Stroke Prediction

Adam

2022-03-07

Stroke Prediction

This project is created by using fedesoriano’s Stroke data set from Keggle. 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.”

I chose this project because I can sadly relate this by real-life experience, and I am interested in prediction of diseases since it is probably playing a big part in well-being and health assessing in the future.

Exploration

Structure of the Data set:

```
str(stroke_data)
```

```
## spec_tbl_df [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>
```

All stroke cases found in the data set and their percentage in regard of the observations:

```
sum(stroke_data$stroke)
```

```
## [1] 249
```

```
mean(stroke_data$stroke)
```

```
## [1] 0.04872798
```

Summary of the data:

```
summary(stroke_data)
```

```
##        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
```

We can see a few interesting things here. Since "hypertension" and "heart_disease" are numeric values but have only 0 or 1 as someone having heart disease or not, there is no in-between. It is better to represent them as factors so it won't interfere with calculations and won't be that confusing. Also changing the stroke column to be more clear and also be a factor since it is a classification problem (having stroke or not). BMI (body mass ratio) that is used to create an overall overview of the patients being over-weight, under-weight etc… Here BMI should be numeric for better analysis so it is a good idea to convert it as well. I removed the id column since it won't do any good in our prediction model and I don't need it overall.

```
data <- stroke_data %>%
  mutate(bmi = as.numeric(bmi), hypertension = as.factor(hypertension), heart_disease = as.factor(heart_
  na.omit(bmi) %>%
  mutate_if(is.character, as.factor)%>%
  dplyr::select(!id)
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
is.null(data)
```

```
## [1] FALSE
```

```
summary(data)
```

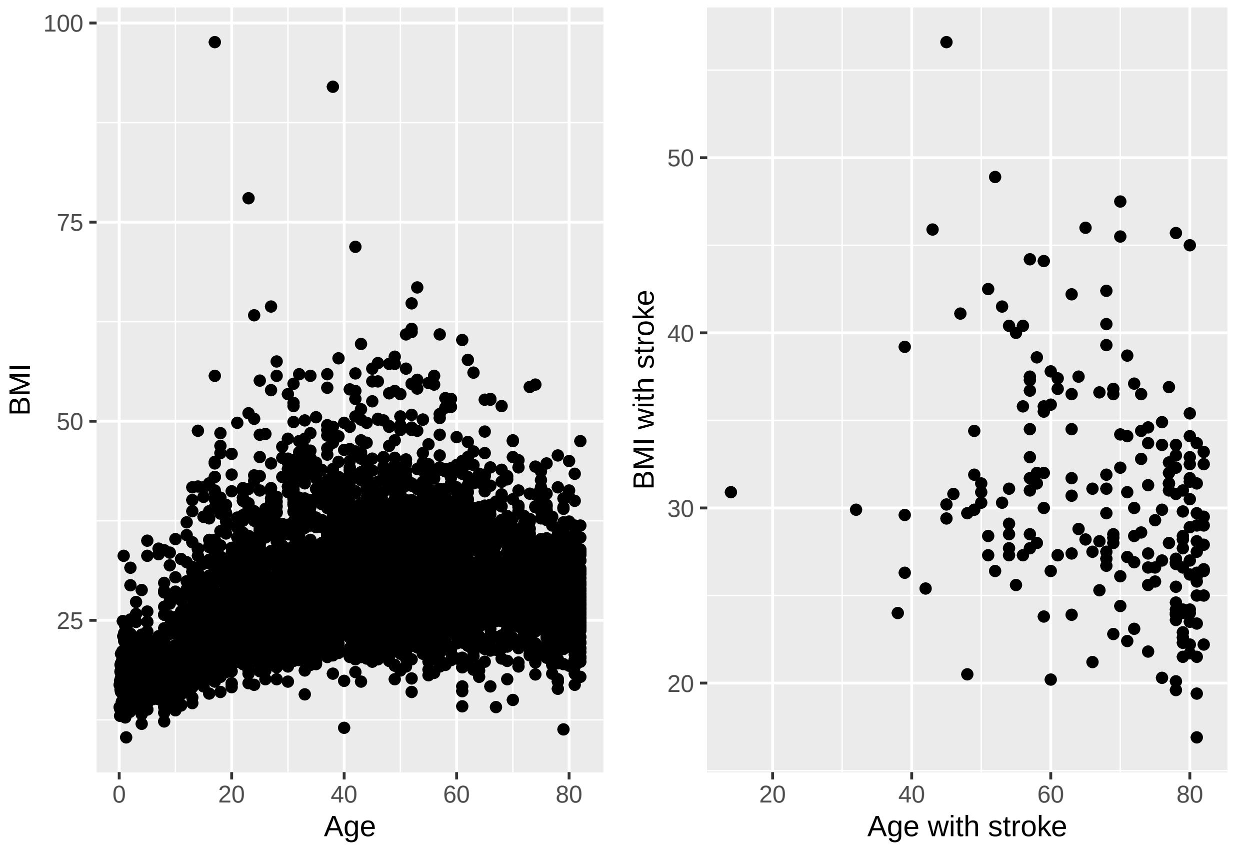
```
##     gender          age        hypertension heart_disease ever_married
##  Female:2897   Min.   : 0.08   0:4458       0:4666        No :1705
##  Male   :2011   1st Qu.:25.00   1: 451       1: 243        Yes:3204
##  Other :   1   Median :44.00
##                 Mean   :42.87
##                 3rd Qu.:60.00
##                 Max.   :82.00
##          work_type    Residence_type avg_glucose_level      bmi
##  children     : 671   Rural:2419     Min.   : 55.12    Min.   :10.30
##  Govt_job     : 630   Urban:2490     1st Qu.: 77.07    1st Qu.:23.50
##  Never_worked :  22                  Median : 91.68    Median :28.10
##  Private      :2811                  Mean   :105.31    Mean   :28.89
##  Self-employed: 775                  3rd Qu.:113.57    3rd Qu.:33.10
##                                      Max.   :271.74    Max.   :97.60
##          smoking_status stroke
##  formerly smoked: 837   NO :4700
##  never smoked   :1852   YES: 209
##  smokes         : 737
##  Unknown        :1483
##
##
```

