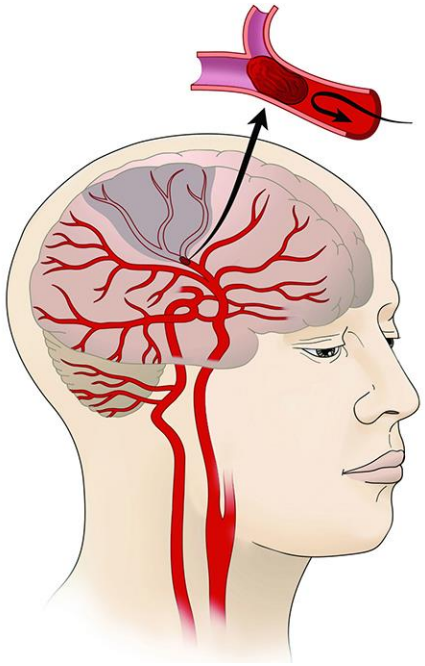
EARLY STROKE PREDICTION USING NAÏVE BAYESIAN TECHNIQUE

Bayesian Statistics – II

ST – 406

Final Project



H.H. Pramod Nipunajith

S/17/436

Department of Statistics and Computer Science

Introduction

Stroke is a serious global health issue that affects people everywhere, regardless of where they live or their background. It happens when something disrupts the flow of blood to the brain, causing harm and, in many cases, leading to death. Shockingly, it stands as the second leading cause of death worldwide, making up about 11% of all recorded deaths, according to the World Health Organization (WHO).

What makes stroke particularly challenging is that it happens quietly inside the complex structure of the brain. Unlike some other health emergencies, it requires quick and accurate action for the best possible results.

Detecting stroke early is crucial for effective treatment, as it helps stop the chain of events triggered by the interruption of blood flow. This project is all about using advanced statistical methods called Bayesian techniques to improve the early identification of stroke.

Methodology

Data set description

This dataset is used to predict whether a patient is likely to get stroke based on the input parameters like gender, age, various diseases, and smoking status. Each row in the data provides relavant information about the patient. The attribute information is given as follows. (data source: https://www.kaggle.com/code/ashokkumarpalivela/stroke-prediction-end-to-end-project/input)

Variable	Description
id	unique identifier
gender	"Male", "Female" or "Other"
age	age of the patient
hypertension	0 if the patient doesn't have hypertension, 1 if the patient has hypertension
heart_disease	0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
ever_married	"No" or "Yes"
work_type	"children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
Residence_type	"Rural" or "Urban"
avg_glucose_level	average glucose level in blood
bmi	body mass index
smoking_status	"formerly smoked", "never smoked", "smokes" or "Unknown"*
stroke	1 if the patient had a stroke or 0 if not

**Note: "Unknown" in smoking_status means that the information is unavailable for this patient.*

Statistical Method

This study employs Bayesian techniques, specifically Naive Bayesian Classifier to identify early signs of stroke. The Naive Bayesian approach assumes independence between predictors, providing a straightforward initial assessment of the individual impact of each variable. Bayesian multiple linear regression, on the other hand, considers the relationships between multiple predictors, offering a more nuanced understanding of their combined influence on the likelihood of a stroke.

Dealing with/Overcoming Assumption Violations

To address potential violations of assumptions, such as normality of residuals in linear regression, we utilize Bayesian methods that are inherently robust and less reliant on strict assumptions. Bayesian statistics, by nature, allows for more flexibility in modeling complex relationships without being overly constrained by assumptions that may not hold in real-world scenarios.

Detailed Procedure

The procedure that I have to deal can be state as follows. There I have included all the steps from data preprocessing to model evaluation.

Import Dataset

```
stroke <- read_csv('../Data/healthcare-dataset-stroke-data.csv')
```

```
## Rows: 5110 Columns: 12
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr (6): gender, ever_married, work_type, Residence_type, bmi,
smoking_status
## dbl (6): id, age, hypertension, heart_disease, avg_glucose_level, stroke
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
## Display head of the dataset
head(stroke)
```

```
## # A tibble: 6 × 12
##      id gender   age hypertension heart_disease ever_married work_type
##   <dbl> <chr>  <dbl>        <dbl>         <dbl> <chr>        <chr>
## 1  9046 Male      67            0             1 Yes          Private
## 2 51676 Female    61            0             0 Yes          Self-employed
## 3 31112 Male      80            0             1 Yes          Private
## 4 60182 Female    49            0             0 Yes          Private
## 5  1665 Female    79            1             0 Yes          Self-employed
## 6 56669 Male      81            0             0 Yes          Private
## # ℹ 5 more variables: Residence_type <chr>, avg_glucose_level <dbl>, bmi
<chr>,
## #   smoking_status <chr>, stroke <dbl>
```

```r
## Look at the structure of the variables
str(stroke)
```

```
## spc_tbl_ [5,110 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id               : num [1:5110] 9046 51676 31112 60182 1665 ...
##  $ gender           : chr [1:5110] "Male" "Female" "Male" "Female" ...
##  $ age              : num [1:5110] 67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : num [1:5110] 0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : num [1:5110] 1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr [1:5110] "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr [1:5110] "Private" "Self-employed" "Private"
"Private" ...
##  $ Residence_type   : chr [1:5110] "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num [1:5110] 229 202 106 171 174 ...
##  $ bmi              : chr [1:5110] "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr [1:5110] "formerly smoked" "never smoked" "never
smoked" "smokes" ...
##  $ stroke           : num [1:5110] 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   gender = col_character(),
##   ..   age = col_double(),
##   ..   hypertension = col_double(),
##   ..   heart_disease = col_double(),
##   ..   ever_married = col_character(),
##   ..   work_type = col_character(),
##   ..   Residence_type = col_character(),
##   ..   avg_glucose_level = col_double(),
##   ..   bmi = col_character(),
##   ..   smoking_status = col_character(),
##   ..   stroke = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
## Describe the dataset
glimpse(stroke)
```

```
## Rows: 5,110
## Columns: 12
## $ id                <dbl> 9046, 51676, 31112, 60182, 1665, 56669, 53882,
10434…
## $ gender            <chr> "Male", "Female", "Male", "Female", "Female",
"Male"…
## $ age               <dbl> 67, 61, 80, 49, 79, 81, 74, 69, 59, 78, 81, 61,
54, …
## $ hypertension      <dbl> 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1,
0, 1…
## $ heart_disease     <dbl> 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0,
```

```
1, 0…
## $ ever_married      <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",
"No…
## $ work_type         <chr> "Private", "Self-employed", "Private",
"Private", "S…
## $ Residence_type    <chr> "Urban", "Rural", "Rural", "Urban", "Rural",
"Urban"…
## $ avg_glucose_level <dbl> 228.69, 202.21, 105.92, 171.23, 174.12, 186.21,
70.0…
## $ bmi               <chr> "36.6", "N/A", "32.5", "34.4", "24", "29",
"27.4", "…
## $ smoking_status    <chr> "formerly smoked", "never smoked", "never
smoked", "…
## $ stroke            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1…
```

```
## Find the dimension of the dataset
dim(stroke)
```

```
## [1] 5110   12
```

- There are 5110 observations that belongs to the 12 variables.

