Continuous multivariate methods Project

Hanie Jalili

Data description:

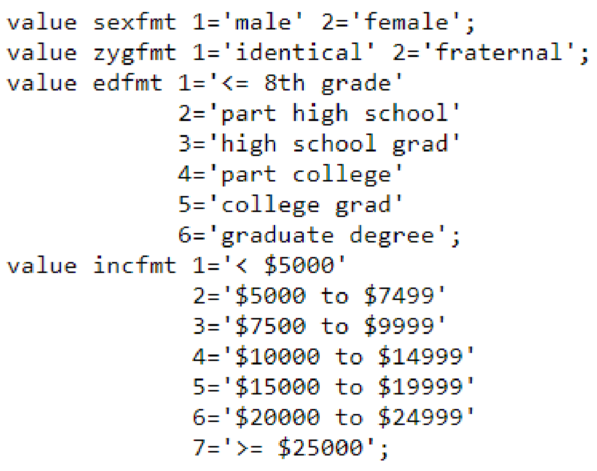
This data is about the University of Texas National Scholarship Eligibility Test scores for a set of twins, the scores are shown separately for each individual. (That is, for example, for the English course, we have two columns related to the score of the English test, each of which is for one grade.)

We have 15 columns and 740 rows.

Variable name	Description	Variable type
Sex	gender (categorical)	Independent
Zygosity	Whether twins are identical or not (categorical)	Independent
Courage	Mother's education level (categorical)	Independent
Faed	father's education level (categorical)	Independent
Fin	Financial level of the family (categorical)	Independent
English1	First person English language test score	Independent
Math1	First person math test score	Independent
SocSci1	First person social science exam score	Response
NatSci1	First person natural science test score	Independent
Vocab1	First person spelling score	Response
English2	The score of the second person's English language test	Independent
Math2	Second person's math test score	Independent
SocSci2	Second person social science test score	Response

NatSci2	The score of the natural science exam of the second person	Independent
Vocab2	Second person spelling score	Response

Classification of categorical variables:



```
value sexfmt 1='male' 2='female';
value zygfmt 1='identical' 2='fraternal';
value edfmt 1='<= 8th grade'
            2='part high school'
            3='high school grad'
            4='part college'
            5='college grad'
            6='graduate degree';
value incfmt 1='< $5000'
             2='$5000 to $7499'
             3='$7500 to $9999'
             4='$10000 to $14999'
             5='$15000 to $19999'
             6='$20000 to $24999'
             7='>= $25000';
```

In these steps, we call the data and specify the categorical variables.

```
library('carData')
library('car')
library(readxl)
twins_test_data <-
read_excel("E:/Users/ASUS/Documents/term8/multivariate
analysis/multivariate project/twins_test_data.xlsx")
attach(twins_test_data)

#prepare data
faed = as.factor(faed)
faminc = as.factor(faminc)
sex = as.factor(sex)
zygosity = as.factor(zygosity)
Courage= as.factor(courage)
twins_test_data$sex = sex
twins_test_data$faed = faed
twins_test_data$faminc = faminc
twins_test_data$courage= Courage
twins_test_data$zygosity = zygosity
summary(twins_test_data)
```

```
##  sex     zygosity moed    faed    faminc     English1         Math1
## 1:327 1:468 1: 49 1: 85 1: 88 Min. : 3.00 Min. : 4.00
## 2:441 2:300 2:96 2:95 2:197 1st Qu.:17.00 1st Qu.:17.00
##                   3:288   3:202   3:164   Median :20.00   Median
:21.00
##                   4:186   4:166   4:183   Mean   :19.62   Mean
:21.02
## 5:113 5:110 5: 73 3rd Qu.:23.00 3rd Qu.:25.00
##                   6: 36   6:110   6: 23   Max.   :30.00   Max.
:35.00
##                                   7: 40
## SocSci1 NatSci1 Vocab1 English2 Math2
## Min. : 4.00 Min. : 3.00 Min. : 4.00 Min. : 5.0 Min. : 3.00
## 1st Qu.:17.00 1st Qu.:16.00 1st Qu.:18.00 1st Qu.:17.0 1st
Qu.:17.00
##  Median :20.50   Median :21.00   Median :21.00   Median :20.0
Median :21.00
##  Mean   :20.51   Mean   :19.88   Mean   :20.91   Mean   :19.9
Mean   :21.49
## 3rd Qu.:24.00 3rd Qu.:24.00 3rd Qu.:24.00 3rd Qu.:23.0 3rd
Qu.:26.00
##  Max.   :32.00   Max.   :32.00   Max.   :32.00   Max.   :31.0
Max.   :35.00
##
## SocSci2 NatSci2 Vocab2
## Min. : 3.00 Min. : 4.0 Min. : 5.00
## 1st Qu.:17.00 1st Qu.:16.0 1st Qu.:18.00
##  Median :21.00   Median :21.0   Median :21.00
##  Mean   :20.87   Mean   :20.3   Mean   :21.22
## 3rd Qu.:25.00 3rd Qu.:25.0 3rd Qu.:24.00
##  Max.   :32.00   Max.   :33.0   Max.   :32.00
##
```

To select the dependent variables from among the grades of the courses, we first check which courses have a high correlation with each other and can be considered as dependent variables. For this, we first combine the grades of the twins' courses and calculate the correlation between their grades. The correlation matrix shows that the spelling course and the social science course have a high correlation with each other and the scores of these courses can be considered as dependent variables. The scores of the rest of the courses are considered as independent variables in the model.

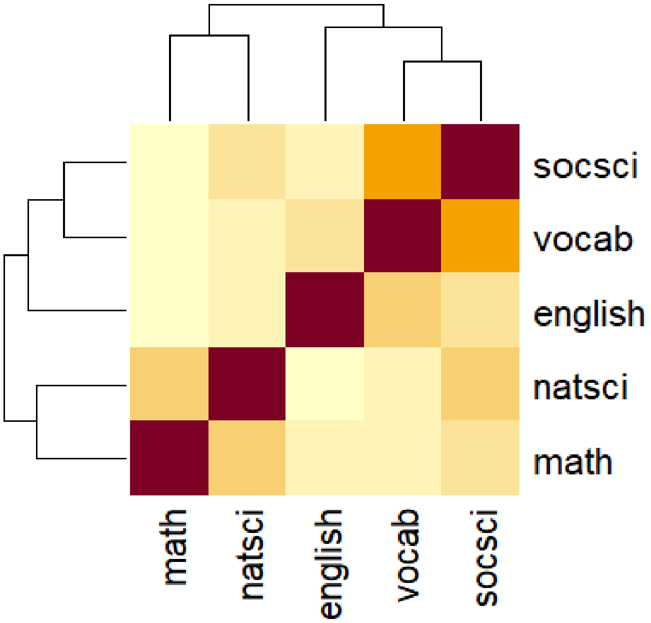
Note: Before calculating the correlation, we removed the unspecified data (na).

```
#correlation analysis
math = rbind(Math1,Math2)
```

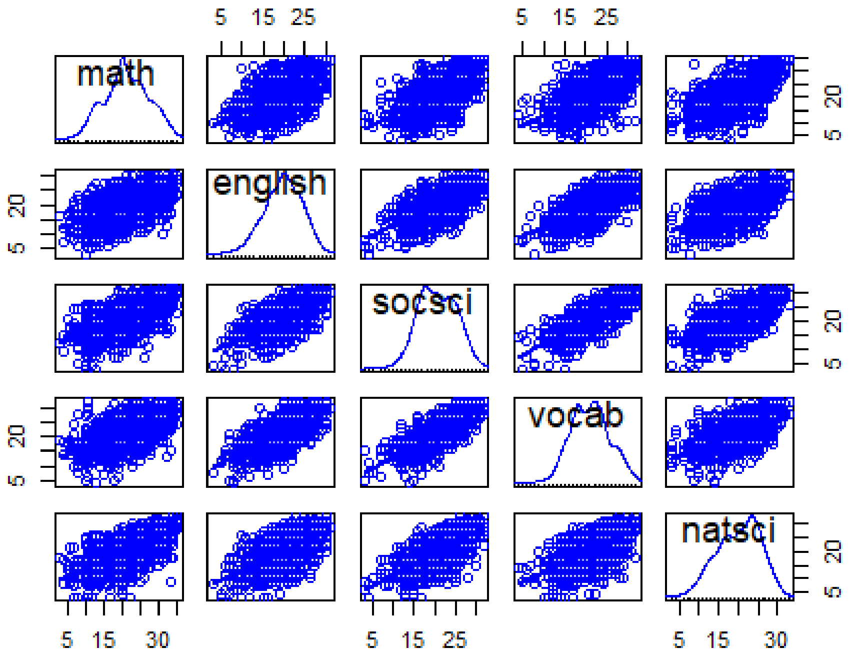
```
english = rbind(English1,English2)
socsci= rbind(SocSci1,SocSci2)
vocab = rbind(Vocab1,Vocab2)
natsci = rbind(NatSci1,NatSci2)
scores.data = matrix(c(math,english,socsci,vocab,natsci),ncol =
5,dimnames = list(c(),c("math","english","socci","vocab","natsci")))
scores.data.cor = as.data.frame(omit(scores.data))
cor(scores.data.cor)
```

```
## math english socsci vocab natsci
## math    1.0000000 0.5412699 0.6086261 0.5556737 0.6583328
## english 0.5412699 1.0000000 0.6339606 0.6692687 0.5783527
## socsci 0.6086261 0.6339606 1.0000000 0.7719540 0.6749258
## vocab   0.5556737 0.6692687 0.7719540 1.0000000 0.6004781
## natsci  0.6583328 0.5783527 0.6749258 0.6004781 1.0000000
```

```
heatmap(cor(scores.data.cor))
```