In my theory age, bmi and glucose level might be a big factor in predicting stroke so I plot them.

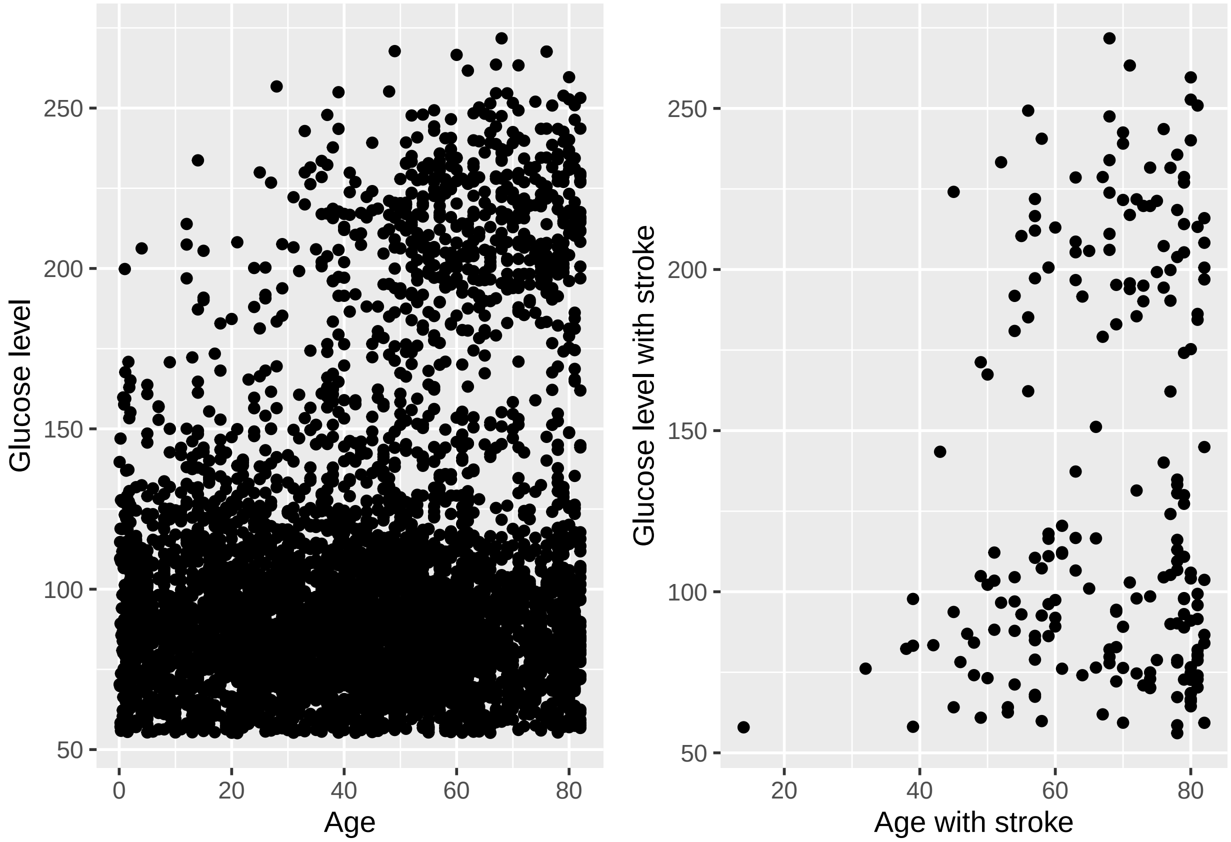
```
data_stroke <- data %>% filter(stroke == "YES")

par(mfrow=c(2,2))
p1 <- qplot(data$age, data$bmi, xlab = "Age", ylab ="BMI")
p2 <- qplot(data_stroke$age, data_stroke$bmi, xlab = "Age with stroke", ylab ="BMI with stroke")
p3 <- qplot(data$age, data$avg_glucose_level, xlab = "Age", ylab ="Glucose level")
p4 <- qplot(data_stroke$age, data_stroke$avg_glucose_level, xlab = "Age with stroke", ylab ="Glucose le

data %>% group_by(stroke) %>% summarise(avg_bmi = mean(bmi))
```

```
## # A tibble: 2 x 2
##   stroke avg_bmi
##   <fct>    <dbl>
## 1 NO        28.8
## 2 YES       30.5
```

```
grid.arrange(p1, p2, ncol=2)
```



```
grid.arrange(p3, p4, ncol=2)
```



