Sushobhan Parajuli
MTH 4130 : Mathematics of Data Analysis
Professor Giulio Trigila
Final Project
12/18/2021

Linear Regression Model with R

For this project, a file train.txt was given and it contained two columns of data. The first column represented the independent variable that we interpreted as time and the second column was relative to the price of some good, say ice cream. I used train.txt to train the linear regression model. A file test.txt was also given which I used, in the end, to test the linear regression model on. I coded with R in the Jupyter Notebook.

First, training and testing datasets were imported using R code :

```
train = read.table("train.txt")
test = read.table("test.txt")
```

I worked with the training dataset here after till model building.

Then, first few entries in the training dataset were inspected:

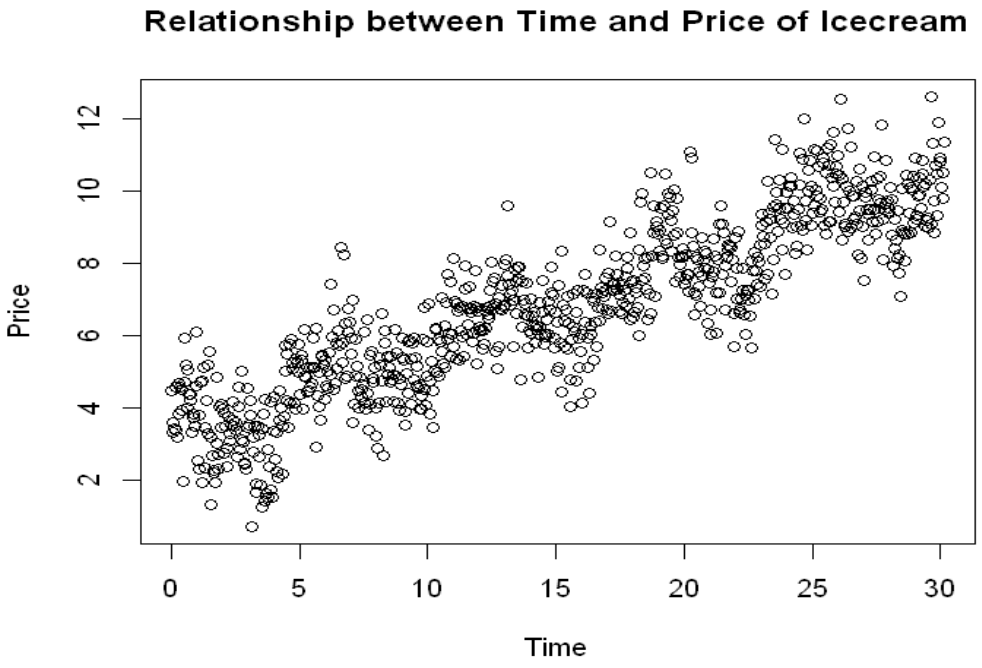
```
head(train)
```

V1	V2
0.00000000	4.456329
0.03773685	3.301815
0.07547370	3.595955
0.11321055	3.401495
0.15094739	4.571147
0.18868424	3.396323

The first column V1 contains Time, the independent variable and the second column V2 contains Price, the response variable.

A scatter plot showing the relationship between time and price was plotted:

```
#plot the data in train table
plot(train, main="Relationship between Time and Price", xlab="Time", ylab="Price")
```



There seems to be a linear relationship between time and price.

Simple Linear Regression Model:

A simple linear regression is a way to model the linear relationship between two quantitative variables, predictor variable (independent variable) and response variable. Here, I used R to fit a linear model y = β0 + β1x to the data contained in train.txt and printed a summary table.

```
model.1 = lm(V2 ~ V1, data = train) # fit a linear model
summary(model.1)
```

```
Call:
lm(formula = V2 ~ V1, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2429 -0.6847  0.0114  0.6594  3.6746

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.191038   0.077903   40.96   <2e-16 ***
V1          0.238398   0.004474   53.29   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.103 on 798 degrees of freedom
Multiple R-squared:  0.7806,    Adjusted R-squared:  0.7804
F-statistic:  2840 on 1 and 798 DF,  p-value: < 2.2e-16
```

According to the above summary table, the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ were 3.19 and 0.239 respectively. Estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are the sample parameters and are not sufficient to conclude that the entire population follows a linear trend. So, I carried out hypothesis testing to determine the linear trend in the population.

