Factor analysis and CCA

Wenyue Shi, Carlton Washburn, Qijing Zhang

August 11, 2015

Factor Analysis

Was developed around 1904 for determining IQ scores. Developed by Spearman. Its biggest difference from PCA is that it corporates error term.

Factor Models

Feature vector:

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iD} \end{pmatrix} \in \mathbb{R}^D$$

Model

Matrix Notation

$$x_i = \alpha + Bf_i + \epsilon_i$$

where $\alpha \in \mathbb{R}^D$, B is the Factor Loadings Matrix ($D \times K$), K << D. f_i is the Factor Scores $\in \mathbb{R}^K$, and ϵ_i is an error vector $\in \mathbb{R}^D$. i = 1,...,N (N is the number of observations).

Scalar Notation

$$\begin{aligned} x_{ij} &= \alpha_j + B_j^T f_i = \epsilon_{ij} \\ &= \alpha_j + \Sigma_{l=1}^k B_{jl} f_{il} + \epsilon_{ij} \end{aligned}$$

Assumptions of Factor Analysis

$$f_{il} \sim N(0,1), l = 1, \cdots, k$$

$$\epsilon_{ij} \sim N(0, \sigma_j^2), j = 1, \cdots, D$$

σ_j^2 is the idiosyncratic variance for feature j.

Difference between PCA and factor analysis

In terms of assumption, PCA relies on the geometric assumption that each vector has to be perpendicular to each other. But factor analysis relies on a different set of assumptions enumerated above in 'Scalar Notation' section.
Also, since factor analysis has more assumptions than PCA, it generates more outcome as well, the most important of which being predictions. You can get an error bar from factor analysis, which you can't get from PCA.

Canonical correlation analysis

CCA is a method to identify and measure the associations between two sets of variables. Its biggest difference from PCA is that it operates on 2 axes instead of 1.

Two Examples

1. two types (sets) of measurements on students:
   - Academic: maths, reading, etc
   - Psychological: motivation, self concept, etc
2. mouse
   - Genetic: set of genes
   - Physiological: level of lipid expression

Notations

X = feature matrix 1 $\in \mathbb{R}^{N*D_1}$
Y = feature matrix 2 $\in \mathbb{R}^{N*D_2}$
N = # of observations

The Problem

The first pair of canonical variates
$v_1 \in \mathbb{R}^{D_1}$
$w_1 \in \mathbb{R}^{D_2}$
are defined so that
$cor(X_i^T v_1, Y_i^T w_1)$ is as large as possible

Large negative correlation is just as good (aka indicative) as large positive correlation, because they are the same thing once you flip the direction of the vector.

R Markdown files with notes

To see the difference between PCA and Factor Analysis, let's take at a look at the following data set.

This is a exchange rate data set that shows the buying power of USD, with rows being months, and columns being different currencies. So every data indicates how many of that particular currency a dollar would buy for a particular month.

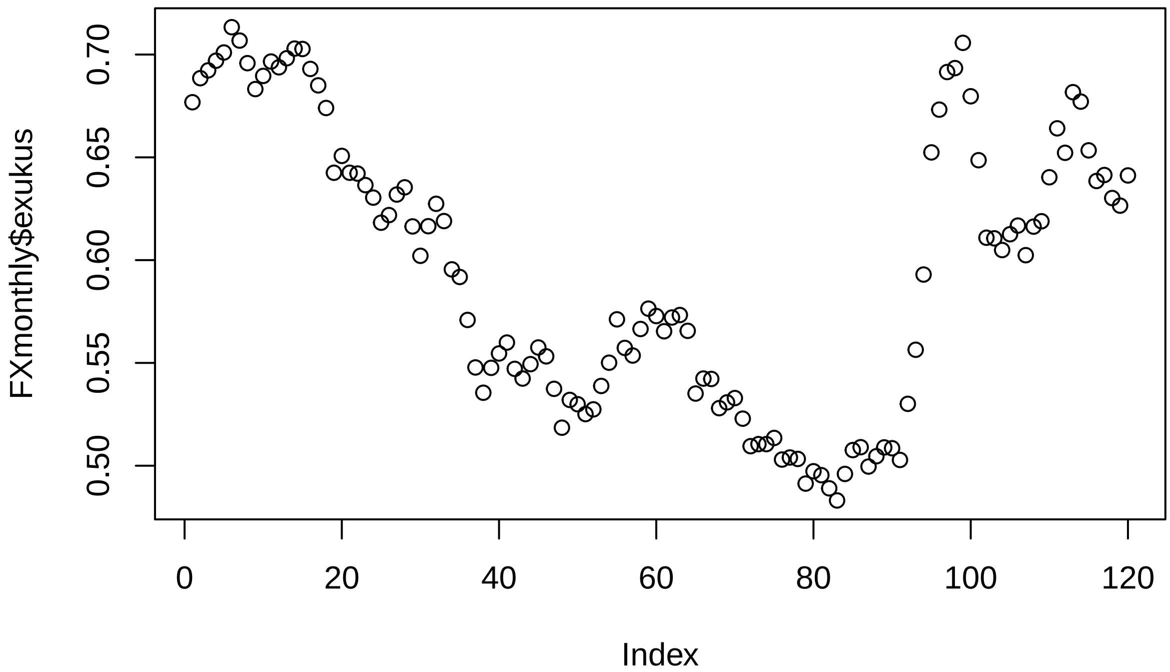
```r
FXmonthly = read.csv('../STA380/data/FXmonthly.csv', header=TRUE)
summary(FXmonthly)
```

```
##      exalus          exbzus          excaus            exchus
##  Min.   :1.007   Min.   :1.590   Min.   :0.9672   Min.   :6.650
##  1st Qu.:1.195   1st Qu.:1.897   1st Qu.:1.0636   1st Qu.:6.888
##  Median :1.328   Median :2.260   Median :1.2102   Median :8.070
##  Mean   :1.408   Mean   :2.343   Mean   :1.2481   Mean   :7.743
##  3rd Qu.:1.535   3rd Qu.:2.721   3rd Qu.:1.3826   3rd Qu.:8.277
##  Max.   :1.994   Max.   :3.797   Max.   :1.5997   Max.   :8.278
```

```
##      exdnus          exhkus          exinus          exjpus     
##  Min.   :4.734   Min.   :7.743   Min.   :39.27   Min.   : 81.73  
##  1st Qu.:5.522   1st Qu.:7.763   1st Qu.:44.16   1st Qu.:103.69  
##  Median :5.892   Median :7.790   Median :45.86   Median :111.84  
##  Mean   :6.224   Mean   :7.783   Mean   :45.55   Mean   :109.98  
##  3rd Qu.:6.556   3rd Qu.:7.800   3rd Qu.:47.67   3rd Qu.:118.69  
##  Max.   :8.740   Max.   :7.820   Max.   :51.13   Max.   :133.64  
##      exkous           exmaus          exmxus           exnzus     
##  Min.   : 914.9   Min.   :3.099   Min.   : 9.064   Min.   :1.249  
##  1st Qu.:1009.5   1st Qu.:3.443   1st Qu.:10.477   1st Qu.:1.399  
##  Median :1159.5   Median :3.756   Median :10.916   Median :1.512  
##  Mean   :1131.8   Mean   :3.616   Mean   :11.110   Mean   :1.650  
##  3rd Qu.:1232.6   3rd Qu.:3.800   3rd Qu.:11.420   3rd Qu.:1.806  
##  Max.   :1449.6   Max.   :3.800   Max.   :14.647   Max.   :2.458  
##      exnous          exsius          exsfus           exslus      
##  Min.   :5.054   Min.   :1.299   Min.   : 5.723   Min.   : 85.73  
##  1st Qu.:6.064   1st Qu.:1.459   1st Qu.: 6.757   1st Qu.: 96.93  
##  Median :6.506   Median :1.631   Median : 7.455   Median :102.92  
##  Mean   :6.749   Mean   :1.601   Mean   : 7.732   Mean   :103.43  
##  3rd Qu.:7.020   3rd Qu.:1.737   3rd Qu.: 8.063   3rd Qu.:110.84  
##  Max.   :9.301   Max.   :1.839   Max.   :11.676   Max.   :117.31  
##      exsdus           exszus           extaus          exthus     
##  Min.   : 5.947   Min.   :0.9686   Min.   :29.90   Min.   :29.87  
##  1st Qu.: 6.993   1st Qu.:1.1288   1st Qu.:32.19   1st Qu.:33.36  
##  Median : 7.534   Median :1.2384   Median :32.93   Median :38.84  
##  Mean   : 7.855   Mean   :1.2744   Mean   :32.97   Mean   :37.87  
##  3rd Qu.: 8.319   3rd Qu.:1.3604   3rd Qu.:33.92   3rd Qu.:41.68  
##  Max.   :10.793   Max.   :1.7856   Max.   :35.07   Max.   :45.64  
##      exukus           exvzus          exeuus      
##  Min.   :0.4831   Min.   :0.700   Min.   :0.6346  
##  1st Qu.:0.5345   1st Qu.:1.600   1st Qu.:0.7401  
##  Median :0.5942   Median :2.140   Median :0.7904  
##  Mean   :0.5946   Mean   :2.028   Mean   :0.8358  
##  3rd Qu.:0.6491   3rd Qu.:2.140   3rd Qu.:0.8818  
##  Max.   :0.7133   Max.   :4.290   Max.   :1.1723  
```

