

MULTIVARIATE STATISTICAL ANALYSIS

PROBLEM SET 2

Exercise 1

Consider the data set `psych`, which contains 24 psychological tests ($t_i, \forall i \in \{1, \dots, 24\}$) administered to 301 students, with ages ranging from 11 to 16, in a suburb of Chicago:

- the 1st group is made of 156 students (74 boys, 82 girls) from the *Pasteur School*;
- the 2nd group is made of 145 students (72 boys, 73 girls) from the *Grant-White School*.

```
psych_0 = read.table("data/psych.txt", header = T)
dim_p = dim(psych_0)
colnames(psych_0) = c(c("case", "sex", "age"), paste0("t_", 1:(dim_p[2] - 4)), "group")
psych_0[2] = tolower(unlist(psych_0[2]))
psych_0[28] = tolower(unlist(psych_0[28]))
```

case	sex	age	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_10	t_11	t_12	t_13
1	m	13.1	20	31	12	3	40	7	23	22	9	78	74	115	229
2	f	13.6	32	21	12	17	34	5	12	22	9	87	84	125	285
3	f	13.1	27	21	12	15	20	3	7	12	3	75	49	78	159
4	m	13.2	32	31	16	24	42	8	18	21	17	69	65	106	175
5	f	12.2	29	19	12	7	37	8	16	25	18	85	63	126	213
6	f	14.1	32	20	11	18	31	3	12	25	6	100	92	133	270
t_14	t_15	t_16	t_17	t_18	t_19	t_20	t_21	t_22	t_23	t_24	group				
170	86	96	6	9	16	3	14	34	5	24	pasteur				
184	85	100	12	12	10	-3	13	21	1	12	pasteur				
170	85	95	1	5	6	-3	9	18	7	20	pasteur				
181	80	91	5	3	10	-2	10	22	6	19	pasteur				
187	99	104	15	14	14	29	15	19	4	20	pasteur				
164	84	104	6	6	14	9	2	16	10	22	pasteur				

The 24 tests corresponds to the following subjects:

	test
t_1	visual perception
t_2	cubes
t_3	paper form board
t_4	flags
t_5	general information
t_6	paragraph comprehension
t_7	sentence completion
t_8	word classification
t_9	word meaning

	test
t_10	addition
t_11	code
t_12	counting dots
t_13	straight-curved capitals
t_14	word recognition
t_15	number recognition
t_16	figure recognition
t_17	object-number
t_18	number-figure
t_19	figure-word
t_20	deduction
t_21	numerical puzzles
t_22	problem reasoning
t_23	series completion
t_24	arithmetic problems

Note that the variable `case` does not provide any important information as it only corresponds to an enumeration of the students, who were tested in sequential order (containing some gaps probably due to the absence of data for some of the students).

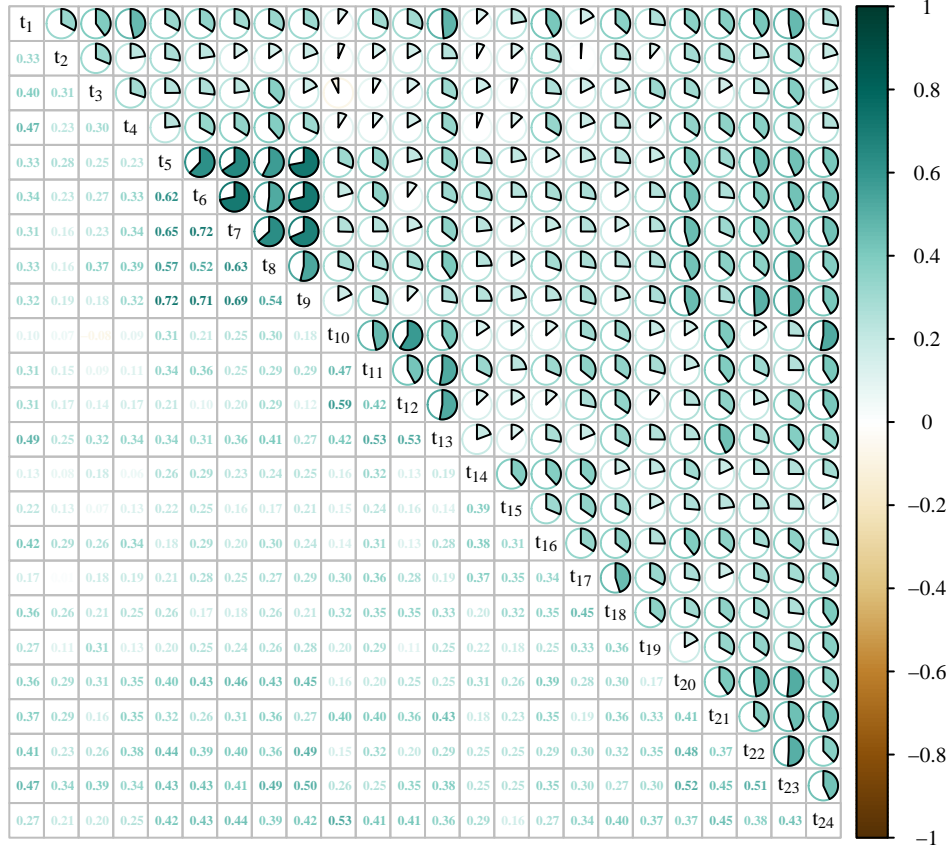
1.1

In performing the factor analysis we are only interested in the 24 variables corresponding to the psychological tests, hence we remove the variables `case`, `age` and `sex` from our dataset. Moreover, we are asked to use only the Grant-White students data, so we subset the remaining data frame according to the request.

```
psych_1 = psych_0[, 4:28]
gw = subset(psych_1, group == "grant", select = -group)
```

Before fitting the model, we scale our data and examine the correlation matrix. Indeed correlation between variables is the object of interest in *Factor Analysis*. Since we have a very large number of variables, we choose not to display the values of the matrix directly, but we rather visualize them with a plot.

```
gws = scale(gw)
cor_gws = cor(gws)
dim_gws = dim(gws)
colnames(cor_gws) = paste0("$t[", 1:(dim_p[2] - 4), "]")
rownames(cor_gws) = colnames(cor_gws)
par(family = "serif")
corrplot.mixed(cor_gws, upper = "pie",
               upper.col = COL2("BrBG"), lower.col = COL2("BrBG"),
               number.cex = 0.4, tl.col = "black", tl.cex = 0.7, cl.cex = 0.7)
```



```
neg_cor_gws = ((24^2 - sum(sign(cor_gws)))) / 2 / 2
```

From the correlation matrix we can note that:

- all the correlation except for `neg_cor_gws = 1` are positive, moreover the majority of them is less than 0.5;
- by just looking at the correlation matrix, it is difficult to guess whether 5 or 6 common factors are an appropriate choice or not.

In order to obtain the maximum likelihood solution for $m = 5$ and $m = 6$ factors in R we can use the built-in function `factanal()`.

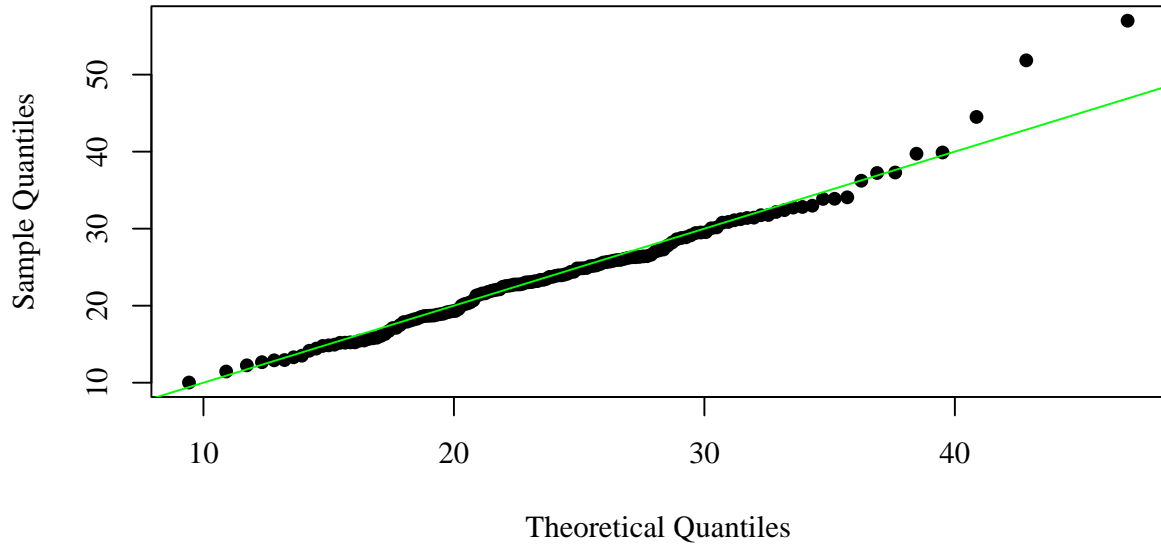
Before proceeding with the computation, we would like to recall that the *maximum likelihood* method, unlike the *principal component method*, relies on the necessary assumption of normality of the *common factors* (\mathbf{F}) and of the *specific error terms* (ε). In particular, if $\mathbf{F} = (F_1, \dots, F_m)$ and $\varepsilon = (\varepsilon_1, \dots, \varepsilon_p)$ are normally distributed, then

$$\mathbf{X} = \mathbf{L}\mathbf{F} + \varepsilon \sim \mathcal{N}(\mu, \Sigma), \text{ with } \mathbf{L} \in \mathbb{R}^{p \times m}.$$

We can check the normality by observing that our input data $\mathbf{x} \in \mathbb{R}^{24}$, which was previously rescaled, actually comes from a $\mathbf{X} \sim \mathcal{N}(0, I)$.