Data Preprocessing

```
## Count missing values in each column
null_counts <- colSums(is.na(stroke))

## Convert the result to a data frame with column name "Null count"
result <- data.frame("Null count" = null_counts)

## Print the result
print(result)
```

```
##                   Null.count
## id                         0
## gender                     0
## age                        0
## hypertension               0
## heart_disease              0
## ever_married               0
## work_type                  0
## Residence_type             0
## avg_glucose_level          0
```

```
## bmi                          0
## smoking_status               0
## stroke                       0
```

```
print(result)
```

```
##               Column Missing_Count
## 1                 id             0
## 2             gender             0
## 3                age             0
## 4       hypertension             0
## 5      heart_disease             0
## 6       ever_married             0
## 7          work_type             0
## 8     Residence_type             0
## 9  avg_glucose_level             0
## 10               bmi           201
## 11    smoking_status             0
## 12            stroke             0
```

```
## Convert column A to numeric and fill missing values with the mean
stroke$bmi <- as.numeric(as.character(stroke$bmi))
```

```
## Warning: NAs introduced by coercion
```

```
mean_A <- mean(stroke$bmi, na.rm = TRUE)
```

```
## Replace missing values in column A with the mean
stroke$bmi[is.na(stroke$bmi)] <- mean_A
```

```
## Print the corrected data
print(stroke)
```

```
## # A tibble: 5,110 × 12
##       id gender   age hypertension heart_disease ever_married work_type
##    <dbl> <chr>  <dbl>        <dbl>         <dbl> <chr>        <chr>
##  1  9046 Male      67            0             1 Yes          Private
##  2 51676 Female    61            0             0 Yes          Self-
employed
##  3 31112 Male      80            0             1 Yes          Private
##  4 60182 Female    49            0             0 Yes          Private
##  5  1665 Female    79            1             0 Yes          Self-
employed
##  6 56669 Male      81            0             0 Yes          Private
##  7 53882 Male      74            1             1 Yes          Private
##  8 10434 Female    69            0             0 No           Private
##  9 27419 Female    59            0             0 Yes          Private
## 10 60491 Female    78            0             0 Yes          Private
## # ℹ 5,100 more rows
```

```
## # ℹ 5 more variables: Residence_type <chr>, avg_glucose_level <dbl>, bmi
<dbl>,
## #   smoking_status <chr>, stroke <dbl>
```

- There are 201 NA values in 'bmi' variable. Treated for them by using the mean of the bmi as 28.89.

Exploratory Data Analysis

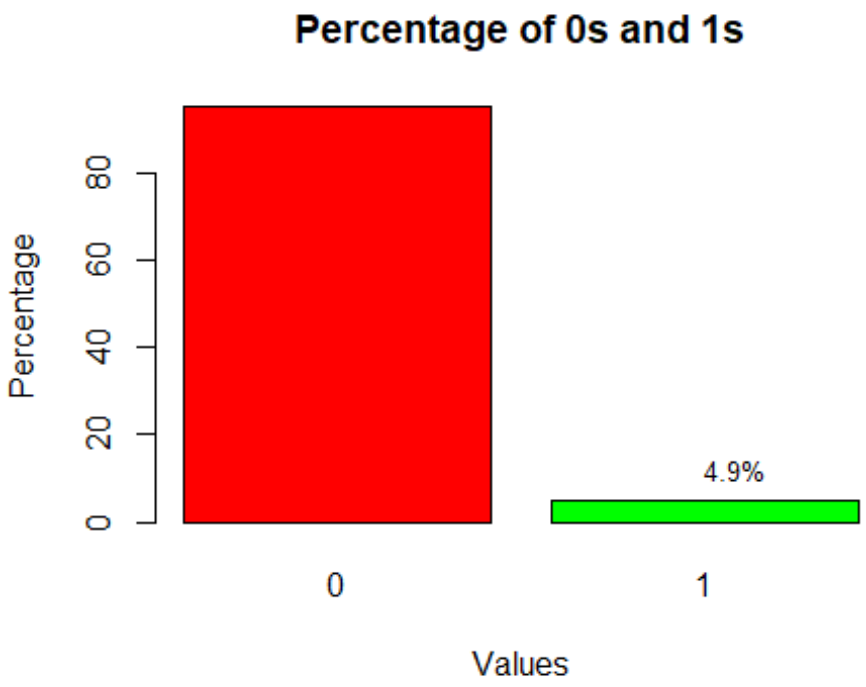
```
## Display summary statistics
summary(stroke)
```

```
##        id            gender                age          hypertension
##  Min.   :   67   Length:5110        Min.   : 0.08   Min.   :0.00000
##  1st Qu.:17741   Class :character   1st Qu.:25.00   1st Qu.:0.00000
##  Median :36932   Mode  :character   Median :45.00   Median :0.00000
##  Mean   :36518                      Mean   :43.23   Mean   :0.09746
##  3rd Qu.:54682                      3rd Qu.:61.00   3rd Qu.:0.00000
##  Max.   :72940                      Max.   :82.00   Max.   :1.00000
##  heart_disease     ever_married        work_type         Residence_type
##  Min.   :0.00000   Length:5110        Length:5110        Length:5110
##  1st Qu.:0.00000   Class :character   Class :character   Class :character
##  Median :0.00000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.05401
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  avg_glucose_level      bmi         smoking_status         stroke
##  Min.   : 55.12    Min.   :10.30   Length:5110        Min.   :0.00000
##  1st Qu.: 77.25    1st Qu.:23.80   Class :character   1st Qu.:0.00000
##  Median : 91.89    Median :28.40   Mode  :character   Median :0.00000
##  Mean   :106.15    Mean   :28.89                      Mean   :0.04873
##  3rd Qu.:114.09    3rd Qu.:32.80                      3rd Qu.:0.00000
##  Max.   :271.74    Max.   :97.60                      Max.   :1.00000
```

- Obtained the summary statistics for numerical variables in the data set.

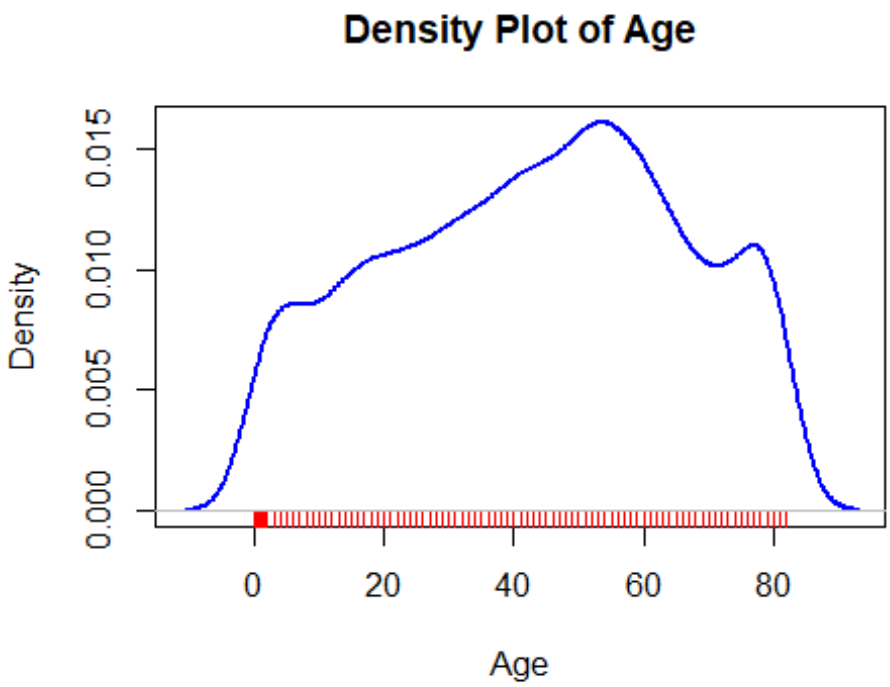
Summary statistics	age	avg_glucose_level	bmi
Min	0.08	55.12	10.30
Max	82.00	271.74	97.60
Mean	43.23	106.15	28.89
1st Quartile	25.00	77.25	23.80
Median	45.00	91.89	28.40
3rd Quartile	61.00	114.09	32.80

```
# Calculate the percentage of each value
percentage_0 <- (sum(stroke$stroke == 0) / length(stroke$stroke)) * 100
percentage_1 <- (sum(stroke$stroke == 1) / length(stroke$stroke)) * 100

# Create a bar plot
bar_heights <- c(percentage_0, percentage_1)
bar_names <- c('0', '1')

# Plot the bar graph
barplot(bar_heights, names.arg = bar_names, col = c('red', 'green'),
        xlab = 'Values', ylab = 'Percentage',
        main = 'Percentage of 0s and 1s')

# Add percentage values on top of each bar
text(seq_along(bar_heights), bar_heights, labels = sprintf("%.1f%%",
bar_heights),
     pos = 3, cex = 0.8, col = 'black', font = 1)
```



