Project

2022-11-17

```
#Data Science project by
#Jenil Sheth : 7754395981
#Shruti More : 528873433
#Vedant Patil : 235168671
#Collin Taylor : 563919675
```

```
library(readr)
library(imputeTS)
```

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

```
library(ggplot2)
library(kernlab)
```

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

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

```
library(caret)
```

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v purrr   0.3.4      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x kernlab::alpha() masks ggplot2::alpha()
## x purrr::cross()   masks kernlab::cross()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
```

```
library(dplyr)
library(rio)
library(rpart)
library(rpart.plot)
library(e1071)
library(arules)
```

```
## 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 object is masked from 'package:kernlab':
##
##     size
##
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(arulesViz)
library(rsample)
```

```
##
## Attaching package: 'rsample'
##
## The following object is masked from 'package:e1071':
##
##     permutations
```

```
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)
```

```
## Loading required package: maps
##
## Attaching package: 'maps'
##
```

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

```
# library(sf)
# library(units)
library(shiny)
library(shinyWidgets)
```

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

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

```
proj_df <- data.frame(data)
str(proj_df)
```

```
## '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 ...
```

```
summary(proj_df)
```

```
##        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
##
```

```
any(is.na(proj_df$X))
```

```
## [1] FALSE
```

```
any(is.na(proj_df$age))
```

```
## [1] FALSE
```

```
any(is.na(proj_df$bmi))
```

```
## [1] TRUE
```

```
proj_df$bmi <- na.interpolation(proj_df$bmi, option = "linear")
```

```
## Warning: na.interpolation will be replaced by na_interpolation.
##            Functionality stays the same.
##            The new function name better fits modern R code style guidelines.
##            Please adjust your code accordingly.
```

```
any(is.na(proj_df$bmi))
```

```
## [1] FALSE
```

```
any(is.na(proj_df$children))
```

```
## [1] FALSE
```

```
any(is.na(proj_df$smoker))
```

```
## [1] FALSE
```

```
any(is.na(proj_df$location))
```

```
## [1] FALSE
```

```
any(is.na(proj_df$location_type))
```

```
## [1] FALSE
```

```
any(is.na(proj_df$hypertension))
```

```
## [1] TRUE
```

```
proj_df$hypertension <- na.interpolation(proj_df$hypertension)
```

```
## Warning: na.interpolation will be replaced by na_interpolation.
##            Functionality stays the same.
##            The new function name better fits modern R code style guidelines.
##            Please adjust your code accordingly.
```

```
any(is.na(proj_df$hypertension))
```

```
## [1] FALSE
```

```
any(is.na(proj_df$cost))
```

```
## [1] FALSE
```

```
proj_df$state_name <- proj_df$location
quantile(proj_df$cost)
```

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

```
proj_df$expensive <- with(proj_df, ifelse(cost > 4775, "TRUE", "FALSE"))
proj_df$expensive <- proj_df$cost>4775
Expensive <- proj_df %>% group_by(expensive) %>% filter(expensive=="TRUE")
InExpensive <- proj_df %>% group_by(expensive) %>% filter(expensive=="FALSE")
head(proj_df)
```

```
##   X age    bmi children smoker      location location_type education_level
## 1 1  18 27.900        0    yes   CONNECTICUT         Urban        Bachelor
## 2 2  19 33.770        1     no  RHODE ISLAND         Urban        Bachelor
## 3 3  27 33.000        3     no MASSACHUSETTS         Urban          Master
## 4 4  34 22.705        0     no  PENNSYLVANIA       Country          Master
## 5 5  32 28.880        0     no  PENNSYLVANIA       Country             PhD
## 6 7  47 33.440        1     no  PENNSYLVANIA         Urban        Bachelor
##   yearly_physical   exercise married hypertension gender cost    state_name
## 1              No     Active Married            0 female 1746   CONNECTICUT
## 2              No Not-Active Married            0   male  602  RHODE ISLAND
## 3              No     Active Married            0   male  576 MASSACHUSETTS
## 4              No Not-Active Married            1   male 5562  PENNSYLVANIA
## 5              No Not-Active Married            0   male  836  PENNSYLVANIA
## 6              No Not-Active Married            0 female 3842  PENNSYLVANIA
##   expensive
## 1     FALSE
## 2     FALSE
## 3     FALSE
## 4      TRUE
## 5     FALSE
## 6     FALSE
```

```
str(proj_df)
```

```
## 'data.frame':    7582 obs. of  16 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 ...
##  $ state_name     : chr  "CONNECTICUT" "RHODE ISLAND" "MASSACHUSETTS" "PENNSYLVANIA" ...
##  $ expensive      : logi  FALSE FALSE FALSE TRUE FALSE FALSE ...
```

```
mean(Expensive$bmi)
```

```
## [1] 32.83403
```

```
mean(Expensive$age)
```

```
## [1] 45.3847
```

```
mean(Expensive$children)
```

```
## [1] 1.238522
```

```
mean(InExpensive$bmi)
```

```
## [1] 30.11793
```

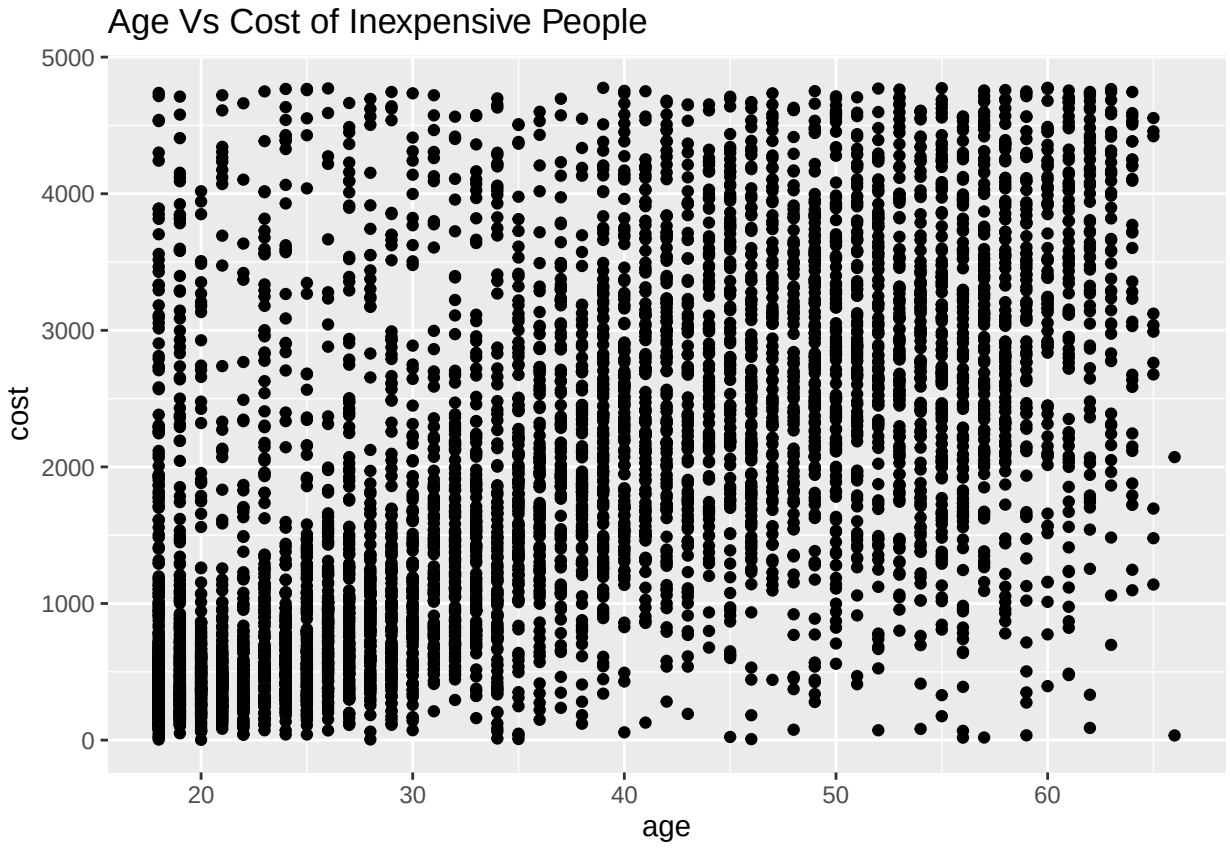
```
mean(InExpensive$age)
```

```
## [1] 36.72006
```

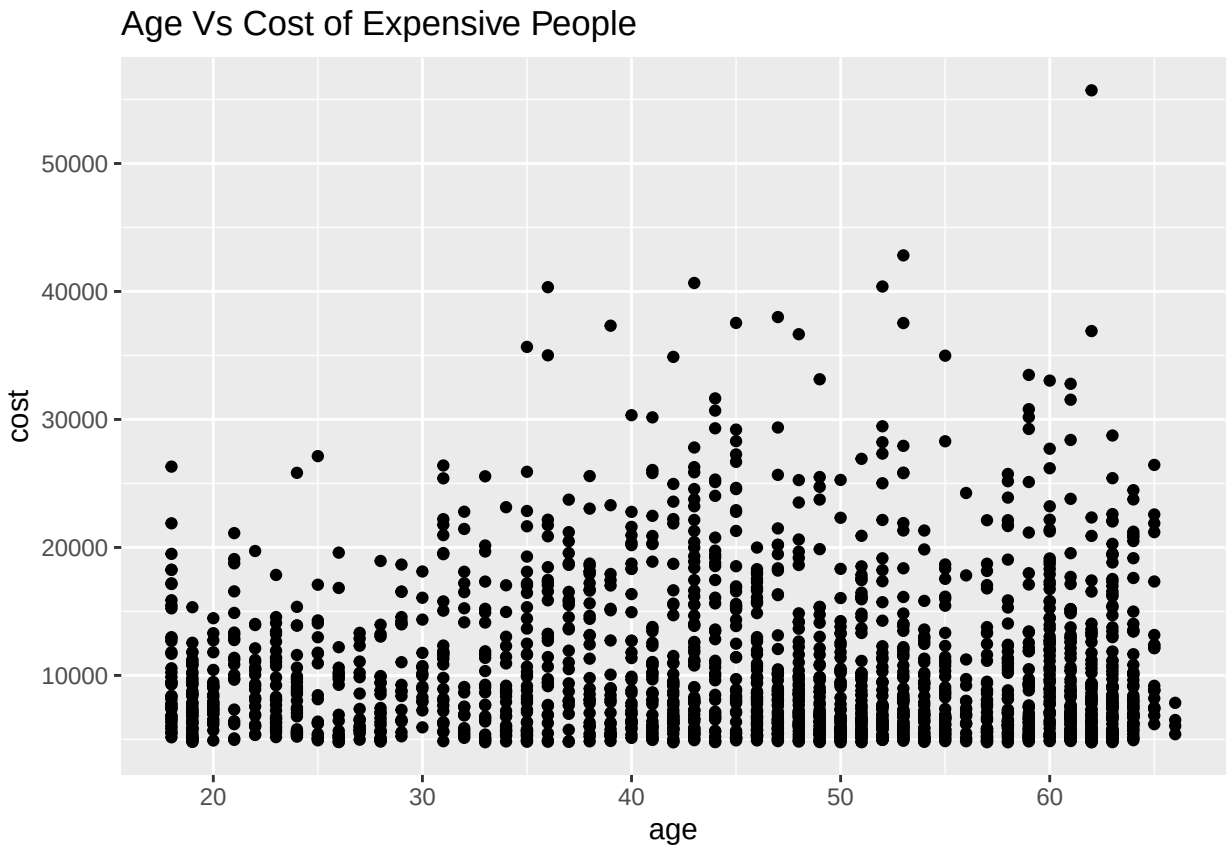
```
mean(InExpensive$children)
```

```
## [1] 1.066467
```

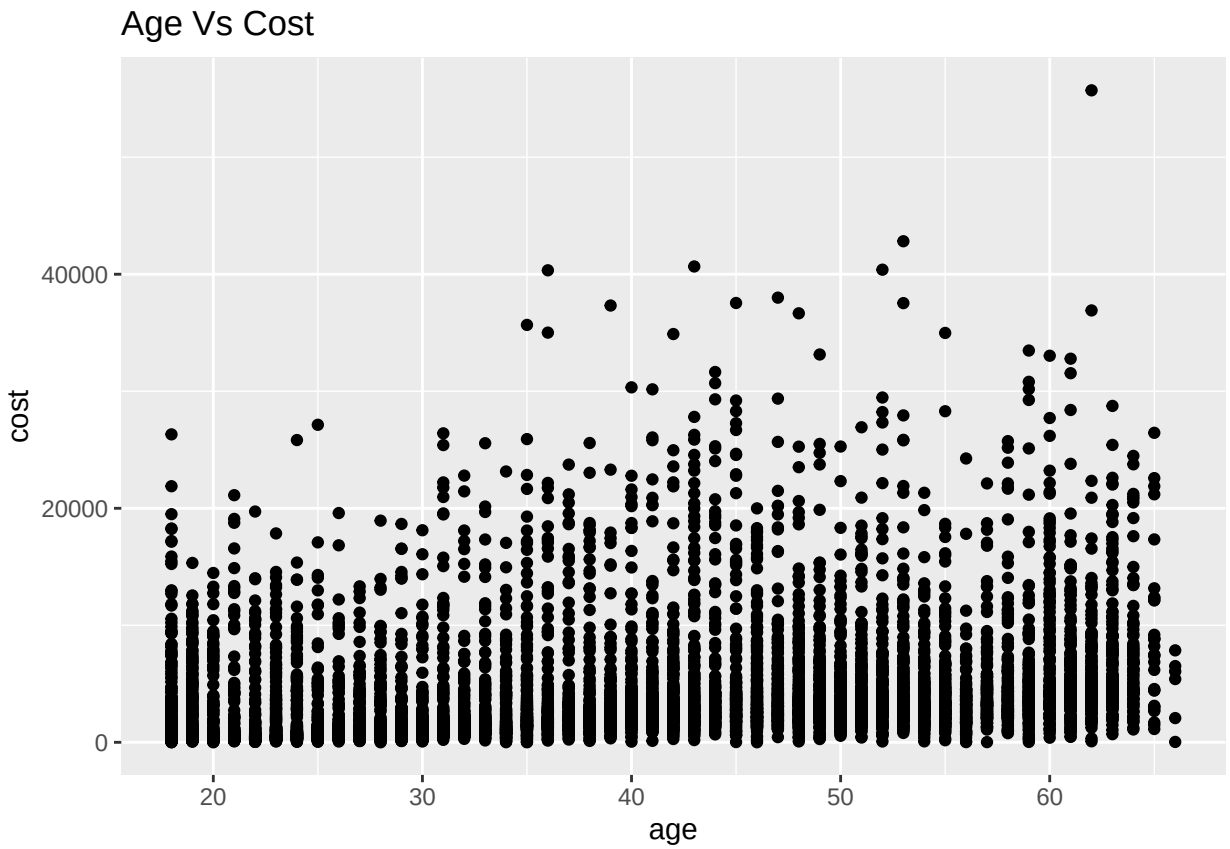
```
ggplot(InExpensive, aes(x=age,y=cost))+geom_point() + ggtitle("Age Vs Cost of Inexpensive People")
```



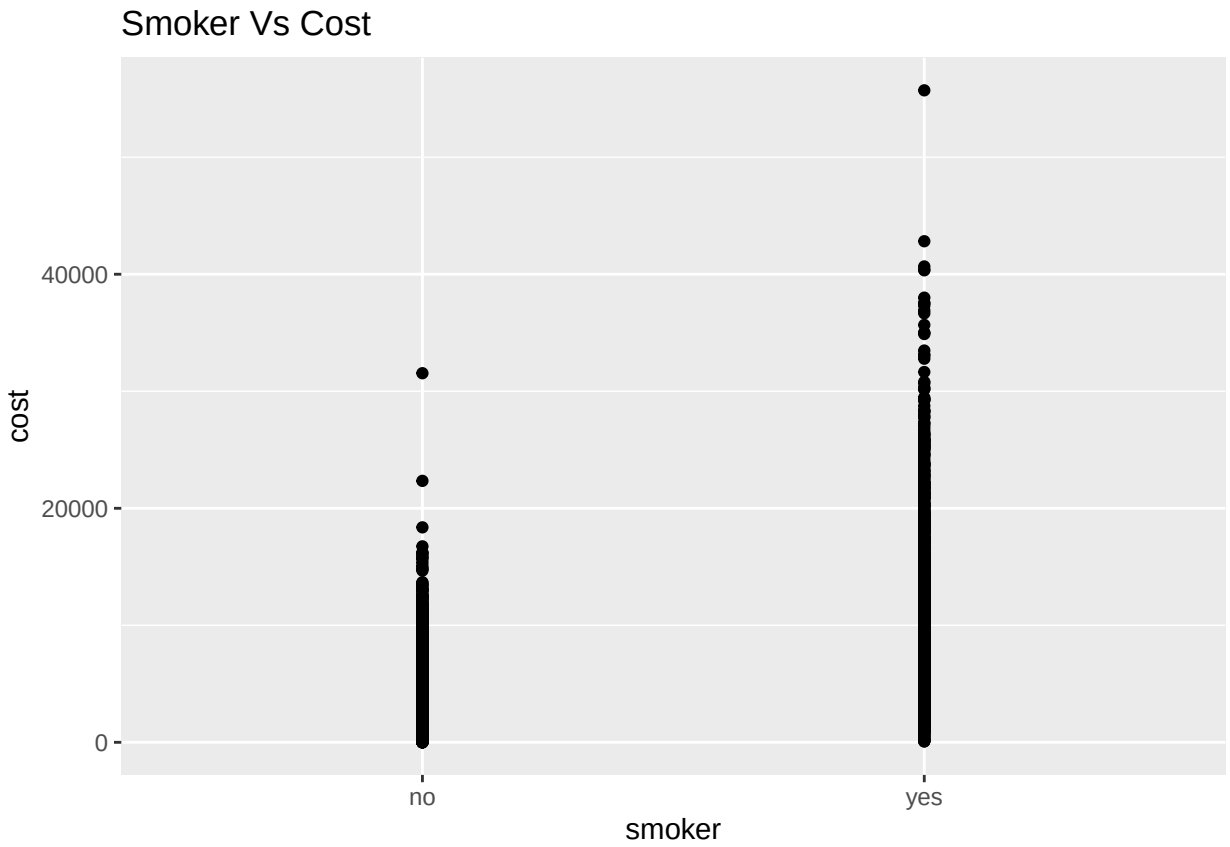

