Stroke Data Analysis Documentation

Documented By: Nush Ojha

Date: January 9, 2026

Executive Summary

This documentation presents a comprehensive workflow for analyzing stroke-related healthcare data. The study covers data cleaning, exploratory data analysis (EDA), predictive modeling using logistic regression and random forest classifiers, and the creation of an integrated visual dashboard. The main objective is to identify patterns, risk factors, and predictive capabilities to assess stroke risk within the population.

Key highlights include:

- Robust data cleaning and feature engineering to handle missing values and outliers.
- Statistical summaries and visualizations revealing demographic and clinical distributions.
- Predictive modeling with logistic regression and random forest, including class imbalance handling with SMOTE.
- ROC curve and AUC evaluation for model performance.
- A final dashboard combining prevalence, risk factors, and predictive confidence in a single interface.

The documentation demonstrates an end-to-end data science workflow suitable for clinical and public health applications.

Data Source

The dataset used in this project was obtained from Kaggle:

- **Dataset Name:** Stroke Prediction Dataset
- **Source:** Kaggle
- **URL:** https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset
- **Description:** The dataset contains demographic and clinical information for patients, including age, gender, BMI, smoking status, glucose levels, and stroke outcome (0 = No Stroke, 1 = Stroke).

Contents

Chapter 1

Data Cleaning

1.1 Library Installation and Loading

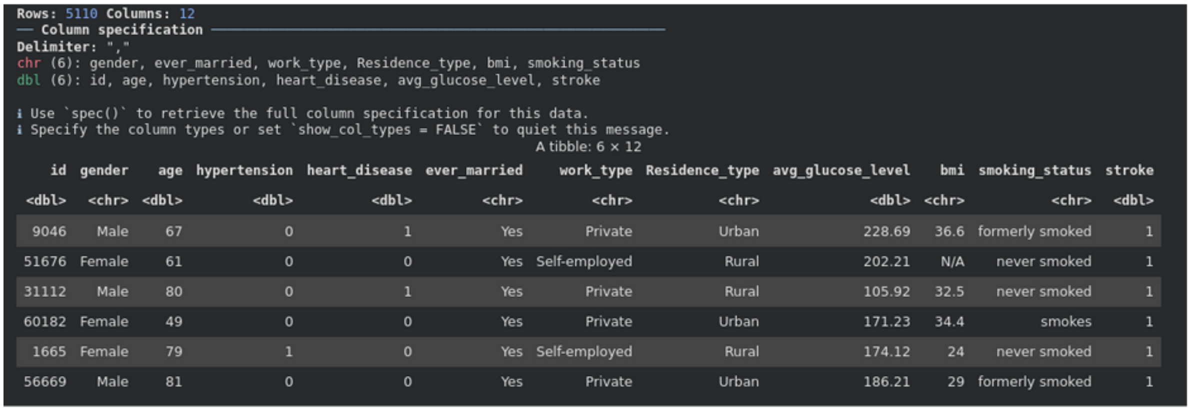
```
install.packages("tidyverse")
install.packages("skimr")
install.packages("janitor")

library(tidyverse) # For data wrangling and visualization
library(skimr) # For quick data overview
library(janitor)  # For cleaning column names
```

—

1.2 Data Loading

```
stroke <- read_csv("healthcare-dataset-stroke-data.csv")
head(stroke)
```

Output



```
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.
```

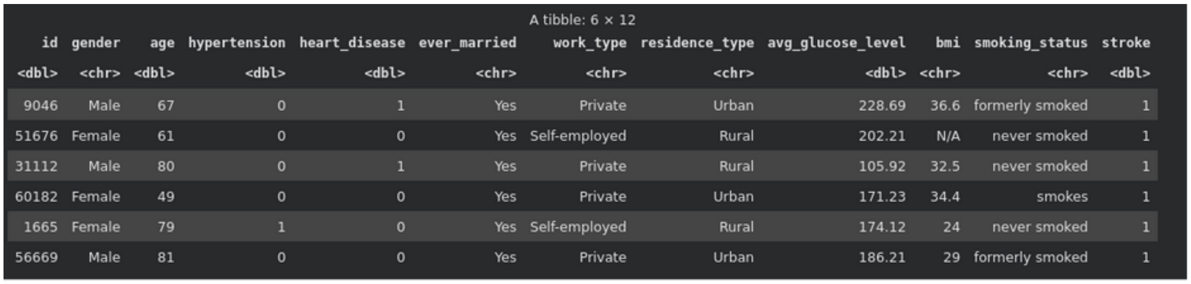
```
                                              A tibble: 6 × 12
   id  gender  age  hypertension  heart_disease  ever_married     work_type  Residence_type  avg_glucose_level   bmi  smoking_status  stroke
<dbl>   <chr> <dbl>        <dbl>          <dbl>         <chr>         <chr>           <chr>              <dbl> <chr>           <chr>   <dbl>
 9046    Male   67             0              1           Yes       Private           Urban             228.69  36.6 formerly smoked       1
51676  Female   61             0              0           Yes Self-employed           Rural             202.21   N/A    never smoked       1
31112    Male   80             0              1           Yes       Private           Rural             105.92  32.5    never smoked       1
60182  Female   49             0              0           Yes       Private           Urban             171.23  34.4          smokes       1
 1665  Female   79             1              0           Yes Self-employed           Rural             174.12    24    never smoked       1
56669    Male   81             0              0           Yes       Private           Urban             186.21    29 formerly smoked       1
```

—

1.3 Column Name Cleaning

```
stroke <- stroke %>% clean_names()
head(stroke)
```

Output



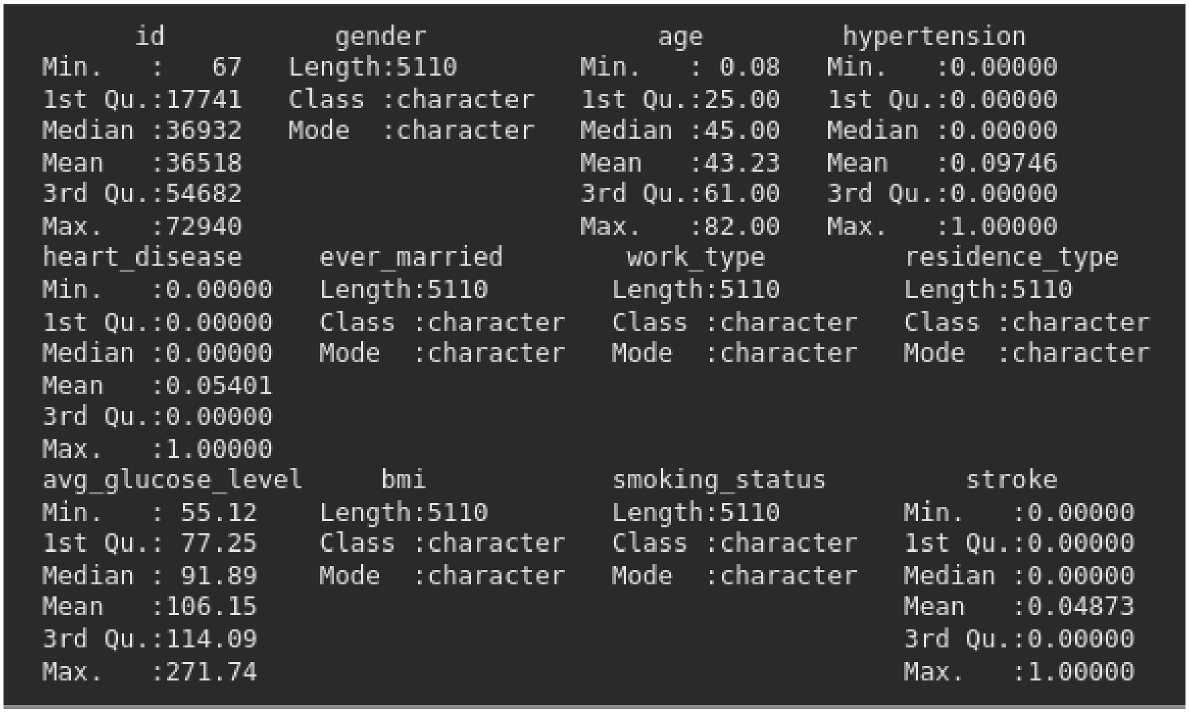
```
                                              A tibble: 6 × 12
   id  gender  age  hypertension  heart_disease  ever_married     work_type  residence_type  avg_glucose_level   bmi  smoking_status  stroke
<dbl>   <chr> <dbl>        <dbl>          <dbl>         <chr>         <chr>           <chr>              <dbl> <chr>           <chr>   <dbl>
 9046    Male   67             0              1           Yes       Private           Urban             228.69  36.6 formerly smoked       1
51676  Female   61             0              0           Yes Self-employed           Rural             202.21   N/A    never smoked       1
31112    Male   80             0              1           Yes       Private           Rural             105.92  32.5    never smoked       1
60182  Female   49             0              0           Yes       Private           Urban             171.23  34.4          smokes       1
 1665  Female   79             1              0           Yes Self-employed           Rural             174.12    24    never smoked       1
56669    Male   81             0              0           Yes       Private           Urban             186.21    29 formerly smoked       1
```

—

1.4 Dataset Summary

```
summary(stroke)
```

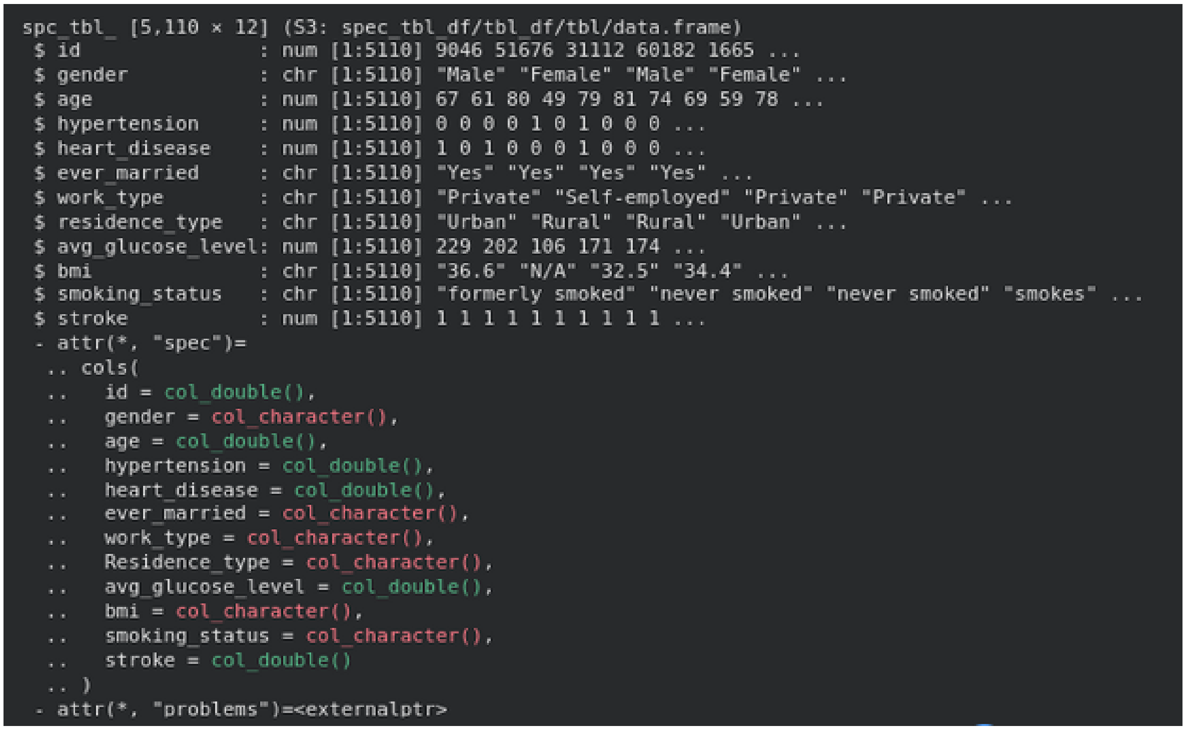
Output



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

```
str(stroke)
```

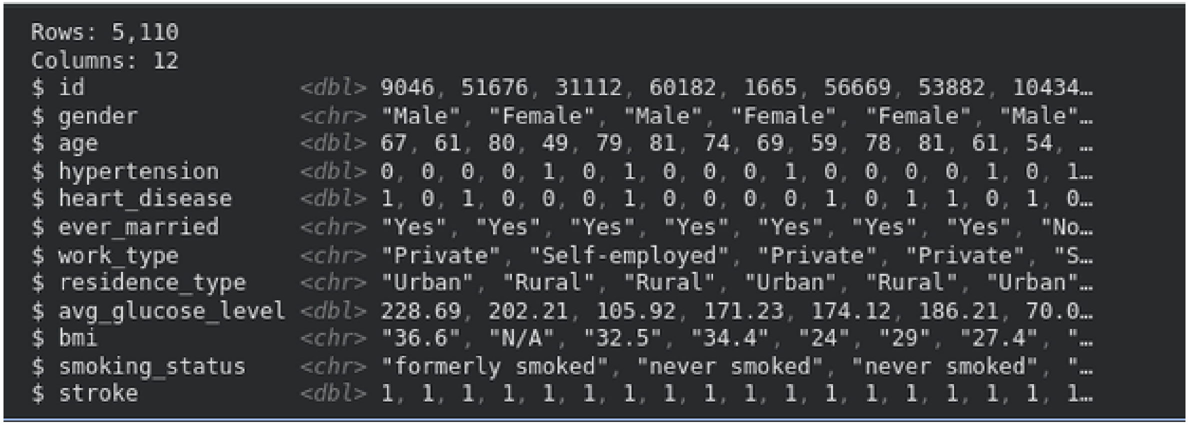
Output



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

```
glimpse(stroke)
```

Output

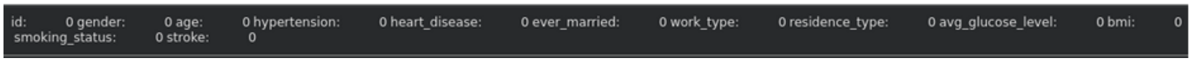


```
Rows: 5,110
Columns: 12
$ id                <dbl> 9046, 51676, 31112, 60182, 1665, 56669, 53882, 10434…
$ gender            <chr> "Male", "Female", "Male", "Female", "Female", "Male"…
$ age               <dbl> 67, 61, 80, 49, 79, 81, 74, 69, 59, 78, 81, 61, 54, …
$ hypertension      <dbl> 0, 0, 0,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

1.5 Missing Value Detection

