EDA - Part 1 - Data Exploration

A glimpse on "classic" insurance data.

This study shows different steps to analyze the data before diving into the modeling part.

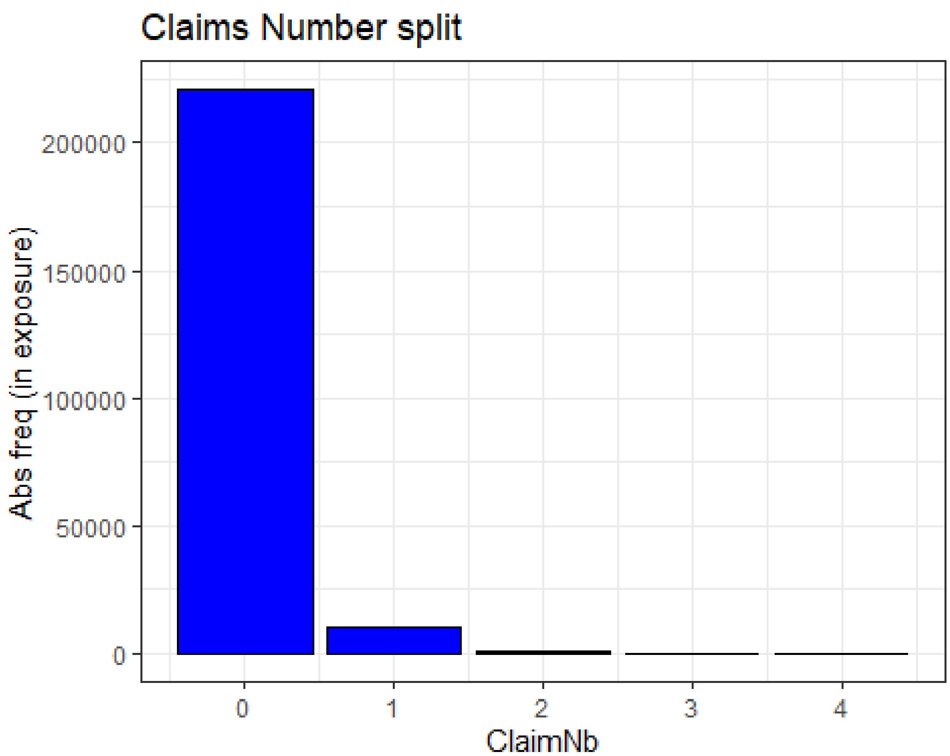
```
# Usual libraries
library(dplyr)

##
## 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

library(rlang)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(ggplot2)
library(tidyr)
library(broom) # convert statistical object into tidy table

# Load the data
df<-read.csv("C:\\Users\\William\\Documents\\Data Science - ML\\Pricing Proje
ct_GLM_vs_GBM\\data.csv")

# Replace the NA by 0 for severity
df <- df %>% mutate(ClaimAmount = ifelse(is.na(ClaimAmount), 0, ClaimAmount))

dim(df)

## [1] 413960     11

glimpse(df)

## Rows: 413,960
## Columns: 11
## $ PolicyID    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17,…
## $ ClaimNb     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
```

```
## $ Exposure    <dbl> 0.09, 0.84, 0.52, 0.45, 0.15, 0.75, 0.81, 0.05, 0.76,
0.34…
## $ Power       <chr> "g", "g", "f", "f", "g", "g", "d", "d", "d", "i", "f",
"f"…
## $ CarAge      <int> 0, 0, 2, 2, 0, 0, 1, 0, 9, 0, 2, 2, 0, 0, 0, 0, 0, 0,
0, 0…
## $ DriverAge   <int> 46, 46, 38, 38, 41, 41, 27, 27, 23, 44, 32, 32, 33, 33
, 33…
## $ Brand       <chr> "Japanese (except Nissan) or Korean", "Japanese (excep
t Ni…
## $ Gas         <chr> "Diesel", "Diesel", "Regular", "Regular", "Diesel", "D
iese…
## $ Region      <chr> "Aquitaine", "Aquitaine", "Nord-Pas-de-Calais", "Nord-
Pas-…
## $ Density     <int> 76, 76, 3003, 3003, 60, 60, 695, 695, 7887, 27000, 23,
23,…
## $ ClaimAmount <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
```

