CSE3506 - Essentials of Data Analytics - G2 Review - 3


```
 Name: Mayank Yadav - 20BCE1674
       Soumik Kabiraj - 20BCE1504
       Panav Sinha - 20BCE1640
```


```
library(dplyr)
```

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

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

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

```
library(tidyverse)
```

```
## ── Attaching packages
## ─────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✔ ggplot2 3.4.0     ✔ purrr   1.0.0
## ✔ tibble  3.1.8     ✔ stringr 1.5.0
## ✔ tidyr   1.2.1     ✔ forcats 0.5.2
## ✔ readr   2.1.3
## ── Conflicts ──────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(grid)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(mlbench)
library(caretEnsemble)
```

```
##
## Attaching package: 'caretEnsemble'
##
## The following object is masked from 'package:ggplot2':
##
##     autoplot
```

```
library(ggrepel)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(ggExtra)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(hrbrthemes)
```

```
## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.
##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and
##       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

```
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```
df <- read.csv("smoking.csv")
summary(df)
```

```
##        ID           gender               age          height.cm.
##  Min.   :    0   Length:55692       Min.   :20.00   Min.   :130.0
##  1st Qu.:13923   Class :character   1st Qu.:40.00   1st Qu.:160.0
##  Median :27846   Mode  :character   Median :40.00   Median :165.0
##  Mean   :27846                      Mean   :44.18   Mean   :164.6
##  3rd Qu.:41768                      3rd Qu.:55.00   3rd Qu.:170.0
##  Max.   :55691                      Max.   :85.00   Max.   :190.0
##    weight.kg.       waist.cm.      eyesight.left.  eyesight.right.
##  Min.   : 30.00   Min.   : 51.00   Min.   :0.100   Min.   :0.100
##  1st Qu.: 55.00   1st Qu.: 76.00   1st Qu.:0.800   1st Qu.:0.800
##  Median : 65.00   Median : 82.00   Median :1.000   Median :1.000
##  Mean   : 65.86   Mean   : 82.05   Mean   :1.013   Mean   :1.007
##  3rd Qu.: 75.00   3rd Qu.: 88.00   3rd Qu.:1.200   3rd Qu.:1.200
##  Max.   :135.00   Max.   :129.00   Max.   :9.900   Max.   :9.900
##  hearing.left.   hearing.right.     systolic       relaxation
##  Min.   :1.000   Min.   :1.000   Min.   : 71.0   Min.   : 40
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:112.0   1st Qu.: 70
##  Median :1.000   Median :1.000   Median :120.0   Median : 76
##  Mean   :1.026   Mean   :1.026   Mean   :121.5   Mean   : 76
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:130.0   3rd Qu.: 82
##  Max.   :2.000   Max.   :2.000   Max.   :240.0   Max.   :146
##  fasting.blood.sugar  Cholesterol     triglyceride        HDL
##  Min.   : 46.00      Min.   : 55.0   Min.   :  8.0   Min.   :  4.00
##  1st Qu.: 89.00      1st Qu.:172.0   1st Qu.: 74.0   1st Qu.: 47.00
##  Median : 96.00      Median :195.0   Median :108.0   Median : 55.00
##  Mean   : 99.31      Mean   :196.9   Mean   :126.7   Mean   : 57.29
##  3rd Qu.:104.00      3rd Qu.:220.0   3rd Qu.:160.0   3rd Qu.: 66.00
##  Max.   :505.00      Max.   :445.0   Max.   :999.0   Max.   :618.00
##       LDL         hemoglobin    Urine.protein   serum.creatinine
##  Min.   :   1   Min.   : 4.90   Min.   :1.000   Min.   : 0.1000
##  1st Qu.:  92   1st Qu.:13.60   1st Qu.:1.000   1st Qu.: 0.8000
##  Median : 113   Median :14.80   Median :1.000   Median : 0.9000
##  Mean   : 115   Mean   :14.62   Mean   :1.087   Mean   : 0.8857
##  3rd Qu.: 136   3rd Qu.:15.80   3rd Qu.:1.000   3rd Qu.: 1.0000
##  Max.   :1860   Max.   :21.10   Max.   :6.000   Max.   :11.6000
##       AST              ALT              Gtp             oral
##  Min.   :   6.00   Min.   :   1.00   Min.   :  1.00   Length:55692
##  1st Qu.:  19.00   1st Qu.:  15.00   1st Qu.: 17.00   Class :character
##  Median :  23.00   Median :  21.00   Median : 25.00   Mode  :character
##  Mean   :  26.18   Mean   :  27.04   Mean   : 39.95
##  3rd Qu.:  28.00   3rd Qu.:  31.00   3rd Qu.: 43.00
##  Max.   :1311.00   Max.   :2914.00   Max.   :999.00
##  dental.caries       tartar            smoking
##  Min.   :0.0000   Length:55692       Min.   :0.0000
##  1st Qu.:0.0000   Class :character   1st Qu.:0.0000
##  Median :0.0000   Mode  :character   Median :0.0000
##  Mean   :0.2133                      Mean   :0.3673
##  3rd Qu.:0.0000                      3rd Qu.:1.0000
##  Max.   :1.0000                      Max.   :1.0000
```

```
str(df)
```

```
## 'data.frame':    55692 obs. of  27 variables:
##  $ ID                 : int  0 1 2 3 4 5 6 7 9 10 ...
##  $ gender             : chr  "F" "F" "M" "M" ...
##  $ age                : int  40 40 55 40 40 30 40 45 50 45 ...
##  $ height.cm.         : int  155 160 170 165 155 180 160 165 150 175 ...
##  $ weight.kg.         : int  60 60 60 70 60 75 60 90 60 75 ...
##  $ waist.cm.          : num  81.3 81 80 88 86 85 85.5 96 85 89 ...
##  $ eyesight.left.     : num  1.2 0.8 0.8 1.5 1 1.2 1 1.2 0.7 1 ...
##  $ eyesight.right.    : num  1 0.6 0.8 1.5 1 1.2 1 1 0.8 1 ...
##  $ hearing.left.      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ hearing.right.     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ systolic           : num  114 119 138 100 120 128 116 153 115 113 ...
##  $ relaxation         : num  73 70 86 60 74 76 82 96 74 64 ...
##  $ fasting.blood.sugar: num  94 130 89 96 80 95 94 158 86 94 ...
##  $ Cholesterol        : num  215 192 242 322 184 217 226 222 210 198 ...
##  $ triglyceride       : num  82 115 182 254 74 199 68 269 66 147 ...
##  $ HDL                : num  73 42 55 45 62 48 55 34 48 43 ...
##  $ LDL                : num  126 127 151 226 107 129 157 134 149 126 ...
##  $ hemoglobin         : num  12.9 12.7 15.8 14.7 12.5 16.2 17 15 13.7 16 ...
##  $ Urine.protein      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ serum.creatinine   : num  0.7 0.6 1 1 0.6 1.2 0.7 1.3 0.8 0.8 ...
##  $ AST                : num  18 22 21 19 16 18 21 38 31 26 ...
##  $ ALT                : num  19 19 16 26 14 27 27 71 31 24 ...
##  $ Gtp                : num  27 18 22 18 22 33 39 111 14 63 ...
##  $ oral               : chr  "Y" "Y" "Y" "Y" ...
##  $ dental.caries      : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ tartar             : chr  "Y" "Y" "N" "Y" ...
##  $ smoking            : int  0 0 1 0 0 0 1 0 0 0 ...
```

```
#Tartar is the hard calcified deposits that form and coat the teeth and gums
unique(df$tartar)
```

```
## [1] "Y" "N"
```

```
unique(df$gender)
```

```
## [1] "F" "M"
```

```
unique(df$oral)
```

```
## [1] "Y"
```

```
df$sex_num <- ifelse(df$gender=="F",0,1)
df$tartar <-  ifelse(df$tartar=="Y",1,0)
class(df$oral)
```

```
## [1] "character"
```

```
df$oral <- as.numeric(as.factor(df$oral))
class(df$oral)
```

```
## [1] "numeric"
```

Are there outliers? #### Outliers for continuous variables

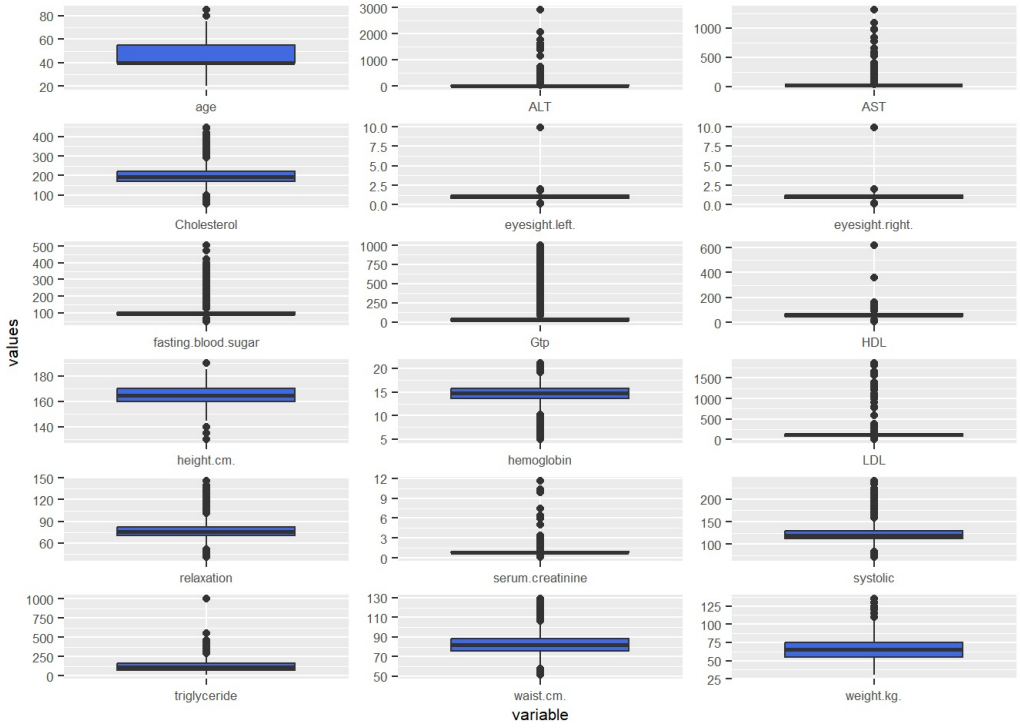
```
df_num<-select_if(df,is.numeric) %>% select(-c(ID,hearing.right.,hearing.left.,smoking,dental.caries,Urine.protei
n))
head(df_num)
```

```
##   age height.cm. weight.kg. waist.cm. eyesight.left. eyesight.right. systolic
## 1  40        155         60      81.3            1.2             1.0      114
## 2  40        160         60      81.0            0.8             0.6      119
## 3  55        170         60      80.0            0.8             0.8      138
## 4  40        165         70      88.0            1.5             1.5      100
## 5  40        155         60      86.0            1.0             1.0      120
## 6  30        180         75      85.0            1.2             1.2      128
##   relaxation fasting.blood.sugar Cholesterol triglyceride HDL LDL hemoglobin
## 1         73                  94         215           82  73 126       12.9
## 2         70                 130         192          115  42 127       12.7
## 3         86                  89         242          182  55 151       15.8
## 4         60                  96         322          254  45 226       14.7
## 5         74                  80         184           74  62 107       12.5
## 6         76                  95         217          199  48 129       16.2
##   serum.creatinine AST ALT Gtp oral tartar sex_num
## 1              0.7  18  19  27    1      1       0
## 2              0.6  22  19  18    1      1       0
## 3              1.0  21  16  22    1      0       1
## 4              1.0  19  26  18    1      1       1
## 5              0.6  16  14  22    1      0       0
## 6              1.2  18  27  33    1      1       1
```

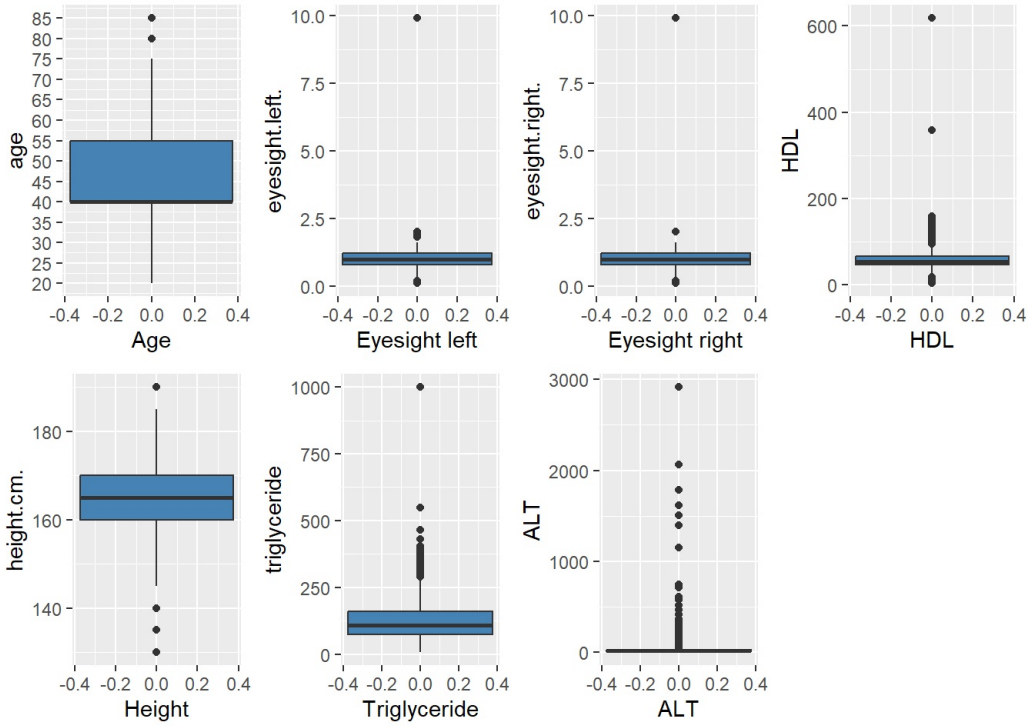
```
df_num_p<-df_num %>% gather(variable,values,1:18)
head(df_num_p)
```

```
##   oral tartar sex_num variable values
## 1    1      1       0      age     40
## 2    1      1       0      age     40
## 3    1      0       1      age     55
## 4    1      1       1      age     40
## 5    1      0       0      age     40
## 6    1      1       1      age     30
```

```
options(repr.plot.width = 18, repr.plot.height = 14)
ggplot(df_num_p)+geom_boxplot(aes(x=variable,y=values),fill="royalblue") + facet_wrap(~variable,ncol=3,scales="fr
ee") + theme(strip.text.x = element_blank(),text = element_text(size=8))
```