```
ggplot(Expensive, aes(x=age,y=cost))+geom_point() + ggtitle("Age Vs Cost of Expensive People")
```



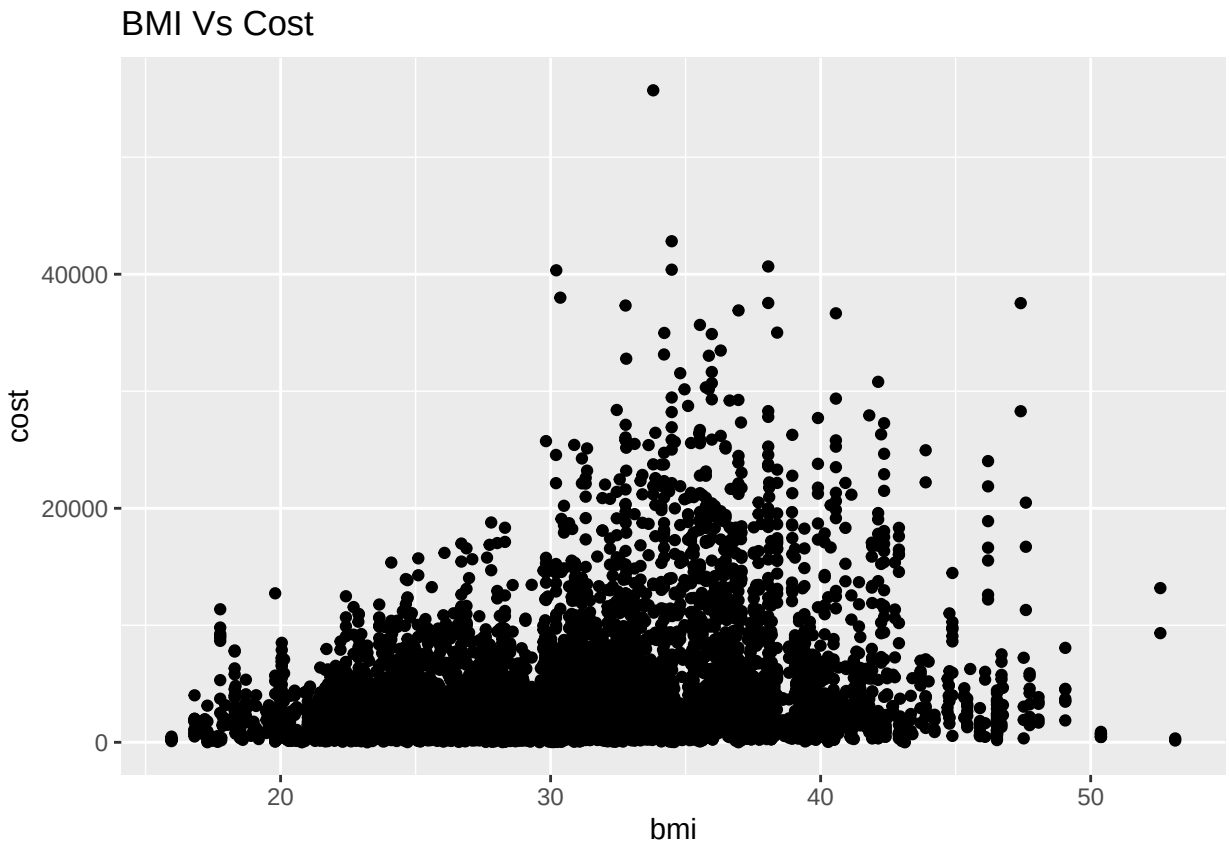

```
ggplot(proj_df, aes(x=age,y=cost))+geom_point() + ggtitle("Age Vs Cost")
```



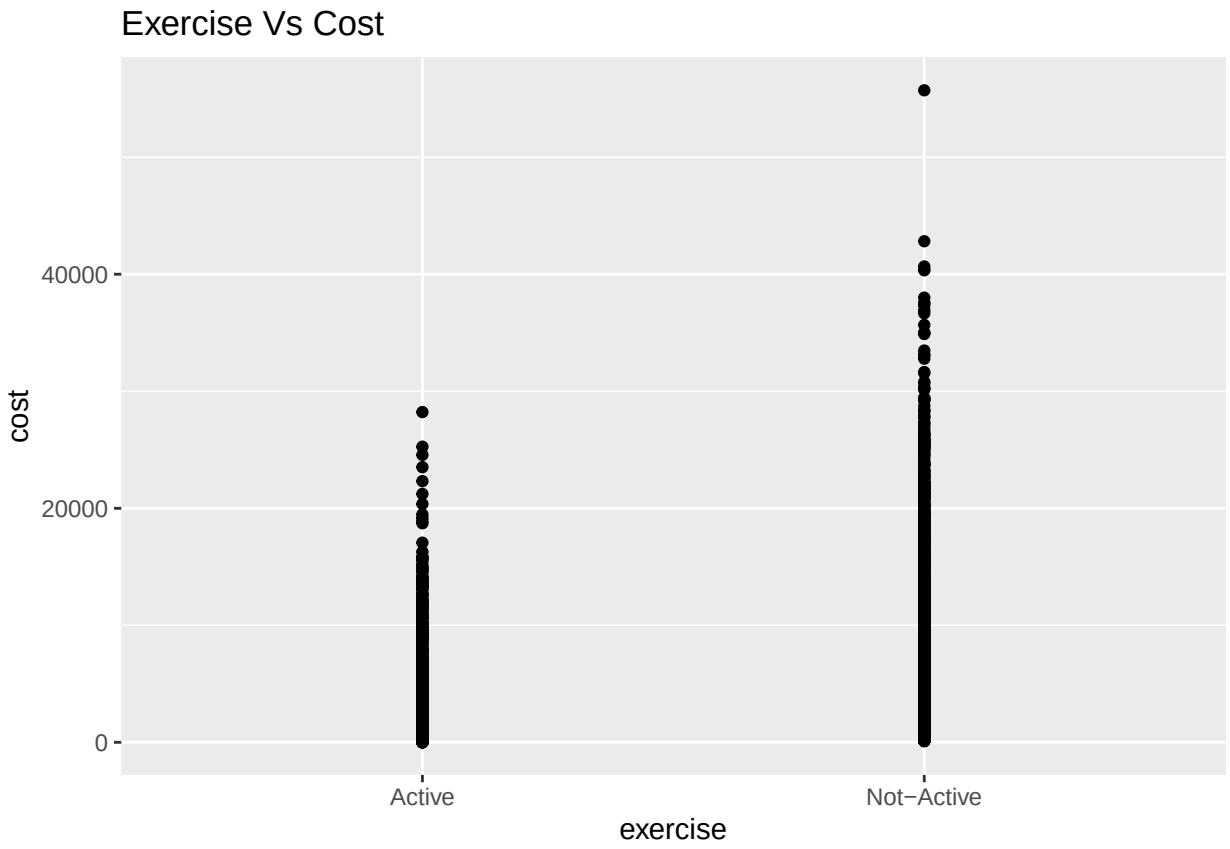

```
ggplot(proj_df, aes(x=smoker,y=cost))+geom_point() + ggtitle("Smoker Vs Cost")
```



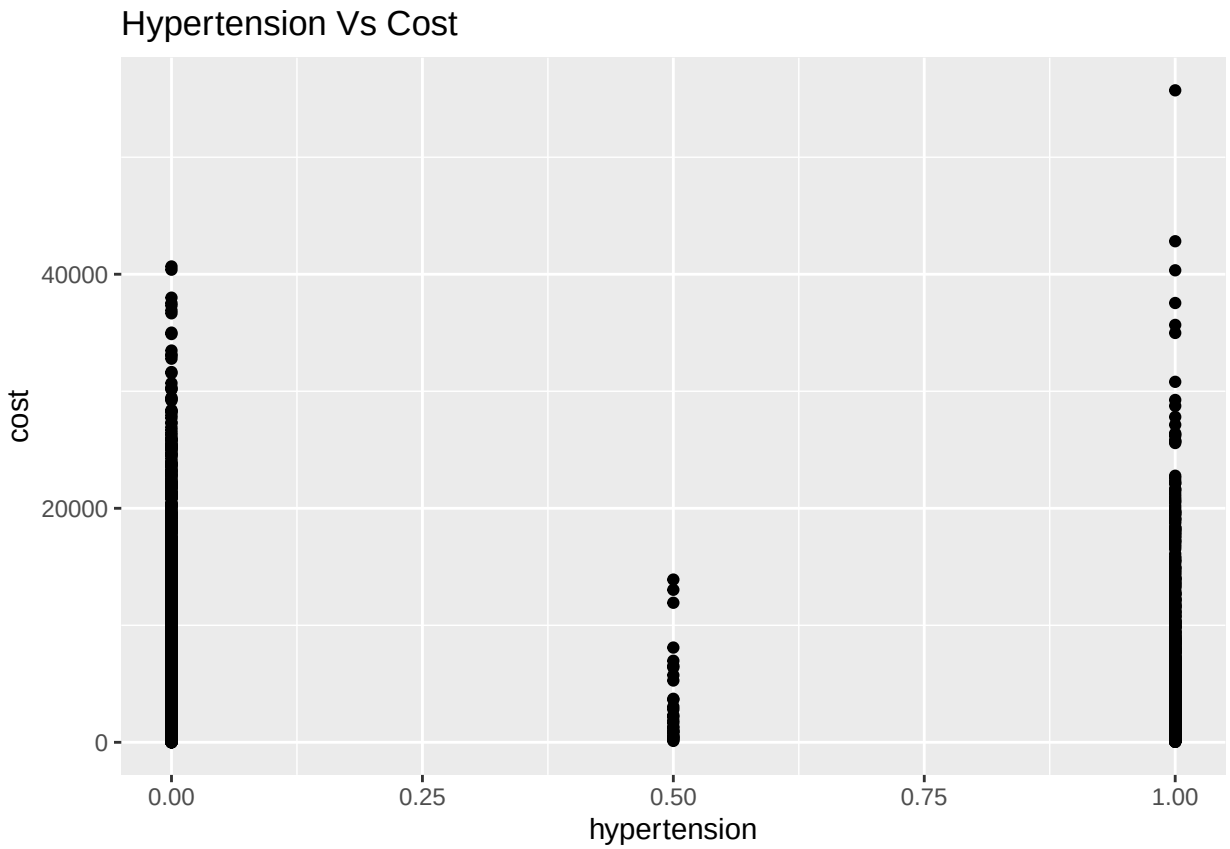

```
ggplot(proj_df, aes(x=bmi,y=cost))+geom_point() + ggtitle("BMI Vs Cost")
```



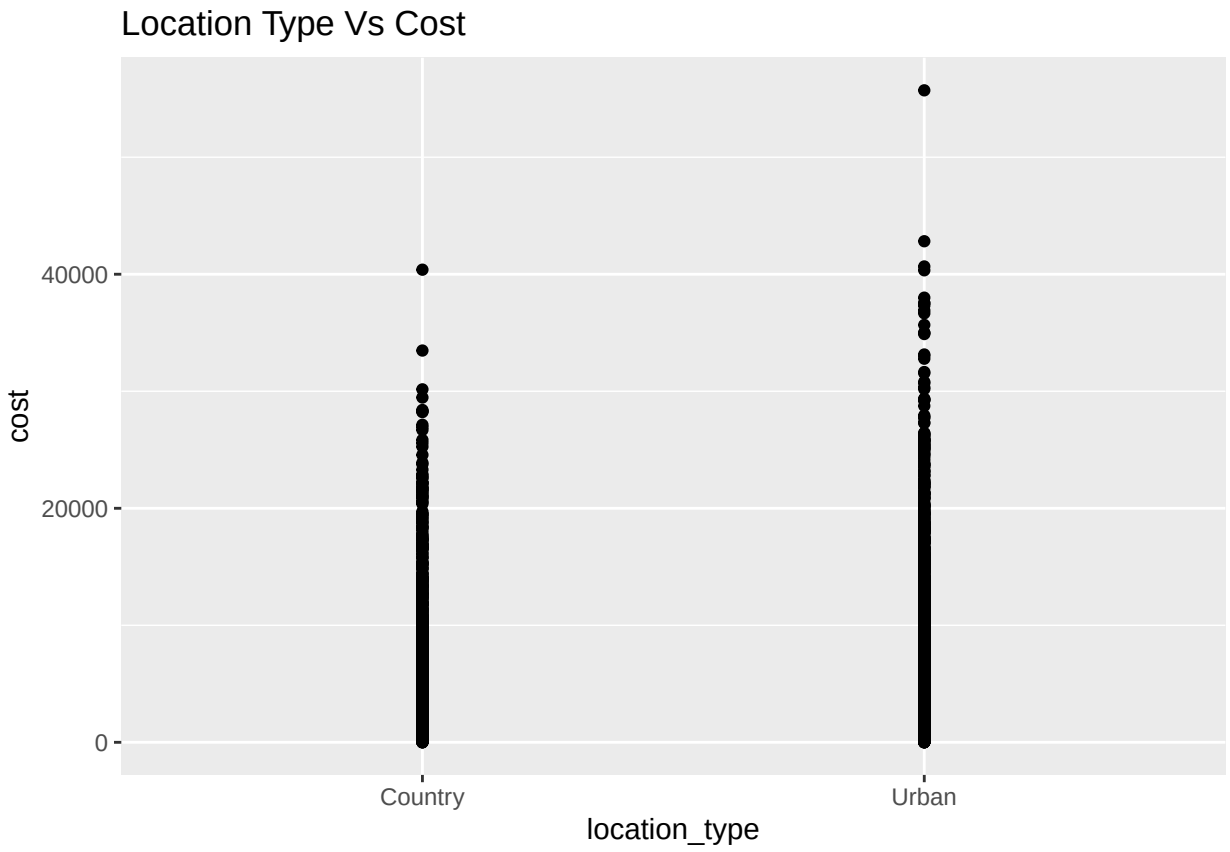

