Assignment-5
- Ds441
- **Arun Subbiah Subbiah,**
- STS_ID - #2105313

AIM:

To analyze and compare Classification and Clustering models on strokes experienced by various groups of people.

a.

The data was collected from the website Kaggle, which comprises 5110 observations. The variables id, gender, age, hypertension, heart disease, ever married, work type, residence type, avg glucose level, BMI, and smoking statues were used to detect Heart stork.

Gender, ever married, employment type, and residence type are category variables, while id, age, bmi, and avg glucose level are numerical variables, and hypertension, heart disease, and stroke are binary variables.

A stroke is a medical condition in which the brain receives insufficient blood supply, resulting in cell death. hence In modern technology era, technological improvements and health factors are entirely lessened, resulting in a range of ailments and strokes in children that are less expected and cared for; so, this project is for stork prediction.

Variable information:

id: unique identifier

2) gender: "Male", "Female" or "Other"

3) age: age of the patient

4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension

5) heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease

6) ever_married: "No" or "Yes"

7) work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"

8) Residence_type: "Rural" or "Urban"

9) avg_glucose_level: average glucose level in blood

10) bmi: body mass index

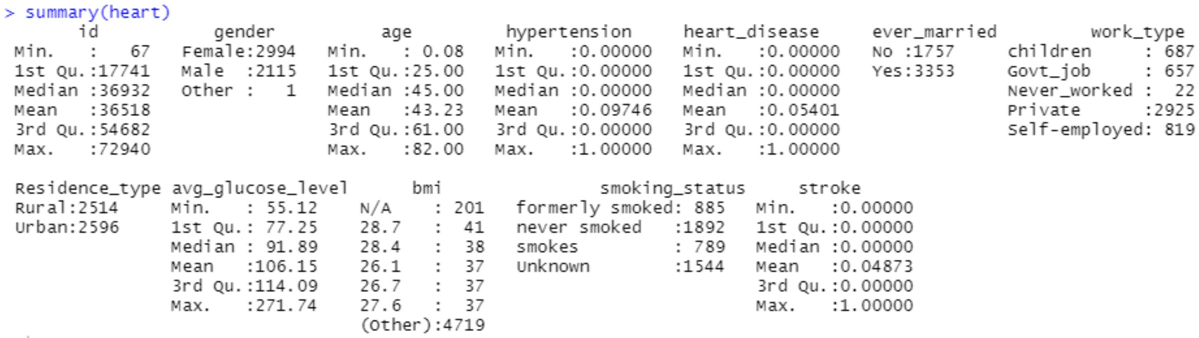
11) smoking_status: "formerly smoked", "never smoked", "smokes" or

"Unknown"*

12) stroke: 1 if the patient had a stroke or 0 if not

*Note: "Unknown" in smoking_status means that the information is unavailable for this patient

b. DATA EXPLORATION



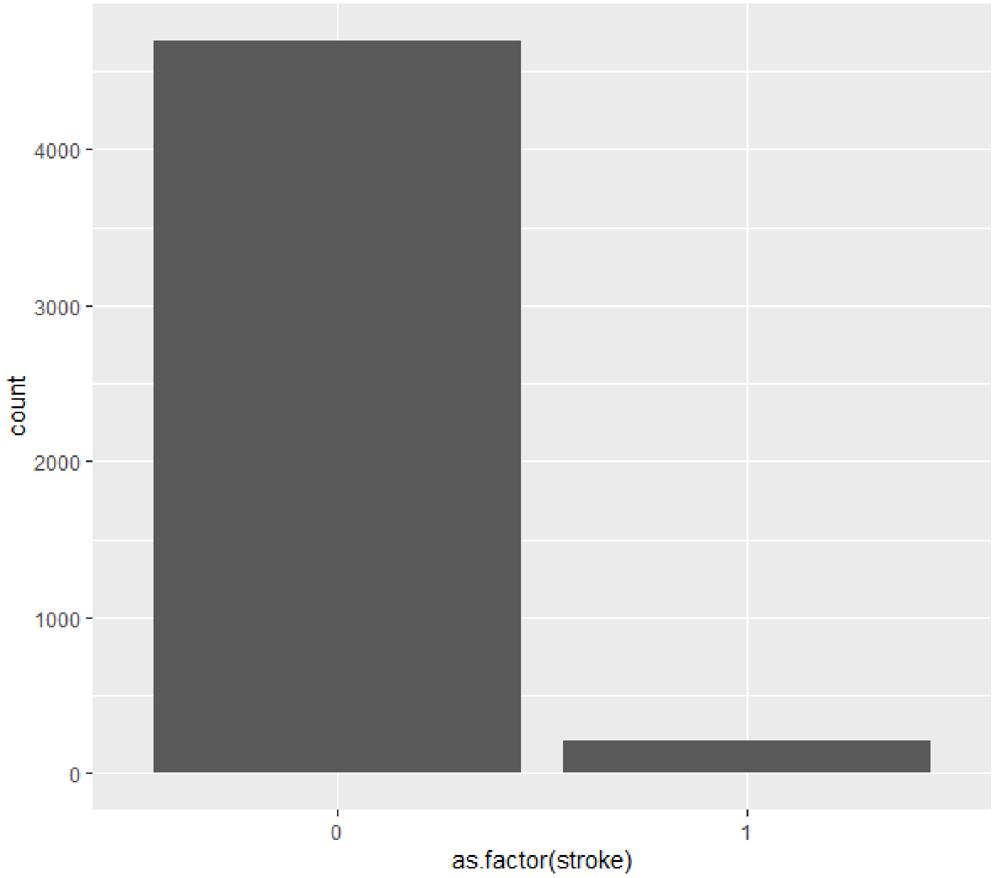
```
> summary(heart)
       id            gender          age         hypertension      heart_disease     ever_married          work_type   
 Min.   :   67   Female:2994   Min.   : 0.08   Min.   :0.00000   Min.   :0.00000   No :1757     children     : 687  
 1st Qu.:17741   Male  :2115   1st Qu.:25.00   1st Qu.:0.00000   1st Qu.:0.00000   Yes:3353     Govt_job     : 657  
 Median :36932   Other :   1   Median :45.00   Median :0.00000   Median :0.00000                Never_worked :  22  
 Mean   :36518                 Mean   :43.23   Mean   :0.09746   Mean   :0.05401                Private      :2925  
 3rd Qu.:54682                 3rd Qu.:61.00   3rd Qu.:0.00000   3rd Qu.:0.00000                Self-employed: 819  
 Max.   :72940                 Max.   :82.00   Max.   :1.00000   Max.   :1.00000                                    

 Residence_type avg_glucose_level      bmi                 smoking_status     stroke       
 Rural:2514     Min.   : 55.12    N/A    : 201   formerly smoked: 885   Min.   :0.00000  
 Urban:2596     1st Qu.: 77.25    28.7   :  41   never smoked   :1892   1st Qu.:0.00000  
                Median : 91.89    28.4   :  38   smokes         : 789   Median :0.00000  
                Mean   :106.15    26.1   :  37   Unknown        :1544   Mean   :0.04873  
                3rd Qu.:114.09    26.7   :  37                          3rd Qu.:0.00000  
                Max.   :271.74    27.6   :  37                          Max.   :1.00000  
                                  (Other):4719                                           
```

The summary function is used to find the Summary Statistics of all the variables in the data set heart.

According to the summary details, the minimum age recorded is 0.08, the maximum age recorded is 82, and the median and mean are 45 and 43, which are nearly identical.

50% of the population is between the ages of 25 and 61. As a result, the variable age is linear.

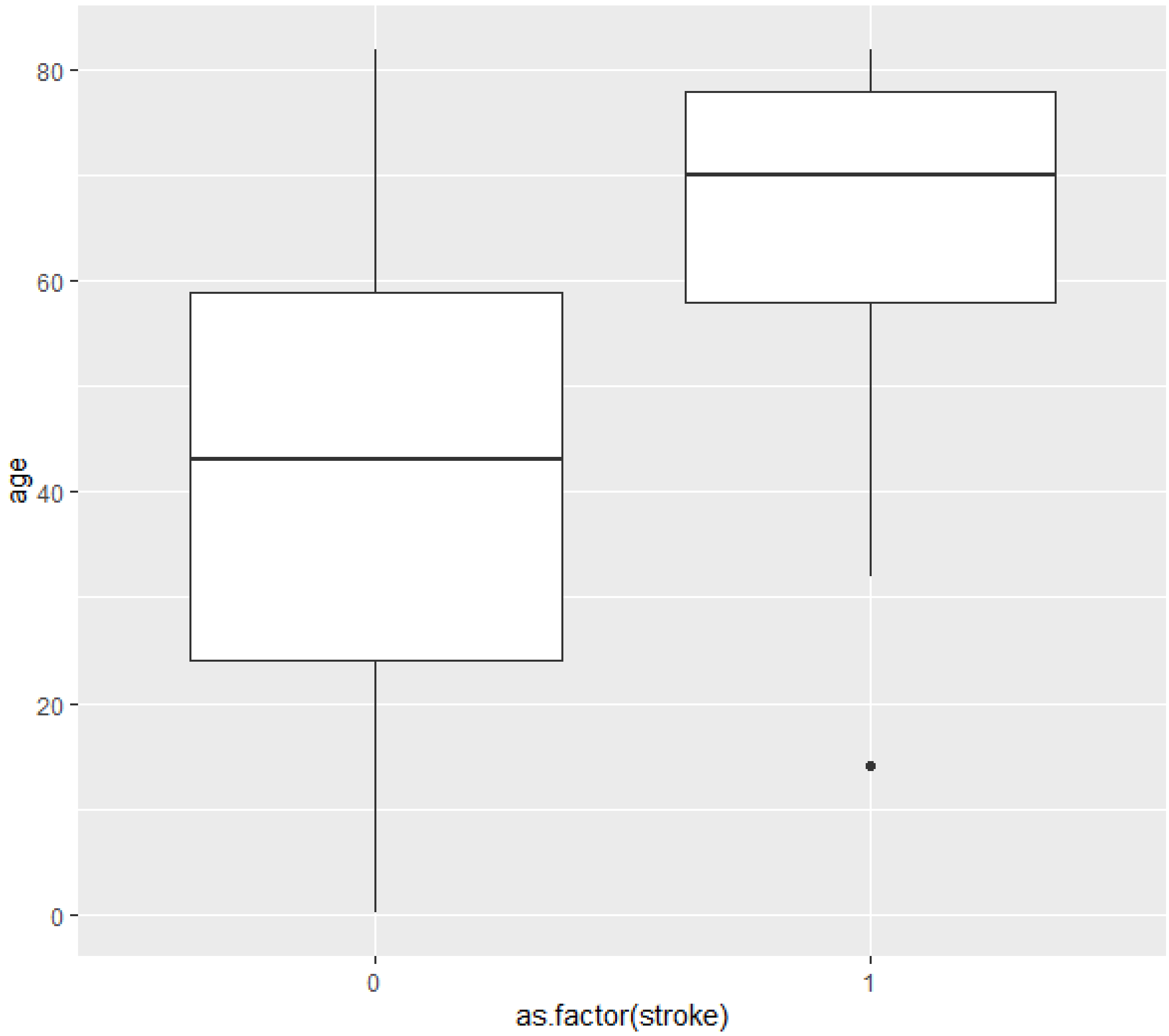
We can see from the Residence type that there are 2596 people from urban and 2514 people from rural. Taking BMI, we can see that there are 201 N/A values, so in the data cleaning procedure, we must eliminate the n/a values and continue.



The above is the bar plot for stroke and its frequency from the dataset we could see that only a less amount of people have had a stroke and the majority does not have had a stroke before

And in conclusion, we could say that there are very fewer data collected for patients who had stork.

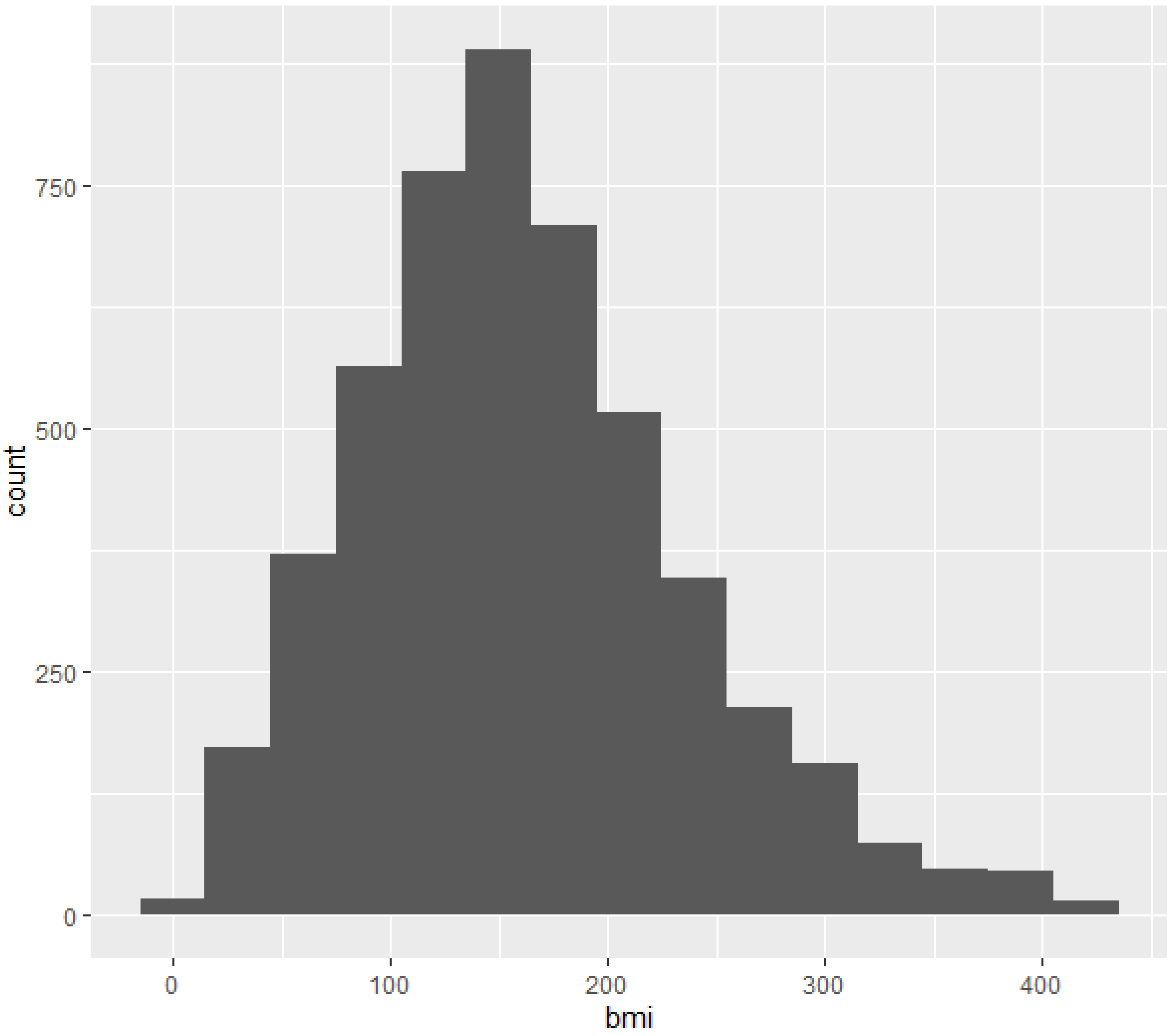
Hence this kind of data would let to a less precise prediction model due to fewer data.



The figure(boxplot) is graphed for the stroke Vs the age Where 0 is never had stroke and 1 is had a stroke before. From the figure, we could see that people who never had a stroke are recorded with a minimum age of 8 months which is lower than the people who have had a stroke before – 15, and the maximum- age has been recorded for both groups which is around 80. Most of the age for people who don’t have a stroke is between 25-60.

We could see from the figure that both are left skewed and hence they are not normally distributed. The interquartile range for people who don’t have a stroke is longer than the other, implying that they are more dispersed.

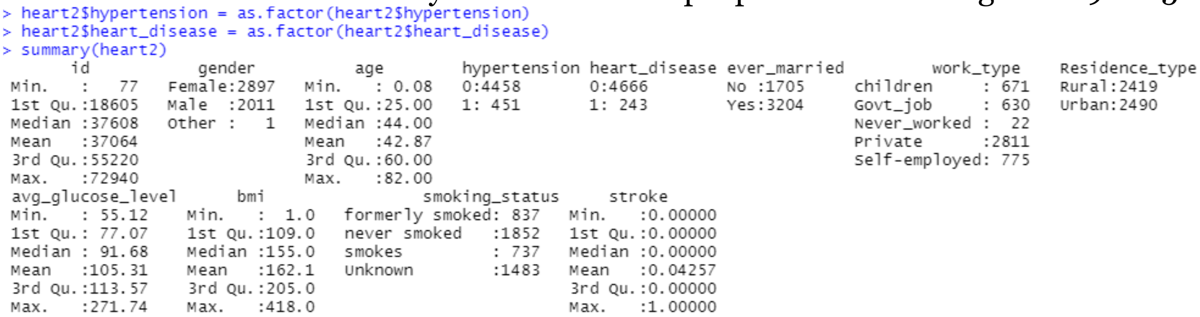
And in the plot people who had a stroke have an outlier at age=15.



