Forecasting High-Cost Healthcare Clients

By Tarun Talreja, Yash Wadhawe, Maulik Ramnani, Dev Jindani, Achala Rao, and Aditya Kulkarni | Group 3

Project Goal:

1. The overall goal of the case is to provide actionable insight, based on the data available, as well as accurately predict which people (customers) will be expensive.
2. The data set contains healthcare cost information from an HMO (Health Management Organization). Each row in the data set represents a person.
3. The goal for our team is to understand the key drivers for why some people are more expensive (i.e. require more health care), as well as predict which people will be expensive (in terms of health care costs).

   Hence, we have two goals:

   1. Predict people who will spend a lot of money on health care next year (i.e., which people will have high healthcare costs).
   2. Provide actionable insight to HMO, in terms of how to lower their total health care costs, by providing a specific recommendation on how to lower health care costs.

Project Deliverables:

The analysis includes exploratory analysis (EDA) (e.g. histograms, scatter plots), mapping visualizations and several machine learning techniques.

Data:

The data file can be located at: https://intro-datascience.s3.us-east-2.amazonaws.com/HMO_data.csv (https://intro-datascience.s3.us-east-2.amazonaws.com/HMO_data.csv)

```
# Importing the necessary library
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.3
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
## Warning: package 'forcats' was built under R version 4.2.3
```

```
## Warning: package 'lubridate' was built under R version 4.2.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.1     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.2     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to
become errors
```

```
# Reading in a CSV file from a remote location and storing it in a data frame called datafile
datafile <- read.csv("https://intro-datascience.s3.us-east-2.amazonaws.com/HMO_data.csv")

# Looking at the dataset
str(datafile)
```

```
## '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 ...
```

Summarizing the Data and checking the statistics of columns in the data frame

```
# Using summary() function to get statistical descriptions of all the columns
summary(datafile)
```

```
##        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
##
```

Summary of the Data Frame

Age:

- Mean - 39
- Median - 39
- 1st Quantile - 26
- 3rd Quantile - 51

BMI:

- Mean - 30.8

- Median - 30.5
- 1st Quantile - 26.6
- 3rd Quantile - 35.77

Cost:

- Mean - 4043
- Median - 2500
- 1st Quantile - 970
- 3rd Quantile - 4775

Children:

- Mean - 1
- Median - 1

Data Cleaning

Our team has discovered a few missing values within the dataset. To begin with, we'll determine which columns contain NAs. Afterward, we will decide on an appropriate course of action, either substituting the missing values with the column's mean or eliminating rows containing NAs, depending on the most suitable strategy.

```
# Using is.na() on every column to check for NA values

sum(is.na(datafile$age))
```

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

```
sum(is.na(datafile$bmi))
```

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

```
sum(is.na(datafile$children))
```

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

```
sum(is.na(datafile$smoker))
```

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

```
sum(is.na(datafile$location))
```

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

```
sum(is.na(datafile$location_type))
```

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

```
sum(is.na(datafile$education_level))
```

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

```
sum(is.na(datafile$yearly_physical))
```

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

```
sum(is.na(datafile$exercise))
```

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

```
sum(is.na(datafile$married))
```

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

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

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

```
sum(is.na(datafile$gender))
```

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

```
sum(is.na(datafile$cost))
```

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

As per our code, there are missing values present in BMI and hypertension columns. Hypertension is a binary data type column, we cannot replace missing values with mean or na_interpolation. So, the best strategy is to delete the rows.

BMI, however, is a continuous value type column, we can use na_interpolation to substitute the NAs present in the column.

```
# Importing relevant library
library(imputeTS)
```

```
## Warning: package 'imputeTS' was built under R version 4.2.3
```

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

```
# Using na_interpolation to replace the NAs in BMI
datafile$bmi <- na_interpolation(datafile$bmi, option = "linear")

# Deleting all rows where hypertension has NA
datafile <- datafile[!is.na(datafile$hypertension),]

# Checking if NAs are removed or replaced

sum(is.na(datafile$bmi))
```

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

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

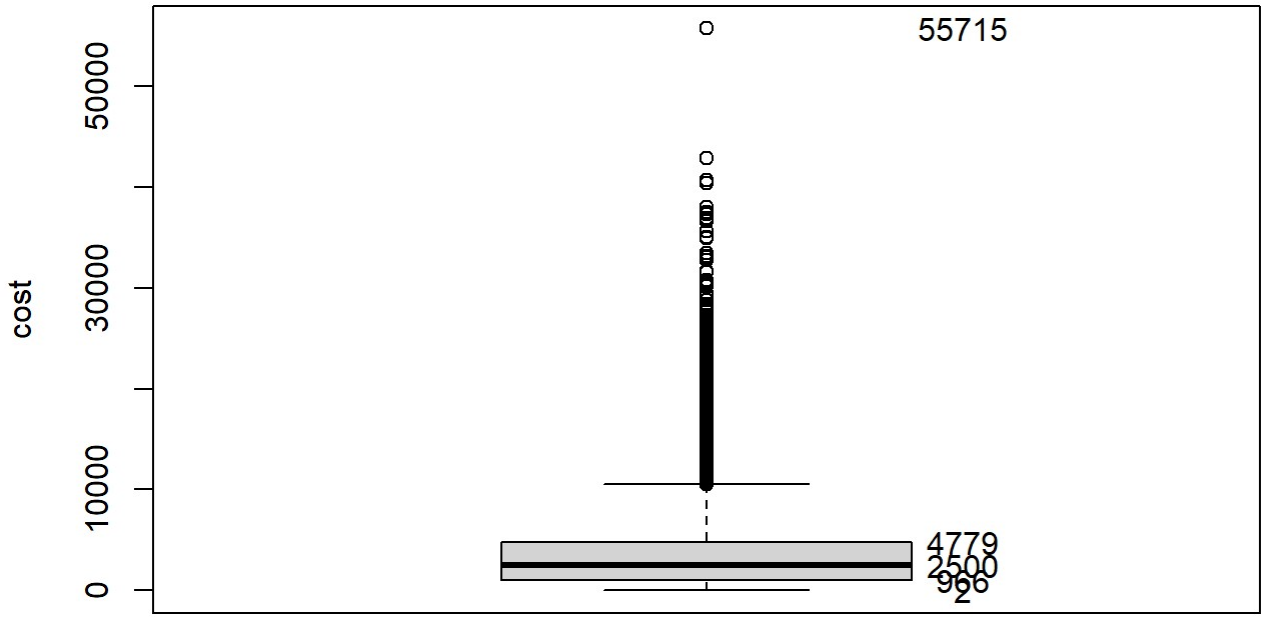
```
## [1] 0
```

```
# There are no NAs now, data seems to be ok

# Creating a safe copy just in case of any trouble
datafile_backup <- datafile
```

Categorizing Expensive / Not Expensive

In order to determine the threshold at which health insurance becomes too expensive for the healthcare company to cover, we should examine the distribution of costs in the available data using tools such as a box plot or histogram. Additionally, we should calculate the mean, median, and range of the cost data, as well as the quantile values, to gain a comprehensive understanding of the cost spread.

```
boxplot(datafile$cost,
  ylab = "cost",
  main = "Boxplot of healthcare cost"
)
text(y=fivenum(datafile$cost),labels=fivenum(datafile$cost),x=1.25)
```



The outliers seems to start from the cost value of 10,000

```
quantile(datafile$cost, probs = c(0.25,0.5,0.75,1))
```

```
##      25%      50%      75%     100%
##   966.50  2500.00  4778.75 55715.00
```

```
# The values between 75th% and 100% have a huge difference

quantile(datafile$cost,probs = seq(from=0.7,to=1,by=0.05))
```

```
##      70%      75%      80%      85%      90%      95%     100%
##  4168.10  4778.75  5789.40  7243.00  9630.90 14362.10 55715.00
```

```
# Huge jump in the value appears to rise exponentially after the 75th quantile

mean(datafile$cost)
```

```
## [1] 4049.492
```

```
# the mean is around the 70th quantile

range(datafile$cost)
```

```
## [1]     2 55715
```

