R User Group Meeting

Gerhard Welzl Benno Pütz

November 29, 2004

Next meeting: Jan. 31, 2005

Contents

1 Minutes

1.1 Follow-Up to last month's meeting (Pütz)

The delayed output problem presented last month has been solved and an update to Sweave has been presented.
Both topics are detailed in a separate document (PDF format).

1.2 Upgrading (Wjst, Pütz)

The step to R 2.0.0 required reinstalling all packages. In the meantime, some packages require R version 2.0.1. This may lead to a lot of work when many packages are installed.

There are, however, two R comands, that come in handy

```
install.packages()
update.packages()
```

They are described in R's help system.

1.3 MAAnova (Welzl)

The MAAnova is still not updated to R 2.0. A work around has been suggested on the MAAnova list (untested under Unix, problems under Windows):

unpack the zip.file with the MAANOVA package
edit the DESCRIPTION file by removing the R >1.8.0 dependence
create a new zip.file (".tar.gz")
install the package

2 Regression Methods (Welzl)

Two problems arising in ecotoxicolgical project (Hense):

- association between groups of variables (using R-package `pls.pcr`)
- multiple regression (using R-package `hier.part`)

2.1 Functions

First define some plotting routines for use in the following sections:

```
> plot.SCR <- function(x, y, labels, type = "two PCAs") {
+     plot(x, y, , main = paste("Score plot based on", type))
```

```
+     abline(0, 1, col = "grey")
+     text(x, y, labels, cex = 0.5, col = "grey40", pos = 1)
+     mtext(paste("Correlation ", round(cor(x, y), 3)), adj = 0,
+         cex = 0.7)
+ }
> plot.load <- function(loads, labels) {
+     plot(loads, xlim = c(0.7, 2.3), main = "Loadings", type = "n",
+         axes = FALSE)
+     axis(1, at = 1:2)
+     axis(2)
+     box()
+     text(loads, labels, cex = 0.65)
+     abline(h = 0)
+ }
```

This will put the two previous plots side by side to shorten the output (this function will not be seen in the listings below, but is used for the side by side plots in subsection 2.2)

```
> side.by.side <- function(x, y, s.labels, loads, l.labels, type = "two PCAs") {
+     def.par <- par(no.readonly = TRUE)
+     layout(matrix(1:2, nc = 2), width = 2:1, height = 2.3, respect = T)
+     plot.SCR(x, y, s.labels, type)
+     plot.load(loads, l.labels)
+     par(def.par)
+ }
```

2.2 Association between groups of variables X Y (Welzl)

Here variables in a multivariate system are divided a priori into two sets of p (in X) and q (in Y) variables respectively on external grounds. In the example data there are the abundance data of p=7 phytoplankton species and of q=3 zooplankton species. The aim is to measure the association between the two collections of variables. The analysis is therefore based on the decomposition of the correlation R or covariance matrix C with respect to X and Y. Because the example data are standardized there is no difference betwen R and C. Some techniques are used to

- reduce dimensionality
- gain insight into the system (by plotting and interpreting the linear combinations (loadings) for some few components)
- inspect the data for irregularities (by plotting the data transformed by the linear combinations (standardized scores))

Literature: Krzanowski, W.J.: Principles of Multivariate Analysis
and for PLS books and articles of Bookstein.

2.2.1 Data

Read the data file

```
> data.exp <- dget("dataexp.put")
> dim(data.exp)
```

```
[1] 27 10
```

