Assignment_5

Barini Simhadri

CONTENTS OF THE PROJECT

a. Data gathering and integration
b. Data Cleaning
c. Data Exploration
d. Data Preprocessing
e. Clustering
f. Classification
g. Evaluation
h. Report
i. Reflection

a. Data gathering and integration

Description of Dataset:
The Diabetes Prediction Dataset was utilized for this study, and it contains a comprehensive collection of medical and demographic data from individuals, as well as their diabetes status (positive or negative). negative). The information includes various important characteristics such as age, gender, body mass index (BMI), and height.Hypertension, heart disease, smoking history, HbA1c level, and blood glucose level are all factors to consider.

Goal:
The study's objective is to develop a credible model for predicting diabetes risk in patients based on their genetics.medical history and demographic information. These projections can be incredibly beneficial to healthcare practitioners.in identifying persons at risk of developing diabetes. Pharmaceutical companies are particularly interested.These projections are useful for client profiling and developing individualized treatment plans.

Importing necessary libraries

```
library(rpart)
library(tidyverse)
library(caret)
library(ggplot2)
library(dplyr)
library(rattle)
library(ROSE)
library(moments)
library(caret)
library(stats)
library(factoextra)
library(e1071)
```

Read the data

```
df = read.csv2("C:/Users/bunty/Desktop/Fundaproject/diabetes.csv", header = T, sep = ",")
```

```
head(df)
```

```
##    gender  age hypertension heart_disease smoking_history   bmi HbA1c_level
## 1  Female 80.0            0             1           never 25.19         6.6
## 2  Female 54.0            0             0        No Info 27.32         6.6
## 3    Male 28.0            0             0           never 27.32         5.7
## 4  Female 36.0            0             0         current 23.45         5.0
## 5    Male 76.0            1             1         current 20.14         4.8
## 6  Female 20.0            0             0           never 27.32         6.6
##    blood_glucose_level diabetes
## 1                  140        0
## 2                   80        0
## 3                  158        0
## 4                  155        0
## 5                  155        0
## 6                   85        0
```

```
str(df)
```

```
## 'data.frame':    100000 obs. of  9 variables:
##  $ gender              : chr  "Female" "Female" "Male" "Female" ...
##  $ age                 : chr  "80.0" "54.0" "28.0" "36.0" ...
##  $ hypertension        : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ heart_disease       : int  1 0 0 0 1 0 0 0 0 0 ...
##  $ smoking_history     : chr  "never" "No Info" "never" "current" ...
##  $ bmi                 : chr  "25.19" "27.32" "27.32" "23.45" ...
##  $ HbA1c_level         : chr  "6.6" "6.6" "5.7" "5.0" ...
##  $ blood_glucose_level: int  140 80 158 155 155 85 200 85 145 100 ...
##  $ diabetes            : int  0 0 0 0 0 0 1 0 0 0 ...
```

```
summary(df)
```

```
##     gender              age            hypertension     heart_disease
##  Length:100000      Length:100000      Min.   :0.00000   Min.   :0.00000
##  Class :character   Class :character   1st Qu.:0.00000   1st Qu.:0.00000
##  Mode  :character   Mode  :character   Median :0.00000   Median :0.00000
##                                        Mean   :0.07485   Mean   :0.03942
##                                        3rd Qu.:0.00000   3rd Qu.:0.00000
##                                        Max.   :1.00000   Max.   :1.00000
##  smoking_history        bmi            HbA1c_level        blood_glucose_level
##  Length:100000      Length:100000      Length:100000      Min.   : 80.0
##  Class :character   Class :character   Class :character   1st Qu.:100.0
##  Mode  :character   Mode  :character   Mode  :character   Median :140.0
##                                                           Mean   :138.1
##                                                           3rd Qu.:159.0
##                                                           Max.   :300.0
##     diabetes
##  Min.   :0.000
```

```
##   1st Qu.:0.000
##  Median :0.000
## Mean :0.085 ##
3rd Qu.:0.000    ##
Max.     :1.000
```

b. Data Cleaning

Some of the columns are character and integer datatypes, which must be transformed for age to integer, bmi and HbA1c_level to numeric, and our goal variable to factor 0/1.

```
df$age = as.integer(df$age)
df$bmi = as.numeric(df$bmi)
df$HbA1c_level = as.numeric(df$HbA1c_level)
df$diabetes = as.factor(df$diabetes)
df$hypertension = as.character(df$hypertension)
df$heart_disease = as.character(df$heart_disease)
```

```
str(df)
```

```
## 'data.frame':    100000 obs. of  9 variables:
##  $ gender              : chr  "Female" "Female" "Male" "Female" ...
##  $ age                 : int  80 54 28 36 76 20 44 79 42 32 ...
##  $ hypertension        : chr  "0" "0" "0" "0" ...
##  $ heart_disease       : chr  "1" "0" "0" "0" ...
##  $ smoking_history     : chr  "never" "No Info" "never" "current" ...
##  $ bmi                 : num  25.2 27.3 27.3 23.4 20.1 ...
##  $ HbA1c_level         : num  6.6 6.6 5.7 5 4.8 6.6 6.5 5.7 4.8 5 ...
##  $ blood_glucose_level: int  140 80 158 155 155 85 200 85 145 100 ...
##  $ diabetes            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...
```

```
summary(df)
```

```
##     gender               age         hypertension        heart_disease
##  Length:100000      Min.   : 0.00   Length:100000      Length:100000
##  Class :character   1st Qu.:24.00   Class :character   Class :character
##  Mode  :character   Median :43.00   Mode  :character   Mode  :character
##                     Mean   :41.88
##                     3rd Qu.:60.00
##                     Max.   :80.00
##  smoking_history        bmi         HbA1c_level     blood_glucose_level
##  Length:100000      Min.   :10.01   Min.   :3.500   Min.   : 80.0
##  Class :character   1st Qu.:23.63   1st Qu.:4.800   1st Qu.:100.0
##  Mode  :character   Median :27.32   Median :5.800   Median :140.0
##                     Mean   :27.32   Mean   :5.528   Mean   :138.1
##                     3rd Qu.:29.58   3rd Qu.:6.200   3rd Qu.:159.0
##                     Max.   :95.69   Max.   :9.000   Max.   :300.0
##  diabetes
```

```
##    0:91500
##   1:   8500
##
##
##
##
##
##         0         1
##  91500    8500
```

Based on the data above, we may conclude that there is a significant class imbalance, which will provide a challenge to our model's construction. In order to improve accuracy, we will attack this challenge further in data exploration and sample out the data to smaller datasets. Furthermore, the maximum values of BMI and Blood Glucose Level appear to be too high for now; we must hunt for outliers later.

Checking the data set for missing values.

```
table(df$diabetes)
```

```
df[rowSums(is.na(df)) > 0, ]
```

```
## [1] gender                age                   hypertension
## [4] heart_disease         smoking_history       bmi
## [7] HbA1c_level           blood_glucose_level   diabetes
## <0 rows> (or 0-length row.names)
```

we can see, there are no missing values.

Checking for any duplicate number of rows. We can see there are 3888 duplicate rows.

```
duplicate = sum(duplicated(df))
duplicate
```

```
## [1] 3888
```

Removing all the duplicate rows

```
df <- subset(df, !duplicated(df))
dim(df)
```

```
## [1] 96112     9
```

```
table(df$gender)
```

```
##
## Female   Male  Other
##  56142  39952     18
```

removing unnecessary gender = Other.

```
df = df[df$gender!="Other", ]
```

There are 5 categories for smoking history with No info and never has greater percentage. we can reduce it to 3 categories for simplicity.

```
table(df$smoking_history)
```

```
##
##         current            ever          former           never  No Info not current
##            9197            3997            9299           34395           32847            6359
```

It is reduced to 3 categories.

```
df<-  df%>%
  mutate(smoking_history  =  case_when(
    smoking_history  %in%  c("never",  "No  Info")  ~  "non-smoker",
    smoking_history  %in%  c("current")  ~  "current",
    smoking_history  %in%  c("ever","former","not  current")  ~  "past_smoker"
  ))

table(df$smoking_history)
```

```
##
##         current     non-smoker past_smoker
##            9197           67242           19655
```

/

c. Data Exploration

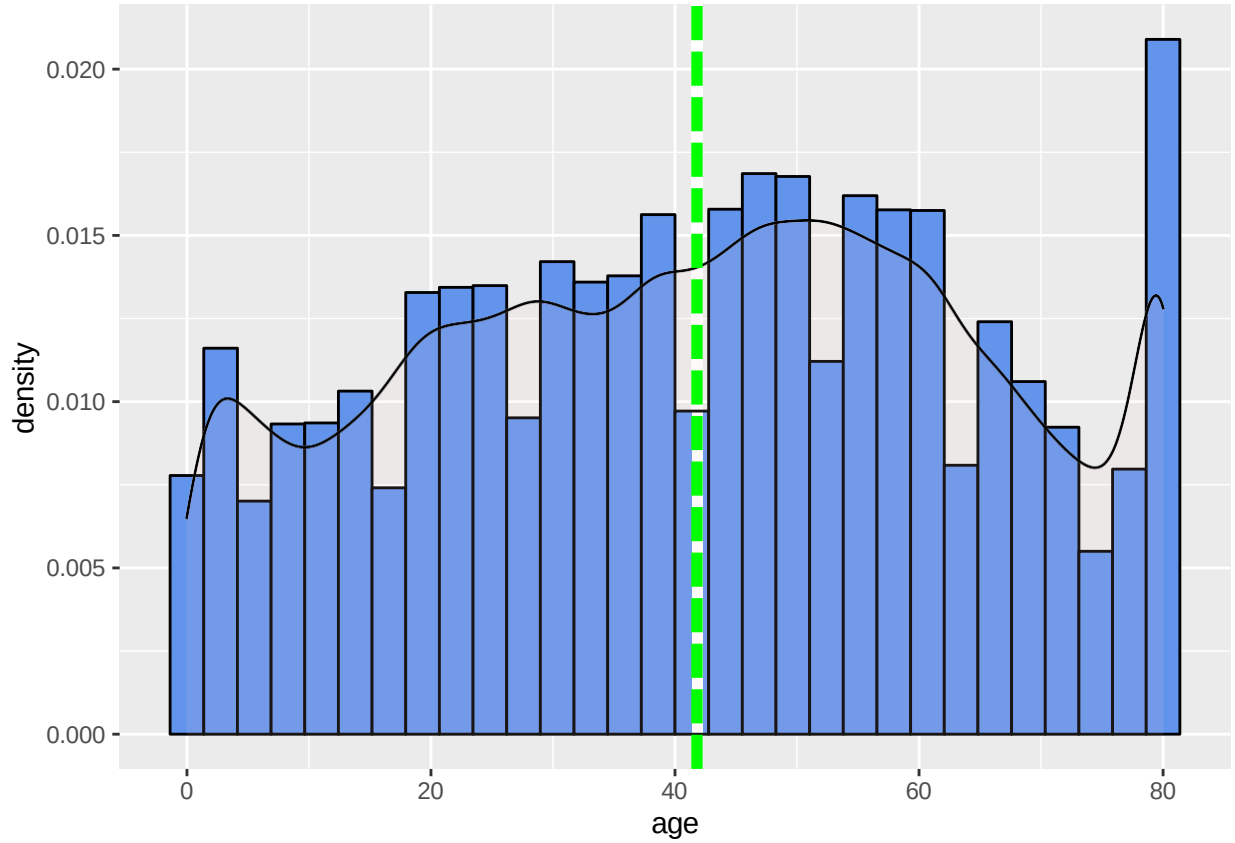
We will visualize the data set with different plots.
Numerical - age, bmi, HbA1c_level, blood_glucose_level
Categorical - hypertension, heart_disease,diabetes, gender

Analysis of age.

```
summary(df$age)
```

```
##    Min.  1st  Qu.   Median     Mean 3rd Qu.    Max.
##     0.0    24.0    43.0    41.8    59.0    80.0
```

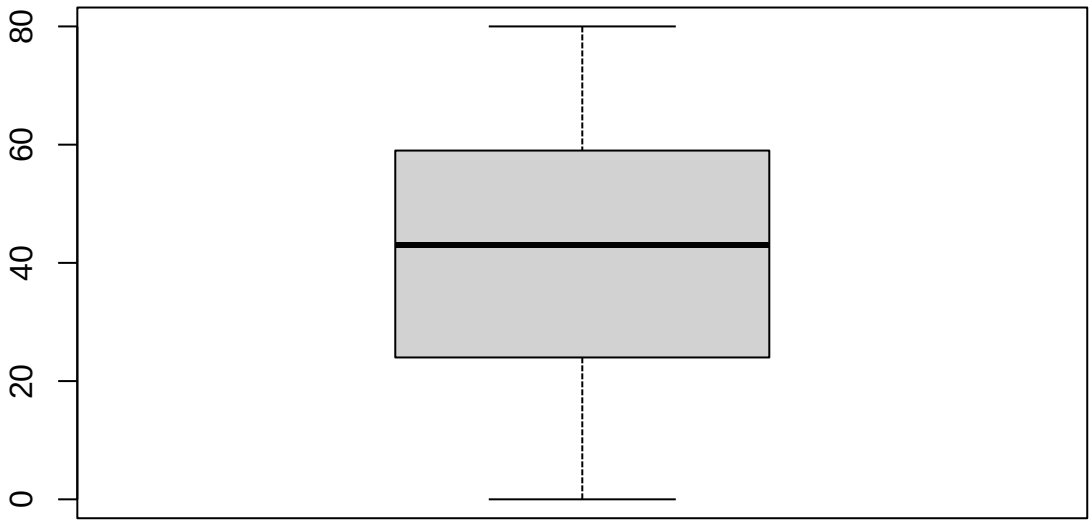
```
ggplot(df,aes(x=age)) + geom_histogram(aes(y = after_stat(density)),color = "black", fill      = "cornflower
```




```
skewness(df$age)
```

```
## [1] -0.06557794
```

```
boxplot(df$age)
```

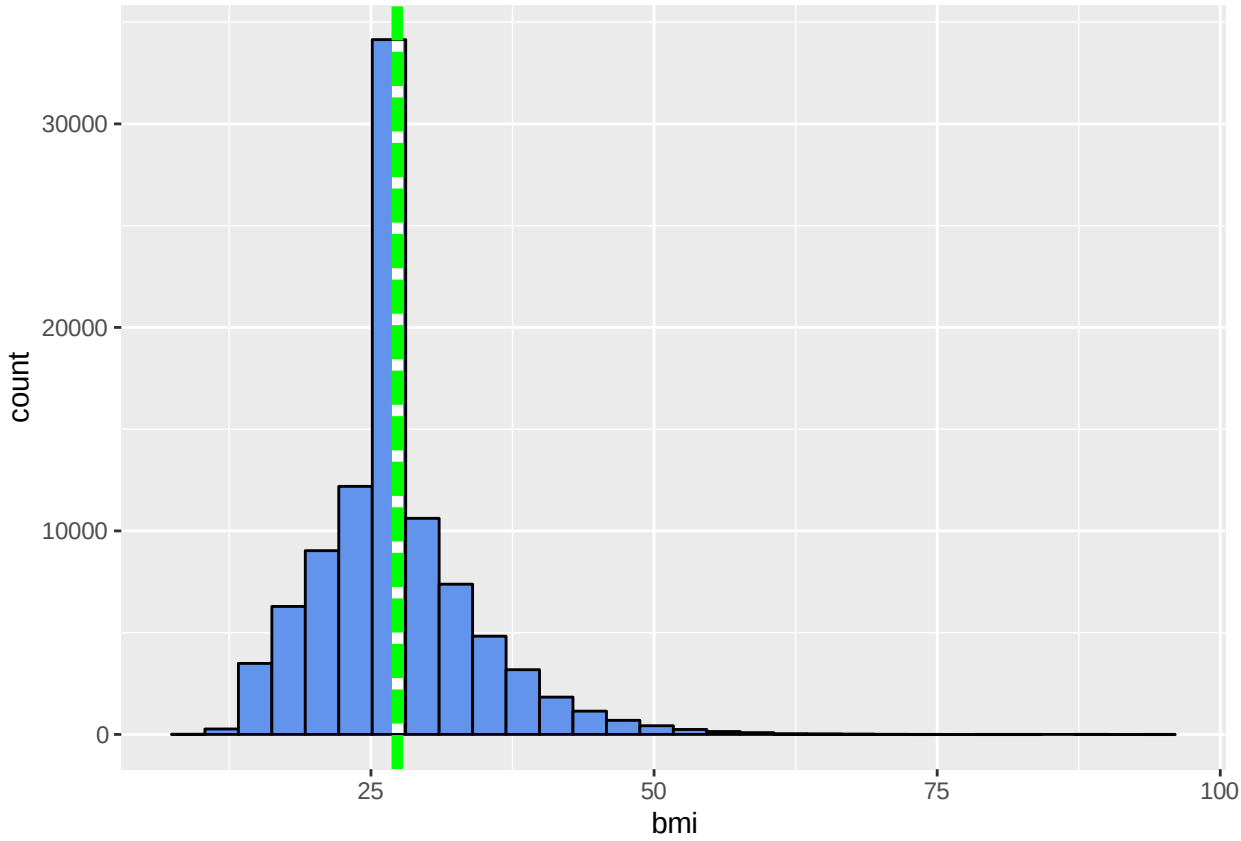


there are No missing values, No outliers ,Data is very slightly left skewed. Analysis of bmi

```
summary(df$bmi)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.01   23.40   27.32   27.32   29.86   95.69
```

