IST387_FinalProj

Nicholas Santangelo

2022-12-05

```r
#1
library(tidyverse) #library calls the tideyverse package
```

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

```r
costDF <- read_csv('https://intro-datascience.s3.us-east-2.amazonaws.com/HMO_data.csv')
```

```
## Rows: 7582 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): smoker, location, location_type, education_level, yearly_physical, ...
## dbl (6): X, age, bmi, children, hypertension, cost
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
##read_csv function takes data from the linked file. Placed into costDF variable.

str(costDF) #str function gets the overall structure of the costDF df
```

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

```
##  $ gender         : chr [1:7582] "female" "male" "male" "male" ...
##  $ cost           : num [1:7582] 1746 602 576 5562 836 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   X = col_double(),
##   ..   age = col_double(),
##   ..   bmi = col_double(),
##   ..   children = col_double(),
##   ..   smoker = col_character(),
##   ..   location = col_character(),
##   ..   location_type = col_character(),
##   ..   education_level = col_character(),
##   ..   yearly_physical = col_character(),
##   ..   exercise = col_character(),
##   ..   married = col_character(),
##   ..   hypertension = col_double(),
##   ..   gender = col_character(),
##   ..   cost = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
#The df has 14 variables with 7,582 observations. All columns are either a number
#or character data type. DF appears to represent personal health care costs

summary(costDF) #summary function produces descriptive stats for costDF columns
```

```
##        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
##
```

```
#cost df appears to have: an age range of 18-66, a child range of 0-5, and an
#overall cost range of $2-$55,715
```

```
#2
names(which(colSums(is.na(costDF))>0)) #is.na checks for null values in the
```

```
## [1] "bmi"          "hypertension"
```

```
#specified costDF Dataframe. colSums gets the sum count of the null values, if
#greater than 0. Which gets the column number of columns with null values. Names
#function produces the columns name's

library(imputeTS) #library calls the imputeTS package
```

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

```
typeof(costDF$bmi) #typeof function gets the data type of the bmi column
```

```
## [1] "double"
```

```
typeof(costDF$hypertension) #typeof function gets the data type of the
```

```
## [1] "double"
```

```
#hypertension column

costDF$bmi <- na_interpolation(costDF$bmi) #redefines bmi column using the
#na_interpolation function from the imputeTS package. Gets calculated replacement
#values for null rows. bmi column defined as data to be used.
costDF$hypertension <- na_interpolation(costDF$hypertension) #redefines bmi
#column using the na_interpolation function from the imputeTS package. Gets
#calculated replacement values for null rows. bmi column defined as data to be
#used.

sum(is.na(costDF$bmi)) #sum function used to callculated the total of null
```

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

```
#values found using the is.na function. bmi column defined as data to be
#checked
sum(is.na(costDF$hypertension)) #sum function used to callculated the total of
```

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

```r
#null values found using the is.na function. hypertension column defined as data
#to be checked
```

```r
#3
excercise_table <- table(costDF$exercise) #table function used to generate
#breakdown of column values. exercise defined as column to be used. put into
#variable excercise_table
excercise_table #displays the table
```

```
##
##     Active Not-Active
##       1888       5694
```

```r
#around one-third of people are active and two-thirds are not.

locationT_table <- table(costDF$location_type) #table function used to generate
#breakdown of column values. location_type defined as column to be used.put into
#variable locationT_table
locationT_table #displays the table
```

```
##
## Country   Urban
##    1903    5679
```

```r
#around 25% of the people live in the country and 75% live in an urban area

education_table <- table(costDF$education_level) #table function used to generate
#breakdown of column values. education_level defined as column to be used.put
#into variable education_table
education_table #displays the table
```

```
##
##          Bachelor            Master No College Degree               PhD
##              4578              1533               759               712
```

```r
#the majority hold a bachelor, then masters is second most common, followed by
#no degree, then PhD.
```

```r
#4
summary(costDF$cost)[5] #summary function displays the quantitative stats on the
```