Basic Charts

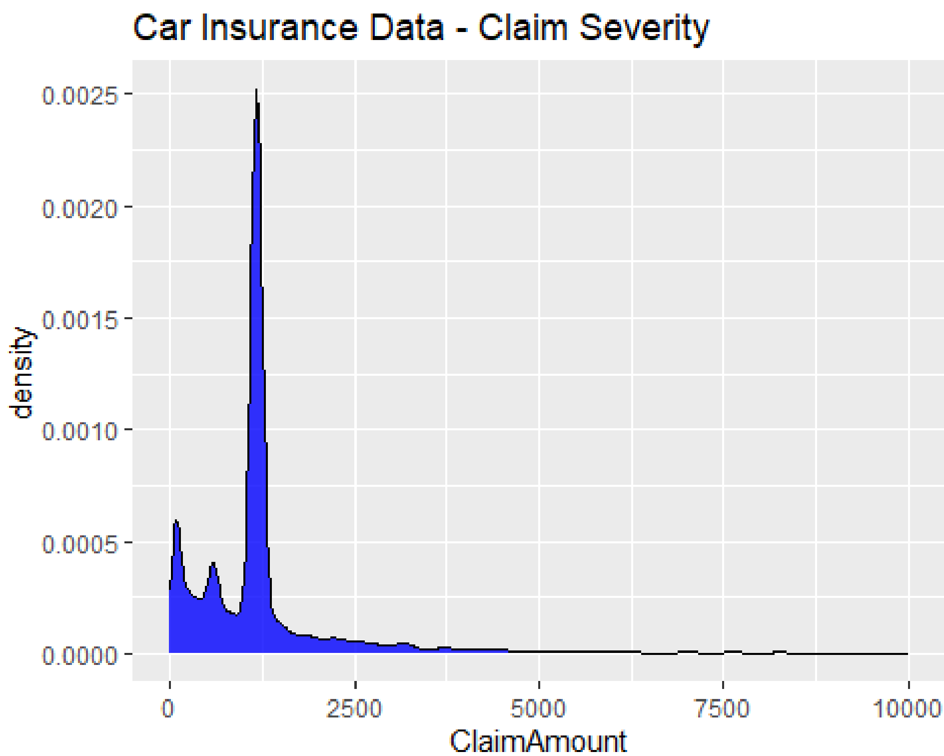
A bar chart showing the claims count split:

```
couleur <- "blue"
g <- ggplot(df, aes(ClaimNb )) + theme_bw() +
geom_bar(aes(weight = Exposure), col = "black",
fill = couleur) +
labs(y = "Abs freq (in exposure)") +
ggtitle("Claims Number split")
g
```



