Bank Marketing

Danilo Ferreira de Oliveira

06/04/2021

Contents

1 Introduction

The data utilized in this project is related to direct marketing campaigns of a Portuguese banking institution, and it's available in UCI's Machine Learning Repository. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ("yes") or not ("no") subscribed.

1.1 Dataset description: Input variables

Bank client data:

- age (numeric)
- job: type of job (categorical: *admin.*, *blue-collar*, *entrepreneur*, *housemaid*, *management*, *retired*, *self-employed*, *services*, *student*, *technician*, *unemployed*, *unknown*)
- marital: marital status (categorical: *divorced*, *married*, *single*, *unknown*; note: *divorced* means divorced or widowed)
- education (categorical: *basic.4y*, *basic.6y*, *basic.9y*, *high.school*, *illiterate*, *professional.course*, *university.degree*, *unknown*)
- default: has credit in default? (categorical: *no*, *yes*, *unknown*)
- housing: has housing loan? (categorical: *no*, *yes*, *unknown*)
- loan: has personal loan? (categorical: *no*, *yes*, *unknown*)

Related with the last contact of the current campaign:

- contact: contact communication type (categorical: *cellular*, *telephone*)
- month: last contact month of year (categorical: *jan*, *feb*, *mar*, ..., *nov*, *dec*)
- day_of_week: last contact day of the week (categorical: *mon*, *tue*, *wed*, *thu*, *fri*)
- duration: last contact duration, in seconds (numeric).

Other attributes:

- campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
- pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
- previous: number of contacts performed before this campaign and for this client (numeric)
- poutcome: outcome of the previous marketing campaign (categorical: *failure*, *nonexistent*, *success*)

Social and economic context attributes:

- emp.var.rate: employment variation rate - quarterly indicator (numeric)
- cons.price.idx: consumer price index - monthly indicator (numeric)
- cons.conf.idx: consumer confidence index - monthly indicator (numeric)
- euribor3m: euribor 3 month rate - daily indicator (numeric) Euribor rates are based on the interest rates at which a panel of European banks borrow funds from one another
- nr.employed: number of employees - quarterly indicator (numeric)

Output variable (desired target):

- y - has the client subscribed a term deposit? (binary: *yes*, *no*)

2 Analysis

Firstly, we must import the libraries we are going to use:

```
library(skimr)
library(ggplot2)
library(tidyverse)
library(dplyr)
library(caret)
library(corrplot)
library(pROC)
```

From all of four options available in UCI's Machine Learning Repo Database, we will utilize *bank-additional-full.csv*, which contains all examples (41188) and 20 inputs, ordered by date (from May 2008 to November 2010), very close to the data analyzed in [Moro et al., 2014].

```
bank_full <- read.csv('bank-additional-full.csv', sep = ';')
bank_full %>% as_tibble()
```

```
>>> # A tibble: 41,188 x 21
>>>      age job   marital education default housing loan  contact month day_of_week
>>>    <int> <chr> <chr>   <chr>     <chr>   <chr>   <chr> <chr>   <chr> <chr>
>>>  1    56 hous~ married basic.4y  no      no      no    teleph~ may   mon
>>>  2    57 serv~ married high.sch~ unknown no      no    teleph~ may   mon
>>>  3    37 serv~ married high.sch~ no      yes     no    teleph~ may   mon
>>>  4    40 admi~ married basic.6y  no      no      no    teleph~ may   mon
>>>  5    56 serv~ married high.sch~ no      no      yes   teleph~ may   mon
>>>  6    45 serv~ married basic.9y  unknown no      no    teleph~ may   mon
>>>  7    59 admi~ married professi~ no      no      no    teleph~ may   mon
>>>  8    41 blue~ married unknown   unknown no      no    teleph~ may   mon
>>>  9    24 tech~ single  professi~ no      yes     no    teleph~ may   mon
>>> 10    25 serv~ single  high.sch~ no      yes     no    teleph~ may   mon
>>> # ... with 41,178 more rows, and 11 more variables: duration <int>,
>>> #   campaign <int>, pdays <int>, previous <int>, poutcome <chr>,
>>> #   emp.var.rate <dbl>, cons.price.idx <dbl>, cons.conf.idx <dbl>,
>>> #   euribor3m <dbl>, nr.employed <dbl>, y <chr>
```

