Explodanary Data analysis stroke prediction dataset

Pascal Visser

2022-09-14

Load libraries

```
# load libraries
library(tidyverse)
library(naniar)
library(readr)
library(ggplot2)
library(lemon)
library(knitr)
library(pander)
library(cowplot)
```

1. Intro

A stroke is a life-threatening medical condition, where there is no or poor blood flow to (parts) the brain. This causes parts of the brain to stop functioning and over minutes time die off. A stroke or brain attack is labelled as the second leading cause of death worldwide and is responsible for an annual mortality of about 5.5 million.

This dataset consists of people that had a stroke and people that have not. The variables are values that indicate there live style and environment. The goal of this research is to calculate the chance of a stroke based of there lifestyle. This can be accomplished by machine learning. This gives the following research question:

Is it possible to produce an accurate algorithm with machine learning, to calculate the risk of a stroke based on life style variables of people with stroke history and people that never had a stroke?

2. Dataset

The data is as said, lifestyle information of people with stroke history or not. In this section the data will be talked trough and the variables explained. First the data is loaded in:

```
# load in the data
strokedata <- read.csv("Data/Stroke_dataset.csv", header = T)
kable(strokedata[1:6, 1:6], format = 'simple', caption = "Head of data")
```

Table 1: Head of data

id	gender	age	hypertension	heart_disease	ever_married
9046	Male	67	0	1	Yes
51676	Female	61	0	0	Yes
31112	Male	80	0	1	Yes
60182	Female	49	0	0	Yes
1665	Female	79	1	0	Yes
56669	Male	81	0	0	Yes

```
cat("The dataset is", dim(strokedata)[1], "rows long, and has", dim(strokedata)[2], "columns")
```

```
## The dataset is 5110 rows long, and has 12 columns
```

Above, the first six rows of the data is shown. Also is given how many records there are. The codebook will clarify the variables and there values. The codebook will be loaded in:

```
# load codebook
codebook <- read.table("Data/codebook.txt", header = T, sep = ';')
kable(codebook, format = 'pipe', caption = "Codebook of stroke data")
```

Table 2: Codebook of stroke data

Column	Description	value	datatype
id	unique identifier of patient	number	int
gender	male of female	sex	chr
age	age of patient	number	dbl
hypertension	suffering from hypertension?	binairy	int
heart_disease	suffering from heart disease?	binary	int
ever_married	married or married in the past?	binary	chr
work_type	sort of work	string	chr
Residence_type	type of residence	string	chr
avg_glucose_level	average level of glucose	number	dbl
bmi	body mass index	number	chr
smoking_status	smoking history	string	chr
stroke	stroke history	binary	int

The codebook gives a detailed description the columns of the dataset and column content about the value/datatypes. This is necessary to understand the data.

Now that the data is loaded in, there can be looked at the negatives of the data. The biggest flaws are the datatypes of the columns that are not equal. For example: the columns age is the type double instead of integer. The column BMI is character, while a double datatype is much more logical. Also the columns hypertension and hearth disease are a yes or no question, in these columns the yes or no is represented by 0 or 1. The columns of marriage history is a yes or no question too, but here yes or no is represented by text. This also needs modifying.

Additionally is it good to know how many NA's are present in the data. These missing values can screw the data, also values like 'unknown' can cast the data to the character data type.

```r
# check unique values of character values
cat("Gender:")
```

```
## Gender:
```

```r
unique(strokedata$gender)
```

```
## [1] "Male"   "Female" "Other"
```

```r
cat("Married:")
```

```
## Married:
```

```r
unique(strokedata$ever_married)
```

```
## [1] "Yes" "No"
```

```r
cat("Work type:")
```

```
## Work type:
```

```r
unique(strokedata$work_type)
```

```
## [1] "Private"       "Self-employed" "Govt_job"      "children"
## [5] "Never_worked"
```

```r
cat("Residence type:")
```

```
## Residence type:
```

```r
unique(strokedata$Residence_type)
```

```
## [1] "Urban" "Rural"
```

```r
cat("Smoking:")
```

```
## Smoking:
```

```r
unique(strokedata$smoking_status)
```

```
## [1] "formerly smoked" "never smoked"    "smokes"          "Unknown"
```

Above reveals the Smoking has a value “Unknown”. This need to be addressed in the modifying section. Also the column Gender has one other record. For the N/A the scan function of the naniar package is used.

```r
# scan for missing values
nas <- miss_scan_count(data = strokedata, search = list("N/A", "Unknown"))
kable(nas, format = 'pipe', caption = "Number of missing values per variable")
```

Table 3: Number of missing values per variable

Variable	n
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

The column bmi is responsible for 201 missing values, these missing values are likely responsible for the column to be parsed as character datatype. the column smoking_status has 1544 missing values. 1544 values is a big number and need to be dealed with in the modifying.

