Analysis and Prediction of subscriptions to a bank term deposit

From data cleaning to model development: How to deal with unbalanced data sets in machine learning using SMOTE technique

In [ ]:
```
library(ggplot2)
library(readxl)
library(tidyverse)
library(caret)
library(corrplot)
library(pROC)
library(car)
library(stargazer)
library(GGally)
library(arsenal)
library(arm)
library(rpart)
library(randomForest)
```

In [ ]:
```
#open the data set and save it as "term"
term <- read_excel("/Users/jaimerd/Desktop/Term-deposit-subscription/term
```

Data cleaning and feature engineering

In [ ]:
```
#convert categorical variables into factors
term <- term %>% mutate_if(is.character,as.factor)

#numeric variables
data_numeric <- term[sapply(term, is.numeric)]
```

Outliers in numeric variables

In [ ]:
```
#age
term %>% count(age>100)
term$age[term$age >100] <- NA
summary(term$age)

#campaign ()
term %>% count(campaign>33)
term$campaign[term$campaign > 33] <- NA
summary(term$campaign)

#The same process was applied to all numerical variables, yet no outliers
```

A tibble: 2 x 2

age > 100	n
<lgl>	<int>
FALSE	40967
TRUE	2

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  17.00   32.00   38.00   40.03   47.00   95.00       2 
```

A tibble: 2 x 2

campaign > 33	n
<lgl>	<int>
FALSE	40951
TRUE	18

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   1.00    1.00    2.00    2.55    3.00   33.00      18 
```

Categorical variables

In [ ]:
```
#month
levels(term$month)[levels(term$month) == "july"] <- "jul"

#day of the week
levels(term$day_of_week)[levels(term$day_of_week) == "tues"] <- "tue"


##missing values = converted them to the level with the highest frequency
term$marital_status[is.na(term$marital_status)] <- 'married'
```

In [ ]:
```
#omit missing values
term <- na.omit(term)
```

Correlations and Descriptive Analysis

Statistic Summary by comparing all variables with the target variable (subscribed). The p-value for numerical variables was calculated using ANOVA, while for categorical variables, the chi-squared test (chisq.test) was employed.. Taking <0.01 as the reference value for statistical significance.

In [ ]:
```
configuration <- tableby.control(
  test = T,
  total = FALSE,
  numeric.test = "anova", cat.test = "chisq",
  numeric.stats = c("meansd"),
  cat.stats = c("countpct"))
```

```
descriptive_stats <- tableby(subscribed ~ .,
                    data = term,
                    control = configuration)

summary(descriptive_stats, title = "Descriptive Statistic", text= TRUE)
```

```
Table: Descriptive Statistic

|                        |     no (N=36316)      |     yes (N=4633)      |
p value|
|:-----------------------|:---------------------:|:---------------------:|-
------:|
|ID                      |                       |                       |
< 0.001|
|-  Mean (SD)            | 19249.111 (11381.580) | 30197.602 (10708.436) |
|
|age                     |                       |                       |
< 0.001|
|-  Mean (SD)            |    39.918 (9.906)     |    40.888 (13.792)    |
|
|occupation              |                       |                       |
< 0.001|
|-   admin.              |     8987 (24.7%)      |     1351 (29.2%)      |
|
|-   blue-collar         |     8593 (23.7%)      |      638 (13.8%)      |
|
|-   entrepreneur        |      1331 (3.7%)      |      123 (2.7%)       |
|
|-   housemaid           |      954 (2.6%)       |      106 (2.3%)       |
|
|-   management          |      2585 (7.1%)      |      328 (7.1%)       |
|
|-   retired             |      1281 (3.5%)      |      431 (9.3%)       |
|
|-   self-employed       |      1266 (3.5%)      |      149 (3.2%)       |
|
|-   services            |     3635 (10.0%)      |      323 (7.0%)       |
|
|-   student             |      600 (1.7%)       |      275 (5.9%)       |
|
|-   technician          |     5930 (16.3%)      |      728 (15.7%)      |
|
|-   unemployed          |      863 (2.4%)       |      144 (3.1%)       |
|
|-   unknown             |      291 (0.8%)       |       37 (0.8%)       |
|
|marital_status          |                       |                       |
< 0.001|
|-   divorced            |     4122 (11.4%)      |      475 (10.3%)      |
|
|-   married             |     22268 (61.3%)     |     2531 (54.6%)      |
|
```

```
|-  single                |     9878 (27.2%)       |     1618 (34.9%)       |
|
|-  unknown               |       48 (0.1%)        |        9 (0.2%)        |
|
|education_level          |                        |                        |
< 0.001|
|-  basic.4y              |     3729 (10.3%)       |      425 (9.2%)        |
|
|-  basic.6y              |      2099 (5.8%)       |      188 (4.1%)        |
|
|-  basic.9y              |     5563 (15.3%)       |      473 (10.2%)       |
|
|-  high.school           |     8432 (23.2%)       |     1031 (22.3%)       |
|
|-  illiterate            |       14 (0.0%)        |        4 (0.1%)        |
|
|-  professional.course   |     4615 (12.7%)       |      593 (12.8%)       |
|
|-  university.degree     |    10387 (28.6%)       |     1668 (36.0%)       |
|
|-  unknown               |      1477 (4.1%)       |      251 (5.4%)        |
|
|credit_default           |                        |                        |
< 0.001|
|-  no                    |    28217 (77.7%)       |     4194 (90.5%)       |
|
|-  unknown               |     8098 (22.3%)       |      439 (9.5%)        |
|
|-  yes                   |        1 (0.0%)        |        0 (0.0%)        |
|
|housing_loan             |                        |                        |
0.057|
|-  no                    |    16501 (45.4%)       |     2025 (43.7%)       |
|
|-  unknown               |      879 (2.4%)        |      106 (2.3%)        |
|
|-  yes                   |    18936 (52.1%)       |     2502 (54.0%)       |
|
|personal_loan            |                        |                        |
0.614|
|-  no                    |    29925 (82.4%)       |     3844 (83.0%)       |
|
|-  unknown               |      879 (2.4%)        |      106 (2.3%)        |
|
|-  yes                   |     5512 (15.2%)       |      683 (14.7%)       |
|
|contact_method           |                        |                        |
< 0.001|
|-  cellular              |    22075 (60.8%)       |     3846 (83.0%)       |
|
|-  telephone             |    14241 (39.2%)       |      787 (17.0%)       |
|
|month                    |                        |                        |
```

```
< 0.001|
|-  apr                  |     2093 (5.8%)      |     539 (11.6%)      |
|
|-  aug                  |    5309 (14.6%)      |     650 (14.0%)      |
|
|-  dec                  |      93 (0.3%)       |      89 (1.9%)       |
|
|-  jul                  |    6516 (17.9%)      |     649 (14.0%)      |
|
|-  jun                  |    4755 (13.1%)      |     559 (12.1%)      |
|
|-  mar                  |     270 (0.7%)       |     276 (6.0%)       |
|
|-  may                  |    12878 (35.5%)     |     886 (19.1%)      |
|
|-  nov                  |    3685 (10.1%)      |     416 (9.0%)       |
|
|-  oct                  |     403 (1.1%)       |     313 (6.8%)       |
|
|-  sep                  |     314 (0.9%)       |     256 (5.5%)       |
|
|day_of_week             |                      |                      |
< 0.001|
|-  fri                  |    6978 (19.2%)      |     844 (18.2%)      |
|
|-  mon                  |    7637 (21.0%)      |     847 (18.3%)      |
|
|-  thu                  |    7573 (20.9%)      |    1045 (22.6%)      |
|
|-  tue                  |    6945 (19.1%)      |     948 (20.5%)      |
|
|-  wed                  |    7183 (19.8%)      |     949 (20.5%)      |
|
|contact_duration        |                      |                      |
< 0.001|
|-  Mean (SD)            |  220.970 (207.269)   |  553.025 (401.286)   |
|
|campaign                |                      |                      |
< 0.001|
|-  Mean (SD)            |    2.613 (2.757)     |    2.050 (1.664)     |
|
|pdays                   |                      |                      |
< 0.001|
|-  Mean (SD)            |  984.019 (121.036)   |  791.938 (403.476)   |
|
|previous_contacts       |                      |                      |
< 0.001|
|-  Mean (SD)            |    0.133 (0.410)     |    0.493 (0.861)     |
|
|poutcome                |                      |                      |
< 0.001|
|-  failure              |    3647 (10.0%)      |     605 (13.1%)      |
|
```

- nonexistent	32190 (88.6%)	3135 (67.7%)	
- success	479 (1.3%)	893 (19.3%)	
emp_var_rate			< 0.001
- Mean (SD)	0.242 (1.485)	-1.235 (1.622)	
cons_price_idx			< 0.001
- Mean (SD)	93.604 (0.561)	93.355 (0.677)	
cons_conf_idx			< 0.001
- Mean (SD)	-40.619 (4.391)	-39.799 (6.137)	
euribor_3m			< 0.001
- Mean (SD)	3.804 (1.641)	2.121 (1.741)	
n_employed			< 0.001
- Mean (SD)	5175.840 (64.647)	5095.006 (87.515)	

Statistical significance is found in variables such as age, occupation, marital status, education level, and credit default. However, variables like 'housing_loan' and 'personal_loan' do not show statistical significance. This finding contradicts the assertions of Colaianni et al. (2016), who suggested that loan status is a major factor influencing subscription decisions. When analysing the occupations of the subscribers, a notable majority fall into the category of 'Admin', with 29.2% (Figure 1). Regarding the marital status, a significant portion of subscribers were married, accounting for 54.6% (Figure 2), which corroborates H2. Finally, the majority of subscribers, 90.5%, had no credit default history (Figure 4). The presence of a default often indicates financial instability or a history of poor credit management, which could reduce a person's willingness to commit money to a term deposit.

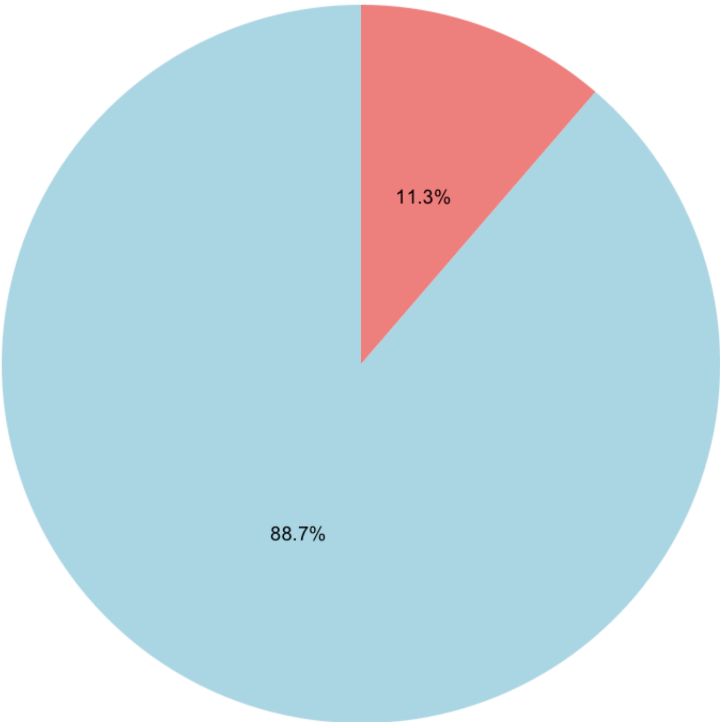
The analysis reveals that all variables pertaining to the last customer contact during the marketing campaign, as well as other variables, exhibit statistical significance. Focusing on the method of contact, the data indicates that most customers, 60.8% of non-subscribers and 83.1% of subscribers, were contacted via cellular (Figure 8). Additionally, a significant proportion of subscribers were successfully contacted in May (19.1%), August (14.0%), and July (14.0%) (Figure 6). The data also demonstrates that subscribers, on average, had fewer contacts (mean of 2.05) compared to non-subscribers (mean of 2.613) (Figure 7). Finally, by examining the results of the previous campaigns ("poutcome"), it is found that most of new subscribers (67.6%) in this campaign were customers who had not been contacted in previous campaigns.

All of the economic and social variables demonstrate statistical significance. It is observed that subscribers are more likely during periods of declining employment rates. Regarding the consumer price index (cons_price_idx), subscribers had an average index of 93.354, slightly lower than the average of 93.604 for non-subscribers. In terms of the consumer confidence index (cons_conf_idx), subscribers were exposed to a higher confidence index during the campaign, aligning with H3. Finally with respect to the 3-month Euribor rate (euribor_3m), subscribing customers had an average rate of 2.117, which is lower than the average rate of 3.804 for non-subscribers.

