Training Machines to See What You See: Applications of Statistical Learning Methods for Pattern Detection in Scatterplots

Cari Comnick, Logan Crowl, Sophie Gunn, Aidan Mullan

Abstract

As statisticians, we are fundamentally interested in the relationships between variables. Perhaps the most common tool for examining these relationships is the scatterplot. Because the human brain has evolved to detect complex patterns, a viewer can use scatterplots to characterize a wide variety of relationships between variables. But as the number of variables in a dataset grows, it becomes infeasible for a single viewer to consider all possible scatterplots. Furthermore, this tendency toward pattern detection can lead to the identification of relationships that do not exist in reality. In this paper, we will discuss how a computer can be trained using statistical learning methods to detect patterns in scatterplots more efficiently and effectively than humans. Our research leads to the development of an automated procedure to detect relationships between variables in large data sets and opens the door to more powerful data analysis.

1 Introduction

As one of the most powerful tools for visual inference, scatterplots remain an essential statistical tool toward determining relationships between quantitative variables. Through inspection, viewers can discern patterns in scatterplots that correspond to fundamental connections within data. These types of inferences are made possible by the human viewer's ability to essentialize the patterns behind the ever-changing placement of points. Furthermore, this remarkable capability makes visual assessments of scatterplots an integral part of any comprehensive statistical analysis. As a result, when the number of variables in a data set is large, exploratory analysis for variable relationships becomes infeasible due to the sheer quantity of possible pairwise associations. Additionally, human inclination for pattern recognition often leads to misattributing meaning to random variability. Thus, a model able to identify behavior in scatterplots could improve the efficiency of data exploration as well as open a door toward more effective automated statistical procedures. In the pursuit of these types of applications, we first must demonstrate that the patterns present in scatterplots can be consistently quantified and used to differentiate pattern from random variability. Our research builds on prior attempts to measure structure in scatterplots and examines whether these metrics provide enough information for statistical learning procedures to identify important behavior. Finally, we compare these statistical learning models to human perception and explore their potential to change how we analyze scatterplots across a variety of settings.

2 Scatterplot Diagnostics

To begin this process of analyzing scatterplots, we needed a way of quantifying observations about patterns in points. In the 1980s, John and Paul Tukey first introduced the idea of scatterplot diagnostics, aptly named *scagnostics*. This set of measures is designed to characterize the structure of two-dimensional scatterplots (Tukey, J., 1974; Tukey, J. & Tukey, P., 1985). Scagnostics were formally defined by Wilkinson et al. in their paper, "Graph-Theoretic Scagnostics" (Wilkinson, et al., 2005). We will begin by explaining how scagnostics are defined. In order to do this, we first must establish the graph theory necessary to understand exactly how these measures are constructed and calculated.

2.1 Computing Scagnostics

Geometric graphs

Geometric graphs are an essential part of how scagnostic measures quantify the behavior present in scatterplots. By looking at geometric graphs, we are able to describe the structure present in a scatterplot and calculate each scagnostic measure based on particular features. We will begin by defining the features and graph theoretic objects necessary to understand scagnostics.

A graph is a set of vertices, V, which are related by edges $e(v, w) \in E$, with $v, w \in V$. For scagnostics, only geometric graphs are used. Geometric graphs are those that can be represented in a metric space, S, as points and lines. Additionally, scagnostics only use geometric graphs that are undirected (all pairs (v, w) are unordered), simple ($v \neq w$), planar (can be represented in 2-dimensional space with no crossed edges), straight (all edges are straight lines), and finite (V and E are finite).

A reference should appear somewhere in this section for further readings.

Feature measures

Certain feature measures of geometric graphs are used to calculate scagnostics:

1. *Length*(e) is the Euclidean distance between the vertices of an edge e.
2. *Length*(G) is the total length of all edges of a graph G.
3. A *path* is a list of vertices such that all successive pairs are an edge.
4. A path is *closed* if its first and last vertex are the same.
5. A *polygon* is the boundary of a closed path.
6. *Area*(P) is the area of polygon P.
7. *Perimeter*(P) is the length of the boundary of polygon P.

use Length? functions should it be italicized

Convex Hull

In order to quantify the most general shape of our scatterplot, we will use the convex hull of a graph. To understand the convex hull, we must define a convex set. A convex set of X contains all the straight line segments connecting any pair of points in X. A convex hull of a given set of points X is the intersection of all convex sets of X, thus making it the smallest convex set containing X. In Euclidean space, the convex hull can be thought of as the polygon created by wrapping the set of points X in a rubber band.

isn't it the convex hull of the set of points?

define 1st time you use X

what is X?

Nonconvex Hull/Alpha Hull

There are many ways to represent the non-convex shape of a set of points. For scagnostic measures, the *alpha hull* is used due to its computational efficiency and status as an erosion method. The *alpha hull* is a graph where points are connected by an edge when they can be touched by an open disk $D(\alpha)$ containing no points, ~~The value~~ where α represents the radius of the disk and is chosen to be the value of the ω parameter. The ω parameter is the cutoff value for identifying outlying edges in the minimum spanning tree (defined below). The *alpha hull* can be constructed by rolling a circle of radius α around the set of points and placing edges between points that touch the circle. It can also be generalized to n dimensional data sets by replacing the 2-dimensional open disk by an n dimensional open ball.

what does this mean?

Now start sentence w/ how to choose α based on ω

Minimum Spanning Trees

A *tree* is any simple graph that is undirected, connected, and acyclic. For a given set of points X, a *spanning tree* is any tree whose vertices are exactly the points in X. The *minimum spanning tree* (MST) of X is defined as the spanning tree whose total edge weight is the minimum ~~for~~ of all possible spanning trees of X. In the context of geometric graphs, edge weight is taken to be the Euclidean distance between the two vertices connected by a given edge, so our MST will be the spanning tree with the smallest total length.

2.2 Scagnostics

With the understanding of the necessary graph feature measures, we now define each scagnostic measure and how it is computed. These scagnostic measures were defined by Wilkinson, et al. (2006).

Clumpy

The clumpy scagnostic indicates the clustering of points. This measure uses the Hartigan and Mohanty RUNT statistic. With the single-linkage hierarchical clustering tree called a dendro-

Citation?

