Anchor University Journal of Science and Technology (AUJST)
A publication of the Faculty of Science and Science Education, Anchor University Lagos

URL: journal.aul.edu.ng

In AJOL: https://www.ajol.info/index.php/aujst

Vol. 2 No 2, December 2022, Pp. 134 - 142

ISSN: 2736-0059 (Print); 2736-0067 (Online)

Programming Multi-Linear Regression equation for faster problem-solving and decision-making

Anichebe G.E.

Department of Computer Science, University of Nigeria, Nsukka

Corresponding Author:
gregory.anichebe@unn.edu.ng

Competing Interests: The authors declare no competing interests.

ABSTRACT

Background: Multi-Linear regression equation is one of the widely used mathematical modeling techniques employed by data scientists for making predictions from a given dataset (Y, X) where X is the set of independent variables also called the predictor variables, and Y is the dependent variable also called the response variable. When the pre-knowledge of the dataset (Y, X) is known, the future value of Y can be predicted whenever there are some variations in X. The accuracy of such prediction, by and large, depends on the total number of predictor variables involved. This is because, the larger the number, the better the prediction. But this in turn increases the complexity of the regression equation. Problem-solving involving such equation becomes extremely complex when the appropriate computer programming language is not employed, let alone doing it manually; and this culminates in delayed results for quick decision-making in a competitive business world.
Objectives: This work shows how the R-programming language can be written for automating a multi-linear regression model for faster processing and quicker decision-making.
Methods: A multi-linear regression equation was formulated from a sample dataset (Y, X) containing 28 values about the market sales of an establishment. The spreadsheet software, Ms-Excel, was used to store the dataset as 'comma separated values' (CSV) on a hard disk of a local computer. The R-programming function, 'read.csv', was used to read the dataset from the computer. Another R-programming function, cor(), was used to check the dataset for linearity. Finally, the coefficients of the formulated regression equation was determined using the R-function, lm().
Results: It took the computer less than 5 seconds to determine the coefficients of the multi-linear regression equation involving 5 predictor variables (x1, x2, x3, x4, and x5) for the response variable, Y, and whose dataset (Y, X) contained 28 values. Predictions about the response variable, Y, for arbitrarily values of market forces involving the predictor variable, X, were easily performed with the computer for quicker and better decision making.
Conclusions: Data processing tasks involving multi-linear regression equations would be snail-slow as well as a clog to quick-decision making in a competitive business environment if a proper computer programming language such as 'R' were not employed.

Keywords: Multi-Linear regression, problem-solving, decision-making, dataset, R-programming

INTRODUCTION

According to AI DATA (2021), every data scientist will likely have to perform Linear Regression tasks and predictive modeling processes at some point in their studies or career. For instance, the growth of a business (which can be denoted by **Y**) can be predicted from the set of values: X_1, X_2, X_3, …, X_n which represent the various variables or market forces affecting the growth of the business by establishing the linear relationships between the dataset (Y, X) through regression analysis, and then fitting a line to the observed data so that one can estimate (or predict) how Y changes as X changes, (Rebecca Bevans, 2022). Regression Analysis is the basic and commonly used technique for

predictive analysis. It is a statistical approach for modeling the linear relationship between a dependent variable (Y) and a given set of independent variables (X_i), (GeeksforGeeks, 2022).

The complexity of a regression model, however, increases as the number of variables in X increases. This in turn increases the complexity of its solution which resultantly constitutes some delay in the computed value of Y for quick decision making. Nevertheless, the **R** programming language is one of the world's acclaimed computer programming languages for solving statistical problems very fast. The Upskill Development Institute (2022) attested to this by stating that "the R language is widely used among statisticians and data miners for developing statistical data analysis. It has become the de-facto standard for writing statistical software among statisticians". The R programming language was therefore adopted in this work as one of the best tools that can be applied to a regression model (no matter how complex it may be) for producing results very quickly for faster decision making. The flexibility and intuitiveness of using R programming language for solving statistical problems, such as Regression Analysis, by data scientists is unmatched by other programming languages such as C, C++, JAVA, etc.

