# Project IDS

1. Loading the file in a dataframe

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6      ✓ purrr   0.3.4
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.2      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
dfhco<-data.frame(read_csv('C:\\SU\\IDS\\Project\\HMO_data.csv',show_col_types = FALSE))
```

2. Finding Missing values in Columns of the dataframe

```
str(dfhco)
```

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

```
sum(is.na(dfhco$X))
```

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

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

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

```
sum(is.na(dfhco$bmi))#78 missing values
```

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

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

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

```
sum(is.na(dfhco$hypertension))#80 missing values
```

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

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

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

3. Clean up the NA's

```
library(imputeTS)
```

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

```
dfhco$bmi<-na_interpolation(dfhco$bmi)
dfhco$hypertension<-na_interpolation(dfhco$hypertension)
summary(dfhco)
```

```
##        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.:34.70   3rd Qu.:2.000
##  Max.   :131101111   Max.   :66.00   Max.   :53.13   Max.   :5.000
##     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
##     gender               cost
##  Length:7582        Min.   :    2
##  Class :character   1st Qu.:  970
##  Mode  :character   Median : 2500
##                     Mean   : 4043
##                     3rd Qu.: 4775
##                     Max.   :55715
```

4. Creating a **expensive** column based on cost

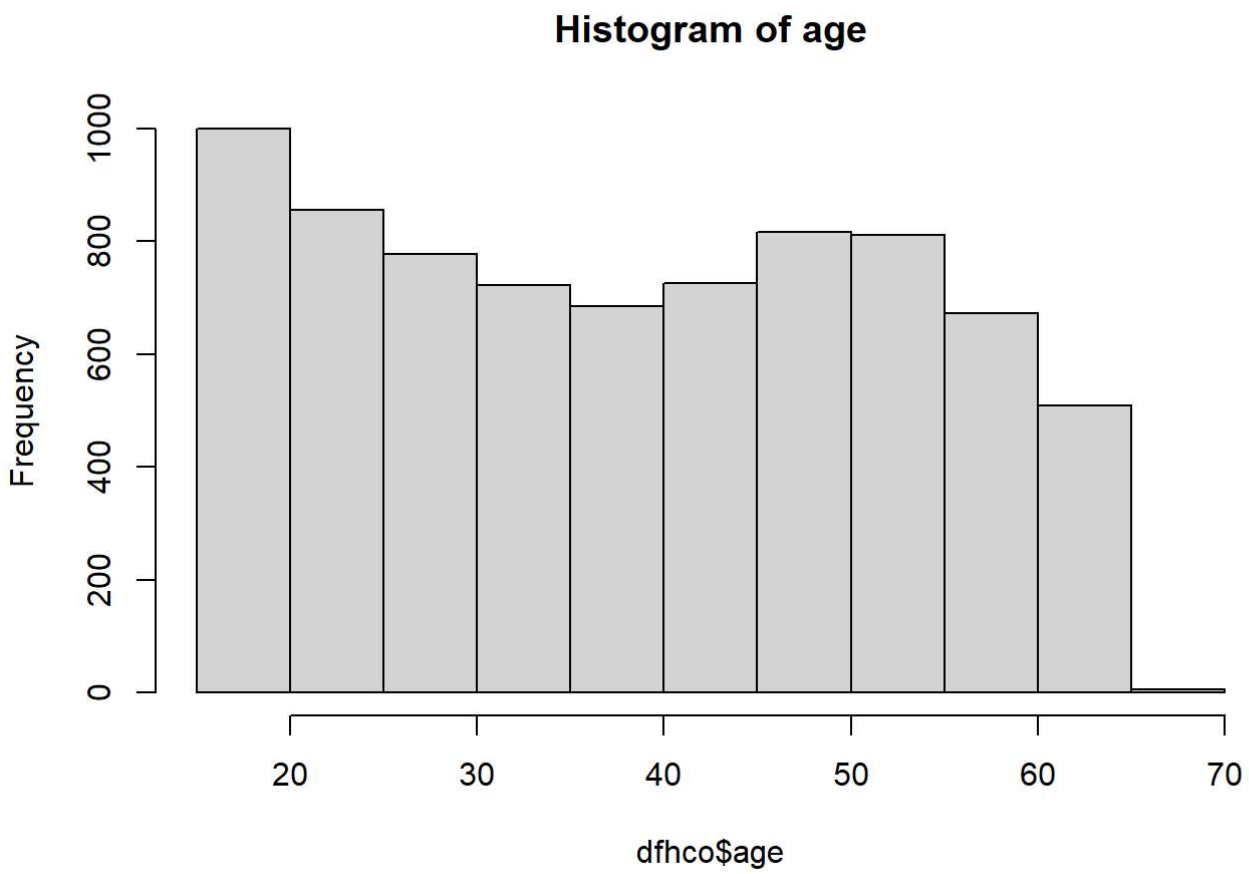
```
quantile(dfhco$cost)
```

```
##    0%   25%   50%   75%  100%
##     2   970  2500  4775 55715
```

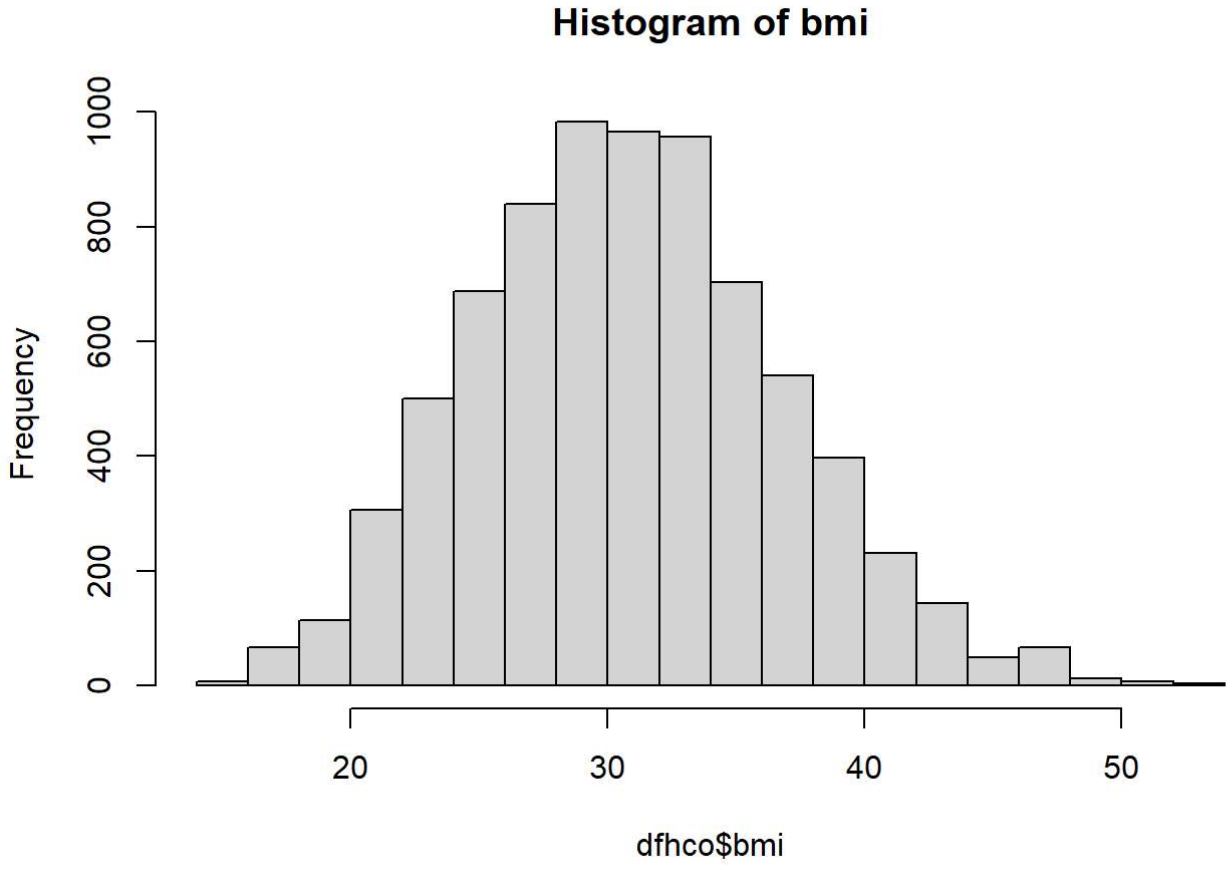
```
#Expensive individuals will be those in the top 25% of the population which is cost greater t
han 4775 according to quantile function
dfhco$expensive <- with(dfhco, ifelse(cost > 4775, 'TRUE', 'FALSE'))
```

5. Plotting various Histogram for numeric variables

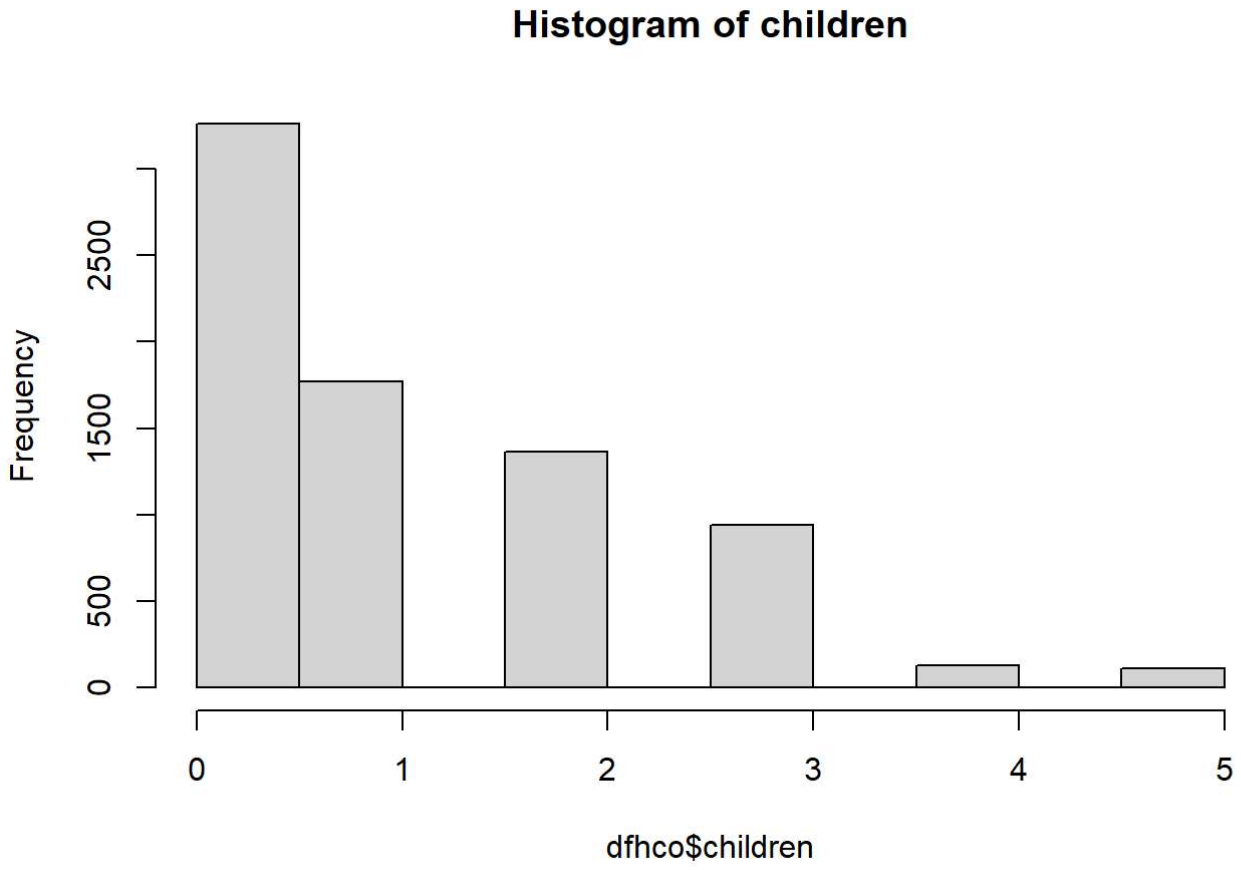
```
hist(dfhco$age,main="Histogram of age")
```



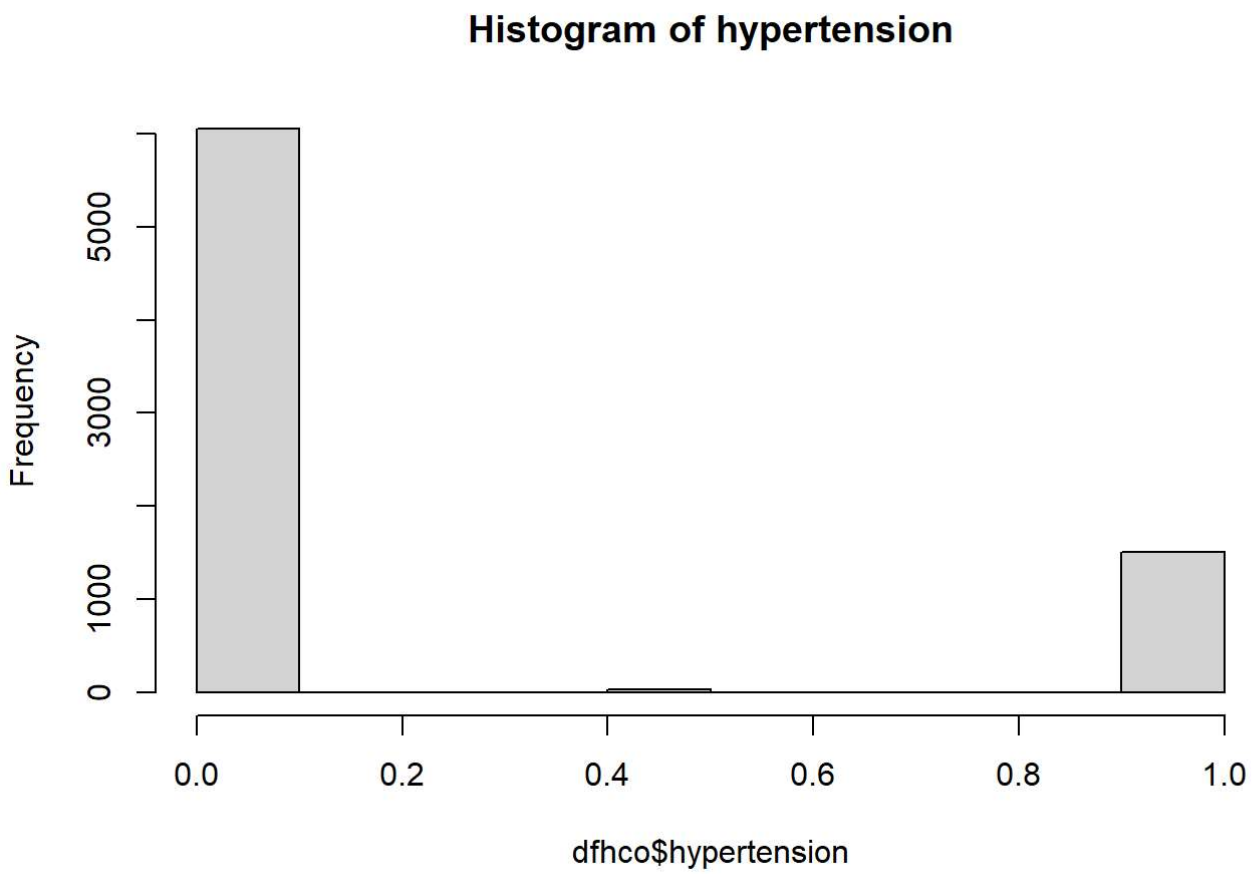