For this purpose we look at the Q-Q plot of the squared Mahalanobis distances *vs* a χ^2_{24} .

```
par(family = "serif", mar = c(4, 4, 1, 1))
d = mahalanobis(gws, center = colMeans(gws), cov = cov(gws))
plot(qchisq(ppoints(d), df = ncol(gws)), sort(d), pch = 16,
     xlab = "Theoretical Quantiles", ylab = "Sample Quantiles")
abline(0, 1, col = "green")
```



The plot shows that the variables jointly seem to follow a gaussian behaviour: except for the last 3 points, which create a heavy right tail the other points lies on the Q-Q line.

We now proceed with the computation of the maximum likelihood solution, first with $m = 5$ factors, then with $m = 6$ factors (without any rotation):

```
faml_5 = factanal(gws, factors = 5, rotation = "none")
load_5 = faml_5$loadings[, ]
```

	Factor1	Factor2	Factor3	Factor4	Factor5
t_1	0.5549	-0.0032	0.4659	-0.1495	0.0015
t_2	0.3444	-0.0287	0.2917	-0.0563	0.1250
t_3	0.3734	-0.1422	0.4267	-0.1045	0.0418
t_4	0.4634	-0.1044	0.3032	-0.1128	0.1482
t_5	0.7226	-0.2536	-0.2249	-0.0756	-0.0044
t_6	0.7208	-0.3742	-0.1685	-0.0139	-0.1453
t_7	0.7278	-0.3355	-0.2323	-0.1317	0.0131
t_8	0.6917	-0.1442	-0.0421	-0.1066	0.0801
t_9	0.7232	-0.4245	-0.1967	0.0169	-0.0214
t_10	0.5182	0.6034	-0.3795	0.0411	0.1158
t_11	0.5701	0.3495	-0.0240	0.0649	-0.3670
t_12	0.4872	0.5444	0.0052	-0.1179	0.1277
t_13	0.6305	0.3467	0.2011	-0.3833	-0.2058
t_14	0.3929	-0.0013	0.0648	0.3688	-0.2378
t_15	0.3456	0.0268	0.1282	0.3678	-0.1281
t_16	0.4559	0.0247	0.3781	0.2755	-0.0855
t_17	0.4530	0.1283	0.0333	0.4382	-0.1130
t_18	0.4749	0.2521	0.2182	0.2588	0.0177
t_19	0.4179	0.0511	0.1376	0.1964	-0.0669
t_20	0.5961	-0.1672	0.1806	0.1546	0.2271
t_21	0.5741	0.2267	0.1539	0.0252	0.1590
t_22	0.5946	-0.1395	0.1803	0.1287	0.0982
t_23	0.6650	-0.0636	0.2131	0.0332	0.2445
t_24	0.6571	0.1864	-0.1262	0.1451	0.1292

```
faml_6 = factanal(gws, factors = 6, rotation = "none")
load_6 = faml_6$loadings[, ]
```

	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
t_1	0.5486	0.0039	0.4562	-0.1968	-0.0599	0.0333
t_2	0.3388	-0.0273	0.3009	-0.1585	0.0715	0.2322
t_3	0.3725	-0.1392	0.4443	-0.1107	0.0336	-0.2323
t_4	0.4600	-0.1066	0.3043	-0.1332	0.1257	-0.0871
t_5	0.7243	-0.2605	-0.2170	-0.0734	-0.0261	0.0920
t_6	0.7240	-0.3674	-0.1557	0.0278	-0.1458	0.0009
t_7	0.7329	-0.3539	-0.2340	-0.0919	0.0229	-0.1481
t_8	0.6953	-0.1550	-0.0401	-0.1049	0.0908	-0.2071
t_9	0.7277	-0.4211	-0.1804	0.0539	-0.0332	0.0922
t_10	0.5131	0.5871	-0.3853	-0.0239	0.1601	0.0291
t_11	0.5786	0.3898	-0.0434	0.0797	-0.4217	0.1269
t_12	0.4816	0.5361	-0.0146	-0.1655	0.1279	-0.1027
t_13	0.6175	0.3280	0.1533	-0.3573	-0.2278	-0.1395
t_14	0.3978	0.0305	0.0803	0.3532	-0.1307	0.0058
t_15	0.3494	0.0578	0.1457	0.3323	-0.0393	0.0969
t_16	0.4568	0.0562	0.3879	0.2097	-0.0402	0.0753
t_17	0.4744	0.1802	0.0697	0.5696	0.0082	-0.2565
t_18	0.4783	0.2777	0.2330	0.2208	0.0730	0.0072
t_19	0.4218	0.0713	0.1544	0.1842	-0.0259	-0.0171
t_20	0.5961	-0.1556	0.2009	0.0750	0.2310	0.0915
t_21	0.5706	0.2318	0.1513	-0.0958	0.1371	0.2158
t_22	0.5970	-0.1208	0.1977	0.0858	0.0702	0.1688
t_23	0.6616	-0.0583	0.2287	-0.0376	0.2257	0.0688
t_24	0.6561	0.1904	-0.1127	0.0757	0.1584	0.0672

It is remarkable that in the case $m = 5$ all but two variables load on the first factor higher than on any other. This makes any factor interpretation very difficult, at least without applying any rotation to the loadings. We will discuss it in more detail in the next point.

Then we proceed with the computation of the proportion of total sample variance due to each factor. We recall that the proportion of total sample variance due to the k^{th} factor is defined as

$$\text{prop_var}(k) = \frac{\sum_{j=1}^p \hat{l}_{j,k}^2}{\text{trace}(\mathbf{S})},$$

with $\hat{\mathbf{L}} = (\hat{l}_{j,k})_{\substack{j=1,\dots,p \\ k=1,\dots,m}}$ factor loadings and \mathbf{S} sample covariance matrix.

Due to the scaling performed at the beginning of the computation in our case $\text{trace}(\mathbf{S}) = \text{size}(\mathbf{S}) = 24$ (it is indeed a sample correlation matrix).

```
prop_var_5 = colSums(load_5^2) / dim_gws[2]
```

	Factor1	Factor2	Factor3	Factor4	Factor5
prop_var_5	0.3159	0.0698	0.0548	0.04	0.0223

```
prop_var_6 = colSums(load_6^2) / dim_gws[2]
```

	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
prop_var_6	0.3168	0.0711	0.0563	0.0417	0.0212	0.0175

We could get the associated cumulative proportion of total sample variance by applying the `cumsum()` function to the previous 2 variables. However, these computations are also performed as a part of the output of the command `factanal()`, together with the sum of the squares of the loadings:

faml_5

	Factor1	Factor2	Factor3	Factor4	Factor5
ss_load_5	7.5813	1.6743	1.3161	0.9589	0.5351
prop_var_5	0.3159	0.0698	0.0548	0.0400	0.0223
cum_var_5	0.3159	0.3856	0.4405	0.4804	0.5027

faml_6

	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
ss_load_6	7.6024	1.7068	1.3515	1.0000	0.5086	0.4192
prop_var_6	0.3168	0.0711	0.0563	0.0417	0.0212	0.0175
cum_var_6	0.3168	0.3879	0.4442	0.4859	0.5071	0.5245

Both models seem to fit very poorly. A general criterion, for the choice of the number of factors is to take the smallest m such that the total proportion of variance due to the m factors is at least 80%. However, in both our cases ($m = 5, 6$), the models explain about 50% (respectively 50.27% and 52.45%) of the total variance collectively. Hence, the result is not satisfactory.

