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

```r
# Define colummn 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\\Book - Predictive Modeling vol1&2 - F
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 ...
```

We import a the data as a pandas dataframe.

```python
# Library
import pandas as pd

# Import file
data = pd.read_csv("C:\\Users\\William.Tiritilli\\Documents\\Project P\\Book - Predictive Modeling vol1&
```

```
## sys:1: DtypeWarning: Columns (9) have mixed types.Specify dtype option on import or set low_memory=Fa
```

```python
data.info()
```

```
## <class 'pandas.core.frame.DataFrame'>
## RangeIndex: 40760 entries, 0 to 40759
```

```
## Data columns (total 27 columns):
##  #   Column          Non-Null Count  Dtype
## ---  ------          --------------  -----
##  0   row.id          40760 non-null  int64
##  1   year            40760 non-null  int64
##  2   exposure        40760 non-null  float64
##  3   nb.rb           40760 non-null  object
##  4   driver.age      40760 non-null  int64
##  5   drv.age         40760 non-null  int64
##  6   driver.gender   40760 non-null  object
##  7   marital.status  40760 non-null  object
##  8   yrs.licensed    40760 non-null  int64
##  9   yrs.lic         40760 non-null  object
##  10  ncd.level       40760 non-null  int64
##  11  region          40760 non-null  int64
##  12  body.code       40760 non-null  object
##  13  vehicle.age     40760 non-null  int64
##  14  veh.age         40760 non-null  int64
##  15  vehicle.value   40760 non-null  float64
##  16  seats           40760 non-null  int64
##  17  ccm             40760 non-null  int64
##  18  hp              40760 non-null  int64
##  19  weight          40760 non-null  int64
##  20  length          40760 non-null  float64
##  21  width           40760 non-null  float64
##  22  height          40760 non-null  float64
##  23  fuel.type       40760 non-null  object
##  24  prior.claims    40760 non-null  int64
##  25  clm.count       40760 non-null  int64
##  26  clm.incurred    40760 non-null  float64
## dtypes: float64(6), int64(15), object(6)
## memory usage: 8.4+ MB
```

Dimension of a data frame

```
dim(dta)
```

```
## [1] 40760    27
```

```
data.shape
```

```
## (40760, 27)
```

Brief look at the data

```
head(dta)
```

```
##   row.id year exposure nb.rb driver.age drv.age driver.gender marital.status
## 1      1 2010     1.00    RB         63      63          Male        Married
## 2      2 2010     1.00    NB         33      33          Male        Married
## 3      3 2010     1.00    RB         68      68          Male        Married
## 4      4 2010     0.08    RB         68      68          Male        Married
## 5      5 2010     1.00    RB         68      68          Male        Married
## 6      6 2010     0.08    RB         68      68          Male        Married
##   yrs.licensed yrs.lic ncd.level region body.code vehicle.age veh.age
## 1            5       5         6      3         A           3       3
## 2            1       1         5     38         B           3       3
## 3            2       2         4     33         C           2       2
## 4            2       2         4     33         C           2       2
## 5            2       2         3      3         D           1       1
## 6            2       2         3      3         D           1       1
##   vehicle.value seats  ccm hp weight length width height fuel.type prior.claims
## 1         21.45     5 1248 70   1285  4.322 1.684  1.801    Diesel            0
## 2         17.05     3 2476 94   1670  4.795 1.740  1.965    Diesel            0
## 3         17.30     5 1948 90   1760  4.910 1.810  1.755    Diesel            0
## 4         17.30     5 1948 90   1760  4.910 1.810  1.755    Diesel            0
## 5         25.00     2 1461 85   1130  4.035 1.672  1.825    Diesel            0
## 6         25.00     2 1461 85   1130  4.035 1.672  1.825    Diesel            0
##   clm.count clm.incurred
## 1         0            0
## 2         0            0
## 3         0            0
## 4         0            0
## 5         0            0
## 6         0            0
```

```
data.head(10)
```

```
##    row.id  year  exposure  ... prior.claims  clm.count  clm.incurred
## 0       1  2010      1.00  ...            0          0           0.0
## 1       2  2010      1.00  ...            0          0           0.0
## 2       3  2010      1.00  ...            0          0           0.0
## 3       4  2010      0.08  ...            0          0           0.0
## 4       5  2010      1.00  ...            0          0           0.0
## 5       6  2010      0.08  ...            0          0           0.0
## 6       7  2011      1.00  ...            4          0           0.0
## 7       8  2010      1.00  ...            0          0           0.0
## 8       9  2011      0.08  ...            0          0           0.0
## 9      10  2010      1.00  ...            0          0           0.0
##
## [10 rows x 27 columns]
```

Check Na's

```
table(is.na(dta))
```

```
##
```

```
##   FALSE
## 1100520
```

```
data.isnull().any()
```

```
## row.id            False
## year              False
## exposure          False
## nb.rb             False
## driver.age        False
## drv.age           False
## driver.gender     False
## marital.status    False
## yrs.licensed      False
## yrs.lic           False
## ncd.level         False
## region            False
## body.code         False
## vehicle.age       False
## veh.age           False
## vehicle.value     False
## seats             False
## ccm               False
## hp                False
## weight            False
## length            False
## width             False
## height            False
## fuel.type         False
## prior.claims      False
## clm.count         False
## clm.incurred      False
## dtype: bool
```

Statistics overview

```
summary(dta)
```

```
##      row.id          year              exposure         nb.rb
##  Min.   :    1   Length:40760       Min.   :0.0800   Length:40760
##  1st Qu.:10191   Class :character   1st Qu.:0.2500   Class :character
##  Median :20381   Mode  :character   Median :0.5000   Mode  :character
##  Mean   :20381                      Mean   :0.5102
##  3rd Qu.:30570                      3rd Qu.:0.7500
##  Max.   :40760                      Max.   :1.0000
##    driver.age      drv.age          driver.gender      marital.status
##  Min.   :18.00   Length:40760       Length:40760       Length:40760
##  1st Qu.:36.00   Class :character   Class :character   Class :character
##  Median :44.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :44.55
```

```
##  3rd Qu.:52.00
##  Max.   :93.00
##   yrs.licensed      yrs.lic           ncd.level           region
##  Min.   : 1.000   Length:40760       Length:40760       Length:40760
##  1st Qu.: 2.000   Class :character   Class :character   Class :character
##  Median : 3.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 3.207
##  3rd Qu.: 4.000
##  Max.   :10.000
##   body.code          vehicle.age       veh.age            vehicle.value
##  Length:40760       Min.   : 0.000   Length:40760       Min.   :  4.50
##  Class :character   1st Qu.: 1.000   Class :character   1st Qu.: 17.00
##  Mode  :character   Median : 3.000   Mode  :character   Median : 22.10
##                     Mean   : 3.256                      Mean   : 23.50
##                     3rd Qu.: 5.000                      3rd Qu.: 28.72
##                     Max.   :18.000                      Max.   :132.60
##     seats                ccm             hp             weight
##  Length:40760       Min.   : 970   Min.   : 42.00   Min.   : 860
##  Class :character   1st Qu.:1398   1st Qu.: 70.00   1st Qu.:1190
##  Mode  :character   Median :1560   Median : 75.00   Median :1320
##                     Mean   :1671   Mean   : 86.38   Mean   :1364
##                     3rd Qu.:1896   3rd Qu.:100.00   3rd Qu.:1475
##                     Max.   :3198   Max.   :200.00   Max.   :2275
##      length          width           height       fuel.type
##  Min.   :1.805   Min.   :1.475   Min.   :1.420   Length:40760
##  1st Qu.:4.035   1st Qu.:1.716   1st Qu.:1.780   Class :character
##  Median :4.278   Median :1.742   Median :1.825   Mode  :character
##  Mean   :4.321   Mean   :1.778   Mean   :1.814
##  3rd Qu.:4.405   3rd Qu.:1.816   3rd Qu.:1.840
##  Max.   :6.945   Max.   :2.119   Max.   :2.524
##   prior.claims       clm.count         clm.incurred
##  Min.   : 0.0000   Min.   :0.00000   Min.   :    0.00
##  1st Qu.: 0.0000   1st Qu.:0.00000   1st Qu.:    0.00
##  Median : 0.0000   Median :0.00000   Median :    0.00
##  Mean   : 0.8313   Mean   :0.08418   Mean   :   66.52
##  3rd Qu.: 1.0000   3rd Qu.:0.00000   3rd Qu.:    0.00
##  Max.   :21.0000   Max.   :5.00000   Max.   :11683.58
```

```
data.describe().transpose
```

```
## <bound method DataFrame.transpose of              row.id          year  ...     clm.count  clm.incurr
## count  40760.000000  40760.000000  ...  40760.000000  40760.000000
## mean   20380.500000   2011.661580  ...      0.084176     66.515863
## std    11766.542823      1.046689  ...      0.301870    406.225496
## min        1.000000   2010.000000  ...      0.000000      0.000000
## 25%    10190.750000   2011.000000  ...      0.000000      0.000000
## 50%    20380.500000   2012.000000  ...      0.000000      0.000000
## 75%    30570.250000   2013.000000  ...      0.000000      0.000000
## max    40760.000000   2013.000000  ...      5.000000  11683.580000
##
## [8 rows x 21 columns]>
```

