first_lesson

Matteo Biglioli

5/24/2021

Data

For this work we used the dataset **Stroke Prediction Dataset** retrieved from Kaggle.

We used the dataset 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 relavant information about the patient.

Below the information about the attributes:

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

Load the dataset

Set the current directory as working directory and load the dataset.

```r
setwd(dirname(rstudioapi::getSourceEditorContext()$path))

source("../utils/plots.R")
source("../utils/data_analysis.R")
dataset = read.csv("stroke.csv")
str(dataset)
```

```
## ’data.frame':    5110 obs. of  12 variables:
##  $ id                : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender            : chr  "Male" "Female" "Male" "Female" ...
##  $ age               : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

Data Manipulation

Convert the character in factor.

```r
dataset$id = NULL

dataset$gender = factor(dataset$gender)
dataset$hypertension = factor(dataset$hypertension, levels = c(0,1), labels = c("No", "Yes"))
dataset$heart_disease = factor(dataset$heart_disease, levels = c(0,1), labels = c("No", "Yes"))
dataset$ever_married = factor(dataset$ever_married)
dataset$work_type = factor(dataset$work_type)
dataset$residence_type = factor(dataset$Residence_type)
dataset$Residence_type = NULL
dataset$smoking_status = factor(dataset$smoking_status)
dataset$stroke = factor(dataset$stroke, levels = c(0,1), labels = c("No", "Yes"))
dataset$bmi = as.numeric(dataset$bmi)

dataset = na.omit(dataset)
```

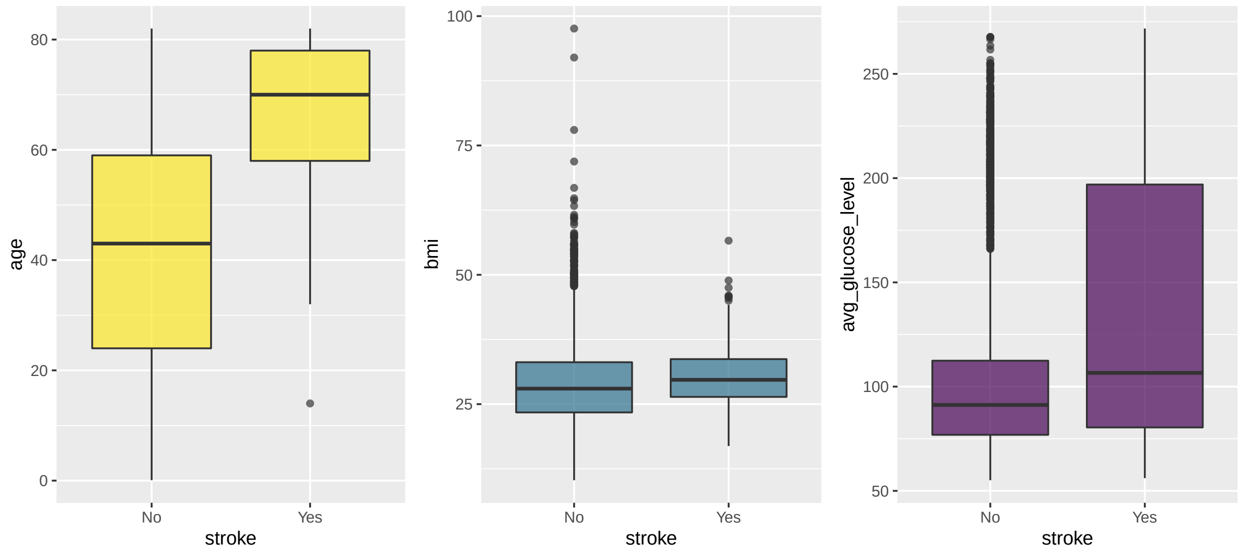
```r
str(dataset)
```

