Final Project

2023-05-09

Final Project

IST687: Introduction to Data Science Health Management Organization data

Analysis of the dataset to see which predictors influence whether a person will be an “expensive” entity or not. Various factors might influence this decision and we will look at a few visuals, work on the data using predictive analysis to reach a conclusion for the management.

```
#First, we have to import the libraries that we will be using for our initial
#reading of data, cleaning the data, and simple analysis.
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.2.1     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#We use the read_csv function to get the data and store it in the HMO_data variable.
HMO_data <- read_csv('https://intro-datascience.s3.us-east-2.amazonaws.com/HMO_data.csv')
```

```
## Rows: 7582 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): smoker, location, location_type, education_level, yearly_physical, ...
## dbl (6): X, age, bmi, children, hypertension, cost
##
## 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.
```

1. Data Analysis and Cleaning

In this section, we will figure out the data structure, it’s attributes, the type of variables in the columns, how they look like statistically (mean, mode, median, etc.), if they have any irregularities or null values, and fix those null values.

```
#We will now have a look at the data to see the structures, identify any missing values.
dim(HMO_data)
```

```
## [1] 7582   14
```

```r
#This dataset has 7582 rows (observations) and 14 columns (attributes).

#We will have a look at the data.
head(HMO_data)
```

```
## # A tibble: 6 x 14
##       X   age   bmi children smoker location     locat~1 educa~2 yearl~3 exerc~4
##   <dbl> <dbl> <dbl>    <dbl> <chr>  <chr>        <chr>   <chr>   <chr>   <chr>
## 1     1    18  27.9        0 yes    CONNECTICUT  Urban   Bachel~ No      Active
## 2     2    19  33.8        1 no     RHODE ISLAND Urban   Bachel~ No      Not-Ac~
## 3     3    27  33          3 no     MASSACHUSET~ Urban   Master  No      Active
## 4     4    34  22.7        0 no     PENNSYLVANIA Country Master  No      Not-Ac~
## 5     5    32  28.9        0 no     PENNSYLVANIA Country PhD     No      Not-Ac~
## 6     7    47  33.4        1 no     PENNSYLVANIA Urban   Bachel~ No      Not-Ac~
## # ... with 4 more variables: married <chr>, hypertension <dbl>, gender <chr>,
## #   cost <dbl>, and abbreviated variable names 1: location_type,
## #   2: education_level, 3: yearly_physical, 4: exercise
```

```r
#Let's look at the types of each variable.
str(HMO_data)
```

```
## spc_tbl_ [7,582 x 14] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ X              : num [1:7582] 1 2 3 4 5 7 9 10 11 12 ...
##  $ age            : num [1:7582] 18 19 27 34 32 47 36 59 24 61 ...
##  $ bmi            : num [1:7582] 27.9 33.8 33 22.7 28.9 ...
##  $ children       : num [1:7582] 0 1 3 0 0 1 2 0 0 0 ...
##  $ smoker         : chr [1:7582] "yes" "no" "no" "no" ...
##  $ location       : chr [1:7582] "CONNECTICUT" "RHODE ISLAND" "MASSACHUSETTS" "PENNSYLVANIA" ...
##  $ location_type  : chr [1:7582] "Urban" "Urban" "Urban" "Country" ...
##  $ education_level: chr [1:7582] "Bachelor" "Bachelor" "Master" "Master" ...
##  $ yearly_physical: chr [1:7582] "No" "No" "No" "No" ...
##  $ exercise       : chr [1:7582] "Active" "Not-Active" "Active" "Not-Active" ...
##  $ married        : chr [1:7582] "Married" "Married" "Married" "Married" ...
##  $ hypertension   : num [1:7582] 0 0 0 1 0 0 0 1 0 0 ...
##  $ gender         : chr [1:7582] "female" "male" "male" "male" ...
##  $ cost           : num [1:7582] 1746 602 576 5562 836 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   X = col_double(),
##   ..   age = col_double(),
##   ..   bmi = col_double(),
##   ..   children = col_double(),
##   ..   smoker = col_character(),
##   ..   location = col_character(),
##   ..   location_type = col_character(),
##   ..   education_level = col_character(),
##   ..   yearly_physical = col_character(),
##   ..   exercise = col_character(),
##   ..   married = col_character(),
##   ..   hypertension = col_double(),
##   ..   gender = col_character(),
##   ..   cost = col_double()
##   .. )
```

```
##  - attr(*, "problems")=<externalptr>
```

```
#There are 6 attributes that are of numerical type and 8 columns
#that are of type character (string).
```

```
#Next, we gather some statistical analysis of the variables.
summary(HMO_data)
```

```
##        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  
## 
```

```
#As we can see, each numerical variable has a mean, median, etc. Also, in this
#analysis, we see that bmi has 78 null values while hypertension has 80 null values.
```

```
#Let's see if the string columns have any null values.
nrow(HMO_data[is.na(HMO_data$smoker),])
```

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

```
nrow(HMO_data[is.na(HMO_data$location),])
```

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

```r
nrow(HMO_data[is.na(HMO_data$location_type),])
```

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

```r
nrow(HMO_data[is.na(HMO_data$education_level),])
```

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

```r
nrow(HMO_data[is.na(HMO_data$yearly_physical),])
```

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

```r
nrow(HMO_data[is.na(HMO_data$exercise),])
```

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

```r
nrow(HMO_data[is.na(HMO_data$married),])
```

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

```r
nrow(HMO_data[is.na(HMO_data$gender),])
```

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

```r
#We see that there are no null values in any of the other columns.
```

```r
#Let's remove the NA values from bmi and hypertension.
#We will use the na_interpolation() function to achieve this.
#It is in the imputeTS package.
#install.packages('imputeTS')
library(imputeTS)
```

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

```r
#Before
print('Number of NA\'s Before')
```

```
## [1] "Number of NA's Before"
```

```r
nrow(HMO_data[is.na(HMO_data$bmi),])
```

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

```r
nrow(HMO_data[is.na(HMO_data$hypertension),])
```

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

```r
#Using na_interpolation() to remove null values from the two columns.
HMO_data$bmi <- na_interpolation(HMO_data$bmi)
HMO_data$hypertension <- na_interpolation(HMO_data$hypertension)

#After
print('Number of NA\'s After')
```

```
## [1] "Number of NA's After"
```

```r
nrow(HMO_data[is.na(HMO_data$bmi),])
```

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

```r
nrow(HMO_data[is.na(HMO_data$hypertension),])
```

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

In this section we will remove the possible outliers. Assuming that the bottom and top 0.5% data contained in the distribution curve contains possible outliers

```r
lower_bound <- quantile(HMO_data$cost, 0.005)
upper_bound <- quantile(HMO_data$cost, 0.995)
lower_bound #0.5th percentile of all data
```

```
##  0.5%
## 79.81
```

```r
upper_bound #99.5th percentile of all data
```

```
##    99.5%
## 27723.03
```

```r
outliers <- which(HMO_data$cost < lower_bound | HMO_data$cost > upper_bound)

nrow(HMO_data[outliers,]) #number of outliers
```

```
## [1] 76
```

```r
HMO_data_new <- HMO_data[-outliers,]

#We now look at the summary statistics of the new dataset.
summary(HMO_data_new)
```

```
##        X                   age             bmi           children
##  Min.   :        1   Min.   :18.00   Min.   :15.96   Min.   :0.00
##  1st Qu.:     5606   1st Qu.:26.00   1st Qu.:26.60   1st Qu.:0.00
##  Median :    24056   Median :39.00   Median :30.50   Median :1.00
##  Mean   :   714251   Mean   :38.87   Mean   :30.78   Mean   :1.11
##  3rd Qu.:   117688   3rd Qu.:51.00   3rd Qu.:34.66   3rd Qu.:2.00
##  Max.   :131101111   Max.   :66.00   Max.   :53.13   Max.   :5.00
##     smoker            location         location_type      education_level
##  Length:7506        Length:7506        Length:7506        Length:7506
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  yearly_physical      exercise           married           hypertension
##  Length:7506        Length:7506        Length:7506        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
##     gender               cost
##  Length:7506        Min.   :   80
##  Class :character   1st Qu.:  978
##  Mode  :character   Median : 2500
##                     Mean   : 3914
##                     3rd Qu.: 4748
##                     Max.   :27714
```

