Predicting Stroke Risk: A Multifactorial Approach

By: Advait Apte, Ahmed Ghabin, Benny Frisella (at 33.3% each)
For: STAT 4354 with Shengjie Jiang
https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

Abstract

This report presents a multifactorial approach to predicting stroke risk. Utilizing advanced statistical techniques and a rich dataset, this study identifies key predictors of stroke and evaluates their significance through various modeling approaches. Key findings suggest that factors such as age, hypertension, BMI, and heart disease significantly influence stroke risk.

Table of Contents

1. Introduction

According to the Centers for Disease Control (CDC), “every 40 seconds someone in the U.S. has a stroke and every 3 minutes and 14 seconds someone dies from a stroke in the U.S., ” (Stroke). An American Heart Association (AHA) study done in April 2023 projected that “**globally**, **stroke is the second leading cause of death**, accounting for **11.6% of all deaths in 2019**, ” (Pu).

2. Dataset Overview

Discrete Variables:

- Gender: "Male", "Female", "Other"
- Hypertension: 0 or 1 (0 if the patient doesn't have hypertension, 1 if the patient has hypertension)

- Heart Disease: 0 or 1 (0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease)
- Ever Married: "No", "Yes"
- Work Type: "Children", "Govt_job", "Never_worked", "Private", "Self-employed"
- Residence Type: "Rural", "Urban"
- Smoking Status: "Formerly smoked", "Never smoked", "Smokes", "Unknown"
- Stroke: 0 or 1 (1 if the patient had a stroke, 0 if not)

Continuous Variables:

- Average Glucose Level: (average glucose level in blood)
- BMI: (body mass index)
- Age: (age of the patient)

3. Methodology

Data Cleaning: BMI was the only category with nulls and had over 200 missing values. We used Expectation Maximization(EM) to impute the missing values. Upon further investigation, we noticed the stroke population accounted for less than 5% of the dataset, indicating class imbalance. Recognizing the potential adverse effects on model performance, we addressed the imbalance by oversampling the minority stroke cases using Bootstrap resampling. Implementing Bootstrap ensured that each observation in the original dataset had the potential to be selected multiple times in the oversampled dataset, which maintained the integrity of the inherent uncertainty and variability in the dataset.
The goal was to raise the stroke population to a benchmark of 12% since that is the minimum percentage of the population that will suffer from a stroke if that’s the global mortality rate.

Linear Regression: To analyze stroke occurrence, a linear regression model is constructed with predictor variables including: gender, age, hypertension, heart disease, marital status, work type, residence type, average glucose level, BMI, and smoking status. This model aims to identify the relationship between these factors and the likelihood of experiencing a stroke. By fitting the model to the dataset, we can estimate the coefficients for each predictor, elucidating their individual impact on stroke risk. Subsequently, the model's performance can be assessed through measures such as R-squared and hypothesis testing to determine the significance of predictor variables. Insights gained from this analysis can inform targeted interventions and preventative strategies to mitigate stroke incidence.

Newton-Raphson method: The Newton-Raphson method is an iterative technique used to find the roots of a differentiable function, typically to solve systems of equations or optimization problems. In the context of predicting stroke likelihood, the method iteratively updates an initial guess for the coefficients of a linear regression model based on the gradient of the prediction function. This process continues until a stopping criterion, often based on the convergence of the coefficients, is met. By applying this method to the provided linear regression model with predictors such as age, gender, and health factors, we aim to estimate the coefficients that best fit the data and thus predict stroke occurrence. The method's success relies on the assumption that the prediction function is differentiable and that the chosen initial guess is reasonably close to the true solution. Additionally, monitoring convergence and setting appropriate tolerance levels are crucial for ensuring the reliability of the results.

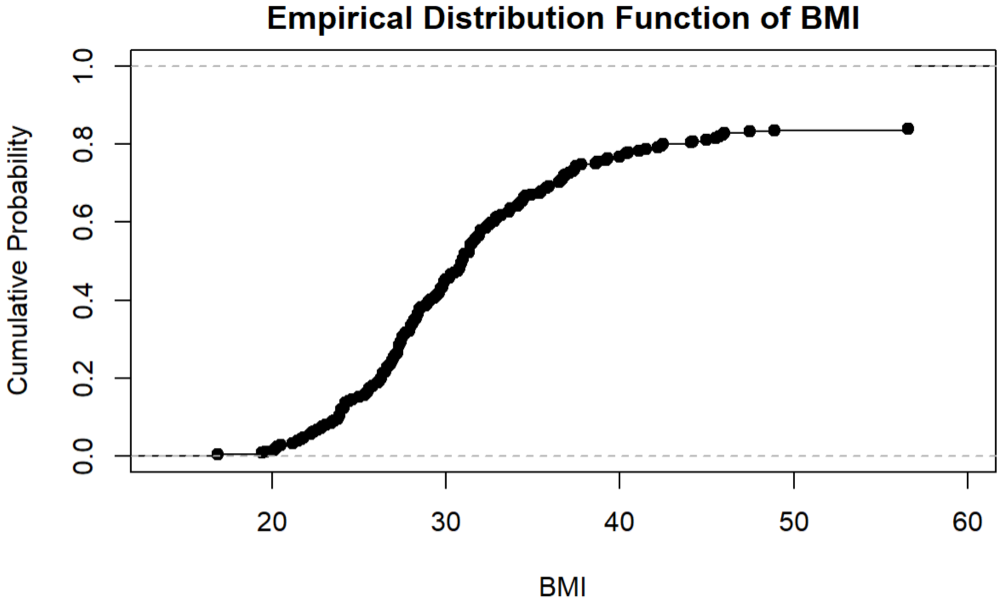
Monte Carlo method: A logistic regression model is employed to predict stroke occurrence based on the average glucose level as the sole predictor variable. This model is fitted to the dataset using the glm function with the family argument set to "binomial" to handle binary outcome data. Subsequently, a Monte Carlo simulation is conducted to estimate the distribution of stroke probabilities. In each iteration of the simulation, a set of simulated average glucose levels is generated from a normal distribution with parameters matching those of the original dataset. The logistic regression model is then used to predict stroke probabilities for the simulated data, and stroke occurrences are simulated based on these probabilities using a binomial distribution.

Clustering: The clustering process in our study was conducted using the K-prototypes algorithm, which is particularly suited for datasets with mixed data types, combining the attributes of K-means for numerical data and K-modes for categorical data. This approach was chosen to effectively handle the variety of variables in our dataset, including continuous variables such as age, average glucose level, and BMI, and categorical variables like gender, hypertension, heart disease, marital status, work type, residence type, and smoking status.

To prepare the data for clustering, categorical variables were first converted to numerical format using label encoding, which is essential for the K-prototypes algorithm to compute distances among categories. The silhouette score, a measure of how similar an object is to its own cluster compared to other clusters, was used to determine the optimal number of clusters. By testing a range from 2 to 10 clusters, we sought to find the configuration that maximized the silhouette score, indicating the best balance of intra-cluster similarity and inter-cluster differentiation.

