Thyroid Disease Prediction and Data Analysis

Rahul Bharadwaj Machiraju(A20502085)
Sai Pavan Kunda(A20496516)
Rahul Maddula(A20488730)

Overview

- Firstly, we need to identify the major target classes and prepare the data accordingly.
- Clean the data based on the target classes and arrange the data accordingly.
- Then perform exploratory data analysis on the data to identify the key features.
- After identifying the key features, split the data into train and test. Use the training data for training the designed models.
- After training, test the trained model with the testing data and see how well it performs.
- Finally, find the Accuracy of the models and conclude which among the models perform well on the data and state reasons why.

Problem Statement

The thyroid is an endocrine gland located in the anterior region of the neck: its main task is to produce thyroid hormones, which are functional to our entire body. Its possible dysfunction can lead to the production of an insufficient or excessive amount of thyroid hormone. We can look at various machine learning models, compare their accuracies, and made feature selections in an effort to identify the best model to predict hyperthyroid and hypothyroid. It is crucial to catch it early so that doctors can give patients better treatment to prevent it from becoming a significant problem.

Research Goal

- What are the various features in the complete dataset which contribute the most to predict the type of disease?
- Out of all the target classes in the dataset, which are the most commonly occurring one’s?
- Which features contribute the most to predict the disease from the selected target classes?
- Which among the K-NN , Naive Bayes or Random Forest predicts the data best?

Introduction

One of the most prevalent disorders among women is thyroid disease. Thyroid illness frequently manifests as hypothyroid. It is obvious that people with hypothyroid are typically female. Because most people are unaware of that illness, it is quickly developing into a severe illness. It is crucial to catch it so that doctors can provide patients with better treatment. Machine learning illness prediction is a challenging task. In forecasting diseases, machine learning is crucial. There are two different thyroid conditions: Hyperthyroid and Hypothyroid. For this, we used 4 Machine Learning model such as K-NN model, Random Forest, Multinomial Regression model, and Naive Bayes.

About Dataset

The dataset was found on Kaggle
https://www.kaggle.com/datasets/emmanuelfwerr/thyroid-disease-data
and describes Thyroid data for 9172 data with 31 features.

Data Cleaning, Wrangling, Transformations

- Removing the Null values from the Dataset.
- Remapping the Target Labels.

Removing the Null values

```
##
## $hypopituitary
## [1] 0
##
## $psych
## [1] 0
##
## $TSH_measured
## [1] 0
##
## $TSH
## [1] 842
##
## $T3_measured
## [1] 0
##
## $T3
## [1] 2604
##
## $TT4_measured
## [1] 0
##
## $TT4
## [1] 442
##
## $T4U_measured
## [1] 0
##
## $T4U
## [1] 809
##
## $FTI_measured
## [1] 0
##
## $FTI
## [1] 802
##
## $TBG_measured
## [1] 0
##
## $TBG
## [1] 8823
```

```
## [1] 0
##
## $on_thyroxine
## [1] 0
##
## $query_on_thyroxine
## [1] 0
##
## $on_antithyroid_meds
## [1] 0
##
## $sick
## [1] 0
##
## $pregnant
## [1] 0
##
## $thyroid_surgery
## [1] 0
##
## $I131_treatment
## [1] 0
##
## $query_hypothyroid
## [1] 0
##
## $query_hyperthyroid
## [1] 0
##
## $lithium
## [1] 0
##
## $goitre
## [1] 0
##
## $tumor
## [1] 0
##
## $hypopituitary
```

Remapping the Target Labels

#remapping target values

```
library(dplyr)
df2 = df2 %>% mutate(target=recode(target,
                                   '-' = 'negative',
             'A'= 'hyperthyroid',
             'B'= 'hyperthyroid',
             'C'= 'hyperthyroid',
             'D'= 'hyperthyroid',
             'E'= 'hypothyroid',
             'F'= 'hypothyroid',
             'G'= 'hypothyroid',
             'H'= 'hypothyroid'))
```

