EDA for Car Insurance Data.

Introduction

In this markdown, we will describe the steps of a classical exploratory analysis on a car insurance dataset. The final goal is to model a Pure Premium of an insurance contract. Modeling techniques using the same dataset are shown in other repositories, as we are focusing here on the preliminary steps. The data in use come from the first chapter of the book "Predictive Modeling Applications in Actuarial Science, Vol.2", Edited by E. Frees et al.. There are 40760 observations and 30 variables and stored at the following address: https://instruction.bus.wisc.edu/jfrees/jfreesbooks/PredictiveModelingVol1/glm/v2-chapter-1.html.

The Pure Premium is by definition the actual future losses per exposure unit. We will see why this notion of exposure is important in the modeling section. For now, let's keep in mind that the pure premium represent the dollars of loss that Insurance companies need to anticipate in order to assess future claims. In a nutshell, it can be defined as the frequency of reporting a claim timed by the average cost of the claim. In this study, we will analyse the distribution of the claims frequency, the average amount of the claim - aka average Severity, as it is called in the industry - and the potential predictors potentially elligible to stand in a model.

Have a good reading!

Data load

```
# Define columnn class for dataset
colCls <- c("integer",          # row id
            "character",        # analysis year
            "numeric",          # exposure
            "character",        # new business / renewal business
            "numeric",          # driver age (continuous)
            "character",        # driver age (categorical)
            "character",        # driver gender
            "character",        # marital status
            "numeric",          # years licensed (continuous)
            "character",        # years licensed (categorical)
            "character",        # ncd level
            "character",        # region
            "character",        # body code
            "numeric",          # vehicle age (continuous)
            "character",        # vehicle age (categorical)
            "numeric",          # vehicle value
            "character",        # seats
            rep("numeric", 6), # ccm, hp, weight, length, width, height (all continuous)
            "character",        # fuel type
            rep("numeric", 3)  # prior claims, claim count, claim incurred (all continuous)
)
```

```r
# Define the data path and filename
data.path <- "C:\\Users\\William.Tiritilli\\Documents\\Project P\\Frees\\Tome 2 - Chapter 1\\"
data.fn <- "sim-modeling-dataset2.csv"

# Read in the data with the appropriate column classes
dta <- read.csv(paste(data.path, data.fn, sep = "/"),
                colClasses = colCls)
str(dta)
```

```
## 'data.frame':    40760 obs. of  27 variables:
##  $ row.id        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ year          : chr  "2010" "2010" "2010" "2010" ...
##  $ exposure      : num  1 1 1 0.08 1 0.08 1 1 0.08 1 ...
##  $ nb.rb         : chr  "RB" "NB" "RB" "RB" ...
##  $ driver.age    : num  63 33 68 68 68 68 53 68 68 65 ...
##  $ drv.age       : chr  "63" "33" "68" "68" ...
##  $ driver.gender : chr  "Male" "Male" "Male" "Male" ...
##  $ marital.status: chr  "Married" "Married" "Married" "Married" ...
##  $ yrs.licensed  : num  5 1 2 2 2 2 5 2 2 2 ...
##  $ yrs.lic       : chr  "5" "1" "2" "2" ...
##  $ ncd.level     : chr  "6" "5" "4" "4" ...
##  $ region        : chr  "3" "38" "33" "33" ...
##  $ body.code     : chr  "A" "B" "C" "C" ...
##  $ vehicle.age   : num  3 3 2 2 1 1 3 1 1 5 ...
##  $ veh.age       : chr  "3" "3" "2" "2" ...
##  $ vehicle.value : num  21.4 17.1 17.3 17.3 25 ...
##  $ seats         : chr  "5" "3" "5" "5" ...
##  $ ccm           : num  1248 2476 1948 1948 1461 ...
##  $ hp            : num  70 94 90 90 85 85 70 85 85 65 ...
##  $ weight        : num  1285 1670 1760 1760 1130 ...
##  $ length        : num  4.32 4.79 4.91 4.91 4.04 ...
##  $ width         : num  1.68 1.74 1.81 1.81 1.67 ...
##  $ height        : num  1.8 1.97 1.75 1.75 1.82 ...
##  $ fuel.type     : chr  "Diesel" "Diesel" "Diesel" "Diesel" ...
##  $ prior.claims  : num  0 0 0 0 0 0 4 0 0 0 ...
##  $ clm.count     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ clm.incurred  : num  0 0 0 0 0 0 0 0 0 0 ...
```

```r
set.seed(54321) # reproducibility
# Create a stratified data partition
train_id <- caret::createDataPartition(
  y = dta$clm.count/dta$exposure,
  p = 0.8,
  groups = 100
)[[1]]
```

```r
# Divide the data in training and test set
dta_trn <- dta[train_id,]
dta_tst <- dta[-train_id,]
```

```r
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Proportions of the number of claims in train data
dta_trn$clm.count %>% table %>% prop.table %>% round(5)
```

```
## .
##       0       1       2       3       4       5
## 0.92257 0.07163 0.00537 0.00037 0.00003 0.00003
```

```
# Proportions of the number of claims in test data
dta_tst$clm.count %>% table %>% prop.table %>% round(5)
```

```
## .
##       0       1       2       3
## 0.92098 0.07252 0.00613 0.00037
```

