Medical Insurance Cost Prediction

Introduction of Data

The cost of health has significantly increased in the past 10 years, due to the rising costs of healthcare service. There are many factors that contribute to determining healthcare costs, including age, sex, body mass index (BMI), number of children, smoking and region. Let's examine how these factors impact the healthcare costs for a sample of the population.

The dataset consists of 7 columns, 6 of which are features and 1 is the target column (Charges).

Features in dataset:

1. age: Age of the Primary beneficiary.
2. sex: Insurance contractor's gender.
3. bmi: Insurance contractor's bmi(body mass index).
4. children: The number of the children(dependents) covered by health insurance.
5. smoker: Is the insurance contractor's smoking or not.
6. region: Region shows residential area in United States (US).
7. charges: The total charges for medical expenses incurred, as billed by the health insurance.

These features can be used to predict healthcare costs and understand the factors that contribute to them.

The associated insurance problem

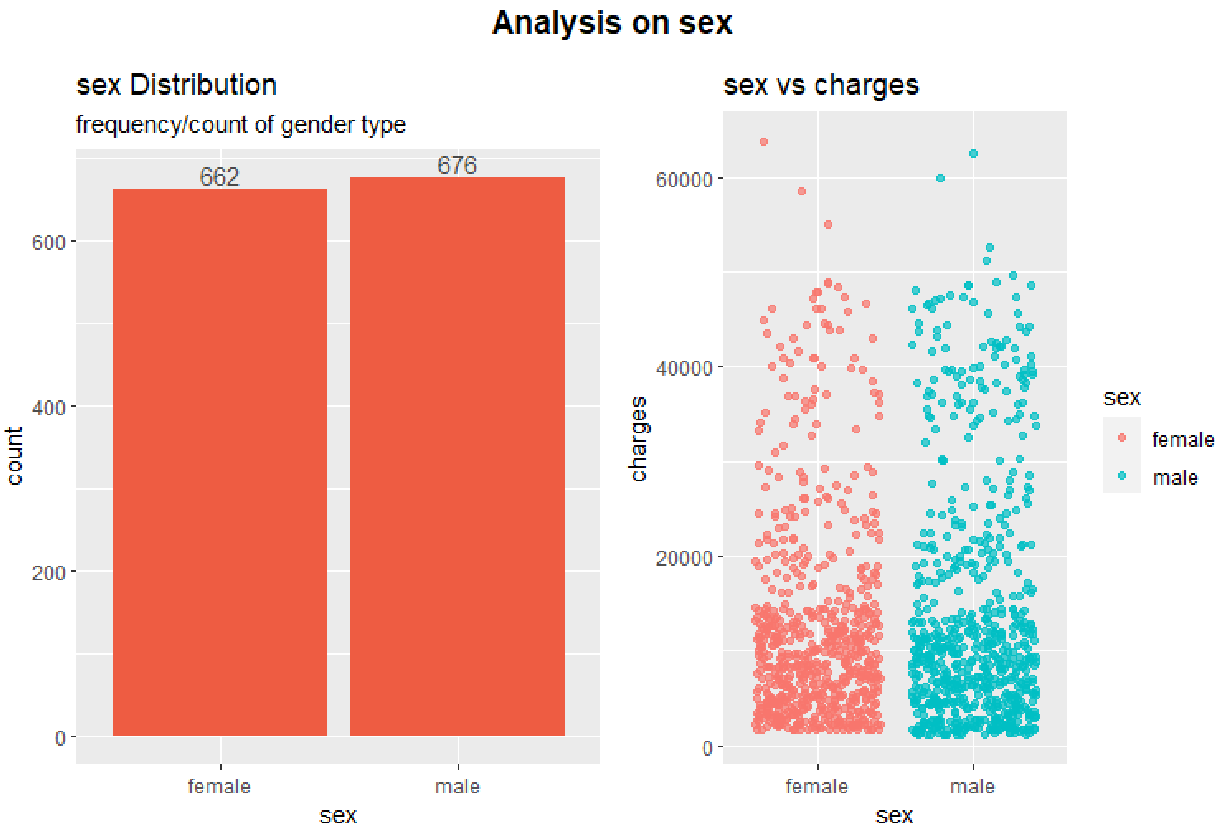
The associated insurance problem is to use this data to identify patterns and relationships between these features and healthcare costs. The data can be analyzed to understand which factors have the greatest impact on individuals. This information can be used to develop strategies for managing healthcare costs and controlling expenses for insurers. Additionally, the data may be used to identify high-risk individuals or groups and develop targeted interventions to improve health outcomes and reduce costs.

The data may be used to identify any disparities in healthcare costs between different groups of people based on demographic features such as sex, region etc. This information can be used to develop targeted policies and interventions to reduce healthcare disparities.

