Job placement gender gap and prediction 👨‍🎓

Giulio Fabbri

```
job<- read.csv("C:\\Users\\Utente\\OneDrive\\Desktop\\esami fatti\\data analytics project\\Job_Placement_Data.csv")
```

Introduction

The proposed dataset shows 215 applicants who were or were not hired, given 13 variables related to their gender and study activity.

The purposes of this analysis involve three main research questions:

1. How is the phenomenon structured?
2. Is there a gender gap in hiring?
3. Can we create a predictive model for future candidates?

Data Dictionary

gender: Gender of the candidate;

ssc_percentage : Senior secondary exams percentage (10th Grade);

ssc_board : Board of education for ssc exams;

hsc_percentage : Higher secondary exams percentage (12th Grade);

hsc_borad : Board of education for hsc exams;

hsc_subject : Subject of study for hsc;

degree_percentage : Percentage of marks in undergrad degree;

undergrad_degree : Undergrad degree majors;

work_experience : Past work experience ;

emp_test_percentage : Aptitude test percentage;

specialization : Postgrad degree majors - (MBA specialization);

mba_percent : Percentage of marks in MBA degree;

status(TARGET) : Status of placement. Placed / Not Placed.

R-Packages used:

```
library(rpart)
library(rpart.plot)
library(ggplot2)
library(gridExtra)
library(corrplot)
library(rpart)
library(caret)
library(randomForest)
library(huxtable)
require(FactoMineR)
```

1. How is the phenomenon structured?

In this first part we ll try to look at our data and understand their structure and how they behave.

```
job$gender<- as.factor(job$gender)
job$ssc_board<- as.factor(job$ssc_board)
job$hsc_board<- as.factor(job$hsc_board)
job$hsc_subject<- as.factor(job$hsc_subject)
job$undergrad_degree<- as.factor(job$undergrad_degree)
job$work_experience<- as.factor(job$work_experience)
job$specialisation<- as.factor(job$specialisation)
job$status<- as.factor(job$status)

str(job)
```

```
## 'data.frame':    215 obs. of  13 variables:
##  $ gender            : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 1 2 2 2 ...
##  $ ssc_percentage    : num  67 79.3 65 56 85.8 ...
##  $ ssc_board         : Factor w/ 2 levels "Central","Others": 2 1 1 1 1 2 2 1 1 1 ...
##  $ hsc_percentage    : num  91 78.3 68 52 73.6 ...
##  $ hsc_board         : Factor w/ 2 levels "Central","Others": 2 2 1 1 1 2 2 1 1 1 ...
##  $ hsc_subject       : Factor w/ 3 levels "Arts","Commerce",..: 2 3 1 3 2 3 2 3 2 2 ...
##  $ degree_percentage : num  58 77.5 64 52 73.3 ...
##  $ undergrad_degree  : Factor w/ 3 levels "Comm&Mgmt","Others",..: 3 3 1 3 1 3 1 3 1 1 ...
##  $ work_experience   : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1 ...
##  $ emp_test_percentage: num  55 86.5 75 66 96.8 ...
##  $ specialisation    : Factor w/ 2 levels "Mkt&Fin","Mkt&HR": 2 1 1 2 1 1 1 1 1 1 ...
##  $ mba_percent       : num  58.8 66.3 57.8 59.4 55.5 ...
##  $ status            : Factor w/ 2 levels "Not Placed","Placed": 2 2 2 1 2 1 1 2 2 1 ...
```

```
summary(job)
```

```
##  gender  ssc_percentage    ssc_board   hsc_percentage    hsc_board
##  F: 76   Min.   :40.89   Central:116   Min.   :37.00   Central: 84
##  M:139   1st Qu.:60.60   Others : 99   1st Qu.:60.90   Others :131
##          Median :67.00                 Median :65.00
##          Mean   :67.30                 Mean   :66.33
##          3rd Qu.:75.70                 3rd Qu.:73.00
##          Max.   :89.40                 Max.   :97.70
##    hsc_subject  degree_percentage  undergrad_degree work_experience
##  Arts    : 11   Min.   :50.00     Comm&Mgmt:145     No :141
##  Commerce:113   1st Qu.:61.00     Others   : 11     Yes: 74
##  Science : 91   Median :66.00     Sci&Tech : 59
##                 Mean   :66.37
##                 3rd Qu.:72.00
##                 Max.   :91.00
##  emp_test_percentage specialisation  mba_percent           status
##  Min.   :50.0        Mkt&Fin:120    Min.   :51.21   Not Placed: 67
##  1st Qu.:60.0        Mkt&HR : 95    1st Qu.:57.95   Placed    :148
##  Median :71.0                       Median :62.00
##  Mean   :72.1                       Mean   :62.28
##  3rd Qu.:83.5                       3rd Qu.:66.25
##  Max.   :98.0                       Max.   :77.89
```

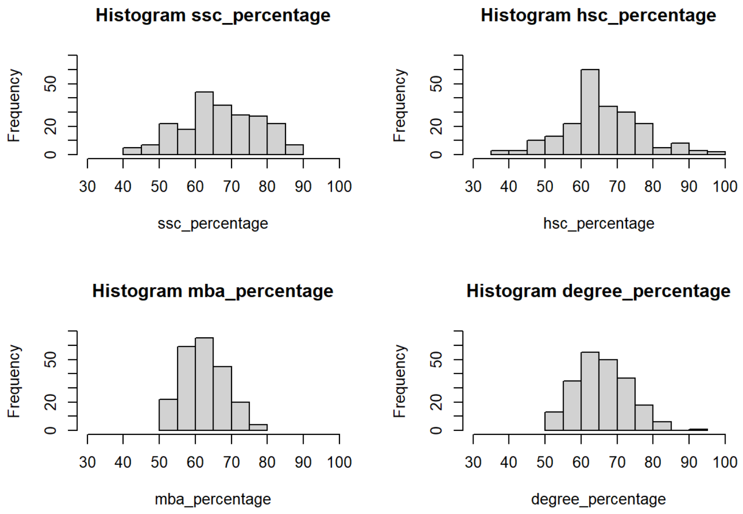
From the summary we can easily notice that there are no unavailable observations (NA), so we can luckily work on a complete dataset.

The first difficulty met on the analysis process is discover that the dataset sample has a different number of males and female on it. The difference is consistent (76 females and 139 males) but still acceptable for making our analysis. As we will explain better below, we are probably talking about observations made in India, Pakistan or Bangladesh, so we need to remind that this difference in our dataset is due to the fact that women in those countries are still less scholarized than men.

1.1 Academic Scores Distribution

Let s now have a look at the distribution of the marks in all the tests that our dataset takes in consideration.

```
par(mfrow=c(2,2))

hist_ssc<-hist(job$ssc_percentage, xlab = 'ssc_percentage', main = 'Histogram ssc_percentage',xlim = c(30,1
00), ylim= c(0,70))

hist_hsc<-hist(job$hsc_percentage, xlab = 'hsc_percentage', main = 'Histogram hsc_percentage',xlim = c(30,1
00), ylim= c(0,70))

hist_mba<-hist(job$mba_percent,  xlab = 'mba_percentage', main = 'Histogram mba_percentage', xlim = c(30,10
0), ylim= c(0,70))

hist_degree<-hist(job$degree_percentage,  xlab = 'degree_percentage', main = 'Histogram degree_percentage',
xlim = c(30,100), ylim= c(0,70))
```



The histograms show how all scores follow a Gaussian distribution. The maximum frequencies are between 60-70% in each case, and the distributions tend to be positively skewed, as we expected from academic scores. However, we can note some peculiarities:

-From the second histogram we can see that hsc_percentage has a similar mean as ssc_percentage, but with a higher frequency and a wider values interval. Regarding this case, it is worth investigating the presence of outliers, since some values seem to be detached from the distribution.