For population slope parameter,
Null hypothesis: $\beta_1 = 0$ (i.e. there is no relationship)
Alternative hypothesis: $\beta_1 \neq 0$
I took the industry standard 0.05 significance level.

From the above summary table, the p-value (Pr(>|t|)) of V1 was observed to be 2e-16 which is less than 0.05. Thus the null hypothesis was rejected in favor of the alternative hypothesis. In other words, β_1 is not equal to zero and therefore there is a linear relationship between time and price.

Similarly, for population y-intercept parameter,
Null hypothesis: $\beta_0 = 0$
Alternative hypothesis: $\beta_0 \neq 0$
I took the industry standard 0.05 significance level.

From the above summary table, the p-value (Pr(>|t|)) of Intercept was observed to be 2e-16 which is less than 0.05. Thus the null hypothesis was rejected in favor of the alternative hypothesis and it concluded that β_0 is not equal to zero.

Above simple linear regression model had an adjusted R-squared value of 0.7804. Which can be interpreted as that time explains 78% of variation in price. The R-squared, also known as the coefficient of determination, explains the degree to which independent variable explains the variation in response variable. In multiple regression, adjusted R-squared value is preferred as it includes whether the additional variables are contributing to the model. The adjusted R-squared value ranges from 0 to 1. A higher adjusted R-squared value infers that the model is better.

Multiple Linear Regression Model:

A multiple linear regression is a way to model the linear relationship between one quantitative response variable and more than one predictor variable or feature.

Non-linear features were added in the above simple linear regression model to improve it, as a fellow economist suggested. The features provided were cos(x), log(x), cos(4x), sin(3x), sin(5x) and $\sin(2x) \times \cos(2x)$.

These features were added in the training dataset using following code:

```
cosx = c()
logx = c()
cos4x = c()
sin3x = c()
sin5x = c()
sin2xcos2x = c()
for (x in train$V1) {
    cosx = c(cosx, cos(x))
    logx = c(logx, log(x))
    cos4x = c(cos4x, cos(4*x))
    sin3x = c(sin3x, sin(3*x))
    sin5x = c(sin5x, sin(5*x))
    sin2xcos2x = c(sin2xcos2x, sin(2*x)*cos(2*x))}

train$cos_x = cosx
train$log_x = logx
train$cos_4x = cos4x
train$sin_3x = sin3x
train$sin_5x = sin5x
train$sin2x_cos2x = sin2xcos2x
```

Then first few entries in the training dataset were inspected:

```
head(train)
```

V1	V2	cos_x	log_x	cos_4x	sin_3x	sin_5x	sin2x_cos2x
0.00000000	4.456329	1.0000000	-Inf	1.0000000	0.0000000	0.0000000	0.00000000
0.03773685	3.301815	0.9992880	-3.277118	0.9886291	0.1129689	0.1875667	0.07518741
0.07547370	3.595955	0.9971532	-2.583971	0.9547748	0.2244914	0.3684754	0.14866492
0.11321055	3.401495	0.9935985	-2.178506	0.8992072	0.3331398	0.5363047	0.21876151
0.15094739	4.571147	0.9886291	-1.890824	0.8231899	0.4375230	0.6850971	0.28388304

Note: values for trigonometry functions are in radian.

The first entry of feature log(x) contained -inf, so the first row was removed with the following code:

```
train = train[!is.infinite(rowSums(train)),]
```

```
head(train)
```

	V1	V2	cos_x	log_x	cos_4x	sin_3x	sin_5x	sin2x_cos2x
2	0.03773685	3.301815	0.9992880	-3.277118	0.9886291	0.1129689	0.1875667	0.07518741
3	0.07547370	3.595955	0.9971532	-2.583971	0.9547748	0.2244914	0.3684754	0.14866492
4	0.11321055	3.401495	0.9935985	-2.178506	0.8992072	0.3331398	0.5363047	0.21876151
5	0.15094739	4.571147	0.9886291	-1.890824	0.8231899	0.4375230	0.6850971	0.28388304

A multiple linear regression model was built using all the features against price and the summary table was calculated to inspect statistically significant features.

