Heath Insurance Cost Analysis

Nanda

2022-11-01

1.Summary

Context

Health insurance coverage can be determined by multiple factors: social, economics (lifestyle, age….) In this study, based on the insurance data, we determine which factors best affect the coverage of insurance costs. Then, using the machine learning technique, we make a prediction of the health insurance costs.

2. Ask Phase

• What is the problem we are trying to solve?

The main objective is to find out the drivers(predictors) of insurance charges and finally make a prediction of these insurance charges.

• How can the insights drive business decisions?

The insights will help the company to accurately estimate its health insurance premiums.

3. Prepare Phase

• Where is our data located?

The data is located in a kaggle dataset.

• Did the data’s integrity verify? Yes

All the files have consistent columns and each column has the correct type of data.

4. Process Phase

a) Installing and loading packages

Before loading library be sure that you have already installed these packages.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6      ✓ purrr   0.3.5
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(skimr)
library(janitor)
```

```
##
## Attachement du package : 'janitor'
##
## Les objets suivants sont masqués depuis 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(ggplot2)
library(dplyr)
library(plotly)
```

```
##
## Attachement du package : 'plotly'
##
## L'objet suivant est masqué depuis 'package:ggplot2':
##
##     last_plot
##
## L'objet suivant est masqué depuis 'package:stats':
##
##     filter
##
## L'objet suivant est masqué depuis 'package:graphics':
##
##     layout
```

```
library("cowplot")
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(caTools)
```

b) Get data

Our dataset is in CSV format, so we use read.csv() function

```
my_data=read.csv("HEATH_DATA.csv")
```

c) Preview our data

```
View(my_data)
typeof(my_data)
```

```
## [1] "list"
```

```
colnames(my_data)
```

```
## [1] "age"      "sex"      "bmi"      "children" "smoker"   "region"   "charges"
```

```
str(my_data)
```

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

```
nrow(my_data)
```

```
## [1] 1338
```

```
ncol(my_data)
```

```
## [1] 7
```

d) Summary of our data

```
summary(my_data)
```

```
##       age            sex                 bmi           children
##  Min.   :18.00   Length:1338        Min.   :15.96   Min.   :0.000
##  1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000
##  Median :39.00   Mode  :character   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
##     smoker             region             charges
##  Length:1338        Length:1338        Min.   : 1122
##  Class :character   Class :character   1st Qu.: 4740
##  Mode  :character   Mode  :character   Median : 9382
##                                        Mean   :13270
##                                        3rd Qu.:16640
##                                        Max.   :63770
```

e) Is there any na variable in our dataset?

No, we can find out this information with is.na() function.

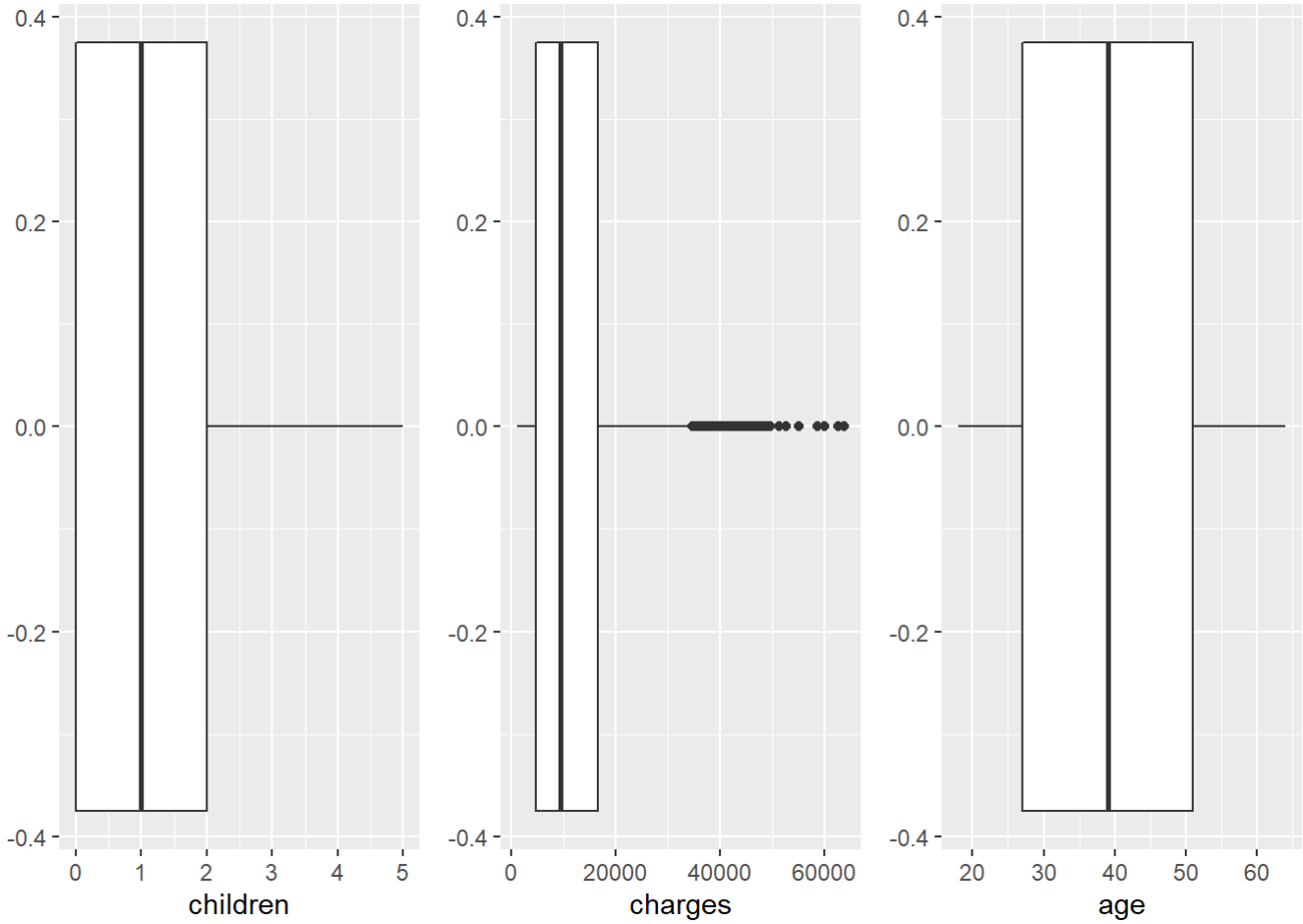
5. Analyse Phase

5.1. Univariate analysis

5.1.1. Boxplot visualization

```
P1=ggplot(my_data, aes(x=children)) +
  geom_boxplot()
P2=ggplot(my_data, aes(x=charges)) +
  geom_boxplot()
P3=ggplot(my_data, aes(x=age)) +
  geom_boxplot()
```

```
plot_grid(P1,P2,P3, labels = c(),nrow = 1, ncol = 3)
```

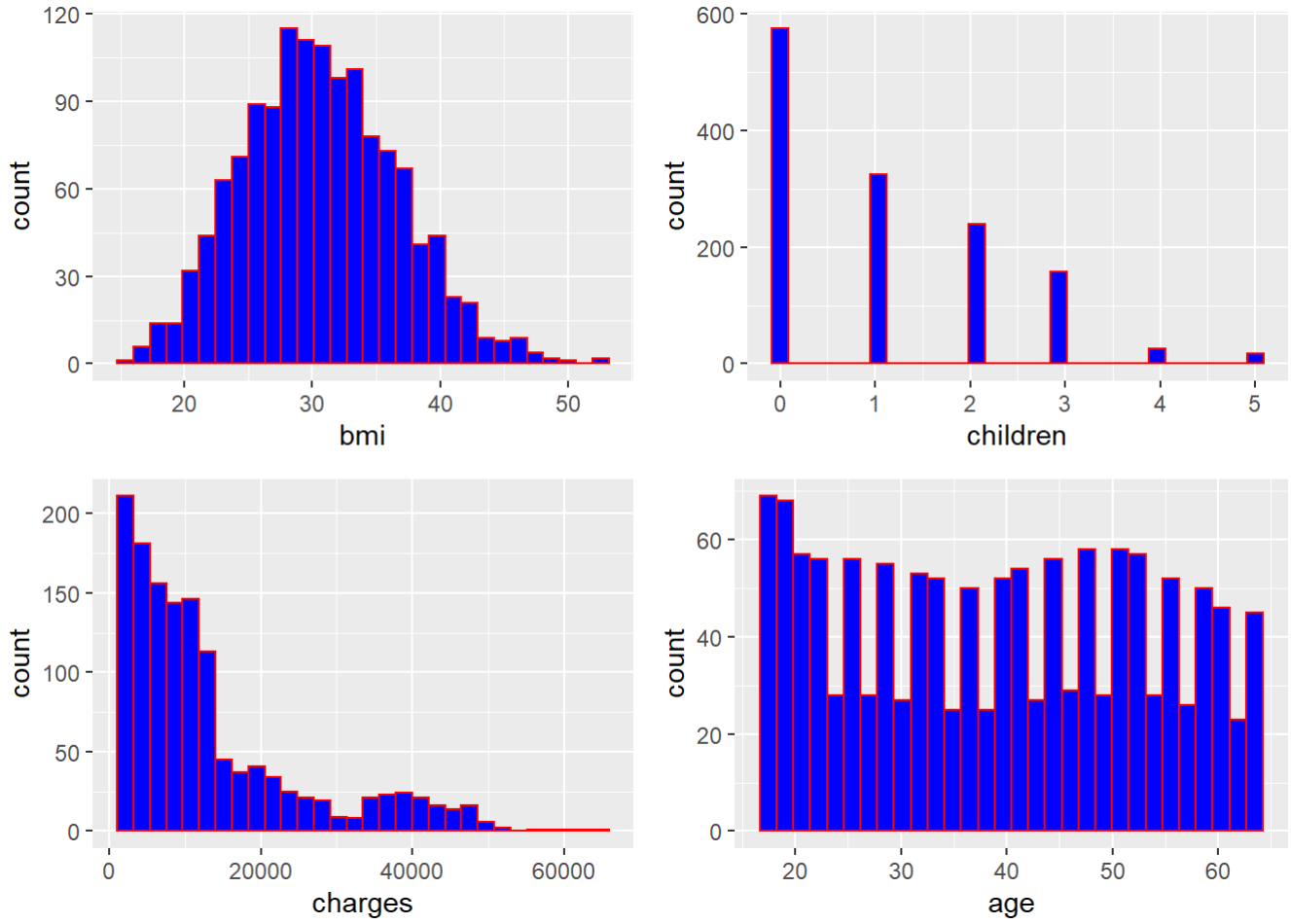


5.1.2. Bar chart visualization

