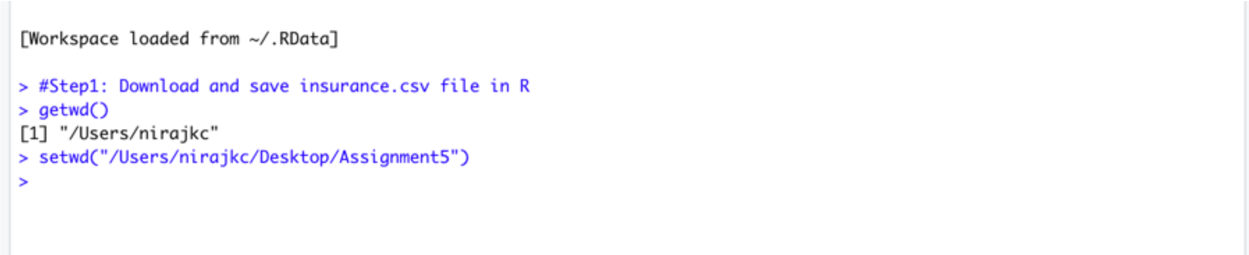
Project: Building regression model for insurance dataset

Step 1: Collecting Data

Download the insurance.csv file which is attached and save it to your R working folder.

# I have downloaded the insurance.csv file. I have downloaded this CSV file in my working folder on the desktop section of my computer.



```
[Workspace loaded from ~/.RData]

> #Step1: Download and save insurance.csv file in R
> getwd()
[1] "/Users/nirajkc"
> setwd("/Users/nirajkc/Desktop/Assignment5")
>
```

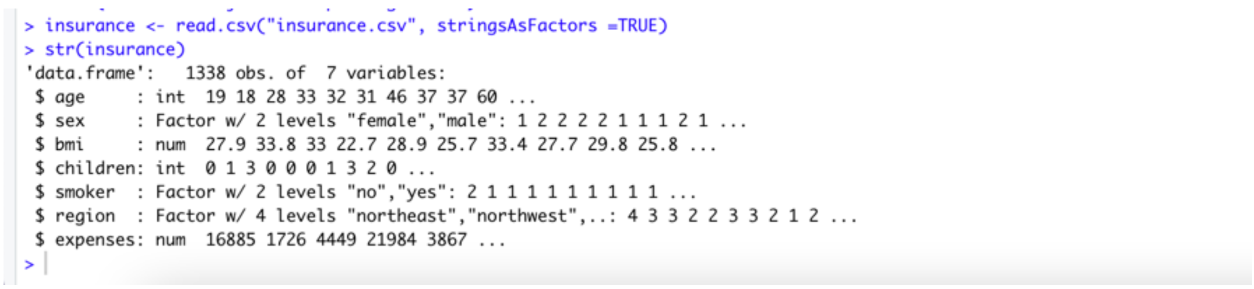
#setwd will set the working directory.

Give some thoughts on how these variables may be related to billed medical expenses.

#These variables like age, bmi, children, and smokers are related to billed medical expenses. For example, we can expect that older people who are addicted to smoking will have higher medical bills, and these medical bills will increase with an increase in age. Also, we can assume that medical bills will be more if people will be older and have the habit of smoking.

Step 2: Exploring and preparing the data:

A. Read your csv file and confirms that the data is formatted as we had expected.

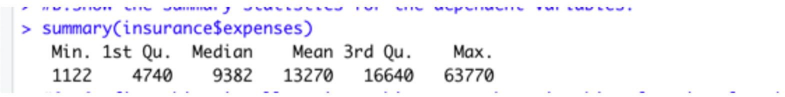


```
> insurance <- read.csv("insurance.csv", stringsAsFactors =TRUE)
> str(insurance)
'data.frame':	1338 obs. of  7 variables:
 $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
 $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
 $ bmi     : num  27.9 33.8 33 22.7 28.9 25.7 33.4 27.7 29.8 25.8 ...
 $ children: int  0 1 3 0 0 0 1 3 2 0 ...
 $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
 $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
 $ expenses: num  16885 1726 4449 21984 3867 ...
>
```

#read.csv will read CSV file from the current working directory and stringasFactor=TRUE will convert three nominal variables sex, smoker, and region into factor. The str() function confirms that the data are formatted as we had expected .

