Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing

Justin Matejka and George Fitzmaurice
Autodesk Research, Toronto Ontario Canada
{*first.last*}@autodesk.com

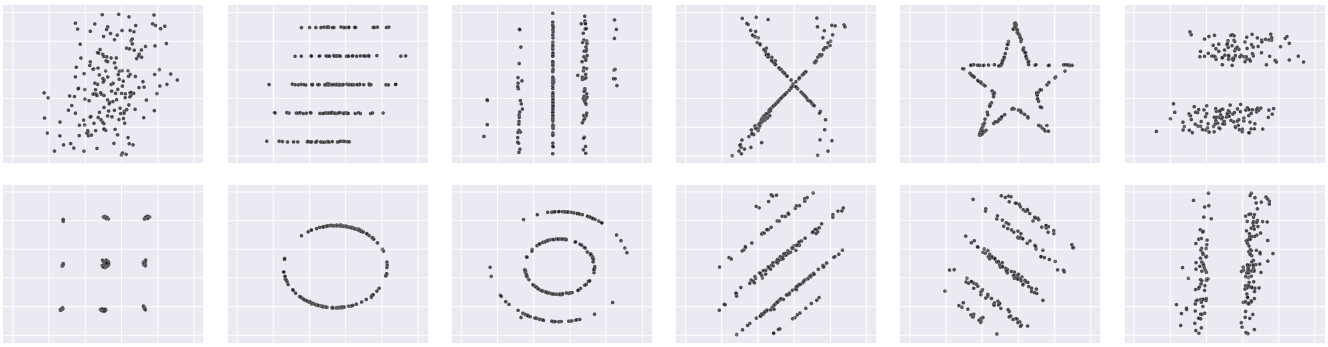

Figure 1. A collection of data sets produced by our technique. While different in appearance, each has the same summary statistics (mean, std. deviation, and Pearson's corr.) to 2 decimal places. ($\bar{x}$ =54.02, $\bar{y}$ = 48.09, sd_x = 14.52, sd_y = 24.79, Pearson's r = +0.32)

ABSTRACT

Datasets which are identical over a number of statistical properties, yet produce dissimilar graphs, are frequently used to illustrate the importance of graphical representations when exploring data. This paper presents a novel method for generating such datasets, along with several examples. Our technique varies from previous approaches in that new datasets are iteratively generated from a seed dataset through random perturbations of individual data points, and can be directed towards a desired outcome through a simulated annealing optimization strategy. Our method has the benefit of being agnostic to the particular statistical properties that are to remain constant between the datasets, and allows for control over the graphical appearance of resulting output.

INTRODUCTION

Anscome's Quartet [1] is a set of four distinct datasets each consisting of 11 (*x,y*) pairs where each dataset produces the same summary statistics (mean, standard deviation, and correlation) while producing vastly different plots (Figure 2A). This dataset is frequently used to illustrate the importance of graphical representations when exploring data. The effectiveness of Anscombe's Quartet is not due to simply having four different data sets which generate the same statistical properties, it is that four *clearly different* and *identifiably distinct* datasets are producing the same statistical properties. Dataset I appears to follow a somewhat noisy linear model, while Dataset II is following a parabolic distribution. Dataset III appears to be strongly linear, except for a single outlier, while Dataset IV forms a vertical line with the regression thrown off by a single outlier. In contrast, Figure 2B shows a series of datasets also sharing the same summary statistics as Anscombe's Quartet, however without any obvious underlying structure to the individual datasets, this quartet is not nearly as effective at demonstrating the importance of graphical representations.

While very popular and effective for illustrating the importance of visualizations, it is not known how Anscombe came up with his datasets [5]. Our work presents a novel method for creating datasets which are identical over a range of statistical properties, yet produce dissimilar graphics. Our method differs from previous by being agnostic to the particular statistical properties that are to remain constant between the datasets, while allowing for control over the graphical appearance of resulting output.

