Problemset 2

Bargetto Cristina 885847, Iavarone Marika 886338, Scanu Anna 1012903

Exercise 1

Consider the pulp_paper data set which consists of n = 62 observations on 8 measurements divided into 4 paper properties and 4 pulp fiber characteristics.

Paper properties:

- BL: Breaking length (length of the paper for which it would break due to its own weight)
- EM: Elastic modulus (flexibility of the paper)
- SF: Stress at failure (percentage of the length of paper you can stretch until it breaks)
- BS: Burst strength (stress that paper can bear before bursting)

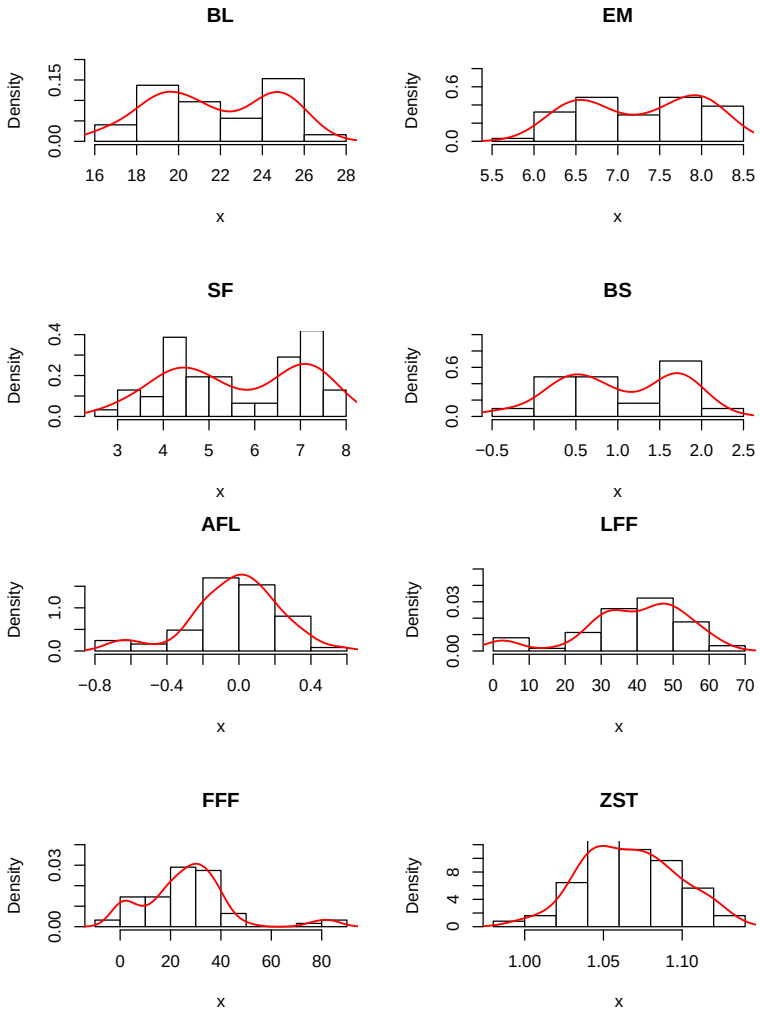
Pulp fiber characteristics:

- AFL: Arithmetic fiber length (arithmetic mean of the length of fibers measured)
- LFF: Long fiber fraction (proportion of long fiber)
- FFF: Fine fiber fraction (proportion of fine fiber)
- ZST: Zero span tensile (tensile strength of fiber)

Table 1: Head of pulp_paper dataset

BL	EM	SF	BS	AFL	LFF	FFF	ZST
21.31	7.039	5.326	0.932	-0.03	35.24	36.99	1.057
21.21	6.979	5.237	0.871	0.015	35.71	36.85	1.064
20.71	6.779	5.06	0.742	0.025	39.22	30.59	1.053
19.54	6.601	4.479	0.513	0.03	39.76	21.07	1.05
20.45	6.795	4.912	0.577	-0.07	32.99	36.57	1.049
20.84	6.919	5.108	0.784	-0.05	31.14	38.12	1.052

In order to compute the Factor Analysis we need to have normally distributed variables.



As we can see from the plot the variables are not actually normally distributed, anyaway we will perform Factor Analysis.

Consider the correlation matrix R

Table 2: Correlation matrix

	BL	EM	SF	BS	AFL	LFF	FFF	ZST
BL	1	0.914	0.984	0.988	0.648	0.735	-0.542	0.822
EM	0.914	1	0.942	0.875	0.537	0.609	-0.556	0.85
SF	0.984	0.942	1	0.975	0.681	0.764	-0.575	0.865
BS	0.988	0.875	0.975	1	0.706	0.796	-0.564	0.813
AFL	0.648	0.537	0.681	0.706	1	0.906	-0.733	0.784
LFF	0.735	0.609	0.764	0.796	0.906	1	-0.711	0.793
FFF	-0.542	-0.556	-0.575	-0.564	-0.733	-0.711	1	-0.785
ZST	0.822	0.85	0.865	0.813	0.784	0.793	-0.785	1

We can see that the first four variables (BL, EM, SF, BS) are highly correlated since they describe measurements about resistance of paper before breaking. So we expect they will load on the same factor in the Factor Analysis (treated later).

Considering the other four characteristics regarding the pulp fiber, we have that AFL and LFF are highly

correlated since both give us information about the length of fibers, in fact we expect that if the proportion of long fiber was high, the arithmetic mean of the length of fibers measured would be high too. On the other hand ZST seems to fit well with all the first six variables since it expresses pulp fiber characteristic measured on a slice of paper.

Note that the FFF variable has negative correlation coefficients quite sizable relative to all the other variables. This is reasonable since it is the only variable which expresses a bad characteristic of the paper because the finer fibers are, the less resistant paper will be.

Now, let us compute the maximum likelihood factor analysis with m=2.

```
pulp_paper.fa_ml<-factanal(covmat=R, factors = 2, rotation = "none")
```

```
##
## Call:
## factanal(factors = 2, covmat = R, rotation = "none")
##
## Uniquenesses:
##    BL    EM    SF    BS   AFL   LFF   FFF   ZST
## 0.005 0.152 0.023 0.016 0.074 0.090 0.416 0.206
##
## Loadings:
##     Factor1 Factor2
## BL   0.995
## EM   0.911  -0.131
## SF   0.988
## BS   0.992
## AFL  0.695   0.665
## LFF  0.778   0.553
## FFF -0.578  -0.500
## ZST  0.845   0.281
##
##                Factor1 Factor2
## SS loadings      5.918   1.099
## Proportion Var   0.740   0.137
## Cumulative Var   0.740   0.877
##
## The degrees of freedom for the model is 13 and the fit was 2.4157
```

Firstly, we analyze the uniqueness of our variables:

Table 3: Specific variances

BL	EM	SF	BS	AFL	LFF	FFF	ZST
0.005	0.1522	0.02349	0.01618	0.0741	0.0899	0.4163	0.2061

the value for FFF is high with respect to the other variables.

The uniqueness $\widehat{\psi}_i$ corresponds to the proportion of variability (specific variance), which can not be explained by a linear combination of the factors. A high uniqueness for a FFF indicates that the factors do not account well for its variance.

On the other hand the first six variables having low values, are well explained by the model with at least 2 factors since

- BL, EM, SF and BS
- AFL and LFF

are two groups of highly correlated variables, each one expressed by a different factor.

Since

$$var(X_j) = h_j^2 + \psi_j = communality + specific\ variance$$

and the variance for the standardized data is equal to one, the communalities are computed by subtracting the specific variances from the variance as expressed below:

$$\widehat{h_i^2} = 1 - \hat{\psi}_i$$

Hence an high value of uniqueness implies a low value of communality.

So the communalities are:

Table 4: Communalities

BL	EM	SF	BS	AFL	LFF	FFF	ZST
0.9954	0.8478	0.9765	0.9838	0.9259	0.9101	0.5836	0.7939

The communality for FFF variable is in fact 0.583, indicating that about 58% of the variation in FFF is explained by the factor model. This suggest that FFF has little in common with the other variables.

Also ZST has a communality value that does not reach the 80%, since it has not a very high correlation coefficient with any variable. So, this probably means that adding a new factor, ZST might be explained by Factor 3.

