Predicting Car Insurance Premiums Using Bayesian Modelling

Pham Quang Bach, Hannes Fant, Irmuun Tuguldur

November 30, 2024

Chapter 1

Introduction

The aim of this project is to determine the relationship between a vehicles insurance premium and several relevant factors such as vehicle type, number of drivers, number of previous complaints etc. We wanted to answer the question of if the premium of a given vehicle can, with a certain degree of accuracy, be predicted given some specific properties of the vehicle and its owner.

In this report, three different models ranging from a simple linear model, to a more complex spline model and a hierarchical model were all trained and examined to determine their ability to accurately predict a suitable insurance premium of a given vehicle.

Chapter 2

Description of the Dataset

The dataset used was taken from the motor vehicle insurance portfolio of a non-life insurance company operating in Spain [1]. The full dataset consist of 105,555 rows each with 30 variables ranging from starting and renewal dates of the contract, to detailed information about the insured vehicle such as weight, power, and fuel type.

Considering our limited time and computational resources, it was deemed impractical to analyze the entire dataset. It was thus decided that we would focus only on insurance of passenger cars (Type_risk=3 in the dataset) with contracts starting in the year 2018 for analysis. This selection, along with the removal of empty rows and NA values resulted in a final reduced dataset of 8935 data points.

Among the 30 variables available, not all variables were relevant in predicting the cost of premiums, such as distribution channel, date of renewal, or number of doors. Furthermore, variables such as the vehicle's value, length, weight, cylinder capacity, and power suffered from multicollinearity. As a result, we decided to only include 5

numerical and 1 categorical variable in the models to predict the premiums. These variables are:

- Seniority: Total number of years the customer has been associated with the insurance entity.
- Policies_in_force: Total number of policies held by the customer in the insurance entity during the reference period.
- Cost_claims_year: Total cost of claims for the insurance policy during the current year.
- N_claims_year: Total number of claims incurred for the insurance policy during the current year.
- Value_vehicle: Market value of the vehicle as of 31/12/2019.
- Area: A dichotomous variable indicating the type of area in terms of traffic conditions. 0 indicates a rural area and 1 indicates an urban area, with the threshold set at 30,000 inhabitants.

As far as we are aware, this dataset has not been used in any other relevant online case study before, as the paper was published in September of this year.

Chapter 3

Priors

Since brms default priors are flat and improper, there is a need to replace them with informative or weakly informative priors. This was done for the intercept and all five main variable of all three models.

The relevant priors were conjured with the use of domain knowledge. For example, we assumed that an average vehicle may be worth around 20,000€ with a standard deviation of about 5000€, which in turn resulted in the prior for the Value_vehicle variable. Due to the CLT we settled on normal or log-normal priors for all our models. The final priors and their justifications were as follows:

- Intercept: normal(250, 100)
- Seniority: normal(10, 5)
  - In general, customers tend to be quite loyal to their insurance company.
- Policies_in_force: normal(1, 1)

  - We believe the insurance company in question only insured vehicles, most customers may only have one or two vehicles.

- Cost_claims_year: normal(250, 200)
  - Most customers do not incur any claim costs, though the ones that do increases the average significantly, therefore 250€ seemed like a suitable choice.

- N_claims_year: normal(1, 1)
  - Once again, a claim is an exception from the norm, though customers who issue claims generally seem to do so more than once a year, therefore we settled on setting the mean and standard deviation to 1.

- Value_vehicle: normal(20 000, 5 000)
  - As already mentioned, we believe the average price of an insured vehicle is around 20 000€, with a relatively large standard deviation of 5 000€.

Chapter 4

Models

For the task of modeling the vehicle's potential insurance premium rate, three different models were evaluated. A linear model, a hierarchical linear model, and a nonlinear model, explored in that order.