```
colSums(is.na(stroke))
```

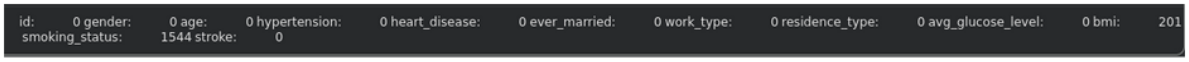
Output



```
id:        0 gender:      0 age:       0 hypertension:       0 heart_disease:       0 ever_married:       0 work_type:       0 residence_type:       0 avg_glucose_level:       0 bmi:       0
smoking_status:       0 stroke:       0
```

```
# Problem:
# colSums(is.na()) shows 0 missing values, but the dataset
# actually contains "N/A", "", and "Unknown" as *text strings*,
# which R does NOT treat as real NA.
# This step identifies and converts them to TRUE NA.

# CHECK "FAKE" MISSING VALUES

sapply(stroke, function(x) sum(x %in% c("N/A", "", "Unknown")))
```

Output



```
id:        0 gender:      0 age:       0 hypertension:       0 heart_disease:       0 ever_married:       0 work_type:       0 residence_type:       0 avg_glucose_level:       0 bmi:       201
smoking_status:       1544 stroke:       0
```

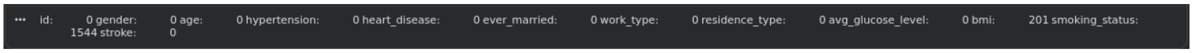
1.6 Missing Value Conversion

```
library(dplyr)
stroke_clean <- stroke %>%
  mutate(
    bmi = na_if(bmi, "N/A"),
    bmi = na_if(bmi, ""),
    smoking_status = na_if(smoking_status, "Unknown"),
    smoking_status = na_if(smoking_status, "N/A")
  )
```

```
# CONVERT BMI BACK TO NUMERIC
stroke_clean$bmi <- as.numeric(stroke_clean$bmi)
```

```
# CONFIRM MISSING VALUES ARE FIXED
colSums(is.na(stroke_clean))
```

Output



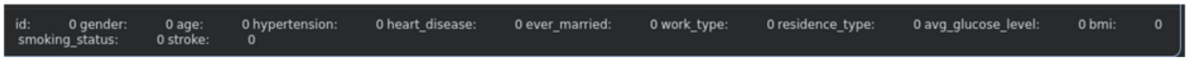
```
••• id:      0 gender:     0 age:      0 hypertension:      0 heart_disease:      0 ever_married:      0 work_type:      0 residence_type:      0 avg_glucose_level:      0 bmi:      201 smoking_status:
        1544 stroke:       0
```

—

1.7 Missing Value Imputation

```
# IMPUTE BMI (NUMERIC)
median_bmi <- median(stroke_clean$bmi, na.rm = TRUE)
stroke_clean$bmi[is.na(stroke_clean$bmi)] <- median_bmi

# IMPUTE SMOKING STATUS (CATEGORICAL)
get_mode <- function(x) {
  tab <- table(x)              # Frequency table
  mode_value <- names(tab)[which.max(tab)]  # Most common value
  return(mode_value)
}
mode_smoking <- get_mode(stroke_clean$smoking_status)
stroke_clean$smoking_status[is.na(stroke_clean$smoking_status)] <- mode_smoking

# CHECK THAT IMPUTATION WORKED
colSums(is.na(stroke_clean))
```

Output



```
id:      0 gender:     0 age:      0 hypertension:      0 heart_disease:      0 ever_married:      0 work_type:      0 residence_type:      0 avg_glucose_level:      0 bmi:      0
 smoking_status:       0 stroke:     0
```

—

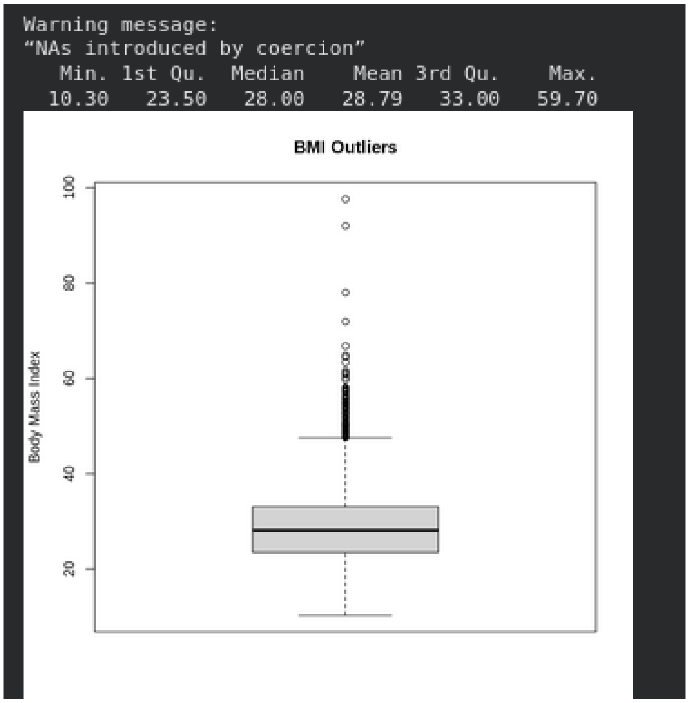
1.8 Data Type Conversion

```
stroke <- stroke %>%
  mutate(
    gender = as.factor(gender),
    ever_married = as.factor(ever_married),
    work_type = as.factor(work_type),
    residence_type = as.factor(residence_type),
    smoking_status = as.factor(smoking_status),
    stroke = as.factor(stroke)
  )
```

1.9 Outlier Detection

```
stroke$bmi <- as.numeric(stroke$bmi)
boxplot(
  stroke$bmi,
  main = "BMI Outliers",
  ylab = "Body Mass Index"
)
stroke <- stroke %>% filter(bmi < 60) # In medical datasets, BMI above 60 is extremely u
summary(stroke$bmi)
```

Output



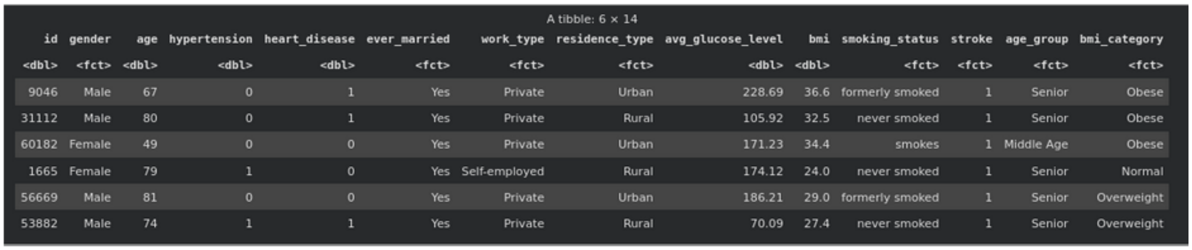
```
Warning message:
“NAs introduced by coercion”
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.30   23.50   28.00   28.79   33.00   59.70
```

—

1.10 Feature Engineering

```
stroke <- stroke %>%
  mutate(
    age_group = case_when(
      age < 18 ~ "Child",
      age >= 18 & age < 40 ~ "Young Adult",
      age >= 40 & age < 60 ~ "Middle Age",
      TRUE ~ "Senior"
    ),
    bmi_category = case_when(
      bmi < 18.5 ~ "Underweight",
      bmi >= 18.5 & bmi < 25 ~ "Normal",
      bmi >= 25 & bmi < 30 ~ "Overweight",
      TRUE ~ "Obese"
    )
  )

stroke$age_group <- as.factor(stroke$age_group)
stroke$bmi_category <- as.factor(stroke$bmi_category)

head(stroke)
```

Output



A tibble: 6 × 14

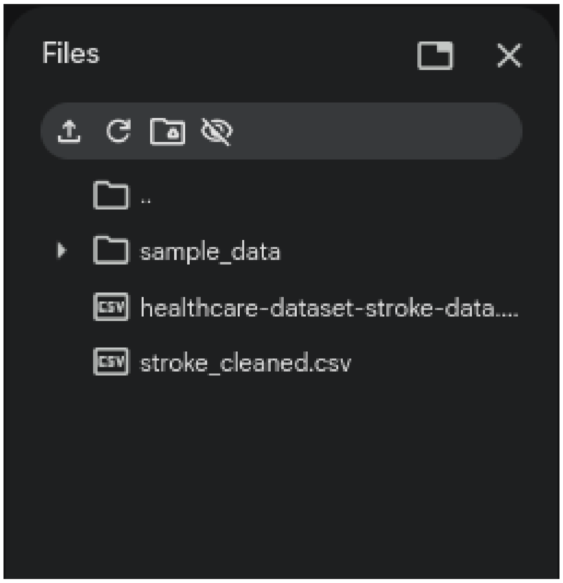
id	gender	age	hypertension	heart_disease	ever_married	work_type	residence_type	avg_glucose_level	bmi	smoking_status	stroke	age_group	bmi_category
<dbl>	<fct>	<dbl>	<dbl>	<dbl>	<fct>	<fct>	<fct>	<dbl>	<dbl>	<fct>	<fct>	<fct>	<fct>
9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1	Senior	Obese
31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1	Senior	Obese
60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1	Middle Age	Obese
1665	Female	79	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1	Senior	Normal
56669	Male	81	0	0	Yes	Private	Urban	186.21	29.0	formerly smoked	1	Senior	Overweight
53882	Male	74	1	1	Yes	Private	Rural	70.09	27.4	never smoked	1	Senior	Overweight

—

1.11 Output File

```
write_csv(stroke, "stroke_cleaned.csv")
```

Output



Files

..
sample_data
healthcare-dataset-stroke-data....
stroke_cleaned.csv

Chapter 2

Exploratory Data Analysis

2.1 Library Installation and Loading

```
packages <- c("tidyverse", "janitor", "skimr", "GGally", "psych")

for (p in packages) {
  if (!requireNamespace(p, quietly = TRUE)) {
    install.packages(p, repos = "https://cloud.r-project.org")
  }
  library(p, character.only = TRUE)
}
```

—

2.2 Library Loading

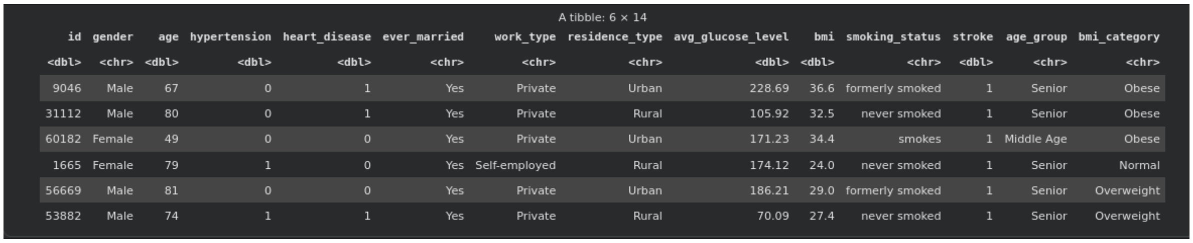
```
library(tidyverse)    # dplyr + ggplot2
library(ggplot2)
library(dplyr)
library(readr)
library(psych)        # for describe()
library(GGally)       # for correlation plots
```

—

2.3 Data Loading

