AutoEDA_Ver1

2014311048 백우현

2019년 3월 11일

```
library(data.table)
library(dplyr)
library(ggplot2)
```

Raw Data에 대한 기본적인 EDA와 편한 기능 자동 함수 Ver1

Data

key is a criterion of your data like ‘UserID’ , ‘Date’, or ‘Product Number’

```
train_data <- fread('insurance.csv')
str(train_data)
```

```
## Classes 'data.table' and 'data.frame':   1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : chr  "female" "male" "male" "male" ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : chr  "yes" "no" "no" "no" ...
##  $ region  : chr  "southwest" "southeast" "southeast" "northwest" ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
train_data$key <- paste0('No.',1:nrow(train_data)) # make a dummy key column
```

Get index

```
## get numerical / categorical / key  vaiables colnames & index ##

get_index <- function(data , key , chr_to_fac = T){
  data <- as.data.frame(data)
  key_idx <- which(colnames(data) %in% key == T )
  dat <- data[,-key_idx]

  if(chr_to_fac == T){
    dat <- dat %>% mutate_if(is.character, as.factor)
  }

  numerical_variables <- dat %>% select_if(is.numeric) %>% colnames()
  categorical_variables <- dat %>% select_if(is.factor) %>% colnames()

  numerical_idx <- which(colnames(data) %in% numerical_variables  == T)
  categorical_idx <- which(colnames(data) %in% categorical_variables == T )
  categorical_levels <- rep(0,length(categorical_idx))
  for ( i in 1:length(categorical_idx)){
    categorical_levels[i] <- nlevels( factor(data[,categorical_idx[i]]) )
  }


  result <- list(key = data.frame(key , index = key_idx ),
                 numerical = data.frame(numerical_variables , index =  numerical_idx ),
                 categorical = data.frame(categorical_variables , index = categorical_idx, leve
ls = categorical_levels))

  return(result)

}

index_list <- get_index(train_data, "key")
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
index_list
```

```
## $key
##   key index
## 1 key     8
##
## $numerical
##   numerical_variables index
## 1                 age     1
## 2                 bmi     3
## 3            children     4
## 4             charges     7
##
## $categorical
##   categorical_variables index levels
## 1                   sex     2      2
## 2                smoker     5      2
## 3                region     6      4
```

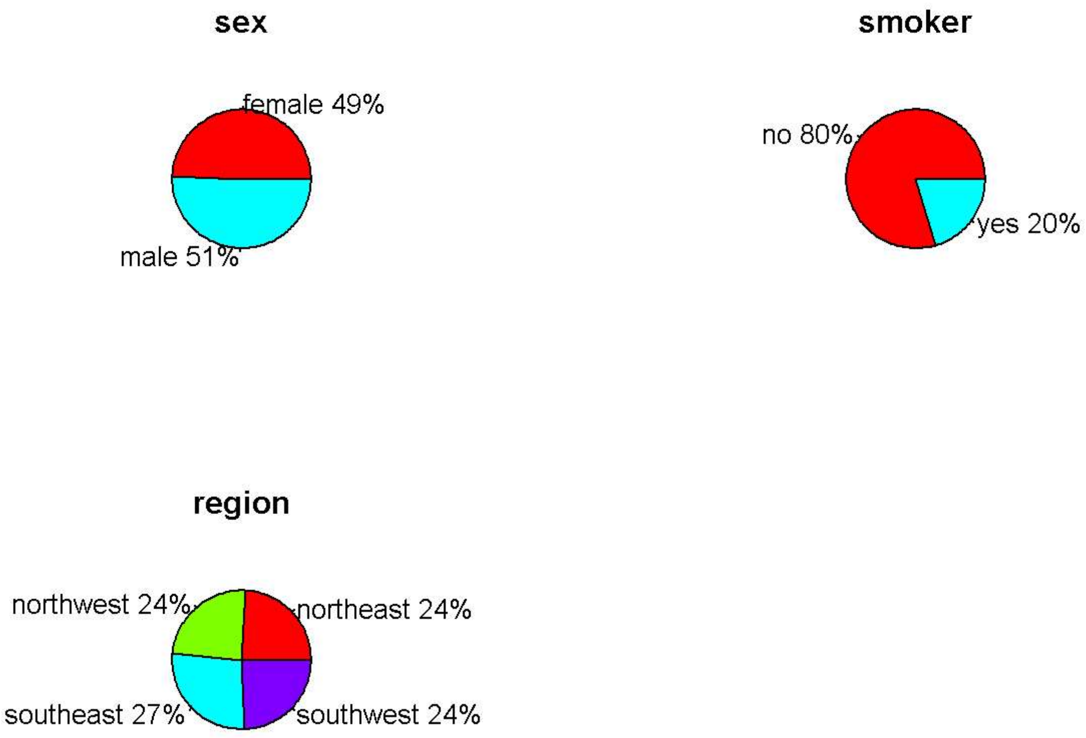
Get Table for categorical features

```
get_table <- function(data , key ){
  index_list <- get_index(data, key = key)
  data <- data  %>% mutate_if(is.factor, as.character)
  cate_num <-nrow(index_list$categorical)
  table_list <- vector(cate_num, mode = 'list')
  for ( i in 1:cate_num){
    cate_idx <-index_list$categorical$index
    table_list[[i]] <- table(data[,cate_idx[i]])

  }
  names(table_list) <- as.character(index_list$categorical$categorical_variables)
  return(table_list)
}

table_list <-get_table(train_data, key = 'key')
table_list
```

```
## $sex
##
## female   male
##    662    676
##
## $smoker
##
##   no  yes
## 1064  274
##
## $region
##
## northeast northwest southeast southwest
##       324       325       364       325
```

