Assignment2

Irene Fernández Rebollo i Àlex Martorell i Locascio

5/12/2021

# Presentation

A company which is active in Big Data and Data Science wants to hire data scientists among people who successfully pass some courses which are conducted by the company. Many people signup for their training. Company wants to know which of these candidates really want to work for the company after training or looking for a new employment because it helps to reduce the cost and time as well as the quality of training or planning the courses and categorization of candidates. Information related to demographics, education, experience are in hands from candidates signup and enrollment.

This dataset designed to understand the factors that lead a person to leave current job for HR researches too. By model(s) that uses the current credentials, demographics, experience data you will predict the probability of a candidate to look for a new job or will work for the company, as well as interpreting affected factors on employee decision.

# Data Preparation

df <- read.csv("aug\_train.csv",header=T, sep=",", na.strings="NA")  
summary(df)

## enrollee\_id city city\_development\_index gender   
## Min. : 1 Length:19158 Min. :0.4480 Length:19158   
## 1st Qu.: 8554 Class :character 1st Qu.:0.7400 Class :character   
## Median :16983 Mode :character Median :0.9030 Mode :character   
## Mean :16875 Mean :0.8288   
## 3rd Qu.:25170 3rd Qu.:0.9200   
## Max. :33380 Max. :0.9490   
## relevent\_experience enrolled\_university education\_level major\_discipline   
## Length:19158 Length:19158 Length:19158 Length:19158   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## experience company\_size company\_type last\_new\_job   
## Length:19158 Length:19158 Length:19158 Length:19158   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## training\_hours target   
## Min. : 1.00 Min. :0.0000   
## 1st Qu.: 23.00 1st Qu.:0.0000   
## Median : 47.00 Median :0.0000   
## Mean : 65.37 Mean :0.2493   
## 3rd Qu.: 88.00 3rd Qu.:0.0000   
## Max. :336.00 Max. :1.0000

## Removing Duplicates and Irrelavant Observations

df <- unique(df) #No duplicates  
#nrow(df) == nrow(unique(df))   
  
set.seed(130798) #Birthday of 1 member of the group as random seed:  
samples <- as.vector(sort(sample(1:nrow(df),5000))) #Subset of 5000 observations  
df <- df[samples,]

## Fix structural errors

For coding purposes, blanks ("") in the dataframe are considered as NA’s.

sum(is.na(df)) #0  
df[df==""] <- NA  
sum(is.na(df)) #5433

Some inconsistencies are also checked and corrected.

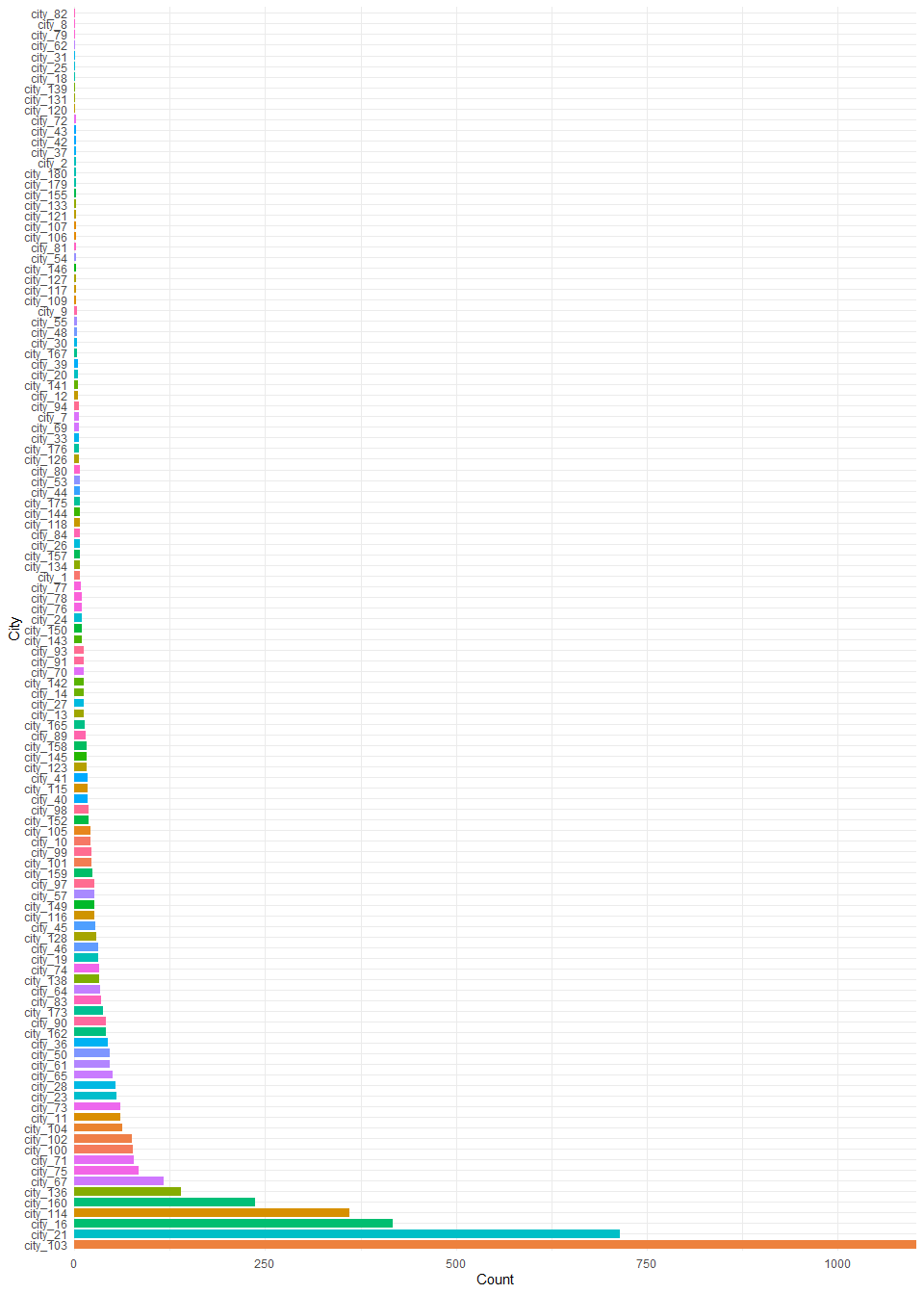
# Column name relevent\_experience should be relevant\_experience  
colnames(df)[5] <- "relevant\_experience"  
# Also their values should be with the "relevant" word  
df$relevant\_experience <- gsub("relevent", "relevant", df$relevant\_experience)  
  
# Correct the format of the company size  
df$company\_size[df$company\_size == "10/49"] <- "10-49"

## Univariate Descriptive Analysis

Out of the 14 variables in the dataset, R detects 9 of them being character-type. They are transformed into factors, taking into consideration NA values. One of the more tricky aspects is the years of experience variable. Two possibilites are taken into consideration, a factor with levels based on quartiles and a numeric variable.

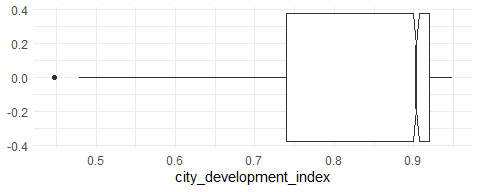
### City code

class(df$city) #"character"  
df$city <- factor(df$city) #Transform to "factor"



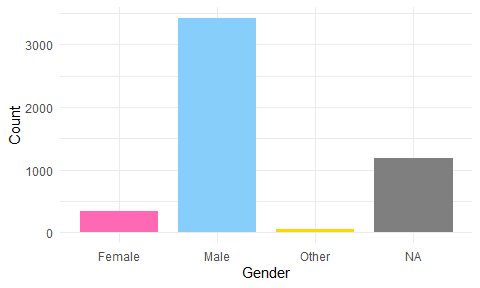
### Developement index of the city

class(df$city\_development\_index) #"numeric"  
# Kept as "numeric"



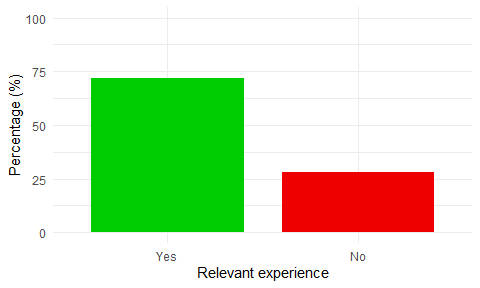
### Gender of candidate

class(df$gender) #"character"  
df$gender <- factor(df$gender) #Transform to "factor"



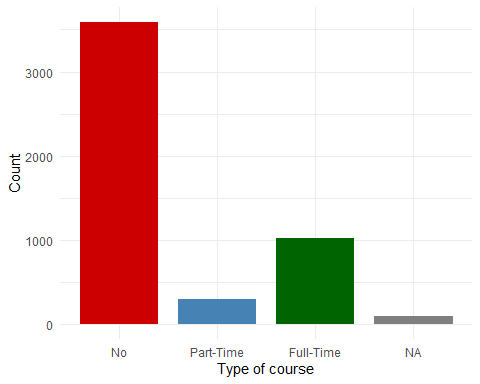
### Relevant experience of candidate

class(df$relevant\_experience) #"character"  
df$relevant\_experience <- factor(df$relevant\_experience,   
 levels=c("Has relevant experience", "No relevant experience"),  
 labels = c("Yes", "No")) #Transform to "factor" and change to simpler values



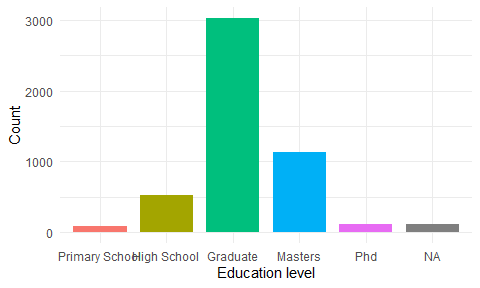
### Type of University course enrolled

class(df$enrolled\_university) #"character"  
df$enrolled\_university <- factor(df$enrolled\_university,   
 levels=c("no\_enrollment", "Part time course", "Full time course"),   
 labels= c("No", "Part-Time", "Full-Time")) #Transform to "factor" and change to simpler values



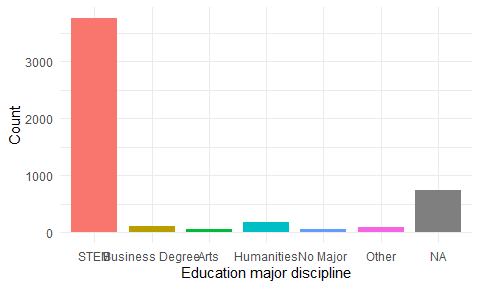
### Education level of candidate

class(df$education\_level) #"character"  
df$education\_level <- factor(df$education\_level,  
 levels = c("Primary School", "High School",  
 "Graduate", "Masters", "Phd")) #Transform to "factor"



