CHECK THE DATASETS, WE HAVE TO USE THE UNBALANCED ONE

A. Gottardi, E. Corrolezzis, A. Minutolo, L.F. Palacios Flores

"Doubt the data until the data leave no room for doubt." - Henri Poincaré

Problem statement

The dataset contains the data of the clients of an Insurance company that has provided Health Insurance. Our goal is to analyze the relationship between the features and the probability of the customers buying a vehichle insurance. Now, in order to predict whether the customer would be interested in Vehicle insurance, we have information about demographics (gender, age, region code type), Vehicles (Vehicle Age, Damage), Policy ins(Premium, sourcing channel) etc.

Our client is an Insurance company that has provided Health Insurance to its customers. Now they need the help in building a model to predict whether the policyholders (customers) from the past year will also be interested in Vehicle Insurance provided by the company.

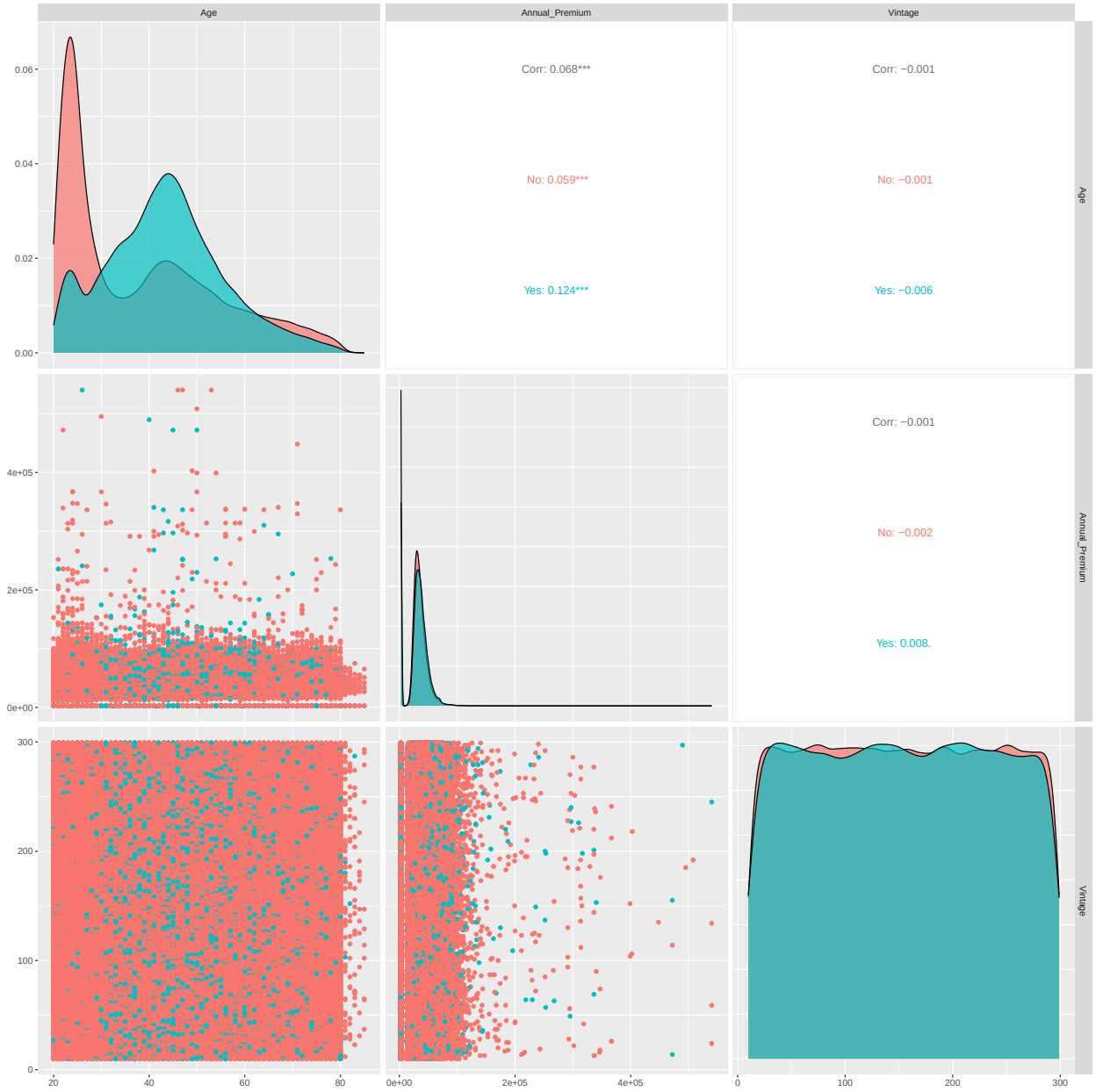
An insurance policy is an arrangement by which a company undertakes to provide a guarantee of compensation for specified loss, damage, illness, or death in return for the payment of a specified premium. A premium is a sum of money that the customer needs to pay regularly to an insurance company for this guarantee.

