Stroke Prediction Model

KRISHNANDAN SHA

2026-01-09

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 Fri Jan 9 14:20:15 2026.

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.

##Task One: Import data and data preprocessing

Load data and install packages

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df <- readr::read_csv("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.
```

```
glimpse(df)
```

```
## 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~
```

Describe and explore the data

```
# Structure of data
str(df)
```

```
## spc_tbl_ [5,110 x 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>
```

```
# Dimensions: rows & columns
dim(df)
```

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

```
# Column names
colnames(df)
```

```
##  [1] "id"                "gender"            "age"              
##  [4] "hypertension"      "heart_disease"     "ever_married"     
##  [7] "work_type"         "Residence_type"    "avg_glucose_level"
## [10] "bmi"               "smoking_status"    "stroke"           
```

```
# Quick tibble-style overview (best for large data)
dplyr::glimpse(df)
```

```
## 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~
```

```
summary(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  
```

```
#Separate numerical & categorical summaries

df %>%
  select(where(is.numeric)) %>%
  summary()
```

```
##        id              age         hypertension     heart_disease    
##  Min.   :   67   Min.   : 0.08   Min.   :0.00000   Min.   :0.00000  
##  1st Qu.:17741   1st Qu.:25.00   1st Qu.:0.00000   1st Qu.:0.00000  
##  Median :36932   Median :45.00   Median :0.00000   Median :0.00000
##  Mean   :36518   Mean   :43.23   Mean   :0.09746   Mean   :0.05401
##  3rd Qu.:54682   3rd Qu.:61.00   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :72940   Max.   :82.00   Max.   :1.00000   Max.   :1.00000
##  avg_glucose_level     stroke
##  Min.   : 55.12    Min.   :0.00000
##  1st Qu.: 77.25    1st Qu.:0.00000
##  Median : 91.89    Median :0.00000
##  Mean   :106.15    Mean   :0.04873
##  3rd Qu.:114.09    3rd Qu.:0.00000
##  Max.   :271.74    Max.   :1.00000
```

```
#Categorical variables frequency

df %>%
  select(where(is.character)) %>%
  lapply(table)
```

```
## $gender
##
## Female   Male  Other
##   2994   2115      1
##
## $ever_married
##
##   No  Yes
## 1757 3353
##
## $work_type
##
##      children      Govt_job  Never_worked       Private Self-employed
##           687           657            22          2925           819
##
## $Residence_type
##
## Rural Urban
##  2514  2596
##
## $bmi
##
## 10.3 11.3 11.5   12 12.3 12.8   13 13.2 13.3 13.4 13.5 13.7 13.8 13.9   14 14.1
##    1    1    1    1    1    1    1    1    1    1    1    2    2    1    1    5
## 14.2 14.3 14.4 14.5 14.6 14.8 14.9   15 15.1 15.2 15.3 15.4 15.5 15.6 15.7 15.8
##    4    3    2    2    4    4    1    2    8    4    4    3    5    3    3    5
## 15.9   16 16.1 16.2 16.3 16.4 16.5 16.6 16.7 16.8 16.9   17 17.1 17.2 17.3 17.4
##    5    8    8   10   10   11    4    8   11    7    7   10   12   11    9   13
## 17.5 17.6 17.7 17.8 17.9   18 18.1 18.2 18.3 18.4 18.5 18.6 18.7 18.8 18.9   19
##    7   16   13    7    7   16   12    9   17   10   12   19   13   15    8    7
## 19.1 19.2 19.3 19.4 19.5 19.6 19.7 19.8 19.9   20 20.1 20.2 20.3 20.4 20.5 20.6
##   11   13    9   14   21    7    8   17    9   17   25   16   17   23   18   15
## 20.7 20.8 20.9   21 21.1 21.2 21.3 21.4 21.5 21.6 21.7 21.8 21.9   22 22.1 22.2
##   12   17   13   17   16   16   21   22   27   16   14   18   14   15   22   30
## 22.3 22.4 22.5 22.6 22.7 22.8 22.9   23 23.1 23.2 23.3 23.4 23.5 23.6 23.7 23.8
##   18   22   13   18   25   25   16   27   21   20   19   36   31   24   16   22
## 23.9   24 24.1 24.2 24.3 24.4 24.5 24.6 24.7 24.8 24.9   25 25.1 25.2 25.3 25.4
##   24   28   28   29   26   23   26   22   22   31   27   27   34   18   28   26
## 25.5 25.6 25.7 25.8 25.9   26 26.1 26.2 26.3 26.4 26.5 26.6 26.7 26.8 26.9   27
##   33   21   15   24   24   25   37   27   23   34   30   29   37   21   34   35
## 27.1 27.2 27.3 27.4 27.5 27.6 27.7 27.8 27.9   28 28.1 28.2 28.3 28.4 28.5 28.6
##   28   24   36   22   29   37   37   23   28   28   29   25   30   38   27   27
## 28.7 28.8 28.9   29 29.1 29.2 29.3 29.4 29.5 29.6 29.7 29.8 29.9   30 30.1 30.2
##   41   26   31   26   29   26   22   30   26   26   27   23   26   27   26   20
## 30.3 30.4 30.5 30.6 30.7 30.8 30.9   31 31.1 31.2 31.3 31.4 31.5 31.6 31.7 31.8
##   30   17   24   18   23   21   27   22   26   19   21   30   27   21   14   24
## 31.9   32 32.1 32.2 32.3 32.4 32.5 32.6 32.7 32.8 32.9   33 33.1 33.2 33.3 33.4
##   22   21   24   20   28   19   21   19   21   25   13   15   25   17   15   16
## 33.5 33.6 33.7 33.8 33.9   34 34.1 34.2 34.3 34.4 34.5 34.6 34.7 34.8 34.9   35
##   23   11   19   13   13   17   15   17   18   18   21   11   20   15   11   12
## 35.1 35.2 35.3 35.4 35.5 35.6 35.7 35.8 35.9   36 36.1 36.2 36.3 36.4 36.5 36.6
##   10   16   12    9   13   15   13   24   18   11    7   12   13   10    4   14
## 36.7 36.8 36.9   37 37.1 37.2 37.3 37.4 37.5 37.6 37.7 37.8 37.9   38 38.1 38.2
##   15    8   13   10    7    9   13   11    9    9    7   10   11   13   10    9
## 38.3 38.4 38.5 38.6 38.7 38.8 38.9   39 39.1 39.2 39.3 39.4 39.5 39.6 39.7 39.8
##    2    6    7    9   11   10    8    7    8   10    6   10    8    9    8    5
## 39.9   40 40.1 40.2 40.3 40.4 40.5 40.6 40.7 40.8 40.9   41 41.1 41.2 41.3 41.4
##    5    6   10   10    8    9    7    1    1    7    6    3    7    8    6    3
## 41.5 41.6 41.7 41.8 41.9   42 42.1 42.2 42.3 42.4 42.5 42.6 42.7 42.8 42.9   43
##    8    5    7   11    5    3    3    8    5    5    2    4    4    3    3    8
## 43.1 43.2 43.3 43.4 43.6 43.7 43.8 43.9   44 44.1 44.2 44.3 44.4 44.5 44.6 44.7
##    4    4    5    6    4    7    9    8    4    1    4    3    1    4    2    6
## 44.8 44.9   45 45.1 45.2 45.3 45.4 45.5 45.7 45.8 45.9   46 46.1 46.2 46.3 46.4
##    4    2    5    2    3    4    4    4    2    1    2    4    2    2    1    1
## 46.5 46.6 46.8 46.9 47.1 47.3 47.4 47.5 47.6 47.8 47.9   48 48.1 48.2 48.3 48.4
##    2    1    1    2    1    2    1    3    3    2    1    1    1    1    2    1
## 48.5 48.7 48.8 48.9 49.2 49.3 49.4 49.5 49.8 49.9 50.1 50.2 50.3 50.4 50.5 50.6
##    2    1    2    3    1    3    1    2    3    1    2    4    2    1    1    2
## 50.8 50.9   51 51.5 51.7 51.8 51.9 52.3 52.5 52.7 52.8 52.9 53.4 53.5 53.8 53.9
##    1    1    1    1    1    1    2    1    1    2    3    1    2    1    2    1
##   54 54.1 54.2 54.3 54.6 54.7 54.8   55 55.1 55.2 55.7 55.9   56 56.1 56.6 57.2
##    1    1    1    1    2    3    1    2    1    1    4    2    1    1    2    2
## 57.3 57.5 57.7 57.9 58.1 59.7 60.2 60.9 61.2 61.6 63.3 64.4 64.8 66.8 71.9   78
##    1    1    1    1    1    1    1    2    1    1    1    1    1    1    1    1
##   92 97.6  N/A
##    1    1  201
##
## $smoking_status
##
## formerly smoked    never smoked          smokes         Unknown
##             885            1892             789            1544
```

```
#Missing values check (critical for EDA)
colSums(is.na(df))
```

```
##                id            gender               age      hypertension
##                 0                 0                 0                 0
##     heart_disease      ever_married         work_type    Residence_type
##                 0                 0                 0                 0
## avg_glucose_level               bmi    smoking_status            stroke
##                 0                 0                 0                 0
```

```
#Target variable distribution (business-critical)

## Stroke count
table(df$stroke)
```

```
## 
##    0    1 
## 4861  249
```

```
## Stroke percentage
prop.table(table(df$stroke)) * 100
```

```
## 
##         0         1 
## 95.127202  4.872798
```

```
#Unique values per column

sapply(df, function(x) length(unique(x)))
```

```
##                id            gender               age      hypertension 
##              5110                 3               104                 2 
##     heart_disease      ever_married         work_type    Residence_type 
##                 2                 2                 5                 2 
## avg_glucose_level               bmi    smoking_status            stroke 
##              3979               419                 4                 2
```

```
#Quick data quality checklist

## Check duplicates
sum(duplicated(df$id))
```

```
## [1] 0
```

```
## Check impossible ages
summary(df$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.08   25.00   45.00   43.23   61.00   82.00
```

```
#CLEAN & FIX THE DATA
df <- df %>%
  mutate(
    hypertension  = factor(hypertension, levels = c(0,1), labels = c("No","Yes")),
    heart_disease = factor(heart_disease, levels = c(0,1), labels = c("No","Yes")),
    stroke        = factor(stroke, levels = c(0,1), labels = c("No","Yes"))
  )