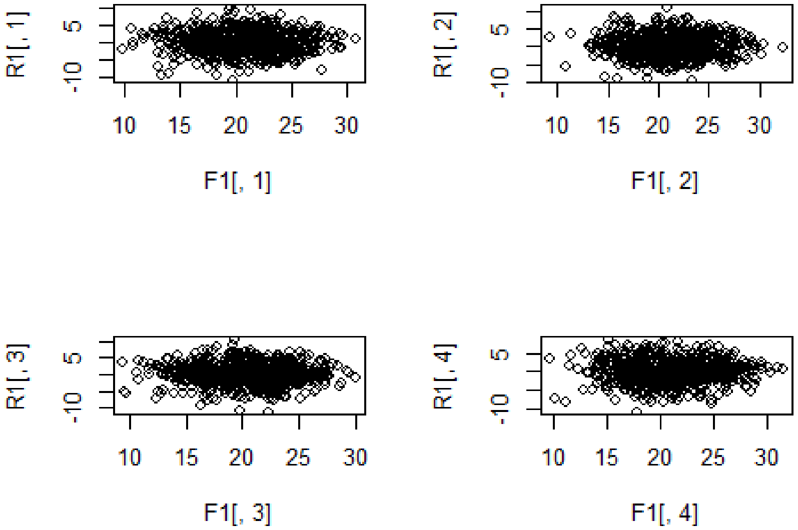
```
scatterplotMatrix(scores.data,smooth = FALSE,ellipse = TRUE)
```



According to the above explanations about the variables, now we apply a multivariate regression model.

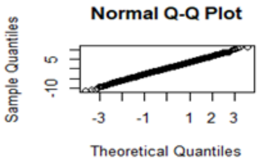
```
#regression model
L1=lm(cbind(Vocab1,Vocab2,SocSci1,SocSci2)~.,data=twins_test_data)
F1=L1$fitted.values
R1=L1$residuals
about(mfrow=c(2,2))
plot(F1[,1],R1[,1])
plot(F1[,2],R1[,2])
plot(F1[,3],R1[,3])
plot(F1[,4],R1[,4])
```

The graph of the residuals against the predicted values for the independent variables does not have any special pattern and this condition is desirable because it shows that the variance of the residuals is constant and this is one of the regression conditions.



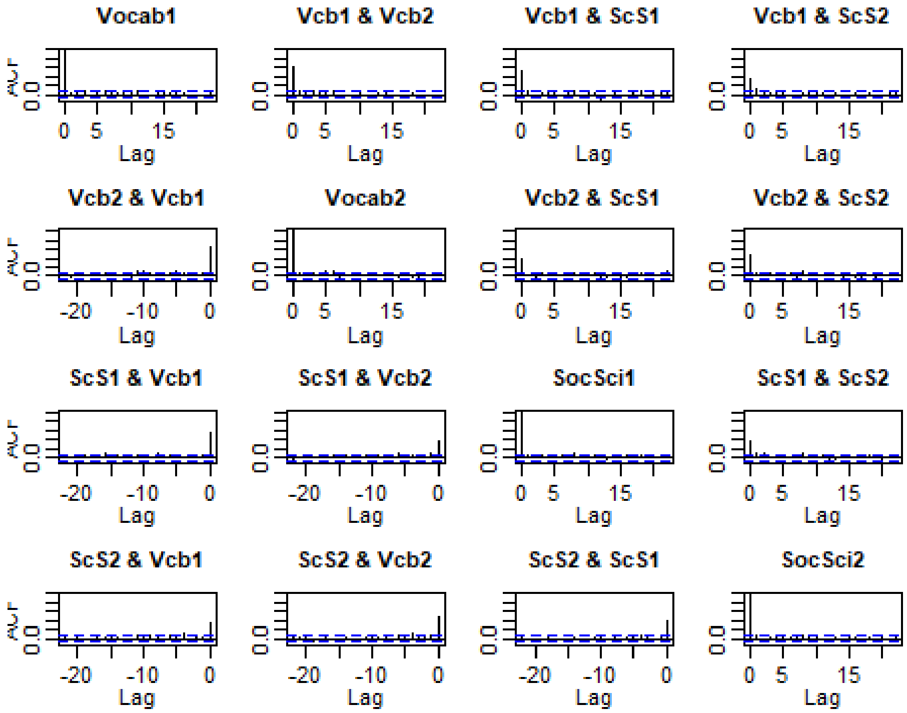
The graph of the distribution of the residuals as well as the Shapiro-Wilk test show that the distribution of the residuals is normal. In the Shapiro test, the assumption of normality of distribution of balances is not rejected.

```
qqnorm(R1)
shapiro.test(R1)
##  Shapiro-Wilk normality test
##
## data: R1
## W = 0.99949, p-value = 0.617
```



```
aoutcorr=acf(R1)
```

The autocorrelation diagram of the balances also shows the independence between the balances, which is also one of the conditions for fitting a suitable regression model.



By using the Wilkes test, which is one of the tests related to multivariate regression, we find that only the twins' grades in other subjects and the level of their father's education have an effect on all the dependent variables of the model, and the other independent variables do not have much effect on all the dependent variables.

```
#tests

Anova(L1,test.statistic = "Wilks")
## Type II MANOVA Tests: Wilks test statistic
##          Df test stat approx F num Df den Df    Pr(>F)
## sex       1   0.99441    1.040      4  740.0  0.385667
## zygosity  1   0.99473    0.980      4  740.0  0.417450
## courage 5 0.97642 0.887 20 2455.3 0.605023
## faed      5   0.95723    1.629     20 2455.3  0.038445 *
## faminc    6   0.97565    0.763     24 2582.8  0.786830
## English1  1   0.83795   35.776      4  740.0 < 2.2e-16 ***
## Math1 1 0.97619 4.513 4 740.0 0.001314 **
## NatSci1 1 0.88466 24.120 4 740.0 < 2.2e-16 ***
## English2 1 0.84853 33.025 4 740.0 < 2.2e-16 ***
## Math2 1 0.94413 10.948 4 740.0 1.252e-08 ***
## NatSci2   1   0.91692   16.762      4  740.0 3.671e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this part, using the update command, we applied another regression model without the variables "mother's education level" and "identical or not twins" on the data, and we want to compare it with the first regression model. It can be seen that the influential independent variables in this model are the same influential independent variables in the first model. (This refers to the scores of other courses and the variable of the father's education level.)

```
L2=update(L1,.~.-moed-zygosity)
Anova(L2,test.statistic = "Wilks")
```

```
##
## Type II MANOVA Tests: Wilks test statistic
##          Df test stat approx F num Df den Df    Pr(>F)
## sex       1   0.99460    1.013      4  746.0  0.399890
## faed      5   0.95635    1.677     20 2475.2  0.030271 *
## faminc    6   0.97542    0.777     24 2603.7  0.770405
## English1  1   0.84101   35.258      4  746.0 < 2.2e-16 ***
## Math1 1 0.97577 4.632 4 746.0 0.001066 **
## NatSci1 1 0.88871 23.354 4 746.0 < 2.2e-16 ***
## English2 1 0.84812 33.398 4 746.0 < 2.2e-16 ***
## Math2 1 0.94379 11.108 4 746.0 9.331e-09 ***
## NatSci2   1   0.91633   17.030      4  746.0 2.254e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

According to the following results and using Hotelling's estimator, we find that comparing the first and second models, the first model is a more suitable model for fitting the data.

```
anova(L1,L2,test = "Hotelling-Lawley")
```

```
## Analysis of Variance Table
##
## Model 1: cbind(Vocab1, Vocab2, SocSci1, SocSci2) ~ sex + zygosity +
moed +
##     faed + faminc + English1 + Math1 + NatSci1 + English2 + Math2 +
##     NatSci2
## Model 2: cbind(Vocab1, Vocab2, SocSci1, SocSci2) ~ sex + faed +
faminc +
##     English1 + Math1 + NatSci1 + English2 + Math2 + NatSci2
##   Res.Df Df Gen.var. Hotelling-Lawley approx F num Df den Df Pr(>F)
## 1    743       7.4895
## 2    749  6   7.4850         0.029963  0.92199     24   2954 0.5717
```

Using the following command, we want to test the null hypothesis of the influence of the variable on the math score of the twins. By using different statistics such as Hotelling, Wilkes, Pillay, etc., we find that this assumption is rejected and the math scores of the twins have an effect on all the dependent variables.

```
linearHypothesis(L1,c("Math1=0","Math2=0"))
```

```
## 
## Sum of squares and products for the hypothesis:
## Vocab1 Vocab2 SocSci1 SocSci2
## Vocab1 125.31957 56.73160 115.02581 63.73648
## Vocab2   56.73160 124.94422   59.67151 209.65003
## SocSci1 115.02581 59.67151 106.15946 72.34359
## SocSci2 63.73648 209.65003 72.34359 361.72069
## 
## Sum of squares and products for error:
## Vocab1 Vocab2 SocSci1 SocSci2
## Vocab1 7939.759 4736.713 4095.410 2512.751
## Vocab2  4736.713 7102.286 2671.441 3456.784
## SocSci1 4095.410 2671.441 7696.847 2824.236
## SocSci2 2512.751 3456.784 2824.236 7123.214
## 
## Multivariate Tests:
##                  Df test stat  approx F num Df den Df     Pr(>F)
## Pillai            2 0.0763255  7.350152      8   1482 1.2675e-09
***
## Wilks             2 0.9248118  7.373411      8   1480 1.1700e-09
***
## Hotelling-Lawley  2 0.0800713  7.396587      8   1478 1.0802e-09
***
## Roy               2 0.0593510 10.994767      4    741 1.1491e-08
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the end, it can be concluded that the first model is a suitable model for fitting the data of the problem and the regression conditions are also established in it.

PCA

In this section, we are going to apply PCA on continuous independent variables.

