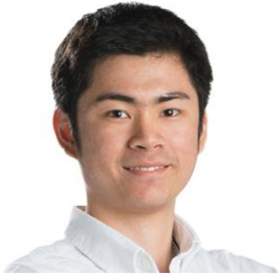
Data Mining with Outliers

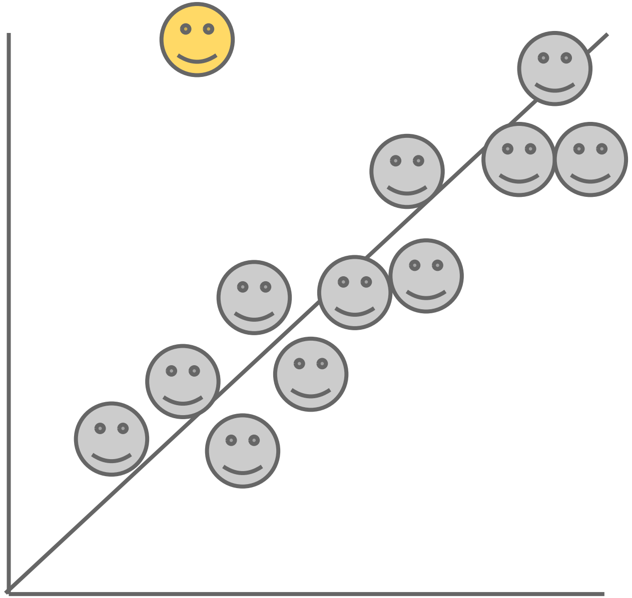
The Easy Ways to Live with Outliers Peacefully



Mark Chatchai **Wang**wiwattana

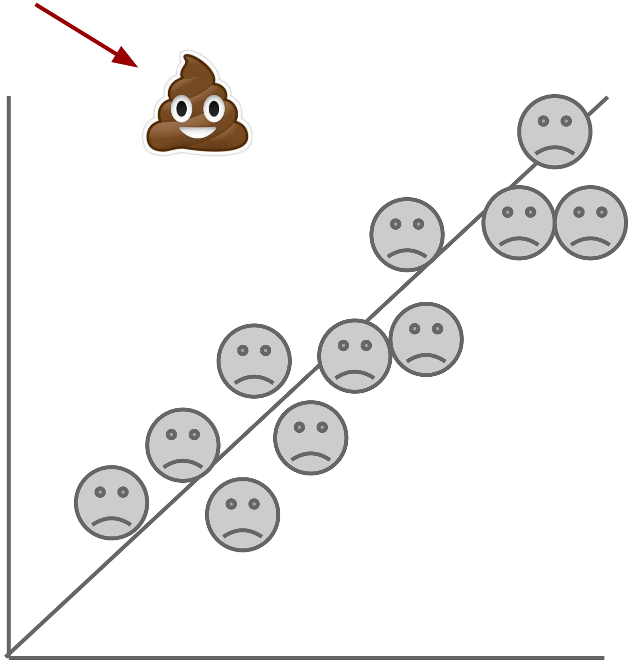
Computer Science
Lyle School of Engineering
Southern Methodist University

