Given the medical dataset that consist of age, sex, BMI(body mass index), children, smoker, and region as features, and charges as target variable, we have to apply linear regression to predict charges.

The hypothesis function looks like

$$h_\theta(x_i)=\theta_0+\theta_1 age+\theta_2 sex+\theta_3 bmi+\theta_4 children+\theta_5 smoker+\theta_6 region$$

Tasks :

1. **Load the dataset and do exploratory data analysis.**

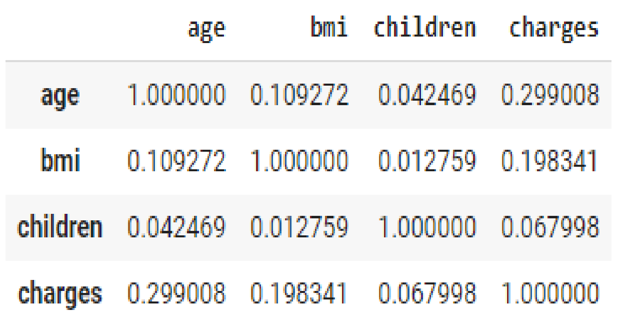
   Exploratory data analysis is done to explore the data . The functions which I used to explore the data are `head()`, `info()`, `describe()`, `unique()`. To check whether there is any null value in the dataset, `df.isna().sum()` function is used .

2. **Plot correlation between different variables and analyze whether there is a correlation between any pairs of variables or not.**

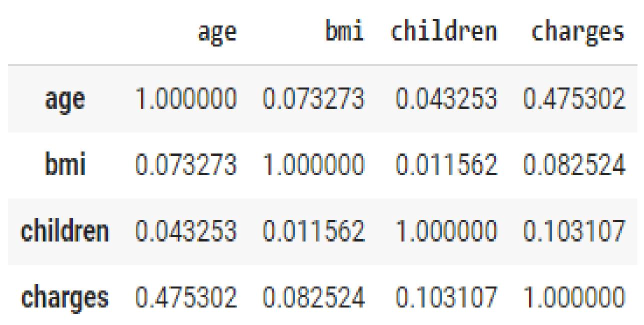
   Pandas `df.corr()` is used to find the pairwise correlation of all columns in the dataframe. The function ignores non-numeric data type columns.

   In the notebook, correlation matrix is created using two methods

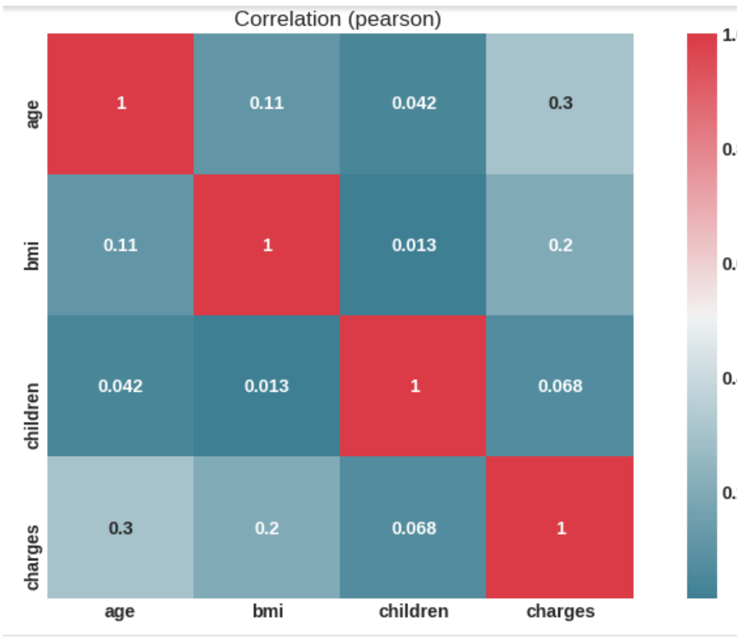
   1. Pearson correlation
   2. Kendall correlation

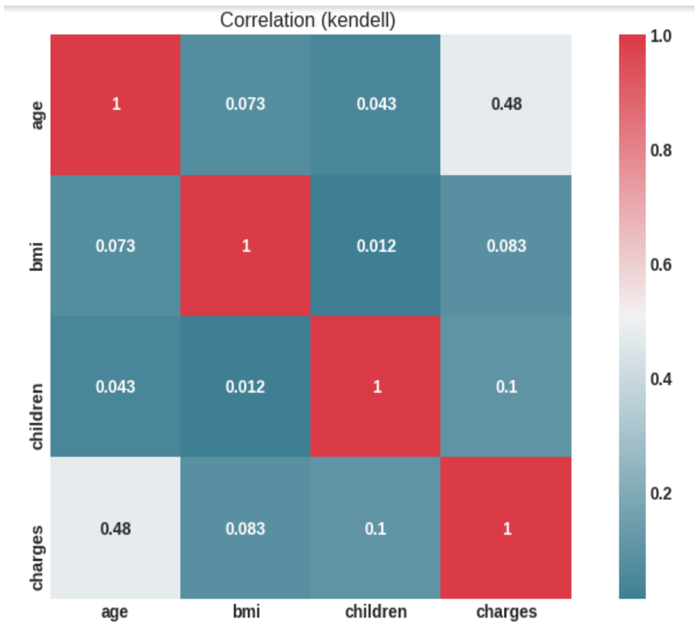


	age	bmi	children	charges
age	1.000000	0.109272	0.042469	0.299008
bmi	0.109272	1.000000	0.012759	0.198341
children	0.042469	0.012759	1.000000	0.067998
charges	0.299008	0.198341	0.067998	1.000000



	age	bmi	children	charges
age	1.000000	0.073273	0.043253	0.475302
bmi	0.073273	1.000000	0.011562	0.082524
children	0.043253	0.011562	1.000000	0.103107
charges	0.475302	0.082524	0.103107	1.000000

