IDS Project

To analyze the data, firstly we have to retrieve data from the link. So, we used the read.csv function in the tidyverse library. This function read and retrieves the data from the link and store it as data set form in a dataset named vector.

Now, the following functions allows us to check the variables and their statistical functions. By using those functions we get to know that there are 14 columns and 7582 rows in the dataset. The whole dataset gives information about the expenses of a person with different habits.In this dataset there are int, num and chr data types variables.

```
str(dataset)
```

```
## 'data.frame':    7582 obs. of  14 variables:
##  $ X              : int  1 2 3 4 5 7 9 10 11 12 ...
##  $ age            : int  18 19 27 34 32 47 36 59 24 61 ...
##  $ bmi            : num  27.9 33.8 33 22.7 28.9 ...
##  $ children       : int  0 1 3 0 0 1 2 0 0 0 ...
##  $ smoker         : chr  "yes" "no" "no" "no" ...
##  $ location       : chr  "CONNECTICUT" "RHODE ISLAND" "MASSACHUSETTS" "PENNSYLVANIA" ...
##  $ location_type  : chr  "Urban" "Urban" "Urban" "Country" ...
##  $ education_level: chr  "Bachelor" "Bachelor" "Master" "Master" ...
##  $ yearly_physical: chr  "No" "No" "No" "No" ...
##  $ exercise       : chr  "Active" "Not-Active" "Active" "Not-Active" ...
##  $ married        : chr  "Married" "Married" "Married" "Married" ...
##  $ hypertension   : int  0 0 0 1 0 0 0 1 0 0 ...
##  $ gender         : chr  "female" "male" "male" "male" ...
##  $ cost           : int  1746 602 576 5562 836 3842 1304 9724 201 4492 ...
```

```
summary(dataset)
```

```
##        X                  age             bmi           children    
##  Min.   :        1   Min.   :18.00   Min.   :15.96   Min.   :0.000  
##  1st Qu.:     5635   1st Qu.:26.00   1st Qu.:26.60   1st Qu.:0.000  
##  Median :    24916   Median :39.00   Median :30.50   Median :1.000  
##  Mean   :   712602   Mean   :38.89   Mean   :30.80   Mean   :1.109  
##  3rd Qu.:   118486   3rd Qu.:51.00   3rd Qu.:34.77   3rd Qu.:2.000  
##  Max.   :131101111   Max.   :66.00   Max.   :53.13   Max.   :5.000  
##                                      NA's   :78                     
##     smoker            location         location_type      education_level   
##  Length:7582        Length:7582        Length:7582        Length:7582       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
## 
## 
## 
## 
```

```
##  yearly_physical      exercise           married           hypertension
##  Length:7582        Length:7582        Length:7582        Min.   :0.0000
##  Class :character   Class :character   Class :character   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Mode  :character   Median :0.0000
##                                                           Mean   :0.2005
##                                                           3rd Qu.:0.0000
##                                                           Max.   :1.0000
##                                                           NA's   :80
##     gender               cost
##  Length:7582        Min.   :    2
##  Class :character   1st Qu.:  970
##  Mode  :character   Median : 2500
##                     Mean   : 4043
##                     3rd Qu.: 4775
##                     Max.   :55715
##
```

```
head(dataset,20)
```

```
##     X age    bmi children smoker      location location_type   education_level
## 1   1  18 27.900        0    yes   CONNECTICUT         Urban          Bachelor
## 2   2  19 33.770        1     no  RHODE ISLAND         Urban          Bachelor
## 3   3  27 33.000        3     no MASSACHUSETTS         Urban            Master
## 4   4  34 22.705        0     no  PENNSYLVANIA       Country            Master
## 5   5  32 28.880        0     no  PENNSYLVANIA       Country               PhD
## 6   7  47 33.440        1     no  PENNSYLVANIA         Urban          Bachelor
## 7   9  36 29.830        2     no  PENNSYLVANIA         Urban          Bachelor
## 8  10  59 25.840        0     no  PENNSYLVANIA       Country          Bachelor
## 9  11  24 26.220        0     no  PENNSYLVANIA         Urban          Bachelor
## 10 12  61 26.290        0    yes   CONNECTICUT         Urban No College Degree
## 11 13  22 34.400        0     no      MARYLAND         Urban          Bachelor
## 12 14  57 39.820        0     no      MARYLAND         Urban          Bachelor
## 13 15  26 42.130        0    yes  PENNSYLVANIA         Urban          Bachelor
## 14 16  18 24.600        1     no  PENNSYLVANIA       Country No College Degree
## 15 18  23 23.845        0     no MASSACHUSETTS         Urban No College Degree
## 16 19  57 40.300        0     no  PENNSYLVANIA         Urban          Bachelor
## 17 20  31 35.300        0    yes  PENNSYLVANIA         Urban               PhD
## 18 21  60 36.005        0     no  PENNSYLVANIA         Urban               PhD
## 19 22  30 32.400        1     no  PENNSYLVANIA         Urban            Master
## 20 23  19     NA        0     no  PENNSYLVANIA         Urban No College Degree
##    yearly_physical   exercise     married hypertension gender  cost
## 1               No     Active     Married            0 female  1746
## 2               No Not-Active     Married            0   male   602
## 3               No     Active     Married            0   male   576
## 4               No Not-Active     Married            1   male  5562
## 5               No Not-Active     Married            0   male   836
## 6               No Not-Active     Married            0 female  3842
## 7               No     Active     Married            0   male  1304
## 8               No Not-Active     Married            1 female  9724
## 9               No     Active     Married            0   male   201
## 10              No     Active     Married            0 female  4492
## 11              No Not-Active     Married            0   male   717
## 12             Yes Not-Active     Married            0 female  4153
## 13              No     Active     Married            0   male  5336
```

```
## 14             Yes Not-Active Not_Married            0   male   382
## 15              No     Active     Married            0   male   294
## 16             Yes     Active Not_Married            0   male  1382
## 17              No Not-Active     Married            0   male 15058
## 18              No     Active     Married            0 female  3384
## 19              No     Active     Married            0 female   761
## 20              No     Active Not_Married            0   male   146
```

```
# Getting to know the dataset
```

Once we are done with the data exploration then the next step is to check if there are any empty cells in the variables.If there are empty cells then we have to clean the data. The following function will give output of number of cells that are empty in the mentioned variable.The results shows that there 78 and 80 empty cells in the bmi and hypertension variables.

```
## [1] 78
```

```
## [1] 0
```

```
## [1] 0
```

```
## [1] 0
```

```
## [1] 80
```

```
## [1] 0
```

Now, we have to clean those empty cells in the bmi and hypertension variables. To do the cleaning we choose to use na_interpolation function in the imputeTS library.This function will clean data in the mentioned variable. Again, the function used is.na() function to verify whether the na_interpolation is worked is or not and the result shows there are no empty cells in the variables.

```
# Dealing with N/A values
library(imputeTS)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
dataset$bmi <- na_interpolation(dataset$bmi)
dataset$hypertension <- na_interpolation(dataset$hypertension)
sum(is.na(dataset$bmi))
```

```
## [1] 0
```

```
sum(is.na(dataset$hypertension))
```

```
## [1] 0
```

```r
sum(is.na(dataset$cost))
```

```
## [1] 0
```

Firstly we stored the dataset in the datalm and then converted all the chr data type into the factor data type so that it will be best to find out which varibles will be affecting the expenses of a person.

```r
# Converting data into factor types
datalm <- dataset
datalm$smoker <- as.factor(datalm$smoker)
datalm$location <- as.factor(datalm$location)
datalm$location_type <- as.factor(datalm$location_type)
datalm$education_level <- as.factor(datalm$education_level)
datalm$yearly_physical <- as.factor(datalm$yearly_physical)
datalm$exercise <- as.factor(datalm$exercise)
datalm$married <- as.factor(datalm$married)
datalm$gender <- as.factor(datalm$gender)
```

This following command is to verify the data types in the datalm dataset.

```r
str(datalm)
```

```
## 'data.frame':    7582 obs. of  14 variables:
##  $ X              : int  1 2 3 4 5 7 9 10 11 12 ...
##  $ age            : int  18 19 27 34 32 47 36 59 24 61 ...
##  $ bmi            : num  27.9 33.8 33 22.7 28.9 ...
##  $ children       : int  0 1 3 0 0 1 2 0 0 0 ...
##  $ smoker         : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 2 ...
##  $ location       : Factor w/ 7 levels "CONNECTICUT",..: 1 7 3 6 6 6 6 6 6 1 ...
##  $ location_type  : Factor w/ 2 levels "Country","Urban": 2 2 2 1 1 2 2 1 2 2 ...
##  $ education_level: Factor w/ 4 levels "Bachelor","Master",..: 1 1 2 2 4 1 1 1 1 3 ...
##  $ yearly_physical: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ exercise       : Factor w/ 2 levels "Active","Not-Active": 1 2 1 2 2 2 1 2 1 1 ...
##  $ married        : Factor w/ 2 levels "Married","Not_Married": 1 1 1 1 1 1 1 1 1 1 ...
##  $ hypertension   : num  0 0 0 1 0 0 0 1 0 0 ...
##  $ gender         : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 2 1 2 1 ...
##  $ cost           : int  1746 602 576 5562 836 3842 1304 9724 201 4492 ...
```

```r
# Understanding the datalm dataset
```