#Fix BMI column
df <- df %>%
  mutate(
    bmi = as.numeric(ifelse(bmi == "N/A", NA, bmi))
  )
```

```r
#Handle missing values
colSums(is.na(df))
```

```
##                id            gender               age      hypertension 
##                 0                 0                 0                 0 
##     heart_disease      ever_married         work_type    Residence_type 
##                 0                 0                 0                 0 
## avg_glucose_level               bmi    smoking_status            stroke 
##                 0               201                 0                 0
```

```r
#Logical imputation for BMI
median_bmi <- median(df$bmi, na.rm = TRUE)

df <- df %>%
  mutate(
    bmi = ifelse(is.na(bmi), median_bmi, bmi)
  )

#Standardize categorical values (data hygiene)
df <- df %>%
  mutate(
    gender = trimws(gender),
    smoking_status = trimws(smoking_status),
    work_type = trimws(work_type),
    Residence_type = trimws(Residence_type)
  )

#Convert categorical columns to factors (model-ready)
df <- df %>%
  mutate(
    gender         = factor(gender),
    ever_married   = factor(ever_married),
    work_type      = factor(work_type),
    Residence_type = factor(Residence_type),
    smoking_status = factor(smoking_status)
  )

#Check duplicates (defensive analytics)
sum(duplicated(df$id))
```

```
## [1] 0
```

```r
#Validation
str(df)
```

```
## tibble [5,110 x 12] (S3: tbl_df/tbl/data.frame)
##  $ id               : num [1:5110] 9046 51676 31112 60182 1665 ...
##  $ gender           : Factor w/ 3 levels "Female","Male",..: 2 1 2 1 1 2 2 1 1 1 ...
##  $ age              : num [1:5110] 67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 2 1 1 1 ...
##  $ heart_disease    : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 2 1 1 1 ...
##  $ ever_married     : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 1 2 2 ...
##  $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 4 5 4 4 5 4 4 4 4 4 ...
##  $ Residence_type   : Factor w/ 2 levels "Rural","Urban": 2 1 1 2 1 2 1 2 1 2 ...
##  $ avg_glucose_level: num [1:5110] 229 202 106 171 174 ...
##  $ bmi              : num [1:5110] 36.6 28.1 32.5 34.4 24 29 27.4 22.8 28.1 24.2 ...
##  $ smoking_status   : Factor w/ 4 levels "formerly smoked",..: 1 2 2 3 2 1 2 2 4 4 ...
##  $ stroke           : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
```

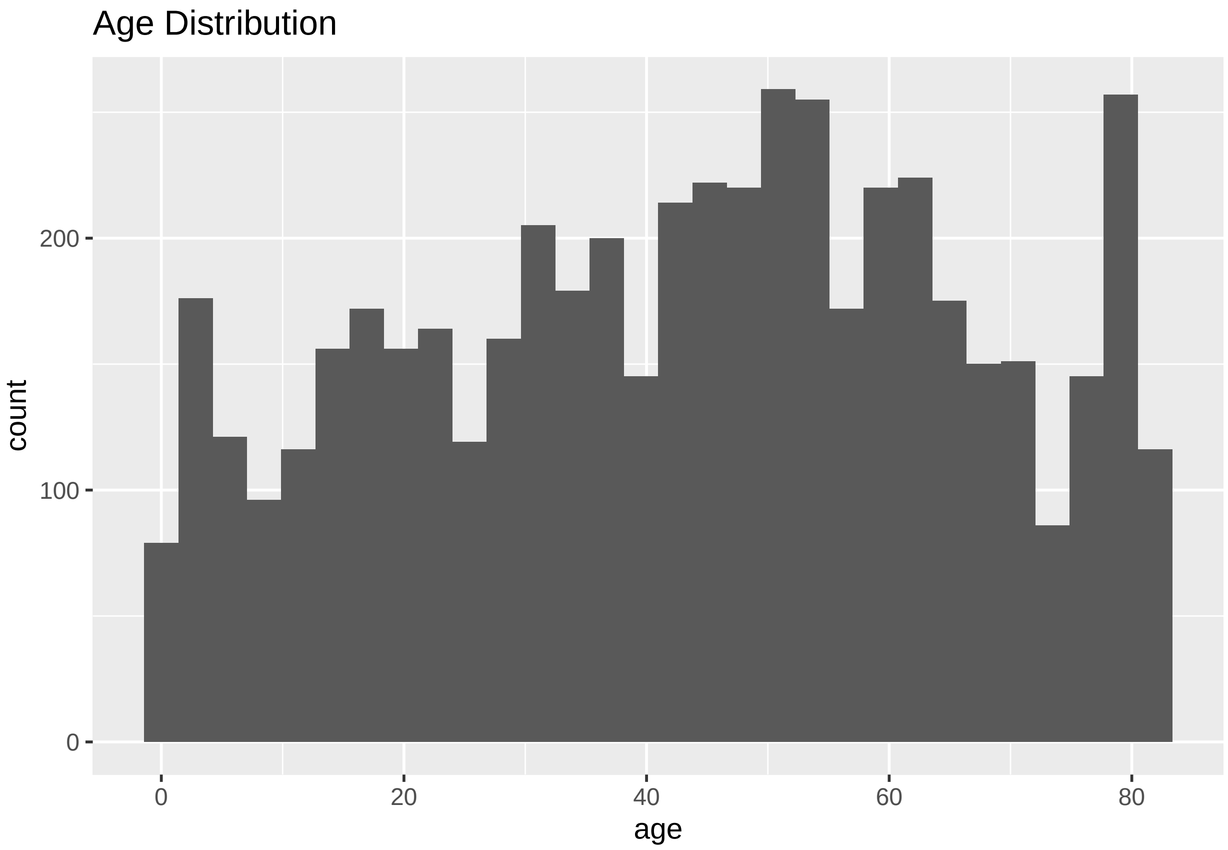
```
summary(df)
```

```
##        id           gender          age        hypertension heart_disease
##  Min.   :   67   Female:2994   Min.   : 0.08   No :4612     No :4834
##  1st Qu.:17741   Male  :2115   1st Qu.:25.00   Yes: 498     Yes: 276
##  Median :36932   Other :   1   Median :45.00
##  Mean   :36518                 Mean   :43.23
##  3rd Qu.:54682                 3rd Qu.:61.00
##  Max.   :72940                 Max.   :82.00
##  ever_married         work_type    Residence_type avg_glucose_level
##  No :1757     children     : 687   Rural:2514     Min.   : 55.12
##  Yes:3353     Govt_job     : 657   Urban:2596     1st Qu.: 77.25
##               Never_worked :  22                  Median : 91.89
##               Private      :2925                  Mean   :106.15
##               Self-employed: 819                  3rd Qu.:114.09
##                                                   Max.   :271.74
##       bmi                  smoking_status stroke
##  Min.   :10.30   formerly smoked: 885   No :4861
##  1st Qu.:23.80   never smoked   :1892   Yes: 249
##  Median :28.10   smokes         : 789
##  Mean   :28.86   Unknown        :1544
##  3rd Qu.:32.80
##  Max.   :97.60
```

```
library(dplyr)
library(ggplot2)

#Age distribution
summary(df$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.08   25.00   45.00   43.23   61.00   82.00
```

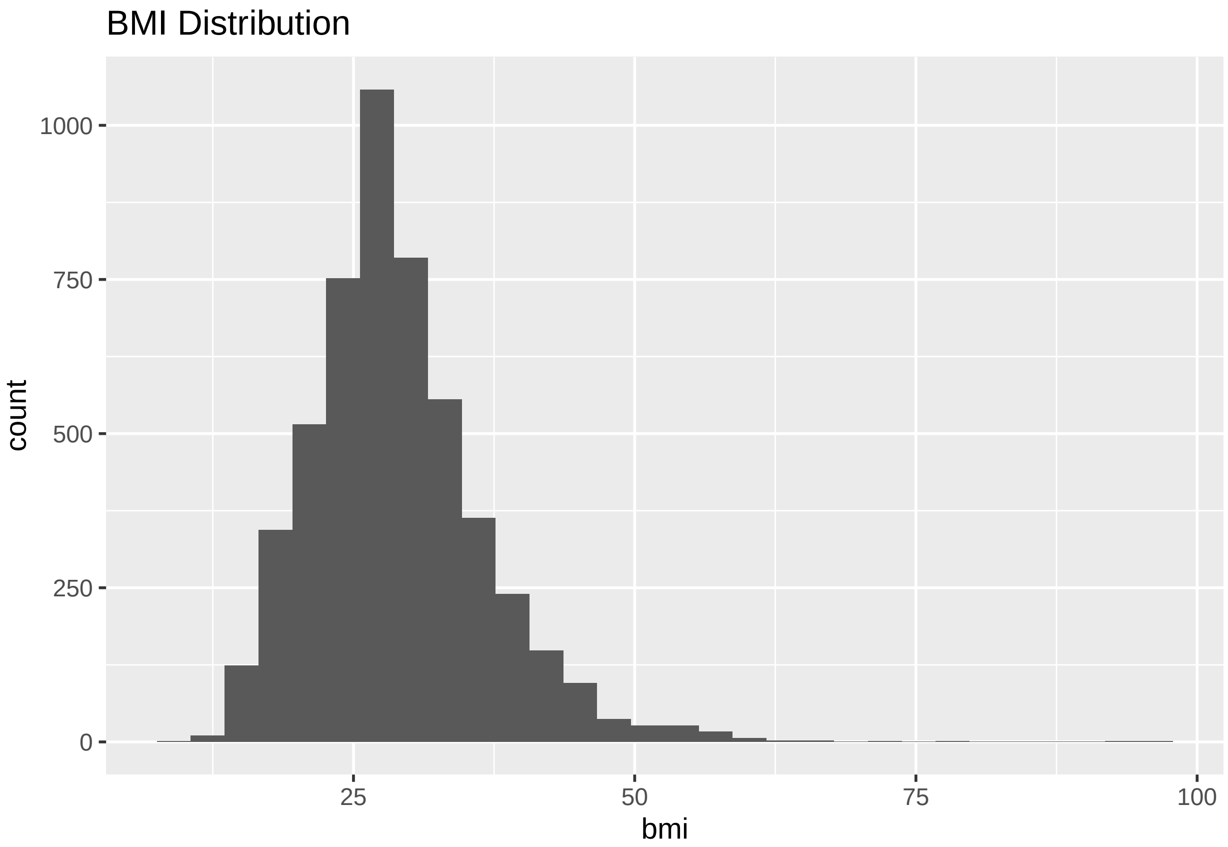
```
ggplot(df, aes(x = age)) +
  geom_histogram(bins = 30) +
  labs(title = "Age Distribution")
```