```
P4=ggplot(my_data, aes(bmi))+
  geom_histogram(fill="blue", color="red")

P5=ggplot(my_data, aes(children))+
  geom_histogram(fill="blue", color="red")

P6=ggplot(my_data, aes(charges))+
  geom_histogram(fill="blue", color="red")

P7=ggplot(my_data, aes(age))+
  geom_histogram(fill="blue", color='red')
```

```
plot_grid(P4,P5,P6,P7, labels = c(),nrow = 2, ncol = 2)
```

```
## `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`.
```



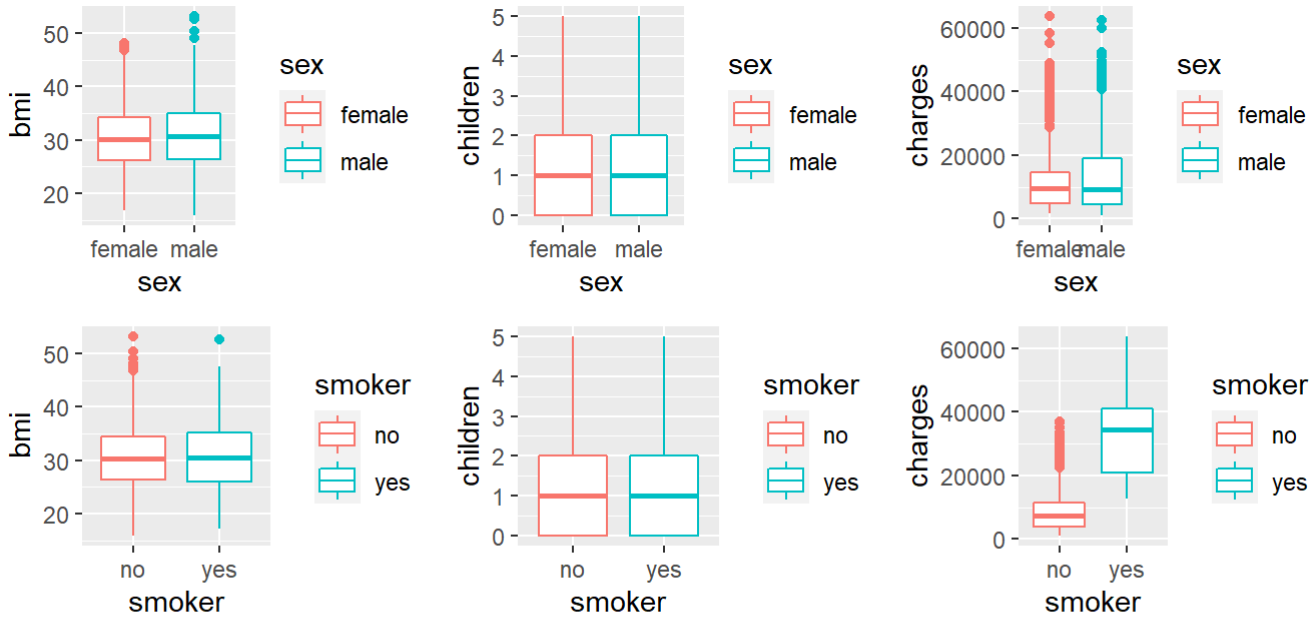
5.2. Bivariate and multivariate analysis

We want to create the boxplot for each 3 numeric variables. For best understanding the relationship between the numeric variables and the nominal variable , we build the boxplot of numeric variables by the level of each nominal variable.

5.2.1. bmi boxplot by the sex and smoker variable

```
P8=ggplot(my_data, aes(x=sex, y=bmi, color=sex)) +
  geom_boxplot()


P9=ggplot(my_data, aes(x=sex, y=children, color=sex)) +
  geom_boxplot()


P10=ggplot(my_data, aes(x=sex, y=charges, color=sex)) +
  geom_boxplot()

P11=ggplot(my_data, aes(x=smoker, y=bmi, color=smoker)) +
  geom_boxplot()

P12=ggplot(my_data, aes(x=smoker, y=children, color=smoker)) +
  geom_boxplot()

P13=ggplot(my_data, aes(x=smoker, y=charges, color=smoker)) +
  geom_boxplot()
```

```
plot_grid(P8,P9,P10,P11,P12,P13, labels = c(),nrow = 3, ncol = 3)
```



We can observe that the nominal variable “smoker” has a strong effect on the variable “charges”.

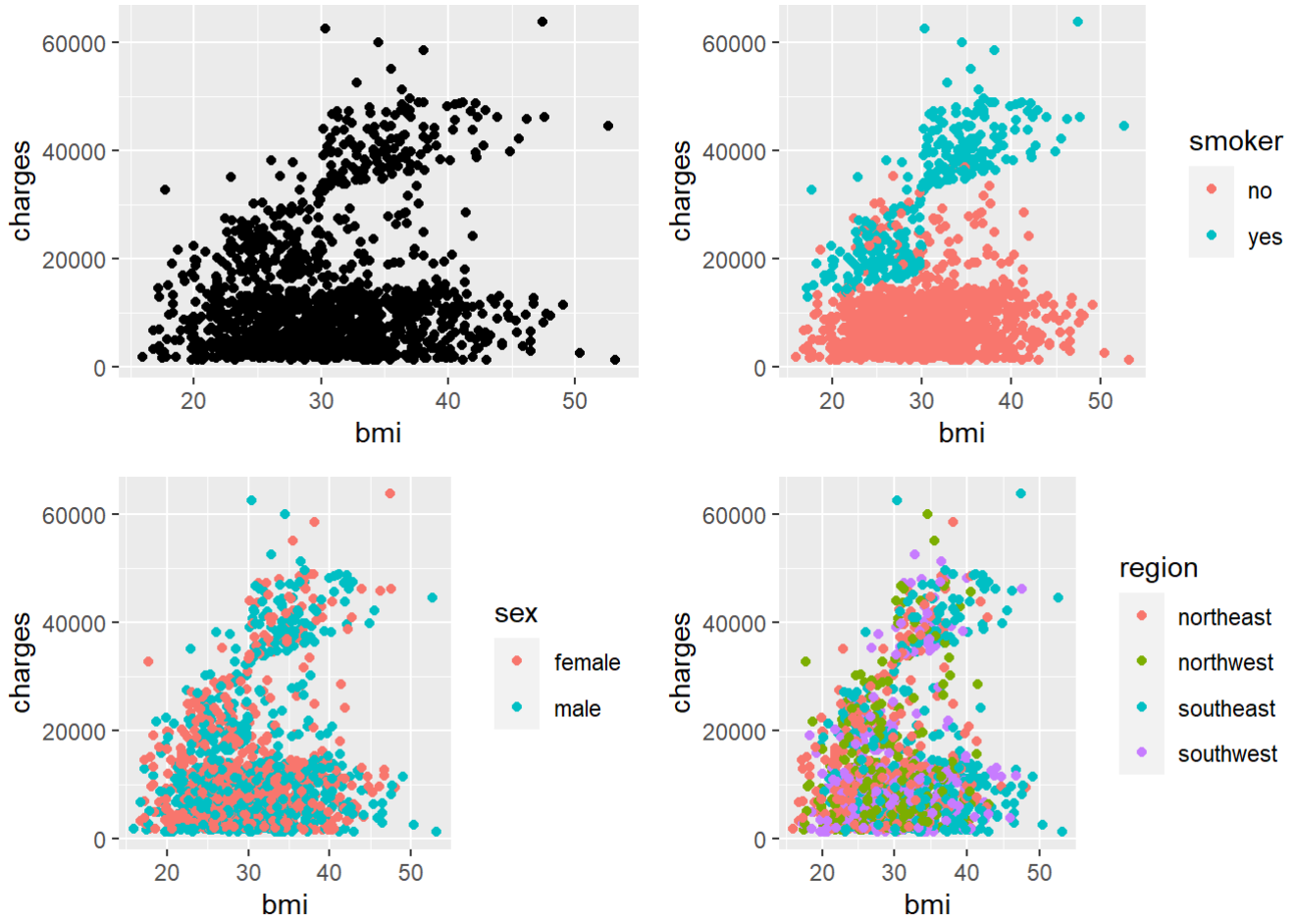
5.2.2. Relationship between bmi and charges

a) Scatter plot

Using scatter plot we are going to find out the relationship between charges and bmi