Fast decision making is very paramount in business so as to quickly respond to customers' demands, and remain very profitable and competitive in business (Sutevski, 2022). This view was held high by Brand Tracking (2020) by stating that in daily market trading, quick decisions have to be made by managers, otherwise the business profit values would dwindle and lead to loss, thereby making the business enterprise lose the market to other competitors.

The "programmability of the computer" in solving all kinds of data processing problems is the most important attribute of a computer. There are hundreds of high level programming languages that can be used to solve various data processing tasks. However, some of these programming languages are more efficient than others in solving specific problems. For instance, **Python** and **R** are the two best state-of-the-art open source programming languages that are widely used for statistical analysis or machine learning projects (Andra, 2022).

According to Techvidran Team (2022), some of the features of R programming language are: (i) it is an *open-source* software which is free of cost, and can be adjusted by adding some packages to it for additional functionalities, and then adapted according to the user's requirements; (ii) it possesses strong graphical capabilities; (iii) it has more than 10,000 different packages and extensions that help solve all sorts of problems in data science; (iv) it has a comprehensive development environment for statistical computing as well as for software development; (v) it can be used to perform both simple and complex mathematical and statistical calculations on data objects of a wide variety; (vi) it can handle a variety of structured and unstructured data; (vii) it is cross-platform compatible (i.e. it can be used on many different operating systems); (viii) it is compatible with other programming languages like JAVA, .NET, PYTHON, C, C++, and FORTRAN. All these features of R programming language enable it to be used for solving statistical problems, and producing intuitive results very fast for good decision making.

R programming language has a repertoire of functions for handling Regression Analysis. For instance, the function **lm()** which means "Linear Model" is used to create a linear regression model between the response variable (Y) and the predictor variable (X). Similarly, the function **cor()** is used to calculate the correlation coefficient for the x-data vector and the y-data vector. Furthermore, the function **summary()** is used to obtain more detailed results for the linear regression such as, *coefficients*, *residuals*, *intercept*, *standard error*, etc. Again, the function **predict()** is used to generate a prediction for the linear regression model. More of such linear regression functions are discussed extensively by openstax (2022).

The steps for using R programming in conducting a multiple linear regression model are outlined as follows by tutorialspoint (2022) and datatofish.com (2022): (i) use MS-Excel to store the data gathered from the observed values of the predictor variables (X) and the response variable (Y) of your experiment; (ii) use the R function **read.csv** to read the values from MS-Excel; (iii) check the data for linearity between the response variable (Y) and each of the predictor variables (X_i, i=1 to n); (iv) create a relationship model between X and Y by using the **lm()** function; (v) find the coefficients from the model and the average error in the prediction by using the **summary()** function; (vi) use the **predict()** function to predict the value of Y for new values of X.

The following "Problem Solving Steps" by University Human Resources (2022) can be used as a working guideline for businesses in order to remain very competitive and profitable: (i) define the business problem (i.e. define the response variable, Y), (ii) determine the root causes of the problem (i.e. define the predictor variables, X_i), (iii) define the goals (i.e. establish the optimal value of Y that can be obtained from X), (iii) develop action plan (i.e. set up a regression equation involving the dataset (Y, X)), (iv) execute action plan (i.e. solve the regression equation using the R programming technique, for instance), (iv) evaluate the results (i.e. compare the predicted results with actual values).

2 METHODOLOGY

A case study of the dataset provided by All Greens Franchise (2022) was used by the researcher to illustrate how R-programming language can be deployed in solving multiple linear regression problems very quickly for prompt decision making.

The dataset contains information about the "annual net sales of a Franchise store" based on the following factors:

X1 = number of square feet of various districts of the store in thousands (1,000)

X2 = inventory of the store in thousands of dollars ($1,000)

X3 = amount spent on advertisement in thousands of dollars ($1,000)

X4 = number of sales district per 1000 families

X5 = number of competing stores in the district

The dataset is shown in the following Table 1.

