Code

R Notebook

Jaechul Kim

2023-02-17

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Linear regression is a statistical method that is used to establish a relationship between a dependent variable and one or more independent variables.

required library

Hide

```
install.packages("ggplot2")
```

```
WARNING: Rtools is required to build R packages but is not currently installed. Please downlo
ad and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
‘C:/Users/leewq/AppData/Local/R/win-library/4.2’의 위치에 패키지(들)을 설치합니다.
(왜냐하면 ‘lib’가 지정되지 않았기 때문입니다)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/ggplot2_3.4.1.zip'
Content type 'application/zip' length 4226907 bytes (4.0 MB)
downloaded 4.0 MB
```

```
패키지 ‘ggplot2’를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다
    C:\Users\leewq\AppData\Local\Temp\RtmpQDz5mP\downloaded_packages
```

Hide

```
install.packages("dplyr")
```

```
WARNING: Rtools is required to build R packages but is not currently installed. Please downlo
ad and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
'C:/Users/leewq/AppData/Local/R/win-library/4.2'의 위치에 패키지(들)을 설치합니다.
(왜냐하면 'lib'가 지정되지 않았기 때문입니다)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/dplyr_1.1.0.zip'
Content type 'application/zip' length 1541927 bytes (1.5 MB)
downloaded 1.5 MB
```

```
패키지 'dplyr'를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다
    C:\Users\leewq\AppData\Local\Temp\RtmpQDz5mP\downloaded_packages
```

Hide

```
install.packages("psych")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
library(ggplot2)
library(dplyr)
```

```

다음의 패키지를 부착합니다: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

read file and make train and test data

Hide

```
# Set the working directory to the folder where the CSV file is located
setwd("C:\\Users\\leewq\\Downloads\\archive")



# Read the CSV file into a data frame
df <- read.csv("CustomerInfo.csv")

# Remove rows with missing values
df <- na.omit(df)

# Set seed for reproducibility
set.seed(123)

# Determine row indices for training and testing sets
train_indices <- sample(1:nrow(df), 0.8*nrow(df), replace = FALSE)
test_indices <- setdiff(1:nrow(df), train_indices)

# Create training and testing sets
train <- df[train_indices, ]
test <- df[test_indices, ]
```

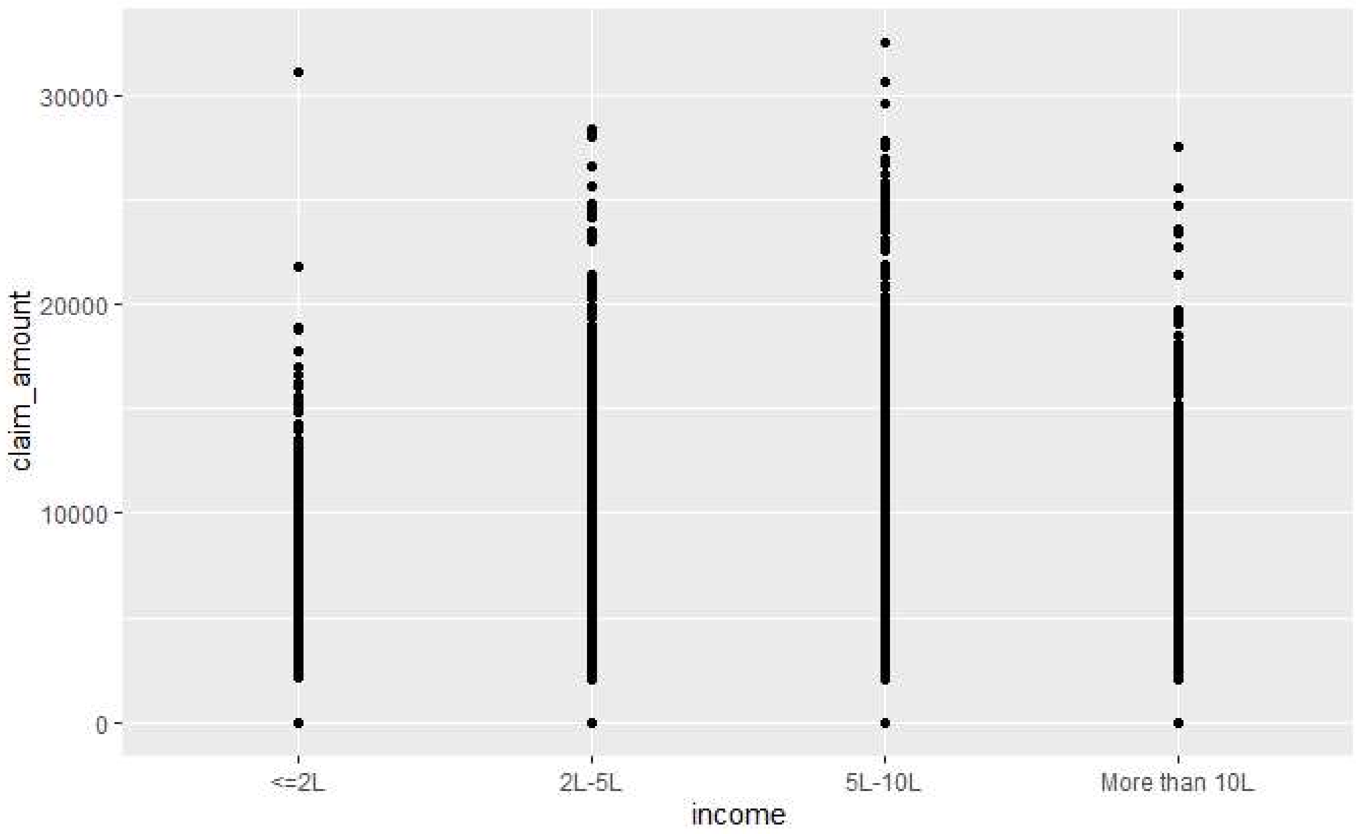
graph The first plot is a scatter plot that shows the relationship between the index (x-axis) and the hourly demand of energy (y-axis). The second plot is a histogram that shows the distribution of the hourly demand of energy.

Hide