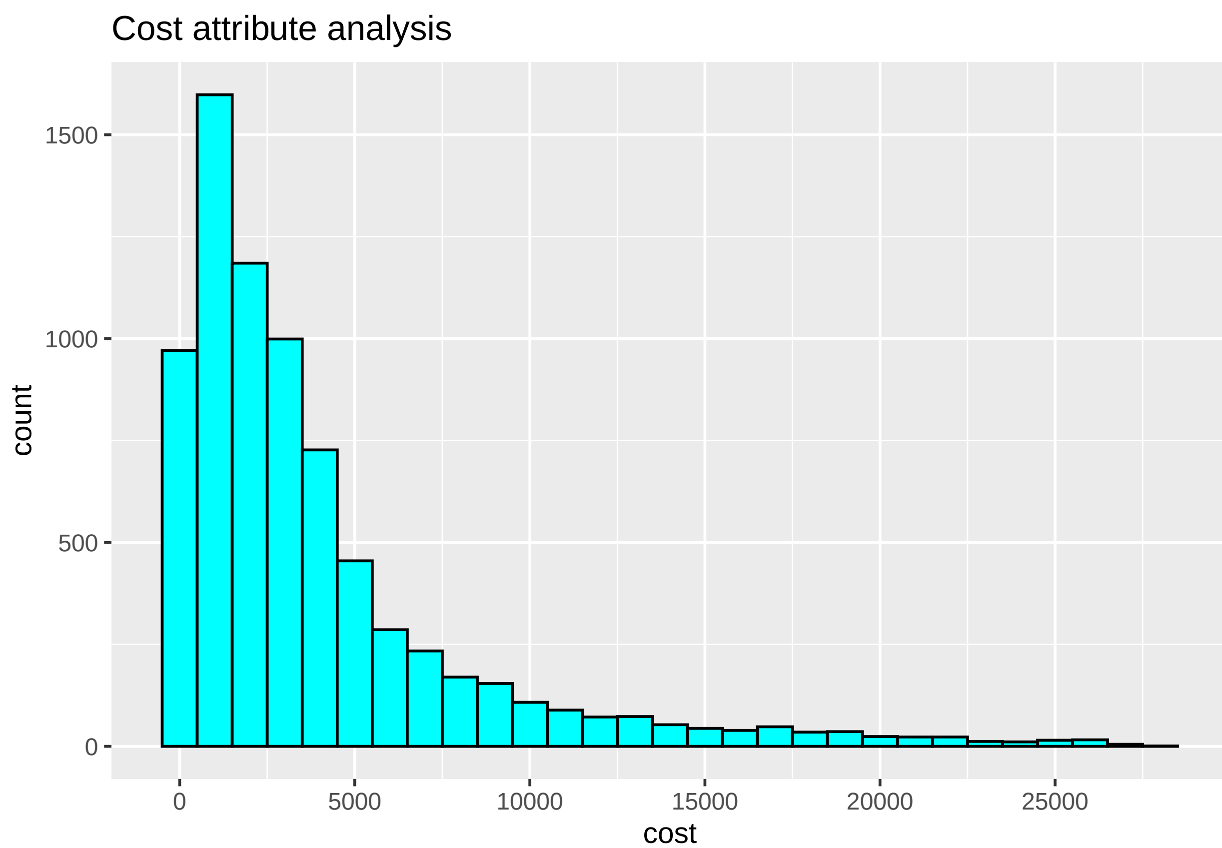
2. Data Visualization

In this section, we will attempt to look at some graphs and charts to get a better idea of the variables and how they are spread out. We will also look at some histograms, some bar charts, scatterplots, and even a map of the different states and regions to see how the cost of each individual's insurance policy varies.

```
#In this section, we will try to visualize the patterns between categorical attributes with
#the cost attribute. Also, analyse the pattern of cost attribute

#Here we're using gridExtra package for arranging multiple graphs in one pane
#install.packages("gridExtra")

library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

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

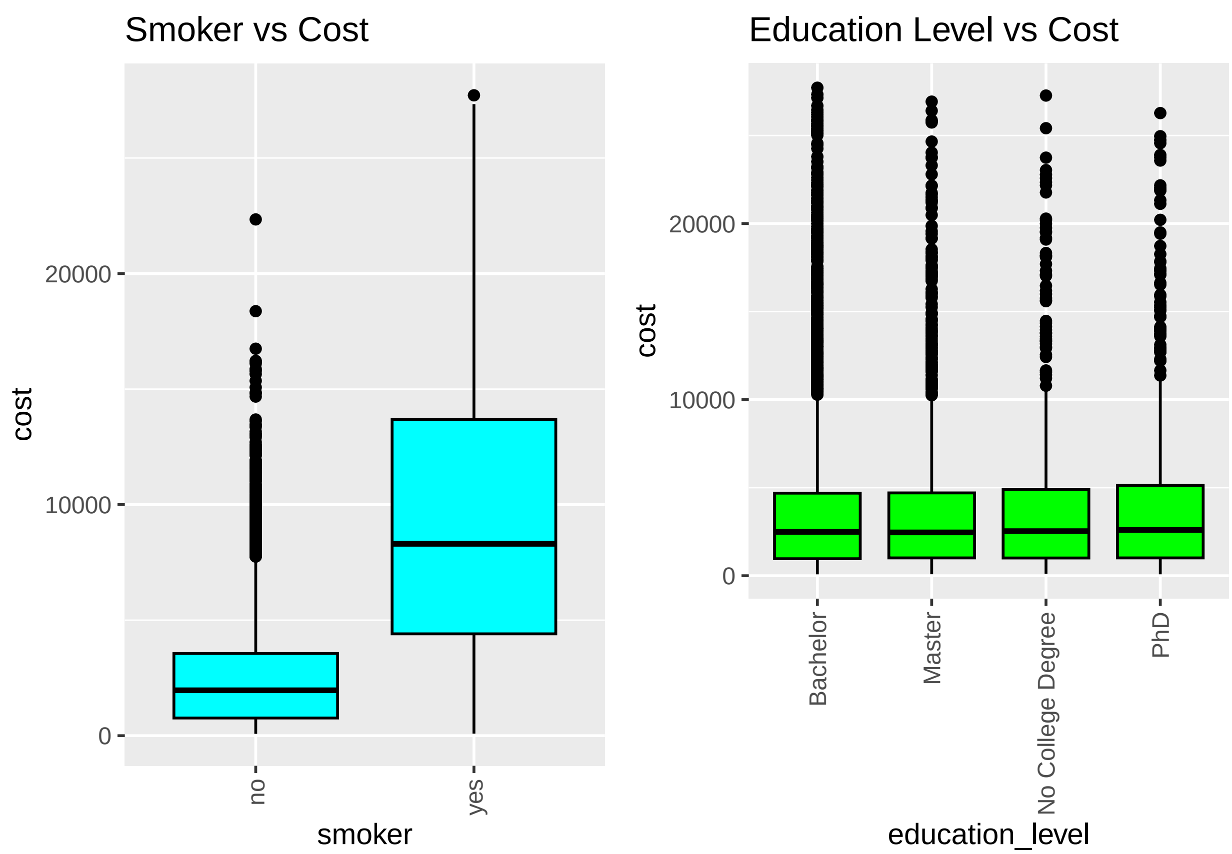
```
ggplot(HMO_data_new) + geom_histogram(aes(x=cost), col='black', fill='cyan', binwidth = 1000) +
  ggtitle("Cost attribute analysis") + scale_x_continuous(breaks = seq(0, 50000, by = 5000))
```



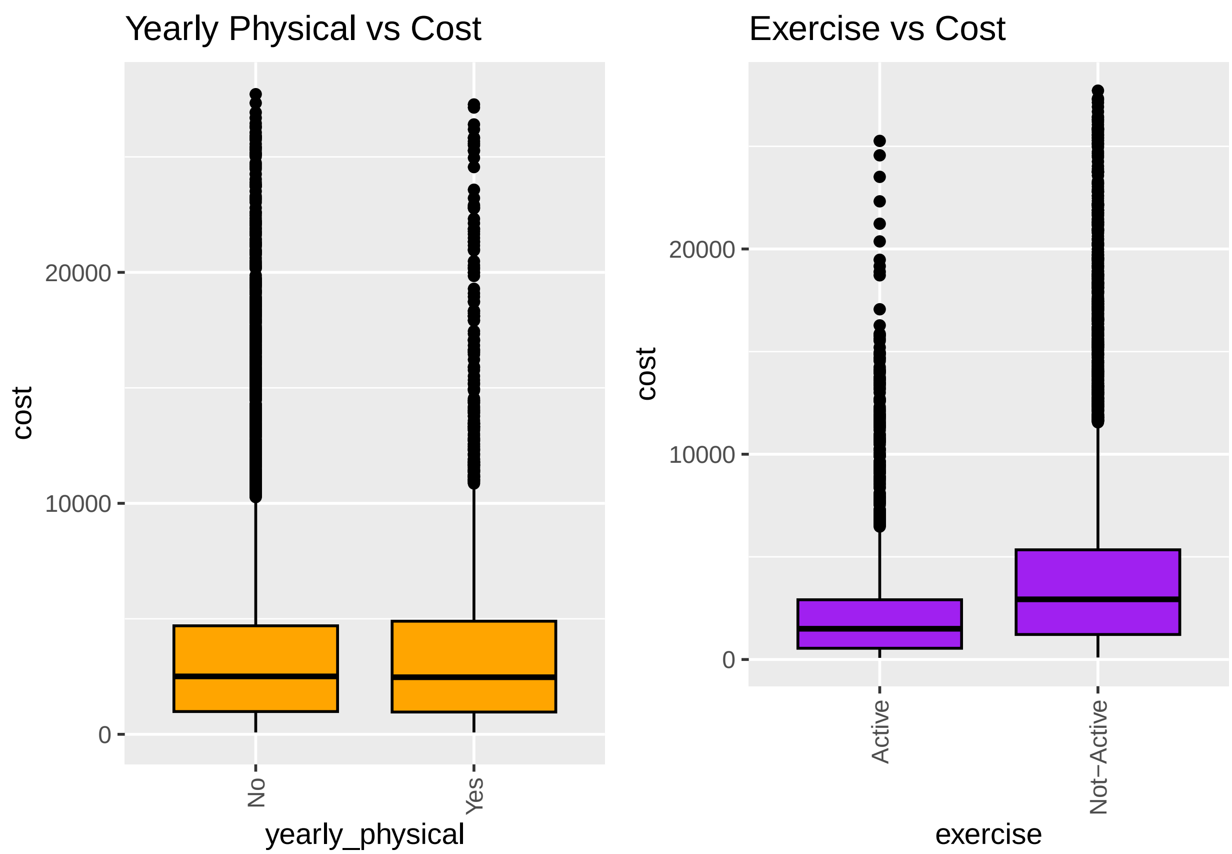

