Assuming we have been provided with a sample data set (medExpense.csv) that includes 1,340 examples of beneficiaries currently enrolled in a health insurance plan, with attributes identifying the characteristics of the insured individual and the total medical expenses charged for the calendar year. Assume you have been tasked with a group discussion and interpretation of this information. We'll show you how to use RStudio to cope with this problem. For this data analysis, we will be using RStudio.

Load the dataset
Code Snippet
Importing the dataset
dataset <- read.csv('medExpense.csv', stringsAsFactors = TRUE)

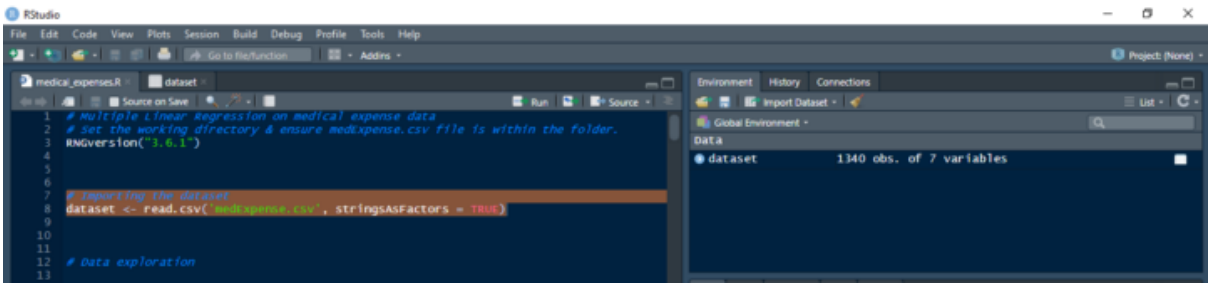


Image 1: Start Rstudio and load the data collection.

All right, let's fire up RStudio and import our data collection to begin. Using the read.csv procedure (Image 1), we can read in the CSV file containing the dataset. Since character data cannot be used directly in machine learning methods, we set stringsAsFactors = TRUE to cause an automatic conversion to Factors. Image 2 displays the dataset after it has been loaded.

RStudio Source Editor

dataset

Filter

	age	sex	bmi	children	smoker	region	expenses
1	19	female	27.9	0	yes	southwest	16884.92
2	18	male	33.8	1	no	southeast	1725.55
3	28	male	33.0	3	no	southeast	4449.46
4	33	male	22.7	0	no	northwest	21984.47
5	32	male	28.9	0	no	northwest	3866.86
6	31	female	25.7	0	no	southeast	3756.62
7	46	female	33.4	1	no	southeast	8240.59
8	37	female	27.7	3	no	northwest	7281.51
9	37	male	29.8	2	no	northeast	6406.41
10	60	female	25.8	0	no	northwest	28923.14
11	25	male	26.2	0	no	northeast	2721.32
12	62	female	26.3	0	yes	southeast	27808.73
13	23	male	34.4	0	no	southwest	1826.84
14	56	female	39.8	0	no	southeast	11090.72
15	27	male	42.1	0	yes	southeast	39611.76
16	19	male	24.6	1	no	southwest	1837.24
17	52	female	30.8	1	no	northeast	10797.34
18	23	male	23.8	0	no	northeast	2395.17
19	56	male	40.3	0	no	southwest	10602.39
20	30	male	35.3	0	yes	southwest	36837.47
21	60	female	36.0	0	no	northeast	13228.85
22	30	female	32.4	1	no	southwest	4149.74
23	18	male	34.1	0	no	southeast	1137.01
24	34	female	31.9	1	yes	northeast	37701.88
25	37	male	28.0	2	no	northwest	6203.90
26	59	female	27.7	3	no	southeast	14001.13

Image 2: Rstudio data set loaded.

Data exploration

Code Snippet

```
## Returns the first parts of the data frame
head(dataset)
## Returns the latter parts of the data frame
tail(dataset)
## Compactly display the internal structure of the data frame
str(dataset)
## Summarize medical expenses
summary(dataset$expenses)
## Histogram of medical expenses
hist(dataset$expenses)
## Table of region
table(dataset$region)
```

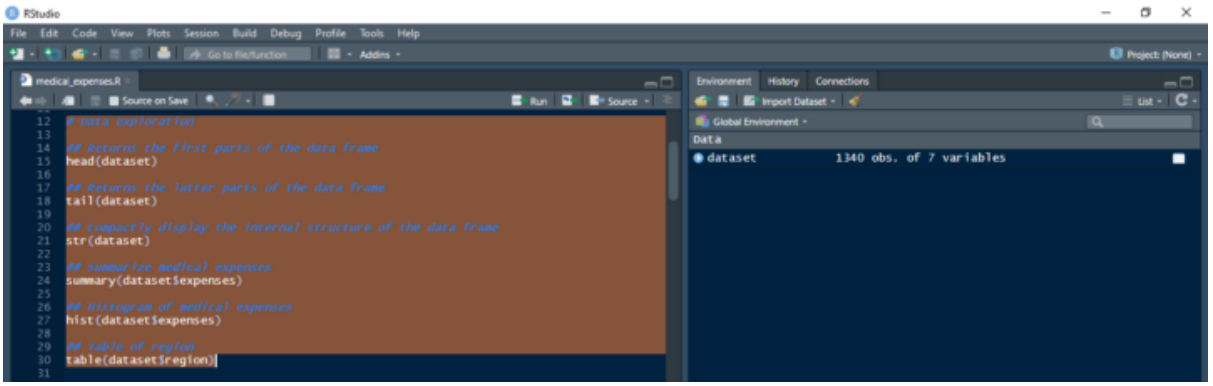


Image 3:Rstudio data analysis.

Now is a good time to investigate our data collection (Image 1). The head & str function was used to initially investigate the data set. The initial bits of the data frame (Image 4) are what are returned by the head function. The tail function provides access to the data frame's tail end (Image 5). This data frame's (Image 6) internal structure is presented succinctly using the str function. The data frame's organization is laid out for inspection here. You'll find a total of 1340 rows and 7 columns. The age column is an integer, and the sex column is a two-level Factor. women and men, The bmi column is a numeric value, the children column is an integer, and the smoker column is a two-level Factor. "no," "yes," Expenses are of type num (Image 6), and the region column is a Factor with four levels ("northeast", "northwest", and "southeast"). While reading the CSV file, if the stringsAsFactors = TRUE option is used, the category variables are transformed into factors. Since machine learning algorithms struggle with character data, we'll be using multiple linear regression instead. The feature detection matrix and the dependent variable detection vector must be identified. Since healthcare costs are of particular relevance, we will use the expenses column as our dependent variable. In the case of the medical expenses column (Image 7), the summary function compiles the minimum, maximum, first and third quartiles, median, and mean. Using the hist function (Image 8), we may see a histogram depicting the distribution of medical costs as a function of frequency. This data demonstrates that the vast majority of occurrences occur at or below the ten thousand mark. Using the table function, we can find out how many instances there are of each unique combination of Factor levels in the region column. In the northeast it says 325 rows, in the northwest it says 326, in the southeast it says 364, and in the southwest it says 325.

```
> head(dataset)
  age    sex  bmi children smoker    region expenses
1  19 female 27.9        0    yes southwest 16884.92
2  18   male 33.8        1     no southeast  1725.55
3  28   male 33.0        3     no southeast  4449.46
4  33   male 22.7        0     no northwest 21984.47
5  32   male 28.9        0     no northwest  3866.86
6  31 female 25.7        0     no southeast  3756.62
```

Image 4: An initial chunk of the data set.

```
> ## Returns the latter parts of the data frame
> tail(dataset)
     age    sex  bmi children smoker    region expenses
1335  18 female 31.9        0     no northeast  2205.98
1336  18 female 36.9        0     no southeast  1629.83
1337  21 female 25.8        0     no southwest  2007.95
1338  61 female 29.1        0    yes northwest 29141.36
1339  65 female 24.7        1    yes northeast 29166.62
1340  20 female 54.0        0     no northwest  1363.46
>
```

Image 5:Late-stage data.

```
> str(dataset)
'data.frame':   1340 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 ...
>
```

Image 6: Animate the data frame's inner workings for the viewer.

```
> summary(dataset$expenses)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1122    4734    9382   13273   16687   63770
>
```

Image 7: Indication of total healthcare costs.