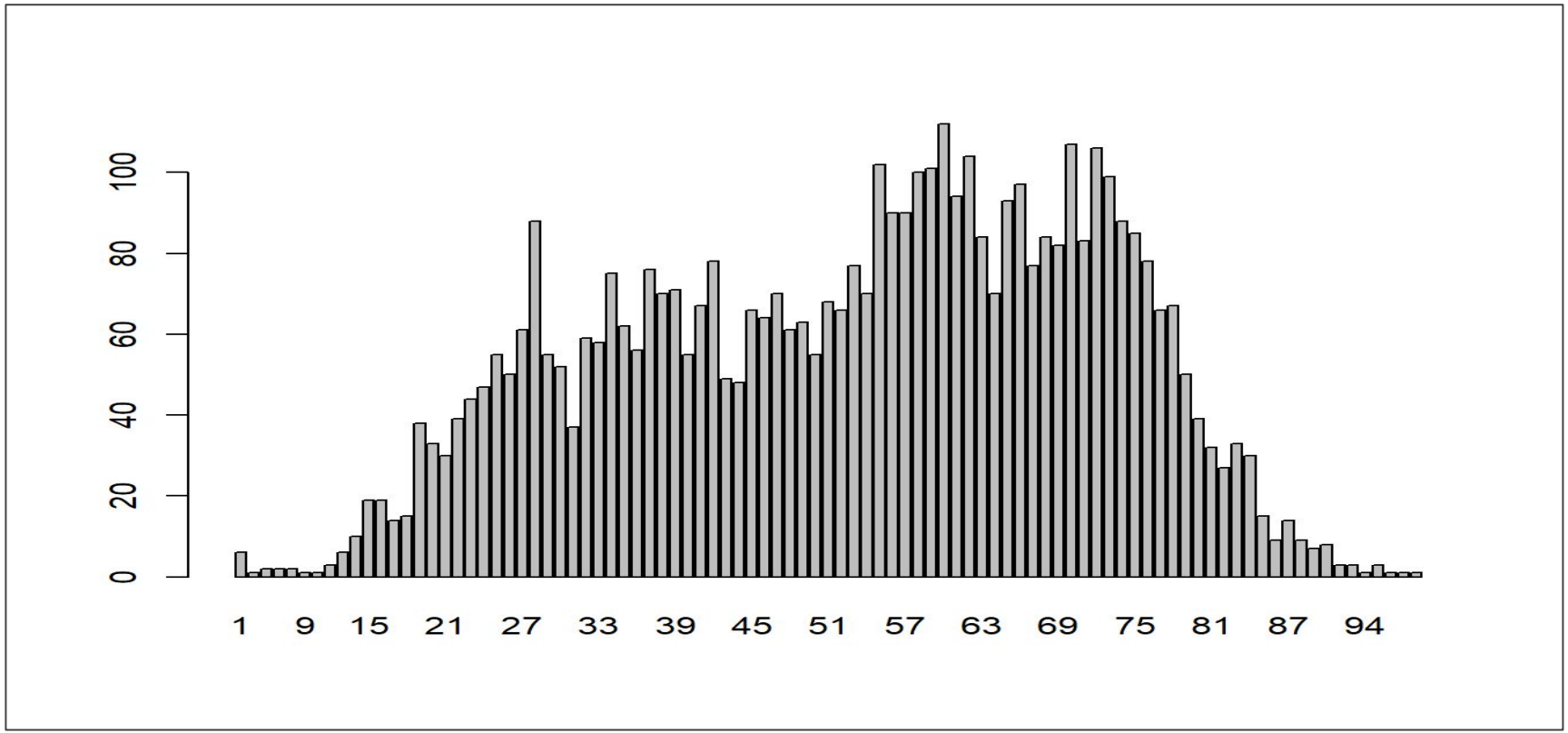
```
View(df2)
```

```
df3 = df2 %>% dplyr::filter(target %in% c("negative", "hyperthyroid","hypothyroid"))
lapply(df3,function(x) { length(which(is.na(x)))})
```

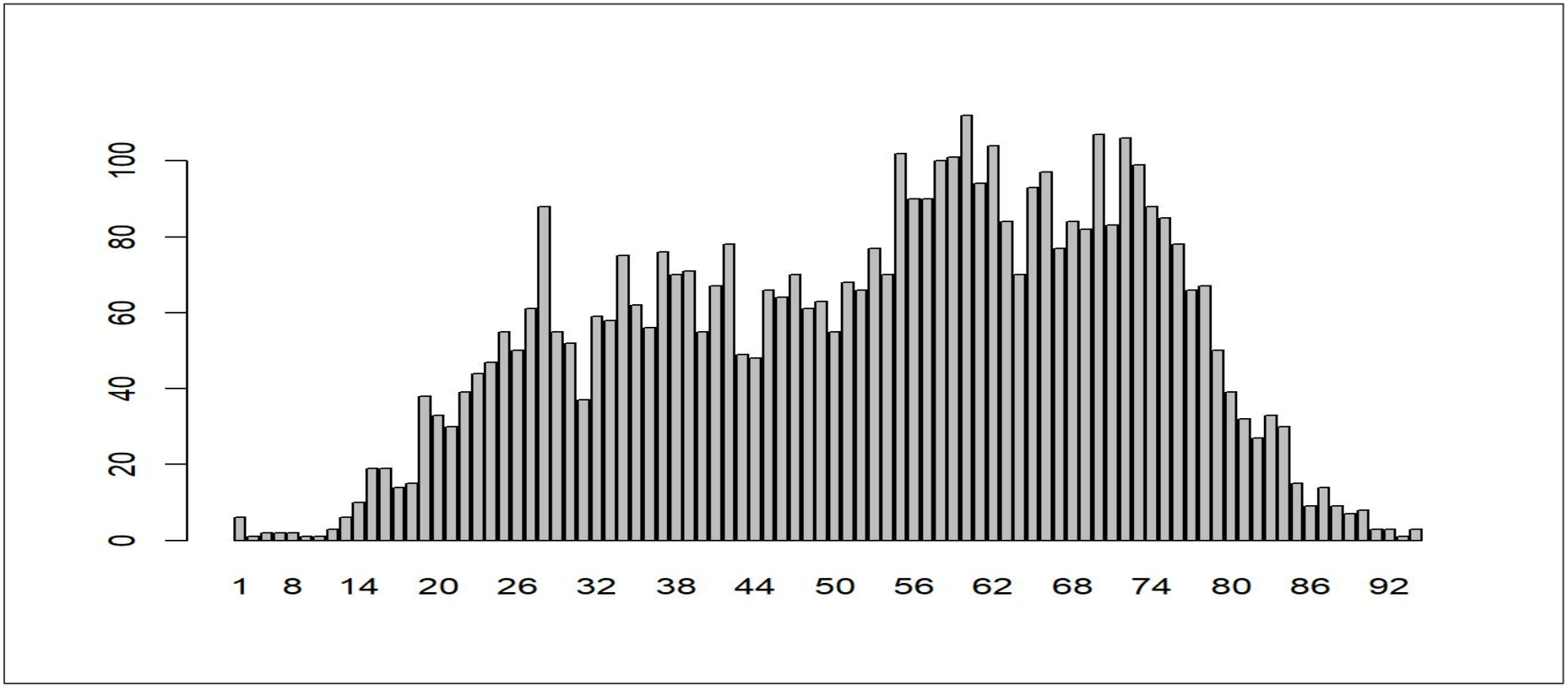
```
## $on_thyroxine
## [1] 0
##
## $query_on_thyroxine
## [1] 0
##
## $on_antithyroid_meds
## [1] 0
##
## $sick
## [1] 0
##
## $pregnant
## [1] 0
##
## $thyroid_surgery
## [1] 0
##
## $I131_treatment
## [1] 0
##
## $query_hypothyroid
## [1] 0
##
## $query_hyperthyroid
## [1] 0
##
## $lithium
## [1] 0
##
## $goitre
## [1] 0
##
## $tumor
## [1] 0
##
## $hypopituitary
## [1] 0
##
```

