Visualizing Big Data Outliers through Distributed Aggregation

Leland Wilkinson

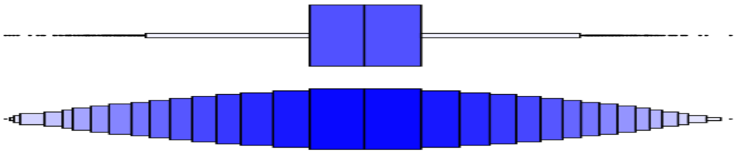

Fig. 1. Outliers revealed in a box plot [72] and letter values box plot [36]. These plots are based on 100,000 values sampled from a Gaussian (Standard Normal) distribution. By definition, the data contain no probable outliers, yet the ordinary box plot shows thousands of outliers. This example illustrates why ordinary box plots cannot be used to discover probable outliers.

Abstract— Visualizing outliers in massive datasets requires statistical pre-processing in order to reduce the scale of the problem to a size amenable to rendering systems like D3, Plotly or analytic systems like R or SAS. This paper presents a new algorithm, called `hdoutliers`, for detecting multidimensional outliers. It is unique for a) dealing with a mixture of categorical and continuous variables, b) dealing with big-*p* (many columns of data), c) dealing with big-*n* (many rows of data), d) dealing with outliers that mask other outliers, and e) dealing consistently with unidimensional and multidimensional datasets. Unlike ad hoc methods found in many machine learning papers, `hdoutliers` is based on a distributional model that allows outliers to be tagged with a probability. This critical feature reduces the likelihood of false discoveries.

Index Terms—Outliers, Anomalies.

1 Introduction

Barnett and Lewis [6] define an outlier in a set of data as "an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data." They go on to explain that their definition rests on the assumption that the data constitute a random sample from a population and that outliers can be represented as points in a vector space of random variables. This restriction, shared in this paper, allows us to assign a probability to our judgments that a point or points are outliers. It excludes other types of anomalies (negative counts, pregnant males) that can appear in a dataset and are detectable through logic or knowledge of the world. All outliers are anomalies, but some anomalies are not outliers.

This paper is concerned with the interplay of visual methods and outlier detection methods. It is not an attempt to survey the vast field of outlier detection or to cover the full range of currently available methods. For general introductions, see the references at the beginning of the Related Work section below.

1.1 Contributions

Our contributions in this paper are:

- We demonstrate why the classical definition of an outlier (a large distance of a point from a central location estimate (mean, median, etc.) is unnecessarily restrictive and often involves a circularity.
- We introduce a new algorithm, called `hdoutliers`, for multidimensional outliers on *n* rows by *p* columns of data that addresses the curse of dimensionality (large *p*), scalability (large *n*), categorical variables, and non-Normal distributions. This algorithm is designed to be paired with visualization methods that can help an analyst explore unusual features in data.
- We demonstrate why other visual analytic tools cannot reliably be used to detect multidimensional outliers.
- We introduce some novel applications of outlier detection and accompanying visualizations based on `hdoutliers`.

2 Related Work

There are several excellent books on outliers written by statisticians [6, 31, 64, 71]. Statisticians have also written survey papers [38, 3]. Computer scientists have written books and papers on this topic as well [1, 14, 35]. The latter include surveys of the statistical sources.

2.1 Univariate Outliers

The detection of outliers in the observed distribution of a single variable spans the entire history of outlier detection [70, 6]. It spans this history not only because it is the simplest formulation of the problem, but also because it is deceptively simple.

2.1.1 The Distance from the Center Rule

The word *outlier* implies lying at an extreme end of a set of ordered values – far away from the center of those values. The modern history of outlier detection emerged with methods that depend on a measure of centrality and a distance from that measure of centrality. As early as the 1860's, Chauvenet (cited in [6]) judged an observation to be an outlier if it lies outside the lower or upper $1/(4n)$ points of the Normal distribution. Barnett and Lewis [6] document many other early rules that depend on the Normal distribution but fail to distinguish between population and sample variance.

Grubbs [28], in contrast, based his rule on the sample moments of the Normal:

$$G = \frac{\max\limits_{1 \le i \le n} |x_i - \bar{x}|}{s}$$

where $\bar{x}$ and s are the sample mean and standard deviation, respectively.

Grubbs referenced G against the t distribution in order to spot an upper or lower outlier:

- *Leland Wilkinson is Chief Scientist at H2O.ai and Adjunct Professor of Computer Science at UIC. E-mail: leland.wilkinson@gmail.com.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx/

$$G > \frac{n-1}{\sqrt{n}} \sqrt{\frac{t^2_{\alpha/(2n),n-2}}{n-2+t^2_{\alpha/(2n),n-2}}}$$

If one knows that the values on a variable are sampled randomly from a Normal distribution and if the estimates of location and scale are unbiased and if one wishes to detect only the largest absolute outlier, it is a valid test.

Unfortunately, the usual sample estimates of the mean and standard deviation are not robust against outliers. So we have a circularity problem. We assume a null distribution (say, the Normal), estimate its parameters, and then use those estimates to test whether a point could have plausibly come from that distribution. But if our alternative hypothesis is that it does not (the usual case), then the outlier should not be included in the estimation. Barnett and Lewis [6] discuss this problem in more detail, where they distinguish *inclusive* and *exclusive* methods. They, as well as [64], also discuss robust estimation methods for overcoming this circularity problem.

Barnett and Lewis discuss other detection methods for non-Normal distributions. The same principals apply in these cases, namely, that the sample is random, the population distributions are known and that the parameter estimates are unbiased.

2.1.2 The Box Plot Rule

A box-plot graphically depicts a batch of data using a few summary statistics called *letter values* [72, 25]. The letter values in Tukey's original definition are the median and the *hinges* (medians of the upper and lower halves of the data). The hinge values correspond closely, but not necessarily, to the lower quartile (Q1) and the upper quartile (Q3). Tukey called the difference between the hinges the *Hspread*, which corresponds closely to the quantity Q3–Q1, or the Inter Quartile Range (IQR). In Tukey's version of the box-plot (see the upper panel of Figure 1), a box is drawn to span the *Hspread*. The median is marked inside the box. Whiskers extend from the edges of the box to the farthest upper and lower data points (*Adjacent values*) inside the so-called *inner fences*. The upper inner fence is the

$$upperhinge + 1.5 \times Hspread$$

and the lower inner fence is the

$$lowerhinge - 1.5 \times Hspread$$

Any data point beyond the *Adjacent values* is plotted as an outlying point. [1]

Tukey designed the box plot (he called it a *schematic plot*) to be drawn by hand on a small batch of numbers. The whiskers were designed not to enable outlier detection, but to locate the display on the interval that supports the bulk of the values. Consequently, he chose the *Hspread* to correspond roughly to three standard deviations on Normally distributed data. This choice led to two consequences: 1) it does not apply to skewed distributions, which constitute the instance many advocates think is the best reason for using a box plot in the first place, and 2) it does not include sample size in its derivation, which means that the box plot will falsely flag outliers on larger samples. As Dawson [19] shows, "regardless of size, at least 30% of samples drawn from a Normally-distributed population will have one or more data flagged as outliers." The top panel of Figure 1 illustrates this problem for a sample of 100,000 Normally distributed numbers. Thousands of points are denoted as outliers in the display.

To deal with the skewness problem, Hubert and Vandervieren [37] and others have suggested modifying the fences rule by using a robust estimate of skewness. By contrast, Tukey's approach for this problem involved transforming the data through his *ladder of powers* [72] before drawing the box plot.

The Letter-Value-Box-Plot [36] was designed to deal with the second problem. The authors compute additional letter values (splitting the splits) until a statistical measure of fit is satisfied. Each letter-value region is represented by a rectangle. The lower panel of Figure 1 shows the result. On the same 100,000 Normal variables, only two points are identified as outliers.

2.1.3 The Gaps Rule

Suppose that we do not know the population distribution and suppose, further, that our idea of outliers is that they do not belong to the generating distribution we suspect underlies our data. Figure 2 shows two dotplots of batches of data that have the same mean and standard deviation. Absent knowledge of the parametric distribution, we cannot apply the usual outlier detection algorithms. Furthermore, we are more inclined to say the largest point in the right dot plot is an outlier, whereas the largest point in the left plot, having the same score, is not.

