Simulation: effects of subgroup differences on principal component analyses

Chris Evans

31/07/2014

Document format and generation

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

General introduction

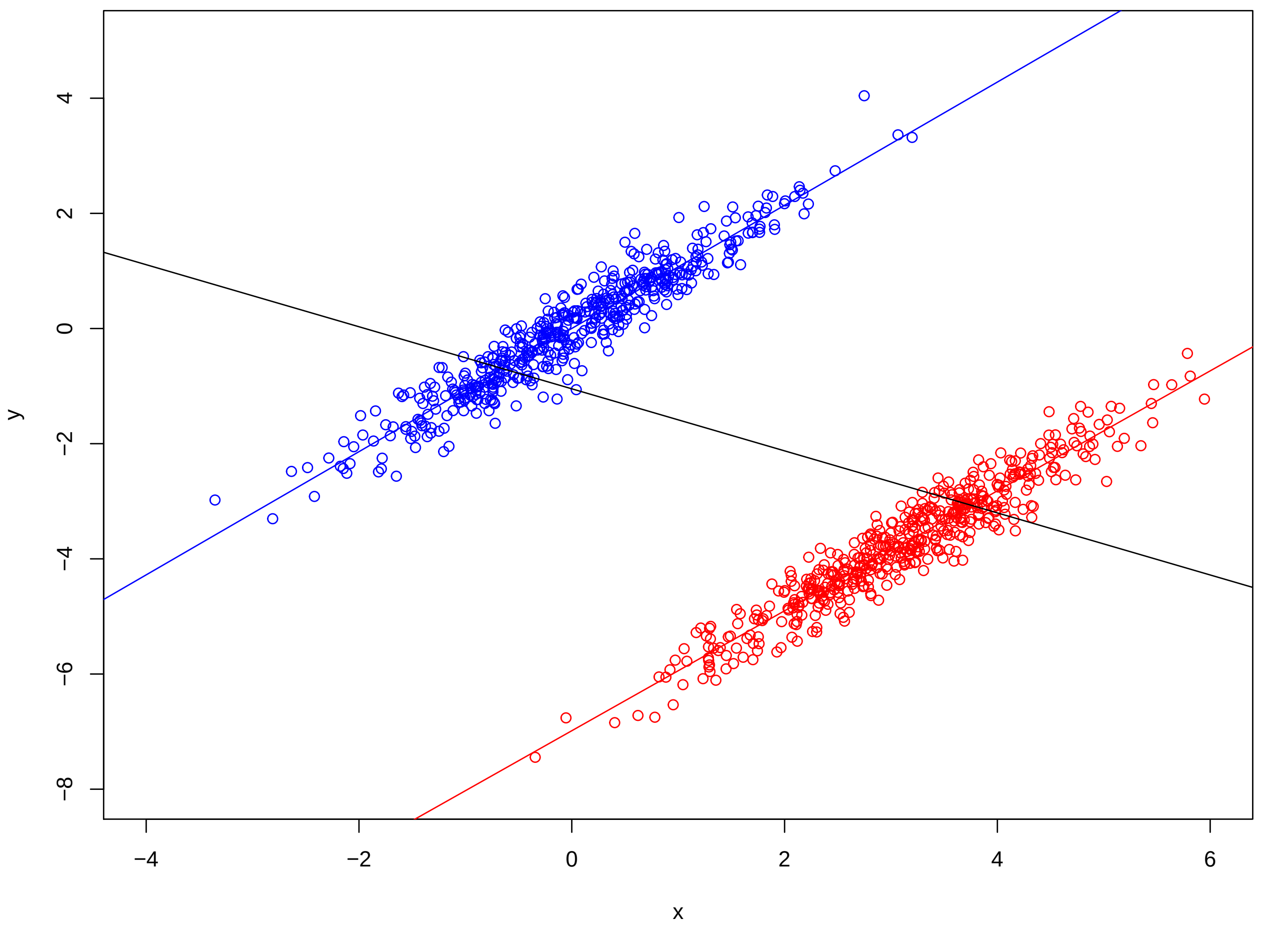
This started as a way of reassuring myself that my anxieties about an exploratory factor analysis (EFA) of data from the COPES scale where the data came from 21 different therapeutic communities (TCs) and from staff and non-staff members of the those communities and included some repeat data as well as one off data *might* find an EFA picture of the psychometric structure of the replies that relected more the differences between the units and perhaps the staff/non-staff difference and even perhaps the initial/later issue. It's turned into a bit of an experiment in using the latest "weaving" capabilities within Rmarkdown and Rstudio to mix text and code and generate HTML, PDF or Word documents.

