

# Palestine Technical University (Kadoorie) Faculty of Engineering and Technology Computer Systems Engineering

**DATA MINING PROHECT**

**THYROID DISEASE DATA SET**

**By:**

|  |  |
| --- | --- |
| **Raja’ Mohammad Asad** | **201911051** |
| **Hala Mohamad Shrem** | **201911474** |
| **Saja Moqadi** | **201910892** |

**Supervisor:**

**Dr. Anas Melhem**

**Project submitted in Data Mining Course**

**Tulkarm, Palestine**

**2022**

I

# TABLE OF CONTACT

[TABLE OF CONTACT I](#_bookmark1)

[CHAPTER 1: INTRODUCTION 3](#_bookmark3)

[CHAPTER 2: DATA SET 4](#_bookmark7)

[CHAPTER 3: PROBLEM DEFINITION 9](#_bookmark8)

[CHAPTER 4: DATA PREPARATION 10](#_bookmark20)

[CHAPTER 5: DATA MINING MODEL 35](#_bookmark25)

[CHAPTER 6: DISCUSSION 36](#_bookmark26)

REFERENCES……………………………………………………………………………………… …..………………….37

# CHAPTER 1: INTRODUCTION

## Problem Statement

The Thyroid Disease Data Set is a dataset that contains information about patients with thyroid disease. It includes several columns of data, such as age, sex, thyroid hormone levels, and thyroid disease diagnosis. The goal of working with this dataset might be to use machine learning techniques to build a model that can predict thyroid disease diagnosis based on the other columns of data.

The tool we will use is R is a programming language and software environment for statistical computing and graphics. It is widely used by statisticians, data scientists, and researchers for data analysis, visualization, and machine learning.

One of the key features of R is its rich ecosystem of packages and libraries that provide a wide range of functions and tools for data manipulation, visualization, statistical analysis, machine learning, and more. This makes R a powerful tool for working with data, as you can easily access and use a wide range of specialized functions and libraries to perform a variety of tasks.

R also has a strong community of users and developers, which means that there is a wealth of documentation, tutorials, and resources available online to help you learn how to use R effectively. In addition, R has a large user base, which means that it is well-supported and has a relatively low learning curve compared to some other programming language

# CHAPTER 2: DATA SET

# The Thyroid Disease Data Set is a machine learning dataset was collected by the Department of Medicine at the University of Pisa, Italy. that contains information about patients with thyroid disease. It includes several columns of data, such as age, sex, thyroid hormone levels, and thyroid disease diagnosis. The dataset is available on Kaggle, a platform for data science and machine learning.

# Link in Kaggle: https://www.kaggle.com/datasets/yasserhessein/thyroid-disease-data-set

# Link in UCI: https://archive.ics.uci.edu/ml/datasets/thyroid+disease

It contains 3772 observation and 30 attribute

1. **age**: The age of the patient in years. numeric
2. ***sex***: The sex of the patient (male or female). binary
3. **on\_thyroxine**: Whether the patient is taking thyroxine (a thyroid hormone). binary
4. **query\_on\_thyroxine**: Whether the patient has had a query about thyroxine in the past. binary
5. **on\_antithyroid\_medication**: Whether the patient is taking antithyroid medication. binary
6. **thyroid\_surgery**: Whether the patient has had thyroid surgery. binary
7. **query\_hypothyroid**: Whether the patient has had a query about hypothyroidism (underactive thyroid) in the past. binary
8. **query\_hyperthyroid:** Whether the patient has had a query about hyperthyroidism (overactive thyroid) in the past. binary
9. **pregnant**: Whether the patient is pregnant. binary
10. **sick**: Whether the patient is sick. binary
11. **tumor**: Whether the patient has a tumor. binary
12. **lithium**: Whether the patient is taking lithium (a medication used to treat bipolar disorder). binary
13. **goitre:** Whether the patient has a goitre (a swelling in the neck caused by an enlarged thyroid gland). binary
14. **TSH\_measured:** Whether the patient's thyroid-stimulating hormone (TSH) levels have been measured. binary
15. **TSH:** The patient's TSH levels. numeric
16. **T3\_measured**: Whether the patient's triiodothyronine (T3) levels have been measured. binary
17. **T3:** The patient's T3 levels. numeric
18. **TT4\_measured**: Whether the patient's thyroxine (TT4) levels have been measured. Logical
19. **TT4:** The patient's TT4 levels. numeric
20. **T4U\_measured**: Whether the patient's thyroxine (T4U) levels have been measured. Logical
21. **T4U:** The patient's T4U levels. numeric
22. **FTI\_measured**: Whether the patient's free thyroxine index (FTI) levels have been measured. binary
23. **FTI:** The patient's FTI levels. numeric
24. **TBG\_measured:** Whether the patient's thyroxine-binding globulin (TBG) levels have been measured. binary
25. **TBG:** The patient's TBG levels. numeric
26. **referral\_source:** The source of referral for the patient (self, SVHC, SVHD, or others). nominal
27. **Binary\_Class:** The diagnosis for the patient (negative, Positive). char factor

