The dataset contains healthcare cost information from an HMO (Health Management Organization). Each row in the dataset represents a person. Health Management Organizations (HMOs) are medical insurance groups that offer health care in exchange for a set annual charge. The dataset that we were given has 14 columns and contains data on 7,583 people. The columns broadly focus on several categories, including the individual's unique identifier, age, geographic location, gender, education level, marital status, number of children, and healthcare expenditure. They also ask about the individual's exercise, smoking, BMI, annual physical examination status, and hypertension status. Based on the facts at hand, we deliver actionable information through our research, and we also successfully anticipate which consumers will spend a lot of money on healthcare.

— Project Goal

• Predict people who will spend a lot of money on health care next year (i.e., which people will have high healthcare costs).

• Provide actionable insight to the HMO, in terms of how to lower their total health care costs, by providing a specific recommendation on how to lower health care costs. — Part 1: Data Acquisition (Loading the data)

```
#loading the readr package to import the data file
library(readr)
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.1     ✓ purrr     1.0.1
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.2     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ── 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

datafile <- "https://intro-datascience.s3.us-east-
2.amazonaws.com/HMO_data.csv"

#using the read_csv function to read the CSV file into a data frame called
hmo_data
hmo_data <- read_csv(datafile)

## 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.

#checking the number of rows and columns in the dataset
dim(hmo_data)

## [1] 7582   14

#the dataset has 7582 rows and 14 columns
```

Part 2: Data Exploration (Finding out the data attributes)

```
#structure of the data frame
str(hmo_data)

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

#first six rows of the dataframe
head(hmo_data)

## # A tibble: 6 × 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 ¹location_type,
## #   ²education_level, ³yearly_physical, ⁴exercise

#last six rows of the dataframe
tail(hmo_data)

## # A tibble: 6 × 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 21222    39  30.9        4 no     PENNSYLVANIA Urban   Bachel… Yes
Not-Ac…
## 2 13023    63  30.9        3 yes    NEW JERSEY   Urban   No Col… No
Not-Ac…
## 3 54813    53  46.7        2 no     PENNSYLVANIA Urban   Bachel… Yes
Not-Ac…
## 4 64221    42  28.3        3 yes    PENNSYLVANIA Urban   Bachel… No
Active
## 5 74732    33  27          2 no     PENNSYLVANIA Country Bachel… No
Not-Ac…
## 6 13531    20  28.8        0 no     NEW YORK     Urban   Bachel… No
Active
```

```
## # … with 4 more variables: married <chr>, hypertension <dbl>, gender
<chr>,
## #   cost <dbl>, and abbreviated variable names ¹location_type,
## #   ²education_level, ³yearly_physical, ⁴exercise

#summary statistics of the dataset and cost column
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
##

summary(hmo_data$cost)

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

#checking for NA's/missing values in the dataset
any(is.na(hmo_data))

## [1] TRUE
```

```r
#checking for duplicated rows
hmo_data_d  <- duplicated(hmo_data)
#hmo_data[hmo_data_d,] #0

#calculating the total cost of the dataset
sum(hmo_data$cost)
```

```
## [1] 30653732
```

```r
#checking the total number and percentage of missing values in each column
total <- colSums(is.na(hmo_data))
percent <- total / nrow(hmo_data) * 100
result <- data.frame(Total = total, Percent = percent)
result
```

```
##                  Total  Percent
## X                    0 0.000000
## age                  0 0.000000
## bmi                 78 1.028752
## children             0 0.000000
## smoker               0 0.000000
## location             0 0.000000
## location_type        0 0.000000
## education_level      0 0.000000
## yearly_physical      0 0.000000
## exercise             0 0.000000
## married              0 0.000000
## hypertension        80 1.055131
## gender               0 0.000000
## cost                 0 0.000000
```

```r
#The 'bmi' and 'hypertension' columns contain missing values.
#bmi has 78 null values
#hypertension has 80 null values

#Inital Analysis:

#1. The dataset contains 7,582 rows and 14 columns, where the 'X' column
represents
    #the unique identifier.

#2. The 'smoker', 'yearly_physical', 'exercise', 'married', and 'gender'
columns
    #have binary values.

#3. The 'bmi' and 'hypertension' columns contain missing values.

#4. The 'location' and 'location_type' columns indicate the location and type
of
    #location, respectively.
```

```
#5. The 'education_level' column indicates the highest education level of the
    #individual.

#6. The 'cost' column represents the medical cost of the individual.
```

Part 3: Data Cleaning (Using na_interpolation)

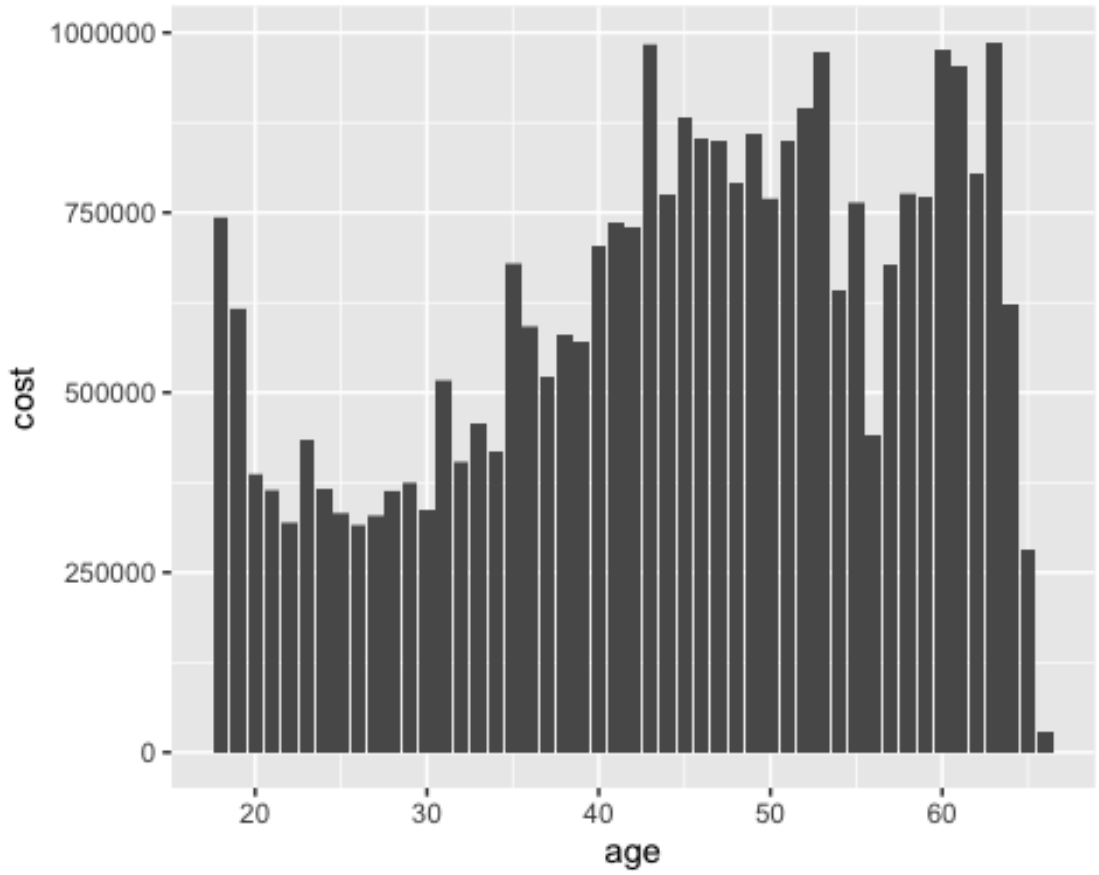
```
#loading the imputeTS function to
library(imputeTS)

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

#imputing missing values in 'bmi' column
hmo_data$bmi <- na_interpolation(hmo_data$bmi)

#removing rows with missing values in 'hypertension' column
hmo_data <- hmo_data[!is.na(hmo_data$hypertension),]

#Checking for null values after cleaning
sum(is.na(hmo_data$bmi)) #0

## [1] 0

sum(is.na(hmo_data$hypertension)) #0

## [1] 0

sum(is.na(hmo_data)) #0

## [1] 0

#the cleaned dataset has 7502 rows and 14 columns
```