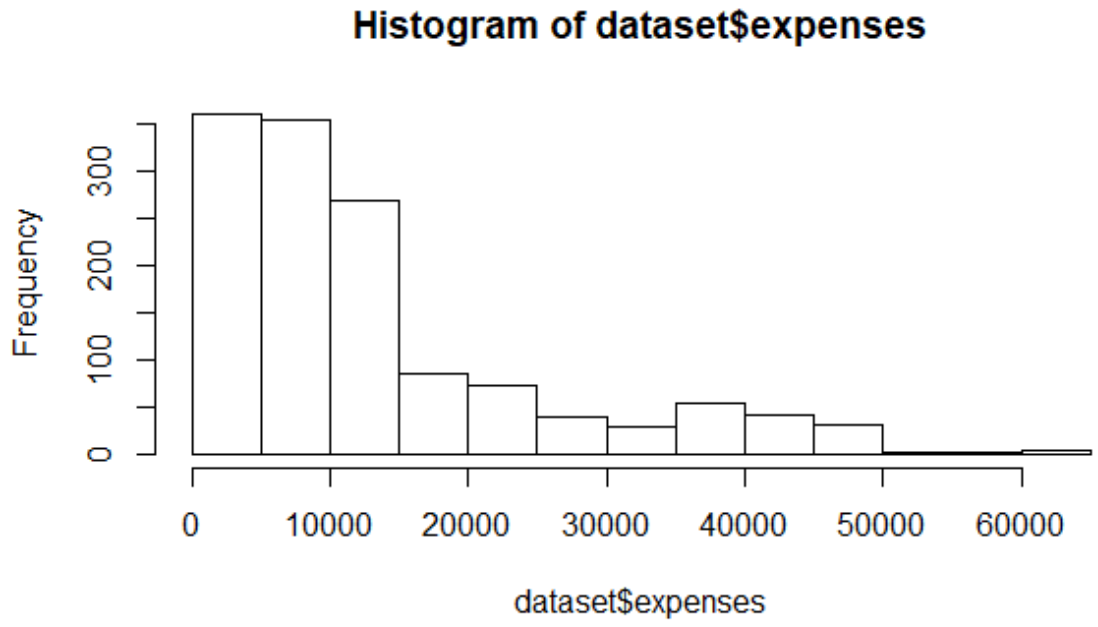


Image 8: Medical expenditures as shown in a histogram.

```
> table(dataset$region)

northeast northwest southeast southwest
      325       326       364       325
>
```

Image 9: Region table.
Examine the relationship among features
Code Snippet
Correlation
cor(dataset$age, dataset$expenses)
cor(dataset$bmi, dataset$expenses)
cor(dataset$children, dataset$expenses)
Correlation matrix
cor(dataset[c("age", "bmi", "children", "expenses")])

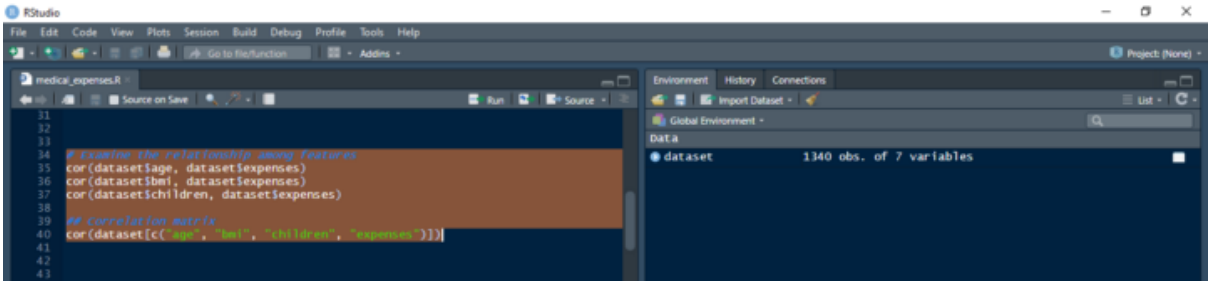


Image 10 : Look at the interdependency of Rstudio's features in Image 10.
Now we must investigate whether or not there are connections between the characteristics. The cor function (Image 10) allows us to investigate the interrelationship of several aspects. Here, we make an effort to decode factors that can be used to estimate future healthcare costs. When the correlation value is 1, it means that the two variables are highly correlated with one another. We have analyzed the cost of living in connection to age, body mass index, and number of children. In that case, with a single function call (Image 11), we have accessed them all. Expenses for medical care are substantially connected with age (0.3009237), while the field identifying the number of dependents (children) is the least correlated (0.0684989). The correlation between BMI and 0.1934809 is moderate. As illustrated in Image 12, we have made a single function call to compare the correlation between all of these variables.

```
> # Examine the relationship among features
> cor(dataset$age, dataset$expenses)
[1] 0.3009237
> cor(dataset$bmi, dataset$expenses)
[1] 0.1934809
> cor(dataset$children, dataset$expenses)
[1] 0.0684989
>
```

Image 11: A comparison of the expenses by age, BMI, and number of children as separate columns.

```
> ## Correlation matrix
> cor(dataset[c("age", "bmi", "children", "expenses")])
                age        bmi   children  expenses
age      1.00000000 0.10328097 0.04319186 0.3009237
bmi      0.10328097 1.00000000 0.01004298 0.1934809
children 0.04319186 0.01004298 1.00000000 0.0684989
expenses 0.30092369 0.19348089 0.06849890 1.0000000
>
```

Image 12: A scatterplot showing the link between the various categories (age, body mass index, children, and expenditures).

Visualize the relationship among features
Code Snippet
Scatterplot matrix
pairs(dataset[c("age", "bmi", "children", "expenses")])
More informative Scatterplot matrix using psych package
install.packages('psych')
library(psych)
pairs.panels(dataset[c("age", "bmi", "children", "expenses")])

