Institute of Statisitcs, Biostatistics and Actuarial Sciences

LDAT2310: Data sciences for insurance and finance

Motor insurance dataset: prediction of the claim frequency

Author:
WARNAUTS Aymeric (DATS2M- 87031800)

Professor:
HAINAUT Donatien

2022-2023

1 Introduction

The collection of large amount of data allows insurers to determine the risk profile of their clients through metrics as key ratio's. These ratio's are the result of response feature X updated by an exposure w. In actuarial sciences the main focus will carry out such ratio's as the **loss ratio**, **claim severity** or **claim frequency**. In fact, the latter is a fundamental information as a basic unit used by insurer to measure the amount of risk insured over a given period is the **pure premium = claim frequency x claim severity**. As the **claim frequency** is given by the ratio $\frac{nbclaims}{duration}$ and the **claim severity** one is given by $\frac{claimcost}{nbclaims}$ it's obvious that the estimation of the individual number of claims is imperative. The estimation of the number of claims as a rating factor will be the core of this project. In fact, as regulators now impose insurance company which offer contracts to hold enough capital to fulfill the promises they sell, companies need to allocate premiums carefully. So, individual analysis - which leads to the identification of risky profiles - indicates how to allocate the total premium income among different policies. That's why it's very interesting to perform data analysis on individuals using regression tools in order to determine what cause the claims.

In the dataset provided, the number of claims and the exposure are given, so it's quite simple to compute the target feature for the claim frequency. But the first trade off to be made is to choose between building a model for the claim frequency directly or to model the number of claims seen as a count feature and to compute the claim frequency key ratio subsequently, considering exposure as given. We recall that as the insurer is interested by the pure premium that is the product of a claim frequency and severity the standard GLM tariff analysis is to do separate analysis for claim frequency and severity. This is due to the fact that claim frequency is usually much more stable than claim severity and often much of the power of rating factors is related to claim frequency.

As the data analysis allows to extract useful information from large samples of customer represented by contracts, it's logical to deal with features that give information about the type of guarantee or the splitting of the premium. Moreover, as the databases provided are about motor insurances, we will deal with features underlying people's characteristics as **Gender** or **DriverAge** or directly car features as **Power** or **Fuel**. And as in motor insurance the clients pay a certain amount of premium to get coverage for a vehicle, they have to pay attention to first-party and third-party involved in car insurance too. In fact, commonly as the first party - the insurer client - has a contract with the second party - the insurer - the third party - someone other than the first and second party that is affected by the damage caused by the car - has to be taken into account. Even if some features such as the **Area** capture small part of this third-part potential damage information, it's not enough and the main focus will be to predict the global feature defining the claim frequency, getting rid of the **claim severity** or the **loss ratio**. As one of the main goal for insurance company will be to predict an optimal price for the contract they offer, a complementary analysis has to be made about the **claim cost** and **earned premium** but this will not be covered by this project.

Descriptive analysis: continuous and categorial (*) variables

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
Gender*	1	70000	1.47	0.50	1.00	1.46	0.00	1.00	2.00	1.00	0.12	-1.99	0.00
DriverAge	2	70000	50.01	14.53	50.00	50.02	14.83	18.00	82.00	64.00	-0.01	-0.44	0.05
CarAge	3	70000	5.02	2.56	5.00	4.99	2.97	0.00	16.00	16.00	0.14	-0.28	0.01
Area*	4	70000	2.60	1.28	2.00	2.50	1.48	1.00	5.00	4.00	0.35	-0.93	0.00
Leasing*	5	70000	1.80	0.40	2.00	1.87	0.00	1.00	2.00	1.00	-1.49	0.22	0.00
Power*	6	70000	2.35	0.85	2.00	2.31	1.48	1.00	4.00	3.00	0.23	-0.54	0.00
Fract*	7	70000	2.21	0.86	2.00	2.26	1.48	1.00	3.00	2.00	-0.41	-1.54	0.00
Contract*	8	70000	1.90	0.70	2.00	1.88	0.00	1.00	3.00	2.00	0.13	-0.95	0.00
Fuel*	9	70000	2.05	0.92	2.00	1.94	1.48	1.00	4.00	3.00	0.67	-0.32	0.00
Exposure	10	70000	1.54	0.93	1.51	1.50	0.99	0.08	5.92	5.84	0.33	-0.36	0.00
Nbclaims	11	70000	0.03	0.18	0.00	0.00	0.00	0.00	3.00	3.00	5.68	33.68	0.00

And thus as the DBtrain motor insurance data set contains information about $n =$ **70 000** insurance policies and

$d = \mathbf{10}$ explanatory features, after removing the ID feature, previously checked as there is no repetitions of member ID's, we have also checked for NA's and correlations between features (see appendices) but apparently the data set doesn't need to be filtered or dimension reduced. Same assumption can be made for the levels of our categorical features and thus we do not need to reduce the levels by clustering.

2 Data Description and Preprocessing

For this project we will be concerned with modelling the number of claims and as we want to build a general claim number forecasting model that takes all the main sources of uncertainty into account, it's of main concern to carry out the changing mix of underlying risk units from heterogeneous population.
As we try to predict the number of claims we can try to focus on the distribution of such a feature. This is very special because we can see a large number of observations that are allowed to the level 0 of this categorical variable. It's very common that the observed distribution of the claims count have zero-inflation issue, here **96.8**% of the whole observations for this feature. Moreover, the **Nbclaims** and the computed **ClaimFreq** are positively skewed with heavy-tailed distributions.

Observed claim frequency distribution				
Number of claims k	Number of policies n_k		Total exposure w_k (years)	
	count	%	count	%
0	67 752	96,79	102 923.7	95,6
1	2190	3,13	4 569.18	4,25
2	56	0,08	130.65	0,12
3	2	0,003	3.99	0.03
Total	70 000	100	107 627.5	100

After the computation of the maximum-likelihood estimate of the canonical parameter

$$\hat{\theta} = ln(\frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} w_i}) = ln(\frac{\sum_{k=1}^{3} kn_k}{\sum_{k=0}^{4} w_k}) = \mathbf{-3.84}$$

with a **95** % confidence interval such that $\hat{\theta} \pm 1.96\sqrt{e^{-\theta}\sum_{i=1}^{n} w_i}$ **= [-3.84** $\pm$ **0.04075]** and the observed information equal to:

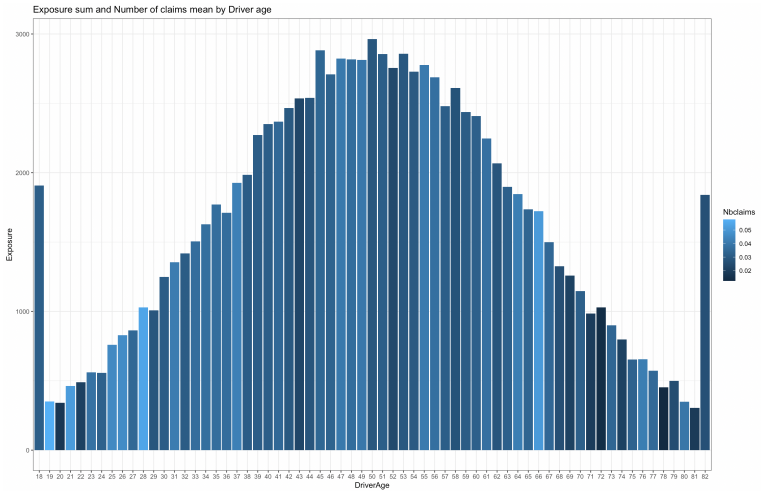
$$I(\hat{\theta}) = exp(\hat{\theta}) \sum_{k=0}^{4} w_k = \mathbf{2313.3}$$

Increasing the total exposure means that more information becomes available, conducing to more comfortable estimate but we have to recall that the more claims are recorded, the more information is contained in the sample data, but we have a huge amount of **0** claims.
Therefore it is obvious that the distribution of claim occurrences should be zero with a very high probability. Such distributions would be discrete distributions with a large parameter (probability of success) since they have the mode at zero and the probability of the other values would become progressively smaller. In fact, as the obvious model for the probability of a claim is the **Bernoulli** one, and assuming the exposure w_i as known with c_i the number of claims over a period, then we can assume a Poisson process with mean number of claims (per unit exposure) λ_i so that $c_i|w_i \sim Poi(w_i\lambda_i)$ with $P(c_i = 1|w_i = 1) = e^{-\lambda_i}$ and finally $P(c_i = 1|w_i) = e^{-w_i\lambda_i} \approx 1 - w_i\lambda_i \approx 1$ when λ_i is small. Then this link the logistic regression and the Poisson GLM's because in this particular case:

$$\frac{\mu_i}{1-\mu_i} = \mu_i = exp(\beta_0) \prod_{j|x_{ij}=1} exp(\beta_j)$$

with the complementary $log - log$ link function that bridges Poisson and Bernoulli GLM's but trying to predict

the probability of causing 0,1,2, or more claims is less robust numerically. So let's go deeper in our analysis.



The classical way to model the distribution of a count feature is the **Poisson distribution** but we will pose more general assumptions:
In fact, from the density function for exponential type we obtain:

$$f_{Y_i}(y_i;\theta_i,\phi) = \exp\{\frac{y_i\theta_i - b(\theta_i)}{\phi/w_i} + c(y_i,\phi,w_i)\}$$

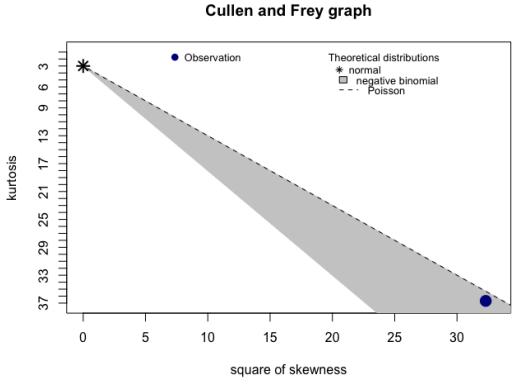
for the density function of Y_i, where the dispersion parameter is the same for all i, equal to **1** for the claim frequency. Moreover, if we assume $N(t)$ the number of claims on [0,t] we can assume that it's distributed following a Poisson Distribution, and thus we will not perform logistic regression under the previously explained Bernoulli assumptions. In addition, for the claim frequency $b(\theta_i) = e^{\theta_i}$ and $\theta_i = log(\mu_i)$.
Let X_i be the number of claims in tariff cell i with duration w_i and let μ_i denote the expectation when $w_i = \mathbf{1}$, so we can write:

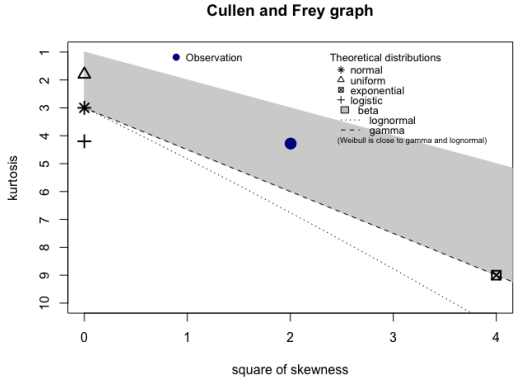
$$f_{X_i}(x_i;\mu_i) = P(X_i = x_i) = e^{-w_i\mu_i}\frac{(w_i\mu_i)^{x_i}}{x_i!}$$

then the **claim frequency** $Y_i = \frac{X_i}{w_i}$ is a **Poisson** variable such that:

$$f_{Y_i}(y_i;\mu_i) = P(Y_i = y_i) = P(X_i = w_iy_i) = e^{-w_i\mu_i}\frac{(w_i\mu_i)^{w_iy_i}}{w_iy_i!}$$

but in practice, the homogeneity within a tariff cell is hard to achieve. This can be modeled by letting the risk parameter μ_i be itself the realization of a random variable and thus building a mixed-Poisson distribution. The following two figures give more information about the convenient distributions that fit the number of claims and the claim frequency respectively in our train data set:





Building our models we will add an offset equal to the $log(w_i)$ because as we want to predict the **claim frequency key ratio** managing the exposure, then $\mathcal{L}(\lambda)$ cannot be applied directly as $\theta_i = ln(w_i\lambda) = ln(w_i) + ln(\lambda) = ln(w_i) + \theta$ with $\theta = ln(\lambda)$ giving the **expected number of claims over one period**. Moreover, the model become $Y_i = \frac{N_i}{w_i} \Longrightarrow log(Y_i) = log(N_i) - log(w_i) = \sum_{i=1}^{r} x_{i,j}\beta_j - log(w_i)$.

Focusing on the **NbClaims** feature we can already assume for a trade-off between the negative binomial and the Poisson distributions. In fact, when data displays over-dispersion, the most likely to be used distribution is the negative binomial distribution instead of the Poisson. In fact, the Negative binomial regression model allows the conditional variance of y_i to exceed the conditional mean. This has to be checked:

One important characteristic of the Poisson family is that $\mathbb{E}(y_i) = \mathbb{V}(y_i) = \lambda_i = w_i\mu_i$ and for the **NbClaims** we get **0.033** and **0.0336** respectively. Even if that's quite good results we will perform supplementary test for over/under-inflation as we know that omitting relevant ratemaking variables induces dispersion. For this test we will already introduce the GLM theory because the *dispersiontest* function in R test for equi-dispersion against the alternative that the variance is of the form:

$$\mathbb{V}(Y) = \mu + \alpha f(\mu)$$

where over-dispersion corresponds to $\alpha > 0$ and under-dispersion $\boldsymbol{\alpha} < 0$. The rule of thumb for this test is that the ratio of deviance to df should be **1**, for the **NbClaims** the value of this ratio is **1.005** and thus, with a p-value of **0.213** we can not reject the null hypothesis of equi-dispersion.
And so, we will try to build GLM's with logarithmic link function. This guarantees that the mean stays between 0 and 1. For the Poisson GLM we get:

$$log(\lambda_i) = log(w_i\mu_i) = \sum_{i=1}^{r} x_{i,j}\beta_j$$

The vector of β is estimated by log-likelihood maximization based on a sample of n observations, each individuals following an EDM distribution.
Another assumption that has to be checked is the **zero inflation** that can occur in count features. Just looking at the statistics previously displayed it seems to appear for the **NbClaims** feature for the **0** level.
Zero-inflated count models are two-component mixture models combining a point mass at zero with a proper count distribution. Thus, there are two sources of zeros: zeros may come from both the point mass and from the count component. Zero-inflated models estimate two equations simultaneously, one for the count model and one for the excess zeros. We will use a score test such that it calculates the rate estimate from the mean $\hat{\lambda} = \bar{x}$ and then the number of observed zeros n_0 with the expected number, $n\hat{p}_0$ where $\hat{p}_0 = \exp(-\hat{\lambda})$. Finally the test statistic is:

$$\frac{(n_0 - n\hat{p}_0)^2}{n\hat{p}_0(1-\hat{p}_0) - n\bar{x}\hat{p}_0^2} \sim \chi_1^2 = \textbf{13.89}$$

which corresponds to a p-value of **0.0002** and thus we reject the null hypothesis of the proportion test.
But let's point out that as p_0 depends on regression parameters and varies across observations, this only works in the non-regression settings. A growing concern about comparing discrete nested and non-nested models is the Vuong test but it has been demonstrated that it's not applicable in the special case of zero-inflation, **Wilson, 2015**.