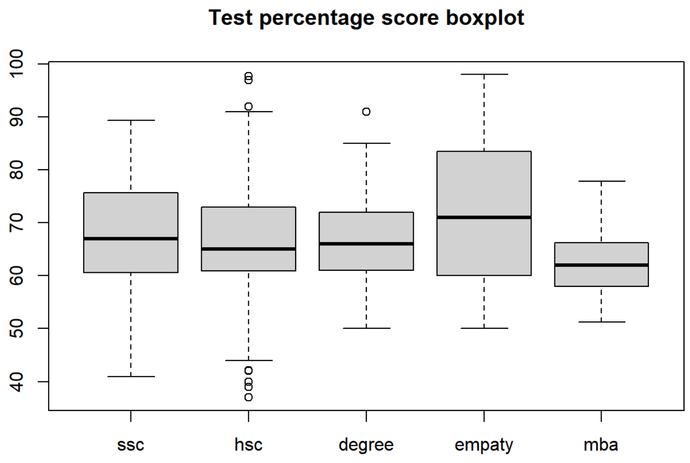
-Mba_percentage distribution is narrower and tends toward lower instead of higher grades.

-there could be an outlier as top score in the degree_percentage variable.

To spot the presence of any outliers in the data we use boxplots of the variables we are interested in.

```
job[job$degree_percentage==max(job$degree_percentage),]
```

```
numerical<- c( "ssc_percentage", "hsc_percentage", "degree_percentage", "emp_test_percentage", "mba_percen
t"  )
boxplot(job[,numerical], main ='Test percentage score boxplot', names = c("ssc", "hsc", "degree", "empaty",
"mba" ))
```



From here we can see that in hsc_percentage there are several extreme values, for both lower and higher scores. However, We can easily understand that the higher scores are acceptable results because it could be reliable that someone had a better mark, and the percentage doesn t exceed 100%.

Instead, we should try to understand better the lower results in hsc because they are under the 50%. The dataset description doesn t tell us anything about which country we are talking about, but we can make the hypotesis that the data are from India, Pakistan or Bangladesh, given the fact that these type of tests are done in those countries and that there the minimum score for passing the test is 35%.

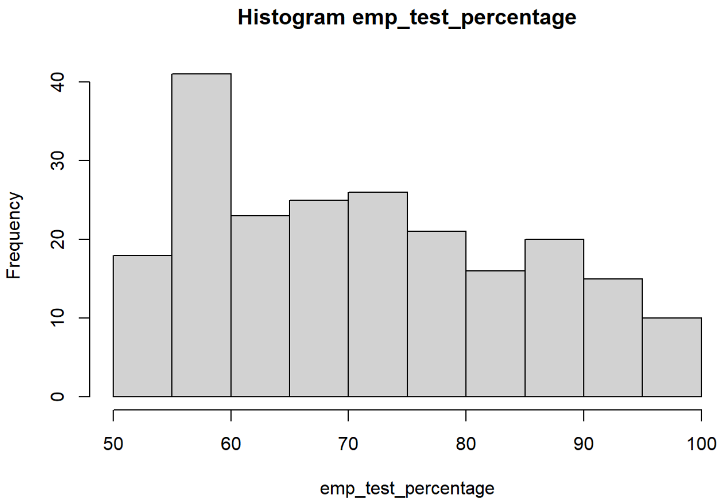
```
#check that all the scores are above 35%
min(boxplot.stats(job$hsc_percentage)$out)
```

```
## [1] 37
```

Since all scores are above 35% our hypothesis is consistent and we can consider all data as reliable.

Now, we still need to check the distribution of the emp test.

```
hist(job$emp_test_percentage,  xlab = 'emp_test_percentage', main = 'Histogram emp_test_percentage')
```

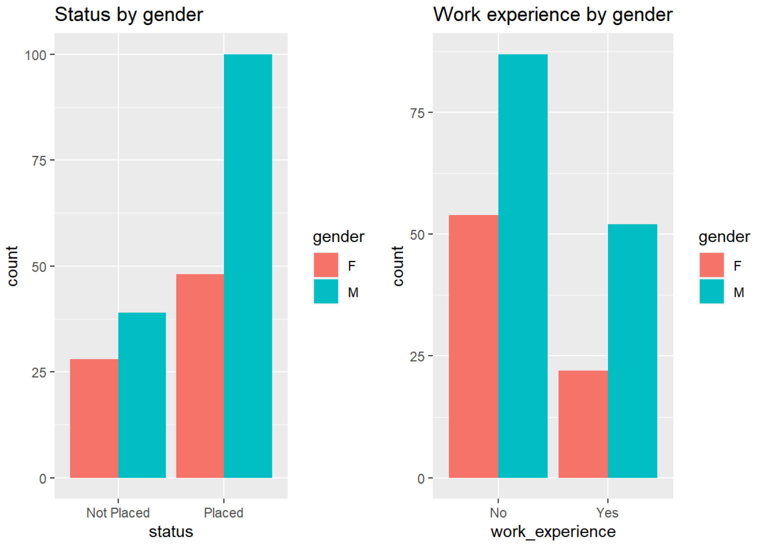


For the first time, we can notice that this test doesn t have a Gaussian distribution as all the others.

1.2 Gender Distribution

We should also check the distribution of males and females between status, work experience, boards and subjects.

```
colnames(job) <- make.unique(names(job))

plot_work_exp<-ggplot(job, aes(x=work_experience, fill=gender))+
  geom_bar(position = "dodge")+
  labs(title='Work experience by gender')

plot_status<-ggplot(job, aes(x=status, fill=gender)) +
  geom_bar(position = "dodge")+
  labs(title='Status by gender')

plot_job_gender = grid.arrange(plot_status,plot_work_exp, ncol=2)
```



As mentioned earlier, there is a different number of males and females in the dataset. The bar graphs therefore do not explain the relative frequencies by gender, but these can be well explained by relative frequency tables

```
#table(job$gender,job$status)

rownames = c("F", "M")
colnames = c("Not Placed", "Placed")

freq<-c((28/76)*100, (48/76)*100, (39/139)*100, (100/139)*100)
round_freq<-round(freq,0)

x<-paste(round_freq,rep("%",4),sep = "")
N <- matrix( x , nrow = 2 , byrow = TRUE, dimnames = list(rownames, colnames))

print(N, quote = FALSE)
```

```
##   Not Placed Placed
## F 37%        63%
## M 28%        72%
```

From this contingency table (in relative values) and the histogram, we can see that, in percentage, more man are placed, but we still don't know anything about their scholastic background, so we can't make any assumption.

We should also check the percentage of males and females that had a previous working experience.

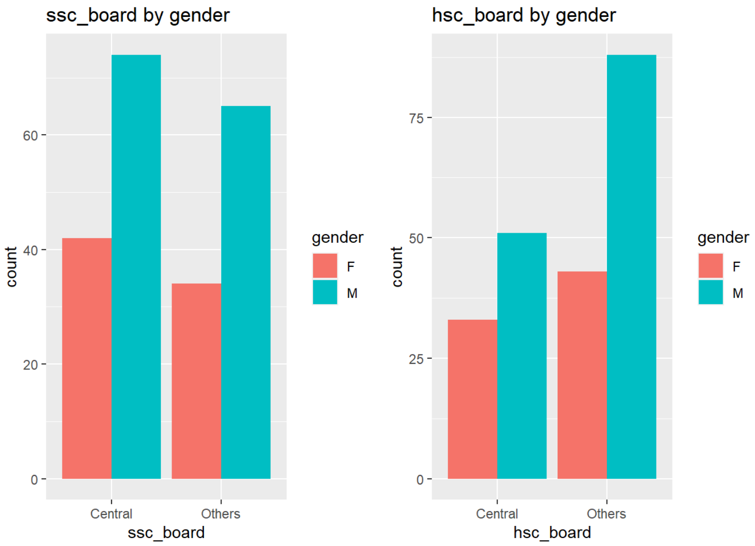
```
#table(job$gender,job$work_experience)


rownames = c("F", "M")
colnames = c("No", "Yes")

freq2=c((54/76)*100, (22/76)*100, (87/139)*100, (52/139)*100)
freq2=round(freq2,0)
x<-paste(freq2,rep("%",4),sep ="" )

N <- matrix(x, nrow = 2, byrow = TRUE, dimnames = list(rownames, colnames))
print(N, quote = FALSE)
```

```
##   No  Yes
## F 71% 29%
## M 63% 37%
```

Both males and females have mostly had no work experience, especially the latter. Given the fact that from literature we know that work experience is a driver for getting placed in a job we should consider this variable (in the regression for analyzing gender gap we should keep this factor constant).