```
stroke <- read_csv("stroke_cleaned.csv")
head(stroke)
```

Output



A tibble: 6 × 14

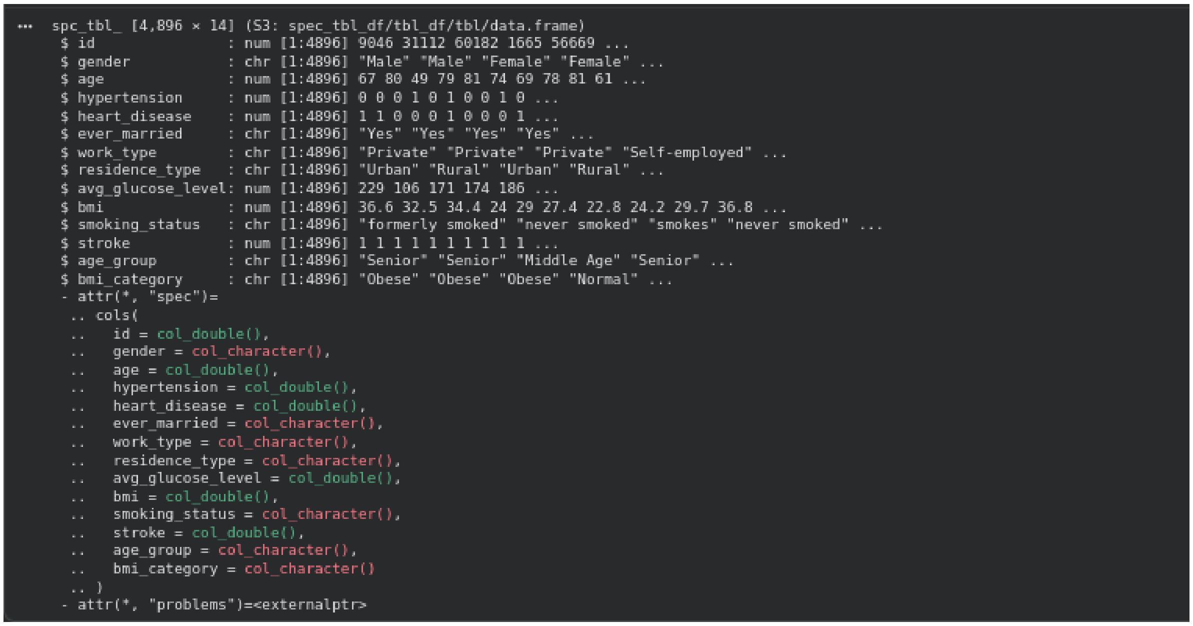
id	gender	age	hypertension	heart_disease	ever_married	work_type	residence_type	avg_glucose_level	bmi	smoking_status	stroke	age_group	bmi_category
<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<chr>	<dbl>	<chr>	<chr>
9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1	Senior	Obese
31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1	Senior	Obese
60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1	Middle Age	Obese
1665	Female	79	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1	Senior	Normal
56669	Male	81	0	0	Yes	Private	Urban	186.21	29.0	formerly smoked	1	Senior	Overweight
53882	Male	74	1	1	Yes	Private	Rural	70.09	27.4	never smoked	1	Senior	Overweight

—

2.4 Dataset Overview

```
str(stroke)
```

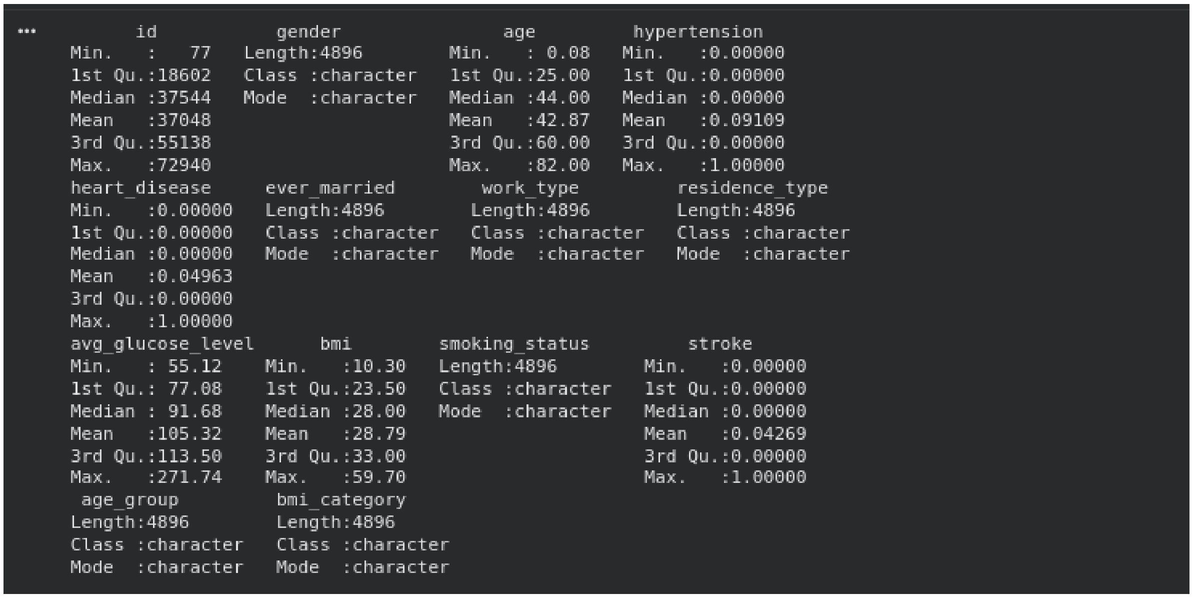
Output



```
spc_tbl_ [4,896 × 14] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ id               : num [1:4896] 9046 31112 60182 1665 56669 ...
 $ gender           : chr [1:4896] "Male" "Male" "Female" "Female" ...
 $ age              : num [1:4896] 67 80 49 79 81 74 69 78 81 61 ...
 $ hypertension     : num [1:4896] 0 0 0 1 0 1 0 0 1 0 ...
 $ heart_disease    : num [1:4896] 1 1 0 0 0 1 0 0 0 1 ...
 $ ever_married     : chr [1:4896] "Yes" "Yes" "Yes" "Yes" ...
 $ work_type        : chr [1:4896] "Private" "Private" "Private" "Self-employed" ...
 $ residence_type   : chr [1:4896] "Urban" "Rural" "Urban" "Rural" ...
 $ avg_glucose_level: num [1:4896] 229 106 171 174 186 ...
 $ bmi              : num [1:4896] 36.6 32.5 34.4 24 29 27.4 22.8 24.2 29.7 36.8 ...
 $ smoking_status   : chr [1:4896] "formerly smoked" "never smoked" "smokes" "never smoked" ...
 $ stroke           : num [1:4896] 1 1 1 1 1 1 1 1 1 1 ...
 $ age_group        : chr [1:4896] "Senior" "Senior" "Middle Age" "Senior" ...
 $ bmi_category     : chr [1:4896] "Obese" "Obese" "Obese" "Normal" ...
 - 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_double(),
  ..   smoking_status = col_character(),
  ..   stroke = col_double(),
  ..   age_group = col_character(),
  ..   bmi_category = col_character()
  .. )
 - attr(*, "problems")=<externalptr>
```

```
summary(stroke)
```

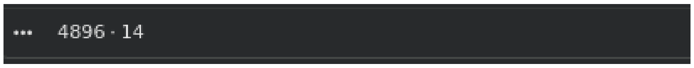
Output



```
•••        id           gender               age         hypertension
     Min.   :   77   Length:4896        Min.   : 0.08   Min.   :0.00000
     1st Qu.:18602   Class :character   1st Qu.:25.00   1st Qu.:0.00000
     Median :37544   Mode  :character   Median :44.00   Median :0.00000
     Mean   :37048                      Mean   :42.87   Mean   :0.09109
     3rd Qu.:55138                      3rd Qu.:60.00   3rd Qu.:0.00000
     Max.   :72940                      Max.   :82.00   Max.   :1.00000
     heart_disease     ever_married        work_type         residence_type
     Min.   :0.00000   Length:4896        Length:4896        Length:4896
     1st Qu.:0.00000   Class :character   Class :character   Class :character
     Median :0.00000   Mode  :character   Mode  :character   Mode  :character
     Mean   :0.04963
     3rd Qu.:0.00000
     Max.   :1.00000
     avg_glucose_level      bmi        smoking_status         stroke
     Min.   : 55.12    Min.   :10.30   Length:4896        Min.   :0.00000
     1st Qu.: 77.08    1st Qu.:23.50   Class :character   1st Qu.:0.00000
     Median : 91.68    Median :28.00   Mode  :character   Median :0.00000
     Mean   :105.32    Mean   :28.79                      Mean   :0.04269
     3rd Qu.:113.50    3rd Qu.:33.00                      3rd Qu.:0.00000
     Max.   :271.74    Max.   :59.70                      Max.   :1.00000
      age_group         bmi_category
     Length:4896        Length:4896
     Class :character   Class :character
     Mode  :character   Mode  :character
```

```
dim(stroke)
```

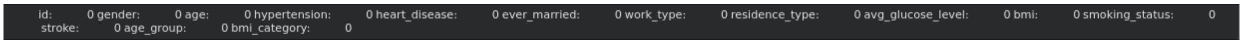
Output



```
•••  4896 · 14
```

```
colSums(is.na(stroke))
```

Output



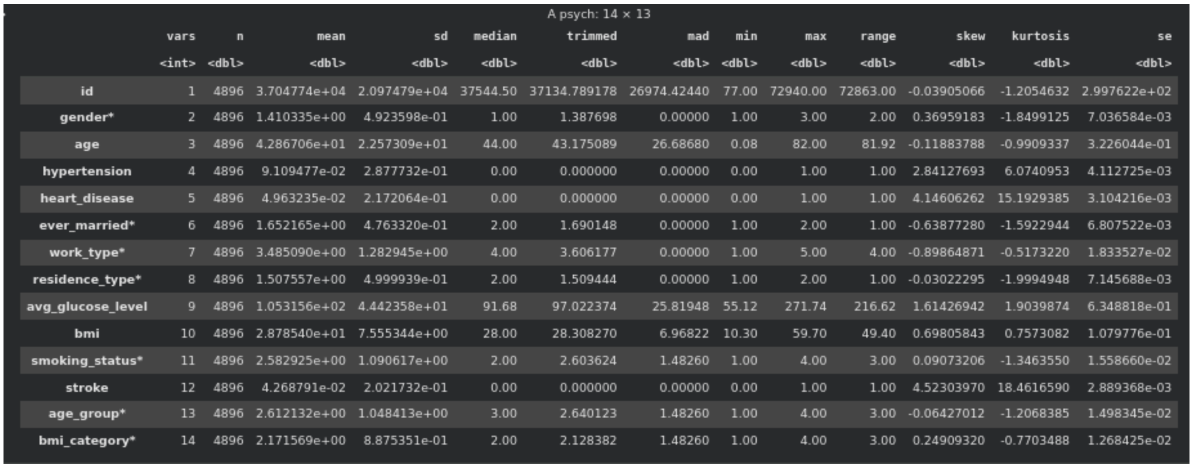
```
id:      0 gender:      0 age:      0 hypertension:      0 heart_disease:      0 ever_married:      0 work_type:      0 residence_type:      0 avg_glucose_level:      0 bmi:      0 smoking_status:      0
stroke:      0 age_group:      0 bmi_category:      0
```

—

2.5 Descriptive Statistics

```
psych::describe(stroke)
```

Output



A psych: 14 × 13

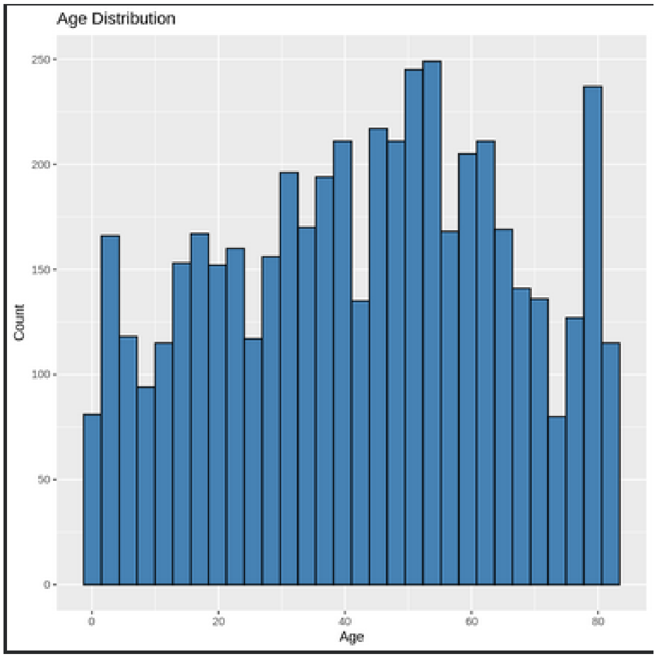
	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
id	1	4896	3.704774e+04	2.097479e+04	37544.50	37134.789178	26974.42440	77.00	72940.00	72863.00	-0.03905066	-1.2054632	2.997622e+02
gender*	2	4896	1.410335e+00	4.923598e-01	1.00	1.387698	0.00000	1.00	3.00	2.00	0.36959183	-1.8499125	7.036584e-03
age	3	4896	4.286706e+01	2.257309e+01	44.00	43.175089	26.68680	0.08	82.00	81.92	-0.11883788	-0.9909337	3.226044e-01
hypertension	4	4896	9.109477e-02	2.877732e-01	0.00	0.000000	0.00000	0.00	1.00	1.00	2.84127693	6.0740953	4.112725e-03
heart_disease	5	4896	4.963235e-02	2.172064e-01	0.00	0.000000	0.00000	0.00	1.00	1.00	4.14606262	15.1929385	3.104216e-03
ever_married*	6	4896	1.652165e+00	4.763320e-01	2.00	1.690148	0.00000	1.00	2.00	1.00	-0.63877280	-1.5922944	6.807522e-03
work_type*	7	4896	3.485090e+00	1.282945e+00	4.00	3.606177	0.00000	1.00	5.00	4.00	-0.89864871	-0.5173220	1.833527e-02
residence_type*	8	4896	1.507557e+00	4.999939e-01	2.00	1.509444	0.00000	1.00	2.00	1.00	-0.03022295	-1.9994948	7.145688e-03
avg_glucose_level	9	4896	1.053156e+02	4.442358e+01	91.68	97.022374	25.81948	55.12	271.74	216.62	1.61426942	1.9039874	6.348818e-01
bmi	10	4896	2.878540e+01	7.555344e+00	28.00	28.308270	6.96822	10.30	59.70	49.40	0.69805843	0.7573082	1.079776e-01
smoking_status*	11	4896	2.582925e+00	1.090617e+00	2.00	2.603624	1.48260	1.00	4.00	3.00	0.09073206	-1.3463550	1.558660e-02
stroke	12	4896	4.268791e-02	2.021732e-01	0.00	0.000000	0.00000	0.00	1.00	1.00	4.52303970	18.4616590	2.889368e-03
age_group*	13	4896	2.612132e+00	1.048413e+00	3.00	2.640123	1.48260	1.00	4.00	3.00	-0.06427012	-1.2068385	1.498345e-02
bmi_category*	14	4896	2.171569e+00	8.875351e-01	2.00	2.128382	1.48260	1.00	4.00	3.00	0.24909320	-0.7703488	1.268425e-02

—

2.6 Univariate Distributions