Exploratory Data Analysis

Age vs No of People

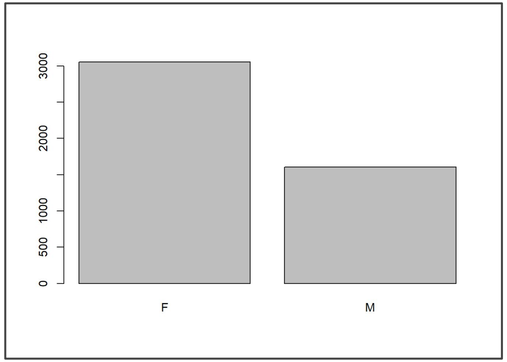


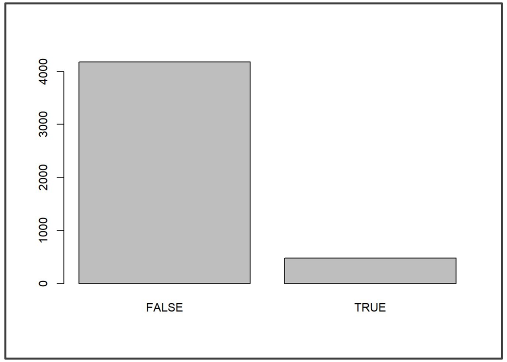
Age(<100) vs Number of People

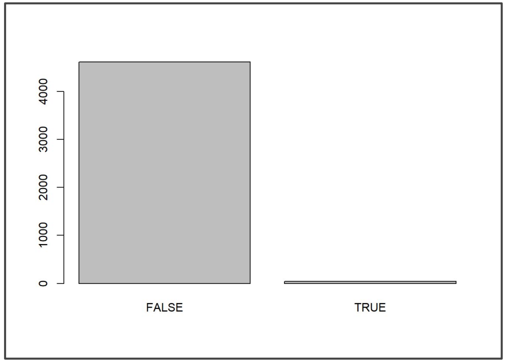


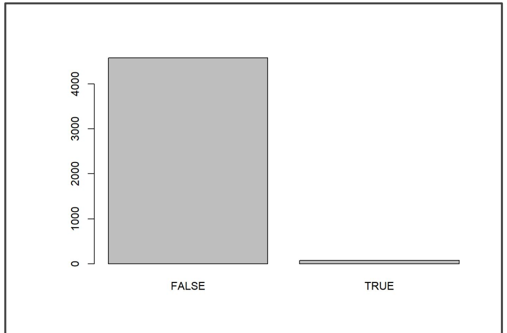
Visualizing the Features

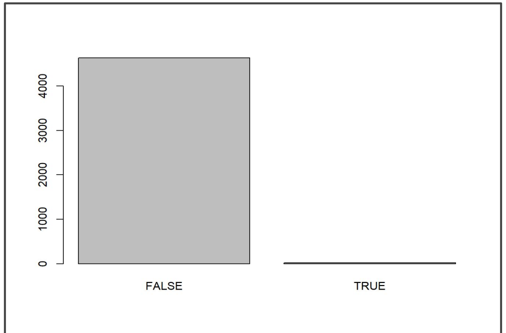
We had visualized the each feature by comparing the count. By this we can get an overview of data.
Below are some plots of our Features.

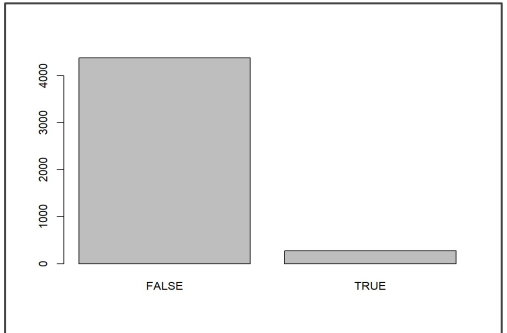






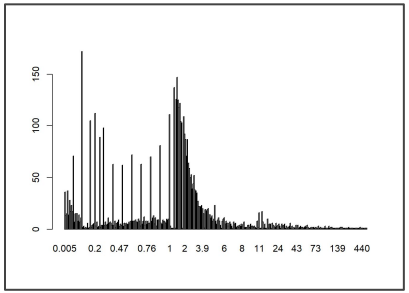


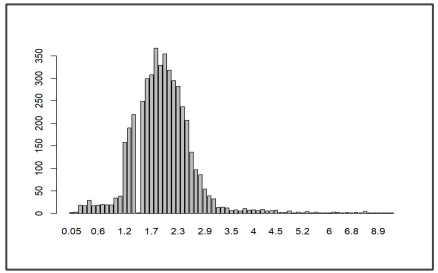


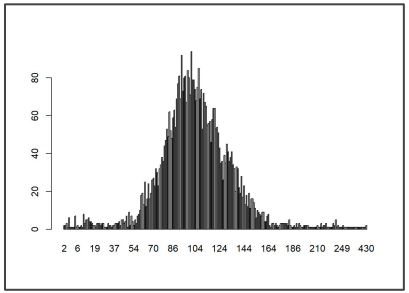


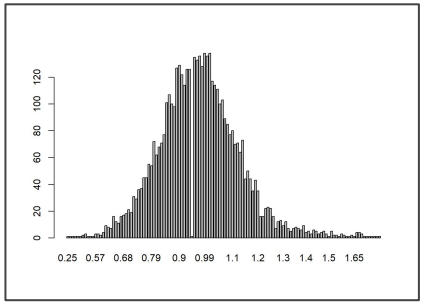
Visualizing the People vs Blood Levels (w.r.t features)

