HarvardX | PH125.9x | CASDatasets Project - Claims Frequency

Reema Sajnani

2024-05-22

Contents

1 EXECUTIVE SUMMARY

It is important for insurers to understand what portion of their policy holders will file claims and, how often. This project puts us in the shoes of insurers and analyzes third party motor insurance claims made in France. The goal is to predict how frequently insurance claims are made on policies, with as much accuracy as possible. In order to test accuracy we will aim to have a Mean Absolute Error closer to 0.

The particular data sets selected from CASdatasets (hosted by Christophe Dutang) and are referred to as FreMTPL2freq (frequency data set) and FreMTPL2sev (severity data set). The former comprises data on insurance policy and claim frequency, while the latter includes information on the associated claim severity amounts.

This project uses Generalized Linear Model and a Decision Tree model. These models obtain Mean Absolute Errors of 7.29% and 11.09% respectively. The GLM is our model of choice due to it's low error value, making it a good fit.

2 PACKAGE INSTALLATION AND DATA SETUP

```
if(!require(knitr)) install.packages("knitr", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: knitr
```

```
## Warning: package 'knitr' was built under R version 4.3.3
```

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
if(!require(lattice)) install.packages("lattice", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 4.3.3
```

```
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: caret
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
if(!require(rpart)) install.packages("rpart", repos = "https://cran.r-project.org/package=rpart")
```

```
## Loading required package: rpart
```

```
## Warning: package 'rpart' was built under R version 4.3.3
```

```
if(!require(rpart.plot)) install.packages("rpart.plot", repos = "https://CRAN.R-project.org")
```

```
## Loading required package: rpart.plot
```

```
## Warning: package 'rpart.plot' was built under R version 4.3.3
```

```
if(!require(repr)) install.packages("repr", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: repr
```

```
## Warning: package 'repr' was built under R version 4.3.3
```

```
if(!require(xts)) install.packages("xts", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.3.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.3.3
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## ########################## Warning from 'xts' package ##########################
## #                                                                              #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                             #
## #                                                                              #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop            #
## # dplyr from breaking base R's lag() function.                                 #
## #                                                                              #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                              #
```

```
## ################################################################################
##
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

```
if(!require(sp)) install.packages("sp", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: sp

## Warning: package 'sp' was built under R version 4.3.3
```

```
if(!require(zoo)) install.packages("zoo", repos = "http://cran.us.r-project.org")
if(!require(modelr)) install.packages("modelr", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: modelr

## Warning: package 'modelr' was built under R version 4.3.3
```

```
install.packages("CASdatasets", repos = "http://cas.uqam.ca/pub/", type="source")
```

```
## Installing package into 'C:/Users/Reema/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
if(!require(AER)) install.packages("AER", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: AER

## Warning: package 'AER' was built under R version 4.3.3

## Loading required package: car

## Warning: package 'car' was built under R version 4.3.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.3

##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
##
## Loading required package: lmtest
```

```
## Warning: package 'lmtest' was built under R version 4.3.3

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.3.3

## Loading required package: survival
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:caret':
##
##     cluster
```

```
library("knitr")
library("tidyverse")
library("dplyr")
library("lattice")
library("caret")
library("rpart")
library("rpart.plot")
library("repr")
library("modelr")
library("CASdatasets")
library("AER")
```

3 LOAD DATA SETS

```
#load the two data sets and convert into tibbles
data(freMTPL2freq)
data <- as_tibble(freMTPL2freq)
data(freMTPL2sev)
sev<- as_tibble(freMTPL2sev)
```

4 DATA EXPLORATION

4.1 Data Class and Details

We first explore the data sets.

freMTPL2freq (frequency data set) - this has 12 vectors and 678,013 rows of observations. Below we explore their class and description.

```
## Dimensions 678013 12
```

CLASS:

- **IDpol** `<numeric>` unique Policy ID
- **ClaimNb** `<table>` Number of claims during the exposed time period
- **Exposure** `<numeric>` time period (years)
- **VehPower** `<integer>` Power of the insured car
- **VehAge** `<integer>` Vehicle age (in years)
- **DrivAge** `<integer>` Driver age (in years)
- **BonusMalus** `<integer>` Bonus/malus, between 50 and 230
- **VehBrand** `<factor>` Car Brand
- **VehGas** `<character>` Car Fuel Type
- **Area** `<factor>` "A" for rural to "F" for urban centre
- **Density** `<integer>` Population density in the area where driver resides
- **Region** `<factor>` Policy region in France

freMTPL2sev (severity data set) - this has 2 vectors and 26,639 rows of observations. Below we explore their class and description.

```
## Dimensions 26639 2
```

CLASS:

- **IDpol** `<numeric>` unique Policy ID
- **ClaimAmount** `<table>` Amount per claim made

4.2 Data Clean-up and Modifications

We start by making sure all columns are in a format that can be filtered and used ahead for visualization and model calculation.

```
## This is what the frequency data set looks like after changing the class of VehGas vector  and double
```

```
## # A tibble: 6 x 12
##   IDpol ClaimNb Exposure VehPower VehAge DrivAge BonusMalus VehBrand VehGas
##   <dbl>   <dbl>    <dbl>    <int>  <int>   <int>      <int> <fct>    <fct>
## 1     1       1     0.1         5      0      55         50 B12      Regular
## 2     3       1     0.77        5      0      55         50 B12      Regular
## 3     5       1     0.75        6      2      52         50 B12      Diesel
## 4    10       1     0.09        7      0      46         50 B12      Diesel
## 5    11       1     0.84        7      0      46         50 B12      Diesel
## 6    13       1     0.52        6      2      38         50 B12      Regular
## # i 3 more variables: Area <fct>, Density <int>, Region <fct>
```

We will add claim numbers to severity data set and join these into the frequency data set. This is because there are many rows in the frequency dataset which show a claim was made, however, their corresponding claim amounts are not mentioned in the severity data set. Vice-Versa also applies.

```
## Updated severity data set with claim frequencies:
```

```
##   IDpol ClaimNb
## 1   139       1
## 2   190       1
## 3   414       1
## 4   424       2
## 5   463       1
## 6   606       1
```

Joining the frequency and severity data sets ensures we work only with those policy IDs that have valid corresponding claim entries in both sets.

```
## Sample View of the Joint data set after taking claim frequencies from the severity data set:
```

```
## # A tibble: 6 x 12
##   IDpol Exposure VehPower VehAge DrivAge BonusMalus VehBrand VehGas  Area
##   <dbl>    <dbl>    <int>  <int>   <int>      <int> <fct>    <fct>   <fct>
## 1     1     0.1         5      0      55         50 B12      Regular D
## 2     3     0.77        5      0      55         50 B12      Regular D
## 3     5     0.75        6      2      52         50 B12      Diesel  B
## 4    10     0.09        7      0      46         50 B12      Diesel  B
## 5    11     0.84        7      0      46         50 B12      Diesel  B
## 6    13     0.52        6      2      38         50 B12      Regular E
## # i 3 more variables: Density <int>, Region <fct>, ClaimNb <dbl>
```