Step 1: a multiple linear regression equation was formulated from Table 1 whereby,

Y = annual net sales = response variable

Xi, i = 1 to 5 = predictor variables (whose descriptions are defined in the table)

Step 2: The multiple linear regression equation was executed using R-programming language

Step 3: a sample of 5 arbitrarily values of Xi was selected to predict the value of Y for decision making.

3 DATA ANALYSIS

A multiple linear regression equation is given as,

$$Y' = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots + \beta_p X_p + \epsilon \quad \textbf{(1)}$$

Where, _____ **Y′** is the predicted value of the response variable

_____ $\mathbf{X_i}$ (i = 1 to p) are the independent or predictor variables used for predicting Y′

_____ $\boldsymbol{\beta_0}$ is the intercept on Y′. It is the estimated value of Y′ when all the predictor variables are equal to zero (that is, $X_1 = X_2 = X_3 = \ldots = X_p = 0$). In other words, β_0 is the estimated value of Y′ when all the predictor variables are ignored (or not taken into consideration).

Table 1. All Greens Franchise store

s/n	Y =annual_sales	X1=shop_sqrft	X2=inventory	X3=adverts	X4=sales_district	X5=competitors
1	231	3	294	8.2	8.2	11
2	156	2.2	232	6.9	4.1	12
3	10	0.5	149	3	4.3	15
4	519	5.5	600	12	16.1	1
5	437	4.4	567	10.6	14.1	5
6	487	4.8	571	11.8	12.7	4
7	299	3.09	512	8.1	10.1	10
8	195	2.5	347	7.7	8.4	12
9	20	1.2	212	3.3	2.1	15
10	68	0.6	102	4.9	4.7	8
11	570	5.4	788	17.4	12.3	1
12	428	4.2	577	10.5	14	7
13	464	4.7	535	11.3	15	3
14	15	0.6	163	2.5	2.5	14
15	65	1.2	168	4.7	3.3	11
16	98	1.6	151	4.6	2.7	10
17	398	4.3	342	5.5	16	4
18	161	2.6	196	7.2	6.3	13
19	397	3.8	453	10.4	13.9	7
20	497	5.3	518	11.5	16.2	1
21	528	5.6	615	12.3	16	0
22	99	0.8	278	2.8	6.5	14
23	0.5	1.1	142	3.1	1.6	12
24	347	3.6	461	9.6	11.3	6
25	341	3.5	382	9.8	11.5	5
26	507	5.09	590	12	15.7	0
27	400	8.6	517	7	12	8
28	231	3	294	8.2	8.2	11

source: https://college.cengage.com/mathematics/brase/understandable_statistics/7e/students/datasets/mlr/frames/mlr05.html

_____ $\boldsymbol{\beta_i}$, (i = 1 to p) are the regression coefficients used in determining the change in Y′ that corresponds to one unit change in X_i when all the other predictor

variables are held constant. For instance, β_1 is the change in Y′ to one unit change in X_1 when X_i (i = 2 to p) are held constant.

_____ ϵ is the error term incurred in the predicted value of Y′.

Now, equation (1) can be rewritten as follows for solving the regression problem of Table 1 that involves 5 predictor variables:

$$Y' = \beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_3 + \beta_4X_4 + \beta_5X_5 + \epsilon \quad \text{..... (2)}$$

Where:

$\mathbf{Y'}$ = predicted annual_sales

$\mathbf{X_1}$ = shop_sqrft

$\mathbf{X_2}$ = inventory

$\mathbf{X_3}$ = adverts

$\mathbf{X_4}$ = sales_district

$\mathbf{X_5}$ = competitors

In other words, equation (2) can finally appear thus:

$$annual\_sales' = \beta_0 + \beta_1(shop\_sqrft) + \beta_2(inventory) + \beta_3(adverts) + \beta_4(sales\_district) + \beta_5(competitors) + \epsilon \quad \textbf{(3)}$$

The steps taken by the researcher in solving the regression model of equation (3) are as follows:
- **Step 1:** Using MS-Excel to capture the data of Table 1

The data were captured in MS-Excel, and saved as "Comma Separated Values (CSV)" on a hard disk with the filename, "salesdata.csv"

