# Stroke Prediction

## 08/01/2021

### Importing packages

```
library(tidyverse)      # Collection of R packages for data science
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ ggplot2 3.3.4     ✓ purrr   0.3.4
## ✓ tibble  3.1.2     ✓ dplyr   1.0.7
## ✓ tidyr   1.1.3     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(naniar)         # Data structures and functions for plotting of missing value
s
library(caTools)        # Several basic utility functions
library(ggplot2)        # Data visualisations Using the Grammar of Graphics
library(superheat)      # Generating customizable heatmaps
library(scatterplot3d)  # Plots a three dimensional point cloud
library(ROCR)           # Creating cutoff-parameterized 2D performance curves
```

# Dataset

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. This dataset 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 relavant information about the patient.

The dataset can be found in the repository (https://github.com/adnanhakim/stroke-prediction) or can be downloaded from Kaggle (https://www.kaggle.com/fedesoriano/stroke-prediction-dataset)

```
data = read.csv("~/Documents/506/second-git/stroke-prediction/stroke_data.csv")
str(data)
```

```
## 'data.frame':    5110 obs. of  10 variables:
##  $ 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" ...
##  $ 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 ...
```

```
glimpse(data)
```

```
## Rows: 5,110
## Columns: 10
## $ gender            <chr> "Male", "Female", "Male", "Female", "Female", "Male"…
## $ age               <dbl> 67, 61, 80, 49, 79, 81, 74, 69, 59, 78, 81, 61, 54, …
## $ hypertension      <int> 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1…
## $ heart_disease     <int> 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…
## $ 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            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
```

# Data Preprocessing

## Checking dataset values

### 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. Residence_type: “Rural” or “Urban”
8. avg_glucose_level: average glucose level in blood
9. bmi: body mass index
10. smoking_status: “formerly smoked”, “never smoked”, “smokes” or “Unknown”*
11. 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*

```
unique(data $ gender)
```

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

```
unique(data $ ever_married)
```

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

```
unique(data $ Residence_type)
```

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

```
unique(data $ smoking_status)
```

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

# Converting character values to numeric values

As seen in the above values, the character values can be converted into numeric values.

```
clean_data <- data %>% mutate(gender = if_else(gender == "Female", 0, if_else(gender
== "Male", 1, 2)), ever_married = if_else(ever_married == "Yes", 1, 0), Residence_typ
e = if_else(Residence_type == "Rural", 0, 1), smoking_status = if_else(smoking_status
== "never smoked", 0, if_else(smoking_status == "formerly smoked", 1, if_else(smoking
_status == "smokes", 2, 3))))
summary(clean_data)
```

```
##      gender            age          hypertension     heart_disease
##  Min.   :0.0000   Min.   : 0.08   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:25.00   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.0000   Median :45.00   Median :0.00000   Median :0.00000
##  Mean   :0.4143   Mean   :43.23   Mean   :0.09746   Mean   :0.05401
##  3rd Qu.:1.0000   3rd Qu.:61.00   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :2.0000   Max.   :82.00   Max.   :1.00000   Max.   :1.00000
##   ever_married    Residence_type  avg_glucose_level     bmi
##  Min.   :0.0000   Min.   :0.000   Min.   : 55.12    Length:5110
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.: 77.25    Class :character
##  Median :1.0000   Median :1.000   Median : 91.89    Mode  :character
##  Mean   :0.6562   Mean   :0.508   Mean   :106.15
##  3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:114.09
##  Max.   :1.0000   Max.   :1.000   Max.   :271.74
##  smoking_status      stroke
##  Min.   :0.000   Min.   :0.00000
##  1st Qu.:0.000   1st Qu.:0.00000
##  Median :1.000   Median :0.00000
##  Mean   :1.388   Mean   :0.04873
##  3rd Qu.:3.000   3rd Qu.:0.00000
##  Max.   :3.000   Max.   :1.00000
```

# Handling missing values

```
miss_scan_count(data = data, search = list("N/A", "Unknown"))
```

```
## # A tibble: 10 x 2
##    Variable              n
##    <chr>             <int>
##  1 gender                0
##  2 age                   0
##  3 hypertension          0
##  4 heart_disease         0
##  5 ever_married          0
##  6 Residence_type        0
##  7 avg_glucose_level     0
##  8 bmi                 201
##  9 smoking_status     1544
## 10 stroke                0
```

There are 201 “N/A” values in the bmi column that likely caused this column to be parsed as character, although it should be numerical. Let’s take care of that by replacing those values with actual NAs. Moreover, there are a lot of “Unknown” values in smoking_status which we have to take care of too. We see that we have 1544 unknown values for smoking status and therefore are missing a lot of information in a potentially informative predictor. We will have to deal with this. Lets replace those values with NAs.