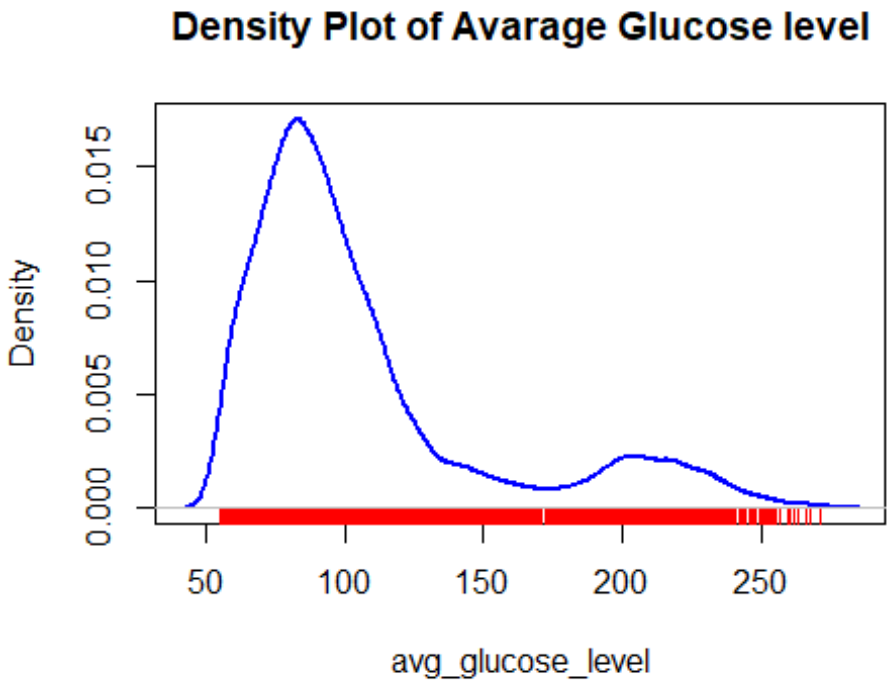
- 4.9% of the target variable is 1s (stroke presence) and 95.1% of the target variable is 0s (stroke absence)

```
## Density plot for age
plot(density(stroke$age), main = "Density Plot of Age", xlab = "Age", col =
"blue", lwd = 2)

# Add a rug plot for individual data points
rug(stroke$age, col = "red")
```



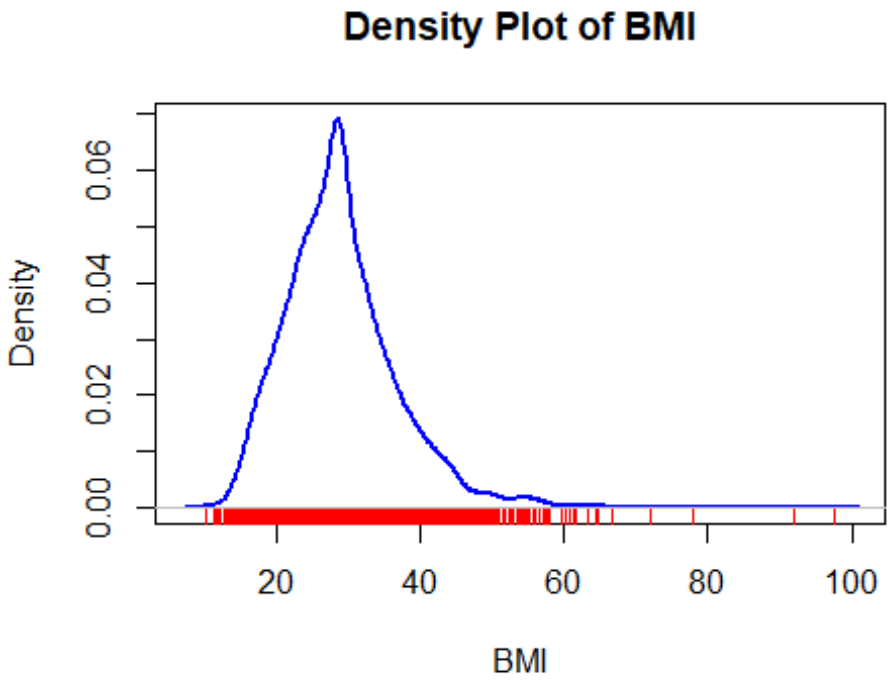
- The distribution of the age variable is aligned with 0.08 and 82.00 and also most of the data points are contain within the 45 - 60 range. This distribution may not have exact shape like normal curve.

```
## Density plot for age
plot(density(stroke$avg_glucose_level), main = "Density Plot of Avarage
Glucose level", xlab = "avg_glucose_level", col = "blue", lwd = 2)

## Add a rug plot for individual data points
rug(stroke$avg_glucose_level, col = "red")
```



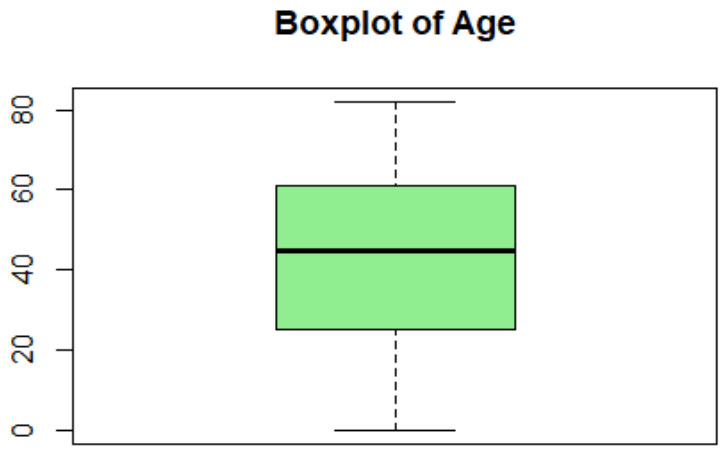
- The data points of the avg_glucose_level from 55.12 to 125 portion of the above graph behave as the normal distribution.

```
## Density plot for age
plot(density(stroke$bmi), main = "Density Plot of BMI", xlab = "BMI", col =
"blue", lwd = 2)

## Add a rug plot for individual data points
rug(stroke$bmi, col = "red")
```



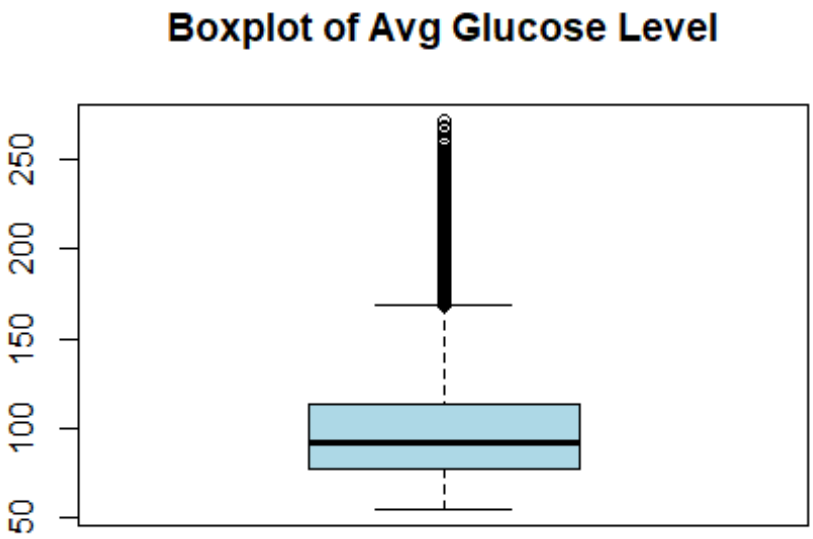
- From 10.30 to 50 part of the above graph has a unique shape approximately normal.

```
## Boxplot for age
boxplot(stroke$age, main = "Boxplot of Age", col = "lightgreen", border =
"black")
```