```
# Age Distribution
ggplot(stroke, aes(age)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "black") +
  labs(title = "Age Distribution", x = "Age", y = "Count")
```

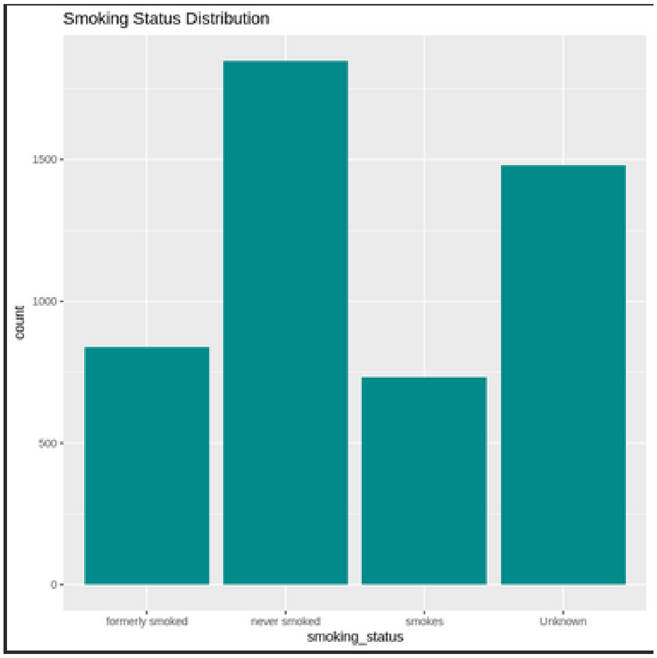
Output



In the same manner, histograms for BMI and Average Glucose Level were created.

```
# Smoking Status Distribution
ggplot(stroke, aes(x = smoking_status)) +
  geom_bar(fill = "cyan4") +
  labs(title = "Smoking Status Distribution")
```

Output

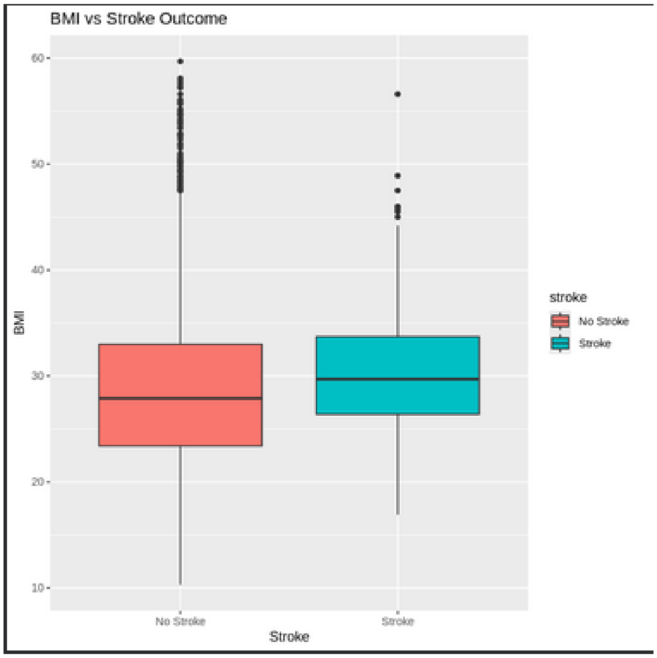


In the same manner, bar plots for Gender , Stroke Occurrence and Work Type were created. —

2.7 Bivariate Analysis

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

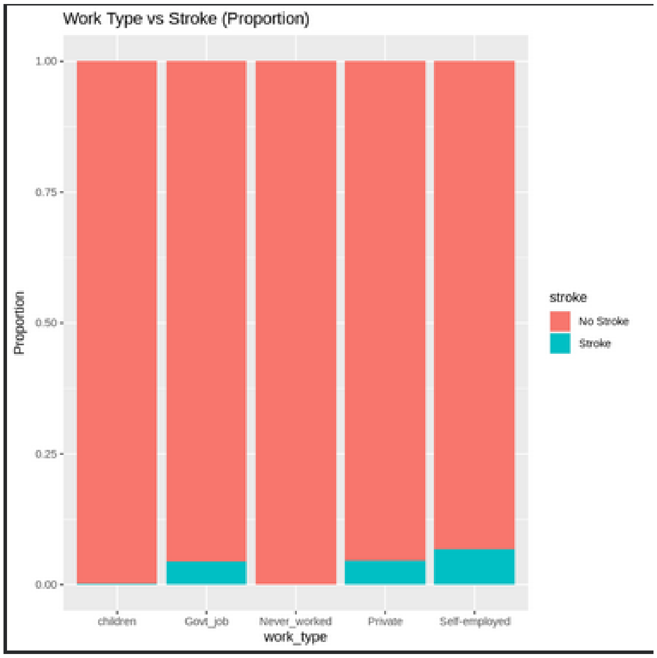
Output



In the same manner, boxplots for Average Glucose Level vs Stroke and Age vs Stroke were created.

```
# Work Type vs Stroke (Proportion)
ggplot(stroke, aes(x = work_type, fill = stroke)) +
  geom_bar(position = "fill") +
  labs(title = "Work Type vs Stroke (Proportion)", y = "Proportion")
```

Output



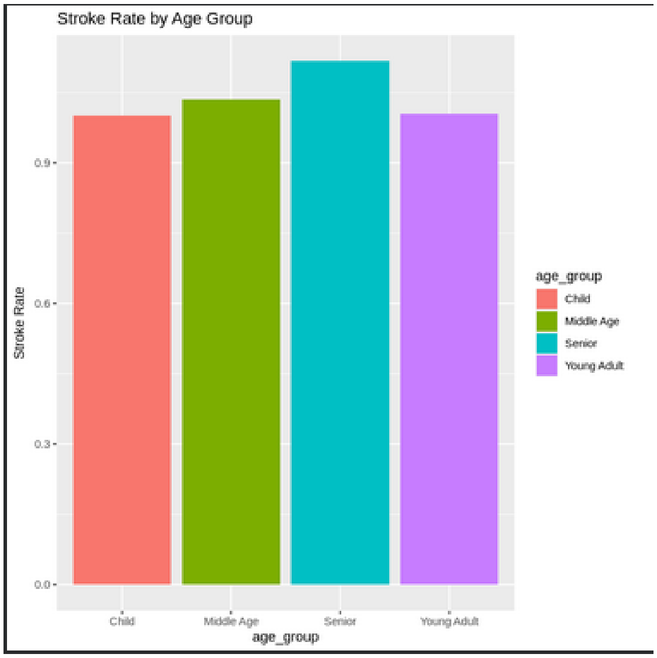
In the same manner , stacked bar charts for Gender vs Stroke and Smoking Status vs Stroke were created.

—

2.8 Stroke Rates

```
# Stroke Rate by Age Group
stroke %>%
  group_by(age_group) %>%                          # Group by age_group
  summarise(stroke_rate = mean(as.numeric(stroke))) %>% # Compute stroke rate
  ggplot(aes(x = age_group, y = stroke_rate, fill = age_group)) +
  geom_col() +
  labs(title = "Stroke Rate by Age Group", y = "Stroke Rate")
```

Output

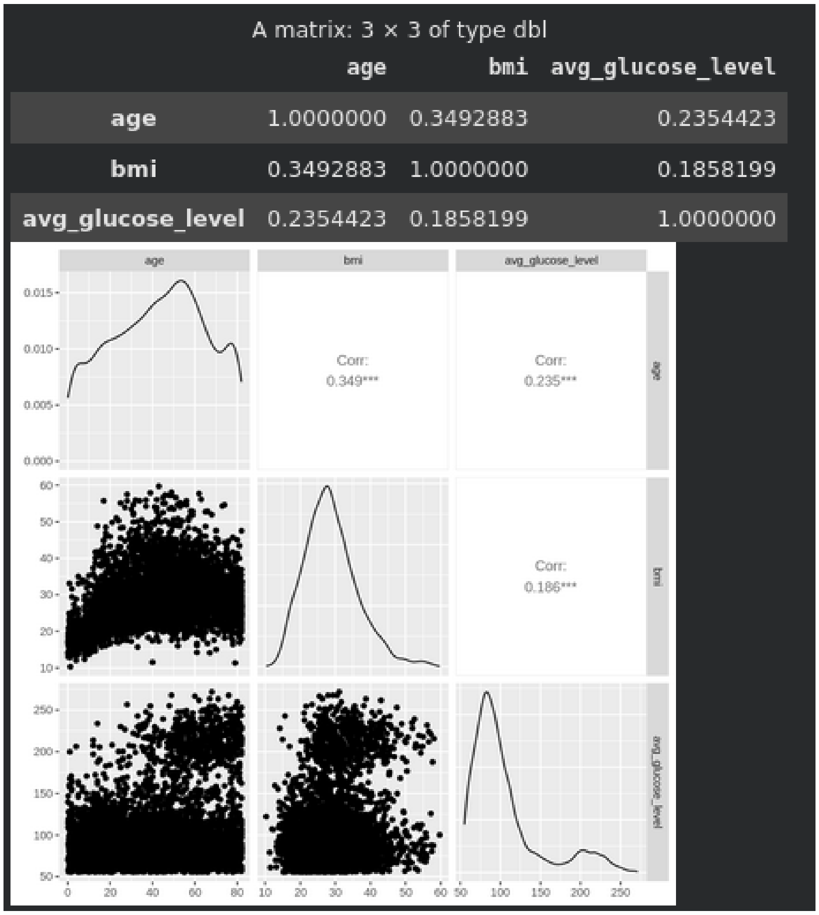


In the same manner, stroke rates by BMI and Smoking Status were calculated and visualized. —

2.9 Correlation Analysis

```
num_data <- stroke %>% select(age, bmi, avg_glucose_level)
cor(num_data)
GGally::ggpairs(num_data)
```

Output



A matrix: 3 × 3 of type dbl

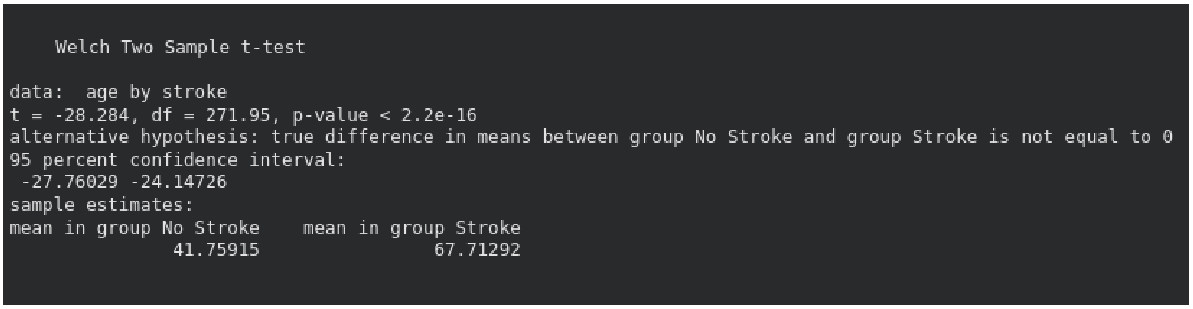
	age	bmi	avg_glucose_level
age	1.0000000	0.3492883	0.2354423
bmi	0.3492883	1.0000000	0.1858199
avg_glucose_level	0.2354423	0.1858199	1.0000000

—

2.10 Statistical Tests: T- tests