Proportions in train and test sets are well balanced.

We usually start our work by exploring individual variables to gain a better understanding on the info present in our dataset. We identify our two outcome variables.

- clm.count: number of claims
- clm.incurred: the ultimate cost of those claims

EDA for Frequency

```
# Create a summary table of frequency and severity
# by analysis period
yr.expo <- with(dta, tapply(exposure, year, sum)) #1. select the data, and 2. apply the sum of exposure
yr.clm.count <- with(dta, tapply(clm.count, year, sum)) # count the claims accross the year
yr.clm.incr <- with(dta, tapply(clm.incurred, year, sum))

yr.summary <- cbind(
  exposure = round(yr.expo,1),
  clm.count = yr.clm.count,
  clm.incurred = round(yr.clm.incr,0),
  frequency = round(yr.clm.count / yr.expo, 3),
  severity = round(yr.clm.incr / yr.clm.count, 1))

yr.summary <- rbind(yr.summary,
                    total = c(
                      round(sum(yr.expo),1),
```

```
                      sum(yr.clm.count),
                      round(sum(yr.clm.incr),0),
                      round(sum(yr.clm.count)/sum(yr.expo),3),
                      round(sum(yr.clm.incr)/sum(yr.clm.count),1)))
print(yr.summary)
```

```
##       exposure clm.count clm.incurred frequency severity
## 2010    3662.4       422       287869     0.115    682.2
## 2011    5221.3       551       314431     0.106    570.7
## 2012    6526.0      1278      1021152     0.196    799.0
## 2013    5385.1      1180      1087735     0.219    921.8
## total  20794.8      3431      2711187     0.165    790.2
```

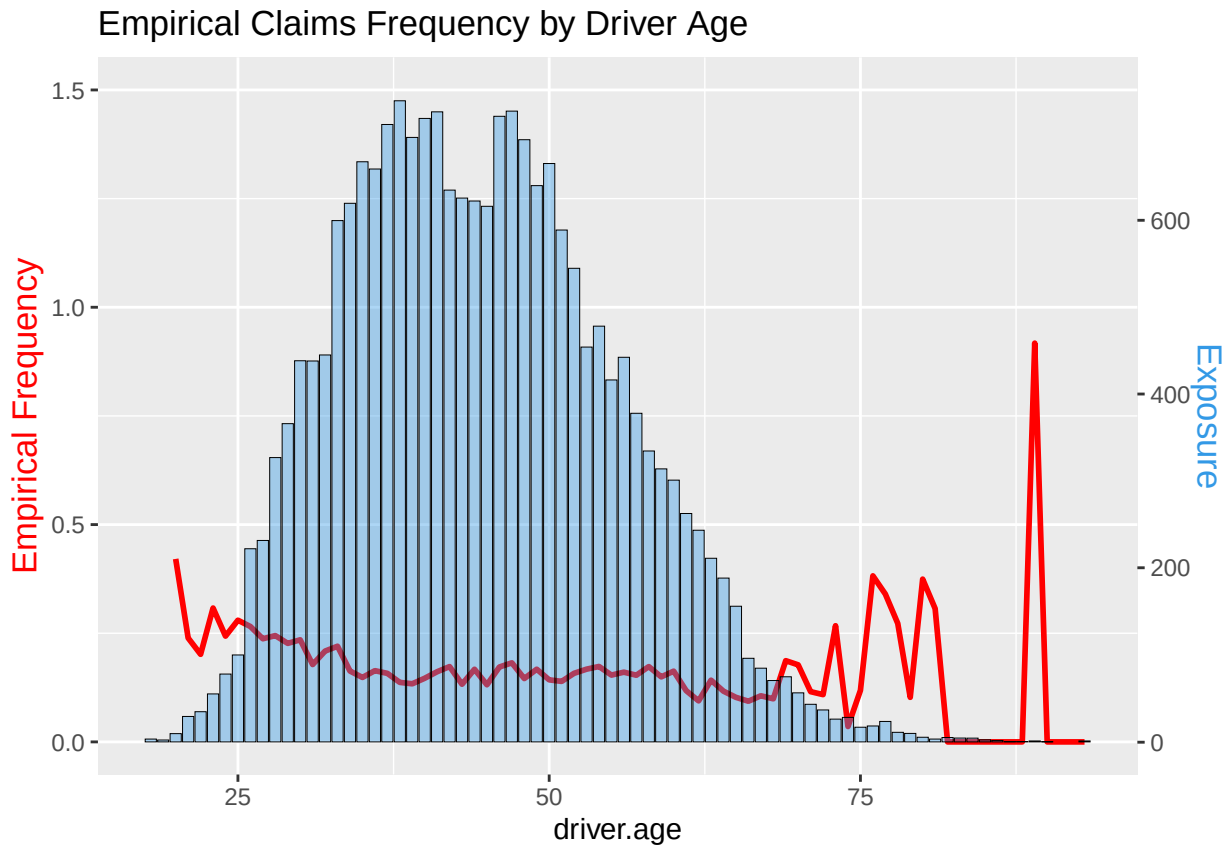
We want to show the Evolution of the Empirical frequency and Exposure for all three years of the training data by driver age.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
library(dplyr)