4.3 Data Insights

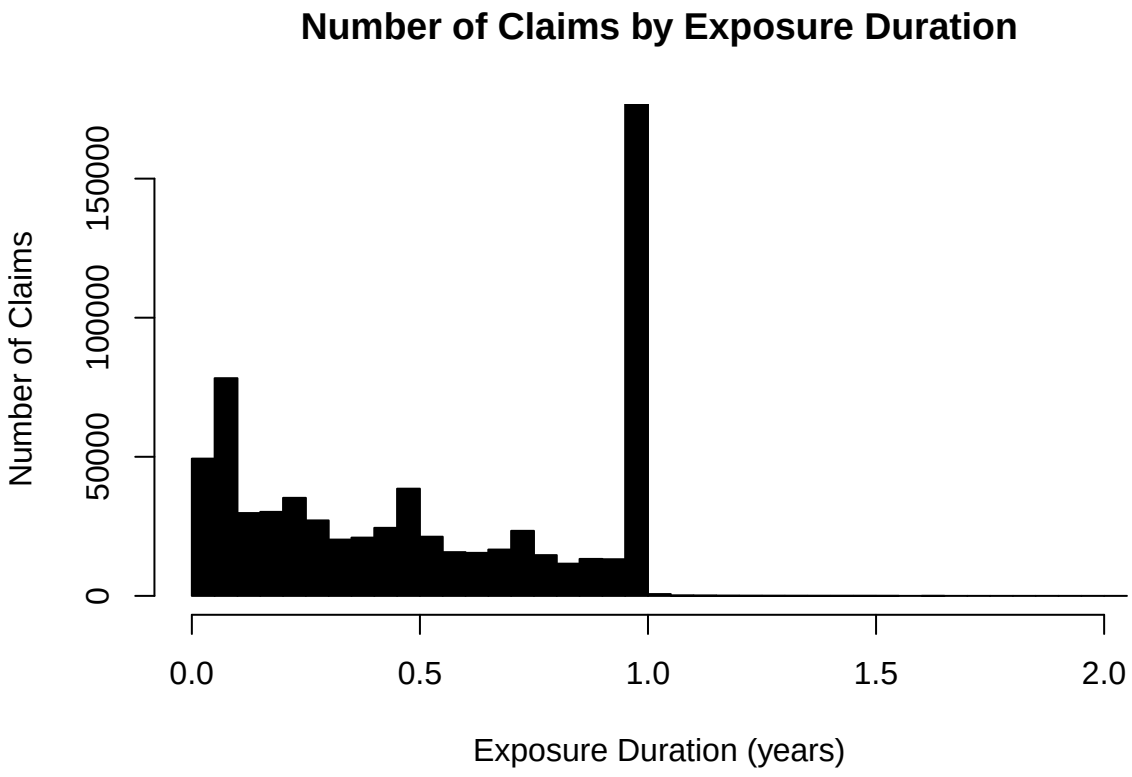
4.3.1 Number of Claims

Only about 3.67% of policies seem to have made claims, with one claim being the most commonly occurring (3.48% policies).

```
## # A tibble: 11 x 3
##    ClaimNb  count   percent
##      <dbl>  <int>     <dbl>
##  1       0 653069 96.3
##  2       1  23571  3.48
##  3       2   1298  0.191
##  4       3     62  0.00914
##  5       4      5  0.000737
##  6       5      2  0.000295
##  7       6      1  0.000147
##  8       8      1  0.000147
##  9       9      1  0.000147
## 10      11      2  0.000295
## 11      16      1  0.000147
```

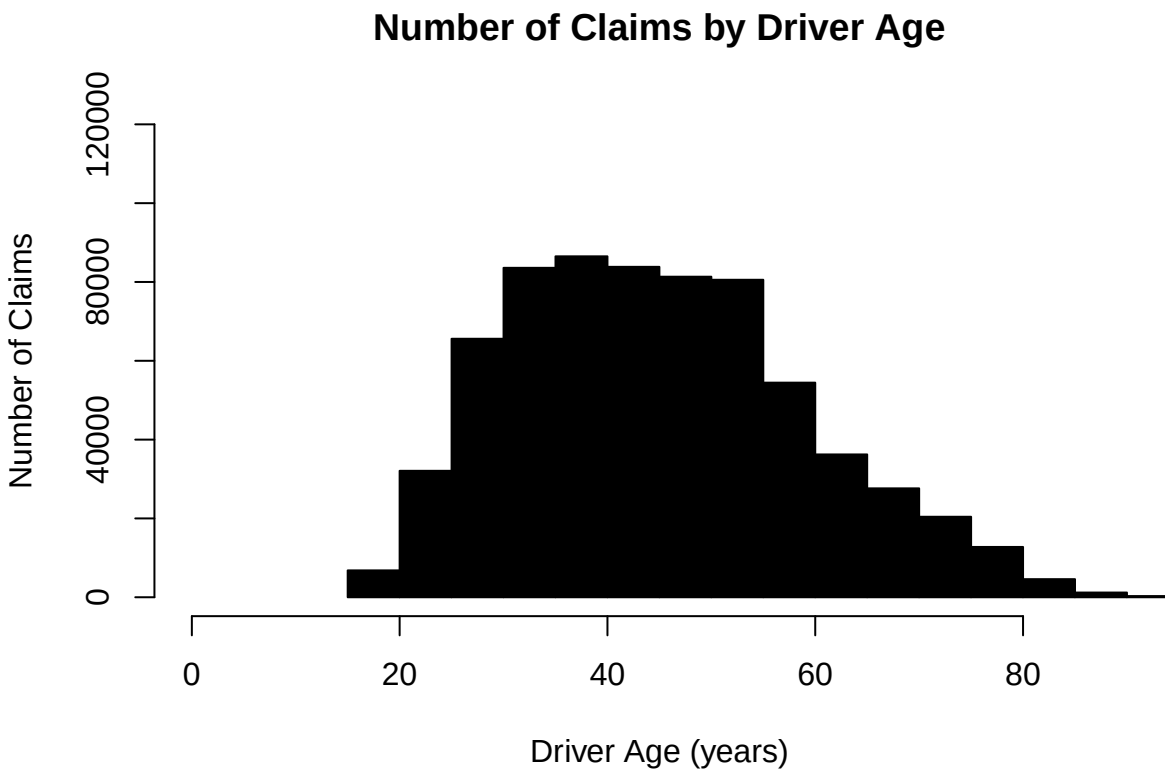
4.3.2 Policy Exposures

In this histogram, we notice that very few cases have exposures durrations greater than one year.



