MIRI
Data Mining & Business Intelligence

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Session 5
Partial Least Squares Regression

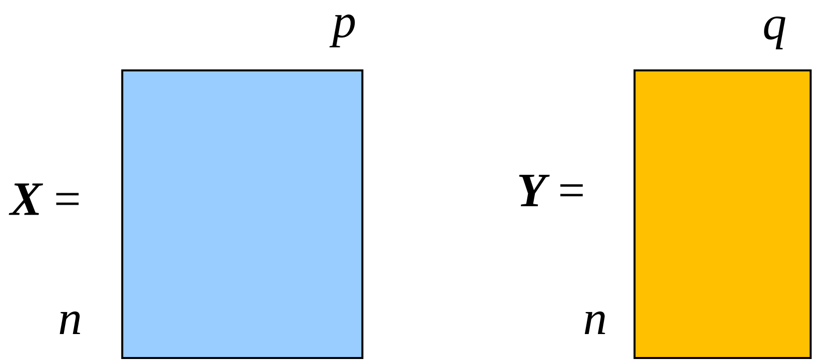
Course on Multivariate Modeling. *KBLMM part*

Tomàs Aluja-Banet

tomas.aluja@upc.edu

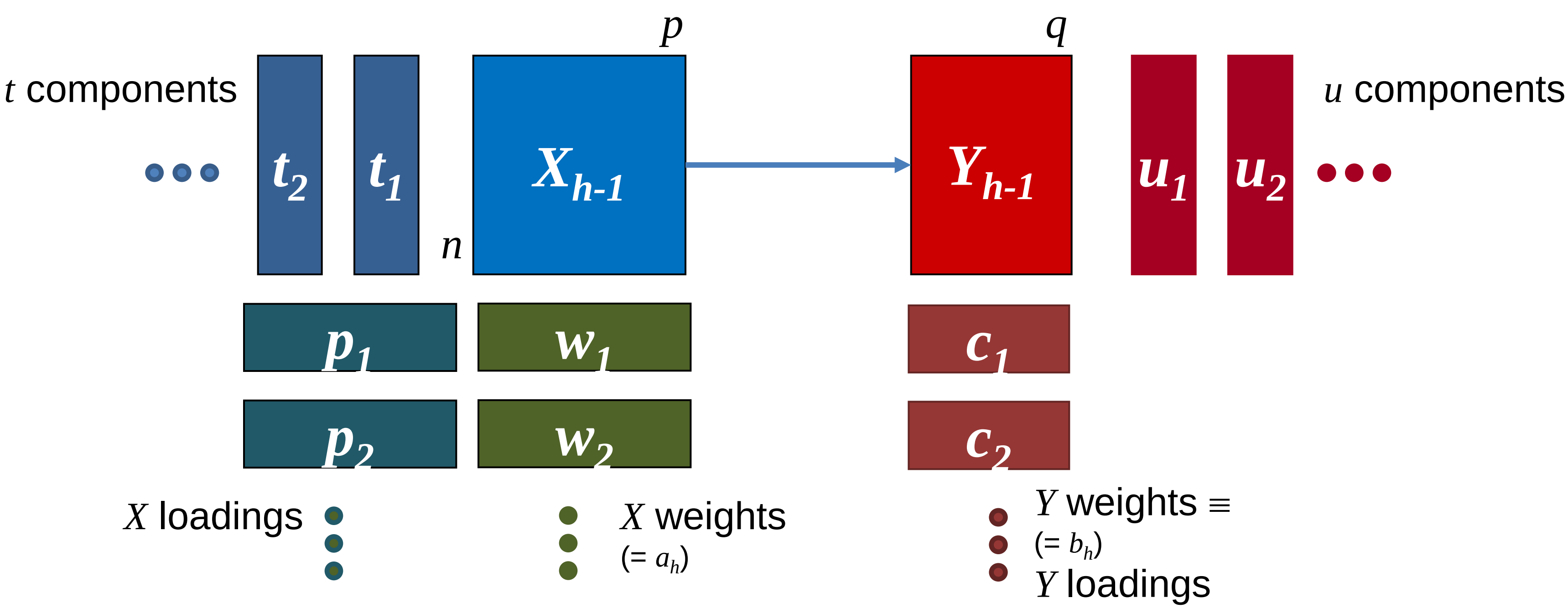
INFORMATION MODELING & PROCESSING
information matters

PLS Regression - PLS2



- **PLS2** refers to the situation when the response block Y has more than one variable (multivariate modeling)
- The goals are the same as in PLS1. We want m orthogonal components t_h and m components u_h, well correlated between them and explaining their own groups.
- The number of components m is determined by cross validation
- Finally, we regress Y on the m t_h components
- We express the regression equation in terms of X
- It allows missing data
- Iterative algorithm

The elements of PLS2



$$\boxed{\begin{array}{l} \max \operatorname{cov}(t_h, u_h) \\ t_h = X_{h-1} w_h \quad u_h = Y_{h-1} c_h \\ t_h' t_{l<h} = 0 \quad w_h' w_h = 1 \end{array}}$$

$$X_h = X_{h-1} - t_h p_h'$$

Deflation of X_{h-1} respect to t_h

$$Y_h = Y_{h-1} - t_h c_h'$$

Deflation of Y_{h-1} respect to t_h

PLS2 algorithm

$$X_0 = X; Y_0 = Y$$

$$h = 1 \cdots rang(X)$$

$$u_h = meanRows(Y_{h-1})$$

iterate till convergence of w_h

$$w_h = X'_{h-1} u_h \,/\, u'_h u_h$$

$$\|w_h\| = 1$$

$$t_h = X_{h-1} w_h \,/\, w'_h w_h$$

$$c_h = Y'_{h-1} t_h \,/\, t'_h t_h$$

$$u_h = Y_{h-1} c_h \,/\, c'_h c_h$$

$$p_h = X'_{h-1} t_h / t'_h t_h$$

$$X_h = X_{h-1} - t_h p'_h$$

$$c_h = Y'_{h-1} t_h \,/\, t'_h t_h$$

$$Y_h = Y_{h-1} - t_h c'_h$$

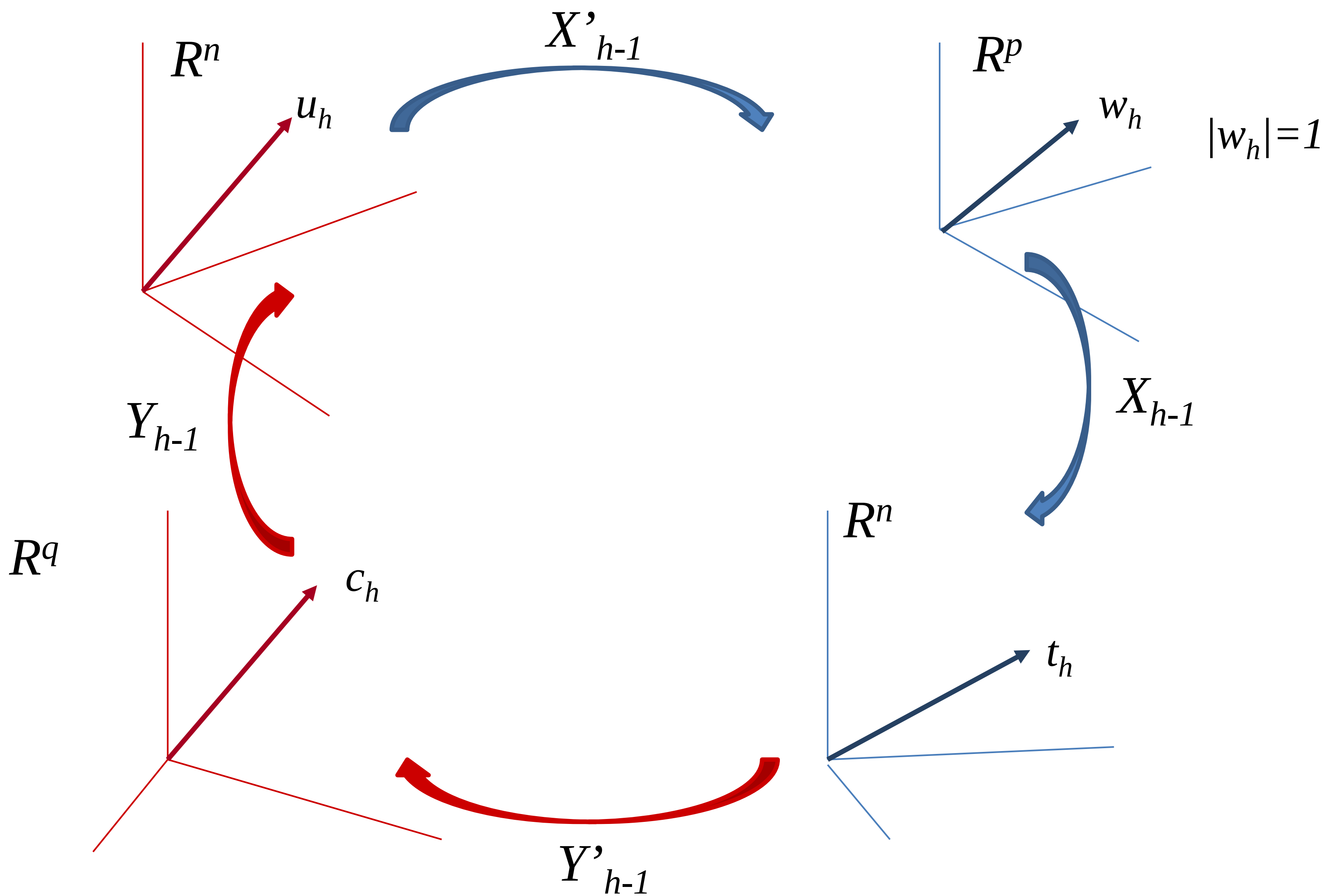
X, Y centered and eventually standardized

$$\Big\} \; w_h = X'_{h-1} u_h \,/\, \left| X'_{h-1} u_h \right|$$

Deflation of X_{h-1}

Deflation of Y_{h-1}

Geometry of iterations in PLS2



In the convergence

Convergence results:

$$(X'_{h-1}Y_{h-1})(Y'_{h-1}X_{h-1})w_h \propto w_h$$
$$(Y'_{h-1}X_{h-1})(X'_{h-1}Y_{h-1})c_h \propto c_h$$

$$\equiv IBA(X_{h-1}, Y_{h-1})$$

In every iteration we obtain equivalent results to the IBA of the residual matrices of X and Y

$$\Rightarrow \max \operatorname{cov}(t_h, u_h)$$

$$t'_h t_{l<h} = 0$$

but now, components t_h are orthogonal
and are limited by the *rang(X)*

We use the t_h components to explain its own block and to predict the Y block

Properties of the PLSR components

PLS1 regression leads in each step to a compromise between multiple regression of y on X_{h-1} and the principal component analysis of X_{h-1}

$$Cov^2(t_h, y) \simeq Cor^2(t_h, y) \times Var(t_h)$$

PLS2 regression leads in each step to a compromise between multiple regression of u_h on X_{h-1} and the principal component analysis of X_{h-1} and Y_{h-1}

$$Cov^2(t_h, u_h) = Cor^2(t_h, u_h) \times Var(t_h) \times Var(u_h)$$

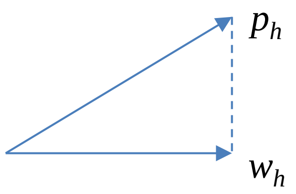
Properties

Components t_h are orthogonal to the columns of X_l and to Y_l $l>=h$

$$t'_h X_h = 0 \quad t'_h Y_h = 0$$

Components t_h are orthogonal

$$t'_h t_{h+1} = 0$$



Weights w_h are normalized

$$w'_h w_h = 1$$

Loadings p_h project on weights w_h

$$w_h \in R^p, p_h \in R^p \qquad w'_h p_h = w'_h X'_{h-1} t_h / t'_h t_h = t'_h t_h / t'_h t_h = 1$$