4.1 Linear Model

We initially made use of a linear model thanks to its simplicity and ease of interpretation. Furthermore it also serves as a good baseline for our analysis. Using the preselected numerical variables above, the resulting formula can be formulated as follows in brms:

```
linear_model_formula <- bf(
  Premium ~ 1 + Seniority + Policies_in_force + Cost_claims_year + N_
   claims_year + Value_vehicle,
  family = "gaussian",
  center = FALSE)
```

The model was ran using the default settings and parameters given by brms along with our aforementioned priors with the following R code:

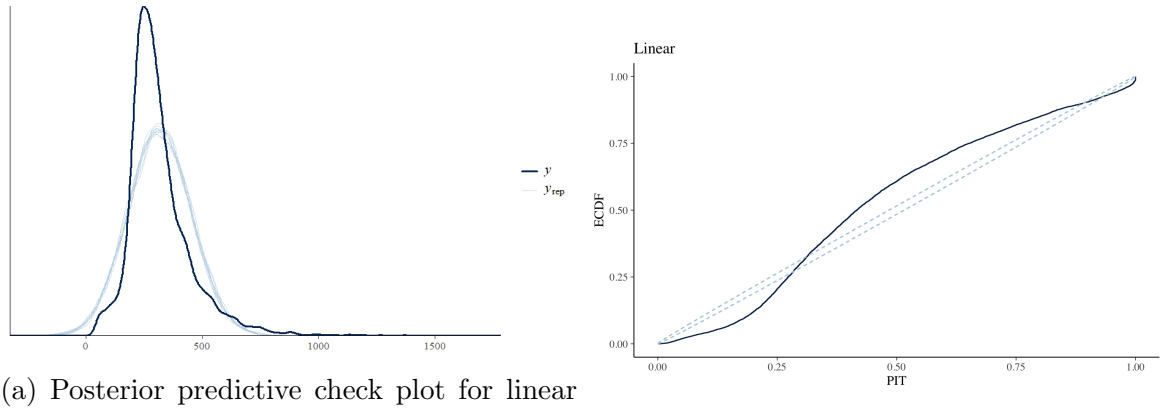
```r
linear_model_fit <- brm(
  formula = linear_model_formula,
  prior = linear_model_priors,
  data = insurance_data
)
```

The chains ran perfectly with zero warning. The summary of the model can be seen in Figure 4.1. As can be seen in the aforementioned figure, the Rhat, Bulk_ESS and Tail_ESS are all satisfactory, indicating that all chains have both mixed and converged, and no further adjustments to the model parameters is necessary.,

```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: Premium ~ 1 + Seniority + Policies_in_force + Cost_claims_year + N_claims_year + Value_vehicle
   Data: insurance_data (Number of observations: 8001)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Regression Coefficients:
                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept          258.46      3.95   250.80   266.50 1.00     3867     3109
Seniority           -1.98      0.34    -2.64    -1.33 1.00     2494     2146
Policies_in_force   -4.23      0.77    -5.72    -2.72 1.00     2977     2617
Cost_claims_year     0.01      0.00     0.01     0.02 1.00     2258     2060
N_claims_year        1.61      0.93    -0.22     3.48 1.00     1112     1276
Value_vehicle        0.00      0.00     0.00     0.00 1.00     4460     3035

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma   131.43      1.02   129.44   133.48 1.00     3617     2145

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Figure 4.1: Summary of the linear model.

However, examining posterior predictive check in figure 4.2 reveals that the model did not capture the distribution of the predicted Premium variable very well. The model has a higher probability for low values of premiums than the dataset. Moreover, the model has a non-zero probability of negative premiums, which naturally should be impossible to obtain. This major difference between the predicted data and observed data suggest that our linear model, at least with these parameters and variables, is inadequate to model the insurance premium rates of cars in 2018.



(a) Posterior predictive check plot for linear model.

(b) ECDF Plot.

Figure 4.2: Posterior predictive checks for linear model.

Examining the posterior predictive R^2 distribution and the LOO R^2 distribution of the model fit, we see that the mean of the bayes_R2 and loo_R2 are both approximately 0.06. This similarity between the two values indicated that the model is not likely to have overfit the data.

To conduct sensitivity analysis with respect to prior choice, we re-trained the model with the following clearly nonsensical priors:

- Intercept: normal(100, 10)
- Seniority: normal(100, 50)
- Policies_in_force: normal(10, 10)
- Cost_claims_year: normal(10, 10)
- N_claims_year: normal(10, 10)
- Value_vehicle: normal(50, 4)

If the model is sensitive to the choice of priors, a significantly different result should be obtained at this stage with these nonsensical priors. Comparing the models resulted in a negligible difference when using loo_compare as shown in figure 4.3. We therefore concluded that the linear model is not very sensitive to changes in its priors.