Step 2: Reading the stored data with the use of the R function, "read.csv"

The R-programming environment was launched on a PC, and used to read the data file, "salesdata.csv" from the hard disk to the computer's memory with the use of the **read.csv** function

Step 3: Checking the dataset for Linearity

The regression equation is a linear equation; therefore the values in the dataset (Y, X) must be linearly related otherwise the computed values from the regression model would not be reliable. A check for linearity of the dataset (Y, X) can be done with the use of a **scatterplot** graph or correlation coefficient function **cor()**. The correlation coefficient technique was adopted in this work. The value of the correlation coefficient, r, lies between -1 and +1. Table 2 shows the value as well as the strength of such linear relationship.

Table 3 shows the correlation coefficients between the variables X and Y of Table1 which was calculated through the use of the R-programming function, **cor()** in Appendix A.

Note that the use of the *cor()* function has immensely simplified the tedious task of writing an algorithm for calculating the *correlation coefficient (r)* which is given by the formula:

Correlation coefficient,

$$r = \frac{n\Sigma xy - (\Sigma x)(\Sigma y)}{\sqrt{[n\sum(x^2 - (\Sigma x)^2][n\sum y^2 - (\Sigma y)^2)]}}$$

Step 4: determining the coefficients of the regression model by using the **lm()** function

In order to evaluate the multiple linear regression model of equation (3), the coefficients: β_0, β_1, β_2, β_3, β_4, and β_5 must first of all be determined. The formula for determining each of the β_i,(i = 1 to n), is given by the formula,

$$\beta_i = [\sum XY - nX'Y'] / [\sum X^2 - n X'^2] \quad \textbf{(4)}$$

where,

$X' = (\sum X) / n$ and $Y' = (\sum Y) / n$

Similarly, the constant, β_0 is given by the formula, $\beta_0 = Y' - \beta_1 X'$ **(5)**

Evaluating the equations in (4) and (5) requires a good knowledge of 'looping' in computer programming, or alternatively a good knowledge of matrices in mathematics. But the *lm()* function in R-programming handles such huge task very easily without bothering the user about any of these basic knowledge.

The computed values of the coefficients: β_0, β_1, β_2, β_3, β_4, and β_5 through the use of the *lm()* function in Appendix A are shown in Table 4.

Step 5: State the Final Regression Formula

Substituting the computed regression coefficients of Table 4 into the multiple linear regression model of equation (3), we have:

Table 2. Scale of correlation coefficient and the strength of its relationship

Negative Correlation value	Positive Correlation value	Strength of Relationship
- 0.19 ≤ r ≤ 0	0 ≤ r ≤ 0.19	Very low correlation
- 0.39 ≤ r ≤ - 0.2	0.2 ≤ r ≤ 0.39	Low correlation
- 0.59 ≤ r ≤ - 0.4	0.4 ≤ r ≤ 0.59	Moderate correlation
- 0.79 ≤ r ≤ - 0.6	0.6 ≤ r ≤ 0.79	High correlation
- 1.0 ≤ r ≤ - 0.8	0.8 ≤ r ≤ 1.0	Very high correlation

Source: https://www.researchgate.net/publication/345693737_Employee_Productivity_in_Malaysian_Private_Higher_Educational_Institutions/figures?lo=1

Table 3. correlation coefficients between the dataset (Y,X) of table 1

Y	X				
	shop_sqrft	inventory	adverts	sales_district	competitors
annual_sales	0.8939057	0.9450971	0.9123125	0.9539178	-0.9105040
	Very high correlation	Very high correlation	Very high correlation	Very high correlation	Very high correlation

Table 4. Computed regression coefficients

β_0	β_1	β_2	β_3	β_4	β_5
-19.67944	16.25812	0.17194	11.63356	13.63818	-5.26146

$$annual\_sales' = -19.67944 + 16.25812(shop\_sqrft) + 0.17194(inventory) + 11.63356(adverts) + 13.63818(sales\_district) - 5.26146(competitors) + \epsilon \quad (6)$$

Equation (6) can now be used for making predictions about "annual_sales" of the **franchise store** for any changes in the five variables: *shop_sqrft, inventory, adverts, sales_district, and competitors*.

