Name - Sarvesh Kumar Mishra Date - 19-12-2018 title: “R Notebook -Exploratory Data Analysis and Multiple Linear Regression on Medical Cost Personal Dataset” Introduction -To give you some background, insurance companies should collect higher premium than the amount paid to the insured person. Due to this, insurance companies invests a lot of time, effort, and money in creating models that accurately predicts health care costs.I will use the the data provided in the Medical Cost Personal Dataset exploring which personal factors are important to predicting medical costs, and then I will perform a linear regression analysis.

-About the File -This dataset consists of 1338 rows.

-Columns:

age: age of primary beneficiary

sex: insurance contractor gender, female, male

bmi: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9

children: Number of children covered by health insurance / Number of dependents

smoker: Smoking

region: the beneficiary’s residential area in the US, northeast, southeast, southwest, northwest.

charges: Individual medical costs billed by health insurance

1.Load packages and dataset

Load libraries

```
#library(ggplot)
library(ggplot2) #visualization
```

```
## Warning: package 'ggplot2' was built under R version 3.5.1
```

```
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 3.5.1
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(relaimpo)
```

```
## Warning: package 'relaimpo' was built under R version 3.5.1
```

```
## Loading required package: MASS
```

```
## Loading required package: boot
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
## Loading required package: survey
```

```
## Warning: package 'survey' was built under R version 3.5.1
```

```
## Loading required package: grid
```

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

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:boot':
##
##     aml
```

```
##
## Attaching package: 'survey'
```

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

```
## Loading required package: mitools
```

```
## Warning: package 'mitools' was built under R version 3.5.1
```

```
## This is the global version of package relaimpo.
```

```
## If you are a non-US user, a version with the interesting additional metric pmvd is available
```

```
## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.
```

```
library(readr) #read in the data
```

```
## Warning: package 'readr' was built under R version 3.5.1
```

```
library(corrplot) #visualization of correlation
```

```
## Warning: package 'corrplot' was built under R version 3.5.1
```

```
## corrplot 0.84 loaded
```

```
library(ggcorrplot) #visualization of correlation
```

```
## Warning: package 'ggcorrplot' was built under R version 3.5.1
```

```
library(reshape2) #melt function
library(dplyr) #used for data transformations
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse) #used for data transformations
```

```
## Warning: package 'tidyverse' was built under R version 3.5.1
```

```
## -- Attaching packages ----------------------------------------------------------------------------------- tidyve
rse 1.2.1 --
```

```
## v tibble  1.4.2     v purrr   0.2.5
## v tidyr   0.8.1     v stringr 1.3.1
## v tibble  1.4.2     v forcats 0.3.0
```

```
## Warning: package 'forcats' was built under R version 3.5.1
```

```
## -- Conflicts -------------------------------------------------------------------------------------- tidyverse_co
nflicts() --
## x psych::%+%()    masks ggplot2::%+%()
## x psych::alpha()  masks ggplot2::alpha()
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
library(naniar)
```

```
## Warning: package 'naniar' was built under R version 3.5.1
```

```
library(Amelia) # Missing Data: Missings Map
```

```
## Warning: package 'Amelia' was built under R version 3.5.1
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2018 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
library(caTools) # Prediction: Splitting Data
```

```
## Warning: package 'caTools' was built under R version 3.5.1
```

```
library(car) # Prediction: Checking Multicollinearity
```

```
## Warning: package 'car' was built under R version 3.5.1
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 3.5.1
```

```
##
## Attaching package: 'car'
```

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

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

```
## The following object is masked from 'package:boot':
##
##     logit
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 3.5.1
```

```
##
## Attaching package: 'GGally'
```

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

```
library(corpcor)
library(mctest)
library(ppcor)
```

```
## Warning: package 'ppcor' was built under R version 3.5.1
```

Read the insurance dataset

```
insurance <- read.csv("E:/Git/sarveshmishra1/machinelearning/LinearRegression/insurance.csv")
```

Verify the data

```
head(insurance, n=5)
```

```
##   age    sex    bmi children smoker    region   charges
## 1  19 female 27.900        0    yes southwest 16884.924
## 2  18   male 33.770        1     no southeast  1725.552
## 3  28   male 33.000        3     no southeast  4449.462
## 4  33   male 22.705        0     no northwest 21984.471
## 5  32   male 28.880        0     no northwest  3866.855
```

Verify the column structure and values

```
str(insurance)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
```

The total number of observation is - 1338 The data set has 7 features. We will predict the charges

2.Exploratory Data Analysis

Summarise the dataset

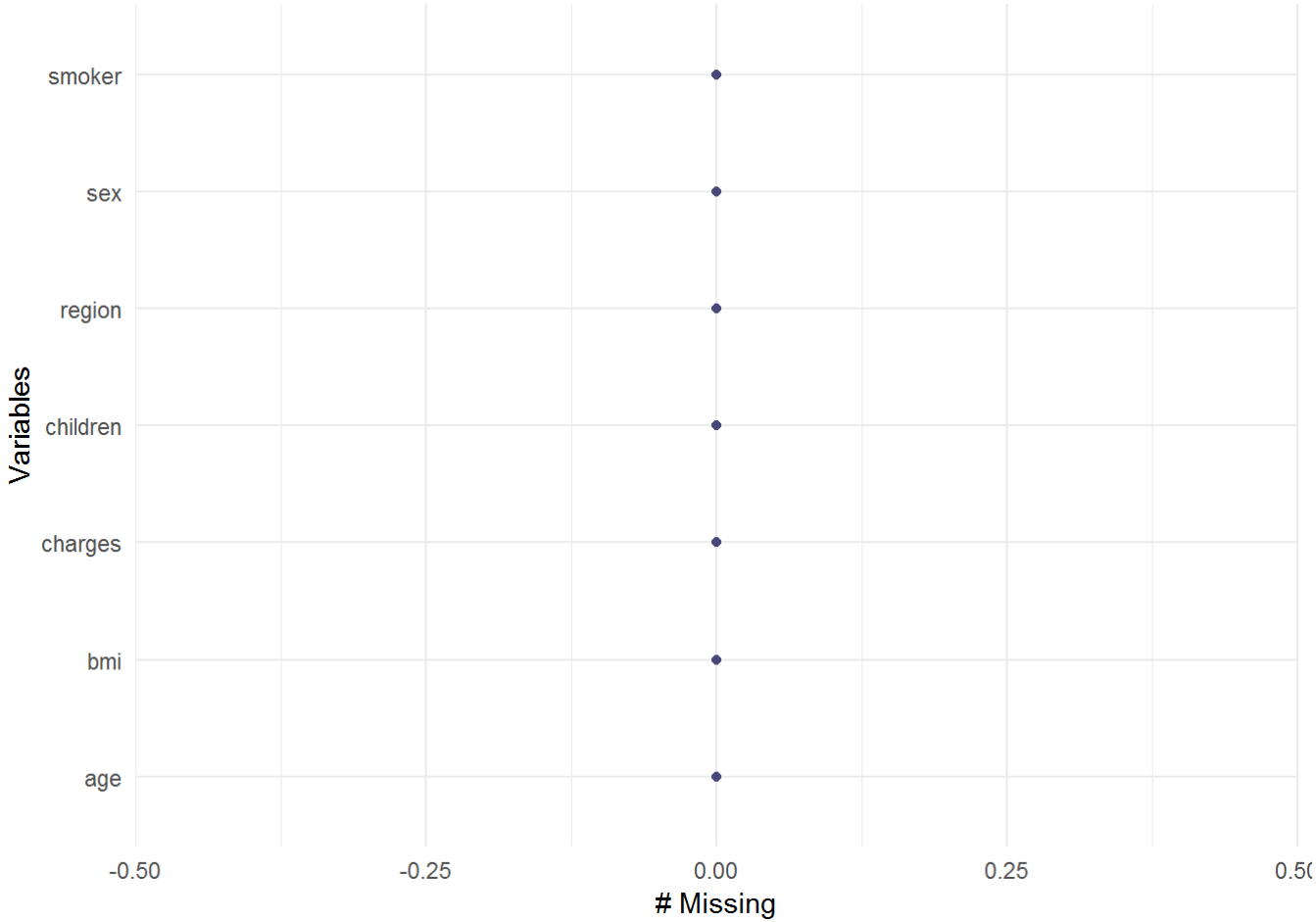
```
summary(insurance)
```

```
##       age            sex           bmi           children     smoker
##  Min.   :18.00   female:662   Min.   :15.96   Min.   :0.000   no :1064
##  1st Qu.:27.00   male  :676   1st Qu.:26.30   1st Qu.:0.000   yes: 274
##  Median :39.00                Median :30.40   Median :1.000
##  Mean   :39.21                Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                Max.   :53.13   Max.   :5.000
##        region       charges
##  northeast:324   Min.   : 1122
##  northwest:325   1st Qu.: 4740
##  southeast:364   Median : 9382
##  southwest:325   Mean   :13270
##                  3rd Qu.:16640
##                  Max.   :63770
```

The dataset does not have missing values in any of the features. The average medical cost os 13,270 USD and median of 9382 USD.

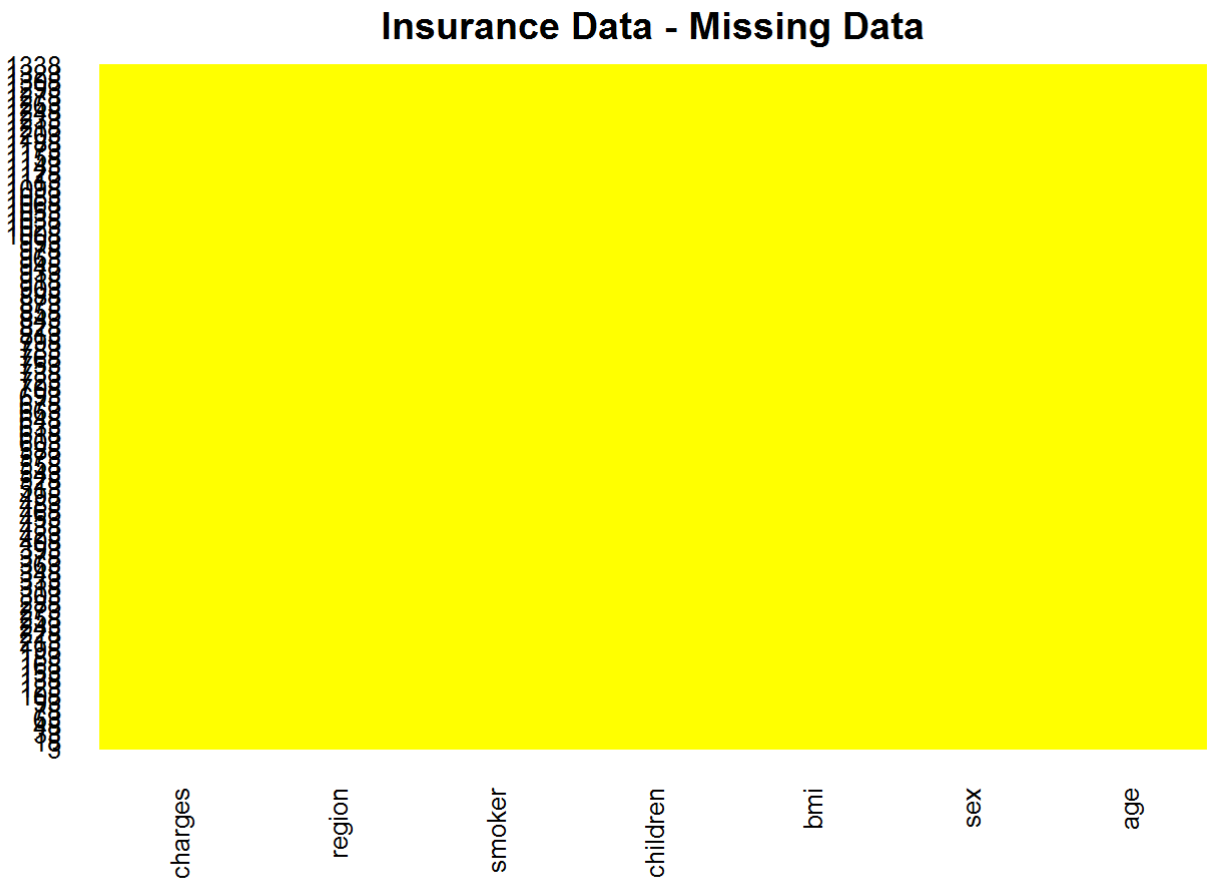
Check the missing values using visualization