A simple example emphasizes this point. Suppose we give a test to 100 students and find the mean score is 50 and the standard deviation is 5. Among these students, we find one perfect score of 100. The next lower score is 65. We might be inclined to suspect the student with a score of 100 is a genius or a cheat. And if there were three students with perfect scores in this overall distribution, we might suspect cheating even more. On the other hand, if the perfect score is at the top of a chain of scores spaced not more than 5 points apart, we might be less suspicious. Classical outlier tests would not discriminate among these possibilities.

These considerations and others led to a different criterion for discovering outliers. Namely, we should look for *gaps* (*spacings*) between the ordered values rather than extremities. A consequence of this point of view is that we can identify unusual scores in the *middle* of distributions as well as in the extremes, as long as they are separated from other scores by a large gap.

Dixon [20] headed in this direction by developing an outlier test based on the gap between the largest point and the second largest point, standardized by the range of scores. His test was originally based on a Normal distribution, but in subsequent publications, he developed nonparametric variations. Dixon tabulated percentage points for a range of Q statistics.

$$Q = \frac{x_n - x_{n-1}}{x_n - x_1}$$

Tukey [72] considered the more general question of identifying gaps anywhere in a batch of scores. Wainer and Schacht [75] adapted Tukey's gapping idea for a version of the test that weighted extreme values more than middle ones. They derived an approximate z score that could be used to test the significance of gaps.

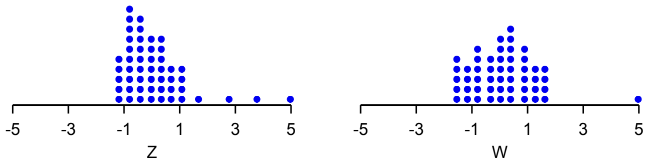

Fig. 2. Dot plots of small batches of data with comparable means and standard deviations.

Burridge and Taylor [12] developed an outlier test based on the extreme-value distribution of gaps between points sampled from the Exponential family of distributions:

$$f(x_i;\theta_i,\phi) = \exp\left[\frac{x\theta - a(\theta)}{b(\phi)} + c(x,\phi)\right]$$

where x is a scalar or vector, θ is a scalar or vector of parameters, ϕ is a scale parameter, and $a(.), b(.), c(.)$ are functions. This family of mathematical distributions is quite large (including the Normal, Exponential, Gamma, Beta, Bernoulli, Poisson, and many others).

[1] Few statistics packages produce box plots according to Tukey's definition [25]. Surprisingly, the boxplot function in the core *R* package does not, despite its ancestry inside Tukey's group at Bell Laboratories.

2.2 Multivariate Outliers

This section covers methods explicitly designed for higher-dimensional outlier detection.

2.2.1 Mahalanobis Distance

The squared Mahalanobis distance (D^2) of a multidimensional point $\mathbf{x}$ from the centroid of a multivariate Normal distribution described by covariance matrix Σ and centroid μ is

$$D^2 = (\mathbf{x} - \mu)'\Sigma^{-1}(\mathbf{x} - \mu)$$

Figure 3 shows how this works in two dimensions. The left panel shows a bivariate Normal distribution with level curves inscribed at different densities. The right panel shows the same level curves as horizontal slices through this mountain. Each is an ellipse. Distance to the centroid of the ellipses is measured differently for different directions. The weights are determined by the covariance matrix Σ. If Σ is an identity matrix, then D^2 is equivalent to squared Euclidean distance.

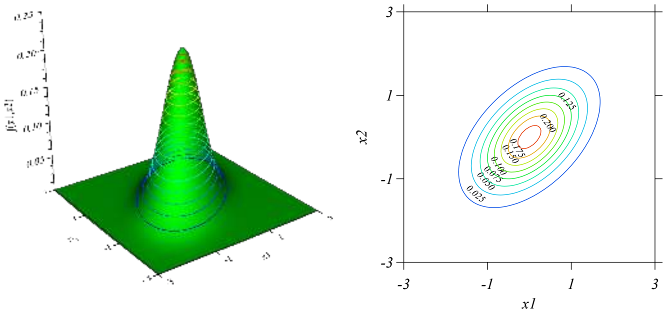

Fig. 3. Mahalanobis Distance. The left panel shows a bivariate Normal distribution. The right shows level curves for that distribution. Each curve corresponds to a value of D^2.

With real data, we substitute sample estimates of μ and Σ in the Mahalanobis formula. If the assumptions of the test are true and our estimates are unbiased, the squared Mahalanobis distance is a chi-square variate with p degrees of freedom. As with univariate outlier tests based on a Normality assumption, this test is valid only if the assumption of multivariate Normality is met. Unfortunately, this is seldom true for real data and, furthermore, estimates of the covariance matrix and centroid are far from robust. Consequently, this outlier test has limited applicability.

Rousseeuw and Van Zomeren [63] introduce a robust Mahalanobis Distance estimator that can be used to overcome some of these problems. Ram Gnanadesikan [26] discusses applications of Gamma probability plots to these multivariate problems (Gamma is a generalization of the chi-square distribution). They can be interpreted similarly to the way univariate probability plots are interpreted.

2.2.2 Multivariate Gap Tests

Multivariate data do not have a simple ordering for computing gaps between adjacent points. There have been several attempts at getting around this problem. Rohlf [60] proposed using the edge lengths of the geometric minimum spanning tree (MST) as a single distribution measure. Assuming these edges approximate a gamma distribution, one could construct a gamma probability plot on them or examine the upper tail for judgments on outliers. There are problems with this method, however, when variates are correlated [13]. Similar methods based on the MST have been proposed [52, 57], but they suffer from the same problem.

2.2.3 Clustering

A popular multivariate outlier detection method has been to cluster the data and then look for any points that are far from their nearest cluster centroids [83, 39, 56, 40, 59]. This method works reasonably well for moderate-size datasets with a few singleton outliers. Most clustering algorithms do not scale well to larger datasets, however.

A related approach, called Local Outlier Factor (LOF) [11], is similar to density-based clustering [30]. Like DBSCAN clustering [22], it is highly sensitive to the choice of input parameter values.

Most clustering methods are not based on a probability model (see [24] for an exception) so they are susceptible to false negatives and false positives.

3 A New Multivariate Outlier Algorithm

The new algorithm `hdoutliers` is designed to meet several criteria uniquely at once:

- It allows us to identify outliers in a mixture of categorical and continuous variables.
- It deals with the curse of dimensionality by exploiting random projections for big-p (number of dimensions).
- It deals with big-n (number of points) by exploiting a one-pass algorithm to aggregate the data.
- It deals with the problem of *masking* [6], in which clusters of outlying points can elude detection by traditional methods.
- It works for both single-dimensional and multi-dimensional data.

`hdoutliers` is introduced in Section 3.1 and explained in Section 3.2.

3.1 The Algorithm

1. If there are any categorical variables in the dataset, convert each categorical variable to a continuous variable by using Correspondence Analysis [27].
2. If there are more than 10,000 columns, use random projections to reduce the number of columns to $p = 4\log n/(\varepsilon^2/2 - \varepsilon^3/3)$, where ε is the error bound on squared distances.
3. Normalize the columns of the resulting n by p matrix X.
4. Let $row(i)$ be the ith row of X.
5. Let $\delta = .1/(\log n)^{1/p}$.
6. Initialize *exemplars*, a list of exemplars with initial entry $[row(1)]$.
7. Initialize *members*, a list of lists with initial entry $[1]$; each exemplar will eventually have its own list of affiliated member indices.
8. Now do one pass through X:

```
forall the $row(i)$, $i = 1, \ldots, n$ do
    $d$ = distance to closest exemplar, found in exemplars
    if $d < \delta$ then
        add $i$ to members list associated with closest exemplar
    else
        add $row(i)$ to exemplars
        add new list to members, initialized with $[i]$
    end
end
```

9. Now compute nearest-neighbor distances between all pairs of exemplars in the *exemplars* list.
10. Fit an Exponential distribution to the upper tail of the nearest-neighbor distances and compute the upper $1 - \alpha$ point of the fitted cumulative distribution function (CDF).
11. For any exemplar that is significantly far from all the other exemplars based on this cutpoint, flag all entries of *members* corresponding to *exemplar* as outliers.

