Stroke Prediction

Antonio Rodriguez

2023-11-12

Loading Libraries

```
library(dplyr)
library(ggplot2)
library(caret)
library(class)
library(yardstick)
library(glmnet)
library(xgboost)
```

Introduction

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths.

This data set 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 relevant information about the patient.

Features

- gender
- age, in years
- hypertension: indicator of whether someone has hypertension
- heart_disease: indicator of whether someone has heart disease
- ever_married: indicator of whether someone has ever been married
- work_type: type of employment classified into government, private sector, self employed, never worked, or they are a child with no work history
- residence_type: geographic location classified as rural or urban
- avg_glucose_level: average glucose level in mg/dL
- bmi (body mass index)
- smoking_status: smoking habits classified as never smoked, currently smokes, formerly smoked, or unknown
- stroke: indicator of whether someone has had a stroke or not

Loading Dataset

```
stroke_df <- read.csv('healthcare-dataset-stroke-data.csv')
summary(stroke_df)
```

```
##        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    Length:5110        Length:5110        Min.   :0.00000
##  1st Qu.: 77.25    Class :character   Class :character   1st Qu.:0.00000
##  Median : 91.89    Mode  :character   Mode  :character   Median :0.00000
##  Mean   :106.15                                          Mean   :0.04873
##  3rd Qu.:114.09                                          3rd Qu.:0.00000
##  Max.   :271.74                                          Max.   :1.00000
```

```
str(stroke_df)
```

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender           : chr  "Male" "Female" "Male" "Female" ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

Cleaning Data

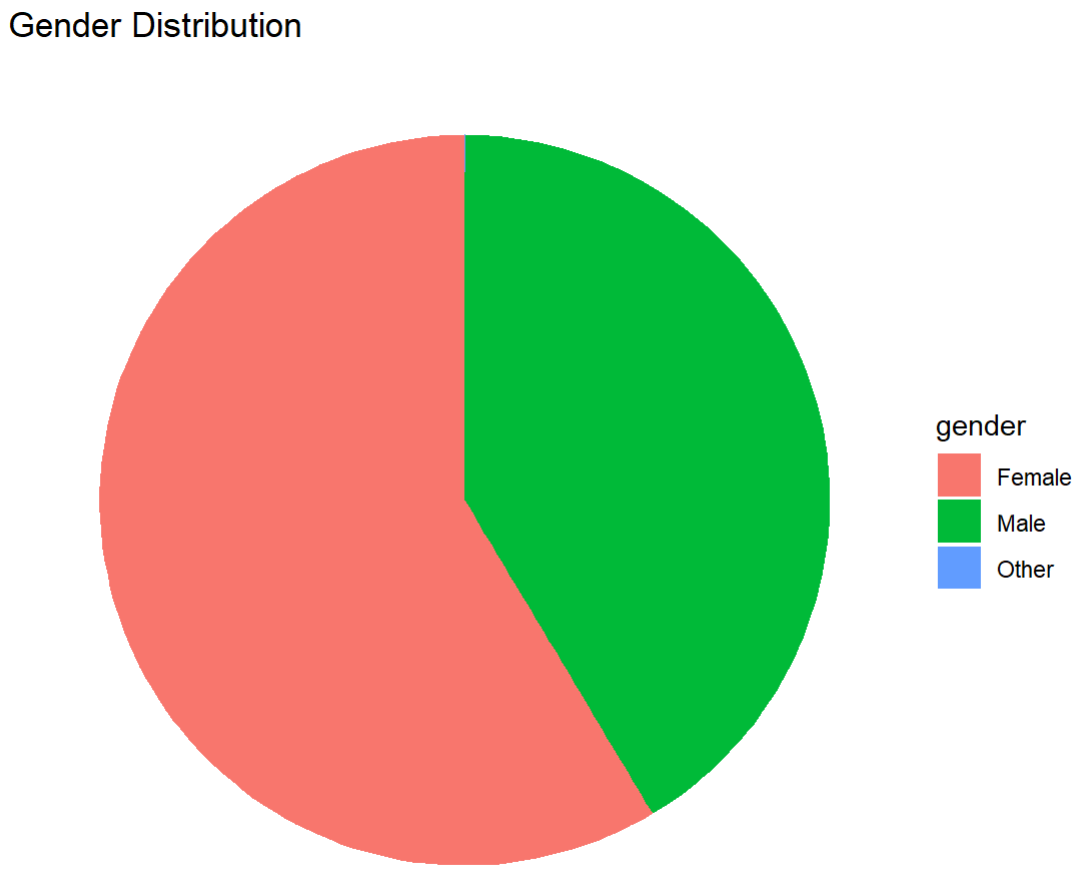
BMI in the summary above is listed a type “chr”. I will be changing it into a numeric value to help the prediction model and for EDA purposes. To deal with the NA values in the BMI column, I will assign the mean BMI to each.

```
stroke_df$bmi <- as.numeric(stroke_df$bmi)
stroke_df$bmi[is.na(stroke_df$bmi)] <- mean(stroke_df$bmi,na.rm=TRUE)
```

EDA

Gender

```
ggplot(data = stroke_df, aes(x = "", fill = gender)) +
  geom_bar(width = 1) +
  coord_polar(theta = "y") +
  theme_void() +
  labs(title = "Gender Distribution")
```




```
gender_counts <- table(stroke_df$gender)
print(gender_counts)
```

```
##
## Female   Male  Other
##   2994   2115      1
```

In the gender column, there were three different values: Male, Female and Other. Females account for most of the data with 2994. Male had a count of 2115. Other only had one occurrence. In a prediction model, there is no reason for this data to be include if it only has one occurrence. There are two ways to deal with it, either assign the person the gender with the highest frequency or just simply delete the row from the dataset. In this scenerio I will simply just delete the row.

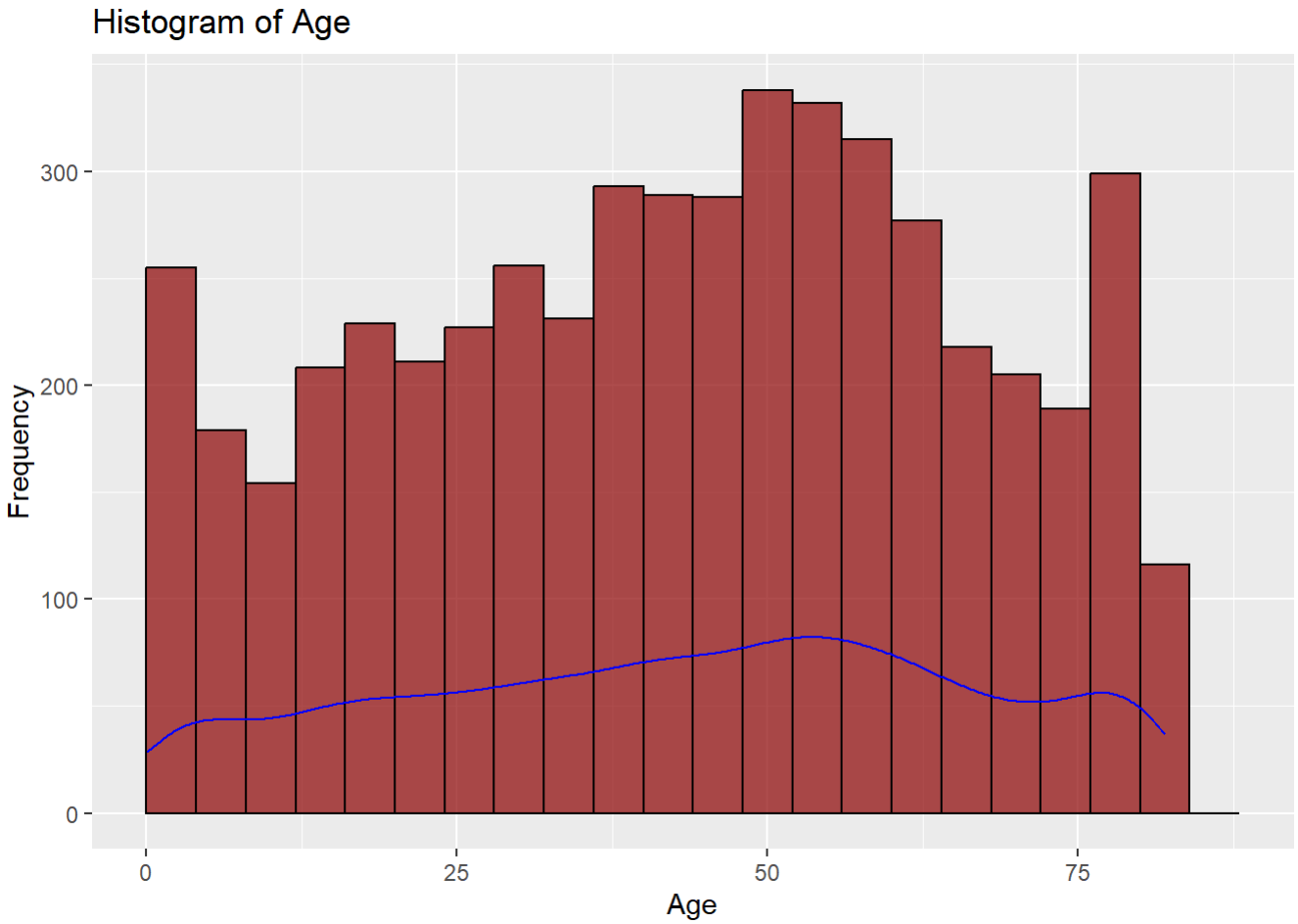
```
stroke_df <- stroke_df %>%
  filter(gender != "Other")
```

```
plot_data <- stroke_df %>%
  group_by(gender, stroke) %>%
  summarise(count = n()) %>%
  mutate(stroke = as.factor(stroke))

ggplot(plot_data, aes(x = gender, y = count, fill = stroke, label = count)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(position = position_dodge(width = 0.9), vjust = -0.5) +
  labs(title = "Distribution of Stroke by Gender") +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal()
```

