MIRI
Data Mining & Business Intelligence

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Session 1: Principal Component Regression

Sessions on Multivariate Modeling.

Course on Kernel Machine Learning and Multivariate Modeling

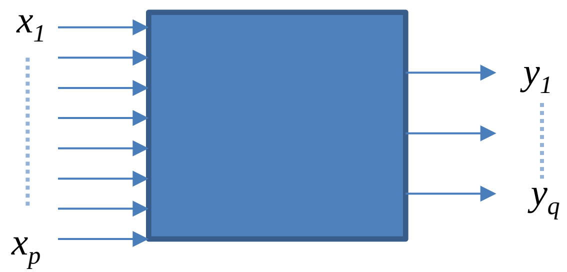
Tomàs Aluja-Banet

tomas.aluja@upc.edu

information matters
INFORMATION MODELING & PROCESSING

The problem

We want to predict an output $y_1, \ldots, y_q$, from an input $x_1, \ldots, x_p$



We are thinking on problems with high p relatively to n, (including when $p>>n$), whereas the q can be $q=1$ depending on the problem:

NIR data: we want to predict the chemical composition of certain elements from the Near Infra Red spectroscopy of elements. According the presence of atoms, spectroscopy will present peaks in certain frequencies.
Genome data: From the expression of genes we want to predict the presence of a disease.
Sensory data: From a set of sensory descriptors we want to predict the liking of a product.
Data Fusion: we want to impute a block of missing variables from the common variables.
…

Idea: instead of using the raw x_j predictors, extract their subjacent (hidden) latent constructs (ideally they refer to intangible concepts), and use them to predict the y_k responses

Program

1. Multivariate Regression and Principal Component Regression
2. Canonical Correlation Analysis, Inter Batteries Analysis and Redundancy Analysis
3. NIPALS algorithm
4. Partial Least Squares Regression 1
5. Partial Least Squares Regression 2

Herman Wold, 1908 - 1992

Svante Wold

Continuing seminar on Partial Least Squares Path Modelling (end of January)

Gaston Sánchez dedicated webpages:
http://www.plsmodeling.com/
http://gastonsanchez.com/software/

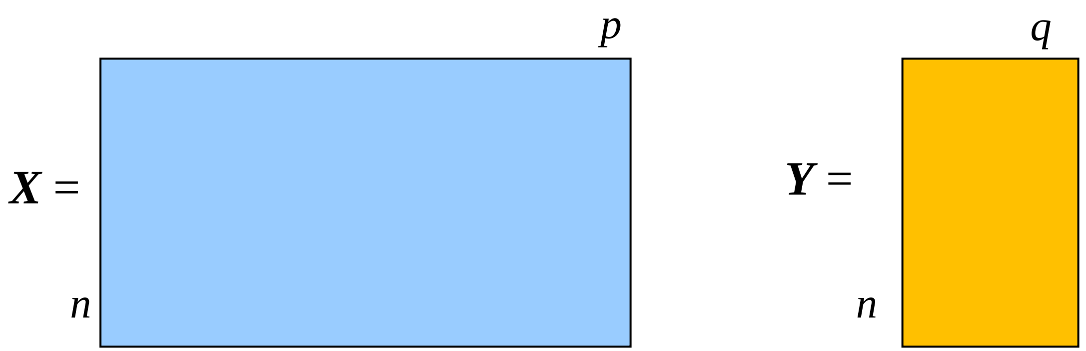
The problem

We have two vectors of variables measured on n individuals: $(x_1, x_2, \cdots, x_p)$ $(y_1, y_2, \cdots, y_q)$

The goal is to measure the relationship between both multivariate vectors ***(multivariate calibration)***

We treat group x as the explanatory and group y is the response one.

$$(y_1, \ldots, y_q) = f(x_1, \ldots, x_p) + (\varepsilon_1, \ldots, \varepsilon_q)$$



$$\mathbf{y} = (y_1, \ldots, y_q) \qquad \mathbf{x} = (x_1, \ldots, x_p) \qquad \boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_q)$$

$$\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\varepsilon}$$

The Multivariate Regression approach

The multivariate regression approach:

$$\boxed{\vec{y}_k = \beta_{1k}\vec{x}_1 + \cdots + \beta_{pk}\vec{x}_p + \vec{\varepsilon}_k \quad k = 1,\ldots,q}$$

We model each response by a linear combination of the $\mathbf{x}$ vector plus a random fluctuation (*we consider centered vectors*)

$$\begin{array}{c} \begin{matrix} \phantom{y_{11}} & \vec{y}_k & \phantom{y_{1q}} \end{matrix} \\ \begin{bmatrix} y_{11} & y_{1k} & y_{1q} \\ \vdots & \vdots & \vdots \\ y_{n1} & y_{nk} & y_{nq} \end{bmatrix} \end{array} = \begin{array}{c} \begin{matrix} \vec{x}_1 & \phantom{x_{12}} & & \vec{x}_p \end{matrix} \\ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \end{array} \begin{bmatrix} \beta_{11} & \beta_{1k} & \beta_{1q} \\ \beta_{21} & \beta_{2k} & \beta_{2q} \\ \vdots & \vdots & \vdots \\ \beta_{p1} & \beta_{pk} & \beta_{pq} \end{bmatrix} + \begin{array}{c} \begin{matrix} \phantom{\varepsilon_{11}} & \vec{\varepsilon}_k & \phantom{\varepsilon_{1q}} \end{matrix} \\ \begin{bmatrix} \varepsilon_{11} & \varepsilon_{1k} & \varepsilon_{1q} \\ \vdots & \vdots & \vdots \\ \varepsilon_{n1} & \varepsilon_{nk} & \varepsilon_{nq} \end{bmatrix} \end{array}$$

$$\underset{(n,q)}{Y} = \underset{(n,p)}{X}\underset{(p,q)}{\mathrm{B}} + \underset{(n,q)}{\mathrm{E}}$$

Y matrix of q response variables (centered)

X matrix of p explanatory variables (centered)

B *is the matrix of $p*q$ β_{jk} parameters*

E random fluctuation matrix

The Multivariate Regression approach

Assumptions

$E[\boldsymbol{\varepsilon}]=0, \quad V[\boldsymbol{\varepsilon}]=\Sigma$

$$E(\varepsilon_k)=0$$
$$\mathrm{var}(\varepsilon_k)=\sigma_k^2$$
$$\mathrm{cov}(\varepsilon_k,\varepsilon_l)=\sigma_{kl}$$

$$E[\boldsymbol{\varepsilon}]=\begin{bmatrix}0\\ \vdots\\ 0\end{bmatrix}, \quad \Sigma=\begin{bmatrix}\sigma_1^2 & \sigma_{1k} & \sigma_{1q}\\ \sigma_{1k} & \sigma_k^2 & \sigma_{kq}\\ \sigma_{1q} & \sigma_{kq} & \sigma_q^2\end{bmatrix}$$

$V(\mathbf{x},\boldsymbol{\varepsilon})=0$

$$\mathrm{cov}(x_j,\varepsilon_k)=0$$

observations are *iid*, observations from diferent individuals are uncorrelated

$$\mathrm{cov}(y_k,y_l)=\sigma_{kl}\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix}$$

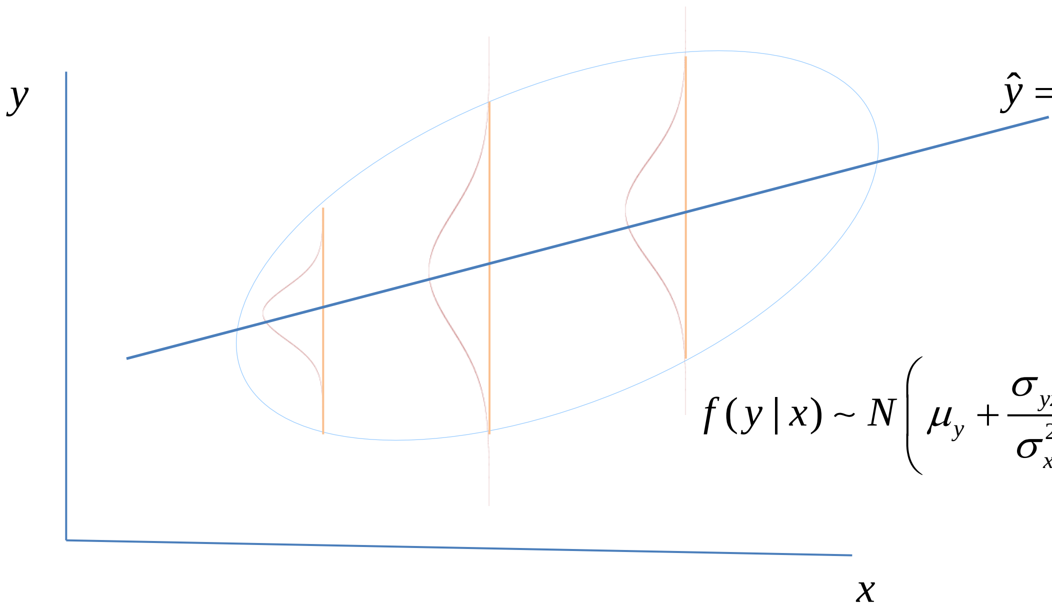
$\boldsymbol{\varepsilon}\sim N_q(0,\Sigma)$ $\longrightarrow$ $E(\mathbf{y}|\mathbf{x})=\mathbf{xB}$ $\quad$ $\mathbf{y}|\mathbf{x}\sim N_q(\mathbf{xB},\Sigma)$

$$\hat{Y}=E[\mathbf{y}|\mathbf{x}]$$

Regression under multinormal distribution

univariate case
with joint normal distribution

$$N\left(\begin{bmatrix}\mu_y\\ \mu_x\end{bmatrix},\begin{bmatrix}\sigma_y^2 & \sigma_{yx}\\ \sigma_{yx} & \sigma_x^2\end{bmatrix}\right)$$



$$\hat{y}=E\left[y\middle|x\right]=\mu_y+\frac{\sigma_{yx}}{\sigma_x^2}(x-\mu_x)$$

$$f(y\,|\,x)\sim N\left(\mu_y+\frac{\sigma_{yx}}{\sigma_x^2}(x-\mu_x),\ \sigma_y^2-\frac{\sigma_{yx}^2}{\sigma_x^2}\right)$$

The Multivariate Regression fit

The multivariate regression fit:

$$y_k = b_{1k}x_1 + \cdots + b_{pk}x_p + e_k = \hat{y}_k + e_k \quad k = 1,\ldots,q$$

$\hat{y}_k$ is the fit
e_k is the residual
b_{jk} is the estimator of β_{jk}

We model each response by a linear combination of the $\mathbf{x}$ vector plus a residual (*centered vectors*)

$$\begin{array}{ccc} & \vec{y}_k & \\ \end{array} \qquad \vec{x}_1 \quad \cdots \quad \vec{x}_p \qquad b_k \qquad \vec{e}_k$$

$$\begin{bmatrix} y_{11} & y_{1k} & y_{1q} \\ \vdots & \vdots & \vdots \\ y_{n1} & y_{nk} & y_{nq} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} b_{11} & b_{1k} & b_{1q} \\ b_{21} & b_{2k} & b_{2q} \\ \vdots & \vdots & \vdots \\ b_{p1} & b_{pk} & b_{pq} \end{bmatrix} + \begin{bmatrix} e_{11} & e_{1k} & e_{1q} \\ \vdots & \vdots & \vdots \\ e_{n1} & e_{nk} & e_{nq} \end{bmatrix}$$

$$\vec{y}_k = Xb_k + e_k \quad k = 1,\ldots,q$$

$$\underset{(n,q)}{Y} = \underset{(n,p)}{X}\underset{(p,q)}{B} + \underset{(n,q)}{E}$$