Outlier

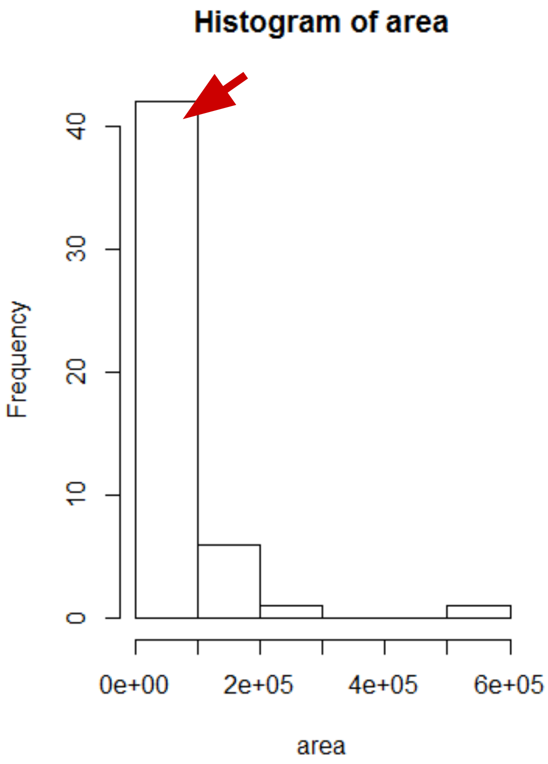


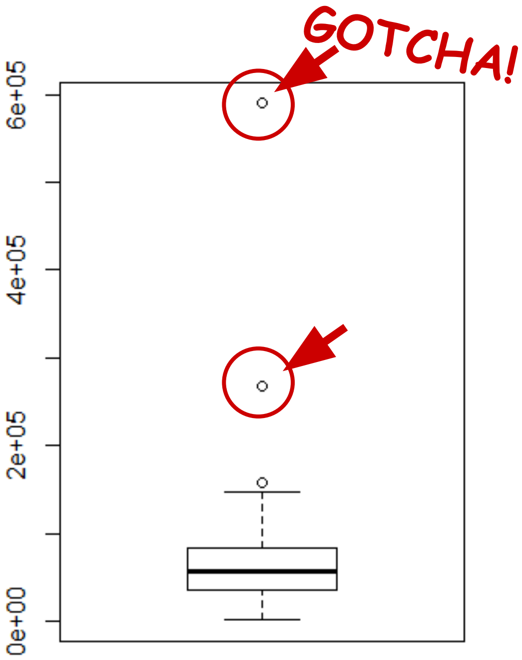
Outlier

What we feel about him



We can Manually Detect Outliers by using Histogram or Boxplot.





Detecting outliers in Dependent variables may be challenging

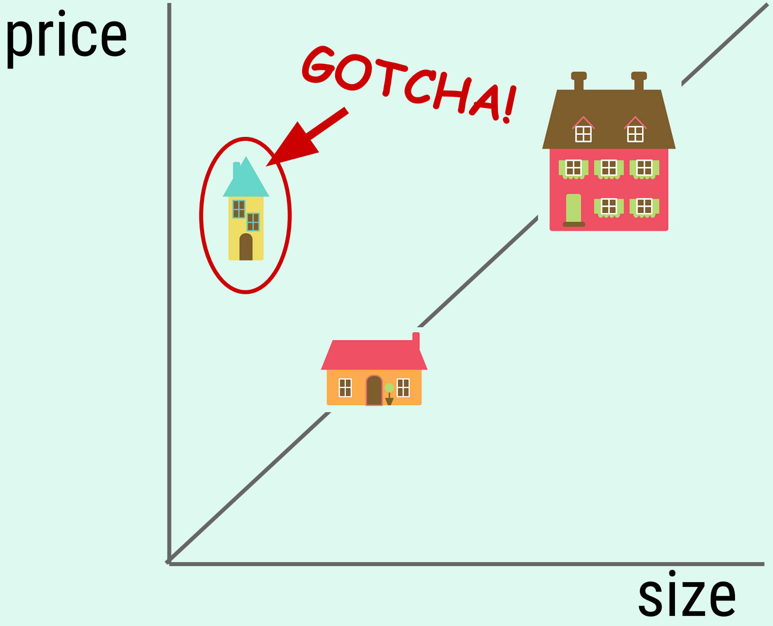
$150,000

$550,000

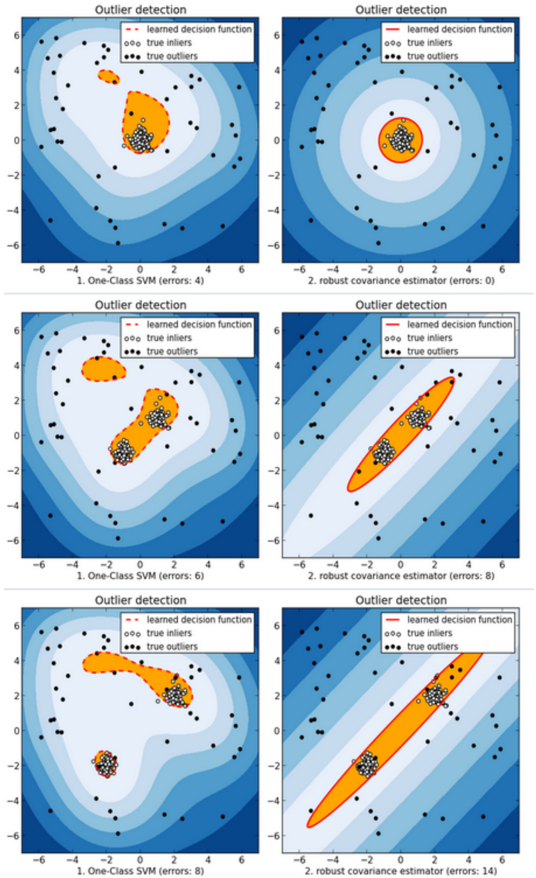
Detecting outliers in Dependent variables may be challenging

$150,000 $550,000 $350,000

When we plot those variables, we will see the outlier



We can automatically detect outliers by using



- Fitting an elliptic envelop
- One-Class-SVM

http://scikit-learn.org/stable/modules/outlier_detection.html

What if we cannot remove outliers?

Ex. in Real-time Applications?