Building a model to predict whether a customer would be interested in Vehicle Insurance is extremely helpful for the company because it can then accordingly plan its communication strategy to reach out to those customers and optimize its business model and revenue.

Exploratory data analysis

#Numerical variables The first step is trying to get some insights about the dataset by plotting and analysing the data. We present the barplots for the categorical variables and the histograms for the numerical ones.

```
p2 <- ggpairs(train_df,
              columns = c("Age", "Annual_Premium", "Vintage"),
              aes(color = Response),
              diag = list(discrete="barDiag",
                          continuous = wrap("densityDiag", alpha=0.7)))
p2
```



The distribution of the Age variable with respect to the Response variable shows that the majority of the customers who are interested in car insurance are middle-aged (between 30 and 60 years old), which coindicdes with the age people are more likely to own a car. The customers not interested in acquiring a car insurance policy are mostly distributed among yourger people and some middle-aged adults in their 50s. The ditributin is skewd to the right.

The plot for Annual Premium suggests that the costs of the car insurance policy is independent of the interest if the customers to buy the product. It has a highly right-skewed distribution, with most of the data concentrated on the lower end of the premium scale and a long tail extending to higher premium values. The lower tail shows a high values as a consequence of entry level health insurance policy as expected.

Since both Age and Annual Premium are skewed to the right, we considered to apply a logarithm transformation for both the variables.

```r
train_df$logAge <- log(train_df$Age)
train_df$logAnnual_Premium <- log(train_df$Annual_Premium)
colnames(train_df)
```

```
##  [1] "Gender"               "Age"                  "Driving_License"
##  [4] "Region_Code"          "Previously_Insured"   "Vehicle_Age"
##  [7] "Vehicle_Damage"       "Annual_Premium"       "Policy_Sales_Channel"
## [10] "Vintage"              "Response"             "logAge"
## [13] "logAnnual_Premium"
```