The seven features of the dataset are age, sex, bmi, children, smoker, region. The target variable is the expenses. Among these features sex, smoker & region are character types, whereas age, bmi, children, and expenses are numeric type variables.

B. Show the summary statistics for the dependent variable

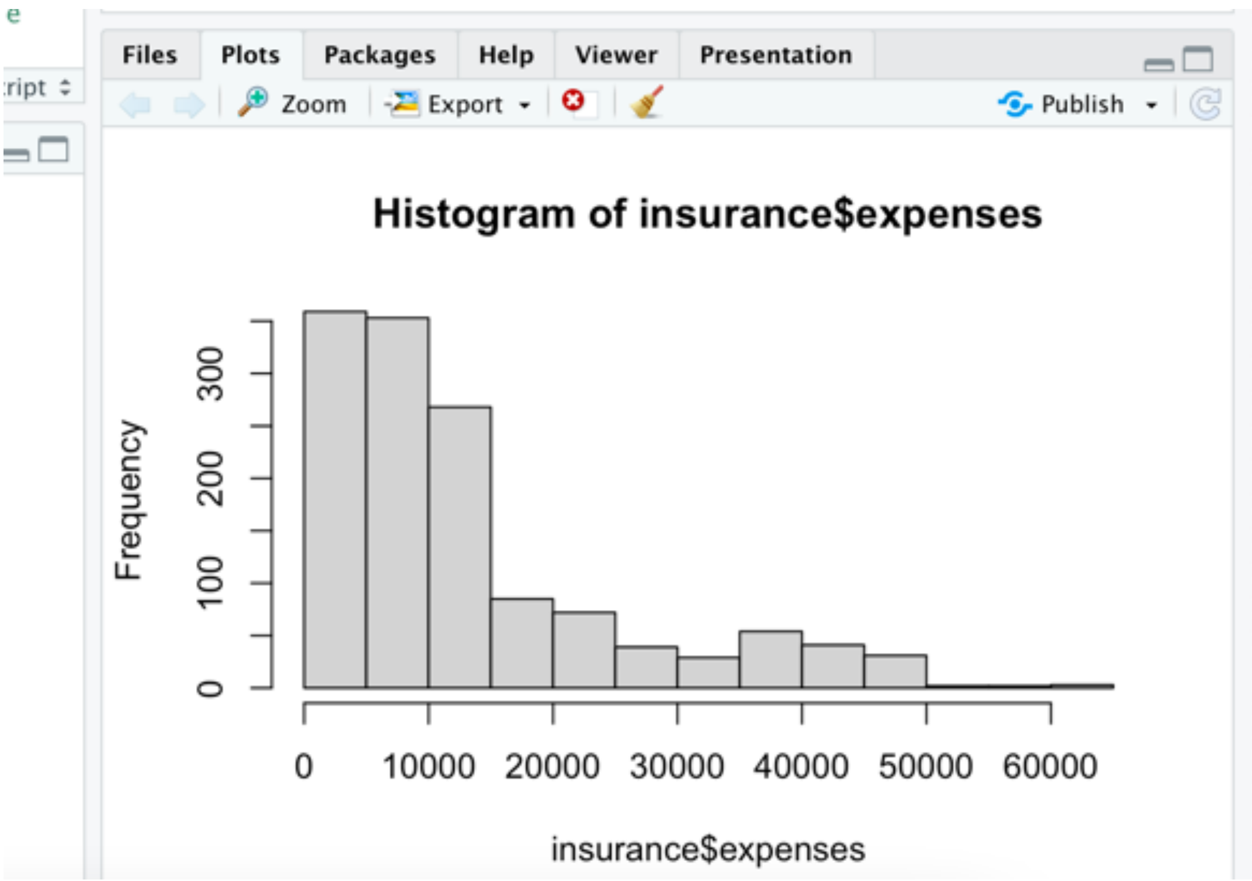


```
> summary(insurance$expenses)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1122    4740    9382   13270   16640   63770 
```

#The dependent variable is expenses, to look at summary statistics of expenses, we will use function summary(). From the summary function, we came to know that Mean is greater than the Median, which implies that the distribution of the expenses is right-skewed. We will confirm it by histogram.