```
P14=ggplot(my_data)+
  geom_point(mapping = aes(x=bmi, y=charges))

P15=ggplot(my_data)+
  geom_point(mapping = aes(x=bmi, y=charges, color=smoker))

P16=ggplot(my_data)+
  geom_point(mapping = aes(x=bmi, y=charges, color=sex))

P17=ggplot(my_data)+
  geom_point(mapping = aes(x=bmi, y=charges, color=region))
```

```
plot_grid(P14,P15,P16,P17, labels = c(),nrow = 2, ncol = 2)
```



b) Trend line

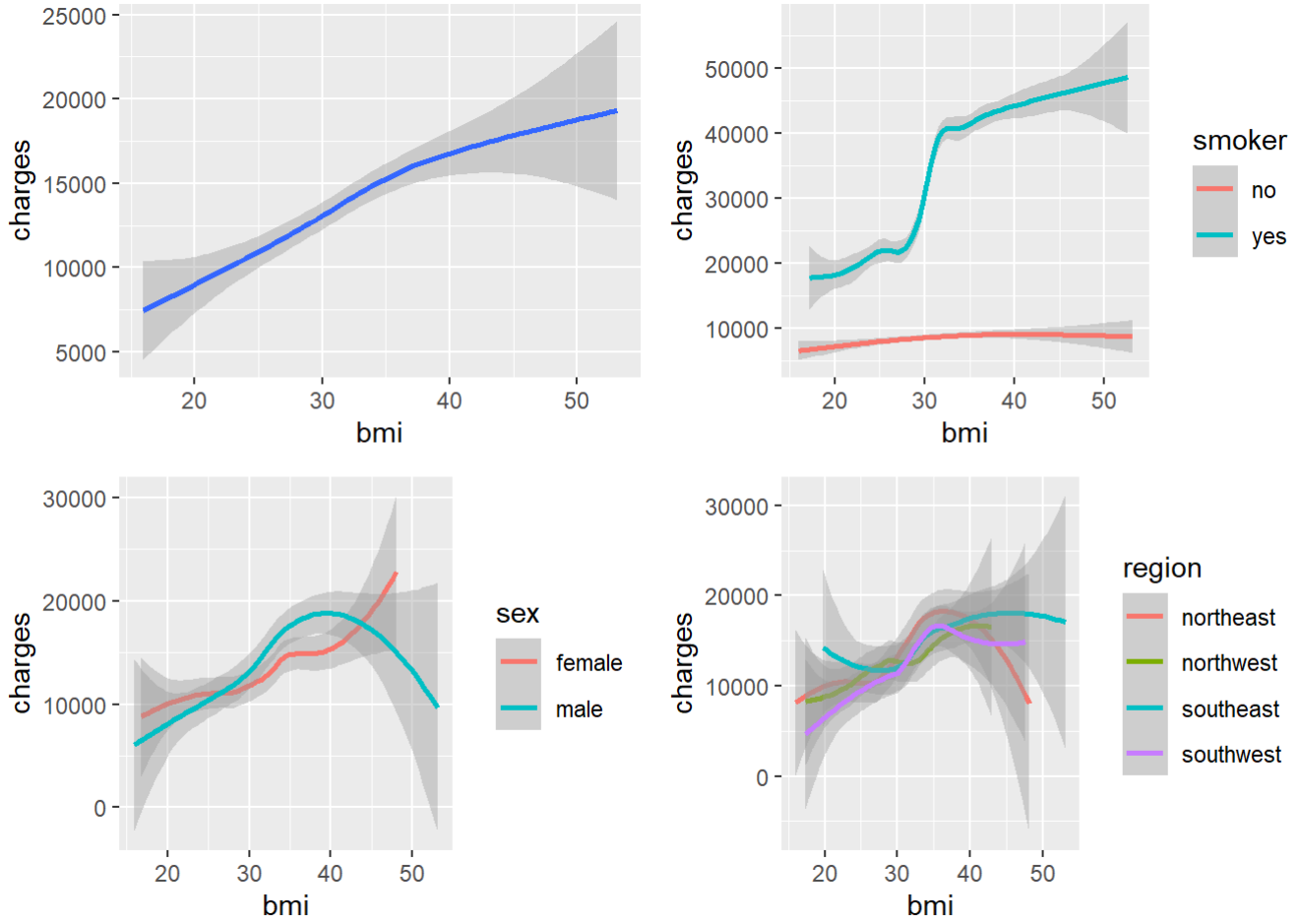
In the graphs below, we use a trend line to approximate a general shape of the previous scatter plot.

```
P18=ggplot(my_data)+
  geom_smooth(mapping = aes(x=bmi, y=charges))

P19=ggplot(my_data)+
  geom_smooth(mapping = aes(x=bmi, y=charges, color=smoker))


P20=ggplot(my_data)+
  geom_smooth(mapping = aes(x=bmi, y=charges, color=sex))

P21=ggplot(my_data)+
  geom_smooth(mapping = aes(x=bmi, y=charges, color=region))
```

```
plot_grid(P18,P19,P20,P21, labels = c(),nrow = 2, ncol = 2)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



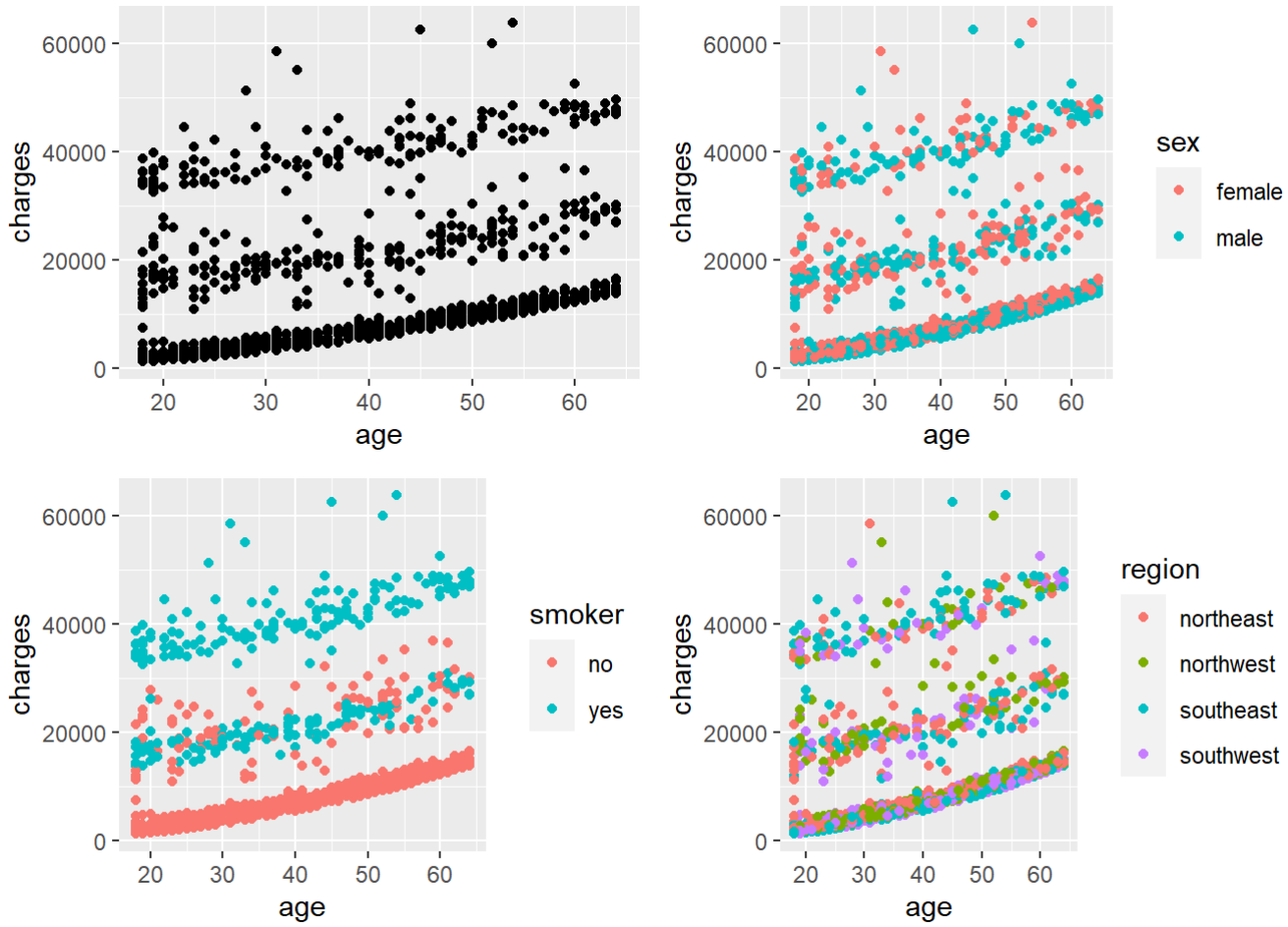
We can also observe that the variable “bmi” seems to be positively correlated to the variable “charges”.

