Monte Carlo Simulation for Capital Modeling

SNHU: DAT 610 Optimization and Risk assessment

Rachel Goldsbury

Instructor: Kyle Camac

September 22nd, 2024

Introduction

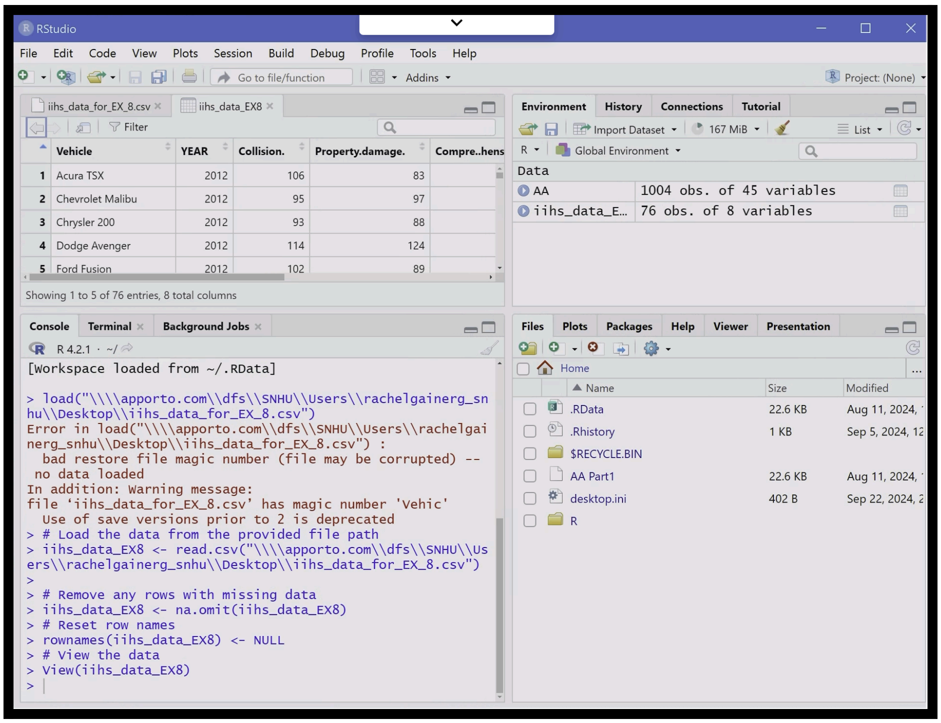
The purpose of this exercise is to execute a Monte Carlo simulation to estimate maximum annual collision loss using the IIHS data. A Monte Carlo simulation is a statistical method used to model and predict the probability of various outcomes based on random sampling. In capital modeling, it is employed to estimate potential future losses with high confidence by simulating numerous scenarios. By generating additional data points from known frequency and severity distributions, the simulation helps organizations estimate capital requirements to cover losses with 99.9% certainty, as outlined in Chapter 12 of Operational Risk Management (Philippa X. Girling, 2024). This method allows for better decision-making in managing financial risks.

Data Loading and Preparation

The read.csv() function was used to successfully load the data from the specified file path. Next, the na.omit() function was applied to remove any rows with missing data, ensuring that incomplete data does not cause errors or inaccuracies in subsequent calculations. The row names were reset using rownames() to maintain proper indexing, and then the View() function was used to visually inspect the dataset within RStudio. The R Code is included below.

```
>
load("\\\\apporto.com\\dfs\\SNHU\\Users\\rachelgainerg_snhu\\Desktop\\iihs_data_for_EX_8.csv
")
Error in
load("\\\\apporto.com\\dfs\\SNHU\\Users\\rachelgainerg_snhu\\Desktop\\iihs_data_for_EX_8.csv
") :
  bad restore file magic number (file may be corrupted) -- no data loaded
In addition: Warning message:
file ‘iihs_data_for_EX_8.csv’ has magic number 'Vehic'
  Use of save versions prior to 2 is deprecated
> # Load the data from the provided file path
```

```
> iihs_data_EX8 <-
read.csv("\\\\apporto.com\\dfs\\SNHU\\Users\\rachelgainerg_snhu\\Desktop\\iihs_data_for_EX_8
.csv")
>
> # Remove any rows with missing data
> iihs_data_EX8 <- na.omit(iihs_data_EX8)
> # Reset row names
> rownames(iihs_data_EX8) <- NULL
> # View the data
> View(iihs_data_EX8)
```



(Figure 1: Data Loading and Preparation)