```
hist(dfhco$bmi,main="Histogram of bmi")
```



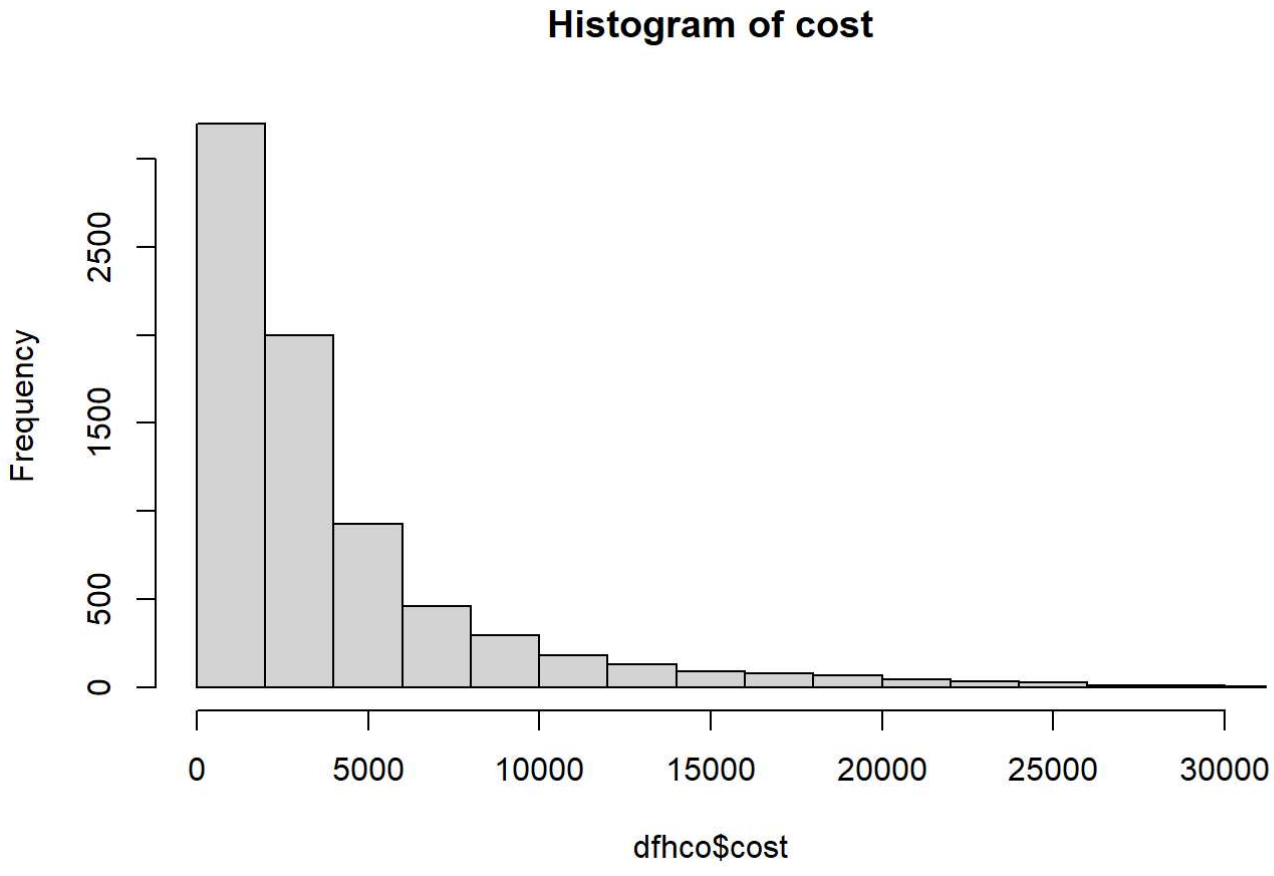
```
hist(dfhco$children,main="Histogram of children")
```




```
hist(dfhco$hypertension,main="Histogram of hypertension")
```

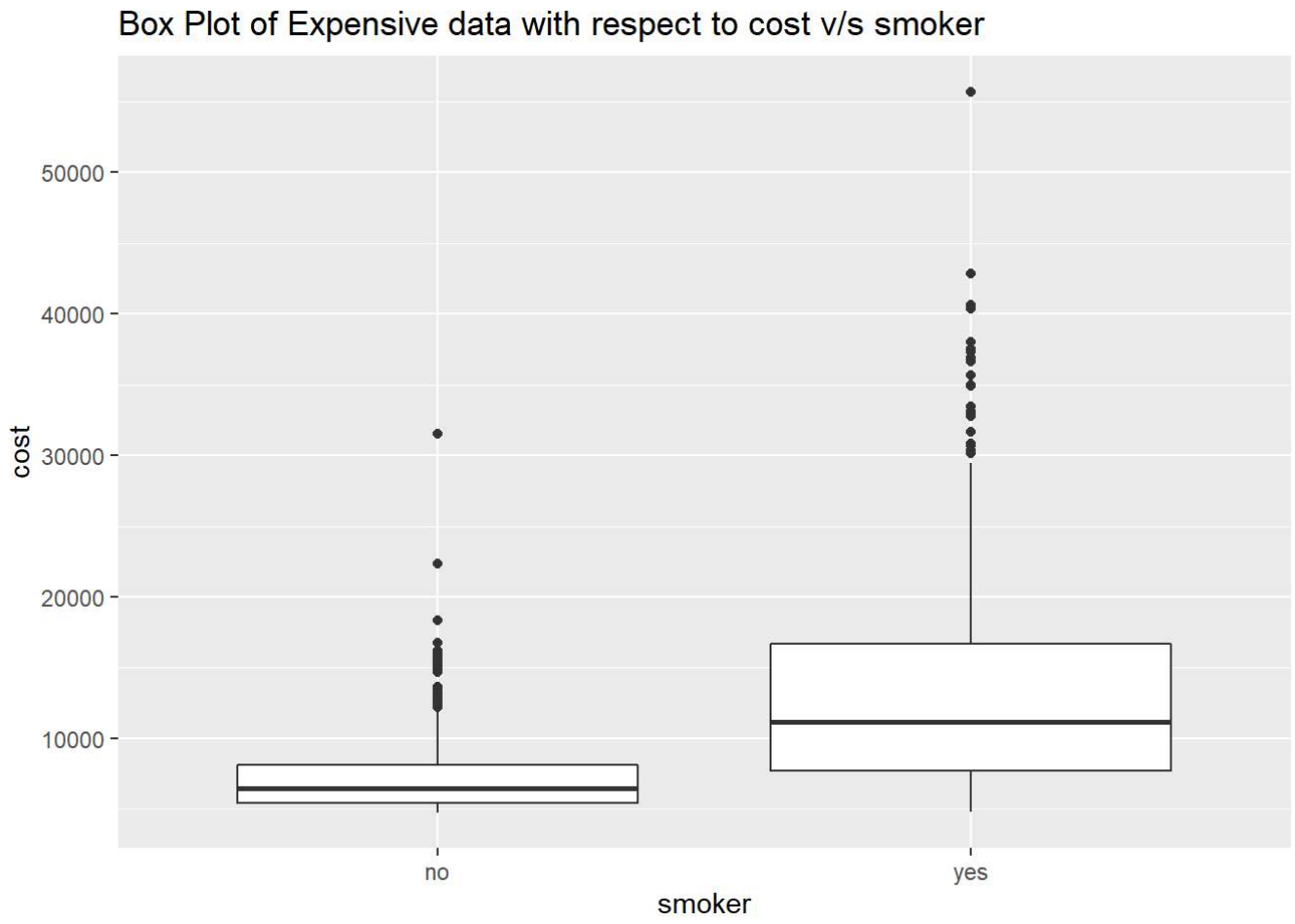