```r
#We can observe that most of the patients in the dataset are incurring costs in the range of $0-10000,
#with the graph showing a right-skewed pattern
```

```r
#Plotting box-whisker plots to analyse the effect of each categorical attribute on the
#'Cost' attribute
#'
g1 <- ggplot(HMO_data_new) + geom_boxplot(aes(x=smoker, y=cost), col="black", fill = 'cyan') +
  ggtitle("Smoker vs Cost")
g1 <- g1 +  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

g2 <- ggplot(HMO_data_new) + geom_boxplot(aes(x=education_level, y=cost), col="black",
                                          fill = 'green', ) + ggtitle("Education Level vs Cost")
g2 <- g2 +  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

g3 <- ggplot(HMO_data_new) + geom_boxplot(aes(x=yearly_physical, y=cost), col="black",
                                          fill = 'orange') + ggtitle("Yearly Physical vs Cost")
g3 <- g3 +  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

g4 <- ggplot(HMO_data_new) + geom_boxplot(aes(x=exercise, y=cost), col="black",
                                          fill = 'purple') + ggtitle("Exercise vs Cost")
g4 <- g4 +  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

g5 <- ggplot(HMO_data_new) + geom_boxplot(aes(x=married, y=cost), col="black",
                                          fill = 'pink') + ggtitle("Marraige Status vs Cost")
g5 <- g5 +  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

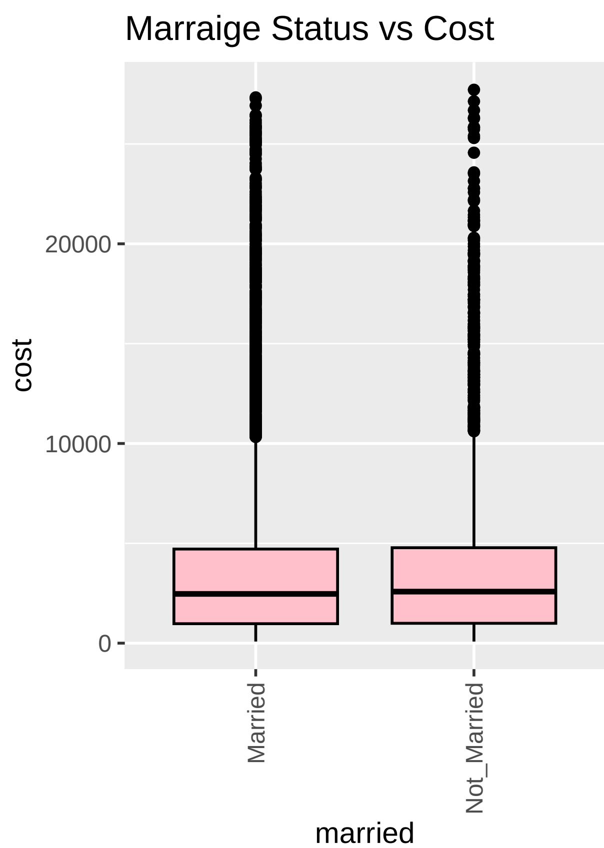
```
g6 <- ggplot(HMO_data_new) + geom_boxplot(aes(x=gender, y=cost), col="black",
                                          fill = 'yellow') + ggtitle("Gender vs Cost")
g6 <- g6 +  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))


grid.arrange(g1, g2, nrow=1)
```

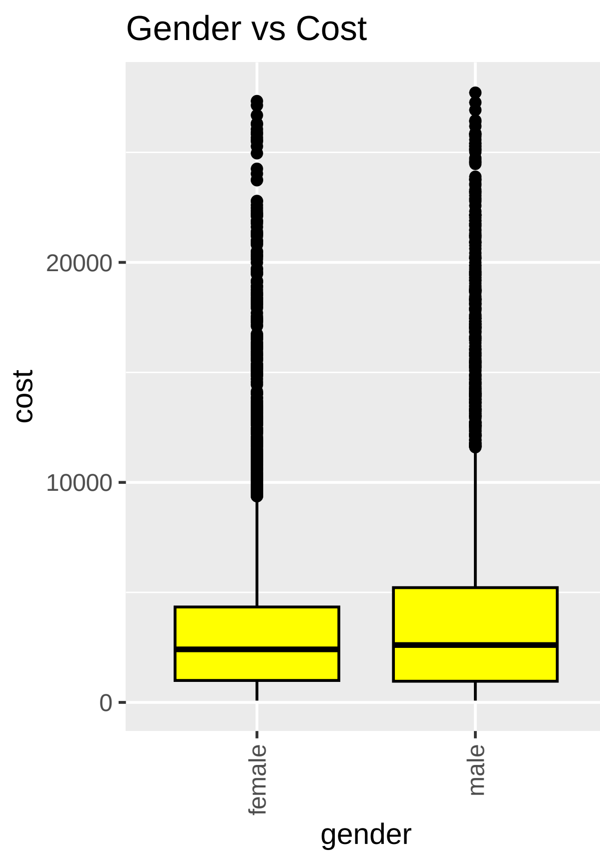



```
grid.arrange(g3, g4, nrow=1)
```




```
grid.arrange(g5, g6, nrow=1)
```





