

Final Project IST 687
Predicting factors affecting the healthcare cost.

Venkata Sai Mani Lakshmi Kavya Darsi
Pranjal Singh
Vipul Rajiv Sarode
Yoon Lee

TABLE OF CONTENTS

INTRODUCTION	3
BUSINESS UNDERSTANDING	3
BUSINESS QUESTIONS	4
PROJECT TECHNICAL DETAILS	4
PACKAGES USED	6
DATA ACQUISITION	7
DATA PREPROCESSING AND CLEANING	8
DATASET AND VARIABLE ANALYSIS	13
CORRELATION ANALYSIS	16
DATA VISUALIZATION	17
MODEL BUILDING	25
LINEAR AND MULTIPLE LINEAR REGRESSION	25
SUPPORT VECTOR MACHINE	28
TREE MODEL	31
SIGNIFICANT PREDICTORS BY MODEL	33
MODEL CONFUSION	33
NUMBERS AND INSIGHTS	35
ANALYSIS AND RECOMMENDATIONS	36
SHINY WEB APPS	38

- Introduction

Health has long been seen as being of utmost importance. The Health Maintenance Organization is a well-known insurance company. A Health Maintenance Organization (HMO) is a sort of medical insurance group that provides health care for a predetermined annual fee. Based on the elements we consider when developing a policy for a customer, insurance can be either expensive or inexpensive.

The dataset contains HMO data that needs to be analyzed to determine Expensive and Inexpensive. Some of these characteristics, such as hypertension, BMI, exercise, and smoking, among others, can have an impact on the Cost Variable. It is critical to predict if the cost will still prevail by evaluating historical data that discloses cost based on the alternatives selected by consumers when purchasing insurance, which can help the health sector make a better prediction.

Using analytics, our project attempts to deliver significant insights on the Health Maintenance Organization.

- Business Understanding

Predicting how much the person needs to pay based on the factors he has opted for is still a challenge for the people and health industry.

- Business Questions

1. Can we predict a pattern based on costs?
2. What threshold number determines whether a customer is paying high cost or low cost?
3. Which age group pays more than others?
4. Can BMI impact on the cost?
5. Does the place of residence affect the cost?
6. Does the lifestyle of a customer affect the cost?

7. Are health related habits such as exercising, smoking, or getting yearly physical related to the cost?

● Project Technical Details

The Health Maintenance Organization has data about people with different factors of their personal lifestyle and according to the details they are supposed to pay the insurance amount. There are some criteria and inbuilt relationships between data where we need to find the relations and need to explore for the future that which type of people pay more insurance and which type of people will not pay high insurance.

For this we need to explore data by feature engineering (Data cleaning and visualization), then need to do the appropriate models by finding significant factors i.e., the variables which have direct impact on the dependent variable. The models we used in this process are Linear Regression, Support Vector Machine and Trees.

Using different Models like Linear Regression, Support Vector Machine, Regression, Association Rule, we aim to provide the best model to provide

insights into what affects the person's cost variable whether they are in the expensive category or the non-expensive category.
are 7582 rows and 14 columns in our dataset

Looking at the columns we can see that 'Cost; is the Variable that tells us if a person is Expensive or Not and this is the column we used to create and predict a new column values Expensive.

X: Integer, Unique identifier for each person like an index or unique key for each person.

age: Integer, the age of the person calculated at the end of the year.

location: Categorical, the name of the state (in the United States) where the person lived (at the end of the year)

1. location type: Categorical, a description of the environment where the person lived (urban or country).

2. exercise: Categorical, “Not-Active” if the person did not exercise regularly during the year, “Active” if the person did exercise regularly during the year.
3. smoker: Categorical, “yes” if the person smoked during the past year, “no” if the person didn’t smoke during the year.
4. BMI: double, the body mass index of the person. The body mass index (BMI) is a measure that uses your height and weight to work out if your weight is healthy.
5. yearly physical: Categorical, “yes” if the person had a good visit (yearly physical) with their doctor during the year. “no” if the person did not have a good visit with their doctor.
6. Hypertension: “0” if the person did not have hypertension.
7. gender: Categorical, the gender of the person whether male and female
8. education level: Categorical, the amount of college education (“No College Degree”, “Bachelor”, “Master”, “PhD”).
9. married: Categorical, describing if the person is “Married” or “Not_Married”
10. Num children: Integer, Number of children
11. cost: integer, the total cost of health care for that person, during the past year.

● PACKAGES USED

The following packages were used:

- Tidyverse - Collection of R Packages for Data Manipulation.
- Ggplot2 - To create complex plots from data in a data frame
- Dplyr - It is used for grammar and data manipulation.
- Usmap - It can be used to transform shape files, spatial points, spatial data frames,
- imputeTS - It helps with the missing data problems.
- Zoo - To Provide an S3 class with methods for indexed totally ordered observations.
- Kernlab - To do Kernel-based Machine Learning Methods.
- Caret - It is used for classification and regression training.
- Cvms - For Calculating Confusion Matrix and Creating ROC curves.
- Tibble - To Manipulate and print data frames.

- E1071 - Provides functions for statistical and probabilistic algorithms like SVM.
- Caret - It is used for Classification and Regression Training
- Rpart - It builds for classification and regression trees.
- Rpart.plot - weighted percentage using the weights passed to rpart.

- Data Acquisition

```
>>> glimpse(health_raw)
```

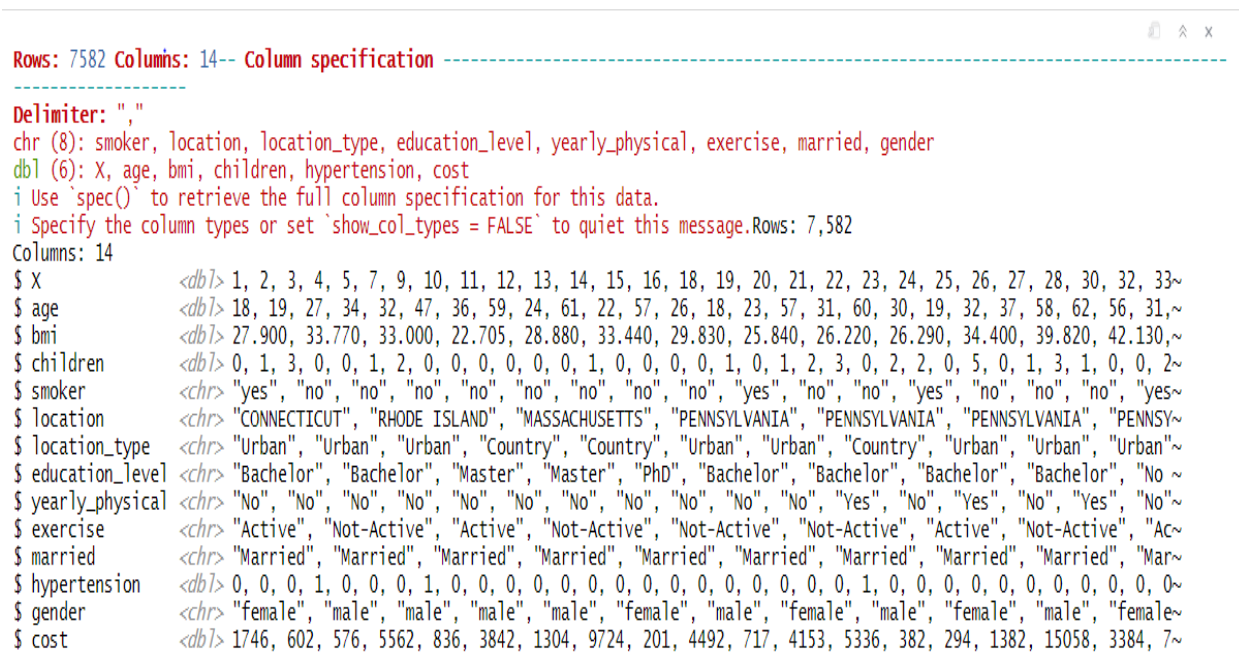
```
glimpse(health_raw)
#> # A tibble: 7,582 x 14
#>   X          age      bmi children smoker location location_type education_level yearly_physical exercise married hypertension gender cost
#>   <dbl> <dbl> <dbl> <dbl> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <dbl> <chr> <dbl>
#> 1  1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28...
#> 2  18, 19, 27, 34, 32, 47, 36, 59, 24, 61, 22, 57, 26, 18, 23, 57, 31, 60, 30, 19, 32, 37, 58,...
#> 3  27.900, 33.770, 33.000, 22.705, 28.880, 33.440, 29.830, 25.840, 26.220, 26.290, 34.400, 39...
#> 4  0, 1, 3, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 2, 3, 0, 2, 2, 0, 5, 0, 1, 3...
#> 5  "yes", "no", "no", "no", "no", "no", "no", "no", "no", "no", "yes", "no", "no", "yes", "no", "no"...
#> 6  "CONNECTICUT", "RHODE ISLAND", "MASSACHUSETTS", "PENNSYLVANIA", "PENNSYLVANIA", "PENNSYLVAN...
#> 7  "Urban", "Urban", "Urban", "Country", "Country", "Urban", "Urban", "Country", "Urban", "Urb...
#> 8  "Bachelor", "Bachelor", "Master", "Master", "PhD", "Bachelor", "Bachelor", "Bachelor", "Bac...
#> 9  "No", "No", "No", "No", "No", "No", "No", "No", "No", "No", "No", "No", "Yes", "No", "Yes", "No",...
#> 10 "Active", "Not-Active", "Active", "Not-Active", "Not-Active", "Not-Active", "Active", "Not-...
#> 11 "Married", "Married", "Married", "Married", "Married", "Married", "Married", "Married", "Ma...
#> 12 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0...
#> 13 "female", "male", "male", "male", "male", "female", "male", "female", "male", "female", "ma...
#> 14 1746, 602, 576, 5562, 836, 3842, 1304, 9724, 201, 4492, 717, 4153, 5336, 382, 294, 1382, 15...
```

- Using glimpse() the output showed that the data set had 14 variables (columns) with 7582 data.

- Data Pre-Processing and Cleaning

Code: library(tidyverse)

```
health_raw <- read.csv("https://intro-datascience.s3.us-east-2.amazonaws.com/HMO_data.csv")
>> glimpse(health_raw)
```



```
Rows: 7582 Columns: 14-- Column specification -----
Delimiter: ","
chr (8): smoker, location, location_type, education_level, yearly_physical, exercise, married, gender
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.Rows: 7,582
Columns: 14
$ X          <dbl> 1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 32, 33~
$ age        <dbl> 18, 19, 27, 34, 32, 47, 36, 59, 24, 61, 22, 57, 26, 18, 23, 57, 31, 60, 30, 19, 32, 37, 58, 62, 56, 31,~
$ bmi        <dbl> 27.900, 33.770, 33.000, 22.705, 28.880, 33.440, 29.830, 25.840, 26.220, 26.290, 34.400, 39.820, 42.130,~
$ children   <dbl> 0, 1, 3, 0, 0, 1, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 2, 3, 0, 2, 2, 0, 5, 0, 1, 3, 1, 0, 0, 2~
$ smoker     <chr> "yes", "no", "no", "no", "no", "no", "no", "no", "no", "no", "yes", "no", "no", "yes", "no", "no", "no", "yes~
$ location   <chr> "CONNECTICUT", "RHODE ISLAND", "MASSACHUSETTS", "PENNSYLVANIA", "PENNSYLVANIA", "PENNSYLVANIA", "PENNSY~
$ location_type <chr> "Urban", "Urban", "Urban", "Country", "Country", "Urban", "Urban", "Country", "Urban", "Urban", "Urban"~
$ education_level <chr> "Bachelor", "Bachelor", "Master", "Master", "PhD", "Bachelor", "Bachelor", "Bachelor", "Bachelor", "Bachelor", "No ~
$ yearly_physical <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No", "No", "No", "Yes", "No", "Yes", "No", "Yes", "No"~
$ exercise   <chr> "Active", "Not-Active", "Active", "Not-Active", "Not-Active", "Not-Active", "Not-Active", "Active", "Not-Active", "Ac~
$ married    <chr> "Married", "Married", "Married", "Married", "Married", "Married", "Married", "Married", "Married", "Married", "Mar~
$ hypertension <dbl> 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ gender     <chr> "female", "male", "male", "male", "male", "female", "male", "female", "male", "female", "male", "female", "male~
$ cost       <dbl> 1746, 602, 576, 5562, 836, 3842, 1304, 9724, 201, 4492, 717, 4153, 5336, 382, 294, 1382, 15058, 3384, 7~
```

- We brought the data in using read.csv() and took an overview of the data using glimpse().

```
>>> Summary(health_raw)
```

Rows: 7582 Columns: 14— Column specification

Delimiter: ","

chr (8): smoker, location, location_type, education_level, yearly_physical, exe...

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.

```
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
```

- There are two variables BMI (Body Mass Index) and hypertension with Not Available values.
- We can remove the rows with the filter() function from the tidyverse package.
- Code:

```
health=health_raw %>% filter(!(is.na(bmi)))
```

 - ```
health=health %>% filter(!(is.na(hypertension)))
```

With this code we can remove the rows which have Not available as values in BMI and hypertension variables.



After filtering or removing rows on doing summary we will get this output with zero NA's in the variables.

```
>>>dim(health_raw)
```

```
dim(health)
[1] 7424 14
```

```
>>>str(health_raw)
```

```
43 str(health_raw)
44
spec_tbl_ [7,582 × 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()
 ..)
```

To find the not available values in the dataset we use `sapply(is.na())`. From the output we got that there are zero Not Available values in the dataset.

```
>>> sapply(health,function(x) sum(is.na(x)))
```

```
sapply(health,function(x) sum(is.na(x)))
[[1]]
X age bmi children
0 0 0 0
smoker location location_type education_level
0 0 0 0
yearly_physical exercise married hypertension
0 0 0 0
gender cost
0 0
```

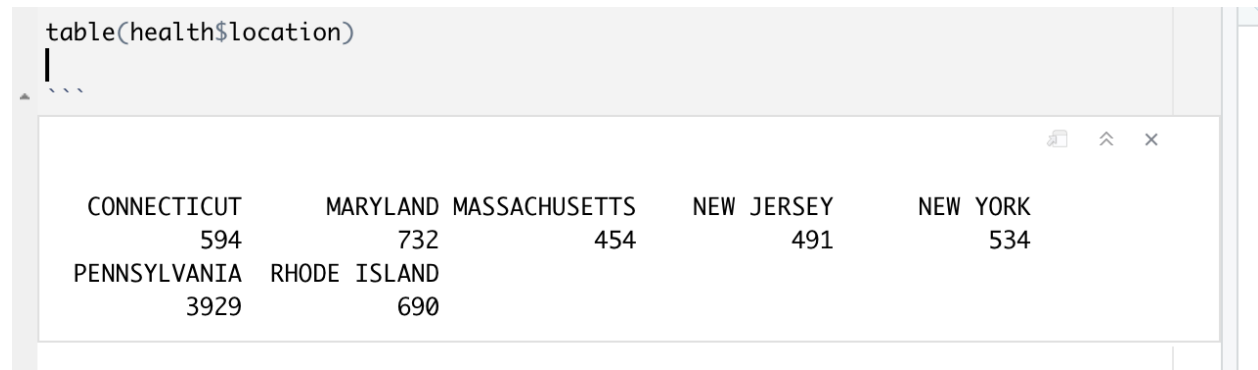
To find the null values in the dataset we use `sapply(is.null())`. From the result we found that there are 0 null values in the dataset.

```
>>>sapply(health, function(x) sum(is.null(x)))
```

```
sapply(health,function(x) sum(is.null(x)))
[[1]]
X age bmi children
0 0 0 0
smoker location location_type education_level
0 0 0 0
yearly_physical exercise married hypertension
0 0 0 0
gender cost
0 0
```

In the dataset there is a location for every person or customer in the HMO. To find the number of people in a particular location we use a `table(health$location)`. So, we can get the frequency of people from each place.

```
>>>table(health$location)
```



```
table(health$location)
```

|              |              |               |            |          |
|--------------|--------------|---------------|------------|----------|
| CONNECTICUT  | MARYLAND     | MASSACHUSETTS | NEW JERSEY | NEW YORK |
| 594          | 732          | 454           | 491        | 534      |
| PENNSYLVANIA | RHODE ISLAND |               |            |          |
| 3929         | 690          |               |            |          |

- We checked the data for missing or null values and found that there are no dummy values in locations and counties.