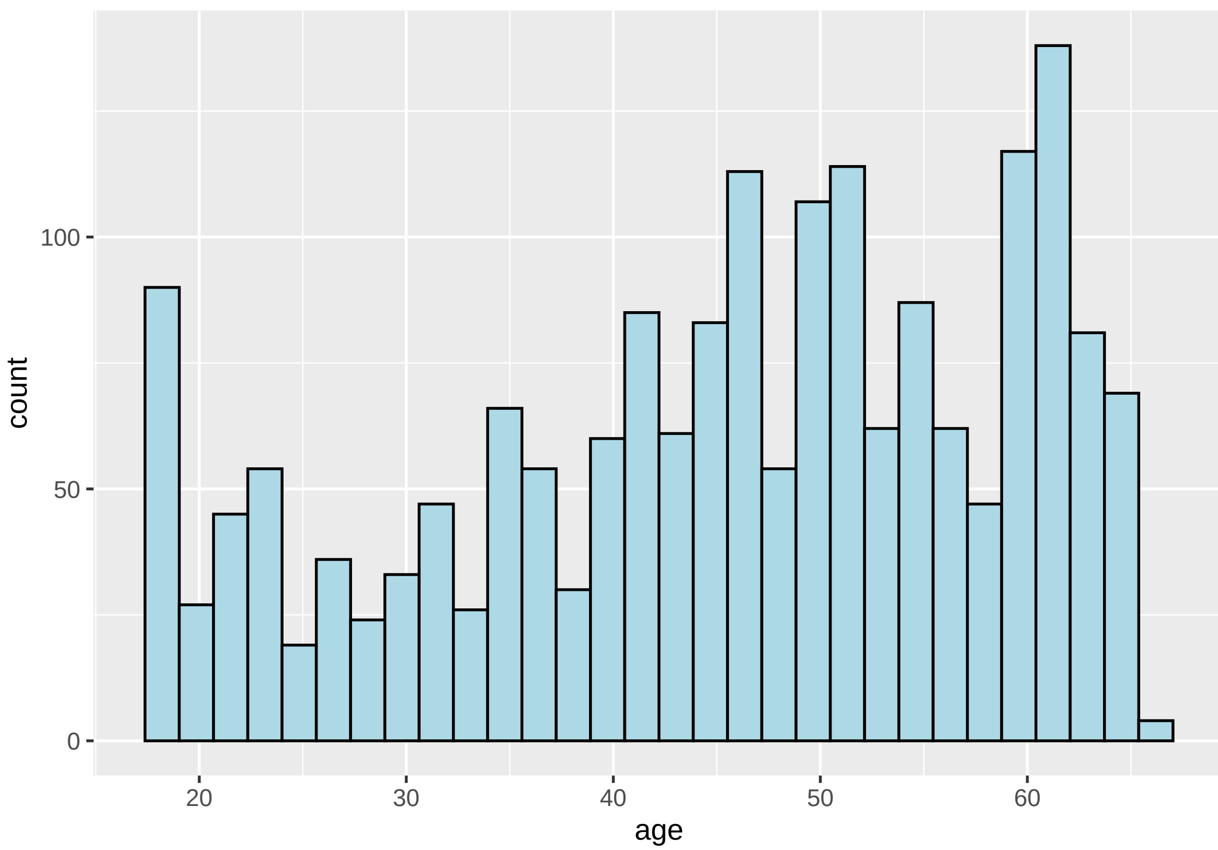
```
## 3rd Qu.
##    4775
```

```r
#defined cost column data. Subset used to get just the third quartile number

costDF$expensive <- ifelse(costDF$cost > summary(costDF$cost)[5],'Expensive',
                           'Inexpensive')
#expensive column defined. ifelse used to create a boolean expression. if cost
#is bigger than the 75% value than it would be 'Expensive', else it would be
```

```r
#'Inexpensive'

#I decided that the 3rd quartile or the 75% mark would be the threshold because
#the mean is significantly bigger than the median, meaning there are a large
#number of values bigger than the median. With that said, I felt like the top
#25% of values would be a good mark for the expensive threshold.
```

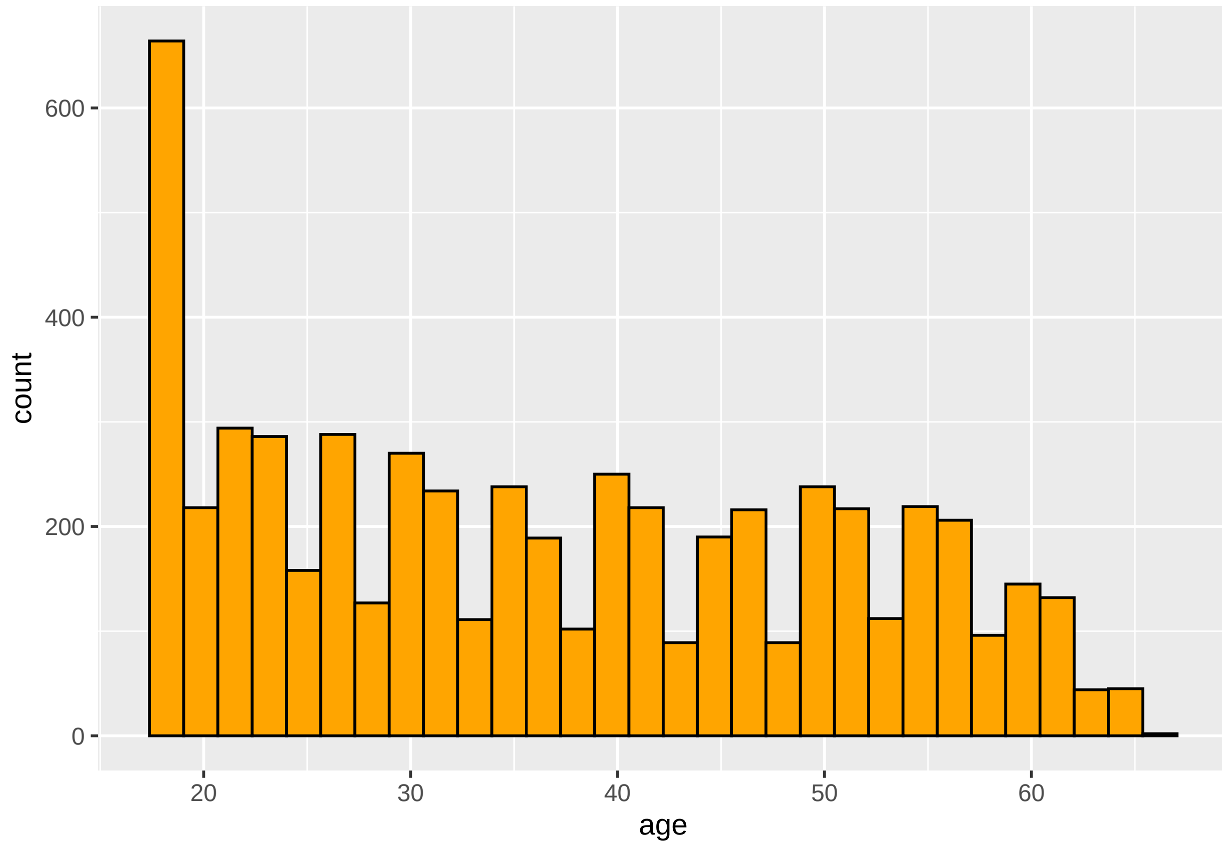
```r
#5
exp_df <- subset(costDF, expensive == 'Expensive') #exp_df defined as a dataframe.
#subset function used to take a portion of the defined costDF data. Expensive
#column used to subset the data, only selects rows where 'Expensive' is the
#expensive column's value
inexp_df <- subset(costDF, expensive == 'Inexpensive') #inexp_df defined as a
#dataframe.subset function used to take a portion of the defined costDF data.
#Expensive column used to subset the data, only selects rows where 'Inexpensive'
#is the expensive column's value

ggplot(exp_df, aes(x=age)) +geom_histogram(fill = "light Blue", color = 'black')
```