In this dataset, it is observed that women experience a higher frequency of strokes. However, the proportion of stroke cases relative to the total number of records is greater among males.

Age

```
ggplot(data = stroke_df, aes(x = age)) +
  geom_histogram(breaks = seq(0, 88, by = 4), fill = "darkred", color = "black", alpha = 0.7) +
  geom_density(aes(y = after_stat(count)), color = "blue") +
  labs(title = "Histogram of Age", x = "Age", y = "Frequency")
```

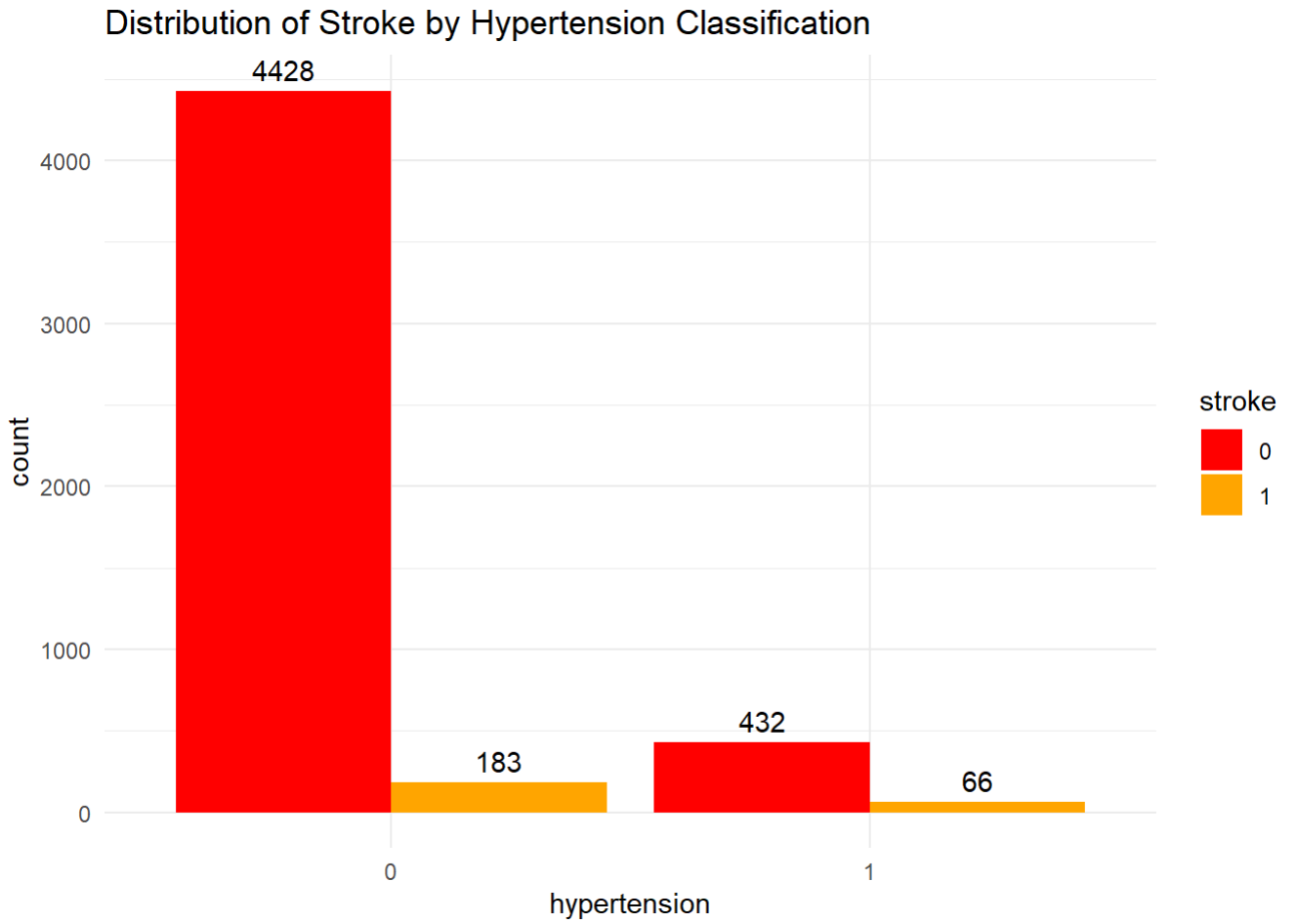



```
stroke_df$stroke <- as.factor(stroke_df$stroke)

ggplot(stroke_df, aes(x=stroke, y=age, fill = stroke)) +
  geom_violin(trim=FALSE, fill="grey", color="darkgreen")+
  geom_boxplot(width=0.1) +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal() +
  labs(title = "Distribution of Stroke Classification by Age")
```

