Final Project(Junyoung Seo)

Step 1 : Data Analysis

For my final project, I have selected the Medical Cost Personal Datasets, driven by my current concern about health insurance as a recent U.S. citizen. This transition is particularly significant for me, having previously relied on student insurance. The project offers a valuable opportunity to delve into the average costs of health insurance for individuals in the U.S. and to pinpoint factors that exhibit strong correlations with these costs.

The overarching objective of this project is to develop a regression model for predicting medical insurance charges. The model will be built using key features such as age, BMI, number of children, region, and smoking status.

Let's see the data.

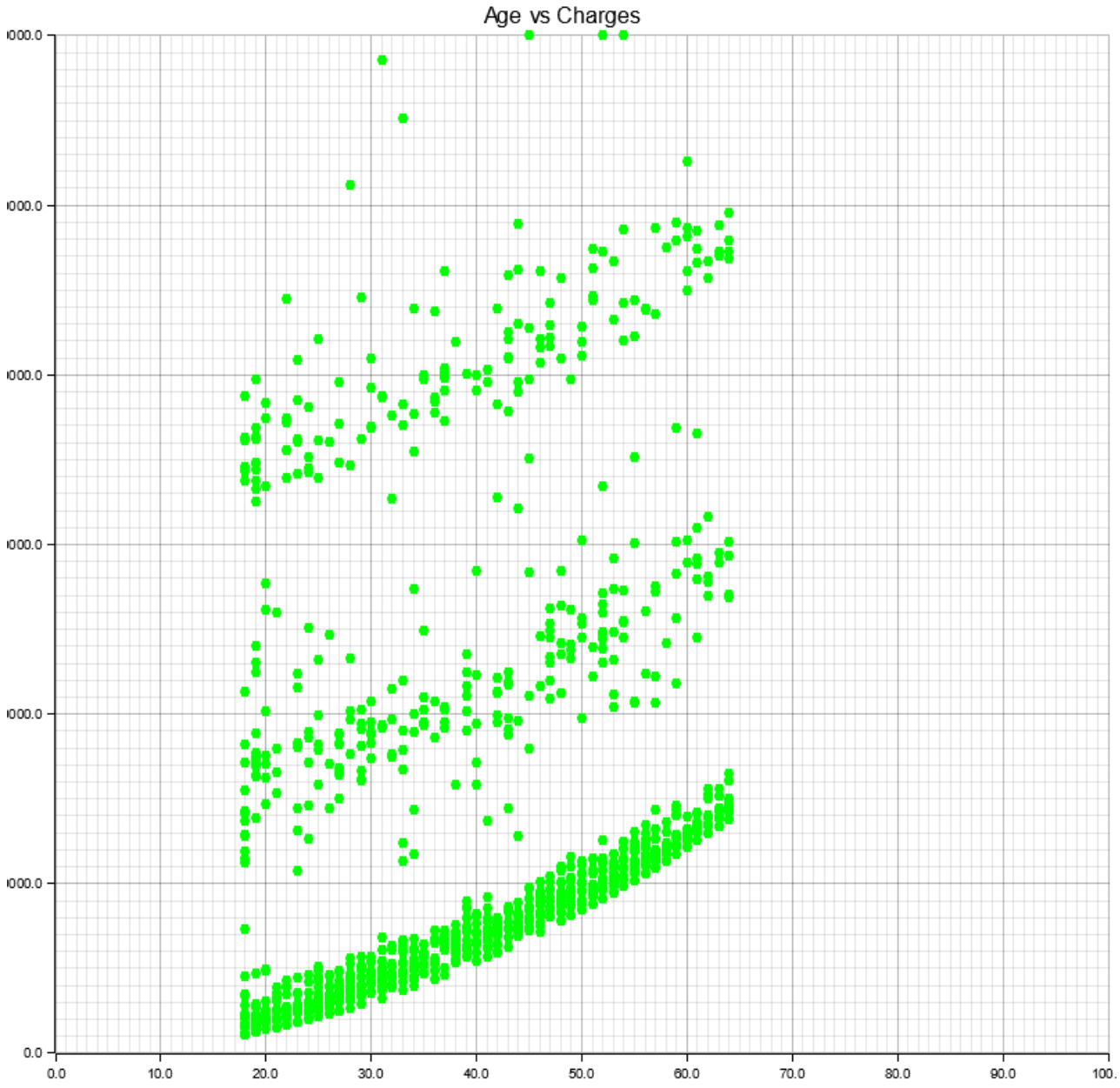
In [26]:
```
import pandas as pd

df = pd.read_csv('insurance.csv')

print(df.head())
```

```
   age     sex     bmi  children smoker     region      charges
0   19  female  27.900         0    yes  southwest  16884.92400
1   18    male  33.770         1     no  southeast   1725.55230
2   28    male  33.000         3     no  southeast   4449.46200
3   33    male  22.705         0     no  northwest  21984.47061
4   32    male  28.880         0     no  northwest   3866.85520
```