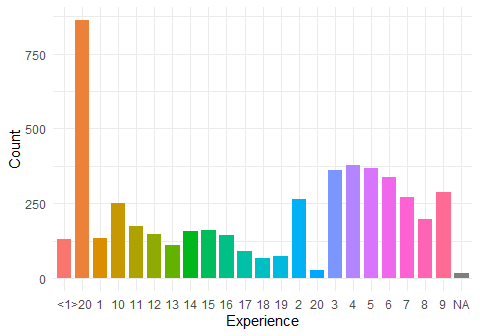
### Education major discipline of candidate

class(df$major\_discipline) #"character"  
df$major\_discipline <-factor(df$major\_discipline,  
 levels=c("STEM", "Business Degree", "Arts",   
 "Humanities", "No Major", "Other")) #Transform to "factor"



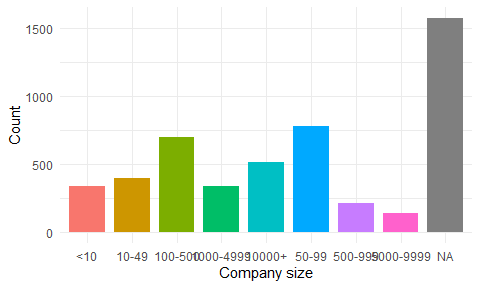
### Candidate total experience in years

class(df$experience) #"character"  
df$experience <- factor(df$experience) #Transform to "factor"



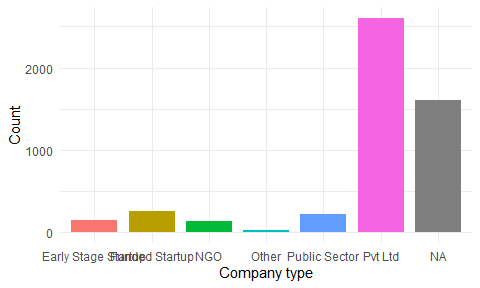
### Number of employees in current employer’s company

class(df$company\_size) #"character"  
df$company\_size <- factor(df$company\_size) #Transform to "factor"



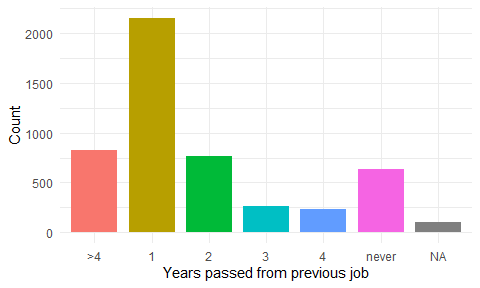
### Type of current employer

class(df$company\_type) #"character"  
df$company\_type <- factor(df$company\_type) #Transform to "factor"



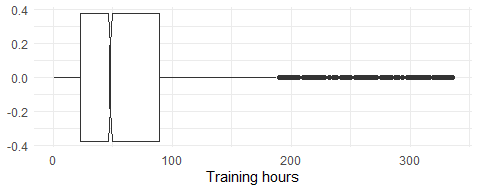
### Difference in years between previous job and current job

class(df$last\_new\_job) #"character"  
df$last\_new\_job <-factor(df$last\_new\_job) #Transform to "factor"



### Training hours completed

class(df$training\_hours) #"integer"  
#Keep as "integer"



## Missing values

The percentage of missing values per column with respect to their total varies significantly along different variables. and have over 30% of missing values. The numeric variables, the target, , and return no missing values.

missing <- unlist(lapply(df, function(x) sum(is.na(x))))/nrow(df)  
sort(missing[missing >= 0], decreasing = TRUE)

## company\_type company\_size gender   
## 0.3216 0.3152 0.2368   
## major\_discipline education\_level last\_new\_job   
## 0.1470 0.0230 0.0212   
## enrolled\_university experience enrollee\_id   
## 0.0186 0.0032 0.0000   
## city city\_development\_index relevant\_experience   
## 0.0000 0.0000 0.0000   
## training\_hours target   
## 0.0000 0.0000

count\_na<-colSums(is.na(df[, 1:14]))  
count\_na

## enrollee\_id city city\_development\_index   
## 0 0 0   
## gender relevant\_experience enrolled\_university   
## 1184 0 93   
## education\_level major\_discipline experience   
## 115 735 16   
## company\_size company\_type last\_new\_job   
## 1576 1608 106   
## training\_hours target   
## 0 0

sum(is.na(df)) #5433

## Outliers

### Univariate Outlier detection

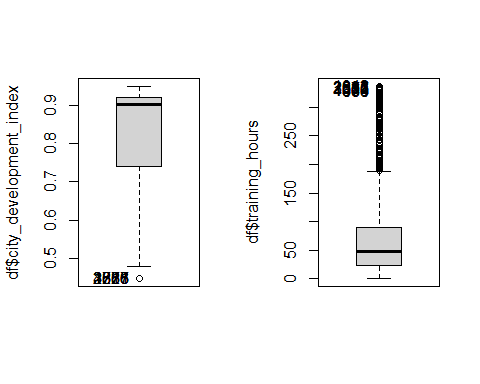
Outliers for the three numerical variables are considered. Training hours and city\_development\_index are fairly straightforward to analyze, but with regards to the experience variable there are no outliers due to the manner in which the variable was built. The characters “<1” and “>20” do not allow enough precision in order to identify for possible outliers, as they had to be converted to numbers 0 and 21 respectively during the conversion to a numeric variable (noting a significant bias).

The outliers found are registered in two different ways. A separate table is built for the Data Quality Report named (stands for Data Quality per Individual). This table stores the rows where mild and extreme outliers for training hours and mild outliers for city development index occur. A new variable called is added to the main data frame also for the Data Quality Report.

par(mfrow = c(1,2))  
Boxplot(df$city\_development\_index)

## [1] 1267 2376 2733 3518 3886 4621

Boxplot(df$training\_hours)



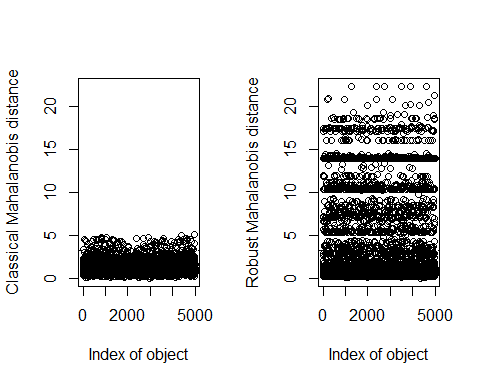
## [1] 1042 811 2685 803 2387 3576 1304 4376 463 664

mout\_city = quantile(df$city\_development\_index)[[2]]-1.5\*IQR(df$city\_development\_index)  
sum(df$city\_development\_index < mout\_city) #6  
mout\_th = quantile(df$training\_hours)[[4]]+1.5\*IQR(df$training\_hours)  
eout\_th = quantile(df$training\_hours)[[4]]+3\*IQR(df$training\_hours)  
sum(df$training\_hours > mout\_th) #227  
sum(df$training\_hours > eout\_th) #58

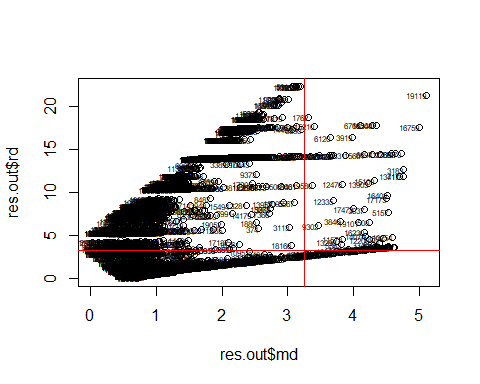
### Multivariate outlier detection

With the two numerical variables, we apply the Moutlier function at 99.5%. 105 multivariate outliers are returned.

res.out<-Moutlier(df[,c(3,13)],quantile=0.995)



#quantile(res.out$md,seq(0,1,0.005))  
multout<-which(res.out$md > res.out$cutoff)  
length(multout) #105



## Data Quality Report

To summarize all the its and buts of the Data Processing in a readable manner, two dataframes are created, one refers to variables and the other to individuals. This is the . For obvious reasons, it is possible to print out the whole data quality report per variable but not per individual (5000 in total). Also, a data quality variable is created.

df$quality <- 0

### Per variable

dqvar <- data.frame(colnames(df[, 1:14]))  
dqvar$outliers <-0  
dqvar$missing <-0  
dqvar$errors <-0  
  
# Outliers  
dqvar[13, "outliers"] <- sum(df$training\_hours > mout\_th) + sum(df$training\_hours > eout\_th)  
dqvar[3, "outliers"] <- sum(df$city\_development\_index < mout\_city)  
  
#Missing  
#We add missing values to the column per variable   
dqvar$missing <- (colSums(is.na(df[, 1:14])))  
dqvar

## colnames.df...1.14.. outliers missing errors  
## 1 enrollee\_id 0 0 0  
## 2 city 0 0 0  
## 3 city\_development\_index 6 0 0  
## 4 gender 0 1184 0  
## 5 relevant\_experience 0 0 0  
## 6 enrolled\_university 0 93 0  
## 7 education\_level 0 115 0  
## 8 major\_discipline 0 735 0  
## 9 experience 0 16 0  
## 10 company\_size 0 1576 0  
## 11 company\_type 0 1608 0  
## 12 last\_new\_job 0 106 0  
## 13 training\_hours 285 0 0  
## 14 target 0 0 0

### Per individual

dqind <- data.frame(df$enrollee\_id)  
colnames(dqind)[1] <- "enrollee\_id"  
dqind$missing <-0  
dqind$outliers <-0  
dqind$errors <-0  
  
#Outliers  
regmout\_th <- subset(df$enrollee\_id, df$training\_hours > mout\_th)  
regeout\_th <-subset(df$enrollee\_id, df$training\_hours > eout\_th)  
regmout\_city <-subset(df$enrollee\_id, df$city\_development\_index < mout\_city)  
  
for(i in 1:length(regmout\_th)) {  
 s<-which(df$enrollee\_id == regmout\_th[i])  
 w<-which(dqind$enrollee\_id == regmout\_th[i])  
 df[s, "quality"] <- df[s , "quality"] + 1  
 dqind[w, "outliers"] <- dqind[w, "outliers"] + 1}  
  
for(i in 1:length(regeout\_th)) { # for-loop over rows  
 s<-which(df$enrollee\_id == regeout\_th[i])  
 w<-which(dqind$enrollee\_id == regeout\_th[i])  
 df[s, "quality"] <- df[s , "quality"] + 1  
 dqind[w, "outliers"] <- dqind[w, "outliers"] + 1}  
  
for(i in 1:length(regmout\_city)) { # for-loop over rows  
 s<-which(df$enrollee\_id == regmout\_city[i])  
 w<-which(dqind$enrollee\_id == regmout\_city[i])  
 df[s , "quality"] <- df[s , "quality"] + 1  
 dqind[w, "outliers"] <- dqind[w, "outliers"] + 1}  
  
df[multout, "quality"] <- df[multout, "quality"] + 1  
dqind[multout, "outliers"] <- dqind[multout, "outliers"] + 1  
  
dqind$missing <- rowSums(is.na(df))  
df$quality <- df$quality + rowSums(is.na(df))  
head(dqind)

## enrollee\_id missing outliers errors  
## 1 666 0 0 0  
## 2 21651 3 0 0  
## 3 8722 4 0 0  
## 4 5764 1 0 0  
## 5 7041 0 0 0  
## 6 10408 0 0 0

Note that the quality variable is the sum of missing values and outliers. Using the package we obtain the correlation with the quantitative and qualitative variables. Note that company\_size and company\_type as factors seem to influence the most on the quality variable (R2 around 0.6). That is because of the large missing value number that these variables have.