3.2 Comments on the Algorithm

1. Correspondence Analysis (CA) begins by representing a categorical variable with a set of dummy codes, one code (1 or 0) for each category. These codes comprise a matrix of 1's and 0's with as many columns as there are categories on that variable. We then compute a principal components decomposition of the covariance matrix of the dummy codes. This analysis is done separately for each of k categorical variables in a dataset. CA scores on the rows are computed on each categorical variable by multiplying the dummy codes on that row's variable times the eigenvectors of the decomposition for that variable. Computing the decomposition separately for each categorical variable is equivalent to doing a multiple correspondence analysis separately for each variable instead of pooling all the categorical variable dummy codes into one matrix. This application of CA to deal with visualization of nomial data was first presented in [61].

2. Euclidean distances in high-dimensional space converge toward a single value [8, 33]. This can cause problems with nearest-neighbor calculations. Consequently, we exploit the Johnson-Lindenstrauss lemma [42] to produce a lower dimensional space for our calculations. This lemma states that if a metric on X results from an embedding of X into a Euclidean space, then X can be embedded in R^p with distortion less than $1+\varepsilon$, where $p \sim O(\varepsilon^2 \log n)$. Remarkably, this embedding is achieved by projecting onto a p-dimensional subspace using random Gaussian coefficients. Because our algorithm depends only on a similarity transformation of Euclidean distances, we can logarithmically reduce the complexity of the problem through random projections and avoid the curse of dimensionality. The number of projected columns based on the formula in this step was based on $\varepsilon = .2$ for the analyses in this paper. The value 10,000 is the lower limit for the formula's effectiveness in reducing the number of dimensions when $\varepsilon = .2$.

3. X is now bounded by the unit (hyper) cube. Normalization is commonly done in clustering [68] and outlier detection [11, 48]. This prevents variables with large variances having disproportional influence on Euclidean distances. See [18] for an illustration of the effect of transformations on the distribution of point sets. When variables are on a common scale (as in the Fisher/Anderson Iris dataset), normalization is generally not needed.

4. A *row* represents a p-dimensional vector in a finite vector space.

5. The value of δ is designed to be well below the expected value of the distances between $n(n-1)/2$ pairs of points distributed randomly in a p dimensional unit hypercube [32]. This δ value defines the boundary of an exemplar neighborhood; it is the radius of each ball.

6. The *exemplars* list contains a list of row values representing points within an exemplar neighborhood..

7. The *members* list of lists contains one list of indices for each exemplar pointing to members of that exemplar's neighborhood.

8. The Leader algorithm [30] in this step creates exemplar-neighborhoods in one pass through the data. It is equivalent to centering balls in p dimensional space on points in the dataset that are considered to be exemplars. Unlike k-means clustering, the Leader algorithm 1) centers balls on actual data points rather than on centroids of clusters. 2) constrains every ball to the same radius rather than allowing them to have different diameters, and 3) involves only one pass through the data rather than iterating to convergence via multiple passes, and 4) produces thousands of balls rather than a few clusters. In rare instances, the resulting exemplars and members can be dependent on the order of the data, but not enough to affect the identification of outliers because of the large number of exemplars produced. We are characterizing a high-dimensional density by covering it with many small balls. Even relatively tight clusters produced by a clustering algorithm will be chopped into pieces by the Leader algorithm.

9. The number of exemplars resulting from δ applied even to large numbers of data points is small enough to allow the simple brute-force algorithm for finding nearest neighbors.

10. We use a modification of the Burridge and Taylor [12] algorithm due to Schwarz [67]. For all examples in this paper, α (the critical value) was set to .05.

11. Flagging all members of an outlying exemplar-neighborhood means that this algorithm can identify outlying sets of points as well as outlying singletons.

3.3 Distributed Computation

The `hdoutliers` algorithm is amenable to distributed computation because the dataset can be mapped into separate blocks and each block can be processed separately before merging results. The final reducing step is a straightforward merging of exemplars and their associated members across blocks. The H2O.ai open-source software [69] implements this algorithm. It reduces over 100 million rows of data to 5,000 rows of aggregated data in less than 20 seconds.

3.4 Validation

We validate `hdoutliers` by examining its performance with regard to 1) false positives and 2) false negatives. If the claims for the algorithm are valid, then we should expect it 1) to find outliers in random data not more than 100α percent of the time and 2) not to miss outliers when they are truly due to mixtures of distributions or anomalous instances.

3.4.1 False Positives

We perform these tests by running the algorithm many times on random data. If the algorithm performs as claimed, then it should not false alarm more frequently than we expect.

- Table 1 contains results of a simulation using random distributions. The entries are based on 1,000 runs of `hdoutliers` on Normally distributed variables with α (the critical value) set to .05. The entries show that `hdoutliers` is generally conservative, meaning its false alarm rate does not exceed .05.
- The results were similar for random Exponential and Uniform variables.

Table 1. Empirical level of `hdoutliers` test based on null model with Gaussian variables and critical value $\alpha = .05$.

	p=1	**p=5**	**p=10**	**p=100**
n=100	.011	.040	.018	.012
n=500	.015	.035	.027	.020
n=1000	.017	.045	.027	.024

3.4.2 False Negatives

We perform these tests by generating data that contain known outliers. We do this for several different density configurations.

- Figure 4 is based on the dataset in Figure 2. The `hdoutliers` identifies the outlier in the right dot plot but finds none in the left.
- Figure 5 shows that `hdoutliers` correctly identifies the inlier in the center of both one-dimensional and two-dimensional configurations.

- Table 2 shows that `hdoutliers` correctly identifies the outlier in a table defined by two categorical variables. The data consist of two columns of strings, one for {A,B,C,W} and one for {A,B,C,X}. There is only one row with the tuple $\langle W, X \rangle$. The `hdoutliers` also handles mixtures of categorical and continuous variables.

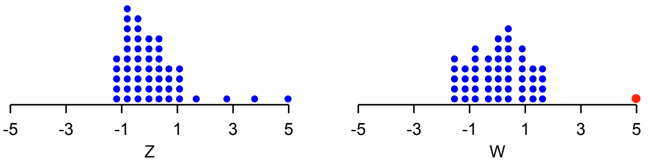

Fig. 4. The `hdoutliers` algorithm applied to data shown in Figure 2.

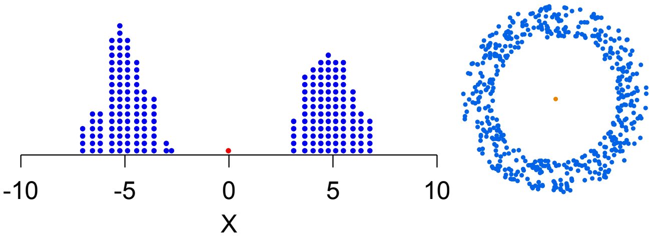

Fig. 5. Inlier datasets; `hdoutliers` correctly identifies the inliers.

Table 2. Crosstab with an outlier (red entry)

	A	B	C	X
A	100	0	0	0
B	0	100	0	0
C	0	0	100	0
W	0	0	0	1

3.4.3 Comparison with other Multidimensional Algorithms

The multidimensional outlier detection algorithm most widely cited by computer scientists is based on a criterion called the Local Outlier Factor (LOF) [11]. Unfortunately, it cannot be compared directly to `hdoutliers` because LOF is not grounded in probability theory. For example, one analysis [84] involves an application of LOF to the Anderson Iris dataset [2]. The analyst discovers five outliers. This dataset is well-known to contain three bivariate Normal clusters. The Mahalanobis distance measure admits no outliers for these clusters at $\alpha = .05$; neither does `hdoutliers`.