```
                                elpd_diff se_diff
linear_model_fit_changed_prior   0.0       0.0
linear_model_fit                -2.1       8.7
```

Figure 4.3: Result of loo_compare on the two linear models with different priors.

4.2 Hierarchical Model

During our initial exploratory data analysis, we discovered that the Area variable could be a good hyper parameter to use within the context of a hierarchical model after having fitted the linear model as a baseline. Our hypothesis was that rural and urban premiums may differ significantly, especially as traffic accidents are more likely to happen within rural areas than urban areas [2].

The adjusted brms formula is as follows:

```
hierarchical_linear_model_formula <- bf(
  Premium ~ 1 + Seniority + Policies_in_force + Cost_claims_year + N_
   claims_year + Value_vehicle + (1 + Seniority + Policies_in_force +
   Cost_claims_year + N_claims_year + Value_vehicle | Area),
  family = "gaussian",
  center = FALSE
)
```

The hierarchical model was also fitted using the default settings and parameters given by brms with the following R code:

```r
hierarchical_linear_model_fit <- brm(
  formula = hierarchical_linear_model_formula,
  prior = linear_model_priors,
  data = insurance_data
)
```

Unfortunately, according to the warnings, all transitions hit the maximum treedepth or diverges as can be seen in Figure 4.4. Increasing the treedepth did not improve the result. This resulted in a significantly suboptimal Bulk_ESS and Tail_ESS as can be seen in Figure 4.5. As both the aforementioned ESS quantities should be above 100 in addition to all the Rhat values being larger than 1.05, with the largest Rhat value being 3.40, it was concluded that the chains did not converge at all.

```
Warning: 63 of 4000 (2.0%) transitions ended with a divergence.
See https://mc-stan.org/misc/warnings for details.

Warning: 3937 of 4000 (98.0%) transitions hit the maximum treedepth limit of 10.
See https://mc-stan.org/misc/warnings for details.
```

Figure 4.4: Hierarchical model errors.

```
Formula: Premium ~ 1 + Seniority + Policies_in_force + Cost_claims_year + N_claims_year + Value_vehicle + (1 + Seniority + Policies_in_force +
Cost_claims_year + N_claims_year + Value_vehicle | Area)
   Data: insurance_data (Number of observations: 8001)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Multilevel Hyperparameters:
~Area (Number of levels: 2)
                                    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)                          99.45    110.62     0.16   347.13 2.55        5       23
sd(Seniority)                           3.59      3.68     0.34    14.16 3.03        5       19
sd(Policies_in_force)                   9.83     10.04     0.50    30.48 2.46        5       20
sd(Cost_claims_year)                    1.54      1.41     0.08     3.74 2.59        5       14
sd(N_claims_year)                       3.84      5.26     0.13    20.85 3.40        4       11
sd(Value_vehicle)                       1.76      1.67     0.25     4.68 2.41        5       16
cor(Intercept,Seniority)               -0.27      0.50    -0.95     0.54 2.27        5       17
cor(Intercept,Policies_in_force)       -0.19      0.43    -0.80     0.41 2.22        5       43
cor(Seniority,Policies_in_force)       -0.10      0.39    -0.66     0.54 1.89        6       16
cor(Intercept,Cost_claims_year)         0.14      0.45    -0.63     0.81 1.94        6       11
cor(Seniority,Cost_claims_year)         0.01      0.40    -0.60     0.63 2.13        5       11
cor(Policies_in_force,Cost_claims_year) -0.10     0.30    -0.64     0.72 1.46       11       15
cor(Intercept,N_claims_year)           -0.15      0.42    -0.65     0.66 2.65        5       12
cor(Seniority,N_claims_year)            0.00      0.41    -0.61     0.71 2.81        5       12
cor(Policies_in_force,N_claims_year)   -0.01      0.45    -0.64     0.72 1.94        6       19
cor(Cost_claims_year,N_claims_year)     0.08      0.30    -0.35     0.73 1.39        9       58
cor(Intercept,Value_vehicle)            0.11      0.38    -0.63     0.69 1.85        6       15
cor(Seniority,Value_vehicle)           -0.24      0.35    -0.87     0.39 2.12        5       11
cor(Policies_in_force,Value_vehicle)    0.15      0.27    -0.42     0.59 1.37        9       50
cor(Cost_claims_year,Value_vehicle)    -0.09      0.49    -0.86     0.53 2.41        5       14
cor(N_claims_year,Value_vehicle)        0.11      0.33    -0.63     0.71 1.57        7       20