5.2.3. Relationship between charges and age character

a) Scatter plot

```
P22=ggplot(my_data)+
  geom_point(mapping = aes(x=age, y=charges))

P23=ggplot(my_data)+
  geom_point(mapping = aes(x=age, y=charges, color=sex))

P24=ggplot(my_data)+
  geom_point(mapping = aes(x=age, y=charges, color=smoker))

P25=ggplot(my_data)+
  geom_point(mapping = aes(x=age, y=charges, color=region))
```

```
plot_grid(P22,P23,P24,P25, labels = c(),nrow = 2, ncol = 2)
```



As with “bmi” variable, we can also observe that the variable “age” seems to be positively correlated to the variable “charges”.

b) Trend line

In the graphs below, we use a trend line to approximate a general shape of the previous scatter plot.

```
P26=ggplot(my_data)+
  geom_smooth(mapping = aes(x=age, y=charges))

P27=ggplot(my_data)+
  geom_smooth(mapping = aes(x=age, y=charges, color=sex))

P28=ggplot(my_data)+
  geom_smooth(mapping = aes(x=age, y=charges, color=smoker))

P29=ggplot(my_data)+
  geom_smooth(mapping = aes(x=age, y=charges, color=region))
```

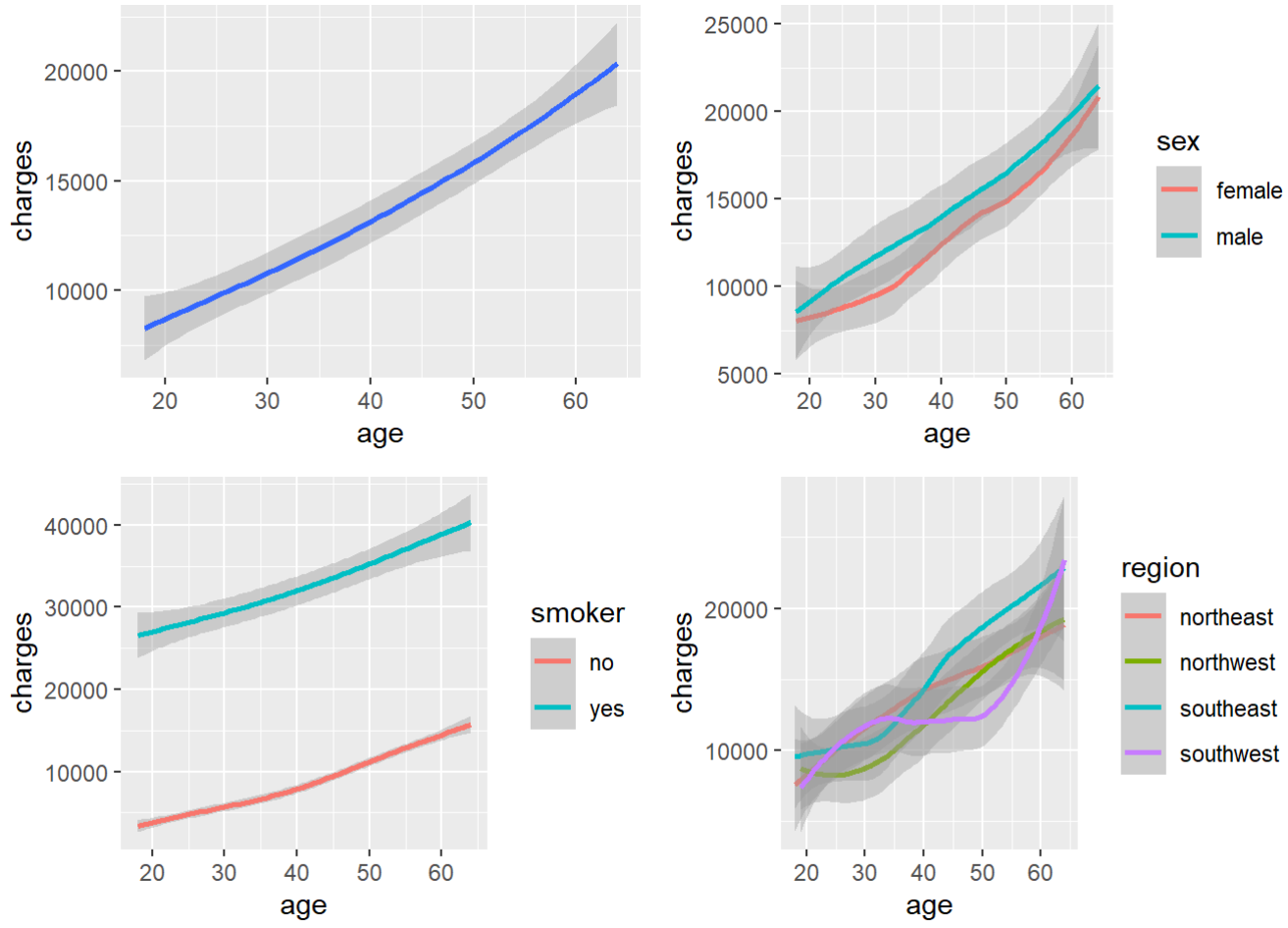
```
plot_grid(P26,P27,P28,P29, labels = c(),nrow = 2, ncol = 2)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

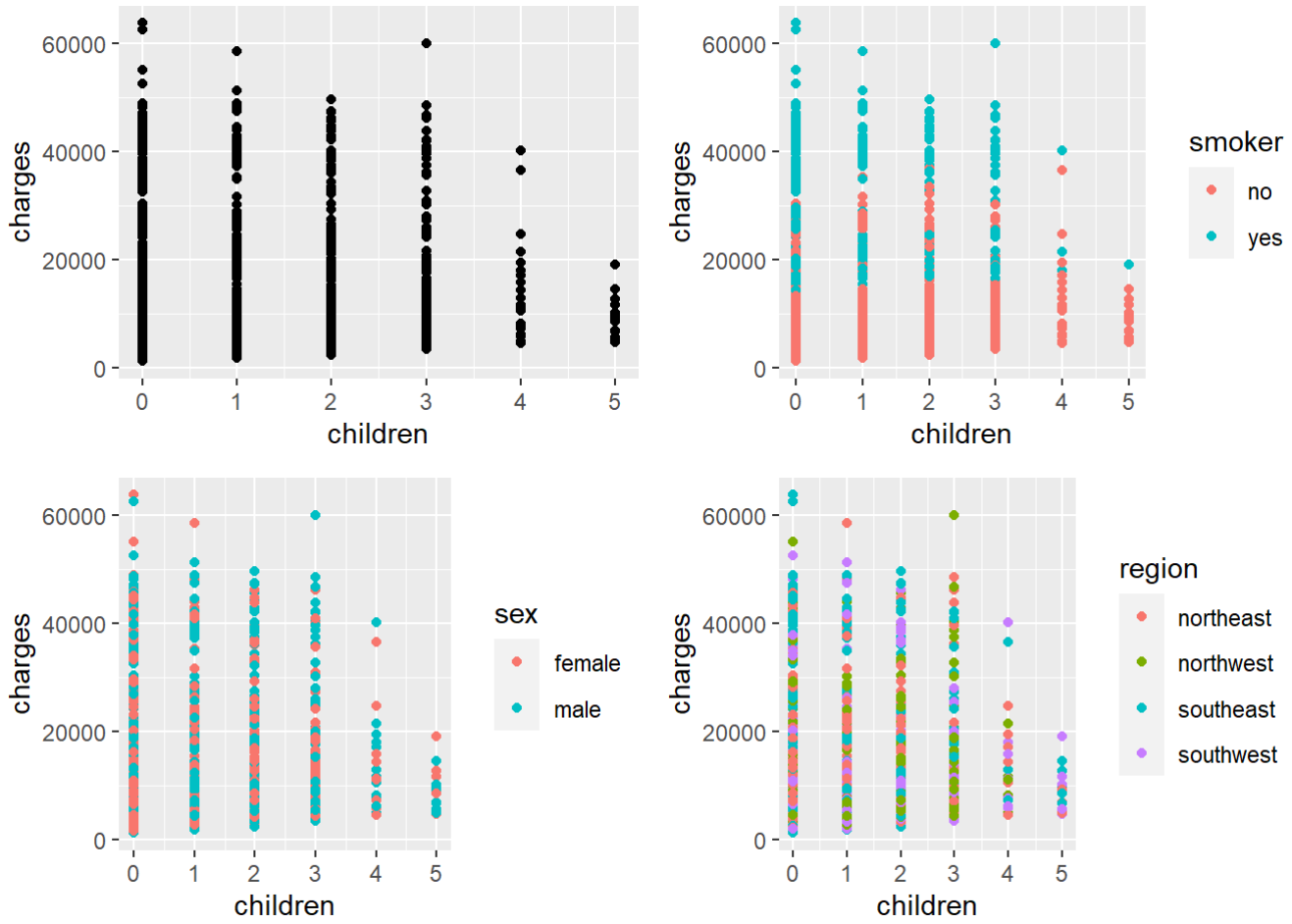


5.2.4. Relationship between charges and children

a) Scatter plot