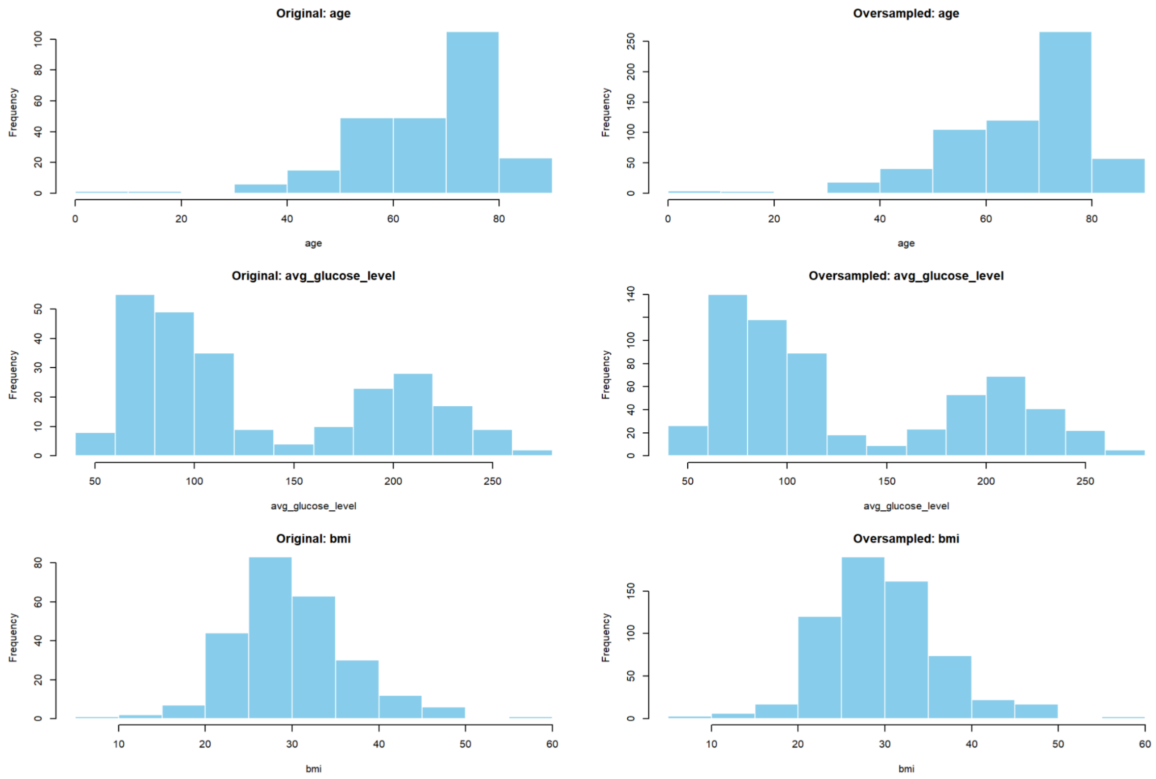
4. Results

Data Cleaning: To impute the missing BMI values, we began by looking at the Empirical Distribution Function (EDF):



The smooth, S-shaped curve of the EDF signifies that the distribution of BMI is approximately Normal which is supported by the Central Limit Theorem. Although it’s not necessary, we used Expectation Maximization here to impute the missing values with slightly more accuracy than just imputing the mean. The BMI mean before EM was 28.89, and marginally increased to 28.93 after EM. Our analysis revealed a *decreased p-value after EM*(0.005 to 0.001), suggesting that BMI became slightly more significant in predicting stroke. However, it must be noted that the adjusted R-squared values were very close to zero, indicating that other variables were much more indicative of a stroke than BMI by itself.

By using Bootstrap Resampling to oversample the stroke population to 12% of the dataset, we addressed class imbalance effectively while preserving data integrity. Each patient had an equal chance of being included, because if it was chosen, it was put back into the pile to possibly be chosen again. This process enriched our dataset with more stroke instances, and lead to the improvement of our models’ ability to detect subtle patterns and nuances associated with strokes.

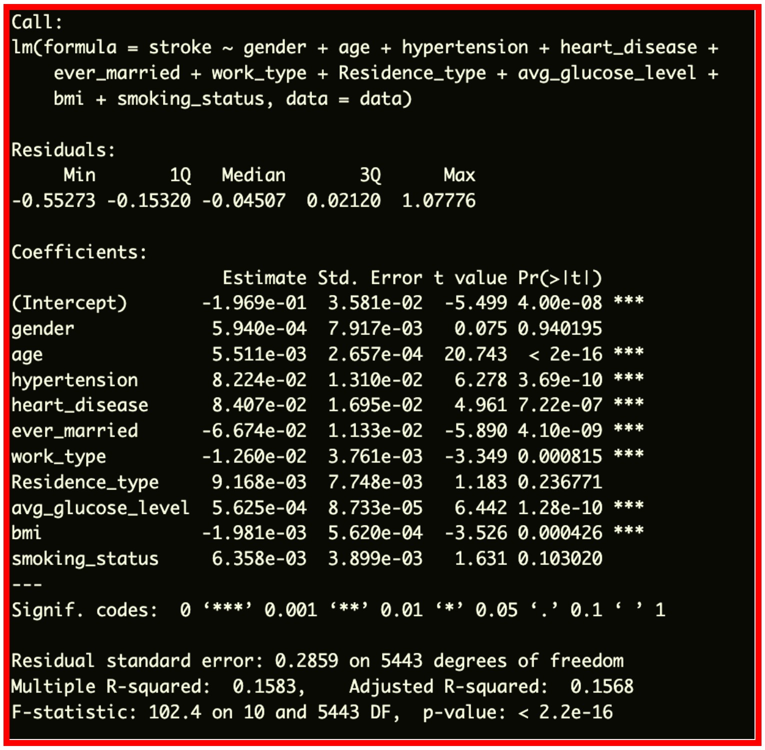


Moreover, maintaining consistent distributions post-oversampling(as seen above) validates the approach’s effectiveness while enhancing the reliability of our predictive models.

Linear Regression Model:

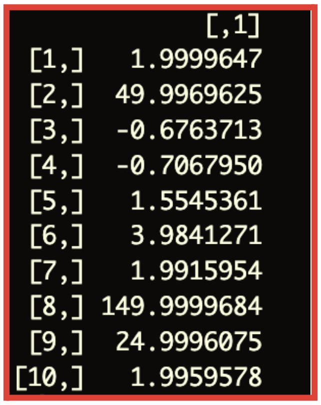
$$\boldsymbol{Stroke} = \beta 0 + \beta 1 \cdot \boldsymbol{gender} + \beta 2 \cdot \boldsymbol{age} + \beta 3 \cdot \boldsymbol{hypertension} + \beta 4 \cdot \boldsymbol{heart\_disease} + \beta 5 \cdot \boldsymbol{ever\_married} + \beta 6 \cdot \boldsymbol{work\_type} + \beta 7 \cdot \boldsymbol{Residence\_type} + \beta 8 \cdot \boldsymbol{avg\_glucose\_level} + \beta 9 \cdot \boldsymbol{bmi} + \beta 10 \cdot \boldsymbol{smoking\_status}$$

The model reveals several significant predictors associated with stroke occurrence. Age demonstrates a strong positive relationship, with each additional year correlating with a higher likelihood of stroke. Hypertension and heart disease also emerge as significant positive predictors, suggesting their substantial impact on stroke risk. Conversely, being “ever married” appears to have a negative association with stroke incidence. Work type and average glucose level exhibit significant relationships as well, indicating their influence on stroke susceptibility. Despite some non-significant predictors such as gender and residence type, the overall model provides valuable insights into the multifaceted nature of stroke risk factors, though further investigation may be warranted to elucidate the role of certain variables.