```
hist(dfhco$cost,xlim=c(1,30000),main="Histogram of cost",breaks=20)
```



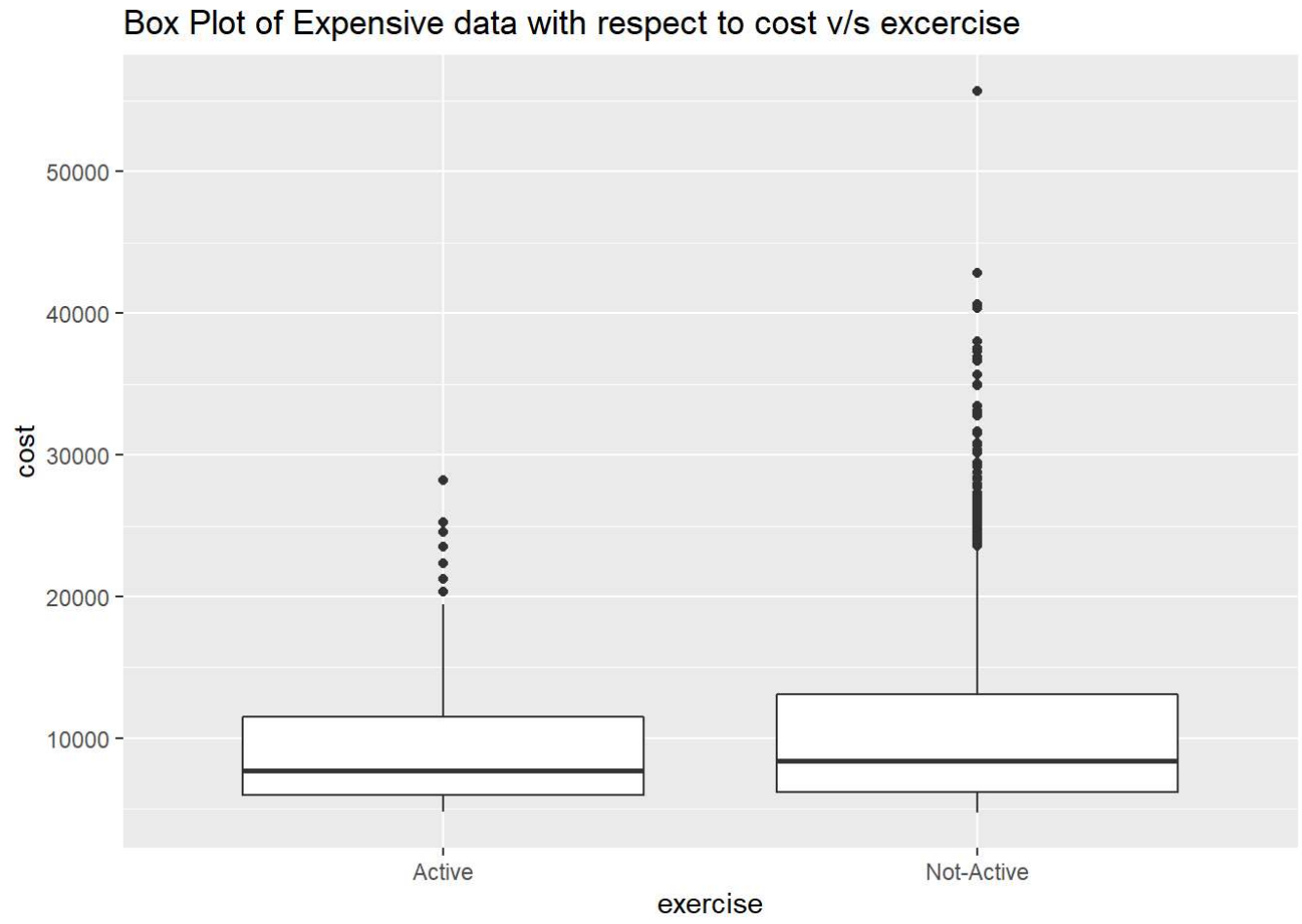

6. Creating Box Plots for expensive and inexpensive data with respect to cost and other variables

```
library(ggplot2)
Expensive <- filter(dfhco, cost > 4775)
Inexpensive <- filter(dfhco, cost <= 4775)
ggplot(Expensive)+aes(x=smoker,y=cost)+geom_boxplot()+ggtitle("Box Plot of Expensive data wit
h respect to cost v/s smoker")
```

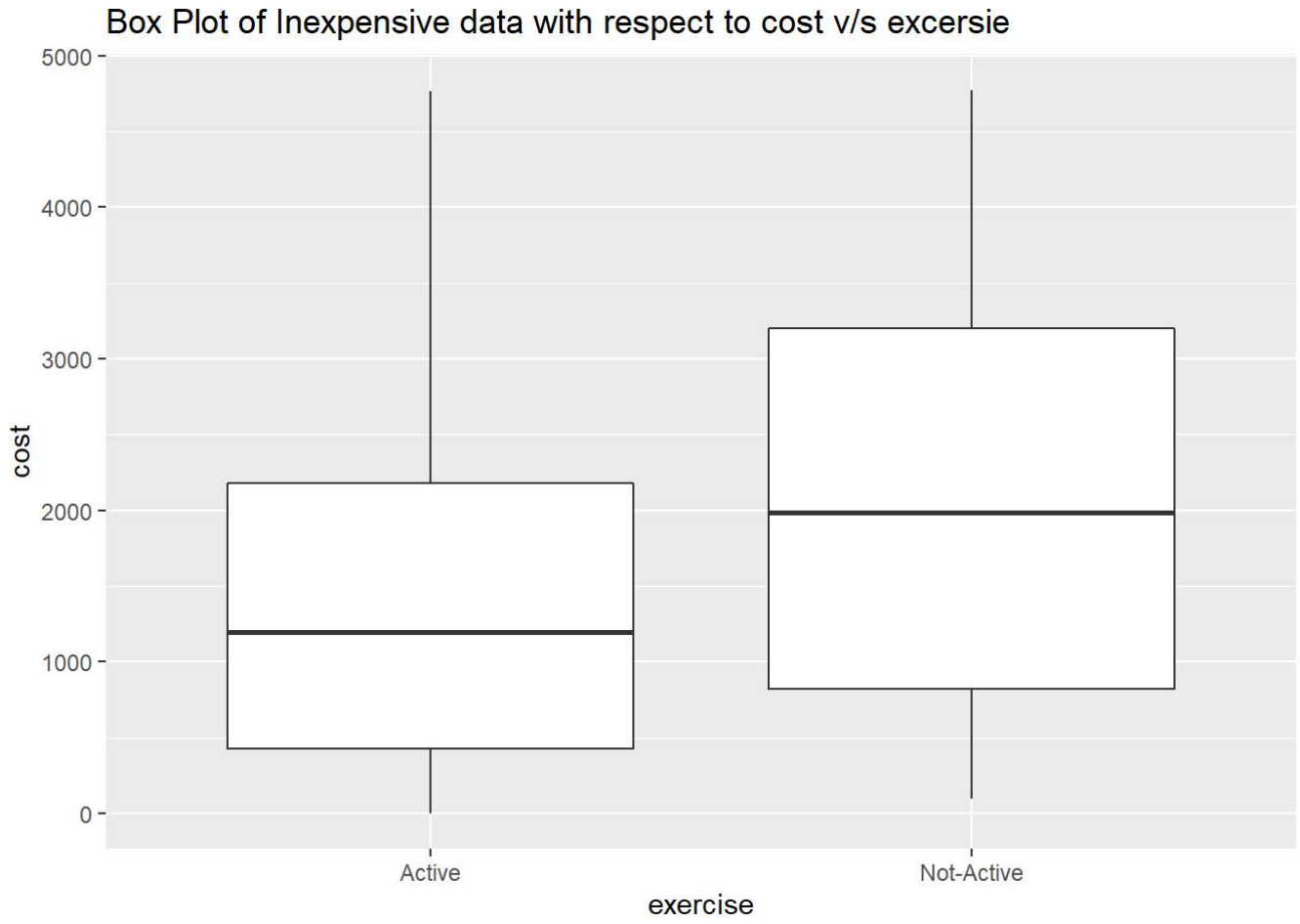



```
ggplot(Inexpensive)+aes(x=smoker,y=cost)+geom_boxplot()+ggtitle("Box Plot of Inexpensive data
with respect to cost v/s smoker")
```

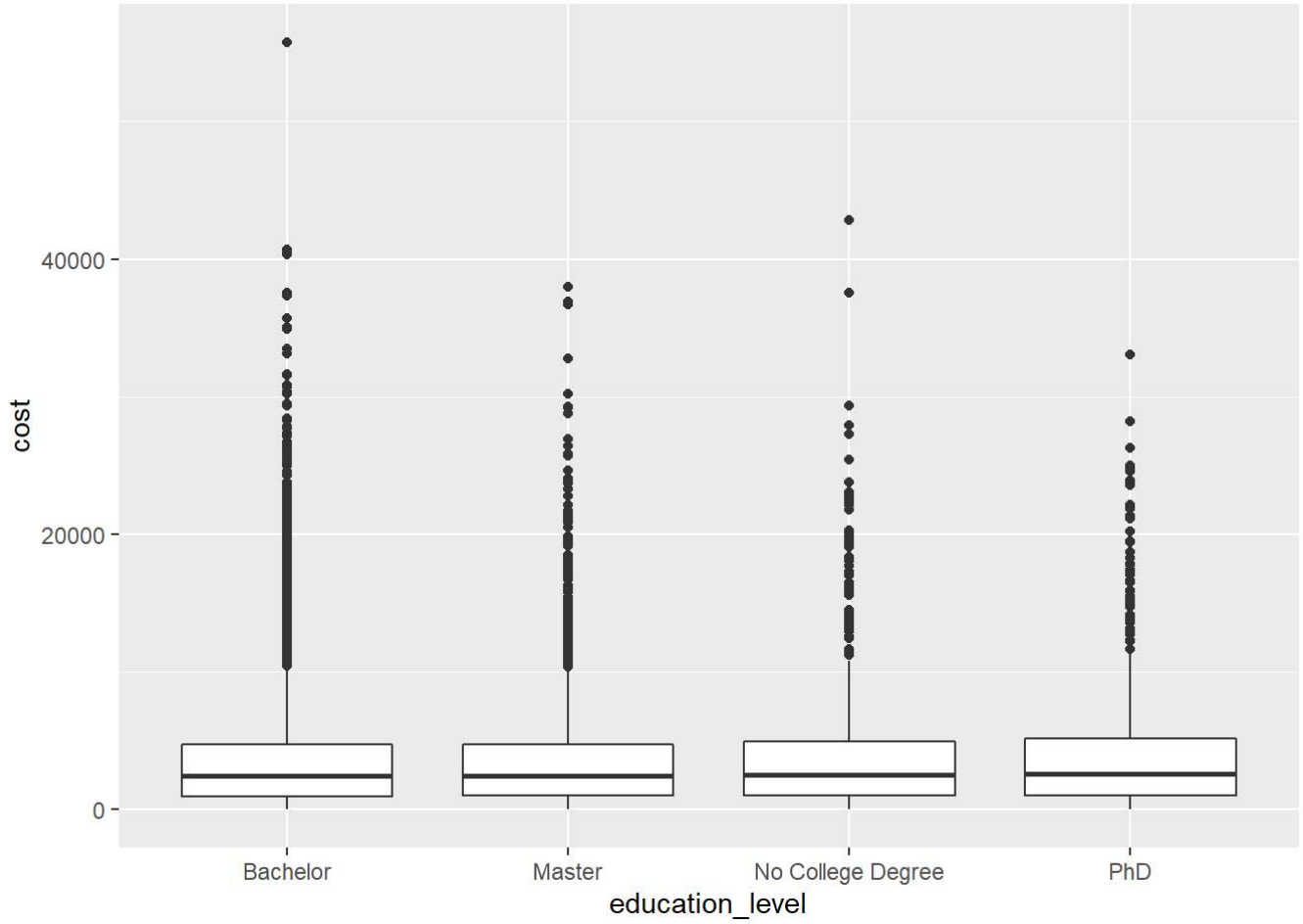