View our Dataset

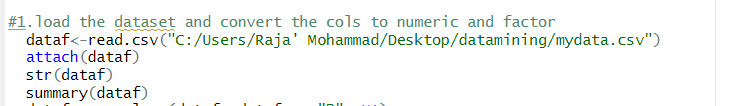
dataf<-read.csv("C:/Users/Raja' Mohammad/Desktop/datamining/mydata.csv")

attach(dataf)

str(dataf)

View(dataf)

summary(dataf)



# 

# 

# All attributes are “char” because they have “?” we will convert all “?” to NA then convert the data type for each attribute

# There are several potential problems that you might encounter when working with the Thyroid Disease Data Set:

# Missing values: Some columns in the dataset may contain missing values, which can be a problem for analysis or machine learning. You will need to decide how to handle missing values, such as by imputing values or dropping rows with missing data.

# 

# Class imbalance: The class labels in the dataset may be imbalanced, with one class significantly more prevalent than the other. This can be a problem when building machine learning models, as the model may be biased towards the majority class. You will need to consider techniques such as oversampling or under sampling to balance the dataset.

# 

# ggplot(dataf, aes(x=reorder(binaryClass, binaryClass, function(x)-length(x)))) +

# geom\_bar(fill='purple') + labs(x='Class')

# 

# Outliers: The dataset may contain outliers, which are values that are significantly different from the majority of the data. Outliers can have a large impact on statistical analysis or machine learning, so you will need to decide whether to keep or remove them.

# 

# 

# CHAPTER 3: PROBLEM DEFINITION

# 

# Business problem:

# we are data scientists working for a healthcare company that is interested in improving the accuracy of thyroid disease diagnosis. The company has collected data on patients with thyroid disease, including information about their age, sex, thyroid hormone levels, and diagnosis, and has asked you to use this data to build a model that can accurately predict thyroid disease diagnosis.

# Data mining problem:

# Given a dataset containing patient data on thyroid disease, including information about age, sex, thyroid hormone levels, and diagnosis, build a model that can accurately predict thyroid disease diagnosis based on the other variables in the dataset. Use data mining techniques, such as feature selection and model evaluation, to optimize the performance of the model. The model should be able to accurately predict thyroid disease diagnosis for new patients based on their age, sex, thyroid hormone levels, and other relevant factors. This is THYROID DISEASE detection

# CHAPTER 4: DATA PREPARATION

# FIRST, we need to replace all “?” to NA

# 1. dataf <- replace(dataf, dataf == "?", NA)

# 

# Then convert to suitable datatype for each attribute

age<-as.numeric(age)

TT4<-as.numeric(TT4)

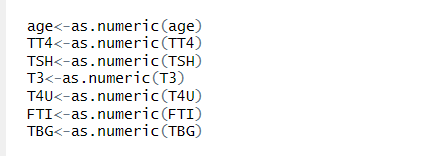
TSH<-as.numeric(TSH)

T3<-as.numeric(T3)

T4U<-as.numeric(T4U)

FTI<-as.numeric(FTI)

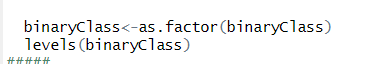
TBG<-as.numeric(TBG)