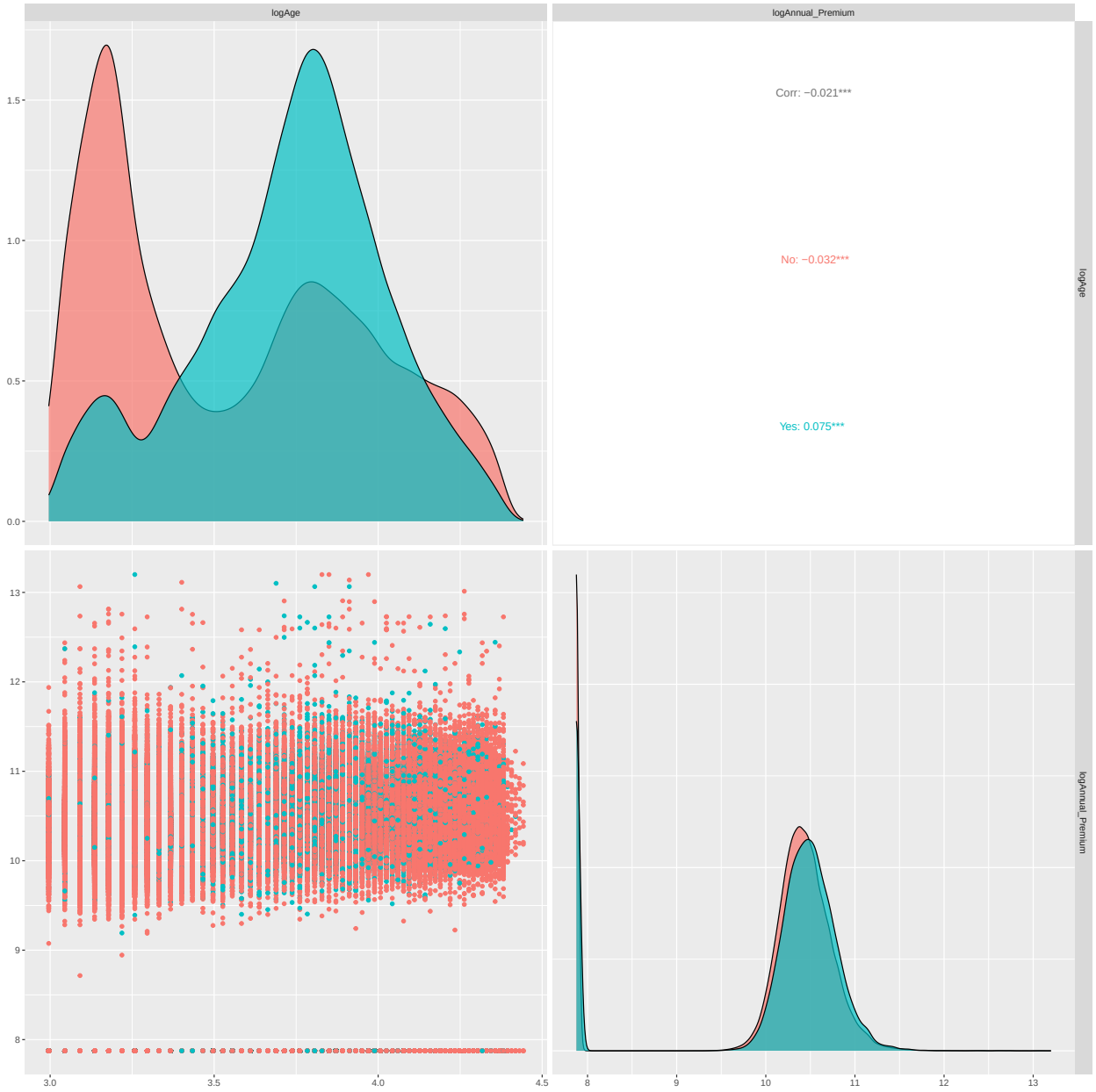
```r
sum(is.null(train_df$logAnnual_Premium))
```

```
## [1] 0
```

```r
sum(is.na(train_df$logAnnual_Premium))
```

```
## [1] 0
```

```r
p3 <- ggpairs(train_df,
              columns = c("logAge", "logAnnual_Premium"),
              aes(color = Response),
              diag = list(discrete="barDiag",
                continuous = wrap("densityDiag", alpha=0.7 )))
p3
```



The plot for Vintage shows a nearly uniform distribution, with a slight increase in frequency towards the middle range of the Vintage variable. It may not be significant in the explanation of the Response.

