Statistical Learning, Homework #1

Jacopo Manenti, id: 247279

Abstract

Introduction

Import the dataset from **bf.csv**.

```
df <- read.csv("bf.csv")
```

We start looking at the first 7 rows of the dataset, to preminarly explore the data. From this, it is clear the strucutre of data and the presence of missing values **na**. With the help of the function **glimpse** it returns a usefull insight of the dataset structure, expecially regarding to its dimension, which is 139 observations and 10 variables.

```
head(df)
```

```
  breast pregnancy howfed howfedfr partner smokenow smokebf age educat
1 Breast Beginning Breast   Breast Partner       No      No  24     19
2 Breast Beginning Bottle   Breast Partner       No      No  27     18
3 Bottle Beginning Breast   Breast Partner       No      No  39     16
4 Bottle Beginning Breast   Breast Partner      Yes     Yes  29     16
5 Breast Beginning Breast   Breast Partner       No      No  21     21
6 Bottle Beginning Breast   Bottle Partner       No      No  NA     28
     ethnic
1 Non-white
2     White
3     White
4     White
5     White
6     White
```

```r
glimpse(df)
```

```
Rows: 139
Columns: 10
$ breast    <chr> "Breast", "Breast", "Bottle", "Bottle", "Breast", "Bottle", ~
$ pregnancy <chr> "Beginning", "Beginning", "Beginning", "Beginning", "Beginni~
$ howfed    <chr> "Breast", "Bottle", "Breast", "Breast", "Breast", "Breast", ~
$ howfedfr  <chr> "Breast", "Breast", "Breast", "Breast", "Breast", "Bottle", ~
$ partner   <chr> "Partner", "Partner", "Partner", "Partner", "Partner", "Part~
$ smokenow  <chr> "No", "No", "No", "Yes", "No", "No", "No", "No", "Yes", "No"~
$ smokebf   <chr> "No", "No", "No", "Yes", "No", "No", "No", "No", "Yes", "Yes~
$ age       <int> 24, 27, 39, 29, 21, NA, 27, 27, 20, 31, 28, 29, 32, 25, 33, ~
$ educat    <int> 19, 18, 16, 16, 21, 28, 19, 22, 19, 17, 16, 18, 23, 25, 21, ~
$ ethnic    <chr> "Non-white", "White", "White", "White", "White", "White", "N~
```

Withing the variables, **breast** is the response variabile, which represents the class were UK's pregnat women of the study conducted were classified into. The outcome of the study could aid in targeting breastfeeding promotions towards women with a lower probability of choosing. Moreover, despite the categorical nature of the response variable "breast," which delineates various feeding practices for infants including breastfeeding and exclusive bottle feeding, it is imperative to encode it as binary values (0 and 1) to accommodate logistic regression analysis. This coding transforms "breast" into a Bernoulli variable, enabling each category to be represented by a singular binary outcome. Such encoding facilitates logistic regression to model the likelihood of belonging to one feeding category relative to the other.

Before recoding, it is worth noting that the majority of predictors are categorical variables, with only **age** and **educat** being numerical. Since the variable is categorical, we can organize everything into factors with levels for classification purposes.

Furthermore, missing data have been observed, primarily regarding the age of mothers and their educational attainment, particularly among those engaged in breastfeeding or bottle-feeding. The majority of missing values are noted among women of white ethnicity, suggesting potential data collection issues. With this assumption allows us to discrad records containing missing values (nota su unbalanced class potrebbe impattre)

```r
df %>%
  filter(!complete.cases(.)) %>%
  View()# see na records
```

At this stage, we proceed to preprocess the dataset, converting categorical variables into factors and adjusting the response variable accordingly. Following this data preparation step, we will perform basic statistical analyses and generate relevant plots. It is essential to ensure data cleanliness before proceeding with further statistical analysis.

```r
df <- df %>%
    select(names(df)) %>%
  na.omit() %>%
  mutate(breast = recode(breast,
                         Breast = "1",
                         Bottle = "0")) %>%
  mutate_if(is.character,as.factor)

head(df)
```

```
  breast pregnancy howfed howfedfr partner smokenow smokebf age educat
1      1 Beginning Breast   Breast Partner       No      No  24     19
2      1 Beginning Bottle   Breast Partner       No      No  27     18
3      0 Beginning Breast   Breast Partner       No      No  39     16
4      0 Beginning Breast   Breast Partner      Yes     Yes  29     16
5      1 Beginning Breast   Breast Partner       No      No  21     21
7      1 Beginning Breast   Breast Partner       No      No  27     19
     ethnic
1 Non-white
2     White
3     White
4     White
5     White
7 Non-white
```

```r
glimpse(df)
```

```
Rows: 135
Columns: 10
$ breast    <fct> 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, ~
$ pregnancy <fct> Beginning, Beginning, Beginning, Beginning, Beginning, Begin~
$ howfed    <fct> Breast, Bottle, Breast, Breast, Breast, Breast, Breast, Brea~
$ howfedfr  <fct> Breast, Breast, Breast, Breast, Breast, Breast, Breast, Bott~
$ partner   <fct> Partner, Partner, Partner, Partner, Partner, Partner, Partne~
$ smokenow  <fct> No, No, No, Yes, No, No, No, Yes, No, No, No, Yes, No, No, N~
$ smokebf   <fct> No, No, No, Yes, No, No, No, Yes, Yes, No, No, Yes, No, No, ~
$ age       <int> 24, 27, 39, 29, 21, 27, 27, 20, 31, 28, 29, 32, 25, 33, 29, ~
$ educat    <int> 19, 18, 16, 16, 21, 19, 22, 19, 17, 16, 18, 23, 25, 21, 17, ~
$ ethnic    <fct> Non-white, White, White, White, White, Non-white, Non-white,~
```

```
attach(df)
```

Statistical Summary

Looking at the statistical summary of the features can be useful in inspecting the feature distribution and anomalies, if any.

```
summary(df)
```

```
 breast     pregnancy     howfed     howfedfr     partner    smokenow  smokebf
 0:36   Beginning:54   Bottle:58   Bottle:50   Partner:114   No :103   No :84
 1:99   End      :81   Breast:77   Breast:85   Single : 21   Yes: 32   Yes:51



      age            educat           ethnic
 Min.   :17.00   Min.   :14.00   Non-white:58
 1st Qu.:25.00   1st Qu.:16.00   White    :77
 Median :28.00   Median :17.00
 Mean   :28.17   Mean   :18.09
 3rd Qu.:32.00   3rd Qu.:19.00
 Max.   :40.00   Max.   :38.00
```

The dataset analysis reveals several key observations:

1. **Response Variable Imbalance**: The response variable, indicating breastfeeding practices, shows a significant imbalance, with 99 out of 135 instances reporting breastfeeding.
2. **Demographic Insights**: Among the 135 participants, only 21 are single, and merely 32 are current smokers. The dataset comprises patients aged between 17 and 40 years, with an average age of 18 years for discontinuing studies.
3. **Limited Influence of Race**: Race does not appear to exert a substantial influence on the observed variables.