	X1	X2	X3	X4	X5	X6	X7	Y1	Y2	Y3
O1	0.42	−2.18	1.20	−1.24	−0.76	0.75	−0.73	0.64	0.49	0.85
O2	−0.48	0.40	1.00	1.02	−1.28	0.27	−0.15	−0.56	0.50	2.26
O3	0.95	−0.02	−3.27	0.17	−0.17	0.63	0.20	−1.35	−0.35	1.11
O4	−0.12	−0.83	1.33	−0.39	−0.43	1.63	0.18	−0.63	0.93	0.60
O5	−0.36	−1.10	0.51	0.08	−0.38	0.63	−0.52	−0.03	0.22	1.68
O6	1.49	0.63	1.29	1.16	0.01	−0.08	0.40	−0.35	−0.56	−0.52
O7	−0.19	−0.69	0.60	−0.97	0.56	0.82	0.23	−1.27	0.63	−0.09
O8	−0.05	−0.19	0.32	0.04	−0.46	0.94	−0.92	0.33	0.58	0.18
O9	0.56	0.91	−0.24	0.61	0.14	−0.18	1.10	−0.36	−1.19	−1.96
O10	0.26	−0.73	0.79	0.43	0.63	0.49	0.95	−0.42	0.78	0.35
O11	1.09	1.12	0.64	0.35	−0.10	2.17	−0.87	0.15	0.33	−0.18
O12	0.31	1.87	0.49	−0.15	−0.19	−0.67	0.70	−0.36	−1.26	−1.95
O13	−1.18	−0.65	−0.05	1.15	0.65	−0.50	−0.06	0.36	0.85	−0.14
O14	0.35	0.27	−0.45	0.72	0.14	1.41	−1.36	0.19	0.44	−0.69
O15	1.08	1.24	0.19	−0.16	−0.56	−1.71	1.24	0.27	−1.62	−1.09
O16	−1.20	−0.15	0.34	0.10	0.97	0.65	0.87	0.60	1.22	0.98
O17	−0.80	−0.38	−0.69	0.80	0.17	−0.49	−1.16	−0.59	0.82	−0.07
O18	1.90	1.20	−0.24	0.39	0.17	−0.67	0.96	1.20	−2.49	−0.90
O19	−1.98	−0.45	0.50	−0.21	0.50	−0.61	1.17	−0.49	1.35	0.60
O20	−0.44	−1.30	−1.88	−0.49	0.71	−1.13	−0.60	−0.35	1.02	0.73
O21	1.18	1.08	−0.28	0.10	0.91	−1.76	−0.52	1.98	−1.83	−0.96
O22	−0.14	1.10	0.56	0.14	−0.44	−0.09	1.77	−1.11	0.21	0.60
O23	0.45	−1.21	−0.22	−3.72	−0.53	0.66	−0.19	−1.20	0.37	−0.79
O24	−0.82	0.64	−0.87	1.47	1.30	−0.25	−2.18	2.54	−1.13	0.51
O25	−2.40	0.62	−0.55	−0.85	−3.58	−1.57	−1.54	0.04	−0.03	0.31
O26	−0.12	−1.40	0.05	0.17	−0.03	−0.78	−0.35	−1.13	0.01	−0.77
O27	0.23	0.22	−1.08	−0.70	2.04	−0.57	1.37	1.91	−0.30	−0.63

Table 1: Data used in section 2, provided by B. Hense

Calculation of correlation matrix R and decomposition with respect to X (phytoplankton, variable 1 to 7 of `data.exp`) and Y (zooplankton, variable 8 to 10 of `data.exp`)

```
> R <- cor(data.exp)
> phy <- 1:7
> zoo <- 8:10
> RXX <- R[phy, phy]
> RXY <- R[phy, zoo]
> RYX <- R[zoo, phy]
> RYY <- R[zoo, zoo]
```