So it is safe to decide the cap cost at 75th quantile or $4778. Now, we add a binary column for expensive and not expensive customers.

```
datafile$expensive <- datafile$cost > 4778

# Saving this dataframe to duplicate data
datafile_backup <- datafile
```

Grouping BMI and Age

```
#Groups for bmi and age
min(datafile$age) #18
```

```
## [1] 18
```

```
max(datafile$age) #66
```

```
## [1] 66
```

```
datafile$age_grouped <- cut(datafile$age, breaks = seq(10,70,10)) #from 10, to 70, 10 width
table((datafile$age_grouped))
```

```
##
## (10,20] (20,30] (30,40] (40,50] (50,60] (60,70]
##     986    1615    1391    1525    1471     514
```

```
min(datafile$bmi) #15.96
```

```
## [1] 15.96
```

```
max(datafile$bmi) #53.13
```

```
## [1] 53.13
```

```
datafile$bmi_grouped <- cut(datafile$bmi, breaks = seq(15,60,10)) #from 15, to 60, 10 width
table(datafile$bmi_grouped)
```

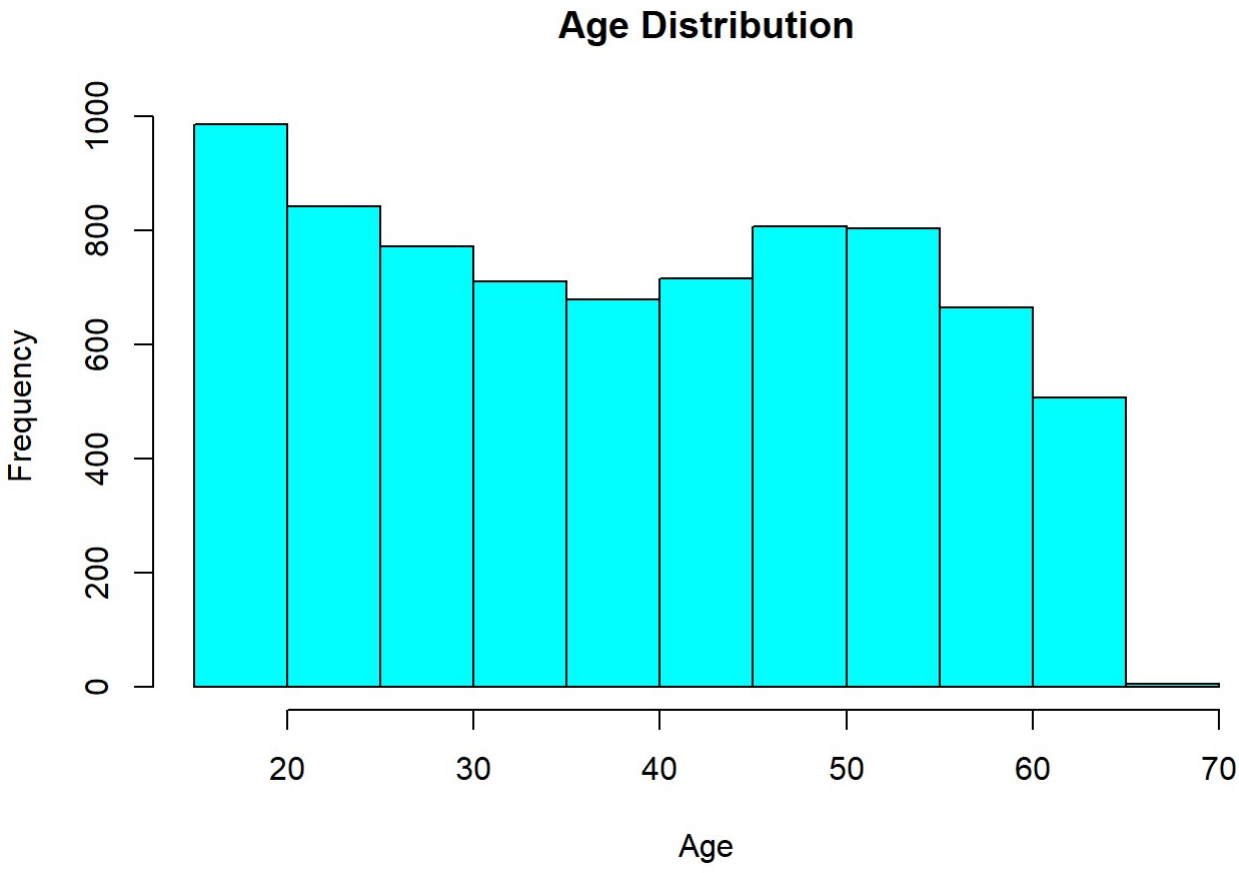
```
##
## (15,25] (25,35] (35,45] (45,55]
##    1290    4427    1674     111
```

```
# Backing up the data
datafile_backup <- datafile
```

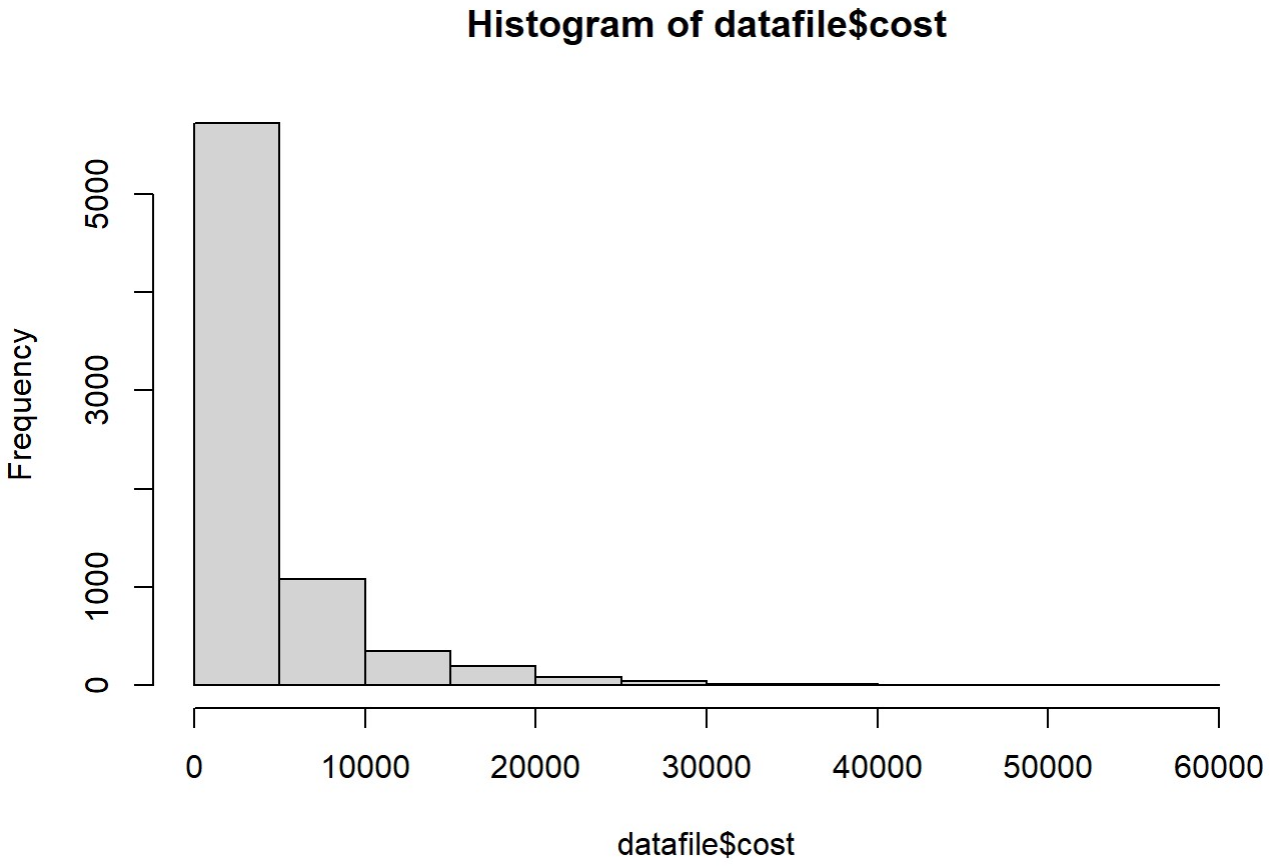
Exploratory Data Analysis

Now, we can try visualizing the chart and find any relationships between the cost with each attribute