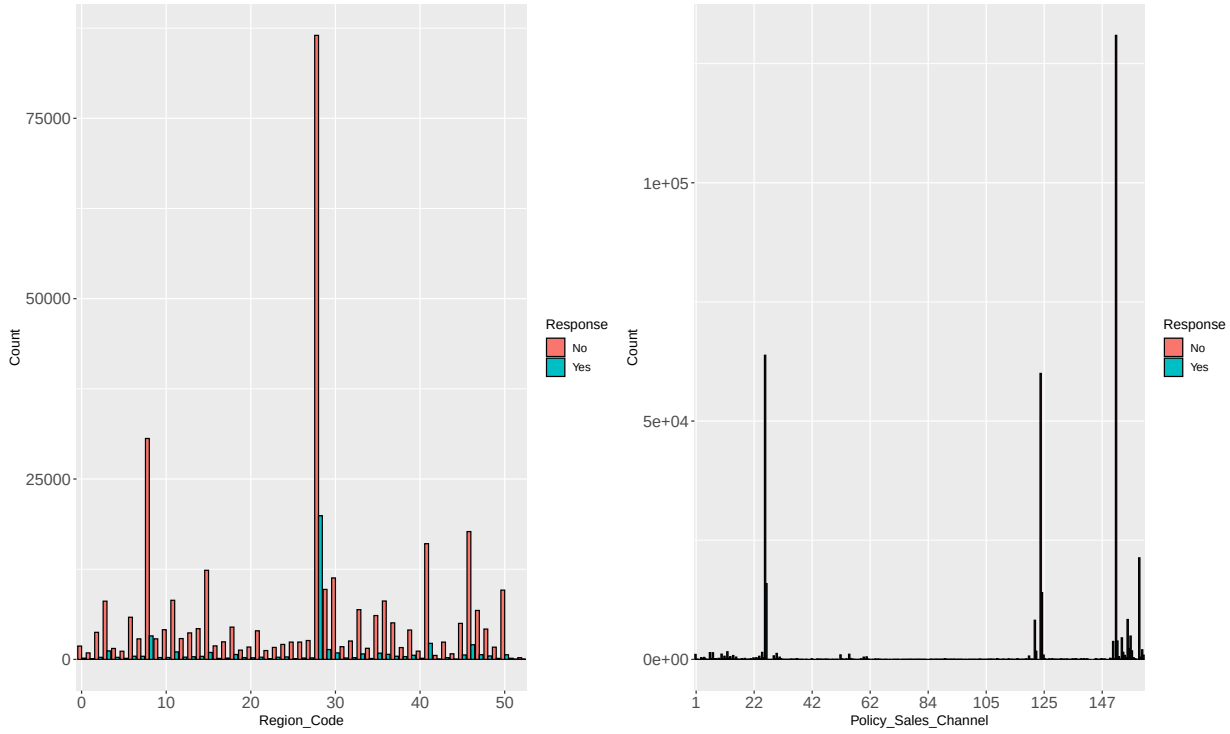
The variables show negligible linear correlation between them, which is clearly shown in the scatter plot and in the correlation coefficients.

#Categorical variables

```
p9 <- ggplot(train_df, aes(x = Region_Code, fill = Response)) +
  geom_bar(position = "dodge", color = "black") +
  labs(y = "Count", fill = "Response") +
  scale_x_discrete(breaks = function(x) x[seq(1, length(x), by = 10)]) +
  theme(
    text = element_text(size = 12),
    axis.text = element_text(size = 14)
```

```
  )


p14 <- ggplot(train_df, aes(x = Policy_Sales_Channel, fill = Response)) +
  geom_bar(position = "dodge", color = "black") +
  labs(y = "Count", fill = "Response") +
  scale_x_discrete(breaks = function(x) x[seq(1, length(x), by = 20)]) +
  theme(
    text = element_text(size = 12),
    axis.text = element_text(size = 14)
  )
p_cat <- p9 + p14
p_cat##Should be clearer
```



Regarding the variable Region_Code, we can notice that the vast majority of the customers are from region 28. The customers from region 28 are also the ones who are most interested in car insurance. Almost half of the customers are distributed among regions 8, 28, 41, 46 accounting for ~47% of the total customers. Since this variable has a lot of labels with low frequency, we decided to consider only the major four ones mentioned above and an additional one with the remaining labels as a unique category.

Also in Policy_Sales_Channel, there are four categories more frequent than others: Channels 26, 124, 152 and 160 alone account for more than 80% of the customers. Channels 26 and 124 are the ones with the highest percentage of customers interested in the product. Only about 20% of the customers interested in the product are distributed in the rest of the channels of outreach. As we did for Region_Code, we grouped the remaining less frequent categories as one.