We should look at the boards now, and how gender are distributed inside them.

```
boardssc<-ggplot(job, aes(x=ssc_board, fill=gender)) + geom_bar(position = "dodge")+labs(title='ssc_board b
y gender')
boardhsc<-ggplot(job, aes(x=hsc_board, fill=gender)) + geom_bar(position = "dodge")+labs(title='hsc_board b
y gender')

plot_board_gender = grid.arrange(boardssc,boardhsc, ncol=2)
```



```
#table(job$gender,job$ssc_board)

rownames = c("F", "M")
colnames = c("Central ssc", "Others ssc")
N <- matrix(c(42/76, 34/76, 74/139, 65/139), nrow = 2, byrow = TRUE, dimnames = list(rownames, colnames))
print(N)
```

```
##   Central ssc Others ssc
## F   0.5526316  0.4473684
## M   0.5323741  0.4676259
```

```
#table(job$gender,job$hsc_board)

rownames = c("F", "M")
colnames = c("Central hsc", "Others hsc")
N <- matrix(c(33/76, 43/76, 51/139, 88/139), nrow = 2, byrow = TRUE, dimnames = list(rownames, colnames))
print(N)
```

```
##   Central hsc Others hsc
## F   0.4342105  0.5657895
## M   0.3669065  0.6330935
```

From here we can see that Central boards are more chosen for the ssc, while for hsc are not. Regarding males and females we see that, in percentage, there s not a big difference.

We could make the hypothesis that the board in not relevant for our analysis, so we should decide if drop the variable or not.

In order to understand that, we perform a Chi-squared test to see if the two variables are independent or not.

```
#ssc_board and hsc_board are higly correlated variable(low pvalue)
#so we can just use one of them

chisq.test(job$ssc_board,job$hsc_board)
```

```
## 
##  Pearson's Chi-squared test with Yates' continuity correction
## 
## data:  job$ssc_board and job$hsc_board
## X-squared = 76.454, df = 1, p-value < 2.2e-16
```

From this test we can see that ssc board and hsc board are highly correlated because the p-value is low. We now proceed to another Chi_squared test to asses the dependency of the boards with the status variable.

```
#ssc_board e job status completely idependent(hig pvalue)
#so if placement is target variable we can discard both ssc and hsc board

chisq.test(job$ssc_board,job$status)
```

```
## 
##  Pearson's Chi-squared test with Yates' continuity correction
## 
## data:  job$ssc_board and job$status
## X-squared = 0.15933, df = 1, p-value = 0.6898
```

P-value is high, and we can therefore confidently say that the board and status variables are independent, and so to simplify the analysis we drop the ssc_board and hsc_board variables.

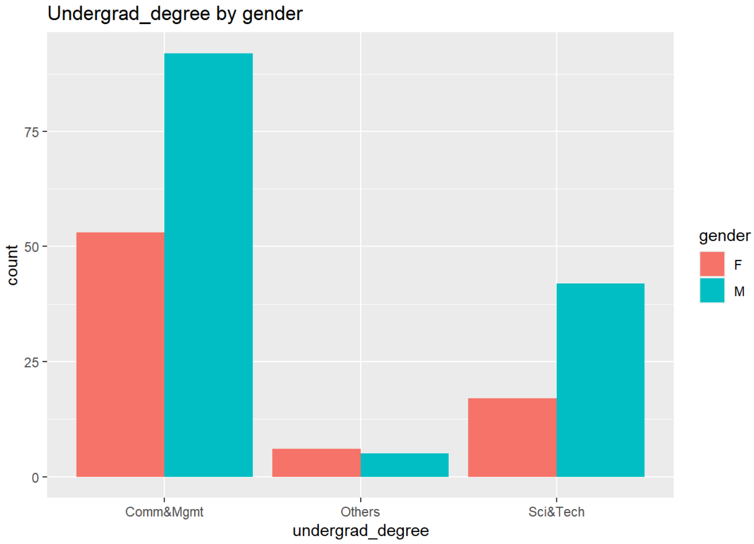
```
job2<- job[,-c(3, 5)]
head(job2)
```

gender	ssc_percentage	hsc_percentage	hsc_subject	degree_percentage	undergrad_degree	work_experience	emp_test_
M	67	91	Commerce	58	Sci&Tech	No	
M	79.3	78.3	Science	77.5	Sci&Tech	Yes	
M	65	68	Arts	64	Comm&Mgmt	No	
M	56	52	Science	52	Sci&Tech	No	
M	85.8	73.6	Commerce	73.3	Comm&Mgmt	No	
M	55	49.8	Science	67.2	Sci&Tech	Yes	

This dataset (job2) will be our referring dataset from now on.

We should now pass at analyzing the gender distribution between the subjects.

```
ggplot(job2, aes(x=undergrad_degree, fill=gender)) + geom_bar(position = "dodge")+labs(title= 'Undergrad_de
gree by gender')
```



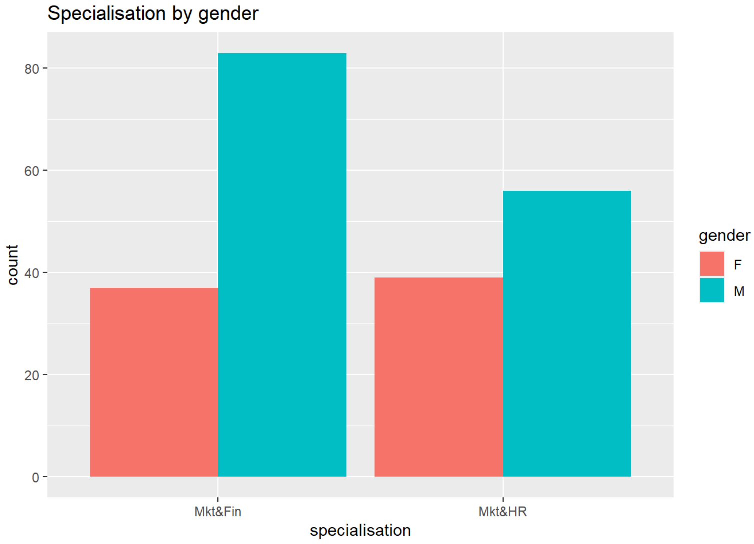
We can see that for both males and females “Communication and management” is the most frequent choice and only in “Others” there seem to be more females.

```
#table(job$gender,job$undergrad_degree)

rownames = c("F", "M")
colnames = c("comm&Mgmt", "Others", 'Sci&tech')
N <- matrix(c(53/76, 6/76, 17/76, 92/139, 5/139, 42/139), nrow = 2, byrow = TRUE, dimnames = list(rownames,
colnames))
print(N)
```

```
##   comm&Mgmt     Others  Sci&tech
## F 0.6973684 0.07894737 0.2236842
## M 0.6618705 0.03597122 0.3021583
```

From this table we can see that, in percentage, more females chose “Communication and management” and “Others”, while males are more oriented to “Science and technologies”. Given the fact that from literature we know that study field is a driver for getting placed in a job we should consider this variable (in the regression for analyzing gender gap we should keep this factor constant).

```
ggplot(job2, aes(x=specialisation, fill=gender)) + geom_bar(position = "dodge")+labs(title= 'Specialisation
by gender')
```



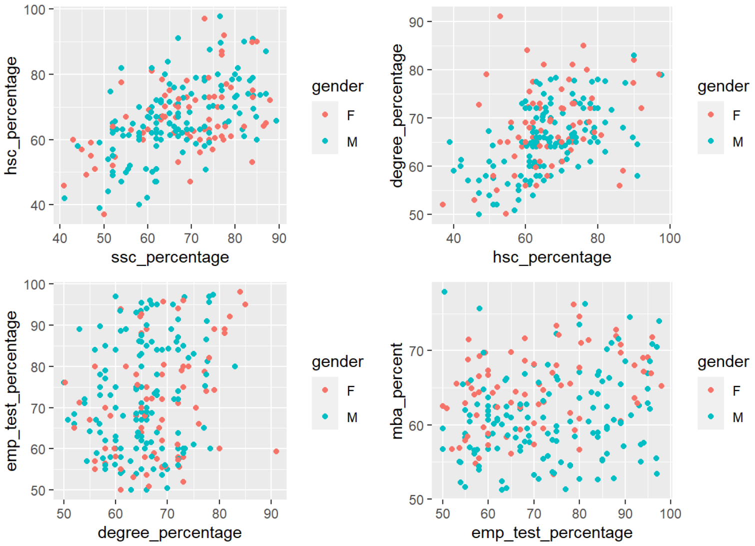
In this histogram we see that females are divided almost equally between the two specializations, while males preferred the “Market and finance” one.