Statistical issue

There's a simple bivariate equivalent of this EFA issue: that the correlation or regression beween two variables may be the same in two groups but the groups so different in mean values on the two variables that the overall correlation or regression between the two in the pooled dataset is quite different from that in each group.

```
options(width=120)
nGrps <- 2
nInGrp <- 500
grp2XInc <- 3
grp2YDec <- 7
ratio <- 3
slope <- 1.05
grp1x <- rnorm(nInGrp)
grp1err <- rnorm(nInGrp)/ratio
grp2x <- rnorm(nInGrp) + grp2XInc
grp2err <- rnorm(nInGrp)/ratio
grp1y1 <- slope*grp1x + grp1err
grp2y <- slope*grp2x + grp2err - grp2YDec
xMeanDiff <- mean(grp2x) - mean(grp1x)
yMeanDiff <- mean(grp2y) - mean(grp1y1)
grp1Pearson <- cor(grp1x,grp1y1)
grp2Pearson <- cor(grp1x,grp1y1)
x <- c(grp1x,grp2x)
y <- c(grp1y1,grp2y)
allPearson <- cor(x,y)
xmin <- floor(min(x))
```

```
xmax <- ceiling(max(x))
ymin <- floor(min(y))
ymax <- ceiling(max(y))
par(col=1)
plot(grp1x,grp1y1,
     xlim=c(xmin,xmax),
     ylim=c(ymin,ymax),
     xlab="x",
     ylab="y",
     type="n")
par(col="blue")
points(grp1x,grp1y1,
     xlim=c(-4,5),
     ylim=c(-15,4))
abline(lm(grp1y1 ~ grp1x))
par(col="red")
points(grp2x,grp2y,
     xlim=c(-4,5),
     ylim=c(-15,4))
abline(lm(grp2y ~ grp2x))
par(col=1)
abline(lm(y ~ x))
```



So that used R's pseudorandom number generator to generate Gaussian x and y variables in 2 groups each with 500 data points and with exactly the same relationship between the x and y variables but with a substantial mean shift on each variable between the groups. The population value difference is a x increase of 3 and a y decrease of 7. The sample mean x difference is 3.12 and the y difference is -3.73. The correlation between x and y in group 1 is 0.96 and that in group 2 is 0.96 but the overall Pearson correlation between x and y pooled across the two groups is very different at -0.46.

It's pretty easy to see how this happens from the plot.

OK, let's go to the simple multivariate case

This is really very similar to the above but this time I have generated four more y variables within each group, still keeping the x/y relationship the same for all y and within both groups. However, I have arranged things so that the group differences for Y1, Y2 and Y3 are the same (and the same as they were in the bivariate example above) but the group differences for Y4 and Y5 are rather different.