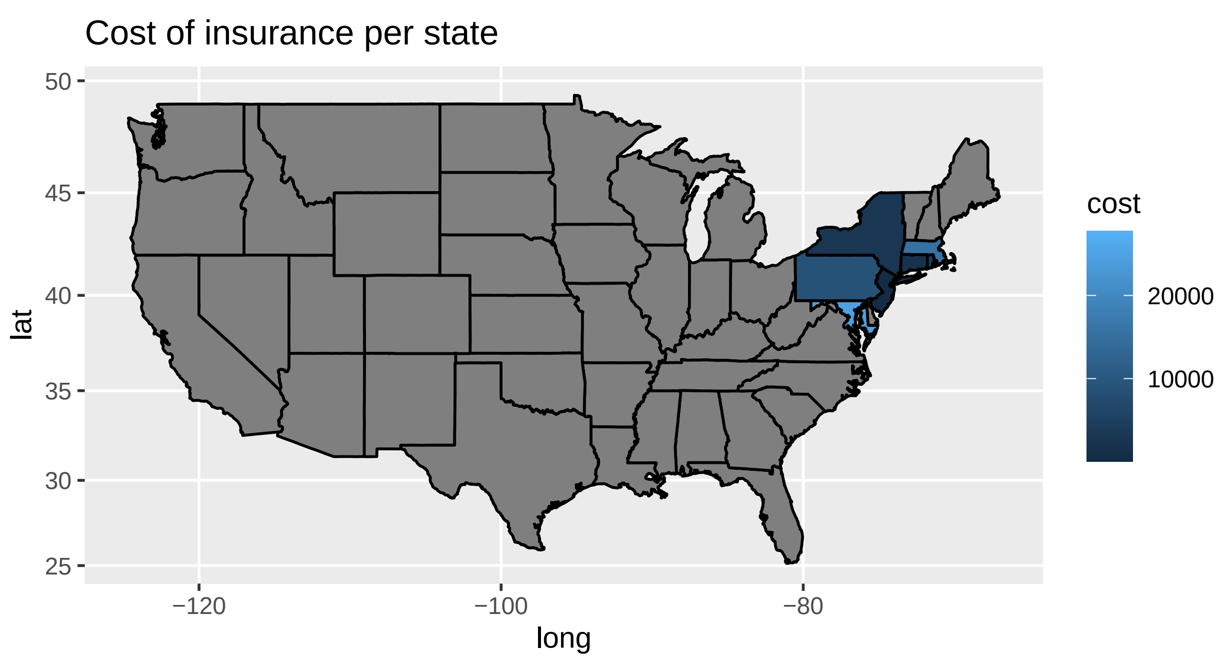

```r
#We can observe that for each attribute, there is a category for which the overall cost
#incurred by people is higher as compared to the other categories.
#Although, these graphs don't provide enough information about which of the attribute is
#most closely related with cost
```

```r
#Map plot to analyse how various regions affect the cost expenditure of patients in the dataset.

#We will plot a map of all the US states and have the color represent the cost of expenditure
#for each state.

#For this we will use the ggplot2 library

# install.packages('ggplot2')
# install.packages('dplyr')
# install.packages('maps')
# install.packages('mapproj')

library(ggplot2)
library(dplyr)
library(maps)
```

```
##
## Attaching package: 'maps'

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

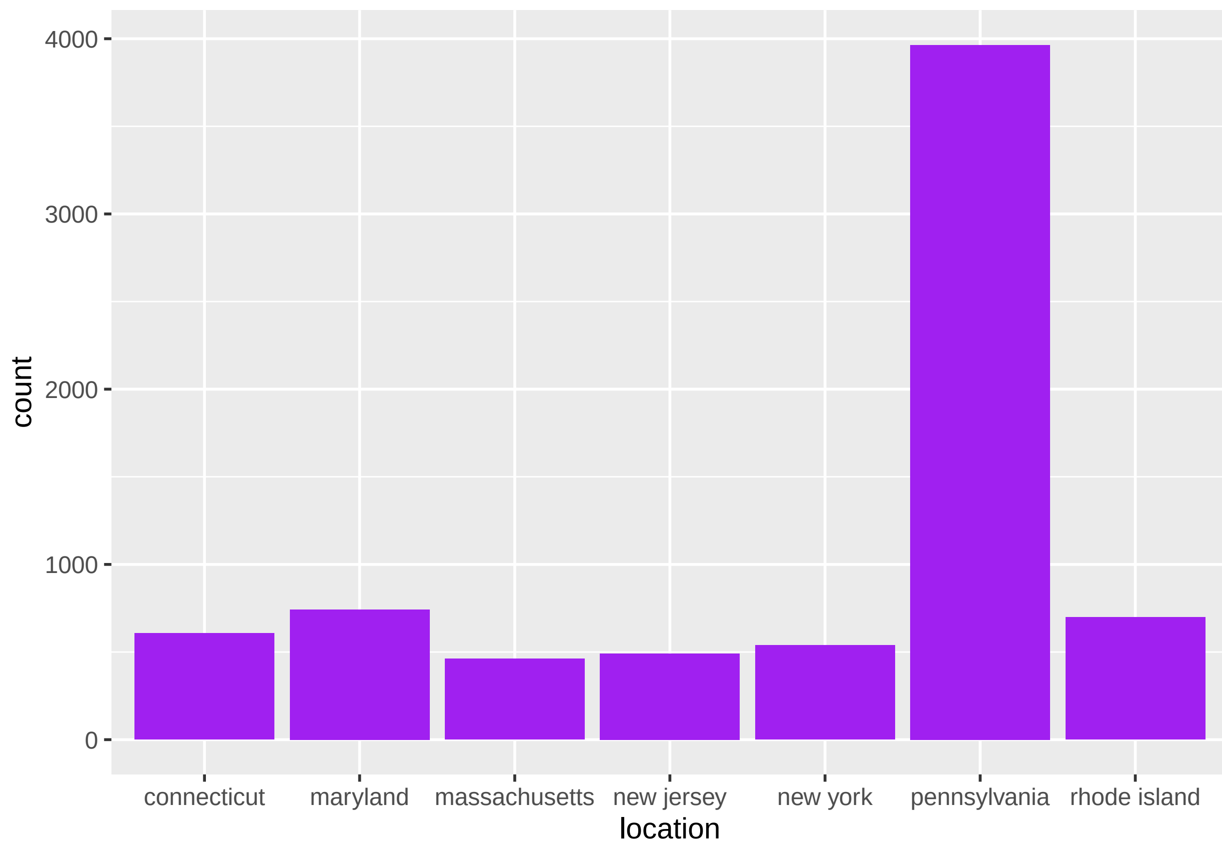
```
##
##     map
```

```r
library(mapproj)

states <- map_data("state")
states$state_name <- tolower(states$region)
HMO_data_new$location <- tolower(HMO_data_new$location)
HMO_data_with_states <- merge(HMO_data_new, states, all.y=TRUE, by.x="location", by.y="region")

