VISVESVARAYATECHNOLOGICALUNIVERSITY
JNANASANGAMA, BELGAUM-590014

ADDITIONAL ACTIVITY-1

FOUNDATIONS OF DATA SCIENCE

INTRODUCTION TO R PROGRAMMING

Team members:

SOHANA C A	**1SI19CS117**
SUHA SAEED	**1SI19CS124**
THANUSHREE S U	**1SI19CS129**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKUR-572103

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belgaum, Recognized by AICTE and Accredited by NBA, New Delhi)

2021-2022

INTRODUCTION

R is an open-source programming language that is widely used as a statistical software and data analysis tool. R generally comes with the Command-line interface. R is available across widely used platforms like Windows, Linux, and macOS. Also, the R programming language is the latest cutting-edge tool.

It was designed by **Ross Ihaka and Robert Gentleman** at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. R programming language is an implementation of the S programming language. It also combines with lexical scoping semantics inspired by Scheme. Moreover, the project conceives in 1992, with an initial version released in 1995 and a stable beta version in 2000.

WHAT IS R PRORAMMING?

- R programming is used as a leading tool for machine learning, statistics, and data analysis. Objects, functions, and packages can easily be created by R.
- It's a platform-independent language. This means it can be applied to all operating system.
- It's an open-source free language. That means anyone can install it in any organization without purchasing a license.
- R programming language is not only a statistic package but also allows us to integrate with other languages (C, C++). Thus, you can easily interact with many data sources and statistical packages.
- The R programming language has a vast community of users and it's growing day by day.
- R is currently one of the most requested programming languages in the Data Science job market that makes it the hottest trend nowadays.

FEATURES OF R

- **Basic Statistics:** The most common basic statistics terms are the mean, mode, and median. These are all known as "Measures of Central Tendency." So using the R language we can measure central tendency very easily.
- **Static graphics:** R is rich with facilities for creating and developing interesting static graphics. R contains functionality for many plot types including graphic maps, mosaic plots, biplots, and the list goes on.
- **Probability distributions:** Probability distributions play a vital role in statistics and by using R we can easily handle various types of probability distribution such as Binomial Distribution, Normal Distribution, Chi-squared Distribution and many more.
- **Data analysis:** It provides a large, coherent and integrated collection of tools for data analysis.

CONCEPTS IN R USED IN OUR PROGRAM

Variables

Variables are used to store the information to be manipulated and referenced in the R program. The R variable can store an atomic vector, a group of atomic vectors, or a combination of many R object.Variables can be created using the "<-" (assignment) operator.

Ex:num<-123

If Else Statement

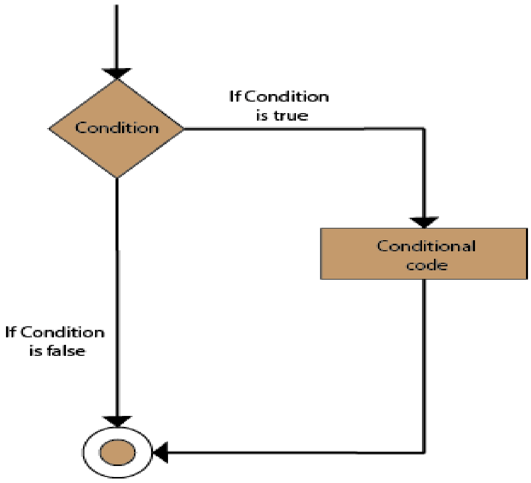
The if statement consists of the Boolean expressions followed by one or more statements. The if statement is the simplest decision-making statement which helps us to take a decision on the basis of the condition.

The block of code inside the if statement will be executed only when the boolean expression evaluates to be true. If the statement evaluates false, then the code which is mentioned after the condition will run.

Syntax:

```
if(boolean_expression) {
// If the boolean expression is true, then statement(s) will be executed.
}
```

Flow Chart



While Loop

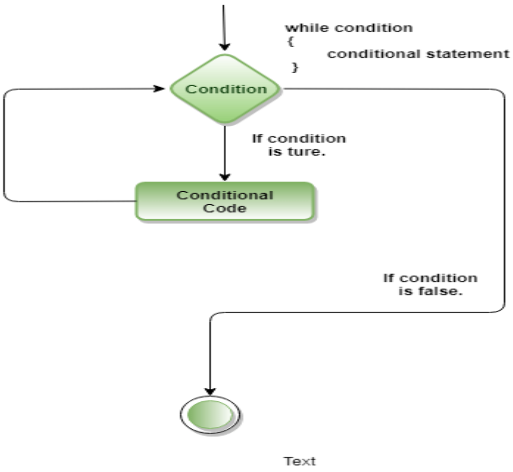
A while loop is a type of control flow statements which is used to iterate a block of code several numbers of times. The while loop terminates when the value of the Boolean expression will be false.

In while loop, firstly the condition will be checked and then after the body of the statement will execute. In this statement, the condition will be checked n+1 time, rather than n times.

Syntax:

```
while (test_expression) {
statement
}
```

Flow Chart:



Paste function

Paste function in R is used to concatenate Vectors by converting them into character.

Ex:

```
paste('one',2,'three',4,'five')
```

Readline function

Reads a line from the terminal (in interactive use).

Ex:

```
readline(prompt=” ”)
```

as.integer() function

Converts a value into an integer type using the **as.integer()** function.