3 Dataset modifying

As talked in the previous section, some columns types makes no sense, these types need to be changed to more sensible types. The columns that will be converted are:

- age, dbl -> int
- bmi, chr -> dbl

and additionally the contents of the column ever_married will be converted to binary, and the smoking_status unknown will be cleared.

First the column age will be cast to integer, because a double type with decimals makes no sense in age, secondly, the column bmi will be cast to double, to keep the decimal precision:

```r
# cast columns to another type
strokedata <- transform(strokedata, age = as.integer(age))
strokedata <- transform(strokedata, bmi = as.double(bmi))
```

```
## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs introduced
## by coercion
```

now the ever_married column need to be changed to binary and a integer datatype:

```r
# replace yes and no by 1 or 0
strokedata$ever_married[strokedata$ever_married == "Yes"] <- "1"
strokedata$ever_married[strokedata$ever_married == "No"] <- "0"
strokedata <- transform(strokedata, ever_married = as.integer(ever_married))
```

The record with gender 'other' will be removed for no making the machine learning algorithms confused.

```r
# remove record where gender is other
strokedata <-strokedata[!(strokedata$gender == "Other"),]
```

Now what is left are the missing values, bmi has N/A and smoking status has "unknown". For bmi the Na's will be replaced with the mean of the column:

```r
strokedata$bmi[is.na(strokedata$bmi)] <- mean(strokedata$bmi, na.rm = T)
strokedata$bmi <- round(strokedata$bmi, digits = 1)
```

The smoking status 'unknown' is harder to replace because the variables are character, first we need to know how many records there are of the column:

```r
# Count the unique variables in the gender column
table(strokedata$smoking_status)
```

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

Based on these numbers, we can calculate the probability of the occurring smoke statuses.

```r
# Calculate the probability of formerly smoker, current smokers and non-smokers given that there's only
prob.Formerly <- 885 / (885 + 1892 + 789)
prob.Never <- 1892 / (885 + 1892 + 789)
prob.Smoke <- 789 / (885 + 1892 + 789)
```

With the probabilities, we can replace the unknowns based on there weight in the column.

```r
# Replacing 'Unknown' in smoking_status by the other 3 variables according to their weightage
strokedata$rand <- runif(nrow(strokedata))
strokedata <- strokedata%>%mutate(Probability = ifelse(rand <= prob.Formerly, "formerly smoked", ifelse
strokedata <- strokedata%>%mutate(smoking.status = ifelse(smoking_status == "Unknown", Probability, smo
# View the new Smoking Status column's unique values and their counts
table(strokedata$smoking.status)
```

```
##
## formerly smoked    never smoked          smokes
##            1248            2715            1146
```

```r
# Remove columns that are not needed
health <- subset(strokedata, select = -c(rand,Probability,smoking_status))
# revise the column name of smoking status
colnames(health)[12] <- "smoking_status"
# 'health' is the final modified dataset which will be used for the EDA section below.
```

Now we have a column without the unknowns, but with increased other variables based on there weight.

4. Data analysis

In this part, the modified data is visualised into useful graphs. These graph will tell about the distribution of variables and the relations to each other.

4.1 Summary

First the summary of the values numeric values, we have three columns with important numeric values, age, glucose level and BMI. The summary of these variables tells about the distribution of values and possible outliers

```
#summary of subset data
pander(summary(health[c(3,9:10)],), caption = "Summary of numeric values")
```

Table 4: Summary of numeric values Looking at age, the minimum value is 0, which is a bit odd. maybe this is a unknown. Glucose looks normal, but BMI has a very low minimum, which is also odd.