# Correlation  
table(df$quality)

##   
## 0 1 2 3 4 5 6 7 8   
## 2171 1022 1000 545 178 54 23 4 3

cor(df[, c(3, 13:15)])

## city\_development\_index training\_hours target  
## city\_development\_index 1.000000000 -0.006019761 -0.32971376  
## training\_hours -0.006019761 1.000000000 -0.02210974  
## target -0.329713763 -0.022109738 1.00000000  
## quality -0.105990752 0.201059724 0.15757980  
## quality  
## city\_development\_index -0.1059908  
## training\_hours 0.2010597  
## target 0.1575798  
## quality 1.0000000

res.con <- condes(df, 15)  
res.con$quali

## R2 p.value  
## education\_level 0.29160790 0.000000e+00  
## company\_size 0.57934677 0.000000e+00  
## company\_type 0.57105470 0.000000e+00  
## major\_discipline 0.26162324 4.940656e-324  
## last\_new\_job 0.19613276 2.362457e-232  
## relevant\_experience 0.17783188 8.198615e-215  
## gender 0.16471079 1.284160e-194  
## enrolled\_university 0.13233214 2.088264e-153  
## experience 0.09964441 9.313658e-97  
## city 0.05151068 3.334249e-13

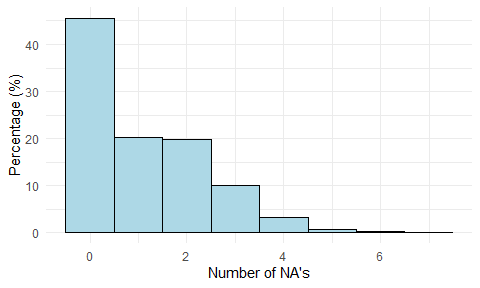
res.con$quanti

## correlation p.value  
## training\_hours 0.2010597 9.170131e-47  
## target 0.1575798 3.646471e-29  
## city\_development\_index -0.1059908 5.735764e-14

## Imputation of values

There are 14 variables and 5000 individuals; for each individual we can have from 0 to 7 NA’s along the variables. We can obtain an histogram were we can observe that there is a tail for higher number of missing values, so we determine a first rule of thumb that is that any row containing 5 missing values or more will be removed because of the cost that results when imputing so many values and only represents aprox. 1% of the samples.

#Create column with na counts for every observation  
df$na\_count <- rowSums(is.na(df))  
  
# Actualitzem aquí el Data Quality per Missing's  
for(i in 1:nrow(df)) { # for-loop over rows  
 df[i , "quality"] <- df[i , "quality"] + df[i, "na\_count"]  
 #w <- which(dqind$enrollee\_id == df[i, "enrollee\_id"])  
 dqind[i, "missing"] <- df$na\_count[i]}



df <- df[df$na\_count < 5,]  
nrow(df) #4946  
100 - nrow(df)/5000\*100 #only removed a 1.08% of the observations

The variables with some NA value are: gender (1184), enrolled\_university (93), education\_level (115), major\_discipline (735), experience (16), company\_size (1576), company\_type (1608), last\_new\_job (106). Each one is analysed separately to decide the required imputation method.

* Gender: In this case, there are a lot of NA’s. We decide to create a new category “Missing” because probably there are individuals that do not want to share this information.
* Enrolled university: In this case, we cannot assign a new category for NA’s since the variable stores already the three possible categories, so we decide to do imputation applying MCA.
* Education level: For this variable we also decide to impute NA’s with MCA because we cannot find any similarity of these values with the values of one category.
* Major discipline: Here the “STEM” category is more represented that the rest, so we try to collapse the rest in “Others” since the probabilities in the target are similar between them compared with the “STEM” value; following this approach, the NA’s were assigned into “Others”
* Experience: In experience, since we only have 10 missing values, we will use MCA imputation.
* Company size: There are a lot of NA’s in this variable, we create another category “Unknown” because is possible that the individual does not know this value.
* Company type: In this case, the number of missing values is very high. We tried to find some pattern in the probabilities of the categories in the target but the NA’s are very different from the rest. We decide to assign these NA’s values into the “Other” category because the number of individuals in the category are very low and does not make sense to create a “Missing” or “Unknown” category for this variable.
* Last new job: For this variable, analyzing the probabilities we saw some similarity with the category “4”, but as we are not sure of this assignation and the number of missing values not exceeds 100, we applied MCA imputation

summary(df)  
df$target <- factor(df$target, levels=c(1, 0), labels=c("Target.Yes", "Target.No"))  
table(df$target)  
  
#Gender  
levels(df$gender)[length(levels(df$gender)) + 1] <- "Missing"  
df[is.na(df$gender), "gender"] <- "Missing"  
  
#Enrolled university  
#levels(df$enrolled\_university)[length(levels(df$enrolled\_university)) + 1] <- "Missing"  
#df[is.na(df$enrolled\_university), "enrolled\_university"] <- "Missing"  
#prop.table(table(df$enrolled\_university, df$target)) #missMDA  
levels(df$enrolled\_university)  
nrow(df[is.na(df$enrolled\_university),]) #66  
  
#Education level  
#levels(df$education\_level)[length(levels(df$education\_level)) + 1] <- "Missing"  
#df[is.na(df$education\_level), "education\_level"] <- "Missing"  
#prop.table(table(df$education\_level, df$target)) #missMDA  
levels(df$education\_level)  
nrow(df[is.na(df$education\_level),]) #75  
  
#Major discipline  
levels(df$major\_discipline)[length(levels(df$major\_discipline)) + 1] <- "Missing"  
df[is.na(df$major\_discipline), "major\_discipline"] <- "Missing"  
prop.table(table(df$major\_discipline, df$target)) #STEM & other only?  
df[df$major\_discipline %in% c("Business Degree", "Arts", "Humanities", "No Major"), "major\_discipline"] <- "Other"  
table(df$major\_discipline)  
prop.table(table(df$major\_discipline, df$target)) #STEM & other only?  
df[df$major\_discipline == "Missing", "major\_discipline"] <- "Other"  
table(df$major\_discipline)  
prop.table(table(df$major\_discipline, df$target)) #STEM & other only?  
df$major\_discipline <- factor(df$major\_discipline)  
levels(df$major\_discipline)  
  
#Experience  
#levels(df$experience)[length(levels(df$experience)) + 1] <- "Missing"  
#df[is.na(df$experience), "experience"] <- "Missing"  
#prop.table(table(df$experience, df$target)) #missMDA  
levels(df$experience)  
nrow(df[is.na(df$experience),]) #10  
  
#Company size  
levels(df$company\_size)[length(levels(df$company\_size)) + 1] <- "Unknown"  
df[is.na(df$company\_size), "company\_size"] <- "Unknown"  
prop.table(table(df$company\_size, df$target)) #unknown  
  
#Company type  
#levels(df$company\_type)[length(levels(df$company\_type)) + 1] <- "Missing"  
#df[is.na(df$company\_type), "company\_type"] <- "Missing"  
#prop.table(table(df$company\_type, df$target)) #assign to other  
df[is.na(df$company\_type), "company\_type"] <- "Other"  
  
#Last new job  
#levels(df$last\_new\_job)[length(levels(df$last\_new\_job)) + 1] <- "Missing"  
#df[is.na(df$last\_new\_job), "last\_new\_job"] <- "Missing"  
#prop.table(table(df$last\_new\_job, df$target)) #missMDA  
levels(df$last\_new\_job)  
nrow(df[is.na(df$last\_new\_job),]) #91

For some variables with NA’s we use missDNA library to impute them. As it is a process that take some time, the code appear commented to avoid its execution; the resulting dataframe without NA’s is stored and we load it in the next step.

#library(missMDA)  
#dfimp <- df  
#colnames(dfimp)  
#res <- MCA(dfimp[, c(4:12)])  
#res <- MCA(dfimp[, c(6, 7, 9, 12)])  
#vars\_dis <- names(dfimp)[c(6, 7, 9, 12)]  
#summary(dfimp[,vars\_dis])  
#nb <- estim\_ncpMCA(dfimp[,vars\_dis],ncp.max=25)  
#res.input<-imputeMCA(dfimp[,vars\_dis],ncp=10)  
  
#Result of Imputation  
#summary(res.input$completeObs)  
#summary(df)  
#df$enrolled\_university <- res.input$completeObs$enrolled\_university  
#df$education\_level <- res.input$completeObs$education\_level  
#df$experience <- res.input$completeObs$experience  
#df$last\_new\_job <- res.input$completeObs$last\_new\_job  
#summary(df)  
#write.csv(df, "df\_imputation.csv", row.names = FALSE)

## Data Profiling

# Check if either training hours or city development index are normally distributed  
shapiro.test(df$city\_development\_index)  
shapiro.test(df$training\_hours)  
  
# B ~ X where B is the response factor variable and X is the explanatory cont. variable  
str(df)  
# Test on means  
oneway.test(df$city\_development\_index ~ df$target)  
kruskal.test(df$city\_development\_index ~ df$target)  
oneway.test(df$training\_hours ~ df$target)  
kruskal.test(df$training\_hours ~ df$target)  
  
#Test on variances  
fligner.test(df$city\_development\_index ~ df$target)  
fligner.test(df$training\_hours ~ df$target)  
  
#Catdes  
str(df)  
res.cat <- catdes(df, 14)  
res.cat$test.chi2  
res.cat$category  
res.cat$quanti.var  
res.cat$quanti

The following conclusions are reached after the Data Profiling: *It is observed using the Shapiro-Wilk Test that neither nor follow a normal distribution.* It is tested if the Target Factor is impacted by the numerical variables. In other words, does a certain value in these continuous variables affect the outcome.? Observe that even though both available tests are done, the valid results are the ones from Kruskal-Wallis test, since it does not assume normality on the explanatory variable. The output shows a 0 p-value for and around 0.05 for . In the first case it is clear, there is no influence on the target depending on the city where the employee resides. For the number of training hours it is not clear, since the p-value is in the gray area where no conclusion can be reached. *It is also important to check if dispersion of the numerical variables affects the target, and in both cases the answer is that it does not.* Using the package from global association between the target factor and categorical variables is assessed. The chi-squared test gives p-values a lot lower than 0.05 for all the factors. \*The output shows if the quantitative variables influence the target. Since all p-values are very low, the answer Yes. Similarly, the output shows if the mean in category varies with the overall mean.