```
a <- ggplot(df, aes(age))+
  geom_boxplot(fill="steelblue")+scale_x_continuous(breaks=seq(0,100,5))+labs(y="Age")+coord_flip()

b <- ggplot(df, aes(eyesight.left.))+
  geom_boxplot(fill="steelblue")+labs(y="Eyesight left")+coord_flip()

c <- ggplot(df, aes(eyesight.right.))+
  geom_boxplot(fill="steelblue")+labs(y="Eyesight right")+coord_flip()

d <- ggplot(df, aes(HDL))+
  geom_boxplot(fill="steelblue")+labs(y="HDL")+coord_flip()

e <- ggplot(df, aes(height.cm.))+
  geom_boxplot(fill="steelblue")+labs(y="Height")+coord_flip()

f <- ggplot(df, aes(triglyceride))+
  geom_boxplot(fill="steelblue")+labs(y="Triglyceride")+coord_flip()

g <- ggplot(df, aes(ALT))+
  geom_boxplot(fill="steelblue")+labs(y="ALT")+coord_flip()

options(repr.plot.width = 18, repr.plot.height = 12)
grid.arrange(a,b,c,d,e,f,g, ncol=4)
```

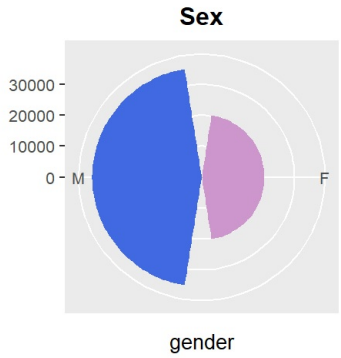


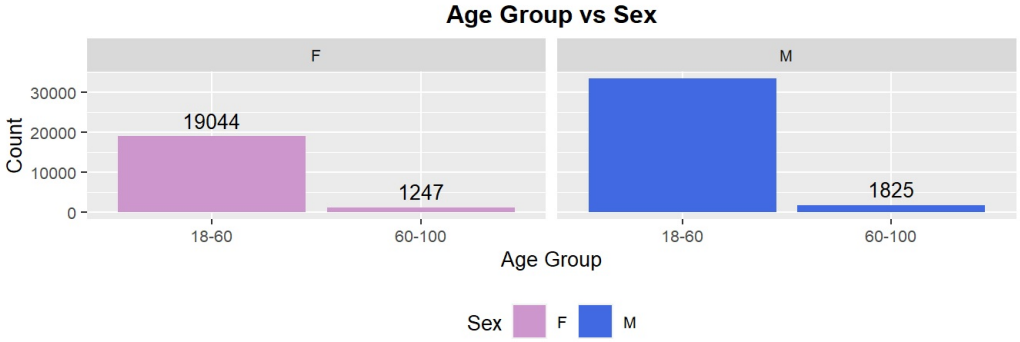

```
df$age_grp <- cut(df$age, c(0,17,60,100,120), labels = c("0-17","18-60","60-100","100+") )
head(df$age_grp)
```

```
## [1] 18-60 18-60 18-60 18-60 18-60 18-60
## Levels: 0-17 18-60 60-100 100+
```

```
h <- ggplot(df,aes(x=`age_grp`, fill=`gender`))+geom_bar()+facet_grid(.~gender)+
  stat_count(aes(y=..count.., label=..count..), vjust=-0.5,geom="text", col="black", size=3.5)+
  labs(x="Age Group", y = "Count", title="Age Group vs Sex", fill= "Sex")+
  theme(plot.title=element_text(face="bold",  hjust=0.5), legend.position = "bottom",text=element_text(size=10))+
    scale_fill_manual(values=c("plum3","royalblue"))

i <- ggplot(df, aes(y = gender,fill=gender))+
  geom_bar()+
  scale_fill_manual(values=c("plum3","royalblue"))+
  labs(title="Sex",x=" ")+
  theme(legend.position = "none", plot.title=element_text(face="bold",  hjust=0.5), text=element_text(size=10))+
  coord_polar("y")

options(repr.plot.width = 16, repr.plot.height = 12)
grid.arrange(i, h, ncol=1)
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(count)` instead.
```

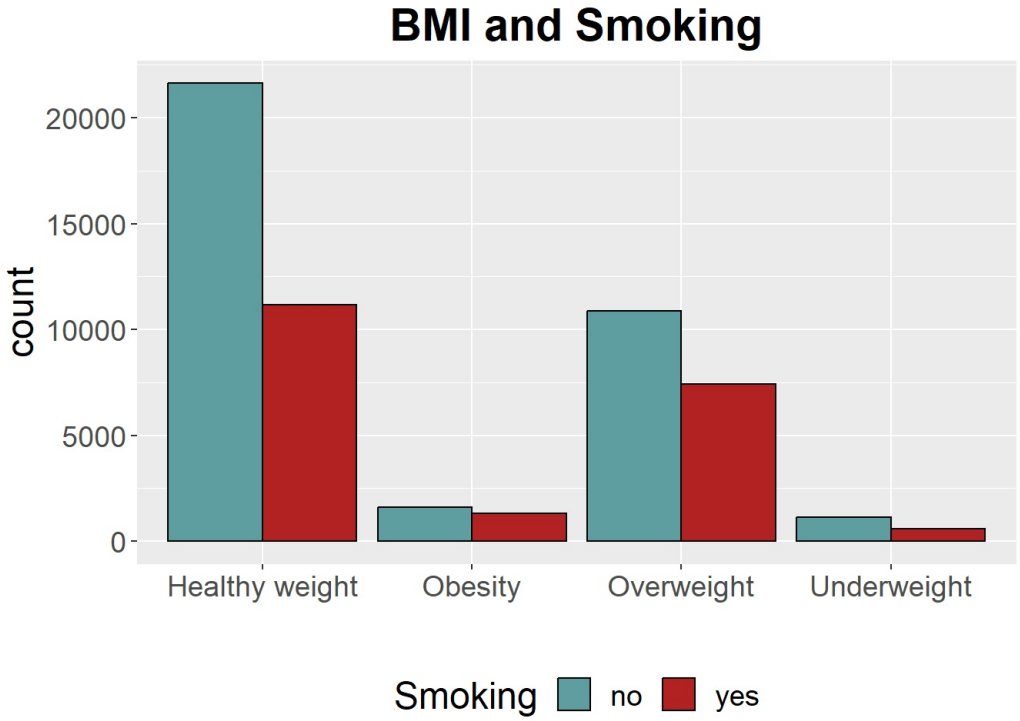





```
df$bmi <- (df$weight.kg./(df$height.cm.*df$height.cm.))*10000
df$bmi_cat <- ifelse(df$bmi<18.5,"Underweight", NA )
df$bmi_cat <- ifelse(df$bmi>=18.5 & df$bmi <=24.9,"Healthy weight", df$bmi_cat )
df$bmi_cat <- ifelse(df$bmi > 24.9 & df$bmi<=29.9,"Overweight", df$bmi_cat)
df$bmi_cat <- ifelse(df$bmi > 30,"Obesity", df$bmi_cat)
table(df$bmi_cat)
```

```
##
## Healthy weight        Obesity     Overweight    Underweight
##          32773           2923          18278           1718
```

```
ggplot(df, aes(x=bmi_cat, fill= as.factor(smoking)))+
  geom_bar(position="dodge", col="black")+
  scale_fill_manual(label=c("no","yes"),values=c("cadetblue", "firebrick"))+
  theme(legend.position = "bottom")+
  labs(x=" ", fill="Smoking", title="BMI and Smoking")+
  theme(text=element_text(size=18), plot.title= element_text(face="bold", hjust=0.5))
```



```
msel <- df%>%
  dplyr::filter(smoking==1)%>%
  summarize(mean(eyesight.left.))
sprintf("Mean left eyepower of smokers: %f",msel)
```

```
## [1] "Mean left eyepower of smokers: 1.051733"
```

```
mnsel <- df%>%
  dplyr::filter(smoking==0)%>%
  summarize(mean(eyesight.left.))
sprintf("Mean left eyepower of non-smokers: %f",mnsel)
```

```
## [1] "Mean left eyepower of non-smokers: 0.989920"
```

```
mser <- df%>%
  dplyr::filter(smoking==1)%>%
  summarize(mean(eyesight.right.))
sprintf("Mean right eyepower of smokers: %f",mser)
```

```
## [1] "Mean right eyepower of smokers: 1.047636"
```

```
mnser <- df%>%
  dplyr::filter(smoking==0)%>%
  summarize(mean(eyesight.right.))
sprintf("Mean right eyepower of non-smokers: %f",mnser)
```

```
## [1] "Mean right eyepower of non-smokers: 0.984110"
```

```
BP <- c("Normal","Elevated","Hypertension I","Hypertension II", "Hypertensive Crisis")
sys <- c("less than 120", "120-129", "130-139", "140 or higher", "more than 180")
dias <- c("less than 80", " less than 80", "80-89", "90 or higher", "120 or higher")
pressure <- data.frame(BP, sys,dias)
colnames(pressure) <- c("Blood Pressure", "Systolic", "Diastolic (Relaxed)")
pressure
```

```
##        Blood Pressure      Systolic Diastolic (Relaxed)
## 1              Normal less than 120        less than 80
## 2            Elevated       120-129        less than 80
## 3      Hypertension I       130-139               80-89
## 4     Hypertension II 140 or higher        90 or higher
## 5 Hypertensive Crisis more than 180       120 or higher
```

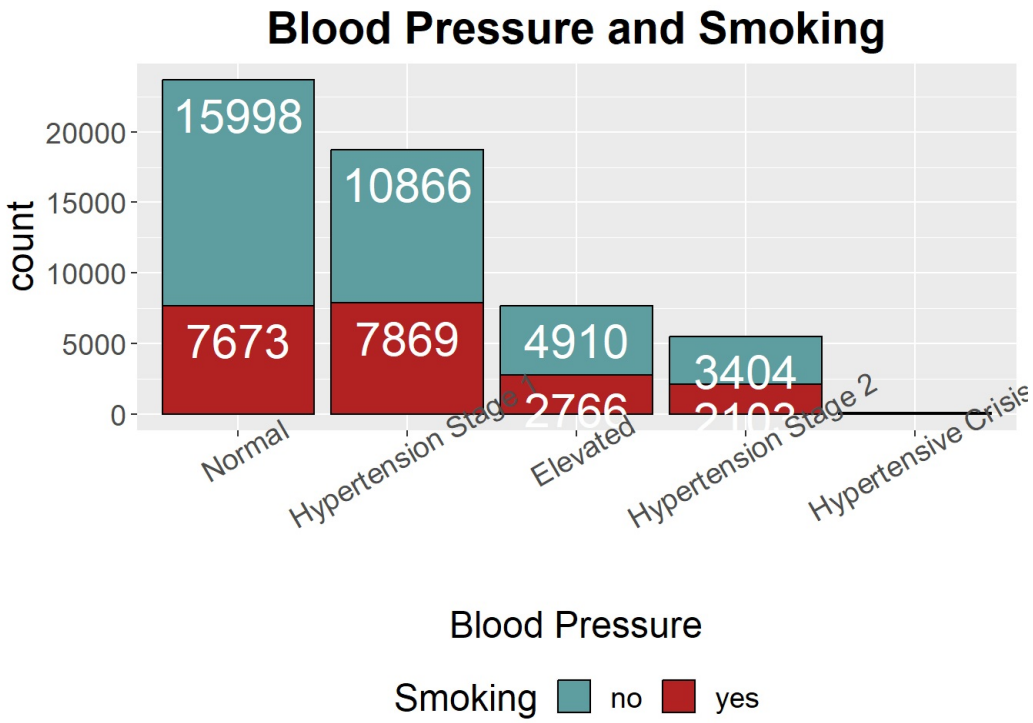
```
df$pressure <- ifelse(df$systolic<120 & df$relaxation<80,"Normal",NA)
df$pressure <- ifelse(df$systolic>=120 & df$relaxation <80, "Elevated",df$pressure)
df$pressure <- ifelse(df$systolic>=130 | df$relaxation>=80,"Hypertension Stage 1",df$pressure)
df$pressure <- ifelse(df$systolic>=140 | df$relaxation>=90,"Hypertension Stage 2",df$pressure)
df$pressure <- ifelse(df$systolic>=180 | df$relaxation>=120,"Hypertensive Crisis",df$pressure)
table(df$pressure)
```

```
##
##             Elevated Hypertension Stage 1 Hypertension Stage 2
##                 7676                18735                 5507
##  Hypertensive Crisis               Normal
##                  103                23671
```

```
table <- df %>%
  group_by(df$pressure)%>%
  summarise(smoker= length(smoking[smoking=="1"]),
            nonsmoker= length(smoking[smoking=="0"]),
            smoker_percent= smoker/sum(smoker,nonsmoker)*100)%>%
  arrange(desc(smoker_percent))

table
```

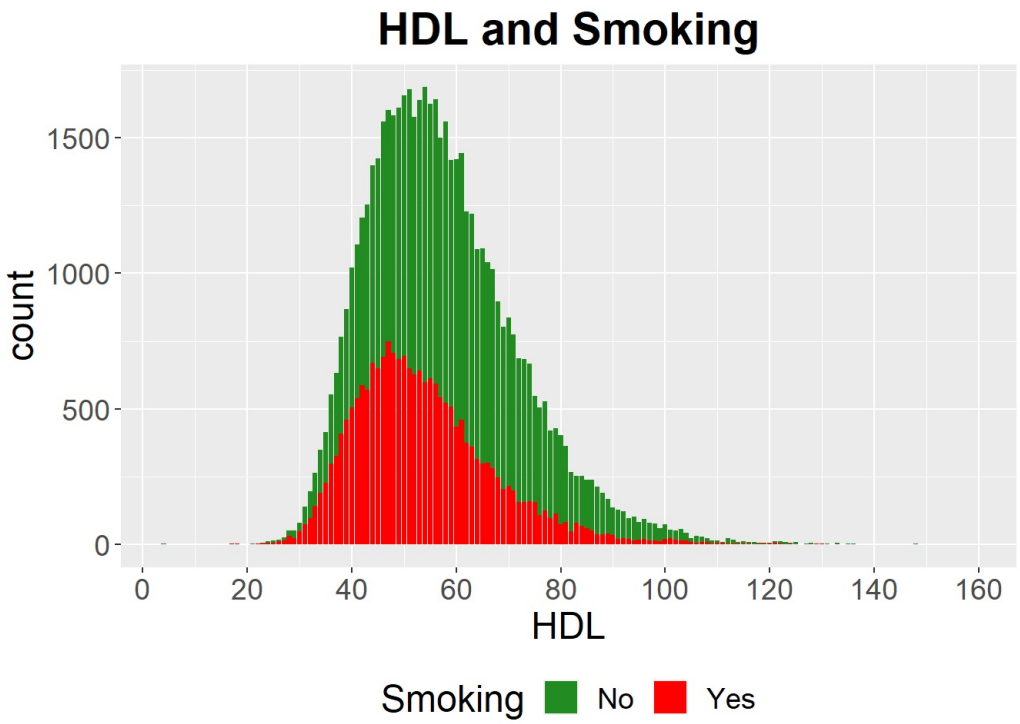
```
## # A tibble: 5 × 4
##   `df$pressure`        smoker nonsmoker smoker_percent
##   <chr>                 <int>     <int>          <dbl>
## 1 Hypertensive Crisis      44        59           42.7
## 2 Hypertension Stage 1   7869     10866           42.0
## 3 Hypertension Stage 2   2103      3404           38.2
## 4 Elevated               2766      4910           36.0
## 5 Normal                 7673     15998           32.4
```