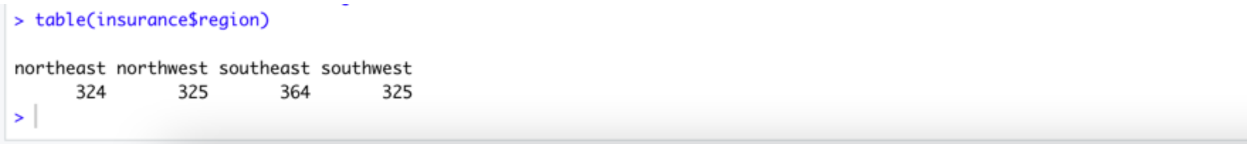
C: Confirm this visually using a histogram, by using hist function for the dependent variable and give your analysis from the histogram.

```
> hist(insurance$expenses)
```



#From the histogram, we came to know that majority of the people have expenses from 0 to $15,000. The tail of the distribution extends past these peaks, this data is not ideal for linear regression as linear regression assumes normal distribution for the dependent variables, we will design the best-suited model later on. Another problem is that linear regression requires features to be numerical, in our dataset we have three-factor variables.

D. Show the four levels of the `region` variable.

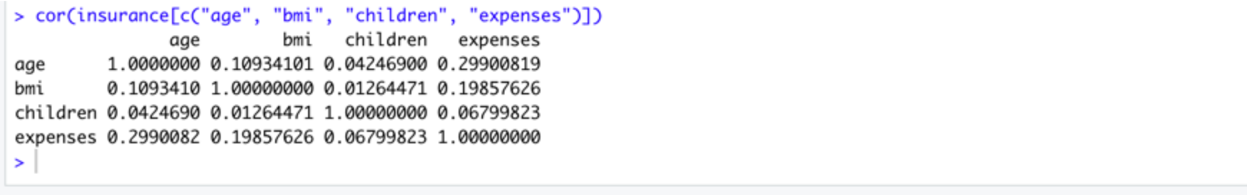


```
> table(insurance$region)

northeast northwest southeast southwest 
      324       325       364       325 
> 
```

#Using the table function, we came to know that data is divided into four regions-northeast, northwest, southeast, and southwest and all the regions have the nearly same amount of data. i.e., data is evenly divided.

E. Exploring relationships among features–the correlation matrix

Create a correlation matrix for the four numeric variables in the insurance data frame, and use the cor() command.

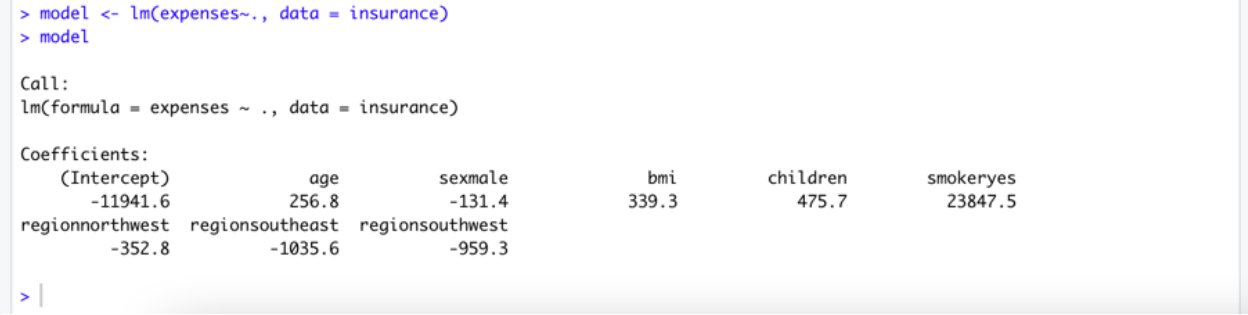


```
> cor(insurance[c("age", "bmi", "children", "expenses")])
               age        bmi   children   expenses
age      1.0000000 0.10934101 0.04246900 0.29900819
bmi      0.1093410 1.00000000 0.01264471 0.19857626
children 0.0424690 0.01264471 1.00000000 0.06799823
expenses 0.2990082 0.19857626 0.06799823 1.00000000
> 
```