There are 12 duplicated rows, which we eliminate by doing the following:

```
idx <- which(duplicated(bank_full)==TRUE)
bank_full <- bank_full[-idx,]
rm(idx)
```

We then check the distribution of people that did and did not subscribe to a term deposit, shown in Figure 1.

```
bank_full %>% ggplot(aes(y)) + geom_bar()
```

2.1 Correlation matrix

We now check the correlation between numeric variables using two different functions/visualizations in R.First we utilize the function `corrplot` and obtain the results presented in Figure 2:

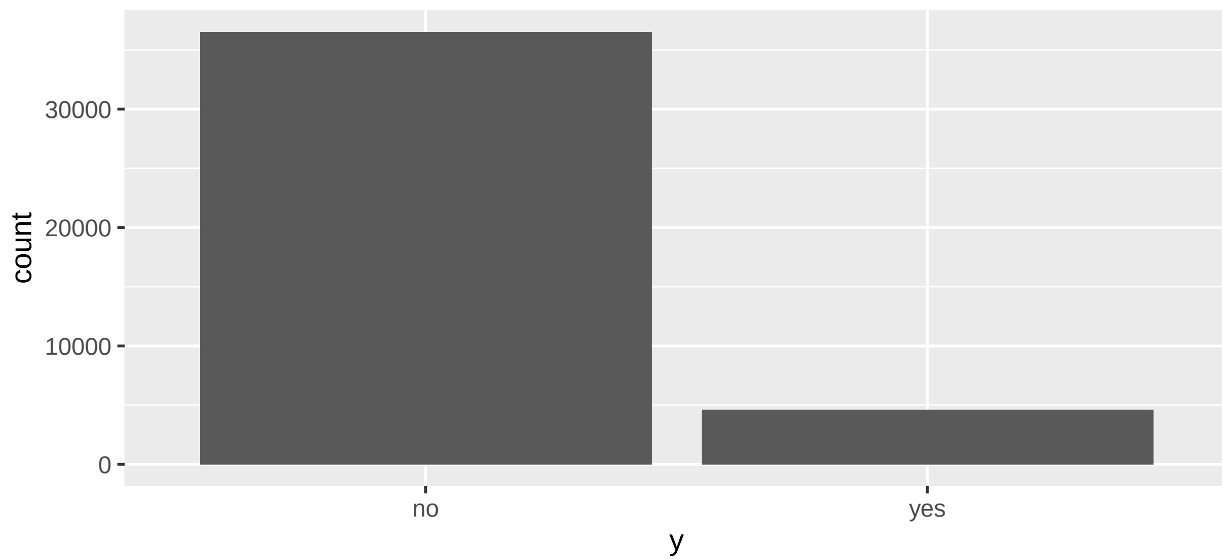

Figure 1: Target value distribution

```
library("Hmisc")
num_bank_full <- bank_full %>% select(age, campaign, pdays, previous, emp.var.rate,
                                       cons.price.idx, cons.conf.idx, euribor3m,
                                       nr.employed)
res <- rcorr(as.matrix(num_bank_full))
corrplot(res$r, type = "lower",tl.col = "black", tl.srt = 45)
```

