DSB#12 _R Markdown_Mini Project

Tae T

2025-10-29

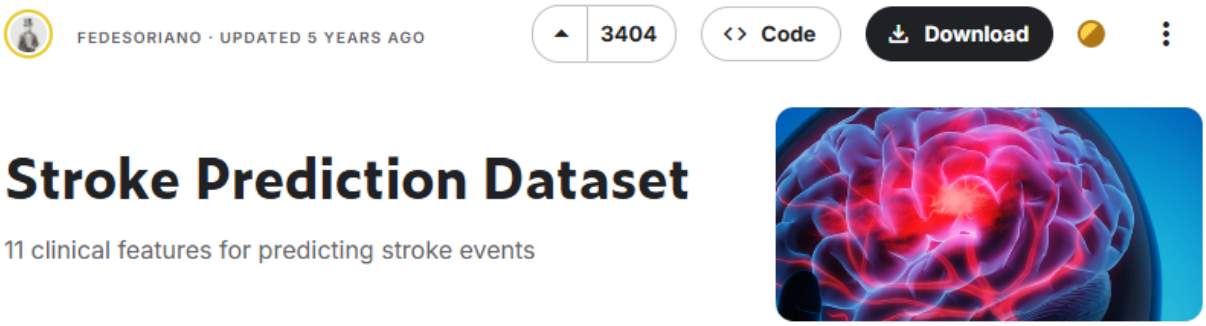
DSB#12 _R Markdown_Mini Project

Hi everyone, welcome to my mini project for R Markdown!

In this project, I will give you all a tour of the Stroke Prediction Dataset and share some meaningful insights from my analysis.

Before we begin, please visit Kaggle and download the dataset using the link provided below. Then download the pdf file.

kaggle.comhttps://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset



This dataset is used to predict the likelyhood of the patient to get stoke base on the given infromation below:

Attribute Information

1) id: unique identifier
2) gender: "Male", "Female" or "Other"
3) age: age of the patient
4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5) heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6) ever_married: "No" or "Yes"
7) work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
8) Residence_type: "Rural" or "Urban"
9) avg_glucose_level: average glucose level in blood
10) bmi: body mass index
11) smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
12) stroke: 1 if the patient had a stroke or 0 if not
*Note: "Unknown" in smoking_status means that the information is unavailable for this patient

Install and load libraries.

Before we begin with the analysis, let's explore the data structure and import the necessary libraries.

1. Install and load libraries.

```
## install packages and load library
## for packages install one time
#install.packages("tidyverse") ## metapackage for r
#install.packages("janitor") ## data cleaning
#install.packages("ggplot2") ## data viz
#install.packages("dplyr") ## data transformation

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(ggplot2)
library(dplyr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

Data Transfromation

2. Import data and perfrom EDA

For this time, I use Google Colab to run this analysis. In order to read the CSV file, you need to upload it to Google Colab first. Then, I read cvs file using `read_cvs()` and clean column name with `clean_name()` from library janitor. As you can see, the dataset contains 5,110 rows and 12 columns. We will continue exploring it further.

```r
## read csv file from Stroke Prediction Dataset
df <- read_csv("healthcare-dataset-stroke-data.csv") %>%
  clean_names()
```

```
## 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.
```

First, I will check the overview of the dataframe by using the `glimpse()` function, which provides a quick and concise summary of a data frame or tibble.

```r
# check overview of data
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~
```

Next, I will convert the dataframe into a tibble using `tibble(df)` to improve readability and data presentation.

```r
## change df to tibble
tibble(df) # make tabel easy to read
```

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

Next, I will examine the `head()`, `tail()`, and missing values to ensure that the dataset is clean and ready for analysis.

