TACKLING HETEROSKEDASTIC FAT TAILED NOISE IN LINEAR REGRESSION

KARTHIK IYER

1. Problem Statement

Gaussian linear models are often insufficient in practical applications, where noise can be heavy tailed. In this problem, we consider a linear model of the form $y_i = \beta_1 x_i + \beta_0 + \epsilon_i$. The ϵ_i are independent noise from a distribution that depends on x as well as on global parameters; however, the noise distribution has conditional mean zero given x. The goal is to derive a good estimator for the parameters a and b based on a sample of observed $(x; y)$ pairs.

2. Solution

We first load the data, which is provided as (x; y) pairs in CSV format. Each file contains a data set generated with different values of a and b. There are 5 such files. The noise distribution, conditional on x, is the same for all data sets. We have a total of 350 observations in these 5 files (split up as 100, 100, 50, 50, 50 data points respectively).

Before we begin, let us state the assumptions of the simple linear regression model. For multiple data points, $(x_1, y_1), (x_2, y_2), ...(x_n, y_n)$, the model says that, for each $k \in 1 : n$, $y_k = \beta_0 + \beta_1 x_k + \epsilon_k$ where the noise variables ϵ_k all have the same conditional expectation (0) and the same variance (σ^2), and $Cov[\sigma_l, \sigma_j] = 0$ (unless l = j, of course). Furthermore, it is assumed that $\epsilon \sim N(0, \sigma^2)$.

For this simple linear regression model, the residual at the ith data point is the difference between the realized value of the response y_i and what the estimated model would predict:

$$e_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i).$$

We can rewrite the residual as

$$e_i = (\beta_0 - \hat{\beta}_0) + (\beta_1 - \hat{\beta}_1)x_i + \epsilon_i.$$

If we run an ordinary least squares (OLS) regression (note that the maximum likelihood formulation is equivalent to the OLS minimization under the assumptions above and the OLS estimates are in fact the best unbiased linear estimators), then it is well known (see [3]) that

$$\begin{aligned}\hat{\beta}_0 &= \beta_0 + \frac{1}{n}\sum_{i=1}^{n}\left(1 - \bar{x}\frac{x_i - \bar{x}}{s_x^2}\right)\epsilon_i,\\ \hat{\beta}_1 &= \beta_1 + \sum_{i=1}^{n}\frac{x_i - \bar{x}}{ns_X^2}\epsilon_i.\end{aligned}$$

where s_X^2 denotes the sample variance for x. We plug this in to the equation for ϵ_i:

$$\begin{aligned}e_i &= \sum_{j=1}^{n}\left(\delta_{ij} + \frac{1}{n} + (x_i - \bar{x})\frac{(x_j - \bar{x})}{ns_x^2}\right)\epsilon_j,\\ &= \sum_{j=1}^{n} c_{ij}\epsilon_j \ (c_{ij} \text{ depends on x's alone}).\end{aligned}$$

Since our model assumes that $\mathbb{E}[\epsilon_i|X] = 0$, it follows that $\mathbb{E}[e_i|X] = 0$, We know that the ϵ's are uncorrelated (in fact, we assume they are independent). Moreover,

(1) If we assume that all ϵ's have variance σ^2, even conditional on X, then

$$Var[e_i|X] = \sum_j c_{ij}^2 Var[\epsilon_j|X] = \sigma^2 \sum_j c_{ij}^2.$$

This can be used to show that $Var[e_i] = (1 - \frac{2}{n})\sigma^2$.

(2) If we assume that the ϵ_j are independent Gaussian, it follows that e_i also has a Gaussian distribution.

(As an aside, let us observe that, if the simple linear model is estimated by least squares, then $\sum_i e_i = 0$ and $\sum_i (x_i - \bar{x}) e_i = 0$ [1] This implies that even if we assume that the ϵ's are independent, the residuals are not.)

The observations above can be used as diagnostic tests to check the validity of our assumptions. Under assumptions of constant variance and Gaussian noise,

(1) The residuals should have expectation zero, conditioned on x. (The residuals should have an overall sample mean of 0.)

(2) If noise is Gaussian, it follows that e_i also has a Gaussian distribution.

(3) The residuals e_i should show a constant variance, unchanging with x.

Before we proceed, let us look at some plots and check the validity of assumptions of the simple linear regression model. We begin by looking at the scatter plot of the raw data along with the ordinary least squares (OLS) fit.

[1] These equations follow from the estimating equations and apply on any data set.