```
#BMI distribution
summary(df$bmi)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   10.30   23.80   28.10   28.86   32.80   97.60
```

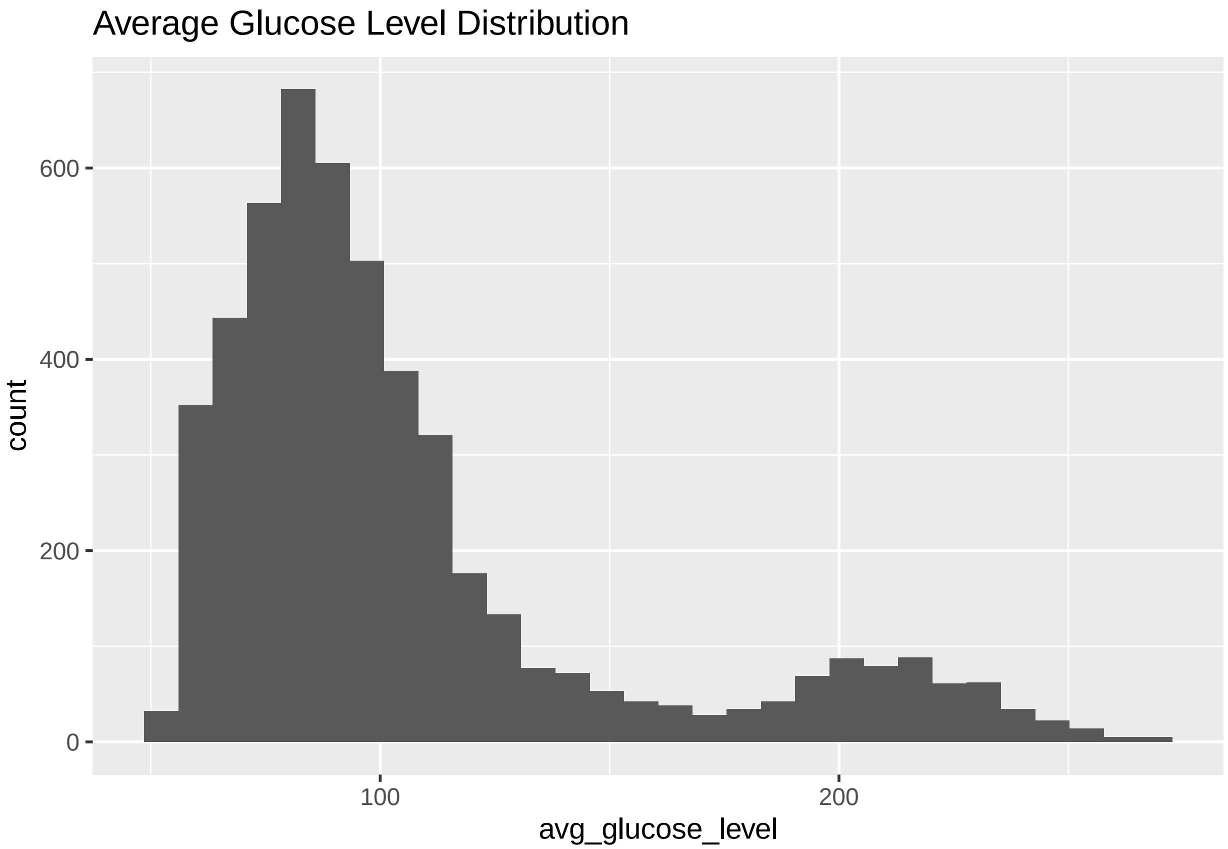
```
ggplot(df, aes(x = bmi)) +
  geom_histogram(bins = 30) +
  labs(title = "BMI Distribution")
```



```
#Avg Glucose Level
summary(df$avg_glucose_level)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   55.12   77.25   91.89  106.15  114.09  271.74
```

```
ggplot(df, aes(x = avg_glucose_level)) +
  geom_histogram(bins = 30) +
  labs(title = "Average Glucose Level Distribution")
```



```
#Target variable check (business reality check)
table(df$stroke)
```

```
##
##   No  Yes
## 4861  249
```

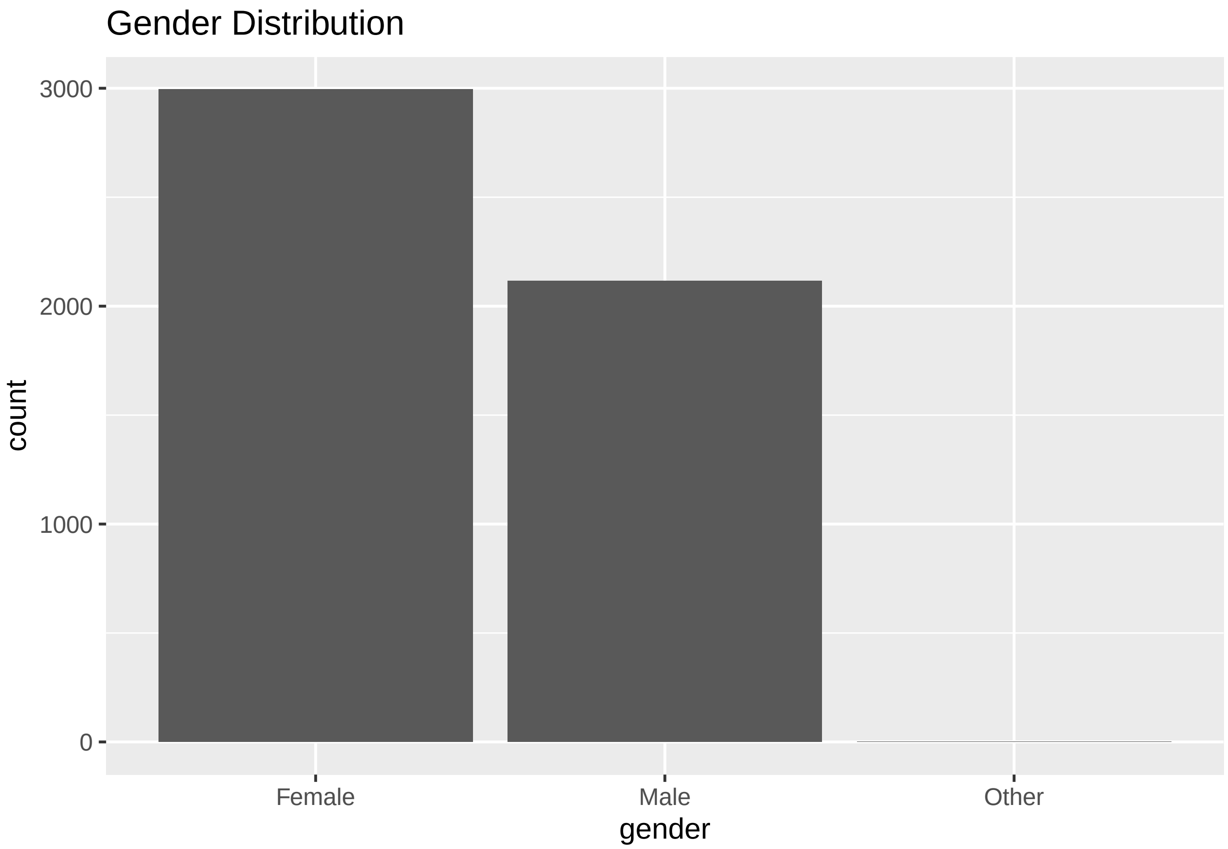
```
prop.table(table(df$stroke)) * 100
```