Univariate Analysis

Empirical Claim frequency

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## 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
```

```
dta %>% summarise(emp_freq = sum(clm.count)/sum(exposure))
```

```
##    emp_freq
## 1 0.1649933
```

```
emp_freq = sum(data['clm.count'])/sum(data['exposure'])
print(emp_freq)
```

```
## 0.16499325071327606
```

By gender

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

```
## # A tibble: 2 x 2
##   driver.gender emp_freq
##   <chr>            <dbl>
## 1 Female           0.200
## 2 Male             0.161
```

```
import pandas as pd
result = data.groupby('driver.gender').apply(lambda x: (x['clm.count'].sum() / x['exposure'].sum())).res
print(result)
```

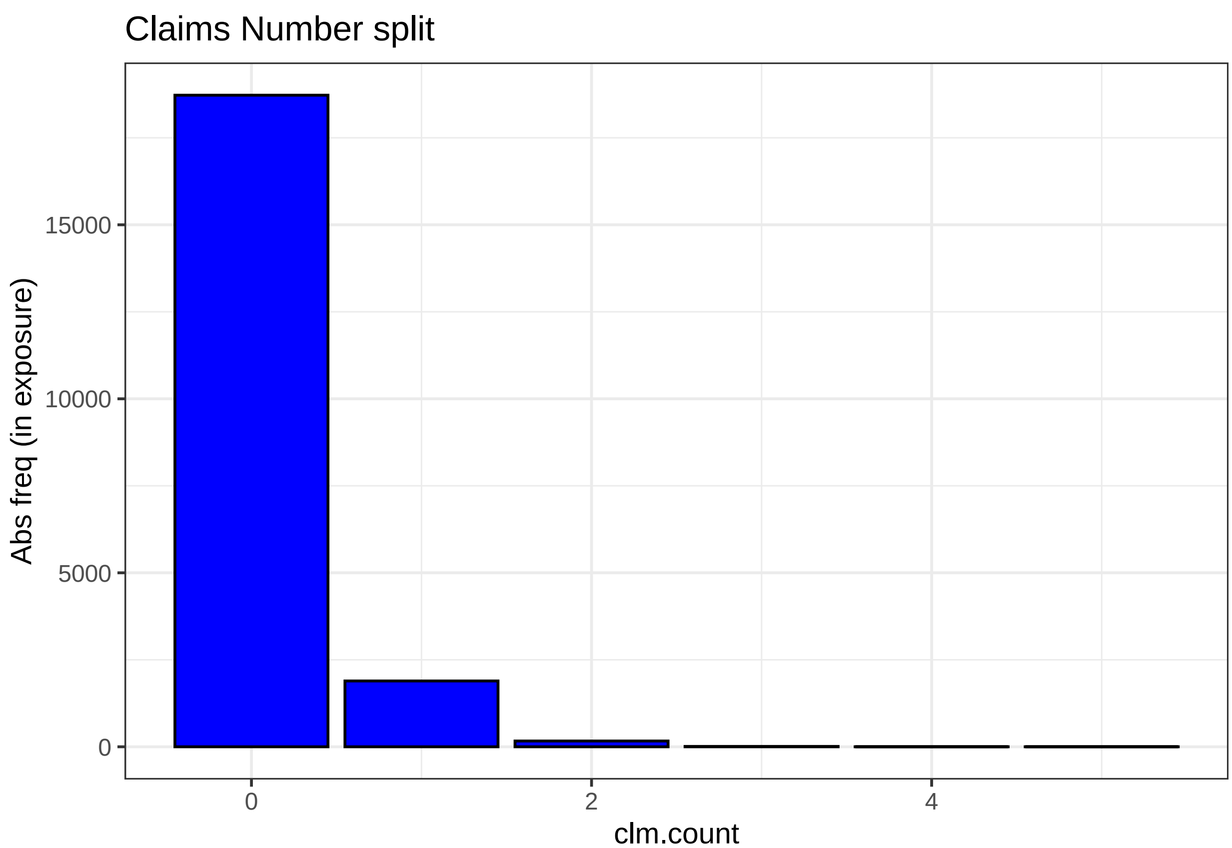
```
##   driver.gender  emp_freq
## 0        Female  0.199714
## 1          Male  0.160668
```

Plot

```
couleur <- "blue"
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

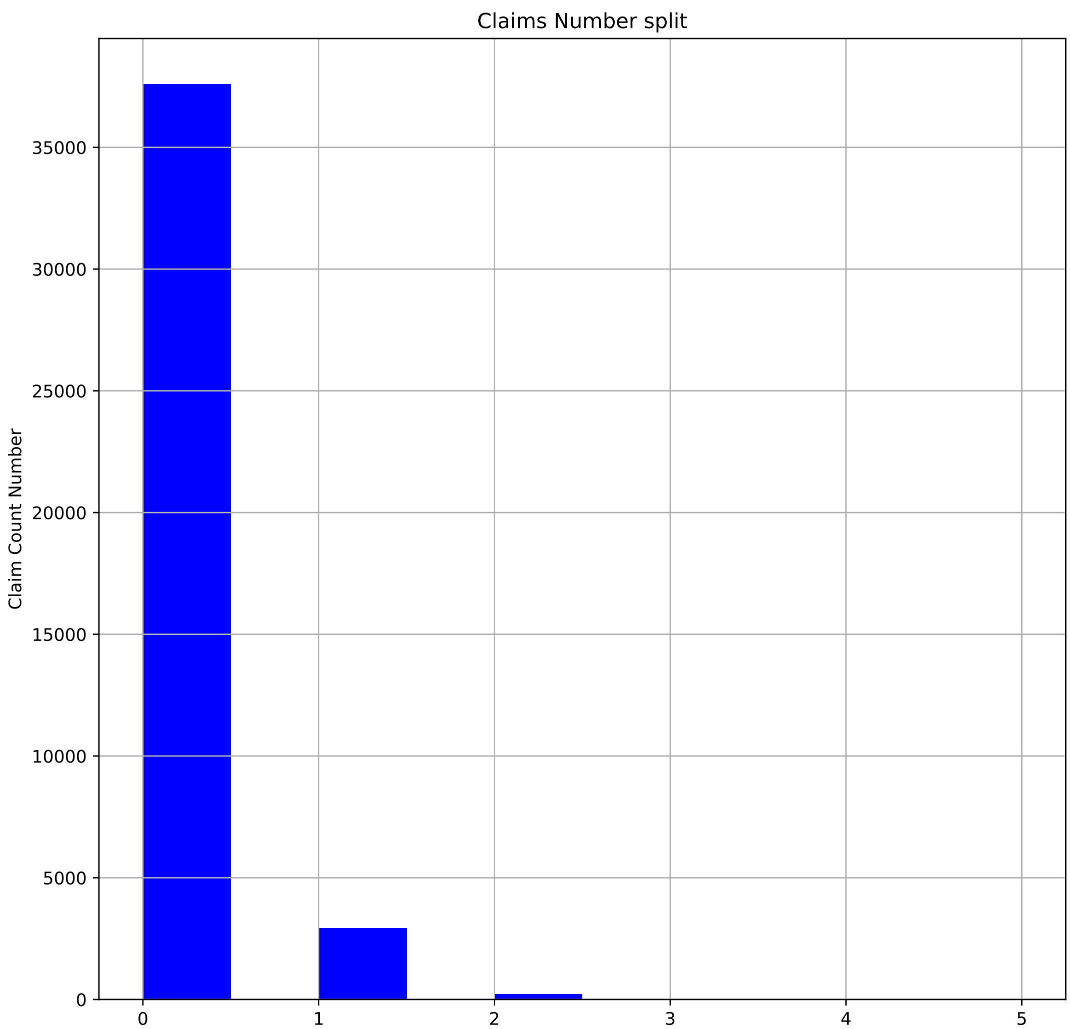
```
g <- ggplot(dta, aes(clm.count)) + theme_bw() +
  geom_bar(aes(weight = exposure), col = "black",
           fill = couleur) +
  labs(y = "Abs freq (in exposure)") +
  ggtitle("Claims Number split")
g
```




```
import numpy as np
import matplotlib.pyplot as plt

data[['clm.count']].hist(bins=10, figsize = (10,10), color = 'blue')
```

```
## array([[<AxesSubplot:title={'center':'clm.count'}>]], dtype=object)
```

```
plt.ylabel('Count')
plt.ylabel('Claim Count Number')
plt.title('Claims Number split')

plt.show()
```