## ● DATASET AND VARIABLE ANALYSIS

To find the model for the dataset problem we examined each variable upon the cost variable. Depending on the cost variable we divide the dataset into two groups such as Expensive and Inexpensive.

The below is the summary for the cost of health dataset.

```
summary(health$cost)
#cor(health)
```

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max.  |
|------|---------|--------|------|---------|-------|
| 2    | 970     | 2504   | 4052 | 4778    | 55715 |

Based on the above output we created a variable called Expensive with a threshold value as 3rd quartile value. If the row value for the cost is above the threshold value, then it is considered as the Expensive and less than the threshold is inexpensive.

```
#Creating a New column
{r}
threshold=quantile(health$cost,probs=(.75))
health$Expensive <- ifelse(health$cost>=threshold, 1, 0)
glimpse(health)
```

After creating a value we get the glimpse of the dataset as following

```
Rows: 7,424
Columns: 15
$ X <dbl> 1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 24, 25~
$ age <dbl> 18, 19, 27, 34, 32, 47, 36, 59, 24, 61, 22, 57, 26, 18, 23, 57, 31, 60, 30,~
$ bmi <dbl> 27.900, 33.770, 33.000, 22.705, 28.880, 33.440, 29.830, 25.840, 26.220, 26.~
$ children <dbl> 0, 1, 3, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 2, 3, 0, 2, 2, ~
$ smoker <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, ~
$ location <dbl> 0, 6, 2, 5, 5, 5, 5, 5, 5, 0, 1, 1, 5, 5, 2, 5, 5, 5, 5, 3, 5, 5, 5, 5, 5, ~
$ location_type <dbl> 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, ~
$ education_level <dbl> 1, 1, 2, 2, 3, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 3, 3, 2, 0, 3, 1, 0, 2, 1, ~
$ yearly_physical <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, ~
$ exercise <dbl> 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, ~
$ married <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, ~
$ hypertension <dbl> 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, ~
$ gender <dbl> 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, ~
$ cost <dbl> 1746, 602, 576, 5562, 836, 3842, 1304, 9724, 201, 4492, 717, 4153, 5336, 38~
$ Expensive <dbl> 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, ~
```

There are so many Character values such as gender, location, location\_type, smoker, education\_level, yearly\_physical, exercise, married and location.

So, we change them according to the data model requirement by adding the numeric values for the respective column values in the form of 0 and 1.

```
#smoker
health$smoker<-str_replace_all(health$smoker,"no","0")
health$smoker<-str_replace_all(health$smoker,"yes","1")
#location_type
health$location_type<-str_replace_all(health$location_type,"Country","0")
health$location_type<-str_replace_all(health$location_type,"Urban","1")
#education_level
health$education_level<-str_replace_all(health$education_level,"No College Degree","0")
health$education_level<-str_replace_all(health$education_level,"Bachelor","1")
health$education_level<-str_replace_all(health$education_level,"Master","2")
health$education_level<-str_replace_all(health$education_level,"PhD","3")
#yearly_physical
health$yearly_physical<-str_replace_all(health$yearly_physical,"No","0")
health$yearly_physical<-str_replace_all(health$yearly_physical,"Yes","1")
#exercise
health$exercise<-str_replace_all(health$exercise,"Not-Active","0")
health$exercise<-str_replace_all(health$exercise,"Active","1")
#married
health$married<-str_replace_all(health$married,"Not-Married","0")
health$married<-str_replace_all(health$married,"Married","1")
#gender
Make sure to re-code female first
health$gender<-str_replace_all(health$gender,"female","1")
health$gender<-str_replace_all(health$gender,"male","0")

health$location<-str_replace_all(health$location,"CONNECTICUT","0")
health$location<-str_replace_all(health$location,"MARYLAND","1")
health$location<-str_replace_all(health$location,"MASSACHUSETTS","2")
health$location<-str_replace_all(health$location,"NEW JERSEY","3")
health$location<-str_replace_all(health$location,"NEW YORK","4")
health$location<-str_replace_all(health$location,"PENNSYLVANIA","5")
health$location<-str_replace_all(health$location,"RHODE ISLAND","6")
```

>>glimpse(health)

```
glimpse(health)
...

Rows: 7,424
Columns: 15
 $ x <dbl> 1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34~
 $ age <dbl> 18, 19, 27, 34, 32, 47, 36, 59, 24, 61, 22, 57, 26, 18, 23, 57, 31, 60, 30, 32, 37, 58, 62, 56, 31, 19,~
 $ bmi <dbl> 27.900, 33.770, 33.000, 22.705, 28.880, 33.440, 29.830, 25.840, 26.220, 26.290, 34.400, 39.820, 42.130,~
 $ children <dbl> 0, 1, 3, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 2, 3, 0, 2, 2, 0, 5, 0, 1, 3, 0, 0, 2, 1, 2~
 $ smoker <chr> "1", "0", "0", "0", "0", "0", "0", "0", "0", "1", "0", "0", "1", "0", "0", "0", "1", "0", "0", "1", "0", "0", "1", "0"~
 $ location <chr> "0", "6", "2", "5", "5", "5", "5", "5", "5", "0", "1", "1", "5", "5", "2", "5", "5", "5", "5", "3", "5", "5", "5"~
 $ location_type <chr> "1", "1", "1", "0", "0", "1", "1", "0", "1", "1", "1", "1", "1", "0", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1"~
 $ education_level <chr> "1", "1", "2", "2", "3", "1", "1", "1", "1", "0", "1", "1", "1", "0", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1"~
 $ yearly_physical <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "1", "0", "1", "0", "1", "0", "1", "0", "1", "0", "1", "0"~
 $ exercise <chr> "1", "0", "1", "0", "0", "0", "1", "0", "1", "1", "0", "0", "1", "0", "1", "0", "1", "1", "0", "1", "1", "0", "1", "1"~
 $ married <chr> "1", "1"~
 $ hypertension <dbl> 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
 $ gender <chr> "1", "0", "0", "0", "0", "1", "0", "1", "0", "1", "0", "1", "0", "0", "1", "0", "0", "1", "0", "1", "0", "1", "0", "0"~
 $ cost <dbl> 1746, 602, 576, 5562, 836, 3842, 1304, 9724, 201, 4492, 717, 4153, 5336, 382, 294, 1382, 15058, 3384, 7~
 $ Expensive <dbl> 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0~
```

The Variable values are still in character values. So, we change them into numerics by using the `as.numeric()` function.

```
health$location <- as.numeric(health$location)
health$smoker <- as.numeric(health$smoker)
health$location_type<- as.numeric(health$location_type)
health$education_level <- as.numeric(health$education_level)
health$yearly_physical <- as.numeric(health$yearly_physical)
health$married <- as.numeric(health$married)
health$gender <- as.numeric(health$gender)
health$exercise <- as.numeric(health$exercise)
health$bmi=as.numeric(health$bmi)
health$Expensive=as.numeric(health$Expensive)
health$X <- as.numeric(health$X)
```

`>>glimpse(health)`

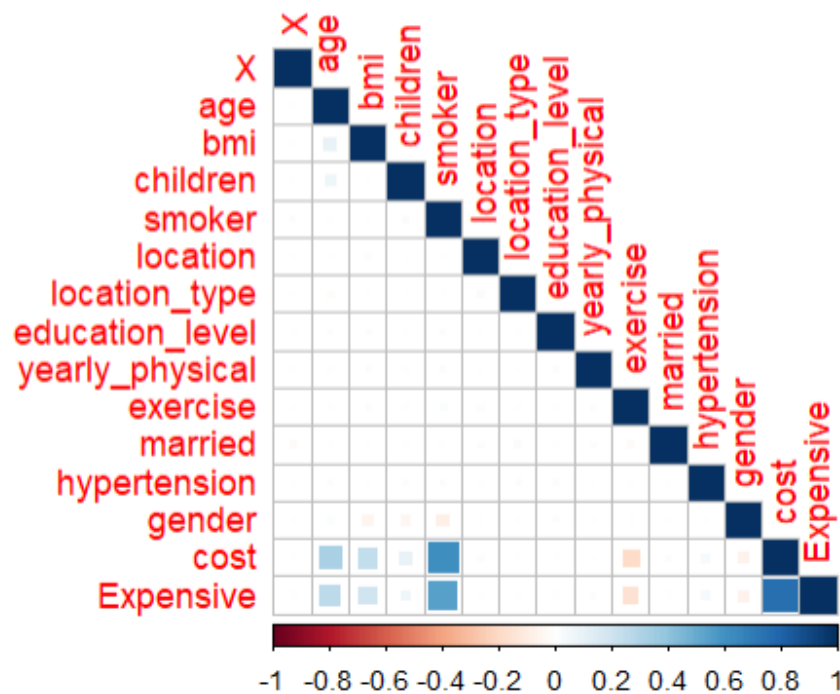
```
Rows: 7,424
Columns: 15
$ X <dbl> 1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34~
$ age <dbl> 18, 19, 27, 34, 32, 47, 36, 59, 24, 61, 22, 57, 26, 18, 23, 57, 31, 60, 30, 32, 37, 58, 62, 56, 31, 19,~
$ bmi <dbl> 27.900, 33.770, 33.000, 22.705, 28.880, 33.440, 29.830, 25.840, 26.220, 26.290, 34.400, 39.820, 42.130,~
$ children <dbl> 0, 1, 3, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 2, 3, 0, 2, 2, 0, 5, 0, 1, 3, 0, 0, 2, 1, 2~
$ smoker <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,~
$ location <dbl> 0, 6, 2, 5, 5, 5, 5, 5, 5, 0, 1, 1, 5, 5, 2, 5, 5, 5, 5, 3, 5, 5, 5, 5, 5, 5, 1, 2, 5, 2, 5, 5, 5, 5, 2~
$ location_type <dbl> 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1~
$ education_level <dbl> 1, 1, 2, 2, 3, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 3, 3, 2, 0, 3, 1, 0, 2, 1, 2, 1, 2, 1, 1, 0, 1, 2, 1, 2~
$ yearly_physical <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0~
$ exercise <dbl> 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0~
$ married <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1~
$ hypertension <dbl> 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ gender <dbl> 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1~
$ cost <dbl> 1746, 602, 576, 5562, 836, 3842, 1304, 9724, 201, 4492, 717, 4153, 5336, 382, 294, 1382, 15058, 3384, 7~
$ Expensive <dbl> 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0~
```

After changing each and every variable to numeric we get the result like above on applying `glimpse(health)`

## Correlational Analysis

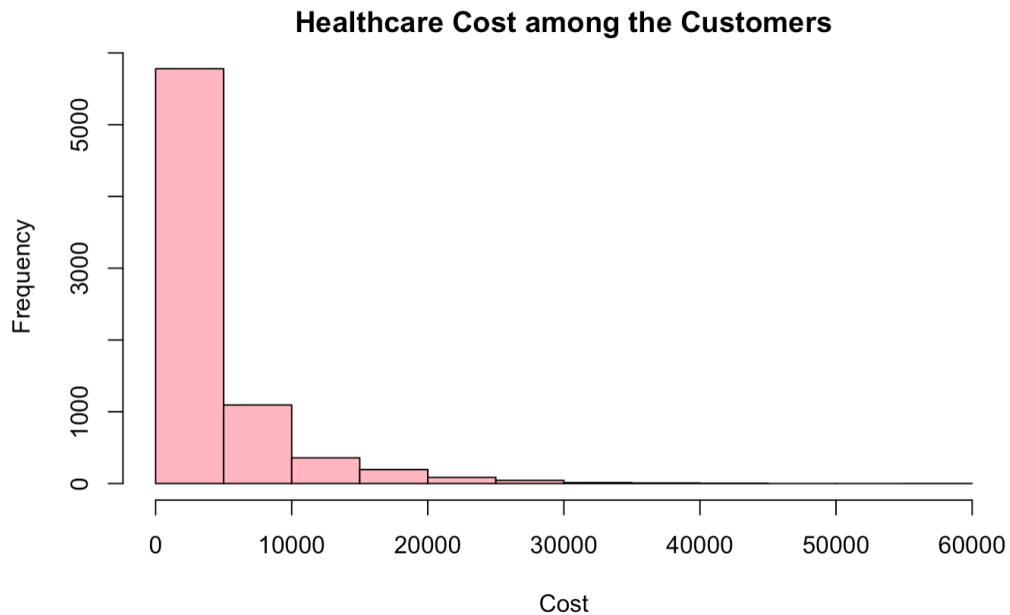
- The Correlational Analysis of the dataset gives us the relation among the data variables.

```
library(corrplot)
health_numeric = health %>% filter()
health_numeric = health %>%
 mutate_all(as.numeric)
corrplot(cor(health_numeric), type = "lower", method = 'square')
#view(health)
```



- VARIABLE ANALYSIS:

Define Expensive:

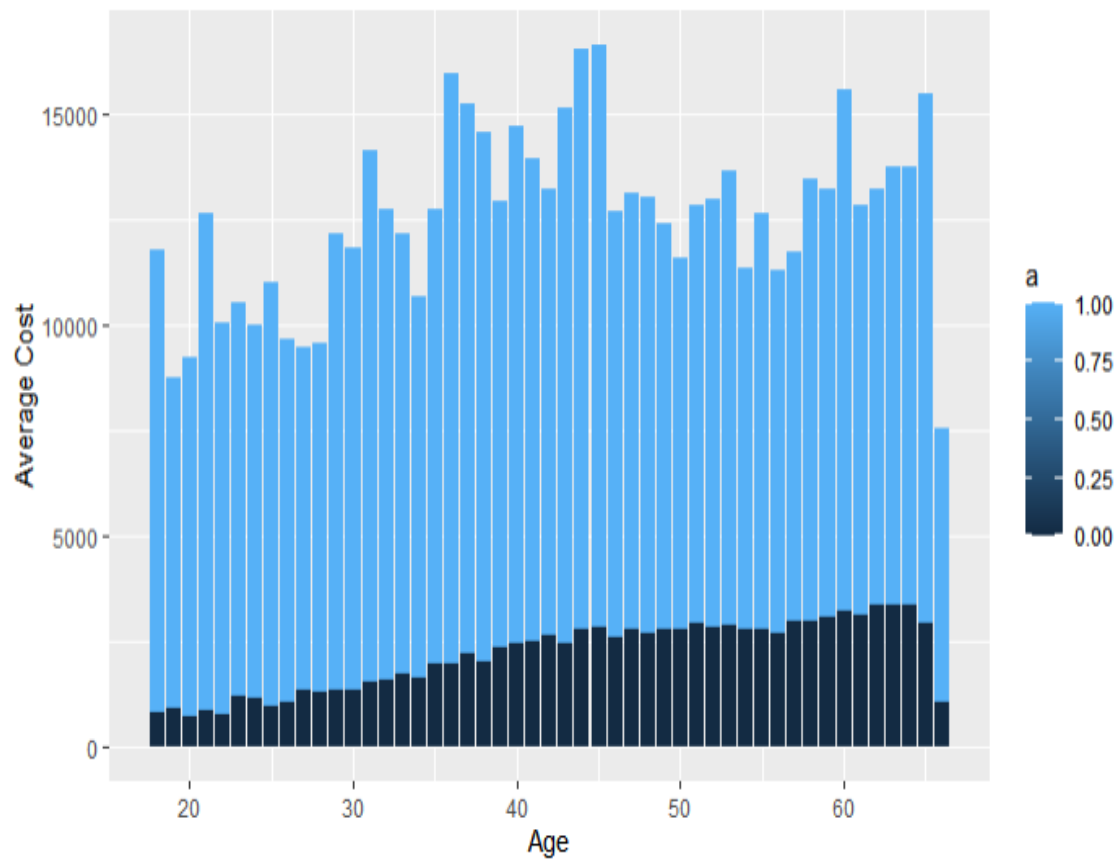
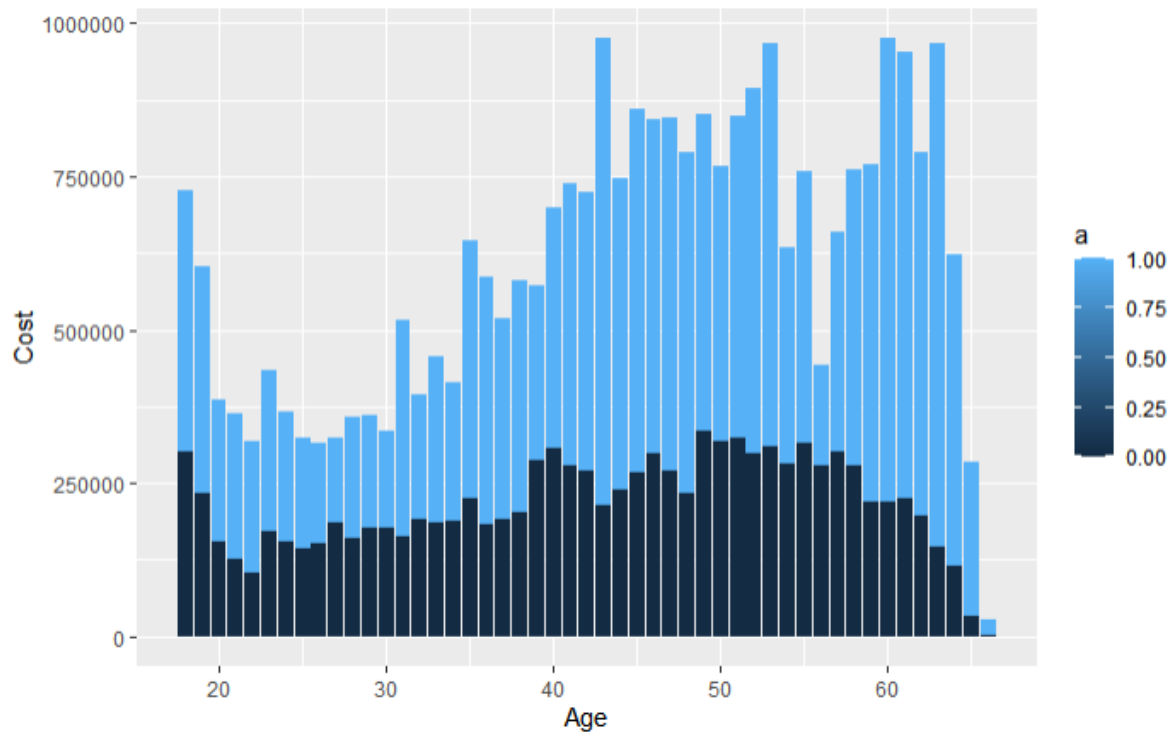


- The distribution of healthcare costs was right-skewed.
- We defined 'Expensive' as being above the 3rd quartile (\$4775).

For the purpose of data visualization, we are looking at the various metrics grouped by our 'Expensive' threshold and the metric, and we visualize the cumulative cost and the average cost for all the members.

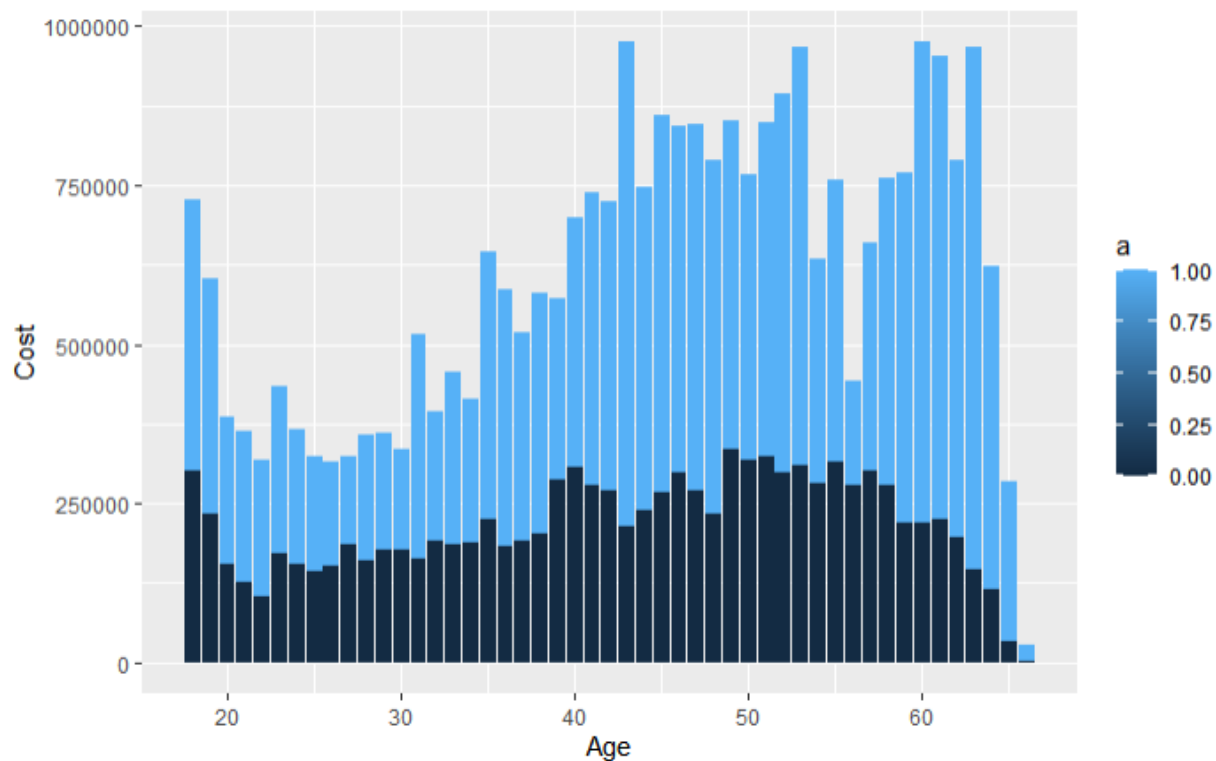


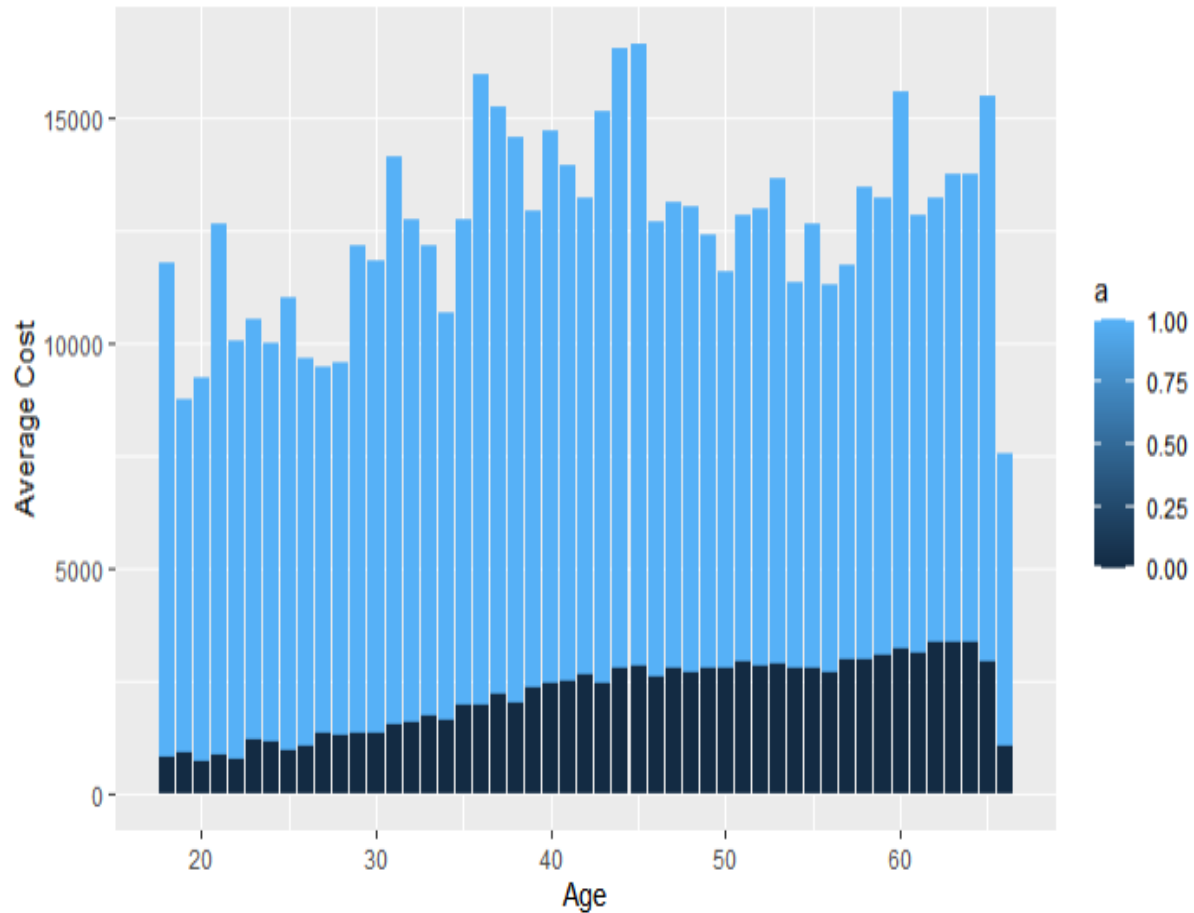
## Age VS Health Care Cost



- Excluding some outliers, we observe from the data that the cost of healthcare claims is the lowest around the age of 26 and increases drastically after the age of 35.
- We also observe that the high-cost claims make up a larger share of the cost as the age increases. The ratio of non-expensive claims increases as the age nears 55; and then the expensive claims increase drastically once the customers retire
- We also observe that the average amount of claims increases with age, with the share expensive claims increasing with an increase in age.

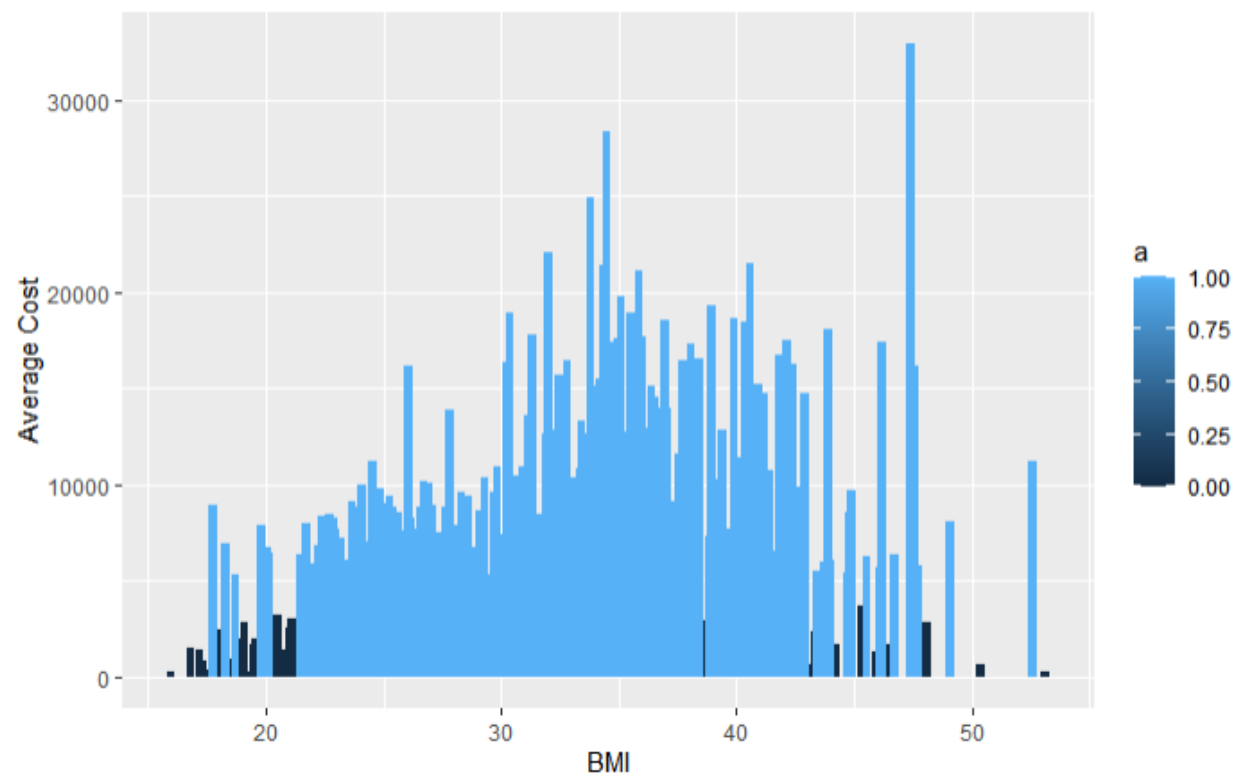
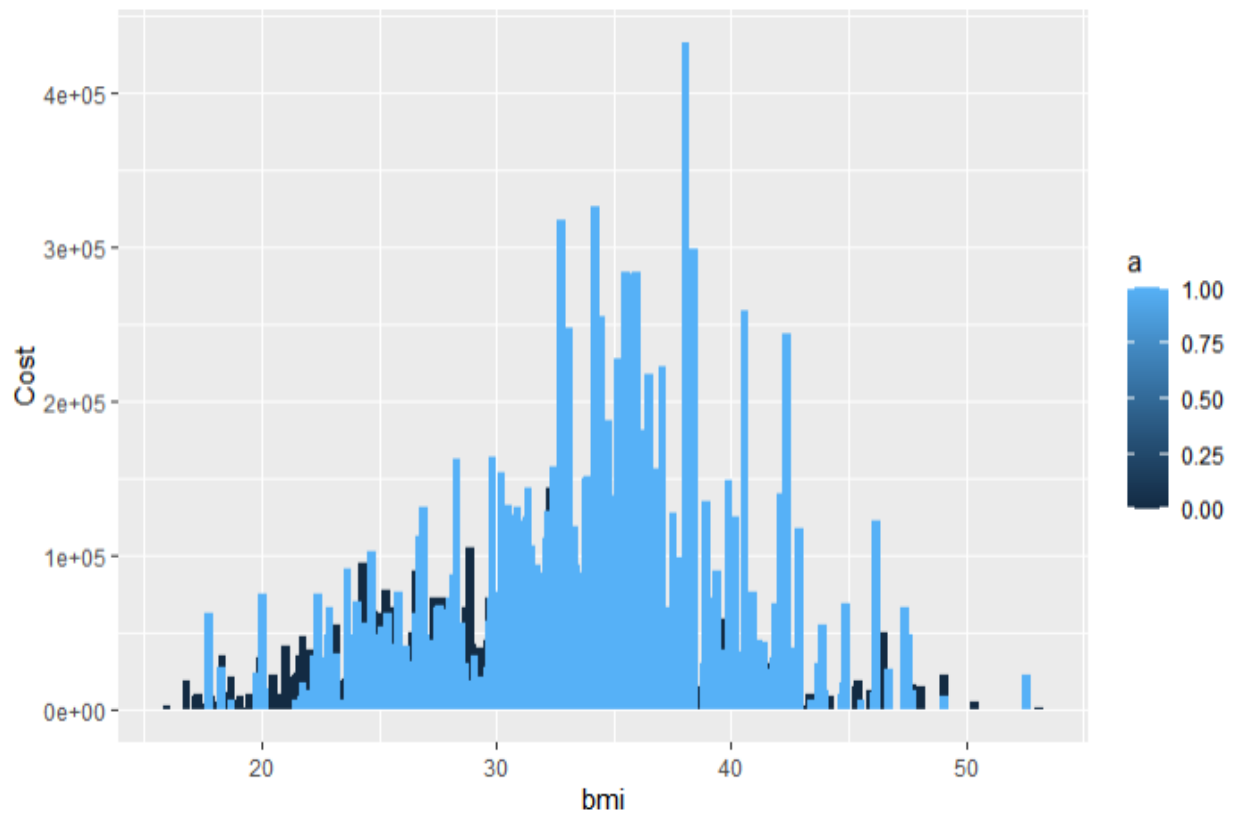
### Age Vs Health Care Cost