I don't see an in-text reference to figure 1. If you include a figure you need to tell the reader what it is and what they should see in the text.

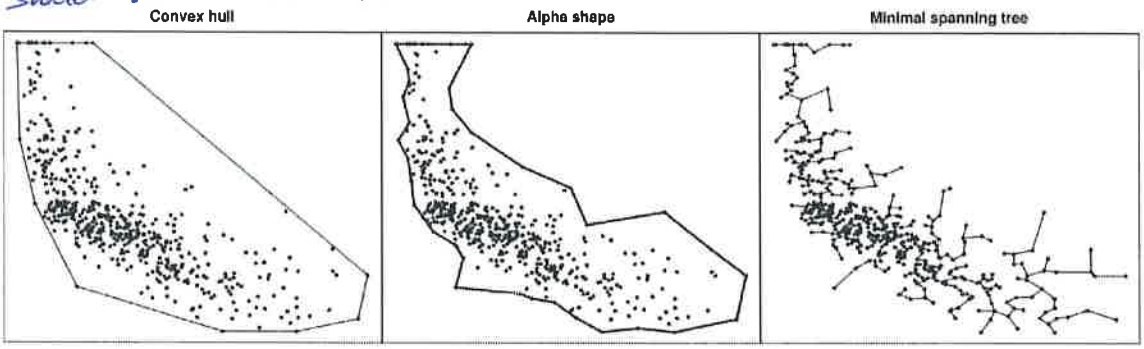


Figure 1: Geometric Graphs used in Computing Scagnostics

First define the runt size, then use it.

gram, this statistic uses the runt size of each node (r_j), which is the smaller of the number of leaves of each of the two subtrees joined at that node. Each r_j is associated with an e_j. The runt graph R_j corresponding to each edge c_j is the smaller of the two subsets of edges that are still connected to each of the two vertices in e_j after deleting edges in the MST with lengths less than $length(e_j)$. This measure emphasizes clusters with small intracluster difference relative to their connecting edge. In this formula, j indexes edges in the MST and k indexes edges in each runt set derived from an edge indexed by j.

Give possible values for j and k. eg j = 1, ... and k = ...

$$c_{clumpy} = \max_j \left[1 - \frac{\max_k [length(e_k)]}{length(e_j)} \right]$$

Sparse

Sparseness measures whether points are confined to a small number of locations on the plane. This happens with small ~~numbers of points~~ sets, or discrete, and when plotting categorical variables. It is measured using the 90th percentile of the edge lengths of the MST. If this exceeds 1, it is capped at 1. If points tend to be separated, then the 90th percentile of the minimum spanning tree will be large, but as points draw closer together, q_{90} will decrease.

Sparseness

$$c_{sparse} = q_{90}$$

What about: $c_{sparse} = \begin{cases} q_{90} & \text{if } q_{90} < 1 \\ 1 & \text{O.W.} \end{cases}$

Striated

This defn is a bit hard to follow

The striated measure captures how smooth paths in the minimum spanning tree are. An example of this smoothness would be found in a plot of categorical versus continuous variables (producing stripes), ~~but it can also be found in time series, curves, and smooth algebraic functions~~. To generalize, the measure is based on the number of adjacent edges (the set of adjacent edges is $V^{(2)}$) whose cosine is less than -0.75.

$$c_{striated} = \frac{1}{|V|} \sum_{v \in V^{(2)}} I(cos\theta_{e(v,a)e(v,b)} < -.75)$$

Into the defn., then clarify $V^{(2)}$ and cos ...

Convex

The convex scagnostic measures how well the convex hull captures the shape of the points in the scatterplot. It is calculated using ~~the~~ as ratio of the area of the alpha hull to the area of the convex hull:

$$c_{convex} = \frac{area(A)}{area(H)}$$

Skinny

The skinny scagnostic is a normalized ratio of the area to the perimeter of the alpha hull. Normalization ensures a circle earns a value of 0, a square a value of 0.012 and a skinny polygon near 1.

$$c_{skinny} = 1 - \frac{\sqrt{4\pi area(A)}}{perimeter(A)}$$

Stringy

A *stringy* plot is a skinny plot with no branches. This scagnostic uses the vertices of the minimum spanning tree. It compares the number of degree-2 vertices to the total number of vertices minus degree 1 vertices. Thus, if the minimum spanning tree consists of mostly one straight path or diameter with few branches, then most vertices will be of degree 2, creating a high stringy value. If the plot does not have a clear path through the minimum spanning tree, then it will have many branches, and most vertices will have degrees greater than two, so the stringy value will be low. If V are the vertices of the MST, then our *stringy* scagnostic is defined as follows:

$$c_{stringy} = \frac{|V^{(2)}|}{|V| - |V^{(1)}|}$$

Monotonic

In order to measure the monotonicity of a set of points, the squared Spearman correlation coefficient is used. The Spearman correlation coefficient is the Pearson correlation on the ranks of the x and y coordinates. As a result, the Spearman correlation will be high when points have similar rank for the x and y coordinates and close to -1 when the coordinates have very dissimilar rank. In order to capture both positive and negative monotonic behavior, we take the square of this correlation as our measure of monotonicity.

$$c_{monotonic} = r_{spearman}{}^2$$

Outlying

The outlying scagnostic is a measure of the proportion of overall edge length that is due to the presence of long edges connected to leafs, or points of single degree, in the minimum spanning tree. Large values for this scagnostic measure indicate that the relatively few edges with lengths that are significantly longer than most other edges contribute a substantial portion to the overall length of the minimum spanning tree. Edges are defined to be outliers if the edge weight is greater than ω, where ω is calculated by

$$\omega = q_{75} + 1.5(q_{75} - q_{25})$$

Here, q_{75} and q_{25} are the 75th and 25th percentile of edge lengths in the minimum spanning tree. Then, the outlying scagnostic is given as

$$c_{outlying} = \frac{length(T_{outliers})}{length(T)}$$

Skewed

The skewness of a minimum spanning tree is a measure of the extreme-value distribution of the edge lengths. The skewed scagnostic is calculated by

$$c_{skew} = \frac{q_{90} - q_{50}}{q_{90} - q_{10}}$$

3 Methods: Building Our Model

To begin our analysis of scagnostics and their ability to detect patterns and differentiate scatterplots, we simulated data from 2-dimensional point distributions. We generated linear, clustered, striated, exponential, increasing-variance (funnel), and quadratic data for our primary family dataset. For this dataset, we had 14865 signal plots belonging to the six distributions mentioned above, as well as 14865 null plots generated to have no interesting underlying distribution. The initial goal of creating these plots was to serve as a proof of concept that scagnostics could be used in a statistical learning model to classify signal and null plots correctly. They were later used in a lineup scenario to compare scagnostics to human perception. Using a variety of statistical learning models, both supervised and unsupervised, we computed accuracy on both individual plots and lineups.