```
ggplot(Expensive)+aes(x=exercise,y=cost)+geom_boxplot()+ggtitle("Box Plot of Expensive data w
ith respect to cost v/s excercise")
```



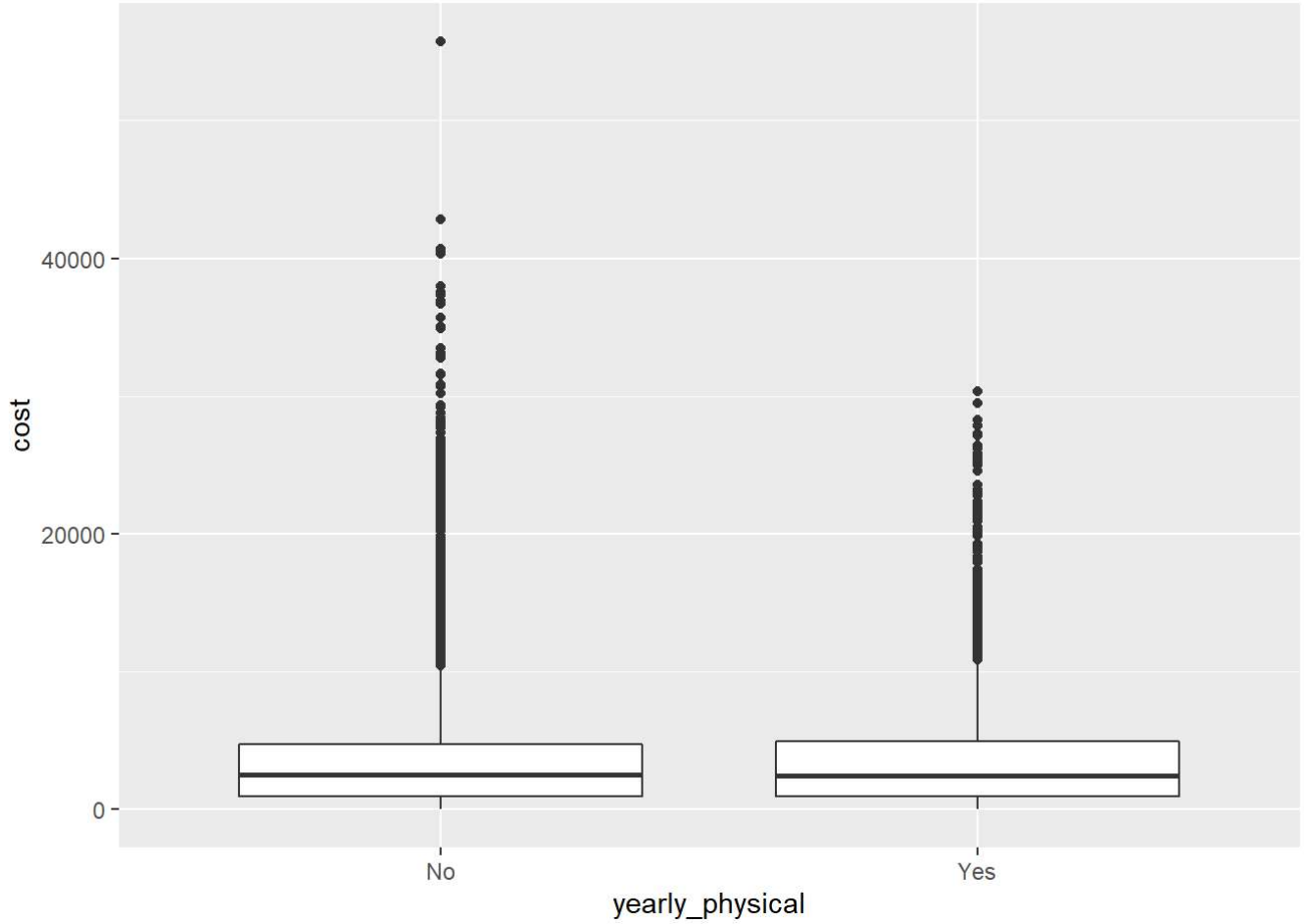
```
ggplot(Inexpensive)+aes(x=exercise,y=cost)+geom_boxplot()+ggtitle("Box Plot of Inexpensive da
ta with respect to cost v/s excersie")
```




```
ggplot(dfhco)+aes(x=education_level,y=cost)+geom_boxplot()
```




```
ggplot(dfhco)+aes(x=yearly_physical,y=cost)+geom_boxplot()
```




7. Plotting Data on US Map with color shading as per cost

```
library(maps)
```

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

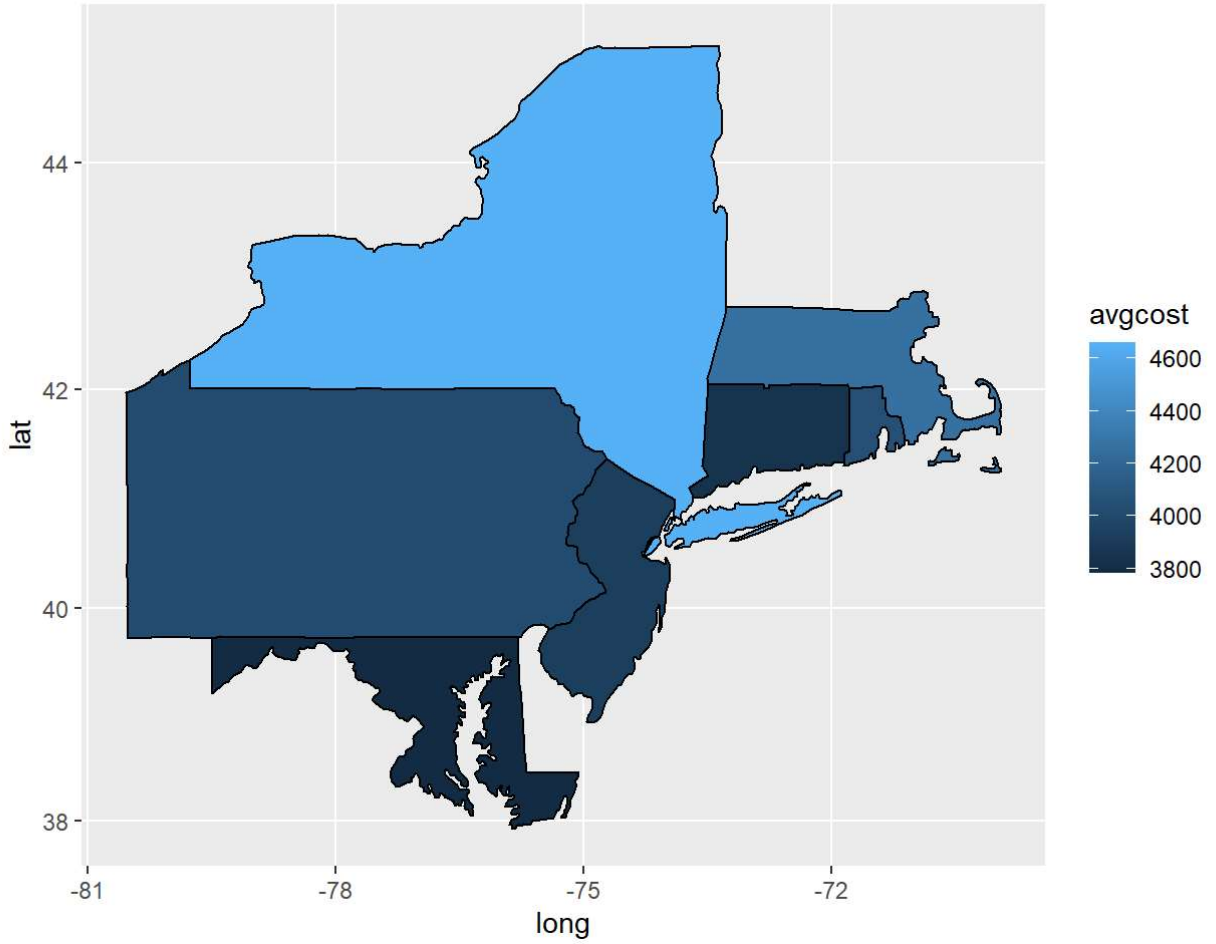
```
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(ggmap)
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

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

```
library(mapproj)
DF <- dfhco %>% group_by(location) %>% summarise(avgcost = mean(cost))
us<- map_data("state")
us$state_name <- tolower(us$region)
DF$location <- tolower(DF$location)
mappy <- merge(us,DF,by.x="state_name",by.y="location")
mappy<-mappy %>% arrange(order)
ggplot(mappy, aes(map_id= state_name))+aes(x=long, y=lat, group=group) +geom_polygon(aes(fill
= avgcost), color = "black")+expand_limits(x=mappy$long, y=mappy$lat)+coord_map(projection =
"mercator")
```




8. Using Association Rules to determine patterns i.e to get which other columns are more frequently appeared when cost was expensive

```
library(arules)
```

```
## Warning: package 'arules' was built under R version 4.2.2
```

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

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

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

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

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

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

```
dfhcon<-dfhco[,-14]
ruleset1<- apriori(dfhcon,
 parameter=list(supp=0.09, conf=0.3),
 control=list(verbose=F),
 appearance=list(default="lhs",rhs=("expensive=TRUE")))
```

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

```
## Warning in discretize(x = c(0, 1, 3, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, : The calculated
breaks are: 0, 0, 2, 5
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for d
etails.
```