HMO_data_with_states <- HMO_data_with_states %>% arrange(order)

ggplot(HMO_data_with_states) + geom_polygon(color="black",
                 aes(x=long,y=lat,group=group,fill=cost)) +
        ggtitle('Cost of insurance per state') +
coord_map()
```

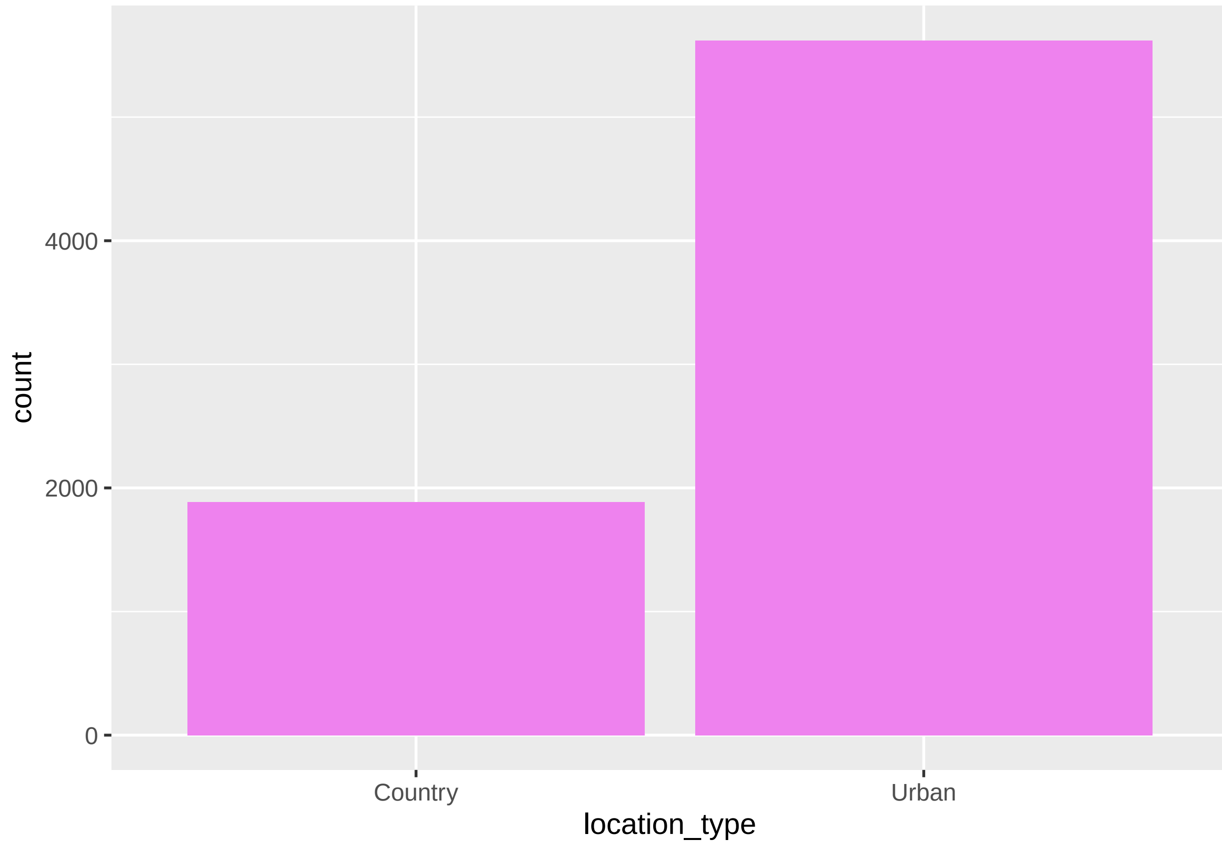



```r
#As we can see, the dataset is limited to some of the northeastern states (7), and it shows
#the average cost per state of insurance. It is evident that Maryland has a higher
#average than other states.
```

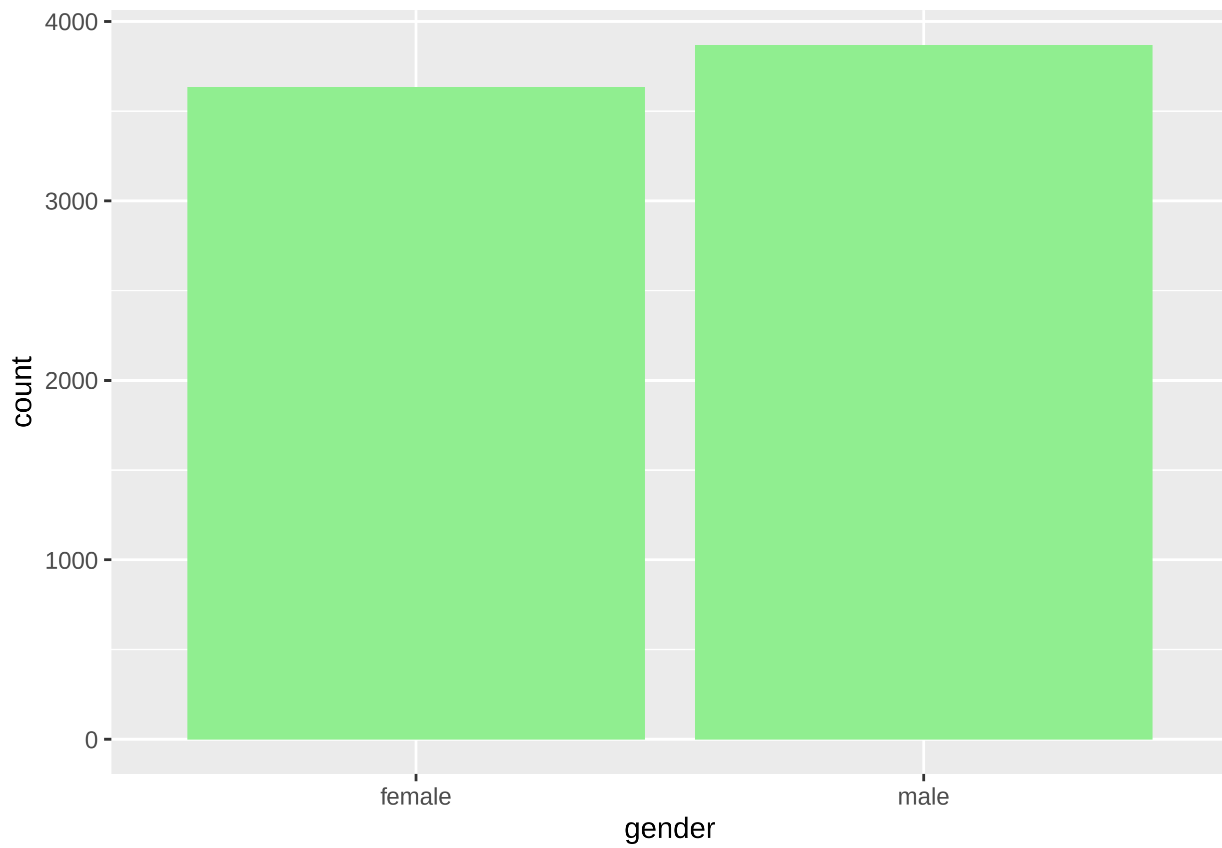
```r
#Creating a bar graph representing the count of people from different states.

countPlot <- ggplot(data = HMO_data_new) + aes(x=location) + geom_bar(fill='purple')
countPlot
```



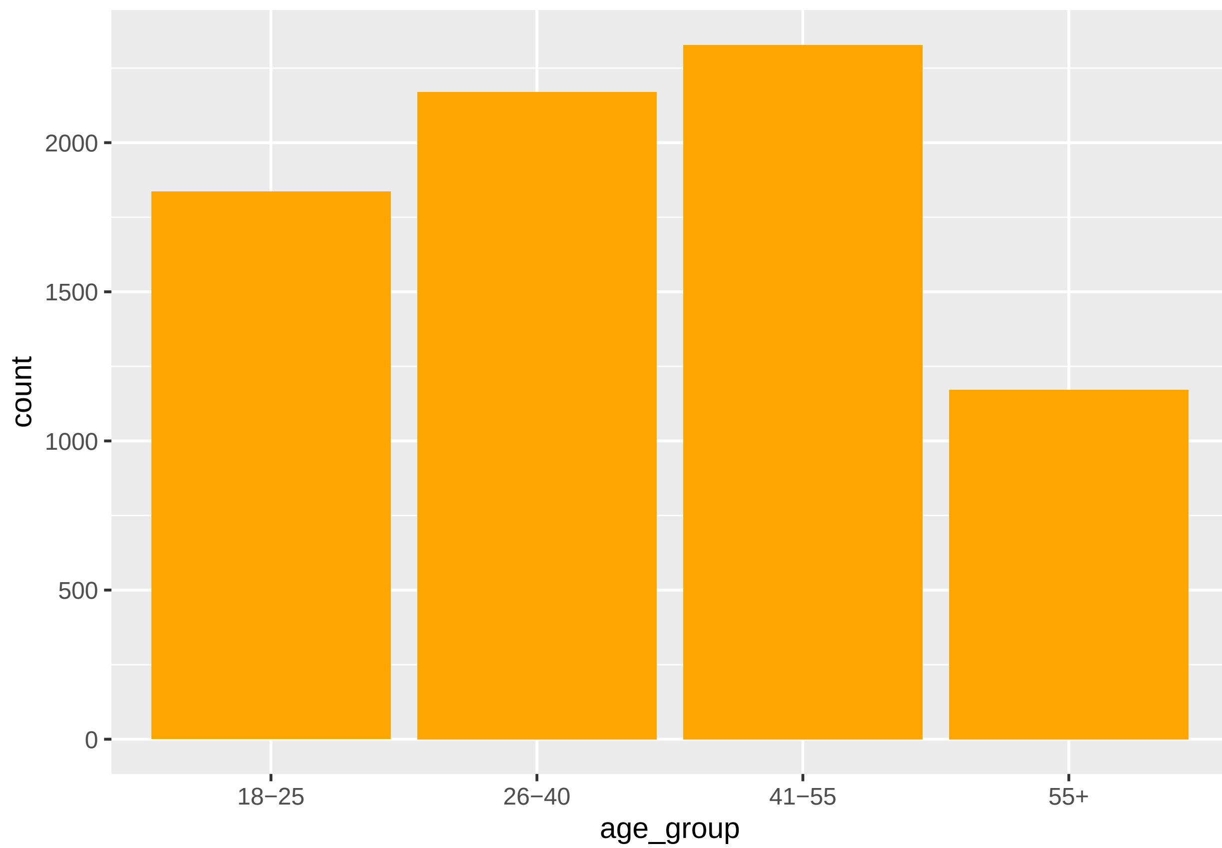