3.1 Statistical Learning Algorithms

Our initial goal is essentially a classification problem: we want to train a computer to use scagnostics to classify a plot as being either signal or null. To achieve this, we used statistical learning algorithms drawn from both unsupervised and supervised methods. Supervised learning methods are the creation of models based on predictors and known classifications of objects. These models relate the predictors to the classification in a way that we can use to predict future objects' groups. Unsupervised methods, on the other hand, form groups or infer patterns from unlabeled data. The classifications of the training data are not given to the unsupervised model, and thus it creates categories based on patterns we may not even know are present.

We used only supervised methods in our proof-of-concept case, in which we classified individual plots as being signal or null. The purpose of this was to evaluate the efficacy of using scagnostics for pattern detection. With these supervised models, we trained our models on a subset of the primary family dataset mentioned previously, and then tested and cross-validated our models on the the remaining data. Since our end goal was to use lineups to test human perception compared to scagnostics, we created lineups from this data. The 2000 lineups consisted of 20 plots; half of the generated lineups contained 19 null plots and 1 signal, and half contained an unknown number of signal plots. For the lineups, we used both supervised and unsupervised methods: supervised similarly to before, and unsupervised as outlier detection. We wanted to see if scatterplots marked as outlying based on their scagnostic values, as compared to the rest of the plots in the lineup, were likely to be signal. We tested a variety of models which are explained below, and compared them all to human perception.

3.1.1 Supervised Learning

Logistic Regression

Logistic regression is used when the response variable is categorical and there are only two responses. The outcome can be predicted using one or more explanatory variables X. For

example, if detecting a signal plot from a field of nulls, the dependent variable Y is signal (1 for signal, 0 for noise), and the explanatory variables are the nine scagnostics. We can model the logit of $p(X)$ linearly as

$$log\frac{p(X)}{1-p(X)} = \beta_0 + \beta_1 X$$

Then, the odds of the response $Y = 1$ is

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}$$

A high odds indicates a high likelihood of being a success. Thus, probability of the response $Y = 1$ given all explanatory variables X is given by the logistic function

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

The estimation of $\hat{\beta}_0$ and $\hat{\beta}_1$ is carried out by maximizing the likelihood function:

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1-p(x_{i'}))$$

Logistic regression can fail if there is perfect separation in some models if the probability of the event occurring is 1 ($p(X) = 1$). There is not a closed form for the maximum likelihood estimates of the $\hat{\beta}$s, but with perfect separation, they cannot be calculated numerically since the likelihood approaches infinity and the estimates of $\hat{\beta}$s get very large. We can also see that in this case, the odds can't be calculated since it involves a division by 0; this makes sense as likelihood of a success approaches infinity if the probability of success is 1. We encountered this problem when we tested our model on some of our individual distributions that measures very high on a single scagnostic, namely striated and linear. However, we still were able to use logistic regression in our combined primary family dataset and lineup cases.

K-Nearest Neighbors

K-nearest neighbors is a classification method that classifies a new data point as being the same class as majority of its k nearest neighbors. We used Euclidean distance to determine the distances between points in our model, and tried values of k ranging from 1 to 25. For each model, the value of k that maximized classification accuracy was chosen.

Random Forest

A random forest model is one of the more flexible statistical learning approaches because it can be used for quantitative or categorical response variables as well as any form of explanatory variable. At the base of every random forest is the decision tree algorithm. A decision tree produces a series of binary splits based on the predictor set in order to divide the data into homogeneous subsets. In order to determine a split, the algorithm considers all possible splits of each predictor and chooses the split that results in the largest possible drop in node impurity. Node impurity is typically measured using either the Gini impurity or total entropy–two numerically similar measures of total variance across all classes. Our random forest models were constructing based on Gini impurity, which is a measure of the likelihood of a randomly chosen element within a subset being mislabeled if its labelled according to the distribution of labels within that subset. This splitting procedure is then repeated for the two resulting subsets and the subset whose maximal split produces the biggest drop in impurity is split. This process repeats until a stopping criteria is met. The predicted class of an observation is generally the

majority class of the terminal node in which it falls.

While decision trees offer an intuitive way of classifying observations, they are not very robust, which makes them prone to overfitting. In order to combat this disadvantage, random forests use many decision trees and aggregate their results. In order to avoid overfitting, random forests begin by drawing a bootstrap sample to use as a training set. Then, a random subset of the overall predictor set is considered at each split in order to decorrelate the trees. Each tree with a strict stopping criteria, so that there are many splits in each tree giving each low bias and high variance. This process is repeated for potentially hundreds of trees, and the accuracy of the model is determined by predicting an observation using the trees trained on bootstrap samples that did not include that observation. The final prediction is made by receiving a prediction from each individual tree and taking the majority prediction for that observation.

Linear Discriminant Analysis

Logistic regression can be effective when modeling data into two response classes that are not well separated. However, if we are concerned with data that can be sorted into more than two classes or classes that are well separated, linear discriminant analysis proves more robust. *Linear discriminant analysis* (LDA) allows us to model the probability that a given observation belongs to a particular response class indirectly by first modeling the distribution of predictors for each response class individually.