```
t.test(age ~ stroke, data = stroke)
```

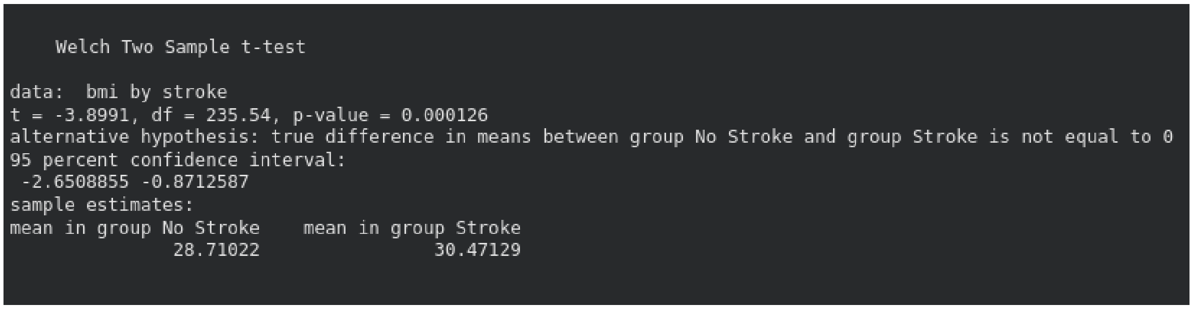
Output



```
    Welch Two Sample t-test

data:  age by stroke
t = -28.284, df = 271.95, p-value < 2.2e-16
alternative hypothesis: true difference in means between group No Stroke and group Stroke is not equal to 0
95 percent confidence interval:
 -27.76029 -24.14726
sample estimates:
mean in group No Stroke    mean in group Stroke
               41.75915                67.71292
```

```
t.test(bmi ~ stroke, data = stroke)
```

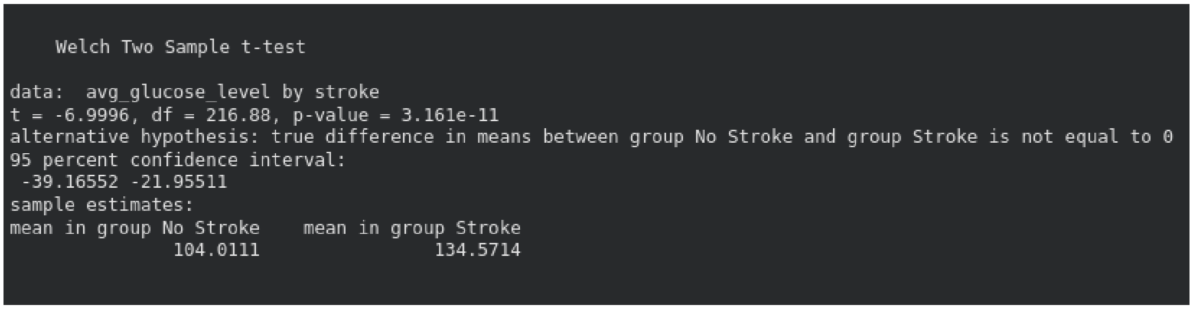
Output



```
	Welch Two Sample t-test

data:  bmi by stroke
t = -3.8991, df = 235.54, p-value = 0.000126
alternative hypothesis: true difference in means between group No Stroke and group Stroke is not equal to 0
95 percent confidence interval:
 -2.6508855 -0.8712587
sample estimates:
mean in group No Stroke    mean in group Stroke 
               28.71022                30.47129 
```

```
t.test(avg_glucose_level ~ stroke, data = stroke)
```

Output

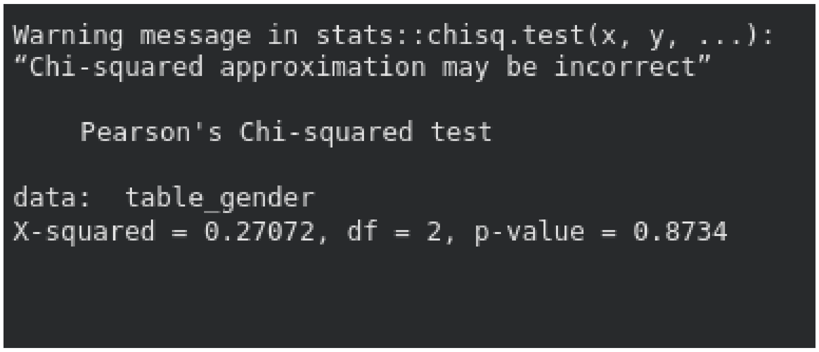


```
	Welch Two Sample t-test

data:  avg_glucose_level by stroke
t = -6.9996, df = 216.88, p-value = 3.161e-11
alternative hypothesis: true difference in means between group No Stroke and group Stroke is not equal to 0
95 percent confidence interval:
 -39.16552 -21.95511
sample estimates:
mean in group No Stroke    mean in group Stroke 
               104.0111                134.5714 
```

2.11 Statistical Tests:Chi-square Tests

```
table_gender <- table(stroke$gender, stroke$stroke)
chisq.test(table_gender)
```

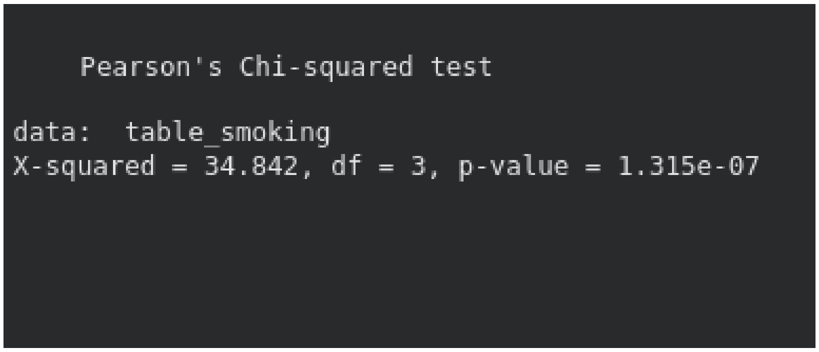
Output



```
Warning message in stats::chisq.test(x, y, ...):
“Chi-squared approximation may be incorrect”

    Pearson's Chi-squared test

data:  table_gender
X-squared = 0.27072, df = 2, p-value = 0.8734
```

```
table_smoking <- table(stroke$smoking_status, stroke$stroke)
chisq.test(table_smoking)
```

Output



```

    Pearson's Chi-squared test

data:  table_smoking
X-squared = 34.842, df = 3, p-value = 1.315e-07
```

Chapter 3

Modeling and Evaluation

3.1 Library Loading

```
library(tidyverse)
library(caret)
library(stats)
library(randomForest)
library(pROC)
library(themis)
```

—

3.2 Data Preparation

```
stroke_data$stroke <- factor(stroke_data$stroke, levels = c(0, 1))
set.seed(123)

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

train_data <- stroke_data[train_index, ]
test_data  <- stroke_data[-train_index, ]
```

—

3.3 Feature Scaling

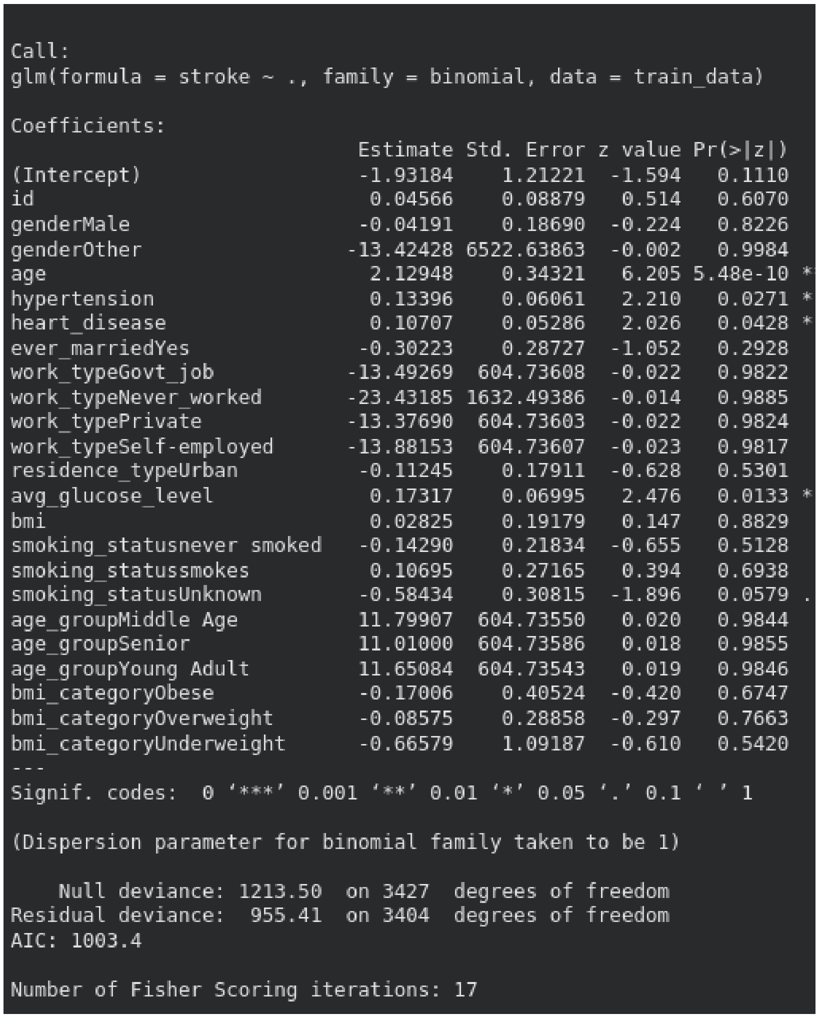
```
numeric_cols <- sapply(train_data, is.numeric)

train_data[, numeric_cols] <- scale(train_data[, numeric_cols])
test_data[, numeric_cols]  <- scale(test_data[, numeric_cols])
```

—

3.4 Logistic Regression

```
log_model <- glm(
  stroke ˜ .,
  data = train_data,
  family = binomial
)

summary(log_model)
```

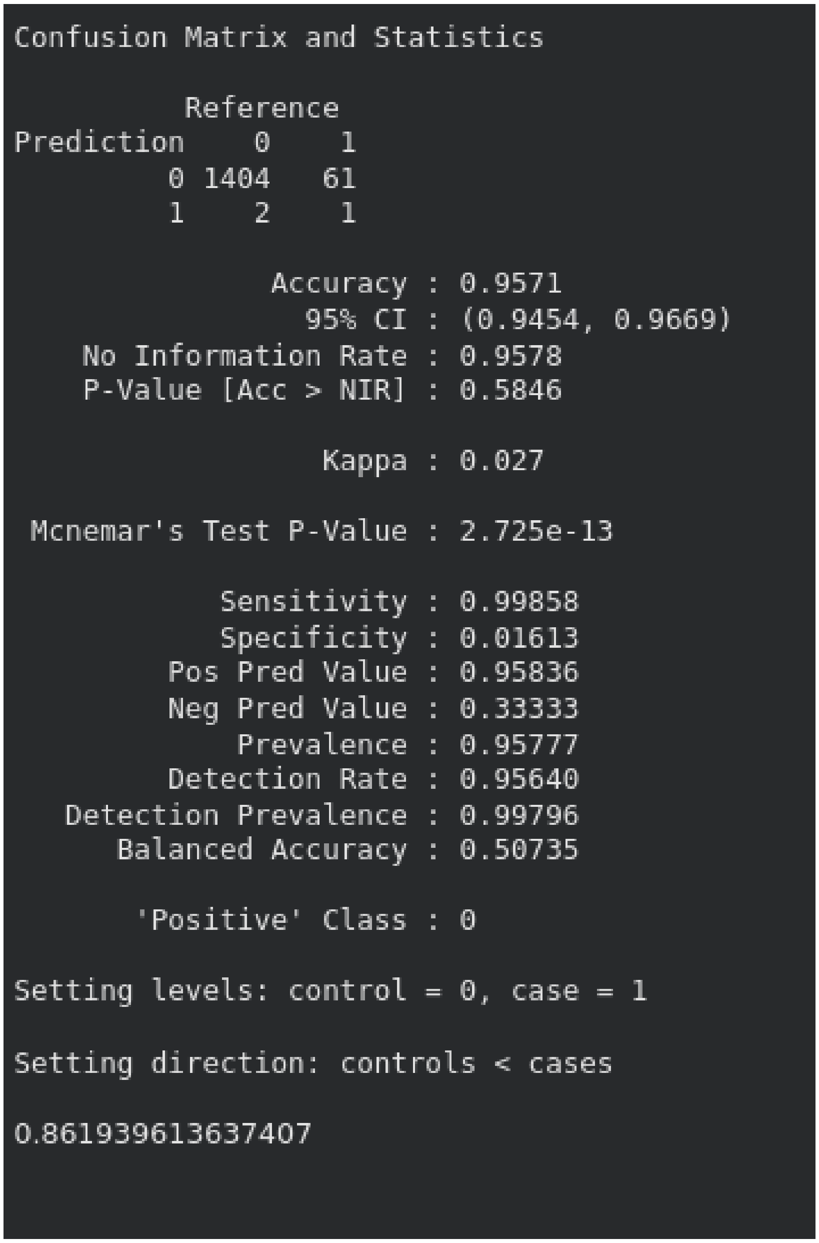
Output



```
Call:
glm(formula = stroke ~ ., family = binomial, data = train_data)