Graphs

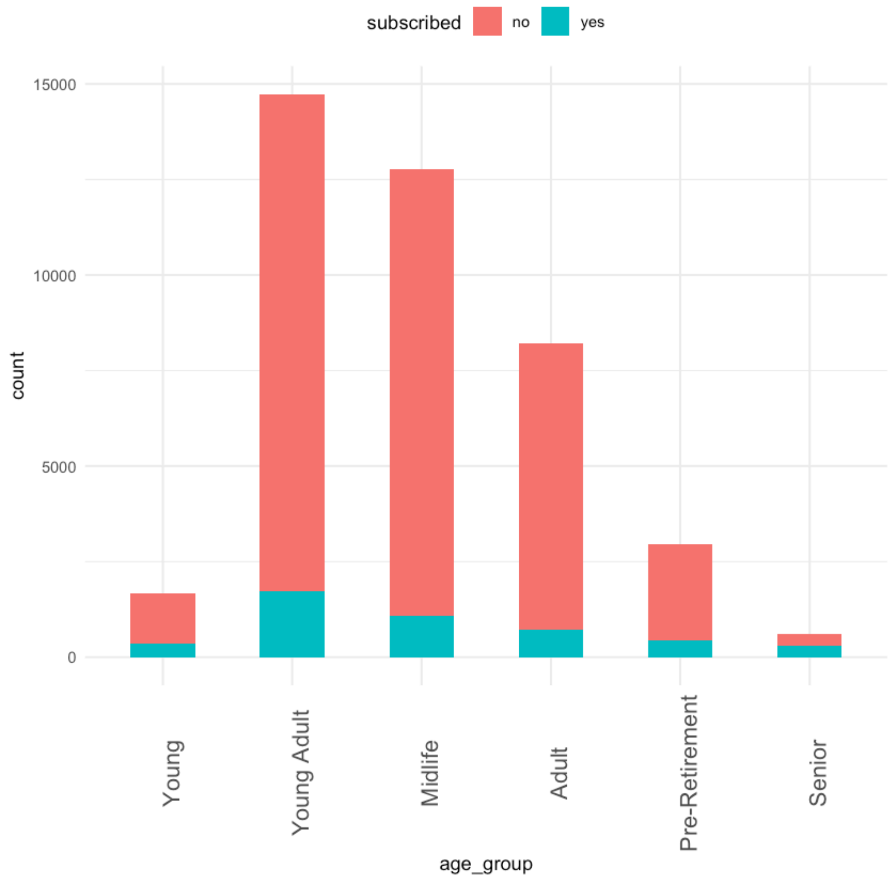
In [ ]:

```
#graph comparing subscribed yes vs no
#calculate the percentage of "yes" "no". 1. A table is created to see the
per_subscribed <- data.frame(prop.table(table(term$subscribed)) * 100)
ggplot(per_subscribed, aes(x = "", y = Freq, fill = Var1)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) +
  theme_void() +
  scale_fill_manual(values = c("lightblue", "lightcoral")) +
  geom_text(aes(label = sprintf("%.1f%%", Freq)), position = position_sta
  theme(legend.position = "none")
```



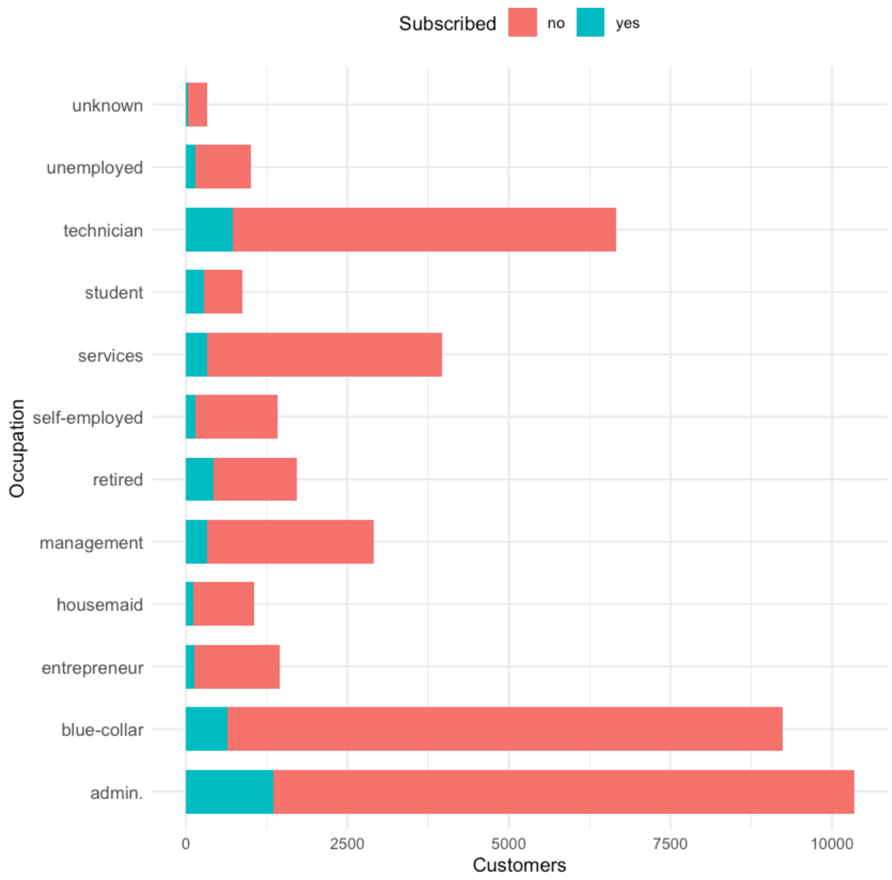
Some demographic and time variables

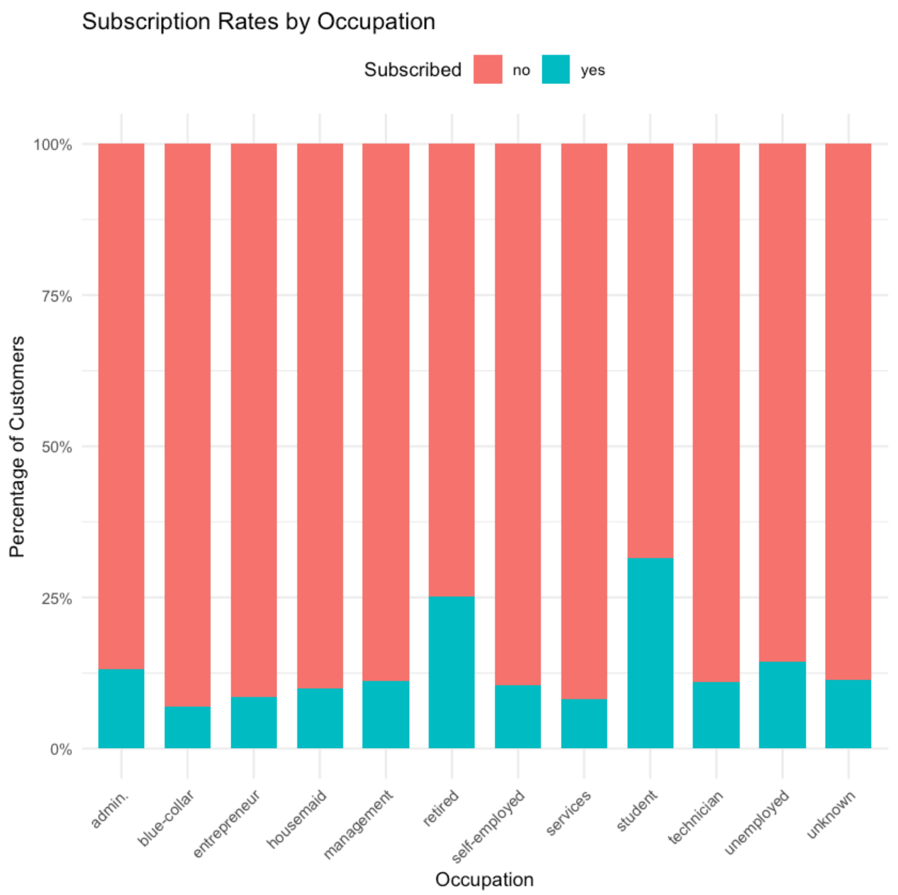
In [ ]:
```
#Categorize ages into groups and create a bar plot of subscriptions by ag
term$age_group <- cut(term$age, breaks = c(0, 25, 35, 45, 55, 65, Inf), l
ggplot(term, aes(x=age_group, fill = subscribed)) +
  geom_bar(position= "stack", width = 0.5) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, size=13))+
  theme(legend.position = "top")
```



In [ ]:
```
#bar plot showing the occupation
ggplot(term, aes(x = occupation, fill = subscribed)) +
  geom_bar(position = "stack", width = 0.7) +
  labs(
       x = "Occupation",
       y = "Customers",
       fill = "Subscribed") +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 10),legend.position = "top") +
  coord_flip()

  #Subscription Rates by occupation
ggplot(term, aes(x = occupation, fill = subscribed)) +
  geom_bar(position = "fill", width = 0.7) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(
    x = "Occupation",
    y = "Percentage of Customers",
    fill = "Subscribed",
```

```
    title = "Subscription Rates by Occupation"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "top")
```



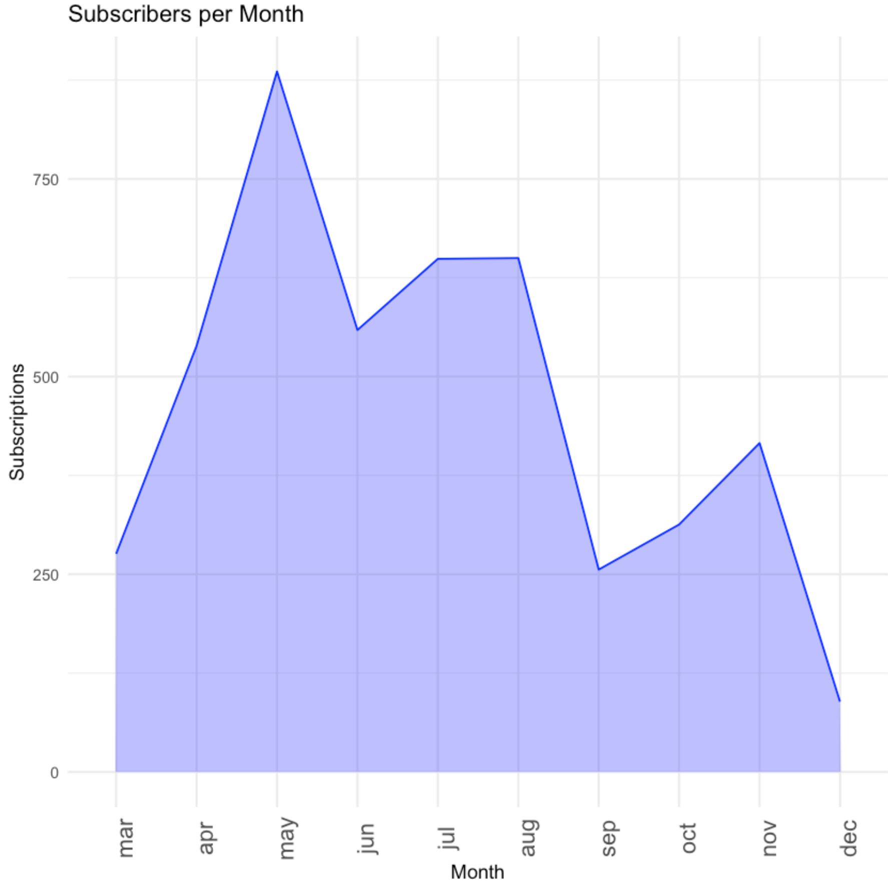


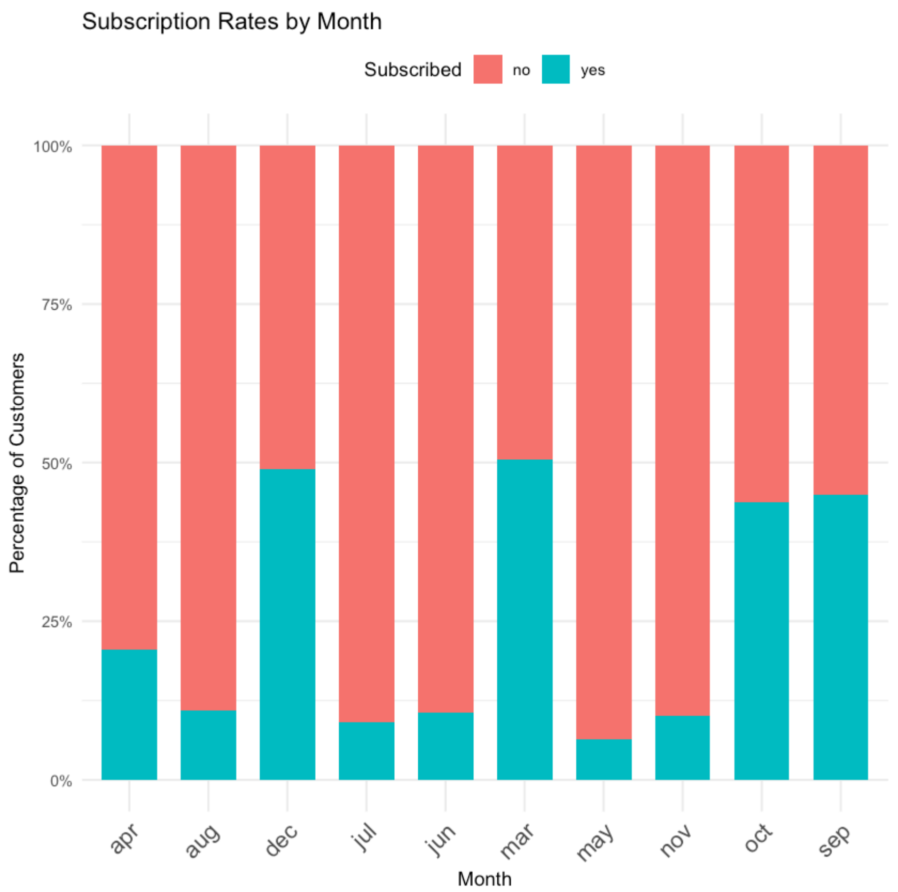
In [ ]:

```
#subscriptions per month
#Create a data frame with the number of subscribers and group them by mon
month_subs <- term %>%
  filter(subscribed == "yes") %>%
  mutate(month = factor(month, levels = c("mar", "apr", "may", "jun", "ju
  group_by(month) %>%
  summarise(subscriptions = n())

#line graph with subscribers per month
ggplot(month_subs, aes(x = month, y = subscriptions, group=1)) +
  geom_area(fill = "blue", alpha = 0.3) +
  geom_line(color = "blue", size =0.5) +
  labs(title = "Subscribers per Month",
       x = "Month",
       y = "Subscriptions") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90 ,size = 13))

#bar graph with the subscription rates by month
ggplot(term, aes(x = month, fill = subscribed)) +
  geom_bar(position = "fill", width = 0.7) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(
    x = "Month",
    y = "Percentage of Customers",
    fill = "Subscribed",
    title = "Subscription Rates by Month"
  ) +
```

```
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 13),lege
```

```
Warning message:
"Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead."
```