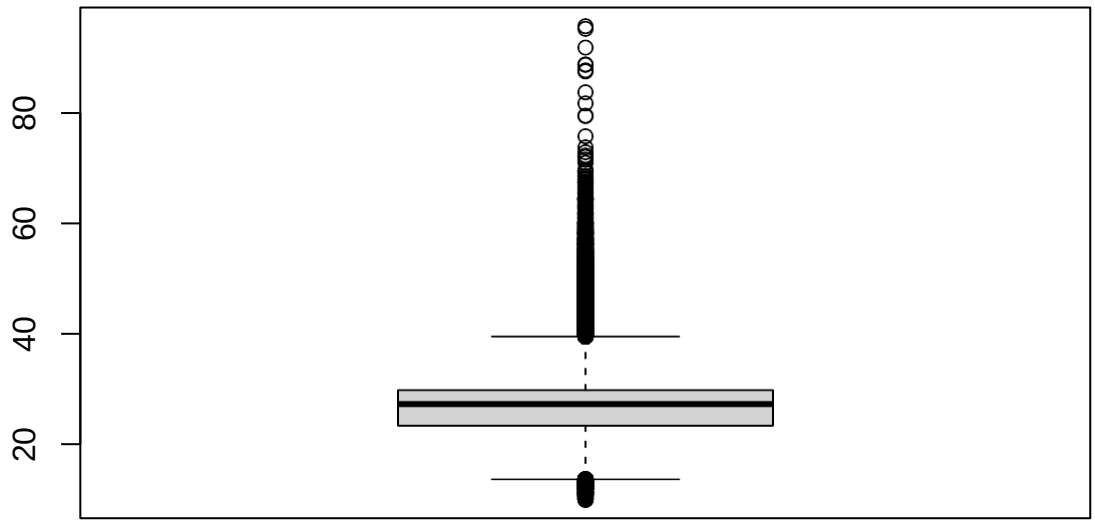
```
ggplot(df,aes(x=bmi)) + geom_histogram(color = "black", fill = "cornflowerblue")+geom_vline(aes(xinterc
```




```
skewness(df$bmi)
```

```
## [1] 1.023884
```

```
boxplot(df$bmi)
```



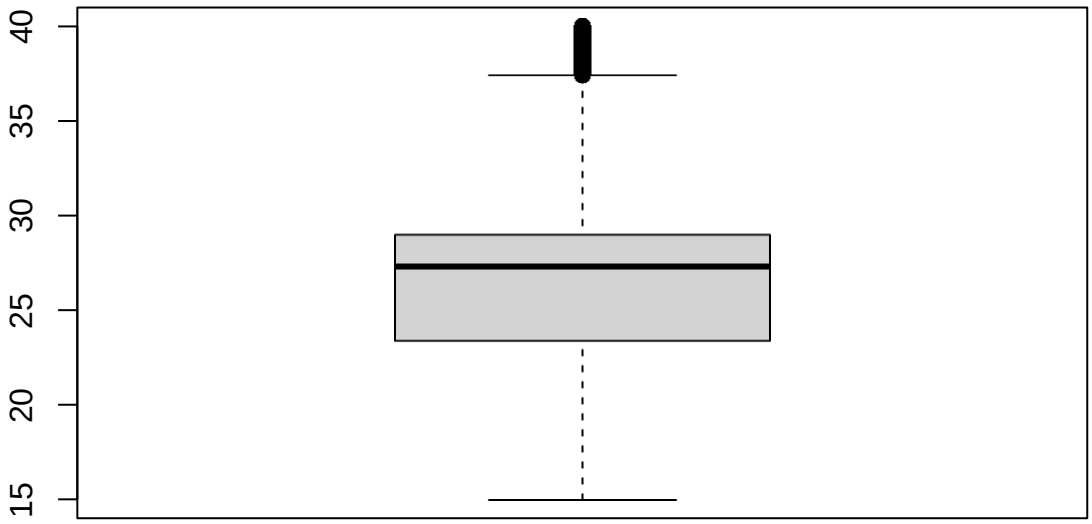
there are outliers in bmi. We can analyze further. Below we can see, there are around 6000 data points which are outliers. we will remove it from the dataset.

```
sum(df$bmi>40 | df$bmi<15)
```

```
## [1] 6068
```

```
df= subset(df,!(df$bmi>40 | df$bmi<15))
```

```
boxplot(df$bmi)
```




```
summary(df$bmi)
```

```
##     Min.  1st Qu.   Median     Mean 3rd Qu.     Max.
##    15.00    23.39    27.32    26.60    29.00    40.00
```

There are many data points where the BMI is same for all of them i.e, 27.32

```
age2 = df[df$bmi == 27.32,]
head(age2)
```

```
##      gender  age  hypertension  heart_disease  smoking_history  bmi  HbA1c_level
## 2    Female   54             0              0      non-smoker  27.32          6.6
## 3      Male   28             0              0      non-smoker  27.32          5.7
## 6    Female   20             0              0      non-smoker  27.32          6.6
## 10   Female   32             0              0      non-smoker  27.32          5.0
## 11   Female   53             0              0      non-smoker  27.32          6.1
## 15   Female   76             0              0      non-smoker  27.32          5.0
##      blood_glucose_level  diabetes
## 2                     80         0
## 3                    158         0
## 6                     85         0
## 10                   100         0
## 11                    85         0
## 15                   160         0
```

It also has some points where age is less than 10 and BMI is 27.32 which is not possible. We will remove such points.

```
age1 = df[df$age<10 & df$bmi == 27.32,]
head(age1)
```

```
##        gender  age  hypertension  heart_disease  smoking_history  bmi  HbA1c_level
## 42       Male    5             0              0      non-smoker  27.32          6.6
## 174      Male    8             0              0      non-smoker  27.32          6.6
## 184      Male    9             0              0      non-smoker  27.32          6.5
## 206      Male    6             0              0      non-smoker  27.32          5.7
## 227      Male    2             0              0      non-smoker  27.32          5.7
## 266    Female    4             0              0      non-smoker  27.32          5.0
##        blood_glucose_level  diabetes
## 42                     130         0
## 174                    155         0
## 184                     85         0
## 206                    200         0
## 227                     85         0
## 266                    140         0
```

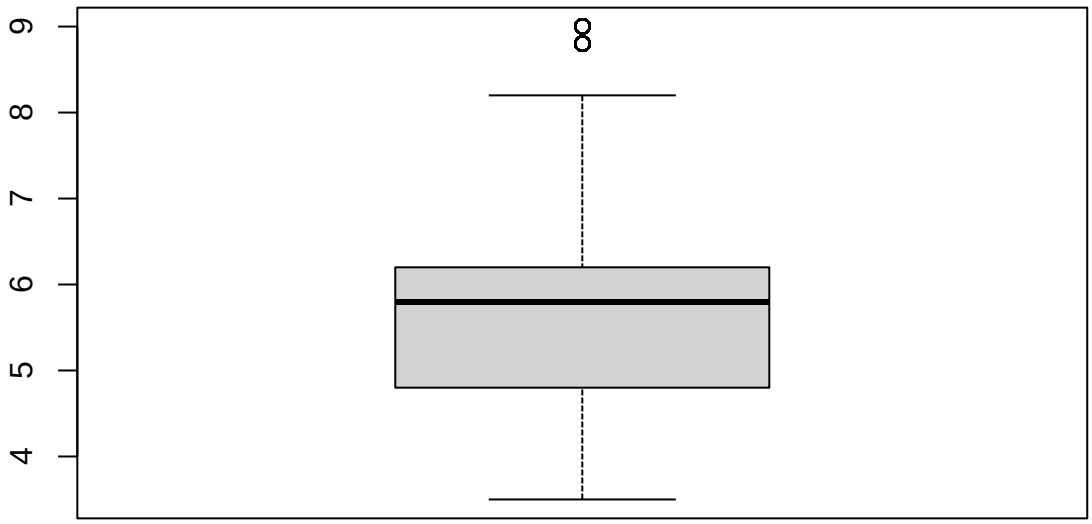
```
df = subset(df,!(df$age<10 & df$bmi == 27.32))
```

Analysis of HbA1c_level

```
summary(df$HbA1c_level)
```

```
##    Min.  1st Qu.   Median    Mean 3rd Qu.    Max.
##   3.500   4.800   5.800   5.523   6.200   9.000
```

```
boxplot(df$HbA1c_level)
```




```
sum(df$HbA1c_level>8.5)
```

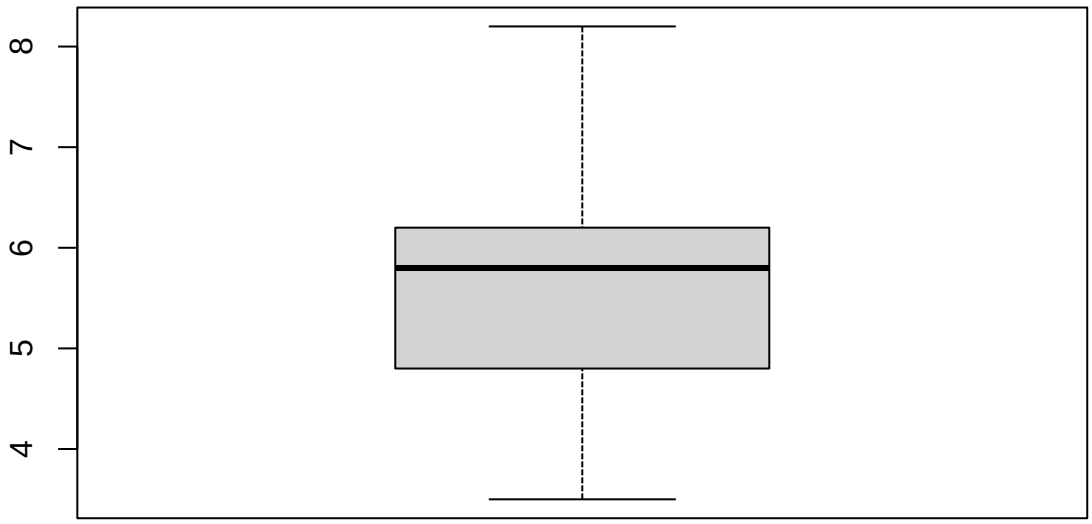
```
## [1] 1154
```

Removing outliers.

```
df = subset(df,!(df$HbA1c_level>8.5))
```

plot after removing outliers.

```
boxplot(df$HbA1c_level)
```



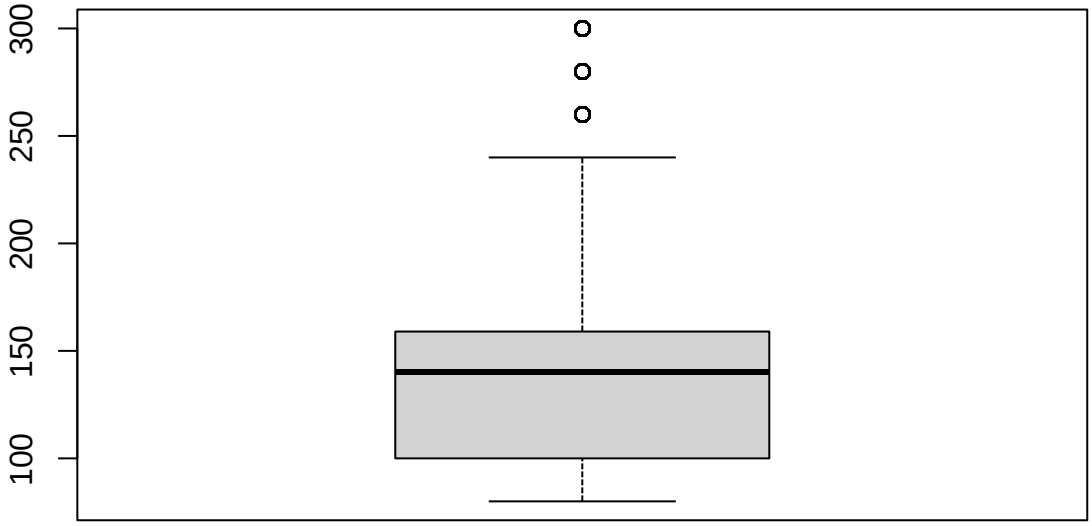
Analysis of Blood glucose level

```
summary(df$blood_glucose_level)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    80.0   100.0   140.0   137.1   159.0   300.0
```

Outliers detected

```
boxplot(df$blood_glucose_level)
```

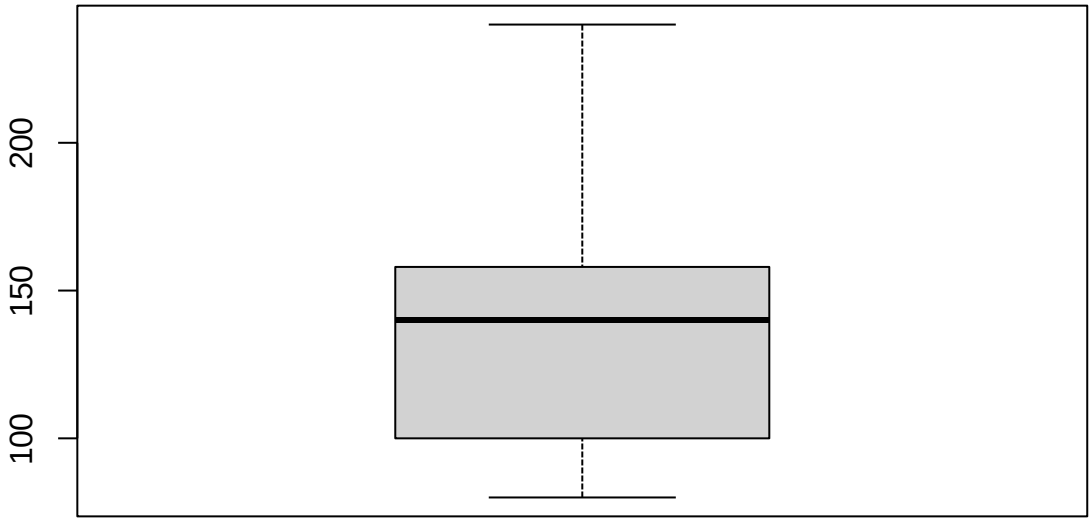



```
sum(df$blood_glucose_level>245)
```

```
## [1] 1473
```

```
df = subset(df,!(df$blood_glucose_level>245))
```

```
boxplot(df$blood_glucose_level)
```



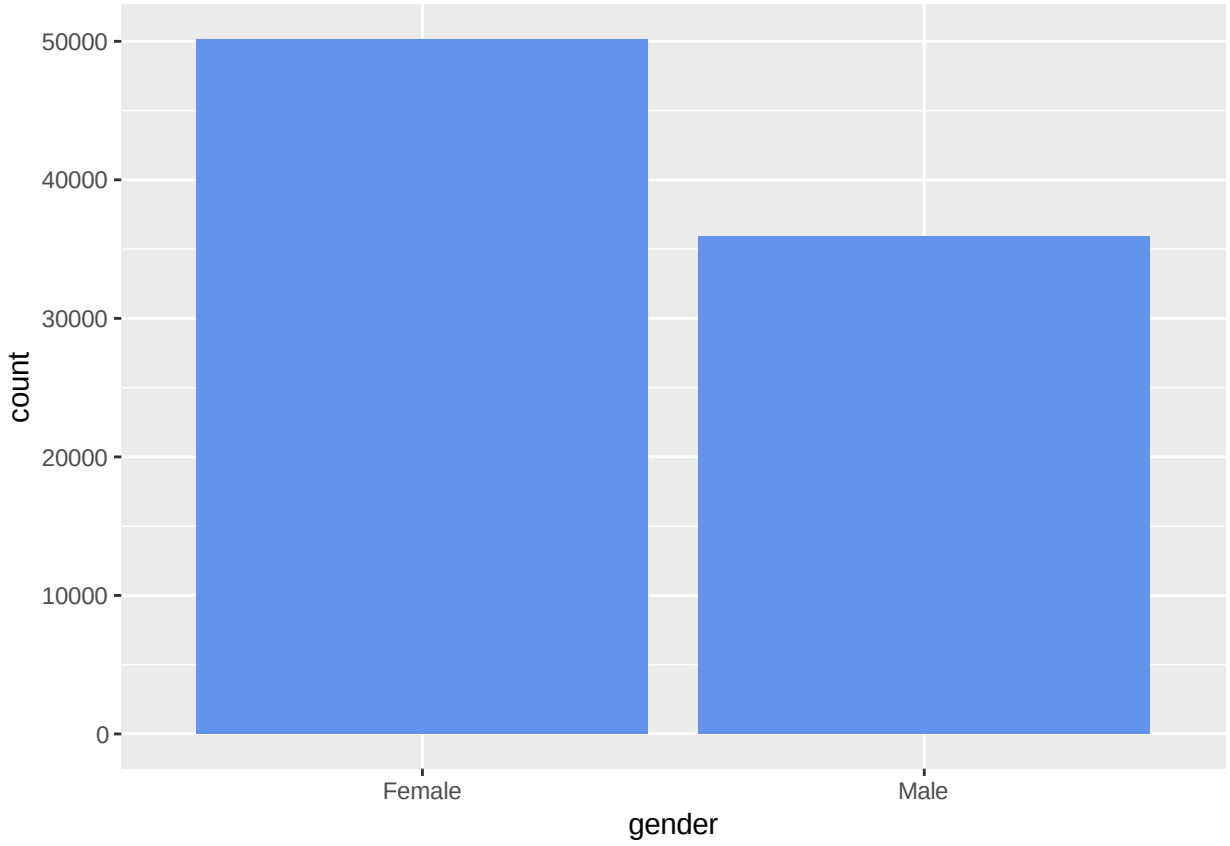

```
skewness(df$blood_glucose_level)
```

```
## [1] 0.1101641
```