```
ggplot(proj_df, aes(x=exercise,y=cost))+geom_point() + ggtitle("Exercise Vs Cost")
```



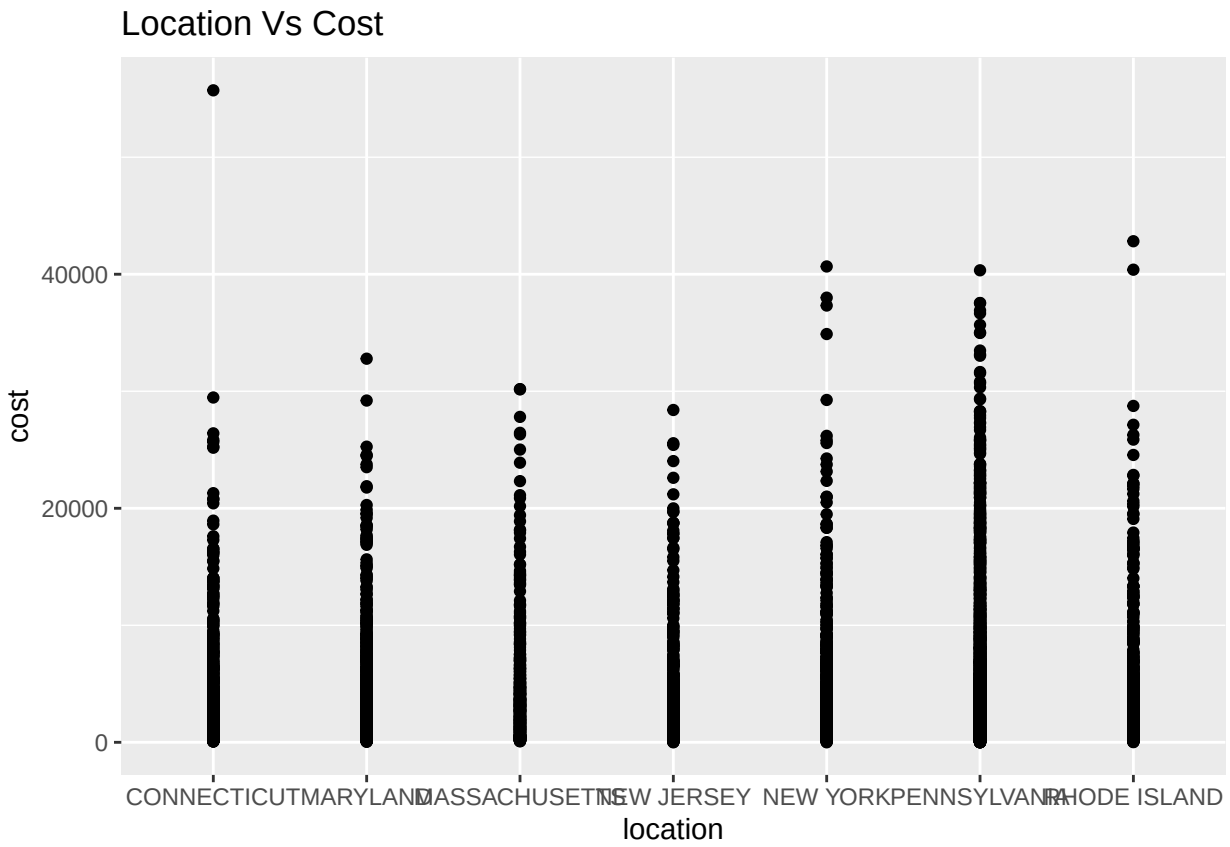

```
ggplot(proj_df, aes(x=hypertension,y=cost))+geom_point() + ggtitle("Hypertension Vs Cost")
```



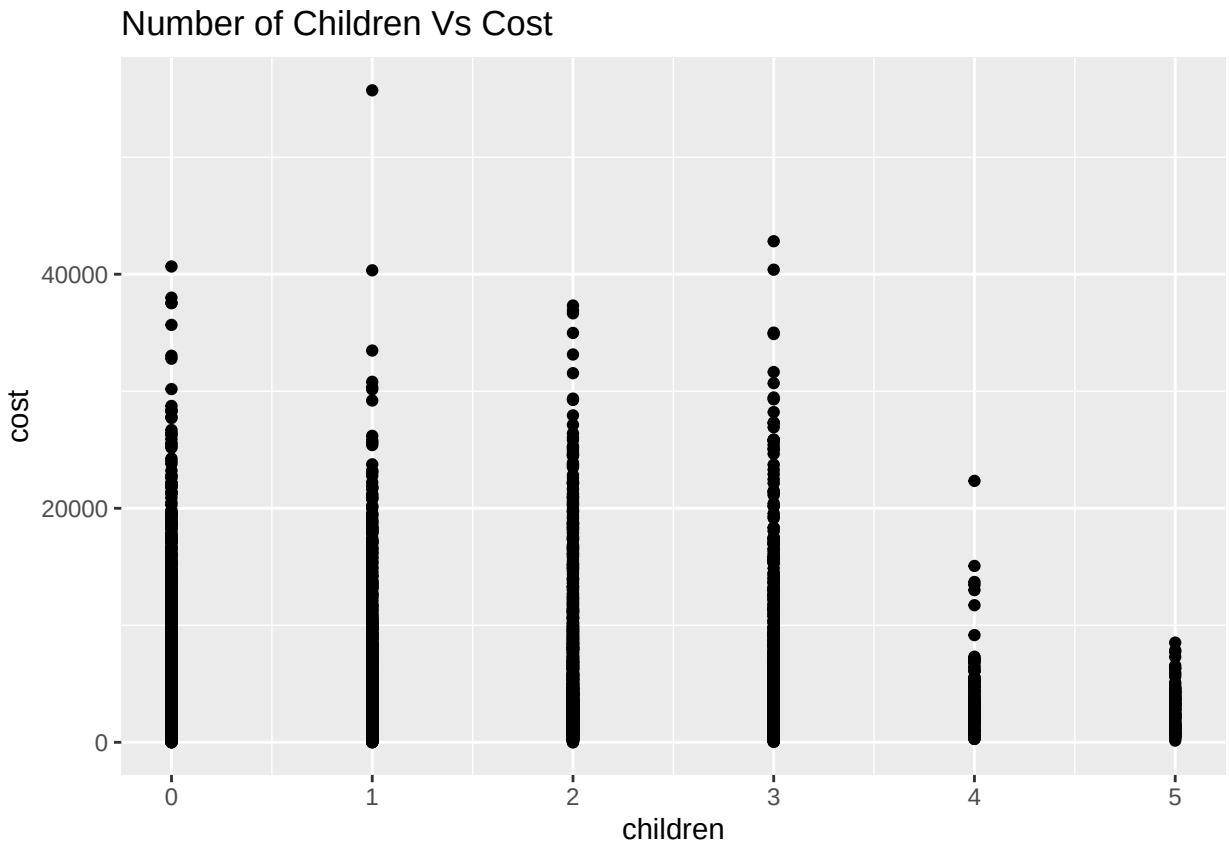

```
ggplot(proj_df, aes(x=location_type,y=cost))+geom_point() + ggtitle("Location Type Vs Cost")
```



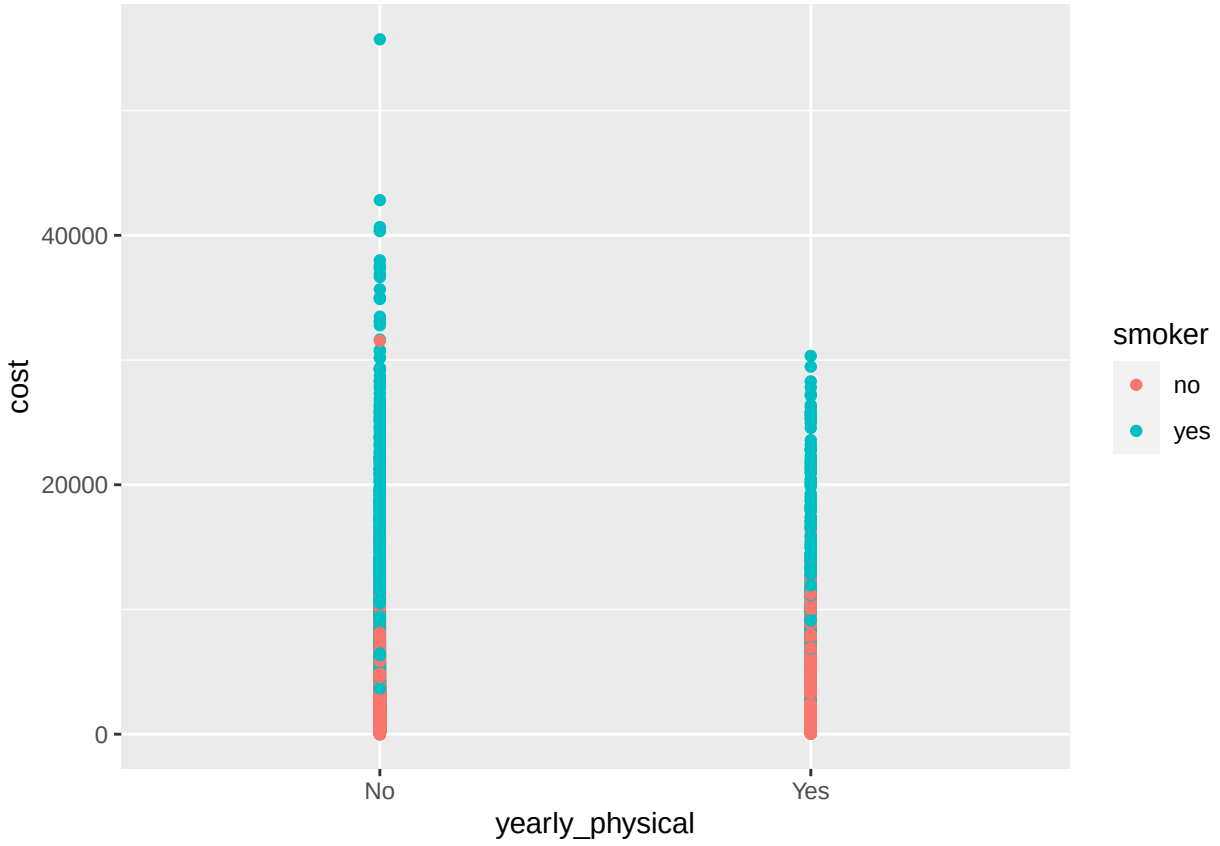

```
ggplot(proj_df, aes(x=location,y=cost))+geom_point() + ggtitle("Location Vs Cost")
```




```
ggplot(proj_df, aes(x=children,y=cost))+geom_point() + ggtitle("Number of Children Vs Cost")
```

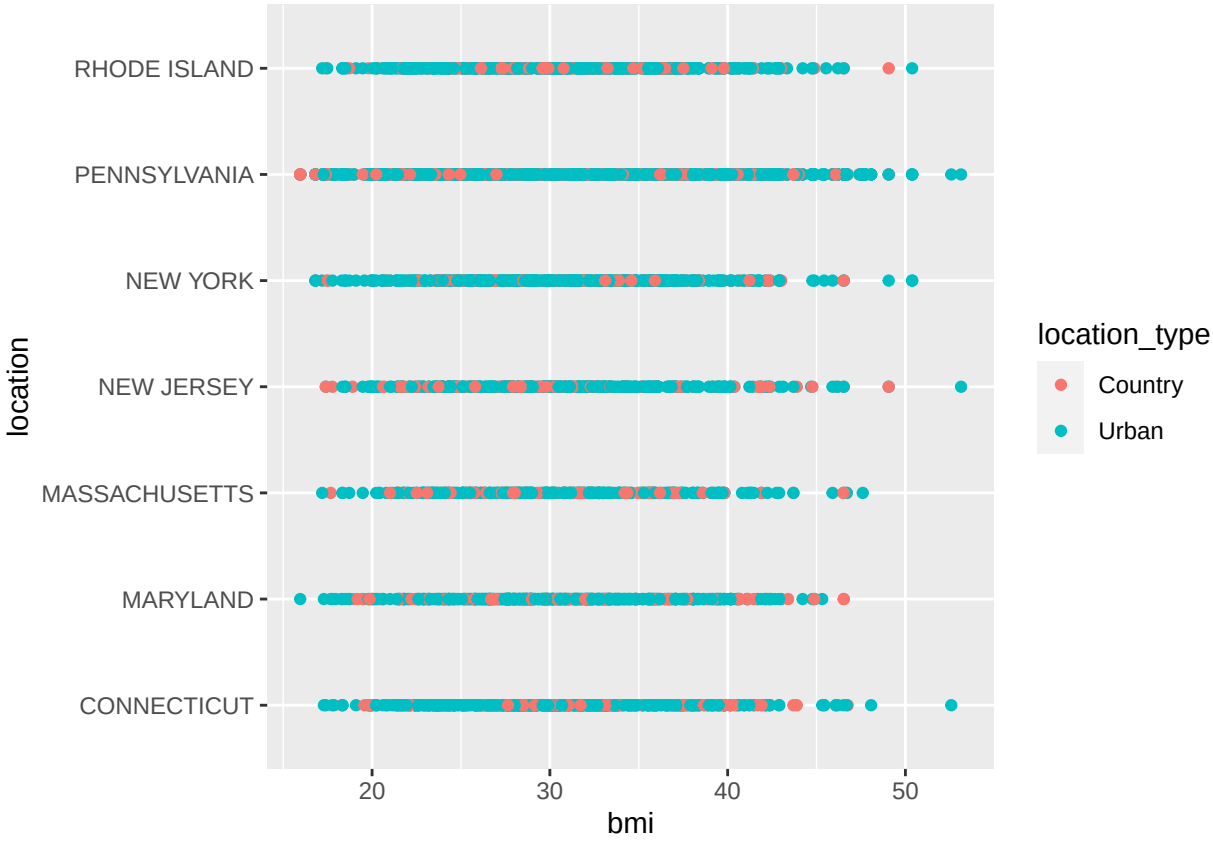



```
sctr_plot <- ggplot(proj_df)
sctr_plot <- sctr_plot + aes(x =yearly_physical, y=cost, color = smoker)
sctr_plot <- sctr_plot + geom_point()
sctr_plot
```

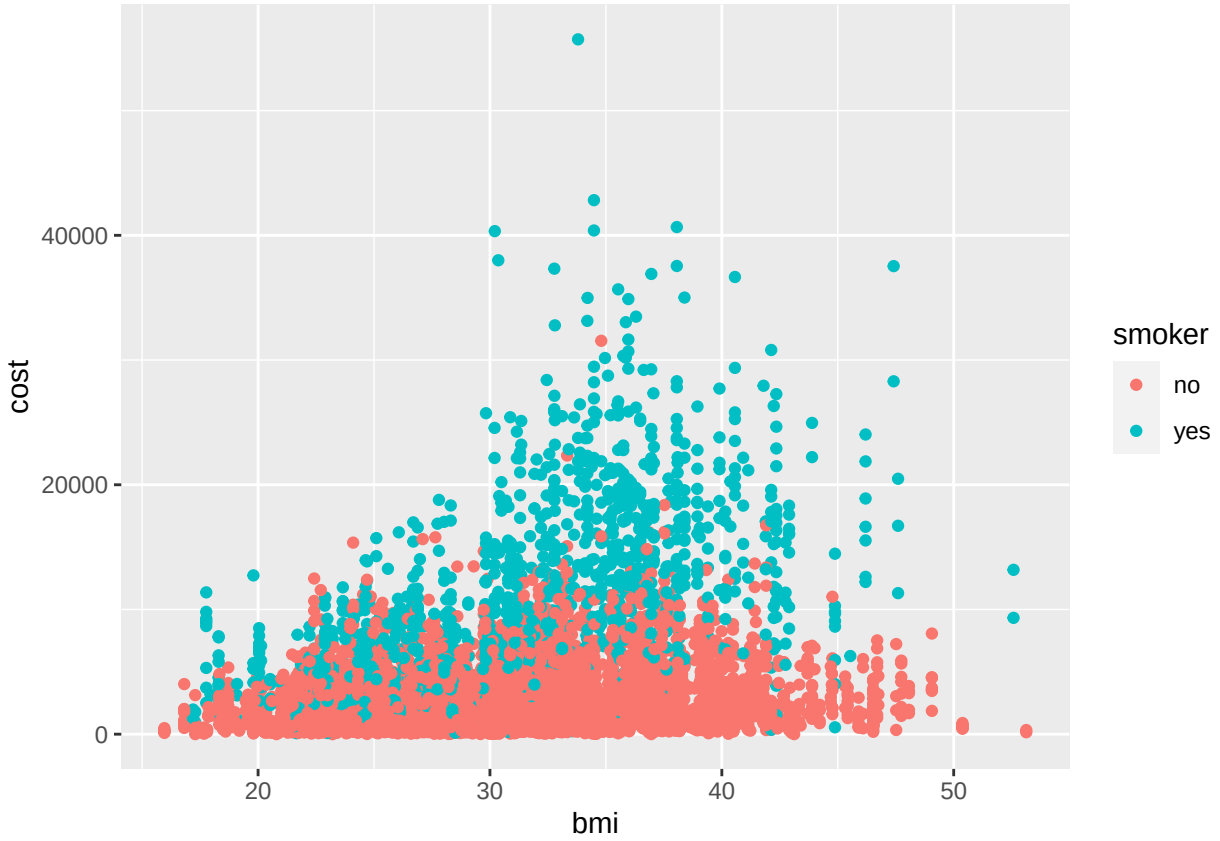