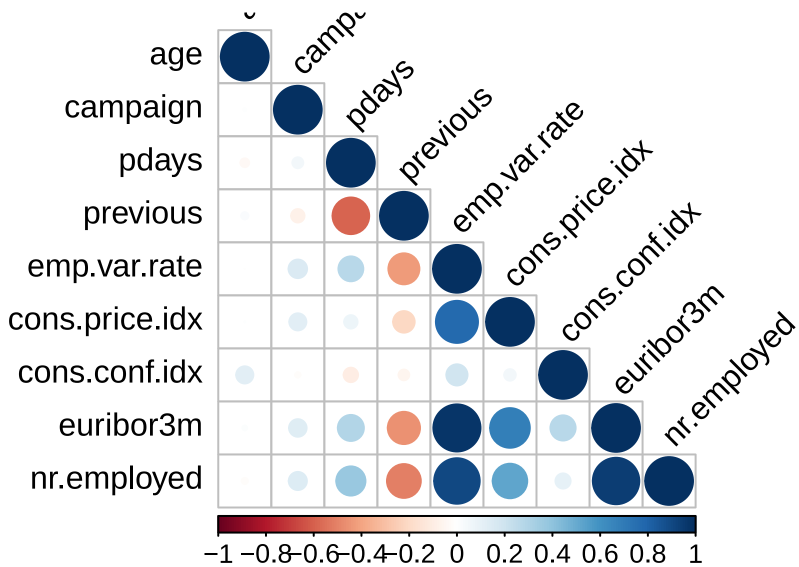

Figure 2: Numeric variables correlation matrix

The second type of visualization is the heat map, which we can see in Figure 3.

```r
heatmap(res$r,symm = TRUE)
```

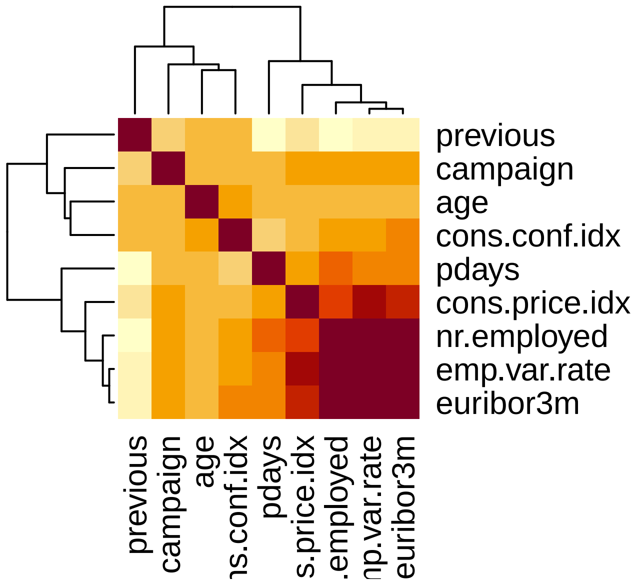

Figure 3: Correlation heat map

Analyzing both graphs, we can see high correlations between variables `emp.var.rate`, `nr.employed` and `euribor3m`. The first two are not only correlated but there is cause, since one is the employment variation rate and the other is the number of employees.

2.2 Variable `duration`

This attribute highly affects the output target (e.g., if duration = 0 then y = *no*). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

```r
bank_full <- bank_full %>% select(-duration)
```

2.3 Age

Analyzing age, we can see that it is very difficult for older people to not subscribe to a term deposit, as seen in Figure .

```r
bank_full %>% ggplot(aes(age)) + facet_grid("y", scales='free') +
  geom_histogram(color='black', bins=35) +
  theme(axis.text.x = element_text(angle=45))
```

2.4 Day of week

The day of the week which occurs the contact does not alter the shape of the distribution for `yes` or `no` values.