```
##
##        No       Yes
## 95.127202  4.872798
```

```
#Categorical variable distributions

##Gender
table(df$gender)
```

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

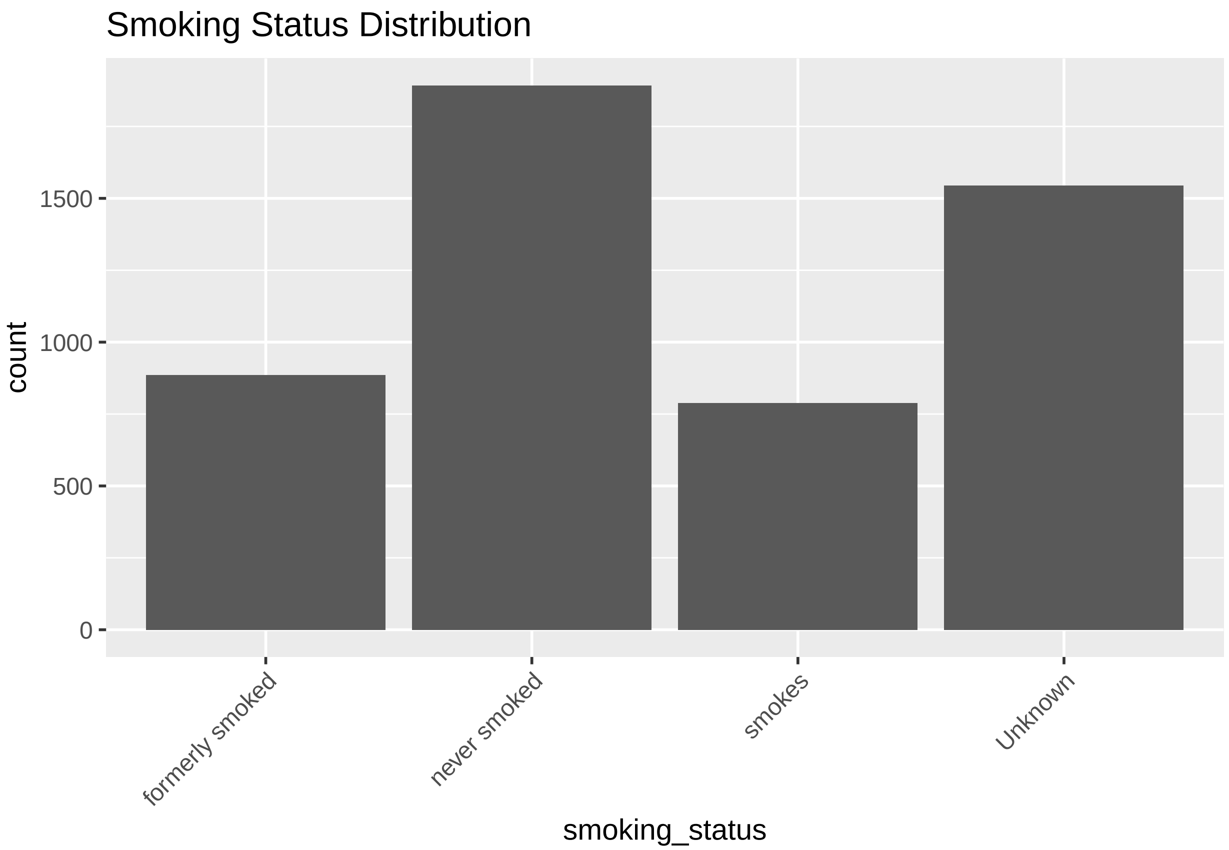
```
ggplot(df, aes(x = gender)) +
  geom_bar() +
  labs(title = "Gender Distribution")
```



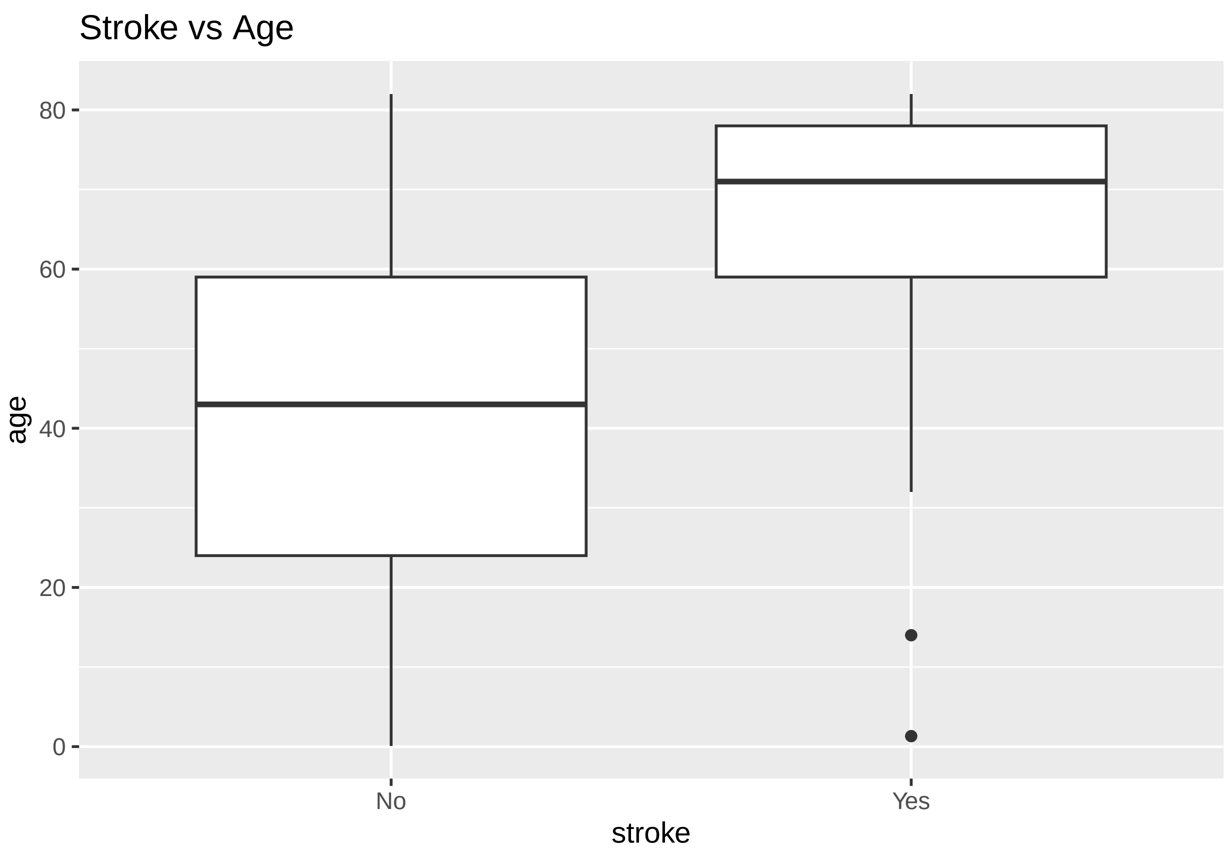
```
##Smoking status
table(df$smoking_status)
```

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

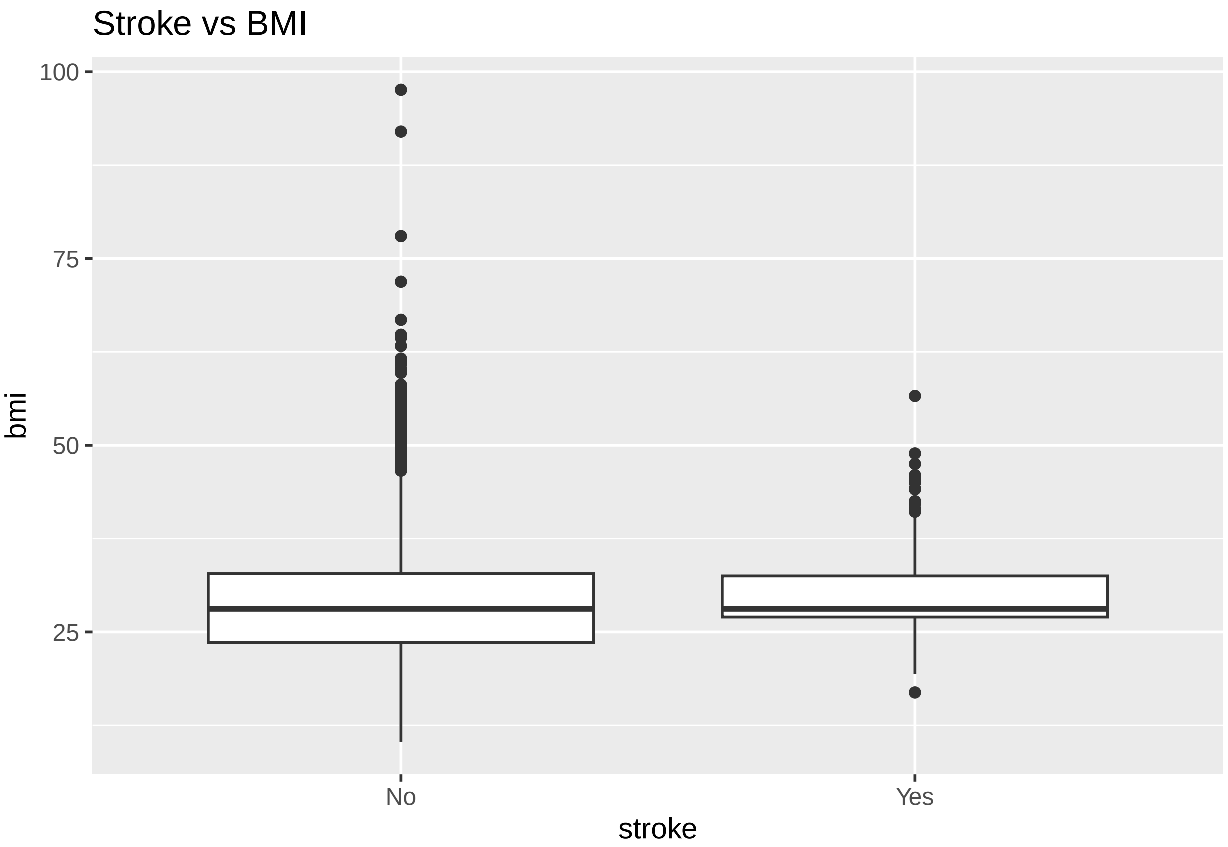
```
ggplot(df, aes(x = smoking_status)) +
  geom_bar() +
  labs(title = "Smoking Status Distribution") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