Next, as requested, we report below the specific variances $(\psi_j)_{j=1}^{24}$, again for both $m = 5$ and $m = 6$. In this case we directly exploit the output of `factanal()` in order not to have to recalculate the values of the specific variances of the factors by hand. We report the results of the computation below:

psi_5 = faml_5\$uniquenesses

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_10	t_11	t_12
0.4526	0.7766	0.6456	0.6477	0.3573	0.2907	0.2863	0.4812	0.2573	0.2082	0.4134	0.4361
t_13	t_14	t_15	t_16	t_17	t_18	t_19	t_20	t_21	t_22	t_23	t_24
0.2525	0.6489	0.7118	0.5654	0.5724	0.596	0.7607	0.5086	0.5695	0.5683	0.4474	0.4799

psi_6 = faml_6\$uniquenesses

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_10	t_11	t_12
0.4474	0.7098	0.5771	0.6433	0.346	0.2946	0.2519	0.4288	0.2481	0.2164	0.3111	0.4262
t_13	t_14	t_15	t_16	t_17	t_18	t_19	t_20	t_21	t_22	t_23	t_24
0.2885	0.6925	0.732	0.5864	0.3473	0.5857	0.7583	0.5128	0.5233	0.5491	0.4494	0.4853

Finally, we need to assess the accuracy of the approximations of the correlation matrices. For this purpose, for both models we analyse the residual matrix given by the difference between the actual correlation matrix, \mathbf{R} , and the correlation matrix given by the approximation performed by the maximum likelihood method,

i.e. $\mathbf{S} = \hat{\mathbf{L}}\hat{\mathbf{L}}^T + \hat{\Psi}$, where $\hat{\Psi} = \text{diag}((\psi_j)_{j=1}^{24})$.

We first compare the squared Frobenius norm of the approximation matrices with the sum of the squares of the neglected eigenvalues, i.e. $\sum_{i=m+1}^{\text{size}(\mathbf{S})} \lambda_i^2$, in order to check if the following inequality is fulfilled:

$$\left\| \mathbf{R} - \left(\hat{\mathbf{L}}\hat{\mathbf{L}}^T + \hat{\Psi} \right) \right\|_F^2 \leq \sum_{i=m+1}^{\text{size}(\mathbf{S})} \lambda_i^2.$$

Then, we compare the two squared Frobenius norms in order to see which approximation is more accurate.

```
eig = eigen(cor_gws)$values
residual_5 = cor_gws - (load_5 %*% t(load_5) + diag(psi_5))
eig_negl_5 = eig[(5 + 1):dim_gws[2]]
comparison_5 = c(sum(residual_5^2), sum(eig_negl_5^2))
```

	ss_residual_5	ss_eig_negl_5
comparison_5	0.7335	5.7823

Then we repeat the same computation for $m = 6$:

```
residual_6 = cor_gws - (load_6 %*% t(load_6) + diag(psi_6))
eig_negl_6 = eig[(6 + 1):dim_gws[2]]
comparison_6 = c(sum(residual_6^2), sum(eig_negl_6^2))
```

	ss_residual_6	ss_eig_negl_6
comparison_6	0.602	4.9392

We get

$$\begin{aligned} m = 5 : \quad & 0.7335059 \leq 5.7822848 \\ m = 6 : \quad & 0.6020222 \leq 4.9391922 \end{aligned}$$

so the inequality is satisfied. Moreover, it is evident that in both cases the approximation error of the correlation matrix is not negligible.

Another possible way to see if 5 or 6 factor are enough to explain the observed covariances is to consider test performed automatically by the command `factanal()` whose p-value is displayed at the end of the output. We obtain respectively:

```
faml_5$PVAL
faml_6$PVAL
```

p-value_5	p-value_6
0.1558614	0.2642373

Let us explain the meaning of the test performed above.

The function uses the model's likelihood estimation to check the quality of the fitting of our factors by testing

$$H_0 : \Sigma = \mathbf{L}\mathbf{L}^T + \Psi \quad \text{vs} \quad H_1 : \Sigma \text{ generic positive definite matrix.}$$

Both our models have high p-values (> 0.05) hence it seems that both the number of factor is reasonable in both cases.

In conclusion, both choices are acceptable, but in some sense inaccurate. The improvement given by the choice of $m = 6$ is not particularly significant, hence we tend to prefer $m = 5$. Indeed the last factor obtained with $m = 6$ accounts only for the 1.75% of the total sample variance and the difference between the squared Frobenius norms of the residual matrices shares the same order of magnitude.

1.2

We now have to give an interpretation to the common factors in the $m = 5$ solution. Without any rotation the loadings are pretty difficult to comprehend. Indeed, as we noticed in the previous point, when $m = 5$ almost all variables load on the first factor higher than on the other four factors. Therefore, a rotation may help in the interpretation process. As requested, we perform the **Varimax** rotation.

```
faml_5_var = factanal(gws, factors = 5, rotation = "varimax")
load_5_var = faml_5_var$loadings[, ]
```

	Factor1	Factor2	Factor3	Factor4	Factor5
t_1	0.1654	0.6549	0.1250	0.1810	0.2066
t_2	0.1079	0.4416	0.0871	0.0954	0.0024
t_3	0.1341	0.5595	-0.0473	0.1115	0.0934
t_4	0.2305	0.5333	0.0895	0.0811	0.0124
t_5	0.7383	0.1893	0.1916	0.1486	0.0547
t_6	0.7724	0.1867	0.0318	0.2477	0.1243
t_7	0.7983	0.2140	0.1427	0.0883	0.0502
t_8	0.5710	0.3429	0.2391	0.1275	0.0423
t_9	0.8079	0.2024	0.0332	0.2188	-0.0072
t_10	0.1807	-0.1082	0.8451	0.1803	0.0264
t_11	0.1952	0.0661	0.4233	0.4365	0.4177
t_12	0.0297	0.2322	0.6944	0.1022	0.1285
t_13	0.1863	0.4329	0.4793	0.0775	0.5382
t_14	0.1846	0.0614	0.0443	0.5522	0.0797
t_15	0.1043	0.1223	0.0586	0.5089	-0.0028
t_16	0.0698	0.4061	0.0559	0.5087	0.0540
t_17	0.1543	0.0716	0.2104	0.5947	-0.0269
t_18	0.0323	0.2999	0.3219	0.4576	0.0043
t_19	0.1563	0.2209	0.1440	0.3785	0.0451
t_20	0.3728	0.4614	0.1265	0.2930	-0.1939
t_21	0.1717	0.3980	0.4312	0.2382	-0.0004
t_22	0.3637	0.4232	0.1139	0.3204	-0.0689
t_23	0.3615	0.5421	0.2482	0.2307	-0.1147
t_24	0.3680	0.1786	0.4952	0.3208	-0.0683

We choose not to visualize the results in a plot since there are too many factors and variables and therefore it would not have been helpful.

After the rotation, things become a little better: as expected, the loadings are in general smaller or larger than the previous ones, and this facilitates the interpretation of the factors. In particular:

1. the variables t_5, t_6, t_7, t_8 and t_9 load highly on the first common factor. The psychological tests associated to these variables primarily assess the language-related capacities of an individual, including reading comprehension, vocabulary knowledge, word associations, sentence construction and general knowledge. Hence, we can interpret the first factor as *verbal ability*;
2. the second factor is determined by the variables from t_1 to t_4 together with t_{20}, t_{22} and t_{23} . The first four tests measure the spatial ability of an individual, while the last three tests assess the logical ability of an individual. Hence, we choose to assign the second factor the label *logical and spatial ability*;
3. the variables t_{10} and t_{12} load highly on the third factor, which is also determined by the variables t_{21} and t_{24} . They refer to psychological tests that assess cognitive capacities related to numerical processing, mathematical reasoning and arithmetic skills. We refer to the fourth factor as *numerical/mathematical ability*;

4. the variables from t_{14} to t_{19} determine the fourth common factor. The tests associated to these variables measure an individual's capacity of recognising numbers, words and figures and of making associations between them. Hence, the fourth factor can be interpreted as *recognition and association ability*;
5. the fifth factor is solely determined by the variable t_{13} . Hence we label the factor as its representative test, i.e. *straight-curved capitals*. It is immediate to observe that it is the only factor without an abstract meaning. This could be due to the fact that proportion of variance explained by the factor is 0.026, which is too low to have a significative impact.