Our goal is to determine what are the key variables that affects the most in the expenses of a person and to do that we have to find the cost margin to determine whether a person is considered as expensive or not. Expensive means costliest so the we used max function in the cost variable. Also summary function on the cost to check the mean and quartiles. the results shows a big difference between the 3rd quartiles and maximum in the cost variable.

```r
dataset[which.max(dataset$cost), ]
```

```
##          X age  bmi children smoker    location location_type education_level
## 3493 32921  62 33.8        1    yes CONNECTICUT         Urban        Bachelor
##      yearly_physical   exercise     married hypertension gender  cost
## 3493              No Not-Active Not_Married            1 female 55715
```

```r
dataset[which.min(dataset$cost), ]
```

```
##         X age   bmi children smoker   location location_type education_level
## 2210 9411  20 23.21        0     no NEW JERSEY         Urban        Bachelor
##      yearly_physical exercise married hypertension gender cost
## 2210              No   Active Married            0   male    2
```

```r
dataset[which.max(dataset$bmi), ]
```

```
##        X age   bmi children smoker     location location_type education_level
## 988 1318  19 53.13        0     no PENNSYLVANIA         Urban        Bachelor
##     yearly_physical   exercise     married hypertension gender cost
## 988              No Not-Active Not_Married            0   male  331
```

```r
dataset[which.min(dataset$bmi), ]
```

```
##       X age   bmi children smoker     location location_type education_level
## 131 173  18 15.96        0     no PENNSYLVANIA       Country          Master
##     yearly_physical   exercise married hypertension gender cost
## 131              No Not-Active Married            0   male  213
```

```r
summary(dataset$cost)
```

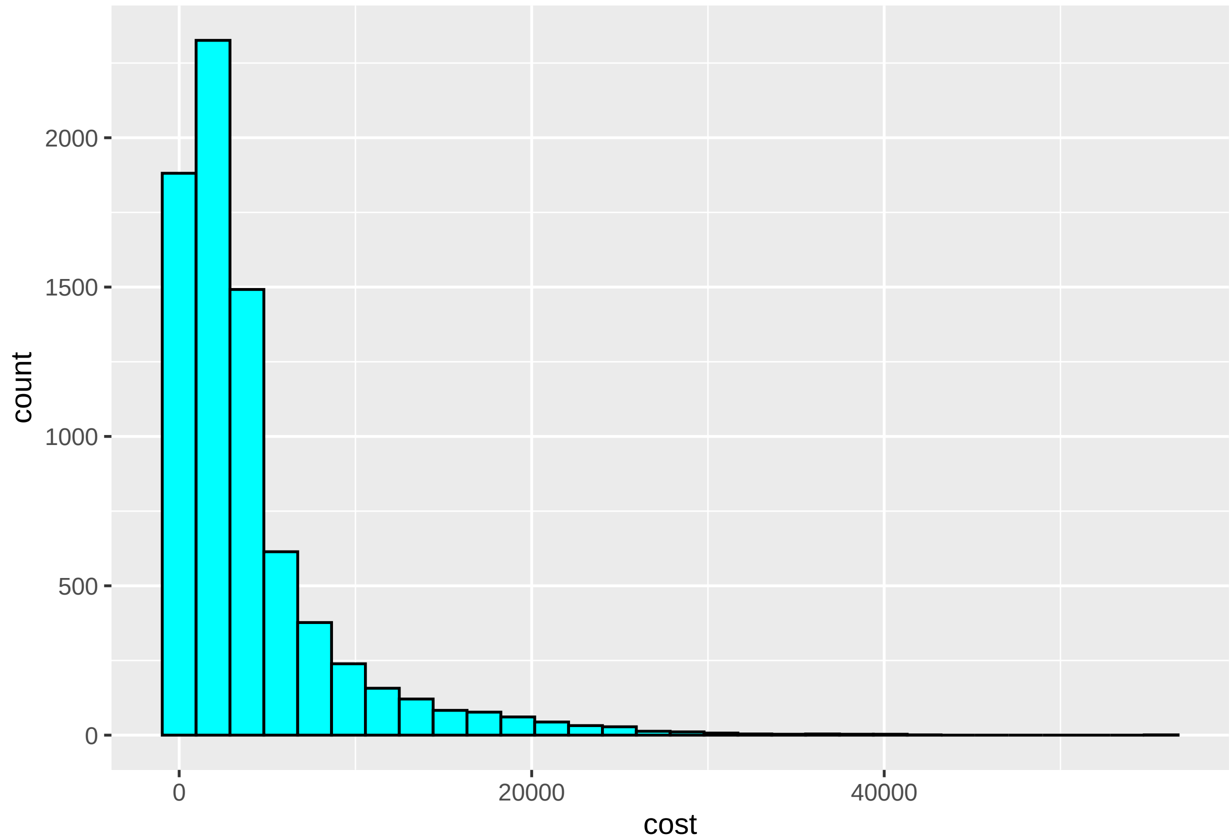
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       2     970    2500    4043    4775   55715
```

```r
# Determining the minimum and maximums of cost and bmi
```

To get more clarity on the cost variable, we created histogram graph and the resulting graph is a right-skewed graph which means most of the bars are on the left side of the graph. these most frequent bars are in range of between 4000 to 5000 and the graphs there are out-liers with only frequency of 1 so we choose mean of the cost variable as a margin cost to determine whether a the expenses considered as are expensive are not.

```r
library(ggplot2)
ggplot(dataset)+aes(cost)+geom_histogram(fill='cyan', col='black')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

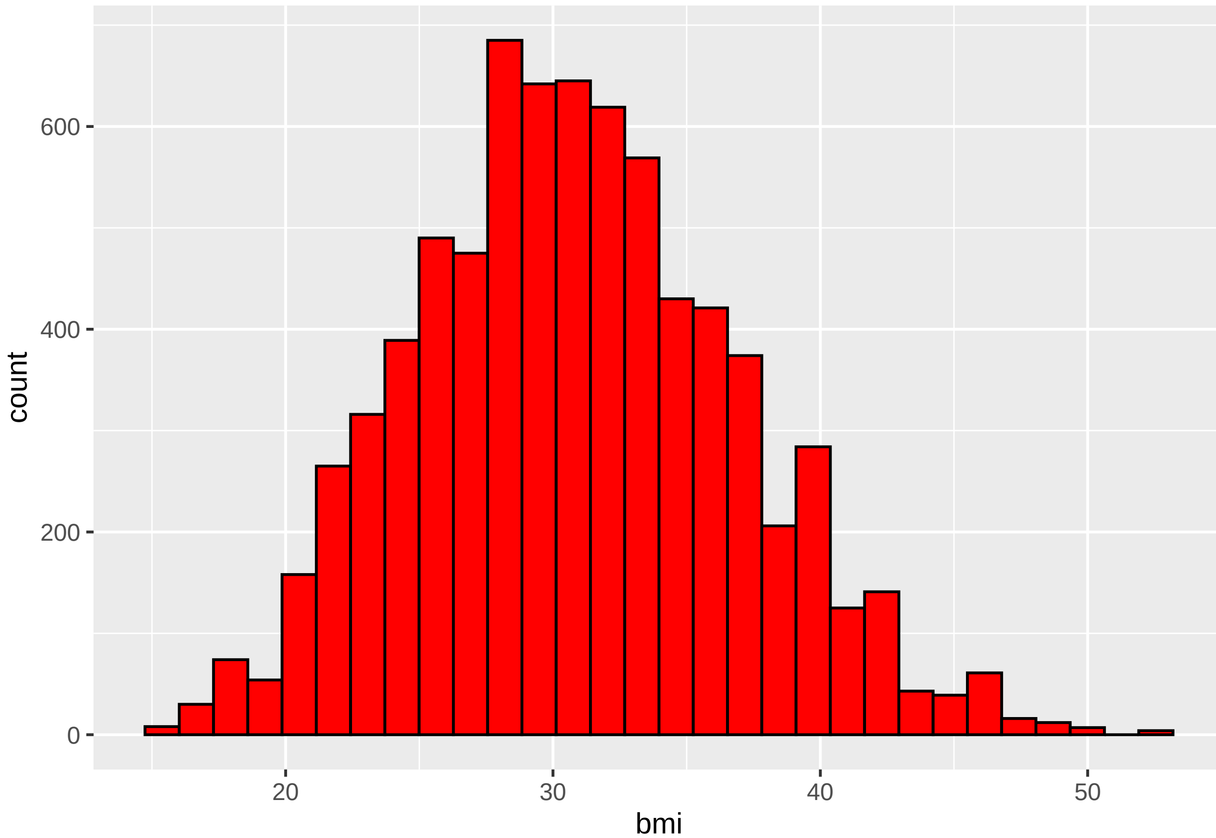



```
# Histogram of cost variable
# It is a right-skewed plot
```