```
#PCA
library("usethis")
library("devtools")
library("ggplot2")
library("factoextra")
```

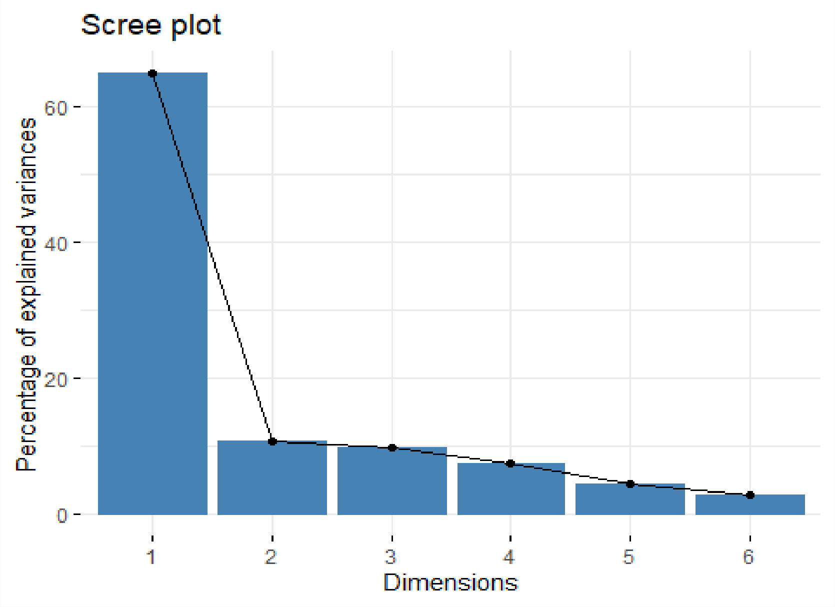
```
continuous_independent_variables =
twins_test_data[,c(-1,-2,-3,-4,-5,-8,-10,-13,-15)]
dim(continuous_independent_variables)
```

```
## [1] 768   6
```

```
pca = princomp(continuous_independent_variables,cor = FALSE,scores =
TRUE)
summary(pca)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4
Comp.5
## Standard deviation     10.9996621 4.4776978 4.28067723 3.70916968
2.88238621
## Proportion of Variance  0.6476594 0.1073242 0.09808734 0.07364468
0.04447258
## Cumulative Proportion   0.6476594 0.7549836 0.85307093 0.92671561
0.97118820
##                          Comp.6
## Standard deviation     2.3200174
## Proportion of Variance 0.0288118
## Cumulative Proportion  1.0000000
```

```
fviz_eig(pca)
```



According to the figure and explanation above, if we want to cover at least 70% of the total dispersion using principal components analysis, it seems that the first and second principal components are suitable for use and other principal components can be omitted. The first principal component alone covers approximately 64% and the second principal component alone covers approximately 10% of the total variance.

```
loadings_pca = as.data.frame.matrix(pca$loading) #coefficients of
variables
loadings_pca
```

```
##             Comp.1      Comp.2      Comp.3     Comp.4      Comp.5
Comp.6
## English1 0.3158796  0.39099148  0.39657617  0.4343670  0.10664212
0.62452051
## Math1 0.4779970 -0.68354514 0.14769793 0.3395589 0.35418410
-0.20426355
## NatSci1  0.4284434 -0.07601641  0.53633834 -0.4864218 -0.52898390
-0.08104807
## English2 0.2919576  0.55532246 -0.00609008  0.3209514 -0.03607793
-0.70854020
## Math2 0.4790469 -0.06241221 -0.69780397 0.1222263 -0.45865998
0.23319546
## NatSci2  0.4164733  0.24866623 -0.21514269 -0.5843541  0.60966099
0.07261141
```

In this section, you can check the influence of each continuous independent variable on all the main components. For example, in the first principal component, all independent variables have a positive effect, and the math course has the highest effect.

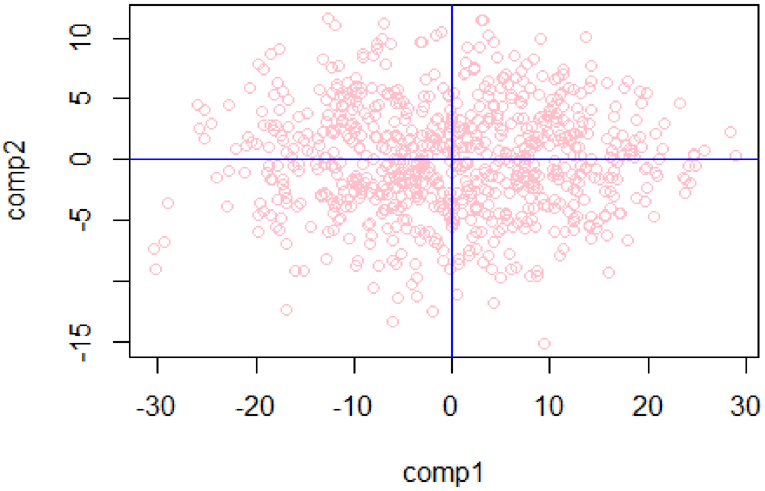
```
head(pca$scores)
```

```
##          Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6
## [1,] -16.888790 -3.611102  3.074103 -1.999348 -4.971372  2.096070
## [2,]  -7.108020 -2.132064  1.243780  4.740387 -1.475047  1.708014
## [3,] -20.935506  1.127374  1.170847 -1.376161 -6.945302 -2.515197
## [4,] -16.030218 -9.109327 -5.196634 -3.051336  4.310934  1.496562
## [5,]  -5.654093 -5.513247  5.078690 -4.310620  3.501118 -1.906432
## [6,]  -8.068550  2.639025 -1.734756  3.265304  3.013145  1.357778
```

We know that observations find new coordinates in the principal components. In the upper part, we have the new coordinates of the observations that we drew in the diagram below. The blue axes show the coordinates of the previous observations and the pink dots show the coordinates of the new observations.

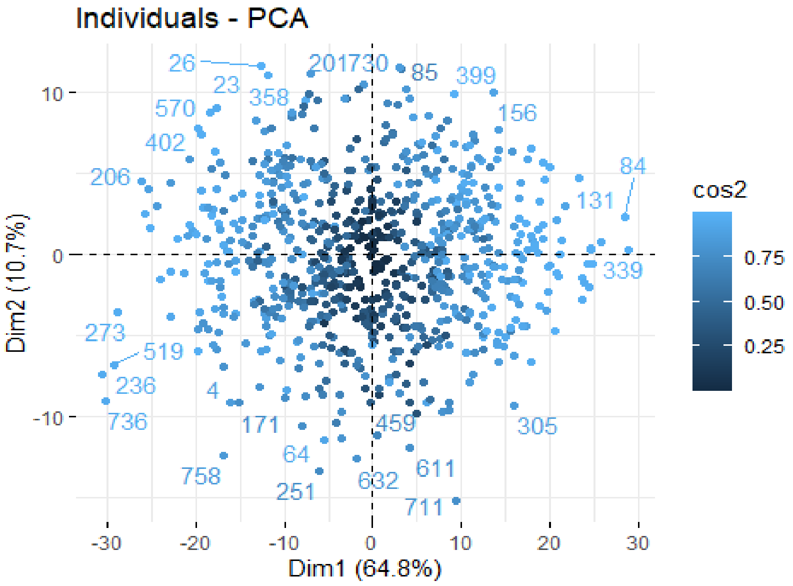
As it is known, the dispersion of points in the first principal component is more than the dispersion of points in the second principal component. This is consistent with the result we obtained in the previous section. (The first principal component covers 64% of the dispersion and the second principal component alone covers 10%.)

```
plot(pca$scores[,1],pca$scores[,2],
     xlab = "comp1", ylab = "comp2", col = "pink")
abline(h=0, col = "blue")
abline(v=0, col = "blue")
```

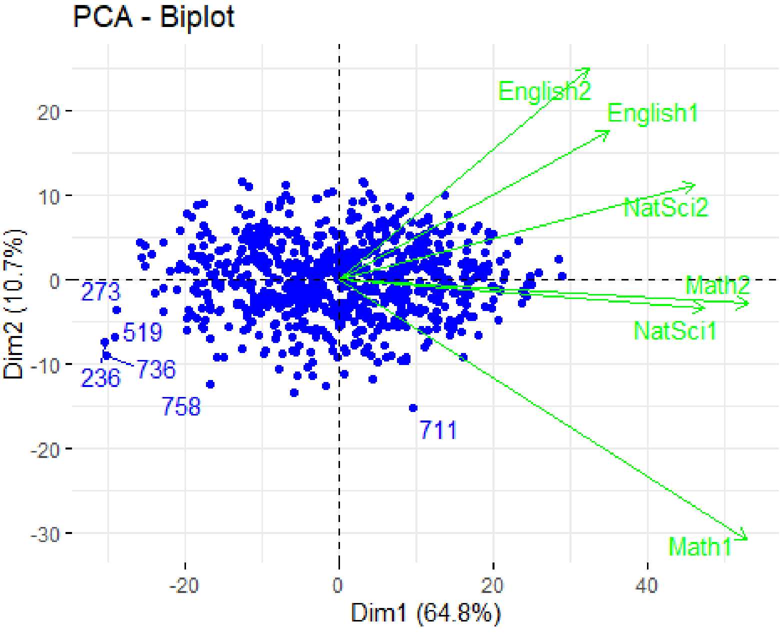


```
fviz_pca_ind(pca, col.ind = "cos2",repel = TRUE)
```

In the graphs below, the dotted lines represent the previous coordinates and the colored dots represent the new coordinates. In the second graph, it can be seen that the first principal component and the math score variable had the greatest impact on the total dispersion.