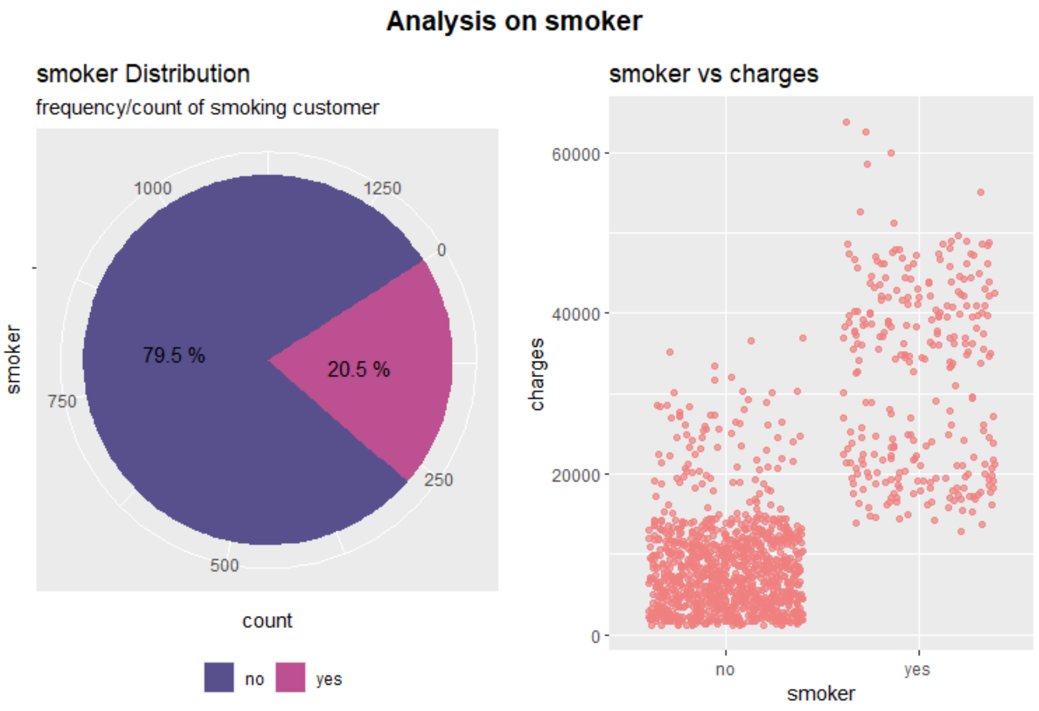
Overall, the medical cost personal data provides valuable insights into the factors that drive healthcare costs and can be used to inform policy decisions and improve healthcare outcomes for individuals. It can help insurance companies to predict the cost of the insurance policy for different individuals and help them to design their policies accordingly.

Exploratory Data Analysis (EDA)

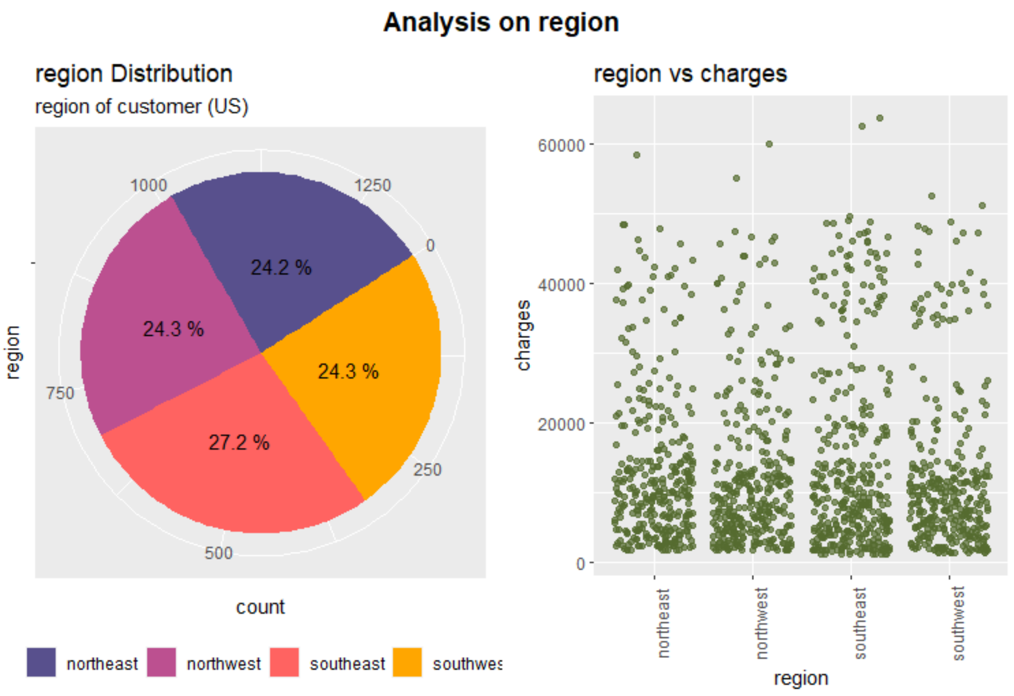
Analysis on categorical features: Categorical features consist of sex, smoker, and region.



Conclusion: Female vs male count/distribution balanced with slightly more male. There is no pattern on charges.

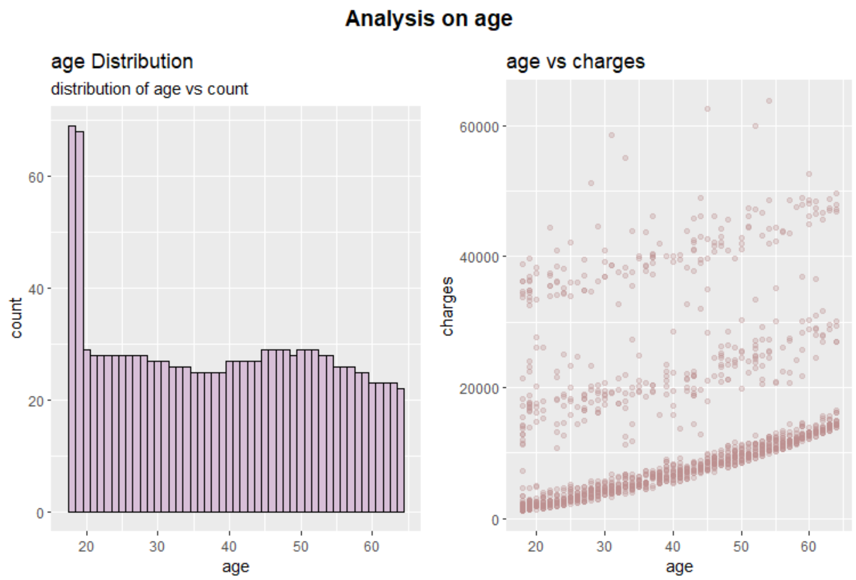


Conclusion: There are less smoker. Customer who smoker shows higher charges than those who don't.

