# Build and deploy a stroke prediction model using R

Ronak Fathi

2024-09-02

## About Data Analysis Report

This RMarkdown file contains the report of the data analysis done for the project on building and deploying a stroke prediction model in R. It contains analysis such as data exploration, summary statistics and building the prediction models. The final report was completed on Mon Sep 2 20:55:05 2024.

**Data Description:**

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.

## Import data and data preprocessing

**Load data and install packages**

```
#install.packages(" ")
library(data.table)
library(visdat)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::between()     masks data.table::between()
## x dplyr::filter()      masks stats::filter()
## x dplyr::first()       masks data.table::first()
## x lubridate::hour()    masks data.table::hour()
## x lubridate::isoweek() masks data.table::isoweek()
## x dplyr::lag()         masks stats::lag()
## x dplyr::last()        masks data.table::last()
## x lubridate::mday()    masks data.table::mday()
## x lubridate::minute()  masks data.table::minute()
## x lubridate::month()   masks data.table::month()
## x lubridate::quarter() masks data.table::quarter()
## x lubridate::second()  masks data.table::second()
```

```
## x purrr::transpose()   masks data.table::transpose()
## x lubridate::wday()    masks data.table::wday()
## x lubridate::week()    masks data.table::week()
## x lubridate::yday()    masks data.table::yday()
## x lubridate::year()    masks data.table::year()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(moments)
library(dplyr)
library(ggcorrplot)
setwd("C:/Users/0&1/OneDrive/Documents/Stroke")
data <- read.csv("healthcare-dataset-stroke-data.csv")
```

### Exploratory Data Analysis

```
# about the dataset
dim(data) # dimension
```

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

```
head(data) # content
```

```
##      id gender age hypertension heart_disease ever_married     work_type
## 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
##   Residence_type avg_glucose_level  bmi  smoking_status stroke
## 1          Urban            228.69 36.6 formerly smoked      1
## 2          Rural            202.21  N/A    never smoked      1
## 3          Rural            105.92 32.5    never smoked      1
## 4          Urban            171.23 34.4          smokes      1
## 5          Rural            174.12   24    never smoked      1
## 6          Urban            186.21   29 formerly smoked      1
```

```
str(data) # structure
```

```
## '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 ...
```

```
summary(data) # summary
```

```
##        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
```

```
#about variables
## check unique values of categorical values
table(data$gender) # found "other"
```

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

```
table(data$ever_married)
```

```
##
##   No  Yes
## 1757 3353
```

```
table(data$work_type)
```

```
##
##      children      Govt_job  Never_worked       Private Self-employed
##           687           657            22          2925           819
```

```
table(data$smoking_status)
```

```
##
## formerly smoked    never smoked          smokes         Unknown
##             885            1892             789            1544
```

```
table(data$Residence_type)
```

```
##
## Rural Urban
##  2514  2596
```

```
# Check for missing values
library(naniar)

miss_scan_count(data = data, search = list("N/A","Unknown","Other"))
```

```
## # A tibble: 12 x 2
##    Variable              n
```

```
##    <chr>             <int>
##  1 id                    0
##  2 gender                1
##  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     1544
## 12 stroke                0
```

```
# 2.1 Imputation BMI
data$bmi <- as.numeric(data$bmi)
```

**For BMI, we are going to use the median to fill, the missing values.**

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

```
idx <- complete.cases(data)

bmi_idx <- is.na(data$bmi)

median_bmi <- median(data$bmi, na.rm = TRUE)
median_bmi
```

```
## [1] 28.1
```

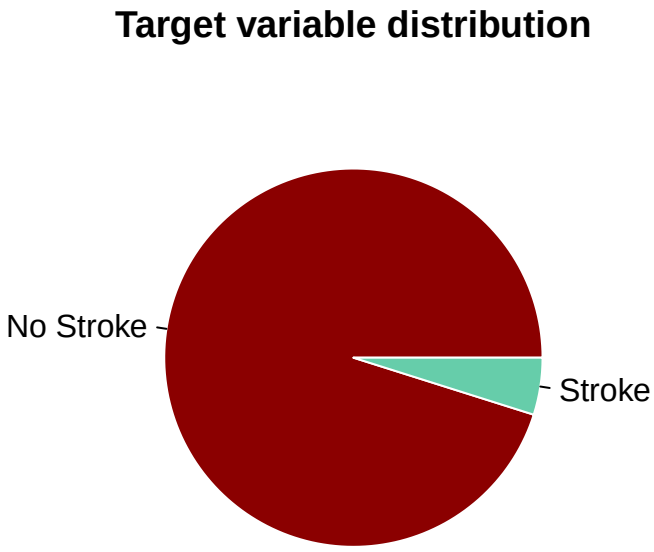
```
data[bmi_idx,]$bmi <- median_bmi
str(data)
```

```
## '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              : num  36.6 28.1 32.5 34.4 24 29 27.4 22.8 28.1 24.2 ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

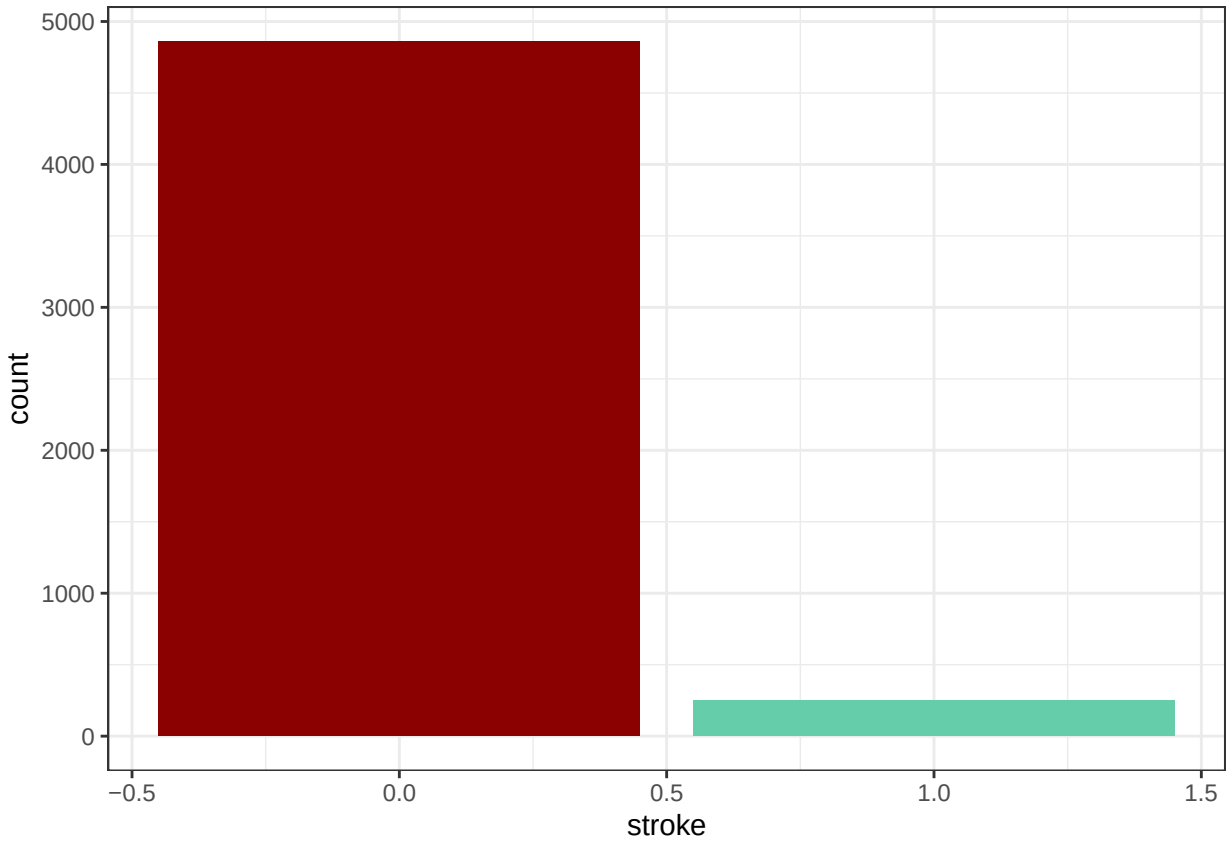
```
#stroke distribution
stroke_counts <- table(data$stroke)

#pie chart
pie(stroke_counts,values = "%", labels = c("No Stroke","Stroke"),border="white", col =c("darkred","aquar
```

```
## Warning in text.default(1.1 * P$x, 1.1 * P$y, labels[i], xpd = TRUE, adj =
## ifelse(P$x < : "values" is not a graphical parameter
## Warning in text.default(1.1 * P$x, 1.1 * P$y, labels[i], xpd = TRUE, adj =
```

```
## ifelse(P$x < : "values" is not a graphical parameter
## Warning in title(main = main, ...): "values" is not a graphical parameter
```



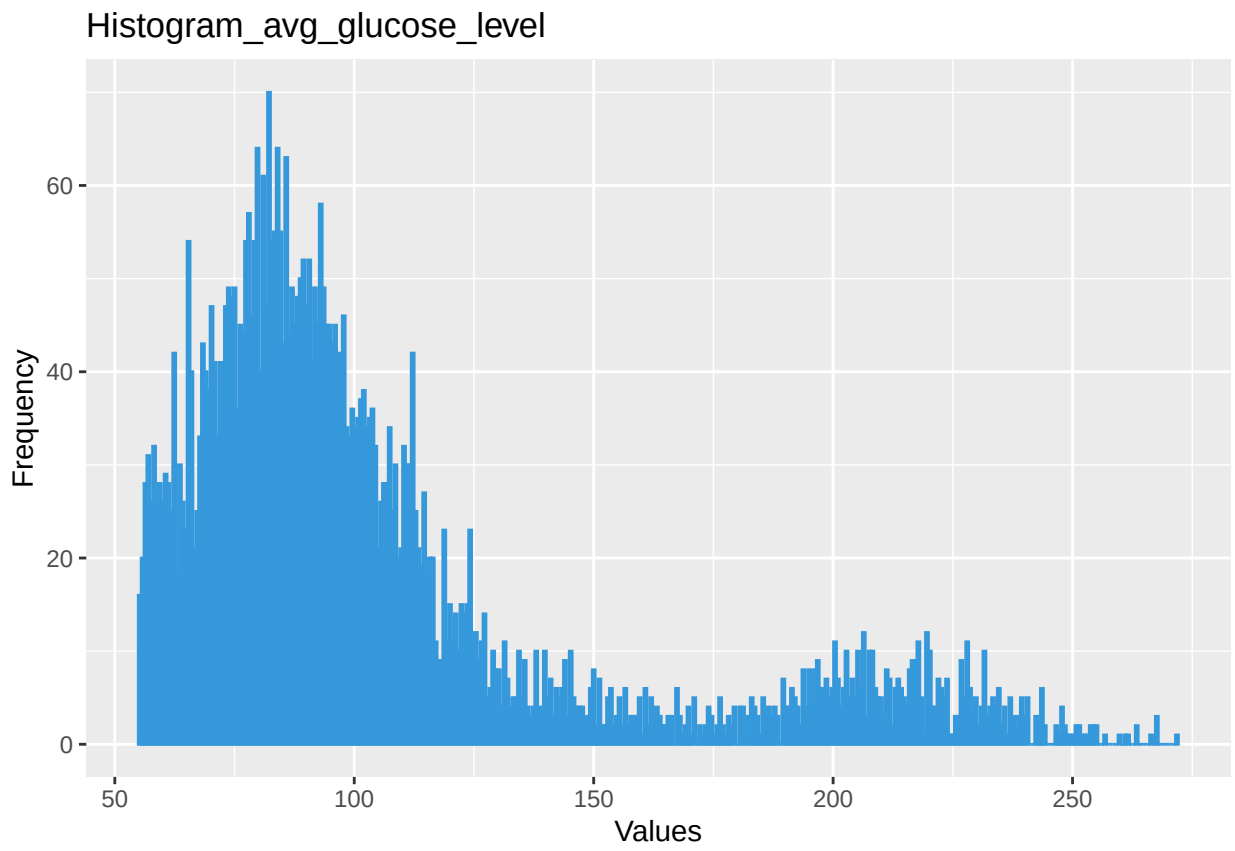