Regression Coefficients:
                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept           34.12     50.39    -1.36   139.85 3.05        5       24
Seniority            0.49      1.37    -1.96     1.94 3.03        5       13
Policies_in_force    1.64      2.35    -0.21     6.63 2.24        5       13
Cost_claims_year     0.66      0.67    -0.02     2.45 3.53        4       11
N_claims_year        0.56      1.19    -0.91     2.66 2.27        5       11
Value_vehicle        0.18      0.66    -0.74     1.25 2.92        5       14

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma   146.49     15.84   129.07   165.39 2.07        5       32
```

Figure 4.5: Hierarchical model summary.

Common ways to improve the convergence would be to improve the choice of priors, increase max treedepth, and increase iterations. However, after many changes to the

settings and priors without any improvements to the convergence, we decided, in line with the Bayesian workflow, to abandon this hierarchical model in favor of our other more promising models.

4.3 Non-linear Model

Finally, a non-linear model was chosen to examine the possibility that the relationships between the premium rate and the chosen variables are not linear. This was accomplished by changing the variables to spline terms and setting the model family to "lognormal" to account for the positively skewed data. The adjusted brms formula is as follows:

```
nonlinear_model_formula <- bf(
  Premium ~ 1 + s(Seniority) + s(Policies_in_force) + s(Cost_claims_
   year) + s(N_claims_year) + s(Value_vehicle),
  family = "lognormal",
  center = FALSE
)
```

The non-linear model was also fitted using the default settings and parameters given by brms with the following R code:

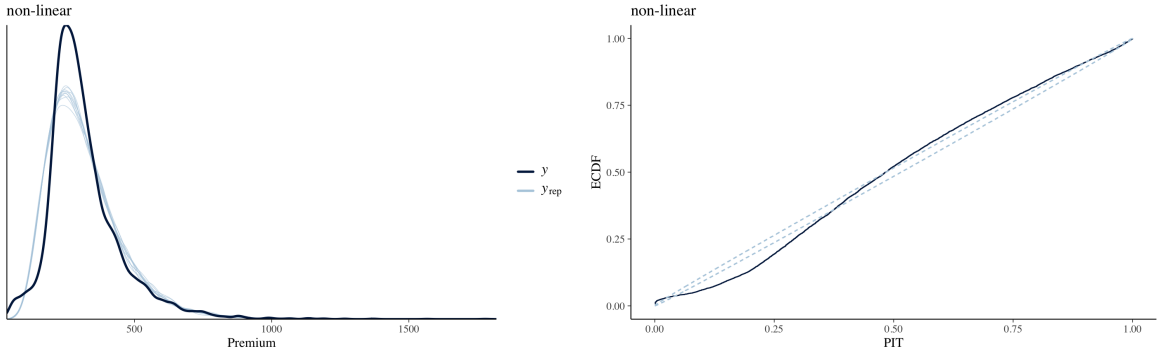
```
nonlinear_model_fit <- brm(
  formula = nonlinear_model_formula,
  prior = nonlinear_model_priors,
  data = insurance_data,
  family = "lognormal"
)
```

With these settings, a warning appeared mentioning that there were 31 divergent transitions after warm up as can be seen in Figure 4.6. However, despite the warning, we observed that the Rhat, Bulk_ESS, and Tail_ESS were all satisfactory, indicating that the chains have all merged and converged. Furthermore, since we are not expecting completely reliable inference in this project, we decided to continue with the analysis despite the warning.