```
sctr_plot <- ggplot(proj_df)
sctr_plot <- sctr_plot + aes(x=age, y=cost, color = smoker)
sctr_plot <- sctr_plot + geom_point()
sctr_plot
```

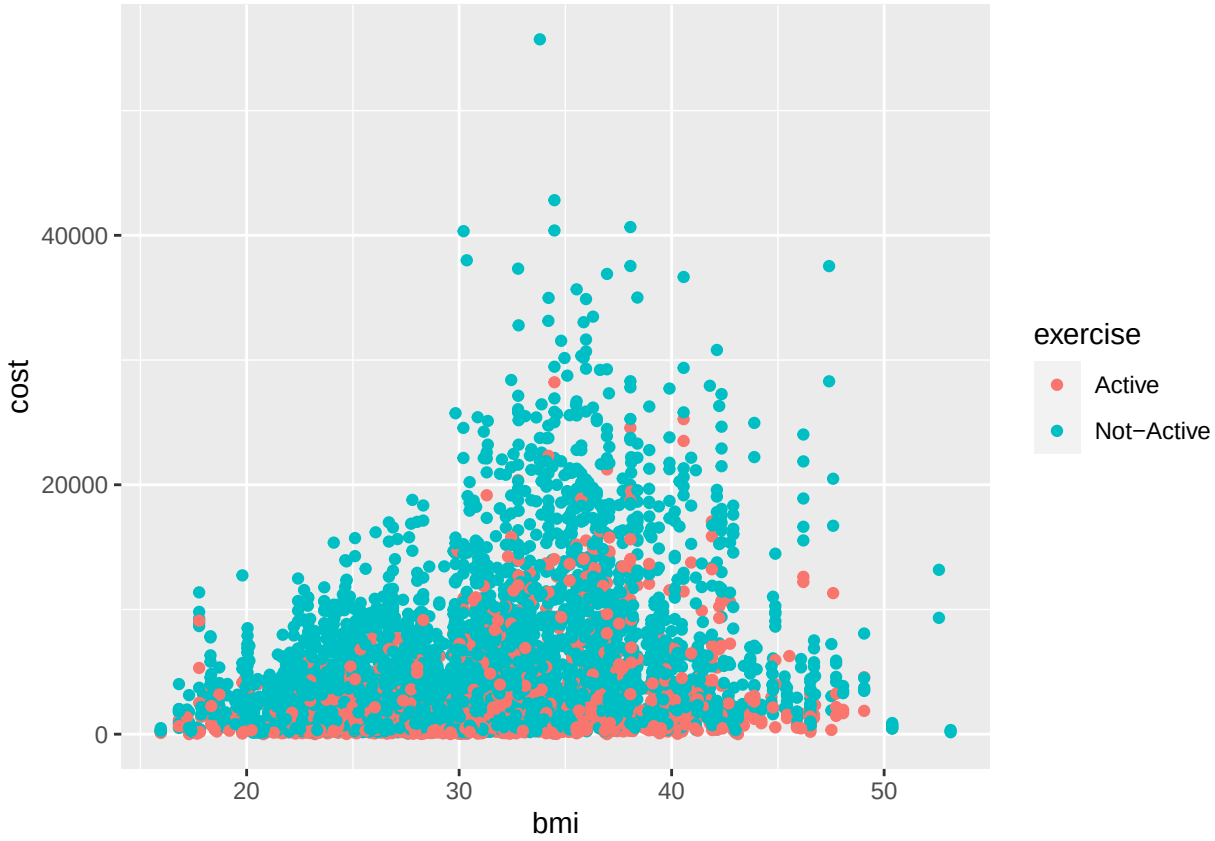

```
sctr_plot <- ggplot(proj_df)
sctr_plot <- sctr_plot + aes(x=bmi, y=location, color = location_type)
sctr_plot <- sctr_plot + geom_point()
sctr_plot
```



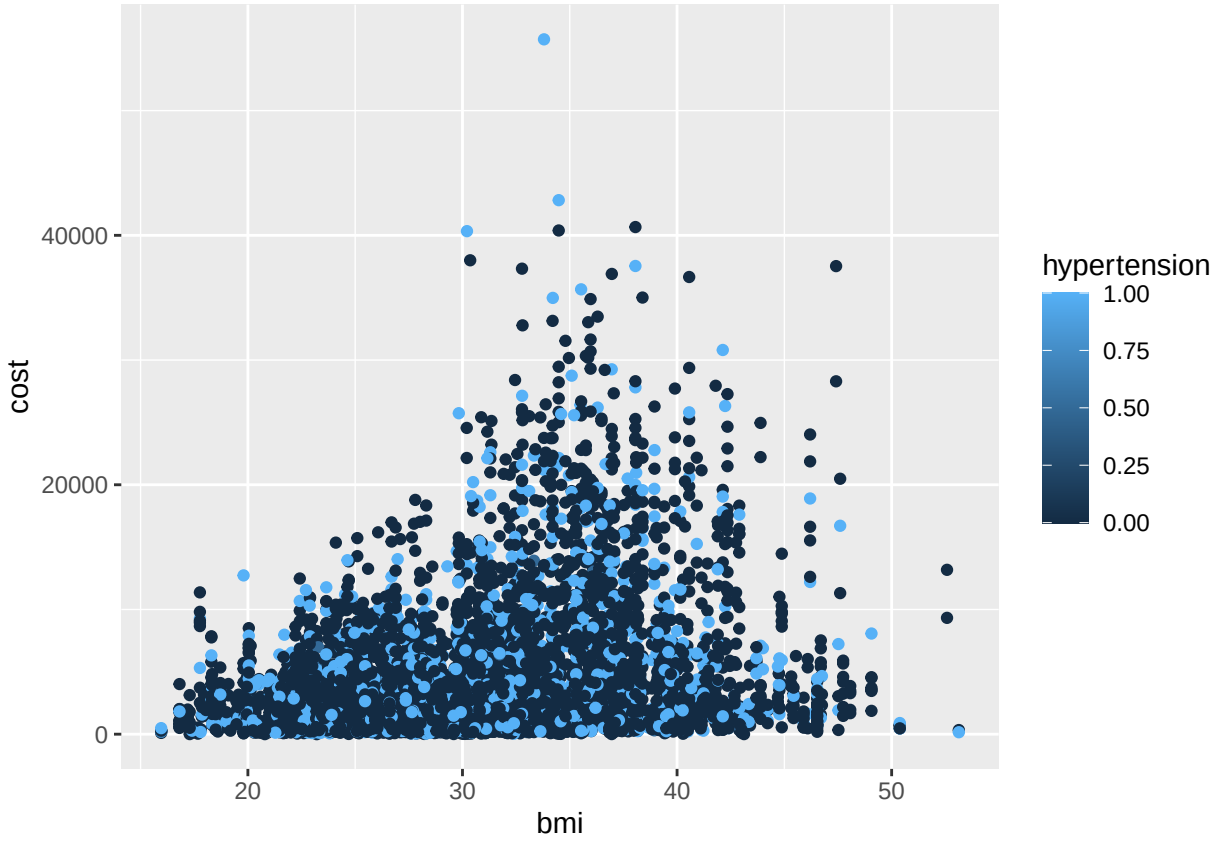

```
sctr_plot <- ggplot(proj_df)
sctr_plot <- sctr_plot + aes(x=bmi, y=cost, color = smoker)
sctr_plot <- sctr_plot + geom_point()
sctr_plot
```



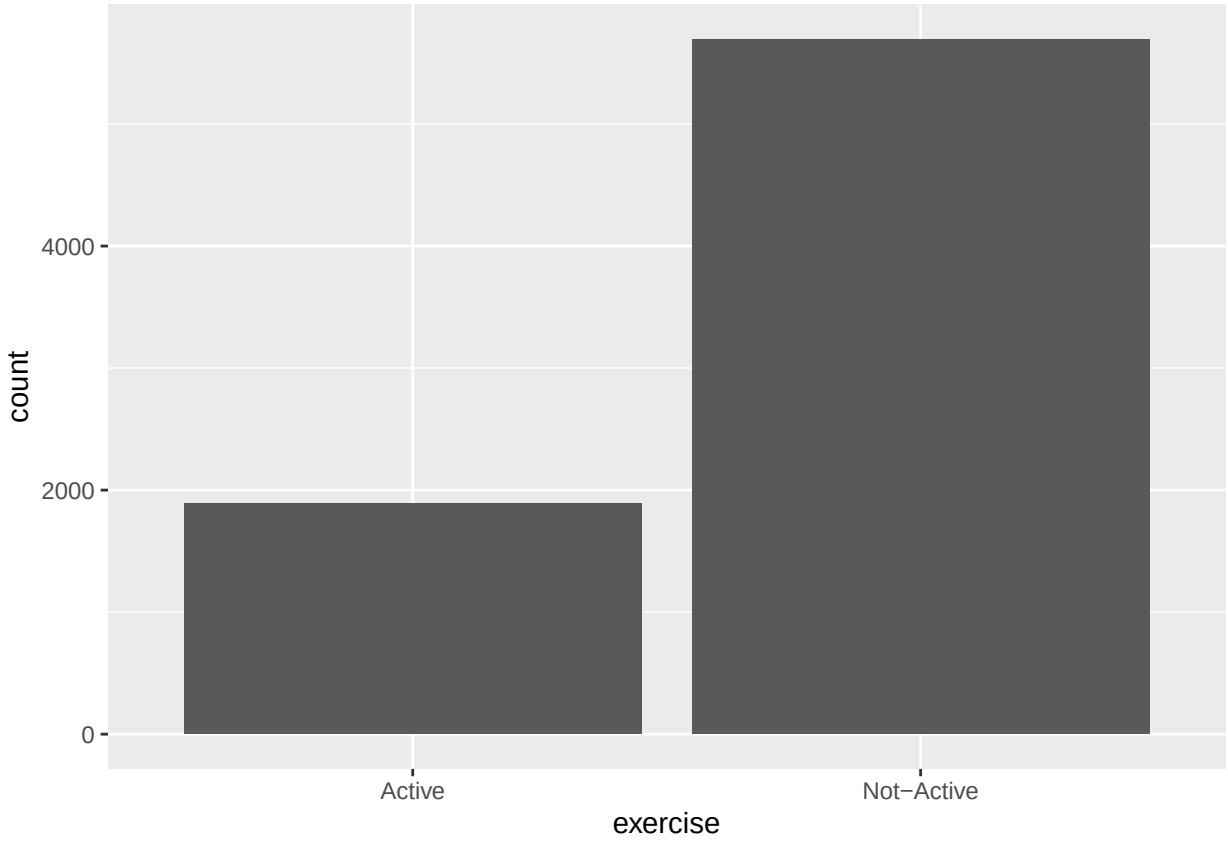

```
sctr_plot <- ggplot(proj_df)
sctr_plot <- sctr_plot + aes(x=bmi, y=cost, color = exercise)
sctr_plot <- sctr_plot + geom_point()
sctr_plot
```



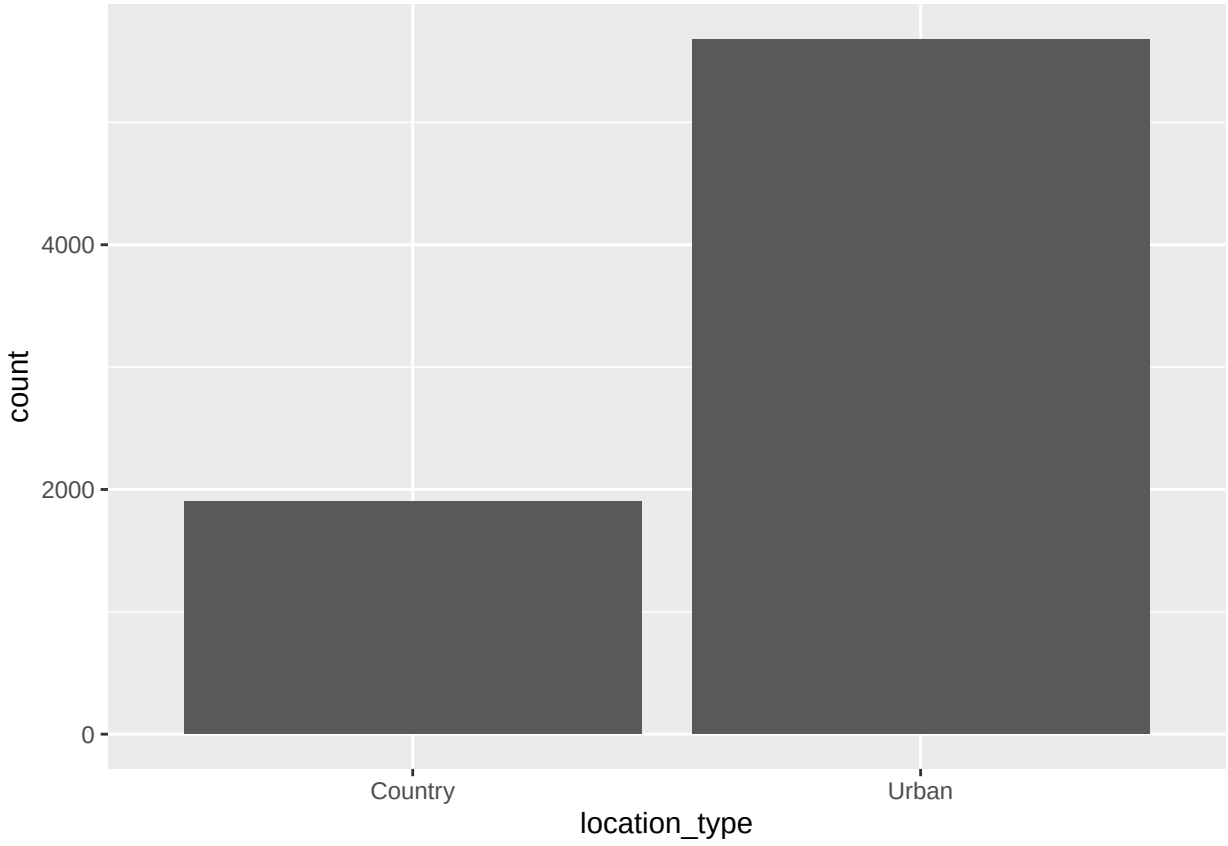

```
sctr_plot <- ggplot(proj_df)
sctr_plot <- sctr_plot + aes(x=bmi, y=cost, color = hypertension)
sctr_plot <- sctr_plot + geom_point()
sctr_plot
```



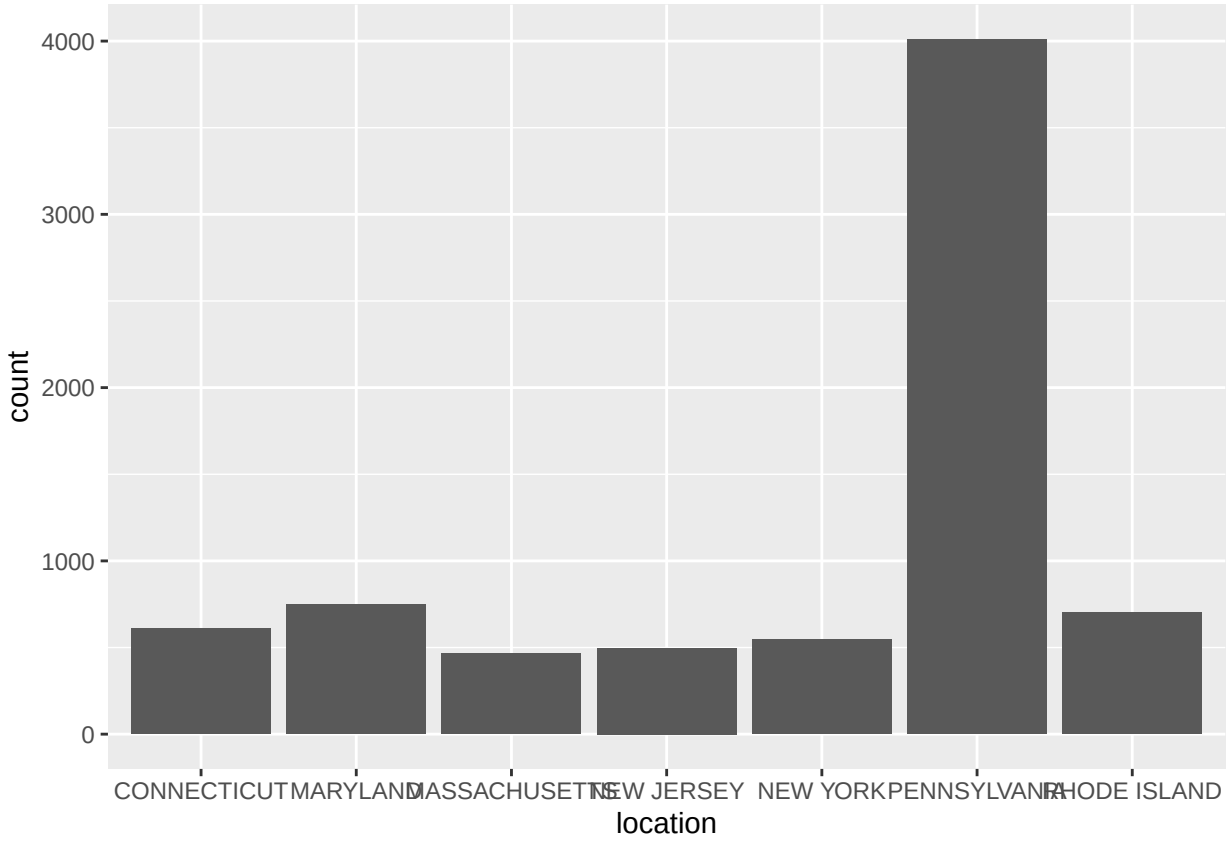