#Before fitting the regression model, it is useful to know how independent and dependent variables are related to each other by using a correlation matrix, the cor() function only works for numeric variables. The correlations are always listed in the rows and columns, the correlation between the variable and itself is 1 and it is always located diagonally. The value above and below the diagonal is always the same since the correlation is symmetrical i.e., cor(x,y)=cor(y,x). As seen in the above correlation matrix, there is a moderately strong positive correlation between expenses and age, expenses and bmi, expenses and children, which implies that insurance expenses increase as the age, bmi and number of children/dependent increases.

Step 3: Training a model on the data.

Using lm() function to fit a linear regression model to your data



```
> model <- lm(expenses~., data = insurance)
> model

Call:
lm(formula = expenses ~ ., data = insurance)

Coefficients:
    (Intercept)              age          sexmale              bmi         children        smokeryes  
       -11941.6            256.8           -131.4            339.3            475.7          23847.5  
regionnorthwest  regionsoutheast  regionsouthwest  
         -352.8          -1035.6           -959.3  

> 
```

#For this step we need stats.package, which was already installed during R installation by default.
#The first argument used in the lm() function is expanses which is the target variable. The R formula syntax uses the tilde character to describe the model, the independent variables go to the right side of the tilde. The dot sign specifies all the independent variables used. The second argument is data that specifies the data frame in which expenses and independent variables can be found.
#At last type the model name "model" to see the estimated beta coefficient.

After building the model, show estimated beta coefficients, and what the beta coefficients indicate.
#The estimated beta coefficient is shown in the above figure.
#The **intercept** is the predicted value of the expanses when all the independent variables are zero.
#The beta coefficient indicates the estimated increase in expenses for an increase of one in each of the features. For example: after each additional year of age, expanses increase by $256.8 each year.

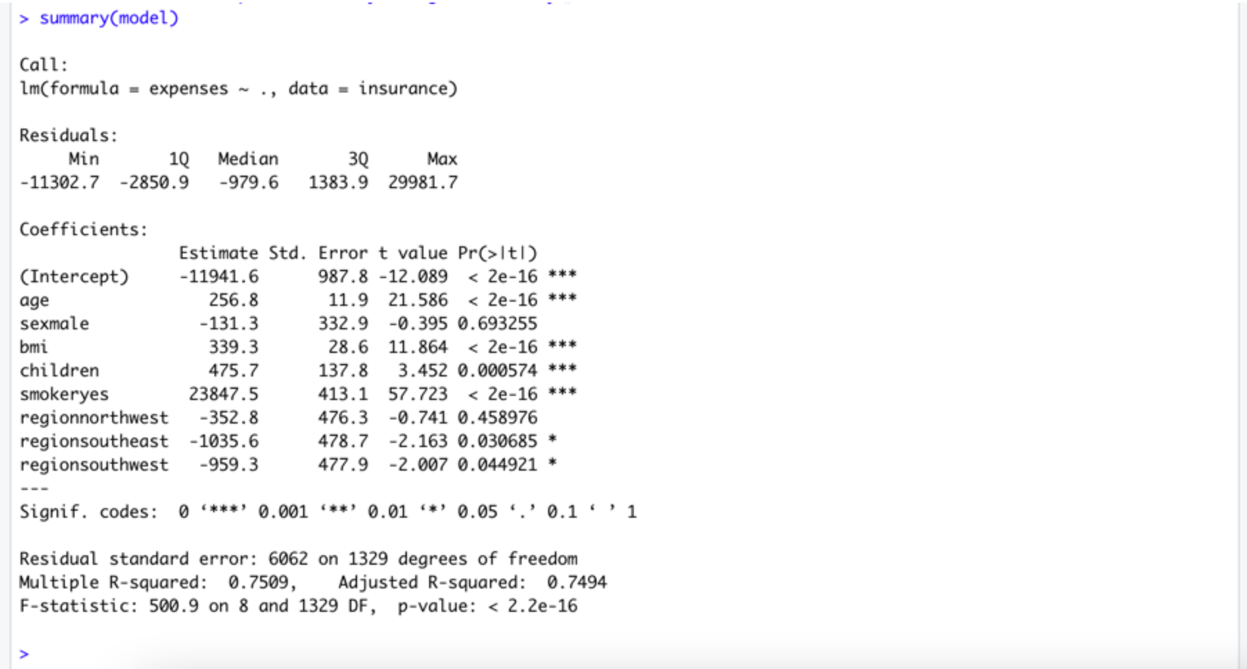
Dummy coding:
#The lm() function automatically applies techniques known as dummy coding for each of the factors type variables. Dummy coding allows nominal features to be treated as numeric by creating variables. For instance, sex has two categories male and female, this will split sex variables into two binary variables sexmale and sex female. For observation sex=male, it would assign sexmale =1 and sexfemale=0 and vice-versa. The same would be the case for others factor-type variables too.