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




```r
#ggplot used to map the exp_df defined data. aesthetics defined as x axis value
#equals age. geom_histogram used to identify the graph type. "light Blue" defined
#as the fill color and 'black' defined as the outline color.
ggplot(inexp_df, aes(x=age)) +geom_histogram(fill = "orange", color = 'black')
```

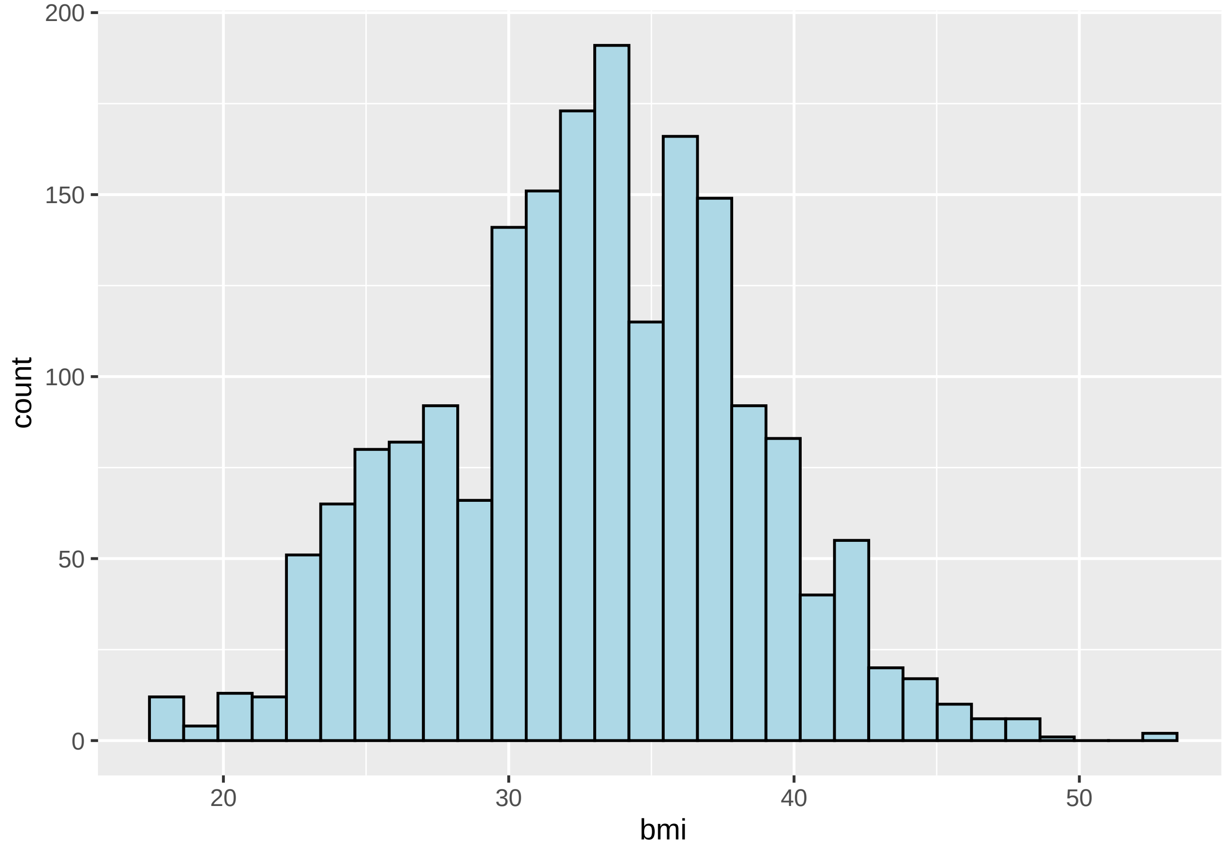
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#ggplot used to map the inexp_df defined data. aesthetics defined as x axis value
#equals age. geom_histogram used to identify the graph type. "orange" defined
#as the fill color and 'black' defined as the outline color

