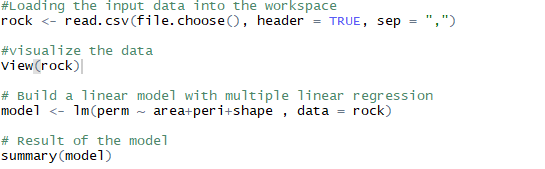
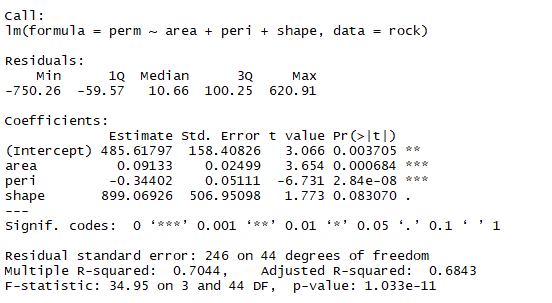
**Machine Learning for Data Science**

1. **Build Linear model and check the relationship:**

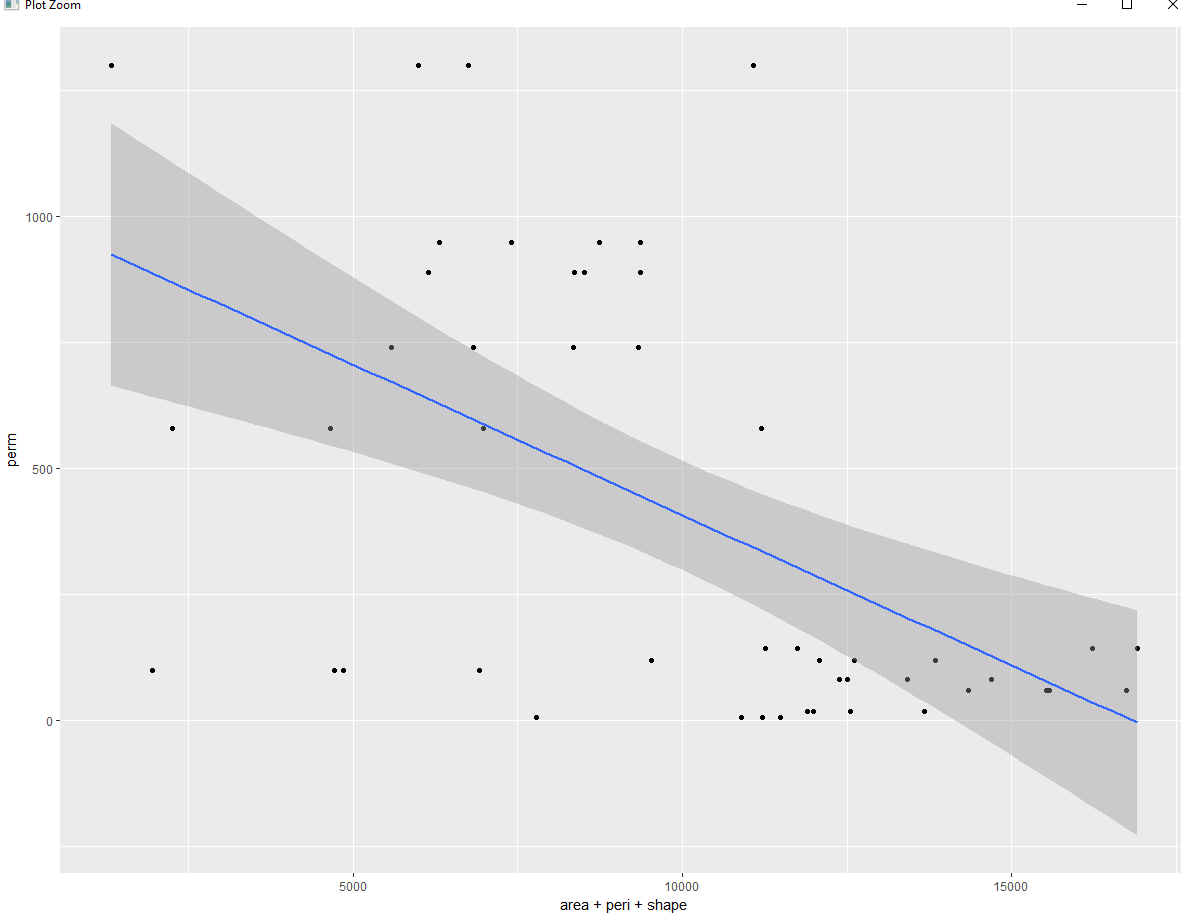
In-order to build the linear model with the given dataset, we have to perform the multiple linear regression as it involves 3 independent variables and one dependent variable

