A Guide to Scatterplot and Biplot Calibration

Jan Graffelman

Department of Statistics and Operations Research
Universitat Politècnica de Catalunya
Avinguda Diagonal 647, 08028 Barcelona, Spain.
email: jan.graffelman@upc.edu

1   Introduction

This guide gives detailed instructions on how to calibrate axes in scatterplots and biplots obtained in the statistical environment R [R Development Core Team (2004)] by using the package `calibrate`. By calibration we refer to the procedure of drawing a (linear) scale along an axis in a plot with tick marks and numeric labels. In an ordinary scatter plot of two variables x and y two calibrated perpendicular scales are typically automatically produced by the routine used for plotting the two variables. However, scatter plots can be extended with additional variables that are represented on oblique additional axes. The software described in this guide can be used to create calibrated scales on these oblique additional axes. Moreover, in a multivariate setting with more than two variables, raw data matrices, correlation matrices, contingency tables, regression coefficients, etc. are often represented graphically by means of biplots [Gabriel, 1971]. Biplots also contain oblique axes representing variables. The described software can also be used to construct scales on biplot axes.

The outline of this guide is as follows. In Section 2 we indicate how the R package `calibrate` can be installed. Section 3 describes in detail how to calibrate additional axes in scatter plots. Section 4 treats the calibration of biplot axes. Several subsections follow with detailed instructions of how to calibrate biplot axis in principal component analysis (PCA, Section 4.1), correspondence analysis (CA. Section 4.2), canonical correlation analysis (CCA, Section 4.3) and redundancy analysis (RDA, Section 4.4). The online documentation of the main routine for calibration `calibrate` is referenced in Section 5.

This guide does not provide the theory for the construction of scales on scatterplot and biplot axes. For a theoretical account of biplot calibration, we refer to Graffelman & van Eeuwijk (2005) and to Gower and Hand (1996). If you appreciate this software then please cite the following paper in your work:

Graffelman, J. & van Eeuwijk, F.A. (2005) Calibration of multivariate scatter plots for exploratory analysis of relations within and between sets of variables in genomic research *Biometrical Journal*, **47**(6) pp. 863-879. (clic here to access the paper)

2   Installation

Packages in R can be installed inside the program with the option "Packages"
in the main menu and then choosing "Install package" and picking the package
"calibrate". Typing:

```
> library(calibrate)
```

will, among others, make the function `calibrate, canocor` and `rda` available.
Several small data sets, also the ones used in this document, are included in the
package (`calves, goblets, heads, linnerud` and `storks`).

3   Calibration of Scatterplot axes

We consider a archaeological data set concerning 6 size measurements $(X_1, \ldots, X_6)$
on 25 goblets. This data was published by Manly (1989). The data can be loaded
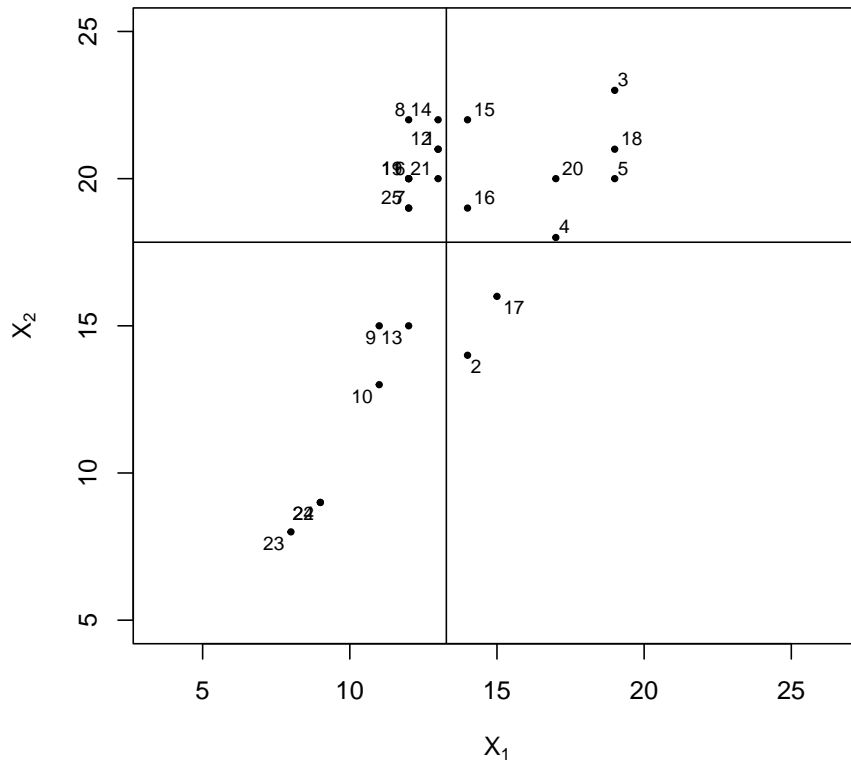with the `data` instruction.

```
> data(goblets)
> X <- goblets
```

Oblique additional axes in a scatterplot

We construct a scatterplot of X_1 versus X_2 and center a set of coordinate axes
on the point $(\bar{x}_1, \bar{x}_2)$ with the function `origin`.

```
> plot(X[,1],X[,2],pch=19,cex=0.5,xlab=expression(X[1]),ylab=expression(X[2]),
+      xlim=c(5,25),ylim=c(5,25),asp=1)
> m <- apply(X[,1:2],2,mean)
> textxy(X[,1],X[,2],1:25,m=m,cex=0.75)
> origin(m)
```

Next, we perform the regression of X_5 onto X_1 and X_2 (all variables being centered) in order to obtain an additional axis for X_5. We represent X_5 in the plot as a simple arrow whose coordinates are given by the regression coefficients:

```
> Xc <- scale(X,center=TRUE,scale=FALSE)
> b <- solve(t(Xc[,1:2])%*%Xc[,1:2])%*%t(Xc[,1:2])%*%Xc[,5]
> print(b)

        [,1]
X1 0.3850425
X2 0.1225419

> bscaled <- 20*b
> arrows(m[1],m[2],m[1]+bscaled[1],m[2]+bscaled[2],col="blue",length=0.1)
> arrows(m[1],m[2],m[1]-bscaled[1],m[2]-bscaled[2],length=0,lty="dashed",col="blue")
```

3
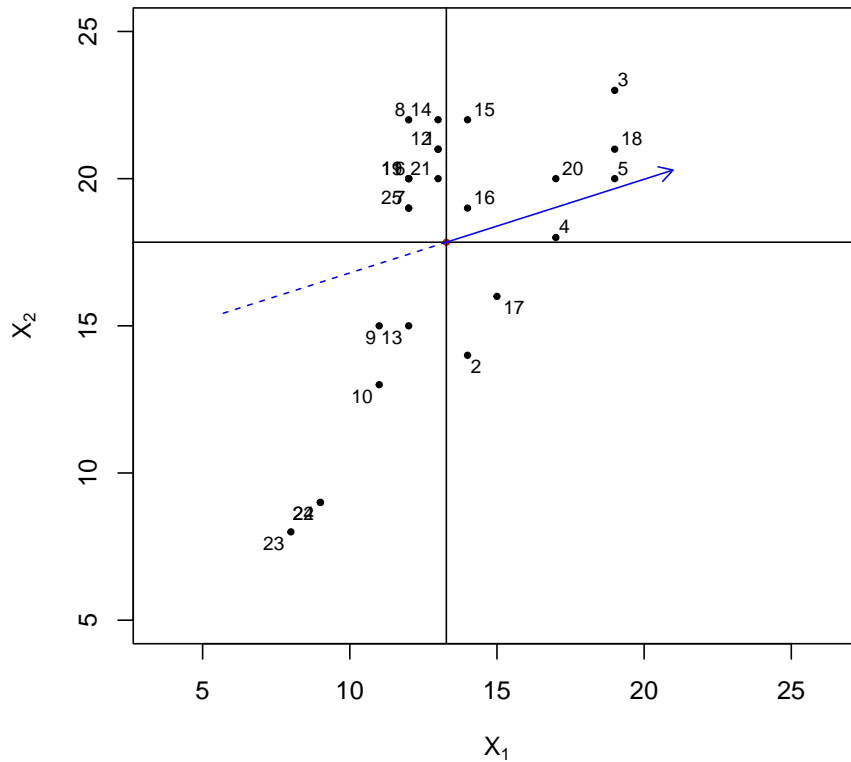
A direction that is optimal in the least squares sense for X_5 is given by the vector of regression coefficients [Graffelman and Aluja-Banet (2003)]. To make this direction more visible, we multiplied it by a constant (20). It is clear that the direction of increase for X_5 runs approximately North-East across the scatterplot. We now proceed to calibrate this direction with a scale for X_5. In order to choose sensible values for the scale of X_5, we first inspect the range of variation of X_5, and then choose a set of values we want to mark off on the scale (`tm`) and also compute the deviations of these values from the mean (`tmc`). We specify a tick length of 0.3 (`tl=0.3`). Depending on the data, some values of `tl` typically have to be tried to see how to obtain a nice scale.

```
> print(range(X[,5]))

[1]  2 11

> yc <- scale(X[,5],scale=FALSE)
> tm <- seq(2,10,by=1)
> tmc <- tm - mean(X[,5])
> Calibrate.X5<-calibrate(b,yc,tmc,Xc[,1:2],tmlab=tm,m=m,tl=0.3,axislab="X_5",
+                           labpos=4,cex.axislab=1)

---------- Calibration Results for  X_5  --------------------
Length of 1 unit of the original variable =  2.4748
```