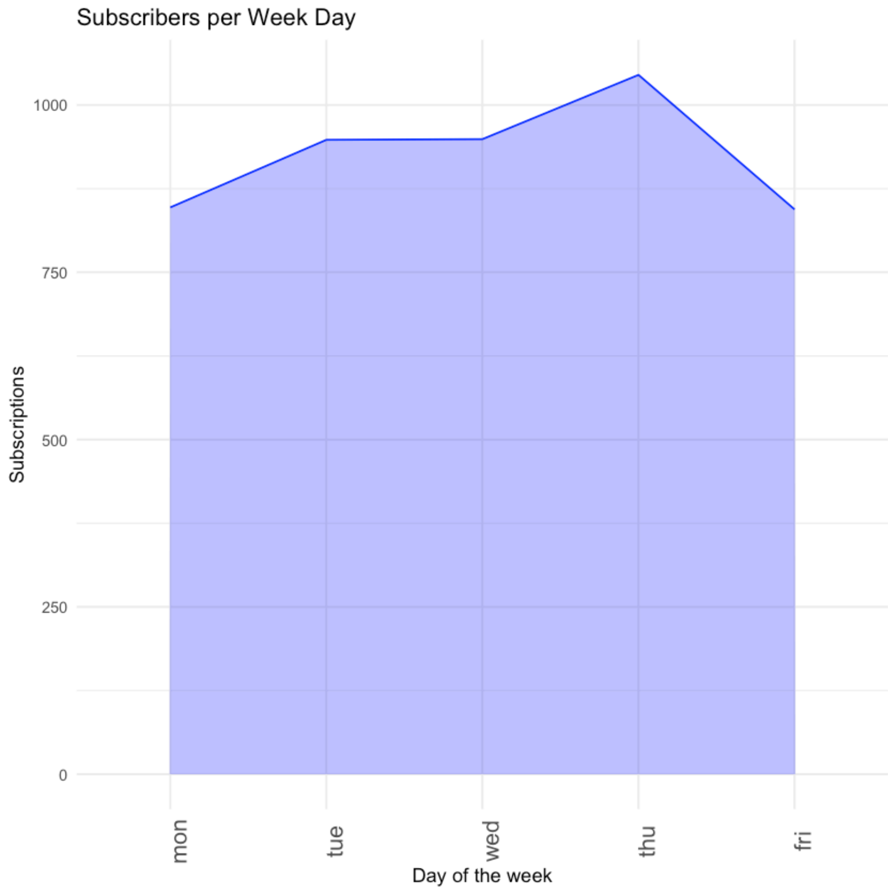


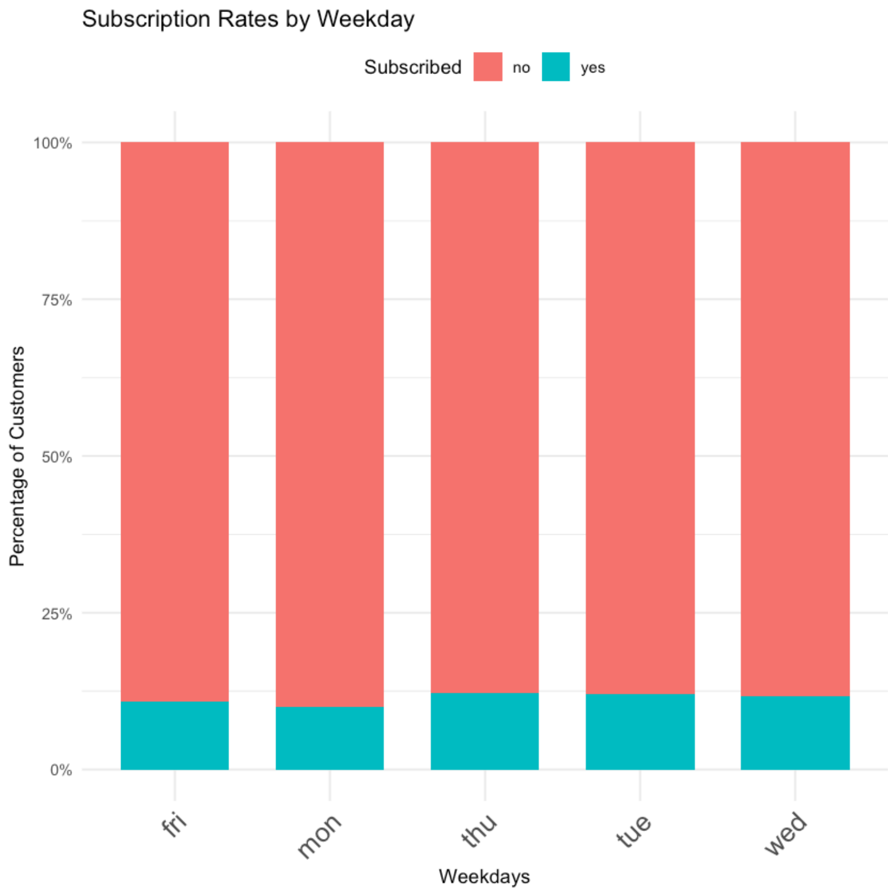


In [ ]:

```
#subscriptions per weekday
#Create a data frame with the number of subscribers and group them by wee
week_subs <- term %>%
  filter(subscribed == "yes") %>%
  mutate(day_of_week = factor(day_of_week, levels = c("mon", "tue", "wed"
  group_by(day_of_week) %>%
  summarise(subscriptions = n())

#line graph with subscribers per weekday
ggplot(week_subs, aes(x = day_of_week, y = subscriptions, group=1)) +
  geom_area(fill = "blue", alpha = 0.3) +
  geom_line(color = "blue", size =0.5) +
  labs(title = "Subscribers per Week Day",
       x = "Day of the week",
       y = "Subscriptions") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, size = 13))

#Subscription Rates by Weekday
ggplot(term, aes(x = day_of_week, fill = subscribed)) +
  geom_bar(position = "fill", width = 0.7) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(
    x = "Weekdays",
    y = "Percentage of Customers",
    fill = "Subscribed",
    title = "Subscription Rates by Weekday"
  ) +
```

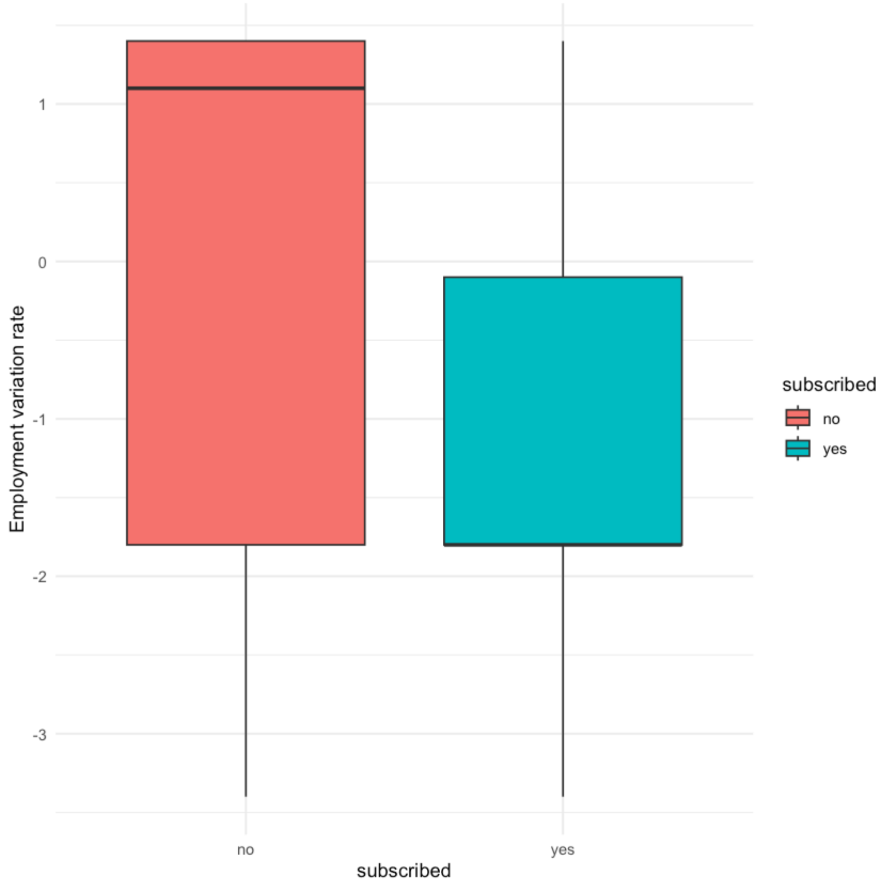
```
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1,size = 15),legend.po
```

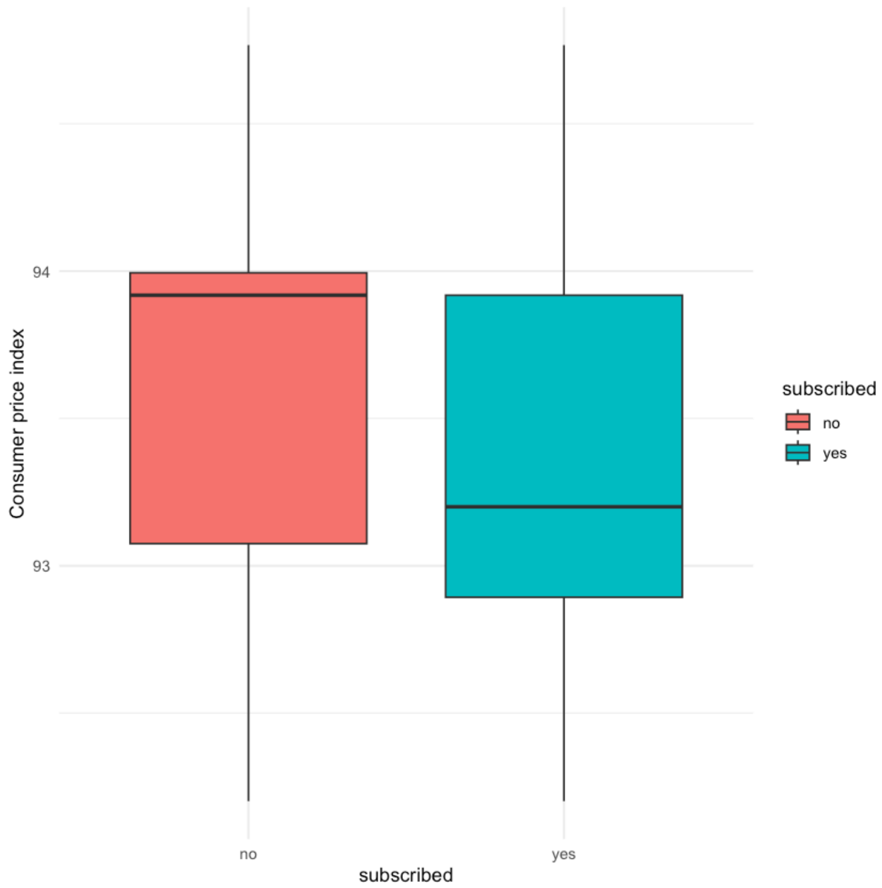


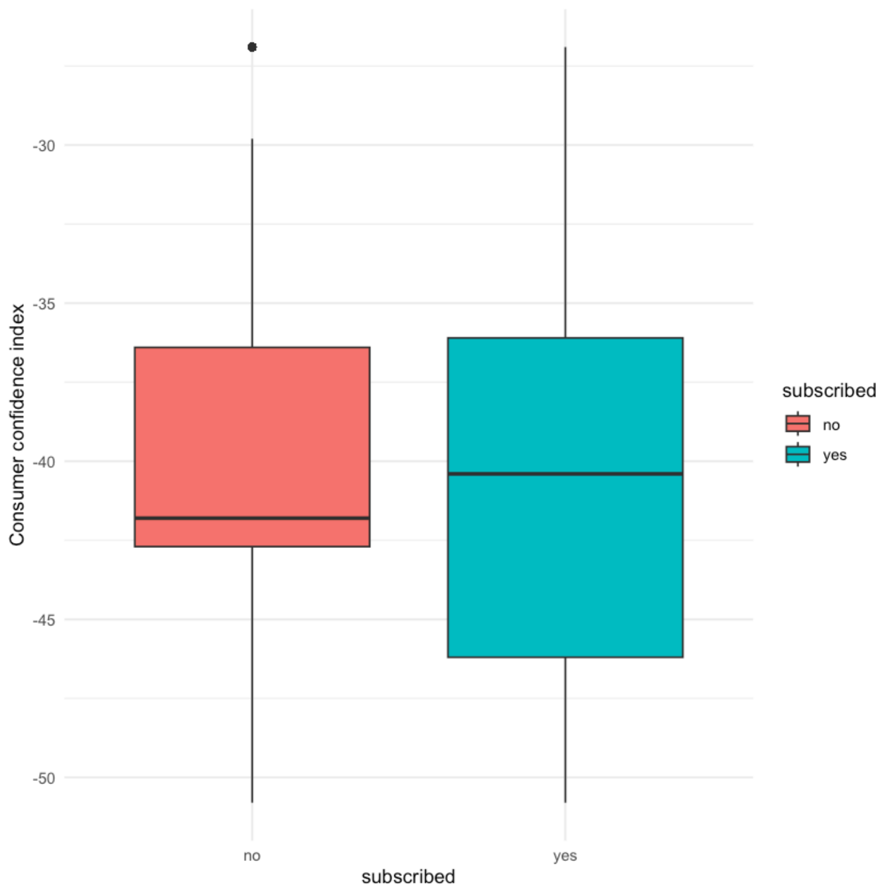


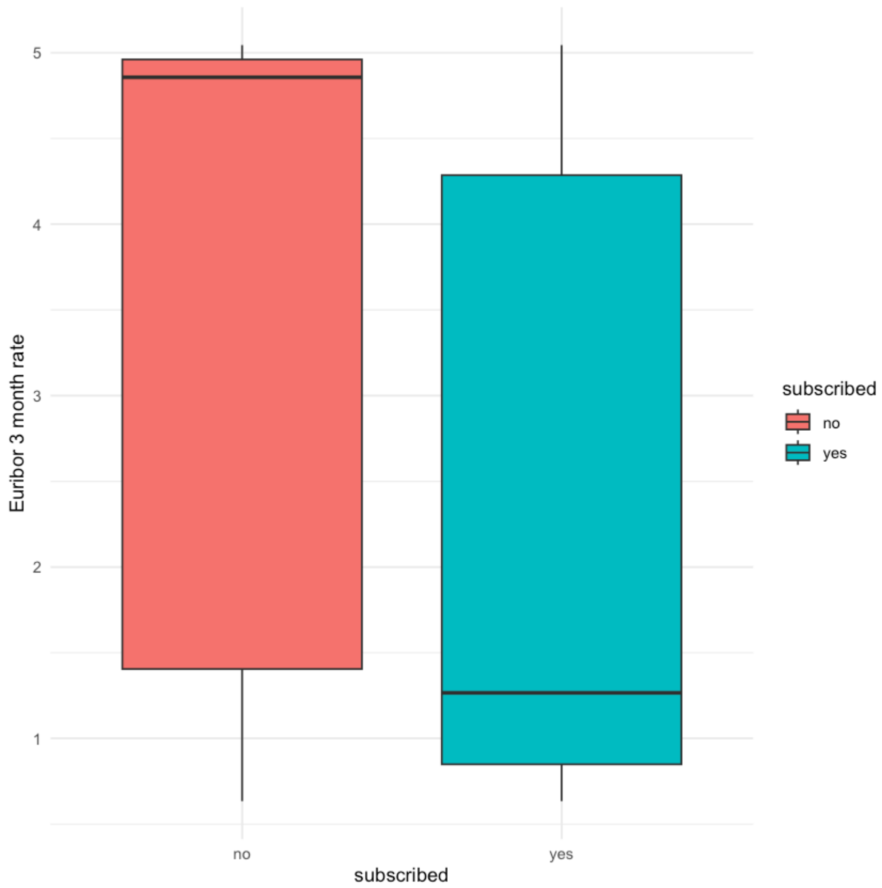
In [ ]:
```
# Function to create box plot
crear_box_plot <- function(data, variable, label_y) {
  ggplot(data, aes_string(y = variable, x = "subscribed", fill = "subscri
    geom_boxplot() +
    labs(y = label_y) +
    theme_minimal()
}

# Economic variables
crear_box_plot(term, "emp_var_rate", "Employment variation rate")
crear_box_plot(term, "cons_price_idx", "Consumer price index")
crear_box_plot(term, "cons_conf_idx", "Consumer confidence index")
crear_box_plot(term, "euribor_3m", "Euribor 3 month rate")
crear_box_plot(term, "n_employed", "Number of employees")
```