```
gg_miss_var(insurance)
```



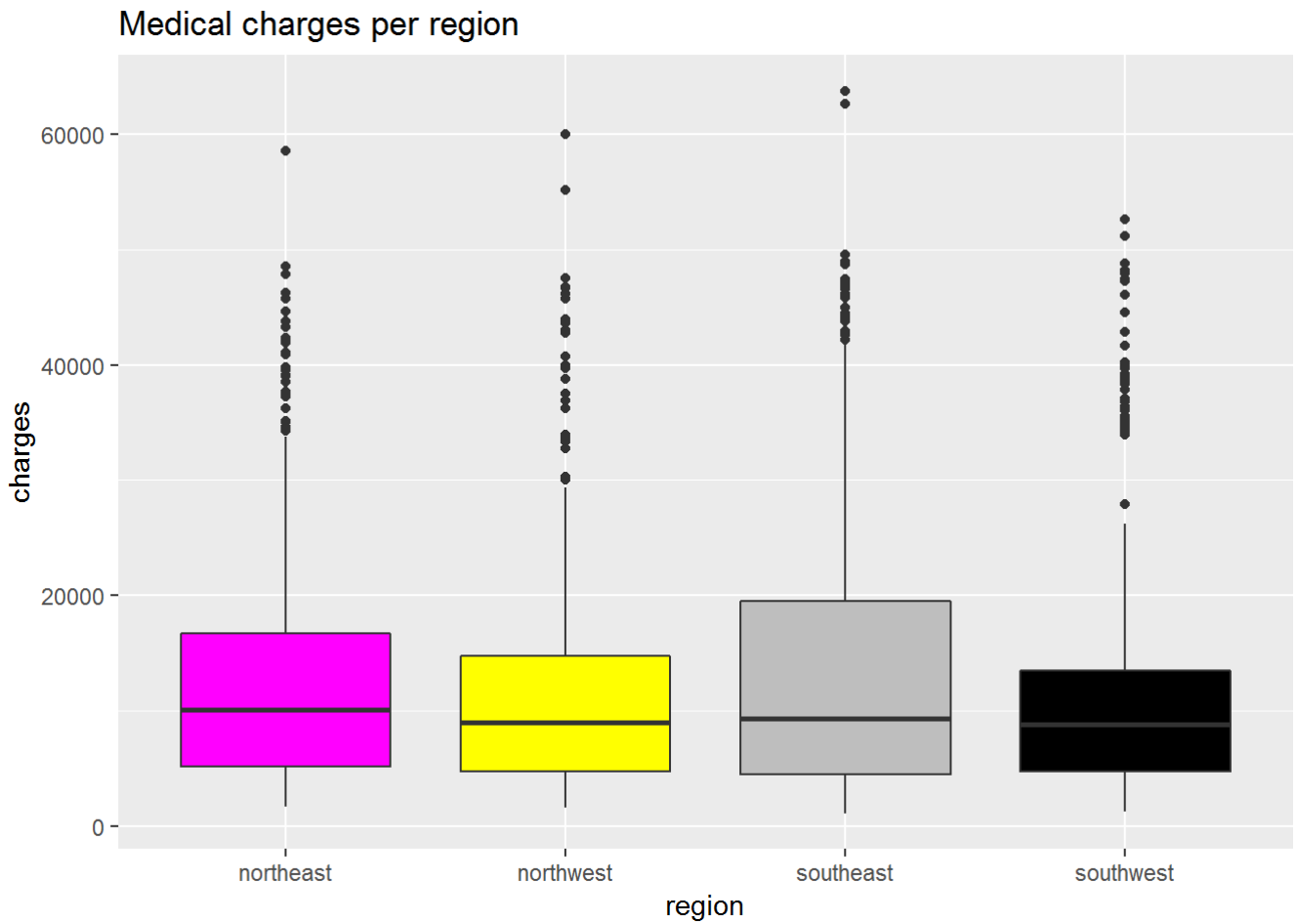
Visualization for missing data

```
missmap(insurance, main = "Insurance Data - Missing Data", col = c("Red", "Yellow"), legend=FALSE)
```



Charges per region

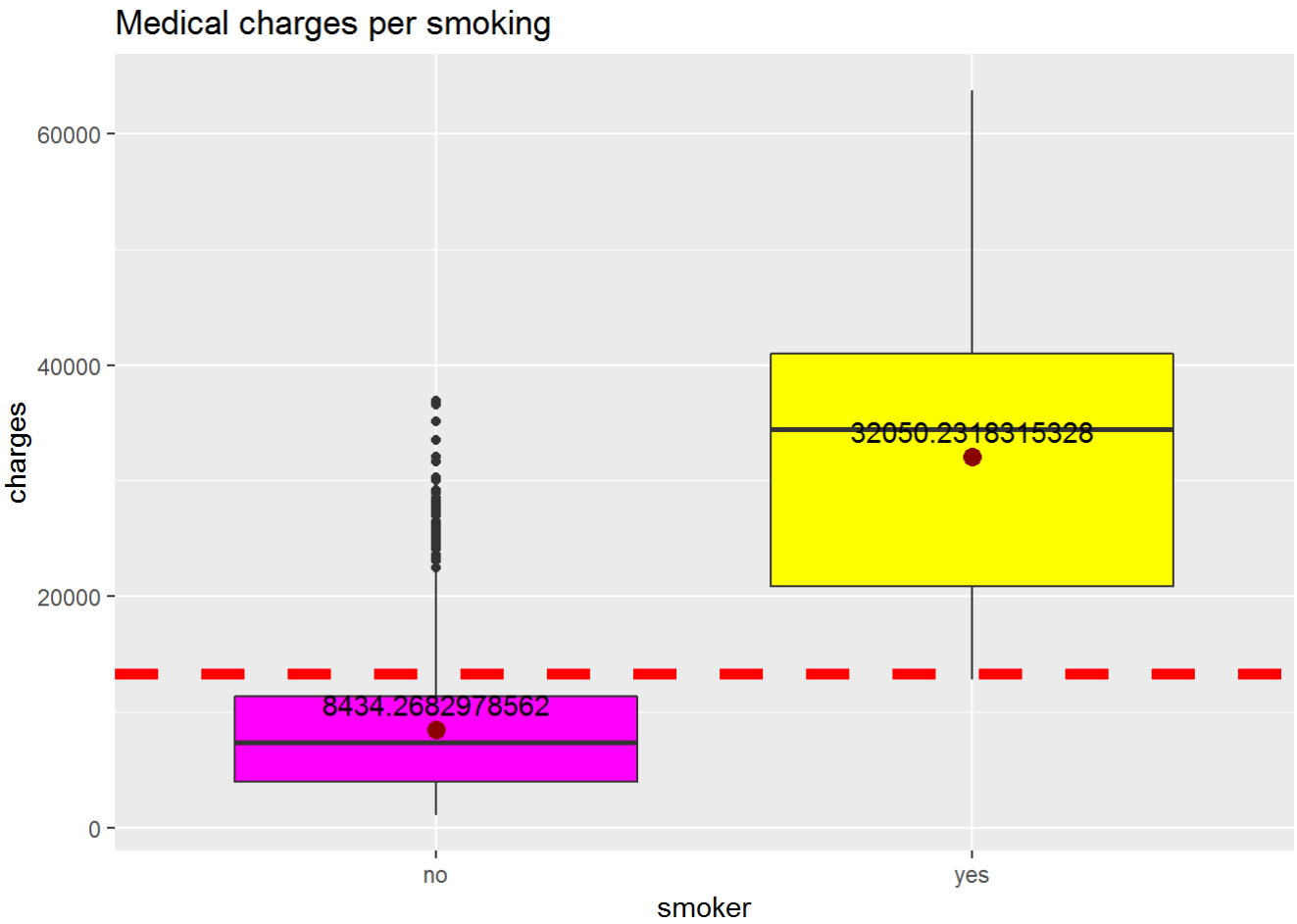
```
ggplot(data = insurance, aes(x=region, y=charges)) +
  geom_boxplot(fill = c(6:9))+
  ggtitle("Medical charges per region")
```



The medical cost is almost same of all the region.

Charges based on smoking

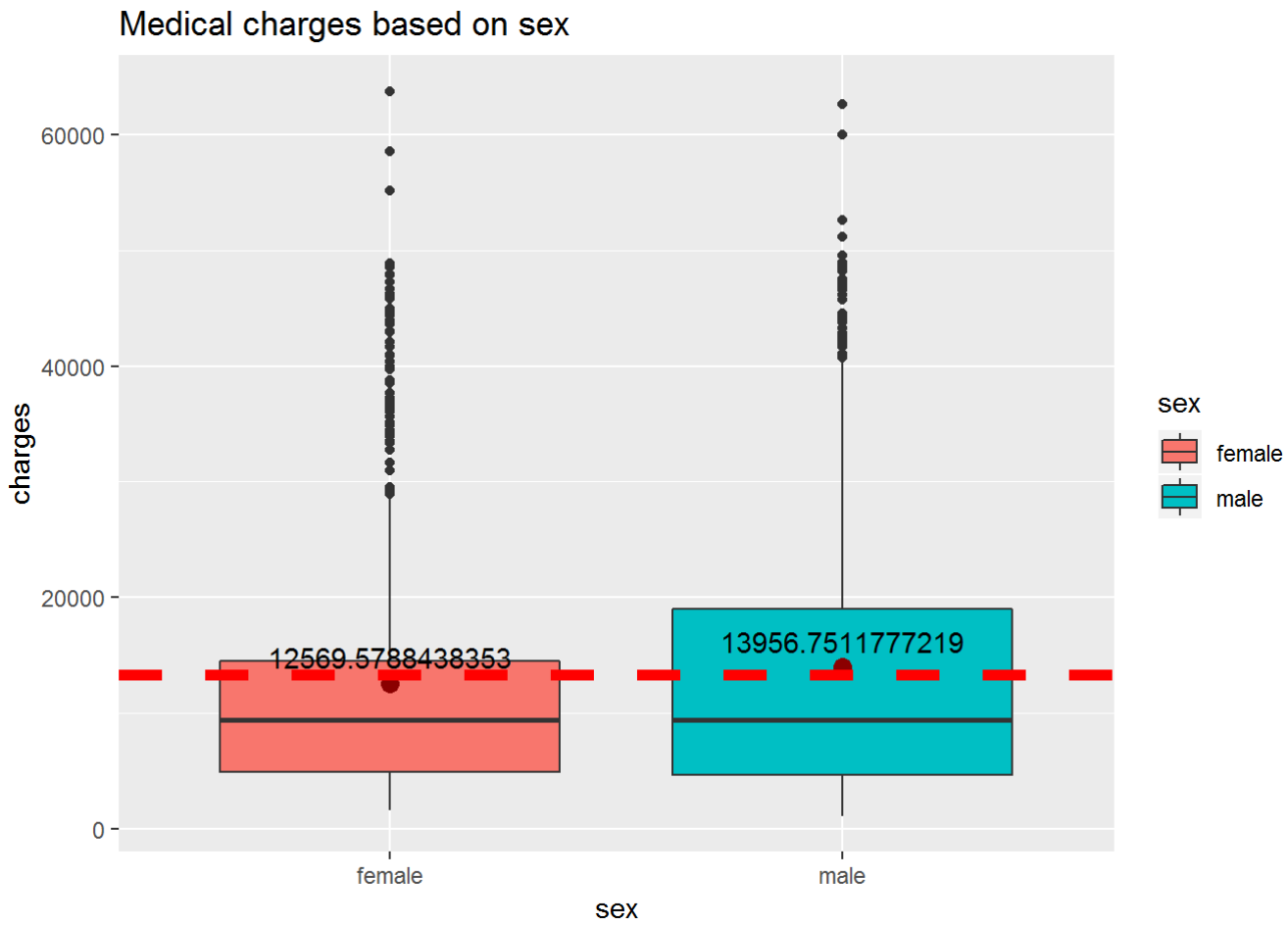
```
fun_mean <- function(x){
  return(data.frame(y=mean(x),label=mean(x,na.rm=T)))}