And convert the binary class to factor

binaryClass<-as.factor(binaryClass)

levels(binaryClass)



fact.png

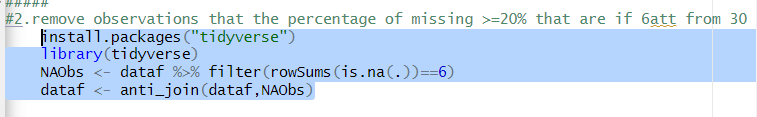
2. remove observations that the percentage of missing >=20% that are if 6 attributes from 30 is null

install.packages("tidyverse")

library(tidyverse)

NAObs <- dataf %>% filter(rowSums(is.na(.))==6)

dataf <- anti\_join(dataf,NAObs)

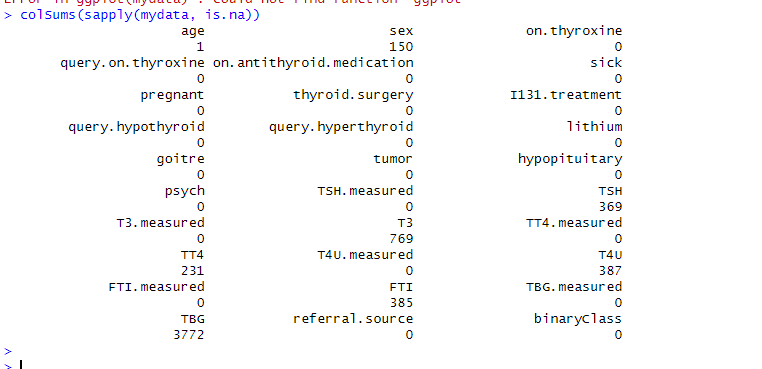


3.check the missing values in each attribute

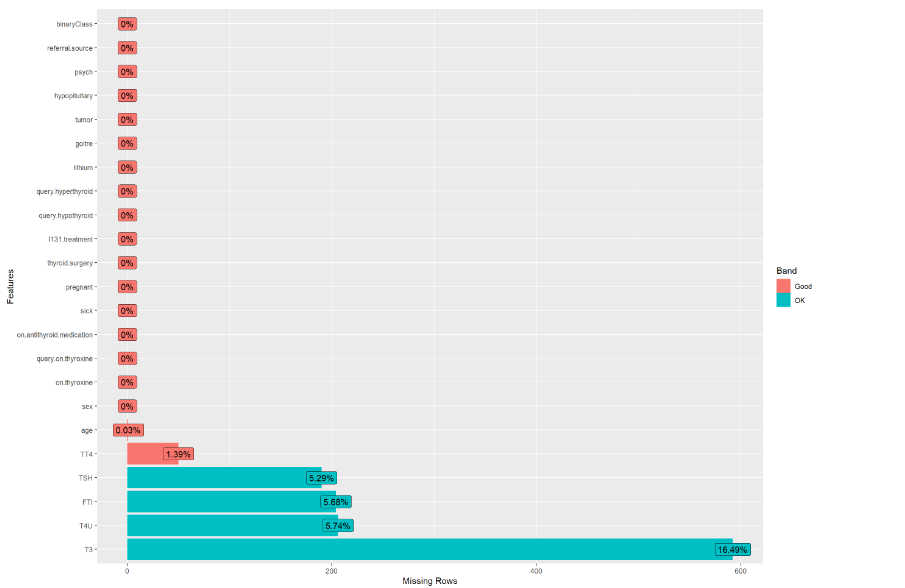
sum(is.na(dataf))

count miss.png

colSums(sapply(dataf, is.na))



plot\_missing(dataf )



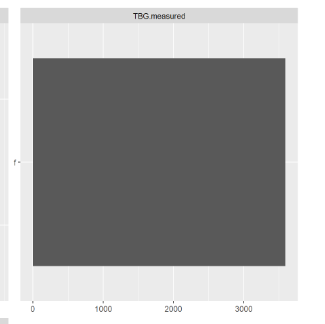
Then we will handle each attribute individual

1.TBG col: all values are NA we will remove it

dataf<-subset(dataf, select = -TBG)



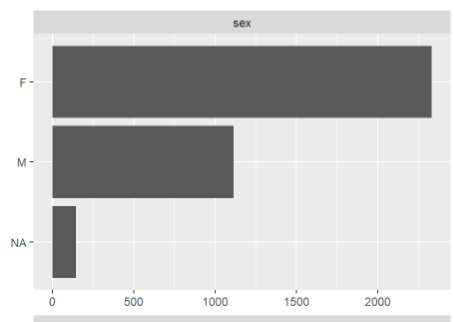
2. TBG.measured: is biased attribute all is "F" no one make TBG test then remove it



dataf<-subset(dataf, select = -TBG.measured)