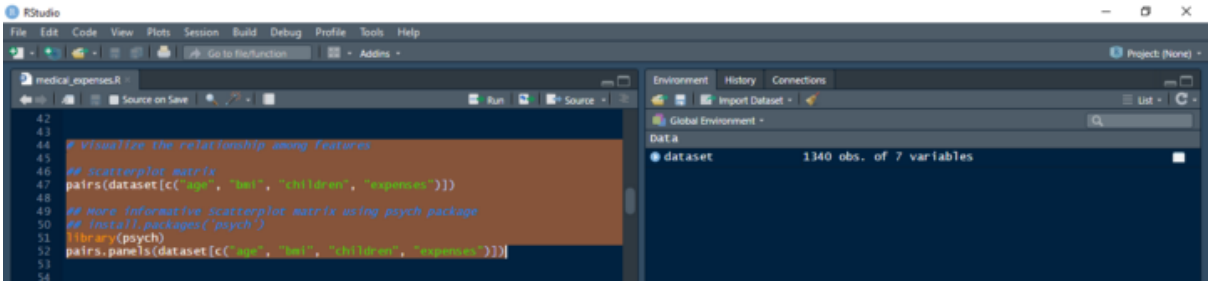


Image 13: Display the interdependencies of features in Rstudio.
With the pairs function, we may see a scatterplot of the features age, body mass index, and number of children (Image 14). Image 15 is an improved diagram made possible by the "pairs.panels" function in the psych package. Each feature's distribution, including whether it is symmetrical, asymmetrical, or skewed, is displayed along the diagonal. There appears to be a normal distribution of age and body mass index (BMI) (Image 15). The distribution of both children and costs is highly skewed to the right, so that the median is close to the maximum. In the top part of Image 15, you can see how these characteristics are related to one another. The other characteristics are less strongly connected with age than the cost does. Therefore, age is a crucial factor. BMI has a weaker relationship with other characteristics and also with costs. Having kids has a smaller effect on the cost of a home and other aspects. The ellipse of significance displaying the relationship between these characteristics is depicted in Image 15's bottom half. The further you stretch it, the more of a connection you'll see. The absence of any connection holds true if and only if the shape is a perfect circle. The line between BMI

and kids looks to be the weakest one, since it's nearly a circle. Since the elliptical expansion, there has been a correlation between age and the number of offspring one has. Longest ellipse segment is between age and cost, indicating the importance of age in cost forecasting; loess curve for cost vs. age is linear (red line in Image 15).

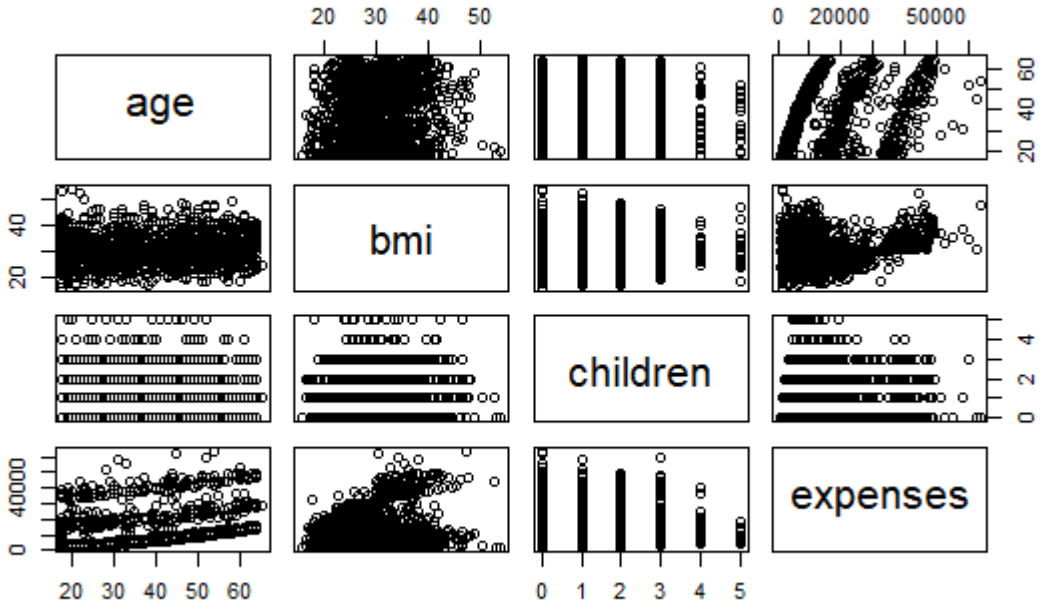


Image_14: Age, body mass index, and number of children can be seen as a scatterplot in a matrix.

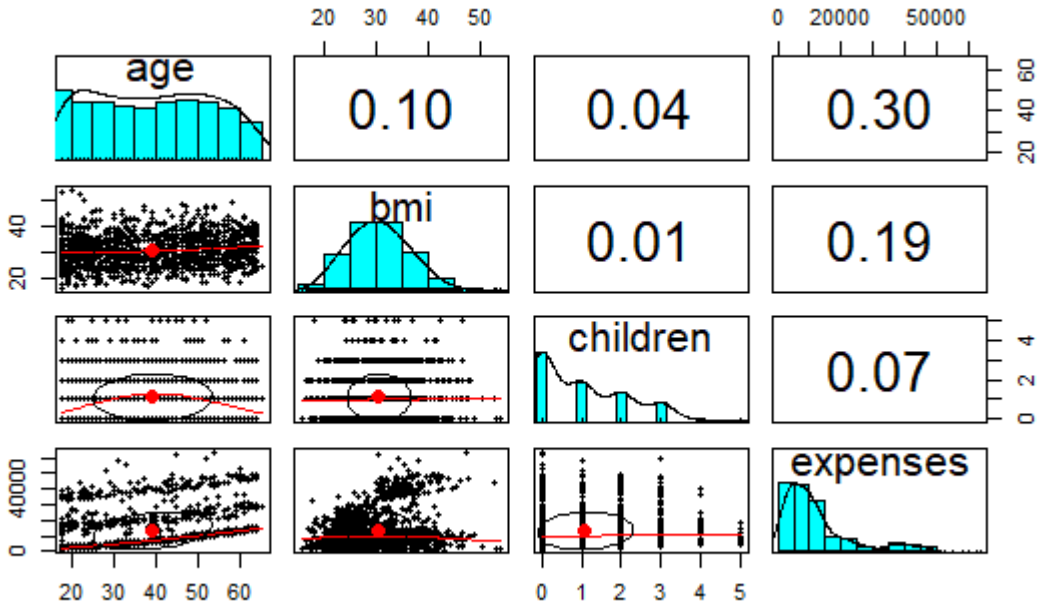


Image_15: Visualize the scatterplot matrix of the features age, bmi, children with enriched diagrams using ‘psych’ package

Splitting the dataset into the training set and the test set

Code snippet

```
# Splitting the dataset into training set and test set
## install.packages('caTools')
library(caTools)
set.seed(123)
## Obtain the training index
training_index <- sample(seq_len(nrow(dataset)), size = floor(0.75 * nrow(dataset)))
## Partition the data
training_set <- dataset[training_index, ]
test_set <- dataset[-training_index, ]
```

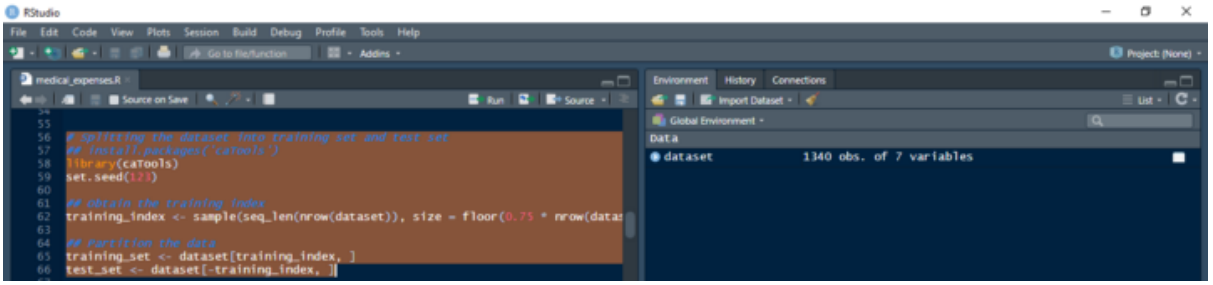


Image 16: Decomposing the data set into a test set and a training set in Rstudio

The model needs to be constructed on one dataset and then tested on another, slightly modified dataset. That means we'll need to buy two sets. We have two data sets: one to train and evaluate the ML model on, and another to put it to the test. No significant deviation in performance from the training set is to be expected on the test set. Which is to say, machine learning models have learned the relationships well enough to be flexible. To replicate the results, we set a random state using a seed. In order to get a representative sample of the rows in the dataset (Image 16), we use the sample function to pick a subset at random. We take 75% of the total number of rows in the dataset and use those indexes to construct the training set. Next, we use these indices to split the dataset into a 75% training set (Image 17) and a 25% test set (Image 18).