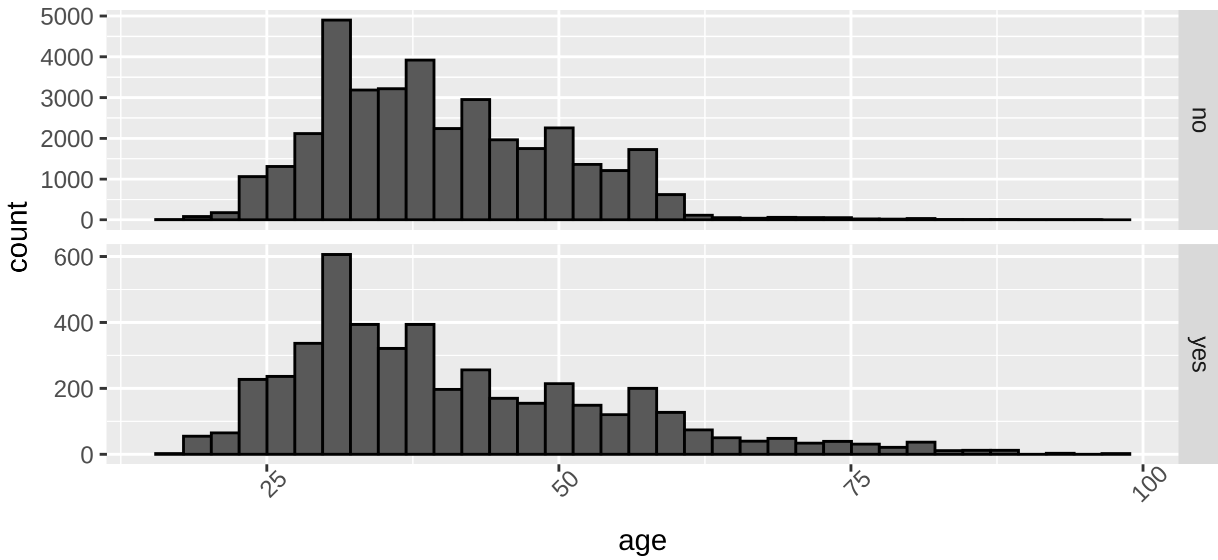

Figure 4: Distribution of ages given the acceptance or not of term deposits

```
bank_full$day_of_week <- factor(bank_full$day_of_week,
                              levels=c("mon", "tue", "wed", "thu", "fri"))
bank_full %>% ggplot(aes(day_of_week)) + facet_grid("y", scales='free') +
  geom_bar() + theme(axis.text.x = element_text(angle=45))
```

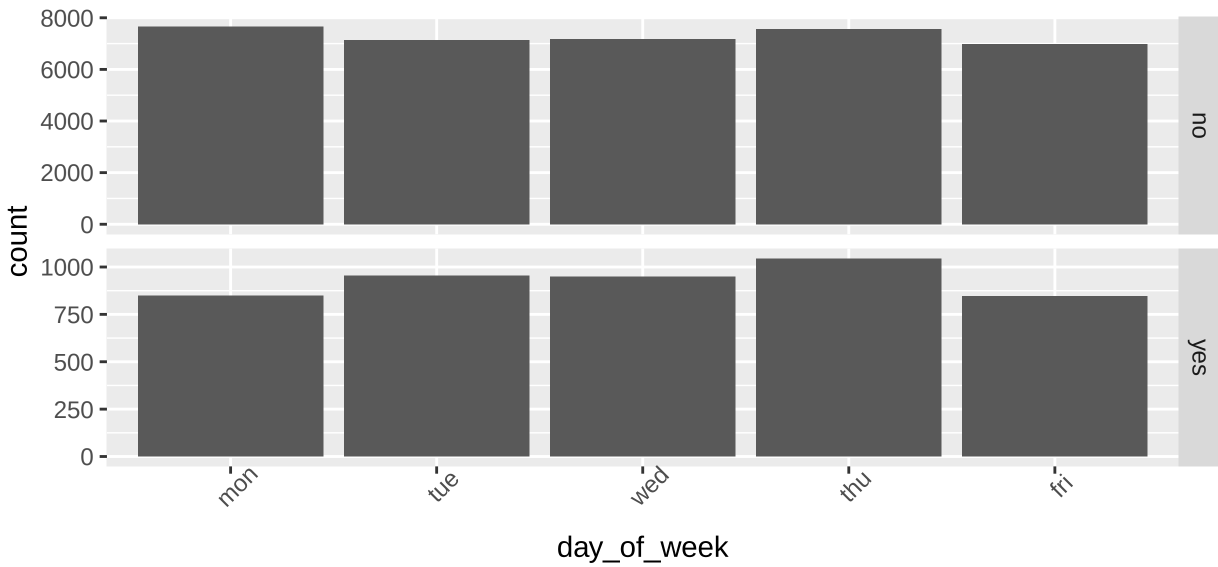

Figure 5: Distribution of contact day of week, given the acceptance or not of term deposits