Data seems very diverse since there is no good indication of the stroke but the chance might increase by cooperation of variables. Checking the correlations of numeric values:

```
df <- data %>%
  mutate(stroke = as.numeric(stroke))%>%
  select(stroke,age,avg_glucose_level,bmi)
cor_matrix <- cor(df$stroke, df)
cor_matrix
```

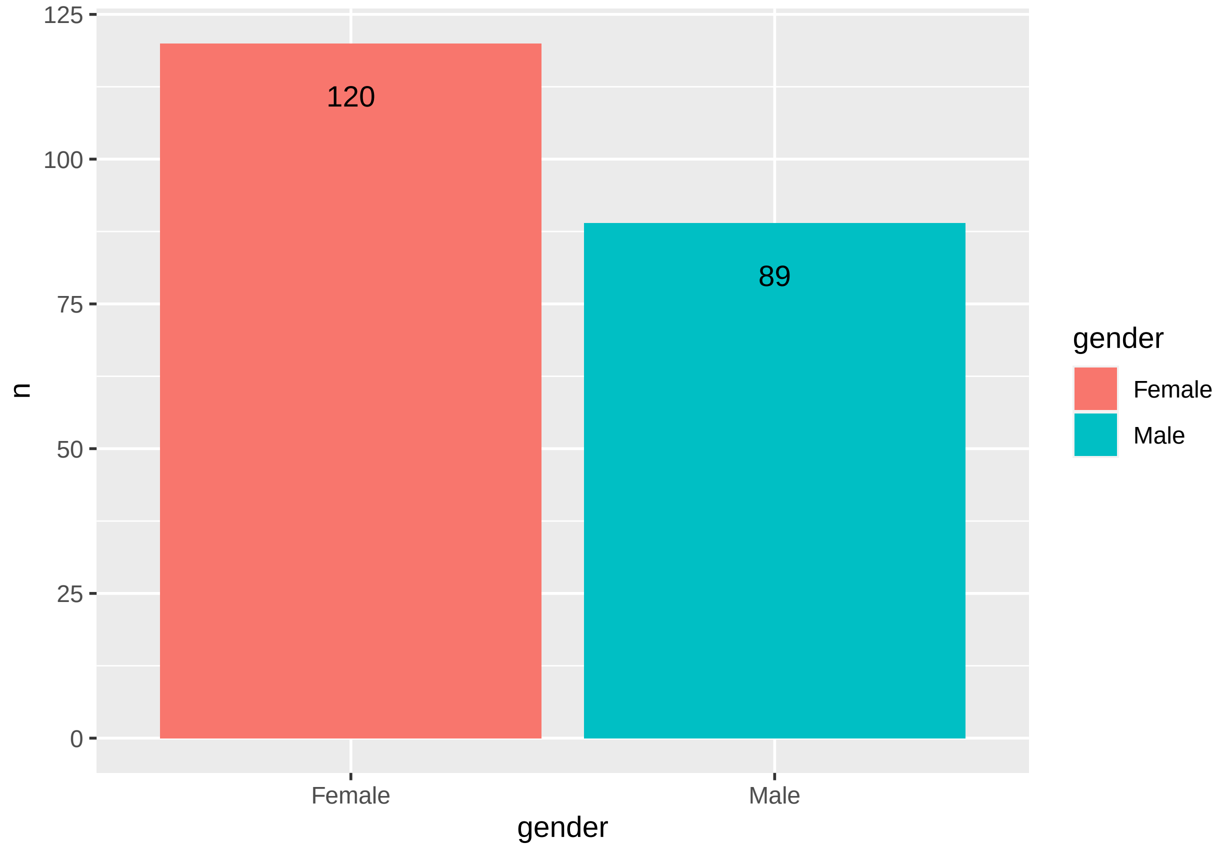
```
##      stroke       age avg_glucose_level        bmi
## [1,]      1 0.2323309         0.1389359 0.04237366
```

From the matrix it is visible that the age having the highest correlation but it is still low.