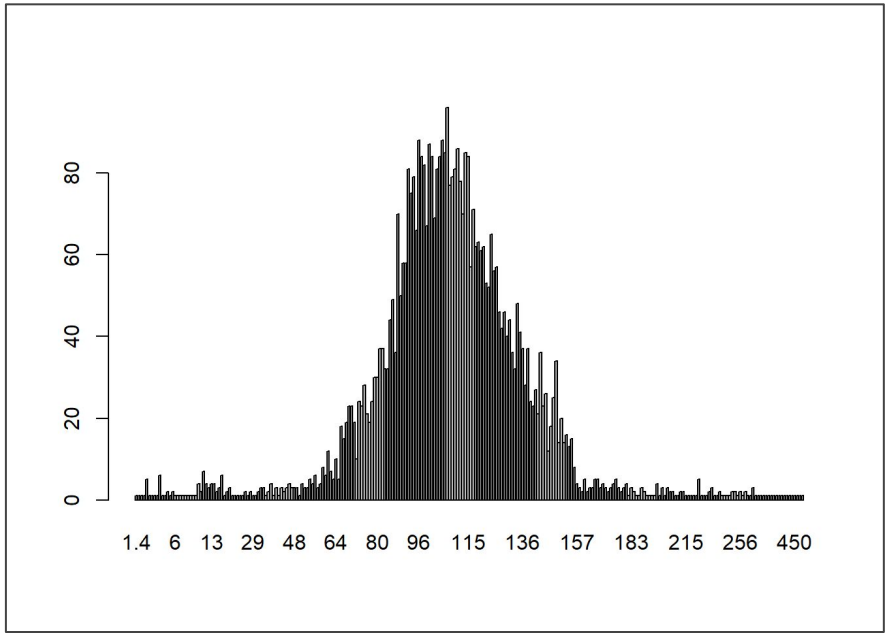
We can visualize the different features TSH, T3, TT4, FTI, T4U through which can get the more insights about the data.