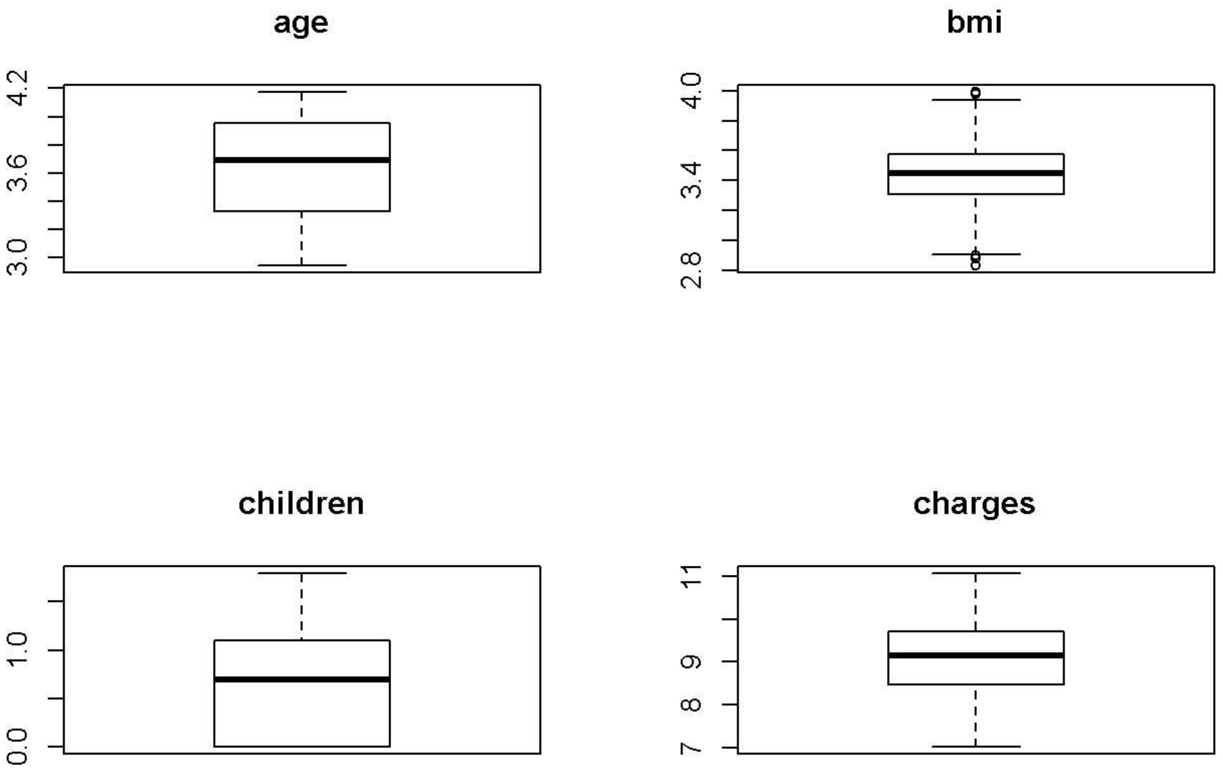
Get Pie Chart for categorical features

```
par(mfrow = c(2,2))
get_pie <- function(data, key){
  table_list <-get_table(train_data, key = key)
  n <- length(table_list)
  for ( i in 1:n){
    table_df <- table_list[[i]] %>% as.data.frame()
    ratio <- round(table_df$Freq/sum(table_df$Freq)*100)
    ratio <- paste(table_df$Var1, ratio , sep =' ')
    ratio <- paste0(ratio,'%')
    pie(table_list[[i]],labels =  ratio , col=rainbow(length(ratio)),
        main = names(table_list)[i]   )
  }
}
get_pie(train_data,'key')
```