```
Warning: There were 31 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help. See
http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup Family: lognormal
  Links: mu = identity; sigma = identity
Formula: Premium ~ 1 + s(Seniority) + s(Policies_in_force) + s(Cost_claims_year) + s(N_claims_year) +
s(Value_vehicle)
   Data: insurance_data (Number of observations: 8001)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Smoothing Spline Hyperparameters:
                         Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sds(sSeniority_1)            0.74      0.41     0.19     1.77 1.00     1263     1975
sds(sPolicies_in_force_1)    2.05      0.84     0.80     4.10 1.01     1067     1512
sds(sCost_claims_year_1)     1.54      1.13     0.33     4.60 1.00     1086     2196
sds(sN_claims_year_1)        0.81      0.69     0.04     2.62 1.00     1677     1968
sds(sValue_vehicle_1)        0.90      0.53     0.21     2.26 1.00     1154     1570

Regression Coefficients:
                    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept               5.68      0.00     5.67     5.68 1.00     8475     2575
sSeniority_1           -0.71      1.09    -3.10     1.42 1.00     2465     2477
sPolicies_in_force_1    0.20      0.98    -1.77     2.11 1.00     2899     2467
sCost_claims_year_1     1.17      1.87    -2.70     5.06 1.00     2218     2159
sN_claims_year_1        0.78      0.80    -0.84     2.35 1.00     4020     3315
sValue_vehicle_1        4.05      1.55     1.60     7.50 1.00     1710     2256

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.40      0.00     0.39     0.40 1.00     8526     3189

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Figure 4.6: Summary of non-linear model.

Based the posterior predictive graphs as shown in Figure 4.7, it was observed that the nonlinear model performs significantly than the simple linear model. However, the model still did not capture the distribution of the predicted variable perfectly. This is demonstrated by the insufficient probability for the lowest and average premiums.



(a) Non-linear post-predictive plot.

(b) Non-linear ECDF plot.

Figure 4.7: Posterior predictive check for nonlinear model.

Examining the posterior predictive R^2 distribution and the LOO R^2 distribution of the model fit, we see that the mean of the bayes_R2 is approximately 0.09 and loo_R2 is approximately 0.07. As the posterior estimate for the residual variance is underestimated, there is a high possibility that the model has slightly overfit the data.

The steps to conduct sensitivity analysis with respect to prior choice for this model is similar to the steps taken for the linear model. As such, we used the same nonsensical priors as detailed above in the section concerning the linear model. Comparing the two

non-linear model shows a negligible difference as detailed in Figure 4.8. We therefore concluded that the non-linear model is not very sensitive to prior changes either.

	elpd_diff	se_diff
nonlinear_model_fit	0.0	0.0
nonlinear_model_fit_changed_priors	-0.3	0.7

Figure 4.8: Results of loo_compare on the two nonlinear models with different priors.

Chapter 5

Model Comparison

Comparing the three models in Figure 5.1, it is clear that the non-linear spline-based model outperforms the two other models. As the difference in the standard error is significantly smaller than the difference in the ELPD, the conclusion that the non-linear model is the best out of the three does seem to hold water.

Looking at the R2 quantities for the three different models, it is again clear that the hierarchical model suffers from a very poor fit. The Pareto-k value for the model is very poor, which in turn makes the other statistical quantities unreliable and is also the reason why the LOO R2 is negative. This further underlines that the two winning approaches are the linear and non-linear models. The two models seem to perform similarly, though according to the posterior predictive checks in Figure 4.2 and Figure 4.7 it can be seen that the the non-linear model seems to outperform the linear model. Though, as already mentioned, neither model perfectly captures the true distribution of the data.

	elpd_diff	se_diff
nonlinear_model_fit	0.0	0.0
linear_model_fit	-998.3	100.6
hierarchical_linear_model_fit	-2296.9	110.3