```
options(repr.plot.width = 16, repr.plot.height = 12)
ggplot(df, aes(x=reorder(pressure,pressure,function(x)-length(x)), fill=as.factor(smoking)))+
  geom_bar(col="black",position="stack")+
  theme(plot.title=element_text(face="bold",hjust=0.5),legend.position = "bottom", axis.text.x = element_text(ang
le=30), text=element_text(size=18))+
  labs(title="Blood Pressure and Smoking",x="Blood Pressure", fill="Smoking")+
  stat_count(aes(y=..count..,label=..count..), geom="text", vjust=1.5, size=8, col="white")+
  scale_fill_manual(label=c("no","yes"),values=c("cadetblue", "firebrick"))
```



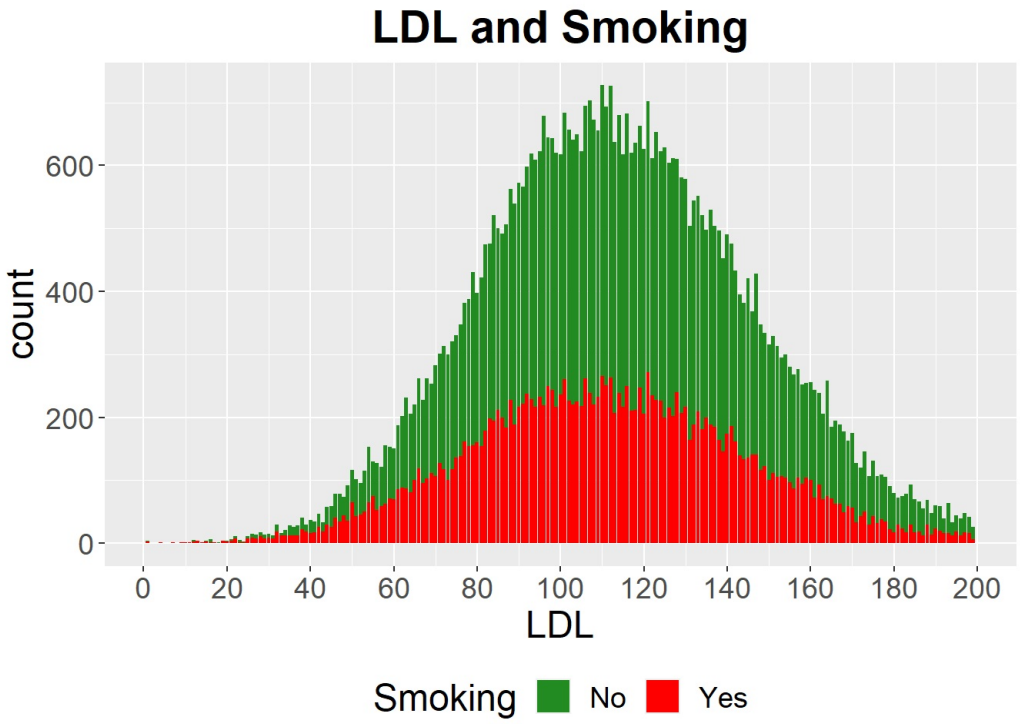
Does smoking increase cholesterol? #### Cholesterol: Do smokers have lower good fat HDL? Does smoking increase harmful fats?

```
options(repr.plot.width = 16, repr.plot.height = 12)
ggplot(subset(df,df$HDL<200), aes(x=HDL, fill=as.factor(smoking)))+
  geom_bar()+
  labs(title="HDL and Smoking", fill="Smoking")+
  theme(legend.position = "bottom", plot.title = element_text(face="bold", hjust=0.5), text=element_text(size=18)
)+
  scale_x_continuous(breaks=seq(0,200,20))+
  scale_fill_manual(labels=c("No","Yes"),values=c("forest green","red"))
```



LDL (the bad cholesterol)

```
options(repr.plot.width = 16, repr.plot.height = 12)
ggplot(subset(df,df$LDL<200), aes(x=LDL, fill=as.factor(smoking)))+
  geom_bar()+
  labs(title="LDL and Smoking", fill="Smoking")+
  theme(legend.position = "bottom", plot.title = element_text(face="bold", hjust=0.5), text=element_text(size=18)
)+
  scale_x_continuous(breaks=seq(0,200,20))+
  scale_fill_manual(labels=c("No","Yes"),values=c("forest green","red"))
```



Triglycerides #### The cutoff for triglyceride is 150, above 150 is at risk.

```
df%>% select(c("triglyceride", "smoking")) %>%
  group_by(smoking) %>%
  summarise(tryg_mean = mean(triglyceride)) %>%
  mutate(., Type=ifelse(tryg_mean<150,"Healthy","Risk"))
```

```
## # A tibble: 2 × 3
##   smoking tryg_mean Type
##     <int>     <dbl> <chr>
## 1       0      113. Healthy
## 2       1      150. Risk
```

Liver: Are liver functions worse for smokers? #### AST normal range - 8 to 33 U/L #### ALT normal range - 4 to 36 U/L #### GTP normal range - 5 to 40 U/L

```
df%>% select(c("ALT", "AST", "Gtp", "smoking")) %>%
  group_by(smoking) %>%
  summarise(Alt_mean = mean(ALT),
            Ast_mean = mean(AST),
            Gtp_mean = mean(Gtp))
```

```
## # A tibble: 2 × 4
##   smoking Alt_mean Ast_mean Gtp_mean
##     <int>    <dbl>    <dbl>    <dbl>
## 1       0     24.7     25.3     30.9
## 2       1     31.0     27.7     55.6
```

Blood: Do smokers have higher hemoglobin?

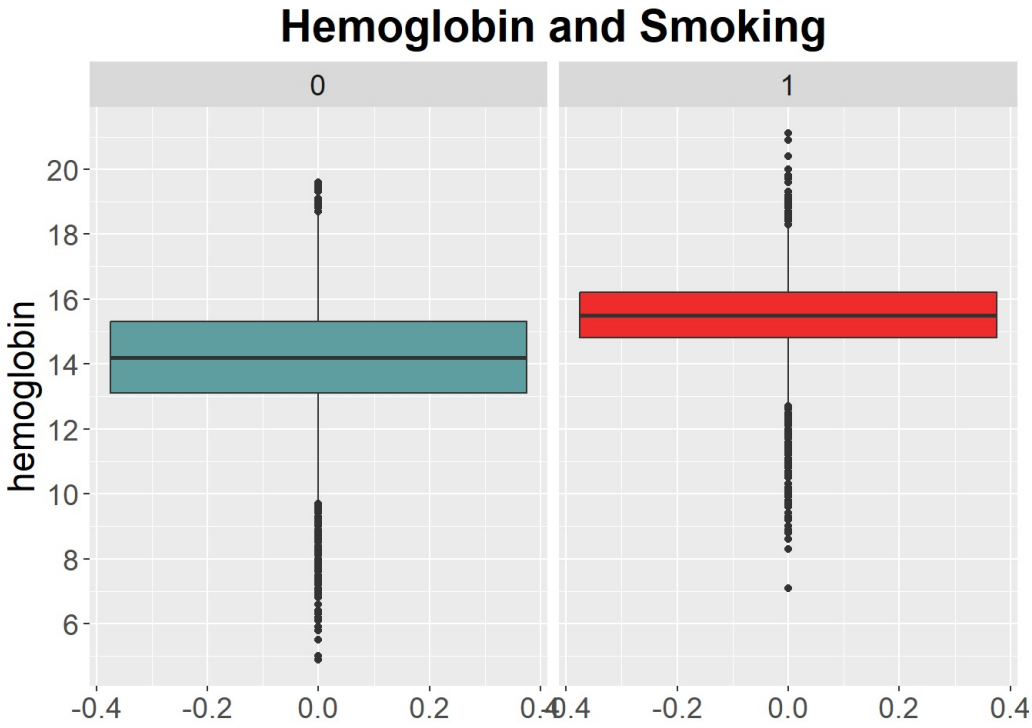
```
mean(df[df$smoking=="1",'hemoglobin'])
```

```
## [1] 15.44534
```

```
mean(df[df$smoking=="0",'hemoglobin'])
```

```
## [1] 14.14499
```

```
options(repr.plot.width = 14, repr.plot.height = 10)
ggplot(df, aes(y=hemoglobin))+
  geom_boxplot(fill=c("cadetblue","firebrick2"))+
  facet_grid(.~as.factor(smoking))+
  scale_y_continuous(breaks= seq(0,20,2))+
  labs(title="Hemoglobin and Smoking", fill="smoking")+
  theme(plot.title = element_text(face="bold", hjust=0.5), text=element_text(size=18))
```



Random Forest Model

```
# Split data
library(caTools)
set.seed(123)
df$split<- sample.split(df$smoking, SplitRatio = 0.7)
df_train <- df %>% filter( split==TRUE) %>% select(-split)
df_test <- df%>% filter(split==FALSE)%>%select(-split)

paste0("Number of rows in train dataset: ",nrow(df_train))
```

```
## [1] "Number of rows in train dataset: 38984"
```

```
paste0("Number of rows in test dataset: ", nrow(df_test))
```

```
## [1] "Number of rows in test dataset: 16708"
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

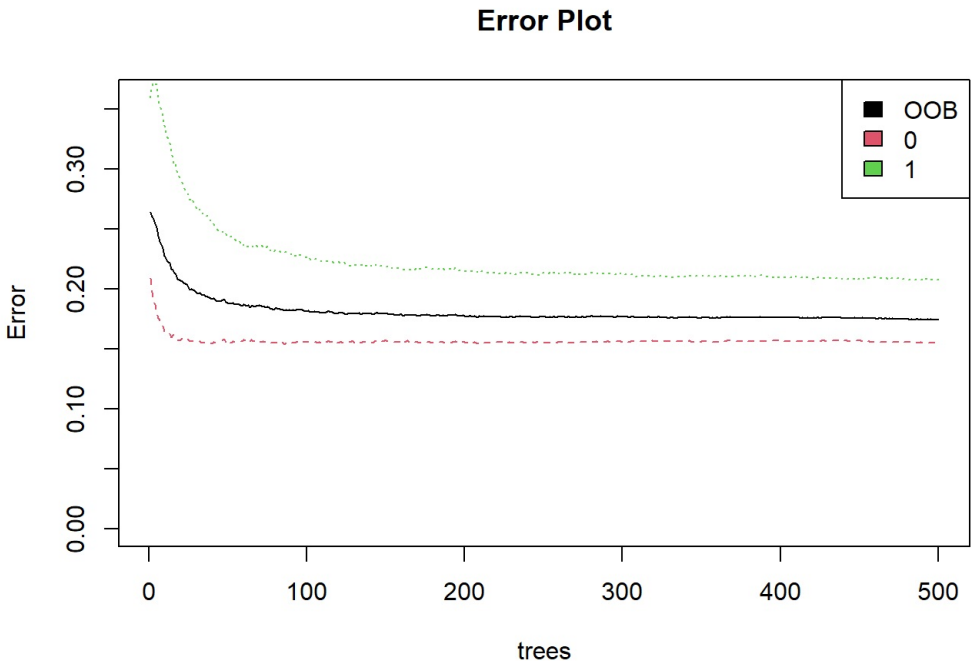
```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

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

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

```
set.seed(1234)

model1 <- randomForest(factor(smoking) ~ age + height.cm. + weight.kg.+ sex_num +
                         eyesight.left. + eyesight.right. + hearing.left. + hearing.right. +
                         systolic + relaxation +
                         Cholesterol + triglyceride + HDL + LDL +
                         fasting.blood.sugar + hemoglobin + Urine.protein + serum.creatinine+
                         AST + ALT + Gtp +
                         dental.caries + tartar,
                                importance=TRUE,
                                ntree= 500,
                                data = df_train)
```

```
options(repr.plot.width = 14, repr.plot.height = 10)
plot(model1, ylim=c(0,0.36),main="Error Plot")
legend('topright', colnames(model1$err.rate), col=1:3, fill=1:3)
```



Plot important variables.

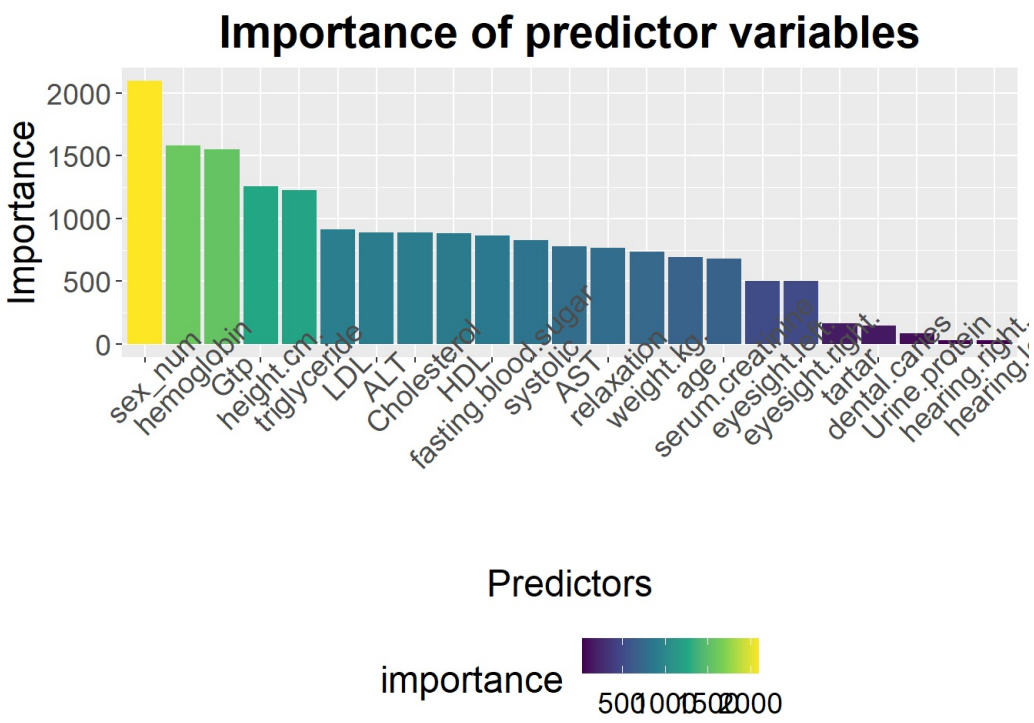
```
imp <- importance(model1)

# Creating a dataframe of variable importance metrics
df_imp<- data.frame(variable=row.names(imp),
                    importance = imp[,'MeanDecreaseGini'],
                    accuracy= imp[,'MeanDecreaseAccuracy'])
noquote("Variable Importance Dataframe")
```

```
## [1] Variable Importance Dataframe
```

```
df_imp
```

```
##                                variable importance  accuracy
## age                                 age  691.90807  54.74566
## height.cm.                   height.cm. 1256.11838  48.47693
## weight.kg.                   weight.kg.  735.36271  53.18362
## sex_num                         sex_num 2097.06889 124.21970
## eyesight.left.           eyesight.left.  502.91932  64.47683
## eyesight.right.         eyesight.right.  500.19518  57.14533
## hearing.left.             hearing.left.   31.74678  18.91422
## hearing.right.           hearing.right.   32.36123  20.30818
## systolic                       systolic  827.89459  97.32116
## relaxation                   relaxation  767.45412  75.53918
## Cholesterol                 Cholesterol  889.35487  92.11063
## triglyceride               triglyceride 1224.85359  93.83893
## HDL                                 HDL  883.24901  65.26985
## LDL                                 LDL  913.34755  91.44271
## fasting.blood.sugar fasting.blood.sugar  866.41434  67.81235
## hemoglobin                   hemoglobin 1582.56022  75.31676
## Urine.protein             Urine.protein   86.59364  29.17171
## serum.creatinine       serum.creatinine  683.43251  54.67041
## AST                                 AST  776.84458  87.56108
## ALT                                 ALT  891.19585  68.97634
## Gtp                                 Gtp 1550.22823 140.48718
## dental.caries             dental.caries  145.17856  45.31770
## tartar                           tartar  164.18581  49.18509
```

```
# Visualizations of importance variables
options(repr.plot.width = 14, repr.plot.height = 10)
ggplot(df_imp, aes(x=reorder(variable,-importance),y=importance, fill=importance))+
  geom_col()+
  labs(x="Predictors", y ="Importance", title="Importance of predictor variables")+
  theme(plot.title=element_text(face="bold",hjust=0.5), legend.position = "bottom", axis.text.x = element_text(an
gle=45), text=element_text(size=18))+
  scale_fill_continuous(type="viridis")
```



Let us remove some of the low-contribution variables and improve the model.