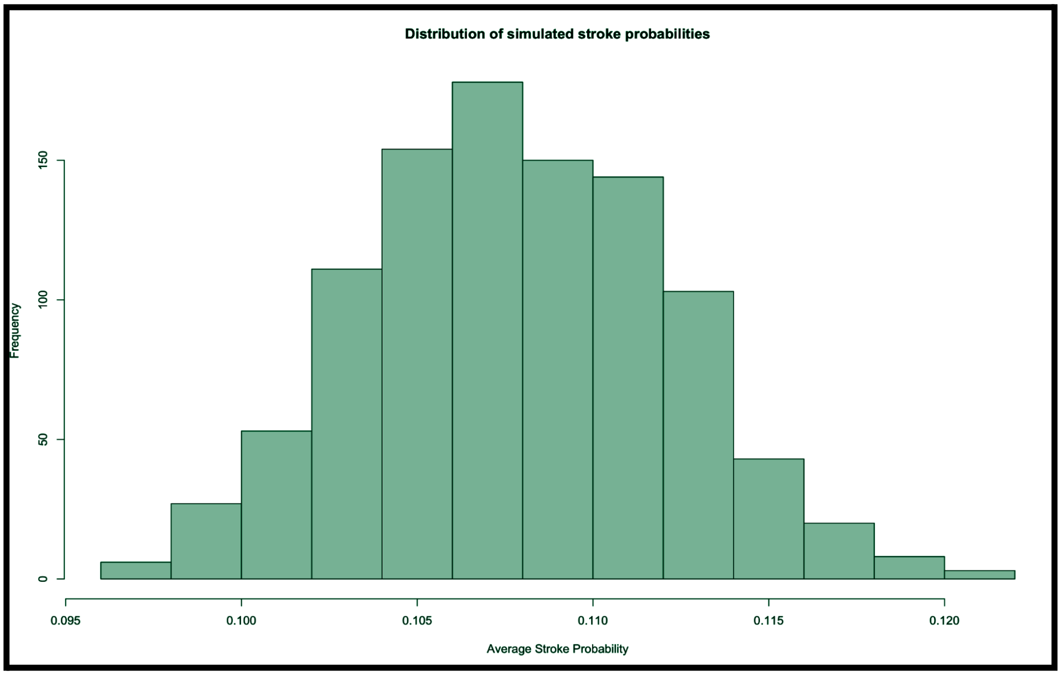
```
Call:
lm(formula = stroke ~ gender + age + hypertension + heart_disease +
    ever_married + work_type + Residence_type + avg_glucose_level +
    bmi + smoking_status, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.55273 -0.15320 -0.04507  0.02120  1.07776

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       -1.969e-01  3.581e-02  -5.499 4.00e-08 ***
gender             5.940e-04  7.917e-03   0.075 0.940195
age                5.511e-03  2.657e-04  20.743  < 2e-16 ***
hypertension       8.224e-02  1.310e-02   6.278 3.69e-10 ***
heart_disease      8.407e-02  1.695e-02   4.961 7.22e-07 ***
ever_married      -6.674e-02  1.133e-02  -5.890 4.10e-09 ***
work_type         -1.260e-02  3.761e-03  -3.349 0.000815 ***
Residence_type     9.168e-03  7.748e-03   1.183 0.236771
avg_glucose_level  5.625e-04  8.733e-05   6.442 1.28e-10 ***
bmi               -1.981e-03  5.620e-04  -3.526 0.000426 ***
smoking_status     6.358e-03  3.899e-03   1.631 0.103020
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2859 on 5443 degrees of freedom
Multiple R-squared:  0.1583,    Adjusted R-squared:  0.1568
F-statistic: 102.4 on 10 and 5443 DF,  p-value: < 2.2e-16
```

Newton-Raphson Method: The output of the Newton-Raphson method provides estimates for the coefficients of the linear regression model used to predict stroke likelihood. Each coefficient corresponds to a predictor variable in the model, such as age, gender, and health indicators like hypertension or heart disease. For instance, the coefficient for age is approximately 50, indicating that each additional year of age is associated with an increased likelihood of stroke. Similarly, coefficients for hypertension and heart disease are negative, suggesting that their presence reduces the likelihood of stroke. Other factors like marriage status and work type also contribute to the predictive model. These coefficients offer insights into the relative importance of each predictor in assessing stroke risk, aiding in understanding and potentially mitigating the factors contributing to stroke occurrence.

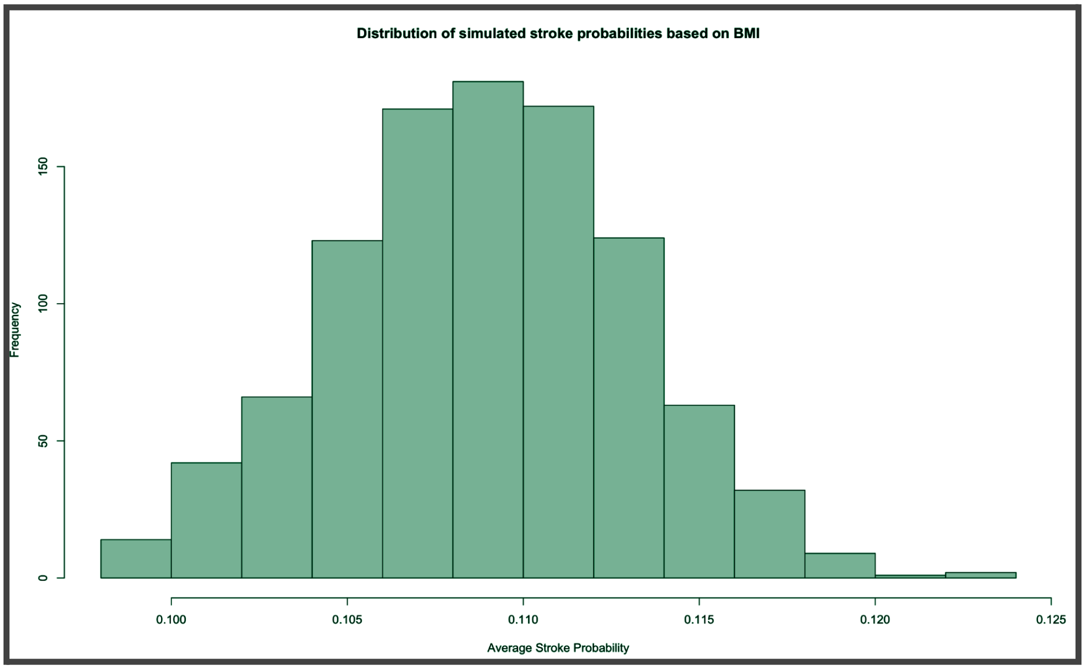


```
             [,1]
 [1,]   1.9999647
 [2,]  49.9969625
 [3,]  -0.6763713
 [4,]  -0.7067950
 [5,]   1.5545361
 [6,]   3.9841271
 [7,]   1.9915954
 [8,] 149.9999684
 [9,]  24.9996075
[10,]   1.9959578
```