4

```
Angle                              =  17.65 degrees
Optimal calibration factor         =  6.1247
Used calibration factor            =  6.1247
Goodness-of-fit                    =  0.5133
Goodness-of-scale                  =  0.5133
------------------------------------------------------------
```
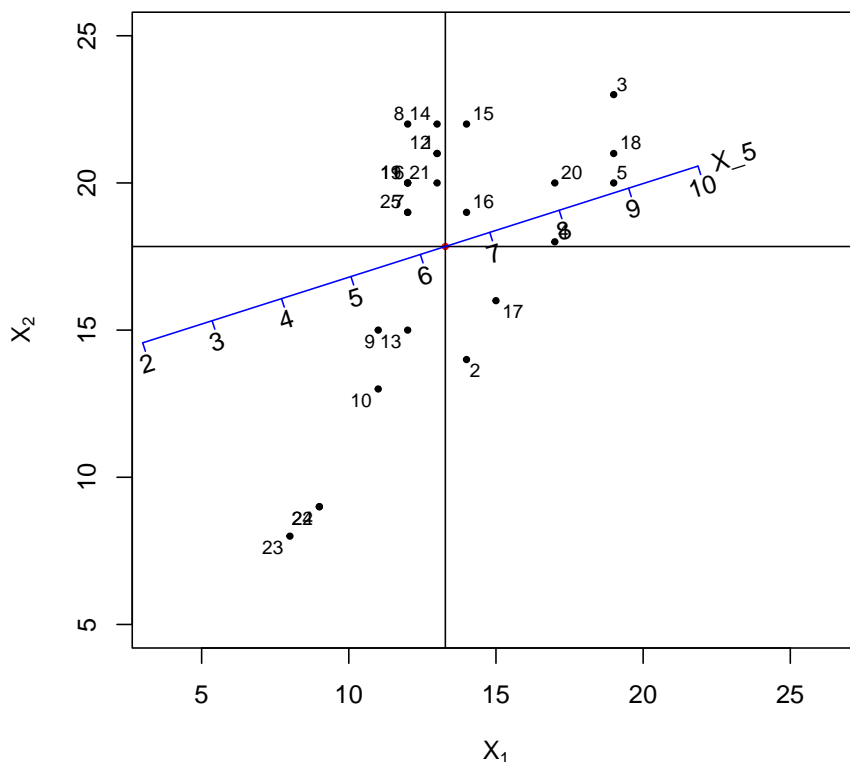

The numerical output from routine `calibrate` shows that one unit along the axis for X_5 occupies 2.47 units in the plotting frame. The axis for X_5 makes an angle of 17.65 degrees with the positive x-axis. The calibration factor is 6.12. Multiplying the vector of regressions coefficients by this factor yields a vector that represents a unit change in the scale of X_5. E.g. for this data we have that the vector $6.12 \cdot (0.385, 0.123) = (2.358, 0.751)$ represents a unit change. This vector has norm $\sqrt{2.358^2 + 0.751^2} = 2.47$. Other calibration factors may be specified by using parameter `alpha`. If `alpha` is left unspecified the optimal value computed by least squares will be used. The goodness-of-fit of X_5 is 0.513. This means that 51.3% of the variance of X_5 can be explained by a regression onto X_1 and X_2 ($R^2 = 0.513$). The goodness-of-scale has the same value. The goodness-of-scale is only relevant if we modify parameter `alpha`. `Calibrate.X5` is a list object containing all calibration results (calibration factor, fitted values according to the scale used, tick marker positions, etc.).

Shifting a calibrated axis

Using many calibrated axes in a plot, all passing through the origin, leads to dense plots that become unreadable. It is therefore a good idea to shift calibrated axes towards the margins of the plot. This keeps the central cloud of data points clear and relegates all information on scales to the margins of the graph. There are two natural positions for a shifted axis: just above the largest data point in a direction perpendicular to the axis being calibrated, or just below the smallest data point in the perpendicular direction. The arguments `shiftdir`, `shiftfactor` and `shiftvec` can be used to control the shifting of a calibrated axis. `shiftvec` allows the user to specify the shift vector manually. This is normally not needed, and good positions for an axis can be found by using only `shiftdir` and `shiftfactor`. Argument `shiftdir` can be set to `'right'` or `'left'` and indicates in which direction the axis is to be shifted, with respect to the direction of increase of the calibrated axis. Setting `shiftdir` shifts the axis automatically just above or below the most outlying data point in the direction perpendicular to the vector being calibrated. In order to move the calibrated axis farther out or to pull it more in, `shiftfactor` can be used. Argument `shiftfactor` stretches or shrinks the shift vector for the axis. A `shiftfactor` larger than 1 moves the axis outwards, and a `shiftfactor` smaller than 1 pulls the axis towards the origin of the plot. If set to 1 exactly, the shifted axis will cut through the most outlying data point. The default `shiftfactor` is 1.05. We redo the previous plot, shifting the calibrated axis below the cloud of points, which is to the right w.r.t. the direction of increase of the variable.

```
> yc <- scale(X[,5],scale=FALSE)
> tm <- seq(2,10,by=1)
> tmc <- tm - mean(X[,5])
> Calibrate.X5<-calibrate(b,yc,tmc,Xc[,1:2],tmlab=tm,m=m,tl=0.3,axislab="X_5",labpos=4,
+                         cex.axislab=1,shiftdir="right")

---------- Calibration Results for  X_5  --------------------
Length of 1 unit of the original variable =  2.4748
Angle                                     =  17.65 degrees
Optimal calibration factor                =  6.1247
Used calibration factor                   =  6.1247
Goodness-of-fit                           =  0.5133
Goodness-of-scale                         =  0.5133
-------------------------------------------------------------
```
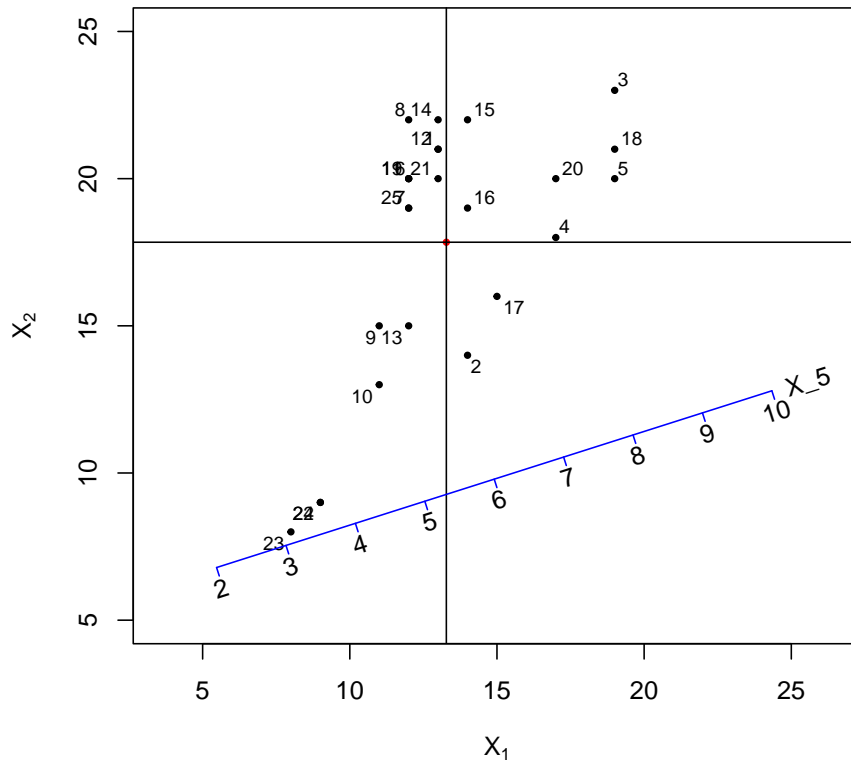
The shift of the axis does not affect the interpretation of the plot, because the projections of the points onto the axis remain the same.