## Interpretation of all the results before modelling

## Separation between Train and Test

# We upload the new dataframe with imputation results  
df <- read.csv("df\_imputation.csv",header=T, sep=",", na.strings="NA")  
for(i in 1:ncol(df)){  
 if(is.character(df[, i])){  
 df[, i] <-factor(df[, i])  
 }  
}  
#summary(df)  
df$quality <-NULL  
df$na\_count <-NULL

set.seed(130798)  
s <- sample(1:nrow(df),round(0.75\*nrow(df),0))  
dfall<-df  
df <- df[s,]  
dftest <-dfall[-s,]

# Modelling the Target

To simplify and differentiate kinds of models, the following table states the notation

The first model to try is the null model. A quick verifaction that the intercept in fact corresponds with the logit link function is performed.

# First model  
m0 <- glm(target ~ 1, family="binomial", data=df)  
ptt <- prop.table(table(df$target))  
summary(m0)

##   
## Call:  
## glm(formula = target ~ 1, family = "binomial", data = df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.7411 -0.7411 -0.7411 -0.7411 1.6890   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.15179 0.03843 -29.97 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4090.4 on 3709 degrees of freedom  
## Residual deviance: 4090.4 on 3709 degrees of freedom  
## AIC: 4092.4  
##   
## Number of Fisher Scoring iterations: 4

oddm0 <- ptt[2] / (1- ptt[2])  
log(oddm0)

## Target.Yes   
## -1.151793

## Modelling the target using numeric variables

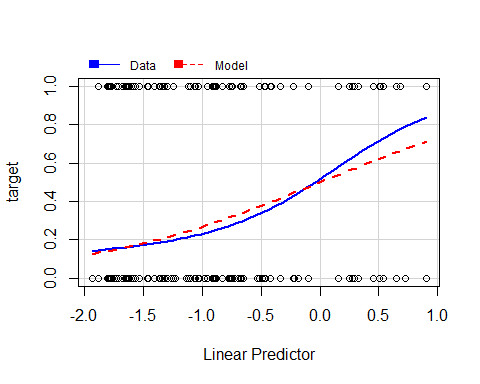
The numerical variables that we originally have are training hours and city development index. A significant improvement with regards to the AIC and the Deviance is observed with respect to the null model. The step function suggests to remove the training hours variable, as the AIC slightly decreases. We have labeled the models in this family as m1. Note that a quick overview of the marginal plots of suggests a variable transformation on city development index. Both a polynomial of degree 2 () and a logarithmic transformation are assessed (), getting a significantly better fit with the latter option, as the marginal plots show. However, the AIC worsens slightly in comparison to the polynomial model, although it is still better than the first model. We choose to keep model . Let us check also for influential observations: The Influence plot shows several observations with a large Cook’s distance (based on the area of the circle). To filter the important observations, we use a BoxPlot for the Cook’s distance which will yield the 10 instances with the largest Cook’s D.It is prudent to delete these observations, as it is a general way of proceeding with a posteriori influential data. The model is reestimated, and although the AIC decreases, this is not relevant because the data frame has 2 fewer observations. The Residual Deviance also shows a significant drop.

With regards to the final numerical model, a rule of thumb is applied to check for goodness of fit since the data is not aggregated. Since the residual deviance and the degrees of freedom are of the same order, the model can be labeled a good fit.

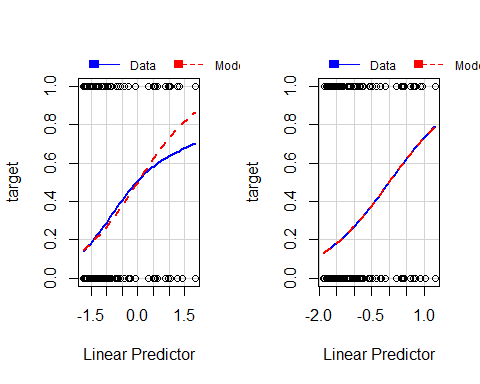
Finally, with function we see the logarithmic effect of the explanatory variable.

m1 <- glm(target ~ . , family="binomial", data=df[, c(3, 13, 14)])  
m11<- step(m1) #city\_development\_index + training\_hours  
m11 <- step(m1, k=log(nrow(df))) #BIC case city\_development\_index

marginalModelPlot(m11) #some transformation needed



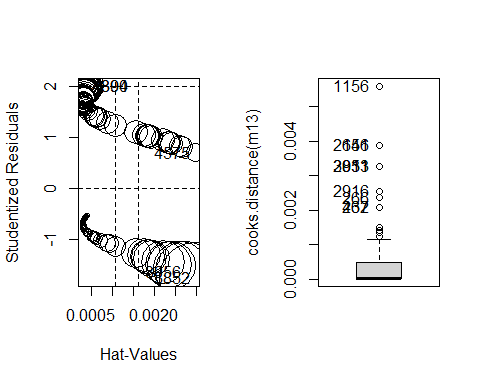
m12 <- glm(target ~ poly(city\_development\_index,2), family="binomial", data=df)  
m13 <- glm(target ~ log(city\_development\_index), family="binomial", data=df)  
#AIC(m11, m12, m13)  
  
par(mfrow = c(1, 2))  
marginalModelPlot(m12)  
marginalModelPlot(m13) #better fit



influencePlot(m13)

## StudRes Hat CookD  
## 4800 1.9931671 0.0004000836 0.0012572246  
## 4575 0.6887775 0.0029700418 0.0003991642  
## 4394 1.9931671 0.0004000836 0.0012572246  
## 3056 -1.6384995 0.0027469216 0.0038851200  
## 3852 -1.7675654 0.0029700418 0.0055908864

cook <- Boxplot(cooks.distance(m13))



cookd <- sort(cooks.distance(m13)[cook], decreasing=TRUE)  
cookd #10 influent observations

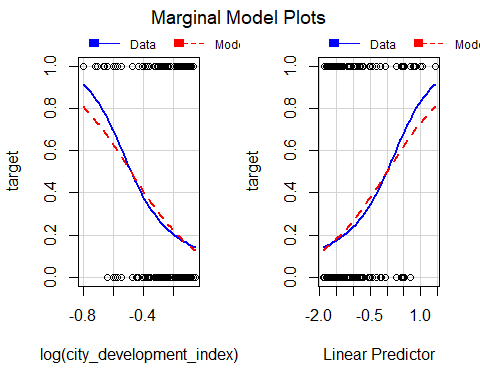
## 3852 3056 204 4271 3648 1969   
## 0.005590886 0.003885120 0.003885120 0.003265413 0.003265413 0.003265413   
## 1350 2998 1561 487   
## 0.002555228 0.002359506 0.002089487 0.002089487

df <- df[!(rownames(df) %in% names(cookd)),]  
  
m131 <- glm(target ~ log(city\_development\_index), family="binomial", data=df)  
AIC(m13, m131)

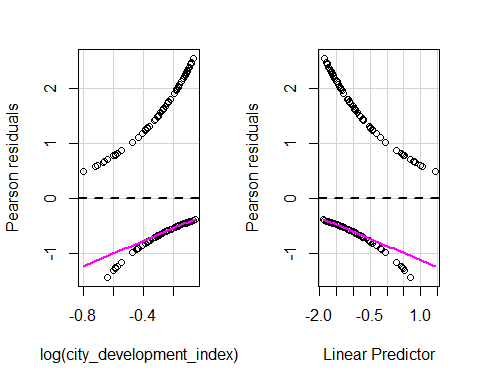
## Warning in AIC.default(m13, m131): models are not all fitted to the same number  
## of observations

## df AIC  
## m13 2 3731.910  
## m131 2 3706.603

#summary(m13)  
#summary(m131)  
marginalModelPlots(m131)



residualPlots(m131) #some values of the tail in m13 are removed



## Test stat Pr(>|Test stat|)   
## log(city\_development\_index) 8.417 0.003717 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#plot(allEffects((m131)))  
  
# Temporarily remove outliers: compare models  
#dfcheck <-df  
#ll <- which(df$enrollee\_id %in% regmout\_th)  
#dfcheck <- dfcheck[-ll,]  
#ll <- which(df$enrollee\_id %in% regeout\_th)  
#df <- df[-ll, ]  
#ll <- which(df$enrollee\_id %in% regmout\_city)  
#dfcheck <- dfcheck[-ll, ]  
  
#m14 <- glm(target ~ log(city\_development\_index) + training\_hours, family="binomial", data=dfcheck)  
#AIC(m14, m13)

## Introducing factor variables

The first important thing to note here is the difference between the Akaike Information Criterion (AIC from now on) and the Bayesian Information Criterion(BIC). The AIC has many different formulas that describe it, but the main idea is that takes into account the number of parameters in the model and the total of number observations. It applies a penalty based on the number of parameters. The BIC, however, applies a larger penalty. So in search of the best model, different results will be achieved according to the criteria used.

First, a general model m2 containing all . A significant drop in comparison with respect to the deviance from the null model is noted, however, there is an egregious amount of parameters and that is not good news. It will be shown that reducing the amount of factors (especially the ones with many factor levels)

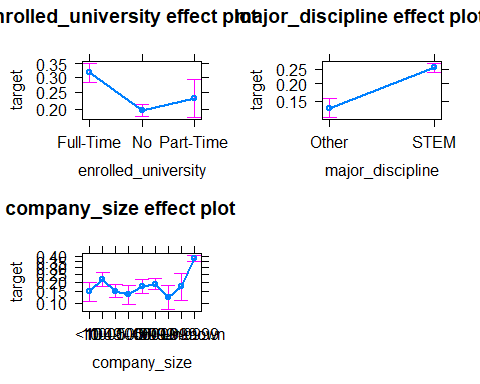
In this specific m2 model with the results printed in Appendix C, many p-values close to 1 are observed for some factor levels. This is clearly noticeable with the city factor. The step functions are now applied, for AIC and BIC criteria respectively. Both eliminate several variables: \* Using AIC criteria we obtain a model that eliminates factors , and \* Using BIC criteria we obtain a model that only considers 3 regressors: , and . We choose the latter model because it will be much easier to check for factor interactions in the enxt section. The function gives low p-values for all regresssors, which suggests a good fit. gives got results with the latter model, showing that all regressors are useful. Let us follow with a bit of interpretation:

m00 <-glm(target ~ 1, family="binomial", data=df)  
summary(m00)  
m2 <- glm(target ~ ., family="binomial", data=df[, c(2, 4:12, 14)])  
AIC(m2);BIC(m2)  
  
# AIC option  
#m21<-step(m2, trace=0)  
#m21$anova  
#summary(m21)  
  
#BIC option  
m211<-step(m2, k=log(nrow(df)), trace=0)  
m211$anova

summary(m211)