This is a histogram of a bmi variable and the resulting graph is bell-curve shaped which means most of the frequent bmi values are situated around the median of the variable, the median is 30.50. As per standard chart of bmi, if the bmi is greater than 30 then that person is suffering with obesity.

```
ggplot(dataset)+aes(bmi)+geom_histogram(fill='red', col='black')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

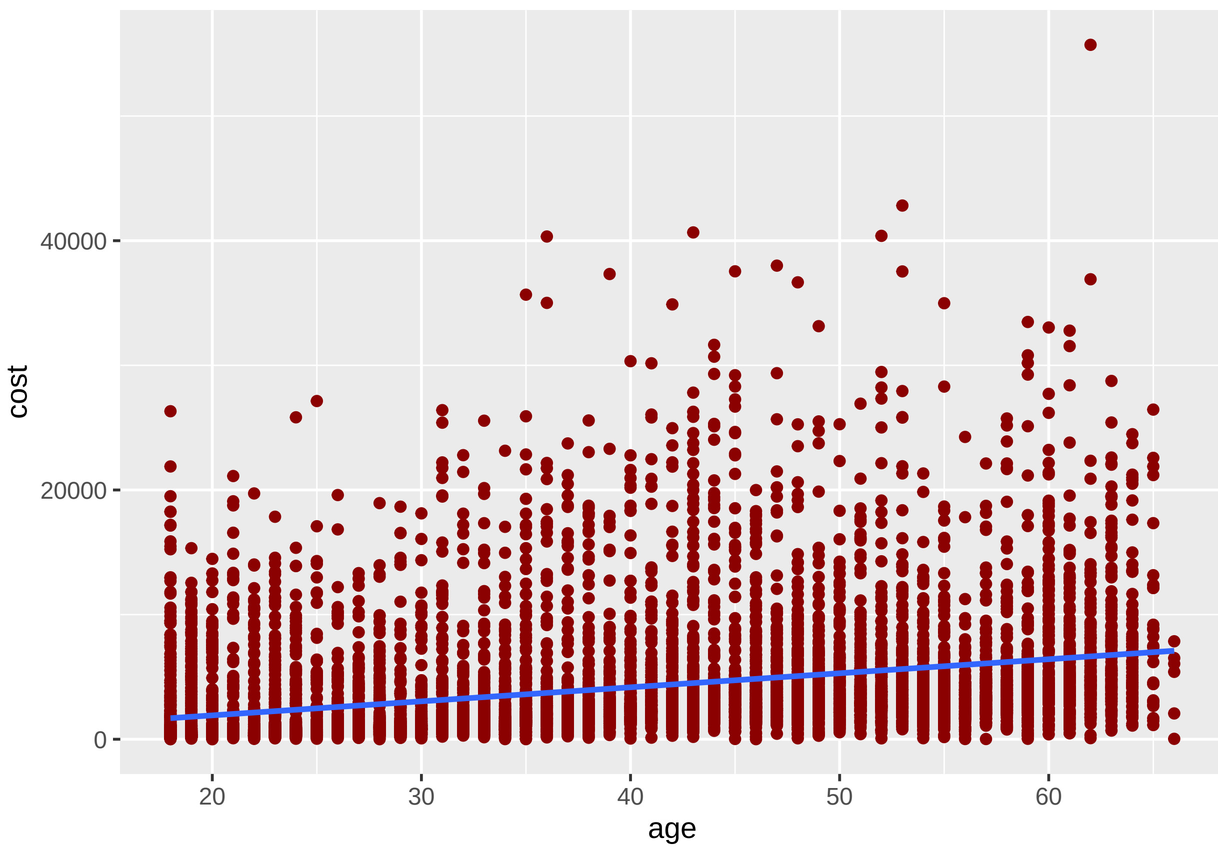



```
# Histogram of BMI variable
# It is a bell-curve plot
```

Now we created scatter plots and box-plots to understand the relationships the between the variables.

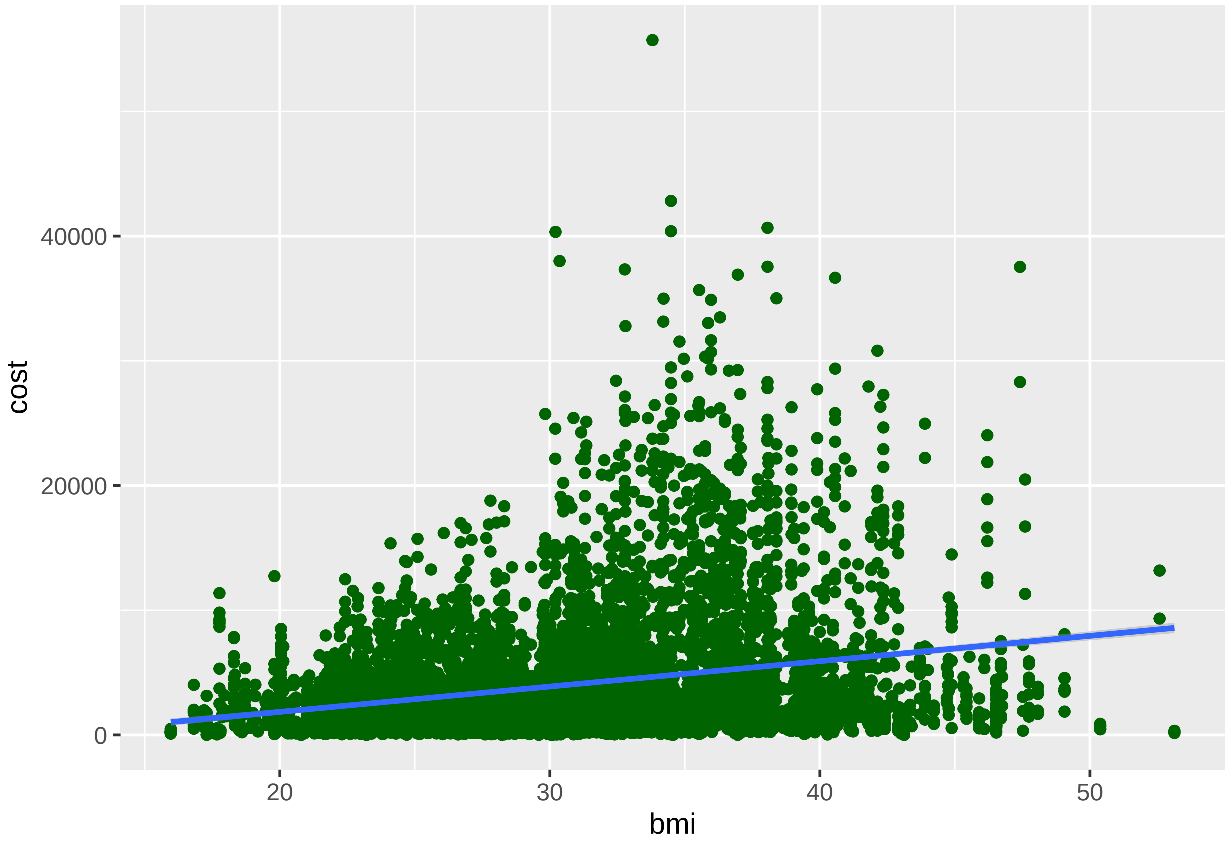
```
ggplot(dataset)+aes(age,cost)+geom_point(col='darkred')+geom_smooth(method="lm", se=TRUE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

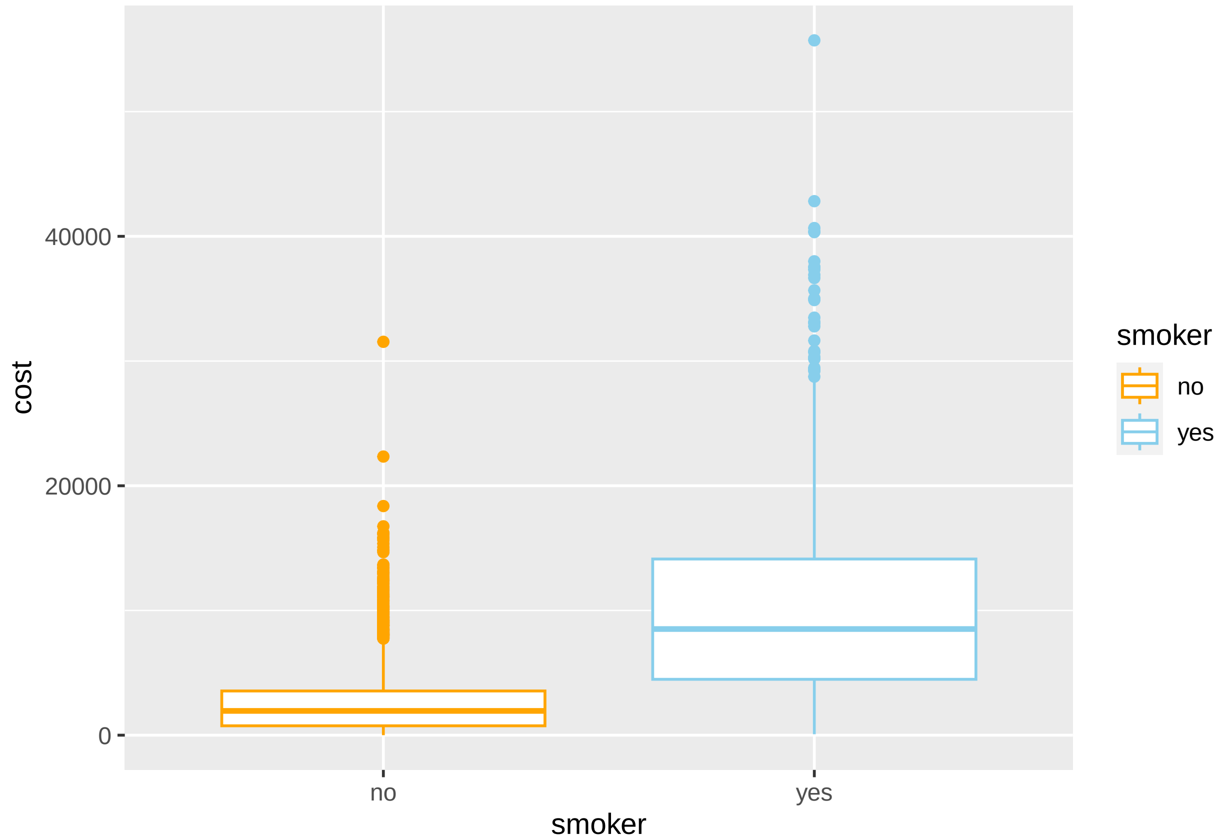



```
# Scatter plot of cost and age

ggplot(dataset)+aes(bmi,cost)+geom_point(col='darkgreen')+geom_smooth(method="lm", se=TRUE)
```