Let us compute the residual matrix

Table 5: Residual matrix

	BL	EM	SF	BS	AFL	LFF	FFF	ZST
BL	0	-0.002	0	0.002	0	-0.002	0	-0.001
EM	-0.002	0	0.041	-0.026	-0.009	-0.027	-0.095	0.116
SF	0	0.041	0	-0.005	-0.001	0	-0.007	0.032
BS	0.002	-0.026	-0.005	0	0.001	0.011	0.022	-0.032
AFL	0	-0.009	-0.001	0.001	0	-0.003	0.001	0.009
LFF	-0.002	-0.027	0	0.011	-0.003	0	0.015	-0.02
FFF	0	-0.095	-0.007	0.022	0.001	0.015	0	-0.155
ZST	-0.001	0.116	0.032	-0.032	0.009	-0.02	-0.155	0

Most of the entries are close to zero, this means that our factor model is appropriate. In fact, the sum of the squared entries of the residual matrix is fairly small.

```
sum(Residual^2)
```

```
## [1] 0.1064641
```

Computing the cumulative proportion of variance, we have that the percentage of the total variation explained in our model by two factors is 87.7%, that is appreciable.

```
sum(hi.sq[1:p])/p
```

```
## [1] 0.8771294
```

Repeat the same maximum likelihood factor analysis with m=3.

```
pulp_paper.fa2_ml<-factanal(covmat=R, factors = 3, rotation = "none")
```

```
##
## Call:
## factanal(factors = 3, covmat = R, rotation = "none")
##
## Uniquenesses:
##    BL    EM    SF    BS   AFL   LFF   FFF   ZST
## 0.007 0.023 0.011 0.005 0.087 0.079 0.307 0.068
##
## Loadings:
##     Factor1 Factor2 Factor3
## BL   0.994
## EM   0.919  -0.253   0.263
## SF   0.991
## BS   0.992
## AFL  0.700   0.620   0.198
## LFF  0.783   0.545
## FFF -0.586  -0.397  -0.438
## ZST  0.853   0.164   0.421
##
##                Factor1 Factor2 Factor3
## SS loadings      5.973   0.938   0.502
## Proportion Var   0.747   0.117   0.063
## Cumulative Var   0.747   0.864   0.927
##
## The degrees of freedom for the model is 7 and the fit was 0.4171
```

The uniquenesses of our variables are:

Table 6: Specific variances

BL	EM	SF	BS	AFL	LFF	FFF	ZST
0.006876	0.02303	0.0106	0.00472	0.08664	0.07912	0.3074	0.0685

As in the previous case, the highest value of the estimated specific variances corresponds to FFF (0.30). Note that the value for ZST is decreased towards 0.06.

Table 7: Communalities

BL	EM	SF	BS	AFL	LFF	FFF	ZST
0.9931	0.977	0.9894	0.9953	0.9134	0.9209	0.6926	0.9315

The value of FFF communality is grater than before (0.69 compared to 0.58), however it is still a low value with respect to the others.

On the other hand, the value of the communality of ZST is remarkably increased (0.93 instead of 0.79). So for what concerns this variable, the model with m=3 would be better.

Let us compute the residual matrix

Table 8: Residual matrix

	BL	EM	SF	BS	AFL	LFF	FFF	ZST
BL	0	-0.002	0	0	0	-0.005	-0.005	0.005
EM	-0.002	0	0.002	0	-0.001	0.001	-0.003	-0.003
SF	0	0.002	0	0	0.001	0.006	0.019	-0.002
BS	0	0	0	0	0	0.002	-0.003	-0.002
AFL	0	-0.001	0.001	0	0	0	0.009	0.002
LFF	-0.005	0.001	0.006	0.002	0	0	0.008	-0.007
FFF	-0.005	-0.003	0.019	-0.003	0.009	0.008	0	-0.035
ZST	0.005	-0.003	-0.002	-0.002	0.002	-0.007	-0.035	0

Also in this case, the model is appropriate since all entries are close to zero and the sum of squares is smaller than before, as we expected.

```
sum(Residual2^2)
```

```
## [1] 0.003930445
```

The cumulative proportion of variance is

```
sum(hi2.sq[1:p])/p
```

```
## [1] 0.9266388
```

This means that 92% of the total sample variation is explained by our model with m = 3. This is higher than before, as we expected with one more factor.

```
pulp_paper.fa2_ml$loadings
```

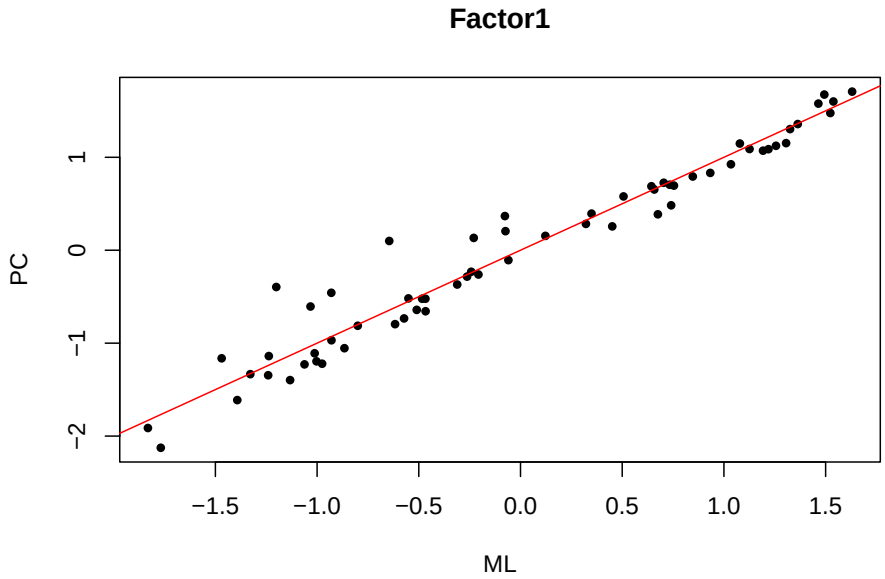
```
##
## Loadings:
##     Factor1 Factor2 Factor3
## BL   0.994
## EM   0.919  -0.253   0.263
## SF   0.991
## BS   0.992
## AFL  0.700   0.620   0.198
## LFF  0.783   0.545
## FFF -0.586  -0.397  -0.438
## ZST  0.853   0.164   0.421
##
##                Factor1 Factor2 Factor3
## SS loadings      5.973   0.938   0.502
## Proportion Var   0.747   0.117   0.063
## Cumulative Var   0.747   0.864   0.927
```

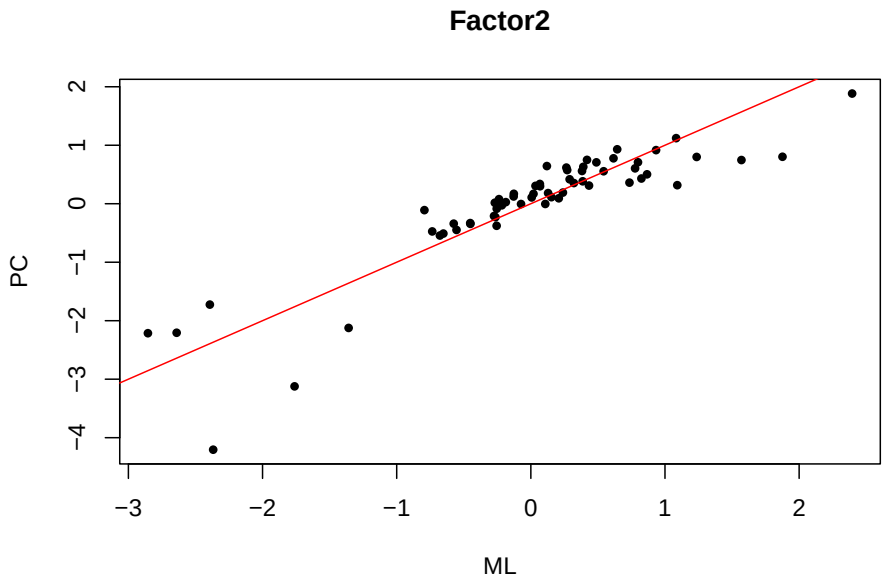
However, the proportion variance of Factor 3 is not very significant (0.063), hence it could be not needed. In fact note that the loadings relative to Factor3 are not significantly high for any variable.

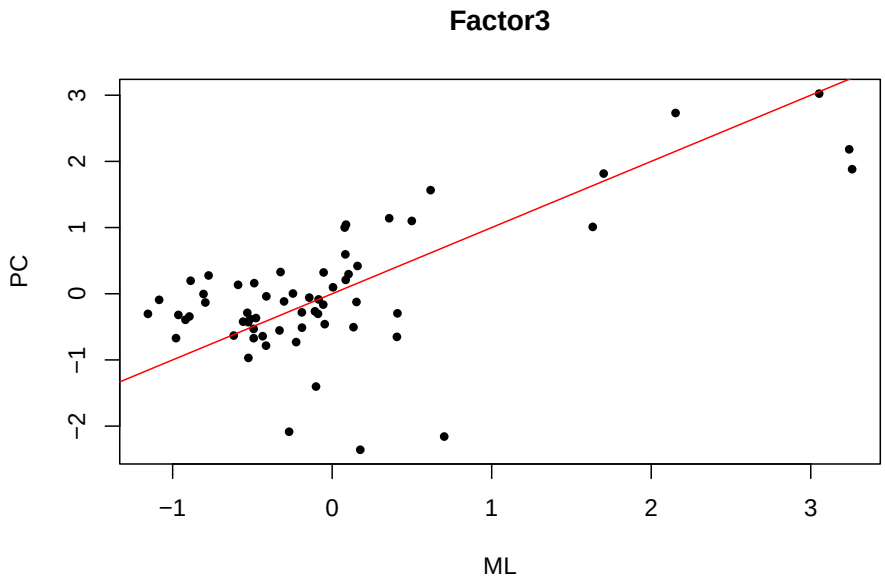
Moreover, comparing the factor scores generated from two different extraction methods (Principal component method with varimax rotation and maximum likelihood method with regression method) and observing the correlations between them, we obtain that the third factor is not necessary for our purpose.