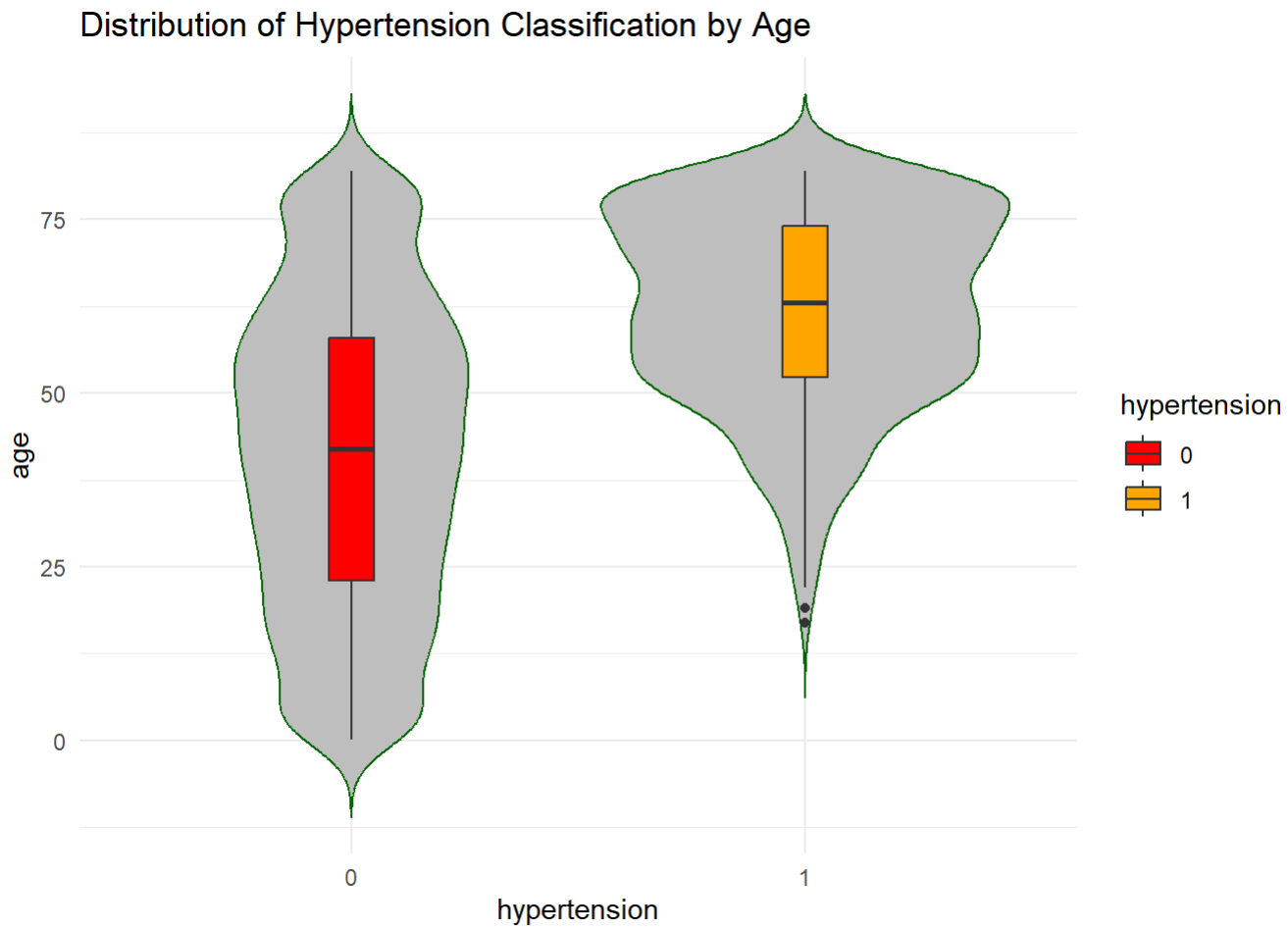
The age distribution in the histogram exhibits a slight left skew, indicating a predominantly balanced distribution. In the violin and box plots, we observe that the majority of stroke victims are aged above 30. However, it's noteworthy that there are two outliers below the age of 30. Additionally, the highest concentration of stroke cases is seen among individuals aged 75 and above.

Hypertension

```
stroke_df$hypertension <- as.factor(stroke_df$hypertension)

plot_data <- stroke_df %>%
  group_by(hypertension, stroke) %>%
  summarise(count = n()) %>%
  mutate(stroke = as.factor(stroke))

ggplot(plot_data, aes(x = hypertension, y = count, fill = stroke, label = count)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(position = position_dodge(width = 0.9), vjust = -0.5) +
  labs(title = "Distribution of Stroke by Hypertension Classification") +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal()
```



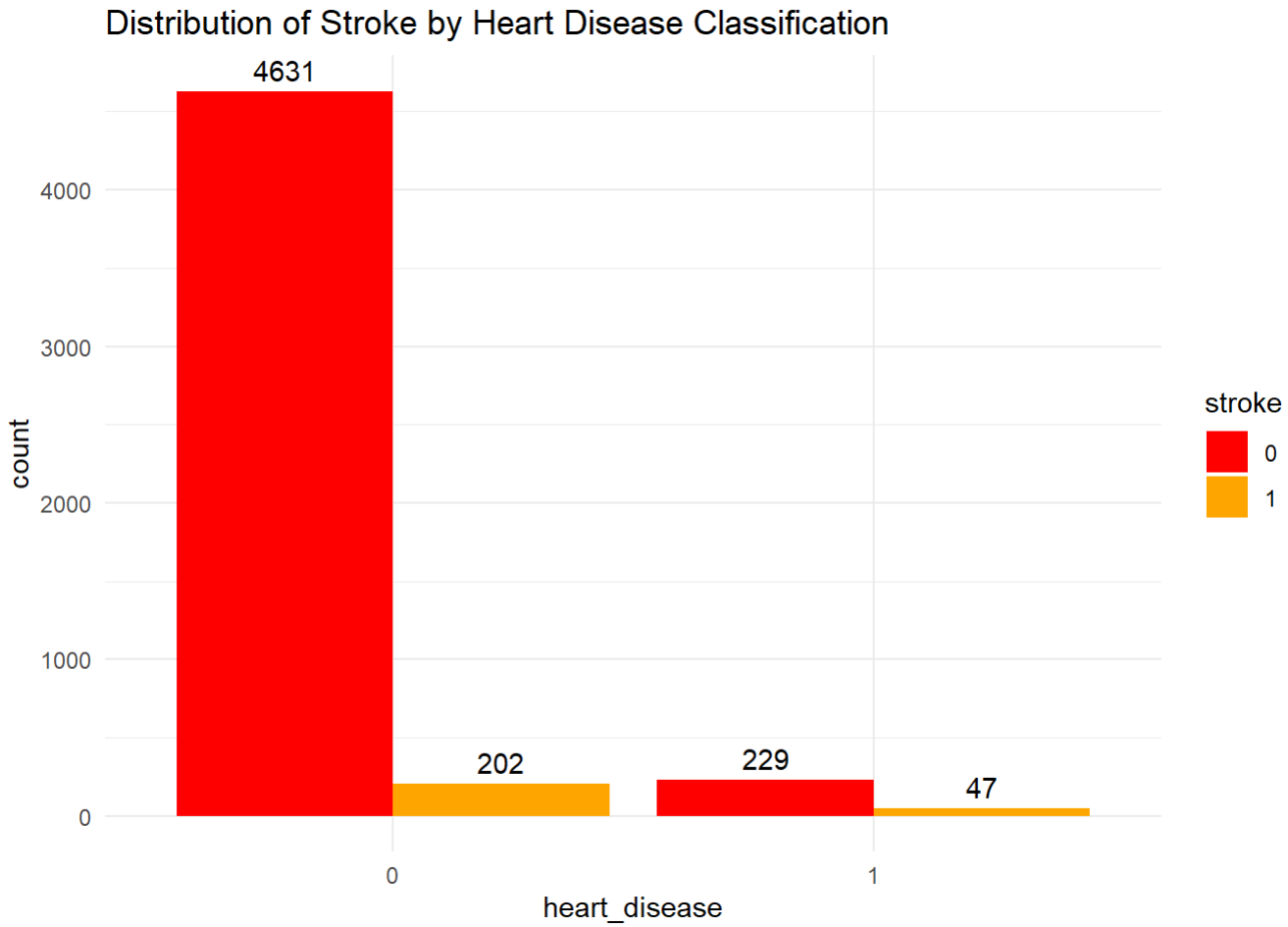

```
ggplot(stroke_df, aes(x=hypertension, y=age, fill = hypertension)) +
  geom_violin(trim=FALSE, fill="grey", color="darkgreen")+
  geom_boxplot(width=0.1) + theme_minimal()+
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  labs(title = "Distribution of Hypertension Classification by Age")
```