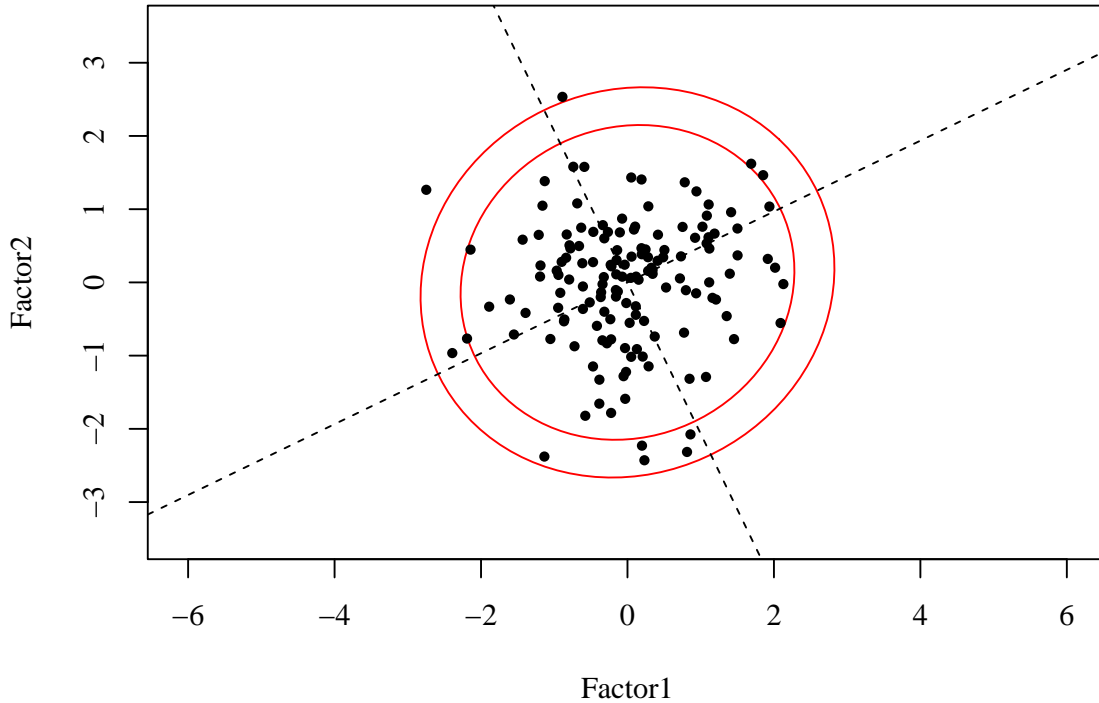
Finally it is remarkable that the variable t_{11} loads uniformly on the last three common factors hence it influences them similarly. This could be reasonable taking into account the psychological test associated with the variable.

Before we move to the next step we want to underline that we decided not to fix a threshold value to assess significance of factor loadings. This choice is motivated by the fact that the total sample variance explained by the 5 factors is only 50.27%. Indeed this leads to the shortage of very high loadings and at the same time allows the presence of variables that have not much influence on any factor. Moreover by doing so we obtained a partition of our variables among the factors (with the only minor exception given by t_{11}).

1.3

We report below the scatterplot of the first two factor scores for the $m = 5$ solution obtained by the regression method, as requested.

```
faml_5_var_reg = factanal(gws, factors = 5, rotation = "varimax", scores = "regression")
score_5_var_reg = faml_5_var_reg$scores[, 1:2]
mu_5_var_reg = colMeans(score_5_var_reg[, 1:2])
sigma_5_var_reg = cov(score_5_var_reg[, 1:2])
eig_var_reg = eigen(sigma_5_var_reg, symmetric = T)
```



It seems there is no particular correlation between the two factors. In fact, if we compute it explicitly we

obtain 0.074. Moreover the covariance matrix turns out to be

$$\Sigma_{fs} = \begin{pmatrix} 0.8673433 & 0.0607289 \\ 0.0607289 & 0.7712268 \end{pmatrix}.$$

As we can see from the scatterplot, the correlation is really close to 0. We should have expected it since the factors scores are the estimated values of the common factors and in the theoretical model the covariance between any couple of common factors is 0, which implies that also their correlation is 0. In particular the theoretical covariance matrix of the factors is equal to the identity matrix and our estimated covariance matrix is quite close to it: the estimated variance of the second factor is slightly smaller than what it should be, but it is still acceptable taking into account that we are considering only 5 common factors which actually explain only the 50.27% of the total sample variance.

Finally, in order to analyse better the distribution of our data we decided to display the ellipsoids containing the 95% and the 99% of the points. We can see that the ellipsoids are in fact circles (nearly) which confirms that the first two common factors are jointly normally distributed.

1.4

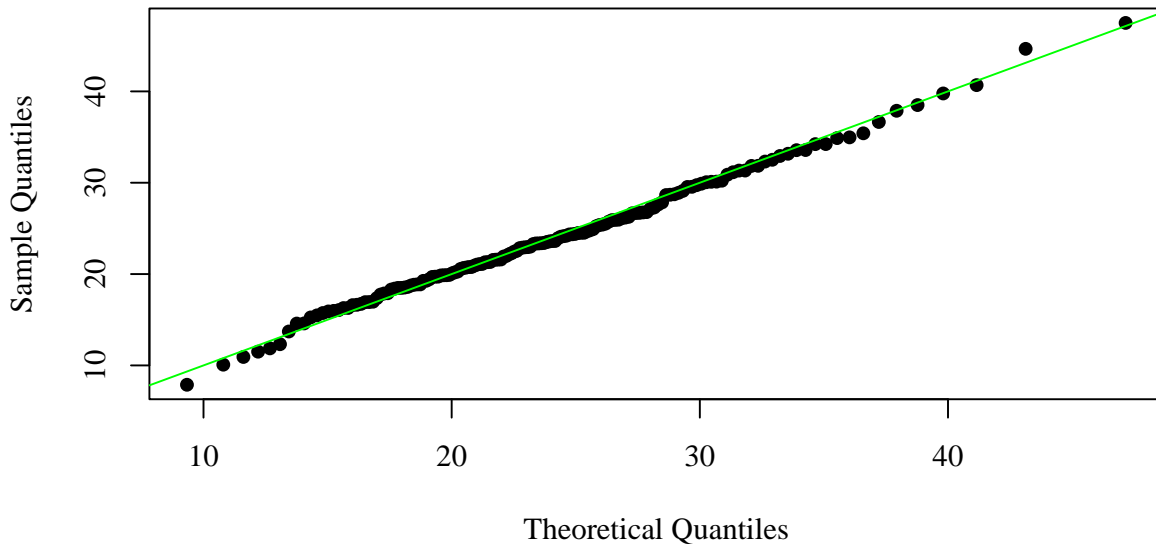
Let us now consider the `psych` dataset restricted to the Pasteur students.

```
pa = subset(psych_1, group == "pasteur", select = -group)
pas = scale(pa)
dim_pas = dim(pas)
```

Before obtaining the maximum likelihood solution (still with $m = 5$ factors), as we did for the Grant-White students data we first check if the normality assumption is satisfied.

Similarly to point 1 we look at the Q-Q plot of the squared Mahalanobis distances *vs* a χ^2_{24} .

```
par(family = "serif", mar = c(4, 4, 1, 1))
d = mahalanobis(pas, center = colMeans(pas), cov = cov(pas))
plot(qchisq(ppoints(d), df = ncol(pas)), sort(d), pch = 16,
     xlab = "Theoretical Quantiles", ylab = "Sample Quantiles")
abline(0, 1, col = "green")
```



The Chi-squared Q-Q plot of the Mahalanobis distance shows that almost all the points lie on the Q-Q line. Hence, we can say that the sum of the squares of our variables $((t_1, \dots, t_{24}))$ is χ^2_{24} distributed and so our variables can be considered jointly distributed as a multivariate gaussian.

We can now proceed with the computation of the maximum likelihood solution with `Varimax` rotation for $m = 5$.

```
faml_5_pas = factanal(pas, factors = 5, rotation = "varimax")
load_5_pas = faml_5_pas$loadings[, ]
```

	Factor1	Factor2	Factor3	Factor4	Factor5
t_1	0.3138	0.5777	0.1376	0.0960	0.0693
t_2	0.0369	0.5170	-0.0111	0.0582	-0.1445
t_3	0.0977	0.4437	-0.1772	0.0028	0.0996
t_4	-0.0161	0.6712	0.1896	0.1702	0.0115
t_5	0.8059	0.0428	-0.0668	0.0997	0.1429
t_6	0.7815	0.1574	0.0975	0.0566	0.2024
t_7	0.9043	0.0790	0.0051	0.1090	-0.0617
t_8	0.6838	0.1689	0.1388	0.1515	0.0140
t_9	0.7751	0.2488	0.0375	0.1015	0.1564
t_10	0.1408	-0.2079	0.1161	0.4998	0.6409
t_11	0.3490	0.0683	0.2313	0.6706	0.0649
t_12	0.0781	0.0967	-0.0064	0.5258	0.2168
t_13	0.0694	0.2719	0.0863	0.5436	0.0517
t_14	0.0418	0.0490	0.6900	-0.0261	0.0405
t_15	-0.1326	0.1253	0.6128	-0.1097	0.0973
t_16	0.0833	0.3864	0.4753	0.1758	0.1610
t_17	0.0675	-0.0544	0.5226	0.2891	0.0875
t_18	0.1004	-0.0052	0.4649	0.0891	-0.0076
t_19	0.0668	0.2442	0.3567	0.2405	0.0340
t_20	0.1231	0.5141	0.1888	0.0130	0.0963
t_21	0.2840	0.3871	0.1411	0.1951	0.4323
t_22	0.4685	0.4807	0.0289	0.1515	0.0513
t_23	0.3571	0.5871	0.1442	0.0916	0.2988
t_24	0.2180	0.2944	0.2265	0.2397	0.5296