The following code was used to build the model:

```
model.2 = lm(V2 ~ V1 + cos_x + log_x + cos_4x + sin_3x + sin_5x + sin2x_cos2x, data=train)
summary(model.2)
```

Executing summary(model.2) printed the following summary table for model.2.
The p-values Pr(>|t|) were inspected and analyzed to select the right features.

```
    Min      1Q  Median      3Q     Max
-2.3670 -0.5776 -0.0038  0.5532  3.1854

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.072374   0.090288  34.029  < 2e-16 ***
V1           0.238831   0.007643  31.248  < 2e-16 ***
cos_x        0.809449   0.045967  17.609  < 2e-16 ***
log_x        0.053559   0.068163   0.786    0.432
cos_4x      -0.022580   0.045351  -0.498    0.619
sin_3x       0.374532   0.045490   8.233  7.5e-16 ***
sin_5x       0.022931   0.045452   0.505    0.614
sin2x_cos2x -0.027492   0.091087  -0.302    0.763
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9058 on 791 degrees of freedom
Multiple R-squared:  0.8531,    Adjusted R-squared:  0.8518
F-statistic: 656.3 on 7 and 791 DF,  p-value: < 2.2e-16
```

The p-values for features log(x), cos(4x), sin(5x), and sin(2x)cos(2x) were greater than 0.05. It was now understood that the regression parameters were equal to zero and thus the null hypothesis for each feature failed to be rejected. Thus no relationship between these 4 features and price was concluded. Such features were dropped in order to avoid redundancy in the model.

On the other hand, the p-values for features cos(x) and sin(3x) were less than 0.05. Thus these features were concluded to be statistically significant and their null hypothesis of regression parameters being zero was rejected in the favor of alternative hypotheses. These features had a relationship with the price and were chosen in the model to improve the model.

Finally, the multiple linear regression model was built with features V1, cos(x), and sin(3x).

The R code along with the summary table is as follows:

```
model.3 = lm(V2 ~ V1 + cos_x + sin_3x, data=train)
summary(model.3)
```

```
Call:
lm(formula = V2 ~ V1 + cos_x + sin_3x, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3969 -0.5837 -0.0073  0.5477  3.1956

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.122868   0.064114  48.708  < 2e-16 ***
V1          0.244023   0.003687  66.188  < 2e-16 ***
cos_x       0.805983   0.045706  17.634  < 2e-16 ***
sin_3x      0.371546   0.045158   8.228 7.78e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9042 on 795 degrees of freedom
Multiple R-squared:  0.8529,    Adjusted R-squared:  0.8523
F-statistic:  1536 on 3 and 795 DF,  p-value: < 2.2e-16
```

From the above summary table, the adjusted R-squared value was observed to be 0.8523 which is greater than the simple linear regression model. Therefore, this multiple linear regression model was concluded to be better than the previous simple linear regression model.

Testing the Model:

First, the features cos(x) and sin(3x) were added in the test dataset.

