R package `plsdepot`
PLS Regression 2

Gaston Sanchez
www.gastonsanchez.com/plsdepot

1 PLS-R2

PLS Regression is the method that most people think of when hearing the acronym **PLS**. Briefly, PLS Regression is just an algorithm for regression analysis in which we want to analyze one block of response variables Y in terms of another block of predictor variables X. When we have more than one response variable, we talk about PLS-R2, the PLS version of the multivariate regression.

1.1 Multivariate Regression model

In a multivariate regression analysis, we model the behavior of a set of responses Y in terms of a set of quantitative predictors X by using a linear model like this one

$$E(Y) = XB$$

The goal is to find the matrix of coefficients B that help us to combine our predictors so we can have a solution as "close" as possible to the expected value of Y.

2 Data `vehicles`

In this demo we are going to use the `vehicles` data set that comes in `plsdepot`. This data consists of 16 variables measured on 30 cars. The original source for this data can be found at http://archive.ics.uci.edu/ml/datasets/Automobile

```
# load the package
library(plsdepot)

# load the data
data(vehicles)

# let's take a peek
head(vehicles)

##           diesel turbo two.doors hatchback wheel.base length width height
## alfaromeo      0     0         1         1       94.5  171.2  65.5   52.4
## audi           0     0         0         0      105.8  192.7  71.4   55.7
## bmw            0     0         1         0      101.2  176.8  64.8   54.3
## chevrolet      0     0         0         0       94.5  158.8  63.6   52.0
```

```
## dodge1          0     1         1         1       93.7  157.3  63.8   50.8
## dodge2          0     0         0         1       93.7  157.3  63.8   50.6
##           curb.weight eng.size horsepower peak.rpm price symbol city.mpg
## alfaromeo        2823      152        154     5000 16500      1       19
## audi             2844      136        110     5500 17710      1       19
## bmw              2395      108        101     5800 16430      2       23
## chevrolet        1909       90         70     5400  6575      0       38
## dodge1           2128       98        102     5500  7957      1       24
## dodge2           1967       90         68     5500  6229      1       31
##           highway.mpg
## alfaromeo          26
## audi               25
## bmw                29
## chevrolet          43
## dodge1             30
## dodge2             38
```

We are going to consider the block of predictors X to be formed by the first 12 variables. In turn, the block of responses Y is going to be formed by variables `price`, `symbol`, `city.mpg` and `highway.mpg`.

3 PLS Regression 2

The idea behind PLS regression is to look for components T and U that allow us to decompose the block of predictors and the block of responses, respectively. In matrix notation, the desired decompositions have the the following expressions:

$$X = TP + Residuals_x$$

$$Y = UC + Residuals_y$$

In addition, we want components T to be able to explain the response variables Y, that is:

$$Y = TD + Error$$

I'm not going to explain the details behind PLS regression. To know more about the nitty gritty of PLS regression, good references are:

- Geladi P., Kowalski B (1986) Partial Least Squares Regression: A tutorial. *Analytica Chimica Acta*, 185: 1-17.
- Tenenhaus M. (1998) *La Regression PLS: Theorie et pratique.* Paris: Editions TECHNIP.
- Helland I.S. (2001) Some theoretical aspects of partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 58: 97-107.