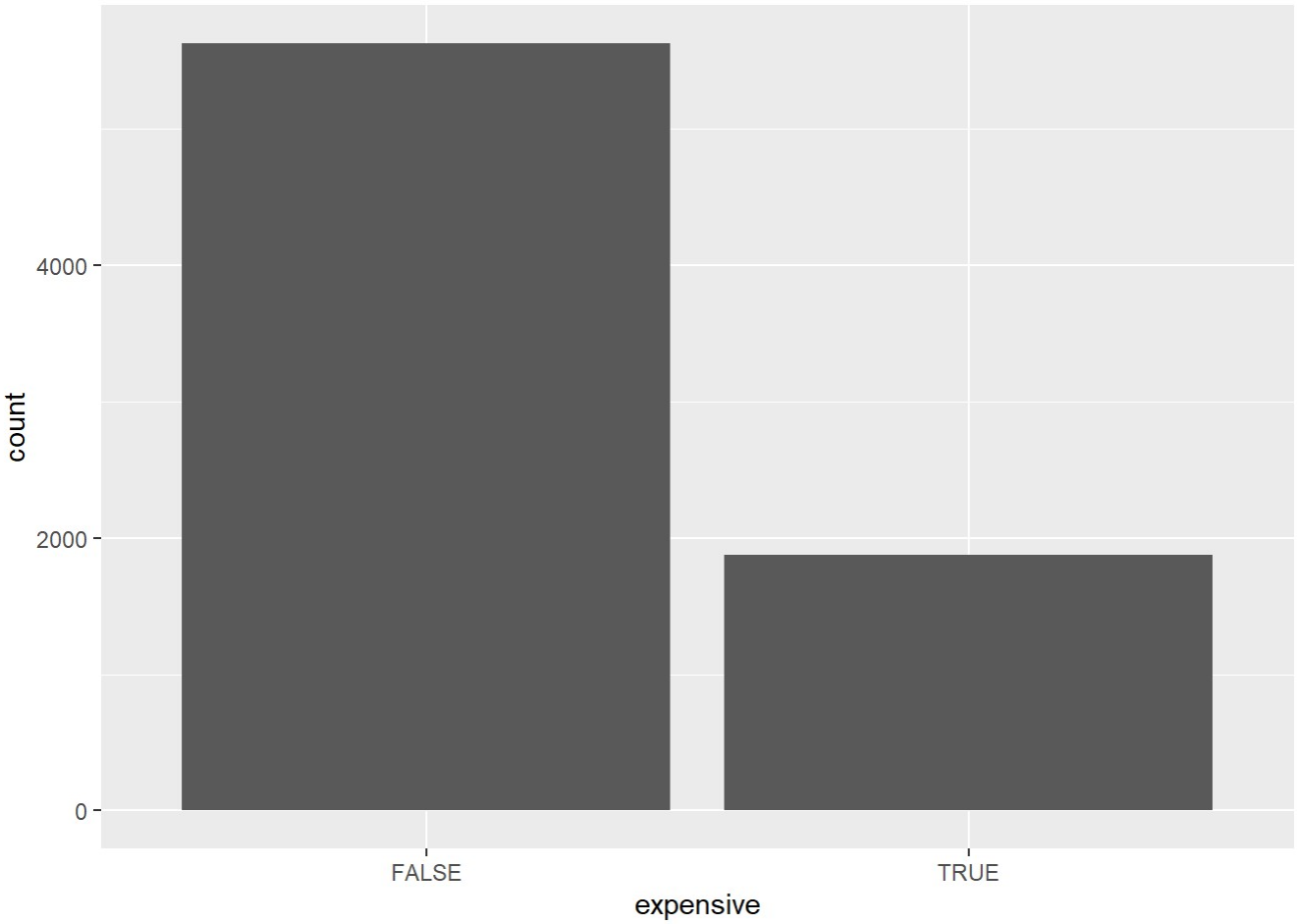
```
#distribution of genders
library(ggplot2)
hist(datafile$age, main = "Age Distribution", xlab = "Age", col = "cyan")
```




```
#Visualisation of cost
hist(datafile$cost)
```




```
#visualize distribution of expensive
plot_expensive <- ggplot(datafile, Beside = TRUE, aes(x=expensive)) + geom_bar()
plot_expensive
```




```
# Graphing Scatter plots to understand correlation of factors

# Importing the relevant library
library(ggplot2)

# age vs cost (smoker)
ggplot(data=datafile_backup, aes(x=age, y=cost,colour = smoker)) + geom_point() +
  geom_smooth(method = "lm")
```

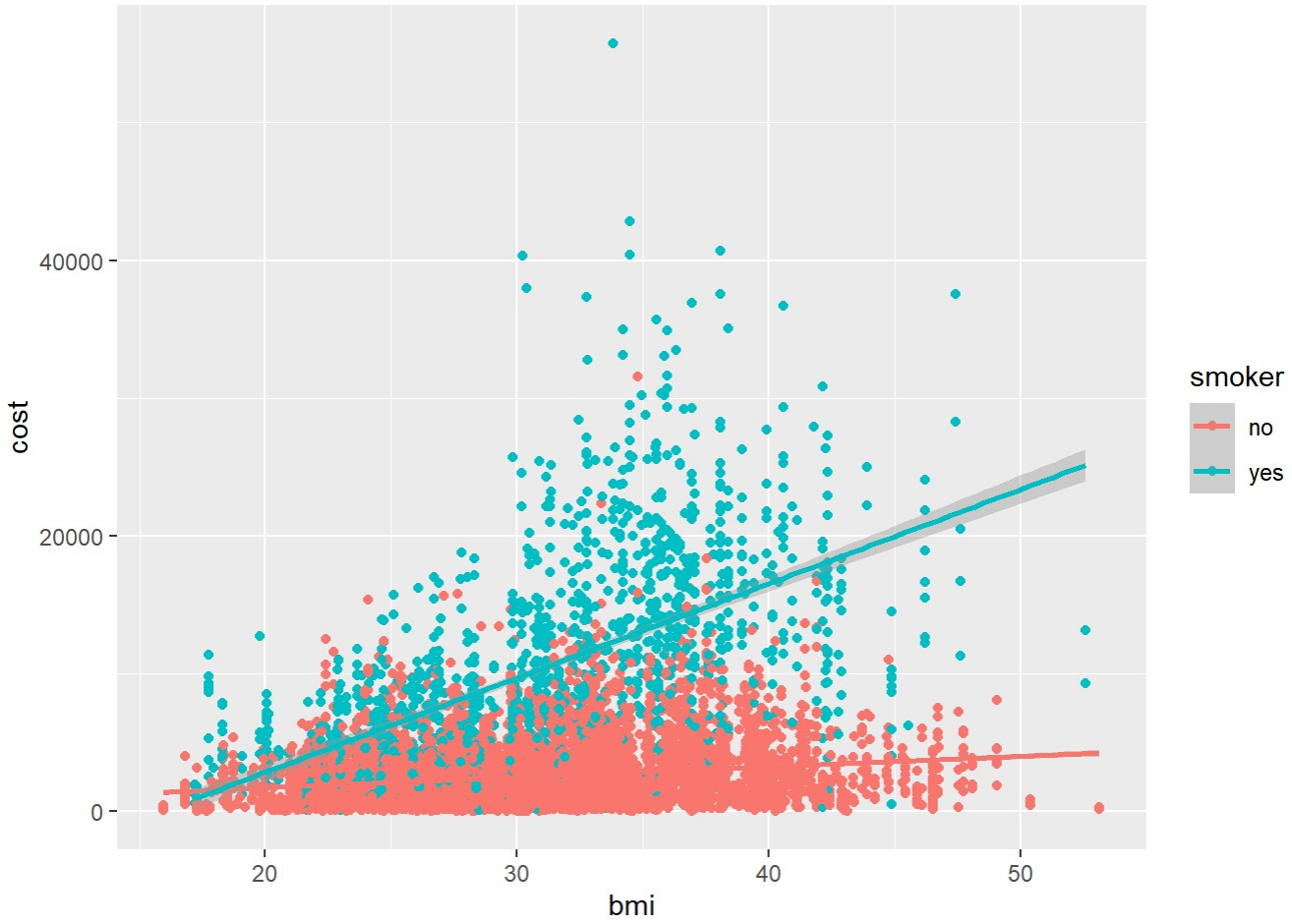
```
## `geom_smooth()` using formula = 'y ~ x'
```


```
# age vs cost (exercise)
ggplot(data=datafile_backup,aes(x=age, y=cost,colour = exercise))+ geom_point() +
  geom_smooth(method = "lm")
```