Second vertical axis in a scatterplot

The oblique direction in the previous section is the preferred direction for X_5, as this direction is optimal in the least squares sense. However, if desired, additional variables can also be represented as a second vertical axis on the right of the plot, or as a second horizontal axis on the top of the plot. We now proceed to construct a second vertical axis on the right hand of the scatter plot for X_5. This can be done by setting the vector to be calibrated (first argument of routine `calibrate`) to the (0,1) vector. By specifying a shiftvector explicitly (`shiftvec`), the axis can be shifted. For this data, setting `shiftvec` to `c(par('usr')[2]-mean(X[,1]),0)` and `shiftfactor = 1`, makes the axis coincide with the right vertical borderline of the graph.

```
> opar <- par('xpd'=TRUE)
> tm <- seq(3,8,by=1)
> tmc <- (tm - mean(X[,5]))
> Calibrate.rightmargin.X5 <- calibrate(c(0,1),yc,tmc,Xc[,1:2],tmlab=tm,m=m,
+                                       axislab="X_5",tl=0.5,
```

```
+                                     shiftvec=c(par('usr')[2]-mean(X[,1]),0),
+                                     shiftfactor=1,where=2,
+                                     laboffset=c(1.5,1.5),cex.axislab=1)

---------- Calibration Results for  X_5  --------------------
Length of 1 unit of the original variable =  3.4603
Angle                                     =  90 degrees
Optimal calibration factor                =  3.4603
Used calibration factor                   =  3.4603
Goodness-of-fit                           =  0.3373
Goodness-of-scale                         =  0.3373
------------------------------------------------------------

> par(opar)
```
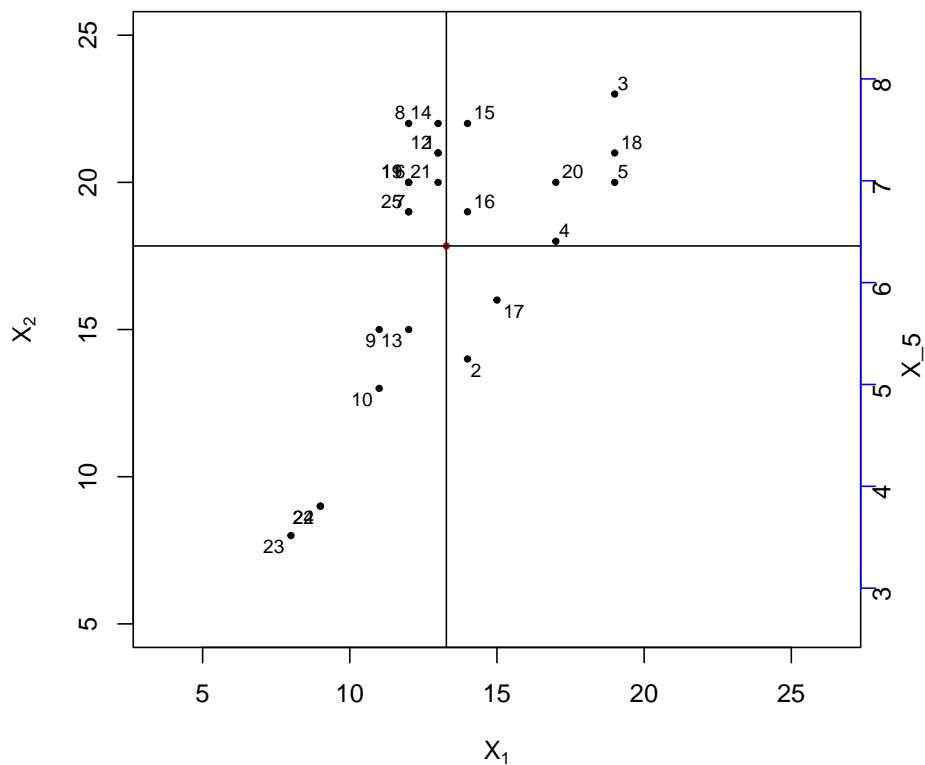

The second vertical axis has calibration factor 3.46, and a goodness of fit of 0.34. The fit of the variable is worse in comparison with the previous oblique direction given by the regression coefficients. Note that graphical clipping in temporarily turned off (par('xpd'=TRUE)) to allow the calibration routine to draw ticks and labels outside the figure region, and that the range of the tick marks was shortened in order not surpass the figure region.
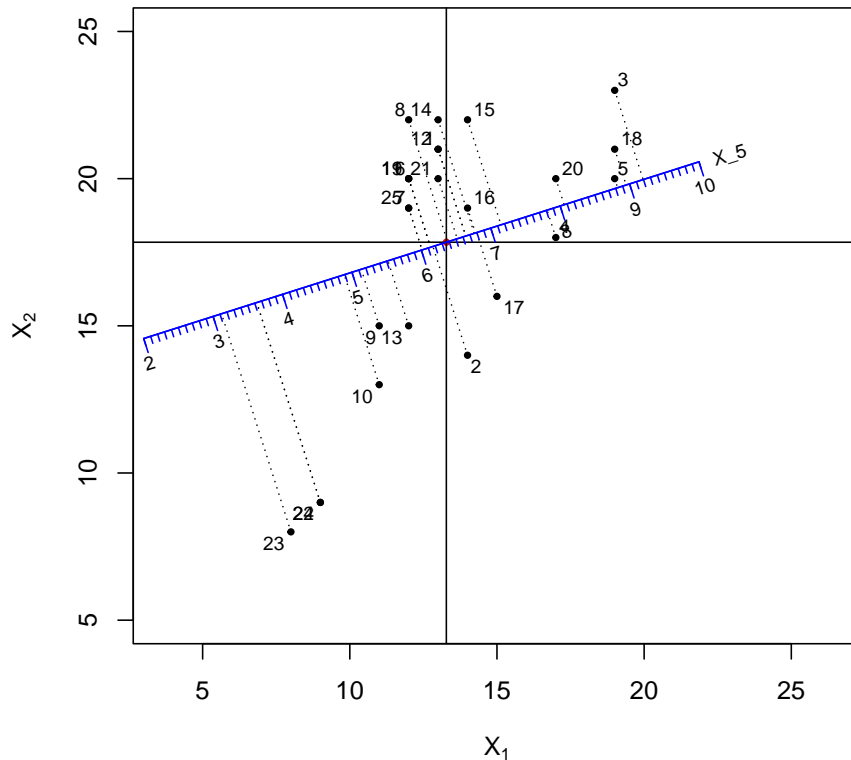
Subscales and double calibrations

Scales with tick marks can be refined by drawing subscales with smaller tick marks. E.g. larger labelled tickmarks can be used to represent multiples of 10, and small unlabelled tick marks can be used to represent units. The subscale allows a more precise recovery of the data values. This can simply be achieved by calling the calibration routine twice, once with a coarse sequence and once with a finer sequence. For the second call one can specify `verb=FALSE` in order to suppress the numerical output of the routine, and `lm=FALSE` to supress the tick mark labels under the smaller ticks. The tickmarks for the finer scale are made smaller by modifying the tick length (e.g. `tl=0.1`). Depending on the data, some trial and error with different values for `tl` may be necessary before nice scales are obtained. This may be automatized in the future. Finally, reading off the (approximate) data values can further be enhanced by drawing perpendiculars from the points to the calibrated axis by setting `dp=TRUE`.

```
> tm <- seq(2,10,by=1)
> tmc <- (tm - mean(X[,5]))
> Calibrate.X5 <- calibrate(b,yc,tmc,Xc[,1:2],tmlab=tm,m=m,axislab="X_5",tl=0.5,
+                             dp=TRUE,labpos=4)

---------- Calibration Results for  X_5  -------------------
Length of 1 unit of the original variable =  2.4748
Angle                                     =  17.65 degrees
Optimal calibration factor                =  6.1247
Used calibration factor                   =  6.1247
Goodness-of-fit                           =  0.5133
Goodness-of-scale                         =  0.5133
------------------------------------------------------------

> tm <- seq(2,10,by=0.1)
> tmc <- (tm - mean(X[,5]))
> Calibrate.X5 <- calibrate(b,yc,tmc,Xc[,1:2],tmlab=tm,m=m,tl=0.25,verb=FALSE,
+                             lm=FALSE)
```

A *double calibration* can be created by drawing two scales, one on each side of the axis. Double calibrations can be useful. For instance, one scale can be used for recovery of the original data values of the variable, whereas the second scale can be used for recovery of standardized values or of correlations with other variables. Double calibrations can also be used to graphically verify if two different calibration procedures give the same result or not.