```
ggplot(proj_df) + aes(x=exercise) + geom_bar()
```



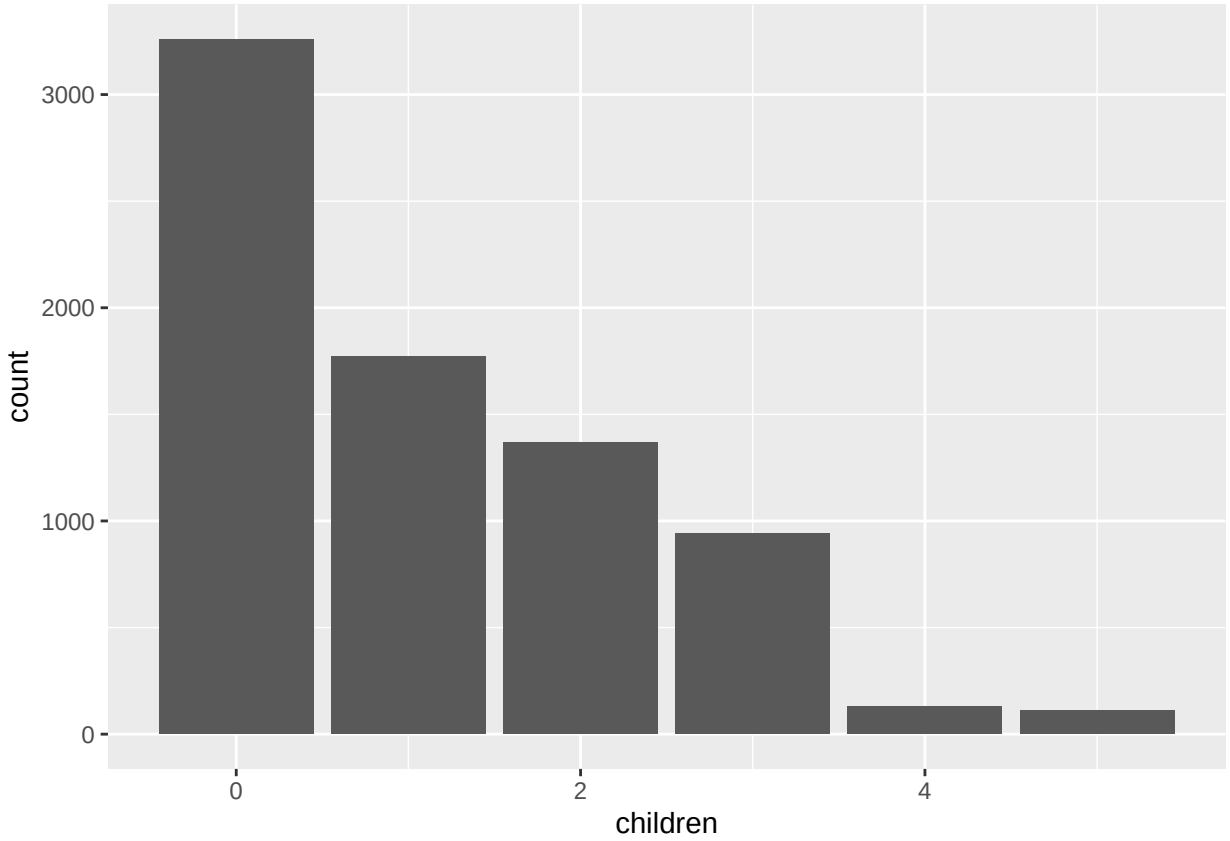

```
ggplot(proj_df) + aes(x=location_type) + geom_bar()
```



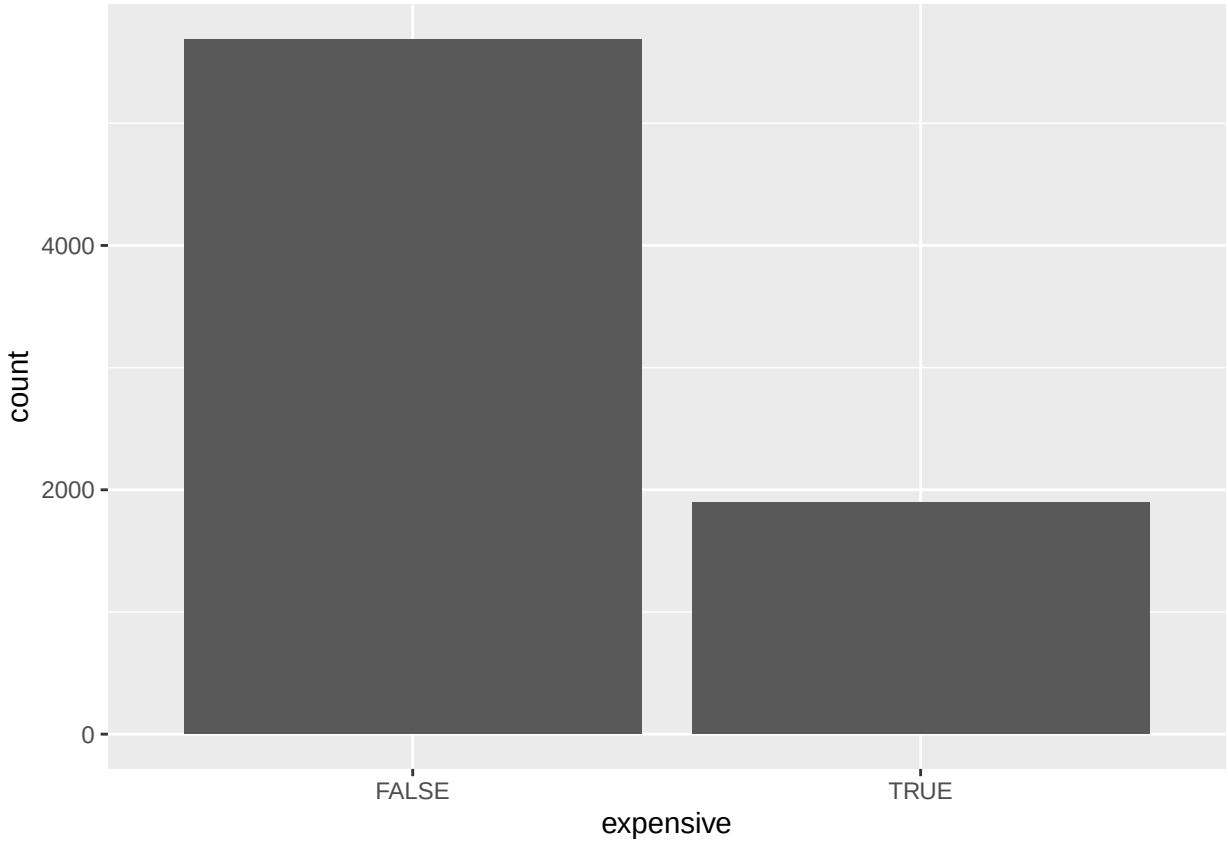

```
ggplot(proj_df) + aes(x=location) + geom_bar()
```




```
ggplot(proj_df) + aes(x=children) + geom_bar()
```

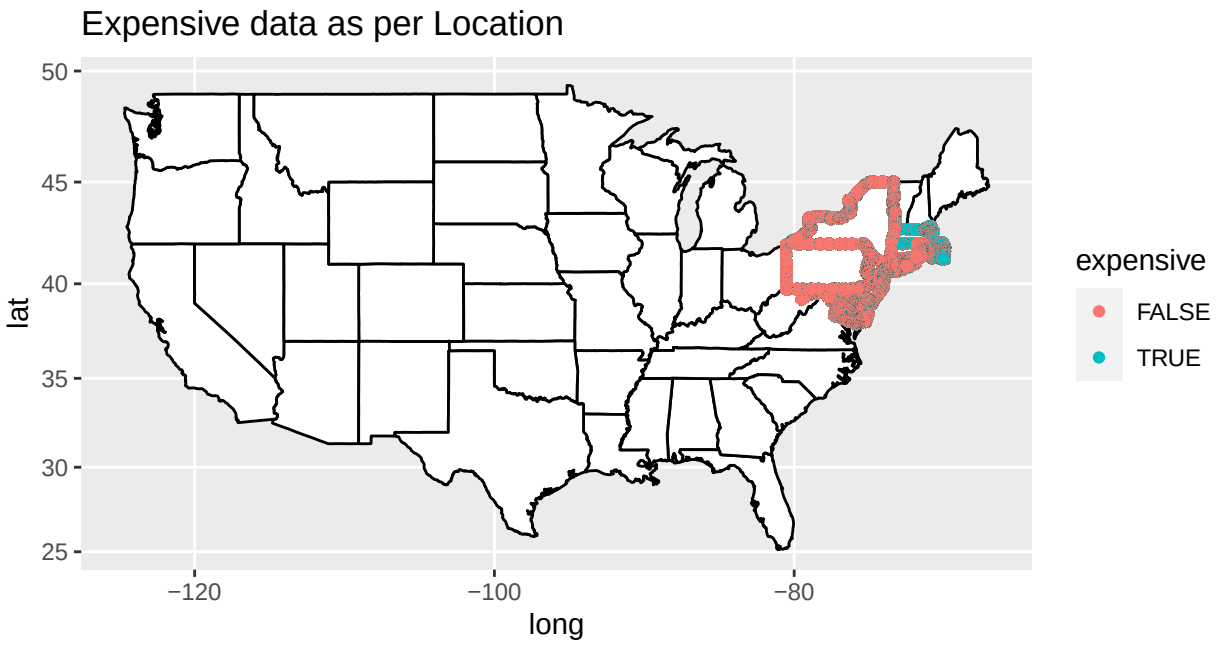



```
ggplot(proj_df) + aes(x=expensive) + geom_bar()
```



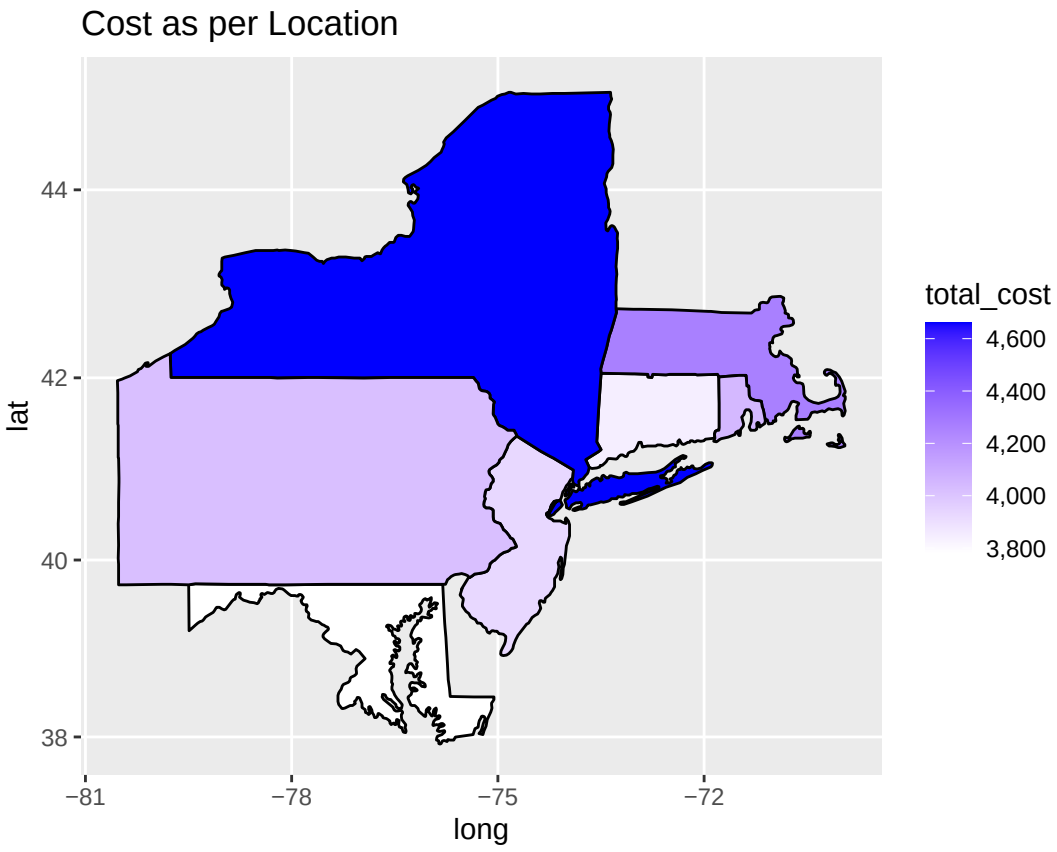

```
us <- map_data("state")
#View(us)
us$state_name <- tolower(us$region)
proj_df$state_name <- tolower(proj_df$state_name)
mapping <- merge(us,proj_df,by="state_name")
mapping <- mapping[order(mapping$order),]
#View(mapping)
```

```
map <- ggplot(us, aes(map_id="state"))
map <- map + aes(x=long, y=lat, group=group) + geom_polygon(fill = "white", color = "black")
map <- map + expand_limits(x=us$long, y=us$lat)
map <- map + coord_map("mercator") + ggtitle("Expensive data as per Location")
map <- map + geom_point(data=mapping,aes(x=long,y=lat,colour=expensive),inherit.aes =F)
map
```

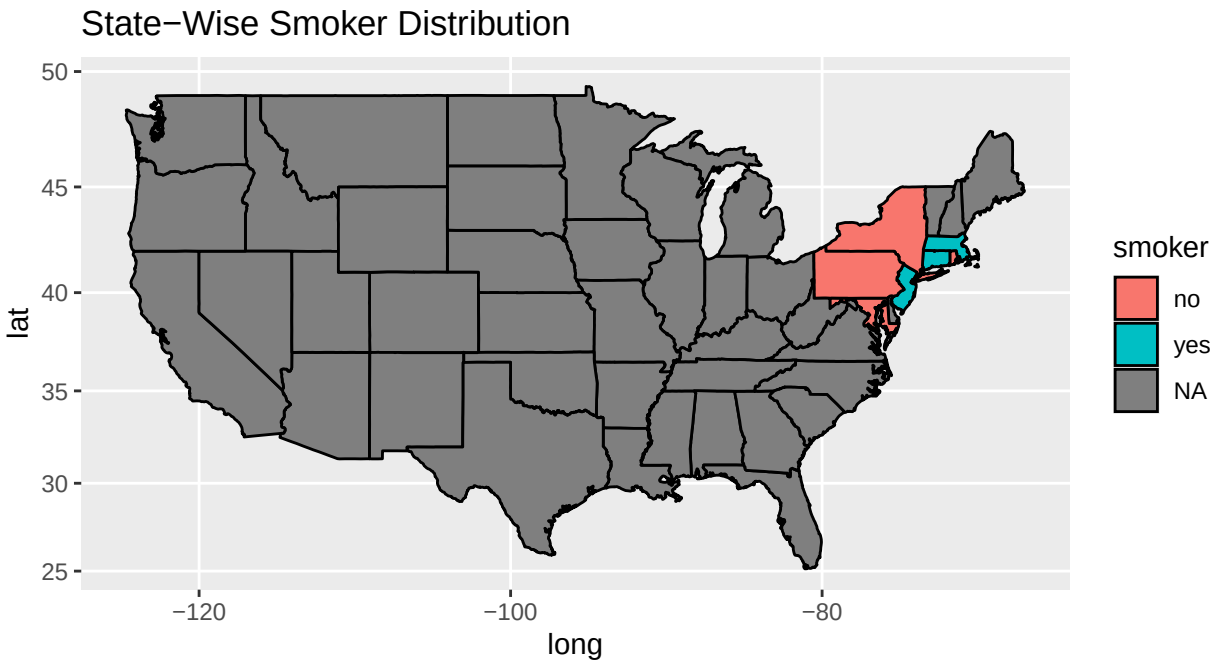