Claim Severity

```
dta %>% filter(clm.count != 0) %>% summarize(avg_severity = mean(clm.incurred))
```

```
##   avg_severity
```

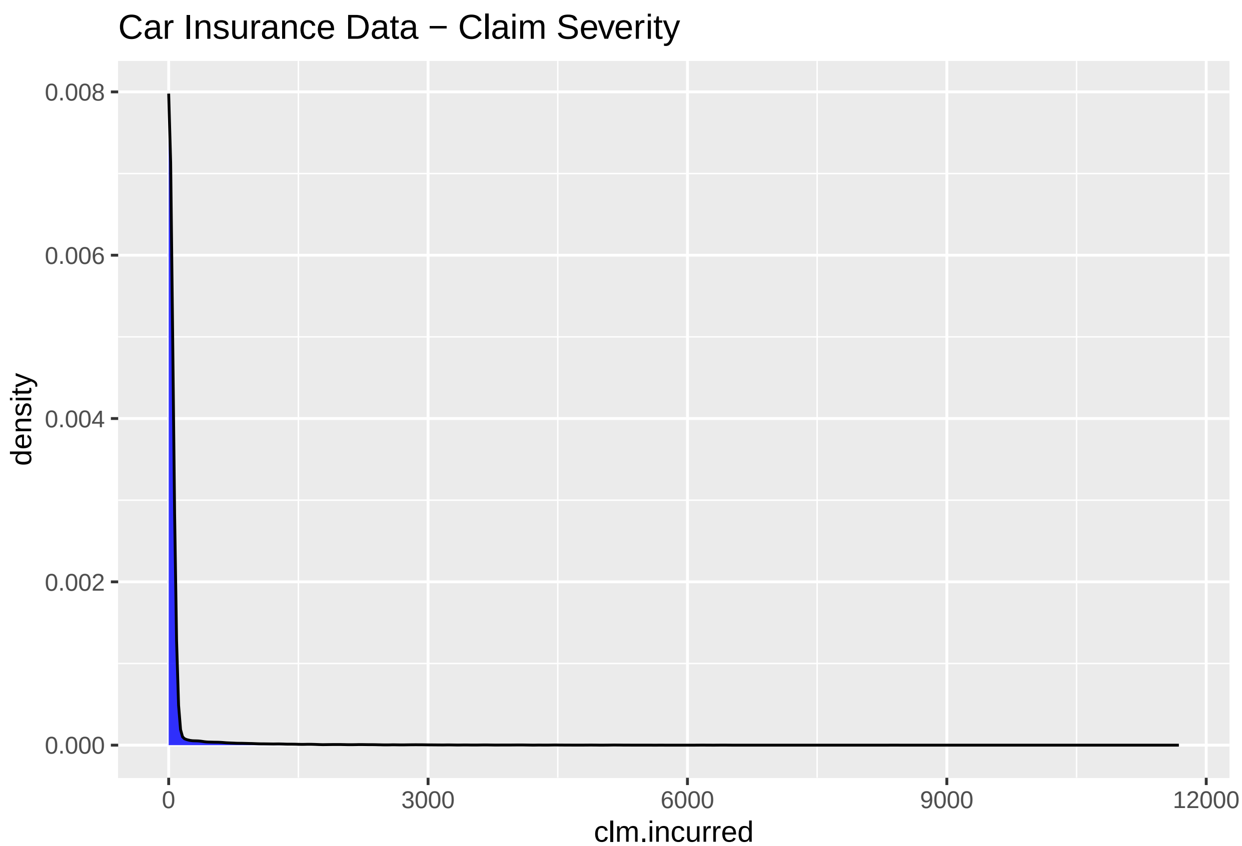
```
## 1     855.5338
```

```
data.loc[data['clm.incurred'] != 0, 'clm.incurred'].mean()
```

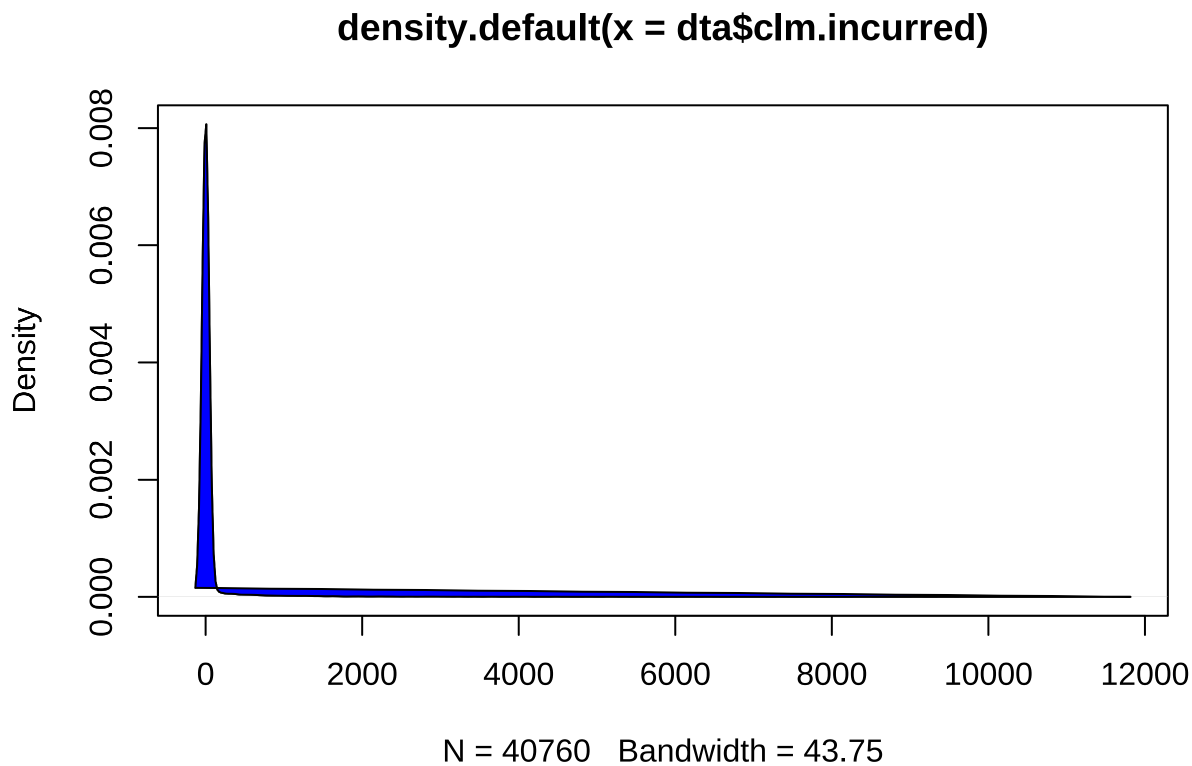
```
## 855.5337835279282
```

Density plot

```
g_dens <- dta %>% filter( clm.incurred<300 ) %>% ggplot( aes(x = clm.incurred)) +
  geom_density(data = dta, col = 'black', fill = couleur, alpha = 0.8) +

  ggtitle("Car Insurance Data - Claim Severity")
g_dens
```




```
# Kernel Density Plot
d <- density(dta$clm.incurred) # returns the density data

plot(d) # plots the results
polygon(d, col="blue", border="black")
```