# Creation of the data frame
graph_data <- dta %>%  group_by(driver.age) %>%  summarise(Sum_Expo = sum(exposure),
                                                  Number_of_Claims = sum(clm.count),
                                                  Emp_freq = sum(clm.count)/sum(exposure))

# Bar plot overlapping with bar chart

# A few constants
freqColor <- "red"
expoColor <- rgb(0.2, 0.6, 0.9, 1)


# For the different scales,
# Set the following two values to values close to the limits of the data
# you can play around with these to adjust the positions of the graphs;
# the axes will still be correct)
ylim.prim <- c(0, 1)      # for claim frequency
ylim.sec <- c(0, 500)     # for Exposure --> need to go way above the max to let
                          # the data appearing in the chart
# For explanation:
# https://stackoverflow.com/questions/32505298/explain-ggplot2-warning-removed-k-rows-containing-missin

# The following makes the necessary calculations based on these limits,
# and makes the plot itself:
b <- diff(ylim.prim)/diff(ylim.sec)
a <- ylim.prim[1] - b*ylim.sec[1]

# Building the graph
graph_freq <- ggplot(graph_data, aes(x=driver.age, Emp_freq)) +
```

```
  geom_line( aes(y=Emp_freq), size=1, color=freqColor) +

    geom_bar( aes(y=a+Sum_Expo*b), stat="identity", size=.1, fill=expoColor, color="black", alpha=.4) +

  scale_y_continuous(

    # Features of the first axis
    name = "Empirical Frequency", limits = c(0, 1.5),

    # Add a second axis and specify its features
    sec.axis = sec_axis(~ (. - a)/b, name = "Exposure")
  ) +

  #theme_ipsum() +
  theme(
    axis.title.y = element_text(color = freqColor, size=13),
    axis.title.y.right = element_text(color = expoColor, size=13)
  ) +

  ggtitle("Empirical Claims Frequency by Driver Age")

# Print the whole graph
graph_freq
```



The frequency decreases over years and become more volatile after 75 years old.

We want to see the pattern for each calendar year. A minor update of the previous code is required.