1.3 Correlation study

In this section, we want to check if there could be a correlation between the marks of all the tests.

We chose to look at the correlation between a mark and its successor, because looking at every combination would be computational expensive, and logically it seemed to be the best option.

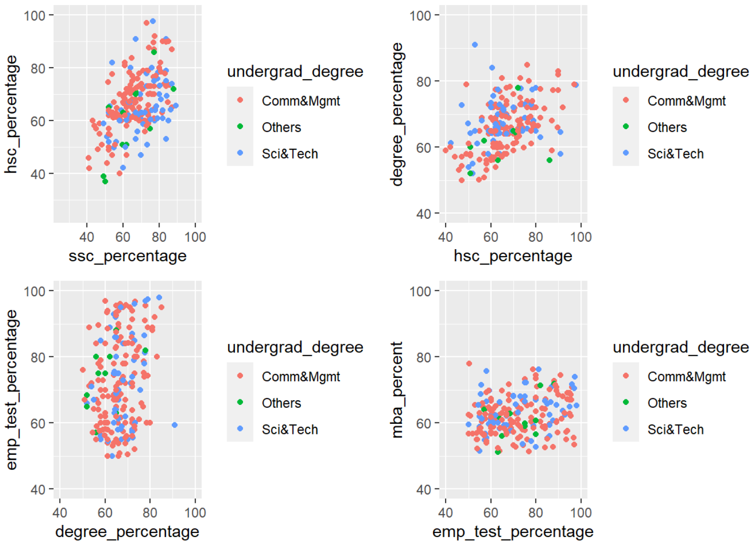
```
a<-qplot(ssc_percentage,hsc_percentage, data = job2, color=gender)
b<-qplot(hsc_percentage, degree_percentage, data = job2, color=gender)
c<-qplot(degree_percentage,emp_test_percentage, data = job2, color=gender)
d<-qplot(emp_test_percentage, mba_percent, data = job2, color=gender)

plot_mark = grid.arrange(a,b,c,d, ncol=2, nrow=2)
```



With those scatterplots we can t assume almost any correlation between the results of the various tests, maybe we can examine a slight one between hsc and degree marks.

Looking at males and females distribution we can t make any preliminary assumption either, with the exception that females tend to have higher results than males in mba test respect to emp test.

```
e<-qplot(ssc_percentage,hsc_percentage, data = job2, color=undergrad_degree,
         xlim = c(25, 100), ylim= c(25,100))

f<-qplot(hsc_percentage,degree_percentage, data = job2, color=undergrad_degree,
         xlim = c(40, 100), ylim= c(40,100))
g<-qplot(degree_percentage,emp_test_percentage, data = job2, color=undergrad_degree,
          xlim = c(40, 100), ylim= c(40,100))
h<-qplot(emp_test_percentage,mba_percent, data = job2, color=undergrad_degree,
          xlim = c(40, 100), ylim= c(40,100))

plot_mark = grid.arrange(e,f,g,h, ncol=2, nrow=2)
```



From those scatterplots we can t assume anything about the choice of the subject based on the mark distribution.

Anyway, we can see that the outcome the academic performance of test seems not correlated to emp test scores. This is also proved by the low Pearson correlation metric:

```
attach(job2)

c1<-round(cor(emp_test_percentage,ssc_percentage),3)
c2<-round(cor(emp_test_percentage,hsc_percentage),3)
c3<-round(cor(emp_test_percentage,degree_percentage),3)
c4<-round(cor(emp_test_percentage, mba_percent),3)

tab <- matrix(c(c1,c2,c3,c4), ncol=1, byrow=TRUE)
colnames(tab) <- c("emp_test")
rownames(tab) <- c("ssc_percentage:","hsc percentage:","degree_percentage:","mba_percentage:")
tab <- as.table(tab)
tab
```

```
##                     emp_test
## ssc_percentage:        0.262
## hsc percentage:        0.245
## degree_percentage:     0.224
## mba_percentage:        0.218
```

Just to be sure, we look at the monotonic correlation with the Kendall method;

```
c1<-round(cor(emp_test_percentage,ssc_percentage, method= 'kendall'),3)
c2<-round(cor(emp_test_percentage,hsc_percentage, method= 'kendall'),3)
c3<-round(cor(emp_test_percentage,degree_percentage, method= 'kendall'),3)
c4<-round(cor(emp_test_percentage, mba_percent, method= 'kendall'),3)

tab <- matrix(c(c1,c2,c3,c4), ncol=1, byrow=TRUE)
colnames(tab) <- c("emp_test")
rownames(tab) <- c("ssc_percentage:","hsc percentage:","degree_percentage:","mba_percentage:")
tab <- as.table(tab)
tab
```

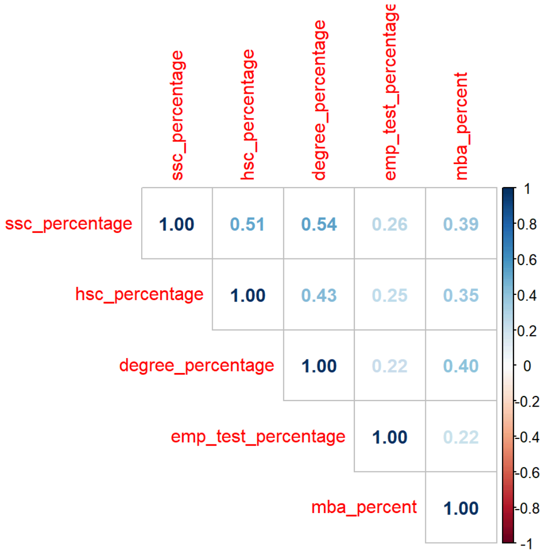
```
##                     emp_test
## ssc_percentage:        0.183
## hsc percentage:        0.146
## degree_percentage:     0.135
## mba_percentage:        0.145
```

In both the cases the correlation is confirmed to be really low.

Let s do a correlation matrix for the academic test scores, only with Pearson method, since it is the one that depicts higher correlations.

```
#take only the test scores varialbes
job_percentages<-job[,-c(1,3,5,6,8,9,11,13,15)]
```

```
c<-cor(job_percentages)
corrplot(c, method = "number",type = "upper")
```

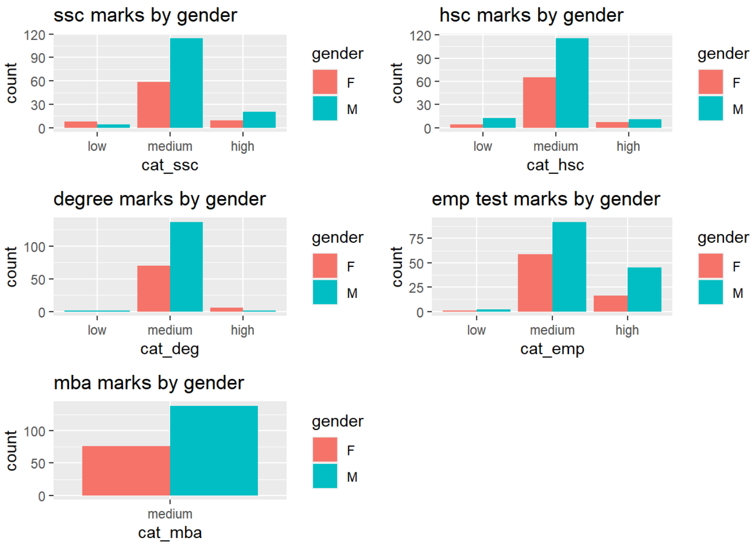


We can see that the correlations among test points are all positives, and not so strong, we should so keep all the test scores for our models cause the correlation between them is not so high so every test could add something to the models.