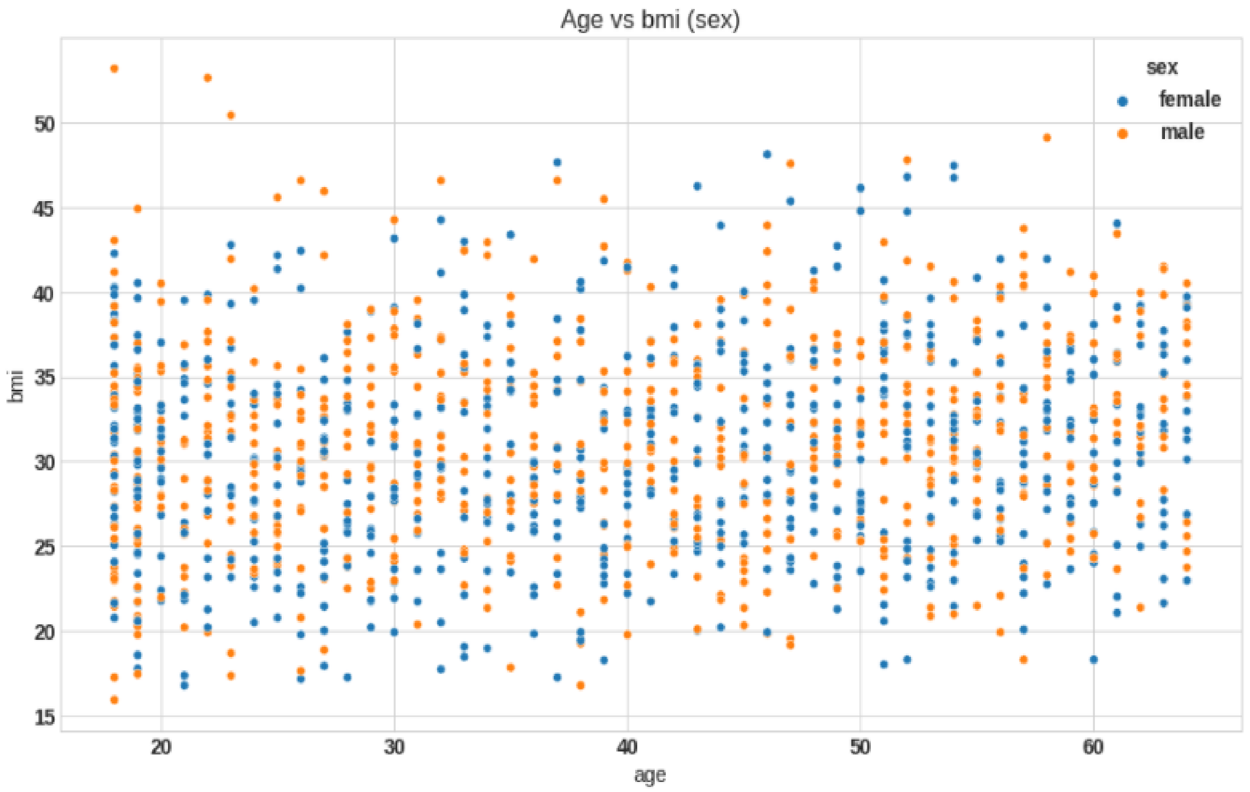


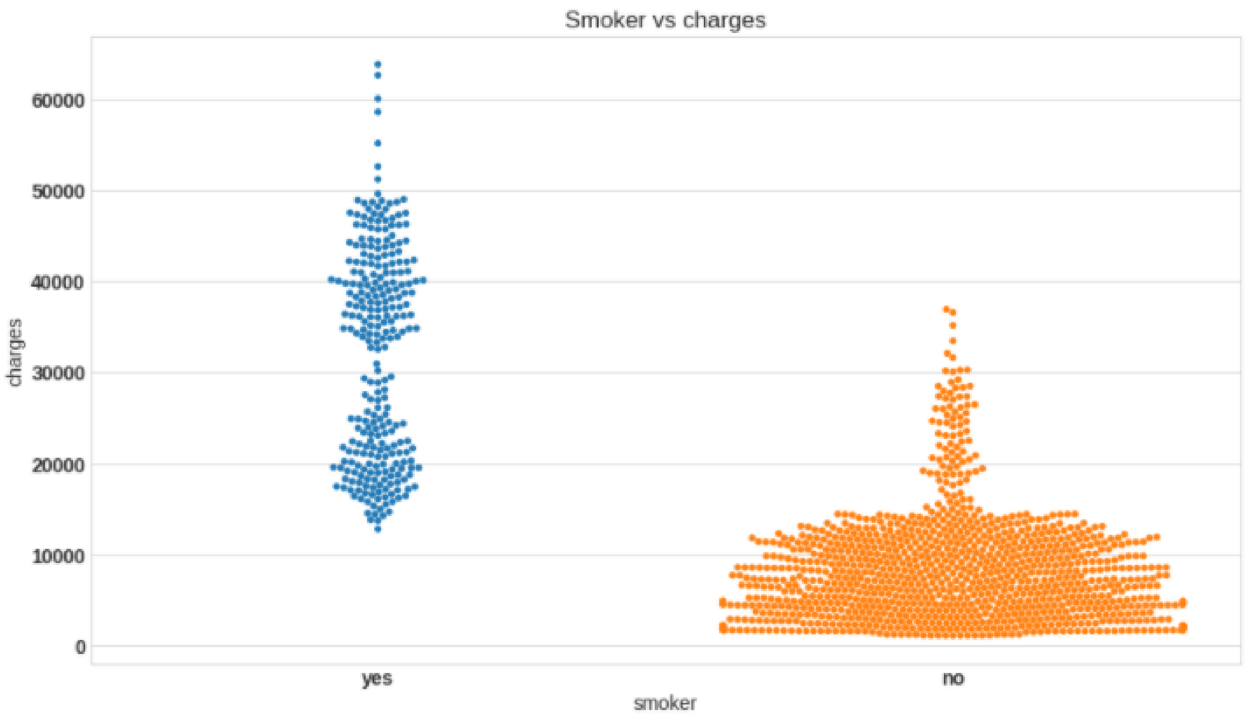


Pearson method

Kendall method

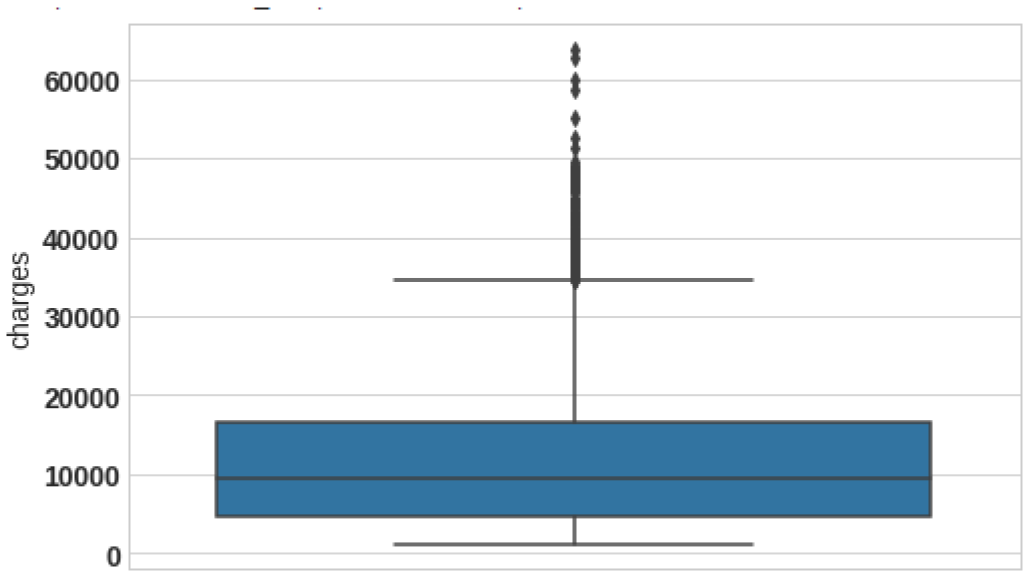
The graphs between pair of variables to get an idea of correlation:


Age vs bmi (sex)
sex
female
male
bmi
age
50
45
40
35
30
25
20
15
20
30
40
50
60


Smoker vs charges
charges
smoker
60000
50000
40000
30000
20000
10000
0
yes
no

3. Plot the distribution of the dependent variable and check for skewness (right or left skewed) in the distribution.

Before checking for skewness, we plot the boxplot model for ‘Charges’ column **.**



From the graph , it is apparent that there are many outliers in the data . Therefore to find those outliers we use z score method ,

```
z = np.abs(stats.zscore(df[['age','children','bmi','charges']]))

threshold = 3

print(np.where(z >threshold))
```

