Linear Models with Non-Gaussian Noise (Documentation)

(Analysis, Code and Modelling by Abhimanu Kumar)

1 Exploratory Data Analysis (data properties, aberrations etc.)

dataset1 properties:

1. For dataset1 there are a total of 100 data points.
2. The summary statistics of the data for x column is: Min. : -9.96120, 1st Quadrant: -7.35047, Median : -0.26748, Mean : -0.05979, 3rd Quadrant: 6.86608, Max. : 9.91277.
3. The summary statistics of the data for y column is: Min. : -38.377, 1st Quadrant: -4.721, Median : 3.142 , Mean : 2.967 , 3rd Quadrant: 10.470 , Max. : 31.728.
4. A plot of the dataset1 with a linear model fit (red line) is shown in figure 1.

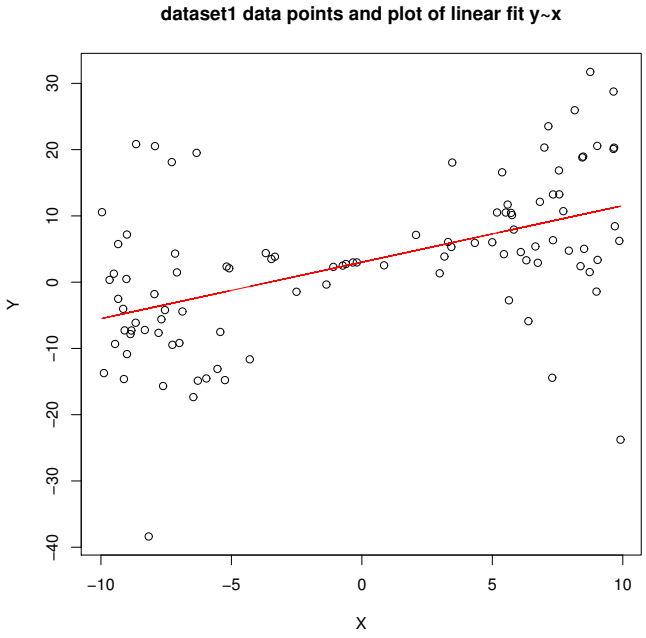

Figure 1: Exploratory Data Analysis for dataset1

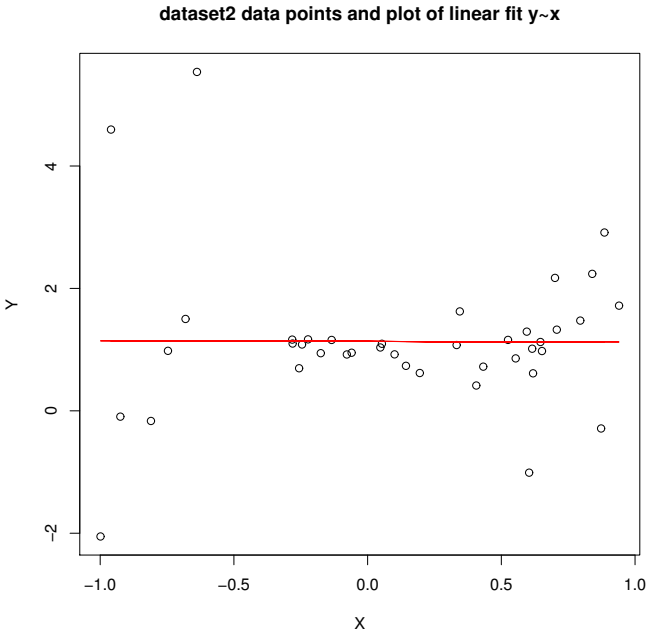

Figure 2: Exploratory Data Analysis for dataset2

dataset2 properties:

1. For dataset2 there are a total of 40 data points.
2. The summary statistics of the data for x column is: Min. : -0.9988 , 1st Qu.: -0.2480 , Median : 0.1692 , Mean : 0.1279 , 3rd Qu.: 0.6162 , Max. : 0.9402.
3. The summary statistics of the data for y column is: Min. :-2.0530 , 1st Qu.: 0.7318 , Median : 1.0549 , Mean : 1.1325 , 3rd Qu.: 1.3018 , Max. : 5.5372.
4. A plot of the dataset2 with a linear model fit (red line) is shown in figure 2.