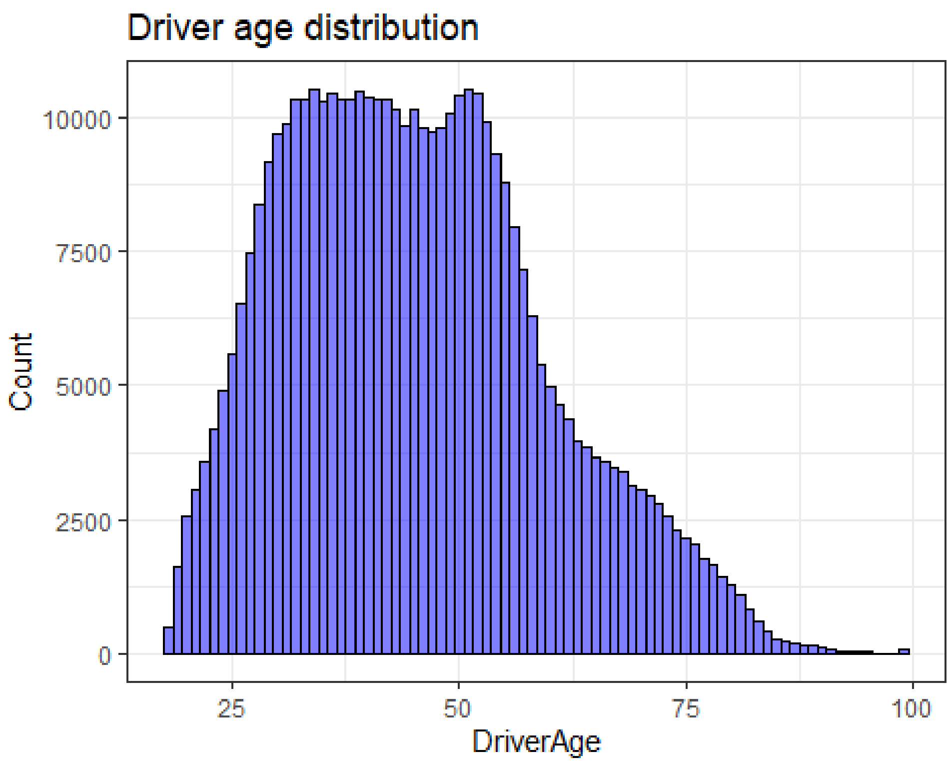
Claims severity density, with its right-skewed shate distribution. Gamma or Negative log-Normal are often the most usual candidates to model the severity of a claim.

```
g_dens <- df%>% filter(ClaimAmount %in% c(1:10000)) %>% ggplot( aes(x = Claim
Amount)) +
geom_density(data = df%>% filter(ClaimAmount %in% c(1:10000)), col = 'black',
fill = couleur, alpha = 0.8) +
ggtitle("Car Insurance Data - Claim Severity")
g_dens
```



We can visualize the age distribution with a histogram:

```
driver.age_hist <-ggplot(df, aes(x=DriverAge)) + theme_bw() +
geom_histogram(binwidth = 1, data=df, col = "black", fill = couleur, alpha =
0.5) +
labs(y = "Count") +
ggtitle("Driver age distribution")
driver.age_hist
```



Basic Interpretation

Null model

We start with the model with no parameters, only the intercept.

```
#########################################
# Training a model for claims frequency #
#########################################

# Split train / test
# index <- createDataPartition(df$ClaimNb, p = 0.7, list = FALSE)
# head(index)
#
# train <- df[index,]
# test <- df[-index,]

set.seed(564738291)
u <- runif(dim(df)[1], min = 0, max = 1)
df$train <- u < 0.7
df$test <- !(df$train)
#mis.vars <- c(mis.vars, "train", "test")

# Step 1:
# Null Model
null_model <- glm(formula = ClaimNb ~ 1,
```

```r
                  family = poisson(link = "log"),
                  data = df,
                  subset = train, offset = log(Exposure))

summary(null_model)

##
## Call:
## glm(formula = ClaimNb ~ 1, family = poisson(link = "log"), data = df,
##     subset = train, offset = log(Exposure))
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.572821   0.008979  -286.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 80222  on 289487  degrees of freedom
## Residual deviance: 80222  on 289487  degrees of freedom
## AIC: 103429
##
## Number of Fisher Scoring iterations: 6

coefficients(null_model)

## (Intercept)
##   -2.572821

# Verification if the exp of the intercept is equal to the
# empirical frequency (mean)
exp(null_model$coefficients) #ok mean of the number of claims per year.

## (Intercept)
##  0.07631992

emp_freq <- sum(df$ClaimNb)/sum(df$Exposure)

predict(null_model,newdata=data.frame(Exposure=1))

##         1
## -2.572821

predict(null_model,type="response", newdata=data.frame(Exposure=1)) # takes t
he exponential of the coefficient

##          1
## 0.07631992
```

We verify that the null model is only composed by the intercept which is equal to the empirical frequency shown by the dataset.