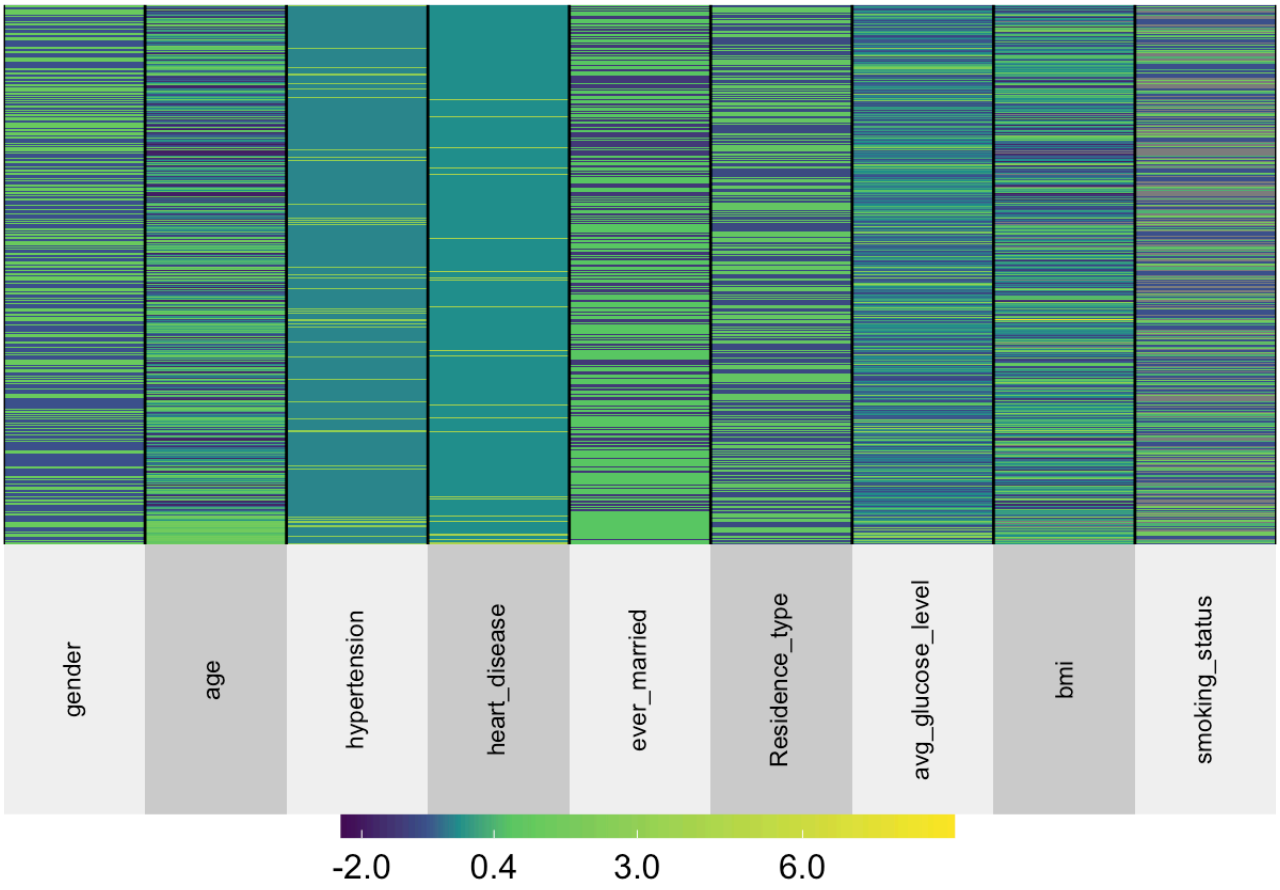
```
clean_data <- replace_with_na(data = clean_data, replace = list(bmi = c("N/A"), smoki
ng_status = c(3))) %>% mutate(bmi = as.numeric(bmi))
summary(clean_data)
```

```
##      gender            age          hypertension     heart_disease
##  Min.   :0.0000   Min.   : 0.08   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:25.00   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.0000   Median :45.00   Median :0.00000   Median :0.00000
##  Mean   :0.4143   Mean   :43.23   Mean   :0.09746   Mean   :0.05401
##  3rd Qu.:1.0000   3rd Qu.:61.00   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :2.0000   Max.   :82.00   Max.   :1.00000   Max.   :1.00000
##
##   ever_married    Residence_type  avg_glucose_level      bmi
##  Min.   :0.0000   Min.   :0.000   Min.   : 55.12    Min.   :10.30
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.: 77.25    1st Qu.:23.50
##  Median :1.0000   Median :1.000   Median : 91.89    Median :28.10
##  Mean   :0.6562   Mean   :0.508   Mean   :106.15    Mean   :28.89
##  3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:114.09    3rd Qu.:33.10
##  Max.   :1.0000   Max.   :1.000   Max.   :271.74    Max.   :97.60
##                                                     NA's   :201
##  smoking_status       stroke
##  Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.0000   Median :0.00000
##  Mean   :0.6907   Mean   :0.04873
##  3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :2.0000   Max.   :1.00000
##  NA's   :1544
```

# Visualizing the input

## Heatmap

```
superheat(subset(clean_data, select = -c(stroke)), scale = TRUE, bottom.label.size =
0.5, bottom.label.text.angle = 90, bottom.label.text.size = 3)
```