2.5 Marital status

```
bank_full %>% filter(marital=="unknown") %>% group_by(y) %>% summarize(n())
```

```
>>> # A tibble: 2 x 2
>>>   y     `n()`
>>>   <chr> <int>
>>> 1 no       68
>>> 2 yes      12
```

2.6 Education

```
bank_full %>% ggplot(aes(education)) + facet_grid("y", scales='free') + geom_bar() +
  theme(axis.text.x = element_text(angle=45))
```

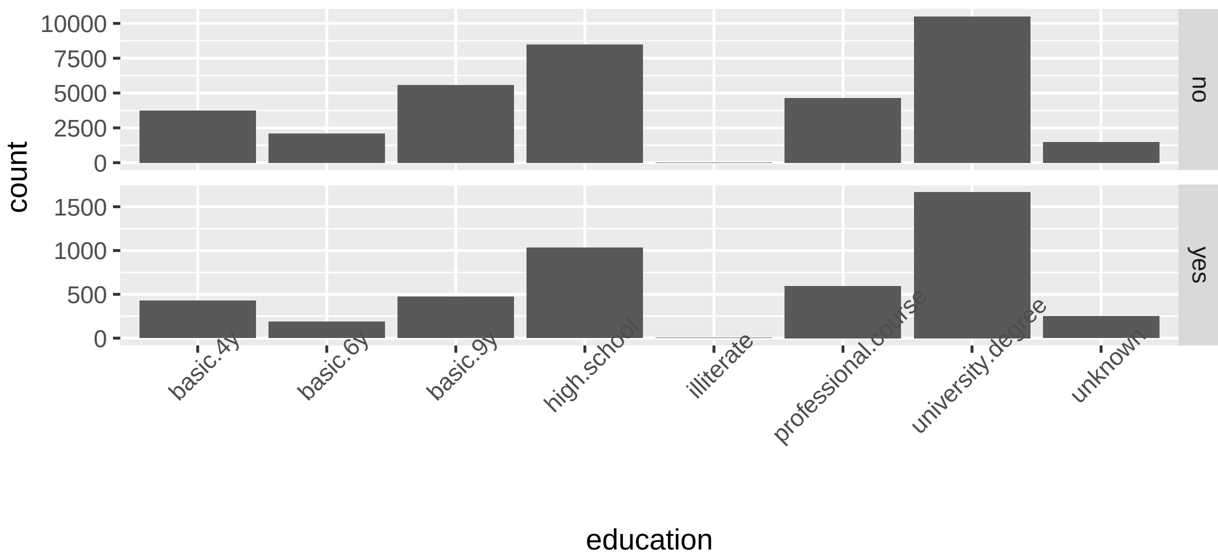

Figure 6: Distribution of education levels given the acceptance or not of term deposits

By analyzing the plot above, we see that there is very little people in this dataset that is considered illiterate, so maybe it is best to include this category into the `unknown` one.

```
bank_full %>% filter(education=="illiterate") %>% group_by(y) %>% summarize(n())
```

```
>>> # A tibble: 2 x 2
>>>   y     `n()`
>>>   <chr> <int>
>>> 1 no       14
>>> 2 yes       4
```

```
bank_full$education[bank_full$education == "illiterate"] <- "unknown"
```

2.7 Default

In the case of this variable, it is possible to see that almost nobody has credit in default, so this is probably a variable that we can discard.

```
bank_full %>% ggplot(aes(default)) + geom_bar()
```