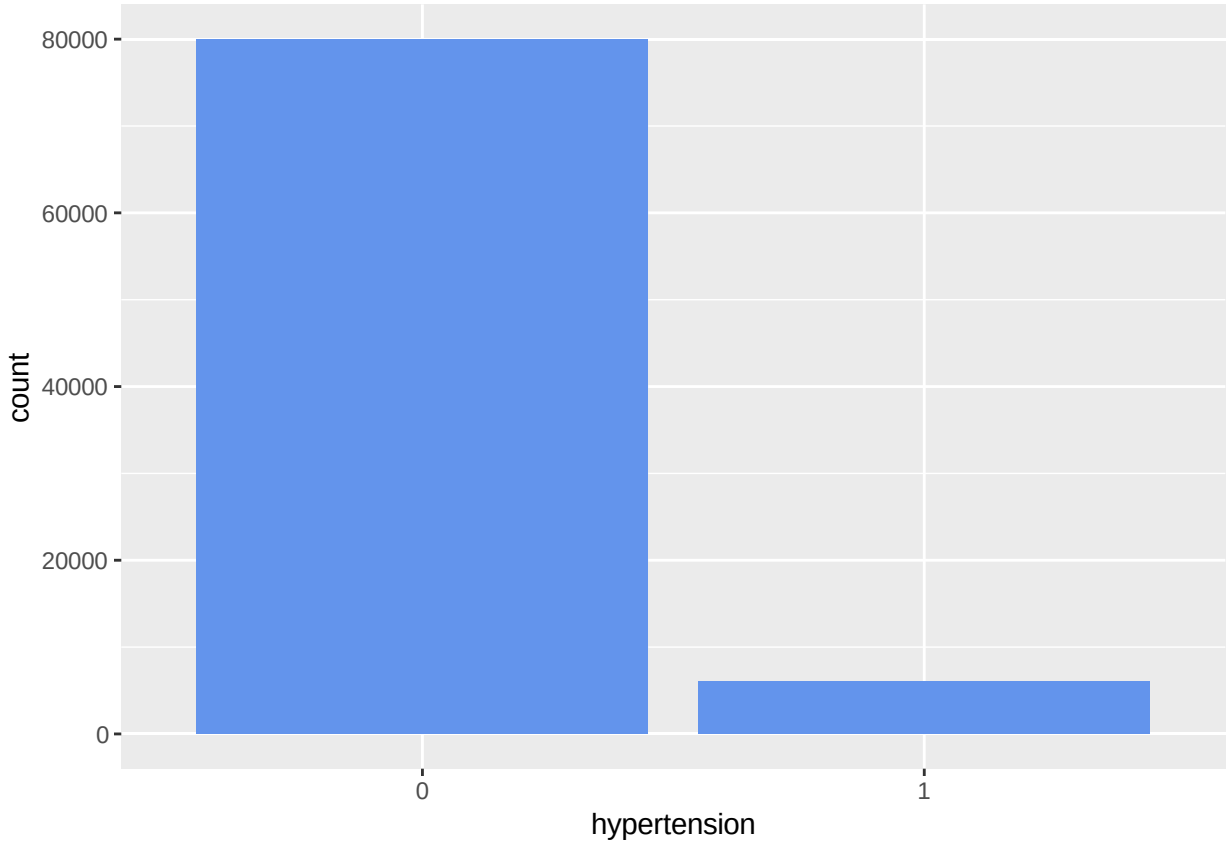
Positvely Skewed ,Has Outliers and removed it, 50% people fall in the 100 to 160 range

Analysis of categorical columns

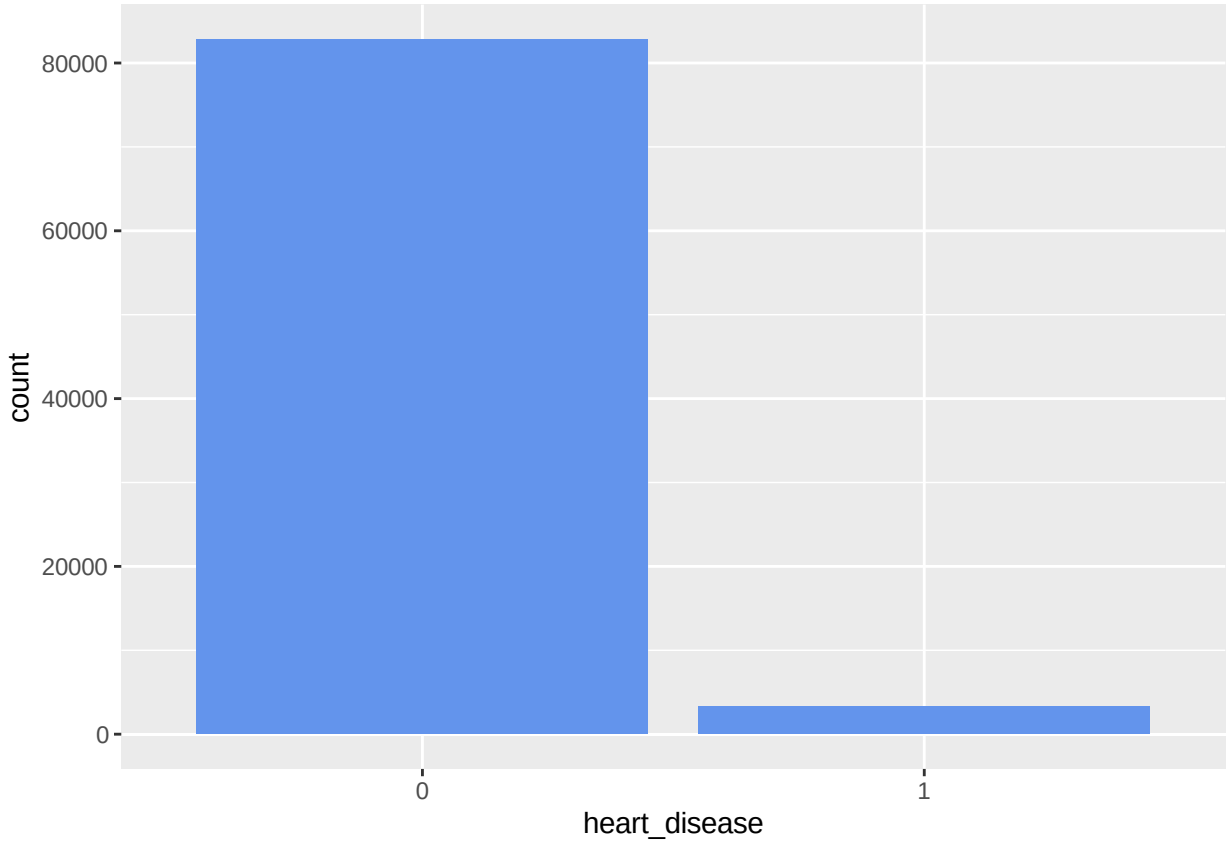
```
ggplot(df,aes(x=gender)) + geom_bar(fill="cornflowerblue")
```



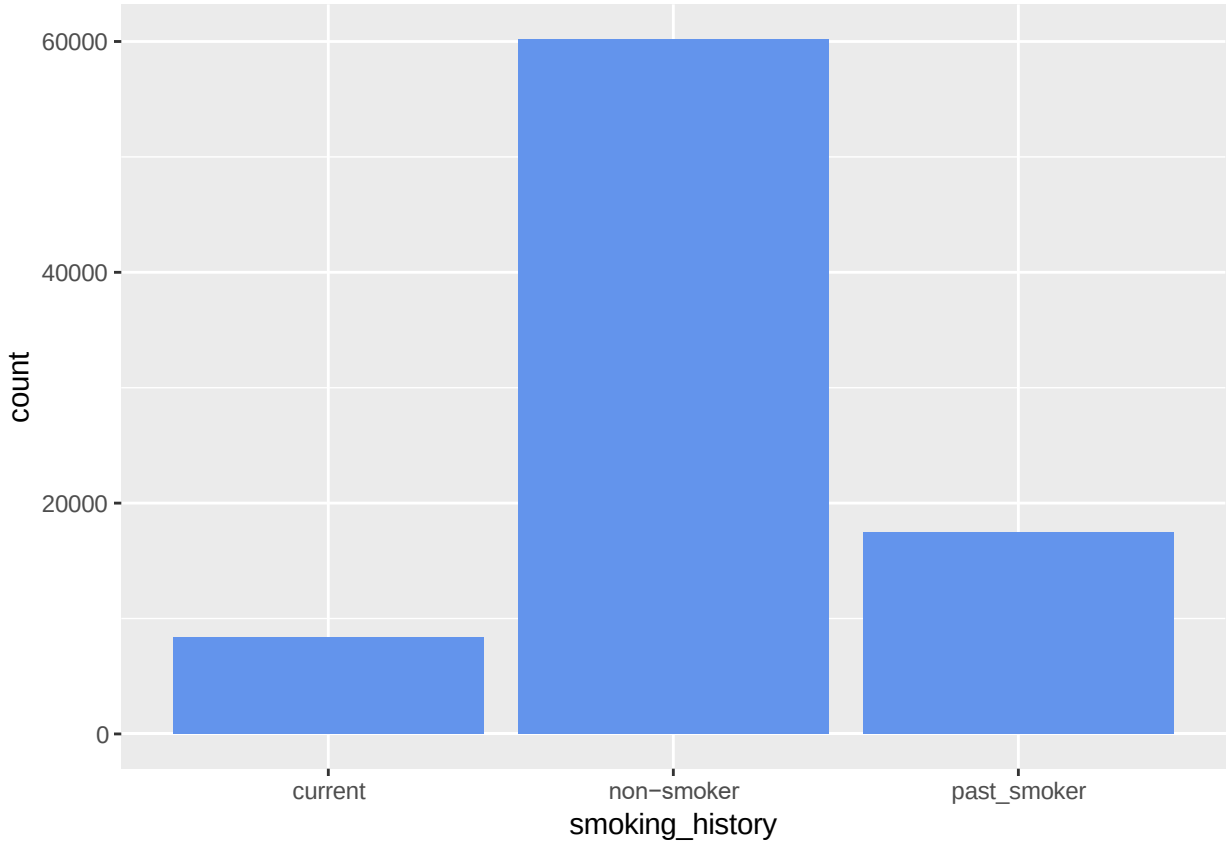

```
ggplot(df,aes(x=hypertension)) + geom_bar(fill="cornflowerblue")
```



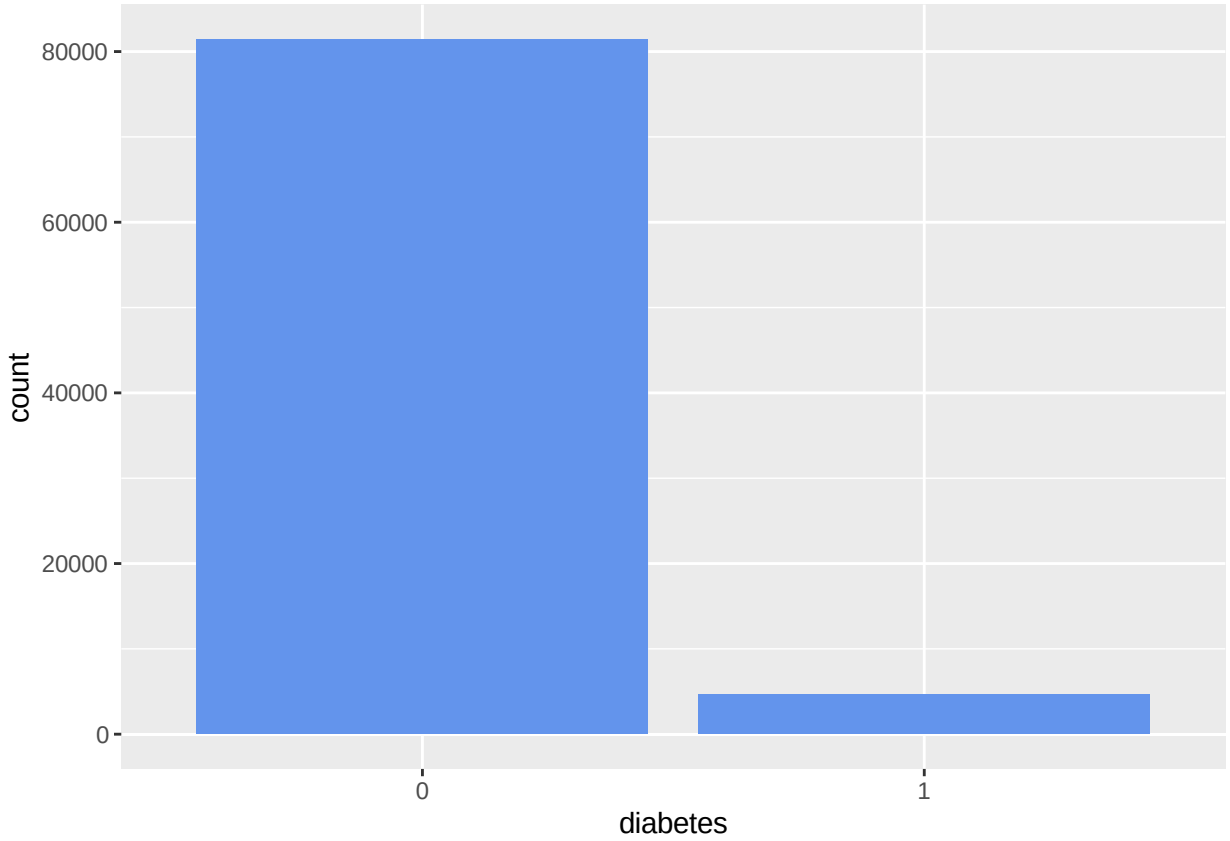

```
ggplot(df,aes(x=heart_disease)) + geom_bar(fill="cornflowerblue")
```



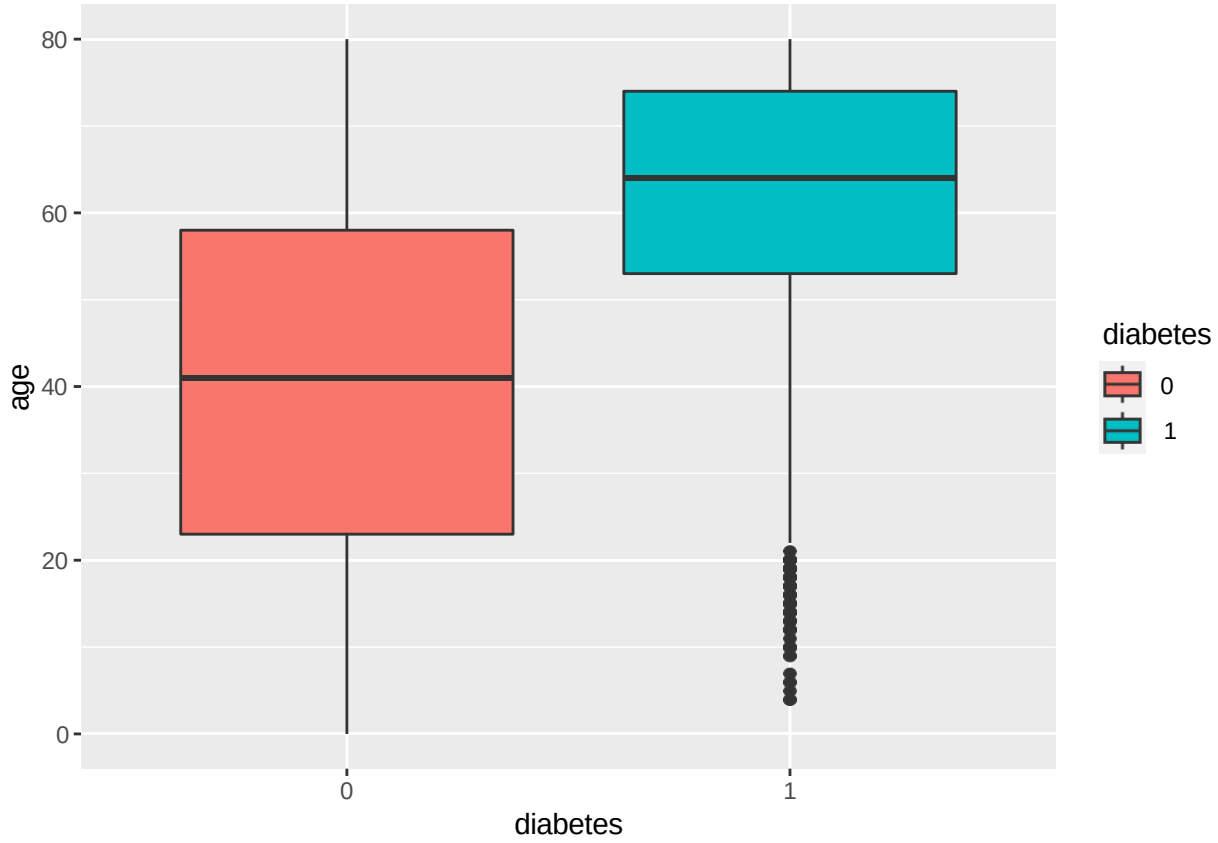

```
ggplot(df,aes(x=smoking_history))  +  geom_bar(fill="cornflowerblue")
```




```
ggplot(df,aes(x=diabetes)) + geom_bar(fill="cornflowerblue")
```