Part 4: Dividing Dataset Into Expensive and Not Expensive People

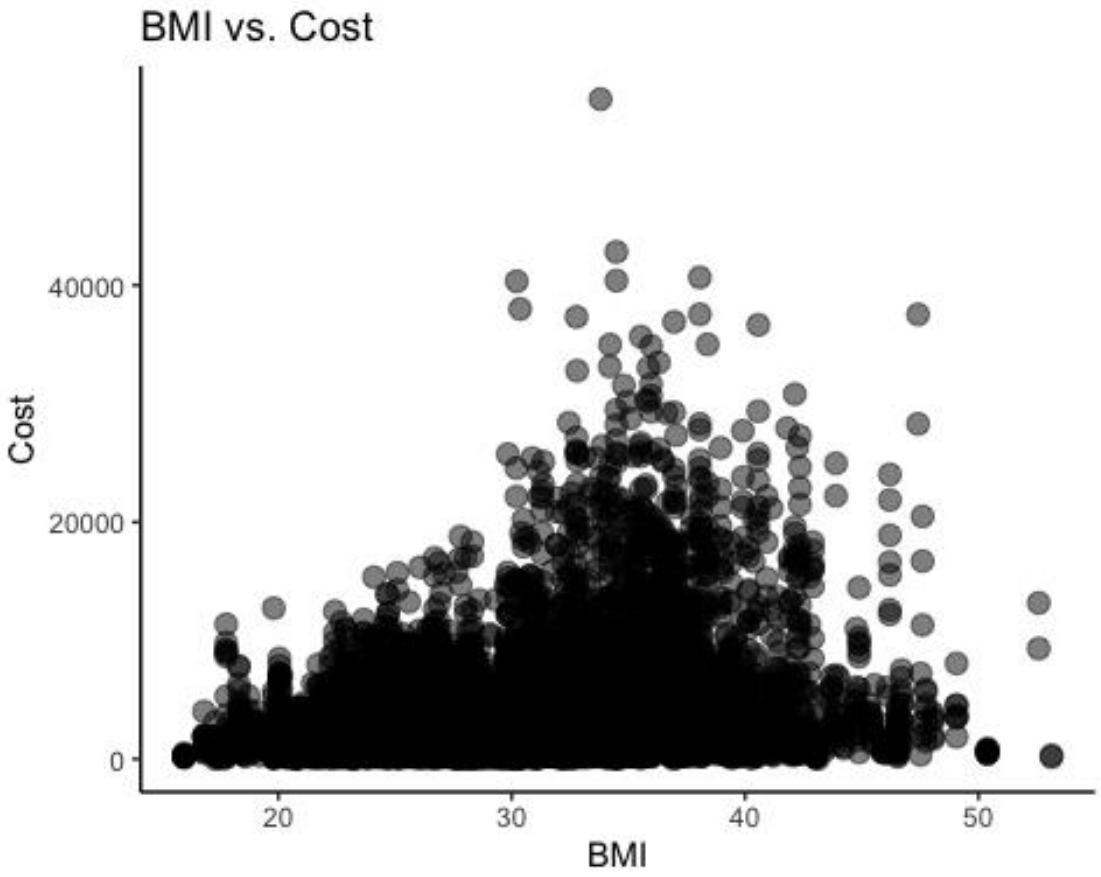
```
#Since the Mean of Cost Column is $4043, we define people who are paying more
than $4200 as expensive
hmo_data$expensive <- ""
for (i in 1:7502){
  if(hmo_data[i,"cost"] > 4200)
    hmo_data[i,"expensive"] <- "yes"
  else
    hmo_data[i,"expensive"] <- "no"
}
hmo_data$expensive <- as.factor(hmo_data$expensive)

#Expensive attribute yes means the customer is expensive and no means it's
not expensive.
```

Part 5: Data Visualization

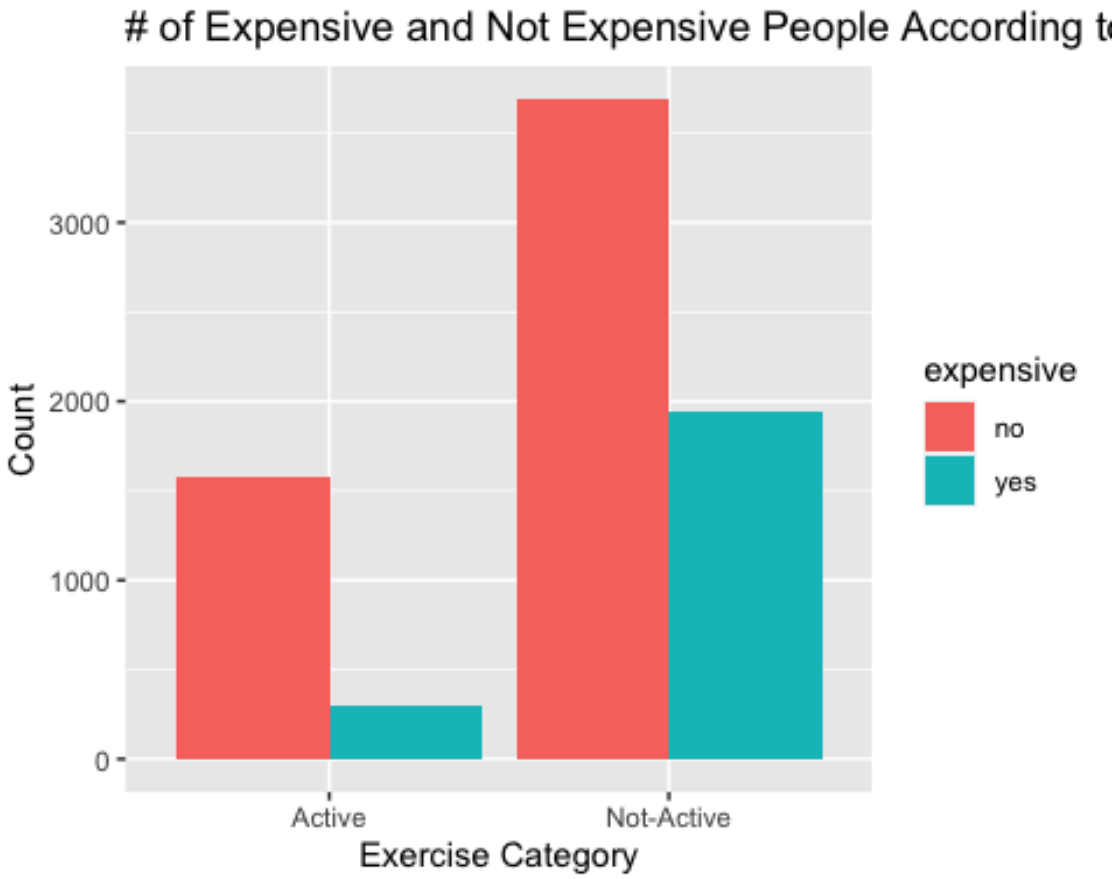
```
#Age vs Cost Bar Plot
library(ggplot2)
ggplot(hmo_data, aes(x=age, y=cost)) +
  geom_bar(stat="identity")
```