To assess the relationship between each factor (age, BMI, number of children, region, and smoking status) and medical insurance charges, I conducted a thorough analysis by comparing these factors with the charges. The results of this analysis are available in the "rs" file, located within the "graph" folder. This step is crucial in understanding how each factor contributes to the overall impact on medical insurance charges.

In [27]:
```
from IPython import display
display.Image("age_plot.png")
```

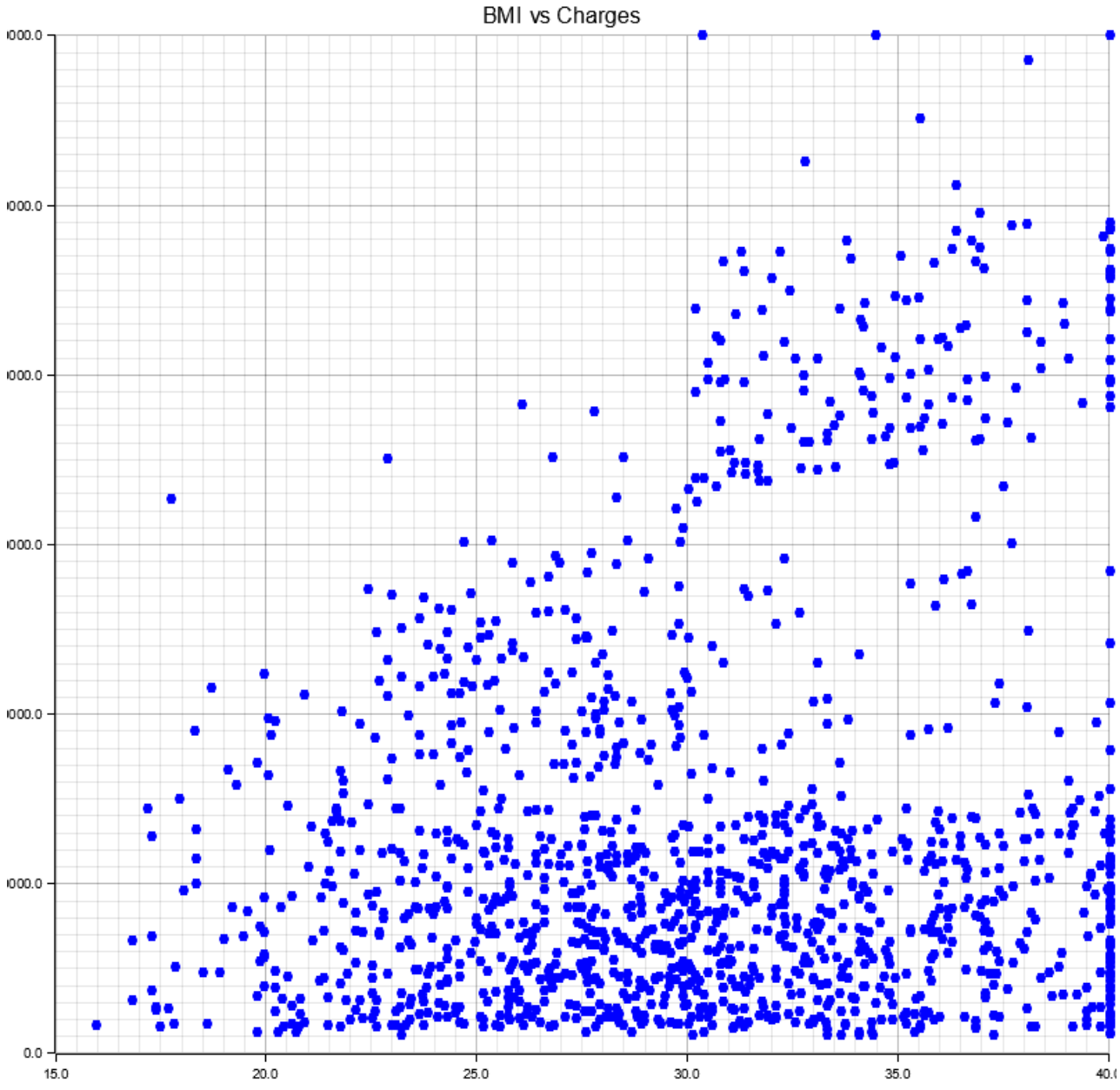
Out[27]:



In [28]:

```
from IPython import display
display.Image("bmi_plot.png")
```

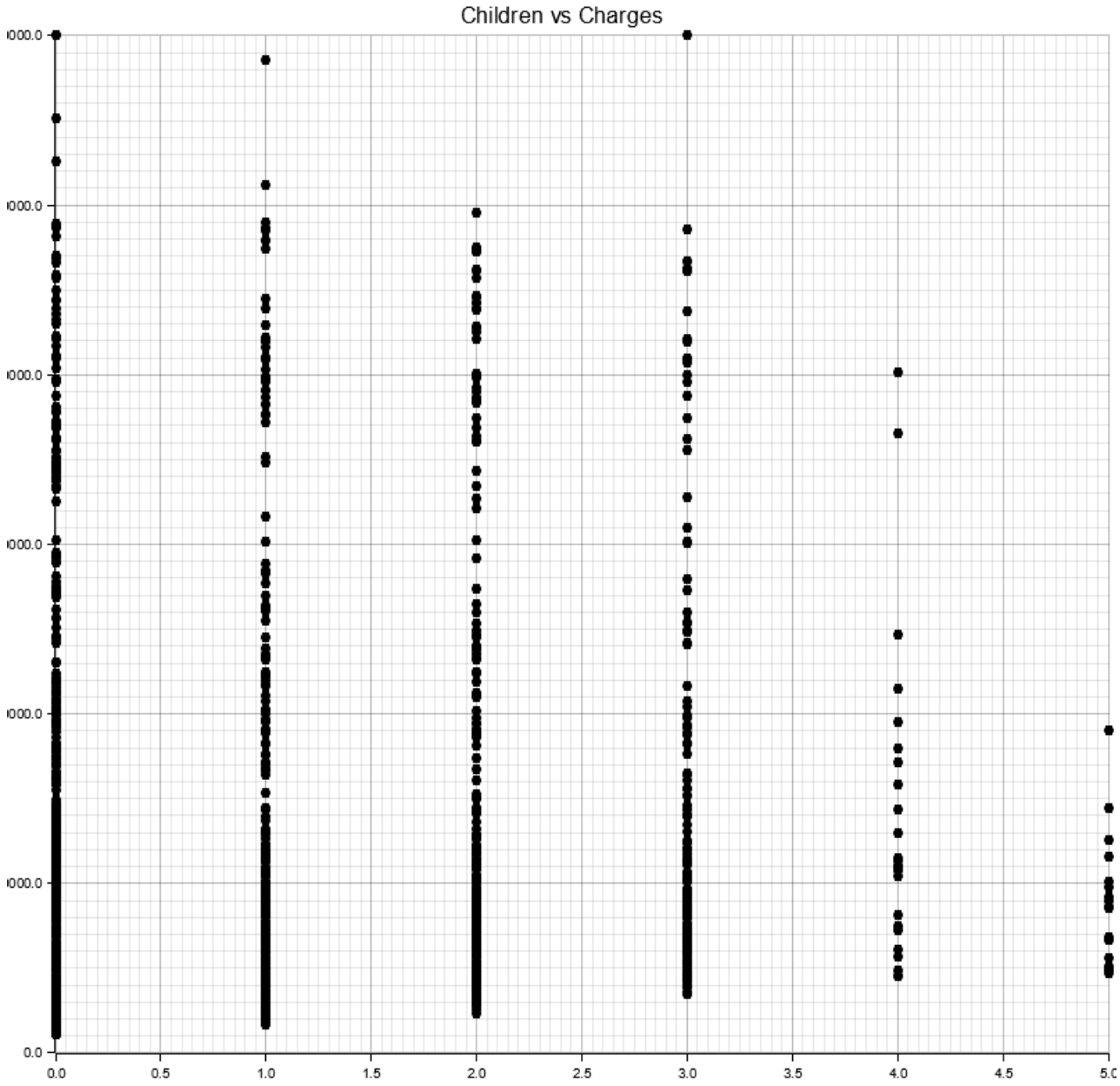
Out[28]:



In [29]:

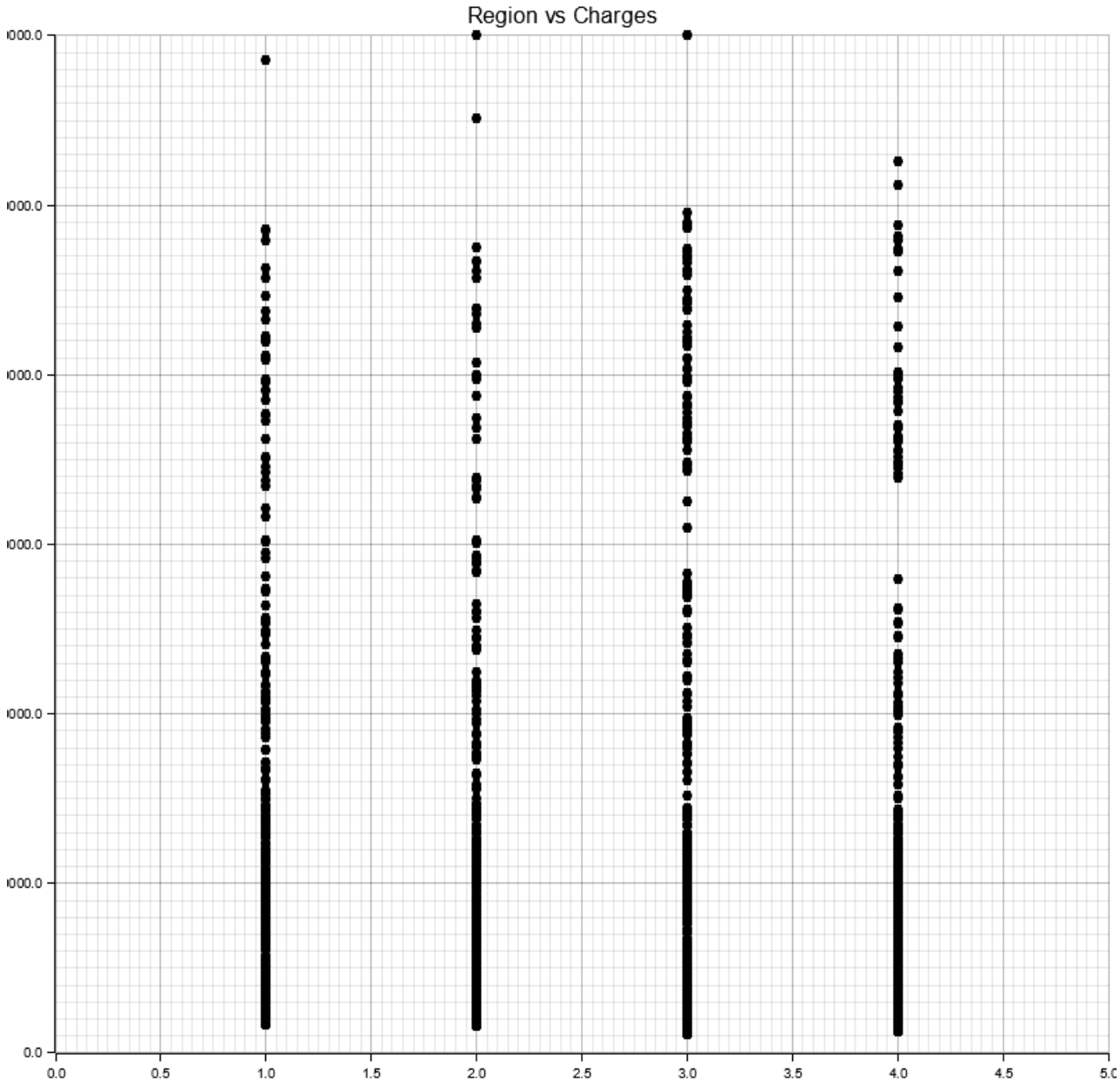
```
from IPython import display
display.Image("children_plot.png")
```