The plot shows the monthly exchange rate of USD-GBP.

```r
# USD-GBP exchange rate
plot(FXmonthly$exukus)
```

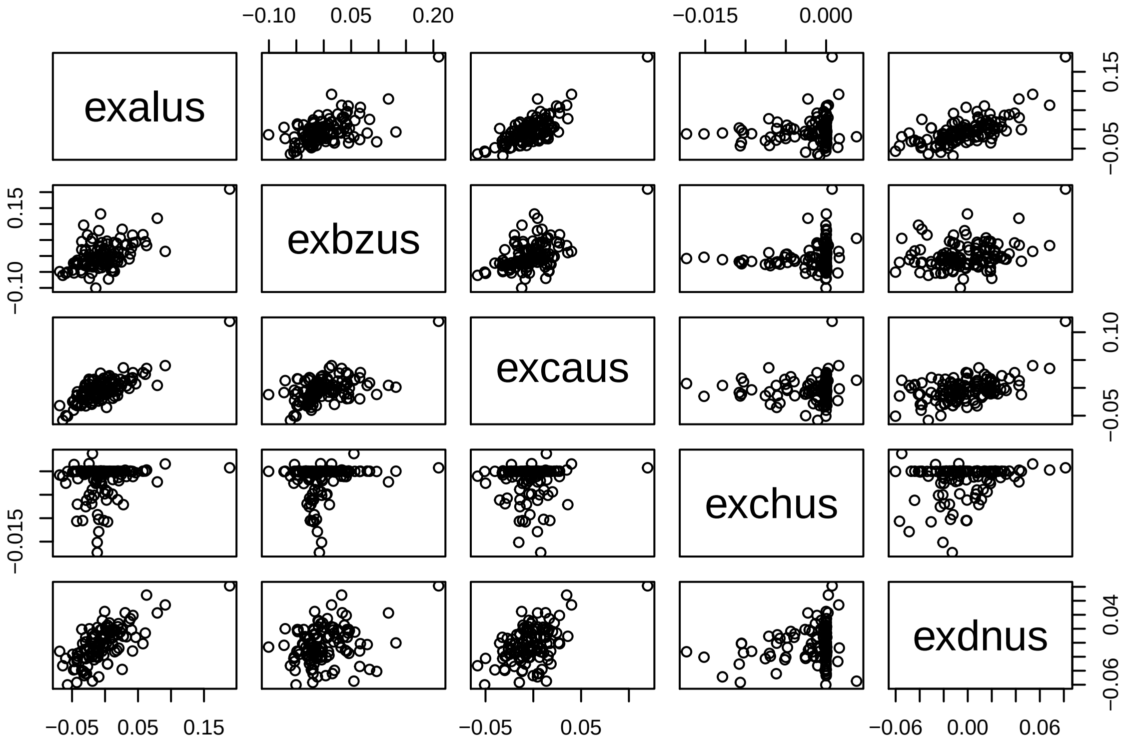


We are converting the exchange rate to a day-to-day returns, so that we can get the correlation between these returns.

```
# Convert everything to returns
FXmonthly <- (FXmonthly[2:120,]-FXmonthly[1:119,])/(FXmonthly[1:119,]) # proportion change
```

Some returns of the currencies here are highly correlated, for example Pounds and European Dollars. That makes sense because they are the main currencies used in the world, and people tend to trade between these currencies.

```
pairs(FXmonthly[,1:5])
```



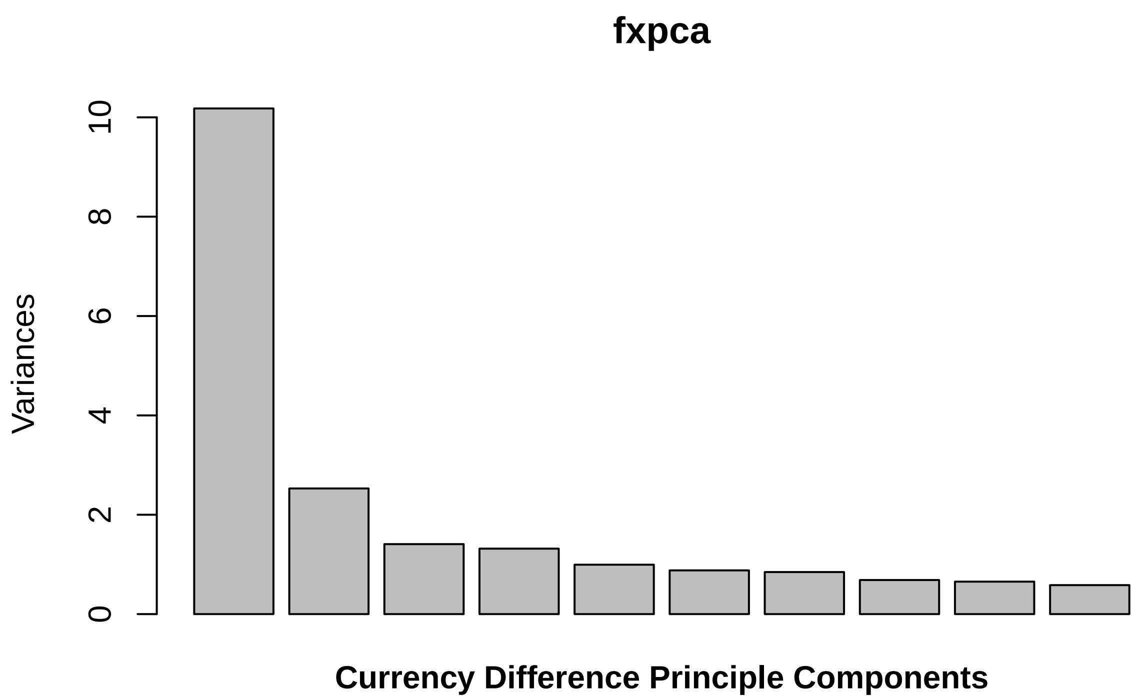

```
cor(FXmonthly[,c('exeuus','exhkus','excaus','exmxus','exukus')])
```

```
##            exeuus     exhkus    excaus     exmxus    exukus
## exeuus 1.0000000  0.1833495 0.5048342  0.2617354 0.7335453
## exhkus 0.1833495  1.0000000 0.1682997 -0.1734493 0.1145097
## excaus 0.5048342  0.1682997 1.0000000  0.5238391 0.4927518
## exmxus 0.2617354 -0.1734493 0.5238391  1.0000000 0.3203351
## exukus 0.7335453  0.1145097 0.4927518  0.3203351 1.0000000
```