ggplot(data = insurance, aes(x=smoker, y=charges)) +
  geom_boxplot(fill = c(6:7))+
  stat_summary(fun.y = mean, geom="point",colour="darkred", size=3) +
  stat_summary(fun.data = fun_mean, geom="text", vjust=-0.7) +
  geom_hline(aes(yintercept=mean(charges, na.rm=T)), colour = "red", linetype="dashed", size=2) +
  ggtitle("Medical charges per smoking")
```



The medical cost for smokers are far more as compared to non smoker. We can see that mean charges for smoker is almost 4 time of non smoker.

Medical Charges based on gender

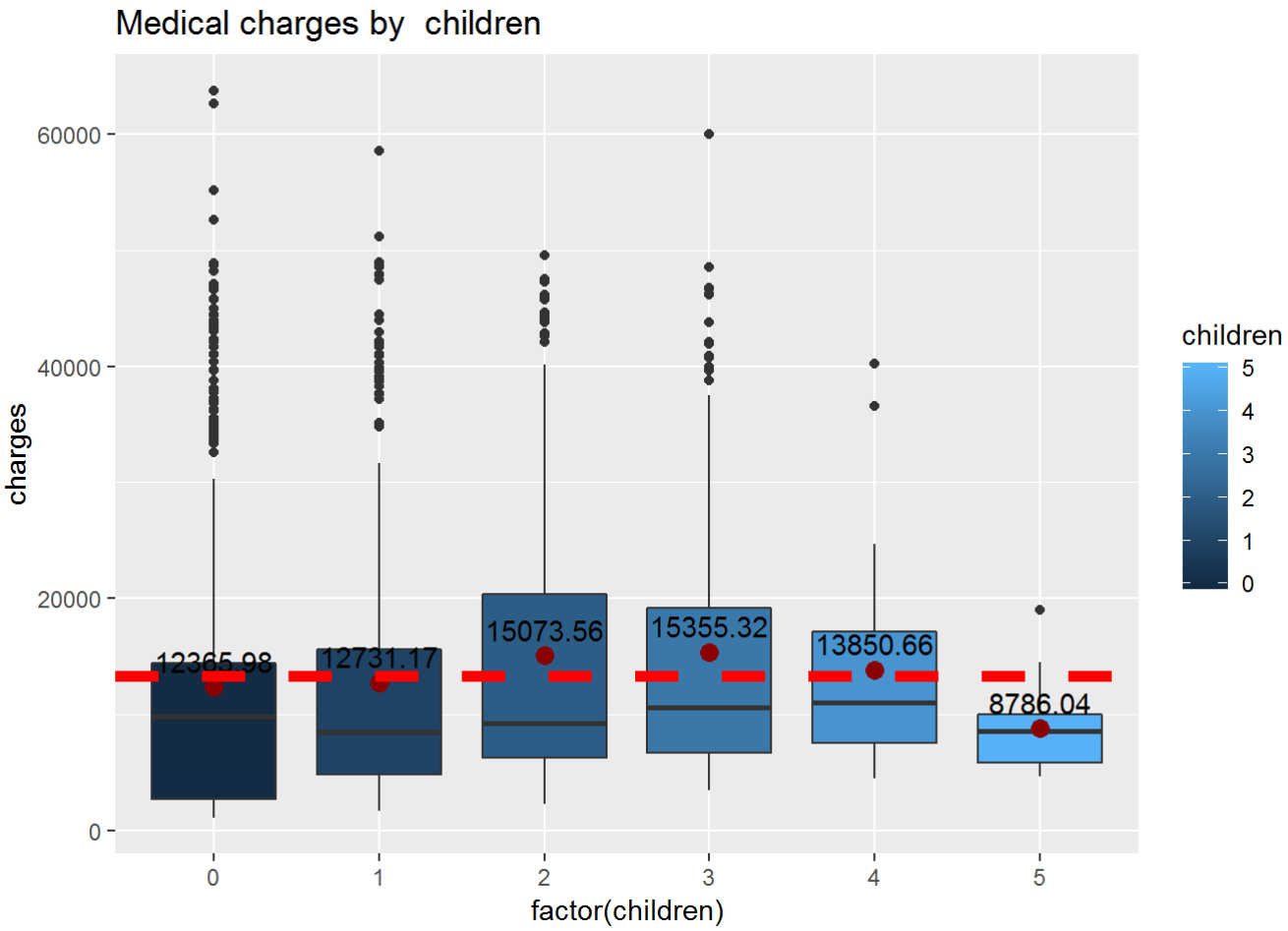
```
fun_mean <- function(x){
  return(data.frame(y=mean(x),label=mean(x,na.rm=T)))}

ggplot(data = insurance, aes(x=sex, y=charges)) +
  geom_boxplot(aes(fill=sex))+
  stat_summary(fun.y = mean, geom="point",colour="darkred", size=3) +
  stat_summary(fun.data = fun_mean, geom="text", vjust=-0.7) +
  geom_hline(aes(yintercept=mean(charges, na.rm=T)), colour = "red", linetype="dashed", size=2) +
  ggtitle("Medical charges based on sex")
```



The medical charges is almost same based on sex, but it is greater for male as compared to female.

Charges based on number of children

```
fun_mean <- function(x){
  return (round((data.frame(y=mean(x),label=mean(x,na.rm=T))),2))}

ggplot(data = insurance, aes(x=factor(children), y=charges)) +
  geom_boxplot(aes(fill=children))+
  stat_summary(fun.y = mean, geom="point",colour="darkred", size=3) +
  stat_summary(fun.data = fun_mean, geom="text", vjust=-0.7) +
  geom_hline(aes(yintercept=mean(charges, na.rm=T)), colour = "red", linetype="dashed", size=2)+
  ggtitle("Medical charges by  children")
```



The children count has no impact on charges but the family with 5 children as low average cost.

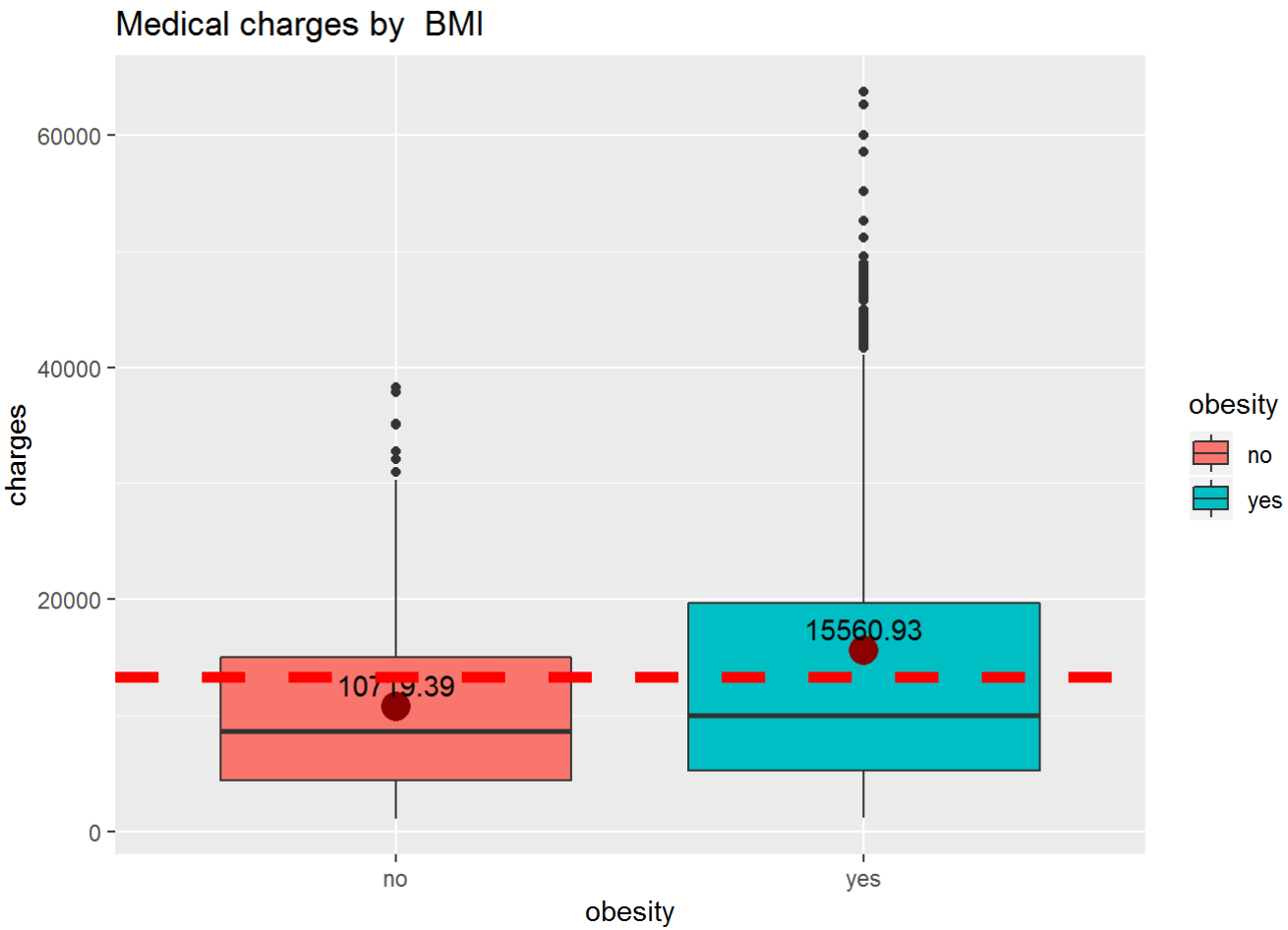
Add the variable for BMI to define the threshold for obesity