Coefficient interpretations

```r
# Step 2:
# Exploration variable per variable
with(df,table(Gas, ClaimNb)) # we don't have the same exposition

##          ClaimNb
## Gas            0      1      2      3      4
##   Diesel  197904   7655    738     45      8
##   Regular 199875   6978    714     39      4

# the exposure avoids to make easy conclusion

# With gas
m1 <- glm(formula = ClaimNb ~ Gas,
          family = poisson(link = "log"),
          data = df,
          subset = train, offset = log(Exposure))
summary(m1)

##
## Call:
## glm(formula = ClaimNb ~ Gas, family = poisson(link = "log"),
##     data = df, subset = train, offset = log(Exposure))
##
## Coefficients:
##             Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -2.50360    0.01243 -201.398  < 2e-16 ***
## GasRegular  -0.13963    0.01798   -7.768 7.97e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 80222  on 289487  degrees of freedom
## Residual deviance: 80162  on 289486  degrees of freedom
## AIC: 103370
##
## Number of Fisher Scoring iterations: 6
```

Interpretation: The variable “regular” is significantly different from “diesel”. We should be -14% less high in term of claim frequency for the regular car.

```r
# Prediction on the levels taken separately
predict(m1,type="response", newdata=data.frame(Gas = c("Regular", "Diesel"),
                                               Exposure=1))

##          1          2
## 0.07113073 0.08178971

# Intercept
m1$coefficients[1]
```

```
## (Intercept)
##   -2.503604
```

```r
exp(m1$coefficients[1])
```

```
## (Intercept)
##  0.08178971
```

```r
# Regular level coefficent
m1$coefficients[2]
```

```
## GasRegular
##  -0.139632
```

```r
exp(m1$coefficients[2])
```

```
## GasRegular
##  0.8696783
```

```r
# We can verify the results:
# A frequency of 7%,
print(0.08178971 * 0.8696783)
```

```
## [1] 0.07113074
```

```r
# Which represent ~13% less than the average claim frequency for Diesel drive
r, everything else constant.
print((0.07113074-0.08178971)/0.08178971)
```

```
## [1] -0.1303217
```

We find the results given by the prediction.

AIC and Deviance graph

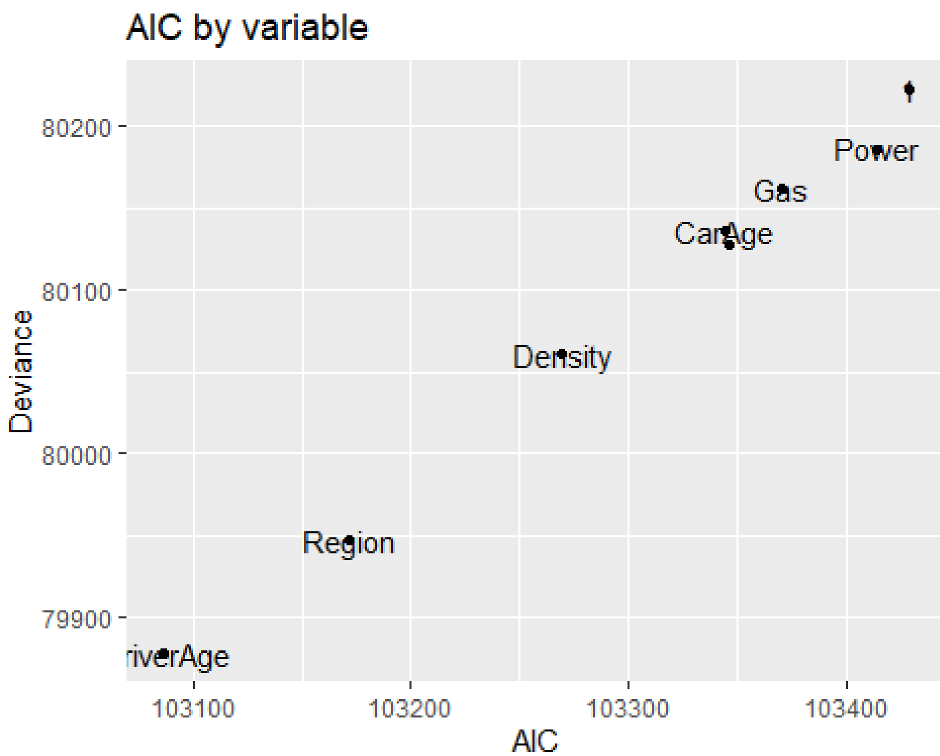
A representation to get a feel of what would be the most “interesting” predictors in terms of AIC and Deviance reduction:

```r
##############################################
# Step 2: Evaluation of potential predictors #
##############################################

# Test of the different potential covariates

# Set up a grid search
result_grid <- expand.grid(
  covariates = c(1, 'Power', 'CarAge', 'DriverAge', 'Brand', 'Gas', 'Region',
'Density'),
  AIC = NA,
  Deviance = NA)
# print(result_grid)
```

```r
# Run a for loop adding building each time a model with one parameter
for(i in seq_len(nrow(result_grid))) {
  fmla <- as.formula(paste("ClaimNb ~ ", result_grid$covariates[i]))
  f <- glm(fmla,
           data = df,
           subset = train,
           family = poisson(link = "log"),
           offset = log(Exposure))
  #rms[v] <- RMSEP(dta$clm.count[dta$train],
  #predict(f, newdata = dta[dta$train,],
  #type = "response"))
  result_grid$AIC[i] <- f$aic
  result_grid$Deviance[i] <- f$deviance
}
knitr::kable(result_grid, format = "markdown")
```