In the analysis of the factor loadings we adopt the same strategy as before: we look at the matrix by rows and we do not set any threshold value. We obtain a perfect partition of the variables among the 5 common factors, in particular:

1. the first factor is determined by the same 5 variables as before, namely t_5, t_6, t_7, t_8 and t_9 . Therefore, it can be interpreted in the exact same way, which is *verbal ability*;
2. as in the previous point, the second factor is influenced by the same 7 variables as before, that are the ones from t_1 to t_4 together with t_{20} , t_{22} and t_{23} . Hence, we can assign to the second factor the same label: *logical and spatial ability*;
3. the third factor is determined by the variables from t_{14} to t_{19} . These same variables previously formed the fourth common factor, hence there was just an exchange of order between the factors. We interpret it as *recognition and association ability*;
4. the variables from t_{11} to t_{13} form the fourth common factor. The tests associated to these variables are respectively *code*, *counting dots* and *straight-curved capitals* which are related to quick visualization skills. We label it *quick visualization/speed ability*;
5. finally the last factor is influenced by the variables t_{10} , t_{21} and t_{23} which previously formed the third common factor together with the variable t_{12} . Despite the absence of t_{12} the factor has not lost its meaning, therefore we interpret it as *numerical/mathematical ability*.

A necessary remark is that the new factors can be viewed as a permutation of the ones we have obtained for the Grant-White students data, but we need to specify that the variable t_{12} moved from the third to the fifth factor (without following the permutation) and that, unlike before, we now manage to give an abstract meaning to all the common factors.

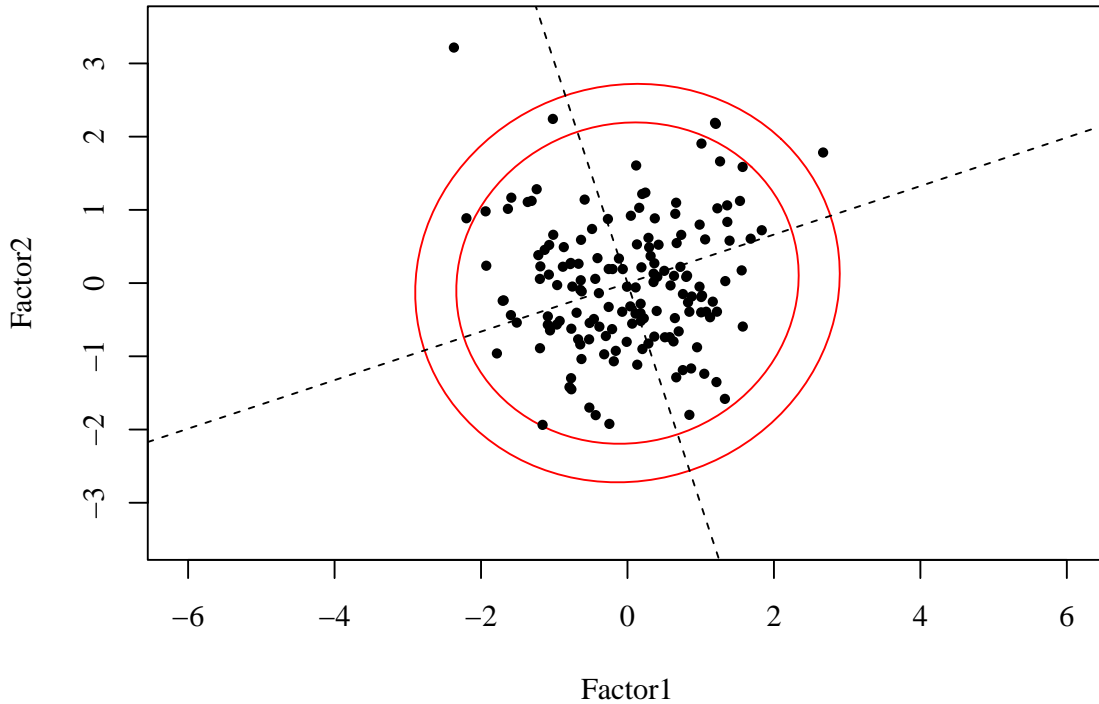
For the sake of completeness, we report the permutation of the factors in compact form:

$$\sigma \in S_5, \text{ such that } \sigma(4) = 3, \sigma(3) = 5, \sigma(5) = 4.$$

1.5

We have already made the scatterplot of the first two factor scores from the rotated MLFA solution for the Grant-School in the point 1.3. We now follow the exact same procedure for the Pasteur school and then we make a comparison between the results.

```
faml_5_pas_reg = factanal(pas, factors = 5, rotation = "varimax", scores = "regression")
score_5_pas_reg = faml_5_pas_reg$scores[, 1:2]
mu_5_pas_reg = colMeans(score_5_pas_reg[, 1:2])
sigma_5_pas_reg = cov(score_5_pas_reg[, 1:2])
eig_pas_reg = eigen(sigma_5_pas_reg, symmetric = T)
```



As for the Grant-White students data it seems there is no particular correlation between the two factors. The correlation between the factors is 0.047 and the covariance matrix is

$$\Sigma_{fs} = \begin{pmatrix} 0.9125682 & 0.040457 \\ 0.040457 & 0.8038656 \end{pmatrix}.$$

Again the factors appears to be jointly normally distributed: almost every point (except for 2) falls inside the ellipsoid containing the 99% of the mass.

Note also that this covariance matrix is closer to the identity matrix than the previous one, however the gap between the variances of the first two factors still holds.

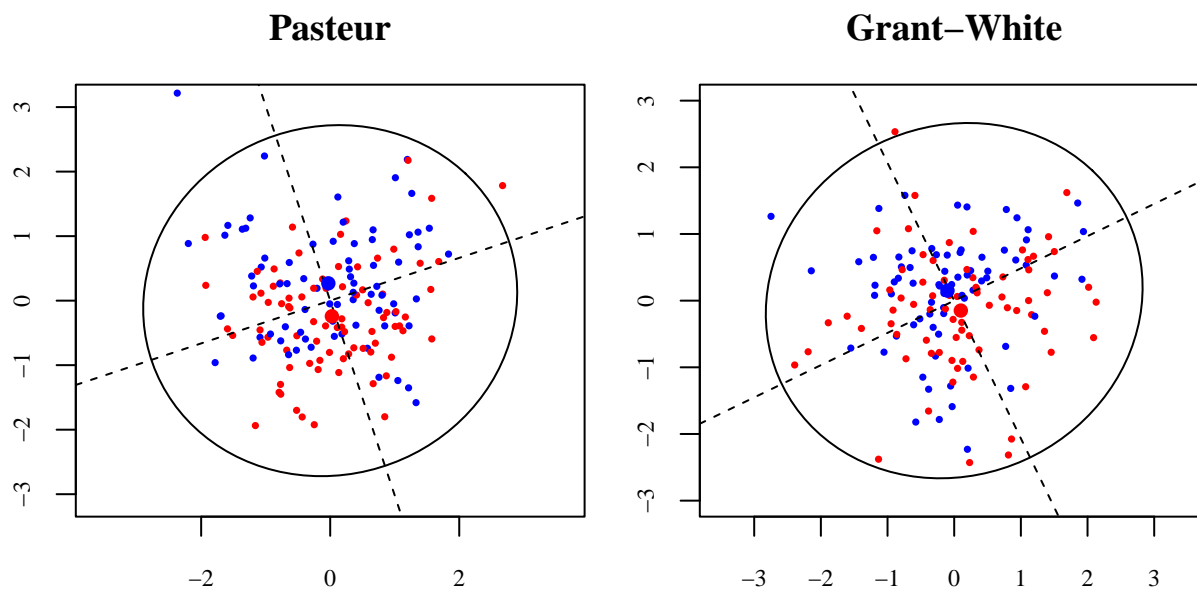
We now analyse the same scatterplots by grouping the data according to some of the initial variables of the dataset `psych` that we have not used in the factor analysis because they did not represent any psychological test, namely `sex` and `age`. Our aim is to see if we can extract any significant relationship between the groups and the results of the tests.

We first group the students according to the variable `sex`.

```
sex_pa = psych_0[1:156, 2]
col_pa = rep("blue", length(sex_pa))
col_pa[sex_pa == "f"] = "red"
mu_pa_sexm = colMeans(score_5_pas_reg[sex_pa == "m", 1:2])
mu_pa_sexf = colMeans(score_5_pas_reg[sex_pa == "f", 1:2])

sex_gw = psych_0[157:301, 2]
col_gw = rep("blue", length(sex_gw))
col_gw[sex_gw == "f"] = "red"
mu_gw_sexm = colMeans(score_5_var_reg[sex_gw == "m", 1:2])
mu_gw_sexf = colMeans(score_5_var_reg[sex_gw == "f", 1:2])
```

In the following plots the blue points (●) refer to male students while the red ones (●) to the female students. We also plot the respective mean points of the two groups with bigger dots (with the same colors).