```r
# Creation of the data frame
graph_data2 <- dta %>%  group_by(driver.age, year) %>%  summarise(Sum_Expo = sum(exposure),
                                                    Number_of_Claims = sum(clm.count),
                                                    Emp_freq = sum(clm.count)/sum(exposure))
```

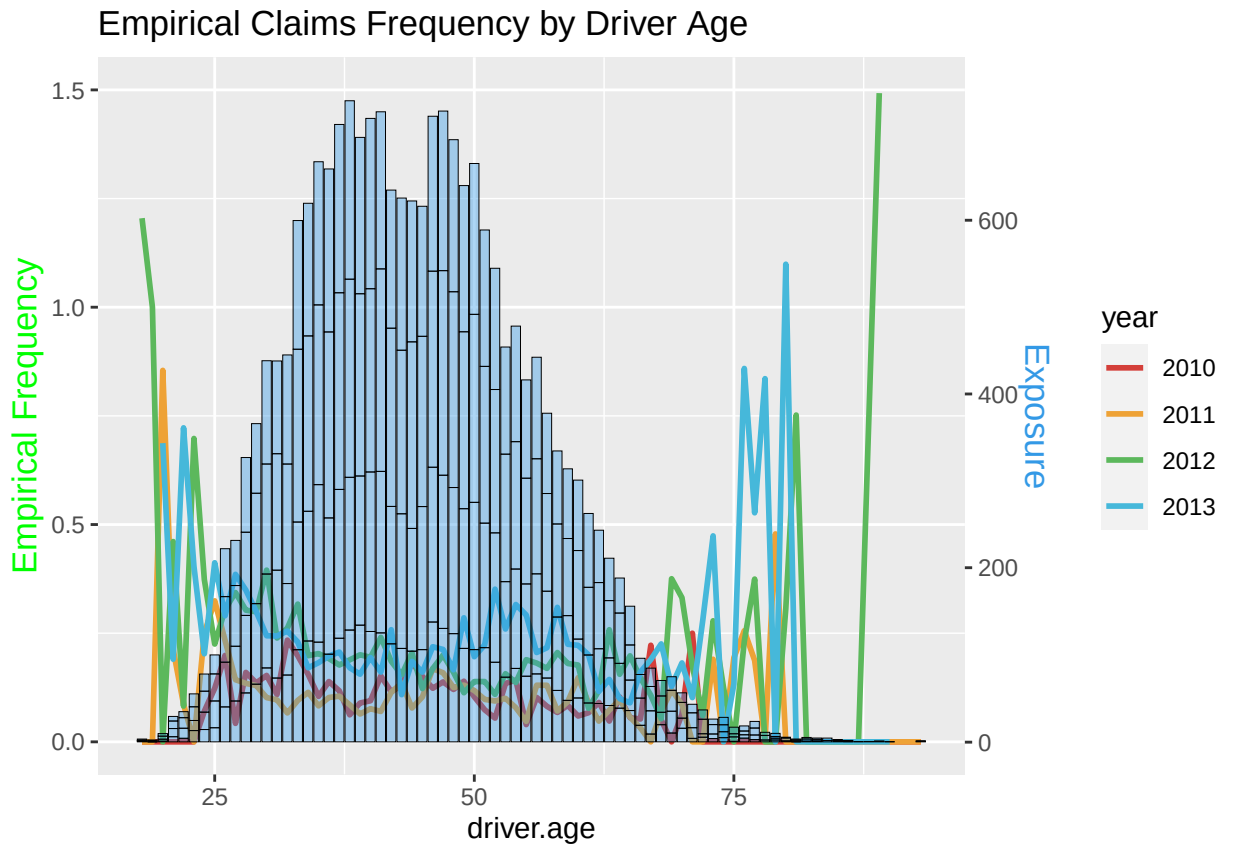
```
## `summarise()` has grouped output by 'driver.age'. You can override using the `.groups` argument.
```

```r
# Sort the dataframe by year
# https://dplyr.tidyverse.org/reference/arrange.html
graph_data2 <- arrange(graph_data2, year)
head(graph_data2)
```

```
## # A tibble: 6 x 5
## # Groups:   driver.age [6]
##   driver.age year  Sum_Expo Number_of_Claims Emp_freq
##        <dbl> <chr>    <dbl>            <dbl>    <dbl>
## 1         18 2010      1                   0   0
## 2         20 2010      0.75                0   0
## 3         21 2010      5.91                0   0
## 4         22 2010      4.08                0   0
## 5         23 2010     13.0                 0   0
## 6         24 2010     14.2                 1   0.0705
```

```r
# A few constants
freqColor <- c("#D43F3A", "#EEA236", "#5CB85C", "#46B8DA", "#9632B8")
expoColor <- rgb(0.2, 0.6, 0.9, 1)


# For the different scales,
# Set the following two values to values close to the limits of the data
# you can play around with these to adjust the positions of the graphs;
# the axes will still be correct)
ylim.prim <- c(0, 1)      # for claim frequency
ylim.sec <- c(0, 500)    # for Exposure --> need to go way above the max to let
                          # the data appearing in the chart
# For explanation:
# https://stackoverflow.com/questions/32505298/explain-ggplot2-warning-removed-k-rows-containing-missin