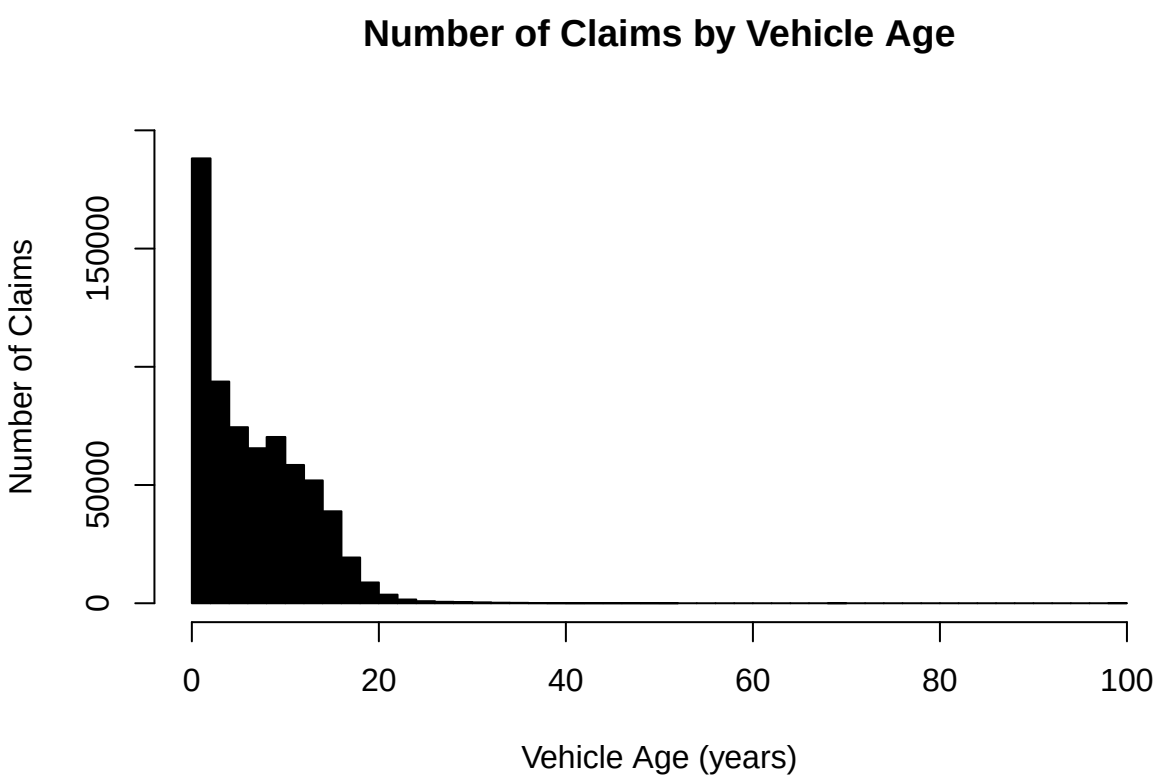
4.3.3 Drivers' Age

We notice that there are very few instances of drivers above the age of 90, making claims.



4.3.4 Vehicle Age

In this case it is visible that very few claims are made on cars that are more than 20 years old.



4.4 Data Pre-Processing

Based on our visualization above, we make some corrections to the data in order to make our analysis more effective. Next steps will include:

a. Correcting number of claims above 4 to equal to 4. We believe greater than 4 claims are highly unlikely and are caused from data entry errors

b. Correcting exposures over 1 year, to 1 year. Since this is yearly claim data set, exposure greater than 1 year could be from data entry errors

c. Correcting all Driver Ages above 90 years, to 90. It is unlikely for someone at that age to be driving

d. Correcting Vehicle Ages over 40 years, to 40 year. It is unlikely cars last that long and we assume these entries are from data entry errors

```
## After making the above changes, the data set looks something like this:

## # A tibble: 678,013 x 12
##    VehBrand ClaimNb DrivAge Exposure Area  BonusMalus IDpol Region        VehGas
##    <fct>      <dbl>   <dbl>    <dbl> <fct>      <int> <dbl> <fct>         <fct>
##  1 B12            0      55     0.1  D             50     1 Rhone-Alpes   Regul~
##  2 B12            0      55     0.77 D             50     3 Rhone-Alpes   Regul~
##  3 B12            0      52     0.75 B             50     5 Picardie      Diesel
##  4 B12            0      46     0.09 B             50    10 Aquitaine     Diesel
```

```
##  5 B12            0      46     0.84 B             50    11 Aquitaine     Diesel
##  6 B12            0      38     0.52 E             50    13 Nord-Pas-de-~ Regul~
##  7 B12            0      38     0.45 E             50    15 Nord-Pas-de-~ Regul~
##  8 B12            0      33     0.27 C             68    17 Languedoc-Ro~ Diesel
##  9 B12            0      33     0.71 C             68    18 Languedoc-Ro~ Diesel
## 10 B12            0      41     0.15 B             50    21 Pays-de-la-L~ Diesel
## # i 678,003 more rows
## # i 3 more variables: Density <int>, VehAge <dbl>, VehPower <int>
```

4.5 Impact on Claim Rate

So far we saw the number of claims made by sub groups in each vector. Now we will analyze how frequently these groups make claims.

We start by adding a column for the claim frequency of every entry.

```
## the New Joint data set looks something like this:
```

```
## # A tibble: 678,013 x 13
##     Freq VehBrand ClaimNb VehPower BonusMalus Region   VehGas IDpol Area  VehAge
##    <dbl> <fct>      <dbl>    <int>      <int> <fct>    <fct>  <dbl> <fct>  <dbl>
##  1     0 B12            0        5         50 Rhone-A~ Regul~     1 D          0
##  2     0 B12            0        5         50 Rhone-A~ Regul~     3 D          0
##  3     0 B12            0        6         50 Picardie Diesel     5 B          2
##  4     0 B12            0        7         50 Aquitai~ Diesel    10 B          0
##  5     0 B12            0        7         50 Aquitai~ Diesel    11 B          0
##  6     0 B12            0        6         50 Nord-Pa~ Regul~    13 E          2
##  7     0 B12            0        6         50 Nord-Pa~ Regul~    15 E          2
##  8     0 B12            0        7         68 Langued~ Diesel    17 C          0
##  9     0 B12            0        7         68 Langued~ Diesel    18 C          0
## 10     0 B12            0        7         50 Pays-de~ Diesel    21 B          0
## # i 678,003 more rows
## # i 3 more variables: Density <int>, DrivAge <dbl>, Exposure <dbl>
```