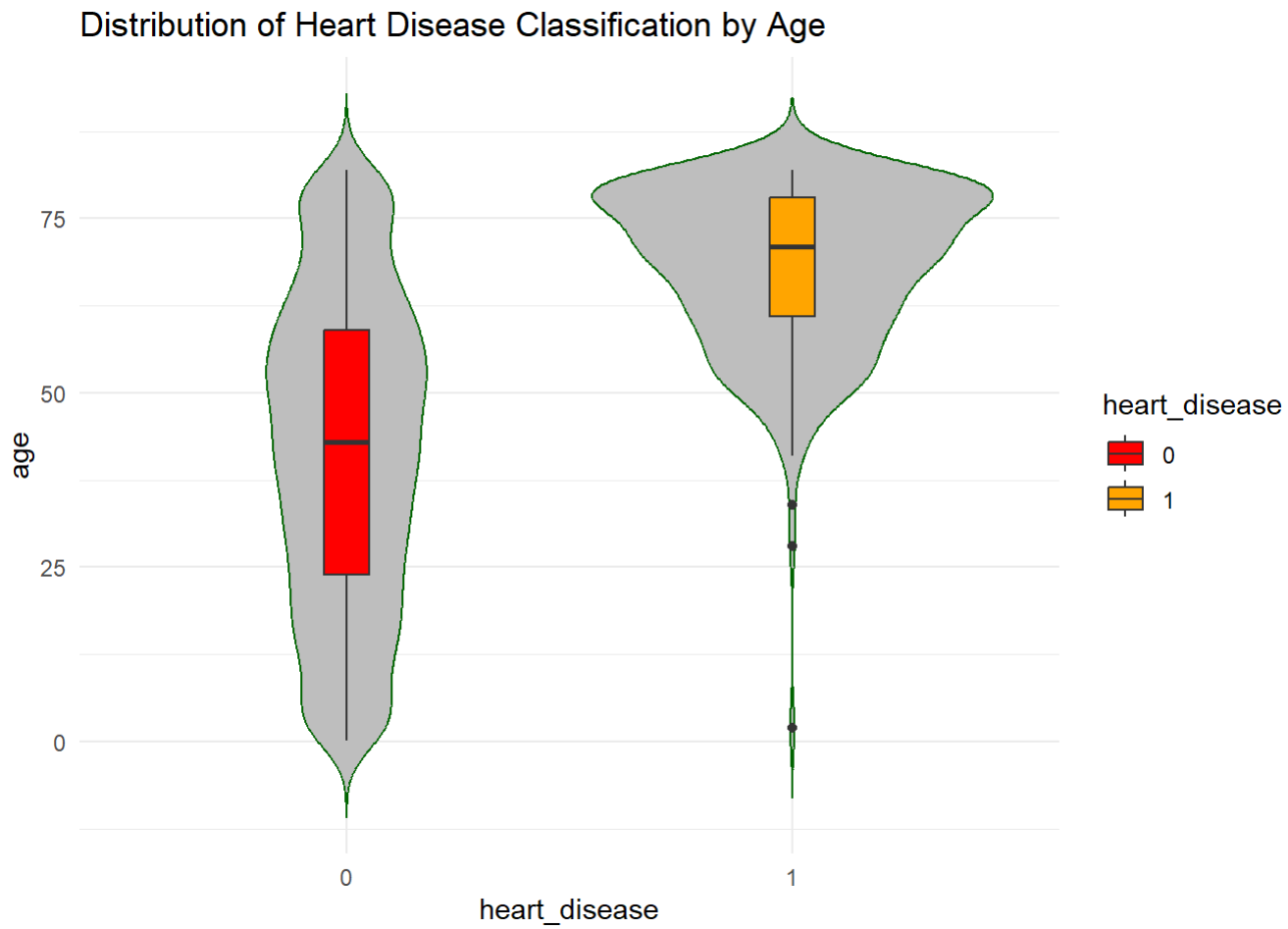
Examining the bar graph, we can see that among all the stroke victims, 73% of them do not have hypertension. Analyzing the violin and box plots, it becomes evident that among those who have hypertension, the majority of them are aged older than 23.

Heart Disease

```
stroke_df$heart_disease <- as.factor(stroke_df$heart_disease)

plot_data <- stroke_df %>%
  group_by(heart_disease, stroke) %>%
  summarise(count = n()) %>%
  mutate(stroke = as.factor(stroke))

ggplot(plot_data, aes(x = heart_disease, y = count, fill = stroke, label = count)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(position = position_dodge(width = 0.9), vjust = -0.5) +
  labs(title = "Distribution of Stroke by Heart Disease Classification") +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal()
```



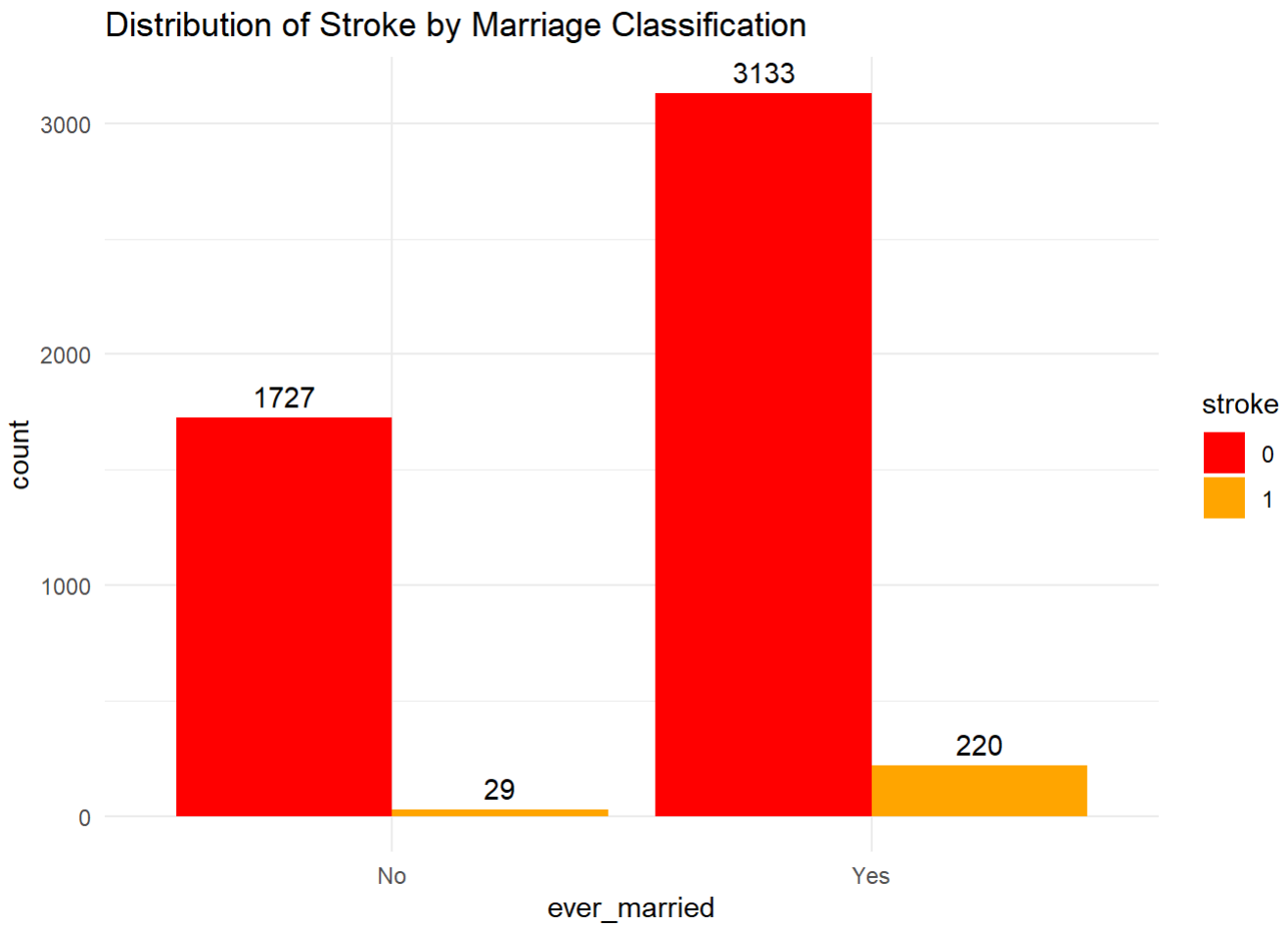

```
ggplot(stroke_df, aes(x=heart_disease, y=age, fill = heart_disease)) +
  geom_violin(trim=FALSE, fill="grey", color="darkgreen")+
  geom_boxplot(width=0.1) +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal()+
  labs(title = "Distribution of Heart Disease Classification by Age")
```



Examining the bar graph, we can see that among all the stroke victims, 81% of them do not have heart disease. Analyzing the violin and box plots, it becomes evident that among those who have heart disease, the majority of them are aged older than 38.