```r
## head and tail
head(df, 5)
```

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

```r
tail(df, 5)
```

```
## # A tibble: 5 x 12
##      id gender   age hypertension heart_disease ever_married work_type
##   <dbl> <chr>  <dbl>        <dbl>         <dbl> <chr>        <chr>
## 1 18234 Female    80            1             0 Yes          Private
## 2 44873 Female    81            0             0 Yes          Self-employed
## 3 19723 Female    35            0             0 Yes          Self-employed
## 4 37544 Male      51            0             0 Yes          Private
## 5 44679 Female    44            0             0 Yes          Govt_job
## # i 5 more variables: residence_type <chr>, avg_glucose_level <dbl>, bmi <chr>,
## #   smoking_status <chr>, stroke <dbl>
```

Identify missing value

I used `sum()` and `is.na(df)` to calculate the total number of missing values in the dataframe. As you can see from the result, it shows zero missing values. However, based on the previous `head()` and `tail()` outputs, I noticed that there are some "N/A" entries in the table. Let's investigate this further.

```r
## Identify missing value
sum(is.na(df))
```

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

```r
## Identify "N/A"
df %>%
  summarise(across(everything(), ~ sum(. == "N/A", na.rm = TRUE))) %>%
  ## pivot table for readability
  pivot_longer(everything(),
               names_to = "column",
               values_to = "na_count") %>%
  arrange(desc(na_count))
```

```
## # A tibble: 12 x 2
##    column            na_count
##    <chr>                <int>
##  1 bmi                    201
##  2 id                       0
##  3 gender                   0
##  4 age                      0
##  5 hypertension             0
##  6 heart_disease            0
##  7 ever_married             0
##  8 work_type                0
##  9 residence_type           0
## 10 avg_glucose_level        0
## 11 smoking_status           0
## 12 stroke                   0
```

Finding Insights

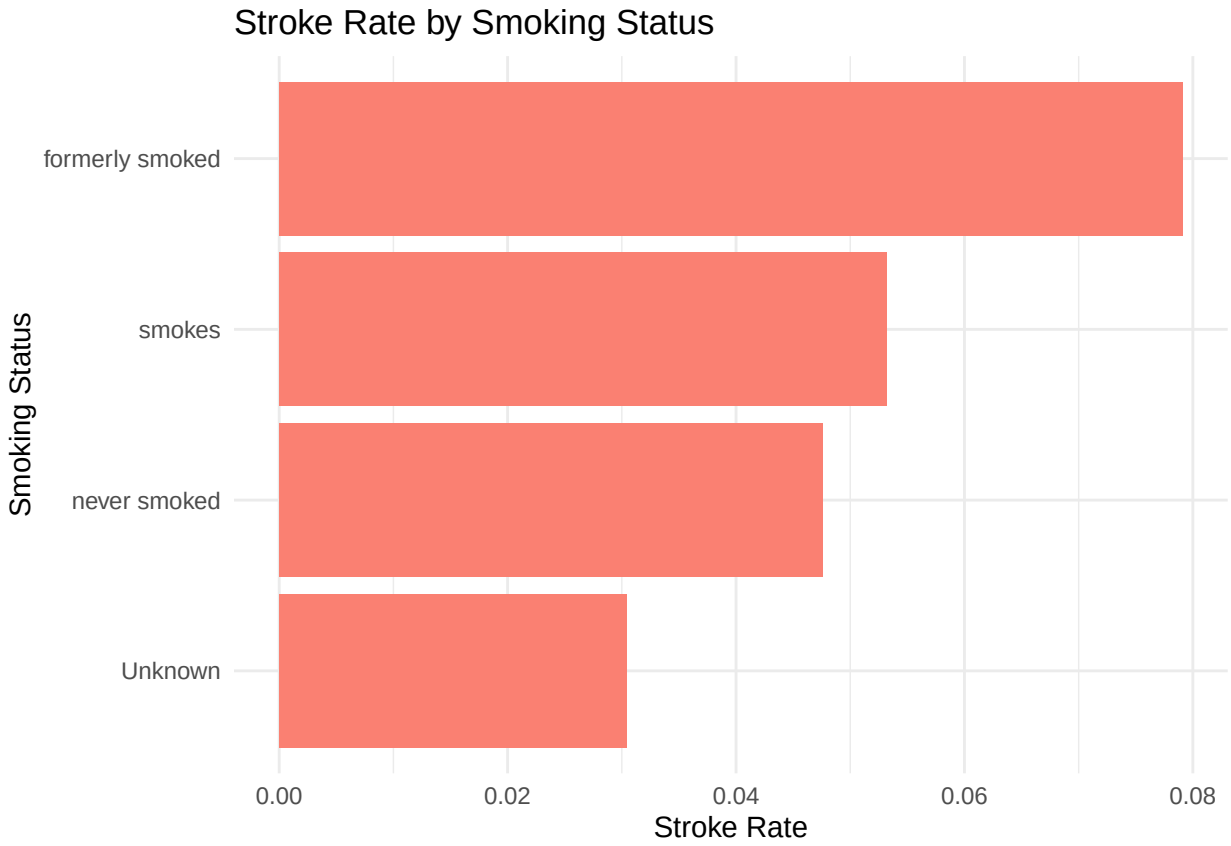
Moving on to the next section, I will dive deeper into the dataset to look for meaningful insights and identify factors that may contribute to a person having a stroke. These factors include age, average glucose level, BMI, hypertension, heart disease, smoking status, and others.

1. How stroke rate differs across smoking categories?