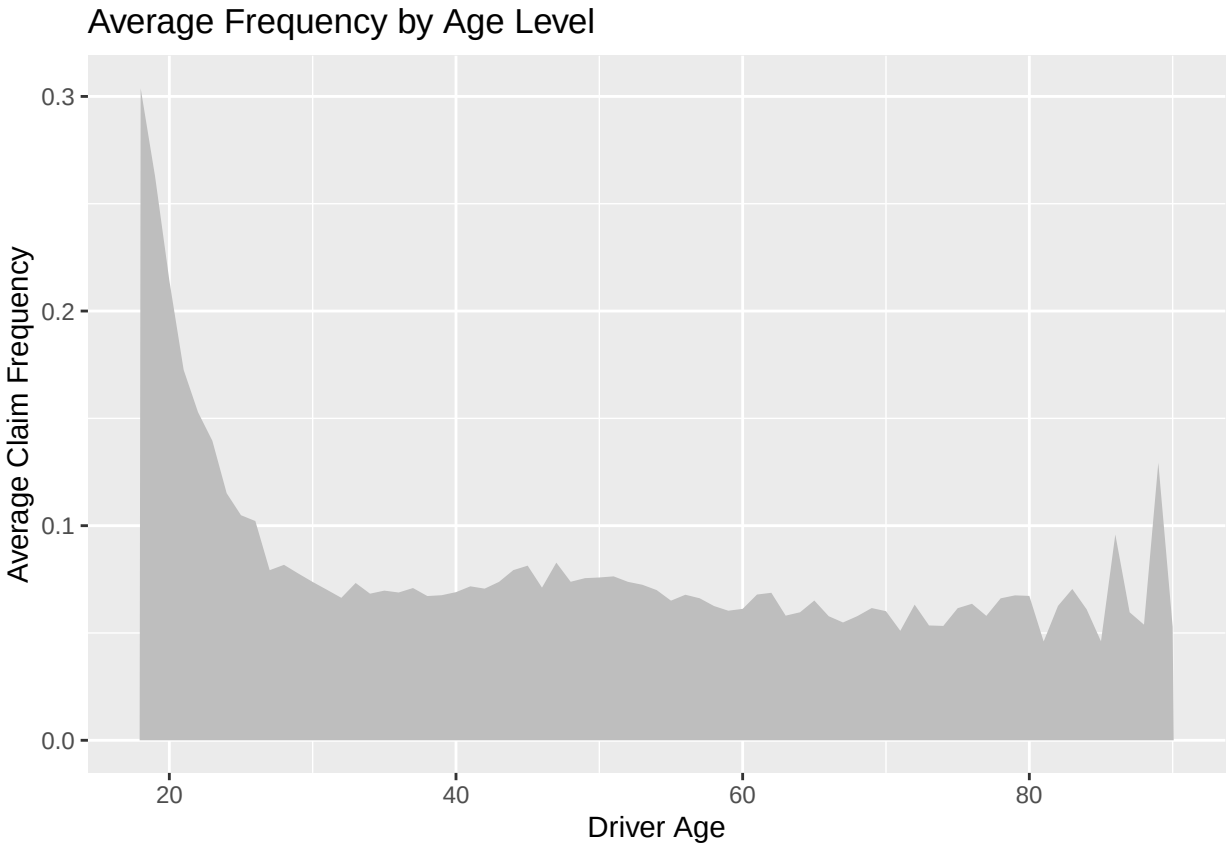
4.5.1 Driver Age

First we analyze driver age impact on claim frequency.

From the below plot, it is clear that the highest claim frequency is coming from the age group under 25, with 18 year-olds being the most frequent claimants.

```
## These are the top 5 age groups with highest average claim rates:
```

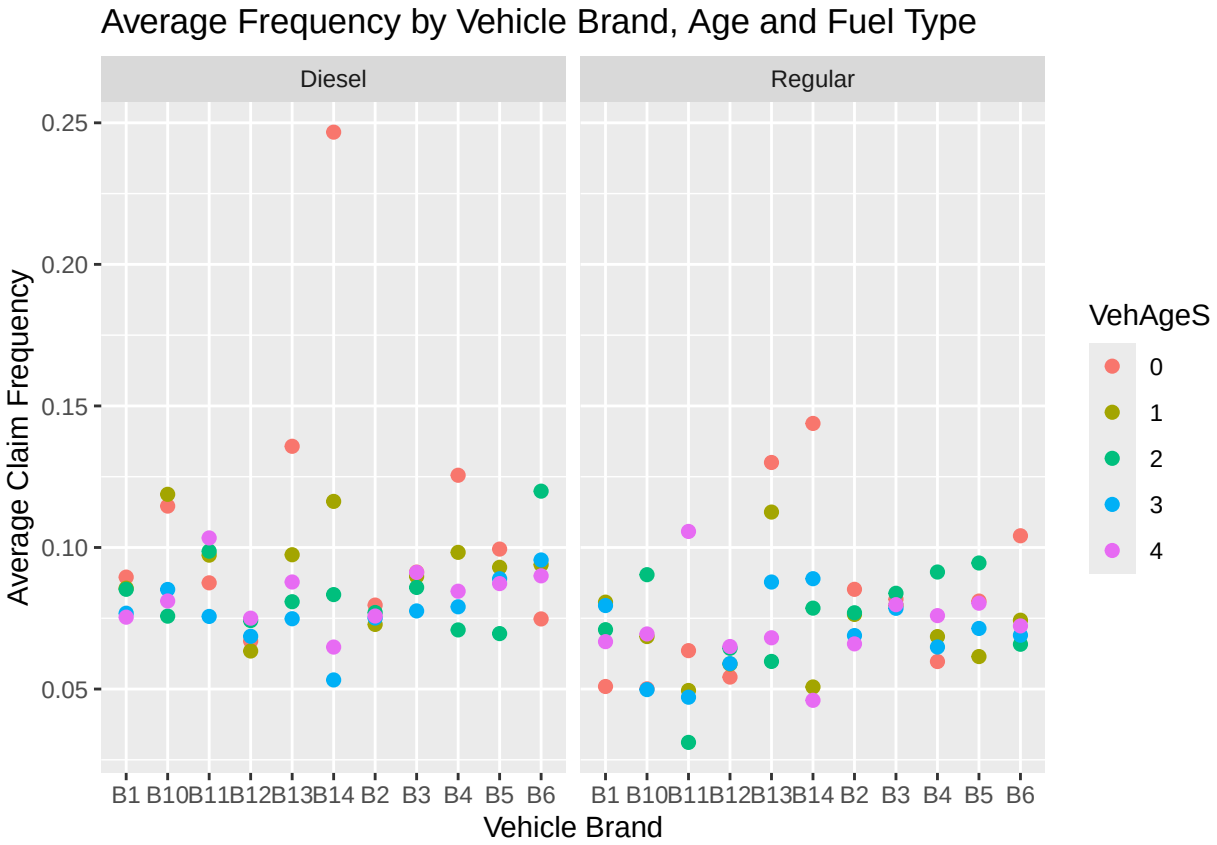
```
## # A tibble: 5 x 2
##   DrivAge claim_frequency
##     <dbl>           <dbl>
## 1      18           0.304
## 2      19           0.263
## 3      20           0.214
## 4      21           0.173
## 5      22           0.153
```