Weights w_h are orthogonal to the rows of X_h

$$w'_h X'_h = w'_h (X'_{h-1} - p_h t'_h) = t'_h - w'_h p_h t'_h = 0$$

Weights w_h are orthogonal

$$w'_h w_{h+1} = w'_h X'_h u_{h+1} = 0$$

Loadings p_h and weights w_l $(l>h)$ are orthogonal

$$w'_h p_{h+1} = w'_h X'_h t_{h+1} / t'_{h+1} t_{h+1} = 0$$

Modeling the Y block from the t_h components

$$Y = t_1 c_1' + Y_1$$
$$Y_1 = t_2 c_2' + Y_2$$
$$\vdots$$
$$Y_{h-1} = t_h c_h' + Y_h$$

$$Y = t_1 c_1' + \cdots + t_h c_h' + Y_h$$

$$T_h = [t_1, t_2, \ldots, t_h]$$
$$C_h = [c_1, c_2, \ldots, c_h]$$

$$Y = T_h C_h' + Y_h = \hat{Y}_h + Y_h$$

$$\hat{Y}_h = T_h C_h'$$

The predictions and confidence intervals are obtained from the regression of y_j on the t_h components with the usual formulae

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Expressing the model as function of the X block

Matrix of components $T_h = [t_1, t_2, \ldots, t_h]$ $\quad t_1 = X_0 w_1 \quad p_1 = X_0' t_1 / t_1' t_1 \quad p_1 = X' t_1 / t_1' t_1$

Matrix of weights $W_h = [w_1, w_2, \ldots, w_h]$ $\quad \vdots \quad \vdots \quad \vdots$

Matrix of loadings $P_h = [p_1, p_2, \ldots, p_h]$ $\quad t_h = X_{h-1} w_h \quad p_h = X_{h-1}' t_h / t_h' t_h \quad p_h = X' t_h / t_h' t_h$

$$X_{h-1}' t_h = X_{h-1}' X_{h-1} w_h = X'(I - \Pi_{h-1})(I - \Pi_{h-1}) X w_h = X' X_{h-1} w_h = X' t_h$$

$$N_h^2 = \begin{pmatrix} t_1' t_1 & & \\ & \ddots & \\ & & t_h' t_h \end{pmatrix} \longrightarrow P_h = X' T_h N_h^{-2} = X' T_h^s N_h^{-1} \qquad T_h^s = T_h N_h^{-1}$$

$$T_h^{s\prime} T_h^s = I \qquad W_h' W_h = I$$

$$X = T_h^s A \qquad X' = W_h B$$

$$X = B' W_h'$$

A and B "convenient" transformation matrices (h,p) and (h,n) respectively

$$B' W_h' = T_h^s A \qquad B' = T_h^s A W_h$$

$$A = T_h^{s\prime} X \qquad B' = X W_h = T_h^s T_h^{s\prime} X W_h$$

Finding the coefficients b_j

$$XW_h = T_h N_h^{-2} T_h' X W_h = T_h P_h' W_h$$

$$T_h = XW_h \underbrace{(P_h'W_h)^{-1}}_{\text{Projection matrix}}$$

$$\hat{Y}_h = T_h C_h' = XW_h (P_h'W_h)^{-1} C_h' = XB_h$$

$$B_h = W_h (P_h'W_h)^{-1} C_h'$$

$$B_h = \begin{array}{ccc} b_1 & \cdots & b_h \\ \left(\vdots\right. & \cdots & \left.\vdots\right) \end{array}$$

y_j is explained from the coefficients with the original variables x_j

Number of components

The number of components are taken by crossvalidation (usually LOO)

$$\hat{Y}_h = T_h C'_h$$

$$RSS_h = \left\| Y - \hat{Y}_h \right\|^2$$

$$\hat{Y}_{h(-i)} = T_{h(-i)} C'_{h(-i)} = X_{(-i)} B_{h(-i)}$$

$$PRESS_h = \left\| Y - Y_{(-i)h} \right\|^2$$

$$RMSEP_{cv} = \frac{PRESS_h}{n}$$

$$R^2_{cv} = 1 - \frac{PRESS_h}{\sum_i (y_i - \bar{y})^2}$$

A component is taken if the $RMSEP$ decreases (or the $R_{cv}{}^2$ increases)

Detecting outliers

m selected number of components

Respect to Y

$$E = Y - T_m C'_m$$

$$s_Y = \sqrt{\frac{\sum_i^n \sum_j^q e_{ji}^2}{(n-m-1)(q-m)}} \qquad DModY_i = \sqrt{\frac{\sum_j^q e_{ji}^2}{q-m}} \qquad DModY_i^{nor} = \frac{DModY_i}{s_Y}$$

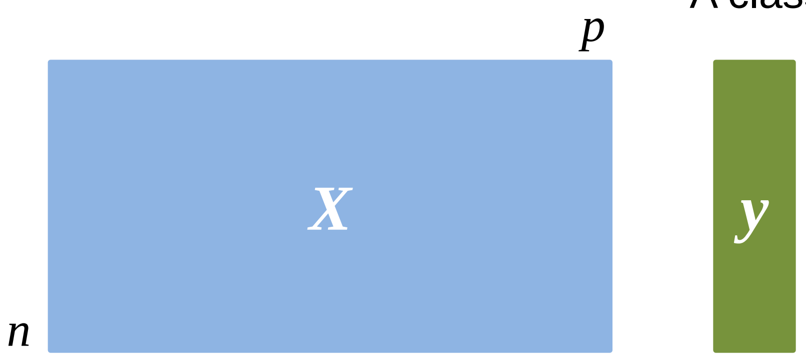
Respect to X

$$F = X - T_m P'_m$$

$$s_X = \sqrt{\frac{\sum_i^n \sum_j^p f_{ji}^2}{(n-m-1)(p-m)}} \qquad DModX_i = \sqrt{\frac{\sum_j^p f_{ji}^2}{p-m}} \times \sqrt{\frac{n}{n-m-1}} \qquad DModX_i^{nor} = \frac{DModX_i}{s_X}$$

LDA – Partial Least squares

A classifying vector in m categories



An important application of microarray technology is tumor diagnosis, i.e. class prediction. The output of n microarray experiments can be summarized as a $n << p$ data matrix, where p is the number of analyzed genes. p is always much larger than the number of experiments n. The response vector is a categorical vector telling the kind of tumor or classes.

The objective is to find a reduced set of new components t_h of X able to predict the response variable y

The default way of dealing with categorical variables is just binarizing

Algorithm: LDA on PLS components

1. PLS regression of the binary variables describing the categories of Y on the X variables
2. LDA of the Y variable on the PLS components
3. Select the number of components by CV (or test sample), minimizing the misclassification error

(simple algorithm, but according Tenenhaus it works as well as other sophisticated ones, Boulasteix 2004, follows the same approach).

Instead of a LDA it can be performed a Logistic Regression or SVM upon the PLS components.

Oliveoil problem

We want to predict the origin of a olive oil (Greek, Italian or Spanish)
We have two block of information: Chemical constituents and Sensory data obtained from a panel of experts

Chemical vars. **Sensory vars.**

	y	Acid	Perox	K232	K270	DK	yellow	green	brown	glossy	transp	syrup
G1	greek	0.73	12.7	1.9	0.139	0.003	21.4	73.4	10.1	79.7	75.2	50.3
G2	greek	0.19	12.3	1.678	0.116	-0.004	23.4	66.3	9.8	77.8	68.7	51.7
G3	greek	0.26	10.3	1.629	0.116	-0.005	32.7	53.5	8.7	82.3	83.2	45.4
G4	greek	0.67	13.7	1.701	0.168	-0.002	30.2	58.3	12.2	81.1	77.1	47.8
G5	greek	0.52	11.2	1.539	0.119	-0.001	51.8	32.5	8	72.4	65.3	46.5
I1	italian	0.26	18.7	2.117	0.142	0.001	40.7	42.9	20.1	67.7	63.5	52.2
I2	italian	0.24	15.3	1.891	0.116	0	53.8	30.4	11.5	77.8	77.3	45.2
I3	italian	0.3	18.5	1.908	0.125	0.001	26.4	66.5	14.2	78.7	74.6	51.8
I4	italian	0.35	15.6	1.824	0.104	0	65.7	12.1	10.3	81.6	79.6	48.3
I5	italian	0.19	19.4	2.222	0.158	-0.003	45	31.9	28.4	75.7	72.9	52.8
S1	spanish	0.15	10.5	1.522	0.116	-0.004	70.9	12.2	10.8	87.7	88.1	44.5
S2	spanish	0.16	8.14	1.527	0.1063	-0.002	73.5	9.7	8.3	89.9	89.7	42.3
S3	spanish	0.27	12.5	1.555	0.093	-0.002	68.1	12	10.8	78.4	75.1	46.4
S4	spanish	0.16	11	1.573	0.094	-0.003	67.6	13.9	11.9	84.6	83.8	48.5
S5	spanish	0.24	10.8	1.331	0.085	-0.003	71.4	10.6	10.8	88.1	88.5	46.7
S6	spanish	0.3	11.4	1.415	0.093	-0.004	71.4	10	11.4	89.5	88.5	47.2

Origin prediction from their chemical constituents

DATA

> X

	Acidity	Peroxide	K232	K270	DK
G1	0.73	12.70	1.900	0.1390	0.003
G2	0.19	12.30	1.678	0.1160	-0.004
G3	0.26	10.30	1.629	0.1160	-0.005
G4	0.67	13.70	1.701	0.1680	-0.002
G5	0.52	11.20	1.539	0.1190	-0.001
I1	0.26	18.70	2.117	0.1420	0.001
I2	0.24	15.30	1.891	0.1160	0.000
I3	0.30	18.50	1.908	0.1250	0.001
I4	0.35	15.60	1.824	0.1040	0.000
I5	0.19	19.40	2.222	0.1580	-0.003
S1	0.15	10.50	1.522	0.1160	-0.004
S2	0.16	8.14	1.527	0.1063	-0.002
S3	0.27	12.50	1.555	0.0930	-0.002
S4	0.16	11.00	1.573	0.0940	-0.003
S5	0.24	10.80	1.331	0.0850	-0.003
S6	0.30	11.40	1.415	0.0930	-0.004

> Y

	greek	italian	spanish
[1,]	1	0	0
[2,]	1	0	0
[3,]	1	0	0
[4,]	1	0	0
[5,]	1	0	0
[6,]	0	1	0
[7,]	0	1	0
[8,]	0	1	0
[9,]	0	1	0
[10,]	0	1	0
[11,]	0	0	1
[12,]	0	0	1
[13,]	0	0	1
[14,]	0	0	1
[15,]	0	0	1
[16,]	0	0	1