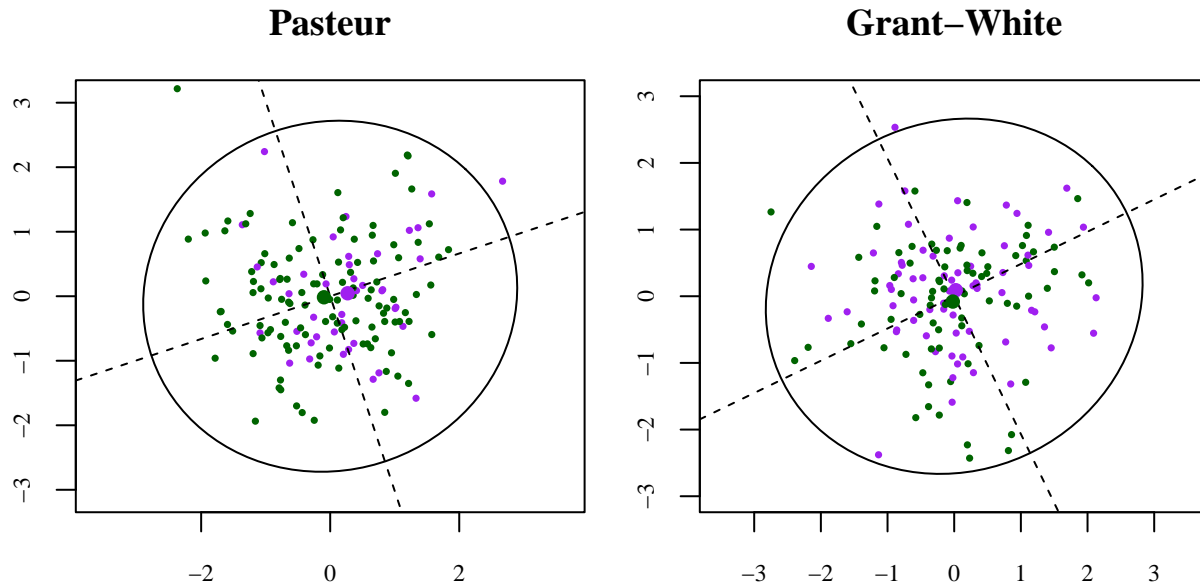
According to the scatterplots female and male students among both schools appears to share almost the same verbal skills, but in both cases male students seems to score higher in the second factor (*logical/spatial ability*).

Then we group the students by age, separating the younger ones (●) (with `age < 13`) from the older ones (●).

```
age_pa = psych_0[1:156, 3]
col_pa = rep("black", length(age_pa))
col_pa[age_pa < 13] = "purple"
col_pa[age_pa >= 13] = "darkgreen"
mu_pa_age1 = colMeans(score_5_pas_reg[age_pa < 13, 1:2])
mu_pa_age2 = colMeans(score_5_pas_reg[age_pa >= 13, 1:2])

age_gw = psych_0[157:301, 3]
```

```
col_gw = rep("black", length(age_gw))
col_gw[age_gw < 13] = "purple"
col_gw[age_gw >= 13] = "darkgreen"
mu_gw_age1 = colMeans(score_5_var_reg[age_gw < 13, 1:2])
mu_gw_age2 = colMeans(score_5_var_reg[age_gw >= 13, 1:2])
```



For the Grant-White students we basically cannot see any particular difference between the distributions of the two groups: the mean points are very close. As for the Pasteur students we get that they score similarly on the second factor (*logical/spatial ability*) while there is a little difference in the first common factor in favour of the younger students. Since we do not know any additional information about the students and about the exact structure on the psychological tests we are not able to say whether it does make sense or not.

Exercise 2

Consider the dataset `pendigits` containing $n = 10992$ observations with 16 numerical variables and 1 categorical variable which is the class attribute ($\text{digit} \in \{0, \dots, 9\}$).

```
pendigits = read.table("data/pendigits.txt", sep = ",", head = F)
names(pendigits) = c(paste0(rep(c("x", "y"), 8), rep(1:8, each = 2)), "digit")
lookup = c("darkgreen", "brown", "lightblue", "magenta", "purple",
           "blue", "red", "lightgreen", "orange", "cyan")
names(lookup) = as.character(0:9)
digit_col = lookup[as.character(pendigits$digit)]
head(pendigits)
```

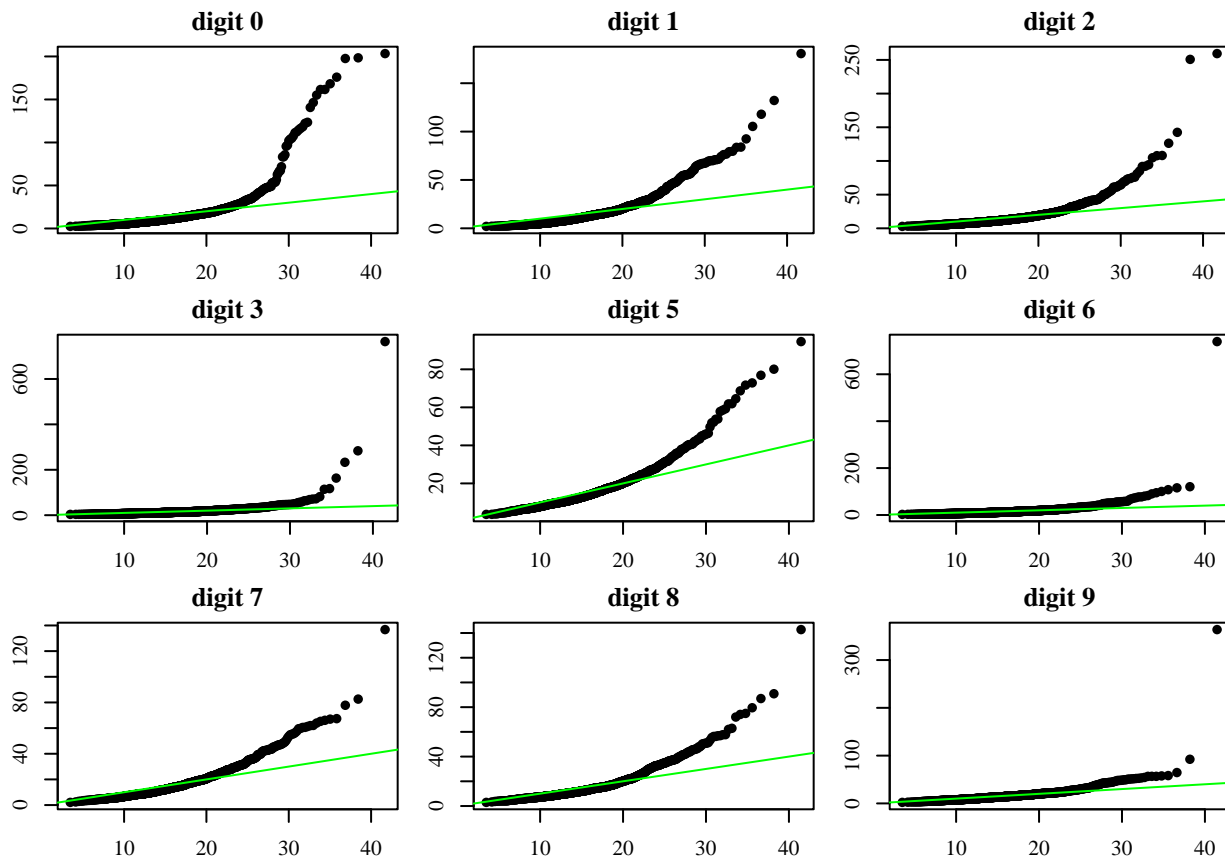
x1	y1	x2	y2	x3	y3	x4	y4	x5	y5	x6	y6	x7	y7	x8	y8	digit
47	100	27	81	57	37	26	0	0	23	56	53	100	90	40	98	8
0	89	27	100	42	75	29	45	15	15	37	0	69	2	100	6	2
0	57	31	68	72	90	100	100	76	75	50	51	28	25	16	0	1
0	100	7	92	5	68	19	45	86	34	100	45	74	23	67	0	4
0	67	49	83	100	100	81	80	60	60	40	40	33	20	47	0	1
100	100	88	99	49	74	17	47	0	16	37	0	73	16	20	20	6

```
count_dig = as.vector(table(pendigits$digit))
prop_dig = as.vector(table(pendigits$digit)) / sum(as.vector(table(pendigits$digit)))
```

2.1

The Linear Discriminant Analysis (LDA) technique relies on the assumption that each different class is multivariate gaussian distributed with different means μ_i (centroids) and with the same covariance matrix Σ . Hence, we should first check if this assumption holds.