Figure 5.1: loo_compare between the three models.

```
[1] "Linear Bayes R2"
     Estimate   Est.Error       Q2.5      Q97.5
R2 0.05807856 0.004866728 0.04868829 0.06794044
[1] "Linear Loo R2"
     Estimate   Est.Error       Q2.5     Q97.5
R2 0.06115097 0.006514494 0.04859518 0.0742284
[1] "Non-linear Bayes R2"
     Estimate   Est.Error      Q2.5      Q97.5
R2 0.08522283 0.006623717 0.0731094 0.09873671
[1] "Non-linear Loo R2"
     Estimate   Est.Error       Q2.5      Q97.5
R2 0.06750993 0.006569289 0.05475567 0.08040557
[1] "Hierarchical Bayes R2"
    Estimate Est.Error       Q2.5     Q97.5
R2 0.2094995  0.134569 0.06624984 0.3573991
[1] "Hierarchical Loo R2"
     Estimate  Est.Error       Q2.5      Q97.5
R2 -0.2812361 0.02825446 -0.3386111 -0.2296532
```

Figure 5.2: Bayes and LOO R^2 scores for the three models.

Chapter 6

Problems And Possible Improvement

During the project we encountered several major issues, with the majority in connection to the data. The first issue we encountered was that the non-linear model still experienced divergent transitions during fitting. This implied that the sampler struggled to explore the distributions effectively, which can lead to unreliable inferences later on. To address this we could change the parameters to allow the sampler to explore the posterior more thoroughly such as increasing the maximum tree depth or increasing the number of iterations. We could also modify the adaptation parameters to improve the performance of the sampler. Furthermore, there is also a possibility that the parameters might have been poorly defined initially, in which case adjusting them could improve the performance.

The most pressing issue by far was that our models performed poorly and were unable to predict the premiums with adequate accuracy. This inaccuracy could be due to several different factors, but we believe the two main ones were variable selection

and the choice of model.

Regarding variable selection, we believe that the poor performance and inadequate distribution is rooted in incorrectly defined parameters or an incorrect number of chosen parameters. To address this, different modeling choices and variable permutations could be explored to further identify the most relevant variables. This would involve testing various sets of predictor variables and evaluating their impact on the performance of the model.

Concerning the choice of model, the three models chosen for evaluation were not suitable for the dataset. In other words, the underlying distribution of the data could not adequately be modeled using our choice of models.

By addressing these key issues, we aim to improve the reliability and accuracy of our model predictions.

Chapter 7

Conclusion

Insurance premiums do not have to be an exact science. Depending on how the process at the company is defined, it may even be more of an art than a science. To accurately predict premiums there must be some underlying true distribution of the data, or in other words, a rigid process or formula for how premiums are determined. In the case of the data concerned in this study, that distribution may not be so simple, especially not if the process at the company where the data was collected is closer to an art, instead of a science.

After evaluating three different models, we can with adequate confidence say that the underlying distribution of the data cannot be modeled linearly. This is further underlined by the performance improvement of the non-linear spline model in comparison with the linear model. Going forward, the model could further be improved not only by adjusting the parameters for the non-linear model while being careful not to overfit the

data, but also by evaluating different families of models and cleaning the data further.

Chapter 8

Self Reflection

In general, the project was very insightful. All group members gained further insight into how to apply the Bayesian workflow in practice, and how to use BRMS to model an unknown problem. The problem at hand, predicting insurance premiums, sounded very straight-forward initially, though after attempting to apply different models to the problem, we quickly realised it is not as straight-forward as expected. Due to this, we gained additional experience in troubleshooting poor fits of models. Even though none of the models we fit perfectly fit the data, we not only still gained valuable experience but also discovered that insurance premiums, at least at this specific company, cannot be modeled linearly. As is said within academia, "the null result is still a result."

References

[1] Segura-Gisbert, Jorge & lledo benito, Josep & Pavia, Jose. (2023). Dataset of an actual motor vehicle insurance portfolio. 10.21203/rs.3.rs-3631821/v1.

[2] Zwerling C, Peek-Asa C, Whitten PS, Choi SW, Sprince NL, Jones MP. Fatal motor vehicle crashes in rural and urban areas: decomposing rates into contributing factors. Inj Prev. 2005 Feb;11(1):24-8. doi: 10.1136/ip.2004.005959.