```
## ’data.frame’:    4909 obs. of  11 variables:
##  $ gender           : Factor w/ 3 levels "Female","Male",..: 2 2 1 1 2 2 1 1 1 1 ...
##  $ age              : num  67 80 49 79 81 74 69 78 81 61 ...
##  $ hypertension     : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 2 1 1 2 1 ...
##  $ heart_disease    : Factor w/ 2 levels "No","Yes": 2 2 1 1 1 2 1 1 1 2 ...
##  $ ever_married     : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 1 2 2 2 ...
##  $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 4 4 4 5 4 4 4 4 4 2 ...
##  $ avg_glucose_level: num  229 106 171 174 186 ...
##  $ bmi              : num  36.6 32.5 34.4 24 29 27.4 22.8 24.2 29.7 36.8 ...
##  $ smoking_status   : Factor w/ 4 levels "formerly smoked",..: 1 2 3 2 1 2 2 4 2 3 ...
##  $ stroke           : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ residence_type   : Factor w/ 2 levels "Rural","Urban": 2 1 2 1 2 1 2 2 1 1 ...
##  - attr(*, "na.action")= ’omit’ Named int [1:201] 2 9 14 20 28 30 44 47 51 52 ...
##   ..- attr(*, "names")= chr [1:201] "2" "9" "14" "20" ...
```

Data Visualization

Boxplots to visualize the stroke status in association with **age**, **bmi** and **glucose level**.

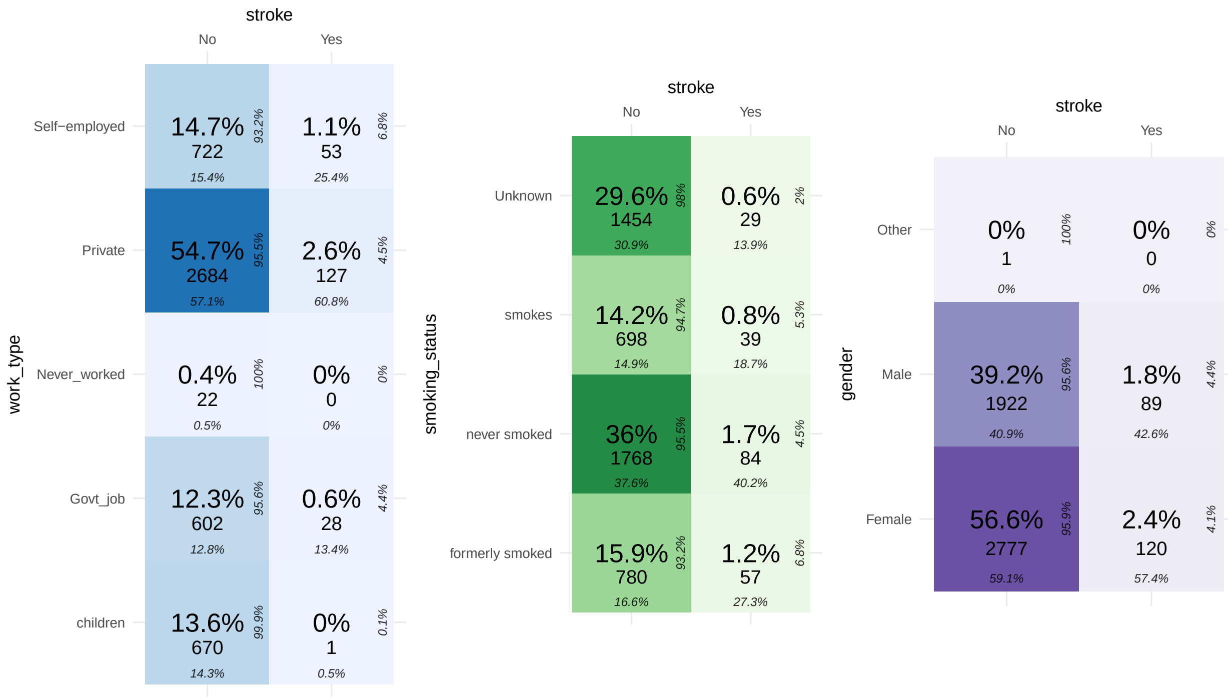
```
grid.arrange(ggplot(dataset, aes(x=stroke ,y=age)) +
               geom_boxplot(fill= "#FDE725FF", alpha= 0.7),
             ggplot(dataset, aes(x=stroke, y=bmi))+
               geom_boxplot(fill= "#2D708EFF", alpha= 0.7),
             ggplot(dataset, aes(x=stroke, y=avg_glucose_level))+
               geom_boxplot(fill= "#440154FF", alpha= 0.7),
             ncol=3)
```



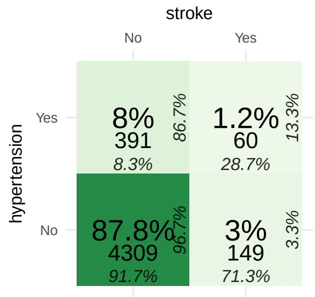
1. **Age** there is a relation with age: older people are more likely to have a stroke
2. **Bmi** there is no evident relation between stroke and bmi
3. **Averagre glucose level** the higher the level of glucose, the higher the relation with stroke

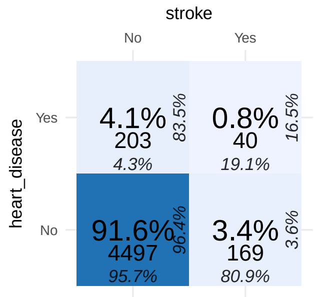
Then we built some matrices to show the relationship between stroke and some **categorical variables** that have different levels.

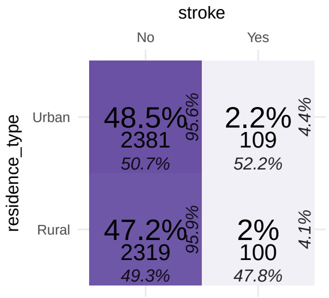
```
source("../utils/plots.R")
#,fig.height=12