```
# Create scatter plot of predictor against target variable
ggplot(train, aes(x = income, y = claim_amount)) +
  geom_point() +
  xlab("income") +
  ylab("claim_amount")
install.packages("psych")
```

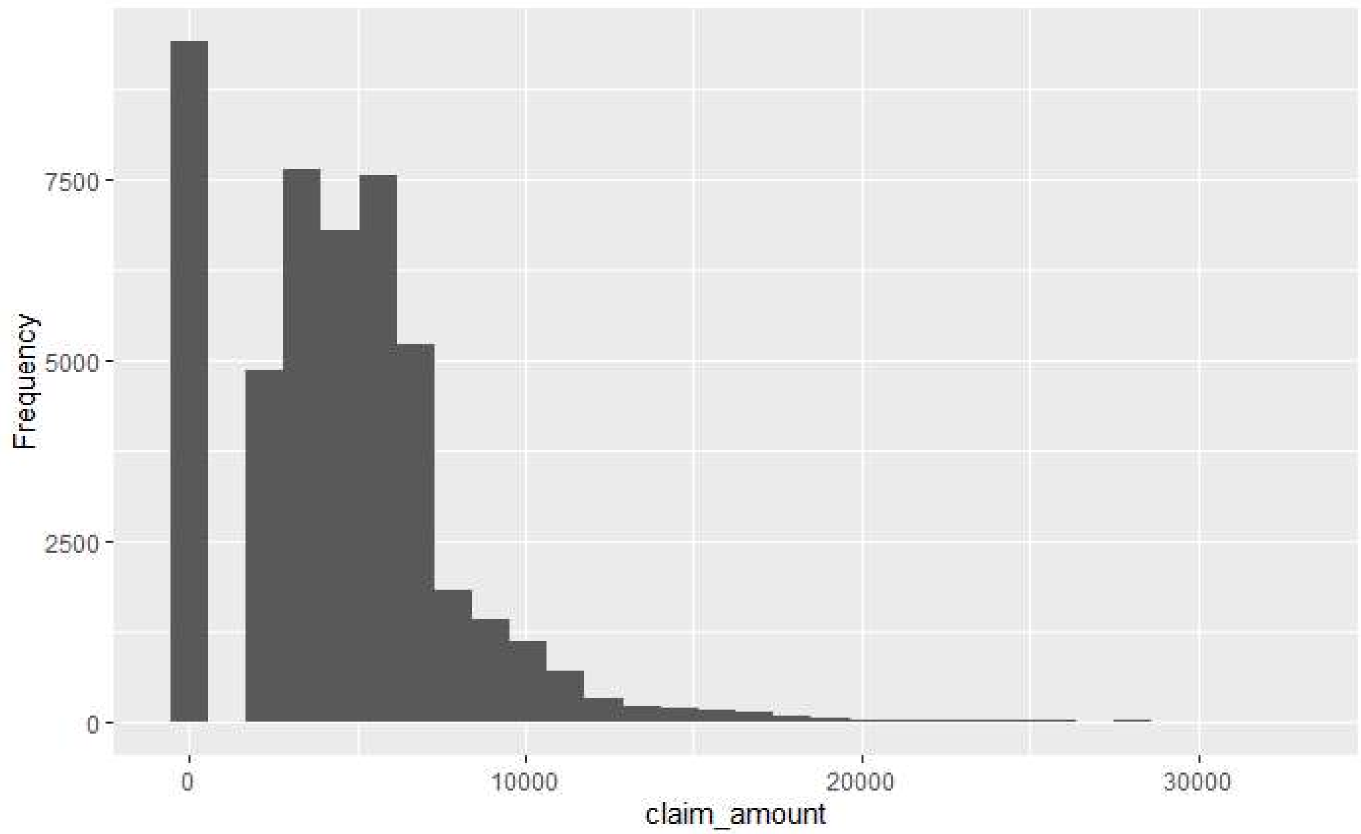
```
WARNING: Rtools is required to build R packages but is not currently installed. Please downlo
ad and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/psych_2.2.9.zip'
Content type 'application/zip' length 3821017 bytes (3.6 MB)
downloaded 3.6 MB
```

```
패키지 ‘psych’를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다
    C:\Users\leewq\AppData\Local\Temp\RtmpQDz5mP\downloaded_packages
```



Hide

```
# Create histogram of target variable
ggplot(train, aes(x = claim_amount)) +
  geom_histogram() +
  xlab("claim_amount") +
  ylab("Frequency")
```



build simple linear regression model

Hide

```
# loading psych package
library(psych)
```

```
다음의 패키지를 부착합니다: ‘psych’

The following objects are masked from ‘package:ggplot2’:

    %+%, alpha
```

Hide

```
psych::describe(train)
```

	v... <dbl>	n <dbl>	mean <dbl>	sd <dbl>	med... <dbl>	trimmed <dbl>	mad <dbl>	min <dbl>	max <dbl>
id	1	47676	119238.78	17206.92	119213	119241.99	22103.34	89394	148987
gender*	2	47676	1.57	0.50	2	1.58	0.00	1	2
area*	3	47676	1.70	0.46	2	1.74	0.00	1	2
qualification*	4	47676	1.60	0.57	2	1.57	0.00	1	3
income*	5	47676	2.88	0.68	3	2.87	0.00	1	4
marital_status	6	47676	0.58	0.49	1	0.60	0.00	0	1
vintage	7	47676	4.62	2.28	5	4.72	2.97	0	8
claim_amount	8	47676	4374.10	3292.76	4101	4089.43	2763.57	0	32534
num_policies*	9	47676	1.68	0.47	2	1.72	0.00	1	2
policy*	10	47676	1.45	0.65	1	1.33	0.00	1	3

1-10 of 11 rows | 1-10 of 13 columns Previous **1** 2 Next

Hide

```
summary(train)
```

```
       id             gender              area          qualification          income         
 Min.   : 89394   Length:47676       Length:47676       Length:47676       Length:47676      
 1st Qu.:104320   Class :character   Class :character   Class :character   Class :character  
 Median :119213   Mode  :character   Mode  :character   Mode  :character   Mode  :character  
 Mean   :119239                                                                              
 3rd Qu.:134135                                                                              
 Max.   :148987                                                                              
 marital_status      vintage       claim_amount   num_policies          policy         
 Min.   :0.0000   Min.   :0.000   Min.   :    0   Length:47676       Length:47676      
 1st Qu.:0.0000   1st Qu.:3.000   1st Qu.: 2402   Class :character   Class :character  
 Median :1.0000   Median :5.000   Median : 4101   Mode  :character   Mode  :character  
 Mean   :0.5785   Mean   :4.617   Mean   : 4374                                        
 3rd Qu.:1.0000   3rd Qu.:6.000   3rd Qu.: 6106                                        
 Max.   :1.0000   Max.   :8.000   Max.   :32534                                        
 type_of_policy    
 Length:47676      
 Class :character  
 Mode  :character  
```

Hide

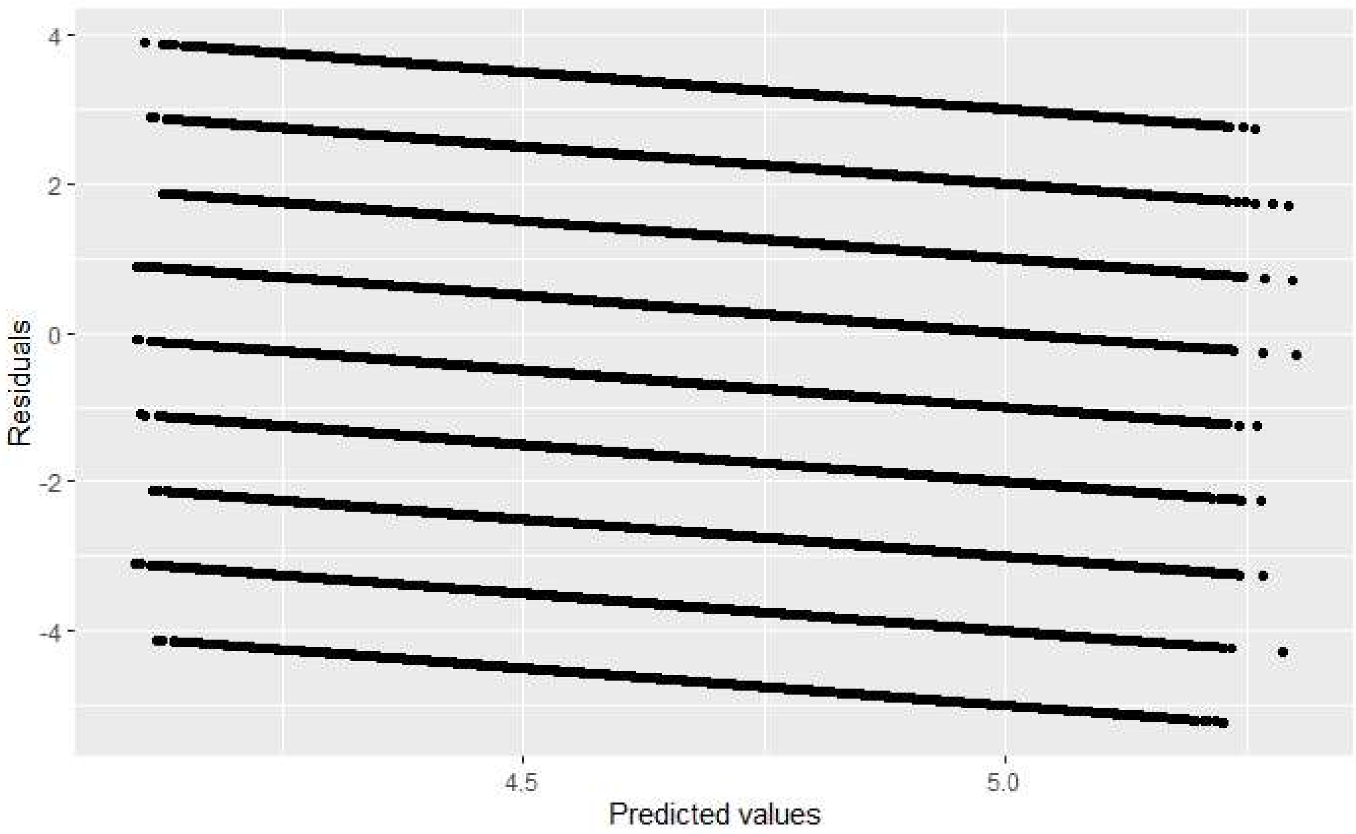
```
# Taining model
lmModel <- lm(vintage ~   . , data = train)
summary(lmModel)
```