#As seen in the above result, when adding a dummy variable to the regression model, one category is always left out, like sexfemale in our case which is left out variable. This left-out category is known as the reference category. The estimates are then interpreted as relatives to the reference category. The negative value -131.4 of sexmale indicates that males will have $131.4 less medical expenses each year than females.

#The result of the linear regression model indicates that people with smoking habits with children, higher bmi, and old age people should pay more expenses bills. People with more children should visit hospital quite often which could have added more expanses.

Step 4: Evaluating model performance.

Evaluate the model performance by using the summary() command on the stored model.



```
> summary(model)

Call:
lm(formula = expenses ~ ., data = insurance)

Residuals:
     Min       1Q   Median       3Q      Max 
-11302.7  -2850.9   -979.6   1383.9  29981.7 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)     -11941.6      987.8 -12.089  < 2e-16 ***
age                256.8       11.9  21.586  < 2e-16 ***
sexmale           -131.3      332.9  -0.395 0.693255    
bmi                339.3       28.6  11.864  < 2e-16 ***
children           475.7      137.8   3.452 0.000574 ***
smokeryes        23847.5      413.1  57.723  < 2e-16 ***
regionnorthwest   -352.8      476.3  -0.741 0.458976    
regionsoutheast  -1035.6      478.7  -2.163 0.030685 *  
regionsouthwest   -959.3      477.9  -2.007 0.044921 *  
---
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: 500.9 on 8 and 1329 DF,  p-value: < 2.2e-16

>
```

#while building the model in earlier steps, typing the stored model name , it only tells us how the independent variables are related to the dependent variables, but it did not tell us about how the model performs.

#By using the summary() function, it provides the three ways to evaluate the performance or fit of our model- the residual, coefficient, and the multiple R-squared values.

What they mean by these three keys of the output

Residual:

- Residual provides the summary statistics for the error in our prediction.
- The residual value is the true value minus the estimated value.
- Max error 29981.7 suggests that the model underestimates the prediction by nearly $30,000 for at least one observation.
- By seeing the results, 50% of the error lies within the first and third quartile i.e. majority of predictions were between 2850.9 over the true value and $1383.90 under the true value.

Coefficient:

- The p value pr(>|t|) provides an estimated probability that the true coefficient is zero.

- A smaller p-value indicates the true coefficient unlikely to be zero which means the feature is unlikely to have a relationship with the dependent variables.
- Some of the p-values have (***), which indicates the significance level met by the estimates. This is the threshold level chosen prior to building the model.
- That threshold level indicates the real findings due to chance alone only. So, a p-value less than the significance level is considered statistically significant.
- The small number of such significant variables in the model indicates that the features used are not predictive of the outcomes of the model, but our model has several variables.

The multiple R squared value:

- The multiple R squared is also known as the coefficient of determination.
- It explains how well our model explains the value of the dependent variable.
- It is like the correlation coefficient if the value is close to 1, it indicates a perfect model.
- Here, the R-squared value is 0.7494, which states that our model explains about 75% of the variation in the dependent variable. The model with more features always explains more variation, the adjusted R-squared value corrects R-squared by penalizing models with many independent variables.