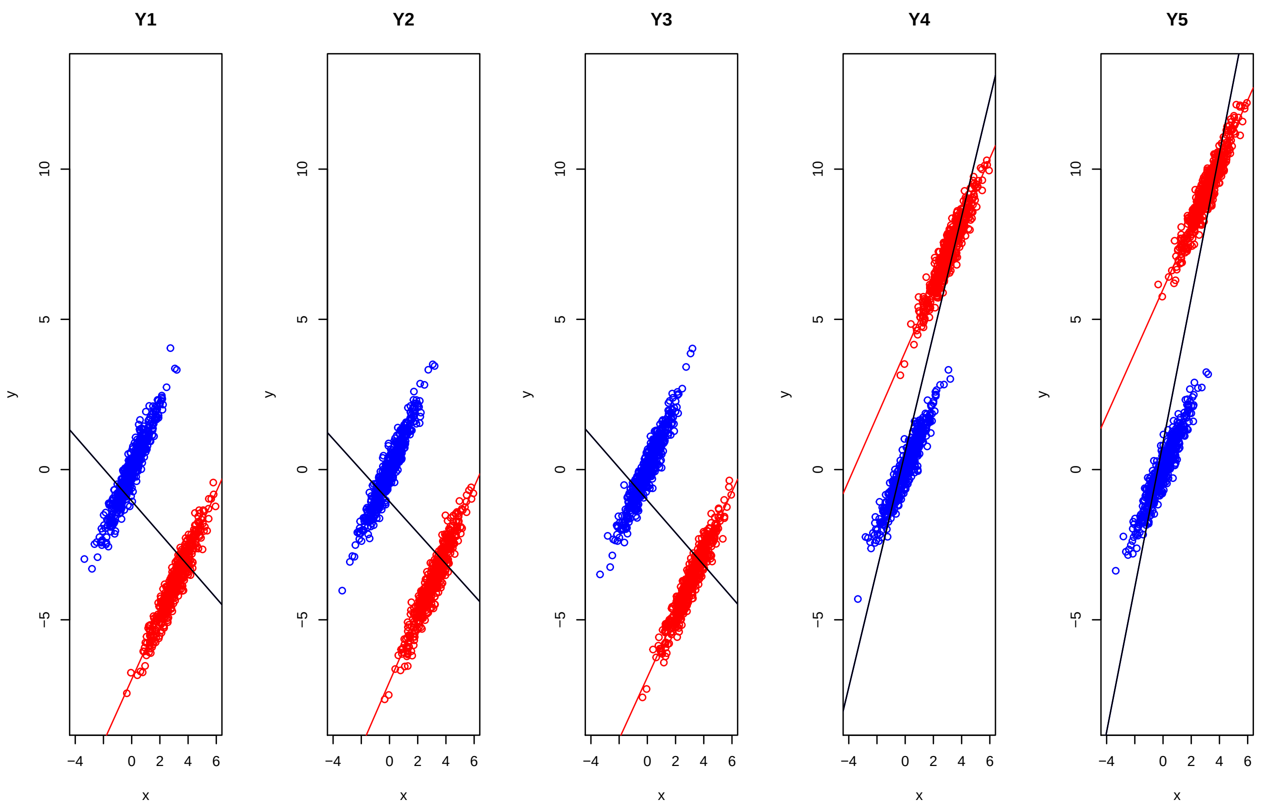
```
### group 1
grp1err2 <- rnorm(nInGrp)/ratio
grp1y2 <- slope*grp1x + grp1err2
grp1err3 <- rnorm(nInGrp)/ratio
grp1y3 <- slope*grp1x + grp1err3
grp1err4 <- rnorm(nInGrp)/ratio
grp1y4 <- slope*grp1x + grp1err4
grp1err5 <- rnorm(nInGrp)/ratio
grp1y5 <- slope*grp1x + grp1err5
### group 2
grp2err2 <- rnorm(nInGrp)/ratio
grp2YDec2 <- grp2YDec
grp2y2 <- slope*grp2x + grp2err2 - grp2YDec2
grp2err3 <- rnorm(nInGrp)/ratio
grp2YDec3 <- grp2YDec
grp2y3 <- slope*grp2x + grp2err3 - grp2YDec3
grp2err4 <- rnorm(nInGrp)/ratio
grp2YDec4 <- -4
grp2y4 <- slope*grp2x + grp2err4 - grp2YDec4
grp2err5 <- rnorm(nInGrp)/ratio
grp2YDec5 <- -6
grp2y5 <- slope*grp2x + grp2err5 - grp2YDec5

grp1yAll <- cbind(grp1y1,grp1y2,grp1y3,grp1y4,grp1y5)
grp2yAll <- cbind(grp2y,grp2y2,grp2y3,grp2y4,grp2y5)
yAll <- rbind(grp1yAll,grp2yAll)
```

