**Designing a Linear Model (May 2019)**

**Deriving a good estimator for the parameters a and b based on a sample of  
observed ​(​x​; ​y​)​ pairs**

Haleh Azizi

[haleh.azizi@ndus.edu](mailto:haleh.azizi@ndus.edu)

713-254-9891

**1. Introduction**

Regression analysis is a popular statistical technique, useful for modelling the relationship between a response variable and explanatory variables. Three assumptions were considered for designing a linear model:

* Relationship between inputs and output might be linear,
* Correlation between errors must be equal to zero: . This means that residuals might be independent,
* means independency between variance and inputs.

where and are intercept and slope respectively (Figure 1). and are also input and output respectively.

|  |
| --- |
| Error ()  Intercept () |
| Figure 1. Linear regression over dataset data\_1\_2.cvs and its parameters. |

Based on the problem statement the noise () distribution has conditional mean zero given ​*x*​, therefore, a modification must be done on the formulation of linear regression, is described as follows:

**2. Heteroskedasticity**

Heteroskedasticity occurs when third mentioned assumption for designing a linear regression was not applied, therefore, the variance for all observations in a data set are not the same. Heteroskedasticity, must be detected and corrected. Different methods were proposed for this purpose [1], weighted least squares regression is used in current study [2].

Suppose that linear model is as follows:

|  |  |
| --- | --- |
|  | (1) |

All parameters were defined earlier. Parameters could be written as matrix format [1-2]:

|  |  |
| --- | --- |
|  | (2) |

|  |  |
| --- | --- |
|  | (3) |

|  |  |
| --- | --- |
|  | (4) |

|  |  |
| --- | --- |
|  | (5) |

Therefore, equation 1 might be re-written as follows:

|  |  |
| --- | --- |
|  | (6) |

Sum of squared error, which is a popular loss function, is used:

|  |  |
| --- | --- |
|  | (7) |

where:

|  |  |
| --- | --- |
|  | (8) |

In this case optimum of Ordinary Least Square (OLS) which minimize error (Eq. 7) can be calculated as follows:

|  |  |
| --- | --- |
|  | (9) |

where is transpose matrix of . In situation occurrence of heteroscedasticity, errors will not be independent, while are function of *x*, therefore, an equation of errors must be approximated (). Based on literature [3], is equivalent with estimation variance of dataset:

|  |  |
| --- | --- |
|  | (10) |

Yobero suggests that inverse of might be the weight for modification of Eq. 9 during heteroscedasticity [2]:

|  |  |
| --- | --- |
|  | (11) |

where is a zero arrays matrix, its diagonal arrays are :

|  |  |
| --- | --- |
|  | (12) |

of Weighted Least Square (WLS) can be approximated using Eq. 13:

|  |  |
| --- | --- |
|  | (13) |

Equations 9 and 13 were the loss functions used to optimize intercept () and slope () in current study.

**3. Statistical Overview**

Five dataset were assigned to be studied. In Figure 2, probability density function (PDF) of X and Y for all datasets as well as their cross correlations are plotted. PDFs show that in majority of cases distributions are far from Gaussian, also correlation between Xs and Ys are not linear. Weak correlation coefficient (CC) between X and Y in Figure 3, shows that maximum CC is 0.63 for data\_1\_2.cvs, confirms low correlation between data, accordingly, weak linear model might be anticipated.

|  |  |
| --- | --- |
| data\_1\_1.cvs | data\_1\_2.cvs |
| data\_1\_3.cvs | data\_1\_4.cvs |
| data\_1\_5.cvs | Figure 2. Probability density function and cross correlation of X and Y for five studied dataset. |

|  |  |
| --- | --- |
| data\_1\_1.cvs | data\_1\_2.cvs |
| data\_1\_3.cvs | data\_1\_4.cvs |
| data\_1\_5.cvs | Figure 3. Correlation coefficient of X and Y for five studied dataset. |

**4. Linear Regression using OLS Approach**

In OLS approach, intercept and slope were optimized in order to minimize error of loss function displayed in Eq. 9. Required codes were developed in Python (Appendix 1 and attached code). Resulted linear models, their equations and corresponding R2 are displayed in Figure 4. As it can be seen in Figure 4, the best model belongs to data\_1\_2.cvs with R2 equal to 0.398, which shows low accuracy. In Figure 5, scatter plots between Y and estimated Y were displayed, which again confirms low correlation. PDF of the subtraction of estimated Y from Y for all datasets are plotted in Figure 6. These PDFs as well as all related figures (Figures 4 and 5) show that the main problem depends to boundary data which are so similar to outliers.

