Digital Intelligence

Task 6: Predictive and Statistical Modelling

Gustavo Boaventura Cruz

Oberönz, BE - 2024

Table of Contents

Information available (a)

We have enough information to fit the data to the collective risk model in both locations (WIS and non-WIS), where we have loss data available over both locations from 2016 to 2020, which are essential for calculating the Poisson distribution (for claim frequency) and Pareto distribution (to claim severity).

The assumptions (b)

We have been provided with assumptions that show us step by step how to claim a better model for the case, also the distribution of parameters like Poisson and Pareto, where it is clear what to do. And in the end, it shows us that we must adjust our date, also available in the loss data.

To validate these models, we examine historical data to determine if there is evidence of correlation or dependence between the two locations. To do this, we examine the Poisson distribution independently, the same for the Pareto distribution, and then we try to check its accuracy and could possibly compare it with estimation methods, for example.

Challenges (c)

In the dataset available for this case, we could see that the data is somewhat limited, just a few years to compare which may not be enough to estimate the parameters and apply machine learning in this case. This means it's still possible to do this, but it might not be necessary. Another challenge that we can observe, the data set provided only shows some information, with approximately 45% of data 0 or NaN, which for this case, it is assumed that this NaN will be equal to zero, where a period has not been reported claims.

Characterizing the dataset of each type of site we have to split them:

- WIS (Waste incinerator stations): As waste incinerator stations, they are more exposed to risks such as machinery problems, and even fire incident risks in the incinerator operation.
- non-WIS (Landfill sites): For landfill sites, may be different from WIS, since they don’t have the same amount of machinery as non-WIS, but they are also exposed to risks, even fire incidents as there may be liquid and gas leaks.

Validate the provided data (d)

To validate this dataset, we have to first analyze data cleaning to see if there is any misinformation or possible errors. But when we analyze the given dataset, it does not have many rows and columns, so it is easy to ensure its integration.

Furthermore, when we work with currencies related to years, it is important to consider the inflection that occurred year after year, in this case it was given that we have to consider a 3% inflection per year, it is clear that we will work with an average inflection of the last five years. With this point, it may affect the validity of the data provided.

Parameter estimation (e)

For this step, we need to open RStudio to code in Rscript. First, you need to enter the loss data into RStudio as a data frame, then set the inflection rate to 1.03 (3%), to inflate the claims data like Jakob's first tip. Then, two functions were created, one for Poisson and the other for Pareto, using Jakob's second tip to calculate, and finally, both were adjusted to the data set, providing the following results:

Table 1. Parameters fitting the claims frequency and severity distributions.

		WIS	Non-WIS
Poisson (frequency)		34.3	14.9
Pareto (severity)	Shape	1.5	1.5
	Location	11.4	5.0

Whit these results, we can understand that the average 34.3 of claims expected per year for WIS sites and 14.9 of claims expected per year for non-WIS sites; 1.5 was the fixed shape assumed; and 11.4 was the minimum value at which claims occur and influences the scale of the Pareto distribution for WIS sites, and 5.0 for non-WIS sites.

Expected Values of the Claims 2021 (f)

We can calculate the expected value of the claims for the current year 2021 using the equation provided by Jakob`s hint. We can expect values be derived analytically, once we calculated the Poisson and Pareto parameters before, we also can use them to stipulate E[N] and E[X], respective:

$$\mathrm{E}[\mathrm{S}] = \mathrm{E}[\mathrm{N}] \times \mathrm{E}[\mathrm{X}] \tag{1}$$

Where: E[S] = Total Claim Amount;

E[N] = Numbers of Claims;

E[X] = Expected Values of Single Claim Amount.

Once we used equation 1 on RStudio, with the parameters mentioned before, we had this result bellow:

Table 2. Expected value of claims for the yar 2021:

WIS	Non-WIS	Aggregate (WIS+Non_WIS)
784.3 mCHF	148.0 mCHF	932.3 mCHF

Value at Risk (VaR) (g)

To stipulate the VaR at the 80% level, we can use Jakob's hint twice (2 e 3), where we used the equation 2 to define Var[X], equation 3 to Var[S] and finally his last hint to define the VaR at 80%:

$$Var[X] = {t^{2*a}}/{(a-2)} \quad (2)$$

Where: t = Scale in Pareto Distribution;

a = Shape in Pareto Distribution.

$$Var[S] = E[N] * Var[X] + Var[N] * E[X]^2 \quad (3)$$

Where: Var[N] = λ Poisson;

Var[S] = Total claim amount.