Observations to be noted:

1. There is more noise associated with higher value of abs(x) as seen in figures 1 and 2 with respect to a linear model fit.
2. For dataset1 correlation between y and x is 0.5102851. Correlation between y and x after removing extreme points (remove points where y>20 or y<-20) is 0.549862. Even if we remove more "noisy points", the correlation between x and y doesnt improve as noted above for dataset1. The trend is similar for dataset2.
3. After fitting a linear model, i.e. $y\ \beta_1 x + \beta_0 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$, for dataset1 and dataset2, we observe that the residues (or error terms $Y - X\beta$) are non-gaussian. We perform 3 normality tests: 1) Jarque - Bera Normalality Test, 2) Shapiro - Wilk Normality Test , and 3) D'Agostino Normality Test.
   For the **dataset1** the results for normality tests are:
   Title: Jarque - Bera Normalality Test
   Test Results:
   STATISTIC:
   X-squared: 17.0869
   P VALUE:
   Asymptotic p Value: 0.0001948

   Title: Shapiro - Wilk Normality Test
   Test Results:

STATISTIC:
W: 0.9551
P VALUE:
0.001835

Title: D'Agostino Normality Test
Test Results:
STATISTIC:
Chi2 | Omnibus: 9.6091
Z3 | Skewness: -1.373
Z4 | Kurtosis: 2.7792
P VALUE:
Omnibus Test: 0.008192
Skewness Test: 0.1697
Kurtosis Test: 0.005449

For the **dataset2** the results for normality tests are as follows. Though given dataset2 has just 40 datapoints the normality tests are less reliable:
Title: Jarque - Bera Normalality Test
Test Results:
STATISTIC:
X-squared: 42.3964
P VALUE:
Asymptotic p Value: 6.219e-10

Title: Shapiro - Wilk Normality Test
Test Results:
STATISTIC:
W: 0.8157
P VALUE:
1.466e-05

Title: D'Agostino Normality Test
Test Results: STATISTIC:
Chi2 | Omnibus: 19.5922
Z3 | Skewness: 2.8855
Z4 | Kurtosis: 3.3565
P VALUE:
Omnibus Test: 5.567e-05
Skewness Test: 0.003908
Kurtosis Test: 0.0007894

All three normality tests for linear model residues for the datasets lead to rejecting the null hypothesis that residue has a Gaussian distribution.

4. Based on the first three observations assuming that the noise is proportional to input x is worth exploring.

2 Modeling Input Dependent Noise

As mentioned in problem definition we have

$$y = \beta_1 x + \beta_0 + \epsilon \tag{1}$$

In the usual linear modeling we assume that the noise ϵ is drawn from a gaussian distribution. But as observed earlier, modelling that the noise ϵ is dependent on the input x is worth exploring. The dependent variable y can be modelled as

$$\begin{aligned} y &= \beta_1 x + \beta_0 + \epsilon \\ where\ \epsilon &\sim x\mathcal{N}(0, \sigma_1^2) + \mathcal{N}(0, \sigma_2^2) \end{aligned} \tag{2}$$

This can also be written as

$$P(y/x) = \prod_{i}^{N} \frac{1}{\sqrt{2\pi\sigma_i^2}} \mathrm{e}^{\frac{-(y_i-\beta_1 x_i-\beta_0)}{2\sigma_i^2}}$$
$$where\ \sigma_i^2 = x_i^2\sigma_1^2 + \sigma_2^2 \tag{3}$$