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

```
inspect(ruleset1)
```

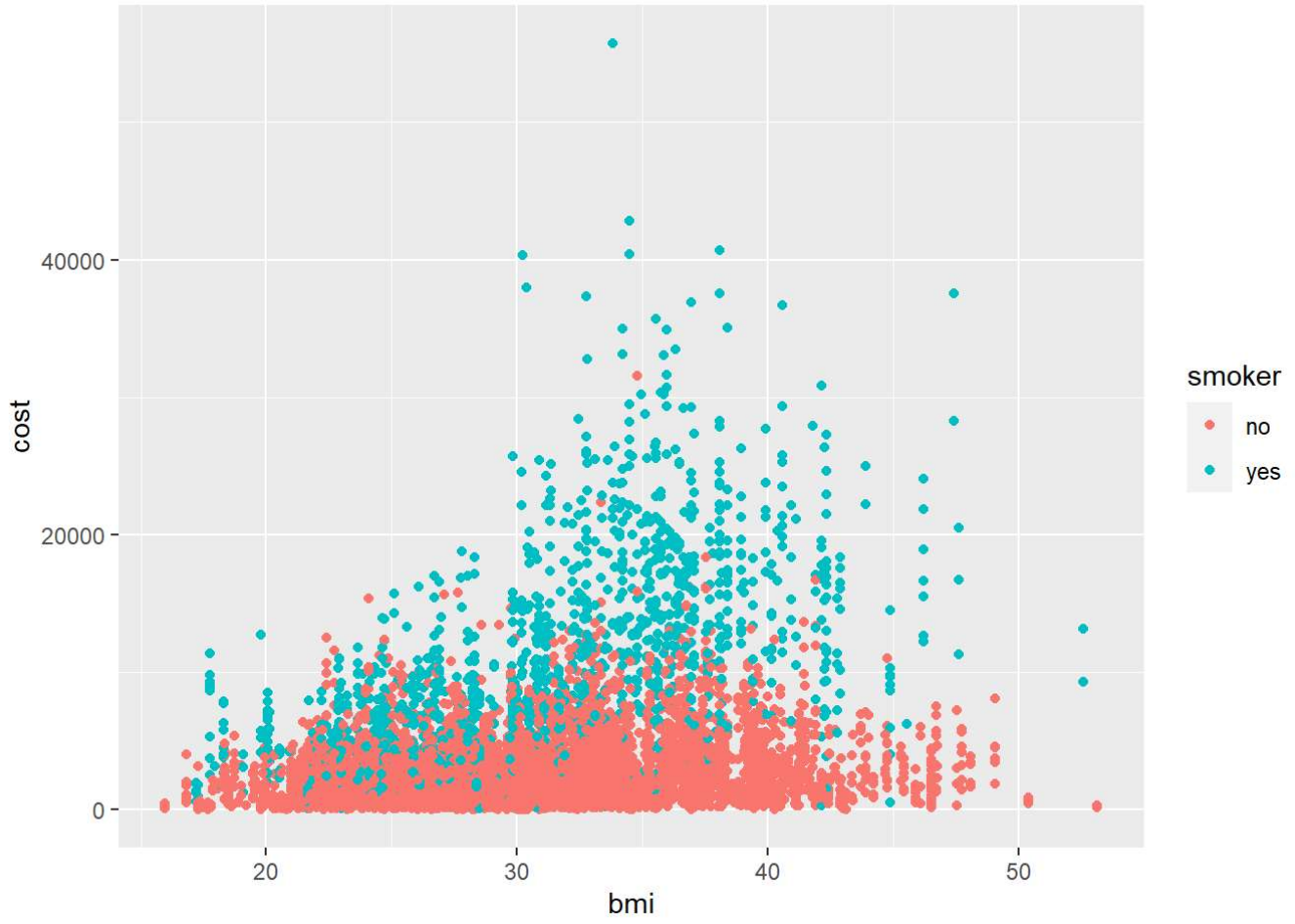
```
##      lhs                       rhs                 support confidence  coverage     lift c
ount
## [1]  {smoker=yes}           => {expensive=TRUE} 0.14244263  0.7302231 0.1950673 2.921663
1080
## [2]  {bmi=[33.1,53.1]}      => {expensive=TRUE} 0.12371406  0.3706045 0.3338169 1.482809
938
## [3]  {age=[47,66]}          => {expensive=TRUE} 0.13175943  0.3779796 0.3485888 1.512317
999
## [4]  {smoker=yes,
##       married=Married}      => {expensive=TRUE} 0.09575310  0.7438525 0.1287259 2.976195
726
## [5]  {smoker=yes,
##       location_type=Urban}  => {expensive=TRUE} 0.10670008  0.7262118 0.1469269 2.905614
809
## [6]  {smoker=yes,
##       exercise=Not-Active}  => {expensive=TRUE} 0.11685571  0.8158379 0.1432340 3.264213
886
## [7]  {smoker=yes,
##       yearly_physical=No}   => {expensive=TRUE} 0.10656819  0.7169476 0.1486415 2.868547
808
## [8]  {smoker=yes,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.14244263  0.7302231 0.1950673 2.921663
1080
## [9]  {bmi=[33.1,53.1],
##       location_type=Urban}  => {expensive=TRUE} 0.09087312  0.3633966 0.2500659 1.453970
689
## [10] {bmi=[33.1,53.1],
##       exercise=Not-Active}  => {expensive=TRUE} 0.10524927  0.4237918 0.2483514 1.695615
798
## [11] {bmi=[33.1,53.1],
##       yearly_physical=No}   => {expensive=TRUE} 0.09403851  0.3694301 0.2545503 1.478110
713
## [12] {bmi=[33.1,53.1],
##       hypertension=[0,1]}   => {expensive=TRUE} 0.12371406  0.3706045 0.3338169 1.482809
938
## [13] {age=[47,66],
##       location_type=Urban}  => {expensive=TRUE} 0.09641256  0.3733401 0.2582432 1.493755
731
## [14] {age=[47,66],
##       exercise=Not-Active}  => {expensive=TRUE} 0.11593247  0.4441637 0.2610129 1.777124
879
## [15] {age=[47,66],
##       yearly_physical=No}   => {expensive=TRUE} 0.10089686  0.3800298 0.2654972 1.520520
765
## [16] {age=[47,66],
##       hypertension=[0,1]}   => {expensive=TRUE} 0.13175943  0.3779796 0.3485888 1.512317
999
## [17] {exercise=Not-Active,
##       gender=male}          => {expensive=TRUE} 0.12437352  0.3216235 0.3867054 1.286833
943
## [18] {smoker=yes,
##       married=Married,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.09575310  0.7438525 0.1287259 2.976195
726
## [19] {smoker=yes,
```

```
##       location_type=Urban,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.10670008  0.7262118 0.1469269 2.905614
809
## [20] {smoker=yes,
##       exercise=Not-Active,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.11685571  0.8158379 0.1432340 3.264213
886
## [21] {smoker=yes,
##       yearly_physical=No,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.10656819  0.7169476 0.1486415 2.868547
808
## [22] {bmi=[33.1,53.1],
##       location_type=Urban,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.09087312  0.3633966 0.2500659 1.453970
689
## [23] {bmi=[33.1,53.1],
##       exercise=Not-Active,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.10524927  0.4237918 0.2483514 1.695615
798
## [24] {bmi=[33.1,53.1],
##       yearly_physical=No,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.09403851  0.3694301 0.2545503 1.478110
713
## [25] {age=[47,66],
##       location_type=Urban,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.09641256  0.3733401 0.2582432 1.493755
731
## [26] {age=[47,66],
##       exercise=Not-Active,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.11593247  0.4441637 0.2610129 1.777124
879
## [27] {age=[47,66],
##       yearly_physical=No,
##       hypertension=[0,1]}   => {expensive=TRUE} 0.10089686  0.3800298 0.2654972 1.520520
765
## [28] {location_type=Urban,
##       exercise=Not-Active,
##       gender=male}          => {expensive=TRUE} 0.09179636  0.3175182 0.2891058 1.270408
696
## [29] {yearly_physical=No,
##       exercise=Not-Active,
##       gender=male}          => {expensive=TRUE} 0.09219203  0.3140162 0.2935901 1.256396
699
## [30] {exercise=Not-Active,
##       hypertension=[0,1],
##       gender=male}          => {expensive=TRUE} 0.12437352  0.3216235 0.3867054 1.286833
943
## [31] {location_type=Urban,
##       exercise=Not-Active,
##       hypertension=[0,1],
##       gender=male}          => {expensive=TRUE} 0.09179636  0.3175182 0.2891058 1.270408
696
## [32] {yearly_physical=No,
##       exercise=Not-Active,
##       hypertension=[0,1],
```

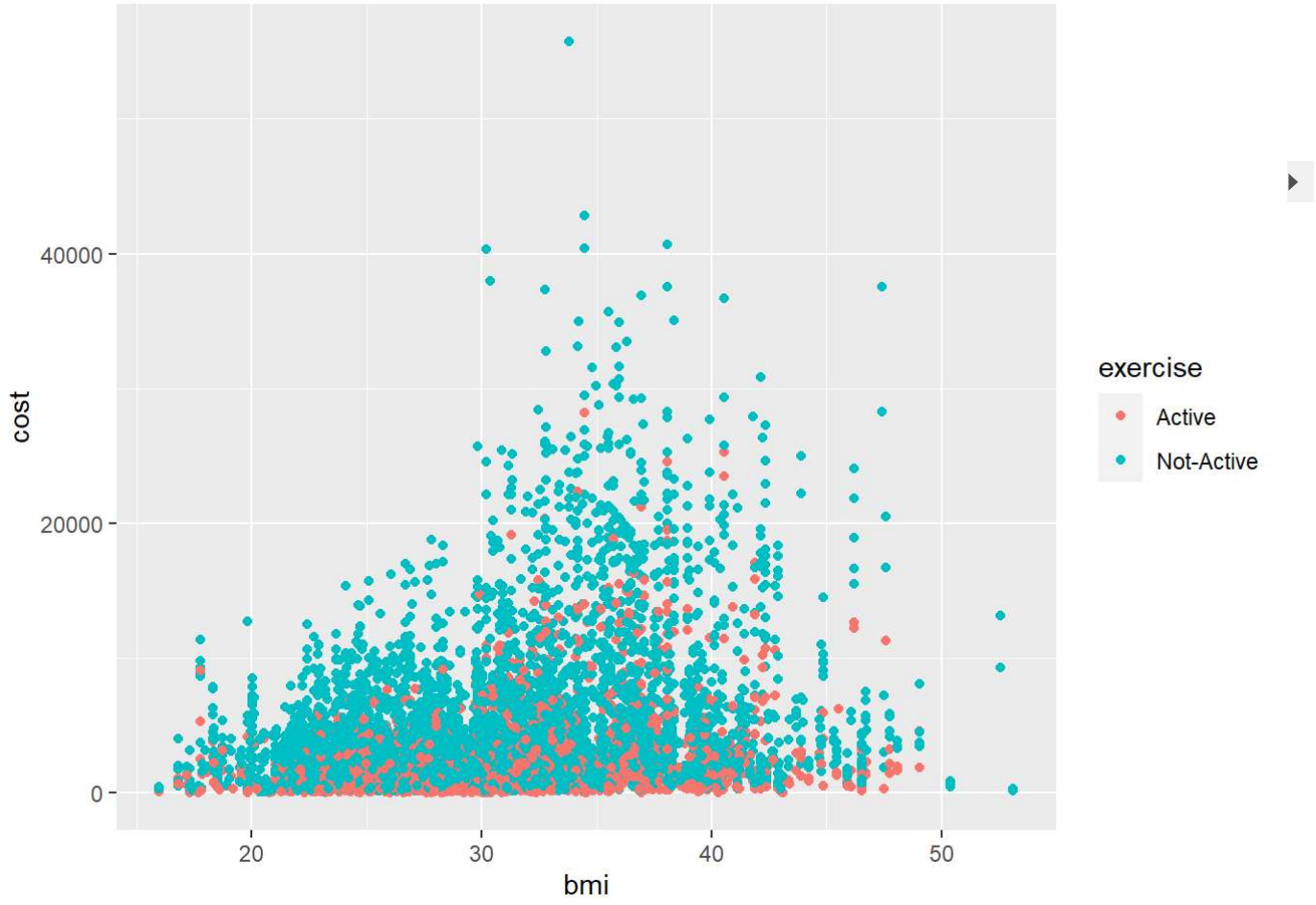
```
##       gender=male}          => {expensive=TRUE} 0.09219203  0.3140162 0.2935901 1.256396
699
```

9. Creating Scatter Plot to determine distribution of data with respect to cost

```
ggplot(dfhco)+aes(x=bmi,y=cost,color=smoker)+geom_point()
```

