# Medical Cost Prediction Using GLMs in R

Angelina Cottone
UWP 104AY
October 31, 2025

# Table of Contents

# Introduction

This manual provides a step-by-step guide to building a predictive model for medical costs using Generalized Linear Models (GLMs) in R. It is intended for data science and actuarial students with basic knowledge of R programming and regression analysis.

Medical cost prediction is an essential task in healthcare analytics and modeling, as it assists insurers and hospitals in estimating future expenses and allocating resources efficiently. Traditionally, analysts have used Ordinary Least Squares (OLS) regression to model healthcare expenses. However, OLS assumes that the response variable (expenses) follows a normal distribution and residuals (errors) have constant variance. These assumptions are largely not followed in medical cost data, where expenditures are often right-skewed and heteroscedastic, a few patients incur high costs while many have low or zero costs.

Generalized Linear Models extend classical linear regression by allowing the response variable to follow a variety of distributions and by utilizing a link function that relates the mean of the response to the linear combination of predictors. In medical costs, a Gamma distribution with a log link is often used, addressing the skewness inherent in the data and allowing for more accurate and interpretable predictions.

By following this guide, students will not only learn how to implement a GLM for medical costs but also to interpret coefficients, assess model fit, and visualize predictions in an informative and professional manner.

# Required Tools and Setup

Before beginning the modeling process, please ensure that the necessary software and packages are installed and configured, and you have downloaded the dataset onto your computer.

## Software & System Requirements

To complete this analysis, you will need:

- **R** (version 3.6.0 or newer): A powerful and widely used language for statistical analysis, providing a range of functions for data manipulation, modeling and visualization. Download here.
- **RStudio**: An integrated development environment that simplifies coding and report generation. It provides syntax highlighting, code completion, and contains built-in tools to assist in the coding process. Download here.
- Basic familiarity with R, including data import, plotting, and regression modeling.
- Internet access to download necessary packages and the dataset.

### Data Requirements

The dataset used in this guide is available on Kaggle at Medical Insurance Cost Prediction Dataset. You will need to sign in or register to download. The dataset contains anonymous, individual-level information on medical costs and patient characteristics, including:

| Variable | Data Type | Description |
|---|---|---|
| age | Integer | Age of the individual |
| sex | Categorical | Gender (male/female) |
| bmi | Numeric | Body Mass Index |
| children | Integer | Number of children |
| smoker | Categorical | Whether the person is a smoker (yes/no) |
| region | Categorical | Region of residence |
| charges | Integer | Total medical costs (response variable) |

It is recommended to save the dataset in a dedicated project folder to maintain an organized workflow and prevent file path errors during analysis.

### Safety and Privacy Note

When working with medical data, ensure that any dataset used is de-identified to protect individual patient privacy. Any personally identifiable information must be removed in compliance with HIPAA regulations. Use simulated or publicly available data whenever possible.

## Step-by-Step Instructions

### Step 0: Set Up Your Markdown File

1. Open RStudio.
2. Go to the top left corner and click the file icon to open a drop-down menu. Find the R Markdown option.
3. Title your document, i.e. *Medical Cost Prediction with GLM*. You can also select the type of output you want (HTML, PDF, Word).
4. A new markdown file will be generated. It is auto-populated with examples of code blocks and introduces “knitting”, which we will use later to generate our report.
5. Delete everything on the page starting from `## R Markdown`.