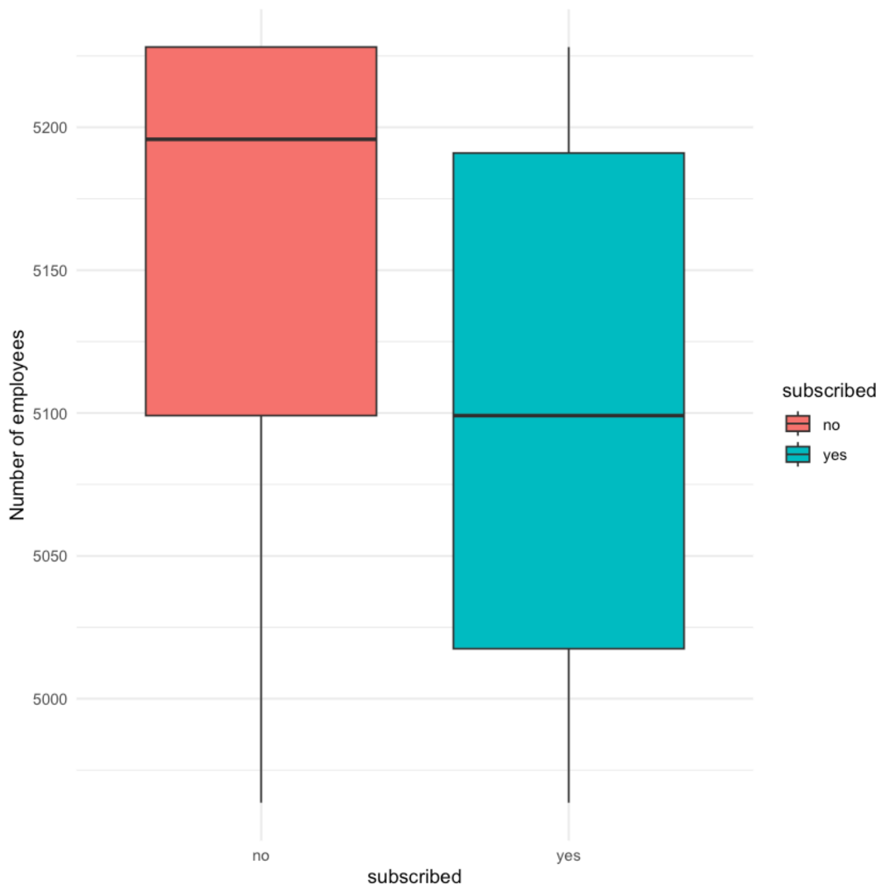
```
Warning message:
"`aes_string()` was deprecated in ggplot2 3.0.0.
i Please use tidy evaluation idioms with `aes()`.
i See also `vignette("ggplot2-in-packages")` for more information."
```




94
93
Consumer price index
no
yes
subscribed
subscribed
no
yes


-30
-35
-40
-45
-50
Consumer confidence index
no
yes
subscribed
subscribed
no
yes





Models Development using the Unbalanced

Dataset

Logistic Regression, Decision Tree and Random Forest

In [ ]:
```
levels(term$subscribed)

#omit missing values
term <- na.omit(term)

#split the data into train (80%) and test (20%) data set.
set.seed(40425150)
index <- createDataPartition(term$subscribed, p=0.8, list=FALSE)
train <- term[index,]
test <- term[-index,]
```

'no' · 'yes'

Logistic Regression

Build Logistic Regression models and select the most accurate

In [ ]:
```
#1 model
formula1 = subscribed  ~ age + occupation + marital_status + euribor_3m +
model1 <- glm(formula = formula1 , data = train, family= "binomial")

#2 model
formula2 = subscribed  ~ age + occupation + marital_status + education_le
model2 <- glm( formula = formula2, data = train, family= "binomial" )

#3 model
formula3 = subscribed  ~ age + marital_status + credit_default + month +
model3 <- glm(formula = formula3, data = train, family= "binomial")

#4 model
formula4 = subscribed ~ day_of_week + occupation + contact_method + campa
model4 <- glm(formula = formula4 , data = train, family = "binomial")
```

Comparison of the models

In [ ]:
```
logisticPseudoR2s <- function(LogModel) {
  dev <- LogModel$deviance
  nullDev <- LogModel$null.deviance
  modelN <- length(LogModel$fitted.values)
  R.l <- 1 - dev / nullDev
  R.cs <- 1- exp ( -(nullDev - dev) / modelN)
  R.n <- R.cs / ( 1 - ( exp (-(nullDev / modelN))))
  cat("Pseudo R^2 for logistic regression\n")
  cat("Hosmer and Lemeshow R^2 ", round(R.l, 3), "\n")
  cat("Cox and Snell R^2 ", round(R.cs, 3), "\n")
```

```
  cat("Nagelkerke R^2 ", round(R.n, 3), "\n")
}

logisticPseudoR2s(model1)
logisticPseudoR2s(model2)
logisticPseudoR2s(model3)
logisticPseudoR2s(model4)
```

```
Pseudo R^2 for logistic regression
Hosmer and Lemeshow R^2  0.148
Cox and Snell R^2  0.099
Nagelkerke R^2  0.196
Pseudo R^2 for logistic regression
Hosmer and Lemeshow R^2  0.149
Cox and Snell R^2  0.1
Nagelkerke R^2  0.197
Pseudo R^2 for logistic regression
Hosmer and Lemeshow R^2  0.191
Cox and Snell R^2  0.126
Nagelkerke R^2  0.249
Pseudo R^2 for logistic regression
Hosmer and Lemeshow R^2  0.201
Cox and Snell R^2  0.133
Nagelkerke R^2  0.262
```

Model 4 is the best performing model among all those constructed. This model holds the lowest AIC and the highest p-pseudo R^2.

In [ ]:
```
summary(model4)
```

```
Call:
glm(formula = formula4, family = "binomial", data = train)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)    
(Intercept)             -2.774e+01  4.042e+00  -6.864 6.69e-12 ***
day_of_weekmon          -1.797e-01  6.362e-02  -2.824 0.004740 ** 
day_of_weekthu           8.393e-02  6.129e-02   1.369 0.170855    
day_of_weektue           8.098e-02  6.346e-02   1.276 0.201947    
day_of_weekwed           1.557e-01  6.272e-02   2.482 0.013059 *  
occupationblue-collar   -3.100e-01  6.180e-02  -5.015 5.30e-07 ***
occupationentrepreneur  -2.449e-01  1.208e-01  -2.027 0.042641 *  
occupationhousemaid     -1.837e-01  1.357e-01  -1.354 0.175868    
occupationmanagement    -9.733e-02  8.132e-02  -1.197 0.231361    
occupationretired        2.426e-01  8.162e-02   2.972 0.002956 ** 
occupationself-employed -1.139e-01  1.130e-01  -1.008 0.313439    
occupationservices      -2.265e-01  7.847e-02  -2.886 0.003901 ** 
occupationstudent        2.376e-01  1.023e-01   2.322 0.020228 *  
occupationtechnician    -1.023e-01  6.107e-02  -1.675 0.093939 .  
occupationunemployed     6.228e-02  1.176e-01   0.530 0.596292    
occupationunknown       -1.490e-01  2.277e-01  -0.654 0.512857    
contact_methodtelephone -2.455e-01  6.045e-02  -4.062 4.87e-05 ***
campaign                -4.194e-02  1.004e-02  -4.175 2.97e-05 ***
monthaug                 4.103e-01  8.415e-02   4.876 1.08e-06 ***
monthdec                 9.265e-01  1.897e-01   4.885 1.04e-06 ***
monthjul                 4.686e-01  8.584e-02   5.458 4.80e-08 ***
monthjun                 3.375e-01  8.670e-02   3.892 9.93e-05 ***
monthmar                 1.209e+00  1.168e-01  10.353  < 2e-16 ***
monthmay                -5.807e-01  7.098e-02  -8.181 2.82e-16 ***
monthnov                 1.551e-01  8.995e-02   1.724 0.084754 .  
monthoct                 6.832e-01  1.081e-01   6.318 2.64e-10 ***
monthsep                 3.969e-01  1.180e-01   3.364 0.000769 ***
euribor_3m              -5.153e-01  1.689e-02 -30.507  < 2e-16 ***
cons_price_idx           3.072e-01  4.338e-02   7.081 1.44e-12 ***
pdays                   -1.456e-03  7.235e-05 -20.126  < 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: 23133  on 32759  degrees of freedom
Residual deviance: 18473  on 32730  degrees of freedom
AIC: 18533

Number of Fisher Scoring iterations: 6
```

Customers contacted on a Wednesday (OR=1.16) are more likely to subscribe than those contacted on a Friday. However, compared to Friday, those who were contacted on a Monday (OR=0.83) are less likely to subscribe. The odds ratio (OR) quantifies the probability of an outcome occurring in relation to a specific exposure and it is commonly used in logistic regression (Szumilas, 2010). For instance, in this context, customers contacted on Wednesday have 1.16 times higher odds of subscribing compared to those contacted on Friday.

Compared to customers working as administrators, those employed in blue-collar roles (OR=0.74), services (OR=0.78), as entrepreneurs (OR=0.81), or as technicians (OR=0.90) are less likely to subscribe. In contract, customers who are retired (OR=1.32) or student (OR=1.33) show a significantly higher likelihood of subscribing compared to administrators.

Customers contacted by telephone (OR=0.78) are less likely to subscribe compared to those contacted by cellular. The fewer contacts made during the campaign (OR=0.95), the more likely a customer is to subscribe to the term deposit. Similarity, the fewer days that have passed since the customer was last contacted (OR=0.99), the higher the likelihood of subscription. These findings corroborate the results of Choi and Choi (2022), who observed similar relationships using a random forest algorithm.

Compared to customers who were contacted in April, those contacted in May (OR=0.55) are less likely to subscribed. In contrast, customers contacted in August (OR=1.5), December (OR=2.52), July (OR=1.59), June (1.40), March (OR=3.35), November (OR=1.16), September (OR=1.48), and October (OR=1.98) are significantly more likely to subscribe than those contacted in February. While this finding contradicts the observations made by Choi and Choi (2022), it is supported by Xie et. al. (2023), who identified the month of contact, day of the week, and occupation as the three most influential variables in determining the likelihood of subscription.

The consumer price index (OR=1.36) is positively correlated with subscription likelihood, indicating that as the index rises, the likelihood of subscribe rises. However, the Euribor 3-month rate (OR=0.59) has a negative relationship with subscription likelihood, meaning that a higher Euribor 3-month rate reduces the probability of a customer subscribing to the term deposit. Similarly, Moro et al. (2014), utilizing a neural network (NN) approach, also found a negative correlation between the Euribor rate and the likelihood of subscription.