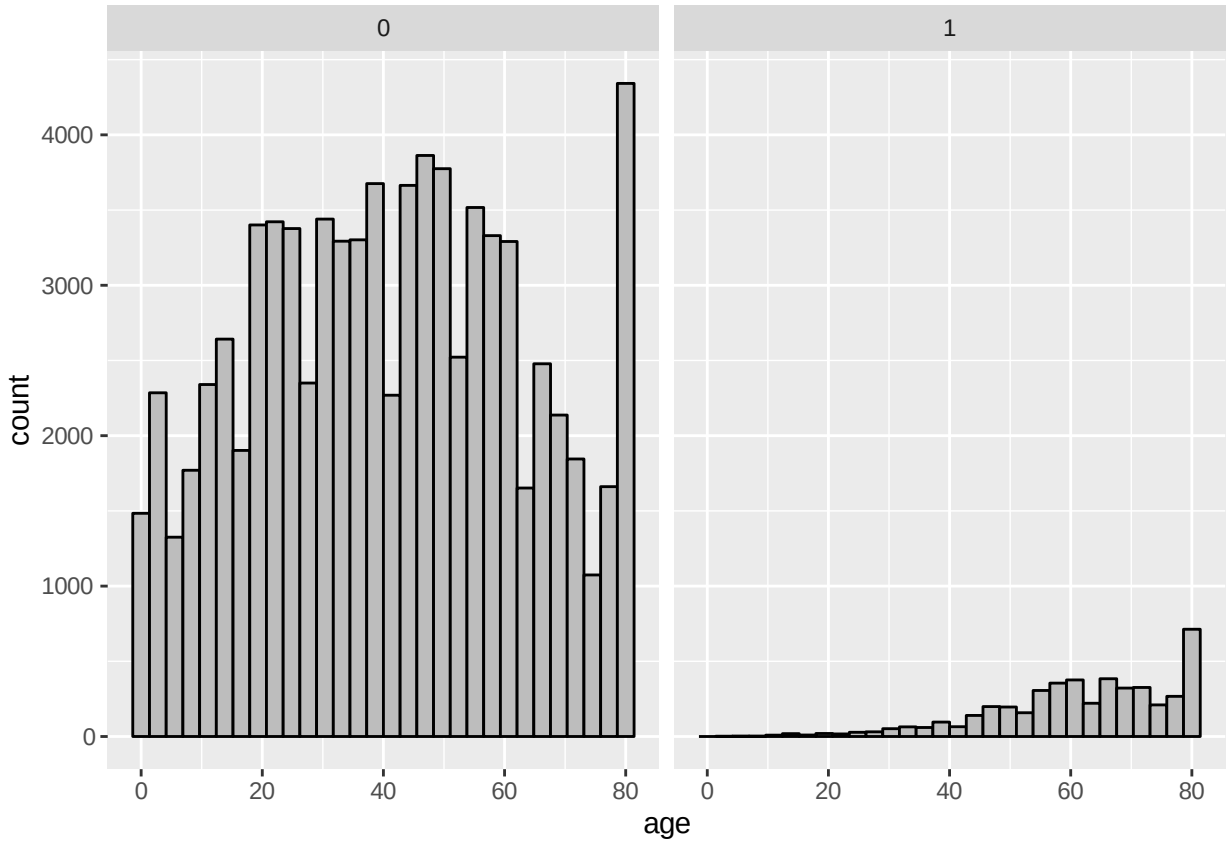



```
ggplot(df,aes(x=diabetes,y=age,fill=diabetes))+geom_boxplot()
```

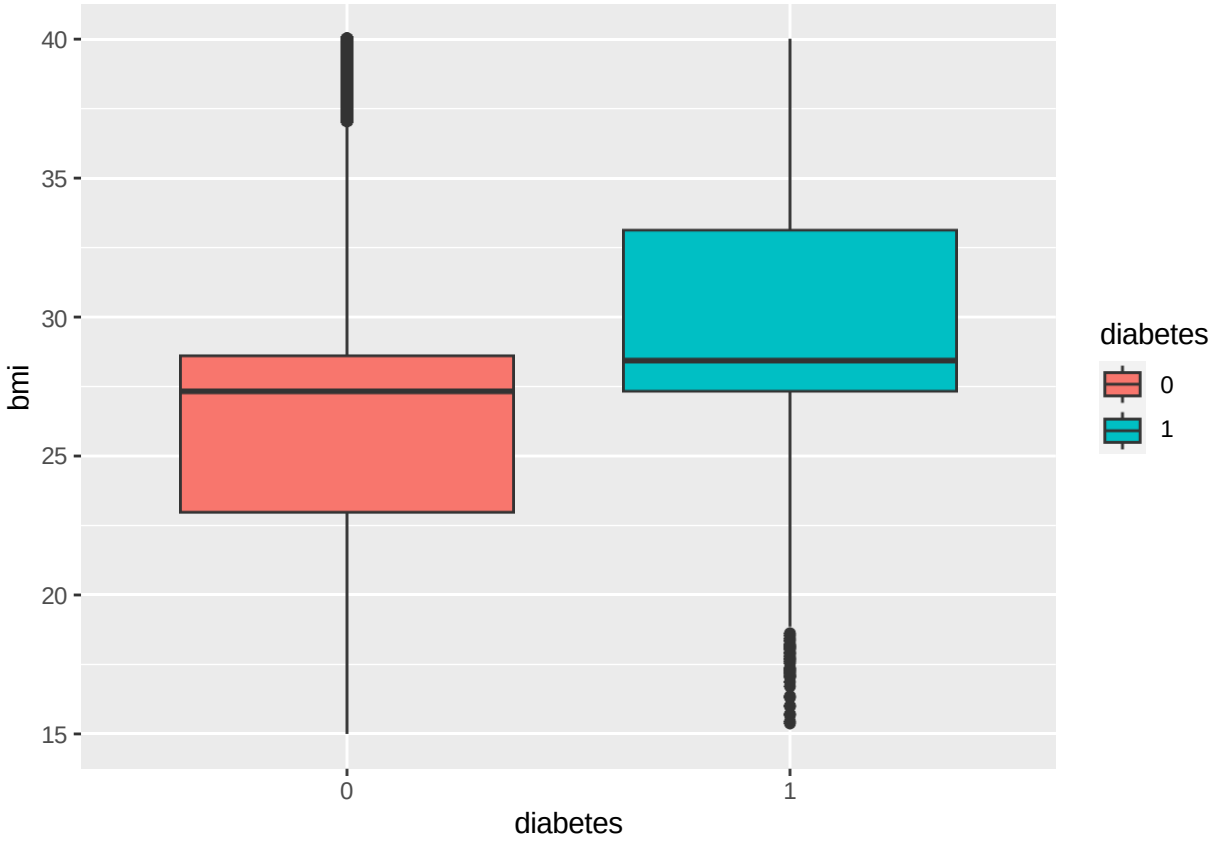


Diabetes is more common among elderly adults.The diabetes risk curve begins to climb gradually in your 30s and reaches a peak around the age of 60.This is consistent with real-world evidence.

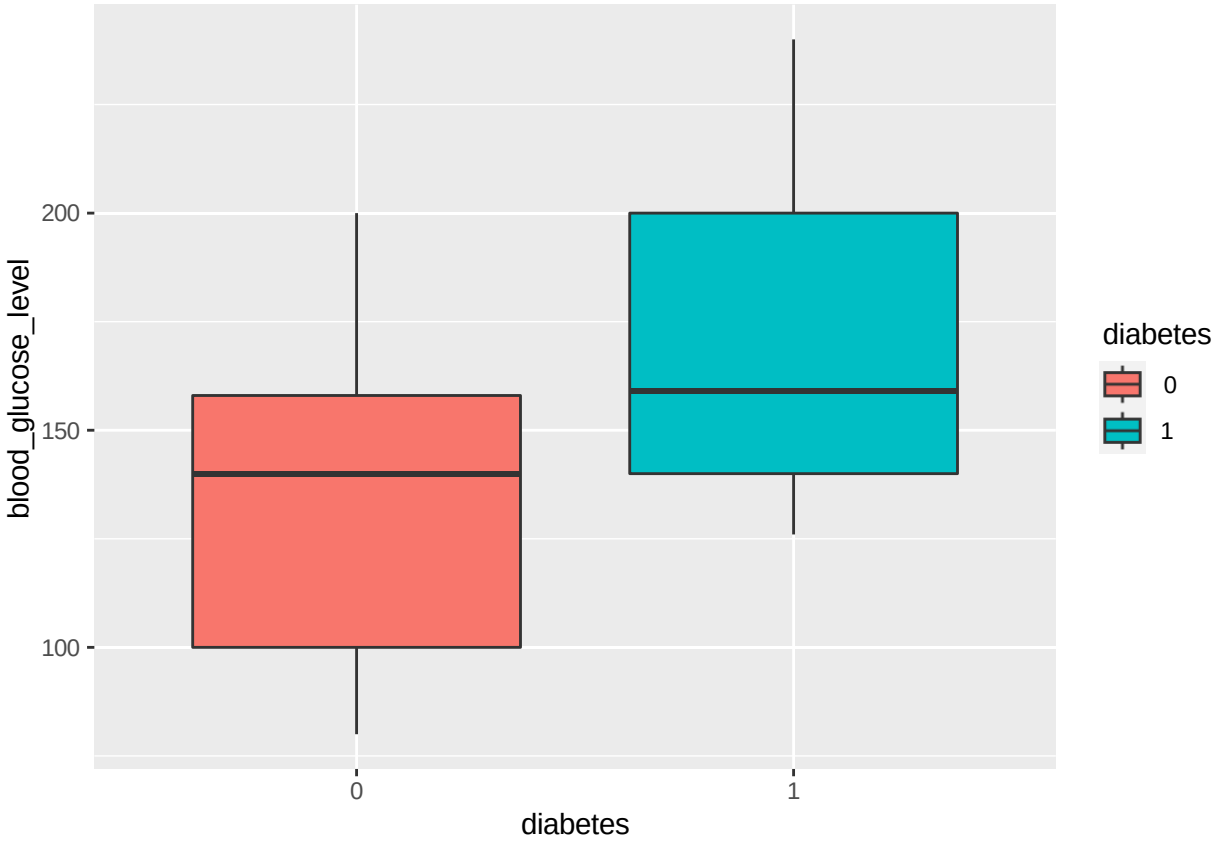
```
ggplot(df,aes(x=age)) + geom_histogram(color="black",fill="grey")+facet_wrap(~diabetes)
```




```
ggplot(df,aes(x=diabetes,y=bmi,fill=diabetes))+geom_boxplot()
```

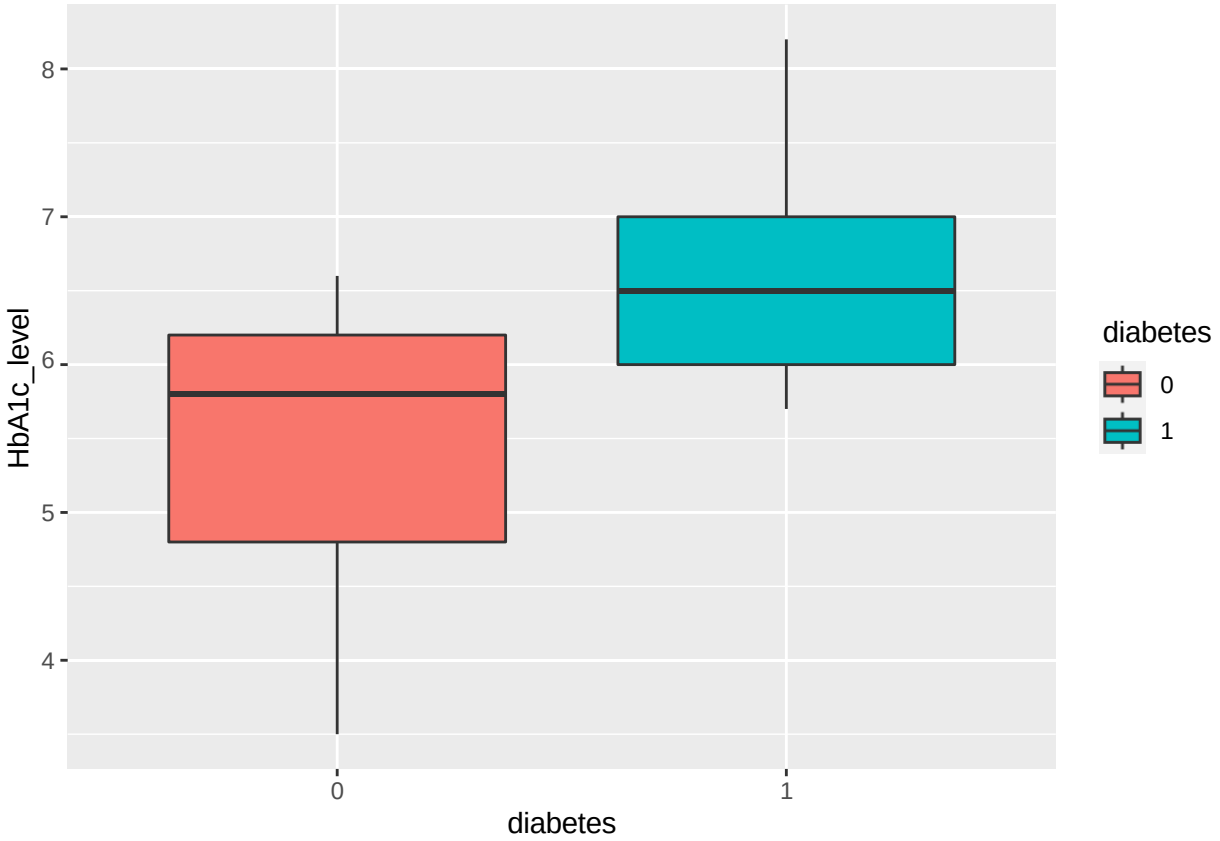



```
ggplot(df,aes(x=diabetes,y=blood_glucose_level,fill=diabetes))+geom_boxplot()
```



With increase in blood glucose level, the chance of diabetes increases the people with diabetes have a blood glucose level of around 160 on average.

```
ggplot(df,aes(x=diabetes,y=HbA1c_level,fill=diabetes))+geom_boxplot()
```

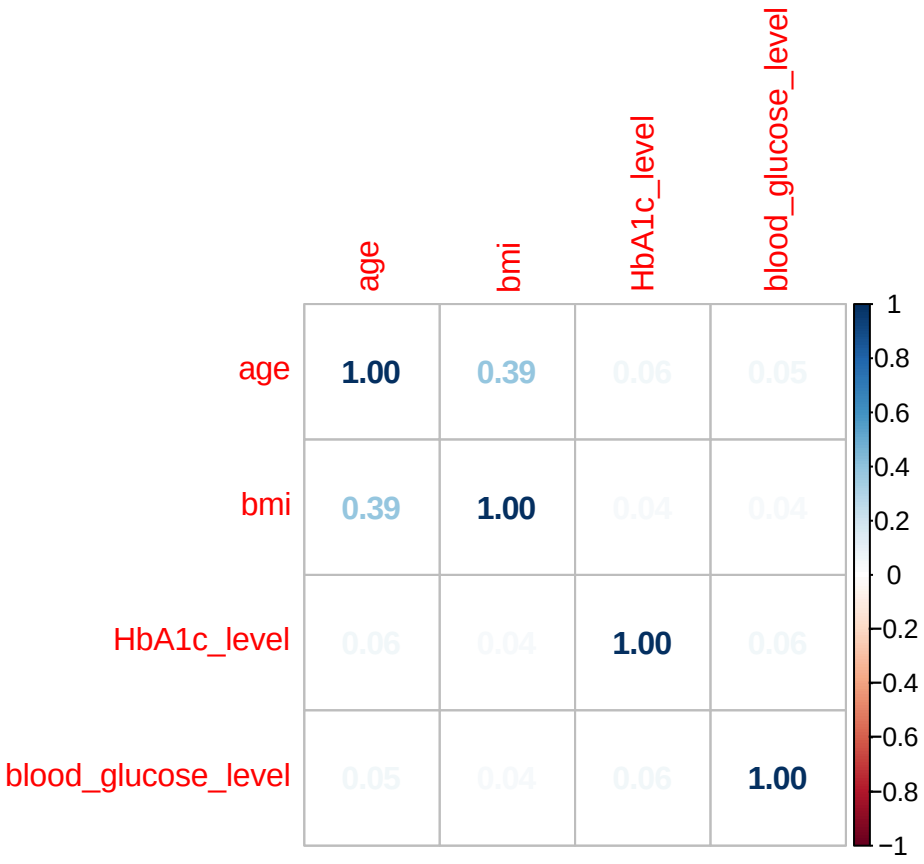


With increase in HbA1c level, the chance of diabetes increases .People who have diabetes have a median HbA1c value of around 6.7