Histogram

Visualize the age distribution with a histogram

```
driver.age_hist <-ggplot(dta, aes(x=driver.age)) + theme_bw() +
  geom_histogram(binwidth = 1, data=dta, col = "black", fill = couleur, alpha = 0.5) +
  labs(y = "Absolute frequency") +
  ggtitle("Driver ager")
driver.age_hist
```

Statistics showing the total number of contract, the total exposure, and the claim reported by age:

```
library(dplyr)
dta %>% group_by(driver.age) %>% summarise(count_obs=n(),
                                           total_expo = sum(exposure),
                                           total_claims = sum(clm.count)
                                           ) %>% as.data.frame()
```

```
##    driver.age count_obs total_expo total_claims
## 1          18         6       3.24            2
## 2          19         4       2.17            4
## 3          20        19       9.50            4
## 4          21        56      29.27            7
## 5          22        66      34.74            7
## 6          23       106      55.19           17
## 7          24       155      78.05           19
## 8          25       189     100.01           28
## 9          26       427     222.20           59
## 10         27       446     231.74           55
## 11         28       632     327.08           80
## 12         29       707     366.08           83
## 13         30       854     438.46          103
## 14         31       855     438.18           78
## 15         32       882     445.10           93
## 16         33      1170     599.69          132
## 17         34      1220     619.56          101
```

```
## 18          35       1322    667.37            99
## 19          36       1307    659.02           108
## 20          37       1385    710.24           112
## 21          38       1447    737.52           101
## 22          39       1371    695.38            93
## 23          40       1407    717.20           105
## 24          41       1425    724.80           117
## 25          42       1245    634.75           110
## 26          43       1248    625.58            83
## 27          44       1219    622.22           104
## 28          45       1220    616.13            81
## 29          46       1389    719.67           124
## 30          47       1430    725.62           132
## 31          48       1371    692.74           101
## 32          49       1255    639.92           107
## 33          50       1293    665.33            95
## 34          51       1139    588.84            82
## 35          52       1064    544.80            86
## 36          53        907    454.21            76
## 37          54        928    478.25            83
## 38          55        801    416.38            64
## 39          56        866    442.46            71
## 40          57        742    378.04            58
## 41          58        659    334.66            58
## 42          59        613    314.08            47
## 43          60        583    301.13            49
## 44          61        513    262.73            31
## 45          62        483    243.49            23
## 46          63        415    211.23            30
## 47          64        367    188.45            22
## 48          65        312    156.04            16
## 49          66        188     96.16             9
## 50          67        168     84.80             9
## 51          68        139     70.58             7
## 52          69        155     74.91            14
## 53          70        111     56.40            10
## 54          71         83     43.24             5
## 55          72         71     36.76             4
## 56          73         50     26.19             7
## 57          74         56     28.18             1
## 58          75         35     16.86             2
## 59          76         36     18.32             7
## 60          77         44     23.52             8
## 61          78         24     11.01             3
## 62          79         18      9.75             1
## 63          80         12      5.34             2
## 64          81          7      3.26             1
## 65          82         10      5.07             0
## 66          83          9      4.41             0
## 67          84          8      4.32             0
## 68          85          5      2.33             0
## 69          86          4      1.66             0
## 70          87          1      0.33             0
## 71          88          1      0.42             0
```

```
## 72          89          2       1.09            1
## 73          90          1       0.25            0
## 74          93          2       1.09            0
```

Pattern detection

```
library(dplyr)
library(car)
```

```
## Warning: package 'car' was built under R version 4.1.1
```

```
## Loading required package: carData
```

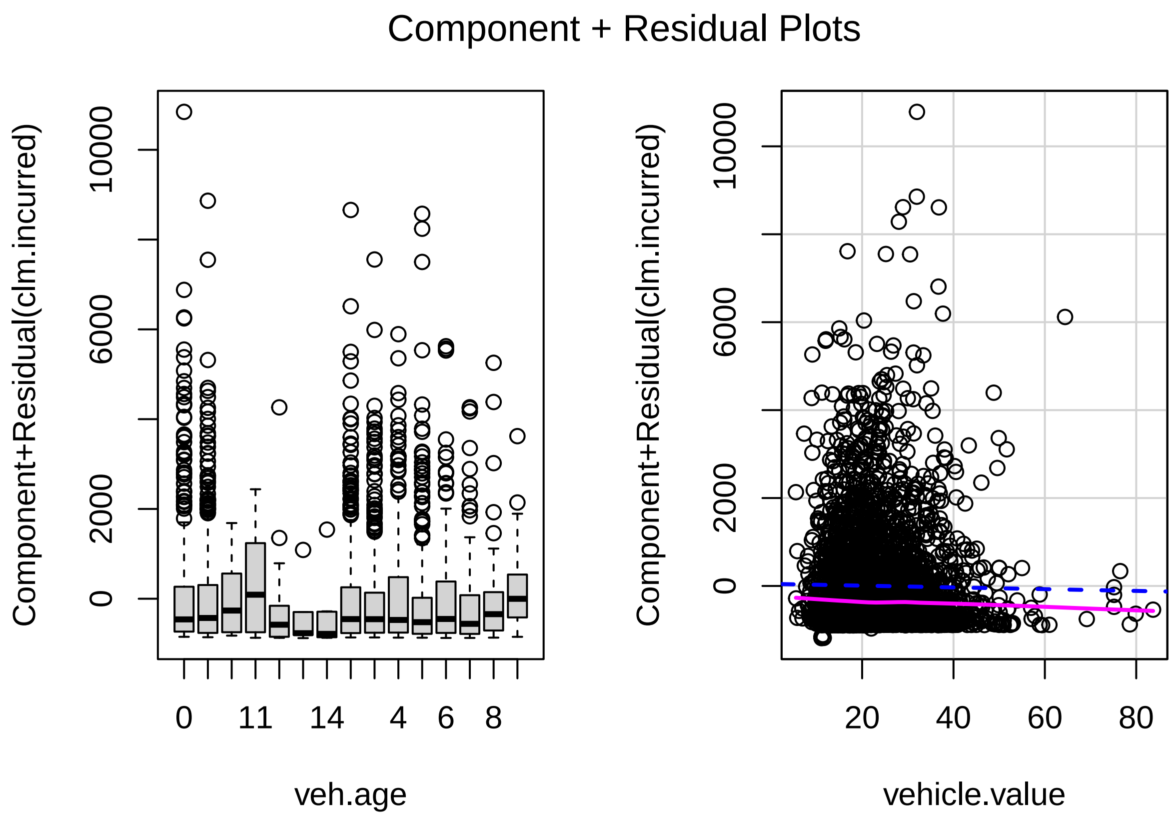
```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
dta2 <- dta %>% filter(clm.incurred > 0)
dim(dta2)
```

```
## [1] 3169   27
```

```
# Fit the model
fit_age_1=lm(clm.incurred ~  veh.age +
vehicle.value , data=dta2)

# Plot the partial residuals
# https://www.statology.org/partial-residual-plot-in-r/
#https://www.r-bloggers.com/2012/01/r-regression-diagnostics-part-1/


# Component residual plots, an extension of partial residual plots, are a good way to see if the predic


crPlots(fit_age_1)
```



The blue line shows the expected residuals if the relationship between the predictor and response variable was linear. The pink line shows the actual residuals.