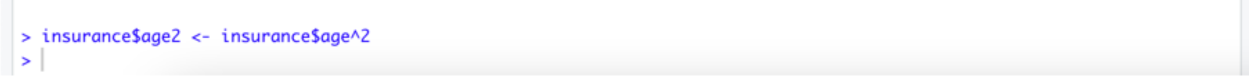
After analyzing these three performance indicators, our model is performing well, in the next step, we will see how the model performs after the improvement of the model.

Step5: Improving model performance

Suggest and work with one way to improve the model performance

#The main difference between regression modeling and other machine learning approaches is that regression typically leaves feature selection to the user themselves, by doing feature selection we can improve the performance of the model. We can improve the model performance by doing following things:

i) Adding non-linear relationship: With the increase in age, the reaction between an age and expenses may not be linear, with the increase in age, the treatment may be extremely high. We will add higher order to the regression model. In the code, we will add both expenses age +age2, here age2 is the polynomial degree of the age.

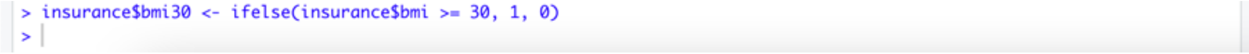


```
> insurance$age2 <- insurance$age^2
> 
```

#age2 variable is obtained by squaring the age variable.

ii) Transformation: Converting numeric variables to a binary indicator:

# The bmi may have an impact on the medical expenses only after a certain bmi value i.e. 30 or above. For this, we can create a binary obesity indicator variable that is 1, if the bmi is at least 30, and 0 if less than bmi < 30. The estimated beta binary features would then indicate the impact on medical expenses.



```
> insurance$bmi30 <- ifelse(insurance$bmi >= 30, 1, 0)
> 
```

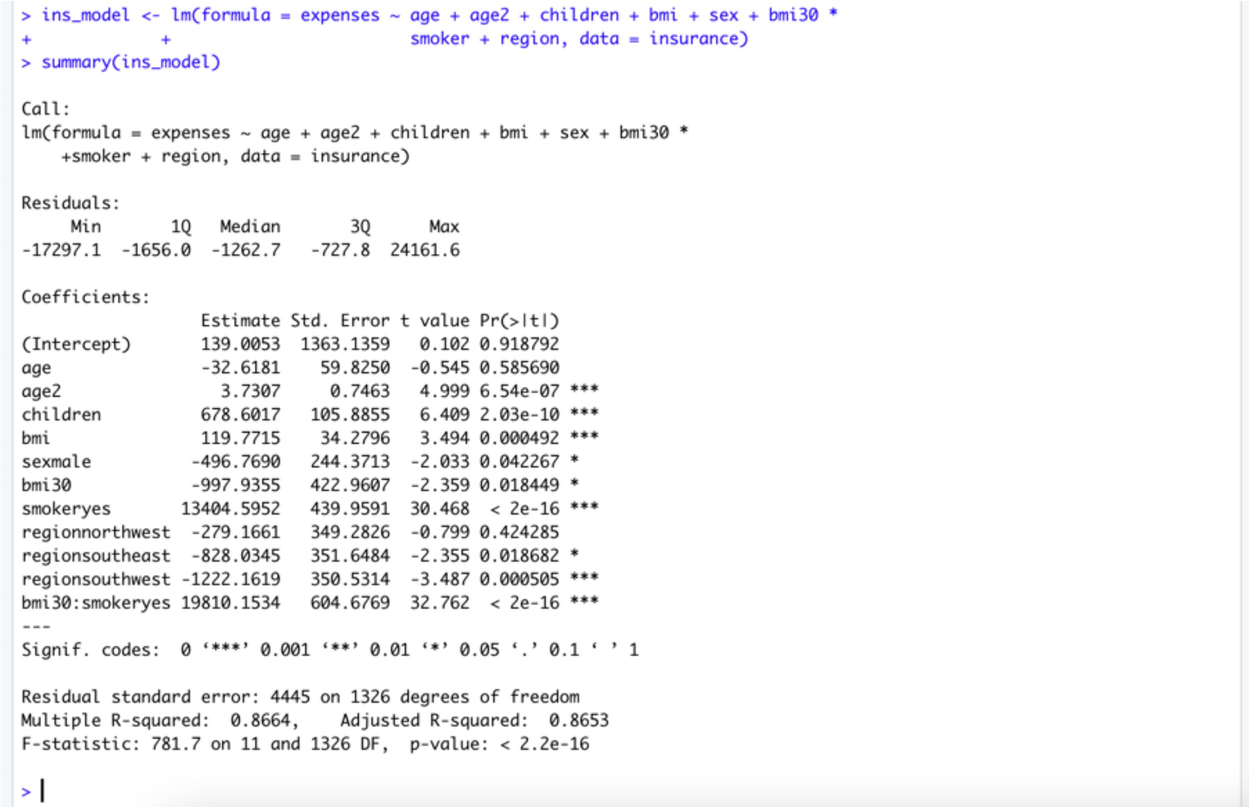
In the above code, binary indicator bmi30 is created using if-else statement for assigning the binary values.