Removing the outliers

```
df= df[(z < threshold).all(axis=1)]
```

The number of samples remaining after removing the outliers are `1309`

We have charges as the dependent variable . To check for skewness we can use ,

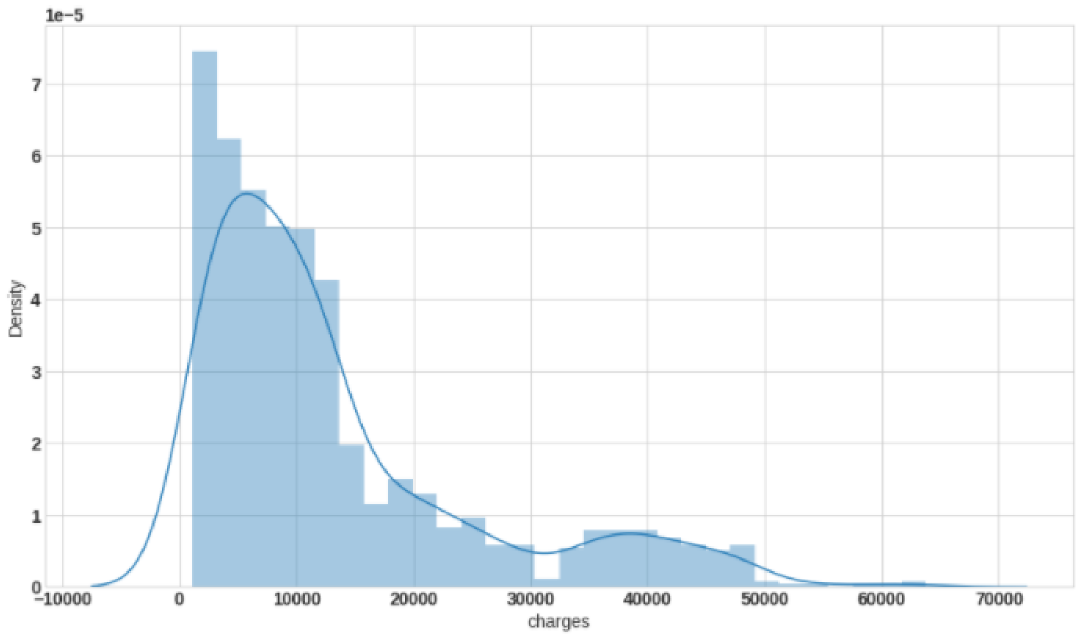
```
df['charges'].skew()
```

The output of this code is

```
Skewness:1.437160
```

It is clear from the value that the data is right skewed .

We can verify that using distplot:



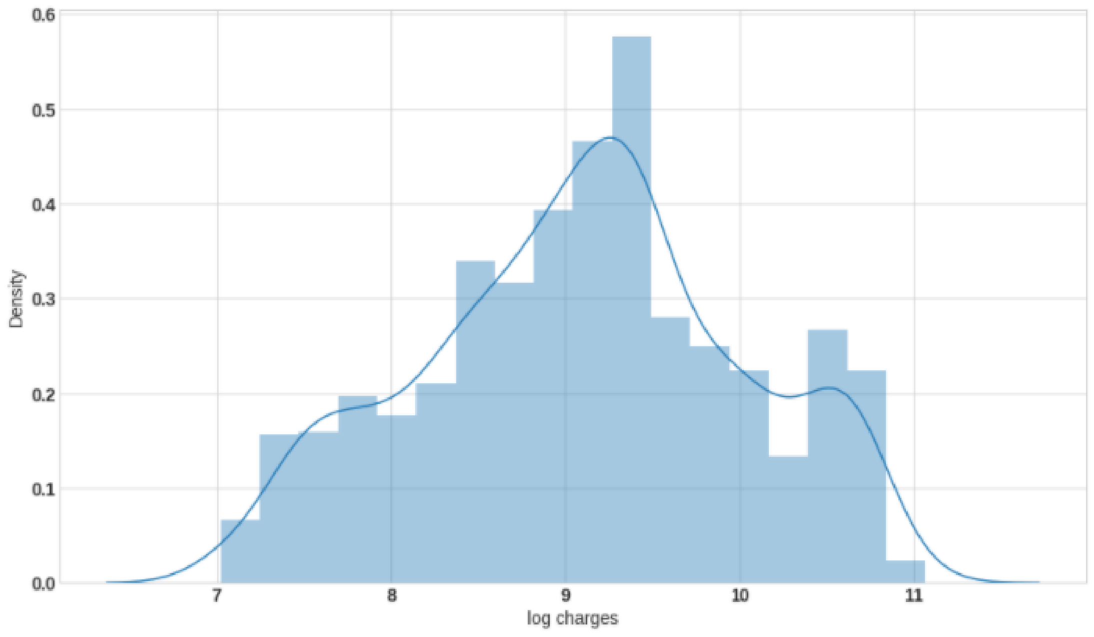
4. Convert this distribution into normal by applying natural log and plot it. (If the distribution is normal then skip this).

The distribution can be converted into normal distribution using natural log

of the values in charges column

```
df['log charges'] = np.log(df['charges'])
```

Plotting the distplot now :

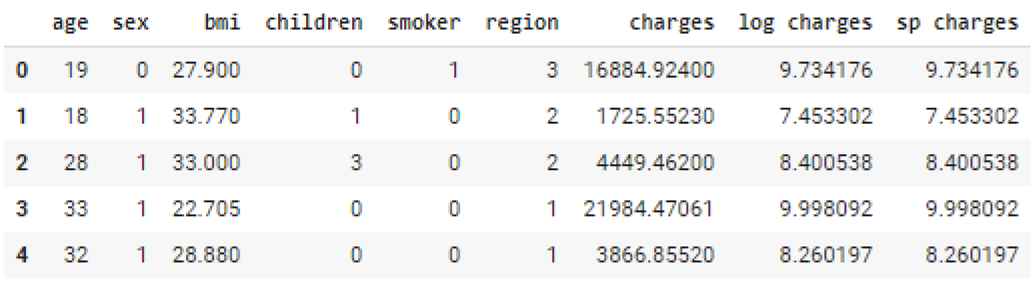


We can do the same process using scipy boxcox function.