4.5.2 Fuel Type, Vehicle Brand and Vehicle Age

When assessing Fuel Type and Car Brand groups, we notice overall lower frequencies for regular fuel cars compared to diesel powered ones. We also see that vehicles that are newer (orange plot dot in the graph), seem to be the ones claiming more often.

```
## `summarise()` has grouped output by 'VehBrand', 'VehAgeS'. You can override
## using the `.groups` argument.
```

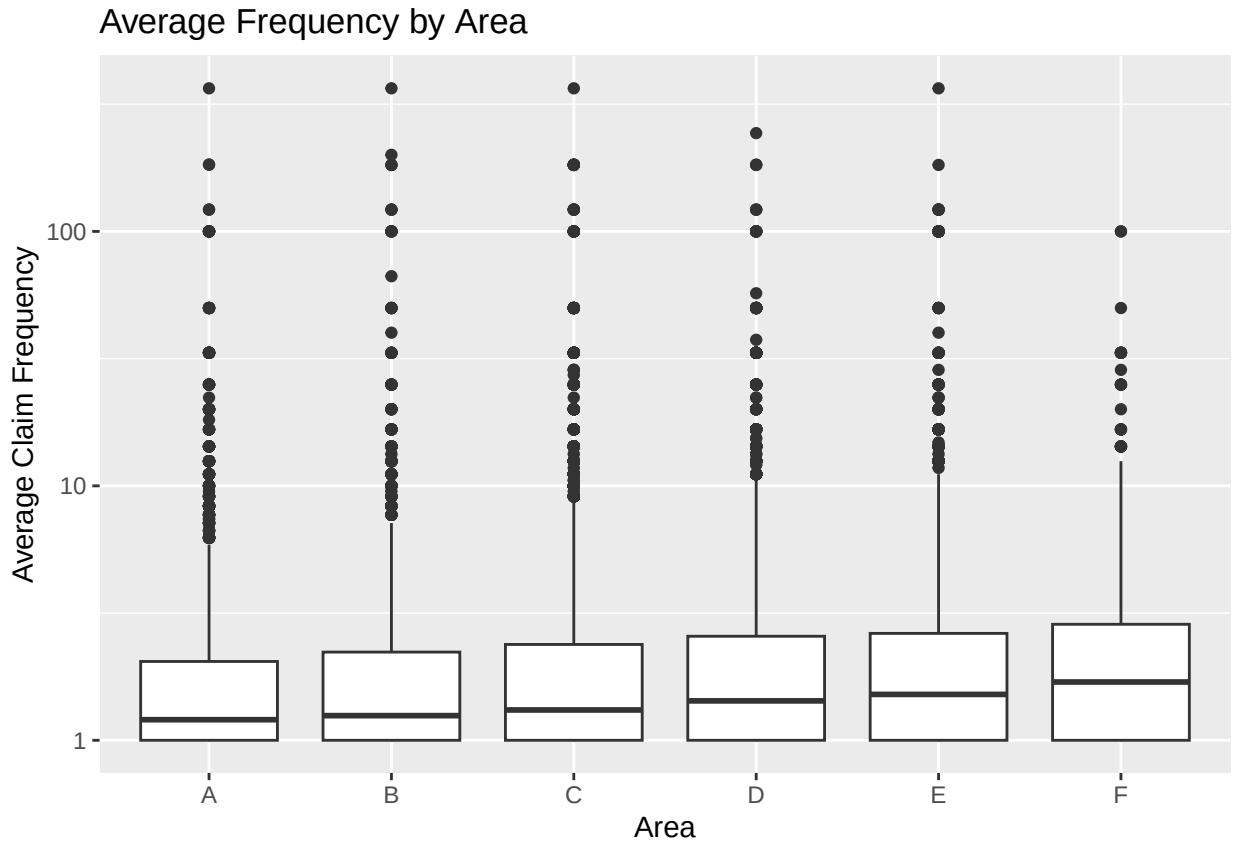


4.5.3 Area

Looking at the below chart, it is clear that claims flow in more often from urban areas (Area F). This Area also has an overall higher median claim frequency.

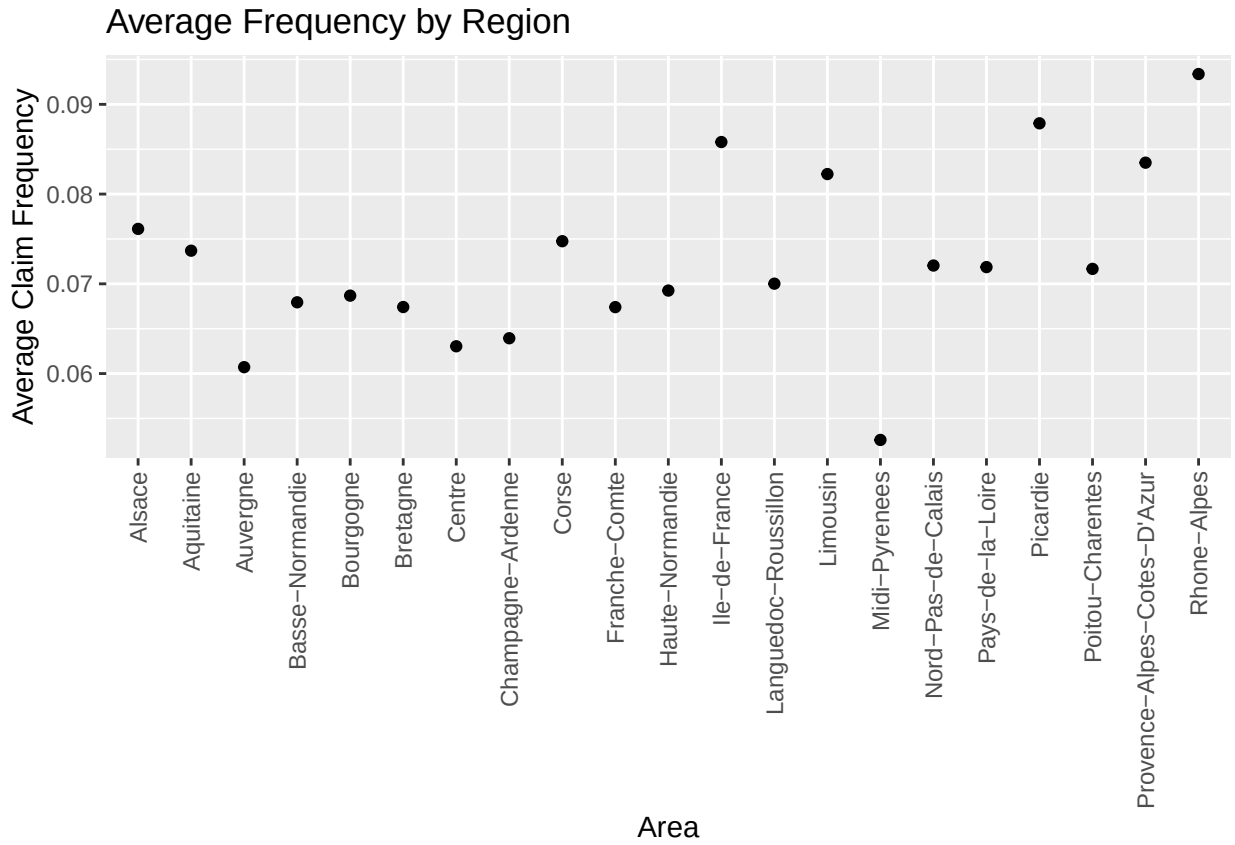
```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```