The plots below show the rather different overall x/y correlations this gives for Y4 and Y5 from Y1, Y2 and Y3.

```
par(mfrow=c(1,5))
par(col=1)
combPlotFun <- function(x,y,title,xmin,xmax,ymin,ymax) {
             par(col=1)
             nGrp <- length(x)/2
             grp1x <- x[1:nGrp]
             grp1y1 <- y[1:nGrp]
             grp2x <- x[nGrp+1:length(x)]
```

```
              grp2y <- y[nGrp+1:length(y)]
              plot(x,y,
                   xlim=c(xmin,xmax),
                   ylim=c(ymin,ymax),
                   xlab="x",
                   ylab="y",
                   main=title,
                   type="n")
              par(col="blue")
              points(grp1x,grp1y1)
              abline(lm(y ~ x))
              par(col="red")
              points(grp2x,grp2y,
                     xlim=c(-4,5),
                     ylim=c(-15,4))
              abline(lm(grp2y ~ grp2x))
              par(col=1)
              abline(lm(y ~ x))
            }
ymin <- floor(min(yAll))
ymax <- ceiling(max(yAll))
for (i in 1:5) {
  combPlotFun(x,yAll[,i],paste("Y",i,sep=""),xmin,xmax,ymin,ymax)
}
```



I'm only going to use these five y variables in the EFA. I'm going to speed things up and keep them simple by actually doing a principal component analysis (PCA). There's ample evidence that the differences beween PCA loading matrices and those for "true EFAs" of whatever type will be small.

Their correlation matrix for the five y variables within group 1 is:

```
##        grp1y1 grp1y2 grp1y3 grp1y4 grp1y5
## grp1y1   1.00   0.92   0.91   0.91   0.91
## grp1y2   0.92   1.00   0.91   0.91   0.91
## grp1y3   0.91   0.91   1.00   0.90   0.90
## grp1y4   0.91   0.91   0.90   1.00   0.91
## grp1y5   0.91   0.91   0.90   0.91   1.00
```

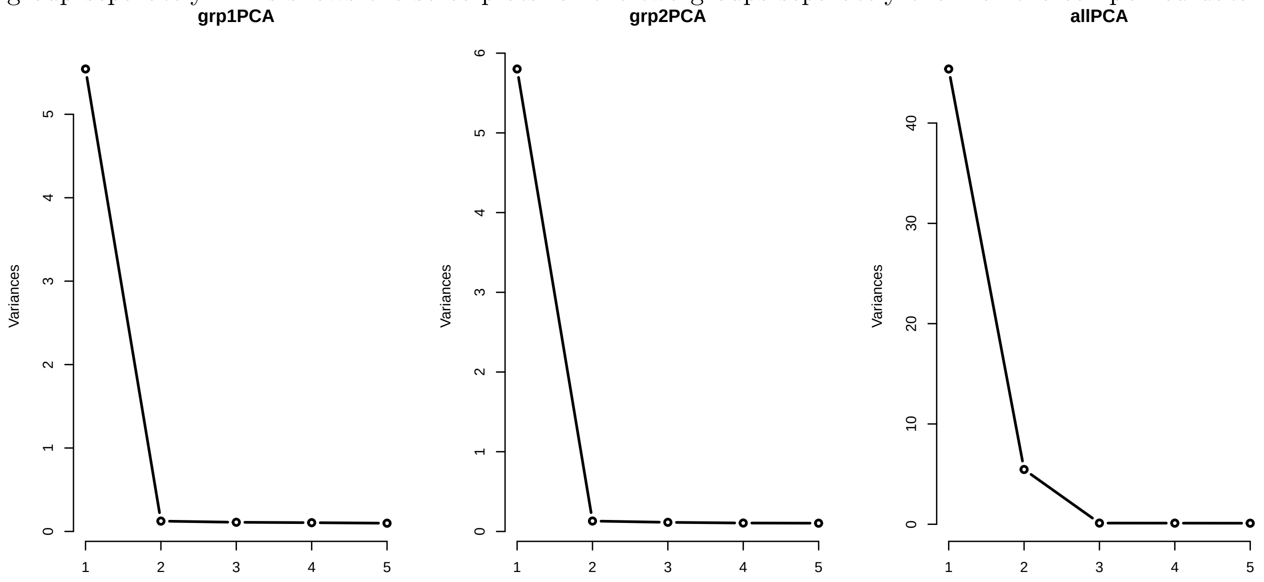
and within group 2 it is:

```
##        grp2y grp2y2 grp2y3 grp2y4 grp2y5
## grp2y   1.00   0.92   0.91   0.91   0.91
## grp2y2  0.92   1.00   0.90   0.91   0.91
## grp2y3  0.91   0.90   1.00   0.91   0.91
## grp2y4  0.91   0.91   0.91   1.00   0.91
## grp2y5  0.91   0.91   0.91   0.91   1.00
```