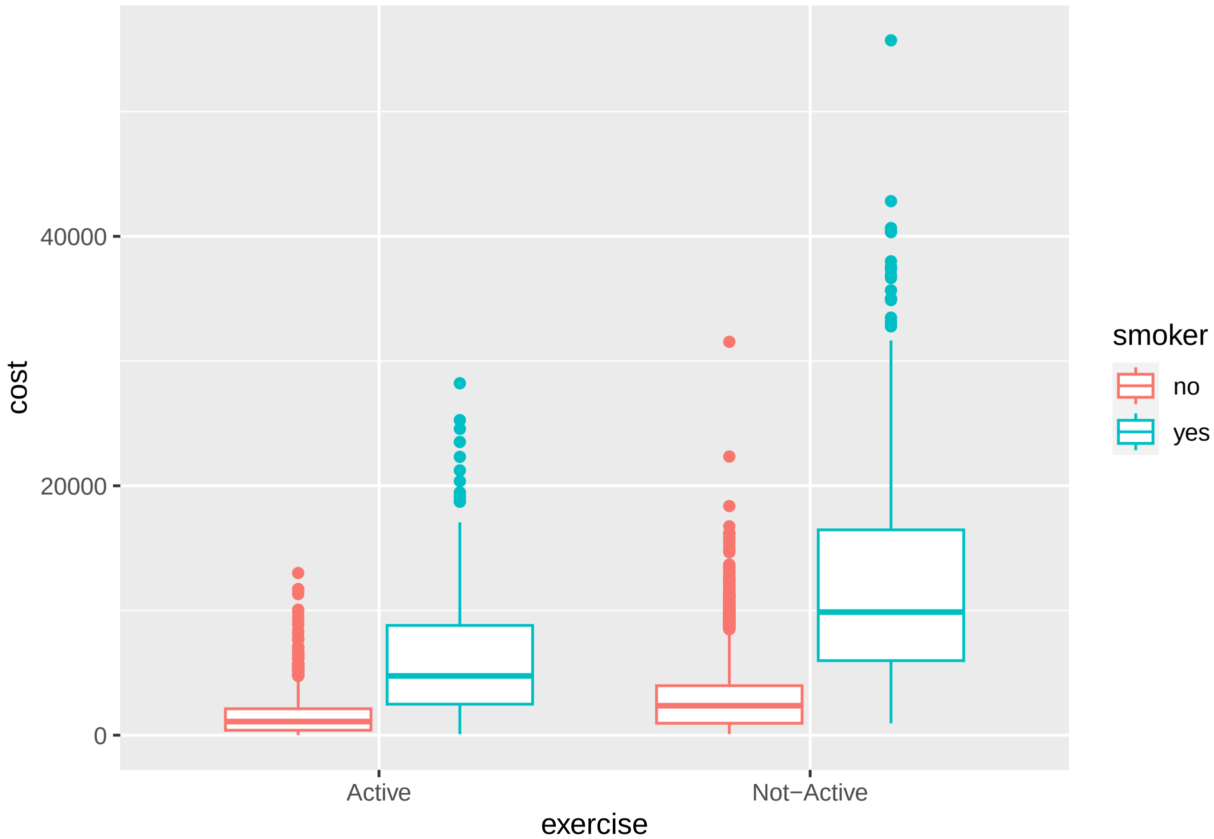
```
## `geom_smooth()` using formula = 'y ~ x'
```



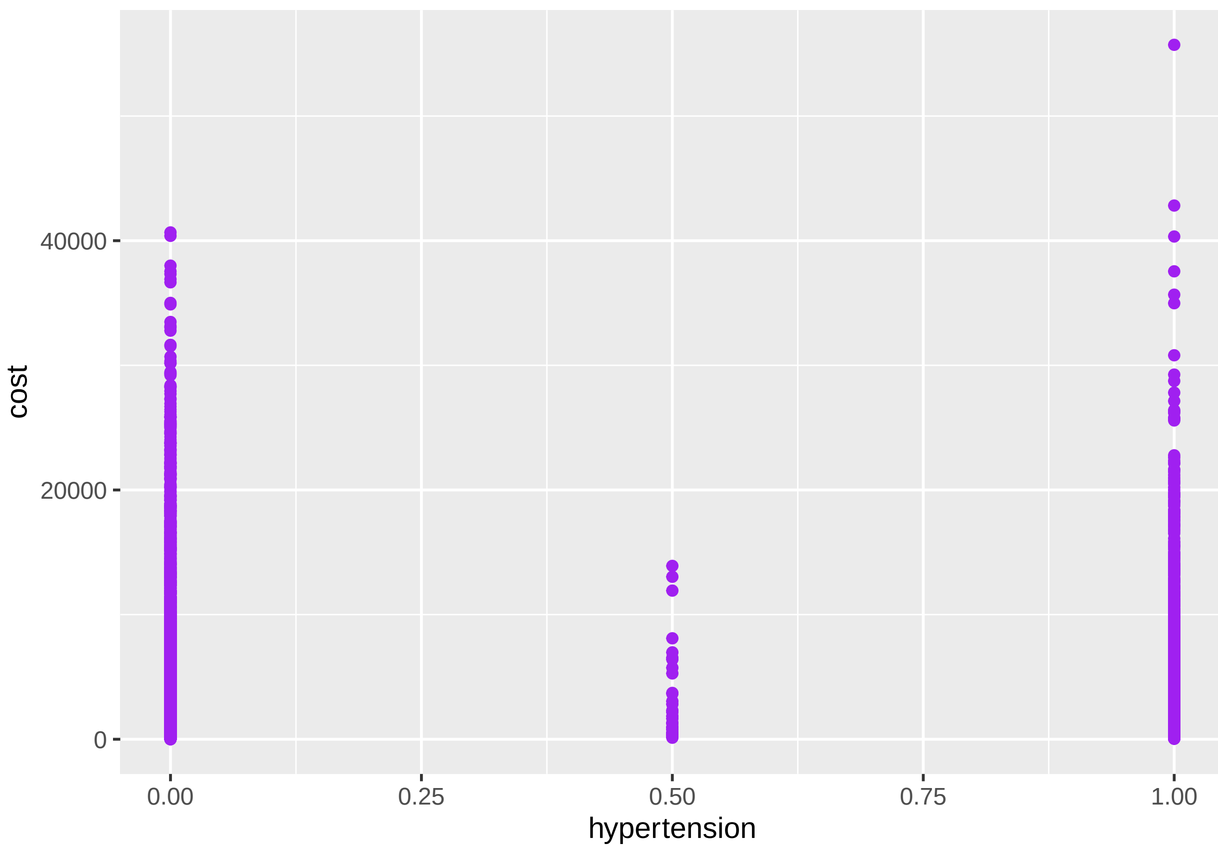

```
# Scatter plot of cost and bmi

ggplot(dataset)+aes(smoker,cost,color=smoker)+geom_boxplot()+scale_color_manual(values = c("orange","sky
```




```r
# Boxplot of cost and smoker

ggplot(dataset)+aes(exercise,cost,color=smoker)+geom_boxplot()
```




```
# Boxplot of cost and exercise

ggplot(dataset)+ aes(hypertension,cost) +geom_point(col='purple')
```




```
# Scatter plot of cost and hypertension
```

Once we have decided the mean of cost is the margin for the expensive but there are other variables which might affect expenses so we used group by function in the tidyverse library. The results show that the variables aren't making any big difference between the mean of the cost.

```
unique(dataset$location)
```

```
## [1] "CONNECTICUT"   "RHODE ISLAND"  "MASSACHUSETTS" "PENNSYLVANIA"
## [5] "MARYLAND"      "NEW JERSEY"    "NEW YORK"
```

```
mean(dataset$cost)
```

```
## [1] 4042.961
```

```
# created a new column called agecategory based on age range to find  the relatinoship between the
#vribles
dataset$agecategory <- ifelse(dataset$age %in% 15:19, "Teenagers", ifelse(dataset$age %in% 20:34, "youn
                                     ifelse(dataset$age %in% 35:54, "Middle_aged_adults",
                                            ifelse(dataset$age >= 55, "Senior_citizens", "Unknown"))))
library(tidyverse)
dataset %>% group_by(location) %>% summarize(cost=mean(cost))
```

```
## # A tibble: 7 x 2
##   location       cost
##   <chr>         <dbl>
## 1 CONNECTICUT   3848.
## 2 MARYLAND      3784.
## 3 MASSACHUSETTS 4268.
## 4 NEW JERSEY    3931.
## 5 NEW YORK      4662.
## 6 PENNSYLVANIA  4023.
## 7 RHODE ISLAND  4051.
```

```
dataset %>% group_by(education_level) %>% summarize(cost=mean(cost))
```

```
## # A tibble: 4 x 2
##   education_level    cost
##   <chr>             <dbl>
## 1 Bachelor          4037.
## 2 Master            3974.
## 3 No College Degree 4089.
## 4 PhD               4181.
```

```
dataset %>% group_by(married) %>% summarize(cost=mean(cost))
```

```
## # A tibble: 2 x 2
##   married      cost
##   <chr>       <dbl>
## 1 Married     4007.
## 2 Not_Married 4114.
```

```
dataset %>% group_by(agecategory) %>% summarize(cost=mean(cost))
```

```
## # A tibble: 4 x 2
##   agecategory         cost
##   <chr>              <dbl>
## 1 Middle_aged_adults 5055.
## 2 Senior_citizens    6031.
## 3 Teenagers          1836.
## 4 young_adults       2370.
```

Now , we created an Expensive_type variable where it has "Expensive" if the cost is greater than the mean of the cost else "Not-Expensive" in the cells.

```
dataset$Expensive_type <- ifelse(dataset$cost > mean(dataset$cost), "Expensive", "Not-Expensive")
table(dataset$Expensive_type)
```