In fact, the scatter plots shows that scores for Factor3 do not follow the red line, while the first two do so and the correlation coefficient is lower.

```
library(psych)
faML<-factanal(x=pulp_paper, factors=3, scores = "regression")
faPC<-principal(r=pulp_paper, nfactors=3, rotate="varimax")
```








```
cor(faML$scores[,1],faPC$scores[,1])
```

```
## [1] 0.9748401
```

```
cor(faML$scores[,2],faPC$scores[,2])
```

```
## [1] 0.8944281
```

```
cor(faML$scores[,3],faPC$scores[,3])
```

```
## [1] 0.6690985
```

So, it is reasonable to choose m = 2 factors because the third factor does not count very much in our analysis and also because the percentage of the total variation explained in our model by two factors is greater than 80%, that is enough. Moreover with m=2 we have a reduction of dimensionality and a good fitting for the data.

Now, let us analyze the possible interpretation of our two factors looking at the estimated factor loading matrix and the plot of the loadings.

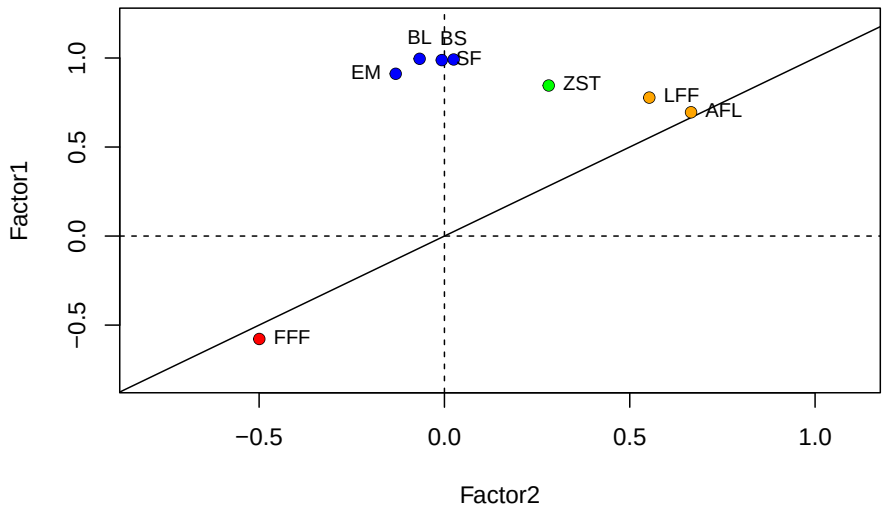


Table 9: Loadings

	Factor1	Factor2
BL	0.995	-0.067
EM	0.911	-0.131
SF	0.988	-0.007
BS	0.992	0.025
AFL	0.695	0.665
LFF	0.778	0.553
FFF	-0.578	-0.5
ZST	0.845	0.281

Loadings close to -1 or 1 indicate that the factor strongly influences the variable. All the variables load on the first factor, in particular the first four paper properties and ZST have loadings larger in size. So it follows that Factor1 describes the quality of the paper in terms of strength.

While Factor2 concerns the length of pulp fibers, since only AFL, LFF and FFF variables load on this factor quite well (considering the absolute value of the loadings).

FFF has similar large negative loadings on both factors and so it is not well explained by any particular factor.

To give a better interpretation to the factors, we try to compute the factor analysis with varimax rotation method.

```
pulp_paper.fa3_ml<-factanal(covmat=R, factors = 2, rotation = "varimax")
```

```
##
## Call:
## factanal(factors = 2, covmat = R, rotation = "varimax")
##
## Uniquenesses:
##    BL    EM    SF    BS   AFL   LFF   FFF   ZST
## 0.005 0.152 0.023 0.016 0.074 0.090 0.416 0.206
##
## Loadings:
##     Factor1 Factor2
## BL    0.921   0.383
## EM    0.875   0.288
## SF    0.888   0.433
## BS    0.877   0.463
## AFL   0.327   0.905
## LFF   0.451   0.841
## FFF -0.296  -0.704
## ZST   0.632   0.628
##
##                Factor1 Factor2
## SS loadings      3.969   3.048
## Proportion Var   0.496   0.381
## Cumulative Var   0.496   0.877
##
## The degrees of freedom for the model is 13 and the fit was 2.4157
```

In this case, the loadings change significantly and give more information about which variable loads most on which factor, as we can see in the plot below.

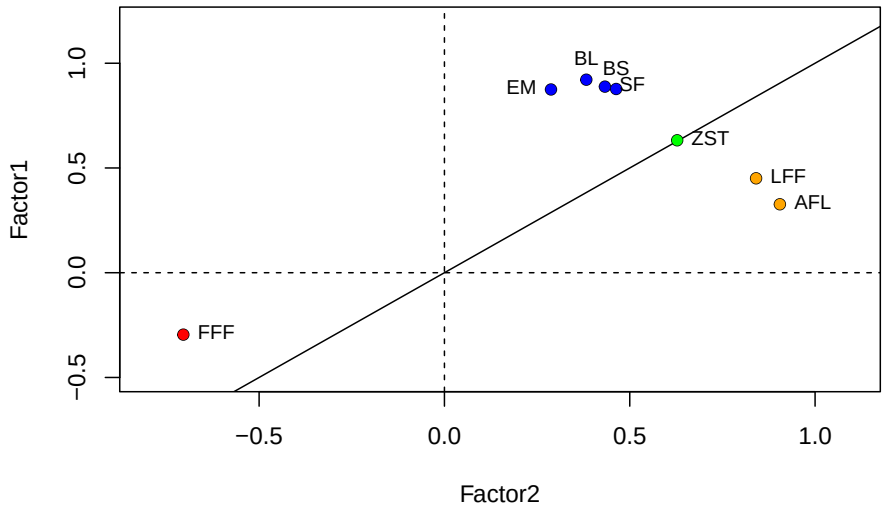


Table 10: Loadings

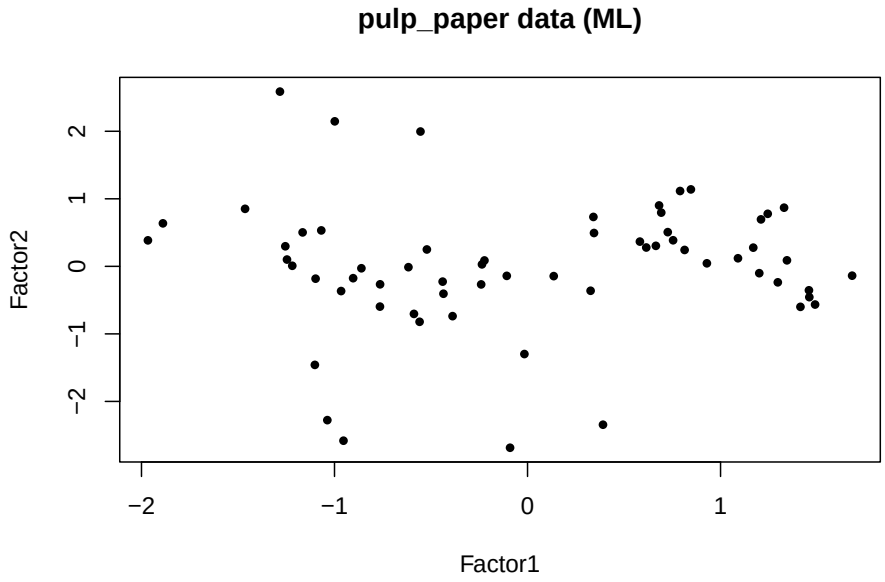
	Factor1	Factor2
BL	0.921	0.383
EM	0.875	0.288
SF	0.888	0.433
BS	0.877	0.463
AFL	0.327	0.905
LFF	0.451	0.841
FFF	-0.296	-0.704
ZST	0.632	0.628

Indeed, the blue points, which represent the paper properties, are above the diagonal and in fact they load on the Factor1 (y-axis). While LFF and AFL are described by Factor2 (x-axis).

Differently from above, ZST loads equally on both factors because it provides information on the fiber strength measured on paper.