```
fviz_pca_biplot(pca, repel = TRUE, col.var = "green", col.ind =
"blue")
```



FACTOR ANALYSIS

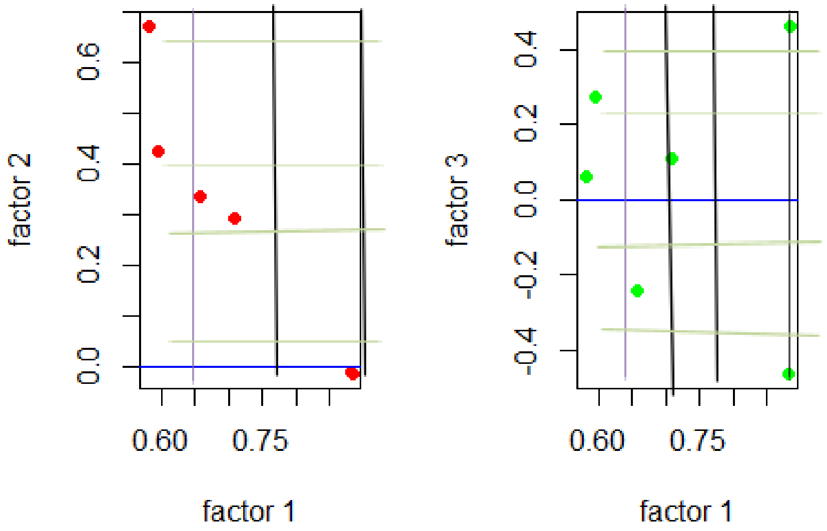
In this section, we intend to apply FACTOR ANALYSIS on continuous independent variables in order to reduce the dimension. For this purpose, we apply 3 different FACTOR ANALYSIS methods on the data and check each one.

```
#FACTOR ANALYSIS
fa1= factanal(continuous_independent_variables, 3, scores =
"regression",
               rotation = "none", cor = "pearson")
fa1
```

```
##
## Call:
## factanal(x = continuous_independent_variables, factors = 3, scores
= "regression",     rotation = "none", cor = "pearson")
##
## Uniquenesses:
## English1    Math1  NatSci1 English2    Math2  NatSci2
##    0.397    0.391    0.207    0.005    0.005    0.400
##
## Loadings:
##          Factor1 Factor2 Factor3
## English1  0.657   0.336  -0.243
## Math1 0.596 0.425 0.271
## NatSci1   0.580   0.673
## English2  0.883          -0.464
## Math2 0.884 0.461
## NatSci2   0.708   0.294   0.108
##
##                Factor1 Factor2 Factor3
## SS loadings      3.187   0.833   0.576
## Proportion Was 0.531 0.139 0.096
## Cumulative Var   0.531   0.670   0.766
##
## The degrees of freedom for the model is 0 and the fit was 0.0311
```

```
about(mfrow=c(1,2))
plot(loadings(fa1)[,1], loadings(fa1)[,2], pch=16, xlab="factor 1",
     ylab="factor 2", col="red")
abline(h=0, col="blue")
abline(v=0, col="blue")
plot(loadings(fa1)[,1], loadings(fa1)[,3], pch=16, xlab="factor 1",
     ylab="factor 3", col="green")
abline(h=0, col="blue")
abline(v=0, col="blue")
```

In the first method, using 3 factors, almost 76% of the variance was covered. The coefficients of the first factor were divided into 3 categories, the coefficients of the second factor into 4 categories and the coefficients of the third factor into 5 categories. In the figure below, the division of the coefficients of each factor is shown using green and purple lines. For the next methods, the coefficients (loadings) of the factors can be grouped in the same way. The farther the coefficients of a factor are from each other, the more suitable it is to reduce the dimension.



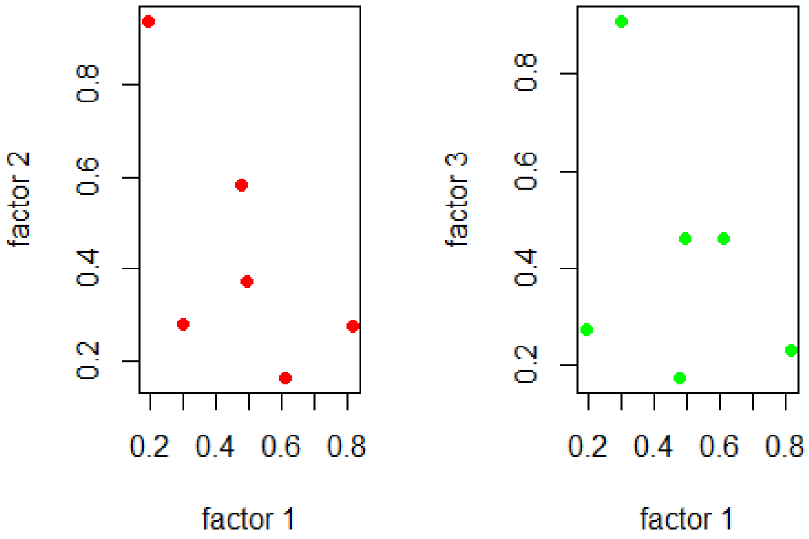
```
fa2= factanal(continuous_independent_variables, 3, scores =
"Bartlett",
               cor = "pearson")
fa2
```

```
##
## Call:
## factanal(x = continuous_independent_variables, factors = 3, scores
= "Bartlett",     cor = "pearson")
##
## Uniquenesses:
## English1    Math1  NatSci1 English2    Math2  NatSci2
##    0.397    0.391    0.207    0.005    0.005    0.400
##
## Loadings:
##          Factor1 Factor2 Factor3
## English1 0.481   0.585   0.173
## Math1 0.609 0.163 0.459
## NatSci1  0.815   0.274   0.232
## English2 0.195   0.940   0.272
## Math2 0.301 0.278 0.909
## NatSci2  0.498   0.373   0.461
```

```
##
##                 Factor1 Factor2 Factor3
## SS loadings       1.644   1.543   1.408
## Proportion Was 0.274 0.257 0.235
## Cumulative Var   0.274   0.531   0.766
##
## The degrees of freedom for the model is 0 and the fit was 0.0311
```

```
about(mfrow=c(1,2))
plot(loadings(fa2)[,1], loadings(fa2)[,2], pch=16, xlab="factor 1",
     ylab="factor 2", col="red")
abline(h=0, col="blue")
abline(v=0, col="blue")
plot(loadings(fa2)[,1], loadings(fa2)[,3], pch=16, xlab="factor 1",
     ylab="factor 3", col="green")
abline(h=0, col="blue")
abline(v=0, col="blue")
```

In the second method, using 3 factors, almost 76% of the variance was covered. The coefficients of the first factor were divided into 5 categories, the coefficients of the second factor into 4 categories and the coefficients of the third factor into 4 categories.

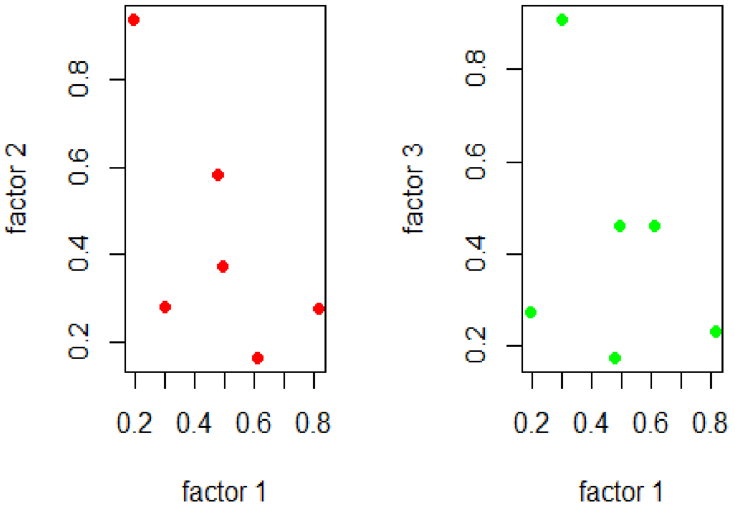


```
fa3= factanal(continuous_independent_variables, 3, scores =
"regression",
               rotation = "varimax", cor = "pearson")
fa3
```

```
##
## Call:
## factanal(x = continuous_independent_variables, factors = 3, scores
= "regression",     rotation = "varimax", cor = "pearson")
##
## Uniquenesses:
## English1    Math1  NatSci1 English2    Math2  NatSci2
##    0.397    0.391    0.207    0.005    0.005    0.400
##
## Loadings:
##          Factor1 Factor2 Factor3
## English1 0.481   0.585   0.173
## Math1 0.609 0.163 0.459
## NatSci1  0.815   0.274   0.232
## English2 0.195   0.940   0.272
## Math2 0.301 0.278 0.909
## NatSci2  0.498   0.373   0.461
##
##                Factor1 Factor2 Factor3
## SS loadings      1.644   1.543   1.408
## Proportion Was 0.274 0.257 0.235
## Cumulative Var   0.274   0.531   0.766
##
## The degrees of freedom for the model is 0 and the fit was 0.0311
```

```
about(mfrow=c(1,2))
plot(loadings(fa3)[,1], loadings(fa3)[,2], pch=16, xlab="factor 1",
     ylab="factor 2", col="red")
abline(h=0, col="blue")
abline(v=0, col="blue")
plot(loadings(fa3)[,1], loadings(fa3)[,3], pch=16, xlab="factor 1",
     ylab="factor 3", col="green")
abline(h=0, col="blue")
abline(v=0, col="blue")
```

In the third method, using 3 factors, almost 76% of the variance was covered. The coefficients of the first factor were divided into 5 categories, the coefficients of the second factor into 4 categories and the coefficients of the third factor into 4 categories.



FACTOR ANALYSIS (part 2)

Because the previous command to perform factor analysis on continuous independent variables was limited, (for example, if the number of variables is 4, it is not possible to produce 3 linear combinations of them.) We want to perform factor analysis with the new command. For this purpose, we apply 3 new FACTOR ANALYSIS methods to the data and check each one.