```
Call:
lm(formula = vintage ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-5.2253 -1.7017  0.4317  1.7263  3.8930

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)               4.258e+00  1.096e-01  38.837  < 2e-16 ***
id                        9.247e-07  6.031e-07   1.533  0.12525
genderMale                5.840e-02  2.114e-02   2.763  0.00573 **
areaUrban                 2.085e-02  2.609e-02   0.799  0.42433
qualificationHigh School -1.143e-01  2.140e-02  -5.342 9.24e-08 ***
qualificationOthers      -4.594e-02  5.418e-02  -0.848  0.39650
income2L-5L               1.979e-02  7.366e-02   0.269  0.78823
income5L-10L             -4.831e-02  7.215e-02  -0.670  0.50308
incomeMore than 10L      -8.045e-02  7.614e-02  -1.057  0.29070
marital_status           -1.674e-02  2.131e-02  -0.786  0.43195
claim_amount              4.726e-06  3.621e-06   1.305  0.19185
num_policiesMore than 1   2.757e-01  2.274e-02  12.124  < 2e-16 ***
policyB                   4.459e-01  2.471e-02  18.045  < 2e-16 ***
policyC                  -1.522e-02  3.741e-02  -0.407  0.68421
type_of_policyPlatinum   -3.087e-02  2.601e-02  -1.187  0.23524
type_of_policySilver     -1.637e-02  3.075e-02  -0.532  0.59446
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.266 on 47660 degrees of freedom
Multiple R-squared:  0.01262,   Adjusted R-squared:  0.01231
F-statistic: 40.61 on 15 and 47660 DF,  p-value: < 2.2e-16
```

plot residuals

Hide