```
model2 <- randomForest(factor(smoking) ~ age + height.cm. + weight.kg.+ sex_num +
                         systolic + relaxation +
                         Cholesterol + triglyceride + HDL + LDL +
                         fasting.blood.sugar + hemoglobin + serum.creatinine+
                         AST + ALT + Gtp +
                         dental.caries ,
                                importance=TRUE,
                                ntree= 500,
                                data = df_train)
paste("done")
```

```
## [1] "done"
```

```
model2
```

```
##
## Call:
##  randomForest(formula = factor(smoking) ~ age + height.cm. + weight.kg. +      sex_num + systolic + relaxation
+ Cholesterol + triglyceride +      HDL + LDL + fasting.blood.sugar + hemoglobin + serum.creatinine +      AST +
ALT + Gtp + dental.caries, data = df_train, importance = TRUE,      ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.76%
## Confusion matrix:
##       0     1 class.error
## 0 20810  3856   0.1563285
## 1  3067 11251   0.2142059
```

```
model1
```

```
##
## Call:
##  randomForest(formula = factor(smoking) ~ age + height.cm. + weight.kg. +      sex_num + eyesight.left. + eyes
ight.right. + hearing.left. +      hearing.right. + systolic + relaxation + Cholesterol + triglyceride +      HDL
+ LDL + fasting.blood.sugar + hemoglobin + Urine.protein +      serum.creatinine + AST + ALT + Gtp + dental.carie
s + tartar,      data = df_train, importance = TRUE, ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.44%
## Confusion matrix:
##       0     1 class.error
## 0 20845  3821   0.1549096
## 1  2978 11340   0.2079899
```

```
cat(paste("Accuracy of model 1: ", (100-17.44), "%"))
```

```
## Accuracy of model 1:  82.56 %
```

```
cat(paste("Accuracy of model 2: ", (100-17.76),"%"))
```

```
## Accuracy of model 2:  82.24 %
```

Prediction and Submission:
Model 1 has higher accuracy than model 2. Hence, for further use, we will go ahead with model 1 for prediction.

```
pred <- predict(model1, newdata =  df_test)
result <- data.frame(id= df_test$ID ,smoking = df_test$smoking)
result[1:20,]
```

```
##    id smoking
## 1   1       0
## 2   4       0
## 3   5       0
## 4  10       0
## 5  13       1
## 6  14       0
## 7  21       1
## 8  30       0
## 9  37       0
## 10 43       1
## 11 44       0
## 12 49       0
## 13 54       0
## 14 63       0
## 15 65       0
## 16 67       1
## 17 73       0
## 18 75       1
## 19 77       1
## 20 82       0
```

DECISION TREE

```
library(party)
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

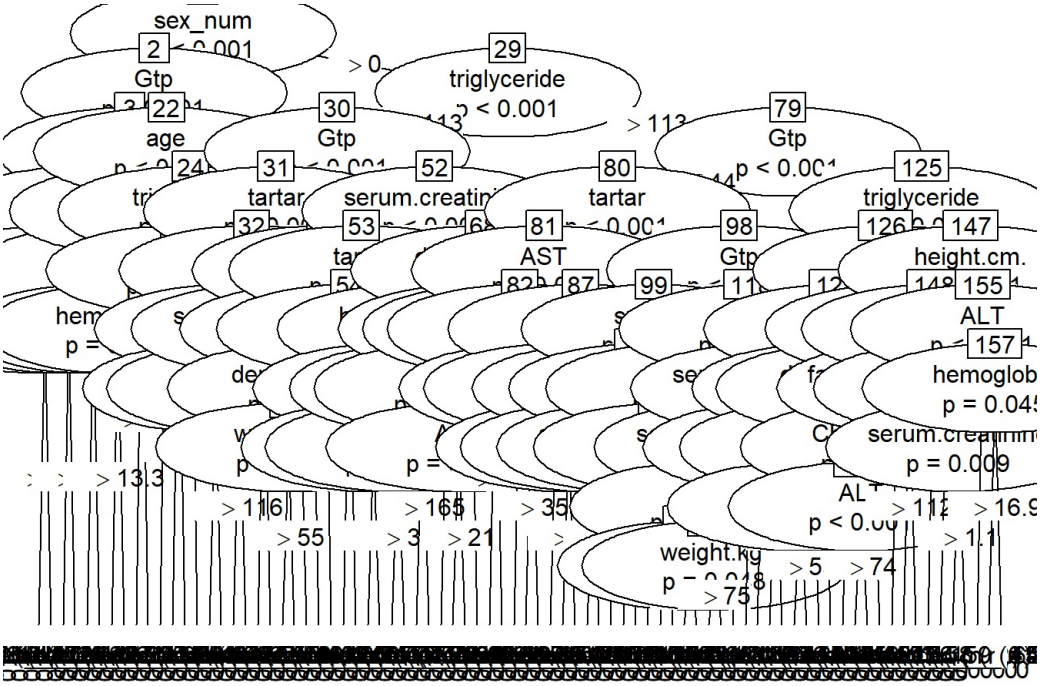
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
##
## Attaching package: 'strucchange'
```

```
## The following object is masked from 'package:stringr':
##
##     boundary
```

```
model3 <- ctree(factor(smoking) ~ age + height.cm. + weight.kg.+ sex_num +
                          eyesight.left. + eyesight.right. + hearing.left. + hearing.right. +
                          systolic + relaxation +
                          Cholesterol + triglyceride + HDL + LDL +
                          fasting.blood.sugar + hemoglobin + Urine.protein + serum.creatinine+
                          AST + ALT + Gtp +
                          dental.caries + tartar, data = df_train)

plot(model3)
```




```
predict_s <- predict(model3, df_test)
head(predict_s)
```

```
## [1] 0 0 1 1 1 0
## Levels: 0 1
```

```
cm3 <- table(df_test$smoking, predict_s)
cm3
```

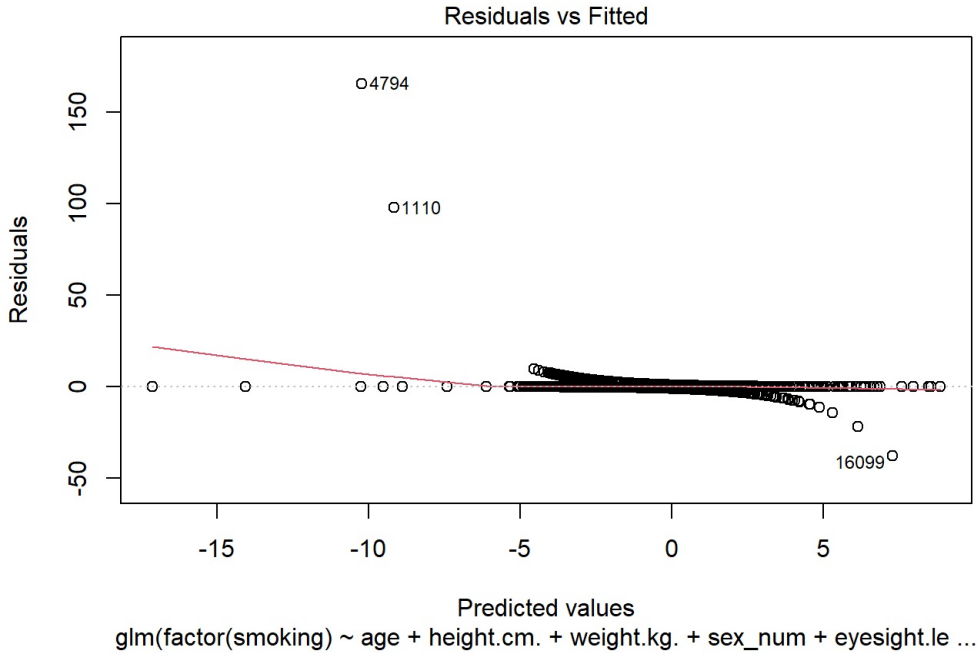
```
##    predict_s
##        0    1
##   0 8016 2555
##   1 1771 4366
```

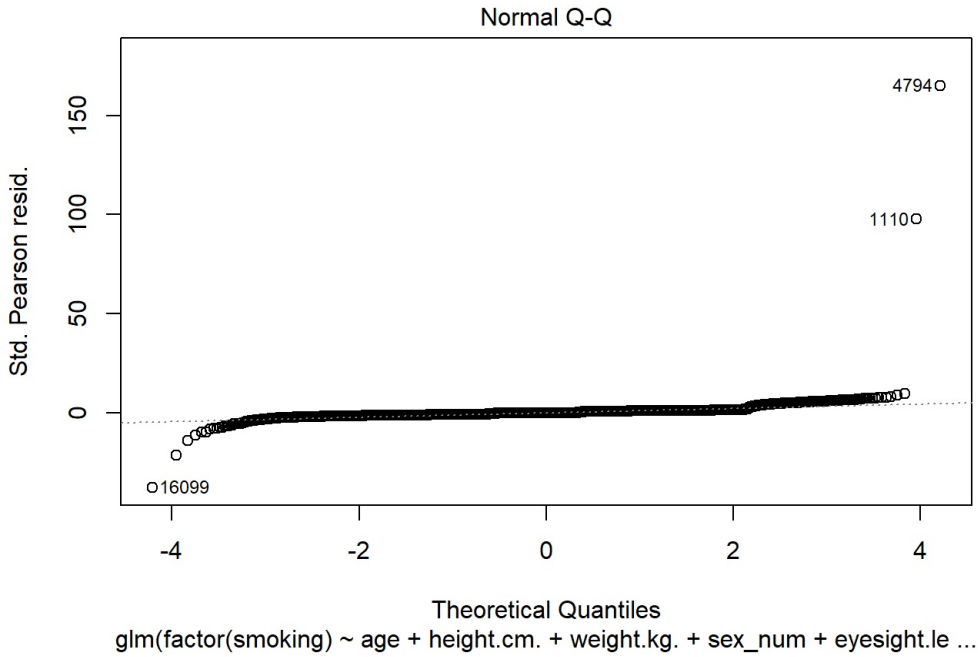
```
accuracy3 <- mean(predict_s!=df_test$smoking)
print(paste('Accuracy: ', 1-accuracy3))
```

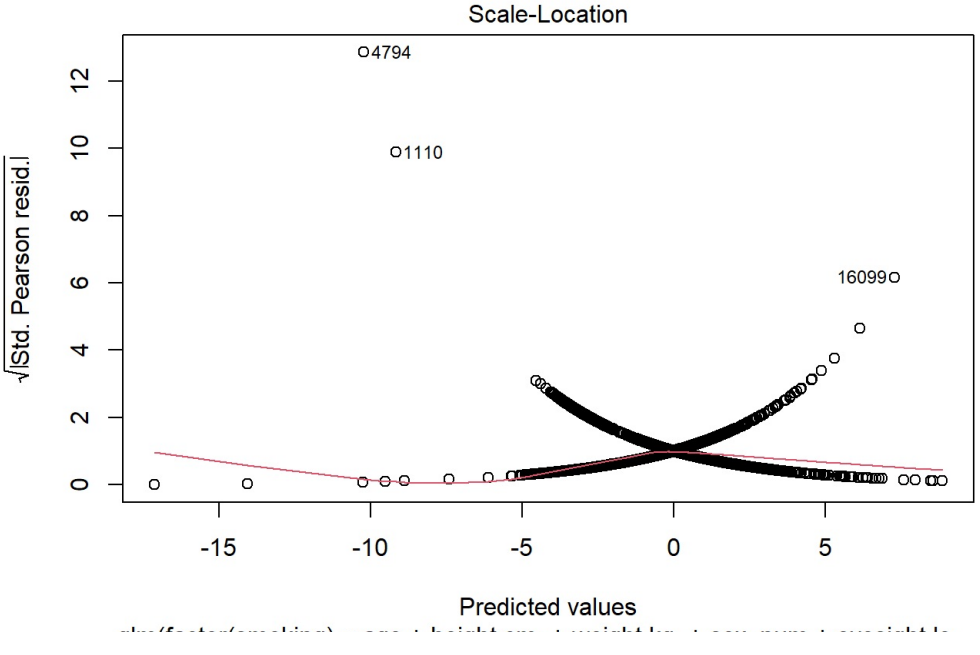
```
## [1] "Accuracy:  0.741082116351448"
```

Logistic Regression

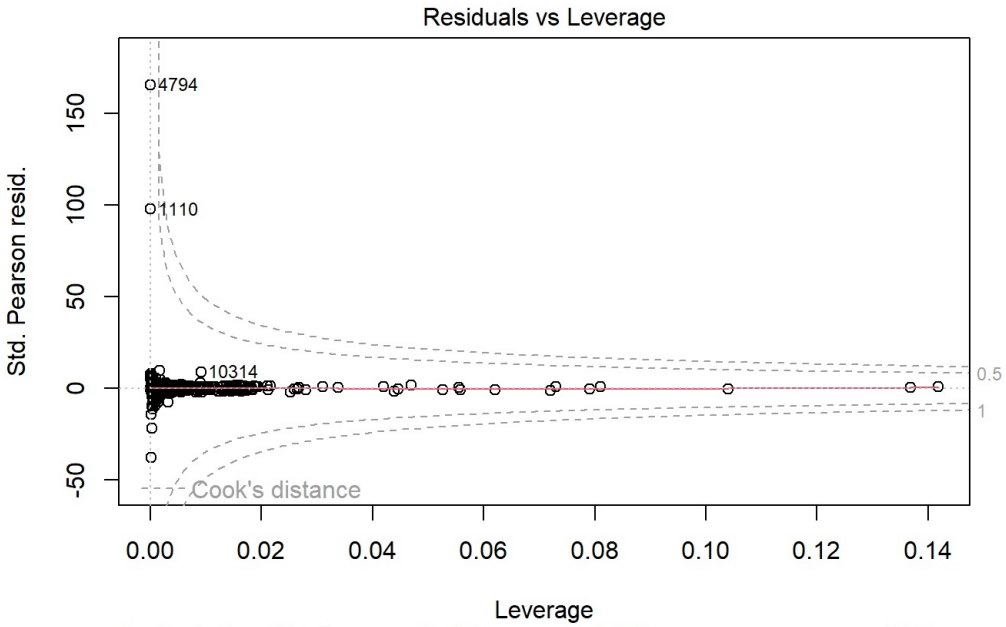
```
model4 <- glm(factor(smoking) ~ age + height.cm. + weight.kg.+ sex_num +
                          eyesight.left. + eyesight.right. + hearing.left. + hearing.right. +
                          systolic + relaxation +
                          Cholesterol + triglyceride + HDL + LDL +
                          fasting.blood.sugar + hemoglobin + Urine.protein + serum.creatinine+
                          AST + ALT + Gtp +
                          dental.caries + tartar, data = df_train, family="binomial")

plot(model4)
```


Residuals vs Fitted
Residuals
Predicted values
glm(factor(smoking) ~ age + height.cm. + weight.kg. + sex_num + eyesight.le ...
4794
1110
16099


Normal Q-Q
Std. Pearson resid.
Theoretical Quantiles
glm(factor(smoking) ~ age + height.cm. + weight.kg. + sex_num + eyesight.le ...
4794
1110
16099


Scale-Location
√|Std. Pearson resid.|
Predicted values
4794
1110
16099

glm(factor(smoking) ~ age + height.cm. + weight.kg. + sex_num + eyesight.le ...



glm(factor(smoking) ~ age + height.cm. + weight.kg. + sex_num + eyesight.le ...