Now let s look at the scores divided by gender. For doing this we split the marks in low, medium and high and we look at the distribution between males and females.

```
cat_ssc<-cut(job2$ssc_percentage, breaks = c(0,50,80,100),labels = c("low", "medium", "high"))
cat_hsc<-cut(job2$hsc_percentage, breaks = c(0,50,80,100),labels = c("low", "medium", "high"))
cat_deg<-cut(job2$degree_percentage, breaks = c(0,50,80,100),labels = c("low", "medium", "high"))
cat_emp<-cut(job2$emp_test_percentage, breaks = c(0,50,80,100),labels = c("low", "medium", "high"))
cat_mba<-cut(job2$mba_percent, breaks = c(0,50,80,100),labels = c("low", "medium", "high"))

plot_ssc<-ggplot(job2, aes(cat_ssc, fill=gender)) + geom_bar(position = "dodge")+labs(title = 'ssc marks by
gender')
plot_hsc<-ggplot(job2, aes(cat_hsc, fill=gender)) + geom_bar(position = "dodge")+labs(title = 'hsc marks by
gender')
plot_deg<-ggplot(job2, aes(cat_deg, fill=gender)) + geom_bar(position = "dodge")+labs(title = 'degree marks
by gender')
plot_emp<-ggplot(job2, aes(cat_emp, fill=gender)) + geom_bar(position = "dodge")+labs(title = 'emp test mar
ks by gender')
plot_mba<-ggplot(job2, aes(cat_mba, fill=gender)) + geom_bar(position = "dodge")+labs(title = 'mba marks by
gender')
```

```
plot_scores_gender<-grid.arrange(plot_ssc,plot_hsc,plot_deg,plot_emp,plot_mba, ncol=2, nrow=3)
```



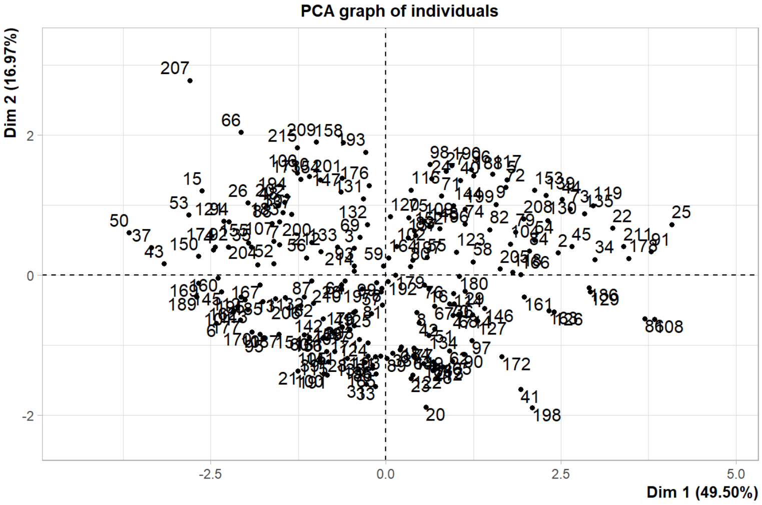
From these graphs we could deduct that in ssc females tend to have lower marks, while in the degree they have way higher marks than males. If we look at the emp test we see that no girls get a low mark, while in mba everyone has a medium mark.

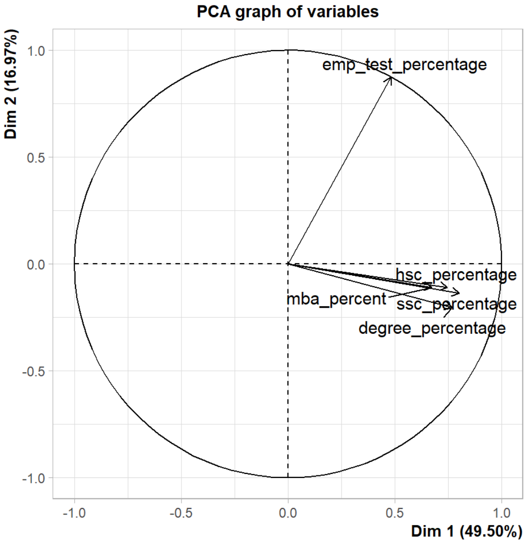
1.4 PCA

Since we finished to plot our attributes singularly or in couples, in this part of our analysis, we would like to visualize our entire dataset.

In order to do that, we need to reduce the number of attributes in our dataset, and plot them preserving the dataset structure. For obtaining this result we can try to apply a PCA.

```
job.pca<- PCA(job2[numerical])
```



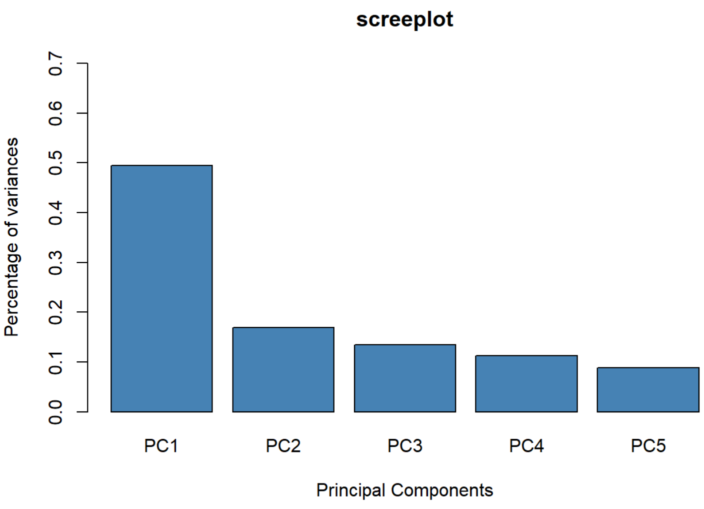


In the first graph, we can see the distribution of all our observations along the firsts two principal components, while in the second we look at the correlation between each numerical attribute and the two components.

```
summary(job.pca)
```

```
##
## Call:
## PCA(X = job2[numerical])
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5
## Variance               2.475   0.849   0.675   0.561   0.441
## % of var.             49.504  16.971  13.496  11.217   8.812
## Cumulative % of var.  49.504  66.476  79.972  91.188 100.000
##
## Individuals (the 10 first)
##                         Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2  
## 1                    |  2.911 | -0.147  0.004  0.003 | -1.165  0.744  0.160 |
## 2                    |  2.530 |  2.458  1.135  0.943 |  0.318  0.056  0.016 |
## 3                    |  0.902 | -0.455  0.039  0.255 |  0.385  0.081  0.182 |
## 4                    |  2.668 | -2.465  1.142  0.854 |  0.366  0.074  0.019 |
## 5                    |  3.017 |  1.726  0.560  0.327 |  1.366  1.023  0.205 |
## 6                    |  2.945 | -2.424  1.104  0.678 | -0.683  0.256  0.054 |
## 7                    |  3.427 | -1.521  0.435  0.197 |  0.439  0.106  0.016 |
## 8                    |  1.431 |  0.439  0.036  0.094 | -0.529  0.153  0.136 |
## 9                    |  2.089 |  1.569  0.463  0.565 |  1.013  0.562  0.235 |
## 10                   |  2.500 | -1.794  0.605  0.515 | -0.840  0.386  0.113 |
##                       Dim.3    ctr   cos2  
## 1                    -1.307  1.178  0.202 |
## 2                    -0.278  0.053  0.012 |
## 3                    -0.644  0.286  0.510 |
## 4                     0.551  0.209  0.043 |
## 5                    -1.844  2.345  0.374 |
## 6                    -0.687  0.326  0.055 |
## 7                    -0.311  0.067  0.008 |
## 8                    -0.315  0.069  0.049 |
## 9                    -0.823  0.466  0.155 |
## 10                   -1.332  1.222  0.284 |
##
## Variables
##                        Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3
## ssc_percentage      |  0.802 25.981  0.643 | -0.137  2.214  0.019 | -0.232
## hsc_percentage      |  0.746 22.491  0.557 | -0.109  1.392  0.012 | -0.323
## degree_percentage   |  0.769 23.887  0.591 | -0.204  4.909  0.042 | -0.068
## emp_test_percentage |  0.483  9.412  0.233 |  0.874 90.055  0.764 | -0.002
## mba_percent         |  0.672 18.228  0.451 | -0.110  1.430  0.012 |  0.715
##                        ctr   cos2  
## ssc_percentage       7.982  0.054 |
## hsc_percentage      15.472  0.104 |
## degree_percentage    0.685  0.005 |
## emp_test_percentage  0.001  0.000 |
## mba_percent         75.860  0.512 |
```

```
var <- c(0.495, 0.1697, 0.1350, 0.1122, 0.08812)
names.arg <- c("PC1", "PC2", "PC3", "PC4", "PC5")
barplot(var,
        names.arg = names.arg,
        ylim = c(0, 0.7),
        main = "screeplot",
        xlab = "Principal Components",
        ylab = "Percentage of variances",
        col = "steelblue")
```