```
## Warning: Removed 653069 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



4.5.4 Region

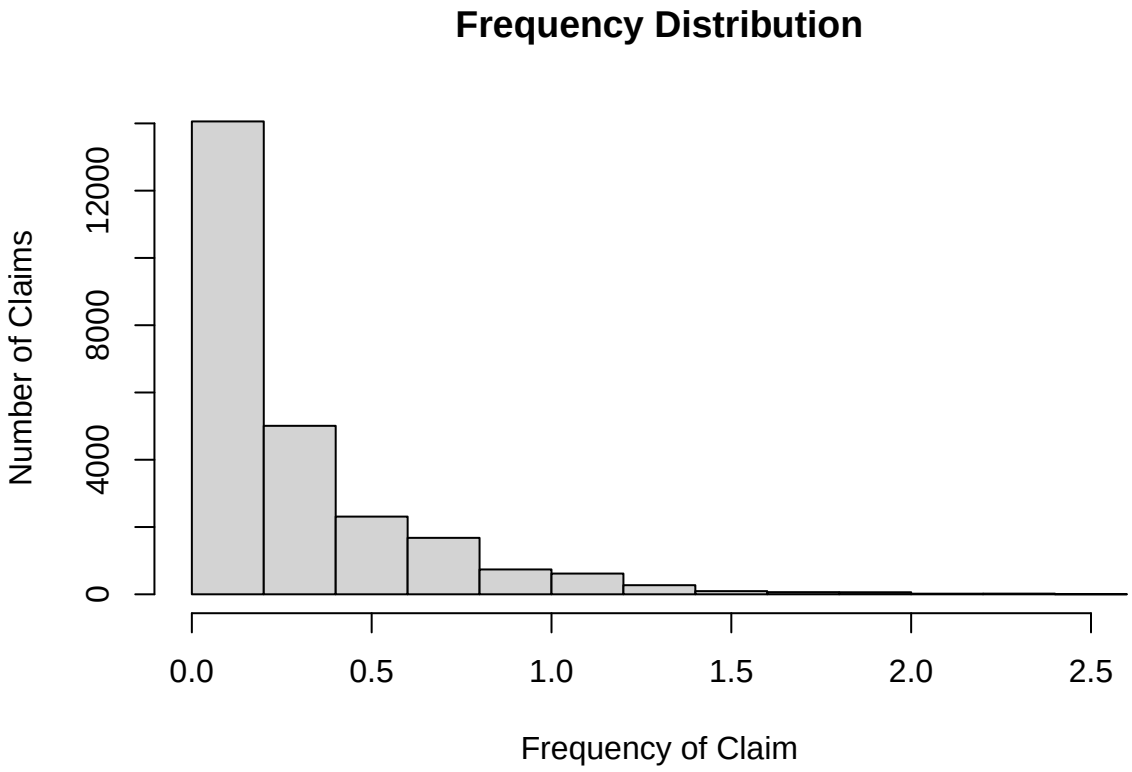
Rhone-Alps, Ile-de-France and Picardie regions have the highest claims frequency. One of these is a major city, which could be a reason for the higher claim rate.


Average Frequency by Region
Average Claim Frequency
0.09
0.08
0.07
0.06
Alsace
Aquitaine
Auvergne
Basse−Normandie
Bourgogne
Bretagne
Centre
Champagne−Ardenne
Corse
Franche−Comte
Haute−Normandie
Ile−de−France
Languedoc−Roussillon
Limousin
Midi−Pyrenees
Nord−Pas−de−Calais
Pays−de−la−Loire
Picardie
Poitou−Charentes
Provence−Alpes−Cotes−D'Azur
Rhone−Alpes
Area

5 DATA MODELLING

5.1 Distribution Type

From the below histogram we conclude that our data is based on a Poisson Distribution (rate related data). This will play an important role in the model we choose later in this project and the arguments to be included in the code. This also means that Poisson GLM will be a natural first modelling approach.



5.2 Train and Test Sets

We split our data into a 70:30 split to ensure there is enough representation of variability in the test set.

NOTE: We also tried 80:20 and 90:10 splits but got very poor results. Hence we conclude that 70:30 is a good representation.

The train set has 203404 observations.