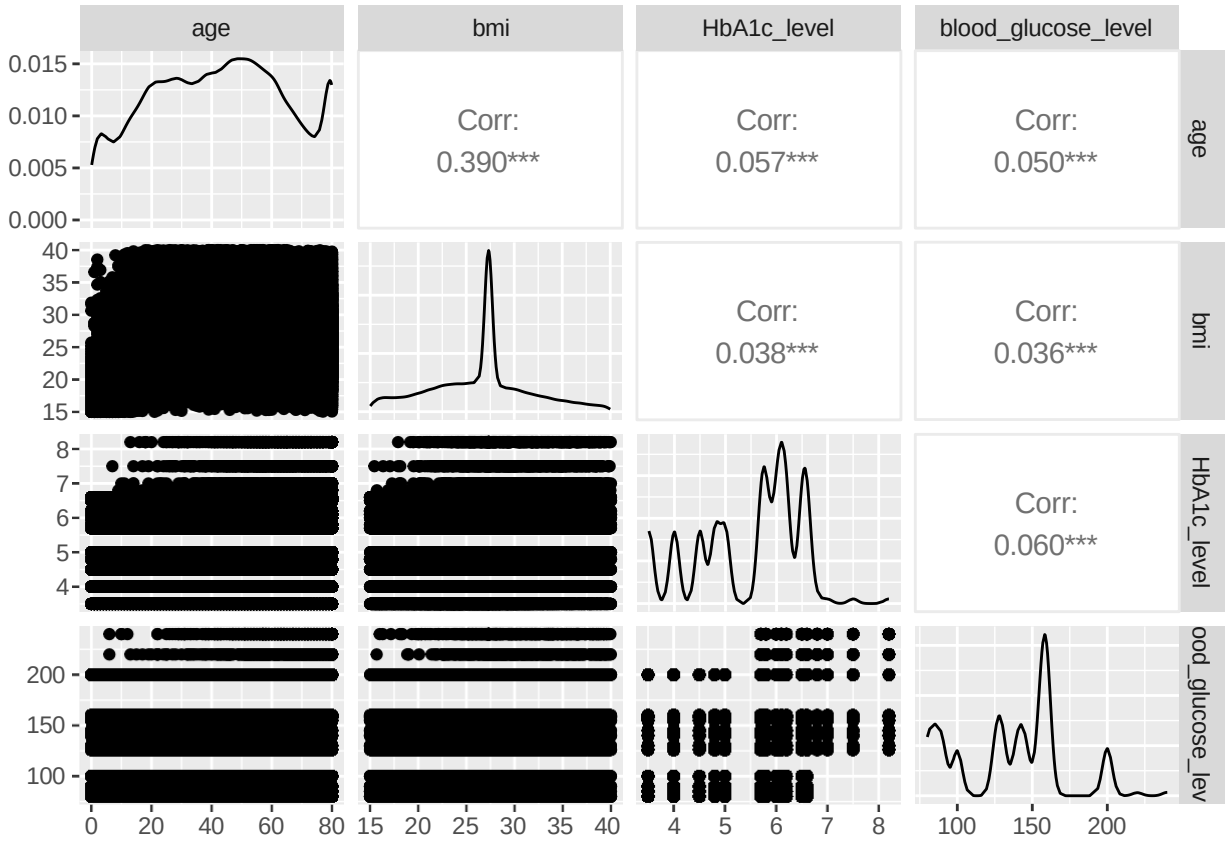
```
library(corrplot)
matrix1 <- cor(select_if(df,is.numeric))
round(matrix1,3)
```

```
##                              age   bmi  HbA1c_level  blood_glucose_level
##  age                       1.000 0.390        0.057                0.050
##  bmi                       0.390 1.000        0.038                0.036
##  HbA1c_level               0.057 0.038        1.000                0.060
##  blood_glucose_level       0.050 0.036        0.060                1.000
```

```
corrplot(matrix1,    method="number")
```




```
library(GGally)
d_num = select_if(df,is.numeric)
ggpairs(d_num)
```



The scatter matrix view demonstrates the connection between the variables in the data set. This viewpoint is important because it helps us to find the association that would otherwise be difficult to observe when looking at the distribution.We can see that the variables have a positive association.

d. Data Preprocessing

converting data into dummy variables as there are categorical values in the data set.

```
dff<- ovun.sample(diabetes~., data=df, method = "both",
                  p = 0.5,
                  seed = 222,
                  N = 800)$data
```

```
table(dff$diabetes)
```

```
##
##   0   1
## 391 409
```

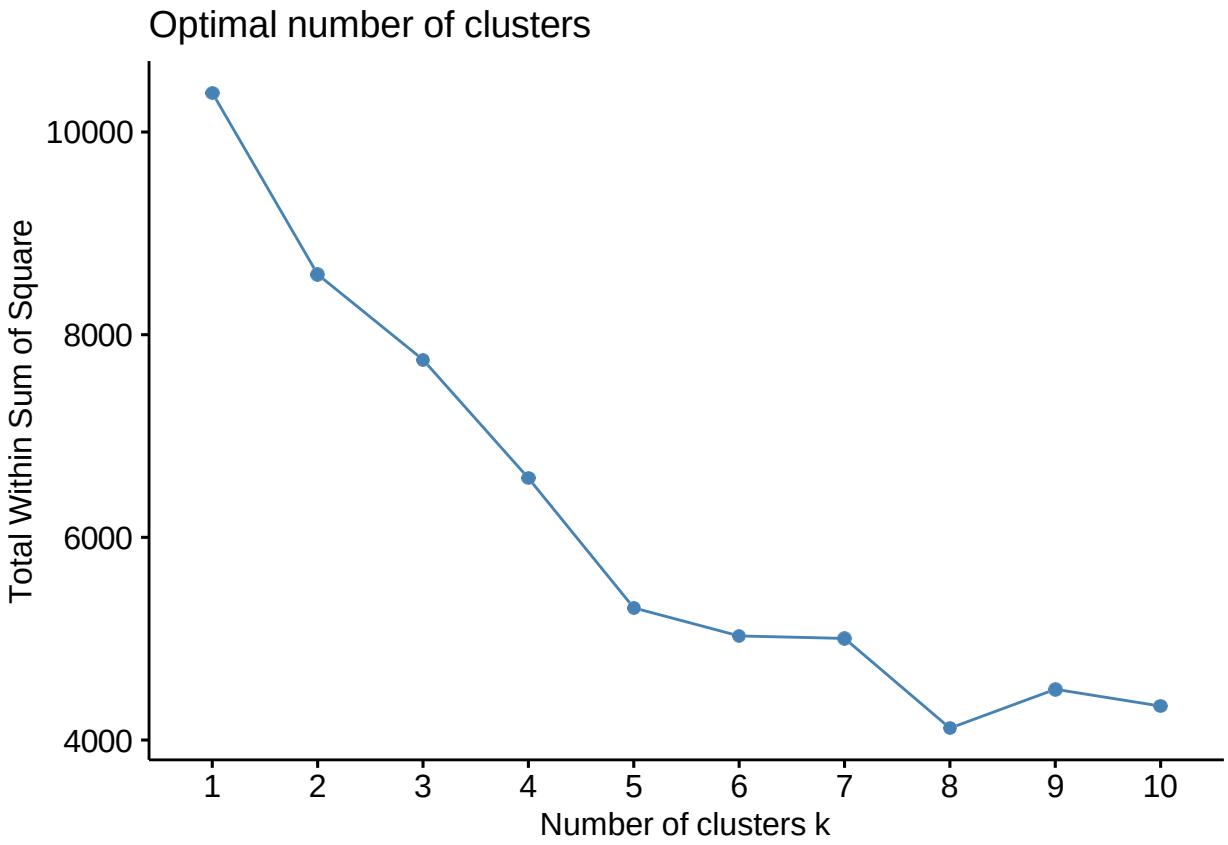
```
dummy <- dummyVars(diabetes~., data = dff)
dummies <- as.data.frame(predict(dummy, newdata = dff))
head(dummies)
```

```
##   genderFemale genderMale age hypertension0 hypertension1 heart_disease0
## 1            1          0  31             1             0              1
## 2            0          1  65             1             0              1
## 3            1          0  67             1             0              1
## 4            0          1  65             1             0              1
## 5            1          0  20             1             0              1
## 6            0          1  18             1             0              1
##   heart_disease1 smoking_historycurrent smoking_historynon-smoker
## 1              0                      0                         1
## 2              0                      1                         0
## 3              0                      0                         1
## 4              0                      0                         1
## 5              0                      0                         1
## 6              0                      0                         1
##   smoking_historypast_smoker   bmi HbA1c_level blood_glucose_level
## 1                          0 27.32         6.2                 159
## 2                          0 27.32         6.5                 140
## 3                          0 28.30         4.0                 126
## 4                          0 27.32         5.0                 200
## 5                          0 20.24         6.6                 159
## 6                          0 27.32         6.2                 159
```

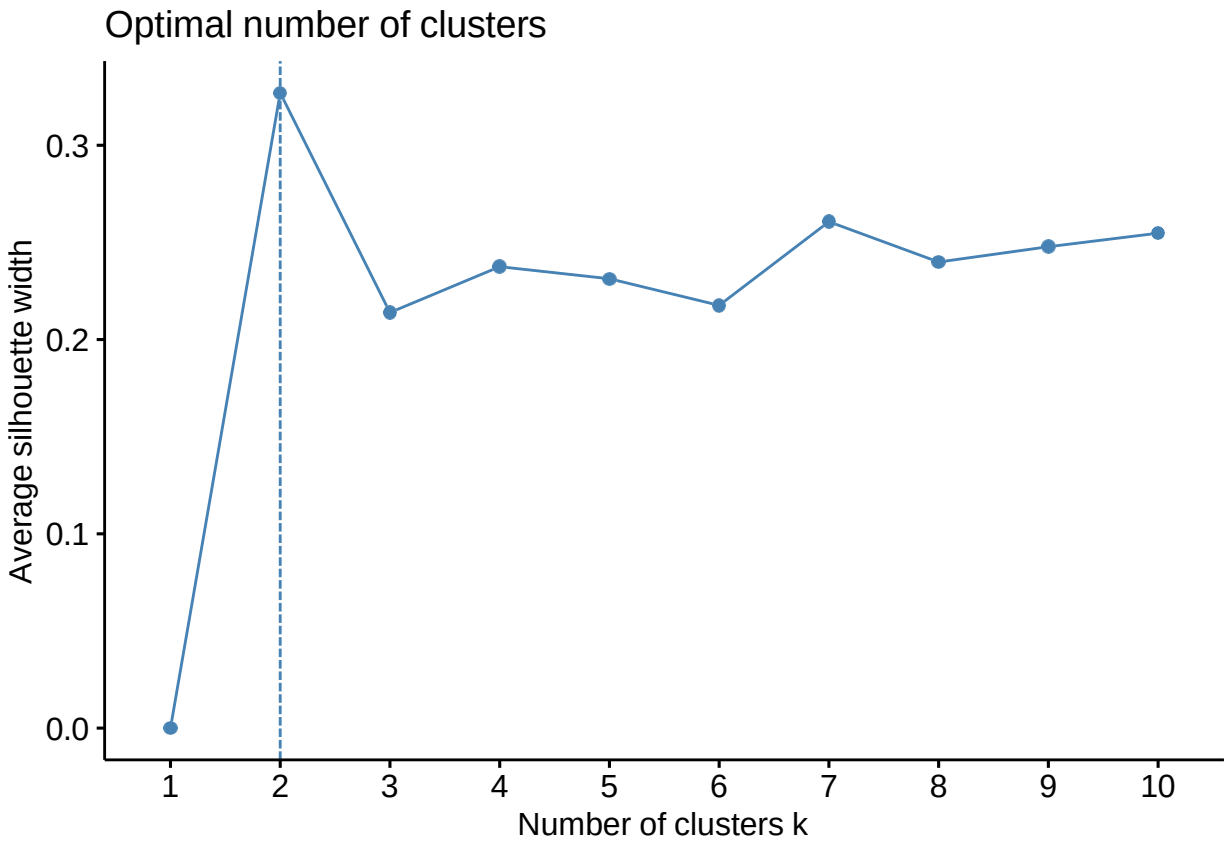
e. Clustering

```
predictors = dummies
preproc <- preProcess(predictors, method=c("center", "scale"))
predictors <- predict(preproc, predictors)
```

```
fviz_nbclust(predictors, kmeans, method = "wss")
```




```
fviz_nbclust(predictors, kmeans, method = "silhouette")
```



from the above 2 graphs, knee suggest k = 5 and silhouette suggest k = 2. We will use k = 5 and fit the data.