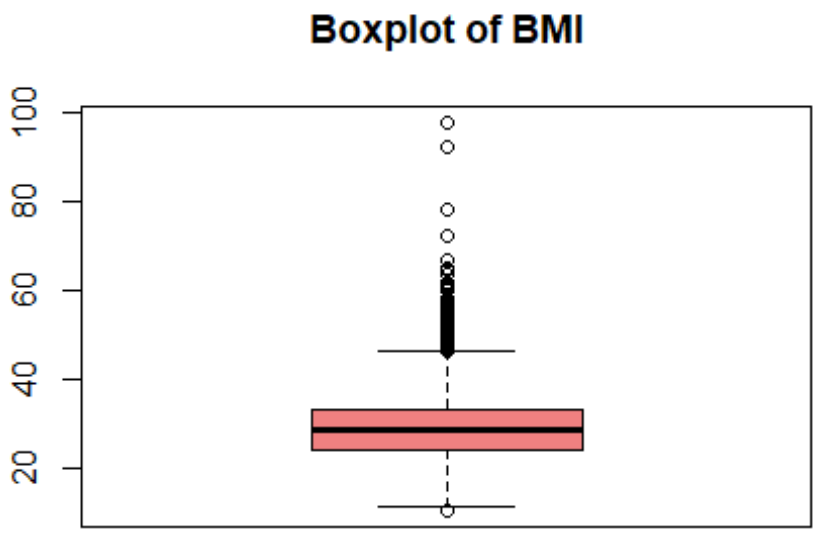
- By looking at the above boxplot, I can conclude that there are no any outlier observations detected in the age variable.

```
## Boxplot for avg_glucose_level
boxplot(stroke$avg_glucose_level, main = "Boxplot of Avg Glucose Level", col
= "lightblue", border = "black")
```



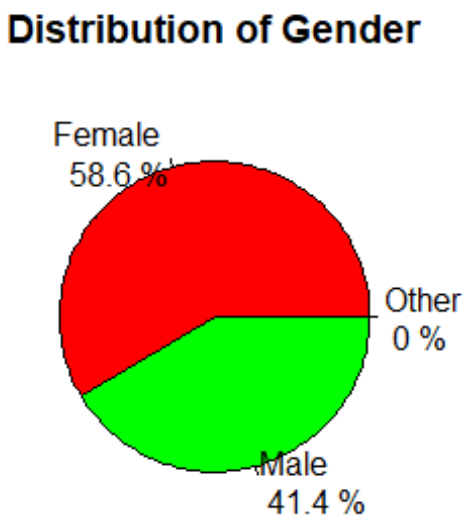
- When consider the average glucose level, there are some outliers can be detected.

```
## Boxplot for bmi
boxplot(stroke$bmi, main = "Boxplot of BMI", col = "lightcoral", border =
"black")
```



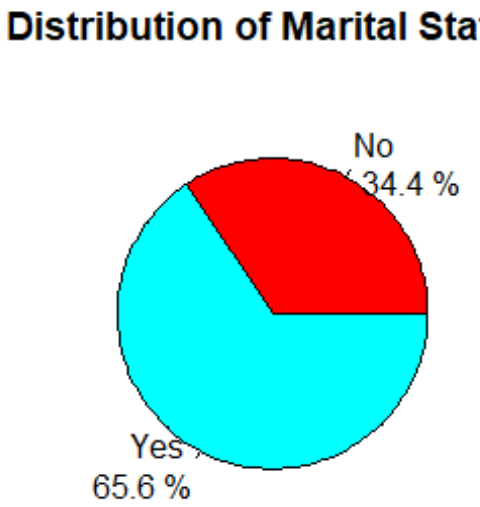
- By looking at the above plot also we can identify some outlier observations in the bmi variable.

```
## Create a pie chart for gender
gender_counts <- table(stroke$gender)
gender_percentages <- round(prop.table(gender_counts) * 100, 1)
pie(gender_counts, labels = paste(names(gender_counts), "\n",
gender_percentages, "%"), main = "Distribution of Gender", col =
rainbow(length(gender_counts)))
```



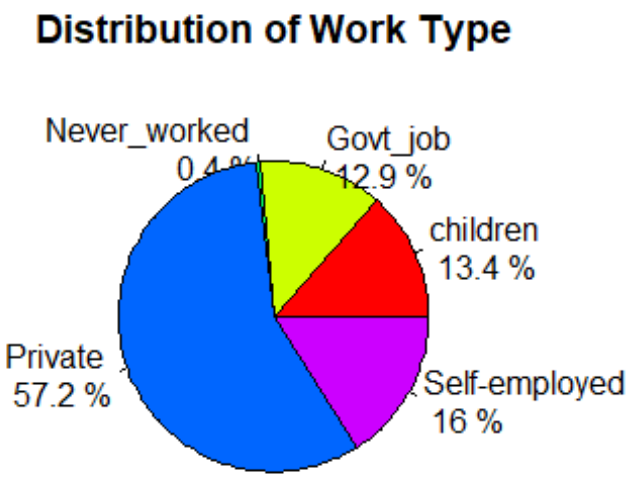
- The distribution represents the gender makeup of patients. 58.6% of the patients are female and remaining is male patients.

```
## Create a pie chart for ever_married
married_counts <- table(stroke$ever_married)
married_percentages <- round(prop.table(married_counts) * 100, 1)
pie(married_counts, labels = paste(names(married_counts), "\n",
married_percentages, "%"), main = "Distribution of Marital Status", col =
rainbow(length(married_counts)))
```



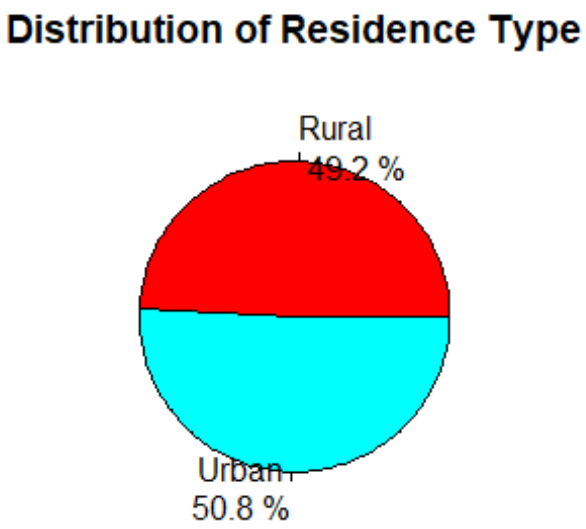
- The “Yes" slice is larger, it suggests a higher percentage of the population (65.6%) has been married. Conversely, "No" slice indicates a smaller percentage (34.4%) of individuals who have not been married.

```
## Create a pie chart for work_type
work_counts <- table(stroke$work_type)
work_percentages <- round(prop.table(work_counts) * 100, 1)
pie(work_counts, labels = paste(names(work_counts), "\n", work_percentages,
"%"), main = "Distribution of Work Type", col = rainbow(length(work_counts)))
```



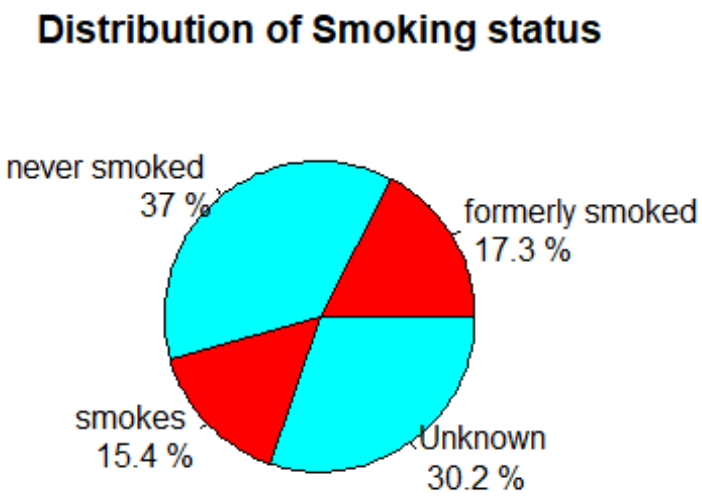
- This pie chart shows 57.2% "Private", 16% "Self-employed", 13.4% "Children", 12.9% "Government job" and 0.4% “Never worked” where private employment is the dominant work type, followed by self-employment, children and government employment, with a smaller percentage of individuals being unemployed.

```
## Create a pie chart for Residence_type
residence_counts <- table(stroke$Residence_type)
residence_percentages <- round(prop.table(residence_counts) * 100, 1)
pie(residence_counts, labels = paste(names(residence_counts), "\n",
residence_percentages, "%"), main = "Distribution of Residence Type", col =
rainbow(length(residence_counts)))
```