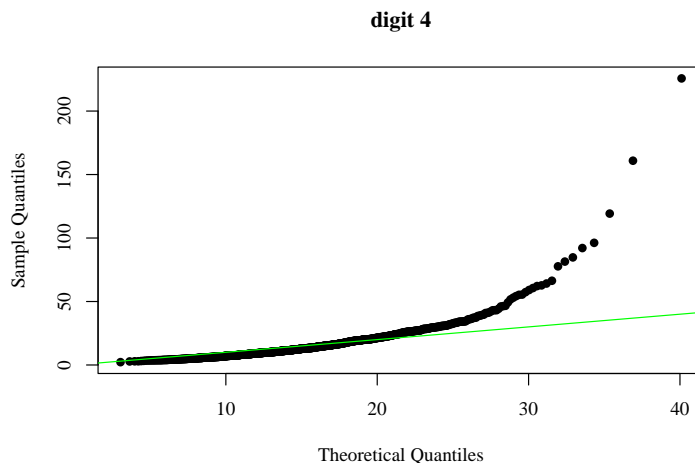
We look at the Q-Q plot of the squared Mahalanobis distances *vs* a χ^2_{16} for each different class.



Let us first note that we removed the digit 4 from our analysis. This is because the last variable of the dataset restricted to this class is entirely made of zeros. Hence, in this case the Mahalanobis distance is not well-defined since the covariance matrix is singular. This problem can be overcome by recalling that by definition a multivariate gaussian vector \mathbf{X} should satisfy the following property:

$$a^T \mathbf{X} + b \sim \mathcal{N}(\mu, \sigma^2), \forall a, b \in \mathbb{R}^{\text{size}(\mathbf{X})}.$$

This definition still holds if $\mathbf{X} = (\mathbf{Y}, 0)$ with \mathbf{Y} multivariate gaussian, indeed the idea is that a constant random variable c can be considered a $\mathcal{N}(c, 0)$. Thus, for the digit 4, we test the normality assumption by just removing the last variable (y_8).



By looking at the Q-Q plots, it is clear that no class is normally distributed: all the digits, except for 6 and 9, seems to have heavy right tails. Actually, the plots of the remaining two variables appear to be closer to the Q-Q line but this is only because in both cases there is an extreme outlier which distorts the figure.

It is also possible to check whether the covariance matrices of the different classes are similar or not. In order to do so we display below the matrix M such that

$$M_{i,j} = \left\| \text{Var}(\text{pendigits}|_{\text{digit}=i}) - \text{Var}(\text{pendigits}|_{\text{digit}=j}) \right\|_F,$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix while $\text{Var}(\cdot)$ denotes the covariance matrix of a random vector.

```
sub_digits_cov = list()
for (i in 1:10){
  sub_digits_cov[[i]] = cov(sub_digits[[i]])
}
mat = matrix(rep(1, 100), ncol = 10)
for (i in 1:10){
  for (j in 1:10){
    mat[i, j] = norm(sub_digits_cov[[i]] - sub_digits_cov[[j]], "f")
  }
}
```

$$\begin{pmatrix} 0 & 5142 & 4321 & 4529 & 3537 & 12931 & 4174 & 4775 & 6987 & 4941 \\ 5142 & 0 & 4301 & 4525 & 4143 & 12660 & 4479 & 3541 & 6711 & 4941 \\ 4321 & 4301 & 0 & 1167 & 2026 & 12320 & 1815 & 2539 & 6235 & 3236 \\ 4529 & 4525 & 1167 & 0 & 2155 & 12348 & 1597 & 2880 & 6219 & 3003 \\ 3537 & 4143 & 2026 & 2155 & 0 & 12315 & 2366 & 3000 & 6429 & 3088 \\ 12931 & 12660 & 12320 & 12348 & 12315 & 0 & 12346 & 11526 & 12410 & 11659 \\ 4174 & 4479 & 1815 & 1597 & 2366 & 12346 & 0 & 2985 & 6211 & 3344 \\ 4775 & 3541 & 2539 & 2880 & 3000 & 11526 & 2985 & 0 & 6185 & 3786 \\ 6987 & 6711 & 6235 & 6219 & 6429 & 12410 & 6211 & 6185 & 0 & 5991 \\ 4941 & 5023 & 3236 & 3003 & 3088 & 11659 & 3344 & 3786 & 5991 & 0 \end{pmatrix}$$

The fact that our classes do not appear to be normally distributed and not to have similar covariance matrices does not compromise the applicability of the LDA since these assumptions are only required for an optimal solution.

We now apply the LDA procedure (already implemented in R). It identifies recursively 9 discriminant variables from the 10 classes.

```
lda_fit = lda(digit ~ ., data = pendigits)
```

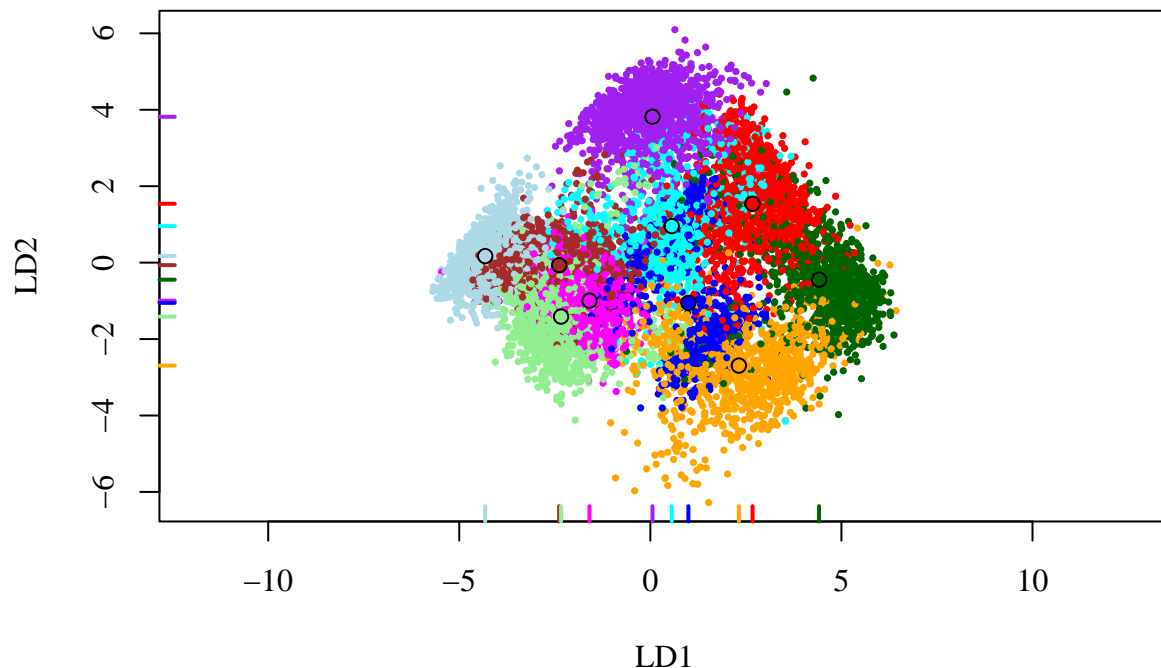
	LD1	LD2	LD3	LD4	LD5	LD6	LD7	LD8	LD9
x1	0.0170	0.0108	0.0011	-0.0302	-0.0096	-0.0052	0.0110	-0.0102	-0.0008
y1	0.0105	0.0302	-0.0186	0.0291	-0.0084	-0.0431	-0.0508	-0.0262	0.0195
x2	0.0062	-0.0024	-0.0024	0.0115	-0.0164	0.0098	-0.0270	-0.0052	0.0114
y2	-0.0341	-0.0386	-0.0114	0.0127	0.0423	-0.0100	0.0372	-0.0501	-0.0700
x3	-0.0243	-0.0223	-0.0123	-0.0172	0.0257	-0.0191	0.0180	-0.0098	0.0010
y3	0.0038	0.0047	-0.0184	-0.0137	-0.0351	-0.0186	-0.0334	0.0452	0.0951
x4	-0.0068	0.0055	0.0056	0.0070	-0.0137	0.0133	-0.0118	0.0015	0.0051
y4	0.0154	-0.0123	0.0102	-0.0004	0.0072	0.0509	-0.0443	-0.0629	-0.0584
x5	0.0029	-0.0088	-0.0244	0.0112	0.0330	0.0155	0.0155	-0.0166	0.0249
y5	-0.0118	-0.0095	-0.0311	-0.0133	-0.0214	-0.0289	0.0292	0.0250	-0.0283
x6	0.0115	0.0190	-0.0064	-0.0102	0.0073	-0.0204	-0.0271	0.0152	-0.0312