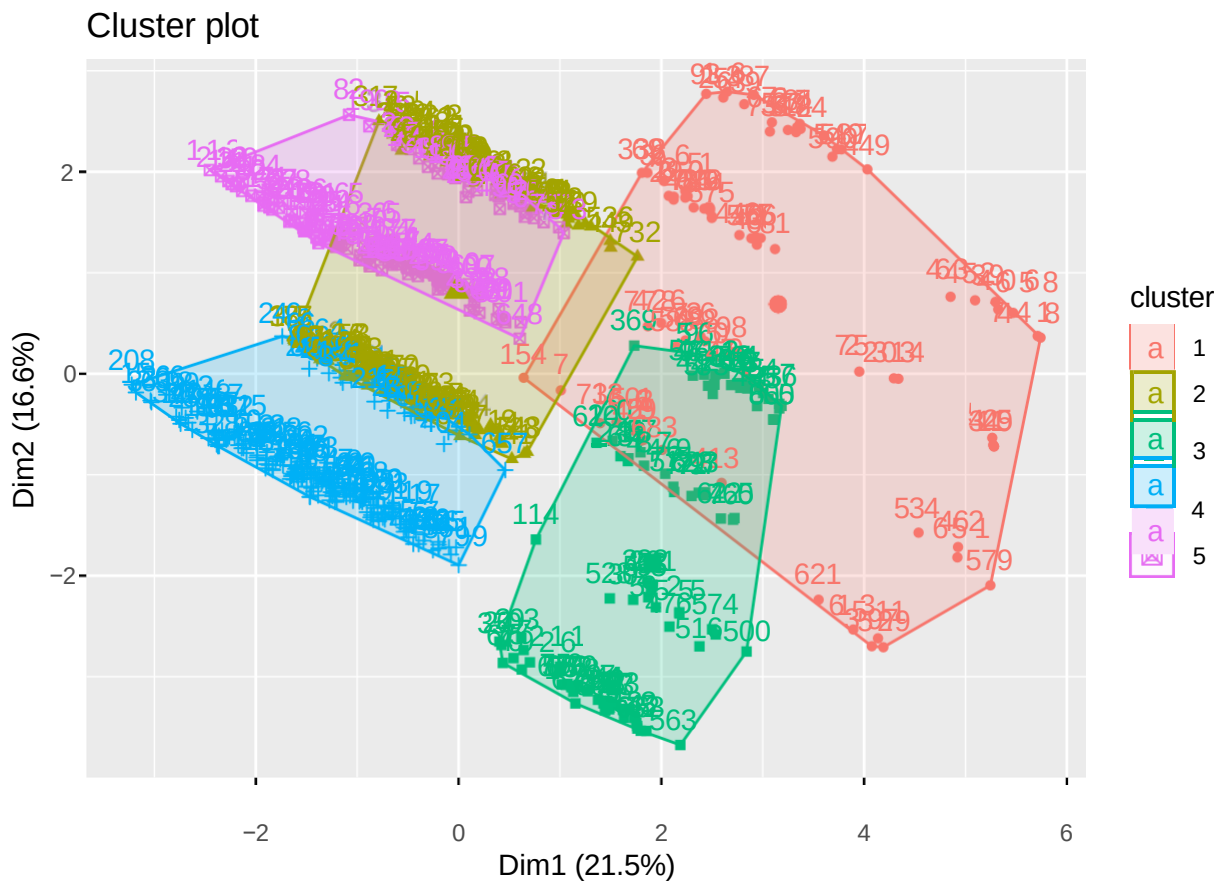
```
# Fit the data
fit <- kmeans(predictors, centers = 5, nstart = 25)
# Display the kmeans object information
fit
```

```
## K-means clustering with 5 clusters of sizes 83, 170, 104, 263, 180
##
## Cluster means:
##   genderFemale genderMale        age hypertension0 hypertension1
## 1   -0.39775606  0.39775606  0.7993877    -0.3236743     0.3236743
## 2   -0.07175041  0.07175041  0.1403524     0.4341329    -0.4341329
## 3    0.03977049 -0.03977049  0.5254338    -2.3005627     2.3005627
## 4    0.90854329 -0.90854329 -0.2350936     0.4341329    -0.4341329
## 5   -1.09928719  1.09928719 -0.4612476     0.4341329    -0.4341329
##   heart_disease0 heart_disease1 smoking_historycurrent
## 1     -2.9373042      2.9373042            -0.01735601
## 2      0.3400227     -0.3400227            -0.37988356
## 3      0.3400227     -0.3400227             0.11196873
## 4      0.3400227     -0.3400227             0.02055087
## 5      0.3400227     -0.3400227             0.27206182
##   smoking_historynon-smoker smoking_historypast_smoker        bmi HbA1c_level
## 1                -0.08044353                0.10083072  0.32288138   0.20908355
## 2                -1.20818523                1.60256491  0.01090699  -0.04338917
```

```
## 3                -0.03424007                  -0.04537176     0.45356554     0.38320599
## 4                 0.55585708                  -0.62321969 -0.15014614 -0.07769517
## 5                 0.38577142                  -0.62321969 -0.20186514 -0.16331872
##   blood_glucose_level
## 1          0.25529289
## 2          0.01996121
## 3          0.31837915
## 4         -0.08378455
## 5         -0.19810450
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   4   5   4   5   4   5   1   4   3   2   1   5   1   4   4   2   2   2   5   3
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   5   2   4   1   5   3   4   4   5   4   5   5   4   5   4   2   5   4   2   4
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   4   5   2   2   2   5   4   4   2   4   4   3   4   3   5   4   4   4   5   5
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   4   2   5   4   5   4   3   5   1   2   2   5   4   5   4   5   4   4   4   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   4   5   4   2   4   4   4   4   5   4   5   5   1   2   4   5   4   4   5   5
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   2   5   4   5   5   5   5   4   4   4   5   4   5   3   4   5   4   2   3   4
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   4   4   4   4   2   4   2   5   4   5   5   4   5   4   2   4   4   4   1   4
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   4   2   4   5   4   5   4   4   2   5   3   2   4   1   4   5   4   2   4   4
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   2   2   4   2   4   1   4   4   2   5   2   5   4   5   4   5   5   2   4   4
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
##   4   4   4   5   4   4   2   4   3   5   4   4   2   4   5   4   2   5   5   4
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220
##   2   4   1   5   5   4   4   4   5   5   3   4   3   2   4   3   4   4   2   3
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
##   5   5   4   4   4   4   5   5   4   5   4   2   4   5   2   4   2   4   5   4
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260
##   4   2   4   4   5   5   4   4   5   5   1   4   2   1   4   4   2   4   2   2
## 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280
##   3   4   5   4   1   4   5   2   2   2   4   5   4   4   4   5   2   5   4   4
## 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
##   2   5   3   4   4   2   4   5   5   1   4   5   3   2   5   5   5   4   4   4
## 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320
##   5   3   4   5   5   4   5   5   2   4   5   4   2   1   4   2   2   4   2   5
## 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340
##   4   4   2   4   5   4   2   4   2   4   4   3   4   2   5   2   5   1   5   5
## 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
##   5   2   2   5   4   1   5   4   1   3   5   2   2   2   4   5   5   5   4   4
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380
##   4   5   4   5   5   1   3   2   3   2   2   2   2   1   2   5   5   5   4   5
## 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
##   4   2   5   2   4   4   1   4   2   4   4   3   5   2   5   1   1   2   4   5
## 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
##   4   2   2   4   5   1   4   3   3   5   4   3   1   5   2   2   3   5   4   5
## 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440
```

```
##   4   4   1   5   1   1   3   4   1   4   5   2   1   1   3   3   3   3   4   2
##  441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460
##   4   2   2   4   4   1   4   1   1   4   2   1   3   2   1   2   3   4   2   2
##  461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480
##   4   1   1   2   3   3   5   5   3   5   5   4   4   2   2   3   2   5   1   4
##  481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500
##   4   4   4   5   1   4   3   1   2   4   3   2   4   4   5   5   3   3   2   3
##  501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520
##   1   4   4   1   4   1   5   2   4   2   1   3   4   2   1   3   1   1   3   2
##  521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540
##   4   4   2   2   2   2   2   3   1   5   2   4   2   1   4   2   3   4   2   5
##  541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560
##   2   1   1   2   3   2   5   2   1   2   2   3   5   2   3   5   1   5   4   4
##  561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580
##   4   2   3   4   5   5   4   2   1   4   1   3   5   3   1   3   2   3   1   2
##  581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600
##   1   3   3   5   3   2   1   1   5   1   1   3   4   4   3   3   2   4   2   5
##  601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620
##   5   2   2   4   2   1   5   3   5   3   4   3   1   4   4   3   4   1   2   5
##  621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640
##   1   3   3   5   4   2   4   3   3   4   2   2   4   2   2   3   3   4   1   4
##  641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660
##   2   4   2   3   2   4   4   5   1   4   1   3   4   2   3   2   4   1   2   2
##  661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680
##   3   2   5   4   4   3   5   4   5   5   2   1   2   2   4   4   4   3   4   4
##  681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700
##   1   3   1   5   3   5   5   3   2   2   2   4   4   2   3   2   5   1   5   5
##  701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720
##   3   5   3   2   2   4   3   5   4   4   1   5   3   5   4   5   2   5   4   2
##  721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740
##   4   2   2   3   1   4   4   5   2   3   4   2   1   3   5   1   5   1   4   4
##  741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760
##   1   4   4   4   3   4   3   3   5   1   2   4   5   4   3   4   4   2   3   3
##  761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780
##   4   5   4   2   2   5   3   3   5   3   5   3   2   3   3   4   2   1   5   2
##  781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800
##   4   5   5   4   5   4   5   5   3   2   5   5   2   2   4   2   3   4   4   3
##
## Within cluster sum of squares by cluster:
## [1]  880.1276  939.0810  846.5955 1489.3824 1149.2601
##  (between_SS / total_SS =  48.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
fviz_cluster(fit, data = predictors)
```



Fit in model with 2 clusters

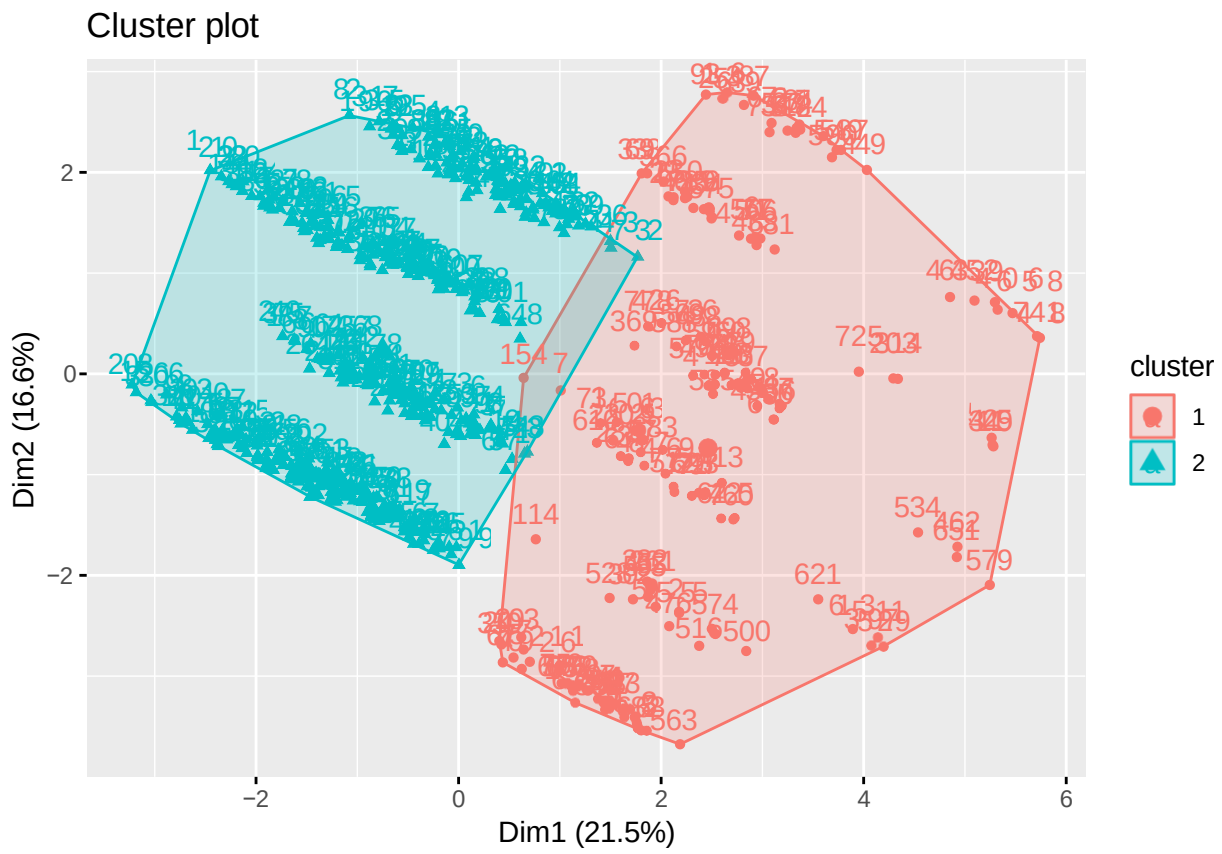
```
# Fit the data
fit1 <- kmeans(predictors, centers = 2, nstart = 25)
# Display the kmeans object information
fit1
```

```
## K-means clustering with 2 clusters of sizes 187, 613
##
## Cluster means:
##   genderFemale genderMale       age hypertension0 hypertension1
## 1   -0.15442579  0.15442579 0.6470283    -1.4231202     1.4231202
## 2    0.04710868 -0.04710868 -0.1973806     0.4341329    -0.4341329
##   heart_disease0 heart_disease1 smoking_historycurrent
## 1     -1.1146197      1.1146197             0.05456791
## 2      0.3400227     -0.3400227            -0.01664633
##   smoking_historynon-smoker smoking_historypast_smoker        bmi HbA1c_level
## 1                -0.05474749                0.019520250  0.3955613   0.30592170
## 2                 0.01670111               -0.005954791 -0.1206688  -0.09332358
##   blood_glucose_level
## 1          0.29037830
## 2         -0.08858196
##
## Clustering vector:
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  2  2  2  2  2  2  1  2  1  2  1  2  1  2  2  2  2  2  2  1
```

```
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   2   2   2   1   2   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   2   2   2   2   2   2   2   2   2   2   2   1   2   1   2   2   2   2   2   2
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   2   2   2   2   2   2   1   2   1   2   2   2   2   2   2   2   2   2   2   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   2   2   2   2   2   2   2   2   2   2   2   2   1   2   2   2   2   2   2   2
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   2   2   2   2   2   2   2   2   2   2   2   2   2   1   2   2   2   2   1   2
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   1   2
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   2   2   2   2   2   2   2   2   2   2   1   2   2   1   2   2   2   2   2   2
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   2   2   2   2   2   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
##   2   2   2   2   2   2   2   2   1   2   2   2   2   2   2   2   2   2   2   2
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220
##   2   2   1   2   2   2   2   2   2   2   1   2   1   2   2   1   2   2   2   1
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260
##   2   2   2   2   2   2   2   2   2   2   1   2   2   1   2   2   2   2   2   2
## 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280
##   1   2   2   2   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
## 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
##   2   2   1   2   2   2   2   2   2   1   2   2   1   2   2   2   2   2   2   2
## 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320
##   2   1   2   2   2   2   2   2   2   2   2   2   2   1   2   2   2   2   2   2
## 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340
##   2   2   2   2   2   2   2   2   2   2   2   1   2   2   2   2   2   1   2   2
## 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
##   2   2   2   2   2   1   2   2   1   1   2   2   2   2   2   2   2   2   2   2
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380
##   2   2   2   2   2   1   1   2   1   2   2   2   2   1   2   2   2   2   2   2
## 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
##   2   2   2   2   2   2   1   2   2   2   2   1   2   2   2   1   1   2   2   2
## 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
##   2   2   2   2   2   1   2   1   1   2   2   1   1   2   2   2   1   2   2   2
## 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440
##   2   2   1   2   1   1   1   2   1   2   2   2   1   1   1   1   1   1   2   2
## 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460
##   2   2   2   2   2   1   2   1   1   2   2   1   1   2   1   2   1   2   2   2
## 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480
##   2   1   1   2   1   1   2   2   1   2   2   2   2   2   2   1   2   2   1   2
## 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500
##   2   2   2   2   1   2   1   1   2   2   1   2   2   2   2   2   1   1   2   1
## 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520
##   1   2   2   1   2   1   2   2   2   2   1   1   2   2   1   1   1   1   1   2
## 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540
##   2   2   2   2   2   2   2   1   1   2   2   2   2   1   2   2   1   2   2   2
## 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560
##   2   1   1   2   1   2   2   2   1   2   2   1   2   2   1   2   1   2   2   2
```

```
##  561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580
##    2   2   1   2   2   2   2   2   1   2   1   1   2   1   1   1   2   1   1   2
##  581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600
##    1   1   1   2   1   2   1   1   2   1   1   1   2   2   1   1   2   2   2   2
##  601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620
##    2   2   2   2   2   1   2   1   2   1   2   1   1   2   2   1   2   1   2   2
##  621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640
##    1   1   1   2   2   2   2   1   1   2   2   2   2   2   2   1   1   2   1   2
##  641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660
##    2   2   2   1   2   2   2   2   1   2   1   1   2   2   1   2   2   1   2   2
##  661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680
##    1   2   2   2   2   1   2   2   2   2   2   1   2   2   2   2   2   1   2   2
##  681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700
##    1   1   1   2   1   2   2   1   2   2   2   2   2   2   1   2   2   1   2   2
##  701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720
##    1   2   1   2   2   2   1   2   2   2   1   2   1   2   2   2   2   2   2   2
##  721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740
##    2   2   2   1   1   2   2   2   2   1   2   2   1   1   2   1   2   1   2   2
##  741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760
##    1   2   2   2   1   2   1   1   2   1   2   2   2   2   1   2   2   2   1   1
##  761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780
##    2   2   2   2   2   2   1   1   2   1   2   1   2   1   1   2   2   1   2   2
##  781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800
##    2   2   2   2   2   2   2   2   1   2   2   2   2   2   2   2   1   2   2   1
##
## Within cluster sum of squares by cluster:
## [1] 3104.494 5490.633
##  (between_SS / total_SS =  17.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
fviz_cluster(fit1, data = predictors)
```