#expensive graph appears to be right leaning, indicating that those that are
#expensive are likely to be older. Inexpensive graph is left leaning indicating
#that younger people are likely to be cheaper.
```

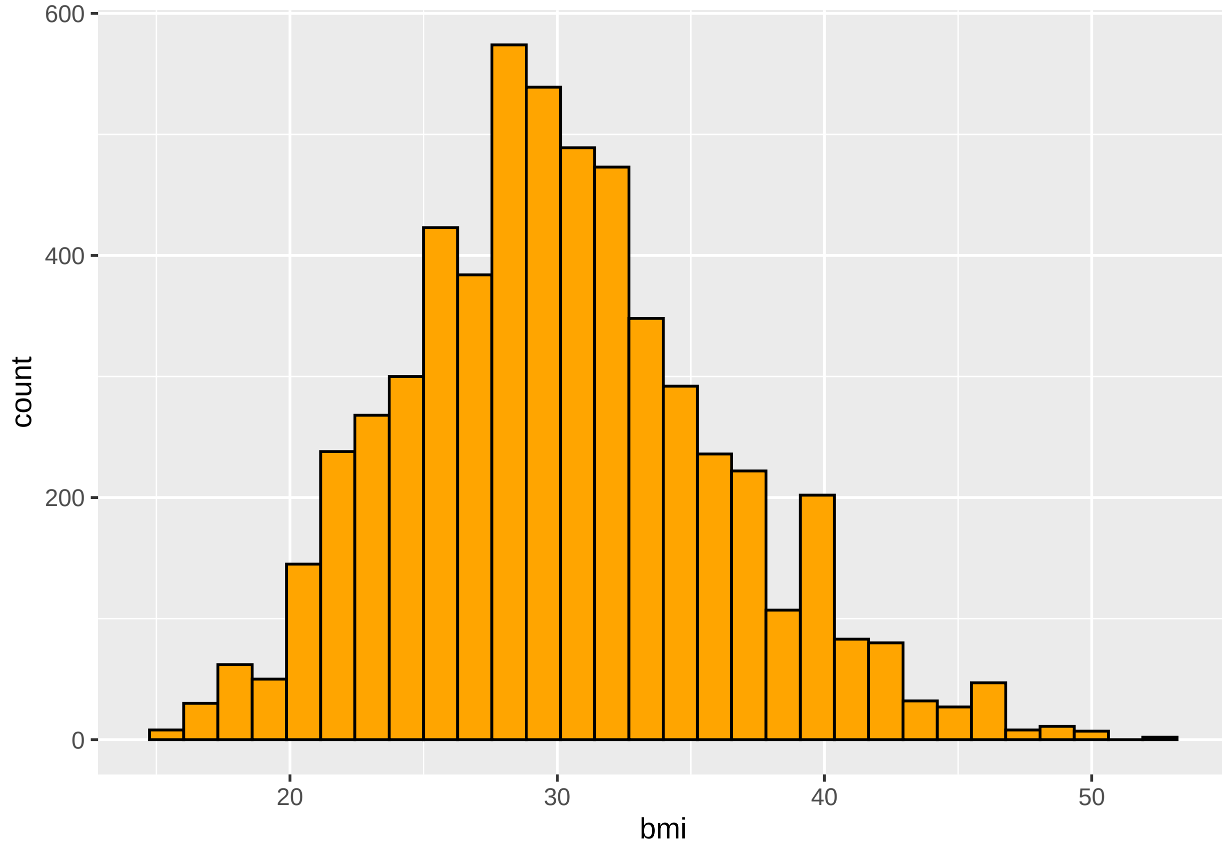
```
#9 continued

ggplot(exp_df, aes(x=bmi)) +geom_histogram(fill = "light Blue", color = 'black')
```

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




```r
#ggplot used to map the exp_df defined data. aesthetics defined as x axis value
#equals bmi geom_histogram used to identify the graph type. "light Blue" defined
#as the fill color and 'black' defined as the outline color.
ggplot(inexp_df, aes(x=bmi)) +geom_histogram(fill = "orange", color = 'black')
```

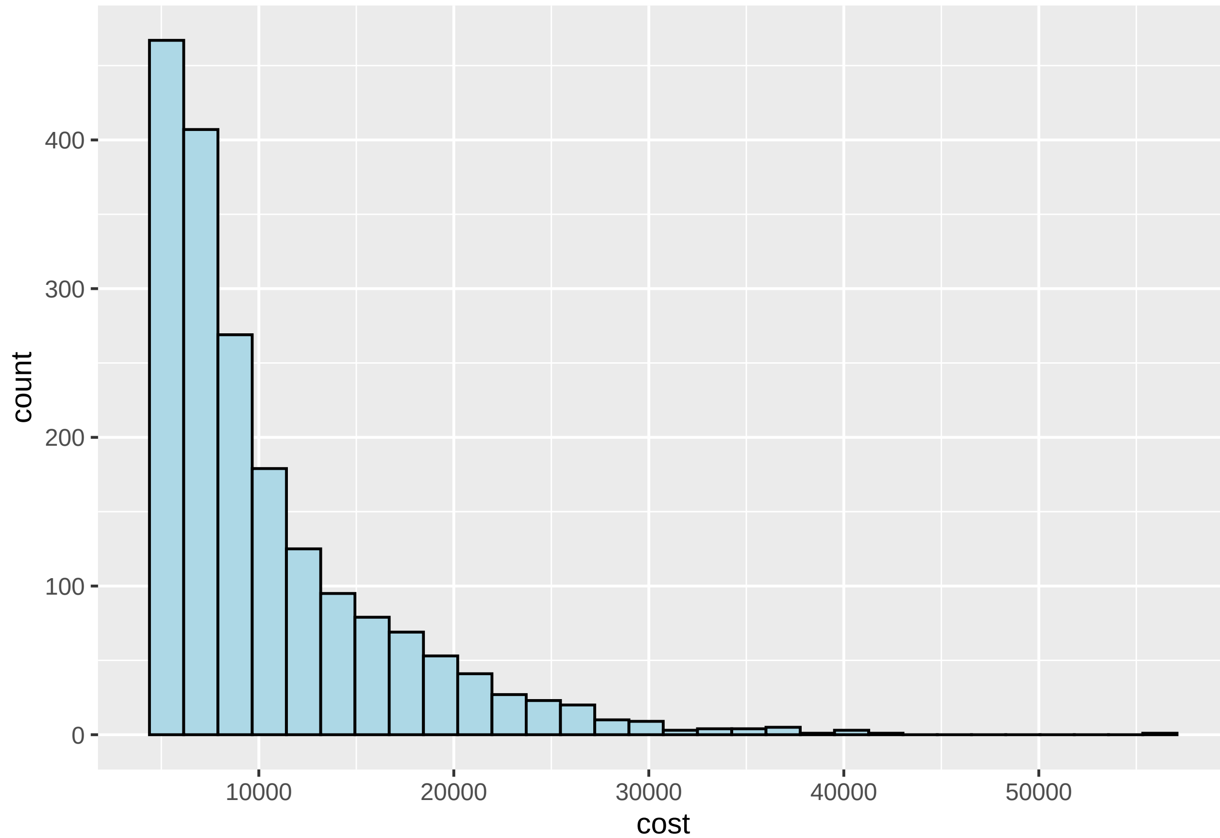
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```