Recalibrating the original scatterplot axes

By calibrating the (0,1) and (1,0) vectors the original axes of the scatter plot can be redesigned. We illustrate the recalibration of the original axes by creating a second scale on the other side of the axes, a refined scale for X_1, and a scale for the standardized data for X_2. For the latter calibration one unit equals one standard deviation.

```
> opar <- par('xpd'=TRUE)
> tm <- seq(5,25,by=5)
> tmc <- (tm - mean(X[,1]))
> yc <- scale(X[,1],scale=FALSE)
> Calibrate.X1 <- calibrate(c(1,0),yc,tmc,Xc[,1:2],tmlab=tm,m=m,tl=0.5,
+                           axislab="X_1",cex.axislab=1,showlabel=FALSE,
+ shiftvec=c(0,-(m[2]-par("usr")[3])),shiftfactor=1,reverse=TRUE)
```

```
---------- Calibration Results for  X_1  -------------------
Length of 1 unit of the original variable =  1
Angle                                     =  0 degrees
Optimal calibration factor                =  1
Used calibration factor                   =  1
Goodness-of-fit                           =  1
Goodness-of-scale                         =  1
------------------------------------------------------------

> tm <- seq(5,25,by=1); tmc <- (tm - mean(X[,1]))
> Calibrate.X1 <- calibrate(c(1,0),yc,tmc,Xc[,1:2],tmlab=tm,m=m,tl=0.25,
+                           axislab="X_1",cex.axislab=1,showlabel=FALSE,
+ shiftvec=c(0,-(m[2]-par("usr")[3])),shiftfactor=1,reverse=TRUE,
+                           verb=FALSE,lm=FALSE)
> yc <- scale(X[,2],scale=TRUE)
> tm <- seq(-3,1,by=1)
> Calibrate.X2 <- calibrate(c(0,1),yc,tm,Xc[,1:2],tmlab=tm,m=m,tl=0.6,
+                           axislab="X_2",cex.axislab=1,showlabel=FALSE,
+ shiftvec=c(-(mean(X[,1])-par('usr')[1]),0),shiftfactor=1,verb=TRUE,lm=TRUE)

---------- Calibration Results for  X_2  -------------------
Length of 1 unit of the original variable =  4.3367
Angle                                     =  90 degrees
Optimal calibration factor                =  4.3367
Used calibration factor                   =  4.3367
Goodness-of-fit                           =  1
Goodness-of-scale                         =  1
------------------------------------------------------------

> tm <- seq(-3,1.5,by=0.1)
> Calibrate.X2 <- calibrate(c(0,1),yc,tm,Xc[,1:2],tmlab=tm,m=m,tl=0.3,
+                           axislab="X_2",cex.axislab=1,showlabel=FALSE,
+ shiftvec=c(-(mean(X[,1])-par('usr')[1]),0),shiftfactor=1,verb=FALSE,lm=FALSE)
> par(opar)
```
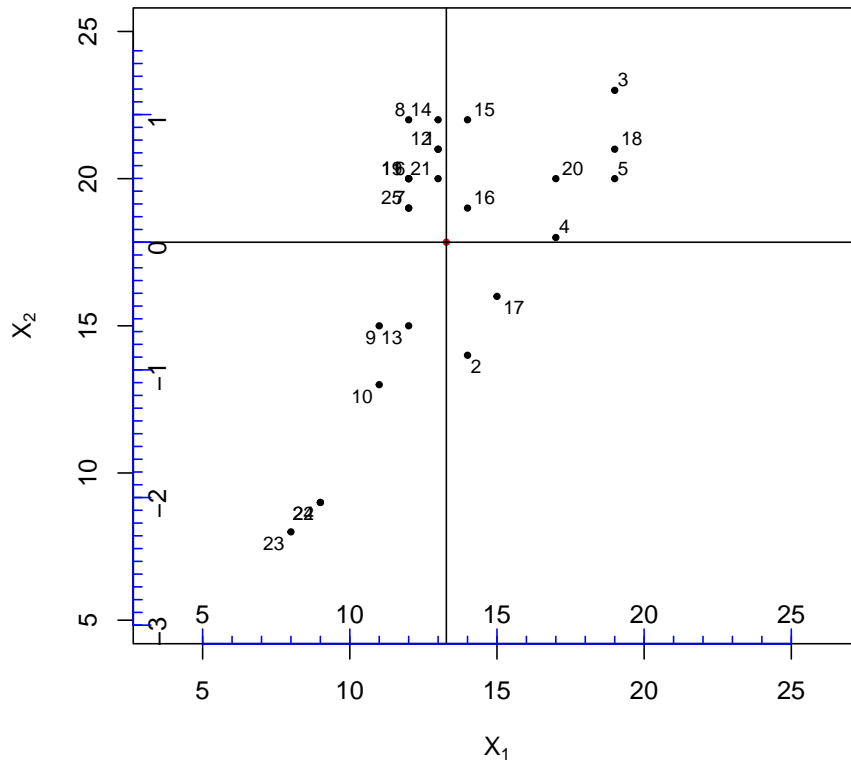
4 Calibration of Biplot axes

In this section we give detailed instructions on how to calibrate biplot axes. We will consider biplots of raw data matrices and correlation matrices obtained by PCA, biplots of profiles obtained in CA, biplots of data matrices and correlation matrices (in particular the between-set correlation matrix) in CCA and biplots of fitted values and regression coefficients obtained by RDA. In principle, calibration of biplot axes has little additional complication in comparison with the calibration of additional axes in scatterplots explained above. The main issue is that, prior to calling the calibration routine, one needs to take care of the proper centring and standardisation of the tick marks.

4.1 Principal component analysis

Principal component analysis can be performed by using routine `princomp` from the `stats` library. We use again Manly's goblets data to create a biplot of the data based on a PCA of the covariance matrix. We use `princomp` to compute the scores for the rows and the columns of the data matrix. The first principal component is seen to be a size component, separating the smaller goblets on