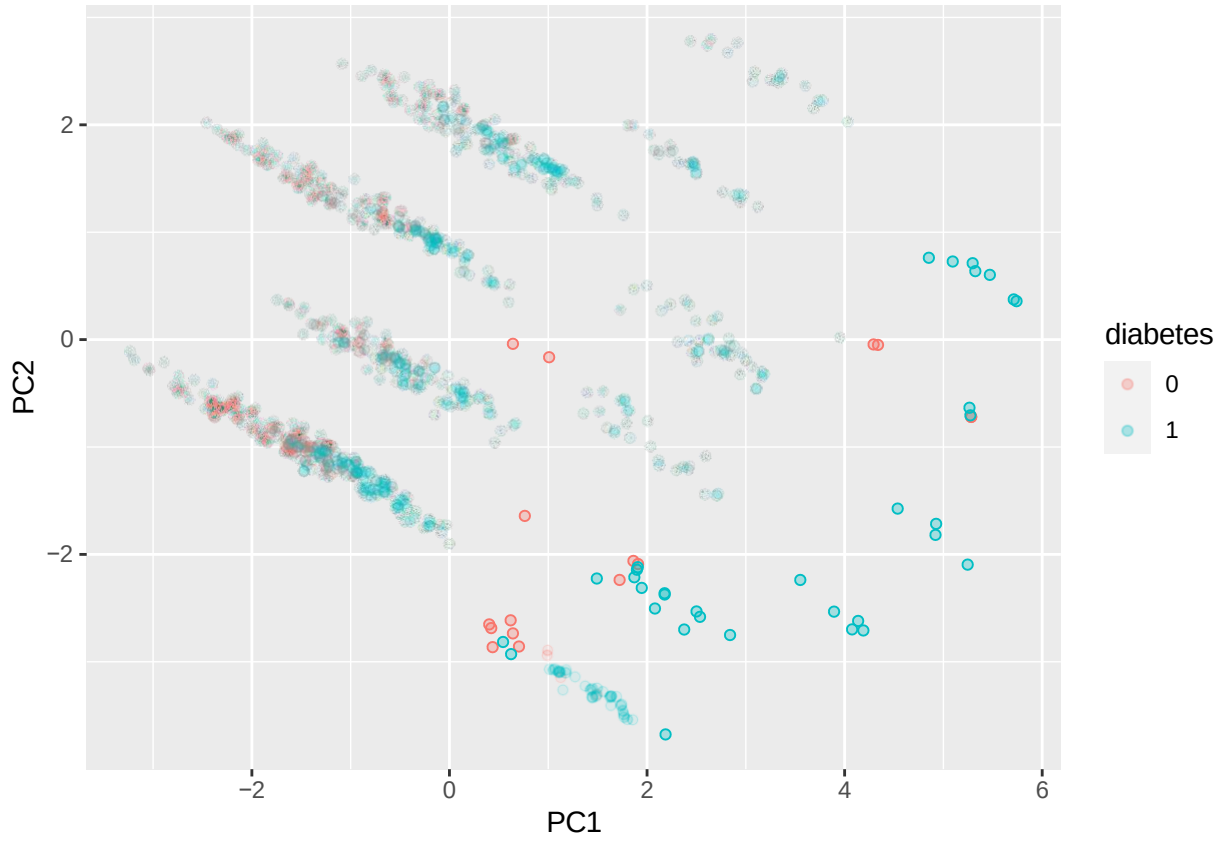


PCA projection

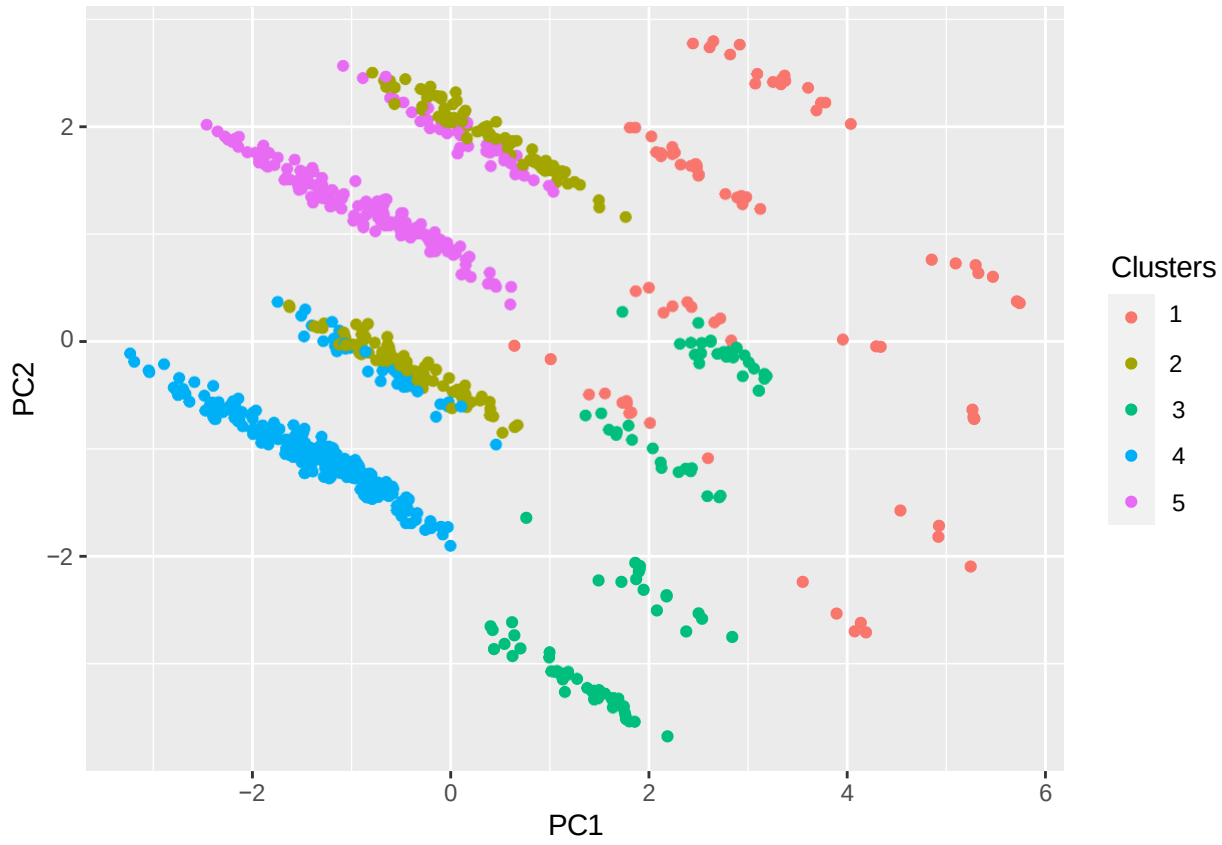
```
pca = prcomp(predictors)
rotated_data = as.data.frame(pca$x)
```

```
rotated_data$diabetes = dff$diabetes
```

```
ggplot(data = rotated_data, aes(x = PC1, y = PC2, col = diabetes)) + geom_point(alpha = 0.3)
```

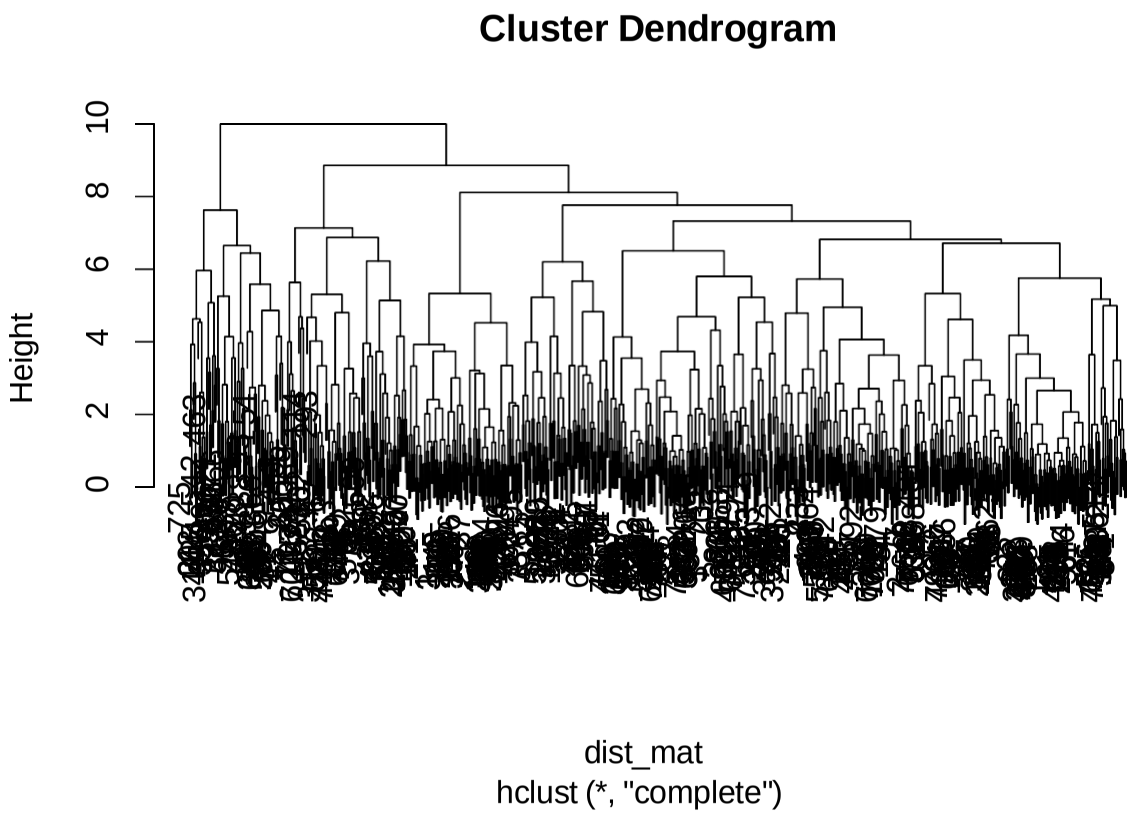



```
rotated_data$Clusters = as.factor(fit$cluster)
ggplot(data = rotated_data, aes(x = PC1, y = PC2, col = Clusters)) + geom_point()
```



Using HAC to cluster the data

```
#Euclidean and complete linkage:
dist_mat <- dist(predictors, method = 'euclidean')
# Determine assembly/agglomeration method and run hclust
hfit1 <- hclust(dist_mat, method = 'complete')
plot(hfit1)
```




```
h1 <- cutree(hfit1, k=5)
```

Comparison of clusters from Kmeans and HAC

```
result <- data.frame(Type = dff$diabetes, HAC1 = h1, Kmeans = fit$cluster)
result %>% group_by(HAC1) %>% select(HAC1, Type) %>% table()
```

```
##        Type
## HAC1    0   1
##    1  218 224
##    2   37  37
##    3   21  62
##    4   22  82
##    5   93   4
```

```
result %>% group_by(Kmeans) %>% select(Kmeans, Type) %>% table()
```

```
##           Type
## Kmeans     0   1
##         1  21  62
##         2  77  93
##         3  22  82
##         4 161 102
##         5 110  70
```

f. Classification

SVM classifier.

```
svm_data = dummies
svm_data$diabetes = dff$diabetes
```

```
# To get the same "random" results every run we need to set the randomizer seed
set.seed(123)
# Partition the data
index = createDataPartition(y=svm_data$diabetes, p=0.7, list=FALSE)
# Everything in the generated index list
train_set = svm_data[index,]
# Everything except the generated indices
test_set = svm_data[-index,]
```

```
svm_split <- train(diabetes ~., data = train_set, method = "svmLinear")
svm_split
```

```
## Support Vector Machines with Linear Kernel
##
## 561 samples
##  13 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 561, 561, 561, 561, 561, 561, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8222008  0.6436914
##
## Tuning parameter 'C' was held constant at a value of 1
```

```
pred_split <- predict(svm_split, test_set)
```

```
confusionMatrix(test_set$diabetes,pred_split)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0   1
##          0  96  21
##          1  21 101
##
##                Accuracy : 0.8243
##                  95% CI : (0.77, 0.8703)
```

```
##    No Information Rate : 0.5105 ##
P-Value [Acc > NIR] : <2e-16  ##
##                       Kappa : 0.6484
##
##   Mcnemar's Test P-Value : 1
##
##                  Sensitivity : 0.8205
##                  Specificity : 0.8279
##               Pos Pred Value : 0.8205
##               Neg Pred Value : 0.8279
##                   Prevalence : 0.4895
 ##               Detection Rate :  0.4017
  ##    Detection Prevalence : 0.4895 ##
         Balanced Accuracy : 0.8242
##
##           'Positive' Class : 0
##
```

Accuracy is 82% with C=1.

Grid search for SVM

```
train_control= trainControl(method = "cv", number = 10)
grid <- expand.grid(C = 10^seq(-5,2,0.5))
svm_grid <- train(diabetes ~., data = svm_data, method = "svmLinear",
                  trControl = train_control, tuneGrid = grid)
# View grid search result
svm_grid
```

```
## Support Vector Machines with Linear Kernel
##
## 800 samples
##  13 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 719, 720, 720, 720, 720, 720, ...
## Resampling results across tuning parameters:
##
##    C                 Accuracy    Kappa
##    1.000000e-05      0.5112502   0.0000000
##    3.162278e-05      0.5112502   0.0000000
##    1.000000e-04      0.5112502   0.0000000
##    3.162278e-04      0.6049709   0.1954467
##    1.000000e-03      0.8261816   0.6513214
##    3.162278e-03      0.8262129   0.6520278
##    1.000000e-02      0.8374937   0.6747818
##    3.162278e-02      0.8324783   0.6646537
##    1.000000e-01      0.8324937   0.6647469
##    3.162278e-01      0.8324937   0.6648254
##    1.000000e+00      0.8362437   0.6723223
##    3.162278e+00      0.8374937   0.6747908
```

```
##     1.000000e+01    0.8374937    0.6747908
##     3.162278e+01    0.8362437    0.6723097
##     1.000000e+02    0.8362437    0.6723097
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 0.01.
```

using grid search, we got greater accuracy then previous 0.8374937.

```
pred_split1 <- predict(svm_grid, svm_data)
confusionMatrix(as.factor(svm_data$diabetes),pred_split1)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction       0     1
##               0  321    70
##               1    60 349
##
##                    Accuracy : 0.8375
##                      95% CI : (0.8101, 0.8624)
##    No Information Rate : 0.5238 ##
P-Value [Acc > NIR] : <2e-16  ##
##                       Kappa : 0.6747
##
##   Mcnemar's Test P-Value : 0.4299
##
##                 Sensitivity : 0.8425
##                 Specificity : 0.8329
##              Pos Pred Value : 0.8210
##              Neg Pred Value : 0.8533
##                  Prevalence : 0.4763
 ##             Detection Rate :  0.4012
  ##   Detection Prevalence : 0.4888 ##
        Balanced Accuracy : 0.8377
##
##          'Positive' Class : 0
##
```

Using grid search, prediction accuracy got improved, setting the tuning paramter to 0.01

Decision Tree Classifier.

```
set.seed(94)

# Fit the model
tree1 <- train(diabetes ~., data = dff, method = "rpart1SE", trControl = train_control)
tree1
```

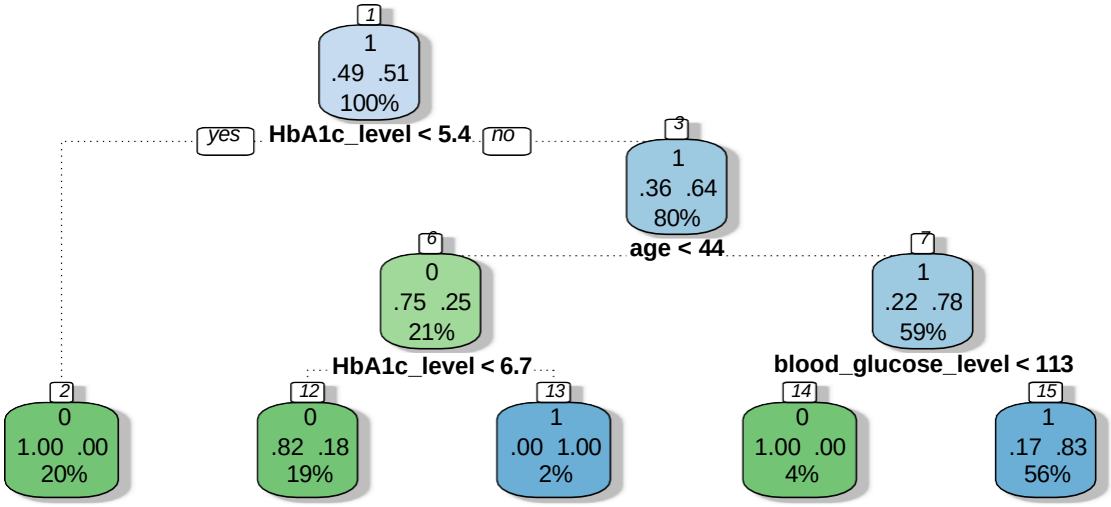
```
## CART
```

```
##
## 800 samples
##   8 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 720, 720, 720, 720, 720, 719, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8525586  0.7043254
```

```
pred_tree <- predict(tree1, dff)
confusionMatrix(dff$diabetes, pred_tree)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0   1
##          0 314  77
##          1  27 382
##
##                Accuracy : 0.87
##                  95% CI : (0.8447, 0.8925)
##     No Information Rate : 0.5738
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7391
##
##  Mcnemar's Test P-Value : 1.549e-06
##
##             Sensitivity : 0.9208
##             Specificity : 0.8322
##          Pos Pred Value : 0.8031
##          Neg Pred Value : 0.9340
##              Prevalence : 0.4263
##          Detection Rate : 0.3925
##    Detection Prevalence : 0.4888 ##
##       Balanced Accuracy : 0.8765
##
##        'Positive' Class : 0
##
```

```
fancyRpartPlot(tree1$finalModel, caption = "")
```



Tuning hyperparameters and checking for increasing accuracy on test and train split

```
hypers = rpart.control(minsplit = 5000, maxdepth = 4, minbucket = 2500)
index = createDataPartition(y=dff$diabetes, p=0.7, list=FALSE)
train_set1 = dff[index,]
test_set1 = dff[-index,]
```

```
tree2 <- train(diabetes ~., data = train_set1, control = hypers, method = "rpart1SE", trControl = train
tree2
```

```
## CART
##
## 561 samples
##   8 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 504, 505, 505, 505, 505, 505, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.5115915  0
```