```r
#ggplot used to map the inexp_df defined data. aesthetics defined as x axis value
#equals bmi geom_histogram used to identify the graph type. "orange" defined
#as the fill color and 'black' defined as the outline color.

#while both graphs do show a realatively similar standard deviation, the peak
#count of the expensive graph is on a higher bmi (~35) than that of the
#inexpensive graph (~27)
```

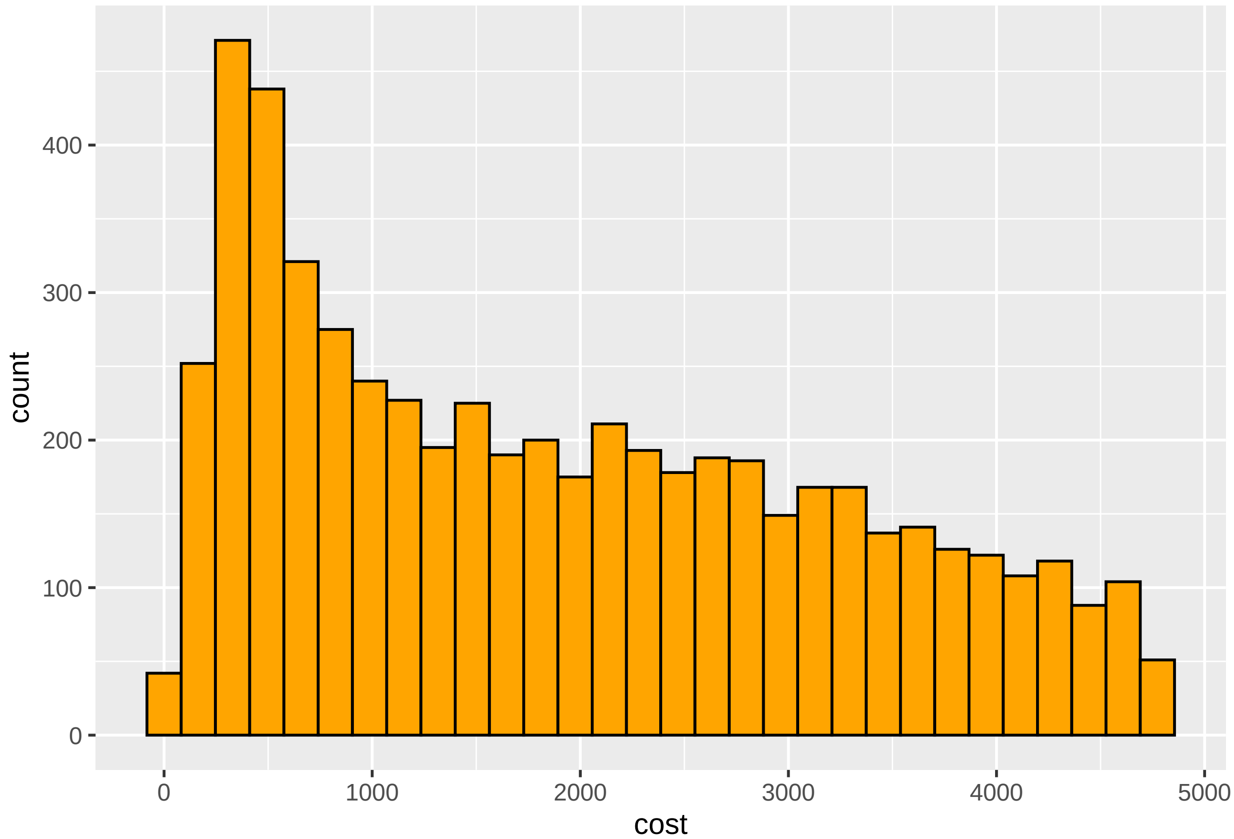
```r
#9 continued

ggplot(exp_df, aes(x=cost))+geom_histogram(fill = "light Blue", color = 'black')
```

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




```
#ggplot used to map the exp_df defined data. aesthetics defined as x axis value
#equals cost geom_histogram used to identify the graph type. "light Blue" defined
#as the fill color and 'black' defined as the outline color.
ggplot(inexp_df, aes(x=cost))+geom_histogram(fill = "orange", color = 'black')
```

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




```
#ggplot used to map the inexp_df defined data. aesthetics defined as x axis value
#equals cost geom_histogram used to identify the graph type. "orange" defined
#as the fill color and 'black' defined as the outline color.