In the following table it is reported the proportion for the imblacence classes

```
prop.table(table(breast))
```

```
breast
        0         1
0.2666667 0.7333333
```

```
prop.table(table(smokenow))
```

```
smokenow
      No      Yes
0.762963 0.237037
```

```
prop.table(table(partner))
```

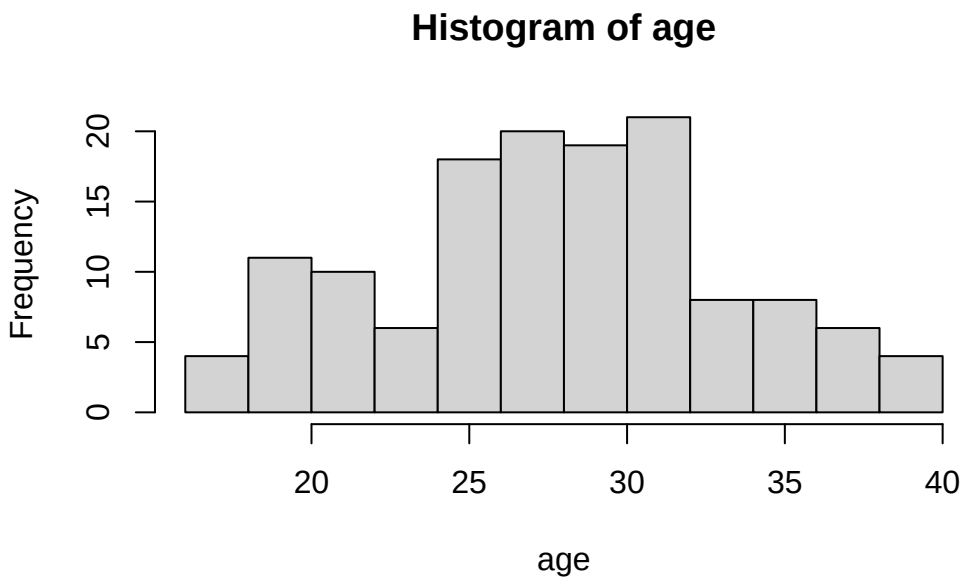
```
partner
  Partner    Single
0.8444444 0.1555556
```