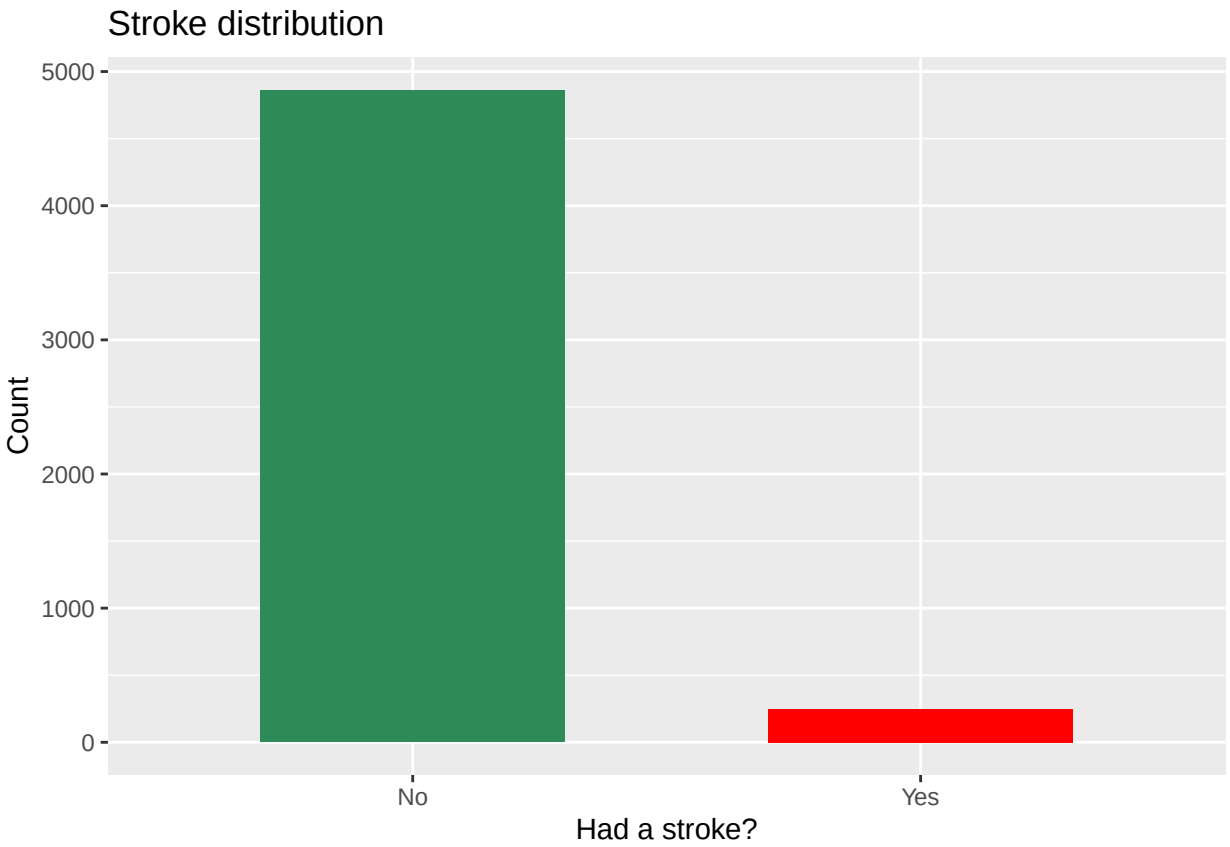
age	avg_glucose_level	bmi
Min. : 0.00	Min. : 55.12	Min. :10.30
1st Qu.:25.00	1st Qu.: 77.24	1st Qu.:23.80
Median :45.00	Median : 91.88	Median :28.40
Mean :43.22	Mean :106.14	Mean :28.89
3rd Qu.:61.00	3rd Qu.:114.09	3rd Qu.:32.80
Max. :82.00	Max. :271.74	Max. :97.60

4.2 Distribution of classifier

The research is about classifying patients changes of stroke based on there lifestyle and health. The figure below is the distribution of the stroke incidents in the data.

```
# Plot stroke distribution
fig1 <- ggplot(health, aes(x=factor(stroke))) +
  geom_bar(width = 0.6, fill = c("seagreen", "red")) +
  labs(title = "Stroke distribution", x = "Had a stroke?", y  = "Count")

# plot the graph
fig1 + scale_x_discrete(breaks=c("0", "1"), labels=c("No", "Yes"))
```