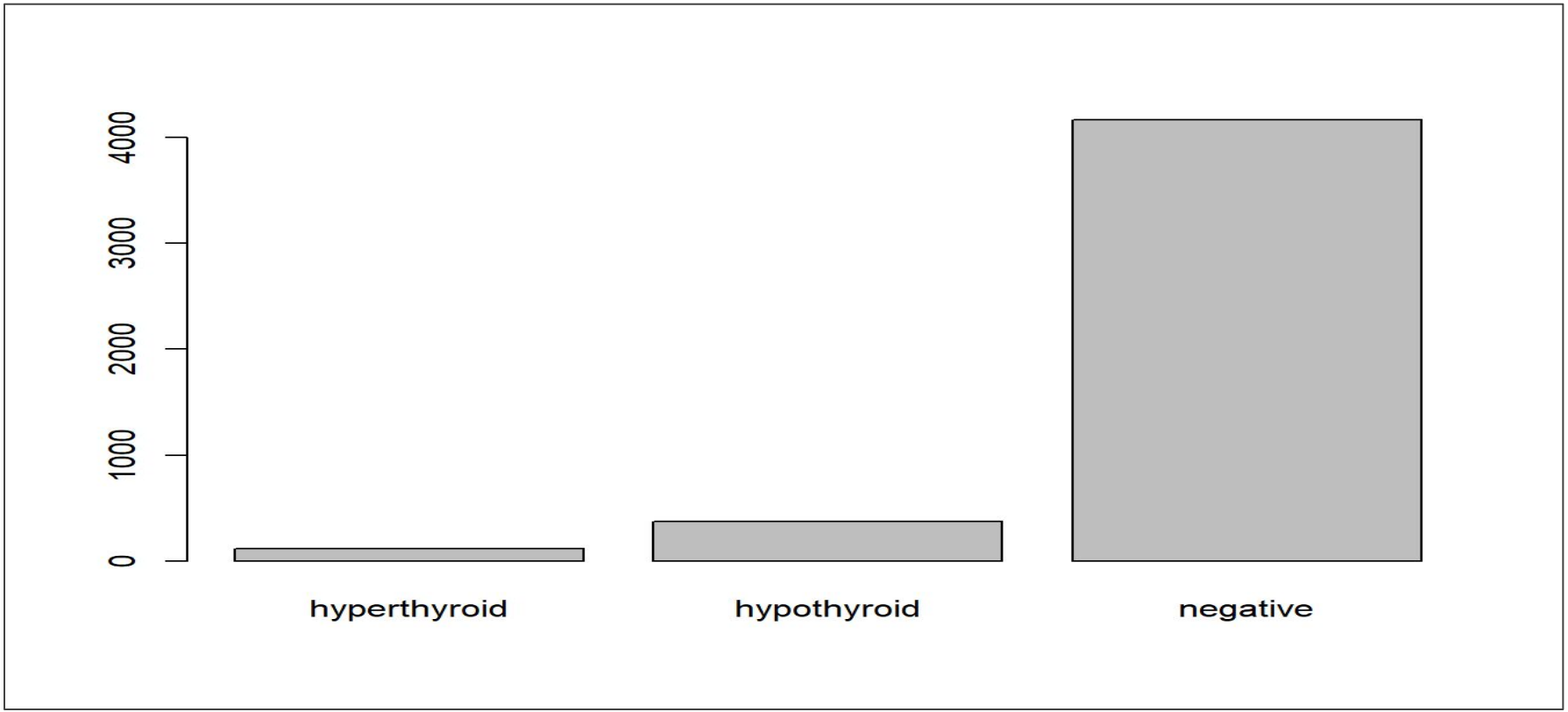




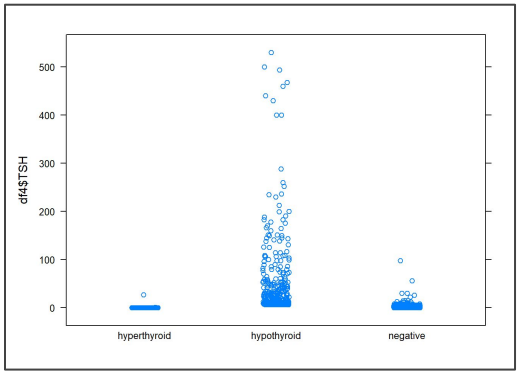


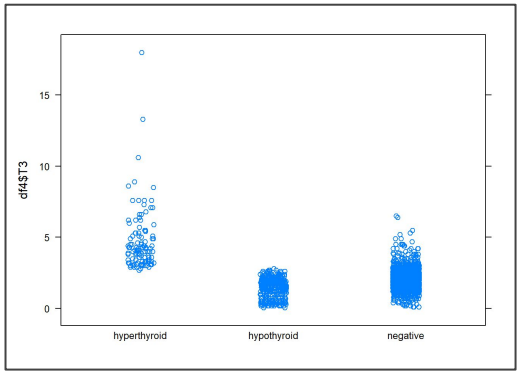


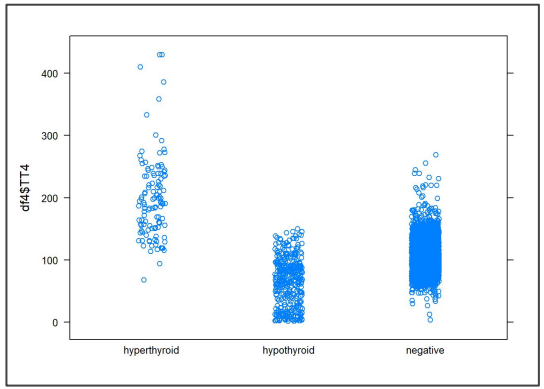
Visualizing the Target Label

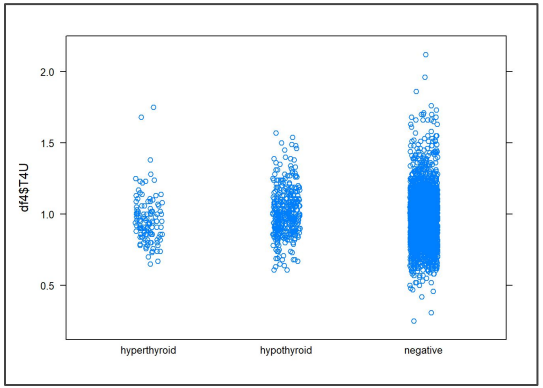


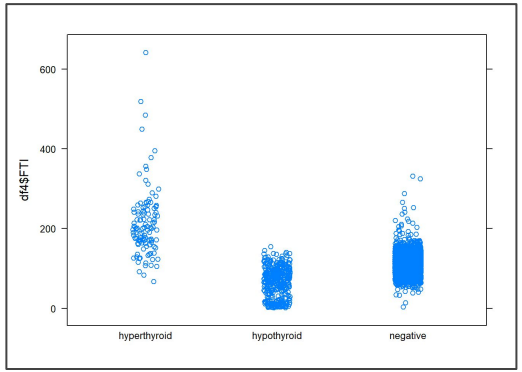
Strip Plots for Target value w.r.t Blood levels