##   
## Call:  
## glm(formula = target ~ enrolled\_university + major\_discipline +   
## company\_size, family = "binomial", data = df[, c(2, 4:12,   
## 14)])  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2626 -0.7110 -0.5838 -0.3718 2.3266   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.88409 0.21639 -8.707 < 2e-16 \*\*\*  
## enrolled\_universityNo -0.63887 0.09468 -6.747 1.50e-11 \*\*\*  
## enrolled\_universityPart-Time -0.44335 0.18114 -2.448 0.0144 \*   
## major\_disciplineSTEM 0.80557 0.10447 7.711 1.25e-14 \*\*\*  
## company\_size10-49 0.47738 0.23042 2.072 0.0383 \*   
## company\_size100-500 0.03440 0.22126 0.155 0.8765   
## company\_size1000-4999 -0.11436 0.26185 -0.437 0.6623   
## company\_size10000+ 0.21821 0.22710 0.961 0.3366   
## company\_size50-99 0.30009 0.21328 1.407 0.1594   
## company\_size500-999 -0.21338 0.30400 -0.702 0.4827   
## company\_size5000-9999 0.19096 0.31521 0.606 0.5446   
## company\_sizeUnknown 1.27656 0.19651 6.496 8.24e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4084.9 on 3699 degrees of freedom  
## Residual deviance: 3769.0 on 3688 degrees of freedom  
## AIC: 3793  
##   
## Number of Fisher Scoring iterations: 4

#vif(m211)  
#Anova(m21, test="LR")  
#Anova(m211, test="LR")  
  
#AIC(m00, m211)  
plot(allEffects(m211), axes=list(y=list(lab="target")))

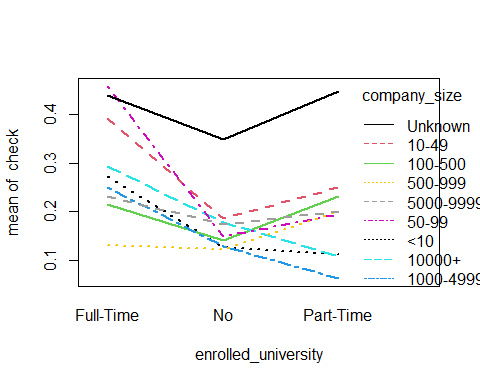


### Factor interactions

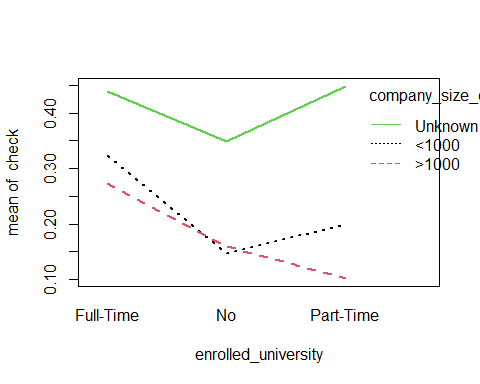
We go ahead and study some more properties for this model. Note that all the factors are additive, and we want to check interaction. We have four factors, and we calculate the different factor interactions:

Firstly, two by two factor interactions show that gender and en

check <- as.numeric(df$target) - 1  
  
# enrolled\_university - major\_discipline  
#with(df, interaction.plot(enrolled\_university, major\_discipline, check, lwd = 2, col = 1:2))  
m22 <- glm(target ~ enrolled\_university + major\_discipline, family="binomial", data=df)  
m23 <- glm(target ~ enrolled\_university \* major\_discipline, family="binomial", data=df)  
Anova(m23, test="LR") #No interaction  
  
# enrolled\_university - company\_size  
with(df, interaction.plot(enrolled\_university, company\_size, check, lwd = 2, col = 1:9))



m22 <- glm(target ~ enrolled\_university + company\_size, family="binomial", data=df)  
m23 <- glm(target ~ enrolled\_university \* company\_size, family="binomial", data=df)  
Anova(m23, test="LR") #No interaction, not clear...  
  
prop.table(table(df$enrolled\_university, df$company\_size), 2)  
table(df$company\_size)  
df$company\_size\_col <- fct\_collapse(df$company\_size, "<1000"= c("<10", "10-49", "50-99", "100-500", "500-999"), ">1000" = c("1000-4999", "5000-9999", "10000+"))  
  
table(df$company\_size\_col)  
with(df, interaction.plot(enrolled\_university, company\_size\_col, check, lwd = 2, col = 1:9))



# We try enrolled\_university \*company\_size\_col   
m24 <- glm(target ~ enrolled\_university \* company\_size\_col, family="binomial", data=df)  
Anova(m24, test="LR")  
AIC(m23, m24)  
summary(m23)  
summary(m24)  
BIC(m23, m24)  
BIC(m2)  
  
# major\_discipline - company\_size  
#with(df, interaction.plot(major\_discipline, company\_size, check, lwd = 2, col = 1:2))  
m22 <- glm(target ~ major\_discipline + company\_size, family="binomial", data=df)  
m23 <- glm(target ~ major\_discipline \* company\_size, family="binomial", data=df)  
Anova(m23, test="LR") #No interaction

The following table summarizes the different possibilities factor interactions:

## Best model

We end up with the best three models so far. The table below summarizes these models according to their Residual Deviance and their BIC. We end up chooosing model because it has the lowest BIC.

m3 <- glm(target ~ log(city\_development\_index) + enrolled\_university + major\_discipline + company\_size, family="binomial", data=df)  
summary(m3)

##   
## Call:  
## glm(formula = target ~ log(city\_development\_index) + enrolled\_university +   
## major\_discipline + company\_size, family = "binomial", data = df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8760 -0.6761 -0.4917 -0.3127 2.4920   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.91921 0.23699 -12.318 < 2e-16 \*\*\*  
## log(city\_development\_index) -4.20814 0.24445 -17.215 < 2e-16 \*\*\*  
## enrolled\_universityNo -0.40489 0.10038 -4.033 5.50e-05 \*\*\*  
## enrolled\_universityPart-Time -0.31651 0.18995 -1.666 0.0957 .   
## major\_disciplineSTEM 0.65921 0.11036 5.973 2.33e-09 \*\*\*  
## company\_size10-49 0.37147 0.24212 1.534 0.1250   
## company\_size100-500 0.08381 0.23095 0.363 0.7167   
## company\_size1000-4999 -0.02022 0.27240 -0.074 0.9408   
## company\_size10000+ 0.28032 0.23731 1.181 0.2375   
## company\_size50-99 0.27582 0.22339 1.235 0.2169   
## company\_size500-999 -0.15387 0.31716 -0.485 0.6276   
## company\_size5000-9999 0.28832 0.32755 0.880 0.3787   
## company\_sizeUnknown 1.35312 0.20566 6.579 4.73e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4084.9 on 3699 degrees of freedom  
## Residual deviance: 3462.6 on 3687 degrees of freedom  
## AIC: 3488.6  
##   
## Number of Fisher Scoring iterations: 4

m31 <- glm(target ~ log(city\_development\_index) + enrolled\_university + major\_discipline + company\_size\_col, family="binomial", data=df)  
summary(m31)

##   
## Call:  
## glm(formula = target ~ log(city\_development\_index) + enrolled\_university +   
## major\_discipline + company\_size\_col, family = "binomial",   
## data = df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8820 -0.6743 -0.4738 -0.3380 2.4277   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.75301 0.15577 -17.674 < 2e-16 \*\*\*  
## log(city\_development\_index) -4.24099 0.24403 -17.379 < 2e-16 \*\*\*  
## enrolled\_universityNo -0.40674 0.10031 -4.055 5.02e-05 \*\*\*  
## enrolled\_universityPart-Time -0.31484 0.18996 -1.657 0.0974 .   
## major\_disciplineSTEM 0.66068 0.11027 5.991 2.08e-09 \*\*\*  
## company\_size\_col>1000 0.01484 0.12254 0.121 0.9036   
## company\_size\_colUnknown 1.17960 0.09848 11.979 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4084.9 on 3699 degrees of freedom  
## Residual deviance: 3470.1 on 3693 degrees of freedom  
## AIC: 3484.1  
##   
## Number of Fisher Scoring iterations: 4

m32 <- glm(target ~ log(city\_development\_index) + enrolled\_university \* company\_size\_col + major\_discipline, family="binomial", data=df)  
summary(m32)

##   
## Call:  
## glm(formula = target ~ log(city\_development\_index) + enrolled\_university \*   
## company\_size\_col + major\_discipline, family = "binomial",   
## data = df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8426 -0.6697 -0.4669 -0.3261 2.4562   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -2.489717 0.186039  
## log(city\_development\_index) -4.219902 0.244814  
## enrolled\_universityNo -0.742248 0.165637  
## enrolled\_universityPart-Time -0.551527 0.287386  
## company\_size\_col>1000 -0.007614 0.298179  
## company\_size\_colUnknown 0.757026 0.179496  
## major\_disciplineSTEM 0.665158 0.110119  
## enrolled\_universityNo:company\_size\_col>1000 0.088418 0.329220  
## enrolled\_universityPart-Time:company\_size\_col>1000 -0.551788 0.619952  
## enrolled\_universityNo:company\_size\_colUnknown 0.596906 0.213501  
## enrolled\_universityPart-Time:company\_size\_colUnknown 0.591548 0.414765  
## z value Pr(>|z|)   
## (Intercept) -13.383 < 2e-16 \*\*\*  
## log(city\_development\_index) -17.237 < 2e-16 \*\*\*  
## enrolled\_universityNo -4.481 7.42e-06 \*\*\*  
## enrolled\_universityPart-Time -1.919 0.05497 .   
## company\_size\_col>1000 -0.026 0.97963   
## company\_size\_colUnknown 4.218 2.47e-05 \*\*\*  
## major\_disciplineSTEM 6.040 1.54e-09 \*\*\*  
## enrolled\_universityNo:company\_size\_col>1000 0.269 0.78826   
## enrolled\_universityPart-Time:company\_size\_col>1000 -0.890 0.37344   
## enrolled\_universityNo:company\_size\_colUnknown 2.796 0.00518 \*\*   
## enrolled\_universityPart-Time:company\_size\_colUnknown 1.426 0.15380   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4084.9 on 3699 degrees of freedom  
## Residual deviance: 3459.5 on 3689 degrees of freedom  
## AIC: 3481.5  
##   
## Number of Fisher Scoring iterations: 4

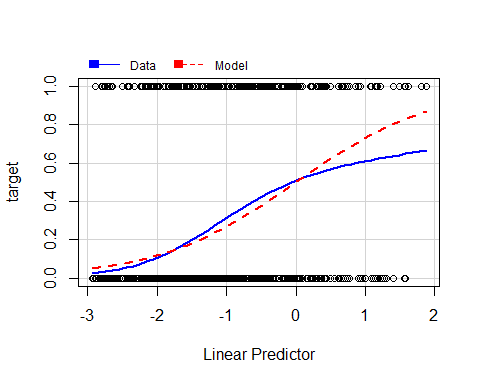
anova(m31, m32, test="Chisq")