PROGRAM TO FIND THE FREQUENCY OF A DIGIT IN THE NUMBER.

```
num = as.integer(readline(prompt="Enter a number: "))
digit = as.integer(readline(prompt="Enter digit: "))
n=num
count = 0
while(num > 0) {
        if(num%%10==digit){
            countcount=count+1
        }
        num=as.integer(num/10)
}
print(paste("The frequency of",digit,"in",n,"is=",count))
```

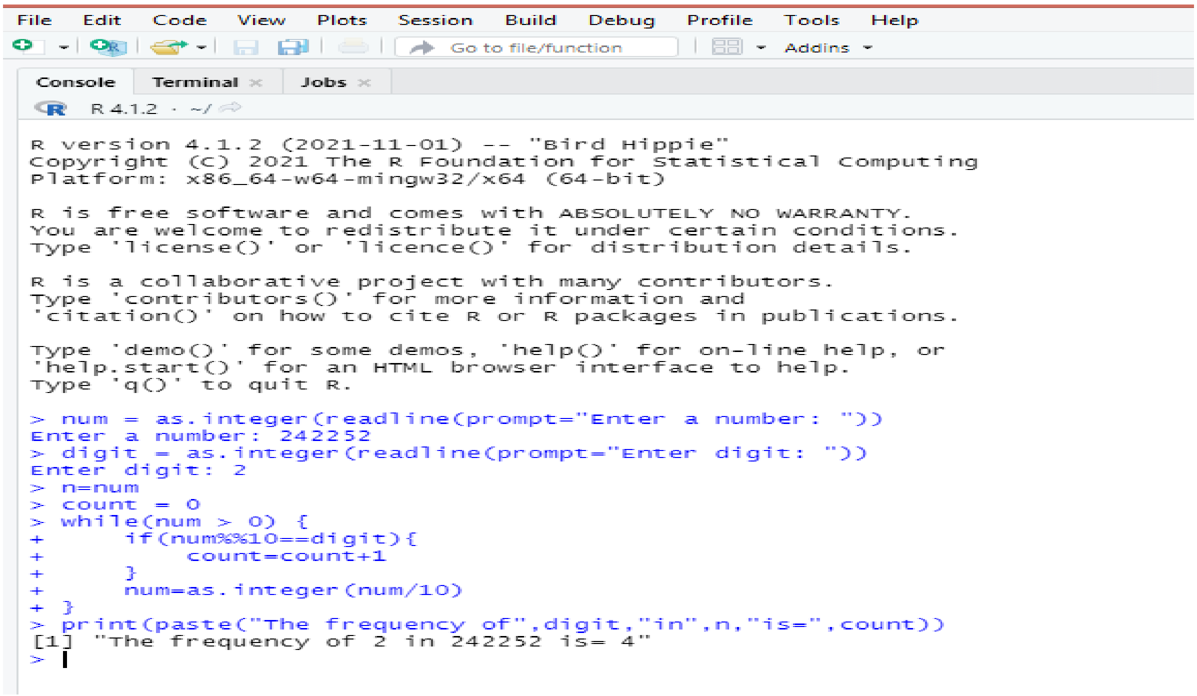
Explanation:

The above program calculates the frequency of a digit in a number. Both the digit and the number are taken from the user. The program prints the count of the digit in the number at the end.

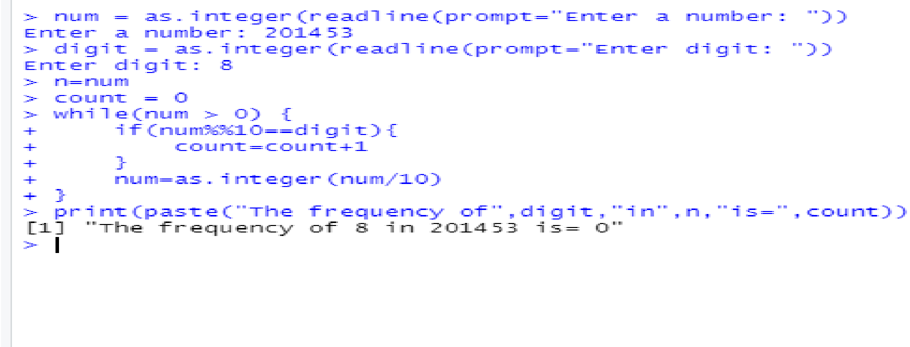
Both ,the 'num' and 'digit' variables are read from the console using the readline function.The as.integer() function receives a value in the form of an integer. The value of 'num' is assigned to another variable as num value gets changed through the process. The value of count is initialized to zero.Inside ,the while loop, the last digit of the number is compared with the digit to be counted, accordingly, the count is incremented, then the 'num' is reduced by 10. Again, the loop is executed if the num is greater than 0.

When the condition of the while loop fails, the count is printed. Here, paste function is used to concatenate the strings inside the bracket.