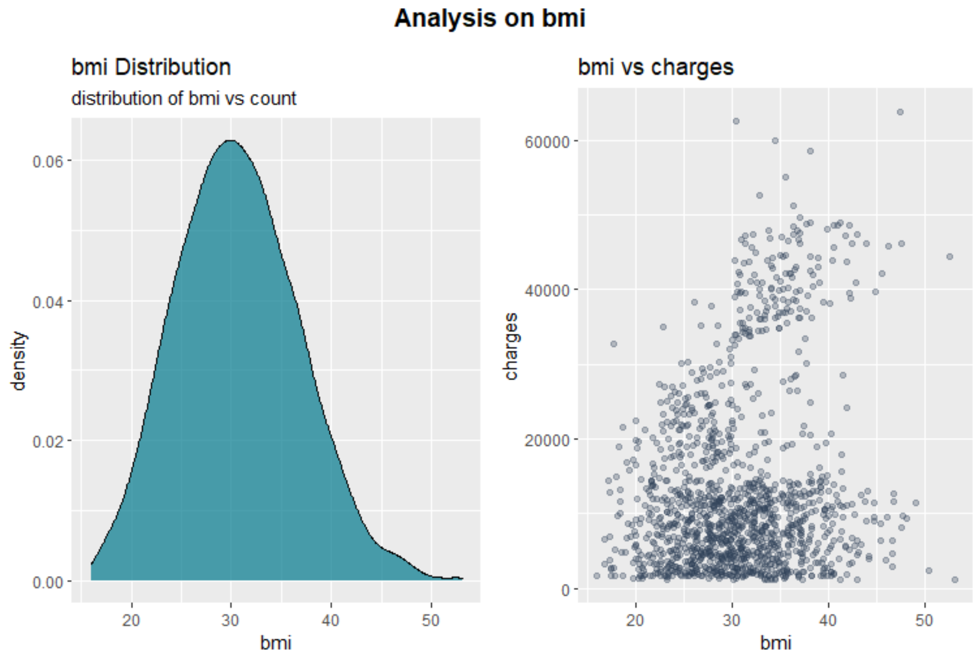


Conclusion: There are 4 region, with 'southeast' has slightly more count than the other 3 that was pretty balanced to each other. There is no pattern on charges.

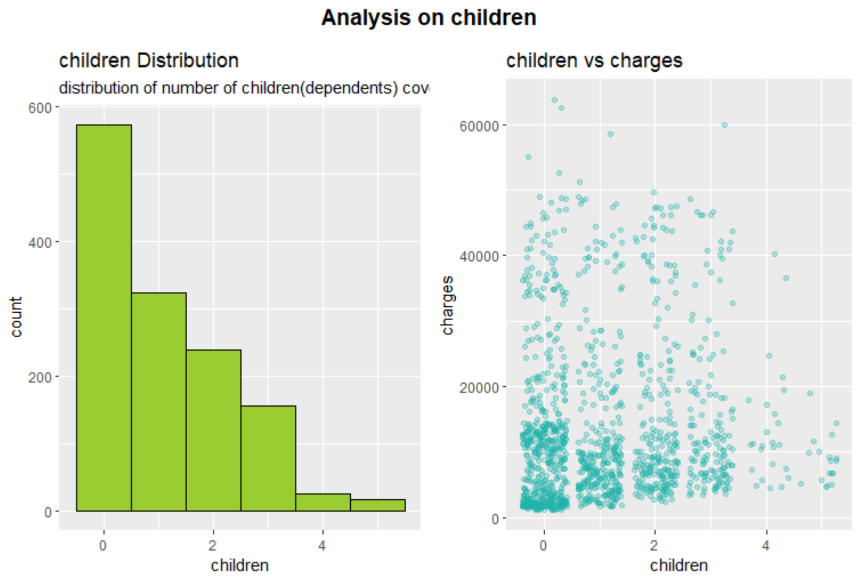
Distribution analysis on numerical features: age, bmi, children and charges are the numerical features.



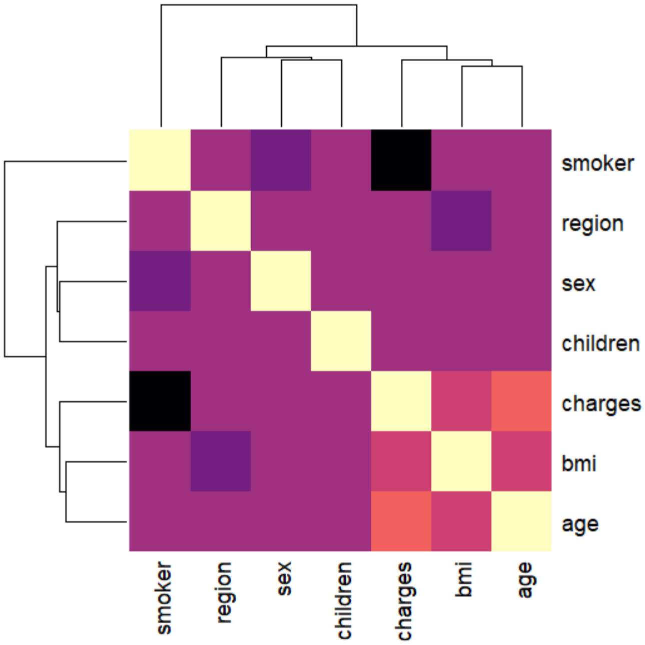
Conclusion: Ages have balanced distribution about 25-ish on age 20-64 with were more count on age 18 and 19. Age seems correspond linearly to charges. however, we can observe 3 cluster of charges, 0-20000, 20000ish, and 40000ish.