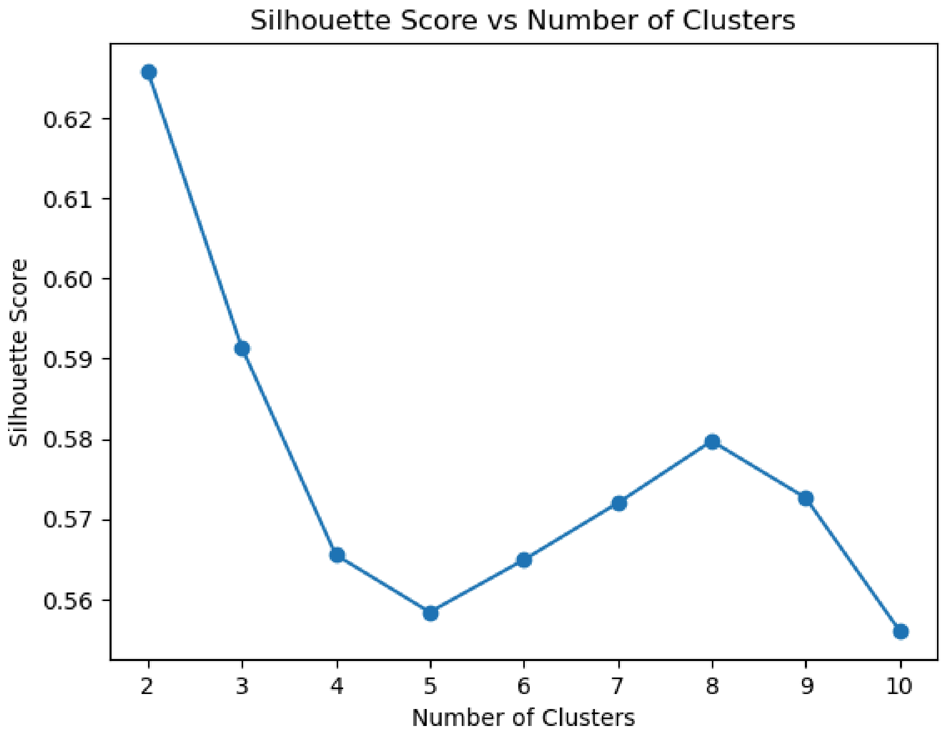
Monte Carlo Method: The simulation generates a distribution of simulated stroke probabilities based on variations in average glucose levels. We can see that the histogram is fairly normally distributed and the range of simulated stroke probabilities is from .095 to .125 while The most common average stroke probability range is between .1075 and .1085. Higher average glucose levels are associated with increased simulated stroke probabilities, reflecting the influence of this predictor variable on stroke likelihood.



For the second Monte Carlo method used, the simulation yields a distribution of simulated stroke probabilities based on variations in BMI. The resulting histogram is also fairly normally distributed and the range of simulated stroke probabilities is from 0.095 to 0.125 and the most common stroke probability range is from 0.1075 to 0.110, with higher BMI values generally associated with increased simulated stroke probabilities. This demonstrates the influence of BMI as a predictor variable on stroke likelihood and provides insights into the variability of stroke risk within the population.

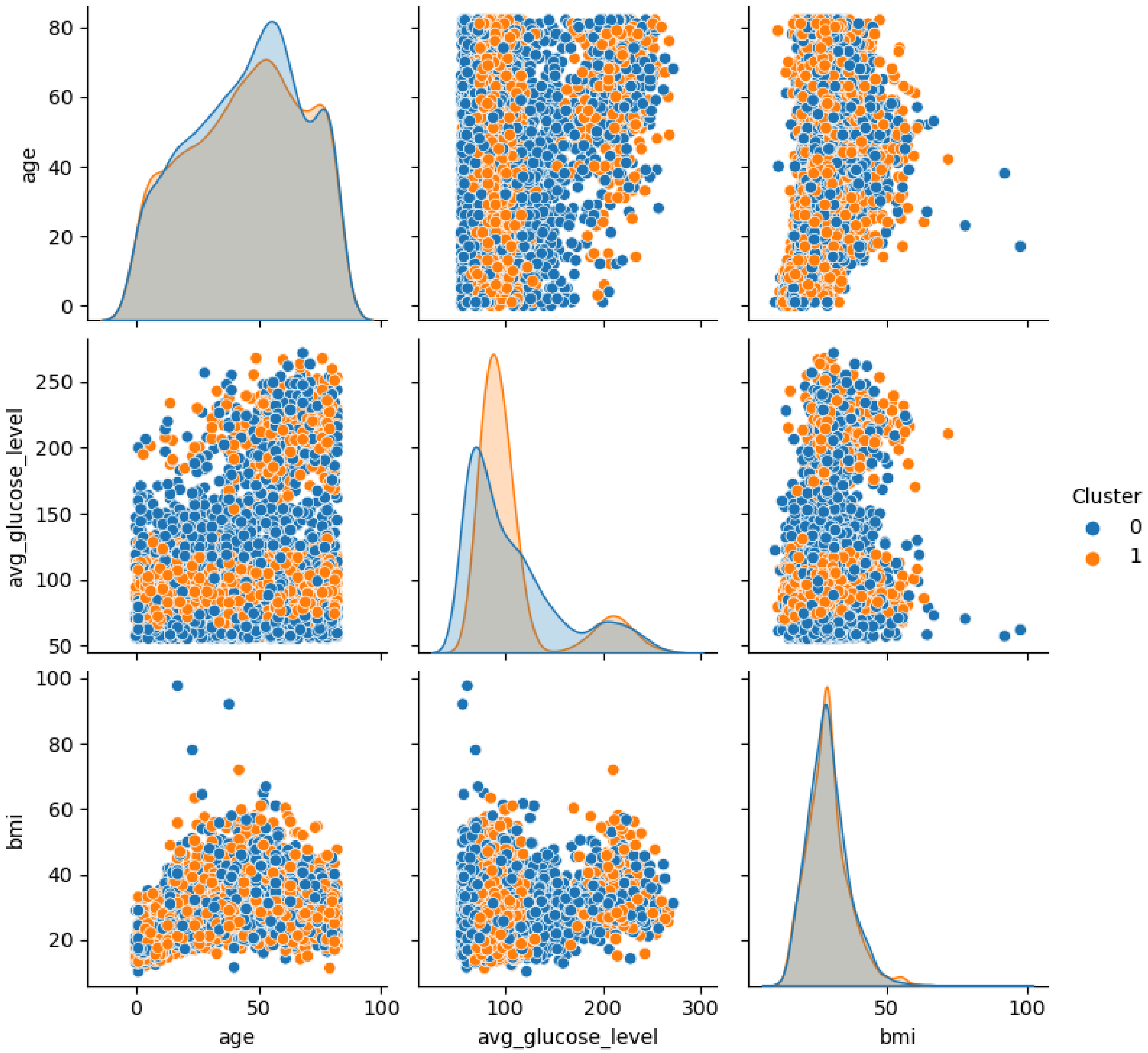


Clustering: Our analysis with the K-prototypes algorithm revealed that the highest silhouette score was obtained with two clusters. This result indicated that dividing the dataset into two clusters provided the most meaningful segmentation, capturing essential differences in stroke risk factors among the subjects.