## Analysis of Deviance Table  
##   
## Model 1: target ~ log(city\_development\_index) + enrolled\_university +   
## major\_discipline + company\_size\_col  
## Model 2: target ~ log(city\_development\_index) + enrolled\_university \*   
## company\_size\_col + major\_discipline  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 3693 3470.1   
## 2 3689 3459.5 4 10.597 0.03149 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

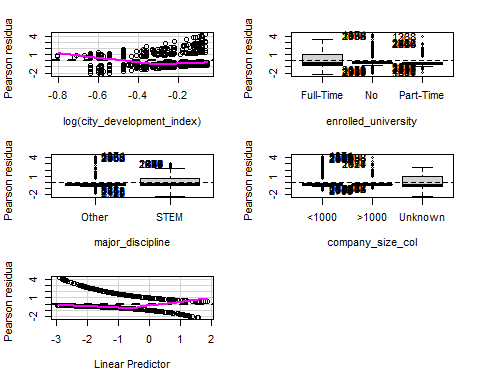
BIC(m3, m31, m32)

## df BIC  
## m3 13 3569.370  
## m31 7 3527.602  
## m32 11 3549.870

marginalModelPlot(m31)

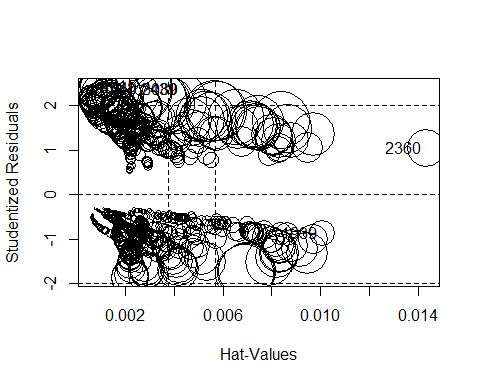


residualPlots(m31)



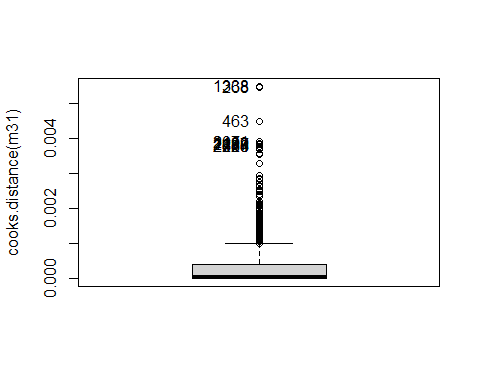
## Test stat Pr(>|Test stat|)   
## log(city\_development\_index) 16.618 4.572e-05 \*\*\*  
## enrolled\_university   
## major\_discipline   
## company\_size\_col   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

influencePlot(m31)



## StudRes Hat CookD  
## 4040 2.396571 0.0008409606 0.0019911097  
## 3080 2.365347 0.0025176483 0.0054558582  
## 2439 2.376315 0.0024654093 0.0054914187  
## 4715 2.430624 0.0007980146 0.0020603622  
## 2360 1.042831 0.0143131740 0.0015017039  
## 1939 -0.882225 0.0100717567 0.0006933083

cook <- Boxplot(cooks.distance(m31))



cookd <- sort(cooks.distance(m31)[cook], decreasing=TRUE)  
cookd

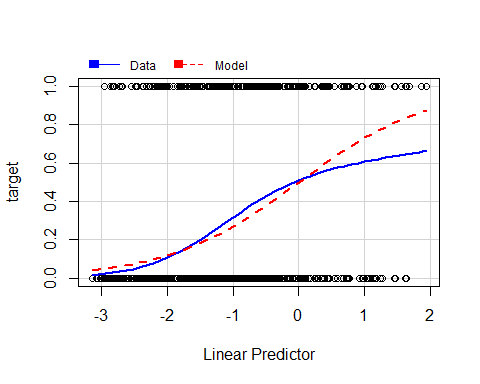
## 2439 3080 1930 3769 2942 4177   
## 0.005491419 0.005455858 0.004488095 0.003904845 0.003858279 0.003829276   
## 2369 2720 2812 561   
## 0.003788255 0.003788255 0.003788255 0.003788255

df<-df[!(rownames(df) %in% names(cookd)),]  
  
m4 <- glm(target ~ log(city\_development\_index) + enrolled\_university + major\_discipline + company\_size\_col, family="binomial", data=df)

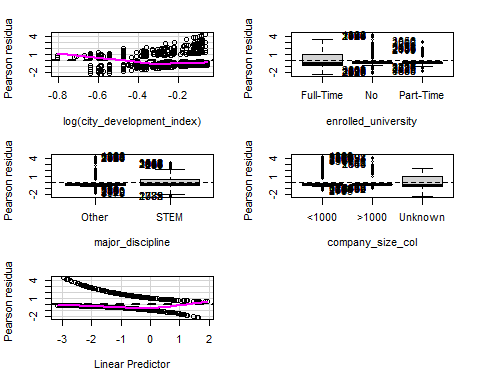
## Analysis, Goodness of Fit for our Final model.

First, we output all the important analysis and results for a final model.

marginalModelPlot(m4)

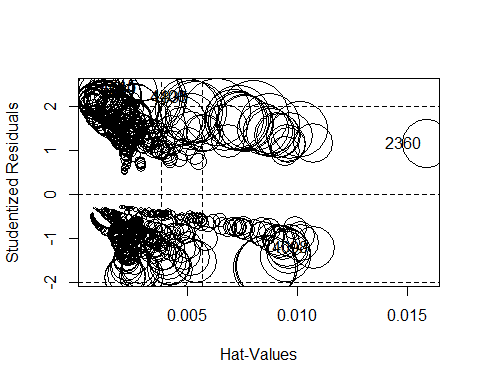


residualPlots(m4)



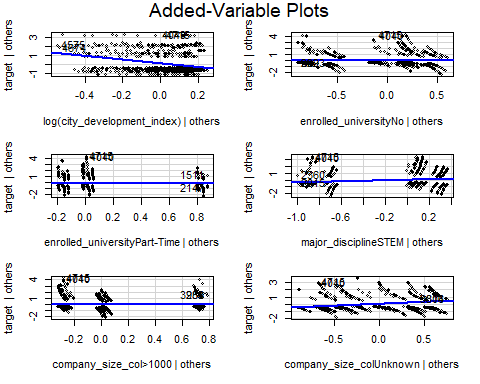
## Test stat Pr(>|Test stat|)   
## log(city\_development\_index) 14.78 0.0001208 \*\*\*  
## enrolled\_university   
## major\_discipline   
## company\_size\_col   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

influencePlot(m4)



## StudRes Hat CookD  
## 4040 2.418009 0.0008267887 0.002066861  
## 4098 -1.206881 0.0107284969 0.001659478  
## 4715 2.452682 0.0007828851 0.002138661  
## 1835 2.202387 0.0031677991 0.004611018  
## 2360 1.153485 0.0158606369 0.002176775  
## 4400 2.222074 0.0030685641 0.004682812

avPlots(m4)



summary(m4)

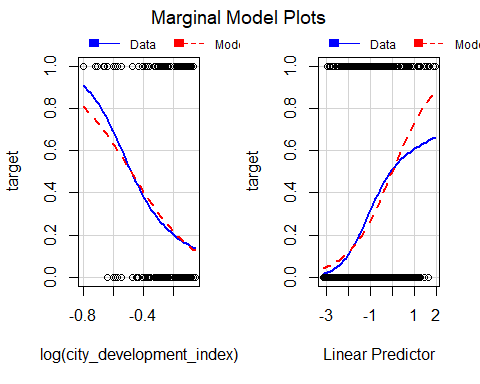
##   
## Call:  
## glm(formula = target ~ log(city\_development\_index) + enrolled\_university +   
## major\_discipline + company\_size\_col, family = "binomial",   
## data = df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9044 -0.6719 -0.4724 -0.3291 2.4496   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.82334 0.15782 -17.890 < 2e-16 \*\*\*  
## log(city\_development\_index) -4.34405 0.24585 -17.669 < 2e-16 \*\*\*  
## enrolled\_universityNo -0.39937 0.10072 -3.965 7.34e-05 \*\*\*  
## enrolled\_universityPart-Time -0.59271 0.20209 -2.933 0.00336 \*\*   
## major\_disciplineSTEM 0.69138 0.11172 6.188 6.08e-10 \*\*\*  
## company\_size\_col>1000 0.03253 0.12350 0.263 0.79224   
## company\_size\_colUnknown 1.20947 0.09941 12.166 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4056.3 on 3689 degrees of freedom  
## Residual deviance: 3423.1 on 3683 degrees of freedom  
## AIC: 3437.1  
##   
## Number of Fisher Scoring iterations: 4

sum( resid( m4, "pearson") ^2 )

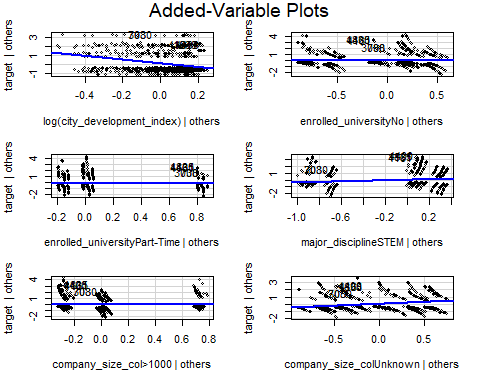
## [1] 3559.038

marginalModelPlots(m4,id=list(labels=row.names(df),method=abs(cooks.distance(m4)), n=5) )

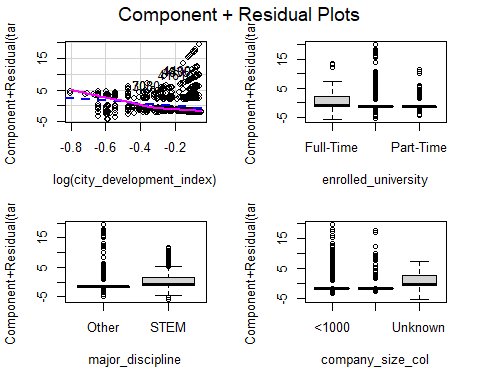
## Warning in mmps(...): Interactions and/or factors skipped



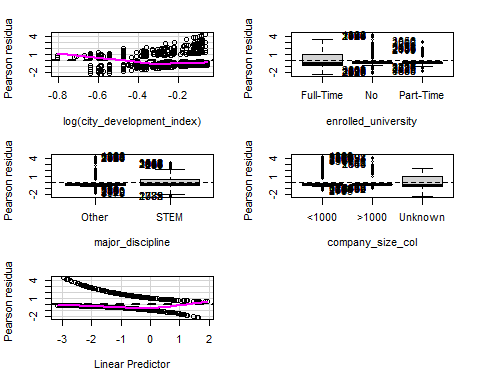
avPlots(m4,id=list(labels=row.names(df),method=abs(cooks.distance(m4)), n=5) )



crPlots(m4,id=list(labels=row.names(df),method=abs(cooks.distance(m4)), n=5) )



residualPlots(m4, layout=c(3, 2))