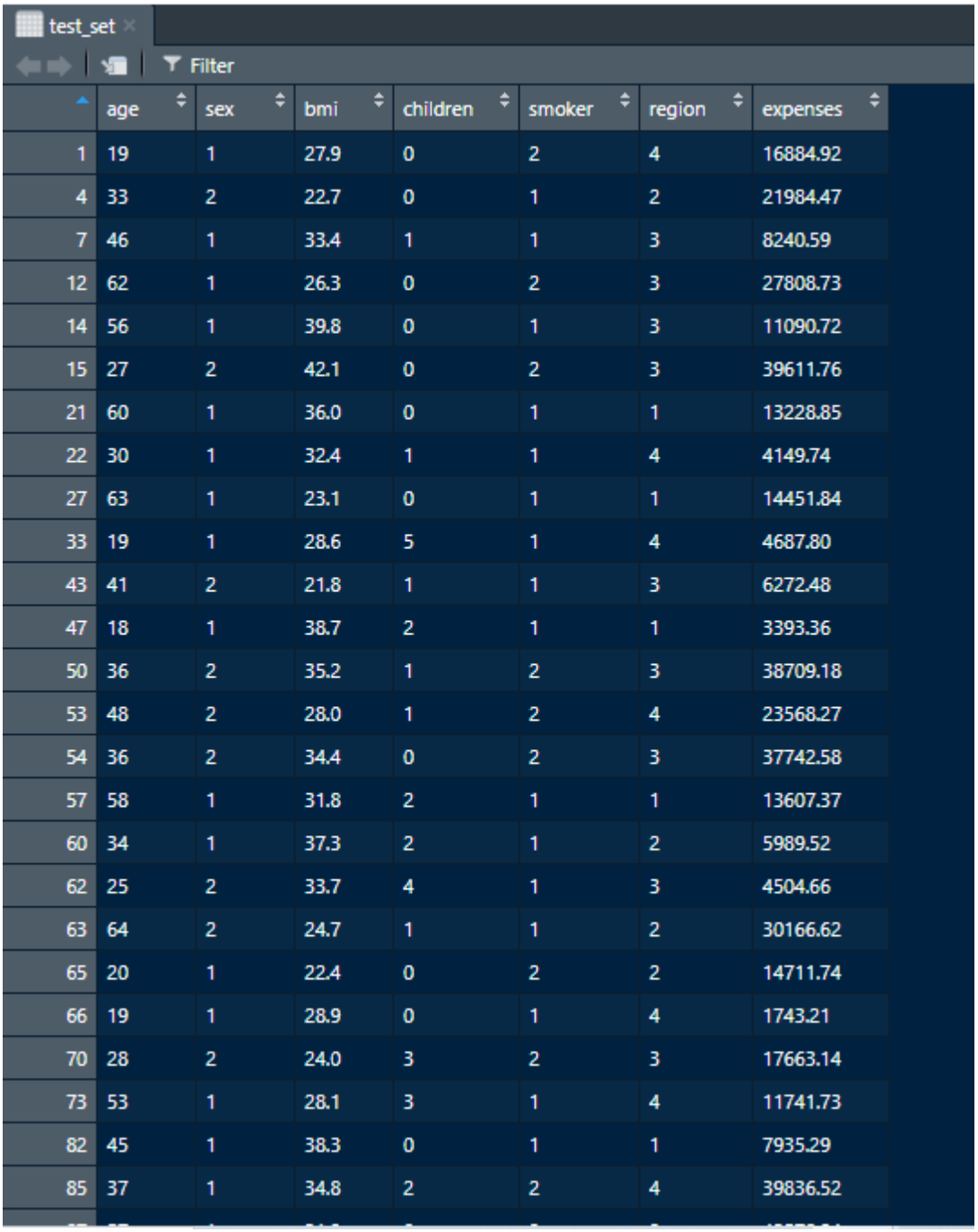
RStudio Source Editor

training_set

Filter

	age	sex	bmi	children	smoker	region	expenses
415	19	1	35.2	0	1	2	2134.90
463	62	1	38.1	2	1	1	15230.32
179	46	1	28.9	2	1	4	8823.28
526	18	1	33.9	0	1	3	11482.63
195	18	2	34.4	0	1	3	1137.47
938	39	1	24.2	5	1	2	8965.80
1142	41	1	32.6	3	1	4	7954.52
1323	62	2	38.8	0	1	3	12981.35
1253	20	2	27.3	0	2	4	16232.85
1268	24	2	31.1	0	2	1	34254.05
1038	45	1	30.5	1	2	2	39725.52
665	64	1	23.0	0	2	3	27037.91
602	51	2	31.6	0	1	2	9174.14
709	31	1	30.5	3	1	1	6113.23
1011	48	1	22.8	0	1	4	8269.04
1115	23	2	24.5	0	1	1	2396.10
953	30	1	28.4	1	1	2	4527.18
348	46	2	33.3	1	1	1	8334.46
1017	19	1	24.6	1	1	2	2709.24
840	59	1	31.4	0	1	2	12622.18
26	59	1	27.7	3	1	3	14001.13
519	35	1	31.0	1	1	4	5240.77
211	20	2	33.0	1	1	4	1980.07
932	39	1	32.5	1	1	4	6238.30
593	20	2	31.1	2	1	3	2566.47

Image 17: Training data partitioned with a 75% success rate.



test_set

Filter

	age	sex	bmi	children	smoker	region	expenses
1	19	1	27.9	0	2	4	16884.92
4	33	2	22.7	0	1	2	21984.47
7	46	1	33.4	1	1	3	8240.59
12	62	1	26.3	0	2	3	27808.73
14	56	1	39.8	0	1	3	11090.72
15	27	2	42.1	0	2	3	39611.76
21	60	1	36.0	0	1	1	13228.85
22	30	1	32.4	1	1	4	4149.74
27	63	1	23.1	0	1	1	14451.84
33	19	1	28.6	5	1	4	4687.80
43	41	2	21.8	1	1	3	6272.48
47	18	1	38.7	2	1	1	3393.36
50	36	2	35.2	1	2	3	38709.18
53	48	2	28.0	1	2	4	23568.27
54	36	2	34.4	0	2	3	37742.58
57	58	1	31.8	2	1	1	13607.37
60	34	1	37.3	2	1	2	5989.52
62	25	2	33.7	4	1	3	4504.66
63	64	2	24.7	1	1	2	30166.62
65	20	1	22.4	0	2	2	14711.74
66	19	1	28.9	0	1	4	1743.21
70	28	2	24.0	3	2	3	17663.14
73	53	1	28.1	3	1	4	11741.73
82	45	1	38.3	0	1	1	7935.29
85	37	1	34.8	2	2	4	39836.52

Image 18: After dividing into quarters, the test set is worth 25%.
Data training using the training data (Fit Multiple Linear Regression to the Training set).