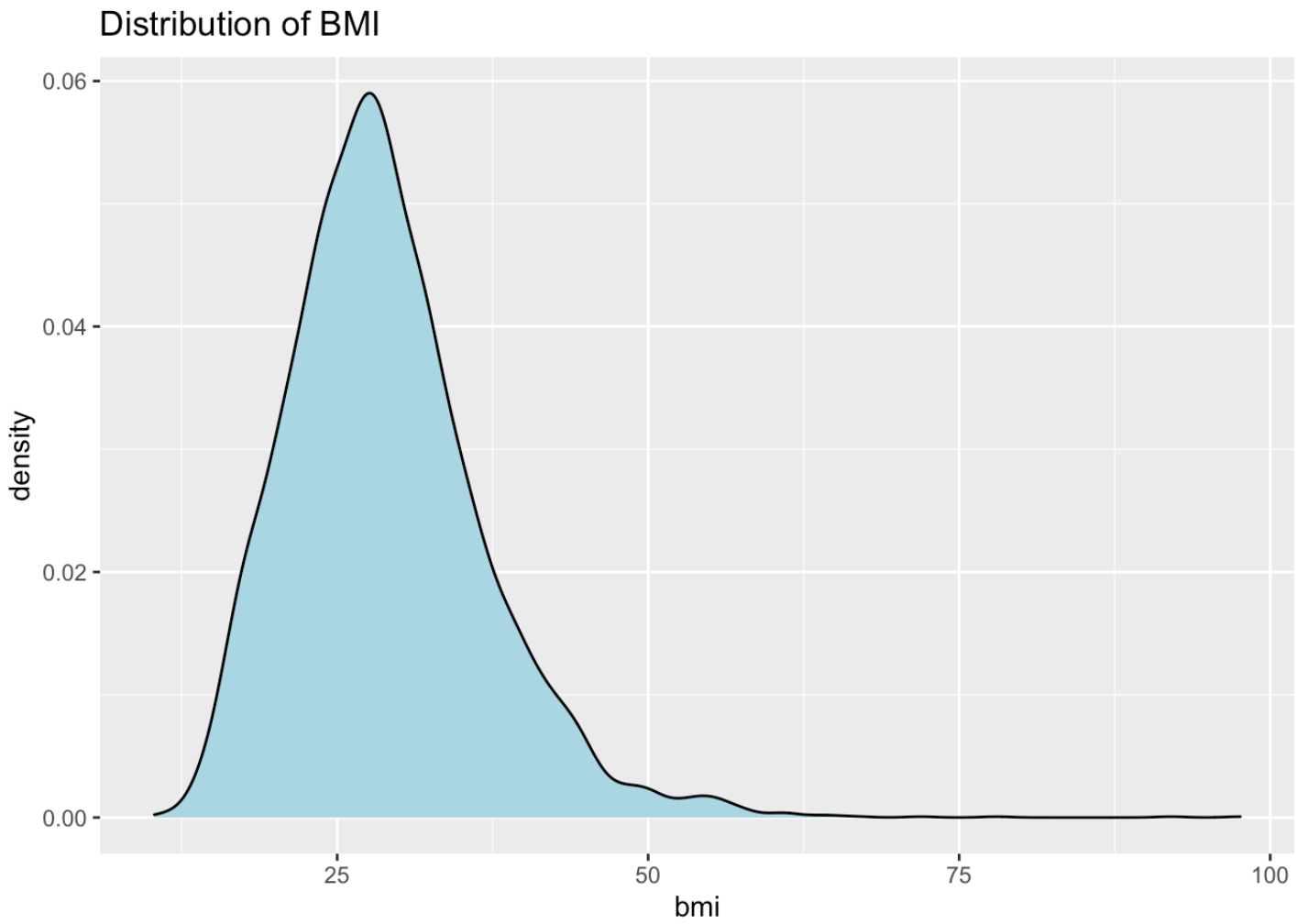



## BMI Distribution

We see that the missingness doesn't show clear association with other variables and therefore we can assume this missingness is MCAR (missing completely at random). The distribution is right skewed (long tail to the right) as this is the only variable with missing data (at least of the numerical variables).