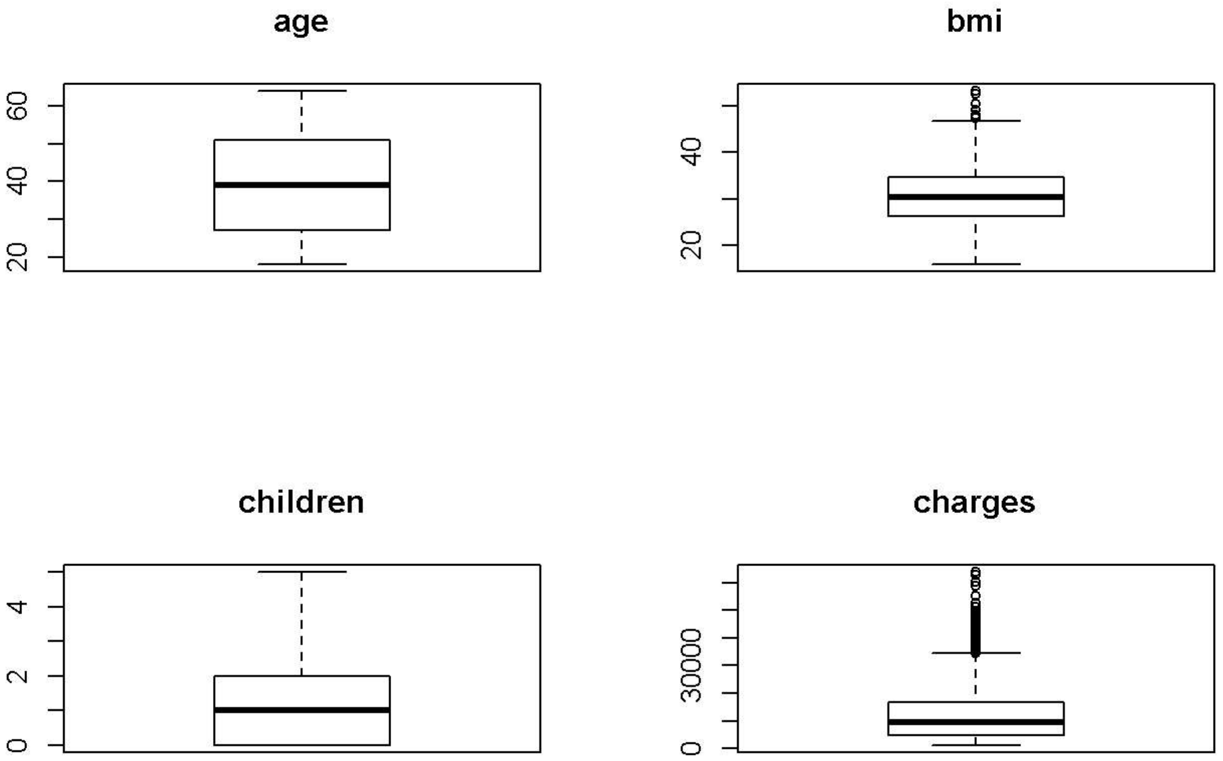


Get box plot for Numerical features

```
par(mfrow = c(2,2))
get_box <- function(data, key, log_option = T){
  data <- as.data.frame(data)
  index_list <- get_index(data, key = key)
  n <- nrow(index_list$numerical)
  if ( log_option == T ){
    for ( i in 1:n){
      n<-index_list$numerical$index[i]
      boxplot(log(data[,n]+1), main = index_list$numerical$numerical_variables[i]  )
    }
  } else {
    for ( i in 1:n){
      n<-index_list$numerical$index[i]
      boxplot(data[,n], main = index_list$numerical$numerical_variables[i])
    }
  }
}

get_box(train_data,'key')
```



