scipy.stats.linregress

slope, intercept, r, p, se = linregress(x, y)

result = linregress(x, y)
print(result.intercept, result.intercept_stderr)

`linregress(x, y=None, alternative='two-sided')` [source]

Calculate a linear least-squares regression for two sets of measurements.

Parameters:

x, y : *array_like*

Two sets of measurements. Both arrays should have the same length N. If only *x* is given (and `y=None`), then it must be a two-dimensional array where one dimension has length 2. The two sets of measurements are then found by splitting the array along the length-2 dimension. In the case where `y=None` and *x* is a 2xN array, `linregress(x)` is equivalent to `linregress(x[0], x[1])`.

> ***Deprecated since version 1.14.0:*** Inference of the two sets of measurements from a single argument *x* is deprecated will result in an error in SciPy 1.16.0; the sets must be specified separately as *x* and *y*.

alternative : *{'two-sided', 'less', 'greater'}, optional*

Defines the alternative hypothesis. Default is 'two-sided'. The following options are available:

- 'two-sided': the slope of the regression line is nonzero
- 'less': the slope of the regression line is less than zero
- 'greater': the slope of the regression line is greater than zero

> ***Added in version 1.7.0.***

result : `LinregressResult` *instance*

The return value is an object with the following attributes:

slope : *float*

Slope of the regression line.

intercept : *float*

Intercept of the regression line.

rvalue : *float*

The Pearson correlation coefficient. The square of `rvalue` is equal to the coefficient of determination.

pvalue : *float*

The p-value for a hypothesis test whose null hypothesis is that the slope is zero, using Wald Test with t-distribution of the test statistic. See *alternative* above for alternative hypotheses.

stderr : *float*

Standard error of the estimated slope (gradient), under the assumption of residual normality.

intercept_stderr : *float*

Standard error of the estimated intercept, under the assumption of residual normality.

p-value

  - 선형 회귀 분석에서 회귀 계수의 통계적 유의성을 평가하는 데 사용
  - 회귀 모델의 각 독립 변수와 종속 변수 간의 관계가 유의한지 여부
- null hypothesis (歸無假說)
  - p-value는 null hypothesis를 검정(testing)하는 데 사용, 독립 변수와 종속 변수 간에 관계가 없다는 가정 (회귀 계수가 0이라는 가설을 검정)
- 유의성
  - p-value가 임계값(보통 0.05)보다 작으면 null hypothesis를 기각할 수 있음, 이는 독립 변수와 종속 변수 간에 통계적으로 유의한 관계가 있다는 것을 의미
- 해석
  - p-value가 낮을수록 독립 변수가 종속 변수에 미치는 영향이 통계적으로 유의함
- 주의사항
  - 관계의 존재 여부만을 나타내며, 그 관계의 크기나 방향에 대한 정보는 제공하지 않음
  - 샘플 크기와 관련이 있으며, 큰 샘플에서는 작은 효과도 유의할 수 있음
  - 모델의 신뢰도 시 여러 다른 요소들 도 고려 (coefficient, R^2 등)

Dealing with Outliers Using Three Robust Linear Regression Models

- Outliers(이상값)
  - values that are located far outside of the expected distribution
  - cause the distributions of the features to be less well-behaved
  - can be found both in the features and the target variable
- Many possible approaches to dealing with outliers
  - removing them from the observations
  - treating them (for example, capping the extreme observations at a reasonable value)
  - using algorithms that are well-suited for dealing with such values on their own
- Huber regression
- RANSAC regression
- Theil-Sen regression

Huber regression

- Robust regression algorithm that assigns less weight to observations identified as outliers

$$\min_{w,\sigma} \sum_{i=1}^{n} \left[\sigma + H_{\varepsilon} \left(\frac{X_i w - y_i}{\sigma} \right) \sigma \right] + \alpha \left\| w \right\|_2^2$$

$$\begin{cases} \sigma: \text{ standard deviation} \\ X_i: \text{ set of features} \\ y_i: \text{ regression?s target variable} \\ w: \text{ vector of the estimated coefficients} \\ \alpha: \text{ regularization parameter} \end{cases}$$