Nevertheless, `hdoutliers` should produce results similar to LOF when both methods identify comparable numbers of outliers. Figure 6 is based on the `dfki` dataset in [23]. The left panel shows what the authors consider to be outliers. The right panel shows the result of implementing `hdoutliers` inside a k-means clustering. On each iteration of the k-means algorithm, we apply `hdoutliers` to the points assigned to each cluster in order to determine if any points belonging to their nearest cluster should be treated as outliers. The outliers are then left out of the computation of the cluster centroids.

There are only slight differences between the two implementations. Notably, `hdoutliers` refuses to flag several cases that are not significantly far from their cluster neighbors. While LOF does not scale to massive datasets the way `hdoutliers` does, it is reassuring that both produce similar results on smaller, low-dimensional datasets.

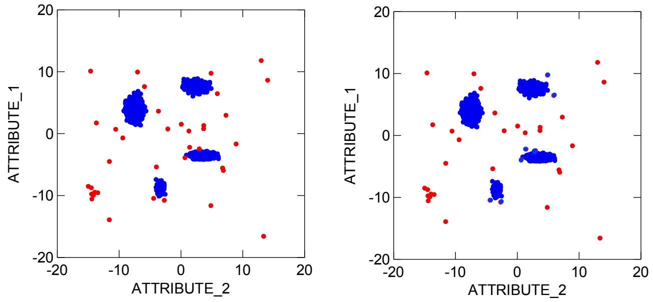

Fig. 6. Test dataset from [23]. The left plot shows what the authors consider to be outliers and the right plot is the result produced by `hdoutliers` inside a k-means clustering. The outliers are colored red in both plots.

Figure 7 is based on a dataset in [49]. The left panel shows outliers flagged by LOF in red. The `hdoutliers` result, shown in the right panel, contains no outliers. The sample size is too small to identify outliers at $\alpha = .05$.

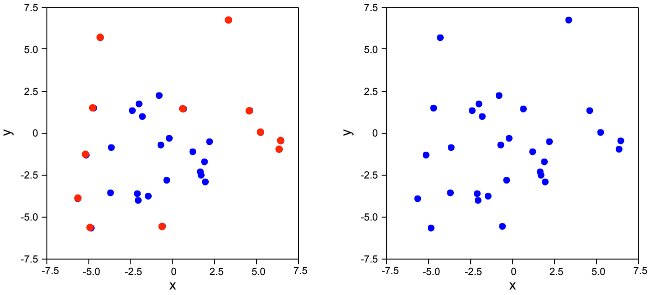

Fig. 7. Test dataset from [23]. The red points in the left panel were identified by LOF as outliers. The right panel shows the result of an analysis by the `hdoutliers` algorithm. With $\alpha = .05$, no points were identified as outliers. The probability model makes clear that no outliers in this small batch of points can be identified with any reasonable level of confidence.

4 Visualization

This section comprises an outlier tutorial. The main point in the following examples is that a statistical algorithm based on probability theory is necessary for reliably discovering outliers but visualizations are necessary for interpreting the results of these discoveries. These visualizations are not novel. What is new is the application of a probability-based algorithm to help manage the possibility of false discoveries and, equally important, the failure to discover outliers when they do exist. Several of the examples illustrate why visual analytics without such a statistical foundation can lead to serious mistakes. Finally, by featurizing certain problems, we introduce novel applications of outlier detection in examples that are not usually considered amenable to outlier detection.

4.1 Visualizing Unidimensional Outliers

For univariate outlier detection, histograms, probability plots [15], and dot plots [76] are most useful. Figure 8 shows a dot plot and Normal probability plot of residuals from a two-factor experiment. In probability plots, we look for major distortions from a straight line. A probability plot can be constructed from any parametric distribution for which a cumulative distribution function can be computed. They are widely used in experimental design and analysis of residuals.

Even though these univariate displays can be helpful in exploratory analysis to detect outliers, they do not yield the kind of risk estimate that hdoutliers or the parametric methods described in the Related Work sections provide. Without a risk estimate, the chance of false discoveries is uncontrolled. In practical terms, we might see terrorists in groups where none exist. Thus, as in Figure 8, it is helpful to highlight outliers using a statistical method like hdoutliers. This approach will also help with false negatives, where significant outliers may not be visually salient.

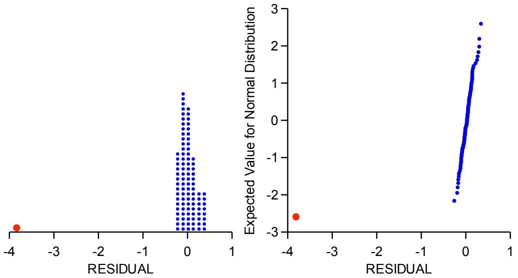

Fig. 8. Dot plot and Normal probability plot of residuals from a two-factor experiment. One lower outlier is evident.

4.2 Low-dimensional visualizations are not reliable ways to discover high-dimensional outliers

There have been many outlier identification proposals based on looking at axis-parallel views or low-dimensional projections (usually 2D) that are presumed to reveal high-dimensional outliers (e.g., [44, 34, 43, 47]). This approach is risky. Figure 9 shows why. The data are samples from a multivariate Normal distribution. The left panel plot illustrates the problem for two dimensions. The figure incorporates a 95 percent joint confidence ellipse based on the sample distribution of points. Two points are outside this ellipse. The red point on the left is at the extreme of both marginal histograms. But the one on the right is well inside both histograms. Examining the separate histograms would fail to identify that point.

The right panel plot shows the situation for three dimensions. The three marginal 2D plots are shown as projections onto the facets of the 3D cube. Each confidence ellipse is based on the pairwise plots. The red outlying point in the joint distribution is inside all three marginal ellipses. The 2D scatterplots fail to reveal the 3D outlier. The situation gets even worse in higher dimensions.

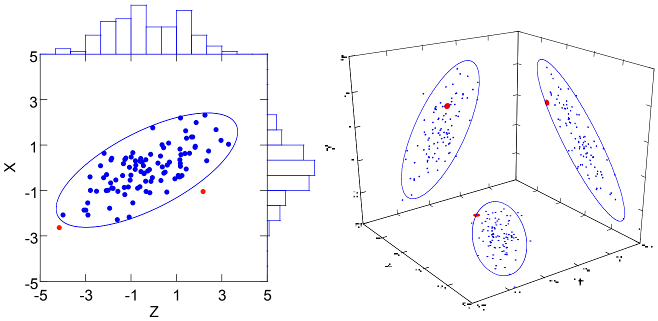

Fig. 9. 2D (left) and 3D (right) joint outliers. The figures show why lower-dimensional projections cannot be used to discern outliers.

Some have proposed projection methods for finding outliers (e.g., [10, 65, 50]). However, finding an outlier in a low-dimensional projection does not guarantee that it is an outlier in higher-dimensional ambient space. Table 3 shows a simple example. When distances are computed on the first two columns, the sixth point (row) has the largest nearest-neighbor distance. In six dimensions, however, the most outlying point is the fourth. The scatterplot matrix of these data (Figure 10) is of little help for peeking into six dimensions. Non-axis-parallel projections (e.g., principal components [6], tSNE [74]) suffer from similar problems. The sixth point is still the most outlying in a plot of the first two principle components, for example. The situation only gets worse with more variables

Table 3. Data for which outlier identification varies with dimensionality

A	B	C	D	E	F
1	0	1	1	1	3
1	1	0	2	0	4
2	0	2	3	3	4
1	-1	2	0	-1	0
0	-2	2	1	2	2
-3	-1	3	2	2	1

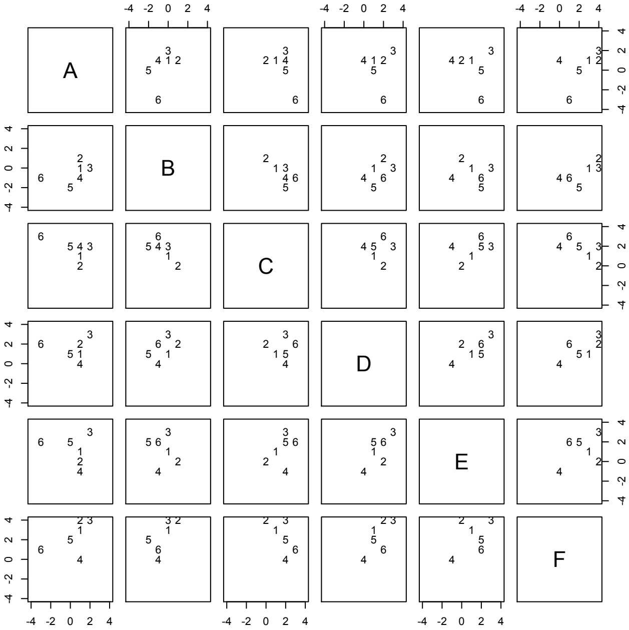

Fig. 10. Scatterplot matrix of data in Table 3.