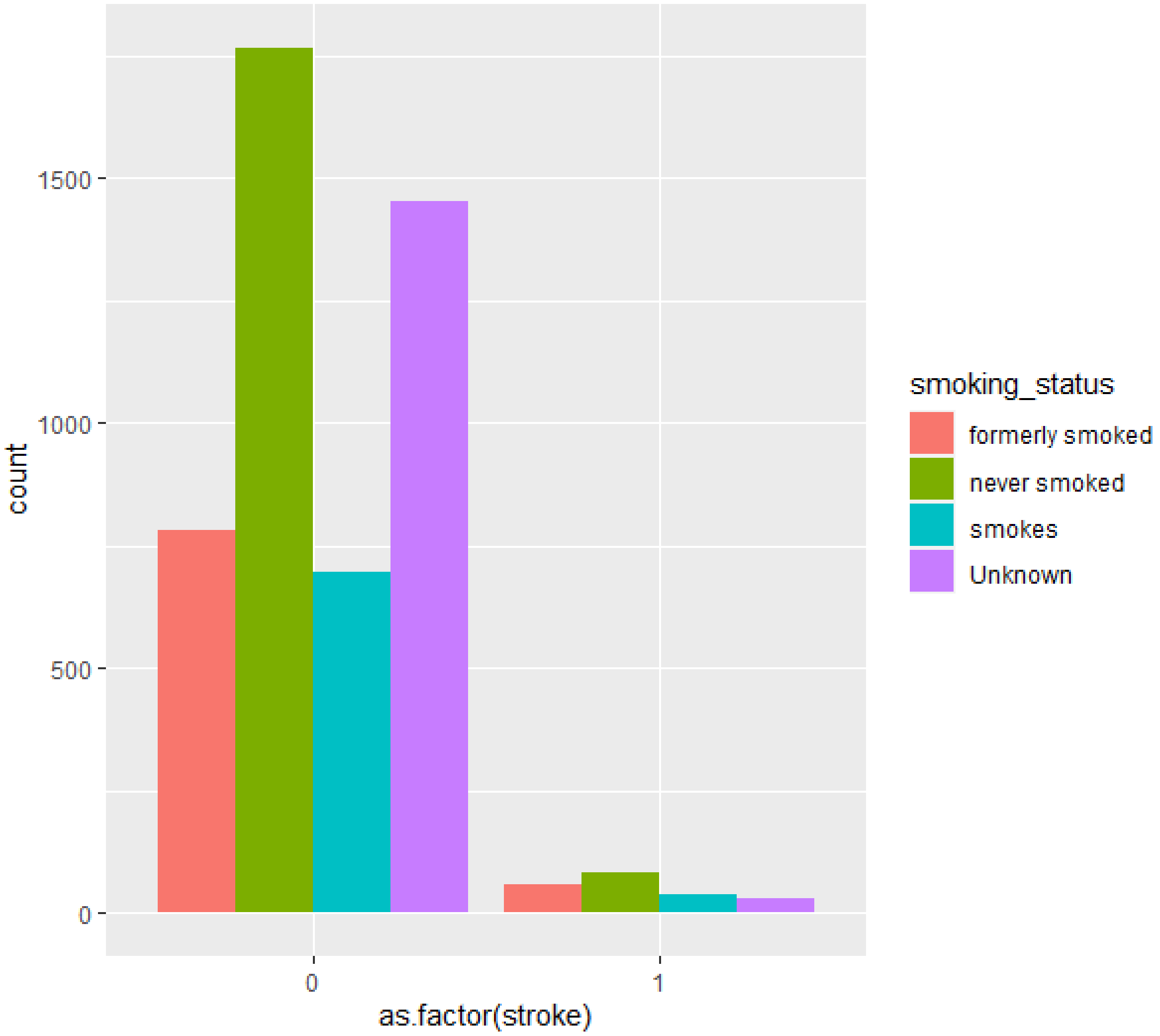
The above histogram is plotted for the bmi variable and its Right skewed.

- It is clear from the histogram plot that it is right skewed. The minimum bmi value is 0 The highest cont1 value is around 400, with a range of 400. Around 150 there is a peak. which is the mode.

The center or the median value is 155 and the mean lies at 162.
50% of the cont1 value is between 109 to 205. Based on this information, we can conclude that bmi has no outliers hence mean and median aren't closest to each other with minium distance. And from the graph, it's very straight that the bmi is right skewed. Analysis of the graph reveals that there are outliers.
And as conclusion we could say that most of the people lies in the range of 109 -205

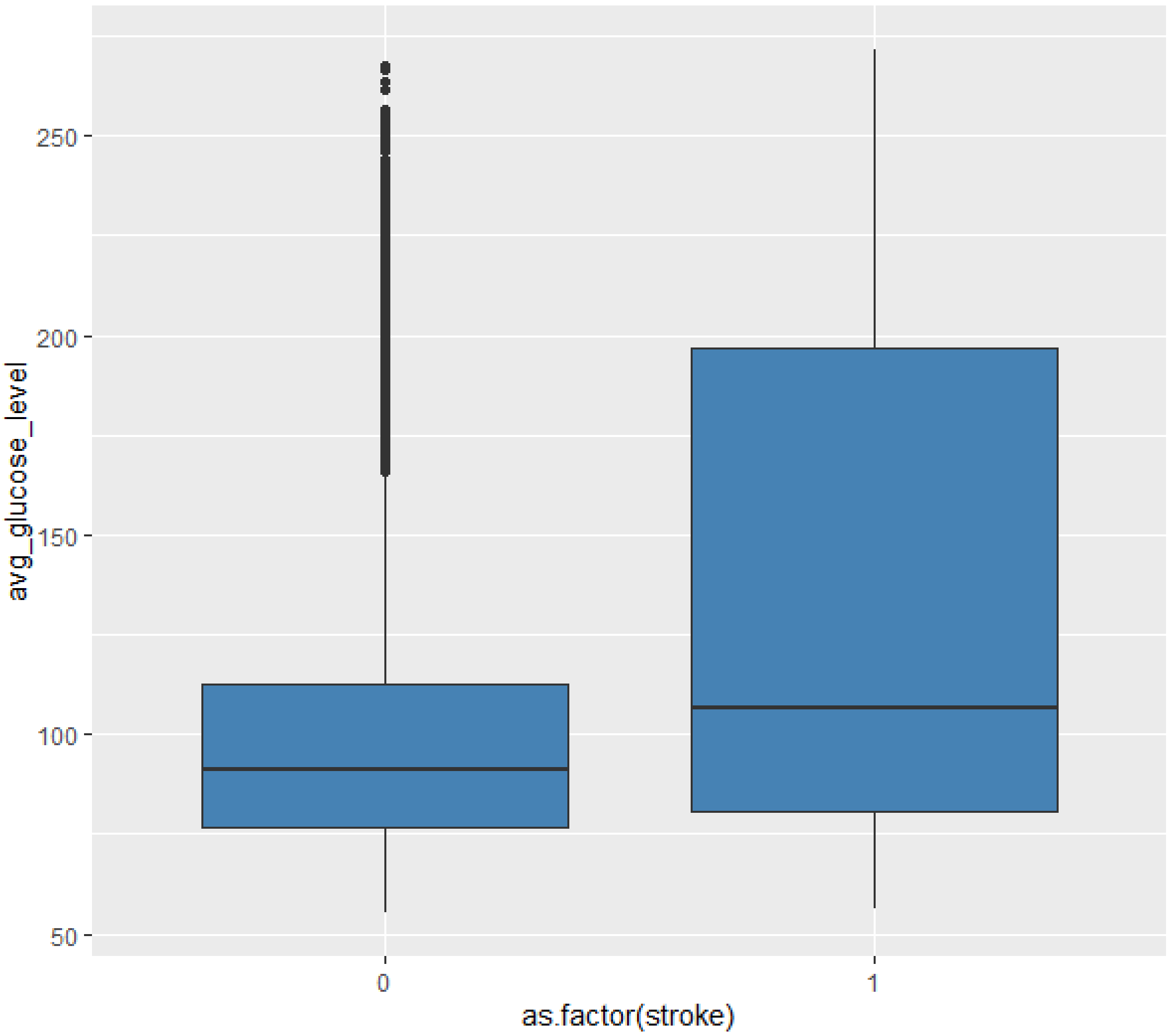


```
> heart2$hypertension = as.factor(heart2$hypertension)
> heart2$heart_disease = as.factor(heart2$heart_disease)
> summary(heart2)
       id           gender          age         hypertension heart_disease ever_married          work_type     Residence_type
 Min.   :   77   Female:2897   Min.   : 0.08   0:4458       0:4666        No :1705     children     : 671   Rural:2419    
 1st Qu.:18605   Male  :2011   1st Qu.:25.00   1: 451       1: 243        Yes:3204     Govt_job     : 630   Urban:2490    
 Median :37608   Other :   1   Median :44.00                                           Never_worked :  22                 
 Mean   :37064                 Mean   :42.87                                           Private      :2811                 
 3rd Qu.:55220                 3rd Qu.:60.00                                           Self-employed: 775                 
 Max.   :72940                 Max.   :82.00                                                                              
 avg_glucose_level      bmi                  smoking_status     stroke       
 Min.   : 55.12    Min.   :  1.0   formerly smoked: 837   Min.   :0.00000  
 1st Qu.: 77.07    1st Qu.:109.0   never smoked   :1852   1st Qu.:0.00000  
 Median : 91.68    Median :155.0   smokes         : 737   Median :0.00000  
 Mean   :105.31    Mean   :162.1   Unknown        :1483   Mean   :0.04257  
 3rd Qu.:113.57    3rd Qu.:205.0                          3rd Qu.:0.00000  
 Max.   :271.74    Max.   :418.0                          Max.   :1.00000  
```



The graph is plotted for stroke and its frequency filled with smoking_status from this graph we could infer that there are more data recorded in people without stroke and from that we could see most of them are non-smokers and the least are recorded for smokers.

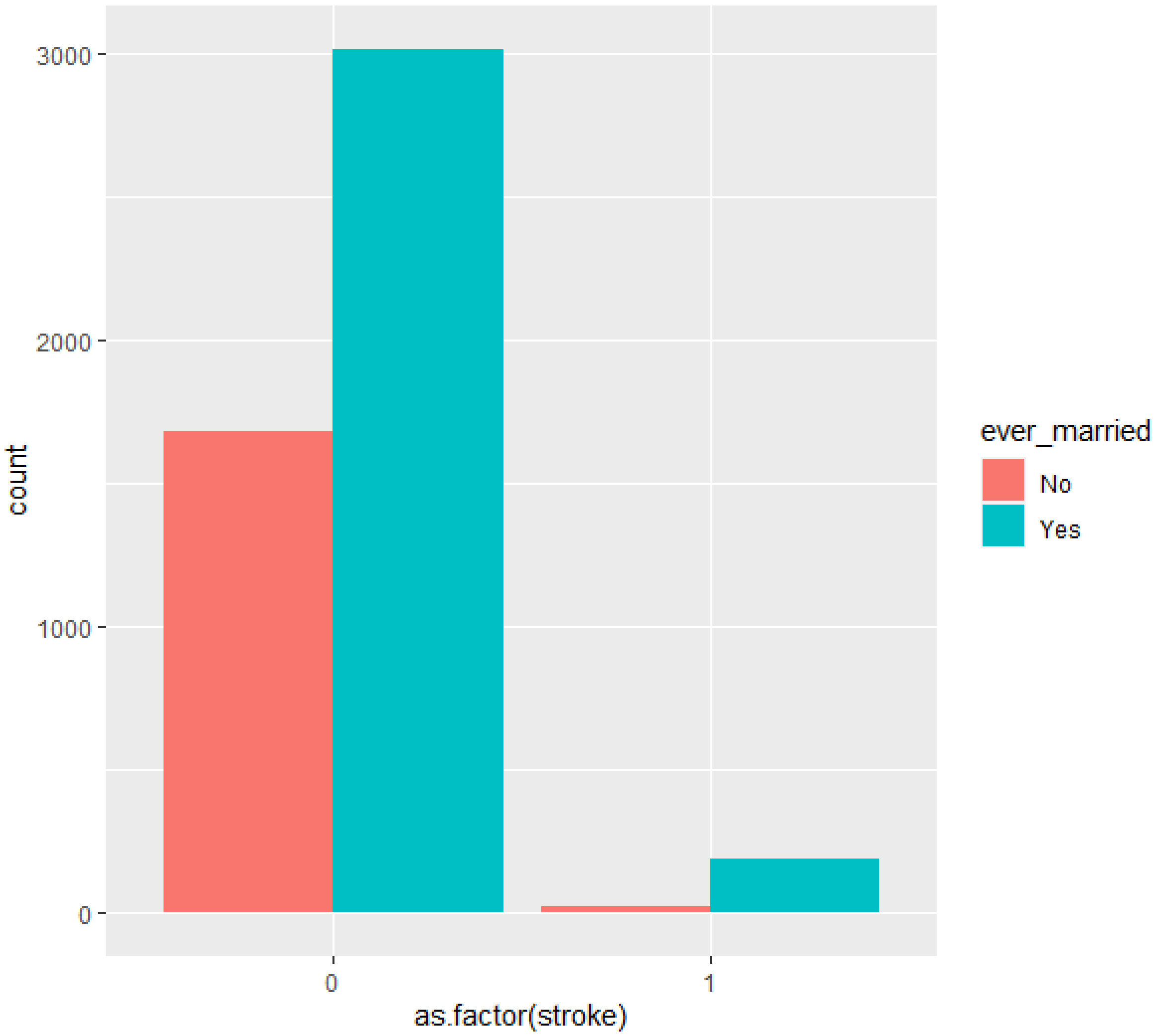
On the other hand, people who had a stroke also have never been smokers at the maximum and unknown at the minimum from the given data we could say that even people who do not smoke get strokes due to other reasons.



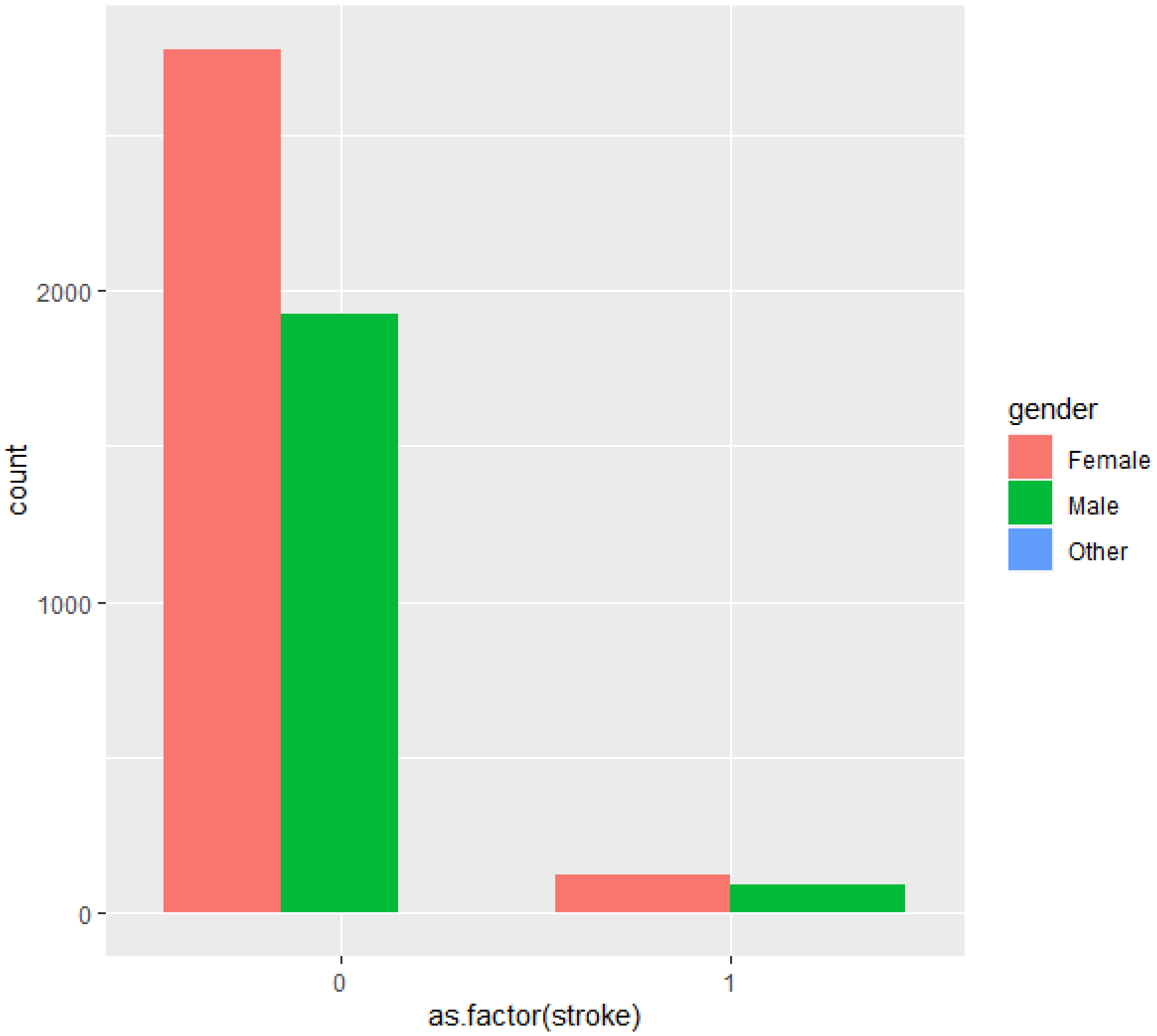
The figure(boxplot) is graphed for the stroke Vs the average glucose level Where 0 is never had a stroke and 1 is had a stroke before. From the figure, we could see that both set of people have the same maximum and minimum values but for people who never had a stroke before graph shows a lot of outliers in the data towards the upper limit.