Plots of categorical variables

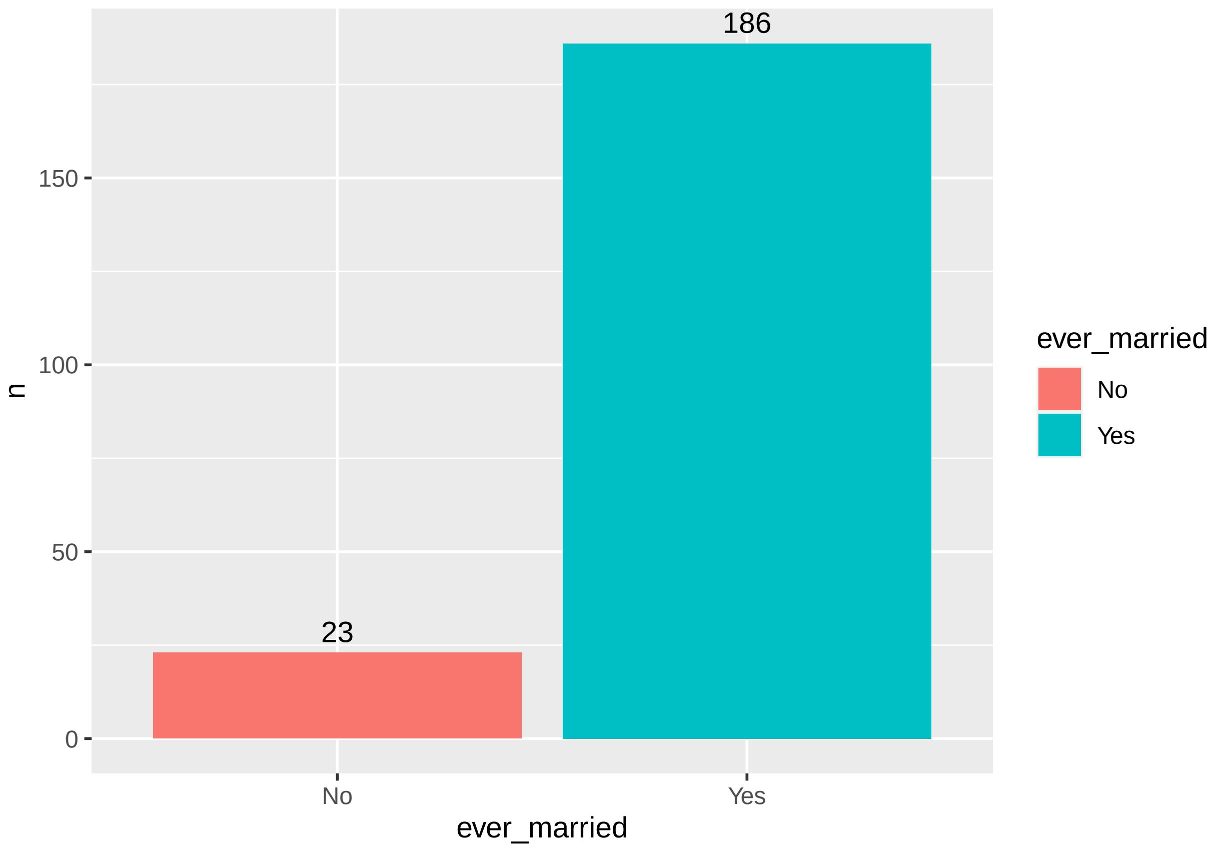
Stroke in gender:

```
data %>% filter(stroke == "YES" ) %>%
  group_by(gender) %>%
  summarise(n = n()) %>%
  ggplot(aes(x = gender, y = n, fill = gender)) +
  geom_col() +
  geom_text(aes(label = n), vjust = 3, colour = "black")
```



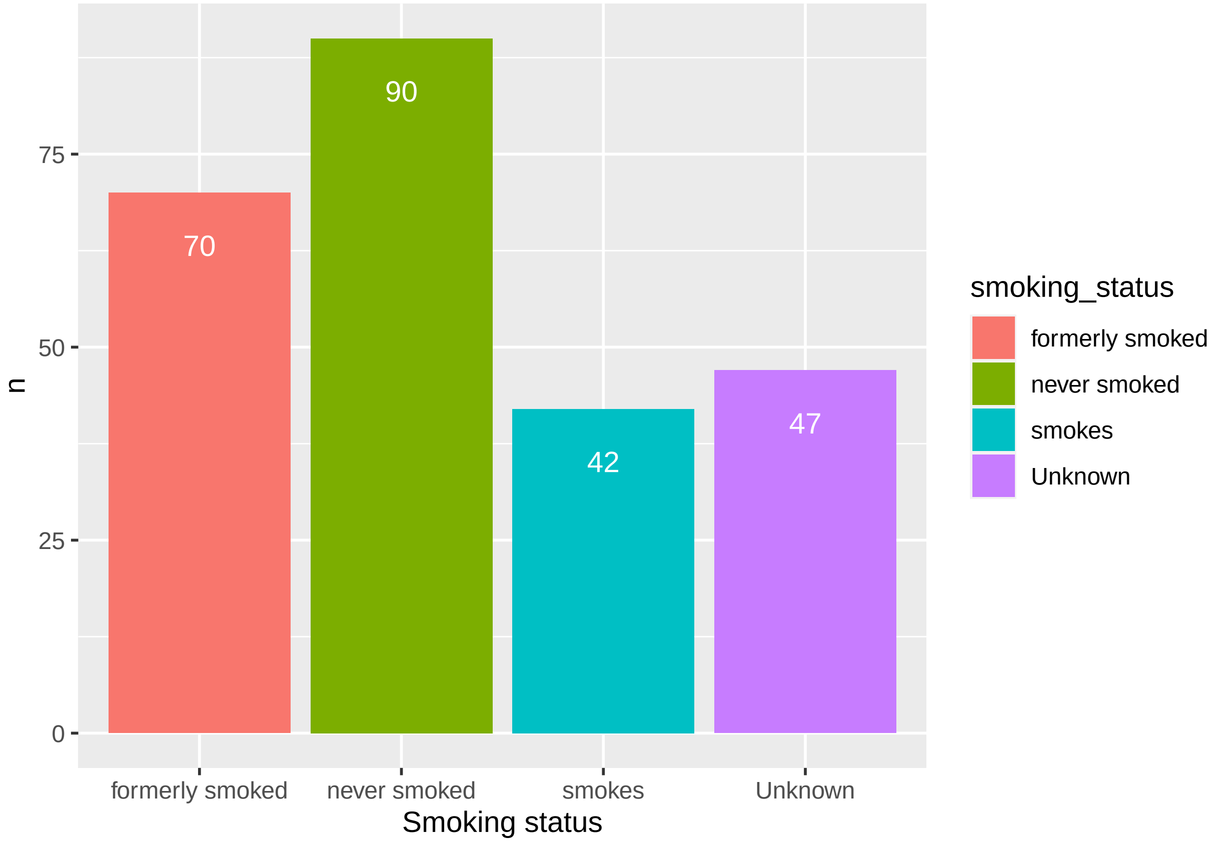
Stroke in relationship status: (note this contains younger patients too)

```
data %>% filter(stroke == "YES" ) %>%
  group_by(ever_married) %>%
  summarise(n = n()) %>%
  ggplot(aes(ever_married, n, fill = ever_married))+
  geom_col() +
  geom_text(aes(label = n), vjust = -0.5, colour = "black")
```