```
# Plot the residuals
ggplot(train, aes(x = predict(lmModel), y = residuals(lmModel))) +
  geom_point() +
  xlab("Predicted values") +
  ylab("Residuals")
```



multiple predictors and residual plot

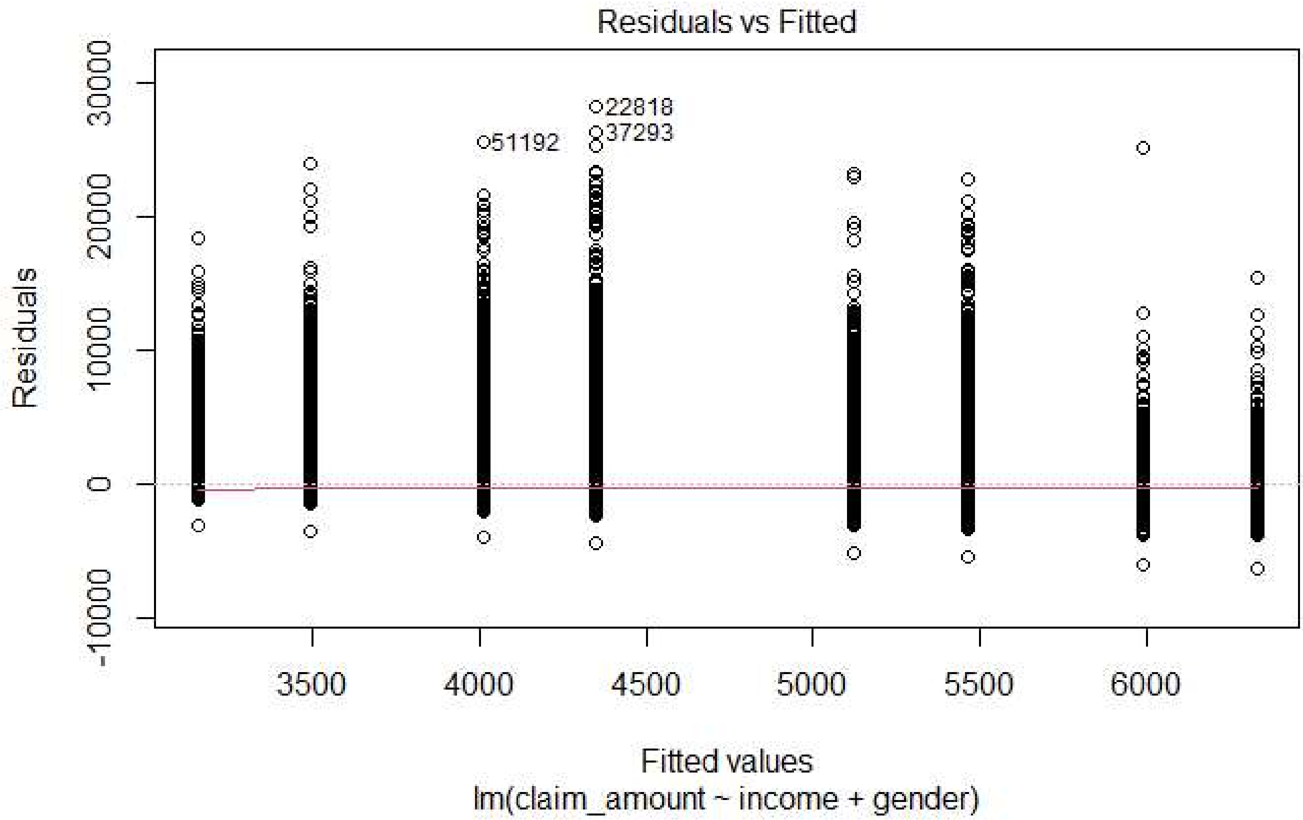
Hide

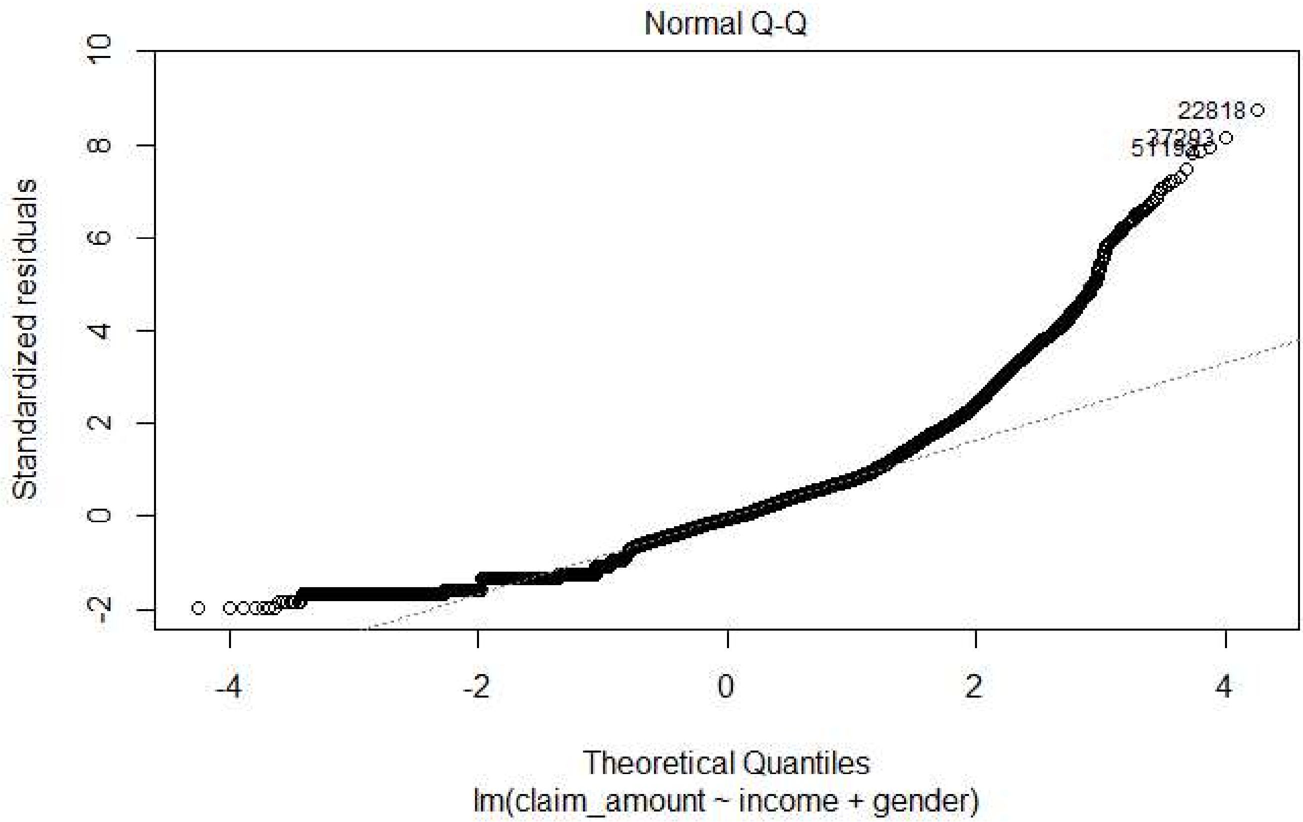
```
lmModel1 <- lm(claim_amount ~ income + gender, data = train)
summary(lmModel1)
```

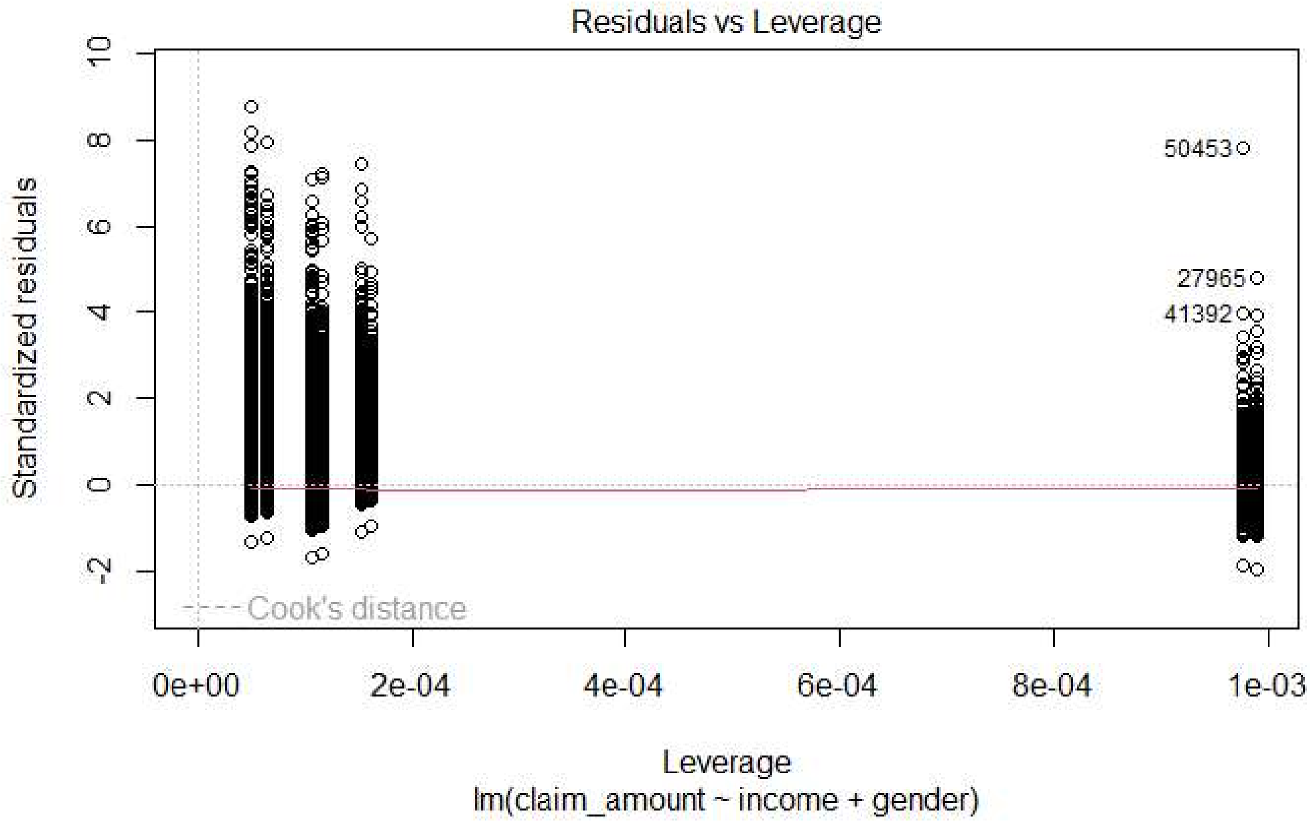
```