Considering the absolute value of the factor loadings of FFF, we can notice that the largest one is the one related to Factor2 (described as before).

Now, let us make the scatterplot of the factor scores obtained by the regression method.



Plot of factors scores should produce elliptical shapes when the assumption of multivariate normality is satisfied.

We can say that the data do not follow a normal distribution since the points are not distributed around the value of 0.

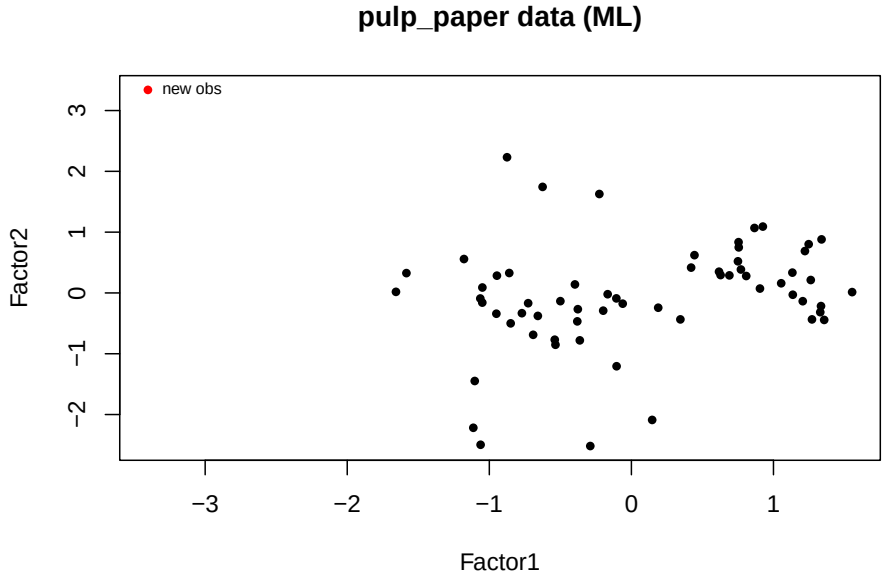
Computing the correlation between them, we see that it is close to zero.

```
cor(faML$scores[,1],faML$scores[,2])
```

```
## [1] 0.03223625
```

This does not surprise us, because by construction of our model, we assume that the correlation between the factors has to be zero. Also variables that are highly correlated load on the same factor and hence it is reasonable that the scores of each factor are uncorrelated.

Suppose now we have a new observation (15.5, 5.5, 2, −0.55, 0.6, 65, −5, 1.2) that we add to our data set. Let us make the scatterplot of the factor scores computed on our new dataset, in which the point colored in red represents the factor scores of our new observation.

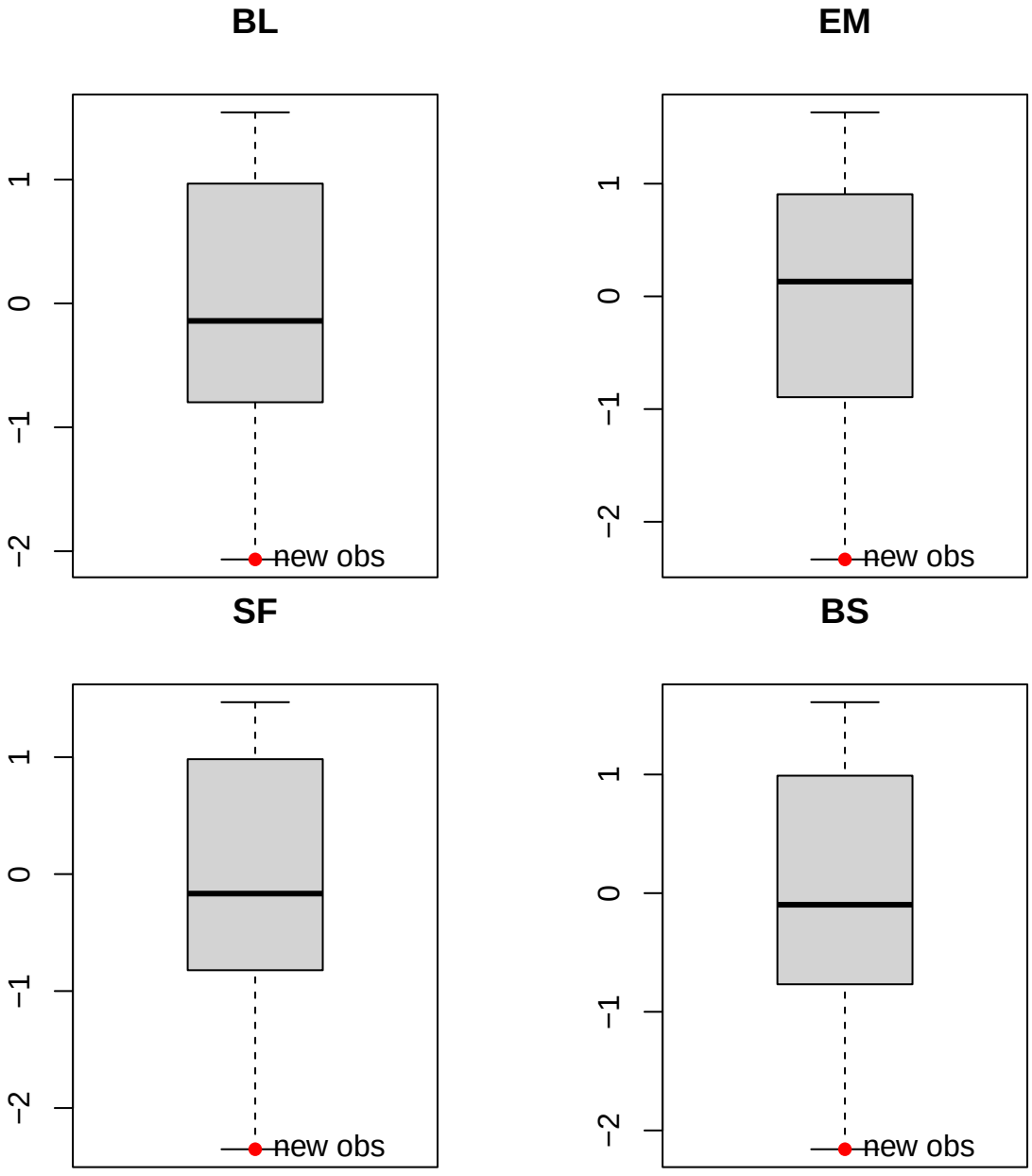


We observe that the last observation has the highest score of the Factor2 and the lowest score of the Factor1.

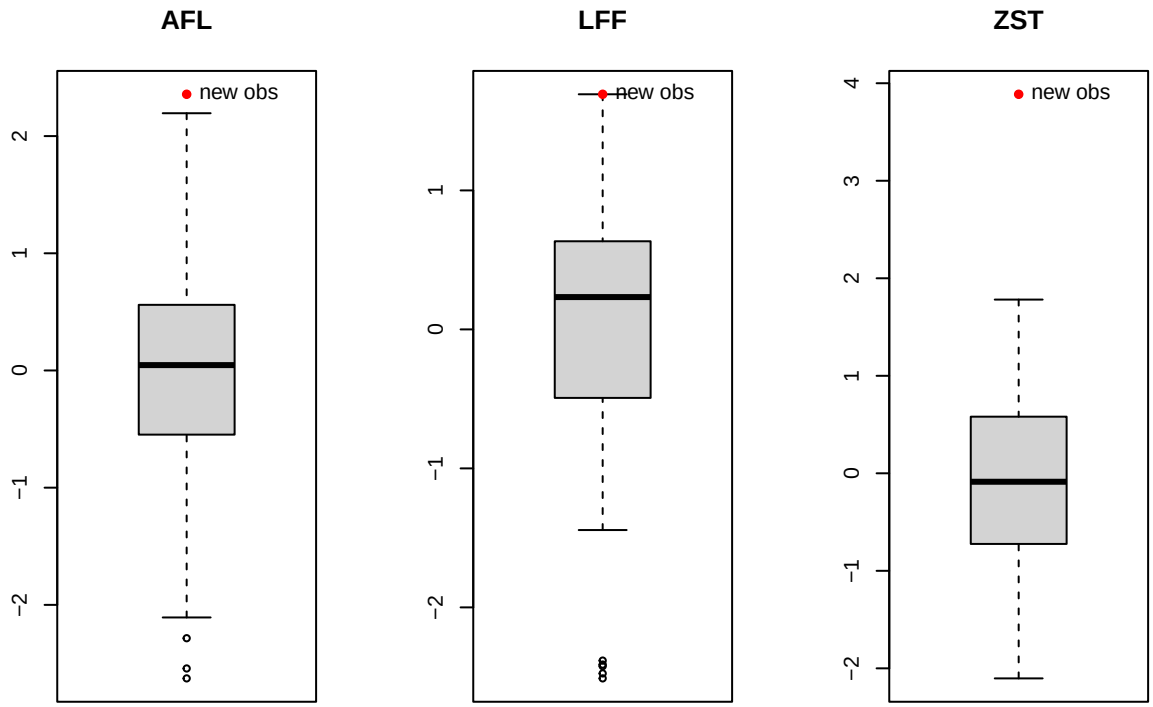
```
##   Factor1   Factor2
## -3.402995  3.339281
```