We could see from the figure that both are right skewed and hence they are not normally distributed. The interquartile range for people who have a stroke is longer than the other, implying that they are more dispersed.

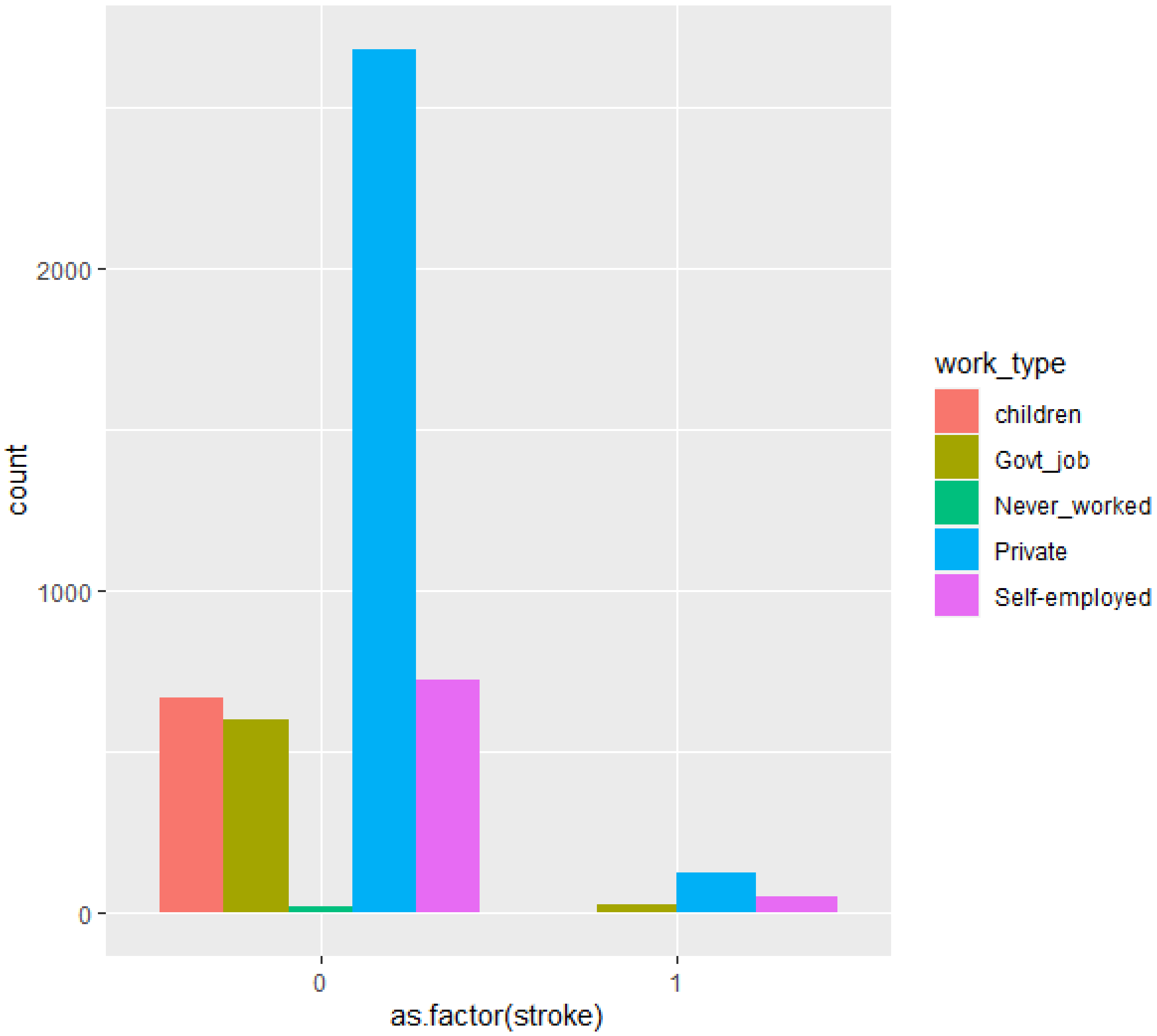
And in the plot people who never had a stroke have lot of outlier.



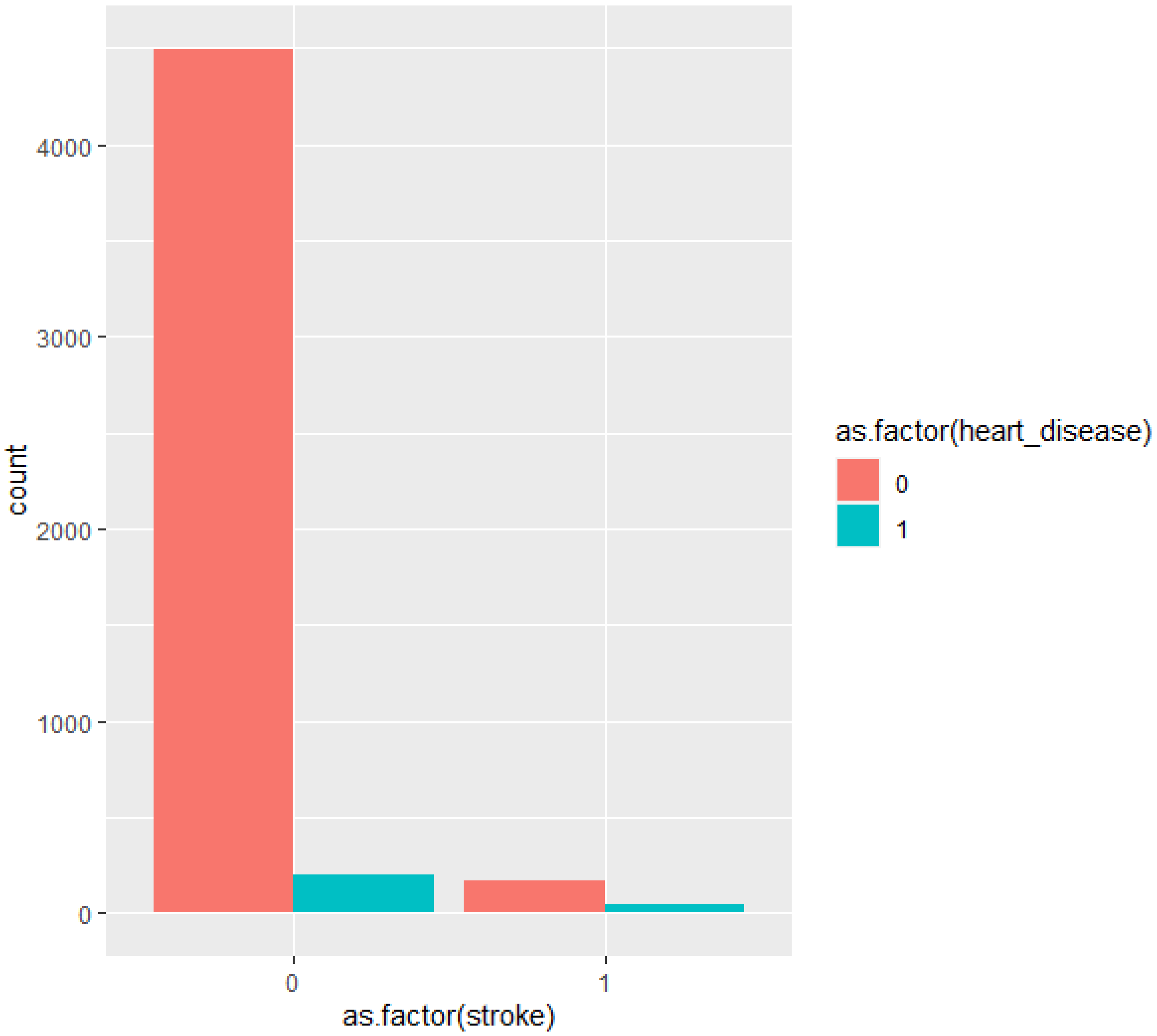
The graph is plotted for stroke and its frequency filled with ever married from this graph we could infer that there are more data recorded in people without stroke and from that we could see most of them are married. People who had a stroke also have married at maximum. from the produced data we could say that even people most married people get strokes.



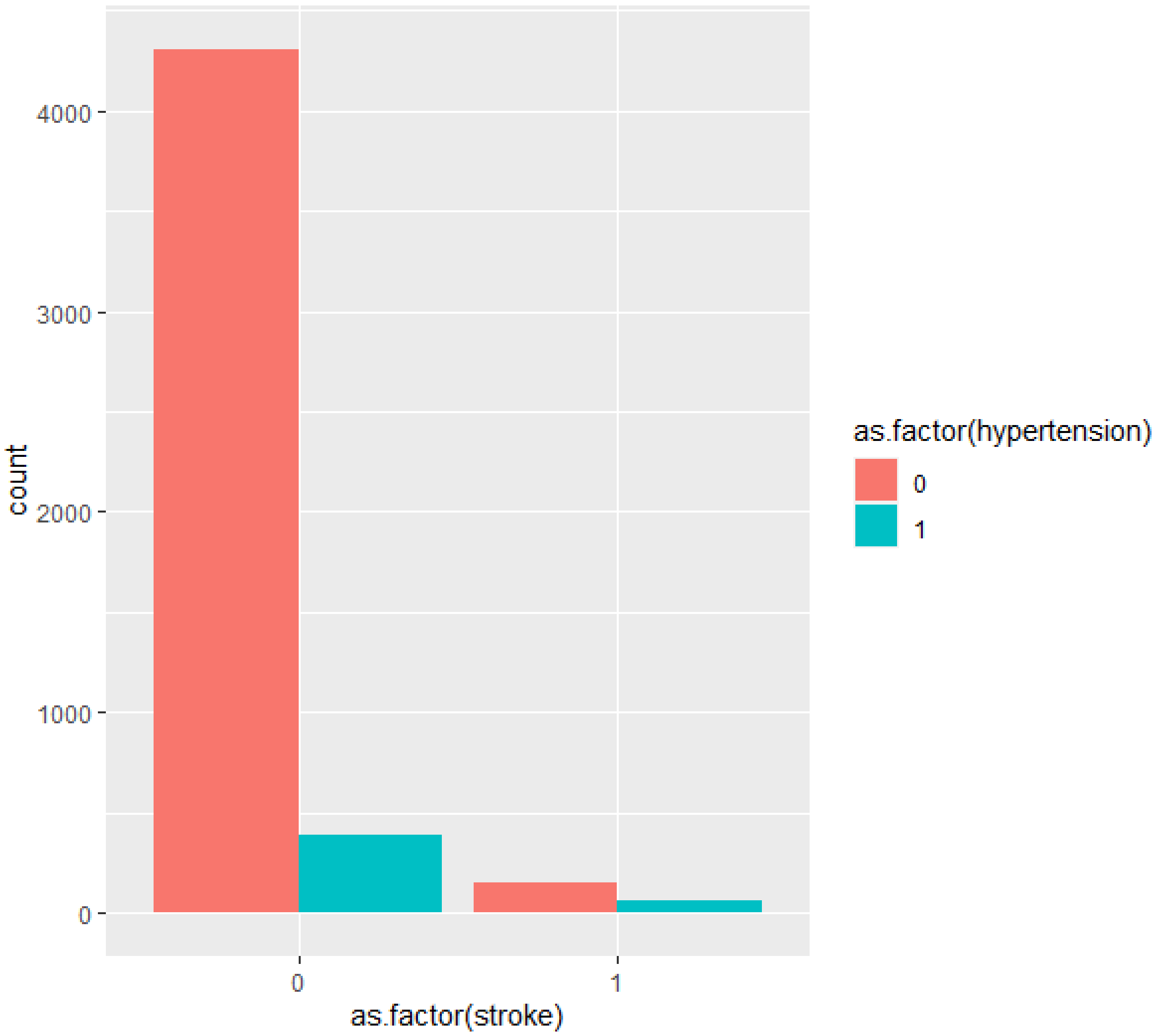
From the summary, we could see that more data has been collected from females than males hence the data for females is more and from the graph, we could see people who never had a stroke and who have had a stroke both the group has female in the maximum and male at minimum. This concludes that even females get strokes.



The graph is plotted for stroke and its frequency filled with every work type from this graph we could infer that there is more data recorded in people without stroke and from that we could see most of them are from the private business sector. People who had a stroke also have private sector jobs or businesses at maximum. from the produced data we could say that private business people are more prone to strokes due to overstress and self-improvement.



The graph is plotted for stroke and its frequency filled with heart disease from this graph we could infer that there is more data recorded in people without stroke and from that we could see most of them are from them do not have any heart. People who had a stroke also do not have any heart disease. from the produced data we could say even people who do not have heart disease are affected by strokes which mean stroke are more complicated to predict with very few factors.



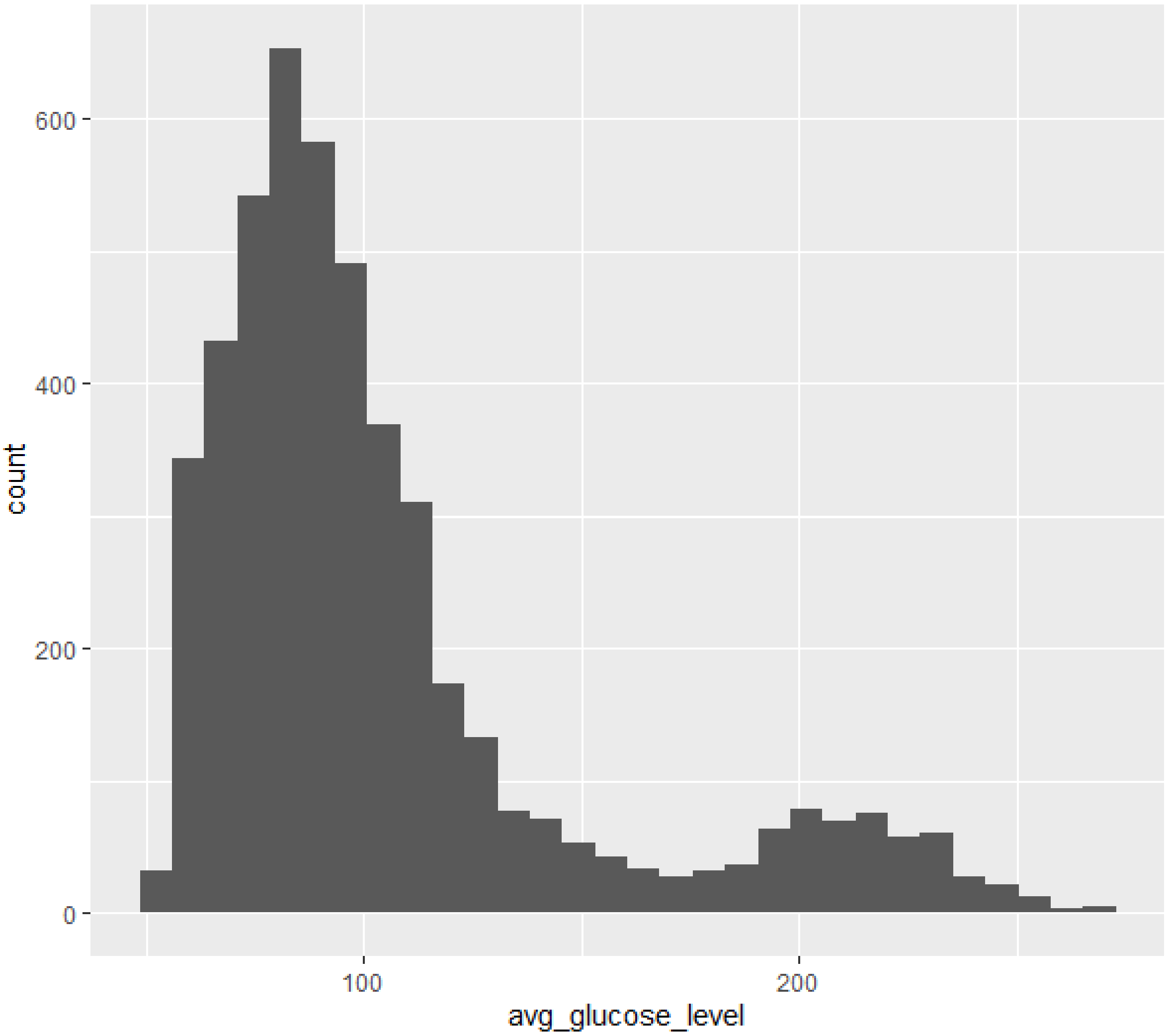
The graph is plotted for stroke and its frequency filled with hypertension from this graph we could infer that there is more data recorded in people without stroke and from that we could see most of them are from them do not have hypertension. People who had a stroke also do not have hypertension. from the produced data we could say that not only people with hyper tension gets strokes also those who are calm gets stroke.

