IT 166 Lab 13

Linear regression in R

Objectives

- Be able to use R to perform linear regression on given datasets.

Preparation

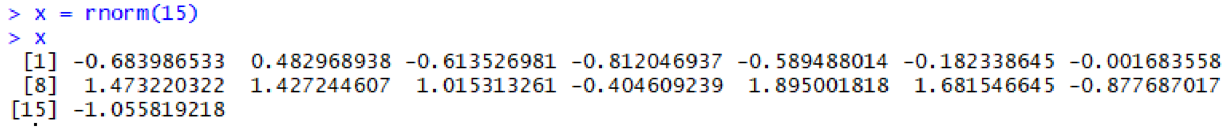
- Launch the R studio.
- Set your current working directory.
- Create a new R script named lab_13.

Follow the steps below to complete the lab.

Problem 1

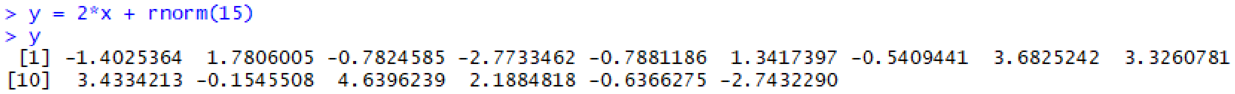
A toy problem

Use the rnorm function to generate a sequence of 15 numbers and assign it to x. (The rnorm function works similarly to Numpy's randn function. By default, it randomly generate floating point numbers following the (0,1) Gaussian distribution.)



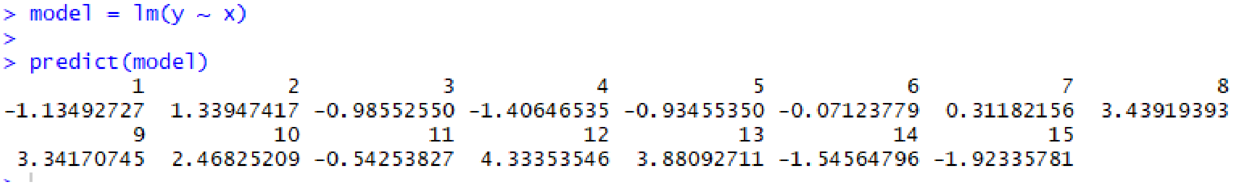
```
> x = rnorm(15)
> x
 [1] -0.683986533  0.482968938 -0.613526981 -0.812046937 -0.589488014 -0.182338645 -0.001683558
 [8]  1.473220322  1.427244607  1.015313261 -0.404609239  1.895001818  1.681546645 -0.877687017
[15] -1.055819218
```

Then use x to define y, where y is a linear function regarding x:



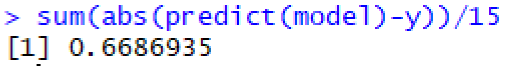
```
> y = 2*x + rnorm(15)
> y
 [1] -1.4025364  1.7806005 -0.7824585 -2.7733462 -0.7881186  1.3417397 -0.5409441  3.6825242  3.3260781
[10]  3.4334213 -0.1545508  4.6396239  2.1884818 -0.6366275 -2.7432290
```

Define a linear model using x and y and make the prediction based on x:



```
> model = lm(y ~ x)
>
> predict(model)
          1           2           3           4           5           6           7           8 
-1.13492727  1.33947417 -0.98552550 -1.40646535 -0.93455350 -0.07123779  0.31182156  3.43919393 
          9          10          11          12          13          14          15 
 3.34170745  2.46825209 -0.54253827  4.33353546  3.88092711 -1.54564796 -1.92335781 
```

Compute the mean absolute error (MAE) of the prediction:



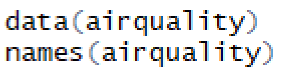
```
> sum(abs(predict(model)-y))/15
[1] 0.6686935
```

Problem 2

Linear regression on a real dataset (part one)

One nice feature of R is that there are several built-in datasets in R. We will be using the airquality dataset in this problem. The dataset is the daily air quality measurements in NewYork, May to September 1973. The dataset has six measures (features): ozone, solar radiation, wind, temperature, month, and day.

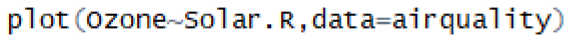
To read in the dataset and check the names (features), do as the follows:



```
data(airquality)
names(airquality)
```

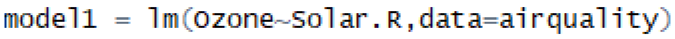
One fun fact, you can use the command: data(), to take a look at all the built-in datasets.

We will use Ozone and Solar.R for the regression. Particularly, Ozone is y and Solar.R is x. Let us visualize the relationship of the two measures by plotting:



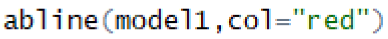
```
plot(Ozone~Solar.R,data=airquality)
```

We can hardly say that ozone and solar radiation are linearly correlated, based on the plot. But it does not hurt to perform the linear regression. Let us build the linear regression model:



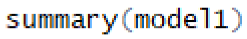
```
model1 = lm(Ozone~Solar.R,data=airquality)
```

In order to visualize the line generated by the regression, you will need to use the following command:



```
abline(model1,col="red")
```

You then will see a red line appeared upon the scattered data points. Finally, you can display the summary for the regression by using the following command:



```
summary(model1)
```