**R code –**







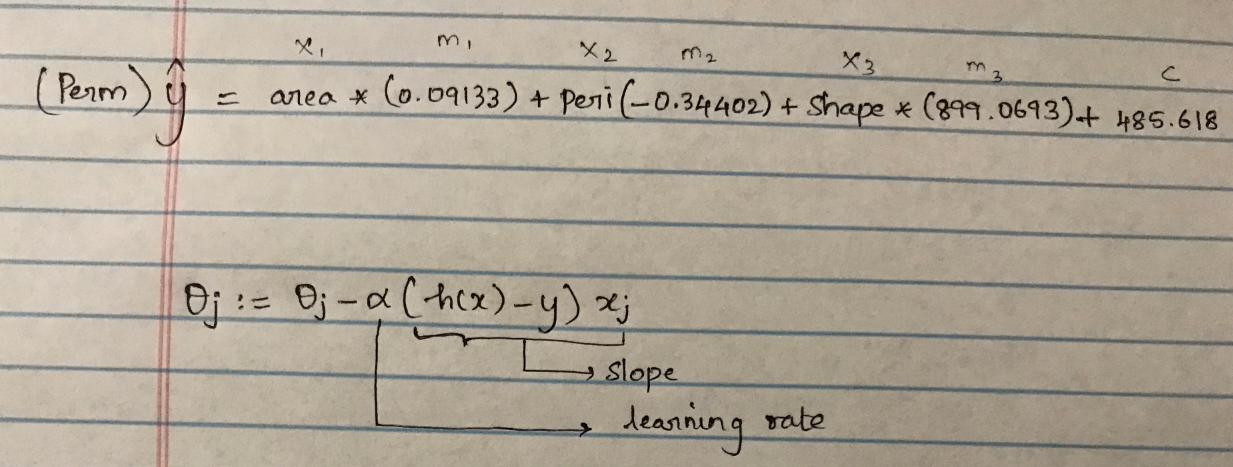


As per the above output, we can see that there is a strong linear relationship between the 3 independent variables and the response variable Perm.

Also, we can notice that the slope of the line is negative which implies that there is a strong negative linear relationship among the attributes.

1. **Gradient Descent Algorithm to find optimal intercept and gradient**

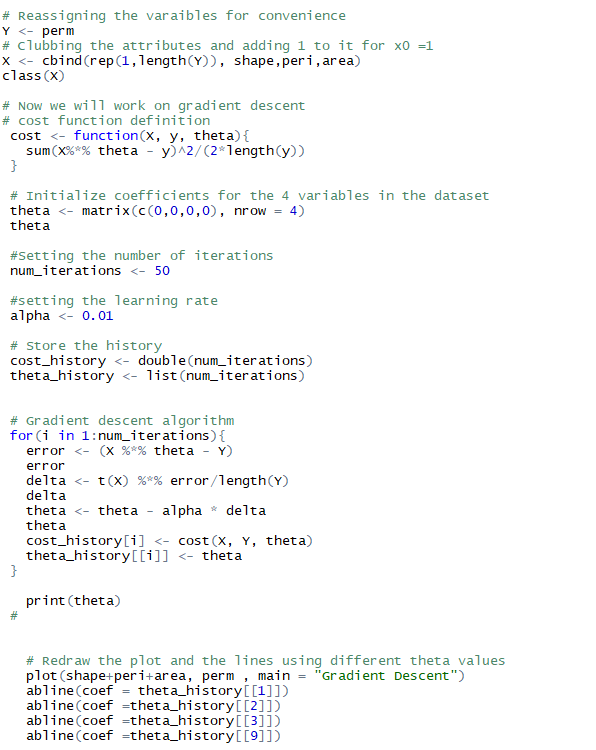
Gradient descent algorithm helps to find the error at its minimum. The formula for the gradient descent is given below.