Y matrix of q response variables (centered)

X matrix of p explanatory variables (centered)

B *is the matrix of* $p*q$ b_{jk} *coefficients*

E residual matrix

The Ordinary Least Squares Fit

$\vec{y}_k = X\vec{\beta}_k + \vec{\varepsilon}_k$

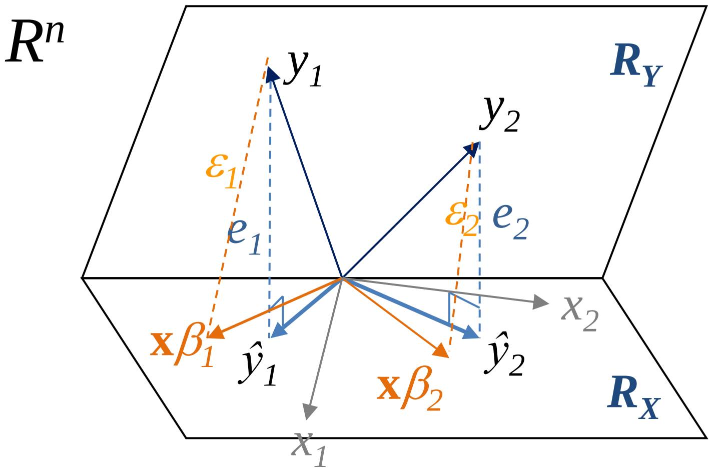
$$Y = XB + E = \hat{Y} + E$$

OLS fitting: $Min\|E\|^2 = \sum_{k=1}^{q}\sum_{i=1}^{n} e_{ik}^2 = \sum_{k=1}^{q}\|e_k\|^2$

Geometrical approach:
Orthogonal projection on $\boldsymbol{R_X}$

N metric of R^n

$\left(N = \frac{1}{n}I\right)$



$$X'NE = 0 \qquad E = (e_1, \ldots, e_q)$$

$\hat{y}_k$ orthogonal projection of y_k upon R_X

$$E = Y - XB \qquad X'NY = X'NXB$$

$$\boxed{B = (X'NX)^{-1}X'NY}$$

$B=(b_1,.., b_q)$ vector of ***OLS*** coefficients

The OLS fitting coincides with the Maximum Likelihood estimators If multivariate

$$\hat{Y} = XB \qquad \hat{y}_j = Xb_j$$

$$\hat{Y} = X(X'NX)^{-1}X'NY \qquad \hat{y}_j = X(X'NX)^{-1}X'Ny_j$$

$$\hat{Y} = HY$$

For every y_j we obtain the classical OLS equations, but now residuals between variables are correlated. $\mathrm{cov}(e_j, e_k) \neq 0$

Residuals and their covariance

Covariance matrix of Residuals

$$\underset{(n,q)}{E} = Y - XB$$

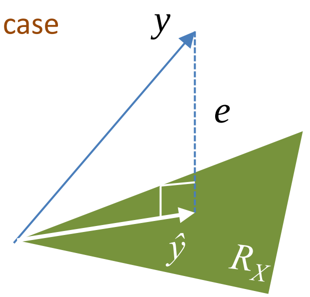
Matrix of residuals, it reflects the variables y_k after having eliminated the (linear) effect of x_j variables

$$\underset{(q,q)}{\hat{\Sigma}} = E'NE = Y'NY - Y'NX(X'NX)^{-1}X'NY) = V_{YY} - V_{YX}V_{XX}^{-1}V_{XY}$$

Covariance matrix of residuals =
Matrix of **partial covariances of Y given X** =
Variability of Y controlling for X

If the linear model is correct E must be *white noise*

MANOVA table



Decomposition of the Sums of Squares

Sum of squares **matrix** decomposition

$$\underset{(q,q)}{Y'Y} = \underset{(q,q)}{\hat{Y}'\hat{Y}} + \underset{(q,q)}{E'E}$$

due to orthogonality between X and E, Y centered,

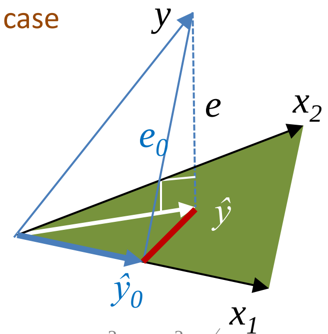
MANOVA table	SS matrix	Degrees of freedom
Explained by the regression	$H_p = \hat{Y}'\hat{Y}$	p
Residual	$E'E$	$n\text{-}p\text{-}1$
Total	$Y'Y$	$n\text{-}1$

Gain due to the last (p-r) regressors

H_0: If we only have the first *r* regressors of *X* $\underset{(n,q)}{Y} = \underset{(n,r)}{X_r}\underset{(r,q)}{B_r} + \underset{(n,q)}{E_0}$ $Y'Y = \hat{Y}_0'\hat{Y}_0 + E_0'E_0$ $\hat{Y}_0 = X_r B_r$

H_1: complete model with *p* regressors $Y = X_p B_p + E$ $Y'Y = \hat{Y}'\hat{Y} + E'E$

univariate case



$$F = \frac{\left(|e_0|^2 - |e|^2\right) / (p-r)}{|e|^2 / (n-p-1)}$$

MANOVA table	SS matrix	Degrees of freedom
Explained with *p* regressors	$H_p = \hat{Y}'\hat{Y}$	*p*
Gain to *p-r* regressors	$H_{(p-r)\mid r} = \hat{Y}'\hat{Y} - \hat{Y}_0'\hat{Y}_0$	*(p-r)*
Explained with *r* regressors	$H_r = \hat{Y}_0'\hat{Y}_0$	*r*
Residual	$E'E$	*n-p-1*
Total	$Y'Y$	*n-1*

Testing the influence of last *(p-r)* regressors (whether superfluous)

Likelihood Ratio Test $\max L = (2\pi)^{-\frac{1}{2}np} \,|\hat{\Sigma}|^{-\frac{1}{2}np} \exp(-\tfrac{1}{2}np)$

Wilks Lambda: $\Lambda = \dfrac{|E'E|}{|E_0'E_0|} = \dfrac{|E'E|}{\left|H_{(p-r)\mid r} + E'E\right|} = \prod_{k=1}^{q}(1+\theta_k)^{-1}$

θ_r eigenvalues of $(E'E)^{-1}H_{(p-r)\mid r}$

$$-\left((n-p-1) - \frac{1}{2}(q-(p-r)+1)\right)\ln\Lambda \sim \chi^2_{q(p-r)}$$

Pillai statistic

$$tr\left(H_{(p-r)\mid r}\left(E'E + H_{(p-r)\mid r}\right)^{-1}\right)$$

Other approximations with the F distribution are possible

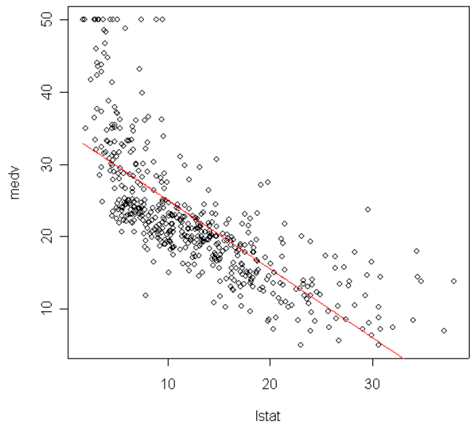
Predictions

Prediction of x_0 $x_0 = (x_{01}, \ldots, x_{0p})$ centered respect to X

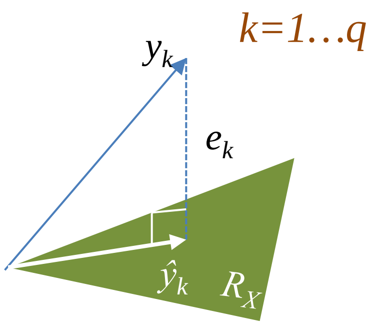
$$\hat{y}_0 = B'x_0 \sim N_q(B'x_0, x_0'(X'X)^{-1}x_0\Sigma)$$

Prediction of the $\hat{y}_{0k} = E[y_k|x_0]$ $\quad b_k'x_0 \sim N_q(\beta_k'x_0, \sigma_k^2 x_0'(X'X)^{-1}x_0)$

Prediction of the y_{0k} $\quad b_k'x_0 \sim N_q(\beta_k'x_0, \sigma_k^2(1 + x_0'(X'X)^{-1}x_0))$



Accuracy of predictions



Global measure

$$R_k^2 = 1 - \frac{|e_k' e_k|}{|y_k' y_k|} \qquad mean\left(R_k^2\right)$$

$$R_{adj.k}^2 = 1 - \frac{|e_k' e_k| \big/ n-p-1}{|y_k' y_k| \big/ n-1} \qquad mean\left(R_{adj.k}^2\right)$$

LOO prediction

$$PRESS_k = \sum_{i=1}^{n} (y_{ik} - \hat{y}_{(-ik)})^2 = \sum_{i=1}^{n} \left(e_{ik} \big/ (1-h_{ii}) \right)^2$$

$$R_{cv.k}^2 = 1 - \frac{PRESS_k}{|y_k' y_k|} \qquad mean\left(R_{cv.k}^2\right)$$

Prediction accuracy of least squares estimates

- Provided the true relationship is linear, if $n>>p$ least squares estimates will have low variance and perform well in test samples. But, if n approaches p, $var(b_j)$ increases yielding poor predictions, and if $n<p$, $var(b_j)$ is infinite. Moreover, in case $p\uparrow\uparrow$ (big data), most of the predictors are redundant (collinear) increasing the $var(b_j)$ as well.

- Solutions:
  - Subset selection. Selecting only the significant predictors (gain in model interpretability)
  - Shrinkage. Constraining the values of the estimates (Ridge, LASSO)
  - Dimensionality reduction. This is the goal of this course.

The linnerud problem

Prediction of performance from physical measurements of 20 athlets of a Gymnasium

We form the vector
x = (Weight, Waist, Pulse)

to predict the vector
y = (Tractions, Push-ups, Jumps)

Editor de datos

	Weight	Waist	Pulse	Tractions	Push.ups	Jumps
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
7	211	38	56	8	101	38
8	167	34	60	6	125	40
9	176	31	74	15	200	40
10	154	33	56	17	251	250
11	169	34	50	17	120	38
12	166	33	52	13	210	115
13	154	34	64	14	215	105
14	247	46	50	1	50	50
15	193	36	46	6	70	31
16	202	37	62	12	210	120
17	176	37	54	4	60	25
18	157	32	52	11	230	80
19	156	33	54	15	225	73
20	138	33	68	2	110	43

Multivariate Regression with *lm* function

```
# X = as.matrix[Weight, Waist, Pulse]
# Y = as.matrix[Tractions, Push.ups, Jumps)

> mreg <- lm(Ys ~ Xs-1)

Coefficients:
             Tractions  Push.ups   Jumps
XsWeight       0.36825    0.28715  -0.25899
XsWaist       -0.88182   -0.88983   0.01460
XsPulse       -0.02585    0.01606  -0.05464

> summary(manova(mreg))
          Df  Pillai approx F num Df den Df Pr(>F)
Xs         3 0.67848   1.6561      9     51 0.1245
Residuals 17
```

$= B$

Multivariate Regression with *lm*

```
> summary(mreg)
Response Tractions :

Call:
lm(formula = Tractions ~ Xs - 1)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.42207 -0.76658  0.07098  0.66034  1.16398 

Coefficients:
         Estimate Std. Error t value Pr(>|t|)  
XsWeight  0.36825    0.40339   0.913   0.3741  
XsWaist  -0.88182    0.40125  -2.198   0.0421 *
XsPulse  -0.02585    0.21238  -0.122   0.9046  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8591 on 17 degrees of freedom
Multiple R-squared:  0.3396,    Adjusted R-squared:  0.223 
F-statistic: 2.914 on 3 and 17 DF,  p-value: 0.0644
```

MVR is like a list of univariate regressions, but taking into account the existing correlation among the y_k variables for hypothesis testing and confidence intervals.