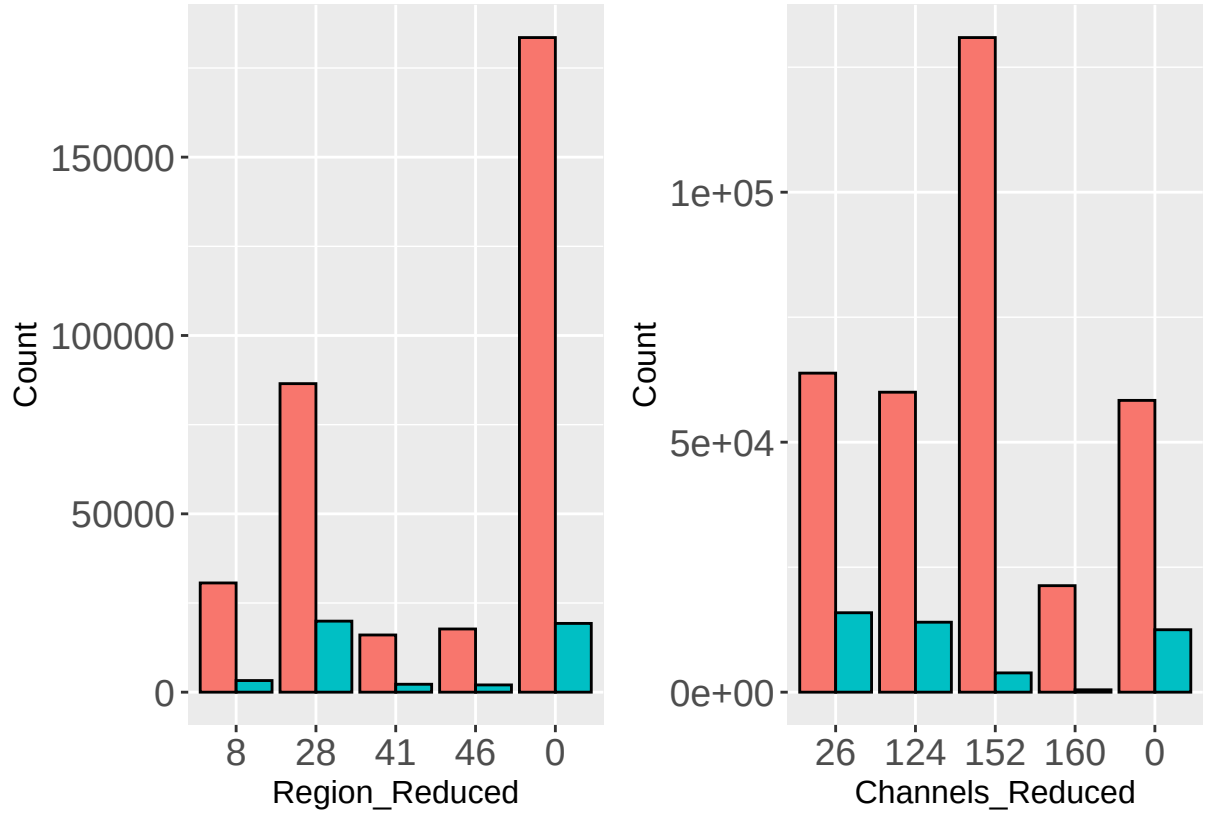
After the grouping:

```
p10 <- ggplot(train_reduced, aes(x = Region_Reduced, fill = Response)) +
  geom_bar(position = "dodge", color = "black") +
```

```
  labs(y = "Count", fill = "Response") +
  theme(
    text = element_text(size = 12),
    axis.text = element_text(size = 14)
  ) +
  guides(fill = FALSE)
```

```
## Warning: The '<scale>' argument of 'guides()' cannot be 'FALSE'. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
p15 <- ggplot(train_reduced, aes(x = Channels_Reduced, fill = Response)) +
  geom_bar(position = "dodge", color = "black") +
  labs(y = "Count", fill = "Response") +
  theme(
    text = element_text(size = 12),
    axis.text = element_text(size = 14)
  ) +
  guides(fill = FALSE)
p10+p15
```



The last categorical variable is ID, which we won't consider for models, since it is a simple identifier of the customers. Hence it doesn't give any more information about the Response.

##Train/Test Data

Among the datasets provided, the only usable one was train.csv, since test.csv dataset there was no observation saying that a customer was interested in car insurance. Hence, we made a static test/train split on train.csv with 80% of train set and 20% of test set. –add code for split??–

##MODELS After exploring the data, now we proceed with the fitting and assessment of different models for binary classification.

glm

Il nostro scopo è capire quali variabili utilizzare e come utilizzarle. -nested models statico con assessment -stepAIC() con assessment -rifare tutto con log?? oppure ne facciamo 2 in parallelo

In order to understand which variables are more significant in the explanation of the response variable, we analyzed nested models with different combinations of selected explanatory variables.