```
#FACTOR ANALYSIS PART 2
library(ggplot2)
library(psych)

library(hrbrthemes)
```

Fourth method: We assume that each of the variables is a linear combination of 6 hidden factors. The chosen rotate method is varimax and we used the variance-covariance matrix. We also obtained the data coordinates of the given rotate using the regression method.

In the output, the order of hidden factors is based on the amount of influence on the total variance. The third hidden factor has the largest share in covering the total variance (30 percent), so it is located in the first column. For example, in the section of non-standard coefficients, the coefficient of the third hidden factor in the linear combination of English1 variable is equal to 2.06. In the section of standard coefficients, the coefficient of the third hidden factor in the linear combination of English1 variable is equal to 0.43.

According to the output below, if we want to cover 80% of the total variance, we can settle for the first 3 latent factors.

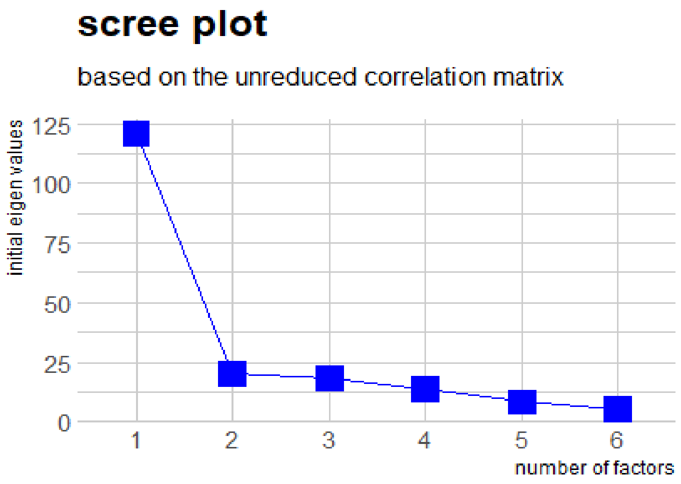
```
fa4= fa(continuous_independent_variables, nfactors  = 6, scores =
"regression",
               rotate = "varimax", covar = TRUE) #use variance
covariance matrix
fa4
```

```
## Factor Analysis using method =  minres
## Call: fa(r = continuous_independent_variables, nfactors = 6, rotate
= "varimax",
##     scores = "regression", covar = TRUE)
## Unstandardized loadings (pattern matrix) based upon covariance
matrix
## MR3 MR1 MR2 MR4 MR5 MR6 h2 u2 H2 U2
## English1 2.06 0.61 3.5 0.68  0.80   0 18 5.1 0.78 0.22
## Math1 4.23 3.22 1.2 0.72 0.51 0 30 9.9 0.75 0.25
## NatSci1  3.98 1.02 1.7 2.22 -0.35   0 25 8.3 0.75 0.25
## English2 0.52 1.35 3.7 1.16 -0.53   0 17 3.4 0.83 0.17
## Math2 1.73 4.62 1.8 2.00 -0.11 0 32 7.5 0.81 0.19
## NatSci2  1.77 2.14 1.8 3.63  0.08   0 24 6.9 0.78 0.22
##
## MR3 MR1 MR2 MR4 MR5 MR6
## SS loadings           44.37 39.51 36.22 24.45 1.32 0.00
## Proportion Var 0.24 0.21 0.19 0.13 0.01 0.00
## Cumulative Var         0.24  0.45  0.64  0.77 0.78 0.78
## Proportion Explained   0.30  0.27  0.25  0.17 0.01 0.00
## Cumulative Proportion  0.30  0.58  0.82  0.99 1.00 1.00
##
##  Standardized loadings (pattern matrix)
## item MR3 MR1 MR2 MR4 MR5 MR6 h2 u2
## English1    1 0.43 0.13 0.72 0.14  0.17   0 0.78 0.22
## Math1 2 0.67 0.51 0.19 0.11 0.08 0 0.75 0.25
## NatSci1     3 0.69 0.18 0.30 0.38 -0.06   0 0.75 0.25
## English2    4 0.11 0.30 0.81 0.26 -0.12   0 0.83 0.17
## Math2 5 0.28 0.74 0.29 0.32 -0.02 0 0.81 0.19
## NatSci2     6 0.32 0.39 0.32 0.65  0.01   0 0.78 0.22
##
## MR3 MR1 MR2 MR4 MR5 MR6
## SS loadings     1.30 1.09 1.49 0.78 0.05 0.00
## Proportion Var 0.22 0.18 0.25 0.13 0.01 0.00
## Cumulative Var  0.22 0.40 0.65 0.77 0.78 0.78
## How. Var factor 0.28 0.51 0.82 0.99 1.00 1.00
```

```
##
## Mean item complexity =  2.1
## Test of the hypothesis that 6 factors are sufficient.
##
## df null model =  15  with the objective function =  163.93 with Chi
Square =  125267.3
## df of  the model are -6  and the objective function was  0
##
## The root mean square of the residuals (RMSR) is  0
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  768 with the empirical chi square  0  with
prob <  NA
## The total n.obs was  768  with Likelihood Chi Square =  0  with
prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
## MR3 MR1 MR2 MR4 MR5 MR6
## Correlation of (regression) scores with factors   0.83 0.82 0.89
0.74  0.42   0
## Multiple R square of scores with factors          0.70 0.67 0.80
0.55  0.17   0
## Minimum correlation of possible factor scores     0.39 0.35 0.59
0.10 -0.65  -1
```

```
n_factors4 = length(fa4$e.values)
scree4 = data.frame(factor_n = as.factor(1:n_factors4),
                   own_values = fa4$e.values)
ggplot(scree4, aes(x = factor_n, and = own_values,group = 1))+
  geom_line( color="blue") +
  geom_point(shape=22, color="blue", fill="blue", size=6) +
  theme_itself() +
  xlab("number of factors") +
  by"initial eigen values") +
  labs(title = "scree plot",
       subtitle = "based on the unreduced correlation matrix")
```

Using eigenvalues, we drew the following diagram. According to this diagram, it can be said that two hidden factors are enough to cover the total variance.



The fifth method: We assume that each of the variables is a linear combination of 6 hidden factors. In this method, we do not rotate the data and use the variance-covariance matrix.

In the output, the order of hidden factors is based on the amount of influence on the total variance. The first hidden factor has the largest share in covering the total variance (78 percent), so it is located in the first column. For example, in the section of non-standard coefficients, the coefficient of the first hidden factor in the linear combination of English1 variable is equal to 3.4, which is a large value. In the section of standard coefficients, the coefficient of the first hidden factor in the linear combination of English1 variable is equal to 0.72.

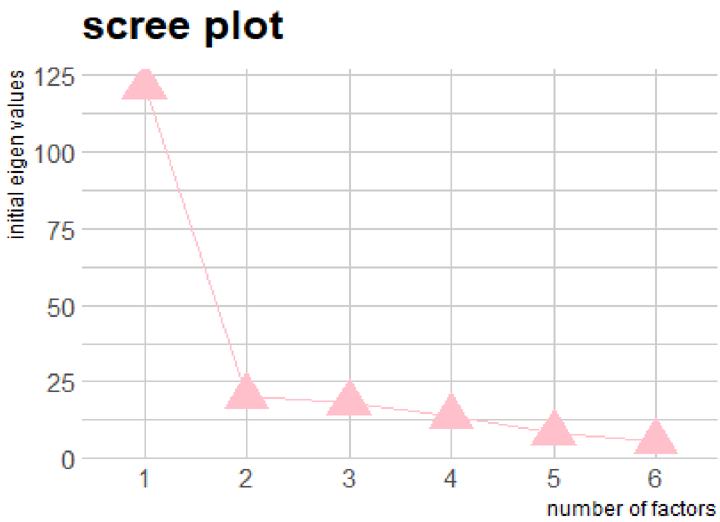
According to the output below, if we want to cover almost 80% of the total variance, we can settle for the first latent factor.

```
fa5= fa(continuous_independent_variables, nfactors  = 6, scores =
"regression",
         rotate = "none", covar = TRUE)
fa5
```