This graph displays the silhouette scores for different numbers of clusters, with a peak at two clusters indicating the optimal balance of internal consistency and clear separation between clusters.

After establishing the optimal cluster count, we visualized the segmentation using pair plots of critical variables: age, average glucose levels, and BMI. These visualizations demonstrated distinct demographic and health patterns that clearly differentiate the two clusters, highlighting the variations in stroke risk factors across the dataset.



This image shows a pair plot visualization of the two clusters based on age, average glucose levels, and BMI. The distinct demographic and health profiles of each group is evident, emphasizing the differences crucial for stroke risk assessment.

5. Challenges and Limitations

- Data Imbalance: Significant class imbalance with only about 4.8% stroke cases, leading to potential prediction bias towards non-stroke cases.
- Variability in Key Measures: Wide ranges in avg_glucose_level and bmi introduce variability, potentially affecting prediction accuracy for critical stroke risk factors.

- Predictive Power of Non-significant Predictors: Variables like gender, residence type, and smoking status show no statistical significance but may still hold clinical relevance, potentially overlooking key risk factors.
- Generalization Across Populations: Limited representativeness of the broader population may restrict the model’s generalizability across different demographics or geographic areas.

6. Conclusion

Linear Regression: The model offers valuable insights into the complex interplay of demographic and health factors influencing stroke likelihood. Age, hypertension, and heart disease emerge as strong predictors, underlining their pivotal roles in assessing stroke risk. Additionally, factors like marital status, work type, and average glucose level demonstrate notable associations with stroke occurrence. While the model provides significant contributions to understanding stroke epidemiology, further research is warranted to explore additional factors and refine predictive accuracy for more effective prevention and intervention strategies.

Newton-Raphson method: The method has provided estimates for the coefficients of the linear regression model aimed at predicting stroke likelihood. These coefficients offer valuable insights into the relationship between various demographic and health factors and the probability of experiencing a stroke. By iteratively refining initial guesses, this method optimizes the model's predictive accuracy, enabling better understanding and management of stroke risk factors. Utilizing these coefficient estimates, healthcare professionals can tailor interventions and preventive measures more effectively, potentially reducing the burden of stroke on individuals and healthcare systems alike.

Monte Carlo method: The logistic regression model coupled with Monte Carlo simulation offers a robust approach to estimate stroke probabilities based on average glucose levels. The simulation results provide valuable insights into the variability and distribution of stroke probabilities within the dataset. This methodology enables the exploration of potential stroke risks associated with different levels of average glucose, aiding in risk assessment and informing preventive measures for individuals with varying glucose levels. Further research and refinement of the model could enhance its predictive accuracy and utility in clinical settings for personalized stroke risk assessment and intervention planning.

For the second monte carlo method, the simulation results highlight the impact of BMI on stroke risk, with higher BMI values correlating with elevated stroke probabilities. This methodology facilitates the exploration of BMI-related stroke risks and can inform preventative strategies tailored to individuals with varying BMI levels. Continued refinement and validation of the model may enhance its utility in clinical settings for personalized stroke risk assessment and intervention planning.

Clustering: The clustering performed using the K-prototypes algorithm successfully identified two distinct groups within our stroke risk data. The separation into two clusters highlighted significant variations in key health indicators such as age, average glucose levels, and BMI. These differences underline the heterogeneity in stroke risk across the population, suggesting different preventive and monitoring needs for each group.

The distinct profiles of the clusters indicate the necessity for tailored health interventions. The cluster comprising younger, lower-risk individuals might benefit from general preventive measures, while the cluster of older, higher-risk individuals requires targeted, intensive monitoring and preventive strategies.

This stratified approach to understanding stroke risk through clustering not only enhances our knowledge of the disease's dynamics but also supports the development of more effective, personalized prevention programs. This method of segmentation paves the way for more nuanced public health strategies in stroke prevention, ultimately aiming to reduce the incidence and impact of stroke across diverse population groups.

Summary of Findings: Through methods of: Linear Regression, Newton-Raphson Method for Logistic Regression, Monte Carlo Simulations, and K-Means Clustering, we identified the most significant predictors of stroke as: age, hypertension, and heart disease. While also observing that marriages, types of work, average glucose levels and BMI all have significance when modeled together.

Impact on Public Health: The integration of advanced statistical techniques has improved the accuracy of stroke prediction. This enables more targeted preventative strategies and treatments, potentially lowering stroke incidence and enhancing patient care.

Reflection on Objectives: Our study achieved key objectives by leveraging diverse analytical methods. However, challenges such as data imbalance and variable significance highlight areas for model improvement.

7. Future Work

- Conduct subgroup analysis to investigate whether the associations between predictors and stroke risk differ across various demographics or clinical groups.
- Create an interactive dashboard that allows users to explore the data, model predictions, and insights dynamically.
- Handle missing data more robustly by using advanced imputation techniques or considering domain-specific knowledge.

