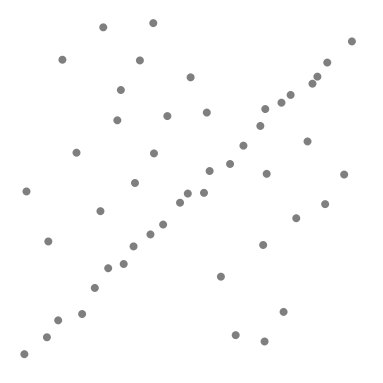
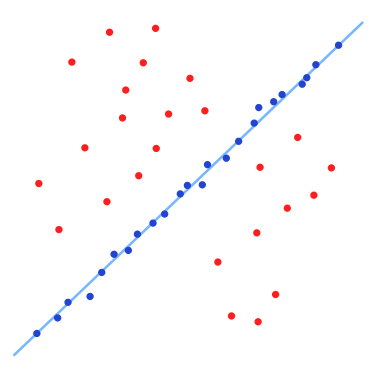
**Random sample consensus** (**RANSAC**) is an [iterative method](https://en.wikipedia.org/wiki/Iterative_method) to estimate parameters of a mathematical model from a set of observed data that contains [outliers](https://en.wikipedia.org/wiki/Outliers), when outliers are to be accorded no influence on the values of the estimates. Therefore, it also can be interpreted as an outlier detection method.

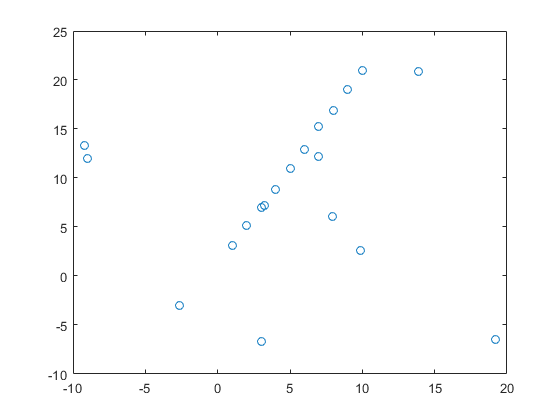
In [statistics](https://en.wikipedia.org/wiki/Statistics), an **outlier** is a [data point](https://en.wikipedia.org/wiki/Data_point) that differs significantly from other observations.[[1]](https://en.wikipedia.org/wiki/Outlier#cite_note-1)[[2]](https://en.wikipedia.org/wiki/Outlier#cite_note-2) An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the [data set](https://en.wikipedia.org/wiki/Data_set).[[3]](https://en.wikipedia.org/wiki/Outlier#cite_note-3) An outlier can cause serious problems in statistical analyses.

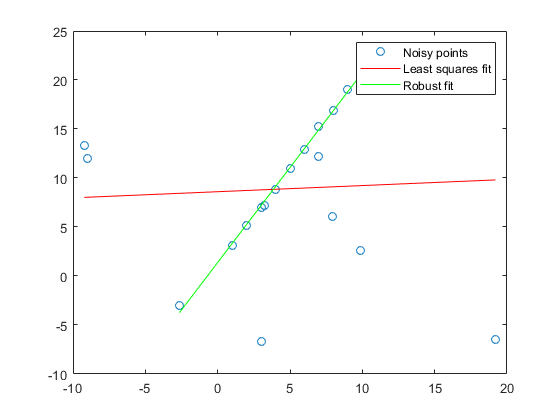
 

A data set with many outliers for which a line has to be fitted. Fitted line with RANSAC; outliers have no influence on the result.

<https://en.wikipedia.org/wiki/Random_sample_consensus>

<https://en.wikipedia.org/wiki/Outlier>





RANSAC Algorithm

Given:

data – A set of observations.

model – A model to explain observed data points.

n – Minimum number of data points required to estimate model parameters.

k – Maximum number of iterations allowed in the algorithm.

t – Threshold value to determine data points that are fit well by model.

d – Number of close data points required to assert that a model fits well to data.

Return:

bestFit – model parameters which best fit the data (or nul if no good model is found)

iterations = 0

bestFit = nul

bestErr = something really large

**while** *iterations* < *k* **do**

maybeInliers := n randomly selected values from data

maybeModel := model parameters fitted to maybeInliers

alsoInliers := empty set

**for** every point in data not in maybeInliers **do**

**if** point fits maybeModel with an error smaller than t

add point to alsoInliers

**end for**

**if** the number of elements in alsoInliers is > d **then**

// This implies that we may have found a good model

// now test how good it is.

betterModel := model parameters fitted to all points in maybeInliers and alsoInliers

thisErr := a measure of how well betterModel fits these points

**if** thisErr < bestErr **then**

bestFit := betterModel

bestErr := thisErr

**end if**

**end if**

increment iterations

**end while**

**return** bestFit

My results