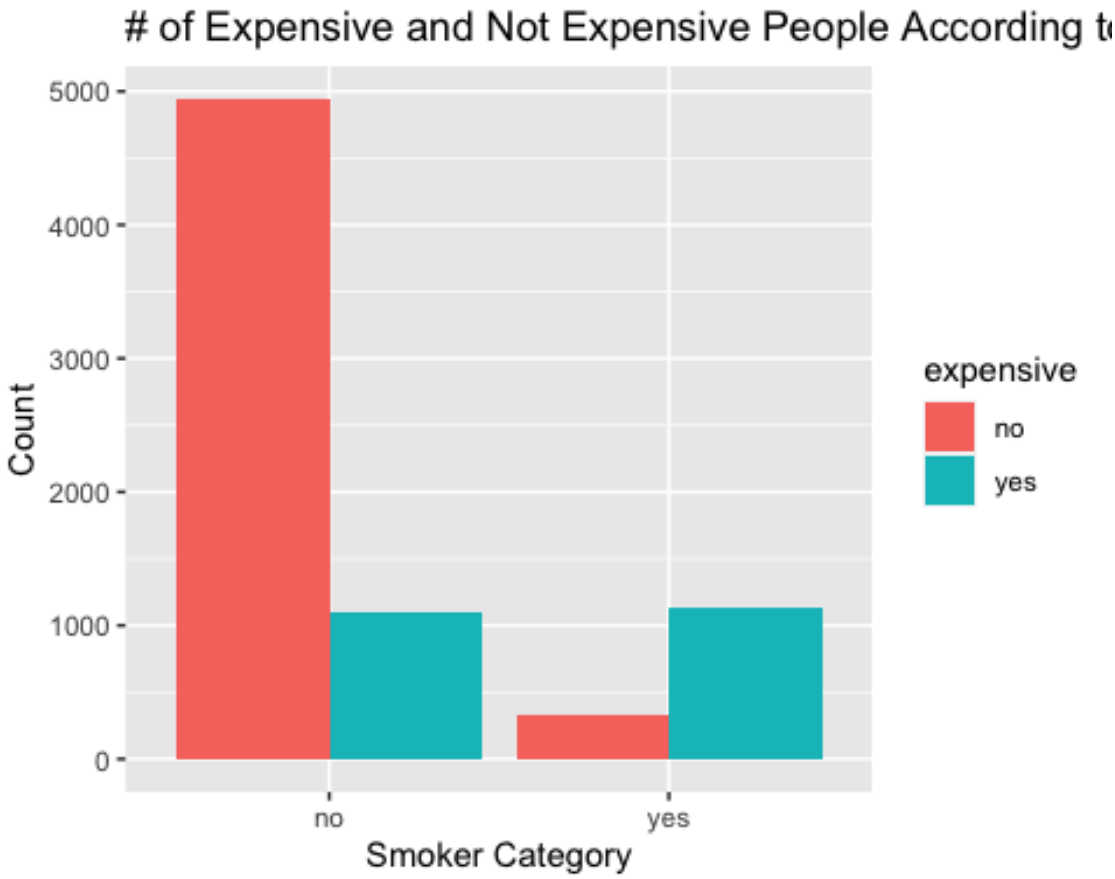

```
#BMI vs Cost Scatter Plot
library(ggplot2)

ggplot(data = hmo_data, aes(x = bmi, y = cost)) +
  geom_point(alpha = 0.5, size = 3) +
  labs(x = "BMI", y = "Cost", title = "BMI vs. Cost") +
  theme_classic()
```



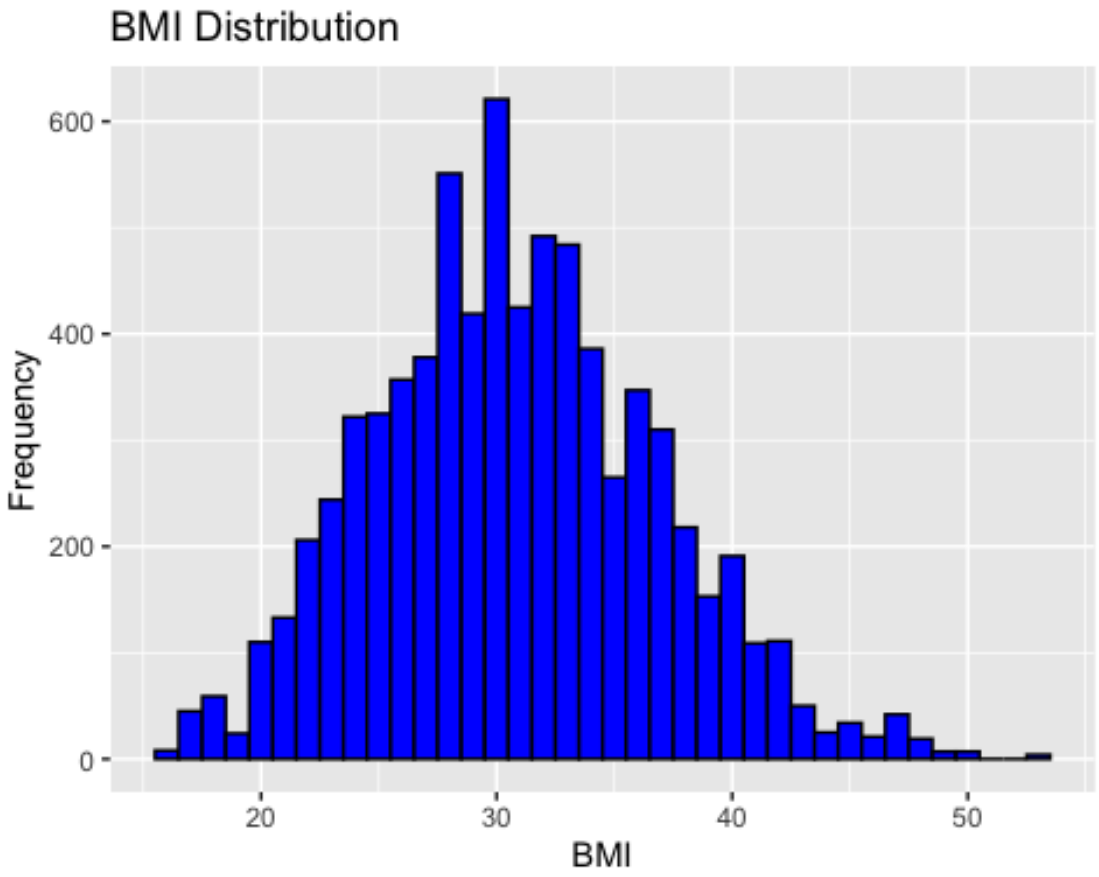

```
#Exercise Level vs Expensive Status
library(ggplot2)
ggplot(hmo_data, aes(x=exercise, fill=expensive)) +
  geom_bar(position="dodge") +
  xlab("Exercise Category") +
  ylab("Count") +
  ggtitle("# of Expensive and Not Expensive People According to Exercise
Level")
```