Basic olive oil chemical analysis

Acidity – this will help you determine olive oil categorization

Oxidation level – determining the state of oxidation and age of olive oil

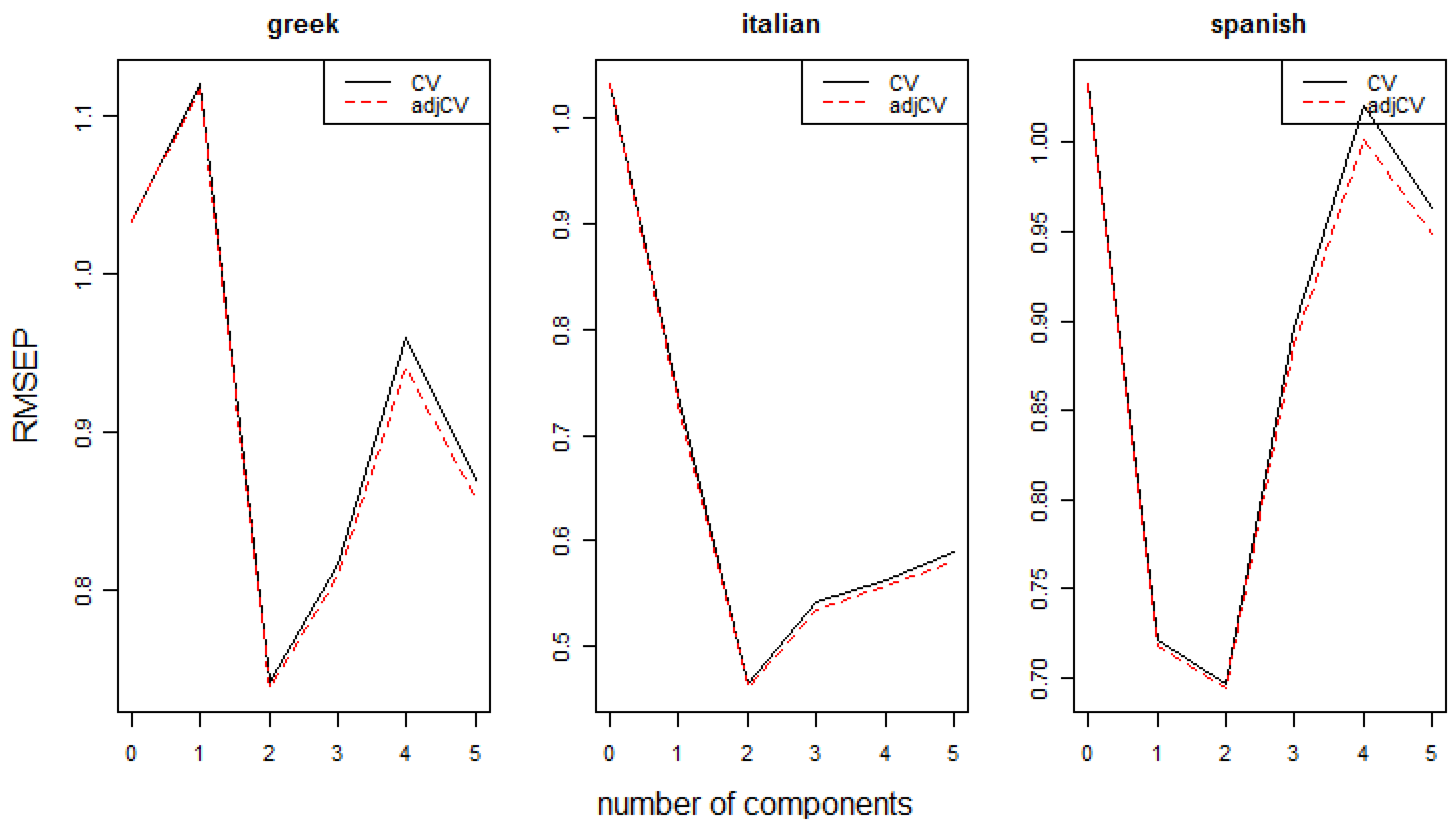
- Peroxide number which indicates the state of oxidation of olive oil
- Spectrophotometric determination of K232, K270, and DK in ultraviolet

  - K232 indicates the age of oil and how long olives have been left in sacks after harvesting , the milling process, and the storage and conditions of the olive oil and the level of oxidation incurred during production and. or storage.
  - K270 parameter test detects the level of adulteration, blends with refined olive oil Wax content which determines the level of adulteration with pomace oil.
  - DK index distinguish poor quality Olive Oil from another distorted by refined Oil, max 0,009

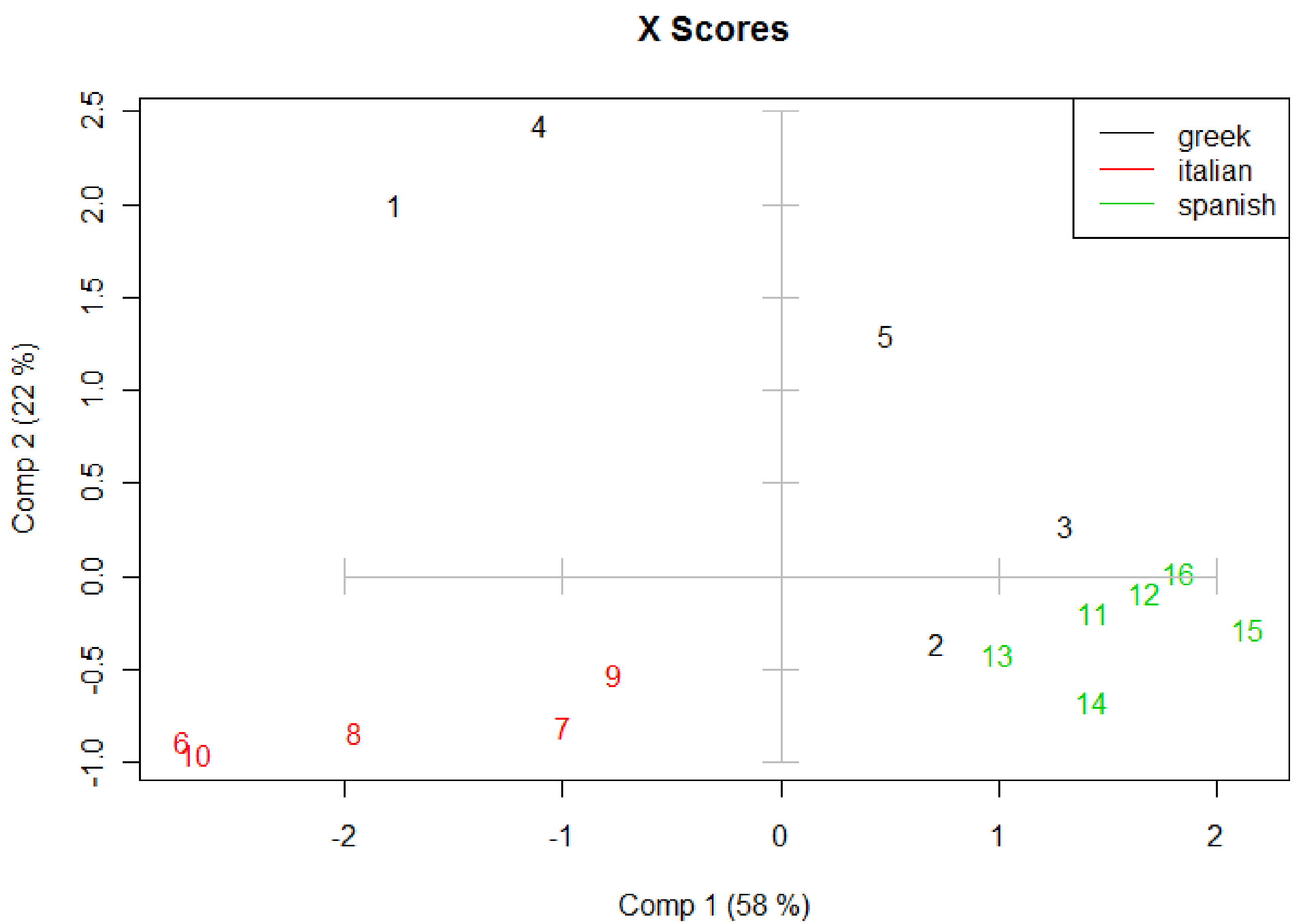
PLS2 of oliveoil to predict the origin

```
> p2 <- plsr(Ys ~ Xs, validation = "LOO")

> summary(p2)

VALIDATION: RMSEP
Cross-validated using 16 leave-one-out segments.

Response: greek
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
CV           1.033    1.120   0.7426   0.8163   0.9598   0.8699
adjCV        1.033    1.117   0.7380   0.8088   0.9407   0.8582

Response: italian
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
CV           1.033   0.7386   0.4653   0.5422   0.5622   0.5898
adjCV        1.033   0.7334   0.4602   0.5352   0.5566   0.5807

Response: spanish
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
CV           1.033   0.7210   0.6967   0.8960    1.020   0.9636
adjCV        1.033   0.7182   0.6939   0.8865    1.002   0.9493

TRAINING: % variance explained
          1 comps  2 comps  3 comps  4 comps  5 comps
X        57.87743    80.28    95.47    96.67   100.00
greek     0.08365    58.32    63.92    69.94    70.11
italian  56.86676    86.36    87.45    87.46    88.33
spanish  56.20303    60.64    62.25    67.24    67.48
```

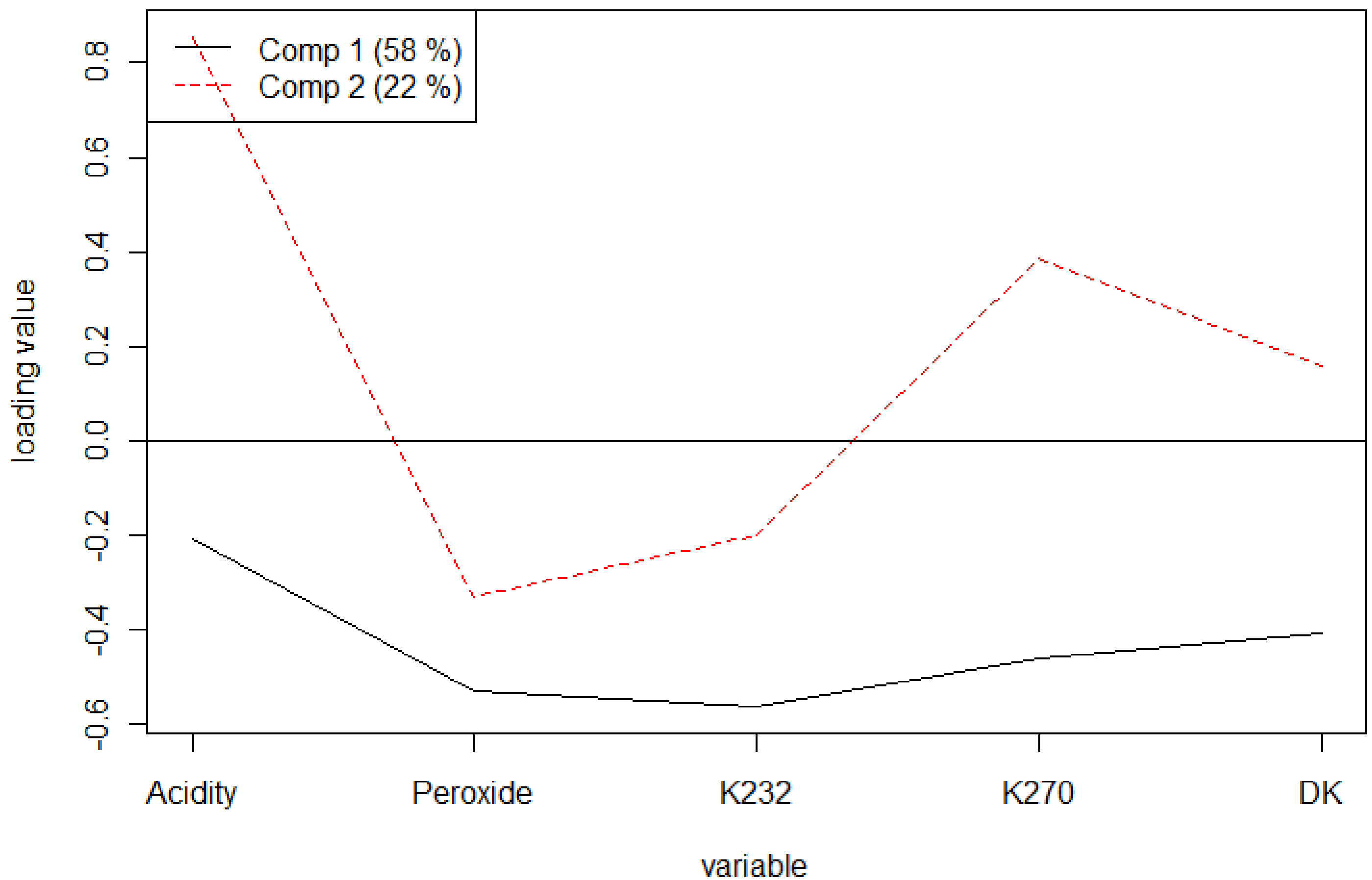
Number of significant components od chemical data