covariates	AIC	Deviance
1	103428.7	80222.18
Power	103414.5	80185.95
CarAge	103344.6	80136.04
DriverAge	103086.7	79878.10
Brand	103346.0	80127.47
Gas	103370.3	80161.77
Region	103171.5	79946.95
Density	103269.6	80060.99

```r
#clipr::write_clip(result_grid)

# Graph AIC & Deviance
scatter <- ggplot(result_grid, aes(x=AIC, y=Deviance)) +
  geom_point() + # Show dots
  geom_text(
    label=result_grid$covariates,
    nudge_x = 0.25, nudge_y = 0.25,
    check_overlap = T
  ) +
  labs(
    title = "AIC by variable")

# Final result
print(scatter)
```



Driver age and Region are two strong candidates to be included in a claims frequency model. Power looks to have less impact.

Exploration of Region

It appears that some Region can be grouped together. We will keep that observation in mind when training the model.

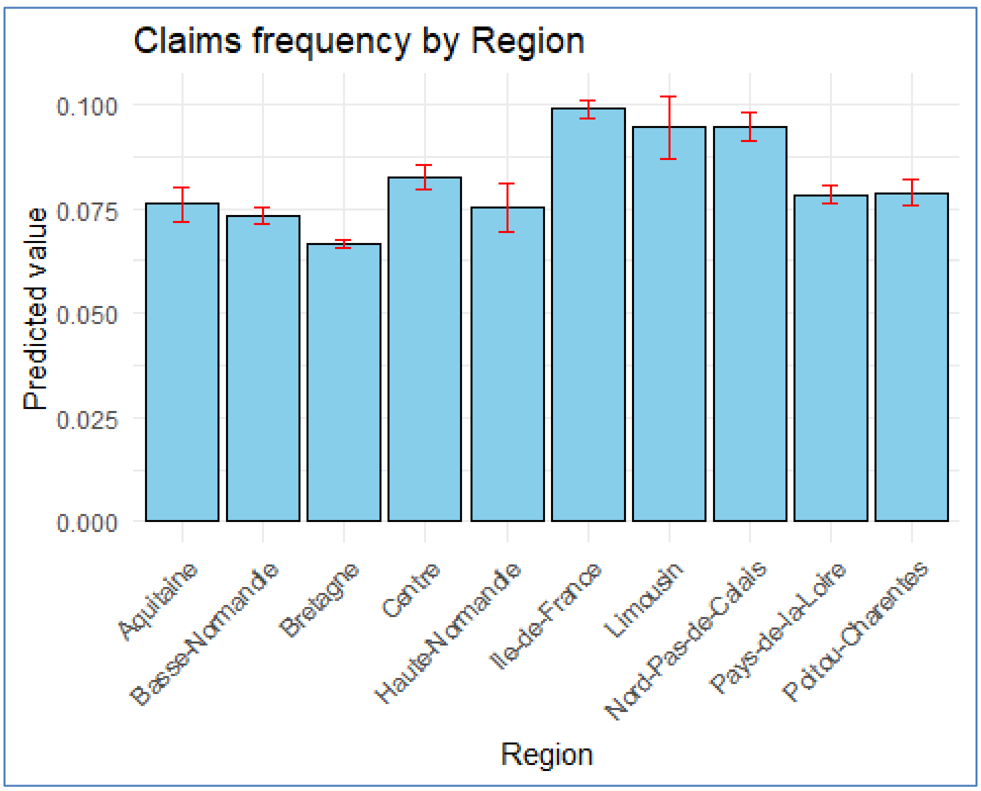
```
# Another variable: Region
with(df, table(Region, ClaimNb))

##                      ClaimNb
## Region                    0      1      2      3      4
##   Aquitaine           30344    919    124     12      0
##   Basse-Normandie     10464    406     46      0      0
##   Bretagne            40329   1718    144      9      0
##   Centre             154339   6053    412      6      4
##   Haute-Normandie      8575    198     22      0      0
##   Ile-de-France       67398   2205    358     24      4
##   Limousin             4383    172     22      3      0
##   Nord-Pas-de-Calais  26413    806    122     12      4
##   Pays-de-la-Loire    37253   1422    148      6      0
##   Poitou-Charentes    18281    734     54     12      0

m2 <- glm(formula = ClaimNb ~ Region,
          family = poisson(link = "log"),
          data = df,