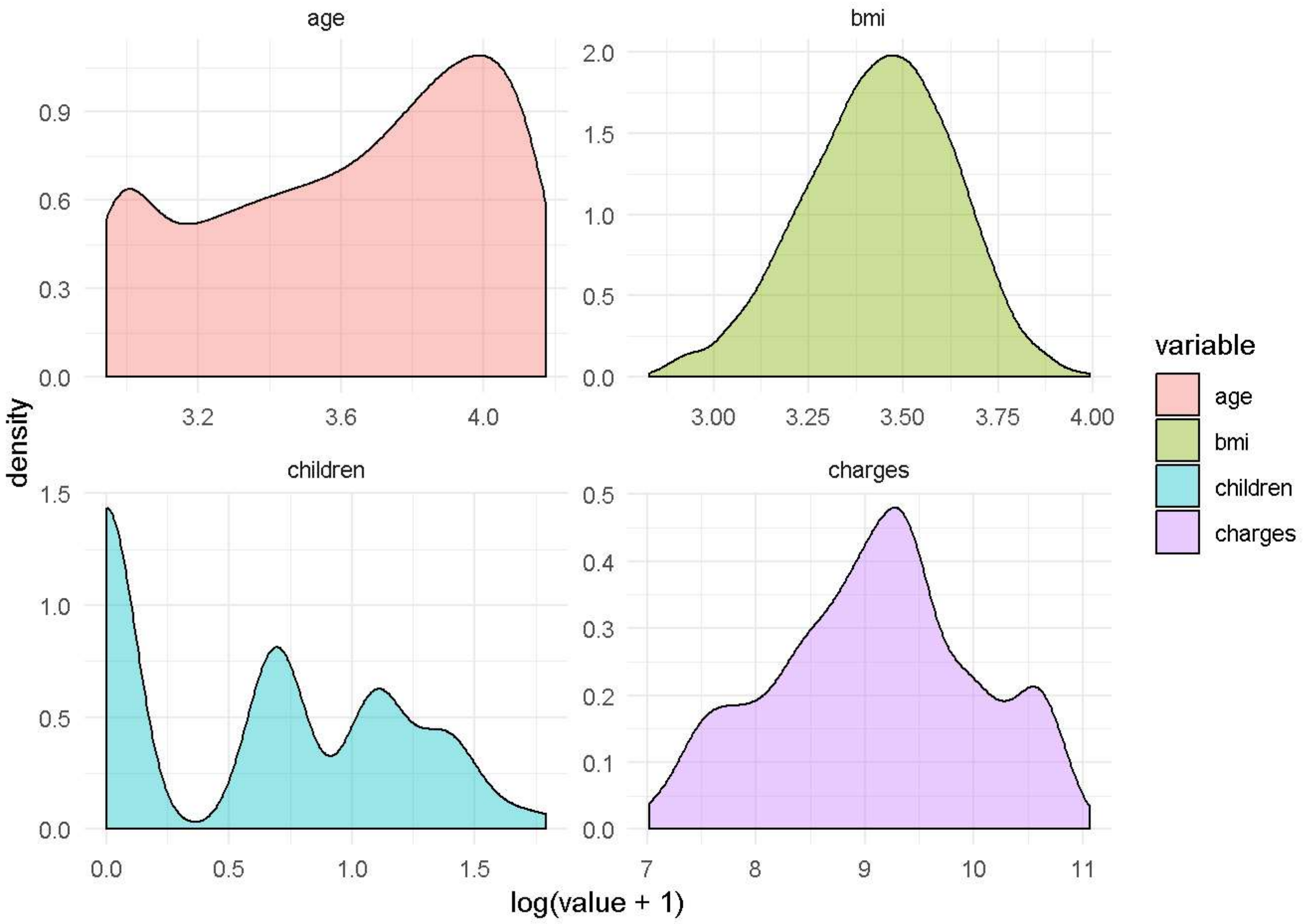
```
get_box(train_data,'key',log_option = F)
```