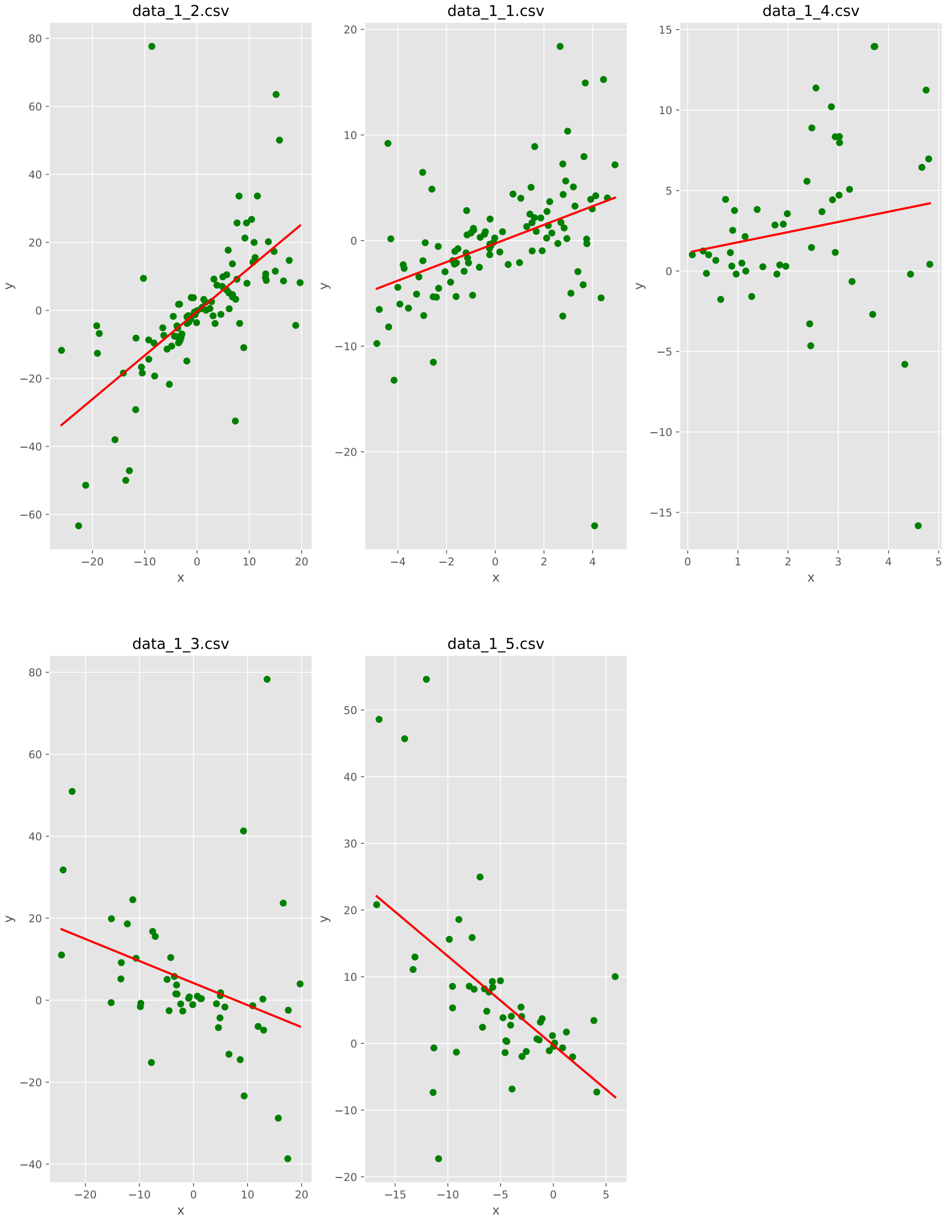

Figure 1: Scatter plot for raw data along with the OLS fit

We note that the first two data sets (’data_1_1.csv’) and (’data_1_2.csv’) contain majority of the observations and the noise looks to be symmetric about $\bar{x}$. OLS will hence be reasonably robust to fat tails in this case. For the other 3 data sets, the noise is clearly asymmetric about $\bar{x}$ and OLS will not be efficient.

Let us also residuals and squared residuals as a function of x. If there is a much greater range of residuals at large absolute values of x than towards the center; this changing dispersion is a sign of non-constant variance (heteroskedasticity). Because $\mathbb{E}[e|X = x] = 0$, $Var[e|X = x] = \mathbb{E}[e^2|X = x]$. This means we can check whether the variance of the residuals is constant by plotting the squared residuals against the predictor variable. This should give a scatter of points around a flat line, whose height should be around the in sample MSE.

Regions of the x axis where the residuals are persistently above or below this level signal a problem with the simple linear regression model either with the constant variance assumption or getting the functional form of the regression incorrect. (In our case, the latter is not possible as the functional form has been specified.) Under the Gaussian noise assumption, the residuals should also follow a Gaussian distribution. We therefore make plots of the distribution of the residuals, and compare that to a Gaussian. When the distribution of errors is heavy-tailed (a number of outliers are present) compared to normal, the normality assumption is violated.