Ever Married

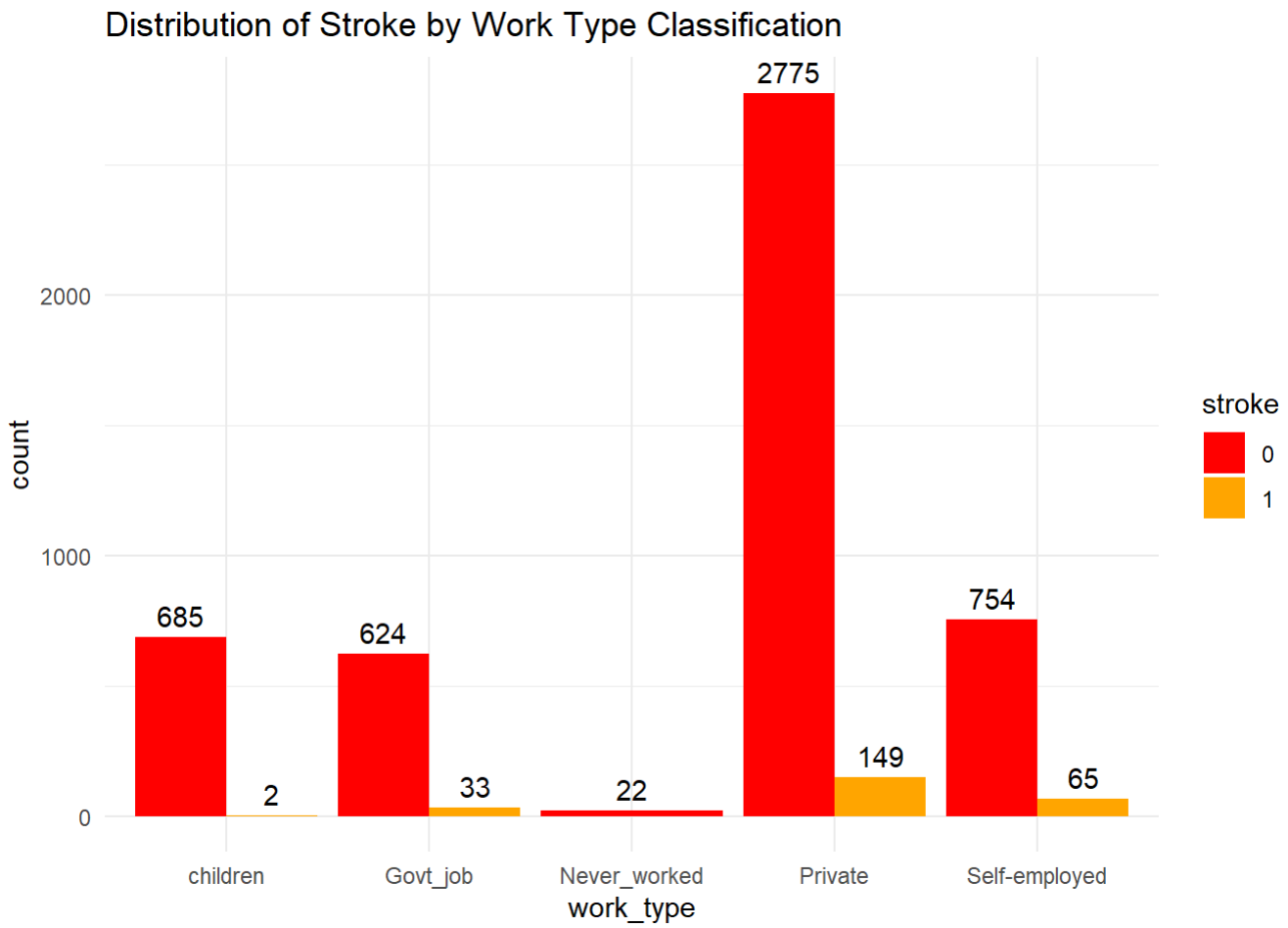
```
stroke_df$ever_married <- as.factor(stroke_df$ever_married)

plot_data <- stroke_df %>%
  group_by(ever_married, stroke) %>%
  summarise(count = n()) %>%
  mutate(stroke = as.factor(stroke))

ggplot(plot_data, aes(x = ever_married, y = count, fill = stroke, label = count)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(position = position_dodge(width = 0.9), vjust = -0.5) +
  labs(title = "Distribution of Stroke by Marriage Classification") +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal()
```



According to the bar chart, 88% of stroke victims in the dataset have a marital status of either ‘married’ or ’previously married.

Work Type

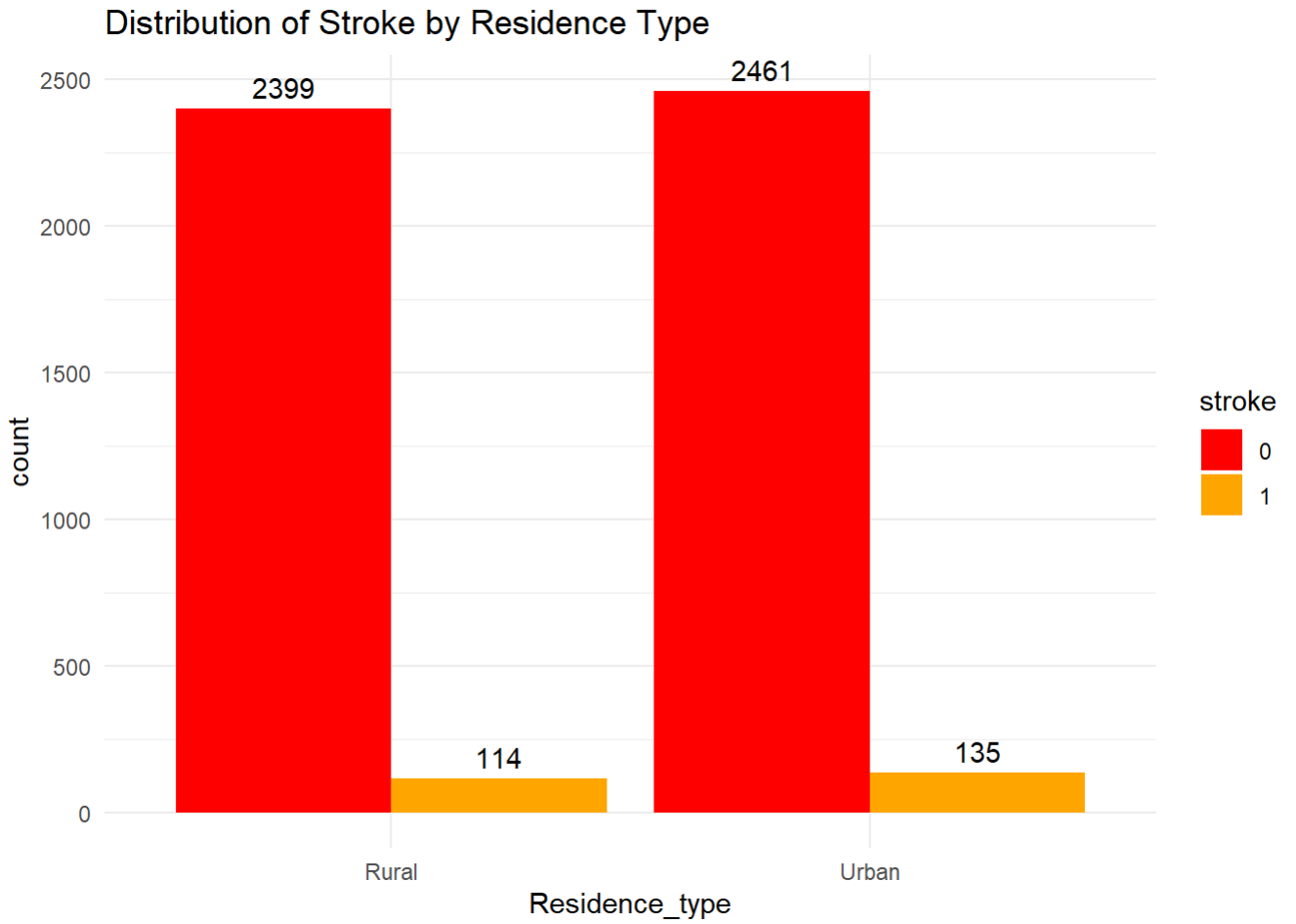
```
stroke_df$work_type <- as.factor(stroke_df$work_type)

plot_data <- stroke_df %>%
  group_by(work_type, stroke) %>%
  summarise(count = n()) %>%
  mutate(stroke = as.factor(stroke))

ggplot(plot_data, aes(x = work_type, y = count, fill = stroke, label = count)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(position = position_dodge(width = 0.9), vjust = -0.5) +
  labs(title = "Distribution of Stroke by Work Type Classification") +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal()
```



Based on the bar chart depicting work types, it is evident that individuals in the private sector constitute the majority of stroke victims in the dataset.