#the expensive histogram shows a heavily left leaning, steep distribution of cost
#where as the inexpensive is also left leaning, just much more evenly distributed.
#This distribution implies that there are a lot of people of the 3rd quartile
#threshold that I decided, meaning I may have set the expensive boundary a
#little too early, maybe 80-90% would have been better.
```

```
#6
state <- data.frame(
  cost = aggregate(costDF$cost, list(costDF$location), mean))
#state defined as a variable name, data.frame used to create this as a dataframe.
#cost identified as column name, uses aggregate function to get mean cost of
#cost column based on the different location values.

colnames(state)[1] <- "name"
#renames location values column as "name"
colnames(state)[2] <- "ave_cost"
#renames average cost values column as "ave_cost"
```

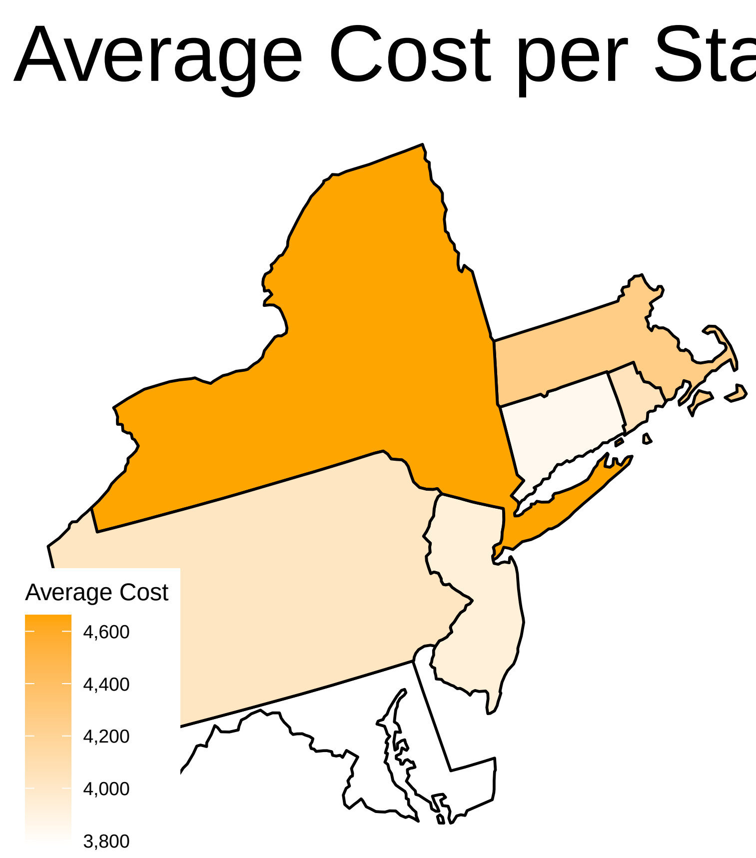
```
#7
state[which.max(state$ave_cost),]
```

```
##       name ave_cost
## 5 NEW YORK 4661.506
```

```r
#state df subseted to output a singular row. which.max function used to
#determine which row has the highest value. ave_cost defined as the data used.
```

```r
#8
#install.packages('usmap')
#install.packages packages function used to install the usmap package
library(usmap)
#library function used to call and load in the usmap package
library(ggplot2)
#library function used to call and load in the ggplot2 package

state$state <- c("CT", "MD", "MA", "NJ", "NY", "PA", "RI")
#state column defined for state df. Abreviations of state names inputed as data
#so states can be mapped


plot_usmap(data= state,
           values = "ave_cost",
           include= c("CT", "MD", "NJ", "NY", "PA", "RI", "MA")) +
           scale_fill_continuous(low= 'white', high = 'orange', name= 'Average Cost', label = scales::co
           ggtitle("Average Cost per State") + theme(plot.title = element_text(size=30))
```

```
## Warning: Ignoring unknown parameters: linewidth
```



```r
#plot_usmap funcction used to plot the map of the US. state df defined as the
#data to be used for color gradient. ave_cost defined as column to base gradient
#on. unclude parameter used to indicate which states I want to appear in the map.
#scale_fill_continuous added to define the mapping colors as well as the title
#to the legend. ggtitle added to create a overall graph title.

#based on the color gradient, New York is the most expensive. Then Massachusetts
#is the second most expensive. After that, Rhode Island and Pennsylvania are
#about equal for third place. Then, New Jersey and Connecticut are about equal
#for fourth place. Lastly is Maryland which is the cheapest.
```

```r
#9
costCat <- data.frame(location=as.factor(costDF$location),
                      location_type=as.factor(costDF$location_type),
                      expensive =as.factor(costDF$expensive),
                      education_level =as.factor(costDF$education_level),
                      yearly_physical =as.factor(costDF$yearly_physical),
                      gender = as.factor(costDF$gender),
                      hypertension = as.factor(costDF$hypertension),
                      exercise = as.factor(costDF$exercise),
                      age= costDF$age,
                      bmi= costDF$bmi,
                      x= costDF$X,
                      married = as.factor(costDF$married),
                      smoker= as.factor(costDF$smoker))


#costCat variable defined. data.frame function used to create a dataframe. Almosts
#all variables from costDF function inputted. Those that have binary-type or
#select number of possible responses, set as a factor type. Those that have a
#wide range of values (age, bmi, and x) simply converted over in their original
#form. Cost column was excluded as we are trying to find what variables
#determine the cost.