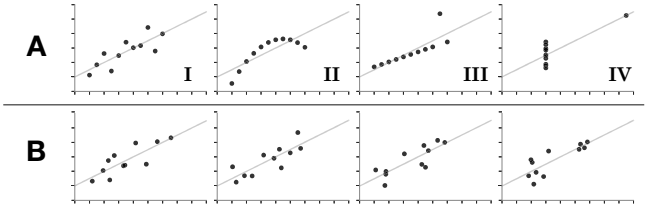

Figure 2. (A) Anscombe's Quartet, with each dataset having the same mean, standard deviation, and correlation. (B) Four *unstructured* datasets, each also having the same statistical properties as those in Anscombe's Quartet.

CHI 2017, May 06 - 11, 2017, Denver, CO, USA

ACM 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025912

RELATED WORK

As alluded to above, producing multiple datasets with similar statistics and dissimilar graphics was introduced by Anscombe in 1973 [1]. "Graphs in Statistical Analysis" starts by listing three notions prevalent about graphs at the time:

(1) *Numerical calculations are exact, but graphs are rough;*
(2) *For any particular kind of statistical data there is just one set of calculations constituting a correct statistical analysis;*
(3) *Performing intricate calculations is virtuous, whereas actually looking at the data is cheating.*

While one cannot argue that there is currently as much resistance towards graphical methods as when Anscombe's paper was originally published, the datasets described in the work (Figure 1A) are still effective and frequently used for introducing or reinforcing the importance of visual methods. Unfortunately, Anscombe does not report how the datasets were created, nor suggest any method to create new ones.

The first attempt at producing a generalized method for creating such datasets was published in 2007 by Chatterjee and Firat [5]. They proposed a genetic algorithm based approach where 1,000 random datasets were created with identical summary statistics, then combined and mutated with an objective function to maximize the "graphical dissimilarity" between the initial and final scatter plots. While the datasets produced were graphically dissimilar to the input datasets, they did not have any discernable structure in their composition. Our technique differs by providing a mechanism to direct the solutions towards a specific shape, as well as allowing for variety in the statistical measures which are to remain constant between the solutions.

Govindaraju and Haslett developed a method for regressing datasets towards their sample means while maintaining the same linear regression formula [7]. In 2009, the same authors extended their procedure to creating "cloned" datasets [8]. In addition to maintaining the same linear regression as the seed dataset, their cloned datasets also maintained the same means (but not the same standard deviations). While Chatterjee and Firat [5] wanted to create datasets as graphically dissimilar as possible, Govindaraju and Haslett's cloned datasets were designed to be visually similar, with a proposed application of confidentializing sensitive data for publication purposes. While our technique is primarily aimed at creating visually distinct datasets, by choosing appropriate statistical tests to remain constant through the iterations (such as a Kolmogorov-Smirnov test) our technique can produce datasets with similar graphical characteristics as well.

In the area of generating synthetic datasets, GraphCuisine [2] allows users to direct an evolutionary algorithm to create network graphs matching user-specified parameters. While this work looks at a similar problem, it differs in that it is focused on network graphs, is an interactive system, and allows for directly specifying characteristics of the output, while our technique looks at 1D or 2D distributions of data, is non-interactive, and perturbs the data such that the initial statistical properties are maintained throughout the process.

Finally, on the topic of using scatter plots to encode graphics, Residual Sur(Realism) [11] produces datasets with hidden images which are only revealed when appropriate statistical measures are performed. Conversely, our technique encodes graphical appearance into the data directly.

METHOD

The key insight behind our approach is that while generating a dataset from scratch to have particular statistical properties is relatively *difficult*, it is relatively *easy* to take an existing dataset, modify it slightly, and maintain (nearly) the same statistical properties. With repetition, this process creates a dataset with a different visual appearance from the original, while maintaining the same statistical properties. Further, if the modifications to the dataset are biased to move the points towards a particular goal, the resulting graph can be directed towards a particular visual appearance.