```
P30=ggplot(my_data)+
  geom_point(mapping = aes(x=children, y=charges))

P31=ggplot(my_data)+
  geom_point(mapping = aes(x=children, y=charges, color=smoker))

P32=ggplot(my_data)+
  geom_point(mapping = aes(x=children, y=charges, color=sex))

P33=ggplot(my_data)+
  geom_point(mapping = aes(x=children, y=charges, color=region))
```

```
plot_grid(P30,P31,P32,P33, labels = c(),nrow = 2, ncol = 2)
```

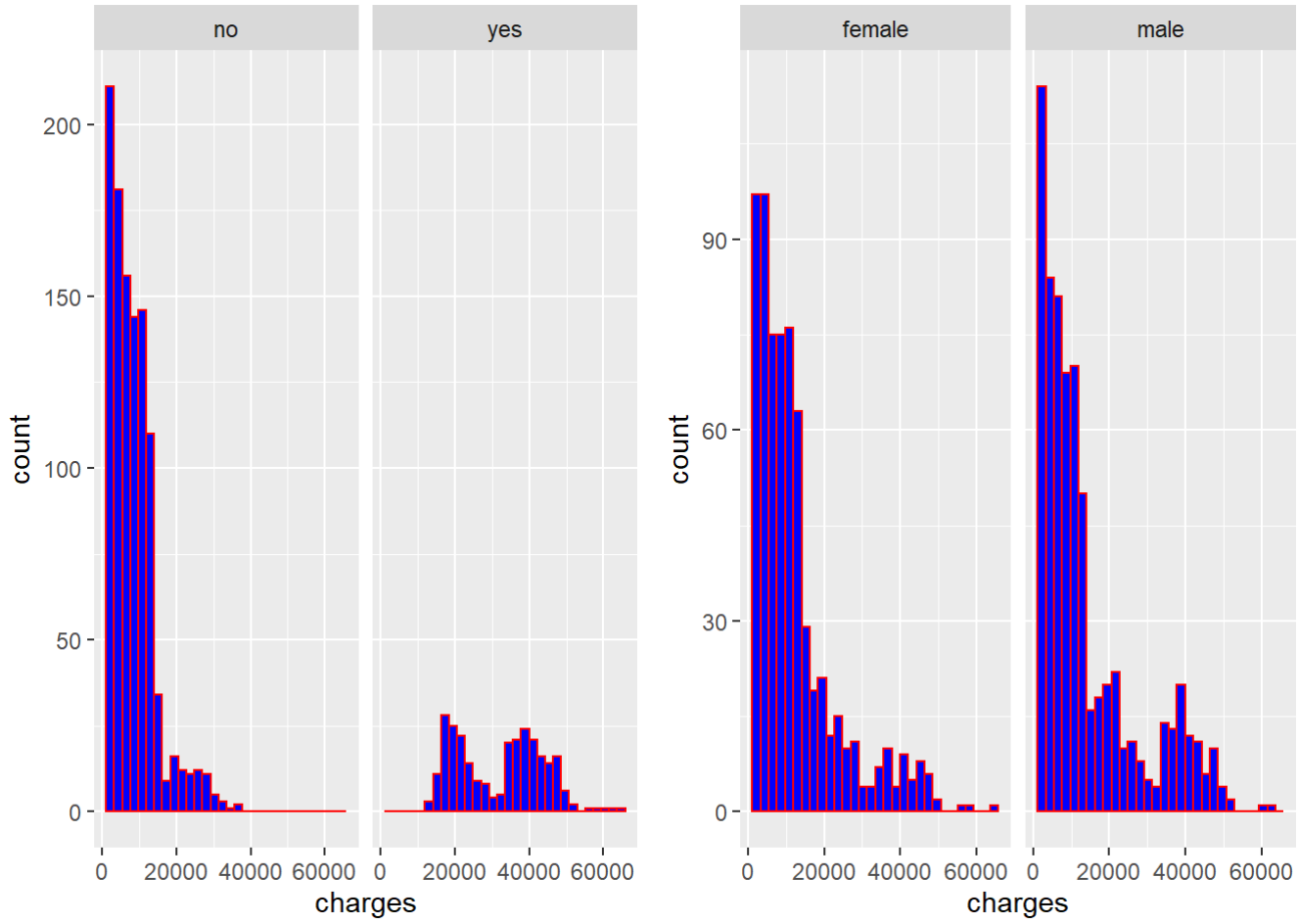


b) Bar chart

```
P34=ggplot(my_data, aes(charges))+
  geom_histogram(fill="blue", color="red")+
  facet_grid(~smoker)

P35=ggplot(my_data, aes(charges))+
  geom_histogram(fill="blue", color="red")+
  facet_grid(~sex)
```

```
plot_grid(P34,P35, labels = c(),nrow = 1, ncol = 2)
```

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



5.2.5. Create a heatmap

In order to better observe the relationship between the response variable(charges) and the numerical predictors(age,bmi,children), we will build the correlation matrix. For that, we need to first select a new dataset containing only numerical variables.

a) Select a new dataset

```
data_num=my_data %>%
  select(age, bmi, children, charges)
View(data_num)
```

b) Correlation matrix

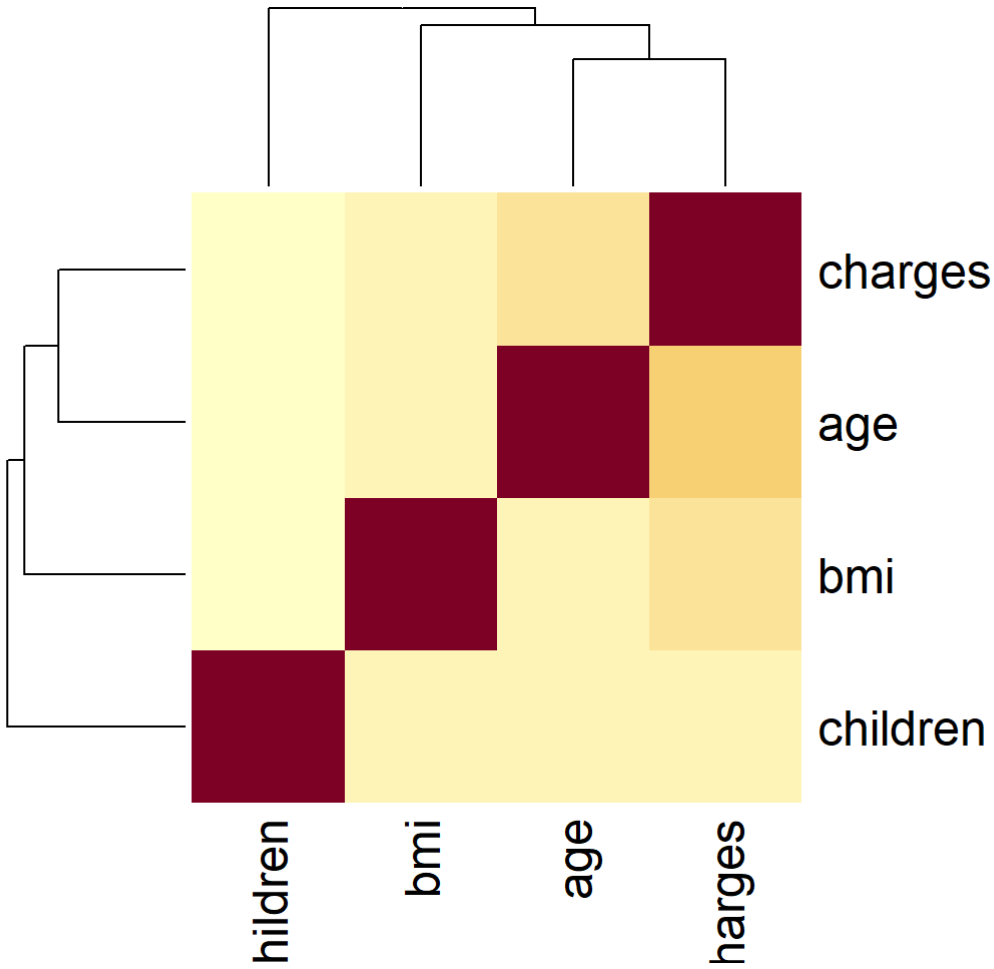
from a new dataset we build a correlation matrix