Residence Type

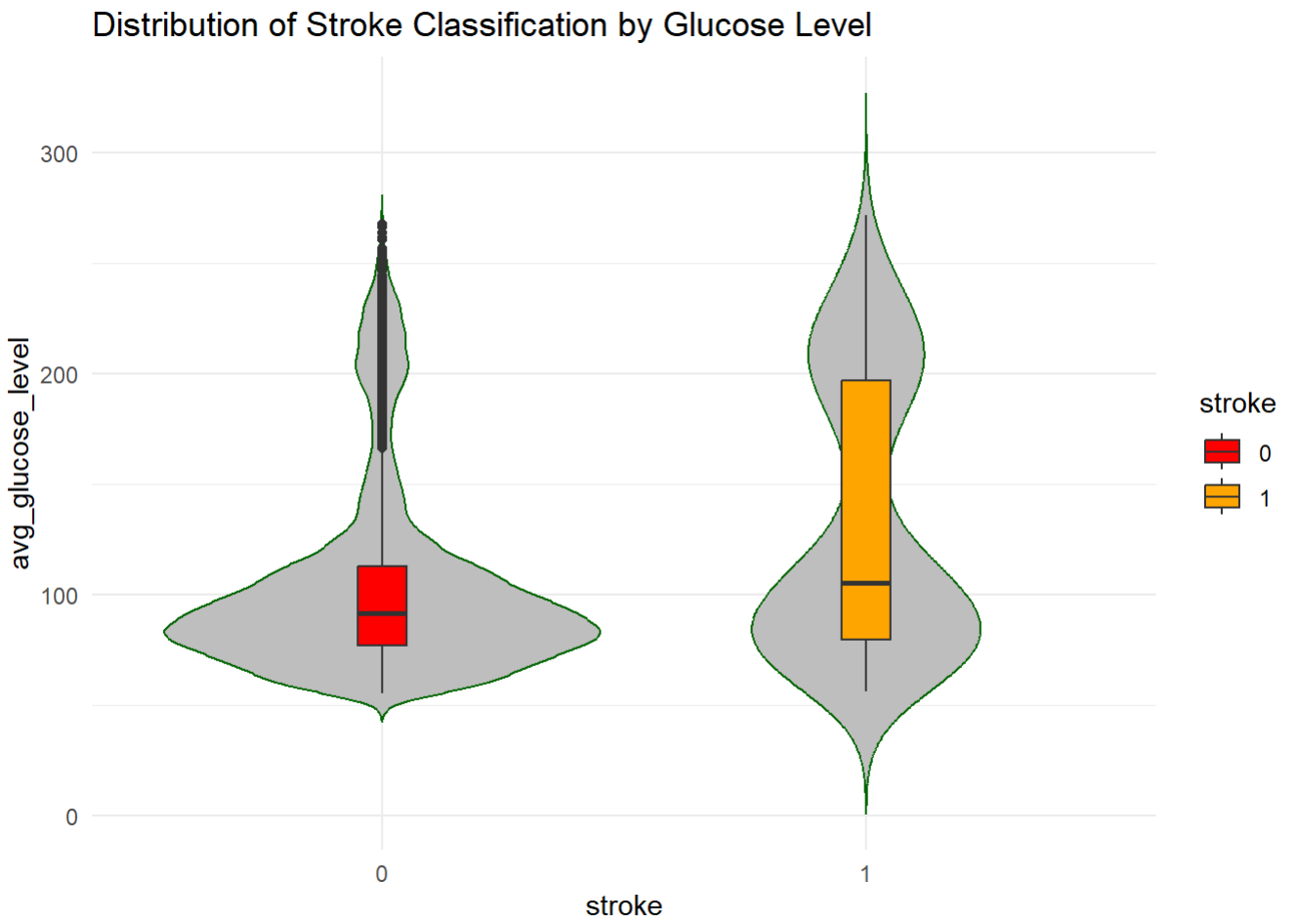
```
stroke_df$Residence_type <- as.factor(stroke_df$Residence_type)

plot_data <- stroke_df %>%
  group_by(Residence_type, stroke) %>%
  summarise(count = n()) %>%
  mutate(stroke = as.factor(stroke))

ggplot(plot_data, aes(x = Residence_type, y = count, fill = stroke, label = count)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(position = position_dodge(width = 0.9), vjust = -0.5) +
  labs(title = "Distribution of Stroke by Residence Type") +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal()
```



The distribution of stroke victims across residence types appears relatively even, with a slightly higher number of cases in urban areas

Average Glucose Level

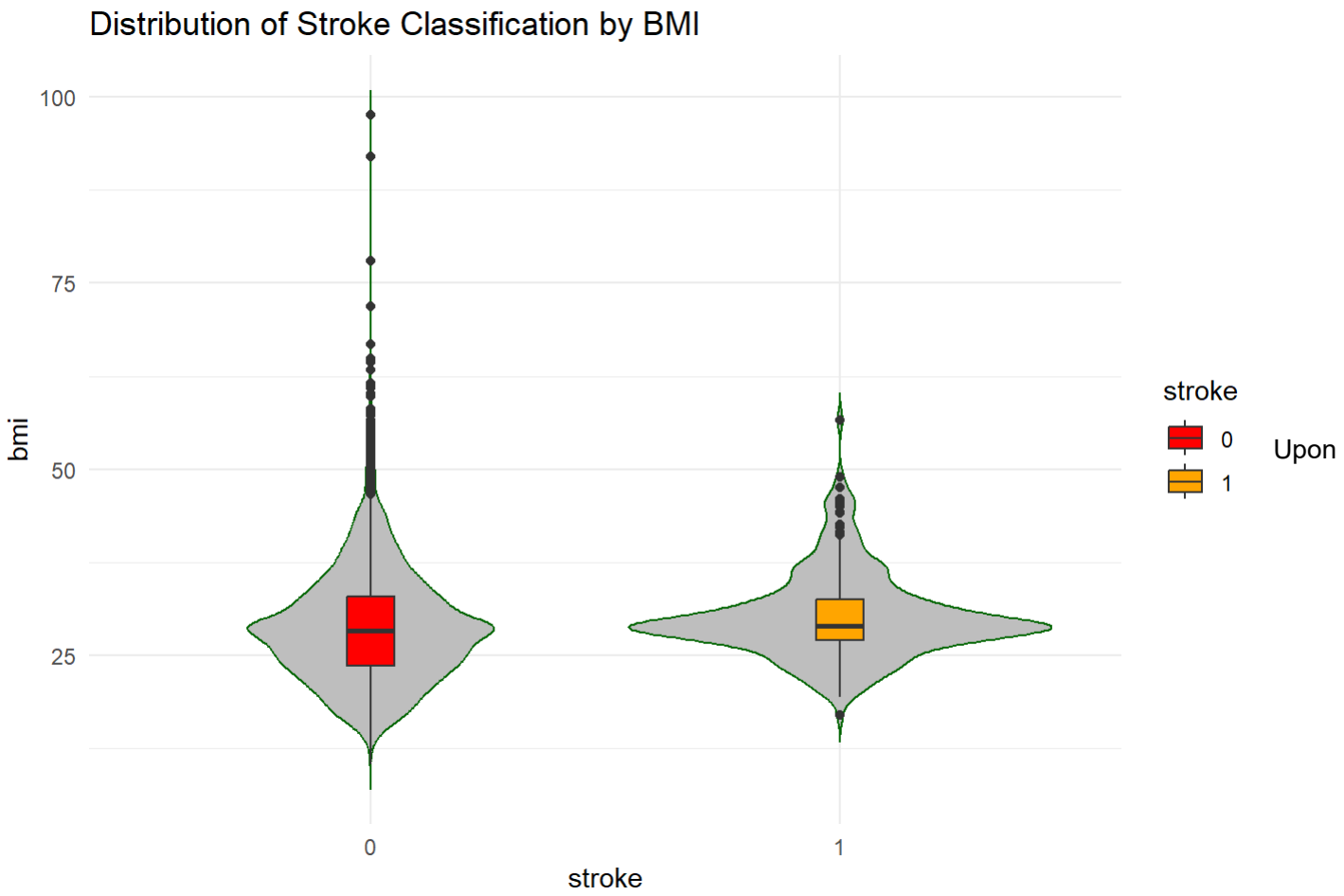
```
ggplot(stroke_df, aes(x=stroke, y=avg_glucose_level, fill = stroke)) +
  geom_violin(trim=FALSE, fill="grey", color="darkgreen")+
  geom_boxplot(width=0.1) +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal()+
  labs(title = "Distribution of Stroke Classification by Glucose Level")
```