	X1	X2	X3	X4	X5	X6	X7	Y1	Y2	Y3
X1	1.00	0.30	0.01	0.01	0.22	0.14	0.24	0.09	−0.61	−0.43
X2	0.30	1.00	−0.04	0.38	−0.05	−0.27	0.22	0.29	−0.66	−0.44
X3	0.01	−0.04	1.00	0.01	−0.18	0.28	0.18	−0.09	0.17	0.03
X4	0.01	0.38	0.01	1.00	0.23	−0.06	−0.08	0.26	−0.16	0.09
X5	0.22	−0.05	−0.18	0.23	1.00	0.02	0.25	0.38	−0.04	−0.19
X6	0.14	−0.27	0.28	−0.06	0.02	1.00	−0.14	−0.26	0.47	0.31
X7	0.24	0.22	0.18	−0.08	0.25	−0.14	1.00	−0.23	−0.13	−0.21
Y1	0.09	0.29	−0.09	0.26	0.38	−0.26	−0.23	1.00	−0.44	−0.12
Y2	−0.61	−0.66	0.17	−0.16	−0.04	0.47	−0.13	−0.44	1.00	0.57
Y3	−0.43	−0.44	0.03	0.09	−0.19	0.31	−0.21	−0.12	0.57	1.00

Table 2: Correlation matrix R with the decomposition indicated

2.2.2 First method: PCA with X, PCA with Y based on R_{xx} and R_{yy}

PCA with X / SVD(R_{xx})

Singular value decomposition (svd) on R_{XX} is equivalent to principal component analysis (princomp) on X.

```
> svd.RXX <- svd(RXX)
> pca.Xscores <- data.exp[, phy] %*% svd.RXX$u
```

standardize coefficients so that the scores will have variance 1

```
> Xscores.sd <- apply(pca.Xscores, 2, sd)
> pca.Xload <- svd.RXX$u/matrix(rep(Xscores.sd, 7), nrow = 7, byrow = TRUE)
> pca.Xscores <- data.exp[, phy] %*% pca.Xload
```

PCA with Y / SVD(R_{YY})

```
> svd.RYY <- svd(RYY)
> pca.Yscores <- data.exp[, zoo] %*% svd.RYY$u
```

again, standardize coefficients so that the scores will have variance 1

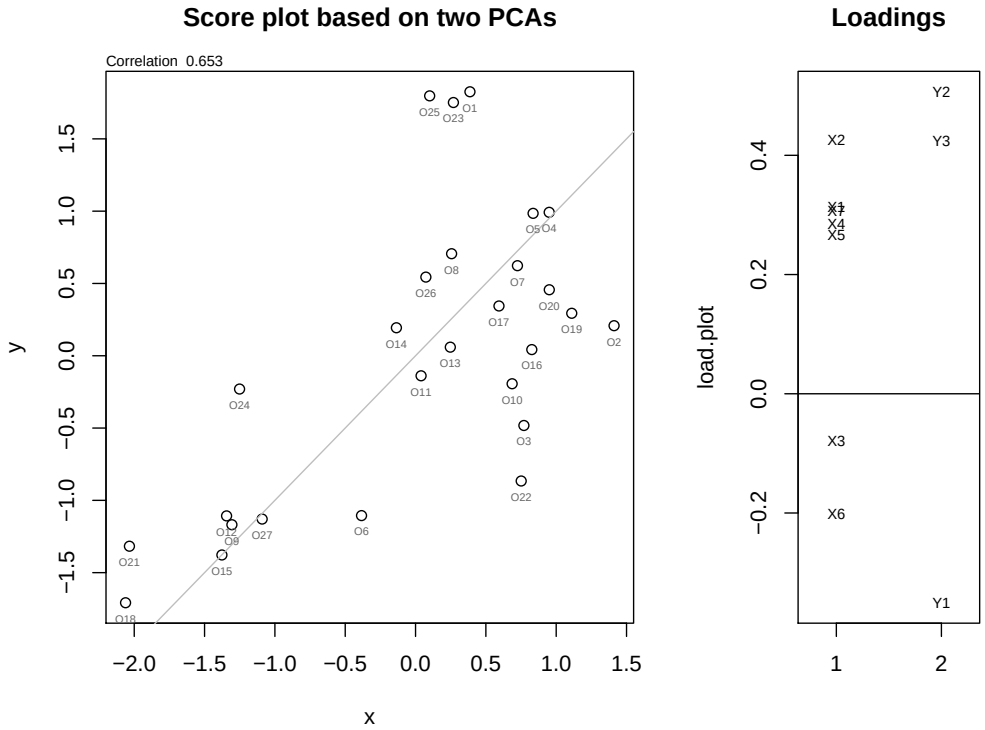
```
> Yscores.sd <- apply(pca.Yscores, 2, sd)
> pca.Yload <- svd.RYY$u/matrix(rep(Yscores.sd, 3), nrow = 3, byrow = TRUE)
> pca.Yscores <- data.exp[, zoo] %*% pca.Yload
```