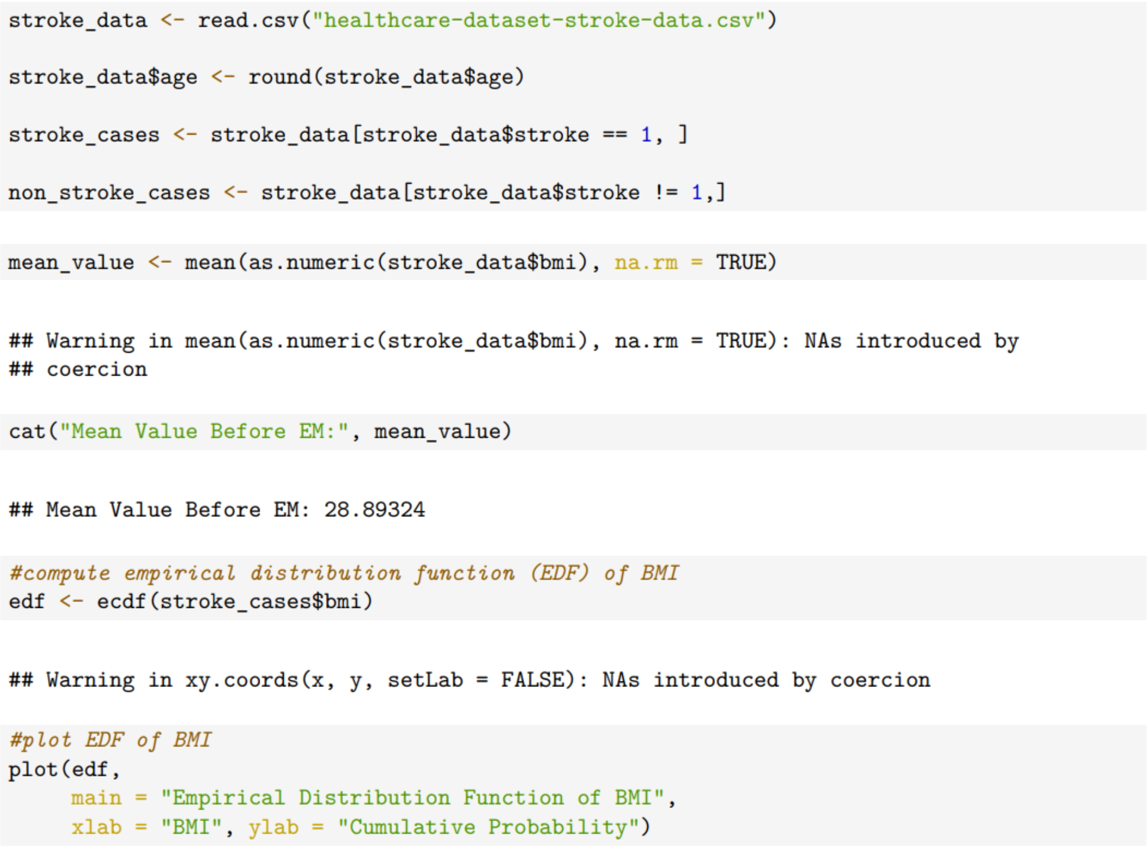
8. Works Cited

Pu, Liyuan, et al. “Projected global trends in ischemic stroke incidence, deaths, and disability-adjusted life years from 2020 to 2030.” Stroke, 2023.

“Stroke Facts.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 4 May 2023, www.cdc.gov/stroke/facts.htm.

9. Appendices

Data Cleaning:



```
stroke_data <- read.csv("healthcare-dataset-stroke-data.csv")

stroke_data$age <- round(stroke_data$age)

stroke_cases <- stroke_data[stroke_data$stroke == 1, ]

non_stroke_cases <- stroke_data[stroke_data$stroke != 1,]
```

```
mean_value <- mean(as.numeric(stroke_data$bmi), na.rm = TRUE)
```

```
## Warning in mean(as.numeric(stroke_data$bmi), na.rm = TRUE): NAs introduced by
## coercion
```

```
cat("Mean Value Before EM:", mean_value)
```

```
## Mean Value Before EM: 28.89324
```

```
#compute empirical distribution function (EDF) of BMI
edf <- ecdf(stroke_cases$bmi)
```

```
## Warning in xy.coords(x, y, setLab = FALSE): NAs introduced by coercion
```

```
#plot EDF of BMI
plot(edf,
     main = "Empirical Distribution Function of BMI",
     xlab = "BMI", ylab = "Cumulative Probability")
```

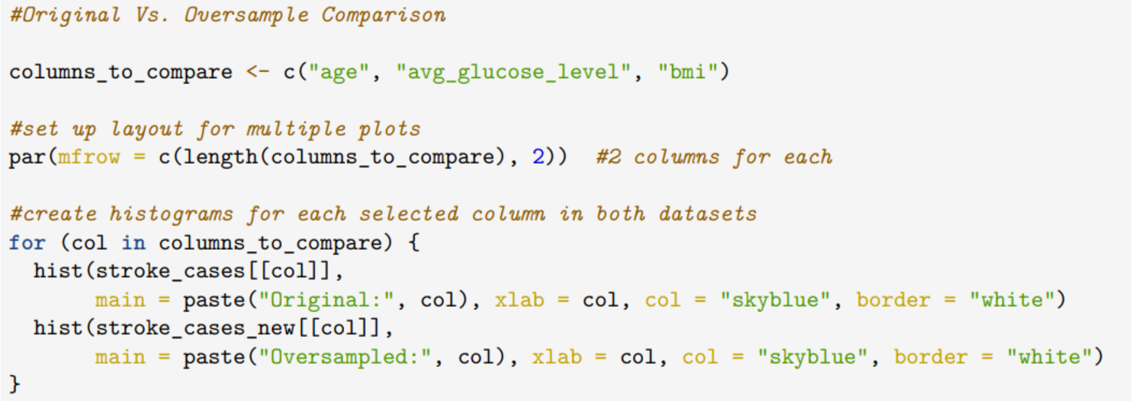
```r
#Expectation Maximization for BMI

#Step 1: Initialization

#replace "N/A" with NA to represent missing values
stroke_data$bmi[stroke_data$bmi == "N/A"] <- NA

#convert BMI
stroke_data$bmi <- as.numeric(stroke_data$bmi)

#Step 2: Expectation-Maximization (EM) Algorithm

#initialize missing values with the mean of observed BMI values
missing_indices <- which(is.na(stroke_data$bmi))
imputed_bmi <- stroke_data$bmi
imputed_bmi[missing_indices] <- mean(stroke_data$bmi, na.rm = TRUE)

## Iteratively Update Missing Values with EM Algorithm ##
max_iter <- 1000
tolerance <- 1e-6

for (iter in 1:max_iter)
{
  #save previous imputed values for comparison
  prev_imputed_bmi <- imputed_bmi

  # E-step: Estimate the parameters
  #(mean and standard deviation) from observed data
  #assuming Normal Distribution
  mu <- mean(imputed_bmi, na.rm = TRUE)
  sigma <- sd(imputed_bmi, na.rm = TRUE)

  # M-step: Impute missing values using estimated parameters
  imputed_bmi[missing_indices] <- rnorm(length(missing_indices),
                                        mean = mu, sd = sigma)

  #check for convergence
  if (max(abs(prev_imputed_bmi[missing_indices]
              - imputed_bmi[missing_indices])) < tolerance)
  {
    break  # Convergence achieved, exit loop
  }
}

#replace missing values with the imputed values
stroke_data$bmi <- imputed_bmi

#check for any missing values left after imputation
if (sum(is.na(stroke_data$bmi)) == 0)
{
  #imputation successful
  print("Missing value imputation using EM algorithm completed.")
}
```

```
mean_value2 <- mean(as.numeric(stroke_data$bmi))
cat("Mean value After EM:", mean_value2)
```

```
## Mean value After EM: 28.93407
```