```r
# Univariate Data Analysis
ggplot(data, aes(stroke,))+
  geom_bar(fill=c("darkred","aquamarine3")) +
  theme_bw()+
  theme(plot.title = element_text(hjust = 0.5))+
  xlab("stroke")
```



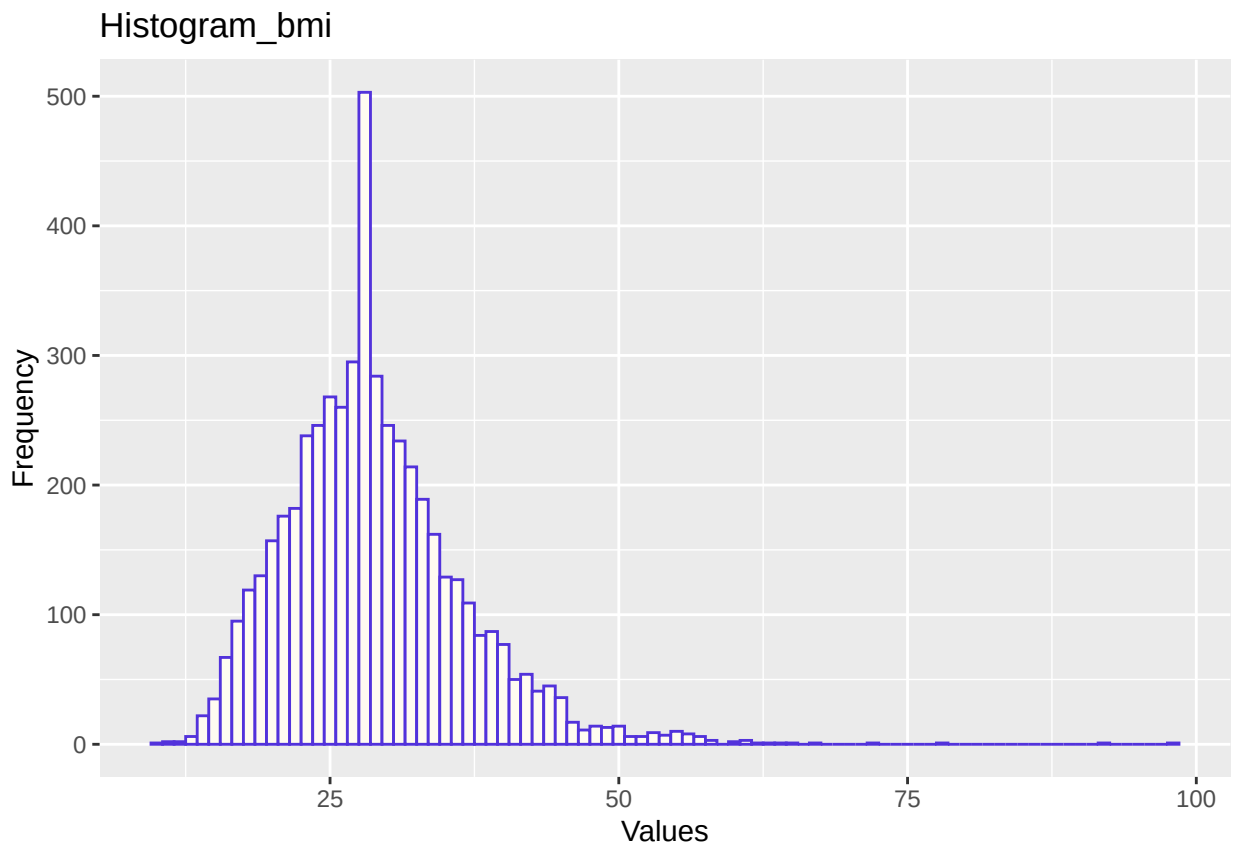

```
#dealing with the type of value
# Converting col_name columns to characters
data <- data %>%
  mutate_at(vars(gender, ever_married, work_type, Residence_type, smoking_status), as.character)

# converting col_name columns to numeric
data <- data %>%
  mutate_at(vars(age, hypertension, heart_disease, avg_glucose_level, bmi, stroke), as.numeric)
```

### Continuous variables

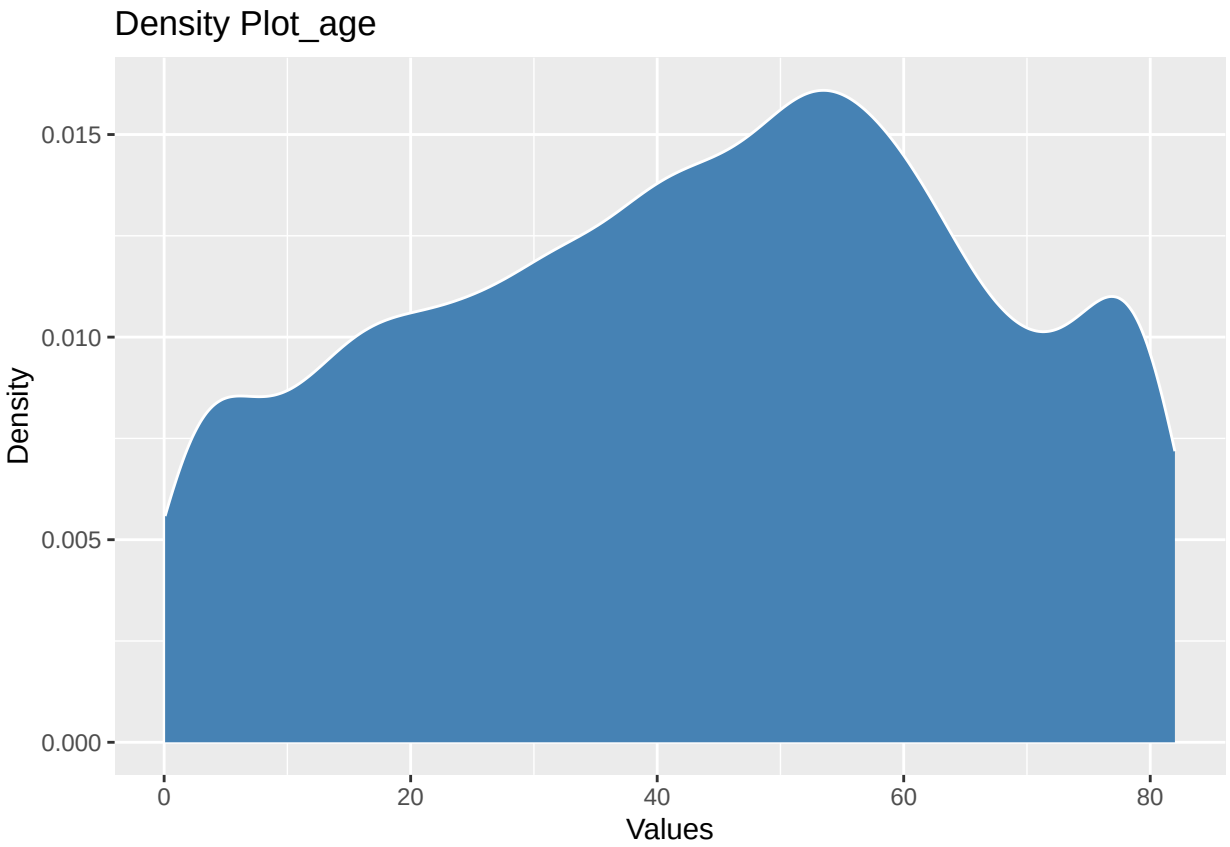
```
# Select the columns to be processed

column <- data$avg_glucose_level
# Histograms specify the dataset and the columns to be used
histogram <- ggplot(data, aes(x = avg_glucose_level))
# Add a histogram layer, set bar widths and colours
histogram <- histogram + geom_histogram(binwidth = 0.6, fill = "#fffbf1", color = "#3498db")
# Add title and axis labels
histogram <- histogram + labs(title = "Histogram_avg_glucose_level", x = "Values", y = "Frequency")

print(histogram)
```




```
column <- data$bmi
# Histograms specify the dataset and the columns to be used
histogram1 <- ggplot(data, aes(x = bmi))
# Add a histogram layer, set bar widths and colours
histogram1 <- histogram1 + geom_histogram(binwidth = 1, fill = "#fffbf1", color = "#5232db")
# Add title and axis labels
histogram1 <- histogram1 + labs(title = "Histogram_bmi", x = "Values", y = "Frequency")

print(histogram1)
```




```r
# Select the columns to be processed
column <- data$agel
# Density map specifies the dataset and the columns to be used
density_plot <- ggplot(data, aes(x = age))
# Add a density map layer, set the fill colour and border colour
density_plot <- density_plot + geom_density(fill = "steelblue", color = "white")
# Add title and axis labels
density_plot <- density_plot + labs(title = "Density Plot_age", x = "Values", y = "Density")

print(density_plot)
```




**Observations:**

- Most patients in the dataset are adults with no outliers.
- There are more adults within the dataset.
- The average glucose level in the data is right skewed.
- Most patients in the dataset have a normal average glucose level.
- Most patients aren't healthy in terms of BMI. There are more obesed and overweight patients than the ones with normal weight.
- There are outliers in the BMI column as well.
- We have more female in the dataset. Also there's a single patient whose gender is "Other".
- Since female is the mode of the gender feature, the patient with 'Other' will be re-categorised to female. This way, we'll have just 2 categories in the column.
- Most of the patients in the data are healthy in terms of heart disease.
- More than 50% of the patients work in the private sector.
- With the assumption that children can't work/never worked, we can move the instances of "children" category to the "Never_worked" category.
- 90% of the patients are not hypertensive.
- We have more patients who have married at one stage in their life than those who haven't.
- We have almost equal amount of patients living in the Rural and Urban areas.

**Feature engineering**

- Analysing the affect of age group, hypertension, heart diseases, living conditions, group of bmi, group of glucose level and smoking status on risk of stroke.
- Classifying body mass.

- Classification of glucose level.

```
data_imp <- data %>%
  mutate(bmi = case_when(bmi < 18.5 ~ "underweight",
                         bmi >= 18.5 & bmi < 25 ~ "normal weight",
                         bmi >= 25 & bmi < 30 ~ "overweight",
                         bmi >= 30 ~ "obese"),
         bmi = factor(bmi, levels = c("underweight",
                                      "normal weight",
                                      "overweight",
                                      "obese"), order = TRUE)) %>%
    mutate(age = case_when(age < 2 ~ "baby",
                           age >= 2 & age < 17 ~ "child",
                           age >= 17 & age < 30 ~ "young adults",
                           age >= 30 & age < 55~ "middle-aged adults",
                           age >= 55 ~ "old-aged adults"),
           age = factor(age, levels = c("baby",
                                        "child",
                                        "young adults",
                                        "middle-aged adults",
                                        "old-aged adults"), order = TRUE)) %>%
  mutate(avg_glucose_level = case_when(avg_glucose_level < 100 ~ "normal",
                                       avg_glucose_level >= 100 & avg_glucose_level < 125 ~ "prediabetes
                                       avg_glucose_level >= 125  ~ "diabetes"),
         avg_glucose_level = factor(avg_glucose_level, levels = c("normal",
                                                                  "prediabetes",
                                                                  "diabetes"), order = TRUE))

table(data_imp$bmi)
```

```
##
##   underweight normal weight    overweight         obese
##           337          1243          1610          1920
```

```
table(data_imp$age)
```

```
##
##               baby              child       young adults middle-aged adults
##                120                676                719               1816
##    old-aged adults
##               1779
```