When examining the violin and plots, we observe that the medians of both groups are relatively close, indicating a similar central tendency. However, the interquartile range (IQR) for the stroke victims is noticeably wider, and the third quartile (Q3) is skewed toward higher values.

Body Mass Index

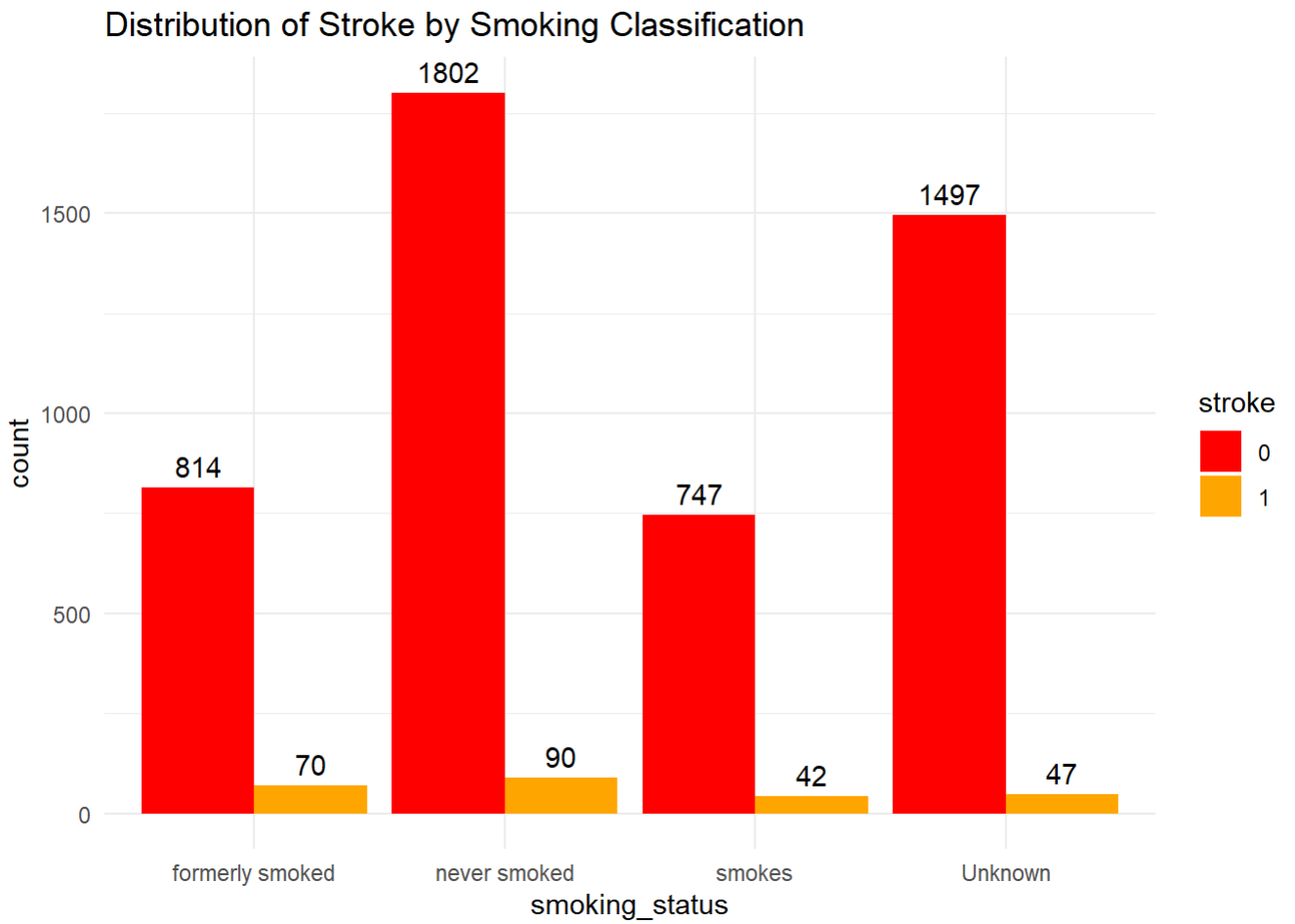
```
ggplot(stroke_df, aes(x=stroke, y=bmi, fill = stroke)) +
  geom_violin(trim=FALSE, fill="grey", color="darkgreen")+
  geom_boxplot(width=0.1) +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal()+
  labs(title = "Distribution of Stroke Classification by BMI")
```



examining the violin and box plots, it is apparent that the medians of both groups are nearly identical, indicating a similar central tendency. The interquartile range (IQR) for the stroke victims is more compact. Notably, the distribution of BMI for stroke victims is concentrated within the 25-32 range.

Smoking Status

```
stroke_df$smoking_status <- as.factor(stroke_df$smoking_status)

plot_data <- stroke_df %>%
  group_by(smoking_status, stroke) %>%
  summarise(count = n()) %>%
  mutate(stroke = as.factor(stroke))

ggplot(plot_data, aes(x = smoking_status, y = count, fill = stroke, label = count)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(position = position_dodge(width = 0.9), vjust = -0.5) +
  labs(title = "Distribution of Stroke by Smoking Classification") +
  scale_fill_manual(values = c("0" = "red", "1" = "orange")) +
  theme_minimal()
```



Analyzing the bar graphs for the four smoking classifications, it's evident that the largest proportion of stroke victims falls into the 'never smoked' category.

Modeling

Logistic Regression

```
model1 <- glm(stroke ~ . - id, family = binomial, data = stroke_df)
summary(model1)
```

```
##
## Call:
## glm(formula = stroke ~ . - id, family = binomial, data = stroke_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1266  -0.3199  -0.1639  -0.0868   3.5608
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -6.830176   0.786172  -8.688  < 2e-16 ***
## genderMale                  0.013534   0.141867   0.095 0.923998
## age                         0.074977   0.005842  12.834  < 2e-16 ***
## hypertension1               0.400601   0.164959   2.428 0.015162 *
## heart_disease1              0.281022   0.191098   1.471 0.141411
## ever_marriedYes            -0.184642   0.225391  -0.819 0.412667
## work_typeGovt_job          -0.978971   0.836441  -1.170 0.241840
## work_typeNever_worked     -10.340873 309.249987  -0.033 0.973325
## work_typePrivate           -0.836153   0.820460  -1.019 0.308143
## work_typeSelf-employed     -1.213027   0.841190  -1.442 0.149292
## Residence_typeUrban         0.083365   0.138323   0.603 0.546719
## avg_glucose_level           0.003976   0.001199   3.317 0.000911 ***
## bmi                         0.003535   0.011288   0.313 0.754173
## smoking_statusnever smoked -0.206054   0.175911  -1.171 0.241457
## smoking_statussmokes        0.113880   0.215347   0.529 0.596929
## smoking_statusUnknown      -0.072099   0.208368  -0.346 0.729331
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1990.3  on 5108  degrees of freedom
## Residual deviance: 1581.1  on 5093  degrees of freedom
## AIC: 1613.1
##
## Number of Fisher Scoring iterations: 14
```