Out[29]:



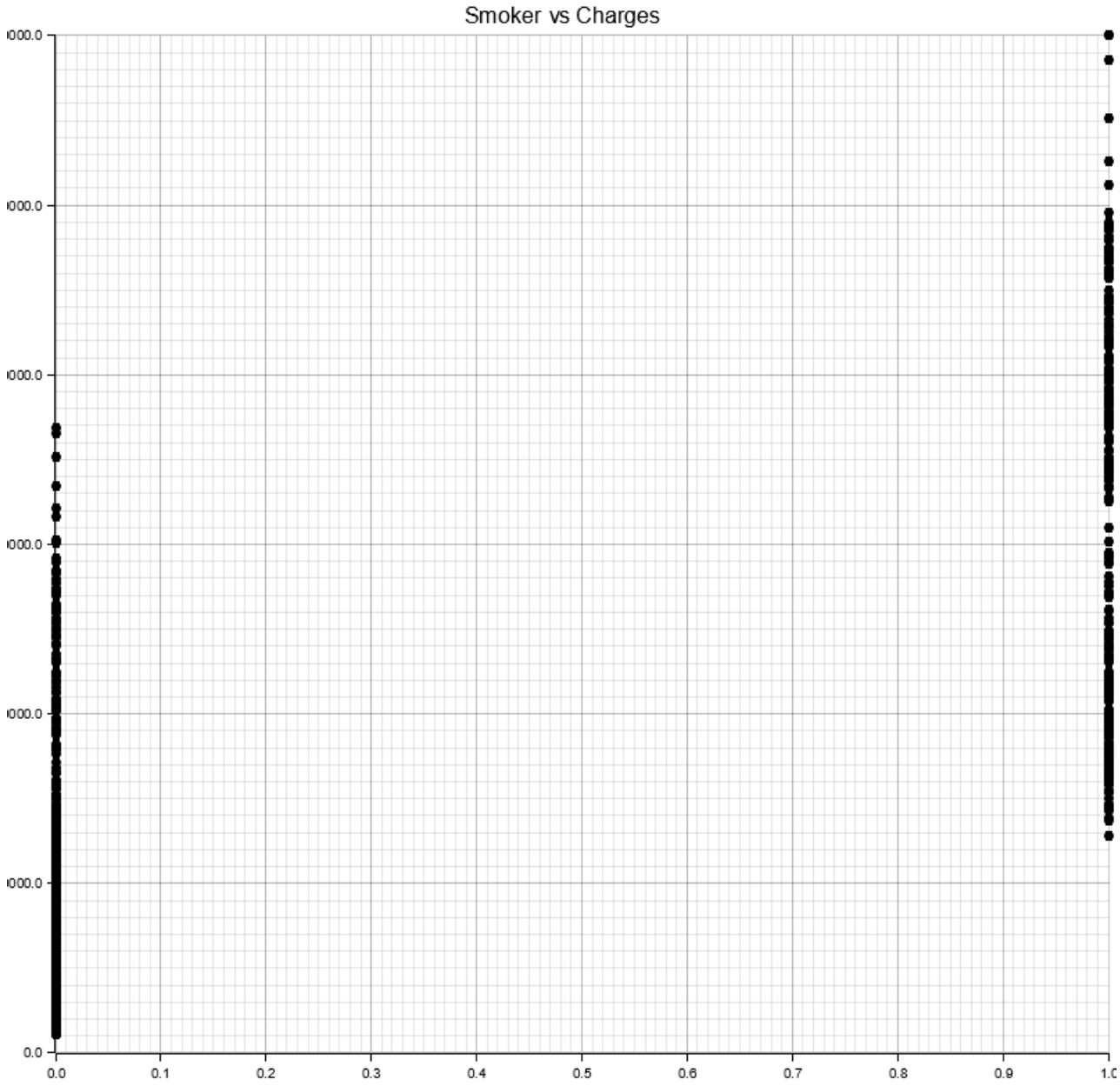
In [30]:
```
from IPython import display
display.Image("region_plot.png")
```