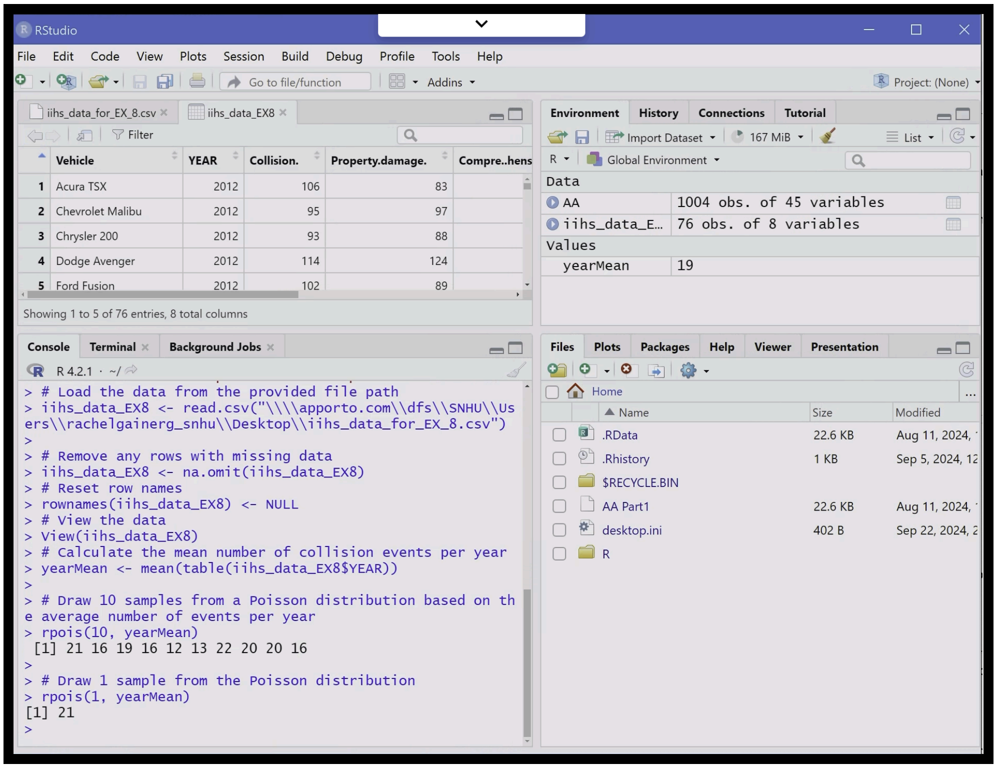
Average Loss Frequency (Poisson Distribution)

The Poisson distribution is used to estimate the number of collision loss events per year by modeling the frequency of events that occur randomly and independently within a fixed time period, such as a year. It allows for predicting the likelihood of different numbers of loss events based on an average rate of occurrence.

We first calculate the mean number of collision events per year (`yearMean`) by averaging the frequency of events across all years in the dataset. The `rpois()` function is then used to draw random samples from a Poisson distribution with the mean set to `yearMean`. First, 10 random samples are drawn, simulating possible yearly collision events, followed by drawing 1 additional sample representing a single year's predicted number of collision events. The Poisson distribution models how often these events occur based on the average rate of occurrence. (Philippa X. Girling, 2024)

The 10 samples generated ([21,16,19,16,12,13,22,20, 20,16]) represent the predicted number of collision events for different years, while the single sample ([21]) shows the expected number of events for one year. Find the code used below.

```
> # Calculate the mean number of collision events per year
> yearMean <- mean(table(iihs_data_EX8$YEAR))
>
> # Draw 10 samples from a Poisson distribution based on the average number of events per year
> rpois(10, yearMean)
 [1] 21 16 19 16 12 13 22 20 20 16
>
> # Draw 1 sample from the Poisson distribution
> rpois(1, yearMean)
[1] 21
```



(Figure 2: Average Loss Frequency (Poisian Distribution))