```
mat_cor=cor(data_num)
View(mat_cor)
mat_cor
```

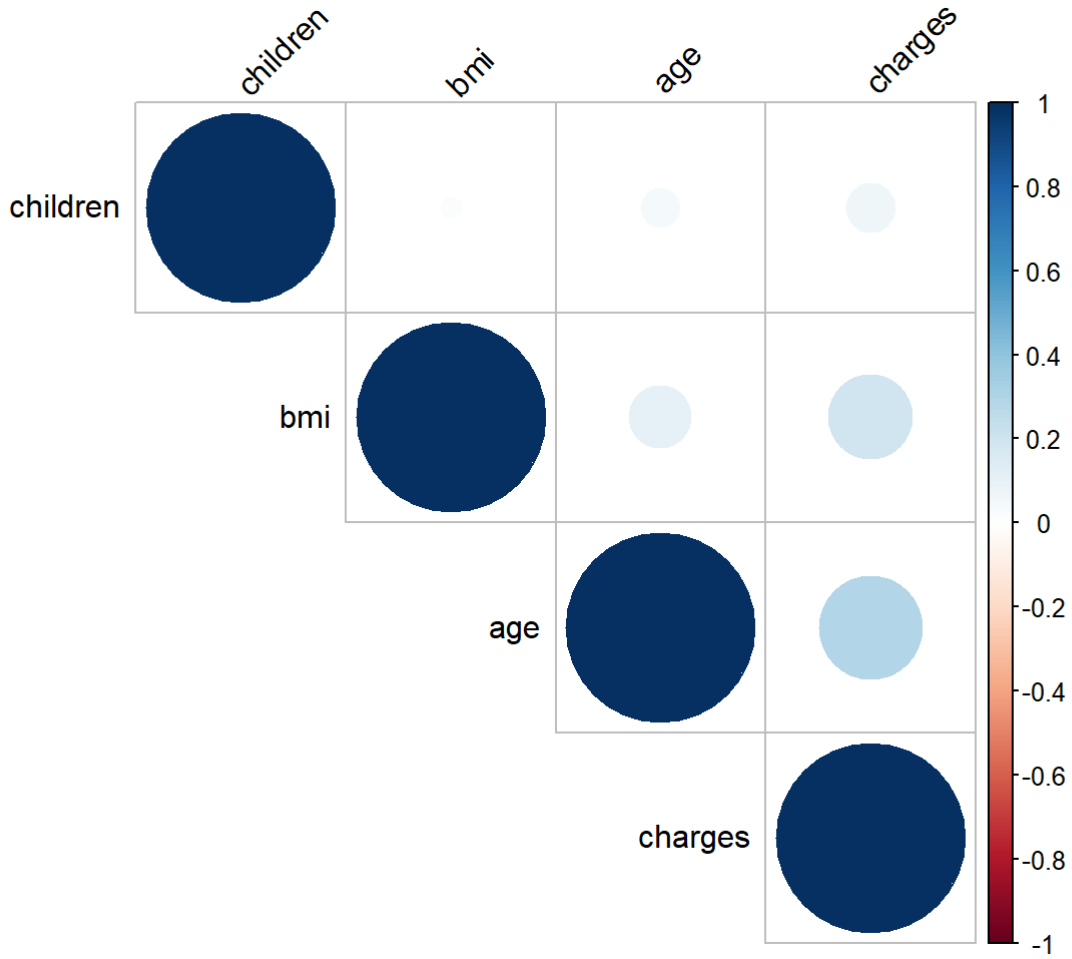
```
##                age       bmi   children    charges
## age      1.0000000 0.1092719 0.04246900 0.29900819
## bmi      0.1092719 1.0000000 0.01275890 0.19834097
## children 0.0424690 0.0127589 1.00000000 0.06799823
## charges  0.2990082 0.1983410 0.06799823 1.00000000
```

c) Heatmap and corrplot

```
heat_map=heatmap(mat_cor)
```




```
corr_plot=corrplot(mat_cor,type="upper", order="hclust", tl.col="black", tl.srt=45)
```



5.3. Regression

5.3.1. First model with all predictors

```
M_1=lm(charges~age+bmi+sex+children+smoker+region, my_data)
```

```
summary(M_1)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi + sex + children + smoker +
##     region, data = my_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11304.9  -2848.1   -982.1   1393.9  29992.8
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -11938.5      987.8 -12.086  < 2e-16 ***
## age                256.9       11.9  21.587  < 2e-16 ***
## bmi                339.2       28.6  11.860  < 2e-16 ***
## sexmale           -131.3      332.9  -0.394 0.693348
## children           475.5      137.8   3.451 0.000577 ***
## smokeryes        23848.5      413.1  57.723  < 2e-16 ***
## regionnorthwest   -353.0      476.3  -0.741 0.458769
## regionsoutheast  -1035.0      478.7  -2.162 0.030782 *
## regionsouthwest   -960.0      477.9  -2.009 0.044765 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6062 on 1329 degrees of freedom
## Multiple R-squared:  0.7509, Adjusted R-squared:  0.7494
## F-statistic: 500.8 on 8 and 1329 DF,  p-value: < 2.2e-16
```

5.3.2. Second model

Based on the result of the first model, we build the second one by removing all non significant variable(region and sex)

```
M_2=lm(charges~age+bmi+children+smoker, my_data)
```

```
summary(M_2)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi + children + smoker, data = my_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11897.9  -2920.8   -986.6   1392.2  29509.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12102.77     941.98 -12.848  < 2e-16 ***
## age            257.85      11.90  21.675  < 2e-16 ***
## bmi            321.85      27.38  11.756  < 2e-16 ***
## children       473.50     137.79   3.436 0.000608 ***
## smokeryes    23811.40     411.22  57.904  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6068 on 1333 degrees of freedom
## Multiple R-squared:  0.7497, Adjusted R-squared:  0.7489
## F-statistic: 998.1 on 4 and 1333 DF,  p-value: < 2.2e-16
```

5.4. Machine Learning

In this part, we are going to perform machine learning approach to predict health insurance costs. First we need to split our dataset in two parts: training set and test set.

```
set.seed(25)
n = nrow(my_data)
n
```

```
## [1] 1338
```

```
my_split = sample(c(TRUE, FALSE), n, replace=TRUE, prob=c(0.8, 0.2))


train_data = my_data[my_split, ]
test_data = my_data[!my_split, ]


nrow(train_data)
```

```
## [1] 1058
```

```
nrow(test_data)
```

```
## [1] 280
```

5.4.1. First model

a) Process with the model training

```
m_T1=lm(charges ~ age+bmi+sex+children+smoker+region, data = train_data)
#### OR
m_T1=lm(charges~., data = train_data)
## to train with all the variable of the model
```