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


```
# bmi vs cost (smoker)
ggplot(data=datafile_backup,aes(x=bmi, y=cost,colour = smoker,))+ geom_point() +
  geom_smooth(method = "lm")
```

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




```
# bmi vs cost (exercise)
ggplot(data=datafile_backup,aes(x=bmi, y=cost,colour = exercise))+ geom_point() +
  geom_smooth(method = "lm")
```

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

Interpretation of Graphs

The provided plots show how healthcare costs are correlated with age and BMI for different smoking and exercise habits.

- For smokers, healthcare costs tend to increase with age, indicating a positive correlation between age and healthcare costs.
- People who are inactive have higher healthcare costs and there are some anomalies in the data for active individuals when considering age and healthcare costs.
- BMI has a positive correlation with healthcare costs, with a stronger correlation for smokers. People who have high BMI and smoke are most likely to have higher healthcare costs.
- Individuals with an inactive lifestyle tend to have higher healthcare costs, while there are some anomalies in the data for active individuals when considering the correlation between BMI and healthcare costs.

```
state_avg_cost <- datafile_backup %>%
  group_by(location) %>%
  summarise_at(vars(cost), list(name = mean)) %>%
  arrange(desc(name))
state_avg_cost
```

```
## # A tibble: 7 × 2
##   location       name
##   <chr>         <dbl>
## 1 NEW YORK      4676.
## 2 MASSACHUSETTS 4285.
## 3 RHODE ISLAND  4076.
## 4 PENNSYLVANIA  4032.
## 5 NEW JERSEY    3943.
## 6 CONNECTICUT   3823.
## 7 MARYLAND      3773.
```

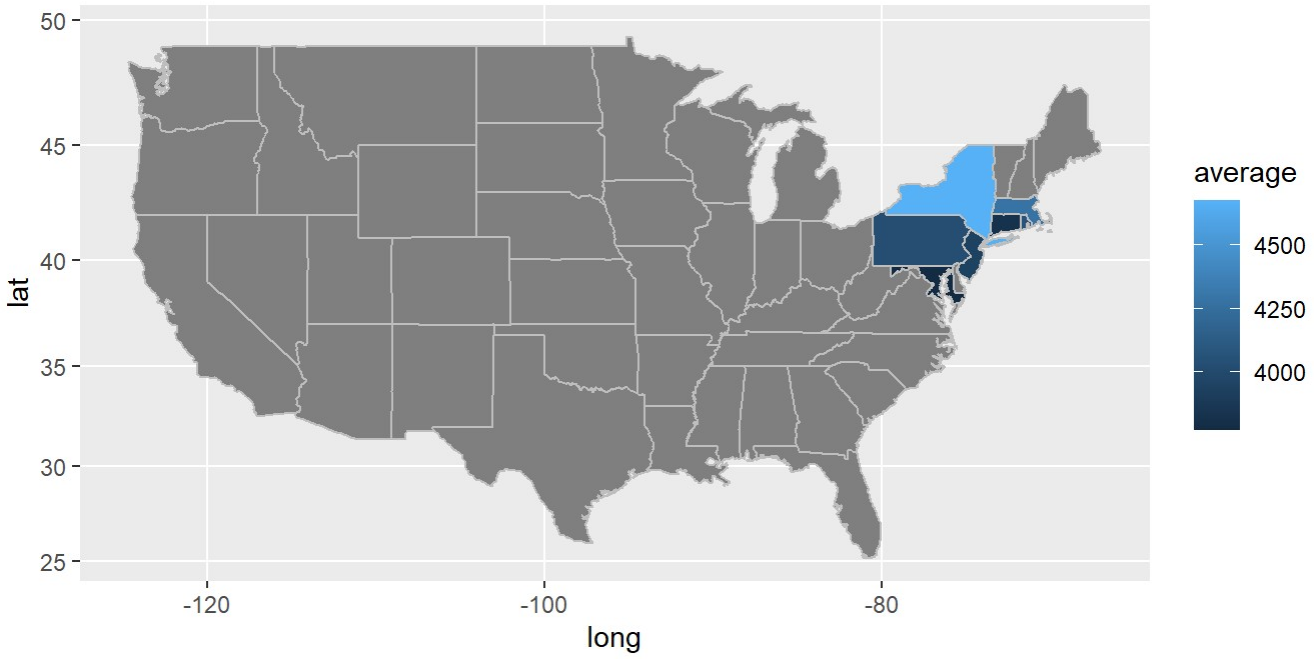
New York has the highest healthcare cost per average among all the states we have.

```
state_wise_count <- datafile_backup %>%
  count(location) %>%
arrange(desc(n))

state_wise_count
```

```
##        location    n
## 1  PENNSYLVANIA 3974
## 2      MARYLAND  742
## 3  RHODE ISLAND  697
## 4   CONNECTICUT  601
## 5      NEW YORK  537
## 6    NEW JERSEY  495
## 7 MASSACHUSETTS  456
```

We have about 50% people from the state of Pennsylvania itself.

```
us<-map_data("state")
datafile_backup$location <- tolower(datafile_backup$location)
m1 <- aggregate(datafile_backup$cost,by=list(datafile_backup$location),FUN=mean)
m2 <- aggregate(datafile_backup$cost,by=list(datafile_backup$location),FUN=max)
m3 <- aggregate(datafile_backup$cost,by=list(datafile_backup$location),FUN=min)
m1 <- m1%>%rename(location=Group.1)
m2 <- m2%>%rename(location=Group.1)
aggmerge1 <- merge(m1,m2,by = "location" )
m3 <- m3 %>% rename(location=Group.1)
aggmerge2 <- merge(aggmerge1,m3,by= "location")
aggmerge2 <- aggmerge2%>%rename(min=x,average=x.x,max=x.y)
m4 <- aggmerge2[,c(2:4)]
usmerge <- merge(us,aggmerge2,all.x=TRUE,by.x="region",by.y="location")
usmerge <- usmerge%>%arrange(order)

usmap1 <- ggplot(usmerge) + geom_polygon(aes(x=long, y=lat, group=group, fill = average), col
or="grey") + coord_map()
usmap1
```