**Figure 1: RStudio Home Page**

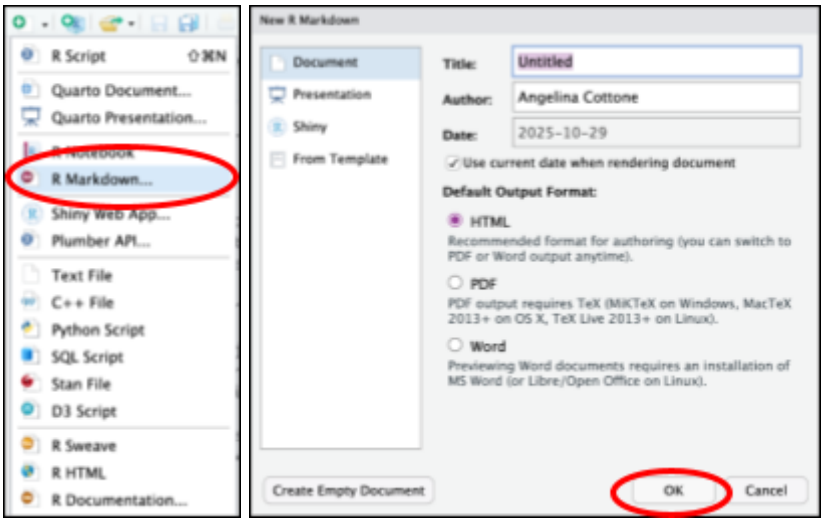



**Figure 2: Drop-Down Menu**

**Figure 3: New Markdown Creation**

```
---
title: "Untitled"
author: "Angelina Cottone"
date: "`r Sys.Date()`"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see
<http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

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

**Figure 4: New Markdown Format**

R Markdown files allow you to combine code, text, and figures into a single reproducible document.

## Step 1: Install and Load Packages

Install or verify the following R packages are present before proceeding:

- **tidyverse** - data manipulation and cleaning
- **ggplot2** - data visualization
- **MASS** - statistical modeling and GLM support
- **caret** - data partitioning and model evaluation
- **scales** - formatting for plots and numerical outputs

```r
install.packages(c("tidyverse", "ggplot2", "MASS", "caret", "scales"))

library(tidyverse)
library(MASS)
library(caret)
library(scales)
library(ggplot2)
```

**Code Block 1: Install and Load Packages**

These packages provide the essential tools for data cleaning, modeling, visualization, and evaluation, and are necessary for completing this analysis.

## Step 2: Import and Inspect Data

Set your working directory to the folder containing the dataset and import the file into R.

```r
setwd("~/Desktop/school/uwp") # Adjust path to your local directory
data <- read.csv("medical_insurance.csv")

# Preview first rows of data set and summary statistics
head(data)
summary(data)

# Check for missing values
colSums(is.na(data))
```

**Code Block 2: Import and Inspect Data**

| | age <int> | sex <chr> | bmi <dbl> | children <int> | smoker <chr> | region <chr> | charges <dbl> |
|---|---|---|---|---|---|---|---|
| 1 | 19 | female | 27.900 | 0 | yes | southwest | 16884.924 |
| 2 | 18 | male | 33.770 | 1 | no | southeast | 1725.552 |
| 3 | 28 | male | 33.000 | 3 | no | southeast | 4449.462 |
| 4 | 33 | male | 22.705 | 0 | no | northwest | 21984.471 |
| 5 | 32 | male | 28.880 | 0 | no | northwest | 3866.855 |
| 6 | 31 | female | 25.740 | 0 | no | southeast | 3756.622 |

6 rows

**Figure 5: Dataset Preview**

```
      age            sex                 bmi           children       smoker
 Min.   :18.00   Length:2772        Min.   :15.96   Min.   :0.000   Length:2772
 1st Qu.:26.00   Class :character   1st Qu.:26.22   1st Qu.:0.000   Class :character
 Median :39.00   Mode  :character   Median :30.45   Median :1.000   Mode  :character
 Mean   :39.11                      Mean   :30.70   Mean   :1.102
 3rd Qu.:51.00                      3rd Qu.:34.77   3rd Qu.:2.000
 Max.   :64.00                      Max.   :53.13   Max.   :5.000
    region             charges
 Length:2772        Min.   : 1122
 Class :character   1st Qu.: 4688
 Mode  :character   Median : 9333
                    Mean   :13261
                    3rd Qu.:16578
                    Max.   :63770
     age       sex       bmi  children    smoker    region   charges
       0         0         0         0         0         0         0
```

**Figure 6: Summary Statistics**

- `head()` displays the first few rows.
- `summary()` provides descriptive statistics for each variable, including min, max, mean, etc.
- `colSums(is.na())` checks for missing values.

No missing values were found in this dataset, so data cleaning is not needed.

## Step 3: Explore the Data

Before we can start building the model, it is important to visualize the relationships between predictors and medical costs. This can give insight into linear relationships between variables and also help you identify potential outliers or unusual patterns in the dataset.
Scatterplots can be generated to identify patterns and potential predictors for continuous variables against the response.