Get density plot for Numerical features

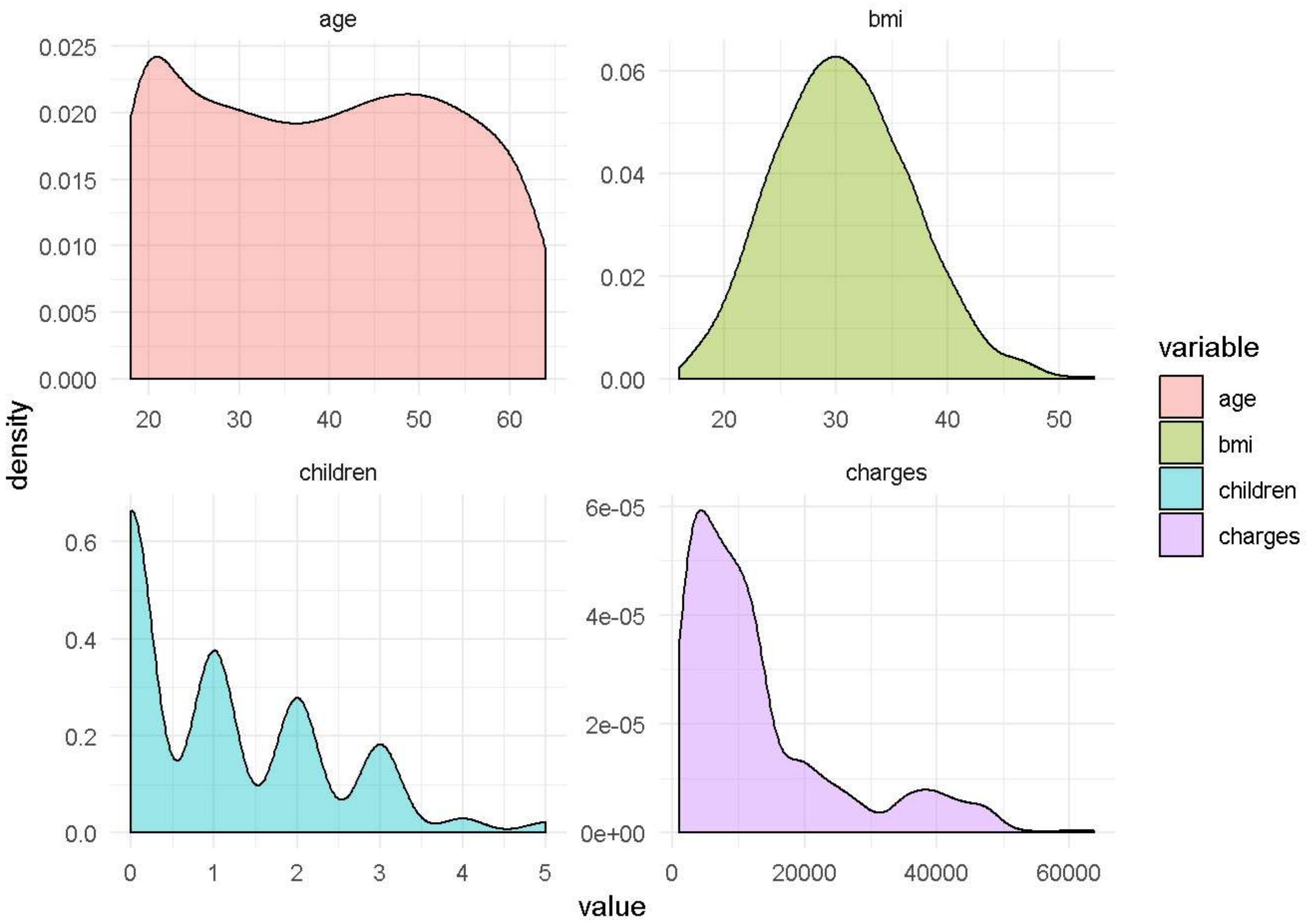
```
get_density <- function(data, key, log_option = T){
  data <- as.data.frame(data)
  index_list <- get_index(data, key = key)
  n <- nrow(index_list$numerical)
  num_data <- data[,index_list$numerical$index] %>% melt()
  if ( log_option == T ){
    result<-ggplot(num_data, aes(x=log(value+1),fill = variable)) +
      geom_density(alpha=0.4) + facet_wrap(~variable, scales = 'free') + theme_minimal()

  } else{
    result<-ggplot(num_data, aes(x=value,fill = variable)) +
      geom_density(alpha=0.4) + facet_wrap(~variable, scales = 'free') + theme_minimal()

  }
  return(result)
}
get_density(train_data, key = 'key',log_option = T)
```

```
## No id variables; using all as measure variables
```



```
get_density(train_data, key = 'key',log_option = F)
```

```
## No id variables; using all as measure variables
```


age
0.025
0.020
0.015
0.010
0.005
0.000
20
30
40
50
60
bmi
0.06
0.04
0.02
0.00
20
30
40
50
children
0.6
0.4
0.2
0.0
0
1
2
3
4
5
charges
6e-05
4e-05
2e-05
0e+00
0
20000
40000
60000
density
value
variable
age
bmi
children
charges