LDA assumes that our data $X = (X_1, X_2, \ldots, X_p)$ are drawn from a multivariate normal distribution $X \sim N(\mu_k, \Sigma)$ where $\mu_k = E(X|Y = k)$ is the class-specific mean vector and $\Sigma = Cov(X)$ is a covariance matrix shared by all classes. If we denote K to be the set of all classes, our classification function for an observation $X = x$ is given by:

$$\delta(x) = \max_{k \in K} \left[x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k \right]$$

where π_k is the prior probability that the observation $X = x$ belongs to the k^{th} response class. Often this prior is taken to be the proportion of training observations that belong to the k^{th} class. It is important to note that the LDA classifier is trying to approximate the Bayesian classifier, which would give us the lowest total error rate out of all classifiers, assuming that our initial assumption of multivariate linear data holds. One key element of this assumption is that all classes follow the same covariance matrix. However, this may not always be the case.

Quadratic Discriminant Analysis

We can relax this assumption of a common covariance matrix by using *quadratic discriminant analysis* (QDA). Now, we assume that observations belonging to a given class k are drawn from a multivariate normal distribution $X \sim N(\mu_k, \Sigma_k)$ where both the mean vector and covariance matrix are class-specific. For a given observation $X = x$, this gives us the classification function

$$\delta(x) = \max_{k \in K} \left[-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) - \frac{1}{2} \log |\Sigma_k| + \log \pi_k \right]$$

where π_k is the prior density for response class k.

The main distinction between LDA and QDA is in the bias-variance trade-off. LDA assumes a shared covariance across all classes, making it less flexible as a classifier. This in turn leads to a decreased variance as compared to QDA, but may suffer from increased bias should the assumption of a common covariance matrix be wrong. Both methods of classification were used during the analysis of our simulated data, which will be discussed later.

3.1.2 Unsupervised Learning

We used unsupervised learning methods specifically in our lineup detection. We made this decision because unsupervised methods more closely mimic how we as humans approach a lineup problem: we go by visual patterns to determine which plot is different from the rest, rather than using a statistical model based on scagnostic values or previous experience with plots. Both unsupervised methods were used for outlier detection and are types of distance metrics.

Scencio not introduced yet

Euclidean Distance

The first unsupervised outlier detection method we used was Euclidean distance. We determined the 9-dimensional Euclidean distance for each lineup plot relative to both the mean and median of the other 19 plots, and selected that which had the greatest distance to be our predicted signal plot. In the case of multiple signal plots, we told ranked the distances from greatest to smallest. In this case, we ~~told the model~~ how many signal plots there were (n) and just chose the top n distances to be our predicted signal plots. If we have coordinates $\vec{x} = (x_1, x_2, ..., x_n)$ and $\vec{y} = (y_1, y_2, ..., y_n)$, then the distance from $\vec{x}$ to $\vec{y}$ or $\vec{y}$ to $\vec{x}$ is

In stats bold face is often used to denote vectors.

$$d(\vec{x}, \vec{y}) = d(\vec{y}, \vec{x}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Mahalanobis Distance

~~However, using~~ Euclidean distance, ~~we did~~ does not take into account correlation between ~~our~~ the scagnostic values. This correlation exists since some use the same length or quantile measures as others. ~~Mahalanobis distance was our solution to this problem: it still uses distance to determine the outlier plot(s) relative to the rest, but weights distances based on the correlation among scagnostic values.~~ To account for this can, we need Mahalanobis dist, which is defined as

The Mahalanobis distance of an observation $\vec{x} = (x_1, x_2, x_3, ..., x_N)^T$ from a set of observations with mean $\vec{\mu} = (\mu_1, \mu_2, \mu_3, ..., \mu_N)^T$ and covariance matrix S is defined as

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}.$$

Σ is "conventional"

4 Results

Avoid too many points to the future, tell the story as you go.

Before we could begin comparing our statistical learning algorithms to human perception, we first needed to ensure that our scagnostic measures could indeed be used to train a model to detect patterns. To test this, we fit our algorithms on a set of training data for each of our six primary family distributions, as well as on the primary family dataset as a whole, and tested the prediction accuracy of our statistical learning methods on a test subset of the data. This process used a 10-fold cross-validation, and accuracies are given in Table 1.

We can see from Table 1 that nearly all of our statistical learning algorithms managed a high degree of accuracy across all distributions of our simulated data, as well as when the individual distributions were combined into the primary family. Now that we have confirmed the ability to detect patterns based solely on the criteria of our nine scagnostics, we can move toward comparing our statistical models to human perception.

make specific statements.

Table 1: Model Accuracies

	Logistic	LDA	QDA	K-Nearest Neighbors	Random Forest
Linear	.958	.854	.946	1.00	.956
Cluster	.998	.987	.998	.761	.997
Striated	.995	.976	.986	1.00	.993
Funnel	.983	.855	.972	.98	.977
Exponential	1.00	.975	.989	1.00	1.00
Quadratic	.999	.989	.995	1.00	.998
Primary Family	.974	.939	.981	.696	.986