```
table(data_imp$avg_glucose_level)
```

```
##
##      normal prediabetes    diabetes
##        3131         979        1000
```

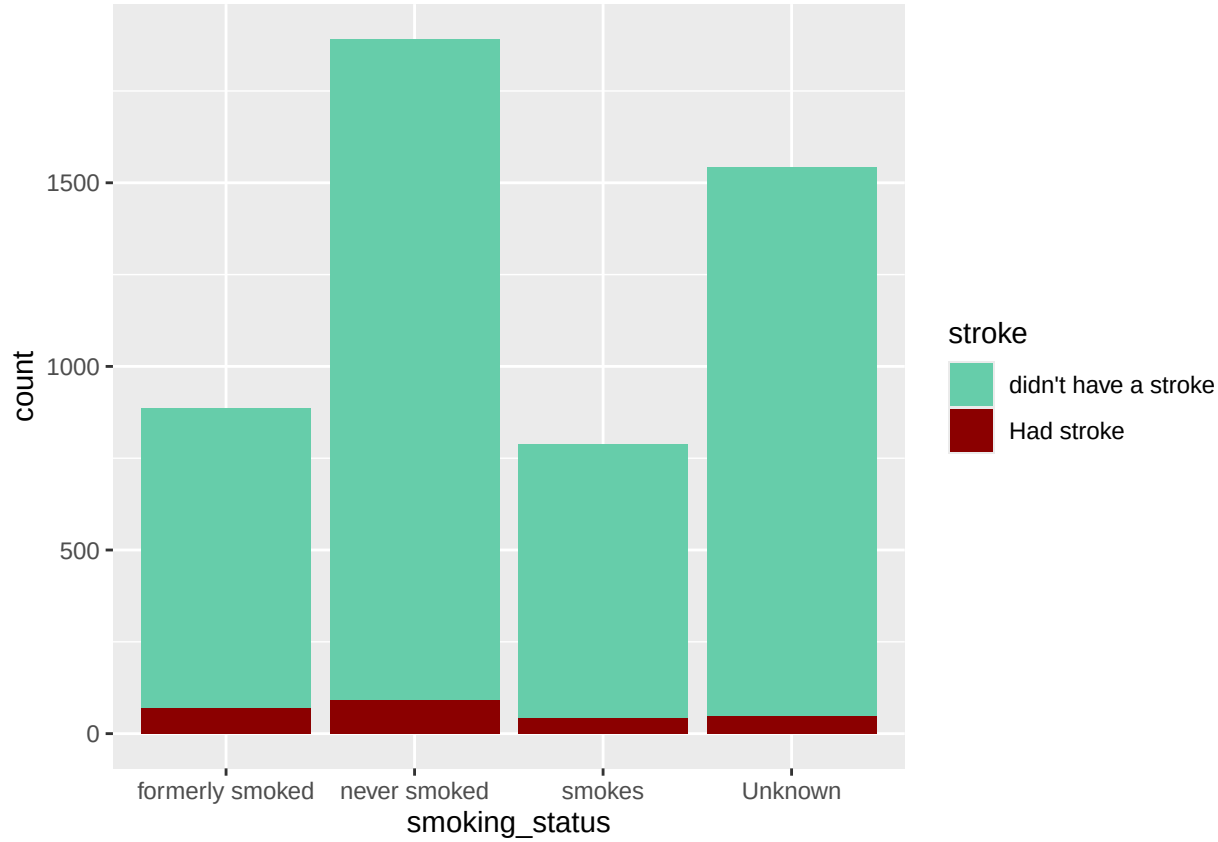
```
# convert data to factor
data_imp$heart_disease <- factor(data_imp$heart_disease)
data_imp$hypertension <- factor(data_imp$hypertension)
data_imp$work_type <- factor(data_imp$work_type)


data_imp$stroke <- factor(data_imp$stroke,
                          levels = c(0,1),
                          labels = c("didn\'t have a stroke","Had stroke"))
```

```r
table(data_imp$stroke)
```

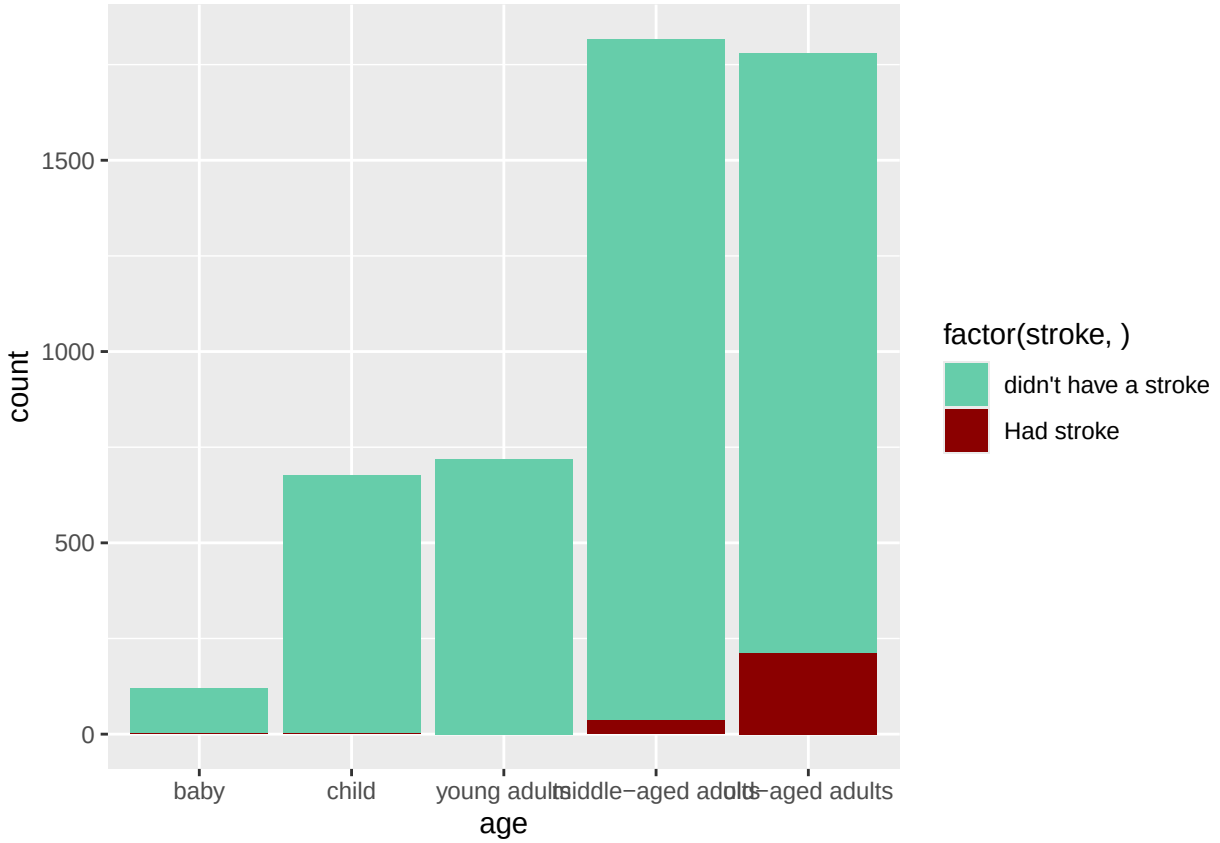
```
##
## didn't have a stroke           Had stroke
##                 4861                  249
```

### Relationship between variables and stroke

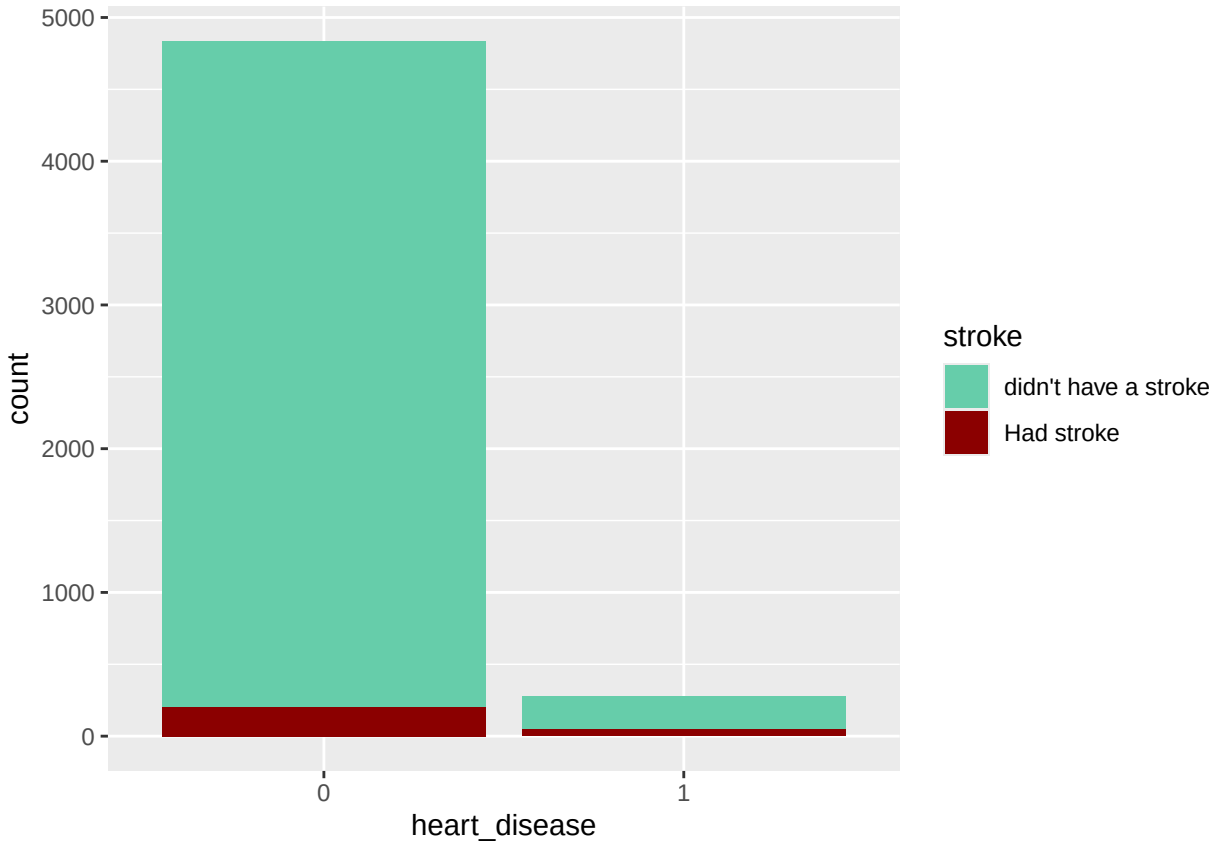
```r
# Biavariate Data Analysis
ggplot(data = data_imp,
       aes(x=smoking_status,
           fill=stroke,)) +
  geom_bar() +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```