Call:
lm(formula = claim_amount ~ income + gender, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-6325.7 -1931.6  -205.8  1678.9 28184.9 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)          5988.02     100.59  59.527   <2e-16 ***
income2L-5L          -865.53     104.38  -8.292   <2e-16 ***
income5L-10L        -1976.53     101.70 -19.435   <2e-16 ***
incomeMore than 10L -2829.12     106.65 -26.527   <2e-16 ***
genderMale            337.63      29.78  11.339   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3219 on 47671 degrees of freedom
Multiple R-squared:  0.04429,	Adjusted R-squared:  0.04421 
F-statistic: 552.4 on 4 and 47671 DF,  p-value: < 2.2e-16
```

Hide

```
plot(lmModel1, which = c(1, 2, 5))
```


Residuals vs Fitted
22818
37293
51192
Residuals
Fitted values
lm(claim_amount ~ income + gender)


Normal Q-Q
22818
37293
51192
Standardized residuals
Theoretical Quantiles
lm(claim_amount ~ income + gender)



Hide

```
NA
NA
```

Third Linear regression

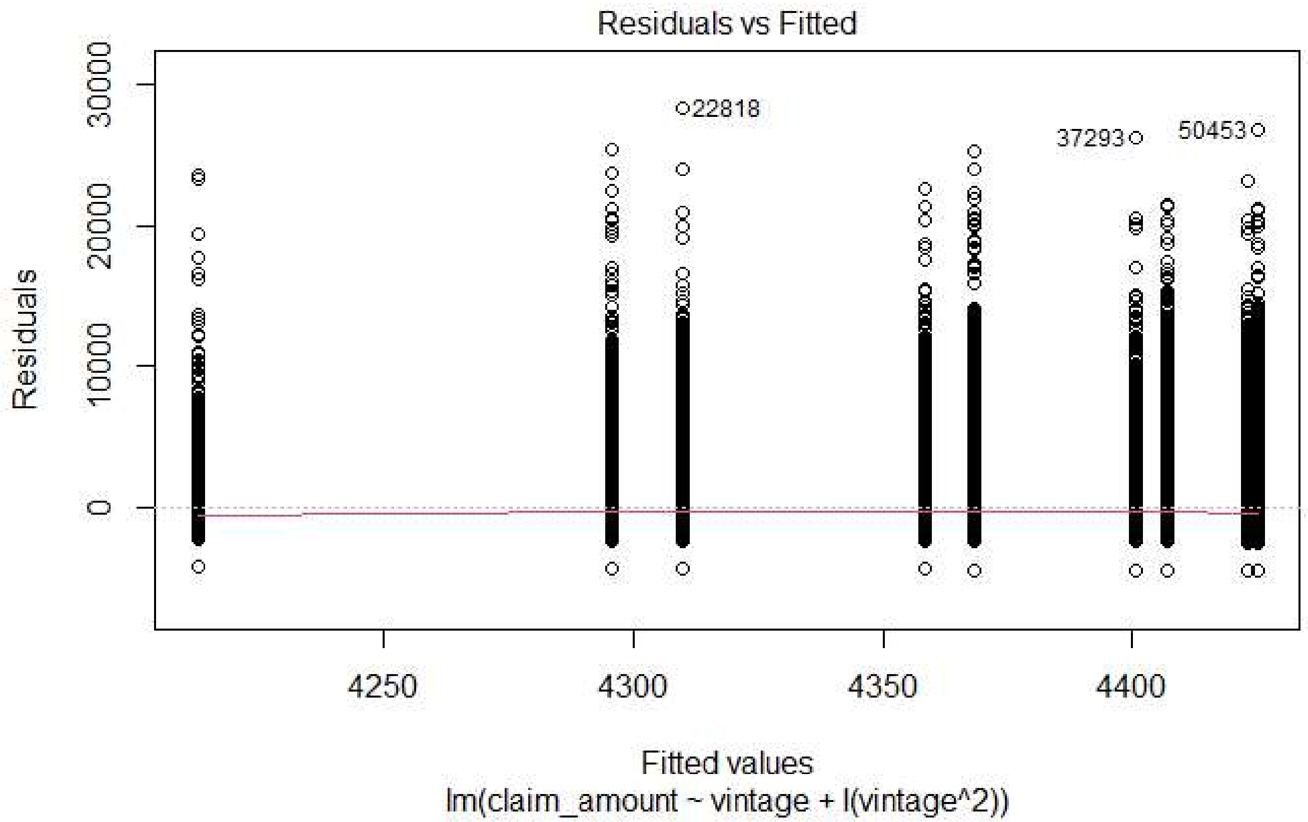
Hide

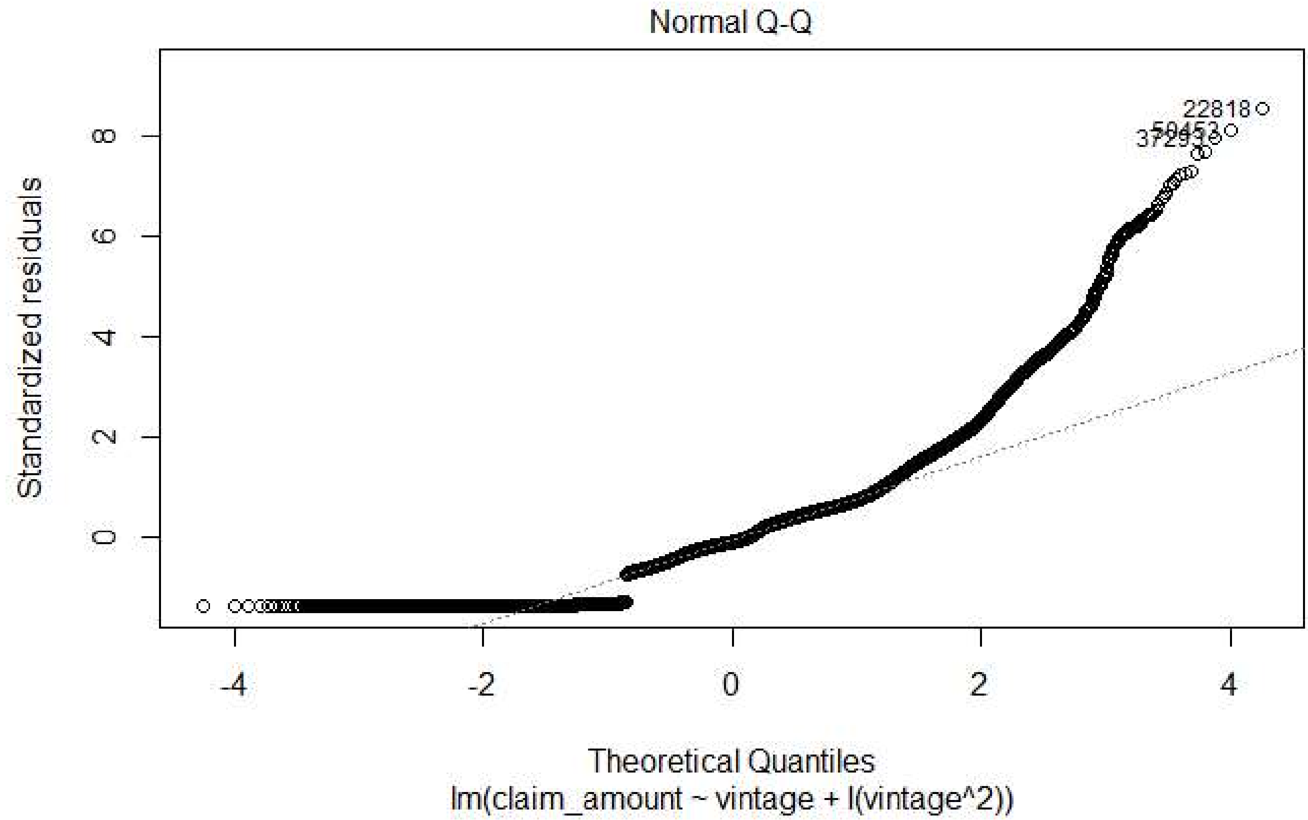
```
lmModel3 <- lm(claim_amount ~ vintage + I(vintage^2), data = train)