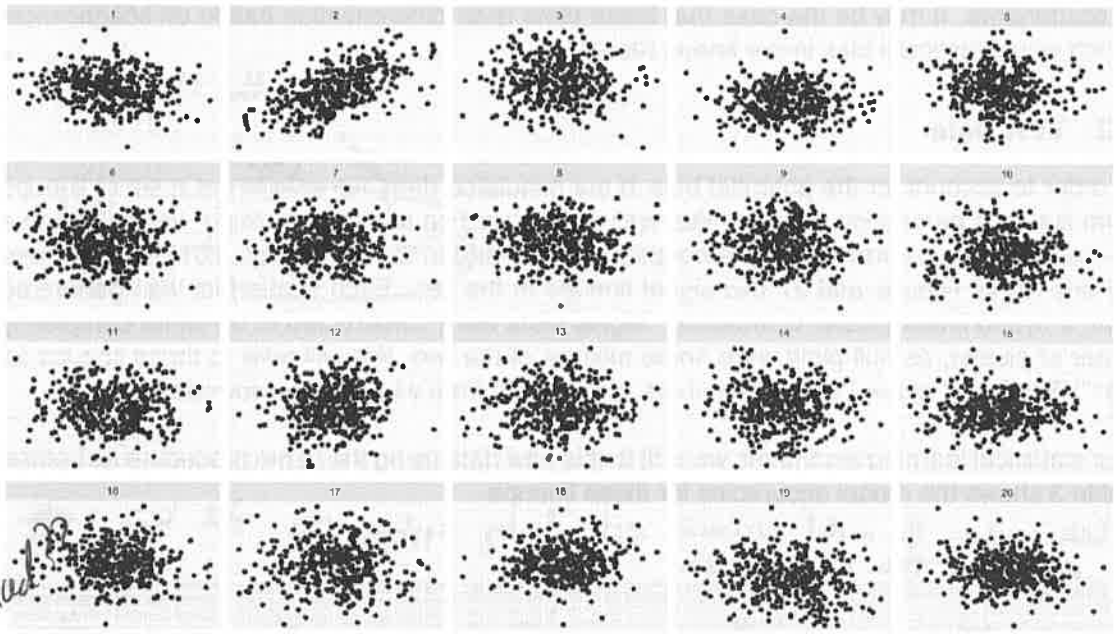
From our initial results, we concluded that machine learning methods armed with scagnostics can be used to determine signal from null not only when trained on individual distributions, but also when trained on our joint-distribution primary family dataset which included plots drawn from linear, cluster, striated, funnel, exponential and quadratic distributions. While all machine learning algorithms performed relatively well, we can see that when trained on our primary family dataset, k-nearest neighbors performed substantially worse than our other methods. Further, random forest performed the best (98.6% accuracy), followed by quadratic discriminant analysis (98.1%), logistic regression (97.4%), and linear discriminant analysis (93.9%).

4.1 Lineups

In order to make a valid comparison between human perception and our statistical learning methods, we require a task that is analogous in both domains. To do this, we employed a common mechanism in visual inference research: the lineup. A scatterplot lineup is a 4-row, 5-column grid of 20 scatterplots in which the object is to identify one or more target scatterplots from a set of decoys. For our purposes, these target scatterplots will demonstrate a pattern while the decoys will be our null plots.

For this research, we distinguished between two main categories of lineups. One-signal lineups used only one target plot and 19 decoys, whereas unknown-signal lineups used multiple target scatterplots. The number of target scatterplots in the unknown condition was not provided to our statistical models, allowing them to identify any number of scatterplots as potential targets. All of our statistical learning methods were tested on 1000 one-signal lineups and 1000 unknown-signal lineups for both linear trend data and our combined primary family dataset. These accuracies are given in Table 2. Our supervised learning methods were trained using the full, original dataset, either linear or primary family, that excluded the scatterplots chosen for the lineup. The lineup scatterplots then served as the test data for which model predictions were determined. For these lineups, we also used two unsupervised learning methods, Euclidean distance and Mahalanobis distance.

In Table 2, accuracies for the one-signal condition are the proportion of lineups in which the correct signal plot was identified. In the unknown-signal plot condition, accuracy is the percentage of target signal plots that were identified. The number in parentheses for the unknown-signal accuracies is the rate of false positives, which is calculated as the percentage of plots identified by the model that were not correct signal plots. Since are models were not given the total number of signal plots to identify, thereby allowing any number of chosen signal plots, accuracy and rate of false positives are not necessarily inverse. These two values are inverses in the one-signal condition, so the rate of false positives is omitted here.



Not referenced??

Figure 2: Example of lineup generated from linear trend data; signal plot is number $\sqrt{9} - 1$.

Table 2: Lineup Accuracies

	One-Signal		Unknown-Signal	
	Linear Trend	Primary Family	Linear Trend	Primary Family
Logistic Regression	.936	.975	.927 (.033)	.962 (.022)
LDA	.924	.941	.705 (.000)	.864 (.026)
QDA	.912	.965	.926 (.059)	.972 (.037)
K-Nearest Neighbors	.959	.952	.898 (.027)	.953 (.018)
Random Forest	.992	.989	.932 (.053)	.979 (.030)
Euclidean Distance	.518	.774	.665 (.335)	.819 (.181)
Mahalanobis Distance	.857	.956	.628 (.372)	.722 (.278)

Note: Rate of false positives is given in parentheses

For lineups with a single signal plot, we found that random forest had the highest accuracy (99.2% for linear trends and 98.9% for our primary family data), while Euclidean distance had the lowest accuracy (51.8% for linear and 77.4% for primary family). For ~~our~~ lineups with multiple signal plots, again, random forest performed best (93.2% for linear and 97.9% for primary family), but in this case, Mahalanobis distance was the least accurate method (62.8% for linear and 72.2% for primary family); this model also had the highest false positive rates (37.2% for linear and 27.8% for primary family). Though we had a few poor performances, the majority of our models gave accuracies over 90% for the one-signal plots, and over 70% for multiple signal plots (along with false positive rates under 6%) from our primary family and linear data.

These results were very promising; however, it is worth noting here that although Table 2 demonstrates a strikingly high degree of accuracy for our statistical learning methods, they may not be entirely accurate. These lineups were generated from our simulated data, which were constructed for the sole purpose of evaluating the ability of scagnostics to detect patterns

in scatterplots. It may be the case that these plots favor differentiation based on scagnostics, which would instigate bias in our lineup results.

4.2 VPH Data

In order to account for the potential bias in our simulated data, we sought out a set of lineups from previous perception research that was not focused on machine learning. We borrowed a series of 47 lineups from a lineup perception study (VanderPlas & Hoffman, 2017). There were 20 one-signal lineups and 27 two-signal lineups in the set. Each scatterplot was generated from a hybrid linear-cluster distribution. Signal plots were constructed to be either completely linear or cluster, as null plots were some mixture of the two. We will refer to these lineups as our "VPH Data," named after the authors of the study from which we borrowed them.

Our statistical learning algorithms were fit to this new data using the same procedure as before. Table 3 shows the model accuracies for these lineups.