“**Robust statistics** is a family of theories and techniques for estimating the parameters of a parametric model while dealing with deviations from idealized assumptions”

http://egret.psychol.cam.ac.uk/statistics/local_copies_of_sources_Cardinal_and_Aitken_ANOVA/A_Brief_Overview_of_Robust_Statistics.htm

http://cran.r-project.org/web/views/Robust.html

Robust statistics is the statistical procedure that can **resist** to **outliers** and non-normality distribution

http://egret.psychol.cam.ac.uk/statistics/local_copies_of_sources_Cardinal_and_Aitken_ANOVA/A_Brief_Overview_of_Robust_Statistics.htm

http://cran.r-project.org/web/views/Robust.html

NOT Robust	Robust
Mean	Trim Mean
	Median
Standard Deviation	Median Absolute Deviation
	Interquartile Range

Ordinary Least Squares Linear Regression, PCA, SVM

?

ELSEVIER

Introduction to Robust Estimation & Hypothesis Testing

Rand Wilcox

3RD EDITION

Copyrighted Material

CRAN Task View: Robust Statistical Methods

Maintainer: Martin Maechler
Contact: Martin.Maechler at R-project.org
Version: 2014-12-07

Robust (or "resistant") methods for statistics modelling have been available in S from the very beginning in the 1980s; and then in R in package `stats`. Examples are `median()`, `mean(*, trim =. )`, `mad()`, `IQR()`, or also `fivenum()`, the statistic behind `boxplot()` in package `graphics`) or `lowess()` (and `loess()`) for robust nonparametric regression, which had been complemented by `runmed()` in 2003. Much further important functionality has been made available in recommended (and hence present in all R versions) package MASS (by Bill Venables and Brian Ripley, see *the* book Modern Applied Statistics with S). Most importantly, they provide `rlm()` for robust regression and `cov.rob()` for robust multivariate scatter and covariance.

This task view is about R add-on packages providing newer or faster, more efficient algorithms and notably for (robustification of) new models.

Please send suggestions for additions and extensions to the task view maintainer .

An international group of scientists working in the field of robust statistics has made efforts (since October 2005) to coordinate several of the scattered developments and make the important ones available through a set of R packages complementing each other. These should build on a basic package with "Essentials", coined robustbase with (potentially many) other packages building on top and extending the essential functionality to particular models or applications. Further, there is the quite comprehensive package robust, a version of the robust library of S-PLUS, as an R package now GPLicensed thanks to Insightful and Kjell Konis. Originally, there has been much overlap between 'robustbase' and 'robust', now robust *depends* on robustbase, the former providing convenient routines for the casual user where the latter will contain the underlying functionality, and provide the more advanced statistician with a large range of options for robust modeling.

We structure the packages roughly into the following topics, and typically will first mention functionality in packages robustbase and robust.

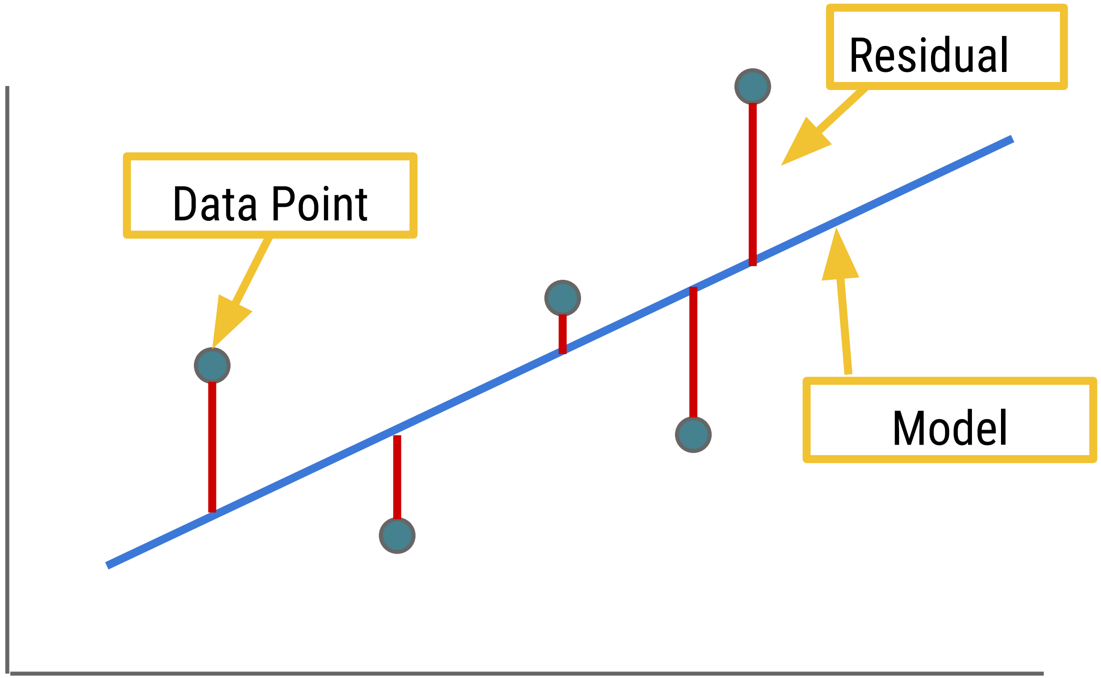
- *Regression (Linear, Generalized Linear, Nonlinear Models, incl. Mixed Effects)* : `lmrob()` (robustbase) and `lmRob()` (robust) where the former uses the latest of the fast-S algorithms and heteroscedasticity and autocorrelation corrected (HAC) standard errors, the latter makes use of the M-S algorithm of Maronna and Yohai (2000), automatically when there are factors among the predictors (where S-estimators (and hence MM-estimators) based on resampling typically badly fail). The `ltsReg()` and `lmrob.S()` functions are available in robustbase, but rather for comparison purposes. `rlm()` from MASS had been the first widely available implementation for robust linear models, and also one of the very first MM-estimation implementations. robustreg provides very simple M-estimates for linear regression (in pure R). Note that Koenker's quantile regression package quantreg contains L1 (aka LAD, least absolute deviations)-regression as a special case, doing so also for nonparametric regression via splines. Quantile regression (and hence L1 or LAD) for mixed effect models, is available in package lqmm, whereas an *MM-like* approach for robust linear **mixed effects** modeling is available from package robustlmm. Package mblm 's function `mblm()` fits median-based (Theil-Sen or Siegel's repeated) simple linear models. Package TEEReg provides trimmed elemental estimators for linear models. Generalized linear models (GLMs) are provided both via `glmrob()` (robustbase) and `glmRob()` (robust), where package robustloggamma focuses on generalized log gamma models. Robust ordinal regression is provided by packages ror (MCDA) and rorutadis (UTADIS). Robust Nonlinear model