```r
# Scatterplot for BMI vs. Charges, colored by smoking status
ggplot(data, aes(x = bmi, y = charges, color = smoker)) +
  geom_point(alpha=0.6) +
  labs(title = "Medical Charges by BMI and Smoking Status",
       x = "Body Mass Index (BMI)",
       y = "Charges (USD)") +
  theme_minimal()
```

**Code Block 3: Explore the Data**

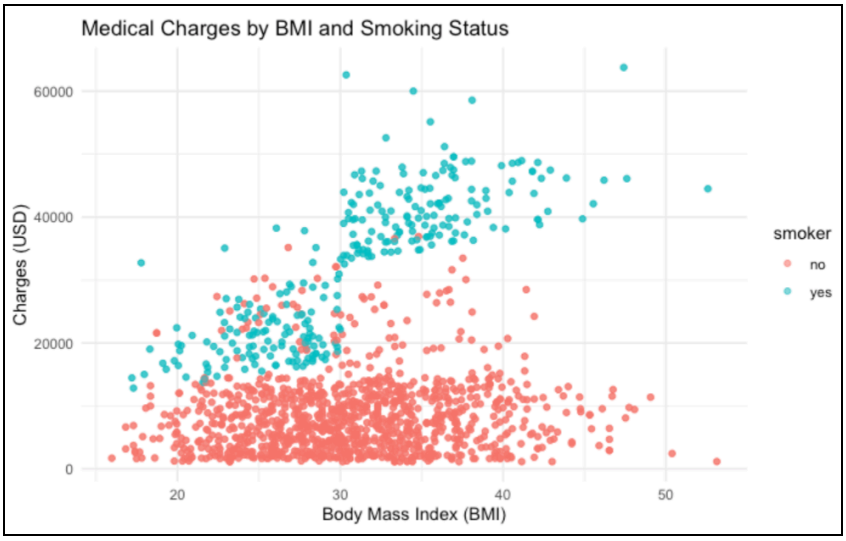



**Figure 7: Scatterplot of Charges vs. BMI Colored by Smoking Status**

- Non-smokers generally have lower medical costs.
- BMI shows a small positive correlation with charges.

Similar plots can be created for other variables such as age, number of children, and sex to visualize the relationships between them. Histograms are also useful for visualizing categorical variables.

## Step 4: Prepare Data for Modeling

Convert categorical variables to factors and split the data into training (80%) and testing (20%) sets for model evaluation. An 80-20 split is standard.

```
```{r}
# Convert categorical data to factors
data <- data %>% mutate(across(c(sex, smoker, region), as.factor))

# Set seed for reproducibility and split data into training (80%) and testing (20%) sets
set.seed(123)
trainIndex <- createDataPartition(data$charges, p = 0.8, list = FALSE)
train <- data[trainIndex, ]
test <- data[-trainIndex, ]
```
```

**Code Block 4: Prepare Data for Modeling**

This step ensures the GLM treats categorical variables properly, and performance evaluations can be done on unseen test data, providing a realistic assessment of the model's ability to predict well.

## Step 5: Build the GLM Model

Since medical cost data is positive and right-skewed, a Gamma distribution with a log link is appropriate. This combination will help stabilize variance, and the log link ensures predictions remain positive and interpretable. After building the GLM, use the `summary()` function to review model results.

```r
glm_model <- glm(charges ~ age + bmi + smoker + region + sex,
                 data = train, family = Gamma(link = "log"))

# Display model summary
summary(glm_model)

# Exponentiate coefficients for interpretation
exp(coef(glm_model))
```

**Code Block 5: Build the GLM Model**

The output includes estimates for each predictor, standard errors, z-values, and p-values.

- Coefficients indicate how each predictor affects medical costs.
- Asterisks (***, **, *) show statistically significant predictors.
- Deviance and AIC measure how well the model predicts, lower AIC indicates better performance.