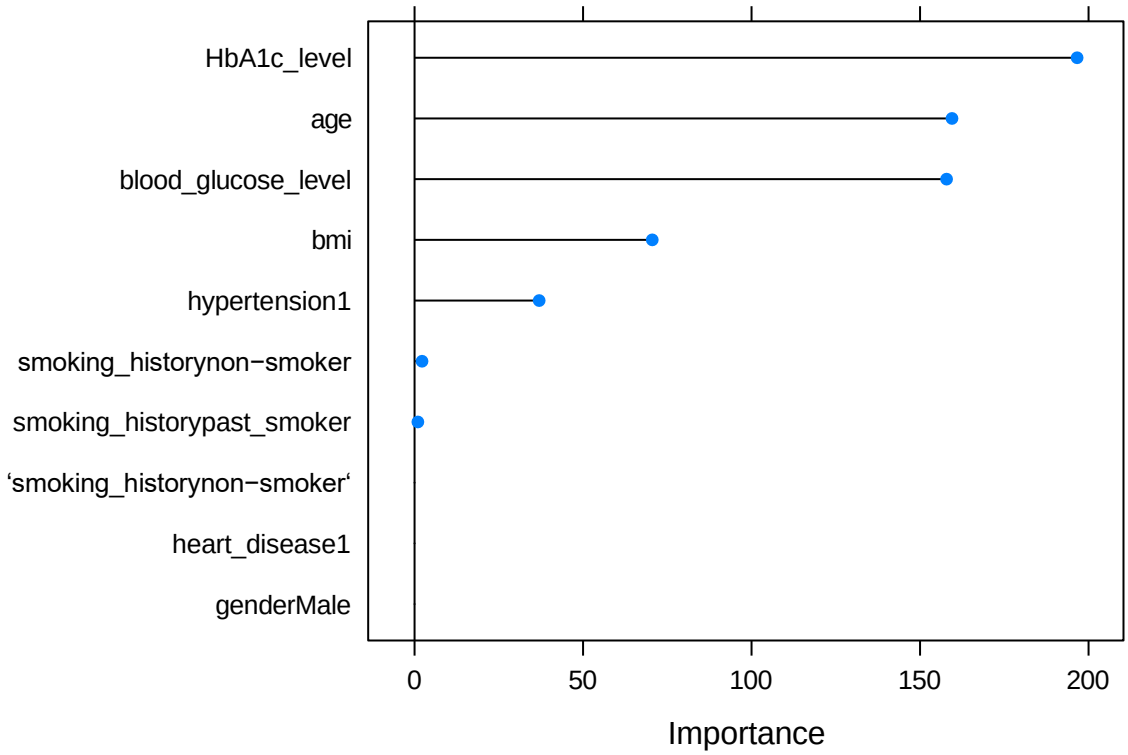
Here, we can see the train/test split and tuning the parameter does not working well and. giving less accuracy so it is not ideal to use it.

```
var_imp <- varImp(tree1, scale = FALSE)
var_imp
```

```
## rpart1SE variable importance
##
##                                  Overall
## HbA1c_level                     196.6060
## age                             159.4750
## blood_glucose_level             157.8747
## bmi                              70.5365
## hypertension1                    36.9751
## smoking_historynon-smoker         2.2170
## smoking_historypast_smoker        0.9818
## heart_disease1                    0.0000
## 'smoking_historynon-smoker'       0.0000
## genderMale                        0.0000
```

Feature Selection Relevance analysis (variable importance score)

```
plot(var_imp)
```



Taking first 4 important predictors and fitting the model.

```
new_data = select(dff,c("HbA1c_level","age","blood_glucose_level","diabetes"))
head(new_data)
```

```
##   HbA1c_level age blood_glucose_level diabetes
## 1         6.2  31                 159        0
## 2         6.5  65                 140        0
## 3         4.0  67                 126        0
## 4         5.0  65                 200        0
## 5         6.6  20                 159        0
## 6         6.2  18                 159        0
```

```
index = createDataPartition(y=new_data$diabetes, p=0.7, list=FALSE)
train_set2 = new_data[index,]
test_set2 = new_data[-index,]
```

```
tree_new <- train(diabetes ~., data = train_set2, method = "rpart1SE", trControl = train_control)
tree_new
```

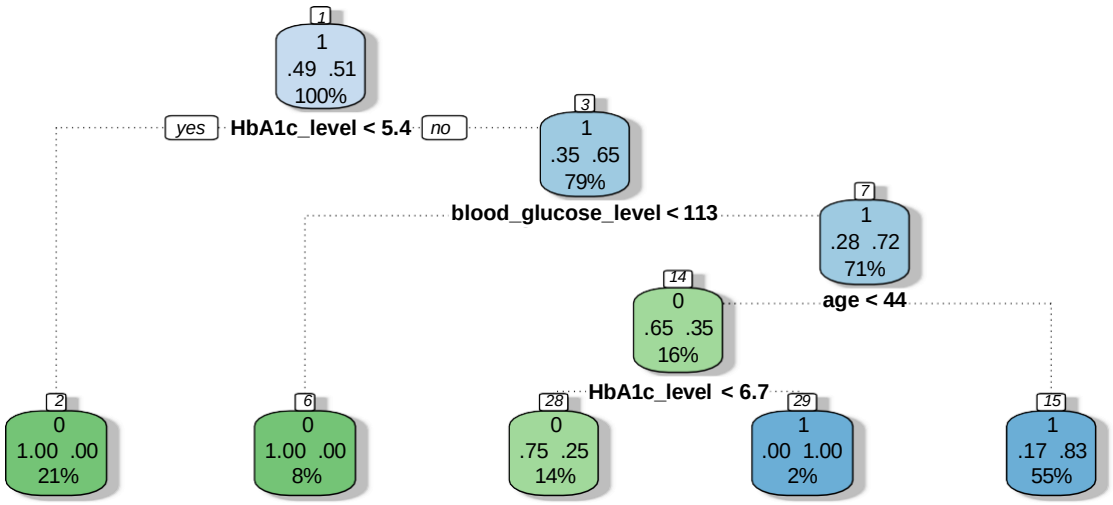
```
## CART
##
## 561 samples
##   3 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 506, 505, 505, 504, 506, 504, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8627649  0.7245085
```

```
pred_tree_new <- predict(tree_new, test_set2)
confusionMatrix(test_set2$diabetes,  pred_tree_new)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0   1
##          0  93  24
##          1   8 114
##
##                Accuracy : 0.8661
##                  95% CI : (0.8163, 0.9066)
##     No Information Rate : 0.5774
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.7313
##
##  Mcnemar's Test P-Value : 0.00801
##
```

```
##                Sensitivity : 0.9208
##                Specificity : 0.8261
##             Pos Pred Value : 0.7949
##             Neg Pred Value : 0.9344
##                 Prevalence : 0.4226
##             Detection Rate : 0.3891
##       Detection Prevalence : 0.4895
##          Balanced Accuracy : 0.8734
##
##          'Positive' Class : 0
##
```

```
fancyRpartPlot(tree_new$finalModel, caption = "")
```



after feature Reducing features boosts the accuracy to 0.87 following feature relevance analysis, however the complexity of the graph remains the same in this instance. When comparing classifier accuracy, the Decision Tree classifier outperforms the SVM classifier.

g. Evaluation

From the above 2 classifier, Evaluating Decision tree classifier.

```
cm=confusionMatrix(test_set2$diabetes,      pred_tree_new)
cm
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      0     1
##              0  93    24
##              1   8 114
##
##                    Accuracy : 0.8661
##                      95% CI : (0.8163, 0.9066)
##     No Information Rate : 0.5774
##     P-Value [Acc > NIR] : < 2e-16
##
##                       Kappa : 0.7313
##
##   Mcnemar's Test P-Value : 0.00801
##
##                 Sensitivity : 0.9208
##                 Specificity : 0.8261
##              Pos Pred Value : 0.7949
##              Neg Pred Value : 0.9344
##                  Prevalence : 0.4226
##              Detection Rate :  0.3891
##    Detection Prevalence : 0.4895 ##
          Balanced Accuracy : 0.8734
##
##           'Positive' Class : 0
##
```

```
m = cm$byClass
metrics <- as.data.frame(m)
metrics
```

```
##                                    m
## Sensitivity              0.9207921
## Specificity              0.8260870
## Pos Pred Value           0.7948718
## Neg Pred Value           0.9344262
## Precision                0.7948718
## Recall                   0.9207921
## F1                       0.8532110
## Prevalence               0.4225941
## Detection Rate           0.3891213
## Detection Prevalence     0.4895397
## Balanced Accuracy        0.8734395
```

Calculating Recall Manually and checking it

```
TP = cm$table[1,1]
FN = cm$table[2,1]
```

```
recall = TP/(TP+FN)
recall
```

```
## [1] 0.9207921
```

```
metrics["Recall",]
```

```
## [1] 0.9207921
```

Yes it is correct.

Calculating Precision Manually.

```
FP = cm$table[1,2]
precision = TP/(TP+FP)
precision
```

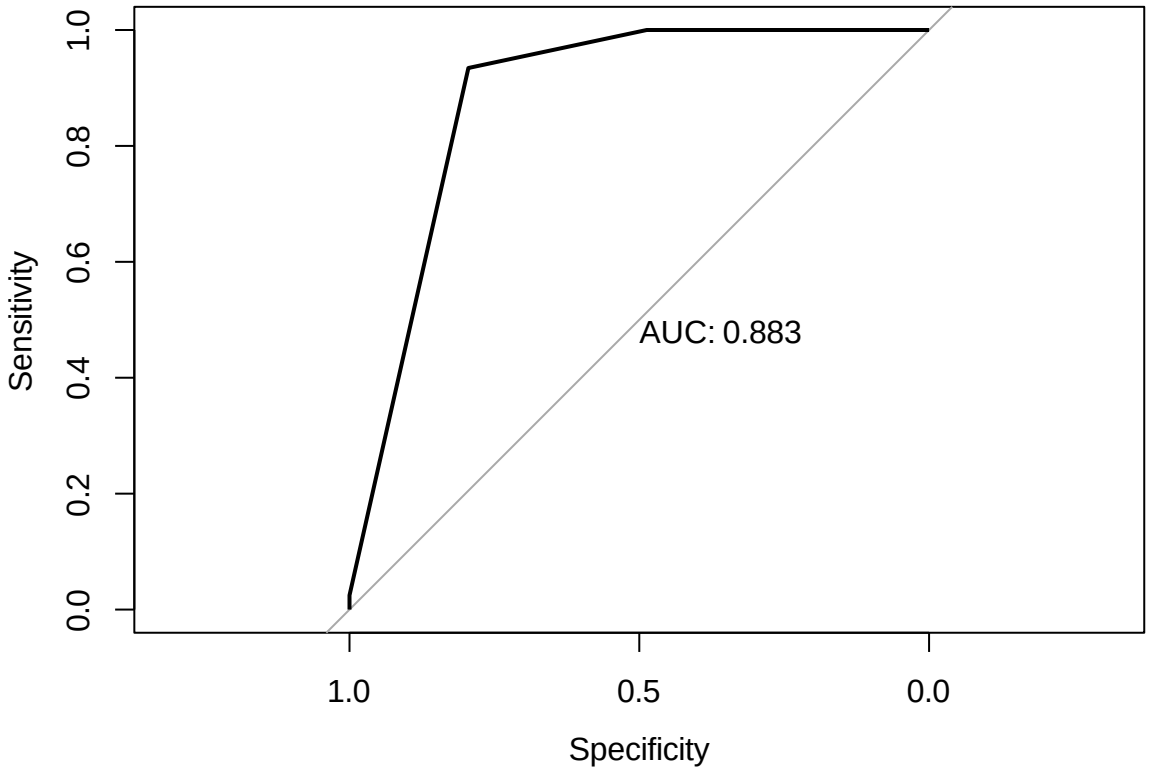
```
## [1] 0.7948718
```

```
metrics["Precision",]
```

```
## [1] 0.7948718
```

```
library(pROC)
pred_prob <- predict(tree_new, test_set2, type = "prob")
roc_obj <- roc((test_set2$diabetes), pred_prob[,1])
```

```
plot(roc_obj, print.auc=TRUE)
```



In a ROC curve, an AUC (Area Under the Curve) of 0.88 suggests that the classifier performs relatively well in differentiating between positive and negative cases. It implies that in the majority of circumstances, the classifier will score a randomly chosen positive instance higher than a randomly picked negative instance.

h. Report

About Data set:
The Diabetes Prediction Dataset was utilized for this study, and it contains a comprehensive collection of medical and demographic data from individuals, as well as their diabetes status (positive or negative). Age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level are all included in the dataset.

Description of columns:
gender: Gender refers to the classification of individuals as male or female. age: Age represents the number of years a person has lived since birth. hypertension: Hypertension, often referred to as high blood pressure, is a medical condition characterized by persistently elevated blood pressure in the arteries. heart_disease: Heart disease refers to a range of conditions affecting the heart, such as coronary artery disease, heart failure, or heart rhythm disorders. smoking_history: Smoking history indicates whether an individual has a past or present habit of smoking tobacco products. bmi: Body Mass Index (BMI) is a measure of body fat based on an individual's weight and height. HbA1c_level: HbA1c (Hemoglobin A1c) level is a laboratory test that measures the average blood sugar (glucose) levels over the past 2-3 months. blood_glucose_level: Blood glucose level refers to the concentration of glucose (sugar) in the bloodstream. diabetes: Diabetes is a

chronic medical condition characterized by elevated blood sugar levels due to insufficient insulin production or ineffective use of insulin in the body.

Data Cleaning:
Data cleaning for a dataset often consists of many stages to guarantee that the data is correct, consistent, and suitable for analysis. There were no null values or missing values in the dataset, however certain datatypes were incorrect and were updated. Aside from that, 3888 duplicate rows were discovered and cleansed. The smoking_history column had five categories, which were reduced to three for simplicity.

Data Exploration (EDA - Exploratary Data Analysis):
Numerical - age, bmi, HbA1c_level, blood_glucose_level
Categorical - hypertension, heart_disease,diabetes, gender.
From the summary of the data set, it was observed that there might be some outliers present in bmi,HbA1c_level and blood_glucose_level, using boxplot and IQR for each, outliers were detected and were removed from the dataset.

Univariate Analysis:
Age - there are no missing values, no outliers, and the data is slightly slanted to the left. bmi - Data is skewed, 25% of persons have a BMI of exactly 27.32, and around 6000 data points are outliers (6% of total data). It was also discovered that children aged 10 had a BMI of 27.32, which is irrelevant. HbA1c_level - Although there were few outliers, most persons fell in the 5 - 6.6 range, which is considered healthy. blood_glucose_level - Positively skewed, has outliers, 50% of persons fall between 100 and 160. Diabetes - there is a class imbalance.

Bivariate Analysis:
Age vs diabetes - Diabetes tends to affect older people generally.It curve of diabetes tends to slowly rise when you hit 30s and the probability is maximum when you are aged around 60.This tend to fit in well with the real world data.
blood_glucose_level vs diabetes - With increase in blood glucose level, the chance of diabetes increases the people with diabetes have a blood glucose level of around 160 on average.
HbA1c_level - with increase in HbA1c level, the chance of diabetes increases.People who have diabetes have a median HbA1c value of around 6.7.
Coorelation Matrix - The scatter matrix view gives the relationship between each variables in the data set. This view is useful because it allows us to identify the correlation which would be difficult to see looking at the distribution. We can observe, there is a positive correlation between the variables.

Data Preparation and Predictive Analysis:
Because the data set contained categorical columns, it was preferable to turn them into dummy variables for cluster analysis and further SVM classifier application. There were two classifiers utilized. 1. SVM Classifier - SVMlinear with data separated into train and test, Accuracy - 82% SVMlinear with grid search and 10-fold CV, 83% (improved) accuracy 2.Decision Tree - rpart1SE with 10 fold CV and 87% accuracy The feature significance analysis identified HbA1c_level, blood_glucose_level, and BMI as the most important variables in predicting Diabetes, with a modest improvement in Accuracy.
In this scenario, Decision Tree outperforms SVM as a classifier.

Evaluation:
These are the metrics we get: Sensitivity 0.9117647
Specificity 0.8248175
Pos Pred Value 0.7948718
Neg Pred Value 0.9262295
Precision 0.7948718

Recall 0.9117647
F1 0.8493151
Prevalence 0.4267782
Detection Rate 0.3891213
Detection Prevalence 0.4895397

ROC curve : An AUC (Area Under the Curve) of 0.88 in a ROC curve indicates that the classifier has reasonably good performance in distinguishing between positive and negative instances. It suggests that there is a high probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance in the majority of cases.

Key takeaways :
There were several aspects in this data collection and during the analysis that I found fascinating.First, I learned the right use case for each and every component from all of the lectures and tutorials in the to the real world problems. Diabetes data collection contained a lot of extraneous stuff that took a long time to clean and make sure it was suitable, as it took 70-80% of the time in analysis and cleaning. Furthermore, other classifiers can be utilized, and in this case, SVM and Decision performed well for predicting, and the metrics provided helpful information about the prediction. Tuning the parameter may not always yield better results; we must experiment extensively in order to achieve optimal accuracy. Finally, there is evaluating and ROC curve results the overall performance of model.

i. Reflection

Thanks to Prof. Roselyne Tchoua, I have gained extensive knowledge of the principles of Data Science and Machine Learning.including data cleansing, data exploration, evaluation, and model construction.The new thing I learnt from the tutorials is the R programming language, which was completely new to me because I was already familiar with the principles of Python, pandas, and numpy. Some of the specific skills I learned: Data Manipulation and Cleaning: learned techniques for cleaning and preprocessing data, handling missing values, dealing with outliers, and transforming data to make it suitable for analysis.Exploratory Data Analysis: Analyzing data, perform descriptive statistics, visualize data using graphs and charts, patterns and insights from data. Machine Learning Algorithms: machine learning algorithms, such as regression, classification, clustering, and dimensionality reduction. Learned how to select appropriate algorithms, train models, and evaluate their performance. Model Evaluation and Validation: learned techniques to evaluate and validate machine learning models, including cross-validation, model selection, and performance metrics such as accuracy, precision, recall, and F1-score. And some ethical consideration such as privacy, bias, and fairness.
thank you for your teaching and mentorship. Your course has had a profound impact on me, and I am grateful for the opportunity to learn from you.

ThankYou,

BariniSimhadri