The negative log likelihood is given as

$$L = \sum_{i=1}^{N} \frac{log(\sigma_i^2)}{2} + \sum_{i=1}^{N} \frac{(y_i - \beta_1 x_i - \beta_0)^2}{2(\sigma_i^2)} \tag{4}$$

Using maximum likelihood estimation (MLE) to obtain parameters $\beta_0, \beta_1, \sigma_1$, and σ_2, we get the following iterates

$$\frac{\partial L}{\partial \beta_0} = \sum_{i=1}^{N} \frac{-(y_i - \beta_1 x_i - \beta_0)}{\sigma_i^2} \tag{5}$$

$$\frac{\partial L}{\partial \beta_1} = \sum_{i=1}^{N} \frac{-x_i(y_i - \beta_1 x_i - \beta_0)}{\sigma_i^2} \tag{6}$$

$$\frac{\partial L}{\partial (\sigma_1^2)} = \sum_{i=1}^{N} \frac{x_i^2}{2\sigma_i^2} - \sum_{i=1}^{N} \frac{x_i^2(y_i - \beta_1 x_i - \beta_0)}{2(\sigma_i^2)^2} \tag{7}$$

$$\frac{\partial L}{\partial (\sigma_2^2)} = \sum_{i=1}^{N} \frac{1}{2\sigma_i^2} - \sum_{i=1}^{N} \frac{(y_i - \beta_1 x_i - \beta_0)}{2(\sigma_i^2)^2} \tag{8}$$

Running the above algorithm for **dataset1** gives us $\beta_1 = 0.909086$, $\beta_0 = 3.112917$, $\sigma_1^2 = 2.014194$ and $\sigma_2^2 = 0.0$. Running the normality tests for the residues $(\frac{Y - X\beta}{X})$ for dataset1 we get good p values for all three tests. This leads us to not rejecting null hypothesis—the residues are drawn from a Gaussian distribution.

Title: Jarque - Bera Normalality Test
Test Results:
STATISTIC:
X-squared: 1.7396
P VALUE:
Asymptotic p Value: 0.419

Title: Shapiro - Wilk Normality Test
Test Results:
STATISTIC:
W: 0.9805
P VALUE:
0.1461

Title: D'Agostino Normality Test
Test Results:
STATISTIC:
Chi2 | Omnibus: 2.1558
Z3 | Skewness: -0.3344
Z4 | Kurtosis: 1.4297
P VALUE:
Omnibus Test: 0.3403
Skewness Test: 0.7381
Kurtosis Test: 0.1528

Running the algorithm on **dataset2** gives us $\beta_1 = -0.140639$, $\beta_0 = 1.010541$, $\sigma_1^2 = 2.825972$ and $\sigma_2^2 = 0.0$. Running the three normality tests for dataset2 residues we obtain (though just 40 datapoints make the tests less reliable):

Title: Jarque - Bera Normalality Test
Test Results:
STATISTIC:
X-squared: 46.7232
P VALUE:

Asymptotic p Value: 7.148e-11

Title: Shapiro - Wilk Normality Test
Test Results:
STATISTIC:
W: 0.8858
P VALUE:
0.0007558

Title: D'Agostino Normality Test
Test Results:
STATISTIC:
Chi2 | Omnibus: 24.3982
Z3 | Skewness: -3.701
Z4 | Kurtosis: 3.2712
P VALUE:
Omnibus Test: 5.035e-06
Skewness Test: 0.0002148
Kurtosis Test: 0.001071

We get better p values for dataset2 residues than earlier linear model residues for dataset2 for last two tests.