4 Function plsreg2

The package `plsdepot` provides the function `plsreg2()` for performing PLS regression with more than one response variable. `plsreg2()` has 4 arguments: `predictors`, `responses`, `comps` and `crosval`. Obviously the first argument `predictors` is the data containing the predictors. This can be either a matrix or a data frame. The second argment, `responses`, is the data containg the response variable, which can be a matrix or a data frame. The third argument `comps` specifies the number of extracted components. And finally, `crosval` indicates whether to perform cross-validation(`TRUE` by default). Let's apply `plsreg2()` on `vehicles`, asking for three components.

```
# apply plsreg2
my_pls2 = plsreg2(vehicles[, 1:12], vehicles[, 13:16], comps = 3)

# what's in my_pls2?
my_pls2

##
## PLS Regression 2
## ---------------------------------------------------
## $x.scores     X-scores (T-components)
## $x.loads      X-loadings
## $y.scores     Y-scores (U-components)
## $y.loads      Y-loadings
## $cor.xt       X,T correlations
## $cor.yt       Y,T correlations
## $cor.xu       X,U correlations
## $cor.yu       Y,U correlations
## $cor.tu       T,U correlations
## $raw.wgs      raw weights
## $mod.wgs      modified weights
## $std.coefs    standard coefficients
## $reg.coefs    regular coefficients
## $y.pred       Y-predicted
## $resid        residuals
## $expvar       explained variance
## $VIP          variable importance for projection
## $Q2           Q2 index
## $Q2cum        cummulated Q2
## ---------------------------------------------------
##
```

What you get in `my_pls2` is an object of class `"plsreg2"`, and everytime you print an object of such class you get a display with the list of results.

4.1 PLS components

The first two elements in the list are `$x.scores` and `$x.loads` which contains the extracted PLS components and its loadings, respectively.

```r
# T components
head(my_pls2$x.scores)

##                 t1      t2      t3
## alfaromeo  0.1634  2.3114  0.8988
## audi       2.2382  0.1906 -0.6049
## bmw       -0.4585  0.8810  0.5309
## chevrolet -2.2470 -0.4829 -0.8480
## dodge1    -1.8938  1.0899 -0.1562
## dodge2    -2.7014  0.1416 -1.0572

# X-loadings
my_pls2$x.loads

##                   p1       p2        p3
## diesel       0.1481 -0.46578  0.572526
## turbo        0.2206 -0.18751  0.008393
## two.doors   -0.1377  0.34001  0.823421
## hatchback   -0.1828  0.27390  0.133586
## wheel.base   0.3833 -0.18221 -0.050316
## length       0.4041 -0.10625 -0.064635
## width        0.3866 -0.04469 -0.017983
## height       0.2599 -0.32601  0.108401
## curb.weight  0.4173  0.05085  0.081731
## eng.size     0.3631  0.24512  0.148532
## horsepower   0.2880  0.44592  0.012545
## peak.rpm    -0.1626  0.39942 -0.350785
```

The third and fourth elements are `$y.scores` and `$y.loads`, which are the U components and loadings associated to the response variable.

```r
# U components
head(my_pls2$y.scores)

##                 u1       u2       u3
## alfaromeo  1.9307  2.26543 -0.96542
## audi       2.2036  1.14652  0.54665
## bmw        0.5358  2.56191  1.03809
## chevrolet -3.5871 -2.67647 -0.01675
## dodge1    -0.2125  1.62574 -1.31521
## dodge2    -2.4455  0.05351 -0.21449

# Y-loadings
my_pls2$y.loads

##                   c1      c2     c3
## price        0.3442  0.2537 0.1743
## symbol      -0.1834  0.3177 0.4701
```

```
## city.mpg    -0.3084 -0.3359 0.2022
## highway.mpg -0.3463 -0.2311 0.1751
```

4.2 Correlations between variables and components

As with the **plsreg1()** function, **plsreg2()** also provides the correlations between variables and components. The difference in this case is that the correlations are divided into five tables. The first two tables are the correlations of each group of components with its set of variables

```
# correlations between X and T
my_pls2$cor.xt

##                  t1       t2        t3
## diesel       0.3469 -0.70811  0.472795
## turbo        0.5168 -0.28506  0.006931
## two.doors   -0.3225  0.51690  0.679986
## hatchback   -0.4282  0.41640  0.110316
## wheel.base   0.8979 -0.27701 -0.041551
## length       0.9468 -0.16152 -0.053376
## width        0.9057 -0.06794 -0.014851
## height       0.6090 -0.49561  0.089518
## curb.weight  0.9776  0.07730  0.067494
## eng.size     0.8506  0.37264  0.122659
## horsepower   0.6746  0.67790  0.010360
## peak.rpm    -0.3810  0.60722 -0.289680

# correlations between Y and U
my_pls2$cor.yu

##                  u1      u2     u3
## price        0.8573  0.3894 0.2034
## symbol      -0.3376  0.6914 0.6811
## city.mpg    -0.9263 -0.5779 0.2210
## highway.mpg -0.9631 -0.4145 0.2738
```