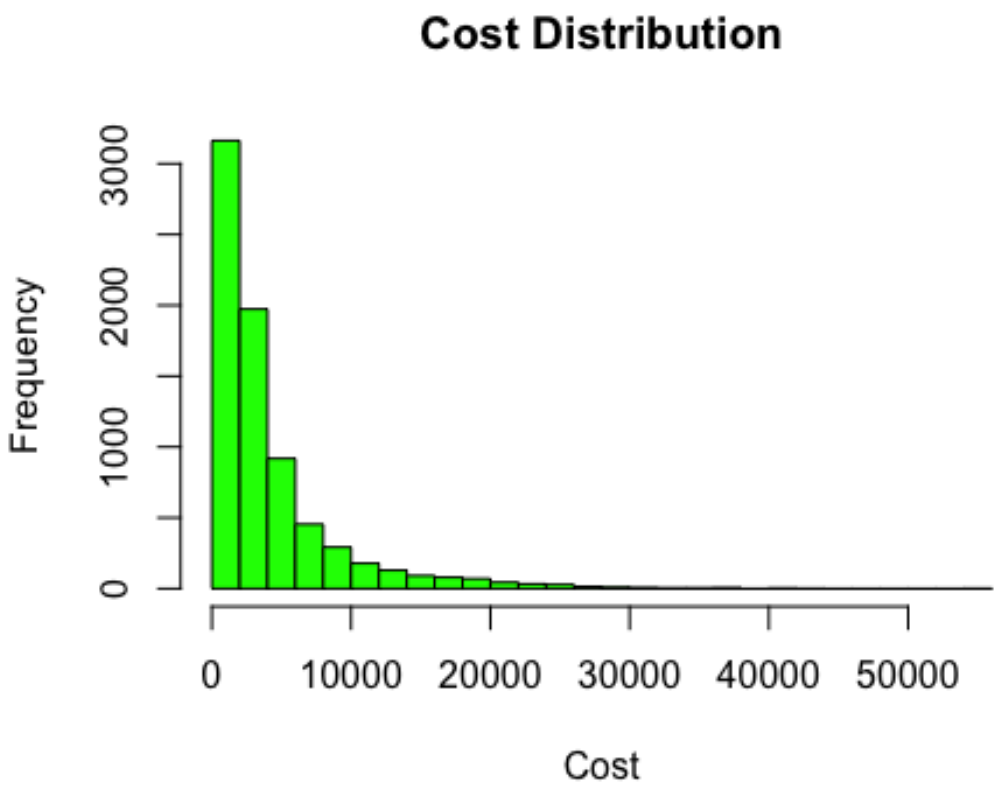

```r
#Smoking Status vs Expensive Status
library(ggplot2)
ggplot(hmo_data, aes(x=smoker, fill=expensive)) +
  geom_bar(position="dodge") +
  xlab("Smoker Category") +
  ylab("Count") +
  ggtitle("# of Expensive and Not Expensive People According to Smoker
Status")
```



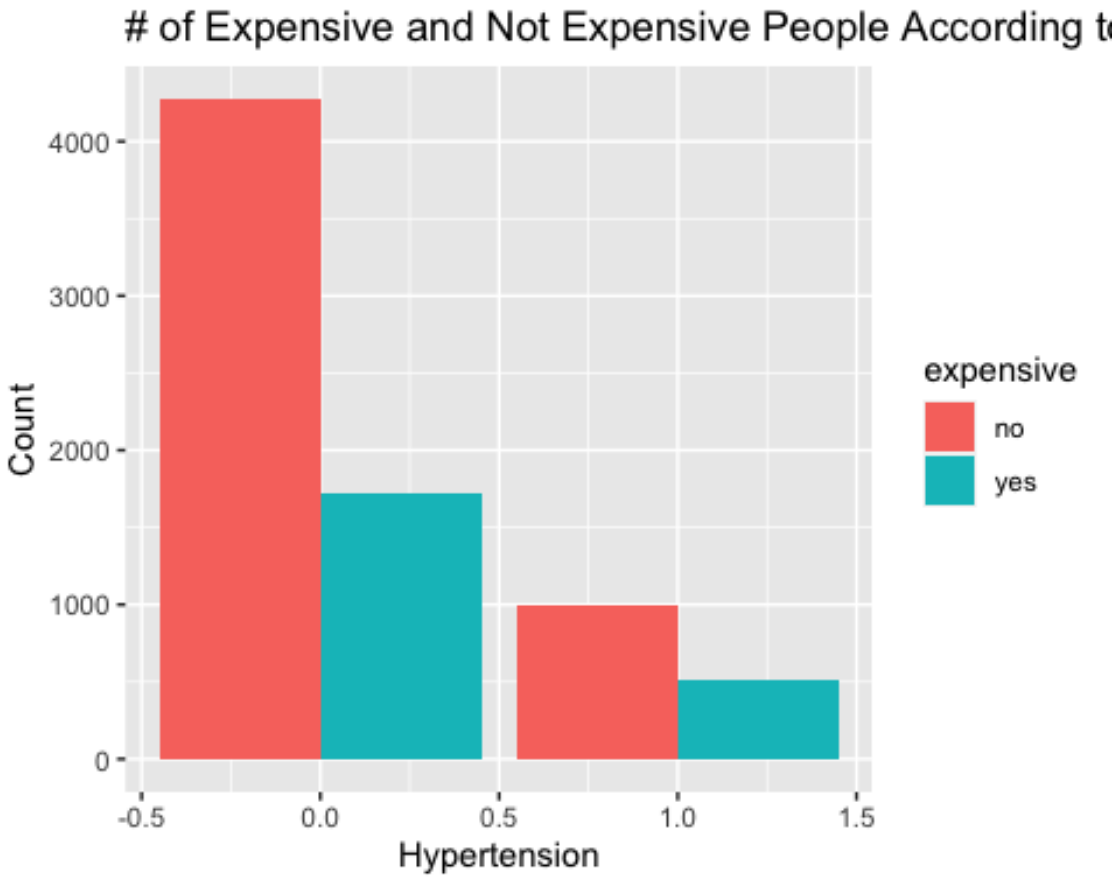

```
#Histogram of BMI
library(ggplot2)
ggplot(hmo_data, aes(x=bmi)) +
  geom_histogram(binwidth=1, color="black", fill="blue") +
  labs(title="BMI Distribution", x="BMI", y="Frequency")
```