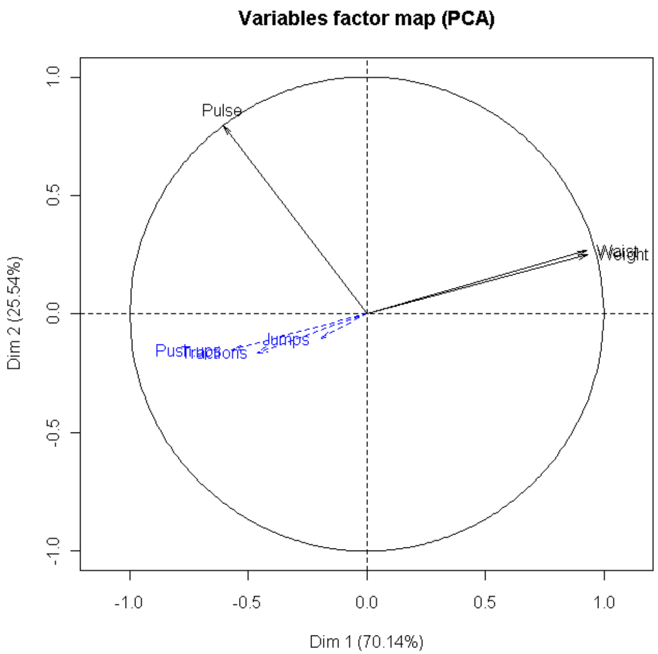
```
Response Push.ups :
Call:
lm(formula = Push.ups ~ Xs - 1)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.1858 -0.5863 -0.1983  0.6438  1.3048 

Coefficients:
         Estimate Std. Error t value Pr(>|t|)  
XsWeight  0.28715    0.37261   0.771   0.4515  
XsWaist  -0.88983    0.37064  -2.401   0.0281 *
XsPulse   0.01606    0.19618   0.082   0.9357  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7936 on 17 degrees of freedom
Multiple R-squared:  0.4365,    Adjusted R-squared:  0.337 
F-statistic: 4.389 on 3 and 17 DF,  p-value: 0.01842


Response Jumps :
Call:
lm(formula = Jumps ~ Xs - 1)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.9339 -0.6811 -0.1974  0.2915  3.2566 

Coefficients:
         Estimate Std. Error t value Pr(>|t|)
XsWeight -0.25899    0.48281  -0.536    0.599
XsWaist   0.01460    0.48025   0.030    0.976
XsPulse  -0.05464    0.25420  -0.215    0.832

Residual standard error: 1.028 on 17 degrees of freedom
Multiple R-squared:  0.0539,    Adjusted R-squared:  -0.1131 
F-statistic: 0.3229 on 3 and 17 DF,  p-value: 0.8088
```

but …

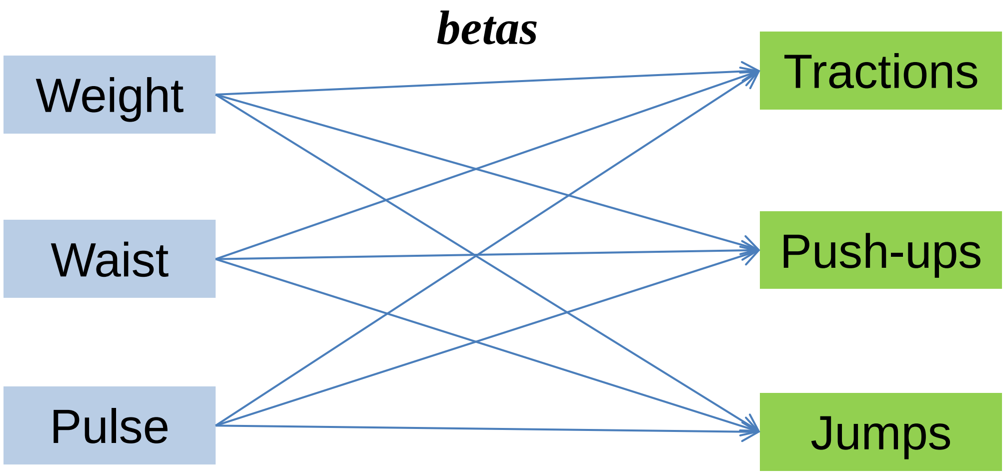
There is a collinearity problem between *Waist* and *Weight*, which makes no sense the coefficients of both variables (to be interpretable, coefficients should have the same sign as the correlation between variables)

```
cor(X,Y)
       Tractions Push.ups   Jumps
Weight    -0.390   -0.493 -0.2263
Waist     -0.552   -0.646 -0.1915
Pulse      0.151    0.225  0.0349
```

```
Coefficients:
           Tractions Push.ups   Jumps
XsWeight       0.368    0.287  -0.259
XsWaist       -0.882   -0.890   0.015
XsPulse       -0.026    0.016  -0.055
```



Summary of results of PCR on Linnerud case



Tractions: R^2=0.3396, R_{adj}^2=0.223, R_{cv}^2= 0.1091

Push-ups: R^2=0.4365, R_{adj}^2=0.337, R_{cv}^2=0.1236

Jumps: R^2=0.0539, R_{adj}^2=-0.1131, R_{cv}^2= -0.2502

Average R^2=0.2766 Average R_{cv}^2=-0.00581

```
# R2 by LOO

PRESS  <- colSums((mreg$residuals/(1-ls.diag(mreg)$hat))^2)
R2cv   <- 1 - PRESS/(diag(var(Ys))*(n-1))
```

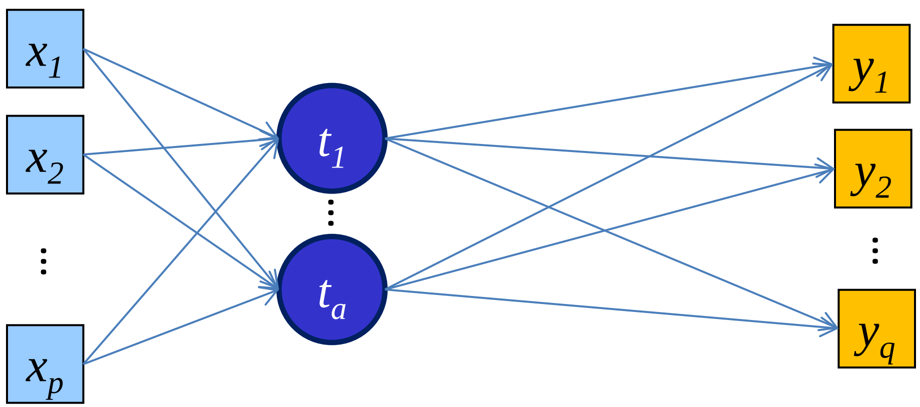
limitations of mvr

- Observed variables may be collinear
- In multivariate contexts, observed variables are indicators of hidden latent concepts.
- Observed variables contain random fluctuation
- Number of observations may be close or lower than the number of variables (case of NIR data, genomic data, …)

→ ***The PCR solution***

Using PCA to extract the hidden latent variables of the X predictors

The PCR model



t_h is a common factor for the X group

We measure the relation of Y on X through the common factors t_h

How many common factors do we need?

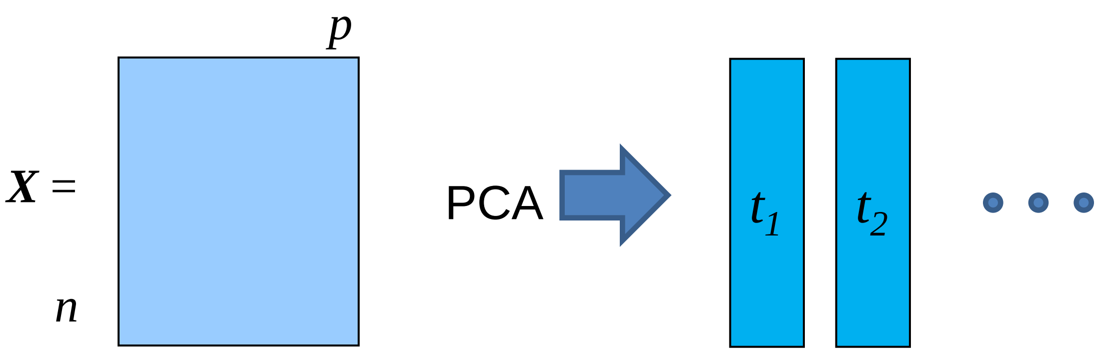
Advantages:

PCR allows to deal with

- multicollinearity of *X*
- more variables than observations
- smoothing of data

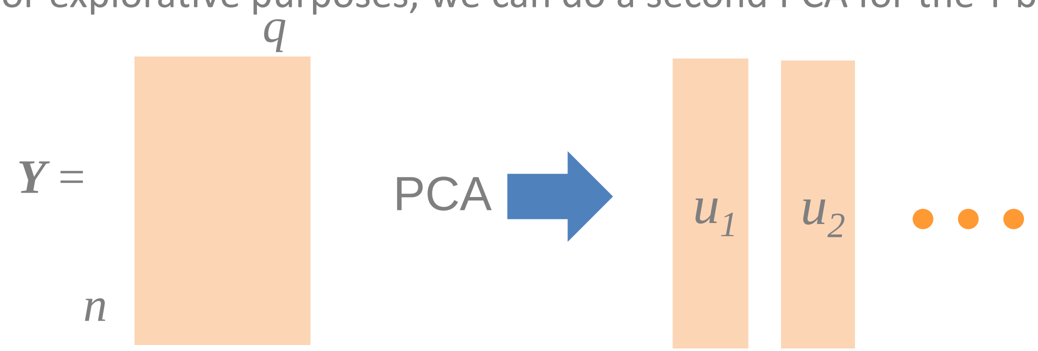
Naif solution: a PCA of the X block

Data has to be centered and we will suppose to be standardized as well, for the sake of comparison with the majority of the sequel methods



$$t_h = Xa_h$$

For explorative purposes, we can do a second PCA for the Y block



$$u_h = Yb_h$$

Each principal component is representative of its own group

review of Principal Component Analysis

First, we perform the PCA of the predictor block X : $\begin{cases} in\ R^p \rightarrow X'NXP = P\Lambda & R^p \text{ eigenvectors } P'P = 1 \\ in\ R^n \rightarrow N^{1/2}XX'N^{1/2}U = U\Lambda & R^n \text{ eigenvectors } U'U = 1 \end{cases}$