Analisi delle numeric variables

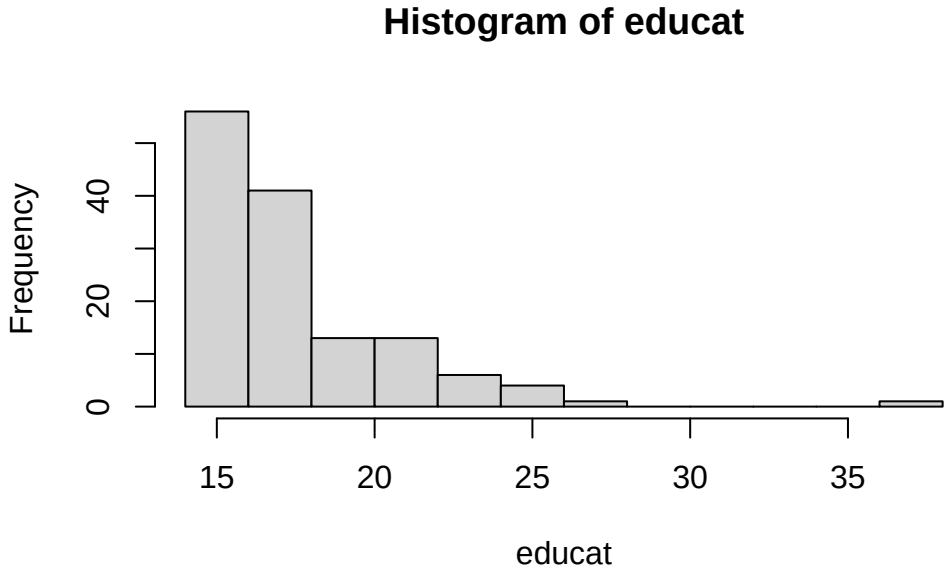
Distribuzione delle numeric variables

First, we'll plot the histograms of numeric variables.

```
hist(age)
```


Histogram of age
Frequency
0
5
10
15
20
20
25
30
35
40
age

```
hist(educat)
```

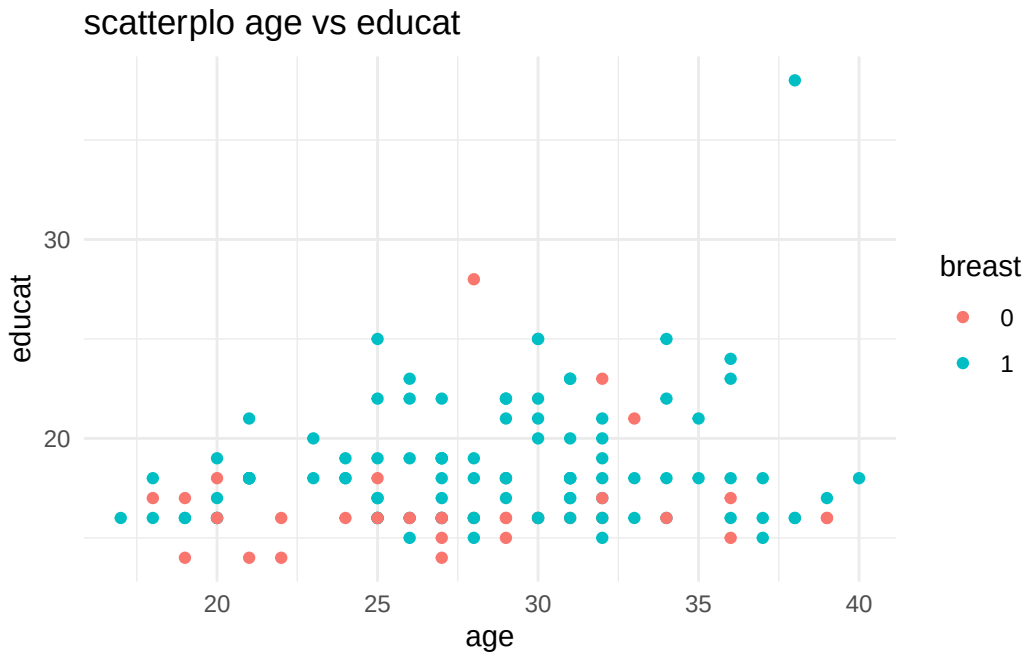

Histogram of educat
Frequency
0
20
40
15
20
25
30
35
educat

Looking at the relationship between numerical and target features

Looking at the numerical features, we explore the relationship between demographic factors and breastfeeding practices among 135 respondents. Figure X provides a visual representation of this investigation, showcasing scatterplots illustrating the age of women (*age*) and the age at which they concluded full-time education (*educat*). Participants who breastfed their babies are distinguished in blue, while those who did not are represented in orange. Notably, a pattern emerges suggesting that individuals who breastfed tended to exhibit a higher educational attainment compared to their counterparts.

Moreover, in Figure X, we present two pairs of boxplots. The first pair illustrates the distribution of age stratified by the binary breast variable, while the second pair demonstrates a similar breakdown for educat. These graphical representations underscore the notable relationship between the predictor variable *educat* and the response variable, emphasizing the significance of educational attainment in influencing breastfeeding practices.

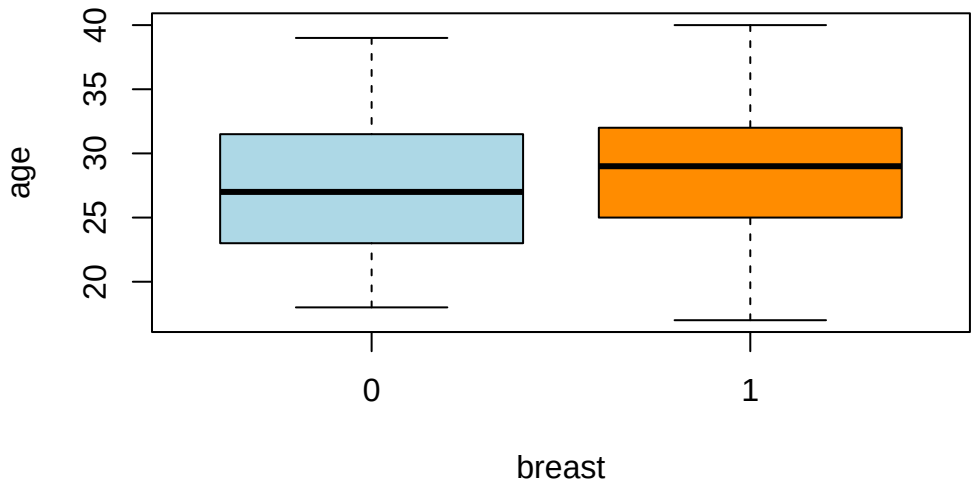
```
# FIGURE 4.1.
df %>%
  ggplot(aes(x= age,
             y= educat,
             colour= breast))+
  geom_point()+
  labs(title="scatterplo age vs educat")+
  theme_minimal()
```



In the scatterplot panel of Figure X, we have depicted the age of women and the age at which they left full-time education (educat) for all 135 respondents. Those who breastfed their baby are indicated in blue, while those who did not are in orange. It appears that individuals who breastfed tended to have a higher educational level than those who did not.

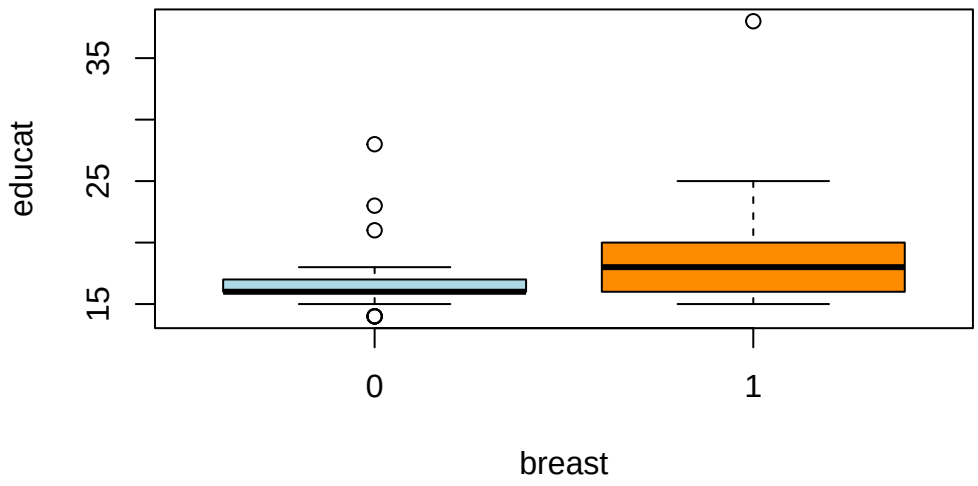
In the above figure (X), two pairs of boxplots are shown. The first shows the distribution of *age* split by the binary *breast* variable; the second is a similar plot for *educat*. It is worth noting that Figure 4.1 displays a pronounced relation- ship between the predictor *educat* and the response.

```
boxplot(age ~ breast, data = df, col = c("lightblue", "darkorange"), xlab = "breast",
ylab = "age")
```