# The following makes the necessary calculations based on these limits,
# and makes the plot itself:
b <- diff(ylim.prim)/diff(ylim.sec)
a <- ylim.prim[1] - b*ylim.sec[1]

# Building the graph
graph_freq <- ggplot(graph_data2, aes(x=driver.age, year, y = Emp_freq, color=year)) +

  geom_line( aes(y=Emp_freq), size=1) +

  scale_color_manual(values = freqColor) +

    geom_bar( aes(y=a+Sum_Expo*b), stat="identity", size=.1, fill=expoColor, color="black", alpha=.4) +
```

```
  scale_y_continuous(

    # Features of the first axis
    name = "Empirical Frequency", limits = c(0, 1.5),


    # Add a second axis and specify its features
    sec.axis = sec_axis(~ (. - a)/b, name = "Exposure")
  ) +

  #theme_ipsum() +
  theme(
    axis.title.y = element_text(color = "green", size=13),
    axis.title.y.right = element_text(color = expoColor, size=13)
  ) +

  ggtitle("Empirical Claims Frequency by Driver Age")

# Print the whole graph
graph_freq
```



We observe that 2013 and 2012 are very volatile for the younger age and the seniors. Moreover, the individual calendar year frequency are more volatile than all the 4 years combined.

1. Size of the engine Let's investigate some other variables, like the size of the engine (ccm). It is a

continuous variable, so we can split it by ranges using the function 'cut'. The package dplyr will help to summarize the information.

```
# Size of Engine
# Creation of a categorical

# Check the quantile
ccm_quantile <- quantile(dta$ccm)
print(ccm_quantile)
```

```
##   0%  25%  50%  75% 100%
##  970 1398 1560 1896 3198
```

```
dta$ccm_range <- cut(dta$ccm, breaks = c(ccm_quantile[1],
                                                    ccm_quantile[2],
                                                    ccm_quantile[3],
                                                    ccm_quantile[4],
                                         ccm_quantile[5]),
                     labels = c("970-1398","1399-1560",
                                "1561-1896", "1897-3198"),include.lowest = TRUE)

# Use of the pipe to pivot the data
dta %>% group_by(ccm_range) %>% summarise(Sum_Expo = sum(exposure),
                                          Number_of_Claims = sum(clm.count),
                                          Emp_freq = sum(clm.count)/sum(exposure))
```

```
## # A tibble: 4 x 4
##   ccm_range Sum_Expo Number_of_Claims Emp_freq
##   <fct>        <dbl>            <dbl>    <dbl>
## 1 970-1398     6342.             1147    0.181
## 2 1399-1560    4872.              842    0.173
## 3 1561-1896    5665.              859    0.152
## 4 1897-3198    3916.              583    0.149
```

```
# Example with another granularity
bk <- unique(quantile(dta$ccm, probs = seq(0, 1, by = 0.05)))
dta$ccm.d <- cut(dta$ccm, breaks = bk, include.lowest = TRUE)

dta %>% group_by(ccm.d) %>% summarise(Sum_Expo = sum(exposure),
                                          Number_of_Claims = sum(clm.count),
                                          Emp_freq = sum(clm.count)/sum(exposure))
```

```
## # A tibble: 12 x 4
##    ccm.d       Sum_Expo Number_of_Claims Emp_freq
##    <fct>          <dbl>            <dbl>    <dbl>
##  1 [970,1248]     4350.              759    0.174
##  2 (1248,1398]    1992.              388    0.195
##  3 (1398,1461]    3240.              540    0.167
##  4 (1461,1560]    1632.              302    0.185
##  5 (1560,1598]     590.               98    0.166
##  6 (1598,1753]    2302.              347    0.151
##  7 (1753,1868]     982.              156    0.159
```

```
##  8 (1868,1896]    1790.             258    0.144
##  9 (1896,1997]    1063.             162    0.152
## 10 (1997,2476]    1440.             196    0.136
## 11 (2476,2477]     557.              79    0.142
## 12 (2477,3198]     856.             146    0.171
```

2. Gender Even if the practice of including the driver gender is not allowed in every country, it might be an interesting predictor to analyze the frequency of accidents.