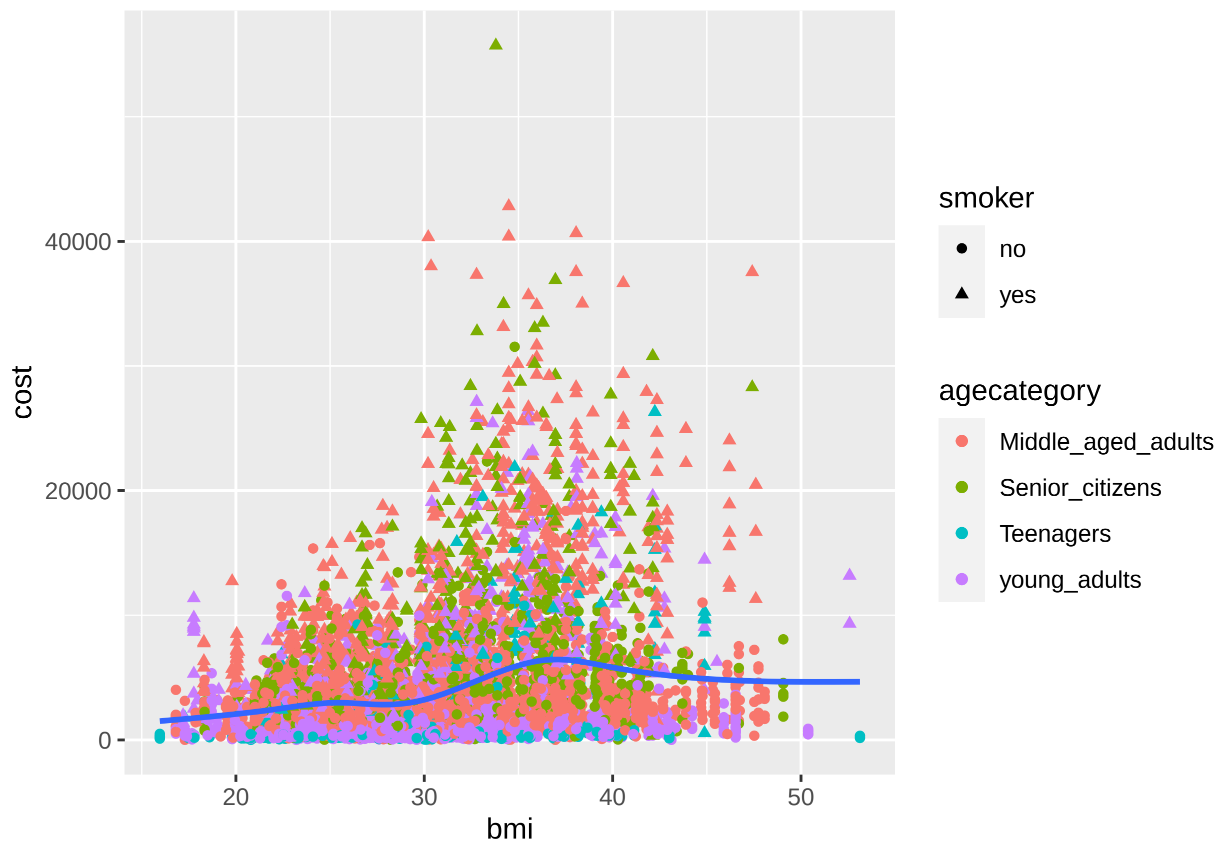
```
##
##     Expensive Not-Expensive
##          2360          5222
```

```r
# Creation of expensive and Not-expensive
```

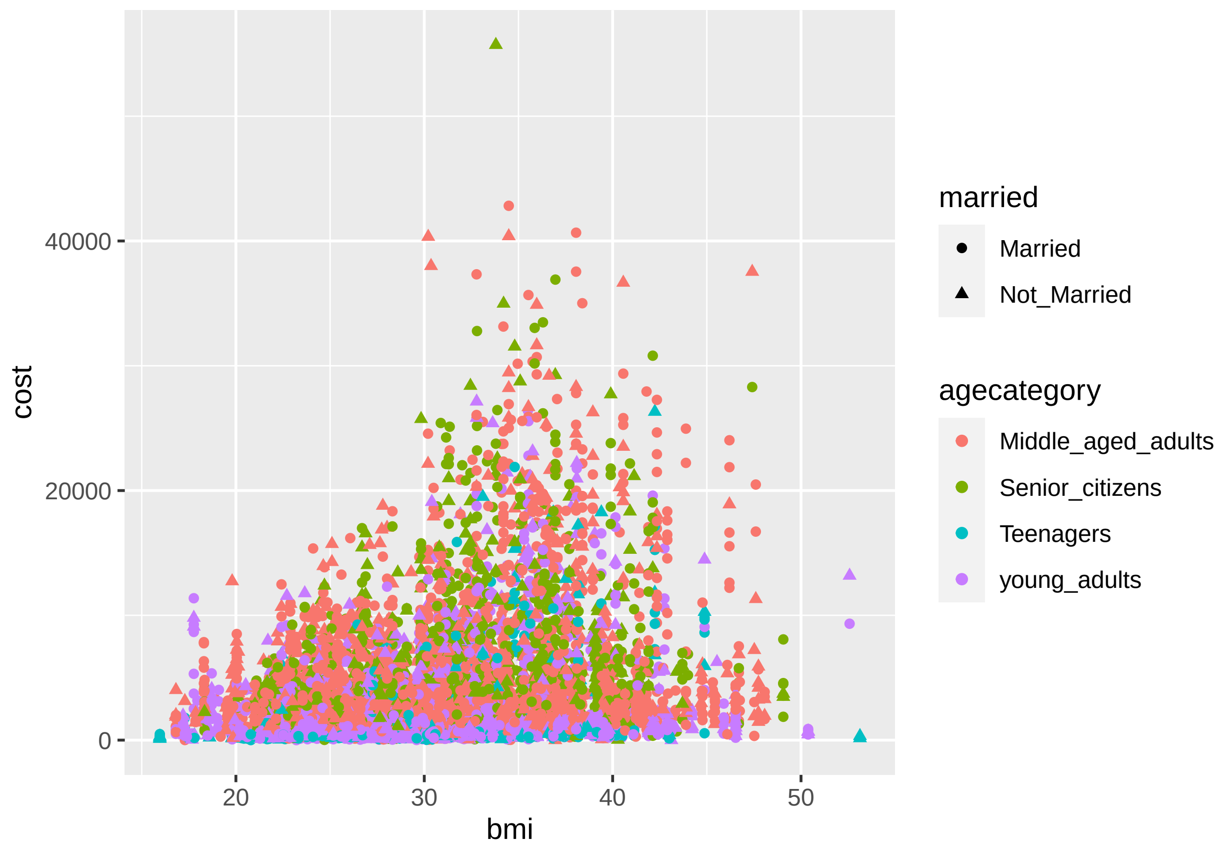
Now we created scatter plots with different variables to analyze which are key variable to determine the expenses for the health. we used ggplot function in the ggplot2 library.

```r
library(ggplot2)
ggplot(dataset, aes(x=bmi,y=cost)) +geom_point(aes(shape=smoker, color=agecategory))+ geom_smooth(se=FAI
```

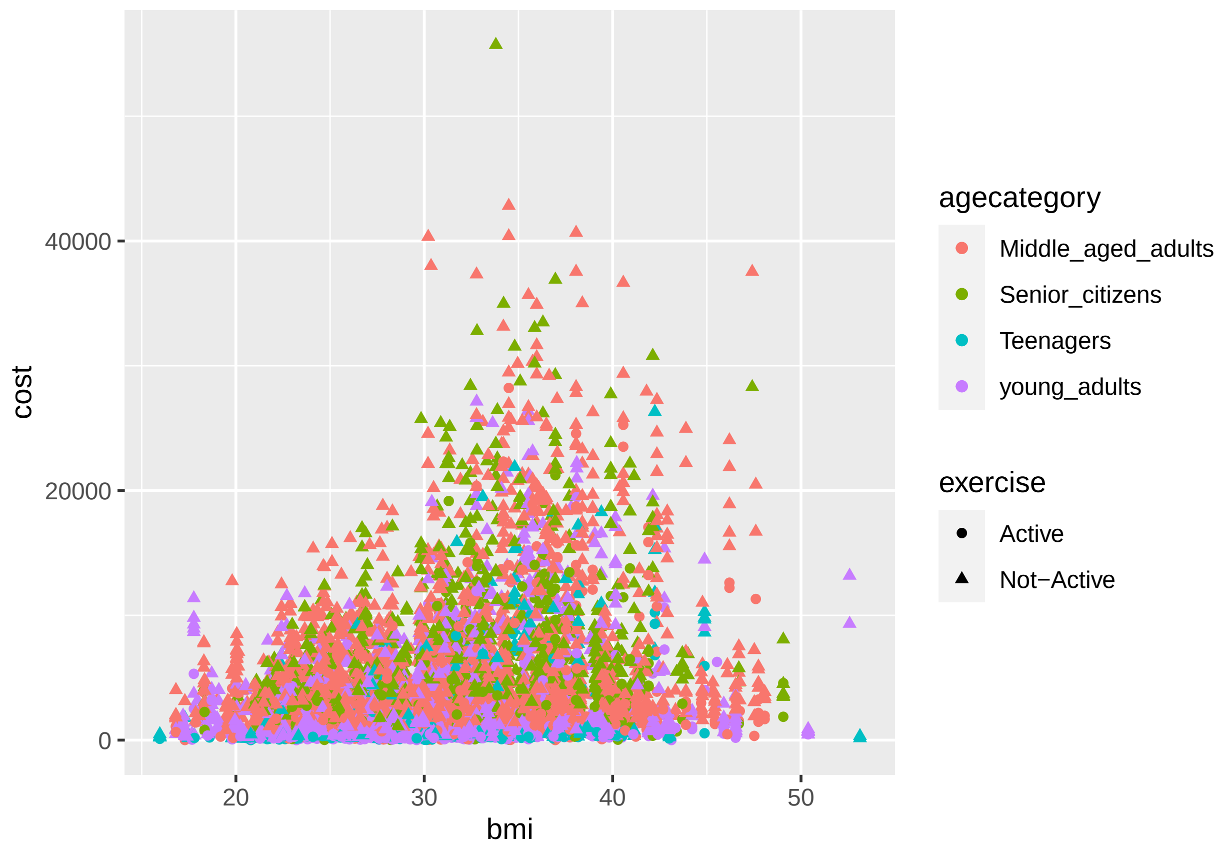
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