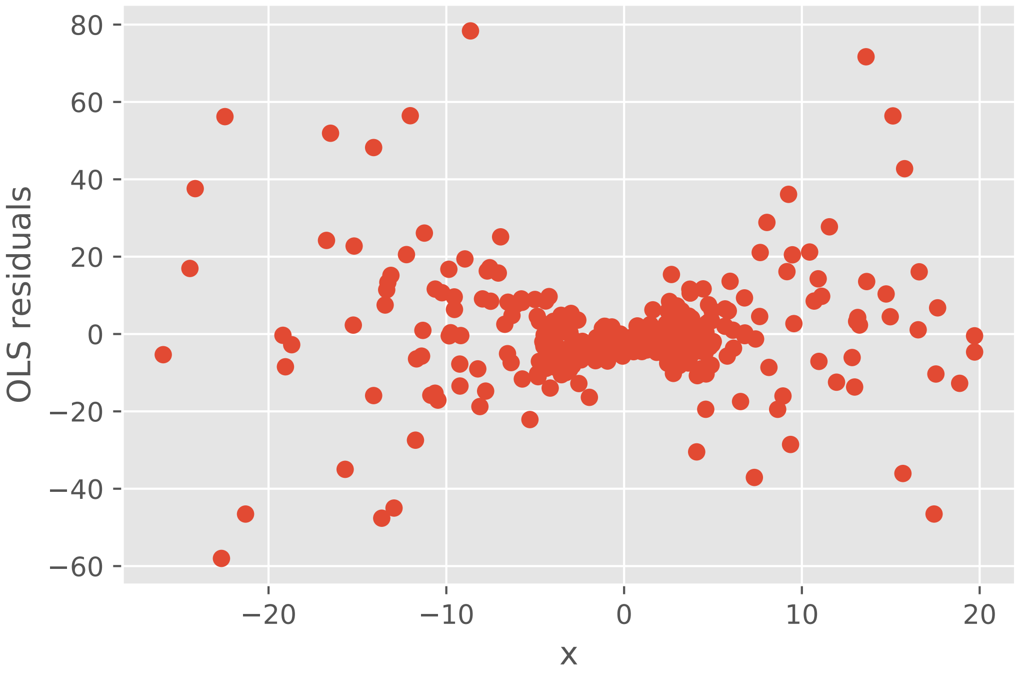

Figure 2: OLS residuals vs predictor; Aggregated

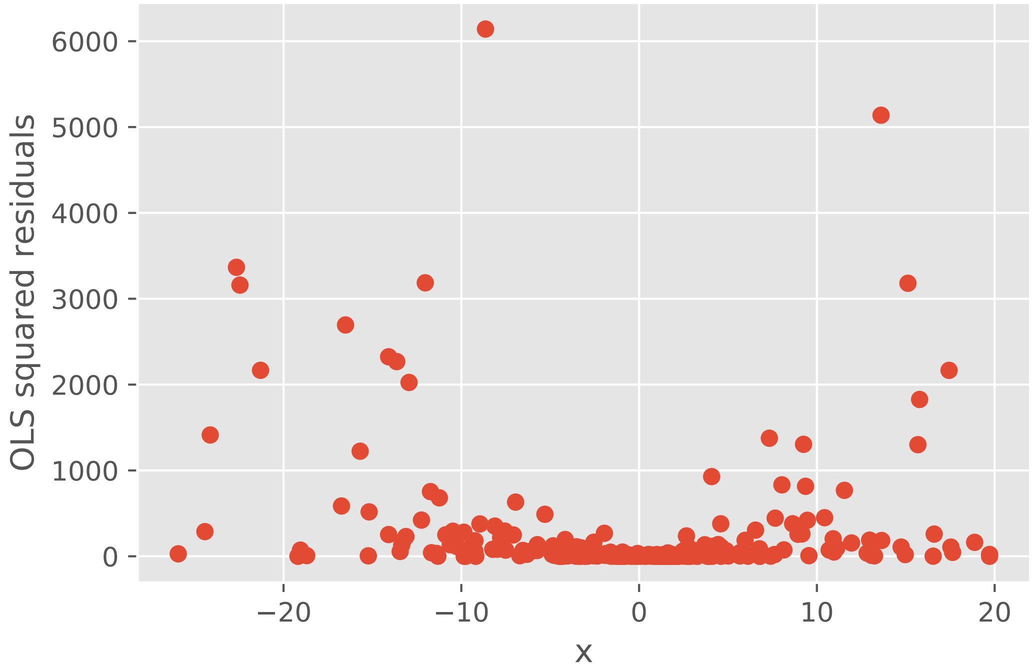

Figure 3: OLS squared residuals vs predictor; Aggregated

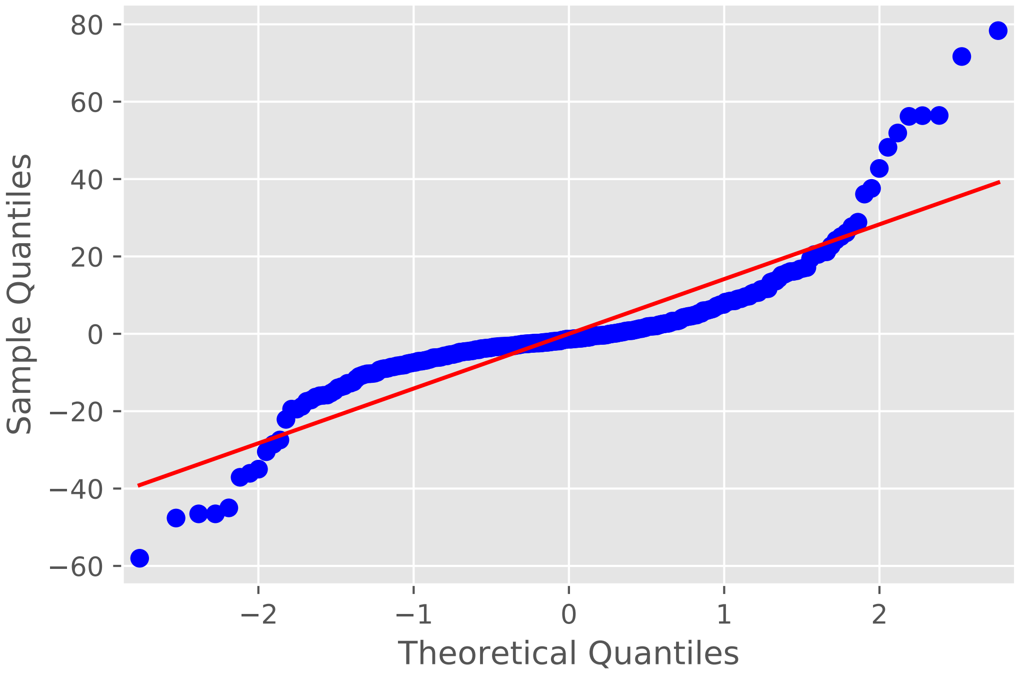

Figure 4: QQ plot to test normality of residuals

Looking at the diagnostic plots convinces us that the noise has non constant variance and the Gaussian noise assumption also fails to hold. Using OLS estimate will give us unbiased but inefficient estimates for the parameters of the model.

Linear least-squares estimates can behave badly when the error distribution is not normal, particularly when the errors are heavy-tailed. One remedy is to remove influential observations from the least-squares fit. Another approach, termed robust regression, is to employ a fitting criterion that is not as vulnerable as least squares to unusual data. Our solution to this problem consists of three different approaches; first tackling heavy tailed errors using robust regression techniques, second tackling non constant error variance by using a weighted least squares approach and third tackling heavy tailed error with non-constant variance.