12

the right from the larger goblets on the left. The variable vectors are multiplied by a factor of 15 to facilitate interpretation. Next we calibrate the vector for X_3, using labelled tickmarks for multiples of 5 units, and shorter unlabelled tickmarks for the units. The goodness of fit of X_3 is very high (0.99), which means that X_3 is close to perfectly represented. Calibrate.X3 is a list object containing the numerical results of the calibration.

```
> # PCA and Biplot construction
> pca.results <- princomp(X,cor=FALSE)
> Fp <- pca.results$scores
> Gs <- pca.results$loadings
> plot(Fp[,1],Fp[,2],pch=16,asp=1,xlab="PC 1",ylab="PC 2",cex=0.5)
> textxy(Fp[,1],Fp[,2],rownames(X),cex=0.75)
> arrows(0,0,15*Gs[,1],15*Gs[,2],length=0.1)
> textxy(15*Gs[,1],15*Gs[,2],colnames(X),cex=0.75)
> # Calibration of X_3
> ticklab  <- seq(5,30,by=5)
> ticklabc <- ticklab-mean(X[,3])
> yc <- (X[,3]-mean(X[,3]))
> g <- Gs[3,1:2]
> Calibrate.X3 <- calibrate(g,yc,ticklabc,Fp[,1:2],ticklab,tl=0.5,
+                           axislab="X3",cex.axislab=0.75,where=1,labpos=4)

---------- Calibration Results for  X3  --------------------
Length of 1 unit of the original variable =  1.1813
Angle                                     = -39.28 degrees
Optimal calibration factor                =  1.3954
Used calibration factor                   =  1.3954
Goodness-of-fit                           =  0.9914
Goodness-of-scale                         =  0.9914
------------------------------------------------------------

> ticklab  <- seq(5,30,by=1)
> ticklabc <- ticklab-mean(X[,3])
> Calibrate.X3.fine <- calibrate(g,yc,ticklabc,Fp[,1:2],ticklab,lm=FALSE,tl=0.25,
+                           verb=FALSE,cex.axislab=0.75,where=1,labpos=4)
```
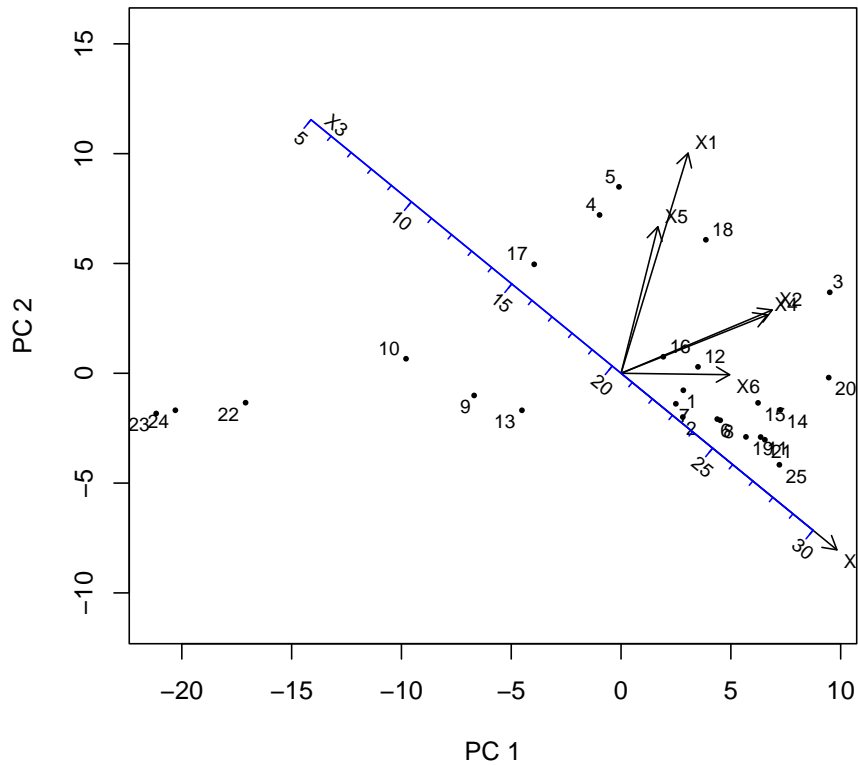
We do a PCA based on the correlation matrix, and proceed to construct a biplot of the correlation matrix. The correlations of X_5 with the other variables are computed, and the biplot axis for X_5 is calibrated with a correlation scale. Routine `calibrate` is repeatedly called to create finer subscales.

```
> # PCA and Biplot construction
> pca.results <- princomp(X,cor=TRUE)
> Fp <- pca.results$scores
> Ds <- diag(pca.results$sdev)
> Fs <- Fp%*%solve(Ds)
> Gs <- pca.results$loadings
> Gp <- Gs%*%Ds
> #plot(Fs[,1],Fs[,2],pch=16,asp=1,xlab="PC 1",ylab="PC 2",cex=0.5)
> #textxy(Fs[,1],Fs[,2],rownames(X))
> plot(Gp[,1],Gp[,2],pch=16,cex=0.5,xlim=c(-1,1),ylim=c(-1,1),asp=1,
+       xlab="1st principal axis",ylab="2nd principal axis")
> arrows(0,0,Gp[,1],Gp[,2],length=0.1)
> textxy(Gp[,1],Gp[,2],colnames(X),cex=0.75)
> ticklab <- c(seq(-1,-0.2,by=0.2),seq(0.2,1.0,by=0.2))
> R <- cor(X)
> y <- R[,5]
```

```
> g <- Gp[5,1:2]
> Calibrate.X5 <- calibrate(g,y,ticklab,Gp[,1:2],ticklab,lm=TRUE,tl=0.05,dp=TRUE,
+                           labpos=2,cex.axislab=0.75,axislab="X_5")

---------- Calibration Results for  X_5  --------------------
Length of 1 unit of the original variable =  1.0634
Angle                                     =  49.36 degrees
Optimal calibration factor                =  1.1308
Used calibration factor                   =  1.1308
Goodness-of-fit                           =  0.9824
Goodness-of-scale                         =  0.9824
-------------------------------------------------------------

> ticklab <- seq(-1,1,by=0.1)
> Calibrate.X5 <- calibrate(g,y,ticklab,Gp[,1:2],ticklab,lm=FALSE,tl=0.05,verb=FALSE)
> ticklab <- seq(-1,1,by=0.01)
> Calibrate.X5 <- calibrate(g,y,ticklab,Gp[,1:2],ticklab,lm=FALSE,tl=0.025,verb=FALSE)
```
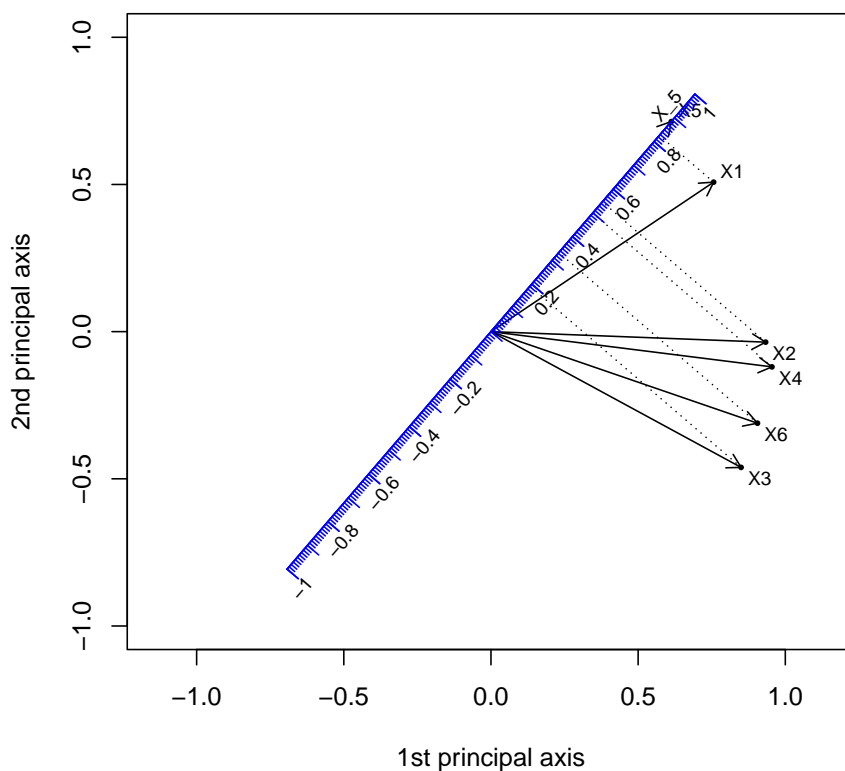

The goodness of fit of the representation of the correlations of X_5 with the other variables is 0.98, the 6 correlations being close to perfectly represented. We compute the sample correlation matrix and compare the observed correlations of X_5 with those estimated from the calibrated biplot axis (yt). Note that

15

PCA also tries to approximate the correlation of a variable with itself, and that the arrow on representing X_5 falls short of the value 1 on its own calibrated scale. The refined subscale allows very precise graphical representation of the correlations as estimated by the biplot.

```
> print(R)

           X1        X2        X3        X4        X5
X1 1.0000000 0.6234051 0.3464089 0.6748429 0.6901040
X2 0.6234051 1.0000000 0.8392292 0.8287898 0.5807725
X3 0.3464089 0.8392292 1.0000000 0.8430518 0.2511584
X4 0.6748429 0.8287898 0.8430518 1.0000000 0.4874610
X5 0.6901040 0.5807725 0.2511584 0.4874610 1.0000000
X6 0.5875703 0.7970192 0.8575089 0.9101886 0.2885165
           X6
X1 0.5875703
X2 0.7970192
X3 0.8575089
X4 0.9101886
X5 0.2885165
X6 1.0000000

> print(cbind(R[,5],Calibrate.X5$yt))

        [,1]      [,2]
X1 0.6901040 0.8257486
X2 0.5807725 0.5462001
X3 0.2511584 0.1914136
X4 0.4874610 0.4992765
X5 1.0000000 0.8843474
X6 0.2885165 0.3326711
```

4.2   Correspondence analysis

We consider a contingency table of a sample of Dutch calves born in the late nineties, shown in Table 1. A total of 7257 calves were classified according to two categorical variables: the method of production (ET = Embryo Transfer, IVP = In Vitro Production, AI = Artificial Insemination) and the ease of delivery, scored on a scale from 1 (normal) to 6 (very heavy). The data in Table 1 were provided by Holland Genetics.

Ease of delivery	Type of calf		
	ET	IVP	AI
1	97	150	1686
2	152	183	1339
3	377	249	1209
4	335	227	656
5	42	136	277
6	9	71	62

Table 1: Calves data from Holland Genetics.