But the correlation for the pooled data is rather different:

```
##        grp1y1 grp1y2 grp1y3 grp1y4 grp1y5
## grp1y1   1.00   0.98   0.98  -0.69  -0.73
## grp1y2   0.98   1.00   0.98  -0.68  -0.72
## grp1y3   0.98   0.98   1.00  -0.69  -0.73
## grp1y4  -0.69  -0.68  -0.69   1.00   0.99
## grp1y5  -0.73  -0.72  -0.73   0.99   1.00
```

Of course, this produces very different PCA results for the pooled data from the analyses for each group separately. This shows the scree plots for the two groups separately then for the compbined data.



It's easy to see that the pooled data has a scree plot supporting a two component solution whereas the plots for the within group analyses show a very clear unicomponent solution is fine to capture the systematic covariance in the data.

This is nice and clear in the loadings: here for group 1.

```
grp1PCA$rotation
```

```
##              PC1     PC2      PC3     PC4      PC5
## grp1y1 -0.4576  0.1803 -0.24659 -0.6782 -0.48717
## grp1y2 -0.4499 -0.2786 -0.04597 -0.3179  0.78534
## grp1y3 -0.4431 -0.7219 -0.06815  0.3881 -0.35681
## grp1y4 -0.4426  0.2811  0.84224  0.1102 -0.05997
## grp1y5 -0.4427  0.5383 -0.47230  0.5255  0.12245
```

And here for the pooled data. (I've omitted group 2 as it's essentially identical to group 1.)

```
allPCA$rotation
```

```
##              PC1     PC2     PC3     PC4       PC5
## grp1y1 -0.2656 -0.5142  0.4021 -0.1008 -0.702276
## grp1y2 -0.2608 -0.5188  0.3692 -0.1459  0.710820
## grp1y3 -0.2640 -0.5032 -0.7370  0.3659 -0.006227
## grp1y4  0.5510 -0.3443 -0.3069 -0.6947 -0.032268
## grp1y5  0.6986 -0.3077  0.2543  0.5934  0.021527
```

Varimax rotation of first two components from the pooled data:

```
varimax(allPCA$rotation[,1:2])
```

```
## $loadings
##
## Loadings:
##        PC1    PC2
## grp1y1        -0.579
## grp1y2        -0.581
## grp1y3        -0.568
## grp1y4  0.647
## grp1y5  0.762
##
##                PC1 PC2
## SS loadings    1.0 1.0
## Proportion Var 0.2 0.2
## Cumulative Var 0.2 0.4
##
## $rotmat
##          [,1]   [,2]
## [1,]  0.8919 0.4523
## [2,] -0.4523 0.8919
```

Discussion

I thinks it's very clear that factor structure from a mixed group pooled sample can reflect group mean differences and show a factor structure that is different from that in any of the subgroups even when all data from each subgroup actually has the same factor structure (quite different from that emerging from analysis of the pooled data). Caveat extractor!

I showed only the bivariate case in which there is the same linear regression linking the two variables within each group but a quite different regression emerges from (unthinking) analysis of the pooled data. It's clearly equally possible to have two different regressions in the two groups and a regression that has a different slope from either and it's possible to find a statistically non-significant and near zero regression/correlation if the effect of the group mean differences neatly cancels a shared correlation in both groups just as equal and opposite regression slopes in two samples will cancel to show a zero correlation in the pooled data (assuming equal sized groups). All manner of misleading findings are clearly possible. It's pretty easy to see the problems with sensible inspection of scattergrams in the bivariate case but hard to do so with multivariate EFA and with more than few groups.