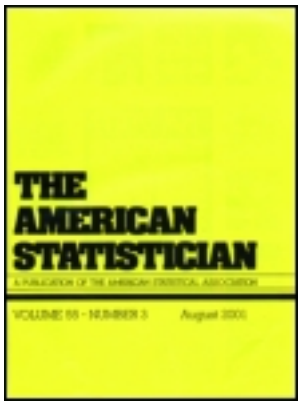


The American Statistician
Publication details, including instructions for authors and subscription information:
http://amstat.tandfonline.com/loi/utas20

Generating Data with Identical Statistics but Dissimilar Graphics

Sangit Chatterjee[a] & Aykut Firat[a]

[a] Sangit Chatterjee is Professor, and Aykut Firat is Assistant Professor, College of Business Administration, Northeastern University, Boston, MA 02115 (E-mail addresses: s and . We greatly appreciate the editor's and an anonymous associate editor's comments that greatly improved the article.

Published online: 01 Jan 2012.

To cite this article: Sangit Chatterjee & Aykut Firat (2007) Generating Data with Identical Statistics but Dissimilar Graphics, The American Statistician, 61:3, 248-254, DOI: 10.1198/000313007X220057

To link to this article: http://dx.doi.org/10.1198/000313007X220057

Generating Data with Identical Statistics but Dissimilar Graphics: A Follow up to the Anscombe Dataset

Sangit CHATTERJEE and Aykut FIRAT

The Anscombe dataset is popular for teaching the importance of graphics in data analysis. It consists of four datasets that have identical summary statistics (e.g., mean, standard deviation, and correlation) but dissimilar data graphics (scatterplots). In this article, we provide a general procedure to generate datasets with identical summary statistics but dissimilar graphics by using a genetic algorithm based approach.

KEY WORDS: Genetic algorithms; Ortho-normalization; Nonlinear optimization.

1. INTRODUCTION

To demonstrate the usefulness of graphics in statistical analysis, Anscombe (1973) produced four datasets each with an independent variable x and a dependent variable y that had the same summary statistics (such as mean, standard deviation, and correlation), but produced completely different scatterplots. The Anscombe dataset is reproduced in Table 1 and the scatterplots of the four datasets are given in Figure 1. The dataset has now become famous as the Anscombe data, and is often used in introductory statistics classes as an example to illustrate the usefulness of graphics: an apt illustration of the well-known wisdom that a scatterplot can often reveal patterns that may be hidden by summary statistics. It is not known, however, how Anscombe came up with his datasets. In this article, we provide a general procedure to generate datasets with identical summary statistics but dissimilar graphics by using a genetic-algorithm-based approach.

2. PROBLEM DESCRIPTION

Consider a given data matrix $\mathbf{X}^*$ consisting of two data vectors of size n: the independent variable $\mathbf{x}^*$, and the dependent variable $\mathbf{y}^*$. (Though we present the case for two data vectors, our methodology is generally applicable.) Let $\overline{x}^*$, $\overline{y}^*$ be the mean value, and s_x^*, s_y^* be the standard deviation of vectors, and r^* be the correlation coefficient between vectors $\mathbf{x}^*$ and $\mathbf{y}^*$. Let $\mathbf{X}$ be another data matrix containing two data vectors of size n : $\mathbf{x}$, $\mathbf{y}$. The problem is to find at least one $\mathbf{X}$ that has identical summary statistics as $\mathbf{X}^*$. At the same time, scatterplots of $\mathbf{x}$, $\mathbf{y}$ should be dissimilar to those of $\mathbf{x}^*$, $\mathbf{y}^*$ according to a function $g(\mathbf{X}, \mathbf{X}^*)$, which quantifies the graphical difference between the scatterplots of $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{x}^*$, $\mathbf{y}^*$. This problem can be formulated as a mathematical program as follows:

$$\text{maximize} \quad g(\mathbf{X}, \mathbf{X}^*)$$
$$\text{s.t.} \left|\overline{x}^* - \overline{x}\right| + \left|\overline{y}^* - \overline{y}\right| + \left|s_x^* - s_x\right| + \left|s_y^* - s_y\right| + \left|r^* - r\right| = 0.$$

In the above formulation, the objective function to be maximized is the graphical dissimilarity between $\mathbf{X}$ and $\mathbf{X}^*$, and the constraint ensures that the summary statistics will be identical. In order to measure the graphical dissimilarity between two scatterplots, we considered the absolute value differences between the following quantities of $\mathbf{X}$ and $\mathbf{X}^*$:

a. ordered data values $\left(g = \sum \left|x_{(i)} - x_{(i)}^*\right| + \left|y_{(i)} - y_{(i)}^*\right|\right)$.

b. Kolmogorov–Smirnov test statistics over an interpolated grid of y values;

$$\left(g = \max\left(\left|F(a) - F^*(a)\right|\right),\right.$$

where $F(a)$ is the proportion of y_i values less than or equal to a and $F^*(a)$ is the proportion of y_i^* values less than or equal to a, where a corresponds to all possible values of y_i and y_i^*.