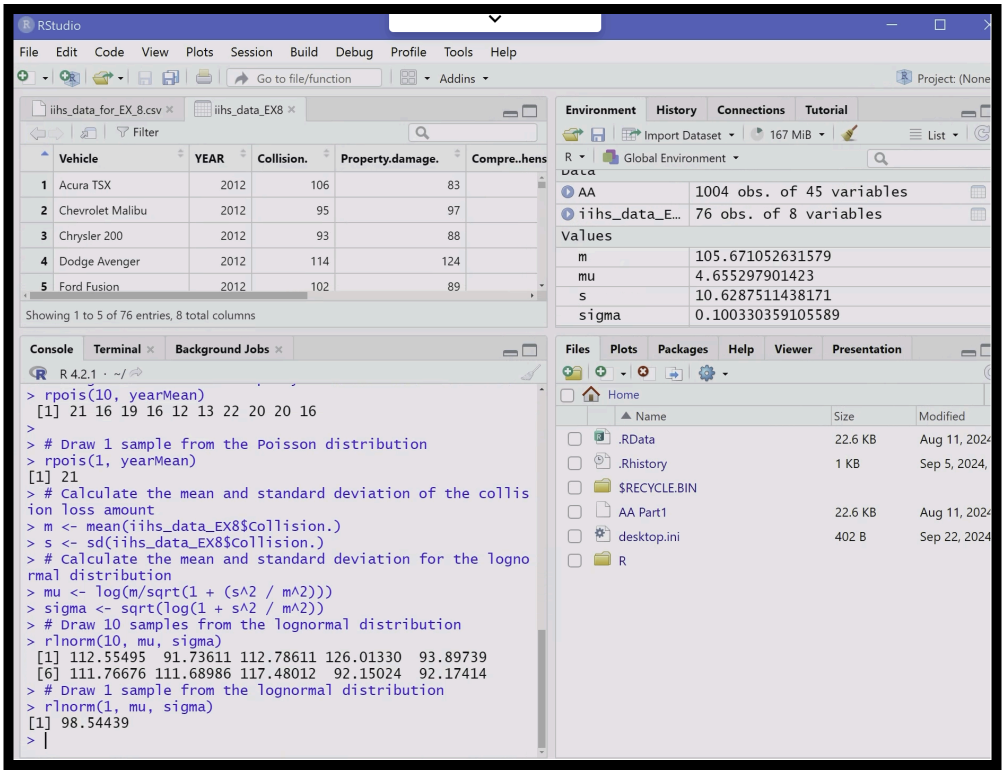
Average Loss Severity (Lognormal Distribution)

The lognormal distribution is used to model the size of collision losses because it captures the skewed nature of loss amounts, where most losses are small and only a few large losses can occur. This distribution allows for realistic representation of the variability in loss severity, ensuring that higher probabilities are assigned to smaller losses while still accounting for rare, large loss events.

In this R code below, the mean (`m`) and standard deviation (`s`) of the collision loss amounts are calculated from the dataset. These values are then used to determine the parameters (`mu` and `sigma`) for the lognormal distribution, which models the size of the collision losses. The lognormal distribution is chosen because it can represent skewed data where most losses are small, but a few large losses may occur. The function `rlnorm(10, mu, sigma)` generates 10 random samples from this distribution, simulating potential collision loss amounts, while `rlnorm(1, mu, sigma)` produces one random sample representing a single event. The generated samples reflect the variation in loss amounts due to the nature of the lognormal distribution.

```
> # Calculate the mean and standard deviation of the collision loss amount
> m <- mean(iihs_data_EX8$Collision.)
> s <- sd(iihs_data_EX8$Collision.)
> # Calculate the mean and standard deviation for the lognormal distribution
> mu <- log(m/sqrt(1 + (s^2 / m^2)))
> sigma <- sqrt(log(1 + s^2 / m^2))
> # Draw 10 samples from the lognormal distribution
> rlnorm(10, mu, sigma)
 [1] 112.55495  91.73611 112.78611 126.01330  93.89739
 [6] 111.76676 111.68986 117.48012  92.15024  92.17414
> # Draw 1 sample from the lognormal distribution
> rlnorm(1, mu, sigma)
[1] 98.54439
```

In the results of this code, the mean (`m = 105.67`) and standard deviation (`s = 10.63`) of the collision loss amounts were calculated from the dataset. These values are then used to determine the parameters for the lognormal distribution: `mu` (the mean of the lognormal distribution) and `sigma` (the standard deviation of the lognormal distribution). The function `rlnorm(10, mu, sigma)` generated 10 random samples of collision loss amounts, with values ranging between approximately 91.7 and 126.0. These values simulate the potential collision loss amounts based on the distribution of past losses. The subsequent function `rlnorm(1, mu, sigma)` generated a single random sample, representing a specific potential loss amount, which is approximately 98.54 in this instance. This sampling illustrates the variability in potential losses, with most values close to the mean but some variation due to the lognormal distribution's shape.