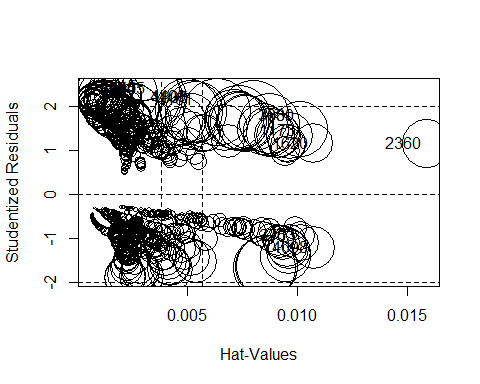


## Test stat Pr(>|Test stat|)   
## log(city\_development\_index) 14.78 0.0001208 \*\*\*  
## enrolled\_university   
## major\_discipline   
## company\_size\_col   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

outlierTest(m4)

## No Studentized residuals with Bonferroni p < 0.05  
## Largest |rstudent|:  
## rstudent unadjusted p-value Bonferroni p  
## 4715 2.452682 0.01418 NA

par(mfrow=c(1,1))  
influencePlot(m4,id=list(n=5) )



## StudRes Hat CookD  
## 2505 2.4065056 0.0012075197 0.0029242437  
## 4040 2.4180088 0.0008267887 0.0020668610  
## 3763 -0.9219496 0.0101404856 0.0007768966  
## 1173 1.4666355 0.0101404856 0.0028136102  
## 4098 -1.2068811 0.0107284969 0.0016594781  
## 1620 1.1573378 0.0107284969 0.0014779106  
## 4715 2.4526817 0.0007828851 0.0021386614  
## 1835 2.2023867 0.0031677991 0.0046110184  
## 2360 1.1534845 0.0158606369 0.0021767747  
## 3030 1.7944989 0.0079789667 0.0045462022  
## 26 2.4180088 0.0008267887 0.0020668610  
## 708 1.7944989 0.0079789667 0.0045462022  
## 4400 2.2220736 0.0030685641 0.0046828120  
## 4161 2.1620738 0.0033790765 0.0044678156  
## 3395 2.4180088 0.0008267887 0.0020668610

model.final <- lrm(target ~ log(city\_development\_index) + enrolled\_university + major\_discipline + company\_size\_col, data=df)  
model.final

## Logistic Regression Model  
##   
## lrm(formula = target ~ log(city\_development\_index) + enrolled\_university +   
## major\_discipline + company\_size\_col, data = df)  
##   
## Model Likelihood Discrimination Rank Discrim.   
## Ratio Test Indexes Indexes   
## Obs 3690 LR chi2 633.23 R2 0.236 C 0.775   
## Target.No 2809 d.f. 6 g 1.124 Dxy 0.549   
## Target.Yes 881 Pr(> chi2) <0.0001 gr 3.078 gamma 0.556   
## max |deriv| 8e-14 gp 0.190 tau-a 0.200   
## Brier 0.151   
##   
## Coef S.E. Wald Z Pr(>|Z|)  
## Intercept -2.8233 0.1578 -17.89 <0.0001   
## city\_development\_index -4.3441 0.2459 -17.67 <0.0001   
## enrolled\_university=No -0.3994 0.1007 -3.97 <0.0001   
## enrolled\_university=Part-Time -0.5927 0.2021 -2.93 0.0034   
## major\_discipline=STEM 0.6914 0.1117 6.19 <0.0001   
## company\_size\_col=>1000 0.0325 0.1235 0.26 0.7922   
## company\_size\_col=Unknown 1.2095 0.0994 12.17 <0.0001   
##

m0 <- glm(target ~ 1, family="binomial", data=df)  
NagelkerkeR2(m4)

## $N  
## [1] 3690  
##   
## $R2  
## [1] 0.236457

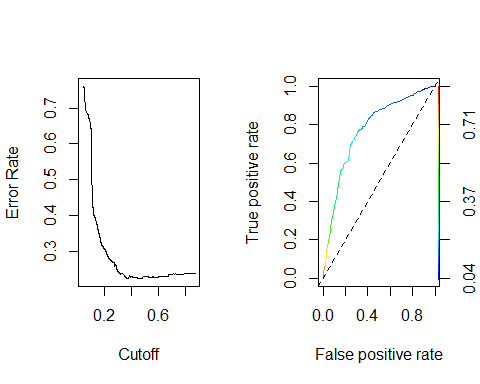
summary(m4)

##   
## Call:  
## glm(formula = target ~ log(city\_development\_index) + enrolled\_university +   
## major\_discipline + company\_size\_col, family = "binomial",   
## data = df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9044 -0.6719 -0.4724 -0.3291 2.4496   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.82334 0.15782 -17.890 < 2e-16 \*\*\*  
## log(city\_development\_index) -4.34405 0.24585 -17.669 < 2e-16 \*\*\*  
## enrolled\_universityNo -0.39937 0.10072 -3.965 7.34e-05 \*\*\*  
## enrolled\_universityPart-Time -0.59271 0.20209 -2.933 0.00336 \*\*   
## major\_disciplineSTEM 0.69138 0.11172 6.188 6.08e-10 \*\*\*  
## company\_size\_col>1000 0.03253 0.12350 0.263 0.79224   
## company\_size\_colUnknown 1.20947 0.09941 12.166 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4056.3 on 3689 degrees of freedom  
## Residual deviance: 3423.1 on 3683 degrees of freedom  
## AIC: 3437.1  
##   
## Number of Fisher Scoring iterations: 4

100\*(1-m4$dev/m4$null.dev)

## [1] 15.61095

dadesroc<-prediction(predict(m4,type="response"),df$target)  
par(mfrow=c(1,2))  
plot(performance(dadesroc,"err"))  
plot(performance(dadesroc,"tpr","fpr"), colorize=TRUE)  
abline(0,1,lty=2)



library(cvAUC)

## Loading required package: data.table

##   
## Attaching package: 'data.table'

## The following object is masked from 'package:purrr':  
##   
## transpose

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

##

## cvAUC version: 1.1.0

## Notice to cvAUC users: Major speed improvements in version 1.1.0

##

AUC(predict(m4,type="response"),df$target)

## [1] 0.7747458

Things to be said: *With regards to the ROC curve, we seem to get a decent fit. The AUC (Area under curve) is 0.77, which is a labeled as a good fit by statisticians.* The Nagel-Kerke test returns a pseudo-coefficient of determination of 0.24. Not knowing if this a good fit, we compare it with Lab 6 Election 92 Diagnostics, that presented a final model that returned an R2 of 0.25, so our fit seems adequate enough. \*

## Forecasting capability of the final model

df.fin <- m4  
dftest$company\_size\_col <- fct\_collapse(dftest$company\_size, "<1000"= c("<10", "10-49", "50-99", "100-500", "500-999"), ">1000" = c("1000-4999", "5000-9999", "10000+") )  
job.vot <- predict(df.fin, newdata=dftest, type="response")  
job.est <- ifelse(job.vot < 0.5, 0, 1)  
table(job.est, dftest$target)

##   
## job.est Target.No Target.Yes  
## 0 888 234  
## 1 49 65

sum(diag(table(job.est,dftest$target)))/dim(dftest)[1]

## [1] 0.7710356

# Null model  
m0<-glm(target ~ 1, family=binomial, data=dftest)  
job.vot0 <- m0$fit  
job.est0 <- ifelse(job.vot0<0.5,0,1)  
table(job.est0,dftest$target)

##   
## job.est0 Target.No Target.Yes  
## 0 937 299

table(job.est0,dftest$target)[1,2]/dim(dftest)[1]

## [1] 0.2419094

library(ResourceSelection)

## ResourceSelection 0.3-5 2019-07-22

hoslem.test(dftest$target, job.vot)

## Warning in Ops.factor(1, y): '-' not meaningful for factors

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: dftest$target, job.vot  
## X-squared = 1236, df = 8, p-value < 2.2e-16

# Appendix

This last section includes extra information or plots that are not part of the main report in an attempt to avoid overloading the plot ## Appendix A: More on Data Preparation

## Appendix B: More on the Target Modelling using Covariates

## Appendix C: More on the Target Moddeling using Factors

summary(m2)