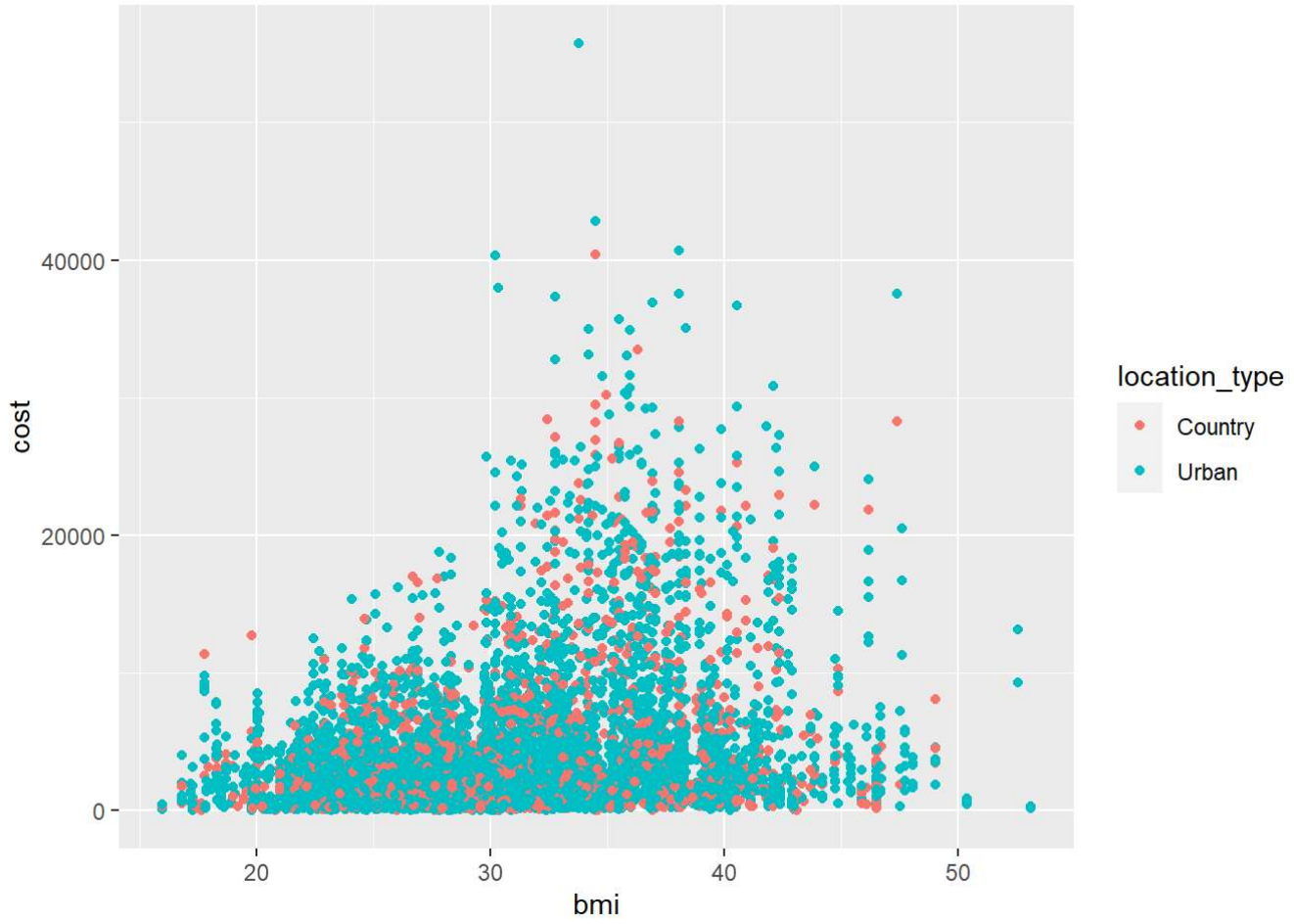



```
ggplot(dfhco)+aes(x=bmi,y=cost,color=exercise)+geom_point()
```



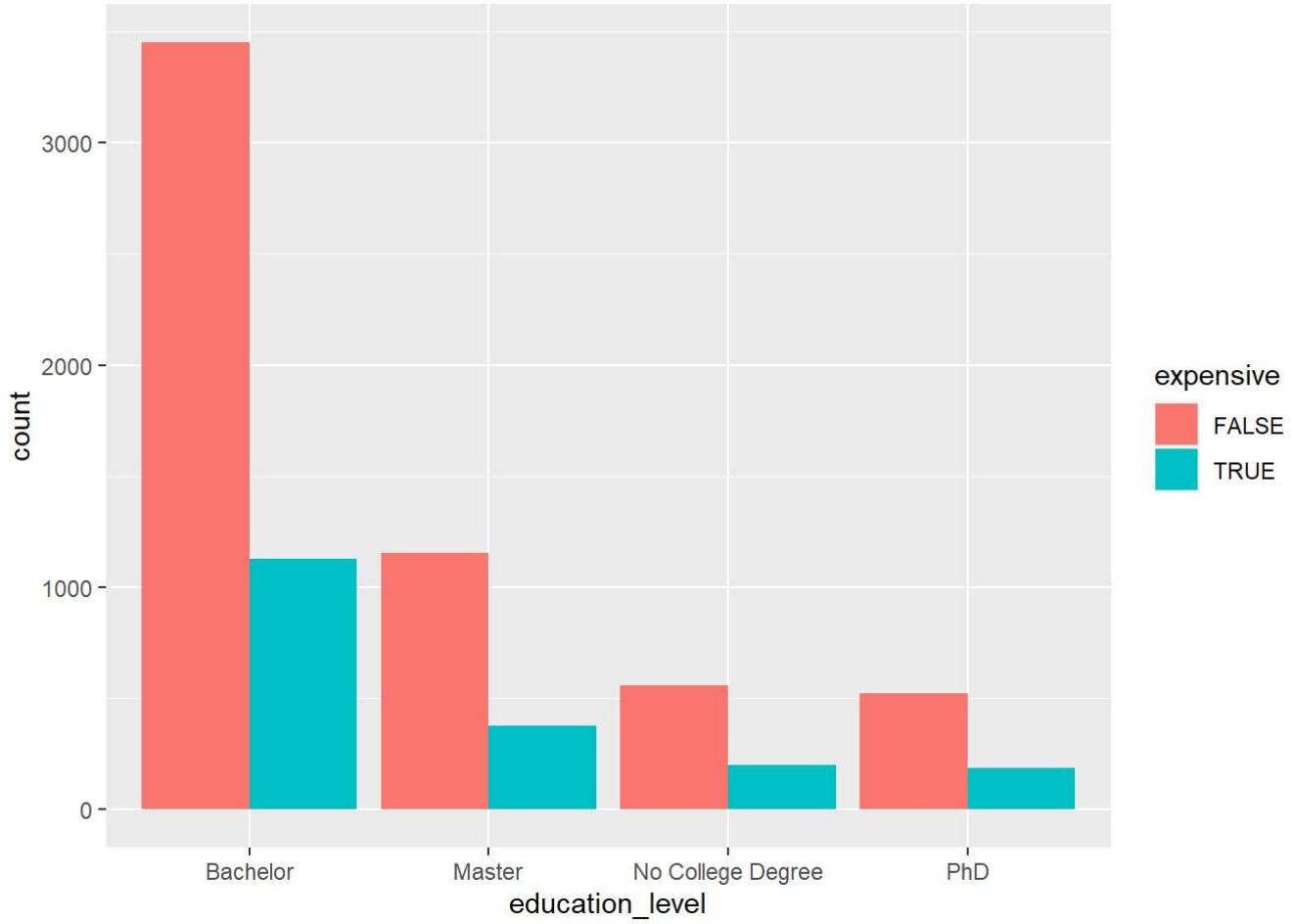

```
ggplot(dfhco)+aes(x=bmi,y=cost,color=yearly_physical)+geom_point()
```

```
ggplot(dfhco)+aes(x=bmi,y=cost,color=location_type)+geom_point()
```