Assumptions checking

In [ ]:

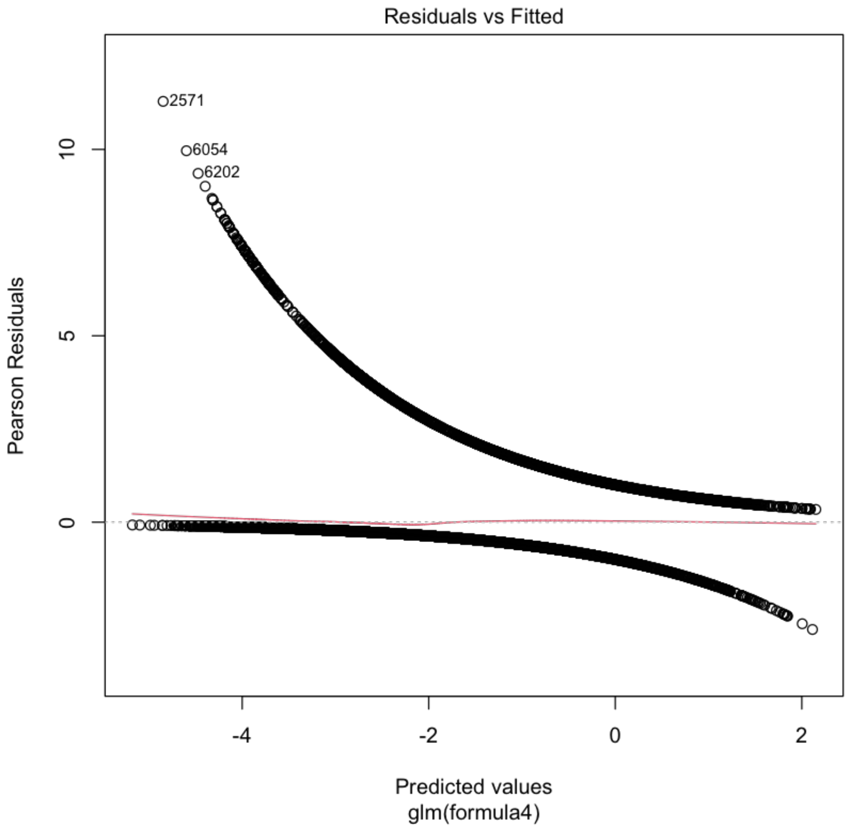
```
#residuals
train$standarisedResiduals <- rstandard(model4)
train$studentdResiduals <- rstudent(model4)
sum(train$standarisedResiduals > 1.96)
sum(train$standarisedResiduals > 2.58)
sum(train$standarisedResiduals > 3)
plot(model4)


#binned residual plot
```

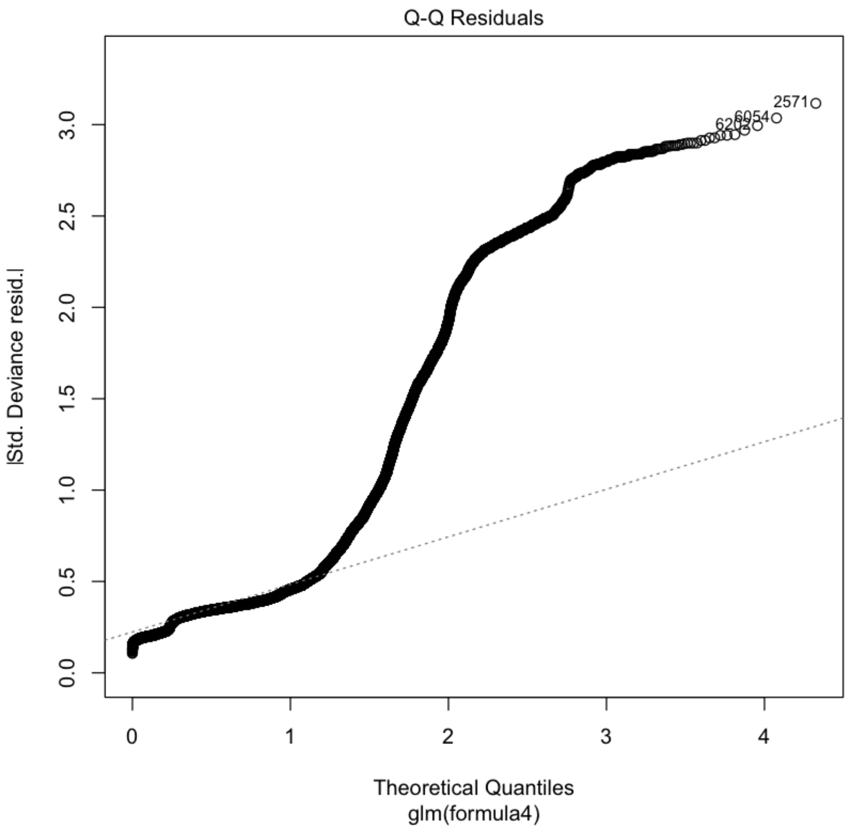
```
binnedplot(fitted(model4),
           residuals(model4, type = "response"),
           col.pts = 1,
           col.int = "red")



#influential cases
train$cook <- cooks.distance(model4)
sum(train$cook > 1)

train$leverage <- hatvalues(model4)
sum(train$leverage > 0.0009)



plot(model4,which=4)

#multicollinearity
vif(model4)

#linearity of the logit (log of numerical variables)
train$log_camp <- log(train$campaign)*train$campaign
train$log_pdays <- log(train$pdays)*train$pdays
train$log_euribor <- log(train$euribor_3m)*train$euribor_3m
train$log_cons <- log(train$cons_price_idx)*train$cons_price_idx



formula_linea = subscribed ~ day_of_week + occupation + contact_method +
model_check <- glm(formula = formula_linea , data = train, family = "bino
```

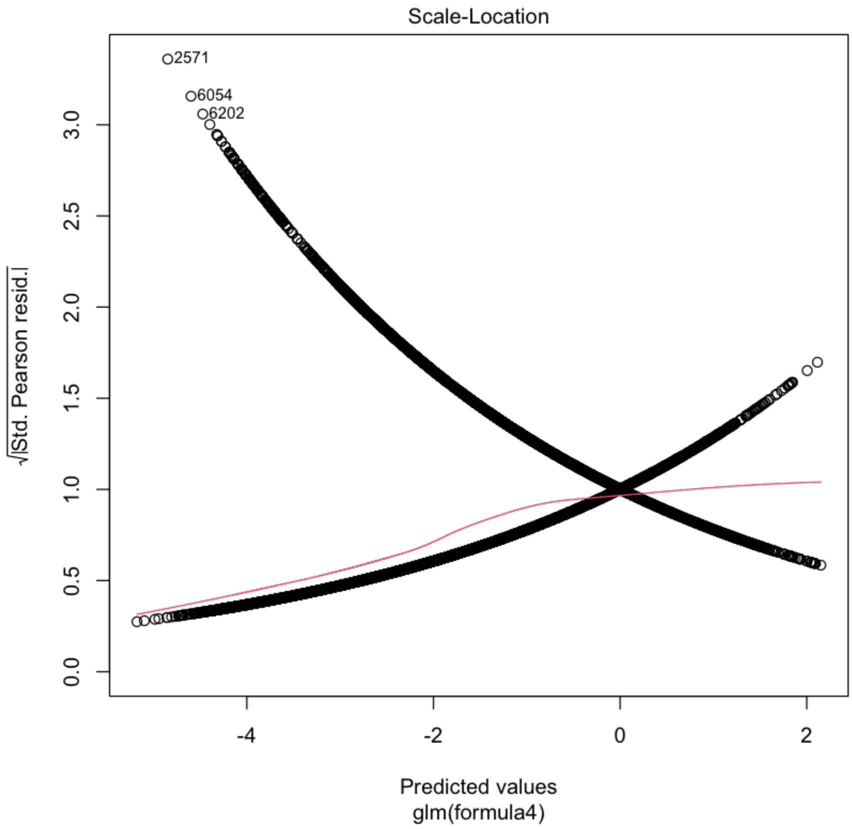
1453

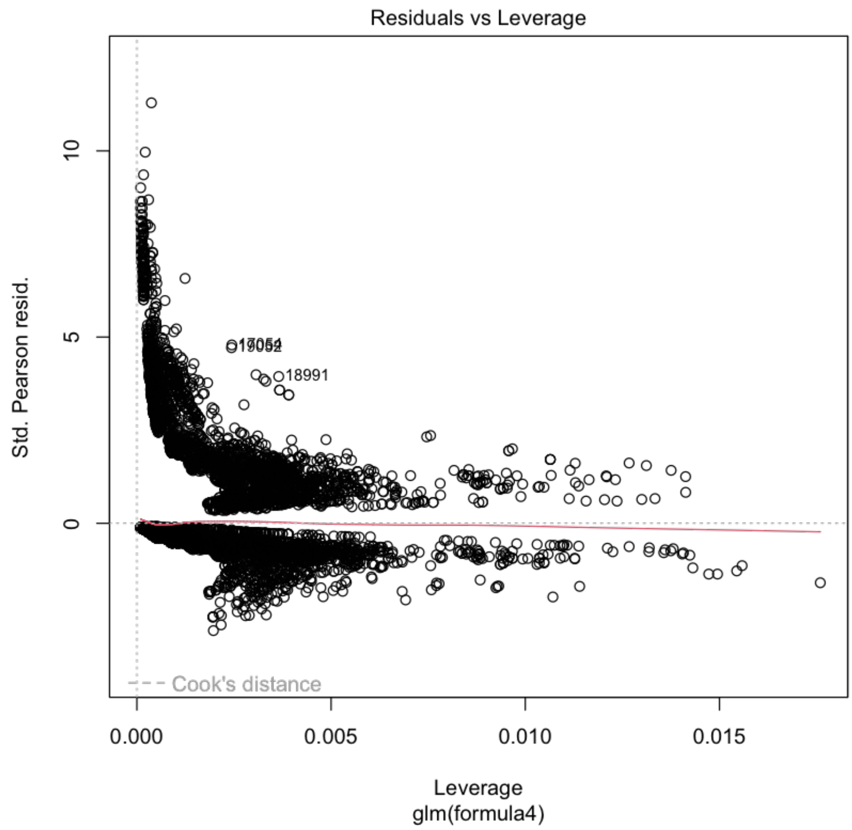
209

2


Residuals vs Fitted
2571
6054
6202
10
5
0
Pearson Residuals
-4
-2
0
2
Predicted values
glm(formula4)

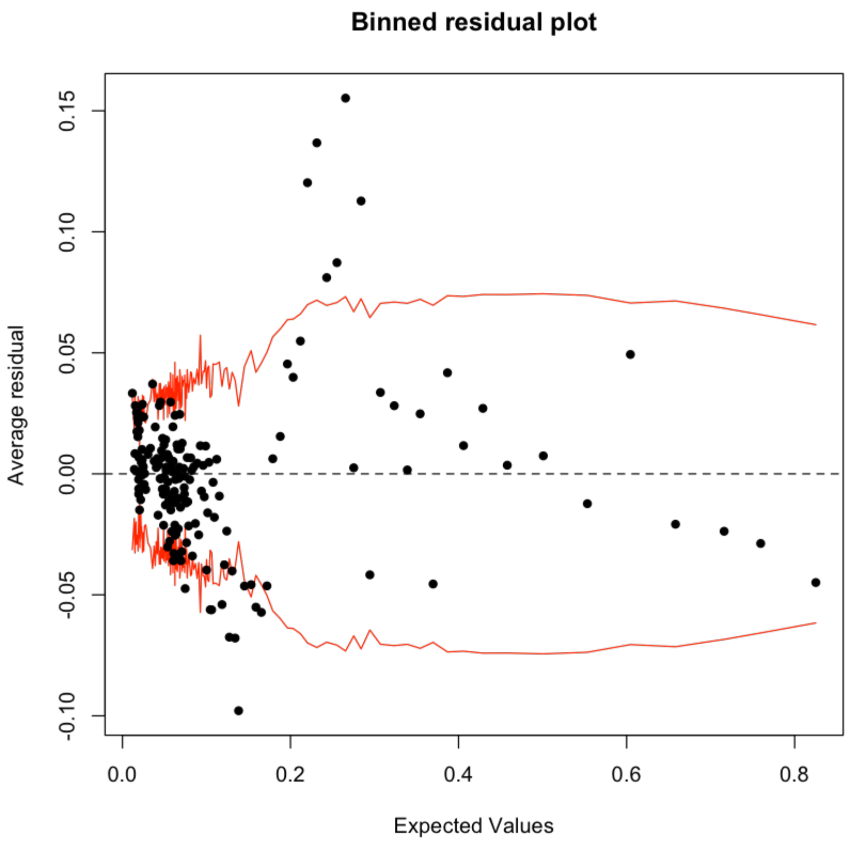

Q-Q Residuals
2571
6054
6202
3.0
2.5
2.0
1.5
1.0
0.5
0.0
|Std. Deviance resid.|
0
1
2
3
4
Theoretical Quantiles
glm(formula4)


Scale-Location
√|Std. Pearson resid.|
Predicted values
glm(formula4)
2571
6054
6202
0.0
0.5
1.0
1.5
2.0
2.5
3.0
-4
-2
0
2


Residuals vs Leverage
Std. Pearson resid.
Leverage
glm(formula4)
18991
Cook's distance
0.000
0.005
0.010
0.015
0
5
10

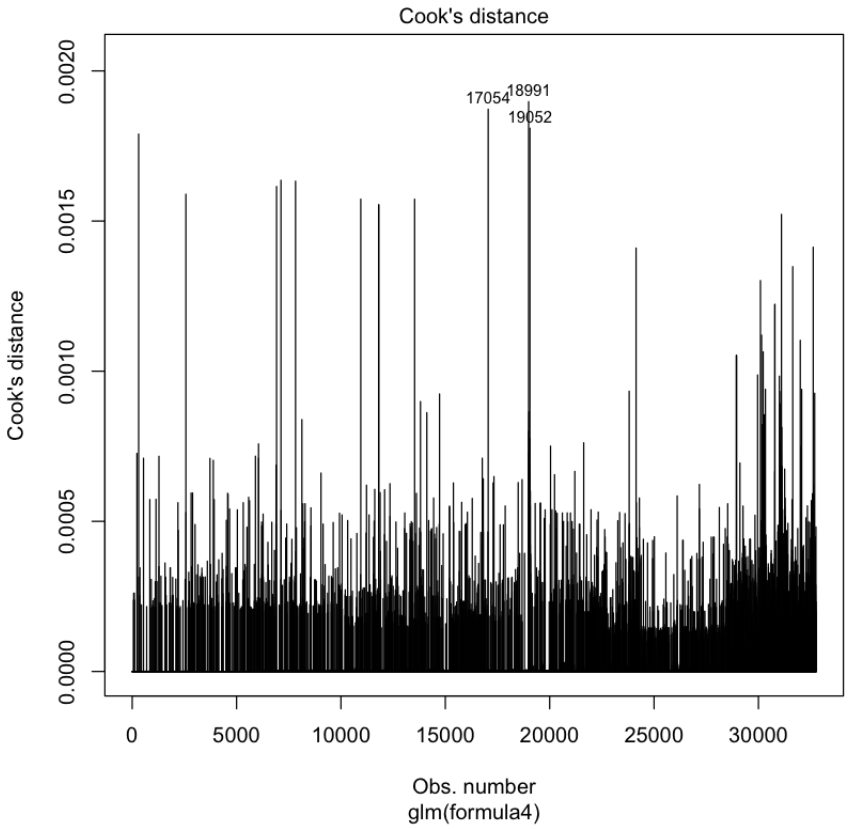
0

8535



A matrix: 8 x 3 of type dbl

	GVIF	Df	GVIF^(1/(2*Df))
day_of_week	1.039494	4	1.004853
occupation	1.181566	11	1.007612
contact_method	1.527067	1	1.235745
campaign	1.041143	1	1.020364
month	2.804263	9	1.058958
euribor_3m	2.480884	1	1.575082
cons_price_idx	2.056299	1	1.433980
pdays	1.155520	1	1.074951



The shaded red areas represent the range where approximately 95% of the observations are expected to be found. While not all values fall within this red area, less than 5% of the observations lie outside these boundaries.

There are no variables in the dataset with Generalized Variance Inflation Factor (GVIF) values exceeding the threshold of 10. This threshold is commonly used as a benchmark to evaluate the presence of multicollinearity.