We suppose that its collocation on the plot of the factor scores is determined by its values of the variables. It has:

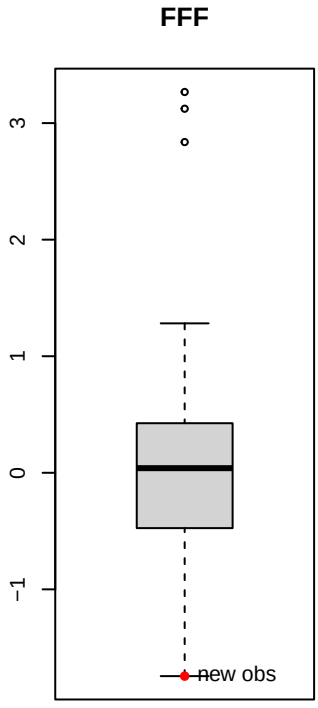
- the minimum value for BL, EM, SF and BS



- the highest values for the variables AFL, LFF and ZST



- the minimum value for the variable FFF



In addition this are the loadings of the FA with m = 2 computed on the new dataset.

Table 11: Loadings

	Factor1	Factor2
BL	0.947	0.315
EM	0.904	0.208
SF	0.93	0.332
BS	0.919	0.371
AFL	0.231	0.942
LFF	0.383	0.872
FFF	-0.228	-0.744
ZST	0.327	0.762

Table 12: Normalized values for observation 63

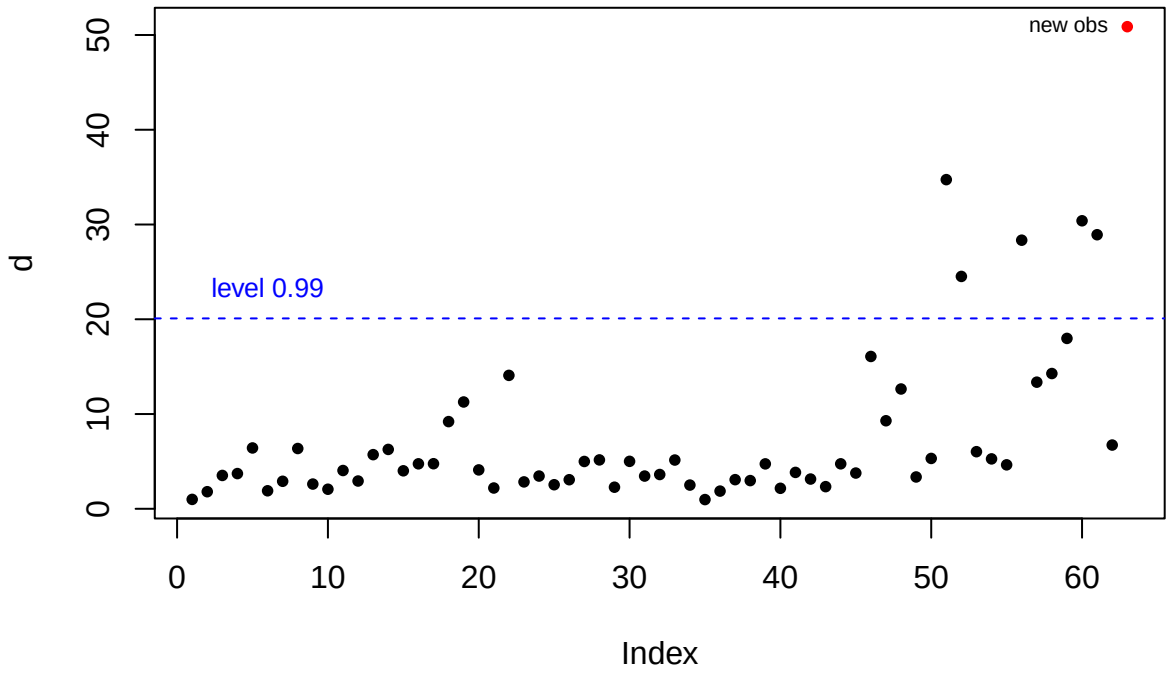
	BL	EM	SF	BS	AFL	LFF	FFF	ZST
63	-2.066	-2.334	-2.352	-2.159	2.357	1.692	-1.744	3.887

Concerning our new observation, looking at the values it takes on the variables and the loadings table, we can say that:

- the variables which load most with positive coefficient on Factor1 are the first four (BL, EM, SF and BS) and it takes the minimum value on them, which is negative. So its score relative to this factor should be negative and minimum;
- the variables which load most on Factor2 are the last four AFL, LFF, ZST (with positive loadings) and FFF (with negative loading) and the 63th observation takes respectively on them the maximum positive and minimum negative value. So its score relative to Factor2 should be positive and maximum.

So we can deduce that the piece of paper, corresponding to the 63th observation, has a low breaking strength but good pulp fiber properties.

Moreover, considering individually the variables, we see from the boxplots that this new observation is an univariate outlier. So we want to compute the Mahalanobis distance in order to verify that it is a multivariate outlier.



From the plot, we note that our new observation has the highest value of Mahalanobis distance and it is above the chi-squared quantile of order 0.99 with some other observations.

Exercise 2

Consider the glass dataset, containing n=214 observations which represent single glass fragments. For each of them refractive index (*RI*) and weight percent of oxides of *Na*, *Mg*, *Al*, *Si*, *K*, *Ca*, *Ba* and *Fe* are measured. The fragments are classified as six types (variable *type*) :

- WinF: window float glass
- WinNF: window non float glass
- Veh: vehicle window glass
- Con: containers
- Tabl: tableware
- Head: vehicle headlamps

Table 13: Head of glass dataset

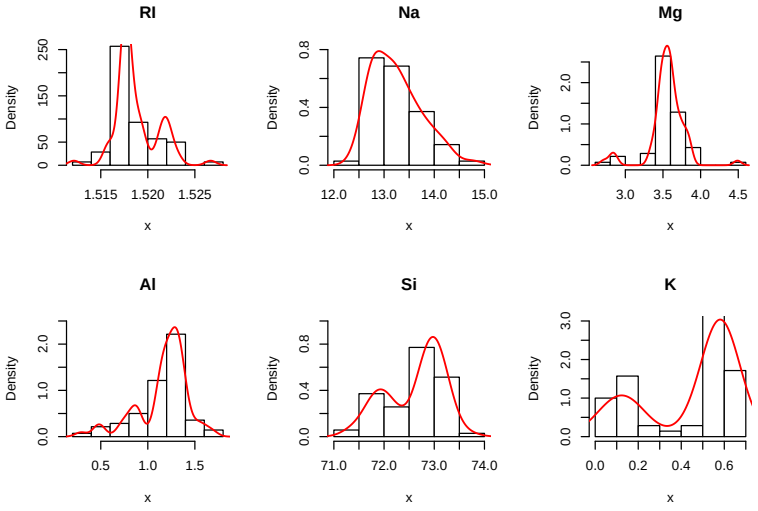
RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	type
1.521	13.64	4.49	1.1	71.78	0.06	8.75	0	0	WinF
1.518	13.89	3.6	1.36	72.73	0.48	7.83	0	0	WinF
1.516	13.53	3.55	1.54	72.99	0.39	7.78	0	0	WinF
1.518	13.21	3.69	1.29	72.61	0.57	8.22	0	0	WinF
1.517	13.27	3.62	1.24	73.08	0.55	8.07	0	0	WinF
1.516	12.79	3.61	1.62	72.97	0.64	8.07	0	0.26	WinF

We notice that we have a different amount of observations for each type, for example we have only 9 observations for Tabl while 76 for WinNF.