c. DATA CLEANING

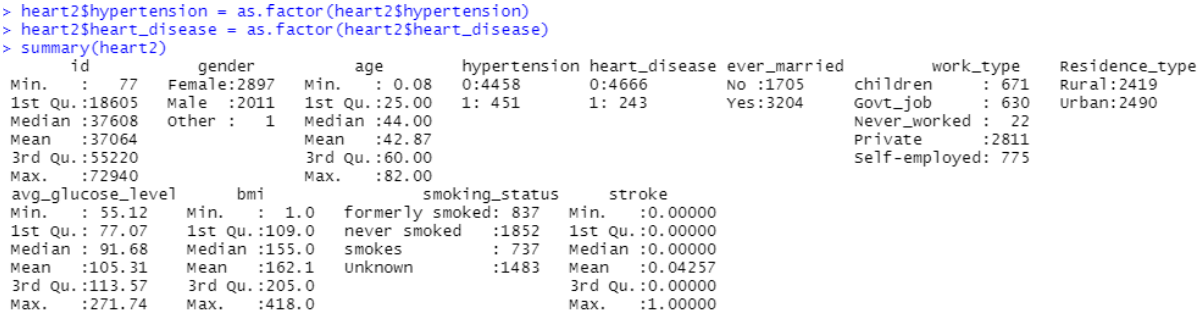
Data cleaning is an important factor to consider from the data selected by using the summary. From the summary, we can see that the bmi has some n/a values but they are read as separate strings, so we must first convert them into a na value and then remove the null values from the data as they are very prone to prediction error.

Once removed the BMI variable started to be read as a categorical variable which gets too much data when converted to a dummy hence changing it to numeric using as. numeric function.

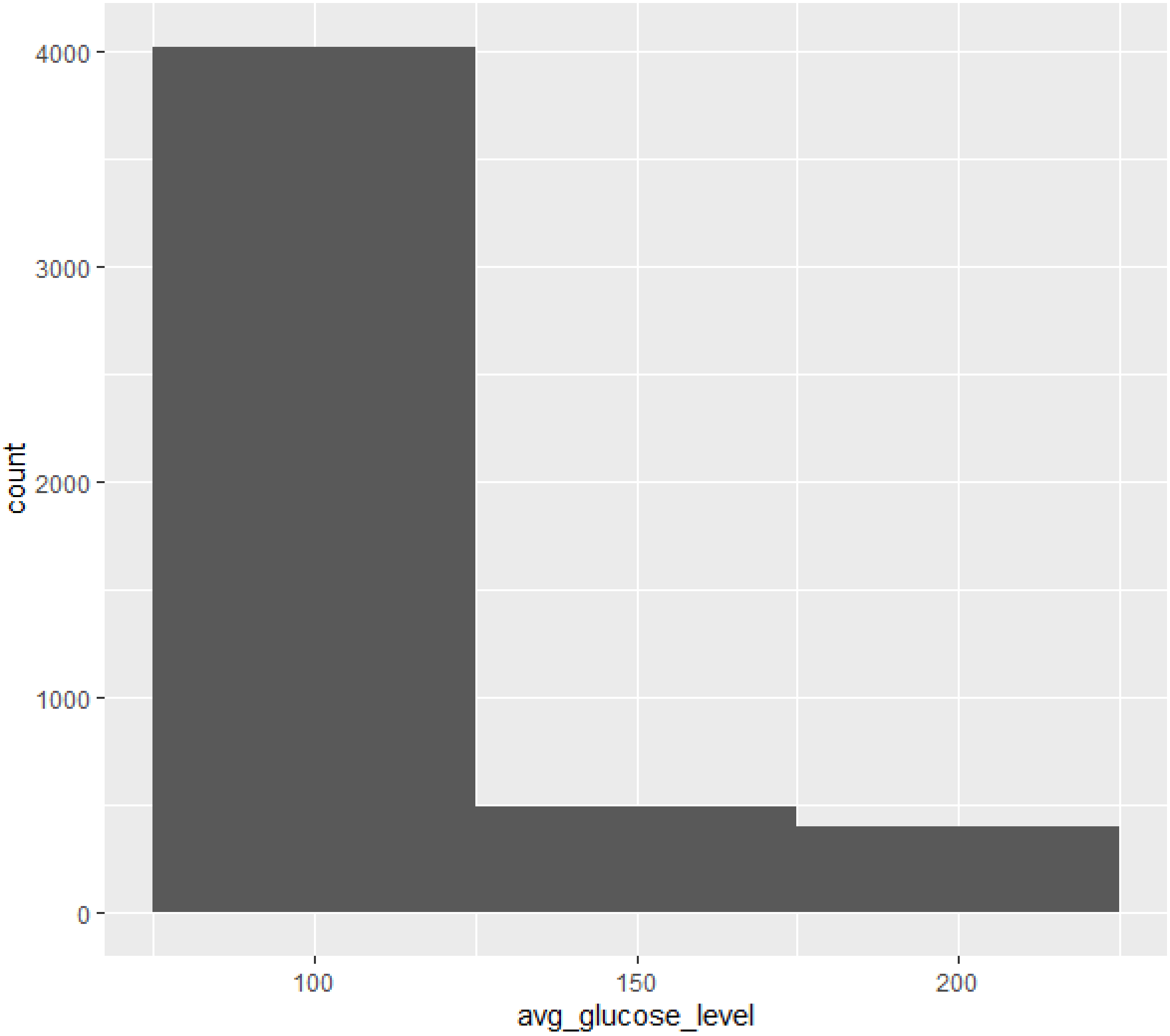
The result of the summary is given in



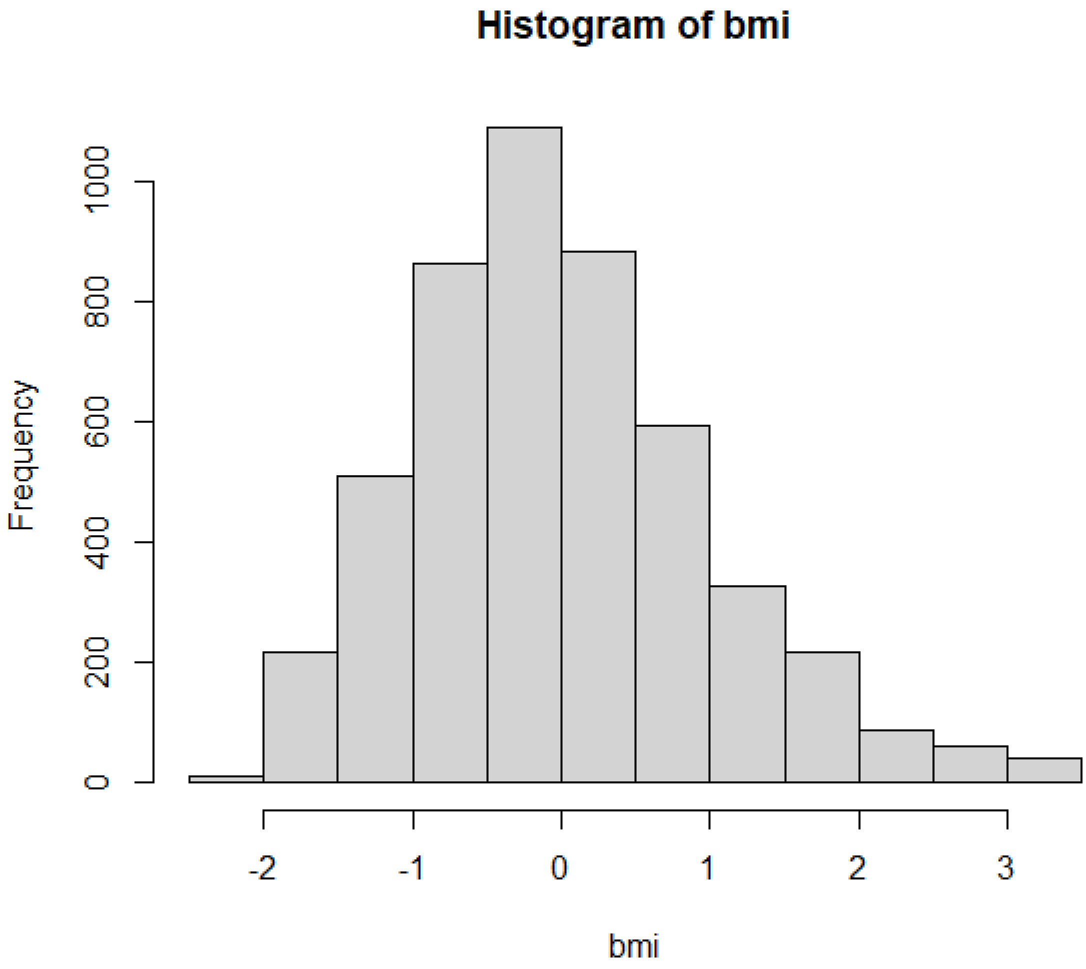
The next step is to bin the avg glucose level data because it is numeric and has a skewed right graph with a lot of unsimilar values and outliers, thus binning and smoothing this variable is a smart alternative. Because the variable is right skewed, I used three breaks binning and smoothing by the median approach. Using the mean method would make it more prone to outlier data. As a result, we employed the median smoothing method.



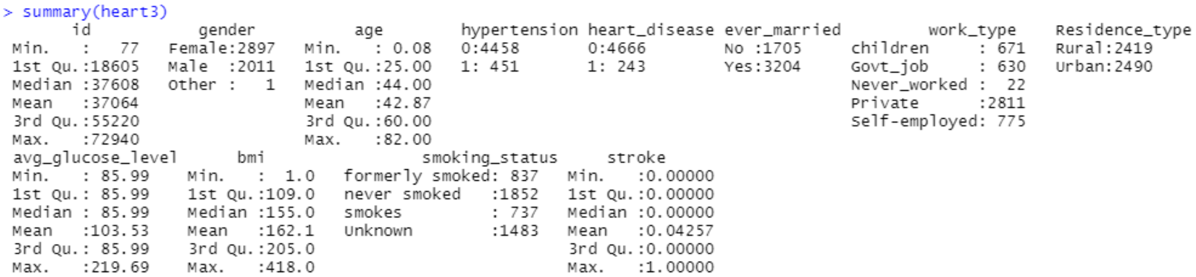
```
> heart2$hypertension = as.factor(heart2$hypertension)
> heart2$heart_disease = as.factor(heart2$heart_disease)
> summary(heart2)
       id           gender          age        hypertension heart_disease ever_married         work_type    Residence_type
 Min.   :   77   Female:2897   Min.   : 0.08   0:4458       0:4666        No :1705     children     : 671   Rural:2419    
 1st Qu.:18605   Male  :2011   1st Qu.:25.00   1: 451       1: 243        Yes:3204     Govt_job     : 630   Urban:2490    
 Median :37608   Other :   1   Median :44.00                                           Never_worked :  22                 
 Mean   :37064                 Mean   :42.87                                           Private      :2811                 
 3rd Qu.:55220                 3rd Qu.:60.00                                           Self-employed: 775                 
 Max.   :72940                 Max.   :82.00                                                                              
 avg_glucose_level      bmi                 smoking_status     stroke       
 Min.   : 55.12    Min.   :  1.0   formerly smoked: 837   Min.   :0.00000  
 1st Qu.: 77.07    1st Qu.:109.0   never smoked   :1852   1st Qu.:0.00000  
 Median : 91.68    Median :155.0   smokes         : 737   Median :0.00000  
 Mean   :105.31    Mean   :162.1   Unknown        :1483   Mean   :0.04257  
 3rd Qu.:113.57    3rd Qu.:205.0                          3rd Qu.:0.00000  
 Max.   :271.74    Max.   :418.0                          Max.   :1.00000  
```



d. DATA PROCESSING



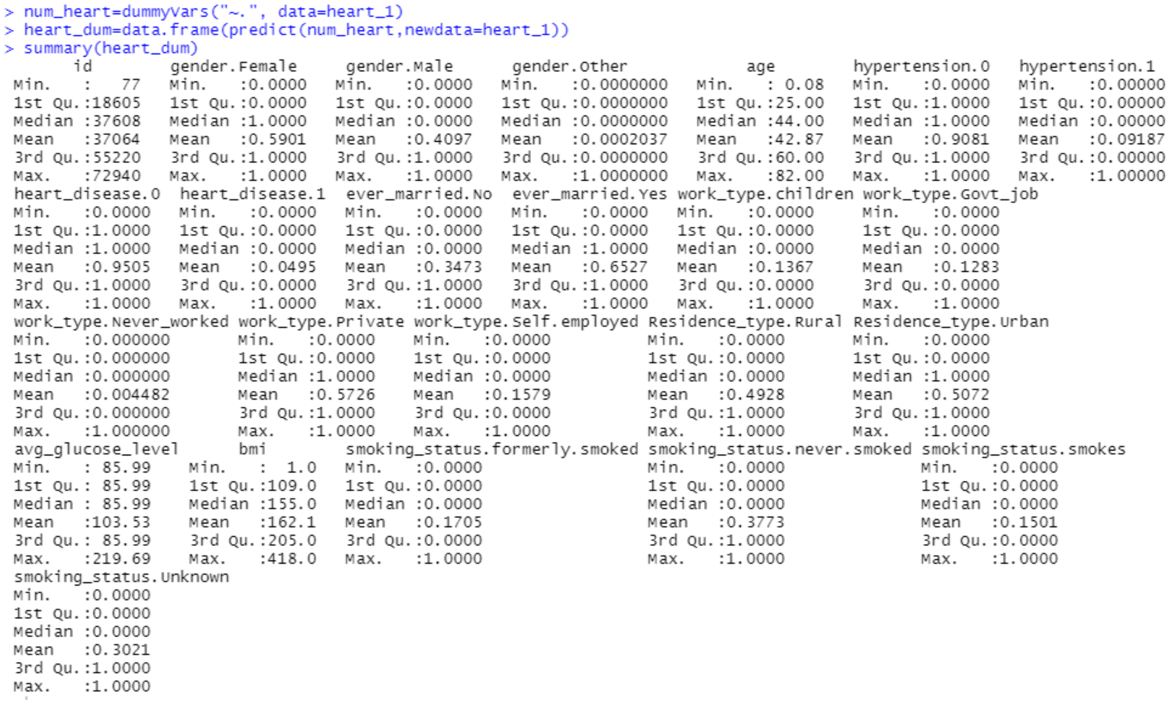
As the BMI values increased, the variable was scaled to a limit between -2 and 3 using the z-scoring technique. As we know, normalising just reduces the data and has no effect on the graph presented, therefore there is no difference when comparing.



```
> summary(heart3)
       id           gender          age        hypertension heart_disease ever_married         work_type    Residence_type
 Min.   :   77   Female:2897   Min.   : 0.08   0:4458       0:4666        No :1705     children     : 671   Rural:2419
 1st Qu.:18605   Male  :2011   1st Qu.:25.00   1: 451       1: 243        Yes:3204     Govt_job     : 630   Urban:2490
 Median :37608   Other :   1   Median :44.00                                           Never_worked :  22
 Mean   :37064                 Mean   :42.87                                           Private      :2811
 3rd Qu.:55220                 3rd Qu.:60.00                                           Self-employed: 775
 Max.   :72940                 Max.   :82.00
 avg_glucose_level      bmi                smoking_status     stroke
 Min.   : 85.99    Min.   :  1.0   formerly smoked: 837   Min.   :0.00000
 1st Qu.: 85.99    1st Qu.:109.0   never smoked   :1852   1st Qu.:0.00000
 Median : 85.99    Median :155.0   smokes         : 737   Median :0.00000
 Mean   :103.53    Mean   :162.1   Unknown        :1483   Mean   :0.04257
 3rd Qu.: 85.99    3rd Qu.:205.0                          3rd Qu.:0.00000
 Max.   :219.69    Max.   :418.0                          Max.   :1.00000
```

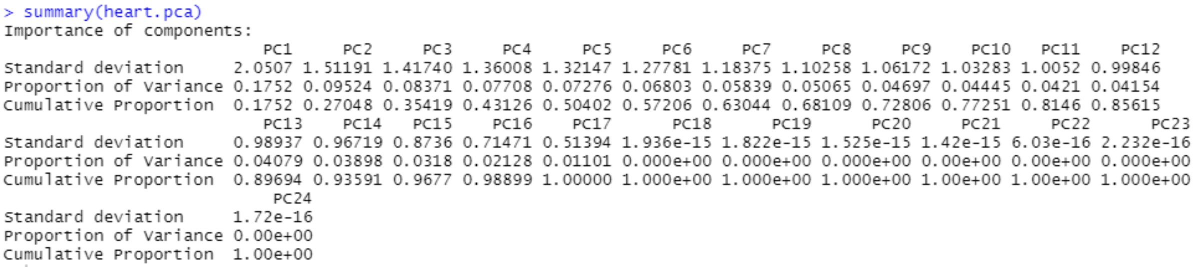
After binning, we can see that there are a number of categorical variables, and for the next step, we require numerical or binary variables, therefore we must convert them to binary. The variables that get converted are gender, ever married, work type, and residence type.