CRAN Task View: Robust Statistical Methods

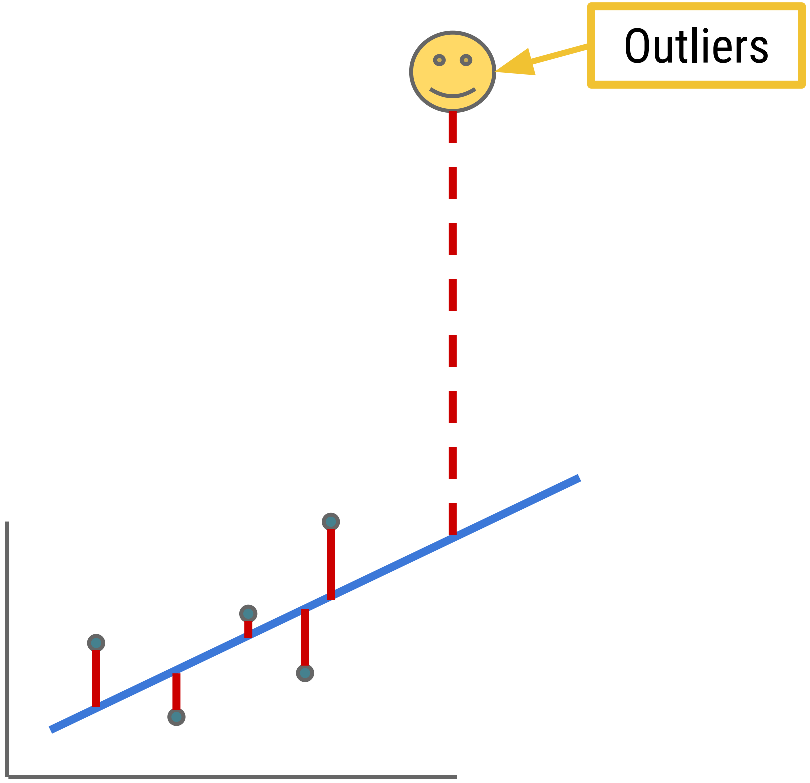
http://cran.r-project.org/web/views/Robust.html

How about a linear regression?

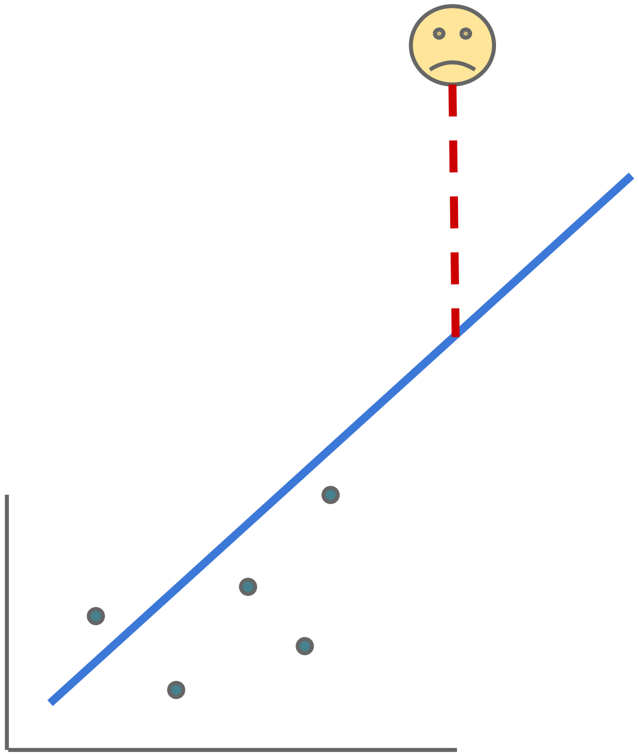
This is a simple linear regression
(The Least Square Linear Regression)