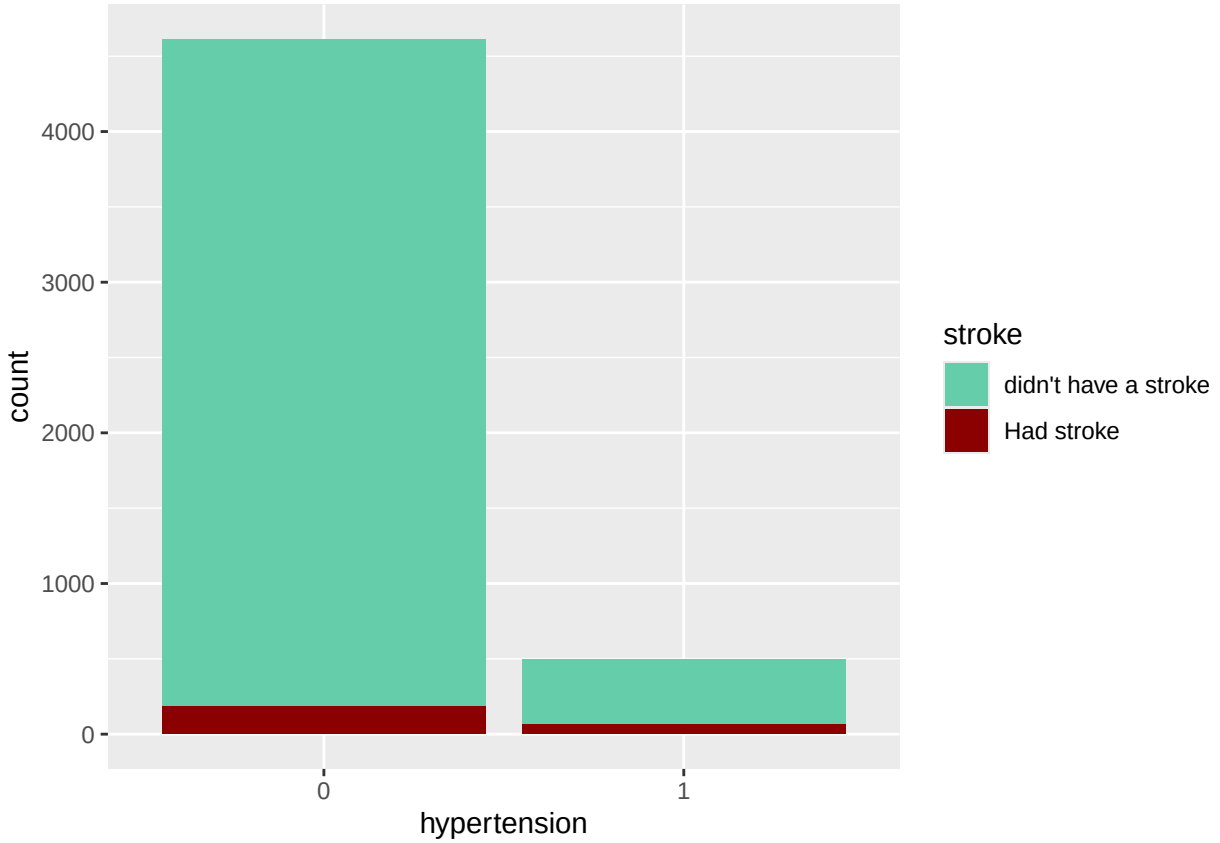

```r
ggplot(data = data_imp,
       aes(x=age,
           fill=factor(stroke,))) +
  geom_bar() +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```



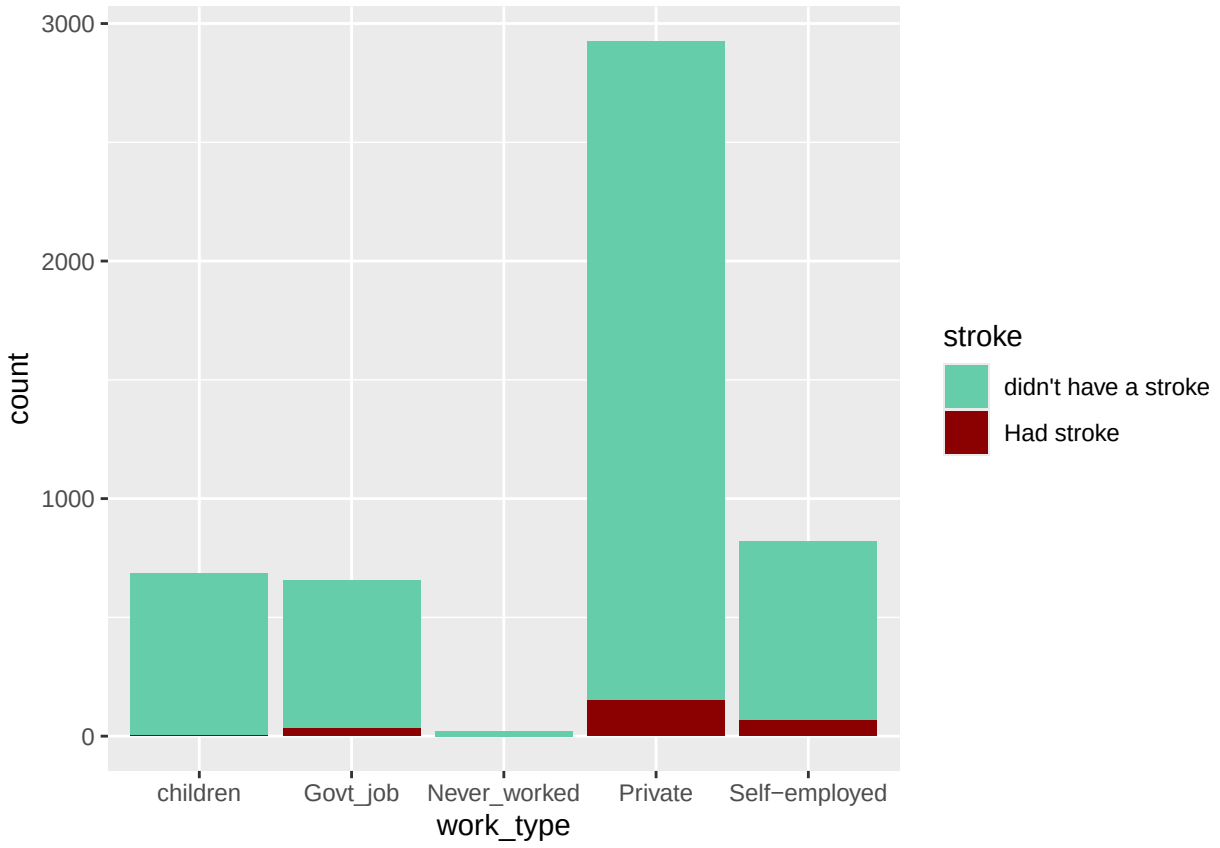

```
ggplot(data = data_imp,
       aes(x=heart_disease,
           fill=stroke,)) +
  geom_bar() +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```



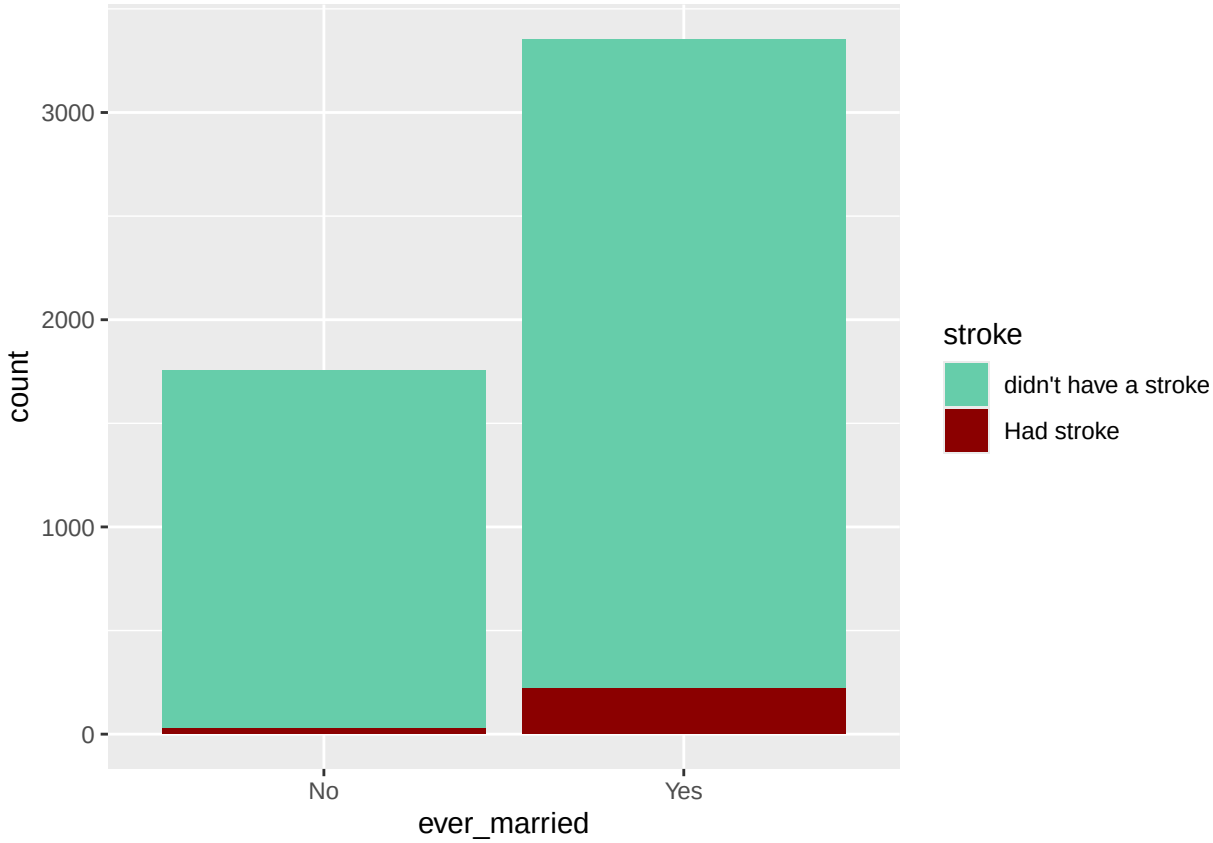

```
ggplot(data = data_imp,
       aes(x=hypertension,
           fill=stroke,)) +
  geom_bar() +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```



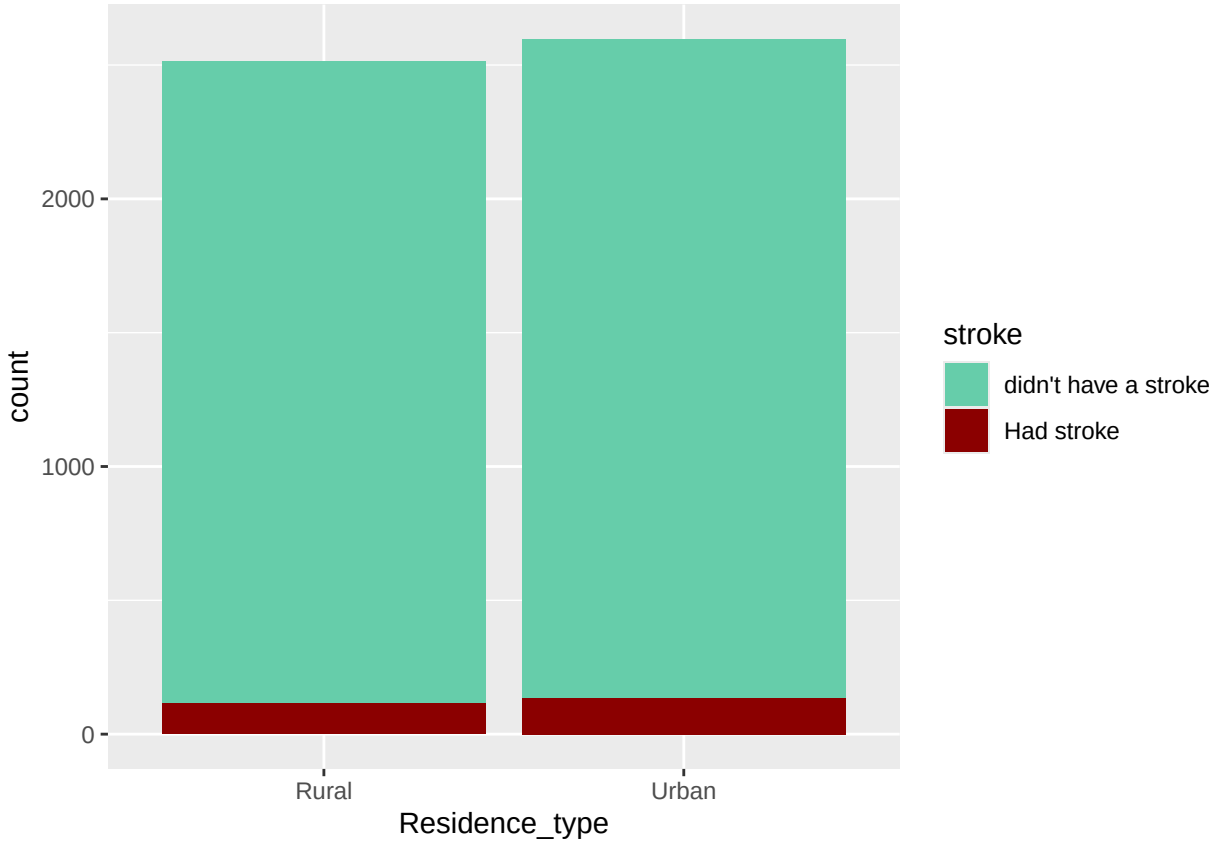

```
ggplot(data = data_imp,
       aes(x=work_type,
           fill=stroke,)) +
  geom_bar() +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```




```
ggplot(data = data_imp,
       aes(x=ever_married,
           fill=stroke,)) +
  geom_bar() +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```




```
ggplot(data = data_imp,
       aes(x=Residence_type,
           fill=stroke,)) +
  geom_bar() +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```