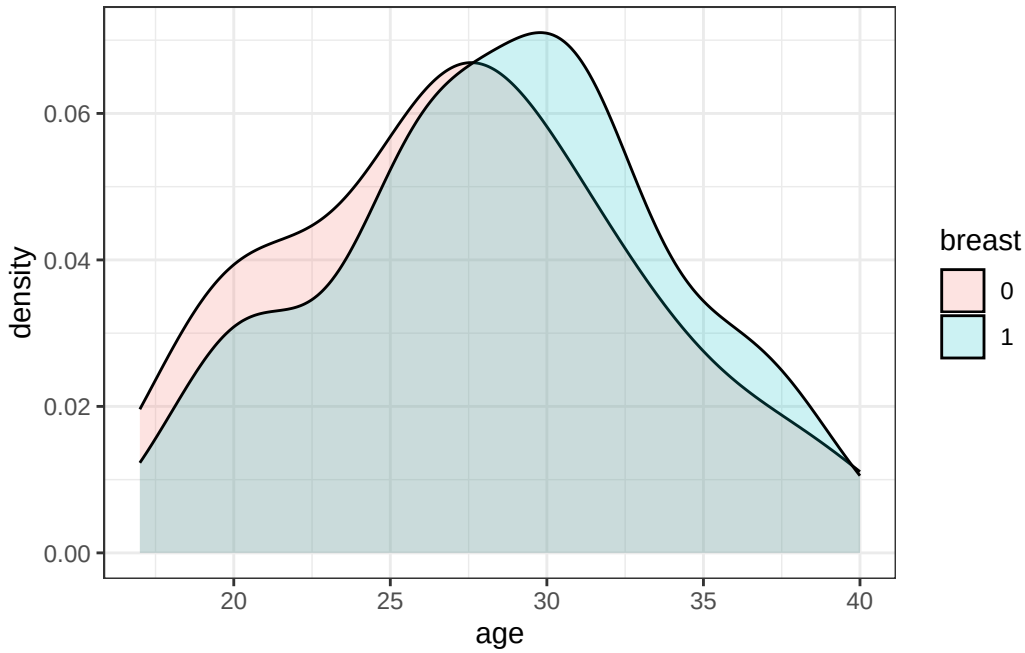


The boxplot tells us that mothers who breast tend to have slightlyhigher age than the one who dont.

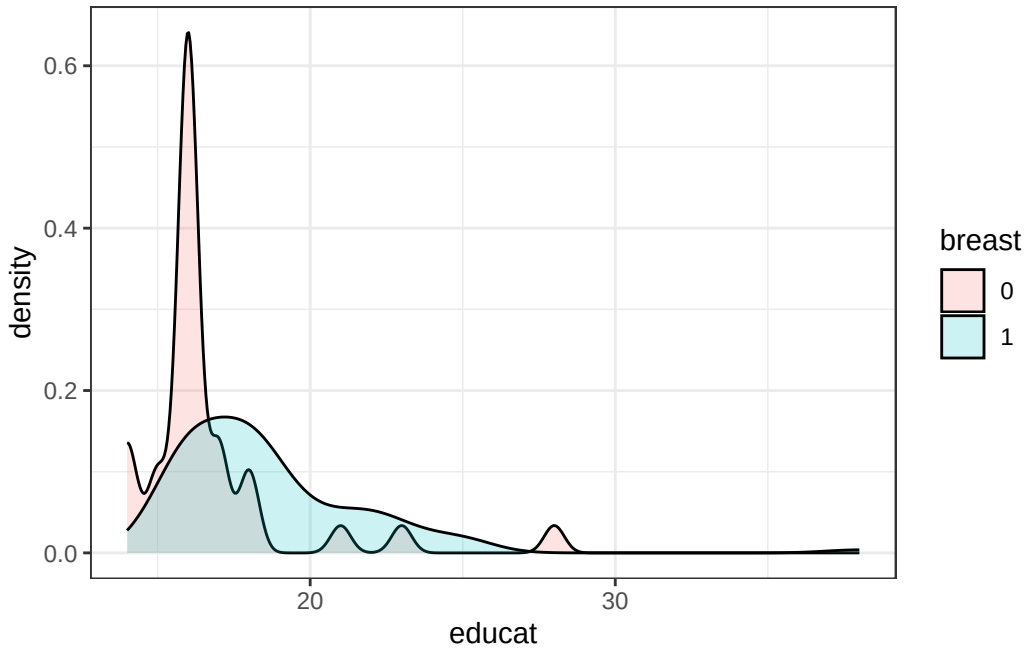
```
boxplot(educat ~ breast, data = df, col = c("lightblue", "darkorange"), xlab = "breast",
ylab = "educat")
```




```
# distribution in respect of breast
df %>%
  ggplot(aes(age, fill= breast))+
  geom_density(alpha = 0.2)+
  theme_bw()
```




```
# distribution in respect of breast
df %>%
  ggplot(aes(educat, fill= breast))+
  geom_density(alpha = 0.2)+
  theme_bw()
```



Looking at distributions of categorical features

Bar plots can be used to view the distribution of categorical variables. This step may be omitted, since we already have an idea of relative frequencies of categorical variables.

Calcola la distribuzione delle categorie nella variabile di risposta.

Verifica la presenza di sbilanciamento di classe, se presente.

Crea grafici (ad esempio, grafici a barre) per visualizzare la distribuzione della variabile di risposta rispetto alle altre variabili.

