Tidy Data: Case Study Exercise

Maria Camila Castaño Martínez[1][0009−0002−8514−1373]

Tecnológico de Monterrey, 45201 Zapopan, Jal.
A01649450@tec.mx

Abstract. The goal of this experiment is to implement 3 different linear regression methods by using a dataset obtained on 2008, related to the deaths in Mexico. The initial revision and manipulation of this dataset comes from a case study presented by Hadley Wickham [1]. The main idea was to follow the same subset created called "*hod*2" to test all the linear methods, obtaining similar results compared to the initial model (RLM) of the case study, compare them following the MAD metric where there was a revision of all the residuals values which gave us really close results between them. Our results show that the best model for this particular case is Theil-Sen Estimator.

Keywords: Data manipulation · Linear Regression Modeling · Ordinary Squares Line · Theil-Sen Estimator · Huber Regressor · Python.

1 Dataset description

The dataset provided by the Github Repository "*tidy_ data*"[2] from Section 5, Case study. This study uses individual-level mortality data from Mexico, contains 539,530 deaths in 2008 and 55 variables. The variables will be presented as abreviations on Table 1 for more clarity. The study only focuses on variables related to time of death and cause of death.

Variables	**Abreviations**
Cause of death	cod
Hour of death	hod
Day of death	dod
Month of death	mod
Year of death	yod

Table 1. Variable abreviations.

2 Linearl Models (Methods)

To complete this benchmark, it is needed to utilize 3 different methods for linear modeling and implement them into the R code extracted from Github. The method used on the paper, Robust Linear Model (RLM).

2.1 Ordinary Least Squares (OLS)

Technique used for the estimation of linear regression equations coefficients which can describe the relationship between one or more independent variables, and a dependent variable. Can be evaluated using r-squared [3]. Aims to minimize the sum of square differences between the observed and predicted values, considering potential issues of multicollinearity. The general OLS formula is:

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{1}$$

- $\hat{\beta}$: Ordinary least squares estimator
- X: Regressors matrix of x
- T: Transposed matrix
- y: Dependent variable

2.2 Theil-Sen Estimator

Known non-parametric method for robust linear regression that focuses on the median slope of all possible pairs of points in the dataset. This approach makes it robust to outliers and extreme values [4]. Mathematically, this estimator works by calculating the slopes between every pair of data points (i, j) with $i > j$:

$$\beta_{ij} = \frac{y_j - y_i}{x_j - x_i} \tag{2}$$

The estimated slope $\hat{\beta}_{TS}$ is the median of all pairwise slopes:

$$\hat{\beta}_{TS} = median\left\{ \frac{y_j - y_i}{x_j - x_i} | 1 \leq i < j \leq n \right\} \tag{3}$$

2.3 Huber Regressor

The Huber Regressor is a robust regression method designed to minimize the impact of outliers in data by combining the advantages of ordinary least squares (OLS) regression and absolute deviation regression. It minimizes a loss function that is quadratic for small errors and linear for large errors, which help reduce impact of outliers [5]. Mathematically, this regressor contain a loss function, objective function and use of gradient for the score function given n observations in a linear model. Obtaining the following final formula:

$$(\hat{\beta}_0, \hat{\beta}) = \underset{\beta_0, \beta}{argmin} \sum_{i=1}^{n} \begin{cases} \frac{1}{2}{}^2_i, & |r_i| \leq \delta \\ \delta(|r_i| - \frac{1}{2}\delta), & |r_i| > \delta \end{cases} \tag{4}$$

3 Code division

To complete this objective, it was necessary to divide the Jupyter Notebook used to code the R code lines from the main paper, which was obtained on Github [2]. This notebook presents all the markdowns needed to understand the meaning of each step.

3.1 Translation of R into Python

There was the need to translate the R code obtained from the paper into Python with the intention to work with it and the subsets created for the model implementation experiments.

First, download and preprocess the complete dataset. Several preprocessing steps are required before any statistical analysis can be implemented. We can see that there was the selection of certain columns with the intention of focus, also, the elimination of null values for a clearer dataset.

It is notorious on the notebook that several subsets were created with the intention of explaining in a clearer way, like the selection of specific columns like year, month, day and hour of death with also the causes. By dping this exercise, it was possible to obtain the set "*hod*2" to test the linear models, in this case, the Robust Linear Model (RLM) presented on the original case study, which contains an important column used on the main deviation, Diseases.