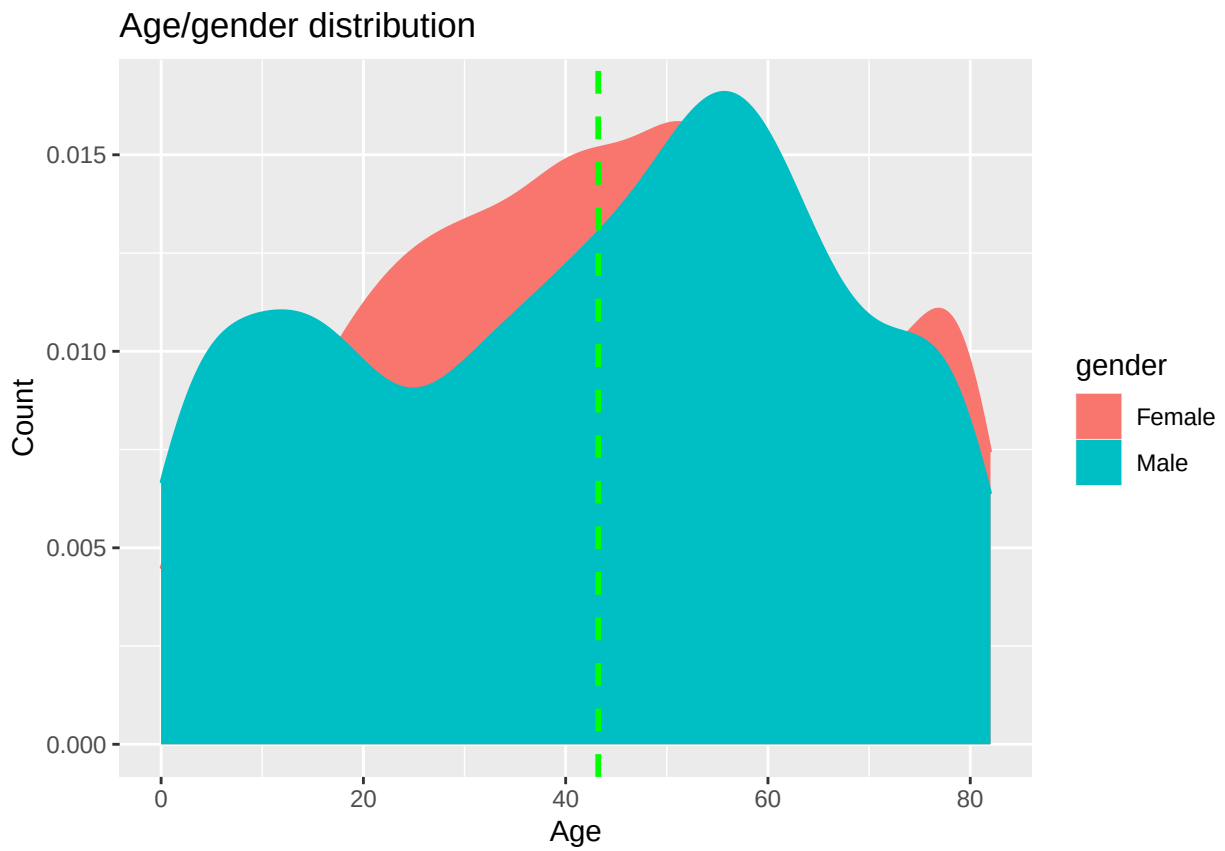


The number of stroke in the dataset is very low. the vast majority of the data didn't had a stroke. this needs to be balanced later.

4.3 Age and gender distribution

There are 5110 people in the dataset from which there is data, the gender and age distribution is as follows:

```
# make age and gender plot
ggplot(health, aes(x=age, fill=gender, color=gender)) +
    geom_density() +
    labs(title = "Age/gender distribution", x = "Age", y  = "Count") +
    geom_vline(aes(xintercept=mean(age)), color = 'green', lwd = 1, linetype = 'dashed')
```



Most people are around 40 - 60 years old, with some more females round the 20- 40 mark.