The pseudocode for the high-level algorithm is listed below:

```
1:  current_ds ← initial_ds
2:  for x iterations, do:
3:      test_ds ← PERTURB(current_ds, temp)
4:      if ISERROROK(test_ds, initial_ds):
5:          current_ds ← test_ds
6:
7:  function PERTURB(ds, temp):
8:      loop:
9:          test ← MOVERANDOMPOINTS(ds)
10:         if FIT(test) > FIT(ds) or temp > RANDOM():
11:             return test
```

INITIAL_DS is the seed dataset from which the statistical values we wish to maintain are calculated. The PERTURB function is called at each iteration of the algorithm to modify the latest version of the dataset (CURRENT_DS) by moving one or more points by a small amount, in a random direction. The "small amount" is chosen from a normal distribution and is calibrated such that >95% of movements result in the statistical properties of the overall dataset remaining unchanged (to two decimal places).

Once the individual points have been moved, the FIT function is used to check if perturbing the points has increased the overall fitness of the dataset. The fitness can be calculated in a variety of ways, but for conditions where we want to coerce the dataset to into a shape, fitness is calculated as the average distance of all points to the nearest point on the target shape.

The naïve approach of accepting only datasets with an improved fitness value results in possibly getting stuck in locally-optimal solutions where other, more globally-optimal solutions are possible. To mitigate this possibility, we employ a simulated annealing technique [9]. With the possible solutions generated in each iteration, simulated annealing works by *always* accepting solutions which

improve the fitness, but also, if the fitness is not improved, the solution *may* be accepted based on the "temperature" of the simulated annealing algorithm. If the current temperature is less than a random number between 0 and 1, the solution is accepted even if it the fitness is worsened. We found that using a quadratically-smoothed monotonic cooling schedule starting with a temperature of 0.4 and finishing with a temperature of 0.01 worked well for the sample datasets.

Once the perturbed dataset has been accepted, either through an improved fitness value or from the simulated annealing process, the perturbed dataset is compared to the initial dataset for statistical equivalence. For the examples in this paper we consider properties to be "the same" if they are equal to two decimal places. The IsErrorOK function compares the statistics between the datasets, and if they are equal (to the specified number of decimal places), the result from the current iteration becomes the new current state.

Example Generated Datasets

Example 1: Coercion Towards Target Shapes

In this first example (Figure 1), each dataset contains 182 points and are equal (to two decimal places) for the "standard" summary statistics (*x*/*y* mean, *x*/*y* standard deviation, and Pearson's correlation). Each dataset was seeded with the plot in the top left. The target shapes are specified as a series of line segments, and the shapes used in this example are shown in Figure 3.

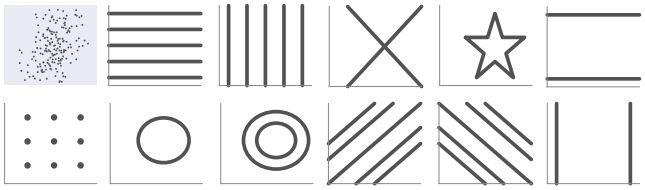

Figure 3. The initial data set (top-left), and line segment collections used for directing the output towards specific shapes. The results are seen in Figure 1.

With this example dataset, the algorithm ran for 200,000 iterations to achieve the final results. On a laptop computer this process took ~10 minutes. Figure 4 shows the progression of one of the datasets towards the target shape.

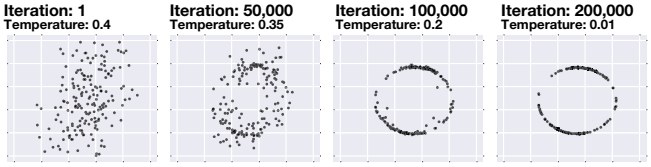

Figure 4. Progression of the algorithm towards a target shape over the course of the cooling schedule.