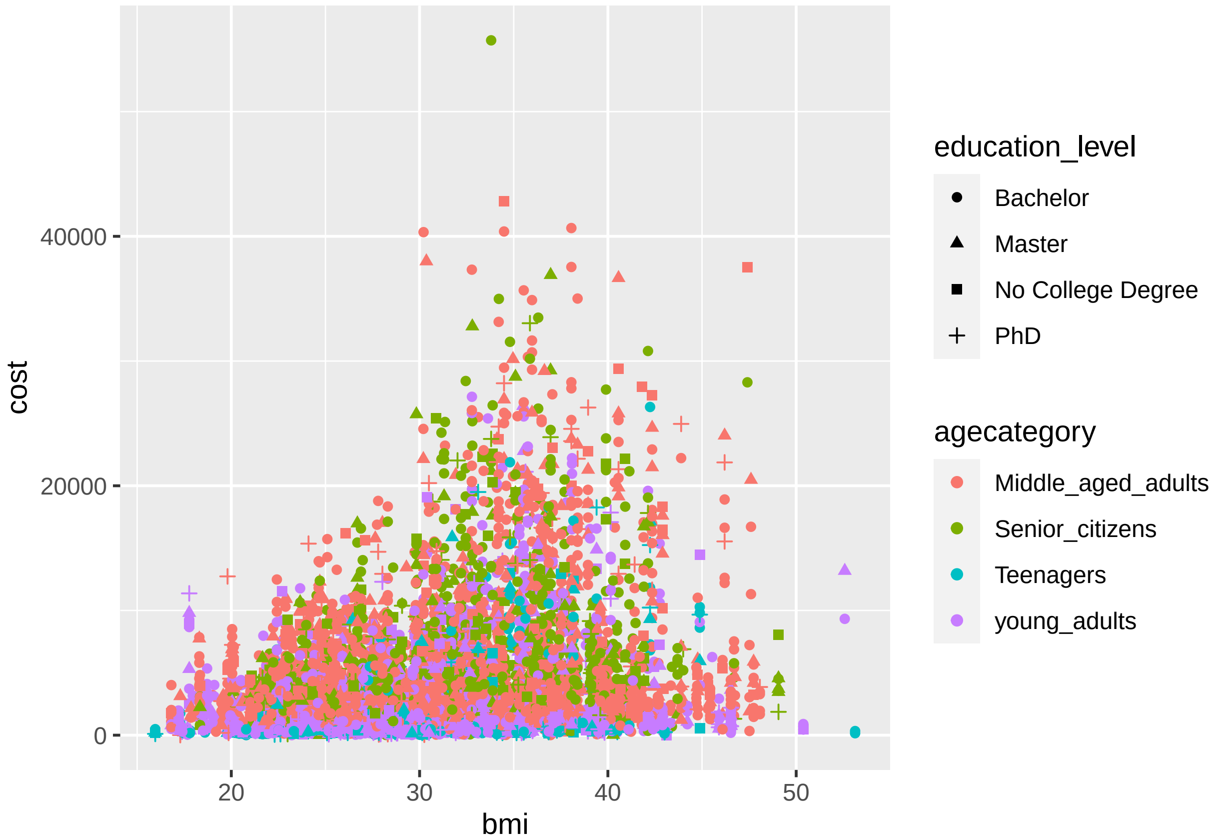

```r
ggplot(dataset, aes(x=bmi)) +geom_point(aes(y=cost ,shape=married, color=agecategory))
```




```
ggplot(dataset, aes(x=bmi)) +geom_point(aes(y=cost ,shape=exercise, color=agecategory))
```

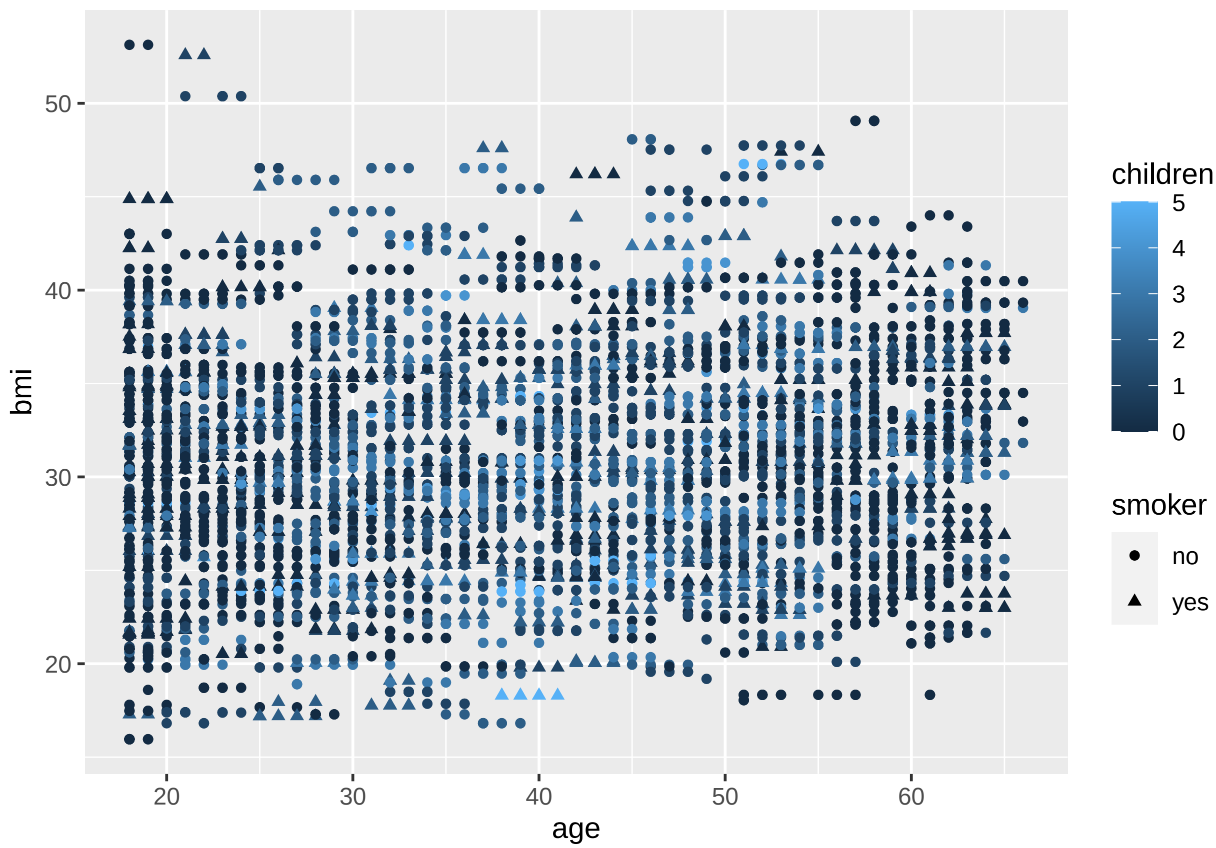



```
ggplot(dataset, aes(x=bmi)) +geom_point(aes(y=cost ,shape=education_level, color=agecategory))
```




```
ggplot(dataset, aes(x=age)) +geom_point(aes(y=bmi ,shape=yearly_physical, color=agecategory))
```


```
ggplot(dataset, aes(x=age)) +geom_point(aes(y=bmi ,shape=smoker, color=children))
```



To get more statistical information between the cost and other variables in the dataset we choose the linear regression model. In this model we used the datalm dataset where there are no chr data types.The resulting linear model is significant due its p-value is less then 0.05 and its r-squared value is 57.31%. By looking at the p-value of the variables we can determine which values are significant and these variable will be considered to the determine the expenses in the further models we used.

```
#LM - model ##datalm
lmout <- lm(cost ~ age+bmi+hypertension+smoker+location+exercise, data = datalm)

summary(lmout)
```

```
##
## Call:
## lm(formula = cost ~ age + bmi + hypertension + smoker + location +
##     exercise, data = datalm)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -12057  -1515   -370   1019  41766
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -8869.325    255.739 -34.681  < 2e-16 ***
## age                       103.681      2.633  39.380  < 2e-16 ***
## bmi                       180.510      6.244  28.911  < 2e-16 ***
## hypertension              347.105     93.085   3.729 0.000194 ***
```

```
## smokeryes              7677.609     93.766  81.880  < 2e-16 ***
## locationMARYLAND       -124.060    176.384  -0.703 0.481857
## locationMASSACHUSETTS    29.445    199.047   0.148 0.882402
## locationNEW JERSEY      128.307    195.226   0.657 0.511059
## locationNEW YORK        484.541    190.402   2.545 0.010953 *
## locationPENNSYLVANIA     16.987    140.450   0.121 0.903737
## locationRHODE ISLAND    128.754    178.829   0.720 0.471558
## exerciseNot-Active     2264.095     85.940  26.345  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3233 on 7570 degrees of freedom
## Multiple R-squared:  0.5703, Adjusted R-squared:  0.5697
## F-statistic: 913.3 on 11 and 7570 DF,  p-value: < 2.2e-16
```

Once the linear model showed which are the key variables that are affecting the cost. We can train the svm and tree bag model to predict the expensive type according to the independent variables. For the models we have to create a two sets . One of them will be used to train the model and the another one is used to test model. we used the caret library for the svm model and createDataPartition function is used to separate the dataset with p=0.62.

```
dataset$Expensive_type <- as.factor(dataset$Expensive_type)
###### using datalm dataframe
datalm$Expensive_type <- dataset$Expensive_type
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
# SVM MODEL
set.seed(6)
trainList <- createDataPartition(y=datalm$Expensive_type,p=.62,list=FALSE)
trainSet <- datalm[trainList,]
testSet <- datalm[-trainList,]
svmmodel <- train(Expensive_type~age+bmi+hypertension+smoker+location+exercise , data = trainSet, metho
```

The following code will test the svm model using testSet.

```
svmpredout <- predict(svmmodel,newdata=testSet)
```

we created the confusion Matrix from the testing results so that we can how much accuracy and sensitivity this model has.

```
confMatrix <- table(svmpredout,testSet$Expensive_type)
confMatrix
```

```
##
## svmpredout      Expensive Not-Expensive
##   Expensive           564           114
##   Not-Expensive       332          1870
```

```
errorRate <- (sum(confMatrix) - sum(diag(confMatrix)))/sum(confMatrix)
errorRate
```

```
## [1] 0.1548611
```

```
accuracy <- 1-errorRate
accuracy
```

```
## [1] 0.8451389
```

This confusionMatrix function will gives use the accuracy without any calculation.