##   
## Call:  
## glm(formula = target ~ ., family = "binomial", data = df[, c(2,   
## 4:12, 14)])  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.3981 -0.6128 -0.4180 -0.0003 2.9047   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.610e+01 1.063e+03 -0.015 0.98791   
## citycity\_10 1.429e+01 1.063e+03 0.013 0.98927   
## citycity\_100 1.545e+01 1.063e+03 0.015 0.98840   
## citycity\_101 1.606e+01 1.063e+03 0.015 0.98795   
## citycity\_102 1.455e+01 1.063e+03 0.014 0.98908   
## citycity\_103 1.525e+01 1.063e+03 0.014 0.98855   
## citycity\_104 1.465e+01 1.063e+03 0.014 0.98901   
## citycity\_105 1.365e+01 1.063e+03 0.013 0.98975   
## citycity\_106 -1.241e+00 2.624e+03 0.000 0.99962   
## citycity\_107 3.156e+01 2.624e+03 0.012 0.99041   
## citycity\_109 3.676e-01 2.624e+03 0.000 0.99989   
## citycity\_11 1.658e+01 1.063e+03 0.016 0.98755   
## citycity\_114 1.439e+01 1.063e+03 0.014 0.98920   
## citycity\_115 1.586e+01 1.063e+03 0.015 0.98809   
## citycity\_116 1.549e+01 1.063e+03 0.015 0.98837   
## citycity\_117 1.690e+01 1.063e+03 0.016 0.98732   
## citycity\_118 1.559e+01 1.063e+03 0.015 0.98830   
## citycity\_12 7.325e-01 1.957e+03 0.000 0.99970   
## citycity\_120 4.291e-01 2.624e+03 0.000 0.99987   
## citycity\_121 -5.503e-01 2.624e+03 0.000 0.99983   
## citycity\_123 1.708e+01 1.063e+03 0.016 0.98718   
## citycity\_127 -1.411e-01 1.729e+03 0.000 0.99993   
## citycity\_128 1.709e+01 1.063e+03 0.016 0.98717   
## citycity\_13 1.451e+01 1.063e+03 0.014 0.98910   
## citycity\_131 3.204e+01 2.624e+03 0.012 0.99026   
## citycity\_133 1.911e-01 2.624e+03 0.000 0.99994   
## citycity\_134 -4.249e-02 1.986e+03 0.000 0.99998   
## citycity\_136 1.413e+01 1.063e+03 0.013 0.98939   
## citycity\_138 1.438e+01 1.063e+03 0.014 0.98920   
## citycity\_139 3.326e+01 2.624e+03 0.013 0.98989   
## citycity\_14 1.581e+01 1.063e+03 0.015 0.98813   
## citycity\_141 1.618e+01 1.063e+03 0.015 0.98786   
## citycity\_142 1.541e+01 1.063e+03 0.014 0.98843   
## citycity\_143 1.638e+01 1.063e+03 0.015 0.98770   
## citycity\_144 1.491e+01 1.063e+03 0.014 0.98881   
## citycity\_145 1.496e+01 1.063e+03 0.014 0.98877   
## citycity\_146 1.244e-01 2.624e+03 0.000 0.99996   
## citycity\_149 1.501e+01 1.063e+03 0.014 0.98873   
## citycity\_150 -2.299e-01 1.377e+03 0.000 0.99987   
## citycity\_152 1.498e+01 1.063e+03 0.014 0.98875   
## citycity\_155 3.313e+01 2.624e+03 0.013 0.98993   
## citycity\_157 1.539e+01 1.063e+03 0.014 0.98845   
## citycity\_158 1.541e+01 1.063e+03 0.014 0.98843   
## citycity\_159 1.365e+01 1.063e+03 0.013 0.98975   
## citycity\_16 1.475e+01 1.063e+03 0.014 0.98893   
## citycity\_160 1.509e+01 1.063e+03 0.014 0.98867   
## citycity\_162 1.548e+01 1.063e+03 0.015 0.98838   
## citycity\_165 1.536e+01 1.063e+03 0.014 0.98847   
## citycity\_167 1.468e+01 1.063e+03 0.014 0.98898   
## citycity\_173 1.305e+01 1.063e+03 0.012 0.99021   
## citycity\_175 1.455e+01 1.063e+03 0.014 0.98908   
## citycity\_176 1.637e+01 1.063e+03 0.015 0.98771   
## citycity\_179 3.350e+01 2.624e+03 0.013 0.98982   
## citycity\_18 1.647e-01 2.624e+03 0.000 0.99995   
## citycity\_180 -2.123e+00 2.624e+03 -0.001 0.99935   
## citycity\_19 1.553e+01 1.063e+03 0.015 0.98835   
## citycity\_2 4.655e-01 2.624e+03 0.000 0.99986   
## citycity\_20 1.516e+01 1.063e+03 0.014 0.98862   
## citycity\_21 1.687e+01 1.063e+03 0.016 0.98733   
## citycity\_23 1.377e+01 1.063e+03 0.013 0.98967   
## citycity\_24 1.472e+01 1.063e+03 0.014 0.98895   
## citycity\_25 3.183e+01 2.624e+03 0.012 0.99032   
## citycity\_26 1.749e+01 1.063e+03 0.016 0.98687   
## citycity\_27 1.468e+01 1.063e+03 0.014 0.98898   
## citycity\_28 1.410e+01 1.063e+03 0.013 0.98941   
## citycity\_30 1.089e+00 1.736e+03 0.001 0.99950   
## citycity\_31 -9.395e-01 2.624e+03 0.000 0.99971   
## citycity\_33 3.330e+01 1.962e+03 0.017 0.98646   
## citycity\_36 1.416e+01 1.063e+03 0.013 0.98937   
## citycity\_37 1.572e+01 1.063e+03 0.015 0.98820   
## citycity\_39 3.988e-01 1.509e+03 0.000 0.99979   
## citycity\_40 1.413e+01 1.063e+03 0.013 0.98939   
## citycity\_41 1.336e+01 1.063e+03 0.013 0.98997   
## citycity\_42 1.613e+01 1.063e+03 0.015 0.98789   
## citycity\_43 3.183e+01 1.995e+03 0.016 0.98727   
## citycity\_44 1.607e+01 1.063e+03 0.015 0.98793   
## citycity\_45 1.611e+01 1.063e+03 0.015 0.98791   
## citycity\_46 1.567e+01 1.063e+03 0.015 0.98824   
## citycity\_48 3.345e+01 2.624e+03 0.013 0.98983   
## citycity\_50 1.463e+01 1.063e+03 0.014 0.98902   
## citycity\_53 -2.239e-01 1.397e+03 0.000 0.99987   
## citycity\_54 6.850e-01 2.001e+03 0.000 0.99973   
## citycity\_55 1.552e+01 1.063e+03 0.015 0.98835   
## citycity\_57 1.458e+01 1.063e+03 0.014 0.98905   
## citycity\_61 1.365e+01 1.063e+03 0.013 0.98976   
## citycity\_62 6.525e-01 2.624e+03 0.000 0.99980   
## citycity\_64 1.460e+01 1.063e+03 0.014 0.98904   
## citycity\_65 1.483e+01 1.063e+03 0.014 0.98887   
## citycity\_67 1.388e+01 1.063e+03 0.013 0.98958   
## citycity\_69 1.588e+01 1.063e+03 0.015 0.98808   
## citycity\_7 1.474e+01 1.063e+03 0.014 0.98894   
## citycity\_70 1.578e+01 1.063e+03 0.015 0.98816   
## citycity\_71 1.490e+01 1.063e+03 0.014 0.98881   
## citycity\_72 1.074e+00 2.624e+03 0.000 0.99967   
## citycity\_73 1.515e+01 1.063e+03 0.014 0.98862   
## citycity\_74 1.696e+01 1.063e+03 0.016 0.98727   
## citycity\_75 1.482e+01 1.063e+03 0.014 0.98888   
## citycity\_76 1.647e+01 1.063e+03 0.015 0.98764   
## citycity\_77 5.594e-01 1.489e+03 0.000 0.99970   
## citycity\_78 1.603e+01 1.063e+03 0.015 0.98797   
## citycity\_8 -1.158e-03 2.624e+03 0.000 1.00000   
## citycity\_80 1.495e+01 1.063e+03 0.014 0.98877   
## citycity\_81 -2.535e-01 1.956e+03 0.000 0.99990   
## citycity\_82 2.443e-01 2.624e+03 0.000 0.99993   
## citycity\_83 1.503e+01 1.063e+03 0.014 0.98872   
## citycity\_84 -5.385e-01 1.649e+03 0.000 0.99974   
## citycity\_89 1.544e+01 1.063e+03 0.015 0.98841   
## citycity\_9 1.657e+01 1.063e+03 0.016 0.98757   
## citycity\_90 1.480e+01 1.063e+03 0.014 0.98889   
## citycity\_91 1.527e+01 1.063e+03 0.014 0.98854   
## citycity\_93 1.659e+01 1.063e+03 0.016 0.98754   
## citycity\_94 -2.100e-02 1.694e+03 0.000 0.99999   
## citycity\_97 1.346e+01 1.063e+03 0.013 0.98990   
## citycity\_98 1.907e-01 1.241e+03 0.000 0.99988   
## citycity\_99 5.112e-02 1.188e+03 0.000 0.99997   
## genderMale 6.565e-02 1.826e-01 0.360 0.71912   
## genderMissing 1.821e-01 1.964e-01 0.927 0.35388   
## genderOther 1.366e-01 5.045e-01 0.271 0.78664   
## relevant\_experienceYes -2.263e-01 1.262e-01 -1.792 0.07308 .   
## enrolled\_universityNo -3.403e-01 1.233e-01 -2.761 0.00577 \*\*   
## enrolled\_universityPart-Time -2.529e-01 2.097e-01 -1.206 0.22785   
## education\_levelHigh School -6.740e-01 2.188e-01 -3.081 0.00207 \*\*   
## education\_levelMasters -1.395e-01 1.193e-01 -1.170 0.24217   
## education\_levelPhd -2.202e-01 3.836e-01 -0.574 0.56584   
## education\_levelPrimary School -9.613e-01 4.309e-01 -2.231 0.02567 \*   
## major\_disciplineSTEM 3.320e-01 1.527e-01 2.174 0.02971 \*   
## experience>20 -7.105e-01 3.047e-01 -2.332 0.01969 \*   
## experience1 -8.226e-02 3.515e-01 -0.234 0.81500   
## experience10 -2.622e-01 3.286e-01 -0.798 0.42485   
## experience11 -4.982e-01 3.536e-01 -1.409 0.15883   
## experience12 -4.259e-01 3.851e-01 -1.106 0.26872   
## experience13 -5.322e-01 4.164e-01 -1.278 0.20121   
## experience14 -3.982e-01 3.749e-01 -1.062 0.28813   
## experience15 -7.311e-01 3.846e-01 -1.901 0.05727 .   
## experience16 -1.148e+00 4.330e-01 -2.651 0.00803 \*\*   
## experience17 -1.375e+00 5.156e-01 -2.667 0.00766 \*\*   
## experience18 -6.482e-01 5.451e-01 -1.189 0.23440   
## experience19 5.682e-03 4.646e-01 0.012 0.99024   
## experience2 -3.157e-01 3.101e-01 -1.018 0.30867   
## experience20 -7.279e-01 9.005e-01 -0.808 0.41891   
## experience3 -2.001e-01 2.944e-01 -0.680 0.49671   
## experience4 -5.171e-01 2.926e-01 -1.767 0.07715 .   
## experience5 -4.538e-01 3.001e-01 -1.512 0.13054   
## experience6 -5.912e-01 3.099e-01 -1.908 0.05641 .   
## experience7 -2.968e-01 3.151e-01 -0.942 0.34635   
## experience8 -3.715e-01 3.343e-01 -1.111 0.26652   
## experience9 -3.823e-01 3.233e-01 -1.182 0.23707   
## company\_size10-49 3.596e-01 2.650e-01 1.357 0.17474   
## company\_size100-500 -1.300e-02 2.591e-01 -0.050 0.95997   
## company\_size1000-4999 -1.877e-01 3.016e-01 -0.622 0.53366   
## company\_size10000+ 1.966e-02 2.677e-01 0.073 0.94146   
## company\_size50-99 2.091e-01 2.503e-01 0.836 0.40330   
## company\_size500-999 -2.826e-01 3.447e-01 -0.820 0.41225   
## company\_size5000-9999 1.179e-01 3.598e-01 0.328 0.74321   
## company\_sizeUnknown 1.280e+00 2.810e-01 4.555 5.24e-06 \*\*\*  
## company\_typeFunded Startup -4.269e-01 3.703e-01 -1.153 0.24901   
## company\_typeNGO -3.929e-01 4.382e-01 -0.897 0.36986   
## company\_typeOther 1.454e-01 3.319e-01 0.438 0.66124   
## company\_typePublic Sector 3.647e-01 3.636e-01 1.003 0.31588   
## company\_typePvt Ltd -1.417e-01 2.946e-01 -0.481 0.63062   
## last\_new\_job1 -1.950e-01 1.532e-01 -1.273 0.20302   
## last\_new\_job2 -3.124e-01 1.780e-01 -1.755 0.07923 .   
## last\_new\_job3 -1.117e-01 2.352e-01 -0.475 0.63474   
## last\_new\_job4 -1.841e-01 2.513e-01 -0.733 0.46378   
## last\_new\_jobnever -1.018e+00 2.068e-01 -4.923 8.53e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4084.9 on 3699 degrees of freedom  
## Residual deviance: 3142.0 on 3535 degrees of freedom  
## AIC: 3472  
##   
## Number of Fisher Scoring iterations: 15