For this contingency table we obtain $\chi^2_{10} = 833.16$ with $p < 0.001$ and the null hypothesis of no association between ease of delivery and type of calf has to be rejected. However, what is the precise nature of this association? Correspondence analysis can be used to gain insight in the nature of this association. We use routine `corresp` form the `MASS` library [Venables and Ripley (2002)] to perform correspondence analysis and to obtain the coordinates for a biplot of the row profiles. We compute the row profiles and then repeatedly call the calibration routine, each time with a different set of `ticklabs`.

```
> library(MASS)
> data(calves)
> ca.results <- corresp(calves,nf=2)
> Fs <- ca.results$rscore
> Gs <- ca.results$cscore
> Ds <- diag(ca.results$cor)
> Fp <- Fs%*%Ds
> Gp <- Gs%*%Ds
> plot(Gs[,1],Gs[,2],pch=16,asp=1,cex=0.5,xlab="1st principal axis",
+       ylab="2nd principal axis")
> textxy(Gs[,1],Gs[,2],colnames(calves),cex=0.75)
> points(Fp[,1],Fp[,2],pch=16,cex=0.5)
> textxy(Fp[,1],Fp[,2],rownames(calves),cex=0.75)
> origin()
> arrows(0,0,Gs[,1],Gs[,2])
> P <- as.matrix(calves/sum(calves))
> r <- apply(P,1,sum)
> k <- apply(P,2,sum)
> Dc <- diag(k)
> Dr <- diag(r)
> RP <- solve(Dr)%*%P
> print(RP)

            ET         IVP          AI
[1,] 0.05018107 0.07759959 0.8722193
[2,] 0.09080048 0.10931900 0.7998805
[3,] 0.20544959 0.13569482 0.6588556
[4,] 0.27504105 0.18637110 0.5385878
[5,] 0.09230769 0.29890110 0.6087912
[6,] 0.06338028 0.50000000 0.4366197

> CRP <- RP - ones(nrow(RP), 1) %*% t(k)
> TCRP <- CRP%*%solve(Dc)
> y <- TCRP[,3]
> g <- Gs[3,1:2]
> ticklab  <- c(0,seq(0,1,by=0.2))
> ticklabs <- (ticklab - k[3])/k[3]
> Calibrate.AI <- calibrate(g,y,ticklabs,Fp[,1:2],ticklab,lm=TRUE,tl=0.10,
+                           weights=Dr,axislab="AI",labpos=4,dp=TRUE)

---------- Calibration Results for  AI  --------------------
Length of 1 unit of the original variable =  1.6057
```

17

```
Angle                              =  -6.82 degrees
Optimal calibration factor         =  2.5784
Used calibration factor            =  2.5784
Goodness-of-fit                    =  1
Goodness-of-scale                  =  1
------------------------------------------------------------

> ticklab  <- c(0,seq(0,1,by=0.1))
> ticklabs <- (ticklab - k[3])/k[3]
> Calibrate.AI <- calibrate(g,y,ticklabs,Fp[,1:2],ticklab,lm=FALSE,tl=0.10,
+                           weights=Dr,verb=FALSE)
> ticklab  <- c(0,seq(0,1,by=0.01))
> ticklabs <- (ticklab - k[3])/k[3]
> Calibrate.AI <- calibrate(g,y,ticklabs,Fp[,1:2],ticklab,lm=FALSE,tl=0.05,
+                           weights=Dr,verb=FALSE)
```
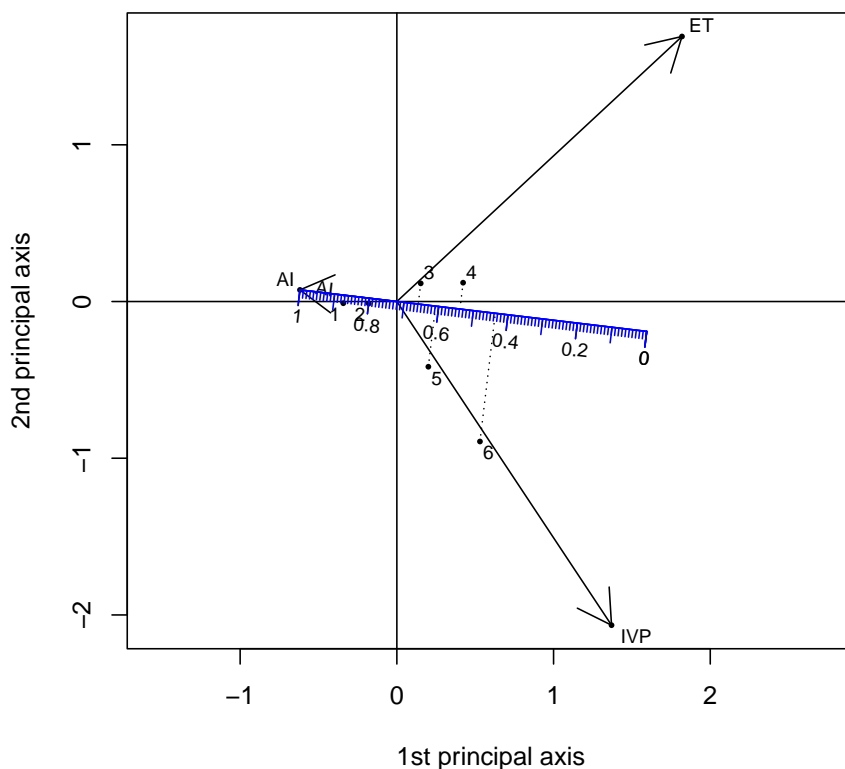

Because the calibration is done by weighted least squares, a diagonal matrix of
weights (`weights=Dr`) is supplied as a parameter to the calibration routine Note
that the calibrated axis for the row profiles with respect to AI has goodness of
fit 1. This is due to the fact that the rank of the matrix of centred profiles is two,
and that therefore all profiles can be perfectly represented in two dimensional
space.

4.3 Canonical correlation analysis

We consider a classical data set on the head sizes of the first and the second son of 25 families [Frets (1921)]. These data have been analysed by several authors [Anderson (1984), Mardia et al.(1979), Graffelman (2005)] We first load the data and perform a canonical correlation analysis, using supplied function `canocor` (a more fully fledged program for canonical correlation analysis in comparison with `cancor` from the `stats` package).

```
> data(heads)
> X  <- cbind(heads$X1,heads$X2)
> Y  <- cbind(heads$Y1,heads$Y2)
> Rxy<- cor(X,Y)
> Ryx<- t(Rxy)
> Rxx<- cor(X)
> Ryy<- cor(Y)
> cca.results <-canocor(X,Y)
> plot(cca.results$Gs[,1],cca.results$Gs[,2],pch=16,asp=1,xlim=c(-1,1),ylim=c(-1,1),
+       xlab=expression(V[1]),ylab=expression(V[2]))
> arrows(0,0,cca.results$Fp[,1],cca.results$Fp[,2],length=0.1)
> arrows(0,0,cca.results$Gs[,1],cca.results$Gs[,2],length=0.1)
> textxy(cca.results$Fp[1,1],cca.results$Fp[1,2],expression(X[1]),cex=0.75)
> textxy(cca.results$Fp[2,1],cca.results$Fp[2,2],expression(X[2]),cex=0.75)
> textxy(cca.results$Gs[1,1],cca.results$Gs[1,2],expression(Y[1]),cex=0.75)
> textxy(cca.results$Gs[2,1],cca.results$Gs[2,2],expression(Y[2]),cex=0.75)
> circle(1)
> ticklab  <- seq(-1,1,by=0.2)
> y <- Rxy[,2]
> g <- cca.results$Gs[2,1:2]
> Cal.Cor.Y2 <- calibrate(g,y,ticklab,cca.results$Fp[,1:2],ticklab,lm=TRUE,tl=0.05,
+                         dp=TRUE,reverse=TRUE,weights=solve(Rxx),
+ axislab="Y_2",cex.axislab=0.75,showlabel=FALSE)


---------- Calibration Results for  Y_2  --------------------
Length of 1 unit of the original variable =  1
Angle                                     =  -15.92 degrees
Optimal calibration factor                =  1
Used calibration factor                   =  1
Goodness-of-fit                           =  1
Goodness-of-scale                         =  1
------------------------------------------------------------
```
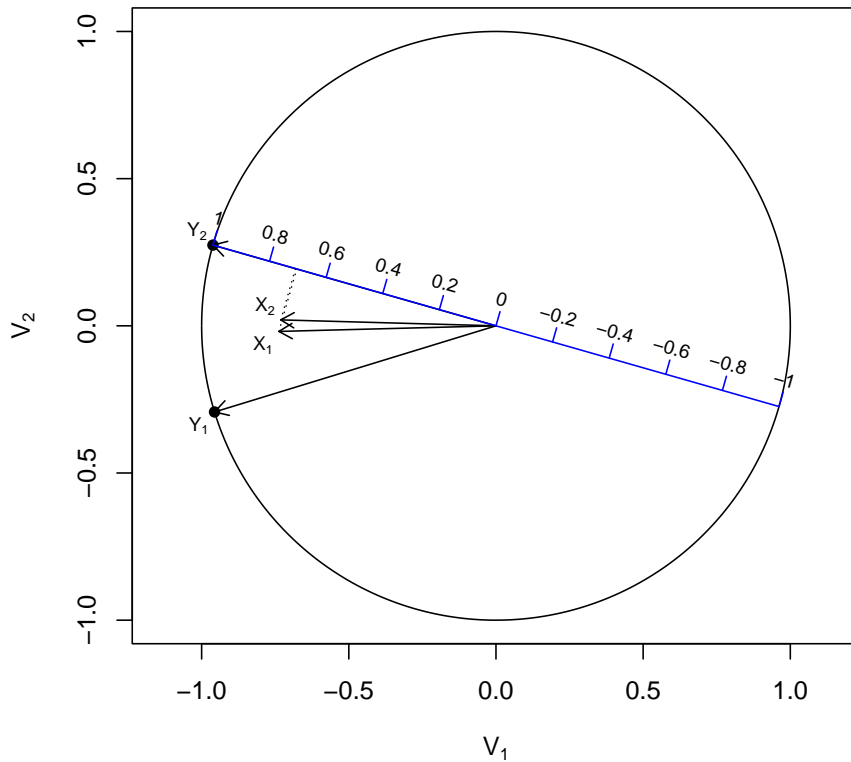
```
> plot(cca.results$Gs[,1],cca.results$Gs[,2],pch=16,asp=1,xlim=c(-2,2),ylim=c(-2,2),
+       xlab=expression(V[1]),ylab=expression(V[2]))
> #arrows(0,0,cca.results$Fp[,1],cca.results$Fp[,2],length=0.1)
> #arrows(0,0,cca.results$Gs[,1],cca.results$Gs[,2],length=0.1)
>
> textxy(cca.results$Fp[1,1],cca.results$Fp[1,2],expression(X[1]))
> textxy(cca.results$Fp[2,1],cca.results$Fp[2,2],expression(X[2]))
> textxy(cca.results$Gs[1,1],cca.results$Gs[1,2],expression(Y[1]))
> textxy(cca.results$Gs[2,1],cca.results$Gs[2,2],expression(Y[2]))
> points(cca.results$V[,1],cca.results$V[,2],pch=16,cex=0.5)
> textxy(cca.results$V[,1],cca.results$V[,2],1:nrow(X),cex=0.75)
> ticklab  <- seq(135,160,by=5)
> ticklabc <- ticklab-mean(Y[,2])
> ticklabs <- (ticklab-mean(Y[,2]))/sqrt(var(Y[,2]))
> y <- (Y[,2]-mean(Y[,2]))/sqrt(var(Y[,2]))
> Fr <- cca.results$V[,1:2]
> g <- cca.results$Gs[2,1:2]
> #points(cca.results$V[,1],cca.results$V[,2],cex=0.5,pch=19,col="red")
> #textxy(cca.results$V[,1],cca.results$V[,2],rownames(Xn))
>
```