As a bogus variable. We can refer to the summary produced by figure() following the dummy variable conversion.

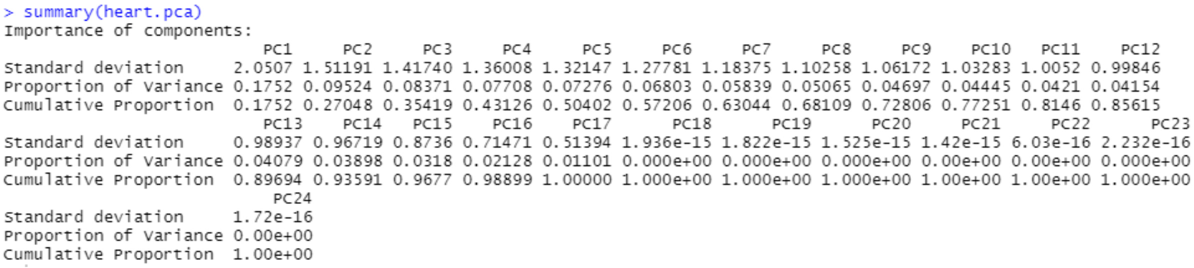


```
> num_heart=dummyVars("~.", data=heart_1)
> heart_dum=data.frame(predict(num_heart,newdata=heart_1))
> summary(heart_dum)
       id        gender.Female     gender.Male      gender.Other            age        hypertension.0   hypertension.1  
 Min.   :   77   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000000   Min.   : 0.08   Min.   :0.0000   Min.   :0.00000  
 1st Qu.:18605   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000000   1st Qu.:25.00   1st Qu.:1.0000   1st Qu.:0.00000  
 Median :37608   Median :1.0000   Median :0.0000   Median :0.0000000   Median :44.00   Median :1.0000   Median :0.00000  
 Mean   :37064   Mean   :0.5901   Mean   :0.4097   Mean   :0.0002037   Mean   :42.87   Mean   :0.9081   Mean   :0.09187  
 3rd Qu.:55220   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000000   3rd Qu.:60.00   3rd Qu.:1.0000   3rd Qu.:0.00000  
 Max.   :72940   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000000   Max.   :82.00   Max.   :1.0000   Max.   :1.00000  
 heart_disease.0  heart_disease.1  ever_married.No  ever_married.Yes work_type.children work_type.Govt_job
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000     Min.   :0.0000    
 1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000     1st Qu.:0.0000    
 Median :1.0000   Median :0.0000   Median :0.0000   Median :1.0000   Median :0.0000     Median :0.0000    
 Mean   :0.9505   Mean   :0.0495   Mean   :0.3473   Mean   :0.6527   Mean   :0.1367     Mean   :0.1283    
 3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000     3rd Qu.:0.0000    
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000     Max.   :1.0000    
 work_type.Never_worked work_type.Private work_type.Self.employed Residence_type.Rural Residence_type.Urban
 Min.   :0.000000       Min.   :0.0000    Min.   :0.0000          Min.   :0.0000       Min.   :0.0000      
 1st Qu.:0.000000       1st Qu.:0.0000    1st Qu.:0.0000          1st Qu.:0.0000       1st Qu.:0.0000      
 Median :0.000000       Median :1.0000    Median :0.0000          Median :0.0000       Median :1.0000      
 Mean   :0.004482       Mean   :0.5726    Mean   :0.1579          Mean   :0.4928       Mean   :0.5072      
 3rd Qu.:0.000000       3rd Qu.:1.0000    3rd Qu.:0.0000          3rd Qu.:1.0000       3rd Qu.:1.0000      
 Max.   :1.000000       Max.   :1.0000    Max.   :1.0000          Max.   :1.0000       Max.   :1.0000      
 avg_glucose_level      bmi         smoking_status.formerly.smoked smoking_status.never.smoked smoking_status.smokes
 Min.   : 85.99    Min.   :  1.0   Min.   :0.0000                 Min.   :0.0000              Min.   :0.0000       
 1st Qu.: 85.99    1st Qu.:109.0   1st Qu.:0.0000                 1st Qu.:0.0000              1st Qu.:0.0000       
 Median : 85.99    Median :155.0   Median :0.0000                 Median :0.0000              Median :0.0000       
 Mean   :103.53    Mean   :162.1   Mean   :0.1705                 Mean   :0.3773              Mean   :0.1501       
 3rd Qu.: 85.99    3rd Qu.:205.0   3rd Qu.:0.0000                 3rd Qu.:1.0000              3rd Qu.:0.0000       
 Max.   :219.69    Max.   :418.0   Max.   :1.0000                 Max.   :1.0000              Max.   :1.0000       
 smoking_status.Unknown
 Min.   :0.0000        
 1st Qu.:0.0000        
 Median :0.0000        
 Mean   :0.3021        
 3rd Qu.:1.0000        
 Max.   :1.0000        
```

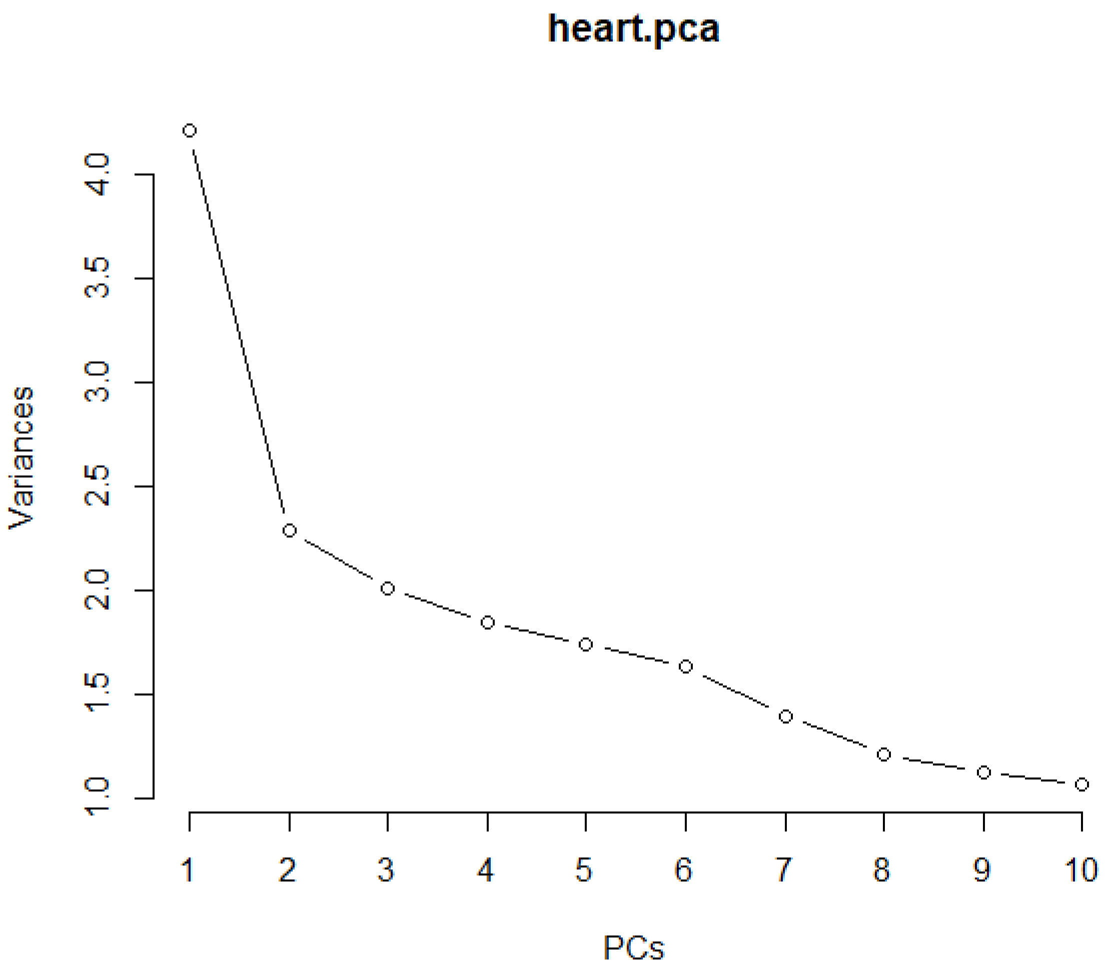
The data has been normalized using the principal component analysis method.



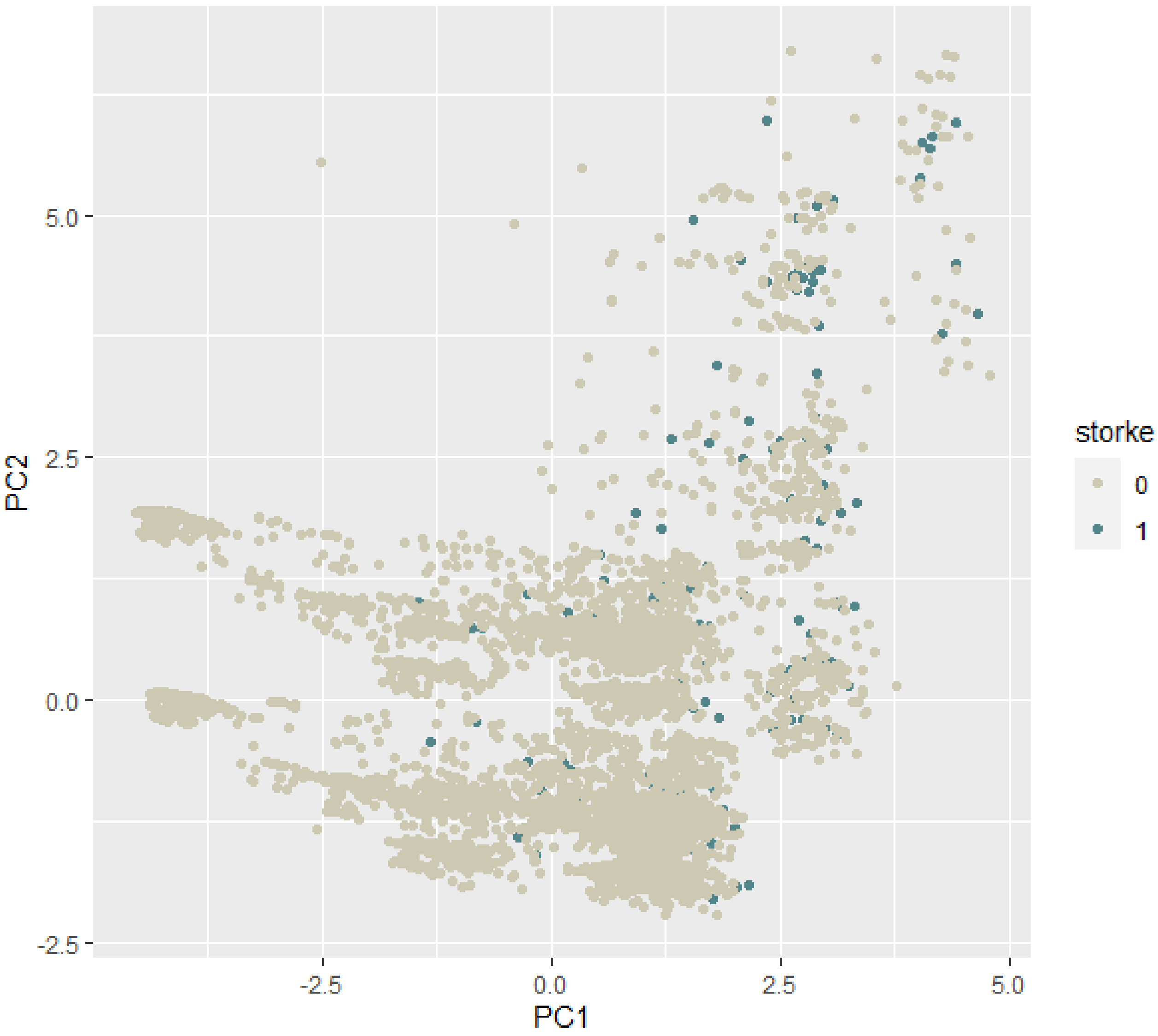
```
> summary(heart.pca)
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10   PC11    PC12
Standard deviation     2.0507 1.51191 1.41740 1.36008 1.32147 1.27781 1.18375 1.10258 1.06172 1.03283 1.0052 0.99846
Proportion of Variance 0.1752 0.09524 0.08371 0.07708 0.07276 0.06803 0.05839 0.05065 0.04697 0.04445 0.0421 0.04154
Cumulative Proportion  0.1752 0.27048 0.35419 0.43126 0.50402 0.57206 0.63044 0.68109 0.72806 0.77251 0.8146 0.85615
                          PC13    PC14   PC15    PC16    PC17      PC18      PC19      PC20     PC21     PC22      PC23
Standard deviation     0.98937 0.96719 0.8736 0.71471 0.51394 1.936e-15 1.822e-15 1.525e-15 1.42e-15 6.03e-16 2.232e-16
Proportion of Variance 0.04079 0.03898 0.0318 0.02128 0.01101 0.000e+00 0.000e+00 0.000e+00 0.00e+00 0.00e+00 0.000e+00
Cumulative Proportion  0.89694 0.93591 0.9677 0.98899 1.00000 1.000e+00 1.000e+00 1.000e+00 1.00e+00 1.00e+00 1.000e+00
                           PC24
Standard deviation     1.72e-16
Proportion of Variance 0.00e+00
Cumulative Proportion  1.00e+00
```



```
> summary(heart.pca)
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10   PC11    PC12
Standard deviation     2.0507 1.51191 1.41740 1.36008 1.32147 1.27781 1.18375 1.10258 1.06172 1.03283 1.0052 0.99846
Proportion of Variance 0.1752 0.09524 0.08371 0.07708 0.07276 0.06803 0.05839 0.05065 0.04697 0.04445 0.0421 0.04154
Cumulative Proportion  0.1752 0.27048 0.35419 0.43126 0.50402 0.57206 0.63044 0.68109 0.72806 0.77251 0.8146 0.85615
                          PC13    PC14   PC15    PC16    PC17      PC18      PC19      PC20     PC21     PC22      PC23
Standard deviation     0.98937 0.96719 0.8736 0.71471 0.51394 1.936e-15 1.822e-15 1.525e-15 1.42e-15 6.03e-16 2.232e-16
Proportion of Variance 0.04079 0.03898 0.0318 0.02128 0.01101 0.000e+00 0.000e+00 0.000e+00 0.00e+00 0.00e+00 0.000e+00
Cumulative Proportion  0.89694 0.93591 0.9677 0.98899 1.00000 1.000e+00 1.000e+00 1.000e+00 1.00e+00 1.00e+00 1.000e+00
                           PC24
Standard deviation     1.72e-16
Proportion of Variance 0.00e+00
Cumulative Proportion  1.00e+00
```