Then, was performed the deviation analysis for the implementation of RLM for the predictions and residuals based on the data selected. Obtaining the following graphs:

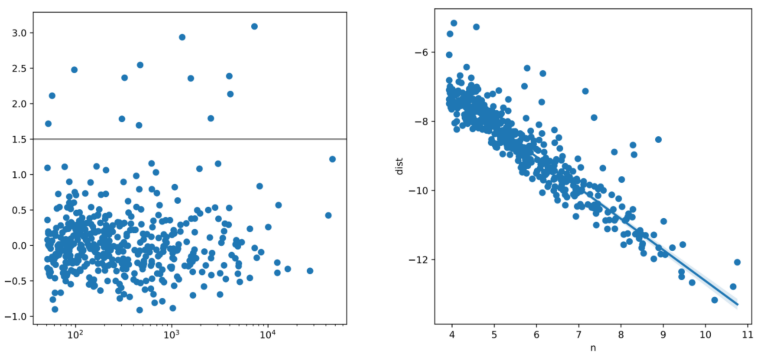

Fig. 1. Residuals and Linear Regression graph from RLM.

3.2 Methods to perform a Linear Model

For this section, the division was by every method used. It si important to note that all the steps worked were the same for every method for a clearer result, the only difference here was the change of models on each division. Also, there is log-log transformation of the variables $(n, dist)$. For these implementation, it was used the Scikit Learn package.

Important to remember this notations:

- Ordinary Least Squares: $ols = LinearRegression()$
- ts: $TheilSenRegressor(random\_state = 0)$
- huber: $huber = HuberRegressor()$

Now, let's explain each step followed during the implementation of each method:

1. State of the log-log variables (X, y).
2. Fitting of the model
3. Calculation of coefficients β_0 and β_1, as the intercept and slope. With the intention of known the scaling relationship between disease prevalence and deviation from the global hourly mortality pattern.
4. Calculation of predicted values and residuals from the data
5. Plotting of n-dist and residuals

Now, we can present the results from these implementations

4 Results

Now, let's present the results obtained from each method. This section will have the β coefficients calculated and graphs for both n-dist and residuals.

4.1 OLS

Here are the coefficients obtained for the model using "$hod2$" and "$devi$".

OLS Slope	OLS Intercept
-0.869011	-3.74324

Table 2. Coefficients beta (OLS).

These coefficients are a slope and intercept coefficient, will help us estimate the disproportion influenced by those cases with extreme deviations.

The slope coefficient β_1 represents the rate at which the deviation from the overall hourly mortality pattern decreases as the number of deaths for a disease increases. A value close to -0.9 shows us something important, as diseases

become more common, their hourly mortality patterns increasingly present a global pattern.

On the oher hand, the intercept β_0 sets the overall scale of the relationship above. Seem slightly biased toward fitting these extreme cases.

Here are the graphs obatined:

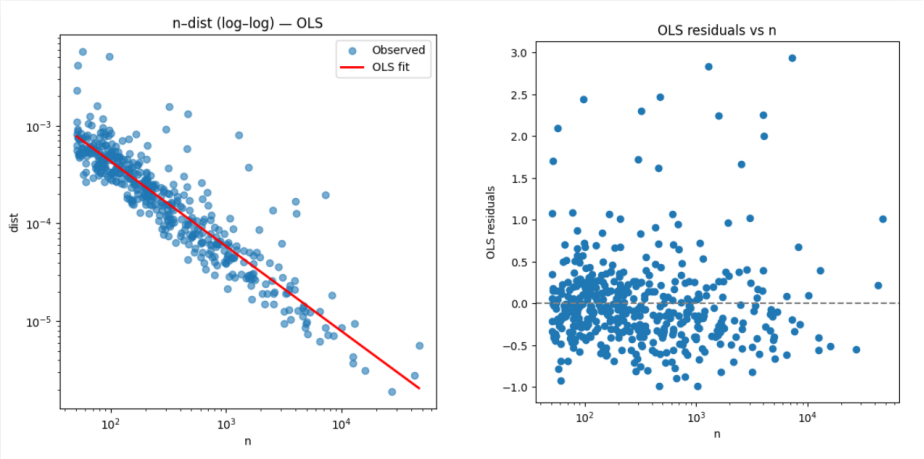

Fig. 2. Line and residuals from OLS.

4.2 TS

Here are the coefficients obtained for the model using "*hod*2" and "*devi*".

TS Slope	TS Intercept
−0.905483	−3.622510

Table 3. Coefficients beta (OLS).

Similar to the OLS slope value obtained, the Theil-Sen case confirms that the inverse scaling observed by the first method is not dirven by a small number of extreme diseases but reflects a robust population-level pattern and results.

The intercept β_0 indicates a similar baseline level of deviation for rare diseases but slightly more conservative that the OLS estimation. Reflect reliance on medians rather than means.