Conclusion: Bmi distributed normally with average of 30.66. There are slightly increase of charges as the bmi increase.

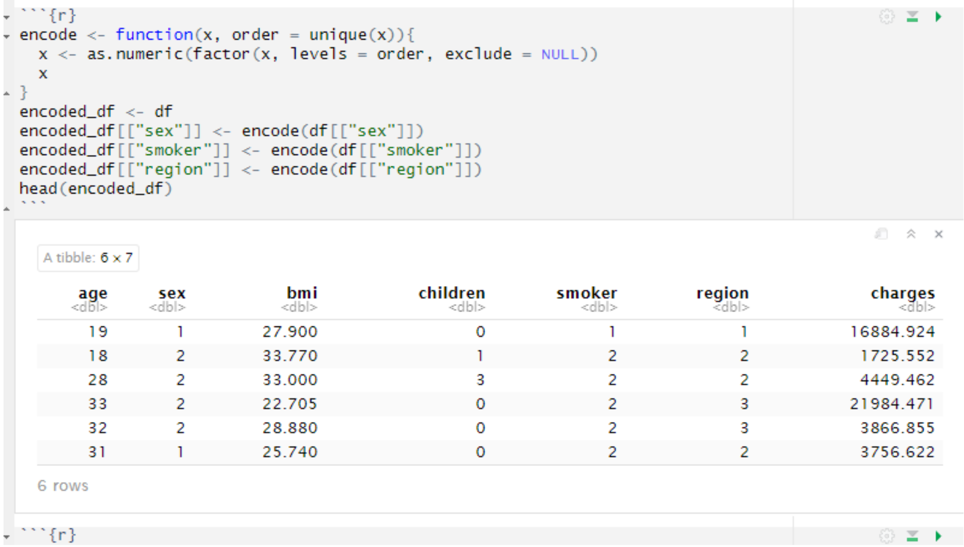


Conclusion: Most customer has no children, the more children count the less the counts. There is no pattern shown on charges.



Conclusion: As we expected, smoker has higher correlation to charges then other, there are also slight correlation to age and bmi.

Encoding:



~~~
```{r}
encode <- function(x, order = unique(x)){
  x <- as.numeric(factor(x, levels = order, exclude = NULL))
  x
}
encoded_df <- df
encoded_df[["sex"]] <- encode(df[["sex"]])
encoded_df[["smoker"]] <- encode(df[["smoker"]])
encoded_df[["region"]] <- encode(df[["region"]])
head(encoded_df)
```
~~~

A tibble: 6 x 7

age <dbl>	sex <dbl>	bmi <dbl>	children <dbl>	smoker <dbl>	region <dbl>	charges <dbl>
19	1	27.900	0	1	1	16884.924
18	2	33.770	1	2	2	1725.552
28	2	33.000	3	2	2	4449.462
33	2	22.705	0	2	3	21984.471
32	2	28.880	0	2	3	3866.855
31	1	25.740	0	2	2	3756.622