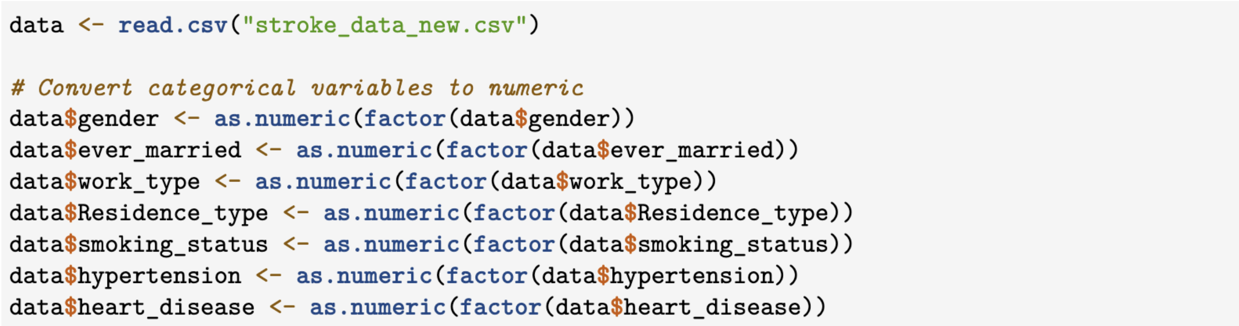
```
#Oversample Stroke Cases

#12% of total - the 249 observed stroke cases
additional_needed <- (0.12 * 5110) - 249

bootstrap_oversampling <- function(stroke_cases, additional_needed)
{
  #initialize oversampled data frame
  oversampled_data <- data.frame(matrix(ncol = ncol(stroke_cases), nrow = 0))
  colnames(oversampled_data) <- colnames(stroke_cases)

  #repeat bootstrap until desired proportion is reached
  while (nrow(oversampled_data[oversampled_data$stroke == 1, ]) < additional_needed)
  {
    #randomly sample observations with replacement
    sampled_indices <- sample(nrow(stroke_cases), size = additional_needed, replace = TRUE)
    sampled_data <- stroke_cases[sampled_indices, ]

    #add sampled data to the oversampled dataset
    oversampled_data <- rbind(oversampled_data, sampled_data)
  }

  #trim excess rows if the desired proportion is exceeded
  oversampled_data <- oversampled_data[1:additional_needed, ]

  return(oversampled_data)
}

stroke_cases <- stroke_data[stroke_data$stroke == 1, ]

bootstrapped_data <- bootstrap_oversampling(stroke_cases, additional_needed)
```

```
stroke_data_new <- rbind(stroke_data, bootstrapped_data)

stroke_cases_new <- stroke_data_new[stroke_data_new$stroke == 1,]

stroke_data_new <- read.csv("stroke_data_new.csv")
```

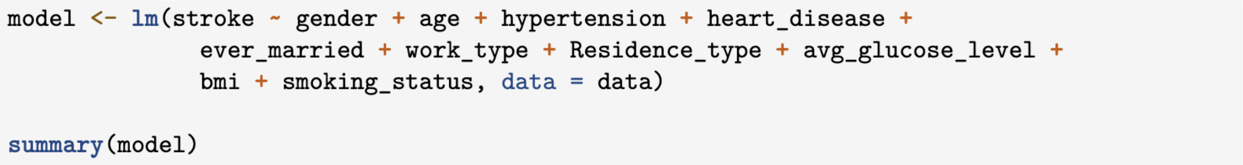


```r
#Original Vs. Oversample Comparison

columns_to_compare <- c("age", "avg_glucose_level", "bmi")

#set up layout for multiple plots
par(mfrow = c(length(columns_to_compare), 2))  #2 columns for each

#create histograms for each selected column in both datasets
for (col in columns_to_compare) {
  hist(stroke_cases[[col]],
       main = paste("Original:", col), xlab = col, col = "skyblue", border = "white")
  hist(stroke_cases_new[[col]],
       main = paste("Oversampled:", col), xlab = col, col = "skyblue", border = "white")
}
```

Linear Regression:



```r
data <- read.csv("stroke_data_new.csv")

# Convert categorical variables to numeric
data$gender <- as.numeric(factor(data$gender))
data$ever_married <- as.numeric(factor(data$ever_married))
data$work_type <- as.numeric(factor(data$work_type))
data$Residence_type <- as.numeric(factor(data$Residence_type))
data$smoking_status <- as.numeric(factor(data$smoking_status))
data$hypertension <- as.numeric(factor(data$hypertension))
data$heart_disease <- as.numeric(factor(data$heart_disease))
```



```r
model <- lm(stroke ~ gender + age + hypertension + heart_disease +
              ever_married + work_type + Residence_type + avg_glucose_level +
              bmi + smoking_status, data = data)

summary(model)
```

Newton-Raphson Method:

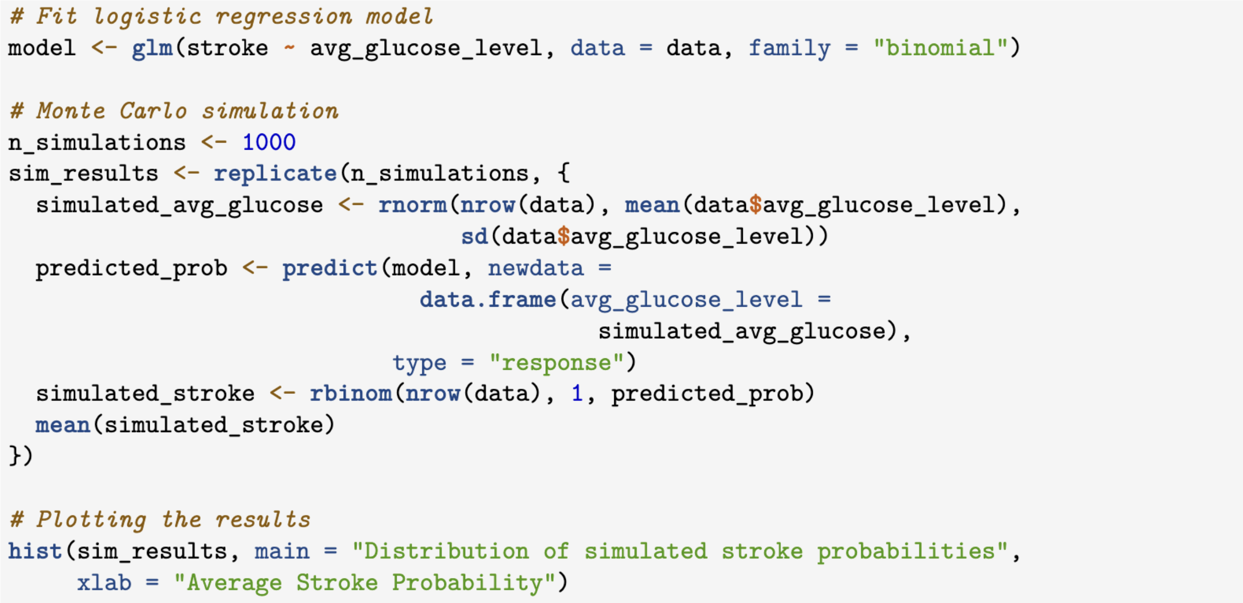
```
# Function to find the root of
f <- function(x) {
  return(predict(model, newdata = data.frame(
    gender = x[1],
    age = x[2],
    hypertension = x[3],
    heart_disease = x[4],
    ever_married = x[5],
    work_type = x[6],
    Residence_type = x[7],
    avg_glucose_level = x[8],
    bmi = x[9],
    smoking_status = x[10]
  )))
}

# Derivative of the function (gradient)
f_prime <- function(x) {
  h <- 1e-5
  gradient <- numeric(length(x))

  for (i in 1:length(x)) {
    x_plus_h <- x
    x_plus_h[i] <- x[i] + h
    gradient[i] <- (f(x_plus_h) - f(x)) / h
  }

  return(gradient)
}

# Initial guess
x0 <- c(1, 44.78144, 0, 0, 1, 1, 1, 107.7767, 28.96, 2)  # Replace with your initial guess
```

```r
# Newton-Raphson method
newton_raphson <- function(f, f_prime, x0, tol = 1e-6, max_iter = 100) {
  x <- x0
  iter <- 0

  while(iter < max_iter) {
    x_new <- x - f_prime(x) * solve(diag(length(x))) %*% f_prime(x)

    # Stopping criterion
    if(sum(abs(x_new - x)) < tol) {
      break
    }

    x <- x_new
    iter <- iter + 1
  }

  if(iter == max_iter) {
    warning("Maximum number of iterations reached without convergence")
  }

  return(x)
}

# Find the root using Newton-Raphson method
root <- newton_raphson(f, f_prime, x0)
```