1. Sex attribute
2. Percentage of missing value is 4% then we will fill the missing with the mode is “F”



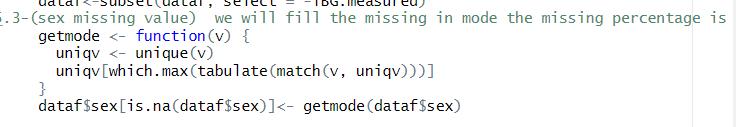
getmode <- function(v) {

uniqv <- unique(v)

uniqv[which.max(tabulate(match(v, uniqv)))]

}

dataf$sex[is.na(dataf$sex)]<- getmode(dataf$sex)



# the attributes (TSH measured, T3 measured, TT4 measured, T4U measured, FTI measured) all attribute are biased and the attributes mean did the patient make a test for this Harmon almost answers are True but anther answers are false then NA in the next attribute,

# for example T3.measured is F then the T3 is ? NA

# 

# And these attributes are biased to true almost patients did these Harmon tests

# 

# Then we will remove it

# dataf<-subset(dataf, select = -TSH.measured)

# dataf<-subset(dataf, select = -T3.measured)

# dataf<-subset(dataf, select = -TT4.measured)

# dataf<-subset(dataf, select = -T4U.measured)

# dataf<-subset(dataf, select = -FTI.measured)

# 

# age attribute:

# draw the normal distribution

# z<-dnorm(age,mean(age,na.rm=T),sd(age,na.rm=T))

# plot(age,z)

# 

# 

# According to this graph there is no biased

# Missing value: the percentage of missing is .03%

# mean(age,na.rm = T) #51.735

# median(age,na.rm = T)#54

# 

# #we will fill the missing in median

# age[is.na(age)]<- median(age,na.rm = T)

# 

# outliers:

# boxplot(age)

# 

# Q1 <- quantile(age, 0.25,na.rm = T)

# Q3 <- quantile(age, 0.75,na.rm = T)

# IQR <- Q3 - Q1

# upper\_limit <- Q3 + 1.5\*IQR

# lower\_limit <- Q1 - 1.5\*IQR

# trim\_age<- age[ age >upper\_limit]

# #the percentage of outliers is

# length(trim\_age)/nrow(dataf)\*100 #.02%

# 

# then replace the outliers with median or mean roughly the same

# 

# dataf$age[dataf$age > upper\_limit]<-median(dataf$age)

# count for each value

# ggplot(dataf, aes(x=reorder(age, age, function(x)-length(x)))) +

# geom\_bar(fill='red') + labs(x='age')

# 

# 

# 7.TSH:

# draw the normal distribution

# z<-dnorm(TSH,mean(TSH,na.rm=T),sd(TSH,na.rm=T))

# plot(TSH,z)

# 

# the attribute is positively skewed

# missing value 5.29%

# mean(TSH,na.rm = T) #5.07

# median(TSH,na.rm = T)#1.4

# we will fill the missing in median because it positively skewed and the mean effected by extremes

# dataf$TSH[is.na(dataf$TSH)]<- median(TSH,na.rm = T)

# c.outliers:

# draw boxplot for outliers