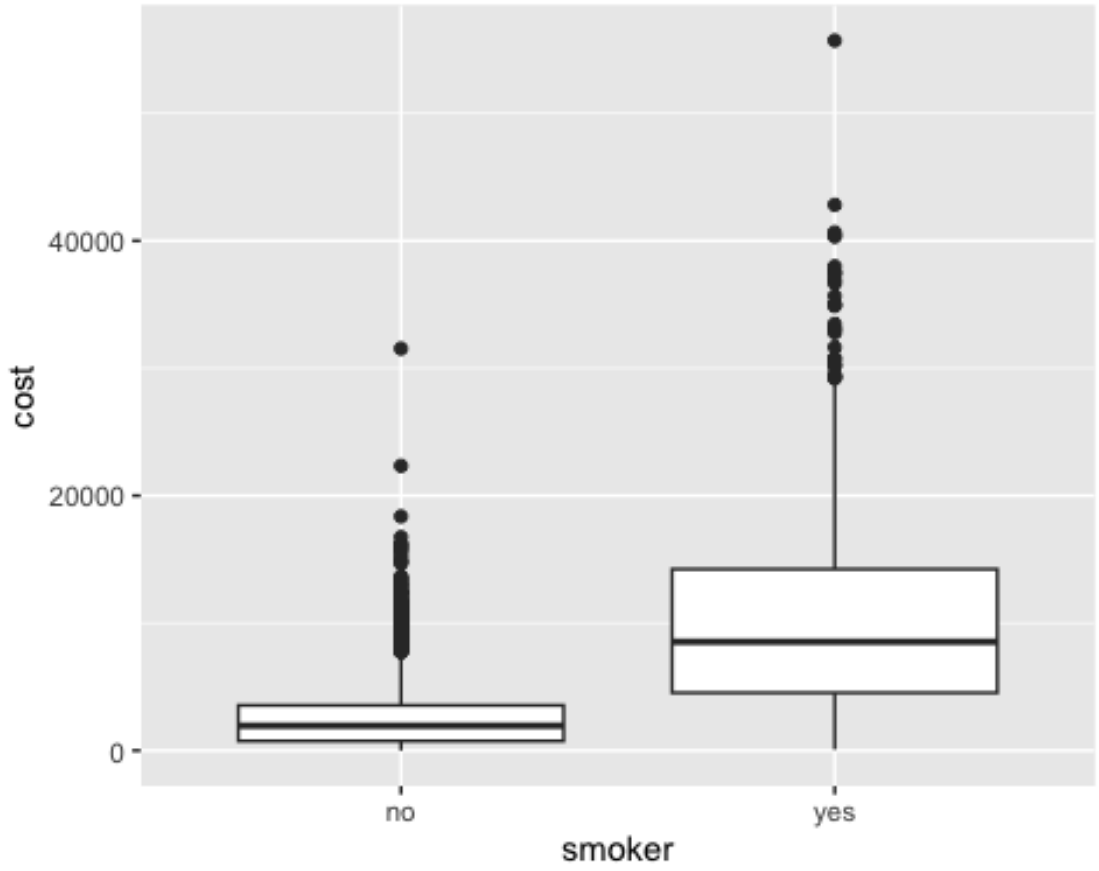

```r
#Cost distribution
hist(hmo_data$cost, breaks = 30, col = "green", main = "Cost Distribution",
xlab = "Cost", ylab = "Frequency")
```



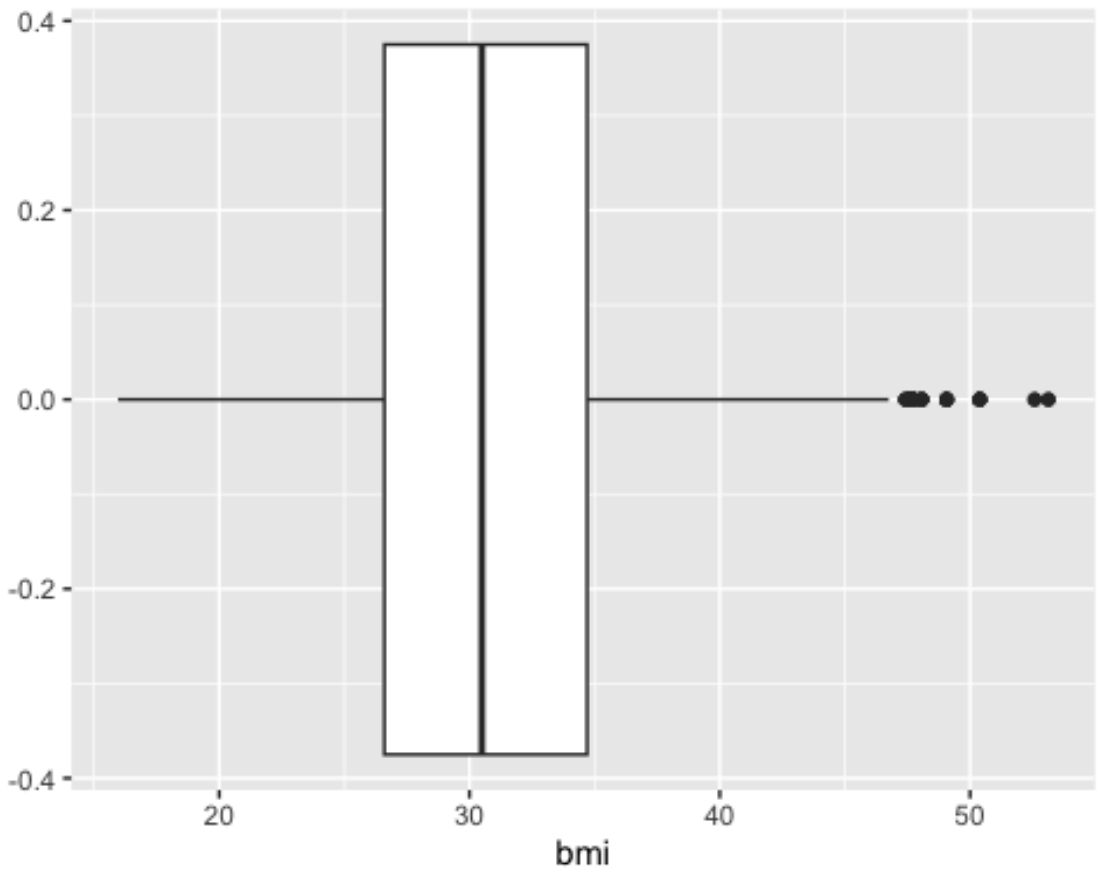

```
#Hypertension vs Expensive Status
library(ggplot2)
ggplot(hmo_data, aes(x=hypertension, fill=expensive)) +
  geom_bar(position="dodge") +
  xlab("Hypertension") +
  ylab("Count") +
  ggtitle("# of Expensive and Not Expensive People According to
Hypertension")
```



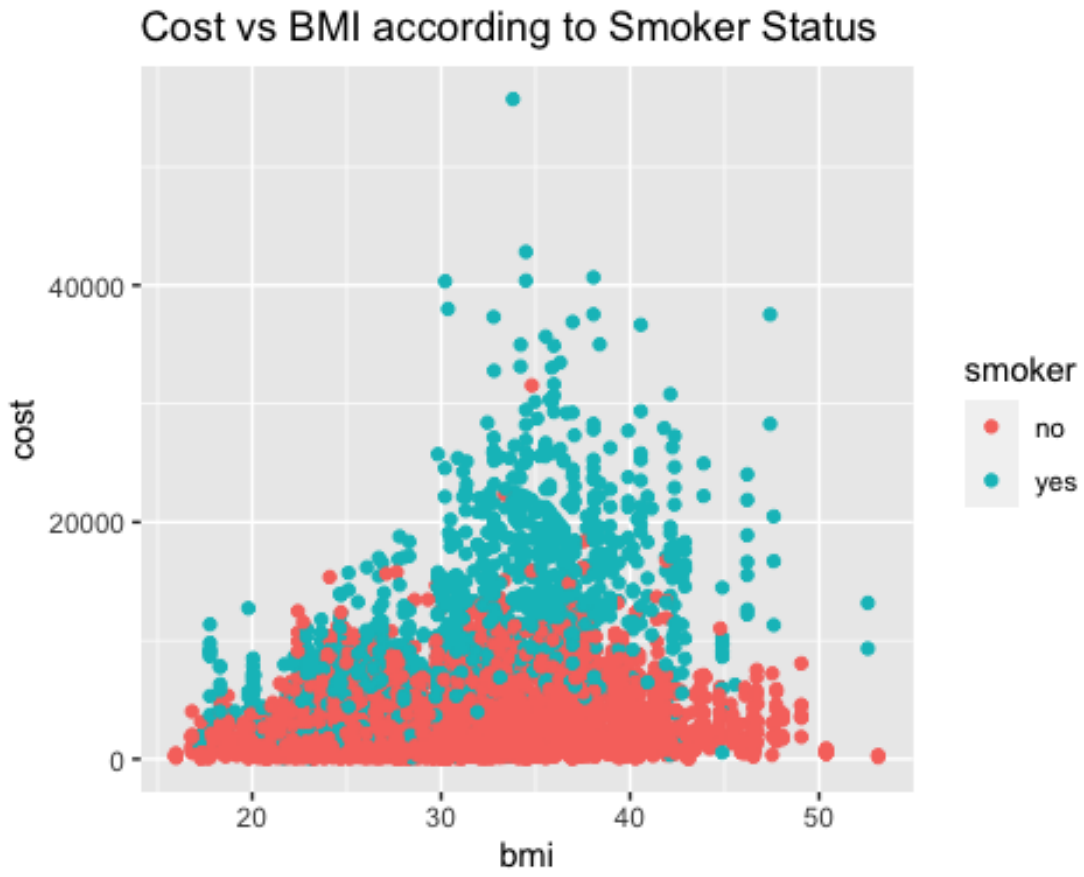

```
#Boxplot of Smoker
ggplot(hmo_data, aes(x = smoker, y = cost)) + geom_boxplot()
```