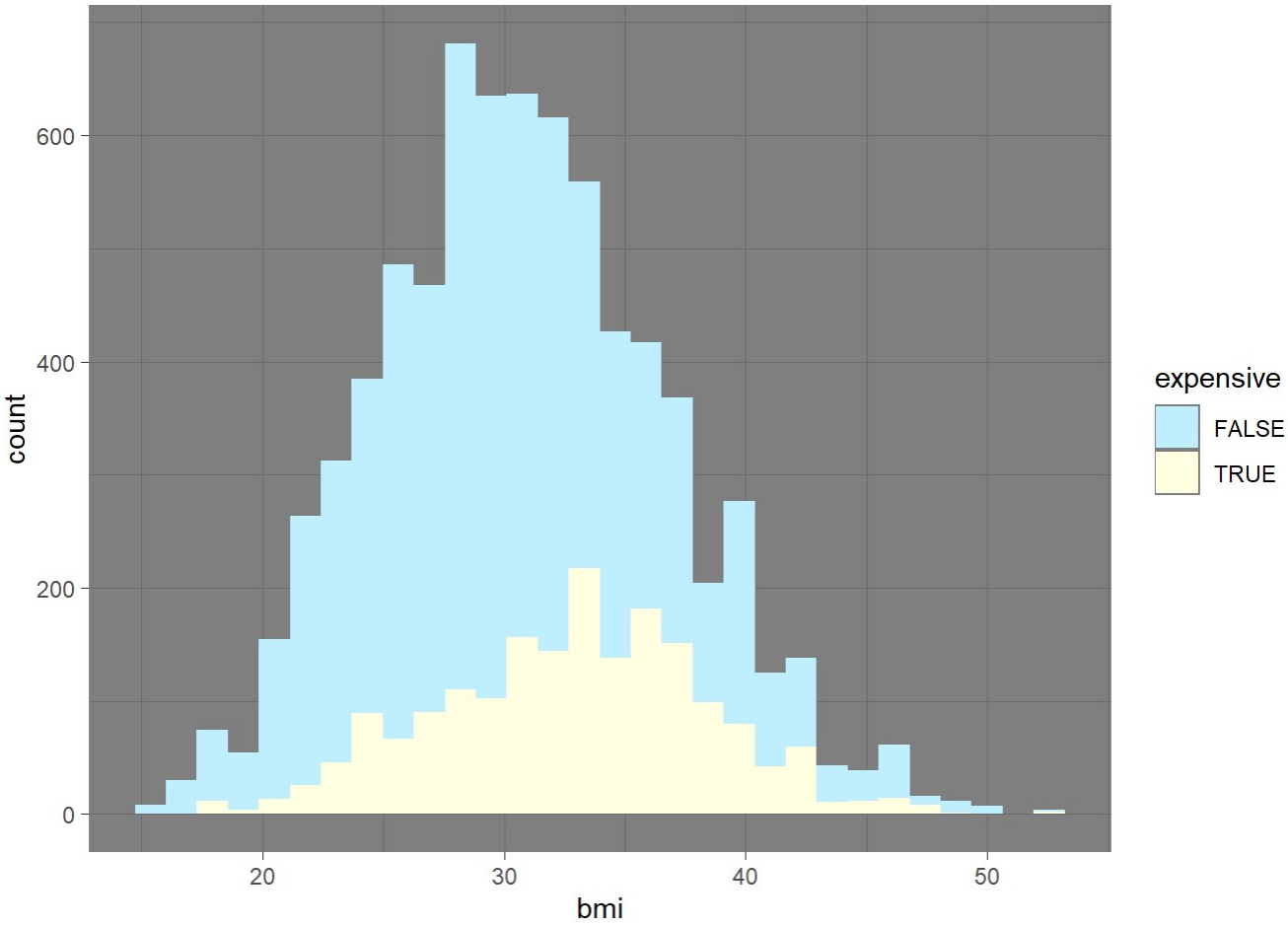
- New York has the highest average health care costs for individuals
- with 50% data representing Pennsylvania it still has the 4th highest average health care cost.
- Maryland has the lowest average health care cost for individuals.

It is evident that age and bmi has a positive correlation with healthcare cost of a customer. Children doesn't have a strong correlation with cost.

Now that we have an attribute that identifies expensive healthcare, we can generate plots for bmi, age, exercise and hypertension

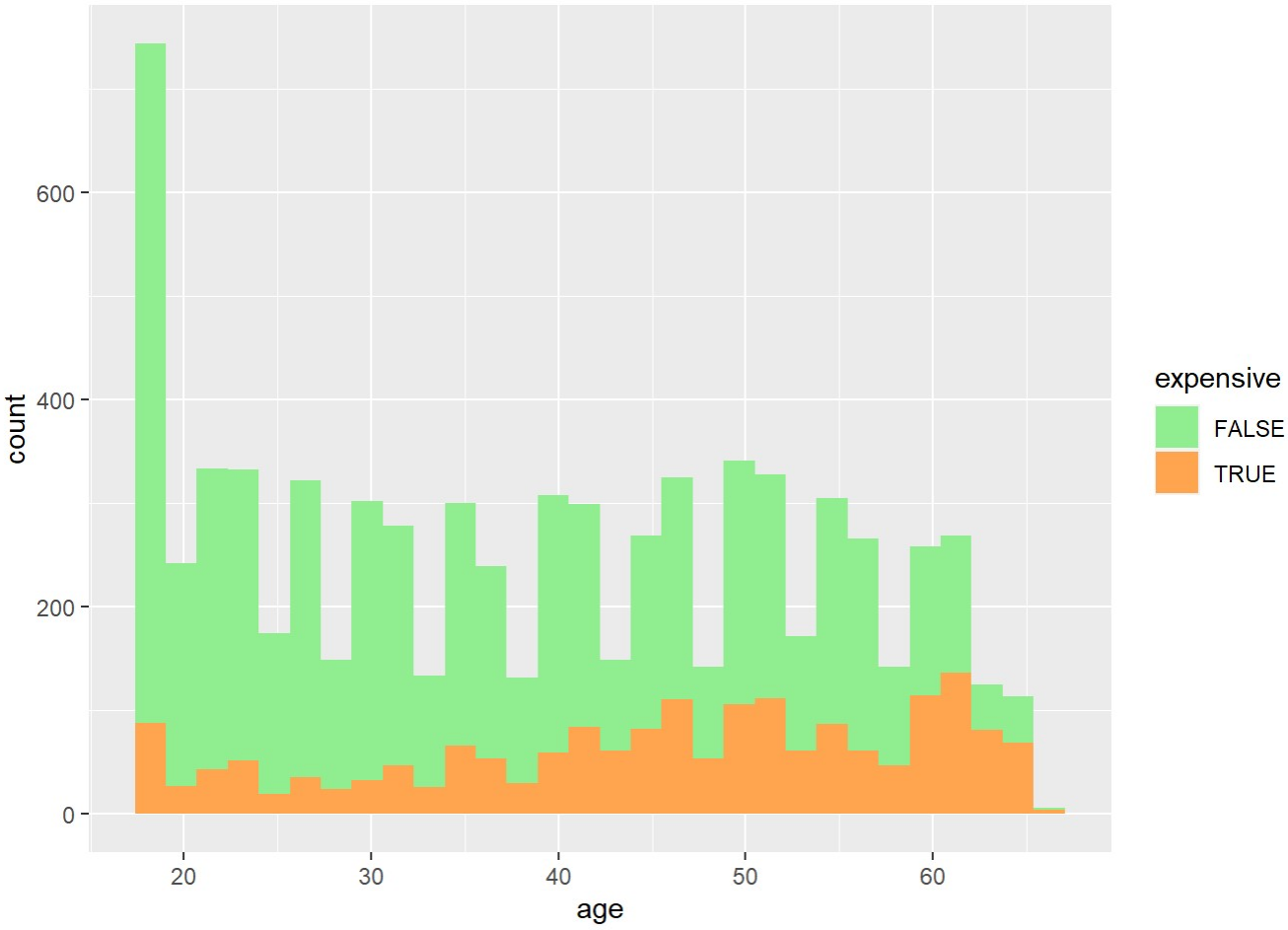
```
ggplot(datafile, aes(x = bmi, fill = expensive)) +
  geom_histogram() +
  scale_fill_manual(values = c("TRUE" = "lightyellow",
                               "FALSE" = "lightblue1"))+
  theme_dark()
```