Finally, plot of Yscores vs Xscores (Because the scores are not unique (e.g. can be multiplied by -1) we multiply the Xscores by -1, to have better comparability with the following plots.)

```
> plot.SCR(pca.Yscores[, 1], -pca.Xscores[, 1], rownames(data.exp))
```

Plot of loadings

```
> lab.plot <- c(rep(1, 7), rep(2, 3))
> pc1Y <- pca.Yload[, 1]
> pc1X <- pca.Xload[, 1]
> load.plot <- c(pc1X, pc1Y)
> matrix.plot <- cbind(c(rep(1, 7), rep(2, 3)), load.plot)
> plot.load(matrix.plot, colnames(data.exp))
```



With a correlation of

```
> cor(pca.Yscores[, 1], -pca.Xscores[, 1])
```

```
[1] 0.6528088
```

as also shown on the top of the plot.

2.2.3 Second method: partial least squares (PLS)

based on R_{xy} SVD(R_{xy})

```
> svd.RXY <- svd(RXY)
```

calculation of X-scores

```
> pls.Xscores <- data.exp[, phy] %*% svd.RXY$u
```

standardize coefficients so that the X-scores will have variance 1

```
> Xscores.sd <- apply(pls.Xscores, 2, sd)
> pls.Xload <- svd.RXY$u/matrix(rep(Xscores.sd, 7), nrow = 7, byrow = TRUE)
> pls.Xscores <- data.exp[, phy] %*% pls.Xload
```

calculation of Y-scores

```
> pls.Yscores <- data.exp[, zoo] %*% svd.RXY$v
```

standardize coefficients so that the Y-scores will have variance 1

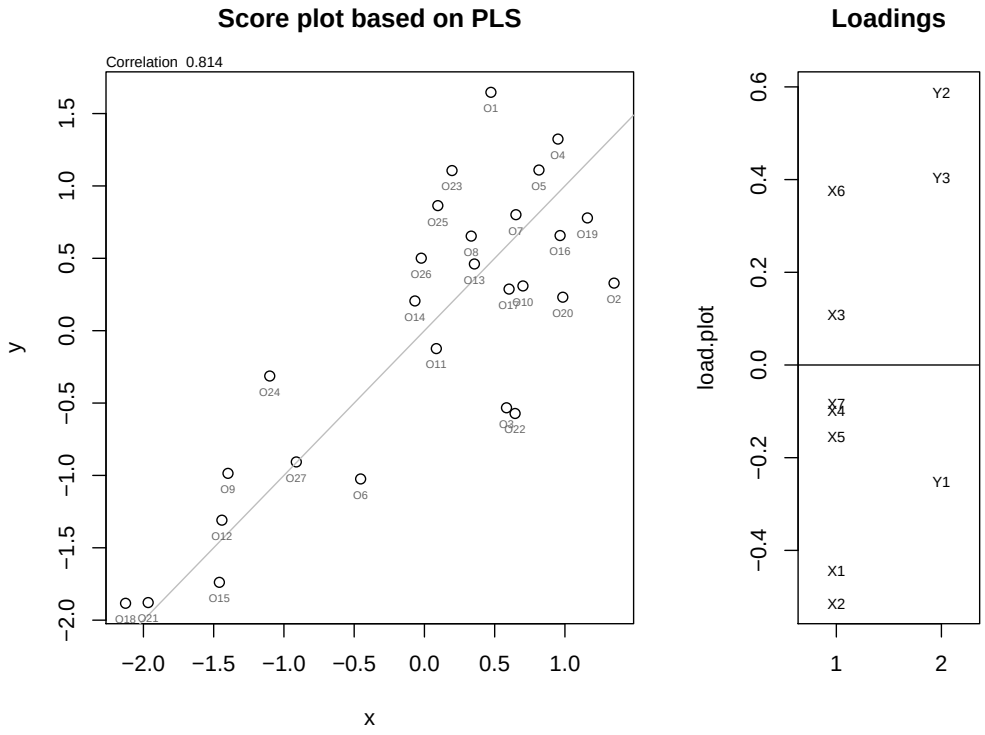
```
> Yscores.sd <- apply(pls.Yscores, 2, sd)
> pls.Yload <- svd.RXY$v/matrix(rep(Yscores.sd, 3), nrow = 3, byrow = TRUE)
> pls.Yscores <- data.exp[, zoo] %*% pls.Yload
```