Appendix A

Code

```r
#Dependencies
if (!require(plyr)){
  install.packages("plyr")
  library(plyr)
}
if (!require(brms)) {
  install.packages("brms")
  library(brms)
}
if(!require(cmdstanr)){
  install.packages("cmdstanr", repos = c("https://mc-stan.org/r-
    packages/", getOption("repos")))
  library(cmdstanr)
}
if(!require(ggplot2)){
  install.packages("ggplot2")
  library(ggplot2)
}
install.packages("corrplot")
library(corrplot)


#Data loading and cleaning
insurance_data <- read.table("Motor vehicle insurance data.csv", sep =
    ';', header = TRUE)

insurance_data <- na.omit(insurance_data)
insurance_data <- insurance_data[as.Date(insurance_data$Date_start_
   contract, tryFormats = c("%d/%m/%Y")) > as.Date("01/01/2018",
   tryFormats = c("%d/%m/%Y")),]

insurance_data <- insurance_data[as.Date(insurance_data$Date_start_
   contract, tryFormats = c("%d/%m/%Y")) < as.Date("01/01/2019",
   tryFormats = c("%d/%m/%Y")),]

insurance_data <- insurance_data[insurance_data$Type_risk == 3,]


#Pairwise plot
data_pair <- insurance_data_test[c(15, 8, 9, 16, 17, 25, 26)]
area_group <- as.numeric(insurance_data_test[, 28]) + 1
l <- length(unique(area_group))
```

```r
pairs(data_pair, col = c("red", "blue")[area_group], upper.panel=NULL,
    xaxt='n', yaxt='n')


#Linear model
linear_model_formula <- bf(
  Premium ~ 1 + Seniority + Policies_in_force + Cost_claims_year + N_
   claims_year + Value_vehicle,
  family = "gaussian",
  center = FALSE)

get_prior(linear_model_formula, data = insurance_data_test)

(linear_model_priors <- c(
  prior(
    normal(250, 100),
    class = "b",
    coef = "Intercept"
  ),
  prior(
    normal(10, 5),
    class = "b",
    coef = "Seniority"
  ),
  prior(
    normal(1, 1),
    class = "b",
    coef = "Policies_in_force"
  ),
  prior(
    normal(250, 200),
    class = "b",
    coef = "Cost_claims_year"
  ),
  prior(
    normal(1, 1),
    class = "b",
    coef = "N_claims_year"
  ),
  prior(
    normal(20000, 5000),
    class = "b",
    coef = "Value_vehicle"
  )
))

linear_model_fit <- brm(
  formula = linear_model_formula,
  prior = linear_model_priors,
  data = insurance_data,
  cores = parallel::detectCores(),
  threads = threading(16),
)

linear_model_fit <- add_criterion(
  linear_model_fit,
```

```
  criterion = "loo"
)

summary(linear_model_fit)
pp_check(linear_model_fit)
loo(linear_model_fit)


#Sensitivity analysis priors
(linear_model_priors_changed <- c(
  prior(
    normal(100, 10),
    class = "b",
    coef = "Intercept"
  ),
  prior(
    normal(100, 50),
    class = "b",
    coef = "Seniority"
  ),
  prior(
    normal(10, 10),
    class = "b",
    coef = "Policies_in_force"
  ),
  prior(
    normal(10, 10),
    class = "b",
    coef = "Cost_claims_year"
  ),
  prior(
    normal(10, 10),
    class = "b",
    coef = "N_claims_year"
  ),
  prior(
    normal(50, 4),
    class = "b",
    coef = "Value_vehicle"
  )
))


#Sensitivity analysis
linear_model_fit_changed_prior <- brm(
  formula = linear_model_formula,
  prior = linear_model_priors_changed,
  data = insurance_data
)

linear_model_fit_changed_prior <- add_criterion(
  linear_model_fit_changed_prior,
  criterion = "loo"
)

loo_compare(linear_model_fit_changed_prior, linear_model_fit)
```

```r


#Hierarchical model
hierarchical_linear_model_formula <- bf(
  Premium ~ 1 + Seniority + Policies_in_force + Cost_claims_year + N_