Apply PCA to the data set and take a look at the variance explained by the components.

```
## PCA
fxpca = prcomp(FXmonthly, scale=TRUE)

plot(fxpca)
mtext(side=1, "Currency Difference Principle Components",  line=1, font=2)
```

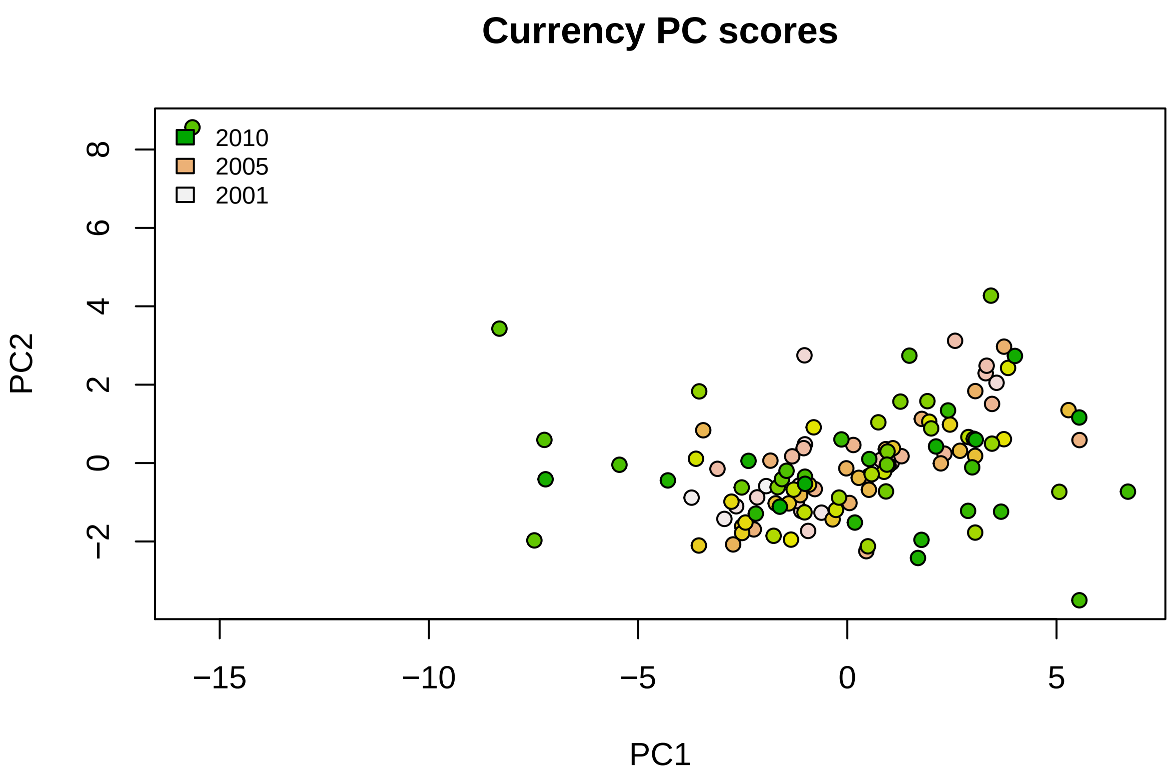


Get the principal component scores. The predict function here works the same as getting scores with the "$x" sign.

```
# Get the principal component scores
fx_scores = predict(fxpca)  # same as fxpca$x
```

Notice there's a huge outliers there when Lehman Brothers collapse and we want to apply CPA without that outlier.

```
# Color each point so that they get darker over time
plot(fx_scores[,1:2], pch=21, bg=terrain.colors(120)[120:1], main="Currency PC scores")
legend("topleft", fill=terrain.colors(3),
       legend=c("2010","2005","2001"), bty="n", cex=0.75)
outlier = identify(fx_scores[,1:2], n=1)
```




```
outlier = 92
```

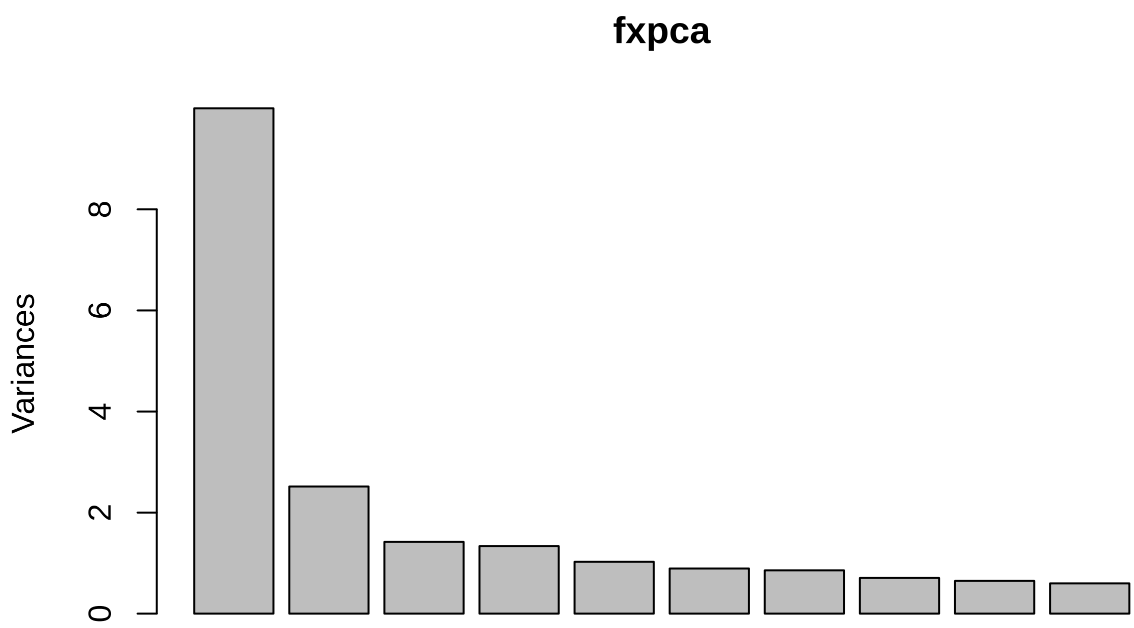
Now we don't have that extreme outlier.

```
# Huge outlier (Oct 2008 = month of the Lehman Brothers collapse)
FXmonthly[outlier,]
```