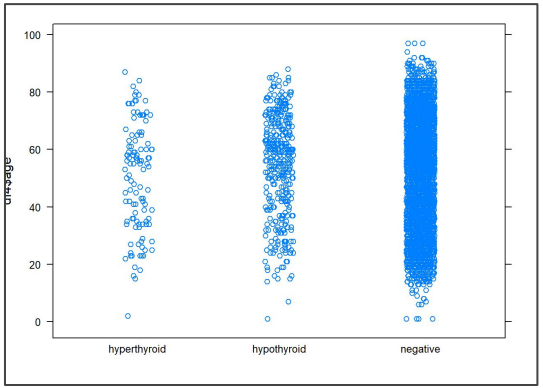




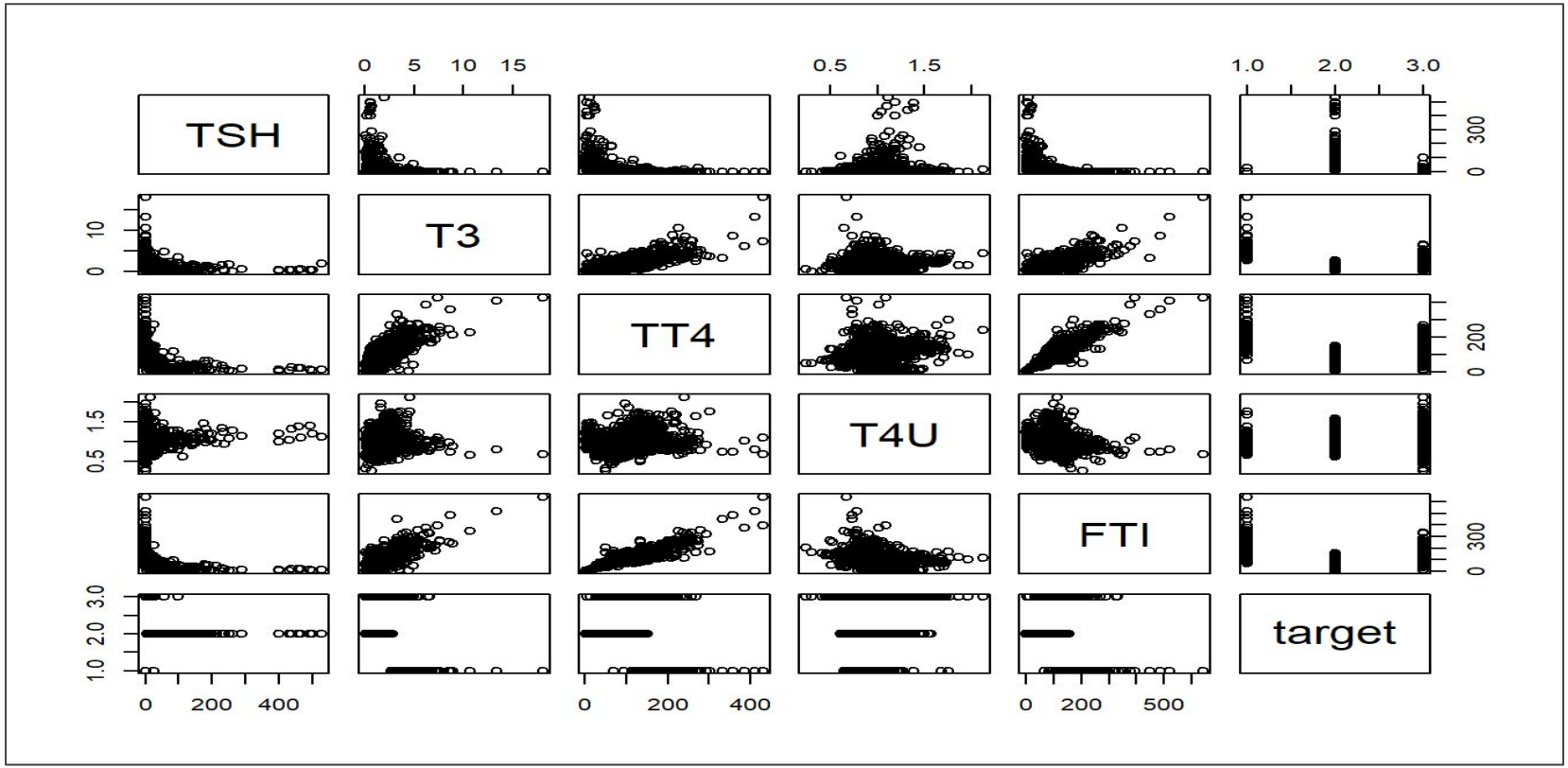








Visualizing the Blood Column Features w.r.t Target



Types of Features in the Dataset

```
## $age
## [1] "double"
##
## $sex
## [1] "character"
##
## $on_thyroxine
## [1] "logical"
##
## $query_on_thyroxine
## [1] "logical"
##
## $on_antithyroid_meds
## [1] "logical"
##
## $sick
## [1] "logical"
##
## $pregnant
## [1] "logical"
##
## $thyroid_surgery
## [1] "logical"
##
## $I131_treatment
## [1] "logical"
##
## $query_hypothyroid
## [1] "logical"
##
## $query_hyperthyroid
## [1] "logical"
##
## $lithium
## [1] "logical"
##
## $goitre
## [1] "logical"
##
```

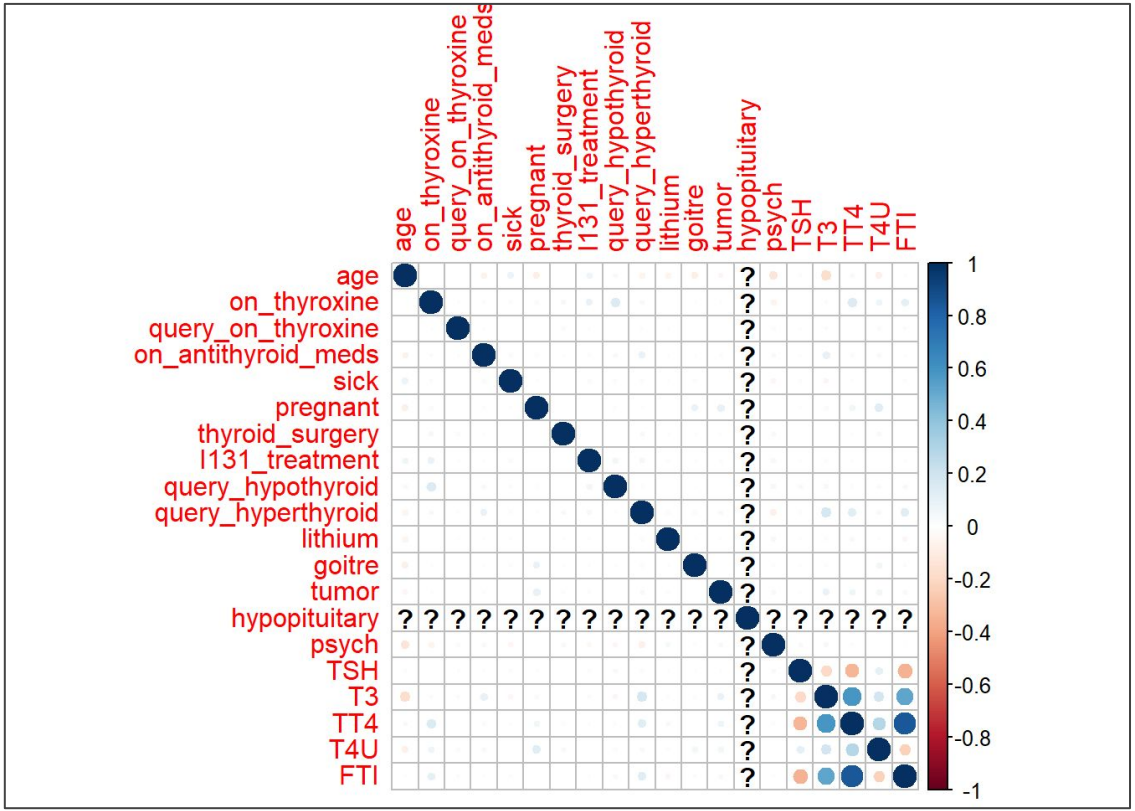
#converting all logical columns to numeric for plotting correlation

```
cols <- sapply(df4, is.logical)
df4[,cols] <- lapply(df4[,cols], as.numeric)
head(df4)
```