$$H_{\varepsilon}(z) = \begin{cases} z^2 & \text{if } |z| < \varepsilon \\ 2\varepsilon |z| - \varepsilon^2 & \text{otherwise} \end{cases} \rightarrow \text{Huber loss: identify outliers}$$

when the errors follow Normal distribution with $\sigma = 1$,

$\varepsilon = 1.35 \rightarrow 95\%$ efficiency relative to the OLS regression

Random sample consensus (RANSAC) regression

- Non-deterministic algorithm that tries to separate the training data into inliers (which may be subject to noise) and outliers. Then, it estimates the final model only using the inliers
  - Select a random subset from the initial data set.
  - Fit a model to the selected random subset. (linear or other regression models)
  - Use the estimated model to calculate the residuals for all the data points in the initial data set. If (absolute residuals <= (threshold: median absolute deviation (MAD) of the target values)), then create the so-called consensus set (inliers).
  - If the current estimated model has the same number of inliers as the current best one, it is only considered to be better if it has a better score.
  - iteratively either a maximum number of times or until a special stop criterion is met

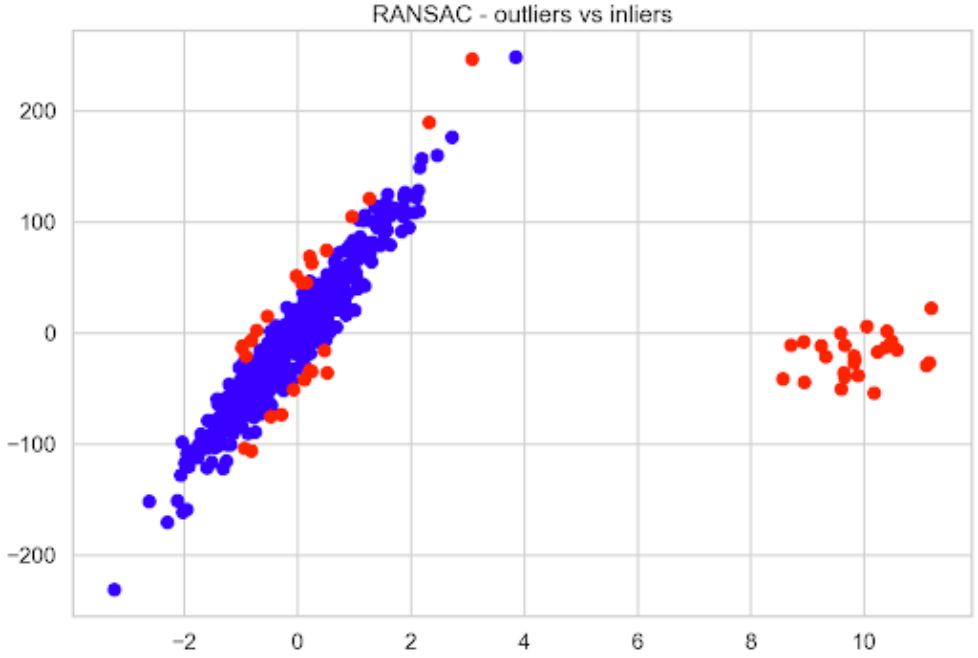
$$\left.\begin{array}{l} \mathrm{MAD} = \mathrm{median}\left(\left|X_i - m\right|\right) \\ \begin{cases} X_i: \text{ dataset} \\ m: \text{ median of a dataset} \end{cases} \end{array}\right\} \rightarrow \text{ex. } \{11,12,12,14,15,16\} \rightarrow m = 13, \mathrm{MAD} = 1.5$$