# boxplot(TSH)

# 

# Q1 <- quantile(TSH, 0.25,na.rm = T)

# Q3 <- quantile(TSH, 0.75,na.rm = T)

# IQR <- Q3 - Q1

# upper\_limit <- Q3 + 1.5\*IQR

# lower\_limit <- Q1 - 1.5\*IQR

# trim\_TSH<- TSH[ TSH <lower\_limit|TSH >upper\_limit]

# #the percentage of outliers is

# length(trim\_TSH)/nrow(dataf)\*100 #11.67%

# #then replace the outliers with median

# dataf$TSH[dataf$TSH < lower\_limit | dataf$TSH > upper\_limit]<-median(dataf$TSH)

# 

# 8. (T3)

# attach(dataf)

# normal distribution

# z<-dnorm(T3,mean(T3,na.rm=T),sd(T3,na.rm=T))

# plot(T3,z)

# 

# Positively skewed

# missing value 16%

# mean(T3,na.rm = T) #2.01

# median(T3,na.rm = T)#2

# we will fill the missing in median

# dataf$T3[is.na(dataf$T3)]<- median(T3,na.rm = T)

# outliers

# #draw boxplot for outliers

# boxplot(T3)

# 

# 

# Q1 <- quantile(T3, 0.25,na.rm = T)

# Q3 <- quantile(T3, 0.75,na.rm = T)

# IQR <- Q3 - Q1

# upper\_limit <- Q3 + 1.5\*IQR

# lower\_limit <- Q1 - 1.5\*IQR

# trim\_T3<- T3[ T3 <lower\_limit|T3 >upper\_limit]

# the percentage of outliers is

# length(trim\_T3)/nrow(dataf)\*100 #26.6%

# we will replace with median

# dataf$T3[dataf$T3 > upper\_limit]<-median(dataf$T3,na.rm = T)

# 

# 9. TT4

# a. normal distribution:

# attach(dataf)

# z<-dnorm(TT4,mean(TT4,na.rm=T),sd(TT4,na.rm=T))

# plot(TT4,z)

# 

# b. missing value 1.39%

# mean(TT4,na.rm = T) #108.3

# median(TT4,na.rm = T)#103

# #we will fill the missing in median because it positively skewed and the effected by extremes

# dataf$TT4[is.na(dataf$TT4)]<- median(TT4,na.rm = T)

# c. draw boxplot for outliers

# boxplot(TT4)

# 

# Q1 <- quantile(TT4, 0.25,na.rm = T)

# Q3 <- quantile(TT4, 0.75,na.rm = T)

# IQR <- Q3 - Q1

# upper\_limit <- Q3 + 1.5\*IQR

# lower\_limit <- Q1 - 1.5\*IQR

# trim\_TT4<- TT4[ TT4 <lower\_limit|TT4 >upper\_limit]

# #the percentage of outliers is

# length(trim\_TT4)/nrow(dataf) #.11%

# #we will trim the outliers

# dataf$TT4 <- subset(dataf, TT4 > lower\_limit & TT4 < upper\_limit)

# 

# 10.T4U:

# a. normal distribution

# attach(dataf)

# z<-dnorm(T4U,mean(T4U,na.rm=T),sd(T4U,na.rm=T))

# plot(T4U,z)

# 

# 

# draw normal distribution no biased

# 

# b. missing value 5.47%

# mean(T4U,na.rm = T) #.99

# median(T4U,na.rm = T)#.98

# #we will fill the missing in median

# dataf$T4U[is.na(dataf$T4U)]<- median(T4U,na.rm = T)