```
insurance$obesity <- ifelse(insurance$bmi > 30, "yes", "no")
head(insurance$obesity, n=2)
```

```
## [1] "no"  "yes"
```

Medical cost by BMI

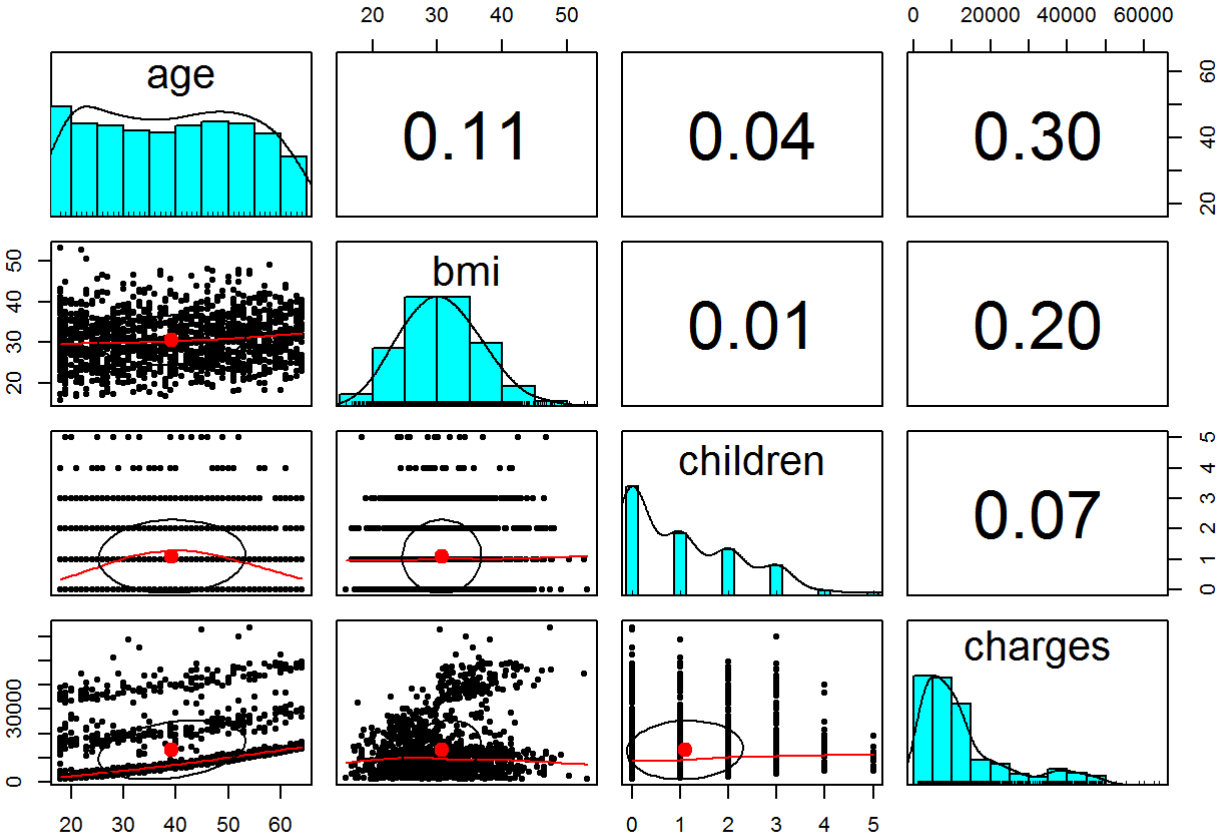
```
fun_mean <- function(x){
  return (round((data.frame(y=mean(x),label=mean(x,na.rm=T))),2))}

ggplot(data = insurance, aes(x=obesity, y=charges)) +
  geom_boxplot(aes(fill=obesity))+
  stat_summary(fun.y = mean, geom="point",colour="darkred", size=5) +
  stat_summary(fun.data = fun_mean, geom="text", vjust=-0.5) +

  #geom_text( aes(label = charges, y = charges + 0.08)) +

  geom_hline(aes(yintercept=mean(charges, na.rm=T)), colour = "red", linetype="dashed", size=2)+
  ggtitle("Medical charges by  BMI")
```



The obesity does not play any role in medical cost, but charges is almost 50% more for more obese individuals.

check the collinearity test between numeric features.

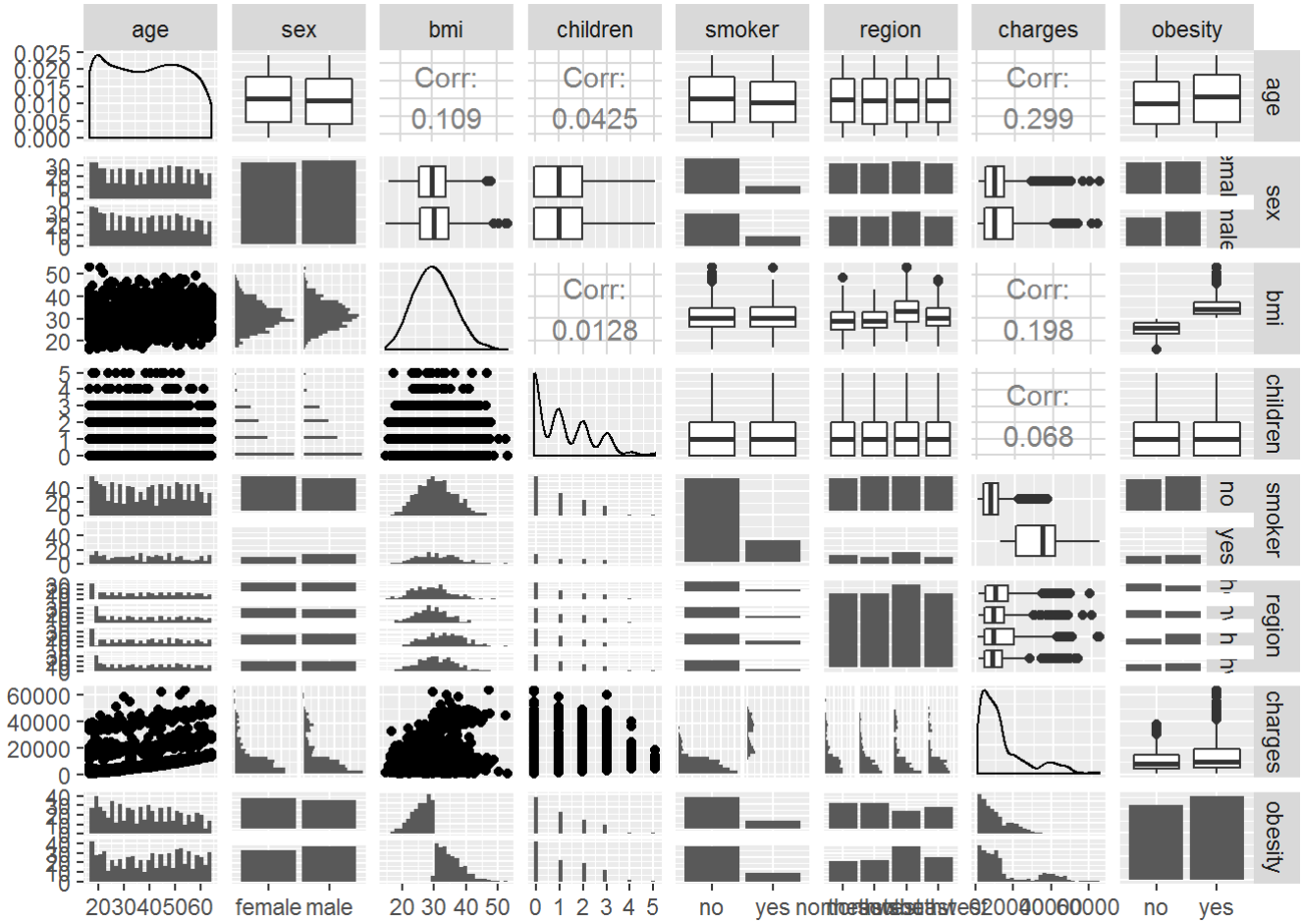
```
pairs.panels(insurance[c("age", "bmi", "children", "charges")])
```



#Pair correlation