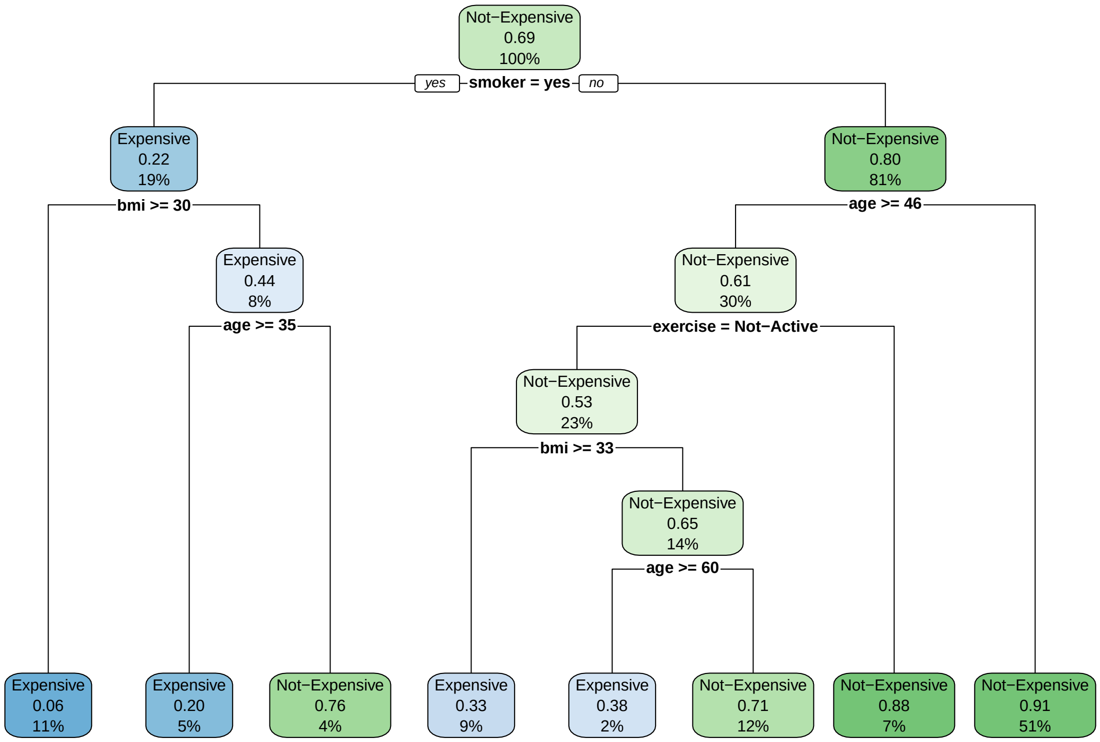
```
confusionMatrix(svmpredout,testSet$Expensive_type)
```

```
## Confusion Matrix and Statistics
##
##                  Reference
## Prediction      Expensive Not-Expensive
##   Expensive           564           114
##   Not-Expensive       332          1870
##
##                Accuracy : 0.8451
##                  95% CI : (0.8314, 0.8582)
##     No Information Rate : 0.6889
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6129
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.6295
##             Specificity : 0.9425
##          Pos Pred Value : 0.8319
##          Neg Pred Value : 0.8492
##              Prevalence : 0.3111
##          Detection Rate : 0.1958
##    Detection Prevalence : 0.2354
##       Balanced Accuracy : 0.7860
##
##        'Positive' Class : Expensive
##
```

```
# the accuracy is 84.51% and sensitivity is 62.95%
```

We used the rpart and e1071 library for thr tree bag model. Similar to the previous model we used two different sets. One of it is to train the model and the other is to test the model.

```r
###tree bag model
library(rpart)
library(e1071)
tree <- train(Expensive_type~age+bmi+hypertension+smoker+location+exercise , data = trainSet, method="t
treerpart <- rpart(Expensive_type~age+bmi+hypertension+smoker+location+exercise , data = trainSet, meth
library(rpart.plot)
rpart.plot(treerpart)
```




```r
# Checking accuracy with confusion matrix
treePred <- predict(tree,newdata=testSet)
confusion <- table(treePred,testSet$Expensive_type)
confMatrix <- table(treePred,testSet$Expensive_type)
confMatrix
```

```
##
## treePred        Expensive Not-Expensive
##   Expensive           627           196
##   Not-Expensive       269          1788
```

```r
errorRate <- (sum(confMatrix) - sum(diag(confMatrix)))/sum(confMatrix)
errorRate
```

```
## [1] 0.1614583
```

```r
accuracy <- 1-errorRate
accuracy
```

```
## [1] 0.8385417
```

```r
confusionMatrix(treePred,testSet$Expensive_type)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Expensive Not-Expensive
##   Expensive           627           196
##   Not-Expensive       269          1788
##
##                Accuracy : 0.8385
##                  95% CI : (0.8246, 0.8518)
##     No Information Rate : 0.6889
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6147
##
##  Mcnemar's Test P-Value : 0.000841
##
##             Sensitivity : 0.6998
##             Specificity : 0.9012
##          Pos Pred Value : 0.7618
##          Neg Pred Value : 0.8692
##              Prevalence : 0.3111
##          Detection Rate : 0.2177
##    Detection Prevalence : 0.2858
##       Balanced Accuracy : 0.8005
##
##        'Positive' Class : Expensive
##
```

```r
# the accuracy is 83.85% and sensitivity is 69.98%
```

We also thought to run the data through transaction model to check which type of variables have the most effect on the expensive type. So, we converted the dataset into transaction form and stored it in datasetr vector. All the required functions are stored in the arules and rulesviz library. We used the itemFrequencyPlot and itemFrequency to get to know all the transactions in the datasetr vecotr.

```r
#### transactions
library(arules);library(arulesViz)
```

```
## Loading required package: Matrix
```

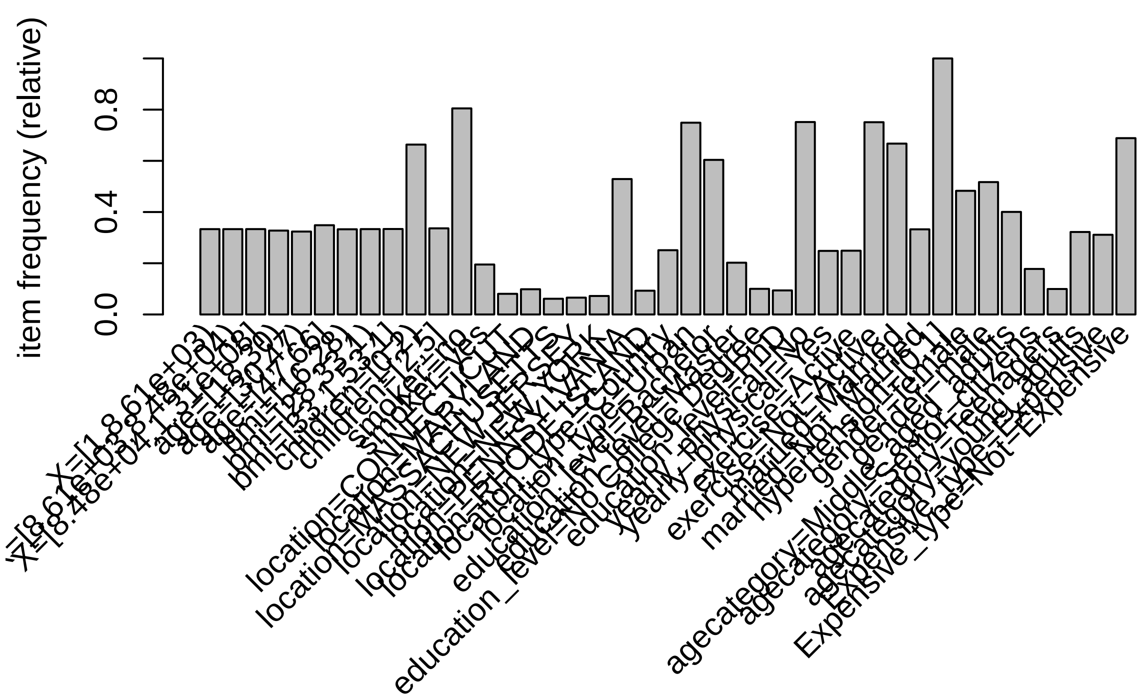
```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

##
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
datasetr <- as(dataset[,c(-1 -13)],"transactions")
```

```
## Warning: Column(s) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 not logical or
## factor. Applying default discretization (see '? discretizeDF').

## Warning in discretize(x = c(0L, 1L, 3L, 0L, 0L, 1L, 2L, 0L, 0L, 0L, 0L, : The calculated breaks are:
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for details.

## Warning in discretize(x = c(0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, : The calculated breaks are
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for details.
```

```
itemFrequencyPlot(datasetr)
```



```r
itemFrequency(datasetr)
```

```
##                      X=[1,8.61e+03)           X=[8.61e+03,8.48e+04)
##                          0.33328937                      0.33328937
##               X=[8.48e+04,1.31e+08]                     age=[18,30)
##                          0.33342126                      0.32761804
##                         age=[30,47)                     age=[47,66]
##                          0.32379319                      0.34858876
##                         bmi=[16,28)                   bmi=[28,33.1)
##                          0.33276180                      0.33342126
##                       bmi=[33.1,53.1]                  children=[0,2)
##                          0.33381693                      0.66354524
##                      children=[2,5]                       smoker=no
##                          0.33645476                      0.80493274
##                          smoker=yes             location=CONNECTICUT
##                          0.19506726                      0.08058560
##                   location=MARYLAND          location=MASSACHUSETTS
##                          0.09852282                      0.06132946
##                 location=NEW JERSEY               location=NEW YORK
##                          0.06568188                      0.07214455
##               location=PENNSYLVANIA           location=RHODE ISLAND
##                          0.52888420                      0.09285149
##               location_type=Country             location_type=Urban
##                          0.25098918                      0.74901082
##            education_level=Bachelor          education_level=Master
##                          0.60379847                      0.20218940
## education_level=No College Degree             education_level=PhD
##                          0.10010551                      0.09390662
##                  yearly_physical=No             yearly_physical=Yes
##                          0.75164864                      0.24835136
##                     exercise=Active             exercise=Not-Active
##                          0.24901082                      0.75098918
##                     married=Married             married=Not_Married
##                          0.66737009                      0.33262991
##                  hypertension=[0,1]                   gender=female
##                          1.00000000                      0.48298602
##                         gender=male   agecategory=Middle_aged_adults
##                          0.51701398                      0.40055394
##        agecategory=Senior_citizens           agecategory=Teenagers
##                          0.17778950                      0.09944606
##           agecategory=young_adults          Expensive_type=Expensive
##                          0.32221050                      0.31126352
##       Expensive_type=Not-Expensive
##                          0.68873648
```