(Figure 3: Average Loss Severity (Lognormal Distribution))

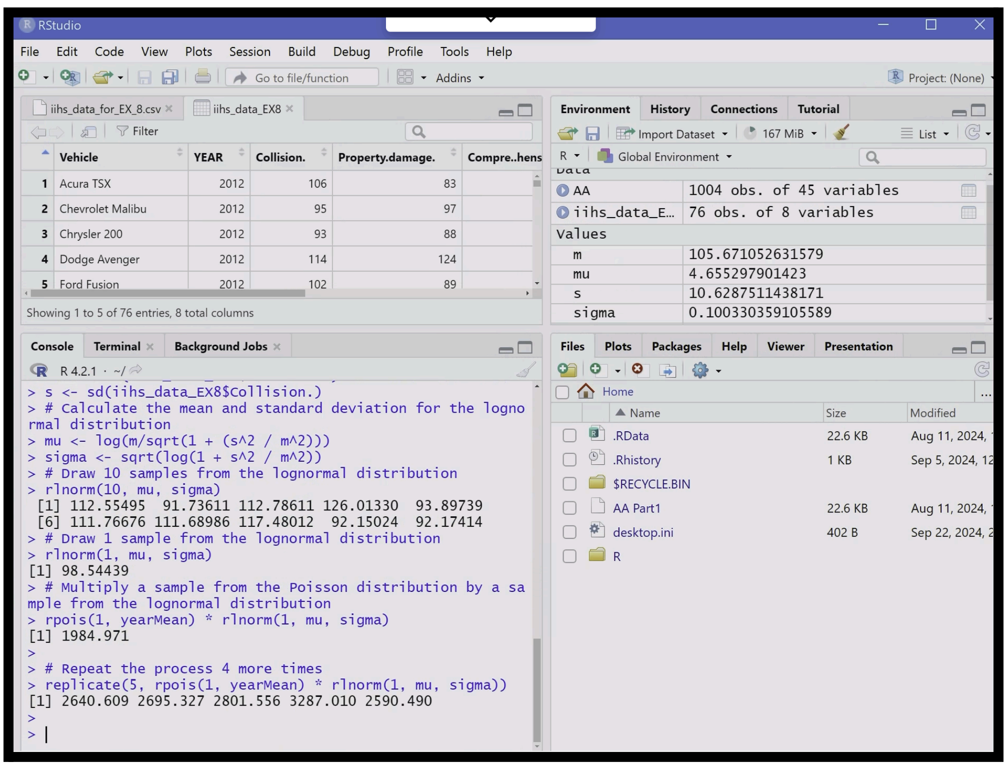
Maximum Total Loss (Monte Carlo Approximation)

The code below, calculates the total loss for a simulated year by multiplying the frequency of collision loss events, drawn from a Poisson distribution, by the severity of those losses, drawn from a lognormal distribution. The first line, `rpois(1, yearMean) rlnorm(1, mu, sigma)`, generates a single total loss value of approximately 1984.971 by combining one random sample of frequency with one random sample of severity. Subsequently, the `replicate(5, rpois(1, yearMean) rlnorm(1, mu, sigma))` function repeats this process five times, producing a range of total loss values: `[2640.609, 2695.327, 2801.556, 3287.010, 2590.490]`.

```
> # Multiply a sample from the Poisson distribution by a sample from the lognormal distribution
> rpois(1, yearMean) * rlnorm(1, mu, sigma)
[1] 1984.971
>
```

```
> # Repeat the process 4 more times
> replicate(5, rpois(1, yearMean) * rlnorm(1, mu, sigma))
[1] 2640.609 2695.327 2801.556 3287.010 2590.490
```

The results illustrate the variability in estimated total annual collision losses derived from frequency and severity distributions. The first calculation shows a total loss of approximately 1984.971 for a simulated year, while the subsequent five values—2640.609, 2695.327, 2801.556, 3287.010, and 2590.490—demonstrate different scenarios based on unique combinations of drawn frequency and severity values. This variation highlights the significance of using Monte Carlo simulations to capture the uncertainty in financial forecasting of collision losses.