**Observations:**

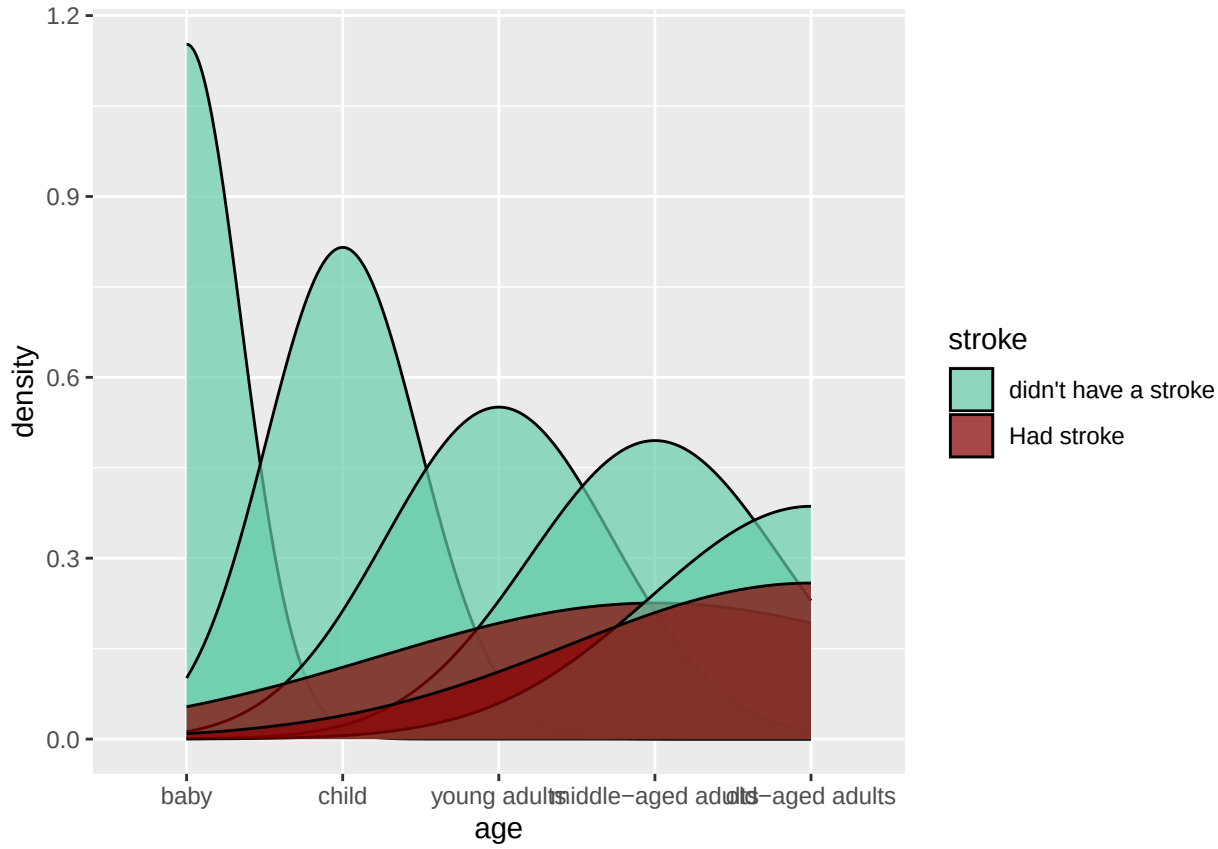
- Patients who does not have hypertension have more stroke than those that does not have hypertension.But we should consider the proportion of both groups when comparing the number of strokes for them.
- Also, patients who does not have heart disease, have more stroke than those that does not have heart disease.Again we need to check the proportion.
- Patients who are married at a point in their lives have more stroke than those that have never married.
- More patients from the private sector has stroke, followed by the self employed, and govt workers respectively.
- More insights could have been determined if we were able to know the industry these patients work.
- Patients with stroke are almost evenly spread across the rural and urban areas.
- The combination of those that formerly smoked and those that smokes has more stroke than those that never smoked.
- We also have lots of unknown smoking status that has stroke.

```
#Biavariate Data Analysis by Density Plot
#age
ggplot(data_imp, aes(x=age, fill=stroke)) +
  geom_density(alpha=0.7) +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```

```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
```

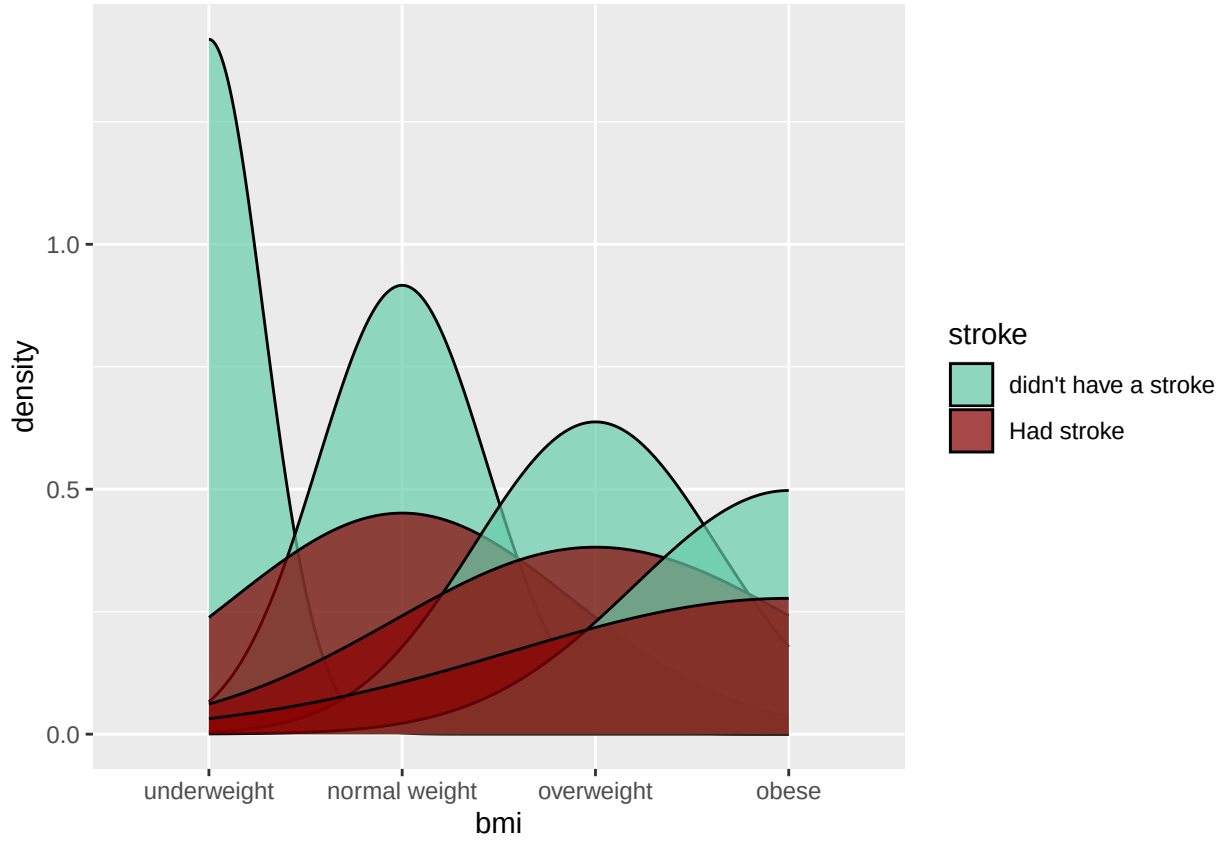
```
## -Inf
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```



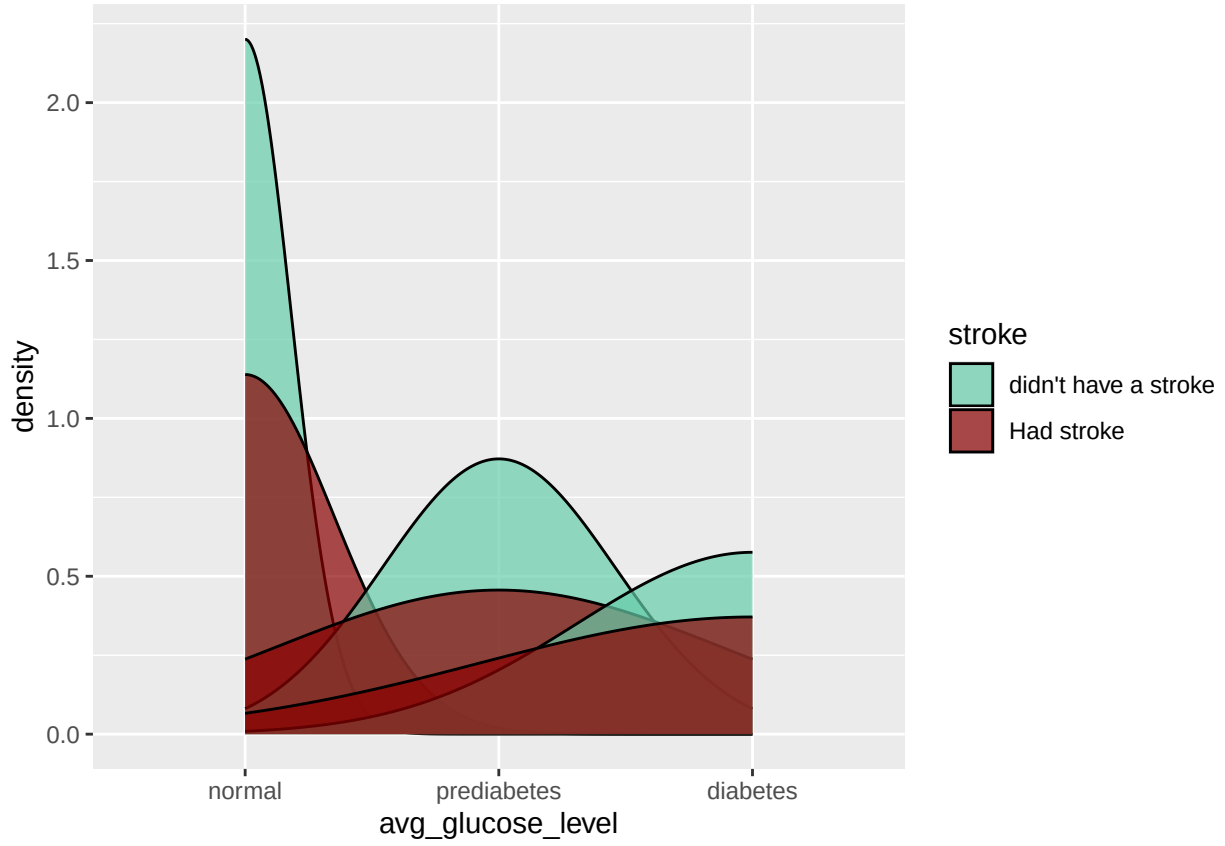

```
#bmi
ggplot(data_imp, aes(x=bmi, fill=stroke)) +
  geom_density(alpha=0.7) +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```

```
## Warning: Groups with fewer than two data points have been dropped.
```

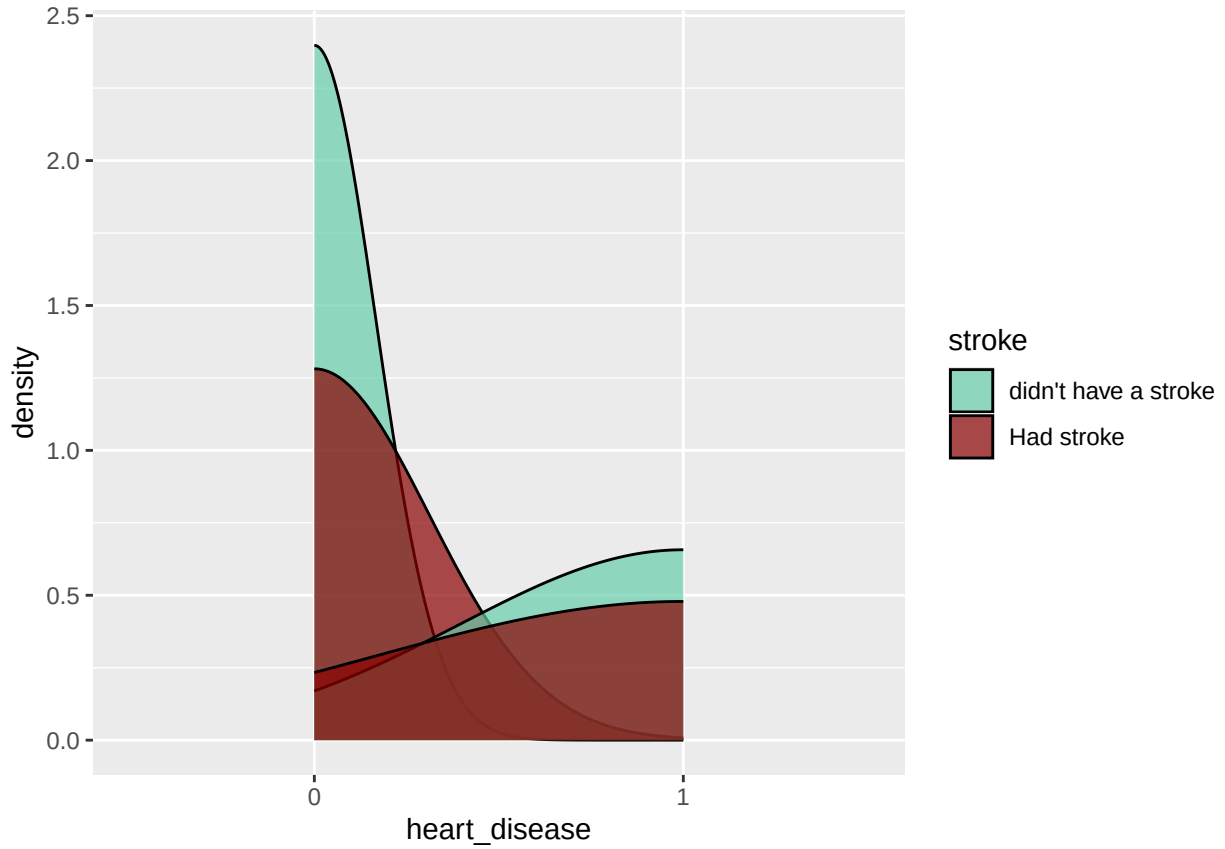
```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```