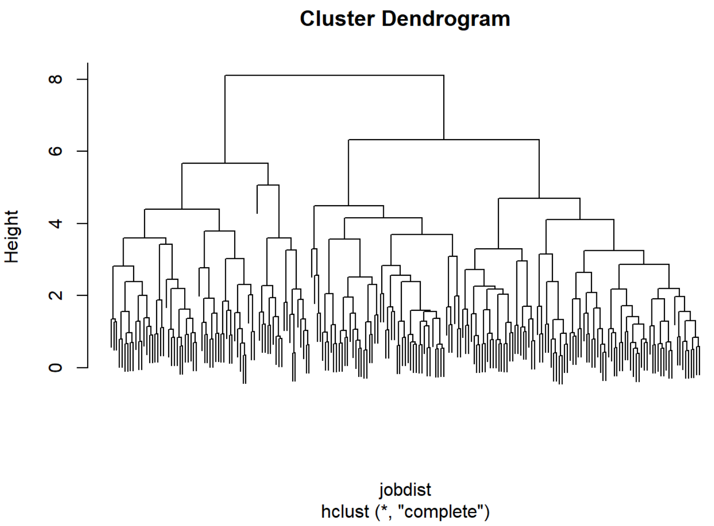
As we can see by the cumulative proportion, the PCA is not really useful in our dataset because we can reduce the dimension only arriving at 4 dimensions, if we want to keep a discrete significativity, but this doesn t help us in plotting the data, for this reason, we decided not to use this model.

1.4 Clustering

At this point, we could try to divide our dataset in clusters, and, since the number of observations is not really high, we can try to use the hierarchical clustering method in order to find the appropriate number of clusters.

Basing on the distribution of our observations we could chose even the k-means method, but, since the hierarchical clustering is simpler and easy to understand thanks to the visual representation, we preferred this technique.

```
jobscaled<- scale(job2[numerical])
jobdist<- dist(jobscaled)

job.hc = hclust(jobdist)

plot(job.hc, labels=FALSE)
```



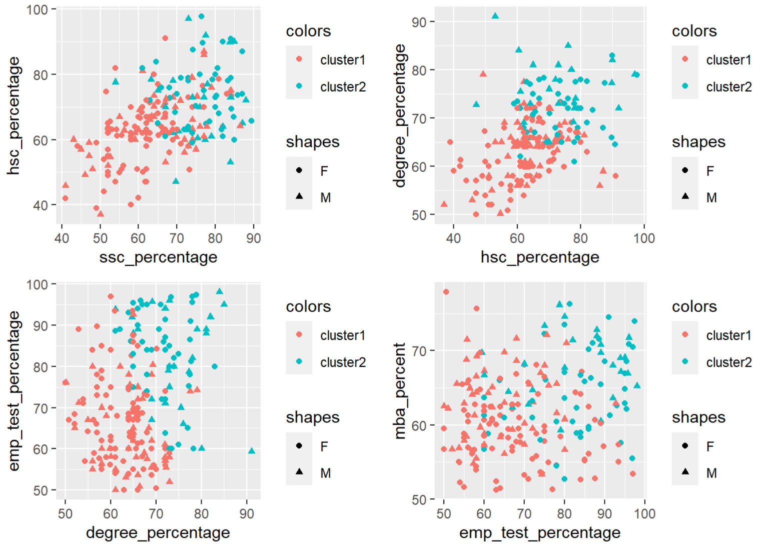
From the dendrogram we can see that we can create two clusters.

```
job.hc.2 = cutree(job.hc, 2)

job.hc.2
```

```
##   [1] 1 2 1 1 2 1 1 1 2 1 1 2 1 1 1 2 1 1 1 1 1 2 1 2 2 1 2 1 1 1 1 1 1 2 1 2 1
##  [38] 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 2 1 1 1 2 1 2 2 2 2
##  [75] 1 1 1 1 2 1 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1 2 2 2 1 1 1 2 1 2 1 1 1 2 2 1 1
## [112] 1 1 1 1 2 2 2 2 1 1 1 1 2 1 2 2 1 2 2 1 1 1 2 2 1 1 1 2 1 1 1 2 2 1 2 1 2
## [149] 1 1 1 2 2 1 1 1 2 1 1 1 2 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1
## [186] 2 1 2 1 2 1 2 1 1 1 2 2 2 2 1 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1
```

```
job2$gender<- as.factor(job2$gender)
job2$hsc_subject<- as.factor(job2$hsc_subject)
job2$undergrad_degree<- as.factor(job2$undergrad_degree)
job2$work_experience<- as.factor(job2$work_experience)
job2$specialisation<- as.factor(job2$specialisation)
job2$status<- as.factor(job2$status)
```

```
myshapes = c("M", "F")
mycolors= c('cluster1', 'cluster2')
colors<-mycolors[job.hc.2]
shapes<-myshapes[as.integer(gender)]

e<-qplot(ssc_percentage,hsc_percentage, data = job2, color = colors,pch=shapes)

f<-qplot(hsc_percentage,degree_percentage, data = job2, color=colors, pch=shapes)

g<-qplot(degree_percentage,emp_test_percentage, data = job2, color=colors,pch=shapes)

h<-qplot(emp_test_percentage,mba_percent, data = job2, color=colors,pch=shapes)

plot_mark1 = grid.arrange(e,f,g, h)
```



Looking at those scatterplots, it seems that the two clusters have been divided by the marks.

We decided to aggregate them by status and see if this table could suggest us something.

```
cluster_means<-aggregate(job2, by=list(job.hc.2, job2$status), FUN = mean)
#delete on numeric columns that are not computabule
cm<-cluster_means[,-c(3,6,8,9,11,13)]
cm
```

Group.1	Group.2	ssc_percentage	hsc_percentage	degree_percentage	emp_test_percentage	mba_percent
1	Not Placed	56.9	57.5	60.3	67.9	61.1
2	Not Placed	64.6	67.1	69.3	86.4	67
1	Placed	67.4	66.9	65.3	65.6	60.3
2	Placed	76.9	73.6	72.9	82.4	65.3

From here we can see that in both the clusters not placed people have a mark average lower than placed one.

Now, we aggregate the clusters by gender:

```
cluster_gender_mean<-aggregate(job2, by=list(job.hc.2, job2$gender), FUN=mean)
cgm<-cluster_gender_mean[,-c(3,6,8,9,11,13)]
cgm
```

Group.1	Group.2	ssc_percentage	hsc_percentage	degree_percentage	emp_test_percentage	mba_percent
1	F	64.6	64.1	64.6	64.5	62.9
2	F	75.4	71.5	74.7	82.3	68
1	M	61.9	62.2	62.4	67.8	59.4
2	M	76.2	73.9	71.4	83	64.1

From this table we can see that in ssc and hsc females tent to be closest to the mean because in cluster one they have higher average than males, while in cluster two they have lower average. Instead, in degree and mba they always have a higher average than males. In the emp test we see that in cluster 1 females have lower marks, while in cluster two they are almost the same.

We want even to look at placed and not placed males and females in both the clusters.

```
head(job2)
```

gender	ssc_percentage	hsc_percentage	hsc_subject	degree_percentage	undergrad_degree	work_experience	emp_test_
M	67	91	Commerce	58	Sci&Tech	No	
M	79.3	78.3	Science	77.5	Sci&Tech	Yes	
M	65	68	Arts	64	Comm&Mgmt	No	
M	56	52	Science	52	Sci&Tech	No	

M	85.8	73.6	Commerce	73.3	Comm&Mgmt	No
M	55	49.8	Science	67.2	Sci&Tech	Yes

```
job_hc<- cbind(job2, job.hc.2)
head(job2)
```

gender	ssc_percentage	hsc_percentage	hsc_subject	degree_percentage	undergrad_degree	work_experience	emp_test_
M	67	91	Commerce	58	Sci&Tech	No	
M	79.3	78.3	Science	77.5	Sci&Tech	Yes	
M	65	68	Arts	64	Comm&Mgmt	No	
M	56	52	Science	52	Sci&Tech	No	
M	85.8	73.6	Commerce	73.3	Comm&Mgmt	No	
M	55	49.8	Science	67.2	Sci&Tech	Yes	

In cluster 1, that is the one with lower marks, we can see this:

```
job_lowscores<- job_hc[job_hc$job.hc.2==1,]
table(job_lowscores$gender,job_lowscores$status)
```

```
##
##     Not Placed Placed
##   F         23     27
##   M         38     54
```

In cluster 2, that is the one with higher marks, this is the result instead:

```
job_highscores<- job_hc[job_hc$job.hc.2==2,]
table(job_highscores$gender,job_highscores$status)
```

```
##
##     Not Placed Placed
##   F          5     21
##   M          1     46
```

From those tables we can see that people with higher marks are almost always employed.