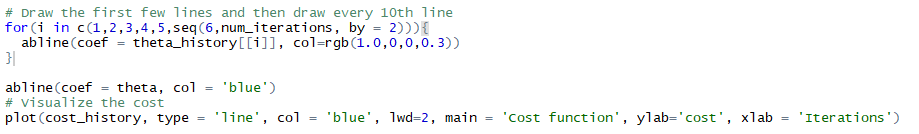
Steps for the gradient descent algorithm:

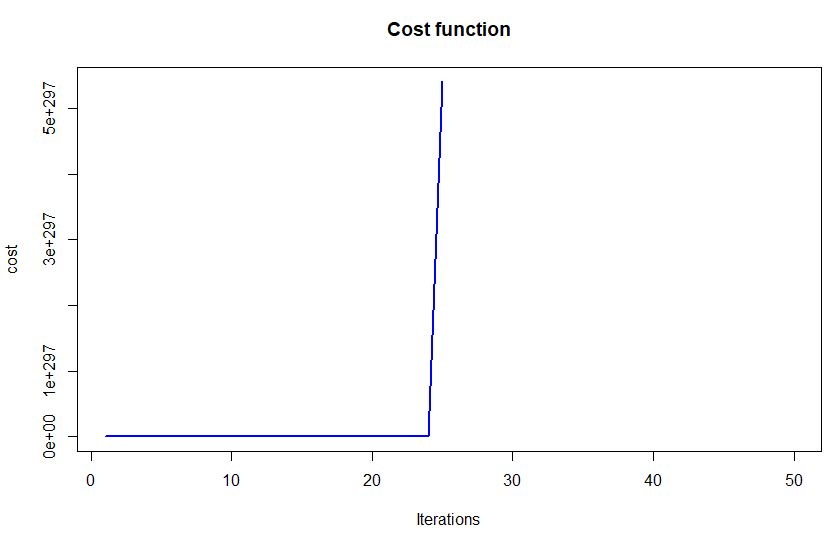
1. Defining the cost function
2. Initializing the co-efficient based on the given input dataset (i.e., four variables)
3. Initialize the number of iterations to get the convergence
4. Set the learning rate (alpha = 0.01/0.03/0.05)
5. Create the variables for theta and cost to store the values for every iteration
6. Call the error, delta and theta values and call the cost functions in the loop with the defined iterations and print the theta values for the required iteration
7. Visualize the cost to check on how it reaches to the minima.