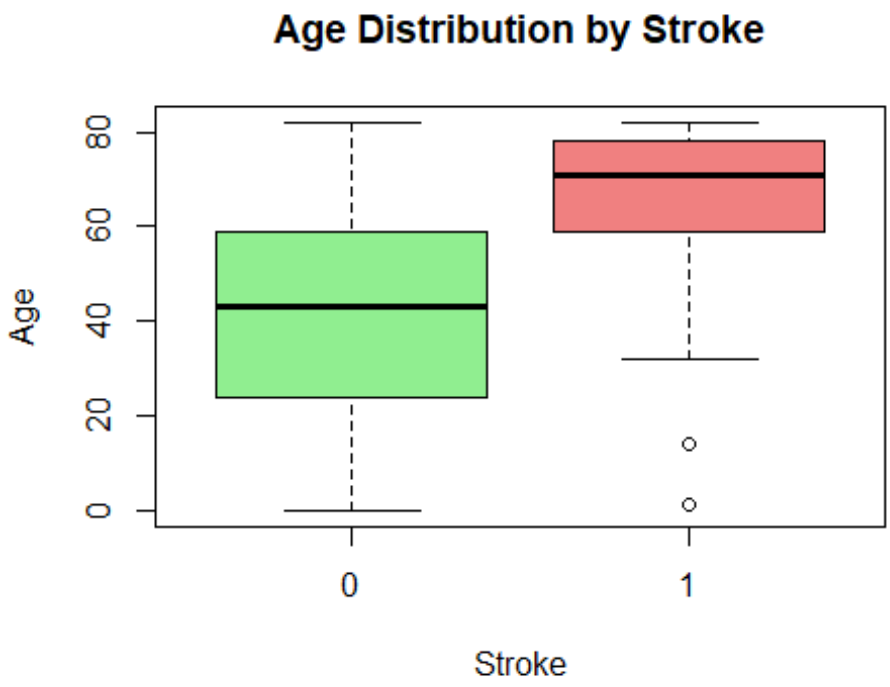
- "Urban" slice is larger with percentage of 50.8% and it suggests that a significant majority of the population resides in urban areas.

```
## Create a pie chart for Residence_type
smoke_counts <- table(stroke$smoking_status)
smoke_percentages <- round(prop.table(smoke_counts) * 100, 1)
pie(smoke_counts, labels = paste(names(smoke_counts), "\n",
smoke_percentages, "%"), main = "Distribution of Smoking status", col =
rainbow(length(residence_counts)))
```



- The distribution shows 37% "Never smoked", 30.2% "Unknown", 17.3% "Formerly smoked" and 15.4% “Smokes” with a majority of non-smokers, a significant proportion of unknow smokers, former smokers, and a relatively lower percentage of current smokers.

```
## Boxplot for age with respect to stroke
boxplot(age ~ stroke, data = stroke, main = "Age Distribution by Stroke",
xlab = "Stroke", ylab = "Age", col = c("lightgreen", "lightcoral"))
```

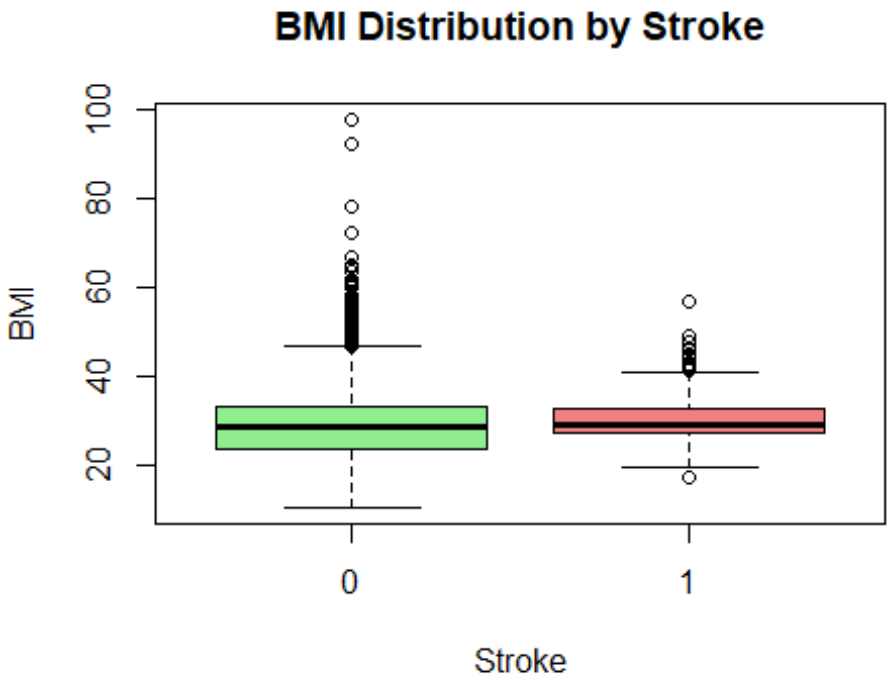


- In the boxplot depicting the relationship between stroke and age, outliers in the "stroke" category 1 suggest that there are individuals who experienced a stroke at ages significantly different from the typical range.

```
## Boxplot for avg_glucose_level with respect to stroke
boxplot(avg_glucose_level ~ stroke, data = stroke, main = "Avg Glucose Level
by Stroke", xlab = "Stroke", ylab = "Avg Glucose Level", col =
c("lightgreen", "lightcoral"))
```

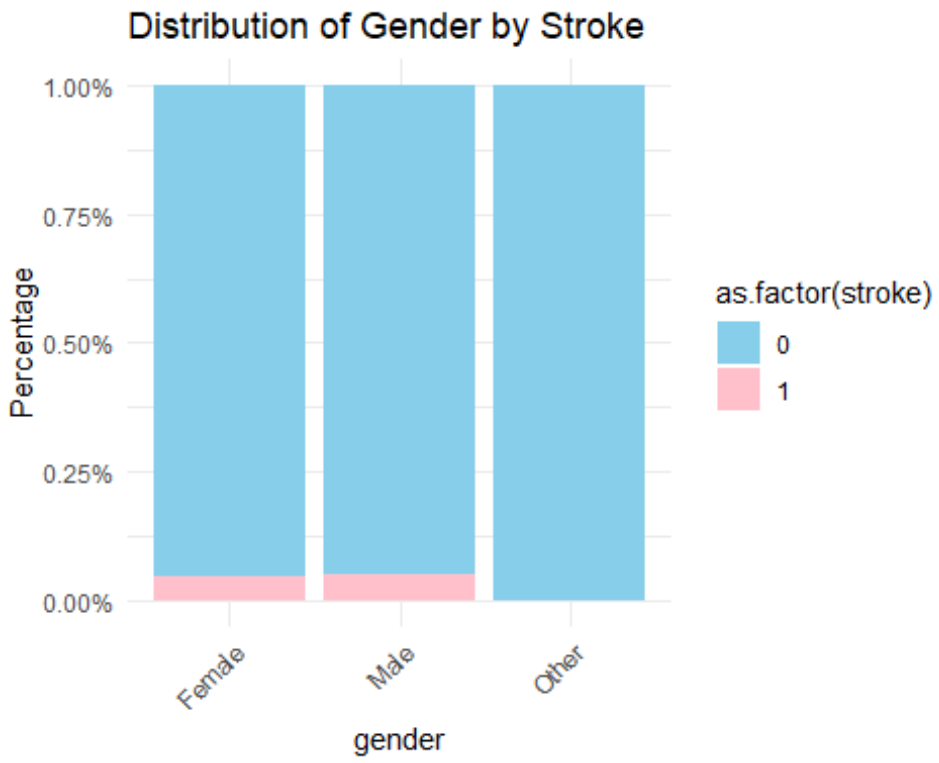
- The presence of outliers in the "no stroke" category for average glucose levels suggests significant variability in glucose levels among individuals without a history of stroke. Elevated or lower glucose levels in these outliers may indicate potential health risks or conditions such as diabetes.

```
## Boxplot for bmi with respect to stroke
boxplot(bmi ~ stroke, data = stroke, main = "BMI Distribution by Stroke",
xlab = "Stroke", ylab = "BMI", col = c("lightgreen", "lightcoral"))
```