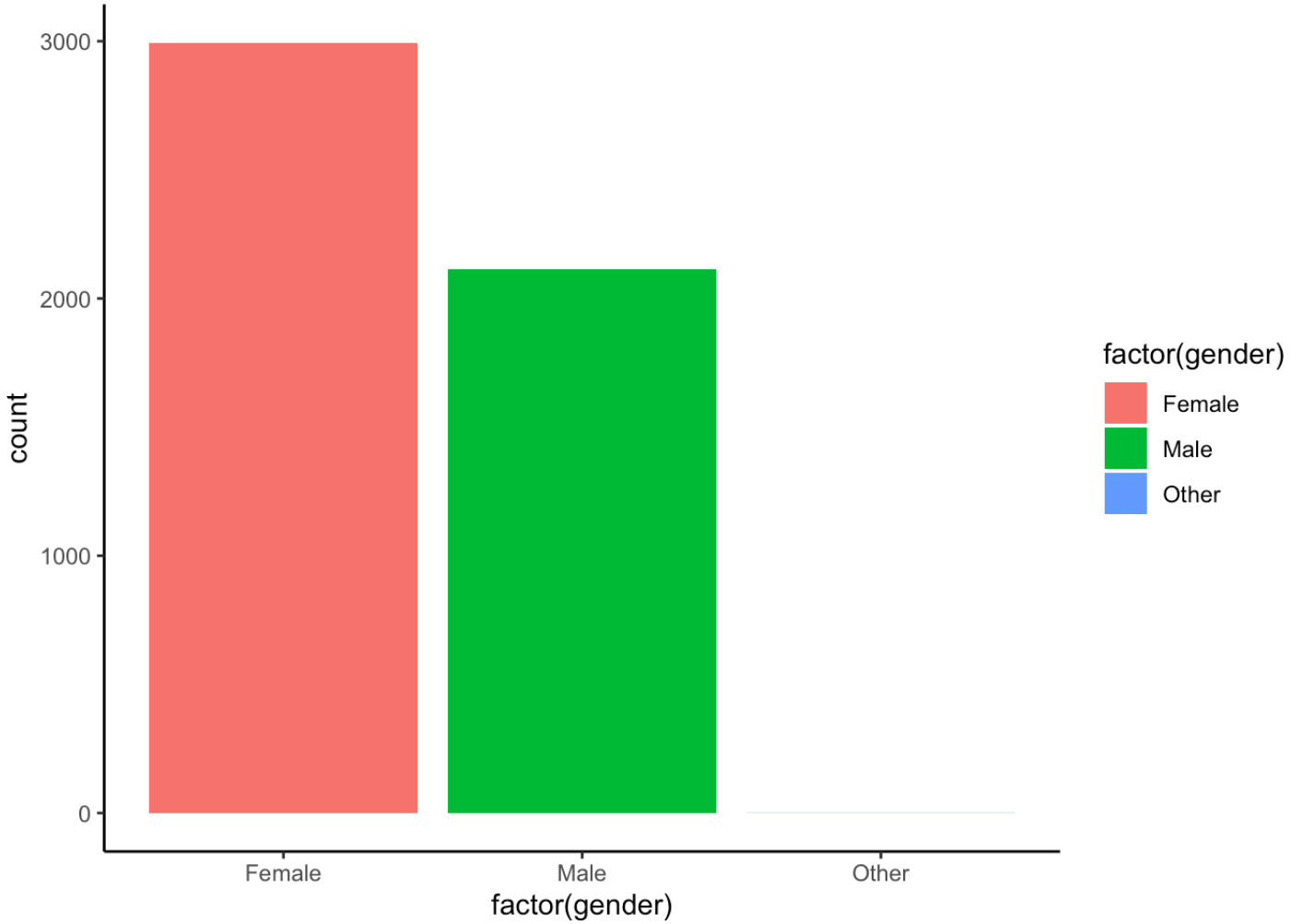
```
ggplot(clean_data, aes(x = bmi)) + geom_density(color="black", fill="lightblue") + la
bs(title = "Distribution of BMI")
```

```
## Warning: Removed 201 rows containing non-finite values (stat_density).
```