Here are the graphs obatined:

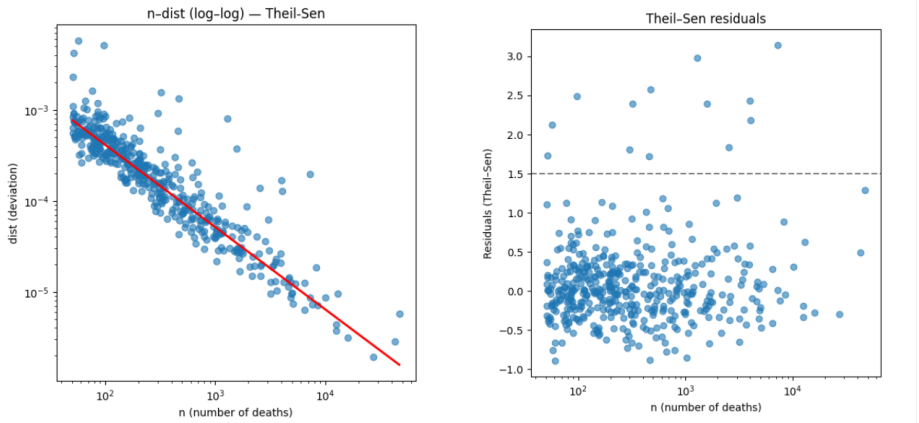

Fig. 3. Line and residuals from TS.

4.3 H

Here are the coefficients obtained for the model using "*hod*2" and "*devi*".

Huber Slope	Huber Intercept
−0.901513	−3.628816

Table 4. Coefficients beta (OLS).

These values lies between those obtained by OLS and Theil-Sen, showing the hybrid nature of the method. The slope coefficient again indicates a strong relationship consistent with the other models. The intercept value places the regression line slightly lower than OLS, suggesting a modest reduction in the estimated magnitude of deviations for rare diseases.
Here are the graphs obatined:

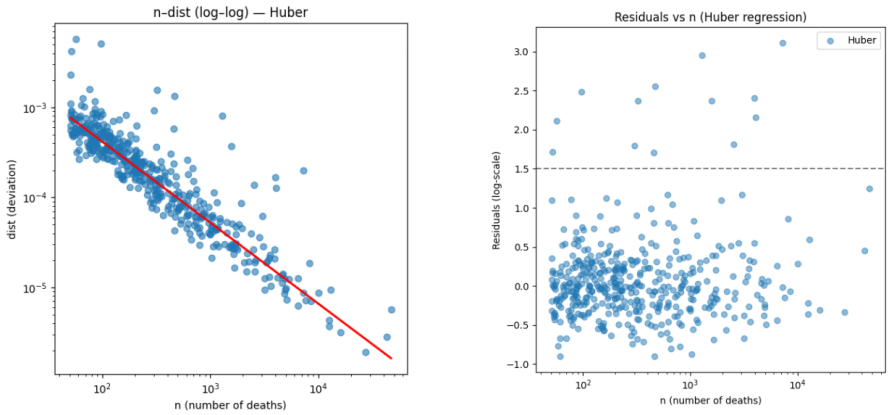

Fig. 4. Line and residuals from H.

4.4 Comparison of models

Figure 2 shows the relationship between the numbers of deaths per disease and the deviation from the overall hourly mortality pattern on a log-log scale for all the models implemented. The scatter points reveals that those diseases with fewer deaths tend to exhibit larger deviations, while common diseases closely follow the global hourly pattern.

The OLS regression line appears slightly elevated compared to the other robust models, and also at low values n, which corresponds to rare diseases. Reflects the model's sensitivity at extreme deviations.

The Theil-Sen regression line lies slightly below the OLS line across most of the range of n. This indicates that this estimator captures a steeper and more stable decline in deviation as disease frequency increases.

The Huber regression line closely tracks the Theil-Sen fit but remains at a margin close to the OLS line obtained, especially in regions where the data density is high. Can reflect how this method treats small residuals similarly to the first method while downweighting large residuals, resulting as a balance for efficiency by producing a fit that is resistant to extreme deviations without completely ignoring them.

These 3 methods shows similar results between them based on the data from the subset *hod*2. However, the differences in vertical placement and slope at low values of n reveal how each method responds to rare diseases with atypical hourly patterns. The Theil-Sen estimator provides the most representative description of the central trend.