Scores plot of chemical data

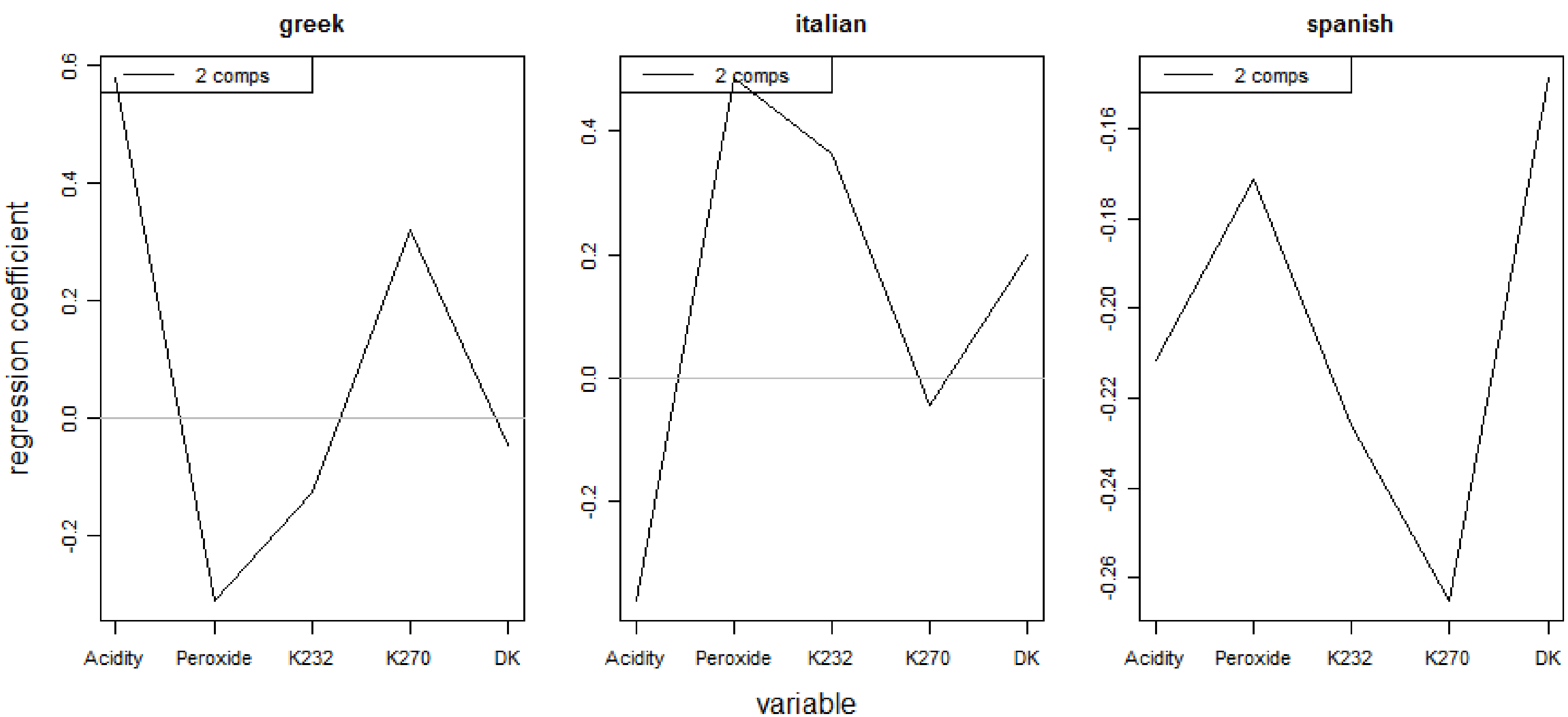


Loadings plot

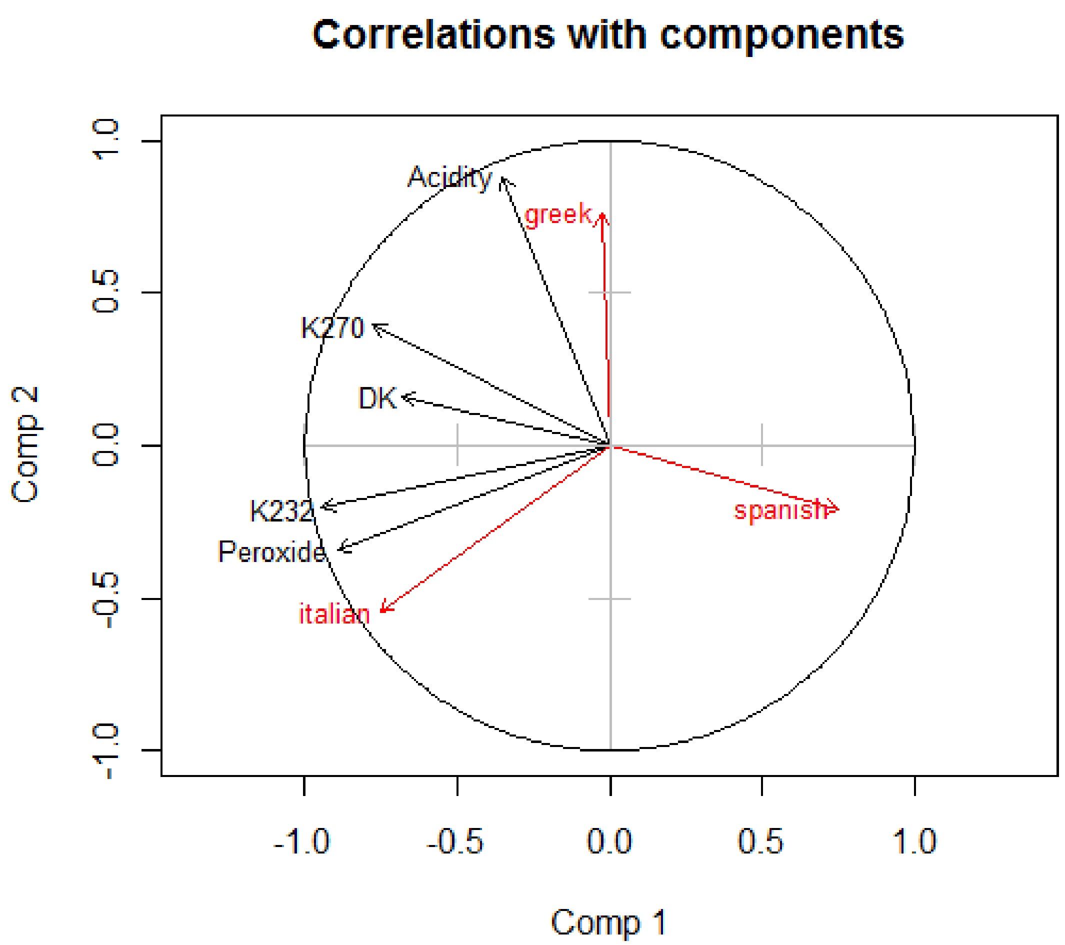


Coefficients plot

```
print(p2$coefficients[,,2],digits=4)
           greek  italian spanish
Acidity   0.58027 -0.35934 -0.2115
Peroxide -0.30838  0.48741 -0.1714
K232     -0.12619  0.36221 -0.2260
K270      0.32146 -0.04473 -0.2649
DK       -0.04503  0.20029 -0.1486
```



Correlation plot of chemical variables



LDA on PLS components

```
da <- lda(p2$scores[,1:2],y,CV=TRUE)
table(y,da$class)
```

```
y           greek italian spanish
  greek         3       0       2
  italian       0       5       0
  spanish       0       0       6
```

```
         y da.class     greek   italian   spanish
1    greek    greek 9.987e-01 3.534e-04 9.348e-04
2    greek  spanish 1.290e-03 1.043e-03 9.977e-01
3    greek  spanish 9.470e-02 1.247e-06 9.053e-01
4    greek    greek 9.995e-01 2.735e-09 4.648e-04
5    greek    greek 8.454e-01 3.930e-08 1.546e-01
6  italian  italian 5.080e-09 1.000e+00 2.106e-09
7  italian  italian 4.139e-04 9.986e-01 9.931e-04
8  italian  italian 3.241e-06 1.000e+00 4.593e-06
9  italian  italian 1.215e-02 9.675e-01 2.035e-02
10 italian  italian 3.391e-09 1.000e+00 2.026e-09
11 spanish  spanish 1.029e-01 1.126e-05 8.971e-01
12 spanish  spanish 1.073e-01 1.439e-06 8.927e-01
13 spanish  spanish 1.171e-01 3.437e-04 8.826e-01
14 spanish  spanish 4.200e-02 1.533e-04 9.578e-01
15 spanish  spanish 4.973e-02 1.620e-07 9.503e-01
16 spanish  spanish 1.256e-01 2.013e-07 8.744e-01
```