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



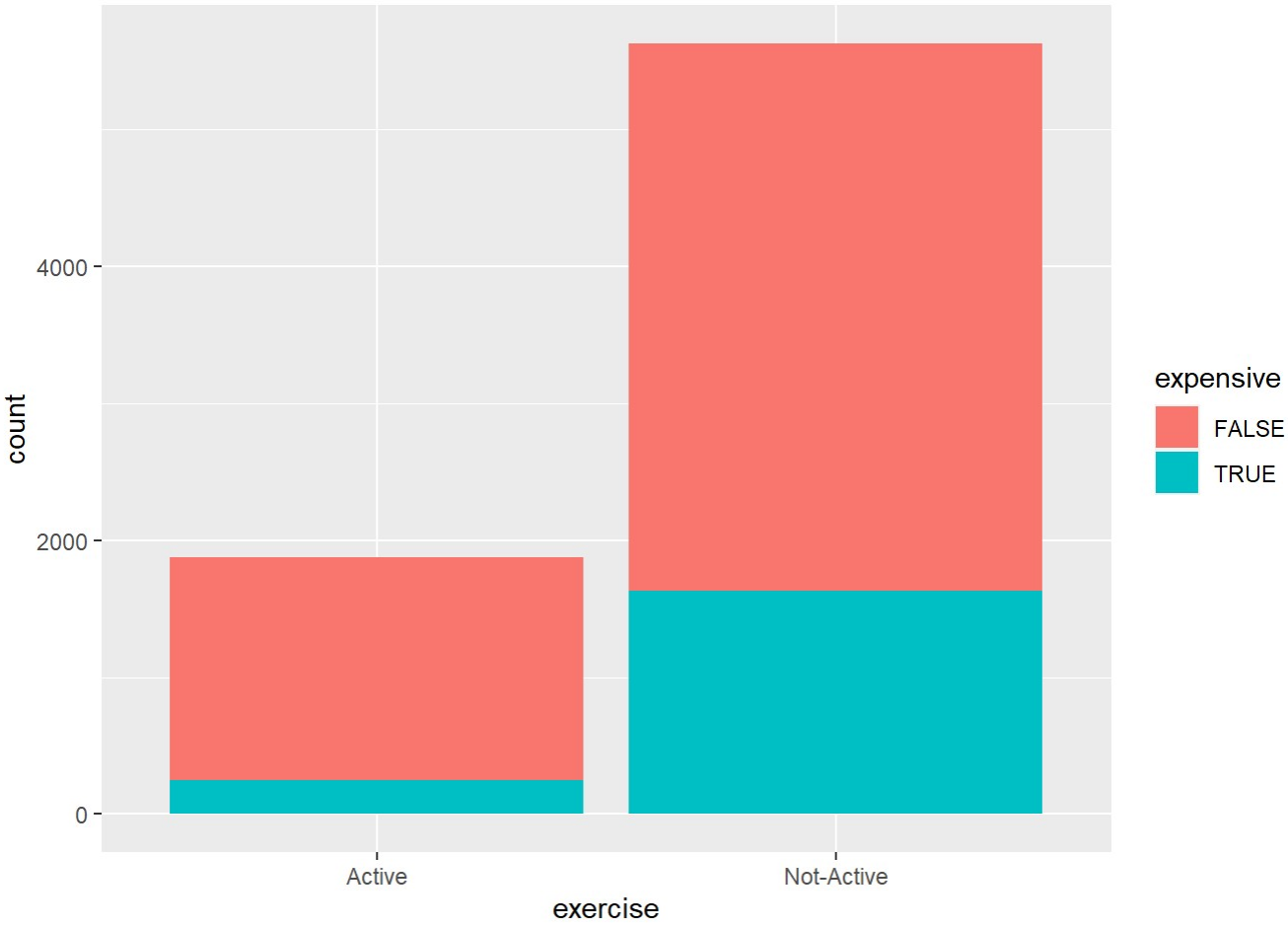

```
ggplot(datafile, aes(x = age, fill = expensive)) +
  geom_histogram() +
  scale_fill_manual(values = c("TRUE" = "tan1",
                               "FALSE" = "lightgreen"))
```

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




```
ggplot(datafile, aes(x = exercise, fill = expensive)) +
  geom_histogram(stat="count",binwidth = 10)
```

```
## Warning in geom_histogram(stat = "count", binwidth = 10): Ignoring unknown
## parameters: `binwidth`, `bins`, and `pad`
```



Interpretation of the Graphs

- bmi vs expensive: the variation of count of expensive people with increasing bmi is high.
- age vs expensive: the variation is visible with age as well.
- exercise vs expensive: the number of expensive people for non active lifestyle is pretty high, the same is also observed with active people. This does not provide a clear difference with exercise and expensive or non expensive people.

Prediction Model

To predict if a customer is Expensive or Not Expensive for a Health Care Company, we can try to create a Support Vector Machine model for our prediction. Before that we will have to change all the object columns to factors.

```
# import kernlab and caret libraries to environment
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.3
```

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

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

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

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.2.3
```

```
# check which columns are chr objects
str(datafile)
```

```
## 'data.frame':    7502 obs. of  17 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 ...
##  $ expensive      : logi  FALSE FALSE FALSE TRUE FALSE FALSE ...
##  $ age_grouped    : Factor w/ 6 levels "(10,20]","(20,30]",..: 1 1 2 3 3 4 3 5 2 6 ...
##  $ bmi_grouped    : Factor w/ 4 levels "(15,25]","(25,35]",..: 2 2 2 1 2 2 2 2 2 2 ...
```

```
# change "chr" columns to "factor"
# use as. factor to change data type of column

datafile$smoker <- as.factor(datafile$smoker)
datafile$location <- as.factor(datafile$location)
datafile$location_type <- as.factor(datafile$location_type)
datafile$education_level <- as.factor(datafile$education_level)
datafile$yearly_physical <- as.factor(datafile$yearly_physical)
datafile$exercise <- as.factor(datafile$exercise)
datafile$married <- as.factor(datafile$married)
datafile$gender <- as.factor(datafile$gender)
datafile$expensive <- as.factor(datafile$expensive)

# Removing Cost Column for prediction
datafile <- datafile[,-14]
```

Now that our columns are factorized we can split the data for training and testing.

```
set.seed(123)

# Randomly allocate data into training and testing by createDataPartition variable

train_list <- createDataPartition(y=datafile$expensive,p=.70, list=FALSE)
train_df <- datafile[train_list,]
test_df <- datafile[-train_list,]
```

```
# Creating a SVM Model
ksvm1 <- ksvm(expensive ~ ., data=train_df,C = 1,cross = 3, prob.model = TRUE)
ksvm1
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.0870814346685549
##
## Number of Support Vectors : 1790
##
## Objective Function Value : -1406.141
## Training error : 0.111555
## Cross validation error : 0.130021
## Probability model included.
```

```
# predicting the values of the test subset we created for model validation
svmPred <- predict(ksvm1, test_df, type = "response")


# creating a confusion matrix of the predicted values
confMat <- confusionMatrix(svmPred, test_df$expensive)
confMat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1632  214
##      TRUE     55  348
##
##                Accuracy : 0.8804
##                  95% CI : (0.8663, 0.8935)
##     No Information Rate : 0.7501
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6477
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9674
##             Specificity : 0.6192
##          Pos Pred Value : 0.8841
##          Neg Pred Value : 0.8635
##              Prevalence : 0.7501
##          Detection Rate : 0.7257
##    Detection Prevalence : 0.8208
##       Balanced Accuracy : 0.7933
##
##        'Positive' Class : FALSE
##
```