2.1. **Tackling heavy tailed errors.** Let us first suppose that the error distribution of $\epsilon_i \sim e^{-\rho(x/s)}$ where s is some scale parameter and ρ is some chosen function such that the resulting distribution for the noise is heavy tailed. [2] Let us also assume ρ is differentiable and that $\rho' = \psi$. The log-likelihood is then proportional to $-\sum_{i=1}^{n} \rho\left(\frac{y_i-\beta_0-\beta_1 x}{s}\right)$ where s is a measure of scale. (Some robust estimators are influenced by the scale of the residuals, so we use a scale-invariant version.) This leads us to minimizing the objective function

$$\sum_{i=1}^{n} \rho\left(\frac{y_i - \beta_0 - \beta_1 x_i}{s}\right). \tag{2.1}$$

To get an explicit solution, we can differentiate the objective function (2.1) with respect to β to get a system of 2 estimating non linear equations for the coefficients:

$$\begin{aligned} &\sum_{i=1}^{n} \frac{1}{s}\psi\left(y_i - \beta_0 - \beta_1 x_i)/s\right) = 0, \\ &\sum_{i=1}^{n} \frac{x_i}{s}\psi\left((y_i - \beta_0 - \beta_1 x_i)/s\right) = 0. \end{aligned} \tag{2.2}$$

Define

$$w_i = \frac{\psi\left(\frac{y_i-\beta_0-\beta_1 x_i}{s}\right)}{\frac{(y_i-\beta_0-\beta_1 x_i)}{s}}. \tag{2.3}$$

We can thus re-write the system (2.2) as

$$\begin{aligned} &\sum_{i=1}^{n} w_i(y_i - \beta_0 - \beta_1 x_i) = 0, \\ &\sum_{i=1}^{n} x_i w_i(y_i - \beta_0 - \beta_1 x_i) = 0. \end{aligned} \tag{2.4}$$

This allows us to re-write the above system as $\sum_{i=1}^{n} w_i(y_i - \mathbf{x_i'}\beta)\mathbf{x_i}' = 0$. where $\mathbf{x_i'} = (1, x_i)^T$.