Code Snippet.

```
# Training the data on the training set
## Fit Multiple Linear Regression to the training set
initial_model <- lm(expenses ~ ., data = training_set)
```

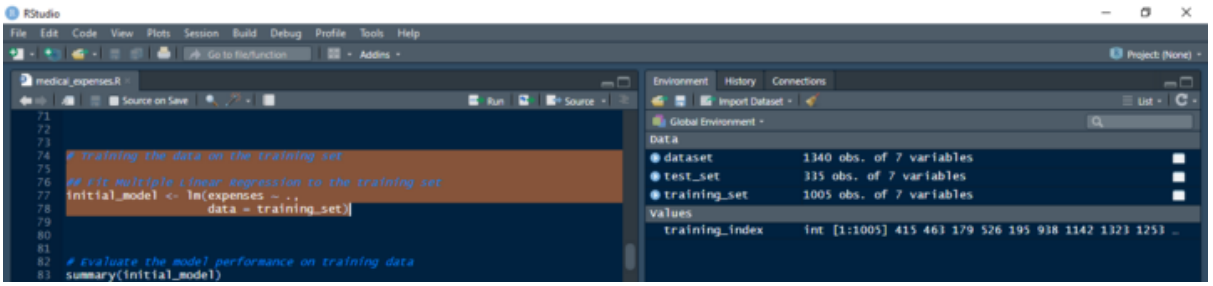


Image 19: Rstudio training data on a data set.

To do this, we use the lm function to apply Multiple Linear Regression to the data in the training set (Image 19). To calculate the total cost of healthcare, we use a linear combination of the independent variables denoted by "expenses. In the first step, we use this function to train our model on the training data. Image 20 displays the final model that was created.

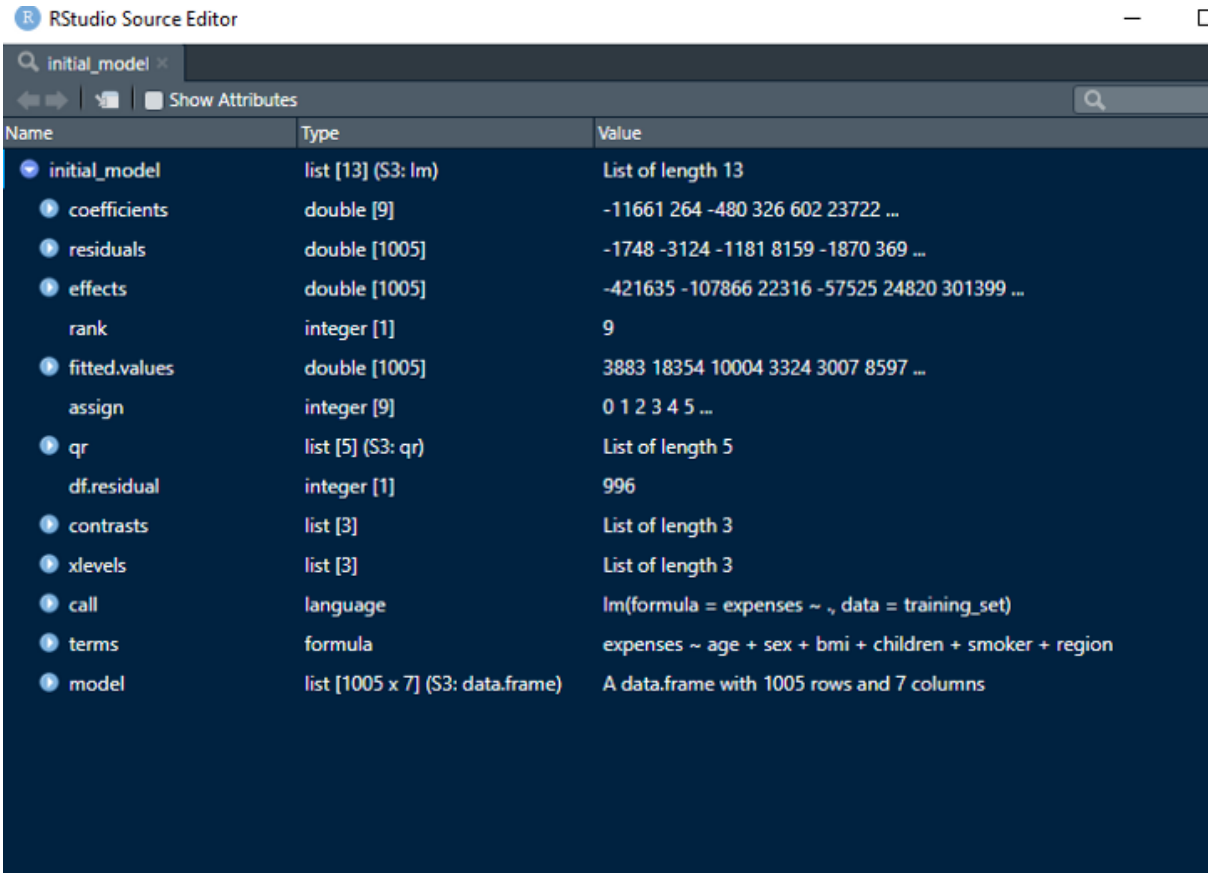


Image 20: This is the trained model following its initialization.

Take a look at how well the model does on the training data.

Code Snippet

```
# Evaluate the model performance on training data
summary(initial_model)
# Interpret on residuals, coefficients, statistical significance of predictors
# & overall model performance — Adjusted R-squared
```

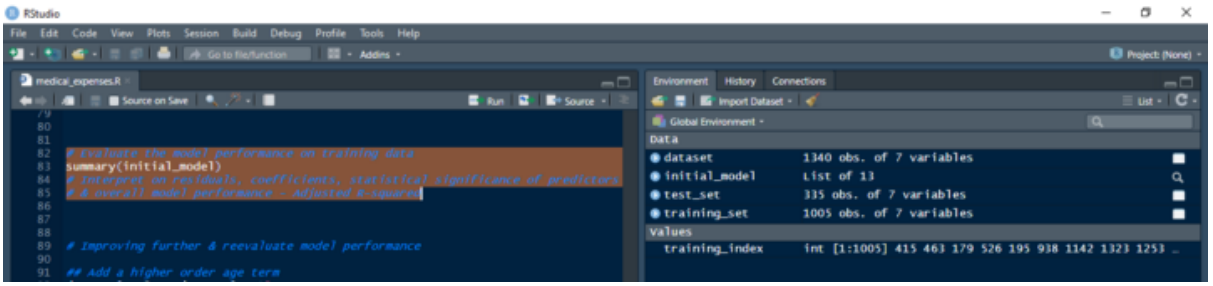


Image 21: Utilizing training data, assess the model's performance in Rstudio.
Using the summary function, we were able to get the initial model's summary in this case. The summary includes four crucial components: adjusted R-squared, residuals, coefficients, and the statistical significance of predictors (Image 22).

Errors are residuals. The highest inaccuracy under an underestimating situation is 30029. This indicates that the actual cost exceeded the estimated cost. The greatest mistake in a case of overestimation is 11439. This indicates that we have overcharged for our services. First 25% cut point is represented by the first quartile, while third quartile is first 75% cut point. An acceptable range is between the first quartile's overestimation of 3060 and the third quartile's underestimating of 1445.

We can see that R does produce dummy variables and hasn't fallen victim to the dummy variable trap when we look at the coefficients section. To prevent unnecessary dependencies, one dummy variable is automatically removed. We have distinct data for each coefficient, including the standard error, t-value, p-value, and significance level from the linear regression equation. We must determine whether we have enough statistically significant variables for the model. Based on the threshold value, this. The 5% threshold is typically a decent one to utilize. The independent variable will be highly statistically significant if the P-value is less than 5%. It will be higher than 5% the more. It won't have as much statistical impact. It is easier to understand coefficients by looking at the final column of stars. Age, BMI, children, and smoking are the statistically significant predictors of expense out of all the independent factors. Certain regions are also effective predictors.