```
predict_s2 <- predict(model4, df_test)
predict_s2<- ifelse(predict_s2 >0.5,1,0)
head(predict_s2)
```

```
## 1 2 3 4 5 6
## 0 0 0 0 1 0
```

```
cm4 <- table(df_test$smoking, predict_s2)
cm4
```

```
##    predict_s2
##        0    1
##   0 9701  870
##   1 3901 2236
```

```
accuracy4 <- mean(predict_s2 == df_test$smoking)
print(paste('Accuracy: ', accuracy4))
```

```
## [1] "Accuracy:  0.714448168542016"
```

Conclusion:

Smoking is injurious to health. Here are some observations supporting this statement:

Higher eye power has been observed among smokers
Hyper-intensive and high blood pressure tend to be correlated with smokers. This can cause heart diseases.
The good cholesterol is seen less in smokers.
Triglyceride is seen at a risk level among smokers.
Gtp, a metric of liver function is seen to be at a risk level among smokers.

```
predict1 = predict(model1, df_test)
predict2 = predict(model2, df_test)
```

```
acc1 <- mean(predict1 != df_test$smoking)
acc2 <- mean(predict2 != df_test$smoking)
cat(paste("Accuracy of model 1:", (1-acc1)*100, "%\n"))
```

```
## Accuracy of model 1: 82.4994014843189 %
```

```
cat(paste("Accuracy of model 2:", (1-acc2)*100, "%"))
```

```
## Accuracy of model 2: 82.4215944457745 %
```

```
set.seed(100)
control1 <- trainControl(sampling="rose",method="repeatedcv", number=5, repeats=5)
bagCART_model <- train(factor(smoking) ~ age + height.cm. + weight.kg.+ sex_num +
                         systolic + relaxation +
                         Cholesterol + triglyceride + HDL + LDL +
                         fasting.blood.sugar + hemoglobin + serum.creatinine+
                         AST + ALT + Gtp +
                         dental.caries, data=df_train, method="treebag", metric="Accuracy", trControl=control1)
#Predictions on the test set
predictTest = predict(bagCART_model, df_test)
```

```
acc3 <- mean(predictTest != df_test$smoking)
cat(paste("Accuracy of model 3:", (1-acc3)*100, "%"))
```

```
## Accuracy of model 3: 70.8702418003352 %
```

Importing the dataset

```
data <- read.csv("Medicalpremium.csv")
head(data)
```

```
##   Age Diabetes BloodPressureProblems AnyTransplants AnyChronicDiseases Height
## 1  45        0                    0              0                  0    155
## 2  60        1                    0              0                  0    180
## 3  36        1                    1              0                  0    158
## 4  52        1                    1              0                  1    183
## 5  38        0                    0              0                  1    166
## 6  30        0                    0              0                  0    160
##   Weight KnownAllergies HistoryOfCancerInFamily NumberOfMajorSurgeries
## 1     57              0                       0                      0
## 2     73              0                       0                      0
## 3     59              0                       0                      1
## 4     93              0                       0                      2
## 5     88              0                       0                      1
## 6     69              1                       0                      1
##   PremiumPrice
## 1        25000
## 2        29000
## 3        23000
## 4        28000
## 5        23000
## 6        23000
```

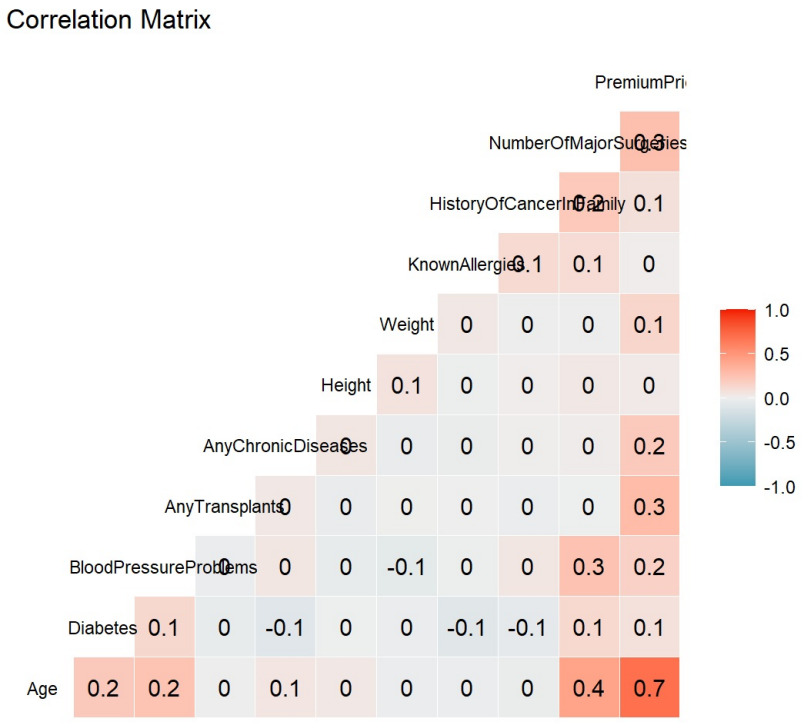
```
str(data)
```

```
## 'data.frame':    986 obs. of  11 variables:
##  $ Age                   : int  45 60 36 52 38 30 33 23 48 38 ...
##  $ Diabetes              : int  0 1 1 1 0 0 0 0 1 0 ...
##  $ BloodPressureProblems  : int  0 0 1 1 0 0 0 0 0 0 ...
##  $ AnyTransplants         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AnyChronicDiseases     : int  0 0 0 1 1 0 0 0 0 0 ...
##  $ Height                : int  155 180 158 183 166 160 150 181 169 182 ...
##  $ Weight                : int  57 73 59 93 88 69 54 79 74 93 ...
##  $ KnownAllergies         : int  0 0 0 0 0 1 0 1 1 0 ...
##  $ HistoryOfCancerInFamily: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NumberOfMajorSurgeries : int  0 0 1 2 1 1 0 0 0 0 ...
##  $ PremiumPrice          : int  25000 29000 23000 28000 23000 23000 21000 15000 23000 23000 ...
```

```
dim(data)
```

```
## [1] 986  11
```

```
ggcorr(data, label = T, color = "black", size = 3)+
  labs(title = "Correlation Matrix")
```



Factorizing various columns

```
data$Diabetes <- as.factor(data$Diabetes)
data$BloodPressureProblems <- as.factor(data$BloodPressureProblems)
data$AnyTransplants <- as.factor(data$AnyTransplants)
data$AnyChronicDiseases <- as.factor(data$AnyChronicDiseases)
data$KnownAllergies <- as.factor(data$KnownAllergies)
data$HistoryOfCancerInFamily <- as.factor(data$HistoryOfCancerInFamily)
data$NumberOfMajorSurgeries <- as.factor(data$NumberOfMajorSurgeries)
head(data)
```

```
##   Age Diabetes BloodPressureProblems AnyTransplants AnyChronicDiseases Height
## 1  45        0                     0              0                  0    155
## 2  60        1                     0              0                  0    180
## 3  36        1                     1              0                  0    158
## 4  52        1                     1              0                  1    183
## 5  38        0                     0              0                  1    166
## 6  30        0                     0              0                  0    160
##   Weight KnownAllergies HistoryOfCancerInFamily NumberOfMajorSurgeries
## 1     57              0                       0                      0
## 2     73              0                       0                      0
## 3     59              0                       0                      1
## 4     93              0                       0                      2
## 5     88              0                       0                      1
## 6     69              1                       0                      1
##   PremiumPrice
## 1        25000
## 2        29000
## 3        23000
## 4        28000
## 5        23000
## 6        23000
```

```
str(data)
```

```
## 'data.frame':    986 obs. of  11 variables:
##  $ Age                   : int  45 60 36 52 38 30 33 23 48 38 ...
##  $ Diabetes              : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 1 ...
##  $ BloodPressureProblems : Factor w/ 2 levels "0","1": 1 1 2 2 1 1 1 1 1 1 ...
##  $ AnyTransplants        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ AnyChronicDiseases    : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 1 1 1 1 ...
##  $ Height                : int  155 180 158 183 166 160 150 181 169 182 ...
##  $ Weight                : int  57 73 59 93 88 69 54 79 74 93 ...
##  $ KnownAllergies        : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 2 2 1 ...
##  $ HistoryOfCancerInFamily: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ NumberOfMajorSurgeries : Factor w/ 4 levels "0","1","2","3": 1 1 2 3 2 2 1 1 1 1 ...
##  $ PremiumPrice          : int  25000 29000 23000 28000 23000 23000 21000 15000 23000 23000 ...
```

Calculating BMI

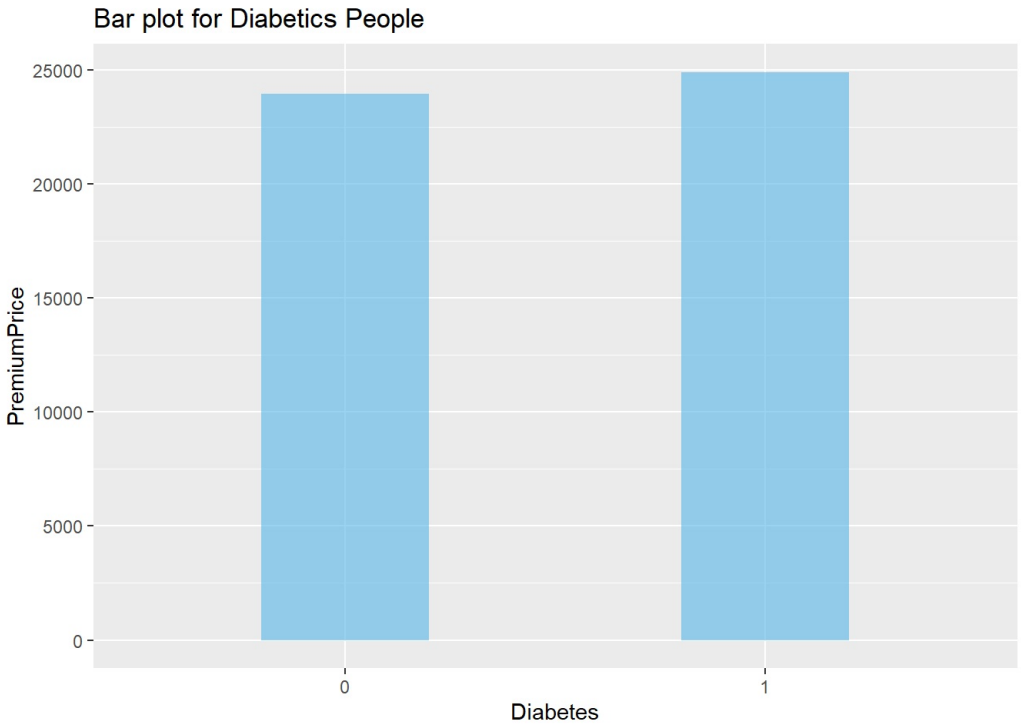
```
data$bmi <- 10000*(data$Weight/(data$Height)^2)
```

Assigning categories to different BMI ranges

```
data <- data %>%
  mutate( bmiCategory = case_when(
                                  bmi<18.49999 ~ "under weight",
                                  bmi>18.5 & bmi<24.99999 ~ "normal weight",
                                  bmi>25 & bmi<29.99999 ~ "over weight",
                                  bmi>30 ~ "obesity"
                                  ))
```

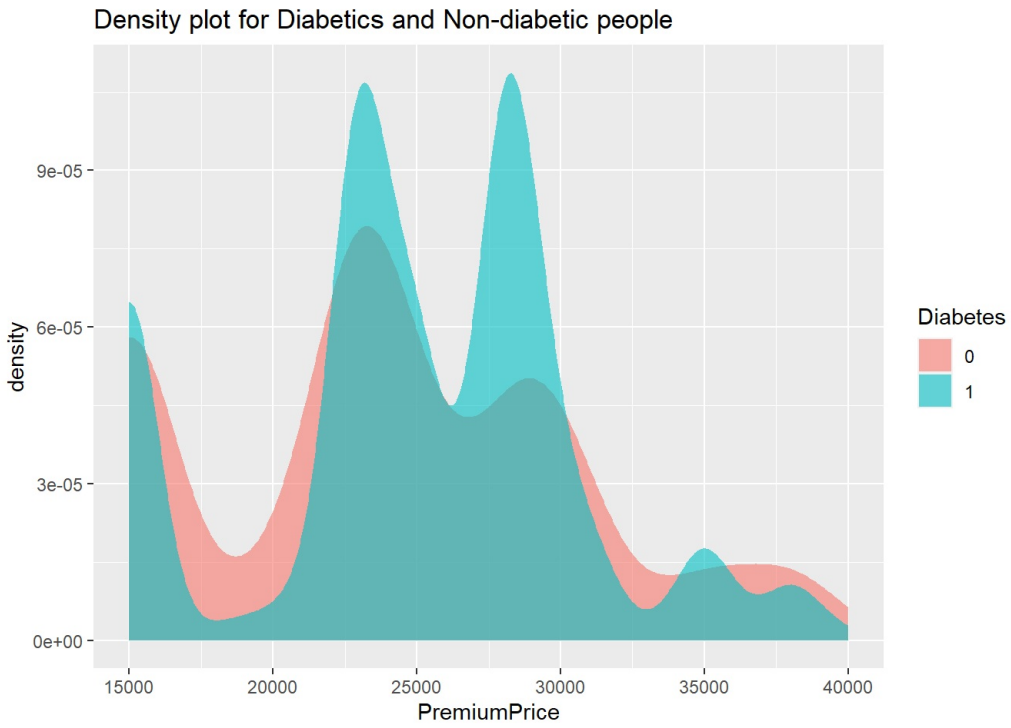
Diabetics People Premium Analysis