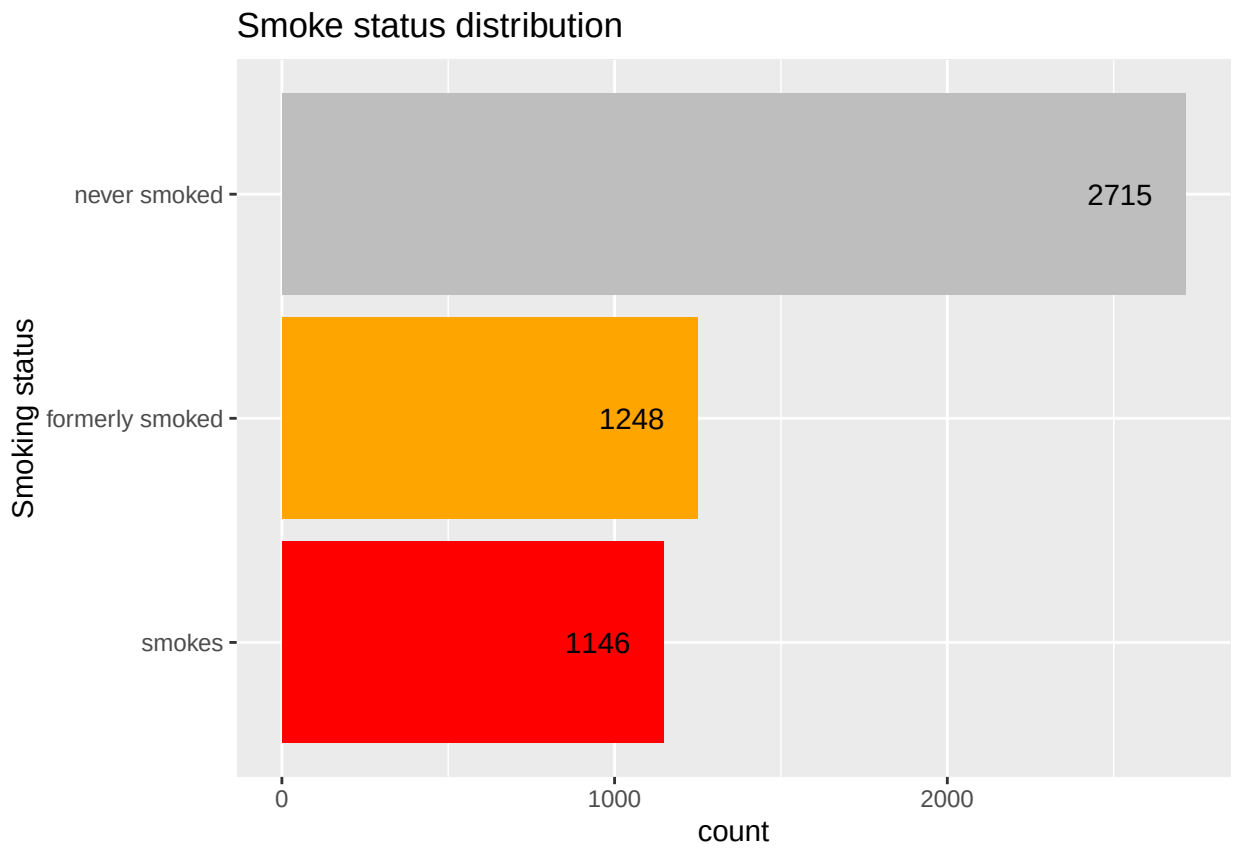
4.4 health related variables

This sections tells about the heath related variables such as, smoke, BMI, hypertension, heart disease and glucose levels

4.4.1 Smoke

Smoking can affect your health in a bad why, therefore it is good to know how many people (formerly) are smoking.

```
health %>%
  group_by(smoking_status) %>%
    summarise(count = length(smoking_status)) %>%
      mutate(smoking_status = factor(smoking_status)) %>%
        ggplot(aes(x = fct_reorder(smoking_status, count), y = count)) +
        geom_col(fill = c("Orange", "gray", "red")) +
        geom_text(aes(label = count, x = smoking_status, y = count), size = 4, hjust = 1.5) +
        coord_flip() +
        labs(x = "Smoking status", title = "Smoke status distribution")
```



Most of the people never have smoke, but a fairly amount of the people smokes or formerly smoked. this can indicated a bad overall health which possible can contribute towards strokes. The term smokes is a bit too general, because it says nothing about how much someone smokes.