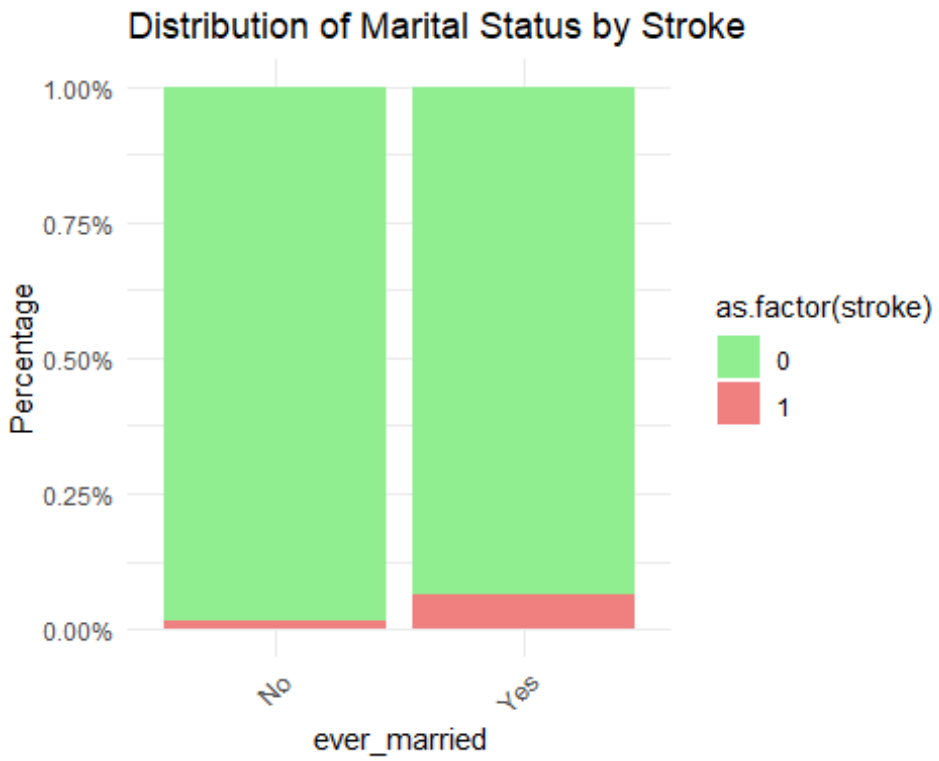
- The boxplot depicting BMI and stroke status reveals outliers in both the "stroke" and "no stroke" categories. Outliers in the "stroke" group may suggest individuals with extreme BMI values who experienced a stroke.

```
## Function to calculate percentages and create bar plots with different
colors
plot_categorical_variable <- function(variable, title, colors) {
  percentages <- prop.table(table(stroke[[variable]], stroke$stroke), margin
= 2) * 100
  ggplot(stroke, aes(x = get(variable), fill = as.factor(stroke))) +
    geom_bar(position = "fill", stat = "count") +
    scale_y_continuous(labels = scales::percent_format(scale = 1)) +
    labs(title = title, x = variable, y = "Percentage") +
    scale_fill_manual(values = colors) +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

## Create bar plots for each categorical variable with different colors
plot_categorical_variable("gender", "Distribution of Gender by Stroke",
c("skyblue", "pink"))
```



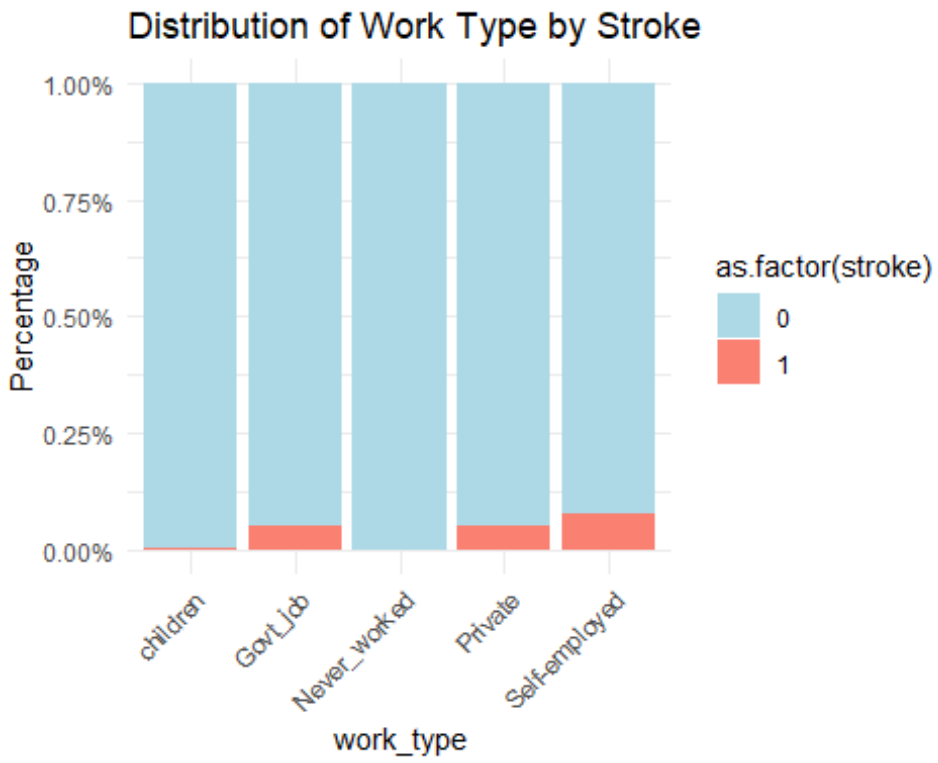
- This distribution reveals an equal percentage of strokes in both males and females. This suggests a gender-neutral pattern indicating that in the studied population, the risk of experiencing a stroke is similar for both genders.

```
plot_categorical_variable("ever_married", "Distribution of Marital Status by
Stroke", c("lightgreen", "lightcoral"))
```



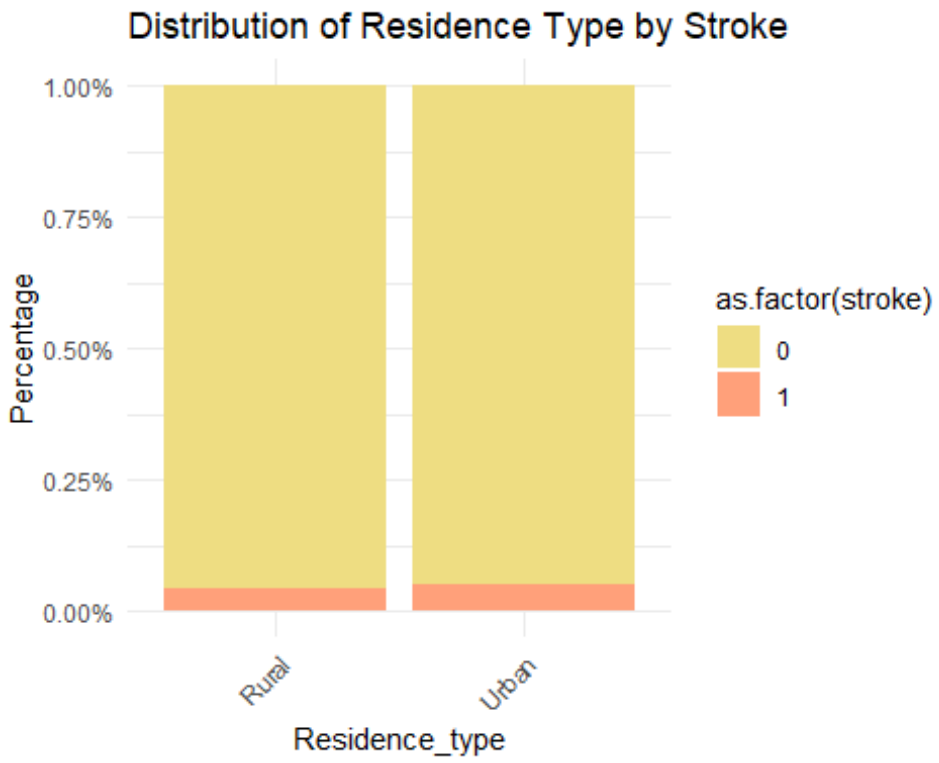
- A notable finding in the distribution of marital status by stroke indicates a higher percentage of patients who have ever married. This observation suggests a potential association between marital status and stroke risk.

```
plot_categorical_variable("work_type", "Distribution of Work Type by Stroke",
c("lightblue", "salmon"))
```