**R code for Gradient Descent Algorithm-**



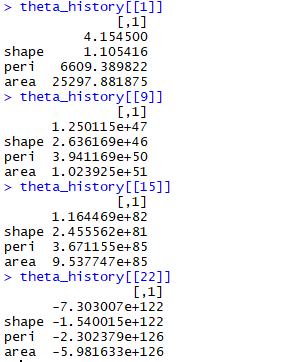
1. **Visualize the cot function:**

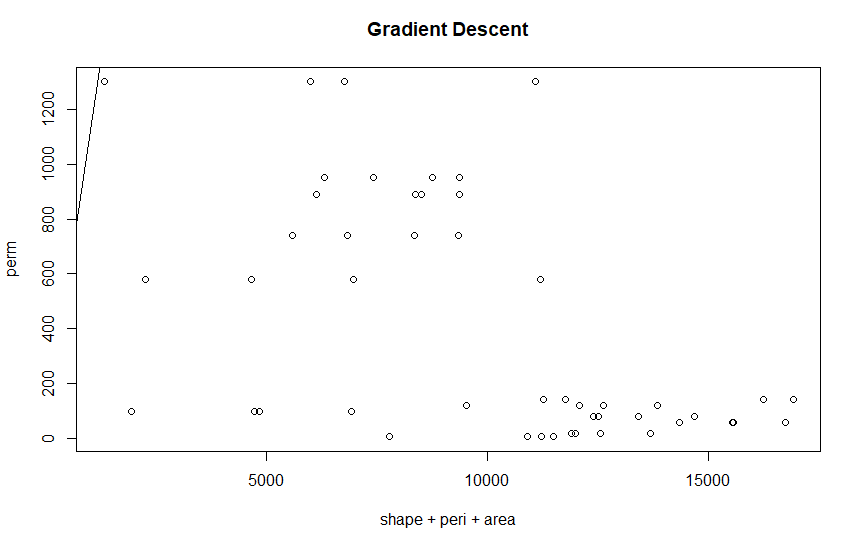


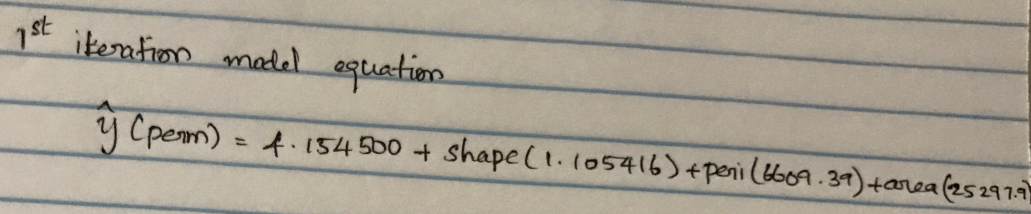


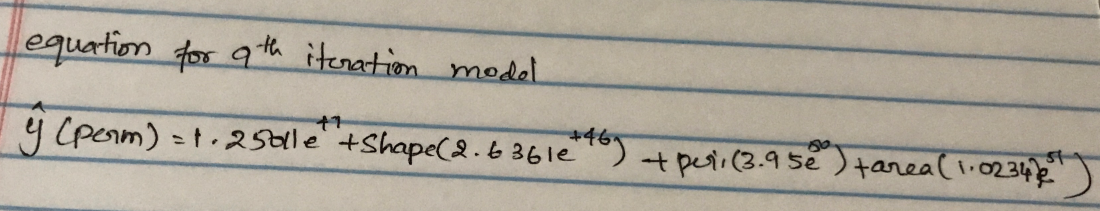
1. **Report the Model:**

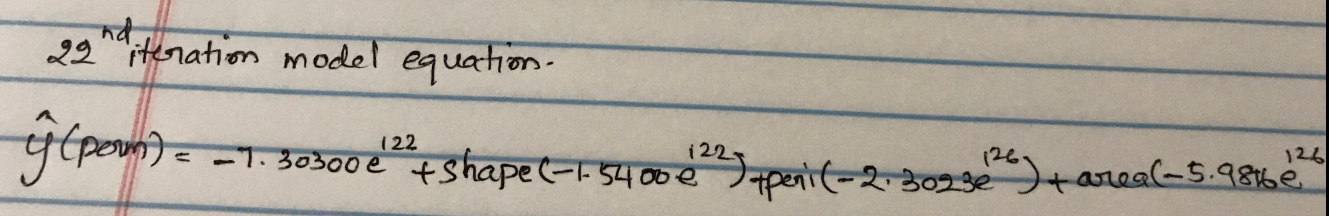
The value of theta for every iteration differs and the absolute fit line is visible only for the first iteration.











The above are the linear models build with different iterations. These are the values from the theta history for some iterations.

We can’t much notice the convergence of the cost function. Also, the plot for the cost function for the number of iterations doesn’t look in a sensible way. We can see the cost function seems to be stable until certain iterations and then it suddenly increases rather than getting minimized.

We can also notice that the cost history gets to Infinity after 20 iterations and theta history also gets infinity after 50 iterations which badly affects the cost function.

Running more iterations (more than 100) on this dataset seems to go beyond the convergence limits and take off the surface giving the undefined values NaN values.

This has also been tried with different learning rate (0.01,0.05,0.03)

Overall, the cost function estimation for this dataset is not in a pretty good state.

This may be because the dataset is not normalized. We can see that all the four variables have very different units. Scaling or normalizing them between 0-1 will be the ideal solution to handle this type of dataset. After scaling, may be, we can find the minimized cost function.

------------------------------------------------------------------------------------------------------------------------------------------