```r
#Creating a bar graph representing the count of people from urban or rural location types.

locationTypePlot <- ggplot(data = HMO_data_new) + aes(x=location_type) + geom_bar(fill='violet')
locationTypePlot
```




```r
#Creating a bar graph representing the distribution based on gender.

genderPlot <- ggplot(data = HMO_data_new) + aes(x=gender) + geom_bar(fill='lightgreen')
genderPlot
```




```
#For the next visualization, we will create a new column in our dataframe based on the existing 'age'
#column. This column will divide our dataset into 4 age categories: '18-25', '26-40', '41-55', and '55+

HMO_data_new <- HMO_data_new %>% mutate(age_group =
                     case_when(age <= 25 ~ "18-25",
                               age <= 40 ~ "26-40",
                               age <= 55 ~ "41-55",
                               age >55 ~ "55+")
)

#We will now look at how our dataset is divided based on the age groups.

ageGroupPlot <- ggplot(data = HMO_data_new) + aes(x=age_group) + geom_bar(fill='orange')
ageGroupPlot
```

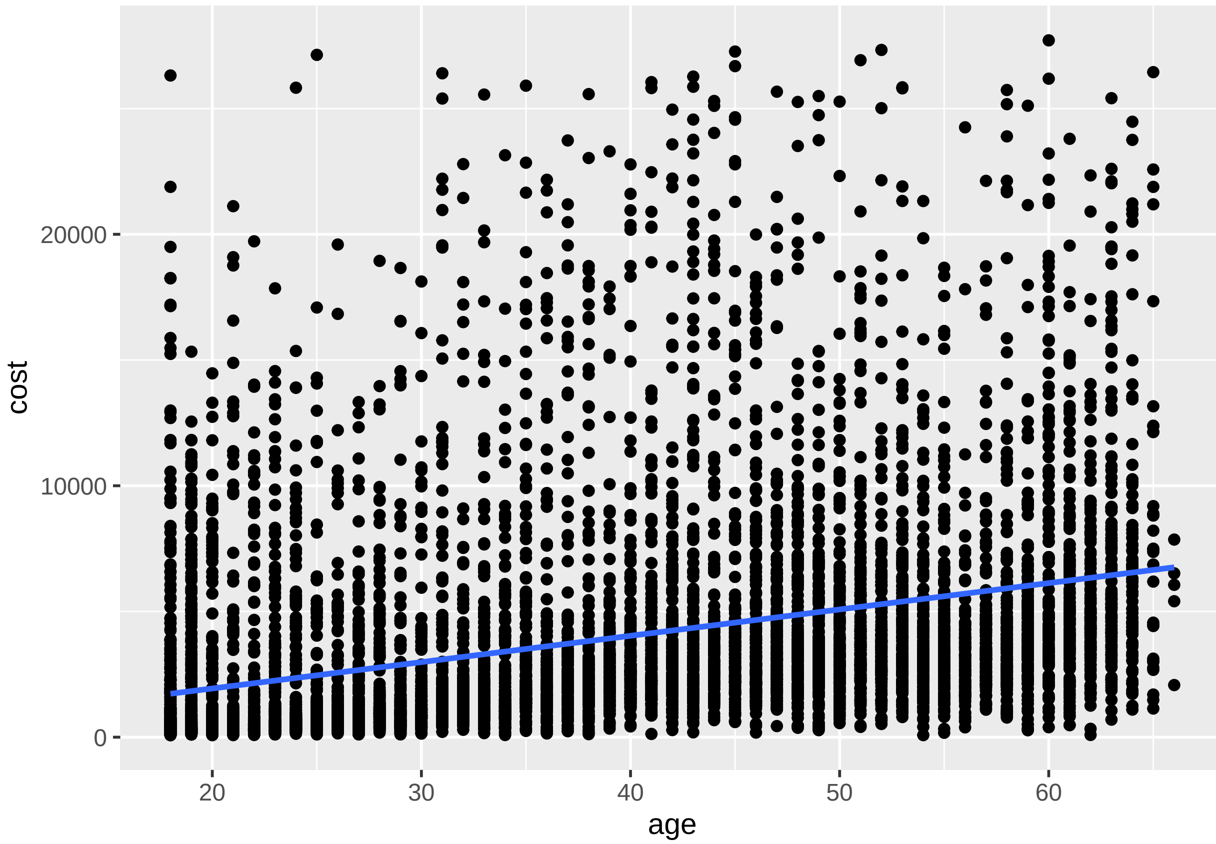


3. Identifying significant predictors

In this section, we will try to identify variables that have a significant impact on the cost variable. Firstly, we will visualize the relationship of multiple variables with cost variable using scatterplots. Then, we'll run regression models to study the dependancy of cost variables with multiple variables.

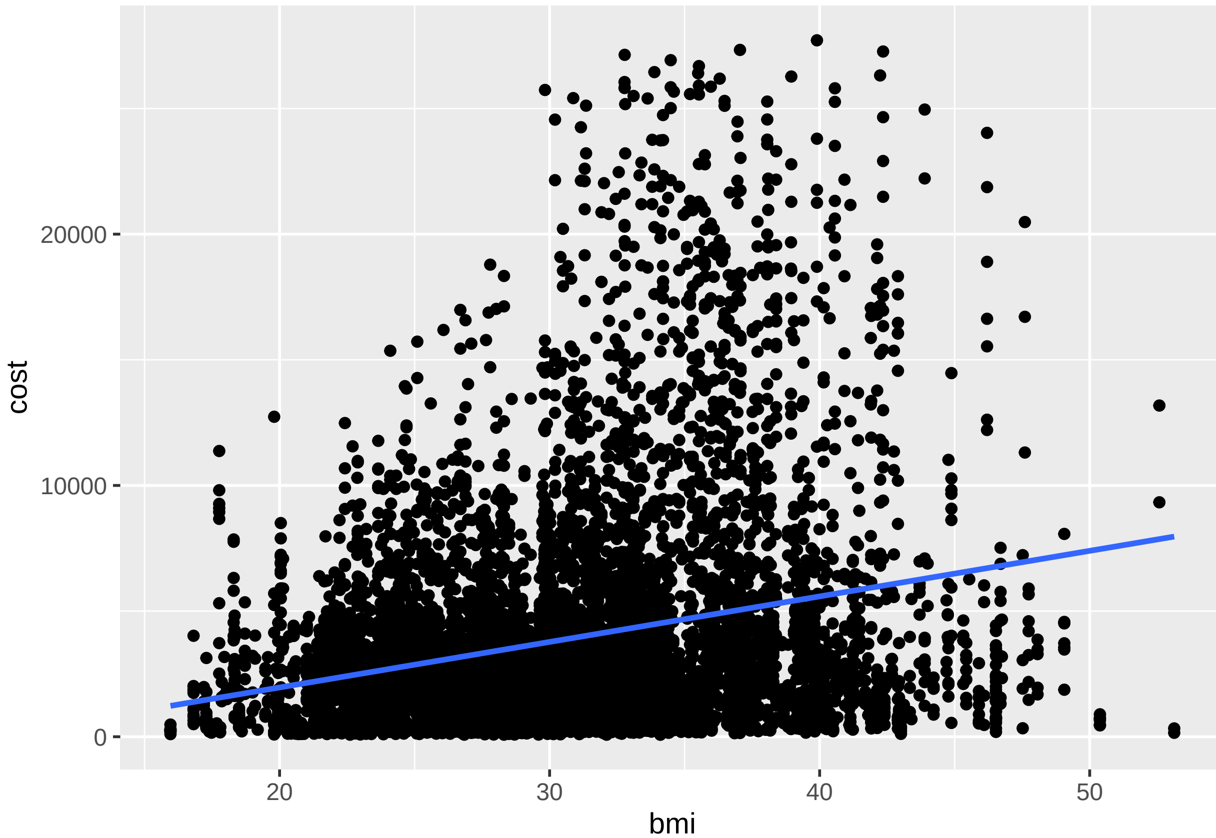
```
#We can observe that there is a slight linear relationship between age and cost variables.
#Although, the age variable alone wouldn't be sufficient to predict the cost values accurately.
ggplot(HMO_data_new, aes(x=age, y=cost)) + geom_point() + geom_smooth(method = "lm", se = FALSE)
```

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