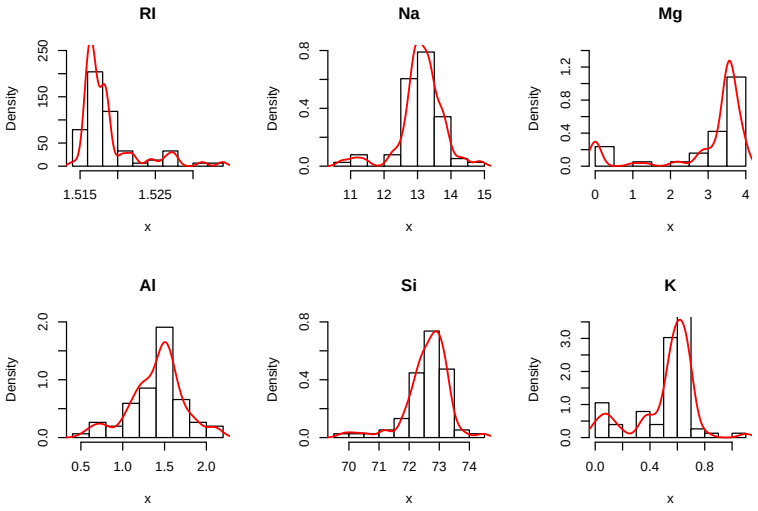
WinF	WinNF	Veh	Con	Tabl	Head
70	76	17	13	9	29

Before performing Linear Discriminant Analysis (LDA), we check the assumption of normality of our predictors variables for each class:

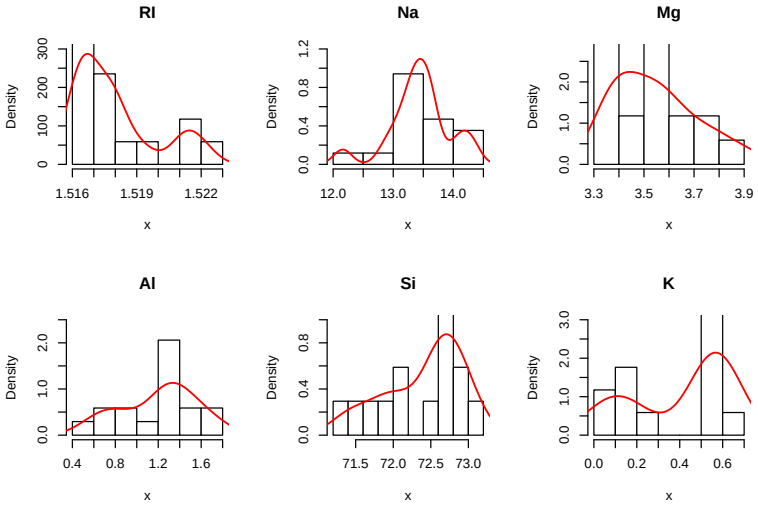
- WinF



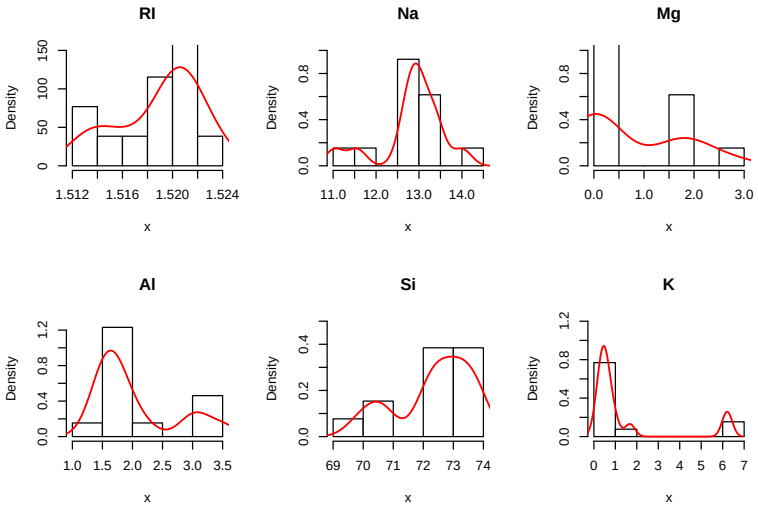
- WinNF



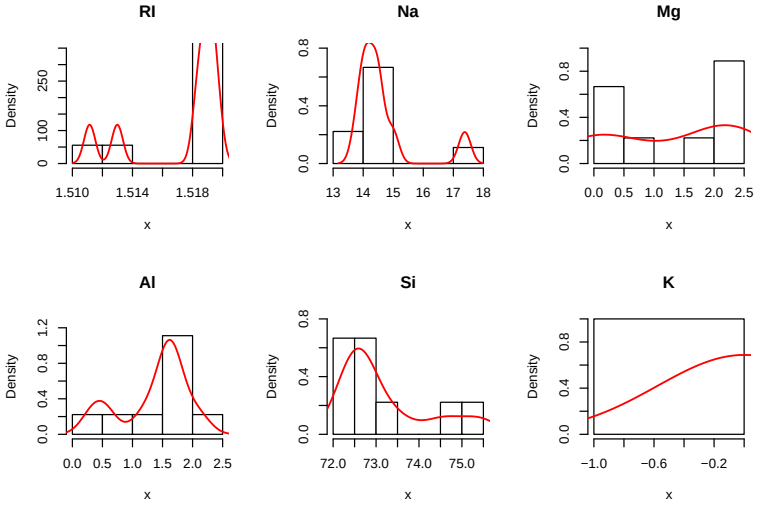
- Veh



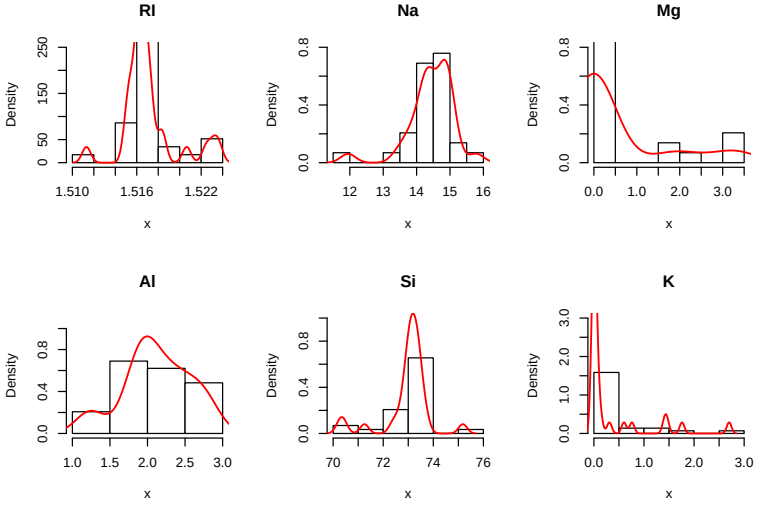
- Con



- Tabl

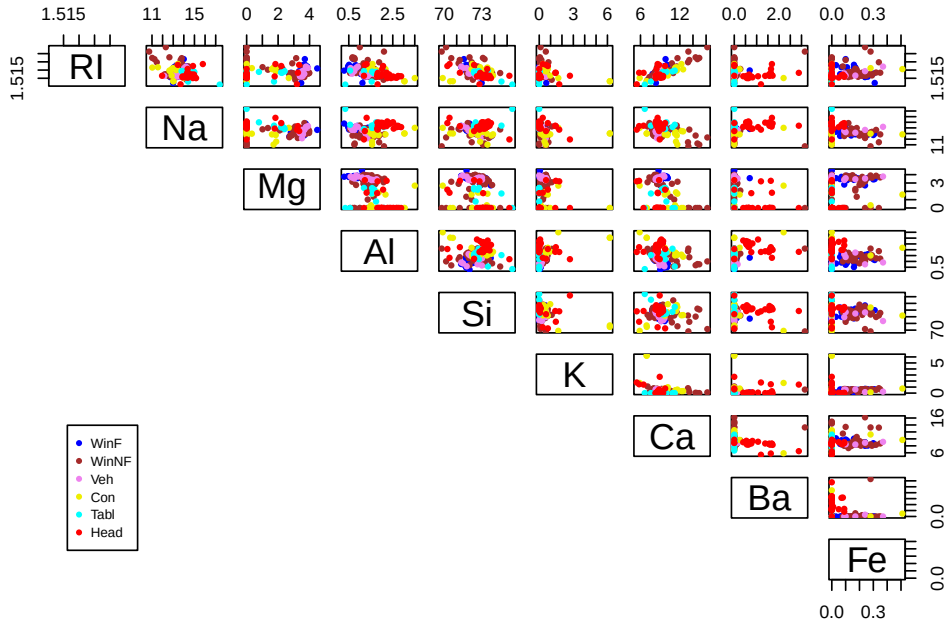


- Head



From the plots we cannot say that all of them are normally distributed since some don't have a bell shape (e.g. K and Mg) while Na, Si and Al could be approximately considered normally distributed. Hence performing LDA we don't expect high accuracy in the results.

We would like to find the variables which most separate classes.



Looking at the scatterplots we can not really separate classes through variables, but there are some variables such as Ca, Si, Na and Al, that seems to do so.