3 Models comparison

For the further analysis we will split our **DBtrain** dataset into a train set which is **80**% of the original data and a **20**% remaining validation set. The criterion of quality of a regression model with non-normally distributed response should be the deviance: in the Poisson case, we can show that the deviance can be written as:

$$D = 2\sum_i w_i(y_i log(y_i) - y_i log(\hat{\mu}_i) - y_i + \hat{\mu}_i)$$

and the smaller deviance indicates the better goodness of fit. We will fit the models on the training set, and use the models to predict claim frequencies for the validation set. Moreover, we will split the test set into groups, and use the predictions to asses if the model is well calibrated. In addition, a common approach in non-life insurance is to cut continuous variables into intervals and thereby transforming them into a few binary variables. This method is useful to detect nonlinear effects of such variables, which would not be possible otherwise as GLM models linear effects. We cut the **DriverAge** and **CarAge** features into 4 intervals, with the quartiles as breaking points:

- **DriverAge**: **Q1** : $[18; 40]$, **Q2** : $(40; 50]$, **Q3** : $(50; 60]$, **Q4** : $(60; 82]$
- **CarAge**: **Q1** : $[0; 3]$, **Q2** : $(3; 5]$, **Q3** : $(5; 7]$, **Q4** : $(7; 16]$

Moreover as we are working under Poisson assumptions of independence and thus Poisson distributed response, grouping our portfolio in risk classes is allowed since the $\mathcal{L}_{ind}(\beta) \propto \mathcal{L}_{group}(\beta)$ as it only involves the total exposure and the total claim numbers and thus aggregating the whole portfolio in **homogeneous** risk classes. Let's recall that this assumption is only valid with ED responses.

3.1 GLM and GAM

3.1.1 Negative binomial regression

This kind of model can be build when dealing with over-dispersed count response feature. Even if we already perform a test for the dispersion of our target feature that doesn't allow us to reject the null hypothesis of equi-dispersion, we will try to fit such a model to assess for its performances. It can be considered as a generalization of Poisson regression since it has the same mean structure as Poisson regression but it has an extra parameter to model the over-dispersion. The impact of this dispersion rate is that the confidence intervals are likely to be wider. More specifically, it can be viewed as a Poisson distribution with parameter λ not fixed but a random variable which follows a Gamma distribution. And so the distribution of such a feature is:

$$P(Y = y_i|\mu_i, \alpha) = \frac{\Gamma(y_i + \alpha^{-1})}{\Gamma(y_i + 1)\Gamma(\alpha^{-1})}(\frac{\alpha^{-1}}{\alpha^{-1} + \mu_i})^{\alpha^{-1}}(\frac{\mu_i}{\alpha^{-1} + \mu_i})^{y_i}$$

where $\mu_i = w_i\mu$ and $\alpha = \frac{1}{\theta}$, θ being a scale parameter.
And as for the parametrization used in negative binomial regression we get:

$$\mathbb{V}(Y) = \mu + \frac{\mu^2}{\theta}$$

and so θ controls the excess variability compared to Poisson regression (μ in this case), and is known as a dispersion parameter. As we perform the negative binomial regression on our data we obtain a $\theta =$ **15.9** after few iterations. That's why building a Poisson regression seems to be convenient. Moreover, when the success probability is close to **0**, that's the case in our dataset, the output of the Poisson regression is easier for interpretation.

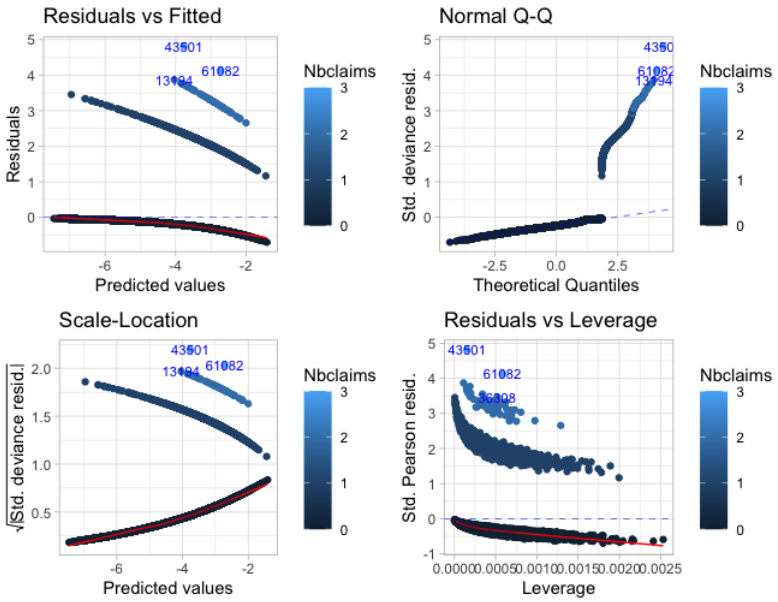
3.1.2 GLM Poisson regression

We have already defined the main assumptions of the Poisson GLM in the previous section but as it's time to compare model performances, we have to go deeper in the way the Poisson regression is working. First let's denote that is we use *goodfit* in R we can assess for the Poisson estimation power. In fact, as the λ estimated with for our count feature of the number of claims is equal to **0.033**, we got the following fitted count distribution:

Nbclaims, $\lambda = 0.033$

	observed	fitted
0	67752	67730
1	2190	2233
2	56	37
3	2	0

But even if the count distribution seems to be correctly fitted, the interest of a GLM with a Poisson distribution is to explain the **random component** of our model with the Poisson distribution but it's crucial that the the explanatory variables of the regression as a combination of linear predictors that correctly explain the **systematic component**. As you will find the regression coefficient estimates attached in the appendices, we can already compute the base $\lambda_0 = exp(\beta_0) =$ **0.035** for a reference profile in the portfolio, the λ_i is then found by multiplying the base by the corresponding estimated relativities $exp(\beta_j)$. In fact, looking at the previously displayed QQ-plot it's quite obvious that observations with non-null number of claims aren't well fitted by the model.
With the following 4 plots we can assess for the influence of the **zero-inflation** on the Poisson regression output:



In fact, as the most common choice is Pearson residuals which are asymptotically normal, we get:

$$r_{P_i} = \frac{y_i - \hat{\mu}_i}{\sqrt{v(\hat{\mu}_i)/w_i)}}$$

And thus, the normal QQ-plot helps us to find outliers-observations that are not well fitted. The colours help us to determine that only 0-level observation are fitted by the model (see red line). Moreover, the *step* R function retain the **Gender, Area, Leasing, Power, Fract and Fuel** features for our model.
Under this clearly bad fit for our count target, and dispersion between groups as defined (see appendices), we could also use quasi-poisson to get more correct standard errors with rate data.

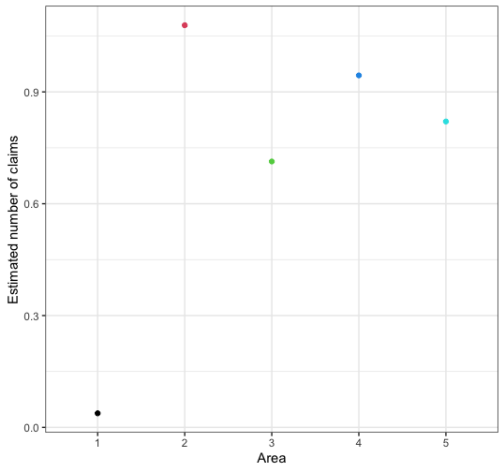
3.1.3 GLMM Poisson regression

As homogeneity within a tariff cell is hard to achieve, this is a model that take over-dispersion into account. This is done by assuming a supplementary structure by trying to explain more precisely the random effects, letting the risk parameter μ_i be itself the realization of a random variable. In fact, the relationships between the feature of interest may differ according to grouping factors, assuming that not all policyholders in the portfolio have an identical expected claim frequency. As we can see in the appendices looking at the correlation matrix, we

can assume independence between features but we have to check the assumption that within a grouping factor, the relationship between the target feature and explanatory features stay the same. More specifically as we have random effect covariates $z_{i,t}$, $t = 1, ..., n_i$ with i.i.d γ_i random effects such that $Y_{it}|\gamma_i$ are independent under a EDM distribution, then:

$$log(\lambda_i) = log(w_i\mu_i) = \sum_{i=1}^{r} x_i\beta_i + z_i\gamma_i$$

As the **Area** feature seems to be a good discriminant feature we will perform the **Poisson-LogNormal** model for claim counts, trying to estimate the number of claims across this feature, allowing the target to vary between the different area's. Thus, we can plot the estimated deviation between each area average number of claims and the overall average since we know that in this case, $\mathbb{E}(e^{\epsilon_i}) = e^{\boldsymbol{z_{i,t}\gamma_i}} = e^{\boldsymbol{\sigma_\epsilon^2}}/2$ and $\mathbb{E}(N_{it}) = \lambda_{it}e^{\boldsymbol{\sigma_\epsilon^2}}/2$



And as each level of the **Area** feature contains at least **7000** observations we can say that this model can explain a part of the **0** level of **Nbclaims**. Finally, let's denote that if panel data's are available, then the static random effects $\boldsymbol{\epsilon_i}$ can be replaced by dynamic ones $\boldsymbol{\epsilon_i 1, ..., \epsilon_i t}$ obeying a **Gaussian** with **AR1** covariance memory effect process. But as a signal $\boldsymbol{S_{it}}$ isn't at our disposal about the policyholder's behavior it will difficult to run such a model. For information, in *R* such models are available in *lme4ord* or in *glmmTMB* packages.

3.1.4 ZIP regression

This kind of model is called Zero-inflated Poisson because it tries to understand why we can encounter an excessive amount of 0 level observation as a Poisson regression is fitted. It's a mixed model that first generates 0 counts, with as second step of the process generates counts under Poisson assumptions.
And thus we get the two probability mass functions:

$$P(Y = 0) = \pi + (1 - \pi)e^{-\lambda}$$
$$P(Y = y_i) = (1 - \pi)\frac{\lambda^{y_i}e^{-\lambda}}{y_i!}$$

for $y_i = 1, 2, 3..., \pi$ being the probability of extra zeros.
And once again we use the GLM theory to explain the link between the regression explanatory features and the λ parameter as:

$$log(\lambda_i) = log(w_i\mu_i) = \sum_{i=1}^{r} x_{i,j}\beta_j$$

Moreover, we already reject the hypothesis of zero inflation after performing a score test previously but for information, we used the R command *zeroinfl* from the package *vcdExtra* to build this model. In fact, as output we

can clearly see the way R used to fit two-part mixed effects models, the count model coefficients estimated by a Poisson regression with log link function and Zero-inflation model coefficients with a binomial, again log linked. In fact, it's the more realistic way of estimating π_i as a function of the explanatory variables. Once it's done, we plug the π-vector into the probability functions of the ZIP model and use what is known as the Maximum Likelihood Estimation technique to train the ZIP model on the data set with excess counts.

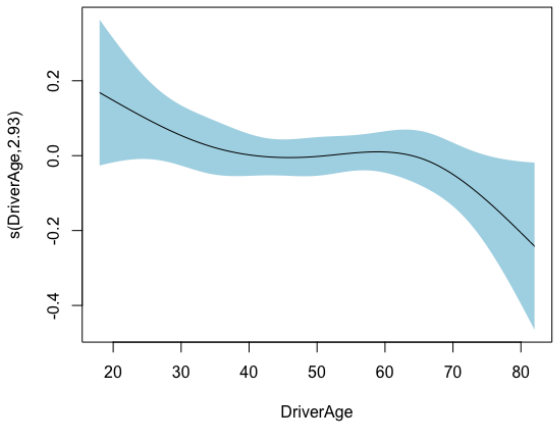
3.1.5 GAM

As we are still dealing with EDM, we will consider the Poisson distribution to fit the number of claims distribution but in the case of **general additive models** linear, target feature depends on unknown smooth functions $f_{j=1,...,J}$ of some explanatory features, and interest focuses on inference about these smooth functions. Thus the link function is used as:

$$g(\mathbb{E}(Y_i)) = g(\mu_i) = \eta_i = \beta_0 + f_1(x_{i,1}) + ... + f_J(x_{i,J})$$