Parallel coordinates, another 2D display, have been advocated for the purpose of outlier detection [55, 4, 41]. Figure 11 shows why this is risky. The figure shows parallel coordinates on four variables from the Adult dataset in the UCI repository [46]. The hdoutliers algorithm discovered two outliers out of 32,561 cases. The profiles appear to run through the middle of the densities even though they are multivariate outliers. The popular conception that outliers will appear at the edges of parallel coordinates displays is false (for reasons similar to those underlying Figure 9). As this example shows, real outliers can sometimes appear to have typical profiles in a parallel coordinates display. No amount of transparency or filtering to reduce overplotting can ameliorate this problem unless the outliers are known in advance.

4.3 Using statistical algorithms to highlight outliers in visualizations

While visualizations alone cannot be used to detect multidimensional outliers, they are invaluable for inspecting and understanding outliers detected by statistical methods. This section covers a variety of visualizations that lend themselves to outlier description. In each case, there is a productive partnership between the statistical outlier detector and the visualization.

4.3.1 Parallel Coordinates

Although parallel coordinates are generally useless for discovering outliers, they can be useful for inspecting outlier profiles detected by a statistical algorithm. The two outliers highlighted in Figure 11 are characterized by their profiles on each of the four variables. Even though the individual values on each variable may seem typical, it can be useful to consider the joint configurations represented by each profile. On the four variables taken together, these two cases are indeed outliers.

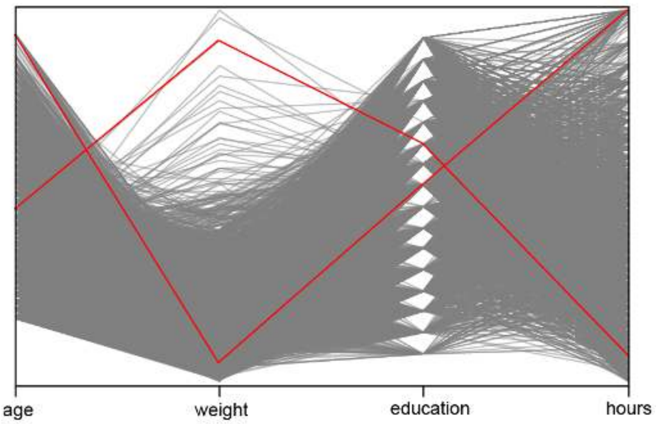

Fig. 11. Parallel coordinates plot of five variables from the Adult dataset in the UCI data repository. The red profiles are multivariate outliers. Even though the profiles are plotted with 50% opacity, the 32,561 profiles mask much of the detail. Using color [54] or stacking [17] would not help to unmask the outliers.

4.3.2 Regression Residuals

The conventional statistical wisdom for dealing with outliers in a regression context is to examine residuals using a variety of diagnostic graphics and statistics [5, 7, 16]. Following this advice is critical before promoting any particular regression model on a dataset. It is a necessary but not sufficient strategy, however. The reason is that some outliers have a high influence on the regression and can pull the estimates so close to them that they are masked.

Figure 12, derived from an example in [64], shows how this can happen in even the simplest bivariate regression. The data are measurements of light intensity and temperature of a sample of stars. In the left panel, the ordinary least squares (OLS) regression line is pulled down by the four outliers in the lower right corner, leaving a bad fit to the bulk of the points. We would detect most, but not all, of the outliers in a residual plot. The right pane, based on a least median of squares regression (LMS) [62], shows six red points as regression outliers. They are, in fact, dwarf stars.

There are numerous robust regression models, but LMS has the lowest breakdown point against outliers [21]. Therefore, the most prudent approach to regression modeling is to compute the fit both ways and see if the regression coefficients and residual plots differ substantially. If they do, then LMS should be the choice. Otherwise, the simpler OLS model is preferable.

4.3.3 Time Series Outliers

Detecting time series outliers requires some pre-processing. In particular, we need to fit a time series model and then examine residuals. Fitting parametric models like ARIMA [9] can be useful for this purpose, but appropriate model identification can be complicated. A simpler approach is to fit a nonparametric smoother. The example in Figure 13 was fit by a kernel smoother with a biweight function on the running mean. The data are measurements of snowfall at a Greenland weather station, used in [77]. The outliers (red dots) are presumably due to malfunctions in the recording equipment.

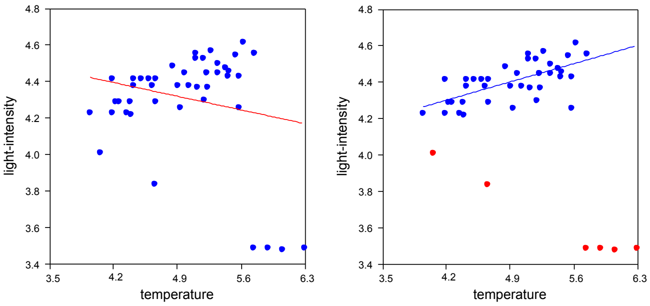

Fig. 12. Ordinary Least Squares regression (left panel) and Least Median of Squares (LMS) regression (right panel) on attributes of stars. Data are from [64]

Computing outlying series for multiple time series is straightforward with the `hdoutliers` algorithm. We simply treat each series as a row in the data matrix. For n series on p time points, we have a p-dimensional outlier problem. Figure 14 shows series for 20 years of the Bureau of Labor Statistics Unemployment data. The red series clearly indicate the consequences of the Great Recession. This example illustrates why a probability-based outlier method is so important. We could rank the series by their average levels of unemployment or use one of the other ad-hoc multidimensional outlier detectors, but we would have no way of knowing how many at the top are significant outliers. The Local Outlier Factor (LOF) algorithm [11], for example, does not provide a probability-based cutoff for determining the number of outliers.

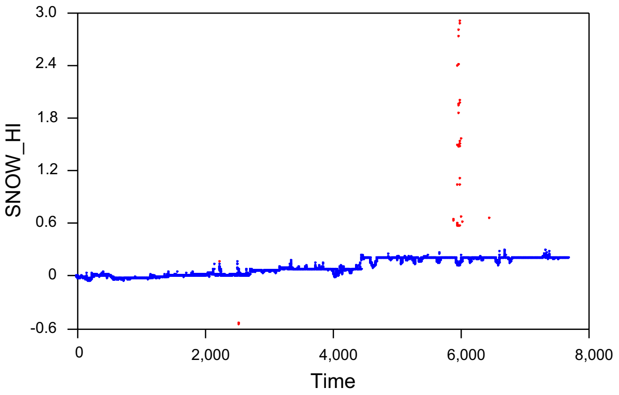

Fig. 13. Outlying measurements of snow cover at a Greenland weather station.

4.3.4 Ipsative Outliers

An *ipsative* outlier is a case that is an outlier with respect to itself. That is, we standardize values within each case (row) and then look for outliers in each standardized profile. Any profile with an outlier identified by `hdoutliers` is considered noteworthy; in other words, we can characterize a person simply by referring to his outliers. It is easiest to understand this concept by examining a graphic. Figure 15 shows an outlying profile for a baseball player who is hit by pitches more frequently than we would expect from looking at his other characteristics. This player may not be hit by pitches significantly more than other players, however. We are instead interested in a player with a highly unusual profile that can be described simply by his outlier(s). In every other respect, the player is not necessarily noteworthy. This method should not be used, of course, unless there are enough features to merit computing the statistical outlier model on a case.