Example 2: Alternate Statistical Measures

One benefit of our approach over previous methods is that the iterative process is agnostic to the particular statistical properties which remain constant between the datasets. In this example (Figure 5) the datasets are derived from the same initial dataset as in Example 1, but rather than being equal on the parametric properties, the datasets are equal in the non-parametric measures of *x/y median*, *x/y interquartile range (IQR)*, and *Spearman's rank correlation coefficient*.

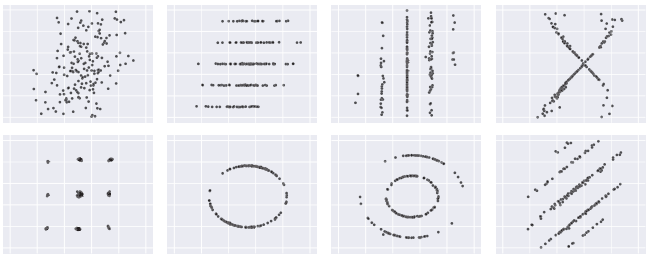

Figure 5. Example datasets are equal in the non-parametric statistics of *x/y median* (53.73, 46.21), *x/y IQR* (19.17, 37.92), and *Spearman's rank correlation coefficient* (+0.31).

Example 3: Specific Initial Dataset

The previous two examples used a rather "generic" dataset of a slightly positively correlated point cloud as the starting point of the optimization. Alternately, it is possible to begin with a very specific dataset to seed the optimization.

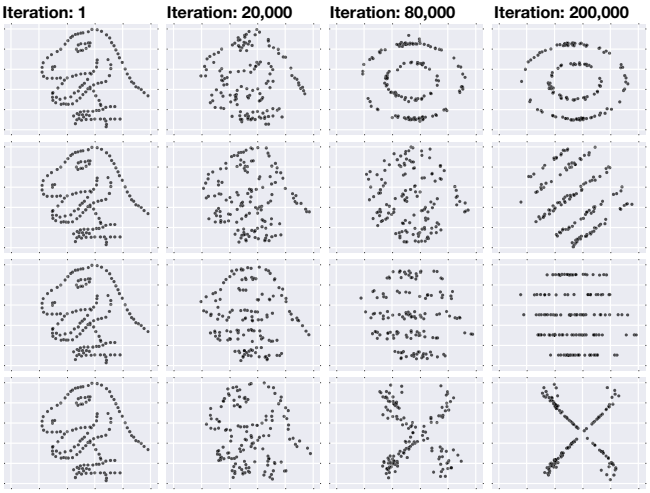

Figure 6. Creating a collection of datasets based on the "dinosaurus" dataset. Each dataset has the same summary statistics to two decimal places: ($\bar{x}$ =54.26, $\bar{y}$ = 47.83, sd_x = 16.76, sd_y = 26.93, Pearson's r = -0.06).

Alberto Cairo produced a dataset called the "Datasaurus" [4]. Like Anscombe's Quartet, this serves as a reminder to the importance of visualizing your data, since, although the dataset produces "normal" summary statistics, the resulting plot is a picture of a dinosaur. In this example we use the "datasaurus" as the initial dataset, and create other datasets with the same summary statistics (Figure 6).

Example 4: Simpson's Paradox

Another instrument for demonstrating the importance of visualizing your data is Simpson's Paradox [3, 10]. This paradox occurs with data sets where a trend appears when looking at individual groups in the data, but disappears or reverses when the groups are combined.

To create a dataset exhibiting Simpson's Paradox, we start with a strongly positively correlated dataset (Figure 7A), and then perturb and direct that dataset towards a series of

negatively sloping lines (Figure 7B). The resulting dataset (Figure 7C) has the same positive correlation as the initial dataset when looked at as a whole, while the individual groups each have a strong negative correlation.