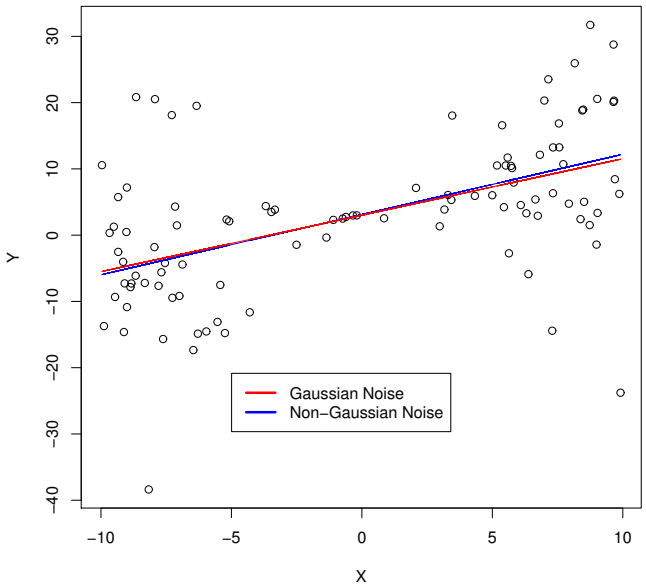

Figure 3: Fit of linear dependent noise on dataset1

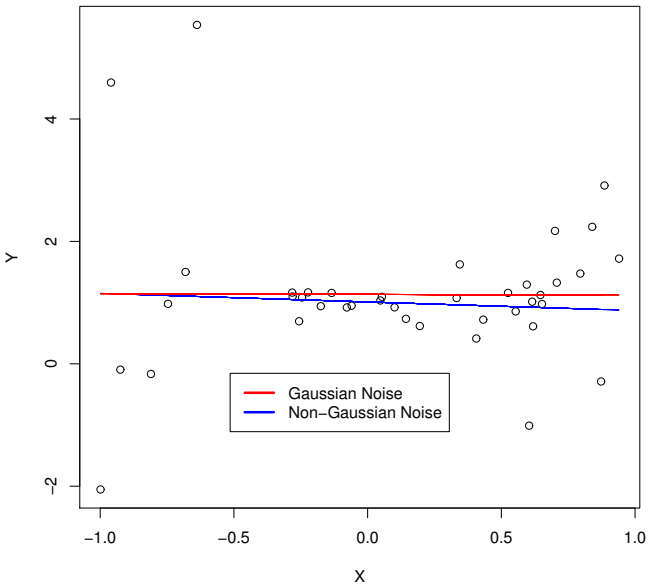

Figure 4: Fit of linear dependent noise on dataset2

Figures 3 and 4 show the fit for Gaussian (red) and Non-Gaussian (blue), linearly dependent, noise assumption models for dataset1 and dataset2 respectively. As noted before the residues for the linearly dependent noise model do not have null hypothesis of Gaussian noise assumption rejected which vindicates our assumptions/observations noted earlier. Given that dataset2 has very few datapoints and given that the noise model for dataset1 and dataset2 are the same, the σ_1^2 and σ_2^2 obtained for dataset1 are more reliable and can be used as the values for these respective parameters for dataset2. The β_1 and β_0 obtained for dataset1 and dataset2 are stable. We observe that their values converges close to the same set of values for both these datasets, respectively.

3 Running the Code

The model described in section above is coded in R. The code is in `LMwithLinearlyDependentNoise.R` file. Steps to run the R code:

1. You need `fBasics` package in R to run the Normality tests. It can be installed by typing `install.packages("fBasics")` at R command prompt.
2. Type `source("LMwithLinearlyDependentNoise.R")` at R command prompt to load the functions in the code.
3. Call `eda("dataset1")` to replicate the figure 1 and tests reported in section 1 for dataset1 and vice-versa for dataset2 (replace dataset1 by dataset2 in `eda("dataset1")`).

4. Call `lmNonGaussian("dataset1")` to replicate the figure 3 and tests reported in section 2 for dataset1 and vice-versa for dataset2.

Call to `lmNonGaussian("dataset1")` also reports the value of β_1, β_0, σ_1^2 and σ_2^2 as b1, b0, var1 and var2 respectively for each iteration of the gradient descent algorithm. After running the code we observe that the objective being minimized (the negative log likelihood, neg_ll) decreases and eventually becomes stable reaching the convergence point and so do the parameters of the model.