The resultant plot will be same for both methods

5. Convert categorical data into numbers. (You may choose one hot encoding or label encoding for that).

Label encoder is used to convert categorical data into numbers . The code can be seen in the notebook. The dataframe after categorial encoding looks like this:



	age	sex	bmi	children	smoker	region	charges	log charges	sp charges
0	19	0	27.900	0	1	3	16884.92400	9.734176	9.734176
1	18	1	33.770	1	0	2	1725.55230	7.453302	7.453302
2	28	1	33.000	3	0	2	4449.46200	8.400538	8.400538
3	33	1	22.705	0	0	1	21984.47061	9.998092	9.998092
4	32	1	28.880	0	0	1	3866.85520	8.260197	8.260197

Where log charges and sp charges are the column created while converting the distribution to normal

6. Split the data into training and testing sets with ratio 0.3.

```
X_train,X_test,Y_train,Y_test=train_test_split(x,y,test_
size=0.3,random_state=42).
```

It is recommended to use the parameter `random_state=42` to produce the same results across a different run.

X_train has `916` samples and X_test has `393` samples

7. Build a model using linear regression equation θ=$(X^TX)^{-1}X^Ty$. (First add a feature X_0 =1 to the original dataset).

To use the linear regression equation , we need to add a feature X0 which will have values=1 for each data sample. We do this using :

```
X_train_s['X0'] = 1

X_test_s['X0'] = 1
```

Now to apply the equation, we will first convert the dataframe to numpy arrays .

```
X = X_train_s.to_numpy()

Y = Y_train.to_numpy()
```

The code below finds the theta using the linear regression equation

```
a = np.matmul(X_trans, X)

ainv = np.linalg.inv(a)

b = np.matmul(ainv , X_trans)
```

```
theta = np.matmul(b, Y)
```

8. Build a linear regression model using the sklearn library. (No need to add X_0 =1, sklearn will take care of it.)

Using sklearn's `LinearRegression(),` we can easily build a regression model .

```
reg = LinearRegression()

reg.fit(X_train, Y_train)
```

9. Get the parameters of the models you built in step 7 and 8, compare them, and print comparisons in a tabular form.

The below table shows the weight of each feature calculated in step 7 and step 8

	Model (from scratch)	Using sklearn
age	`0.03415366`	`0.03415366`
sex	`-0.07787483`	`-0.07787483`
bmi	`0.01175335`	`0.01175335`
children	`0.10252069`	`0.10252069`
smoker	`1.54601666`	`1.54601666`
region	`-0.04802618`	`-0.04802618`
intercept	`7.08517019`	`7.085170193937947`

From the table , it is clear that the parameters are same for both .

10. Get predictions from both the models

For model built using linear regression equation:

```
y_pred = np.matmul(X_test_s_ar, theta)
```

Using sklearn:

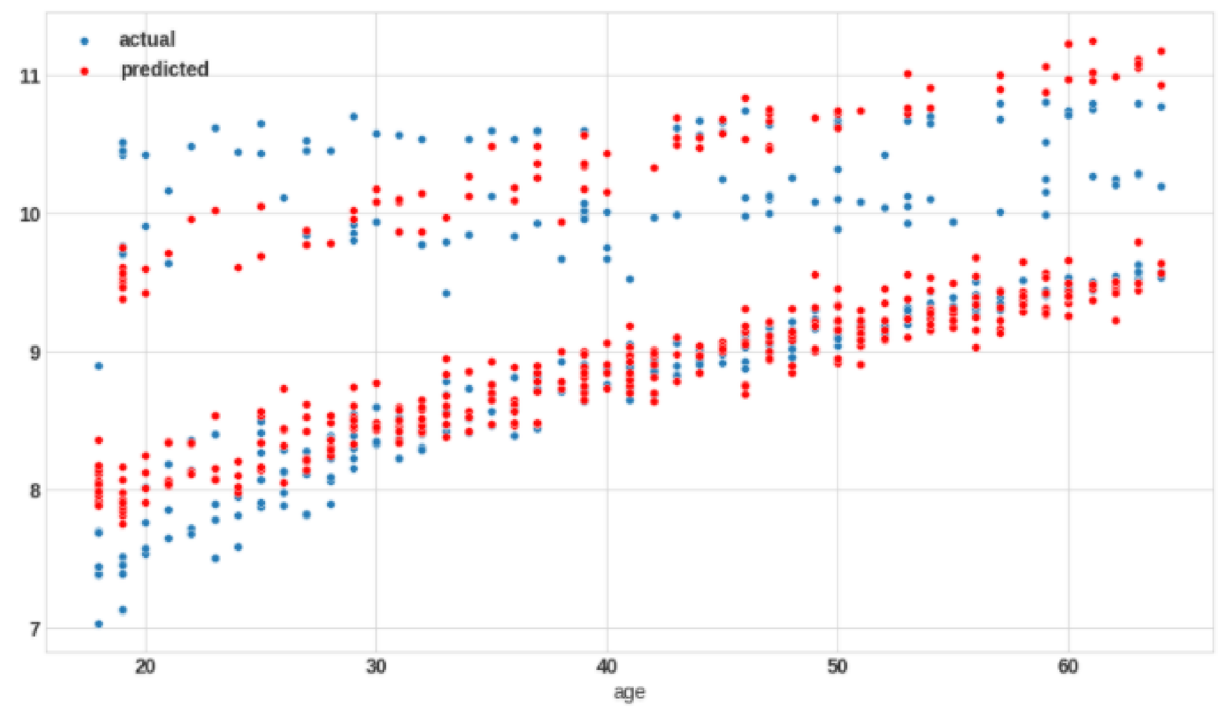
```
y_pred_sk = reg.predict(X_test)
```

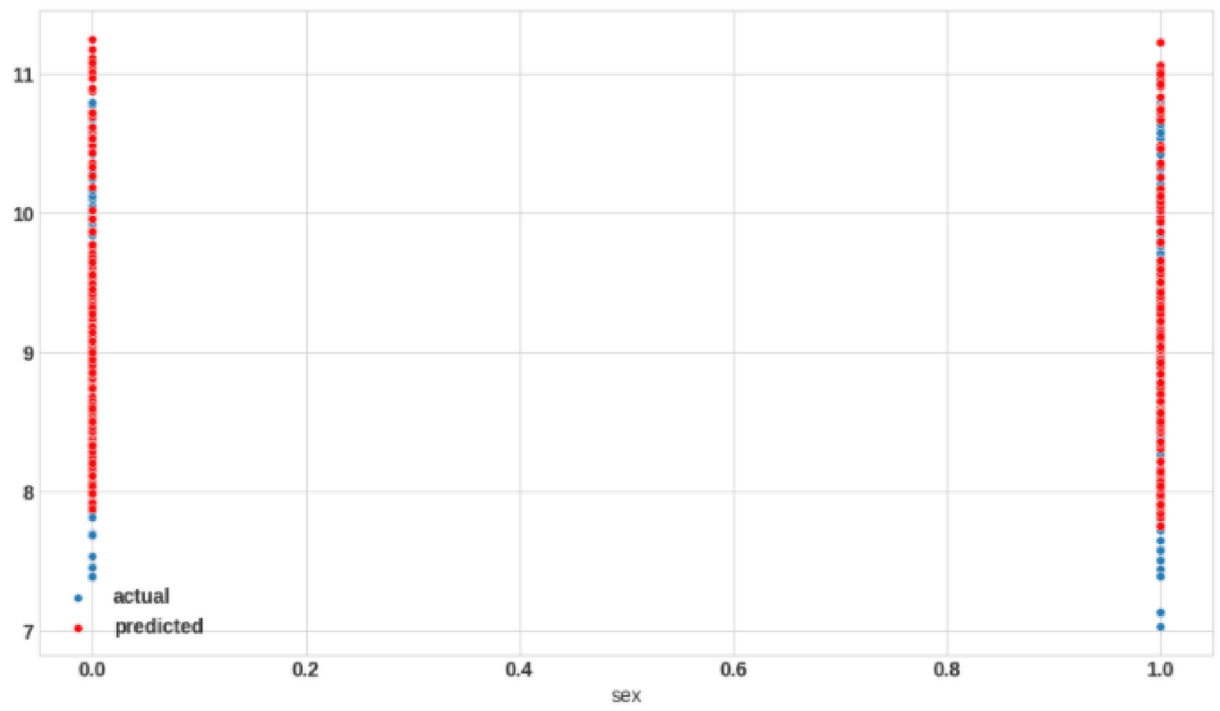
11. Perform evaluation using the MSE of both models

	Model(from scratch)	Using sklearn
MSE	`0.17027210742136664`	`0.1702721074213662`

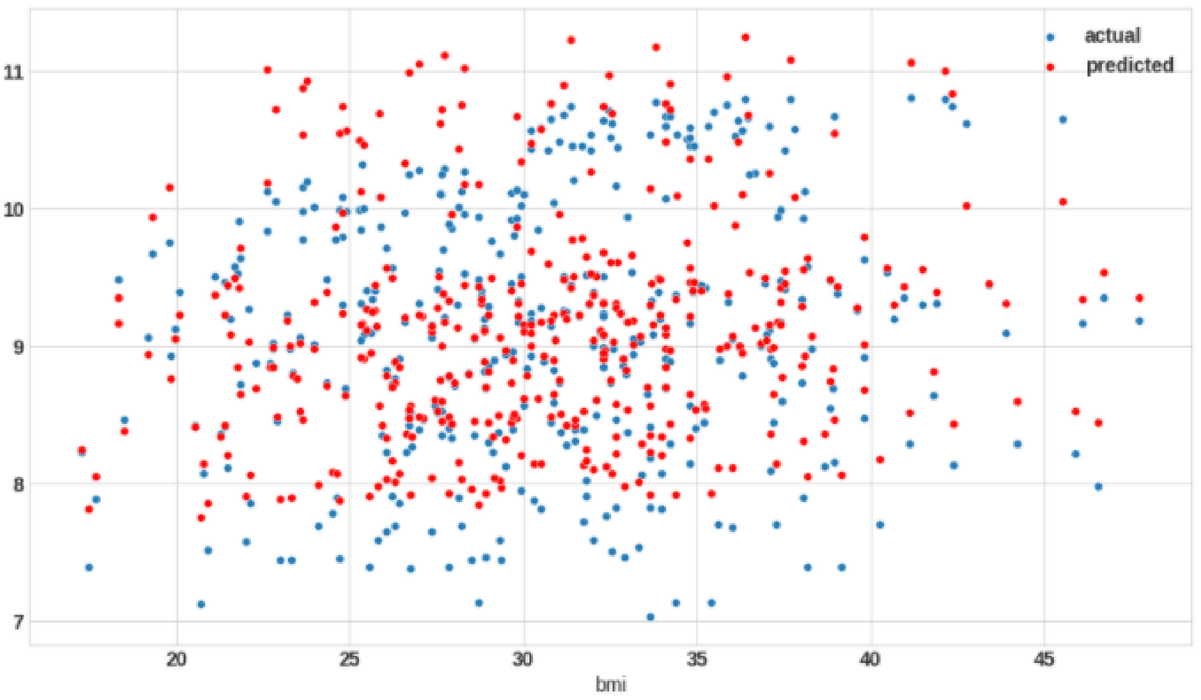
12. Plot the actual and the predicted values to check the relationship between the dependent and independent variables.