# c.outliers:

# Q1 <- quantile(T4U, 0.25,na.rm = T)

# Q3 <- quantile(T4U, 0.75,na.rm = T)

# IQR <- Q3 - Q1

# upper\_limit <- Q3 + 1.5\*IQR

# lower\_limit <- Q1 - 1.5\*IQR

# trim\_T4U<- T4U[ T4U <lower\_limit|T4U >upper\_limit]

# #the percentage of outliers is

# length(trim\_T4U)/nrow(dataf) #.15%

# #we will replace it with median

# dataf$T4U[dataf$T4U > upper\_limit]<-median(dataf$T4U,na.rm = T)

# 

# 

# 

# 10.FTI

# a. normal distribution

# attach(dataf)

# z<-dnorm(FTI,mean(FTI,na.rm=T),sd(FTI,na.rm=T))

# plot(FTI,z)

# 

# 

# b.missing value 5.68%

# mean(FTI,na.rm = T) #110.4

# median(FTI,na.rm = T)#107

# #we will fill the missing in mean

# dataf$FTI[is.na(dataf$FTI)]<- mean(FTI,na.rm = T)

# #draw boxplot for outliers

# boxplot(FTI)

# Q1 <- quantile(FTI, 0.25,na.rm = T)

# Q3 <- quantile(FTI, 0.75,na.rm = T)

# IQR <- Q3 - Q1

# upper\_limit <- Q3 + 1.5\*IQR

# lower\_limit <- Q1 - 1.5\*IQR

# trim\_FTI<- FTI[ FTI <lower\_limit|FTI >upper\_limit]

# #the percentage of outliers is

# length(trim\_FTI)/nrow(dataf) #.16%

# #we will replace it with median

# dataf$FTI[dataf$FTI > upper\_limit]<-median(dataf$FTI,na.rm = T)

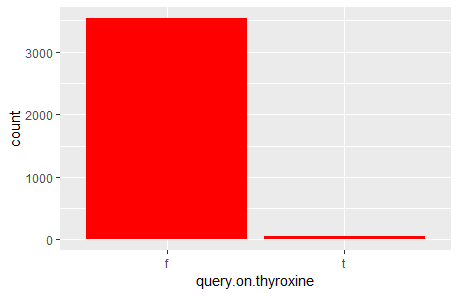
# 

# Now all binary attributes don’t have missing value 0%

11. (query.on.thyroxine)

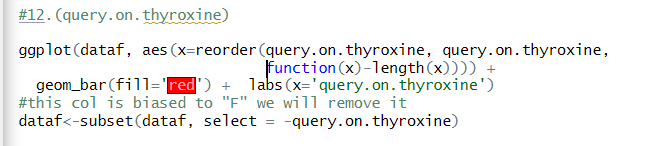
ggplot(dataf, aes(x=reorder(query.on.thyroxine, query.on.thyroxine, function(x)-length(x)))) +

geom\_bar(fill='red') + labs(x='query.on.thyroxine')



this col is biased to "F" we will remove it

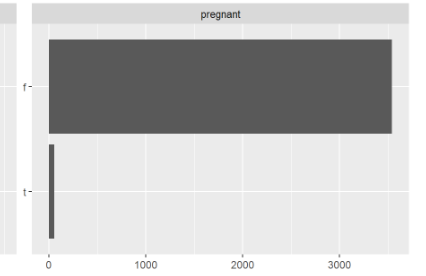
dataf<-subset(dataf, select = -query.on.thyroxine)



12. pregnant attribute: this col is biased to "F"

ggplot(dataf, aes(x=reorder(pregnant, pregnant, function(x)-length(x)))) +

geom\_bar(fill='blue') + labs(x='pregnant')

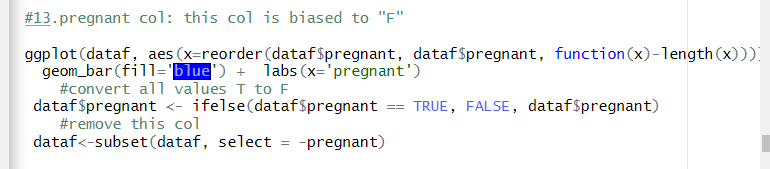


convert all values T to F

dataf$pregnant <- ifelse(dataf$pregnant == TRUE, FALSE, dataf$pregnant)

remove this col

dataf<-subset(dataf, select = -pregnant)

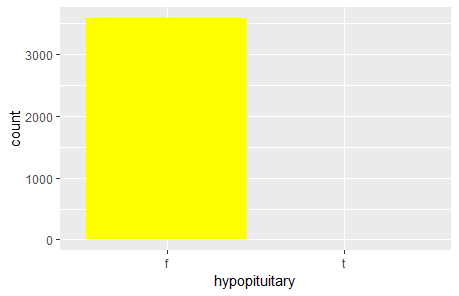


13.hypopituitary attribute

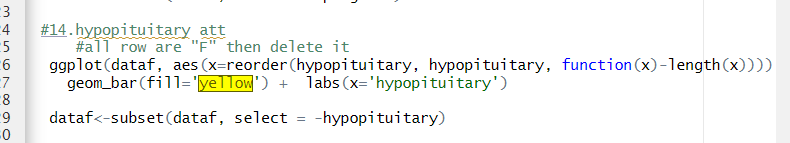
all row are "F" then delete it

ggplot(dataf, aes(x=reorder(hypopituitary, hypopituitary, function(x)-length(x)))) +

geom\_bar(fill='yellow') + labs(x='hypopituitary')



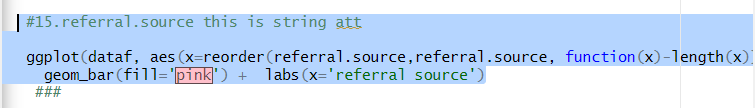
dataf<-subset(dataf, select = -hypopituitary)