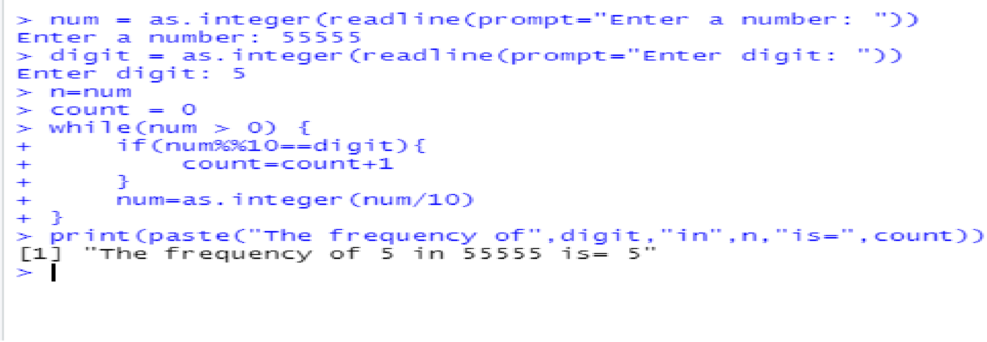
OUTPUT



```
R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> num = as.integer(readline(prompt="Enter a number: "))
Enter a number: 242252
> digit = as.integer(readline(prompt="Enter digit: "))
Enter digit: 2
> n=num
> count = 0
> while(num > 0) {
+       if(num%%10==digit){
+               count=count+1
+       }
+       num=as.integer(num/10)
+ }
> print(paste("The frequency of",digit,"in",n,"is=",count))
[1] "The frequency of 2 in 242252 is= 4"
> 
```



```
> num = as.integer(readline(prompt="Enter a number: "))
Enter a number: 201453
> digit = as.integer(readline(prompt="Enter digit: "))
Enter digit: 8
> n=num
> count = 0
> while(num > 0) {
+       if(num%%10==digit){
+               count=count+1
+       }
+       num=as.integer(num/10)
+ }
> print(paste("The frequency of",digit,"in",n,"is=",count))
[1] "The frequency of 8 in 201453 is= 0"
> 
```



```
> num = as.integer(readline(prompt="Enter a number: "))
Enter a number: 55555
> digit = as.integer(readline(prompt="Enter digit: "))
Enter digit: 5
> n=num
> count = 0
> while(num > 0) {
+       if(num%%10==digit){
+               count=count+1
+       }
+       num=as.integer(num/10)
+ }
> print(paste("The frequency of",digit,"in",n,"is=",count))
[1] "The frequency of 5 in 55555 is= 5"
> 
```

FDS ACTIVITY 2

TOPIC: INSURANCE COST PREDICTION USING LINEAR REGRESSION

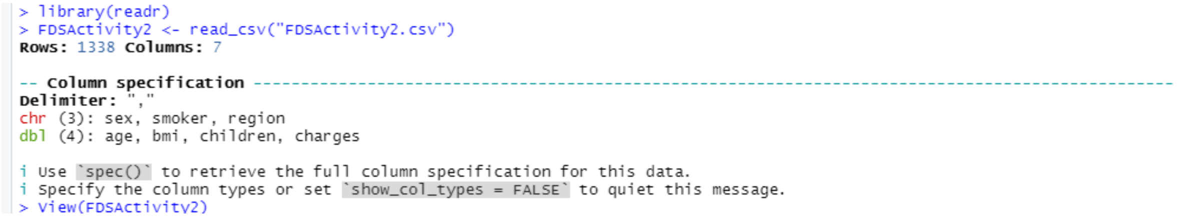
INTRODUCTION

Having insurance in today's scenario is really useful and important. Because we never know if we will have a property with us permanently, as nothing is permanent in this world. We have taken this topic as our activity 2 for Foundations of Data Science.

In this assignment, we're going to use information like a person's age, sex, BMI, no. of children, region and smoking habit to predict the price of yearly medical bills. This kind of model is useful for insurance companies to determine the yearly insurance premium for a person.

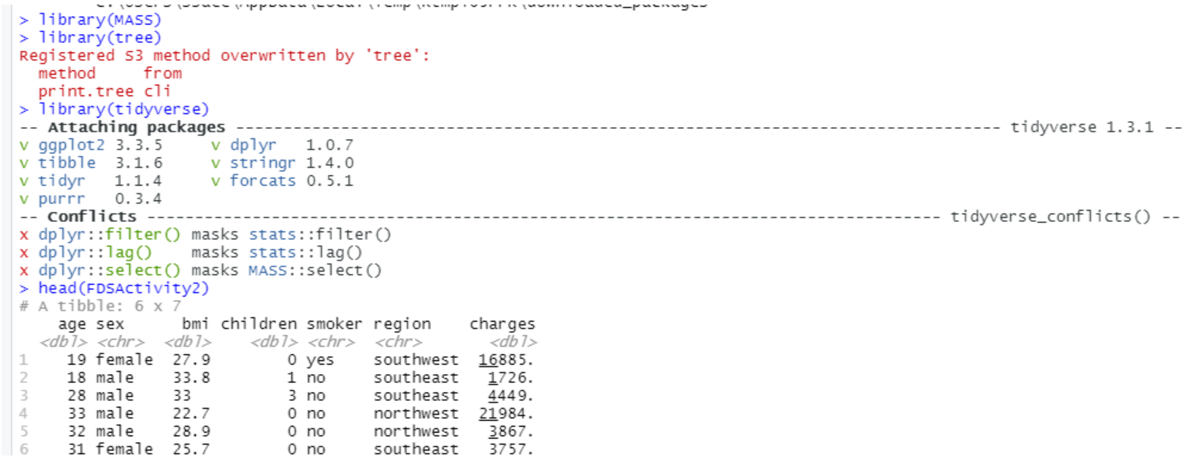
1.Collection of dataset:

The dataset was collected from www.kaggle.com , which consists of 1338 tuples and 7 columns. The dataset was imported as a csv file into Rstudio.