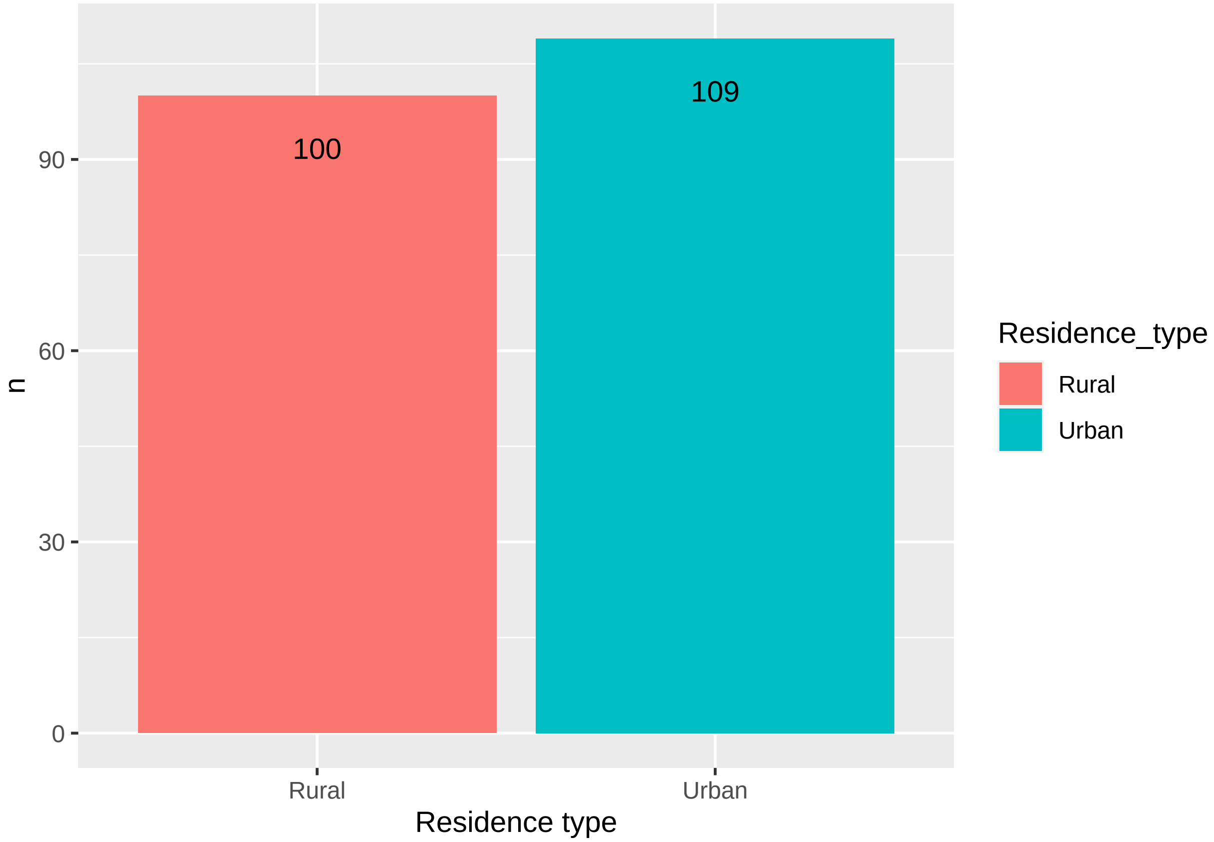
Stroke in smoking habits:

```
stroke_data %>%
  filter(stroke == 1) %>%
  group_by(smoking_status) %>%
  summarise(n = n()) %>%
  ggplot(aes(smoking_status, n, fill = smoking_status)) +
  xlab("Smoking status") +
  geom_col() +
  geom_text(aes(label = n), vjust = 3, colour = "white")
```