If the two lines are significantly different, then this is evidence of a nonlinear relationship. Here, the two lines are matching. The hypothesis of linearity is accepted.

```
library(dplyr)
# Remove null value
dta2 <- dta %>% filter(clm.incurred > 0)
dim(dta2)
```

```
## [1] 3169   27
```

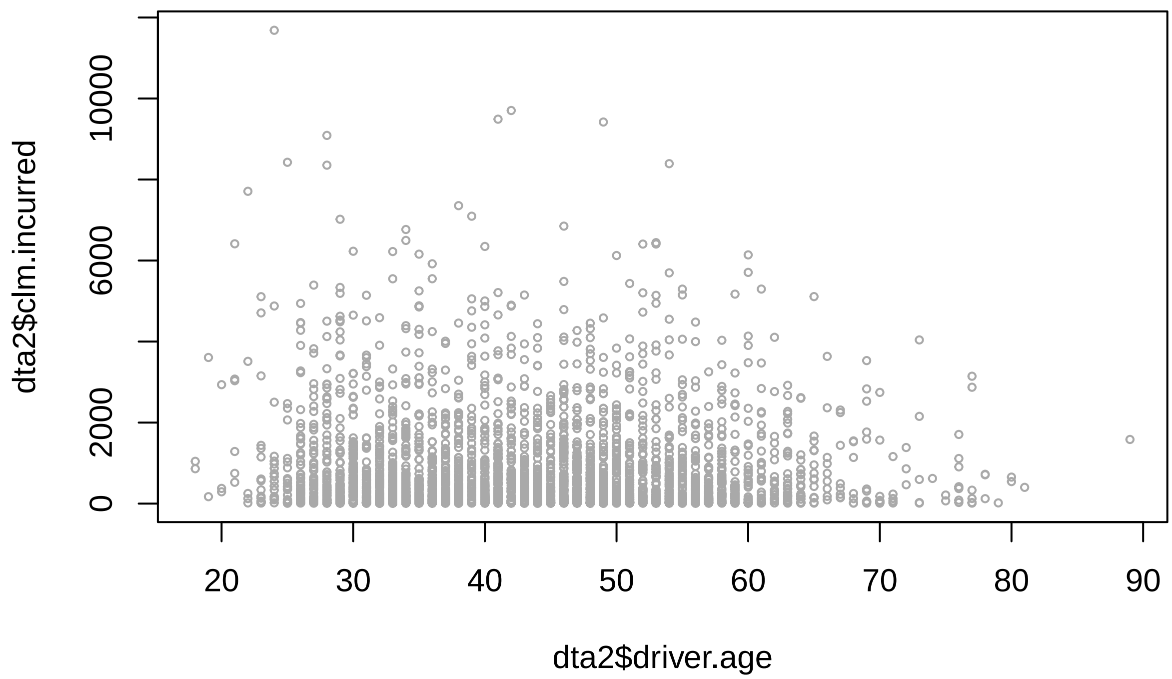
```
# 1. Fit a model polynomial degree-4
fit_age=lm(clm.incurred ~poly(driver.age,4),data=dta2)

# Print the coeff
coef(summary(fit_age))
```

```
##                         Estimate Std. Error   t value      Pr(>|t|)
## (Intercept)             855.5338   21.28593 40.192464 9.717898e-286
## poly(driver.age, 4)1 -2596.5255 1198.26715 -2.166900  3.031685e-02
## poly(driver.age, 4)2  3383.1784 1198.26715  2.823392  4.781550e-03
## poly(driver.age, 4)3 -4163.7644 1198.26715 -3.474821  5.180628e-04
## poly(driver.age, 4)4  2881.8700 1198.26715  2.405031  1.622817e-02
```

```r
# Select min and max ages of the population
agelims=range(dta2$driver.age)
print(agelims)
```

```
## [1] 18 89
```

```r
# 18 80

# Plot the graph of the observation
plot(dta2$driver.age,dta2$clm.incurred,xlim=agelims,cex=.5,col="darkgrey")
```



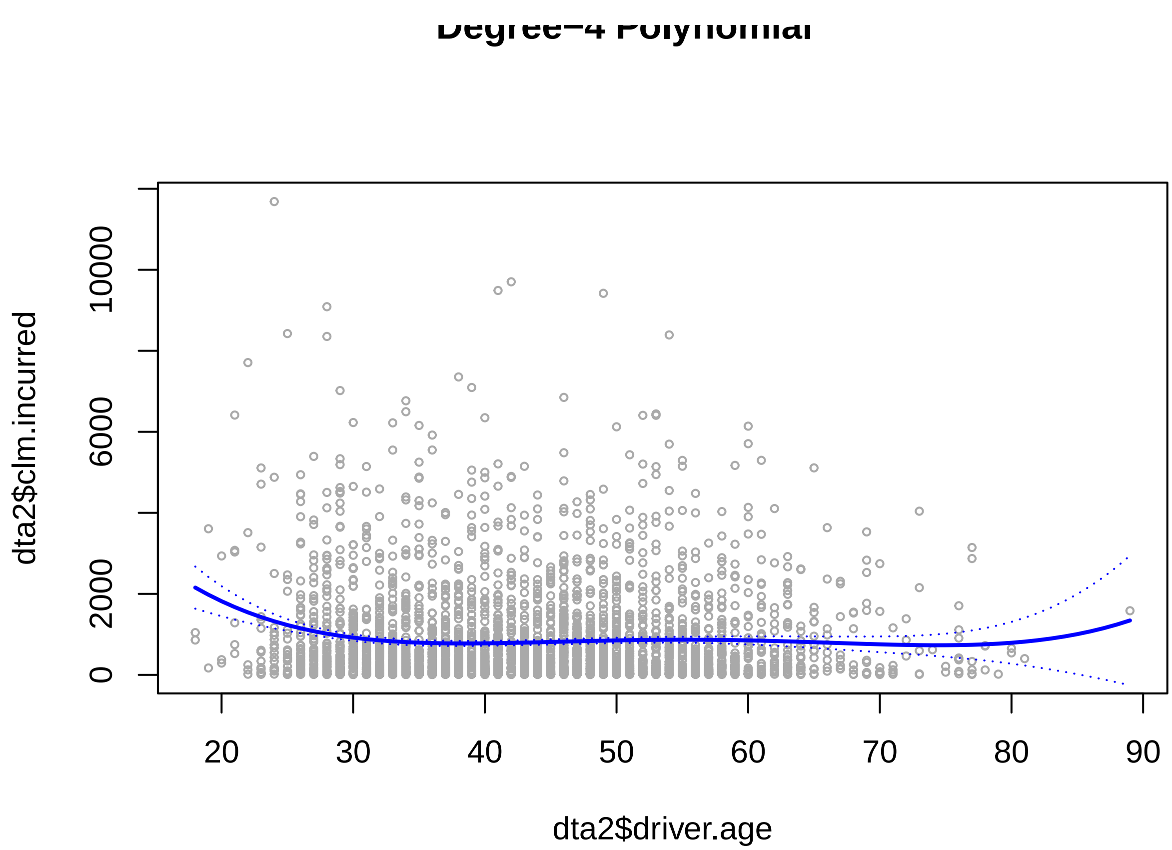

```r
# Create a vector for ages present in the sample
age.grid=seq(from=agelims[1],to=agelims[2])
print(age.grid)
```

```
##  [1] 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
## [26] 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67
## [51] 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
```