#forward selection The first approach consists of adding variables one by one, starting from an empty model and proceeding until all the variables considered in the model.

```
## Caricamento del pacchetto richiesto: qgam
## Registered S3 method overwritten by 'mgcViz':
##   method from
##   +.gg   GGally
##
## Caricamento pacchetto: 'mgcViz'
## I seguenti oggetti sono mascherati da 'package:stats':
##
##     qqline, qqnorm, qqplot
## Loaded ROSE 0.0-4
## Type 'citation("pROC")' for a citation.
##
## Caricamento pacchetto: 'pROC'
## I seguenti oggetti sono mascherati da 'package:stats':
##
##     cov, smooth, var
## Caricamento del pacchetto richiesto: lattice
##
## Caricamento pacchetto: 'lattice'
## Il seguente oggetto è mascherato da 'package:mgcViz':
##
##     qq
##
## Caricamento pacchetto: 'purrr'
## Il seguente oggetto è mascherato da 'package:caret':
##
##     lift
## Il seguente oggetto è mascherato da 'package:car':
##
##     some
## [1] "C:/Users/minut/OneDrive/Desktop/stat/git_project/stats_project/datasets"
```

```
(result <- ranking_nested_models(train_data, test_data, use_model = "glm", use_log = TRUE, use_splines =
```

```
## [1] "Response ~ Previously_Insured"
## [1] "Response ~ Previously_Insured + Vehicle_Damage"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age))"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age"
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age +
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age +
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age +
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age +
## [1] "Response ~ Previously_Insured + Vehicle_Damage + Channels_Reduced + I(log(Age)) + Vehicle_Age +

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases

## $Ranking_Variables
##                        VariableRemoved      AIC
## Previously_Insured Previously_Insured 430426.6
## Vehicle_Damage         Vehicle_Damage 420508.0
## Channels_Reduced     Channels_Reduced 413558.7
## Age                               Age 408665.8
## Vehicle_Age               Vehicle_Age 405994.2
## Region_Reduced         Region_Reduced 405939.6
## Driving_License       Driving_License 405413.0
## Gender                         Gender 405376.7
## Annual_Premium         Annual_Premium 405283.0
## Vintage                       Vintage 405238.5
##
## $Results
##            Model_Name      AIC       AUC  Accuracy       TPR       FPR
## 1  Previously_Insured 444030.7 0.7587983 0.7579950 0.9964179 0.4788213
## 2      Vehicle_Damage 423425.1 0.7884777 0.7823815 0.9767265 0.4106539
## 3    Channels_Reduced 409749.1 0.8263847 0.7895813 0.9676012 0.3872391
## 4                 Age 407152.7 0.8413283 0.7904738 0.9659002 0.3837706
## 5         Vehicle_Age 406354.9 0.8414121 0.7901796 0.9669008 0.3853508
## 6      Region_Reduced 405589.6 0.8422240 0.7914460 0.9016530 0.3180183
## 7     Driving_License 405419.6 0.8424160 0.7917552 0.9008025 0.3165573
## 8              Gender 405281.4 0.8423033 0.7917103 0.9092774 0.3250646
## 9      Annual_Premium 405238.5 0.8426175 0.7915258 0.9082668 0.3244285
## 10            Vintage 405240.0 0.8426163 0.7915358 0.9059855 0.3221427
##          TNR         FNR Precision Threshold
## 1  0.5211787 0.003582078 0.6739441 0.3412586
## 2  0.5893461 0.023273499 0.7025969 0.4603892
## 3  0.6127609 0.032398791 0.7127989 0.4108390
## 4  0.6162294 0.034099778 0.7142783 0.4351748
## 5  0.6146492 0.033099198 0.7136506 0.4471742
## 6  0.6819817 0.098347041 0.7379537 0.5606677
## 7  0.6834427 0.099197535 0.7386610 0.5606885
## 8  0.6749354 0.090722619 0.7353358 0.5479753
## 9  0.6755715 0.091733205 0.7355005 0.5512659
## 10 0.6778573 0.094014528 0.7363858 0.5574755
```

#stepAIC The second approach consists of using the function stepAIC() from the MASS package to find the best combination of predictors with respect to AIC. The stepAIC() function must be applayed to the full model, which serves as the starting point for the variable selection process. We chose the 'both' direction, that considers both adding and removing variables from the model.

training

testing

comparison

gam

training

testing

comparison

RF??

Final conclusions

(skim)We observe that there are not missing values and that the Response variable has an inbalance rate of 7.15 (put code) so we don't need to act to fix that.