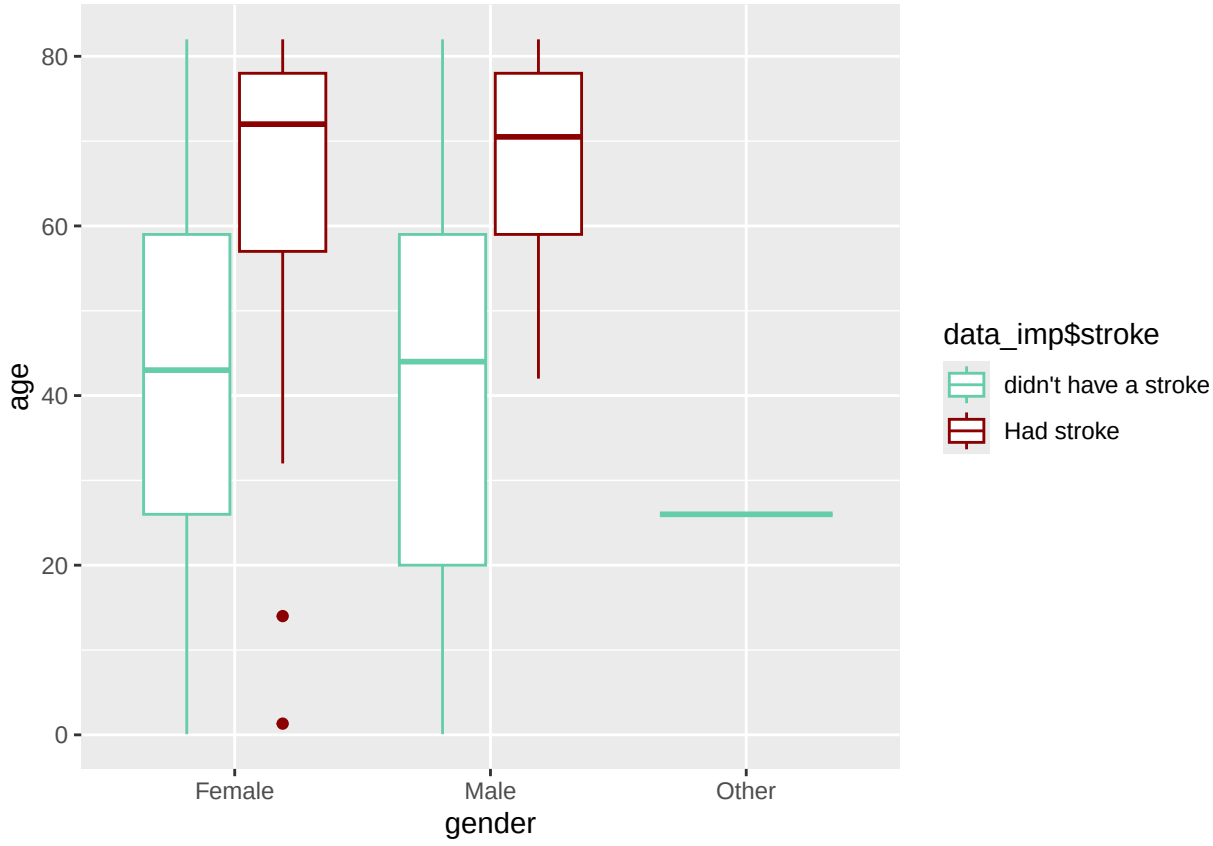

```
#avg_glucose_level
ggplot(data_imp, aes(x=avg_glucose_level, fill=stroke)) +
  geom_density(alpha=0.7) +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```




```
#heart_disease
ggplot(data_imp, aes(x=heart_disease, fill=stroke)) +
  geom_density(alpha=0.7) +
  scale_fill_manual(values=c("aquamarine3",
                             "darkred"))
```



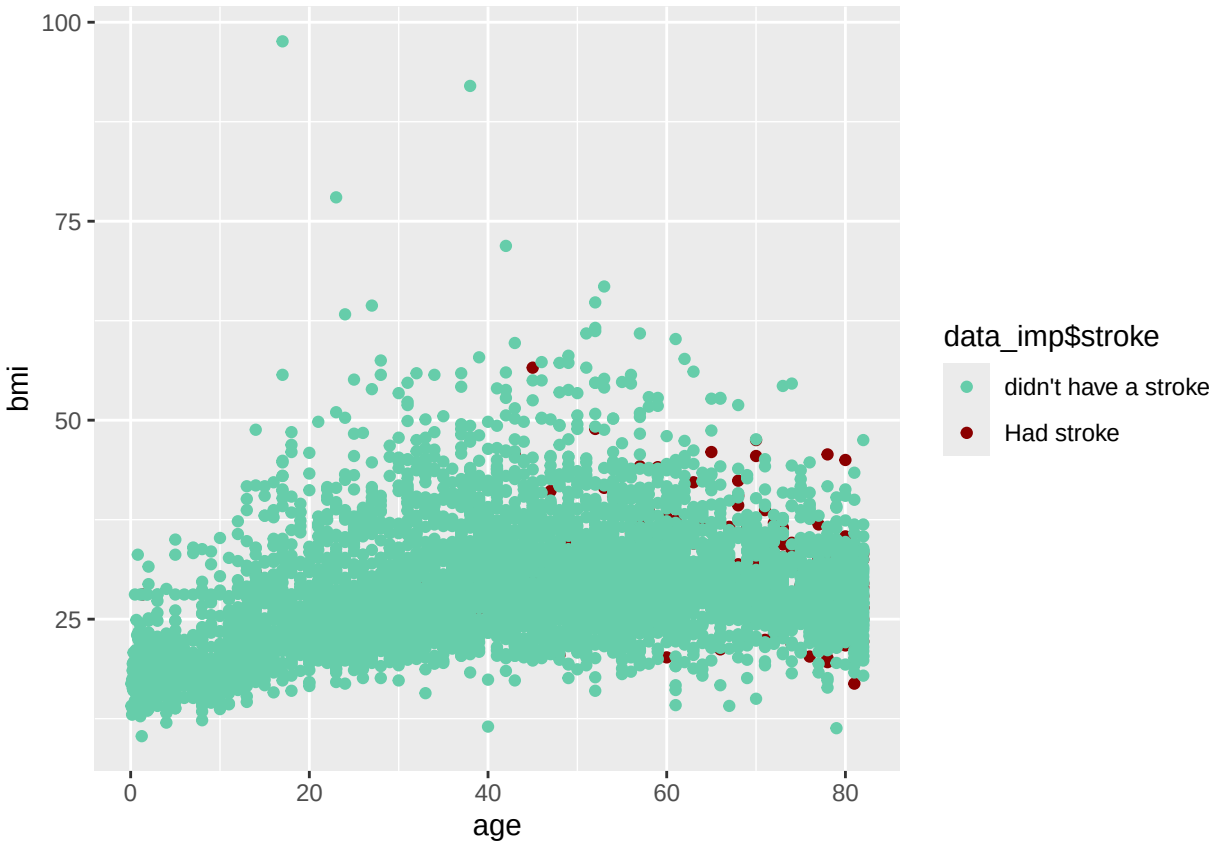

```
ggplot(data, aes(x=gender, y=age, color= data_imp$stroke)) +
  geom_boxplot() +
  scale_color_manual(values=c("aquamarine3",
                              "darkred"))
```




**Observations:**

- patients that are older seems to have more stroke with fewer number of patients who are middle aged.
- The males in the data tend to have stroke at age over 40, while women tends to have stroke from age around 30s.
- There are two children (less than 18) that have stroke.
- The underweight patients are the class with the least number of strokes, followed by the healthy weight class.
- Stroke seems to occur in patients within the overweight and obese classes.
- We have more patients with normal glucose level, and very few of them have stroke.
- The number of patients with prediabetes and diabetes condition that have stroke are fewer than those with normal glucose level.

```
ggplot(data, aes(x = age, y = bmi, color = data_imp$stroke)) +
  geom_point()+
  scale_color_manual(values=c("aquamarine3",
                              "darkred"))
```

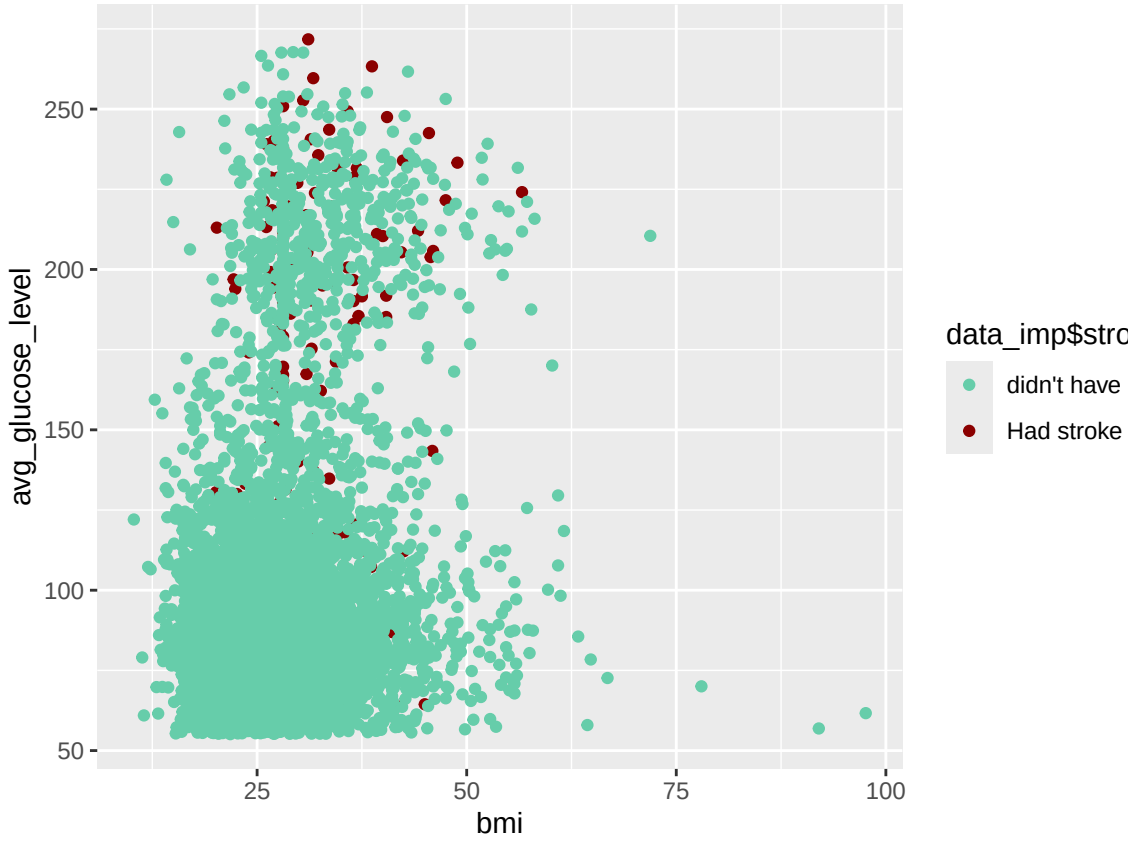



**Age vs BMI**

```
ggplot(data, aes(x = age, y = avg_glucose_level, color = data_imp$stroke)) +
  geom_point()+
  scale_color_manual(values=c("aquamarine3",
                              "darkred"))
```


**Age vs Glucose Level**