5.3 Generalized Linear Model

They key assumptions for this model are:

a. the relationship between number of claims and log(claim frequency) is linear

b. each claim outcome are independent of one another

c. mean is equal to the variance

Formula applicable to Poisson Distribution

$$log(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$$

Where:

y: Is the rate variable or 'Lambda'. In our case this is the claim frequency rate

Beta: numeric coefficients, Beta0 being the intercept

x: predictor variables

Since this is rate based data, we need an offset of the logarithmic denominator. So, customizing the above equation for our model, we use this formula, where log(Exposure) is the offset.

$$log(ClaimNb) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + log(Exposure)$$

After fitting the Generalized Linear Model (GLM), we are able to see the most significant objects.

```
##
## Call:
## glm(formula = ClaimNb ~ VehPower + VehAge + DrivAge + BonusMalus +
##     VehBrand + VehGas + Area + Density + Region + offset(log(Exposure)),
##     family = "poisson", data = train_set)
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -4.735e+00  1.369e-01 -34.576  < 2e-16 ***
## VehPower                       3.734e-02  3.932e-03   9.497  < 2e-16 ***
## VehAge                        -1.720e-02  1.541e-03 -11.159  < 2e-16 ***
## DrivAge                        5.116e-03  5.517e-04   9.273  < 2e-16 ***
## BonusMalus                     2.632e-02  3.941e-04  66.773  < 2e-16 ***
## VehBrandB10                   -9.041e-03  4.704e-02  -0.192  0.84758
## VehBrandB11                    1.219e-01  5.041e-02   2.419  0.01556 *
## VehBrandB12                   -3.062e-01  2.640e-02 -11.602  < 2e-16 ***
## VehBrandB13                    1.372e-02  5.436e-02   0.252  0.80081
## VehBrandB14                   -1.757e-01  1.060e-01  -1.658  0.09730 .
## VehBrandB2                     8.324e-03  2.043e-02   0.407  0.68375
## VehBrandB3                     4.250e-02  2.855e-02   1.489  0.13659
## VehBrandB4                    -6.084e-03  3.929e-02  -0.155  0.87694
## VehBrandB5                     9.005e-02  3.260e-02   2.763  0.00573 **
## VehBrandB6                     1.486e-02  3.734e-02   0.398  0.69056
## VehGasRegular                 -1.913e-01  1.521e-02 -12.574  < 2e-16 ***
## AreaB                          1.256e-01  3.147e-02   3.990 6.60e-05 ***
## AreaC                          1.737e-01  2.625e-02   6.620 3.60e-11 ***
## AreaD                          3.261e-01  2.777e-02  11.743  < 2e-16 ***
## AreaE                          4.334e-01  3.600e-02  12.038  < 2e-16 ***
## AreaF                          5.187e-01  1.238e-01   4.188 2.81e-05 ***
## Density                       -3.571e-06  5.112e-06  -0.699  0.48475
## RegionAquitaine                8.312e-02  1.295e-01   0.642  0.52110
## RegionAuvergne                -3.908e-02  1.603e-01  -0.244  0.80734
## RegionBasse-Normandie         -2.397e-02  1.368e-01  -0.175  0.86090
## RegionBourgogne               -3.304e-03  1.406e-01  -0.023  0.98126
## RegionBretagne                 5.341e-02  1.273e-01   0.420  0.67481
## RegionCentre                   1.665e-02  1.255e-01   0.133  0.89443
## RegionChampagne-Ardenne        2.743e-02  1.854e-01   0.148  0.88240
```

```
## RegionCorse                          1.801e-01  1.626e-01   1.108  0.26796
## RegionFranche-Comte                 -6.481e-03  2.292e-01  -0.028  0.97744
## RegionHaute-Normandie                3.496e-02  1.474e-01   0.237  0.81260
## RegionIle-de-France                  3.485e-02  1.274e-01   0.274  0.78442
## RegionLanguedoc-Roussillon           2.518e-02  1.298e-01   0.194  0.84623
## RegionLimousin                       3.530e-01  1.519e-01   2.324  0.02015 *
## RegionMidi-Pyrenees                 -2.169e-01  1.389e-01  -1.562  0.11837
## RegionNord-Pas-de-Calais            -9.876e-03  1.283e-01  -0.077  0.93864
## RegionPays-de-la-Loire               6.266e-02  1.278e-01   0.490  0.62393
## RegionPicardie                       1.087e-01  1.418e-01   0.767  0.44334
## RegionPoitou-Charentes               8.339e-02  1.314e-01   0.634  0.52582
## RegionProvence-Alpes-Cotes-D'Azur    1.425e-01  1.260e-01   1.131  0.25816
## RegionRhone-Alpes                    2.536e-01  1.255e-01   2.021  0.04327 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 119900  on 474608  degrees of freedom
## Residual deviance: 114732  on 474567  degrees of freedom
## AIC: 150357
##
## Number of Fisher Scoring iterations: 6
```

From the table above we can infer that the most significant predictors are Vehicle Power, Vehicle Age, Driver Age, Bonus Malus, a couple of Car Brands and all areas (except A). We also noticed the difference between the residual deviance and degrees of freedom, leading us to a dispersion test.

```
##
##  Overdispersion test
##
## data:  train_glm_offset
## z = 6.1563, p-value = 3.723e-10
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##     alpha
## 0.1059802
```

Since alpha is quite close to 0 we don't have to worry about dispersion here and it is already a good sign of model fitting our data well.

As a next step, we must take exponential of the GLM fit summary table to ensure the formula mentioned at the start of the model is adhered.

```
##                       (Intercept)                          VehPower
##                             0.009                             1.038
##                           VehAge                           DrivAge
##                             0.983                             1.005
##                        BonusMalus                      VehBrandB10
##                             1.027                             0.991
##                      VehBrandB11                      VehBrandB12
##                             1.130                             0.736
##                      VehBrandB13                      VehBrandB14
##                             1.014                             0.839
```

```
##                         VehBrandB2                         VehBrandB3
##                              1.008                              1.043
##                         VehBrandB4                         VehBrandB5
##                              0.994                              1.094
##                         VehBrandB6                       VehGasRegular
##                              1.015                              0.826
##                              AreaB                              AreaC
##                              1.134                              1.190
##                              AreaD                              AreaE
##                              1.386                              1.543
##                              AreaF                            Density
##                              1.680                              1.000
##                   RegionAquitaine                   RegionAuvergne
##                              1.087                              0.962
##             RegionBasse-Normandie                   RegionBourgogne
##                              0.976                              0.997
##                    RegionBretagne                     RegionCentre
##                              1.055                              1.017
##           RegionChampagne-Ardenne                      RegionCorse
##                              1.028                              1.197
##               RegionFranche-Comte              RegionHaute-Normandie
##                              0.994                              1.036
##               RegionIle-de-France          RegionLanguedoc-Roussillon
##                              1.035                              1.025
##                     RegionLimousin                RegionMidi-Pyrenees
##                              1.423                              0.805
##          RegionNord-Pas-de-Calais             RegionPays-de-la-Loire
##                              0.990                              1.065
##                     RegionPicardie             RegionPoitou-Charentes
##                              1.115                              1.087
## RegionProvence-Alpes-Cotes-D'Azur                  RegionRhone-Alpes
##                              1.153                              1.289
```

We provide couple of examples below of interpreting the coefficients:

- If Vehicle Age increases by 1, the claim frequency goes up by a factor of 0.983
- If Vehicle Power goes up by 1, the claim frequency goes up by a factor of 1.038
- If a driver resides in Rhone-Alpes, the claim frequency goes up by a factor of 1.289

The next step was to calculate predictions and analyze the model's capability to predict on the test set using the below code.