```
# Investigate the frequency of claims by the variables driver.gender and marital
library(tidyr)

# https://www.youtube.com/watch?v=AkaiM-Mm_Ag
dta %>% group_by(driver.gender, year) %>%
  summarise(emp_freq = round(sum(clm.count)/sum(exposure),3)*100) %>%
  spread(driver.gender, emp_freq)
```

```
## `summarise()` has grouped output by 'driver.gender'. You can override using the `.groups` argument.
```

```
## # A tibble: 4 x 3
##   year  Female  Male
##   <chr>  <dbl> <dbl>
## 1 2010    13.9  11.2
## 2 2011    11.5  10.4
## 3 2012    25.9  18.8
## 4 2013    24.8  21.5
```

```
dta %>% group_by(marital.status, year) %>%
  summarise(emp_freq = sum(clm.count)/sum(exposure)) %>%
  spread(marital.status, emp_freq)
```

```
## `summarise()` has grouped output by 'marital.status'. You can override using the `.groups` argument.
```

```
## # A tibble: 4 x 5
##   year  Divorced Married Single Widow
##   <chr>    <dbl>   <dbl>  <dbl> <dbl>
## 1 2010     0.117   0.115 0.0993 0.144
## 2 2011     0.114   0.106 0.0781 0.162
## 3 2012     0.188   0.190 0.254  0.475
## 4 2013     0.246   0.215 0.299  0.272
```

Totals needed here.

This line gives a quick view of the proportion between gender

```
with (dta , table ( driver.gender, clm.count) )
```

```
##              clm.count
## driver.gender     0     1     2     3     4     5
##        Female  4110   365    39     4     0     1
##        Male   33481  2562   186    11     1     0
```

Over the dataset, male drivers have a frequency equal to 15.8%, and females have had a frequency equal to 18.4%. This suggests that gender is a variable that could help segment our policyholders.

Is the difference significant? We will randomly assign the label "married" to 22761 observations. Then, we compute the frequency of each group and take the difference.

```
# Creation of a train set
smp_size <- floor(0.7 * nrow(dta))
# set the seed to make your partition reproductible
set.seed(1234)
train_ind <- sample(seq_len(nrow(dta)), size = smp_size)
train<-dta[train_ind, ]
test<-dta[-train_ind, ]
```

```
set.seed(1029384756)

# We want to run the experiement 10000 times
N <- 10000

tmp <- subset(train, marital.status %in% c("Married", "Widow"), select =c("marital.status",
"exposure", "clm.count") )

# Create a dataframe tagging each label with TRUE or FALSE
f <- tmp$marital.status == "Married"

# Create an empty dataframe of size N
d <- numeric(N)

for(i in 1:N) {

# fct sample takes a sample of the specified size from the elements of x
g <- sample(f, length(f))

#
married.fq <- sum(tmp$clm.count[g]) / sum(tmp$exposure[g])
widow.fq <- sum(tmp$clm.count[!g]) / sum(tmp$exposure[!g])


# Compute the difference in frequencies between married and widow
# and store in a data frame of size N
d[i] <- married.fq - widow.fq