For model 1(implemented from scratch):

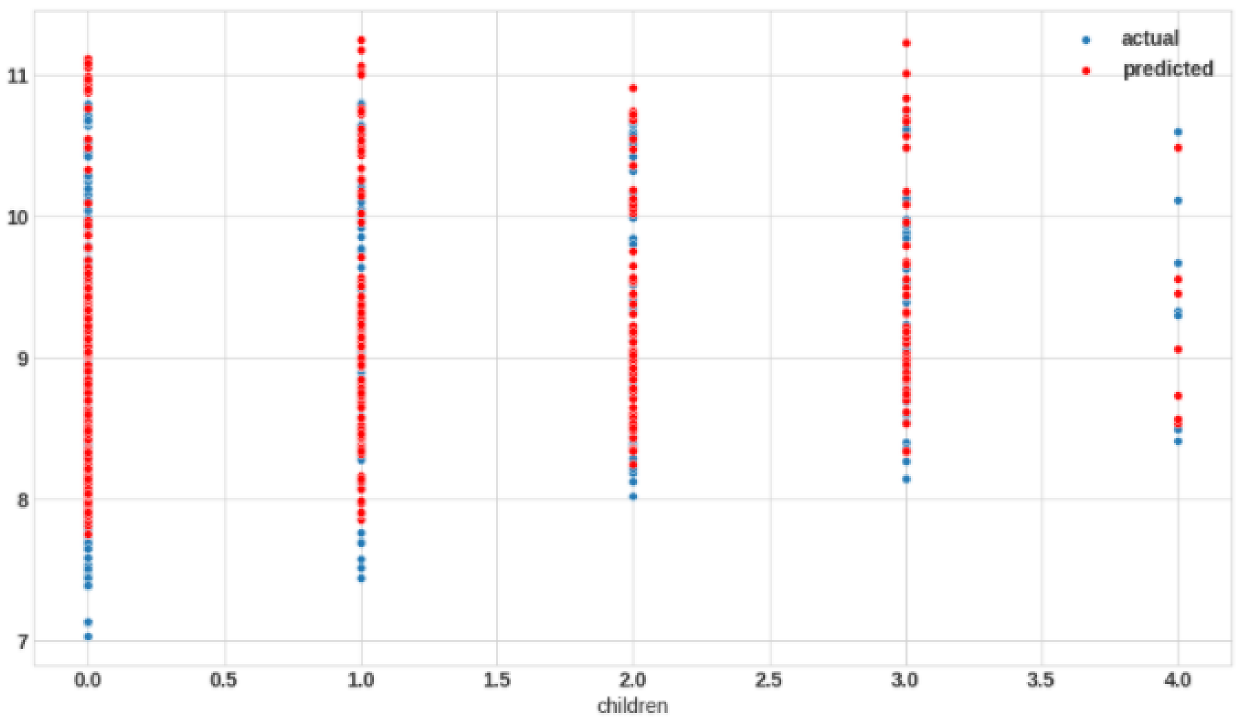




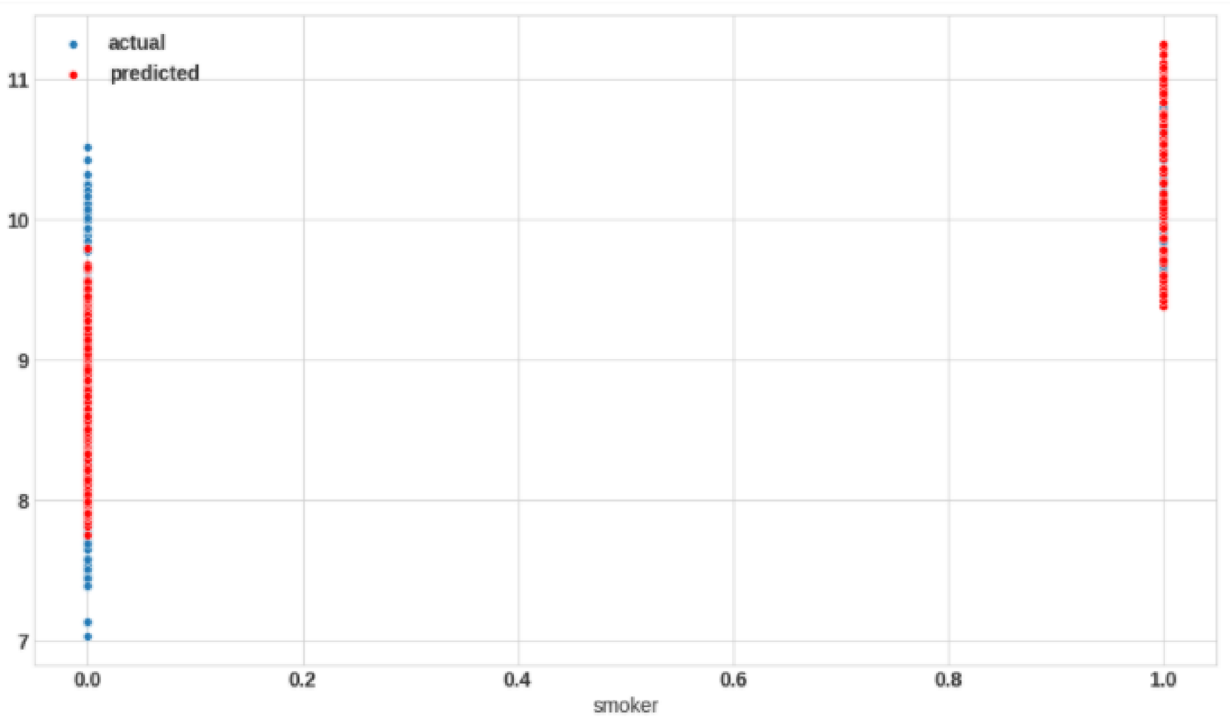
actual
predicted
sex



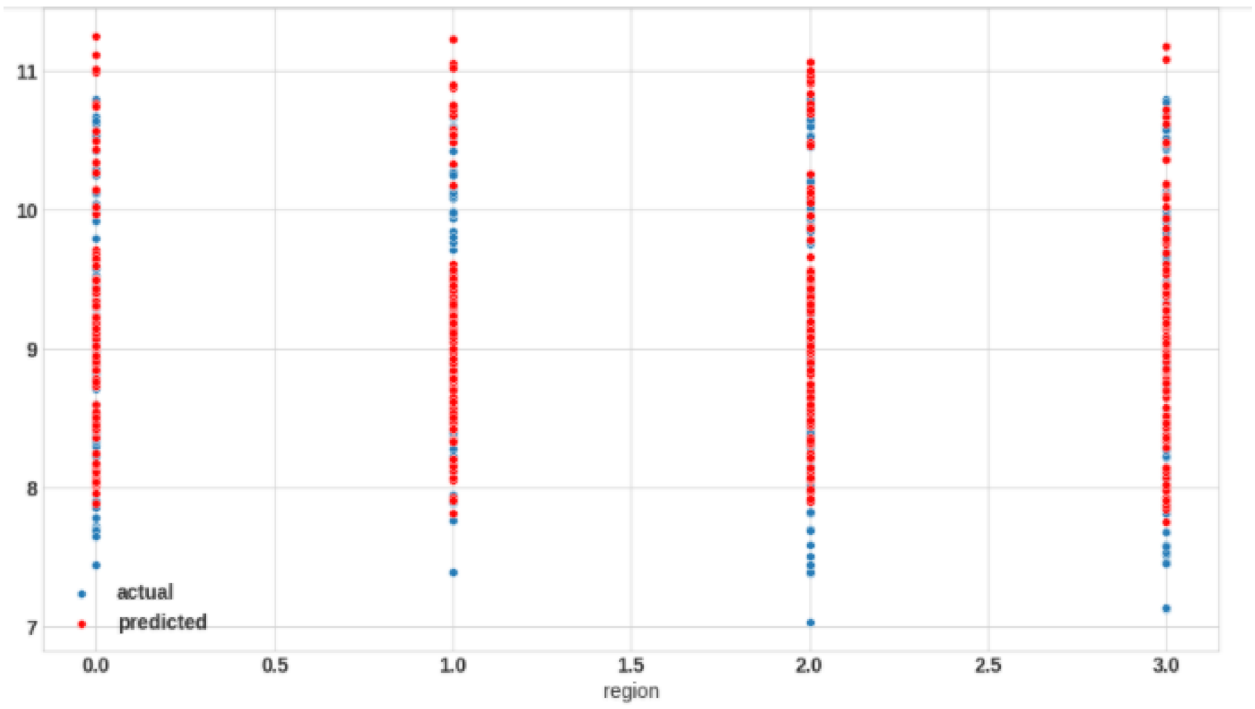
actual
predicted
bmi



actual
predicted
11
10
9
8
7
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
children



actual
predicted
11
10
9
8
7
0.0
0.2
0.4
0.6
0.8
1.0
smoker



Similarily the plots for model made from sklearn can be seen in the notebook.

The plots for both models are same