```
data %>%
  select(Diabetes,PremiumPrice) %>%
  group_by(Diabetes) %>%
  summarise( PremiumPrice = mean(PremiumPrice)) %>%
  ggplot(.,aes(Diabetes,PremiumPrice))+
  geom_bar(stat = "identity",width = 0.4, fill = "#56B4E9", alpha = 0.6)+
  labs(title = "Bar plot for Diabetics People")
```



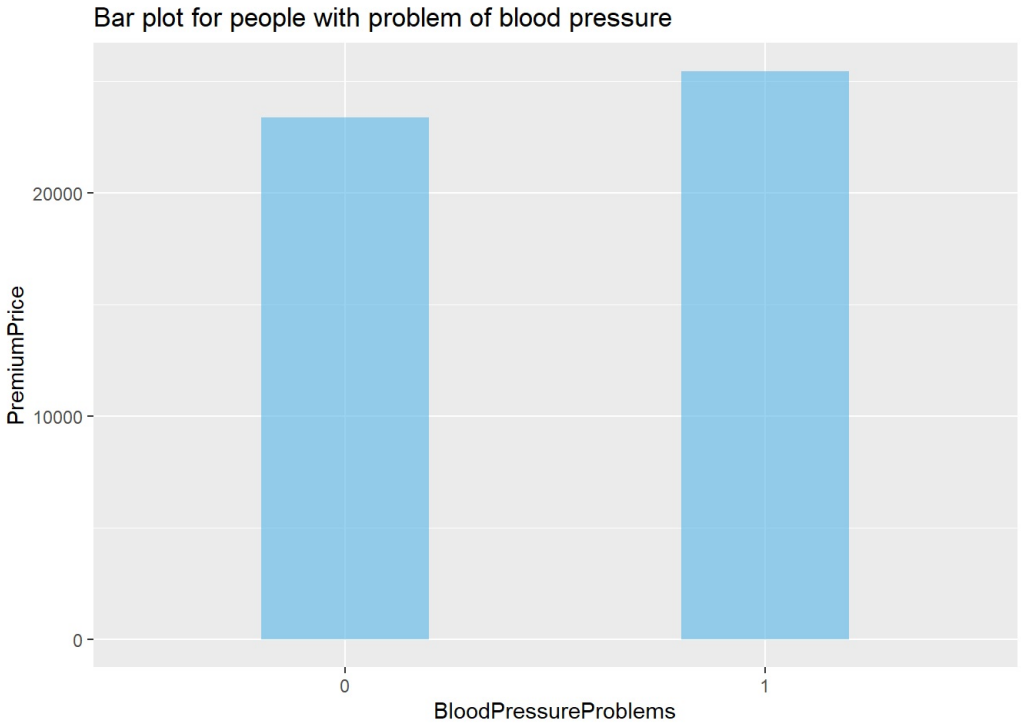
Distribution of Premium Prices for Diabetic and Non-Diabetic People

```
ggplot(data, aes(PremiumPrice))+
  geom_density(aes(fill = Diabetes), color = NA, alpha = 0.6)+
  labs(title = "Density plot for Diabetics and Non-diabetic people")
```



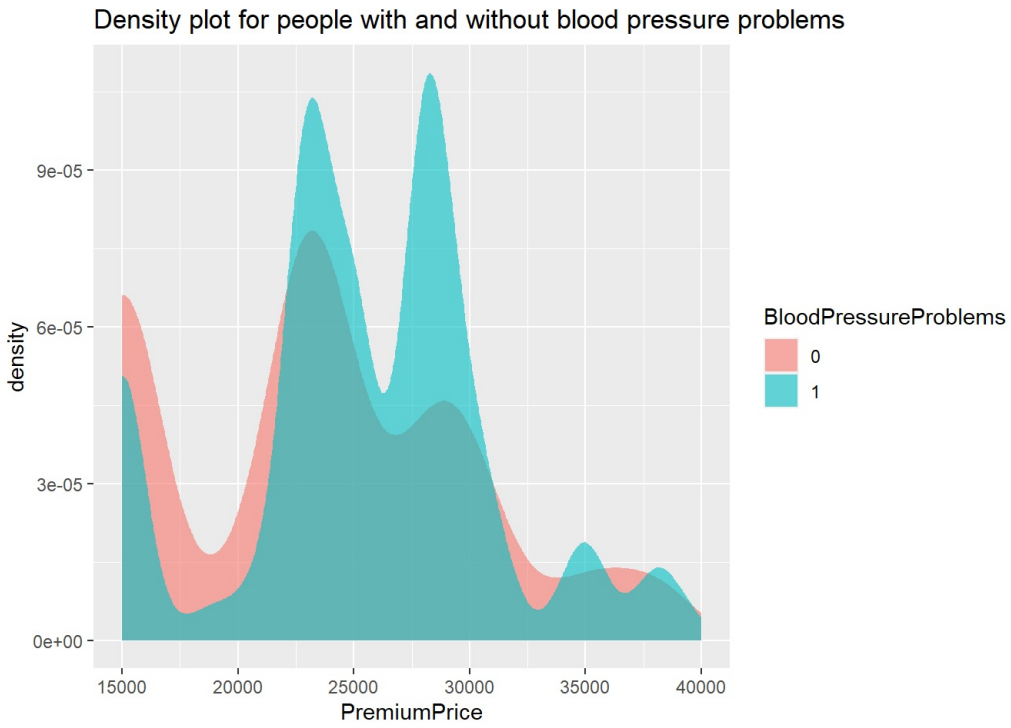
Blood Pressure Patients Premium Analysis

```
data %>%
  select(BloodPressureProblems,PremiumPrice) %>%
  group_by(BloodPressureProblems) %>%
  summarise( PremiumPrice = mean(PremiumPrice)) %>%
  ggplot(.,aes(BloodPressureProblems,PremiumPrice))+
  geom_bar(stat = "identity",width = 0.4, fill = "#56B4E9", alpha = 0.6)+
  labs(title = "Bar plot for people with problem of blood pressure")
```



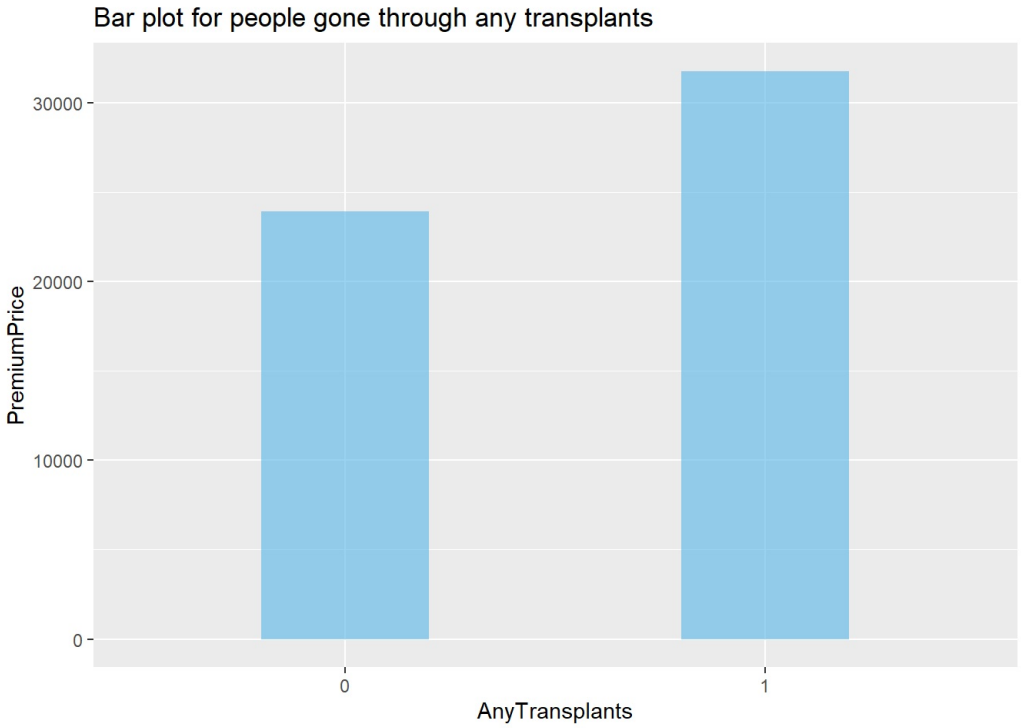
Distribution of Premium Prices for people with and without blood pressure problems

```
ggplot(data, aes(PremiumPrice))+
  geom_density(aes(fill = BloodPressureProblems), color = NA, alpha = 0.6)+
  labs(title = "Density plot for people with and without blood pressure problems")
```



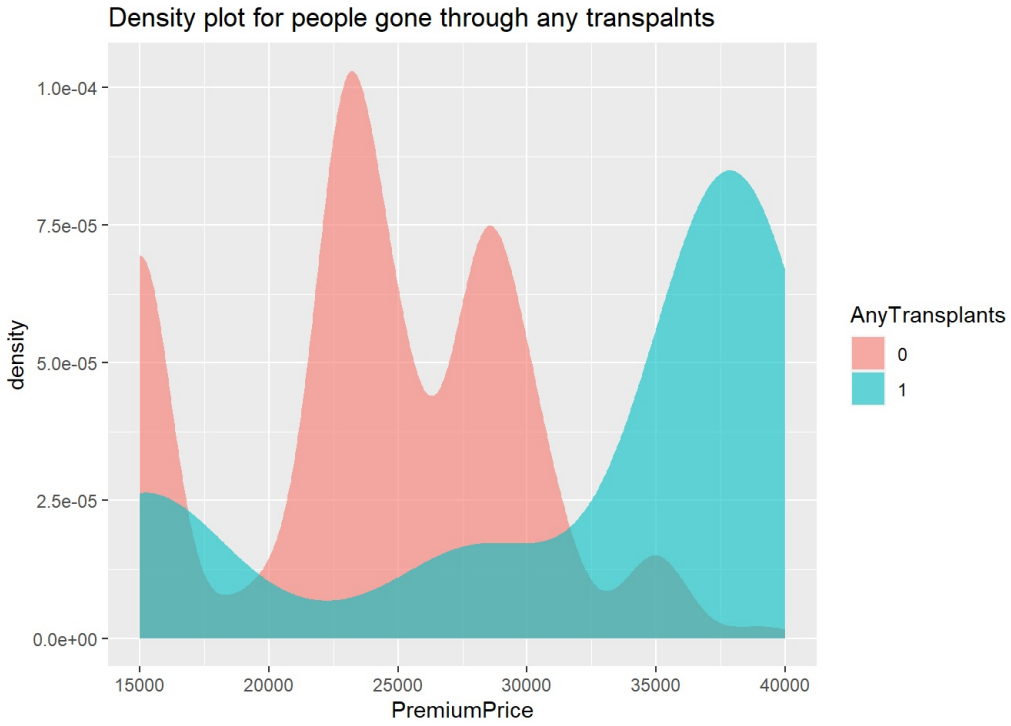
People Gone Through Any Transplants Premium Analysis

```
data %>%
  select(AnyTransplants,PremiumPrice) %>%
  group_by(AnyTransplants) %>%
  summarise( PremiumPrice = mean(PremiumPrice)) %>%
  ggplot(.,aes(AnyTransplants,PremiumPrice))+
  geom_bar(stat = "identity",width = 0.4, fill = "#56B4E9", alpha = 0.6)+
  labs(title = "Bar plot for people gone through any transplants")
```



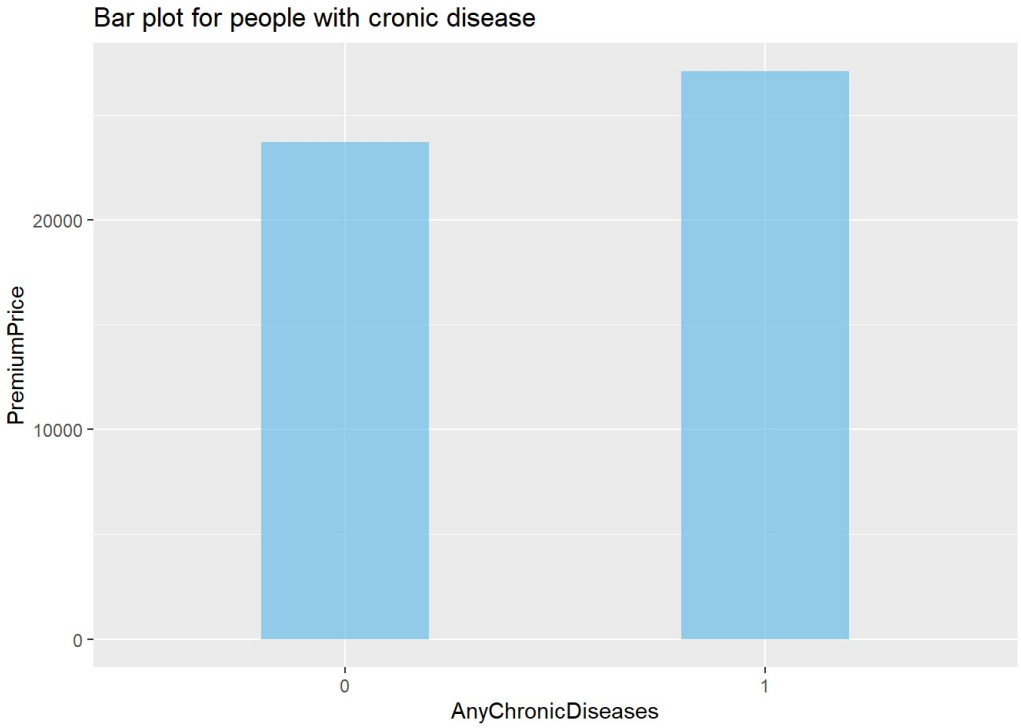
Distribution of Premium Prices for People Gone Through Any Transplants vs Those Who haven't Gone Through Any Transplants

```
ggplot(data, aes(PremiumPrice))+
  geom_density(aes(fill = AnyTransplants), color = NA, alpha = 0.6)+
  labs(title = "Density plot for people gone through any transpalnts")
```



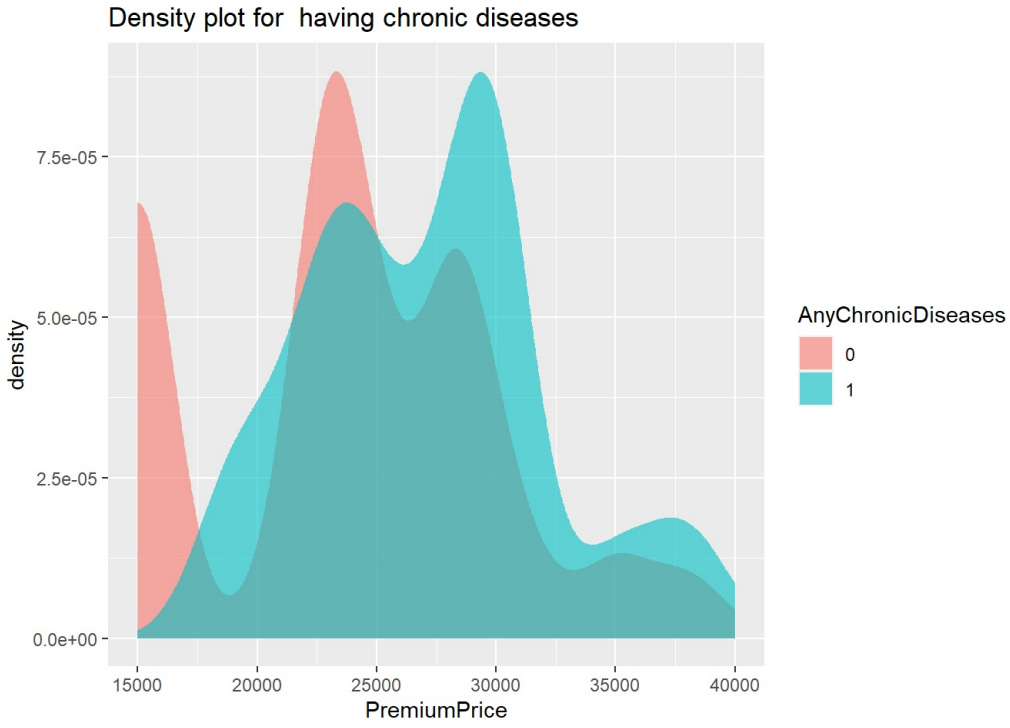
People With Chronic Disease Premium Analysis

