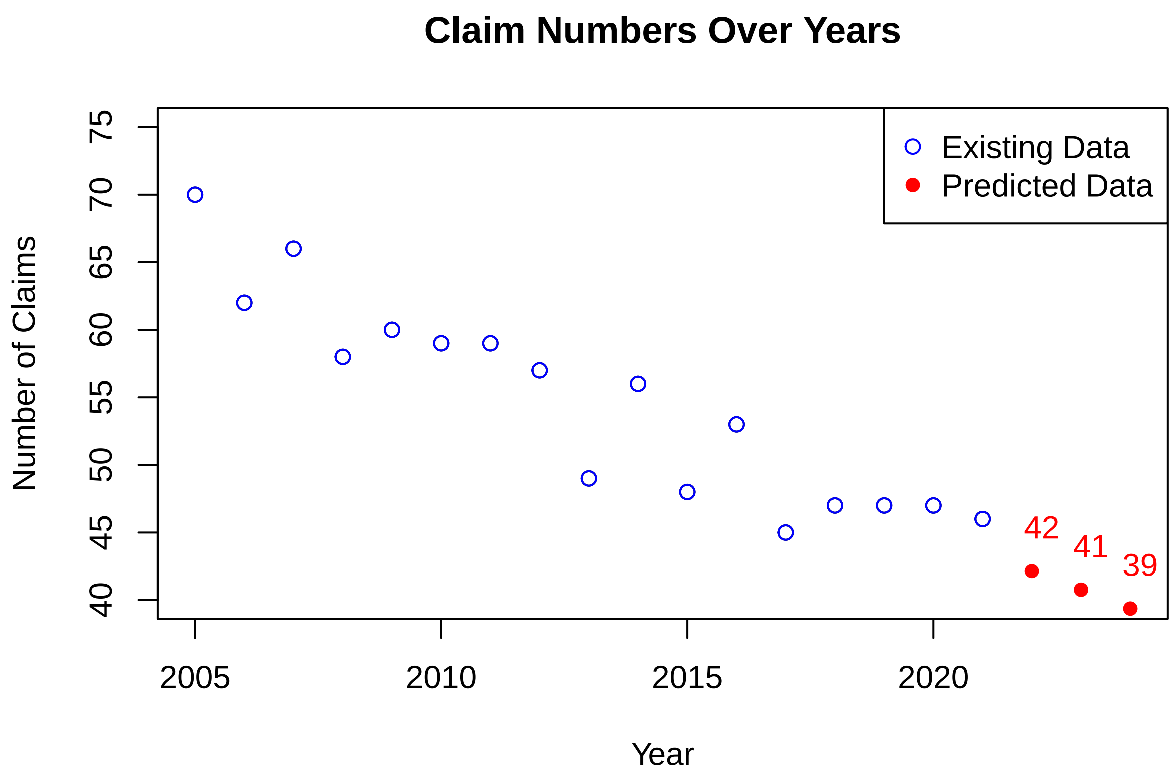
Monte_Carlo_Simulation_for_Aggregate_Claims