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -1.93184    1.21221  -1.594   0.1110
id                           0.04566    0.08879   0.514   0.6070
genderMale                  -0.04191    0.18690  -0.224   0.8226
genderOther                -13.42428 6522.63863  -0.002   0.9984
age                          2.12948    0.34321   6.205 5.48e-10 *
hypertension                 0.13396    0.06061   2.210   0.0271 *
heart_disease                0.10707    0.05286   2.026   0.0428 *
ever_marriedYes             -0.30223    0.28727  -1.052   0.2928
work_typeGovt_job          -13.49269  604.73608  -0.022   0.9822
work_typeNever_worked      -23.43185 1632.49386  -0.014   0.9885
work_typePrivate           -13.37690  604.73603  -0.022   0.9824
work_typeSelf-employed     -13.88153  604.73607  -0.023   0.9817
residence_typeUrban         -0.11245    0.17911  -0.628   0.5301
avg_glucose_level            0.17317    0.06995   2.476   0.0133 *
bmi                          0.02825    0.19179   0.147   0.8829
smoking_statusnever smoked  -0.14290    0.21834  -0.655   0.5128
smoking_statussmokes         0.10695    0.27165   0.394   0.6938
smoking_statusUnknown       -0.58434    0.30815  -1.896   0.0579 .
age_groupMiddle Age         11.79907  604.73550   0.020   0.9844
age_groupSenior             11.01000  604.73586   0.018   0.9855
age_groupYoung Adult        11.65084  604.73543   0.019   0.9846
bmi_categoryObese           -0.17006    0.40524  -0.420   0.6747
bmi_categoryOverweight      -0.08575    0.28858  -0.297   0.7663
bmi_categoryUnderweight     -0.66579    1.09187  -0.610   0.5420
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1213.50  on 3427  degrees of freedom
Residual deviance:  955.41  on 3404  degrees of freedom
AIC: 1003.4

Number of Fisher Scoring iterations: 17
```

```
log_probs <- predict(log_model, test_data, type = "response")
log_pred <- ifelse(log_probs > 0.5, 1, 0)
log_pred <- factor(log_pred, levels = c(0, 1))
confusionMatrix(log_pred, test_data$stroke)

log_roc <- roc(test_data$stroke, log_probs)
auc(log_roc)
```

Output



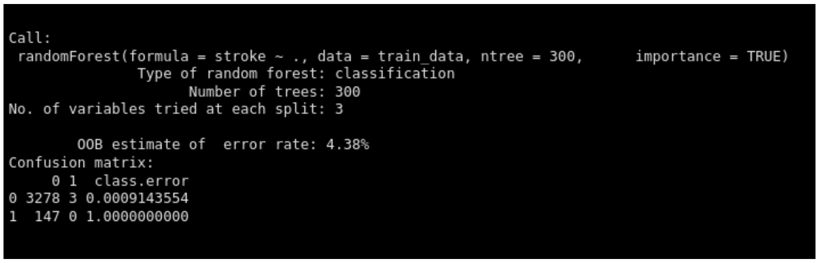
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1404   61
         1    2    1

               Accuracy : 0.9571
                 95% CI : (0.9454, 0.9669)
    No Information Rate : 0.9578
    P-Value [Acc > NIR] : 0.5846

                  Kappa : 0.027

 Mcnemar's Test P-Value : 2.725e-13

            Sensitivity : 0.99858
            Specificity : 0.01613
         Pos Pred Value : 0.95836
         Neg Pred Value : 0.33333
             Prevalence : 0.95777
         Detection Rate : 0.95640
   Detection Prevalence : 0.99796
      Balanced Accuracy : 0.50735

       'Positive' Class : 0

Setting levels: control = 0, case = 1

Setting direction: controls < cases

0.861939613637407
```

—

3.5 Random Forest

```
rf_model <- randomForest(
  stroke ˜ .,
  data = train_data,
  ntree = 300,
  importance = TRUE
)

print(rf_model)
```

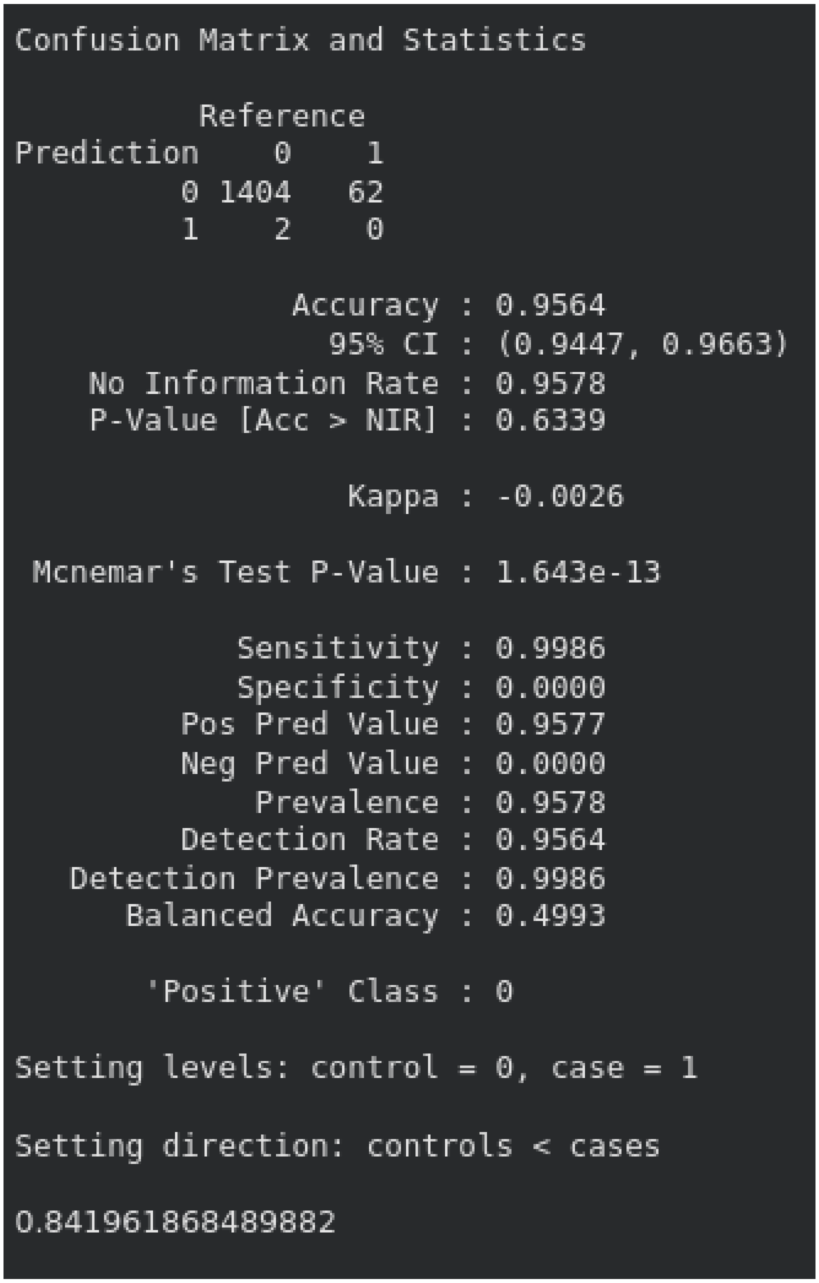
Output



```
Call:
 randomForest(formula = stroke ~ ., data = train_data, ntree = 300,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 300
No. of variables tried at each split: 3

        OOB estimate of  error rate: 4.38%
Confusion matrix:
     0 1  class.error
0 3278 3 0.0009143554
1  147 0 1.0000000000
```

```
rf_pred <- predict(rf_model, test_data)
confusionMatrix(rf_pred, test_data$stroke)

rf_probs <- predict(rf_model, test_data, type = "prob")[,2]
rf_roc <- roc(test_data$stroke, rf_probs)
auc(rf_roc)
```

Output



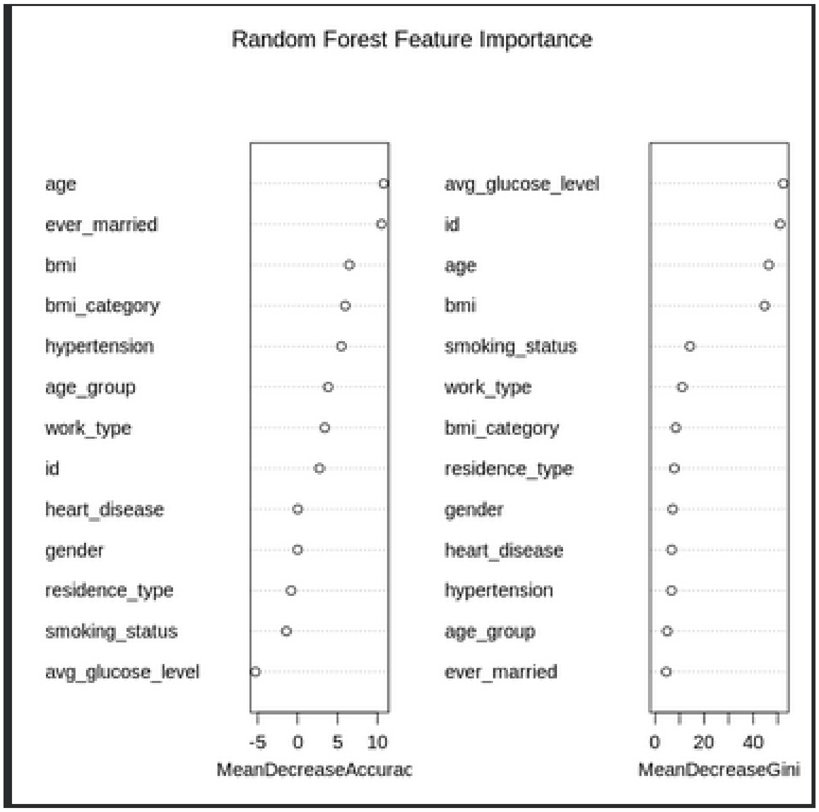
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1404   62
         1    2    0

               Accuracy : 0.9564
                 95% CI : (0.9447, 0.9663)
    No Information Rate : 0.9578
    P-Value [Acc > NIR] : 0.6339

                  Kappa : -0.0026

 Mcnemar's Test P-Value : 1.643e-13

            Sensitivity : 0.9986
            Specificity : 0.0000
         Pos Pred Value : 0.9577
         Neg Pred Value : 0.0000
             Prevalence : 0.9578
         Detection Rate : 0.9564
   Detection Prevalence : 0.9986
      Balanced Accuracy : 0.4993

       'Positive' Class : 0

Setting levels: control = 0, case = 1

Setting direction: controls < cases

0.841961868489882
```

—

3.6 Feature Importance

```
varImpPlot(rf_model, main = "Random Forest Feature Importance")
```

Output

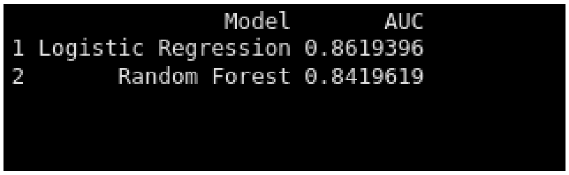


—

3.7 Model Comparison

```
comparison <- data.frame(
  Model = c("Logistic Regression", "Random Forest"),
  AUC = c(auc(log_roc), auc(rf_roc))
)

print(comparison)
```

Output



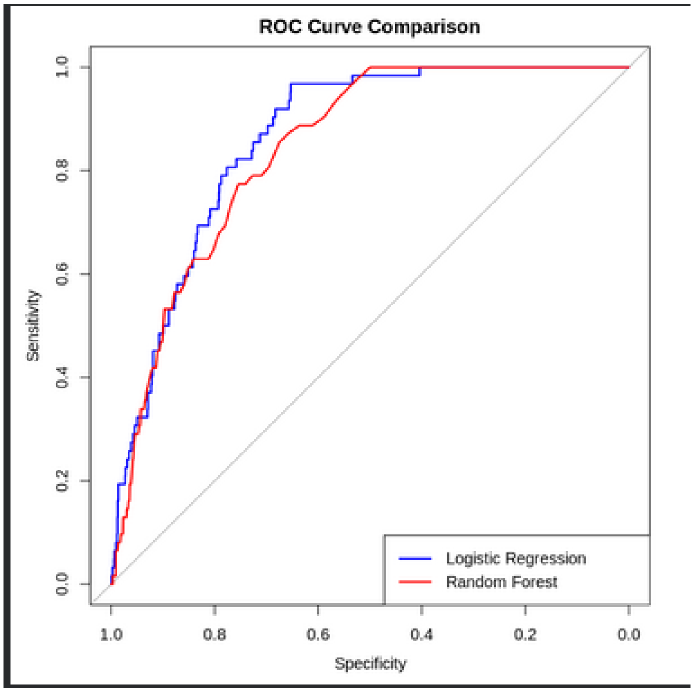
```
                Model       AUC
1 Logistic Regression 0.8619396
2       Random Forest 0.8419619
```

—

3.8 ROC Curve Comparison