As a result, it seems heteroskedasticity has occurred here. To investigation of heteroscedasticity, errors against X are plotted in Figure 7, which shows high errors for boundary data. Therefore, a second order polynomial were fitted to the errors, and corresponding equations are displayed in Figure 7. The equations are in fact in Eq. 10. Consequently, weighted least square formulations (Eqs. 10 to 13) were applied to define a more reliable linear regression, their results are presented in next section.

|  |  |
| --- | --- |
| y = 0.881 x - 0.269  R2 = 0.164  SSE = 2803.67  data\_1\_1.cvs | y = 1.285 x - 0.395  R2 = 0.398  SSE = 23789.31  data\_1\_2.cvs |
| y = -0.538 x + 4.175  R2 = 0.103  SSE = 15344.05  data\_1\_3.cvs | y = 0.633 x + 1.148  R2 = 0.027  SSE = 1286.90  data\_1\_4.cvs |
| y = -1.333 x - 0.241  R2 = 0.280  SSE = 6163.08  data\_1\_5.cvs | Figure 4. Linear model using OLS approach, and corresponding equations, R2, and sum of squared error (SSE) for five studied dataset. |

|  |  |
| --- | --- |
| data\_1\_1.cvs | data\_1\_2.cvs |
| data\_1\_3.cvs | data\_1\_4.cvs |
| data\_1\_5.cvs | Figure 5. Scatter plot between Y and estimated Y for five studied dataset. |

|  |  |
| --- | --- |
| data\_1\_1.cvs | data\_1\_2.cvs |
| data\_1\_3.cvs | data\_1\_4.cvs |
| data\_1\_5.cvs | Figure 6. PDF of errors of OLS for five studied dataset. |

|  |  |
| --- | --- |
| data\_1\_1.cvs[[1]](#footnote-1) | data\_1\_2.cvs |
| data\_1\_3.cvs | data\_1\_4.cvs |
| data\_1\_5.cvs | Figure 7. Fitting second order polynomial model over squared errors of OLS for five studied dataset, and corresponding equation. |

**5. Linear Regression using WLS Approach**

SE equations in Figure 7 are in Eq. 10. Eqs. 10 to 13 were applied to define weighted least square (WLS). In Figure 8, resulted linear models as well as corresponded R2 and SSE are presented. Comparison between results of OLS (Figure 4) and WLS (Figure 8), shows highly similarity between them.

|  |  |
| --- | --- |
| y = 1.004 x + 0.050  R2 = 0.157  SSE = 2826.10  data\_1\_1.cvs | y = 1.286 x - 0.339  R2 = 0.398  SSE = 23789.62  data\_1\_2.cvs |
| y = -0.627 x + 1.473  R2 = 0.079  SSE = 15747.99  data\_1\_3.cvs | y = 0.525 x + 1.034  R2 = 0.022  SSE = 1294.47  data\_1\_4.cvs |
| y = -0.907 x + 0.16  R2 = 0.231  SSE = 6584.49  data\_1\_5.cvs | Figure 8. Linear model using WLS approach, and corresponding equations, R2, and sum of squared error (SSE) for five studied dataset. |

**5. Linear Regression using Semi-WLS Approach**

Following to results of WLS, a modification were applied on the Eqs. 11 to 13. In current approach, weights were defined as follows:

|  |  |
| --- | --- |
|  | (14) |

Then multiplied with X and Y of the dataset. Eq. 9 were applied over weighted dataset, and results are displayed in Figure 9. The problem was that for datasets data\_1\_4.cvs, was rarely less than zero, therefore, no results were achieved for them. In other datasets, SSE and R2 of the semi-WLS were respectively higher and lower than both WLS and OLS. As a result, this modification is not acceptable.

|  |  |
| --- | --- |
| y = 0.987 x – 0.019  R2 = 0.159  SSE = 2819.04  data\_1\_1.cvs | y = 1.282 x - 0.023  R2 = 0.398  SSE = 23803.20  data\_1\_2.cvs |
| y = -0.643 x + 0.203  R2 = 0.054  SSE = 16183.53  data\_1\_3.cvs | y = ? x + ?  R2 = ?  SSE = ?  data\_1\_4.cvs |
| y = -0.618 x + 0.247  R2 = 0.134  SSE = 7415.49  data\_1\_5.cvs | Figure 9. Linear model using semi-WLS approach, and corresponding equations, R2, and sum of squared error (SSE) for five studied dataset. |