2023-11-26

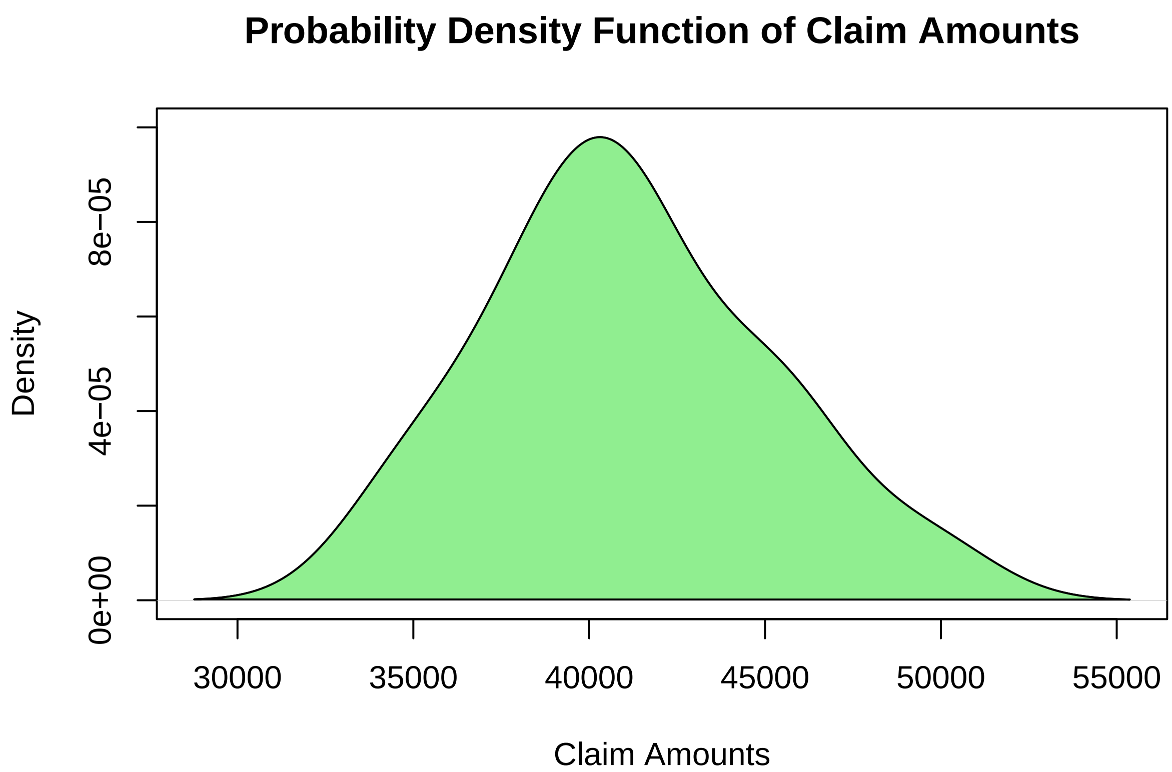
Predicting future number of claims of dwelling fires with linear regression

```r
# Historical data
years <- c(2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016,
           2017, 2018, 2019, 2020, 2021)
claims <- c(70, 62, 66, 58, 60, 59, 59, 57, 49, 56, 48, 53, 45, 47, 47, 47, 46)

# Perform linear regression
model <- lm(claims ~ years)

# Predict future claim numbers for specified years
future_years <- c(2022, 2023, 2024)
predicted_claims <- predict(model, data.frame(years = future_years))

# Create a plot
plot(years, claims, xlim = c(2005, 2024), ylim = c(min(claims) - 5, max(claims) + 5),
     xlab = "Year", ylab = "Number of Claims", main = "Claim Numbers Over Years")

# existing data points
points(years, claims, col = "blue")

# predicted points
points(future_years, predicted_claims, col = "red", pch = 16)

# Display predicted numbers as text on the graph (rounded to 0 decimal places)
text(future_years + 0.2, predicted_claims + 1, labels = format(round(predicted_claims),
                                                   nsmall = 0), pos = 3, col = "red")

# Add legend
legend("topright", legend = c("Existing Data", "Predicted Data"),
       col = c("blue", "red"), pch = c(1, 16))
```



Showing that our claim size amounts due to dwelling fires has distribution of lognormal

```r
# Claim size data
claim_amounts <- c(38620, 40767, 45258, 43618, 35214, 49657, 41098, 37740, 34485, 37459,
                   39650, 41778, 46341, 39539, 41858, 40466,45346)

# Plotting the density function
plot_title <- "Probability Density Function of Claim Amounts"
density_plot <- density(claim_amounts)
plot(density_plot, main = plot_title, xlab = "Claim Amounts", ylab = "Density",
     ylim = c(0, 0.00010))

# Coloring the plot area under the density curve
polygon(density_plot$x, density_plot$y, col = "light green")
```




```r
mean(claim_amounts)
```

```
## [1] 41111.41
```