Prediction of origin using Sensory data

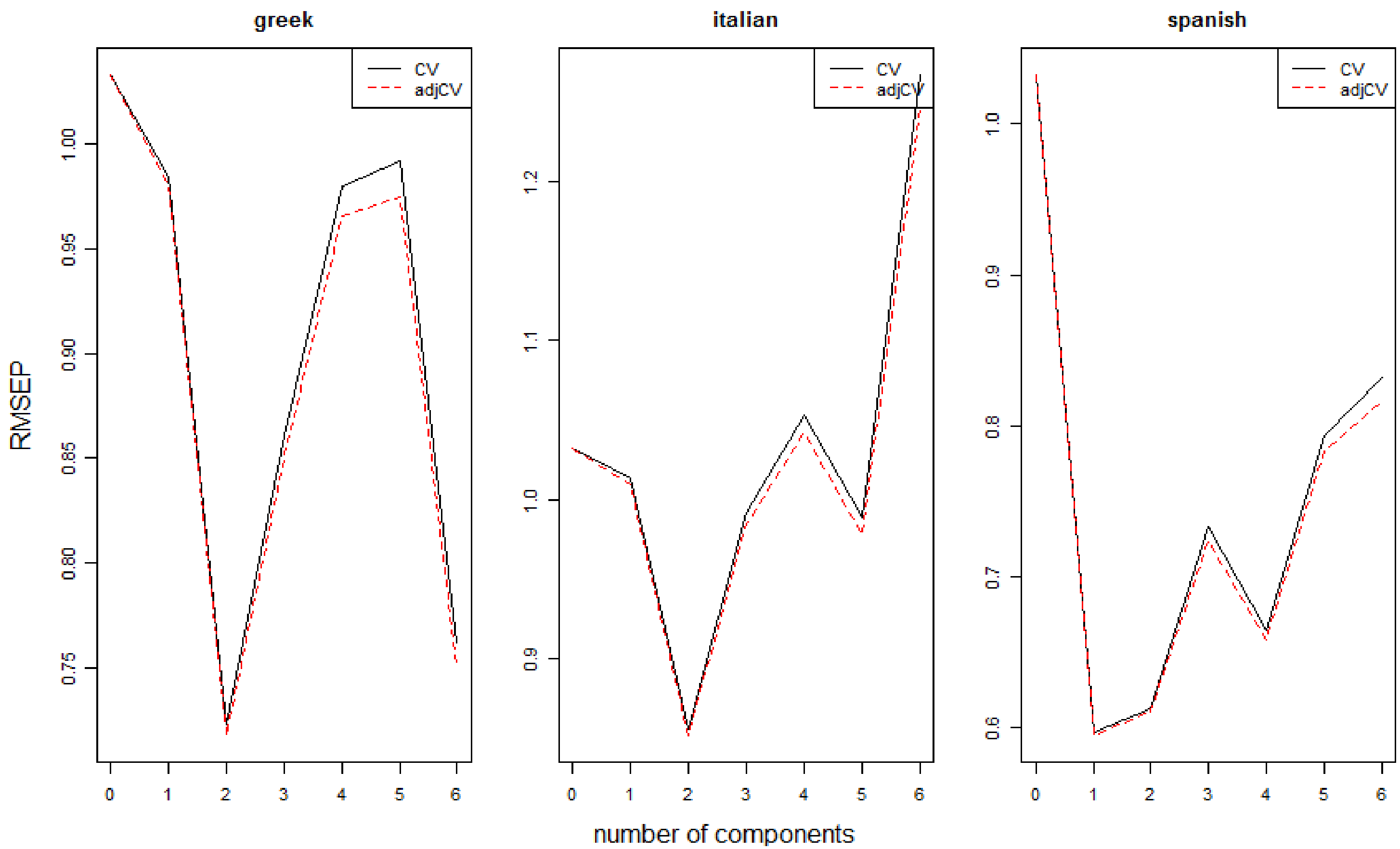
```
> print(X)
   yellow green brown glossy transp syrup
G1   21.4  73.4  10.1   79.7   75.2  50.3
G2   23.4  66.3   9.8   77.8   68.7  51.7
G3   32.7  53.5   8.7   82.3   83.2  45.4
G4   30.2  58.3  12.2   81.1   77.1  47.8
G5   51.8  32.5   8.0   72.4   65.3  46.5
I1   40.7  42.9  20.1   67.7   63.5  52.2
I2   53.8  30.4  11.5   77.8   77.3  45.2
I3   26.4  66.5  14.2   78.7   74.6  51.8
I4   65.7  12.1  10.3   81.6   79.6  48.3
I5   45.0  31.9  28.4   75.7   72.9  52.8
S1   70.9  12.2  10.8   87.7   88.1  44.5
S2   73.5   9.7   8.3   89.9   89.7  42.3
S3   68.1  12.0  10.8   78.4   75.1  46.4
S4   67.6  13.9  11.9   84.6   83.8  48.5
S5   71.4  10.6  10.8   88.1   88.5  46.7
S6   71.4  10.0  11.4   89.5   88.5  47.2
```

```
> print(Y)
      greek italian spanish
 [1,]     1       0       0
 [2,]     1       0       0
 [3,]     1       0       0
 [4,]     1       0       0
 [5,]     1       0       0
 [6,]     0       1       0
 [7,]     0       1       0
 [8,]     0       1       0
 [9,]     0       1       0
[10,]     0       1       0
[11,]     0       0       1
[12,]     0       0       1
[13,]     0       0       1
[14,]     0       0       1
[15,]     0       0       1
[16,]     0       0       1
```

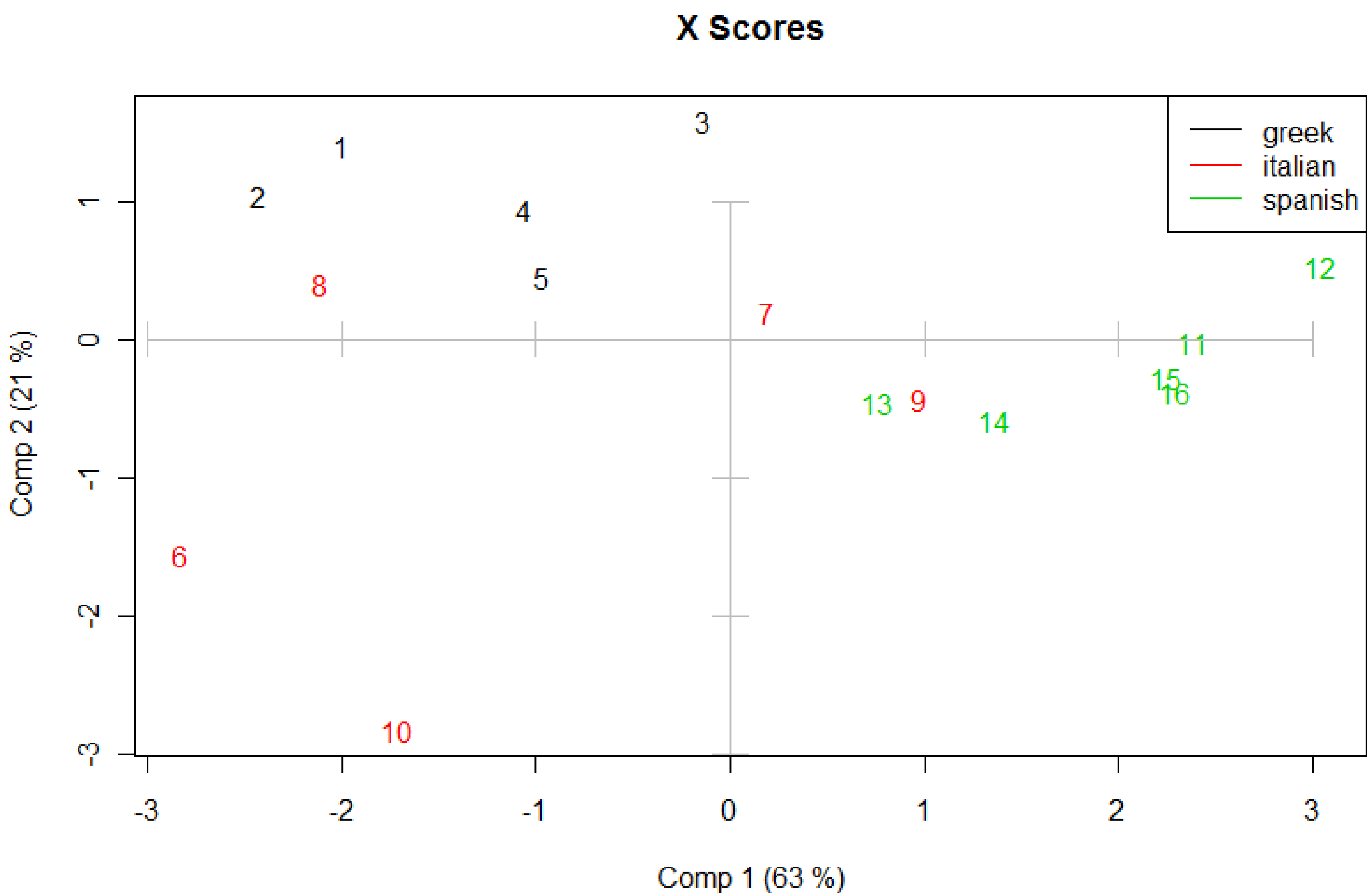
PLS2 results

```
> p2 <- plsr(Ys ~ Xs, validation = "LOO")

VALIDATION: RMSEP
Cross-validated using 16 leave-one-out segments.

Response: greek
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           1.033   0.9841   0.7232   0.8609   0.9794   0.9913   0.7622
adjCV        1.033   0.9809   0.7176   0.8503   0.9652   0.9744   0.7499

Response: italian
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           1.033    1.013   0.8554   0.9918    1.053   0.9888    1.267
adjCV        1.033    1.010   0.8515   0.9836    1.042   0.9779    1.246

Response: spanish
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           1.033   0.5975   0.6133   0.7337   0.6648   0.7933   0.8317
adjCV        1.033   0.5948   0.6103   0.7240   0.6585   0.7828   0.8168

TRAINING: % variance explained
         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X          63.39    84.13    92.15    99.53    99.93   100.00
greek      22.58    68.63    69.23    69.25    74.54    83.55
italian    15.63    43.68    44.78    46.40    56.83    56.91
spanish    69.47    71.51    74.54    75.72    76.51    83.25
```