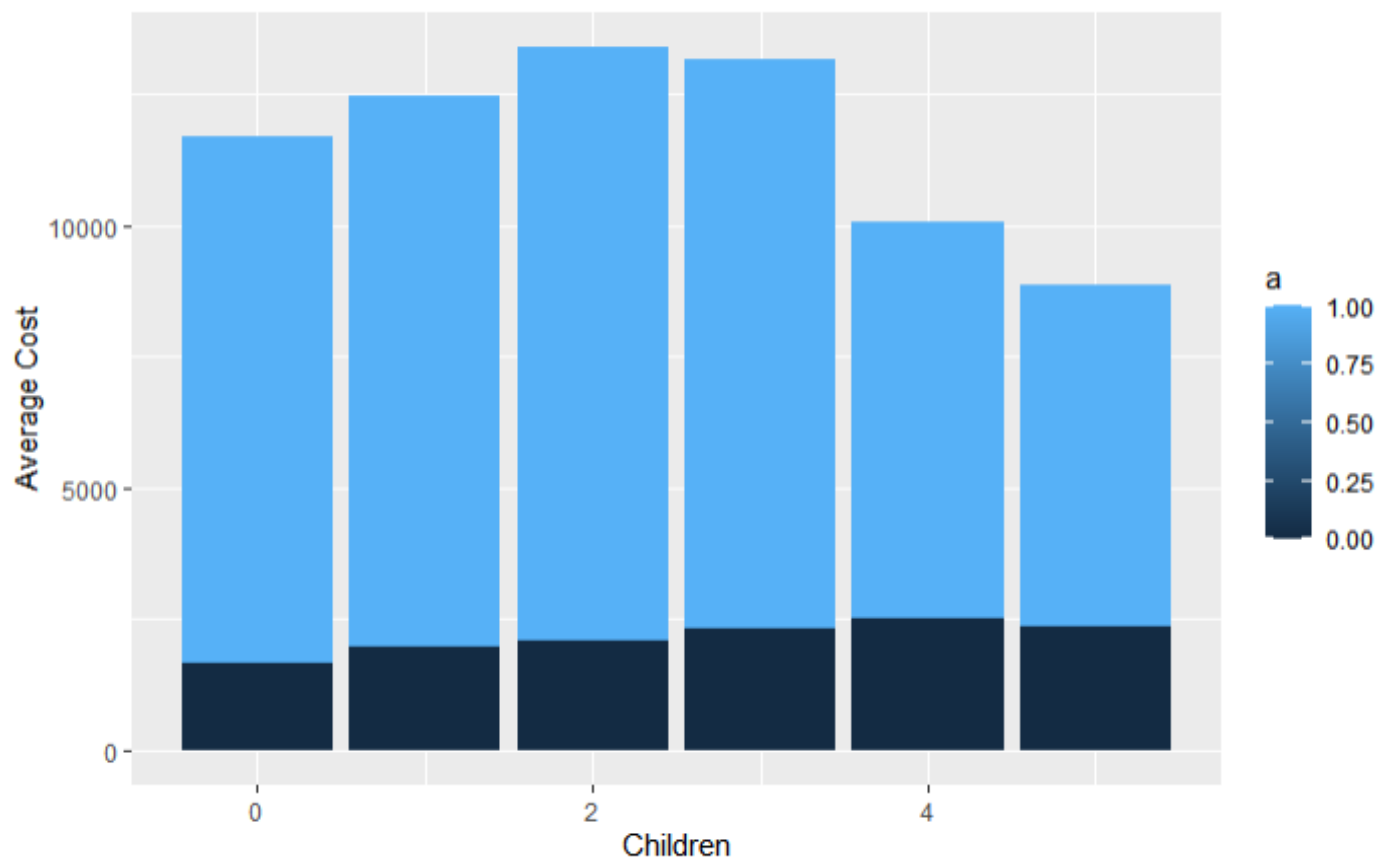
- We also observe that the high-cost claims make up a larger share of the cost as the age increases
- Excluding some outliers, we observe from the data that the cost of healthcare claims is the lowest around the age of 26 and increases drastically after the age of 35.
- The average amount also increases with age with the overall average and ratio of expensive claims peaking around the age of 35-45

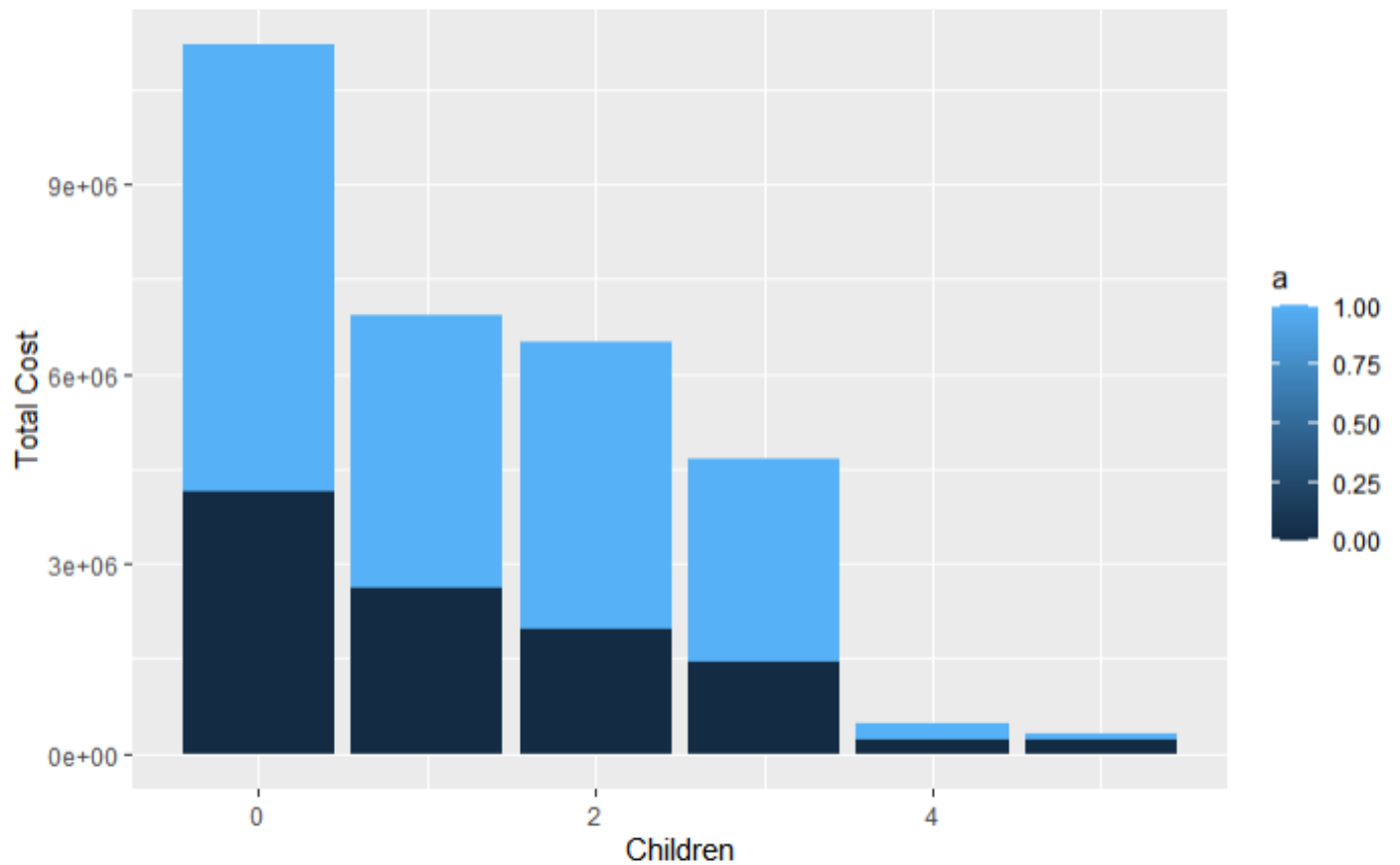
## Body Mass Index Vs Health Care Cost



- Before the BMI index of 30, we observe that the cost of claims is relatively lower, with low-cost claims making up a significant share of the claims.
- Post BMI index of 30, we observe a steep increase in the large value claims with the peak at 38, with almost negligible small value claims till the BMI of 45. Post BMI index of 45, we observe a sharp decline in the claim amount.
- Excluding a few outliers, the average cost is normally distributed across the BMI index, with high value claims taking up the major share of the claims between the age of 35-45.

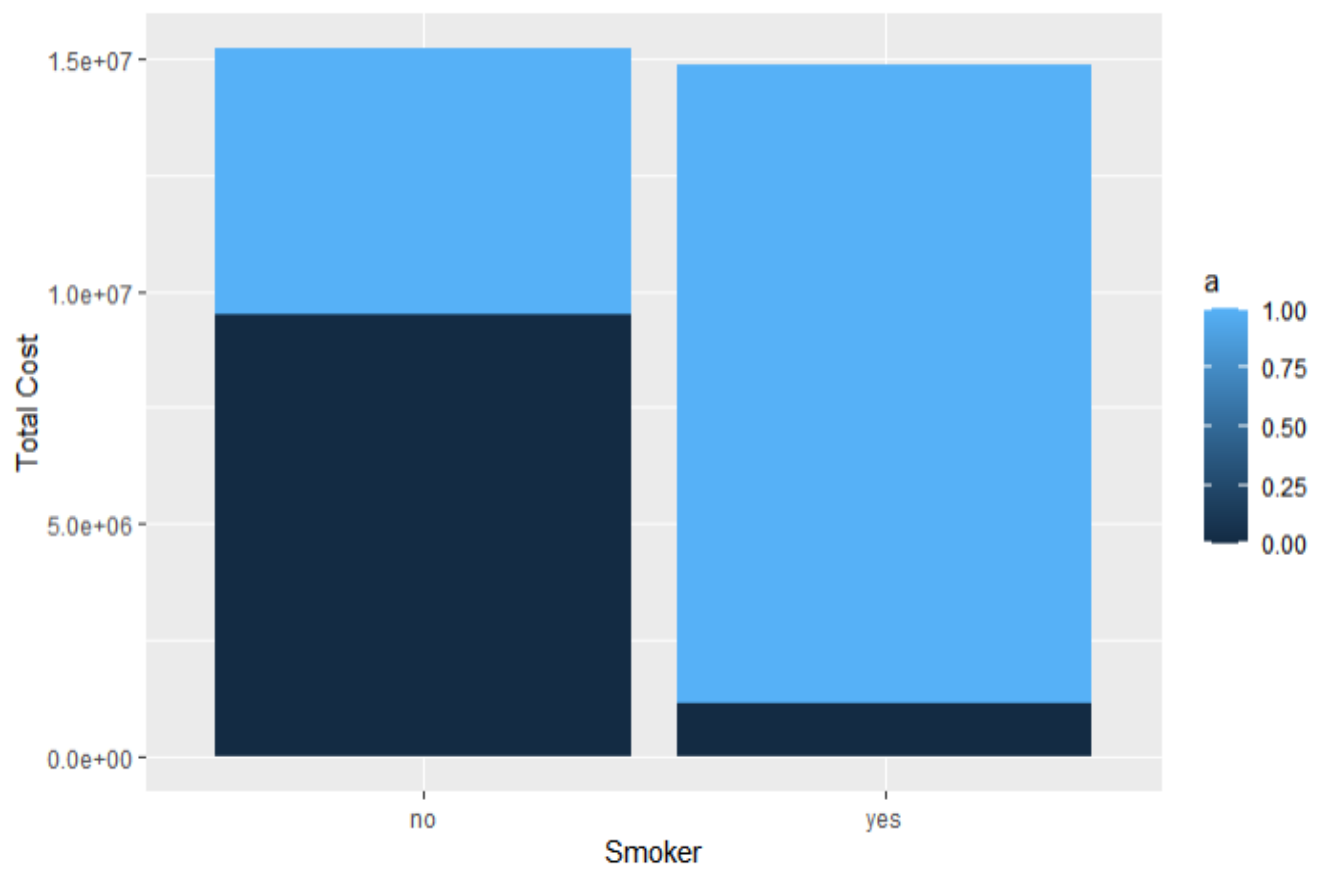
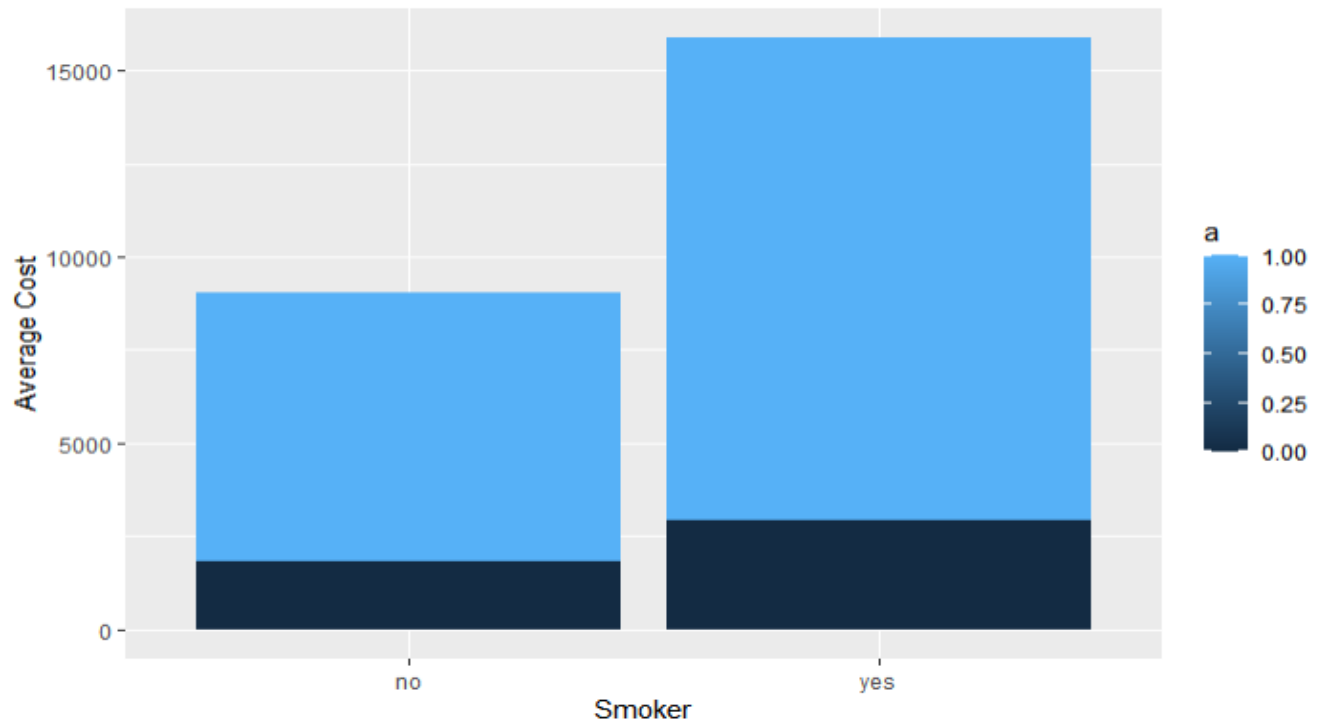
### Children to the Customers VS HealthCare Cost