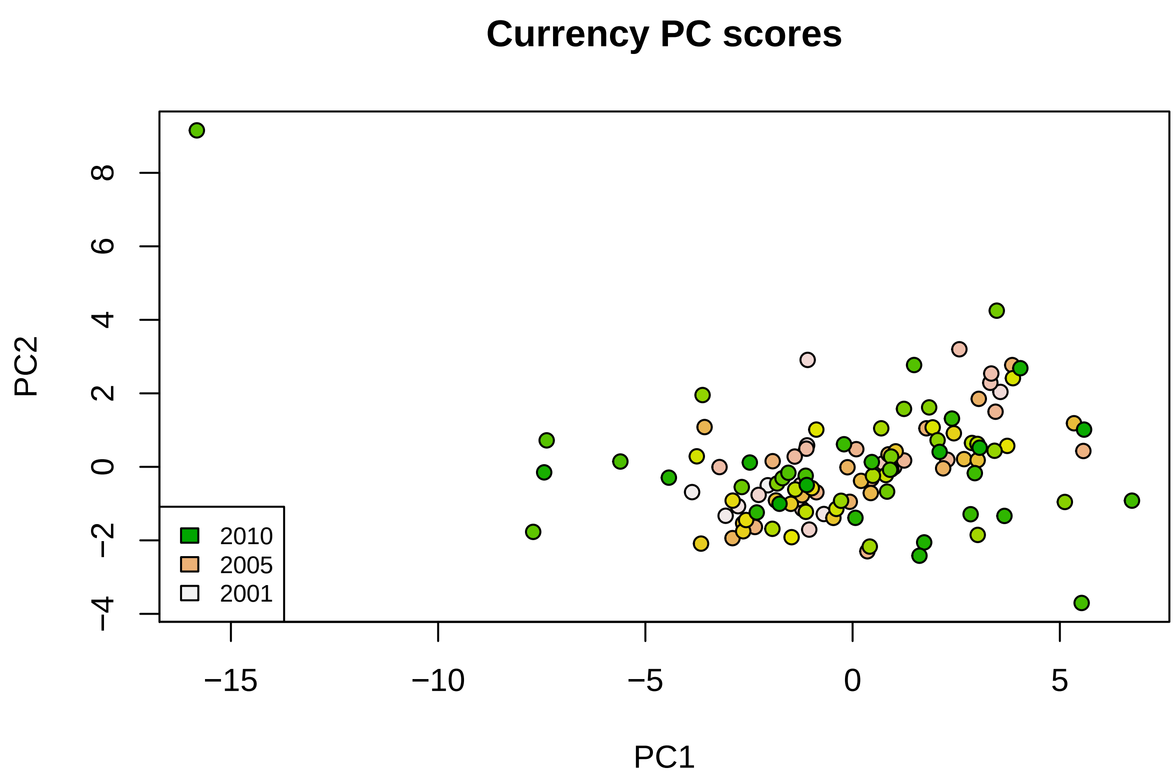
```
##              exalus    exbzus      excaus      exchus      exdnus
## SEP2008 0.07924894 0.1180629 0.004461319 -0.00226403 0.04261033
##                exhkus     exinus      exjpus     exkous     exmaus
## SEP2008 -0.002843383 0.06105803 -0.02551207 0.08484767 0.03318318
##              exmxus     exnzus     exnous     exsius     exsfus
## SEP2008 0.05417696 0.05226209 0.06800923 0.01743524 0.05351528
##                exslus     exsdus     exszus     extaus     exthus
## SEP2008 0.0009747674 0.06318721 0.02407527 0.02469492 0.01235955
##              exukus exvzus     exeuus
## SEP2008 0.04961328      0 0.04276955
```

```
# Re-run without the outlier
fxpca = prcomp(FXmonthly[-outlier,], scale=TRUE)
fx_scores = predict(fxpca)  # same as fxpca$x

plot(fxpca)
```

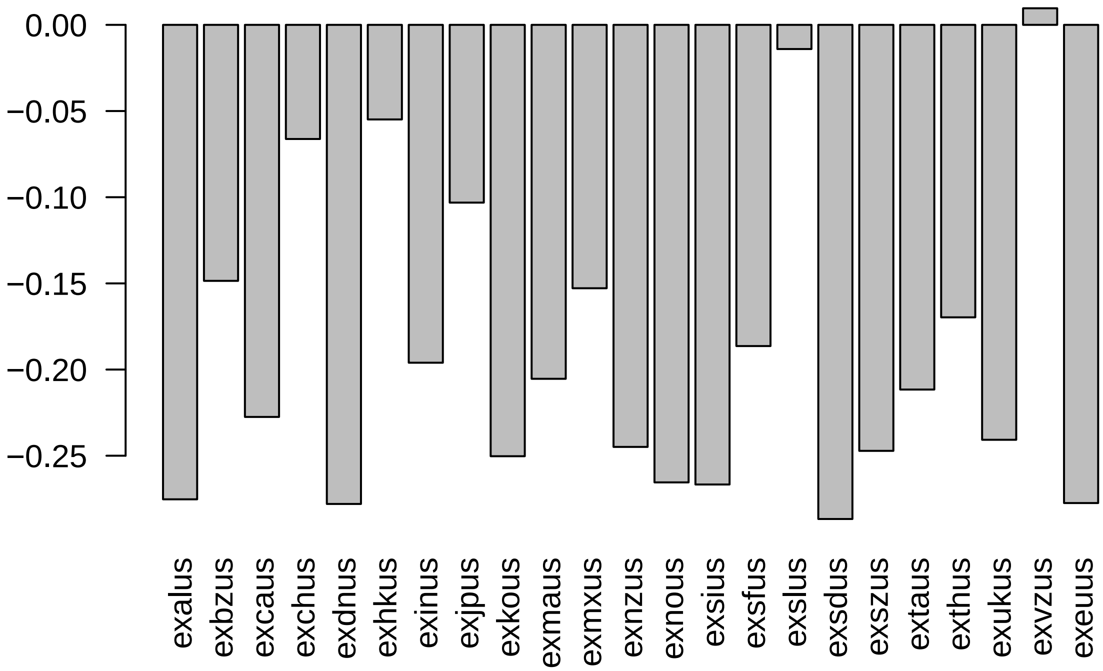



```r
plot(fx_scores[,1:2], pch=21, bg=terrain.colors(119)[119:1], main="Currency PC scores")
legend("bottomleft", fill=terrain.colors(3),
       legend=c("2010","2005","2001"), cex=0.75)
```