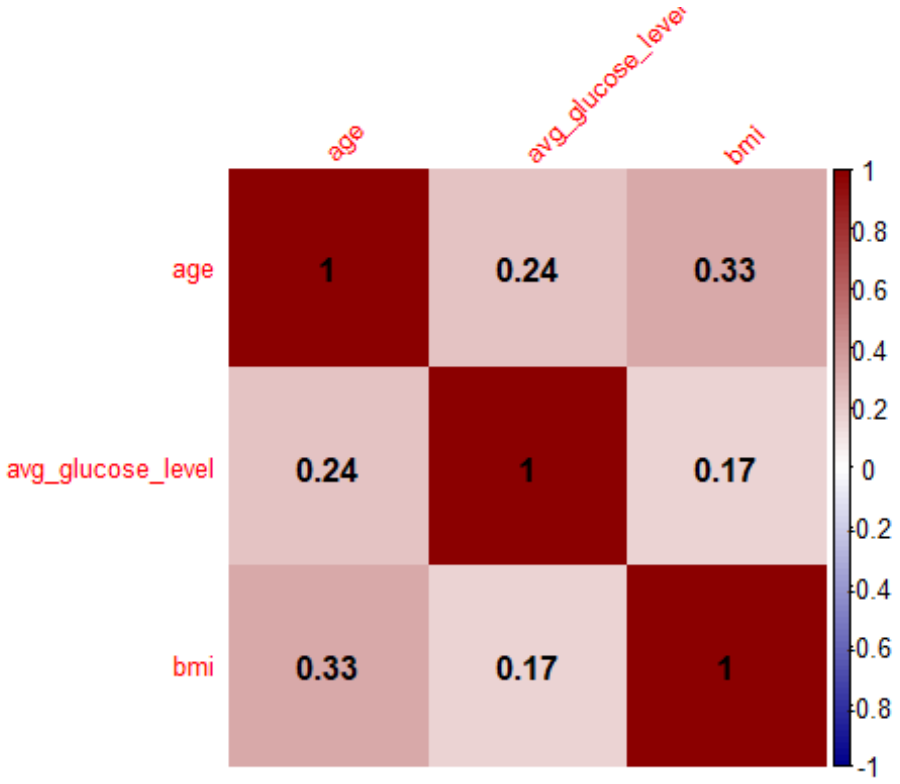
- The distribution of work types by stroke reveals a higher percentage of self-employed patients. This suggests a potential association between being self-employed and the risk of stroke.

```
plot_categorical_variable("Residence_type", "Distribution of Residence Type
by Stroke", c("lightgoldenrod", "lightsalmon"))
```



- The distribution of residence types by stroke indicates that the same percentage of urban and rural patients have experienced a stroke. This suggests an equal risk of stroke across urban and rural populations.

```
## Calculate the correlation matrix
cor_matrix <- cor(stroke[, c("age", "avg_glucose_level", "bmi")])

## Plot the correlation matrix with an attractive color scale
corrplot(cor_matrix, method = "color", col = colorRampPalette(c("darkblue",
"white", "darkred"))(100),
         addCoef.col = "black", tl.cex = 0.8, tl.srt = 45)
```

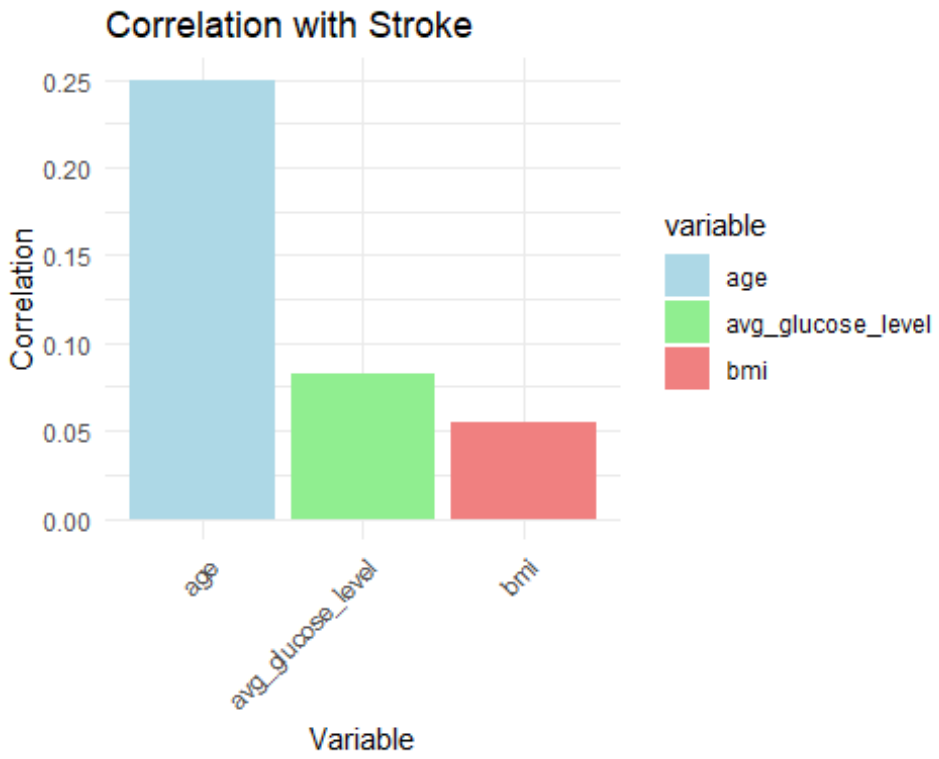


- The correlation matrix shows a modest positive association between age and average glucose level (0.24), a moderate positive correlation between age and BMI (0.33), and a weak positive correlation between average glucose level and BMI (0.17). These findings suggest varying degrees of interdependence among age, BMI, and average glucose levels in the dataset.

```
## Select numerical variables for correlation analysis
numerical_variables <- c("age", "avg_glucose_level", "bmi")

## Create an empty data frame to store correlation results
cor_data_long <- data.frame()

## Loop through numerical variables and calculate correlations
for (variable in numerical_variables) {
  correlation_result <- cor.test(stroke[[variable]], stroke$stroke, method =
"spearman")

## Append results to the data frame
  cor_data_long <- rbind(cor_data_long, data.frame(
    variable = variable,
    correlation = correlation_result$estimate,
    p_value = correlation_result$p.value
```

```
  ))
}

## Warning in cor.test.default(stroke[[variable]], stroke$stroke, method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(stroke[[variable]], stroke$stroke, method =
## "spearman"): Cannot compute exact p-value with ties

## Warning in cor.test.default(stroke[[variable]], stroke$stroke, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
## Create a bar plot
ggplot(cor_data_long, aes(x = variable, y = correlation, fill = variable)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Correlation with Stroke", x = "Variable", y = "Correlation")
+
  scale_fill_manual(values = c("lightblue", "lightgreen", "lightcoral")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Data Modelling

```
## Extract features (X) and target variable (y)
X <- stroke[, !(names(stroke) %in% c("stroke"))]
y <- stroke$stroke
## Split the dataset into training and testing sets
set.seed(42)  # for reproducibility
split_index <- createDataPartition(y, p = 0.8, list = FALSE)
train_data <- stroke[split_index, ]
test_data <- stroke[-split_index, ]