- Here we ignore the large contribution of people with 0-1 children look at the average of the claim amount as most of the population tends to have less than 2 children.
- From the visualization, we observe that people with 2-3 children paid the highest average claim amount, with a larger share of 'Expensive' claims compared to the others.

## Smoker Vs Health Care Cost

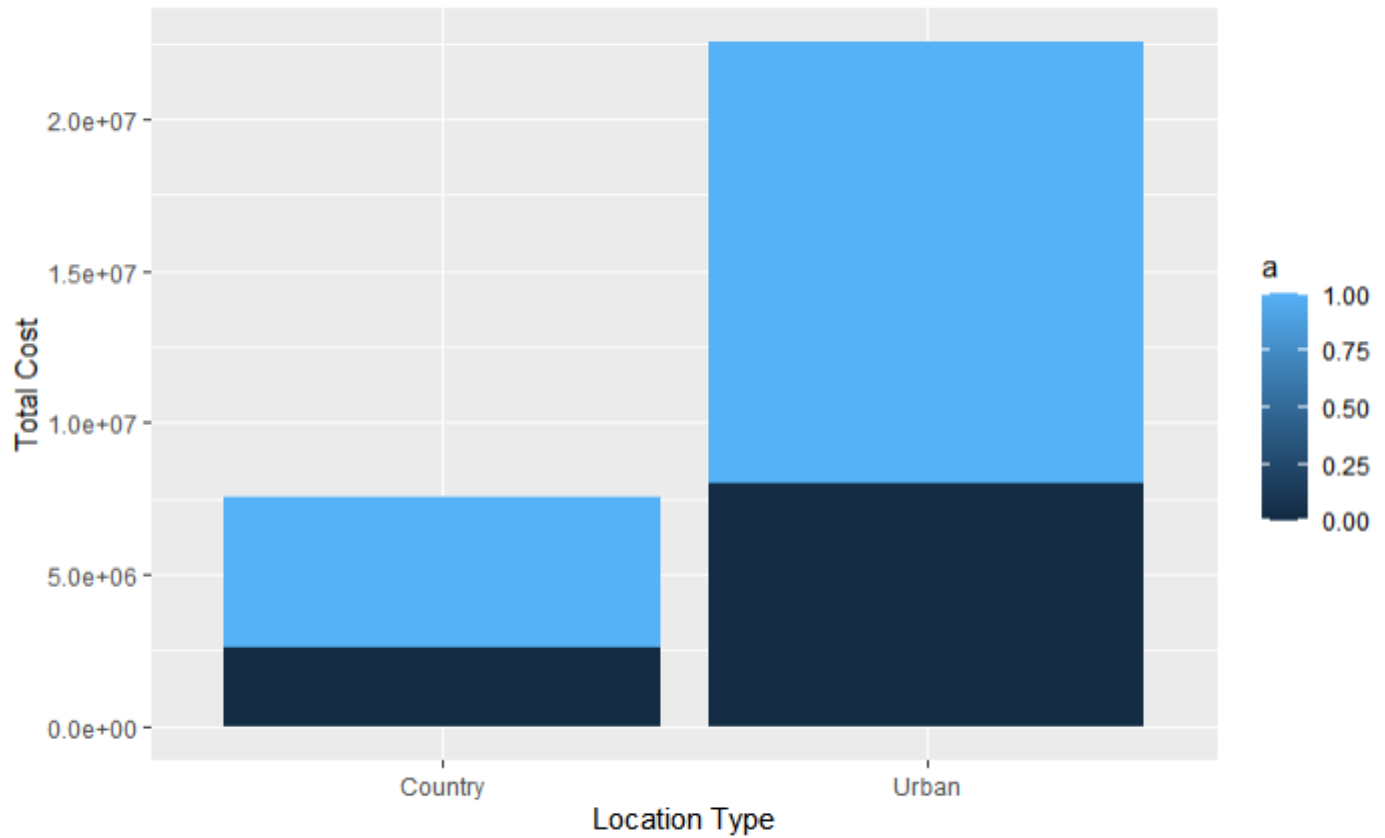


- We see that smokers tend to make more expensive average claims as compared to nonsmokers.
- But the total claim amount for smokers and non-smokers is almost the same with many non-expensive claims for non-smokers which can be attributed to the large number of non-smokers in the dataset.

### Customer Lives Vs Health Care Cost

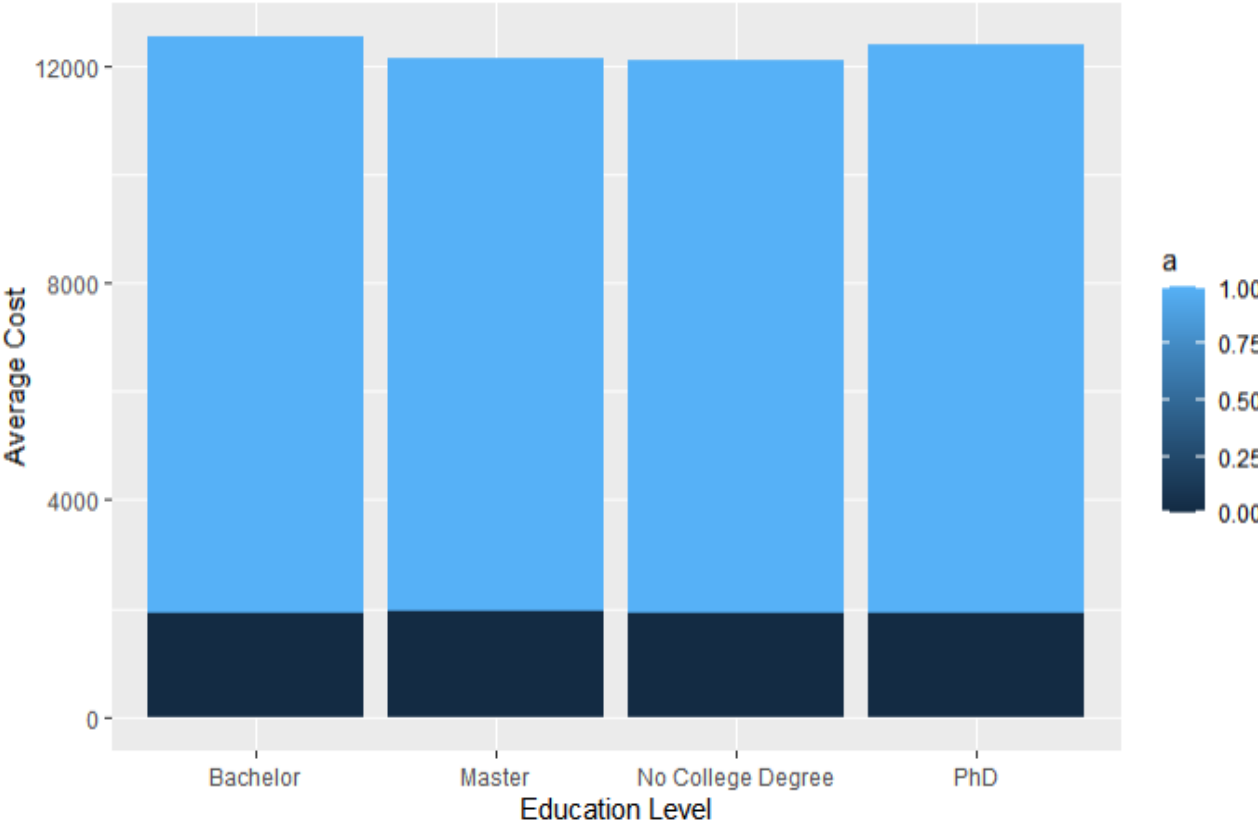
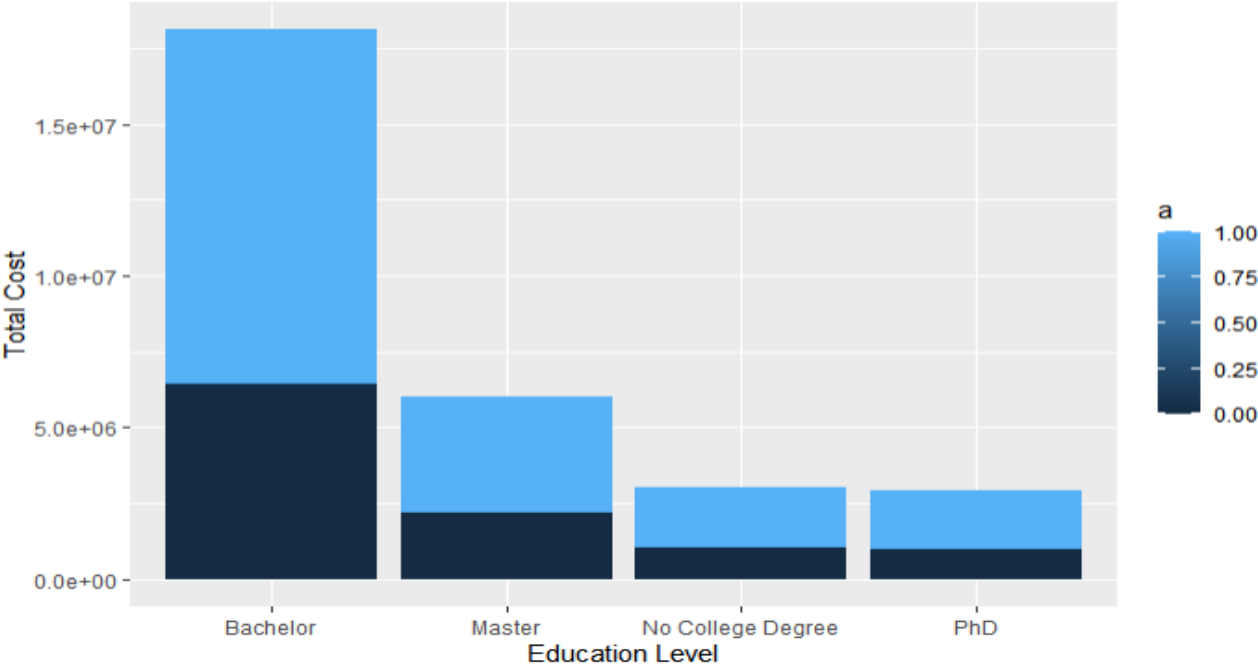






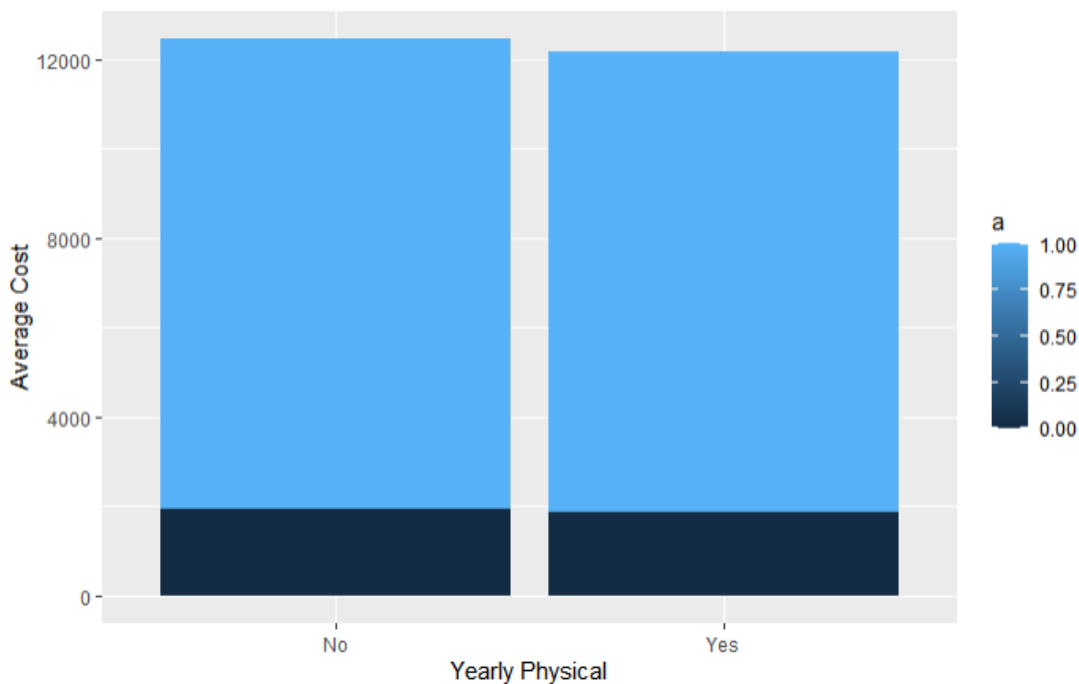
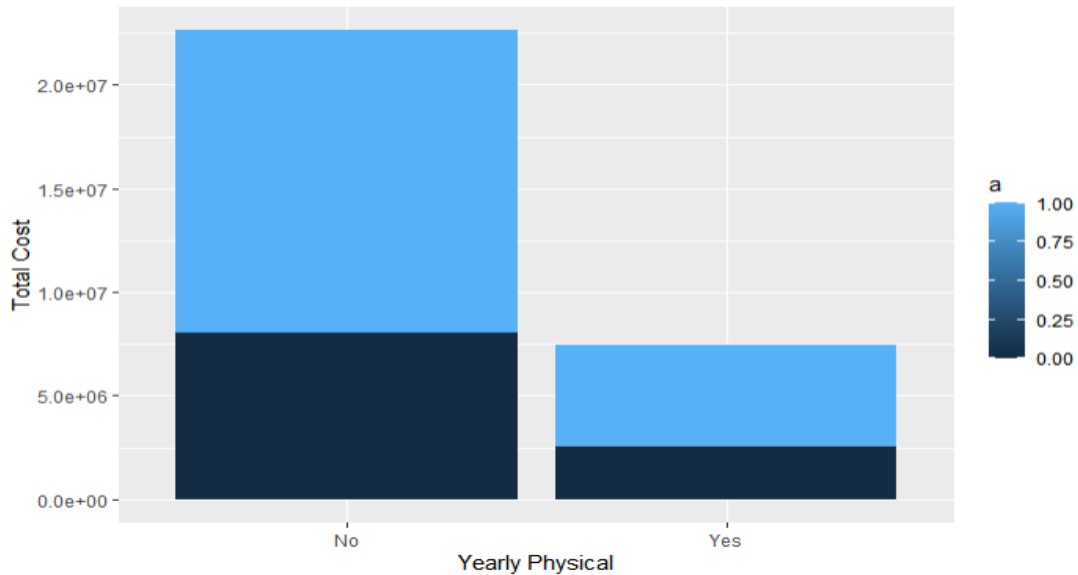
- Here it is observed that the average claim amount is almost the same across the location type.
- However, the total claim amount in urban areas is much higher than the country due to a higher count of claims coming from the urban areas.