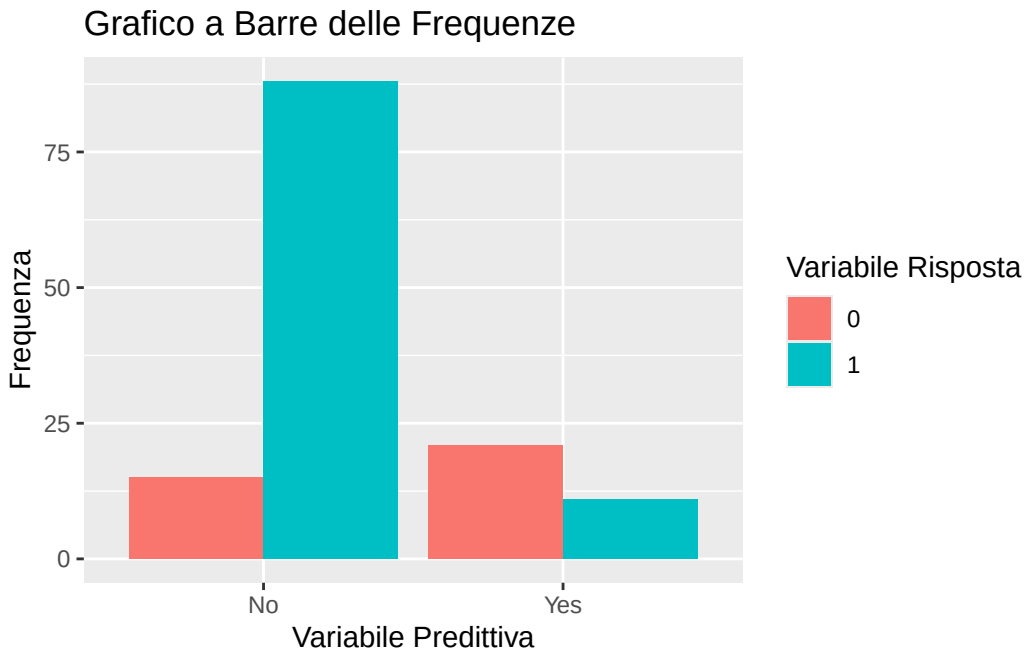
con le numeriche ok

13. Looking at the relationship between categorical and target features

Box plot is one of the best ways to visualize the relationship between a numeric and categorical feature.

```
df %>%
  ggplot( aes(smokenow, fill = breast)) +
  geom_bar(position = "dodge", stat = "count") +
```

```
  labs(x = "Variabile Predittiva", y = "Frequenza", fill = "Variabile Risposta") +
  ggtitle("Grafico a Barre delle Frequenze")
```

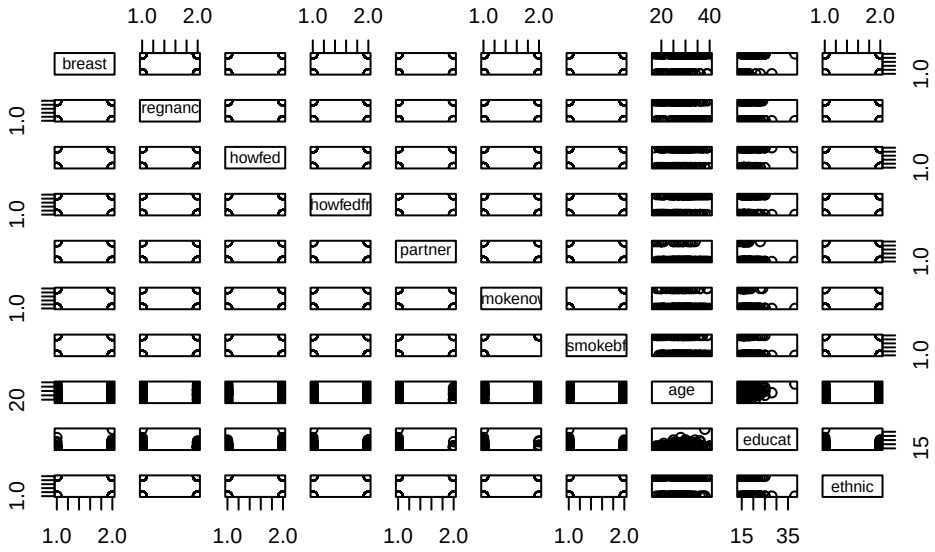


#Looking at pairwise joint distribution of numeric data

The last output for numeric data EDA is a pairwise joint distribution plot.

To generate further insights, we ran the function numeric_eda and added a parameter hue='DayOfWeek'. This allows us to color our pairwise plot by each day of the week.

```
pairs(df)
```



Let's quantify by computing pairwise correlations. Of course if we try that we get an error: there is a qualitative, non-numeric column (Direction) in the dataframe. Remove it and recompute the correlations:

```
df %>%
  select(!is.factor) %>%
    cor()
```

```
              age    educat
age     1.0000000 0.2001478
educat  0.2001478 1.0000000
```

and for categorical data

To analyzed the correlation between categorical variable we can proceed with chisq test

#2,2

```
# select categorical variables
plot_names <- df %>%
  select_if(is.factor) %>%
   select(-breast)
```

```r
# create a grid
# Inizializza una lista per memorizzare i grafici
plots_list <- list()


# plot categorical variables
grid.arrange(
for (i in names(plot_names)){
  tbl <- table(plot_names[[i]],breast)
  br <- barplot(tbl, xlab = "Breadfist method", ylab = paste(i, "counts"), main="test", besi
  plots_list <- br

} ,
nrow = 3,
  top = "Title of the page")
```


test
howfed counts
0
20
40
60
0
1
Breadfist method


test
howfedfr counts
0
20
40
60
0
1
Breadfist method


test
partner counts
0
20
40
60
80
0
1
Breadfist method


test
smokenow counts
0
20
40
60
80
0
1
Breadfist method

Title of the page

A bar plot can suggest a potential correlation between two categorical variables if it exhibits significant differences in bar heights across variable categories. Consistent disparities in bar heights between variable categories may indicate an association between them. Additionally, the presence of a clear trend or pattern in bar heights could imply a correlation between

the variables. However, it is crucial to note that the bar plot alone does not offer definitive evidence of correlation, and further statistical analysis may be required to confirm such an association.

For this reason we proceede comparing correlation between the response variable and the categorical predictors throguh the chi-sqaure test.

The chi-square test assesses the association between categorical variables by comparing observed and expected frequencies. The hypotheses include the null hypothesis of independence and the alternative hypothesis of association. The null hypothesis is rejected if the p-value is below the chosen significance level, indicating a significant association between the variables.

```
# Inizializza un dataframe vuoto per memorizzare i risultati

all_var <- all_chi_s <- all_p_val <- c()

# # Ciclo per calcolare il chi-quadro e memorizzare i risultati nel dataframe


for (i in names(plot_names)){
    tbl <- table(plot_names[[i]],breast)
    chi_sq<- chisq.test(tbl)

    # all in one dataset
    all_var <- append(all_var, i)
    all_chi_s <- append(all_chi_s, chi_sq$statistic)
    all_p_val <-append(all_p_val,round(chi_sq$p.value, 6))
}

test_df <- data.frame(
  Variable = all_var,
  Chi_squared = all_chi_s,
  P_value = all_p_val
)
```

The chi-square test results indicate statistically significant associations between breastfeeding method and variables such as method of feeding ($p = 0.0057$), smoking status ($p < 0.001$), and ethnicity ($p = 0.000016$). Conversely, no significant association is found with pregnancy ($p = 0.968$) and partner status ($p = 0.119$). These findings emphasize the diverse influences shaping breastfeeding practices, with implications for understanding and promoting optimal infant feeding behaviors across different demographic groups.