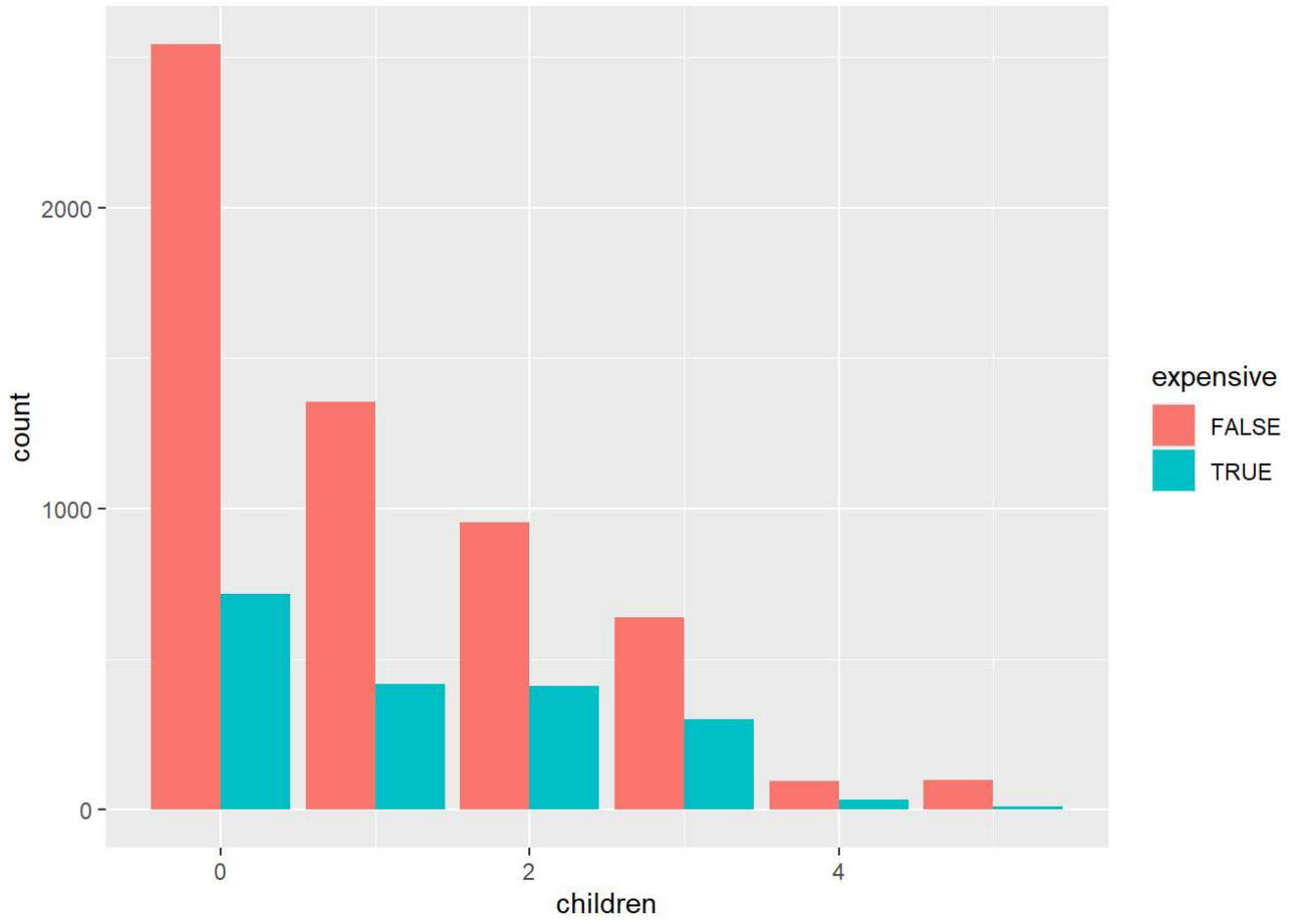



10. Genrating Barplots

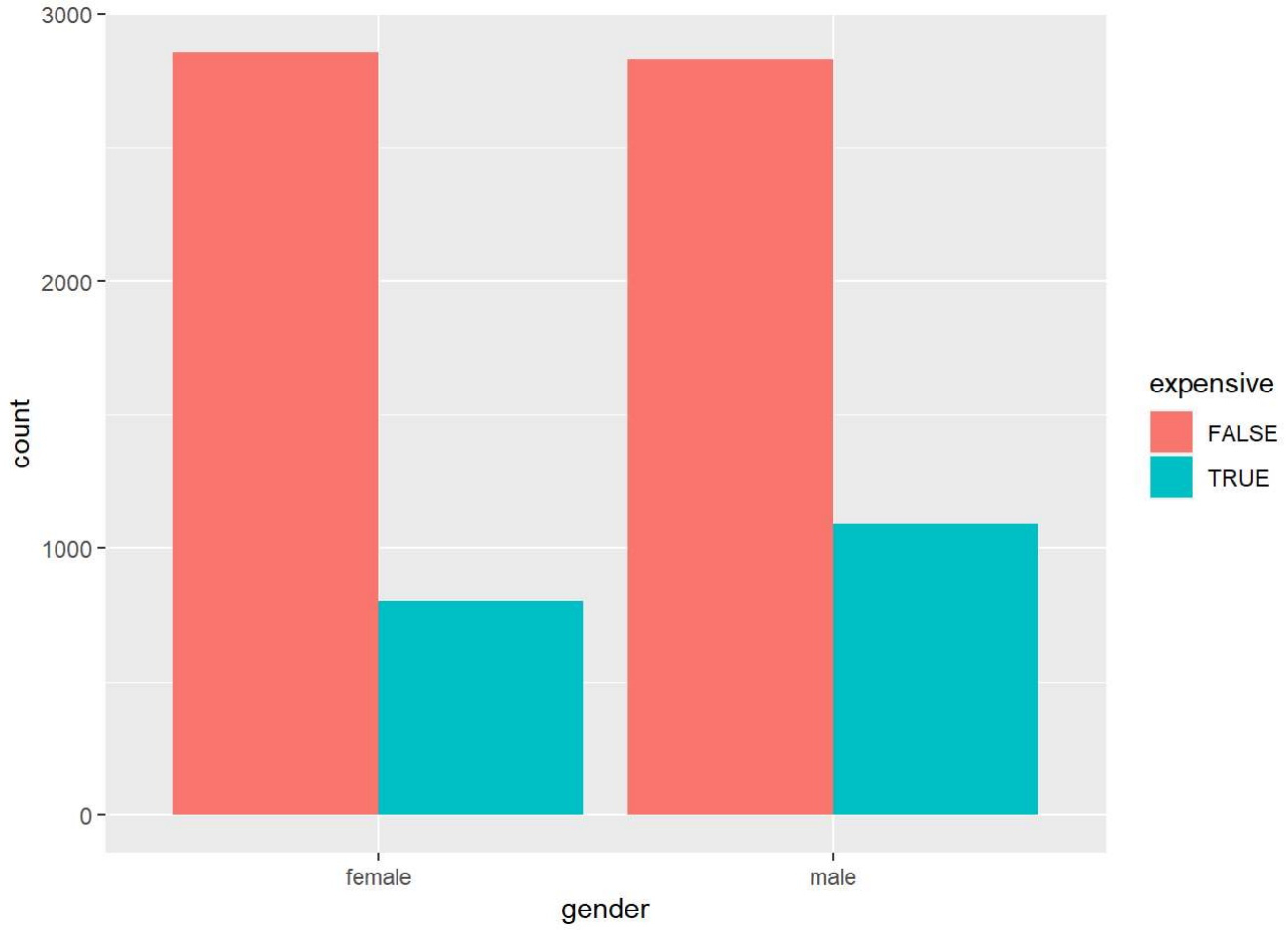
```
ggplot(dfhco)+aes(x=education_level)+geom_bar(position="dodge",aes(fill=expensive))
```



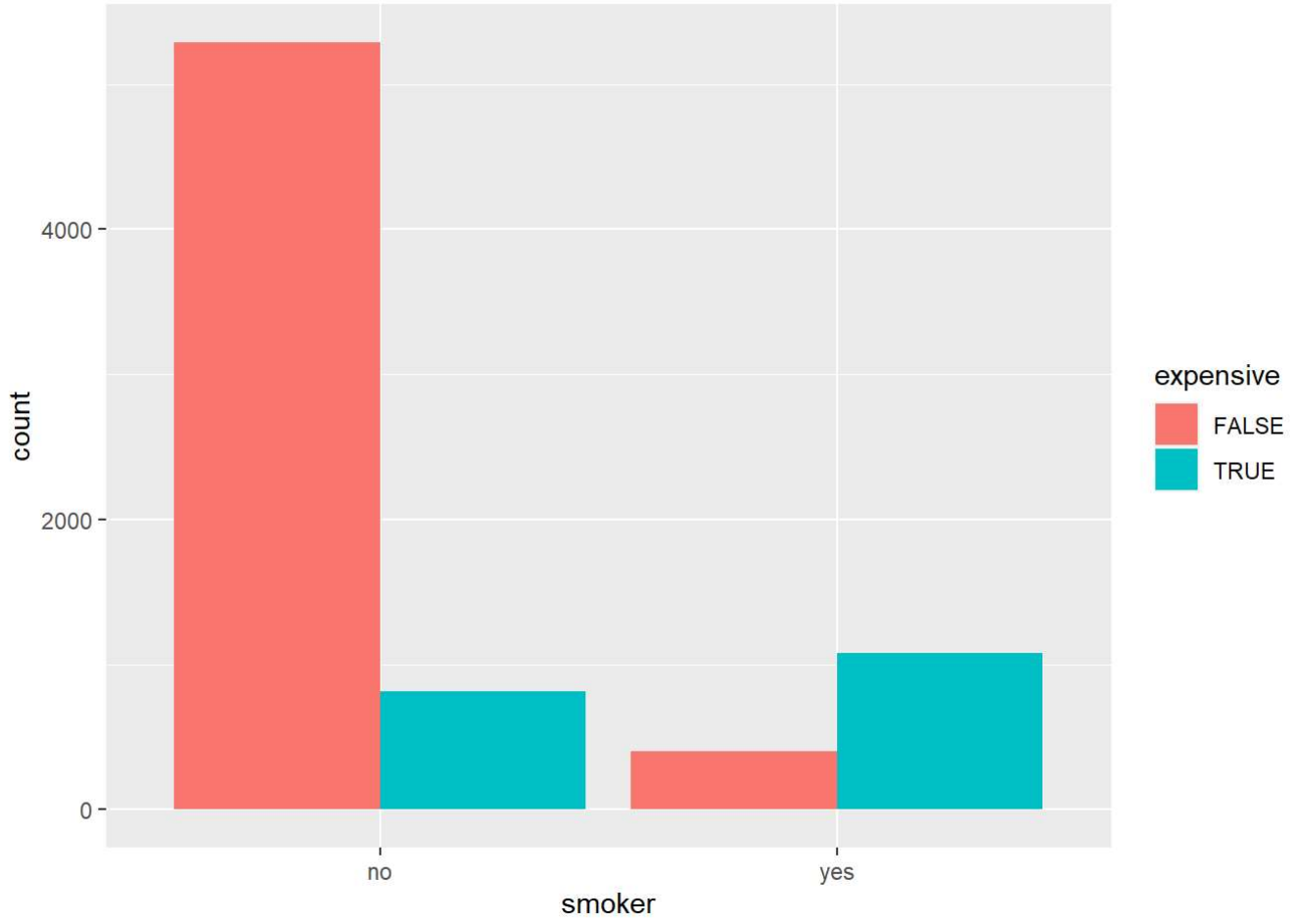

```
ggplot(dfhco)+aes(x=children)+geom_bar(position="dodge",aes(fill=expensive))
```



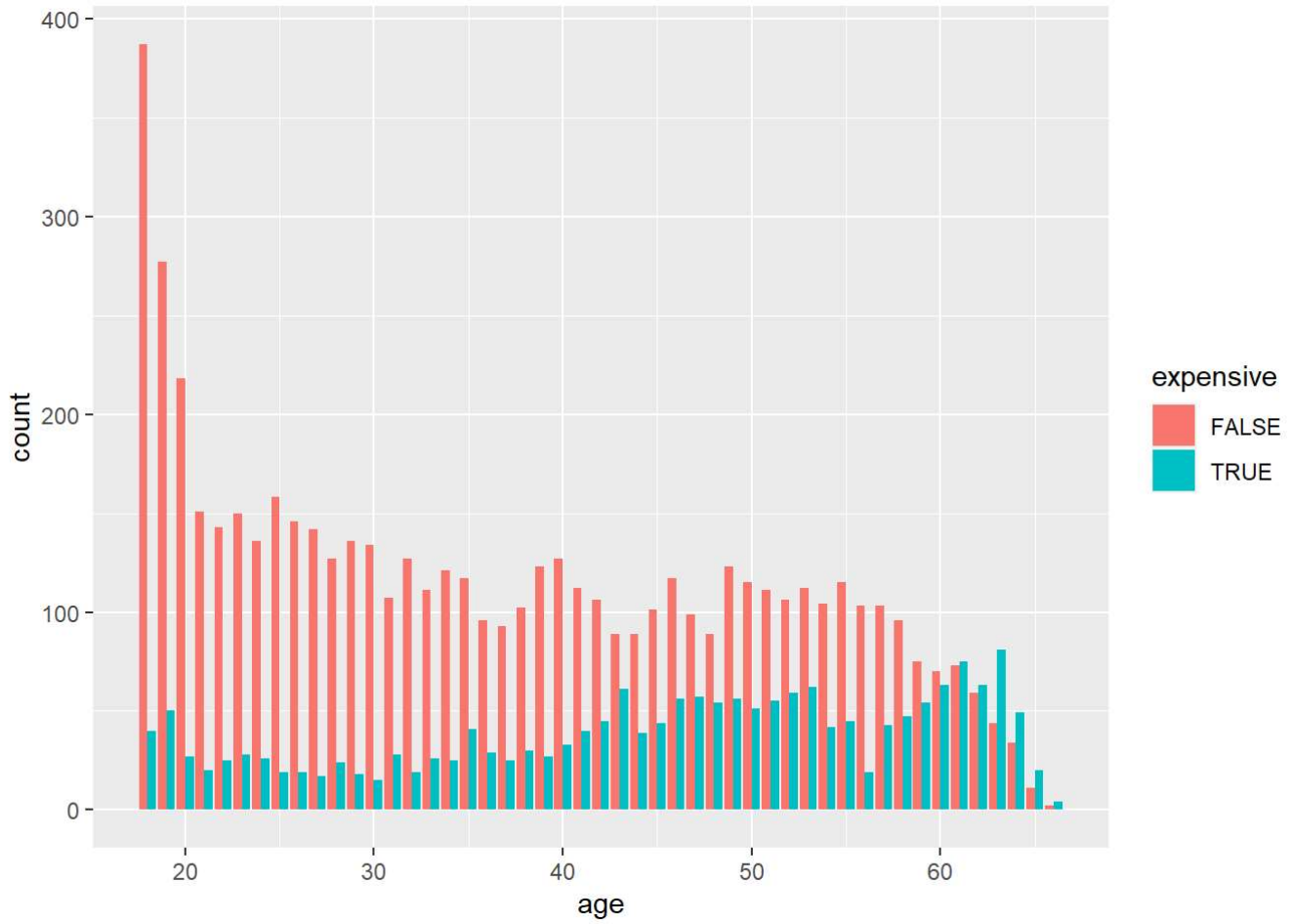
```
ggplot(dfhco)+aes(x=gender)+geom_bar(position="dodge",aes(fill=expensive))
```




```
ggplot(dfhco)+aes(x=smoker)+geom_bar(position="dodge",aes(fill=expensive))
```




```
ggplot(dfhco)+aes(x=age)+geom_bar(position="dodge",aes(fill=expensive))
```



11. Training Model with SVM Against the dataset

```
set.seed(111)
library(rio)
```

```
## Warning: package 'rio' was built under R version 4.2.2
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.2
```

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

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

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

```
library(kernlab)
```

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

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

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

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

```
dfhco$expensive<-as.factor(dfhco$expensive)
trainList <- createDataPartition(dfhco$expensive, p=.7, list=FALSE)
trainSet <- dfhco[trainList,]
testSet <- dfhco[-trainList,]
svmmodel<- train(expensive ~ X+age+bmi+children+smoker+location+location_type+education_level
+yearly_physical+exercise+married+hypertension+gender,data=trainSet , method= "svmRadial", tr
Control=trainControl(method = "none"), preProcess=c("center","scale"))
```

12. Use predict function to validate model against the test data

```
svmPred<-predict(svmmodel,testSet,type='raw')
```

13. Creating Confusion Matrix.

```
confusionMatrix(svmPred,testSet$expensive)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1640  259
##      TRUE     66  309
##
##                Accuracy : 0.8571
##                  95% CI : (0.842, 0.8712)
##     No Information Rate : 0.7502
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5699
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9613
##             Specificity : 0.5440
##          Pos Pred Value : 0.8636
##          Neg Pred Value : 0.8240
##              Prevalence : 0.7502
##          Detection Rate : 0.7212
##    Detection Prevalence : 0.8351
##       Balanced Accuracy : 0.7527
##
##        'Positive' Class : FALSE
##
```