Education Level Vs Health Care Cost



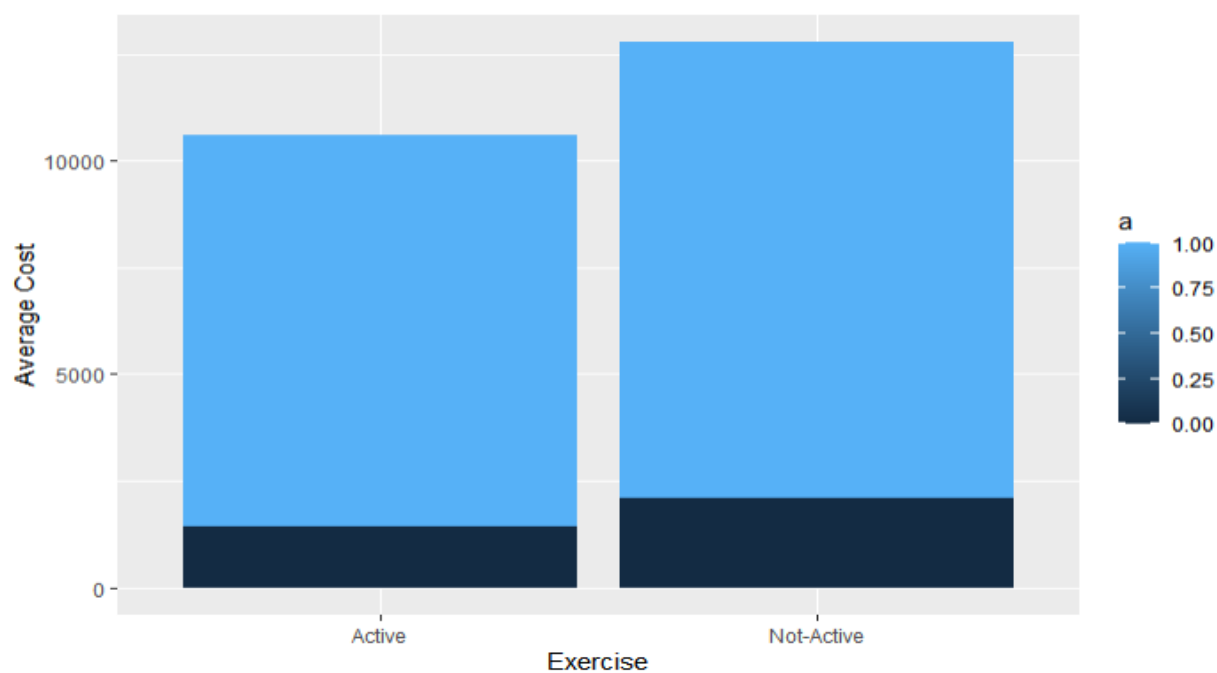
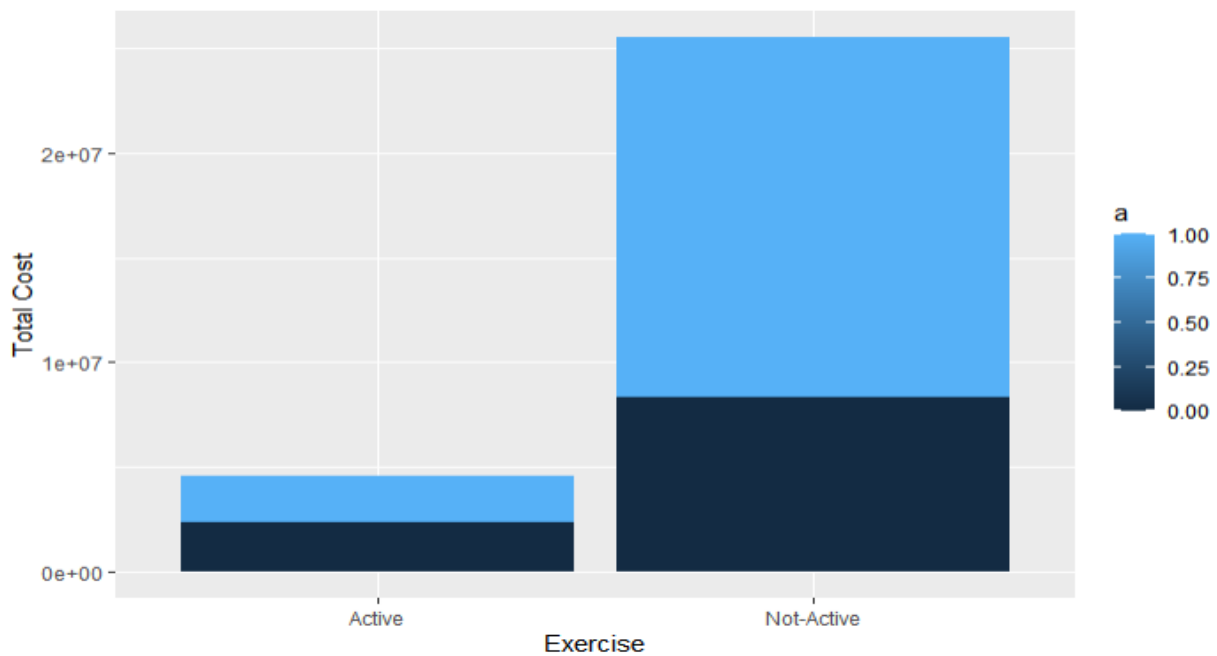
- It is observed that the average amount of claims is almost the same across all education levels, but the total claim amount of bachelor's and Master's student is much higher compared to the others.
- This can be associated with the higher number of claims made by the patients in these segments.

### Yearly Physical Related to Health Care Cost



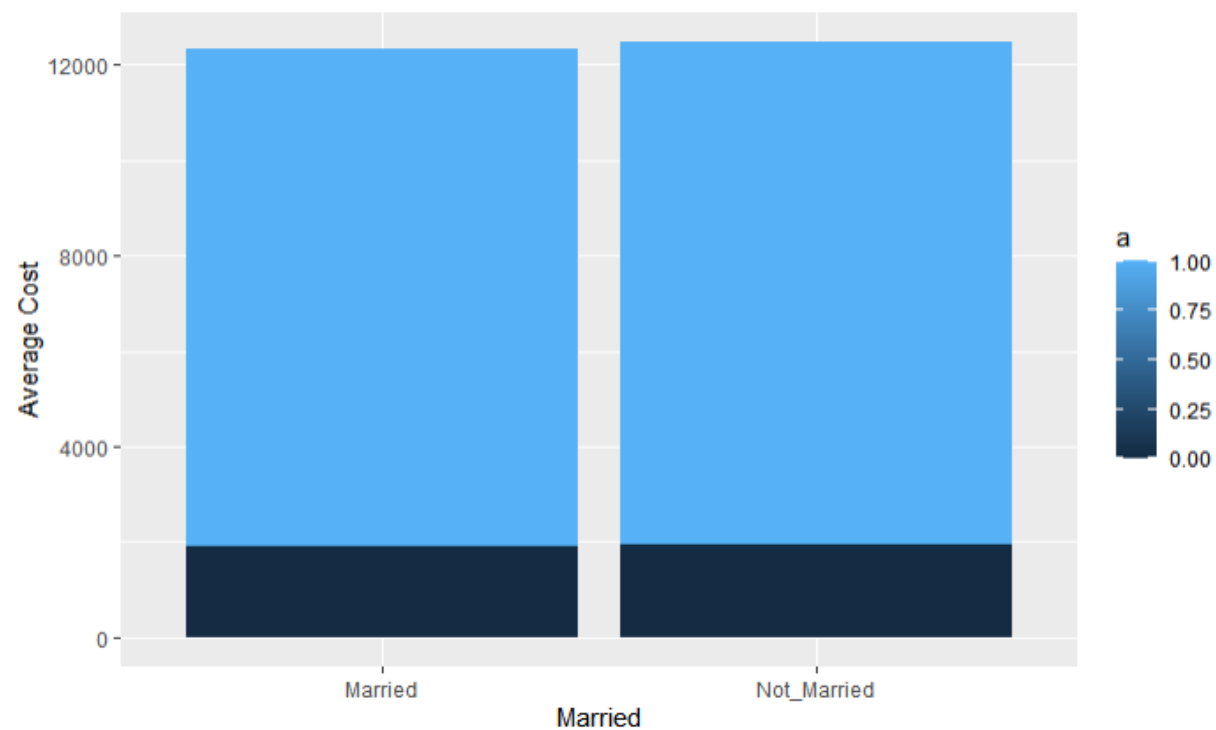
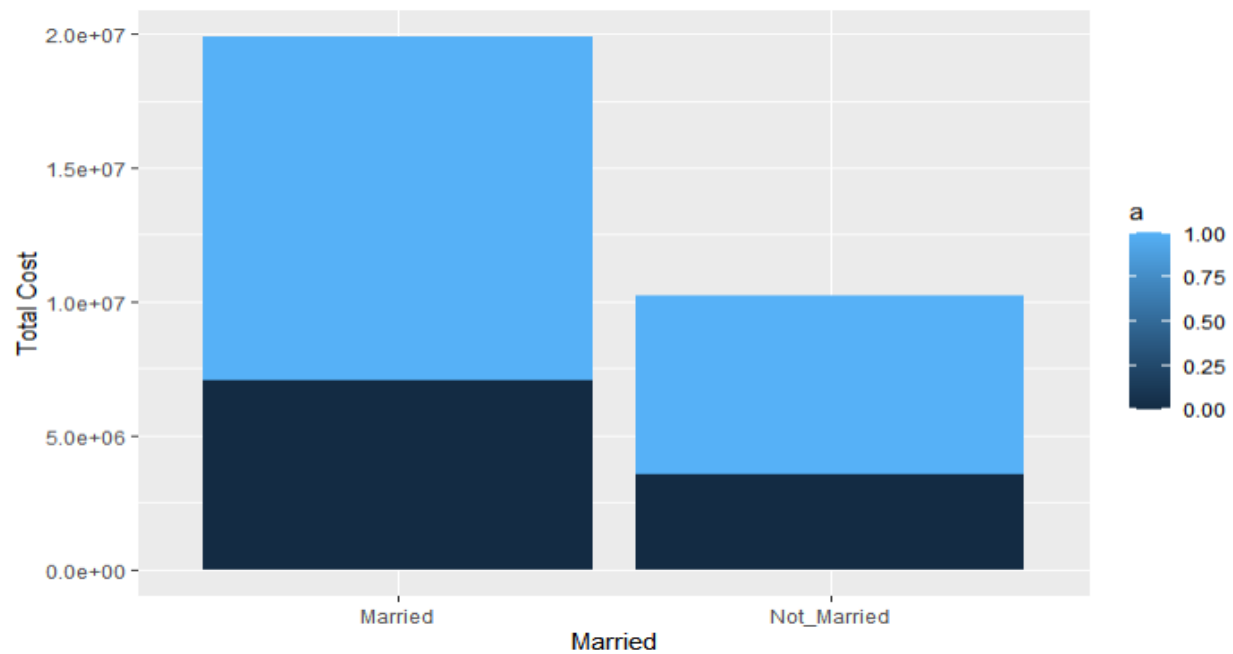
- It is notable that the average claim amount is almost the same irrespective of whether people get yearly physicals.
- However, the people who do not take yearly physicals make up a larger share of the total claim amount.