And thus, here the goal is to find the smoothing functions that offer the best description of the effect of the continuous features. Moreover, the disadvantage of using GLM is the categorization of the continuous variables: two policies with close values that would be classified differently can have very different premiums. Thus, in GAM's the continuous features which have been discretized in the GLM remain continuous. That's what we will explore for the **DriverAge** and **CarAge** features by adding separate penalty $(\lambda_{DriverAge}, \lambda_{CarAGe})$ to the deviance, guaranteeing that the smoothing functions have not a too high variability and by minimizing this deviance through **cubic splines**. We obtain the following results:

```
Approximate significance of smooth terms:
               edf Ref.df Chi.sq p-value
s(DriverAge) 1.002  1.004  7.061 0.00793 **
s(CarAge)    1.017  1.033  0.467 0.50442
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



Where the tuning parameter has been determined by generalized cross validation considering a scaled in-sample error with the criterion $GCV(\lambda) = (1 - \frac{M(\lambda)}{n})^{-2} D(N, f)$, $M(.)$ being the effective degree of freedom and $D(.)$ the deviance. As the ANOVA test the overall significance of the smooth and as it's only significative for the **DriverAge**, we only display this feature. In fact, the confidence intervals help us to assess that we can't draw an horizontal line between them. Moreover, the un-biaised risk estimator is such that $UBRE(\lambda) = \frac{2\lambda M(\lambda)}{n-\lambda} + \frac{D(N,f)}{n}$ = **-0.79** in our case. Performing the *gam.check* R function we can assess for the good choice of the cubic splines with $m =$ **7** subintervals to estimate the smooth function non-parametrically, because model residuals are still considered as random.

3.1.6 Comparison of performances

As we try to compare the models we performed, we will compare several metrics that have to be defined. First, the Akaike information criterion is computed as $AIC = 2k - 2ln(\hat{L})$, $\hat{L}$ being the maximized value of the likelihood

function for the model and the Bayesian information criterion is computed as $BIC = ln(n)k - 2ln(\hat{L})$. Moreover, as in the Poisson case, the deviance is $2\sum_i w_i(y_i log(y_i) - y_i log(\hat{\mu}_i) - y_i + \hat{\mu}_i)$ we recode it in *R*.

GLM's, ZIP and GAM performances						
Dataset	Train			Validation		
	Deviance	AIC	BIC	Deviance	AIC	BIC
NB GLM	11 451	19 286	19 506	2 985	4 000	4 181
Poisson GLM	11 687	15 301	15 506	2 988	3 998	4 172
Poisson GLMM	15 272	15 312	15 490	3 966	4 006	4 157
ZIP	-	15 480	15 819	-	4 053	4 400
GAM	11 804	15 747	15 917	2 972	3950	4086

The poor performances of our previously displayed models my be due to the particular form of the correlation matrix. In fact, as the only risk factor that's correlated with the Number of claims is the **Exposure** we may consider an alternative modeling, basing the claim frequency analysis on the times $T_{i,1}, ...T_{i,N_i}$ at which these N_i claims occured. Then we denote $W_i = T_{i,k} - T_{i,k-1}$, $k = 1, 2, ...$ with $T_{i,0} = 0$ as the waiting periods between two consecutive claims for policy i. Then if $N_i = 0$ it means that $T_{i,1} = W_{i,1} > w_i$ and $P(W_{i,1} > w_i) = e^{-w_i\lambda_i}$ and by generalisation, the likelihood associated to the occurence times of the N_i claims is of the form:

$$\mathcal{L} = \prod_{i=1}^{n} (\lambda_i^{N_i} e^{-\lambda_i w_i})$$

where $N_i = max\{k | T_{i,0} + T_{i,1} + .. + T_{i,k} \leq w_i\}$ and thus Gamma regression on the waiting period W_{ik} is equivalent to Poisson regression on the number of claims. Then we can write the following expression for the **true score** of GLM's:

$$score_i = \boldsymbol{\beta}^\intercal x_i + \boldsymbol{\gamma}^\intercal x_i^+ = \boldsymbol{\beta}^\intercal x_i + \boldsymbol{\epsilon_i}$$

where x_i^+ is the **missing part of risk factors**.

3.2 Binary Poisson Regression trees

As in our dataset the $i - th$ individual is represented by (N_i, X_i, w_i), respectively the number of claims, the features and the exposure of observation i, we will build a **Binary Poisson regression tree**, trying to find a structural form in a non-parametric way. Let f be some impurity function and define the impurity of a Node A of our regression tree as:

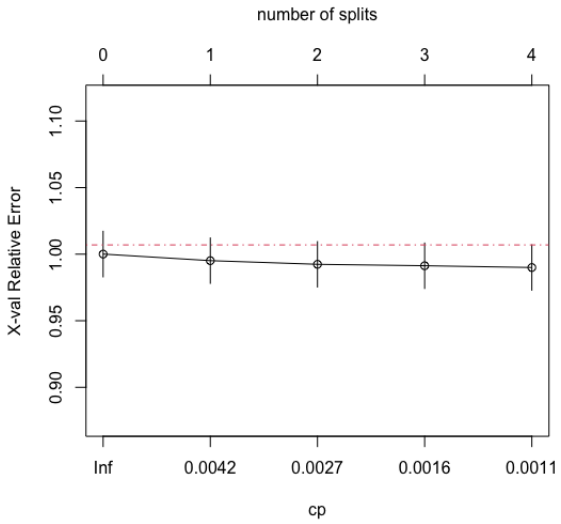
$$I(A) = \sum_{i=1}^{C} f(p_{iA})$$

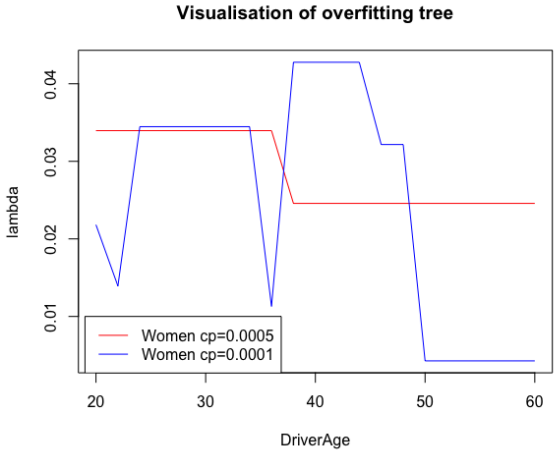
where p_{iA} is the proportion of observations in A that belong to class i for future samples. In R, the cost-complexity measure is rewritten as $R_\alpha(T) = D_\tau(N, \bar{\lambda}_t) + \alpha|\tau| = D_\tau(N, \bar{\lambda}_t) + cpD_\tau(N, \bar{\lambda}_t)|\tau|$ where cp is the cost-complexity parameter that allows to tune the depth of the tree. As we are working under Poisson distribution assumptions for the number of claims, what the **goodness of split** concern we will deal with **deviance statistics**, $D_\tau(N, \bar{\lambda}_t) = \sum_{i:x_i \in \chi_{t\tau}} 2N_i(\frac{\bar{\lambda}_{t\tau} w_i}{N_i} - 1 - log(\frac{\bar{\lambda}_{t\tau} w_i}{N_i}))$ for the leaf τ and binary split $t \subset \tau$ as split criteria to minimize. When the number of claims is **0** then the deviance for the individual i become $2\bar{\lambda}_{t\tau} w_i$. Using the *rpart* *R* function and in order to avoid 0 estimators we compute the following Bayesian estimator for $\lambda_{t\tau}$:

$$\bar{\lambda}_{t\tau}^B = \hat{\alpha}_{t\tau}^B \bar{\lambda}_{t\tau} + (1 - \hat{\alpha}_{t\tau}^B)\bar{\lambda}_0$$

where $\bar{\lambda}_0$ is the prior estimator (*R* take the global mean of layers) and choosing $\chi_{t\tau} \subset \chi$ such that $N_{t\tau} = \sum_{i:x_i \in \chi_{t\tau}} N_i | \Theta \sim Poi(\lambda_{t\tau} \Theta w_{t\tau})$ with $\Theta \sim \Gamma(\gamma, \gamma)$. Moreover, as the $\bar{\lambda}_{t\tau} = \frac{\sum_{i:x_i \in \chi_{t\tau}} N_i}{\sum_{i:x_i \in \chi_{t\tau}} w_i}$ and $\bar{\alpha}_{t\tau} = \frac{w_{t\tau}\bar{\lambda}_{t\tau}^B}{\gamma + w_{t\tau}\bar{\lambda}_{t\tau}^B}$, then good risks $\lambda_{t\tau} < \bar{\lambda}_{t\tau}^B$ obtain a higher credibility weight.

Choosing our cost-complexity threshold we can compare regression tree models and interpret them. In fact looking at the calibration by K-fold cross-validation looking at the **x.error(cp)** based on the deviance at each level of **cp** and the estimate of the standard deviation of this error **x.std(cp)**. We display our model results:





As the **x.error(cp)** become lower when the number of split become bigger and stay stable after **4** splits for the tree with a $cp = \mathbf{0.001}$ it seems to be convenient for interpretations and doesn't over-fit our data instead of a tree with $cp = \mathbf{0.0001}$. Before our tree interpretation (see appendices) we recall that we use a shrink parameter $\gamma = \mathbf{1}$ and a $\mathbf{10}$ K-fold cross validation. These values have been chosen because even by choosing other cp values, we doesn't reduce the **root node error** significantly, approximately equal to **0.213** so doesn't retain the tree with $cp = \mathbf{0.0005}$. So, looking at our retained regression tree (see appendices) we can assess that country side low altitude and island areas seems to indicate lower claim number, which seems to be confirmed looking at the map of European road fatalities in appendices. Moreover our RT says that leased cars are more likely to be involved in accidents that seems logical as several standard private leasing contracts covers various taxes, omnium insurance and wear parts and unforeseen costs. The impact of the split of the contract will be explained in the next section with PDP's. But paradoxically our tree has weird values. In fact, a study done by the American Automobile Association shows that drivers under 25 are actually $\mathbf{188}\%$ likelier than adult drivers to cause an accident, this discrimination factor isn't taken in consideration. In addition, the gender discrimination isn't applied in our tree.

3.3 Gradient Boosting Machine

3.3.1 Gradient Boosting with Poisson Deviance Trees

Gradient Boosting is a sequential method that combines sequentially weak predictive models into a more powerful one, typically solved by gradient descent:

$$f_{GB}(x) = argmin_f \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i))$$

where L is a penalty function as a function of deviance. Moreover, as the frequency of N_i is low in our case, the Poisson distribution cannot be approached by a Normal law by CLT. Thus, an iterative algorithm to locally improve a current approximation is performed by minimizing the in-sample loss. In fact, we perturb f_m, at iteration m, so that it leads to a maximal decrease in this loss, choosing a small step of size $\rho > 0$ so that:

$$f_{m+1} \longrightarrow f_m - \rho \Delta \mathcal{L}_L(f_m)$$

Practically we can avoid to regress gradients on features when the loss function is the Poisson deviance, assuming $log(\lambda(x)) = f(x)$. Then the Poisson deviance become:

$$L(N, \lambda(x), w) = 2w[e^{f(x)} - \frac{N}{w} - \frac{N}{w}f(x) + \frac{N}{w}log(\frac{N}{w})]$$

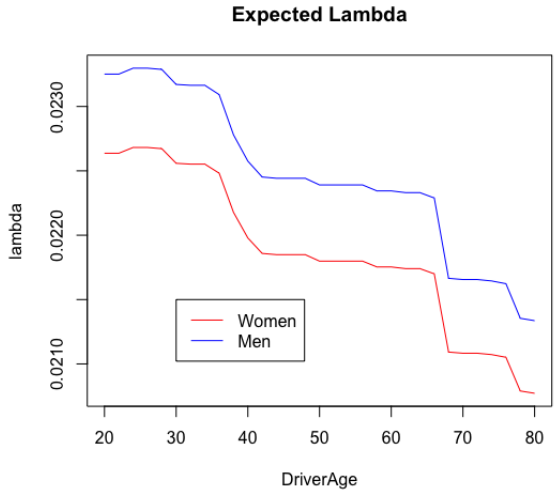
updating the working response at each iteration $R_{m,i} = 2(N_i - \hat{\lambda}_{m-1}(x_i)w_i)$ with $\hat{\lambda}_{m-1}(x) = e^{(\hat{f}_{m_1}(x))}$. This is called the **Boosting** and in this case, we add a basis function:

$$\hat{g}_m = argmin_{g_m} \sum_{i=1}^{n} L(N_i, \mu_m(x_i) = e^{g_m(x_i)}, w_i^{(m)})$$