```
ggpairs(insurance)
```

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



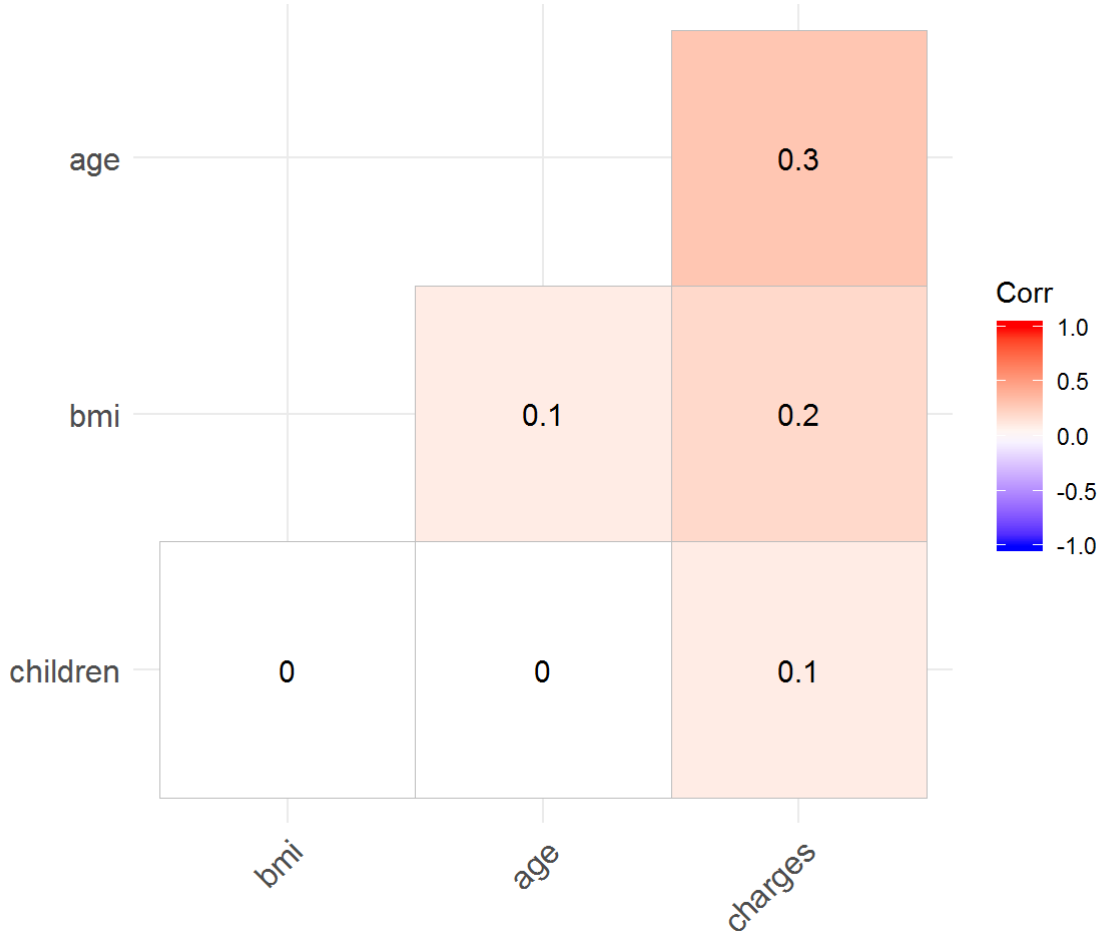
Find correlation among numeric features

```
cor(insurance[sapply(insurance, is.numeric)])
```

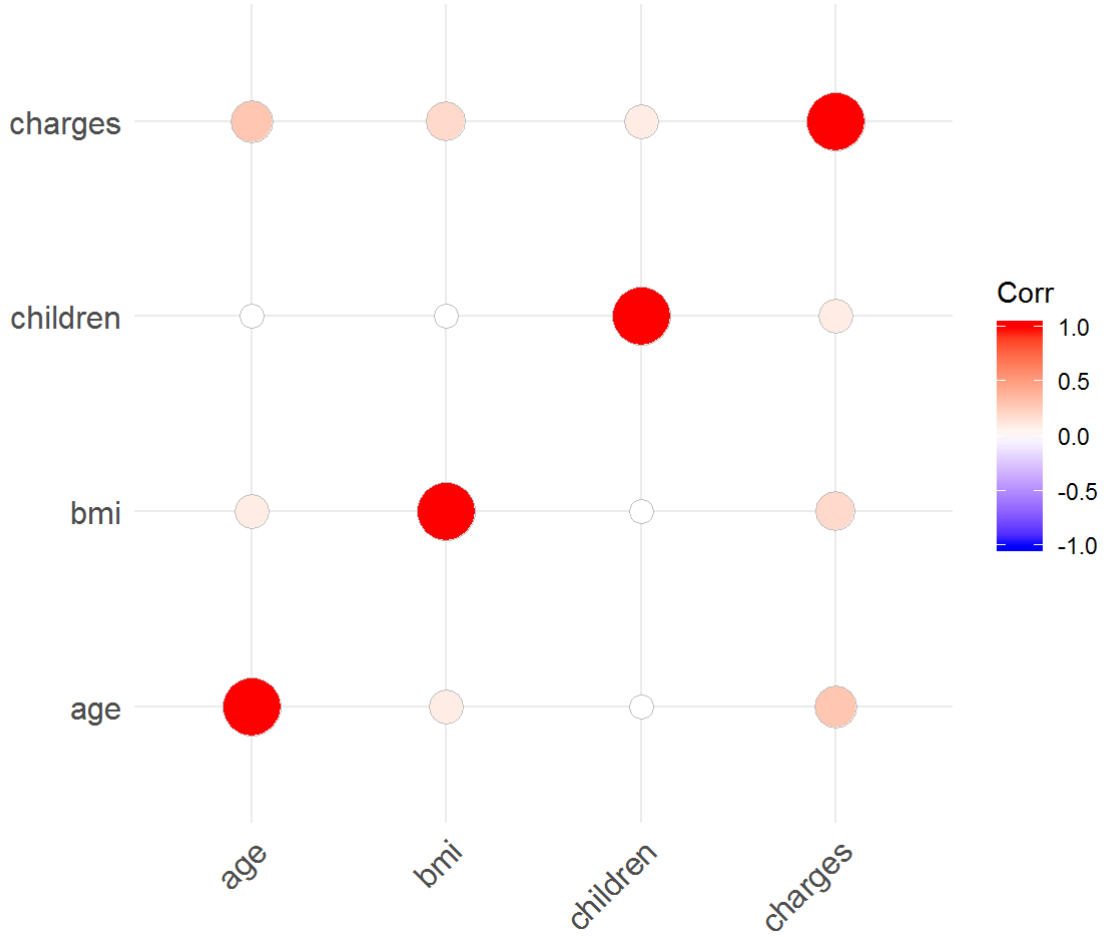
```
##                age       bmi   children    charges
## age      1.0000000 0.1092719 0.04246900 0.29900819
## bmi      0.1092719 1.0000000 0.01275890 0.19834097
## children 0.0424690 0.0127589 1.00000000 0.06799823
## charges  0.2990082 0.1983410 0.06799823 1.00000000
```

None of the above correlation pairs are above 0.75, hence features are not correlated

```
corr <- round(cor(insurance[sapply(insurance, is.numeric)]), 1)
ggcorrplot(corr, hc.order = TRUE, type = "lower",
   lab = TRUE)
```




```
ggcorrplot(corr, method = "circle")
```



We can see that the highest correlation is 0.3, the correlation value above .7 or .8, can be considered as features are correlated.

```
str(insurance)
```

```
## 'data.frame':    1338 obs. of  8 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
##  $ obesity : chr  "no" "yes" "yes" "no" ...
```

Farrar - Glauber Test The ‘mctest’ package in R provides the Farrar-Glauber test and other relevant tests for multicollinearity. There are two functions viz. ‘omcdiag’ and ‘imcdiag’ under ‘mctest’ package in R which will provide the overall and individual diagnostic checking for multicollinearity respectively

```
omcdiag(insurance[,c("age", "bmi", "children", "charges")],insurance$charges)
```

```
## Warning in summary.lm(lm(y ~ x)): essentially perfect fit: summary may be
## unreliable
```

```
## Warning in summary.lm(lm(y ~ x[, i])): essentially perfect fit: summary may
## be unreliable
```

```
##
## Call:
## omcdiag(x = insurance[, c("age", "bmi", "children", "charges")],
##     y = insurance$charges)
##
##
## Overall Multicollinearity Diagnostics
##
##                        MC Results detection
## Determinant |X'X|:         0.8678         0
## Farrar Chi-Square:       189.1647         1
## Red Indicator:             0.1567         0
## Sum of Lambda Inverse:     4.2875         0
## Theil's Method:           -2.7402         0
## Condition Number:         15.0597         0
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

The value of the standardized determinant is found to be 0.8678 which is very small. The calculated value of the Chi-square test statistic is found to be 189.1647 and it is not significant thereby implying the non presence of multicollinearity in the model specification. We can go for the next step of Farrar - Glauber test (F - test) to show that there is no multicollinearity.

```
imcdiag(insurance[,c("age", "bmi", "children", "charges")],insurance$charges)
```

```
## Warning in summary.lm(lm(y ~ x)): essentially perfect fit: summary may be
## unreliable
```

```
## Warning in summary.lm(lm(y ~ x[, i])): essentially perfect fit: summary may
## be unreliable
```

```
## Warning in summary.lm(lm(y ~ x)): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## imcdiag(x = insurance[, c("age", "bmi", "children", "charges")],
##     y = insurance$charges)
##
##
## All Individual Multicollinearity Diagnostics Result
##
##             VIF    TOL      Wi      Fi Leamer CVIF Klein
## age      1.1019 0.9075 45.3235 68.0362 0.9526    0     0
## bmi      1.0439 0.9579 19.5358 29.3257 0.9787    0     0
## children 1.0052 0.9948  2.3090  3.4661 0.9974    0     0
## charges  1.1365 0.8799 60.6928 91.1074 0.9380    0     0
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## children , coefficient(s) are non-significant may be due to multicollinearity
##
## R-square of y on all x: 1
##
## * use method argument to check which regressors may be the reason of collinearity
## ===================================
```

The VIF, TOL and Wi columns provide the diagnostic output for variance inflation factor, tolerance and Farrar-Glauber F-test respectively. Above shows that, there is no corelation between features.

Again by looking at the partial correlation coefficient matrix among the variables, it is also clear that none of the features are high.

```
X<-insurance[,c("age", "bmi", "children", "charges")]
cor2pcor(cov(insurance[c("age", "bmi", "children", "charges")]))
```

```
##           [,1]         [,2]         [,3]       [,4]
## [1,] 1.00000000  0.053454963  0.023325183 0.28277324
## [2,] 0.05345496  1.000000000 -0.001989992 0.17448165
## [3,] 0.02332518 -0.001989992  1.000000000 0.05746056
## [4,] 0.28277324  0.174481649  0.057460560 1.00000000
```

In R, there are several packages for getting the partial correlation coefficients along with the t- test for checking their significance level. We'll the 'ppcor' package to compute the partial correlation coefficients along with the t-statistic and corresponding p-values.

```
pcor(insurance[,c("age", "bmi", "children", "charges")], method = "pearson")
```

```
## $estimate
##                 age          bmi     children    charges
## age      1.00000000  0.053454963  0.023325183 0.28277324
## bmi      0.05345496  1.000000000 -0.001989992 0.17448165
## children 0.02332518 -0.001989992  1.000000000 0.05746056
## charges  0.28277324  0.174481649  0.057460560 1.00000000
##
## $p.value
##                   age          bmi   children      charges
## age      0.000000e+00 5.077008e-02 0.39427824 5.533923e-26
## bmi      5.077008e-02 0.000000e+00 0.94206965 1.354882e-10
## children 3.942782e-01 9.420697e-01 0.00000000 3.572625e-02
## charges  5.533923e-26 1.354882e-10 0.03572625 0.000000e+00
##
## $statistic
##                age         bmi    children   charges
## age       0.000000  1.95518257  0.85216001 10.767454
## bmi       1.955183  0.00000000 -0.07268252  6.472040
## children  0.852160 -0.07268252  0.00000000  2.102161
## charges  10.767454  6.47203994  2.10216070  0.000000
##
## $n
## [1] 1338
##
## $gp
## [1] 2
##
## $method
## [1] "pearson"
```

Above shows that none of the pvalue and t-value are high.