}
```

Results

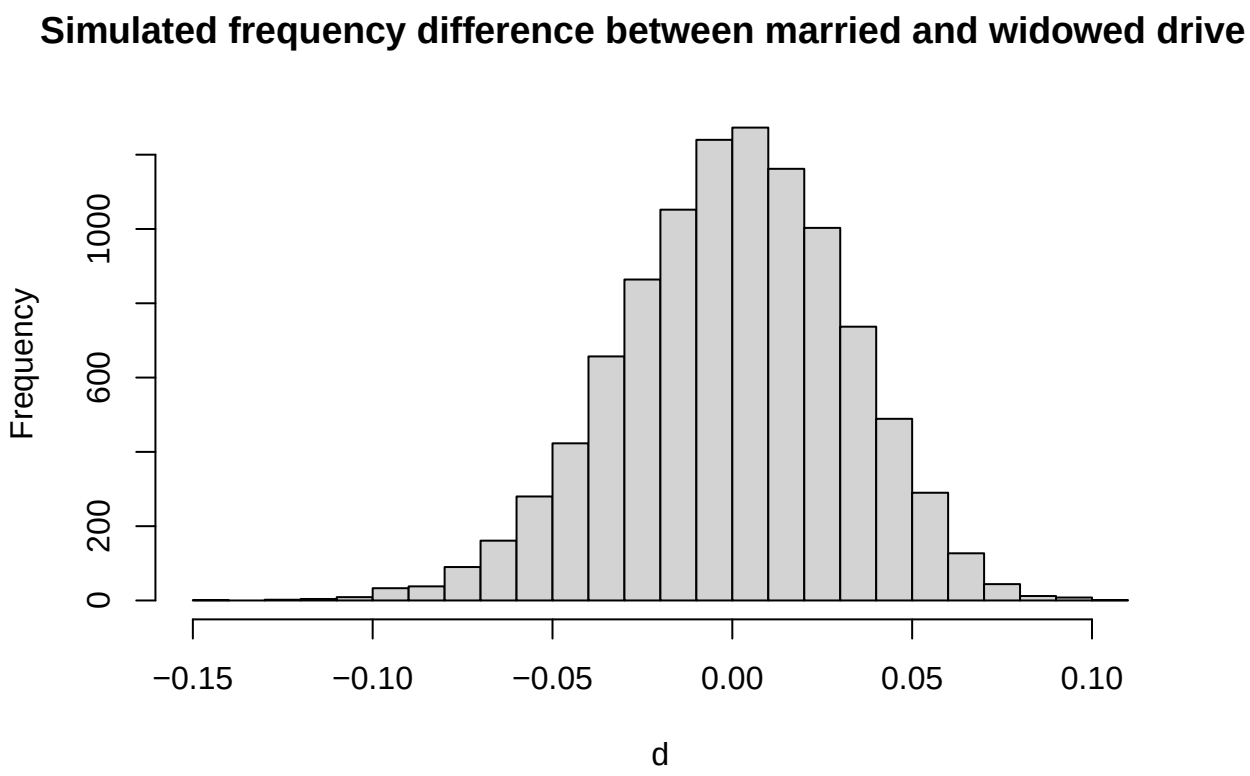
```
quantile(d, c(0.025, 0.05, 0.1, 0.25, 0.5,
0.75, 0.9, 0.95, 0.975))
```

```
##         2.5%           5%          10%          25%          50%          75%
## -0.064571057 -0.053498652 -0.040967588 -0.020689849  0.001010976  0.021646309
##          90%          95%        97.5%
##  0.039423875  0.049595680  0.057364579
```

The confidence interval -0.041 and 0.039. The actual difference we observed is -0.115 and is clearly outside this interval. Therefore, this difference is statistically significant.

We can verify this with a graph:

```
hist(d, breaks = c(20), main = paste("Simulated frequency difference between married and widowed drivers
```



The actual difference between these groups is well outside the bulk of the distribution.

3. Age of the driver

Variable age is crucial for pricing. For a GLM regression, it is easier to bin the age variable into classes. In practice, an insurance product is designed in collaboration between all the player of the company: Actuaries, marketing, sales... Each department plays its partition, with sometime divergence of interest. While Marketing and Sales aim to sell at a competitive price, Actuaries alert on the risks of under reserving and potential future losses. Sometimes, push back simply come from the IT department because the pricing grid by band would be too complex to implement in production. As Golburg et al. says in the CAS Mongraph "Generalized linear model for insurance rating", "choosing between two final models is very often a business decision".

```
dta$age.bins <- cut(dta$driver.age, c(0, 34, 64, 110))

dta %>% group_by(age.bins, driver.gender) %>%
  summarise(emp_freq = sum(clm.count)/sum(exposure)) %>%
  spread(age.bins, emp_freq)
```

```
## `summarise()` has grouped output by 'age.bins'. You can override using the `.groups` argument.
```

```
## # A tibble: 2 x 4
##   driver.gender `(0,34]` `(34,64]` `(64,110]`
##   <chr>            <dbl>     <dbl>      <dbl>
## 1 Female           0.234     0.190      0.193
## 2 Male             0.216     0.149      0.132
```

Number of records by number of claims for the entire dataset. Statistics for new and renewal business.

```
table(dta$clm.count)
```

```
##
##     0     1     2     3     4     5
## 37591  2927   225    15     1     1
```