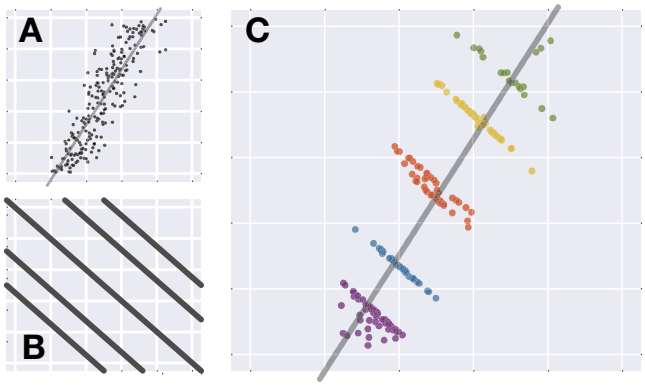

Figure 7. Demonstration of Simpson's Paradox. Both datasets (A and C) have the same overall Pearson's correlation of +0.81, however after coercing the data towards the pattern of sloping lines (B), each subset of data in (C) has an individually negative correlation.

Example 5: Cloned Dataset with Similar Appearance

As discussed by Govindaraju and Haslett [8] another use for datasets with the same statistical properties is the creation of "cloned" datasets to anonymize sensitive data [6]. In this case, it is important that individual data points are changed while the overall structure of the data remains similar. This can be accomplished by performing a Kolmogorov-Smirnov test within the IsErrorOK function for both x and y. By only accepting solutions where both the x and y K-S statistic is <0.05 we ensure that the result will have a similar shape to the original (Figure 8). This approach has the benefit of maintaining the x/y means and correlation as accomplished in previous work [8], and additionally the x/y standard deviations as well. This could also be useful for "graphical inference" [12] to create a collection of variant plots following the same null hypothesis.

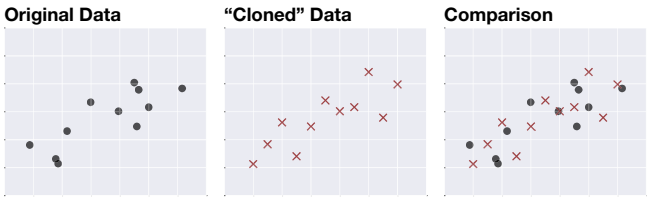

Figure 8. Example of creating a "mirror" dataset as in [8].

Example 6: 1D Boxplots

To demonstrate the applicability of our approach to non 2D-scatterplot data, this example uses a 1D distribution of data as represented by a boxplot. The most common variety of boxplot, the "Tukey Boxplot", presents the 1st quartile, median, and 3rd quartile values on the "box", with the "whiskers" showing the location of the furthest datapoints within 1.5 interquartile ranges (IQR) from the 1st and 3rd quartiles. Starting with the data in a normal distribution (Figure 9A) and perturbing the data to the left (B), right (C), edges (D, E), and arbitrary points along the range (F) while ensuring that the boxplot statistics remain constant produces the results shown in Figure 9.

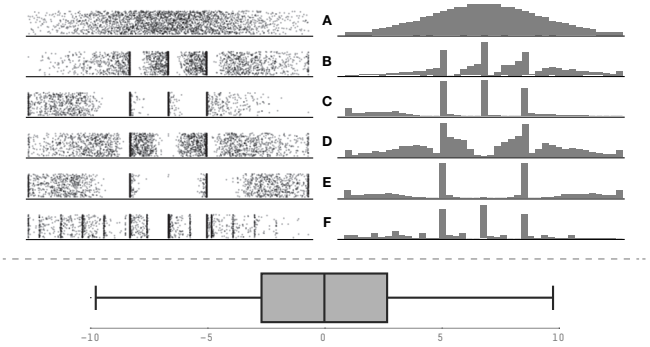

Figure 9. Six data distributions, each with the same 1st quartile, median, and 3rd quartile values, as well as equal locations for points 1.5 IQR from the 1st and 3rd quartiles. Each dataset produces an identical boxplot.