One outlier can mess up a whole model.



One outlier can mess up a whole model.



Random Sample Consensus (RANSAC)

This technique can be used to find inliers for any type of modeling fitting.

Algorithm:

1. Randomly select a sample of *s* data points from *S* and instantiate the model from this subset.
2. Determine the set of data points *Si* which are within a distance threshold *t* of the model. The set Si is the consensus set of the sample and defines the inliers of *S*.
3. If the size of *Si* (the number of inliers) is great than some threshold *T*, re-estimate the model using all the points in *Si*.
4. After *N* trials the largest consensus set *Si* is selected, and the model is re-estimated using all the points in the subset *Si*.

Fischler, M. A., & Bolles, R. C. 1984. Random sampling consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the Associations for Computing Machinery, 24(26), 381-395.

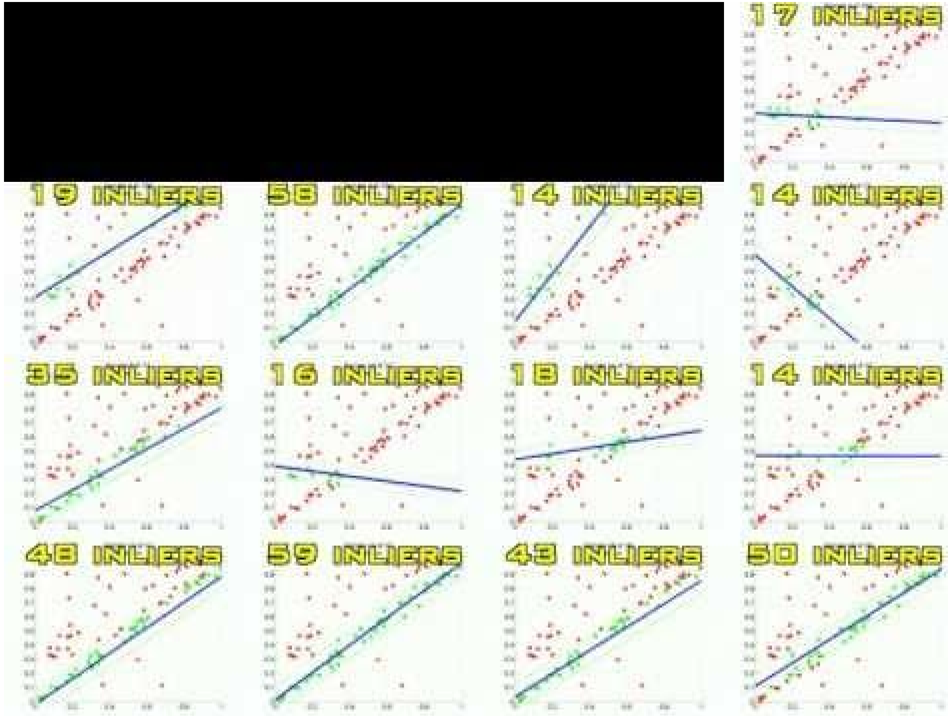
Random Sample Consensus (RANSAC)

This technique can be used to find inliers for any type of modeling fitting.