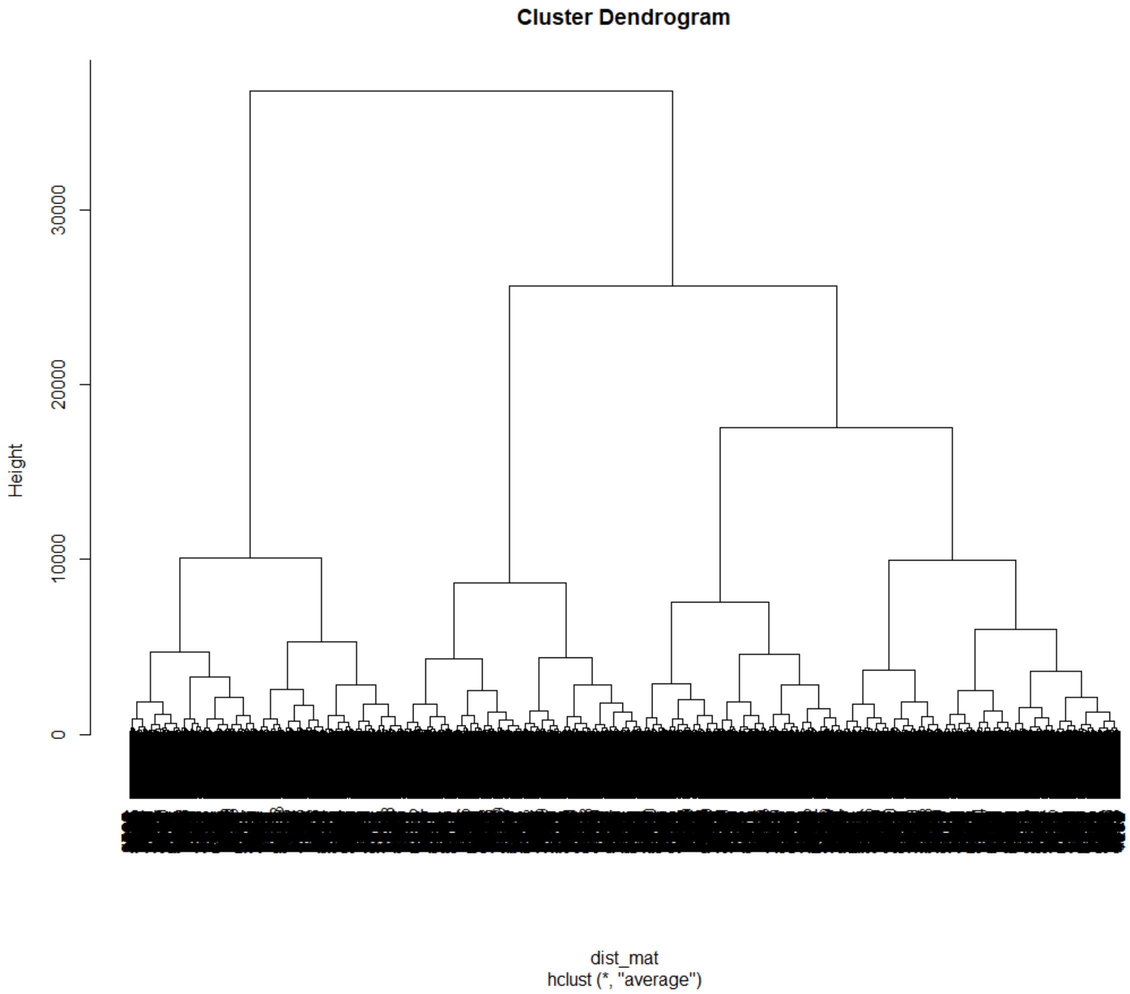


The data has been subjected to principal component analysis. We can see from this that the PCA has been processed, and a summary of it is shown in the picture above. Because PCA decreases the dimensionality of the data set and assigns the predictions in the most significant order, we may conclude that PC1 and PC2 are the most significant variables based on the screen plot.



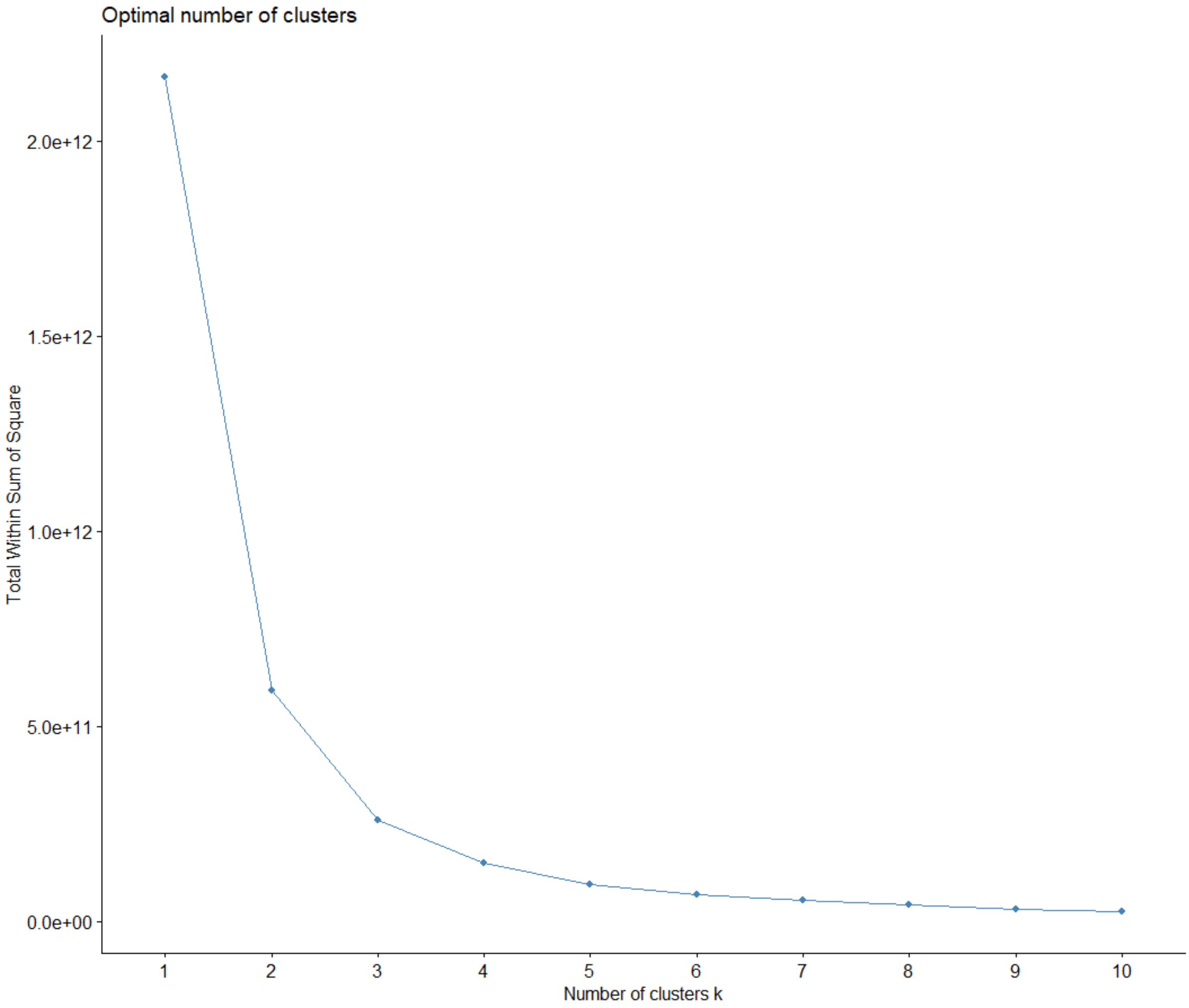
The following graphic illustrates how the data has been spread on a 2-dimensional plane; hence, we can observe that stroke 0 has more data points than stroke 1, and stroke one is intermingled in between stroke 0, which makes classification difficult. Because the prediction would be incorrect, the accuracy would suffer.

e) CLUSTERING



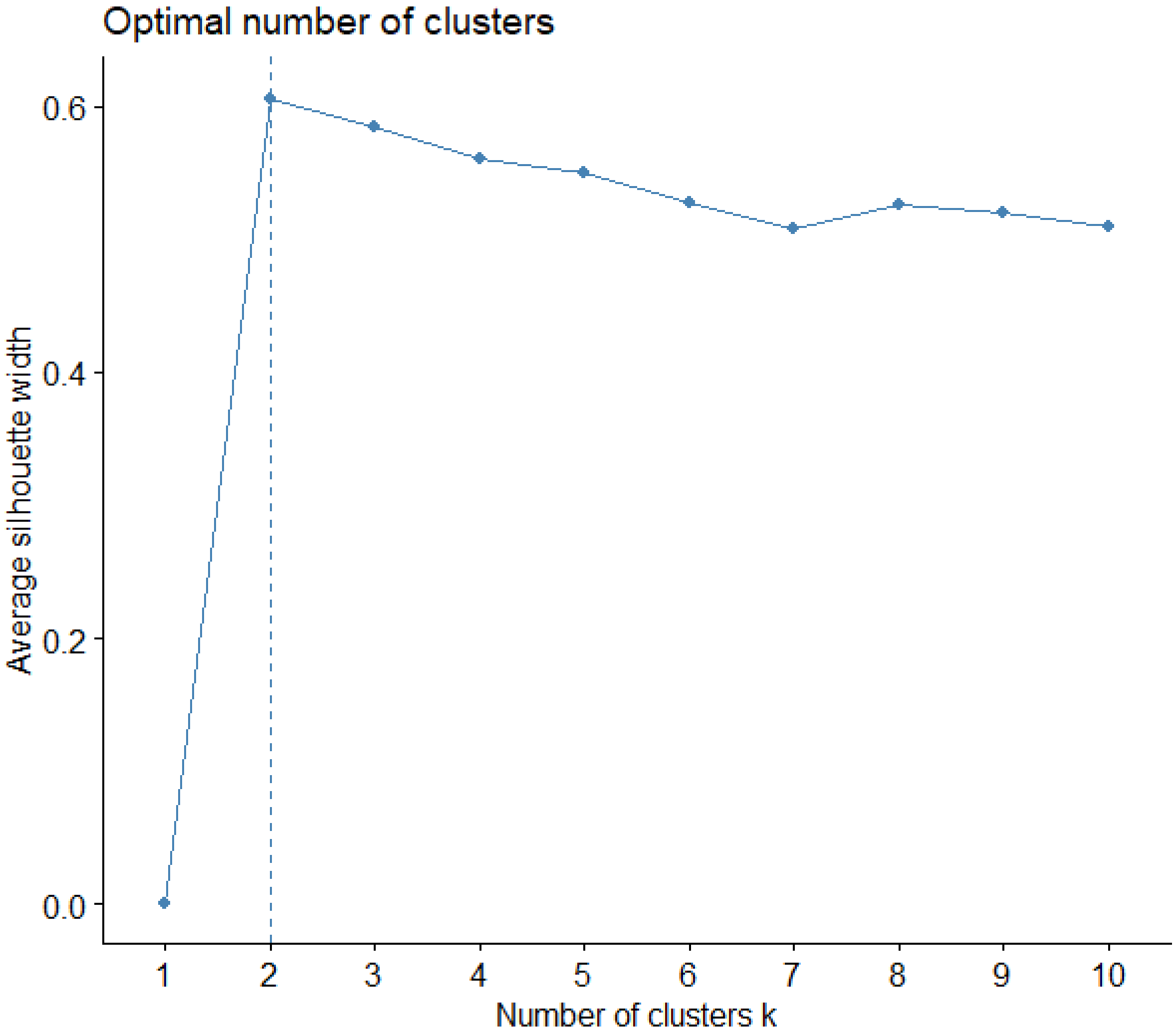
We are utilising the HIERARCHICAL Approach for clustering because the majority of the data points are on the stroke 0 side and the hierarchical method finds the substructure inside the cluster, thus we are using the hierarchical method.

We need to find a distance matrix for this procedure; thus we're using the euclidean method of distance matrix to acquire more precise values. And the method employs average linkage since the data points are distributed in a clustered pattern.

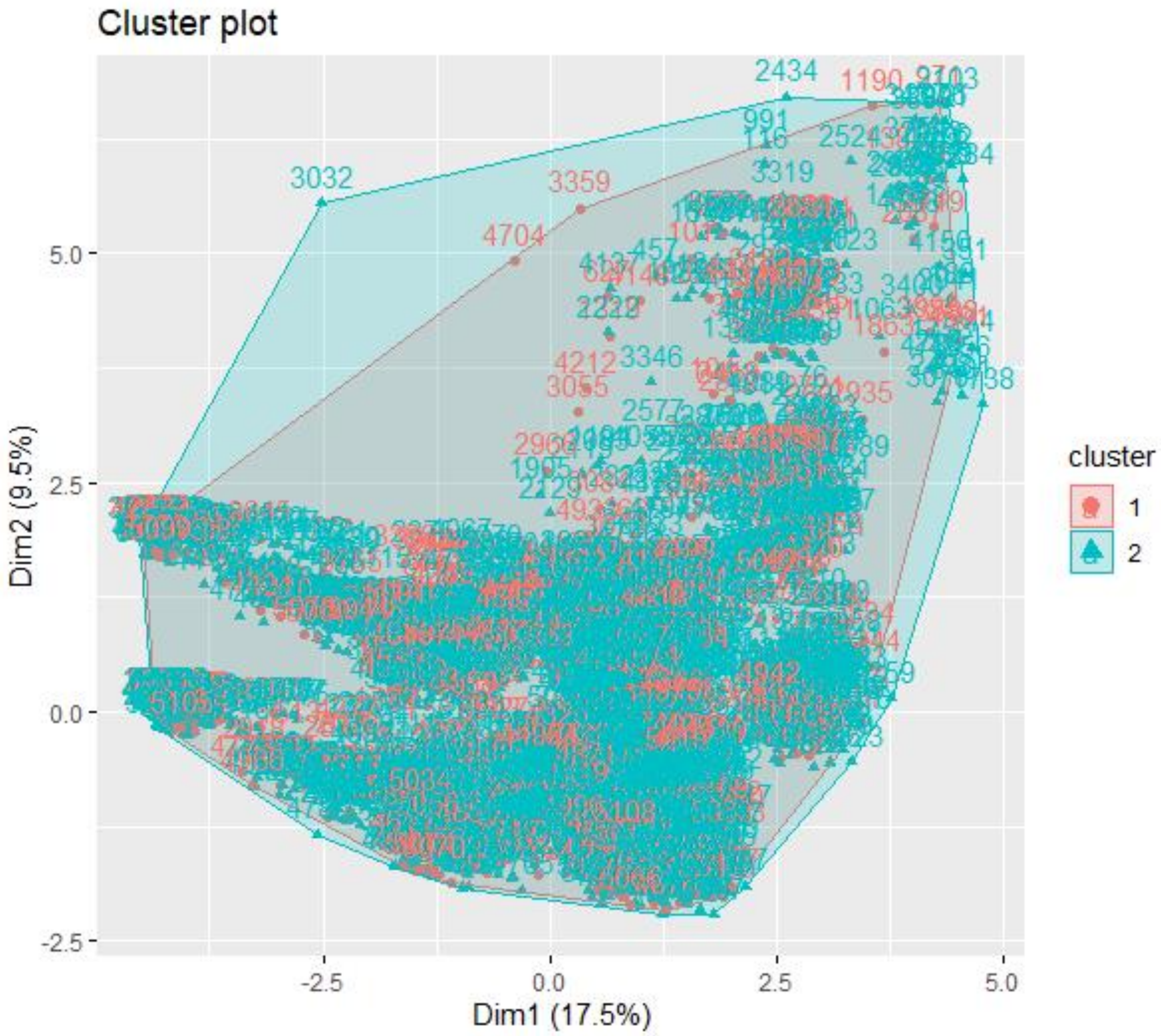


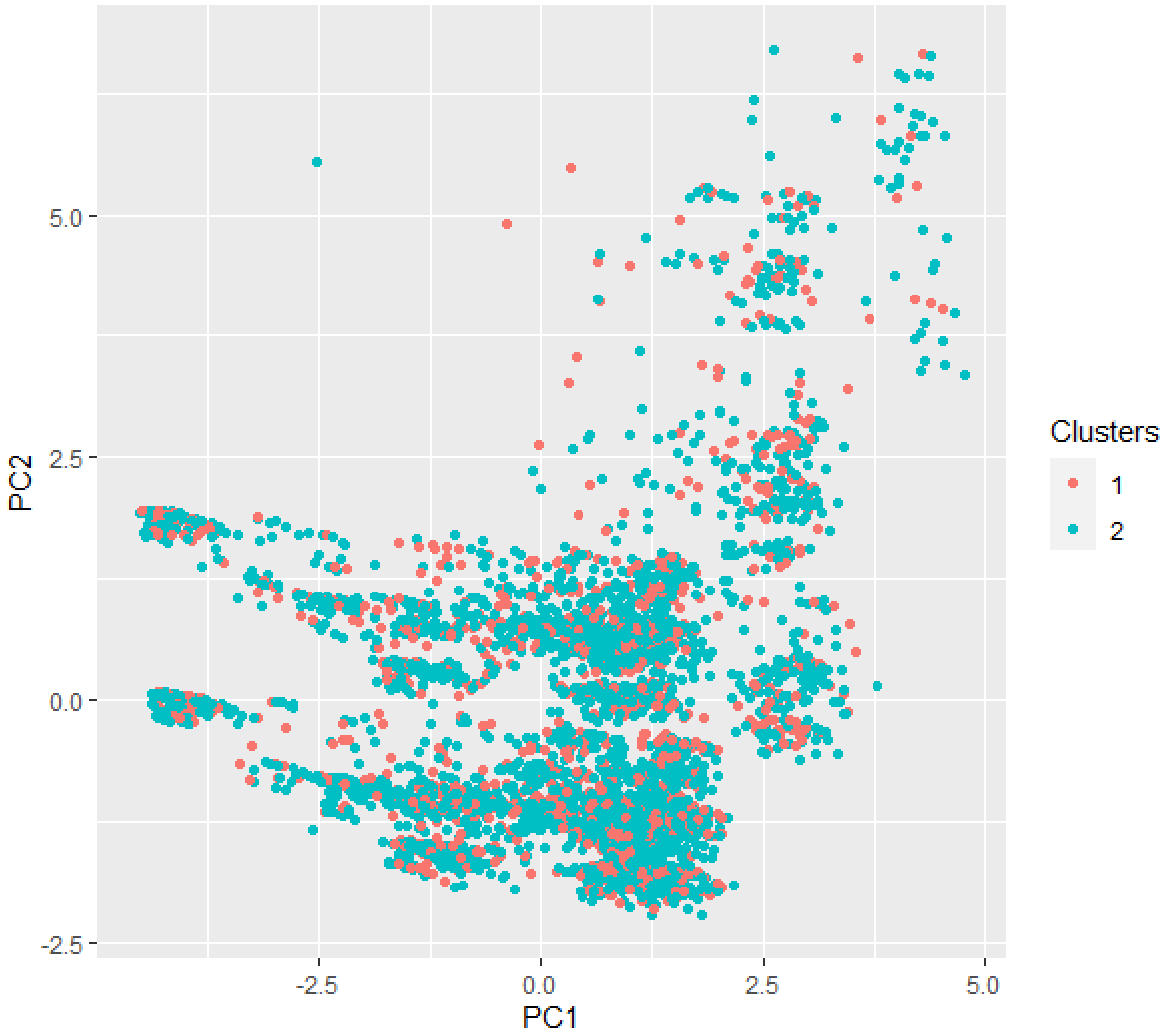
The best amount of clusters A graph has been created to determine the appropriate k for clustering. We can tell from the graph that the elbow point was recorded in point three. We can also tell from the graph that three clusters are good for the data set.