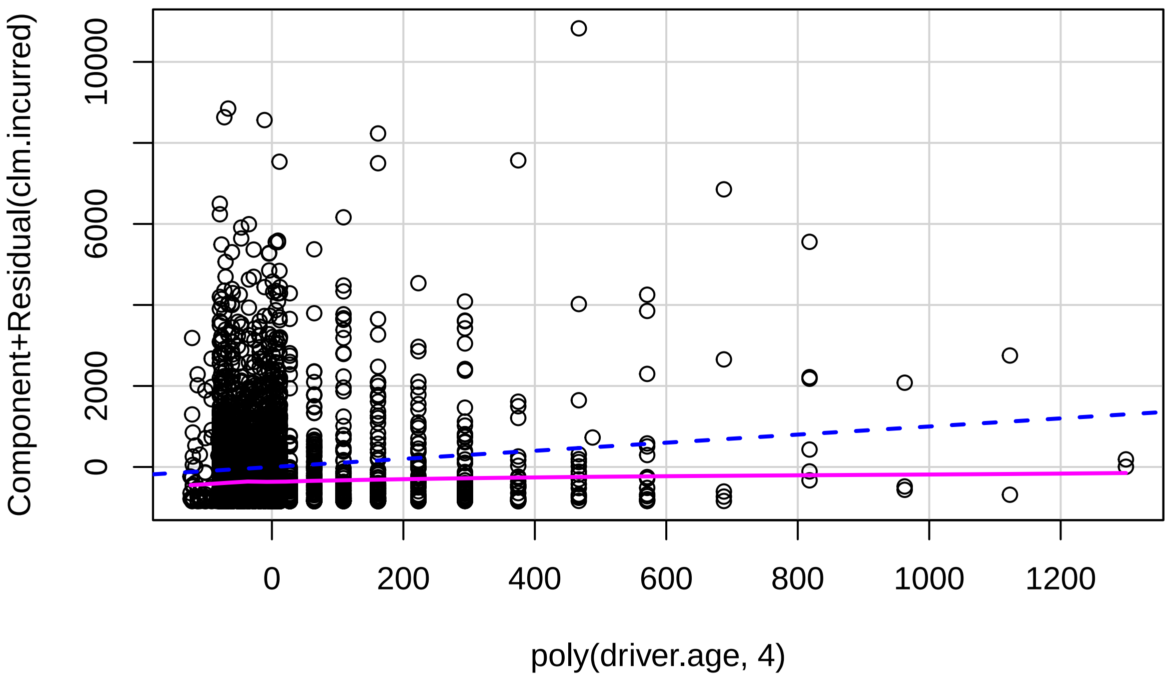
```r
# Prediction for all ages
preds=predict(fit_age,newdata=list(driver.age=age.grid),se=TRUE)

# Creation of the CI bands
se.bands=cbind(preds$fit+2*preds$se.fit,preds$fit-2*preds$se.fit)
```

```r
#par(mfrow=c(1,2),mar=c(4.5,4.5,1,1),oma=c(0,0,4,0))
```

```r
plot(dta2$driver.age,dta2$clm.incurred,xlim=agelims,cex=.5,col="darkgrey")
title("Degree-4 Polynomial",outer=T)

# print the prediction
lines(age.grid,preds$fit,lwd=2,col="blue")

# print the CI
matlines(age.grid,se.bands,lwd=1,col="blue",lty=3)
```




```r
library(car)
crPlots(fit_age)
```



?ns ??plot.Gam

Gam snipset

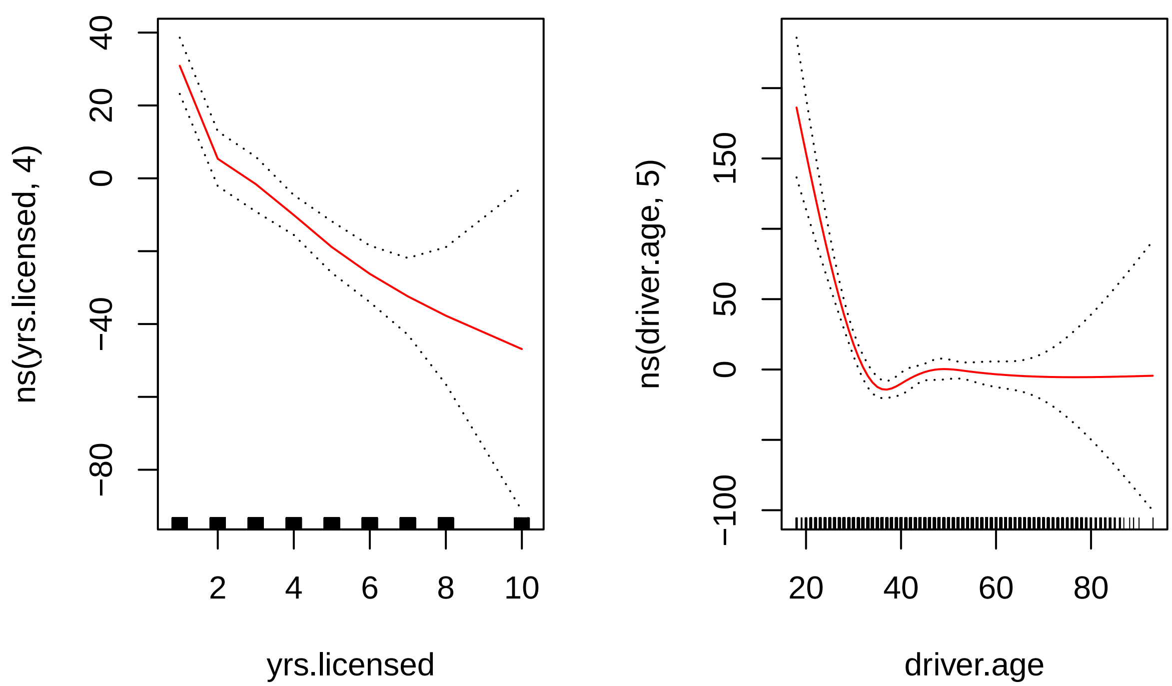
Here 4 and 5 represents the degree of freedom.

```
library(splines)
gam_age=lm(clm.incurred~ns(yrs.licensed,4)+ns(driver.age,5),data=dta)
summary(gam_age)
```

```
##
## Call:
## lm(formula = clm.incurred ~ ns(yrs.licensed, 4) + ns(driver.age,
##     5), data = dta)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -283.6   -74.1   -59.5   -43.3 11494.5
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             283.618     24.914  11.384  < 2e-16 ***
## ns(yrs.licensed, 4)1    -30.764      7.714  -3.988 6.67e-05 ***
## ns(yrs.licensed, 4)2    -46.611     11.175  -4.171 3.04e-05 ***
## ns(yrs.licensed, 4)3    -95.309     14.011  -6.802 1.04e-11 ***
```

```
## ns(yrs.licensed, 4)4  -63.594     22.998  -2.765  0.00569 **
## ns(driver.age, 5)1   -193.090     22.878  -8.440  < 2e-16 ***
## ns(driver.age, 5)2   -184.126     27.000  -6.819 9.27e-12 ***
## ns(driver.age, 5)3   -104.731     19.082  -5.489 4.08e-08 ***
## ns(driver.age, 5)4   -384.913     59.901  -6.426 1.33e-10 ***
## ns(driver.age, 5)5    -86.155     49.164  -1.752  0.07971 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 405.3 on 40750 degrees of freedom
## Multiple R-squared:  0.004875,   Adjusted R-squared:  0.004656
## F-statistic: 22.18 on 9 and 40750 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,2))
gam::plot.Gam(gam_age, se=TRUE, col="red")
```



Holding age fixed, the severity tends to decrease with years of experience. For age, we see an increase of severity after 40 years old.

```
library(gam)
```

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

```
## Loading required package: foreach
```