## Create and train the Naive Bayes model
model <- naiveBayes(stroke ~ ., data = train_data, laplace = 1)
model

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.94985323 0.05014677
##
## Conditional probabilities:
##    id
## Y       [,1]     [,2]
##   0 36373.90 21156.99
##   1 35941.06 21795.54
##
##    gender
## Y      Female      Male
##   0 0.5851146 0.4154005
##   1 0.5707317 0.4390244
##
##    age
## Y       [,1]     [,2]
##   0 41.84808 22.22402
##   1 68.01132 12.37233
##
##    hypertension
## Y         [,1]      [,2]
##   0 0.08653103 0.2811828
##   1 0.24878049 0.4333646
##
##    heart_disease
```

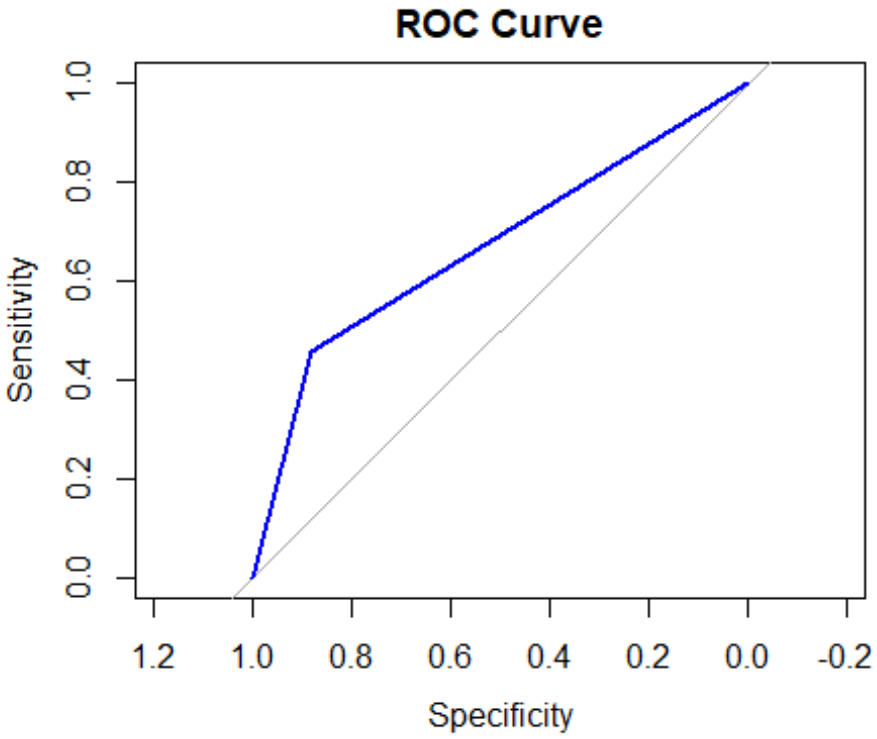
```
## Y         [,1]      [,2]
##   0 0.04661344 0.2108366
##   1 0.20487805 0.4046005
##
##    ever_married
## Y          No       Yes
##   0 0.3533351 0.6471800
##   1 0.1268293 0.8829268
##
##    work_type
## Y      children    Govt_job Never_worked     Private Self-employed
##   0 0.140870461 0.128251352  0.005150657 0.566829771   0.160185424
##   1 0.009756098 0.141463415  0.004878049 0.609756098   0.258536585
##
##    Residence_type
## Y       Rural     Urban
##   0 0.4949781 0.5055370
##   1 0.4536585 0.5560976
##
##    avg_glucose_level
## Y       [,1]     [,2]
##   0 104.4024 43.54058
##   1 134.9826 61.27986
##
##    bmi
## Y       [,1]    [,2]
##   0 28.84367 7.86192
##   1 30.14919 5.75123
##
##    smoking_status
## Y   formerly smoked never smoked    smokes   Unknown
##   0       0.1648210    0.3731651 0.1480814 0.3149627
##   1       0.2780488    0.3707317 0.1756098 0.1951220
```

```r
## Make predictions on the testing set
predictions <- predict(model, newdata = test_data)

## Evaluate the model
conf_matrix <- table(predictions, test_data$stroke)
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.8620352
```

```r
## Display confusion matrix for more detailed metrics
print(conf_matrix)
```

```
##
## predictions   0   1
##           0 861  24
##           1 117  20

# Evaluate the model
conf_matrix <- table(predictions, test_data$stroke)
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
precision <- conf_matrix[2, 2] / sum(conf_matrix[, 2])
recall <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
f1_score <- 2 * (precision * recall) / (precision + recall)

cat("Accuracy:", accuracy, "\n")

## Accuracy: 0.8620352

cat("Precision:", precision, "\n")

## Precision: 0.4545455

cat("Recall:", recall, "\n")

## Recall: 0.1459854

cat("F1 Score:", f1_score, "\n")

## F1 Score: 0.2209945



## Assuming 'predictions' and 'test_data$stroke' are probabilities
roc_curve <- roc(test_data$stroke, as.numeric(predictions == 1))

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

plot(roc_curve, main = "ROC Curve", col = "blue", lwd = 2)
```



- The ROC curve visually assesses the trade-off between true positive rate (sensitivity) and false positive rate (1 - specificity) at different probability thresholds.

Prior Probabilities

- The a-priori probabilities represent the probabilities of each class (0 and 1) in the dataset.
    - Class 0: 94.99%
    - Class 1: 5.01%

Conditional Probabilities

Variable		***Class 0***	***Class 1***
Gender	Male	41.54%	43.90%
	Female	58.51%	57.07%
Hypertension		8.65%	24.88%
Hear disease		4.66%	20.49%
Ever married	Yes	64.72%	88.29%
	No	35.33%	12.68%
Work type	Children	14.09%	0.98%
	Government job	12.83%	14.15%
	Never worked	0.52%	0.49%
	Private	56.68%	60.98%
	Self employed	16.02%	25.85%
Residence type	Rural	49.50%	45.37%
	Urban	50.55%	55.61%
Smoking status	Formerly smoked	16.48%	27.81%
	Never smoked	37.32%	37.07%
	Smokes	14.81%	17.56%
	Unknown	31.50%	19.51%

Model Evaluation

The confusion matrix is showing the counts of true positives (1s predicted correctly), true negatives (0s predicted correctly), false positives (0s predicted as 1s), and false negatives (1s predicted as 0s).

- True negatives (0s predicted as 0s) = 861
- False positives (0s predicted as 1s) = 24
- False negatives (1s predicted as 0s) = 117
- True positives (1s predicted as 1s) = 20

This naive model achieved an accuracy of approximately 86.20% on the testing set, indicating the proportion of correctly classified instances and also this model appears to be relatively good at correctly predicting negative instances (class 0), but there is room for improvement in predicting positive instances (class 1), as suggested by the higher number of false negatives.

Precision indicates the model's ability to correctly identify positive instances when it predicts them, and it stands at approximately 45.45%.

Recall reflects the model's ability to capture all actual positive instances, and it is relatively low at about 14.60%.

The F1 score, considering both precision and recall, is approximately 22.10%, providing a balanced assessment of the model's performance.

Results and Conclusion

In the real-world challenge of predicting strokes, this model shows promise by accurately identifying cases where a stroke is unlikely, as seen in the significant number of true negatives. However, when it comes to recognizing potential stroke cases, the model struggles, leading to a considerable number of false negatives. This means it often misses instances where a stroke actually occurs. The model's low precision indicates it sometimes makes mistakes when predicting strokes, and its low recall suggests it fails to capture many real stroke cases. In summary, while the model performs reasonably well in certain situations, its limitations, especially in predicting strokes accurately, highlight the need for improvements to ensure its effectiveness in real-world stroke classification scenarios. Strategies could involve refining input features and exploring alternative algorithms to better capture patterns indicative of stroke risk.

In conclusion, the model's performance in stroke classification, though promising in certain aspects, reveals shortcomings in accurately predicting positive instances of strokes. Enhancing the model's ability to identify potential stroke cases, possibly through feature refinement and algorithm exploration, is crucial for its practical application in real-world healthcare settings. Ongoing monitoring and adjustments based on new insights and data will be essential to ensure the model's reliability and effectiveness in addressing the critical challenge of stroke prediction.

***