```
summary(svmPred)
```

```
## FALSE  TRUE
##  1846   403
```

R-Part | Decision Tree

We will make a decision tree to understand the outcome of each variable and understand how everything factors to the choices an individual makes.

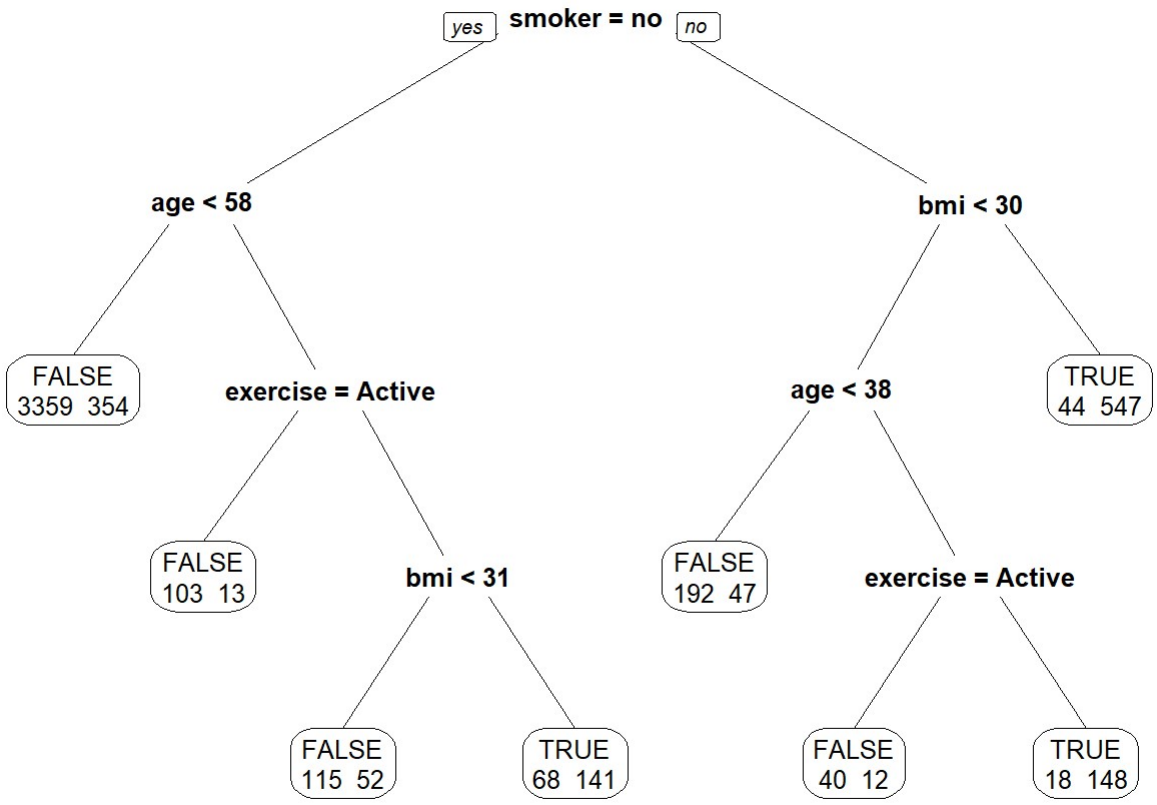
```
#install.packages("rpart.plot")
#install.packages("e1071")
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.2.3
```

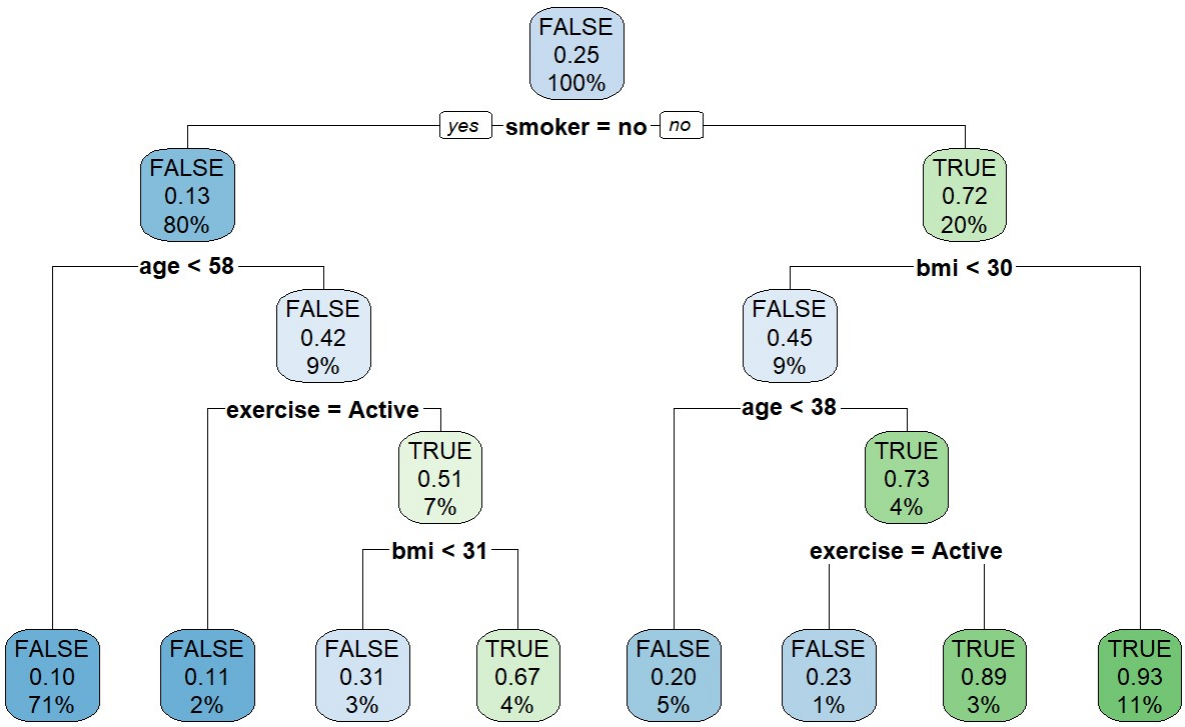
```
## Loading required package: rpart
```

```
## Warning: package 'rpart' was built under R version 4.2.3
```

```
library(rpart)
library(caret)
library(imputeTS)

datafile$bmi <- na_interpolation(datafile$bmi)
datafile$hypertension <- na_interpolation(datafile$hypertension)
datafile$expensive <- as.factor(datafile$expensive)
trainList <- createDataPartition(y=datafile$expensive,p=0.70,list=FALSE)
trainSet <- datafile[trainList, ]
testSet <- datafile[-trainList, ]

cartTree <- rpart(expensive ~ ., data = trainSet)
prp(cartTree, faclen = 0, cex = 0.8, extra = 1)
```




```
rpart.plot(cartTree, main = "expensive\n(binary response)")
```




```
predictValues <- predict(cartTree, newdata=testSet, type = "class")
confMatrix <- confusionMatrix(testSet$expensive, predictValues)
confMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1639   48
##      TRUE    239  323
##
##                Accuracy : 0.8724
##                  95% CI : (0.8579, 0.8859)
##     No Information Rate : 0.835
##     P-Value [Acc > NIR] : 4.868e-07
##
##                   Kappa : 0.6161
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8727
##             Specificity : 0.8706
##          Pos Pred Value : 0.9715
##          Neg Pred Value : 0.5747
##              Prevalence : 0.8350
##          Detection Rate : 0.7288
##    Detection Prevalence : 0.7501
##       Balanced Accuracy : 0.8717
##
##        'Positive' Class : FALSE
##
```

Association Rules

We also need to find trends on why health care costs a subset of customers expensive.

For this purpose we will explore Association Rules to define rules leading to expensive cost

- We need to remove index column from the train_set data frame
- Change numerical columns to factor data type