LIMITATIONS AND FUTURE WORK

When the source dataset and the target shape are vastly different, the produced output might not be desirable. An example is show Figure 10, where the data set from Figure 7A is coerced into a star (Figure 10). This problem can be mitigated by coercing the data towards "simpler" patterns with more coverage of the coordinate space – such as lines spanning the grid, or pre-scaling and positioning the target shape to better align with the initial dataset.

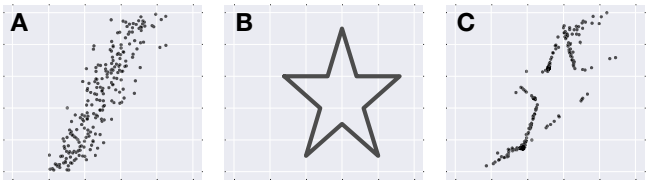

Figure 10. Undesirable outcome (C) when coercing a strongly positively correlated dataset (A) into a star (B).

The currently implemented fitness function looks only at the position of individual points in relation to the target shape, which can result in "clumping" of data points and sparse areas on the target shape. A future improvement could consider an additional goal to "separate" the points to encourage better coverage of the target shape in the output.

The parameters chosen for the algorithm (95% success rate, quadratic cooling scheme, start/end temperatures, etc.) were found to work well, but should not be considered "optimal". Such optimization is left as future work.

The code and datasets presented in this work are available at www.autodeskresearch.com/publications/samestats.

CONCLUSION

We presented a technique for creating visually dissimilar datasets which are equal over a range of statistical properties. The outputs from our method can be used to demonstrate the importance of visualizing your data, and may serve as a starting point for new data anonymization techniques.

REFERENCES

1. Anscombe, F.J. (1973). Graphs in Statistical Analysis. *The American Statistician 27*, 1, 17–21.
2. Bach, B., Spritzer, A., Lutton, E., and Fekete, J.-D. (2012). Interactive Random Graph Generation with Evolutionary Algorithms. *SpringerLink*, 541–552.
3. Blyth, C.R. (1972). On Simpson's Paradox and the Sure-Thing Principle. *Journal of the American Statistical Association 67*, 338, 364–366.
4. Cairo, A. Download the Datasaurus: Never trust summary statistics alone; always visualize your data. http://www.thefunctionalart.com/2016/08/download-datasaurus-never-trust-summary.html.
5. Chatterjee, S. and Firat, A. (2007). Generating Data with Identical Statistics but Dissimilar Graphics. *The American Statistician 61*, 3, 248–254.
6. Fung, B.C.M., Wang, K., Chen, R., and Yu, P.S. (2010). Privacy-preserving Data Publishing: A Survey of Recent Developments. *ACM Comput. Surv. 42*, 4, 14:1–14:53.
7. Govindaraju, K. and Haslett, S.J. (2008). Illustration of regression towards the means. *International Journal of Mathematical Education in Science and Technology 39*, 4, 544–550.
8. Haslett, S.J. and Govindaraju, K. (2009). Cloning Data: Generating Datasets with Exactly the Same Multiple Linear Regression Fit. *Australian & New Zealand Journal of Statistics 51*, 4, 499–503.
9. Hwang, C.-R. Simulated annealing: Theory and applications. *Acta Applicandae Mathematica 12*, 1, 108–111.
10. Simpson, E.H. (1951). The Interpretation of Interaction in Contingency Tables. *Journal of the Royal Statistical Society. Series B (Methodological) 13*, 2, 238–241.
11. Stefanski, L.A. (2007). Residual (Sur)Realism. *The American Statistician*, .
12. Wickham, H., Cook, D., Hofmann, H., and Buja, A. (2010). Graphical inference for infovis. *IEEE Transactions on Visualization and Computer Graphics 16*, 6, 973–979.