c. the quadratic coefficients of the regression fit ($g = |b_2 - b_2^*|$, where $y_i = b_0 + b_1 x_i + b_2 x_i^2 + e_i$ and $y_i^* = b_0^* + b_1^* x_i^* + b_2^* x_i^{*2} + e_i^*$.

d. Breusch-Pagan (1979) Lagrange multiplier (LM) statistics as a measure of heteroscedasticity; ($g = |\text{LM} - \text{LM}^*|$).

e. standardized skewness

$$\left(g_{\text{skewness}} = \left|\frac{\sum (y_i - \overline{y})^3}{s_y^3} - \frac{\sum \left(y_i^* - \overline{y}^*\right)^3}{s_{y^*}^3}\right|\right).$$

f. standardized kurtosis

$$\left(g_{\text{kurtosis}} = \left|\frac{\sum (y_i - \overline{y})^4}{s_y^4} - \frac{\sum \left(y_i^* - \overline{y}^*\right)^4}{s_{y^*}^4}\right|\right).$$

g. maximum of the Cook's D statistic (Cook 1977) ($g = |\max(d_i) - \max(d_i^*)|$, where d_i is Cook's D statistic for observation i).

We also experimented with various combinations of the above items such as the multiplicative combination of standardized skewness and kurtosis measures ($g = g_{\text{skewness}} * g_{\text{kurtosis}}$). We report on such experiments in the results section.

Sangit Chatterjee is Professor, and Aykut Firat is Assistant Professor, College of Business Administration, Northeastern University, Boston, MA 02115 (E-mail addresses: *s.chatterjee@neu.edu* and *a.firat@neu.edu*). We greatly appreciate the editor's and an anonymous associate editor's comments that greatly improved the article.

  DOI: 10.1198/000313007X220057

Table 1. Anscombe's Original Dataset. All four datasets have identical summary statistics: means ($\overline{x} = 9.0, \overline{y} = 7.5$), regression coefficients ($b_0 = 3.0, b_1 = 0.5$), standard deviations ($s_x = 3.32, s_y = 2.03$), correlation coefficients, etc.

Dataset 1		Dataset 2		Dataset 3		Dataset 4	
x	y	x	y	x	y	x	y
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.76	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.10	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.10	4	5.39	8	5.56
12	10.84	12	9.13	12	8.15	8	7.91
7	4.82	7	7.26	7	6.42	8	6.89
5	5.68	5	4.74	5	5.73	19	12.5

3. METHODOLOGY

We propose a genetic algorithm (GA) (Goldberg 1989) based solution to our problem. GAs are often used for problems that are difficult to solve with traditional optimization techniques; therefore a good choice for our problem that has a discontinuous, and nonlinear objective function with undefined derivatives. See also Chatterjee, Laudoto, and Lynch (1996) for applications of genetic algorithms to problems of statistical estimation.

In a GA an individual solution is called a gene, and is typically represented as a vector of real numbers, bits (0/1), or character strings. In the beginning, an initial population of genes is created. The GA, then, repeatedly modifies this population of individual solutions over many generations. At each generation, children genes are produced from randomly selected parents (crossover), or from randomly modified individual genes (mutation). In accord with the Darwinian principle of "natural selection," genes with high "fitness values" have a higher chance of survival in the next generations. Over successive generations, the population evolves toward an optimal solution. We now explain the details of this algorithm applied to our problem.

3.1 Representation

We conceptualize a gene as a matrix of size $n \times 2$ having real values. For example, when $n = 11$ (the size of Anscombe's data), an example gene $\mathbf{X}$ would be as follows (note that the transpose of $\mathbf{X}$ is shown below):

$$\mathbf{X}' = \begin{bmatrix} -0.43 & 1.66 & 0.12 & 0.28 & -1.14 & 1.19 & 1.18 & -0.03 & 0.32 & 0.17 & -0.18 \\ 0.72 & -0.58 & 2.18 & -0.13 & 0.11 & 1.06 & 0.05 & -0.09 & -0.83 & 0.29 & -1.33 \end{bmatrix}.$$

3.2 Initial Population Creation

Individual solutions in our population should satisfy the constraint in our mathematical formulation in order to be a feasible solution. Given an original data matrix $\mathbf{X}^*$ of size $n \times 2$, we accomplish this through orthonormalization and a transformation as outlined in the following for a single gene (Matlab statements for a specific case ($n = 11$) are also given for each step).