to $\hat{f}_{m-1}$ and thus we obtain a new Poisson regression expression for the expected frequency with exposure. In fact, we will choose a depth of the tree J and a learning rate $v \in (0, 1]$, setting $\hat{f}_0(x) = log(\frac{\sum_{i=1}^{n} N_i}{\sum_{i=1}^{n} w_i})$ constructing a Poisson regression tree of depth J such that $\hat{\mu}_m(x) = \sum_{t \in \tau^{(m)}} \bar{\mu}_t^{(m)} 1_{x \in \chi_t^{(m)}}$ updating the function $\hat{f}_m(x) = \hat{f}_{m-1}(x) + \sum_{t \in \tau^{(m)}} log(\bar{\mu}_t^{(m)}) 1_{x \in \chi_t^{(m)}}$ and thus finally giving the expected frequency estimator:

$$x \longrightarrow \hat{\lambda}(x) = e^{(\hat{f}_M(x))}$$

So in our case, as we want to perform this algorithm on our count target of the number of claims, as hyperparameters, we use binary decision trees as sequential models, with maximum depth $J = \mathbf{4}$. So, the function *gbm.perf* computes the iteration estimate with smaller values of the shrinkage parameter that offer improved predictive performance, but with decreasing marginal improvement. Thus, we set the learning rate $v = \mathbf{0.001}$, a maximum number of trees $M = \mathbf{7000}$ and with K-folds cross validation parameter $K = \mathbf{5}$. Concerning the optimization method, we specify that the target is Poisson or Tweedie with exponent parameter $p = \mathbf{1}$ distributed.



Here we clearly observe the well known **U-shape** of the impact of **DriverAge** on the expected number of claims in the motor insurance. As expected, looking at the intereaction of **DriverAge** and **Genre** we can conclude that men drivers are more dangerous compared to women. Moreover, looking at the marginal **barplots** (see appendices) we clearly recognize the assumption that diesel vehicles report more claims, due to its low price compared to other fuel and thus pushes you to drive more kilometres. Interpretation of other categorical features may be more tricky because it doesn't match the expected assumptions that **female** drivers tend to report more claims or that the number of claims increases with the power of the car. That's why cross features analyze or **PDP**'s plots are usefull. In fact looking at our partial dependence plots (see appendices) we clearly denote that the mean claim frequency tends to increase with the **power**. Moreover, looking at the type of liability insurance it suggests higher claim frequencies for drivers with more guarantees, maybe due to adverse selection. As we look at the **Fract** we denote that mean claim frequency is higher when the premium payment is split accross the yearn, usually

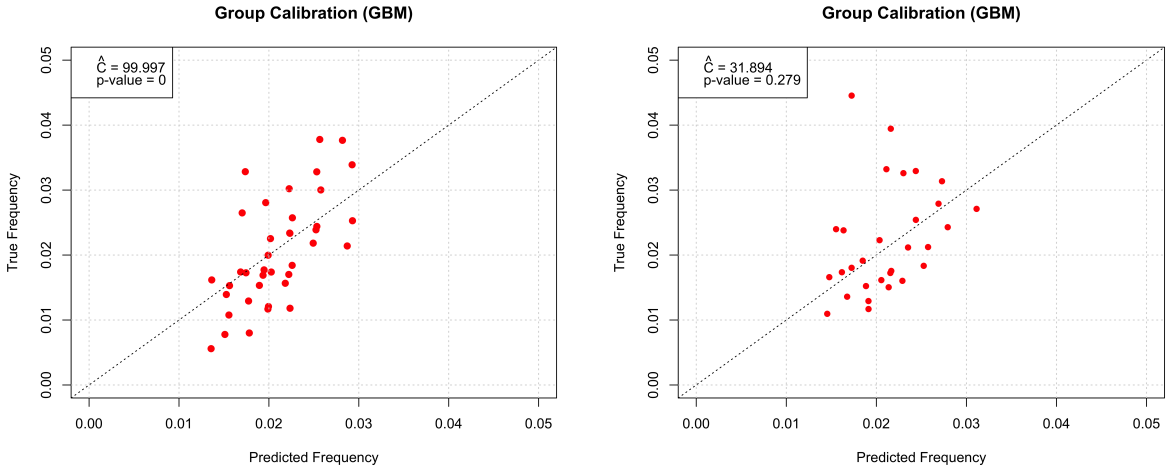
attributed to a lower socio-economic profile when splitting premium payment results in an increase of the total amount of premium paid.

3.3.2 Calibration for relevant subgroups

We will split the train and test set into groups, and use the predictions to asses if the model is well calibrated. To do so, we will use the **Chi-Squared** test that is $\boldsymbol{H_0}$: identical observed and expected frequencies given by the test statisitc:

$$\hat{C} = \sum_{g=1}^{G} \frac{(O_g - E_g)^2}{E_g} = \sum_{g=1}^{G} \frac{(\lambda_g \times n_g - \hat{\lambda}_g \times n_g)^2}{\hat{\lambda}_g \times n_g}$$

This statistic is computed through groups discriminated by the levels of the following features **Area, Gender** and **Power** that gives a total number of groups $\boldsymbol{G = 40}$ displayed on the next plots.



3.4 Neural networks

3.4.1 Neural net comparisons

In this section we will build neural network models with one and two hidden layers. While it's less interpretable than GLMs, it does have the universal approximation property that allows for non-linear function estimations (which is not possible for GLM's). In fact, according to the universal approximation theorem, a single hidden-layer neural network can approximate any measurable function arbitrarily well, provided the number K of hidden units is sufficiently large. As for our GLM construction, for the NN preprocessing categorical variables are modeled as dummy variables and as we assume that the **number of claims** is Poisson distributed, we transform the output signal of the neural network such that:

$$\hat{y}_i = \hat{\lambda}_i = g(y_{i,n^{net}}) = exp(y_{i,n^{net}}) = exp(\phi_0(\nu + \sum_{k=1}^{K} \omega_k \phi_h(\nu_k + \sum_{i \to k} \omega_{ik} x_i)))$$

where ω_{ik} are weights that connect the feature vector $\boldsymbol{x_i}$ to the k-th hidden neuron, $\phi(.)$ are activation functions. Common issues with neural network models are the selection of the hyperparameters of the architecture (number of hidden neurons, activation function,...) and overfitting.
Even if it is very hard to attain the global minimum of the loss and therefore to get the theoretically zero coefficients, we get the loss function (with Ω the set of parameters):

$$R(\Omega) = \frac{1}{n}\sum_{i=1}^{n} D(y_i, \hat{y}_i^{net})$$

And as the optimal weights are found by minimizing $R(\Omega)$ such that $\Omega = argmin_\Omega \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(y_i, \hat{y}_i)$, adjusting the vector of weights Ω_t by a small step in the opposite direction of the gradient, and working under Poisson deviance function we obtain the following results:

Poisson neural networks performances					
	Train			Validation	Weights #
	Deviance	AIC	BIC	Deviance	
NN(K_1=3)	11 564	15 297	15 788	3 075	55
NN(K_1=4)	11 554	15 299	15 951	3 062	73
NN(K_1=5)	11 456	15 237	16 050	3 101	91
NN(K_1=2, K_2=2)	11 646	15 331	15 715	3 050	43
NN(K_1=3, K_2=3)	11 527	15 261	15 859	3 105	67

We choose **3000** maximum steps for the training of the neural networks with a threshold for the partial derivatives of the error function of **1**. If $m = card\{\Omega\}$ is the number of weights, then we can compute the **AIC** = $2m - 2\hat{l}(\hat{\boldsymbol{y}}, \boldsymbol{\phi}, \boldsymbol{y})$ and given a set of networks, the preferred model is the one with the lowest AIC.
Building a neural network with only **1** layer, **5 ReLu** activation neurons, **Poisson deviance** loss function and **resilient back-propagation** algorithm with momentum, we will train our **train set** under *keras* R cross Python package. We will add a dropout layer with the fraction of the input units to drop that consists in randomly setting a fraction rate (**rate = 0.3** choosen) of input units to 0 at each update during training time, which helps prevent overfitting. Moreover, we choose a batchsize of **20000** as the larger the batch, the better the approximation and samples in a batch are processed independently, in parallel.

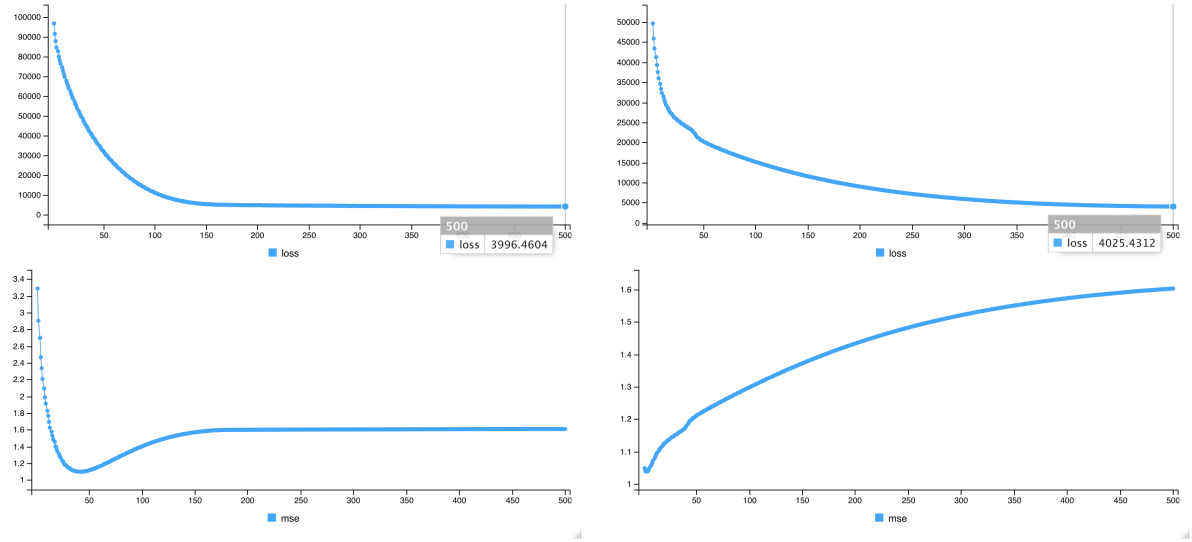


Figure 5: NN($K_1 = 5$ ReLu) & NN($K_1 = 5$, $K_2 = 5$ ReLu, $K_3 = 10$ SoftMax)

3.4.2 Calibration for relevant subgroups

The same approach has been chosen for the calibration of our **NN** than for our Gradient Boosting Machine. Although it seems to be better calibrated than our GBM looking at the alignment of points, we get a bad Chi-squared statistics. This is due to the fact that it doesn't predict very well heavy or risky groups with higher claim frequencies. Other NN calibrations have been displayed in the appendices.

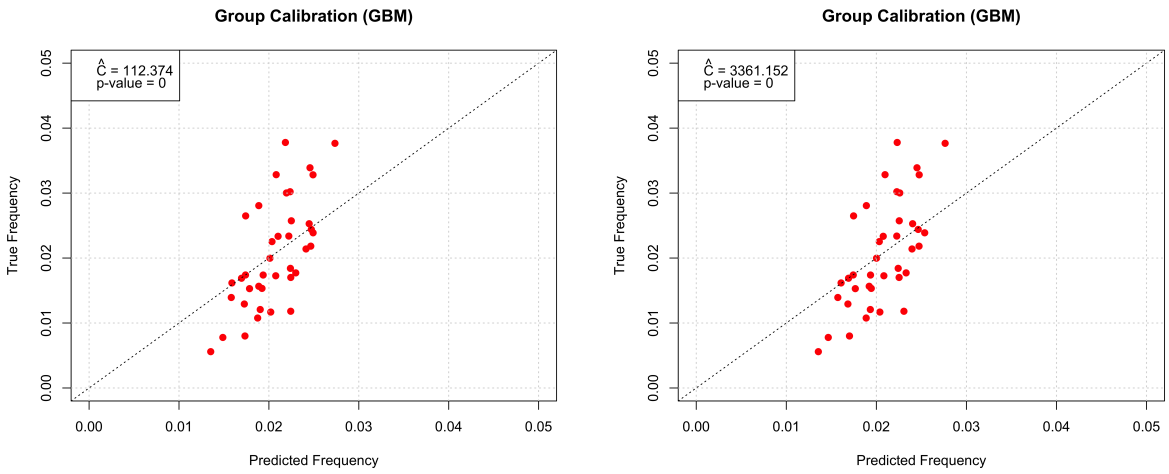


Figure 6: Group calib. for NN with 5 ReLu on 1 layer

3.5 Global & Local interpretation

As we have already used the **PDP's** to evaluate the influence of features separetely on the outcome, by averaging predictions with feature values at given features of interest values, we will implement other interpretation models. Let's point out that we need features independence but this has been checked previously. In fact as we become more experienced with our data set, it is time to conclude and refine our analysis about the features involved. In the appendices, section **Global & Local interpretation** we display the **3D partial dependence plots** that seems to be useful for interpretation. We display on each plot a different feature in interaction with the **DriverAge** for which the partial dependence function is computed, assuming other features as random. The most interesting is the PDP of **Area** cross **DriverAge**. In fact, we clearly understand that for drivers aged between **30** and **60** we distinguish two plateau's of same influence for the levels of **Area**; **(1,2)** and **(3,4,5)** with opposite partial dependence of respsectively **0.10** and **-0.10** on the target claim frequency that make sense with our previous binary trees conclusions. Moreover as the the slope between the two plateau's only display the difference between the lower and higher levels of **Area**, once again we recognize the well know **U-shape** of **DriverAge** but with different starting and ending points for the two plateau's. We can clearly conclude for the importance of the **Area** feature pointing out that for lower levels the **PD** is positive till the **DriverAge** become higher than **60**. What the **CarAge** concern, looking at this feature cross the **DriverAge** we will point out that the **PD** of car aged of **9** years are detached slightly from the plateau's with a **PD** higher than **0**. What the **Power** concern we clearly denote that the partial dependence become higher as the power of the car increase, becoming higher than **0** for **high horsepowered** cars. But cross the **DriverAge** this relationship is not as smooth as we have seen before, as a plateau take place for **normal horsepower** and **intermediate horsepower** levels. We will not give more information about the **Leasing** and **Gender** features **PD** as it easy to understand that **women** and **leased** cars are linked to lower partial dependence's than the other levels, what we already concludes previously with lower claim frequencies attributed. Moreover, no additional interpretation can be made about the **Contract** feature, which doesn't seem to be useful for our analysis, looking at the **Relative influence plot**.