```
dfAgg1 <- proj_df %>% group_by(location) %>% summarise(total_cost = mean(cost))
dfAgg1$state <- tolower(dfAgg1$location)
#View(dfAgg1)
us <- map_data("state")
#View(us)
us$state_name <- tolower(us$region)
dfAgg1$state_name <- tolower(dfAgg1$state)
mapping <- merge(us,dfAgg1,by="state_name")
mapping <- mapping[order(mapping$order),]
#View(mapping)
```

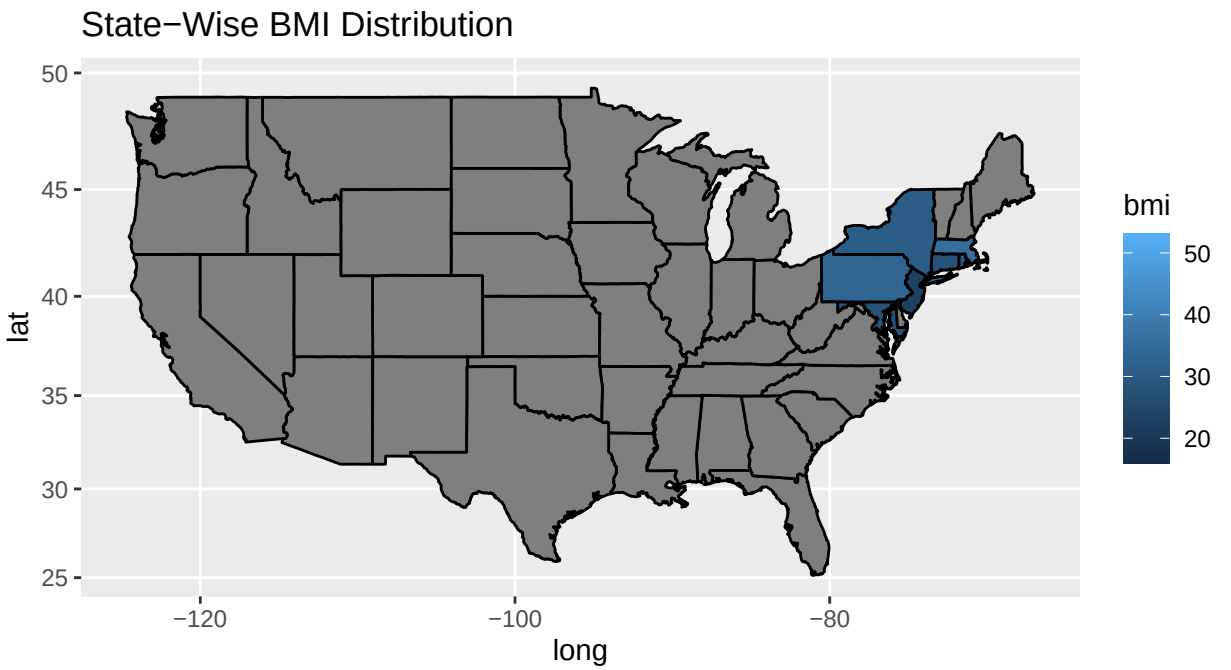
```
map <- ggplot(mapping)+ geom_polygon(aes(x=long, y=lat, group=group, fill = total_cost),color="black")+
map <- map + scale_fill_continuous(low = "white", high = "blue", name = "total_cost", label = scales::c
map
```



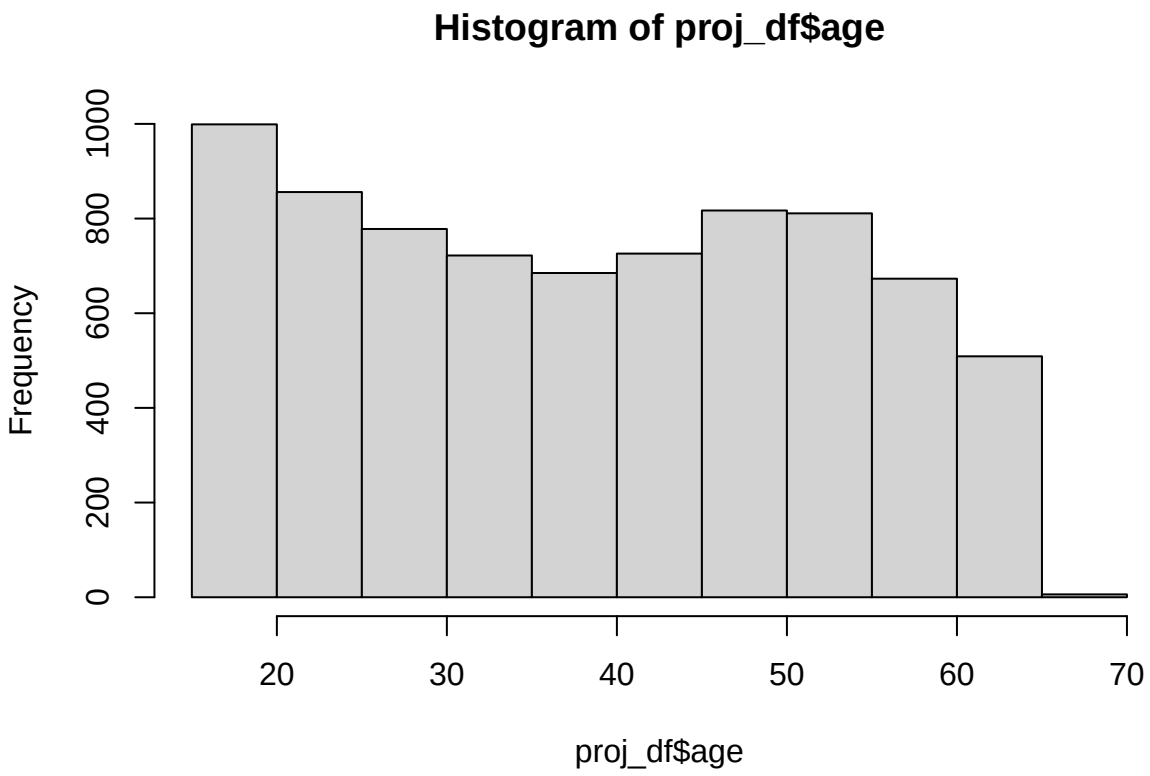

```
us <- map_data("state")
us$state_name = tolower(us$region)
proj_df$location <- tolower(proj_df$location)
dfMerged <- merge(proj_df, us, all.y = TRUE, by.x="location", by.y = "state_name")
dfMerged <- dfMerged %>% arrange(order)
map <- ggplot(dfMerged)
map <- map + aes(x=long, y=lat, group=group, fill=smoker) + geom_polygon(color = "black")
map <- map + expand_limits(x=dfMerged$long, y=dfMerged$lat)
map <- map + coord_map() + ggtitle("State-Wise Smoker Distribution")
map
```



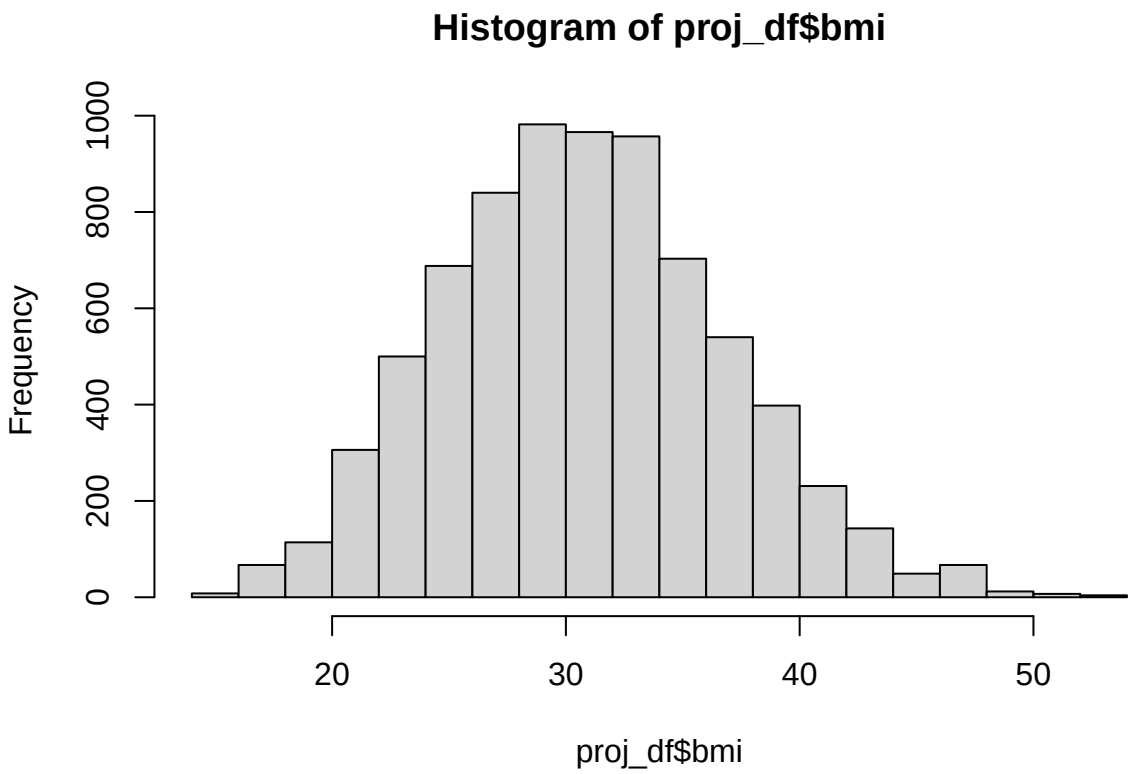

```
us <- map_data("state")
us$state_name = tolower(us$region)
proj_df$location <- tolower(proj_df$location)
dfMerged <- merge(proj_df, us, all.y = TRUE, by.x="location", by.y = "state_name")
dfMerged <- dfMerged %>% arrange(order)
map <- ggplot(dfMerged)
map <- map + aes(x=long, y=lat, group=group, fill=bmi) + geom_polygon(color = "black")
map <- map + expand_limits(x=dfMerged$long, y=dfMerged$lat)
map <- map + coord_map() + ggtitle("State-Wise BMI Distribution")
map
```



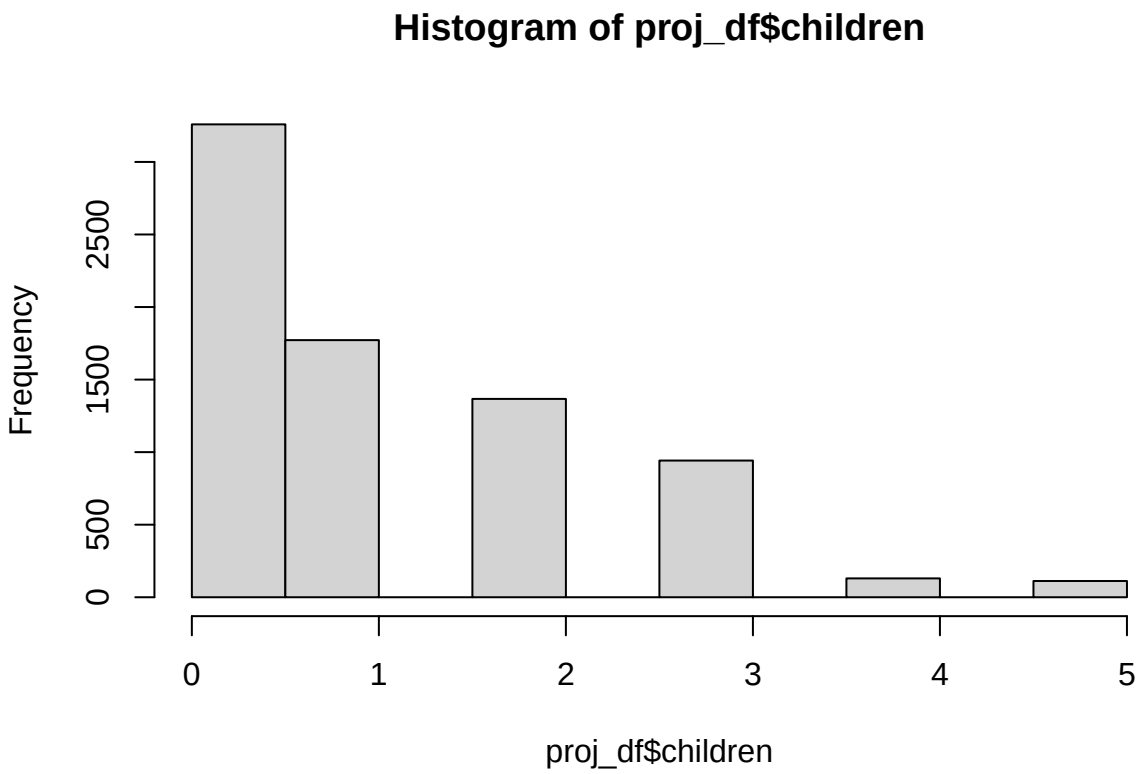