```
ggplot(data, aes(x = bmi, y = avg_glucose_level, color = data_imp$stroke)) +
  geom_point()+
  scale_color_manual(values=c("aquamarine3",
                              "darkred"))
```

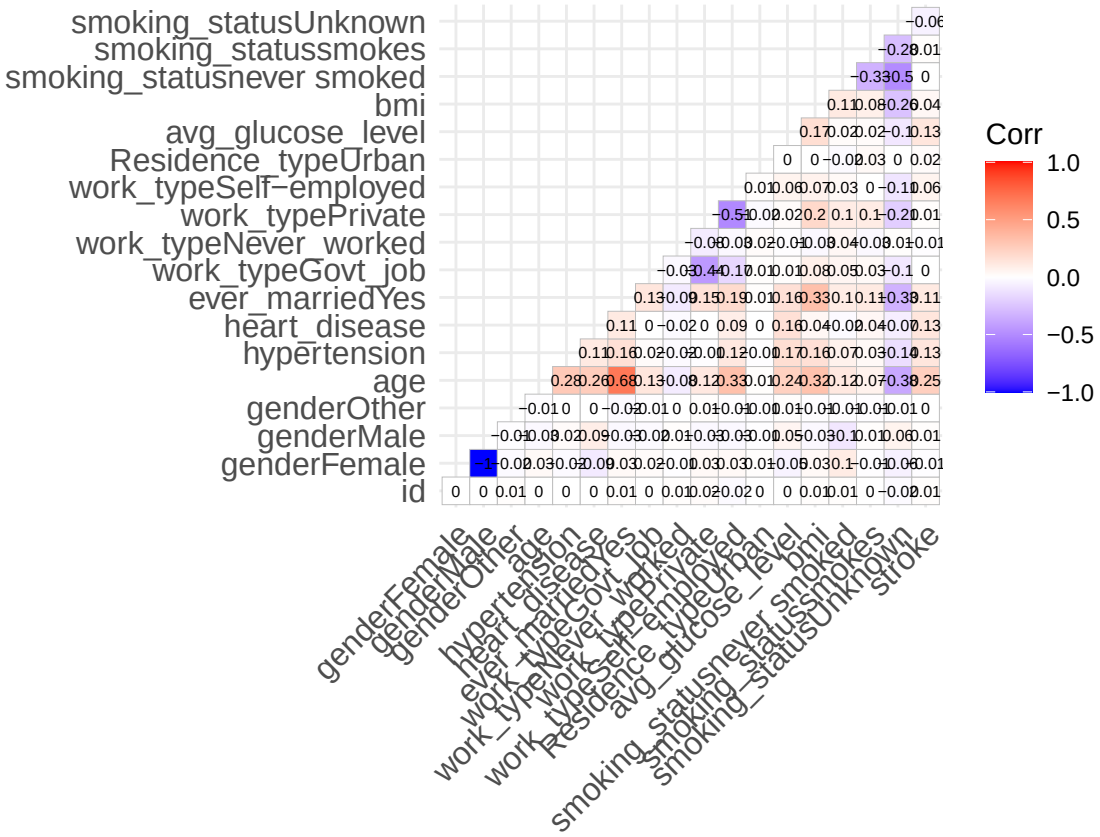



**BMI vs Glucose Level**

**Observations:**

- Most of the patients have BMI under 40, and stroke occurs more in patients over 60 years old.
- Patients with average glucose level higher than 150 and over 60 years old tends to have stroke.
- Stroke tends to happen among those with BMI over 25 and with average glucose level of over 150.

```
# visualizing correlogram
#creating correlation matrix

model.matrix(~0+., data=data) %>%
cor(use="pairwise.complete.obs") %>%
 ggcorrplot(show.diag=FALSE, type="lower", lab=TRUE, lab_size=2)
```




**Conclusions of EDA:**

1. Patients with the most strokes are old-aged adults >= 55 years old
2. Patients who have never smoked can have a stroke
3. Patients who have never smoked, do not have hypertension, do not have heart disease and are expected to maintain a healthy body, can also have a stroke.
4. Patients with a body mass index <18.5 are advised to take better care of their health by eating nutritious and protein-rich foods.
5. BMI is the least correlated with stroke, and age is the most correlated to stroke among the numerical features.

# Build prediction models

### Data Preprocessing

```
#New Data frame
data_transformed <- data.frame(data_imp)
str(data_transformed)
```

```
## '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              : Ord.factor w/ 5 levels "baby"<"child"<..: 5 5 5 4 5 5 5 5 5 5 ...
##  $ hypertension     : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 2 1 1 1 ...
##  $ heart_disease    : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 2 1 1 1 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
```

```
##  $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 4 5 4 4 5 4 4 4 4 4 ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: Ord.factor w/ 3 levels "normal"<"prediabetes"<..: 3 3 2 3 3 3 1 1 1 1 ...
##  $ bmi              : Ord.factor w/ 4 levels "underweight"<..: 4 3 4 4 2 3 3 2 3 2 ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : Factor w/ 2 levels "didn't have a stroke",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
#Remove
## remove id in dataframe
data_transformed$id <- NULL

## remove other in gender
table(data_transformed$gender)
```

**Removing id column and removing other in gender**

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

```
idx <- which(data_transformed$gender %in% c("Other"))
idx
```

```
## [1] 3117
```

```
data_transformed <- (data_transformed)[-idx,]

table(data_transformed$gender)
```

```
##
## Female   Male
##   2994   2115
```

**Label Encoding**

```
#ever married
table(data_transformed$ever_married)
```

```
##
##   No  Yes
## 1756 3353
```

```
data_transformed$ever_married <- ifelse(data_transformed$ever_married == "Yes", 1, 0)

table(data_transformed$ever_married)
```

```
##
##    0    1
## 1756 3353
```

```
#smoking status
table(data_transformed$smoking_status)
```

```
##
## formerly smoked    never smoked          smokes         Unknown
##             884            1892             789            1544
```

```r
data_transformed$smoking_status <- as.character(data_transformed$smoking_status)

for (i in 1:length(data_transformed$gender)) {
  if (data_transformed$smoking_status[i] == "Unknown") {
    data_transformed$smoking_status[i] <- 0
  }
  #never smoked is 0
  else if (data_transformed$smoking_status[i] == "never smoked") {
    data_transformed$smoking_status[i] <- 1
  }
  #formerly smoked is 20
  else if (data_transformed$smoking_status[i] == "formerly smoked") {
    data_transformed$smoking_status[i] <- 2
  }
  #smokes is 30
  else if (data_transformed$smoking_status[i] == "smokes") {
    data_transformed$smoking_status[i] <- 3
  }
}
table(data_transformed$smoking_status)
```

```
##
##    0    1    2    3
## 1544 1892  884  789
```

```r
#bmi
data_transformed$bmi <- as.character(data_transformed$bmi)

table(data_transformed$bmi)
```

```
##
## normal weight         obese    overweight   underweight
##          1242          1920          1610           337
```

```r
for (i in 1:length(data_transformed$bmi)) {
  if (data_transformed$bmi[i] == "obese") {
    data_transformed$bmi[i] <- 3
  }
  else if (data_transformed$bmi[i] == "overweight") {
    data_transformed$bmi[i] <- 2
  }
  #bmi
  else if (data_transformed$bmi[i] == "normal weight") {
    data_transformed$bmi[i] <- 0
  }
  #bmi
  else if (data_transformed$bmi[i] == "underweight") {
    data_transformed$bmi[i] <- 1
  }
}

table(data_transformed$bmi)
```

```
##
##    0    1    2    3
```

```
## 1242  337 1610 1920
```

```
# avg glucose
data_transformed$avg_glucose_level <- as.character(data_transformed$avg_glucose_level)

table(data_imp$avg_glucose_level)
```

```
##
##      normal prediabetes    diabetes
##        3131         979        1000
```

```
for (i in 1:length(data_transformed$gender)) {
  if (data_transformed$avg_glucose_level[i] == "normal") {
    data_transformed$avg_glucose_level[i] <- 0
  }
  else if (data_transformed$avg_glucose_level[i] == "prediabetes") {
    data_transformed$avg_glucose_level[i] <- 1
  }
  else if (data_transformed$avg_glucose_level[i] == "diabetes") {
    data_transformed$avg_glucose_level[i] <- 2
  }
}
table(data_transformed$avg_glucose_level)
```

```
##
##    0    1    2
## 3131  979  999
```

```
#age
data_transformed$age <- as.character(data_transformed$age)

table(data_transformed$age)
```

```
##
##               baby              child middle-aged adults    old-aged adults
##                120                676               1816               1779
##       young adults
##                718
```

```
for (i in 1:length(data_transformed$age)) {
  if (data_transformed$age[i] == "baby") {
    data_transformed$age[i] <- 0
  }
  else if (data_transformed$age[i] == "child") {
    data_transformed$age[i] <- 1
  }
  else if (data_transformed$age[i] == "middle-aged adults") {
    data_transformed$age[i] <- 2
  }
  else if (data_transformed$age[i] == "old-aged adults") {
    data_transformed$age[i] <- 3
  }
  else if (data_transformed$age[i] == "young adults") {
    data_transformed$age[i] <- 4
  }
}
```

```
table(data_transformed$age)
```

```
##
##    0    1    2    3    4
##  120  676 1816 1779  718
```

### One Hot Encoding

- Label Encoding will be used for the ordinal features so we can preserve the order of the categories
- One Hot Encoding will be used for other nominal features since there are no inherent order in the categories.

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
# data split
df1 <- data_transformed[, 2:5]

df2 <- data_transformed[, 8:11]


df3 <- data.frame(data_transformed$gender,
                        data_transformed$work_type,
                        data_transformed$Residence_type)

df4 <- dummyVars("~.", data = df3)
df5 <- data.frame(predict(df4, df3))

# combinasi data set
final <- cbind(df1,df2,df5)

str(final)
```

```
## 'data.frame':    5109 obs. of  17 variables:
##  $ age                                    : chr  "3" "3" "3" "2" ...
##  $ hypertension                           : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 2 1 1 1 ...
##  $ heart_disease                          : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 2 1 1 1 ...
##  $ ever_married                           : num  1 1 1 1 1 1 1 0 1 1 ...
##  $ avg_glucose_level                      : chr  "2" "2" "1" "2" ...
##  $ bmi                                    : chr  "3" "2" "3" "3" ...
##  $ smoking_status                         : chr  "2" "1" "1" "3" ...
##  $ stroke                                 : Factor w/ 2 levels "didn't have a stroke",..: 2 2 2 2 2
##  $ data_transformed.genderFemale          : num  0 1 0 1 1 0 0 1 1 1 ...
##  $ data_transformed.genderMale            : num  1 0 1 0 0 1 1 0 0 0 ...
##  $ data_transformed.work_type.children    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ data_transformed.work_type.Govt_job    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ data_transformed.work_type.Never_worked : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ data_transformed.work_type.Private     : num  1 0 1 1 0 1 1 1 1 1 ...
##  $ data_transformed.work_type.Self.employed: num  0 1 0 0 1 0 0 0 0 0 ...
```

```
##  $ data_transformed.Residence_typeRural    : num  0 1 1 0 1 0 1 0 1 0 ...
##  $ data_transformed.Residence_typeUrban    : num  1 0 0 1 0 1 0 1 0 1 ...
```

```
## convert to factor
final$smoking_status <- factor(final$smoking_status)
final$avg_glucose_level <- factor(final$avg_glucose_level)
final$bmi <- factor(final$bmi)
final$age <- factor(final$age)
final$ever_married <- factor(final$ever_married)
final$data_transformed.genderFemale <- factor(final$data_transformed.genderFemale )
final$data_transformed.genderMale  <- factor(final$data_transformed.genderMale)
final$data_transformed.work_type.children <- factor(final$data_transformed.work_type.children)
final$data_transformed.work_type.Govt_job <- factor(final$data_transformed.work_type.Govt_job)
final$data_transformed.work_type.Private <- factor(final$data_transformed.work_type.Private)
final$data_transformed.work_type.Self.employed <- factor(final$data_transformed.work_type.Self.employed)
final$data_transformed.work_type.Never_worked <- factor(final$data_transformed.work_type.Never_worked)
final$data_transformed.Residence_typeRural  <- factor(final$data_transformed.Residence_typeRural)
final$data_transformed.Residence_typeUrban <- factor(final$data_transformed.Residence_typeUrban)
str(final)
```

```
## 'data.frame':    5109 obs. of  17 variables:
##  $ age                                  : Factor w/ 5 levels "0","1","2","3",..: 4 4 4 3 4 4 4 4 4
##  $ hypertension                         : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 2 1 1 1 ...
##  $ heart_disease                        : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 2 1 1 1 ...
##  $ ever_married                         : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 2 ...
##  $ avg_glucose_level                    : Factor w/ 3 levels "0","1","2": 3 3 2 3 3 3 1 1 1 1 ...
##  $ bmi                                  : Factor w/ 4 levels "0","1","2","3": 4 3 4 4 1 3 3 1 3 1
##  $ smoking_status                       : Factor w/ 4 levels "0","1","2","3": 3 2 2 4 2 3 2 2 1 1
##  $ stroke                               : Factor w/ 2 levels "didn't have a stroke",..: 2 2 2 2 2
##  $ data_transformed.genderFemale        : Factor w/ 2 levels "0","1": 1 2 1 2 2 1 1 2 2 2 ...
##  $ data_transformed.genderMale          : Factor w/ 2 levels "0","1": 2 1 2 1 1 2 2 1 1 1 ...
##  $ data_transformed.work_type.children  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ data_transformed.work_type.Govt_job  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ data_transformed.work_type.Never_worked : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ data_transformed.work_type.Private   : Factor w/ 2 levels "0","1": 2 1 2 2 1 2 2 2 2 2 ...
##  $ data_transformed.work_type.Self.employed: Factor w/ 2 levels "0","1": 1 2 1 1 2 1 1 1 1 1 ...
##  $ data_transformed.Residence_typeRural : Factor w/ 2 levels "0","1": 1 2 2 1 2 1 2 1 2 1 ...
##  $ data_transformed.Residence_typeUrban : Factor w/ 2 levels "0","1": 2 1 1 2 1 2 1 2 1 2 ...
```