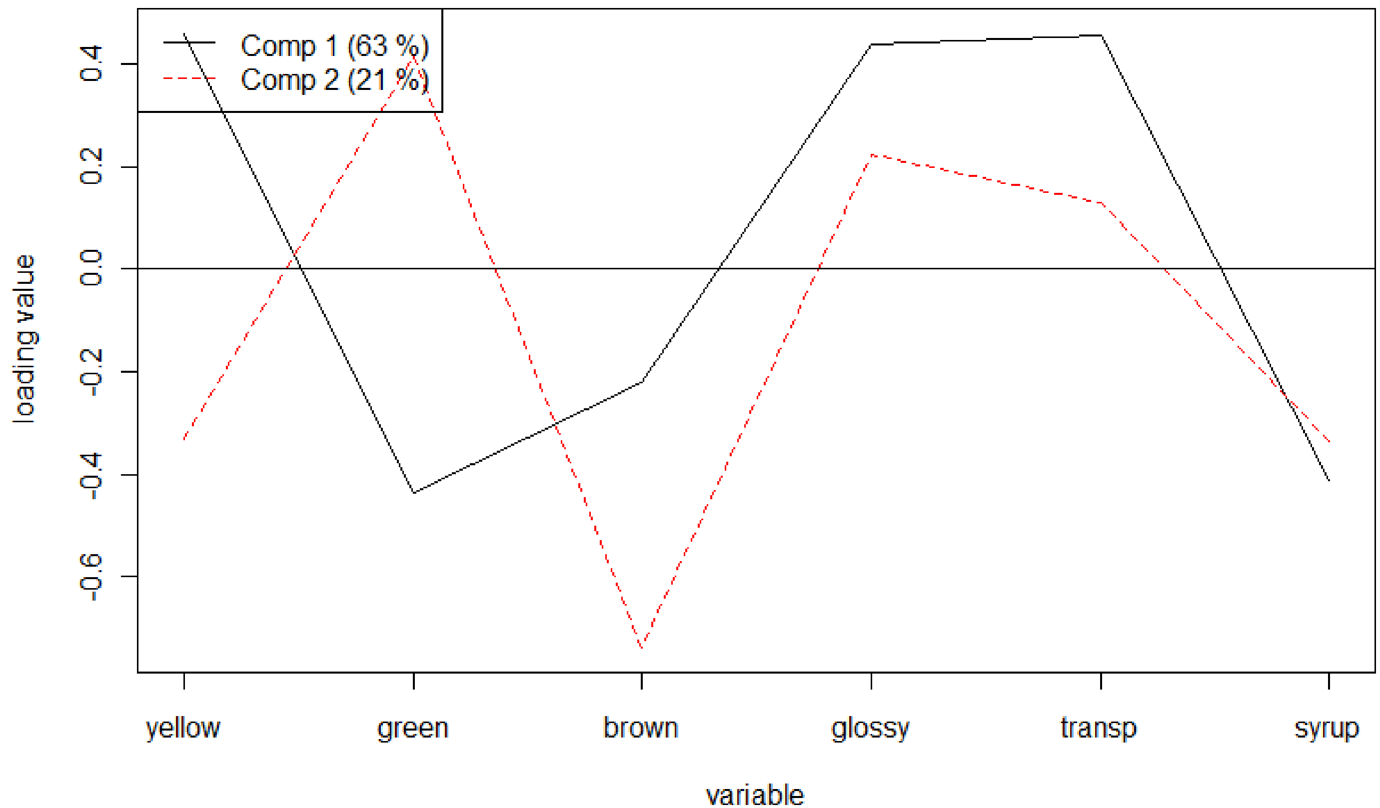
Number of components of sensory data



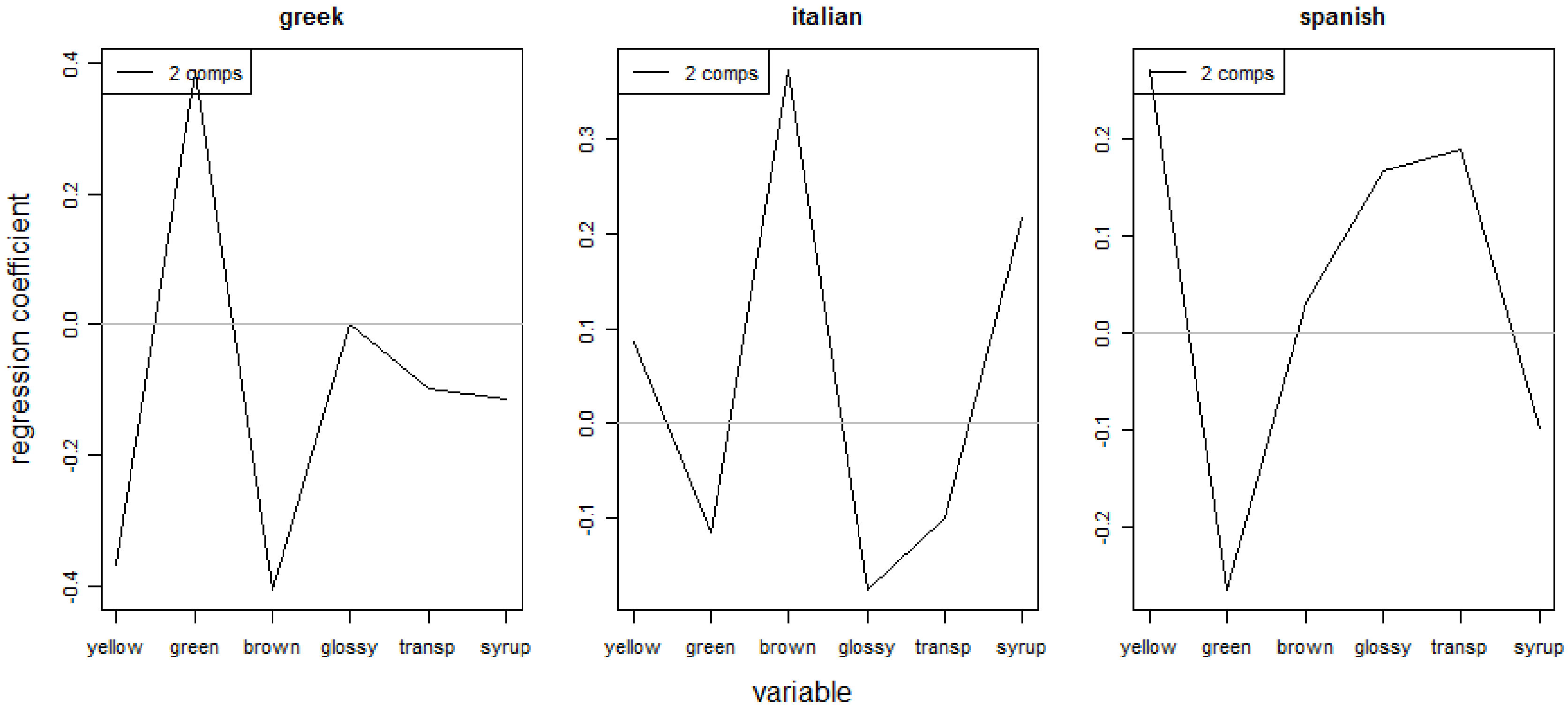
Scores plot of sensory data



Loadings plot

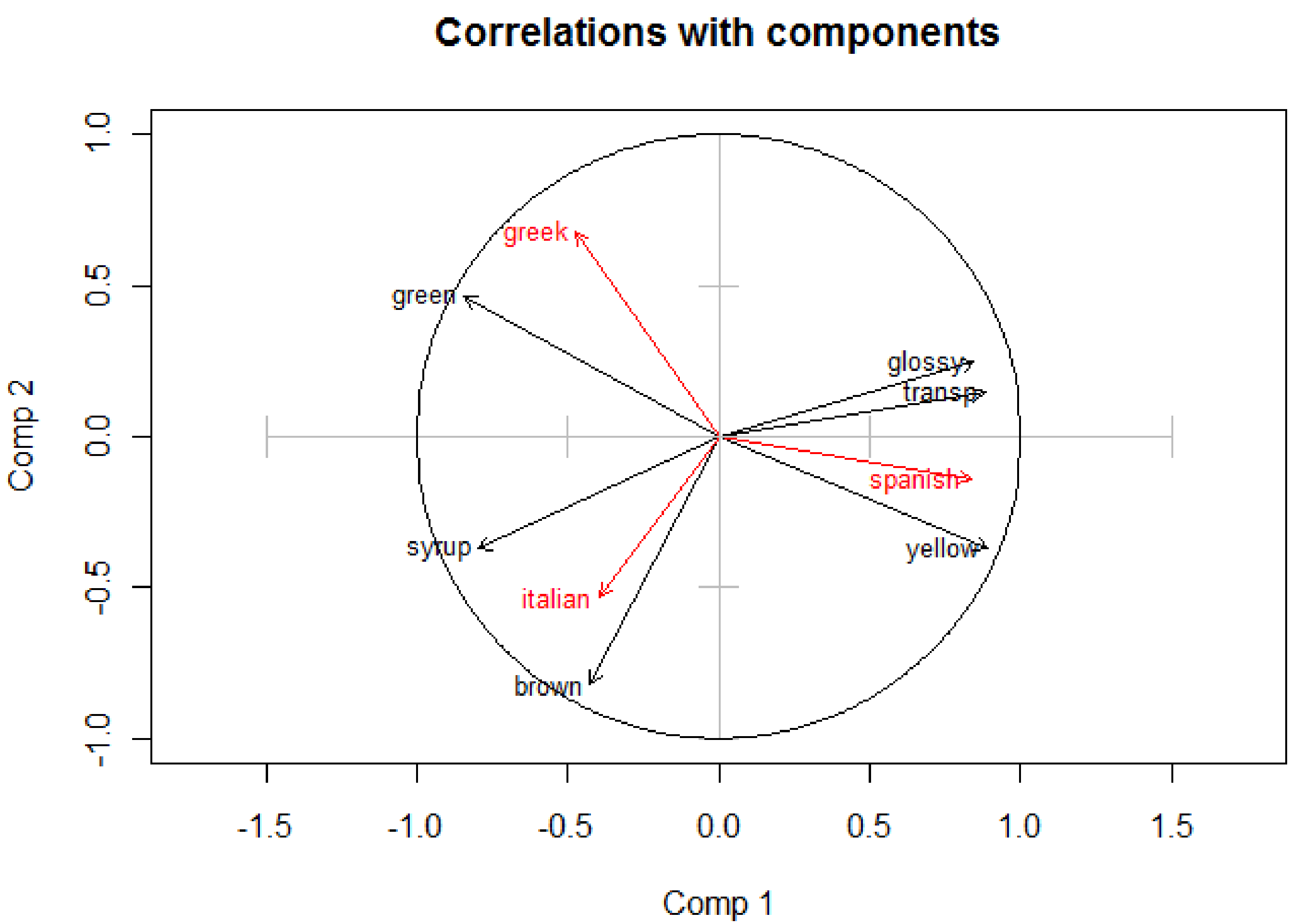


Coefficicients



	greek	italian	spanish
yellow	-0.367887251	0.08524330	0.27061098
green	0.390706944	-0.11546350	-0.26352554
brown	-0.405799598	0.37306163	0.03134422
glossy	0.000981105	-0.17504440	0.16665292
transp	-0.096961670	-0.09963388	0.18822591
syrup	-0.114878209	0.21750074	-0.09825359