The second kind of correlations are the cross-correlations: X, U and Y, T

```
# correlations between X and U
my_pls2$cor.xu

##                   u1       u2        u3
## diesel       0.03981 -0.52934  0.284545
## turbo        0.39333 -0.25692 -0.120854
## two.doors   -0.21570  0.48379  0.347775
## hatchback   -0.29019  0.26049 -0.052160
## wheel.base   0.66920 -0.30769 -0.047506
## length       0.77180 -0.10145  0.007463
```

```
## width        0.72966 -0.09452  0.055646
## height       0.34722 -0.36082  0.030004
## curb.weight  0.88900  0.06269  0.009475
## eng.size     0.87386  0.31362  0.078575
## horsepower   0.84448  0.59342 -0.018841
## peak.rpm    -0.10331  0.48974 -0.095885
```

```
# correlations between Y and T
my_pls2$cor.yt
```

```
##                 t1      t2     t3
## price        0.8064  0.3856 0.1439
## symbol      -0.4298  0.4830 0.3882
## city.mpg    -0.7224 -0.5107 0.1670
## highway.mpg -0.8113 -0.3513 0.1446
```

The fifth table of correlations is the one with the cross-correlations: T, U

```
# correlations between T and U
my_pls2$cor.tu
```

```
##         u1        u2         u3
## t1  0.8811 1.367e-16 -9.063e-17
## t2  0.3312 8.285e-01 -1.014e-16
## t3 -0.1263 1.156e-01  5.949e-01
```

4.3 Modified weights

The modified weights, `$mod.wgs`, are the weights W^* used for calculating the components with the original block of predictors: $X = TW^*$.

```
# pls components
head(my_pls2$mod.wgs)
```

```
##                w*1      w*2       w*3
## diesel      0.01926 -0.41552  0.546380
## turbo       0.19052 -0.15751 -0.271090
## two.doors  -0.10445  0.35866  0.742260
## hatchback  -0.14056  0.17251 -0.083443
## wheel.base  0.32416 -0.16520 -0.131618
## length      0.37388  0.01066 -0.009839
```

4.4 Regression coefficients

The function `plsreg2()` provides two types of regression coefficients. `$std.coefs` are the coefficients for the standardized data. `$reg.coefs` are the coefficients for the unstandardized data, and so there will be an intercept term.

```r
# standardized coefficients
round(my_pls2$std.coefs, 4)

##                price  symbol city.mpg highway.mpg
## diesel      -0.0035  0.1213   0.2441      0.1851
## turbo       -0.0216 -0.2124  -0.0606     -0.0771
## two.doors    0.1844  0.4821   0.0618      0.0833
## hatchback   -0.0192  0.0414  -0.0315     -0.0058
## wheel.base   0.0467 -0.1738  -0.0711     -0.0971
## length       0.1297 -0.0698  -0.1209     -0.1337
## width        0.1401 -0.0191  -0.0946     -0.1093
## height       0.0010 -0.0944   0.0368      0.0039
## curb.weight  0.1877 -0.0292  -0.1845     -0.1847
## eng.size     0.2625  0.1083  -0.2170     -0.2004
## horsepower   0.2816  0.0954  -0.3226     -0.2777
## peak.rpm     0.0499  0.0521  -0.1441     -0.0983

# regular (unstandardized) coefficients
round(my_pls2$reg.coefs, 4)

##                   price  symbol city.mpg highway.mpg
## diesel      -7.514e+01  0.3377   3.8735      3.0237
## turbo       -4.354e+02 -0.5593  -0.9101     -1.1907
## two.doors    3.258e+03  1.1139   0.8140      1.1296
## hatchback   -3.560e+02  0.1005  -0.4360     -0.0828
## wheel.base   5.798e+01 -0.0282  -0.0658     -0.0925
## length       7.978e+01 -0.0056  -0.0554     -0.0631
## width        4.973e+02 -0.0089  -0.2504     -0.2977
## height       4.296e+00 -0.0534   0.1186      0.0128
## curb.weight  2.838e+00 -0.0001  -0.0021     -0.0021
## eng.size     5.993e+01  0.0032  -0.0369     -0.0351
## horsepower   6.147e+01  0.0027  -0.0525     -0.0465
## peak.rpm     8.851e-01  0.0001  -0.0019     -0.0013
## INTERCEPT   -6.587e+04  6.3141  76.7841     91.2076
```