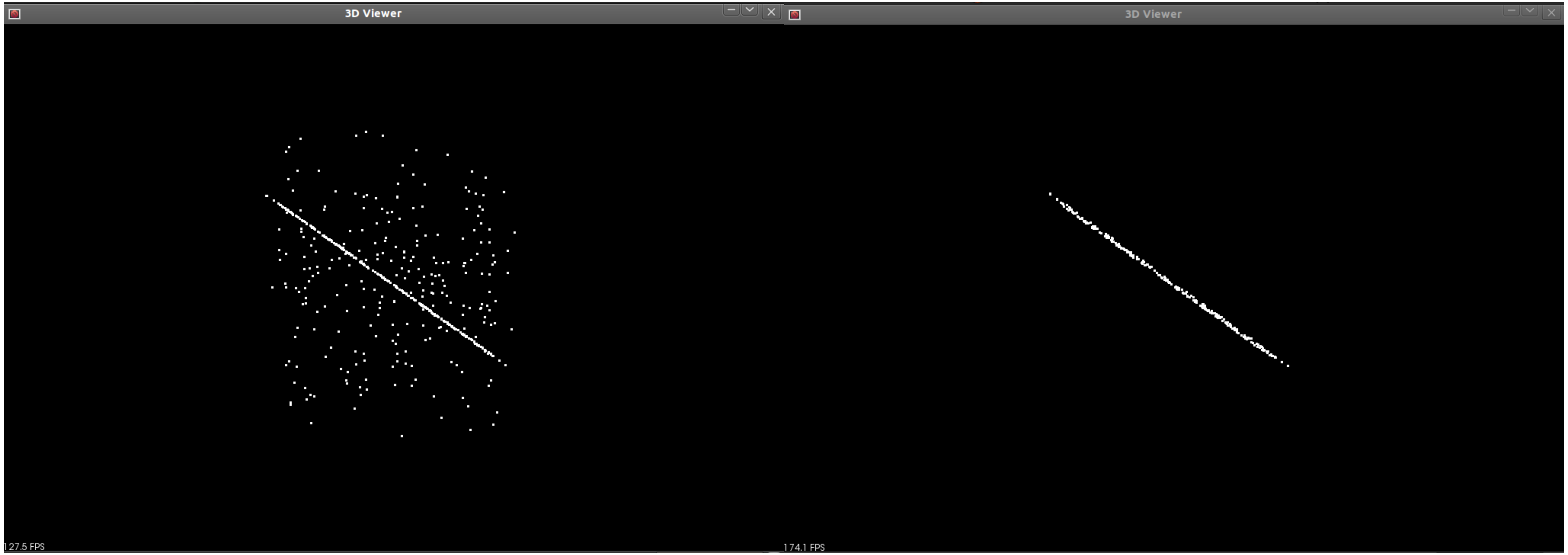
Algorithm:

1. Randomly selecting a subset of the data set.
2. Fitting a model to the selected subset.
3. Determining the number of outliers.
4. Repeating steps 1-3 for a prescribed number of iterations

Fischler, M. A., & Bolles, R. C. 1984. Random sampling consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the Associations for Computing Machinery, 24(26), 381-395.


17 INLIERS
19 INLIERS
58 INLIERS
14 INLIERS
14 INLIERS
35 INLIERS
16 INLIERS
18 INLIERS
14 INLIERS
48 INLIERS
59 INLIERS
43 INLIERS
50 INLIERS