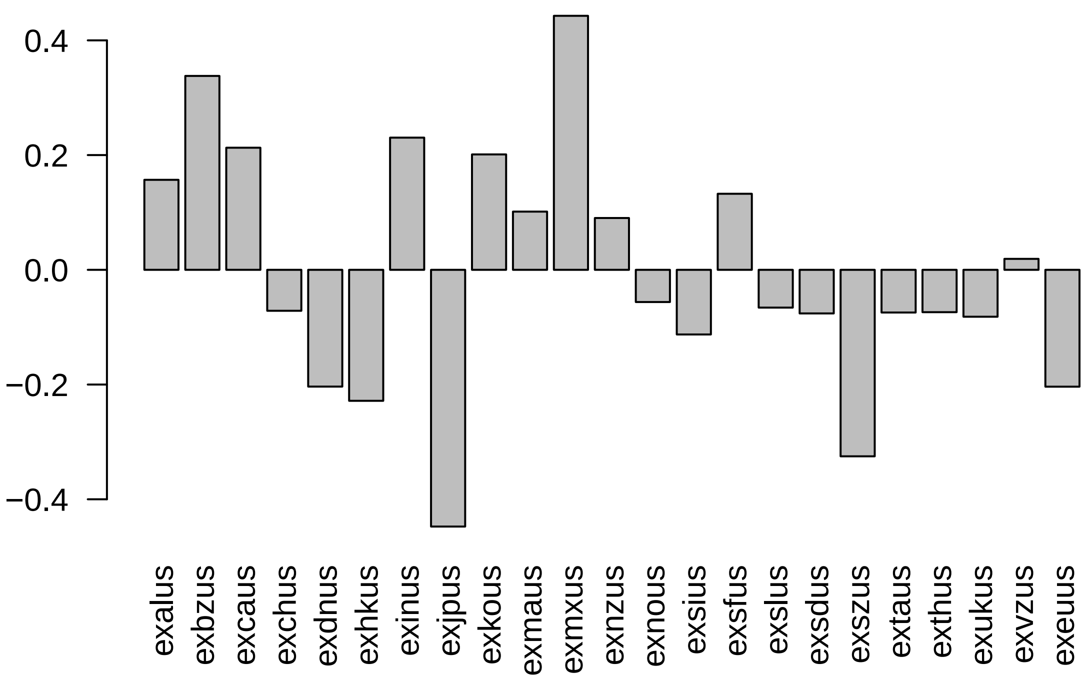
Bar plot the loadings of the first PC.

```r
barplot(fxpca$rotation[,1], las=2)
```



The bar plot for PC1 gives us a image about a portforlio that most of the countries would buy. And it shouws the U.S.'s overall strength.

```
barplot(fxpca$rotation[,2], las=2)  # fluctuation in trade
```



However the plot for PC2 shows more fluctuation in trading, with the positive ones indicating that the U.S. sells the currency and negative ones indicating that the U.S. purchases the currency.

The following table indicates the name of the country in terms of their code.

```r
currency_codes = read.table('../STA380/data/currency_codes.txt')
currency_codes
```

```
##    V1           V2
## 1  al    australia
## 2  bz       brazil
## 3  ca       canada
## 4  ch        china
## 5  dn      denmark
## 6  eu         euro
## 7  hk    hong kong
## 8  in        india
## 9  jp        japan
## 10 ko  south korea
## 11 ma     malaysia
## 12 mx       mexico
## 13 no       norway
## 14 nz  new zealand
## 15 sd       sweden
## 16 sf south africa
## 17 si    singapore
## 18 sl    sri lanka
## 19 sz  switzerland
## 20 ta       taiwan
## 21 th     thailand
## 22 uk           uk
## 23 vz    venezuela
```

Factor Analysis

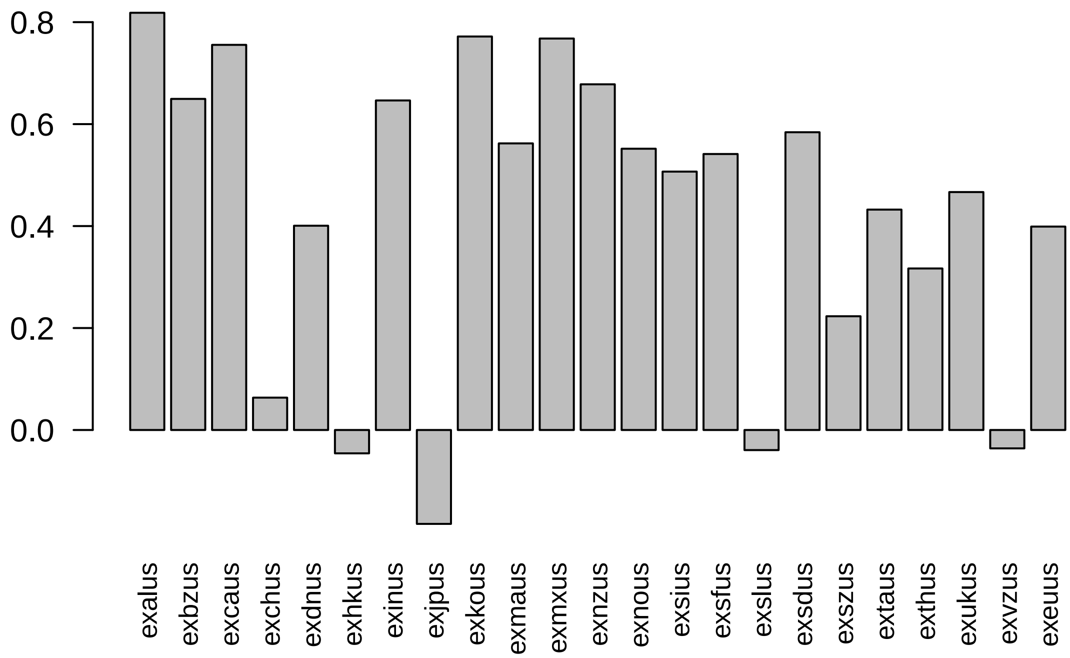
```r
# Compare with factor analysis
Y = scale(FXmonthly[-outlier,], center=TRUE, scale=FALSE)
fa_fx = factanal(Y, 3, scores='regression')
# 3 means you need 3 factors
print(fa_fx)
```

```
##
## Call:
## factanal(x = Y, factors = 3, scores = "regression")
##
## Uniquenesses:
## exalus exbzus excaus exchus exdnus exhkus exinus exjpus exkous exmaus
##  0.177  0.575  0.377  0.945  0.005  0.923  0.524  0.371  0.289  0.542
## exmxus exnzus exnous exsius exsfus exslus exsdus exszus extaus exthus
##  0.372  0.387  0.304  0.209  0.646  0.965  0.169  0.137  0.431  0.643
## exukus exvzus exeuus
##  0.432  0.967  0.005
##
## Loadings:
##        Factor1 Factor2 Factor3
## exalus  0.818   0.389
```

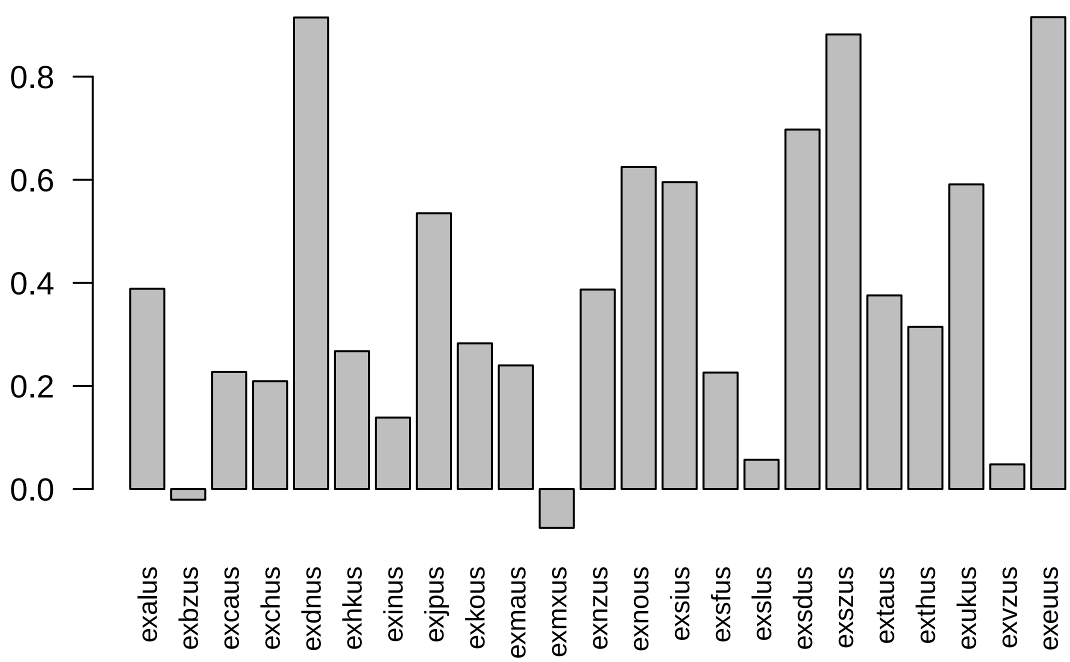
```
## exbzus  0.649
## excaus  0.755   0.227
## exchus          0.209
## exdnus  0.401   0.915
## exhkus          0.267
## exinus  0.646   0.139   0.198
## exjpus -0.184   0.535   0.556
## exkous  0.772   0.283   0.188
## exmaus  0.562   0.240   0.290
## exmxus  0.768          -0.180
## exnzus  0.678   0.387
## exnous  0.552   0.625
## exsius  0.507   0.595   0.424
## exsfus  0.541   0.226   0.101
## exslus                  0.174
## exsdus  0.584   0.697
## exszus  0.223   0.882   0.188
## extaus  0.432   0.376   0.491
## exthus  0.317   0.315   0.397
## exukus  0.467   0.591
## exvzus                 -0.172
## exeuus  0.399   0.915
##
##                Factor1 Factor2 Factor3
## SS loadings      6.154   5.247   1.209
## Proportion Var   0.268   0.228   0.053
## Cumulative Var   0.268   0.496   0.548
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 516.84 on 187 degrees of freedom.
## The p-value is 1.05e-32
```