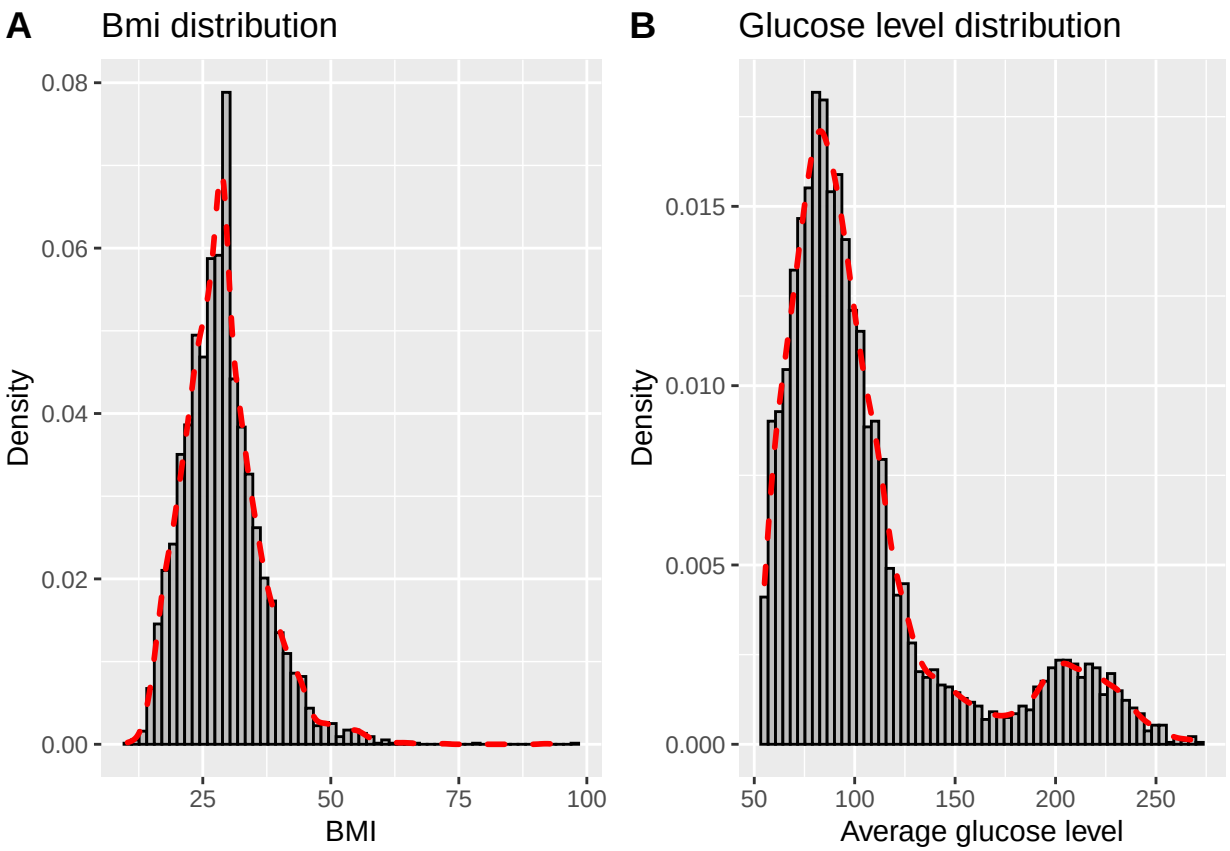
4.4.2 BMI and Glucose levels

BMI and glucose levels are also a good indication of general health.

```
fig2 <- ggplot(health, aes(x = bmi)) +
  geom_histogram(aes(y = ..density..), bins = 60, color = 1, fill = "gray") +
  geom_density(lwd = 1, linetype = 2, color = 'red') +
  labs(title = "Bmi distribution", x = "BMI", y = "Density")

fig3 <- ggplot(health, aes(x = avg_glucose_level)) +
  geom_histogram(aes(y = ..density..), bins = 60, color = 1, fill = "gray") +
  geom_density(lwd = 1, linetype = 2, color = 'red') +
  labs(title = "Glucose level distribution", x = "Average glucose level", y = "Density")

plot_grid(fig2, fig3, labels = "AUTO", rel_widths = c(1,1))
```



The BMI look normally with a bit right skewed. which means the the most people are big. The glucose levels are also right skewed, but this is more worrying. High glucose levels are not safe for health.

4.4.3 Cardiovascular diseases

There are to cardiovascular diseases in the data, which are heart disease and hypertension.