plot of Yscores vs Xscores

```
> plot.SCR(pls.Yscores[, 1], pls.Xscores[, 1], rownames(data.exp),
+     type = "PLS")
```

plot of loadings

```
> pc1Y <- pls.Yload[, 1]
> pc1X <- pls.Xload[, 1]
> load.plot <- c(pc1X, pc1Y)
> matrix.plot <- cbind(c(rep(1, 7), rep(2, 3)), load.plot)
> plot.load(matrix.plot, colnames(data.exp))
```



PLS can also be done by `mvr()` in package `pls.pcr`

```
> library(pls.pcr)
> pls.XY <- mvr(data.exp[, phy], data.exp[, zoo], method = "SIMPLS")
```

`pls.XY$training$Xscores` and `pls.XY$training$Yscores` are (not standardized) score values; standardize coefficients so that the X- and Y-scores will have variance 1

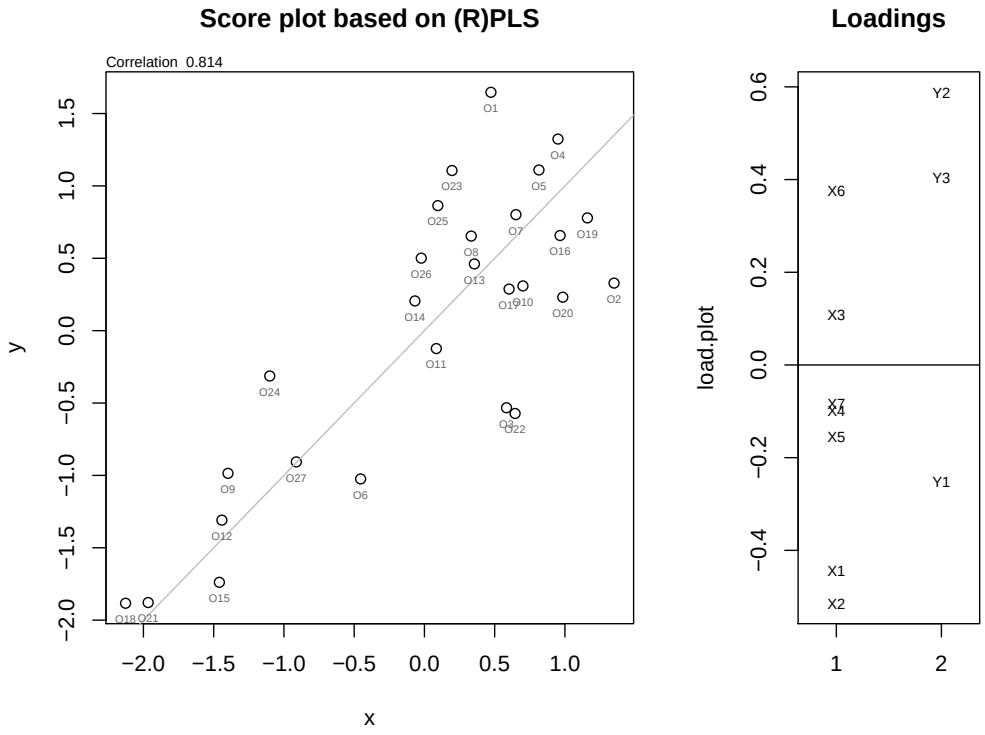
```
> Xscores.sd <- apply(pls.XY$training$Xscores, 2, sd)
> Yscores.sd <- apply(pls.XY$training$Yscores, 2, sd)
> Rpls.Xscores <- pls.XY$training$Xscores/matrix(rep(Xscores.sd,
+     27), nrow = 27, byrow = TRUE)
> Rpls.Yscores <- pls.XY$training$Yscores/matrix(rep(Yscores.sd,
+     27), nrow = 27, byrow = TRUE)
> plot.SCR(Rpls.Yscores[, 1], Rpls.Xscores[, 1], rownames(data.exp),
+     type = "(R)PLS")
```