iii) Adding interaction effects:

#Smoking and obesity may increase medical expenses rather than sum of each variables alone. This combined effect is known as interaction.

#We will use the formula in the form: expense~bmi30 *smoker where * is shorthand that instruct R to model expense~bmi30+ smokeryes + bmi30:smokeryes

The colon(:) operator indicates interaction between two variables.

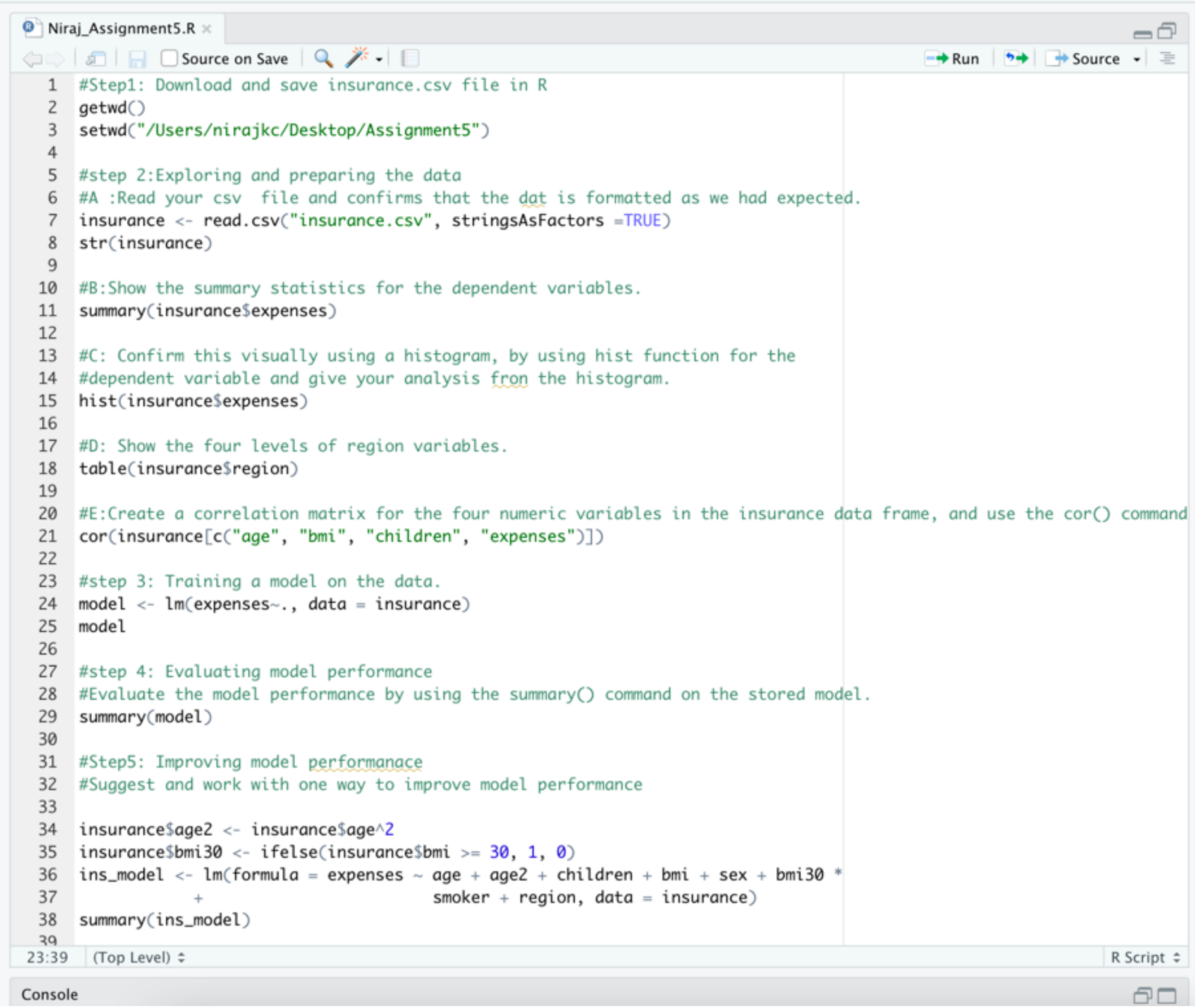


```
> ins_model <- lm(formula = expenses ~ age + age2 + children + bmi + sex + bmi30 *
+             +                       smoker + region, data = insurance)
> summary(ins_model)

Call:
lm(formula = expenses ~ age + age2 + children + bmi + sex + bmi30 *
    +smoker + region, data = insurance)

Residuals:
     Min       1Q   Median       3Q      Max
-17297.1  -1656.0  -1262.7   -727.8  24161.6

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       139.0053  1363.1359   0.102 0.918792
age               -32.6181    59.8250  -0.545 0.585690
age2                3.7307     0.7463   4.999 6.54e-07 ***
children          678.6017   105.8855   6.409 2.03e-10 ***
bmi               119.7715    34.2796   3.494 0.000492 ***
sexmale          -496.7690   244.3713  -2.033 0.042267 *
bmi30            -997.9355   422.9607  -2.359 0.018449 *
smokeryes       13404.5952   439.9591  30.468  < 2e-16 ***
regionnorthwest  -279.1661   349.2826  -0.799 0.424285
regionsoutheast  -828.0345   351.6484  -2.355 0.018682 *
regionsouthwest -1222.1619   350.5314  -3.487 0.000505 ***
bmi30:smokeryes 19810.1534   604.6769  32.762  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4445 on 1326 degrees of freedom
Multiple R-squared:  0.8664,	Adjusted R-squared:  0.8653
F-statistic: 781.7 on 11 and 1326 DF,  p-value: < 2.2e-16

> |
```