```
plot(log_roc, col = "blue", lwd = 2, main = "ROC Curve Comparison")
plot(rf_roc, col = "red", lwd = 2, add = TRUE)

legend(
  "bottomright",
  legend = c("Logistic Regression", "Random Forest"),
  col = c("blue", "red"),
  lwd = 2
)
```

Output



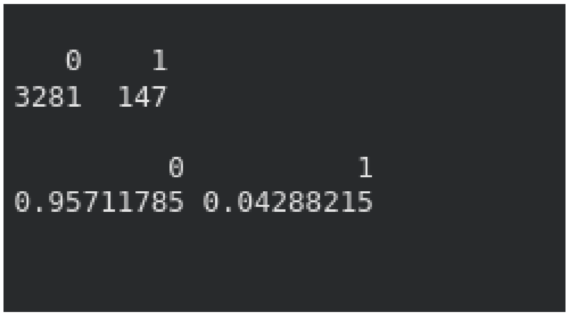
—

3.9 Class Distribution

```
table(train_data$stroke)
```

```
prop.table(table(train_data$stroke))
```

Output



```
   0    1 
3281  147 

         0          1 
0.95711785 0.04288215 
```

—

3.10 SMOTE Preprocessing

```
smote_recipe <- recipe(stroke ~ ., data = train_data) %>%
  step_mutate(stroke = factor(stroke)) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_smote(stroke)

smote_prep <- prep(smote_recipe)

train_smote <- bake(smote_prep, new_data = NULL)
test_smote  <- bake(smote_prep, new_data = test_data)
```

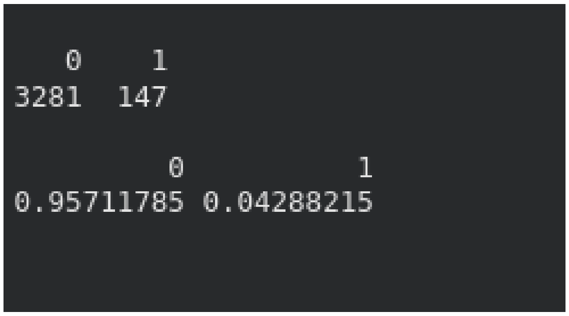
—

3.11 Models After SMOTE: Logistic Regression

```
log_smote_model <- glm(
  stroke ~ .,
  data = train_smote,
  family = binomial
)
```

```
log_smote_probs <- predict(
  log_smote_model,
  newdata = test_smote,
  type = "response"
)

log_smote_pred <- ifelse(log_smote_probs > 0.5, 1, 0)
log_smote_pred <- factor(log_smote_pred, levels = c(0, 1))

confusionMatrix(log_smote_pred, test_smote$stroke)
```

Output



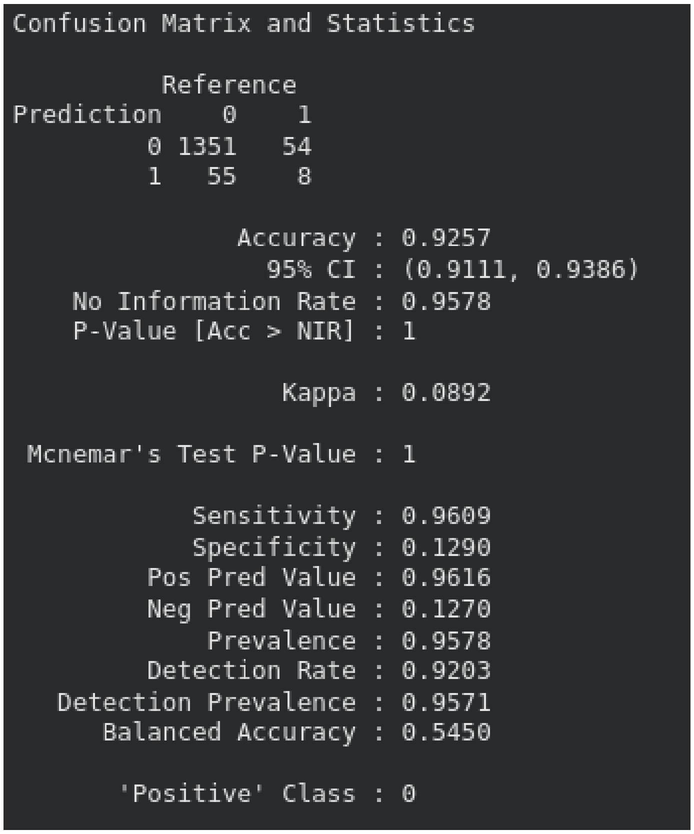
```
   0    1
3281  147

         0          1
0.95711785 0.04288215
```

—

3.12 Models After SMOTE:Random Forest

```
set.seed(123)
rf_smote_model <- randomForest(
  stroke ˜ .,
  data = train_smote,
  ntree = 300,
  importance = TRUE
)

rf_smote_pred <- predict(rf_smote_model, test_smote)
confusionMatrix(rf_smote_pred, test_smote$stroke)
```

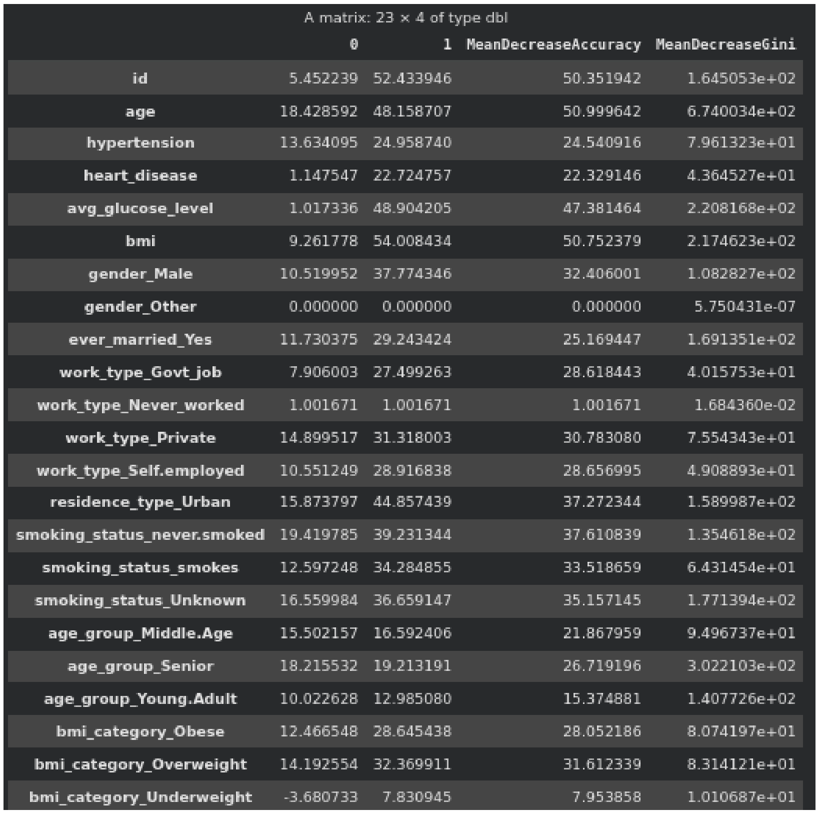
Output



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1351   54
         1   55    8

               Accuracy : 0.9257
                 95% CI : (0.9111, 0.9386)
    No Information Rate : 0.9578
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0892

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9609
            Specificity : 0.1290
         Pos Pred Value : 0.9616
         Neg Pred Value : 0.1270
             Prevalence : 0.9578
         Detection Rate : 0.9203
   Detection Prevalence : 0.9571
      Balanced Accuracy : 0.5450

       'Positive' Class : 0
```

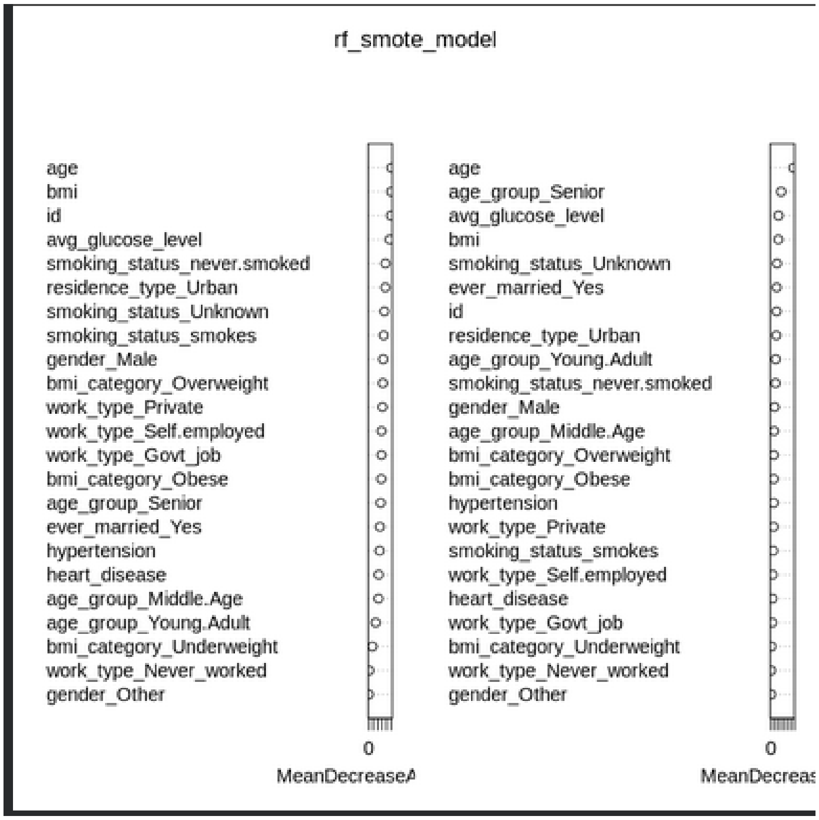
```
importance(rf_smote_model)
varImpPlot(rf_smote_model)
```

Output



A matrix: 23 × 4 of type dbl

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
id	5.452239	52.433946	50.351942	1.645053e+02
age	18.428592	48.158707	50.999642	6.740034e+02
hypertension	13.634095	24.958740	24.540916	7.961323e+01
heart_disease	1.147547	22.724757	22.329146	4.364527e+01
avg_glucose_level	1.017336	48.904205	47.381464	2.208168e+02
bmi	9.261778	54.008434	50.752379	2.174623e+02
gender_Male	10.519952	37.774346	32.406001	1.082827e+02
gender_Other	0.000000	0.000000	0.000000	5.750431e-07
ever_married_Yes	11.730375	29.243424	25.169447	1.691351e+02
work_type_Govt_job	7.906003	27.499263	28.618443	4.015753e+01
work_type_Never_worked	1.001671	1.001671	1.001671	1.684360e-02
work_type_Private	14.899517	31.318003	30.783080	7.554343e+01
work_type_Self.employed	10.551249	28.916838	28.656995	4.908893e+01
residence_type_Urban	15.873797	44.857439	37.272344	1.589987e+02
smoking_status_never.smoked	19.419785	39.231344	37.610839	1.354618e+02
smoking_status_smokes	12.597248	34.284855	33.518659	6.431454e+01
smoking_status_Unknown	16.559984	36.659147	35.157145	1.771394e+02
age_group_Middle.Age	15.502157	16.592406	21.867959	9.496737e+01
age_group_Senior	18.215532	19.213191	26.719196	3.022103e+02
age_group_Young.Adult	10.022628	12.985080	15.374881	1.407726e+02
bmi_category_Obese	12.466548	28.645438	28.052186	8.074197e+01
bmi_category_Overweight	14.192554	32.369911	31.612339	8.314121e+01
bmi_category_Underweight	-3.680733	7.830945	7.953858	1.010687e+01