# Output summary of the model
summary(lmModel3)
```

```
Call:
lm(formula = claim_amount ~ vintage + I(vintage^2), data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-4425.0 -1966.9  -270.7  1731.5 28224.1 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4212.866     49.828  84.548  < 2e-16 ***
vintage        92.902     25.996   3.574 0.000352 ***
I(vintage^2)  -10.096      3.006  -3.359 0.000784 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3292 on 47673 degrees of freedom
Multiple R-squared:  0.0002709, Adjusted R-squared:  0.0002289 
F-statistic: 6.459 on 2 and 47673 DF,  p-value: 0.001568
```

Hide

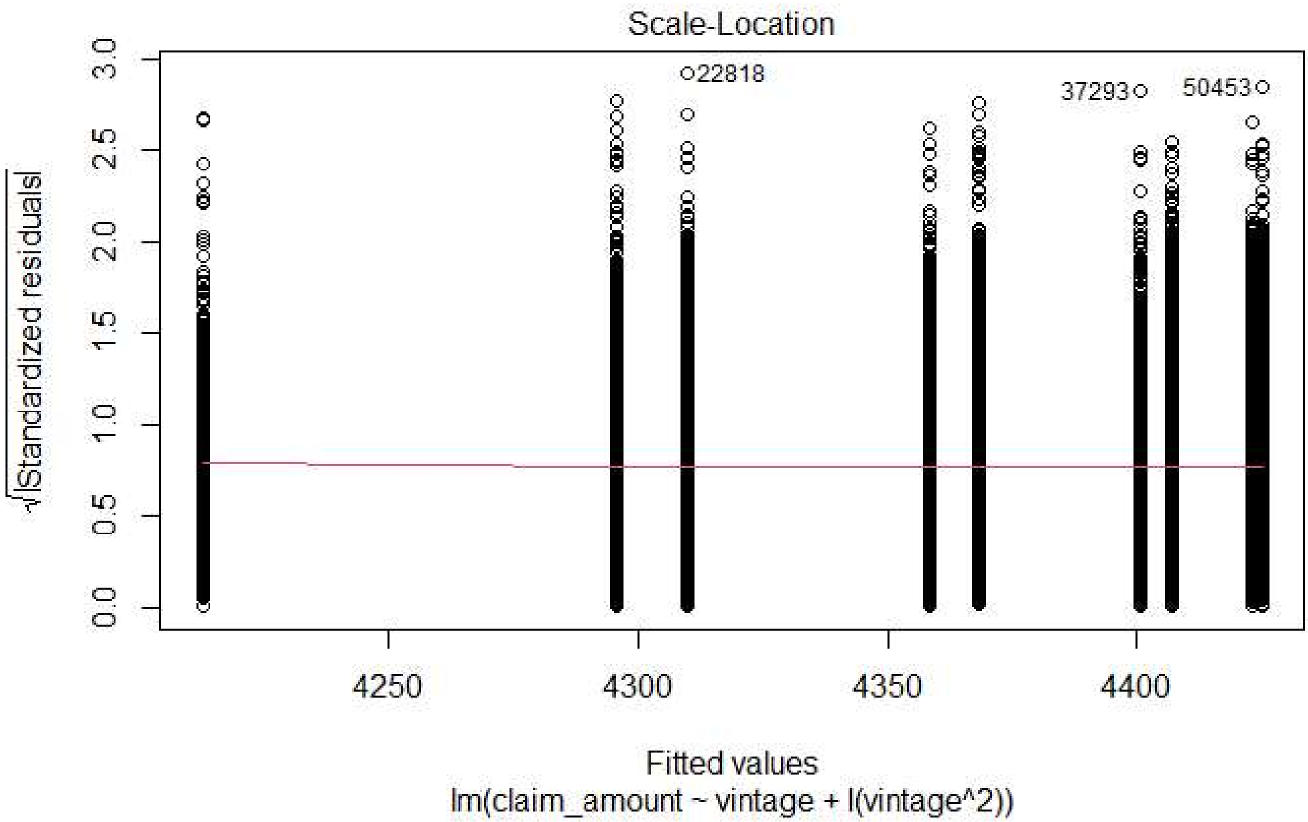
```
# Plot residuals
plot(lmModel3, which = c(1, 2, 3))
```





Hide

NA



we can see that three linear regression models have been built to predict the claim amount. The first model has only one predictor variable, vintage, while the second model has two predictor variables, income and gender. The third model also has two predictor variables, vintage and the squared term of vintage.To compare the models, we can look at their respective R-squared values, residual standard errors, and p-values. The first model has an R-squared value of 0.005327 and a residual standard error of 3265, while the second model has an R-squared value of 0.0002709 and a residual standard error of 3292. The third model has an R-squared value of 0.0002458 and a residual standard error of 3293.Compared to the second and third models, the first model has a higher R-squared value and a lower residual standard error, indicating that it provides a better fit to the data. However, we cannot definitively conclude that the first model is the best model without further analysis.

Hide

```
# Model 1: Simple Linear Regression
# Predict on test data
pred1 <- predict(lmModel, newdata = test)

# Calculate correlation and MSE
cor1 <- cor(pred1, test$vintage)
mse1 <- mean((pred1 - test$vintage)^2)

# Model 2: Multiple Linear Regression
# Predict on test data
pred2 <- predict(lmModel1, newdata = test)

# Calculate correlation and MSE
cor2 <- cor(pred2, test$claim_amount)
mse2 <- mean((pred2 - test$claim_amount)^2)

# Model 3: Polynomial Regression
# Predict on test data
pred3 <- predict(lmModel3, newdata = test)

# Calculate correlation and MSE
cor3 <- cor(pred3, test$claim_amount)
mse3 <- mean((pred3 - test$claim_amount)^2)

# Print the results
cat("Model 1 - Simple Linear Regression\n")
```

```
Model 1 - Simple Linear Regression
```

Hide

```
cat("Correlation: ", cor1, "\n")
```

```
Correlation:  0.09614855
```

Hide

```
cat("MSE: ", mse1, "\n\n")
```

```
MSE:  5.21234
```

Hide

```
cat("Model 2 - Multiple Linear Regression\n")
```

```
Model 2 - Multiple Linear Regression
```

Hide

```
cat("Correlation: ", cor2, "\n")
```

```
Correlation:  0.2203687
```

Hide

```
cat("MSE: ", mse2, "\n\n")
```

```
MSE:  10172446
```

Hide

```
cat("Model 3 - Polynomial Regression\n")
```

```
Model 3 - Polynomial Regression
```

Hide

```
cat("Correlation: ", cor3, "\n")
```

```
Correlation:  0.02490154
```

Hide

```
cat("MSE: ", mse3, "\n")
```

```
MSE:  10684325
```

Based on the results, it appears that Model 2 (Multiple Linear Regression) performed the best, with the highest correlation and lowest MSE. IT was not as my expectation, I thought Model 1 will perform the best.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.