20

```
> Cal.Data.Y2 <- calibrate(g,y,ticklabs,Fr,ticklab,lm=TRUE,tl=0.1,dp=TRUE,
+                          reverse=TRUE,verb=TRUE,axislab="Y_2",
+                          cex.axislab=0.75,showlabel=FALSE)

---------- Calibration Results for  Y_2  --------------------
Length of 1 unit of the original variable =  1
Angle                                     = -15.92 degrees
Optimal calibration factor                =  1
Used calibration factor                   =  1
Goodness-of-fit                           =  1
Goodness-of-scale                         =  1
-------------------------------------------------------------

>
> #cca.results<-lm.gls(Rxy[,5]~-1+Fr,W=solve(Rxx))
>
```
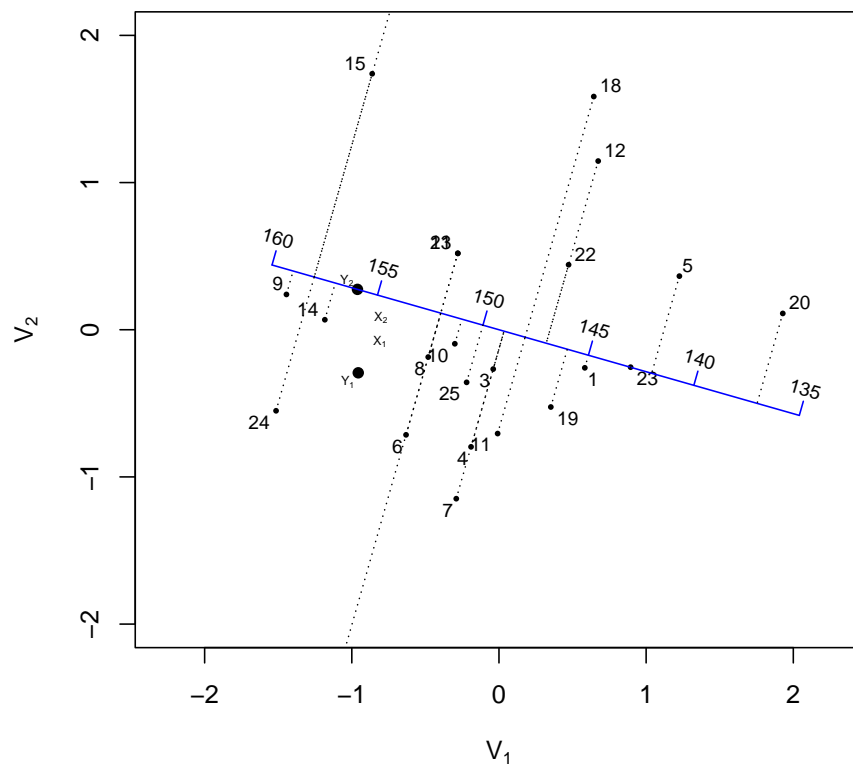

We construct the biplot of the between-set correlation matrix (the joint plot of $\mathbf{F}_p$ and $\mathbf{G}_s$). Firstly we calibrate the biplot axis for Y_2 with a correlation scale. This calibration is done by generalised least squares with the inverse of the correlation matrix of the X-variables as a weight matrix (`weights=solve(Rxx)`). Secondly, we calibrate the biplot axis for Y_2 with a scale for the original values.

This second calibration has no weight matrix and is obtained by ordinary least squares. Both calibrations have a goodness of fit of 1 and allow perfect recovery of correlations and original data values.