However, for more precision, we use a "silhouette" graph.



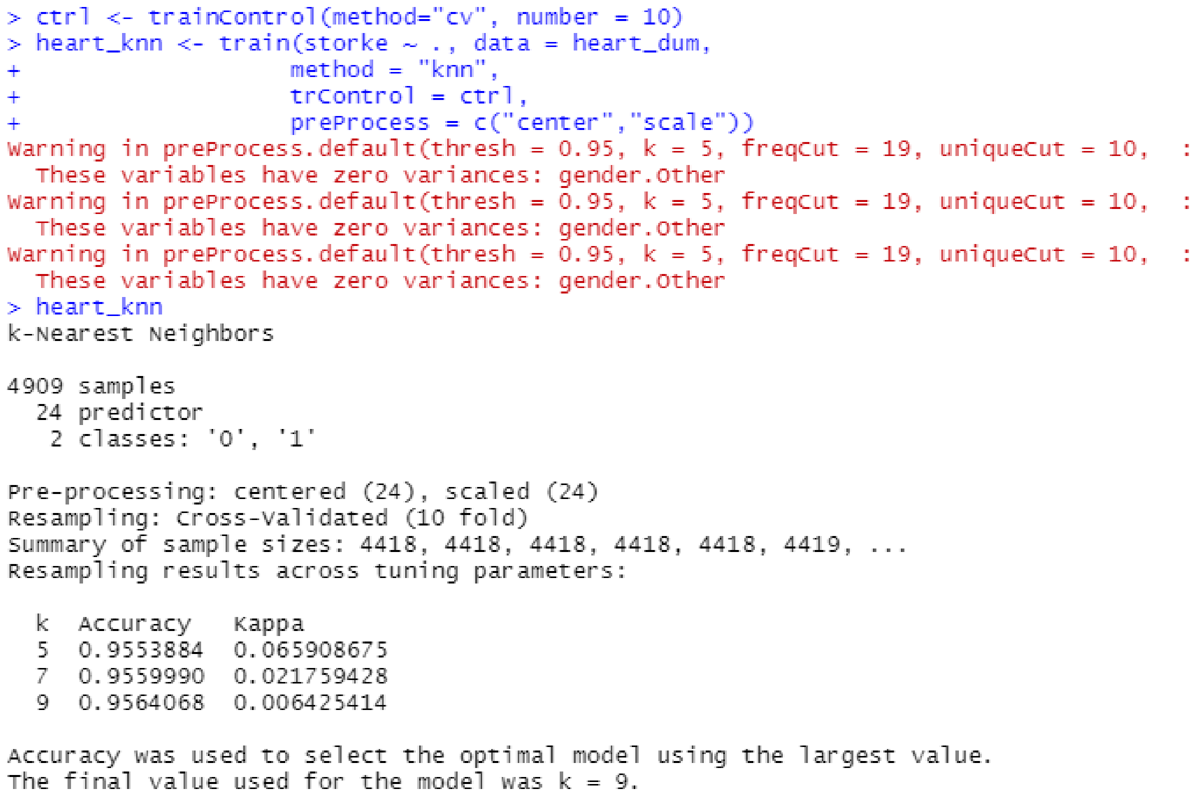
This is the "silhouette," and we can see the suggestion of clustering k = 2, therefore we chose two clusters for the data set.


Cluster plot
Dim2 (9.5%)
Dim1 (17.5%)
5.0
2.5
0.0
-2.5
-2.5
0.0
2.5
5.0
cluster
1
2
2434
1190
991
116
2524
3319
3032
3359
4704
4212
3346
3055
2577
2964
1905
2935
1863
5105
4066
5034
4942
4159
3400
138

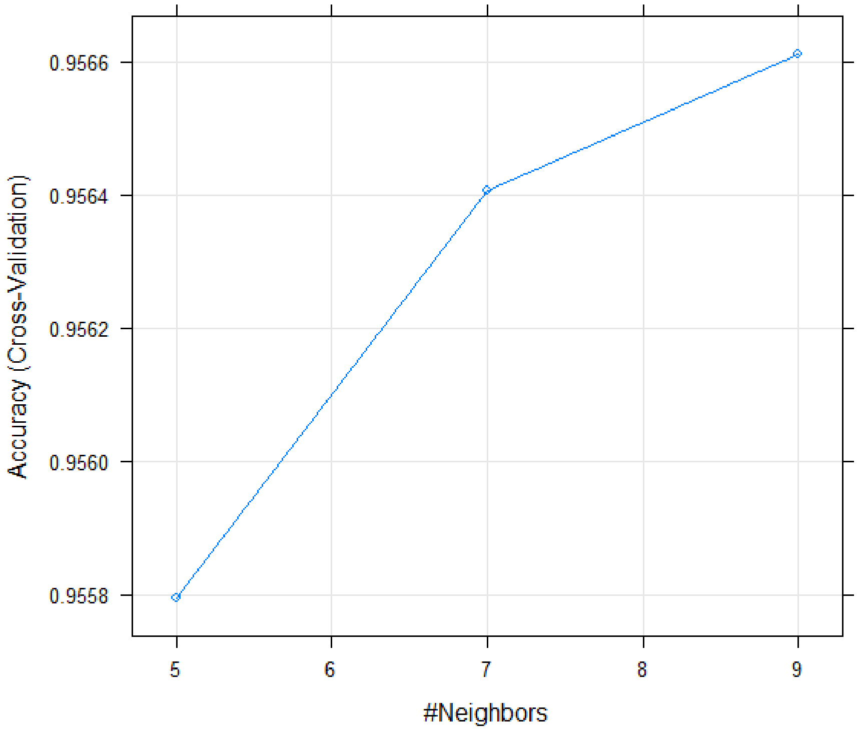


The data shown above is after clustering with k=2. As a result, we have two clusters, and we can observe that the clusters have collided with each other because the data is more complex and has two clusters variable.

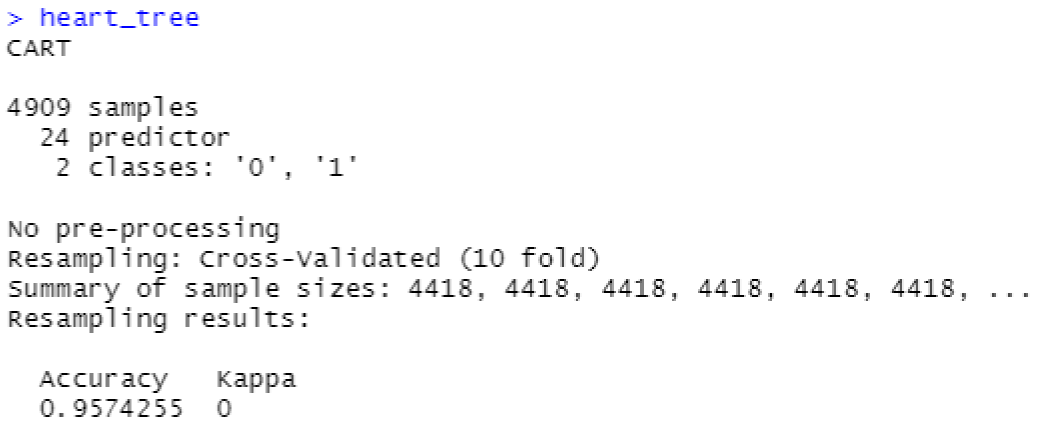
f) CLASSIFICATION



```
> ctrl <- trainControl(method="cv", number = 10)
> heart_knn <- train(storke ~ ., data = heart_dum,
+                    method = "knn",
+                    trControl = ctrl,
+                    preProcess = c("center","scale"))
Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut = 10,  :
  These variables have zero variances: gender.Other
Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut = 10,  :
  These variables have zero variances: gender.Other
Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut = 10,  :
  These variables have zero variances: gender.Other
> heart_knn
k-Nearest Neighbors 

4909 samples
  24 predictor
   2 classes: '0', '1' 

Pre-processing: centered (24), scaled (24) 
Resampling: Cross-Validated (10 fold) 
Summary of sample sizes: 4418, 4418, 4418, 4418, 4418, 4419, ... 
Resampling results across tuning parameters:

  k  Accuracy   Kappa      
  5  0.9553884  0.065908675
  7  0.9559990  0.021759428
  9  0.9564068  0.006425414

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 9.
```



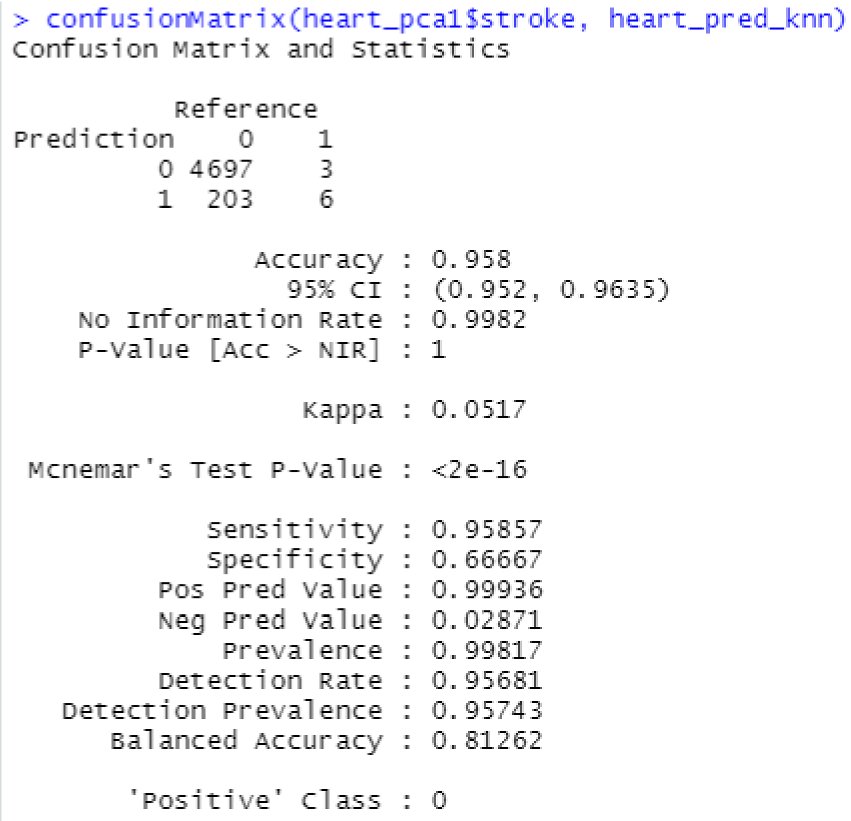
First, I used the k-nearest neighbour clustering approach, and the accuracy is 0.958 using a confusion matrix, it is thought to be a better mode.



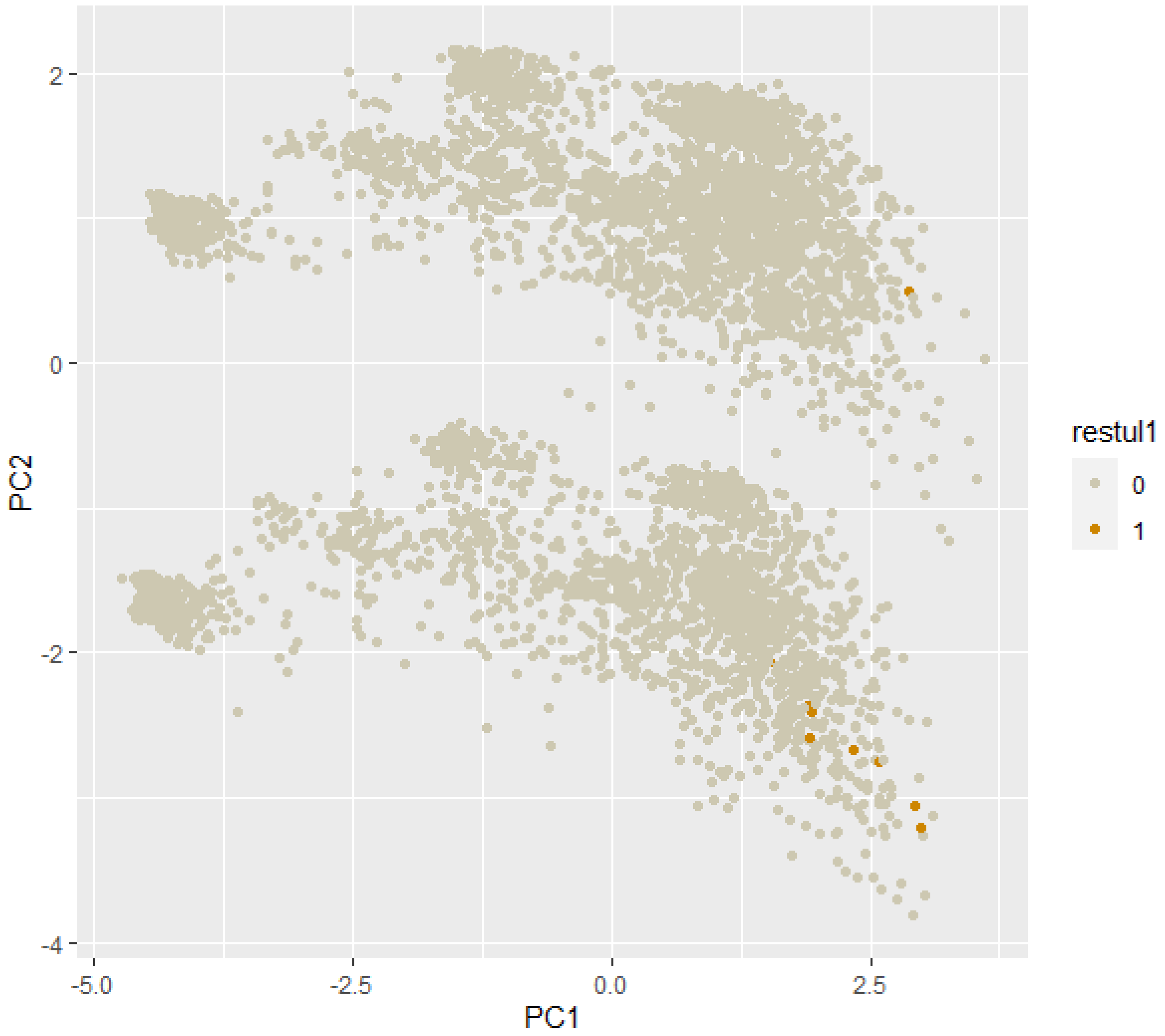
```
> heart_tree
CART

4909 samples
  24 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 4418, 4418, 4418, 4418, 4418, 4418, ...
Resampling results:

  Accuracy   Kappa
  0.9574255  0
```

The decision tree was the second classification I tried because, as we know, SVM is for linear classification, but we could tell from the clusters that the values could not be separated in a linear method, so we went with the decision tree, and we could see its accuracy to be 0.957, which is good, but the KNN has better prediction than the decision tree.