	LD1	LD2	LD3	LD4	LD5	LD6	LD7	LD8	LD9
y6	-0.0016	0.0159	-0.0625	-0.0082	-0.0201	-0.0309	0.0093	-0.0062	0.0384
x7	-0.0021	-0.0062	0.0074	-0.0002	0.0045	0.0283	0.0366	-0.0359	0.0439
y7	0.0235	-0.0081	0.0571	0.0383	0.0125	0.0629	-0.0496	-0.0171	-0.0183
x8	-0.0207	0.0178	-0.0157	0.0087	-0.0139	-0.0100	-0.0084	-0.0047	-0.0188
y8	0.0322	-0.0526	-0.0524	-0.0209	-0.0142	-0.0335	0.0125	-0.0040	-0.0073

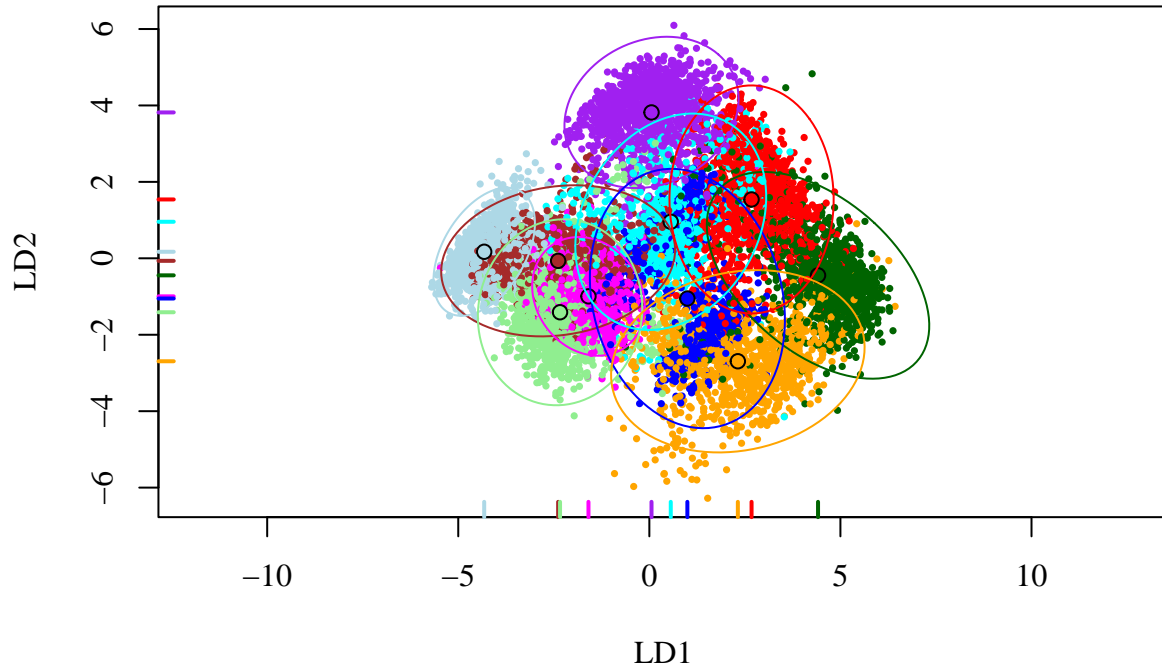
In particular, as the discriminant variables are found iteratively as the direction of highest discrimination in smaller and smaller subspaces, they are ordered by their *power of discrimination*. Therefore a plot of the first two discriminant variables should be rather informative of the degree by which classes can be discriminated.

We now consider only the first two discriminant variables (LD1 and LD2) and we perform a scatterplot of the data restricted on these two components, color coding the observations according to variable `digit_col` and adding the centroids for each class.

```
pred = predict(lda_fit)
centroids = aggregate(pred$x, by = list(pendigits$digit), FUN = mean)
centroids = centroids[, -1]
covariances = list()
for (i in 1:10){
  covariances[[i]] = cov(pred$x[pendigits[, 17] == (i - 1), ])
}
```



The most difficult to discriminate are brown = 1, blue = 5, and cyan = 9.



We have also tried to compute in a more theoretical way the centroids of our 9 classes. First we have collected in a 16×10 matrix the means of our classes, then we have computed a *grand mean* by multiplying this matrix for the vector containing the proportion of observations for each class (`prop_dig`). Finally we have *centered* the mean vector of each class by subtracting the grand mean and we have applied the linear transformation given by the `scaling` given as output by the `lda` procedure.

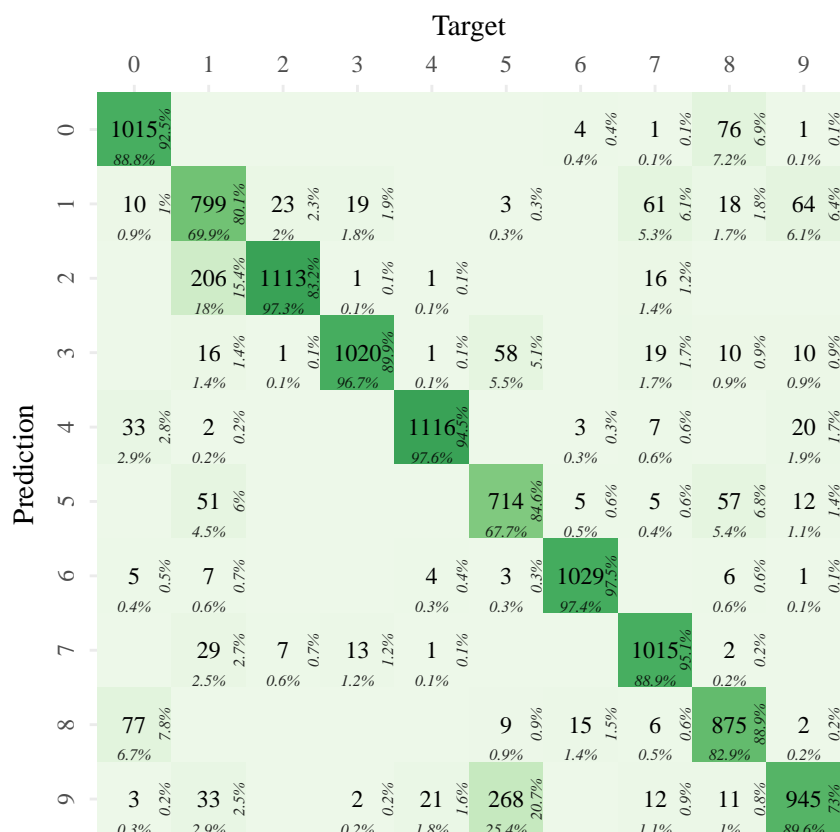
```
all = matrix(rep(0, 160), ncol = 10)
for (i in 1:10){
  all[, i] = as.matrix(colMeans(sub_digits[[i]]))
}
grand_mean = all %*% as.matrix(prop_dig)
centroids_other = matrix(rep(0, 20), ncol = 2)
for (i in 1:10){
  centroids_other[i, ] = t(t(lda_fit$scaling[, 1:2]) %*%
    (as.matrix(colMeans(sub_digits[[i]])) - grand_mean))
}

norm_diff = norm(centroids_other - as.matrix(centroids[, 1:2]), "i")
```

What we obtained was the 10×2 matrix `centroids_other`, practically identical to the first two columns of the previously computed matrix `centroids`. Indeed the uniform norm of the difference of the two distinct computation is $2.4910629 \times 10^{-15}$.

2.2

One way to make this analysis more rigorous is to take into account the confusion matrix.



Indeed it is conspicuous that the worst performance are indeed in classifying 1's and 5's. Also the 9's are classified quite poorly. Surprisingly, 8's are classified quite badly too. In particular, it is quite surprising that they are classified even worse than the 9's. We observe that LDA yields the following value for train MSE (mean square error):

```
## [1] 0.1229
```

2.3

The MSE computed in the previous point is a good indicator of how well the model can describe the data. However, this is potentially of little use in making predictions. It is more interesting to assess the performances of the model on novel observations. One way to do so is to estimate the test MSE through a procedure called leave-one-out cross validation ($LOOCV$). With this procedure, as the name suggests, the model is trained leaving out one observation at a time, and using it as test, and providing then an estimator for the test MSE (or missclassification error respectively) repeating this procedure for all observations.

```
lda_cv = lda(digit ~ ., data = pendigits, CV = T)
```

Which yields the following missclassification rate:

mse	missclassification_rate
0.1242	1.026 %

which constitutes an increase of roughly 1.026 with respect to the previous estimate. It is not surprising. A remark...

2.4

2.5 (optional)