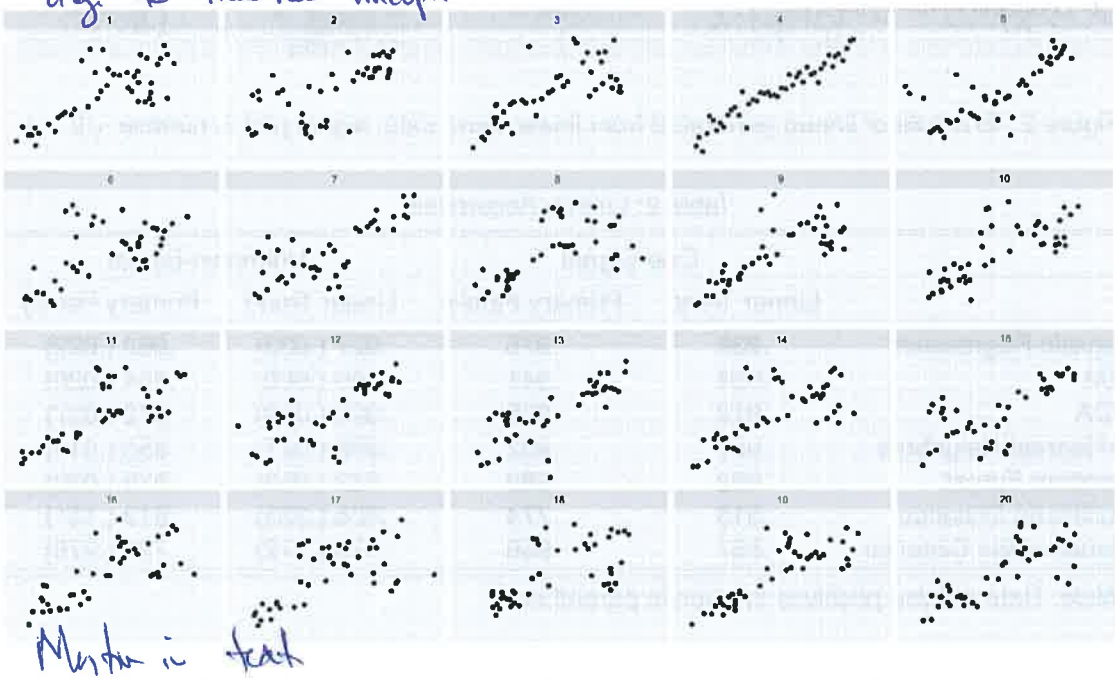


Figure 3: Example of VPH-generated lineup. In this lineup there is a single signal plot, number $2^3 - 4$.

Again, we find that our most of our models managed a relatively high degree of accuracy, with random forest again out-performing all of the other predictive methods (Table 3). Across the board, however, we notice a slight drop in accuracies for the unknown-signal condition, with most algorithms returning accuracies around 70-85%. Logistic regression and random forest both had 100% accuracy on the one signal lineups; all other models - except Mahalanobis with 60% - were above 90% correct. In the unknown signal lineups, k-nearest neighbors and random forest did best, with 87% and 92.6% accuracy, respectively. Again, all other models except Mahalanobis (53.7%) did well and were above 70% accurate. Overall, we showed that both supervised and unsupervised models using scagnostics could very effectively pick out both single and multiple signal plots from a field of nulls.

Mention in text.

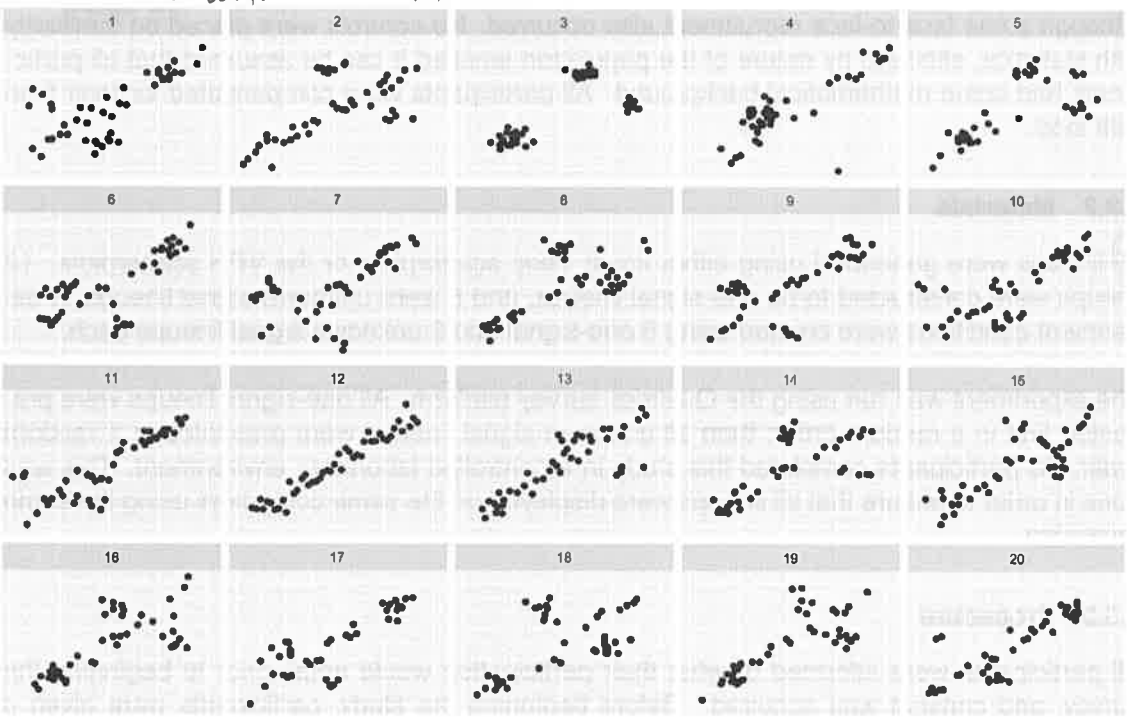


Figure 4: Example of VPH-generated lineup. In this lineup there is a multiple signal plots: numbers $\frac{6\times2}{4}$ and $\frac{30}{2} - 3$.