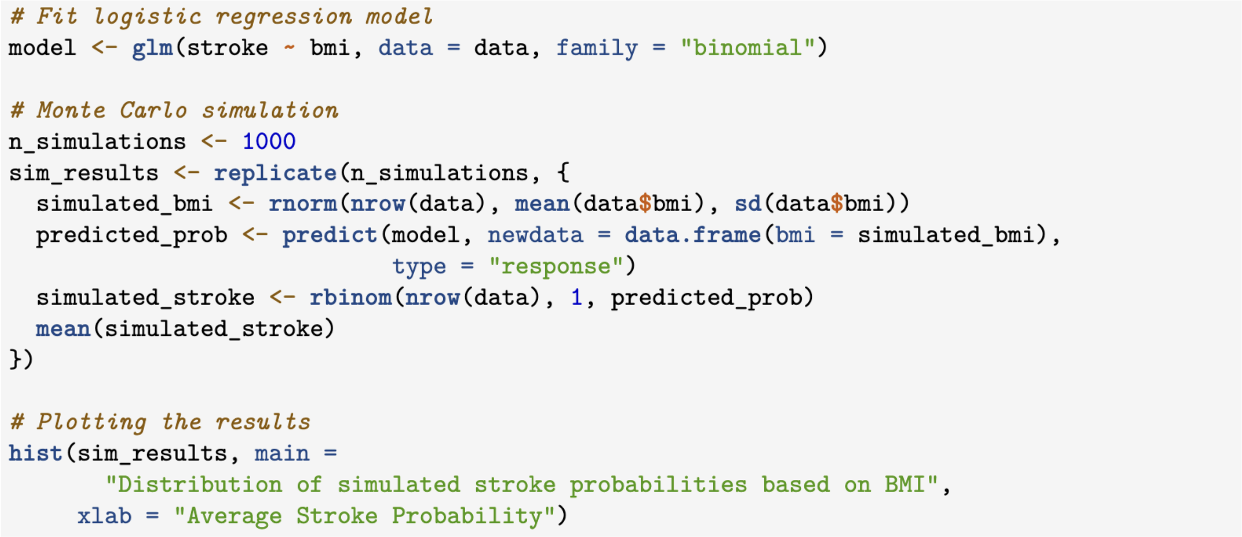
```
## Warning in newton_raphson(f, f_prime, x0): Maximum number of iterations reached
## without convergence
```

```r
print(root)
```

Monte Carlo Method:



```r
# Fit logistic regression model
model <- glm(stroke ~ avg_glucose_level, data = data, family = "binomial")

# Monte Carlo simulation
n_simulations <- 1000
sim_results <- replicate(n_simulations, {
  simulated_avg_glucose <- rnorm(nrow(data), mean(data$avg_glucose_level),
                                 sd(data$avg_glucose_level))
  predicted_prob <- predict(model, newdata =
                              data.frame(avg_glucose_level =
                                           simulated_avg_glucose),
                            type = "response")
  simulated_stroke <- rbinom(nrow(data), 1, predicted_prob)
  mean(simulated_stroke)
})

# Plotting the results
hist(sim_results, main = "Distribution of simulated stroke probabilities",
     xlab = "Average Stroke Probability")
```



```r
# Fit logistic regression model
model <- glm(stroke ~ bmi, data = data, family = "binomial")

# Monte Carlo simulation
n_simulations <- 1000
sim_results <- replicate(n_simulations, {
  simulated_bmi <- rnorm(nrow(data), mean(data$bmi), sd(data$bmi))
  predicted_prob <- predict(model, newdata = data.frame(bmi = simulated_bmi),
                            type = "response")
  simulated_stroke <- rbinom(nrow(data), 1, predicted_prob)
  mean(simulated_stroke)
})

# Plotting the results
hist(sim_results, main =
       "Distribution of simulated stroke probabilities based on BMI",
     xlab = "Average Stroke Probability")
```

Clustering:

```
# %%
import pandas as pd
import numpy as np
from kmodes.kprototypes import KPrototypes
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import LabelEncoder

# %%
# Load data
df = pd.read_csv("/Users/ahmedghabin/Classes/CurrentClasses/STAT4354/Final
Project/stroke_data_new.csv")

# Sample a subset of the data for quicker execution
df = df.sample(frac=0.1, random_state=42)  # Adjust frac to change the sample size

# Define categorical columns
categorical_columns = ['gender', 'hypertension', 'heart_disease', 'ever_married', 'work_type',
'Residence_type', 'smoking_status']

# Convert categorical columns to category types and encode them
for col in categorical_columns:
    df[col] = df[col].astype('category')
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])

# Get the indices of categorical columns
cat_columns_idx = [df.columns.get_loc(col) for col in categorical_columns]

# Define a range of cluster numbers to try
clusters_range = range(2, 11)  # Try clusters from 2 to 10

# Initialize a list to store silhouette scores
silhouette_scores = []

# Iterate over each cluster number
for n_clusters in clusters_range:
    # Initialize K-Prototypes with the current number of clusters
    kproto = KPrototypes(n_clusters=n_clusters, init='Huang', random_state=42)

    # Fit the model on the data
    clusters = kproto.fit_predict(df, categorical=cat_columns_idx)

    # Calculate silhouette score using only numeric data
    numeric_data = df.drop(columns=categorical_columns)
    silhouette_avg = silhouette_score(numeric_data, clusters)
    silhouette_scores.append(silhouette_avg)

# Plot the silhouette scores
plt.plot(clusters_range, silhouette_scores, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score vs Number of Clusters')
plt.show()


# %%
# Load data
df = pd.read_csv("/Users/ahmedghabin/Classes/CurrentClasses/STAT4354/Final
Project/stroke_data_new.csv")

# Identify numeric columns
numeric_columns = df.select_dtypes(include=[np.number]).columns.tolist()

# Fill missing values in numeric columns only
df[numeric_columns] = df[numeric_columns].fillna(df[numeric_columns].mean())


# %%
# Define categorical columns
categorical_columns = ['gender', 'hypertension', 'heart_disease', 'ever_married', 'work_type',
'Residence_type', 'smoking_status']

# Convert categorical columns to category types
for col in categorical_columns:
    df[col] = df[col].astype('category')

# Get the indices of categorical columns
cat_columns_idx = [df.columns.get_loc(col) for col in categorical_columns]


# %%
# Initialize K-Prototypes model
kproto = KPrototypes(n_clusters=2, init='Huang', random_state=42)

# Fit the model on the dataset
clusters = kproto.fit_predict(df, categorical=cat_columns_idx)

# Add the cluster labels to the dataframe
df['Cluster'] = clusters


# %%
# Visualize clusters
```

```
sns.pairplot(df, hue='Cluster', vars=['age', 'avg_glucose_level', 'bmi'])
plt.show()
```