```
data %>%
  select(AnyChronicDiseases,PremiumPrice) %>%
  group_by(AnyChronicDiseases) %>%
  summarise( PremiumPrice = mean(PremiumPrice)) %>%
  ggplot(.,aes(AnyChronicDiseases,PremiumPrice))+
  geom_bar(stat = "identity",width = 0.4, fill = "#56B4E9", alpha = 0.6)+
  labs(title = "Bar plot for people with cronic disease")
```



Distribution of Premium Prices for People With Chronic Disease and People With No Chronic Disease

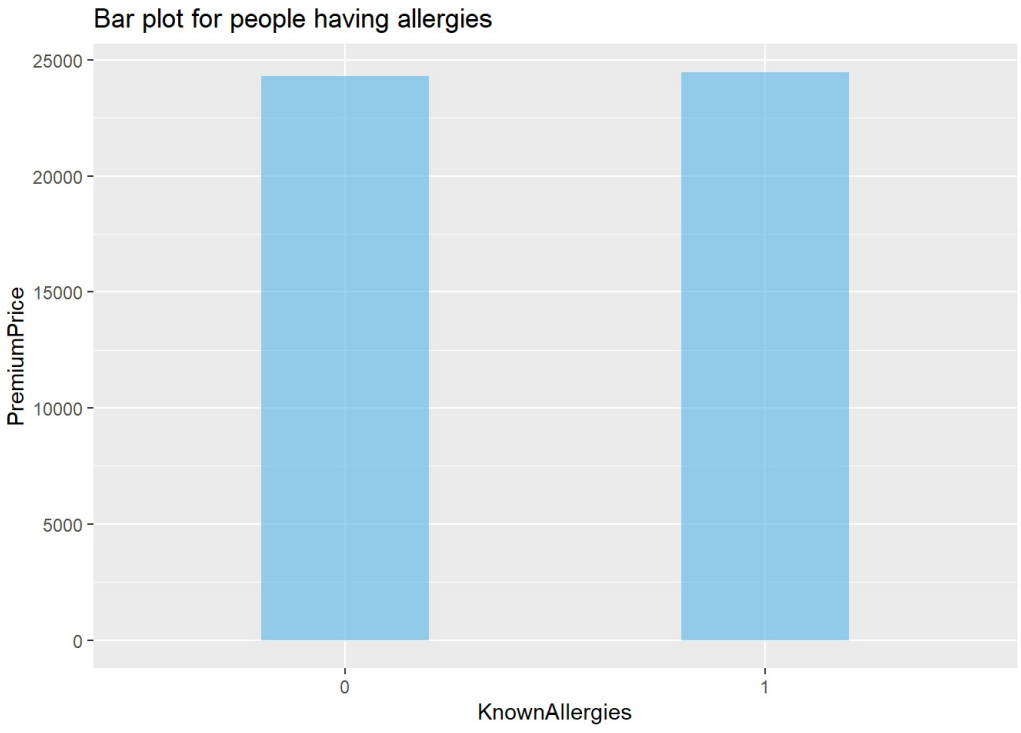
```
ggplot(data, aes(PremiumPrice))+
  geom_density(aes(fill = AnyChronicDiseases), color = NA, alpha = 0.6)+
  labs(title = "Density plot for  having chronic diseases")
```



Allergy Patients Premium Analysis

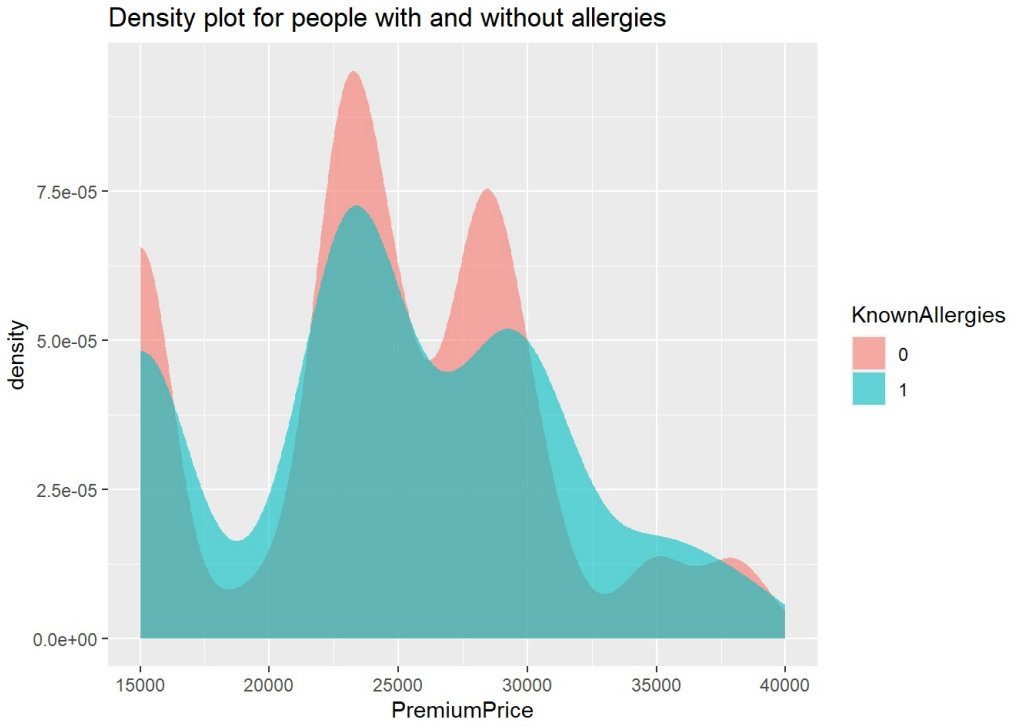
Average Difference in Premium Prices for Allergy patients and No Allergy Patients

```
data %>%
  select(KnownAllergies,PremiumPrice) %>%
  group_by(KnownAllergies) %>%
  summarise( PremiumPrice = mean(PremiumPrice)) %>%
  ggplot(.,aes(KnownAllergies,PremiumPrice))+
  geom_bar(stat = "identity",width = 0.4, fill = "#56B4E9", alpha = 0.6)+
  labs(title = "Bar plot for people having allergies")
```



Distribution of Premium Prices for Allergy Patients and No Allergy Patients

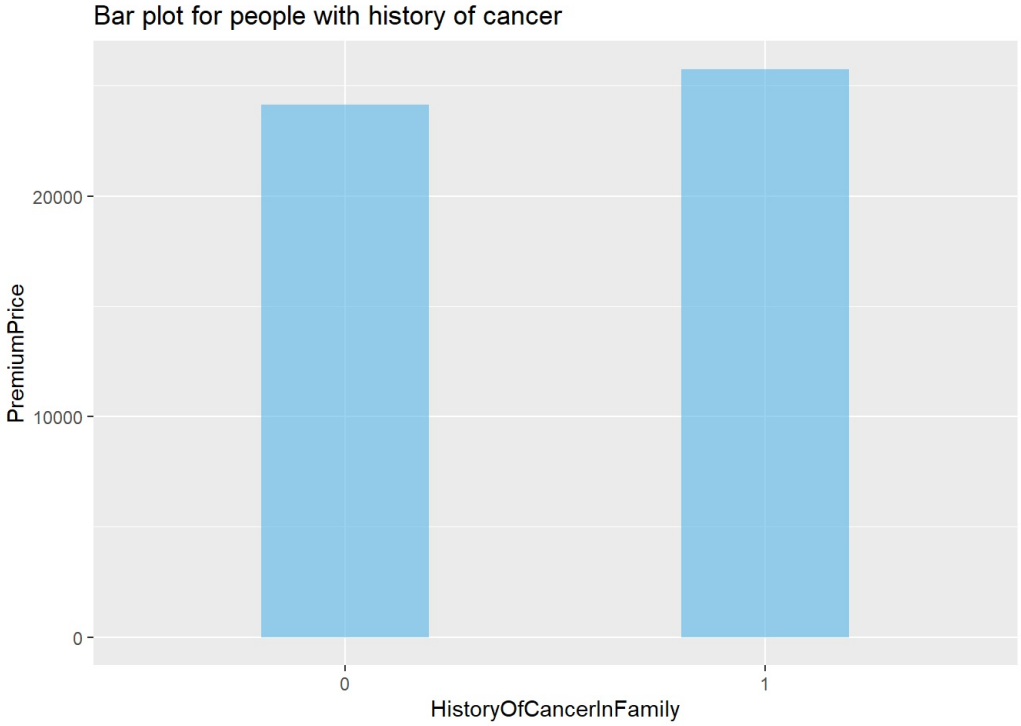
```
ggplot(data, aes(PremiumPrice))+
  geom_density(aes(fill = KnownAllergies), color = NA, alpha = 0.6)+
  labs(title = "Density plot for people with and without allergies")
```



Patients with History of Cancer in Family Premium Analysis

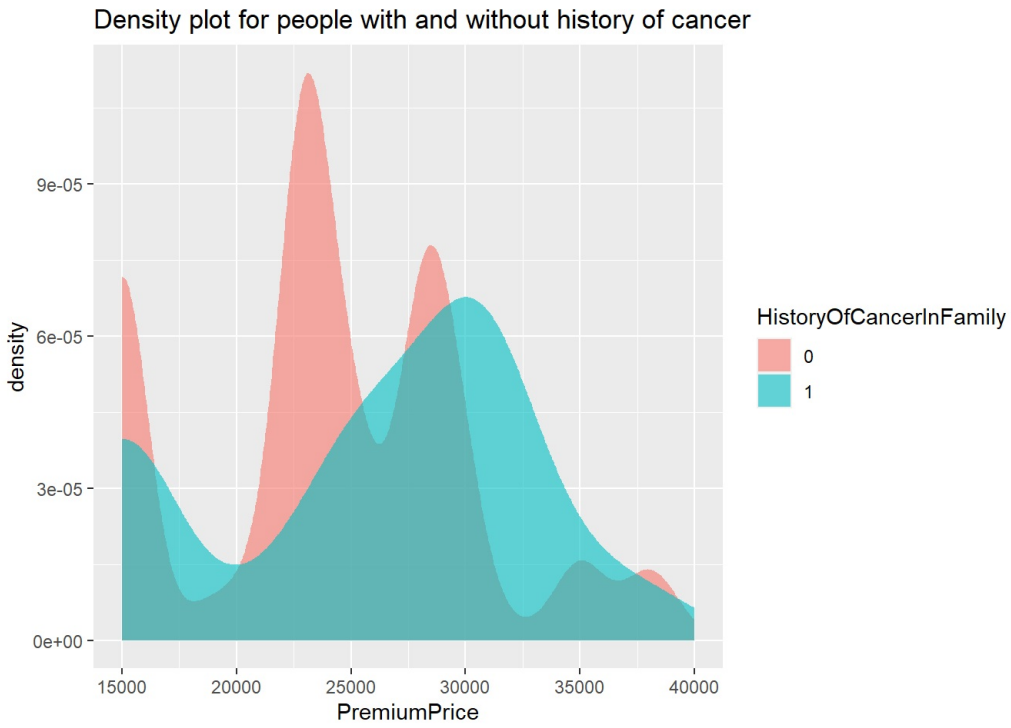
Average Difference in Premium Prices for Patients with History of Cancer and Patients without History of Cancer