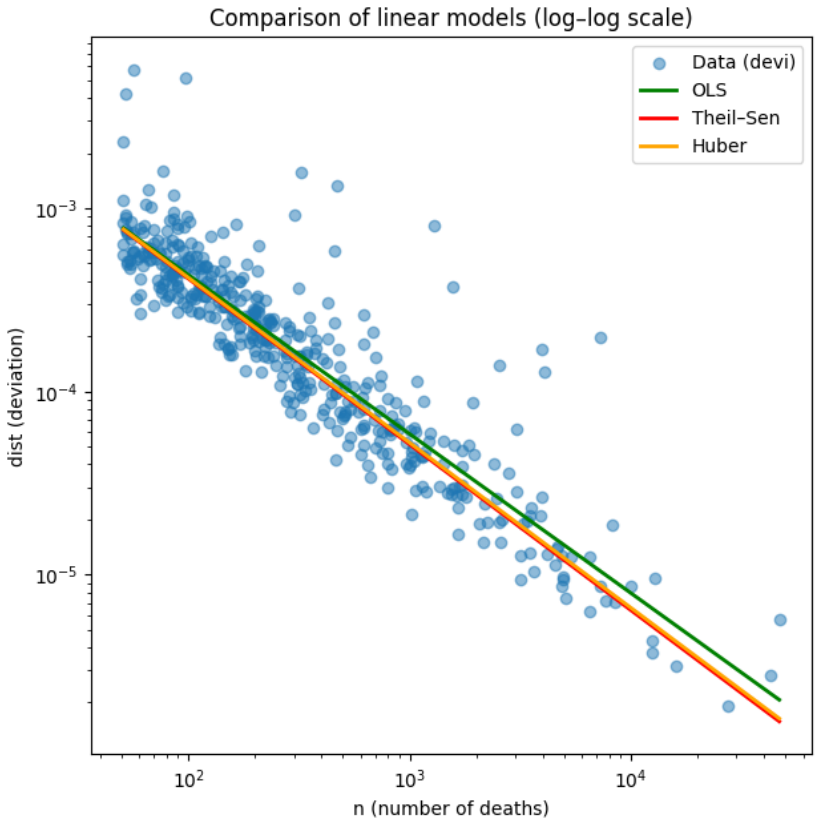

Fig. 5. Comparison between the 3 methods.

5 Performance Comparison

5.1 Median Absolute Deviation

Using the median absolute deviation of residuals as a robust performance metric, the Theil–Sen estimator achieves the lowest typical prediction error, outperforming both ordinary least squares and Huber regression. This indicates that robust estimators better capture the central scaling relationship between disease frequency and hourly mortality deviation, while ordinary least squares is disproportionately influenced by rare causes of death.

Used as a regression metric for the comparison of results between the 3 models.

Overall, these results demonstrate that robust regression techniques outperform OLS when modeling the log-log relationship between disease frequency and mortality deviation. The lower MAD values obtained by the Theil-Sen and Huber

Model	MAD	Approx. typical error
OLS	0.262647	$\sim 30\%$
Theil-Sen	0.243991	$\sim 28\%$
OLS	0.245828	$\sim 28.5\%$

Table 5. Caption

models confirm that downweighting or excluding the influence of outliers leads to more reliable and representative estimates of the underlying scaling behavior.

6 Conclusions

This study presents that the choice of a regression model must be guided by both the structure of the data and the goal of the analysis. When some datasets exhibit heavy-tailed distributions or influential outliers, as is the case for rare causes of death, robust linear models such as Theil-Sen and Huber regression provide more representative estimations than OLS. These methods better capture the central scaling relationship by reducing the undue influence of extreme observations.

Additionally, the results emphasize the critical role of data tidying and preprocessing prior to modeling. Transformations such as log-log scaling and adequate subset selection are essential to secure useful comparisons and statistical interpretations. Without these steps, model performance metrics and inferred relationships may be misleading, regardless of the sophistication of the regression technique employed.

References

1. Journal of Statistical Software: Tidy Data, *https* : *//github.com/hadley/tidy* − *data/blob/master/tidy* − *data.tex*, last accessed 2026/01/19
2. Tidy data, *https* : *//github.com/hadley/tidy* − *data/tree/master*, last accessed 2026/01/19
3. ORDINARY LEAST SQUARES REGRESSION (OLS), *https* : *//www.xlstat.com/solutions/features/ordinary* − *least* − *squares* − *regression* − *ols*, last accessed 2026/01/19
4. Theil-Sen estimator, MATLAB, *https* : *//www.mathworks.com/matlabcentral/fileexchange/*34308− *theil* − *sen* − *estimator*, last accessed 2026/01/20
5. HuberRegressor, Scikit Learn, *https* : *//scikit* − *learn.org/stable/modules/generated/sklearn.linear*$_m$*odel.HuberRegressor.html*, last accessed 2026/01/20