When examining the influential cases within the model, no value has a Cook's Distance greater than 1. However, there are 8535 instances with a leverage higher than 0.0009.

Lastly, it appears that the linearity of the logit assumption is violated, as the log of numerical variables are statistically significant for the model. This violation leads to reduced confidence in the model's generalizability to the population from which the sample was drawn.

Model Performance with test data set

In [ ]:
```
#Predictions with test data

predictions <- predict(model4,test, type = "response")
class_pred <- as.factor(ifelse(predictions > 0.5, "yes", "no"))
postResample(class_pred, test$subscribed)
```

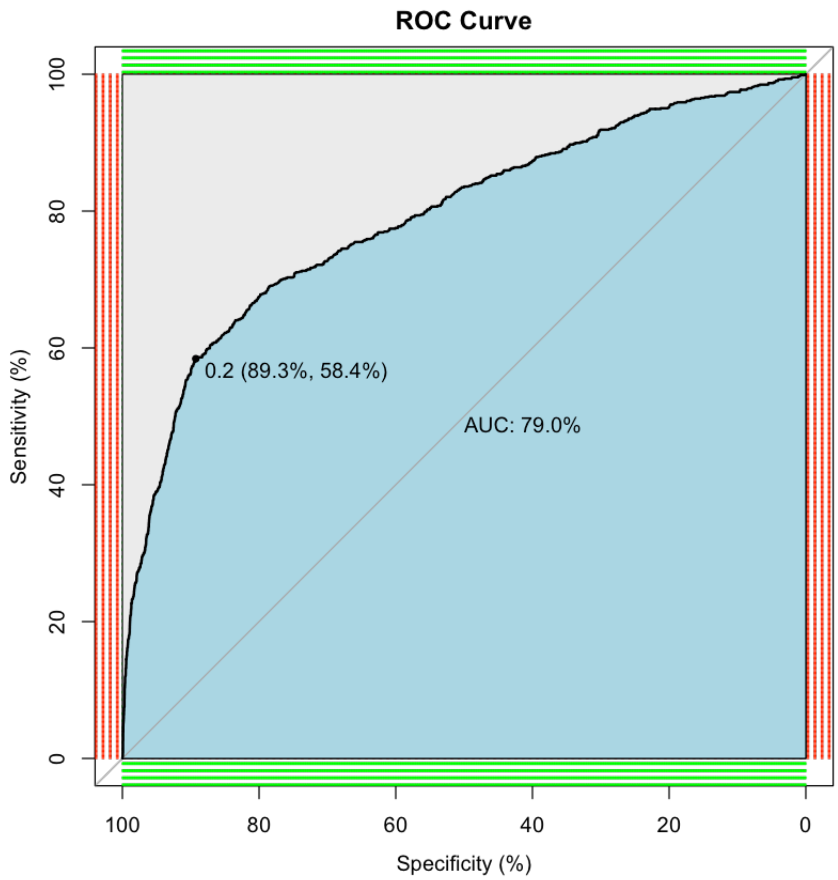
```
conf_matrix <- confusionMatrix(class_pred, test$subscribed, positive = "y
conf_matrix
```

Accuracy: 0.900598363658566 **Kappa:** 0.303670428980682

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  7162  713
       yes  101  213

               Accuracy : 0.9006
                 95% CI : (0.8939, 0.907)
    No Information Rate : 0.8869
    P-Value [Acc > NIR] : 3.779e-05

                  Kappa : 0.3037

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.23002
            Specificity : 0.98609
         Pos Pred Value : 0.67834
         Neg Pred Value : 0.90946
             Prevalence : 0.11308
         Detection Rate : 0.02601
   Detection Prevalence : 0.03834
      Balanced Accuracy : 0.60806

       'Positive' Class : yes
```

In [ ]:
```
#ROC Curve
r <- multiclass.roc(test$subscribed, predictions, percent = TRUE)
roc <- r[['rocs']]
r1 <- roc[[1]]
plot.roc(r1,
         print.auc=TRUE,
         auc.polygon=TRUE,
         grid=c(0.1, 0.2),
         grid.col=c("green", "red"),
         max.auc.polygon=TRUE,
         auc.polygon.col="lightblue",
         print.thres=TRUE,
         main= 'ROC Curve')
```

```
Setting direction: controls < cases
```



Decision Tree Classifier

In [ ]:
```
#build decision tree with all variables except ID and contact duration
tree <- rpart(subscribed ~ age + occupation + marital_status + education_
```

Model Performace with test dataset

In [ ]:
```
#Predictions with test data
predictions_tree <- predict(tree, test, type = "class")
postResample(predictions_tree, test$subscribed)
cm_tree <- confusionMatrix(predictions_tree, test$subscribed)
cm_tree
```

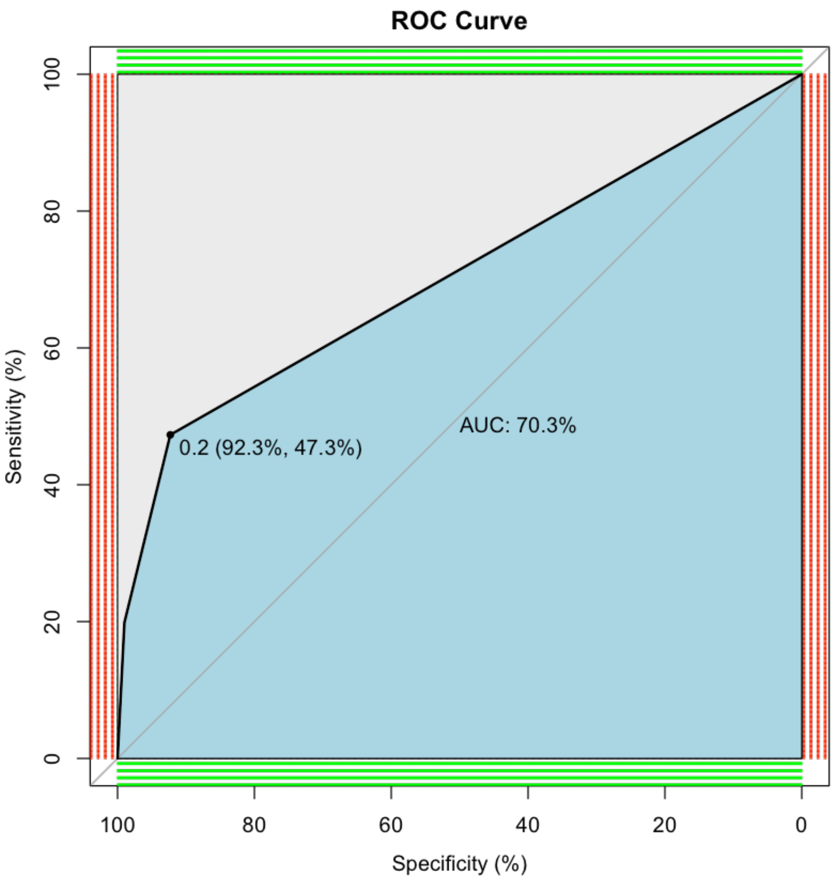
Accuracy: 0.900476248626206 **Kappa:** 0.275476753662194

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  7190  742
       yes   73  184

               Accuracy : 0.9005
                 95% CI : (0.8938, 0.9069)
    No Information Rate : 0.8869
    P-Value [Acc > NIR] : 4.394e-05

                  Kappa : 0.2755

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9899
            Specificity : 0.1987
         Pos Pred Value : 0.9065
         Neg Pred Value : 0.7160
             Prevalence : 0.8869
         Detection Rate : 0.8780
   Detection Prevalence : 0.9686
      Balanced Accuracy : 0.5943

       'Positive' Class : no
```

In [ ]:

```
#ROC Curve
predictions_tree_1 <- predict(tree, test, type = "prob")
probabilities_tree <- predictions_tree_1[, "yes"]

r1 <- multiclass.roc(test$subscribed, probabilities_tree, percent = TRUE)
roc1 <- r1[['rocs']]
r2 <- roc1[[1]]
plot.roc(r2,
         print.auc=TRUE,
         auc.polygon=TRUE,
         grid=c(0.1, 0.2),
         grid.col=c("green", "red"),
         max.auc.polygon=TRUE,
         auc.polygon.col="lightblue",
         print.thres=TRUE,
         main= 'ROC Curve')
```

```
Setting direction: controls < cases
```



Random Forest

In [ ]:
```
rf <- randomForest(subscribed ~ age + occupation + marital_status + educa
```

Model Performance with test data

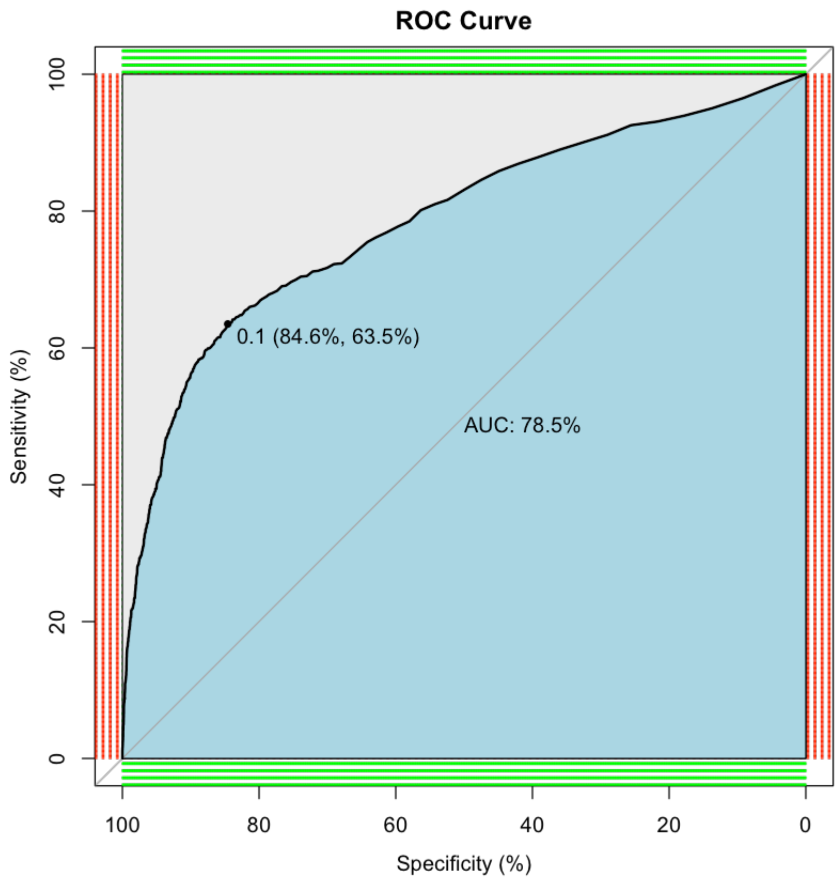
In [ ]:
```
#Confusion Matrix
pred_rf <- predict(rf,test)
cm_rf <- confusionMatrix(pred_rf,test$subscribed,positive = "yes")
cm_rf
```

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  7066  652
       yes  197  274

               Accuracy : 0.8963
                 95% CI : (0.8895, 0.9028)
    No Information Rate : 0.8869
    P-Value [Acc > NIR] : 0.003487

                  Kappa : 0.3421

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.29590
            Specificity : 0.97288
         Pos Pred Value : 0.58174
         Neg Pred Value : 0.91552
             Prevalence : 0.11308
         Detection Rate : 0.03346
   Detection Prevalence : 0.05752
      Balanced Accuracy : 0.63439

       'Positive' Class : yes
```

In [ ]:

```r
#ROC Curve
predictions_rf_1 <- predict(rf, test, type = "prob")
probabilities_rf <- predictions_rf_1[, "yes"]