For the future process, we are going with KNN.



```
> confusionMatrix(heart_pca1$stroke, heart_pred_knn)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4697    3
         1  203    6

               Accuracy : 0.958
                 95% CI : (0.952, 0.9635)
    No Information Rate : 0.9982
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0517

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.95857
            Specificity : 0.66667
         Pos Pred Value : 0.99936
         Neg Pred Value : 0.02871
             Prevalence : 0.99817
         Detection Rate : 0.95681
   Detection Prevalence : 0.95743
      Balanced Accuracy : 0.81262

       'Positive' Class : 0
```

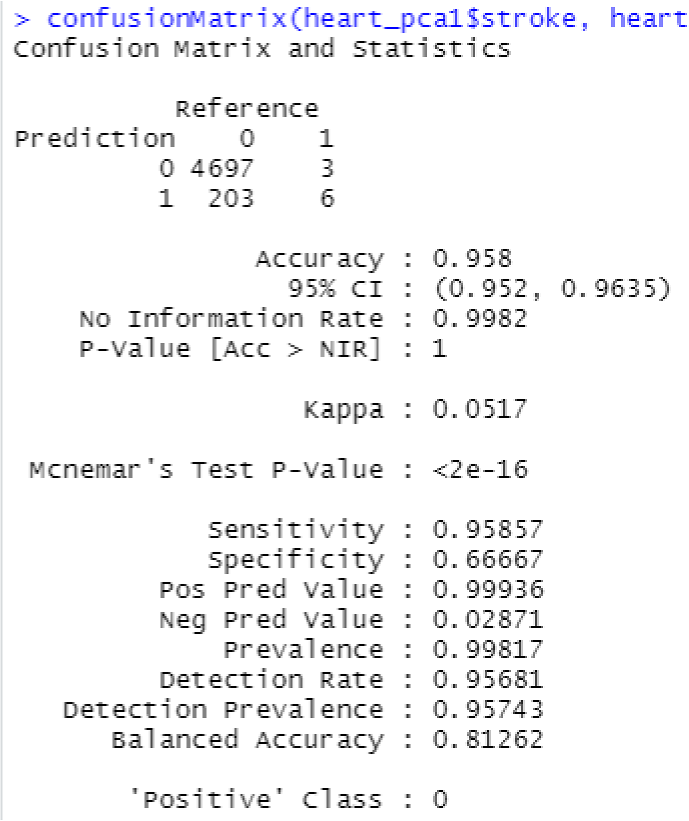


We can observe how the knn classification algorithm marks the points in the scatter plot above. Based on this, we may conclude that the model predicts 0 strokes better than stroke 1, implying that additional work and data collection is required for the people who had strokes in order to create a proper model.

g. EVALUATION

Selecting knn

(1) produce a 2x2 confusion matrix (if your dataset has more than two classes, bin the classes into two groups and rebuild the model),



```
> confusionMatrix(heart_pca1$stroke, heart_pred_knn)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4697    3
         1  203    6

               Accuracy : 0.958
                 95% CI : (0.952, 0.9635)
    No Information Rate : 0.9982
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0517

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.95857
            Specificity : 0.66667
         Pos Pred Value : 0.99936
         Neg Pred Value : 0.02871
             Prevalence : 0.99817
         Detection Rate : 0.95681
   Detection Prevalence : 0.95743
      Balanced Accuracy : 0.81262

       'Positive' Class : 0
```

(2) calculate the precision and recall manually, and finally

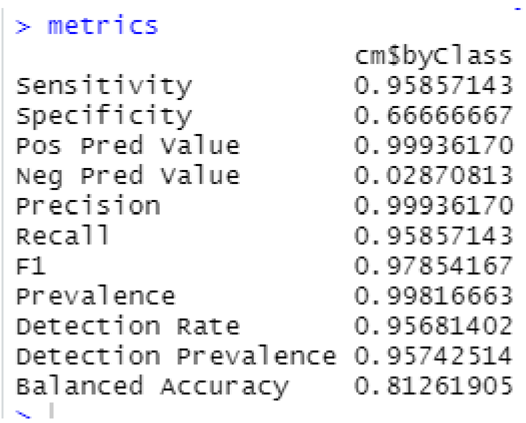
Confusion matrix for knn prediction.

To calculate precision using the formula → TP/TP+FP

Which is equals to 4697/4697+203 = 0.9585

To calculate recall using the formula → TP/TP+FN

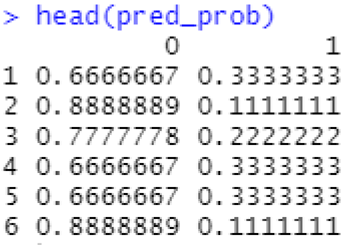
Which is equals to 4697/4697+6 = 0.9987



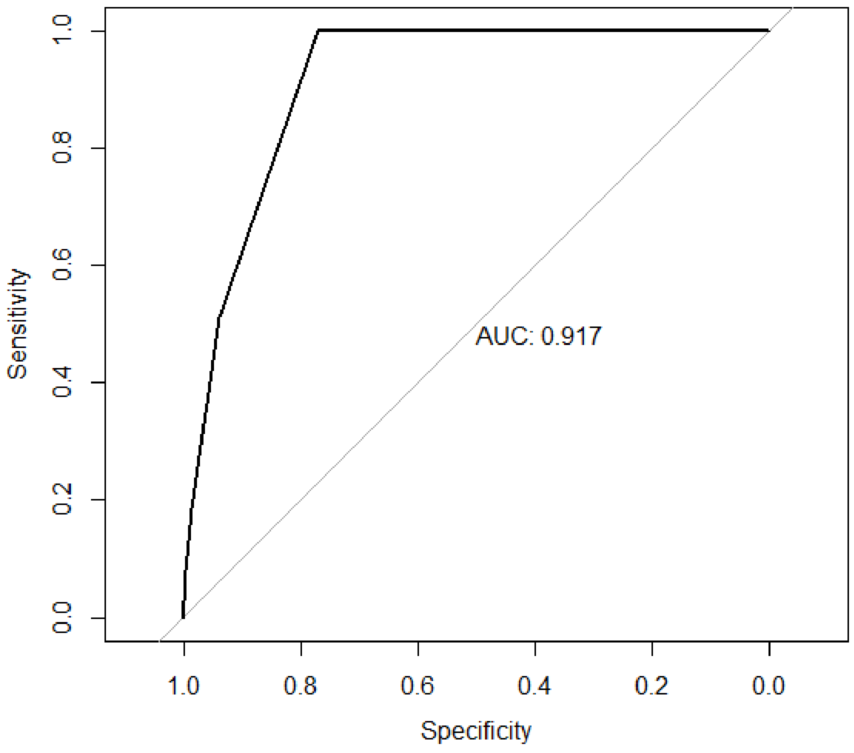
```
> metrics
                     cm$byClass
Sensitivity          0.95857143
Specificity          0.66666667
Pos Pred Value       0.99936170
Neg Pred Value       0.02870813
Precision            0.99936170
Recall               0.95857143
F1                   0.97854167
Prevalence           0.99816663
Detection Rate       0.95681402
Detection Prevalence 0.95742514
Balanced Accuracy    0.81261905
```

From the above figure, we could see more performance analysis of the produced model.

3) produce an ROC plot (see Tutorial 9).



```
> head(pred_prob)
          0         1
1 0.6666667 0.3333333
2 0.8888889 0.1111111
3 0.7777778 0.2222222
4 0.6666667 0.3333333
5 0.6666667 0.3333333
6 0.8888889 0.1111111
```



REFLECTION:

The lectures helped me better understand the fundamentals of data science concepts such as datasets, data preprocessing, and various ways for cleaning and addressing missing values using a variety of techniques like as binning, smoothing, normalising, and many more. Furthermore, Machine learning with SVM, Decision Tree, and KNN parameters can be used to predict the values of labels with known values. Furthermore, by learning to use "clustering" and models such as k-means and other approaches, one can predict the values of unknown labels. I learned advance evaluation to deal with prejudice and class imbalance. Furthermore, knowing a model's error rate and employing accuracy, recall, and ROC. The assignments provided me with additional hands-on experience working with various datasets.

CODE:

```
heart <- read.table("healthcare-dataset-stroke-data.csv",header = T,sep = ",",stringsAsFactors = T)
summary(heart)
#removing na values

heart[heart == 'N/A'] <- NA
heart2 = na.omit(heart)
which(is.na(heart))
heart2$bmi = as.numeric(heart2$bmi)
summary(heart2)

#visualization

ggplot(data = heart2, aes(x = bmi)) +
  geom_histogram(binwidth = 30)

ggplot(heart2, aes(x=as.factor(stroke), y=age)) +
  geom_boxplot()

ggplot(heart2, aes(x=as.factor(stroke), fill=smoking_status))+
  geom_bar(position = position_dodge())

ggplot(heart2, aes(x=as.factor(stroke), fill=ever_married))+
  geom_bar(position = position_dodge())

ggplot(heart2, aes(x=as.factor(stroke), fill=Residence_type))+
  geom_bar(position = position_dodge())

ggplot(heart2, aes(x=as.factor(stroke), y=avg_glucose_level)) +
  geom_boxplot(fill='steelblue')
```

```
ggplot(heart2, aes(x=as.factor(stroke), fill=gender))+
  geom_bar(position = position_dodge())

ggplot(heart2, aes(x=as.factor(stroke), fill=work_type))+
  geom_bar(position = position_dodge())

ggplot(heart2, aes(x=as.factor(stroke), fill=as.factor(hypertension )))+
  geom_bar(position = position_dodge())

ggplot(heart2, aes(x=as.factor(stroke), fill=as.factor(heart_disease)))+
  geom_bar(position = position_dodge())

ggplot(heart2, aes(x = as.factor(stroke)))+
  geom_bar()

summary(heart2)
colnames(heart2)

plot(heart2$age,heart2$stroke)




#processing
#na values
#outliers
heart2$bmi = as.numeric(heart2$bmi)
heart2$hypertension = as.factor(heart2$hypertension)
heart2$heart_disease = as.factor(heart2$heart_disease)

ggplot(data = heart2, aes(x = avg_glucose_level)) +
  geom_histogram()


bmi <- scale(heart2$bmi)
summary(bmi)
hist(bmi)



# breaks = 3 gives us 3 equal width bins
heart3 <- heart2 %>%
  mutate(glucose_level_factor = cut(avg_glucose_level, breaks = 3,
                      labels=c("low","medium","high")))
head(heart3)

# Mutate and store each
low <- heart3 %>%
  filter(glucose_level_factor == 'low') %>%
  mutate(avg_glucose_level = median(avg_glucose_level, na.rm = T))
medium <- heart3 %>%
  filter(glucose_level_factor == 'medium') %>%
  mutate(avg_glucose_level = median(avg_glucose_level, na.rm = T))
high <- heart3 %>%
  filter(glucose_level_factor == 'high') %>%
  mutate(avg_glucose_level = median(avg_glucose_level, na.rm = T))

# The resulting set for each pipeline is immutable and therefore need to be concatenated
# Tidyverse has a bind_rows function that helps us combine these separate sets
heart_copy <- bind_rows(list(low, medium, high))

summary(heart3)
summary(heart_copy)
summary(heart2)

remove(heart_copy)
ggplot(data = heart2, aes(x = avg_glucose_level)) +
  geom_histogram(binwidth = 50)

ggplot(data = heart_copy, aes(x = avg_glucose_level)) +
  geom_histogram()
head(heart_copy)

heart3$avg_glucose_level = heart_copy$avg_glucose_level
heart3<-heart2[,c(-13)]
view(heart_copy)
summary(heart3)


#connverting dummy variables
heart_1 = heart3[,(-12)]
num_heart=dummyVars("~.", data=heart_1)
heart_dum=data.frame(predict(num_heart,newdata=heart_1))

summary(heart_dum)

#pca

heart.pca <- prcomp(heart_dum,center = T,scale. = T)
summary(heart.pca)

screeplot(heart.pca, type = "l") + title(xlab = "PCs")

heart_pca1 = as.data.frame(heart.pca$x)

heart_data1 = as.data.frame(heart.pca$x)
```

```
heart_data1$storke <- as.factor(heart2$stroke)

#visualization of data
ggplot(data = heart_data1, aes(x = PC1, y = PC2, col = storke)) + geom_point()+
  scale_color_manual(values=c('cornsilk3','cadetblue4'))


preproc <- preProcess(heart_dum, method=c("center", "scale"))
heart1 <- predict(preproc, heart_dum)


#HAC
dist_mat <- dist(heart_dum, method = 'manhattan')

hfit <- hclust(dist_mat, method = 'average')
plot(hfit)
fviz_nbclust(heart_dum, FUN = hcut, method = "wss")
fviz_nbclust(heart_dum, FUN = hcut, method = "silhouette")

h3 <- cutree(hfit, k=2)
fviz_cluster(list(data = heart_dum, cluster = h3))

heart_data1$Clusters = as.factor(h3)
# Plot and color by labels
ggplot(data = heart_data1, aes(x = PC1, y = PC2, col = Clusters)) + geom_point()


#kmeans
fviz_nbclust(heart1, kmeans, method = "wss")

fviz_nbclust(heart1, kmeans, method = "silhouette")

# Fit the data
fit_kmeans <- kmeans(heart1, centers = 10, nstart = 25)
# Display the kmeans object information
fit_kmean
fviz_cluster(fit_kmeans, data = heart1)
view(heart1)
#classification

heart_dum$storke = as.factor(heart2$stroke)
summary(heart_dum)

heart_pca1$stroke = as.factor(heart2$stroke)


ctrl <- trainControl(method="cv", number = 10)
heart_knn <- train(storke ~ ., data = heart_dum,
                method = "knn",
                trControl = ctrl,
                preProcess = c("center","scale"))
#Output of kNN fit


heart_knn <- train(stroke ~ ., data = heart_pca1,
                 method = "knn",
                 trControl = ctrl,
                 preProcess = c("center","scale"))


heart_pred_knn <- predict(heart_knn,heart_pca1)
cm = confusionMatrix(heart_pca1$stroke, heart_pred_knn)
heart_knn

#visualization

heart_data1$restul1 <- heart_pred_knn
ggplot(heart_data1,aes(x=PC1,y=PC2,group=restul1))+
  geom_point(aes(color=restul1))+
  scale_color_manual(values=c('cornsilk3','orange3'))





#decision tree
hypers = rpart.control(minsplit = 5000, maxdepth = 4, minbucket = 2500)
heart_tree <- train(stroke ~ ., data = heart_pca1, method = "rpart1SE",control = hypers, trControl =
ctrl)
heart_pred_tree <- predict(heart_tree,heart_pca1)
confusionMatrix(heart_pca1$stroke, heart_pred_tree)


view(heart_dum)
summary(heart_dum)
colnames(heart_dum)


library(caret)
library(rpart)
library(tidyverse)
library(rattle)
library(ggplot2)
library(pROC)
```

```
ctrl <- trainControl(method="cv", number = 10)
heart_knn_1<- train(stroke ~ ., data = heart_pca1,
             method = "knn",
             trControl = ctrl,
             preProcess = c("center","scale"))
#Output of kNN fit

heart_pred_knn <- predict(heart_knn,heart_dum)
cm <- confusionMatrix(heart_dum$storke, heart_pred_knn)
# Store the byClass object of confusion matrix as a dataframe
metrics <- as.data.frame(cm$byClass)
# View the object
metrics
library(pROC)
# Get the precision value for each class
metrics %>% select(row("Precision"))

summary(heart_dum)

index = createDataPartition(y=heart_pca1$stroke, p=0.7, list=FALSE)
# Everything in the generated index list
train_pima = heart_pca1[index,]
# Everything except the generated indices
test_pima = heart_pca1[-index,]

# Set control parameter
train_control = trainControl(method = "cv", number = 10)
# Fit the model
knn <- train(stroke ~., data = train_pima, method = "knn", trControl = train_control, tuneLength = 20)
# Evaluate fit
knn

library(pROC)
# Get the precision value for each class
metrics %>% select(row("Precision"))


library(pROC)
# Get class probabilities for KNN
pred_prob <- predict(heart_knn, heart_pca1 , type = "prob")
head(pred_prob)

# And now we can create an ROC curve for our model.
roc_obj <- roc((heart_pca1$stroke), pred_prob[,1])
plot(roc_obj, print.auc=TRUE)

plot(heart_knn)
```