SVD equivalent: $N^{1/2}X = U\Lambda^{1/2}P'$

weight matrix:

$$N = \begin{bmatrix} \ddots & & \\ & n_i & \\ & & \ddots \end{bmatrix} \qquad \sum_{i=1}^{n} n_i = 1 \qquad n_i = \frac{1}{n}$$

Transition Relationships: $P = X'N^{1/2}U\Lambda^{-1/2}$, $U = N^{1/2}XP\Lambda^{-1/2}$

standardized

$T^{stan} = T\Lambda^{-1/2}$

$T^{stan\prime}NT^{stan} = I$

$G^{stan} = G\Lambda^{-1/2}$

$G^{stan\prime}G^{stan} = I$

Principal Components (scores): $T = [t_1, t_2, \ldots, t_p]$ $\quad T'NT = \Lambda$

$G^{stan} = P$

$$\boxed{T = XP} = (N^{-1/2}N^{1/2})XX'N^{1/2}U\Lambda^{-1/2} = XG^{stan} \boxed{= N^{-1/2}U\Lambda^{1/2}}$$

Variable projections (loadings): $G = [g_1, g_2, \ldots, g_p]$ $\quad G'G = \Lambda$

$= Cor(X,T)$

$$\boxed{G = X'N^{1/2}U} = X'NXP\Lambda^{-1/2} = X'NT^{stan} \boxed{= P\Lambda^{1/2}}$$

$G = RP\Lambda^{-1/2}$ $\quad \boxed{T = XR^{-1}G\Lambda^{1/2}}$

scores and loadings relation

Biplot: $X = ((N^{-1/2}U)(\Lambda^{1/2})(P')) = TP' = TG^{stan\prime} = T^{stan}G'$

in R^p in R^n

Lets suppose we have r significant components: $\hat{X}_{(r)} = T_{(r)}P'_{(r)}$

PCA model: $X = \hat{X}_{(r)} + \varepsilon_X = T_{(r)}P'_{(r)} + \varepsilon_X$

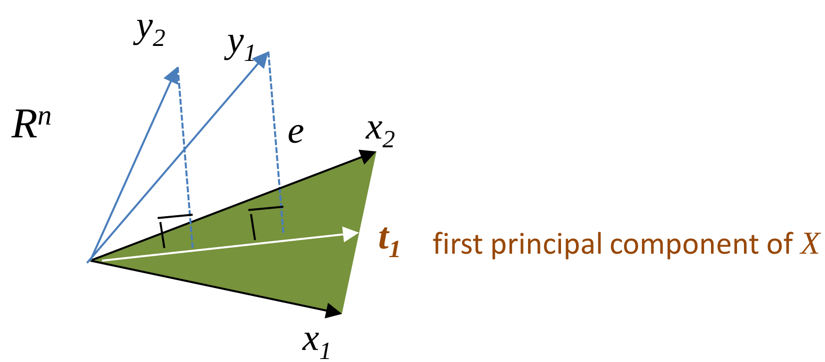
Principal Component Regression

PCR means to regress the Y variables on the r significant components of X

$$Y = T_{(r)}\tilde{B}_{(r)} + \varepsilon_Y \qquad \tilde{B}_{(r)} = (T'_{(r)}NT_{(r)})^{-1}T'_{(r)}NY = \Lambda_{(r)}{}^{-1}T'_{(r)}NY$$

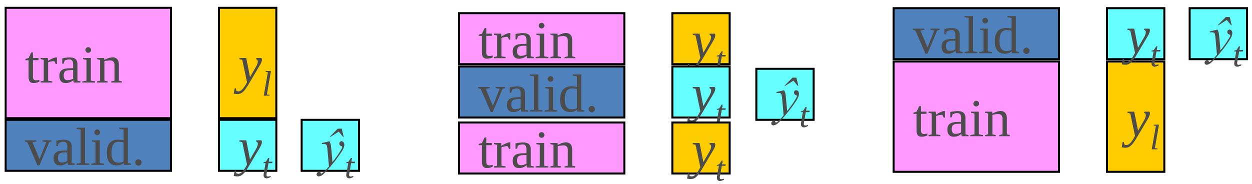
But at the end, we want to express the regression in terms of the original variables

$$Y = XP_{(r)}\tilde{B}_{(r)} + \varepsilon_Y = XB + \varepsilon_Y \qquad \boxed{B = P_{(r)}\Lambda_{(r)}{}^{-1}T'_{(r)}NY}$$



Selecting the number of components

We select the number of components by crossvalidation:



For each response variable

$$PRESS_j = \sum_{i=1}^{n} (y_{ij} - \hat{y}_{(-ij)})^2$$

CV prediction

$$MSEP_j = \frac{1}{n} PRESS_j$$

$$R^2_{cv.j} = 1 - \frac{PRESS_j}{\sum_i (y_{ij} - \bar{y}_j)^2}$$

$$PRESS_j = \sum_{i=1}^{n} \left(e_{ij} \big/ (1 - h_{ii}) \right)^2$$

diagonal element of the *hat* matrix

If LOO (leave one out crossvalidation), computing the PRESS is straitghtforward

```
> n      <- nrow(Yvar)
> PRESS  <- colSums((mymodel$residuals/(1-ls.diag(mymodel)$hat))^2)
> R2cv   <- 1 - PRESS/(diag(var(Yvar))*(n-1))
> R2cv
```

We select the components till (on average) the R^2_{cv} doesn't increase significantly

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Interpreting the results

Loadings (Coefficients of comp.)

(biplots)

$$\hat{X}_{(r)} = T_{(r)}P'_{(r)} = T^{stan}_{(r)}G'_{(r)}$$

Loadings
OLS coefficients of X respect to the components

$$\Rightarrow P = \left(p_1, p_2, \ldots, p_p\right)$$

$$\Rightarrow G = \begin{pmatrix} \vdots \\ cor(x_j, t_h) \\ \vdots \end{pmatrix}$$

(components can be standardized or not)

Yloadings

If regr. coeff of Y on $T_{(r)}$ = *Yloadings*

$$Yloadings = \begin{pmatrix} \vdots \\ cor(y_k, t_h) \\ \vdots \end{pmatrix}$$

R^2:

Communality part of explained variance of a variable(s) by a (group of) t_h factors respect to its own block

$$R^2(x_j; t_1, \cdots, t_s) = \sum_{h=1}^{s} cor^2(x_j, t_h) \qquad R^2(X; t_h) = \frac{1}{p}\sum_{j=1}^{p} cor^2(x_j, t_h)$$

Redundancy part of the variance of y_k variables explained by the t_h components

$$Rd^2(y_k; t_1, \cdots, t_s) = \sum_{h=1}^{s} cor^2(y_k, t_h) \qquad Rd^2(Y; t_h) = \frac{1}{q}\sum_{j=1}^{q} cor^2(y_k, t_h)$$

Graphical displays of variables

***X* and *Y* variables**: on t_1, t_2, ... basis

We represent either the x_j and the y_k variables as their correlation with the t_1, t_2, ... basis

$$G_h^X = \underbrace{X'N \frac{t_h}{\sqrt{\lambda_h}}}_{= Cor(X,T)} = \begin{pmatrix} \vdots \\ cor(x_j, t_h) \\ \vdots \end{pmatrix}$$

$$G_h^Y = \underbrace{Y'N \frac{t_h}{\sqrt{\lambda_h}}}_{= Cor(Y,T)} = \begin{pmatrix} \vdots \\ cor(y_k, t_h) \\ \vdots \end{pmatrix}$$

(as supplementary)

***X* individuals**

Displays of x_i individuals on p_1, p_2, ... basis

$$Xp_h = t_h$$

Application of PCR: The linnerud problem

Prediction of performance from physical measurements of 20 athlets of a Gymnasium

We form the vector

$x = (Weight, Waist, Pulse)$

to predict the vector

$y = (Tractions, Push\text{-}ups, Jumps)$

Editor de datos

	Weight	Waist	Pulse	Tractions	Push.ups	Jumps
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
7	211	38	56	8	101	38
8	167	34	60	6	125	40
9	176	31	74	15	200	40
10	154	33	56	17	251	250
11	169	34	50	17	120	38
12	166	33	52	13	210	115
13	154	34	64	14	215	105
14	247	46	50	1	50	50
15	193	36	46	6	70	31
16	202	37	62	12	210	120
17	176	37	54	4	60	25
18	157	32	52	11	230	80
19	156	33	54	15	225	73
20	138	33	68	2	110	43

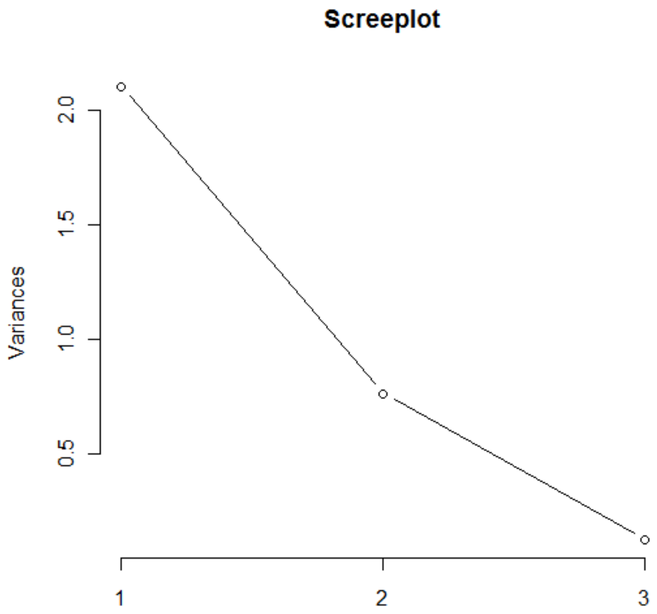
PCR in R

```
library(pls)

pcr <- pcr(formula, ncomp, data, subset, na.action,
    scale = FALSE, validation = c("none", "CV", "LOO"))

# Results

pcr$scores          # principal components of X
pcr$loadings        # OLS coef. of scale(X) ~ PC (= eigenvectors of the X space)

# looking for the significant components
plot(R2(pcr), legendpos = "topright")

# fitted values versus observed
plot(pcr, ncomp = 1, asp = 1, line = TRUE)

# plot of loadings
plot(pcr, "loadings", comps = 1, legendpos = "topleft", labels =
    rownames(pcr$loadings))
abline(h = 0)
```

Selecting the number of components

PCA of X block:

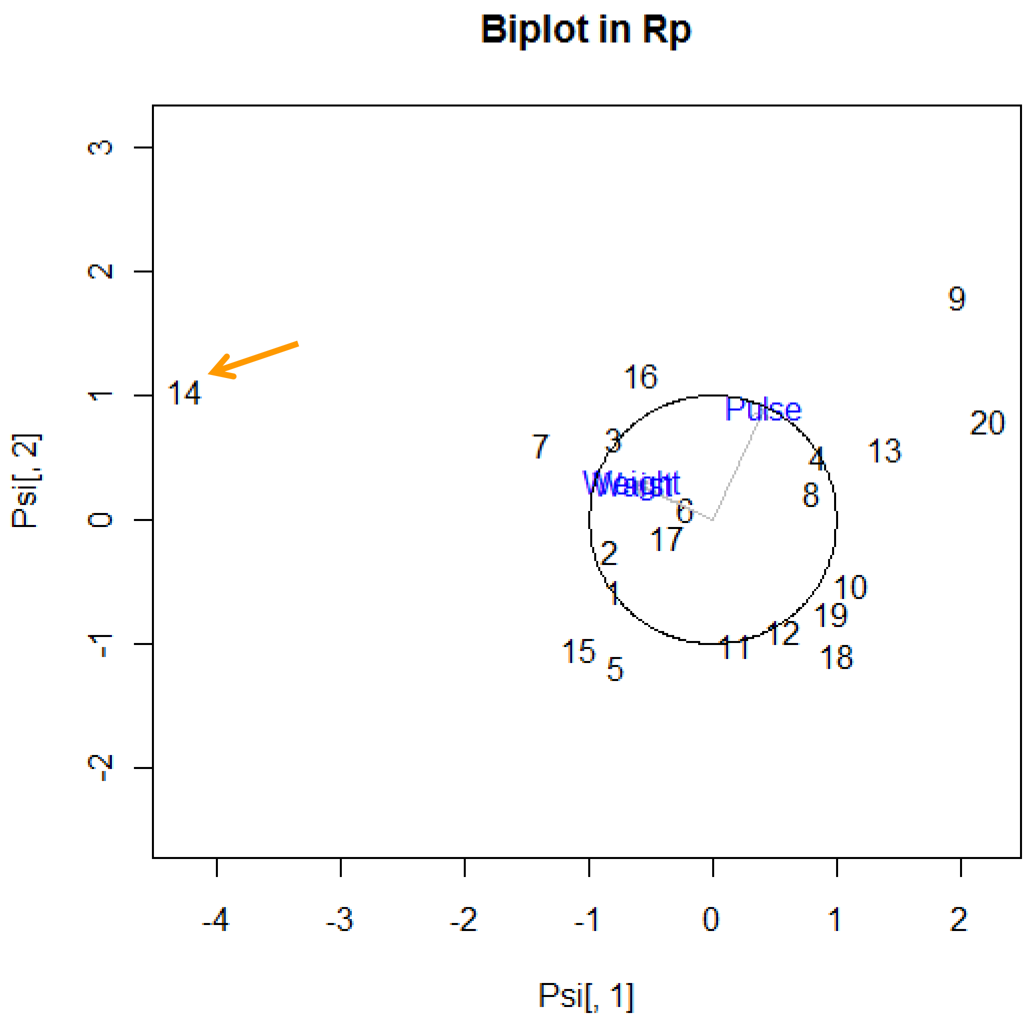
```
Total inertia:  3
 eigenvalue % explained % cumulated
1      2.1041      70.138       70.14
2      0.7662      25.541       95.68
3      0.1296       4.321      100.00
```



We select 2 components,
These componets are optimal regarding X, but it is assumed regarding Y

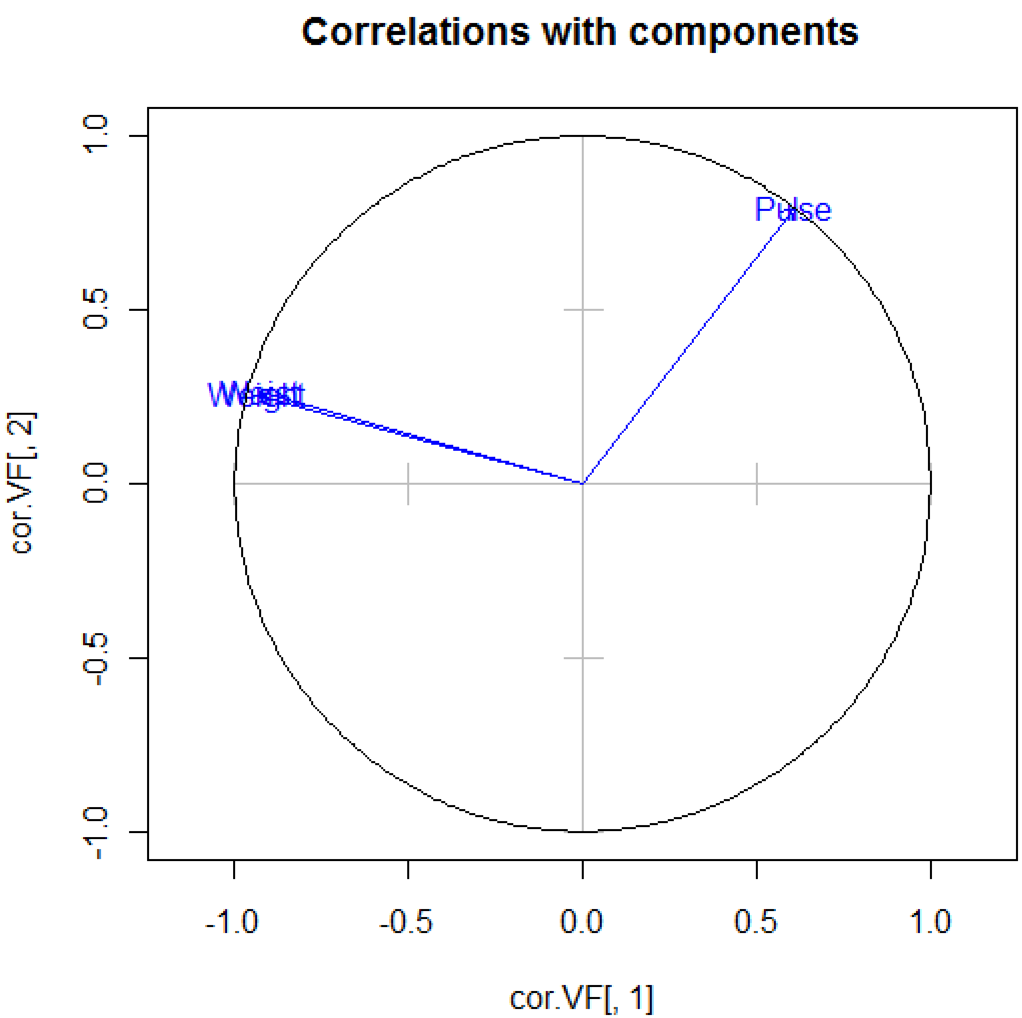
(even though by the last elbow rule 1 component is enough)

Biplot in R^p



Look at individual 14, …
and 9, …

Biplot in R^n (only variables)



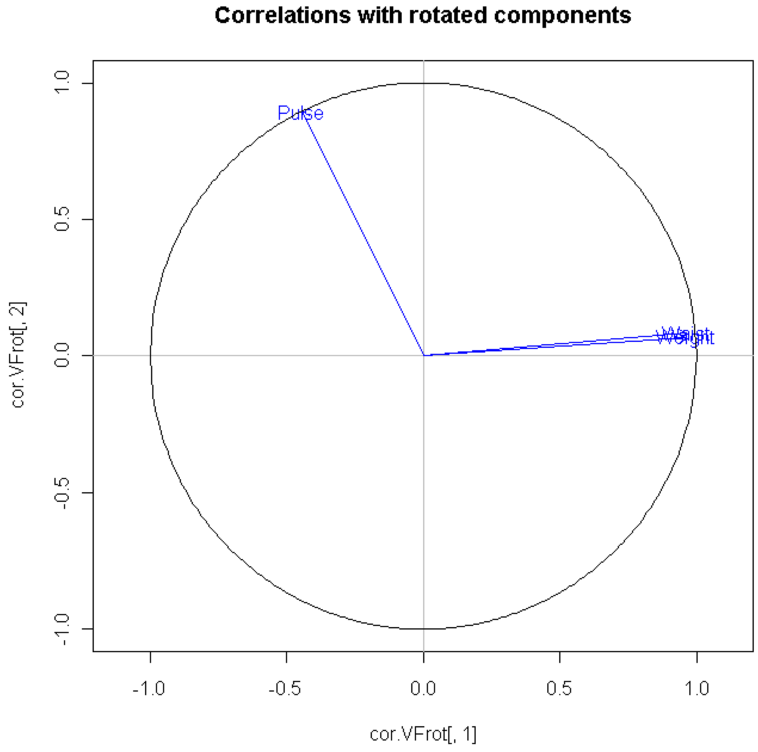
Weight and Waist very correlated and very few and negatively with Pulse

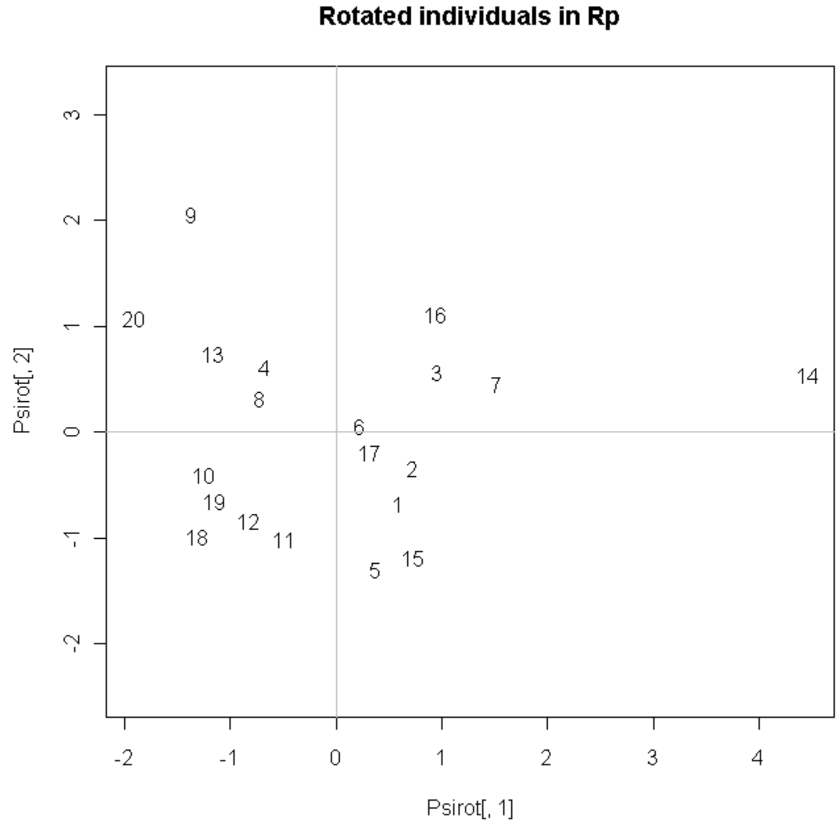
```
Varimax rotation
>pcrot = varimax(cor.VF)

Loadings:
         PC1    PC2
Weight -0.964
Waist  -0.964
Pulse   0.444  0.896
```