## Gender Distribution

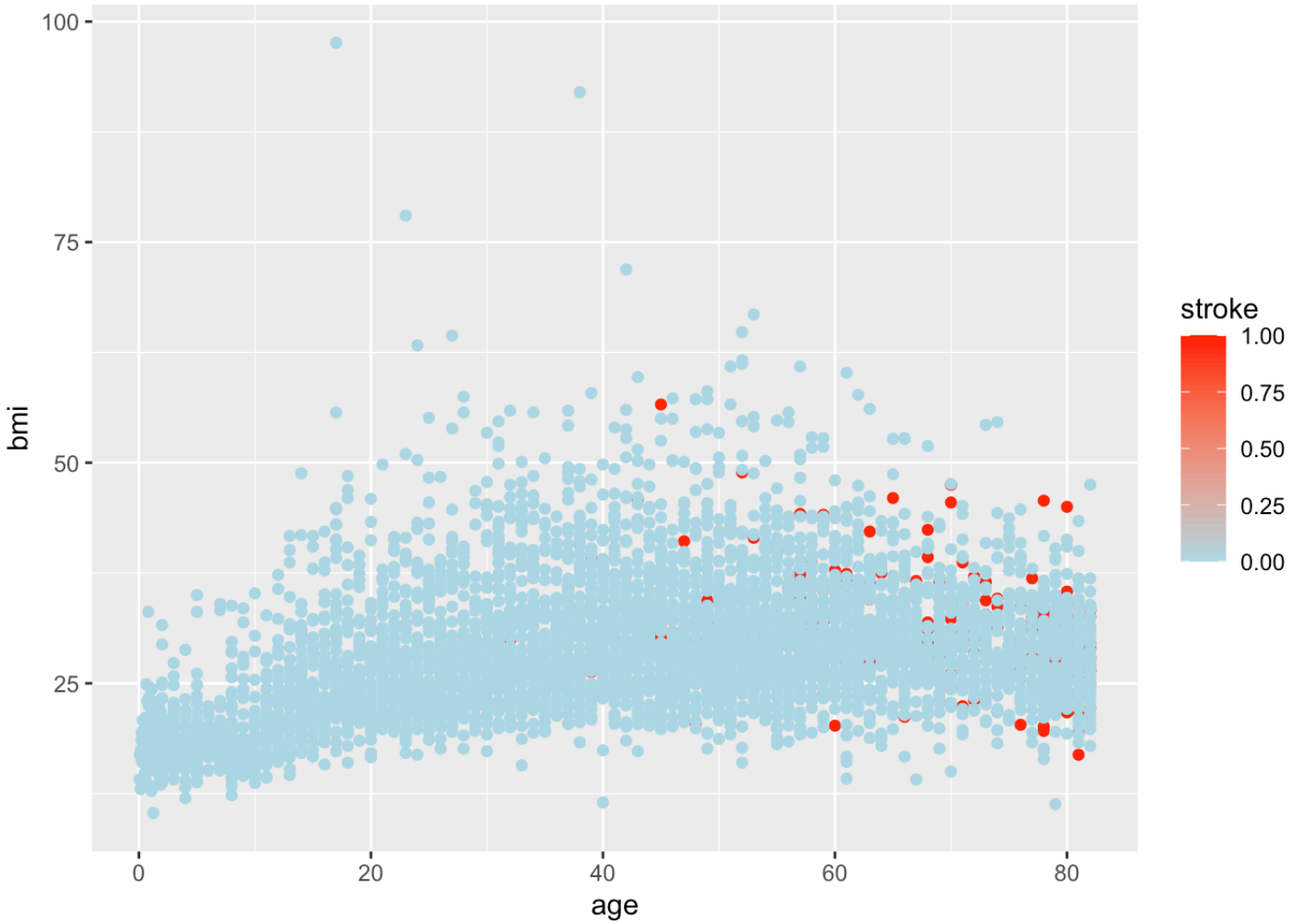
```
ggplot(data, aes(x = factor(gender), fill = factor(gender))) + geom_bar() + theme_cla
ssic()
```




## Age and BMI wrt stroke

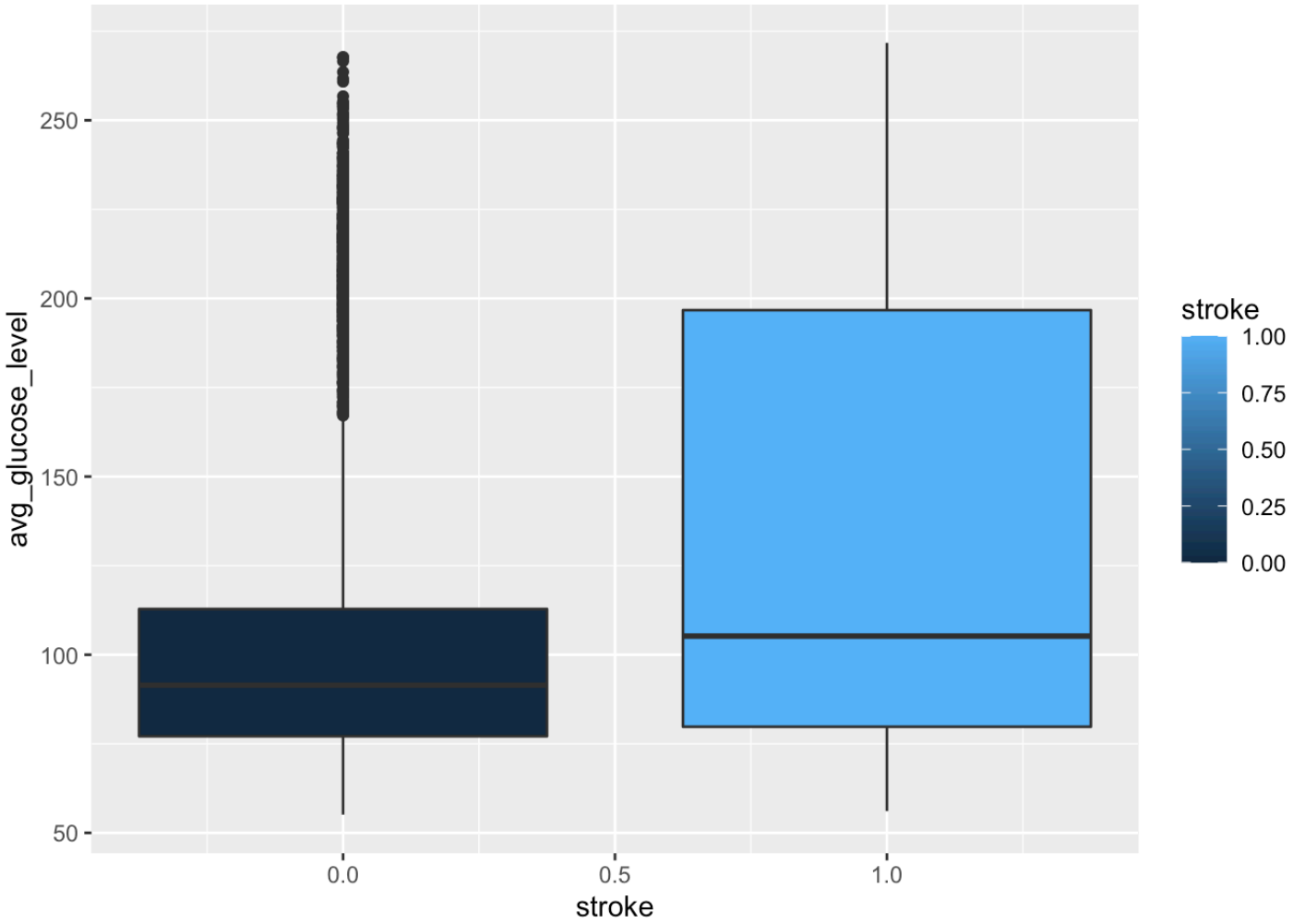
```
ggplot(clean_data, aes(x = age, y = bmi, color = stroke)) + geom_point() + scale_colo
r_gradient(low = "lightblue", high = "red")
```

```
## Warning: Removed 201 rows containing missing values (geom_point).
```