Correlation plot of sensory variables

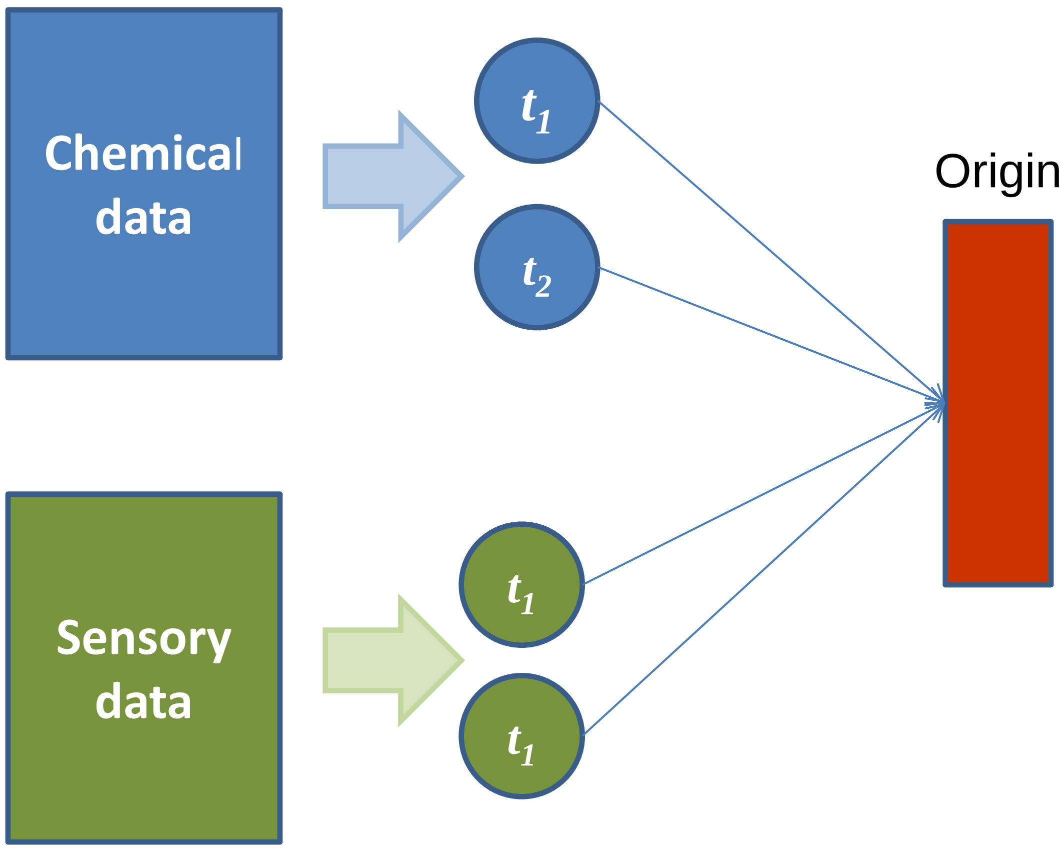


Prediction of origin from sensory data by LDA

```
> da <- lda(p2$scores[,1:2],y,CV=TRUE)
> table(da$class,y)
         y
          greek italian spanish
  greek       5       1       0
  italian     0       2       0
  spanish     0       2       6
```

```
         y da.class     greek  italian   spanish
1    greek    greek 9.937e-01 0.006323 6.121e-06
2    greek    greek 9.850e-01 0.015015 4.447e-06
3    greek    greek 9.589e-01 0.036270 4.803e-03
4    greek    greek 9.195e-01 0.079073 1.464e-03
5    greek    greek 6.346e-01 0.357021 8.389e-03
6  italian  italian 4.916e-02 0.950587 2.476e-04
7  italian  spanish 2.425e-01 0.350141 4.074e-01
8  italian    greek 9.981e-01 0.001890 1.644e-06
9  italian  spanish 1.380e-03 0.039685 9.589e-01
10 italian  italian 2.908e-08 0.759681 2.403e-01
11 spanish  spanish 1.547e-04 0.011562 9.883e-01
12 spanish  spanish 1.338e-04 0.002539 9.973e-01
13 spanish  spanish 5.573e-03 0.337999 6.564e-01
14 spanish  spanish 6.299e-04 0.094555 9.048e-01
15 spanish  spanish 9.237e-05 0.014403 9.855e-01
16 spanish  spanish 5.064e-05 0.012574 9.874e-01
```

The conceptual model

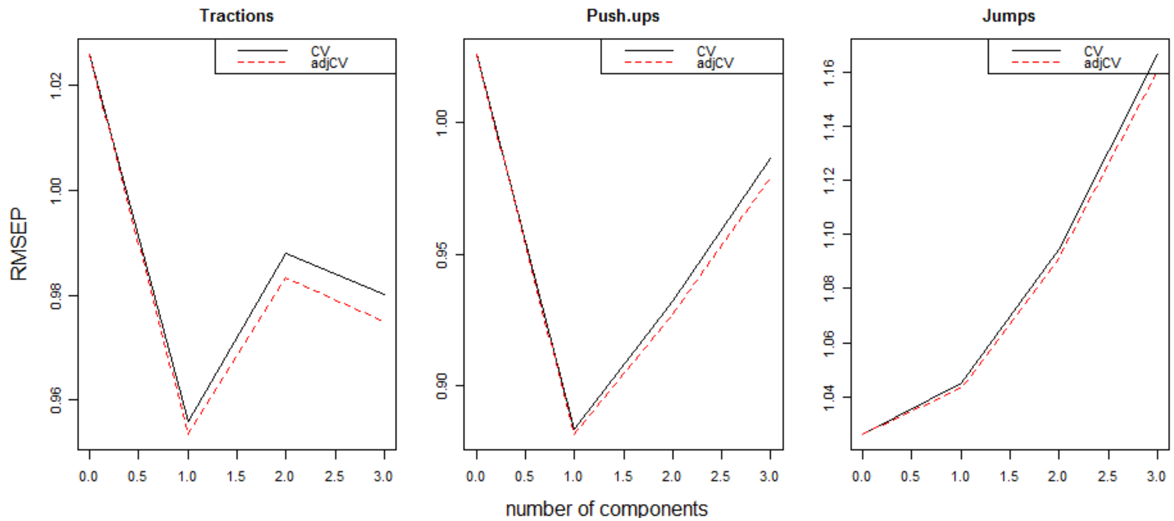


Prediction from PL2 chemical & sensory components by LDA

```
> da <- lda(cbind(chem,sens),y,CV=TRUE)
> table(da$class,y)
         y
          greek italian spanish
  greek       5       0       0
  italian     0       5       0
  spanish     0       0       6
```

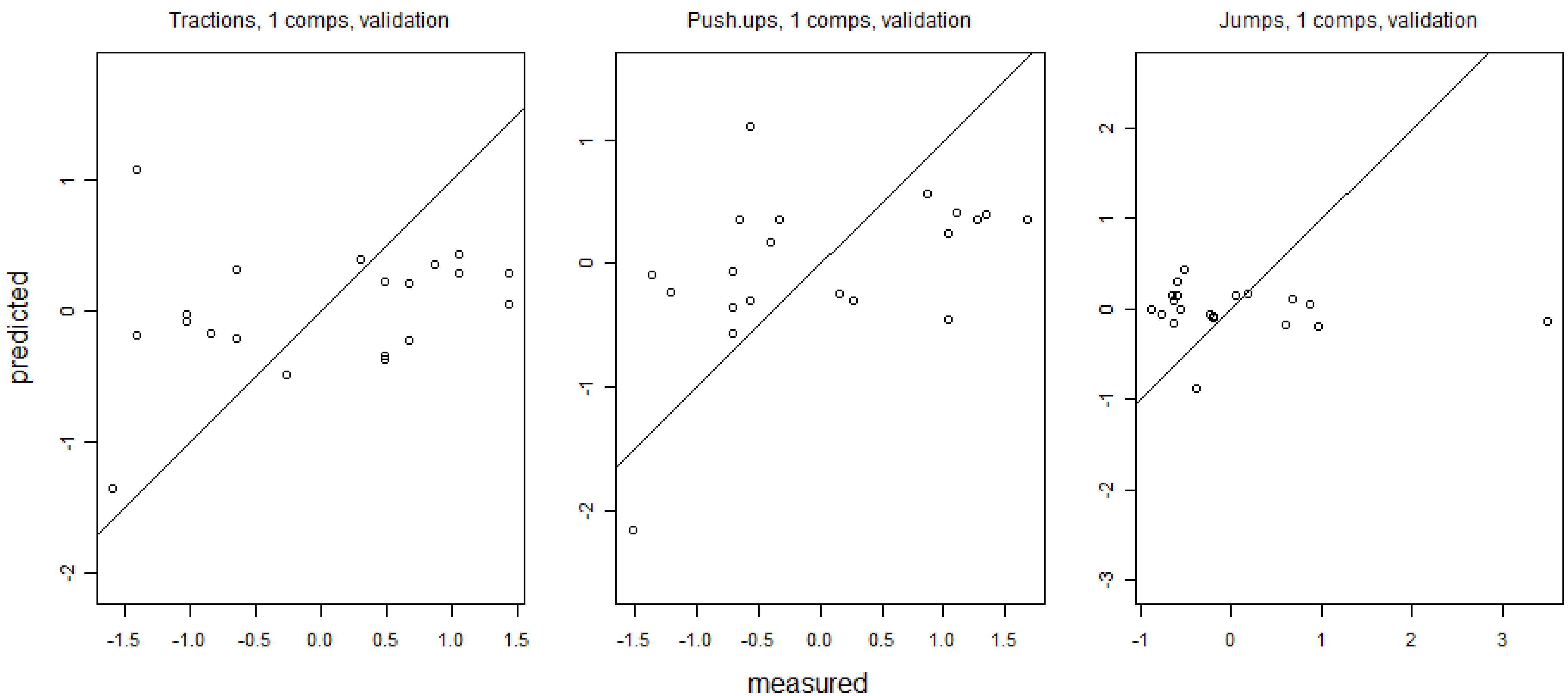
```
y da.class     greek   italian   spanish
1    greek    greek 1.000e+00 1.615e-19 1.409e-09
2    greek    greek 1.000e+00 1.096e-20 8.111e-07
3    greek    greek 1.000e+00 5.618e-21 1.668e-05
4    greek    greek 1.000e+00 1.408e-30 1.706e-12
5    greek    greek 1.000e+00 5.357e-42 1.153e-11
6  italian  italian 3.091e-23 1.000e+00 2.013e-10
7  italian  italian 1.866e-24 1.000e+00 3.901e-08
8  italian  italian 1.367e-19 1.000e+00 1.382e-06
9  italian  italian 2.634e-23 1.000e+00 4.576e-06
10 italian  italian 4.296e-72 1.000e+00 2.654e-21
11 spanish  spanish 1.385e-11 7.683e-08 1.000e+00
12 spanish  spanish 4.945e-13 4.155e-07 1.000e+00
13 spanish  spanish 6.951e-08 2.538e-08 1.000e+00
14 spanish  spanish 2.662e-10 3.966e-08 1.000e+00
15 spanish  spanish 9.251e-08 1.487e-14 1.000e+00
16 spanish  spanish 1.005e-07 3.826e-14 1.000e+00
```