14. Training Model with SVM K-fold cross validation

```
svmmodel.kfold<- train(expensive ~ X+age+bmi+children+smoker+location+location_type+education
_level+yearly_physical+exercise+married+hypertension+gender,data=trainSet , method= "svmRadia
l", trControl=trainControl(method = "repeatedcv",number=10), preProcess=c("center","scale"))
```

15. Predicting on k-fold model

```
svmPredkfold<-predict(svmmodel.kfold,testSet,type='raw')
```

16. Creating Confusion Matrix for k-fold.

```
confusionMatrix(svmPredkfold,testSet$expensive)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1662  275
##      TRUE     44  293
##
##                Accuracy : 0.8597
##                  95% CI : (0.8448, 0.8737)
##     No Information Rate : 0.7502
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.567
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9742
##             Specificity : 0.5158
##          Pos Pred Value : 0.8580
##          Neg Pred Value : 0.8694
##              Prevalence : 0.7502
##          Detection Rate : 0.7309
##    Detection Prevalence : 0.8518
##       Balanced Accuracy : 0.7450
##
##        'Positive' Class : FALSE
##
```

17. Training rpart model

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.2.2
```

```
modelrpart<-train(expensive ~ X+age+bmi+children+smoker+location+location_type+education_leve
l+yearly_physical+exercise+married+hypertension+gender, method = "rpart",data = trainSet)
```

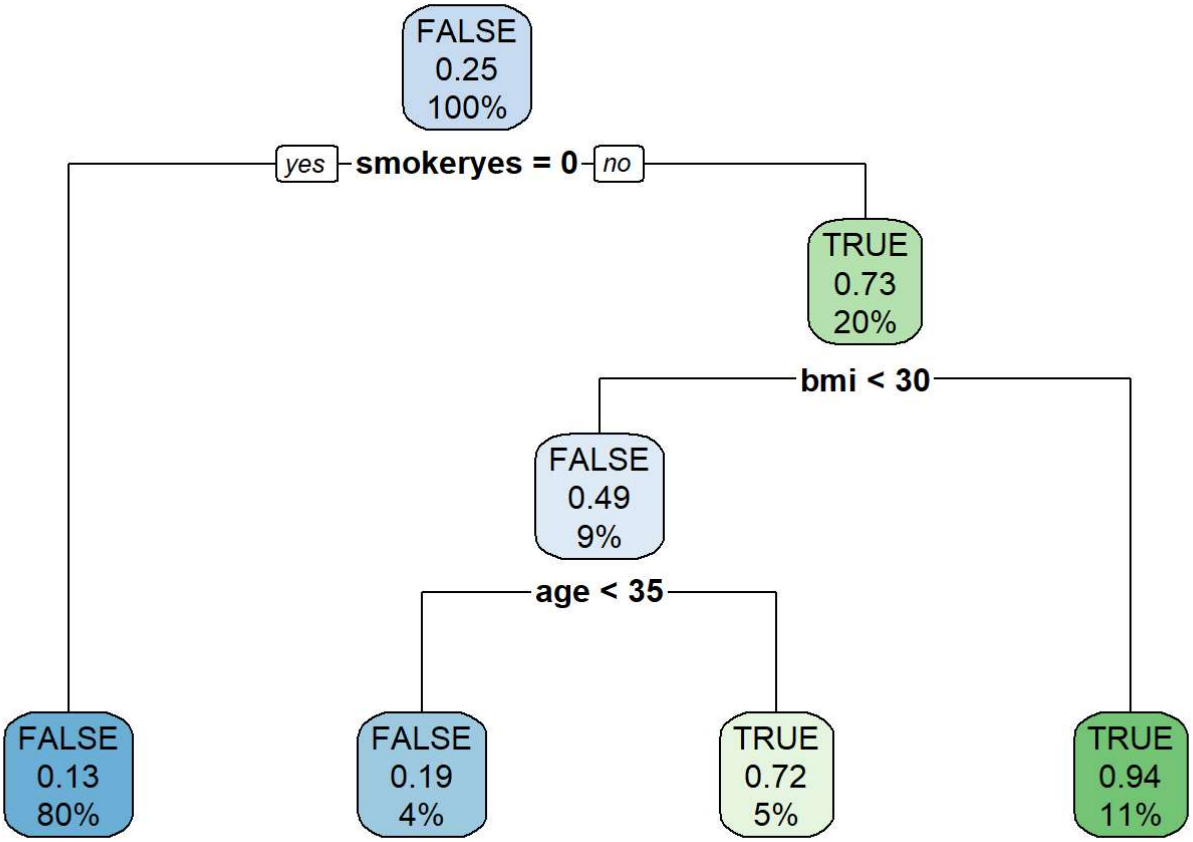
18. Plotting Rpart Plot final model

```
library(rpart.plot)
```

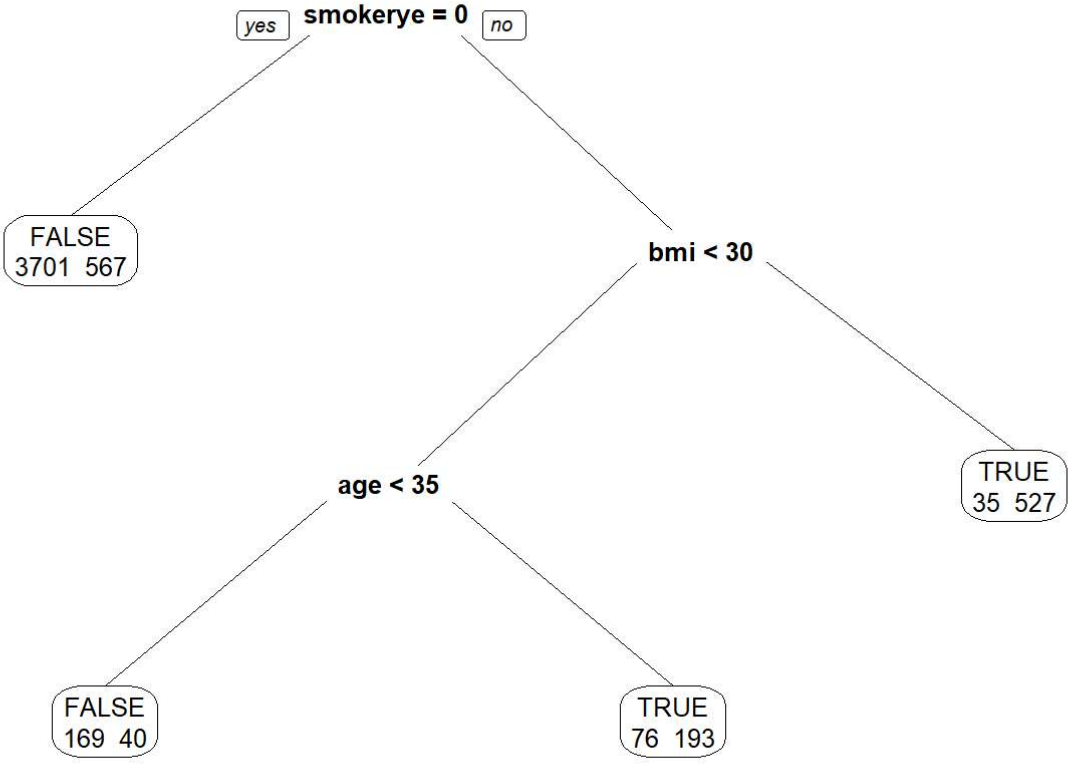
```
## Warning: package 'rpart.plot' was built under R version 4.2.2
```

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

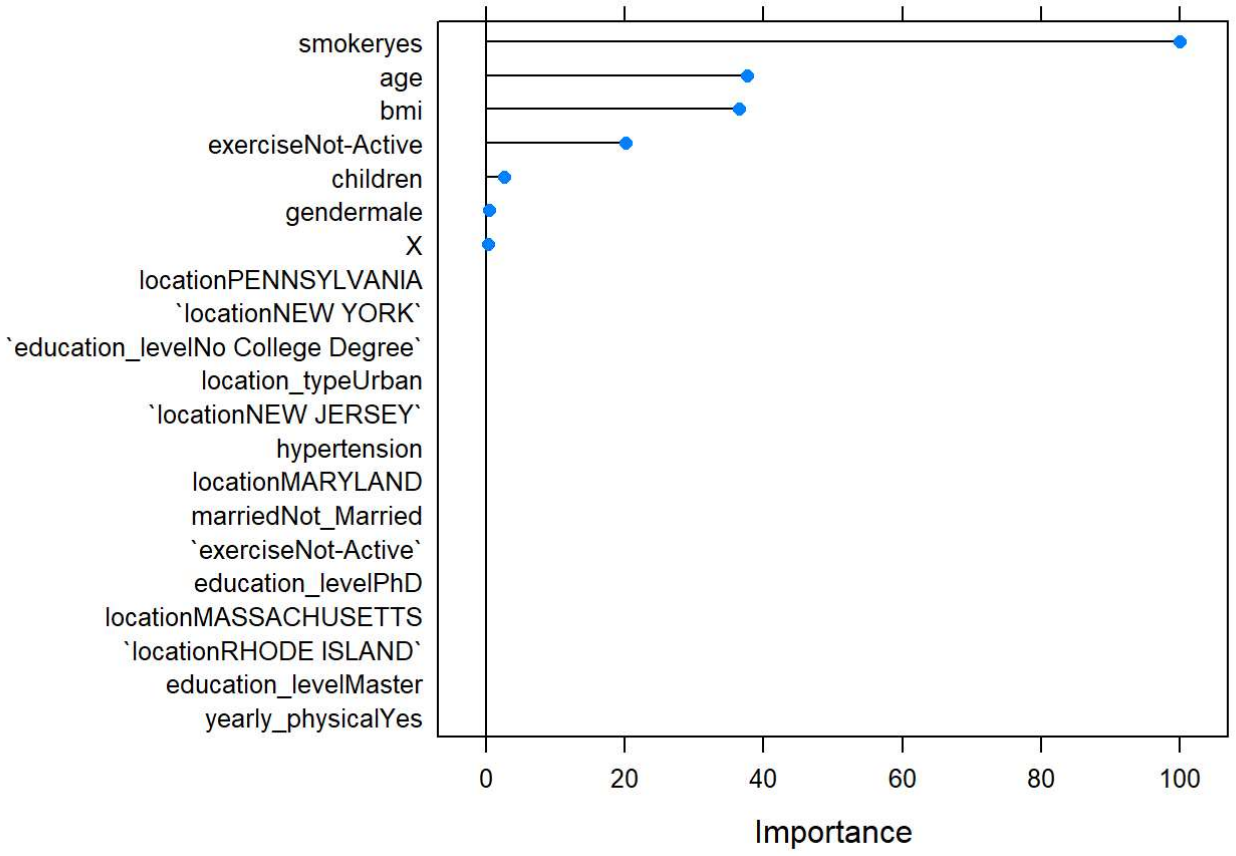
```
rpart.plot(modelrpart$finalModel)
```




```
prp(modelrpart$finalModel, faclen = 0, cex = 0.8, extra = 1)
```



```
plot(varImp(modelrpart))
```




19. Creating Confusion Matrix To calculate accuracy

```
rpartpred<-predict(modelrpart,testSet,type='raw')
confusionMatrix(rpartpred,testSet$expensive)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1661  265
##      TRUE     45  303
##
##                Accuracy : 0.8637
##                  95% CI : (0.8489, 0.8775)
##     No Information Rate : 0.7502
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5823
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9736
##             Specificity : 0.5335
##          Pos Pred Value : 0.8624
##          Neg Pred Value : 0.8707
##              Prevalence : 0.7502
##          Detection Rate : 0.7304
##    Detection Prevalence : 0.8470
##       Balanced Accuracy : 0.7535
##
##        'Positive' Class : FALSE
##
```

20. Loading best model into rda file to save our model

```
our_model <- modelrpart
save(our_model, file = "our_model.rda")
```