4 RESULTS AND DISCUSSION

The regression coefficients of equation (6) enable us to determine the magnitude effect of each of the predictor variables: *shop_sqrft, inventory, adverts, sales_district, and competitors* on the response variable, "annual_sales", as discussed below.

For every 1% increase in *shop_sqrft*, there is an associated 16.25812% increase in *annual_sales*

For every 1% increase in *inventory*, there is an associated 0.17194% increase in *annual_sales*

For every 1% increase in *adverts*, there is an associated 11.63356% increase in *annual_sales*

For every 1% increase in *sales_district*, there is an associated 13.63818% increase in *annual_sales*

For every 1% increase in *competitors*, there is an associated 5.26146% decrease in *annual_sales*

Lastly, if all the predictor variables are ignored (or equal to zero), there is an associated 19.67944% decrease in *annual_sales*

The above calculated regression coefficients are of immense help for good decision making.

For instance, suppose we have the following

values: *shop_sqrft = 2.6, inventory = 196,, adverts = 7.2, sales_district = 6.3, and competitors = 13*. Equation (6) will be evaluated as follows:

annual_sales´ = -19.67944 + 16.25812(2.6) + 0.17194 (196) + 11.63356(7.2) +

13.63818(6.3) - 5.26146(13)

= **157.575**

Now, suppose the store increases its ***adverts*** from 7.2 thousand dollars to 9 thousand dollars, and also increases its ***sales_district*** from 6.3 to 10. The predicted ***annual_sales*** of the store would now be:

annual_sales´ = -19.67944 + 16.25812(2.6) + 0.17194 (196) + 11.63356(9) +

13.63818(10) - 5.26146(13)

= **228.977** (which is an increase of about **45%** from its previous annual sales of **157.575**)

Such important predictions for good decision making can be quickly made with the use of **predict()** function of R-programming language.

Appendix A contains the complete R-program for the implementation of the multiple linear regression problem of this work. The program could easily be adopted by any user for solving any manner of regression problems very quickly.

5 CONCLUSION

Problem-solving involving multi-linear regression equation is usually very tasking and time-consuming if a computer programming approach is not employed. The R-programming language is one of the best computer programming languages for solving statistical problems. It was therefore applied in this work for solving multi-linear regression problem (no matter how complex it may be) effortlessly. Results showed that it took the computer less than 5 seconds to determine the coefficients of a given multi-linear regression problem involving 5 predictor variables for the value of Y whose dataset (Y, X) contained 28 values. Ordinarily such arduous task would have taken a couple of minutes or even hours to solve with a calculator.

The use of R-programming language for statistical analysis is therefore highly invaluable for quick decision-making especially in a very competitive business environment.

REFERENCES

AI DATA (2021), "10 Open datasets for linear regression", Retrieved from: https://www.telusinternational.com/articles/10-open-datasets-for-linear-regression (Accessed June 5, 2022)

All Greens Franchise dataset (2022), Available at: https://college.cengage.com/mathematics/brase/understandable_statistics/7e/students/datasets/mlr/frames/mlr05.html (Accessed June 10, 2022)

Andra (2022), "Top 5 Statistical Programming Languages in demand (2022)", Retrieved from: https://dataresident.com/statistical-programming-languages/ (Accessed June 10, 2022)

Brand Tracking (2020), "Why Quick decision-making is your competitive advantage", Retrieved from: https://www.askattest.com/blog/articles/why-quick-decision-making-is-your-competitive-advantage (Accessed June 10, 2022)

Datatofish.com (2020), "Example of Multiple Linear Regression in R", Retrieved from: https://datatofish.com/multiple-linear-regression-in-r/ (Accessed June 11, 2022)

Dragon Sutevski (2022), "Fast Decision-Making Process Vs Quality Decisions", Retrieved from: https://www.entrepreneurshipinabox.com/1084/importance-quick-decisions/ (Accessed June 9, 2022)