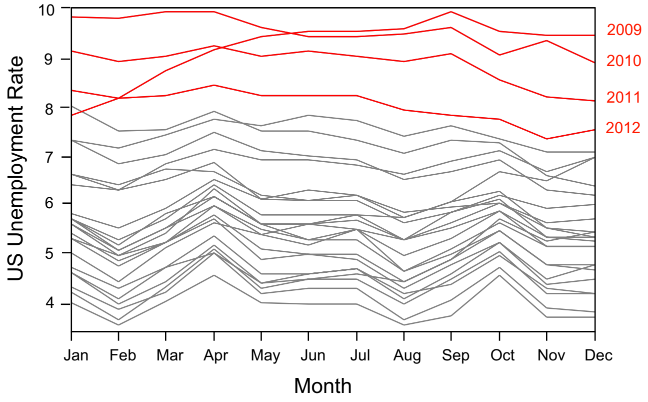

Fig. 14. US Unemployment series outliers. The shock and ensuing recovery from the Great Recession is clearly indicated in the outliers.

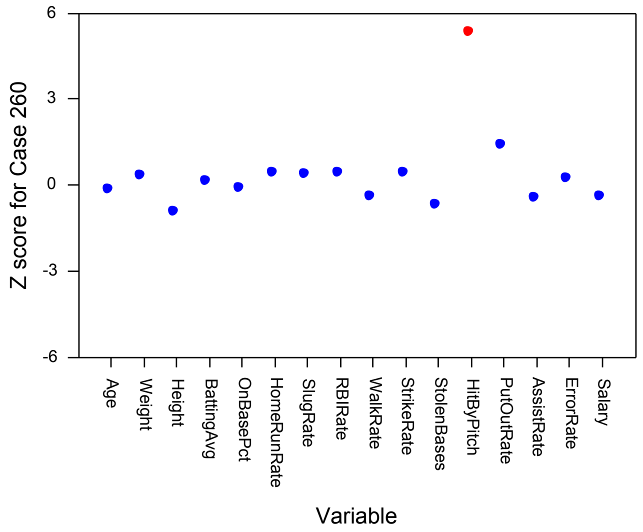

Fig. 15. One baseball player's profile showing an outlier (hit by pitch) that deviates significantly from his other features.

4.3.5 Text Outliers

An important application for multivariate outlier detection involves document analysis. Given a collection of documents (Twitter messages, Wikipedia pages, emails, news pages, etc.), one might want to discover any document that is an outlier with respect to the others. The simplest approach to this problem is to use a bag-of-words model. We collect all the words in the documents, stem them to resolve variants, remove stopwords and punctuation, and then apply the tf-idf measure [66] on the words within each document. The resulting vectors for each document are then submitted to `hdoutliers`.

Figure 16 shows the results for an analysis of 21 novels from the Guttenberg Web site [29]. This problem requires the use of random projections. Before projection, there are 21,021 columns (tf-idf measures) in the dataset. After projection there are 653. Not surprisingly, *Ulysses* stands out as an outlier. Distinctively, it contains numerous neologisms.

Tristram Shandy was identified by `hdoutliers` as the second largest, but not significant, outlier. It too contains numerous neologisms. These two novels lie outside most of the points in Figure 16. Not all multivariate outliers will fall on the periphery of 2D projections, however, as we showed in Section 4.2.

Without the random projection stage of `hdoutliers`, these outliers could not be recognized. The number of dimensions is far too large for even distributed versions of LOF and other algorithms.

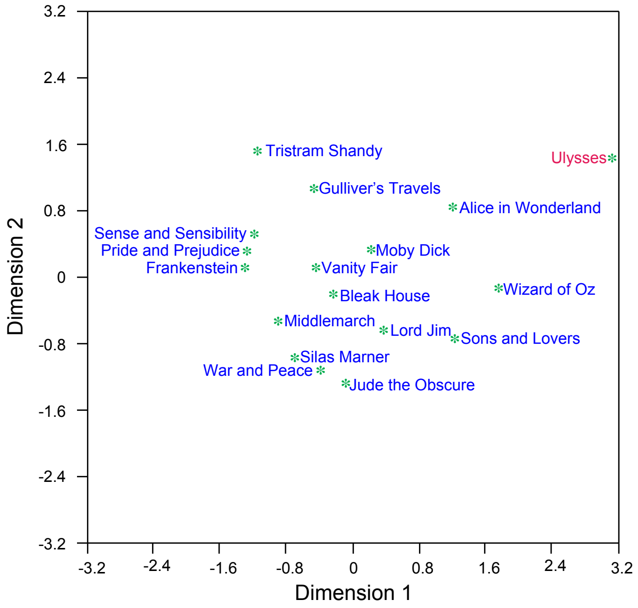

Fig. 16. Document outliers. Nonmetric multidimensional scaling on matrix of Spearman correlations computed on tfidf scores. The stress for this solution is .163 and one document (*Ulysses*) is flagged as an outlier by `hdoutliers`.

4.3.6 Graph Outliers

There are several possibilities related to finding outliers in graphs. One popular application is the discovery of outliers among nodes of a network graph. The best way to exploit `hdoutliers` in this context is to featurize the nodes. Common candidates are Prominence, Transitivity (Watts-Strogatz Clustering Coefficient), Closeness Centrality, Betweenness Centrality, Node Degree, Average Degree of Neighbors, and Page Rank [53]. Figure 17 shows an example for the Les Miserables dataset [45]. The nodes were featurized for Betweenness Centrality in order to discover any extraordinarily influential characters. Not surprisingly, Valjean is connected to significantly more characters than anyone else in the book. Notice that Valjean's node is not at the periphery of the graph layout. His outlyingness is a consequence of his centrality, which not surprisingly places him near the centroid of the layout. Other graph theoretic measures could reveal outliers in other nodes or edges of the graph.

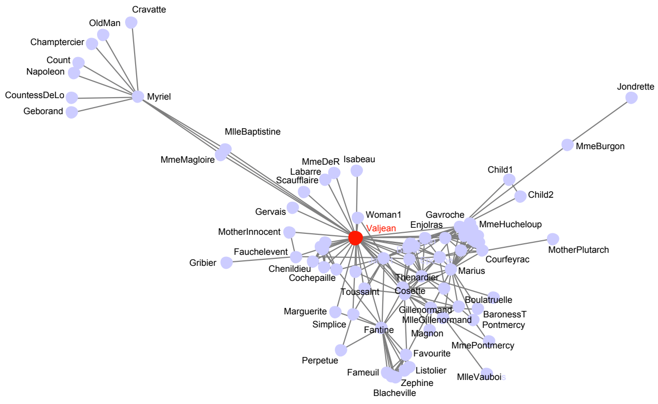

Fig. 17. Les Miserables characters network graph. Valjean is identified as outlying on Betweenness Centrality.

An alternative application involves discovering outlying graphs in a collection of graphs. For this problem, we need to find a way to characterize a whole graph and to derive a distance measure that can be fed to hdoutliers. This application depends on assuming the collection of graphs is derived from a common population model and that any outliers involve a contamination from some alternative model. We need a measure of the distance between two graphs to do this. Unfortunately, graph matching and related graph edit distance calculations have impractical complexities. Approximate distances are easier to calculate, however [73]. The approach we take is as follows:

First, we compute the adjacency matrix for each graph. We then convert the adjacencies above the diagonal to a single binary string. When doing that, however, we have to reorder the adjacency matrix to a canonical form; otherwise, arbitrary input orderings could affect distance calculations on the string. A simple way to do this is to compute the eigendecomposition of the Laplacian matrix and permute the adjacencies according to the ordering of the values of the eigenvector corresponding to the smallest nonzero eigenvalue. After permuting and encoding the adjacency matrices into strings, we compute the Levenshtein distances [51] between pairs of strings. Finally, we assemble the nearest-neighbor distances from the resulting distance matrix and subject them to the hdoutliers algorithm.