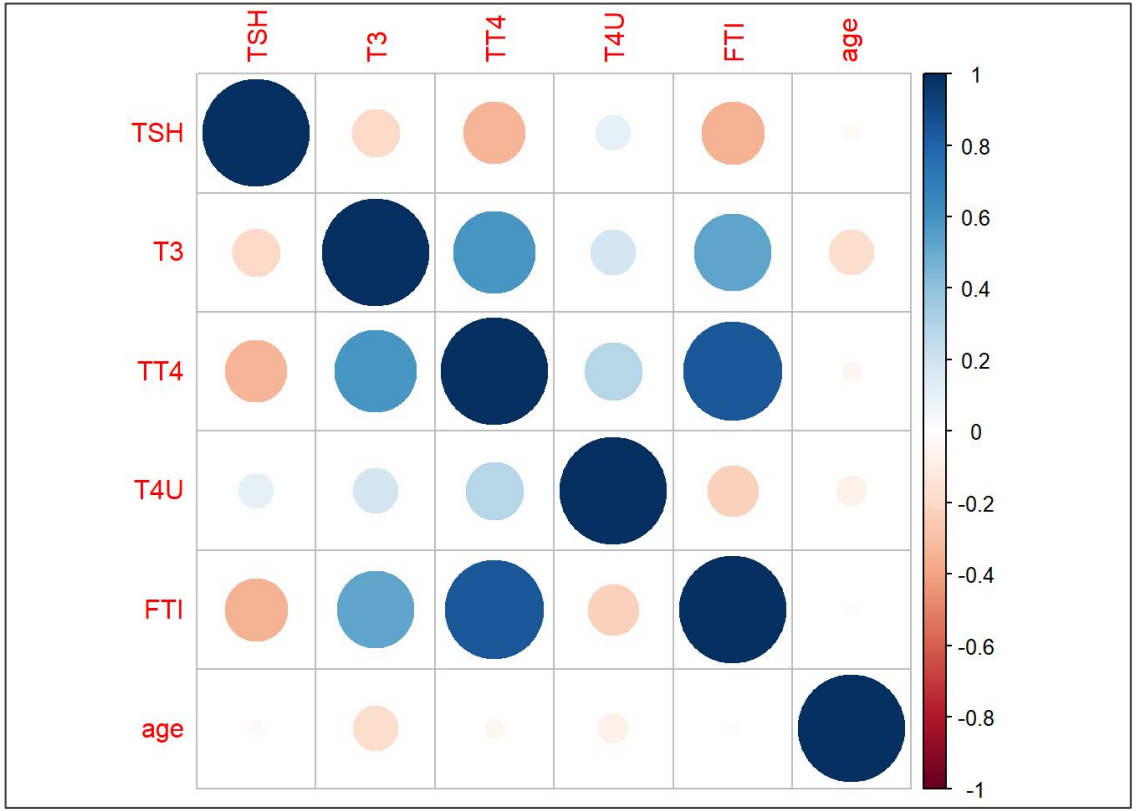
a...	s...	on_thyroxine	query_on_thyroxine	on_antithyroid_meds	sick	pregnant	thyroid_surgery
<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
36	F	0	0	0	0	0	0
40	F	0	0	0	0	0	0
40	F	0	0	0	0	0	0
77	F	0	0	0	0	0	0
51	F	0	0	0	0	0	0
56	M	0	0	0	0	0	0