Linnerud data: Selecting the number of components



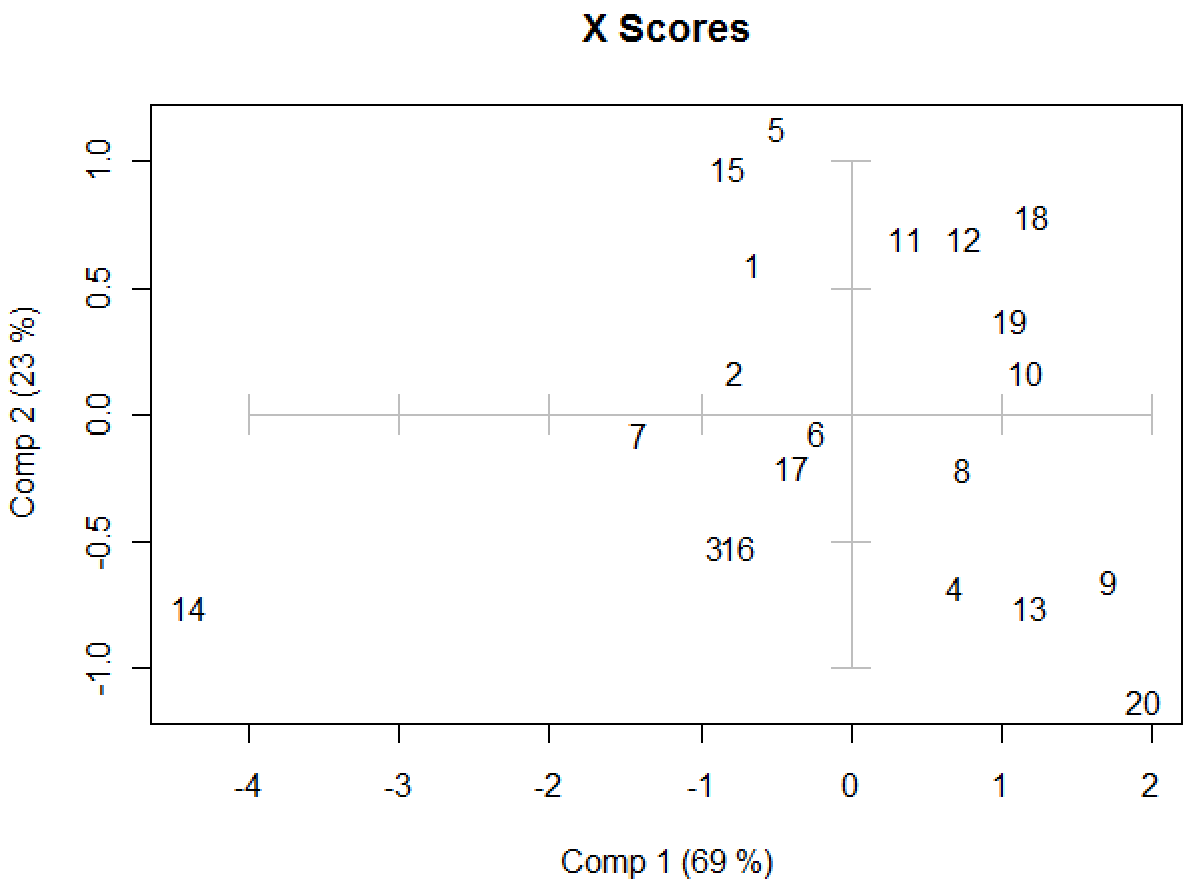

```
1 component
RMPRESS   Tractions    Push.ups     Jumps
          0.9395220    0.8659075    1.0457392
R2cv      Tractions    Push.ups     Jumps
          0.07084046   0.21074123   -0.15112677
2 components
RMPRESS   Tractions    Push.ups     Jumps
          0.9831243    0.9259299    1.0790209
R2cv      Tractions    Push.ups     Jumps
          -0.01740362  0.09753039   -0.22556429
3 components
RMPRESS   Tractions    Push.ups     Jumps
          0.9800705    0.9862713    1.1665417
R2cv      Tractions    Push.ups     Jumps
          -0.01109291  -0.02392737  -0.43244169
```

Prediction

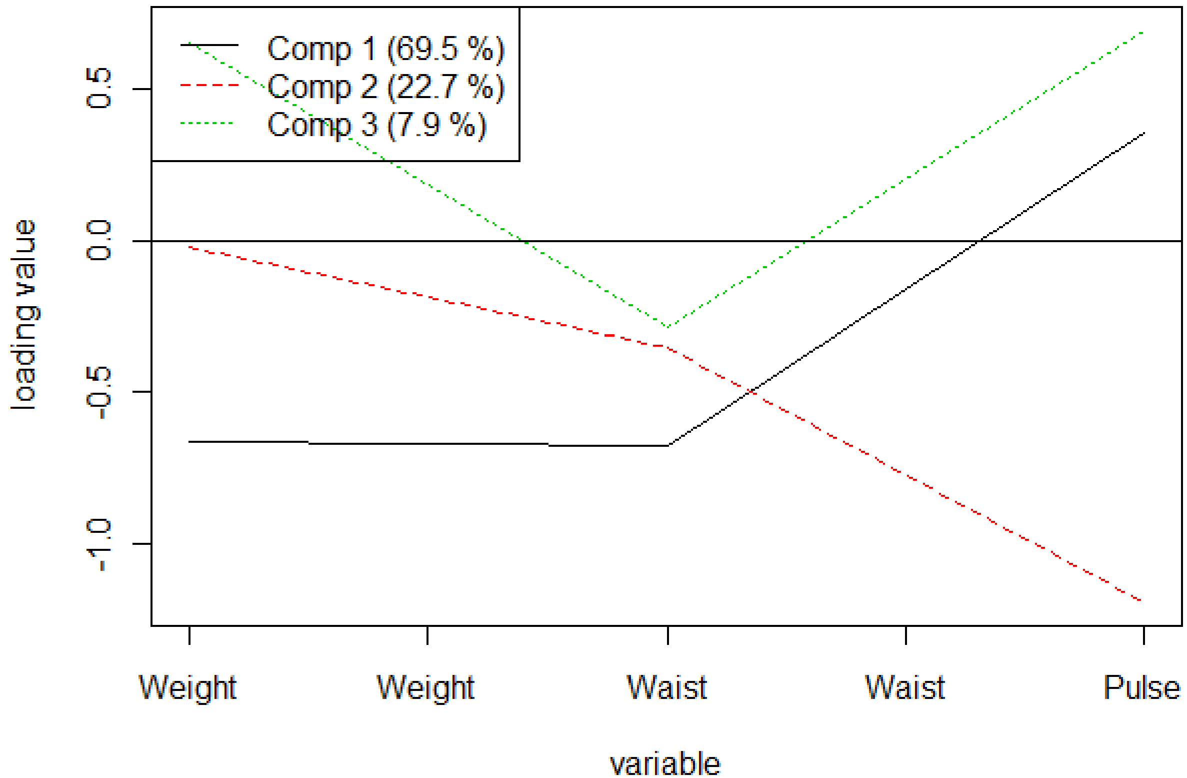


Tractions, 1 comps, validation
Push.ups, 1 comps, validation
Jumps, 1 comps, validation
predicted
measured
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
-2
-1
0
1
2
3
-3

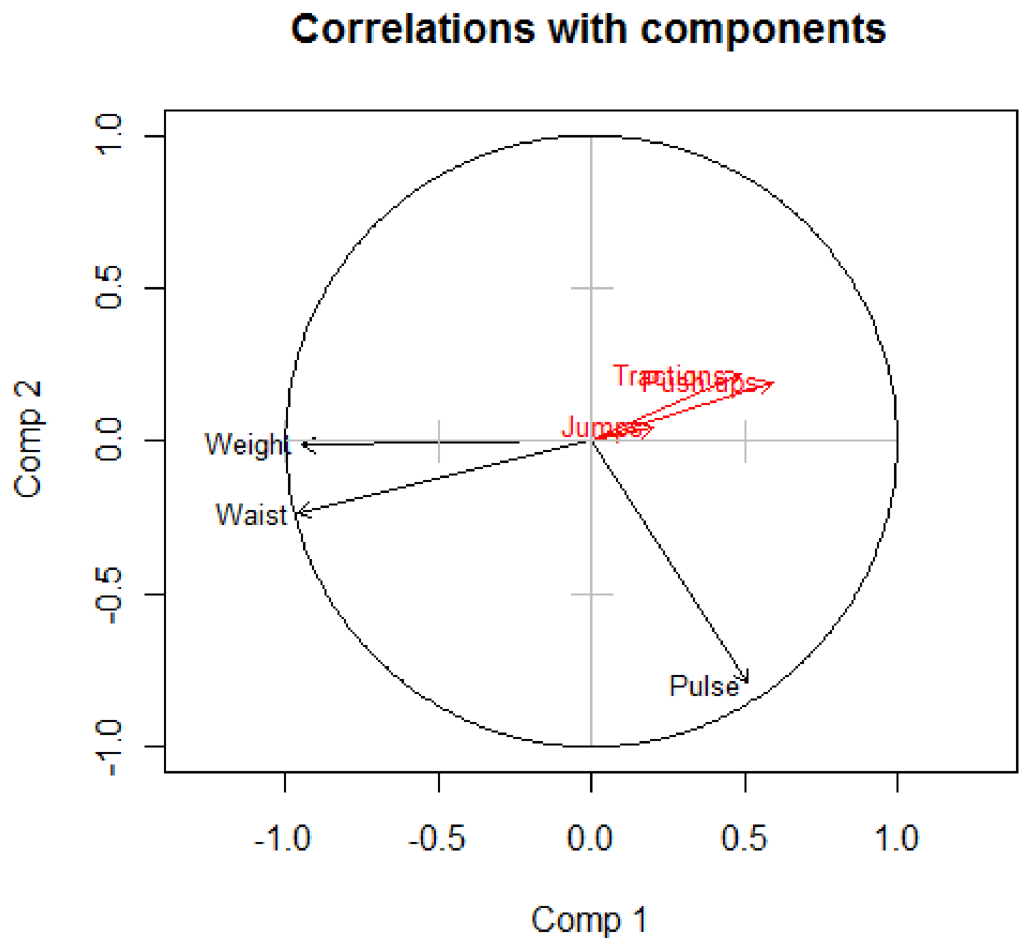
Scores plot


X Scores
Comp 2 (23 %)
Comp 1 (69 %)
1.0
0.5
0.0
-0.5
-1.0
-4
-3
-2
-1
0
1
2
5
15
1
11
12
18
19
2
10
7
6
17
8
316
4
13
9
14
20

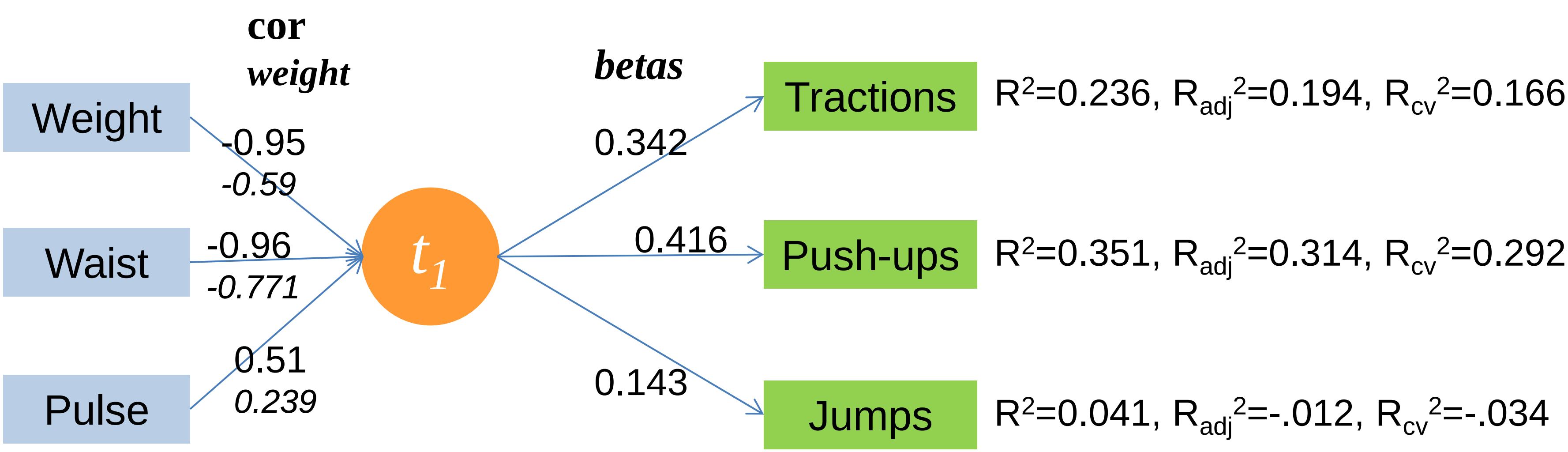
Loadings plot



Correlations plot



Results of PLS2 on Linnerud data



Com = 0.69 Red = 0.21