6 rows

~~~
```{r}
~~~

Brief description of the models used

Linear Regression Model: Linear regression is a statistical model that is used to understand the relationship between a dependent variable (in this case, healthcare costs) and one or more independent variables (such as age, sex, BMI, number of children, smoking and region). The basic idea behind linear regression is to find the line of best fit that minimize the difference between the predicted values and the actual values. The formula for a linear regression model is given by:

Y = b0 + b1X1 + ……. +bnXn

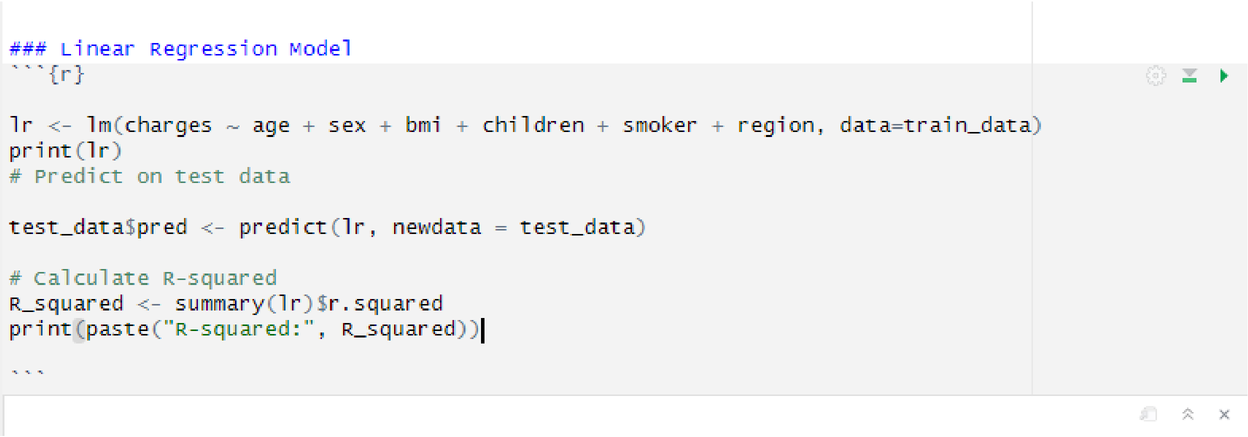
Where Y is the dependent variable (healthcare costs), X1, X2, …., Xn are the independent variables (such as age, sex, BMI, number of children, smoking and region), and b0, b1, b2, …, bn are the coefficients that are estimated by the model.

The coefficient b0 is the y-intercept of the line of best fit, and the coefficients b1, b2, …, bn represent the slope of the line of best fit for each independent variable. The coefficients are estimated using a method called least squares, which finds the line of best fit that minimizes the sum of the squared differences between the predicted values and the actual values.

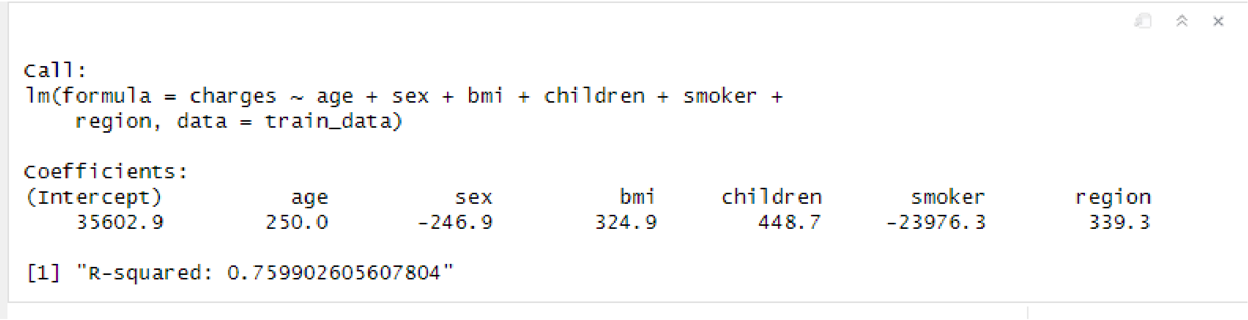
Once the coefficients are estimated, the linear regression model can be used to predict healthcare costs for new individuals based on their features. The model can also be used to identify which independent variables have the greatest impact on healthcare costs and how these costs vary between different groups of individuals.

It's important to mention that linear regression model assumes that the relationship between the independent variables and the dependent variables is linear, also it assumes that the errors are normally distributed and independent from each other.

Model Building: Split the data to train and test with the ratio 7:3. Then build our linear model.

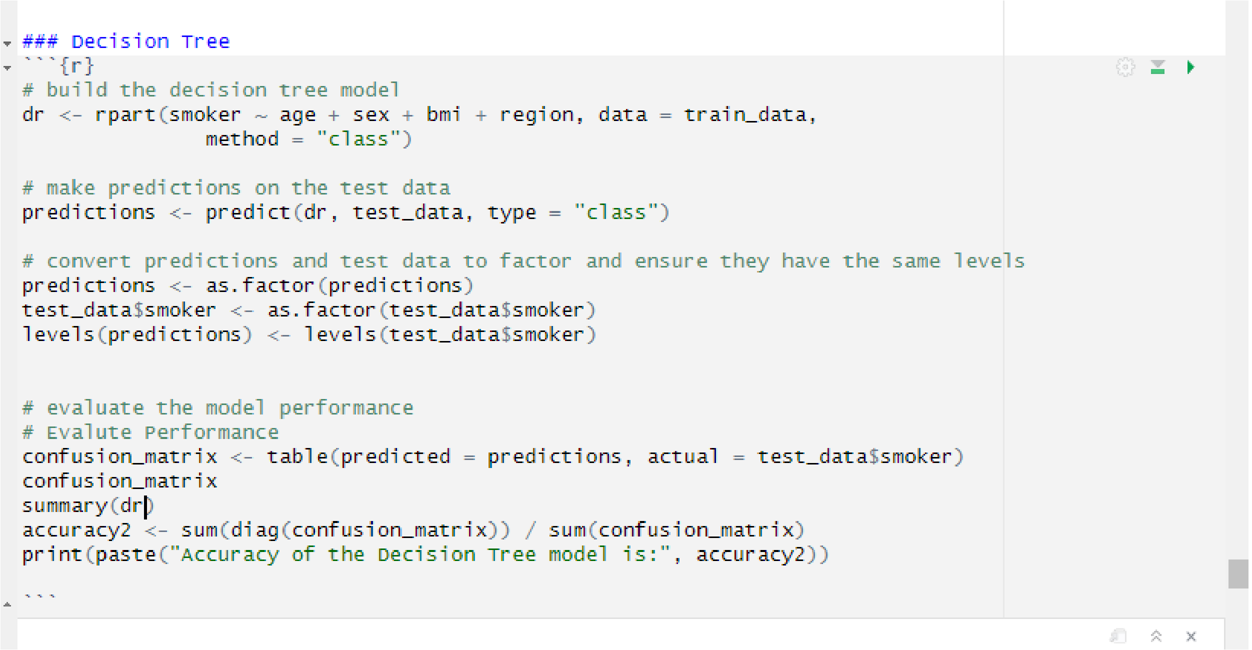


~~~
### Linear Regression Model
```{r}

lr <- lm(charges ~ age + sex + bmi + children + smoker + region, data=train_data)
print(lr)
# Predict on test data

test_data$pred <- predict(lr, newdata = test_data)

# Calculate R-squared
R_squared <- summary(lr)$r.squared
print(paste("R-squared:", R_squared))