2. Is there a gender gap in hiring?

2.1 The Logistic Regression

For answering this question we implemented a logistic regression.

```
job2$status<- factor(job2$status)
logistic<-glm(status~., data = job2, family = "binomial")
huxreg("logistic model"= logistic)
```

	logistic model
(Intercept)	-18.185 ***
	(5.308)
genderM	1.334 *
	(0.665)
ssc_percentage	0.226 ***
	(0.043)
hsc_percentage	0.102 **
	(0.036)
hsc_subjectCommerce	-1.492
	(1.339)
hsc_subjectScience	-0.870

	(1.408)
degree_percentage	0.185 ***
	(0.056)
undergrad_degreeOthers	-1.111
	(1.466)
undergrad_degreeSci&Tech	-1.702 *
	(0.775)
work_experienceYes	2.041 **
	(0.701)
emp_test_percentage	-0.015
	(0.022)
specialisationMkt&HR	-0.277
	(0.547)
mba_percent	-0.204 ***
	(0.056)
N	215
logLik	-50.229
AIC	126.457

*** p < 0.001; ** p < 0.01; * p < 0.05.

In this first model we discover a statistically significant gender gap (95% confidence level). According to this model being a male on average leads to an increase of 12.5% (log(1.334)) in the probability of get the job placement. However, we note that the value of beta and confidence for the gender variable is the lowest of all significant factors, so further analysis are required to prove that we are really in front of a gender gap in placements.

2.2 Evaluation of the Logistic Regression

The Logistic regression cannot be evaluated by the R squared measure so we must find another way. The most straightforward way to evaluate what s the best logistic regression model to use for our analysis is the AIC (Akaike Information Criterion) that measures the residual deviance adjusted for the number of parameters. The absolute value is not important, the key is its variation: a model with a lower AIC is indeed better. Variations of two units are already a good result. For the first model with all the variables is 126.5, let s delete the not significant variable to see if we can improve this measure.

```
#second model without the emp_test,specialization,undergrad e hsc subjects
logistic2 <-glm(status~. -emp_test_percentage -specialisation- undergrad_degree -hsc_subject,data= job2, fa
mily = "binomial")
#third model deleting also the gender
logistic3 <-glm(status~. -gender -emp_test_percentage -specialisation- undergrad_degree -hsc_subject,data=
job2, family = "binomial")

huxreg("model1"=logistic ,"model2"= logistic2,"model3"=logistic3, statistics = "AIC")
```

	model1	model2	model3
(Intercept)	-18.185 ***	-19.320 ***	-15.489 ***
	(5.308)	(4.697)	(3.961)
genderM	1.334 *	1.087	
	(0.665)	(0.578)	
ssc_percentage	0.226 ***	0.205 ***	0.192 ***
	(0.043)	(0.041)	(0.038)
hsc_percentage	0.102 **	0.118 ***	0.119 ***
	(0.036)	(0.035)	(0.035)
hsc_subjectCommerce	-1.492		
	(1.339)		
hsc_subjectScience	-0.870		
	(1.408)		
degree_percentage	0.185 ***	0.164 **	0.150 **

	(0.056)	(0.050)	(0.048)
undergrad_degreeOthers	-1.111		
	(1.466)		
undergrad_degreeSci&Tech	-1.702 *		
	(0.775)		
work_experienceYes	2.041 **	2.079 **	2.267 ***
	(0.701)	(0.646)	(0.647)
emp_test_percentage	-0.015		
	(0.022)		
specialisationMkt&HR	-0.277		
	(0.547)		
mba_percent	-0.204 ***	-0.203 ***	-0.228 ***
	(0.056)	(0.051)	(0.050)
AIC	126.457	122.779	124.471

*** p < 0.001; ** p < 0.01; * p < 0.05.

The best model according to the AIC measure is the second one. However, we prefer the first model. We can see how the gender factor that was statistically significant in the first model becomes not statistically significant in the second. This change can be explained by the elimination of factors related to the subjects studied in the hsc, degree and specialization.

Why we choose the first model even if the AIC measure is worse? We know that males are more present in technical subjects that lead to a greater chance of placement, so we cannot discard those factor to not fall in an omitted variable bias. Infact, in order to verify a “direct” gender gap in the likelihood of hiring, we need to verify that a woman is less likely to be hired while holding the other factors(score and subject of degrees) constant. For this reason the first model is better and we can say that there is a well-founded, if slight, suspicion of discrimination in the hiring process, which is also not totally verifiable given the small number of observations and the imbalance between the number of males and females.

3. Can we create a predictive model for future candidates?

3.1 Logistic Prediction

We can use the previous logistic model to predict whether a student has been placed based on his or her gender, qualifications, test scores, and subjects of study.

```
#divide the dataset in train set and test set
nrow(job2)
215*0.8
#setseed to keep the same sample and therefore compare models
set.seed(1)
job.idx = sample(215, 172)

job.train<-job2[job.idx,]
job.test<- job2[-job.idx,]

#Define status as numeric binomial variable with 1 = placed and 0 = not placed
job.train$status <-ifelse(job.train$status=="Placed",1,0)
job.test$status <-ifelse(job.test$status=="Placed",1,0)

as.numeric(job.train$status)
as.numeric(job.test$status)
#train the logistic model
logistic.fit <- glm(status~.  ,
                    data = job.train,
                    family = "binomial")
#test the logistic model
logistic.test<- predict(logistic.fit,
                    newdata = job.test,
                    type = "response")
```

```
#logistic accuracy
logistic.pred <- ifelse(logistic.test > 0.5, "1", "0")

t<-table(logistic.pred,job.test$status)
logistic_accuracy= sum(diag(t)/sum(t))
logistic_accuracy
```

```
## [1] 0.8372093
```

The logistic model does not have such a good prediction so we can see if the decision tree can be more accurate.

3.2 Decision Tree

First at all, for building a decision tree, we need to split our dataset in training and tasting dataset, using the standard percentages of 80% of the observations for the training dataset, and 20% for the testing one.

```
#same traiing and test data
nrow(job2)
```

```
## [1] 215
```

```
215*0.8
```

```
## [1] 172
```

```
set.seed(1)
job.idx = sample(215, 172)

job.train<-job2[job.idx,]
job.test<- job2[-job.idx,]
```

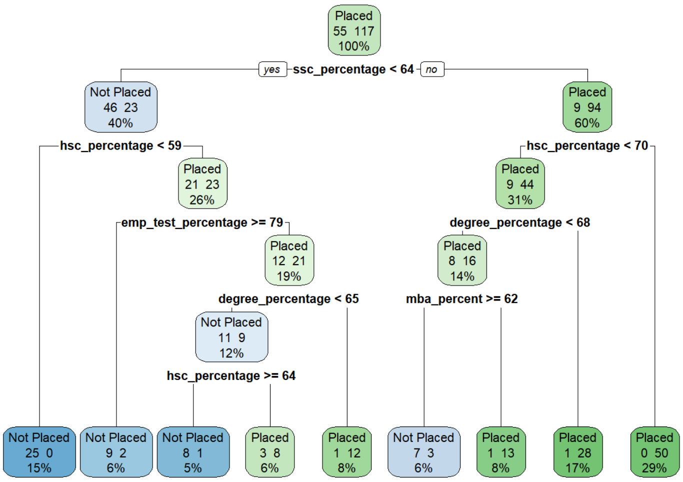
```
summary(job.train)
```

```
##  gender  ssc_percentage  hsc_percentage    hsc_subject degree_percentage
##  F: 59   Min.   :40.89   Min.   :37.00   Arts    : 7   Min.   :50.00    
##  M:113   1st Qu.:60.17   1st Qu.:60.68   Commerce:89   1st Qu.:61.00    
##          Median :66.75   Median :65.00   Science :76   Median :65.80    
##          Mean   :66.97   Mean   :66.55                 Mean   :66.39    
##          3rd Qu.:74.97   3rd Qu.:73.25                 3rd Qu.:72.00    
##          Max.   :89.40   Max.   :97.70                 Max.   :91.00    
##   undergrad_degree work_experience emp_test_percentage specialisation
##  Comm&Mgmt:115     No :108         Min.   :50.00       Mkt&Fin:93    
##  Others   :  8     Yes: 64         1st Qu.:60.00       Mkt&HR :79    
##  Sci&Tech : 49                     Median :70.00                     
##                                    Mean   :71.23                     
##                                    3rd Qu.:80.60                     
##                                    Max.   :97.40                     
##   mba_percent           status   
##  Min.   :51.21   Not Placed: 55  
##  1st Qu.:58.17   Placed    :117  
##  Median :62.08                   
##  Mean   :62.43                   
##  3rd Qu.:66.33                   
##  Max.   :77.89                   
```

Looking at the summary of our training dataset, we see that the structure is not particularly changed from the original dataset, so we can accept this division as representative.