### Train & Test dataset

```
row <- dim(final)[1]


train_idx <- sample(row, 0.7 * row)


training_data <- final[train_idx,]
testing_data <- final[-train_idx,]
```

### imbalanced Data

```
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```r
library(rpart)
```

```r
training_data %>%
  group_by(stroke) %>%
  summarize(n = n()) %>%
  mutate(prop = round(n / sum(n), 2))
```

```
## # A tibble: 2 x 3
##   stroke                   n  prop
##   <fct>                <int> <dbl>
## 1 didn't have a stroke  3397  0.95
## 2 Had stroke             179  0.05
```

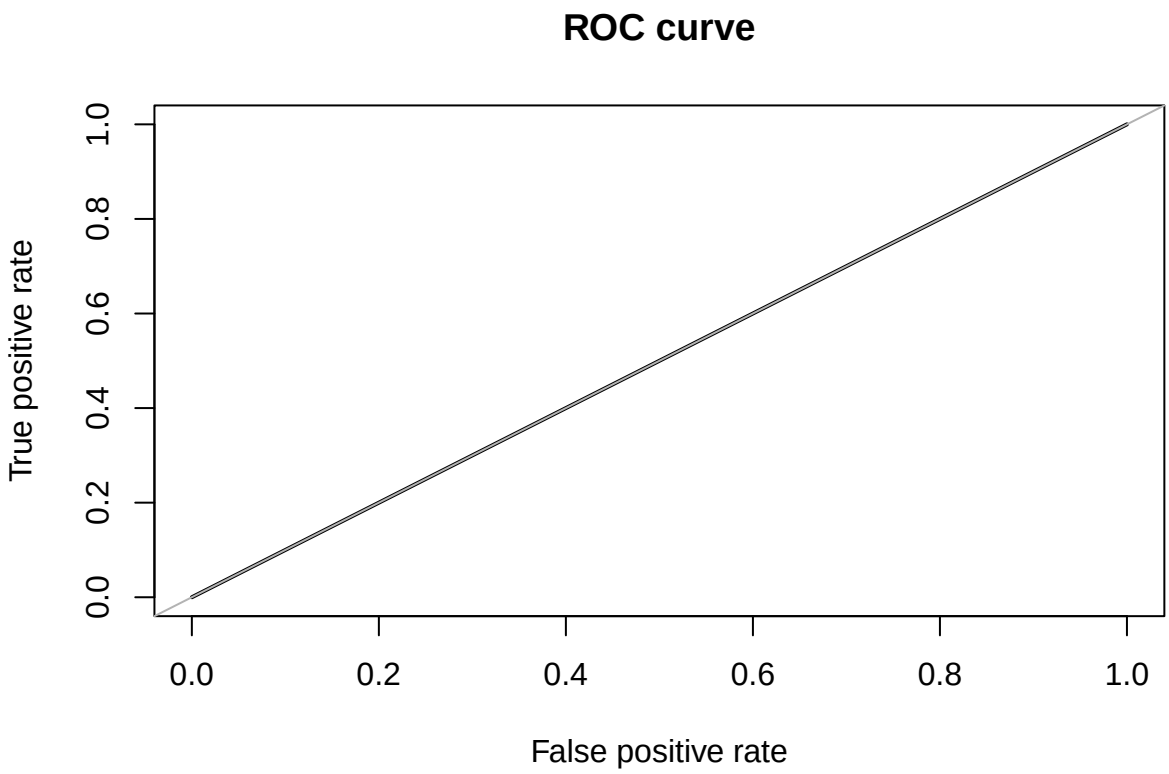
### 1. Dcision Tree

```r
ti <- rpart(stroke~., data = training_data)
pred.ti <- predict(ti, newdata = testing_data)


answer <- testing_data$stroke


accuracy.meas(answer, pred.ti[,2])
```

```
##
## Call:
## accuracy.meas(response = answer, predicted = pred.ti[, 2])
##
## Examples are labelled as positive when predicted is greater than 0.5
##
## precision: NaN
## recall: 0.000
## F: NaN
```

### AUC

```r
# AUC ( Area under the curve)
roc.curve(answer, pred.ti[,2])
```




```
## Area under the curve (AUC): 0.500
```

### Oversampling and Undersampling

```
# Over Sampling
training_data %>%
  group_by(stroke) %>%
  summarize(n = n()) %>%
  mutate(prop = round(n / sum(n), 2))
```

```
## # A tibble: 2 x 3
##   stroke                    n  prop
##   <fct>                 <int> <dbl>
## 1 didn't have a stroke  3397  0.95
## 2 Had stroke             179  0.05
```

```
table(training_data$stroke)
```

```
##
## didn't have a stroke           Had stroke
##                 3397                  179
```

```
data_balanced_over <- ovun.sample(stroke~.,
                                  data = training_data,
                                  method = "over",
                                  N = 6810)$data # N =  0 x 2
```

```r
data_balanced_over %>%
  group_by(stroke) %>%
  summarize(n = n()) %>%
  mutate(prop = round(n / sum(n), 2))
```

```
## # A tibble: 2 x 3
##   stroke                   n  prop
##   <fct>                <int> <dbl>
## 1 didn't have a stroke  3397   0.5
## 2 Had stroke            3413   0.5
```

```r
# Undersampling
data_balanced_under <- ovun.sample(stroke~.,
                                   data = training_data,
                                   method = "under",
                                   N = 342, # data 1 x2
                                   seed = 1)$data

table(data_balanced_under$stroke)
```

```
##
## didn't have a stroke           Had stroke
##                  163                  179
```

```r
# BOth => Undersampling + Oversampling
data_balanced_both <- ovun.sample(stroke~.,
                                  data = training_data,
                                  p=0.5,
                                  N = 3577, # N= data train
                                  seed = 1)$data

table(data_balanced_both$stroke)
```

```
##
## didn't have a stroke           Had stroke
##                 1838                 1739
```

```r
data.rose <- ROSE(stroke~.,
                  data = training_data,
                  seed = 1)$data


table(data.rose$stroke)
```

```
##
## didn't have a stroke           Had stroke
##                 1838                 1738
```

### 2. Logistic Regression

```r
logit <- glm(formula = stroke~.,
             data=data.rose,
             family=binomial)
```

```r
answer <- testing_data$stroke
```

```r
pred.prob <- predict(logit,
                     testing_data,
                     type="response")
```

```r
# pred < 0.5 => class 0 stroke
# pred >= 0.5 => class 1 no stroke

pred.logit <- ifelse(pred.prob > 0.5, "YES", "NO")

table(pred.logit)
```

```
## pred.logit
##   NO  YES
## 1033  500
```

### 3. Dicision Tree

```r
if(!require(multcomp)) install.packages("multcomp",repos = "http://cran.us.r-project.org")
```

```
## Loading required package: multcomp
## Loading required package: mvtnorm
## Loading required package: survival
##
## Attaching package: 'survival'
## The following object is masked from 'package:caret':
##
##     cluster
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##     select
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
##     geyser
```

```r
if(!require(party)) install.packages("party",repos = "http://cran.us.r-project.org")
```

```
## Loading required package: party
## Loading required package: grid
## Loading required package: modeltools
## Loading required package: stats4
## Loading required package: strucchange
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:data.table':
##
##     yearmon, yearqtr
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package: sandwich
##
## Attaching package: 'strucchange'
## The following object is masked from 'package:stringr':
##
##     boundary
##
## Attaching package: 'party'
## The following object is masked from 'package:dplyr':
##
##     where
```

```
library(multcomp)

library(party)
```

```
dt <- ctree(formula = stroke~.,
            data=data_balanced_over)

pred.dt <- predict(dt,
                   testing_data)
```

### 4. Random Forest

```
if(!require(randomForest)) install.packages("randomForest",repos = "http://cran.us.r-project.org")
```

```
## Loading required package: randomForest
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:dplyr':
##
##     combine
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(randomForest)
```

```r
rf <- randomForest(formula=stroke~.,
                   data=data_balanced_both)

pred.rf <- predict(rf,
                   testing_data)



performance <- function(prediction, actual, nama_model){
  #confusion matrix
  cm <- table(actual, prediction,
              dnn = c("Actual", "Prediction"))
  #dnn -> The dimension Names

  TP <- cm[2, 2]
  TN <- cm[1, 1]
  FN <- cm[2, 1]
  FP <- cm[1, 2]

  accuracy <- (TP + TN) / (TP + TN + FP + FN)
  precision <- TP / (TP / FP)
  Recall <- TP / (TP +FN)
  f1_score <- (2*precision*Recall) / (precision + Recall)

  result <- paste("Model : ", nama_model,
                  "\nAccuracy : ", round(accuracy, 3),
                  "\nPrecision : ", round(precision, 3),
                  "\nRecall : ", round(Recall, 3),
                  "\nf1 Score : ", round(f1_score, 3))

  cat(result)
}
```

# Evaluate and select prediction models

### Logistic Regression

```r
performance(pred.logit, answer, "Logistic Regression")
```

```
## Model :  Logistic Regression
## Accuracy :  0.701
## Precision :  444
## Recall :  0.8
## f1 Score :  1.597
```

The model's performance in terms of accuracy is fair. A good Recall and poor Precision resulted in the poor F1-score. The model predicted 456 non stroke as stroke which resulted in the poor precision. Since we are looking to predict a medical diagnosis, it's important to have an very good Recall and Precision.

**Descision Tree**

```
performance(pred.dt, answer, "Decision Tree")
```

```
## Model :  Decision Tree
## Accuracy :  0.727
## Precision :  392
## Recall :  0.614
## f1 Score :  1.227
```

The model's performance on the test set is poor

**Random Forest**

```
performance(pred.rf, answer, "Random Forest")
```

```
## Model :  Random Forest
## Accuracy :  0.757
## Precision :  342
## Recall :  0.571
## f1 Score :  1.141
```

# Findings and Conclusions

Precision and recall are indeed critical metrics in medical diagnosis, as false positive and false negative predictions can have serious consequences. In the context of stroke prediction, it is important to accurately identify stroke cases to ensure appropriate interventions and timely treatment.

The results of the models in terms of precision, recall, F1-score, indicate that they faced challenges in correctly identifying stroke cases. This can be attributed to the significant class imbalance between non-stroke and stroke instances in the test set, with a much larger number of non-stroke instances compared to stroke instances. This class imbalance creates a bias in the models towards predicting the majority class, which in this case is non-stroke.

Among the models, logistic regression stands out with a high recall of 0.78. This suggests that the model was successful in correctly identifying a large proportion of the actual stroke cases in the dataset. However, the low precision value indicates that the model also classified a considerable number of non-stroke cases as strokes, resulting in a high rate of false positive predictions.

On the other hand, the remaining models, including decision tree, and random forest, demonstrate relatively lower values for precision, recall, and F1-score. These models seem to perform better in predicting non-stroke cases accurately rather than identifying stroke cases.

In conclusion, the results suggest that logistic regression and random forest have potential for predicting strokes, with random forest showing the most promising performance. These findings have implications for healthcare providers, as accurate prediction of strokes can help in early identification, prevention, and appropriate allocation of resources for stroke management.