Linear Regression I: Parameters of a Linear Model

Mikel Ignacio Barajas Martínez

Ingeniería en Sistemas Inteligentes, 2021, 336483, 202102300012

Machine Learning, 281601

January 28th, 2024

1 Abstract

This paper serves as a primer to regression models, introducing the concept of linear regression and proposing two different methods to compute the parameters of an approximating line: ordinary least squares and the Theil-Sen estimator. Both of these methods are tested using the "accidents" dataset, which describes the size of the population per state (x-axis) and the number of fatal traffic accidents (y-axis) in the US.

2 Introduction

Linear regression, or linear curve fitting, can be described as finding the line that best adheres to some data (Cuevas, 2020). Linear regression can be seen as a method from approximation theory: fitting functions to given data and finding the "best" in a certain class to represent said data (Burden & Faires, 2001).

Linear regression is useful since, in contrast to higher-degree polynomial approximators, it does not introduce oscillations that were not originally present in the data. Finding the best approximating line, even if it does not precisely agree with the data at any point, can be more useful to study data behavior (Burden & Faires, 2001).

This model is represented by the equation of the line. Let a_1xi+a_0 denote the ith value on the approximating line, and y_i be the corresponding y-value. It is assumed that the independent x variable is exact and the dependent y-value is unknown.

There are multiple methods for finding the parameters that describe a linear regression, such as the Ordinary Least Squares Method (OLS) and the Theil-Sen estimator (TSE). The accuracy of any of these models can be determined through metrics such as the mean squared error and the coefficient of determination (R^2 score), which determines how well the model fits the data in the range of 0-1.

3 Ordinary Least Squares

The OLS approach involves finding the best approximating line when the error represents the sum of the squares of the differences between the y-value on said line and the actual y-values. This turns the situation into an optimization problem, where the coefficients must be found so that the least squares error is minimized.

Given this criteria, the slope of the approximating line (a_1) can be computed as follows:

$$a_1 = \frac{\sum i(x_i - \bar{x})(y_i - \bar{y})}{\sum i(x_i - \bar{x})^2}$$

Where $\bar{x}$ and $\bar{y}$ represent the mean x and y values. The y-intercept (a_0) can then be obtained using the slope and both $\bar{x}$ and $\bar{y}$:

$$a_0 = \bar{y} - a_1\bar{x}$$

3.0.1 Implementation

```
x_mean = np.mean(x)
y_mean = np.mean(y)

top = 0
bottom = 0

for i in range(len(df)):
    top += (x[i] - x_mean) * (y[i] - y_mean)
    bottom += (x[i] - x_mean) ** 2

a1 = top / bottom
a0 = y_mean - a1 * x_mean
```

3.0.2 Testing

Using the accidents dataset, the following coefficients were obtained:

$$y = 0.00012563942738769 83x + 142.71201717265376$$

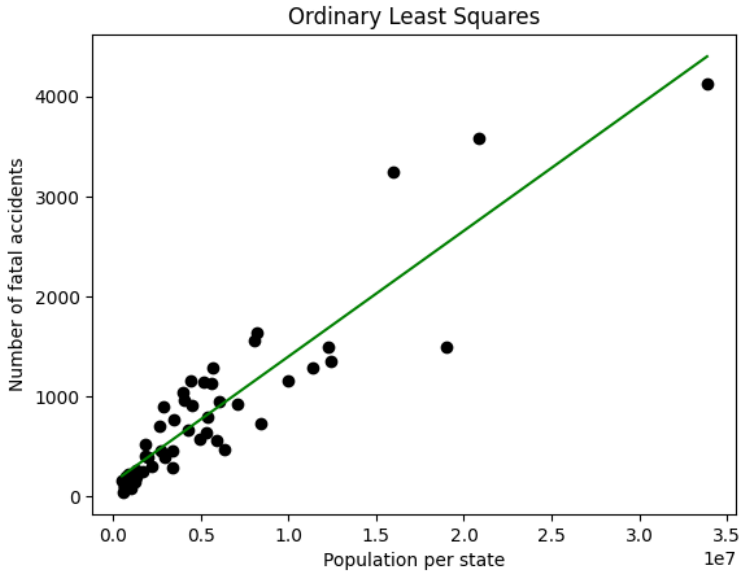

Figure 1: OLS regression.

- **Mean squared error**: 113495.26
- **Coefficient of determination**: 0.83

4 Theil-Sen Estimator

This approach consists of computing the median of the slopes of all possible lines that can be traced with any pair of points in the dataset. This approach, while having a worse time complexity than the OLS regression ($O(n^2)$ vs. $O(n)$, has some advantages, such as being more robust when faced with outliers, skewed, and even normally distributed data. Additionally, the cost of computation could be explained by the naive implementation chosen for this paper.