## Avg Glucose Level with stroke

```
ggplot(clean_data, aes(x = stroke, y = avg_glucose_level, group = stroke, fill = stro
ke)) + geom_boxplot()
```




# Logistic Regression

```
set.seed(99)  # Set a seed for reproducible results
split = sample.split(clean_data $ stroke, SplitRatio = 0.7)
train = subset(clean_data, split == TRUE)
test = subset(clean_data, split == FALSE)
logistic_regression_1 = glm(stroke~., data = train, family = 'binomial')
summary(logistic_regression_1)
```

```
##
## Call:
## glm(formula = stroke ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1462  -0.3419  -0.1978  -0.1186   3.0953
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.648670   0.754584 -10.136  < 2e-16 ***
## gender            -0.197661   0.201751  -0.980  0.32722
## age                0.066603   0.007607   8.755  < 2e-16 ***
## hypertension       0.691216   0.212789   3.248  0.00116 **
## heart_disease      0.349765   0.273365   1.279  0.20073
## ever_married      -0.401483   0.287007  -1.399  0.16185
## Residence_type    -0.053097   0.193027  -0.275  0.78326
## avg_glucose_level  0.004067   0.001664   2.443  0.01455 *
## bmi                0.014647   0.014746   0.993  0.32058
## smoking_status     0.254129   0.125064   2.032  0.04215 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 992.58  on 2389  degrees of freedom
## Residual deviance: 815.57  on 2380  degrees of freedom
##   (1187 observations deleted due to missingness)
## AIC: 835.57
##
## Number of Fisher Scoring iterations: 7
```

A lot of variables are not significant. Hence we will be removing Variables based on significance level. The least significant variable as seen is Residence_type with a Pr-value of 0.78326. Hence we will remove Residence_type.

```
logistic_regression_2 = glm(stroke ~ gender + age + hypertension + heart_disease + ev
er_married + avg_glucose_level + bmi + smoking_status, data = train, family = 'binomi
al')
summary(logistic_regression_2)
```

```
##
## Call:
## glm(formula = stroke ~ gender + age + hypertension + heart_disease +
##     ever_married + avg_glucose_level + bmi + smoking_status,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1348  -0.3413  -0.1978  -0.1186   3.0857
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.670767   0.750364 -10.223  < 2e-16 ***
## gender            -0.195520   0.201597  -0.970  0.33212
## age                0.066508   0.007594   8.758  < 2e-16 ***
## hypertension       0.692810   0.212713   3.257  0.00113 **
## heart_disease      0.350092   0.273423   1.280  0.20040
## ever_married      -0.399230   0.286831  -1.392  0.16396
## avg_glucose_level  0.004052   0.001664   2.436  0.01485 *
## bmi                0.014700   0.014747   0.997  0.31886
## smoking_status     0.251534   0.124711   2.017  0.04370 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 992.58  on 2389  degrees of freedom
## Residual deviance: 815.65  on 2381  degrees of freedom
##   (1187 observations deleted due to missingness)
## AIC: 833.65
##
## Number of Fisher Scoring iterations: 7
```

The least significant variable as seen is gender with a Pr-value of 0.33212. Hence we will remove gender.

```
logistic_regression_2 = glm(stroke ~ age + hypertension + heart_disease + ever_marrie
d + avg_glucose_level + bmi + smoking_status, data = train, family = 'binomial')
summary(logistic_regression_2)
```

```
##
## Call:
## glm(formula = stroke ~ age + hypertension + heart_disease + ever_married +
##     avg_glucose_level + bmi + smoking_status, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1760  -0.3427  -0.1969  -0.1183   3.1056
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.724040   0.750464 -10.292  < 2e-16 ***
## age                0.066570   0.007599   8.761  < 2e-16 ***
## hypertension       0.692631   0.212493   3.260  0.00112 **
## heart_disease      0.317594   0.270992   1.172  0.24121
## ever_married      -0.413988   0.286149  -1.447  0.14797
## avg_glucose_level  0.003888   0.001651   2.354  0.01856 *
## bmi                0.015353   0.014791   1.038  0.29928
## smoking_status     0.234714   0.123694   1.898  0.05776 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 992.58  on 2389  degrees of freedom
## Residual deviance: 816.60  on 2382  degrees of freedom
##   (1187 observations deleted due to missingness)
## AIC: 832.6
##
## Number of Fisher Scoring iterations: 7
```