Hence, our estimate is

$$\hat{\beta} = (\mathbf{x}^{\mathrm{T}}\mathbf{w}\mathbf{x})^{-1}\mathbf{x}^{\mathrm{T}}\mathbf{w}\mathbf{y},$$

where $\mathbf{x} \in \mathbb{R}^{n\times 2}$ with the first column equal to $(1, 1, .., 1)^T$ and the second column equal to $(x_1, x_2, .., x_n)^T$, $\mathbf{y} = (y_1, y_2, ..., y_n)^T$ and $\mathbf{w}$ is a $n \times n$ diagonal matrix with the ith entry equal to w_i as defined in (2.3).

We see that solving the estimating equations is tantamount to solving a weighted least-squares problem; minimizing $\sum_{i=1}^{n} w_i^2 e_i^2$. The weights, however, depend upon the residuals, the residuals in turn depend upon the estimated coefficients, and the estimated coefficients depend upon the weights. An iterative solution (called iteratively re-weighted least squares, IRLS) is thus needed.

In an application, we need an estimate of the standard deviation of the errors to use these results. Usually a robust measure of spread is employed in preference to the standard deviation of the residuals. We will standardize the residuals by a robust estimate of their scale s, which is

[2] Note that we have assumed here that the noise across different observations is iid.

estimated simultaneously. Using a robust measure of spread is used in preference to the standard deviation of the residuals. For example, a common approach is to take $\hat{s} = MAR/0.6745$, where MAR is the median absolute residual. The constant $0.6745 = \Phi^{-1}(0.75)$, where Φ is the cumulative function for standard normal, is chosen so that s is asymptotically unbiased for σ if the $\epsilon_i \sim N(0, \sigma^2)$. [1]

We thus have the following iterative algorithm to estimate the regression coefficients and the scale parameter.

IRLS algorithm

(Step 0) Fit an OLS regression to the data and set $w_i^0 = \left(\frac{1}{e_i^{(0)}}\right)^2$ where $e_i^{(0)}$ is the initial residual for $i = 1, 2.., n$.

(Step 1) (Do a weighted least squares regression for β^j using $\mathbf{w}^{\mathrm{j-1}}$) : $\beta^{\mathrm{j}} = (\mathbf{x}^T\mathbf{w}^{\mathrm{j-1}}\mathbf{x})^{-1}\mathbf{x}^T\mathbf{w}^{\mathrm{j-1}}\mathbf{y}$.

(Step 2) Set $\hat{s}^j$ = Median absolute deviation $(y_i - \beta_0^j - \beta_1^j x_i) \ : \ i = 1, 2, .., n$

(Step 3) Set $w_i^j = \frac{\psi\left(\frac{y_i - \beta_0^j - \beta_1^j x_i}{\hat{s}^j}\right)}{\frac{(y_i - \beta_0^j - \beta_1^j x_i)}{\hat{s}^j}}$.

Set j = j +1

(Step 4) Repeat steps 1 through 3 till $\hat{s}^j$ and β^j stabilize.

Choices for ρ

Common functions that are used for ρ are Huber's function and Tukey's bi weight functions. Both of these functions depend on a tuning hyper parameter. The value c for the Huber and bi-square estimators is called a tuning constant; smaller values of c produce more resistance to outliers, but at the expense of lower efficiency when the errors are normally distributed. (Huber penalty will penalize outliers only linearly while Tukey penalty penalty effectively removes outliers.)

1. Huber's function:

$$\rho(z) = \begin{cases} \frac{1}{2}z^2 & \text{if } |z| \leq c \\ c(|z| - \frac{c}{2}) & \text{if } |z| \geq c \end{cases}$$

In this case,

$$w(z) = \begin{cases} 1 & \text{if } |z| \leq c \\ \frac{c}{|z|} & \text{if } |z| \geq c \end{cases}$$

2. Tukey's biweight:

$$\rho(z) = \begin{cases} \frac{c^2}{3}\left(1 - [1 - (\frac{z}{c})^2]^3\right) & \text{if } |z| \leq c \\ 2c & \text{if } |z| \geq c \end{cases}$$

In this case,

$$w(z) = \begin{cases} [1 - (\frac{z}{c})^2]^2 & \text{if } |z| \leq c \\ 0 & \text{if } |z| \geq c \end{cases}$$

The tuning constant is generally picked to give reasonably high efficiency in the normal case; in particular, c = 1.345 for the Huber and c = 4.685 for the Tukey bi-square produce 95-percent efficiency when the errors are normal, and still offer protection against outliers.[3]

[3] Ideally, the value of c should be chosen depending on the data set and can be fine-tuned via a k-fold cross validation.