```r
## How stroke rate differs across smoking categories?
df %>%
    ## mutate column, convert chr to factor and set NA with "Unknown"
    mutate(smoking_status = fct_explicit_na(as.factor(smoking_status), na_level = "Unknown")) %>%
    group_by(smoking_status) %>%

    ## summarise stroke_rate and igmore missing value
    summarise(stroke_rate = mean(stroke, na.rm=TRUE)) %>%

    ## plot graph
    ggplot(aes

    ## reorders the categories by stroke rate
    (fct_reorder (smoking_status, stroke_rate), stroke_rate)) +
    geom_col(fill = "salmon") +
    coord_flip() + ## rotate bar chart
```

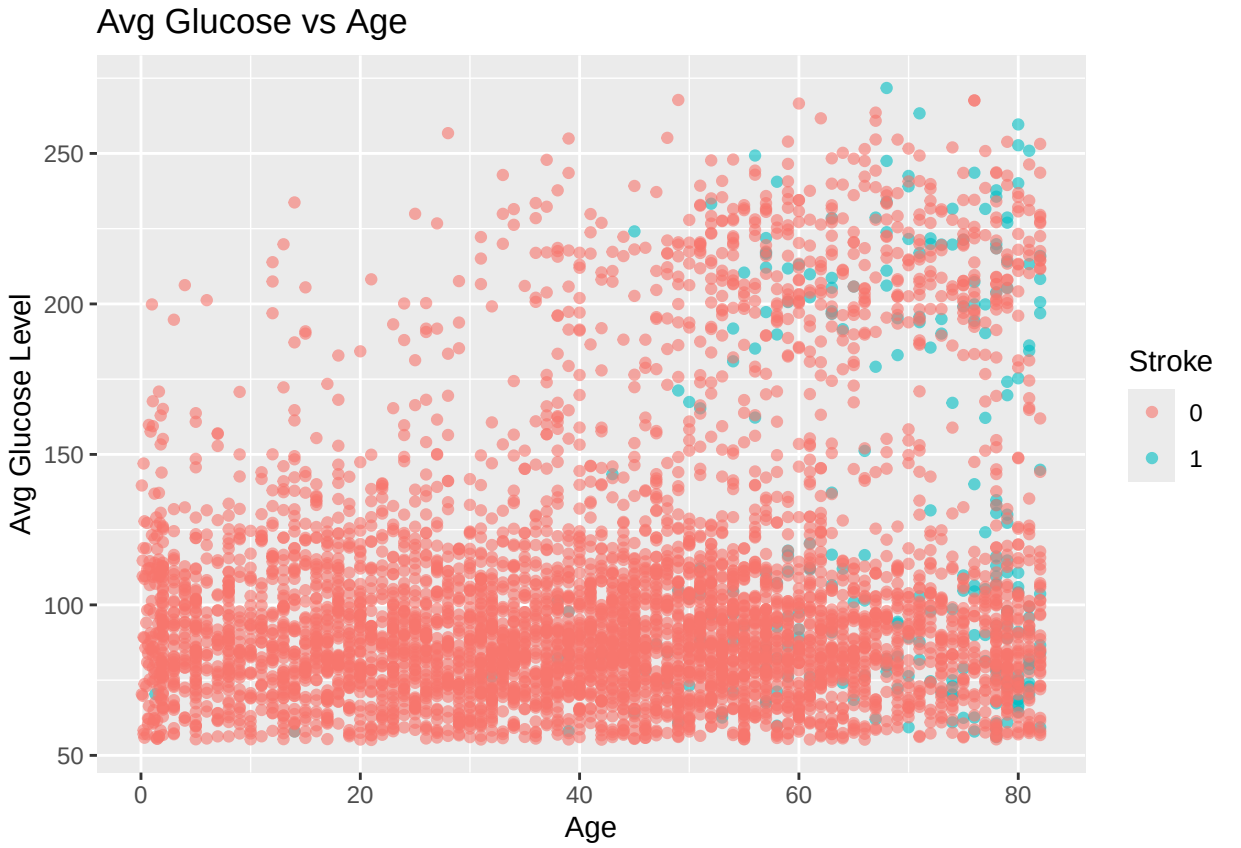
```
    ## set chart detail
    labs(title="Stroke Rate by Smoking Status", x="Smoking Status", y="Stroke Rate")+
    theme_minimal()
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `smoking_status = fct_explicit_na(as.factor(smoking_status),
##   na_level = "Unknown")`.
## Caused by warning:
## ! `fct_explicit_na()` was deprecated in forcats 1.0.0.
## i Please use `fct_na_value_to_level()` instead.
```



2. Older people with higher average glucose levels have a higher risk of having a stroke.