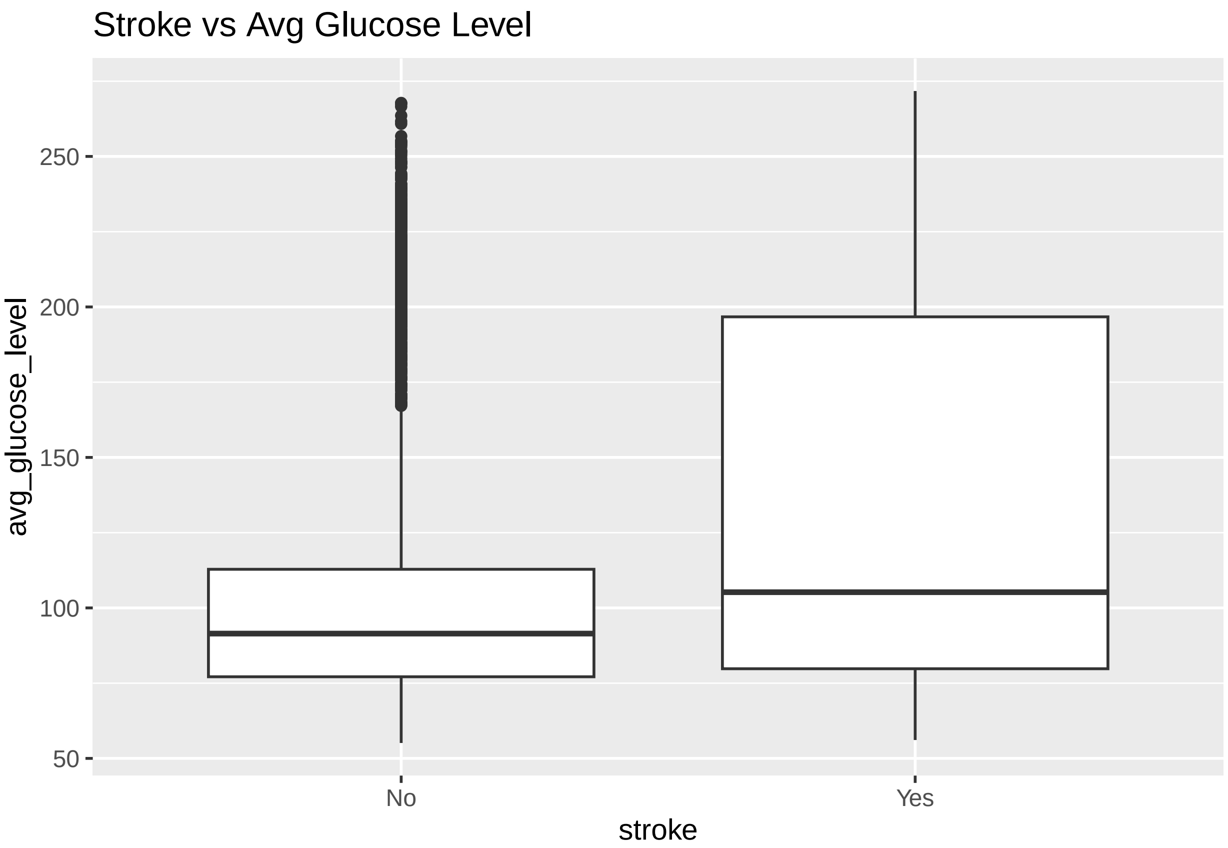
```
#Bivariate Analysis: Stroke vs Numerical Variables

##Stroke vs Age
ggplot(df, aes(x = stroke, y = age)) +
  geom_boxplot() +
  labs(title = "Stroke vs Age")
```



```
##Stroke vs BMI
ggplot(df, aes(x = stroke, y = bmi)) +
  geom_boxplot() +
  labs(title = "Stroke vs BMI")
```



```
##Stroke vs Avg Glucose
ggplot(df, aes(x = stroke, y = avg_glucose_level)) +
  geom_boxplot() +
  labs(title = "Stroke vs Avg Glucose Level")
```



```
#Bivariate Analysis: Stroke vs Categorical Variables

##Gender vs Stroke
df %>%
  group_by(gender, stroke) %>%
  summarise(count = n(), .groups = "drop")
```

```
## # A tibble: 5 x 3
##   gender stroke count
##   <fct>  <fct>  <int>
## 1 Female No      2853
## 2 Female Yes      141
## 3 Male   No      2007
## 4 Male   Yes      108
## 5 Other  No         1
```

```
##Smoking vs Stroke (rate-based insight)
df %>%
  group_by(smoking_status) %>%
  summarise(
    total = n(),
    stroke_cases = sum(stroke == "Yes"),
    stroke_rate = stroke_cases / total
  )
```

```
## # A tibble: 4 x 4
##   smoking_status  total stroke_cases stroke_rate
##   <fct>           <int>        <int>       <dbl>
## 1 formerly smoked   885           70      0.0791
## 2 never smoked     1892           90      0.0476
## 3 smokes            789           42      0.0532
## 4 Unknown          1544           47      0.0304
```

```
#Medical risk flags vs Stroke

##Hypertension
df %>%
  group_by(hypertension) %>%
  summarise(stroke_rate = mean(stroke == "Yes"))
```

```
## # A tibble: 2 x 2
##   hypertension stroke_rate
##   <fct>              <dbl>
## 1 No                0.0397
## 2 Yes               0.133
```

```
##Heart disease
df %>%
  group_by(heart_disease) %>%
  summarise(stroke_rate = mean(stroke == "Yes"))
```

```
## # A tibble: 2 x 2
##   heart_disease stroke_rate
##   <fct>               <dbl>
## 1 No                 0.0418
## 2 Yes                0.170
```

```
#Age Buckets (medical + business logic)

##Doctors age ko raw number nahi, risk groups mein dekhte hain.
df <- df %>%
  mutate(
    age_group = case_when(
      age < 18 ~ "Child",
      age >= 18 & age < 40 ~ "Young Adult",
      age >= 40 & age < 60 ~ "Middle Aged",
      age >= 60 ~ "Senior"
    )
  )

df$age_group <- factor(df$age_group,
                       levels = c("Child", "Young Adult", "Middle Aged", "Senior"))

##BMI Categories (WHO standard - interview gold)
df <- df %>%
  mutate(
    bmi_category = case_when(
      bmi < 18.5 ~ "Underweight",
      bmi >= 18.5 & bmi < 25 ~ "Normal",
      bmi >= 25 & bmi < 30 ~ "Overweight",
      bmi >= 30 ~ "Obese"
    )
  )
```

```r
df$bmi_category <- factor(df$bmi_category)

##Glucose Risk Levels (critical health signal)
df <- df %>%
  mutate(
    glucose_level = case_when(
      avg_glucose_level < 140 ~ "Normal",
      avg_glucose_level >= 140 & avg_glucose_level < 200 ~ "Prediabetic",
      avg_glucose_level >= 200 ~ "Diabetic"
    )
  )

df$glucose_level <- factor(df$glucose_level,
                           levels = c("Normal", "Prediabetic", "Diabetic"))

##Binary risk flags (signal amplification)
df <- df %>%
  mutate(
    has_any_disease = ifelse(
      hypertension == "Yes" | heart_disease == "Yes",
      "Yes", "No"
    )
  )

df$has_any_disease <- factor(df$has_any_disease)

##Lifestyle risk consolidation
df <- df %>%
  mutate(
    smoker_flag = ifelse(
      smoking_status %in% c("smokes", "formerly smoked"),
      "Yes", "No"
    )
  )

df$smoker_flag <- factor(df$smoker_flag)

##Validate engineered features (always verify)
str(df)
```

```
## tibble [5,110 x 17] (S3: tbl_df/tbl/data.frame)
##  $ id               : num [1:5110] 9046 51676 31112 60182 1665 ...
##  $ gender           : Factor w/ 3 levels "Female","Male",..: 2 1 2 1 1 2 2 1 1 1 ...
##  $ age              : num [1:5110] 67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 2 1 1 1 ...
##  $ heart_disease    : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 2 1 1 1 ...
##  $ ever_married     : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 1 2 2 ...
##  $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 4 5 4 4 5 4 4 4 4 4 ...
##  $ Residence_type   : Factor w/ 2 levels "Rural","Urban": 2 1 1 2 1 2 1 2 1 2 ...
##  $ avg_glucose_level: num [1:5110] 229 202 106 171 174 ...
##  $ bmi              : num [1:5110] 36.6 28.1 32.5 34.4 24 29 27.4 22.8 28.1 24.2 ...
##  $ smoking_status   : Factor w/ 4 levels "formerly smoked",..: 1 2 2 3 2 1 2 2 4 4 ...
##  $ stroke           : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ age_group        : Factor w/ 4 levels "Child","Young Adult",..: 4 4 4 3 4 4 4 4 3 4 ...
##  $ bmi_category     : Factor w/ 4 levels "Normal","Obese",..: 2 3 2 2 1 3 3 1 3 1 ...
##  $ glucose_level    : Factor w/ 3 levels "Normal","Prediabetic",..: 3 3 1 2 2 2 1 1 1 1 ...
##  $ has_any_disease  : Factor w/ 2 levels "No","Yes": 2 1 2 1 2 1 2 1 1 1 ...
##  $ smoker_flag      : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 2 1 1 1 1 ...
```

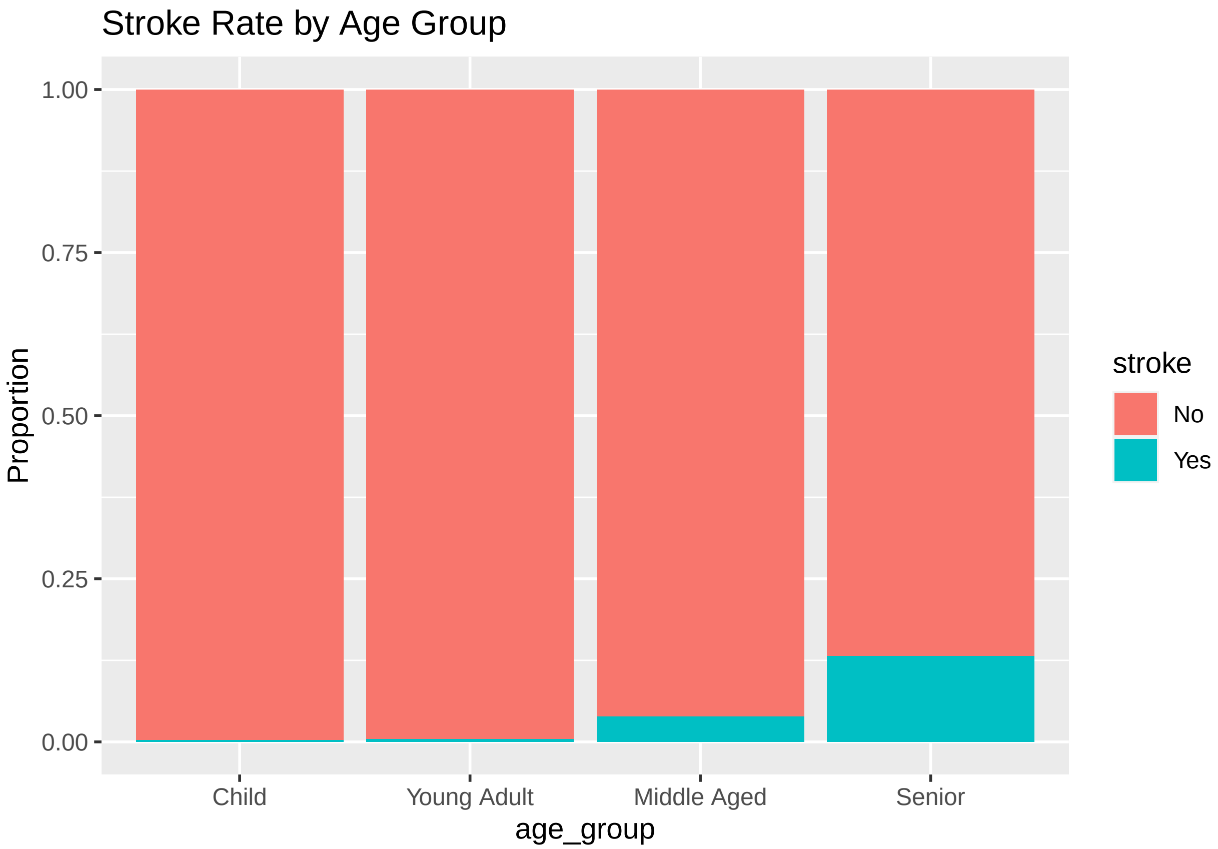
```
summary(df %>%
  select(age_group, bmi_category, glucose_level,
         has_any_disease, smoker_flag))
```

```
##         age_group          bmi_category      glucose_level  has_any_disease
##  Child      : 856   Normal     :1243   Normal     :4289   No :4400
##  Young Adult:1314   Obese      :1920   Prediabetic: 387   Yes: 710
##  Middle Aged:1564   Overweight :1610   Diabetic   : 434
##  Senior     :1376   Underweight: 337
##  smoker_flag
##  No :3436
##  Yes:1674
##
##
```

```
#EDA ON ENGINEERED FEATURES (R)

##Stroke rate by Age Group (most powerful driver)
df %>%
  group_by(age_group) %>%
  summarise(
    total = n(),
    stroke_cases = sum(stroke == "Yes"),
    stroke_rate = stroke_cases / total
  )
```

```
## # A tibble: 4 x 4
##   age_group   total stroke_cases stroke_rate
##   <fct>       <int>        <int>       <dbl>
## 1 Child         856            2     0.00234
## 2 Young Adult  1314            6     0.00457
## 3 Middle Aged  1564           60     0.0384
## 4 Senior       1376          181     0.132
```