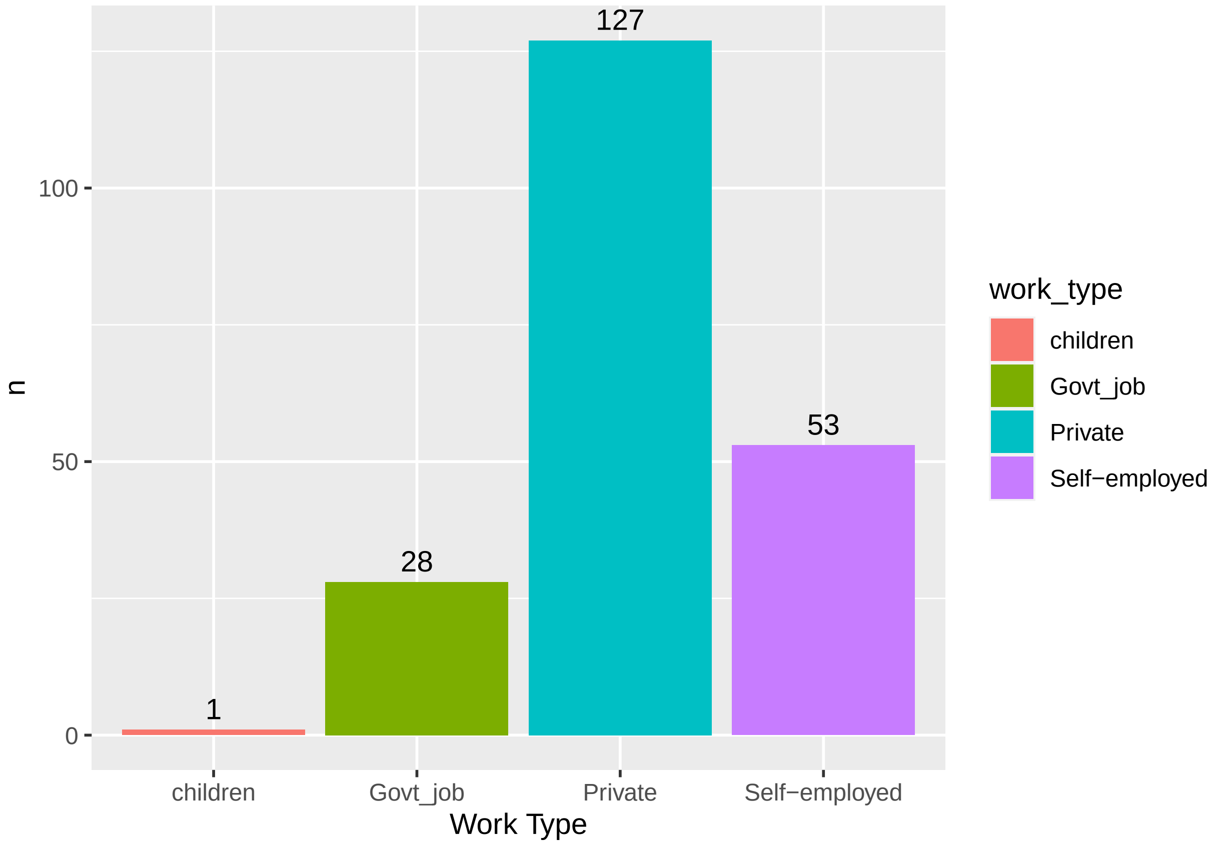
Stroke in Urban or rural environment:

```
data %>% filter(stroke == "YES" ) %>%
  group_by(Residence_type) %>%
  summarise(n = n()) %>%
  ggplot(aes(x = Residence_type, y = n, fill = Residence_type)) +
  xlab("Residence type") +
  geom_col() +
  geom_text(aes(label = n), vjust = 3, colour = "black")
```



Job type:

```
data %>% filter(stroke == "YES") %>%
  group_by(work_type) %>%
  summarise(n = n()) %>%
  ggplot(aes(x = work_type, y = n, fill = work_type)) +
  xlab("Work Type") +
  geom_col() +
  geom_text(aes(label = n), vjust = -0.5, colour = "black")
```



Partitioning Data and building a model

```
test_index <- createDataPartition(y = data$stroke, times = 1, p = 0.2, list = FALSE)
train_set <- data[-test_index,]
test_set <- data[test_index,]
```