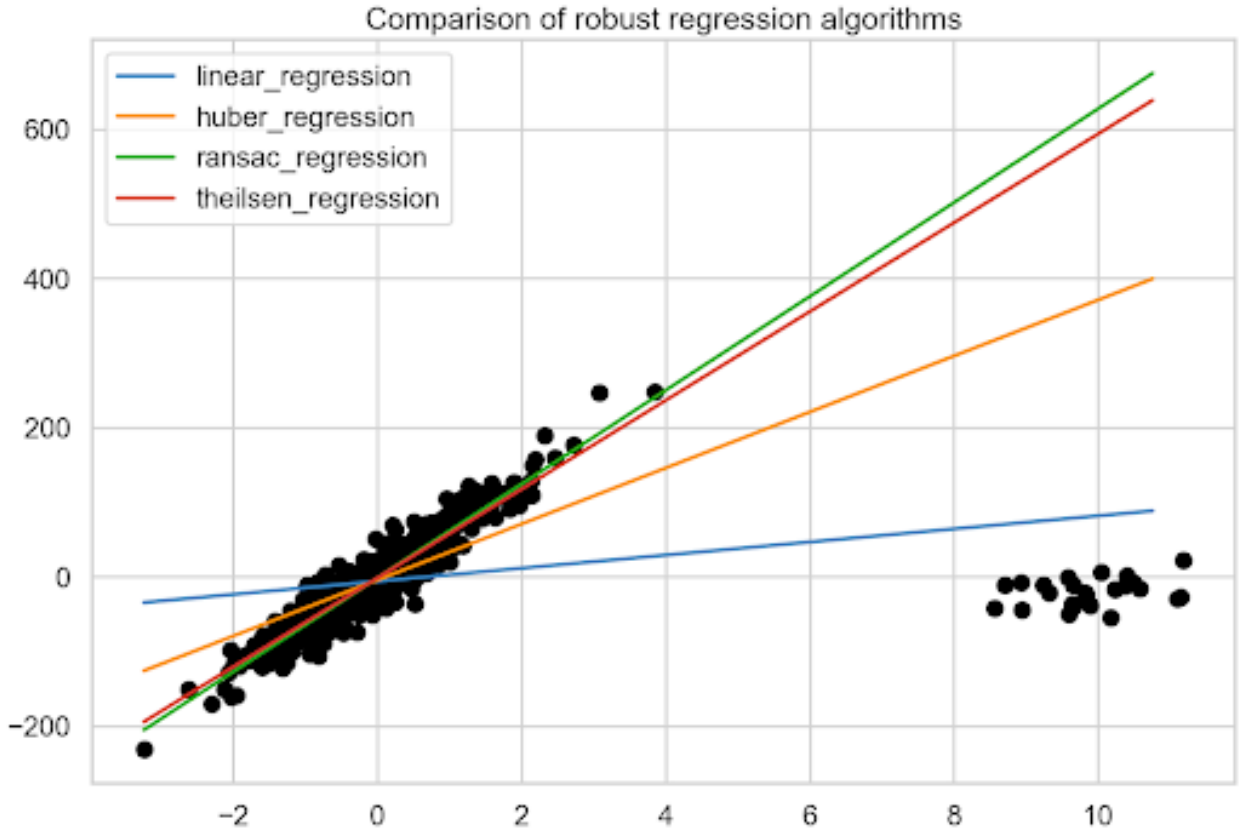
Theil-Sen regression

- Non-parametric regression method (no assumption about the underlying data distribution): fitting multiple regression models on subsets of the training data and then aggregating the coefficients at the last step
  - Calculate the least square solutions (slopes and intercepts) on subsets of size p created from all the observations in the training set X ($p \geq n\_features+1$)
  - Final slope of the line (and possibly the intercept) is defined as the (spatial) median of all the least square solutions
  - A lower p value leads to higher robustness to outliers at the cost of lower efficiency, while a higher p value leads to lower robustness and higher efficiency
  - estimator's robustness decreases quickly with the dimensionality of the problem

Comparison of the Models

- dataset of 500 observations, replace the first 25 observations (5% of the observations) with outliers, far outside of the mass of generated observations





	model	coef
0	original_coef	64.59
1	linear_regression	8.77
2	huber_regression	37.52
3	ransac_regression	62.85
4	theilsen_regression	59.49

Which robust regression algorithm is the best? As is often the case, the answer is: "it depends."

- In general, robust fitting in a high-dimensional setting is difficult.
- In contrast to Theil-Sen and RANSAC, Huber regression is not trying to completely filter out the outliers. Instead, it lessens their effect on the fit.
- Huber regression should be faster than RANSAC and Theil-Sen, as the latter ones fit on smaller subsets of the data.
- Theil-Sen and RANSAC are unlikely to be as robust as the Huber regression using the default hyperparameters.
- RANSAC is faster than Theil-Sen and it scales better with the number of samples.
- RANSAC should deal better with large outliers in the y-direction, which is the most common scenario.