r2 <- multiclass.roc(test$subscribed, probabilities_rf, percent = TRUE)
roc2 <- r2[['rocs']]
r3 <- roc2[[1]]
plot.roc(r3,
         print.auc=TRUE,
         auc.polygon=TRUE,
         grid=c(0.1, 0.2),
         grid.col=c("green", "red"),
         max.auc.polygon=TRUE,
         auc.polygon.col="lightblue",
         print.thres=TRUE,
         main= 'ROC Curve')
```

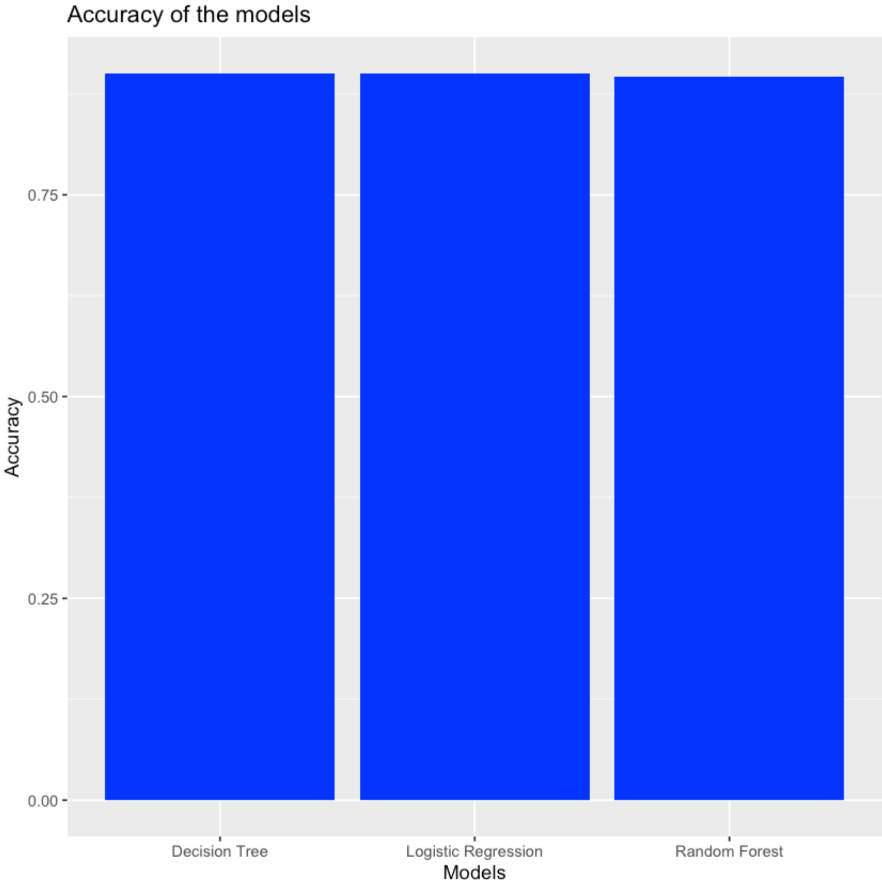
```
Setting direction: controls < cases
```

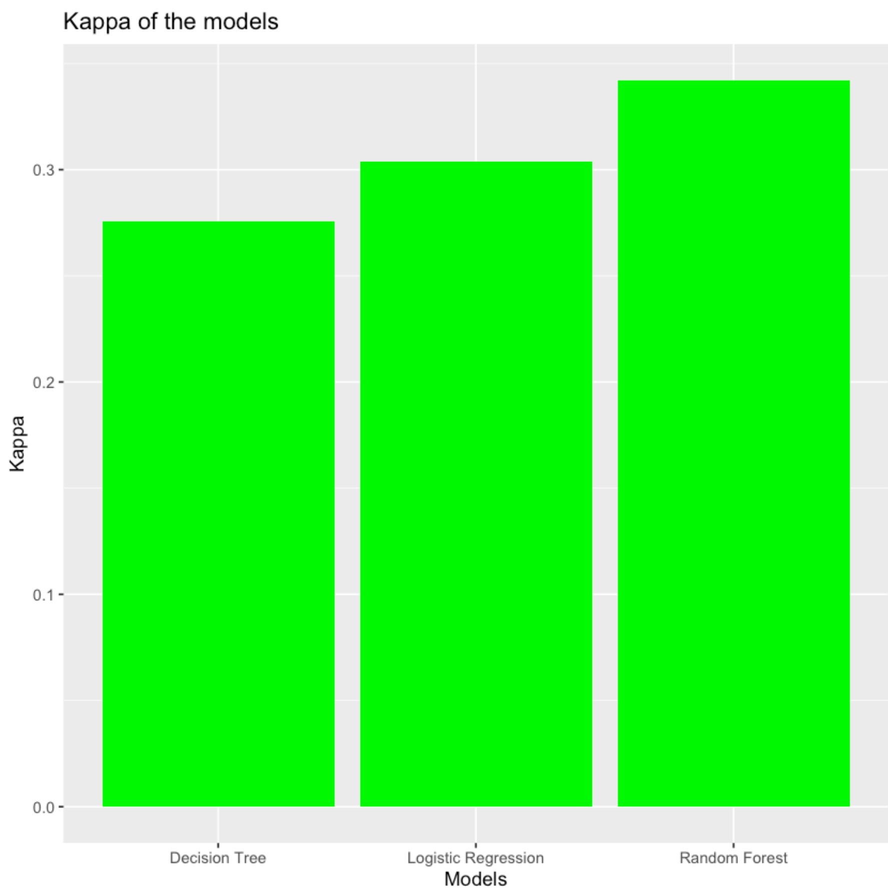


Comparative of Models Performance

In [ ]:
```
##Comparative of models accuracy
models <- data.frame(Model = c('Logistic Regression',
                                    'Decision Tree',
                                    'Random Forest'),
                            Accuracy = c(conf_matrix$overall[1],
                                         cm_tree$overall[1],
                                         cm_rf$overall[1]))

models1 <- data.frame(Model = c('Logistic Regression',
                                    'Decision Tree',
                                    'Random Forest'),
                            kappa = c(conf_matrix$overall[2],
                                         cm_tree$overall[2],
                                         cm_rf$overall[2]))


#Plot comparing accuracy
ggplot(aes(x=Model, y=Accuracy), data=models) +
  geom_bar(stat='identity', fill = 'blue') +
  ggtitle('Accuracy of the models') +
  xlab('Models') +
  ylab('Accuracy')

#Plot comparing kappa
ggplot(aes(x=Model, y=kappa), data=models1) +
```

```
  geom_bar(stat='identity', fill = 'green') +
  ggtitle('Kappa of the models') +
  xlab('Models') +
  ylab('Kappa')
```





When working with unbalanced data sets there is a risk that a model will be biased towards predicting the majority class. The models have an accuracy of around 90%, which may seem impressive at first glance. However, 88% of the data represents "no" subscriptions, so a random guess would be almost as effective.

This is where metrics like Kappa become crucial. They provide deeper insight into the performance of a model, beyond what standard accuracy measures can offer. In the context of our data set, the Kappa statistic is especially revealing. It shows that the **Random Forest** model is the most effective at correctly identifying True Positives.

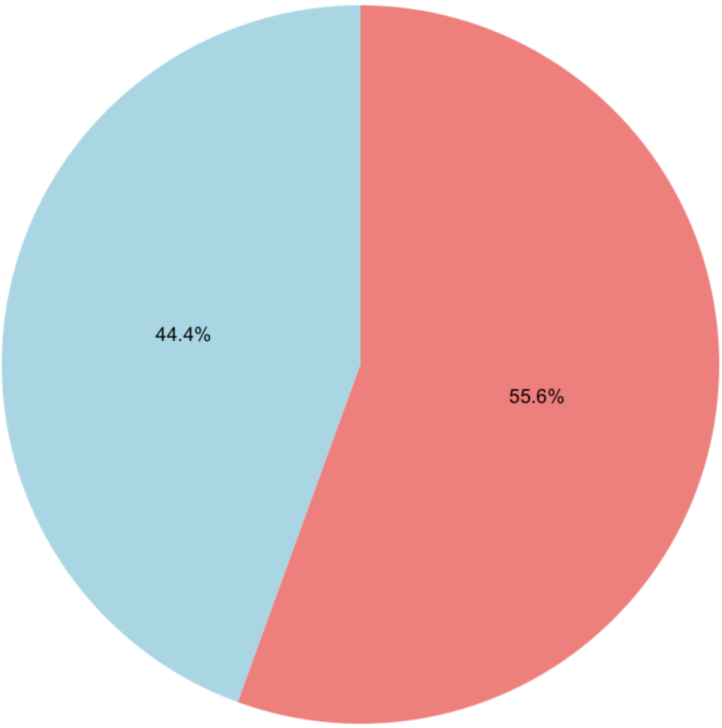
Models Development using SMOTE

In [ ]:
```
library ("DMwR")

smote_dataset <- as.data.frame(term)

#balancing the data
smote <-
  SMOTE(
    form = subscribed ~ .,
    data = smote_dataset,
    perc.over = 400,
    perc.under = 100
  )
```

```
Loading required package: grid

Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

In [ ]:

```
#graph comparing subscribed yes vs no
#calculate the percentage of "yes" "no". 1. A table is created to see the
per_subscribed_smote <- data.frame(prop.table(table(smote$subscribed)) *
ggplot(per_subscribed_smote, aes(x = "", y = Freq, fill = Var1)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) +
  theme_void() +
  scale_fill_manual(values = c("lightblue", "lightcoral")) +
  geom_text(aes(label = sprintf("%.1f%%", Freq)), position = position_sta
  theme(legend.position = "none")
```



Logistic Regression, Decision Tree and Random Forest

In [ ]:

```
set.seed(40425150)
index <- createDataPartition(smote$subscribed, p=0.8, list=FALSE)
train_smote <- smote[index,]
test_smote <- smote[-index,]
```

Logistic Regression

In [ ]:
```
formula_smote = subscribed ~ day_of_week + occupation + contact_method +
model_smote <- glm(formula = formula_smote , data = train_smote, family =
summary(model_smote)
```

```
Call:
glm(formula = formula_smote, family = "binomial", data = train_smote)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)    
(Intercept)             -3.277e+01  3.669e+00  -8.931  < 2e-16 ***
day_of_weekmon          -1.301e-01  4.058e-02  -3.205  0.00135 ** 
day_of_weekthu          -6.977e-02  3.992e-02  -1.747  0.08056 .  
day_of_weektue          -5.363e-02  4.122e-02  -1.301  0.19326    
day_of_weekwed          -2.217e-02  4.069e-02  -0.545  0.58592    
occupationblue-collar   -3.935e-02  3.791e-02  -1.038  0.29936    
occupationentrepreneur  -3.250e-01  8.038e-02  -4.043 5.28e-05 ***
occupationhousemaid     -8.761e-02  8.348e-02  -1.049  0.29399    
occupationmanagement    -7.030e-03  5.539e-02  -0.127  0.89899    
occupationretired        3.699e-01  6.266e-02   5.902 3.58e-09 ***
occupationself-employed  1.480e-02  7.146e-02   0.207  0.83594    
occupationservices       1.135e-01  4.765e-02   2.382  0.01723 *  
occupationstudent        8.270e-01  7.872e-02  10.506  < 2e-16 ***
occupationtechnician     2.846e-02  4.122e-02   0.690  0.49002    
occupationunemployed     2.982e-01  7.620e-02   3.913 9.11e-05 ***
occupationunknown        7.316e-01  1.277e-01   5.730 1.00e-08 ***
contact_methodtelephone  2.123e-01  3.275e-02   6.482 9.06e-11 ***
campaign                -5.517e-02  6.054e-03  -9.112  < 2e-16 ***
monthaug                 1.420e-01  6.277e-02   2.263  0.02365 *  
monthdec                 7.753e-01  1.539e-01   5.037 4.72e-07 ***
monthjul                 4.639e-01  5.969e-02   7.773 7.69e-15 ***
monthjun                 1.445e-01  5.925e-02   2.439  0.01474 *  
monthmar                 1.159e+00  1.089e-01  10.646  < 2e-16 ***
monthmay                -5.731e-01  4.992e-02 -11.481  < 2e-16 ***
monthnov                 5.450e-02  6.397e-02   0.852  0.39424    
monthoct                 5.664e-01  9.657e-02   5.866 4.48e-09 ***
monthsep                 3.145e-01  1.058e-01   2.972  0.00295 ** 
euribor_3m              -5.444e-01  1.306e-02 -41.685  < 2e-16 ***
cons_price_idx           3.850e-01  3.937e-02   9.778  < 2e-16 ***
pdays                   -1.429e-03  8.055e-05 -17.737  < 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: 45831  on 33357  degrees of freedom
Residual deviance: 36514  on 33328  degrees of freedom
AIC: 36574

Number of Fisher Scoring iterations: 5
```

Model Performance with test data

In [ ]:
```
#Predictions with test data
```

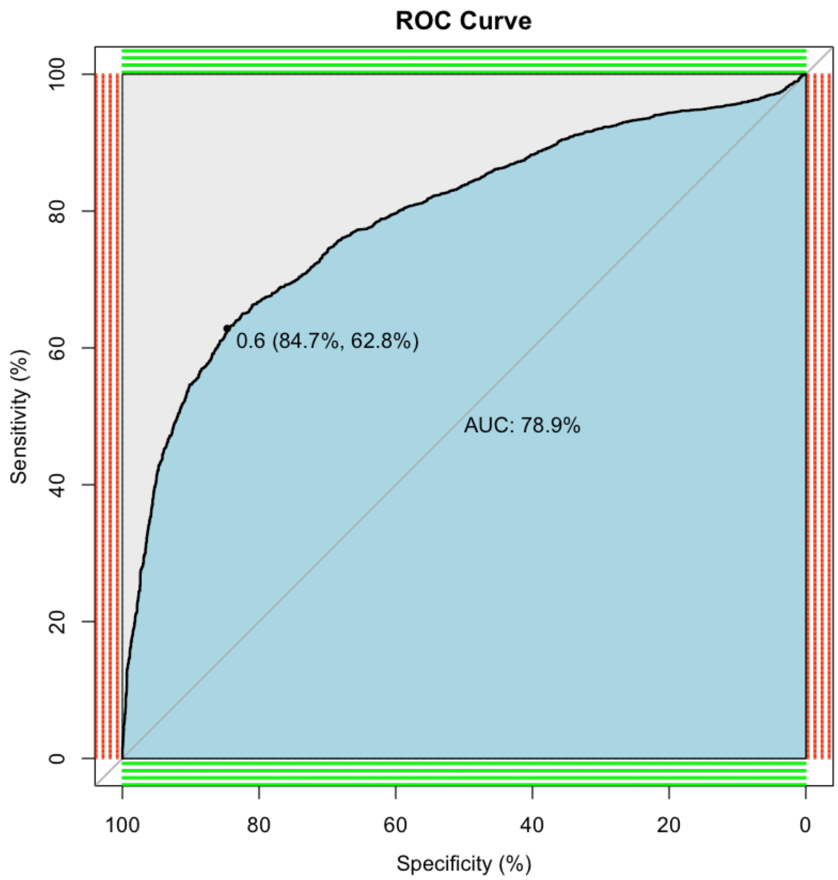
```
predictions_smote <- predict(model_smote,test_smote, type = "response")
class_pred_smote <- as.factor(ifelse(predictions_smote > 0.5, "yes", "no"
postResample(class_pred_smote, test_smote$subscribed)
conf_matrix_smote <- confusionMatrix(class_pred_smote, test_smote$subscri
conf_matrix_smote
```

Accuracy: 0.719870488068114 **Kappa:** 0.437257904875433

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  2687 1317
       yes 1019 3316

               Accuracy : 0.7199
                 95% CI : (0.7101, 0.7295)
    No Information Rate : 0.5556
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4373

 Mcnemar's Test P-Value : 7.998e-10

            Sensitivity : 0.7157
            Specificity : 0.7250
         Pos Pred Value : 0.7649
         Neg Pred Value : 0.6711
             Prevalence : 0.5556
         Detection Rate : 0.3976
   Detection Prevalence : 0.5198
      Balanced Accuracy : 0.7204

       'Positive' Class : yes
```

In [ ]:
```
#ROC Curve
r_smote <- multiclass.roc(test_smote$subscribed, predictions_smote, perce
roc_smote <- r_smote[['rocs']]
r1_smote <- roc_smote[[1]]
plot.roc(r1_smote,
         print.auc=TRUE,
         auc.polygon=TRUE,
         grid=c(0.1, 0.2),
         grid.col=c("green", "red"),
         max.auc.polygon=TRUE,
         auc.polygon.col="lightblue",
         print.thres=TRUE,
         main= 'ROC Curve')
```