### Customer Exercises Vs Health Care Costs



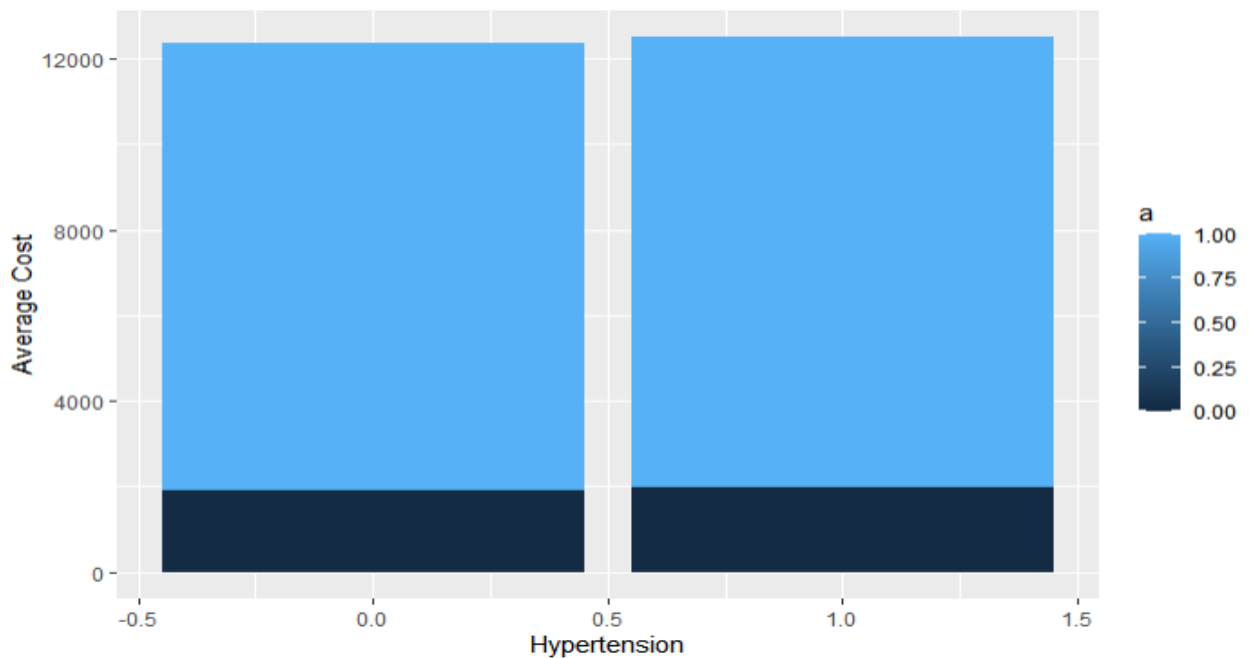
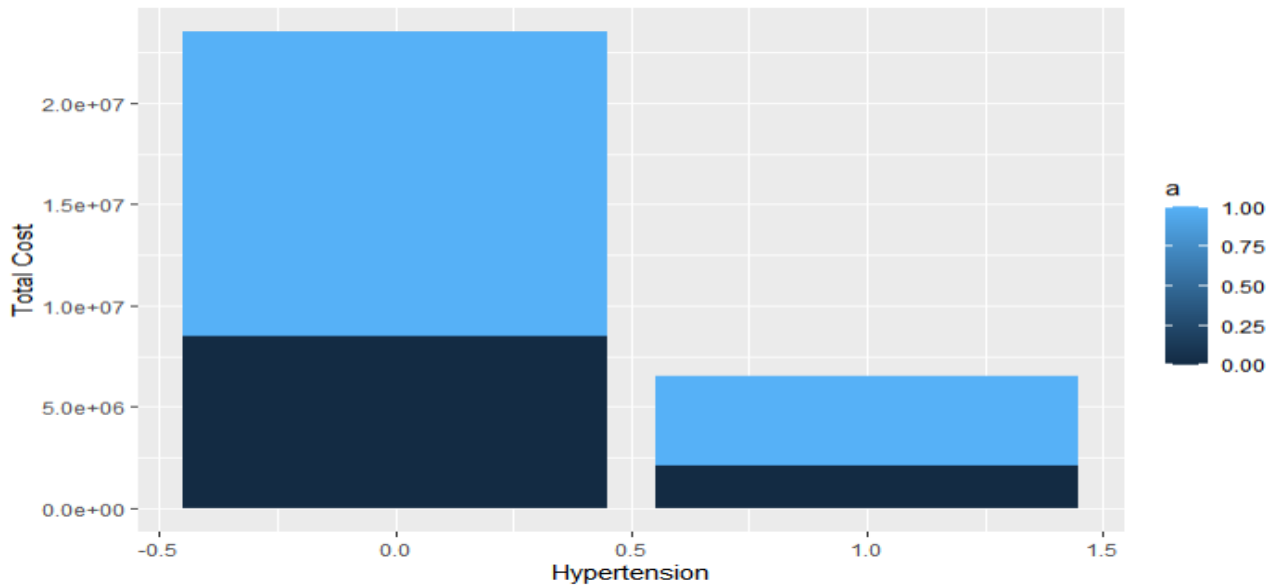
- The average claim amount is almost the same for customers irrespective of whether they are physically active.
- However, the customers who are not physically active make up for a larger share of claims compared to their counterparts

### Marriage VS Health Care Costs



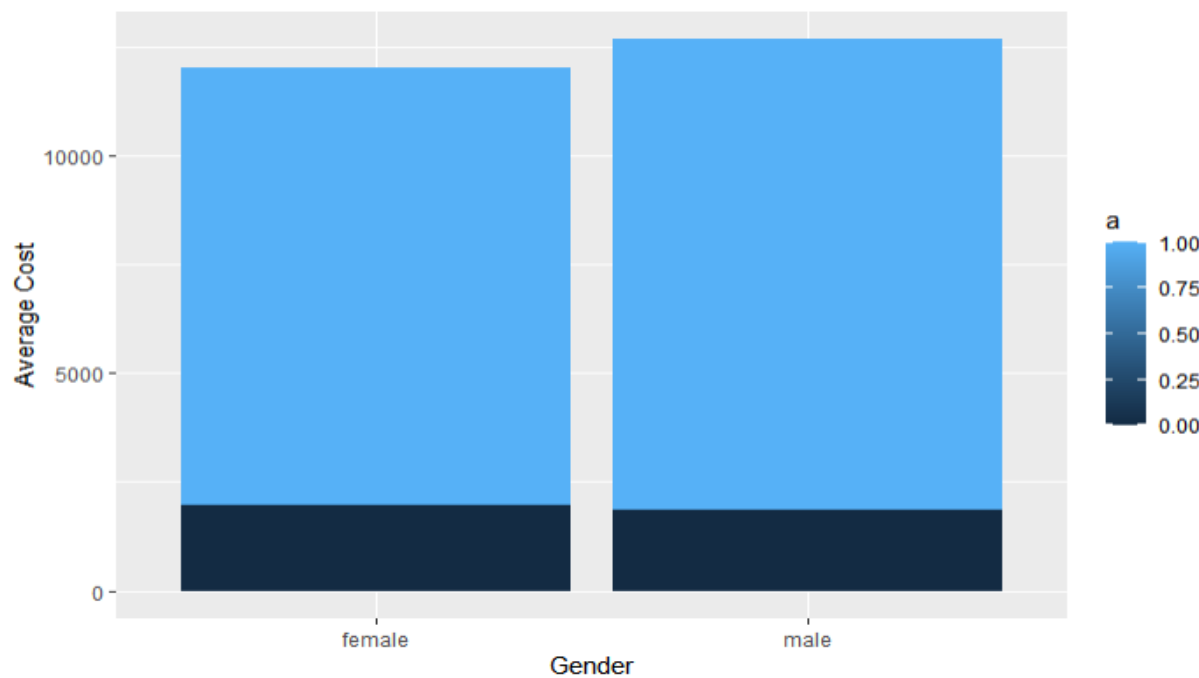
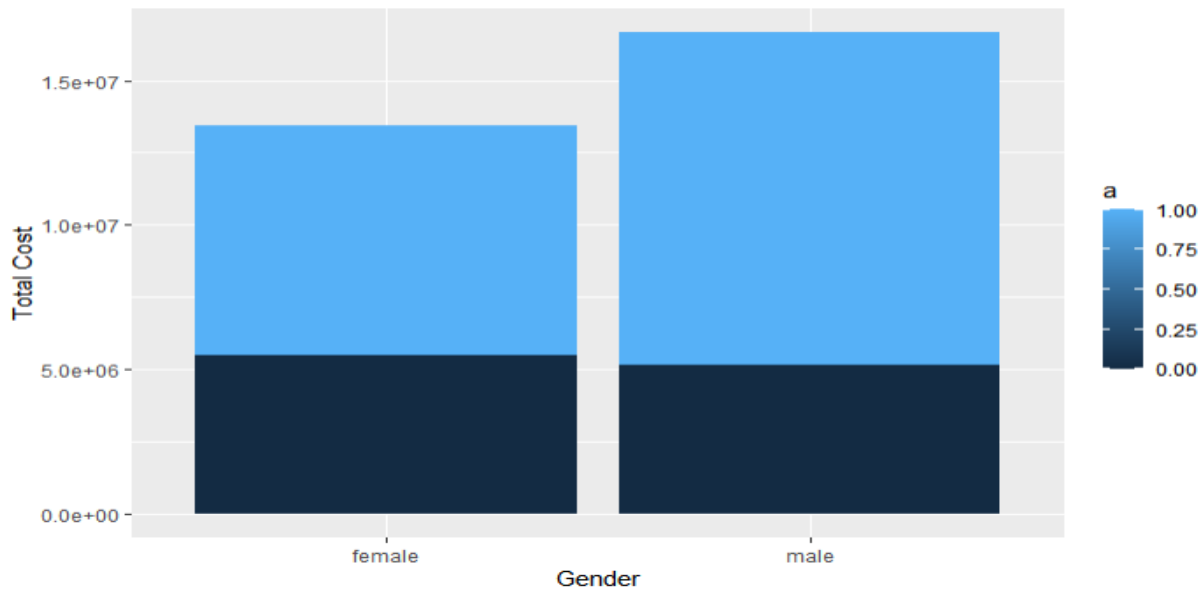
- The total claim amount is much higher for married people than compared to non-married people, with the average claim of non-married people having a slight larger share of 'Expensive' claims

### Hypertension VS HealthCare Cost



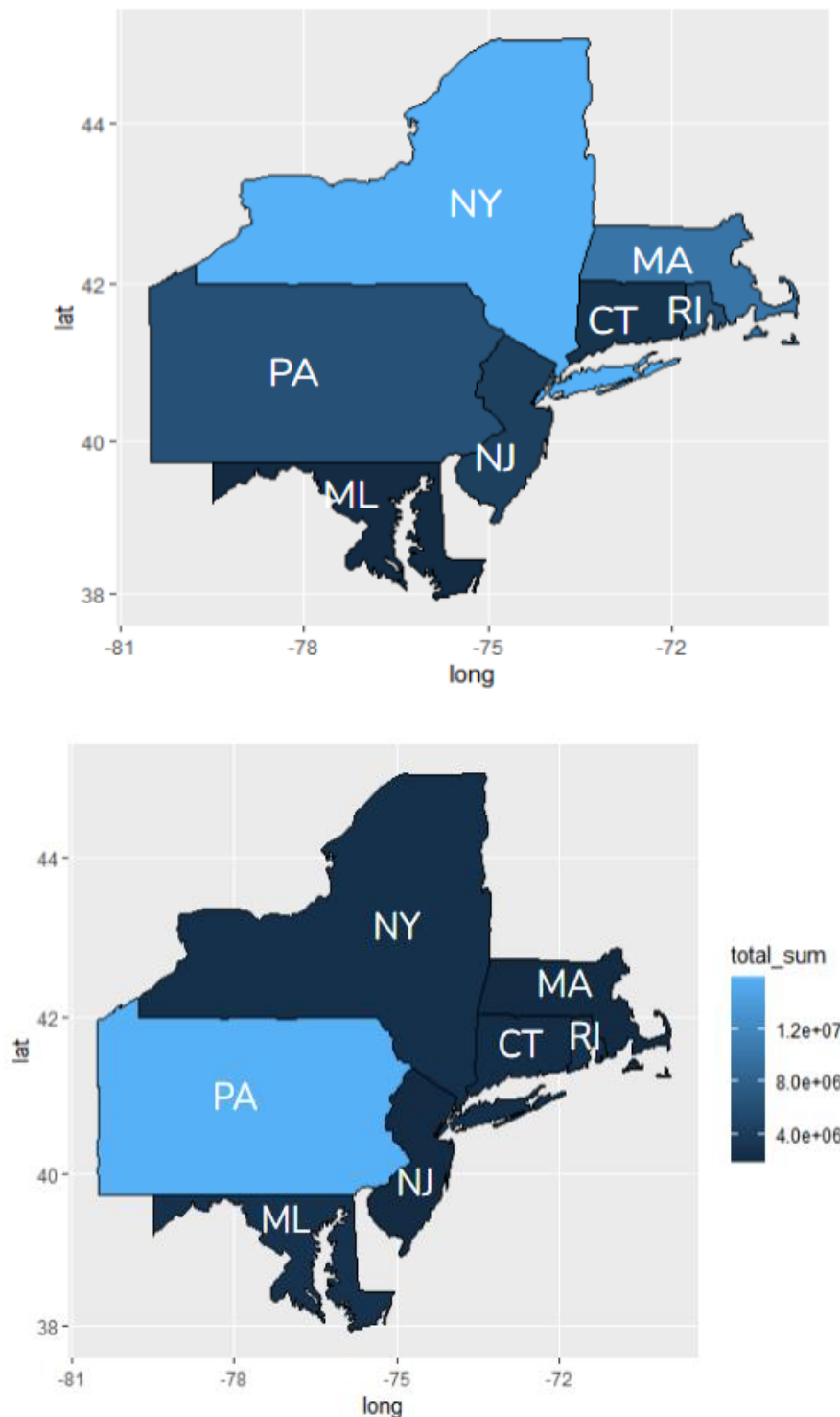
- The average cost is the same for people irrespective of whether they have hypertension or not.
- Notably, the total claim amount is much higher for people who do not have hypertension.

### Gender Vs HealthCare Cost



- Males make up for a larger cumulative claim amount as well as a larger share of 'Expensive' claims.

### Location VS HealthCare Cost





- From the above, we see that New York has the highest average 'Expensive' claims, whereas Pennsylvania has the highest cumulative 'Expensive' claims.

- Model Building

After analyzing the data and guaranteeing that no null values exist. We compiled a list of every variable which was considered to be a valuable input for the model. On our dataset, we used every variable to create a model to the dependent Variable Expensive.

### a) Linear Model Regression

Firstly, we created a linear regression model for the given data by removing each not available value.

```
``{r}
#First Iteration
lm<-lm(Expensive~.,data=health)
summary(lm)

res=resid(lm)
plot(fitted(lm),res)+abline(0,0)

qqnorm(res)+qqline(res)
```

Call:

```
lm(formula = Expensive ~ ., data = health)
```

Residuals:

| Min      | 1Q       | Median   | 3Q      | Max     |
|----------|----------|----------|---------|---------|
| -2.18223 | -0.17019 | -0.05122 | 0.04619 | 0.81907 |

Coefficients:

|                       | Estimate   | Std. Error | t value | Pr(> t ) |     |
|-----------------------|------------|------------|---------|----------|-----|
| (Intercept)           | -1.561e-01 | 2.489e-02  | -6.273  | 3.74e-10 | *** |
| X                     | 4.627e-10  | 6.020e-10  | 0.769   | 0.4422   |     |
| age                   | 1.990e-03  | 2.535e-04  | 7.852   | 4.69e-15 | *** |
| bmi                   | 3.075e-03  | 5.768e-04  | 5.332   | 9.98e-08 | *** |
| children              | -9.469e-04 | 2.697e-03  | -0.351  | 0.7255   |     |
| smoker                | 1.915e-01  | 1.133e-02  | 16.899  | < 2e-16  | *** |
| locationmaryland      | 6.558e-03  | 1.550e-02  | 0.423   | 0.6722   |     |
| locationmassachusetts | -4.495e-03 | 1.750e-02  | -0.257  | 0.7973   |     |
| locationnew jersey    | 1.945e-02  | 1.712e-02  | 1.136   | 0.2560   |     |
| locationnew york      | 1.583e-02  | 1.675e-02  | 0.945   | 0.3449   |     |
| locationpennsylvania  | 4.463e-03  | 1.236e-02  | 0.361   | 0.7180   |     |
| locationrhode island  | -4.084e-03 | 1.571e-02  | -0.260  | 0.7949   |     |
| location_type         | -9.293e-03 | 7.530e-03  | -1.234  | 0.2172   |     |
| education_level       | -8.505e-04 | 4.230e-03  | -0.201  | 0.8407   |     |
| yearly_physical       | 1.426e-02  | 7.551e-03  | 1.889   | 0.0590   | .   |
| exercise              | -5.037e-02 | 7.878e-03  | -6.393  | 1.72e-10 | *** |
| married               | -9.093e-04 | 6.917e-03  | -0.131  | 0.8954   |     |
| hypertension          | 1.653e-02  | 8.140e-03  | 2.030   | 0.0424   | *   |
| gender                | -1.232e-02 | 6.560e-03  | -1.879  | 0.0603   | .   |
| cost                  | 5.252e-05  | 1.010e-06  | 52.029  | < 2e-16  | *** |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