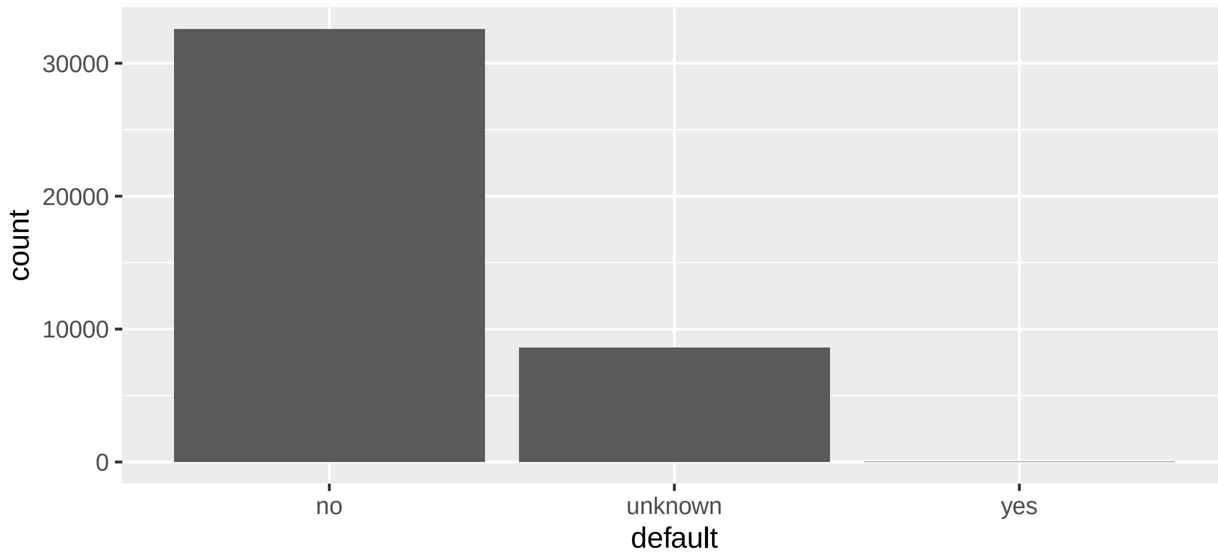

Figure 7: Count of people that have credit in default

```
bank_full[which(bank_full$default=="yes"),] # should we drop it? lots of unknown and little yeses
```

```
>>>       age        job marital           education default housing loan  contact
>>> 21581  48 technician married professional.course     yes      no   no cellular
>>> 21582  48 technician married professional.course     yes     yes   no cellular
>>> 24867  31 unemployed married         high.school     yes      no   no cellular
>>>       month day_of_week campaign pdays previous    poutcome emp.var.rate
>>> 21581   aug         tue        1   999        0 nonexistent          1.4
>>> 21582   aug         tue        1   999        0 nonexistent          1.4
>>> 24867   nov         tue        2   999        1     failure         -0.1
>>>       cons.price.idx cons.conf.idx euribor3m nr.employed  y
>>> 21581         93.444         -36.1     4.963      5228.1 no
>>> 21582         93.444         -36.1     4.963      5228.1 no
>>> 24867         93.200         -42.0     4.153      5195.8 no
```

```
bank_full %>% ggplot(aes(default)) + facet_grid("y", scales='free') + geom_bar() +
  theme(axis.text.x = element_text(angle=45))
```

2.8 pdays

This variable represents of days that passed by after the client was last contacted from a previous campaign. When its value is 999, it means that they were not previously contacted. Let's see the distribution of the ones that were:

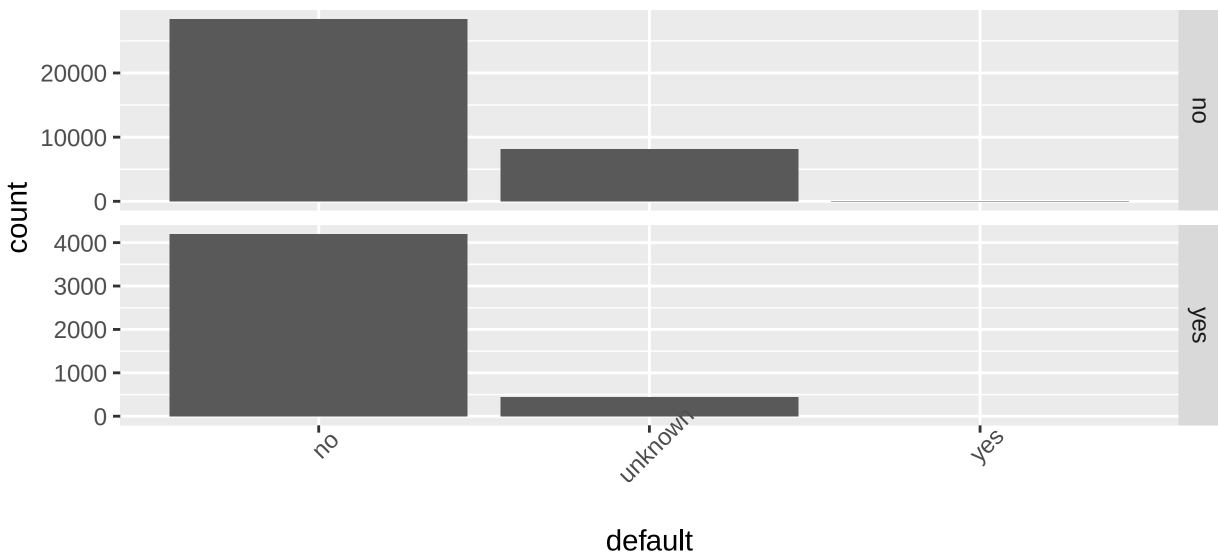

Figure 8: Distribution of people that have credit in default given the acceptance or not of term deposits

```
contacted_before <- bank_full %>% filter(pdays<999) %>% select(pdays)
contacted_before %>% ggplot(aes(pdays)) + geom_histogram(bins=15, color='black')
```

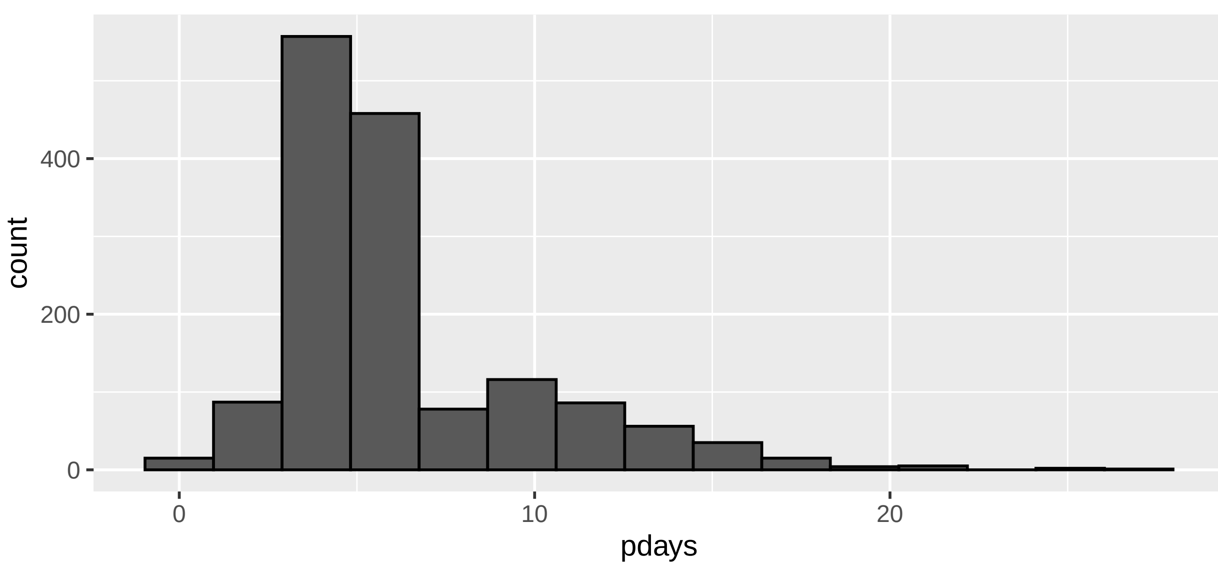

Figure 9: Distribution of days that have passed by since last contact

2.9 previous

Similarly, `previous` represent the number of contacts performed before this campaign and for this client. So we compare both variables:

```
summary(bank_full$pdays) # 999 values
```

```
>>>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
>>>     0.0   999.0   999.0   962.5   999.0   999.0
```

```
bank_full %>% filter(pdays==999) %>% dim() %>% .[1]
```

```
>>> [1] 39661
```

```
bank_full %>% filter(pdays==999) %>% dim() %>% .[1]/dim(bank_full)[1] #percentage
```

```
>>> [1] 0.9632067
```

```
bank_full %>% filter(previous==0) %>% dim() %>% .[1]
```

```
>>> [1] 35551
```

```
bank_full %>% filter(previous==0) %>% dim() %>% .[1]/dim(bank_full)[1]
```

```
>>> [1] 0.8633913
```