```
ggplot(df, aes(age,
    avg_glucose_level,
    ## set color for stroke and non-stroke
    color=factor(stroke))) +
  geom_point(alpha =.6) +
  labs(title="Avg Glucose vs Age", x="Age", y="Avg Glucose Level", color="Stroke")
```



3. Individuals with higher BMI levels are at an increased risk of experiencing a stroke.

```
## Individuals with higher BMI levels are at an increased risk of experiencing a stroke.

## setup & cleaning
# convert stroke to numeric and keep BMI as numeric
df <- df %>%
  mutate(
    stroke_num = as.numeric(as.character(stroke)),  # if stroke was a factor
    bmi = as.numeric(bmi)
  )
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `bmi = as.numeric(bmi)`.
## Caused by warning:
## ! NAs introduced by coercion
```

```
# drop rows with missing BMI
df_bmi <- df %>%
  filter(!is.na(bmi), !is.na(stroke_num))
```

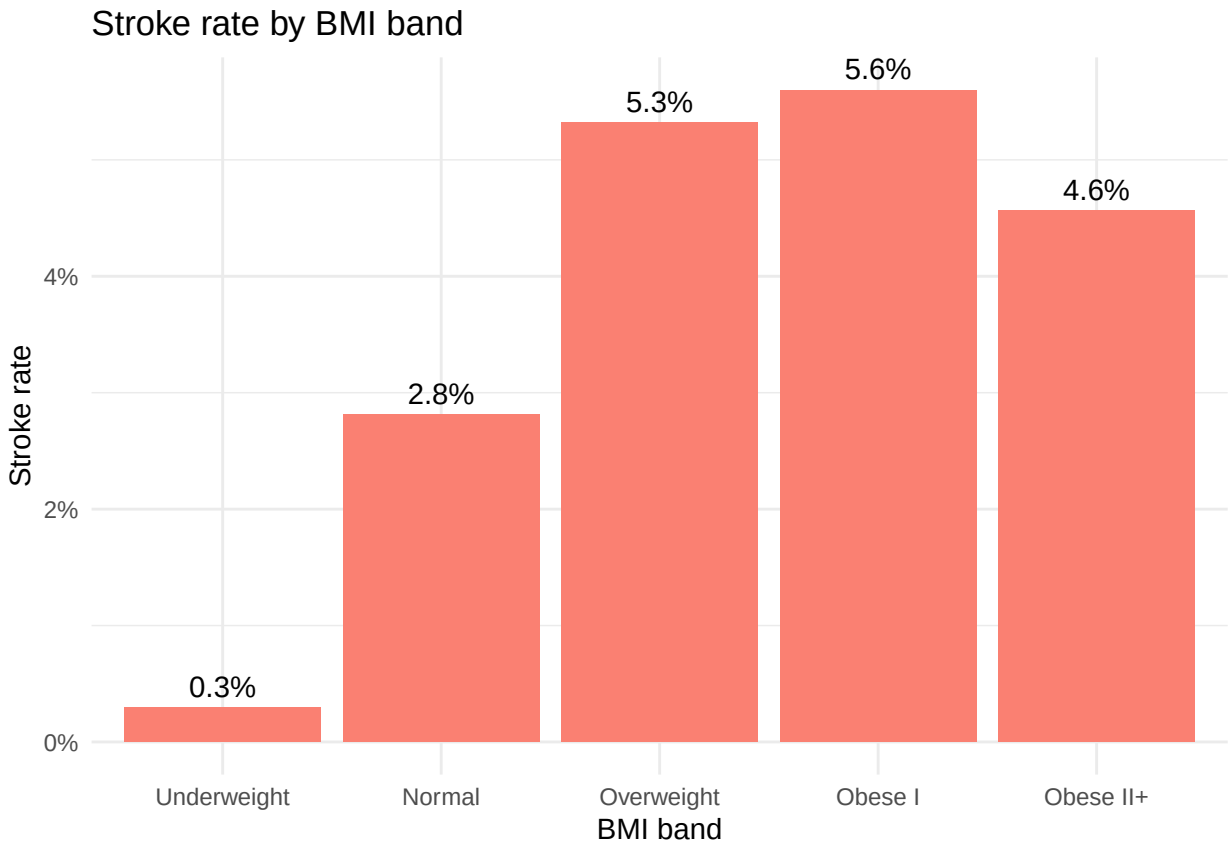
```
## segment individual bmi into band
df_bmi <- df_bmi %>%
  mutate(bmi_band = cut( ## divide into interval and convert to vector
    bmi,
    breaks = c(-Inf, 18.5, 25, 30, 35, Inf), ## +- infinity