```
dta %>% group_by(nb.rb) %>%
  summarise(clm_inc = sum(clm.incurred),
            clm.cnt = sum(clm.count),
            severity = clm_inc/clm.cnt) %>% as.data.frame()
```

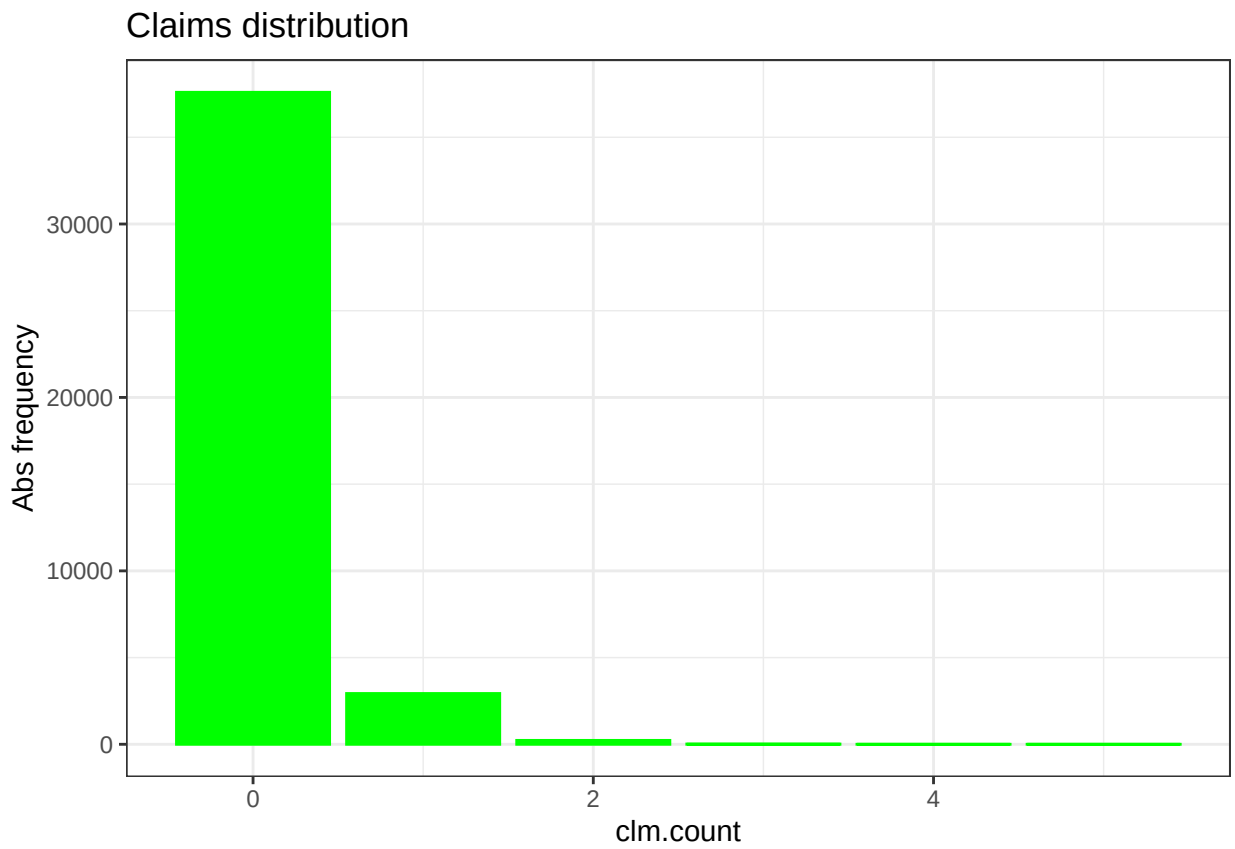
```
##   nb.rb   clm_inc clm.cnt severity
## 1    NB 2160463.5    2633 820.5330
## 2    RB  550723.1     798 690.1291
```

Frequency Distribution

To apply a GLM, we need to make two choices: link function and response distribution. For most insurance pricing, we would like to have a multiplicative rating plan so we will be using a logarithm link function. Concerning the distribution, we are modeling a claim count, so the natural candidate are Poisson or Negative Binomial.

Claims distribution

```
KULBg = "green"
# same graph with the weight of the expo
g1 <- ggplot(dta, aes(clm.count)) + theme_bw()+
  geom_bar( col = KULBg, fill = KULBg) +
  labs(y="Abs frequency")+
  ggtitle("Claims distribution")

print(g1)
```



Let's check if the assumption of Poisson having mean=variance are respected.

```
f <- with(dta, clm.count / exposure) # frequency for each record
w <- with(dta, exposure) # weight for each record
mean.f <- sum(f * w) / sum(w) # mean frequency
second.f <- sum(f**2 * w) / sum(w) # second moment
var.f <- second.f - (mean.f)**2
print(var.f)
```

```
## [1] 0.3391072
```

```
print(mean.f)
```

```
## [1] 0.1649933
```

We can see that mean and variance are not equal. In this case the variance is large than the mean, and we have an overdispersed dataset.