Let us perform the linear discriminant analysis to predict the glass type.

```
## Call:
## lda(type ~ ., data = glass)
##
## Prior probabilities of groups:
##       WinF      WinNF        Veh        Con       Tabl       Head
## 0.32710280 0.35514019 0.07943925 0.06074766 0.04205607 0.13551402
##
## Group means:
##              RI       Na        Mg       Al       Si         K        Ca
## WinF  1.518718 13.24229 3.5524286 1.163857 72.61914 0.4474286  8.797286
## WinNF 1.518619 13.11171 3.0021053 1.408158 72.59803 0.5210526  9.073684
## Veh   1.517964 13.43706 3.5435294 1.201176 72.40471 0.4064706  8.782941
## Con   1.518928 12.82769 0.7738462 2.033846 72.36615 1.4700000 10.123846
## Tabl  1.517456 14.64667 1.3055556 1.366667 73.20667 0.0000000  9.356667
## Head  1.517116 14.44207 0.5382759 2.122759 72.96586 0.3251724  8.491379
##                Ba         Fe
## WinF  0.012714286 0.05700000
## WinNF 0.050263158 0.07973684
## Veh   0.008823529 0.05705882
## Con   0.187692308 0.06076923
## Tabl  0.000000000 0.00000000
## Head  1.040000000 0.01344828
##
## Coefficients of linear discriminants:
##            LD1         LD2          LD3           LD4           LD5
```

```
## RI 311.6912516 29.3910394 356.0188308 246.85720802 -804.6553938
## Na   2.3812158  3.1650800   0.4596785   6.92435141    2.3987509
## Mg   0.7403818  2.9858720   1.5728838   6.84983896    2.8002951
## Al   3.3377416  1.7247396   2.2024668   6.41923638    0.9371345
## Si   2.4516520  3.0063507   1.7026191   7.54220302    0.9562989
## K    1.5714954  1.8620159   1.2861127   8.07611300    2.8209927
## Ca   1.0063101  2.3729126   0.6475200   6.69663574    3.7110859
## Ba   2.3140953  3.4431987   2.5964981   6.43849270    4.4077058
## Fe  -0.5114573  0.2166388   1.2026071  -0.04474935   -1.3029207
##
## Proportion of trace:
##    LD1    LD2    LD3    LD4    LD5
## 0.8145 0.1169 0.0413 0.0163 0.0111
```

When the variables are measured on different scales, given a the discriminant coordinate vector, the standardized one

$$a^* = [diag(\hat{\Sigma})]^{1/2} a$$

provides better information than a about the relative contribution of each variable to the discriminant variable. So we use it to find the coefficients of Linear Discriminat variables.

```
# Compute the pooled covariance matrix
S1<-var(glass[glass$type=="WinF",1:9]); S2<-var(glass[glass$type=="WinNF",1:9])
S3<-var(glass[glass$type=="Veh",1:9]); S4<-var(glass[glass$type=="Con",1:9])
S5<-var(glass[glass$type=="Tabl",1:9]); S6<-var(glass[glass$type=="Head",1:9])

n1<-70; n2<-76; n3<-17; n4<-13; n5<-9; n6<-29
n<-n1+n2+n3+n4+n5+n6
S<-((n1-1)*S1+(n2-1)*S2+(n3-1)*S3+(n4-1)*S4+(n5-1)*S5+(n6-1)*S6)/(n-6)

# Scale the coefficient vector
scaling<-sqrt(diag(diag(S)))%*%as.matrix((glass.lda$scaling[,1:2]))
rownames(scaling) <- c("RI", "Na", "Mg", "Al", "Si", "K", "Ca", "Ba", "Fe")
set.caption("Coefficients of linear discriminants")
pander(scaling)
```

Table 15: Coefficients of linear discriminants

	LD1	LD2
RI	0.9399	0.08863
Na	1.515	2.014
Mg	0.6734	2.716
Al	1.237	0.6391
Si	1.86	2.281
K	0.9428	1.117
Ca	1.4	3.301
Ba	0.8366	1.245
Fe	-0.04886	0.0207

Looking at the absolute values of the coefficients, we have that LD1 is mostly described by Si, Na, Ca and Al, while LD2 by Ca, Mg, Si, Na, Ba and K. So these are supposed to be the most important variables in separating the classes.

Moreover if we consider LD1 and LD2 that are the linear combination of the variables, we have that LD2 is significantly described by a lot of variables hence, for example considering Ca and Mg that respectively assume high values for the classes Con/Tabl and WinF/WinNF/Veh, it is less meaningful than LD1 for separating classes.

Now let's compute the confusion matrix in order to see the missclassified observations in the prediction. In our matrix the rows correspond to what the LDA predicted and the columns correspond to the known true.

Table 16: Confusion matrix

	WinF	WinNF	Veh	Con	Tabl	Head
WinF	52	17	11	0	1	1
WinNF	15	54	6	5	2	2
Veh	3	0	0	0	0	0
Con	0	3	0	7	0	1
Tabl	0	2	0	0	6	0
Head	0	0	0	1	0	25

We can notice that all observations which correspond to Veh type are missclassified, identified 11 in WinF and 6 in WinNF. This make sense since from the scatterplot above we have seen that the pink points (Veh) are overlapping the blue and brown points (corresponding to WinF and WinNF, respectively).

More precisely the total number of missclassifications is

```
n<-dim(glass)[1]
n-sum(diag(conf.mat))
```

```
## [1] 70
```

For what concerns the training error of the full lda classifier, we have

```
1-sum(diag(conf.mat))/n
```

```
## [1] 0.3271028
```

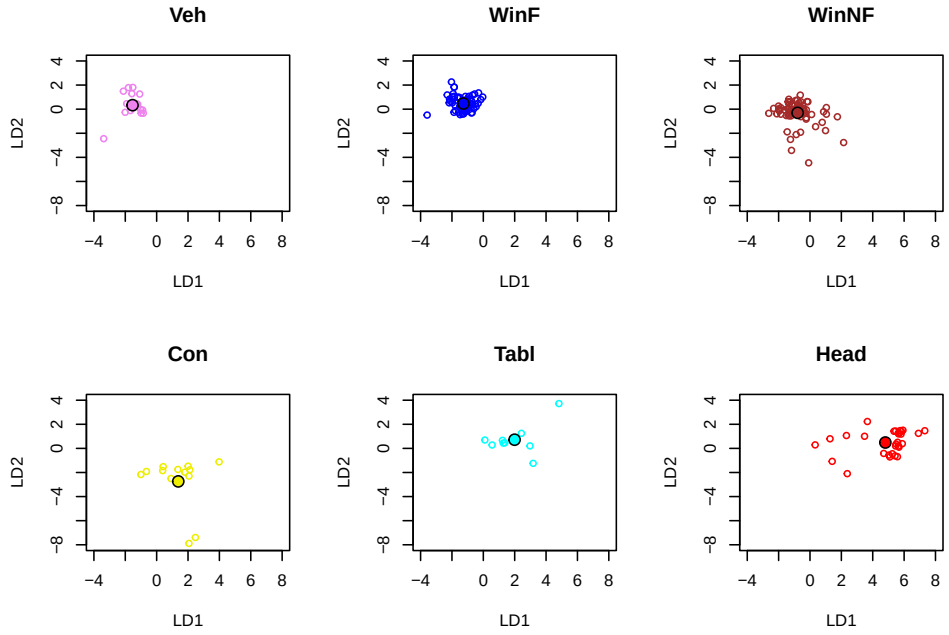
Comparing the training errors obtained for each class, it seems that the type Veh (which has all missclassified observations and hence training error equal to one) is the most heterogeneous class. On the other hand Head seems to be the most homogeneous class since its error is the minimum.

Table 17: Training errors

WinF	WinNF	Veh	Con	Tabl	Head
0.2571	0.2895	1	0.4615	0.3333	0.1379

However plotting the two discriminant variables for observations that were previously classified in each type, we can see that Veh does not seem to be so heterogeneous since the points are not spread out but are concentrated around the centroid.

Hence it is possible that the training error concerning the Veh type is influenced by the fact that the observations have almost the same measurements of the observations in the classes WinF and WinNF (respectively identified with color blue and brown).



Looking at the plots we can identify Con as the most heterogeneous class. In fact its error is the biggest one excluding the Veh type.

Moreover let us compute the prediction with reduced rank classifier with dimension equal to 1 and 2 and compare their training error.