It confirms 2 components for X

Rotated components

```
> eigrot
    Dim.1     Dim.2
2.0556298 0.8147284
```

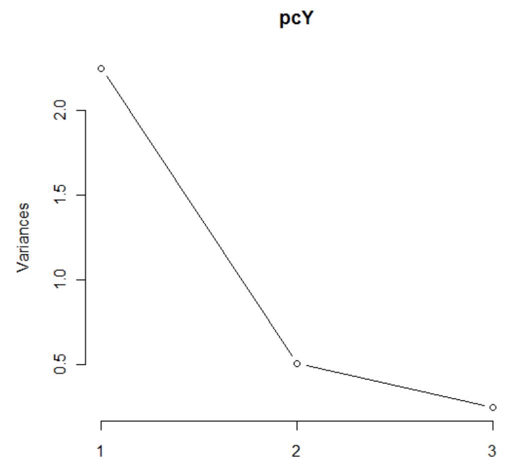


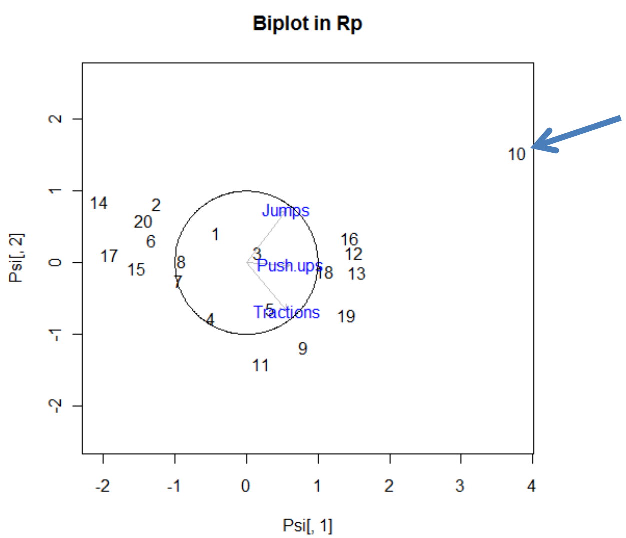


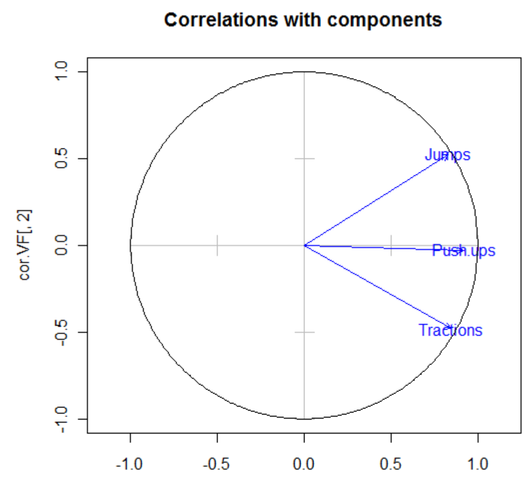
PCA of Y block

Just for explorative purposes

```
Total inertia:  3
eigenvalue % explained % cumulated
1      2.2444      74.814      74.81
2      0.5050      16.834      91.65
3      0.2505       8.351     100.00
```



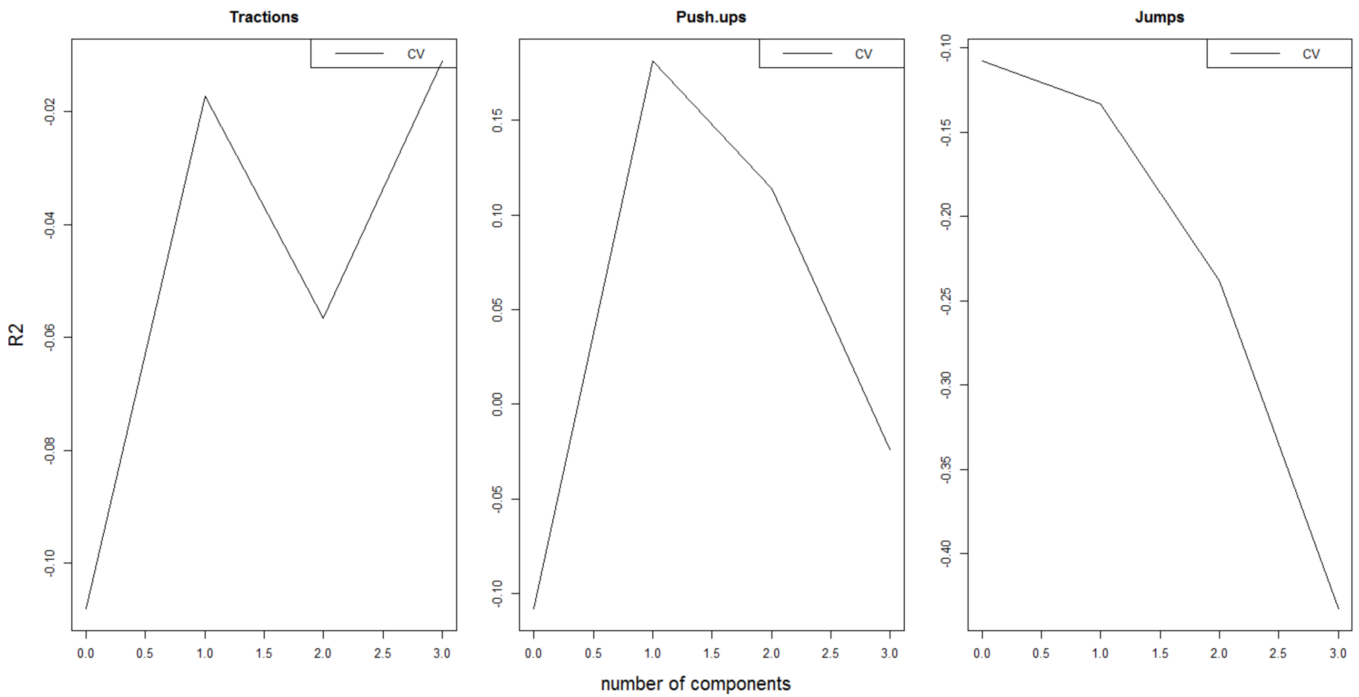




How many components do we have in this case? (unidimensional block)

Principal Component Regression
Looking for the significative components

```
> library(pls)
> pc <- pcr(Ys~Xs, validation="LOO")
> plot(R2(pc), legendpos = "topright")
```



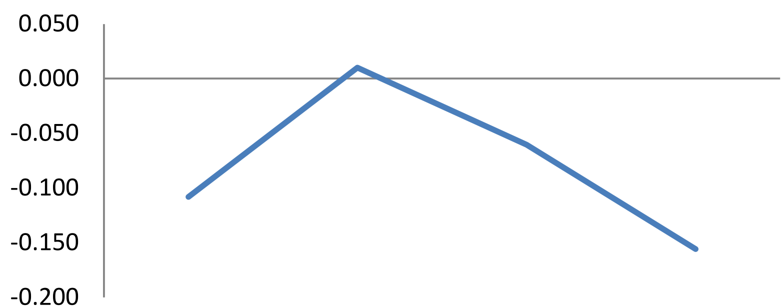
How many significative components t_h would you select?. What is the difference with the previous selections?

Selecting the number of dimensions by CV

```
> RMSEP(pcr)

Response: Tractions
       (Intercept)  1 comps  2 comps  3 comps
CV           1.026   0.9831    1.002    0.980
adjCV        1.026   0.9794    0.998    0.975

Response: Push.ups
       (Intercept)  1 comps  2 comps  3 comps
CV           1.026   0.8820   0.9176   0.9863
adjCV        1.026   0.8786   0.9141   0.9790

Response: Jumps
       (Intercept)  1 comps  2 comps  3 comps
CV           1.026    1.038    1.085    1.167
adjCV        1.026    1.035    1.081    1.160
```

```
> R2(pcr)

Response: Tractions
(Intercept)      1 comps      2 comps      3 comps
   -0.10803     -0.01732     -0.05660     -0.01109

Response: Push.ups
(Intercept)      1 comps      2 comps      3 comps
   -0.10803      0.18117      0.11374     -0.02393

Response: Jumps
(Intercept)      1 comps      2 comps      3 comps
    -0.1080      -0.1333      -0.2384      -0.4324
```

Average R2	-0.108	0.010	-0.060	-0.156



The regression coefficients

$B_1 =$

```
> pc$coefficients
, , 1 comps

          Tractions   Push.ups       Jumps
Weight -0.2042933 -0.2524757 -0.08658108
Waist  -0.2034283 -0.2514067 -0.08621450
Pulse   0.1326935  0.1639892  0.05623655

, , 2 comps

          Tractions     Push.ups       Jumps
Weight -0.2577975 -0.302430547 -0.12077405
Waist  -0.2603632 -0.304564644 -0.12259989
Pulse  -0.0369661  0.005584523 -0.05218781

, , 3 comps

           Tractions    Push.ups       Jumps
Weight  0.36825422  0.28715480 -0.25898620
Waist  -0.88182433 -0.88982675  0.01459879
Pulse  -0.02584743  0.01605555 -0.05464246
```

```
cor(X,Y)
       Tractions Push.ups   Jumps
Weight    -0.390   -0.493 -0.2263
Waist     -0.552   -0.646 -0.1915
Pulse      0.151    0.225  0.0349
```

```
Coefficients MVR:
         Tractions Push.ups   Jumps
XsWeight     0.368    0.287  -0.259
XsWaist     -0.882   -0.890   0.015
XsPulse     -0.026    0.016  -0.055
```

The scores

```
> pc$scores
       Comp 1      Comp 2      Comp 3
1  -0.7970972 -0.56746028 -0.21373303
2  -0.8291653 -0.24352078  0.06025763
3  -0.7857148  0.65392128  0.15480618
4   0.8549609  0.51257829  0.37879847
5  -0.7767178 -1.18975262 -0.36926091
6  -0.2145473  0.08396556  0.03445480
7  -1.3709584  0.61080266 -0.35904893
8   0.8088168  0.22344345  0.01858264
9   1.9865215  1.80589573 -0.92457340
10  1.1159738 -0.52646599  0.17921371
11  0.1768191 -1.01308650 -0.02142135
12  0.5711831 -0.89114055 -0.15869518
13  1.3796783  0.57651646  0.38523325
14 -4.2588620  1.03418680  0.37676158
15 -1.0811812 -1.04815040 -0.26422469
16 -0.5882278  1.16703250 -0.33086421
17 -0.3742751 -0.14239439  0.43040413
18  1.0059999 -1.09075055 -0.12002278
19  0.9478612 -0.75520949  0.12522615
20  2.2289324  0.79958880  0.61810592
```

$=T$

The loadings (coefficients)

```
lm(Xs~pc$scores-1)     # Xs = T * loadings'
```

```
> pc$loadings

Loadings:
       Comp 1 Comp 2 Comp 3
Weight -0.644  0.286 -0.710
Waist  -0.641  0.305  0.704
Pulse   0.418  0.908

               Comp 1 Comp 2 Comp 3
SS loadings     1.000  1.000  1.000
Proportion Var  0.333  0.333  0.333
Cumulative Var  0.333  0.667  1.000
```

$=P$

The Yloadings

```
lm(Ys~pc$scores-1)     # Ys = T * Yloadings'
```

```
> pc$Yloadings

Loadings:
          Comp 1 Comp 2 Comp 3
Tractions  0.317 -0.187 -0.882
Push.ups   0.392 -0.174 -0.831
Jumps      0.135 -0.119  0.195

               Comp 1 Comp 2 Comp 3
SS loadings     0.273  0.080  1.506
Proportion Var  0.091  0.027  0.502
Cumulative Var  0.091  0.117  0.620
```