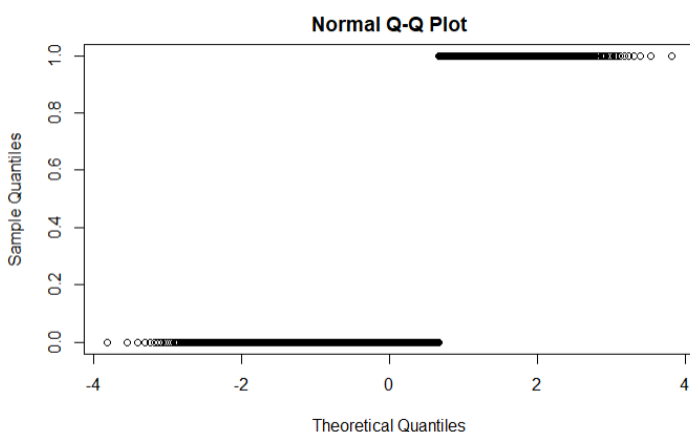
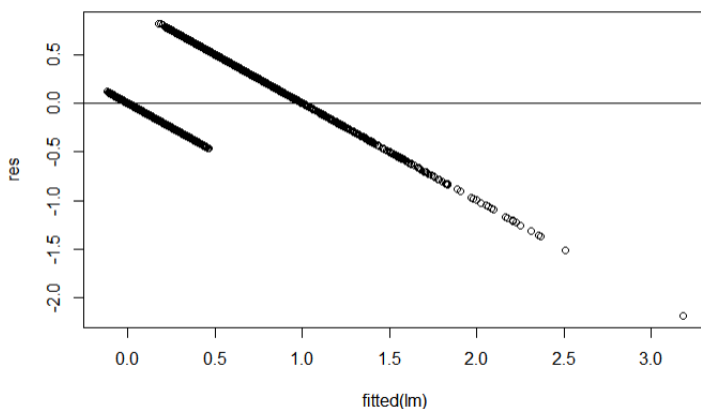
Residual standard error: 0.2805 on 7404 degrees of freedom

Multiple R-squared: 0.5816, Adjusted R-squared: 0.5805

F-statistic: 541.7 on 19 and 7404 DF, p-value: < 2.2e-16

The adjusted R-squared from this model is 0.5805. The adjusted R-squared of 0.5805 signifies how much a change in the dependent variable (Expensive) is affected by a change in the independent variables such as age, BMI, smoker, exercise and cost. The P-value of the model is less than alpha that is 0.05 which means that these independent variables are statistically significant to the dependent variable 'Expensive'.

From the output we have some variables which directly tell us whether the customer paying chances to be in expensive or not. For every unit of age increase there is 0.000199 chances that the customer will be Expensive or paying high value. For every unit of BMI increase there is 0.0003075 chances that customer will be in Expensive Group or paying high. For every unit increase in smoker there is a chance that there are 0.0195 chances that the customer is in Expensive Group. For every Unit increase in exercise the category of being in Expensive will decrease by 0.00503.



- Negative slope residual graph due to frequently obtained lower bound (Zero) values.
- Quantile-Quantile graph due to Binary classified values.

| p-value | R-Squared |
|---------|-----------|
| 2.2e-16 | 58.05%    |

| Variable    | Age      | BMI      | Smoker  | Exercise  |
|-------------|----------|----------|---------|-----------|
| Coefficient | 5.83e-15 | 9.58e-08 | < 2e-16 | -1.83e-10 |

## b) SVM Model

For the SVM Model we are dividing the dataset into two parts: train and test data set. In the train data set we have 65% of the dataset whereas in the test data set we have 35% of the dataset.

```

```{r}
library(kernlab)
library(caret)
health$cost=as.factor(health$cost)
trainList <- createDataPartition(y=health$cost,p=.65,list=FALSE)
trainset <- health[trainList,]
testset <- health[-trainList,]
```

```

```

 REFERENCE
Prediction 0 1
 0 1903 287
 1 45 362

```

```

 Accuracy : 0.8722
 95% CI : (0.8587, 0.8848)
 No Information Rate : 0.7501
 P-Value [Acc > NIR] : < 2.2e-16

```

```

 Kappa : 0.6106

```

```

 McNemar's Test P-Value : < 2.2e-16

```

```

 Sensitivity : 0.9769
 Specificity : 0.5578
 Pos Pred Value : 0.8689
 Neg Pred Value : 0.8894
 Prevalence : 0.7501
 Detection Rate : 0.7328
 Detection Prevalence : 0.8433
 Balanced Accuracy : 0.7673

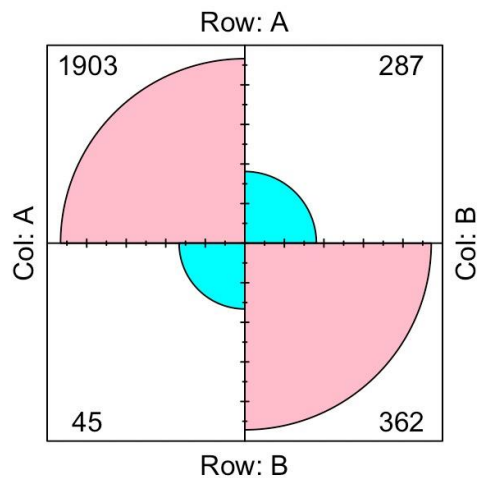
```

```

 'Positive' Class : 0

```

## Confusion Matrix



A p-value of  $<2.2e-16$  for  $\text{Accuracy} > \text{NIR}$ , the null hypothesis is rejected, therefore the accuracy is statistically significant.

## c) Tree Model

To achieve as much homogeneity as is practical in each final subspace, decision tree models continually subdivide the data into various subspaces. Recursive partitioning is the formal name for this method.

```
#Recursive Partitioning and Regression Trees
library(e1071)
library(caret)
library(rpart)
library(ggplot2)
library(rpart.plot)
rpartmodel<-train(cost~age+bmi+smoker+exercise+gender,data=trainset,method="rpart")
rpart.plot(rpartmodel$finalModel)
rpartPred<-predict(rpartmodel,testset)
confM2<-confusionMatrix(rpartPred,testset$cost)
confM2
```
```

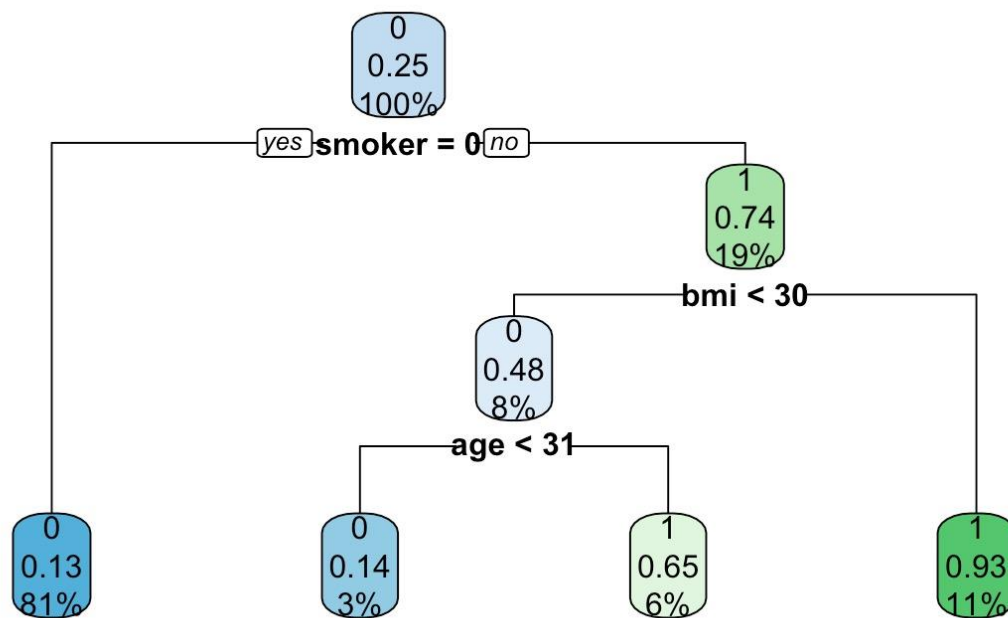
Accuracy : 0.8685
95% CI : (0.8523, 0.8836)
No Information Rate : 0.75
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.6052

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9684
Specificity : 0.5690
Pos Pred Value : 0.8708
Neg Pred Value : 0.8571
Prevalence : 0.7500
Detection Rate : 0.7263
Detection Prevalence : 0.8341
Balanced Accuracy : 0.7687

'Positive' Class : 0



Model Confusion

For the Dataset Problem we created 3 models to analyze the best significant predictors for the dependent variable. From Linear Regression model there are 5 variables which are significant to the dependent variable. Those are age, BMI, smoker, exercise, and hypertension. The accuracy for the linear regression is 58% which is low. So, we go for the SVM model and create a model using all the variables. From that we got an accuracy of 87.22%. We also created a Tree model by only using significant predictors and we got an accuracy of 86.85%.

From the data, we ran multiple models to determine which model was best to determine which clients would be expensive next year. Out of the above, we found SVM to have the highest accuracy of 87.22%, so we conclude that the SVM model is the best out of the three models to predict the same, which would be used for shiny app deployment.

Numbers And Insights

- People lying in the range of age: 20-30 and BMI: 20-30 have very low claims and relatively lower average claims; it is recommended to target more people in that demographic
- Customers who have more than 4 children as they have lower average claim with very low claims
- Target nonsmokers as they have lower average claims with high ratio of non-expensive claims
- Customers from Country rather than urban as they have very low claims
- Customers who either had a PhD or have no college degree as they have minimal claims
- People who exercise frequently and have yearly physicals as they have very low claims with low average claim
- People who are not married as they have very low claims.
- Patients who have hypertension as they have very low claims
- People outside the state of Pennsylvania as they have very high claims despite having a high ratio of non-expensive average claims

Analysis and Recommendations

Providing more affordable healthcare draws in more customers, resulting in higher revenue. With the healthcare cost information provided by HMO, our team went over each factor and created a model to select factors that we can use to lower the healthcare cost.

Specifically, in this report, we focused on three objectives. First, we aimed to find out the conditions that influenced the cost of healthcare. Second, we predicted customers who have higher healthcare costs by looking at various conditions. Lastly, we recommend possible solutions to HMO to lower their total healthcare costs.

With these objectives in mind, we had six major business questions. First, can we predict a pattern based on costs? This question allowed us to find out which factors affect the cost. Second, what threshold number determines whether a customer is paying the high or low cost? A threshold number had to be determined to find out the conditions that made the healthcare cost high. In this case, the threshold number was selected after carefully examining the distribution of the costs. Third, which age group pays more than others? Age is often the major factor in raising healthcare costs. We tried to go over each age group to compare each age group. Fourth, Can BMI impact the cost? BMI affects health conditions which in turn would be reflected in healthcare costs. We sought to look at how BMI affects costs in ways such as generating BMI and the average cost and BMI and the total cost. Five, Does the place of residence affect the cost? In the healthcare cost information provided, the location of the place of residence (state) as well as the type of location (urban vs. country). Using this information, we were able to evaluate whether the place of residence has an impact on the cost. Six, does the lifestyle of a customer affect the cost? To answer this question, we factored in Lifestyle conditions such as being married or having children. Last, are health related to habits related to the cost? Here, we looked at factors that include exercising, smoking, or getting yearly physical.

After going through the necessary data cleaning, we conducted descriptive analyses to get a better understanding of the data. Next, we used three different models to examine factors that affect the cost and most accurately predict the cost. We used a linear regression model including all the factors(predictors), which showed that age, BMI, smoking habit, and exercise are significantly related to cost. The model

successfully explained 58.05% of the data (total variance). Next, by conducting SVM within the provided dataset, we were able to create a model with 85.81% accuracy. Specifically, we were able to accurately predict who would fall under the expensive group (paying the higher cost) by 96.74% while predicting who would fall under the rest by 53.22%. Lastly, using a partition tree within the provided dataset, we were able to create a model with 86.26% accuracy. Specifically, we were able to accurately predict who would fall under the expensive group (paying the higher cost) by 96.41% while predicting who would fall under the rest by 55.82%.

From these analyses, we provide the following actionable insight, which focuses on targeting demographics. First, customers between the age of 20 and 30 have low costs as well as a relatively low average cost. Also, customers between BMI 20 and 30 also share the same trait. Thus, we recommend targeting more people between 20-30 or people whose BMI ranges between 20-30. Second, in terms of lifestyle, customers who had more than 4 children had a low average cost with low total cost. In addition, customers who were not married had a lower cost. Here, we recommend targeting those who are not married or who have more than four children. Third, regarding health-related habits, we were able to identify several patterns concerning costs. Non-smokers have lower average costs with a high ratio of non-expensive claims. This means that non-smokers pay less and are even in the lower-cost group of the given data. Moreover, customers who exercise frequently and have yearly physicals have low total cost and average costs. Notably, customers who have hypertension have low costs. Thus, we recommend targeting people who don't smoke, exercise frequently, get yearly physicals and have hypertension. Fourth, regarding the place of residence, customers from the country rather than the urban area had low costs. Especially customers outside the state of Pennsylvania had higher costs despite having a high ratio of non-expensive average claims. Thus, we recommend recruiting more people from the Pennsylvania area as well as those who reside in the countryside. Lastly, customers whose highest educational degrees were either Ph.D. or no college degree had lower claims. Thus, targeting those specific educational levels would be helpful.

Shiny Web Apps

HMO input file

Browse... HMO_TEST_data_sample(1).c
Upload complete

HMO solution file

Browse... HMO_TEST_data_sample_sol
Upload complete

Number of Rows

5

X	age	bmi	children	smoker	location	location_type	education_level	yearly_physical	exercise	married	hypertension	gender
8.00	37.00	27.74	3.00	no	NEW JERSEY	Urban	Bachelor	Yes	Not-Active	Not_Married	0.00	female
10.00	60.00	25.84	0.00	no	PENNSYLVANIA	Urban	Bachelor	No	Not-Active	Married	0.00	female
20.00	30.00	35.30	0.00	yes	NEW YORK	Country	PhD	No	Not-Active	Married	0.00	male
24.00	34.00	31.92	1.00	yes	PENNSYLVANIA	Urban	Bachelor	No	Not-Active	Married	0.00	female
30.00	31.00	36.30	2.00	yes	PENNSYLVANIA	Urban	Master	Yes	Not-Active	Not_Married	0.00	male

This is the first interface of our shiny app. The first box is an input for the file that we need predictions for. The second box is for the solution file to check the accuracy of our model. There is a 'Number of Rows' option wherein we can put a number of our choice and the app displays that many number of the initial rows of our sample data. After uploading both the files, the model makes a prediction, and we get a confusion matrix as the output.

Confusion Matrix and Statistics

```
Reference
Prediction 0 1
0 9 5
1 3 3

Accuracy : 0.6
95% CI : (0.3605, 0.8088)
No Information Rate : 0.6
P-Value [Acc > NIR] : 0.5956

Kappa : 0.1304

McNemar's Test P-Value : 0.7237

Sensitivity : 0.7500
Specificity : 0.3750
Pos Pred Value : 0.6429
Neg Pred Value : 0.5000
Prevalence : 0.6000
Detection Rate : 0.4500
Detection Prevalence : 0.7000
Balanced Accuracy : 0.5625

'Positive' Class : 0
```

This is the confusion matrix that we got for our sample data. It shows an accuracy of 60% with the 95% confidence interval ranging from 0.3605 to 0.8088. We get a sensitivity of 75%.