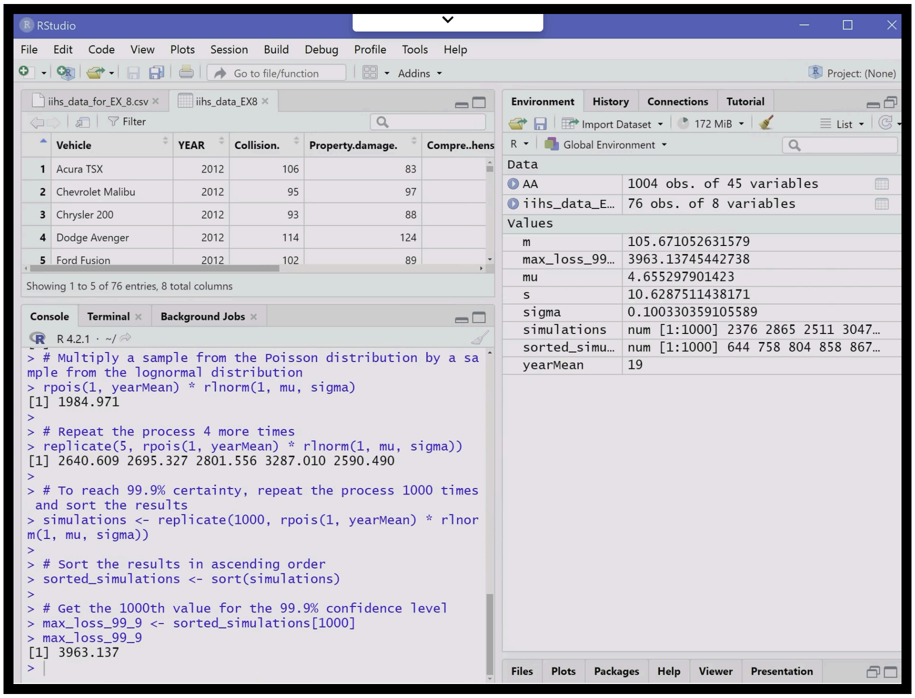
(Figure 4: Monte Carlo Approximation)

Monte Carlo Extrapolation Process

Monte Carlo simulations can be extended to 1000 iterations to estimate maximum annual losses with 99.9% confidence by repeatedly sampling from the frequency and severity distributions. After running the simulation, the 1000th value in the sorted results represents the maximum loss expected with 99.9% certainty, effectively capturing the variability in loss events for robust risk analysis.

In the results, the value of approximately 3963.137 represents the maximum estimated annual collision loss at a 99.9% confidence level. This means that there is only a 0.1% chance that the actual collision loss will exceed this amount in any given year. This figure is crucial for financial planning and risk management, as it helps organizations determine the capital needed to cover potential extreme losses. The code for this is below.

```
> # To reach 99.9% certainty, repeat the process 1000 times and sort the results
> simulations <- replicate(1000, rpois(1, yearMean) * rlnorm(1, mu, sigma))
>
> # Sort the results in ascending order
> sorted_simulations <- sort(simulations)
>
> # Get the 1000th value for the 99.9% confidence level
> max_loss_99_9 <- sorted_simulations[1000]
> max_loss_99_9
[1] 3963.137
```



(Figure 5: Monte Carlo Extrapolation Process)

The Monte Carlo simulation can be effectively applied in various real-world scenarios, like estimating losses in insurance claims and assessing credit risk. In insurance, the simulation can model the frequency and severity of claims by drawing from historical data, allowing insurers to predict potential losses over time (Philippa X. Girling, 2024). This helps in determining appropriate premium rates and reserves. Similarly, in credit risk assessment, Monte Carlo simulations can evaluate the likelihood of borrower defaults by simulating various economic scenarios and their impacts on a portfolio of loans. This enables financial institutions to estimate potential losses and make informed decisions regarding lending practices and risk management strategies.

Conclusion

In this exercise, I executed a Monte Carlo simulation to estimate the maximum annual collision loss using IIHS data, achieving a maximum loss value of approximately 3963.137 at the 99.9% confidence level. This finding underscores the significance of employing Monte Carlo simulations in operational risk modeling, as it allows organizations to effectively capture uncertainty and variability in potential losses. By combining frequency and severity distributions, the simulation not only aids in predicting extreme loss events but also informs financial planning and risk management strategies. Overall, the results demonstrate the practical application of Monte Carlo simulations in various domains, such as insurance and credit risk assessment, providing valuable insights for informed decision-making.

References

1. *CHAPTER 12 Capital Modeling*. Vitalsource Bookshelf Online. (n.d.). https://mbsdirect.vitalsource.com/reader/books/9781119836056/epubcfi/6/40[%3Bvnd.vst.idref%3DAc12]!/4
2. Module 8 Capital Modeling. (n.d.-b). https://learn.snhu.edu/d2l/le/news/1678567/7539515/view
3. YouTube. (n.d.). *0:01 /1:34 Loading Data Into R Software - (read.table, Data/CSV Import Tutorial)*. YouTube. https://www.youtube.com/watch?v=VLtazaiYo-c