    claims_year + Value_vehicle + (1 + Seniority + Policies_in_force +
    Cost_claims_year + N_claims_year + Value_vehicle | Area),
  family = "gaussian",
  center = FALSE)

hierarchical_linear_model_fit <- brm(
  formula = hierarchical_linear_model_formula,
  prior = linear_model_priors,
  data = insurance_data,
  cores = parallelly::availableCores(),
  threads = threading(8),
  control=list(max_treedepth=10),
  backend = "cmdstanr"
)

hierarchical_linear_model_fit <- add_criterion(
  hierarchical_linear_model_fit,
  criterion = "loo"
)

summary(hierarchical_linear_model_fit)
pp_check(hierarchical_linear_model_fit)
loo(hierarchical_linear_model_fit)


#Nonlinear model
nonlinear_model_formula <- bf(
  Premium ~ 1 + s(Seniority) + s(Policies_in_force) + s(Cost_claims_
    year) + s(N_claims_year) + s(Value_vehicle),
  family = "lognormal",
  center = FALSE)

get_prior(nonlinear_model_formula, data = insurance_data)

(nonlinear_model_priors <- c(
  prior(
    normal(250, 100),
    class = "b",
    coef = "Intercept"
  ),
  prior(
    normal(10, 5),
    class = "b",
    coef = "sSeniority_1"
  ),
  prior(
    normal(1, 1),
    class = "b",
    coef = "sPolicies_in_force_1"
  ),
  prior(
```

```
    normal(250, 200),
    class = "b",
    coef = "sCost_claims_year_1"
  ),
  prior(
    normal(1, 1),
    class = "b",
    coef = "sN_claims_year_1"
  ),
  prior(
    normal(20000, 5000),
    class = "b",
    coef = "sValue_vehicle_1"
  )
))

nonlinear_model_fit <- brm(
  formula = nonlinear_model_formula,
  prior = nonlinear_model_priors,
  data = insurance_data,
  family = "lognormal",
  cores = parallel::detectCores(),
  threads = threading(16)
)

nonlinear_model_fit <- add_criterion(
  nonlinear_model_fit,
  criterion = "loo"
)

summary(nonlinear_model_fit)
loo(nonlinear_model_fit)
loo_compare(nonlinear_model_fit, linear_model_fit)


#Sensitivity analysis priors
(nonlinear_model_priors_changed <- c(
  prior(
    normal(100, 10),
    class = "b",
    coef = "Intercept"
  ),
  prior(
    normal(100, 50),
    class = "b",
    coef = "sSeniority_1"
  ),
  prior(
    normal(10, 10),
    class = "b",
    coef = "sPolicies_in_force_1"
  ),
  prior(
    normal(10, 10),
    class = "b",
    coef = "sN_claims_year_1"
```

```
  ),
  prior(
    normal(10, 10),
    class = "b",
    coef = "sCost_claims_year_1"
  ),
  prior(
    normal(200, 50),
    class = "b",
    coef = "sValue_vehicle_1"
  )
))


#Sensitivity analysis
nonlinear_model_fit_changed_priors <- brm(
  formula = nonlinear_model_formula,
  prior = nonlinear_model_priors_changed,
  data = insurance_data,
  family = "lognormal",
  cores = parallel::detectCores(),
  threads = threading(16)
)

nonlinear_model_fit_changed_priors <- add_criterion(
  nonlinear_model_fit_changed_priors,
  criterion = "loo"
)

loo_compare(nonlinear_model_fit_changed_priors, nonlinear_model_fit)


#Model comparison
print("Linear Bayes R2")
bayes_R2(linear_model_fit)
print("Linear Loo R2")
loo_R2(linear_model_fit)

print("Non-linear Bayes R2")
bayes_R2(nonlinear_model_fit)
print("Non-linear Loo R2")
loo_R2(nonlinear_model_fit)

print("Hierarchical Bayes R2")
bayes_R2(hierarchical_linear_model_fit)
print("Hierarchical Loo R2")
loo_R2(hierarchical_linear_model_fit)

loo_compare(linear_model_fit, nonlinear_model_fit, hierarchical_linear
   _model_fit)
```