```
summary(m_T1)
```

```
##
## Call:
## lm(formula = charges ~ ., data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11229.8  -2909.0   -909.8   1671.8  29660.0
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -12713.44    1104.84 -11.507  < 2e-16 ***
## age                267.88      13.14  20.385  < 2e-16 ***
## sexmale           -444.35     370.71  -1.199  0.23094
## bmi                352.54      32.02  11.010  < 2e-16 ***
## children           491.94     151.26   3.252  0.00118 **
## smokeryes        24126.01     455.64  52.950  < 2e-16 ***
## regionnorthwest    -91.59     530.47  -0.173  0.86295
## regionsoutheast   -792.76     535.61  -1.480  0.13915
## regionsouthwest  -1223.78     531.97  -2.300  0.02162 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5997 on 1049 degrees of freedom
## Multiple R-squared:  0.7668, Adjusted R-squared:  0.765
## F-statistic: 431.1 on 8 and 1049 DF,  p-value: < 2.2e-16
```

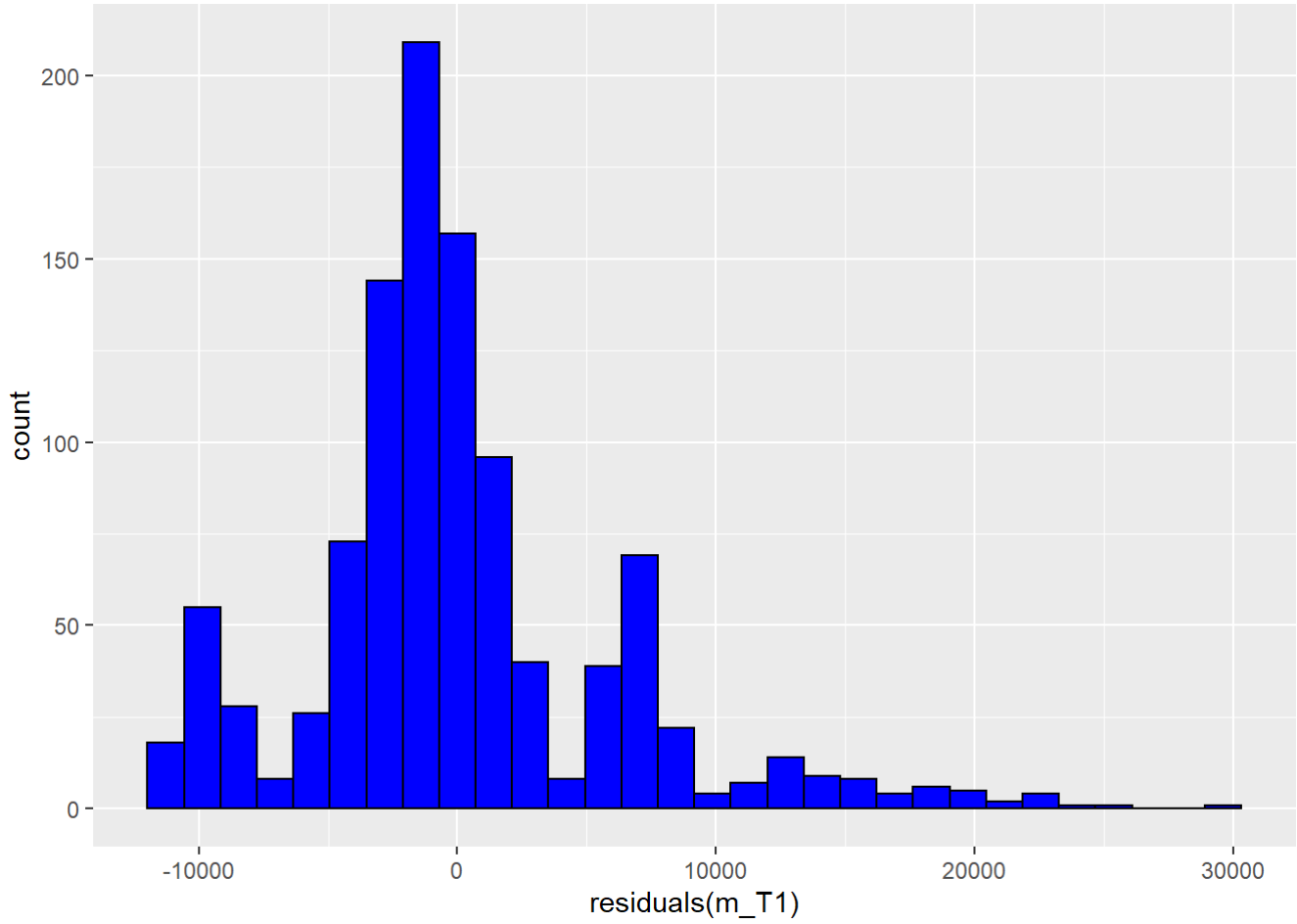
b) Residual plot or residual histogram

We are expect to see something approximatly normally

```
residual_model=as.data.frame(residuals(m_T1))
```

```
ggplot(residual_model, aes(residuals(m_T1)))+
  geom_histogram(fill="blue", color='black')
```

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



5.4.2. Second model

a) Process with the model training

Base on the result of the first model, we build the second one by removing all non significant variable(region and sex)

```
m_T2=lm(charges ~ age+bmi+children+smoker, data = train_data)
```

```
summary(m_T2)
```

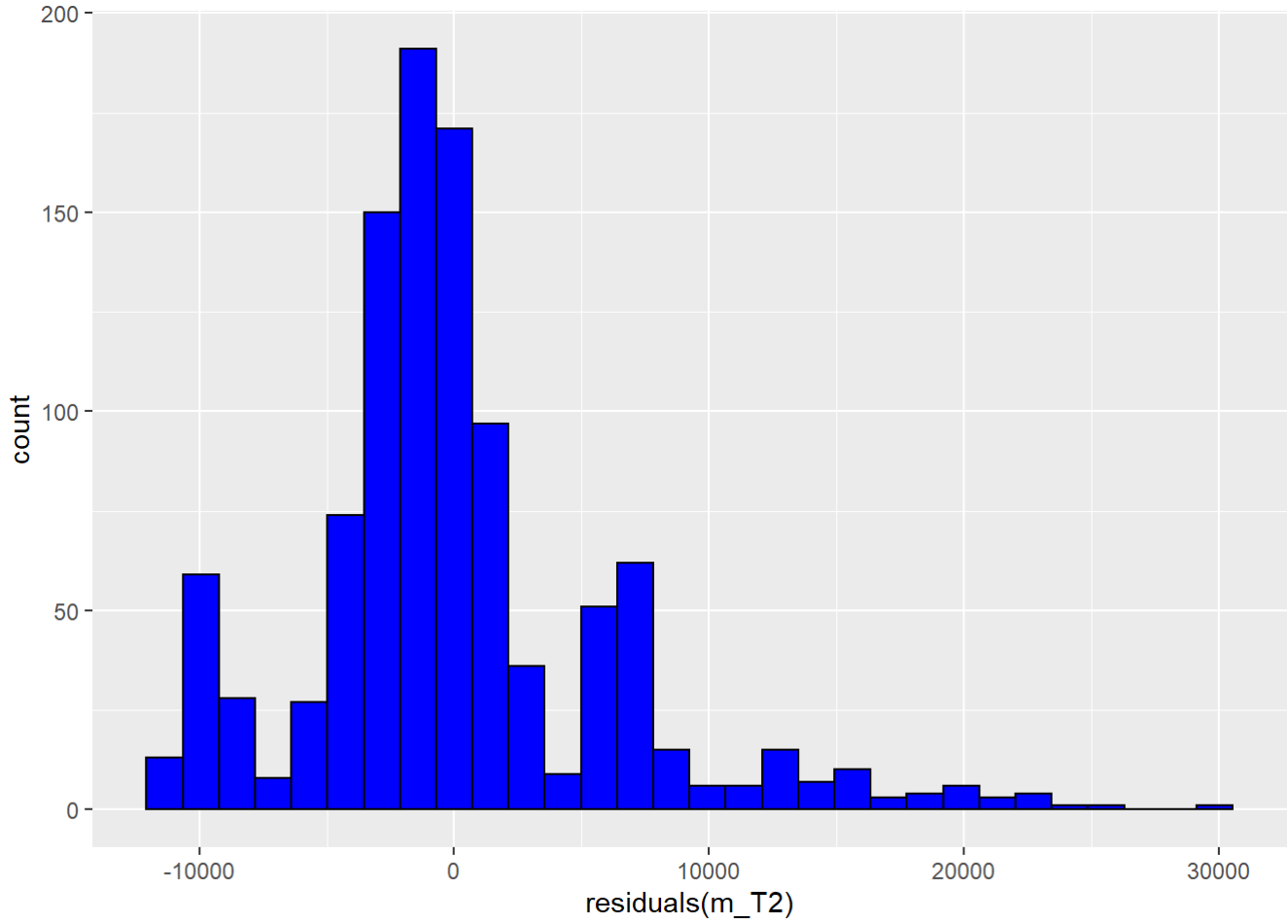
```
##
## Call:
## lm(formula = charges ~ age + bmi + children + smoker, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11978.6  -2945.0   -865.3   1588.7  29193.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12977.48    1054.26 -12.310  < 2e-16 ***
## age            268.22      13.16  20.378  < 2e-16 ***
## bmi            336.55      30.76  10.942  < 2e-16 ***
## children       479.89     151.48   3.168  0.00158 **
## smokeryes    24088.92     452.85  53.194  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6010 on 1053 degrees of freedom
## Multiple R-squared:  0.7648, Adjusted R-squared:  0.7639
## F-statistic: 856.2 on 4 and 1053 DF,  p-value: < 2.2e-16
```

b) Residual plot or residual histogram

```
residual_model=as.data.frame(residuals(m_T2))
```

```
ggplot(residual_model, aes(residuals(m_T2)))+
  geom_histogram(fill="blue", color='black')
```

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



5.4.3. Predicted value of charges

```
predict_=predict(m_T2,test_data)
```

```
model_Predict_Actual=cbind(test_data$charges,predict_)
```

```
colnames(model_Predict_Actual)=c("Actual values", "Predicted values")
```

```
model_Predict_Actual=as.data.frame(model_Predict_Actual)
```

```
is.data.frame(model_Predict_Actual)
```

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

```
View(model_Predict_Actual)
```

5.4.4. Using MSE and RMSE metrics to evaluate our model

```
MSE=mean(model_Predict_Actual$`Actual values`- model_Predict_Actual$`Predicted value`)^2