grid.arrange(factors_plot(tidy(table(dataset %>% select(stroke,work_type))), palette='Blues',
                          font_count_size=4, font_normalized_size=5.1, font_percentages_size=2.5,
                          font_categories_size=10),
             factors_plot(tidy(table(dataset %>% select(stroke,smoking_status))), palette='Greens',
                          font_count_size=4, font_normalized_size=5.1, font_percentages_size=2.5,
                          font_categories_size=10),
             factors_plot(tidy(table(dataset %>% select(stroke,gender))), palette='Purples',
                          font_count_size=4, font_normalized_size=5.1, font_percentages_size=2.5,
                          font_categories_size=10),
             ncol=3, nrow=1)
```

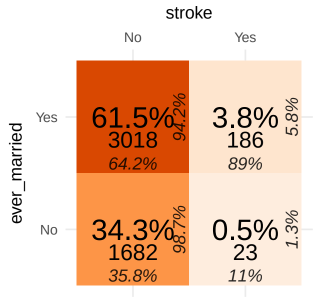


```
grid.arrange(factors_plot(tidy(table(dataset %>% select(stroke,hypertension))), palette='Greens',
                         font_count_size=4.5, font_normalized_size=6, font_percentages_size=3.5,
                         font_categories_size=10),
            factors_plot(tidy(table(dataset %>% select(stroke,heart_disease))), palette='Blues',
                         font_count_size=4.5, font_normalized_size=6, font_percentages_size=3.5,
                         font_categories_size=10),
            factors_plot(tidy(table(dataset %>% select(stroke,residence_type))), palette='Purples',
                         font_count_size=4.5, font_normalized_size=6, font_percentages_size=3.5,
                         font_categories_size=10),
            factors_plot(tidy(table(dataset %>% select(stroke,ever_married))), palette='Oranges',
                         font_count_size=4.5, font_normalized_size=6, font_percentages_size=3.5,
                         font_categories_size=10),
            ncol=2, nrow=2)
```

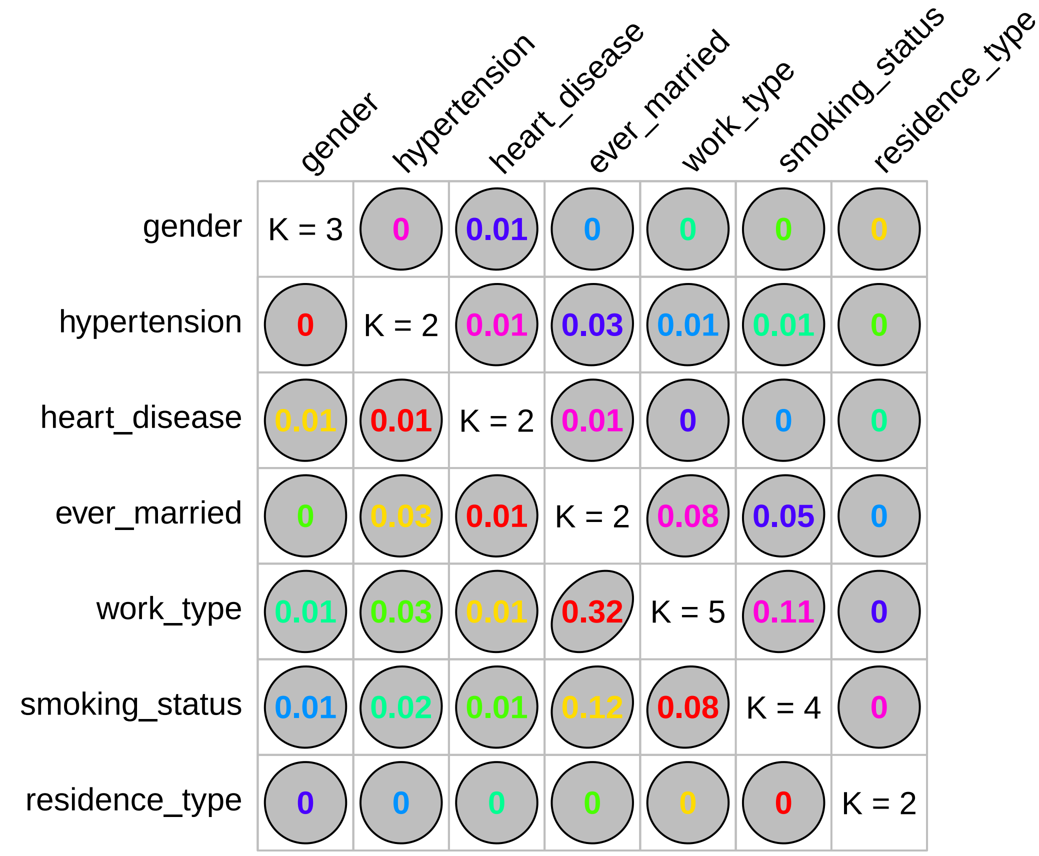






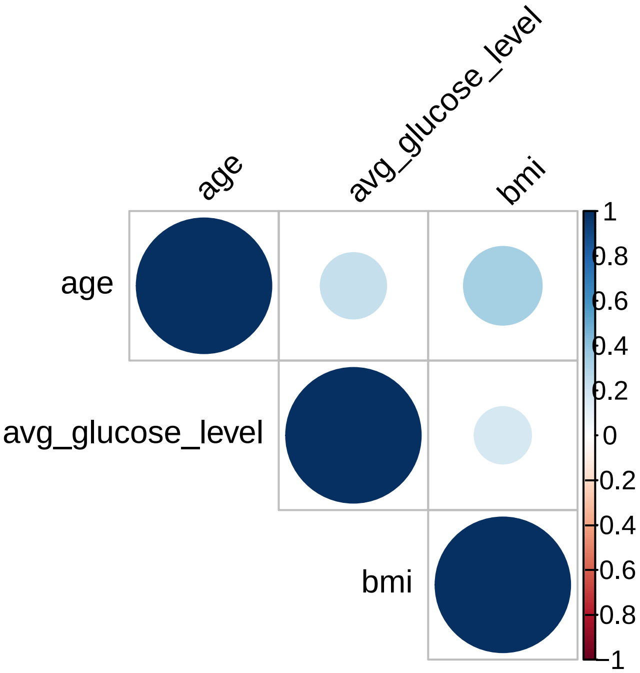