In the analysis of the GLM model summary, it is evident that the variables age, hypertension, and average glucose level exhibit statistical significance in predicting the occurrence of a stroke, with p-values below the 0.05 threshold. Armed with this insight, I will proceed to construct a refined model that exclusively incorporates these significant features.

```
model2 <- glm(stroke ~ age + hypertension + avg_glucose_level, family = binomial, data = stroke_
df)
summary(model2)
```

```
##
## Call:
## glm(formula = stroke ~ age + hypertension + avg_glucose_level,
##     family = binomial, data = stroke_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0062  -0.3242  -0.1734  -0.0815   3.7894
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.578436   0.355595 -21.312  < 2e-16 ***
## age                0.070579   0.005062  13.942  < 2e-16 ***
## hypertension1      0.384447   0.162330   2.368 0.017870 *
## avg_glucose_level  0.004354   0.001152   3.779 0.000158 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1990.3  on 5108  degrees of freedom
## Residual deviance: 1594.4  on 5105  degrees of freedom
## AIC: 1602.4
##
## Number of Fisher Scoring iterations: 7
```

In our model, we find that as age increases by 1 unit, the likelihood of experiencing a stroke increases by 0.070579. Similarly, a 1-unit increase in hypertension is associated with a 0.384447 increase in the likelihood of having a stroke. Lastly, a 1-unit rise in average glucose level is linked to a 0.004354 increase in the likelihood of suffering a stroke.

Logistical Regression model

```
set.seed(123)

split <- createDataPartition(stroke_df$stroke, p = 0.8, list = FALSE)
data_stroke_train <- stroke_df[split,]
data_stroke_test <- stroke_df[-split,]


predictors <- stroke_df[, c("age", "hypertension", "avg_glucose_level")]
response <- stroke_df$stroke

ctrl <- trainControl(method = "cv", number = 10)

model <- train(
  stroke ~ age + hypertension + avg_glucose_level, data = data_stroke_train, method = "glm",
  trControl = ctrl, family = binomial
)

predictions <- predict(model, newdata = data_stroke_test, type = "raw")

conf_matrix <- confusionMatrix(predictions, data_stroke_test$stroke)
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 972  49
##          1   0   0
##
##                Accuracy : 0.952
##                  95% CI : (0.937, 0.9643)
##     No Information Rate : 0.952
##     P-Value [Acc > NIR] : 0.5379
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 7.025e-12
##
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 0.952
##          Neg Pred Value :   NaN
##              Prevalence : 0.952
##          Detection Rate : 0.952
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : 0
##
```

```
precision <- conf_matrix$byClass["Pos Pred Value"]
recall <- conf_matrix$byClass["Sensitivity"]
logReg_f1 <- 2 * (precision * recall) / (precision + recall)

cat("F1-Score:", logReg_f1, "\n")
```

```
## F1-Score: 0.9754139
```

k-NN

```
knn_df <- stroke_df[, c("age", "hypertension", "avg_glucose_level", "stroke")]
knn_split <- createDataPartition(knn_df$stroke, p = 0.8, list = FALSE)
knn_train <- knn_df[split,]
knn_test <- knn_df[-split,]

k_values <- c(1, 3, 5, 7, 9)
f1_scores <- numeric(length(k_values))
best_conf_matrix <- NULL
knn_f1 <- 0

for (i in 1:length(k_values)) {
  k <- k_values[i]
  classifier_knn <- knn(train = knn_train,
                        test = knn_test,
                        cl = knn_train$stroke,
                        k = k)

  conf_matrix <- confusionMatrix(data = classifier_knn, reference = knn_test$stroke)
  precision <- conf_matrix$byClass["Pos Pred Value"]
  recall <- conf_matrix$byClass["Sensitivity"]
  f1_scores[i] <- 2 * (precision * recall) / (precision + recall)

  if (f1_scores[i] > knn_f1){
    best_conf_matrix <- conf_matrix
    knn_f1 <- f1_scores[i]
  }

}

best_k <- k_values[which.max(f1_scores)]
cat("Best k:", best_k, " with F1-Score of ", knn_f1, "\n")
```

```
## Best k: 7  with F1-Score of  0.9763938
```

```
best_conf_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 972  47
##          1   0   2
##
##                Accuracy : 0.954
##                  95% CI : (0.9393, 0.966)
##     No Information Rate : 0.952
##     P-Value [Acc > NIR] : 0.4213
##
##                   Kappa : 0.0749
##
##  Mcnemar's Test P-Value : 1.949e-11
##
##             Sensitivity : 1.00000
##             Specificity : 0.04082
##          Pos Pred Value : 0.95388
##          Neg Pred Value : 1.00000
##              Prevalence : 0.95201
##          Detection Rate : 0.95201
##    Detection Prevalence : 0.99804
##       Balanced Accuracy : 0.52041
##
##        'Positive' Class : 0
##
```

XGBoost

```
XGB_df <- knn_df
XGB_df$age <- as.numeric(XGB_df$age)
XGB_df$hypertension <- as.numeric(XGB_df$hypertension)
XGB_df$avg_glucose_level <- as.numeric(XGB_df$avg_glucose_level)
XGB_df$stroke <- as.numeric(XGB_df$stroke)

XGB_split <- createDataPartition(XGB_df$stroke, p = 0.8, list = FALSE)
XGB_train <- XGB_df[split,]
XGB_test <- XGB_df[-split,]

X <- XGB_train[, c("age", "hypertension", "avg_glucose_level")]
Y <- XGB_train$stroke

dtrain <- xgb.DMatrix(data = as.matrix(X), label = as.numeric(Y) - 1)

params <- list(
  booster = "gbtree",
  objective = "binary:logistic",
  eval_metric = "logloss",
  eta = 0.3,
  max_depth = 6,
  subsample = 0.8,
  colsample_bytree = 0.8,
  nrounds = 100
)

xgb_model <- xgb.train(params = params, data = dtrain, nrounds = params$nrounds)
```

```
## [17:50:57] WARNING: src/learner.cc:767:
## Parameters: { "nrounds" } are not used.
```

```
X_test <- as.matrix(XGB_test[, c("age", "hypertension", "avg_glucose_level")])
dtest <- xgb.DMatrix(data = X_test)
predictions <- predict(xgb_model, dtest)

predicted_labels <- ifelse(predictions > 0.5, 1, 0)

ref <- as.numeric(XGB_test$stroke) - 1
ref <- factor(ref, levels = c(0, 1))
predicted_labels <- factor(predicted_labels, levels = c(0, 1))

conf_matrix <- confusionMatrix(data = predicted_labels, reference = ref)
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 965  48
##          1   7   1
##
##                Accuracy : 0.9461
##                  95% CI : (0.9305, 0.9592)
##     No Information Rate : 0.952
##     P-Value [Acc > NIR] : 0.83
##
##                   Kappa : 0.0219
##
##  Mcnemar's Test P-Value : 6.906e-08
##
##             Sensitivity : 0.99280
##             Specificity : 0.02041
##          Pos Pred Value : 0.95262
##          Neg Pred Value : 0.12500
##              Prevalence : 0.95201
##          Detection Rate : 0.94515
##    Detection Prevalence : 0.99216
##       Balanced Accuracy : 0.50660
##
##        'Positive' Class : 0
##
```

```
precision <- conf_matrix$byClass["Pos Pred Value"]
recall <- conf_matrix$byClass["Sensitivity"]
xgb_F1 <- 2 * (precision * recall) / (precision + recall)

cat("F1-Score:", xgb_F1, "\n")
```

```
## F1-Score: 0.9722922
```

Conclusion

```
scores <- data.frame(
  Models = c("Generalized Linear Model", "k-NN", "XGBoost"),
  "F1 Scores" = c(logReg_f1, knn_f1, xgb_F1)
)
print(scores)
```

```
##                     Models F1.Scores
## 1 Generalized Linear Model 0.9754139
## 2                    k-NN 0.9763938
## 3                 XGBoost 0.9722922
```

Upon comparing the performance of the three models with a controlled set of features, the k-Nearest Neighbors model achieved the highest F1 score with 0.9763938 and exhibited an accuracy of 95.4%.