library(arules) #library calls and loads in the arules package
```

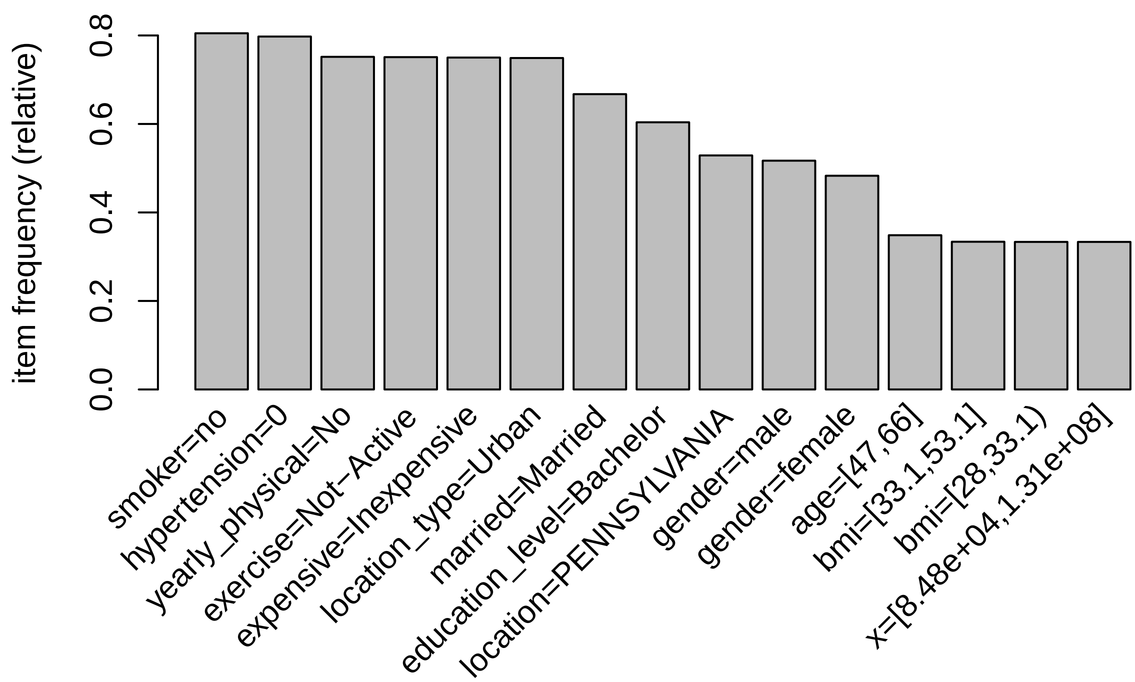
```
## Loading required package: Matrix

##
## 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
```

```
library(arulesViz) #library calls and loads in the arulesViz package

costCat_1 <- as(costCat, 'transactions')
```

```
## Warning: Column(s) 9, 10, 11 not logical or factor. Applying default
## discretization (see '? discretizeDF').
```

```
#costCat_1 defined as the variable name. as function used to convert costCat
#dataframe to a transaction class data type. We do this to make the data able
#for further mining and rule identification.

itemFrequencyPlot(costCat_1, topN=15)
```




```
#itemFrequencyPlot function used to get the most common single column value
#occurences. costCat_1 defined as the data to be used.topN paramater function
#used to only select the tope 15 most common values.

#looks like 80% of the people are not smokers, and roughly 80% of people don't
#have hyper tension. And arguably because of this, around 75% of people are
#inexpensive.
```

```r
#10
cost_analysis <- apriori(costCat_1,
                         parameter=list(supp=0.1, conf=0.7),
                         control=list(verbose=F),
                         appearance=list(default="lhs",rhs= 'expensive=Expensive'))

#cost_analysis defined as the variable name. apriori function used to generate
#association rules based on the costCat_1 value occurence data from the previous
#question. parameter attribute used to designate required support level of 10%
#and confidence level of 70%, this was to narrow down strongest correlations.
#defines control as a list with the verbose flow type as F. appearance is
#defined as a list with the default labeled as lhs. the rhs variable is
#identified as expensive=Expensive

inspect(cost_analysis)
```

```
##     lhs                          rhs                     support   confidence  coverage  lift     count
## [1] {smoker=yes}              => {expensive=Expensive} 0.1424426  0.7302231 0.1950673 2.921663  1080
## [2] {location_type=Urban,
##      smoker=yes}              => {expensive=Expensive} 0.1067001  0.7262118 0.1469269 2.905614   809
## [3] {exercise=Not-Active,
##      smoker=yes}              => {expensive=Expensive} 0.1168557  0.8158379 0.1432340 3.264213   886
## [4] {yearly_physical=No,
##      smoker=yes}              => {expensive=Expensive} 0.1065682  0.7169476 0.1486415 2.868547   808
## [5] {hypertension=0,
##      smoker=yes}              => {expensive=Expensive} 0.1109206  0.7237522 0.1532577 2.895772   841
```

```r
#inspect function displays the results of cost_analysis showing the
#frequency and association of various attribute combinations.


#not surprising, the most common value association with expensive people is
#smoking. With a confidence of 73%, it means that nearly 3 out of 4 expensive
#people smoke. The second greatest association was smokers living in an urban
#environment with a confidence slightley lower than just smokers alone. In third
#was smokers who did not exercise, which had the highest confidence of 81.5%.
#This means you are more likely to be expensive if you do both than if you just
#smoked. Because this pairing isn't as common (support) it only ranks third on
#highest association.

#support: The percentage this pairing occurs in the whole datatset (out of
#expensive people)

#confidence: The percentage that if you have the lhs charectoristics then you
#wil have the rhs.

#lift: Shows the effectiveness of the model, meaning it is the ratio between the
#confidence of the rule and the expected confidence. Higher lift indicates a
#higher correlation.
```

```r
#11
library(caret) #library calls and loads in the caret package
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

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

```
cost_train <- createDataPartition(y=costDF$expensive,p=.70,list=FALSE)
#cost_train assigned as variable name. createDataPartition function used to get
#a sample of the data. expensive column defined as the data to be sampled. P set
#at .7 to sample 70% of the data. list attribute set as false.
train_set <- costDF[cost_train,]
#train_set defined of subset of costDF dataframe to only include rows from the
#cost_train data partition
test_set <- costDF[-cost_train,]
#test_set defined of subset of costDF dataframe to exclude all rows from the
#cost_train data partition

dim(test_set)[1] + dim(train_set)[1]
```