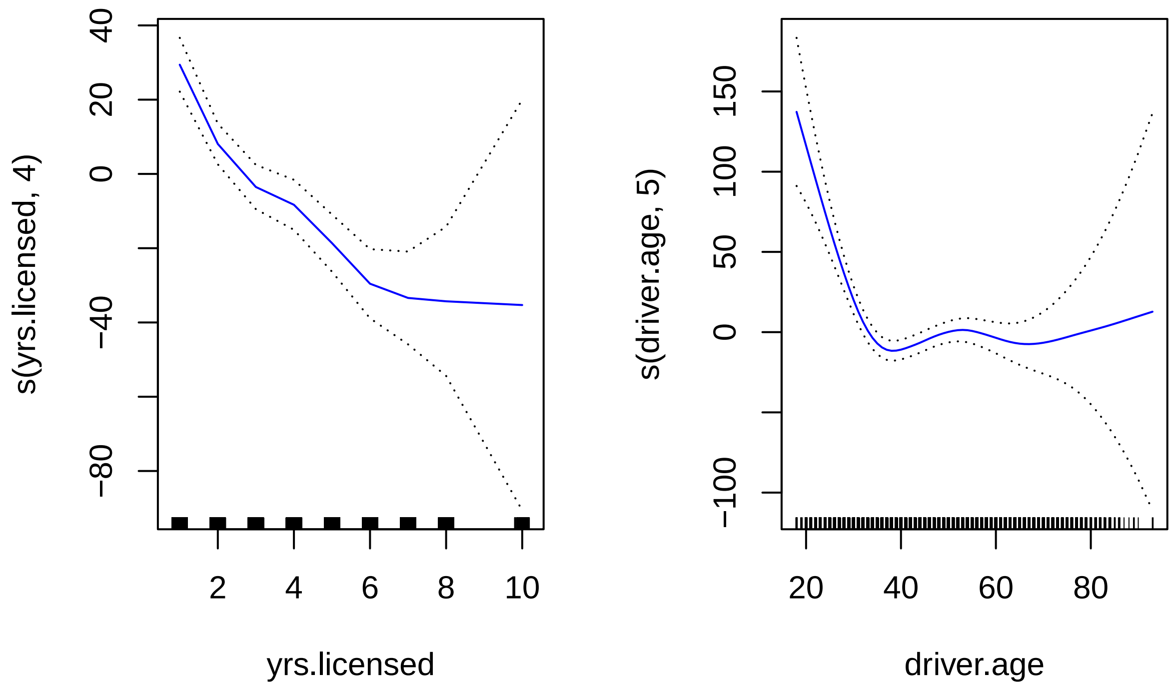
```
## Warning: package 'foreach' was built under R version 4.1.1
```

```
## Loaded gam 1.20
```

```r
gam.age2=gam(clm.incurred~s(yrs.licensed,4)+s(driver.age,5),data=dta)
summary(gam.age2)
```

```
##
## Call: gam(formula = clm.incurred ~ s(yrs.licensed, 4) + s(driver.age,
##     5), data = dta)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
##  -233.21   -80.96   -59.59   -44.36 11512.95
##
## (Dispersion Parameter for gaussian family taken to be 164250.2)
##
##     Null Deviance: 6726015692 on 40759 degrees of freedom
## Residual Deviance: 6693197584 on 40750 degrees of freedom
## AIC: 605176.7
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                       Df     Sum Sq  Mean Sq  F value   Pr(>F)
## s(yrs.licensed, 4)     1   17275736 17275736 105.1794 < 2e-16 ***
## s(driver.age, 5)       1    1485115  1485115   9.0418 0.00264 **
## Residuals          40750 6693197584   164250
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                    Npar Df  Npar F     Pr(F)
## (Intercept)
## s(yrs.licensed, 4)       3  3.7941  0.009836 **
## s(driver.age, 5)         4 16.7108 1.081e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
par(mfrow=c(1,2))
plot(gam.age2, se=TRUE,col="blue")
```



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
```

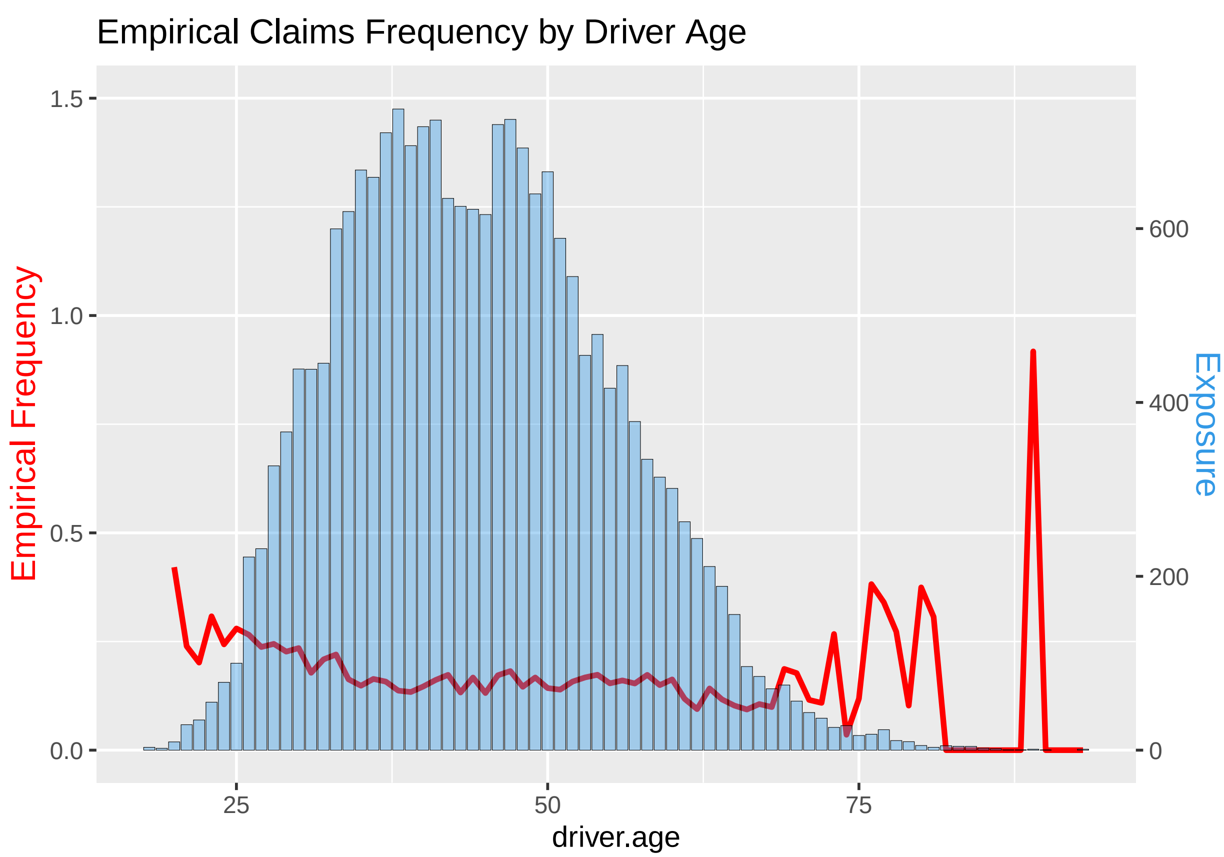
We want to show the Evolution of the Empirical frequency and Exposure for all three years of the training data by driver age.

```
library(ggplot2)
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")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# 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.

```
# 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))
```

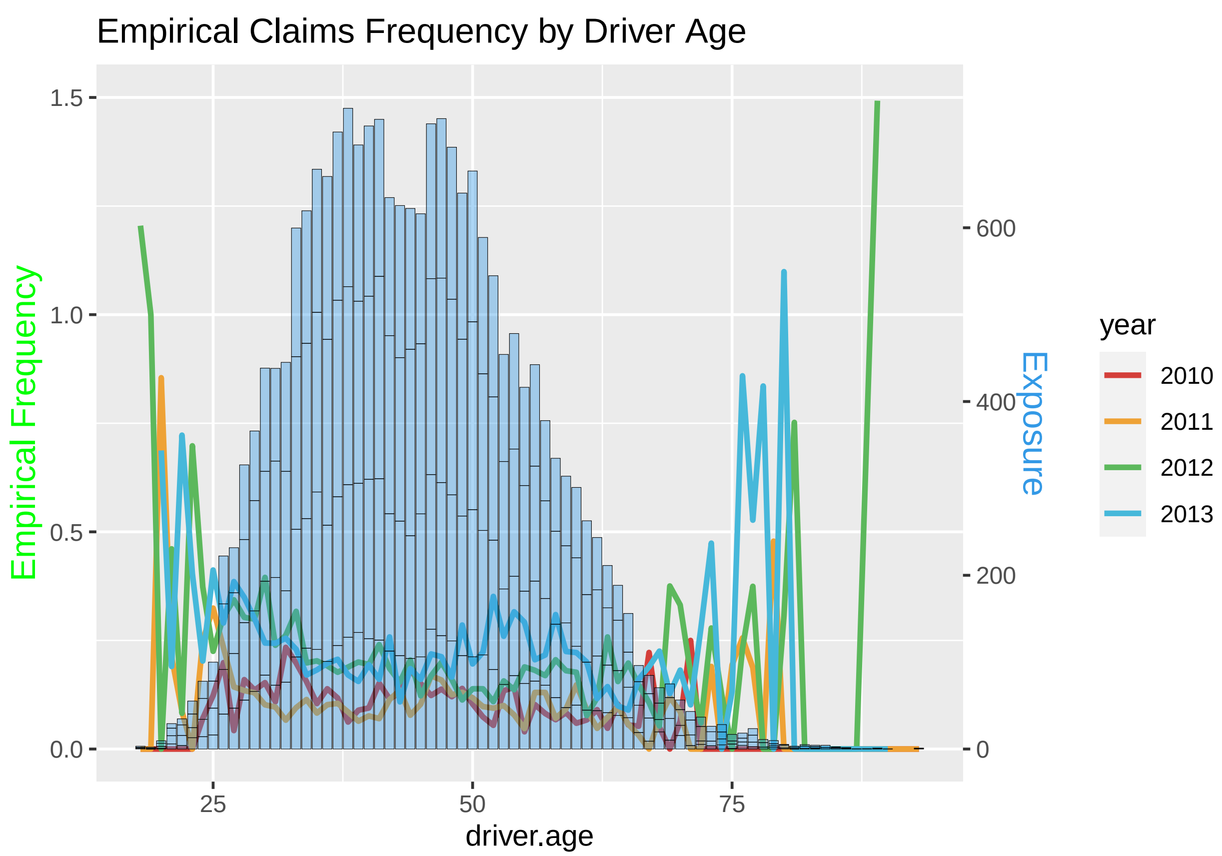
```
## `summarise()` has grouped output by 'driver.age'. You can override using the
## `.groups` argument.
```

```
# 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
```

```
# 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.