Table 18: Confusion matrix dimen = 1

	WinF	WinNF	Veh	Con	Tabl	Head
WinF	36	26	12	0	0	0
WinNF	34	48	5	5	2	1
Veh	0	0	0	0	0	0
Con	0	2	0	6	3	2
Tabl	0	0	0	1	2	2
Head	0	0	0	1	2	24

Table 19: Confusion matrix dimen = 2

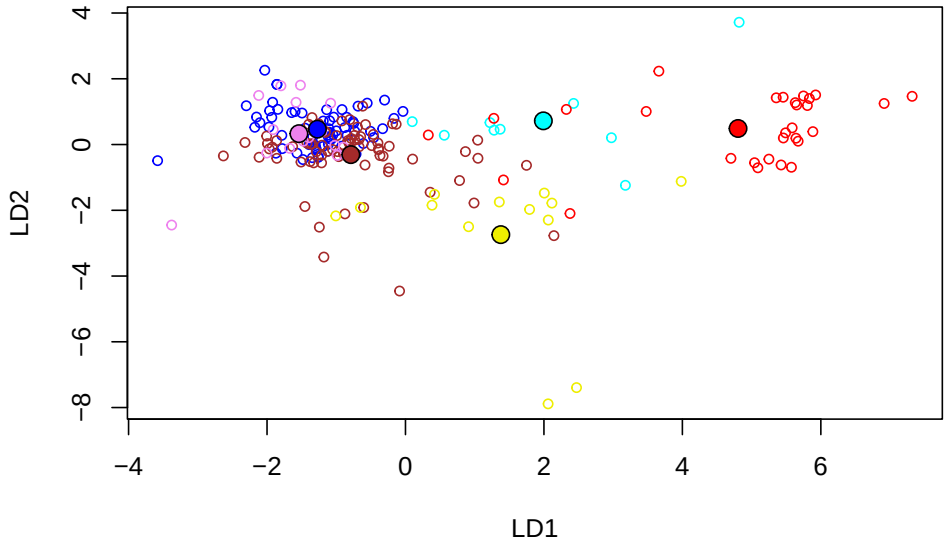
	WinF	WinNF	Veh	Con	Tabl	Head
WinF	51	24	12	0	0	0
WinNF	19	48	5	4	3	2
Veh	0	0	0	0	0	0
Con	0	3	0	8	0	1
Tabl	0	1	0	0	3	2
Head	0	0	0	1	3	24

Table 20: Training errors varying the dimension

	dimen=1	dimen=2
WinF	0.4857	0.2714
WinNF	0.3684	0.3684
Veh	1	1
Con	0.5385	0.3846
Tabl	0.7778	0.6667
Head	0.1724	0.1724

We observe that the training error for WinF (blue), Con (yellow) and Tabl (cyan), considering dimen = 2 instead of dimen = 1, decreases.

In fact plotting the two discriminant directions with heavy circles corresponding to the projected centroids for each class, we can see a better separation of cyan and yellow points due to LD2.



On the other hand, the training error of the class Head does not change with dimen = 2. This is because its separation from the others is determined by LD1. We had already seen that its training error was the lowest one and its homogeneity is confirmed.

Through the plot we see that Con observations (yellow) are spread out as we expected from the training errors.

Let us now implement a 10-fold cross validation using the partition of the observations provided by the variable groupCV to estimate the error rate:

```
groupCV<-scan(file="data/groupCV.txt")
glass2<-cbind(glass,groupCV)
k <- length(unique(glass$type))
errorCV<-c()
for (i in 1:10)
{
  v<-c(which(glass2$groupCV==i))
  # split the dataset in train data and test data
  test_data<-glass2[v,1:10]
  train_data<-glass2[-v,1:10]
  # perform LDA on the training set
```

```
  lda.fitCV<-lda(type~., data=train_data)
  # predict the class for test data
  lda.fitCV.pred<-predict(lda.fitCV, test_data, dimen=k-1)
  # get the confusion  matrix
  conf.mat<-table(predicted=lda.fitCV.pred$class, true=test_data$type)
  # compute the error rate for each groupCV
  errorCV[i]<-1-sum(diag(conf.mat))/dim(test_data)[1]
}
error.rate<-sum(errorCV)/10
```

```
## [1] 0.3854482
```

The error rate is higher than the training error computed before without 10-fold cross validation (0.3271028). It seems reasonable because we split our data set in test-data and train-data using the partition provided by the variable groupCV and we perform LDA on the train-data and we make the prediction on the test-data.

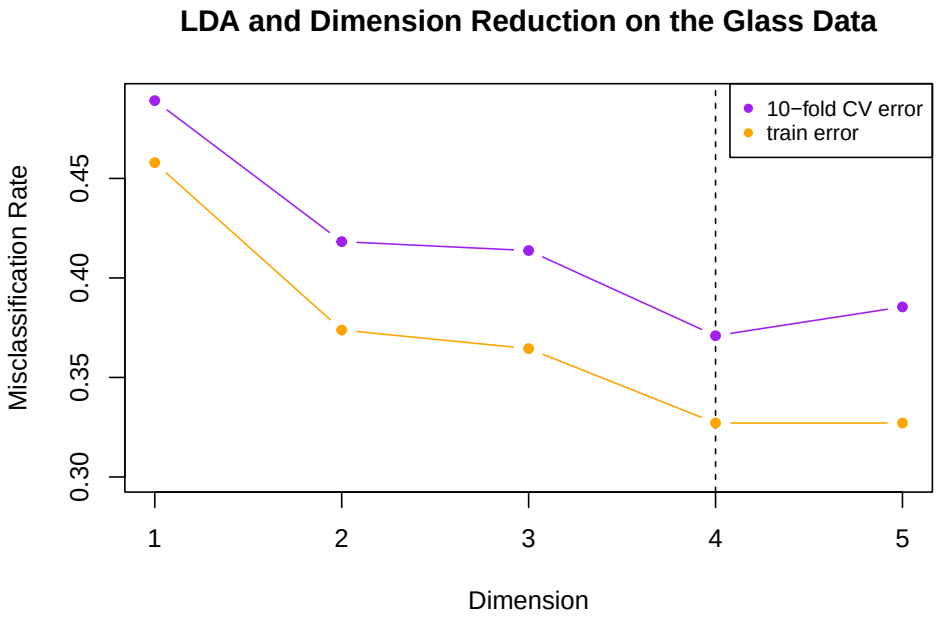
In order to compare the training error and 10-fold cross validation error for each reduced-rank LDA classifier we plot them.

```
k <- length(unique(glass$type))
train.error<-c()
for (j in 1:k-1)
{
  train_data<-glass[,1:10]
  lda.fit<-lda(type~., data=train_data)
  lda.fit.pred<-predict(lda.fit, train_data, dimen=j)
  conf.mat<-table(predicted=lda.fit.pred$class, true=train_data$type)
  train.error[j]<-1-sum(diag(conf.mat))/dim(train_data)[1]
}
train.error
```

```
## [1] 0.4579439 0.3738318 0.3644860 0.3271028 0.3271028
```

```
CV.error<-c()
for (j in 1:k-1)
{
  errorCV<-c()
  for (i in 1:10)
  {
    v<-c(which(glass2$groupCV==i))
    test_data<-glass2[v,1:10]
    train_data<-glass2[-v,1:10]
    lda.fitCV<-lda(type~., data=train_data)
    lda.fitCV.pred<-predict(lda.fitCV, test_data, dimen=j)
    conf.mat<-table(predicted=lda.fitCV.pred$class, true=test_data$type)
    errorCV[i]<-1-sum(diag(conf.mat))/dim(test_data)[1]
  }
  CV.error[j]<-sum(errorCV)/10
}
CV.error
```

```
## [1] 0.4890922 0.4182441 0.4137569 0.3709695 0.3854482
```



As we expected the 10-fold CV error is higher than the train error with respect to each reduced-rank LDA classifier, since it is computed on predicted data which have not been used in the model.

From the plot we obtain the minimum value corresponds to dimen = 4, hence it is preferable to choose that dimension with respect to the others.

In addition, considering the errors computed for each reduced-rank LDA classifier obtained by 10-fold cross validation of each class, we can see that for every type the error decreases significantly until dimension 4, while the full-rank LDA error increases for class WinF and WinNF.

```
matrix.error<-matrix(0,nrow = 6,ncol = 5)
test.error<-c()
for (j in 1:5)
{
  folds.class<-matrix(0,nrow=6,ncol = 10)
  for (i in 1:10)
  {
    v<-c(which(glass2$groupCV==i))
    test_data<-glass2[v,1:10]
    train_data<-glass2[-v,1:10]
    lda.fitCV<-lda(type~., data=train_data)
    lda.fitCV.pred<-predict(lda.fitCV, test_data, dimen=j)
    conf.mat<-table(predicted=lda.fitCV.pred$class, true=test_data$type)
    for (k in 1:6)
    {
      folds.class[k,i]<-1-diag(conf.mat)[k]/sum(conf.mat[,k])
    }
    matrix.error[,j]<-apply(folds.class,1,FUN=mean,na.rm=TRUE)
  }
}
```

Table 21: 10-fold CV error varying the dimension

	dimen1	dimen2	dimen3	dimen4	dimen5
WinF	0.4638	0.3229	0.2962	0.3252	0.3355
WinNF	0.3333	0.3974	0.4128	0.3165	0.3486
Veh	1	1	1	1	1
Con	0.7778	0.5556	0.5556	0.5556	0.5556
Tabl	1	0.75	0.5833	0.5	0.5
Head	0.2333	0.2333	0.2111	0.2111	0.2111

This means that dimension 4 could be considered an optimal classifier in our model, since with dimen = 4 all the errors for each class correspond to the minimum possible value.