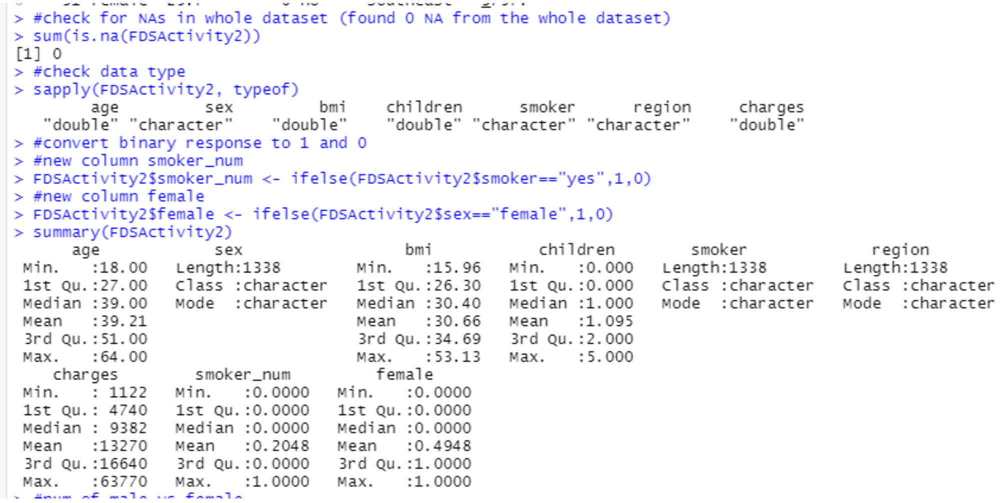
```
> library(readr)
> FDSActivity2 <- read_csv("FDSActivity2.csv")
Rows: 1338 Columns: 7

-- Column specification --------------------------------------------------------------------------------------------------
Delimiter: ","
chr (3): sex, smoker, region
dbl (4): age, bmi, children, charges

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
> View(FDSActivity2)
```

Then we load the necessary libraries to facilitate the execution of our code. Then we load the dataset also.



```
> library(MASS)
> library(tree)
Registered S3 method overwritten by 'tree':
  method     from
  print.tree cli
> library(tidyverse)
-- Attaching packages ---------------------------------------------------------------------------------------- tidyverse 1.3.1 --
v ggplot2 3.3.5     v dplyr   1.0.7
v tibble  3.1.6     v stringr 1.4.0
v tidyr   1.1.4     v forcats 0.5.1
v purrr   0.3.4
-- Conflicts ------------------------------------------------------------------------------------- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
x dplyr::select() masks MASS::select()
> head(FDSActivity2)
# A tibble: 6 x 7
    age sex      bmi children smoker region    charges
  <dbl> <chr>  <dbl>    <dbl> <chr>  <chr>       <dbl>
1    19 female  27.9        0 yes    southwest  16885.
2    18 male    33.8        1 no     southeast   1726.
3    28 male    33          3 no     southeast   4449.
4    33 male    22.7        0 no     northwest  21984.
5    32 male    28.9        0 no     northwest   3867.
6    31 female  25.7        0 no     southeast   3757.
```

After this, we clean the dataset. We check for NAs in the whole dataset, check the datatype of each attribute, and convert binary response to 1 and 0.

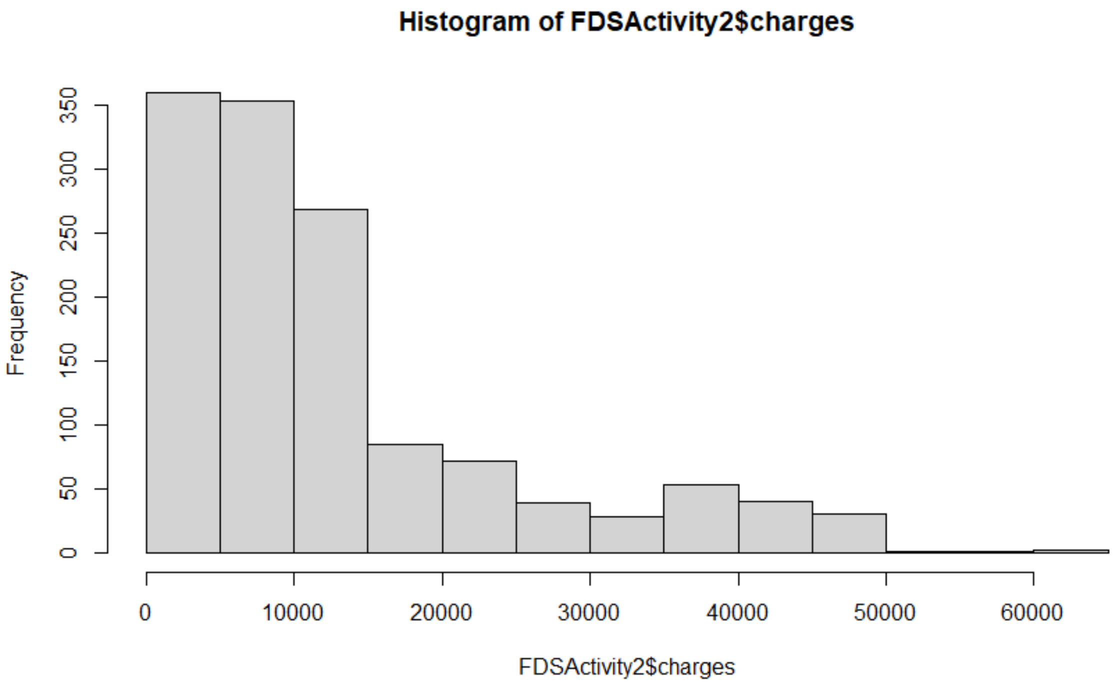


```
> #check for NAs in whole dataset (found 0 NA from the whole dataset)
> sum(is.na(FDSActivity2))
[1] 0
> #check data type
> sapply(FDSActivity2, typeof)
        age         sex         bmi    children      smoker      region     charges 
   "double" "character"    "double"    "double" "character" "character"    "double" 
> #convert binary response to 1 and 0
> #new column smoker_num
> FDSActivity2$smoker_num <- ifelse(FDSActivity2$smoker=="yes",1,0)
> #new column female
> FDSActivity2$female <- ifelse(FDSActivity2$sex=="female",1,0)
> summary(FDSActivity2)
      age            sex                 bmi           children        smoker             region         
 Min.   :18.00   Length:1338        Min.   :15.96   Min.   :0.000   Length:1338        Length:1338       
 1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000   Class :character   Class :character  
 Median :39.00   Mode  :character   Median :30.40   Median :1.000   Mode  :character   Mode  :character  
 Mean   :39.21                      Mean   :30.66   Mean   :1.095                                        
 3rd Qu.:51.00                      3rd Qu.:34.69   3rd Qu.:2.000                                        
 Max.   :64.00                      Max.   :53.13   Max.   :5.000                                        
    charges        smoker_num         female      
 Min.   : 1122   Min.   :0.0000   Min.   :0.0000  
 1st Qu.: 4740   1st Qu.:0.0000   1st Qu.:0.0000  
 Median : 9382   Median :0.0000   Median :0.0000  
 Mean   :13270   Mean   :0.2048   Mean   :0.4948  
 3rd Qu.:16640   3rd Qu.:0.0000   3rd Qu.:1.0000  
 Max.   :63770   Max.   :1.0000   Max.   :1.0000  
```