```r
# 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
```

```r
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
```

```r
# 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)
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
# 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.

```r
# 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, ]
```

```r
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

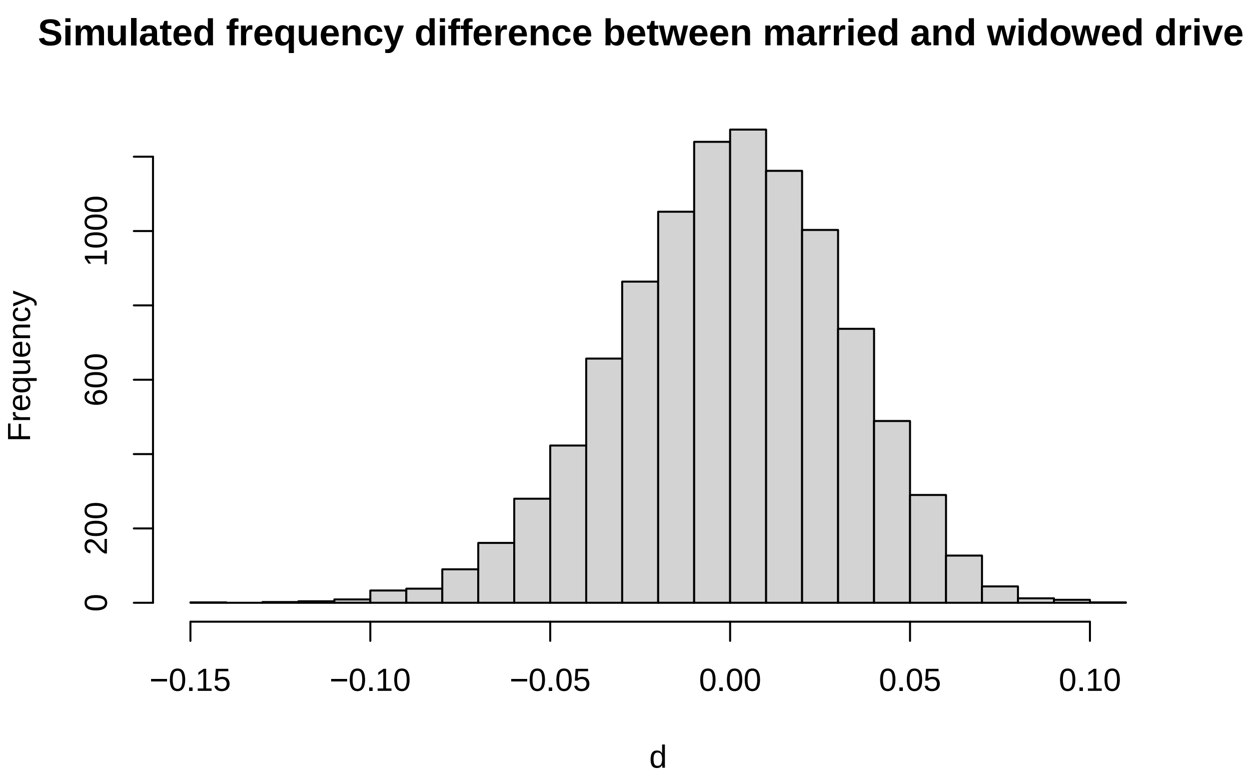
```r
quantile(d, c(0.025, 0.05, 0.1, 0.25, 0.5,
0.75, 0.9, 0.95, 0.975))
```

```
##         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".

```r
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.

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

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

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

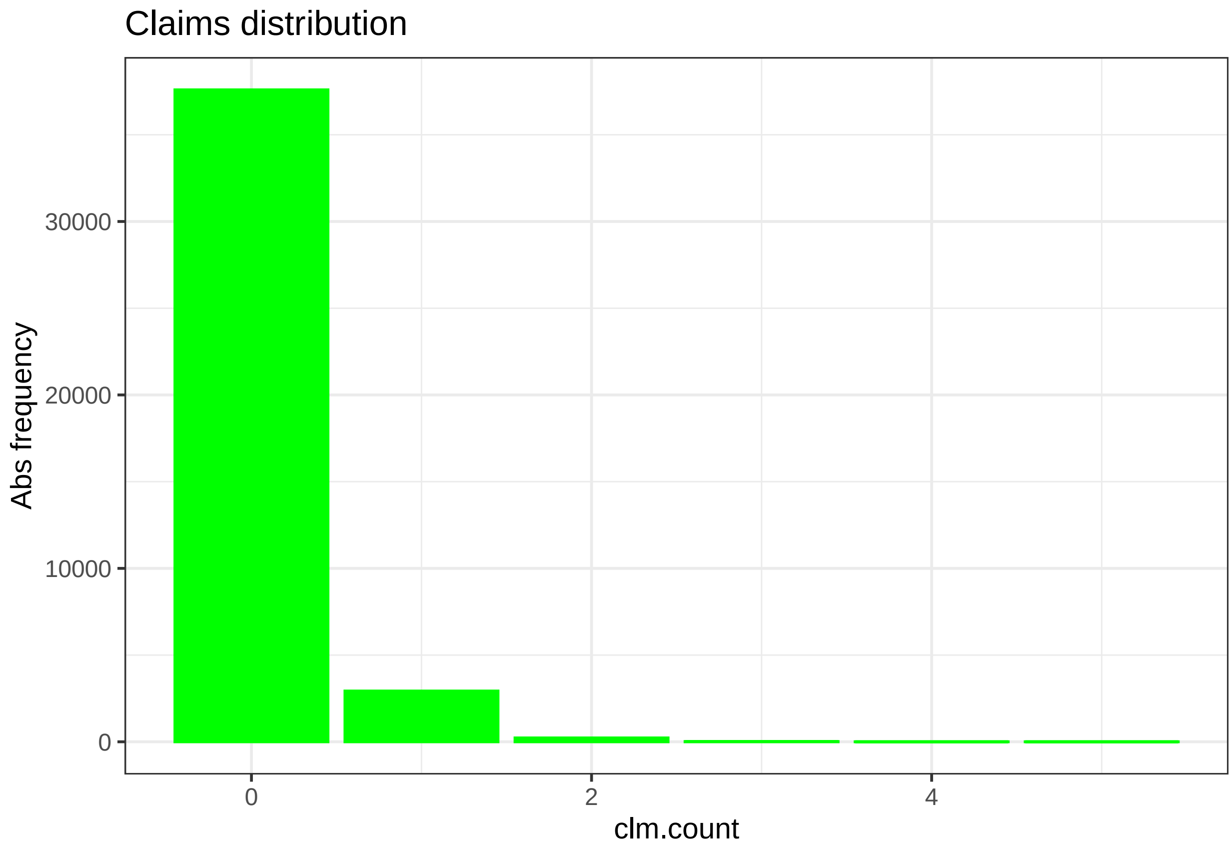
```
##   nb.rb   clm_inc clm.cnt severity
## 1    NB 2160463.5    2633 820.5330
## 2    RB  550723.1     798 690.1291
```

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

```r
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.