Take a look at the loadings on the factors. Factor analysis here give us a little bit more information than PCA, as it tells us which currencies are least related with the factor. The higher the bar, the less they are related to this factor.

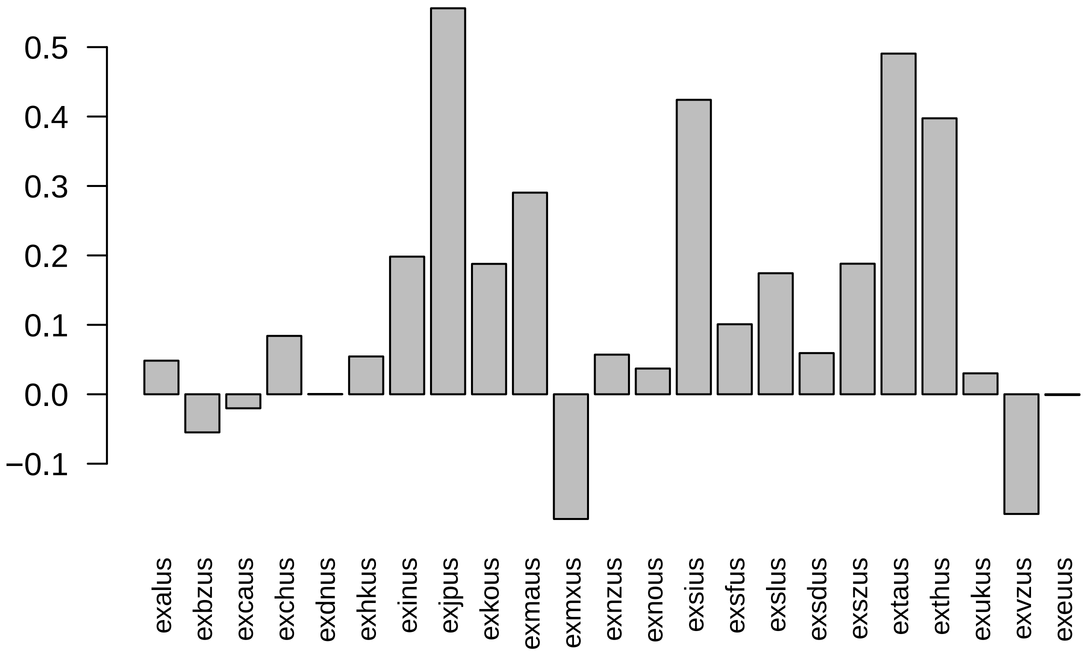
```
barplot(fa_fx$loadings[,1], las=2, cex.names=0.8)
```




```r
# which of the currencies are least related to the factors
# higher the bar, the less related to the factors, and this are the uniqueness
barplot(fa_fx$loadings[,2], las=2, cex.names=0.8)
```

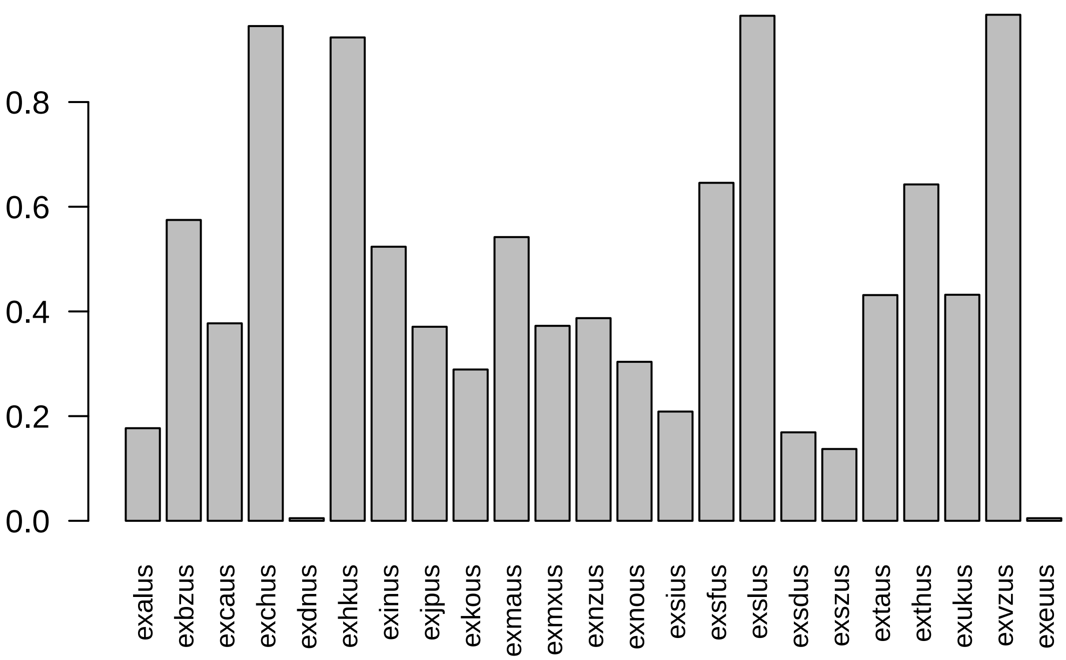



```r
barplot(fa_fx$loadings[,3], las=2, cex.names=0.8)
```