```r
# qualitative corr
qualitative_vars = c('gender', 'hypertension', 'heart_disease', 'ever_married',
                     'work_type', 'smoking_status', 'residence_type')
plot(GKtauDataframe(dataset %>% select(all_of(qualitative_vars))))
```



	gender	hypertension	heart_disease	ever_married	work_type	smoking_status	residence_type
gender	K = 3	0	0.01	0	0	0	0
hypertension	0	K = 2	0.01	0.03	0.01	0.01	0
heart_disease	0.01	0.01	K = 2	0.01	0	0	0
ever_married	0	0.03	0.01	K = 2	0.08	0.05	0
work_type	0.01	0.03	0.01	0.32	K = 5	0.11	0
smoking_status	0.01	0.02	0.01	0.12	0.08	K = 4	0
residence_type	0	0	0	0	0	0	K = 2

```
# TODO find a way to plot table and image side-by-side

quantitative_vars = c('age', 'avg_glucose_level', 'bmi')

corr <- rcorr(as.matrix(dataset %>% select(all_of(quantitative_vars))))

corrplot(corr$r, type = "upper", tl.col = "black", tl.srt = 45)
```



```
#flattenCorrMatrix(corr$r, corr$P)
```

```
# TODO clean code and stuff

# data Preprocessing, Econding with OneHotEncoding ------------------------------
#
# dummy <- dummyVars(" ~ gender + work_type + smoking_status + ever_married + Residence_type", data=str
# newdata <- data.frame(predict(dummy, newdata = stroke))
# a <- stroke[,2:4]
# b <- stroke[,8:9]
# dt <- cbind(a, b, newdata, stroke['stroke'])
# dt <- as_tibble(dt)
#
# y <- stroke['stroke']
```

```
# Fill missing bmi data w/ tree prediction
# TODO keep this way? should we just remove them?
#
# missing_index <-which(is.na(dataset$bmi))
#
# train_set <- dataset[-c(missing_index),]
#
# tree = caret::train(bmi ~ .,
#                      data=train_set,
#                      method="rpart",
#                      trControl = trainControl(method = "cv"))
# predicted_bmi <- predict(tree, newdata = dataset[missing_index,])
#
# ######################
# # What? Why?
# # x <- mean(bmi_pred)
# # bmi_pred[202] <- x
# ######################
#
# dataset[missing_index,"bmi"] <- predicted_bmi
#
# dataset = na.omit(dataset)
#
# # Check quantitative correlation is under control w/ new data
# TODO side by side :)
#
# new_corr <- rcorr(as.matrix(dataset %>% select(all_of(quantitative_vars))))
# flattenCorrMatrix(new_corr$r, new_corr$P)
#
# grid.arrange(corrplot(corr$r,     type = "upper", tl.col = "black", tl.srt = 45),
#              corrplot(new_corr$r, type = "upper", tl.col = "black", tl.srt = 45),
#              ncol=2, nrow=1)
```

```
# Solve the under sampling problem with SMOTE algho to create synth new data

# dataset <- as.data.frame(lapply(dataset, as.factor)) # What? Why?
#
# trainSplit <- DMwR::SMOTE(stroke ~ ., dt, perc.over = 2000, perc.under=10)
#
# dt_synth<- rbind(trainSplit,dt)
#
#
# dt_synth$work_type.Never_worked <- NULL # Again, but why :D
#
# dt_synth$avg_glucose_level <- as.numeric(dt_synth$avg_glucose_level)
# dt_synth$bmi <- as.numeric(dt_synth$bmi)
# dt_synth$age <- as.numeric(dt_synth$age)

# TODO Show some stats of new dataset
```

```
 # TODO recheck correlations, maybe w/ function to plot both side-by-side
```

```
 # TODO add models/predictions
```