We check for the no. of males and females, mean insurance charges of smokers vs non-smokers, mean insurance charges of male vs female, etc.

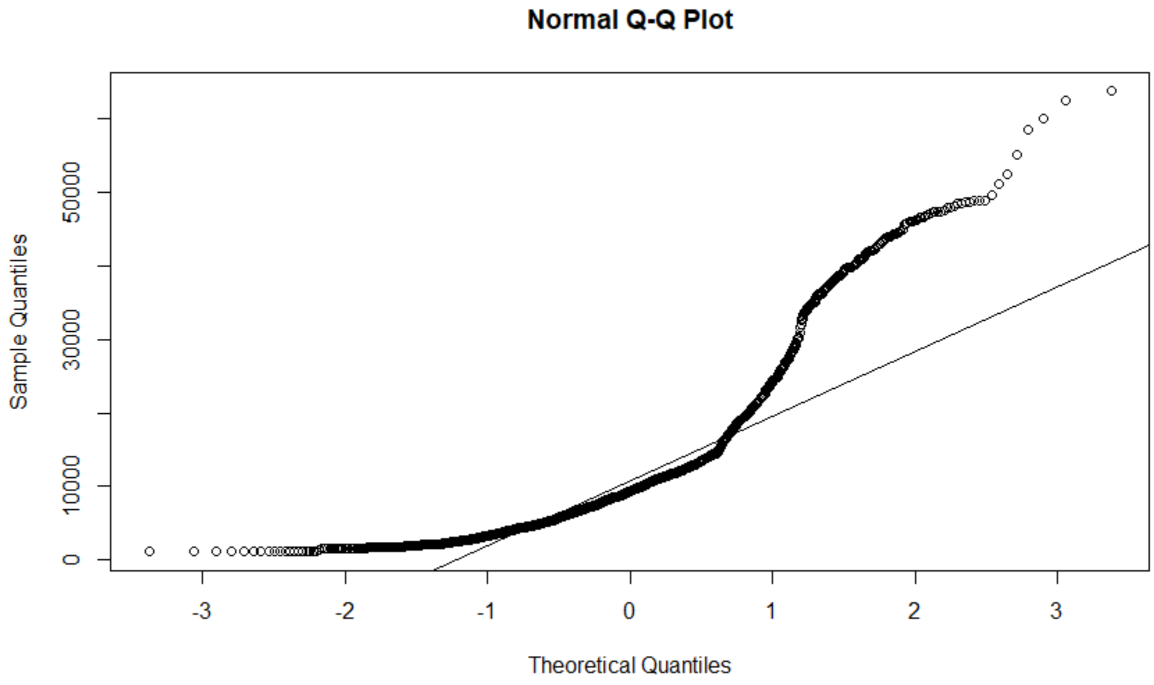
Here’s the histogram of the charges variable to check whether the data is normally distributed or not.

```
#histogram of charges variable
#checking whether the data is normally distributed or not
hist(FDSActivity2$charges)
```



Then we have the QQ plot below to check how linear the data is.

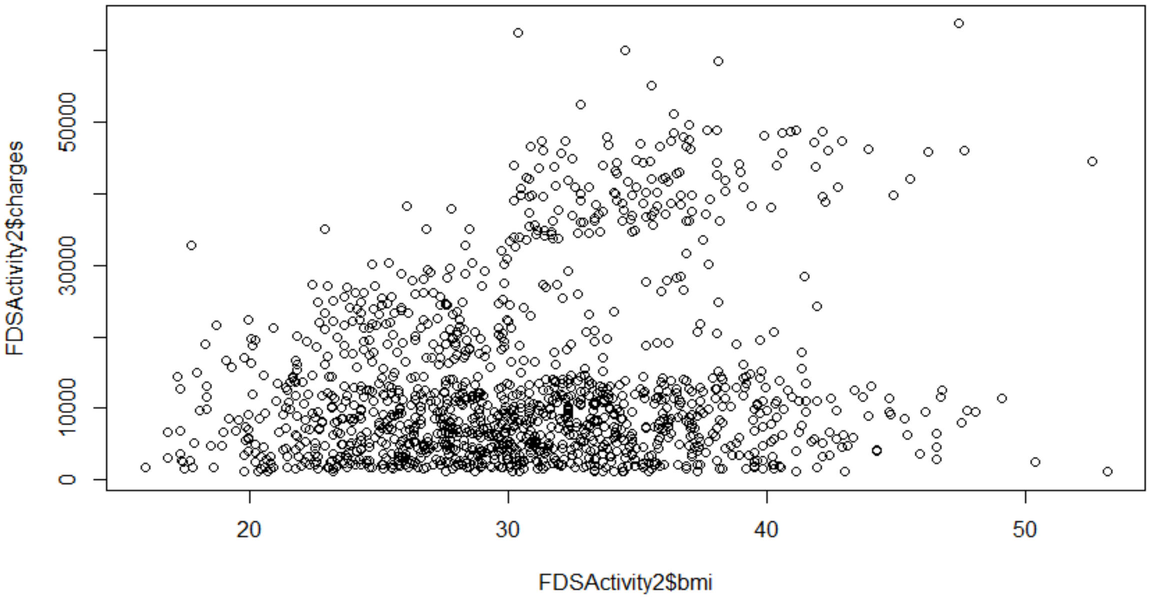
```
#qqplot and qqline for charges variable
#checking on how linear the data is
qqnorm(FDSActivity2$charges)
qqline(FDSActivity2$charges)
```