Show the p-value of Chi Square tests

Convert features into factor

```
#insurance$children <- factor(insurance$children)
#insurance$obesity <- factor(insurance$obesity)
#insurance$sex <- factor(insurance$sex)
```

```
#(ncol(head(insurance$sex, n=5)))
```

```
#This is used when x and y has different level of factor
# make up some data
set.seed(32892917)
#mydata <- data.frame(group=as.factor(insurance$sex),race=as.factor(insurance$children))

# look at the table:
# (mytab <- with(mydata,table(group,race)) )
#chisq.test(mytab)$p.value
#sc = chisq.test(insurance$Sex, insurance$children)$p.value
```

```
#mapply(function(x, y) chisq.test(x, y)$p.value, insurance[, c('sex','children','smoker','region','obesity')], #MoreArgs=lis
t(insurance[, c('sex','children','smoker','region','obesity')]))
```

```
#sc = chisq.test(insurance$Sex, insurance$children)$p.value
#ss = chisq.test(insurance$Sex, insurance$smoker)$p.value
#sr = chisq.test(insurance$Sex, insurance$region)$p.value
#so = chisq.test(insurance$Sex, insurance$obesity)$p.value
#cs = chisq.test(insurance$children, insurance$smoker)$p.value
#cr = chisq.test(insurance$children, insurance$region)$p.value
#co = chisq.test(insurance$children, insurance$obesity)$p.value
#sr = chisq.test(insurance$smoker, insurance$region)$p.value
#so = chisq.test(insurance$smoker, insurance$obesity)$p.value

#ro = chisq.test(insurance$region, insurance$obesity)$p.value

sc =chisq.test(with(data.frame(group=as.factor(insurance$sex),race=as.factor(insurance$children)),
               table(group,race)))$p.value

ss =chisq.test(with(data.frame(group=as.factor(insurance$sex),race=as.factor(insurance$smoker)),
               table(group,race)))$p.value

sr =chisq.test(with(data.frame(group=as.factor(insurance$sex),race=as.factor(insurance$region)),
               table(group,race)))$p.value

so =chisq.test(with(data.frame(group=as.factor(insurance$sex),race=as.factor(insurance$obesity)),
               table(group,race)))$p.value

cs =chisq.test(with(data.frame(group=as.factor(insurance$children),race=as.factor(insurance$smoker)),
               table(group,race)))$p.value
```

```
## Warning in chisq.test(with(data.frame(group = as.factor(insurance
## $children), : Chi-squared approximation may be incorrect
```

```
cr =chisq.test(with(data.frame(group=as.factor(insurance$children),race=as.factor(insurance$region)),
               table(group,race)))$p.value
```

```
## Warning in chisq.test(with(data.frame(group = as.factor(insurance
## $children), : Chi-squared approximation may be incorrect
```

```
co =chisq.test(with(data.frame(group=as.factor(insurance$children),race=as.factor(insurance$obesity)),
               table(group,race)))$p.value

sr =chisq.test(with(data.frame(group=as.factor(insurance$smoker),race=as.factor(insurance$region)),
               table(group,race)))$p.value

so =chisq.test(with(data.frame(group=as.factor(insurance$smoker),race=as.factor(insurance$obesity)),
               table(group,race)))$p.value

ro =chisq.test(with(data.frame(group=as.factor(insurance$region),race=as.factor(insurance$obesity)),
               table(group,race)))$p.value


cormatrix = matrix(c(0, sc, ss, sr, so,
                     sc, 0, cs, cr, co,
                     ss, cs, 0, sr, so,
                     sr, cr, sr, 0, ro,
                     so, co, so, ro, 0),
                   5, 5, byrow = TRUE)


cormatrix
```

```
##             [,1]      [,2]        [,3]         [,4]         [,5]
## [1,] 0.000000000 0.9809804 0.006548144 6.171955e-02 1.000000e+00
## [2,] 0.980980392 0.0000000 0.229125551 5.428264e-01 4.392379e-01
## [3,] 0.006548144 0.2291256 0.000000000 6.171955e-02 1.000000e+00
## [4,] 0.061719548 0.5428264 0.061719548 0.000000e+00 8.897810e-10
## [5,] 1.000000000 0.4392379 1.000000000 8.897810e-10 0.000000e+00
```

```
colnames(cormatrix) = c("Sex", "children", "smoker", "region", "obesity")
row.names(cormatrix) = c("Sex", "children", "smoker", "region", "obesity")
cormatrix
```

```
##                 Sex  children      smoker       region      obesity
## Sex      0.000000000 0.9809804 0.006548144 6.171955e-02 1.000000e+00
## children 0.980980392 0.0000000 0.229125551 5.428264e-01 4.392379e-01
## smoker   0.006548144 0.2291256 0.000000000 6.171955e-02 1.000000e+00
## region   0.061719548 0.5428264 0.061719548 0.000000e+00 8.897810e-10
## obesity  1.000000000 0.4392379 1.000000000 8.897810e-10 0.000000e+00
```

We can conclude following 1. The features are correlated if p < 0.05 2. Correlation between sex and smoker (0.006), sex and region, sex and obesity 3. smoker, region ; smoker, obesity 4. region, sex; region, smoker; region, obesity 5. obesity, region;

We will verify above multicollinearity below during model build.

Create training and test set

```
set.seed(42)
#Suffle the row
rows <- sample(nrow(insurance))

insurance <- insurance[rows, ]

#Split the 80/20 train and test data

split <- round(nrow(insurance) * 0.80  )
insurance_train <- insurance[1 : split, ]
insurance_test <- insurance[(split +1) : nrow(insurance),]
```

Create Model

```
lm.fit <- lm(charges ~ . , data = insurance_train)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = charges ~ ., data = insurance_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -12185  -3456   -109   1663  27905
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -7963.72    1429.25  -5.572 3.19e-08 ***
## age               258.51      13.24  19.528  < 2e-16 ***
## sexmale           143.22     371.07   0.386 0.699595
## bmi               153.60      51.77   2.967 0.003072 **
## children          520.81     154.01   3.382 0.000747 ***
## smokeryes       24162.03     458.97  52.644  < 2e-16 ***
## regionnorthwest  -529.21     528.97  -1.000 0.317322
## regionsoutheast  -879.85     535.74  -1.642 0.100823
## regionsouthwest -1288.64     535.94  -2.404 0.016368 *
## obesityyes       2929.22     619.73   4.727 2.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6045 on 1060 degrees of freedom
## Multiple R-squared:  0.7627, Adjusted R-squared:  0.7607
## F-statistic: 378.5 on 9 and 1060 DF,  p-value: < 2.2e-16
```

Using the best model by selecting AIC

Step() will run the model for each variable iteratively and give the best model on top as per ranking on AIC

```
lm.fit <- step(lm.fit)
```

```
## Start:  AIC=18642.88
## charges ~ age + sex + bmi + children + smoker + region + obesity
##
##            Df  Sum of Sq        RSS   AIC
## - sex       1 5.4437e+06 3.8739e+10 18641
## <none>                   3.8734e+10 18643
## - region    3 2.2557e+08 3.8959e+10 18643
## - bmi       1 3.2172e+08 3.9055e+10 18650
## - children  1 4.1788e+08 3.9151e+10 18652
## - obesity   1 8.1635e+08 3.9550e+10 18663
## - age       1 1.3935e+10 5.2668e+10 18970
## - smoker    1 1.0127e+11 1.4000e+11 20016
##
## Step:  AIC=18641.03
## charges ~ age + bmi + children + smoker + region + obesity
##
##            Df  Sum of Sq        RSS   AIC
## <none>                   3.8739e+10 18641
## - region    3 2.2552e+08 3.8964e+10 18641
## - bmi       1 3.2256e+08 3.9062e+10 18648
## - children  1 4.1974e+08 3.9159e+10 18651
## - obesity   1 8.1975e+08 3.9559e+10 18661
## - age       1 1.3929e+10 5.2668e+10 18968
## - smoker    1 1.0180e+11 1.4054e+11 20018
```

```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi + children + smoker + region +
##     obesity, data = insurance_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12225.8  -3463.2    -80.8   1627.1  27966.1
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -7898.95    1418.79  -5.567 3.28e-08 ***
## age               258.41      13.23  19.532  < 2e-16 ***
## bmi               153.80      51.74   2.972 0.003023 **
## children          521.89     153.92   3.391 0.000723 ***
## smokeryes       24173.59     457.81  52.803  < 2e-16 ***
## regionnorthwest  -531.04     528.74  -1.004 0.315439
## regionsoutheast  -879.93     535.53  -1.643 0.100657
## regionsouthwest -1288.86     535.73  -2.406 0.016306 *
## obesityyes       2934.58     619.33   4.738 2.45e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6042 on 1061 degrees of freedom
## Multiple R-squared:  0.7627, Adjusted R-squared:  0.7609
## F-statistic: 426.2 on 8 and 1061 DF,  p-value: < 2.2e-16
```

We can see that std error is very high for region, smoker and obesity, importance to be verified with VIF test. R-Squared = 0.76