```
Call:
glm(formula = charges ~ age + bmi + smoker + region + sex, family = Gamma(link = "log"),
    data = train)

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      7.441832   0.086231  86.301  < 2e-16 ***
age              0.028446   0.001037  27.421  < 2e-16 ***
bmi              0.015202   0.002502   6.075 1.46e-09 ***
smokeryes        1.503089   0.036346  41.355  < 2e-16 ***
regionnorthwest -0.021496   0.042184  -0.510  0.61039
regionsoutheast -0.156755   0.042351  -3.701  0.00022 ***
regionsouthwest -0.131912   0.041917  -3.147  0.00167 **
sexmale         -0.041281   0.029214  -1.413  0.15778
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 0.4687488)

    Null deviance: 1779.68  on 2219  degrees of freedom
Residual deviance:  588.09  on 2212  degrees of freedom
AIC: 43841

Number of Fisher Scoring iterations: 7

    (Intercept)             age             bmi       smokeryes regionnorthwest regionsoutheast
   1705.8731631       1.0288541       1.0153179       4.4955536       0.9787331       0.8549138
regionsouthwest         sexmale
      0.8764180       0.9595594
```

**Figure 8: GLM Model Summary**

- Significant predictors: age (positive), bmi (positive), smoker (‘yes’, positive), region (‘southeast’ and ‘southwest’, negative).
- Null deviance = 1779.68, Residual Deviance = 588.09, AIC = 43841
- Exponentiated coefficients represent multiplicative effects on expected medical costs.

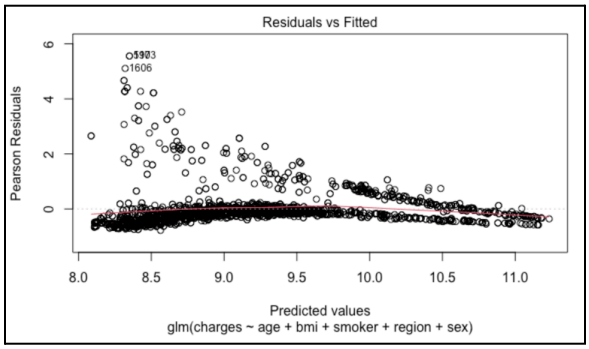
## Step 6: Evaluate Model Fit

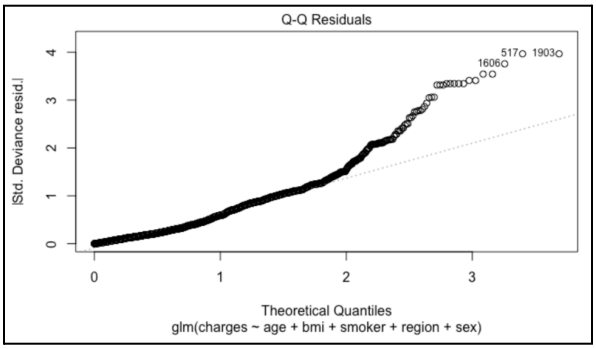
The `plot()` function will generate several plots of residuals.

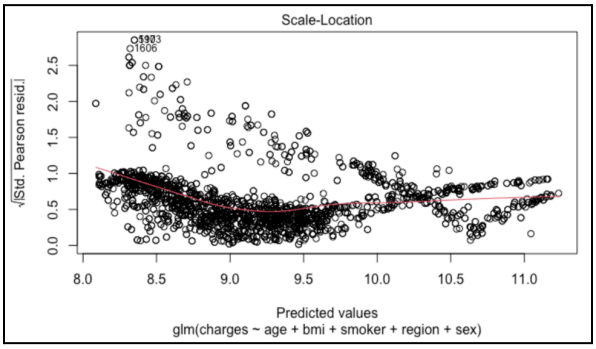
```r
# Visualize residual plots
plot(glm_model)

# Calculate pseudo-R^2, gives approximate measure of explained variation
1 - (glm_model$deviance / glm_model$null.deviance)
```

**Code Block 6: Evaluate Model Fit**



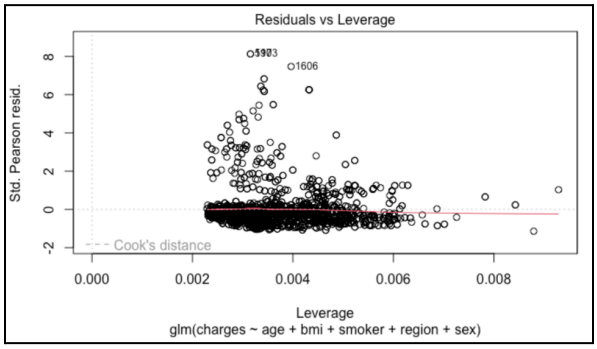




**Figure 9: Residuals vs. Fitted Plot**
**Figure 10: Q-Q Residuals Plot**
**Figure 11: Scale-Location Plot**
**Figure 12: Residuals vs. Leverage Plot**