    labels = c("Underweight","Normal","Overweight","Obese I","Obese II+"),
    right = FALSE
```

```
  ))
```

```
# Stroke rate by band
rate_tbl <- df_bmi%>%
  group_by(bmi_band) %>%
  summarise(
    n = n(),
    strokes = sum(stroke_num),
    stroke_rate = strokes / n
  ) %>%
  arrange(bmi_band)

print (rate_tbl)
```

```
## # A tibble: 5 x 4
##   bmi_band        n strokes stroke_rate
##   <fct>       <int>   <dbl>       <dbl>
## 1 Underweight   337       1     0.00297
## 2 Normal       1243      35     0.0282
## 3 Overweight   1409      75     0.0532
## 4 Obese I      1000      56     0.056
## 5 Obese II+     920      42     0.0457
```

```
# Bar chart of stroke rates
ggplot(rate_tbl, aes(x = bmi_band, y = stroke_rate)) +
  geom_col(fill = "salmon") +
  geom_text(aes(label = scales::percent(stroke_rate, accuracy = 0.1)),
            vjust = -0.5, size = 4) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Stroke rate by BMI band", x = "BMI band", y = "Stroke rate") +
  theme_minimal()
```



R Markdown

Finally, I will wrap up my project and create a report using R Markdown in Posit Cloud.

*** This is the end of the report***