2.10 Chi-square test

We use chi-square tests to check if the distribution of the numeric variables is approximately normal. We checked all of them, but with the purpose of making this report shorter, we show only the ones that failed the hypothesis.

```
chisq.test(bank_full$y, bank_full$housing, correct=FALSE)
```

```
>>>
>>>     Pearson's Chi-squared test
>>>
>>> data:  bank_full$y and bank_full$housing
>>> X-squared = 5.7422, df = 2, p-value = 0.05664
```

```
chisq.test(bank_full$y, bank_full$loan, correct=FALSE)
```

```
>>>
>>>     Pearson's Chi-squared test
>>>
>>> data:  bank_full$y and bank_full$loan
>>> X-squared = 1.0993, df = 2, p-value = 0.5772
```

Both these variables have *p-value* higher than 0.05, which was our threshold.

2.11 Selected and treated data for modeling

```
bank_full_2 <- bank_full %>%
  mutate(job = as.factor(job), marital = as.factor(marital),
         education = as.factor(education), contact = as.factor(contact),
         day_of_week = as.factor(day_of_week), poutcome = as.factor(poutcome)) %>%
  select(age, job, marital, education, contact, month, day_of_week, campaign, previous,
         poutcome, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, y)
```

3 Results

```
set.seed(123, sample.kind = "Rounding")
```

```
>>> Warning in set.seed(123, sample.kind = "Rounding"): non-uniform 'Rounding'
>>> sampler used
```

```
index <- createDataPartition(bank_full_2$y, p=0.7, list=FALSE)
train <- bank_full_2[-index,]
test <- bank_full_2[index,]
rm(index)
```

To make training possible, we turn the y variable to factors.

```
train <- train %>% mutate(y = as.factor(y))
```

In this section, we will train three different models: Random Forest, Generalized Bossted Regression Modeling and Logistic Regression.

3.1 Random Forest

```
ctrl <- trainControl(method = "cv",number=5, classProbs = TRUE, summaryFunction = twoClassSummary)
set.seed(123, sample.kind = "Rounding")
fit_rf <- train(y ~ ., data = train, method = 'rf', ntree = 100,
                tuneGrid = data.frame(mtry = seq(1:7)), trControl= ctrl, metric='ROC')

fit_rf$results$mtry[which.max(fit_rf$results$ROC)]
```

```
>>> [1] 6
```

```
rf_pred <- predict(fit_rf, test)

roc_rf <- roc(response=as.ordered(rf_pred), predictor=as.ordered(test$y), auc=TRUE)
roc_rf$auc
```

```
>>> Area under the curve: 0.7406
```

3.2 Generalized Boosted Regression Modeling

```r
set.seed(123, sample.kind = "Rounding")
fit_gbm <- train(y ~ ., data = train,
                 method = "gbm",
                 trControl = ctrl,
                 metric = "ROC",
                 verbose = FALSE)

gbm_pred <- predict(fit_gbm, test)

roc_gbm <- roc(response=as.ordered(gbm_pred), predictor=as.ordered(test$y), auc=TRUE)
roc_gbm$auc
```

```
>>> Area under the curve: 0.7802
```

3.3 Logistic Regression

```r
set.seed(123, sample.kind = "Rounding")
model <- glm(y ~ ., data = train, family = binomial)

probabilities <- model %>% predict(test, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, "yes", "no")

roc_glm <- roc(response=as.ordered(predicted.classes), predictor=as.ordered(test$y), auc=TRUE)
roc_glm$auc
```

```
>>> Area under the curve: 0.79
```

4 Conclusion

mama