```
~~~



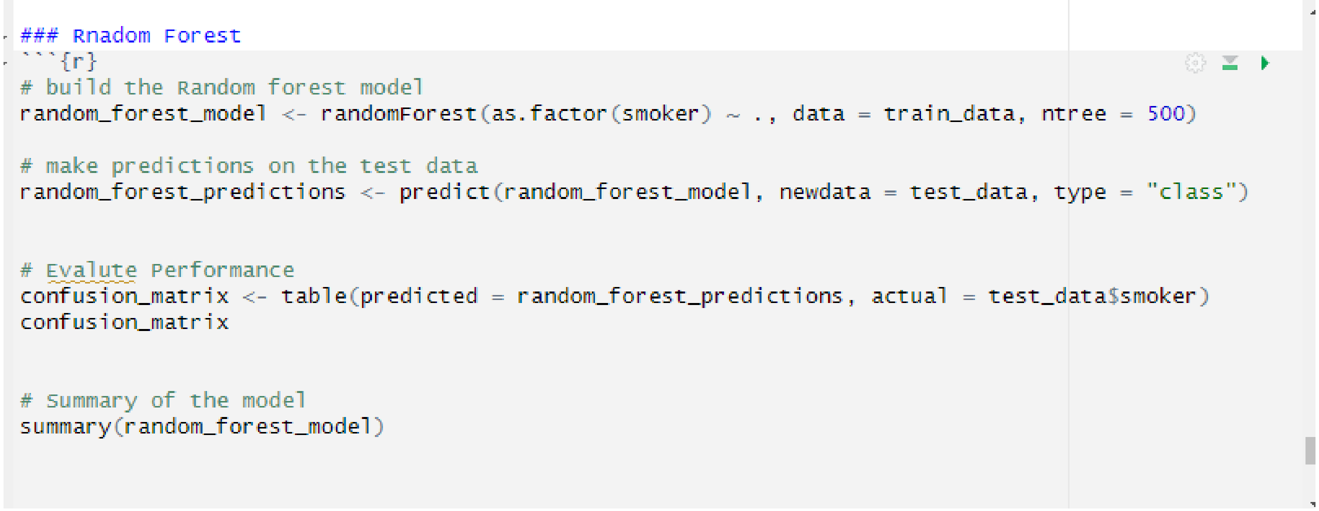
```
Call:
lm(formula = charges ~ age + sex + bmi + children + smoker + 
    region, data = train_data)

Coefficients:
(Intercept)          age          sex          bmi     children       smoker       region  
    35602.9        250.0       -246.9        324.9        448.7     -23976.3        339.3  

[1] "R-squared: 0.759902605607804"
```

Decision Tree: A decision tree is a type of supervised machine learning algorithm that is used for both classification and regression tasks. The algorithm works by recursively splitting the dataset into subsets based on the values of the input features. Each internal node of the tree represents a feature, and each leaf node represents a predicted label or value. The goal of the decision tree is to create a model that accurately predicts the value of the target variable based on the values of the input features.



~~~
### Decision Tree
```{r}
# build the decision tree model
dr <- rpart(smoker ~ age + sex + bmi + region, data = train_data,
                method = "class")

# make predictions on the test data
predictions <- predict(dr, test_data, type = "class")

# convert predictions and test data to factor and ensure they have the same levels
predictions <- as.factor(predictions)
test_data$smoker <- as.factor(test_data$smoker)
levels(predictions) <- levels(test_data$smoker)


# evaluate the model performance
# Evalute Performance
confusion_matrix <- table(predicted = predictions, actual = test_data$smoker)
confusion_matrix
summary(dr)
accuracy2 <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Accuracy of the Decision Tree model is:", accuracy2))