And here are some graphs that we have plotted of various attributes against the target attribute. These help in determining how the attributes determine the target value.

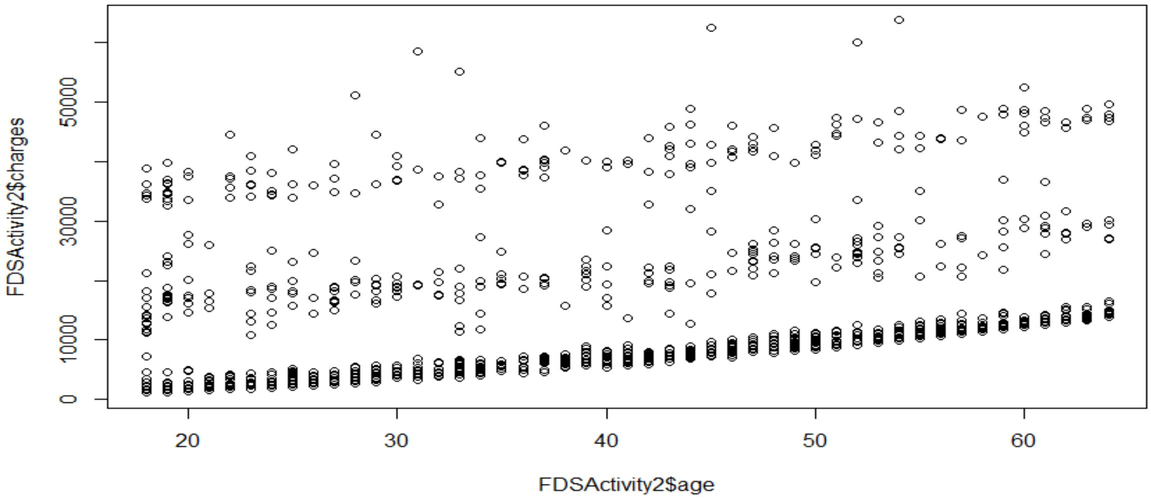
1. BMI vs CHARGES PLOT:

```
#plot bmi vs charges
plot(x=FDSActivity2$bmi,y=FDSActivity2$charges)
```



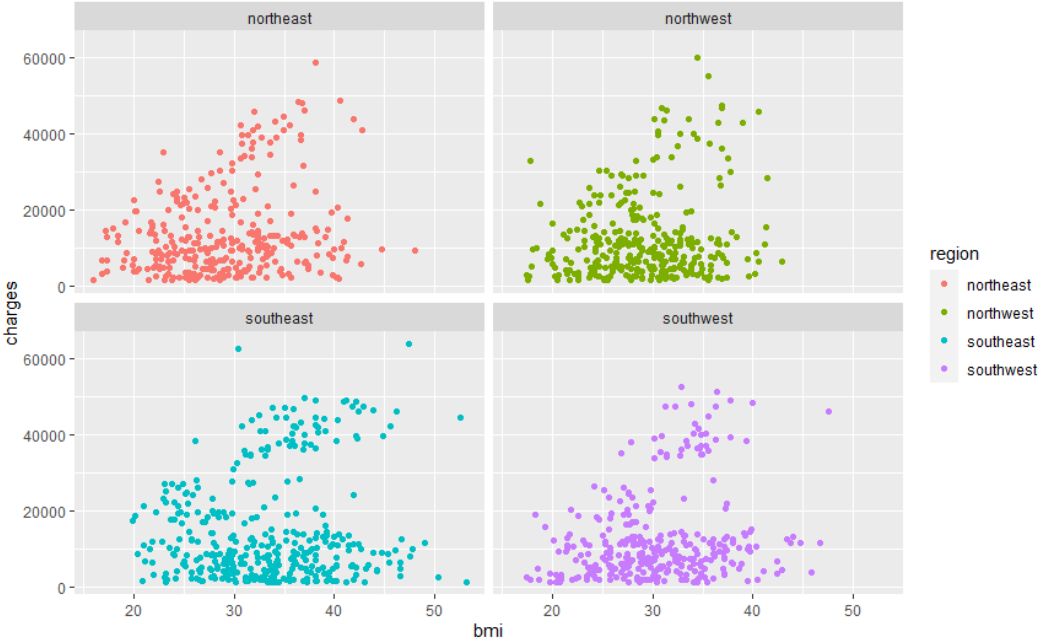
2. AGE vs CHARGES PLOT:

```
#plot age vs charges
plot(x=FDSActivity2$age,y=FDSActivity2$charges)
```