Figure 18 shows an example of this approach using the Karate Club graph [82]. We generated 15 random minimum spanning tree graphs having the same number of nodes as the Karate Club graph. Then we applied the above procedure to identify outliers. The Karate Club graph was strongly flagged as an outlier by the algorithm.

Large collections of graphs (as in social networks) present formidable computational problems for this type of application. These problems usually require distributed computation on many computing nodes. Fortunately, the distributed version of the hdoutliers algorithm is well-suited for large-scale problems.

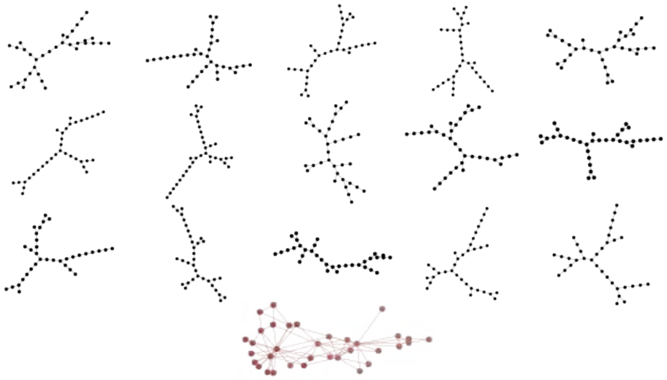

Fig. 18. Karate Club graph (red) is an outlier with respect to comparably scaled random minimum spanning tree graphs.

4.3.7 Scagnostics Outliers

Scagnostics [80] can be used to identify outlying scatterplots in a large collection of scatterplots. Because the calculations are relatively efficient, these measures can be computed on many thousands of plots in practical time. This outlier application is multivariate, because there are nine scagnostics for each scatterplot, so a multivariate detection algorithm like hdoutliers is required.

Figure 19 shows two outlying scatterplots identified by hdoutliers when applied to a dataset of baseball player characteristics featured in [81]. While the left plot in the figure is clearly unusual, the surprising result is to see an evidently bivariate Normal scatterplot of Weight against Height in the right plot. Although the dataset includes many physical and performance features of real baseball players, the type of Normal bivariate distribution found in many introductory statistics books is an outlier among the 120 scatterplots considered in this example.

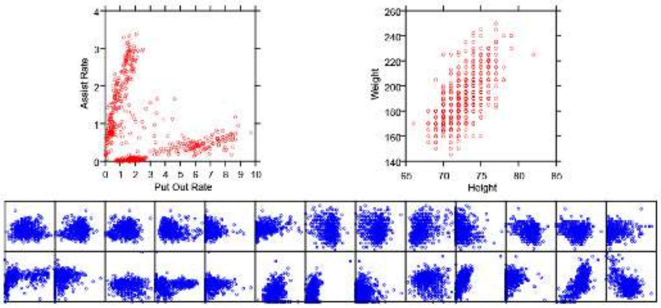

Fig. 19. Scatterplot outliers based on Scagnostics computed on 120 scatterplots of baseball player features. A sample of the 120 scatterplots is shown in blue.

4.3.8 Geographic Outliers

We can compute spatial outliers using the hdoutliers algorithm. More frequently, however, maps are a convenient way to display the results of outlier detection on non-spatial exogenous variables. Figure 20 shows an example of outlier detection on marriage and divorce rates by US state. The quantile plot in the left subframe reveals how extreme is this outlier.

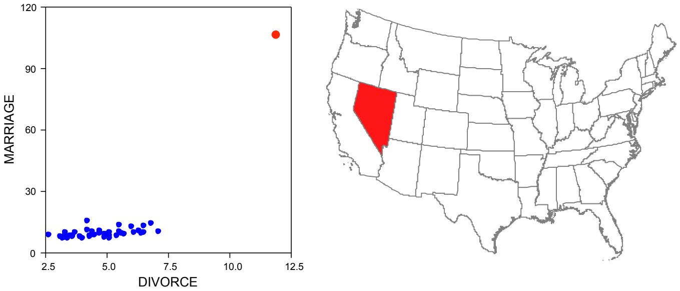

Fig. 20. Marriage and Divorce rates in the US. There is one state that is an outlier.

5 Conclusion

There is a huge assortment of papers on outlier detection in the machine learning community; only a fraction is cited here. While many of these approaches are ingenious, few rest on a foundation that takes risk into account. If we label something as an outlier, we had better be able to quantify or control our risk.

Outliers are anomalies. An anomaly is not a thing; literally, *anomaly* means lack of a law. It is a judgment based on evidence. Sometimes evidence is a collection of facts. Sometimes it is a collection of indications that cause us to modify our prior belief that what we are observing is not unusual. The statistical detection of outliers is concerned with the latter case. Lacking certainty of the process that generated what we think might be an outlier, we must derive a judgment that an observation is inconsistent with our belief in that process.

In order to be consistent in our behavior, we need to assign a probability to the strength of our belief that we are looking at an outlier. Methods that do not do this, that simply rank discrepancies or flag observations above an arbitrary threshold (like most of the algorithms in the Related Work section), can lead to inconsistent results.

The hdoutliers algorithm reduces the risk of making a false outlier discovery for a broad class of prior beliefs. Even for unusual applications such as the graph outlier problem, this algorithm provides a foundation for framing the judgment concerning an outlier. And importantly for the applications in this paper, hdoutliers is designed specifically to guide, protect, and deepen our visual analysis of data.

Acknowledgments

This work was supported by NSF/DHS Supplement to grant DMS-FODAVA-0808860. All figures and analyses were computed using SYSTAT [78], Skytree Adviser [79], and R[58]. The R package `hdoutliers`, developed by Chris Fraley, is available in CRAN. The distributed version of the algorithm, developed by Leland Wilkinson and Arno Candel, is available at http://www.H2O.ai

References

[1] C. Aggarwal. *Outlier Analysis*. Springer Verlag, 2013.

[2] E. Anderson. The irises of the Gaspe Peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.

[3] F. Anscombe. Rejection of outliers. *Technometrics*, 2:123–147, 1960.

[4] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *IEEE Symposium on Information Visualization*, pages 81–88, 2004.

[5] A. Atkinson. *Plots, Transformations and Regression: An Introduction to Graphical Methods of Diagnostic Regression Analysis*. Oxford University Press, 1985.

[6] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley & Sons, 1994.

[7] D. A. Belsley, E. Kuh, and R. E. Welsch. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. John Wiley & Sons, 1980.

[8] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *ICDT '99: Proceedings of the 7th International Conference on Database Theory*, pages 217–235, 1999.

[9] G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control (rev. ed.)*. Holden-Day, Oakland, CA, 1976.

[10] M. Breaban and H. Luchian. Outlier detection with nonlinear projection pursuit. *International Journal of Computers Communications & Control*, 8(1):30–36, 2013.

[11] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 93–104, New York, NY, USA, 2000. ACM.

[12] P. Burridge and A. Taylor. Additive outlier detection via extreme-value theory. *Journal of Time Series Analysis*, 27:685–701, 2006.

[13] C. Caroni and P. Prescott. On Rohlf's method for the detection of outliers in multivariate data. *Journal of Multivariate Analysis*, 52:295–307, 1995.

[14] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surveys*, 41:15:1–15:58, July 2009.

[15] W. S. Cleveland. *The Elements of Graphing Data*. Hobart Press, Summit, NJ, 1985.

[16] R. D. Cook and S. Weisberg. *Residuals and Influence in Regression*. Chapman & Hall, London, 1982.

[17] T. Dang, L. Wilkinson, and A. Anand. Stacking graphic elements to avoid over-plotting. In *INFOVIS '10: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'10)*, Washington, DC, USA, 2010. IEEE Computer Society.

[18] T. N. Dang and L. Wilkinson. Transforming scagnostics to reveal hidden features. *IEEE Transactions on Visualization and Computer Graphics*, 20:1624–1632, November 2014.

[19] R. Dawson. How significant is a boxplot outlier? *Journal of Statistics Education*, 19, 2011.