```
## Factor Analysis using method =  minres
## Call: fa(r = continuous_independent_variables, nfactors = 6, rotate
= "none",
##     scores = "regression", covar = TRUE)
## Unstandardized loadings (pattern matrix) based upon covariance
matrix
## MR1 MR2 MR3 MR4 MR5 MR6 h2 u2 H2 U2
## English1 3.4  1.87  1.22  0.73 -0.56   0 18 5.1 0.78 0.22
## Math1 5.0 -1.88 1.00 1.05 -0.07 0 30 9.9 0.75 0.25
## NatSci1  4.5 -0.30  1.69 -1.11  0.48   0 25 8.3 0.75 0.25
## English2 3.2  2.41 -0.64  0.51  0.50   0 17 3.4 0.83 0.17
## Math2 5.1 -0.88 -2.07 0.66 0.07 0 32 7.5 0.81 0.19
## NatSci2  4.5  0.21 -0.94 -1.73 -0.42   0 24 6.9 0.78 0.22
##
## MR1 MR2 MR3 MR4 MR5 MR6
## SS loadings           113.67 13.74 10.90 6.57 0.97 0.00
## Proportion Var 0.61 0.07 0.06 0.04 0.01 0.00
## Cumulative Var          0.61  0.68  0.74 0.77 0.78 0.78
## Proportion Explained    0.78  0.09  0.07 0.05 0.01 0.00
## Cumulative Proportion   0.78  0.87  0.95 0.99 1.00 1.00
##
##  Standardized loadings (pattern matrix)
## item MR1 MR2 MR3 MR4 MR5 MR6 h2 u2
## English1    1 0.72  0.39  0.26  0.15 -0.12   0 0.78 0.22
## Math1 2 0.78 -0.30 0.16 0.17 -0.01 0 0.75 0.25
## NatSci1     3 0.79 -0.05  0.29 -0.19  0.08   0 0.75 0.25
## English2    4 0.71  0.53 -0.14  0.11  0.11   0 0.83 0.17
## Math2 5 0.82 -0.14 -0.33 0.11 0.01 0 0.81 0.19
## NatSci2     6 0.80  0.04 -0.17 -0.31 -0.07   0 0.78 0.22
##
## MR1 MR2 MR3 MR4 MR5 MR6
## SS loadings     3.57 0.55 0.33 0.21 0.04 0.00
## Proportion Var 0.60 0.09 0.06 0.03 0.01 0.00
## Cumulative Var  0.60 0.69 0.74 0.78 0.78 0.78
## How. Var factor 0.76 0.88 0.95 0.99 1.00 1.00
##
## Mean item complexity =  1.6
## Test of the hypothesis that 6 factors are sufficient.
##
## df null model =  15  with the objective function =  163.93 with Chi
Square =  125267.3
## df of  the model are -6  and the objective function was  0
##
## The root mean square of the residuals (RMSR) is  0
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  768 with the empirical chi square  0  with
prob <  NA
## The total n.obs was  768  with Likelihood Chi Square =  0  with
prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
## MR1 MR2 MR3 MR4 MR5
## Correlation of (regression) scores with factors     0.97 0.85 0.78
0.69  0.39
## Multiple R square of scores with factors            0.94 0.72 0.61
0.48  0.15
## Minimum correlation of possible factor scores       0.88 0.44 0.21
-0.05 -0.70
##MR6
## Correlation of (regression) scores with factors       0
## Multiple R square of scores with factors              0
## Minimum correlation of possible factor scores        -1

n_factors5 = length(fa5$e.values)
scree5 = data.frame(factor_n = as.factor(1:n_factors5),
                    own_values = fa5$e.values)
ggplot(scree5, aes(x = factor_n, and = own_values,group = 1))+
  geom_line( color="pink") +
  geom_point(shape=24, color="pink", fill="pink", size=6) +
  theme_itself() +
  xlab("number of factors") +
  by"initial eigen values") +
  labs(title = "scree plot")
```

According to the diagram below, it can be said that two hidden factors are enough to cover the total variance.



Sixth method: We assume that each of the variables is a linear combination of 6 hidden factors. The selection rotate method is varimax and we used the correlation matrix. We also obtained the data coordinates of the given rotate using the regression method.

In the output, the order of hidden factors is based on the amount of influence on the total variance. The second hidden factor has the largest share in covering the total variance (32 percent), so it is located in the first column. For example, in the section of standard coefficients, the coefficient of the second hidden factor in the linear combination of English1 variable is equal to 0.75.

According to the output below, if we want to cover almost 80% of the total variance, we can settle for the second, fifth, and third latent factors.

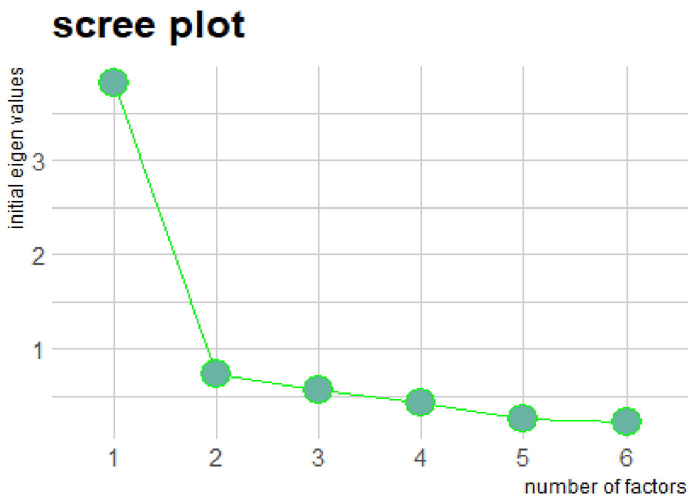
```
fa6 = fa(continuous_independent_variables, nfactors  = 6, scores =
"regression",
         rotate = "varimax")
fa6
```

```
## Factor Analysis using method =  minres
## Call: fa(r = continuous_independent_variables, nfactors = 6, rotate
= "varimax",
##     scores = "regression")
## Standardized loadings (pattern matrix) based upon correlation
matrix
## MR2 MR5 MR3 MR1 MR4 MR6 h2 u2 com
## English1 0.75 0.39 0.10 0.19 -0.06   0 0.77 0.23 1.7
## Math1 0.20 0.75 0.38 0.17 -0.03 0 0.77 0.23 1.8
## NatSci1  0.33 0.58 0.15 0.51  0.09   0 0.74 0.26 2.8
## English2 0.77 0.09 0.36 0.22  0.06   0 0.79 0.21 1.7
## Math2 0.27 0.35 0.76 0.23 0.01 0 0.82 0.18 1.9
## NatSci2  0.33 0.26 0.47 0.60 -0.05   0 0.76 0.24 2.9
##
## MR2 MR5 MR3 MR1 MR4 MR6
## SS loadings           1.49 1.24 1.10 0.80 0.02 0.00
## Proportion Var 0.25 0.21 0.18 0.13 0.00 0.00
## Cumulative Var        0.25 0.46 0.64 0.77 0.77 0.77
## Proportion Explained  0.32 0.27 0.24 0.17 0.00 0.00
## Cumulative Proportion 0.32 0.59 0.82 1.00 1.00 1.00
##
## Mean item complexity =  2.1
## Test of the hypothesis that 6 factors are sufficient.
##
## df null model =  15  with the objective function =  3.32 with Chi
Square =  2538.78
## df of  the model are -6  and the objective function was  0
```

```
##
## The root mean square of the residuals (RMSR) is  0
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  768 with the empirical chi square  0  with
prob <  NA
## The total n.obs was  768  with Likelihood Chi Square =  0  with
prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.006
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
## MR2 MR5 MR3 MR1 MR4 MR6
## Correlation of (regression) scores with factors   0.88 0.83 0.84
0.73  0.26   0
## Multiple R square of scores with factors          0.77 0.70 0.70
0.53  0.07   0
## Minimum correlation of possible factor scores     0.53 0.39 0.41
0.06 -0.86  -1
```

```
n_factors6 = length(fa6$e.values)
scree6 = data.frame(factor_n = as.factor(1:n_factors6),
                    own_values = fa6$e.values)
ggplot(scree6, aes(x = factor_n, and = own_values,group = 1)) +
  geom_line(color="green") +
  geom_point(shape=21, color="green", fill="#69b3a2", size=6) +
  theme_itself() +
  xlab("number of factors") +
  by"initial eigen values") +
  labs(title = "scree plot")
```

According to the diagram below, it can be said that two hidden factors are enough to cover the total variance.



CCA

Canonical Correlation Analysis (CCA) is used when we want to maximize the correlation between two data sets. In fact, we want to know what linear combination of these two data sets makes the correlation between them to be the highest possible. For this purpose, we used continuous independent variables and divided them into two categories to have two data sets.

```
#Approx
library(splines)
library(Matrix)
library(fds)

library(rainbow)
library(MASS)
library(pcaPP)
library(RCurl)
library(deSolve)
library(lattice)
library(graphics)
library(fda)

library(viridis)

library(viridisLite)
library(base)
library(spam)

library(fields)
library(CCA)
library(exchange)

library(vegan)

head(continuous_independent_variables)
##   English1 Math1 NatSci1 English2 Math2 NatSci2
##      <dbl> <dbl>   <dbl>    <dbl> <dbl>   <dbl>
## 1       14    13      18       11    14      10
## 2       20    20      16       17    19      13
## 3       11     8      16       16    13       8
## 4        9    19      10        8    16      17
## 5       15    23      21       15    13      20
## 6       20    17      12       19    18      18

x1= continuous_independent_variables[,1:3]
x2= continuous_independent_variables[,4:6]
```

X1: first grade math, language and natural science scores (three columns)

X2: math, language and natural science scores of the second term (three columns)

We perform canonical correlation analysis with two different functions in R: CC1 is formed with the CANCOR command and CC2 is formed with the command available in the CCA (CC) package.

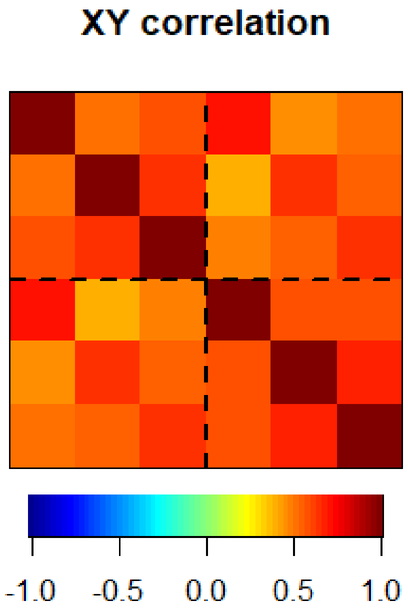
```
CC1= cancor(x1,x2)
cc2 = cc(x1,x2)
correl = matcor(x1,x2)
img.matcor(correl,type = 1)
```

The image below shows the variance-covariance matrix of the data sets. The green part of the variance-covariance matrix of the data set X1 and the blue part of the variance-covariance matrix of the data set X2 and the red parts of the covariance matrix between the two data sets X1 and X2.

According to the graph map, it can be said that the closer the graph is to the brown color, the higher the covariance between the variables. Therefore, it can be concluded:

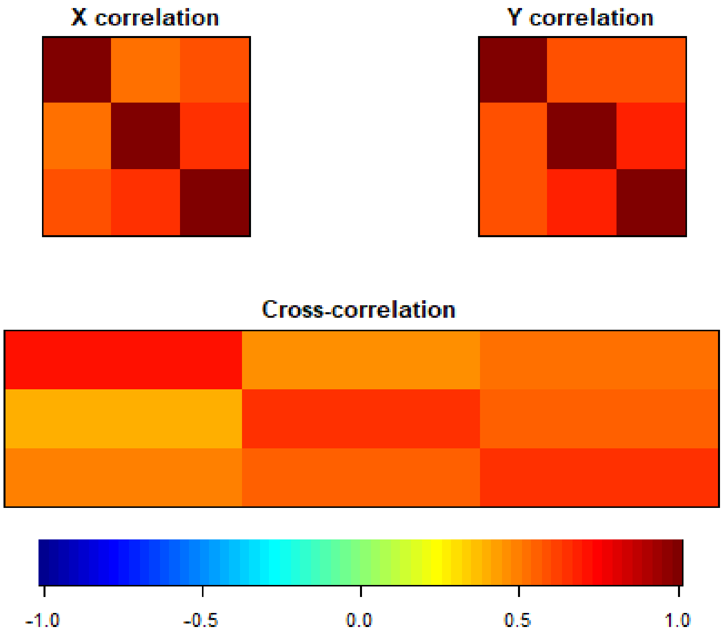
- The variables of the X1 data set are highly correlated with each other. That is, the scores of different courses are related to each other for the same grade.
- The variables of the X2 data set are also highly correlated with each other.
- The data set variables X1 and X2 are also highly correlated with each other and this is favorable for us in the CCA topic because we seek to maximize the

correlation between the two data sets. This shows that the scores of different courses of the twins are related to each other.