However, unlike partial dependence plots, which show the average effect of the features of interest, **ICE plots** visualize the dependence of the prediction on a feature for each sample separately, with one line per sample. You will find the results attached in the appendices.

What local analysis concern, we will implement **Local Interpretable Model-Agnostic Explanations** and **Shapley values**. As LIME explanations are the result of a random sampling process, performed with random forest for this project, we shouldn't expect to get the exact same explanation every time. As we want more confidence in the explanation, we increase the number of samples instead of running it multiple times and taking an average. In order to attain this LIME minimizes the following :

$$g(x) = argmin_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

where f is our regression tree model, $\pi_x(z)$ is the proximity measure in the neighborhood of x, $\mathcal{L}$ is the loss-like fidelity term which is the weigthed sum of squared errors and $\Omega(g)$ is a complexity term that is proportional to our **cp** criterion. That's what we obtain running this litlle algorithm on individuals with **100 000** samples as size of the neighborhood chosen. Moreover, reducing the kernel width (set to **0.6**) will make decrease the divergence between the local prediction and original prediction. But LIME does not guarantee that the prediction is fairly distributed among the features.

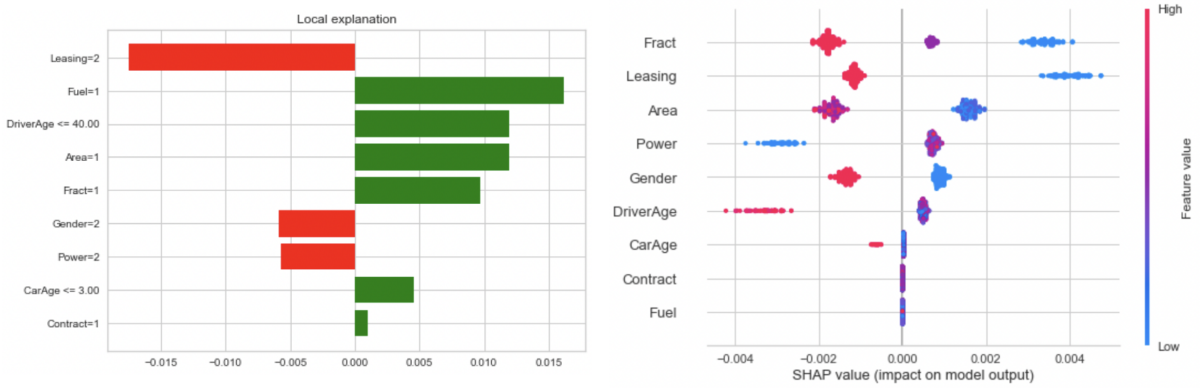


Figure 7: LIME for the individual of row 25 (RT) and Shapley summary with GBM

The **Shapley values** will help us determine how much the features of the model contribute to the difference between a value of the response and the average response value. This method link optimal credit allocation with local explanations. This little algorithm will help us to avoid undesirable effects because thanks to it, features changes simultaneously and even if we can get improbable combinations we are keeping batches of features together, mitigating this effect.In fact, this is the only attribution method that satisifes the properties of **Efficiency, Symmetry, Dummy and Additivity**. Therefore we will summarize the impact of each feature on the claim frequency under **gradient boosting method**.

3.6 Conclusion and choice of the best model

As it is time to finish our journey through the different actuarial methods we will select the best model for our individual claim frequency predictions. We point out that we use the regression approach for this project because as classification aims to predict internal features of a category, for our predictions two identical contracts may generate a different number of accidents and therefore do not have the same intrinsic features.
Let's check our results summary:

Retained models for each implementation			
	Train Deviance	Validation Deviance	$\mathbb{E}(\lambda \mid X_{valid})$
Poisson GLM	11 687	2 988	0.03286
GAM	11 804	2 972	0.03312
GBM	11 693	3 032	0.02121
NN(K_1=5)	11 456	3 101	0.02097

And as the expectation of the claim frequency in our data set is of **0.02179**, we report for each method the predicted mean claim frequency over the validation set, and the train/validation deviance. Both our machine learning implementations seems to perform good predictions. Looking at the calibration chi-squared statistics, we will choose the **Gradient Boosting with Poisson Deviance Trees** implementation for our predictions.

References

Denuit, M., Hainaut, D., & Trufin, J. (2019a). *Effective Statistical Learning Methods for Actuaries I: GLM and Extensions*. Springer. doi: 10.1007/978-3-030-25827-6

Denuit, M., Hainaut, D., & Trufin, J. (2019b). *Effective Statistical Learning Methods for Actuaries III: Neural Networks and Extensions*. Springer. doi: 10.1007/978-3-030-25827-6

Denuit, M., Hainaut, D., & Trufin, J. (2019c). *Effective Statistical Learning Methods for Actuaries II: Tree-Based Methods and Extensions*. Springer. doi: 10.1007/978-3-030-25827-6

Directorate-General for Mobility, & Transport. (2022). *Road safety in the eu: fatalities in 2021 remain well below pre-pandemic level*. Retrieved from https://transport.ec.europa.eu/news/preliminary-2021-eu-road-safety-statistics-2022-03-28_en

Frees, E. J. (2010). Regression modeling with actuarial and financial applications. *Cambridge University Press*. Retrieved from http://research.bus.wisc.edu/RegActuaries

Hainaut, D. (2021). *LDAT2230 : Data sciences in insurance and finance slides.* University Lecture, https://moodle.uclouvain.be/course/view.php?id=3305.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction* (Second ed.). Springer. Retrieved from http://www-stat.stanford.edu/~tibs/ElemStatLearn/

Omari, C. O., Nyambura, S. G., & Mwangi, J. M. W. (2018). Modeling the frequency and severity of auto insurance claims using statistical distributions. *Journal of Mathematical Finance*, *8*(1). Retrieved from https://www.scirp.org/journal/paperinformation.aspx?paperid=82638 doi: 10.4236/jmf.2018.81012

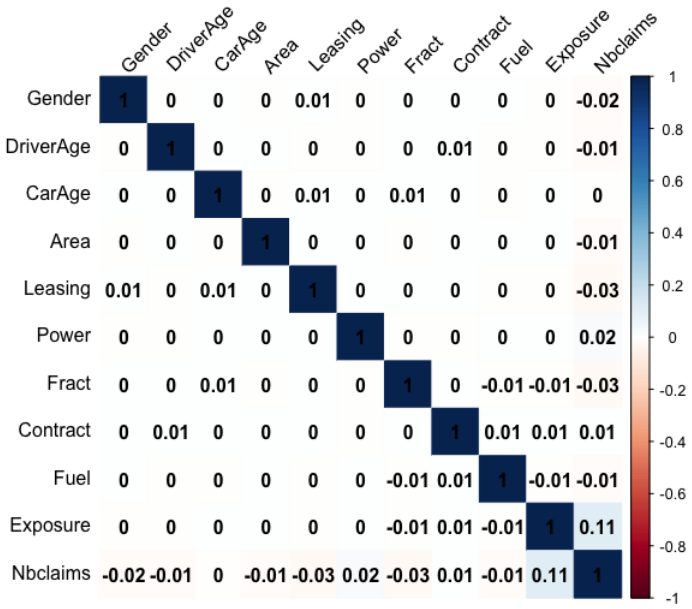
Parhi, R., & Nowak, R. D. (2021). Banach space representer theorems for neural networks and ridge splines. *Journal of Machine Learning Research*, *22*(43), 1–40. Retrieved from https://arxiv.org/abs/2006.05626 doi: 10.48550/ARXIV.2006.05626

Ridgeway, G. (2007). *Generalized boosted models: A guide to the gbm package*. Retrieved from https://moodle.uclouvain.be/pluginfile.php/173575/mod_resource/content/0/gbmExplanations.pdf

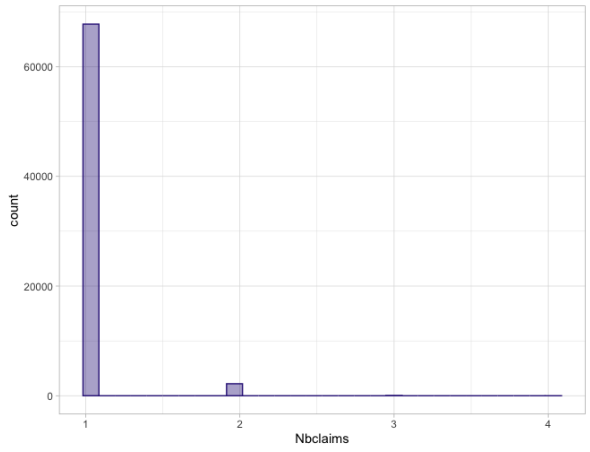
Unknown. (n.d.). Modelling claim frequency. *UiO*, 1–20. Retrieved from https://www.uio.no/studier/emner/matnat/math/nedlagte-emner/STK2510/v08/undervisningsmateriale/ch8b.pdf

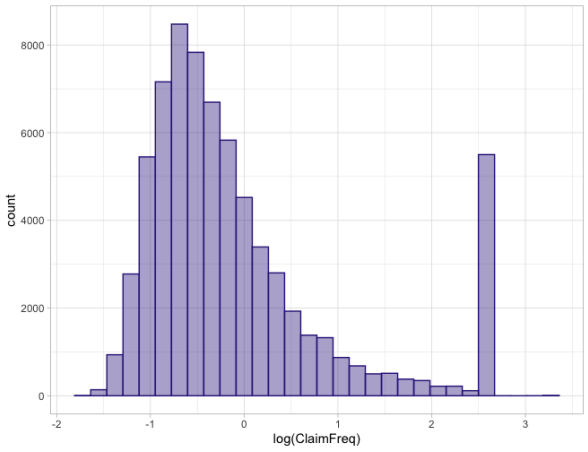
3.7 Appendices

3.7.1 Correlation matrix



3.7.2 Distributions of Number of claims and the logarithmic claim frequencies

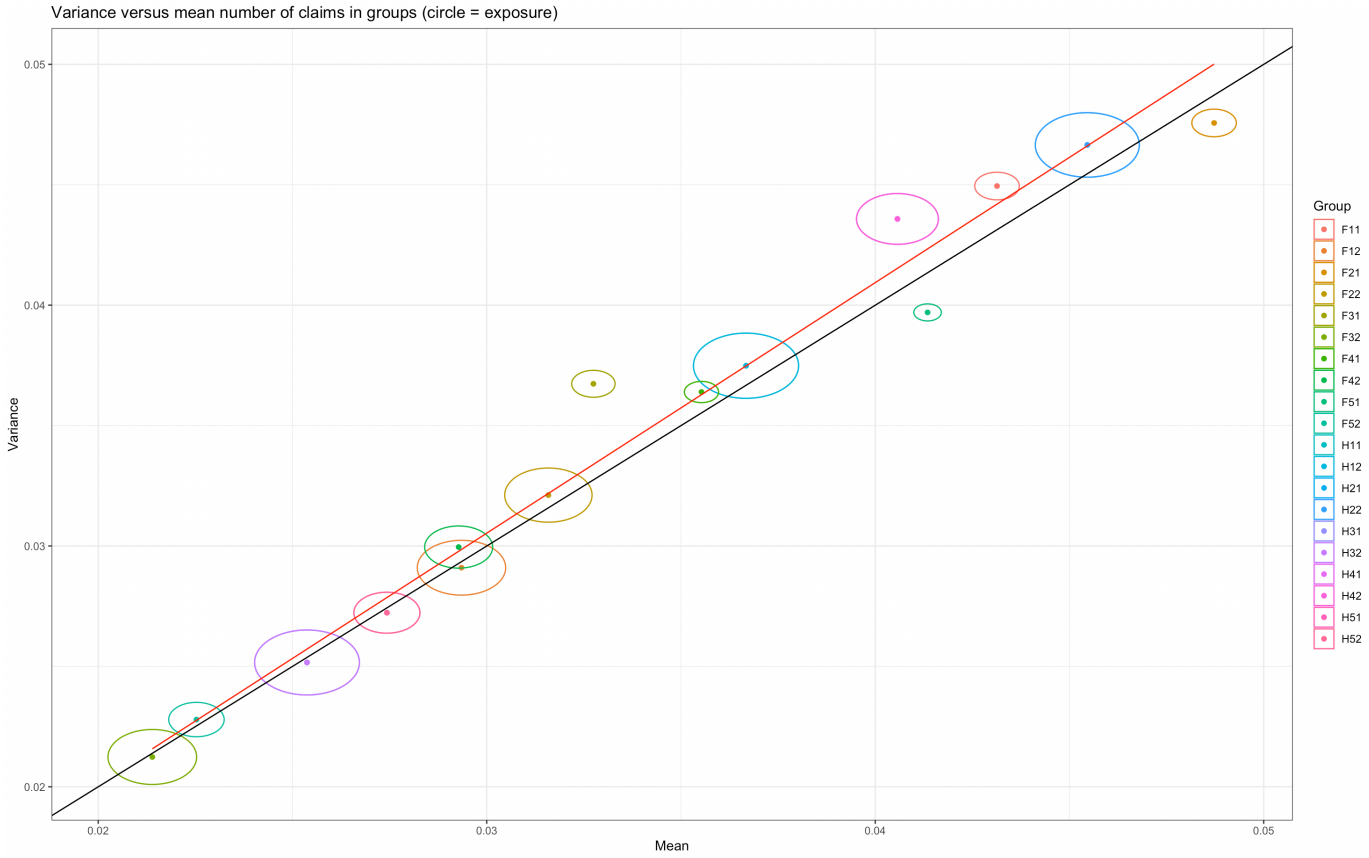


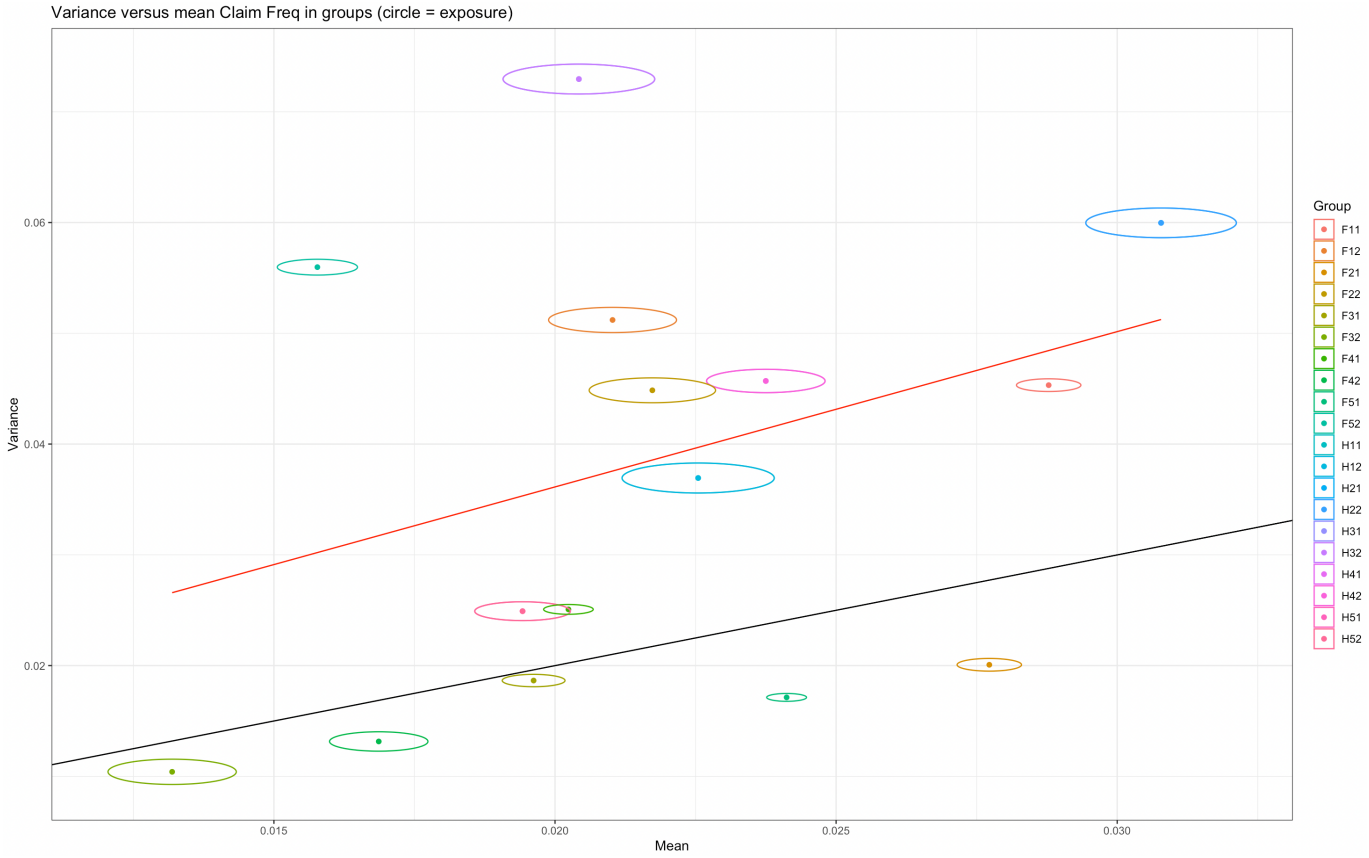


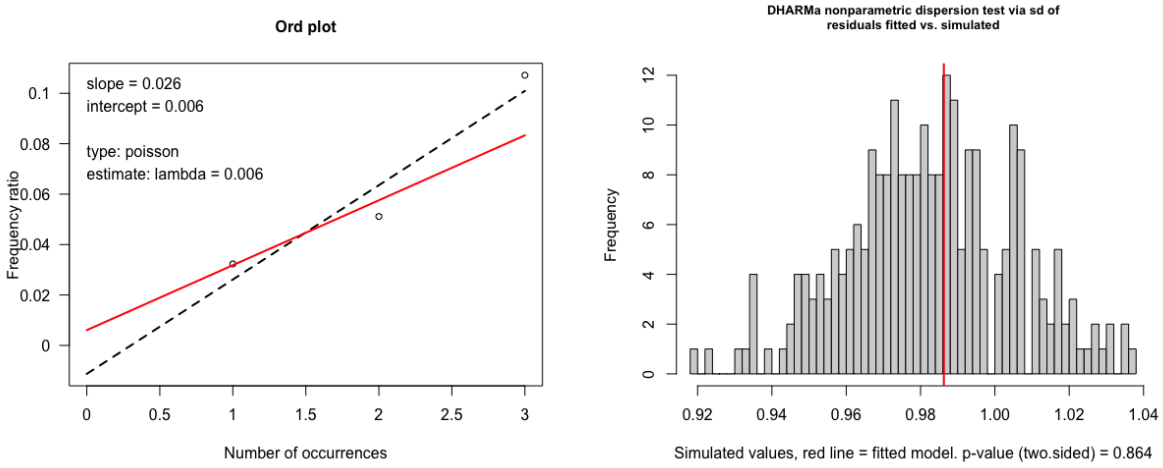
3.7.3 Statistics Number of claims and claim frequencies

	Nbclaims	ClaimFreq
mean	0.03	0.02
sd	0.18	0.21
median	0.00	0.00
trimmed	0.00	0.00
mad	0.00	0.00
min	0.00	0.00
max	3.00	12.50
range	3.00	12.50
skew	5.68	36.11
kurtosis	33.68	1935.73
se	0.00	0.00

3.7.4 Over-dispersion







3.7.5 Categorical features distributions

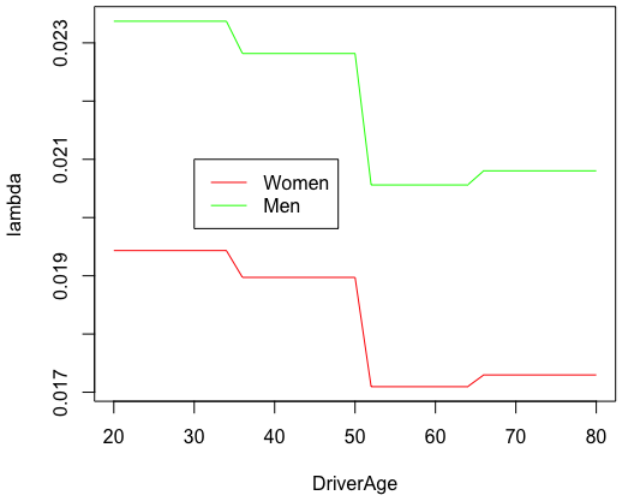
3.7.6 Negative binomial GLM coefficients

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.216385   0.125294 -25.671  < 2e-16 ***
Gender2     -0.170152   0.042160  -4.036 5.44e-05 ***
DriverAge   -0.003815   0.001438  -2.653 0.007983 **
CarAge      -0.005559   0.008149  -0.682 0.495130
Area2        0.173548   0.055424   3.131 0.001741 **
Area3       -0.334285   0.063143  -5.294 1.20e-07 ***
Area4        0.031335   0.065772   0.476 0.633773
Area5       -0.249829   0.083767  -2.982 0.002860 **
Leasing2    -0.351543   0.047298  -7.432 1.07e-13 ***
Power2       0.191462   0.068059   2.813 0.004906 **
Power3       0.293318   0.070607   4.154 3.26e-05 ***
Power4       0.520906   0.083213   6.260 3.85e-10 ***
Fract2      -0.199776   0.057283  -3.488 0.000488 ***
Fract3      -0.335827   0.047180  -7.118 1.09e-12 ***
Contract2    0.062699   0.049022   1.279 0.200902
Contract3    0.106487   0.059983   1.775 0.075851 .
Fuel2       -0.168181   0.048188  -3.490 0.000483 ***
Fuel3       -0.124255   0.065065  -1.910 0.056170 .
Fuel4       -0.194728   0.077518  -2.512 0.012003 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

3.7.7 Poisson GLM

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.35161    0.11926 -28.104  < 2e-16 ***
Gender2     -0.18454    0.04736  -3.897 9.74e-05 ***
DriverAgeQ2 -0.02400    0.06377  -0.376 0.706590
DriverAgeQ3 -0.12817    0.06556  -1.955 0.050564 .
DriverAgeQ4 -0.11645    0.06607  -1.762 0.077996 .
CarAgeQ2    -0.08809    0.06136  -1.436 0.151079
CarAgeQ3    -0.06956    0.06356  -1.094 0.273832
CarAgeQ4    -0.03812    0.07107  -0.536 0.591691
Area2        0.15310    0.06202   2.469 0.013567 *
Area3       -0.33865    0.07035  -4.814 1.48e-06 ***
Area4        0.01660    0.07362   0.225 0.821630
Area5       -0.30865    0.09562  -3.228 0.001247 **
Leasing2    -0.36923    0.05295  -6.973 3.11e-12 ***
Power2       0.22517    0.07733   2.912 0.003594 **
Power3       0.35780    0.07984   4.481 7.42e-06 ***
Power4       0.55610    0.09426   5.900 3.64e-09 ***
Fract2      -0.21657    0.06479  -3.343 0.000829 ***
Fract3      -0.31981    0.05279  -6.059 1.37e-09 ***
Contract2    0.06892    0.05479   1.258 0.208418
Contract3    0.08589    0.06777   1.267 0.205012
Fuel2       -0.14043    0.05418  -2.592 0.009537 **
Fuel3       -0.09949    0.07303  -1.362 0.173125
Fuel4       -0.21035    0.08771  -2.398 0.016472 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

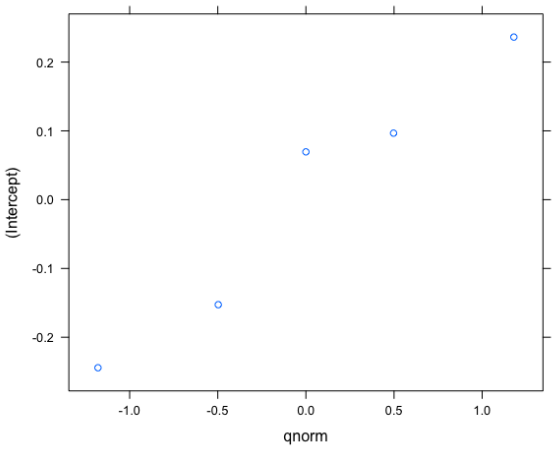
```
Area=factor(3),
Leasing=factor(2),CarAge=3, Contract=factor(3), Exposure=1, Fuel=factor(2), Power=factor(3), Fract= factor(1)
```

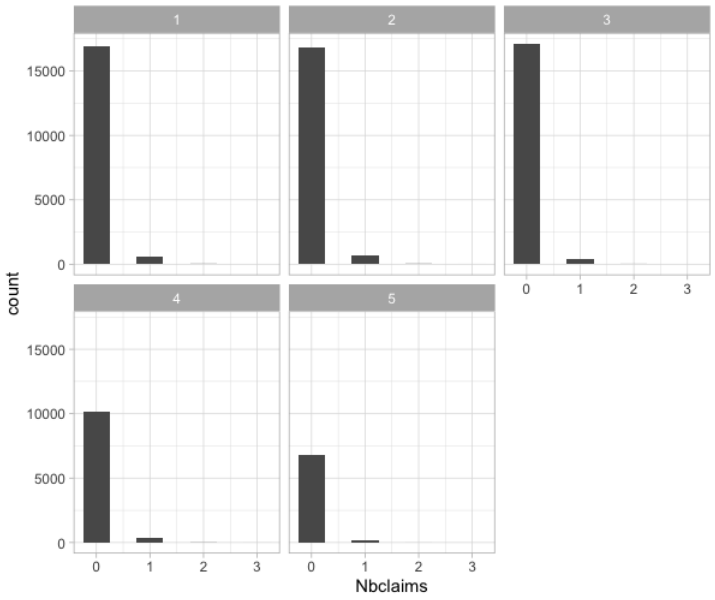


3.7.8 Poisson Mixed GLM coefficients