`pls.XY$training$Yload` are (not standardized) coefficients for Y-matrix,
but attention !!!!!!!!
`pls.XY$training$Xload` are **not** the coefficients for the X-matrix (they are regression based), to get coefficients it would be easier to use scores and inverted data matrices

```
> library(MASS)
> Rpls.Xload <- ginv(data.exp[, phy]) %*% Rpls.Xscores
> Rpls.Yload <- ginv(data.exp[, zoo]) %*% Rpls.Yscores
```

plot of `loadingspc1Y=pca.Yload[,1]`

```
> pc1X <- Rpls.Xload[, 1]
> pc1Y <- Rpls.Yload[, 1]
> load.plot <- c(pc1X, pc1Y)
> matrix.plot <- cbind(c(rep(1, 7), rep(2, 3)), load.plot)
> plot.load(matrix.plot, colnames(data.exp))
```



2.2.4 Third method: canonical correlation analysis (CCA)

Based on R_{xx}, R_{yy}, and R_{xy}
Note: CCA very often means canonical *correspondence* analysis

The single value decomposition of the following matrix is the solution of the problem to find linear combinations on X and Y with maximal correlation.

SVD of `inv(t(chol(RYY)))*RYX*inv(RXX)*RXY*inv(chol(RYY))` can also be done by R-function `cancor()`

```
> cca.XY <- cancor(data.exp[, phy], data.exp[, zoo])
```

coefficients for matrix X are `cca.XY$xcoef` (not standardized) coefficients for matrix Y are `cca.XY$ycoef` (not standardized) calculation and standardization of X- and Y-scores

```
> cca.Xscores <- data.exp[, phy] %*% cca.XY$xcoef
> cca.Yscores <- data.exp[, zoo] %*% cca.XY$ycoef
> Xscores.sd <- apply(cca.Xscores, 2, sd)
> Yscores.sd <- apply(cca.Yscores, 2, sd)
> cca.Xload <- cca.XY$xcoef/matrix(rep(Xscores.sd, 7), nrow = 7,
+     byrow = TRUE)
> cca.Xscores <- data.exp[, phy] %*% cca.Xload
```