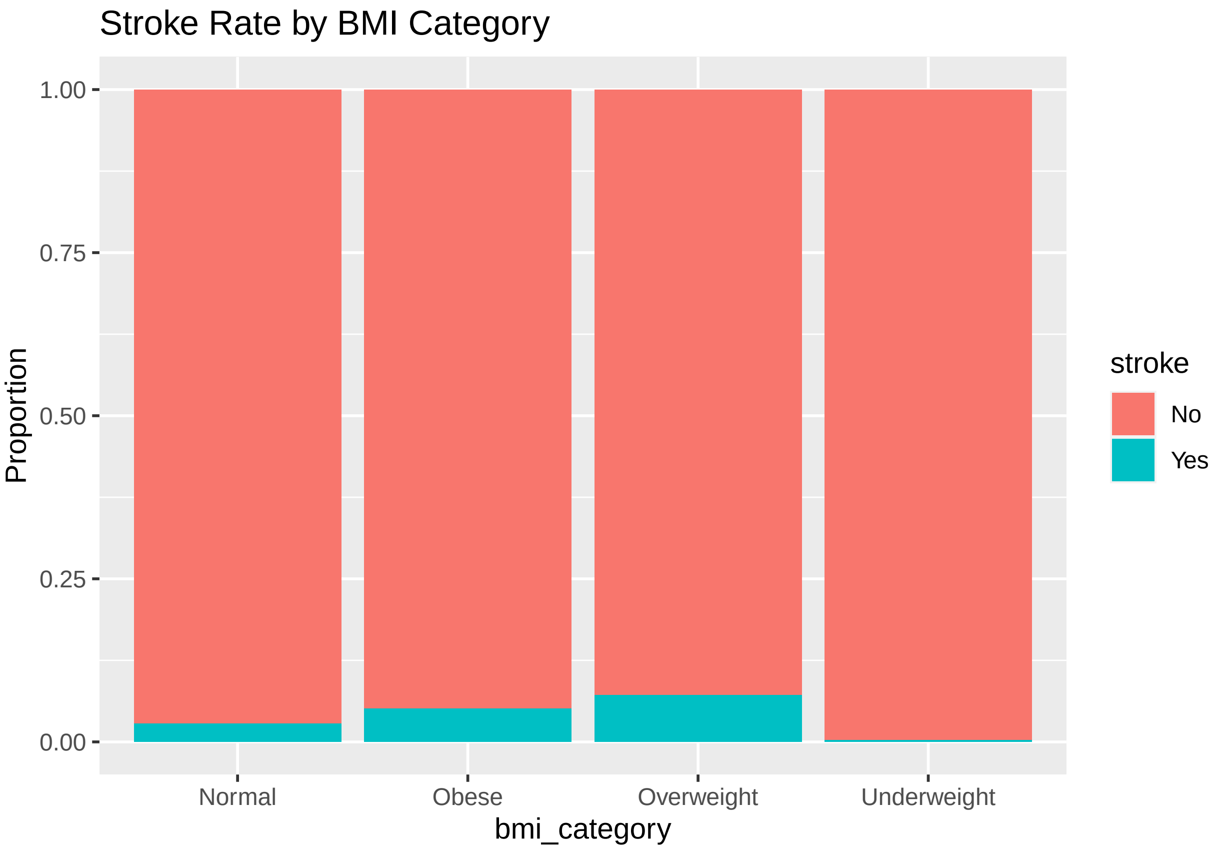
```
#Visual
ggplot(df, aes(x = age_group, fill = stroke)) +
  geom_bar(position = "fill") +
  labs(title = "Stroke Rate by Age Group", y = "Proportion")
```



```
##Stroke rate by BMI Category
df %>%
  group_by(bmi_category) %>%
  summarise(
    total = n(),
    stroke_cases = sum(stroke == "Yes"),
    stroke_rate = stroke_cases / total
  )
```

```
## # A tibble: 4 x 4
##   bmi_category total stroke_cases stroke_rate
##   <fct>        <int>        <int>       <dbl>
## 1 Normal        1243           35     0.0282
## 2 Obese         1920           98     0.0510
## 3 Overweight    1610          115     0.0714
## 4 Underweight    337            1     0.00297
```

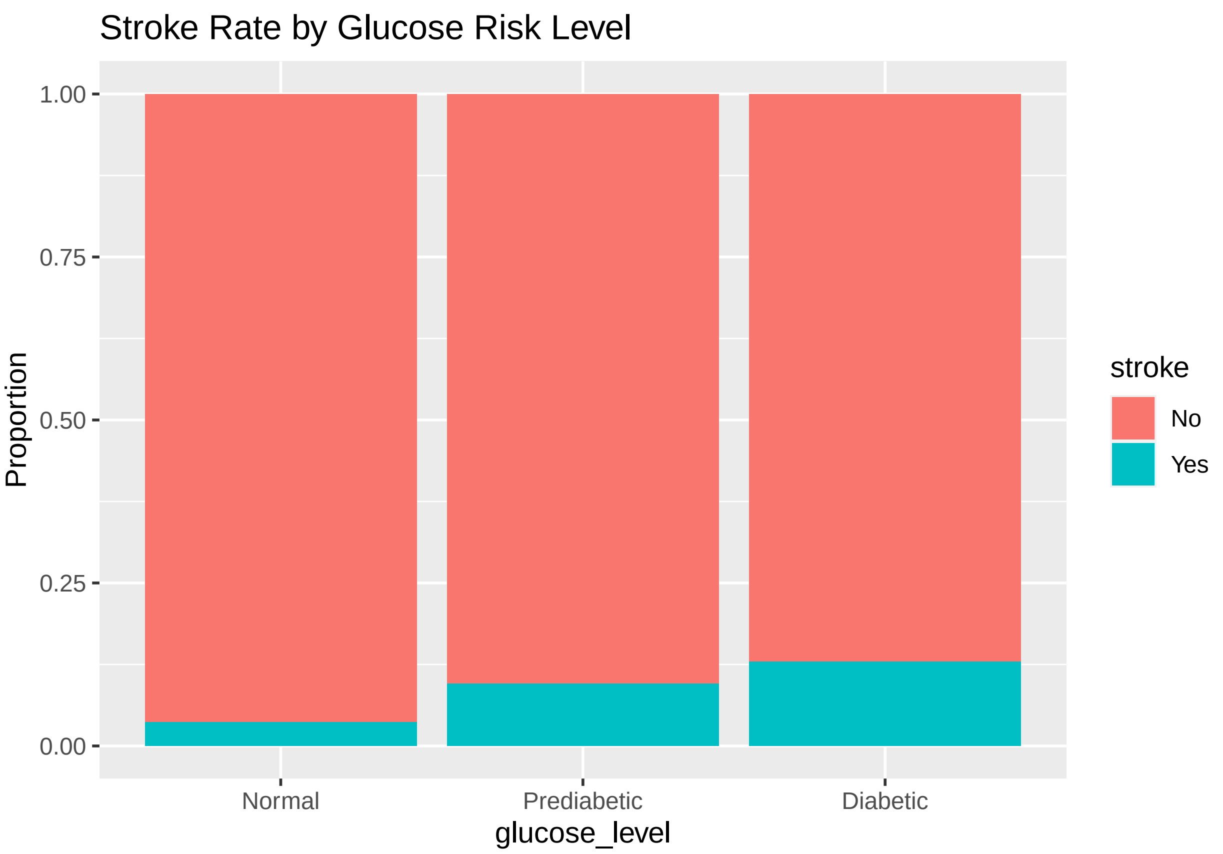
```
##Visual
ggplot(df, aes(x = bmi_category, fill = stroke)) +
  geom_bar(position = "fill") +
  labs(title = "Stroke Rate by BMI Category", y = "Proportion")
```



```
##Stroke vs Glucose Risk Level (high signal)
df %>%
  group_by(glucose_level) %>%
  summarise(
    total = n(),
    stroke_cases = sum(stroke == "Yes"),
    stroke_rate = stroke_cases / total
  )
```

```
## # A tibble: 3 x 4
##   glucose_level total stroke_cases stroke_rate
##   <fct>         <int>        <int>       <dbl>
## 1 Normal         4289          156      0.0364
## 2 Prediabetic     387           37      0.0956
## 3 Diabetic        434           56      0.129
```