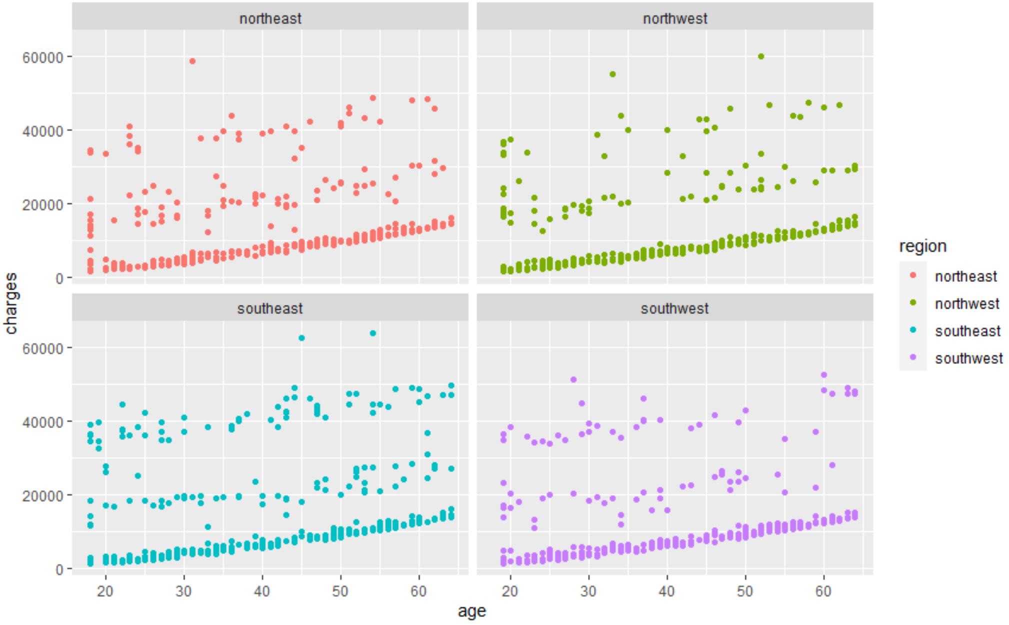
3. GG Plot for BMI vs CHARGES grouped by region:

```
#ggplot for bmi vs charges grouped by region
ggplot(data=FDSActivity2,aes(x=bmi,y=charges)) + geom_point(aes(color=r
egion)) +facet_wrap(~region)
```

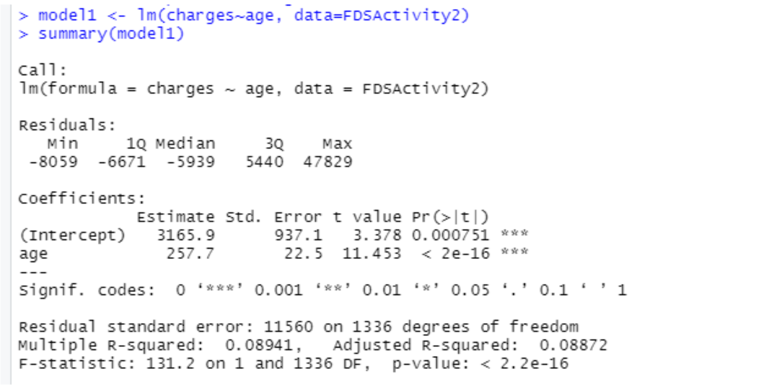


4. GG Plot for AGE vs CHARGES grouped by region:

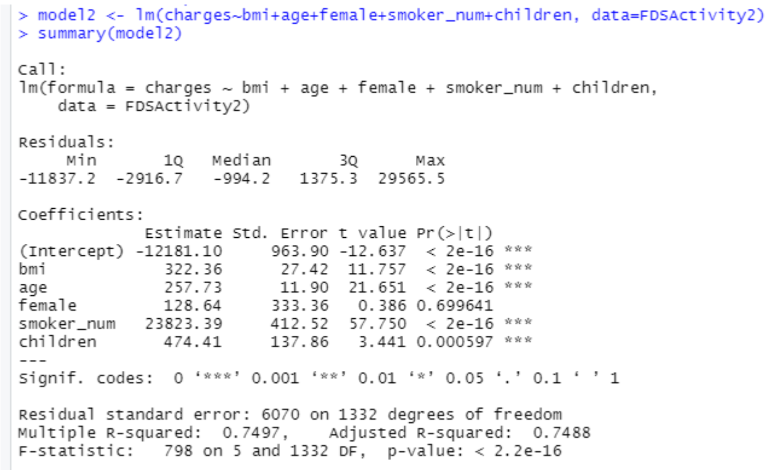
```
#ggplot for age vs charges grouped by region
ggplot(data=FDSActivity2,aes(x=age,y=charges)) + geom_point(aes(color=r
egion)) +facet_wrap(~region)
```



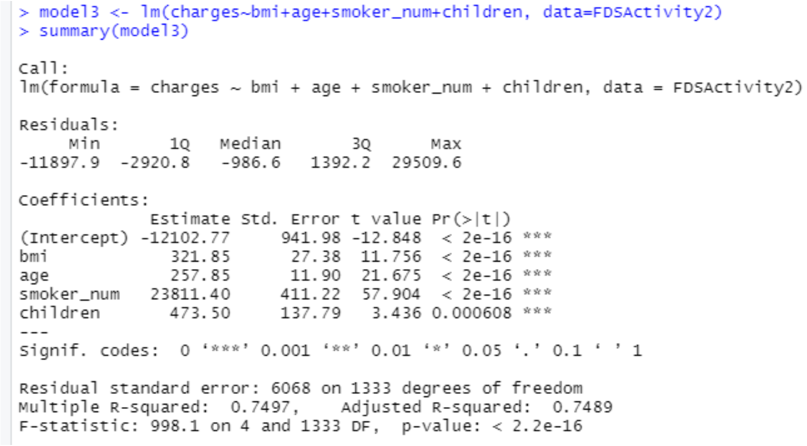
After this, we have created 3 models, by taking various attribute values against the target variable.



```
> model1 <- lm(charges~age, data=FDSActivity2)
> summary(model1)

Call:
lm(formula = charges ~ age, data = FDSActivity2)

Residuals:
   Min     1Q Median     3Q    Max 
 -8059  -6671  -5939   5440  47829 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   3165.9      937.1   3.378 0.000751 ***
age            257.7       22.5  11.453  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11560 on 1336 degrees of freedom
Multiple R-squared:  0.08941,	Adjusted R-squared:  0.08872 
F-statistic: 131.2 on 1 and 1336 DF,  p-value: < 2.2e-16
```