How much of the overall variability of the dependent variable (medical expenses) can be explained by the model is indicated by the adjusted R squared (selected independent variables). Here, we have the strong model of 0.7352, which we have obtained. Using this metric, we may determine whether the model has improved or not.

```
> summary(initial_model)

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

Residuals:
   Min     1Q Median     3Q    Max
-11439  -3060  -1098   1445  30029

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -11661.11    1162.27 -10.033  < 2e-16 ***
age            264.40      14.11  18.742  < 2e-16 ***
sex2          -479.58     392.84  -1.221 0.222454
bmi            325.93      33.03   9.866  < 2e-16 ***
children       602.13     164.80   3.654 0.000272 ***
smoker2      23721.98     489.59  48.453  < 2e-16 ***
region2       -951.78     565.75  -1.682 0.092817 .
region3       -823.22     557.37  -1.477 0.140000
region4      -1120.68     564.67  -1.985 0.047456 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6189 on 996 degrees of freedom
Multiple R-squared:  0.7373,    Adjusted R-squared:  0.7352
F-statistic: 349.4 on 8 and 996 DF,  p-value: < 2.2e-16
```

Image 22: Summary of the initial model's performance on training data is shown in image 22. Reassessing model performance and making more improvements.

Code Snippet:

```
## Add a higher order age term
dataset$age2 <- dataset$age^2
## summary of the BMI column
summary(dataset$bmi)
## Add an indicator for BMI
dataset$bmi30 <- ifelse(dataset$bmi >= 30, 1, 0)
## Partition the data again with the additional columns but using the same index
training_set_new <- dataset[training_index, ]
test_set_new <- dataset[-training_index, ]
## Create the final model
final_model <- lm(expenses ~ sex + bmi + children + region + age2 + bmi30*smoker,data = training_set_new)
# Evaluate the model performance on the new training data
summary(final_model)
# Interpret on residuals, coefficients, statistical significance of predictors
# & overall model performance — Adjusted R-squared
```

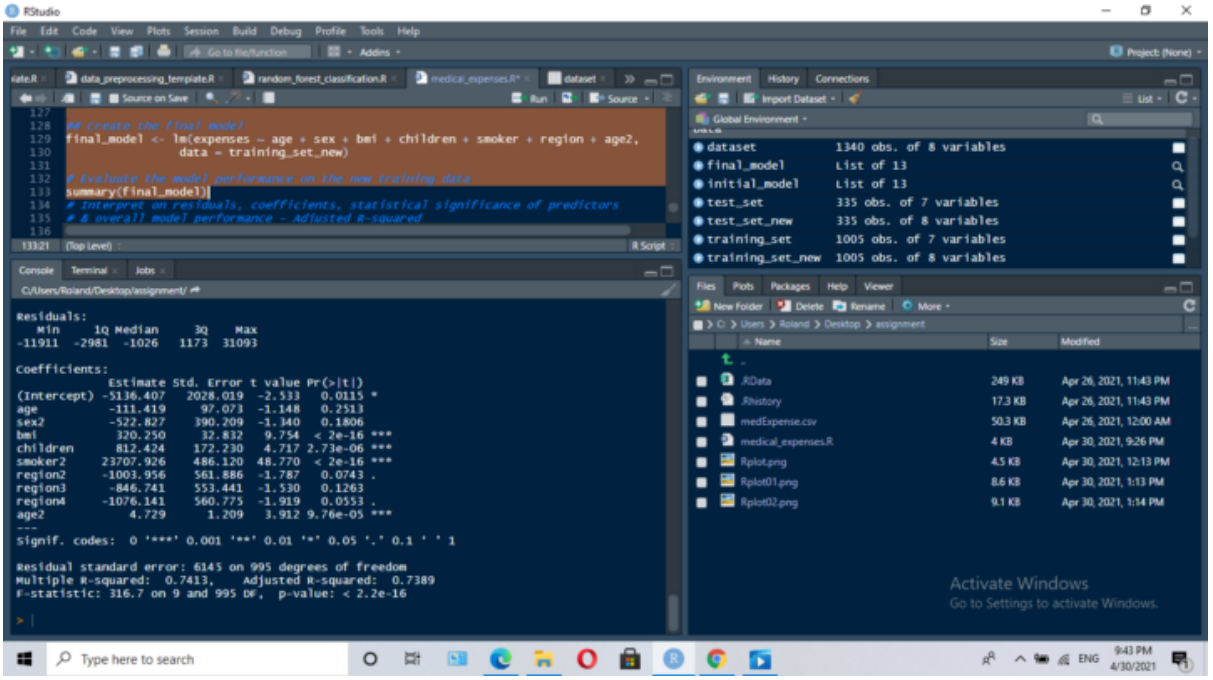


Image 23: Utilize Rstudio to assess the model's performance with training data that includes an additional age 2 columns.

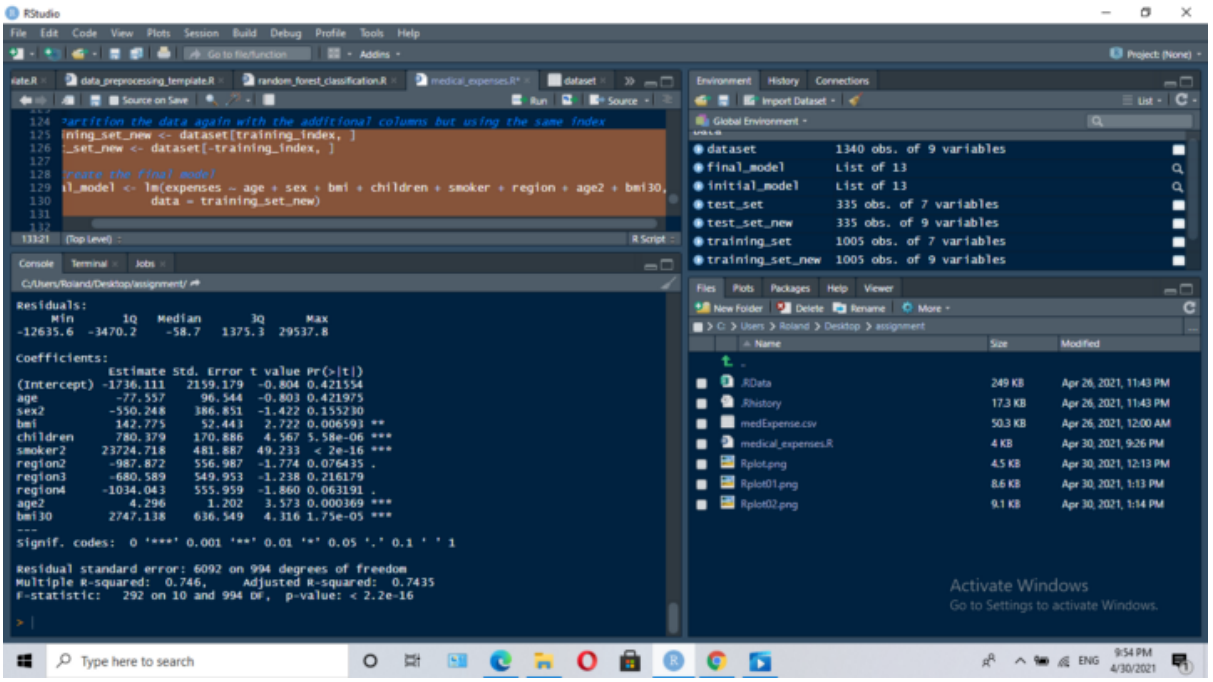


Image 24: In Rstudio, assess the model's performance on the training set of data by adding an additional age-2 and bmi indicator column.