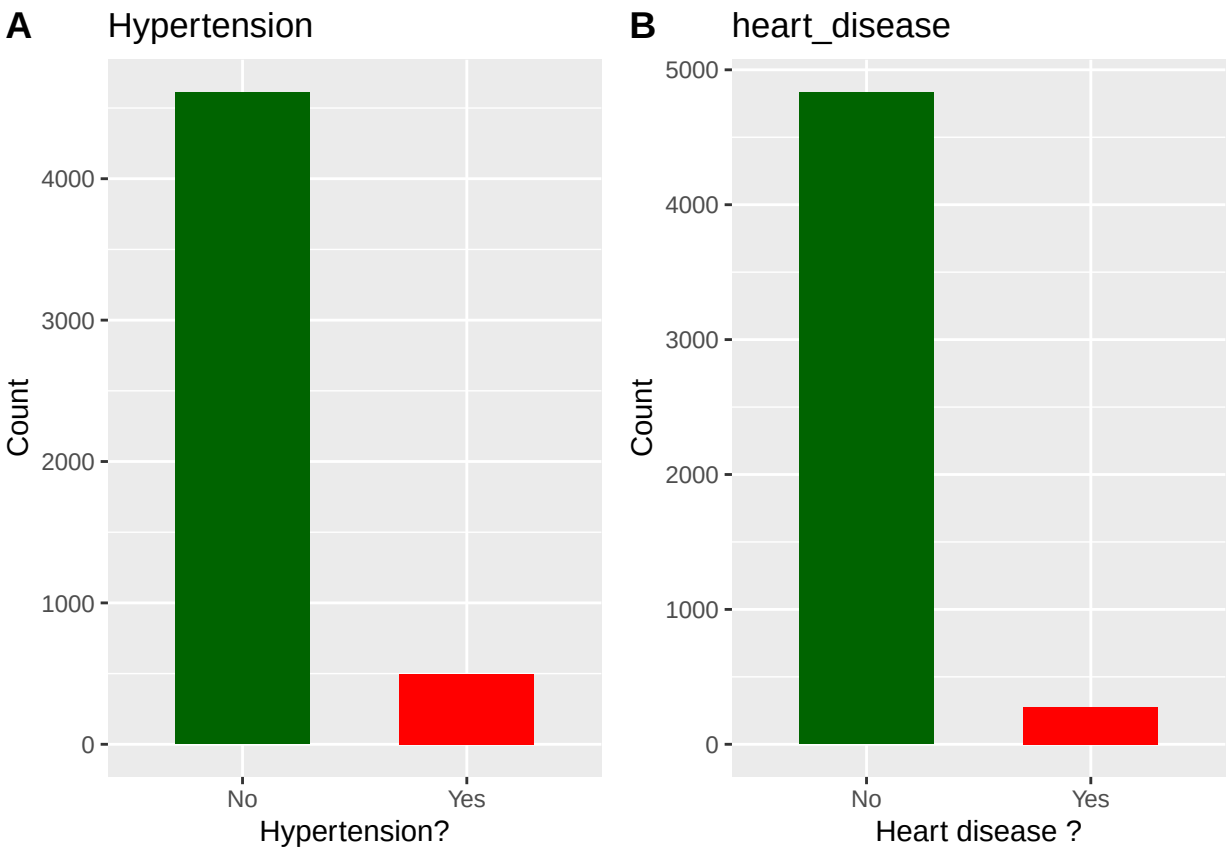
```
fig4 <- ggplot(health, aes(x=factor(hypertension))) +
  geom_bar(width = 0.6, fill = c("darkgreen", "red")) +
  labs(title = "Hypertension", x = "Hypertension?", y  = "Count")

fig4 <- fig4 + scale_x_discrete(breaks=c("0", "1"), labels=c("No", "Yes"))

fig5 <- ggplot(health, aes(x=factor(heart_disease))) +
  geom_bar(width = 0.6, fill = c("darkgreen", "red")) +
  labs(title = "heart_disease", x = "Heart disease ?", y  = "Count")

fig5 <- fig5 + scale_x_discrete(breaks=c("0", "1"), labels=c("No", "Yes"))


plot_grid(fig4, fig5, labels = "AUTO")
```



The cardiovascular diseases are both overwhelming no, this is also imbalanced.S

5. Conclusion

Overall the dataset is good and without very strange numbers. by casting the different types and removing the unknowns the quality of the data was improved. The biggest flaw is the imbalance with the classifier and the hypertension and heart_disease variables.