The fitted values

```
> pc$fitted.values
, , 1 comps

     Tractions    Push.ups       Jumps
1  -0.25297810 -0.31264279 -0.10721409
2  -0.26315566 -0.32522071 -0.11152742
3  -0.24936562 -0.30817830 -0.10568310
4   0.27134252  0.33533844  0.11499708
5  -0.24651018 -0.30464941 -0.10447294
6  -0.06809177 -0.08415116 -0.02885782
7  -0.43510683 -0.53772644 -0.18440167
8   0.25669759  0.31723952  0.10879044
9   0.63047066  0.77916667  0.26719839
10  0.35418130  0.43771468  0.15010480
11  0.05611782  0.06935317  0.02378317
12  0.18127877  0.22403323  0.07682736
13  0.43787428  0.54114659  0.18557454
14 -1.35165293 -1.67043921 -0.57284107
15 -0.34313900 -0.42406807 -0.14542499
16 -0.18668833 -0.23071863 -0.07911997
17 -0.11878525 -0.14680066 -0.05034212
18  0.31927841  0.39457997  0.13531268
19  0.30082669  0.37177643  0.12749270
20  0.70740564  0.87424670  0.29980403
```

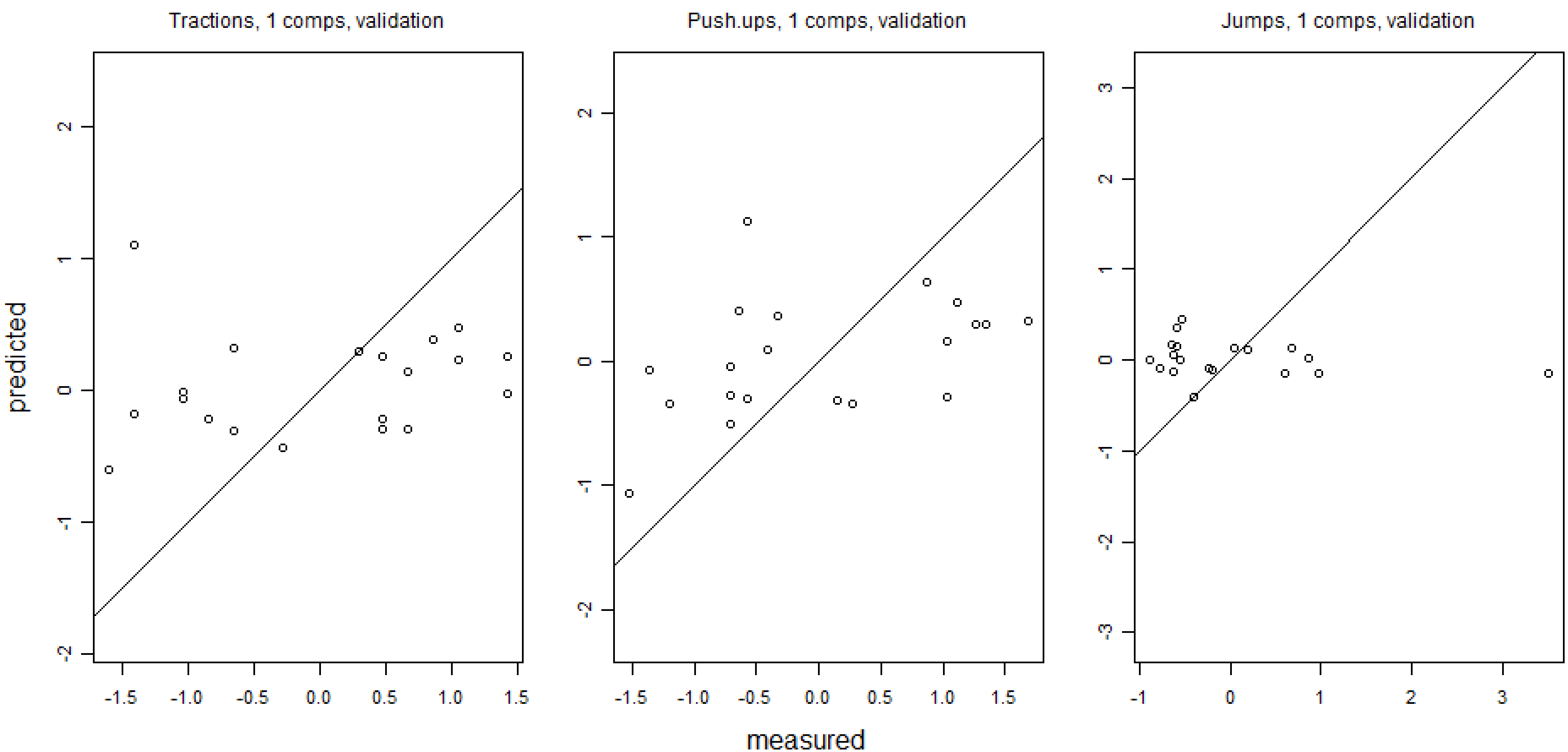
```
, , 2 comps

     Tractions    Push.ups       Jumps
1  -0.14698496 -0.21368107 -0.03947707
2  -0.21766960 -0.28275213 -0.08245865
3  -0.37150838 -0.42221830 -0.18374086
4   0.17560052  0.24594782  0.05381125
5  -0.02428209 -0.09716360  0.03754634
6  -0.08377529 -0.09879426 -0.03888069
7  -0.54919568 -0.64424681 -0.25731242
8   0.21496168  0.27827230  0.08211828
9   0.29315620  0.46422914  0.05163087
10  0.45251731  0.52952724  0.21294839
11  0.24534731  0.24602946  0.14471404
12  0.34773057  0.37944286  0.18320169
13  0.33018959  0.44060553  0.11675649
14 -1.54482364 -1.85079527 -0.69629066
15 -0.14736010 -0.24127684 -0.02030859
16 -0.40467264 -0.43424219 -0.21842719
17 -0.09218810 -0.12196792 -0.03334468
18  0.52301438  0.58480041  0.26551421
19  0.44188858  0.50348049  0.21764111
20  0.55805434  0.73480315  0.20435812

, , 3 comps

      Tractions    Push.ups       Jumps
1   0.041570495 -0.03610864 -0.08110407
2  -0.270828934 -0.33281503 -0.07072278
3  -0.508078522 -0.55083347 -0.15359055
4  -0.158575809 -0.06876333  0.12758667
…
```

The residuals

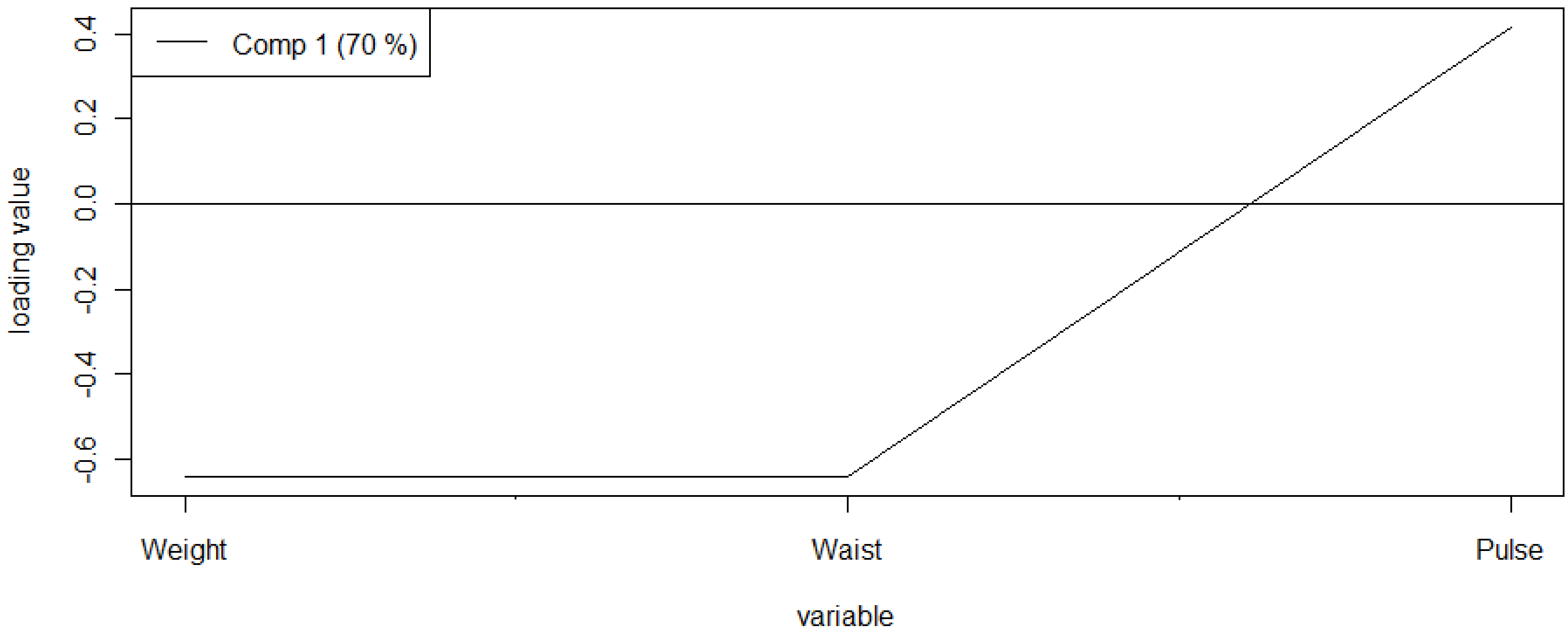
```
> pc$residuals
, , 1 comps

     Tractions     Push.ups       Jumps
1  -0.58882399  0.57556273 -0.09365385
2  -1.14615346 -0.24297404 -0.08934052
3   0.73174659 -0.40386323  0.70438658
4   0.21103845 -0.98344808 -0.76440509
5   0.91806016  0.45568853 -0.13539849
6  -0.96287933 -0.62789037 -0.52304148
7   0.16081177 -0.17431509 -0.44550460
8  -0.90933067 -0.64568965 -0.69969323
9   0.41941734  0.09110632 -0.85810117
10  1.07404471  1.24769002  3.35435826
11  1.37210819 -0.47771818 -0.65368944
12  0.49027121  0.80606951  0.79490051
13  0.42284471  0.56887102  0.49113591
14 -0.24682519  0.14326596  0.17695570
15 -0.30949408 -0.78344569 -0.62099348
16  0.66906930  1.26082136  1.04835656
17 -0.91218584 -1.22054284 -0.83308680
18 -0.02606644  0.95518226  0.05385422
19  0.74906131  0.89807093 -0.07483800
20 -2.11671475 -1.44244146 -0.83220159
```

```
, , 2 comps

     Tractions     Push.ups       Jumps
1  -0.69481713  0.47660101 -0.16139087
2  -1.19163952 -0.28544263 -0.11840929
3   0.85388935 -0.28982323  0.78244434
4   0.30678045 -0.89405746 -0.70321926
5   0.69583207  0.24820271 -0.27741777
6  -0.94719581 -0.61324728 -0.51301861
7   0.27490062 -0.06779472 -0.37259385
8  -0.86759476 -0.60672244 -0.67302106
9   0.75673180  0.40604384 -0.64253366
10  0.97570870  1.15587747  3.29151467
11  1.18287871 -0.65439446 -0.77462031
12  0.32381941  0.65065987  0.68852618
13  0.53052940  0.66941208  0.55995396
14 -0.05365449  0.32362202  0.30040529
15 -0.50527298 -0.96623691 -0.74610988
16  0.88705361  1.46434492  1.18766377
17 -0.93878300 -1.24537558 -0.85008424
18 -0.22980242  0.76496182 -0.07634731
19  0.60799941  0.76636687 -0.16498641
20 -1.96736346 -1.30299791 -0.73675568

, , 3 comps

     Tractions    Push.ups       Jumps
1  -0.88337258  0.2990286 -0.11976388
2  -1.13848018 -0.2353797 -0.13014517
3   0.99045949 -0.1612081  0.75229404
4  0.64095678 -0.5793463 -0.77699468
5 …
```