```

```r
          subset = train, offset = log(Exposure))
summary(m2)
```

```
##
## Call:
## glm(formula = ClaimNb ~ Region, family = poisson(link = "log"),
##     data = df, subset = train, offset = log(Exposure))
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -2.49619    0.03488 -71.569  < 2e-16 ***
## RegionBasse-Normandie      -0.08024    0.06351  -1.263  0.20643
## RegionBretagne             -0.11403    0.04378  -2.605  0.00920 **
## RegionCentre               -0.21578    0.03776  -5.715 1.10e-08 ***
## RegionHaute-Normandie      -0.09080    0.08508  -1.067  0.28590
## RegionIle-de-France         0.18308    0.04113   4.451 8.54e-06 ***
## RegionLimousin              0.13843    0.08641   1.602  0.10916
## RegionNord-Pas-de-Calais    0.13750    0.05018   2.740  0.00615 **
## RegionPays-de-la-Loire     -0.05041    0.04521  -1.115  0.26489
## RegionPoitou-Charentes     -0.04529    0.05329  -0.850  0.39543
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 80222  on 289487  degrees of freedom
## Residual deviance: 79947  on 289478  degrees of freedom
## AIC: 103172
##
## Number of Fisher Scoring iterations: 6
```

```r
# Some region are not significant

# Isolate the region's name
region_name <- df %>% group_by(Region) %>% summarise(count=n())

# Run a prediction for each of the Region
# We retrieve 10 avg frequency
y=predict(m2,newdata=
            data.frame(Region=region_name$Region,
                       Exposure=1),type="response",
          se.fit =TRUE) # we add the CI

# Predictions and CI
pred_values <- y$fit
lower_CI <- y$fit-y$se.fit
upper_CI <- y$fit+y$se.fit

# Definition of the region for each prediction
vec_Region <-c("Centre", "Aquitaine", "Basse-Normandie", "Bretagne", "Haute-N