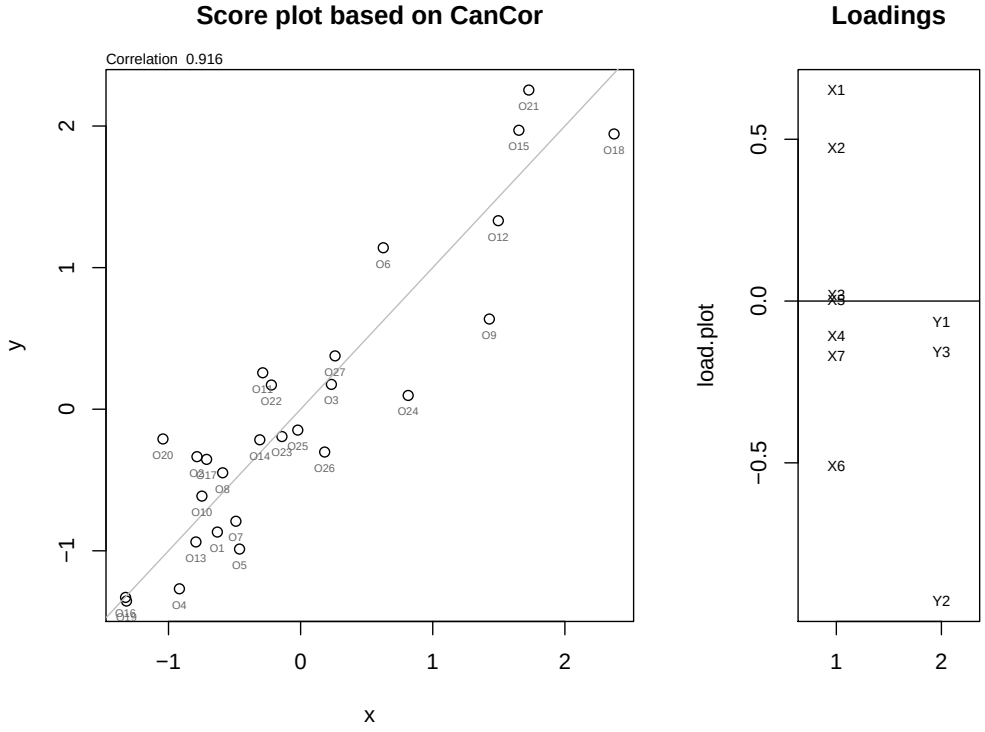
```
> cca.Yload <- cca.XY$ycoef/matrix(rep(Yscores.sd, 3), nrow = 3,
+     byrow = TRUE)
> cca.Yscores <- data.exp[, zoo] %*% cca.Yload
```

plot of Yscores vs Xscores

```
> plot.SCR(cca.Yscores[, 1], cca.Xscores[, 1], rownames(data.exp),
+     type = "CanCor")
```

plot of loadings

```
> pc1Y <- cca.Yload[, 1]
> pc1X <- cca.Xload[, 1]
> load.plot <- c(pc1X, pc1Y)
> matrix.plot <- cbind(c(rep(1, 7), rep(2, 3)), load.plot)
> plot.load(matrix.plot, colnames(data.exp))
```



2.3 Multiple regression y (vector) on X (matrix)

Many users of regression methods want to determine the relative importance of independent variables. All multivariate regression techniques yield a measure of fit (e.g. R^2). It is obvious to decompose this goodness-of-fit measure through incremental partitioning. The standard method is to follow only one order among the many possible orders available (variation partitioning). By taking a hierarchical approach in which all orders of variables are used, the average independent contribution of a variable is obtained (hierarchical partitioning).

Literature: Legendre, P. Legendre, L.: Numerical Ecology
and Chevan,A., Sutherland, M.: Hierarchical Partitioning. The American Statistician. 45,2, p90-96 (1991).

Example: y pls linear combination of phytoplankton data
X zooplankton data

```
> Y <- pls.Yscores[, 1]
> X <- data.exp[, zoo]
```

2.3.1 First method: Variation Partitioning

Calculate regressions of y on all subsets of X [1]

```
> library(hier.part)
```

```
Loading required package: gtools
```

```
> library(MASS)
> all.regs(Y, X, gof = "Rsqu", print.vars = TRUE)
```

```
regressions done: formatting results
  variable.combination       gof
1                Theta 0.0000000
2                   Y1  0.3138482
3                   Y2  0.8667001
4                   Y3  0.5963617
5               Y1 Y2  0.8942909
6               Y1 Y3  0.8164685
7               Y2 Y3  0.9509413
8            Y1 Y2 Y3  1.0000000
```

```
> design <- matrix(rep(1, 49), nrow = 7)
> design[1:3, 1:3] <- diag(3)
> design[1, 6] <- 0
> design[2, 5] <- 0
> design[3, 4] <- 0
> design[4, 3] <- 0
> design[5, 2] <- 0
> design[6, 1] <- 0
> design
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]    1    0    0    1    1    0    1
[2,]    0    1    0    1    0    1    1
[3,]    0    0    1    0    1    1    1
[4,]    1    1    0    1    1    1    1
[5,]    1    0    1    1    1    1    1
[6,]    0    1    1    1    1    1    1
[7,]    1    1    1    1    1    1    1
```

```
> abc <- ginv(design) %*% all.regs(Y, X, gof = "Rsqu")[-1] * 100
```