It works with a line

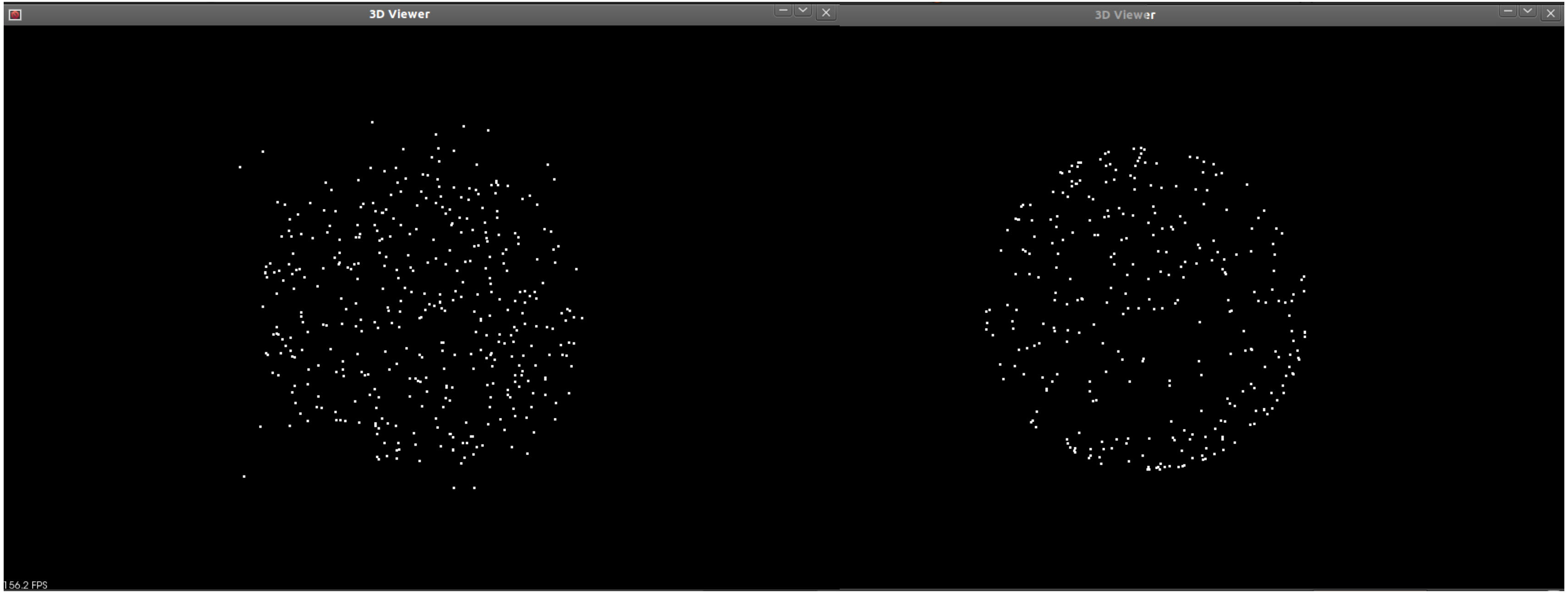


Input

output

http://pointclouds.org/documentation/tutorials/random_sample_consensus.php

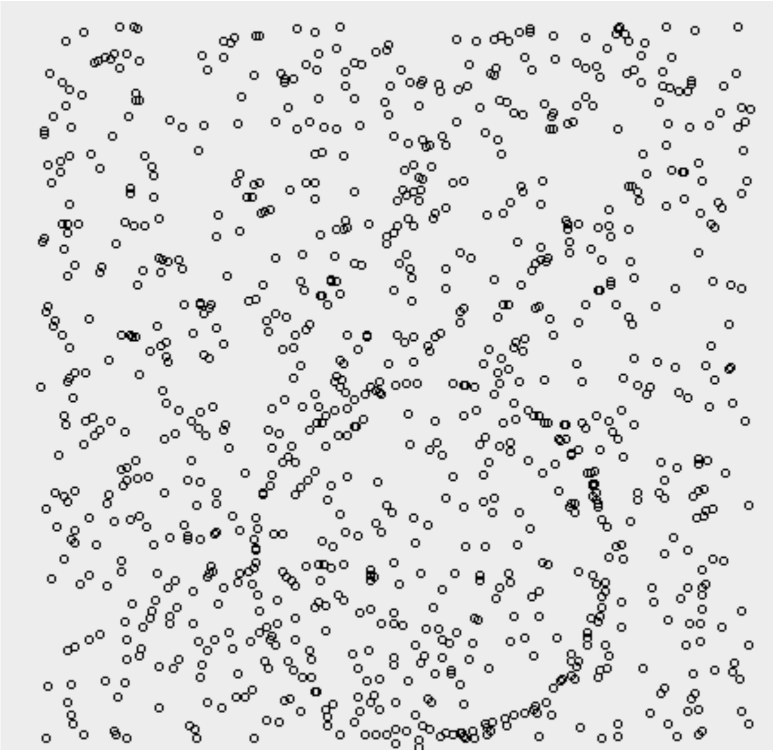
It works with a sphere



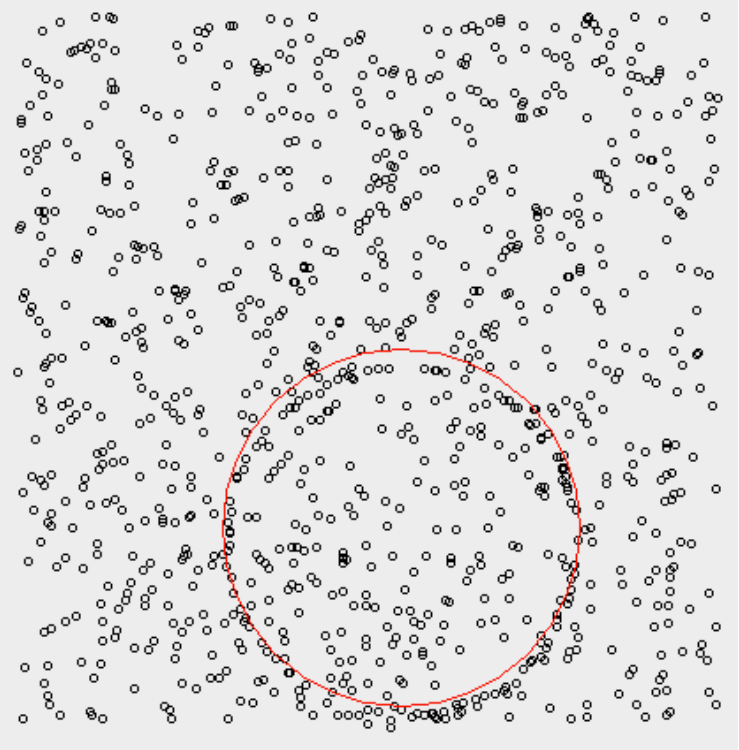
Input output

http://pointclouds.org/documentation/tutorials/random_sample_consensus.php

It works with a circle



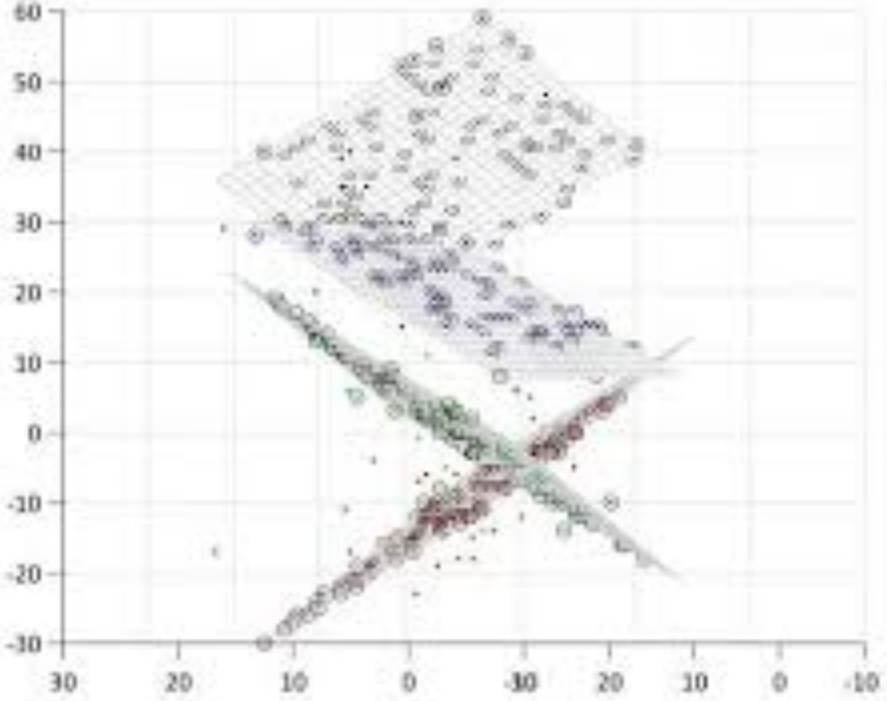
Input



output

https://github.com/oleander/ransac-and-hough-transform-java

It works with a plane



www.timzaman.com

How many iteration do we need?

ϵ = the probability that a point is an outlier

s = a sample size

p = a probability of one sample has no outlier.

N = number of iterations

$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)}.$$

Fischler, M. A., & Bolles, R. C. 1984. Random sampling consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the Associations for Computing Machinery, 24(26), 381-395.