Find VIF

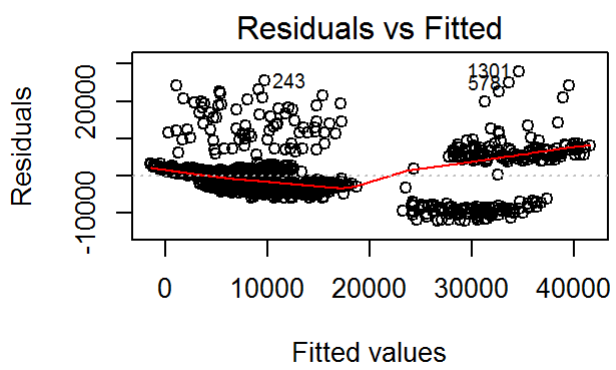
```
vif(lm.fit)
```

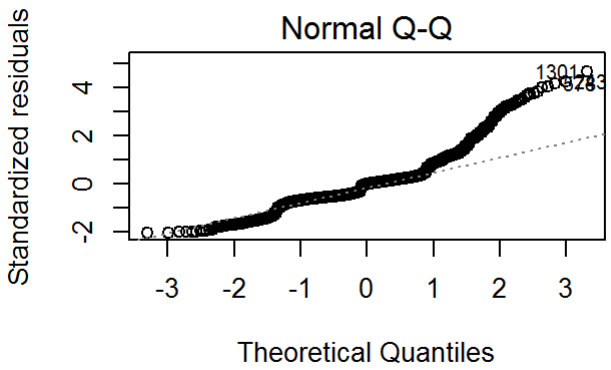
```
##              GVIF Df GVIF^(1/(2*Df))
## age      1.016560  1        1.008246
## bmi      2.975868  1        1.725070
## children 1.004617  1        1.002306
## smoker   1.006575  1        1.003282
## region   1.111911  3        1.017837
## obesity  2.801905  1        1.673889
```

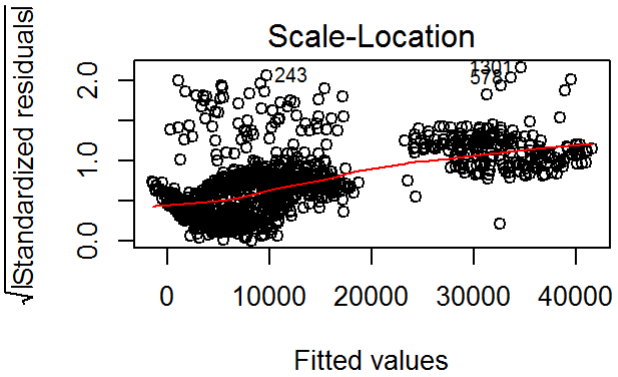
The above VIF shows that, the general VIF's are below 5, hence the variables are not correlated. Also smoker, region and obesity contribute to model

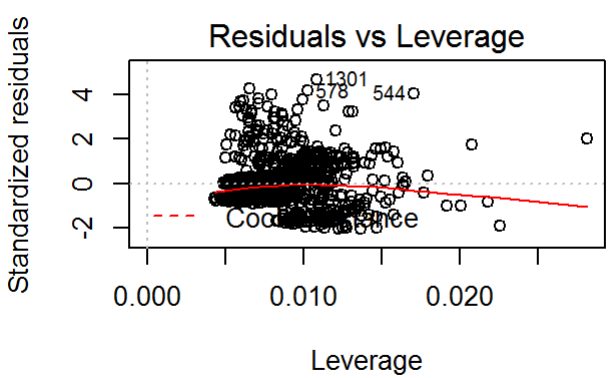
Further we can plot the model diagnostic checking for other problems such as normality of error term, heteroscedasticity etc.

```
par(mfrow=c(2,2))
plot(lm.fit)
```









We can consider the synergy effect of variables ie between obesity and smoker.

```
lm.fit2 <- lm(charges ~ age +sex + bmi + children + smoker + region + obesity + obesity * smoker , data=insurance_train)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = charges ~ age + sex + bmi + children + smoker +
##     region + obesity + obesity * smoker, data = insurance_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4052.5 -1819.2 -1242.0  -460.9 24795.0
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -4878.263   1066.018  -4.576 5.30e-06 ***
## age                    265.694      9.829  27.032  < 2e-16 ***
## sexmale               -410.635    276.063  -1.487  0.13719
## bmi                    112.531     38.449   2.927  0.00350 **
## children               532.031    114.312   4.654 3.66e-06 ***
## smokeryes            13479.327    497.968  27.069  < 2e-16 ***
## regionnorthwest       -307.151    392.697  -0.782  0.43430
## regionsoutheast       -818.557    397.653  -2.058  0.03979 *
## regionsouthwest      -1195.685    397.809  -3.006  0.00271 **
## obesityyes            -782.642    476.986  -1.641  0.10113
## smokeryes:obesityyes 20055.315    681.876  29.412  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4487 on 1059 degrees of freedom
## Multiple R-squared:  0.8694, Adjusted R-squared:  0.8682
## F-statistic: 704.9 on 10 and 1059 DF,  p-value: < 2.2e-16
```

We could see that R-square is - 86.94%

Predict the charges on test data and find R-squared

```
insurance_test$predict <- predict(lm.fit2, newdata=insurance_test)
write.csv(insurance_test, file = 'InsuranceCostForecast.csv', row.names = FALSE, quote=FALSE)

rss <- sum((insurance_test$predict - insurance_test$charges)^2)
tss <- sum((insurance_test$charges - mean(insurance_test$charges))^2)

(rsq = 1- (rss/tss))
```

```
## [1] 0.8458934
```

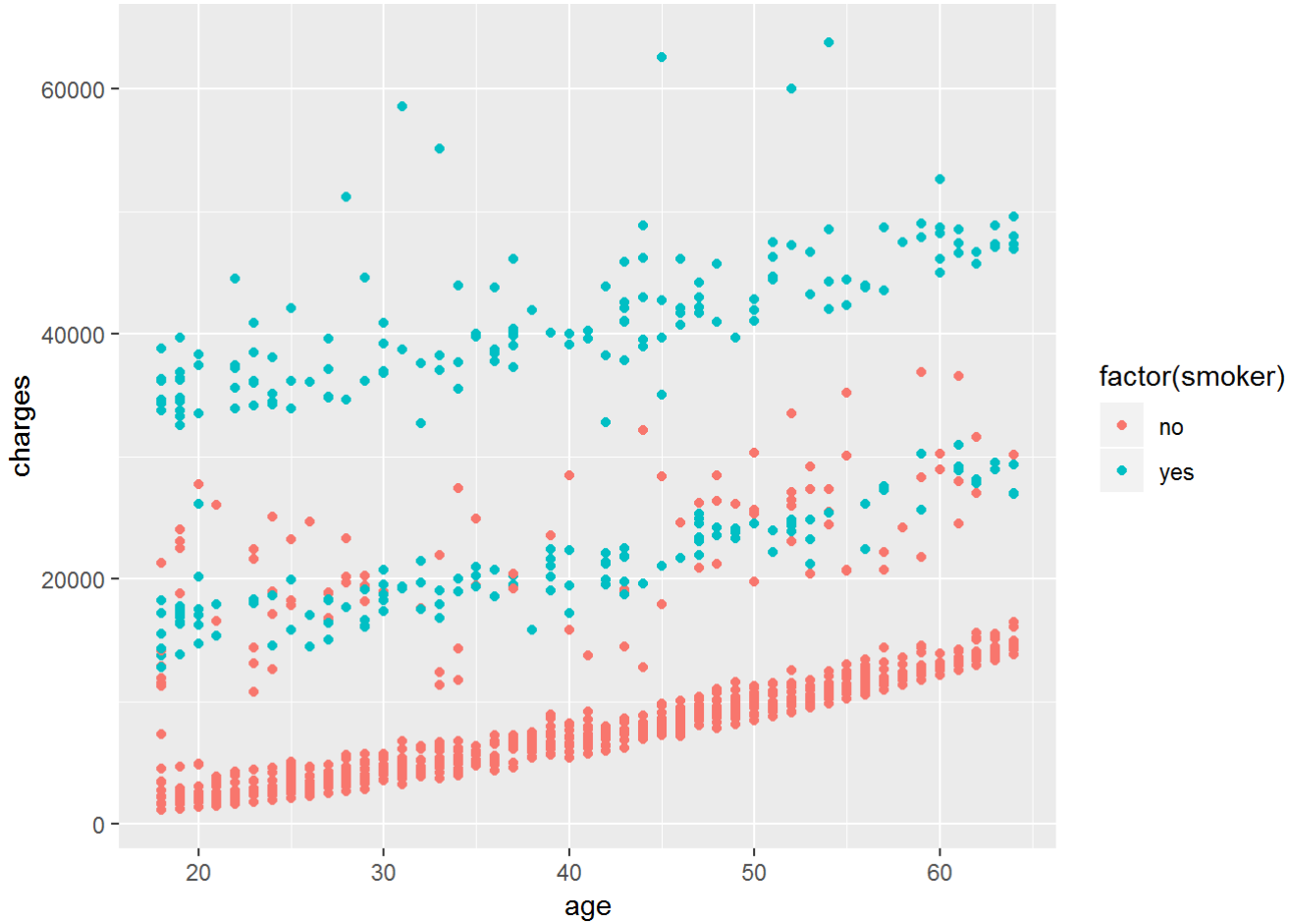
We can see that model predicted on test data which is almost good at accuracy of - 84.60%

Calculate RMSE

```
#res <- (insurance_test$predict - insurance_test$charges)
#(rmse <- sqrt(mean(res^2)))
#(sd <- sd(insurance_test$charges))
```

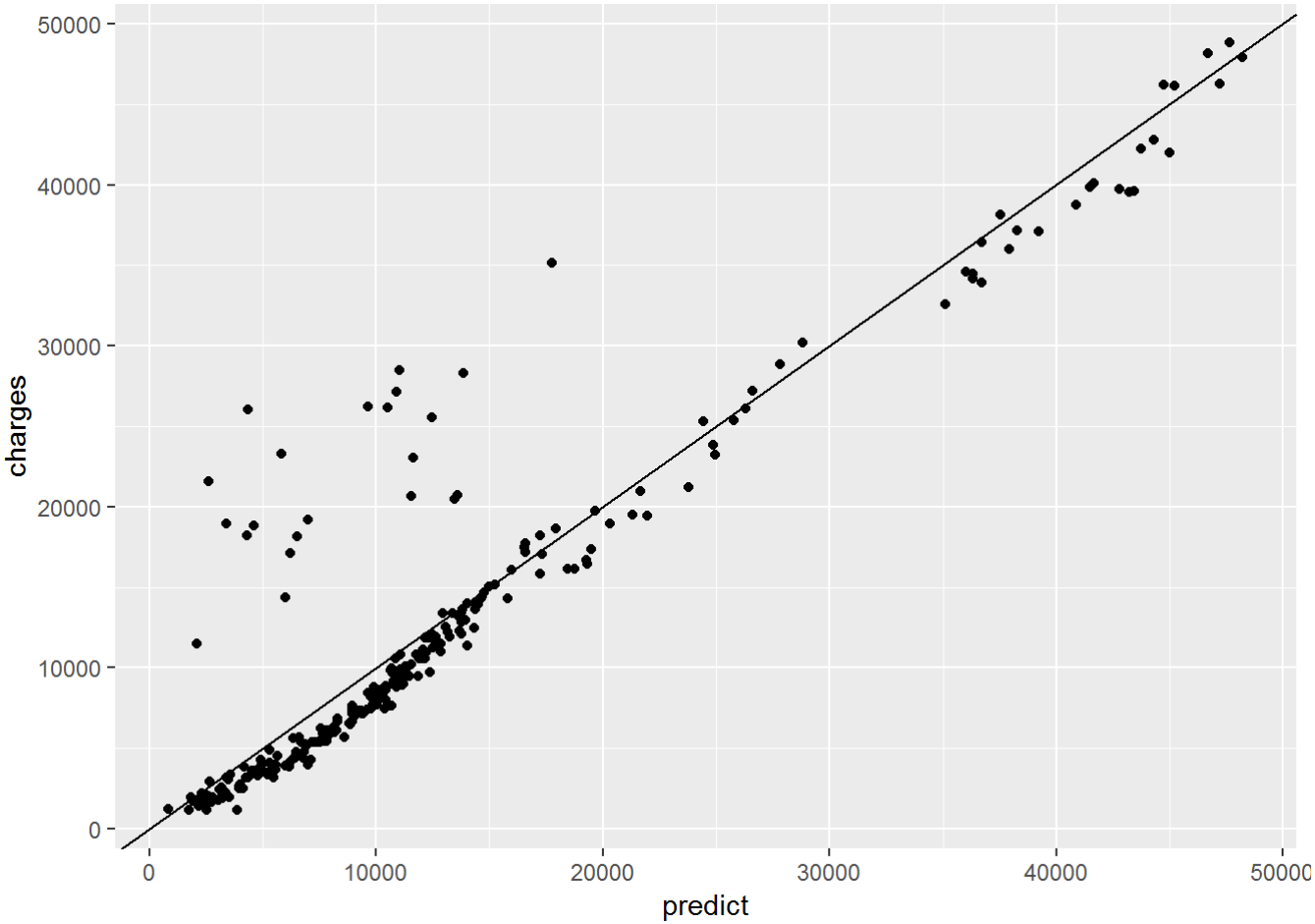
Plot the model

```
#plot(age,charges,col=smoker)
ggplot(data=insurance, aes(x=age, y=charges)) +
  geom_point(aes(colour = factor(smoker)))
```