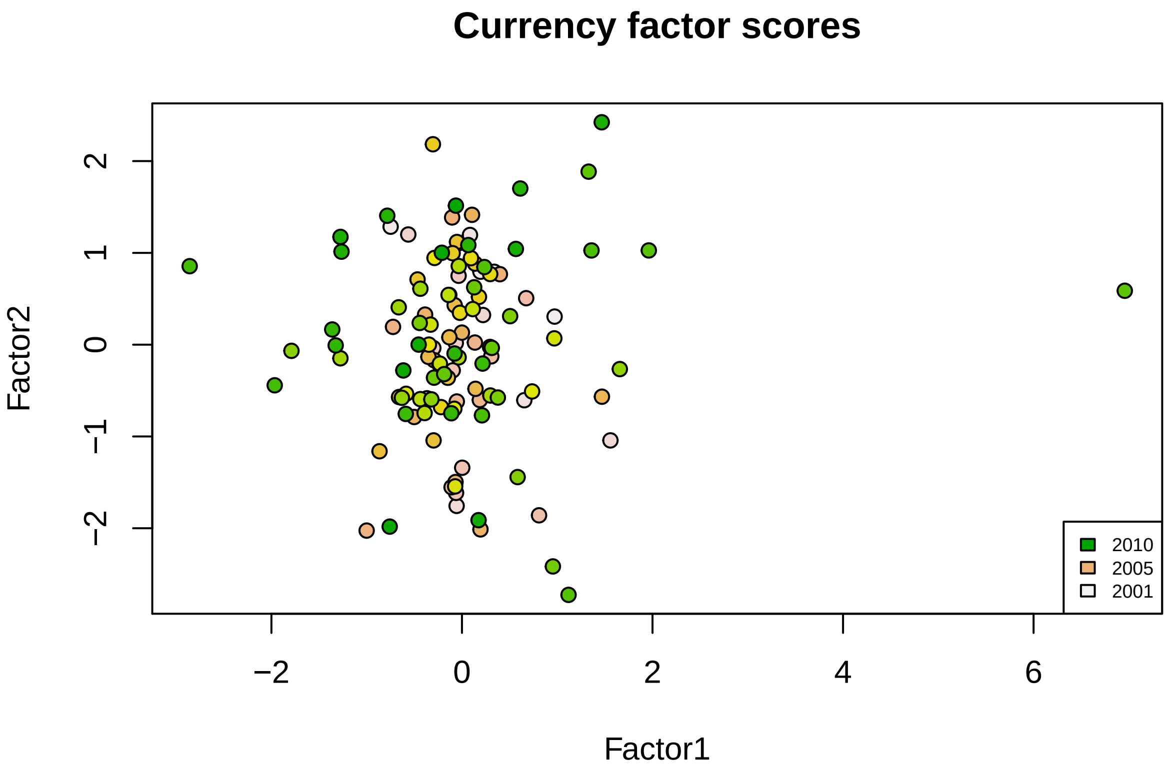


One thing we have with Factor Analysis but not with PCA is the information about the error. Here we can see the variances of the idiosyncratic noise terms, also known as uniquenesses or error bar.

```
# The variances of the idiosyncratic noise terms
barplot(fa_fx$uniquenesses, las=2, cex.names=0.8)
```




```
# Scatter plot of first two factor scores
plot(fa_fx$scores[,1:2], pch=21,
     bg=terrain.colors(119)[119:1],
     main="Currency factor scores")
legend("bottomright", fill=terrain.colors(3),
       legend=c("2010","2005","2001"), cex=0.6)
```



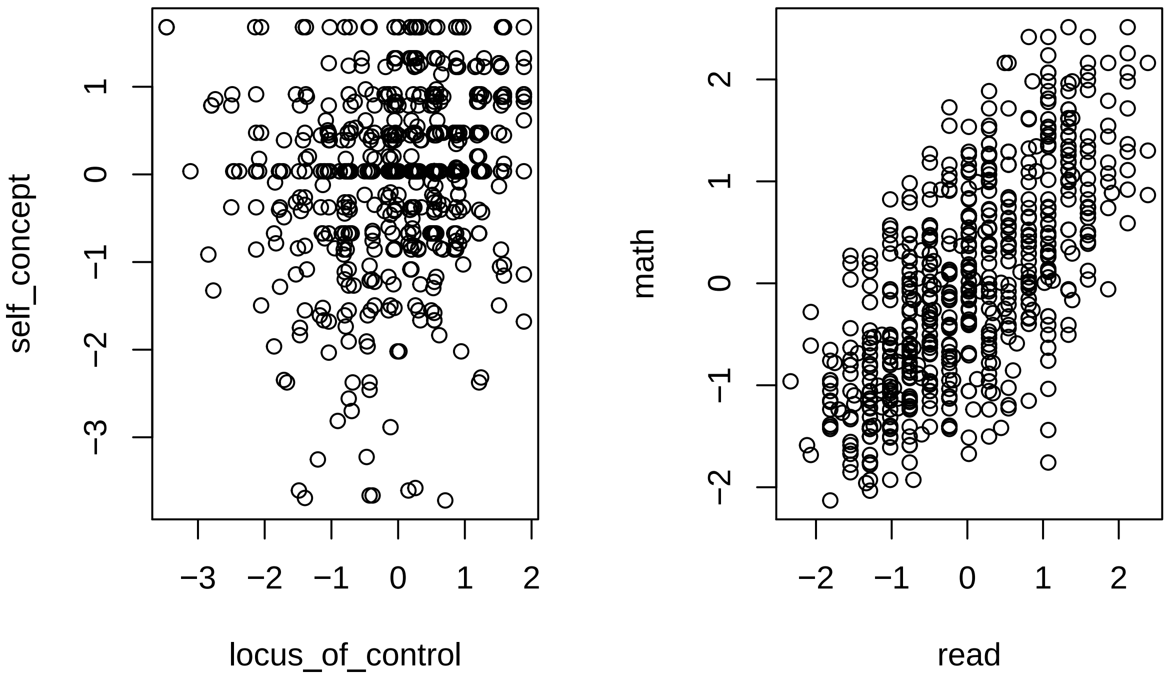
As we know, Cananical Correlation Analysis is used to see the correlation between two distinguished data sets. Let's take a look at the following example.

```
# Canonical correlation analysis
mmreg = read.csv('../STA380/data/mmreg.csv')
head(mmreg)
```

```
##   locus_of_control self_concept motivation read write math science female
## 1            -0.84        -0.24       1.00 54.8  64.5 44.5    52.6      1
## 2            -0.38        -0.47       0.67 62.7  43.7 44.7    52.6      1
## 3             0.89         0.59       0.67 60.6  56.7 70.5    58.0      0
## 4             0.71         0.28       0.67 62.7  56.7 54.7    58.0      0
## 5            -0.64         0.03       1.00 41.6  46.3 38.4    36.3      1
## 6             1.11         0.90       0.33 62.7  64.5 61.4    58.0      1
```

Split the data set to two seperate data sets, X and Y.

```
# Focus on two sets of variables
X = scale(mmreg[,c(1,2)], center=TRUE, scale=TRUE)
# x is for sychological variables and Y is for test scores
Y = scale(mmreg[,c(4,6)], center=TRUE, scale=TRUE)
par(mfrow=c(1,2))
plot(X)
plot(Y)
```



We can see that there isn't much correlation within X, but a stronger correlation within Y.

Let's try some random vectors to X and Y.