**6. Conclusion**

Three approaches entitled OLS, WLS and semi-WLS were applied to model a linear regression over five studied datasets. In Table 1, the abstracted results of all three methods were displayed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table 1. Abstracted results of three linear regressions methods. | | | | |
| Dataset | Method | Equation | SSE | R2 |
| data\_1\_1.cvs | Polynomial regression over squared errors\* |  | --- | --- |
| OLS | y = 0.881 x - 0.269 | 2803.67 | 0.164 |
| WLS | y = 1.004 x + 0.050 | 2826.10 | 0.157 |
| Semi-WLS | y = 0.987 x – 0.019 | 2819.04 | 0.159 |
| data\_1\_2.cvs | Polynomial regression over squared errors |  | --- | --- |
| OLS | y = 1.285 x - 0.395 | 23789.31 | 0.398 |
| WLS | y = 1.286 x - 0.339 | 23789.62 | 0.398 |
| Semi-WLS | y = 1.282 x - 0.023 | 23803.20 | 0.398 |
| data\_1\_3.cvs | Polynomial regression over squared errors |  | --- | --- |
| OLS | y = -0.538 x + 4.175 | 15344.05 | 0.103 |
| WLS | y = -0.627 x + 1.473 | 15747.99 | 0.079 |
| Semi-WLS | y = -0.643 x + 0.203 | 16183.53 | 0.054 |
| data\_1\_4.cvs | Polynomial regression over squared errors |  | --- | --- |
| OLS | y = 0.633 x + 1.148 | 1286.90 | 0.027 |
| WLS | y = 0.525 x + 1.034 | 1294.47 | 0.022 |
| Semi-WLS |  |  |  |
| data\_1\_5.cvs | Polynomial regression over squared errors |  | --- | --- |
| OLS | y = -1.333 x - 0.241 | 6163.08 | 0.280 |
| WLS | y = -0.907 x + 0.160 | 6584.49 | 0.231 |
| Semi-WLS | y = -0.618 x + 0.247 | 7415.49 | 0.134 |
| In this case, because on an unknown reason, the polynomial equation achieved of python was incorrect, therefore, just for this dataset, the equation calculated using R. | | | | |

Based on the literature, because of the dependency of errors (residuals) of linear regression to the variance given x, it was anticipated that weighted regression might be better than ordinary ones to estimates of y. Results of WLS and OLS are similar (Table 1), therefore, results of both methods were reported in Excel file: OLS is simple, and WLS is more general.

**References:**

[1] Atkinson A.C., Riani M., Torti F., Robust methods for heteroskedastic regression, Computational Statistics and Data Analysis **104** (2016) 209–222.

[2] <https://rpubs.com/cyobero/187387>

# [3] Klein A.G., Gerhard C., Buchner R., Diestel S., [Schermelleh-Engel](https://www.researchgate.net/profile/Karin_Schermelleh-Engel) K., The Detection of Heteroscedasticity in Regression Models for Psychological Data,  [Psychological Test and Assessment Modeling](https://www.researchgate.net/journal/2190-0493_Psychological_Test_and_Assessment_Modeling) 58(4) (2016) 567-592.

**Appendix 1**

**Developed Scripts**

from statistics import mean

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from matplotlib import style

import seaborn as sns

style.use('fivethirtyeight')

data\_1 = pd.read\_csv('F:\Job Application\ data\_1\_3.cvs.csv')

X = data\_1['x']

Y = data\_1['y']

sns.pairplot(data\_1) # Distribution od Dataset

plt.show()

sns.heatmap(data\_1.corr(),annot=True) # Correlation Between Variables

plt.show()

def squer\_error(Y\_orig,Y\_line): # Defining the Loss Function

return sum((Y\_line - Y\_orig)\*\*2) # Sum of Squared Error (Loss Function)

def r\_squer(Y\_orig,Y\_line): # Defining R Squared Value

Y\_mean\_line = [mean(Y\_orig)]

squered\_error\_regr = squer\_error(Y\_orig,Y\_line)

squered\_error\_mean = squer\_error(Y\_orig,Y\_mean\_line)

return 1 - (squered\_error\_regr/squered\_error\_mean)

def a\_b(x,y): # Calculation (Optimization) of a (Slope) and b (Intercept)

a = (((mean(x)\*mean(y)) - mean(x\*y))/

((mean(x)\*\*2)-mean(x\*\*2))) # Slope

b = mean(y) - a\* mean(x) # Intercept

return a , b

a , b = a\_b(X,Y)

linear\_reg = [(a\*x)+b for x in X]

r = r\_squer(Y,linear\_reg)

print('\n\n',r)