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

```
#calculates the total observations between the test_set and train_set. dim
#used to get the dimensions, subset used to isolate the observations
dim(costDF)[1]
```

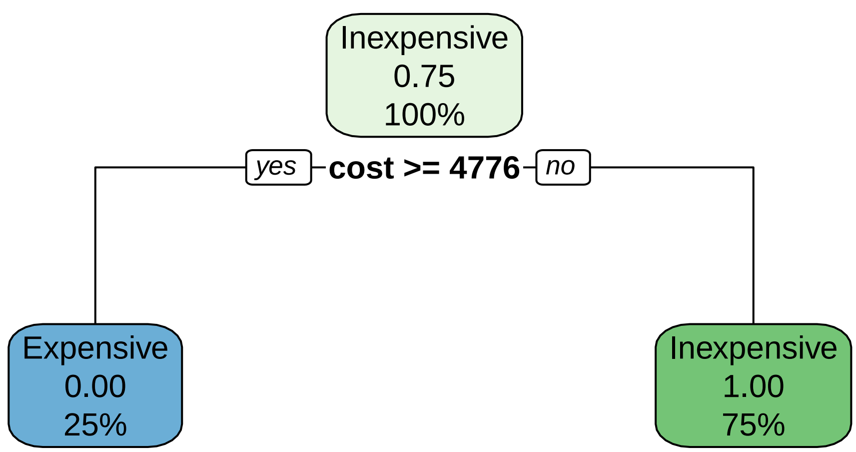
```
## [1] 7582
```

```
#dim function used on the costDF dataframe to check if the number of
#observations match. subset used to isolate the observations
```

```
#12
library(kernlab) #library calls and loads in the kernlab package
```

```
##
## 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
```

```r
library(rpart) #library calls and loads in the rpart package
library(rpart.plot) #library calls and loads in the rpart.plot package
library(caret) #library calls and loads in the caret package

model <- ksvm(as.factor(expensive)~ .,data=train_set, C =5, cross =3,
              prob.model    = TRUE)
#model defined as variable name. ksvm function called from kernlab package to
#create a support vector. Defines expensive as dependant variable, defines
#train_set as data, defines count as 5, defines cross as 3, defines prob.model
#as true to builds a model for calculating probabilities

cost_tree <- rpart(expensive~ .,data=train_set)
#cost_tree assigned as variable name. uses the rpart function to develop a data
#map of the expensive column in the train_set df
rpart.plot(cost_tree)
```




```r
#rpart.plot function used to plot the cost_tree rpart of the expensive variable
```

```r
#12 Continued
cost_predict <- predict(model, test_set, type = 'response') #uses predict
#function to test test data, defines type as response

confusion_table <- table(cost_predict, test_set$expensive)
#Creates table of prediction's data, identifies cost_predict and expensive as
#the data. The 4 numbers of the table represent the count of expensive and
```

```r
#inexpensive of the model. The diagnosis of the table are the corespondent and
#prediction of the model.
confusion_table
```

```
##
## cost_predict  Expensive Inexpensive
##   Expensive         550          14
##   Inexpensive        18        1692
```

```r
#displays the table confusion_table

error <- (sum(confusion_table) - sum(diag(confusion_table))) / sum(confusion_table)
#Gets the sum of table subtracted by the sum of the diaganal, devided by the
#total sum to get the error rate
error
```

```
## [1] 0.01407212
```

```r
#Displays the error rate
```

```r
#12 Continued
confusion_matrix <- confusionMatrix(cost_predict, as.factor(test_set$expensive))
#confusionMatrix function gets the calculation of observed and predicted classes.
#cost_predict defined as the prediction used, expensive defined as the column to
#be examined.
confusion_matrix
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Expensive Inexpensive
##   Expensive         550          14
##   Inexpensive        18        1692
##
##                Accuracy : 0.9859
##                  95% CI : (0.9802, 0.9904)
##     No Information Rate : 0.7502
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9624
##
##  Mcnemar's Test P-Value : 0.5959
##
##             Sensitivity : 0.9683
##             Specificity : 0.9918
##          Pos Pred Value : 0.9752
##          Neg Pred Value : 0.9895
##              Prevalence : 0.2498
##          Detection Rate : 0.2419
##    Detection Prevalence : 0.2480
##       Balanced Accuracy : 0.9801
##
```

```
##        'Positive' Class : Expensive
##
```

```
#displays the confusion matrix
```

```
#13
#Based on the nearly identical error rates of 1.18% and 1.19%, It would be
#incorrect of me to chose one model over the other. With their incredible
#accuracy, both of these models are more than adequate.
```

```
#14
#Hello CEO,

#after a lot of examination and analysis, there are a few takeaways regarding
#what variables effect a persons insurance cost the most. First and most
#importantly, smoking is something that has a massive impact on a person's cost.
#With a confidence percentage of 73%, smoking is the single greatest contributor
#to expensive cost. Beyond smoking, factors like lack of exercise and living in
#an urban environment can also play a pretty important role. Beyond factors like
#these, we've discovered that the a person's age can be an indication of their
#cost, with the majority of our expensive clients over the age of 45. As far as
#the expectations for next year go, I would expect your out-of-shape elderly
#smokers to be your most expensive clients. But, out-of-shape smokers of all ages
#will tend to make up the majority of your expensive clients. To do risk
#assessment, I suggest isolating the smokers and then examine them based on their
#age and bmi. Through this you should find your expensive clientele for the year
#to come.

#As far as how to lower your total health care costs. I think making some sort
#of incentivisation program for the office smokers would yield the biggest total
#cost cut. Getting even a percentage of the smoking population to quit could
#reduce the total cost enough the make the incentives financially feasible.

#Feel free to reach out with any questions!

#thanks,
#Nick
```