```

```
ormandie", "Ile-de-France", "Limousin", "Nord-Pas-de-Calais", "Pays-de-la-Loi
re", "Poitou-Charentes")

# Create the data frame
predicted_df <- data.frame(predicted_value=pred_values, Region = vec_Region,
upper = upper_CI, lower = lower_CI)

#print(predicted_df)

# Load the ggplot2 package
library(ggplot2)

# Create a bar plot
ggplot(predicted_df, aes(x = Region, y = predicted_value)) +
  geom_bar(stat = "identity",fill = "skyblue", color = "black") +
   geom_errorbar(aes(ymin = lower, ymax = upper),
                width = 0.2, color = "red") +
  labs(title = "Claims frequency by Region", x = "Region", y = "Predicted val
ue") +
  theme_minimal() + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



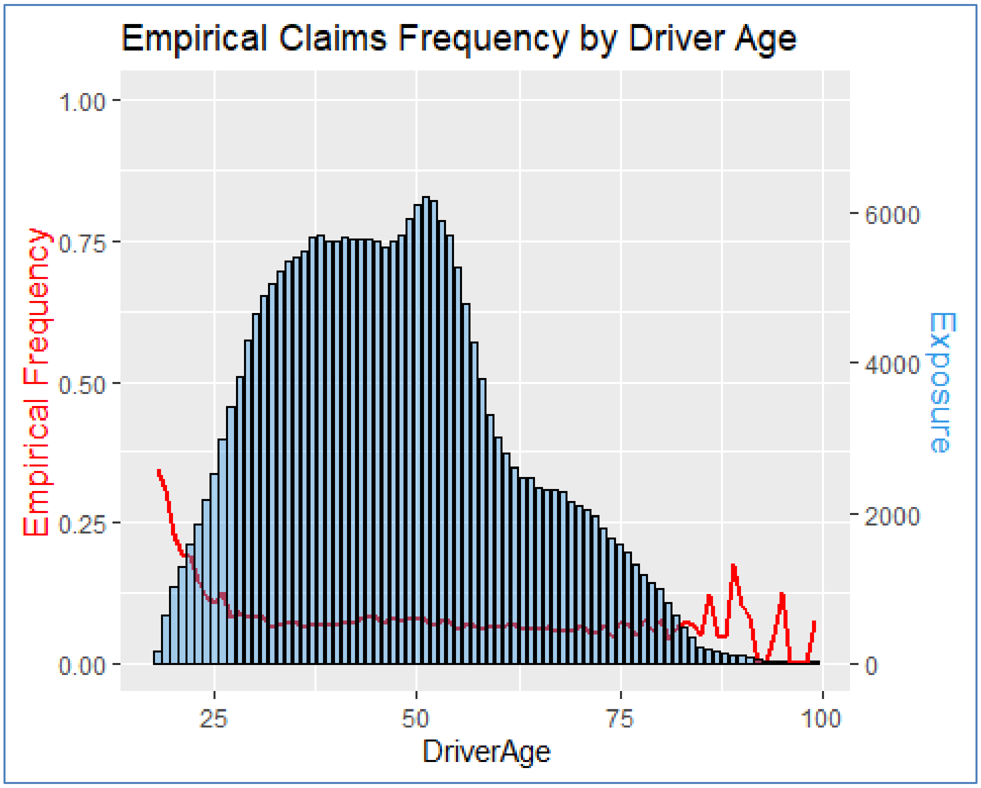
Exploration of Driver's age

```
library(ggplot2)
library(dplyr)
```

```r
# Creation of the data frame
graph_data <- df %>% group_by(DriverAge) %>% summarise(Sum_Expo = sum(Exposur
e),
Number_of_Claims = sum(ClaimNb),
Emp_freq = sum(ClaimNb)/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, 7500) # 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-remove
d-k-rows-containing-missing-values
# 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=DriverAge, 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, colo
r="black", alpha=.4) +
scale_y_continuous(
# Features of the first axis
name = "Empirical Frequency", limits = c(0, 1.0),
# 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.
```

```r
graph_freq
```



The frequency decreases as the driver is more experienced, with a noticeable drop between 18 and 25 years old. The rate becomes more volatile after 75 years old.