The apriori function with the supp = 0.08, conf = 0.8, lhs will be defulath which mean everything else except the rhs and rhs is set to “Expensive_type=Expensive”. By running this function we will get the all the transactions with only RHS in “Expensive_type=Expensive”. To look at all the transactions we used insect

```r
##datatset - all
####### important
rulesetb <- apriori(datasetr, parameter = list(supp = 0.08, conf = 0.8), appearance = list(default="lhs
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.08      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 606
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[41 item(s), 7582 transaction(s)] done [0.01s].
## sorting and recoding items ... [38 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.13s].
## writing ... [10 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(rulesetb)
```

```
##      lhs                        rhs                           support confidence   coverage     lift c
## [1]  {smoker=yes,
##       gender=male}          => {Expensive_type=Expensive} 0.09562121  0.8155231 0.11725138 2.620041
## [2]  {smoker=yes,
##       exercise=Not-Active}  => {Expensive_type=Expensive} 0.12371406  0.8637201 0.14323398 2.774884
## [3]  {smoker=yes,
##       hypertension=[0,1],
##       gender=male}          => {Expensive_type=Expensive} 0.09562121  0.8155231 0.11725138 2.620041
## [4]  {smoker=yes,
##       exercise=Not-Active,
##       married=Married}      => {Expensive_type=Expensive} 0.08335532  0.8657534 0.09628066 2.781416
## [5]  {smoker=yes,
##       location_type=Urban,
##       exercise=Not-Active}  => {Expensive_type=Expensive} 0.09364284  0.8700980 0.10762332 2.795374
## [6]  {smoker=yes,
##       yearly_physical=No,
##       exercise=Not-Active}  => {Expensive_type=Expensive} 0.09311527  0.8526570 0.10920601 2.739341
## [7]  {smoker=yes,
##       exercise=Not-Active,
##       hypertension=[0,1]}   => {Expensive_type=Expensive} 0.12371406  0.8637201 0.14323398 2.774884
## [8]  {smoker=yes,
##       exercise=Not-Active,
##       married=Married,
##       hypertension=[0,1]}   => {Expensive_type=Expensive} 0.08335532  0.8657534 0.09628066 2.781416
## [9]  {smoker=yes,
##       location_type=Urban,
##       exercise=Not-Active,
##       hypertension=[0,1]}   => {Expensive_type=Expensive} 0.09364284  0.8700980 0.10762332 2.795374
## [10] {smoker=yes,
##       yearly_physical=No,
```

```
##       exercise=Not-Active,
##       hypertension=[0,1]}   => {Expensive_type=Expensive} 0.09311527  0.8526570 0.10920601 2.739341
```

```r
##dataset - smoker, yearly_physical,exercise,bmi,hypertension
ruleseta <- apriori(datasetr, parameter = list(supp = 0.05, conf = 0.7), appearance = list(lhs = c("smol
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5    0.05      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 379
##
## set item appearances ...[11 item(s)] done [0.00s].
## set transactions ...[11 item(s), 7582 transaction(s)] done [0.01s].
## sorting and recoding items ... [11 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [14 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
inspect(ruleseta)
```

```
##      lhs                          rhs                          support confidence   coverage     lift
## [1]  {smoker=yes}             => {Expensive_type=Expensive} 0.15338961  0.7863421 0.19506726 2.526291
## [2]  {bmi=[33.1,53.1],
##       smoker=yes}             => {Expensive_type=Expensive} 0.06831970  0.9628253 0.07095753 3.093280
## [3]  {smoker=yes,
##       yearly_physical=No}    => {Expensive_type=Expensive} 0.11474545  0.7719610 0.14864152 2.480088
## [4]  {smoker=yes,
##       exercise=Not-Active}   => {Expensive_type=Expensive} 0.12371406  0.8637201 0.14323398 2.774884
## [5]  {smoker=yes,
##       hypertension=[0,1]}    => {Expensive_type=Expensive} 0.15338961  0.7863421 0.19506726 2.526291
## [6]  {bmi=[33.1,53.1],
##       smoker=yes,
##       yearly_physical=No}    => {Expensive_type=Expensive} 0.05249275  0.9567308 0.05486679 3.073700
## [7]  {bmi=[33.1,53.1],
##       smoker=yes,
##       exercise=Not-Active}   => {Expensive_type=Expensive} 0.05420733  1.0000000 0.05420733 3.212712
## [8]  {bmi=[33.1,53.1],
##       smoker=yes,
##       hypertension=[0,1]}    => {Expensive_type=Expensive} 0.06831970  0.9628253 0.07095753 3.093280
## [9]  {smoker=yes,
##       yearly_physical=No,
##       exercise=Not-Active}   => {Expensive_type=Expensive} 0.09311527  0.8526570 0.10920601 2.739341
## [10] {smoker=yes,
```

```
##       yearly_physical=No,
##       hypertension=[0,1]}   => {Expensive_type=Expensive} 0.11474545  0.7719610 0.14864152 2.480088
## [11] {smoker=yes,
##       exercise=Not-Active,
##       hypertension=[0,1]}   => {Expensive_type=Expensive} 0.12371406  0.8637201 0.14323398 2.774884
## [12] {bmi=[33.1,53.1],
##       smoker=yes,
##       yearly_physical=No,
##       hypertension=[0,1]}   => {Expensive_type=Expensive} 0.05249275  0.9567308 0.05486679 3.073700
## [13] {bmi=[33.1,53.1],
##       smoker=yes,
##       exercise=Not-Active,
##       hypertension=[0,1]}   => {Expensive_type=Expensive} 0.05420733  1.0000000 0.05420733 3.212712
## [14] {smoker=yes,
##       yearly_physical=No,
##       exercise=Not-Active,
##       hypertension=[0,1]}   => {Expensive_type=Expensive} 0.09311527  0.8526570 0.10920601 2.739341
```

```r
##datatset - childrena,agecategory,married,educationlevel
rulesetb <- apriori(datasetr, parameter = list(supp = 0.005, conf = 0.55), appearance = list(lhs = c("c
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.55    0.1    1 none FALSE            TRUE       5   0.005      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 37
##
## set item appearances ...[13 item(s)] done [0.00s].
## set transactions ...[13 item(s), 7582 transaction(s)] done [0.00s].
## sorting and recoding items ... [13 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [16 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
inspect(rulesetb)
```

```
##      lhs                                       rhs                           support confidence    co
## [1]  {education_level=PhD,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.010683197  0.5869565 0.018
## [2]  {education_level=No College Degree,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.011870219  0.6122449 0.019
## [3]  {children=[2,5],
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.024795568  0.6482759 0.038
## [4]  {children=[0,2),
```

```
##       education_level=PhD,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.007517805  0.5588235 0.0134
## [5]  {education_level=PhD,
##       married=Married,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.007517805  0.5757576 0.0130
## [6]  {children=[0,2),
##       education_level=No College Degree,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.009759958  0.5873016 0.0160
## [7]  {education_level=No College Degree,
##       married=Married,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.008704827  0.6055046 0.0143
## [8]  {education_level=Master,
##       married=Married,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.013848589  0.5706522 0.0242
## [9]  {children=[2,5],
##       married=Not_Married,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.008572936  0.6565657 0.0130
## [10] {children=[2,5],
##       education_level=Bachelor,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.015299393  0.6408840 0.0238
## [11] {children=[2,5],
##       married=Married,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.016222633  0.6439791 0.0251
## [12] {children=[0,2),
##       education_level=PhD,
##       married=Married,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.005671327  0.5733333 0.0098
## [13] {children=[0,2),
##       education_level=No College Degree,
##       married=Married,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.007122131  0.5869565 0.0121
## [14] {children=[0,2),
##       education_level=Master,
##       married=Married,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.011342654  0.5695364 0.0199
## [15] {children=[2,5],
##       education_level=Bachelor,
##       married=Not_Married,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.005011870  0.5937500 0.0084
## [16] {children=[2,5],
##       education_level=Bachelor,
##       married=Married,
##       agecategory=Senior_citizens}        => {Expensive_type=Expensive} 0.010287523  0.6666667 0.0154
```