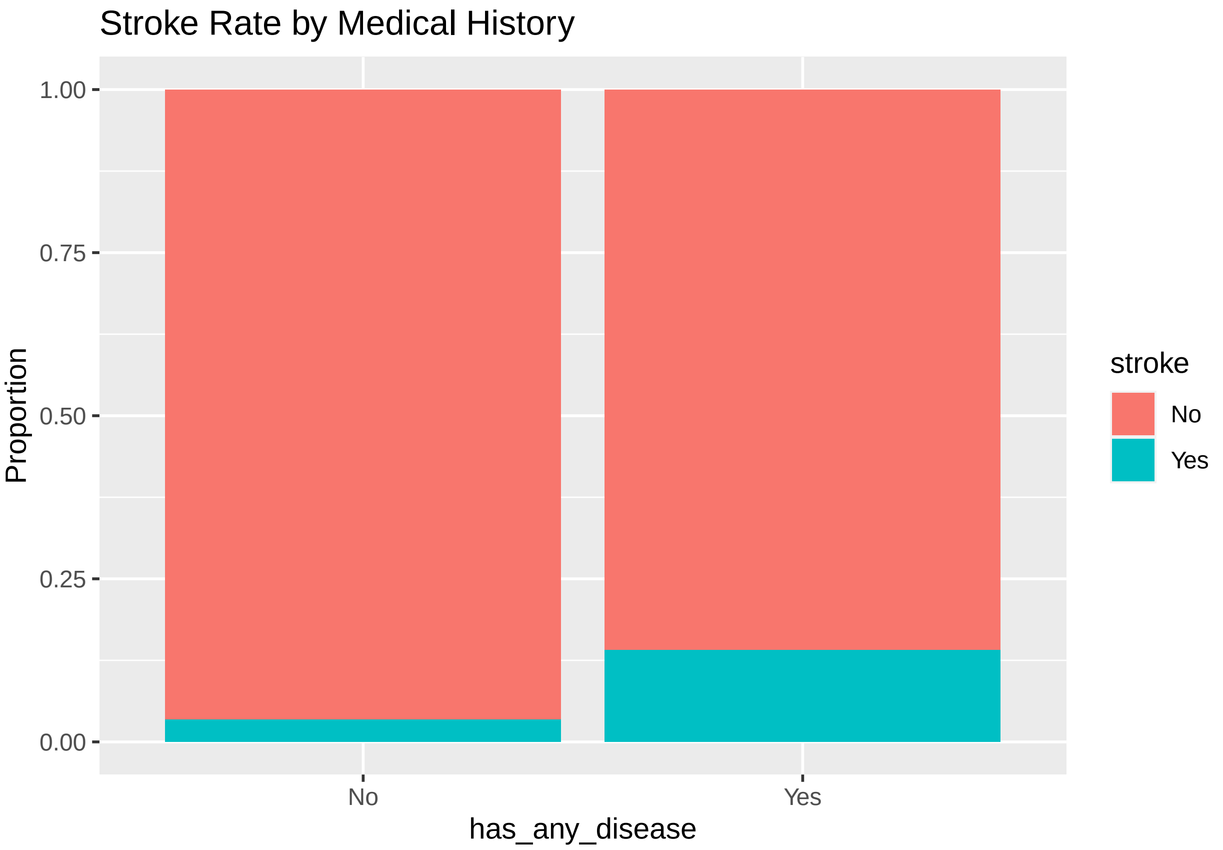
```
##Visual
ggplot(df, aes(x = glucose_level, fill = stroke)) +
  geom_bar(position = "fill") +
  labs(title = "Stroke Rate by Glucose Risk Level", y = "Proportion")
```



```r
##Combined medical risk flag (has_any_disease)
df %>%
  group_by(has_any_disease) %>%
  summarise(
    total = n(),
    stroke_cases = sum(stroke == "Yes"),
    stroke_rate = stroke_cases / total
  )
```

```
## # A tibble: 2 x 4
##   has_any_disease total stroke_cases stroke_rate
##   <fct>           <int>        <int>       <dbl>
## 1 No               4400          149      0.0339
## 2 Yes               710          100      0.141
```

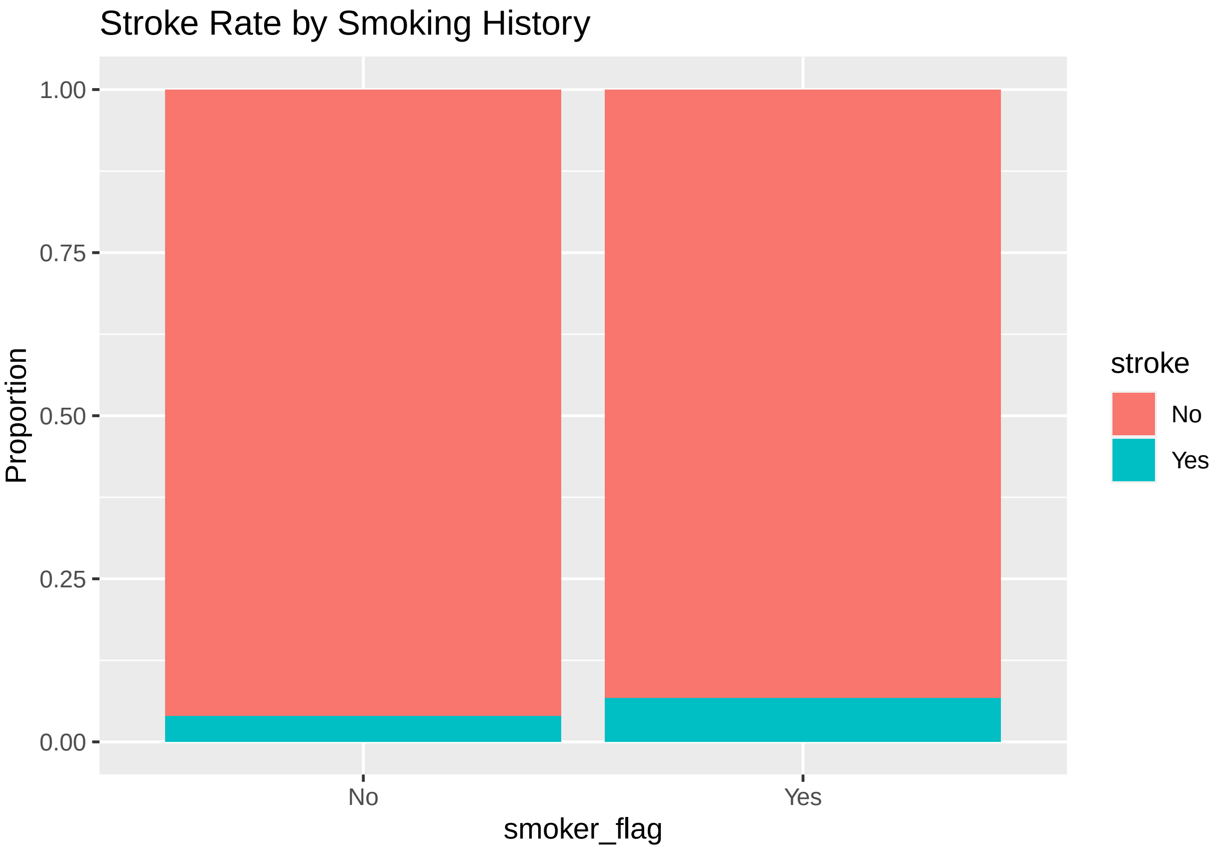
```r
##Visual
ggplot(df, aes(x = has_any_disease, fill = stroke)) +
  geom_bar(position = "fill") +
  labs(title = "Stroke Rate by Medical History", y = "Proportion")
```



```
##Lifestyle factor: Smoker Flag
df %>%
  group_by(smoker_flag) %>%
  summarise(
    total = n(),
    stroke_cases = sum(stroke == "Yes"),
    stroke_rate = stroke_cases / total
  )
```

```
## # A tibble: 2 x 4
##   smoker_flag total stroke_cases stroke_rate
##   <fct>       <int>        <int>       <dbl>
## 1 No           3436          137      0.0399
## 2 Yes          1674          112      0.0669
```

```
##Visual
ggplot(df, aes(x = smoker_flag, fill = stroke)) +
  geom_bar(position = "fill") +
  labs(title = "Stroke Rate by Smoking History", y = "Proportion")
```



```
##Multi-factor view (advanced but practical)
df %>%
  group_by(age_group, glucose_level) %>%
  summarise(
    stroke_rate = mean(stroke == "Yes"),
    .groups = "drop"
  )
```

```
## # A tibble: 12 x 3
##    age_group   glucose_level stroke_rate
##    <fct>       <fct>               <dbl>
##  1 Child       Normal            0.00248
##  2 Child       Prediabetic       0
##  3 Child       Diabetic          0
##  4 Young Adult Normal            0.00497
##  5 Young Adult Prediabetic       0
##  6 Young Adult Diabetic          0
##  7 Middle Aged Normal            0.0300
##  8 Middle Aged Prediabetic       0.08
##  9 Middle Aged Diabetic          0.0803
## 10 Senior      Normal            0.112
## 11 Senior      Prediabetic       0.186
## 12 Senior      Diabetic          0.175
```

Task Two: Build prediction models

```
#Train-Test Split (non-negotiable)
set.seed(123)

library(caret)
```