[1] The library `hier.part` requires `gtools`, which is called *gregmisc* on CRAN.

For example, here the independent effect of variable 1 is described by one difference of the goodness-of-fit measures for two models: The full model (all three variables) and the model without variable 1.

The plot of the declared variance for each variable (independent of other variables) can be seen in the left panel of Figure 1.

```
> barplot(abc[1:3], names.arg = colnames(X), col = "grey", main = "Variation Partitioning")
```

2.3.2 Second method: Hierarchical Partitioning

Calculate regressions of y on all subsets of X

```
> hp.XY <- hier.part(Y, X, gof = "Rsqu")
> hp.XY
```

```
$gfs
[1] 0.0000000 0.3138482 0.8667001 0.5963617 0.8942909 0.8164685 0.9509413
[8] 1.0000000

$IJ
           I         J     Total
Y1 0.1622519 0.1515963 0.3138482
Y2 0.5059142 0.3607859 0.8667001
Y3 0.3318339 0.2645279 0.5963617

$I.perc
          I
Y1 16.22519
Y2 50.59142
Y3 33.18339
```

the `$gfs` part of which can be compared to

```
> all.regs(Y, X, gof = "Rsqu", print.vars = TRUE)
```

```
regressions done: formatting results
  variable.combination       gof
1                Theta 0.0000000
2                   Y1  0.3138482
3                   Y2  0.8667001
4                   Y3  0.5963617
5                Y1 Y2  0.8942909
6                Y1 Y3  0.8164685
7                Y2 Y3  0.9509413
8             Y1 Y2 Y3  1.0000000
```

For example, here the independent effect of variable 1 is described by a weighted average of four differences of the goodness-of-fit measures:

- the full model (all three variables) and the model with variable 2 and variable 3 (the only one used in variation partitioning)

- the model with variables 1 and 2 and the model with variable 2
- the model with variables 1 and 3 and the model with variable 3
- the model with variable 1.

Plot of `hp.XY$I.perc`

```
> barplot(hp.XY$I.perc[, 1], names.arg = colnames(X), col = "grey",
+     main = "Hierarchical Partitioning")
```

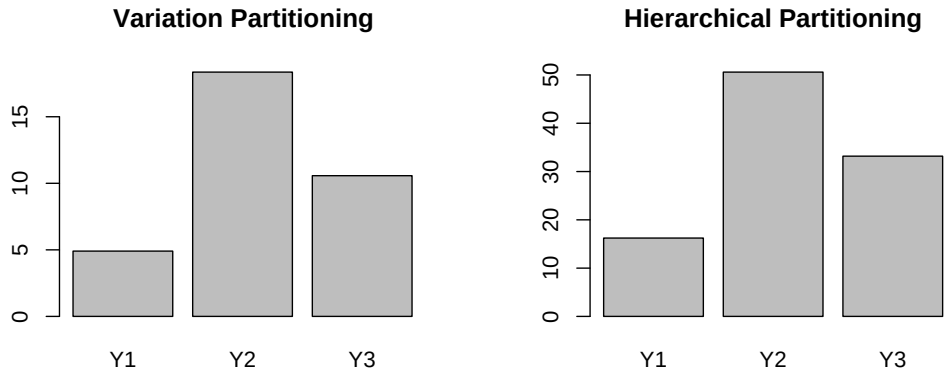


Figure 1: Declared variances for the two partitioning methods