[20] W. Dixon. Ratios involving extreme values. *Annals of Mathematical Statistics*, 22:68–78, 1951.

[21] D. Donoho and P. Huber. The notion of breakdown point. In P. Bickel, K. Doksum, and J. Hodges, editors, *A Festschrift for Erich L. Lehman*, pages 157–184. Wadsworth, Belmont, CA, 1983.

[22] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. M. Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[23] G. R. C. for Artificial Intelligence. dataset: dfki-artificial-3000-unsupervised-ad.csv. `http://madm.dfki.de/downloads`. Accessed: 2016-02-08.

[24] C. Fraley and A. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

[25] M. Frigge, D. Hoaglin, and B. Iglewicz. Some implementations of the boxplot. *The American Statistician*, 43:50–54, 1989.

[26] R. Gnanadesikan. *Methods for statistical data analysis of multivariate observations*. John Wiley & Sons, New York, 1977.

[27] M. Greenacre. *Theory and Applications of Correspondence Analysis*. Academic Press, 1984.

[28] F. Grubbs. Sample criteria for testing outlying observations. *The Annals of Mathematical Statistics*, 21:27–58, 1950.

[29] M. Hart. Project Gutenberg. `https://www.gutenberg.org`.

[30] J. Hartigan. *Clustering Algorithms*. John Wiley & Sons, New York, 1975.

[31] D. Hawkins. *Identification of Outliers*. Chapman & Hall/CRC, 1980.

[32] Henry. How is the distance of two random points in a unit hypercube distributed? Mathematics Stack Exchange, 2016.

[33] A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases*, pages 506–515, 2000.

[34] A. Hinneburg, D. Keim, and M. Wawryniuk. HD-Eye: Visual mining of high-dimensional data. *IEEE Computer Graphics and Applications*, 19(5):22–31, Sept. 1999.

[35] V. Hodge. *Outlier and Anomaly Detection: A Survey of Outlier and Anomaly Detection Methods*. LAP LAMBERT Academic Publishing, 2011.

[36] H. Hofmann, K. Kafadar, and H. Wickham. Letter-value plots: Boxplots for large data. Technical report, had.co.nz, 2011.

[37] M. Hubert and E. Vandervieren. An adjusted boxplot for skewed distributions. *Computational Statistics & Data Analysis*, 52:5186–5201, 2008.

[38] I. Iglewicz and D. Hoaglin. How to detect and handle outliers. In E. Mykytka, editor, *The ASQC Basic References in Quality Control: Statistical Techniques*. ASQC, 1993.

[39] M. Jiang, S. Tseng, and C. Su. Two-phase clustering process for outliers detection. *Pattern Recognition Letters*, 22:691–700, 2001.

[40] J. Jobe and M. Pokojovy. A cluster-based outlier detection scheme for multivariate data. *Journal of the American Statistical Association*, 110:1543–1551, 2015.

[41] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *IEEE Symposium on Information Visualization*, pages 125–132, 2005.

[42] W. B. Johnson and J. Lindenstrauss. Lipschitz mapping into Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[43] E. Kandogan. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 107–116, New York, NY, USA, 2001. ACM.

[44] E. Kandogan. Just-in-time annotation of clusters, outliers, and trends in point-based data visualizations. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, VAST '12, pages 73–82, Washington, DC, USA, 2012. IEEE Computer Society.

[45] D. Knuth. *The Stanford GraphBase: A Platform for combinatorial computing*. Addison-Wesley, Reading, MA, 1993.

[46] R. Kohavi and B. Becker. Adult data set. `http://archive.ics.uci.edu/ml/datasets/Adult`, 1996.

[47] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *13th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD 2009)*, Tangkok, Thailand, 2009.

[48] H.-P. Kriegel, P. Kroger, E. Schubert, and A. Zimek. Interpreting and unifying outlier scores. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 13–24, 2011.

[49] D. Kuznar. `https://github.com/damjankuznar/pylof/blob/master/README.md`. Python implementation of Local Outlier Factor algorithm by Markus M. Breunig.

[50] D. J. Lehmann and H. Theisel. General projective maps for multidimensional data projection. *Computer Graphics Forum*, 35(2), 2016.

[51] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710, 1966.

[52] J. Lin, D. Ye, C. Chen, and M. Gao. Minimum spanning tree based spatial outlier mining and its applications. In *Proceedings of the 3rd International Conference on Rough Sets and Knowledge Technology*, pages 508–515, Berlin, Heidelberg, 2008. Springer-Verlag.

[53] M. E. J. Newman, A.-L. Barabasi, and D. J. Watts. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[54] H. Nguyen and P. Rosen. DSPCP: A data scalable approach for identi-

fying relationships in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[55] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, Sept. 2006.

[56] R. Pamula, J. Deka, and S. Nandi. An outlier detection method based on clustering. In *International Conference on Emerging Applications of Information Technology, EAIT*, pages 253–256, 2011.

[57] S. Peter and S. Victor. Hybrid – approach for outlier detection using minimum spanning tree. *International Journal of Computational & Applied Mathematics*, 5:621, 2010.

[58] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[59] B. Rieck and H. Leitte. Exploring and Comparing Clusterings of Multivariate Data Sets Using Persistent Homology. *Computer Graphics Forum*, 2016.

[60] F. Rohlf. Generalization of the gap test for the detection of multivariate outliers. *Biometrics*, 31:93–101, 1975.

[61] G. E. Rosario, E. A. Rundensteiner, D. C. Brown, M. O. Ward, and S. Huang. Mapping nominal values to numbers for effective visualization. *Information Visualization*, 3(2):80–95, June 2004.

[62] P. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880, 1984.

[63] P. Rousseeuw and B. V. Zomeren. Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85:633–651, 1990.

[64] P. J. Rousseeuw and A. Leroy. *Robust Regression & Outlier Detection*. John Wiley & Sons, 1987.

[65] A. Ruiz-Gazen, S. L. Marie-Sainte, and A. Berro. Detecting multivariate outliers using projection pursuit with particle swarm optimization. In *COMPSTAT2010*, pages 89–98, 2010.

[66] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24:513–523, 1988.

[67] K. Schwarz. *Wind Dispersion of Carbon Dioxide Leaking from Underground Sequestration, and Outlier Detection in Eddy Covariance Data using Extreme Value Theory*. PhD thesis, Physics, University of California, Berkeley, 2008.

[68] D. Steinley. Standardizing variables in k-means clustering. In D. Banks, L. House, F. R. McMorris, P. Arabie, and W. Gaul, editors, *Classification, clustering, and data mining applications*, pages 53–60. Springer, New York, 2004.

[69] C. Stick, S. Ambati, L. Wilkinson, and A. Candel. `http://h2O.ai`. Mountain View, CA.

[70] S. Stigler. *The History of Statistics: The Measurement of Uncertainty Before 1900*. Harvard University Press, 1986.

[71] H. Thode. *Testing For Normality*. Taylor & Francis, 2002.

[72] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley Publishing Company, Reading, MA, 1977.

[73] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:695–703, 1988.

[74] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[75] H. Wainer and S. Schacht. Gapping. *Psychometrika*, 43:203–212, 1978.

[76] L. Wilkinson. Dot plots. *The American Statistician*, 53:276–281, 1999.

[77] L. Wilkinson. *The Grammar of Graphics*. Springer-Verlag, New York, 1999.

[78] L. Wilkinson. *SYSTAT, Version 10*. SPSS Inc., Chicago, IL, 2004.

[79] L. Wilkinson. A second opinion statistical analyzer. In *International Symposium on Business and Industrial Statistics*, 2010.

[80] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Proceedings of the IEEE Information Visualization 2005*, pages 157–164. IEEE Computer Society Press, 2005.

[81] L. Wilkinson, A. Anand, and R. Grossman. High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1363–1372, 2006.

[82] W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[83] C. T. Zahn. Graph-theoretical methods for detecting and describing Gestalt clusters. *IEEE Transactions on Computers*, C-20:68–86, 1971.

[84] Y. Zhao. Outlier detection. `http://www.rdatamining.com/examples/outlier-detection`.