```
Setting direction: controls < cases
```



Decision Tree

In [ ]:
```
tree_smote <- rpart(subscribed ~ age + occupation + marital_status + educ
```

Model Performance with test data

In [ ]:
```
#Confusion Matrix
predictions_tree_smote <- predict(tree_smote, test_smote, type = "class")
postResample(predictions_tree_smote, test_smote$subscribed)
cm_tree_smote <- confusionMatrix(predictions_tree_smote, test_smote$subsc
cm_tree_smote
```

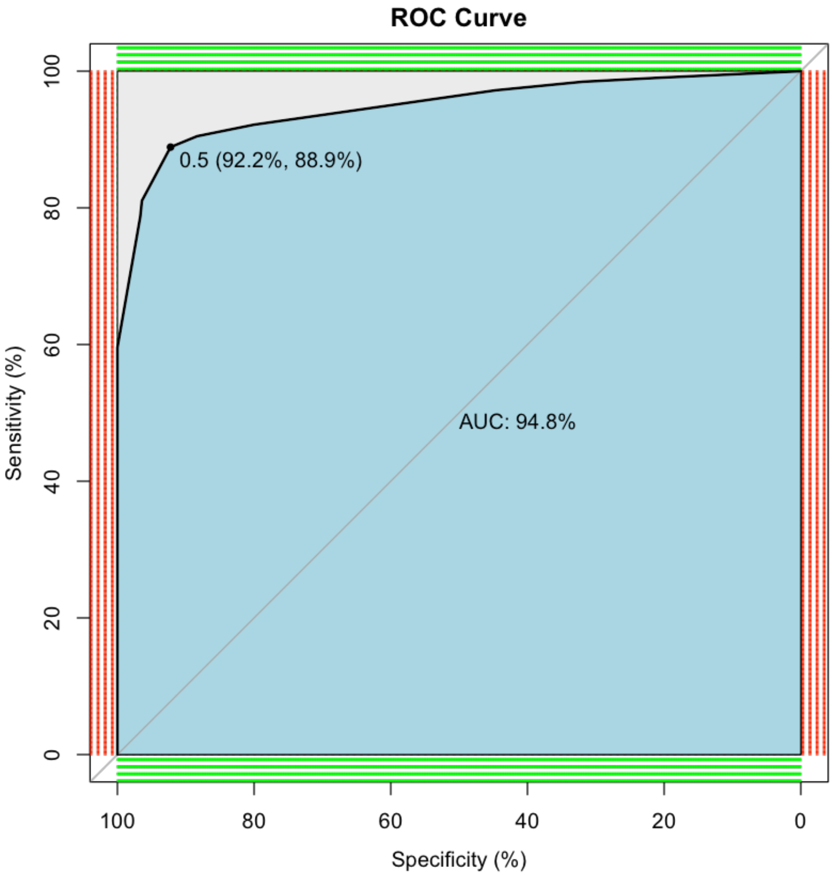
Accuracy: 0.903705480273414 **Kappa:** 0.806188903256057

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  3418  515
       yes  288 4118

               Accuracy : 0.9037
                 95% CI : (0.8972, 0.91)
    No Information Rate : 0.5556
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8062

 Mcnemar's Test P-Value : 1.519e-15

            Sensitivity : 0.9223
            Specificity : 0.8888
         Pos Pred Value : 0.8691
         Neg Pred Value : 0.9346
             Prevalence : 0.4444
         Detection Rate : 0.4099
   Detection Prevalence : 0.4716
      Balanced Accuracy : 0.9056

       'Positive' Class : no
```

In [ ]:

```r
#ROC Curve
predictions_tree_2 <- predict(tree_smote, test_smote, type = "prob")
probabilities_tree2 <- predictions_tree_2[, "yes"]

r5<- multiclass.roc(test_smote$subscribed, probabilities_tree2, percent =
roc_smote1 <- r5[['rocs']]
r1_smote1 <- roc_smote1[[1]]
plot.roc(r1_smote1,
         print.auc=TRUE,
         auc.polygon=TRUE,
         grid=c(0.1, 0.2),
         grid.col=c("green", "red"),
         max.auc.polygon=TRUE,
         auc.polygon.col="lightblue",
         print.thres=TRUE,
         main= 'ROC Curve')
```

```
Setting direction: controls < cases
```



Random Forest

In [ ]:
```
rf_smote <- randomForest(subscribed ~ age + occupation + marital_status +
```

Model Performance with test data set

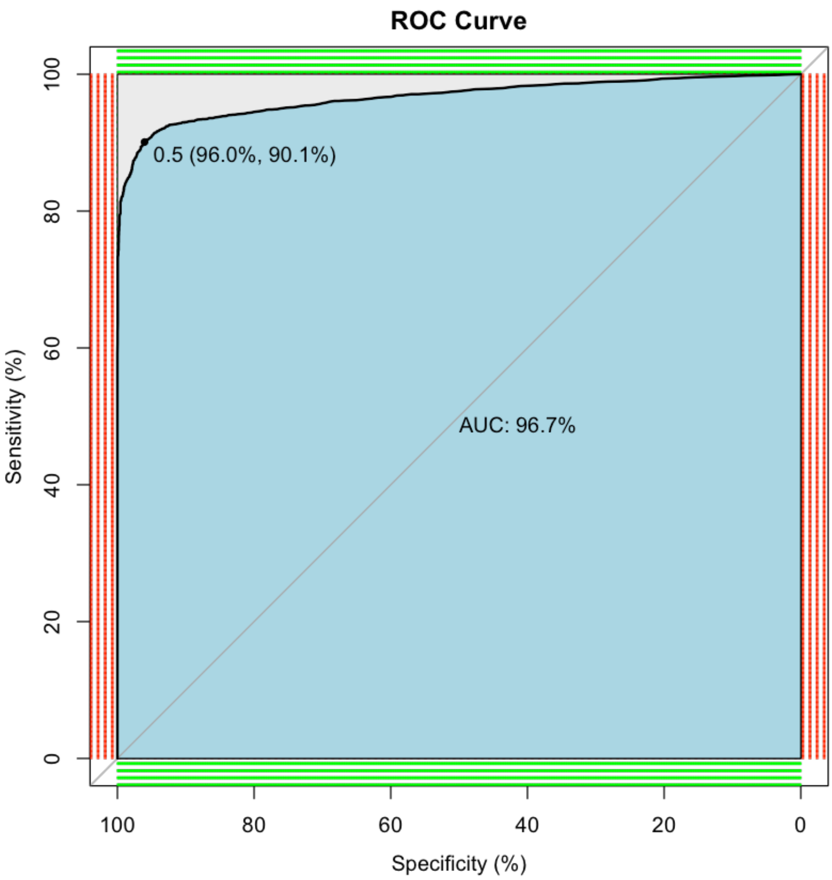
In [ ]:
```
#ROC Curve
pred_rf_smote <- predict(rf_smote,test_smote)
cm_rf_smote <- confusionMatrix(pred_rf_smote,test_smote$subscribed,positi
cm_rf_smote
```

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  3543  442
       yes  163 4191

               Accuracy : 0.9274
                 95% CI : (0.9217, 0.9329)
    No Information Rate : 0.5556
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8542

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9046
            Specificity : 0.9560
         Pos Pred Value : 0.9626
         Neg Pred Value : 0.8891
             Prevalence : 0.5556
         Detection Rate : 0.5026
   Detection Prevalence : 0.5221
      Balanced Accuracy : 0.9303

       'Positive' Class : yes
```

In [ ]:

```
#ROC Curve
predictions_rf_7 <- predict(rf_smote, test_smote, type = "prob")
probabilities_rf7 <- predictions_rf_7[, "yes"]

r7 <- multiclass.roc(test_smote$subscribed, probabilities_rf7, percent =
roc7 <- r7[['rocs']]
r7 <- roc7[[1]]
plot.roc(r7,
         print.auc=TRUE,
         auc.polygon=TRUE,
         grid=c(0.1, 0.2),
         grid.col=c("green", "red"),
         max.auc.polygon=TRUE,
         auc.polygon.col="lightblue",
         print.thres=TRUE,
         main= 'ROC Curve')
```

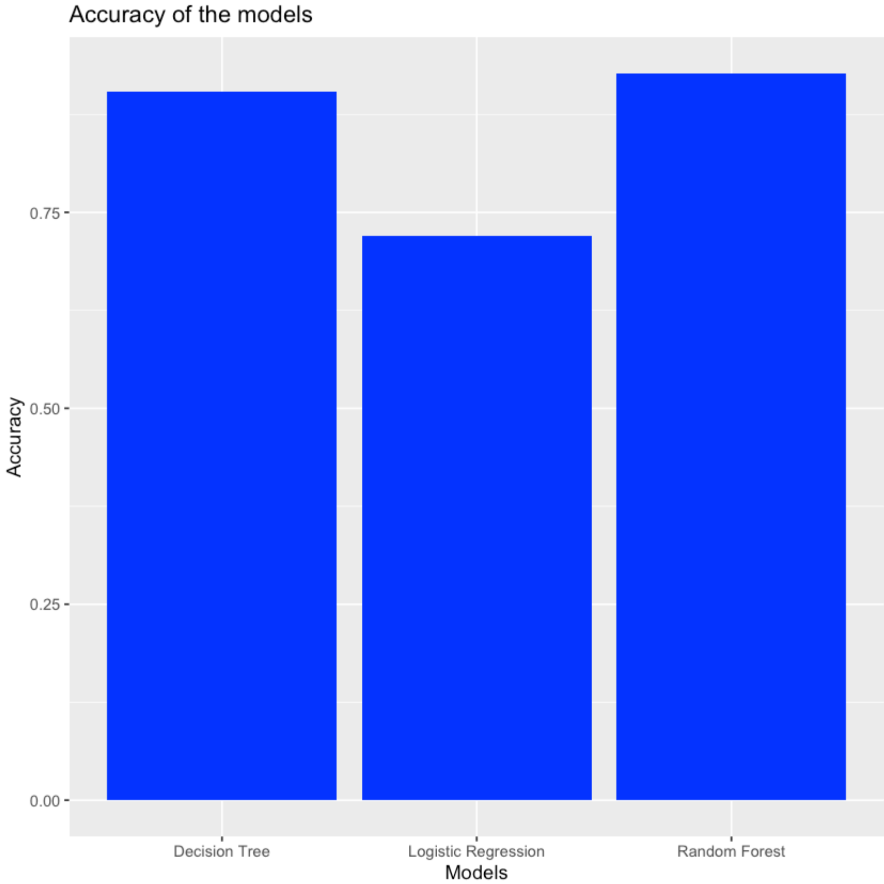
```
Setting direction: controls < cases
```

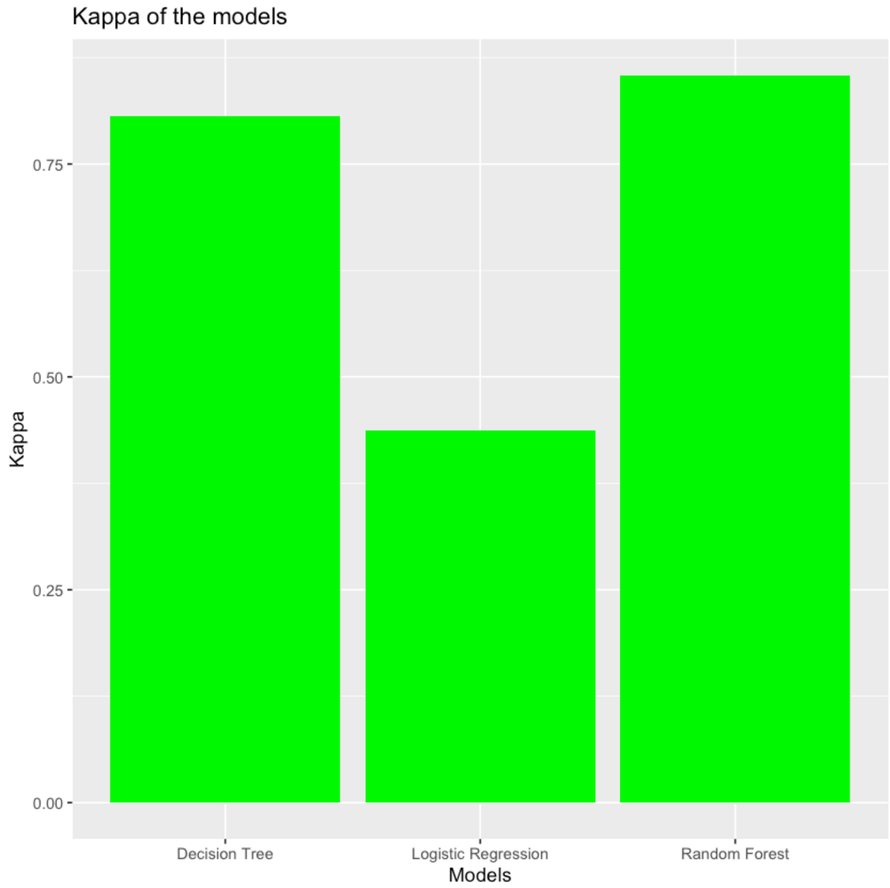


Comparative of Models Performance

In [ ]:
```
##Comparative of models accuracy
models <- data.frame(Model = c('Logistic Regression',
                              'Decision Tree',
                              'Random Forest'),
                    Accuracy = c(conf_matrix_smote$overall[1],
                                 cm_tree_smote$overall[1],
                                 cm_rf_smote$overall[1]))

models1 <- data.frame(Model = c('Logistic Regression',
                               'Decision Tree',
                               'Random Forest'),
                     kappa = c(conf_matrix_smote$overall[2],
                               cm_tree_smote$overall[2],
                               cm_rf_smote$overall[2]))


#Plot comparing accuracy
ggplot(aes(x=Model, y=Accuracy), data=models) +
  geom_bar(stat='identity', fill = 'blue') +
  ggtitle('Accuracy of the models') +
  xlab('Models') +
  ylab('Accuracy')

#Plot comparing kappa
ggplot(aes(x=Model, y=kappa), data=models1) +
```

```r
  geom_bar(stat='identity', fill = 'green') +
  ggtitle('Kappa of the models') +
  xlab('Models') +
  ylab('Kappa')
```





After applying the SMOTE technique to balance our dataset, the performance of our models has improved notably. The best model is again the **Random Forest** model, which presents an accuracy of 0.92. This metric is particularly valuable, given that it reflects the performance in a balanced dataset scenario, unlike our previous model evaluations where the data was heavily biased towards the 'no' subscriptions.

Moreover, in terms of the Kappa statistic, the Random Forest model again stands out with the highest score of 0.84.