4.4 Redundancy analysis

Redundancy analysis can be seen as a constrained PCA. It allows two biplots, the biplot of the fitted values and a biplot of regression coefficients. Function `rda` of the package provides a routine for redundancy analysis. We use Linnerud's data on physical exercise and body measurement variables [Tenenhaus (1998)] to illustrate calibrated biplots in redundancy analysis.

```
> data(linnerud)
> X <- linnerud[,1:3]
> Y <- linnerud[,4:6]
> rda.results <- rda(X,Y)
> plot(rda.results$Fs[,1],rda.results$Fs[,2],pch=16,asp=1,xlim=c(-2,2),ylim=c(-2,2),
+      cex=0.5,xlab="1st principal axis",ylab="2nd principal axis")
> arrows(0,0,2*rda.results$Gyp[,1],2*rda.results$Gyp[,2],length=0.1)
> textxy(rda.results$Fs[,1],rda.results$Fs[,2],rownames(X),cex=0.75)
> textxy(2*rda.results$Gyp[,1],2*rda.results$Gyp[,2],colnames(Y),cex=0.75)
> y <- rda.results$Yh[,3]
> g <- rda.results$Gyp[3,1:2]
> Fr <- rda.results$Fs[,1:2]
> ticklab <- c(seq(-0.6,-0.1,by=0.1),seq(0.1,0.6,by=0.1))
> Calibrate.Yhat3 <- calibrate(g,y,ticklab,Fr,ticklab,lm=TRUE,dp=TRUE,tl=0.1,
+                                axislab="Sauts",showlabel=FALSE)

---------- Calibration Results for  Sauts  -----------------
Length of 1 unit of the original variable =  4.3103
Angle                                     = -46.38 degrees
Optimal calibration factor                =  18.5787
Used calibration factor                   =  18.5787
Goodness-of-fit                           =  0.9986
Goodness-of-scale                         =  0.9986
-------------------------------------------------------------
```
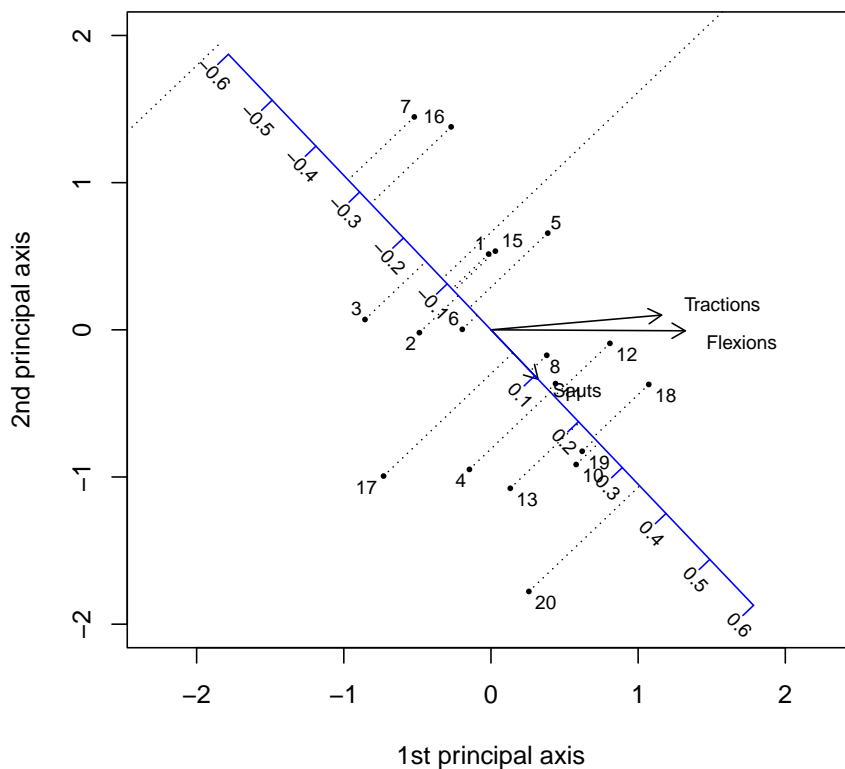
```
> plot(rda.results$Gxs[,1],rda.results$Gxs[,2],pch=16,asp=1,xlim=c(-2,2),
+       ylim=c(-2,2),cex=0.5,xlab="1st principal axis",
+ ylab="2nd principal axis")
> arrows(0,0,rda.results$Gxs[,1],rda.results$Gxs[,2],length=0.1)
> arrows(0,0,rda.results$Gyp[,1],rda.results$Gyp[,2],length=0.1)
> textxy(rda.results$Gxs[,1],rda.results$Gxs[,2],colnames(X),cex=0.75)
> textxy(rda.results$Gyp[,1],rda.results$Gyp[,2],colnames(Y),cex=0.75)
> y <- rda.results$B[,3]
> g <- rda.results$Gyp[3,1:2]
> Fr <- rda.results$Gxs[,1:2]
> ticklab <- seq(-0.4,0.4,0.2)
> W <-cor(X)
> Calibrate.Y3 <- calibrate(g,y,ticklab,Fr,ticklab,lm=TRUE,dp=TRUE,tl=0.1,
+                           weights=W,axislab="Sauts",showlabel=FALSE)

---------- Calibration Results for  Sauts  -----------------
Length of 1 unit of the original variable =  4.3103
Angle                                     = -46.38 degrees
Optimal calibration factor                =  18.5787
Used calibration factor                   =  18.5787
Goodness-of-fit                           =  0.9986
```
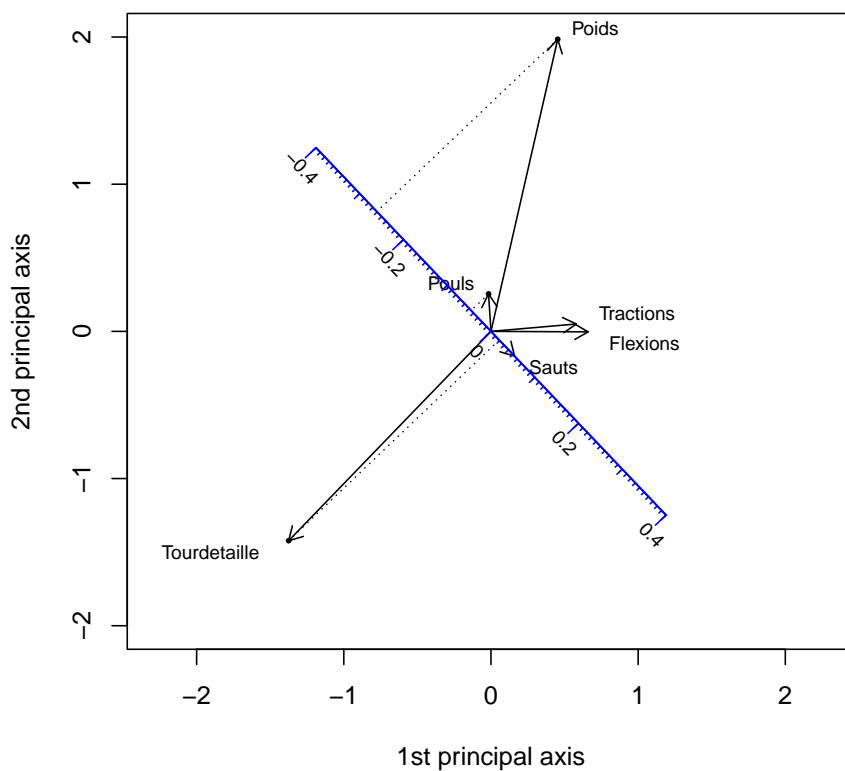
23

```
Goodness-of-scale                          =  0.9986
-----------------------------------------------------------------

> ticklab <- seq(-0.4,0.4,0.1)
> Calibrate.Y3 <- calibrate(g,y,ticklab,Fr,ticklab,lm=FALSE,tl=0.05,verb=FALSE,
+                                 weights=W)
> ticklab <- seq(-0.4,0.4,0.01)
> Calibrate.Y3 <- calibrate(g,y,ticklab,Fr,ticklab,lm=FALSE,tl=0.025,verb=FALSE,
+                                 weights=W)
```


The first biplot shown is a biplot of the fitted values (obtained from the regression of Y onto X). Vectors for the response variables are multiplied by a factor of 3 to increase readability. The fitted values of the regression of Sauts onto the body measurements have a goodness of fit of 0.9984 and can very well be recovered by projection onto the calibrated axis. The second biplot is a biplot of the matrix of regression coefficients. We calibrated the biplot axis for "Sauts", such that the regression coefficients of the explanory variables with respect to "Sauts" can be recovered. The goodness of fit for "Sauts" is over 0.99, which means that the regression coefficients are close to perfectly displayed. Note that the calibration for Sauts for the regression coefficients is done by GLS with weight matrix equal to the correlation matrix of the X variables (`weights=W`).

5 Online documentation

Online documentation for the package can be obtained by typing `vignette("CalibrationGuide"` or by accessing the file `CalibrationGuide.pdf` in the `doc` directory of the installed package.

6 Version history

Version 1.6:

- Function `rad2degree` and `shiftvector` have been added.

- Function calibrate has changed. Argument `shift` from previous versions is obsolete, and replaced by `shiftdir, shiftfactor` and `shiftvec`.

Version 1.7.2:

- Function `textxy` has been modified and improved. Arguments `dcol` and `cx` no longer work, and their role has been taken over by `col` and `cex`. A new argument `offset` controls the distance between point and label.

Acknowledgements

References

[Anderson (1984)] Anderson, T. W. (1984) An Introduction to Multivariate Statistical Analysis John Wiley, Second edition, New York.

[Frets (1921)] Frets, G. P. (1921) Heredity of head form in man, Genetica, 3, pp. 193-384.

[Gabriel, 1971] Gabriel, K. R. (1971) The biplot graphic display of matrices with application to principal component analysis. Biometrika 58(3) pp. 453-467.

[Gower and Hand (1996)] Gower, J. C. and Hand, D. J. (1996) Biplots Chapman & Hall, London.

[Graffelman (2005)] Graffelman, J. (2005) Enriched biplots for canonical correlation analysis Journal of Applied Statistics 32(2) pp. 173-188.

[Graffelman and Aluja-Banet (2003)] Graffelman, J. and Aluja-Banet, T. (2003) Optimal Representation of Supplementary Variables in Biplots from Principal Component Analysis and Correspondence Analysis Biometrical Journal, 45(4) pp. 491-509.

[Graffelman and van Eeuwijk (2005)] Graffelman, J. and van Eeuwijk, F. A., (2005) Calibration of multivariate scatter plots for exploratory analysis of relations within and between sets of variables in genomic research, Biometrical Journal, 47, 6, 863-879.

[Leisch (2002)] Leisch, F. (2002) Sweave: Dynamic generation of statistical reports using literate data analysis Compstat 2002, Proceedings in Computational Statistics pp. 575-580, Physica Verlag, Heidelberg, ISBN 3-7908-1517-9 URL http:/www.ci.tuwien.ac.at/ leisch/Sweave.

[Manly (1989)] Manly, B. F. J. (1989) Multivariate statistical methods: a primer Chapman and Hall, London.

[Mardia et al.(1979)] Mardia, K. V. and Kent, J. T. and Bibby, J. M. (1979) Multivariate Analysis Academic Press London.

[R Development Core Team (2004)] R Development Core Team (2004) R: A language and environment forstatistical computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-00-3, http://www.R-project.org.

[Tenenhaus (1998)] Tenenhaus, M. (1998) La Régression PLS Paris, Éditions Technip.

[Venables and Ripley (2002)] Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S-Plus New York, Fourth edition, Springer.