Table 3: VPH Lineup Accuracies

	One-Signal	Unknown-Signal
Logistic Regression	1.000	.722 (.278)
LDA	.900	.704 (.296)
QDA	.950	.767 (.233)
K-Nearest Neighbors	.900	.870 (.130)
Random Forest	1.000	.926 (.074)
Euclidean Distance	.900	.852 (.148)
Mahalanobis Distance	.600	.537 (.463)

Note: Rate of false positives is given in parentheses

On the whole, it appears that our statistical learning methods are able to effectively use scagnostics to make predictions regarding the presence of patterns in scatterplots. Our next step is to compare how these algorithms perform relative to human perception.

4.3 Human Perception Experiment

In order to compare our machine learning models to perception of patterns in humans, we conducted a study at our college.

4.3.1 Participants

The Participants were 50 college students between 18 and 22 years of age. Recruitment was managed primarily through email to a list of students interested in mathematics and statistics,

although some face-to-face recruitment also occurred. No controls were placed on familiarity with statistics, although by nature of the population emailed it can be assumed that all participants had some mathematical background. All participants were compensated for their time with food.

4.3.2 Materials

18 lineups were generated using either linear trend scatterplots or the VPH scatterplots. 12 lineups were constructed to be one-signal lineups, and 6 were unknown-signal lineups. 2 experiment conditions were created using 6 one-signal and 3 unknown-signal lineups each.

The experiment was run using the Qualtrics survey platform. All one-signal lineups were presented first in a random order, then all unknown-signal lineups were presented in a random order. All participants completed this study in a controlled laboratory environment. This was done in order to ensure that all lineups were displayed on the same computers using the same resolution.

4.3.3 Procedure

All participants were informed of what their participation would entail prior to beginning the survey, and consent was acquired. Before beginning the study, participants were given a practice lineup in order to familiarize themselves with the task. 6 one-signal lineups were then presented, one at a time, and responses were recorded. For these lineups, participants were informed: "For the following lineups, there may be multiple plots that are different from the majority. This means that there could only be 1 different plot, or there could be several plots that are different from the others. Please select all plots that you believe are different from the others." After completing all 9 lineups, participants were thanked for their time and were provided with food as compensation.

Fit or applied? Did you rerun estimation?

4.3.4 Experimental Results

Our seven statistical learning algorithms were fit to the 18 lineups used in this study using the same procedure as before. Prediction accuracies are given in Table 4.

Table 4: Experiment Lineup Accuracies

	One-Signal	Unknown-Signal
Logistic Regression	.833	.883 (.150)
LDA	.833	.917 (.139)
QDA	.833	.767 (.189)
K-Nearest Neighbors	.917	.917 (.309)
Random Forest	.917	.917 (.083)
Euclidean Distance	.833	.878 (.122)
Mahalanobis Distance	.750	.628 (.291)
Participant Accuracy	.805	.720 (.105)

Note: Rate of false positives is given in parentheses

Our participants had an accuracy of 80.5% with the single signal plots. This was worse than each of our models except Mahalanobis distance. With multiple-signal lineups, our participants had an accuracy of 72%, which was again worse than every model except Mahalanobis.

However, our participants did have a lower false positive rate than every model except random forest (which also happened to have the highest accuracy, 91.7%). Thus, while our participants tended to perform worse than the models in terms of number of correct plots selected, they were also more conservative in picking plots overall: both the model and the participants were allowed to pick as many plots as they thought were signal, but our human participants seemed to pick fewer. Perhaps there is a cutoff for the human brain in terms of being able to detect a variety of patterns in plots compared to a majority.

4.4 QQ plots

Although our primary concern was determining whether scagnostics could aid in detecting relationships between variables in a traditional scatterplot setting, we also considered how scagnostics might help in differentiating between other types of plots. Specifically, we explored the effectiveness of our scagnostic measures at assessing normality in QQ plots. In order to do this, we generated 16,000 QQ plots from a variety of normal and non-normal distributions. Half of our data came from a normal distribution, and the remaining plots were created from data simulated from t-distributions, exponential distributions, log-normal distributions, and Chi-squared distributions. Sample sizes and parameter values were varied in order to generate a variety of non-normal behavior. Then, scagnostic values were calculated for each plot as predictors for non-normality.

As a baseline, the data for each of our 16,000 QQ plots were classified using the Anderson-Darling normality test. We found that this traditional normality test accurately identified 81.8% of our simulated data as either normal or non-normal. We then tested the effectiveness of our scagnostic measures using 10-fold cross validation for a variety of statistical learning models fit to our simulated data. The top cross-validated accuracy came from our random forest model with 78.6% accuracy.

Although our nine scagnostic values were developed to detect patterns in scatterplots, they were not designed to specifically quantify the behavior in QQ plots. QQ plots are entirely monotonic and as a viewer, we understand that perfect adherence to the normal distribution should result in all points falling along the line $y = x$ (for standardized data). Therefore, we decided that we may need an additional scagnostic in order to mirror the visual cues that a viewer uses to identify non-normality. In order to do this, we attempted to quantify deviations from this theoretical line by taking an average of the vertical distances between the points on our QQ plot and the line $y = x$. To justify this comparison, we standardized our data by subtracting by the mean and dividing by the standard deviation. Therefore, under the null hypothesis that our original data is normal, our transformed data should fall under the standard normal distribution. Furthermore, we experimented with placing a larger penalty for deviations from our theoretical line in the tails of our plot in order to potentially detect some of the behavior visible in t-distributions with low degrees of freedom. We tested a variety of different weighting functions in order to empirically determine how much to penalize extreme values. Ultimately, based on accuracy within our simulated data, we arrived at our "deviation" scagnostic shown below.

$$c_{deviation} = \frac{1}{n}\sum_{i=1}^{k}((x_i^2 + 1)(y_i - x_i)^2)$$

Armed with this new scagnostic, we refit our models and tested their accuracy on our simulated data. Ultimately, after including our deviation scagnostic, we found that our top cross-validated accuracy was 84.0% coming again from our random forest model. This performance was

not only an improvement over our original model, but also provided more accuracy than the Anderson-Darling test alone. There is still substantial room for improvement, but it appears that developing further measures to quantify the visual cues available through QQ plots could aid in identifying non-normality. Overall, the effectiveness of this model illustrates how scagnostics and statistical learning can be used in a variety of settings as well as the potential to develop specialized scagnostics to detect particular behavior in specific plot types.