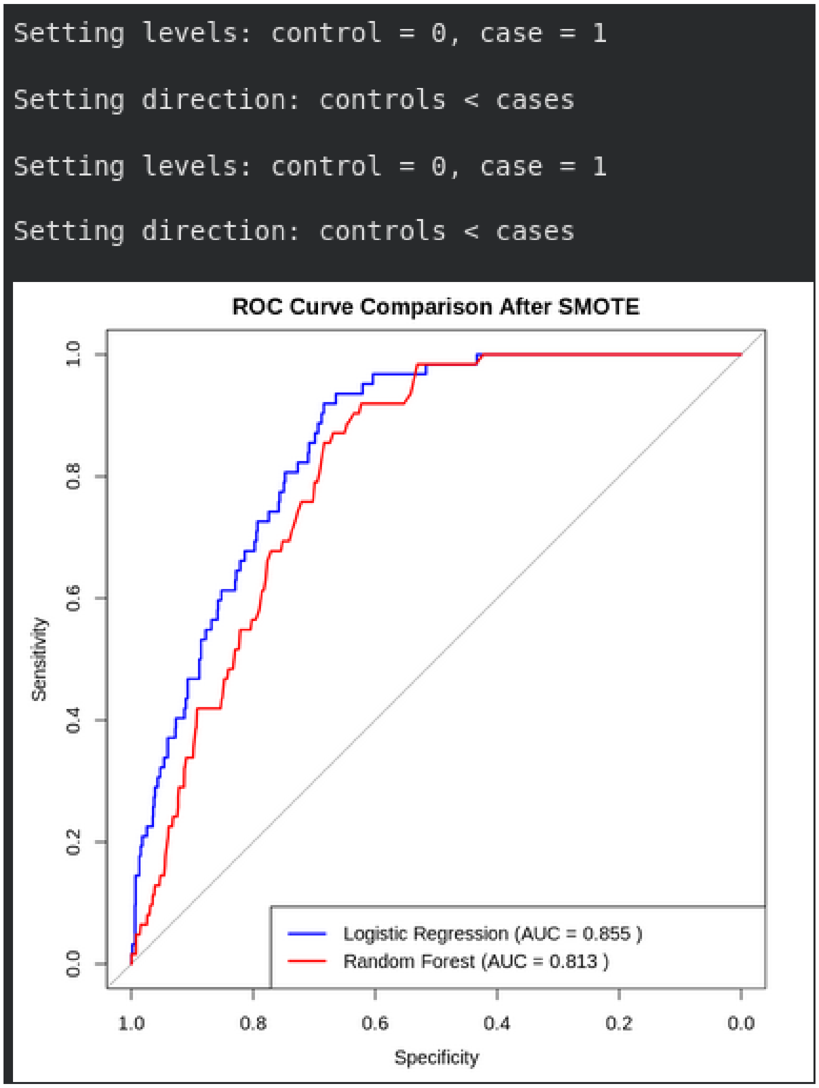


—

3.13 ROC Curves After SMOTE

```
# Logistic Regression ROC
roc_log_smote <- roc(
  test_smote$stroke,
  log_smote_probs
)

# Random Forest ROC
rf_smote_probs <- predict(
  rf_smote_model,
  test_smote,
  type = "prob"
)[, 2]

roc_rf_smote <- roc(
  test_smote$stroke,
  rf_smote_probs
)

# Plot ROC curves
plot(
  roc_log_smote,
  col = "blue",
  lwd = 2,
  main = "ROC Curve Comparison After SMOTE"
)

lines(
  roc_rf_smote,
  col = "red",
  lwd = 2
)

legend(
  "bottomright",
  legend = c(
    paste("Logistic Regression (AUC =", round(auc(roc_log_smote), 3), ")"),
    paste("Random Forest (AUC =", round(auc(roc_rf_smote), 3), ")")
  ),
  col = c("blue", "red"),
  lwd = 2
)
```

Output



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

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

ROC Curve Comparison After SMOTE
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression (AUC = 0.855)
Random Forest (AUC = 0.813)

—

3.14 Dashboard Creation

```
library(tidyverse)
library(cowplot)
library(pROC)

stroke_data$stroke <- factor(
  stroke_data$stroke,
  levels = c(0,1),
  labels = c("No Stroke","Stroke")
)

# Overview Plots

# a. Stroke prevalence (pie)
p1 <- stroke_data %>%
  count(stroke) %>%
  ggplot(aes(x="", y=n, fill=stroke)) +
  geom_col(width=1) +
  coord_polar("y") +
  labs(title="Stroke Prevalence in Population", fill="Stroke Status") +
  theme_void() +
  theme(plot.title = element_text(hjust=0.5, face="bold"))

# b. Stroke by age group
p2 <- stroke_data %>%
  count(age_group, stroke) %>%
  ggplot(aes(age_group, n, fill=stroke)) +
  geom_col(position="dodge") +
  labs(title="Stroke Cases by Age Group",
       x="Age Group", y="Count", fill="Stroke Status") +
  theme_minimal() +
  theme(plot.title = element_text(hjust=0.5, face="bold"))

# Risk Factor Plots

# a. Glucose
p3 <- ggplot(stroke_data,
             aes(x=stroke, y=avg_glucose_level, fill=stroke)) +
  geom_boxplot() +
  labs(title="Average Glucose Level by Stroke Status",
       x="Stroke", y="Avg Glucose (mg/dL)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust=0.5, face="bold"))
```

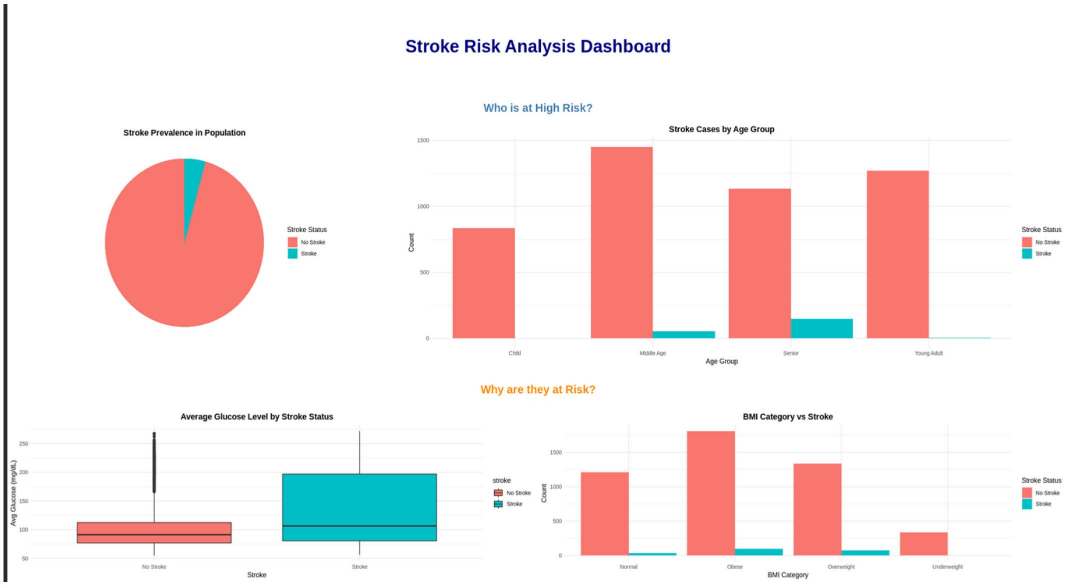
```
# b. BMI
p4 <- stroke_data %>%
  count(bmi_category, stroke) %>%
  ggplot(aes(bmi_category, n, fill=stroke)) +
  geom_col(position="dodge") +
  labs(title="BMI Category vs Stroke",
       x="BMI Category", y="Count", fill="Stroke Status") +
  theme_minimal() +
  theme(plot.title = element_text(hjust=0.5, face="bold"))

# c. Hypertension
if("hypertension" %in% colnames(stroke_data)){
  p5 <- stroke_data %>%
    count(hypertension, stroke) %>%
    ggplot(aes(factor(hypertension), n, fill=stroke)) +
    geom_col(position="dodge") +
    labs(title="Hypertension vs Stroke",
         x="Hypertension (0=No,1=Yes)",
         y="Count", fill="Stroke Status") +
    theme_minimal() +
    theme(plot.title = element_text(hjust=0.5, face="bold"))
} else { p5 <- NULL }

# d. Heart disease
if("heart_disease" %in% colnames(stroke_data)){
  p6 <- stroke_data %>%
    count(heart_disease, stroke) %>%
    ggplot(aes(factor(heart_disease), n, fill=stroke)) +
    geom_col(position="dodge") +
    labs(title="Heart Disease vs Stroke",
         x="Heart Disease (0=No,1=Yes)",
         y="Count", fill="Stroke Status") +
    theme_minimal() +
    theme(plot.title = element_text(hjust=0.5, face="bold"))
} else { p6 <- NULL }
```

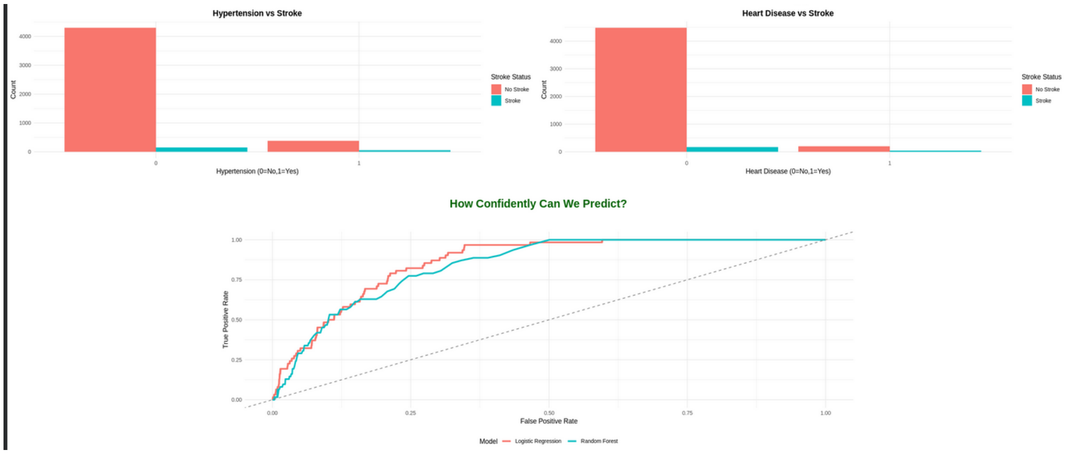
```
# ROC Plot

library(ggplot2)

log_df <- data.frame(
  fpr = 1 - log_roc$specificities,
  tpr = log_roc$sensitivities,
  model = "Logistic Regression"
)

rf_df <- data.frame(
  fpr = 1 - rf_roc$specificities,
  tpr = rf_roc$sensitivities,
  model = "Random Forest"
)

roc_gg <- ggplot(
  rbind(log_df, rf_df),
  aes(x=fpr, y=tpr, color=model)
) +
  geom_line(linewidth=1.2) +
  geom_abline(linetype="dashed", color="grey50") +
  labs(x="False Positive Rate",
       y="True Positive Rate",
       color="Model") +
  theme_minimal() +
  theme(legend.position="bottom")

# Arrange Dashboard
options(repr.plot.width=24, repr.plot.height=22)

who_title <- ggdraw() +
  draw_label("Who is at High Risk?",
             fontface="bold", size=18,
             color="steelblue", hjust=0.5)
```

```
top_row <- plot_grid(p1, p2, ncol=2,
                     rel_widths=c(1.2,2))

who_section <- plot_grid(
  who_title, top_row,
  ncol=1, rel_heights=c(0.12,1)
)

why_title <- ggdraw() +
  draw_label("Why are they at Risk?",
             fontface="bold", size=18,
             color="darkorange", hjust=0.5)

risk_plots <- list(p3, p4, p5, p6)
risk_plots <- risk_plots[!sapply(risk_plots, is.null)]

middle_row <- plot_grid(plotlist=risk_plots, ncol=2)

why_section <- plot_grid(
  why_title, middle_row,
  ncol=1, rel_heights=c(0.12,1)
)

confidence_title <- ggdraw() +
  draw_label("How Confidently Can We Predict?",
             fontface="bold", size=18,
             color="darkgreen", hjust=0.5)

roc_section <- plot_grid(
  confidence_title,
  plot_grid(NULL, roc_gg, NULL,
            ncol=3, rel_widths=c(0.2,0.6,0.2)),
  ncol=1, rel_heights=c(0.18,0.82)
)
```

```
main_dashboard <- plot_grid(
  who_section, why_section, roc_section,
  ncol=1, rel_heights=c(1,1.4,1)
)

dashboard_title <- ggdraw() +
  draw_label("Stroke Risk Analysis Dashboard",
             fontface="bold", size=28,
             color="navy", hjust=0.5)

final_dashboard <- plot_grid(
  dashboard_title, main_dashboard,
  ncol=1, rel_heights=c(0.1,1)
)

print(final_dashboard)
```

Output





```
ggsave(
  "stroke_dashboard_full.png",
  final_dashboard,
  width=20, height=12, dpi=300, bg="white"
)
```

Chapter 4

Conclusion

This project presented a complete end-to-end stroke risk analysis pipeline, progressing from data preparation and exploratory analysis to predictive modeling and visual analytics. Multiple machine learning models were developed and evaluated to assess their ability to distinguish between stroke and non-stroke cases, with performance measured using confusion matrices, ROC curves, and AUC metrics.

Traditional logistic regression provided a transparent and interpretable baseline model, while the random forest classifier demonstrated stronger predictive performance by capturing nonlinear relationships and complex interactions among clinical and demographic features. The application of SMOTE effectively addressed class imbalance, leading to improved sensitivity toward stroke cases and more stable model behavior.

The final dashboard consolidated population-level trends, key risk factors, and model performance into a single visual interface. This enabled intuitive interpretation of who is at higher risk, why those risks occur, and how confidently the models can predict stroke outcomes. Such an integrated analytical workflow supports data-driven decision-making and highlights the potential of machine learning techniques in healthcare risk assessment.

Overall, the study demonstrates that combining robust preprocessing, balanced modeling strategies, and clear visual communication can significantly enhance the practical value of predictive analytics in clinical and public health contexts.