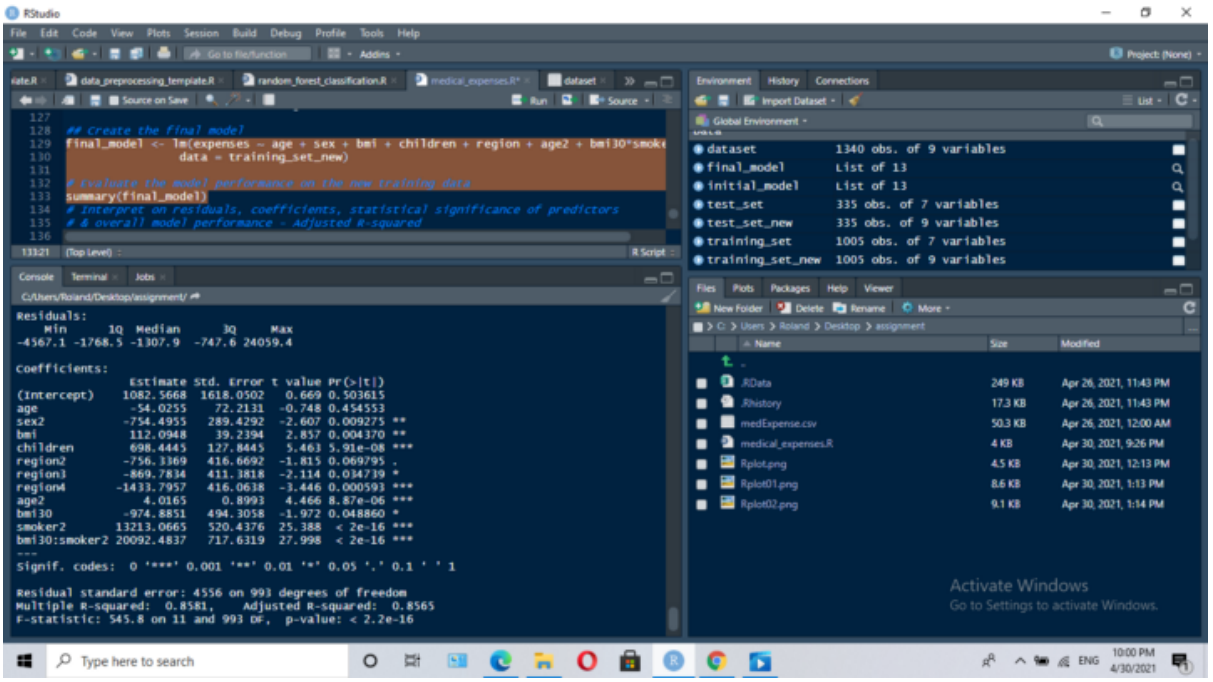


Image 25: In Rstudio, assess the model's performance on the training set of data by adding a second age column and combining it with the smoking and BMI indicator columns.

We can manipulate our basic model in order to improve it. One option is to add interactivity columns or features. We then present newly trained data with these additional columns, revising our original model and its performance. By adding a second column with the name age 2, the age, which has a linear relationship, is handled as a non-linear relationship. The model's performance was enhanced by the addition of this new column, which resulted in an increase in the adjusted R squared value to 0.7389 (Images 23 and 26).

Furthermore, we use the Bmi as a healthy or unhealthy component rather than as a continuous value. To determine if a bmi value is healthy or not, we use a value of 30, which is near to the mean. The model's performance was enhanced by the addition of this new column, which resulted in an increase in the adjusted R squared value to 0.7435 (Images 24 and 28). Then, the two highly statistically significant columns with the lowest P-value were concatenated. The adjusted R squared value improved to 0.8565 with the addition of this new column, enhancing model performance and providing us with the ideal model (Image 25 & 29).

Using the summary function, we were able to get the final model's summary in this case. The summary includes three crucial components: the adjusted R-squared, residuals, coefficients, and the statistical significance of predictors (Image 29).

Relative to residuals, The largest mistake in a case of underestimation is 24059.4. This indicates that the actual expense exceeded the anticipated expense. The maximum mistake under an overestimation is 4567.1. This indicates that we have overcharged for our services. It is acceptable that the first quartile overestimates by 1768.5 and the third quartile overestimates by 747.6.

We must determine whether we have enough statistically significant variables for the model. Based on the threshold value, this. The 5% threshold is typically a decent one to utilize. The independent variable will be highly statistically significant if the P-value is less than 5%. It will be higher than 5% the more. It won't have as much statistical impact. It is easier to understand coefficients by looking at the final column of stars. The combination of the bmi indicator and smoker, which has the lowest P-value among all the independent factors, is the statistically best predictor. Some of the regions and higher-order age are also excellent predictors. Each of the sex and bmi columns has some bearing on how much money will be spent.

How much of the overall variability of the dependent variable (medical expenses) can be explained by the model is indicated by the adjusted R squared (selected independent variables). The model we got here, with a performance of 0.8565, is the best one we have so far. In comparison to the initial model, this value is significantly higher.

```
Residuals:
    Min      1Q  Median      3Q     Max
-11911   -2981   -1026    1173   31093

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5136.407   2028.019  -2.533   0.0115 *
age          -111.419     97.073  -1.148   0.2513
sex2         -522.827    390.209  -1.340   0.1806
bmi           320.250     32.832   9.754  < 2e-16 ***
children      812.424    172.230   4.717 2.73e-06 ***
smoker2     23707.926    486.120  48.770  < 2e-16 ***
region2     -1003.956    561.886  -1.787   0.0743 .
region3      -846.741    553.441  -1.530   0.1263
region4     -1076.141    560.775  -1.919   0.0553 .
age2            4.729      1.209   3.912 9.76e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6145 on 995 degrees of freedom
Multiple R-squared:  0.7413,    Adjusted R-squared:  0.7389
F-statistic: 316.7 on 9 and 995 DF,  p-value: < 2.2e-16
```

Image 28: Summary of the model's training data performance with an additional age 2 columns.

```
> summary(dataset$bmi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  16.00   26.27   30.40   30.68   34.70   54.00
>
> ## Add an indicator for BMI
> dataset$bmi30 <- ifelse(dataset$bmi >= 30, 1, 0)
>
```

Image 27: Summary of the BMI column in image 27.

```
Residuals:
     Min       1Q   Median       3Q      Max
-12635.6  -3470.2    -58.7   1375.3  29537.8

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1736.111   2159.179  -0.804 0.421554
age           -77.557     96.544  -0.803 0.421975
sex2         -550.248    386.851  -1.422 0.155230
bmi           142.775     52.443   2.722 0.006593 **
children      780.379    170.886   4.567 5.58e-06 ***
smoker2     23724.718    481.887  49.233  < 2e-16 ***
region2      -987.872    556.987  -1.774 0.076435 .
region3      -680.589    549.953  -1.238 0.216179
region4     -1034.043    555.959  -1.860 0.063191 .
age2            4.296      1.202   3.573 0.000369 ***
bmi30        2747.138    636.549   4.316 1.75e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6092 on 994 degrees of freedom
Multiple R-squared:  0.746,     Adjusted R-squared:  0.7435
F-statistic:   292 on 10 and 994 DF,  p-value: < 2.2e-16
```