A sample of predictions generated are seen here:

```
## 103230  85624  89437 108287 196069 106591
##  0.028  0.042  0.061  0.003  0.019  0.005
```

We will use Mean Absolute Error to evaluate accuracy of the model.

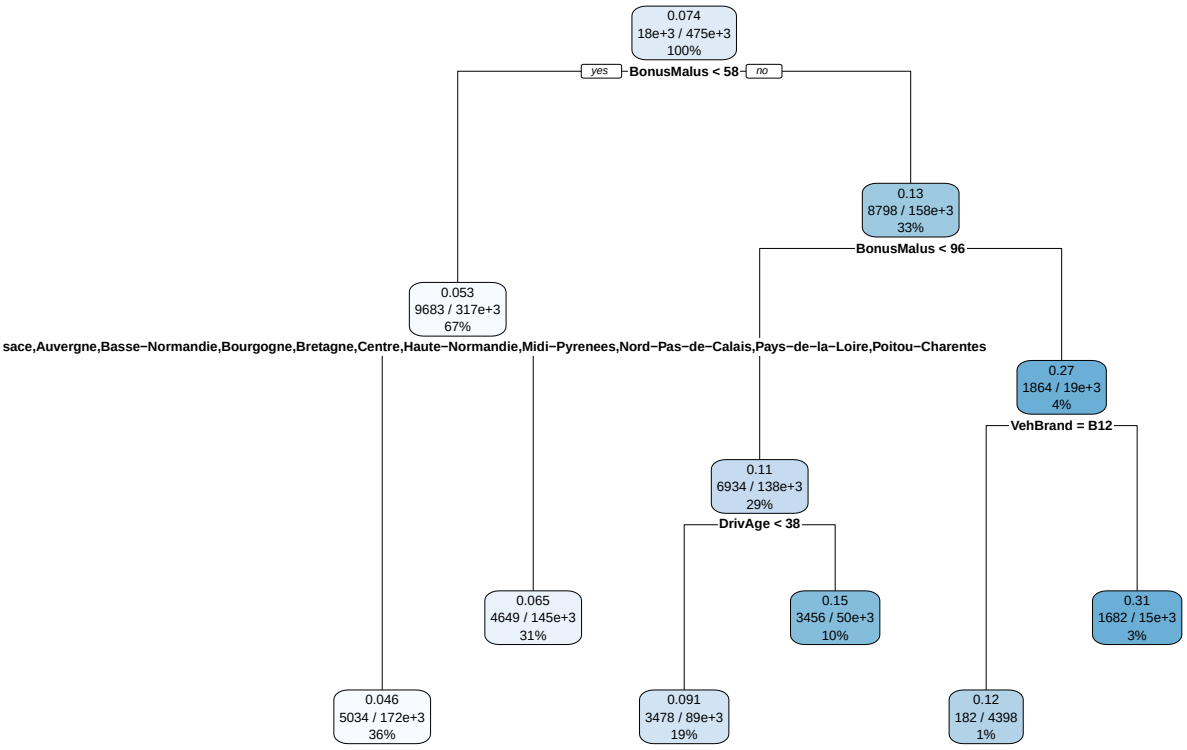
$$MAE = (\Sigma|yi–xi|)/n$$

```
## [1] 0.07293719
```

Our model achieves an MAE of 7.29% which is not too far from 0, meaning the model is quite good at predicting the rate of claims.

5.4 Decision Tree Model

Our second model in this project is a decision tree model. The first node of the the below decision tree shows that overall claim frequency is about 7.4%. The highest claim frequency is of the group that has a Bonus Malus above 96 and car brand is not B12 (31%).



The next step was to calculate predictions and analyze the model's capability to predict on the test set using the below code.

A sample of predictions generated are seen here:

```
##  67294 197989 146557  36874  23059 152205
##  0.046  0.046  0.065  0.091  0.046  0.123
```

We will use Mean Absolute Error to evaluate accuracy of the model.

$$MAE = (\Sigma|yi–xi|)/n$$

```
## [1] 0.1108888
```

Our model achieves an MAE of 11.09% which is a bit high, meaning the model is not too great at predicting the rate of claims

6 RESULTS

The objective was to forecast the frequency of insurance claims on policies with utmost precision. To assess accuracy, we used Mean Absolute Error on the predictions from GLM and Decision Tree models. The models resulted in an error of 7.29% and 11.09% respectively. The GLM is our model of choice as the error is not too far from 0, indicating a strong fit.

7 Limitations

Due to limited computing power, and this data set being quite large, we couldn't try other models such as KNN.

8 References

Dutang, Christophe, and Arthur Charpentier. "CASdatasets: Insurance Datasets." Cas.uqam.ca, Christophe Dutang, 12 Nov. 2020, cas.uqam.ca/. Accessed 10 May 2024.

Floser. "Comparing-Claims-FreMTPL2freq-Sev." Kaggle, www.kaggle.com/code/floser/comparing-claims-fremtpl2freq-sev. Accessed 10 May 2024.

Floser, and Daniel_K. "GLM, Neural Nets and XGBoost for Insurance Pricing." Kaggle, www.kaggle.com/code/floser/glm-neural-nets-and-xgboost-for-insurance-pricing. Accessed 10 May 2024.

Schelldorfer, Jürg and Wuthrich, Mario V., Nesting Classical Actuarial Models into Neural Networks (January 22, 2019). Available at SSRN: https://ssrn.com/abstract=3320525 or http://dx.doi.org/10.2139/ssrn.3320525

Noll, Alexander and Salzmann, Robert and Wuthrich, Mario V., Case Study: French Motor Third-Party Liability Claims (March 4, 2020). Available at SSRN: https://ssrn.com/abstract=3164764 or http://dx.doi.org/10.2139/ssrn.3164764

"57 Poisson Regression | R for Epidemiology." Www.r4epi.com, www.r4epi.com/poisson-regression. Accessed 15 May 2024.

Chapter 4 | the Poisson Distribution. University of Wisconsin, Department of Statistics.

MarinStatsLectures-R Programming & Statistics. "9.10 Poisson Regression in R: Fitting a Model to Rate Data (with Offset) in R." YouTube, 24 Feb. 2021, www.youtube.com/watch?v=QP4F98ysrEA. Accessed 14 May 2024.

Phil Chan. "GLM in R: Poisson Regression Example (Basic Version)." YouTube, 21 Dec. 2012, www.youtube.com/watch?v=VzYSrCLugtY. Accessed 15 May 2024.

Proteus. "Interpreting Effect Sizes in Poisson Regression (or When Using a Log-Link Function)." YouTube, 4 Oct. 2022, www.youtube.com/watch?v=TCVinvY6nRQ. Accessed 17 May 2024.