The least significant variable as seen is bmi with a Pr-value of 0.29928. Hence we will remove bmi.

```
logistic_regression_2 = glm(stroke ~ age + hypertension + heart_disease + ever_marrie
d + avg_glucose_level + smoking_status, data = train, family = 'binomial')
summary(logistic_regression_2)
```

```
##
## Call:
## glm(formula = stroke ~ age + hypertension + heart_disease + ever_married +
##     avg_glucose_level + smoking_status, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1298  -0.3561  -0.2082  -0.1257   3.0848
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.166651   0.513513 -13.956  < 2e-16 ***
## age                0.066847   0.007042   9.493  < 2e-16 ***
## hypertension       0.575626   0.203317   2.831  0.00464 **
## heart_disease      0.198980   0.261232   0.762  0.44624
## ever_married      -0.380014   0.276856  -1.373  0.16987
## avg_glucose_level  0.003918   0.001539   2.545  0.01092 *
## smoking_status     0.179974   0.117392   1.533  0.12525
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1077.58  on 2486  degrees of freedom
## Residual deviance:  893.96  on 2480  degrees of freedom
##   (1090 observations deleted due to missingness)
## AIC: 907.96
##
## Number of Fisher Scoring iterations: 7
```

The least significant variable as seen is heart_disease with a Pr-value of 0.44624. Hence we will remove heart_disease.

```
logistic_regression_2 = glm(stroke ~ age + hypertension + ever_married + avg_glucose_
level + smoking_status, data = train, family = 'binomial')
summary(logistic_regression_2)
```

```
##
## Call:
## glm(formula = stroke ~ age + hypertension + ever_married + avg_glucose_level +
##     smoking_status, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0774  -0.3597  -0.2084  -0.1251   3.0873
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.213448   0.510203 -14.138  < 2e-16 ***
## age                0.067823   0.006918   9.803  < 2e-16 ***
## hypertension       0.579361   0.203039   2.853  0.00432 **
## ever_married      -0.398364   0.275464  -1.446  0.14813
## avg_glucose_level  0.004081   0.001524   2.678  0.00740 **
## smoking_status     0.186886   0.116936   1.598  0.11000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1077.58  on 2486  degrees of freedom
## Residual deviance:  894.53  on 2481  degrees of freedom
##   (1090 observations deleted due to missingness)
## AIC: 906.53
##
## Number of Fisher Scoring iterations: 7
```

The least significant variable as seen is ever_married with a Pr-value of 0.14813. Hence we will remove ever_married.

```
logistic_regression_2 = glm(stroke ~ age + hypertension + avg_glucose_level + smoking
_status, data = train, family = 'binomial')
summary(logistic_regression_2)
```

```
##
## Call:
## glm(formula = stroke ~ age + hypertension + avg_glucose_level +
##     smoking_status, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0285  -0.3647  -0.2113  -0.1189   3.0586
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.448500   0.497525 -14.971  < 2e-16 ***
## age                0.066389   0.006947   9.557  < 2e-16 ***
## hypertension       0.587793   0.202469   2.903  0.00369 **
## avg_glucose_level  0.003947   0.001516   2.604  0.00921 **
## smoking_status     0.177741   0.116688   1.523  0.12771
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1077.58  on 2486  degrees of freedom
## Residual deviance:  896.49  on 2482  degrees of freedom
##   (1090 observations deleted due to missingness)
## AIC: 906.49
##
## Number of Fisher Scoring iterations: 7
```

The least significant variable as seen is smoking_status with a Pr-value of 0.12771. Hence we will remove smoking_status.

```
logistic_regression_2 = glm(stroke ~ age + hypertension + avg_glucose_level, data = t
rain, family = 'binomial')
summary(logistic_regression_2)
```

```
##
## Call:
## glm(formula = stroke ~ age + hypertension + avg_glucose_level,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9796  -0.3283  -0.1788  -0.0880   3.7420
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.398490   0.412644 -17.929  < 2e-16 ***
## age                0.067642   0.005925  11.416  < 2e-16 ***
## hypertension       0.389762   0.191117   2.039  0.04141 *
## avg_glucose_level  0.004391   0.001377   3.188  0.00143 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1391.5  on 3576  degrees of freedom
## Residual deviance: 1125.8  on 3573  degrees of freedom
## AIC: 1133.8
##
## Number of Fisher Scoring iterations: 7
```

Hence we get the three most significant variables having Pr-values less than 0.05.