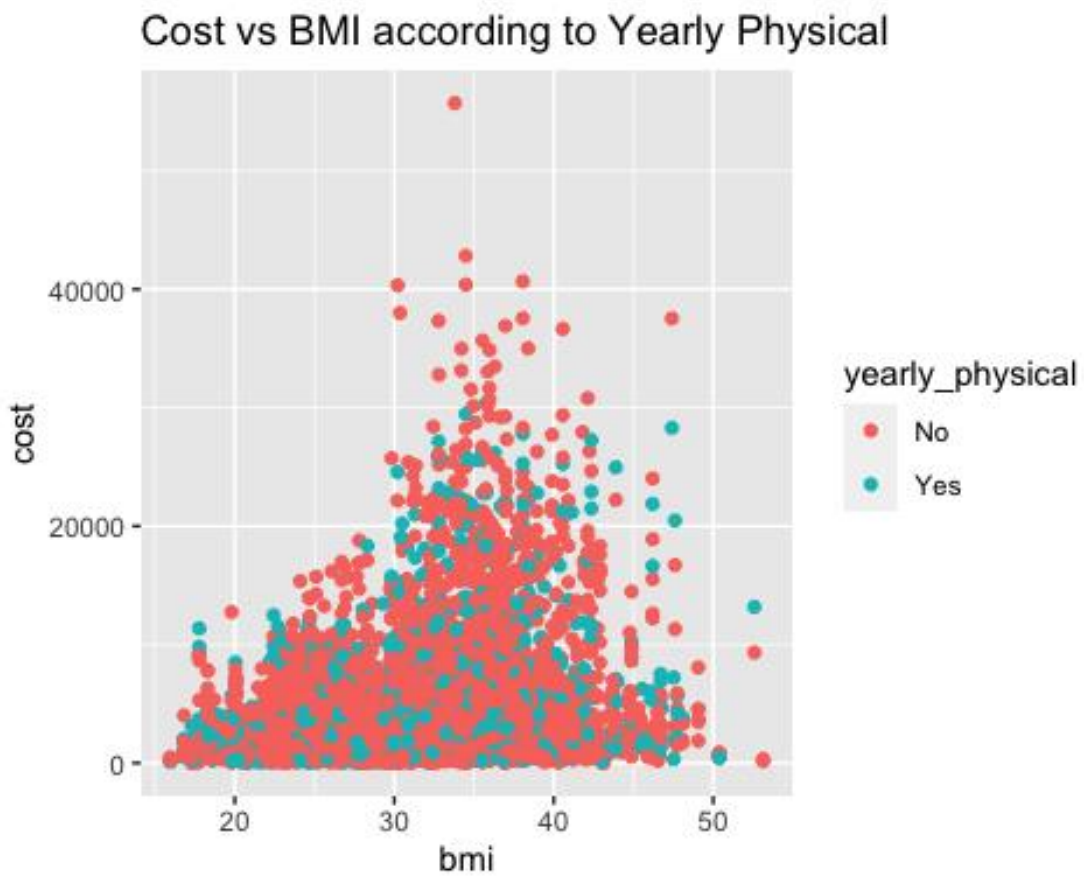

```
#Boxplot of BMI
ggplot(hmo_data, aes(x = bmi)) + geom_boxplot()
```



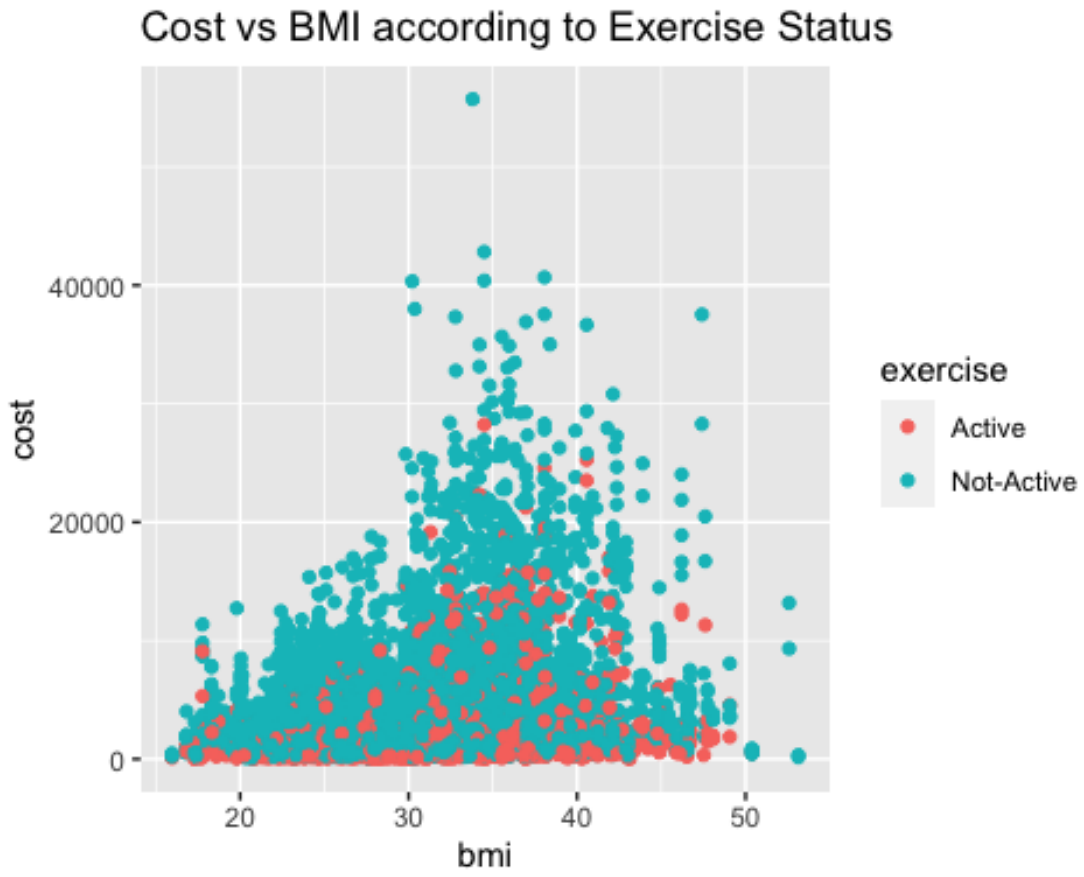

```
#Scatterplot of BMI and Cost with Smoker Status
ggplot(hmo_data)+geom_point(aes(x=bmi ,y=cost, color = smoker))+
ylab('cost')+xlab('bmi')+ggtitle("Cost vs BMI according to Smoker Status")
```