Question 2

Given the class imbalance observed in the investigation conducted within the UK hospital setting regarding factors influencing pregnant women's decisions on breastfeeding their infants, we split the data into training and test sets while preserving this characteristic, with proportions of 26.67% for bottle feeding and 73.33% for breastfeeding.

```
# class imbalance
prop.table(table(breast))
```

```
breast
        0         1
0.2666667 0.7333333
```

install.packages("caret")

```
# Set the seed for reproducible results
set.seed(5)

# Obtain indices for creating a data partition with the desired proportion of observations be
props <- prop.table(table(breast)) # retrieving the proportion
train_indices <- caret::createDataPartition(breast, p=props[["1"]], list = FALSE)

#create training set
df_train <- df[train_indices,]
y_tr <- breast[train_indices]

# check for training set's class unbalance
prop.table(table(y_tr)) == round(props,2)
```

```
y_tr
   0    1
TRUE TRUE
```

```
#create test set

df_test  <- df[-train_indices, ]
y_ts <- breast[-train_indices]
```

```r
# check for test set's class unbalance
prop.table(table(y_ts)) == round(props,2) # close, probably due to small sample
```

```
y_ts
    0     1
FALSE FALSE
```

Fit the following GLM model:

In our case study, the responses were classified into two categories based on the variable "breast" in the dataset, "breastfeeding", coded as 1,and "bottle" coded as 0.

Instead of directly modeling these response categories, we use a logistic regression that aims to model the probability that an observation belongs to a particular category. For example, we can model the probability of a certain breastfeeding behavior given specific factors $Pr(breast = 1|factors)$. Subsequently, for any individual for whom $p(\text{breast} = 1) > 0.5$, predictions can be made indicating a higher likelihood of engaging in breastfeeding behaviors. In the upcoming section, we'll analyze how performance varies with different threshold values (T) for (p(breast = 1) > T).

Understanding these probabilities can be instrumental in our case study. For instance, it can help target breastfeeding promotion efforts more effectively by identifying women with a higher probability of engaging in breastfeeding or related behaviors. This introduction sets the stage for fitting a generalized linear model (GLM) to the data, allowing us to explore the relationships between various factors and breastfeeding behaviors further. We will employ the following model:

$$\text{logit}(E(\text{breast})) = \beta_0+\beta_1\text{pregnancy}+\beta_2\text{howfed}+\beta_3\text{howfedfr}+\beta_4\text{partner}+\beta_5\text{age}+\beta_6\text{educat}+\beta_7\text{ethnic}+\beta_8\text{smoken}$$

This model will help us analyze and quantify the influence of various factors on breastfeeding behaviors.

```r
mod1 <- glm(breast~., data = df_train, family=binomial)
summary(mod1)
```

```
Call:
glm(formula = breast ~ ., family = binomial, data = df_train)

Coefficients:
```

```
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.464290   3.078009  -0.801  0.42336
pregnancyEnd    0.872990   0.684265   1.276  0.20202
howfedBreast    0.536842   0.660482   0.813  0.41633
howfedfrBreast  1.648941   0.675558   2.441  0.01465 *
partnerSingle  -0.926535   0.758662  -1.221  0.22198
smokenowYes    -3.211827   1.243997  -2.582  0.00983 **
smokebfYes      2.050966   1.215980   1.687  0.09167 .
age            -0.001659   0.058073  -0.029  0.97721
educat          0.183740   0.144596   1.271  0.20383
ethnicWhite    -1.661770   0.774674  -2.145  0.03194 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 116.652  on 99  degrees of freedom
Residual deviance:  73.382  on 90  degrees of freedom
AIC: 93.382

Number of Fisher Scoring iterations: 6
```

Table [inserire numero] presents the estimated coefficients and associated information derived from fitting a logistic regression model with multiple predictors to the *breast* training set. The model aims to predict the probability of *breast* being equal to 1.

Given the reported p-values associated with each β_i, most of them do not fall below the standard confidence level of 0.05. Consequently, we conclude that, apart from current smoking status and ethnicity, there is no significant association between the predictors and the likelihood of breastfeeding.

Furthermore, we observe that $\beta_3 = 1.648941$ (coefficient assosiated with *howfedfrBreast*), $\beta_8 = -3.211827$ (coefficient assosiated with *smokenow*) and $\beta_7 = -1.661770$ (coefficient assosiated with *ethnic*) respectively. This suggests that, while holding other predictors constant, current maternal smoking is linked to a reduction in the probability of breastfeeding. Similarly, being of White ethnicity is associated with a decrease in the likelihood of breastfeeding. Conversely, if the mother's friends breastfeed their children, it is linked to an increase in the probability of breastfeeding. To be more specific, holding other variables constant, smoking is associated with a decrease in the log odds of breastfeeding by 3.51100 units, being White is associated with a decrease in the log odds of breastfeeding by 18.88204 units, whereas having friends breastfeed their children makes log odds of breastfeeding increase by 1.648941 units.

k-nn classifier

In our case study, alongside the logistic regression model, we may explore using the k-nearest neighbors (k-NN) classifier to predict breastfeeding behaviors. This non-parametric method predicts the class for each test observation based on its proximity to other observations in the training dataset.