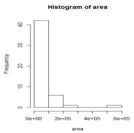
How many iteration do we need?

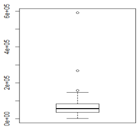
Sample size	Proportion of Outliers ϵ						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Hartley and Zisserman, 2000: The number N of samples required to ensure, with a probability p = 0.99, that at least one sample has no outliers for a given sample size s and a proportion of outliers ϵ.

To sum up

- Manually detect and filter outlier
  - Histogram
  - Boxplot

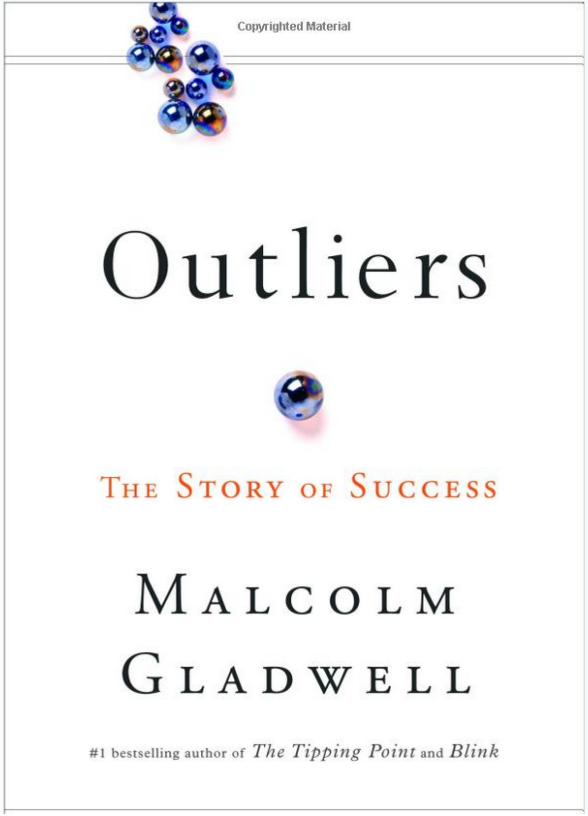




- Robust Statistics

NOT Robust	Robust
Mean	Trim Mean
	Median
Standard Deviation	Median Absolute Deviation
	Interquartile Range

- RANSAC Algorithm



The world of "outliers"--the best and the brightest, the most famous and the most successful.

- Software billionaires
- Great soccer player
- The Beatles

Questions & Answer