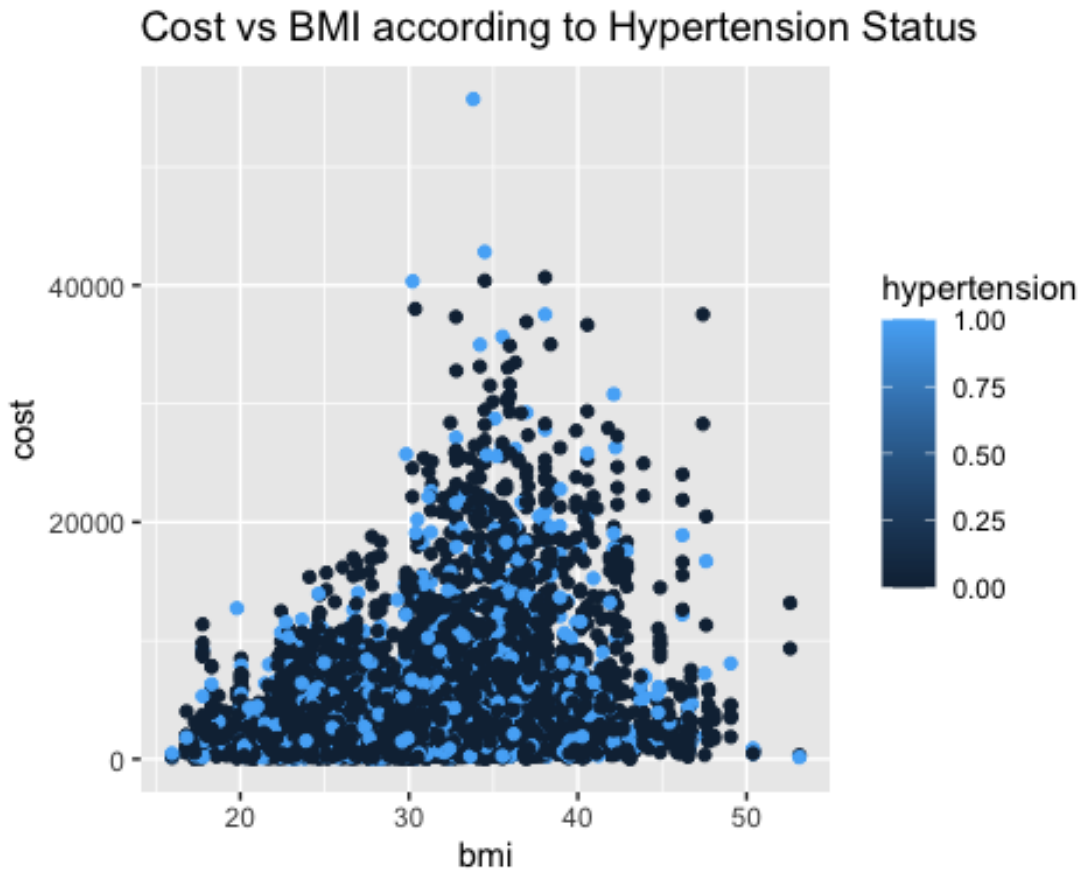

```
#Scatterplot of BMI and Cost with Yearly Physical Checkup
ggplot(hmo_data)+geom_point(aes(x=bmi ,y=cost ,color=yearly_physical))+
ylab('cost')+xlab('bmi')+ggtitle("Cost vs BMI according to Yearly Physical")
```



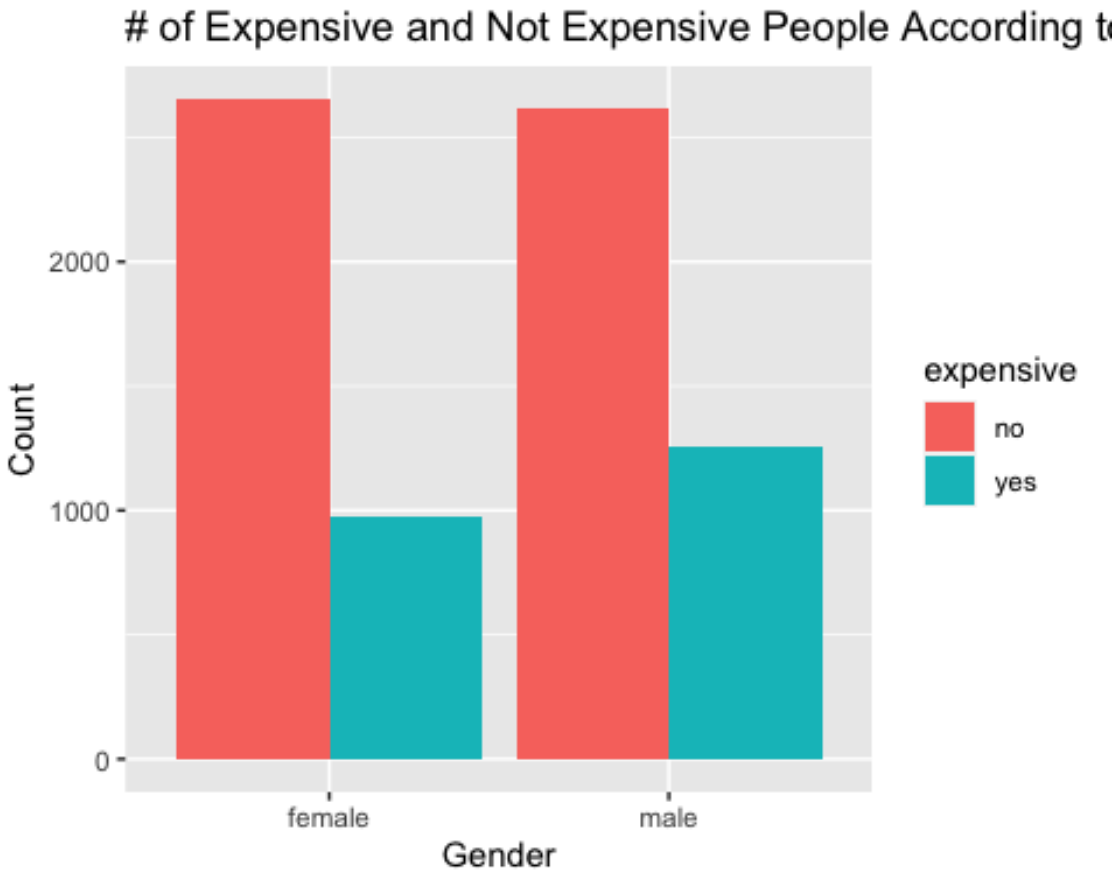

```
#Scatterplot of BMI and Cost with Exercise Status
ggplot(hmo_data)+geom_point(aes(x=bmi ,y=cost ,color=exercise))+
ylab('cost')+xlab('bmi')+ggtitle("Cost vs BMI according to Exercise Status")
```




```
#Scatterplot of BMI and Cost with Hypertension Status
ggplot(hmo_data)+geom_point(aes(x=bmi ,y=cost ,color=hypertension))+
ylab('cost')+xlab('bmi')+ggtitle("Cost vs BMI according to Hypertension
Status")
```




```
#Gender vs Expensive Status
library(ggplot2)
ggplot(hmo_data, aes(x=gender, fill=expensive)) +
  geom_bar(position="dodge") +
  xlab("Gender") +
  ylab("Count") +
  ggtitle("# of Expensive and Not Expensive People According to Gender")
```