Fitted values versus observed

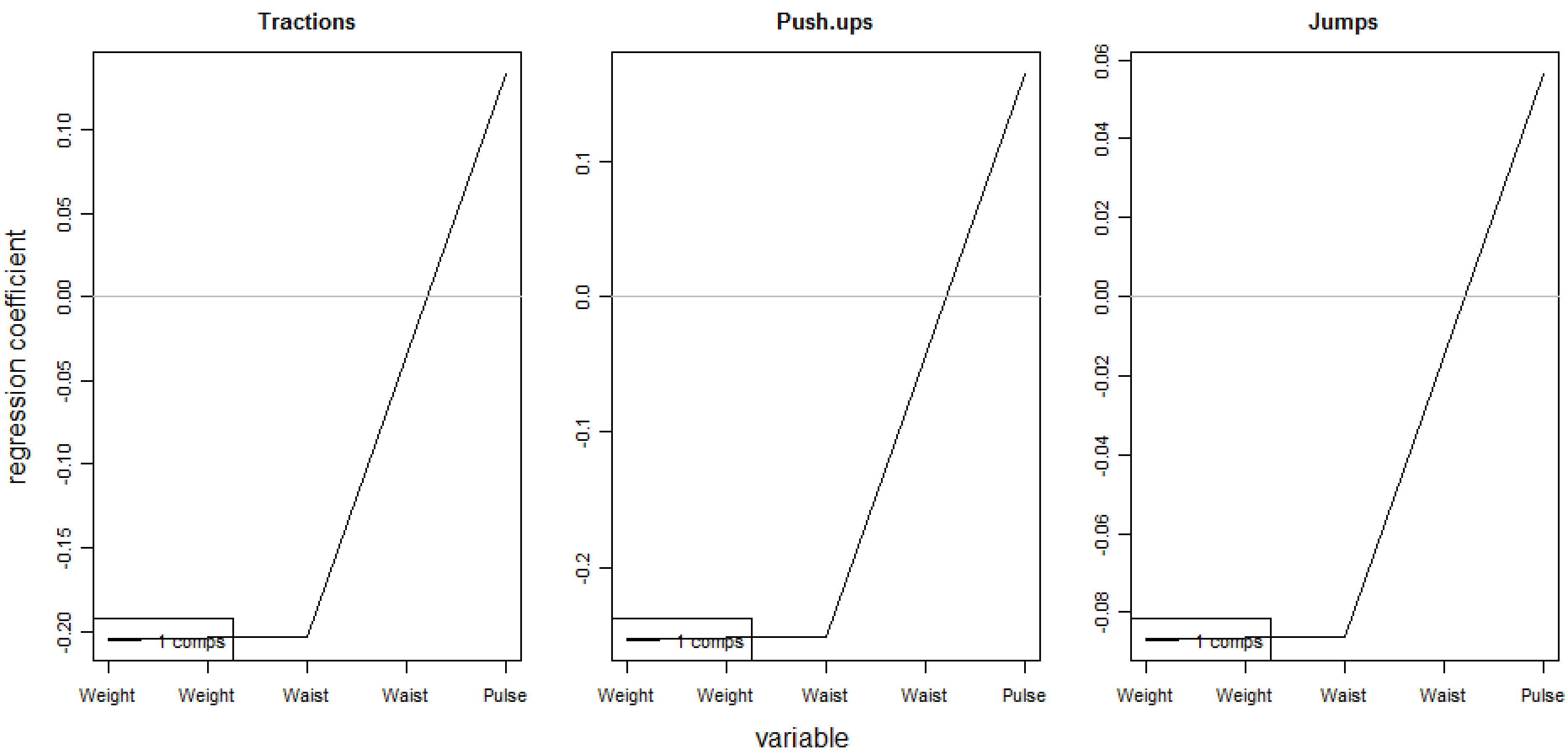


Loadings plot



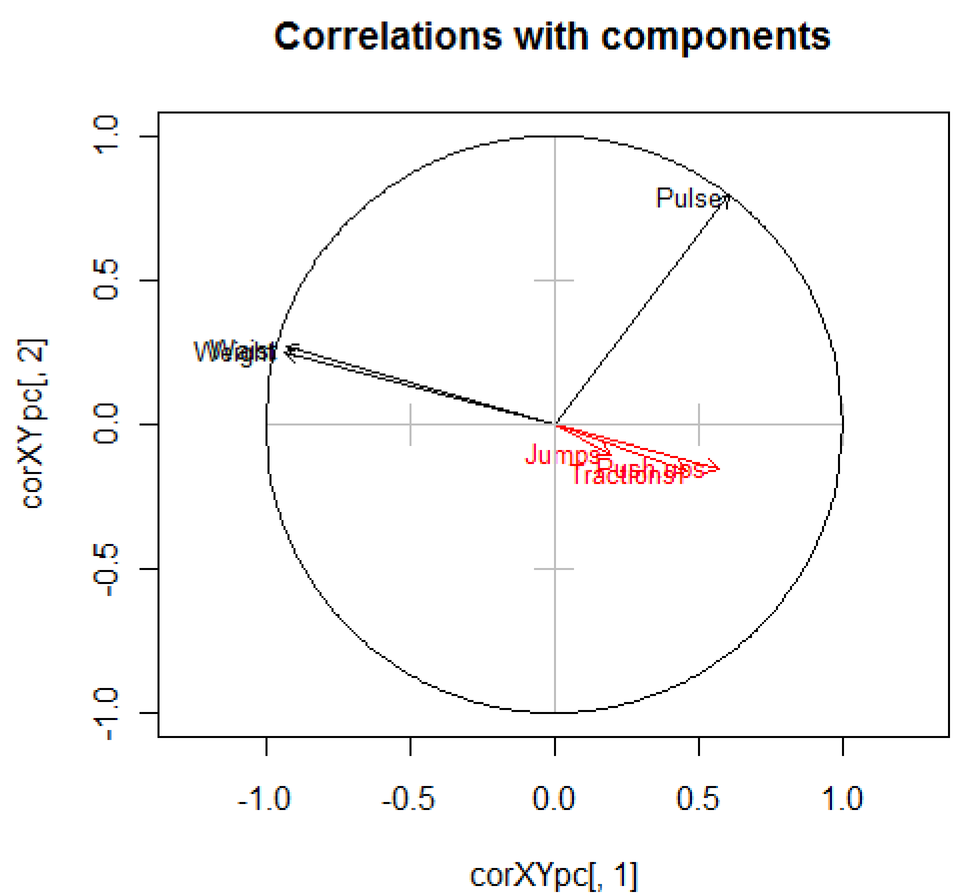
Loadings give the importance of each variable in the formation of each component

Regression coefficients



Regression coefficients give the importance of each variable in the prediction of each response variable

Plot of correlations

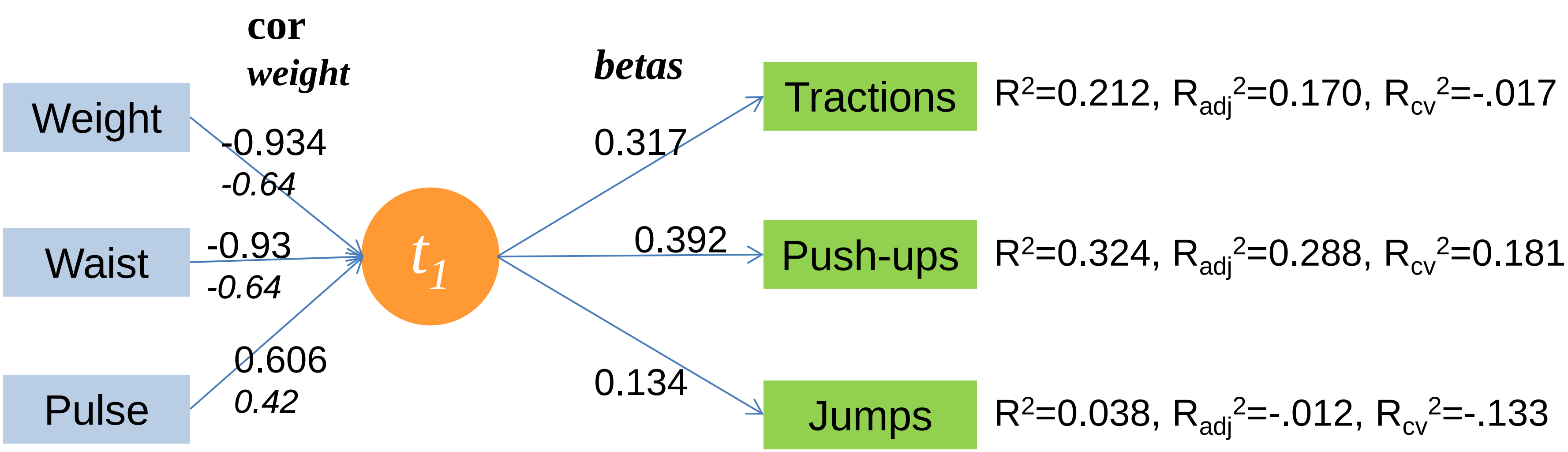


It is a good plot?

Regressing the responses on t_1

```
> lmY <- lm(Ys~pc$scores[,1:nd]-1)
> summary(lmY)
Response Pulls :

                  Estimate Std. Error t value Pr(>|t|)
pc$scores[, 1:nd]   0.3174     0.1404   2.261   0.0357 *

Residual standard error: 0.8877 on 19 degrees of freedom
Multiple R-squared:  0.2119,    Adjusted R-squared:  0.1705
F-statistic:  5.11 on 1 and 19 DF,  p-value: 0.03572

Response Squats :
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
pc$scores[, 1:nd]   0.3922     0.1301   3.016  0.00711 **

Residual standard error: 0.8224 on 19 degrees of freedom
Multiple R-squared:  0.3237,    Adjusted R-squared:  0.2881
F-statistic: 9.094 on 1 and 19 DF,  p-value: 0.007111

Response Jumps :
                  Estimate Std. Error t value Pr(>|t|)
pc$scores[, 1:nd]   0.1345     0.1551   0.867    0.397

Residual standard error: 0.9808 on 19 degrees of freedom
Multiple R-squared:  0.03807,   Adjusted R-squared:  -0.01256
F-statistic: 0.7519 on 1 and 19 DF,  p-value: 0.3967

> summary(manova(lmY))
                  Df  Pillai approx F num Df den Df  Pr(>F)
pc$scores[, 1:nd]  1 0.39696   3.7301      3     17 0.03158 *
Residuals         19
```

Summary of results of PCR on Linnerud case



Communality=0.701

```
corXpc <- cor(Xs,pc$scores)

# communalities of X
rowMeans(apply(corXpc^2,1,cumsum))
```

Redundancy=0.191

```
corYpc <- cor(Ys,pc$scores)

# redundancies of Y
rowMeans(apply(corYpc^2,1,cumsum))
```