```
~~~

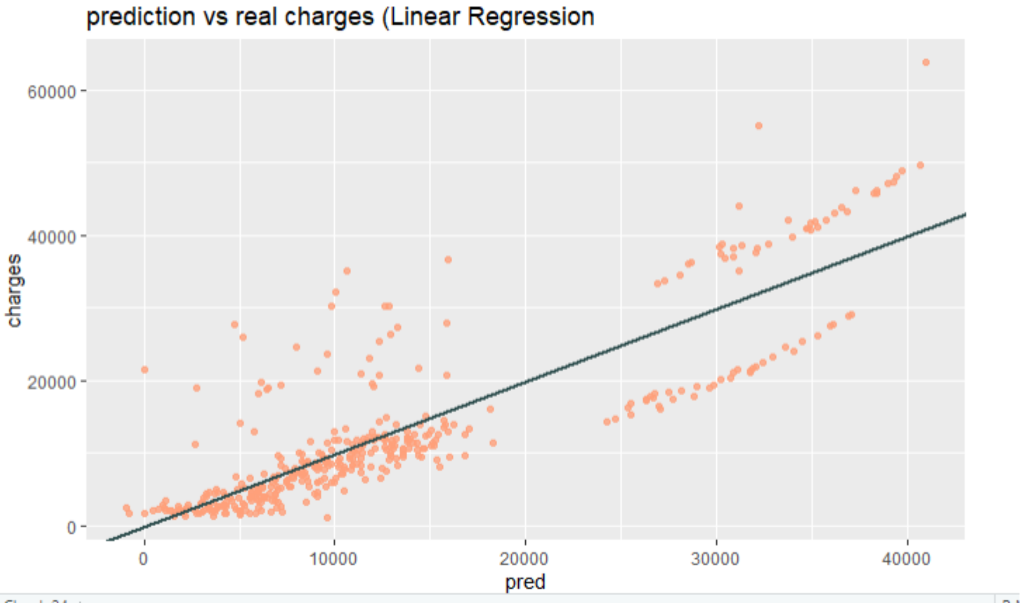
Random Forest: A random forest is an ensemble learning method for classification and regression that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual tress. In random forests, each tree in the ensemble is built from a sample drawn with replacement (bootstrap sample) from the training set. It is a more robust model compared to decision tree, it reduces the overfitting problem.



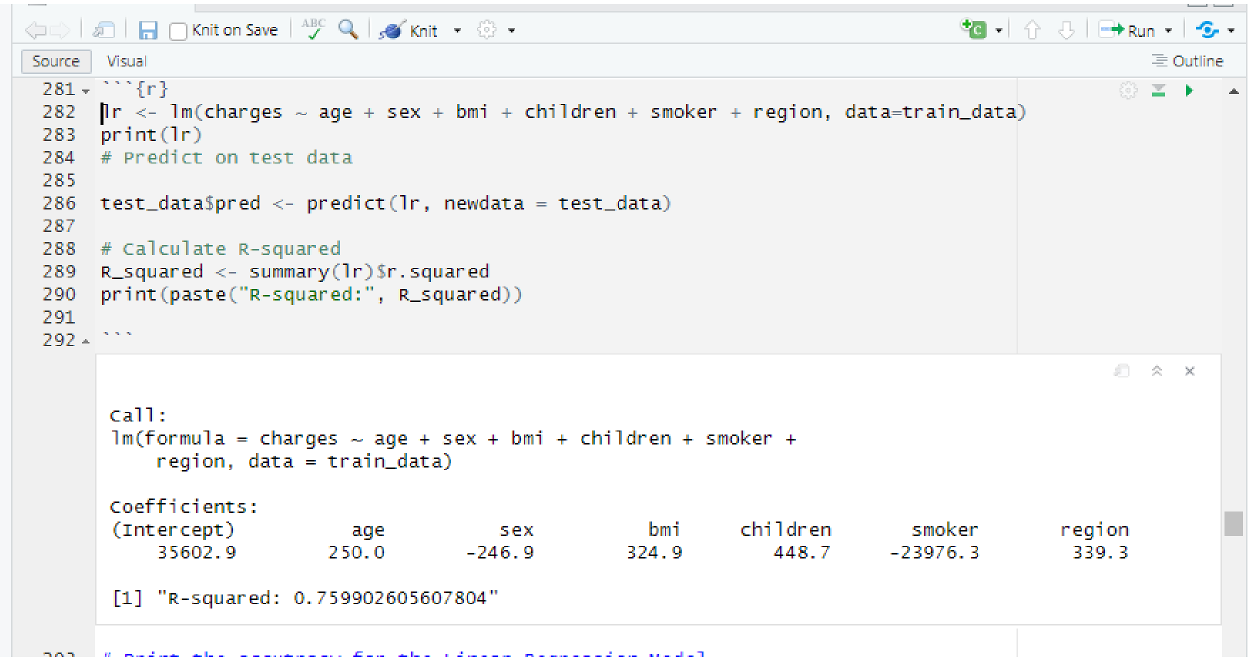
~~~
### Rnadom Forest
```{r}
# build the Random forest model
random_forest_model <- randomForest(as.factor(smoker) ~ ., data = train_data, ntree = 500)

# make predictions on the test data
random_forest_predictions <- predict(random_forest_model, newdata = test_data, type = "class")


# Evalute Performance
confusion_matrix <- table(predicted = random_forest_predictions, actual = test_data$smoker)
confusion_matrix


# Summary of the model
summary(random_forest_model)
~~~

Analysis of the results and conclusion

Output and accuracy of the Linear Model:



Result for Linear Model:



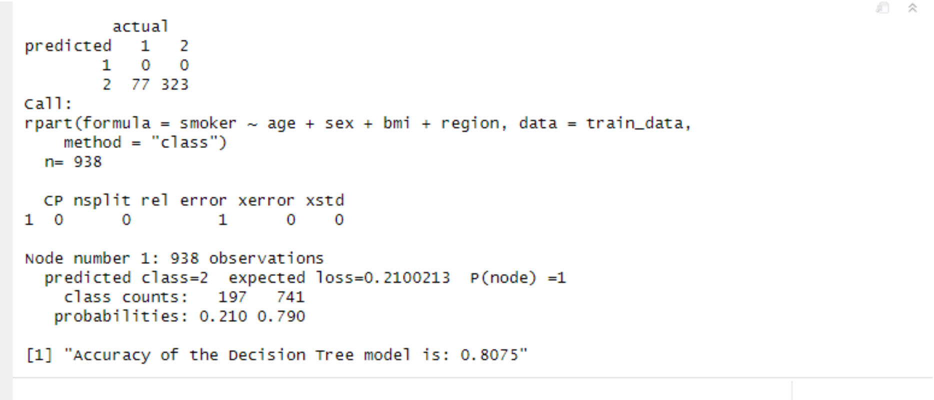
~~~
```{r}
lr <- lm(charges ~ age + sex + bmi + children + smoker + region, data=train_data)
print(lr)
# Predict on test data

test_data$pred <- predict(lr, newdata = test_data)

# Calculate R-squared
R_squared <- summary(lr)$r.squared
print(paste("R-squared:", R_squared))

```
~~~

```
Call:
lm(formula = charges ~ age + sex + bmi + children + smoker + 
    region, data = train_data)

Coefficients:
(Intercept)          age          sex          bmi     children       smoker       region  
    35602.9        250.0       -246.9        324.9        448.7     -23976.3        339.3  

[1] "R-squared: 0.759902605607804"
```

```
# Print the accuracy for the Linear Regression Model
```