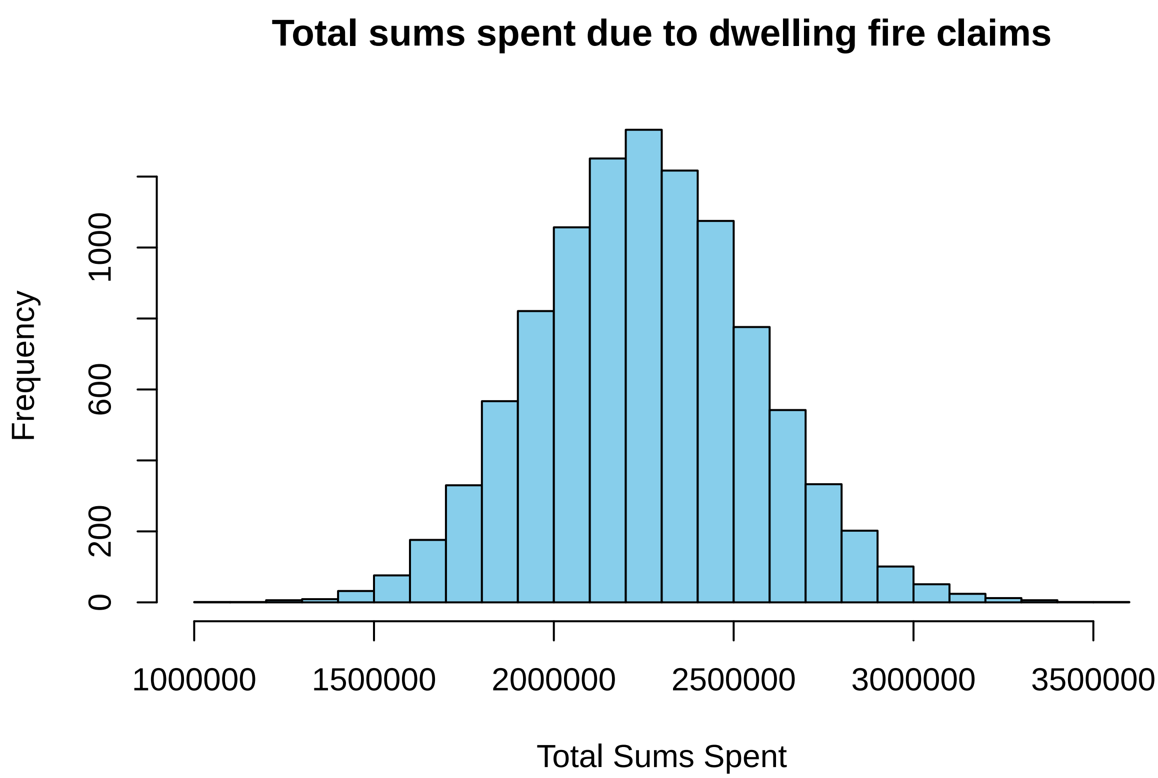
```r
median(claim_amounts)
```

```
## [1] 40767
```

##This is Monte Carlo-Simulation of dwelling fires total sum of payment by the insurance company.

```r
set.seed(123) # For reproducibility of results

# Function to simulate N (number of claims) based on Poisson distribution

simulate_N <- function(num_simulations) {
  lambda <- 55  # Adjust lambda parameter for Poisson distribution
  return(rpois(num_simulations, lambda))
}

# Function to simulate Xi (size of claims) based on lognormal distribution
simulate_Xi <- function(num_simulations) {
  mu <- 10.62  # Adjust mu parameter for lognormal distribution
  sigma <- 0.09  # Adjust sigma parameter for lognormal distribution
  return(rlnorm(num_simulations, meanlog = mu, sdlog = sigma))
}

# Function to estimate the total sum spent (S) using Monte Carlo simulation
simulate_total_sum <- function(num_simulations_N, num_simulations_Xi) {
  # Simulate N (number of claims)
  num_claims <- simulate_N(num_simulations_N)
```

```
  # Simulate Xi (size of claims)
  claim_sizes <- simulate_Xi(num_simulations_Xi)

  # Calculate the total sum spent (S) for each simulation
  total_sum_spent <- sapply(1:length(num_claims), function(i) sum(simulate_Xi(num_claims[i])))
  return(total_sum_spent)
}

# Perform simulations
num_simulations_N <- 10000  # Number of simulations for number of claims
num_simulations_Xi <- 10000  # Number of simulations for size of claims

# Perform the Monte Carlo simulation
Total_sum_dwelling_fires <- simulate_total_sum(num_simulations_N, num_simulations_Xi)

# Create a histogram to visualize the distribution of total sums spent
hist(Total_sum_dwelling_fires, breaks = 20, col = "skyblue",
     main = "Total sums spent due to dwelling fire claims", xlab = "Total Sums Spent")
```




```
mean(Total_sum_dwelling_fires)
```

```
## [1] 2255427
```

```
median(Total_sum_dwelling_fires)
```

```
## [1] 2249899
```

```
var(Total_sum_dwelling_fires)
```

```
## [1] 92100513306
```

```r
sd(Total_sum_dwelling_fires)
```