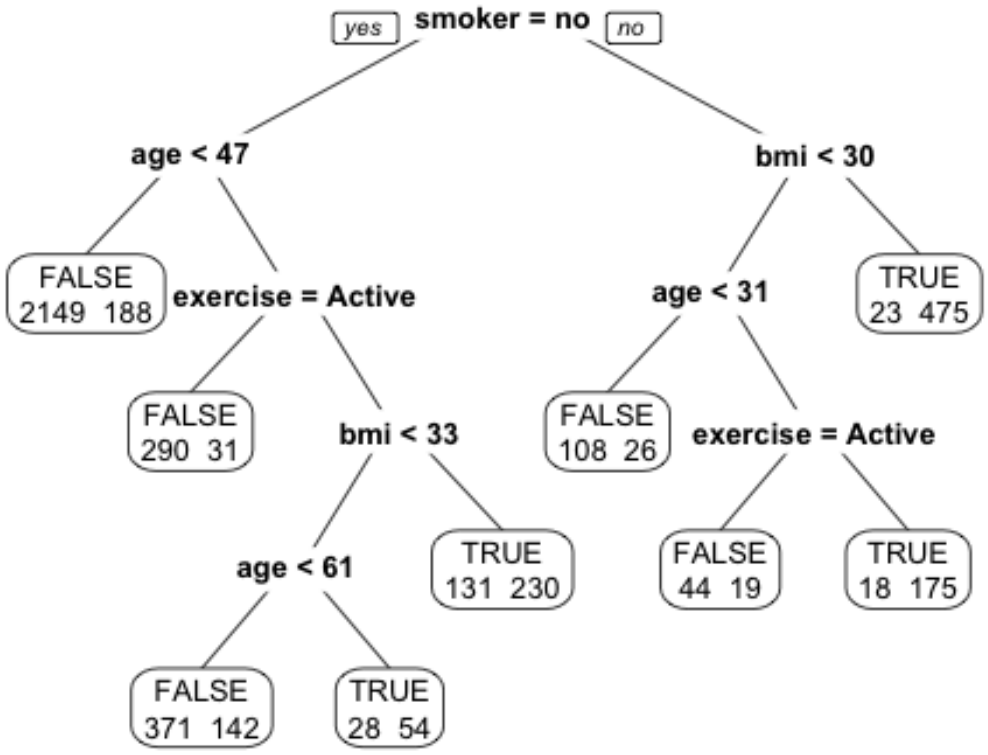



```
#Creating a duplicate dataset from the original dataset to use for model
training
hmodata1 <- data.frame(hmo_data)

#Predictive model svm

hmodata1$cost_status<-with(hmodata1,ifelse(cost>4200,"TRUE","FALSE"))
hmodata1$cost_status<-as.factor(hmodata1$cost_status)

library(caret)

## Loading required package: lattice

##
## Attaching package: 'caret'

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

set.seed(123)
hmodata_model <-data.frame(hmodata1)
#Creating duplicate dataset to utilize for prediction models
trainList <-
```

```
createDataPartition(y=hmodata_model$cost_status,p=.60,list=FALSE) #Creating
data partition of our data frame to create a trainset for model training and
a testset for testing predictions
trainSet <- hmodata_model[trainList,]
testSet <- hmodata_model[-trainList,]
hmodata_svm1 <- train(cost_status ~
X+age+bmi+children+smoker+location+location_type+education_level+yearly_physi
cal +exercise+married+hypertension+gender, data = trainSet ,method =
"svmRadial",trControl=trainControl(method ="none"), preProcess = c("center",
"scale"))
predict_svm <- predict(hmodata_svm1, newdata=testSet)

confusionMatrix(predict_svm, testSet$cost_status)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1988  454
##      TRUE    120  438
##
##                Accuracy : 0.8087
##                  95% CI : (0.7941, 0.8226)
##     No Information Rate : 0.7027
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4867
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9431
##             Specificity : 0.4910
##          Pos Pred Value : 0.8141
##          Neg Pred Value : 0.7849
##              Prevalence : 0.7027
##          Detection Rate : 0.6627
##    Detection Prevalence : 0.8140
##       Balanced Accuracy : 0.7171
##
##        'Positive' Class : FALSE
##

#SVM Model accuracy =80.87%
#SVM Model sensitivity =94.31%

#install.packages("rio")
library(rio)
library(kernlab)

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

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

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

library(rlang)

##
## Attaching package: 'rlang'

## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice

library(caret)
set.seed(123)
hmodata_ksvm1<-
ksvm(data=trainSet,cost_status~X+age+bmi+children+smoker+location+location_ty
pe+education_level+yearly_physical+exercise+married+hypertension+gender,C=5,
cross=3, prob.model=TRUE)
predict_ksvm <- predict(hmodata_ksvm1, newdata=testSet)
confusionMatrix(predict_ksvm, testSet$cost_status)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1986  342
##      TRUE    122  550
##
##                Accuracy : 0.8453
##                  95% CI : (0.8319, 0.8581)
##     No Information Rate : 0.7027
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6015
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9421
##             Specificity : 0.6166
##          Pos Pred Value : 0.8531
##          Neg Pred Value : 0.8185
##              Prevalence : 0.7027
##          Detection Rate : 0.6620
##    Detection Prevalence : 0.7760
##       Balanced Accuracy : 0.7794
```

```
##
##        'Positive' Class : FALSE
##

#KSVM Model Sensitivity 94.21%
#KSVM Model Accuracy 84.53%

#Prediction Model training rpart tree

#install.packages('e1071', dependencies = TRUE)
#install.packages("rpart.plot")
library(rpart)
library(rpart.plot)

hmodata_tree<-data.frame(hmodata1)

Treeplot<-rpart(cost_status ~
X+age+bmi+children+smoker+location+location_type+education_level+yearly_physi
cal +exercise+married+hypertension+gender, data = trainSet, control =
c(maxdepth = 5, cp=0.002))
prp(Treeplot, faclen = 0, cex = 0.8, extra = 1)
```



```
predict_tree <- predict(Treeplot, newdata=testSet, type = "class")

confusionMatrix(predict_tree,testSet$cost_status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1969  299
##      TRUE    139  593
##
##                Accuracy : 0.854
##                  95% CI : (0.8409, 0.8665)
##     No Information Rate : 0.7027
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6315
##
##  Mcnemar's Test P-Value : 3.023e-14
##
##             Sensitivity : 0.9341
##             Specificity : 0.6648
##          Pos Pred Value : 0.8682
##          Neg Pred Value : 0.8101
##              Prevalence : 0.7027
##          Detection Rate : 0.6563
##    Detection Prevalence : 0.7560
##       Balanced Accuracy : 0.7994
##
##        'Positive' Class : FALSE
##
```

```
#Tree Model Sensitivity 93.41%
#Tree Model Accuracy 85.4%
```

Map

```
library(ggplot2);
library(maps);
```

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

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

```
library(ggmap);
```

```
## ℹ Google's Terms of Service:
<]8;;https://mapsplatform.google.comhttps://mapsplatform.google.com]8;;>
```

```
## ℹ Please cite ggmap if you use it! Use `citation("ggmap")` for details.

library(mapproj);
library(tidyverse)
hmodatasortedDF <- hmo_data %>% group_by(location) %>% summarise(avgCost =
mean(hmo_data$cost))
us<- map_data("state")
us$state_name <- tolower(us$region)
hmodatasortedDF$location <- tolower(hmodatasortedDF$location)
mergeddata <-
merge(us,hmodatasortedDF,all.x=TRUE,by.y="location",by.x="state_name")
ggplot(mergeddata, aes(map_id= state_name)) + aes(x=long, y=lat, group=group)
+
geom_polygon(aes(fill = avgCost), color = "black") +
  scale_colour_gradient(low="blue", high="red")+
expand_limits(x=mergeddata$long, y=mergeddata$lat)
```