5 Discussion

Unsupervised vs. Supervised Models

An important question in our research is whether supervised or unsupervised methods are preferred for pattern detection in scatterplots. Overall, we found that supervised methods led to not only higher accuracies, but in the multiple-signal lineups also lower false positive rates. The initial lineup accuracies (Table 2) demonstrate a clear disconnect between the pattern detection capabilities of supervised and unsupervised learning methods. In the one-signal condition, the Euclidean distance predictions vastly under-perform relative to our other learning methods. Mahalanobis distance, however, manages to return similar accuracies to the five supervised algorithms. When we look at the unknown-signal condition, however, both Euclidean and Mahalanobis distance fail to measure comparably with our supervised methods. With the VPH lineups, the Mahalanobis distance predictions are far less accurate than the other measures, with Euclidean predicting comparably to the supervised methods. When looking at the human perception experiment data, we see again that Mahalanobis distance performs worse than the other metrics, with Euclidean distance performing like the supervised methods once again. Thus, in our three different lineup comparison studies, we see that supervised methods are consistently more accurate than Euclidean and Mahalanobis distance. Therefore, while we hoped that unsupervised methods could reveal unknown structure within the data, and use this structure to determine the signal plots with high accuracy, it appears that supervised methods are more appropriate for our research interests.

Limitations

The topic of supervised versus unsupervised learning brings us to one of our main limitations: the use of Euclidean and Mahalanobis distance as our only unsupervised methods. These are both examples of outlier detection. Further research could try to use different unsupervised methods. For example, we could potentially use unsupervised forms of clustering in order to detect more general patterns in how scagnostics differentiate plots - these methods would not give us a signal or null classification, but they could further illuminate exactly which visual patterns scagnostics distinguish among scatterplots..

Another limitation we faced was the scope of our human perception study. While our models performed very well compared to our participants, our population of volunteers was not very diverse or large. Additionally, most of our participants had taken at least one course in statistics, so their familiarity with scatterplots may have aided their performance. Conversely, most of our participants were not statistics majors, so it's possible that while our model outperformed the average participant, certain viewers could still be more effective than our models. In order to explore this further, more lineup perception studies would have to be conducted on a variety of different populations.

Applications

Considering our model's success to detect pattern from null, we believe that our research can be used to aid statisticians in their analysis of large data sets. Our model could be used to both identify signal plots, flagging them for further visual and quantitative analysis, as well as suggest which relationships between other variables may not have significant patterns and do not warrant further examination. Moving toward this goal, we created a simple Shiny application that when given a large dataset, can return a list of signal and null plots within a matter of seconds, and even classify the signal plots as being one of our primary family data types.

Future Directions

While our use of scagnostics in our tested types of scatterplots proved to be very promising, they were less helpful when it came to types of scatterplots with specific structure including QQ plots, time series, and residual plots. Through our successful initial exploration of a potential new scagnostic for classifying QQ plots as normal or not, we suggest that future scagnostic-type values could potentially be of use for detecting patterns in these unique scatterplots.

6 Conclusion

Throughout the various steps of our research, we have shown that scagnostics can be used to characterize scatterplots more effectively and efficiently than humans. First, we established that scagnostics can be used to distinguish signal from null when trained on specific distributions. Next, we determined that computers can be trained to "see" a wide variety of distributions as signal using scagnostics. Finally, using lineups, we found that our model performs better than humans to detect the most dissimilar plot or plots from a series of null plots. Thus, we believe scagnostics can be a promising tool for statisticians in the future.

A Do we have a IRB #? If so, we need to include this.

References

Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E., Swayne, D., & Wickham, H. (2009). Statistical inference for exploratory data analysis and model diagnostics. Philosophical Transactions Of The Royal Society A: Mathematical, Physical And Engineering Sciences, 367(1906), 4361-4383.

Hartigan, J.A., & Mohanty, S. (1992). The runt terrst for multimodality.Journal of Classification, 9:63–70.

Hofmann, H., Follett, L., Majumder, M., & Cook, D. (2012). Graphical Tests for Power Comparison of Competing Designs. IEEE Transactions On Visualization And Computer Graphics, 18(12), 2441-2448.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R (pp. 15-57, 127-173, 303-335), New York: Springer.

Loy, A., Follett, L., & Hofmann, H. (2016). Variations of Q–Q Plots: The Power of Our Eyes!. The American Statistician, 70(2), 202-214.

Marchette, D. (2004). Random graphs for statistical pattern recognition (pp. 1-71). Hoboken, NJ: Wiley-Interscience.

Sigbert. (2012). Scagnostics Base (digital image). Retrieved from Wikimedia Commons website: https://commons.wikimedia.org/wiki/File:ScagnosticsBase.svg

Tukey, J. W. (1974) Mathematics and the picturing of data. Proceedings of the International Congress of Mathematicians, (pp. 523–531), Vancouver, Canada.

Tukey, J. W. & Tukey, P. A. (1985). Computer Graphics and Exploratory Data Analysis: An Introduction. Proceedings of the Sixth Annual Conference and Exposition: Computer Graphics, Fairfax, VA, United States, National Computer Graphics Association.

VanderPlas, S., & Hofmann, H. (2017). Clusters Beat Trend!? Testing Feature Hierarchy in Statistical Graphics. Journal of Computational and Graphical Statistics, 26(2), 231-242.

Wilkinson, L. (2018). Visualizing Big Data Outliers Through Distributed Aggregation. IEEE Transactions On Visualization And Computer Graphics, 24(1), 256-266.

Wilkinson, L., Anand, A., & Grossman, R. (2005). Graph Theoretic Scagnostics. Symposium on Information Visualization, Minneapolis, MN, United States, October 25.

Wilkinson, L., Anand, A., & Grossman, R. (2006). High-Dimensional Visual Analytics: Interactive Exploration Guided by Pairwise Views of Point Distributions. IEEE Transactions On Visualization And Computer Graphics, 12(6), 1363-1372.

Wilkinson, L., & Wills, G. (2008). Scagnostics Distributions. Journal Of Computational And Graphical Statistics, 17(2), 473-491.