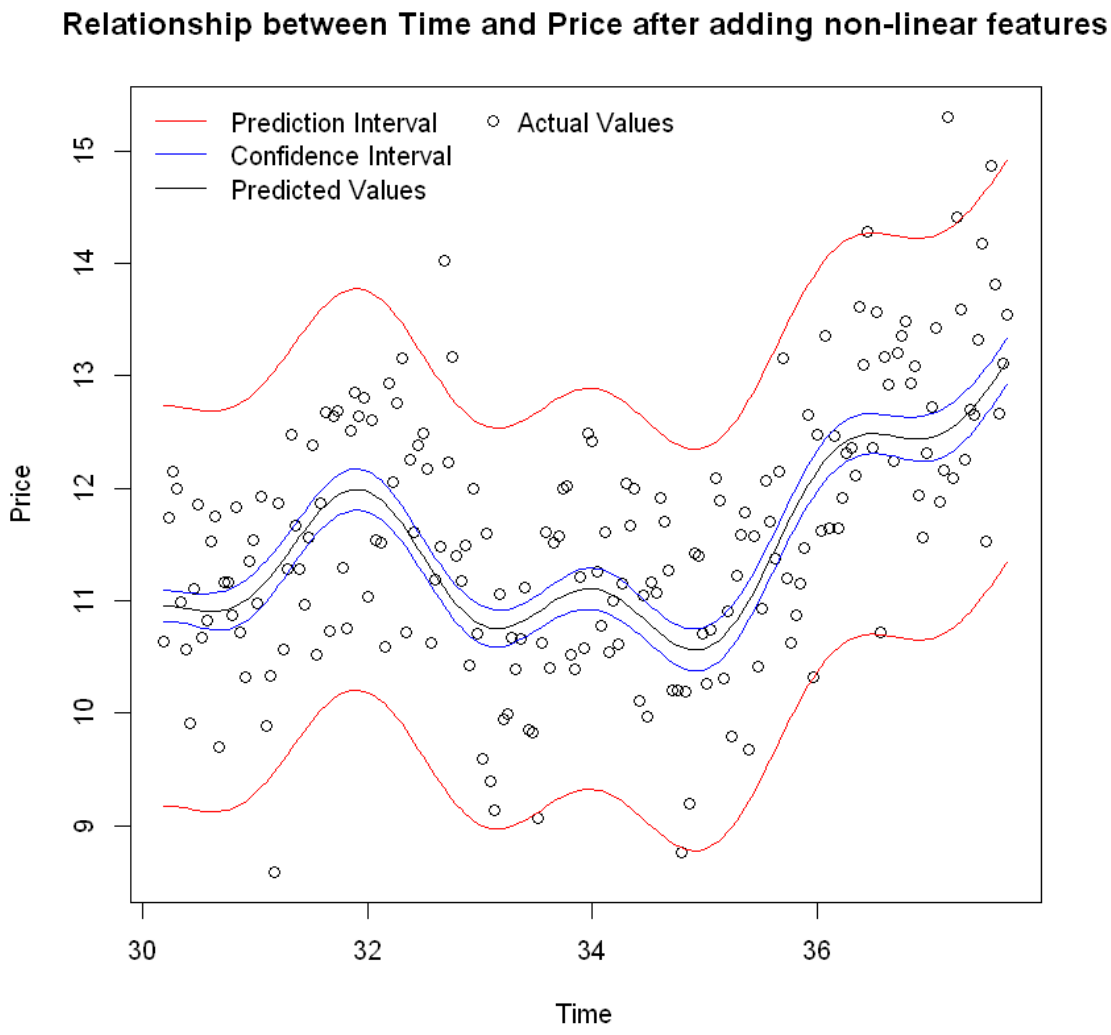
```
cosx = c()
sin3x = c()
for (x in test$V1) {
    cosx = c(cosx, cos(x))
    sin3x = c(sin3x, sin(3*x))}

test$cos_x = cosx
test$sin_3x = sin3x
head(test)
```

V1	V2	cos_x	sin_3x
30.18948	10.63221	0.3375838	0.51220481
30.22722	11.74352	0.3728567	0.41190071
30.26495	12.14874	0.4075986	0.30632302

Then the predicted values, prediction interval and confidence interval were calculated with the following code:

```
conf_V2 = predict(model.3, newdata=test, interval = "confidence", level = 0.95)
pred_V2 = predict(model.3, newdata=test, interval = "prediction", level = 0.95)
#plot the data in test table
plot(test$V1, test$V2,main="Relationship between Time and Price after adding non-linear features",
     xlab="Time", ylab="Price")
lines(test$V1,pred_V2[,1])
lines(test$V1,pred_V2[,2],col="red")
lines(test$V1,pred_V2[,3],col="red")
lines(test$V1,conf_V2[,2],col="blue")
lines(test$V1,conf_V2[,3],col="blue")
legend("topleft",
       legend = c("Prediction Interval", "Confidence Interval", "Predicted Values"),
       lwd = 1, col = c("red","blue","black"),  bty = "n")
legend("top",legend = c("Actual Values"), pch = 1,  bty = "n")
```

Following is the plot for actual values, predicted values line, predicted and confidence intervals:



From the above plot, we can be 95% confident that the mean price of all ice creams at a given time falls within blue lines. And we can be 95% confident that a single observation of price at a given time falls within red lines. The line for predicted values is within the confidence interval (blue lines) and most of the actual values are within the prediction interval (red lines). Thus, there is a good agreement between the model (model.3) and the data in the test dataset.

The regression model, with the integration of non-linear features cos(x) and sin(3x), is better than the simple linear regression model and can be used to predict the price of Icecream for a given time.