```
hist(proj_df$age)
```



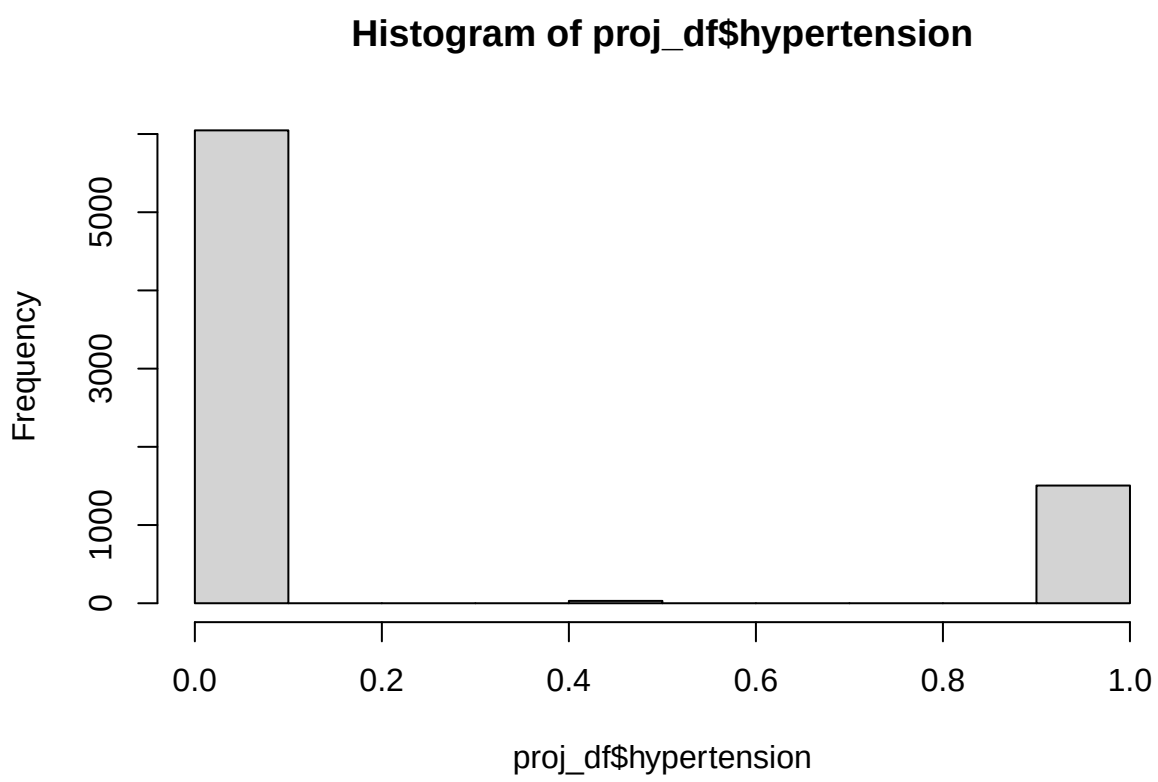

```
hist(proj_df$bmi)
```



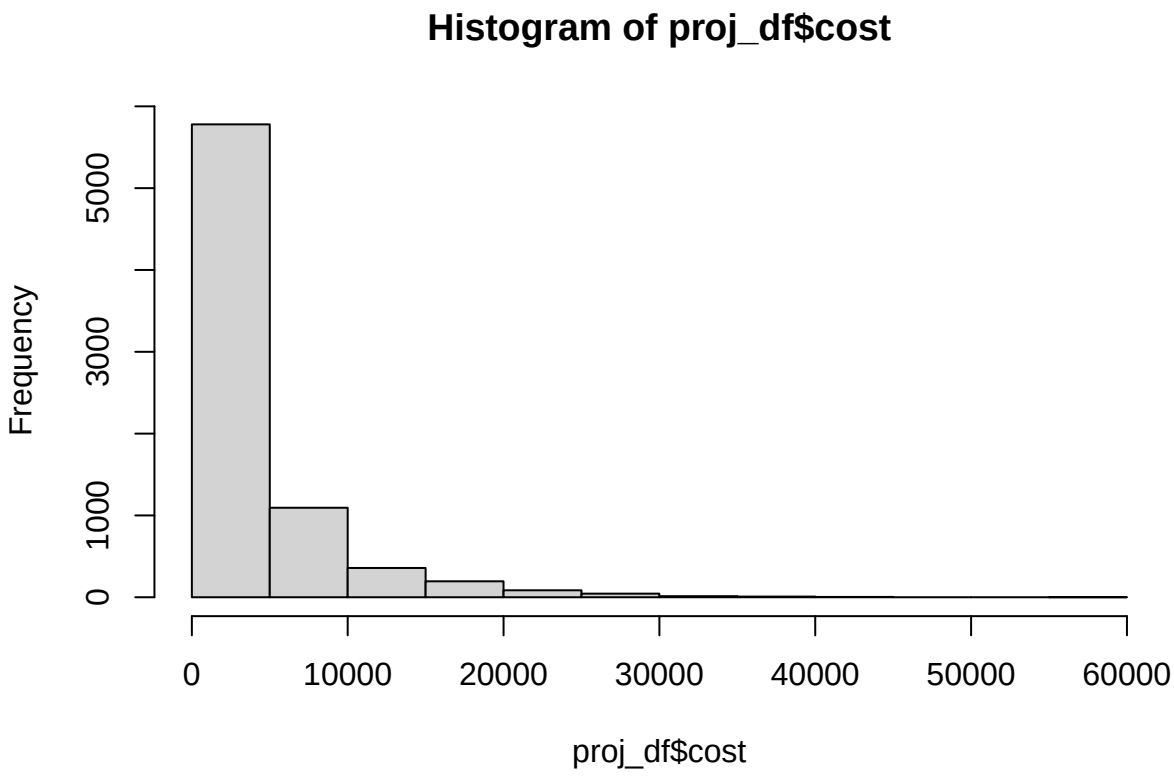

```
hist(proj_df$children)
```




```
hist(proj_df$hypertension)
```

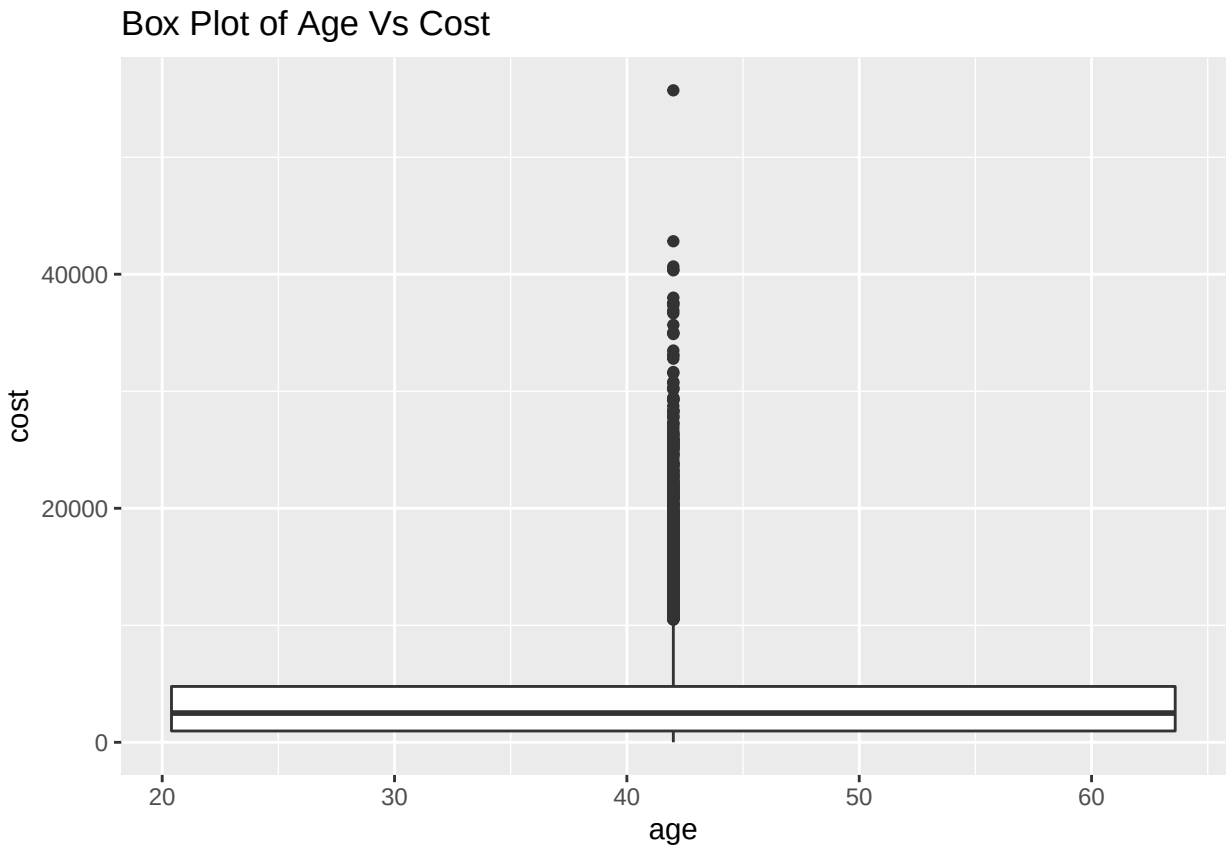



```
hist(proj_df$cost)
```



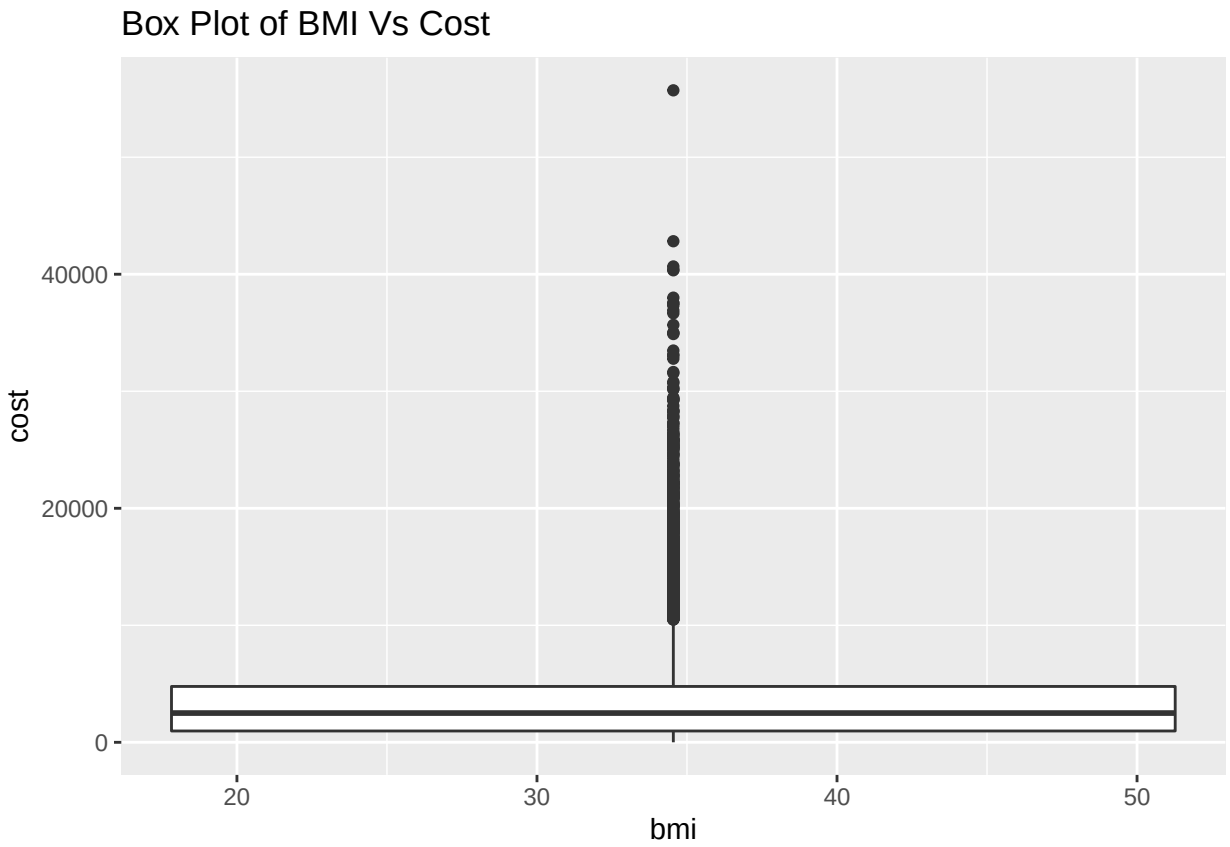

```
ggplot(proj_df)+aes(x=age,y=cost,)+geom_boxplot()+ ggtitle("Box Plot of Age Vs Cost")
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

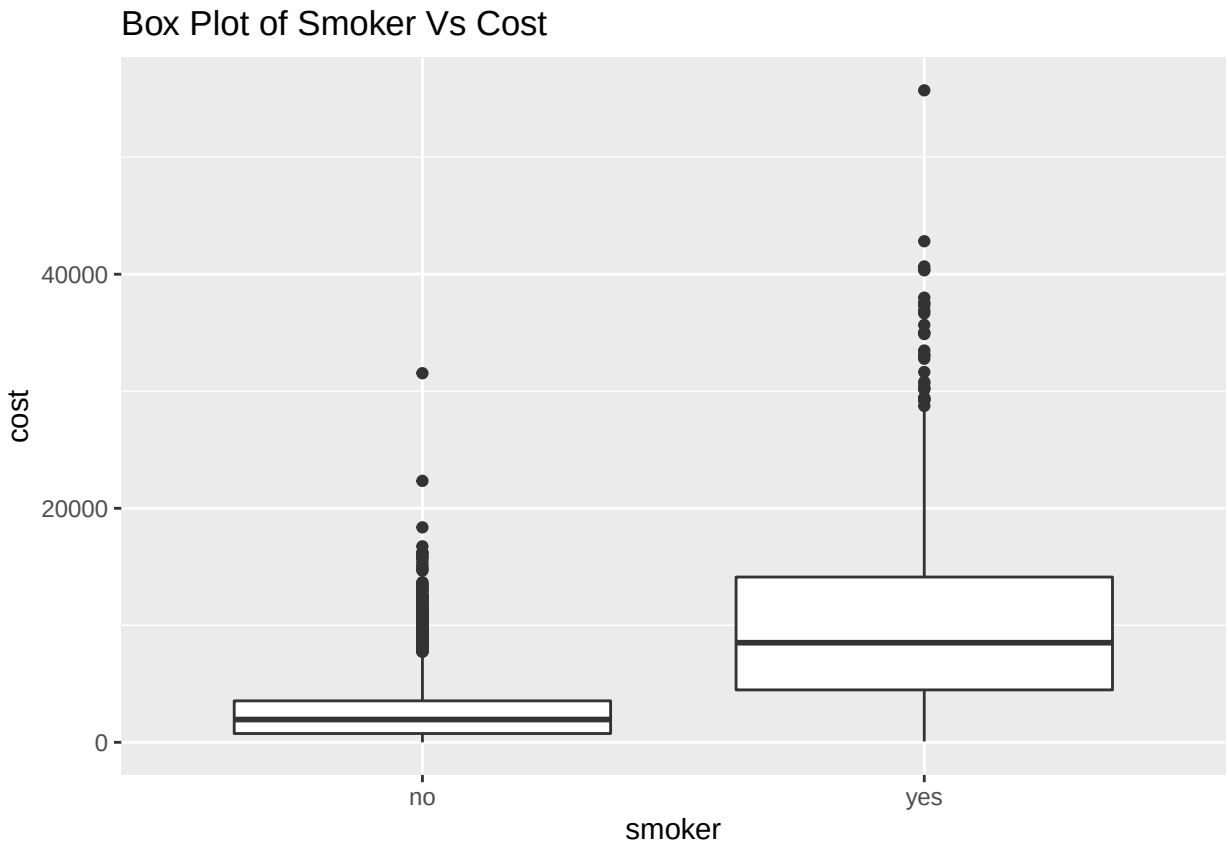



```
ggplot(proj_df)+aes(x=bmi,y=cost)+geom_boxplot()+ ggtitle("Box Plot of BMI Vs Cost")
```

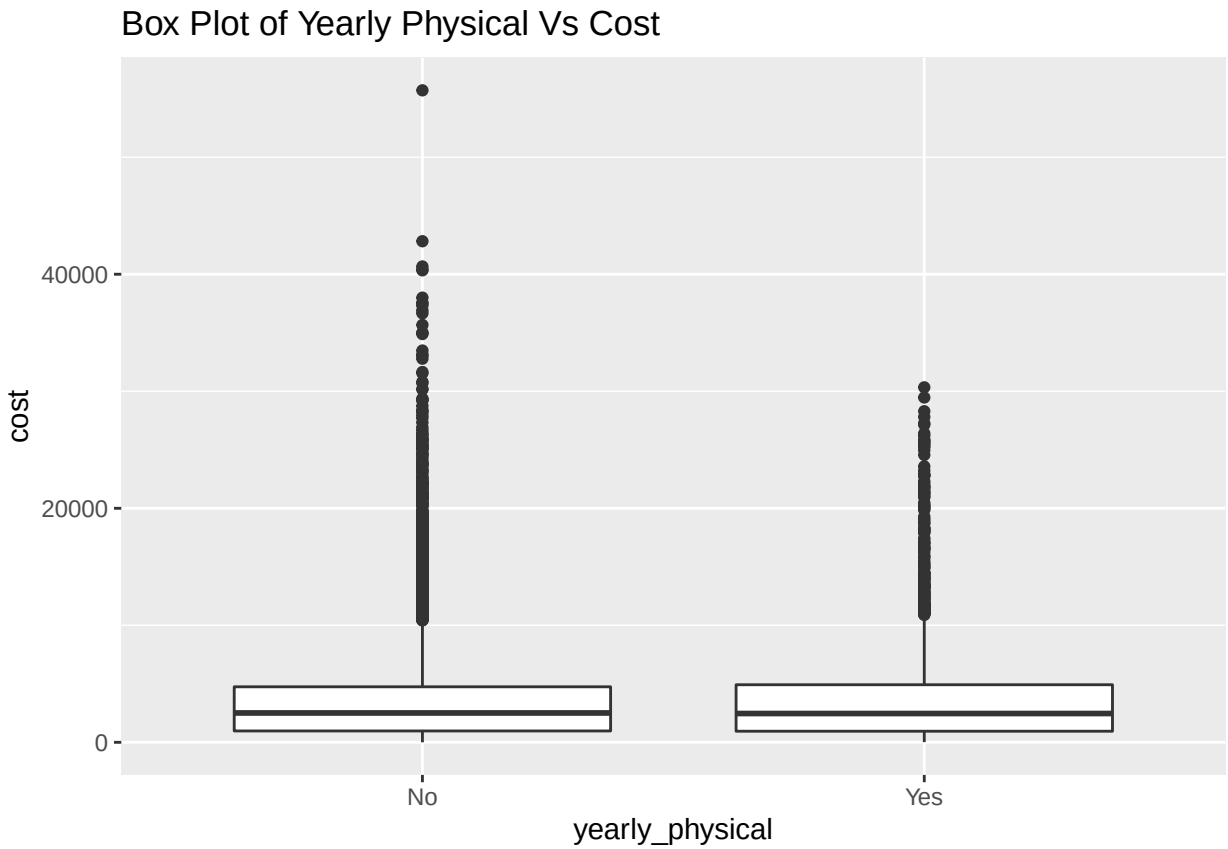
```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



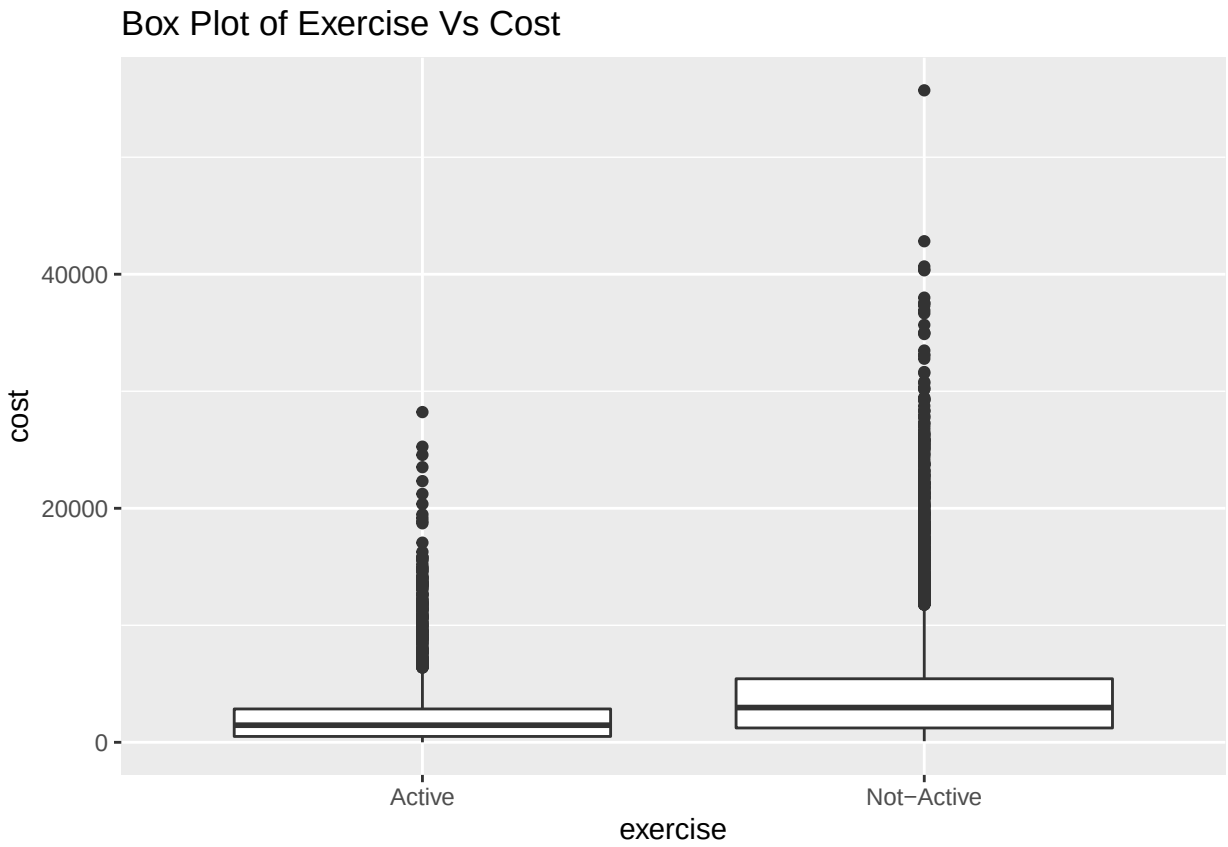