For instance, if you want to get the regression model for `price` expressed in terms of the predictors in the original scale, you need to use the regular coefficients:

$$\widehat{\texttt{price}} = -65872.69 - 75.13\texttt{diesel} - 435.35\texttt{turbo} + 3257.56\texttt{two.doors}$$

$$-356.02\texttt{hatchback} + 57.97\texttt{wheel.base} + 79.78\texttt{length} + 497.30\texttt{width}$$

$$+4.29\texttt{height} + 2.83\texttt{curb.weight} + 59.92\texttt{eng.size} + 61.46\texttt{horsepower} + 0.88\texttt{peak.rpm}$$

4.5 R-squared and explained variance

Another important result has to do with the R-squared coefficient and the variance proportion explained by the PLS components. The first two columns refer to the predictors X. Columns three and four refer to the responses Y.

```
# explained variance
my_pls2$expvar

##          R2X R2Xcum     R2Y R2Ycum
## t1 0.48735 0.4874 0.50375 0.5037
## t2 0.19578 0.6831 0.19155 0.6953
## t3 0.06788 0.7510 0.05506 0.7504
```

The column `R2X` shows the R2 coefficient of each component for the predictor variables. The column `R2Xcum` is the cumulative value of the R2 coefficients.

4.6 Variable Importance for Projection

Together witn `$expvar`, we also have the Variable Importance for Projection in `$VIP`. VIP measures the explanatory power of a given predictor x_j on the block of responses Y,

```
# VIP
my_pls2$VIP

##                   t1     t2     t3
## diesel      0.06672 0.7662 0.9162
## turbo       0.65998 0.6732 0.6879
## two.doors   0.36182 0.7633 0.9905
## hatchback   0.48691 0.5596 0.5478
## wheel.base  1.12292 1.0540 1.0186
## length      1.29515 1.1121 1.0706
## width       1.22445 1.0511 1.0174
## height      0.58261 0.7192 0.6947
## curb.weight 1.49186 1.2731 1.2256
## eng.size    1.46650 1.3278 1.2870
## horsepower  1.41726 1.4795 1.4247
## peak.rpm    0.17328 0.7222 0.7189
```

We could measure the contribution of a given variable x_j to the construction of a PLS component t_h by calculating the squared of weights w_{hj}^2. However, VIP gives us another way to classify the predictors in terms of their explanatory power of Y. Those predictors with a VIP >1 are considered to be the most relevant to the construction of Y.

4.7 Cross-validation results

The function `plsreg2()` offers the option to perform a cross-validation procedure. In this process the data set is randomly split in 10 segments of approximately equal size. Then, the observations in one of the segments are left outside as a test set. The other nine segments are used as learning set to estimate a model and predict the observations in the test segment. This procedure is applied consecutively for each of the 10 segments. The cross-valition results can be checked in `$Q2`.