2.2. **Weighted least squares.** To combat heteroskedastic noise, a common approach is to use weighted least squares. Instead of minimizing the mean square error, we could minimize the weighted mean square error $\frac{1}{n}\sum_{i=1}^{n} w_i(y_i - \mathbf{x_i}'\beta)^2$. Minimizing this error leads us to the following estimate:

$$\hat{\beta}_{WLS} = (\mathbf{x}^{\mathrm{T}}\mathbf{w}\mathbf{x})^{-1}\mathbf{x}^{\mathrm{T}}\mathbf{w}\mathbf{y}$$

Note that when we do weighted lest squares, our estimates for β are unbiased. By Gauss Markov theorem, WLS with the weight matrix equal to the inverse covariance matrix Σ^{-1}, the least variance among all possible linear, unbiased estimators of the regression coefficients. When we have heteroskedasticity, OLS is no longer the optimal estimate. If however we know the noise variance σ_i^2 at each measurement i, and set $w_i = \frac{1}{\sigma_i^2}$, we minimize the variance of estimation. [4]

To see, why WLS is unbiased, note that

$$\hat{\beta} = (\mathbf{x}^{\mathrm{T}}\mathbf{w}\mathbf{x})^{-1}\mathbf{x}^{\mathrm{T}}\mathbf{w}\mathbf{y}$$

Since $\mathbb{E}[\epsilon_i|x] = 0$ and $Var[\epsilon_i|x] = \sigma_i^2$. So

$$\hat{\beta} = (\mathbf{x}^{\mathrm{T}}\mathbf{w}\mathbf{x})^{-1}\mathbf{x}^{\mathrm{T}}\mathbf{w}\mathbf{x}\beta + (\mathbf{x}^{\mathrm{T}}\mathbf{w}\mathbf{x})^{-1}\mathbf{1}\mathbf{x}^{\mathrm{T}}\mathbf{w}\epsilon)) = \beta + (\mathbf{x}^{\mathrm{T}}\mathbf{w}\mathbf{x})^{-1}\mathbf{x}^{\mathrm{T}}\mathbf{w}\epsilon$$

Since $\mathbb{E}[\epsilon|x] = 0$, the WLS estimate is unbiased. $\mathbb{E}[\hat{\beta}_{WLS}|x] = \beta$. It can be shown by generalized Gauss Markov theorem that picking weights to minimize the variance in the WLS estimate has the unique solution $w_i = \frac{1}{\sigma_i^2}$. It does not require us to assume that the noise is Gaussian!

The argument in the previous paragraph works well if we know what the conditional variance is. What if we don't know the conditional variances? We can use non-parametric smoothing to estimate conditional variances. As a first step we can do an OLS and construct log squared residuals, then use a non-parametric method to estimate their conditional mean and predict variance using $\sigma_x^2 = exp(s\hat{s}(x))$. ([2] 2006, pp 87-88)

Suppose that $Y_i = r(x_i) + \sigma(x_i)\epsilon_i^*)$. Let $Z_i = \log((Y_i - r(x_i))^2)$ and $\delta_i = \log\epsilon^{*2}_i$. Thus $Z_i = \log(\sigma^2(x_i)) + \delta_i$. Note that $Var(\epsilon_i^*|x_i) = 1$. This suggests the following method for estimating $\log\sigma^2(x)$.

(1) Construct the log squared residuals $z_i = \log((y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2)$.

(2) Use any non-parametric method to estimate the conditional mean of z_i, call it $\hat{s}(x)$. (We use Kernel regression in Python. By applying a log-transformation, this problem of estimating $\sigma(x)$ is similar to non-parametric estimation of the conditional mean.)

(3) Predict the variance using $\hat{\sigma}^2(x) = \exp\hat{s}(x)$.

This approach is more flexible as we do not assume any parametric model for $\sigma^2(x)$.

The estimate $\sigma^2(x)$ depends on the initial estimate of the regression function $\hat{\beta}_0 + \hat{\beta}_1 x$. Taking heteroskedasticity into account can change our estimates of the regression function. This suggests an iterative approach, where we alternate between estimating the regression function and the variance function, using each to improve the other. That is, we take either method above, and then, once we have estimated the variance function $\hat{\sigma}^2(x)$, we re-estimate $\hat{m}$ using weighted least squares, with weights inversely proportional to our estimated variance. Since this will generally change our estimated regression, it will change the residuals as well. Once

[4] We do not have to worry about correlated errors as our problem specifies that errors across different observations are independent. Hence the covariance matrix will be diagonal and weighted linear regression can be used to combat heteroskedasticity.

the residuals have changed, we should re-estimate the variance function. We continue until the coefficients for regression stabilize.

2.3. **Robust heteroskedastic regression: A parametric approach.** Another approach to this problem via assuming that there exists a non random function $v : \mathbb{R}^n \longrightarrow \mathbb{R}^+$ such that $\mathbb{E}[\epsilon_i^2|x_i] = v(x_i)$. We further assume that v depends on certain parameters θ. This assumption is particularly amenable to a maximum likelihood estimation approach. Now, suppose we place a heavy tailed distribution (say, double exponential) on the likelihood and then take the negative logarithm to obtain an objective (cost) function to be minimized. [5] To wit, assume that for each i, $\epsilon_i|x_i \sim DE(0, \sqrt{v(x_i)})$, where DE denotes the double exponential distribution. This implies $y_i|x_i \sim DE(\beta_0 + \beta_1 x_i, \sqrt{v(x_i)})$. Since the conditional errors are independent, taking negative logarithm gives the objective function

$$L(\beta_0, \beta_1; \theta) = \frac{1}{n}\sum_{i=1}^{n}\left(0.5\log(v(x_i)) + \frac{|y_i - \beta_0 - \beta_1 x_i|}{\sqrt{v(x_i)}}\right). \tag{2.5}$$

For the sake of tractability, we again make a modeling choice and assume $v(x) = \exp(\alpha_0 + \alpha_1 x)$.