# Predictions on training set and confusion matrix

```
predict_train = predict(logistic_regression_2, type = 'response')
table(train $ stroke, predict_train>0.2)
```

```
##
##     FALSE TRUE
##   0  3274  129
##   1   136   38
```

# Accuracy on training set

```
(3274 + 38) / nrow(train)
```

```
## [1] 0.9259156
```

# Predictions on test set

```
predict_test = predict(logistic_regression_2, newdata = test, type = 'response')
table(test $ stroke, predict_test>0.2)
```

```
##
##     FALSE TRUE
##   0  1414   44
##   1    57   18
```

## Accuracy on testing set

```
(1414 + 18) / nrow(test)
```

```
## [1] 0.9341161
```

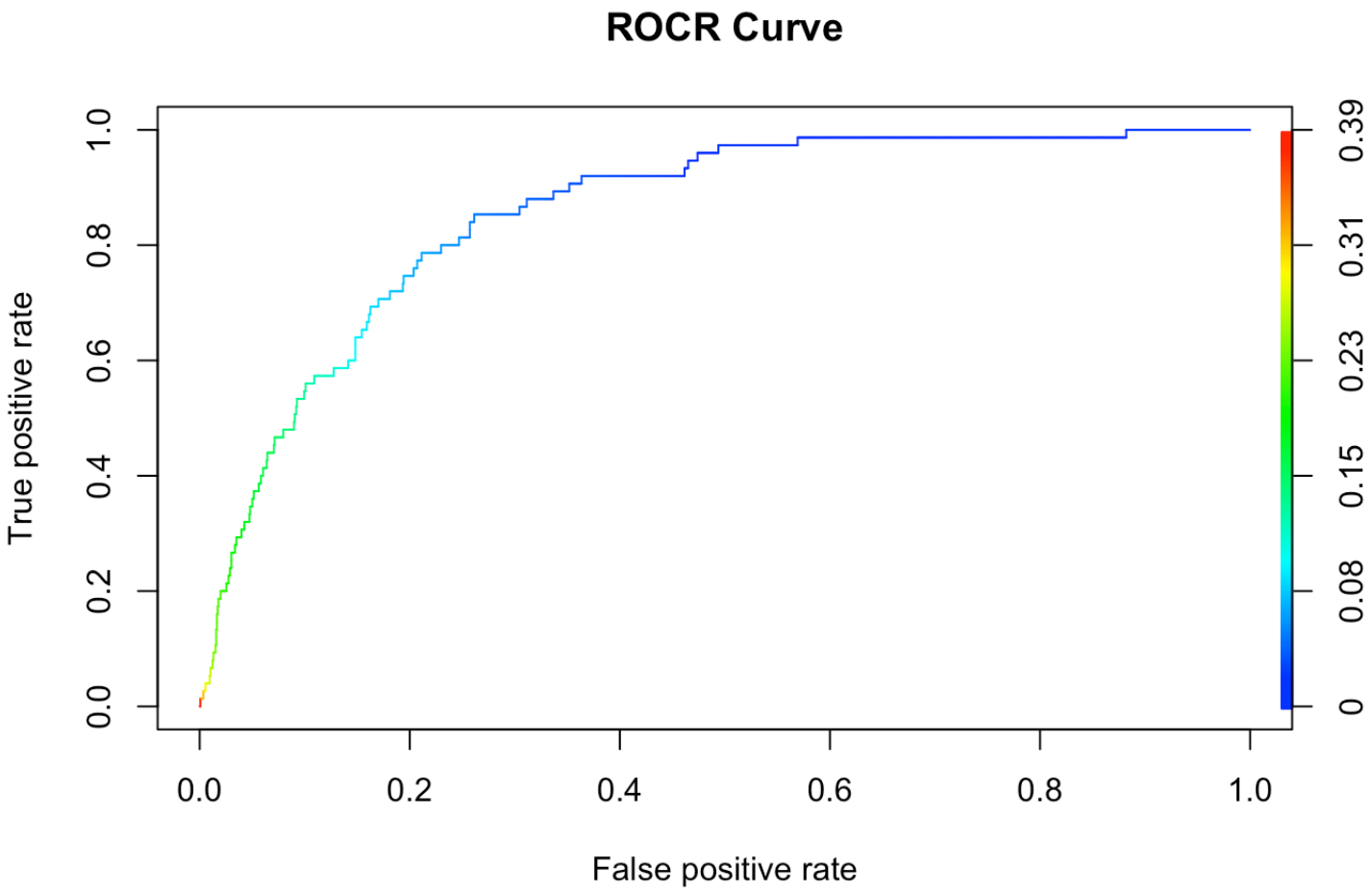
# Plotting results

## ROCR Curve and area under the curve

```
rocr_prediction = prediction(predict_test, test $ stroke)
auc = as.numeric(performance(rocr_prediction, 'auc') @ y.values)
auc
```

```
## [1] 0.8552629
```

```
rocr_performance = performance(rocr_prediction, 'tpr','fpr')
plot(rocr_performance, colorize = TRUE, main = 'ROCR Curve')
```




## 3D Scatterplot

```
with(clean_data, {scatterplot3d(x = age,  y = hypertension, z = avg_glucose_level, ma
in = "Stroke Prediction Scatterplot", xlab = "Age", ylab = "Hypertension", zlab = "Av
erage Glucose Level")})
```

# Stroke Prediction Scatterplot