```
[1] 0.6695551
```

**Figure 13: Pseudo-R^2 Results**

- Residual plots help detect non-linearity or heteroscedasticity in the data.
- Q-Q plots check if residuals approximate the assumed distribution.
- Pseudo-R^2 gives an approximate measure of explained variation, similar to R^2 in linear regression.

## Step 7: Model Comparison

You can remove insignificant predictors to make a better-fitting model.

```r
# Comparing two models using AIC
glm_model2 <- glm(charges ~ age + bmi + smoker + region,
                  data = train,
                  family = Gamma(link = "log"))

AIC(glm_model, glm_model2)
```

**Code Block 7: Model Comparison**

| | df <dbl> | AIC <dbl> |
|---|---|---|
| glm_model | 9 | 43841.03 |
| glm_model2 | 8 | 43842.70 |

2 rows

**Figure 14: Model Comparison Results**

In this case, there is not an improvement in the AIC when removing insignificant predictors (lower AIC → better fit), so we will stick with the original model with all predictors.

## Step 8: Make Predictions on Test Data

Use your trained model to predict medical costs for new patients or test data.

```r
predictions <- predict(glm_model, newdata = test, type = "response")

# Compare actual vs prediction
comparison <- data.frame(actual = test$charges, predicted = predictions)
head(comparison)
```

**Code Block 8: Make Predictions on Test Data**

| | actual <dbl> | predicted <dbl> |
|---|---|---|
| 2 | 1725.552 | 3901.754 |
| 5 | 3866.855 | 6175.483 |
| 14 | 11090.718 | 13139.361 |
| 17 | 10797.336 | 11955.131 |
| 19 | 10602.385 | 13019.792 |
| 22 | 4149.736 | 5743.513 |

6 rows

**Figure 15: Actual vs. Predicted Results**

We can see from these results that our model tends to predict higher than the actual value.

## Step 9: Visualize Model Performance

```r
ggplot(comparison, aes(x = actual, y = predicted)) +
  geom_point(alpha = 0.5) +
  geom_abline(color = "red", linetype = "dashed") +
  labs(title = "Actual vs. Predicted Medical Costs",
       x = "Actual Charges (USD)",
       y = "Predicted Charges (USD)") +
  theme_minimal()
```

**Code Block 9: Visualize Model Performance**

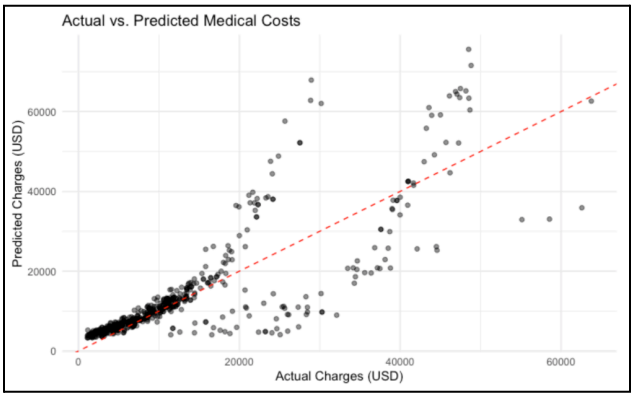



**Figure 16: Actual vs. Predicted Plot**

Points close to the red line indicate accurate predictions. This plot shows the model struggles are charges get higher.

## Step 10: Saving Your Analysis

There is a button on the top of RStudio with a yarn ball. You can click this to knit your markdown file together and save it as a PDF or HTML file. This will save your code and results together.

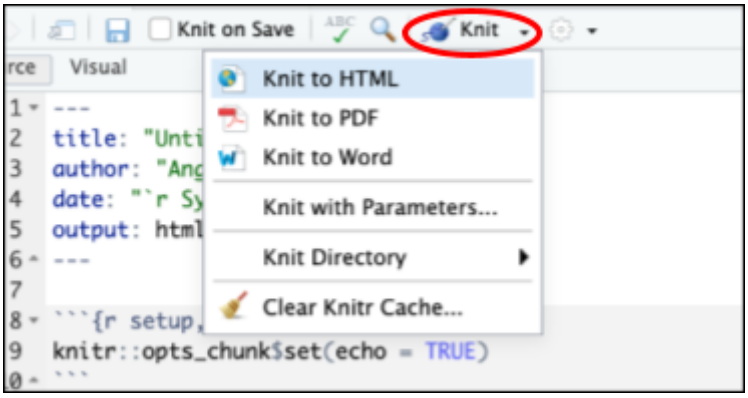



**Figure 17: RStudio Knit**

**Step 3: Exploratory Data Analysis**