```
Random effects:
 Groups Name        Variance Std.Dev.
 Area   (Intercept) 0.03311  0.182
Number of obs: 70000, groups:  Area, 5

Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.288536   0.145482 -22.604  < 2e-16 ***
Gender2     -0.170188   0.042076  -4.045 5.24e-05 ***
DriverAge   -0.003818   0.001435  -2.660 0.007811 **
CarAge      -0.005614   0.008132  -0.690 0.489940
Leasing2    -0.351704   0.047189  -7.453 9.12e-14 ***
Power2       0.191313   0.067945   2.816 0.004867 **
Power3       0.293143   0.070483   4.159 3.20e-05 ***
Power4       0.520693   0.083043   6.270 3.61e-10 ***
Fract2      -0.200031   0.057162  -3.499 0.000466 ***
Fract3      -0.335716   0.047081  -7.131 9.99e-13 ***
Contract2    0.062739   0.048926   1.282 0.199734
Contract3    0.106463   0.059860   1.779 0.075319 .
Fuel2       -0.168391   0.048090  -3.502 0.000463 ***
Fuel3       -0.124459   0.064930  -1.917 0.055260 .
Fuel4       -0.194696   0.077366  -2.517 0.011851 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```





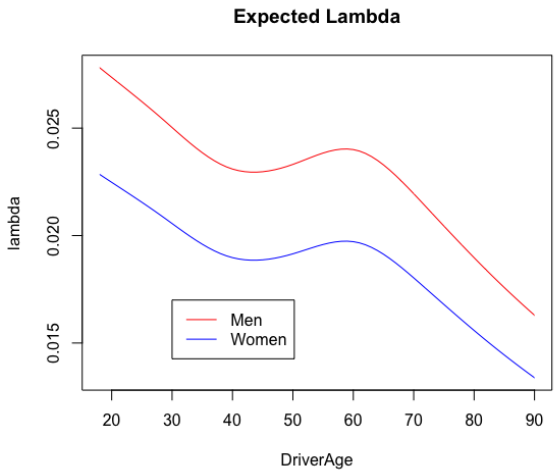
3.7.9 Zero-inflated Poisson regression

```
Count model coefficients (poisson with log link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.283952   0.223607 -19.158  < 2e-16 ***
Gender2     -0.239983   0.076625  -3.132  0.00174 **
DriverAge   -0.002984   0.002652  -1.125  0.26060
CarAge      -0.011197   0.014182  -0.790  0.42980
Leasing2    -0.243619   0.085883  -2.837  0.00456 **
Power2       0.333623   0.122870   2.715  0.00662 **
Power3       0.371297   0.125713   2.954  0.00314 **
Power4       0.611787   0.147539   4.147 3.37e-05 ***
Fract2      -0.340345   0.104783  -3.248  0.00116 **
Fract3      -0.372722   0.087009  -4.284 1.84e-05 ***
Contract2   -0.038880   0.088482  -0.439  0.66036
Contract3    0.135542   0.107284   1.263  0.20645
Fuel2       -0.211092   0.088577  -2.383  0.01716 *
Fuel3       -0.132085   0.116942  -1.129  0.25869
Fuel4       -0.195009   0.138354  -1.409  0.15869
Area2        0.152298   0.114507   1.330  0.18351
Area3       -0.263873   0.138345  -1.907  0.05647 .
Area4       -0.601557   0.103665  -5.803 6.52e-09 ***
Area5       -0.108088   0.182722  -0.592  0.55416
```