Geeksforgeeks (2022), "R tutorial", Available at: https://www.geeksforgeeks.org/r-tutorial/?ref=lbp (Accessed June 5, 2022)

Openstax (2022), "Use of R Statistical Analysis Tool for Regression Analysis", Retrieved from: https://openstax.org/books/principles-finance/pages/14-6-use-of-r-statistical-analysis-tool-for-regression-analysis (Accessed June 11, 2022)

Rebecca Bevans(2022), "Multiple Linear Regression-A Quick Guide(Examples)", Retrieved from: https://www.scribbr.com/statistics/multiple-linear-regression/ (Accessed June 8, 2022)

TechVidvan Team (2019), "15 features of R programming you can't afford to overlook", Retrieved from: https://www.google.com/amp/s/techvidvan.com/tutorials/r-features/%3famp=1 (Accessed June 7, 2022)

Tutorialspoint (2022), "R-Linear Regression", Retrieved from: https://www.tutorialspoint.com/r/r_linear_regression.htm (Accessed June 10, 2022)

Upskill Development Institute (2022), "Data Management and Statistical Data Analysis Using R course", Retrieved from: https://upskilldevelopment.com/data-management-and-statistical-data-analysis-using-r-course (Accessed June 9, 2022)

University Human Resources (2022), "8-Step Problem Solving Process", Retrieved from: https://hr.uiowa.edu/development/organizational-development/lean/8-step-problem-solving-process (Accessed June 8, 2022)

APPENDIX A (The R-program)

========== THIS IS THE R-PROGRAM ==========

This program shows how to calculate a multiple linear regression model very fast

Using R-programming language

The multiple linear regression model is of the form:

$Y' = \beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_3 + \beta_4X_4 + \beta_5X_5 + \epsilon$

where,

Y' = predicted annual_sales

X_1 = shop_sqrft

X_2 = inventory

X_3 = adverts

X_4 = sales_district

X_5 = competitors

The values for the dataset(Y,X) are shown in table1

#STEP 1: Perform initial settings

```
setwd("C:\\myRprojects")       #This is the working directory of the project

dataset <- "C:\\myRprojects\\myRfiles\\salesdata.csv"
#This is the name of the Excel file that contains the dataset(Y,X)
```

#STEP 2: Use the function read.csv() to read the dataset

```
theData <- read.csv(dataset)    #reads the dataset

theData                         #displays all the contents of the dataset(Y,X) on the screen
```

#STEP 3: Check the dataset for linearity using the correlation function, cor()

```
correlationValue <- cor(theData)
     #determines the correlation coefficient for dataset linearity

print(correlationValue)                    #displays the correlation between the dataset(Y,X) on the screen
```

#Alternatively, you can use a scatterplot to check for data linearity

```
#Example: checking for data linearity between "annual_sales" and "shop_sqrft"

x <- theData$shop_sqrft

y <- theData$annual_sales
```

```
plot(x,y,main = "Scatterplot between annual_sales and
shop_sqrft", xlab = "shop_sqrft", ylab = "annual_sales")
                        #plot the graph

abline(lm(annual_sales~shop_sqrft,data=theData))
            #draw the regression line in the graph


#Similar graphical checks for data linearity for the re-
maining values in the dataset(Y,X) can equally be per-
formed, such as:

# "annual_sales" vs "inventory", "annual_sales" vs
"adverts", "annual_sales" vs "sales_district", and
"annual_sales" vs "competitors"
```

#STEP 4: Use the lm() function to determine the coeffi-cients of the regression model

```
theLinearModel <- lm
(annu-
al_sales~shop_sqrft+inventory+adverts+sales_district+c
ompetitors, data=theData)

summary(theLinearModel)                    #view the
LinearModel summary
```

#STEP 5: Supply new set of data for making predictions

```
newData <- data.frame(shop_sqrft = c(2.6), inventory = c
(196), adverts = c(9), sales_district = c(10), competitors =
c(13))

predict(theLinearModel, newData)

==========================================
==================================
```