```
# Scatterplot for BMI vs. Charges, colored by smoking status
ggplot(data, aes(x = bmi, y = charges, color = smoker)) +
  geom_point(alpha=0.6) +
  labs(title = "Medical Charges by BMI and Smoking Status",
       x = "Body Mass Index (BMI)",
       y = "Charges (USD)") +
  theme_minimal()
```

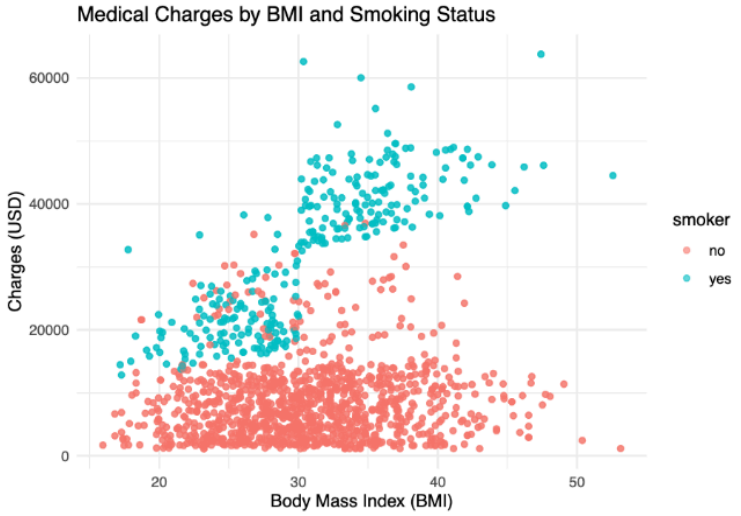



**Figure 18: Knit PDF**

## Troubleshooting and Tips

If the model fails to converge:

- Ensure there is no missing data.
- Try simplifying the model by removing multicollinearity between variables.
- Adjust optimization settings if necessary.

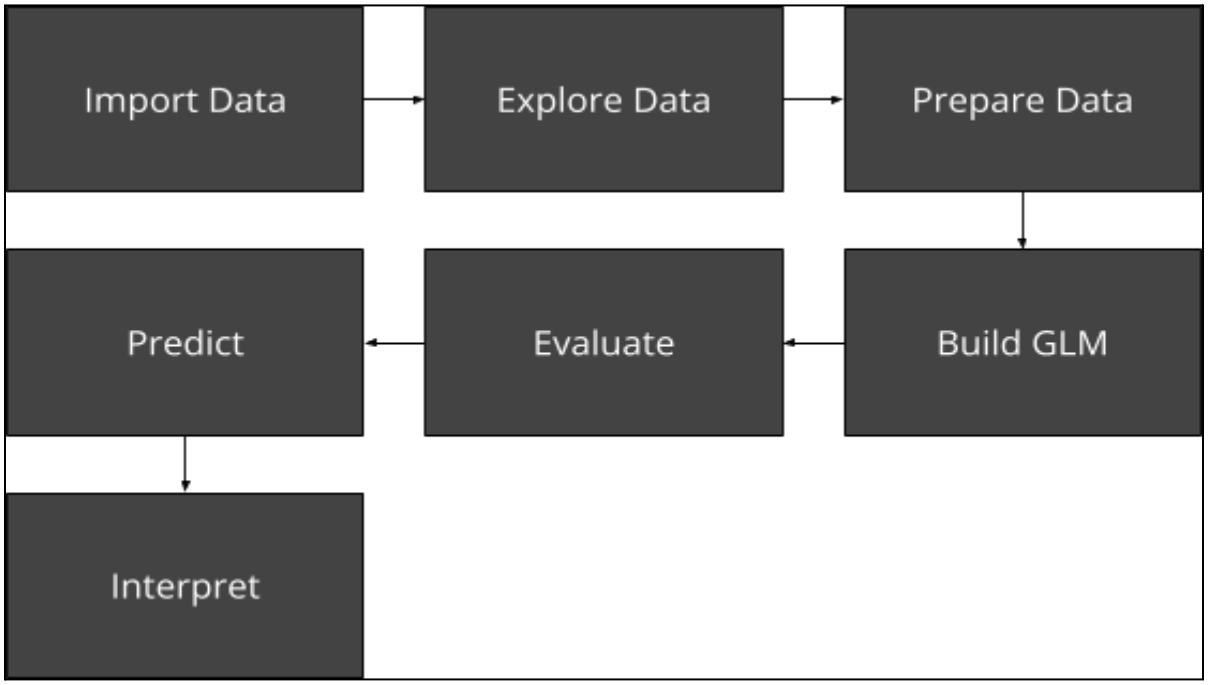
If predictions seem unrealistic:

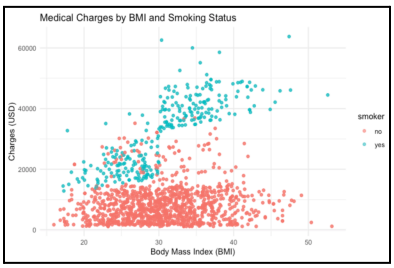
- Verify correct variable types.
- Consider rescaling numeric predictors.

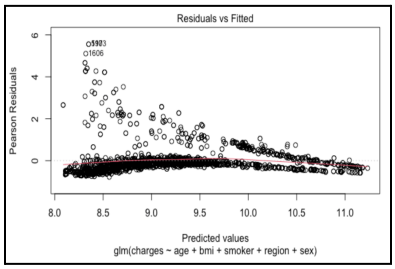
Improving model fit:

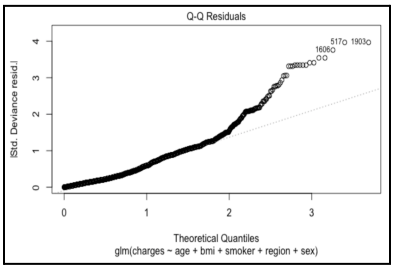
- Use stepwise selection to optimize predictors based on AIC.
- Try including interaction terms (e.g. smoker * bmi)
- Experiment with alternative link functions if appropriate

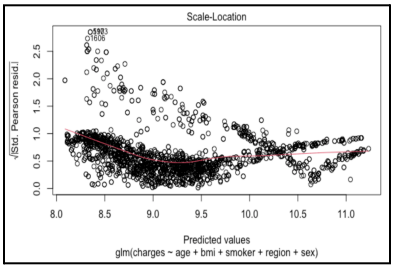
# Visual Summary

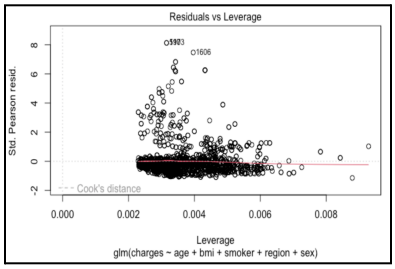


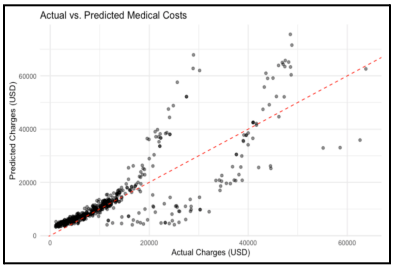




| **Metric** | **Value** |
|---|---|
| Pseudo R^2 | 0.67 |
| Null Deviance | 1779.68 |
| Residual Deviance | 588.09 |
| AIC | 43841 |