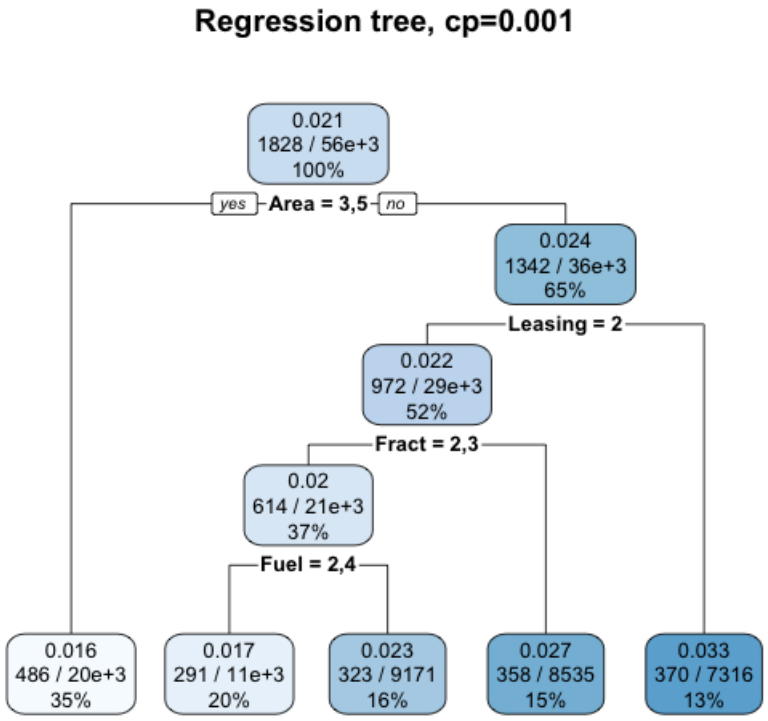
```
Zero-inflation model coefficients (binomial with logit link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.23764    0.52701  -6.143 8.08e-10 ***
Gender2      0.13970    0.23296   0.600   0.5487
DriverAgeQ2  0.40630    0.29987   1.355   0.1754
DriverAgeQ3  0.50526    0.32641   1.548   0.1216
DriverAgeQ4  0.17776    0.34033   0.522   0.6015
CarAgeQ2     0.10012    0.29426   0.340   0.7337
CarAgeQ3    -0.27904    0.31566  -0.884   0.3767
CarAgeQ4     0.01514    0.33058   0.046   0.9635
Leasing2     0.40236    0.26852   1.498   0.1340
Power2      -0.17875    0.34833  -0.513   0.6078
Power3      -0.07730    0.36255  -0.213   0.8312
Power4      -0.43106    0.47335  -0.911   0.3625
Fract2      -0.28664    0.32011  -0.895   0.3706
Fract3      -0.21841    0.26739  -0.817   0.4140
Contract2    0.25038    0.27932   0.896   0.3701
Contract3    0.24841    0.33477   0.742   0.4581
Fuel2       -0.36327    0.25387  -1.431   0.1525
Fuel3        0.06485    0.32572   0.199   0.8422
Fuel4       -0.22780    0.40970  -0.556   0.5782
Area2        0.18661    0.28636   0.652   0.5146
Area3        0.46883    0.34740   1.350   0.1772
Area4       -0.79510    0.43847  -1.813   0.0698 .
Area5        0.19302    0.43225   0.447   0.6552
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.7.10 General additive model

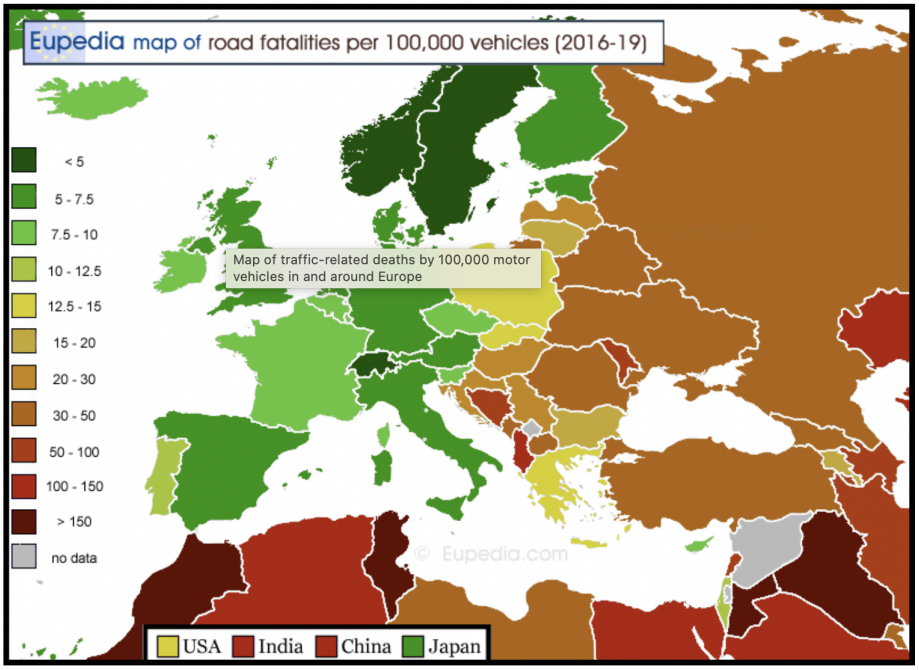
```
Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.43510    0.09545 -35.989  < 2e-16 ***
Gender2     -0.17018    0.04209  -4.043 5.28e-05 ***
Area2        0.17348    0.05533   3.135 0.001716 **
Area3       -0.33435    0.06306  -5.302 1.14e-07 ***
Area4        0.03153    0.06566   0.480 0.631139
Area5       -0.25000    0.08366  -2.988 0.002805 **
Leasing2    -0.35152    0.04721  -7.446 9.63e-14 ***
Power2       0.19143    0.06797   2.816 0.004857 **
Power3       0.29322    0.07051   4.158 3.20e-05 ***
Power4       0.52087    0.08308   6.270 3.62e-10 ***
Fract2      -0.19986    0.05719  -3.495 0.000475 ***
Fract3      -0.33581    0.04710  -7.129 1.01e-12 ***
Contract2    0.06272    0.04895   1.281 0.200079
Contract3    0.10656    0.05989   1.779 0.075180 .
Fuel2       -0.16813    0.04811  -3.495 0.000475 ***
Fuel3       -0.12414    0.06496  -1.911 0.055988 .
Fuel4       -0.19452    0.07740  -2.513 0.011960 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



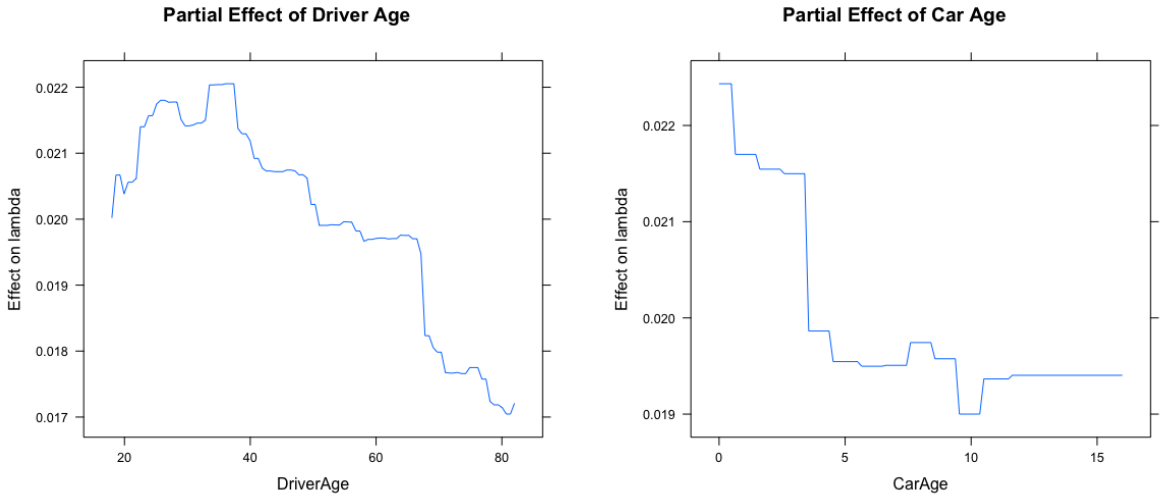
3.7.11 Retained regression tree

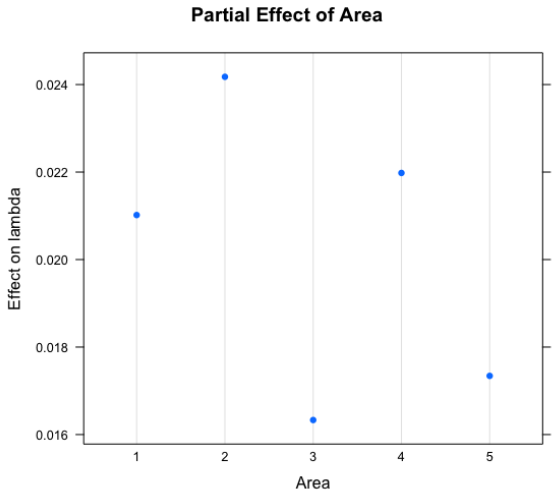


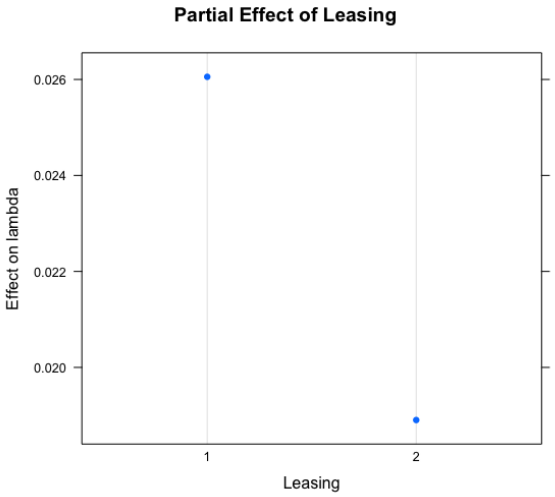
3.7.12 Maps

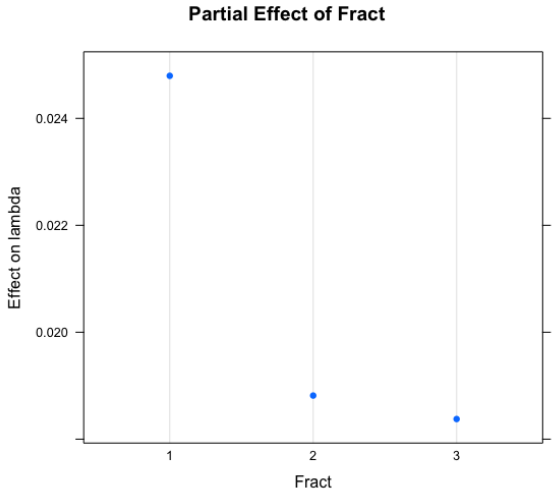


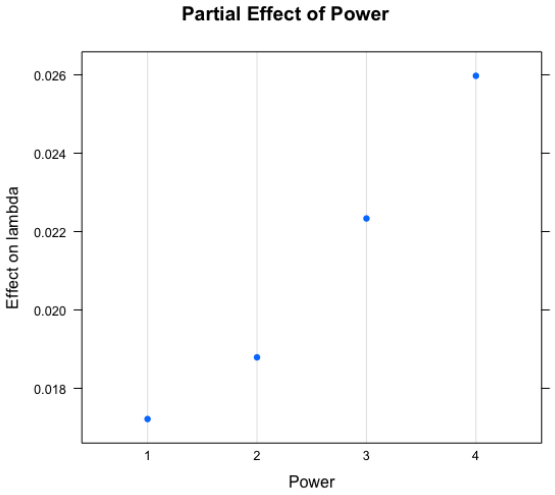
3.7.13 Boosting: function $f(.)$ of explanatory features

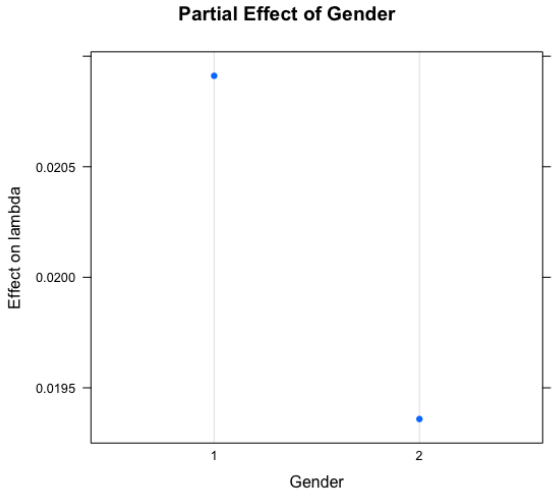


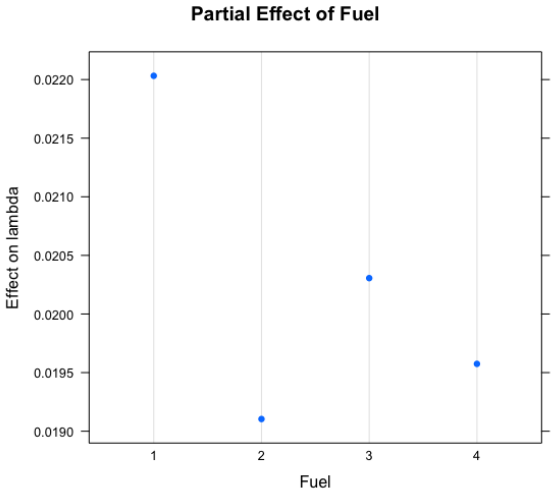

Partial Effect of Area
Effect on lambda
0.024
0.022
0.020
0.018
0.016
1
2
3
4
5
Area

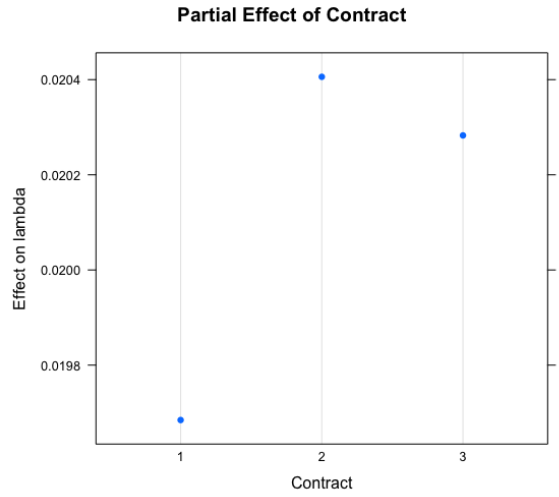

Partial Effect of Leasing
Effect on lambda
0.026
0.024
0.022
0.020
1
2
Leasing

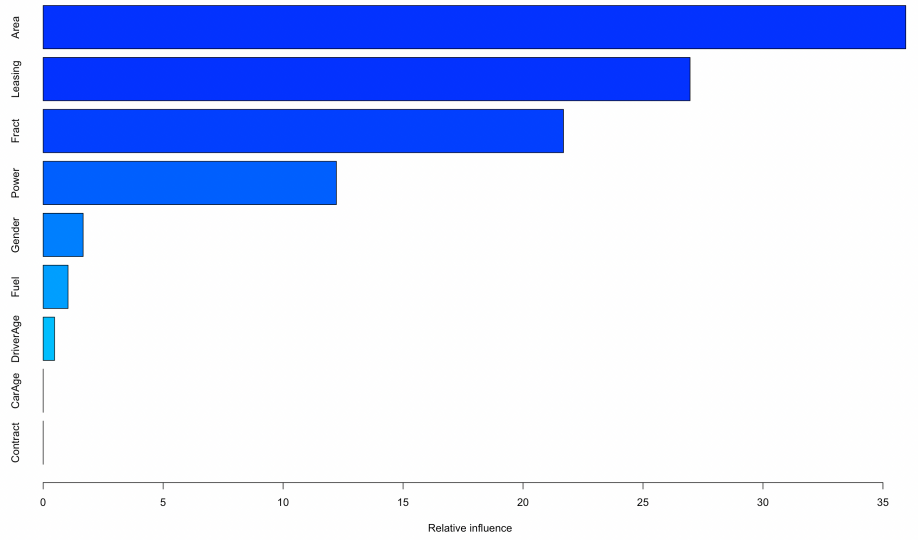

Partial Effect of Fract
Effect on lambda
0.024
0.022
0.020
1
2
3
Fract


Partial Effect of Power
Effect on lambda
0.026
0.024
0.022
0.020
0.018
1
2
3
4
Power