6 rows | 1-8 of 22 columns

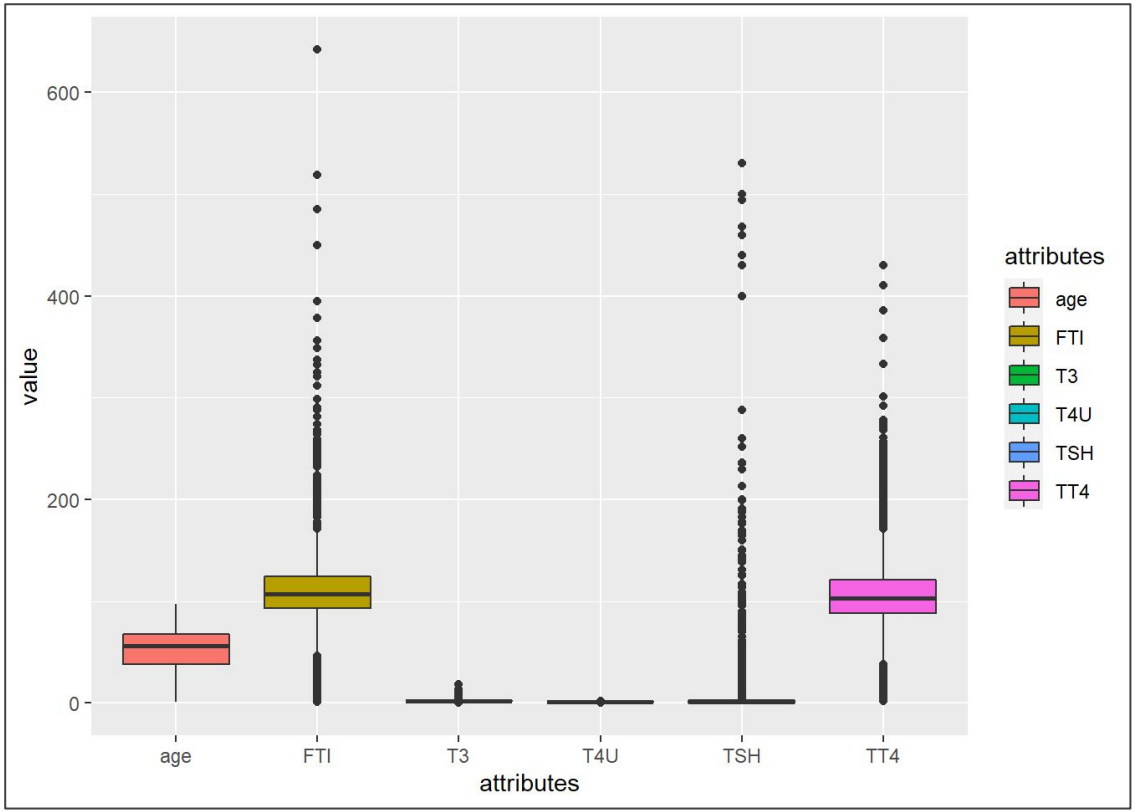
Correlation Between Features



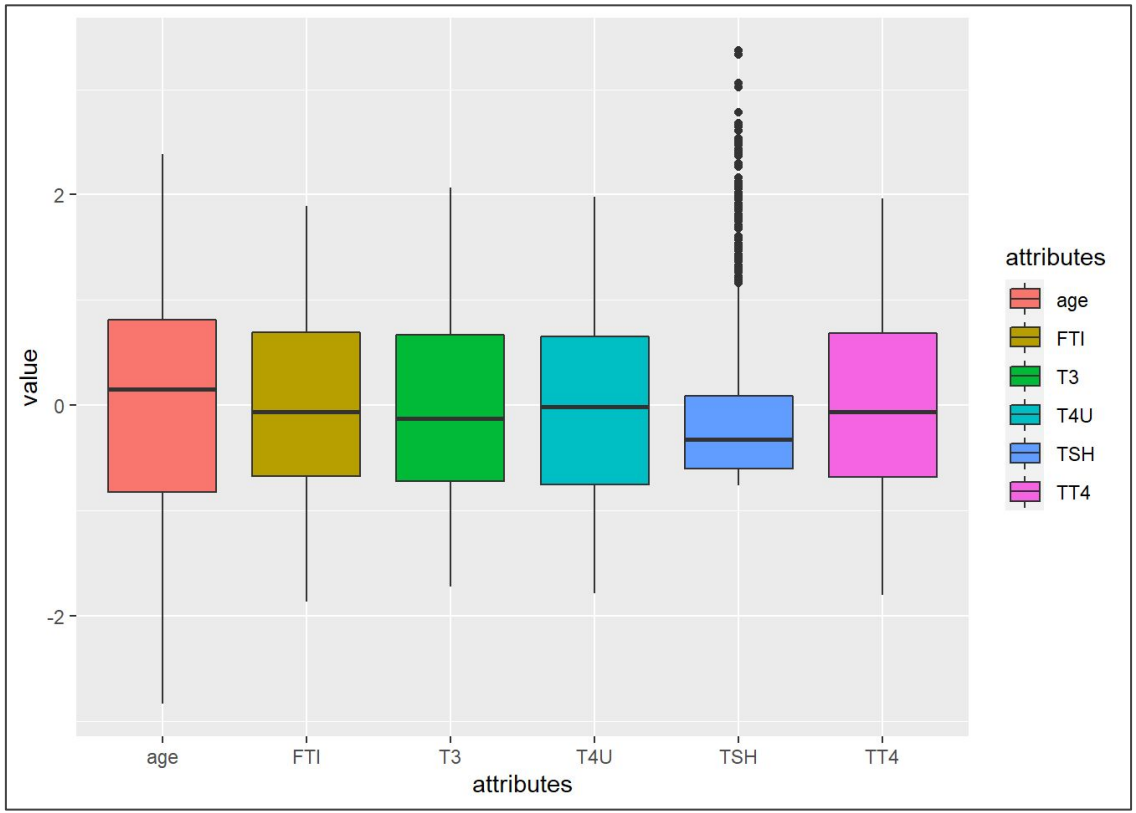
Correlation between Blood level columns and age



Box Plot for KNN- classifier before scaling



Box Plot for KNN- classifier after scaling



Modeling

- K-NN
- Random Forest
- Naive Bayes
- Multinomial Regression Model

Model's Training and Testing Accuracies

Model	Training Accuracy	Testing Accuracy
K-NN before scaling	95.66%	94.44%
K-NN after Scaling	98.46%	97.31
Random Forest	99.78%.	98.86%
Naive Bayes	96.8%	97.1%
Multinomial Regression Model	97.13%	97.64%.

Future Scope

- We compared our results using the Classification models but we can also design a neural network model which might be more efficient in prediction.

- We may employ feature selection techniques, which offer me a better understanding of which features to include in modeling and which might be more accurate.

Conclusion

- Out of all models we designed ,Random Forest model has better testing accuracy.
- While the trees are developing, the random forest adds more randomness to the model. When splitting a node, it looks for the best feature from a random subset of features rather than the most crucial feature.
- As you could see the difference in accuracies of K-NN model before processing the data and after processing the data by removing outliers and scaling which is definitely a considerable one, we get to see the importance of preparing our data well to achieve better model accuracies.
- Also, the scenarios would have been the same if tested on other models rather than KNN where the before preparation the accuracies would be lower and post preparation the accuracies would be greater.

THANK YOU !!