Visualization of prediction

```
ggplot(data=insurance_test, aes(x=predict, y=charges))+
  geom_point()+
  geom_abline()
```



The prediction is almost good for 80

prcnt of data.

Gain plot

```
library(WVPlots)
```

```
## Warning: package 'WVPlots' was built under R version 3.5.1
```

```
GainCurvePlot(insurance_test, "predict", "charges", "lm.fit2")
```

Apply model on new data

Let's imagine 3 different people and see what charges on health care will be for them.

A: 19 years old, BMI 27.9, has no children, smokes, from northwest region.

B: 40 years old, BMI 50, 2 children, doesn't smoke, from southeast region.

C: 30 years old. BMI 31.2, no children, doesn't smoke, from northeast region.

```
A <- data.frame(age = 19,
                  bmi = 27.9,
                  children = 0,
                  smoker = "yes",
                   sex="male",
                obesity="no",
                  region = "northwest")
print(paste0("Health care charges for A: ", round(predict(lm.fit2, A), 2)))
```

```
## [1] "Health care charges for A: 16071.07"
```

```
B <- data.frame(age = 40,
                  bmi = 50,
                  children = 2,
                  smoker = "no",
                   sex="female",
                    obesity="yes",
                  region = "southeast")
print(paste0("Health care charges for B: ", round(predict(lm.fit2, B), 2)))
```

```
## [1] "Health care charges for B: 10838.9"
```

```
C <- data.frame(age = 30,
                  bmi = 31.2,
                  children = 0,
                  smoker = "no",
                   sex="male",
                    obesity="yes",
                  region = "northeast")
print(paste0("Health care charges for C: ", round(predict(lm.fit2, C), 2)))
```

```
## [1] "Health care charges for C: 5410.24"
```

Shapley Value regression is a technique for working out the relative importance of predictor variables in linear regression. Its principal application is to resolve a weakness of linear regression, which is that it is not reliable when predicted variables are moderately to highly correlated. Shapley Value regression is also known as Shapley regression

We will use a statistical method called shapley value regression which is a solution that originated from the Game Theory concept developed by Lloyd Shapley in the 1950s. It's aim is to fairly allocate predictor importance in regression analysis. Given n number of independent variables (IV), we will run all combination of linear regression models using this list of IVs against the dependent variable (DV) and get each model's R-Squared.

We have used the calc.relimp() function from the relaimpo package to determine the Shapley Value of our predictors.

```
ins_model2_shapley<-calc.relimp(lm.fit2,type="lmg")
ins_model2_shapley
```

```
## Response variable: charges
## Total response variance: 152689138
## Analysis based on 1070 observations
##
## 10 Regressors:
## Some regressors combined in groups:
##         Group  region : regionnorthwest regionsoutheast regionsouthwest
##
##  Relative importance of 8 (groups of) regressors assessed:
##  region age sex bmi children smoker obesity smoker:obesity
##
## Proportion of variance explained by model: 86.94%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
##                        lmg
## region         0.003508525
## age            0.089564897
## sex            0.001460615
## bmi            0.015728188
## children       0.003798175
## smoker         0.623350378
## obesity        0.026304658
## smoker:obesity 0.105674181
##
## Average coefficients for different model sizes:
##
##                      1group    2groups    3groups    4groups    5groups
## age                 269.0165   265.1620   262.3045   260.7897   261.2498
## sex                1580.4783  1289.8392  1024.9522   718.9677   333.2118
## bmi                 417.9505   370.0922   323.1570   272.4534   217.9561
## children            742.0159   707.5212   671.8575   637.5034   607.8378
## smoker            24159.1001 24132.2419 23466.6178 22018.4231 19905.4724
## regionnorthwest    -830.7132  -807.8512  -772.2894  -707.2350  -598.3814
## regionsoutheast    1674.2588   982.1597   408.1055   -95.3329  -521.4930
## regionsouthwest   -1314.2487 -1399.8415 -1445.5227 -1440.3795 -1378.1052
## obesity            5180.1425  4634.2722  3976.4939  3133.8229  2110.6893
## smoker:obesity          NaN        NaN 19603.6687 19692.0588 19781.3804
```

```
##                     6groups     7groups    8groups
## age               263.40988   265.36771   265.6943
## sex               -63.16146  -323.37984  -410.6351
## bmi               168.68117   134.46050   112.5306
## children          583.51704   559.62556   532.0313
## smoker          17513.88401 15282.28309 13479.3275
## regionnorthwest  -464.63794  -359.50648  -307.1513
## regionsoutheast  -796.58294  -874.80318  -818.5566
## regionsouthwest -1286.66801 -1220.32279 -1195.6854
## obesity          1016.74818    10.88403  -782.6422
## smoker:obesity  19871.67297 19962.97057 20055.3149
```

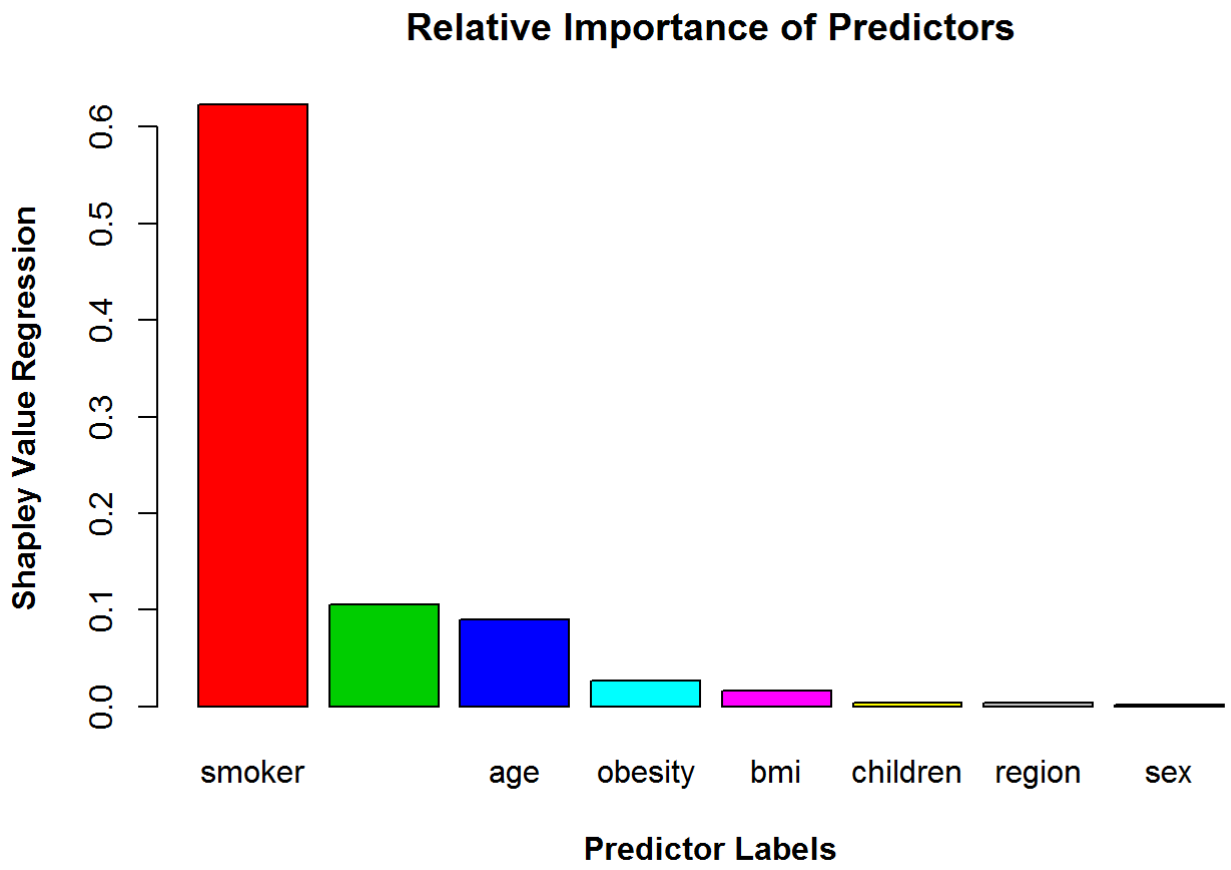
```
summary(ins_model2_shapley)
```

```
##   Length    Class     Mode
##        1 relimplm       S4
```

```
sum(ins_model2_shapley$lmg)
```

```
## [1] 0.8693896
```

```
barplot(sort(ins_model2_shapley$lmg,decreasing = TRUE),col=c(2:10),main="Relative Importance of Predictors",xlab="Predictor
 Labels",ylab="Shapley Value Regression",font.lab=2)
```



Above plot shows the important features in the order which is almost similar to our model features.

Closure

I have tried to test the features with multiple method and tried to show that features are correlated or not. I will add or update the model accordingly when ever I learn new stuffs.

Thankyou all for visiting my kernel and reading this!!!