There are countless choices for v that could have been made. We choose a composition of a linear and exponential function. We choose an exponential function as it makes the minimization problem in (2.5) reduce to a simpler form and we choose a linear function as the number of extra parameters to be estimated in this case is only 2 and ensures that we don't over optimize.

Thus, our optimization problem is reduced to

$$\text{argmin}_{\beta_0,\beta_1;\alpha_0,\alpha_1} L(\beta_0, \beta_1; \alpha_0, \alpha_1) = \text{argmin}_{\beta_0,\beta_1;\alpha_0,\alpha_1} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}(\alpha_0 + \alpha_1 x_i) + e^{-0.5(\alpha_0+\alpha_1 x_i)}|y_i - \beta_0 - \beta_1 x_i|\right).$$

To minimize the above cost function, we can use a gradient descent algorithm. [6] For that, we need to compute the partial derivatives with respect to the parameters which we now do.

$$\begin{aligned}
\nabla_{\alpha_0} L &= \frac{1}{2} - \frac{1}{2n}\sum_{i=1}^{n} e^{-0.5(\alpha_0+\alpha_1 x_i)}|y_i - \beta_0 - \beta_1 x_i| \\
\nabla_{\alpha_1} L &= \frac{1}{2n}\sum_{i=1}^{n} x_i\left(1 - e^{-0.5(\alpha_0+\alpha_1 x_i)}|y_i - \beta_0 - \beta_1 x_i|\right) \\
\nabla_{\beta_0} L &= -\frac{1}{n}\sum_{i=1}^{n} e^{-0.5(\alpha_0+\alpha_1 x_i)} sgn(y_i - \beta_0 - \beta_1 x_i) \\
\nabla_{\beta_0} L &= -\frac{1}{n}\sum_{i=1}^{n} x_i e^{-0.5(\alpha_0+\alpha_1 x_i)} sgn(y_i - \beta_0 - \beta_1 x_i)
\end{aligned}$$

[5] Double exponential distribution is one example of a heavy tailed distribution for the noise. We use it primarily since the minimization of the objective functions boils down to an L_1 like minimization and there is no involvement of any hyper-parameter. Other robust regression methods like Huber or Tukey's bi-weight use a hyper-parameter which has to be found using a cross validation technique. We wish to avoid all of this and hence our choice of the double exponential distribution.

[6] Technically, |.| is not differentiable everywhere, so its use in a gradient descent algorithm might seem suspicious. However, |.| is differentiable *almost everywhere*; the only points where it is not differentiable in our case is when the coefficients of the regression are such that the predict values exactly fit the given responses. This is an unlikely scenario (it can potentially happen for only a small number of responses.) Hence, we ignore this possibility and proceed with this method.

This method requires that we specify the learning rate for gradient descent and initial values of β and α. We choose the same small learning rate of 10^{-8} and initial values of α to be $[0, 0.1]$ for every data set. We set maximum number of iterations to 30000. Because this optimization problem is non-convex, we can at most hope to converge to a local minima (maybe even a saddle point). We deliberately choose a small learning rate to make sure we don't cross a local critical point. [7] This approach is thus sensitive to the initial values of β because of the slow convergence. Instead of randomly choosing an initial β, we choose the OLS estimates as our initial β, thereby turning this approach in to a two-step approach To make sure the gradient descent is working correctly, we plot $L(\beta_0, \beta_1; \alpha_0, \alpha_1)$ against the number of iterations. If gradient descent is working correctly, $L(\beta_0, \beta_1; \alpha_0, \alpha_1)$ should decrease after every iteration which we visually confirm.

2.4. **Plots of fitted line.**

[7] As a possible improvement to this approach, we can dynamically update the learning rate by doing a line search.

