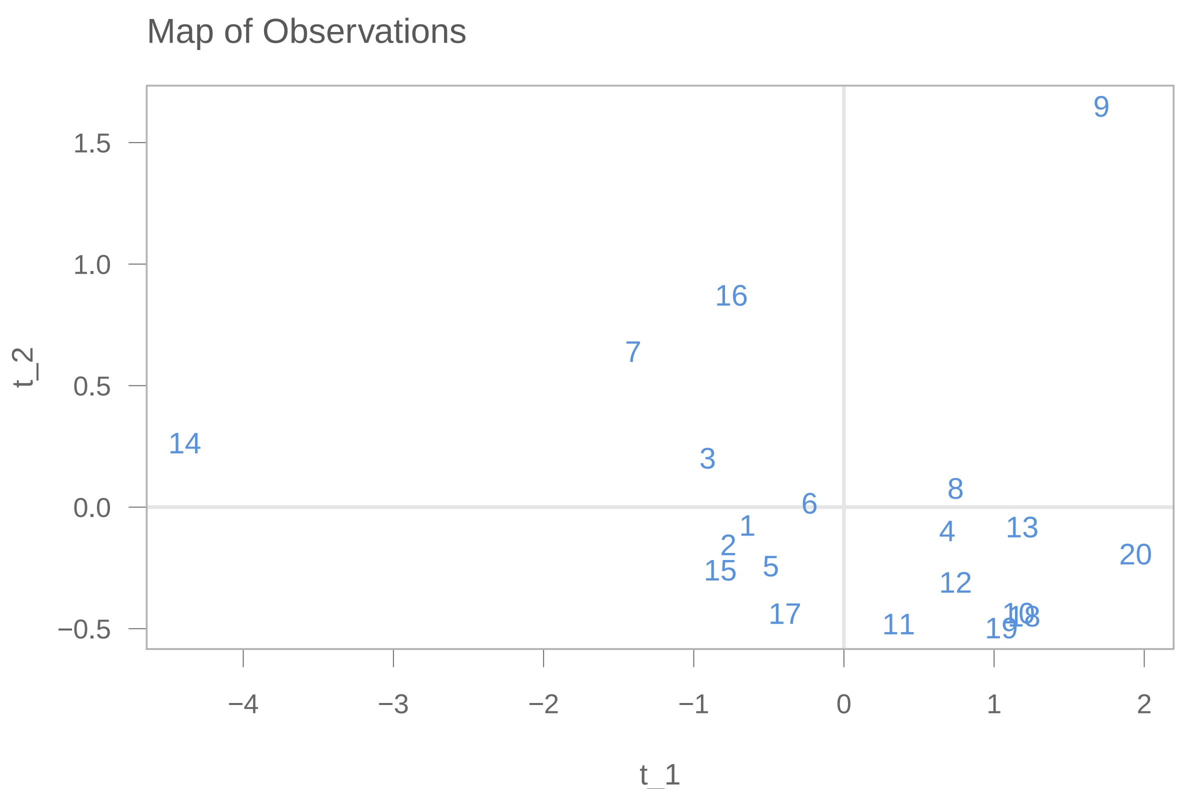

Figure 3: Plot of observations (comps 1-2)