Using General Logistic Regression model

```
options(warn=-1)
train_glm <- train(stroke ~ ., data = train_set, method = 'glm', family= 'binomial')
pred_glm <- predict(train_glm, test_set)
cm_glm <- confusionMatrix(pred_glm, test_set$stroke)
cm_glm
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  NO YES
##        NO  940  42
##        YES   0   0
##
##                Accuracy : 0.9572
##                  95% CI : (0.9426, 0.969)
##     No Information Rate : 0.9572
##     P-Value [Acc > NIR] : 0.5409
##
##                   Kappa : 0
##
##  Mcnemar’s Test P-Value : 2.509e-10
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.9572
##          Neg Pred Value :    NaN
##              Prevalence : 0.9572
##          Detection Rate : 0.9572
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        ’Positive’ Class : NO
##
```

As we can see the model gives very high accuracy. This is because the data set having actually around 4% of stroke cases. The model generally predicting “NO” for every case to reach this accuracy. This introduce false negative for the actually positive cases leading a 0 or close to zero Specificity value. This is bad because it means it is not able to detect positive cases. In this case not trying means failing.

Checking model with the usage of Random Forest:

```r
train_rf <- randomForest(stroke ~., data = train_set)
pred_rf <- predict(train_rf, test_set)
cm_rf <- confusionMatrix(pred_rf, test_set$stroke)
cm_rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  NO YES
##        NO  939  42
##        YES   1   0
##
##                Accuracy : 0.9562
##                  95% CI : (0.9415, 0.9681)
##     No Information Rate : 0.9572
##     P-Value [Acc > NIR] : 0.6023
##
##                   Kappa : -0.002
##
##  Mcnemar’s Test P-Value : 1.061e-09
##
##             Sensitivity : 0.9989
##             Specificity : 0.0000
##          Pos Pred Value : 0.9572
##          Neg Pred Value : 0.0000
##              Prevalence : 0.9572
##          Detection Rate : 0.9562
##    Detection Prevalence : 0.9990
##        Balanced Accuracy : 0.4995
##
##         'Positive' Class : NO
##
```

It performs similarly as the previous model leading balanced accuracy of 0.5 that basically means no prediction just guessing.

The problem originates the logic, where both model just working with a very low percentage of positive stroke data thus it just working towards high accuracy given the train_set.

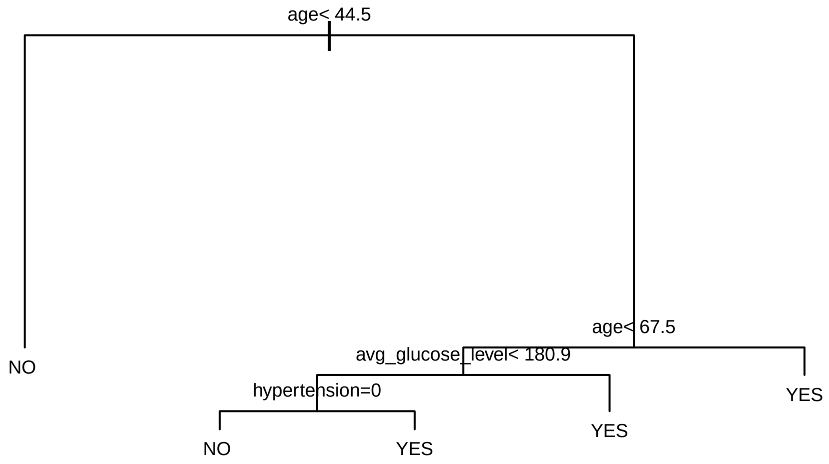
For improving a model that actually tries to guess positive cases the set need to be altered in order to boost the models confidence a bit. In theory it should improve the specificity (True positive for stroke), but given the previous plots and correlation values, the accuracy and sensitivity probably will decrease. The aim now is to increase the specificity and maximizing the balanced accuracy.

Introducing ROSE library containing ovun sampling method. "OVUN" stands for over-sampling minority examples (stroke positive) and under-sampling majority examples (stroke negative). In our case the best is to use the combination of these two cases.

```
ovun_set <- ovun.sample(stroke~.,
                    data = train_set,
                    method = "both",
                    N = 1000,
                    p = 0.5,
                    seed = 10)$data
```

According to the best votes by the forest the decision tree is constructed:

```
fit <- rpart(stroke ~ ., data = ovun_set)
plot(fit, margin = 0.1)
text(fit,  cex = 0.6, minlength = 4)
```