Image 28: A overview of the model's performance using the training data, with an additional column for the age-2 and BMI indicators.

```
Residuals:
    Min      1Q  Median      3Q     Max
-4567.1 -1768.5 -1307.9  -747.6 24059.4

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    1082.5668  1618.0502   0.669 0.503615
age             -54.0255    72.2131  -0.748 0.454553
sex2           -754.4955   289.4292  -2.607 0.009275 **
bmi             112.0948    39.2394   2.857 0.004370 **
children        698.4445   127.8445   5.463 5.91e-08 ***
region2        -756.3369   416.6692  -1.815 0.069795 .
region3        -869.7834   411.3818  -2.114 0.034739 *
region4       -1433.7957   416.0638  -3.446 0.000593 ***
age2              4.0165     0.8993   4.466 8.87e-06 ***
bmi30          -974.8851   494.3058  -1.972 0.048860 *
smoker2       13213.0665   520.4376  25.388  < 2e-16 ***
bmi30:smoker2 20092.4837   717.6319  27.998  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4556 on 993 degrees of freedom
Multiple R-squared:  0.8581,    Adjusted R-squared:  0.8565
F-statistic: 545.8 on 11 and 993 DF,  p-value: < 2.2e-16
```

Image 29: A summary of the final model's performance using training data that included an extra age-related column and a combination of a smoker and bmi indicator column.
Using the upgraded model to forecast the outcomes of the test set.

The snippet of code.

```
# Predicting test set outcomes using the enhanced

modelmedicalExpensesPredicted = predict(final_model, newdata = test_set_new)
cor(medicalExpensesPredicted, test_set_new$expenses)
plot(medicalExpensesPredicted, test_set_new$expenses)
abline(a = 0, b = 1, col = “red”, lwd = 3, lty = 2)
```

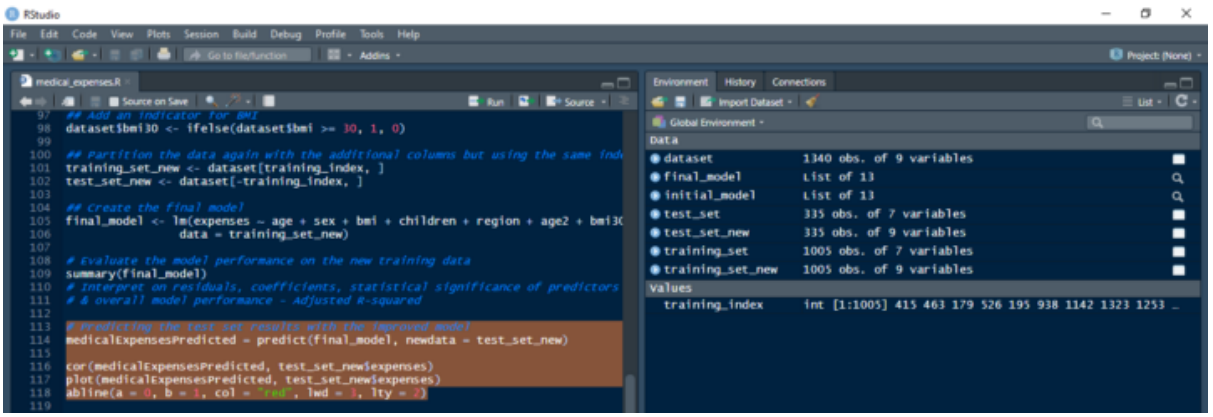


Image 30: Using the modified model in Rstudio to predict the test set results.

Here, we use the predict function of the upgraded model to generate predictions on the fresh test data (test set new) (Image 30). By using the cor function to evaluate the correlation between the predicted medical costs and the actual medical costs in the test set, we can see that they are highly connected and have a high degree of accuracy (Image 31). Here, in Image 32, we've plotted the expected versus real medical costs. There aren't many outliers, but the plot shows that in the majority of the instances, the actual and projected results match up. Download the R script used in the aforementioned example here.

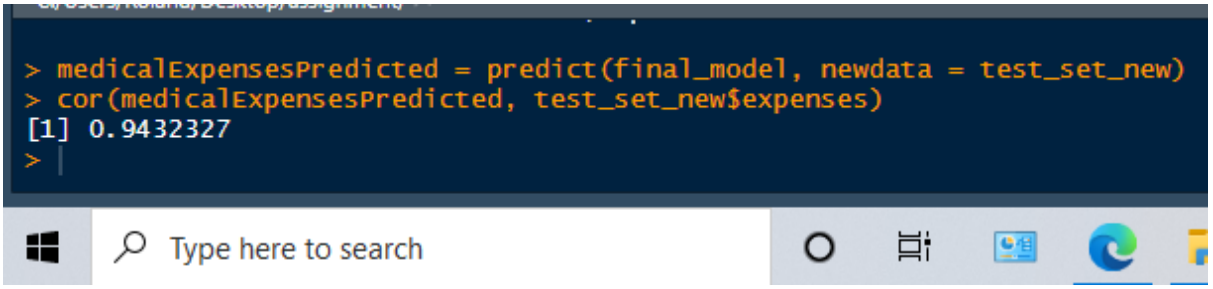


Image 31: Test set predictions using the upgraded model.

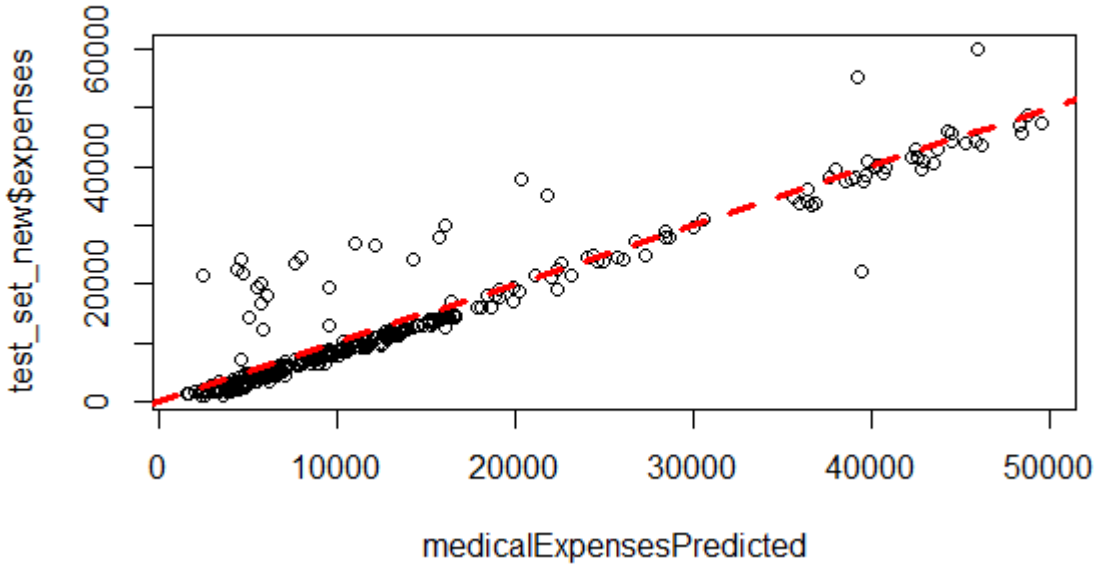


Image 32: Compare the anticipated and actual medical costs in Image 32.
You may have recognized by this point that using a tool like RStudio makes data analysis not all that difficult. The knowledge you learned from this guide can be used in any other area of interest.