Result:

After modification, the R-squared value improved from 0.75 to 0.87. The adjusted R-squared value also increased from 0.75 to 0.87. Our model now explains 87% of the variation in medical treatment cost. Modified higher-order age2 and obesity indicator bmi30 are also statistically significant according to the model. The model shows that obese smokers spend another $19810 per year then $13404 for smoking alone. This model suggests that the smoking with obesity increases medical expanses.

Screenshot of all codes:



```r
#Step1: Download and save insurance.csv file in R
getwd()
setwd("/Users/nirajkc/Desktop/Assignment5")

#step 2:Exploring and preparing the data
#A :Read your csv  file and confirms that the dat is formatted as we had expected.
insurance <- read.csv("insurance.csv", stringsAsFactors =TRUE)
str(insurance)

#B:Show the summary statistics for the dependent variables.
summary(insurance$expenses)

#C: Confirm this visually using a histogram, by using hist function for the
#dependent variable and give your analysis fron the histogram.
hist(insurance$expenses)

#D: Show the four levels of region variables.
table(insurance$region)

#E:Create a correlation matrix for the four numeric variables in the insurance data frame, and use the cor() command
cor(insurance[c("age", "bmi", "children", "expenses")])

#step 3: Training a model on the data.
model <- lm(expenses~., data = insurance)
model

#step 4: Evaluating model performance
#Evaluate the model performance by using the summary() command on the stored model.
summary(model)

#Step5: Improving model performanace
#Suggest and work with one way to improve model performance

insurance$age2 <- insurance$age^2
insurance$bmi30 <- ifelse(insurance$bmi >= 30, 1, 0)
ins_model <- lm(formula = expenses ~ age + age2 + children + bmi + sex + bmi30 *
            +                     smoker + region, data = insurance)
summary(ins_model)
```