The k-NN classifier requires selecting a positive integer value $ k $. It identifies the $ k $ nearest points in the training dataset to the test observation $x_0 $, forming the set $ N_0 $. Then, it estimates the conditional probability for each class $j $ using the formula:

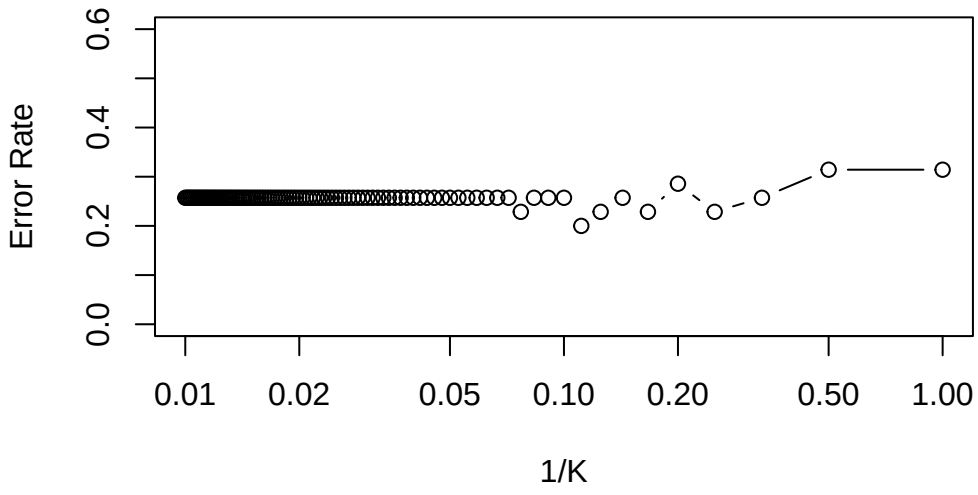
$$\Pr(Y = j|X = x_0) = \frac{1}{k} \sum_{i \in N_0} I(y_i = j)$$

Finally, it classifies the test observation x_0 by assigning it to the class with the highest estimated probability. For instance, if we consider a pregnant woman with specific traits and utilize the k-NN classifier with $ k = 5 $, we identify the 5 most similar cases in our training dataset. Then, we compute the percentage of neighbors belonging to a certain class and assign the observation to the class with the highest probability among the neighbors.

To determine the optimal value of $ k $, we compare a range of values (from 1 to 100) to find the one with the lowest error rate. The error rate indicates the fraction of test observations misclassified, guiding us to select the $ k $ value that minimizes this error for a more accurate and generalizable model.

```
# library
library(class)

# create train and test set for K-NN

x_train <- df_train %>%
    select(-breast) %>%
    mutate_if(is.factor,as.numeric) %>%  # mutate factors into numeri, avoiding coercion warn
    mutate(across(where(is.numeric), ~case_when(. == 1 ~ 0, . == 2 ~ 1, TRUE ~ .))) #recoding

x_test <- df_test %>%
    select(-breast) %>%
    mutate_if(is.factor,as.numeric) %>%  # mutate factors into numeri, avoiding coercion warn
    mutate(across(where(is.numeric), ~case_when(. == 1 ~ 0, . == 2 ~ 1, TRUE ~ .))) # recodin

# define a function to compute the rate
calc_error_rate <- function(predicted.value, true.value) {
mean(true.value != predicted.value)
```

```r
}

# initialize the error rate vector
errors_ts <- c()

# initialize the k values vector
kvec <- c(seq(1:100))

# loop for computing different value of error rate in respect to chaning k'values

for (k in kvec) {
  pred_ts <- knn(x_train, x_test, y_tr, k = k)
  err_ts <- calc_error_rate(pred_ts, y_ts)
  errors_ts <- append(errors_ts, err_ts)
}

plot(x= 1/kvec, y=errors_ts, type = "b", xlim = c(0.01, 1), ylim = c(0, 0.6), log = "x", xlab
ylab = "Error Rate")
```




```r
# select the k value that minimizes the error rate
k_star <- kvec[which.min(errors_ts)]
```

Evaluate the performance of the two methods.

First of all, we take prediction on the logistic model once we have fitted the model. With the k-nn we will be using k =9 for evaluating purposes. After the predictions on the test datasets are made, we use thershold value = 0.5and then test other values

```
# take predictio knn with k = 9
knn_pred <- knn(x_train, x_test, y_tr, k = 9)

## take prediction with fitted logistic regression
mod1_predict <- predict(mod1, df_test, type= "response")

# We want actual labels instead of probabilities (and we are using our standard 0.5 threshol

glm_pred <- rep(0, nrow(df_test))
glm_pred[mod1_predict > 0.5] <- 1
```

Then, we use confusion matrix asa performance metric technique for summarizing the performance of the two classification algorithms. It is a matrix that gives you insight not only into the errors being made by your classifier but more importantly the types of errors that are being made. It reports The number of correct and incorrect predictions are summarized with count values and listed down by each class of predicted and actual value.

- TN: Instances where the mother does not breastfeed, and we correctly predict that she uses bottle feeding.
- TP (True Positive): Cases where the mother breastfeeds, and we correctly predict that she breastfeeds.
- FP (False Positive): Instances where we predict the mother breastfeeds, but she actually uses bottle feeding.
- FN (False Negative): Instances where we predict the mother uses the bottle feeding, but she actually breastfeeds.

```
# confusion matrix for knn where k = 9
table(knn_pred, y_ts)
```

```
        y_ts
knn_pred  0  1
       0  1  0
       1  8 26
```

```
# confusion matrix for logist regression and T = 0.5
table(glm_pred, y_ts)
```

```
        y_ts
glm_pred  0  1
       0  8  4
       1  1 22
```

Both confusion matrix states that the true positves and true negatives are 28 and 30 respectively. But in each we have different values of sensitivity, sensitivity, specificity and precision. Whereas, the level of accurancy it similar

It's evident that both models perform differently in their predictions. The logistic regression model has a higher number of true negatives and true positives, indicating better performance in correctly identifying both negative and positive cases. In addition, the higher number of false positives observed in the k-NN model suggests that it may struggle with classifying negative cases accurately, as it tends to misclassify them as positive. This phenomenon could be attributed to class imbalance.

Persuing further analysis in classification model assessment, key metrics evaluate its ability to distinguish classes effectively. Three common metrics are sensitivity (or recall or true positive rate), specificity (or true negative rate), and false positive rate.