Partial Effect of Gender
Effect on lambda
0.0205
0.0200
0.0195
1
2
Gender


Partial Effect of Fuel
Effect on lambda
0.0220
0.0215
0.0210
0.0205
0.0200
0.0195
0.0190
1
2
3
4
Fuel





3.7.14 Neural Net Calibration

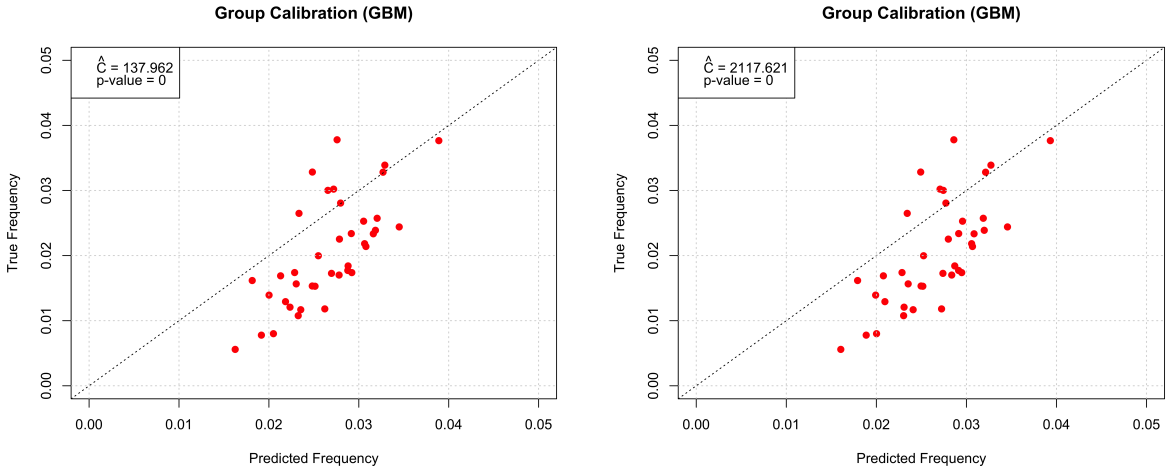


Figure 21: Group calib. for NN with 50 ReLu on 1 layer

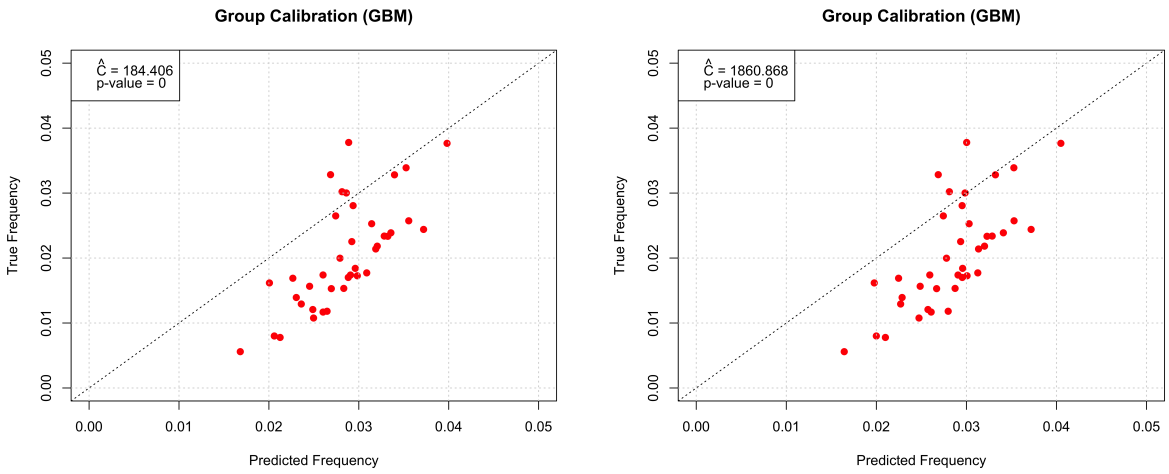
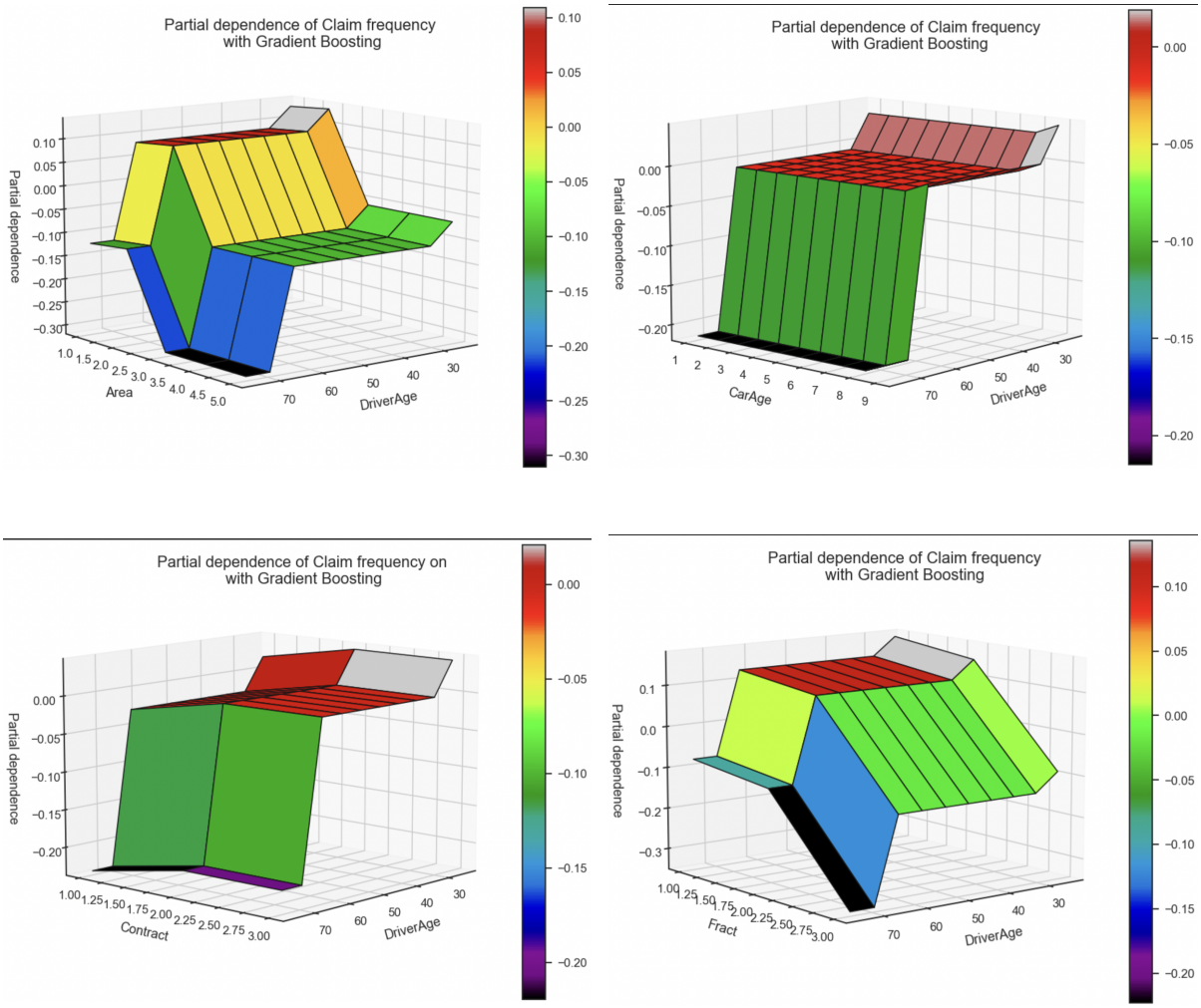
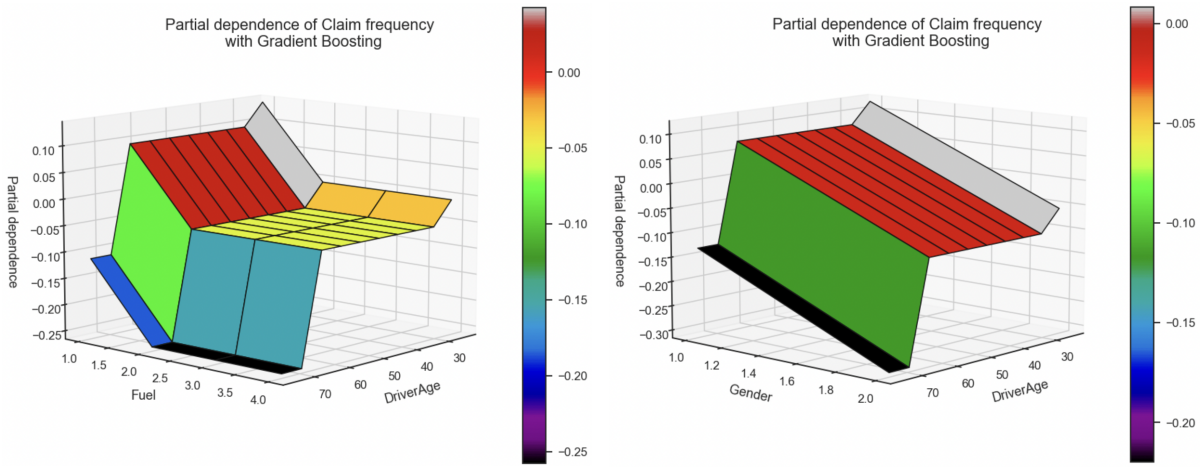
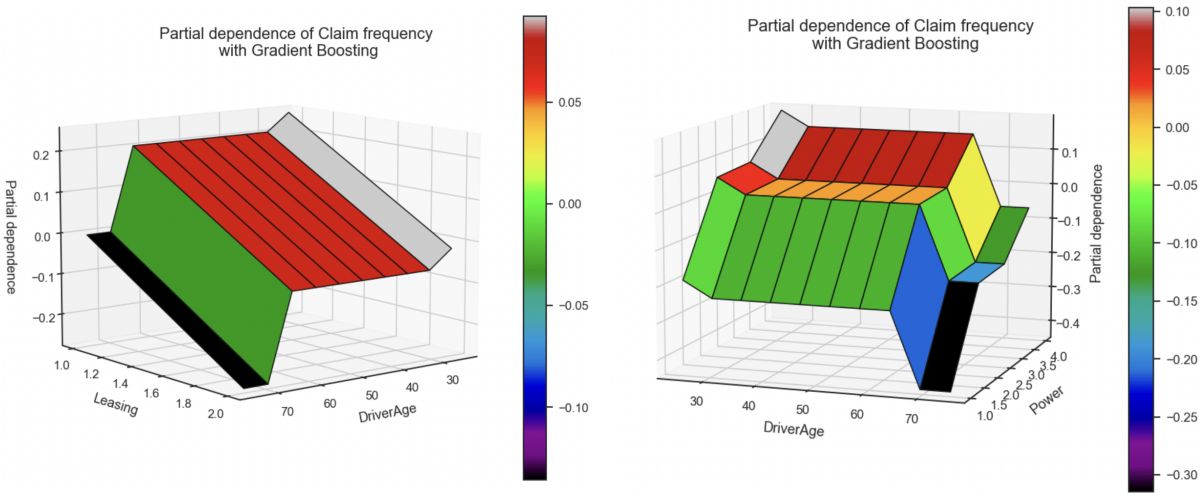


Figure 22: Group calib. for NN with 100 ReLu on 1 layer

3.7.15 Global & Local interpretation




Partial dependence of Claim frequency
with Gradient Boosting
Partial dependence
Fuel
DriverAge
Partial dependence of Claim frequency
with Gradient Boosting
Partial dependence
Gender
DriverAge


Partial dependence of Claim frequency
with Gradient Boosting
Partial dependence
Leasing
DriverAge
Partial dependence of Claim frequency
with Gradient Boosting
Partial dependence
DriverAge
Power

3.8 Additional explanations about implemented Bootstrap & Random Forest

3.8.1 Bootstrap

Bootstrapping methods such as **bagging** or **boosting** are used to avoid the lack of reproducibility of methods previously seen. We can perform the parametric bootstrapping which generates its samples from the assumed distribution of the data, using estimated parameter values or non-parametric bootstrapping that approximates the empirical distribution function F by:

$$\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} 1_{Y_i \leq x}$$

In addition, as a decision tree is a function that partitions the input space χ into subsets χ_t, to which a unique prediction value $\hat{Y}_t$ is associated and is trained according to some splitting criterion, such a function is unstable and may change a lot if one trained it on a different sample. Thus, in order to improve accuracy of prediction, we use a bootstrap aggregating version of decision trees. And so as the regression tree estimator is $\hat{\lambda}(x)$ and simulating independent observations such that $N_t^* \sim Pois(\hat{\lambda}(x_i)w_i)$ then the bagging estimator is:

$$x \longrightarrow \hat{\lambda}^m_{BAG}(x) = \frac{1}{M}\sum_{m=1}^{M} \hat{\lambda}^{m*}(x)$$

where $\hat{\lambda}^{m*}(x)$ are the parametric bootstrap samples. But bagging algorithm is too static and it does not really lead to improvement and bias reduction.

3.8.2 Random forest

The random forest predictor is a function that averages the predictions of M decision trees of fixed depth estimated on M bootstrapped samples of size n, where the split is based on a subset of p^* inputs $(1 \leq p^* \leq p)$:

$$f_{RF}(x) = \frac{1}{M}\sum_{m=1}^{M} \tau(x^*_m)$$

the Poisson Random Forest algorithm differs from the Bagging for Poisson Regression Tree algorithm as soon as we select a number of features $p^* \leq p$ and thus we may miss the optimal split. Choosing our number of binary trees M we apply $d = [log_2(M) + 1]$ to determine the depth of each tree.