```
ggplot(proj_df)+aes(x=smoker,y=cost)+geom_boxplot()+ ggtitle("Box Plot of Smoker Vs Cost")
```




```
ggplot(proj_df)+aes(x=yearly_physical,y=cost)+geom_boxplot()+ ggtitle("Box Plot of Yearly Physical Vs C
```




```
ggplot(proj_df)+aes(x=exercise,y=cost)+geom_boxplot()+ ggtitle("Box Plot of Exercise Vs Cost")
```




```
ggplot(proj_df)+aes(x=hypertension,y=cost)+geom_boxplot()+ ggtitle("Box Plot of Hypertension Vs Cost")
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

1. Multiple regression model

```
mrLmOut <- lm(expensive ~ age+bmi+hypertension+smoker+exercise,proj_df)
summary(mrLmOut)
```

```
##
## Call:
## lm(formula = expensive ~ age + bmi + hypertension + smoker +
##     exercise, data = proj_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9337 -0.2020 -0.0588  0.1293  1.1509
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.6786414  0.0227758 -29.797  < 2e-16 ***
## age                 0.0074486  0.0002674  27.853  < 2e-16 ***
## bmi                 0.0126240  0.0006344  19.899  < 2e-16 ***
## hypertension        0.0352105  0.0094574   3.723 0.000198 ***
## smokeryes           0.5966752  0.0095265  62.633  < 2e-16 ***
## exerciseNot-Active  0.1687156  0.0087288  19.329  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.3286 on 7576 degrees of freedom
## Multiple R-squared:  0.4245, Adjusted R-squared:  0.4241
## F-statistic:  1118 on 5 and 7576 DF,  p-value: < 2.2e-16
```

Conversion to factors

```
proj_fact <- data.frame(
  #X= as.factor(proj_df$X),
  age= (proj_df$age),
  bmi = (proj_df$bmi),
  #children = as.factor(proj_df$children),
  smoker = (proj_df$smoker),
  #location = as.factor(proj_df$location),
  #location_type = as.factor(proj_df$location_type),
  #education_level = as.factor(proj_df$education_level),
  yearly_physical = (proj_df$yearly_physical),
  exercise = (proj_df$exercise),
  #married = as.factor(proj_df$married),
  hypertension = (proj_df$hypertension),
  #gender = (proj_df$gender),
  #cost= (proj_df$cost),
  expensive = as.factor(proj_df$expensive)
)
```

SVM MODELS

```
proj_df$expensive <- as.factor(proj_df$expensive)
TrnList <- createDataPartition(y=proj_df$expensive, p=.60,list=FALSE)
TrnSet <- proj_df[TrnList,]
TstSet <- proj_df[-TrnList,]
#proj_df$expensive <- as.factor(proj_df$expensive)
#View(TrnSet)
```

```
SVMmod <- ksvm(data = TrnSet, expensive~ age+bmi+children+smoker+hypertension+exercise+yearly_physical,
summary(SVMmod)
```

```
## Length  Class   Mode
##      1   ksvm     S4
```

```
svmPredict <- predict(SVMmod, newdata = TstSet, type = "response" )
confusionMatrix(svmPredict,as.factor(TstSet$expensive))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  2202  304
##      TRUE     72  454
##
##                Accuracy : 0.876
##                  95% CI : (0.8637, 0.8875)
##     No Information Rate : 0.75
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6317
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9683
##             Specificity : 0.5989
##          Pos Pred Value : 0.8787
##          Neg Pred Value : 0.8631
##              Prevalence : 0.7500
##          Detection Rate : 0.7263
##    Detection Prevalence : 0.8265
##       Balanced Accuracy : 0.7836
##
##        'Positive' Class : FALSE
##
```

Apriori Algorithm

```
data_apr <- proj_fact
data_apr <- as(data_apr,'transactions')
```

```
## Warning: Column(s) 1, 2, 3, 4, 5, 6 not logical or factor. Applying default
## discretization (see '? discretizeDF').
```

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

```
proj_rules <- apriori(data_apr,
 parameter=list(supp=0.030, conf=0.7),
 control=list(verbose=F),
 appearance=list(default="lhs",rhs=("expensive=TRUE")))

summary(proj_rules)
```

```
## set of 40 rules
##
## rule length distribution (lhs + rhs):sizes
##  2  3  4  5  6
##  1  8 16 12  3
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.0     4.0     4.0     4.2     5.0     6.0
##
## summary of quality measures:
##     support          confidence        coverage            lift
##  Min.   :0.03231   Min.   :0.7169   Min.   :0.03653   Min.   :2.869
##  1st Qu.:0.03884   1st Qu.:0.7717   1st Qu.:0.04501   1st Qu.:3.087
##  Median :0.04524   Median :0.8489   Median :0.05394   Median :3.397
##  Mean   :0.05838   Mean   :0.8511   Mean   :0.07080   Mean   :3.405
##  3rd Qu.:0.05958   3rd Qu.:0.9506   3rd Qu.:0.07112   3rd Qu.:3.803
```

```
##  Max.   :0.14244   Max.   :1.0000   Max.   :0.19507   Max.   :4.001
##      count
##  Min.   : 245.0
##  1st Qu.: 294.5
##  Median : 343.0
##  Mean   : 442.6
##  3rd Qu.: 451.8
##  Max.   :1080.0
##
## mining info:
##      data ntransactions support confidence
##  data_apr          7582    0.03        0.7
##
##  apriori(data = data_apr, parameter = list(supp = 0.03, conf = 0.7), appearance = list(default = "lhs
```

```
#inspect(proj_rules)
```

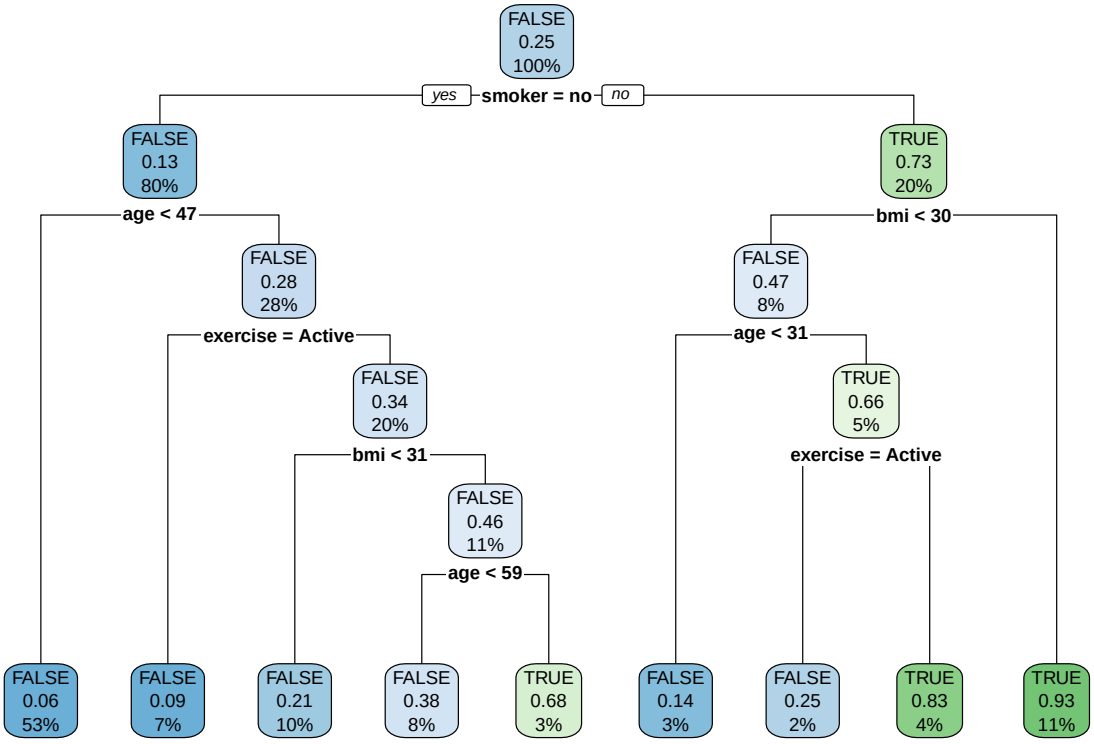
Tree Model

```
proj_rpart <- rpart(expensive ~ age+bmi+children+smoker+hypertension+exercise+yearly_physical, data = Tr
rpart_Pred <- predict(proj_rpart, newdata= TstSet, type= "class")

confusionMatrix(rpart_Pred, TstSet$expensive)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  2199  279
##      TRUE     75  479
##
##                Accuracy : 0.8832
##                  95% CI : (0.8713, 0.8945)
##     No Information Rate : 0.75
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.658
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9670
##             Specificity : 0.6319
##          Pos Pred Value : 0.8874
##          Neg Pred Value : 0.8646
##              Prevalence : 0.7500
##          Detection Rate : 0.7253
##    Detection Prevalence : 0.8173
##       Balanced Accuracy : 0.7995
##
##        'Positive' Class : FALSE
##
```

```
rpart.plot(proj_rpart)
```



Association Rule

```
#asso_Data <- proj_fact[,-7]
#asso_Data[,1:14] <- lapply(asso_Data[,1:7],factor)
#str(asso_Data)
```

```
our_Model3 <- proj_rpart
saveRDS(our_Model3,file="/Users/vedantpatil/Documents/IDS Project/our_Model3.rds")
readRDS(file="/Users/vedantpatil/Documents/IDS Project/our_Model3.rds")
```

```
## n= 4550
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 4550 1137 FALSE (0.75010989 0.24989011)
##    2) smoker=no 3656  481 FALSE (0.86843545 0.13156455)
##      4) age< 46.5 2403  136 FALSE (0.94340408 0.05659592) *
##      5) age>=46.5 1253  345 FALSE (0.72466081 0.27533919)
##       10) exercise=Active 325   28 FALSE (0.91384615 0.08615385) *
##       11) exercise=Not-Active 928  317 FALSE (0.65840517 0.34159483)
##         22) bmi< 31.295 448   94 FALSE (0.79017857 0.20982143) *
##         23) bmi>=31.295 480  223 FALSE (0.53541667 0.46458333)
```

```
##           46) age< 58.5 344  130 FALSE (0.62209302 0.37790698) *
##           47) age>=58.5 136   43 TRUE (0.31617647 0.68382353) *
##    3) smoker=yes 894  238 TRUE (0.26621924 0.73378076)
##      6) bmi< 29.75 379  178 FALSE (0.53034301 0.46965699)
##       12) age< 30.5 139   20 FALSE (0.85611511 0.14388489) *
##       13) age>=30.5 240   82 TRUE (0.34166667 0.65833333)
##         26) exercise=Active 71   18 FALSE (0.74647887 0.25352113) *
##         27) exercise=Not-Active 169   29 TRUE (0.17159763 0.82840237) *
##      7) bmi>=29.75 515   37 TRUE (0.07184466 0.92815534) *
```

```
library(shiny)
library(shinydashboard)
```

```
##
## Attaching package: 'shinydashboard'
```

```
## The following object is masked from 'package:graphics':
##
##     box
```

```
library(shiny)
library(caret)
library(kernlab)
library(e1071)
library(tidyverse)
ui <- fluidPage (
  setBackgroundColor(
  color = c("#F7FBFF", "#2171B5"),
  gradient = "linear",
  direction = "bottom"),
  h1("IDS Project Group 4"),
  hr(),
  br(),
  # div("div creates segments of text with a similar style. This division of text is all blue because I
  h4(p(em("This App gives predictions based on the Rpart model"))),
  hr(),
  #Read the data
  fileInput("upload", label="UPLOAD SAMPLE TEST FILE", accept = c(".csv")),
  #Read the actual (solution) data
  fileInput("upload_Solution", label="UPLOAD SOLUTION FILE", accept = c(".csv")),
  #get a number (how much of the dataframe to show)
  numericInput("n", "Number of Rows", value = 5, min = 1, step = 1),
  #a place to output a table (i.e., a dataframe)
  tableOutput("headForDF"),
  #output the results (for now, just simple text)
  verbatimTextOutput("txt_results", placeholder = TRUE)
)

server <- function(input, output, session) {
  #load a model, do prediction and compute the confusion matrix
  use_model_to_predict <- function(df, df_solution){
    #load the pre-built model, we named it 'out_model.rda')
    my_model <- readRDS("/Users/vedantpatil/Documents/IDS Project/our_Model3.rds")
```

```
    print('enter')
    prd <- predict(my_model, df, type = "class")
    #show how the model performed
    print(prd)
    #glimpse(df)
    #df_solution$isexpensive<- as.factor(df_solution$isexpensive)
    confusionMatrix(prd, as.factor(df_solution$expensive))
  }
  #require an input file, then read a CSV file
  getTestData <- reactive({
    req(input$upload)
    read_csv(input$upload$name)
  })
  #require an the actual values for the prediction (i.e. solution file)
  getSolutionData <- reactive({
    req(input$upload_Solution)
    read_csv(input$upload_Solution$name)
  })
  output$txt_results <- renderPrint({
    #load the data
    dataset <- getTestData()
    dataset_solution <- getSolutionData()
    #load and use the model on the new data
    use_model_to_predict(dataset, dataset_solution)
  })
  #show a few lines of the dataframe
  output$headForDF <- renderTable({
    df <- getTestData()
    head(df, input$n)
  })
}
shinyApp(ui, server)
```