Using n as the number of unique pairs in the dataset, the slope of the approximating line (a_1) is:

$$a_1 = \frac{\sum i \sum j \frac{y_i - y_j}{x_i - x_j}}{n}$$

The y-intercept (a_0) can then be obtained using the slope and the mean x and y values:

$$a_0 = \bar{y} - a_1 \bar{x}$$

4.0.1 Implementation

```
x_mean = np.mean(x)
y_mean = np.mean(y)

m_acum = 0
n = len(df)

for i in range(n):
    for j in range(i + 1, n):
        m_acum += (y[j] - y[i])/(x[j] - x[i])

num_of_pairs = n*(n-1)/2
a1 = m_acum / num_of_pairs
a0 = y_mean - a1 * x_mean #y-intercept
```

4.0.2 Testing

Using the accidents dataset, the following coefficients were obtained:

$$y = 0.0001504985303040703x + 5.53758336528972$$

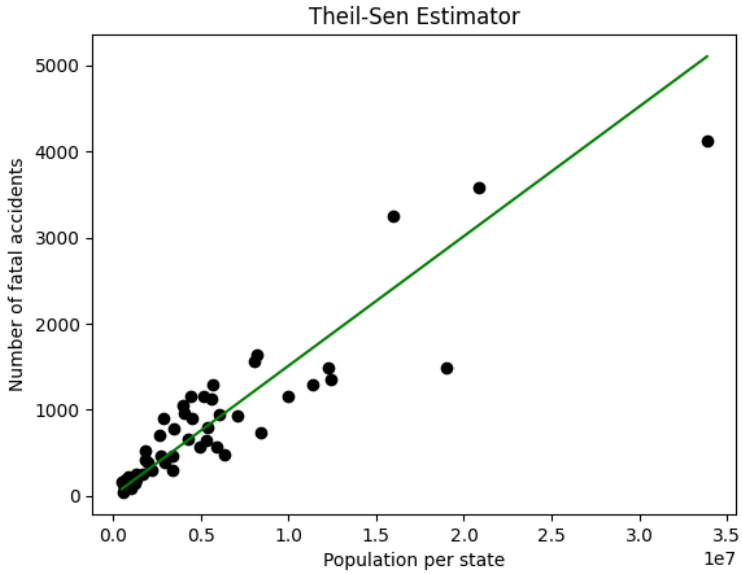

Figure 2: TSE regression.

- **Mean squared error**: 136514.966
- **Coefficient of determination**: 0.80539

5 Additional Testing

Just for fun, these models were compared to Scikit's own implementation; however, Scikit's documents indicate that the linear regression model uses the OLS method. The coefficients thrown are exactly the same as the ones obtained using the first approach, as expected.

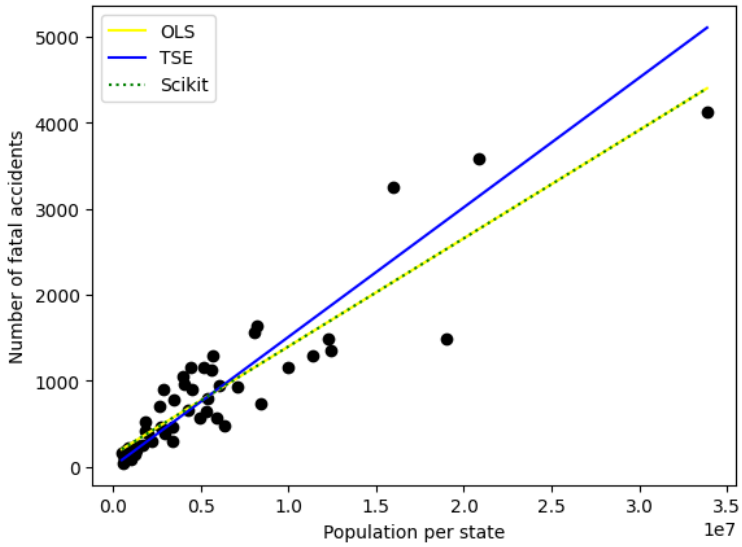

Figure 3: Method comparison.

6 Conclusions

Given how prevalent linear regression is, deconstructing and trying to come up with a method that can find an approximation line presented a fun challenge. This assignment was a great refresher for past classes, specifically numerical analysis, since we had covered the OLS method and other forms of polynomial regression, but trying to develop a method from scratch brought an interesting problem. My first approach was loosely based on the bisection method, where the slope would be calculated using a binary search from 1 to 0, minimizing the absolute distance between the samples and the line. However, I could not decide on an appropriate method for determining the y-intercept. Then, I devised a brute force approach that, as it turns out, already existed (TSE). Still, linear regression proved to be a non-trivial problem and a great introduction to optimization and approximation theory.

7 References

Cuevas, J. (2020). Handouts on Classification Algorithms. DOI:10.13140/RG.2.2.23597.03043/1

Burden, R. & Faires, J. (2001). Numerical Analysis, Ninth Edition. Brooks/Cole CENGAGE Learning. ISBN: 978-0-538-73351-9.