```
img.matcor(correl,type = 2)
```

The diagram below is the same as the diagram above, which is drawn separately for each data set and their effect on each other.



The following command is related to CC1 and shows that there is a linear combination of two data sets that makes the correlation between them as high as possible (75%). 3 different linear combinations are displayed in this command, each independent and perpendicular to each other. For example, if we leave the first linear combination aside, among the remaining independent and orthogonal linear combinations, the second linear combination has the highest correlation (52%).

```
cc1$cor
```

```
## [1] 0.7507408 0.5235511 0.2696782
```

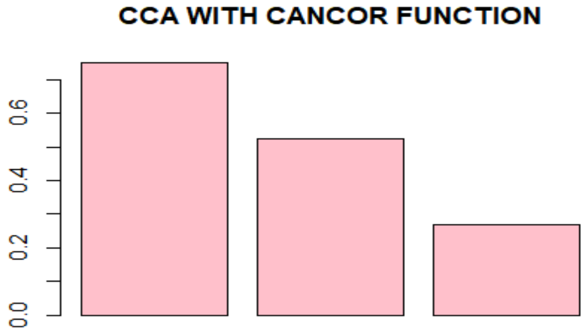
The following command is related to CC2 and its output is equal to CC1.

```
cc2$cor
```

```
## [1] 0.7507408 0.5235511 0.2696782
```

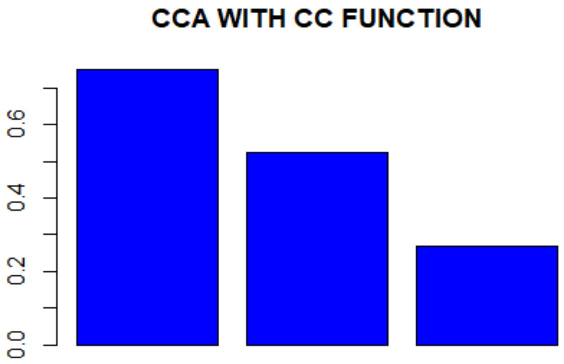
The diagram below shows the correlation of linear combinations obtained from CC1, whose output interpretation is the same as the ones mentioned in the above section.

```
barplot(cc1$cor, main = "CCA WITH CANCOR FUNCTION", col = "pink")
```



```
barplot(cc2$cor, main = "CCA WITH CC FUNCTION", col = "blue")
```

The graph below plots the correlation of linear combinations obtained from CC2.



The following output corresponds to the coefficients of variables of dataset X1 in CC1, displayed for all linear combinations. We know that every linear combination contains a combination of the first data set and the second data set. For example, it can be said that in the first linear combination CC1, the coefficient of the English lesson in the first data set (X1) is equal to the number `-0.004072042`.

```
cc1$xcoef
```

```
##                  [,1]         [,2]         [,3]
## English1 -0.004072042  0.007720015  0.003859869
## Math1 -0.001620082 -0.005507904 0.005023178
## NatSci1  -0.002189120 -0.001077307 -0.008507155
```

In the first linear combination of CC2, the coefficient of the English lesson in the first data set (X1) is equal to the number `-0.11277425`.

```
cc2$xcoef
```

```
##                 [,1]        [,2]       [,3]
## English1 -0.11277425  0.21380399 -0.1068982
## Math1 -0.04486779 -0.15254012 -0.1391157
## NatSci1  -0.06062716 -0.02983575  0.2356037
```

In the first linear combination of CC1, the coefficient of the English course in the second data set (X2), represented here by the symbol Y, is equal to `-0.003785535`.

```
cc1$ycoef
```

```
##                  [,1]         [,2]         [,3]
## English2 -0.003785535  0.008882575  0.003166737
## Math2 -0.001648490 -0.005567098 0.005772791
## NatSci2  -0.002616678 -0.001309038 -0.008771076
```

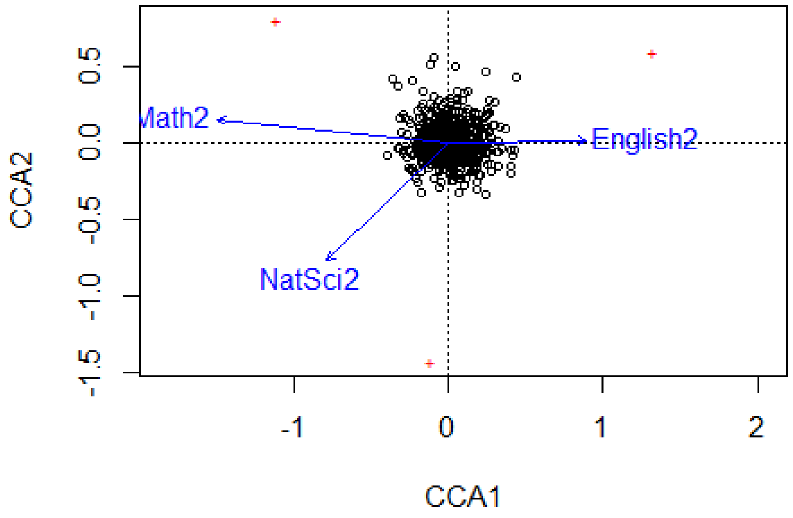
In the first linear combination of CC2, the coefficient of the English course in the second data set (X2), which is represented here by the symbol Y, is equal to `-0.10483950`.

```
cc2$ycoef
```

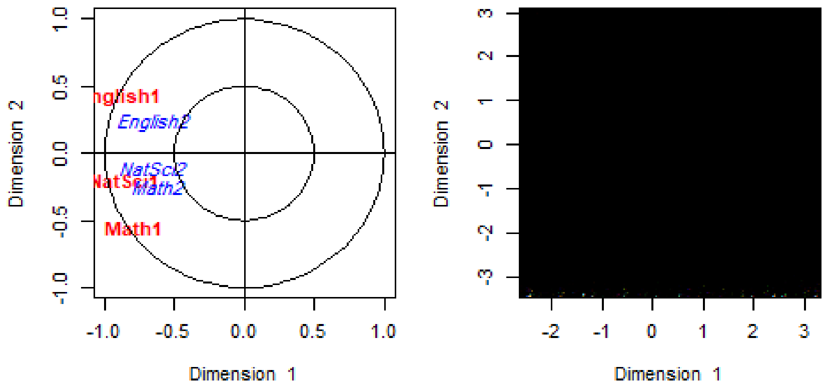
```
##                  [,1]        [,2]        [,3]
## English2 -0.10483950  0.24600082 -0.08770204
## Math2 -0.04565454 -0.15417947 -0.15987610
## NatSci2  -0.07246827 -0.03625349  0.24291289
```

CC3 is a conventional correlation analysis using the CCA command. In the diagram below, all three linear combinations are perpendicular to each other.

```
cc3= cca(x1,x2)
plot(cc3, scaling = 1)
```



```
plt.cc(cc2, var.label = TRUE, ind.names =
continuous_independent_variables)
```



LDA

A linear classifier is a classification method used to separate labeled data. The way lda works is that it plots the data in one dimension on a line and tries to maximize the distance between the means of the communities we want to separate from each other. (taking into account the amount of data dispersion)

```
# LDA (Linear Discriminant Analysis)
library(MASS)
```

Here we want to apply lda on the variable categories "faminc" which is related to the household income of the twins. This variable has 7 categories. We want to know in which level the twins are placed in the household income categories with different gender, education level of parents and grades. For this, we use the lda command.

```
m1= in l(faminc~.,data = twins_test_data)
m1
```

```
## Call:
## lda(faminc ~ ., data = twins_test_data)
##
```

SectionPrior probabilities of groups calculates the prior probability for each of the income categories.

```
## Prior probabilities of groups:
##          1          2          3          4          5          6
```

```
7
## 0.11458333 0.25651042 0.21354167 0.23828125 0.09505208 0.02994792
0.05208333
##
```

In the following section, the average of each of the variables is calculated for all categories of the faminc variable.