Now, we can look at the decision tree that we can obtain from these data.

```
job.dt.1 = rpart(status ~., data=job.train)
rpart.plot(job.dt.1, extra=101)
```



Our tree seems to be too articulated and a bit confusing, but we should look if, at least, makes a good prediction of our testing dataset.

```
job.predict = predict(job.dt.1, job.test, type='class')
accuracy_table= table(job.test$status, job.predict)
```

```
accuracy= sum(diag(accuracy_table)/sum(accuracy_table))
accuracy
```

```
## [1] 0.6976744
```

```
accuracy_table
```

```
##             job.predict
##              Not Placed Placed
##   Not Placed          8      4
##   Placed              9     22
```

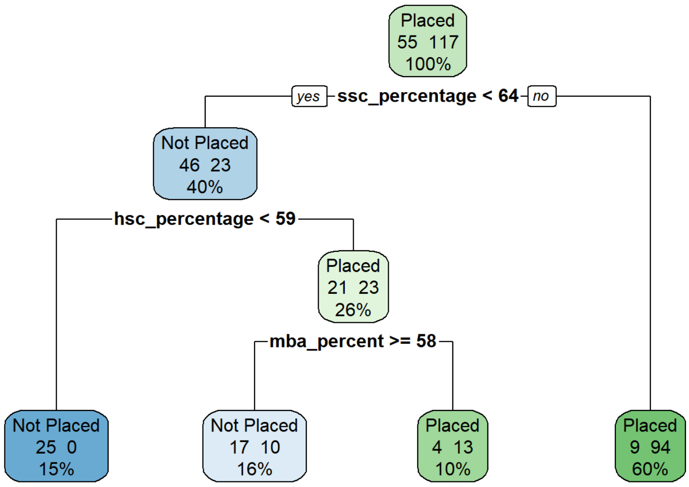
As we can see, the accuracy is not really high, and we can also see the misclassifications in the table, especially we notice that our tree is not really good in predicting who will not be hired.

Improving the decision tree

So, our decision tree does not perform well, but can definitely be improved. Looking at the plot, it is clear that the graph seems too complex and redundant, a simplification could lead to an improvement. In order to simplify the decision tree we can:

1. remove some variables that from previous analyses (EDA and Logistic regression) emerge as not so important like emp_test percentage and hsc_subject
2. reduce the depth of the three to a maximum of tree "layers"

```
#reduce the depth of the tree to max 3 "layers"
control <- rpart.control(cp = 0, maxdepth = 3)
#create the decision three
job.dt.2 <- rpart(status~. -emp_test_percentage -hsc_subject  , data = job.train, method = 'class', control
= control)
rpart.plot(job.dt.2, extra=101)
```



Now the decision tree is more understandable and applies a more precise division. We can clearly see how academic career path affects the likelihood of being placed. The improvement seen in the graph s clarity is certified by the accuracy:

```
#accuracy decision tree improved
job.predict = predict(job.dt.2, job.test, type='class')

prediction_table_impr= table(job.test$status, job.predict)

acc_imp= sum(diag(prediction_table_impr)/sum(prediction_table_impr))
acc_imp
```

```
## [1] 0.8139535
```

```
prediction_table_impr
```

```
##             job.predict
##              Not Placed Placed
##   Not Placed          6      6
##   Placed              2     29
```

Continuing to change the parameters does not improve the decision tree by much, so to make a final leap in the quality of prediction we decided to build a Random Forest.

3.3 Random Forest

```
#same training and test data
nrow(job2)
```

```
## [1] 215
```

```
215*0.8
```

```
## [1] 172
```

```
set.seed(1)
job.idx = sample(215, 172)

job.train<-job2[job.idx,]
job.test<- job2[-job.idx,]
```

```
#build the random forest
as.numeric(job.train$status)
```

```
##   [1] 2 2 2 1 1 2 1 2 2 2 1 1 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 2 2
##  [38] 1 2 1 2 2 1 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2
##  [75] 1 2 2 2 2 2 1 2 2 1 1 1 2 2 2 2 1 1 2 2 2 1 1 2 1 2 1 2 1 1 2 1 1 2 2 2 2
## [112] 2 1 1 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 2 2 2 2 1 1 1 2 2 2
## [149] 2 2 2 2 2 2 1 2 2 2 1 2 1 1 2 1 2 1 1 2 1 2 2 2
```

```
job.rf = randomForest(status ~ ., data=job.train, ntree = 500)
```

```
#random forest accuracy
job.rf.pred = predict(job.rf, job.test)

forest_table=table(job.test$status, job.rf.pred)

acc_forest= sum(diag(forest_table)/sum(forest_table))
acc_forest
```

```
## [1] 0.8604651
```

The random forest at the start performs better in the decision tree and logistic regression. This can be the best model for the placement prediction but there is still room for further improvements.

Improving the random forest

As in the decision tree, we tried to improve the model by moving some parameters. Decreasing the number of variables, however, did not prove as useful as in the previous case. The winning path turned out to be changing the mtry parameter that controls how many of the input features a decision tree has available to consider during the bagging.

```
#check the numbers of variables amd the out of bag error
job.rf
```

```
## 
## Call:
##  randomForest(formula = status ~ ., data = job.train, ntree = 500) 
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
## 
##         OOB estimate of  error rate: 15.7%
## Confusion matrix:
##            Not Placed Placed class.error
## Not Placed         36     19  0.34545455
## Placed              8    109  0.06837607
```

Having 10 variables available, the mtry parameter can vary from 1 to 10. Through a for loop it is then possible to check them all and see which parameter minimizes the out of bag error(the model error on the prediction of data left out of the bootstrap sample that is used to train each single decision tree).

```
oob.values<-vector(length=10)

for(i in 1:10){
  job.rf2 = randomForest(status ~.,
                      data=job.train,
                      mtry= i,
                      ntree= 500)
  oob.values[i]<-job.rf2$err.rate[nrow(job.rf2$err.rate),1]
}
oob.values
```

```
##  [1] 0.1511628 0.1511628 0.1569767 0.1453488 0.1686047 0.1627907 0.1627907
##  [8] 0.1627907 0.1744186 0.1686047
```

The second mtry is optimum so mtry= 2

```
job.rf3 = randomForest(status ~.,
                       data=job.train,
                       mtry= 2,
                       ntree= 500)
```

```
job.rf3.pred = predict(job.rf3, job.test)

forest_table3 = table(job.test$status, job.rf3.pred)
acc_forest3= sum(diag(forest_table3)/sum(forest_table3))
acc_forest3
```

```
## [1] 0.8837209
```

This Random forest has achieved good predictive ability and can therefore be a valuable tool for predicting a student s placement by knowing his test scores, subjects and gender.

Conclusion

In the first part of this study we provided a descriptive analysis of student placement and its relationship to gender and school scores, managing to note through a cluster analysis how students with higher grades are also those who are more likely to be taken to work.

In the second part we discovered that in the country we are considering (India, Bangladesh or Pakistan) there may be a gender gap in placements even if the data are not enough to clearly state it.

Finally in the third part we try various methods to find the best predictive model of student placement,which turned out to be the “improved” random forest.