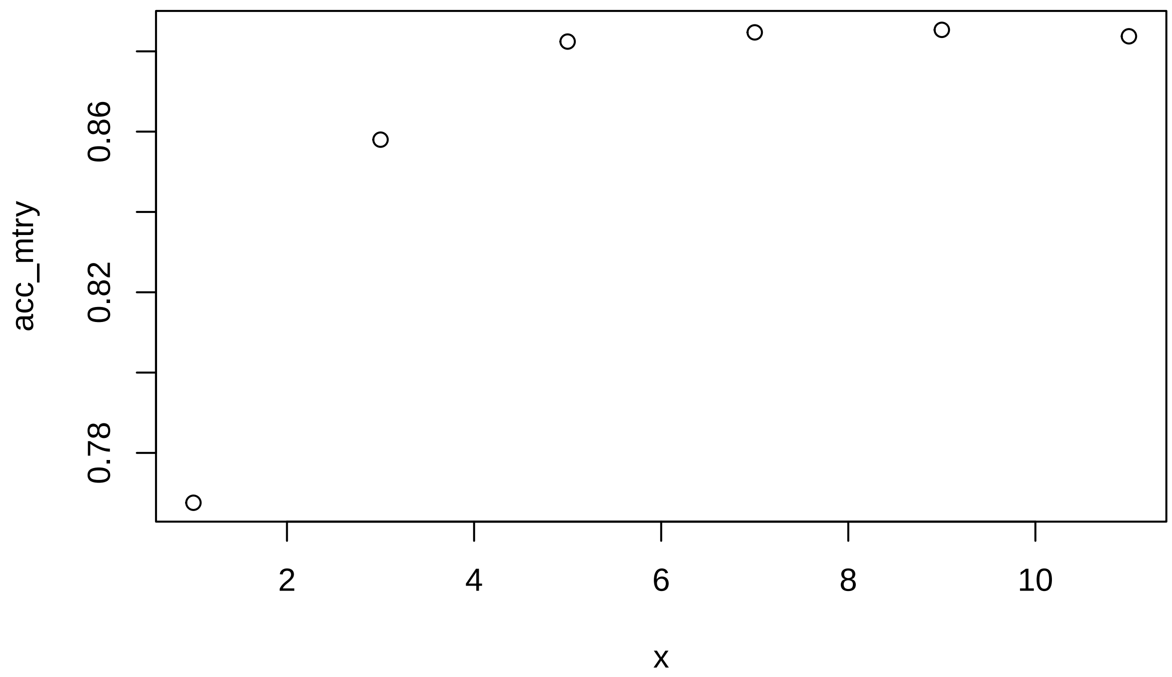


Also Introducing a control parameter that will perform 10 folds validation with 10 repetition.

```
control <- trainControl(method = "repeatedcv",
                   number = 10,
                   repeats = 10,
                   seed=10)
```

Tuning the mtry for best value:

```
x <- seq(1,11,2)
acc_mtry <- lapply(x, function(xs){
  train(stroke ~ ., method = "rf",
        data = {ovun.sample(stroke~.,
                            data = train_set,
                            method = "both",
                            N = 1000,
                            p = 0.5,
                            seed = 10)$data},
        tuneGrid = data.frame(mtry = xs))$results$Accuracy
})
plot(x,acc_mtry)
```



After value 5 the curve dumped and stayed approximately on the same accuracy level.

With these the previous models can be improved.

Improved GLM:

```
options(warn=-1)
train_glm <- train(stroke ~ ., data = ovun_set, method = 'glm', family= 'binomial')
pred_glm <- predict(train_glm, test_set)
cm_glm <- confusionMatrix(pred_glm, test_set$stroke)
cm_glm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  NO YES
##        NO  704  12
##        YES 236  30
##
##                Accuracy : 0.7475
##                  95% CI : (0.7191, 0.7744)
##     No Information Rate : 0.9572
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1306
##
##  Mcnemar's Test P-Value : <2e-16
```

```
##
##             Sensitivity : 0.7489
##             Specificity : 0.7143
##          Pos Pred Value : 0.9832
##          Neg Pred Value : 0.1128
##              Prevalence : 0.9572
##          Detection Rate : 0.7169
##    Detection Prevalence : 0.7291
##       Balanced Accuracy : 0.7316
##
##        'Positive' Class : NO
##
```

Improved Random forest model:

```
model_rf <- randomForest(stroke~.,
                       data = ovun_set,
                       mtry=x[which.max(acc_mtry)],
                       trControl = control,
                       ntrees = 500,
                       seed = 10)

pred_rf <- predict(model_rf, test_set)
cm_rf <- confusionMatrix(pred_rf, test_set$stroke)
cm_rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  NO YES
##        NO  801  22
##        YES 139  20
##
##                Accuracy : 0.836
##                  95% CI : (0.8114, 0.8587)
##     No Information Rate : 0.9572
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1409
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8521
##             Specificity : 0.4762
##          Pos Pred Value : 0.9733
##          Neg Pred Value : 0.1258
##              Prevalence : 0.9572
##          Detection Rate : 0.8157
##    Detection Prevalence : 0.8381
##       Balanced Accuracy : 0.6642
##
##        'Positive' Class : NO
##
```

Conclusion

The models shown above are increasing the prediction chance of the specificity (True positive rate) even is the overall accuracy is decreased. Random forest and generalized logistical regression used to form the models.

GLM model performed better than the random forest in overall prediction. It did predict more False positive but it did recognize more positive cases and have a better specificity and balanced and overall-accuracy. It would help more in detecting stroke.

Overall the data set seems a bit small, with larger set a better, more confident model could be set up.

In case of disease in my opinion it is better to have a higher specificity and having a better rate in pre detection of stroke. How ever the stroke is instant and there is more elements, more predictors that is added to the chance that is not presented in the data set. BUT, it might serve a good indicator for the people to watch their health even in later ages.