```
## Group means:
##        sex2 zygosity2      moed2     moed3     moed4      moed5
moed6
## 1 0.5568182 0.3181818 0.27272727 0.3068182 0.1250000 0.04545455
0.01136364
## 2 0.5685279 0.3807107 0.19796954 0.4111675 0.1928934 0.07614213
0.02538071
## 3 0.5975610 0.4024390 0.12195122 0.4634146 0.2256098 0.12804878
0.03048780
## 4 0.5136612 0.4207650 0.04918033 0.3879781 0.2732240 0.20218579
0.07650273
## 5 0.5753425 0.3972603 0.04109589 0.2191781 0.3835616 0.30136986
0.02739726
## 6 0.6086957 0.3913043 0.04347826 0.2173913 0.2608696 0.34782609
0.13043478
## 7 0.8000000 0.4000000 0.00000000 0.3000000 0.4000000 0.15000000
0.15000000
##        faed2     faed3      faed4      faed5      faed6 English1
Math1
## 1 0.23863636 0.2500000 0.11363636 0.02272727 0.01136364 18.28409
18.04545
## 2 0.17258883 0.3857868 0.19289340 0.05076142 0.03553299 18.92893
20.10152
## 3 0.11585366 0.2804878 0.28658537 0.12804878 0.09146341 19.42073
20.95122
## 4 0.08196721 0.2185792 0.27322404 0.21857923 0.19125683 19.93443
21.97268
## 5 0.06849315 0.1232877 0.19178082 0.32876712 0.27397260 21.42466
23.32877
## 6 0.04347826 0.0000000 0.08695652 0.26086957 0.60869565 21.43478
23.95652
## 7 0.00000000 0.2250000 0.12500000 0.17500000 0.45000000 20.95000
22.07500
## SocSci1 NatSci1 Vocab1 English2 Math2 SocSci2 NatSci2 Vocab2
## 1 18.90909 17.69318 18.82955 18.53409 18.72727 19.37500 18.95455
```

```
19.52273
## 2 19.45178 19.15228 19.82234 18.89340 20.09137 19.64467 19.00508
19.87310
## 3 20.65854 19.71951 20.80488 20.37805 21.46341 21.24390 20.45732
21.14634
## 4 21.22951 21.14754 21.84699 19.89071 22.20765 21.40984 20.95628
21.98907
## 5 22.27397 21.45205 22.89041 21.93151 24.63014 22.65753 22.20548
23.16438
## 6 22.26087 21.34783 22.78261 21.78261 25.43478 22.91304 22.69565
24.04348
## 7 21.15000 19.42500 22.30000 21.20000 23.20000 21.82500 21.20000
23.15000
##
```

Because the faminc variable has 7 categories, we can have a maximum of 6 linear separators. In the following section, the coefficient of each variable in these 6 linear separators is written.

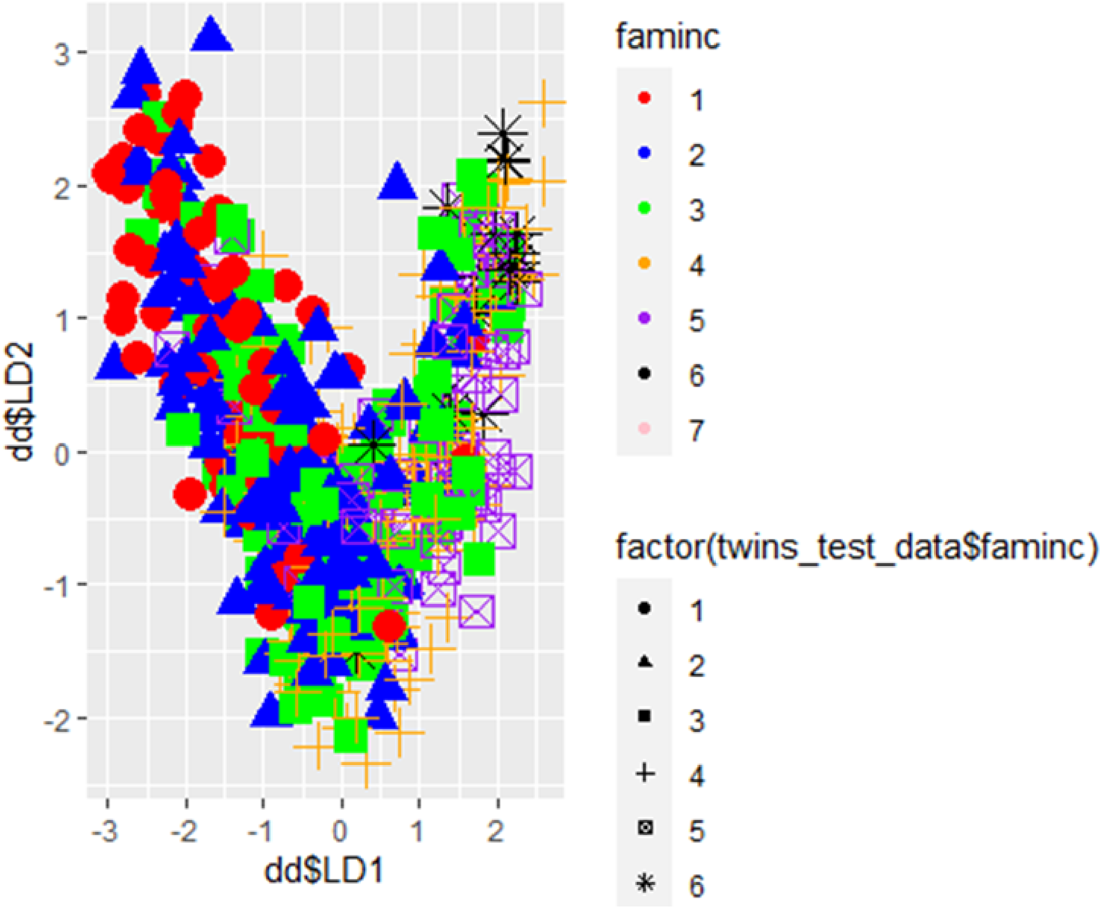
```
## Coefficients of linear discriminants:
##                    LD1          LD2          LD3          LD4
LD5
## sex2       0.308763942  0.270739979  0.932086718  0.636565700
-0.271506532
## zygosity2  0.084617815 -0.244360412  0.091265852  0.135073287
0.087273106
## moed2 0.329220531 -0.989817598 0.929439131 0.869818252 0.004095897
## moed3 0.979174738 -1.621166795 1.385220538 1.096184264 0.932482306
## courage4 1.223392416 -1.371367427 1.120585279 1.280421468
-0.192291112
## courage5 1.242237465 -1.516829212 0.013379448 0.746131264
0.325349385
## courage6 1.328511200 -1.064670307 2.732362706 -0.008365500
1.584452706
## faed2      0.617078170 -0.847261444 -0.460812733 -0.877403169
-0.742098164
## faed3 0.853141884 -1.229917511 0.768428484 -0.876249249
-1.297609404
## faed4      1.351181195 -1.566261040 -0.027709871 -0.740448066
0.088689884
## faed5 2.270246655 -0.869473259 -1.127025017 -0.909440820
-0.588838677
## faed6 2.630979062 0.687629651 0.665606158 -1.227420361 -0.137365490
## English1  -0.016430134  0.063869795  0.013406177 -0.091496538
-0.077610167
## Math1 0.013569668 -0.007557136 0.045549456 0.041810127 -0.037169012
## SocSci1 -0.007069388 0.026740402 -0.025758064 0.065720916
0.070500484
## NatSci1   -0.002433278 -0.067004216 -0.014135940 -0.135200084
-0.015172601
## Vocab1 0.028845737 -0.098696426 -0.022497336 0.022197024
-0.079069426
## English2  -0.015487076 -0.047497612 -0.053903904  0.225899989
-0.006639196
## Math2 0.036623196 0.013400976 -0.008565274 0.002863247 -0.094409062
## SocSci2 -0.016786790 -0.039736789 -0.013430759 0.086145752
0.079750424
## NatSci2    0.009947971  0.032527918 -0.025811665  0.014576426
0.068988378
## Vocab2     0.026704574  0.109114101  0.053839475 -0.136496421
0.064512277
##LD6
## sex2       0.169749181
## zygosity2  0.001196700
## moed2 -1.776021488
## moed3 -1.009728030
## courage4 0.362529671
## courage5 -1.090419307
## courage6 0.279053552
## faed2     -0.607313398
## faed3 -0.180850314
## faed4     -0.268560381
## faed5 0.046273185
## faed6 -1.480443063
## English1   0.014757194
## Math1 -0.038818577
## SocSci1 -0.027108351
## NatSci1   -0.046871353
## Vocab1 0.104591485
## English2  -0.053162443
## Math2 -0.031471173
## SocSci2 -0.003643055
## NatSci2    0.058746749
## Vocab2     0.033374159
##
```

It seems that the first linear separator is the best linear separator.

```
## Proportion of trace:
##    LD1    LD2    LD3    LD4    LD5    LD6
## 0.7003 0.1337 0.0733 0.0509 0.0287 0.0131
```

```
library(ggplot2)
pp = predict(m1)
dd = data.frame(LD1 = pp$x[,1], LD2 = pp$x[,2])
ggplot(data = dd, aes(x = dd$LD1, and = dd$LD2, col = faminc)) +
  geom_point(aes(shape = factor(twins_test_data$faminc), size = 1),
data = dd) +
  scale_color_manual(values = c("red","blue","green",
"orange","purple",
                                "black","pink"))
```

In this section, we applied the predict command to the data and linear separators, and then drew the first and second linear separator in the diagram below. Each of the following shapes represents one of the categories of the faminc variable.



Data source:

- http://psych.colorado.edu/~carey/Courses/PSYC7291/DataSets/Documentation/NMTwinsDataDoc.txt
- http://psych.colorado.edu/~carey/Courses/PSYC7291/ClassDataSets.htm