```
data %>%
  select(HistoryOfCancerInFamily,PremiumPrice) %>%
  group_by(HistoryOfCancerInFamily) %>%
  summarise( PremiumPrice = mean(PremiumPrice)) %>%
  ggplot(.,aes(HistoryOfCancerInFamily,PremiumPrice))+
  geom_bar(stat = "identity",width = 0.4, fill = "#56B4E9", alpha = 0.6)+
  labs(title = "Bar plot for people with history of cancer")
```



Distribution of Premium Prices for Patients with History of Cancer and Patients without History of Cancer

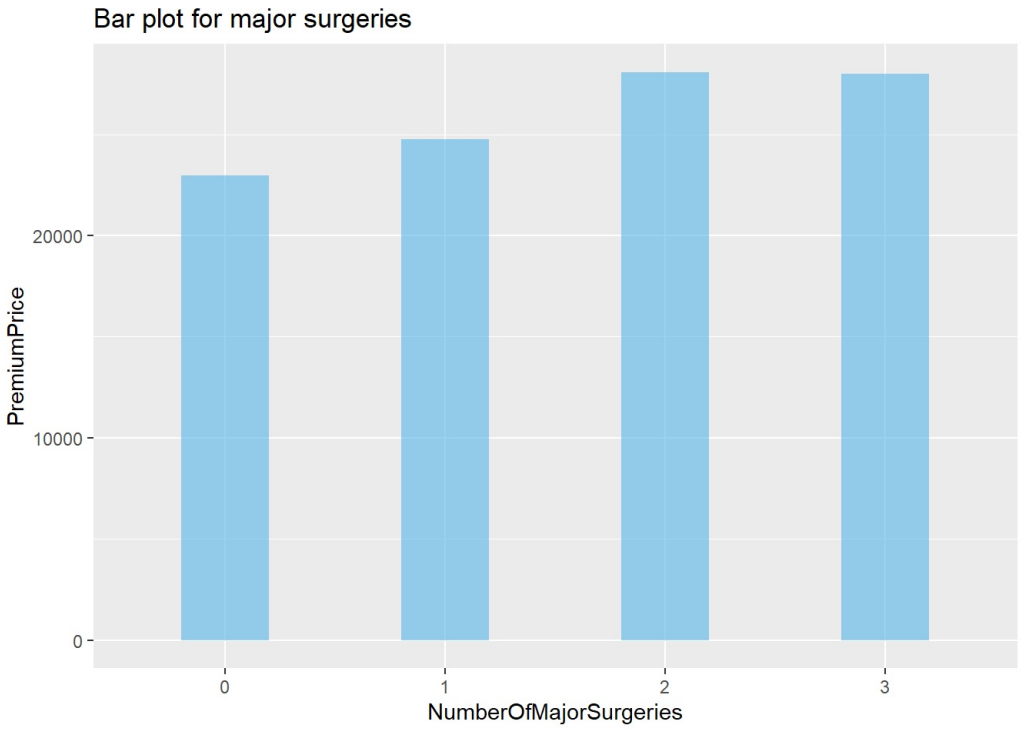
```
ggplot(data, aes(PremiumPrice))+
  geom_density(aes(fill = HistoryOfCancerInFamily), color = NA, alpha = 0.6)+
  labs(title = "Density plot for people with and without history of cancer")
```



People Gone Through Major Surgeries Premium Analysis

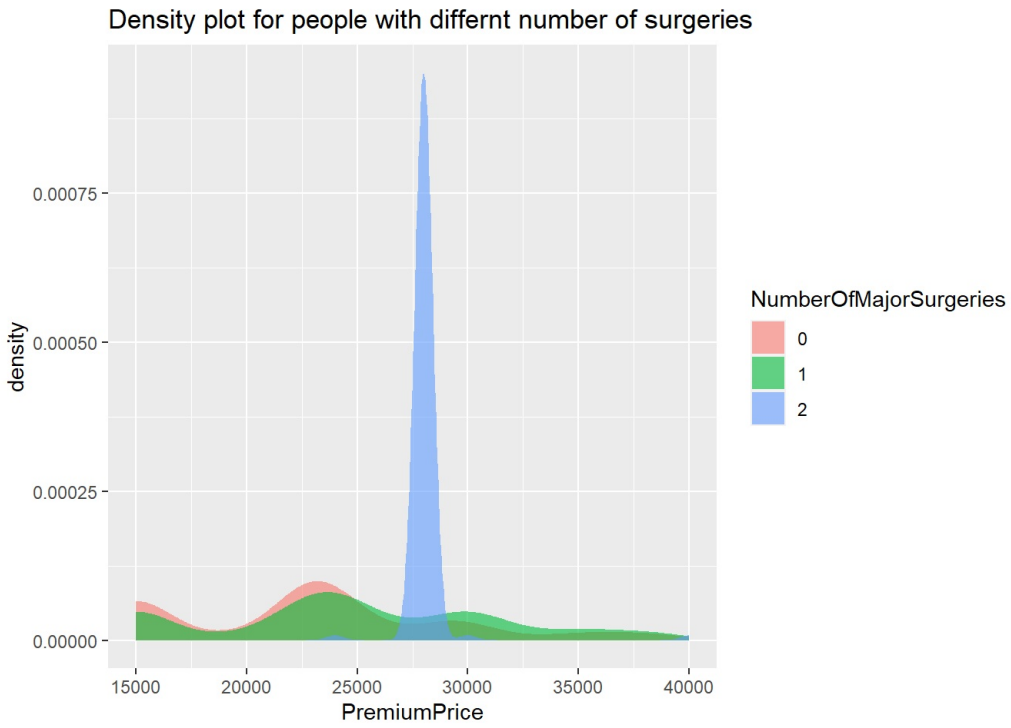
Average Difference in Premium Prices for People gone through major surgeries

```
data %>%
  select(NumberOfMajorSurgeries,PremiumPrice) %>%
  group_by(NumberOfMajorSurgeries) %>%
  summarise( PremiumPrice = mean(PremiumPrice)) %>%
  ggplot(.,aes(NumberOfMajorSurgeries,PremiumPrice))+
  geom_bar(stat = "identity",width = 0.4, fill = "#56B4E9", alpha = 0.6)+
  labs(title = "Bar plot for major surgeries")
```



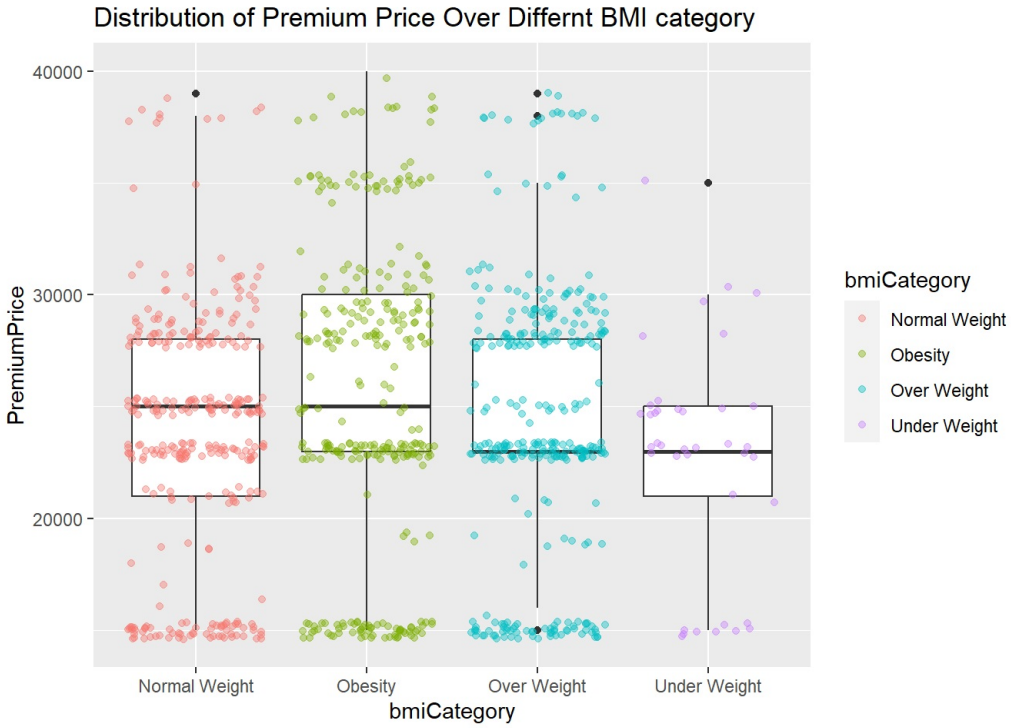
Distribution of Premium Prices for People gone through major surgeries

```
ggplot(data %>%
         select(NumberOfMajorSurgeries,PremiumPrice) %>%
         filter(!NumberOfMajorSurgeries == 3),
         aes(PremiumPrice))+
  geom_density(aes(fill = NumberOfMajorSurgeries), color = NA, alpha = 0.6)+
  labs(title = "Density plot for people with differnt number of surgeries")
```



Box Plot for Different BMI category

```
data %>%
  mutate(bmiCategory = str_to_title(bmiCategory)) %>%

  ggplot(aes(bmiCategory, PremiumPrice))+
  geom_boxplot()+
  geom_jitter(aes(color = bmiCategory),alpha = 0.4)+
  labs(title = "Distribution of Premium Price Over Differnt BMI category")
```



Prediction Model

```
library(randomForest)
library(caTools)
library(rsq)
```

```
## Warning: package 'rsq' was built under R version 4.2.3
```

```
data$PremiumPrice <- as.factor(data$PremiumPrice)
summary(data)
```

```
##       Age        Diabetes BloodPressureProblems AnyTransplants
##  Min.   :18.00   0:572    0:524                 0:931
##  1st Qu.:30.00   1:414    1:462                 1: 55
##  Median :42.00
##  Mean   :41.75
##  3rd Qu.:53.00
##  Max.   :66.00
##
##  AnyChronicDiseases     Height          Weight       KnownAllergies
##  0:808              Min.   :145.0   Min.   : 51.00   0:774
##  1:178              1st Qu.:161.0   1st Qu.: 67.00   1:212
##                     Median :168.0   Median : 75.00
##                     Mean   :168.2   Mean   : 76.95
##                     3rd Qu.:176.0   3rd Qu.: 87.00
##                     Max.   :188.0   Max.   :132.00
##
##  HistoryOfCancerInFamily NumberOfMajorSurgeries  PremiumPrice      bmi
##  0:870                   0:479                  23000  :249   Min.   :15.16
##  1:116                   1:372                  15000  :202   1st Qu.:23.39
##                          2:119                  28000  :132   Median :27.16
##                          3: 16                  25000  :103   Mean   :27.46
##                                                 29000  : 72   3rd Qu.:30.76
##                                                 30000  : 47   Max.   :50.00
##                                                 (Other):181
##  bmiCategory
##  Length:986
##  Class :character
##  Mode  :character
##
##
##
##
```

Spliting Data into two Subset for Training and Testing use.

```
sample <- sample.split(data$PremiumPrice, SplitRatio = 0.75)
train <- subset(data, sample == TRUE)
test <- subset(data, sample == FALSE)
dim(train)
```

```
## [1] 743  13
```

```
dim(test)
```

```
## [1] 243  13
```

Random Forest

```
set.seed(123)
rf <- randomForest(
  PremiumPrice ~ .,
  data=train
)
print(rf)
```

```
##
## Call:
##  randomForest(formula = PremiumPrice ~ ., data = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 8.88%
## Confusion matrix:
##       15000 16000 17000 18000 19000 20000 21000 22000 23000 24000 25000 26000
## 15000   152     0     0     0     0     0     0     0     0     0     0     0
## 16000     2     0     0     0     0     0     0     0     0     0     0     0
## 17000     0     0     0     0     0     0     0     0     0     0     0     0
## 18000     1     0     0     0     0     0     0     0     0     0     0     0
```

```
## 19000     0     0     0     0    11     0     0     0     0     0     0     0
## 20000     0     0     0     0     0     0     0     0     0     0     0     0
## 21000     1     0     0     0     0     0    12     0     6     0     0     0
## 22000     0     0     0     0     0     0     0     0     0     0     0     0
## 23000     0     0     0     0     0     0     0     0   187     0     0     0
## 24000     1     0     0     0     0     0     0     0     0     0     0     0
## 25000     0     0     0     0     1     0     0     0     2     0    69     0
## 26000     2     0     0     0     0     0     0     0     2     0     0     0
## 27000     0     0     0     0     0     0     0     0     0     0     0     0
## 28000     0     0     0     0     0     0     0     0     0     0     0     0
## 29000     0     0     0     0     0     0     0     0     0     0     0     0
## 30000     0     0     0     0     1     0     0     0     3     0     2     0
## 31000     0     0     0     0     0     0     0     1     0     0     1     0
## 32000     1     0     0     0     0     0     0     0     2     0     0     0
## 34000     0     0     0     0     0     0     0     0     2     0     0     0
## 35000     0     0     0     0     0     0     0     0     2     0     1     0
## 36000     1     0     0     0     0     0     0     0     0     0     0     0
## 38000     0     0     0     0     1     0     0     0     0     0     0     0
## 39000     3     0     0     0     0     0     0     0     1     0     0     0
## 40000     0     0     0     0     0     0     0     0     0     0     0     0
##       27000 28000 29000 30000 31000 32000 34000 35000 36000 38000 39000 40000
## 15000     0     0     0     0     0     0     0     0     0     0     0     0
## 16000     0     0     0     0     0     0     0     0     0     0     0     0
## 17000     0     0     0     0     0     0     0     0     0     1     0     0
## 18000     0     0     0     1     0     0     0     0     0     0     0     0
## 19000     0     0     0     0     0     0     0     0     0     0     0     0
## 20000     0     0     1     0     0     0     0     0     0     0     0     0
## 21000     0     0     0     0     0     0     0     0     0     1     0     0
## 22000     0     0     0     0     1     0     0     0     0     0     0     0
## 23000     0     0     0     0     0     0     0     0     0     0     0     0
## 24000     0     1     0     0     0     0     0     1     0     0     0     0
## 25000     0     0     0     1     4     0     0     0     0     0     0     0
## 26000     0     0     1     0     0     0     0     0     0     0     0     0
## 27000     0     0     0     0     0     0     0     0     0     1     0     0
## 28000     0    99     0     0     0     0     0     0     0     0     0     0
## 29000     0     0    54     0     0     0     0     0     0     0     0     0
## 30000     0     1     0    28     0     0     0     0     0     0     0     0
## 31000     0     0     0     1    20     0     0     0     0     0     0     0
## 32000     0     0     0     0     0     0     0     0     0     0     0     0
## 34000     0     0     0     0     0     0     0     0     0     0     0     0
## 35000     0     0     2     4     1     0     0    20     0     1     0     0
## 36000     0     0     0     0     0     0     0     1     0     0     0     0
## 38000     0     0     0     0     0     0     0     0     0    25     0     0
## 39000     0     0     0     0     0     0     0     0     0     0     0     0
## 40000     0     1     0     0     0     0     0     0     0     0     0     0
##       class.error
## 15000  0.00000000
## 16000  1.00000000
## 17000  1.00000000
## 18000  1.00000000
## 19000  0.00000000
## 20000  1.00000000
## 21000  0.40000000
## 22000  1.00000000
## 23000  0.00000000
## 24000  1.00000000
## 25000  0.10389610
## 26000  1.00000000
## 27000  1.00000000
## 28000  0.00000000
## 29000  0.00000000
## 30000  0.20000000
## 31000  0.13043478
## 32000  1.00000000
## 34000  1.00000000
## 35000  0.35483871
## 36000  1.00000000
## 38000  0.03846154
## 39000  1.00000000
## 40000  1.00000000
```

```
pred <- predict(rf, newdata = test[-11])
cm <- table(pred,obs = test[,11])
sum <- 0
for (i in 1:24){
    for(j in 1:24){
        if(i!=j){
            sum <-sum+cm[i,j]
        }
    }
}
sum
```

```
## [1] 21
```

```
print(paste("The Accuracy of Random Forest Model is",(243-sum)/2.43))
```

```
## [1] "The Accuracy of Random Forest Model is 91.358024691358"
```

The accuracy for predictions while using 75% data for training and 25% for testing has turned out to be around 92% using Random Forest model.

```
train$PremiumPrice <- as.numeric(as.character(train$PremiumPrice))
test$PremiumPrice <- as.numeric(as.character(test$PremiumPrice))
class (test$PremiumPrice)
```

```
## [1] "numeric"
```

```
class (train$PremiumPrice)
```

```
## [1] "numeric"
```

k fold cross validation with treebag method

```
set.seed(100)
control2 <- trainControl(method = "repeatedcv", number = 5, repeats = 5)
bagCART_model2 <- train(PremiumPrice ~ ., data = train, method = "treebag", metric = "RMSE", trControl = control2
)
```

```
# Predictions on the test set
predictTest2 <- predict(bagCART_model2, test)
rmse <- RMSE(predictTest2, test$PremiumPrice)
cat(paste("RMSE of model is:", rmse))
```

```
## RMSE of model is: 2570.19509751601
```

```
r2 <- R2(predictTest2, test$PremiumPrice)
cat(paste("R-squared of model is:", r2))
```

```
## R-squared of model is: 0.828062348333068
```

k fold cross validation with lasso method

```
set.seed(100)
control3 <- trainControl(method = "repeatedcv", number = 10, repeats = 10)
bagCART_model3 <- train(PremiumPrice ~ ., data = train, method = "lasso", metric = "RMSE", trControl = control3)
```

```
# Predictions on the test set
predictTest3 <- predict(bagCART_model3, test)
rmse3 <- RMSE(predictTest3, test$PremiumPrice)
cat(paste("RMSE of model is:", rmse3))
```

```
## RMSE of model is: 3597.35794120654
```

```
r23 <- R2(predictTest2, test$PremiumPrice)
cat(paste("R-squared of model is:", r23))
```

```
## R-squared of model is: 0.828062348333068
```