View(MSE)
MSE
```

```
## [1] 49216.32
```

```
RMSE=sqrt(MSE)
View(RMSE)
RMSE
```

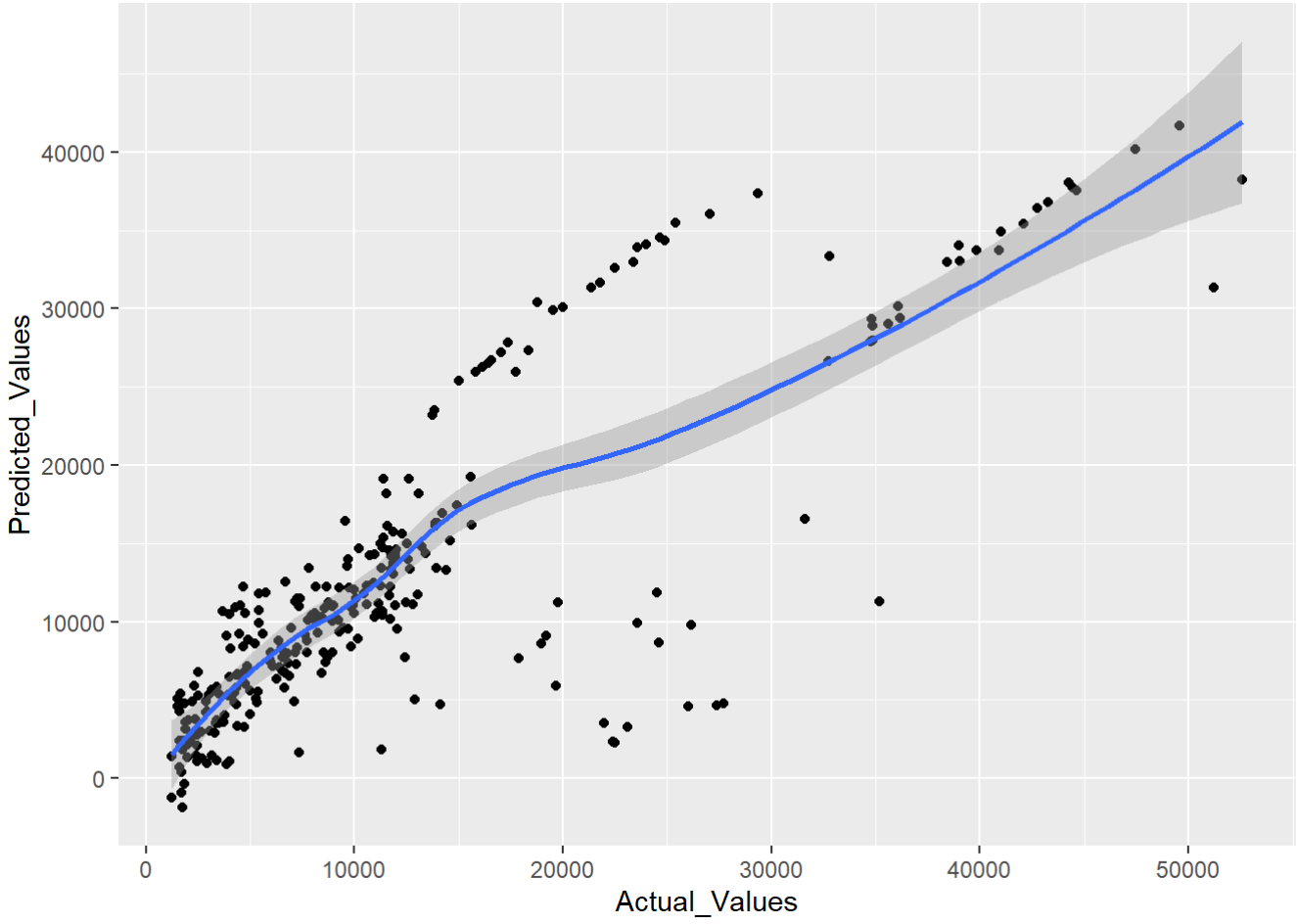
```
## [1] 221.8475
```

RMSE=233.7015 means that we are on average wrong by 221.8475 unit of charges.

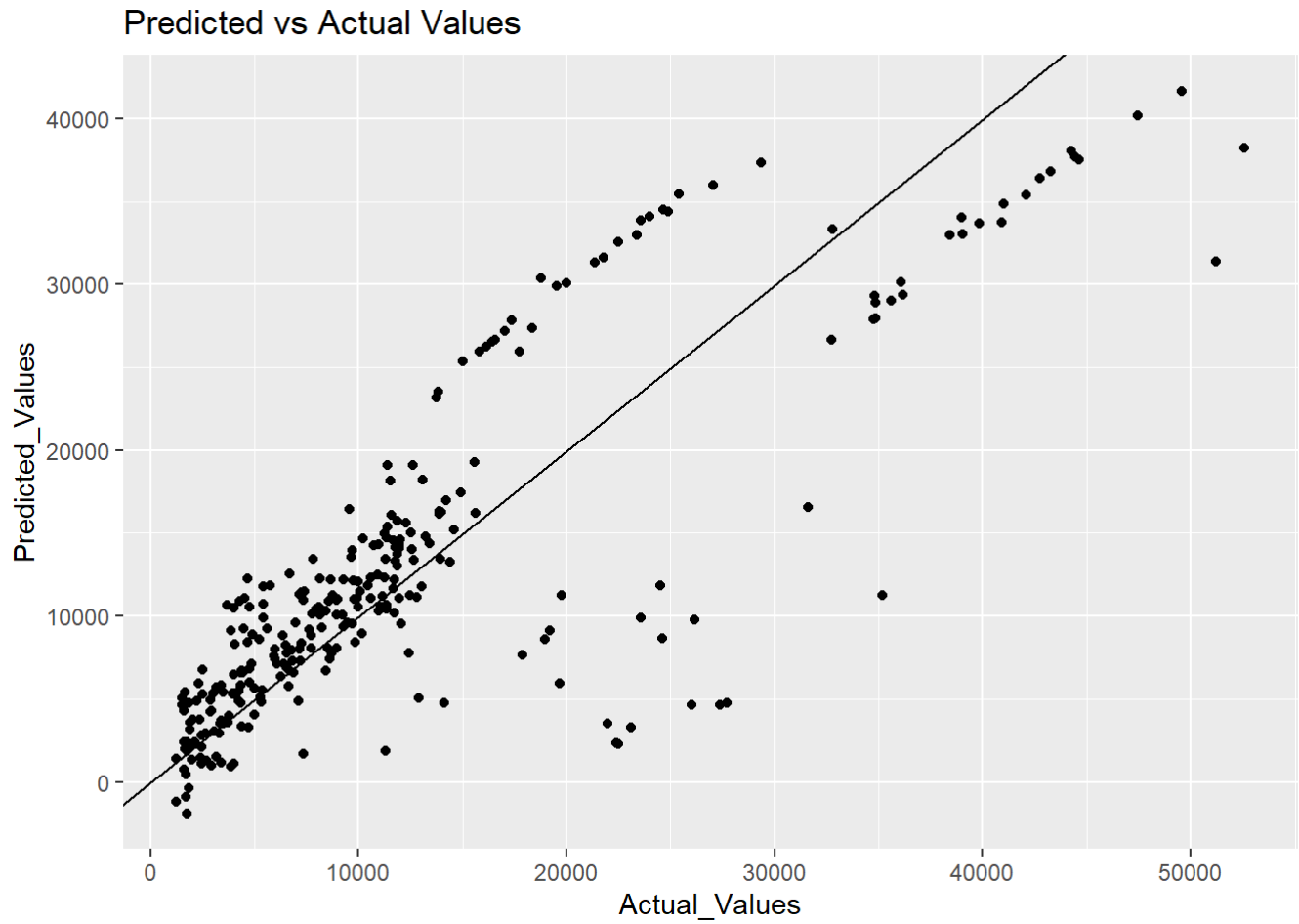
5.4.5. Actual Value vs Predicted Value: Scatter plot

```
colnames(model_Predict_Actual)=c("Actual_Values","Predicted_Values")
ggplot(data=model_Predict_Actual)+
  geom_point(mapping = aes(x=Actual_Values, y=Predicted_Values))+
  geom_smooth(mapping = aes(x=Actual_Values, y=Predicted_Values))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(data=model_Predict_Actual)+
  geom_point(mapping = aes(x=Actual_Values, y=Predicted_Values))+
  geom_abline(slope = 1, intercept = 0)+
  labs(x="Actual_Values", y="Predicted_Values", title = "Predicted vs Actual Values")
```



We can observe that for the small values of insurance cost, the model predict very well. That is because below 10000(charges), we have enough data for the good quality of the forecast.

```
mat_cor_2=cor(model_Predict_Actual)
mat_cor_2
```

```
##                 Actual_Values Predicted_Values
## Actual_Values       1.0000000        0.8254713
## Predicted_Values    0.8254713        1.0000000
```