Out[30]:



In [31]:
```
from IPython import display
display.Image("smoker_plot.png")
```

Out[31]:



Based on the graphs, it is evident that, excluding the region factor, there is a clear linear correlation with medical insurance charges for each of the considered factors. Given this observation, I am inclined to believe that linear regression is the most suitable model for predicting charges based on the available data.

Step 2 : Encoding

In [32]:
```
print(df.head())
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

To facilitate the processing of linear regression, it's essential to convert all non-numerical values to numerical ones. In this context, I made the following replacements:

For 'sex,' I assigned 0 to 'female' and 1 to 'male.' For 'smoker,' I assigned 0 to 'no' and 1 to 'yes.' For 'region,' I replaced 'northeast' with 0, 'northwest' with 1, 'southeast' with 2, and 'southwest' with 3.

The outcome of these transformations is reflected in the modified_insurance.csv file.

In [33]:
```
df1 = pd.read_csv('modified_insurance.csv')

print(df1.head())
```

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	3	16884.92400
1	18	1	33.770	1	0	2	1725.55230
2	28	1	33.000	3	0	2	4449.46200
3	33	1	22.705	0	0	1	21984.47061
4	32	1	28.880	0	0	1	3866.85520

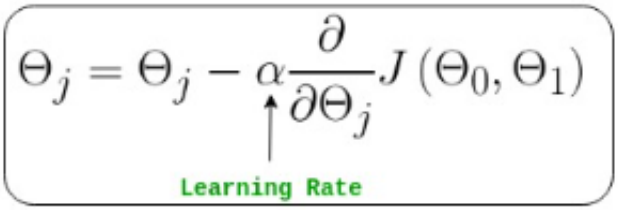
Following the replacements and modifications described earlier, I exported the resulting CSV file under the name "modified_insurance.csv."

Step 3 : Prediction

Now, let's delve into the implementation of linear regression for prediction. After experimenting with various algorithms, I settled on utilizing the gradient descent algorithm to construct the linear regression model.

In [35]:
```
from IPython import display
display.Image("gradient descent.png")
#image resource from https://www.geeksforgeeks.org/gradient-descent-in-linea
```

Out[35]: **Gradient Descent**



First, I contructed linear regression model with items(input features), label(output labels), iteration(the number of iterations for gradient descent), learning rate(in this case, it's 0.01), weight(the coefficients of each feature), bias(y-intercept of the linear equation), and y_mean(this is for testing). And then, I implemented fit, prediction and calculate_r_squared function to linear regression model.

I initiated the linear regression model, incorporating essential components such as items (input features), label (output labels), iteration (the number of iterations for gradient descent), learning rate (set at 0.01), weight (representing the coefficients of each feature), bias (the y-intercept of the linear equation), and y_mean (utilized for testing).

Subsequently, I implemented key functions within the linear regression model, including fit, predict, and calculate_r_squared. These functions contribute to the training and evaluation processes of the linear regression model.

In the main function, the initial step involves reading data from the "modified_insurance.csv" file, which is the encoded version of the dataset. The program then proceeds to execute linear regression with the provided dataset, ultimately returning the R^2 value. This value serves as an interpretative accuracy score for evaluating the model's predictions.

The following is the result of the R^2 for the prediction.

Finished dev [unoptimized + debuginfo]

target(s) in 0.02s

R^2: 0.9290774733449599

As R^2 approaches 1, the accuracy of the prediction improves. Normally, a model is considered effective when R^2 exceeds 0.7. In this case, with an R^2 value of approximately 0.93 (rounded), we can confidently assert that this model demonstrates a high level of accuracy.

Step 4 : Test Case

Here's the result for the test case:

running 1 test

test tests::testing ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.05s

Given the successful testing and the model's ability to produce accurate predictions, I confidently conclude that this linear regression model is reliable for predicting medical insurance charges.