- *Accuracy*: Measures the proportion of correctly classified cases (both true positives and true negatives) among all cases. $\frac{TN+TF}{TN+TF+FP+FN}$
- *sensitivity*= Measures the proportion of true positives among all actual positive cases. $\frac{TP}{(TP+FN)}$
- *specificity* (or *true negative rate*) = Measures the proportion of true negatives among all actual negative cases. $\frac{TN}{(TN+FP)}$
- *false positive rate* = Indicates the proportion of negative cases mistakenly classified as positive among all actual negative cases. $\frac{FP}{(FP+TN)} = 1 - specificity$

```
# K-nn k=7 accurancy
k_acc<- mean(knn_pred == y_ts)
# Logistic regression with T=0.5 accurancy
glm_acc <- mean(glm_pred == y_ts)
# K-nn k=7 sensitivity

k_sens <- sum(knn_pred ==1  & y_ts == 1) / sum(y_ts == 1)

# K-nn k=7 specificity
k_spec <- sum(knn_pred ==0  & y_ts == 0) / sum(y_ts == 0)
```

```r
# K-nn k=7 false positive rate
k_fpr <- 1- k_spec

# Logistic regression with T=0.5 sensitivity
log_sens <- sum(glm_pred ==1  & y_ts == 1) / sum(y_ts == 1)
# Logistic regression with T=0.5 specificity
log_spec <- sum(glm_pred ==0  & y_ts == 0) / sum(y_ts == 0)

# Logistic regression with T=0.5 precision
log_fpr <- 1- log_spec
```

In conclusion, logistic regression exhibits slightly higher accuracy compared to the k-nearest neighbors (k-NN) model, with scores of 0.8571429 and 0.8, respectively. However, a closer analysis reveals that the k-NN model has perfect sensitivity (1), indicating its ability to correctly identify all positive cases. Despite this, when considering specificity, k-NN has a score of 0.22, indicating a high false positive rate (0.77).

In contrast, although logistic regression has lower sensitivity (0.84), it outperforms k-NN in terms of specificity with a score of 0.88, resulting in a significantly lower false positive rate (0.11).

In summary, the k-NN model demonstrates perfect sensitivity, which may be crucial in contexts prioritizing accurate identification of positive cases. However, logistic regression offers higher specificity and a lower false positive rate, suggesting better classification ability for negative cases.

In light of the dataset's imbalance, logistic regression emerges as the superior performer compared to K-nearest neighbors (KNN), particularly due to its adeptness in classifying negative cases, which are significantly fewer. This observation holds particular significance within the context of our study's objective, aimed at optimizing the targeting of breastfeeding promotions towards women less inclined to choose it. Ensuring precise targeting of women with specific characteristics is paramount to optimize resource allocation and, ultimately, enhance the efficacy of the promotional campaign. Consequently, the choice of model should be contingent upon the specific requirements of the problem and the criticality of accurately classifying both positive and negative cases to advance the campaign's success.

discuss possible limitations of the study that you have conducted which may have have an impact on

Draw some overall conclusions about which method may be more suitable in answering the question of interest and discuss possible limitations of the study that you have conducted

which may have have an impact on these conclusions

looking at the counfusion matrix

Given that the class variable is very unbalanced, it is more informative to compare the methods based on the ROC curve. Plot the ROC curves of each model, computing also the areas under the curves (AUCs). Based on the results, which method would you use for future predictions?

The ingredients of the ROC curve are:

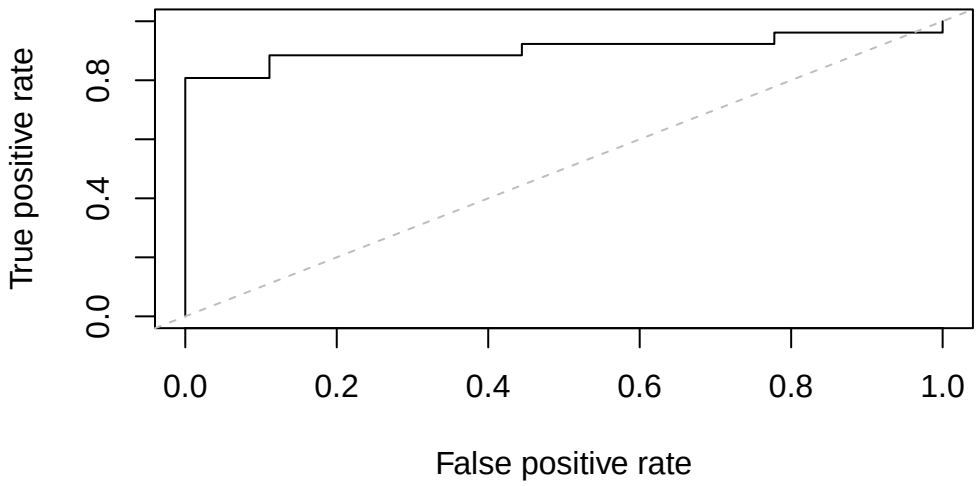
- *sensitivity* (or *recall*, or *true positive rate*) = TP / (TP + FN)
- *specificity* (or *true negative rate*) = TN / (TN + FP)
- *false positive rate* = FP / (FP + TN) = 1 - specificity

where TP (true positives), TN (true negatives), FP (false positives), and FN (false negatives) are the four cells of the confusion matrix: TP and TN stay on the main diagonal; FP and FN on the antidiagonal.

ROC curves are obtained by plotting the sensitivity vs. the false positive rate, which is 1 - specificity.

In our Caravan example, the "Positive" class corresponds to the target variable being "Yes" (people purchase a policy).

```
# first, we need to create a "prediction" object
# I specify ROCR:: to highlight that "prediction()" is part of this library
predob <- ROCR::prediction(mod1_predict, y_ts)
# then we need a "performance" object
perf <- ROCR::performance(predob, "tpr", "fpr")
# now the plot!
plot(perf)
abline(0, 1, col = "gray", lty = 2)
```




```
#

res.auc <- ROCR::performance(predob, "auc")
res.auc@y.values
```

```
[[1]]
[1] 0.9059829
```

One has to decide how much TPR and FPR you want according to the business problem. If you want to increase TPR, your FPR will also increase eventually. So depending on whether you want to detect all the positives (higher TPR) and willing to incur some error in terms of FPR, you decide the optimal cut-off. The threshold value can then be selected according to the requirement. We would want to reduce the FALSE NEGATIVES as much as possible in our case study; hence, hence, we can choose a threshold value that increases our TPR and reduces our FPR. i.e we can choose a threshold of 0.3 and create a confusion matrix and check the accuracy of the model.

```
cost_perf = performance(predob, "cost")
predob@cutoffs[[1]][which.min(cost_perf@y.values[[1]])]
```

```
      54
0.4986297
```

The best cutoff optimal value comes out to be 0.49, this will lead to reduction in the FN which is important. But using the threshold value of 0.49, will lead to increase in FP(false positive rate)

All models are performing equally well, so I would choose one of the simpler

ones (LDA or LR), as they are less variable and more robust than the more

complex model (QDA).