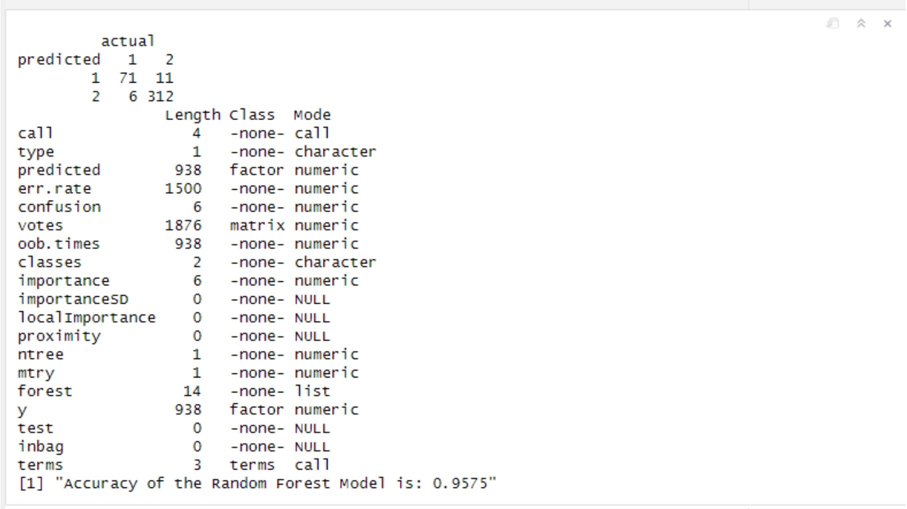
It's important to note that this graph can help you evaluate how well the model is performing by checking how closely the points are aligned to the line of perfect prediction. If the points are closely aligned to the line, the model is performing well. However, if the points are scattered away from the line, the model is not performing well and needs to be improved. In our case the R_squared error is 75.9

Result for Decision Tree:



```
         actual
predicted   1   2
        1   0   0
        2  77 323
Call:
rpart(formula = smoker ~ age + sex + bmi + region, data = train_data, 
    method = "class")
  n= 938 

  CP nsplit rel error xerror xstd
1  0      0         1      0    0

Node number 1: 938 observations
  predicted class=2  expected loss=0.2100213  P(node) =1
    class counts:   197   741
   probabilities: 0.210 0.790 

[1] "Accuracy of the Decision Tree model is: 0.8075"
```

Result of Random Forest Model:



```
         actual
predicted   1   2
        1  71  11
        2   6 312
                Length Class  Mode
call               4   -none- call
type               1   -none- character
predicted        938   factor numeric
err.rate        1500   -none- numeric
confusion          6   -none- numeric
votes           1876   matrix numeric
oob.times        938   -none- numeric
classes            2   -none- character
importance         6   -none- numeric
importanceSD       0   -none- NULL
localImportance    0   -none- NULL
proximity          0   -none- NULL
ntree              1   -none- numeric
mtry               1   -none- numeric
forest            14   -none- list
y                938   factor numeric
test               0   -none- NULL
inbag              0   -none- NULL
terms              3   terms  call
[1] "Accuracy of the Random Forest Model is: 0.9575"
```

Conclusion

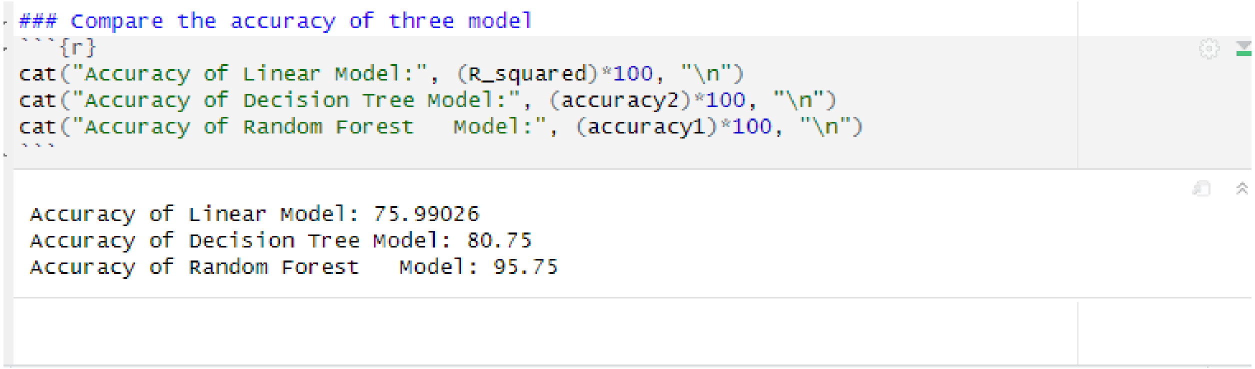
We first performed Exploratory Data Analysis (EDA) on the the data to understand the distribution and correlation of the variables. We then built and evaluated the following models: linear model, random forest, and decision tree.

The decision tree model was built the rpart package in R and had an accuracy of 80.75. The random forest model was built using randomForest package in R and had an accuracy of 95.75. The linear regression model was built using lm function in R and had a R_squared of 75.99. As we can see the random forest model performed the best as compared to the linear regression model. The linear regression model performance can be evaluated by R_squared.

It's important to note that the accuracy of a model is not the only metric to consider when comparing models, it’s also important to consider other factors such as interpretability, computational cost, and generalization ability. Moreover, the accuracy of a model can be affected by factors such as the nature of your data, the model’s parameter and the split of the data into training and testing sets. So, it’s always a good idea to evaluate the model performance on a separate validation dataset.

Overall, we can conclude that random forest performed the best on the insurance data and can be used for further predictions.

The accuracy of these three models is given below.



~~~
### Compare the accuracy of three model
```{r}
cat("Accuracy of Linear Model:", (R_squared)*100, "\n")
cat("Accuracy of Decision Tree Model:", (accuracy2)*100, "\n")
cat("Accuracy of Random Forest   Model:", (accuracy1)*100, "\n")
```
~~~

```
Accuracy of Linear Model: 75.99026
Accuracy of Decision Tree Model: 80.75
Accuracy of Random Forest   Model: 95.75
```