With these equations, we can calculate the VaR at the 80%, important to add here Jakob's last hint, where the standard normal distribution is 84%:

$$\text{VaR}\ (80\%) = 0.84\sqrt{Var[S]} \quad (4)$$

Table 3. VaR at the 80% level

WIS claims	Non-WIS claims
225.0 mCHF	64.4 mCHF

Quantify the impact on the expected aggregate loss for the year 2021 (h)

To quantify the impact, we can double the expected numbers of claims E[N] and recount to find out this impact. In this way, we found the impact on the expected aggregate loss for year 2021 if the claims frequency for both sites doubles was **932.3 mCHF.**

Modelling Approach (i)

The modelling can be improved in some ways, I would mention some sections to maybe increase the accuracy and the real scenarios as well. So, in my view, the most important this is to increase the to increase the sample size, it can improve the estimations for the claims frequency and severity distributions.

Machine Learning (j)

Machine Learning can be applied in this model. Machine Learning techniques are important to implement not just in insures make but every make in general, it can provide a lot of insights from data, which can help them about pricing, risk management and such more.

To implement Machine Learning in our model, we could use PoissonRegressior as a parameter from sklearn library to claims the frequency model, then we could fit them to our trains variables, then define a model algorithm, then train them and evaluate them.

Appendice (Rscript)

```
loss_data <- data.frame(
  Year = c(2016, 2016, 2016, 2017, 2018, 2018, 2019, 2019, 2020),
  WIS = c(3.1, 2.1, 10.5, 2.0, 0, 0, 230.5, 51.0, 0.5),
  Non_WIS = c(4.5, 0, 0, 0, 125.3, 0, 0.4, 0, 0)
)

inflation_rate <- 1.03

fit_poisson <- function(data) {
  mean_claims <- mean(data) * inflation_rate ^ (2021 - max(loss_data$Year))
  lambda <- mean_claims
  return(lambda)
}

fit_pareto <- function(data) {
  data <- data * inflation_rate ^ (2021 - max(loss_data$Year))
  mean_claims <- mean(data)
  shape <- 1.5
  location <- mean_claims * (shape - 1) / shape
  return(list(shape = shape, location = location))
}

lambda_WIS <- fit_poisson(loss_data$WIS)
```

```
params_WIS <- fit_pareto(loss_data$WIS)

lambda_non_WIS <- fit_poisson(loss_data$Non_WIS)
params_non_WIS <- fit_pareto(loss_data$Non_WIS)

# Results
lambda_WIS #WIS frequency
params_WIS$shape #WIS severity shape
params_WIS$location#WIS severity location

lambda_non_WIS #Non-WIS frequency
params_non_WIS$shape #Non-WIS severity shape
params_non_WIS$location #Non-WIS severity location

E_N_WIS <- lambda_WIS
E_N_non_WIS <- lambda_non_WIS

E_X_WIS <- params_WIS$location / (params_WIS$shape - 1)
E_X_non_WIS <- params_non_WIS$location / (params_non_WIS$shape - 1)

E_S_WIS <- E_N_WIS * E_X_WIS
E_S_non_WIS <- E_N_non_WIS * E_X_non_WIS

E_S_aggregate <- E_S_WIS + E_S_non_WIS

# Results
E_S_WIS #Value WIS in mCHF
E_S_non_WIS #Value Non-WIS in mCHF
E_S_aggregate #Aggregate

Var_N_WIS <- lambda_WIS
```

```
Var_X_WIS <- (params_WIS$location^2 * params_WIS$shape) / ((params_WIS$shape - 2)^2 *
(params_WIS$shape - 1))

Var_S_WIS <- E_N_WIS * Var_X_WIS + Var_N_WIS * E_X_WIS^2


Var_N_non_WIS <- lambda_non_WIS

Var_X_non_WIS <- (params_non_WIS$location^2 * params_non_WIS$shape) /
((params_non_WIS$shape - 2)^2 * (params_non_WIS$shape - 1))

Var_S_non_WIS <- E_N_non_WIS * Var_X_non_WIS + Var_N_non_WIS * E_X_non_WIS^2


VaR_80_WIS <- sqrt(Var_S_WIS) * 0.84

VaR_80_non_WIS <- sqrt(Var_S_non_WIS) * 0.84


#Results

VaR_80_WIS

VaR_80_non_WIS


E_N_WIS_doubled <- 2 * E_N_WIS

E_N_non_WIS_doubled <- 2 * E_N_non_WIS


E_S_WIS_doubled <- E_N_WIS_doubled * E_X_WIS

E_S_non_WIS_doubled <- E_N_non_WIS_doubled * E_X_non_WIS


E_S_aggregate_doubled <- E_S_WIS_doubled + E_S_non_WIS_doubled


impact_on_expected_loss <- E_S_aggregate_doubled - E_S_aggregate

# Results

impact_on_expected_loss
```

~~You can access the RScript clicking here.~~