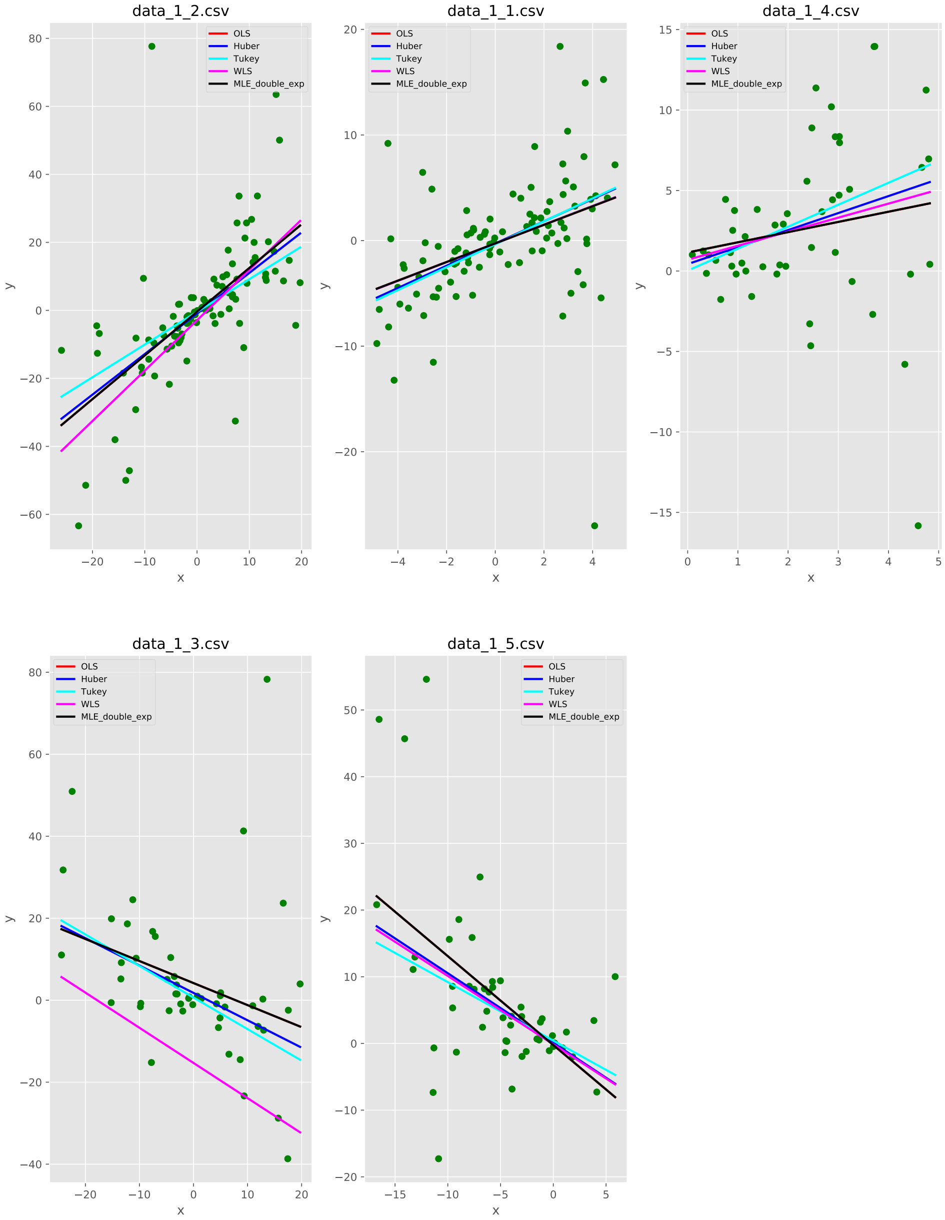

Figure 5: Linear fit to the data

The MLE fits are similar to OLS fits. This is expected as the initial values of MLE fits are given by an OLS regression and our learning rate is small. We note that for `('data_1_1.csv')` and for `('data_1_2.csv')`, the linear fit is very similar for every approach while for the remaining data sets, the robust regression fits are more robust to noise than OLS fits.

2.5. **Performance comparison and conclusion.** We have used 5 techniques to solve this problem, OLS regression, Robust regression with Huber penalty, Robust regression with Tukey penalty, Weighted least squares with iterative refinement of mean and variance and MLE estimates with double exponential noise distribution. To compare their relative performance, we used 5 fold cross validation to compute their average test errors and save the results in to a .csv file.

We notice that for `('data_1_1.csv')`, the average test errors are about the same which validates our initial observation about this data set which we made by looking at the scatter plot and OLS fit. For all other data sets, the WLS approach gives a considerably worse average test error. [8] We also notice that the Huber approach gives the least test error for `('data_1_3.csv')` and `('data_1_4.csv')` and gives close to good average test errors for other data sets too. This leads us to conclude robust regression techniques work best in this case.

As an improvement to the robust regression, we can factor in heteroskedastic noise by considering a robust *heteroskedastic* regression where we would set up a iterative re-weighted least squares with weights that capture heteroskedasticity. We could also consider other robust regression techniques like *least trimmed squares* or *least trimmed absolute value* to combat heavy tails and adapt those techniques to factor in heteroskedasticity.

References

[1] http://statweb.stanford.edu/ jtaylo/courses/stats203/notes/robust.pdf
[2] https://web.stanford.edu/class/ee378a/books/book2.pdf
[3] http://www.stat.cmu.edu/ cshalizi/ADAfaEPoV/ADAfaEPoV.pdf

[8] Perhaps, this is an indication that fat tails is a bigger issue here compared to heteroskedasticity.