SSE = squer\_error(Y,linear\_reg)

print('\n\n',SSE)

plt.scatter(X,Y,c = 'b') # Plotting Dataset

plt.plot(X,linear\_reg) # Plotting the Linnear Regression

plt.title('Linear Model',fontsize=15)

plt.xlabel(print('Y = {a} X + {b}' .format (a = a, b = b)))

plt.xlabel('X',fontsize=12)

plt.ylabel('Y',fontsize=12)

plt.show()

plt.scatter(Y,linear\_reg) # Scatter Plot Between Y and Predicted Y

plt.title('Scatter Plot Between Y and Predicted Y',fontsize=15)

plt.xlabel('Y',fontsize=12)

plt.ylabel('Predicted Y',fontsize=12)

plt.show()

# Calculation of Residuals, Noise which is Conditioned to X

residuals = list(np.array(Y) - np.array(linear\_reg))

sns.distplot((residuals))

plt.title('PDF of Error of LM',fontsize=15)

plt.xlabel('Residuals',fontsize=12)

plt.ylabel('Frequency',fontsize=12)

plt.show()

# Second Degree Polynomial Modeling of Residuals, Noises which are Conditioned to X

degree = 2

sqrresiduals = list((np.array(Y) - np.array(linear\_reg))\*\*2)

weights = np.polyfit(X, sqrresiduals, degree) # Weights of 2nd Degree P.M.

model = [(weights[0]\*x\*\*2+weights[1]\*x+weights[2]) for x in X]

plt.scatter(X,residuals) # Plotting the Residuals

plt.plot(X,model) # Plotting the P.M on Residuals

plt.title('Polynomial Model on Squared Residuals',fontsize=15)

plt.xlabel('X',fontsize=12)

plt.ylabel('Squared Residuals',fontsize=12)

plt.show()

# Defining a New Linear Regression Considering Polynomial Model over Residuals

# Detecting and Resolving Heteroskedasticity

# Defining the Matrix of X = (1,X)

matrix\_x = np.zeros((len(X),2))

x\_1 = np.ones((len(X),1))

x\_2 = np.array(X)

matrix\_x[:,:1] = x\_1

matrix\_x[:,1] = x\_2

matrix\_x

# Defining the Matrix of weights

matrix\_w = np.zeros((len(X),len(X)))

weight\_s = [1/k for k in model]

def replaceDiagonal(matrix, replacementList):

for i in range(len(replacementList)):

matrix[i][i] = replacementList[i]

replaceDiagonal(matrix\_w,weight\_s)

# Calculation of intercept and slope using the equation B=(inv(X'WX))X'WY

# B\_1 = inv(X'WX) and B\_2 = X'WY

B\_1 = np.linalg.inv(np.matmul(np.matmul(matrix\_x.transpose(), matrix\_w), matrix\_x))

B\_2 = np.matmul(np.matmul(matrix\_x.transpose(), matrix\_w), Y)

B = np.matmul(B\_1, B\_2)

linear\_reg\_final = [(B[1]\*x) + B[0] for x in X]

r = r\_squer(Y,linear\_reg\_final)

print('\n\n',r)

SSE = squer\_error(Y,linear\_reg\_final)

print('\n\n',SSE)

plt.scatter(X,Y,c = 'b') # Plotting Dataset

plt.plot(X,linear\_reg\_final) # Plotting the Modified Linnear Regression

plt.title('Final Modified Linear Model',fontsize=15)

plt.xlabel(print('Y = {a} X + {b}' .format (a = B[1], b = B[0])))

plt.xlabel('X',fontsize=12)

plt.ylabel('Y',fontsize=12)

plt.show()

# Defining a New Linear Regression Considering Polynomial Model over Residuals

# Detecting and Resolving Heteroskedasticity

#weight = [1/k for k in model]

weight = [1/(k\*\*0.5) for k in model]

Xnew = [m\*n for m,n in zip(X,weight)]

Ynew = [m\*n for m,n in zip(Y,weight)]

aopt , bopt = a\_b(np.array(Xnew),np.array(Ynew))

linear\_reg\_new = [(aopt\*x)+bopt for x in X]

r = r\_squer(Y,linear\_reg\_new)

print('\n\n',r)

SSE = squer\_error(Y,linear\_reg\_new)

print('\n\n',SSE)

plt.scatter(X,Y,c = 'b') # Plotting Dataset

plt.plot(X,linear\_reg\_new) # Plotting the Modified Linnear Regression

plt.title('Modified Linear Model',fontsize=15)

plt.xlabel(print('Y = {a} X + {b}' .format (a = aopt, b = bopt)))

plt.xlabel('X',fontsize=12)

plt.ylabel('Y',fontsize=12)

plt.show()

1. In this case, because on an unknown reason, the polynomial equation achieved of python was incorrect, therefore, just for this dataset, the equation calculated using R. [↑](#footnote-ref-1)