```
#We can observe that there is a slight linear relationship between bmi and cost variables.
#Although, the bmi variable alone wouldn't be sufficient to predict the cost values accurately.
ggplot(HMO_data_new, aes(x=bmi, y=cost)) + geom_point() + geom_smooth(method = "lm", se = FALSE)
```

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



4. Identifying the threshold cost value to generate Expensive attribute

Here we will fixate on a cost value to create the expensive attribute column. Based on the statistical summary of the 'cost' attribute, we are picking the 3rd Quartile value for the cost column as the threshold value.

```
lm_hmo1 <- lm(cost ~ age + smoker + exercise + bmi + hypertension + yearly_physical,
             data = HMO_data_new)
summary(lm_hmo1)
```

```
##
## Call:
## lm(formula = cost ~ age + smoker + exercise + bmi + hypertension +
##     yearly_physical, data = HMO_data_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11516.7  -1411.3   -356.8    965.5  17558.5
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -8152.072    200.584 -40.642  < 2e-16 ***
## age                    99.064      2.328  42.559  < 2e-16 ***
## smokeryes            7155.439     83.439  85.756  < 2e-16 ***
## exerciseNot-Active   2124.804     76.277  27.856  < 2e-16 ***
## bmi                   166.570      5.524  30.151  < 2e-16 ***
## hypertension          285.472     82.310   3.468 0.000527 ***
```

```
## yearly_physicalYes   227.076     75.910   2.991 0.002786 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2846 on 7499 degrees of freedom
## Multiple R-squared:  0.5927, Adjusted R-squared:  0.5924
## F-statistic:  1819 on 6 and 7499 DF,  p-value: < 2.2e-16
```

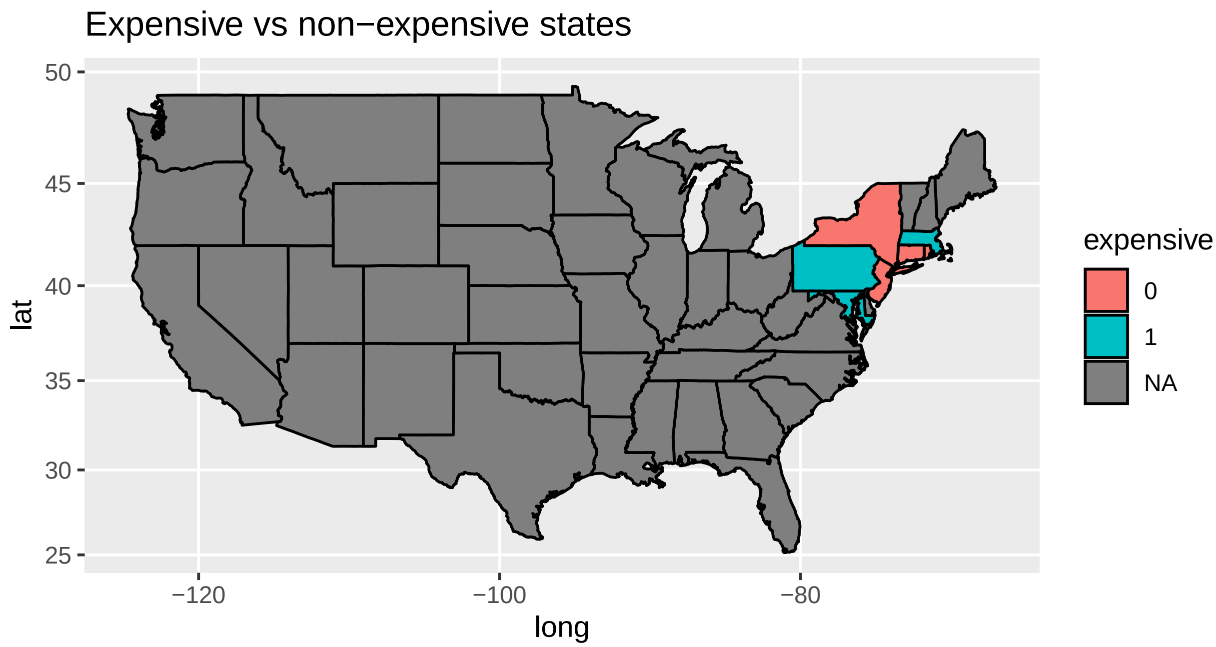
```
#Using linear regression model, we can predict the cost variable with almost 59% accuracy
#using the predictors such as age, smoker, exercise, bmi and hypertension.
```

```
#Next, we set the threshold for a person to be termed as 'expensive' as the cost value greater
#than the 75th percentile cost value (4748.00). All other records will be viewed as 'non-expensive'
HMO_t_quar <- quantile(HMO_data_new$cost, 0.75)
HMO_data_new$expensive <- ifelse(HMO_data_new$cost >= HMO_t_quar, 1, 0)
head(HMO_data_new)
```

```
## # A tibble: 6 x 16
##       X   age   bmi children smoker location     locat~1 educa~2 yearl~3 exerc~4
##   <dbl> <dbl> <dbl>    <dbl> <chr>  <chr>        <chr>   <chr>   <chr>   <chr>
## 1     1    18  27.9        0 yes    connecticut  Urban   Bachel~ No      Active
## 2     2    19  33.8        1 no     rhode island Urban   Bachel~ No      Not-Ac~
## 3     3    27  33          3 no     massachuset~ Urban   Master  No      Active
## 4     4    34  22.7        0 no     pennsylvania Country Master  No      Not-Ac~
## 5     5    32  28.9        0 no     pennsylvania Country PhD     No      Not-Ac~
## 6     7    47  33.4        1 no     pennsylvania Urban   Bachel~ No      Not-Ac~
## # ... with 6 more variables: married <chr>, hypertension <dbl>, gender <chr>,
## #   cost <dbl>, age_group <chr>, expensive <dbl>, and abbreviated variable
## #   names 1: location_type, 2: education_level, 3: yearly_physical, 4: exercise
```

```
#Converting the expensive variable into two level factor variable to run regression on it
HMO_data_new$expensive <- as.factor(HMO_data_new$expensive)
```

```
#Here, we use maps to display the states as expensive/non-expensive based on overall state average.
HMO_data_new_with_states <- merge(HMO_data_new, states, all.y=TRUE, by.x="location", by.y="region")

HMO_data_new_with_states <- HMO_data_new_with_states %>% arrange(order)

ggplot(HMO_data_new_with_states) + geom_polygon(color="black",
                 aes(x=long,y=lat,group=group,fill=expensive)) +
        ggtitle('Expensive vs non-expensive states') +
coord_map()
```



5. Dividing the dataset into training and testing set

We are dividing 70% of dataset into training data and 30% of dataset into testing data.