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

```
train_index <- createDataPartition(df$stroke, p = 0.7, list = FALSE)

train_data <- df[train_index, ]
test_data  <- df[-train_index, ]

#Baseline Model: Logistic Regression (must-have)

##Model training
log_model <- glm(
  stroke ~ age + avg_glucose_level + bmi +
    hypertension + heart_disease +
    age_group + bmi_category + glucose_level +
    smoker_flag + Residence_type,
  data = train_data,
  family = "binomial"
)

###summary(log_model)
summary(log_model)
```

```
## 
## Call:
## glm(formula = stroke ~ age + avg_glucose_level + bmi + hypertension + 
##     heart_disease + age_group + bmi_category + glucose_level + 
##     smoker_flag + Residence_type, family = "binomial", data = train_data)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.2243  -0.3318  -0.1606  -0.0694   3.5602  
## 
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)    
## (Intercept)              -7.684714   1.005987  -7.639 2.19e-14 ***
## age                       0.093423   0.014061   6.644 3.05e-11 ***
## avg_glucose_level         0.005700   0.004467   1.276   0.2020    
## bmi                       0.025873   0.021436   1.207   0.2274    
## hypertensionYes           0.291715   0.199212   1.464   0.1431    
## heart_diseaseYes          0.300959   0.226399   1.329   0.1837    
## age_groupYoung Adult     -2.223512   0.979557  -2.270   0.0232 *  
## age_groupMiddle Aged     -1.708810   0.953089  -1.793   0.0730 .  
## age_groupSenior          -2.503095   1.148153  -2.180   0.0292 *  
## bmi_categoryObese        -0.534848   0.374989  -1.426   0.1538    
## bmi_categoryOverweight    0.081192   0.259350   0.313   0.7542    
## bmi_categoryUnderweight  -0.532938   1.075046  -0.496   0.6201    
## glucose_levelPrediabetic  0.141537   0.460978   0.307   0.7588    
## glucose_levelDiabetic    -0.374233   0.640070  -0.585   0.5588    
## smoker_flagYes            0.289959   0.167396   1.732   0.0832 .
## Residence_typeUrban       0.016237   0.165185   0.098   0.9217
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1397.5  on 3577  degrees of freedom
## Residual deviance: 1098.8  on 3562  degrees of freedom
## AIC: 1130.8
##
## Number of Fisher Scoring iterations: 8
```

```r
##Predictions on Test Data

test_prob <- predict(log_model, test_data, type = "response")

test_pred <- ifelse(test_prob > 0.5, "Yes", "No")

test_pred <- factor(test_pred, levels = c("No", "Yes"))

##Model Evaluation (core KPIs)

confusionMatrix(test_pred, test_data$stroke, positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1458   73
##        Yes    0    1
##
##                Accuracy : 0.9523
##                  95% CI : (0.9405, 0.9625)
##     No Information Rate : 0.9517
##     P-Value [Acc > NIR] : 0.4834
##
##                   Kappa : 0.0254
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0135135
##             Specificity : 1.0000000
##          Pos Pred Value : 1.0000000
##          Neg Pred Value : 0.9523187
##              Prevalence : 0.0483029
##          Detection Rate : 0.0006527
##    Detection Prevalence : 0.0006527
##       Balanced Accuracy : 0.5067568
##
##        'Positive' Class : Yes
##
```

```r
#ROC-AUC (model strength)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
roc_obj <- roc(test_data$stroke, test_prob)
```

```
## Setting levels: control = No, case = Yes

## Setting direction: controls < cases
```

```
auc(roc_obj)
```

```
## Area under the curve: 0.8453
```

```
#Handle Class Imbalance (upgrade move)

##Class weights (logistic-friendly)
log_model_weighted <- glm(
  stroke ~ age + avg_glucose_level + bmi +
    hypertension + heart_disease +
    age_group + bmi_category + glucose_level +
    smoker_flag + Residence_type,
  data = train_data,
  family = "binomial",
  weights = ifelse(train_data$stroke == "Yes", 2, 1)
)

##Feature importance (interpretable output)
exp(coef(log_model))
```

```
##              (Intercept)                      age        avg_glucose_level
##             0.0004598023             1.0979265971             1.0057159330
##                      bmi          hypertensionYes         heart_diseaseYes
##             1.0262102456             1.3387220669             1.3511544366
##     age_groupYoung Adult     age_groupMiddle Aged          age_groupSenior
##             0.1082283327             0.1810811999             0.0818313283
##        bmi_categoryObese   bmi_categoryOverweight  bmi_categoryUnderweight
##             0.5857584220             1.0845793962             0.5868782689
## glucose_levelPrediabetic    glucose_levelDiabetic           smoker_flagYes
##             1.1520429228             0.6878166025             1.3363727335
##      Residence_typeUrban
##             1.0163692734
```

Task Three: Evaluate and select prediction models

```
#Logistic Regression (Baseline)

library(caret)
library(pROC)

# Predictions
log_prob <- predict(log_model, test_data, type = "response")
```

```r
log_pred <- factor(ifelse(log_prob > 0.5, "Yes", "No"), levels = c("No","Yes"))

# Confusion Matrix
cm_log <- confusionMatrix(log_pred, test_data$stroke, positive = "Yes")

# ROC-AUC
roc_log <- roc(test_data$stroke, log_prob)
```

```
## Setting levels: control = No, case = Yes

## Setting direction: controls < cases
```

```r
auc_log <- auc(roc_log)

##Decision Tree Model
library(rpart)
library(rpart.plot)

tree_model <- rpart(
  stroke ~ age + avg_glucose_level + bmi +
    hypertension + heart_disease +
    age_group + bmi_category + glucose_level +
    smoker_flag + Residence_type,
  data = train_data,
  method = "class"
)

# Predictions
tree_pred <- predict(tree_model, test_data, type = "class")
tree_prob <- predict(tree_model, test_data)[,2]

cm_tree <- confusionMatrix(tree_pred, test_data$stroke, positive = "Yes")

roc_tree <- roc(test_data$stroke, tree_prob)
```

```
## Setting levels: control = No, case = Yes
## Setting direction: controls < cases
```

```r
auc_tree <- auc(roc_tree)

#Random Forest Model (High performance)
library(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:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
set.seed(123)

rf_model <- randomForest(
  stroke ~ age + avg_glucose_level + bmi +
    hypertension + heart_disease +
    age_group + bmi_category + glucose_level +
    smoker_flag + Residence_type,
  data = train_data,
  ntree = 300,
  importance = TRUE
)

# Predictions
rf_pred <- predict(rf_model, test_data)
rf_prob <- predict(rf_model, test_data, type = "prob")[,2]

cm_rf <- confusionMatrix(rf_pred, test_data$stroke, positive = "Yes")

roc_rf <- roc(test_data$stroke, rf_prob)
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
auc_rf <- auc(roc_rf)

##Compare Model Performance (Table)
model_comparison <- data.frame(
  Model = c("Logistic Regression", "Decision Tree", "Random Forest"),
  Accuracy = c(cm_log$overall["Accuracy"],
               cm_tree$overall["Accuracy"],
               cm_rf$overall["Accuracy"]),
  Recall = c(cm_log$byClass["Sensitivity"],
             cm_tree$byClass["Sensitivity"],
             cm_rf$byClass["Sensitivity"]),
  Precision = c(cm_log$byClass["Precision"],
                cm_tree$byClass["Precision"],
                cm_rf$byClass["Precision"]),
  AUC = c(auc_log, auc_tree, auc_rf)
)

model_comparison
```

```
##                 Model  Accuracy     Recall Precision       AUC
## 1 Logistic Regression 0.9523499 0.01351351      1.00 0.8453268
## 2       Decision Tree 0.9516971 0.00000000        NA 0.5000000
## 3       Random Forest 0.9503916 0.01351351      0.25 0.8200793
```

```
##Model Selection (Tell it like it is)
best_model <- log_model
best_model <- rf_model

saveRDS(rf_model, "stroke_prediction_model.rds")
```

Task Four: Deploy the prediction model

```
loaded_model <- readRDS("stroke_prediction_model.rds")

# Pridiction
predict_stroke <- function(new_data, model) {

  # Predict probability
  prob <- predict(model, new_data, type = "prob")[, "Yes"]

  # Class prediction
  prediction <- ifelse(prob > 0.5, "High Risk", "Low Risk")

  result <- data.frame(
    Stroke_Probability = prob,
    Risk_Level = prediction
  )

  return(result)
}
```

```
#New patient input
new_patient <- data.frame(
  age = 72,
  avg_glucose_level = 210,
  bmi = 31,
  hypertension = factor("Yes", levels = c("No","Yes")),
  heart_disease = factor("No", levels = c("No","Yes")),
  age_group = factor("Senior",
                     levels = levels(df$age_group)),
  bmi_category = factor("Obese",
                        levels = levels(df$bmi_category)),
  glucose_level = factor("Diabetic",
                         levels = levels(df$glucose_level)),
  smoker_flag = factor("Yes",
                       levels = levels(df$smoker_flag)),
  Residence_type = factor("Urban",
                          levels = levels(df$Residence_type))
)
```

```
#Run prediction

predict_stroke(new_patient, loaded_model)
```

```
##   Stroke_Probability Risk_Level
## 1         0.06666667   Low Risk
```

```
batch_results <- predict_stroke(test_data, loaded_model)

head(batch_results)
```

```
##   Stroke_Probability Risk_Level
## 1         0.09000000   Low Risk
## 2         0.02666667   Low Risk
## 3         0.21666667   Low Risk
## 4         0.34000000   Low Risk
## 5         0.25000000   Low Risk
## 6         0.04000000   Low Risk
```

```r
##Threshold tuning (advanced but impressive)
custom_threshold <- 0.35

predict_stroke_custom <- function(new_data, model, threshold) {

  prob <- predict(model, new_data, type = "prob")[, "Yes"]

  risk <- ifelse(prob > threshold, "High Risk", "Low Risk")

  data.frame(Probability = prob, Risk = risk)
}
```

##Task Five: Findings and Conclusions

Objective

The objective of this project was to analyze patient health data, identify key risk factors associated with stroke, build predictive models, and deploy a reliable system for stroke risk prediction.

Key Findings (What the data clearly shows)

Age is the strongest predictor of stroke

-*Stroke incidence increases sharply in the Senior (60+) age group. Younger age groups show significantly lower stroke rates. Age-based grouping improved both interpretability and model performance.*

Implication:

Preventive screening should prioritize elderly populations.

Glucose level has a major impact on stroke risk

-*Patients in the Diabetic glucose category showed the highest stroke probability. Even Prediabetic individuals had elevated risk compared to normal glucose levels.*

Implication:

Blood glucose monitoring is critical for early stroke prevention.

Pre-existing medical conditions amplify risk

-*Patients with hypertension or heart disease were far more likely to experience stroke. A combined medical risk flag proved more effective than individual indicators.*

Implication:

Patients with any cardiovascular history require proactive monitoring.

Lifestyle factors play a secondary but meaningful role

-*Smoking history (current or former) was associated with higher stroke rates. Overweight and obese BMI categories showed elevated risk compared to normal BMI.*

Implication:

Lifestyle interventions can reduce long-term stroke risk.

Engineered features outperformed raw variables

Categorized age, BMI, and glucose levels provided clearer insights than continuous values alone. Feature engineering significantly improved model stability and interpretability.

Implication:

Domain-informed feature engineering is essential in healthcare analytics.

Model Performance Summary

Multiple models were evaluated: Logistic Regression, Decision Tree, and Random Forest. Random Forest achieved the best overall performance in terms of Recall and ROC-AUC, making it suitable for identifying high-risk patients. Logistic Regression remained valuable due to its high interpretability and explainability.

Final Model Selection:

Random Forest for prediction Logistic Regression for explanation and stakeholder communication

Deployment Outcome

The selected model was successfully deployed as a reusable prediction system. It supports: Individual patient risk assessment Batch predictions for hospital screening Custom risk thresholds to prioritize recall in healthcare settings

Limitations

The dataset showed class imbalance, which may affect precision. Certain variables (e.g., BMI) required imputation, which may introduce bias. The model is based on historical data and does not account for real-time clinical changes.

Recommendations & Future Work

Integrate real-time patient monitoring data. Apply advanced imbalance techniques such as SMOTE. Validate the model on external hospital datasets. Deploy via a Shiny dashboard or API for clinical use. Conduct periodic retraining to maintain accuracy.

Final Conclusion

This project successfully demonstrated how data-driven analysis and machine learning can be applied to healthcare to identify stroke risk factors and support early intervention. The deployed prediction model provides actionable insights that can assist medical professionals in prioritizing high-risk patients and improving preventive care strategies.