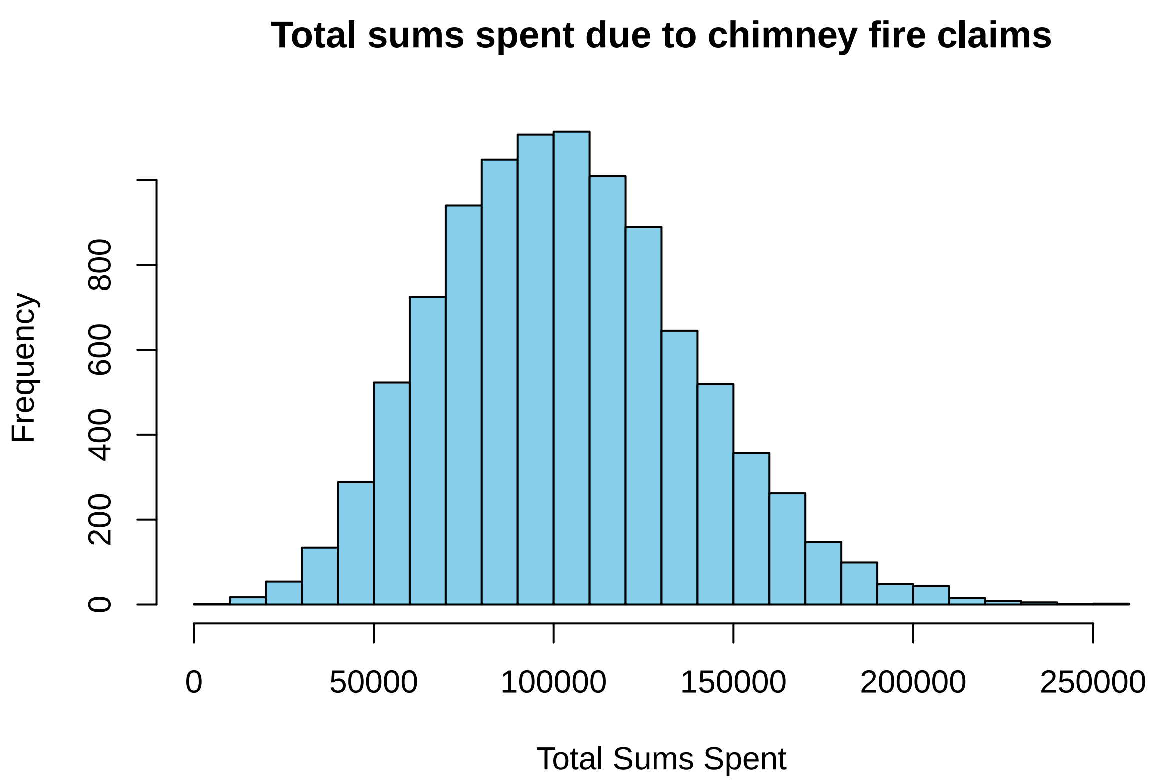
```
## [1] 303480.7
```

```r
# Calculate the 95th percentile
percentile_95 = quantile(Total_sum_dwelling_fires, probs = 0.95, na.rm = TRUE)

print(percentile_95)
```

```
##     95%
## 2762467
```

##This is Monte Carlo-Simulation of chimney fires total sum of payment by the insurance company.

```r
set.seed(4321) # For reproducibility of results

# Function to simulate N (number of claims) based on Negative-binomial distribution

simulate_N <- function(num_simulations) {
  size <- 121/3   # Adjust size parameter for Negative-binomial distribution
  prob <- 11/14    # Adjust probability parameter for Negative-binomial distribution
  return(rnbinom(num_simulations, size = size, prob = prob))
}

# Function to simulate Xi (size of claims) based on lognormal distribution
simulate_Xi <- function(num_simulations) {
  mu <- 9.14  # Adjust mu parameter for lognormal distribution
  sigma <- 0.15  # Adjust sigma parameter for lognormal distribution
  return(rlnorm(num_simulations, meanlog = mu, sdlog = sigma))
}

# Function to estimate the total sum spent (S) using Monte Carlo simulation
simulate_total_sum <- function(num_simulations_N, num_simulations_Xi) {
  # Simulate N (number of claims)
  num_claims <- simulate_N(num_simulations_N)

  # Simulate Xi (size of claims)
  claim_sizes <- simulate_Xi(num_simulations_Xi)

  # Calculate the total sum spent (S) for each simulation
  total_sum_spent <- sapply(1:length(num_claims), function(i) sum(simulate_Xi(num_claims[i])))
  return(total_sum_spent)
}

# Perform simulations
num_simulations_N <- 10000  # Number of simulations for number of claims
num_simulations_Xi <- 10000  # Number of simulations for size of claims

# Perform the Monte Carlo simulation
Total_sum_chimney_fires <- simulate_total_sum(num_simulations_N, num_simulations_Xi)

# Create a histogram to visualize the distribution of total sums spent
hist(Total_sum_chimney_fires, breaks = 20, col = "skyblue",
     main = "Total sums spent due to chimney fire claims", xlab = "Total Sums Spent")
```




```r
mean(Total_sum_chimney_fires)
```

```
## [1] 103413.3
```

```r
median(Total_sum_chimney_fires)
```

```
## [1] 101358.6
```

```r
var(Total_sum_chimney_fires)
```

```
## [1] 1219285385
```

```r
sd(Total_sum_chimney_fires)
```

```
## [1] 34918.27
```

```r
# Calculate the 95th percentile
percentile_95 = quantile(Total_sum_chimney_fires, probs = 0.95, na.rm = TRUE)

print(percentile_95)
```

```
##      95%
## 164556.6
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.