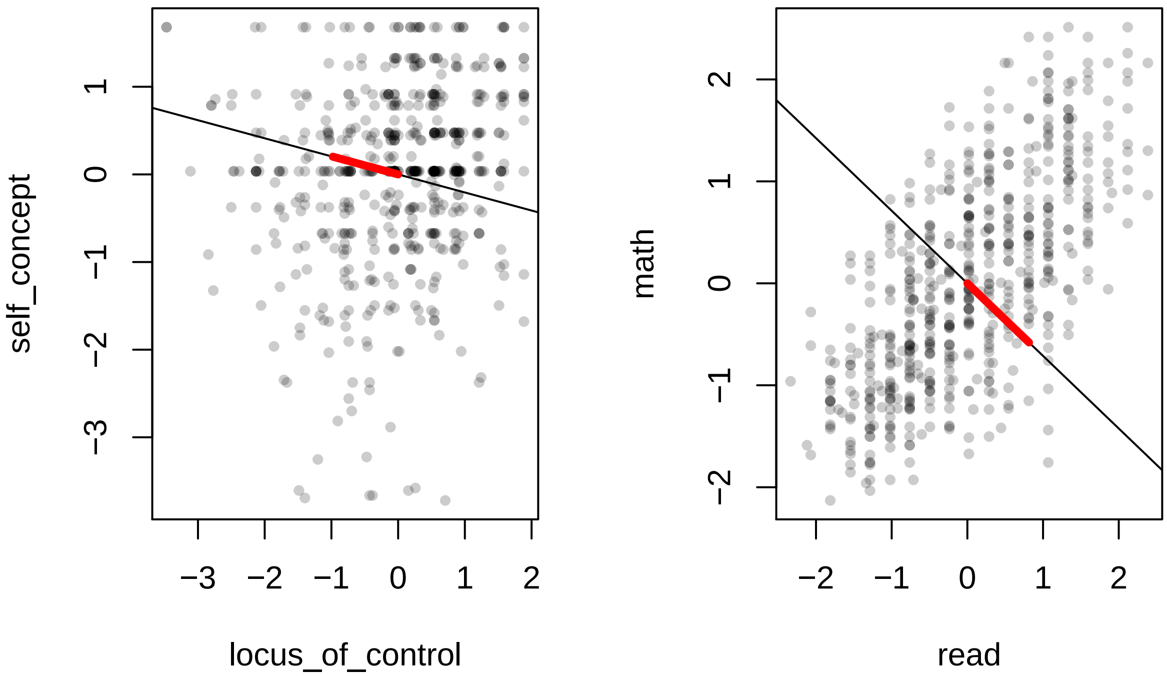
```
# Let's try some random canonical vectors
set.seed(2)
v_x = rnorm(2); v_x = v_x / sqrt(sum(v_x^2))
slope_x = v_x[2]/v_x[1]

v_y = rnorm(2); v_y = v_y / sqrt(sum(v_y^2))
slope_y = v_y[2]/v_y[1]

par(mfrow=c(1,2))

plot(X, pch=19, cex=0.6, col=rgb(0,0,0,0.2))
abline(0, slope_x)
segments(0, 0, v_x[1], v_x[2], col='red', lwd=4)

plot(Y, pch=19, cex=0.6, col=rgb(0,0,0,0.2))
abline(0, slope_y)
segments(0, 0, v_y[1], v_y[2], col='red', lwd=4)
```



Plot the positions of the projected points, and then plot the positions of the twe subsets and see the correlation.

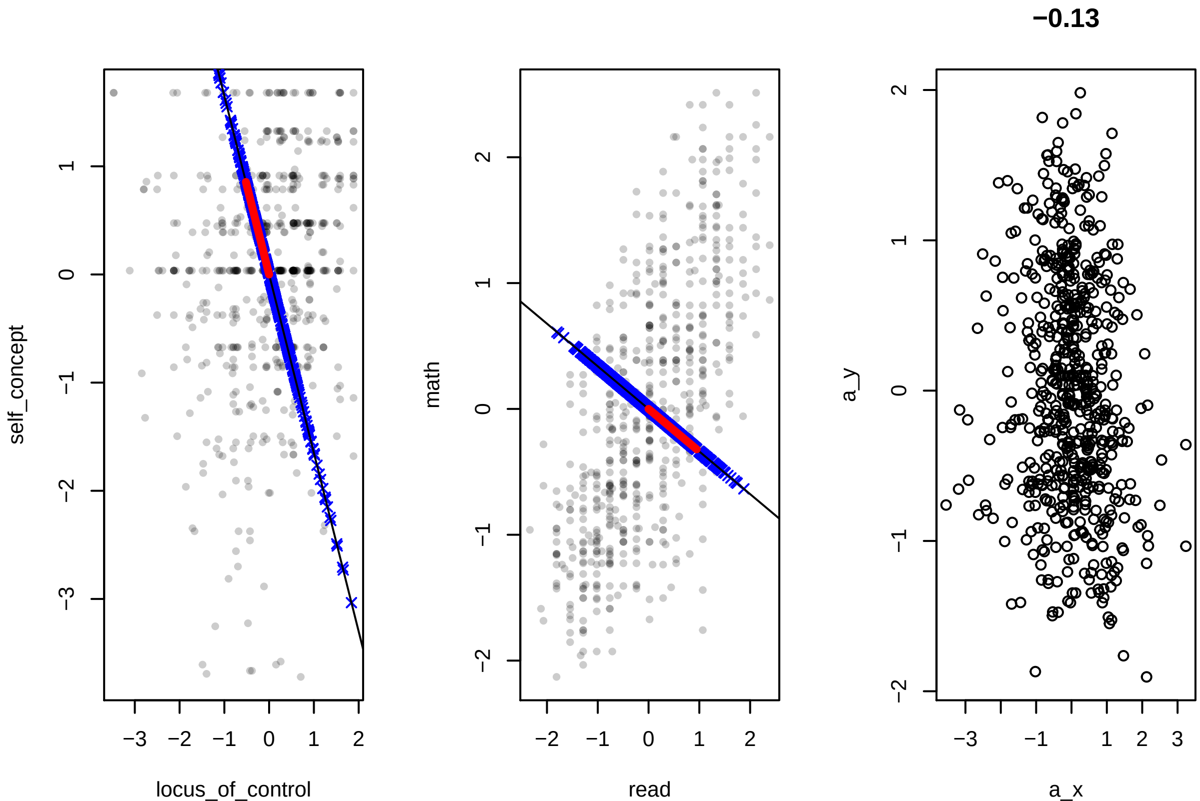
```r
# Now look at the projected points
par(mfrow=c(1,3))

# Random canonical vectors
v_x = rnorm(2); v_x = v_x / sqrt(sum(v_x^2))
slope_x = v_x[2]/v_x[1]

v_y = rnorm(2); v_y = v_y / sqrt(sum(v_y^2))
slope_y = v_y[2]/v_y[1]

plot(X, pch=19, cex=0.6, col=rgb(0,0,0,0.2))
a_x = X %*% v_x
points(a_x %*% v_x, pch=4, col='blue')
abline(0, slope_x)
segments(0, 0, v_x[1], v_x[2], col='red', lwd=4)

plot(Y, pch=19, cex=0.6, col=rgb(0,0,0,0.2))
a_y = Y %*% v_y
points(a_y %*% v_y, pch=4, col='blue')
abline(0, slope_y)
segments(0, 0, v_y[1], v_y[2], col='red', lwd=4)
plot(a_x, a_y, main=round(cor(a_x, a_y), 2))
```



A strongly negative correlation is as good as a strongly positive correlation.

```r
# Run CCA
cc1 = cancor(X, Y)
# xcoef is v
cc1$xcoef
```

```
##                             [,1]         [,2]
## locus_of_control -0.0409288550 -0.006684201
## self_concept      0.0004209878  0.041468934
```

```r
# ycoef is w
cc1$ycoef
```

```
##             [,1]        [,2]
## read -0.02806295  0.04808501
## math -0.01622630 -0.05325791
```

```r
cc1$cor
```

```
## [1] 0.390534026 0.001540878
```

CCA is designed to use X to predict Y while you want to reduce the features in both X and Y, as you are trying to retain the correlation between the two data sets while preserving the distinction of each of the data sets.