```
assoc_data <-datafile[,-1]

assoc_data$age <- as.factor(assoc_data$age)
assoc_data$bmi <- as.factor(assoc_data$bmi)
assoc_data$children <- as.factor(assoc_data$children)
assoc_data$hypertension <- as.factor(assoc_data$hypertension)

str(assoc_data)
```

```
## 'data.frame':    7502 obs. of  15 variables:
##  $ age            : Factor w/ 49 levels "18","19","20",..: 1 2 10 17 15 30 19 42 7 44 ...
##  $ bmi            : Factor w/ 601 levels "15.96","16.815",..: 210 387 367 73 240 379 273 1
48 158 159 ...
##  $ children       : Factor w/ 6 levels "0","1","2","3",..: 1 2 4 1 1 2 3 1 1 1 ...
##  $ 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   : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 2 1 1 ...
##  $ gender         : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 2 1 2 1 ...
##  $ expensive      : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 2 1 1 1 2 1 1 ...
##  $ age_grouped    : Factor w/ 6 levels "(10,20]","(20,30]",..: 1 1 2 3 3 4 3 5 2 6 ...
##  $ bmi_grouped    : Factor w/ 4 levels "(15,25]","(25,35]",..: 2 2 2 1 2 2 2 2 2 2 ...
```

- Import required libraries (arules, arulesViz)
- Change data into transaction
- Create apriori function to generate rules

```
library(arules)
```

```
## Warning: package 'arules' was built under R version 4.2.3
```

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

```
## Warning: package 'Matrix' was built under R version 4.2.3
```

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

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

```
##
## Attaching package: 'arules'
```

```
## The following object is masked from 'package:kernlab':
##
##     size
```

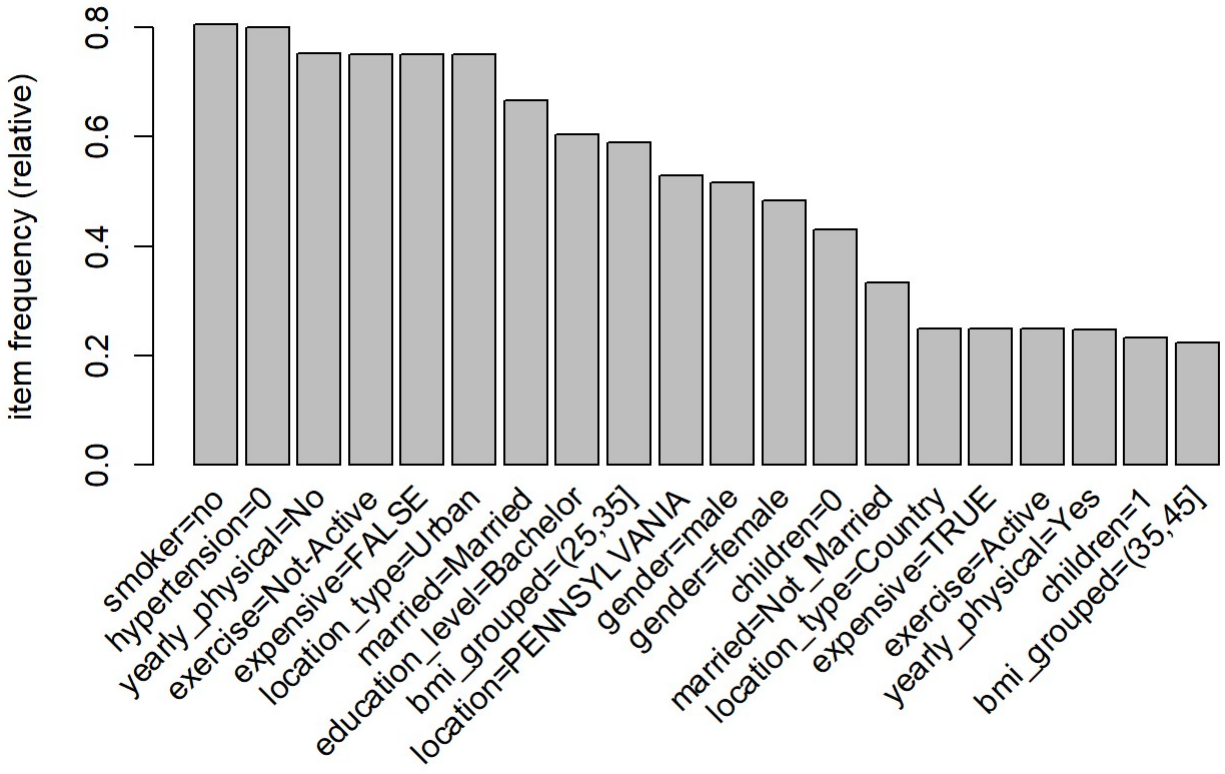
```
## The following object is masked from 'package:dplyr':
##
##     recode
```

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

```
library(arulesViz)
```

```
## Warning: package 'arulesViz' was built under R version 4.2.3
```

```
tranData <- as(assoc_data, "transactions")


itemFrequencyPlot(tranData, topN=20)
```



```
# Rules
rules <- apriori(tranData,
                   parameter=list(supp=0.060, conf=0.82),
                   control=list(verbose=F), # control algorithm performance
                   appearance=list(default="lhs",rhs=("expensive=TRUE")))
summary(rules)
```

```
## set of 6 rules
##
## rule length distribution (lhs + rhs):sizes
## 4 5
## 4 2
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.000   4.000   4.000   4.333   4.750   5.000
##
## summary of quality measures:
##     support          confidence        coverage            lift
##  Min.   :0.06145   Min.   :0.8205   Min.   :0.07438   Min.   :3.281
##  1st Qu.:0.06408   1st Qu.:0.8221   1st Qu.:0.07768   1st Qu.:3.288
##  Median :0.07151   Median :0.8264   Median :0.08531   Median :3.305
##  Mean   :0.07247   Mean   :0.8296   Mean   :0.08738   Mean   :3.318
##  3rd Qu.:0.07815   3rd Qu.:0.8278   3rd Qu.:0.09364   3rd Qu.:3.310
##  Max.   :0.08838   Max.   :0.8556   Max.   :0.10770   Max.   :3.421
##      count
##  Min.   :461.0
##  1st Qu.:480.8
##  Median :536.5
##  Mean   :543.7
##  3rd Qu.:586.2
##  Max.   :663.0
##
## mining info:
##      data ntransactions support confidence
##  tranData          7502    0.06       0.82
##
call
##  apriori(data = tranData, parameter = list(supp = 0.06, conf = 0.82), appearance = list(de
fault = "lhs", rhs = ("expensive=TRUE")), control = list(verbose = F))
```

```
inspect(rules)
```

```
##     lhs                             rhs                 support confidence   coverage     l
ift count
## [1] {smoker=yes,
##      exercise=Not-Active,
##      gender=male}               => {expensive=TRUE} 0.07344708  0.8555901 0.08584377 3.421
448   551
## [2] {smoker=yes,
##      education_level=Bachelor,
##      exercise=Not-Active}       => {expensive=TRUE} 0.06958144  0.8207547 0.08477739 3.282
144   522
## [3] {smoker=yes,
##      exercise=Not-Active,
##      married=Married}           => {expensive=TRUE} 0.07971208  0.8282548 0.09624100 3.312
136   598
## [4] {smoker=yes,
##      location_type=Urban,
##      exercise=Not-Active}       => {expensive=TRUE} 0.08837643  0.8205446 0.10770461 3.281
303   663
## [5] {smoker=yes,
##      location_type=Urban,
##      exercise=Not-Active,
##      married=Married}           => {expensive=TRUE} 0.06145028  0.8261649 0.07438017 3.303
779   461
## [6] {smoker=yes,
##      exercise=Not-Active,
##      married=Married,
##      hypertension=0}            => {expensive=TRUE} 0.06225007  0.8265487 0.07531325 3.305
314   467
```

```
View(assoc_data)
```

With Support = 0.065 and Confidence = 0.80, we are able to generate association rules for condition where cost is expensive.

- Smoker = yes and exercise = Not-Active for all 4 rules.
- Hypertension didn’t seem to increase health care cost.
- Gender and education level are all observed once.