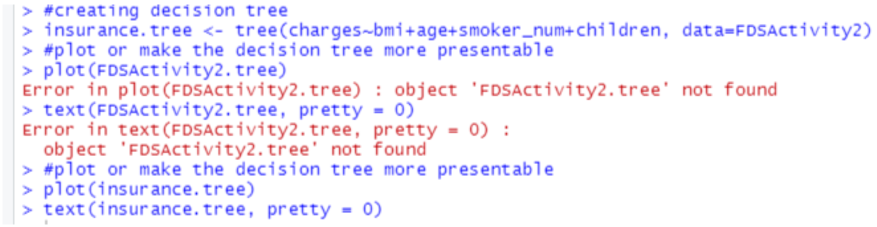


```
> model2 <- lm(charges~bmi+age+female+smoker_num+children, data=FDSActivity2)
> summary(model2)

Call:
lm(formula = charges ~ bmi + age + female + smoker_num + children, 
    data = FDSActivity2)

Residuals:
     Min       1Q   Median       3Q      Max 
-11837.2  -2916.7   -994.2   1375.3  29565.5 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -12181.10     963.90 -12.637  < 2e-16 ***
bmi            322.36      27.42  11.757  < 2e-16 ***
age            257.73      11.90  21.651  < 2e-16 ***
female         128.64     333.36   0.386 0.699641    
smoker_num   23823.39     412.52  57.750  < 2e-16 ***
children       474.41     137.86   3.441 0.000597 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6070 on 1332 degrees of freedom
Multiple R-squared:  0.7497,	Adjusted R-squared:  0.7488 
F-statistic:   798 on 5 and 1332 DF,  p-value: < 2.2e-16
```

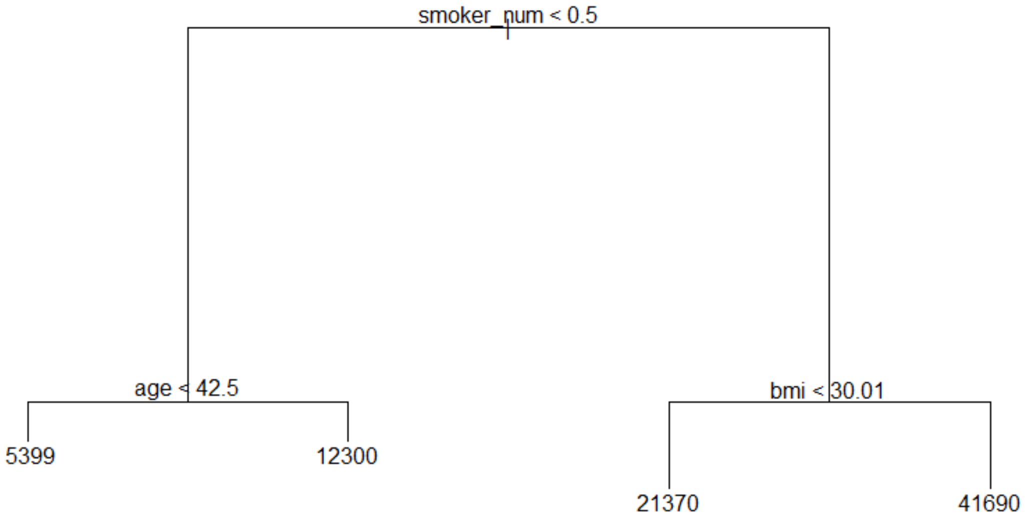


```
> model3 <- lm(charges~bmi+age+smoker_num+children, data=FDSActivity2)
> summary(model3)

Call:
lm(formula = charges ~ bmi + age + smoker_num + children, data = FDSActivity2)

Residuals:
     Min       1Q   Median       3Q      Max 
-11897.9  -2920.8   -986.6   1392.2  29509.6 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -12102.77     941.98 -12.848  < 2e-16 ***
bmi            321.85      27.38  11.756  < 2e-16 ***
age            257.85      11.90  21.675  < 2e-16 ***
smoker_num   23811.40     411.22  57.904  < 2e-16 ***
children       473.50     137.79   3.436 0.000608 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6068 on 1333 degrees of freedom
Multiple R-squared:  0.7497,	Adjusted R-squared:  0.7489 
F-statistic: 998.1 on 4 and 1333 DF,  p-value: < 2.2e-16
```

After we created all the models, we create the decision tree as to what attributes accurately determine the target value.



```
> #creating decision tree
> insurance.tree <- tree(charges~bmi+age+smoker_num+children, data=FDSActivity2)
> #plot or make the decision tree more presentable
> plot(FDSActivity2.tree)
Error in plot(FDSActivity2.tree) : object 'FDSActivity2.tree' not found
> text(FDSActivity2.tree, pretty = 0)
Error in text(FDSActivity2.tree, pretty = 0) :
  object 'FDSActivity2.tree' not found
> #plot or make the decision tree more presentable
> plot(insurance.tree)
> text(insurance.tree, pretty = 0)
```


smoker_num < 0.5
age < 42.5
bmi < 30.01
5399
12300
21370
41690