```
#In this section, we divide our dataset into training and testing data for further analysis.

#install.packages('caret')
library('caret')
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

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

```
trainlist <- createDataPartition(y=HMO_data_new$cost, p=0.70, list=FALSE)
trainSet <- HMO_data_new[trainlist,]
testSet <- HMO_data_new[-trainlist,]
```

```
str(trainSet)
```

```
## tibble [5,256 x 16] (S3: tbl_df/tbl/data.frame)
```

```
##  $ X              : num [1:5256] 2 3 5 7 9 10 11 12 13 14 ...
##  $ age            : num [1:5256] 19 27 32 47 36 59 24 61 22 57 ...
##  $ bmi            : num [1:5256] 33.8 33 28.9 33.4 29.8 ...
##  $ children       : num [1:5256] 1 3 0 1 2 0 0 0 0 0 ...
##  $ smoker         : chr [1:5256] "no" "no" "no" "no" ...
##  $ location       : chr [1:5256] "rhode island" "massachusetts" "pennsylvania" "pennsylvania" ...
##  $ location_type  : chr [1:5256] "Urban" "Urban" "Country" "Urban" ...
##  $ education_level: chr [1:5256] "Bachelor" "Master" "PhD" "Bachelor" ...
##  $ yearly_physical: chr [1:5256] "No" "No" "No" "No" ...
##  $ exercise       : chr [1:5256] "Not-Active" "Active" "Not-Active" "Not-Active" ...
##  $ married        : chr [1:5256] "Married" "Married" "Married" "Married" ...
##  $ hypertension   : num [1:5256] 0 0 0 0 0 1 0 0 0 0 ...
##  $ gender         : chr [1:5256] "male" "male" "male" "female" ...
##  $ cost           : num [1:5256] 602 576 836 3842 1304 ...
##  $ age_group      : chr [1:5256] "18-25" "26-40" "26-40" "41-55" ...
##  $ expensive      : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
```

```r
str(testSet)
```

```
## tibble [2,250 x 16] (S3: tbl_df/tbl/data.frame)
##  $ X              : num [1:2250] 1 4 18 19 25 26 28 30 34 37 ...
##  $ age            : num [1:2250] 18 34 23 57 37 58 56 31 63 63 ...
##  $ bmi            : num [1:2250] 27.9 22.7 23.8 40.3 28 ...
##  $ children       : num [1:2250] 0 0 0 0 2 3 2 2 0 3 ...
##  $ smoker         : chr [1:2250] "yes" "no" "no" "no" ...
##  $ location       : chr [1:2250] "connecticut" "pennsylvania" "massachusetts" "pennsylvania" ...
##  $ location_type  : chr [1:2250] "Urban" "Country" "Urban" "Urban" ...
##  $ education_level: chr [1:2250] "Bachelor" "Master" "No College Degree" "Bachelor" ...
##  $ yearly_physical: chr [1:2250] "No" "No" "No" "Yes" ...
##  $ exercise       : chr [1:2250] "Active" "Not-Active" "Active" "Active" ...
##  $ married        : chr [1:2250] "Married" "Married" "Married" "Not_Married" ...
##  $ hypertension   : num [1:2250] 0 1 0 0 0 0 0 0 0 0 ...
##  $ gender         : chr [1:2250] "female" "male" "male" "male" ...
##  $ cost           : num [1:2250] 1746 5562 294 1382 1496 ...
##  $ age_group      : chr [1:2250] "18-25" "26-40" "18-25" "55+" ...
##  $ expensive      : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 2 1 1 ...
```

6. Prediction analysis of Expensive variable using various models listed below:

SVM K-SVM

```r
#Now, we will use some models on our datasets to see which model fits the best on our data
#for future predictions.

# SVM Model
#First, we will use the SVM model where we apply the 'svmRadial' method. This is the most popular
#and most commonly used method as this is similar to a Gaussian distribution.
#install.packages('kernlab')
library('caret')

#Training the SVM model on our train dataset.
hmo_svm <- train(expensive ~. , data=trainSet, method = "svmRadial", trControl =
                   trainControl(method = "none"), preProcess = c("center", "scale"))
hmo_svm
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 5256 samples
##   15 predictor
##    2 classes: '0', '1'
##
## Pre-processing: centered (24), scaled (24)
## Resampling: None
```

```
#We will now test the model by predicting values in our test dataset.
pred_out <- predict(hmo_svm, newdata=testSet)
conf_matrix <- table(pred_out, testSet$expensive)

#Confusion matrix of the prediction.
conf_matrix
```

```
##
## pred_out    0    1
##        0 1685  147
##        1    2  416
```

```
#As we see here, the error (1-accuracy) rate is 93.51%

error <- (sum(conf_matrix) - sum(diag(conf_matrix)))/sum(conf_matrix)
accuracy <- 1- error
accuracy
```

```
## [1] 0.9337778
```

```
#Here we use the confusionMatrix function from the caret package.
confusionMatrix(pred_out, testSet$expensive)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1685  147
##          1    2  416
##
##                Accuracy : 0.9338
##                  95% CI : (0.9227, 0.9437)
##     No Information Rate : 0.7498
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8069
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9988
##             Specificity : 0.7389
##          Pos Pred Value : 0.9198
##          Neg Pred Value : 0.9952
```

```
##               Prevalence : 0.7498
##           Detection Rate : 0.7489
##     Detection Prevalence : 0.8142
##        Balanced Accuracy : 0.8689
##
##         'Positive' Class : 0
##
```

```r
# KSVM Model
#Second, we will use the K-SVM model. This is a Kernel-SVM approach. The difference here as
#compared to the normal SVM approach is that we specify the number of Kernels (points) that
#we will use closest to the current point to determine if they are similar or not for classification.
#This method uses the efficiency of SVM along with the accuracy of KNN (nearest neighbor) method.

library(kernlab)
```

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

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

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

```r
#Training the model on train dataset
ksvmHMO <- ksvm(expensive ~ ., data=trainSet, C=5, cross=3, prob.model=TRUE)
ksvmHMO
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 5
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.086533626941855
##
## Number of Support Vectors : 466
##
## Objective Function Value : -1031.399
## Training error : 0.002473
## Cross validation error : 0.016743
## Probability model included.
```

```r
#Now we predict using the model on our test dataset.
ksvmPred <- predict(ksvmHMO, newdata=testSet)
ksvm_conf_matrix <- table(ksvmPred, testSet$expensive)
#Confusion Matrix
ksvm_conf_matrix
```

```
##
## ksvmPred    0    1
##        0 1680   30
##        1    7  533
```

```
#As we see, the accuracy for this model increased as compared to SVM. It is around 98.75%

ksvmError <- (sum(ksvm_conf_matrix) - sum(diag(ksvm_conf_matrix)))/sum(ksvm_conf_matrix)
ksvmAccuracy <- 1- ksvmError
ksvmAccuracy
```

```
## [1] 0.9835556
```

```
#Using the confusionMatrix function to verify our result in the above block.
confusionMatrix(ksvmPred, testSet$expensive)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1680   30
##          1    7  533
##
##                Accuracy : 0.9836
##                  95% CI : (0.9774, 0.9884)
##     No Information Rate : 0.7498
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9556
##
##  Mcnemar's Test P-Value : 0.0002983
##
##             Sensitivity : 0.9959
##             Specificity : 0.9467
##          Pos Pred Value : 0.9825
##          Neg Pred Value : 0.9870
##              Prevalence : 0.7498
##          Detection Rate : 0.7467
##    Detection Prevalence : 0.7600
##       Balanced Accuracy : 0.9713
##
##        'Positive' Class : 0
##
```