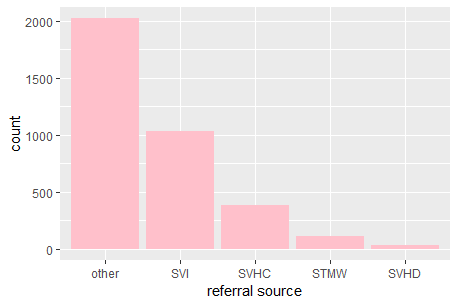


14.referral.source this is nominal attribute

ggplot(dataf, aes(x=reorder(referral.source,referral.source, function(x)-length(x)))) +

geom\_bar(fill='pink') + labs(x='referral source')



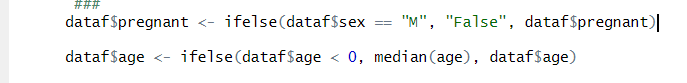


And we have inconsistency problems like sex=”M” and Pregnant=”T” it solves manually

dataf$pregnant <- ifelse(dataf$sex == "M", "False", dataf$pregnant)

if the age is negative convert it to median of age

dataf$age <- ifelse(dataf$age < 0, median(age), dataf$age)



# CHAPTER 5: DATA MINING MODEL CLASSIFICATION

# the binaryClass is the factor but it baised

# ggplot(dataf, aes(x=reorder(binaryClass, binaryClass, function(x)-length(x)))) +

# geom\_bar(fill='green') + labs(x='binaryClass')

# 

# so we must solve this problem first,

# divide the all "p" rows in dataset1 and all "N" in dataset2 and make train and test data

# data1<- subset(dataf, binaryClass == "P")

# data2<- subset(dataf, binaryClass == "N")

# nrow(data1)#3301

# nrow(data2)#290

for train data take 70% from data1 "p" and 70% from data2 "N"

for test data take 30% from data1 "p" and 30% from data2 "N"

split <- sample.split(data1$binaryClass, SplitRatio = 0.7)

train <- subset(data1, split == TRUE)

test <- subset(dataf, split == FALSE)

split <- sample.split(data2$binaryClass, SplitRatio = 0.7)

train <- rbind(subset(data2, split == TRUE),train)

test <- rbind(subset(data2, split == FALSE),test)

classification model

model <- naiveBayes(binaryClass ~ ., data = train)

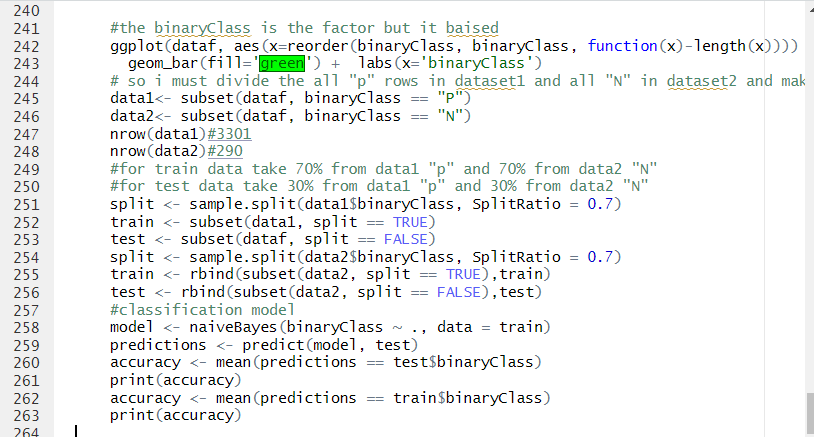
predictions <- predict(model, test)

accuracy <- mean(predictions == test$binaryClass)

print(accuracy)

accuracy <- mean(predictions == train$binaryClass)

print(accuracy)



The accuracy on Train data:



The accuracy on Test data



# CHAPTER 6: DISCUSSIONS

First the accuracy on test data is more than the accuracy on the train and this is a good sign for our working

And the accuracy is high we will accept it

Then we can diagnosis any person if it positive or negative according to some features with accuracy 93%

# REFERENCES

# We use these libraries: first install this libraries

1. > library(ggplot2)

2. > library(base)

3. > library(datasets)

4. > library(graphics)

5. > library(tools)

6. > library(stats)

7. > library(utils)

8. > library(e1071)

9. > library(tidyverse)

10. > library(DataExplorer)