$$Q_h^2 = 1 - \frac{PRESS_h}{RSS_{h-1}}$$

where RSS_{h-1} is the squared sum of residuals using the t_{h-1} component, and $PRESS_h$ is the PRediction Error Sum of Squares using the t_h component. A component t_h is considered to be significant if Q_h^2 is greater than or equal to 0.0975

```
# cross-validation indices
my_pls2$Q2

##    Q2.price Q2.symbol Q2.city.mpg Q2.highway.mpg       Q2
## t1  0.59543   0.16607     0.46199         0.6183  0.46046
## t2  0.25260   0.23102     0.49101         0.2947  0.30842
## t3 -0.05588   0.04256    -0.02971        -0.1714 -0.02491

# cumulative cross-validation indices
my_pls2$Q2cum

##    Q2cum.price Q2cum.symbol Q2cum.city.mpg Q2cum.highway.mpg  Q2cum
## t1      0.5954       0.1661         0.4620            0.6183 0.4605
## t2      0.6976       0.3587         0.7262            0.7308 0.6269
## t3      0.6807       0.3860         0.7180            0.6847 0.5332
```

`my_pls2$Q2` contains the Q_2 index for each component on each response variable. The last column is the mean value in each row. Likewise, `my_pls2$Q2cum` contains the cumulative values of `my_pls2$Q2`.

5 Plotting "plsreg2" 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.

5.1 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.

```
# default plot (variabls inside circle of correlations)
plot(my_pls2)
```

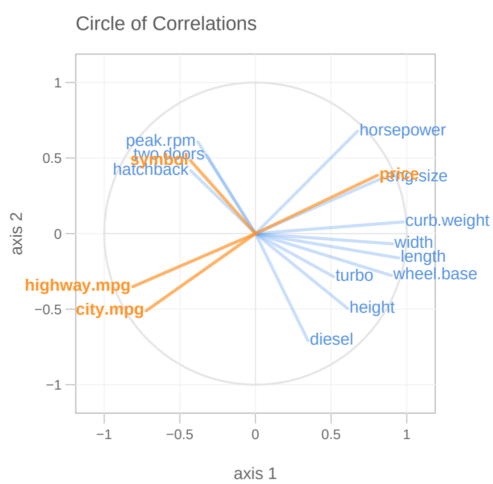

Figure 1: Circle of correlations

5.2 Plotting observations

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

```
# plot of observations
plot(my_pls2, what = "observations", show.names = TRUE)
```

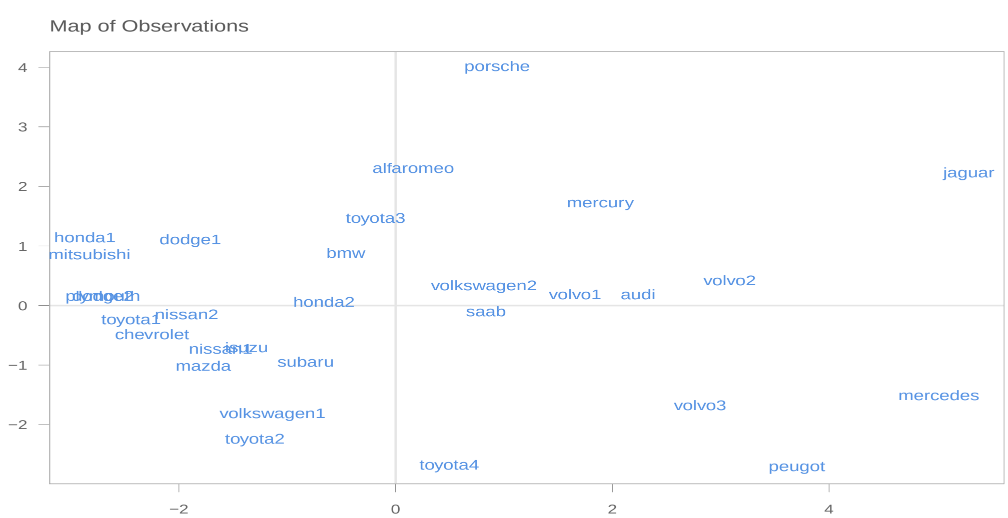

Figure 2: Plot of observations