(i) Generate a matrix $\mathbf{X}$ of size $n \times 2$ with randomly generated

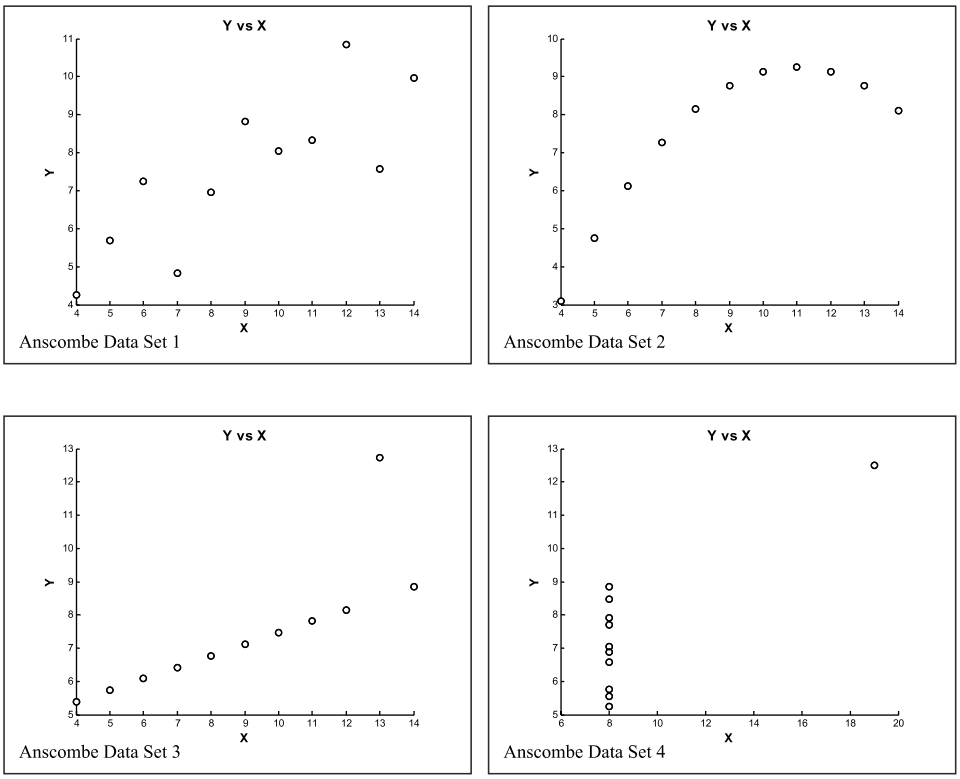


Figure 1. Scatterplots of Anscombe's data. Scatterplots of the Anscombe datasets reveal different data graphics.

data from a standard normal distribution. Distrubutions other than the standard normal can also be used in this step.

```
Matlab> X = randn(11,2)
```

(ii) Set the mean values of X's columns to zero using $\mathbf{X} = \mathbf{X} - \mathbf{e}_{n\times 1} * \overline{\mathbf{X}}$, where $\mathbf{e}_{n\times 1}$ is an n-element column vector of ones. This step is needed to make sure that after ortho-normalization the standard deviation of the columns will be equal to the unit vector norm.

```
Matlab> X = X - ones(11,1)*mean(X)
```

(iii) Ortho-normalize the columns of $\mathbf{X}$. For this, we use the Gram-Schmidt process (Arfken 1985), by taking a nonorthogonal set of linearly independent vectors $\mathbf{x}$ and $\mathbf{y}$ constructing an orthogonal basis $\mathbf{e}_1$ and $\mathbf{e}_2$ as follows (in $\mathbf{R}^2$):

$$\mathbf{u}_1 = \mathbf{x}, \mathbf{u}_2 = \mathbf{y} - \text{proj}_{\mathbf{u}_1}\mathbf{y},$$

where

$$\text{proj}_{\mathbf{u}}\mathbf{v} = \frac{\langle \mathbf{v}, \mathbf{u}\rangle}{\langle \mathbf{u}, \mathbf{u}\rangle}\mathbf{u},$$

and $\langle \mathbf{v}_1, \mathbf{v}_2\rangle$ represents the inner product. Then

$$\mathbf{e}_1 = \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}, \quad \text{and} \quad \mathbf{e}_2 = \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|},$$

and

$$\mathbf{X}_{\text{ortho-normalized}} = [\mathbf{e}_1, \mathbf{e}_2].$$

```
Matlab> X = grams(X);
```

where grams is a custom function that performs Gram-Schmidt ortho-normalization.

(iv) Transform $\mathbf{X}$ with the following equation:

$$\mathbf{X} = \sqrt{n-1} * \mathbf{X}_{\text{ortho-normalized}} * \text{cov}(\mathbf{X}^*)^{1/2} + \mathbf{e}_{n\times 1} * \overline{\mathbf{X}}^*,$$

where $\text{cov}(\mathbf{X})^*$ is the covariance matrix of $\mathbf{X}^*$, $\overline{\mathbf{X}}^* = [\overline{x}_1^*, \overline{x}_2^*]$. $\sqrt{n-1}$ is needed since we are using the *sample* standard deviation in covariance calculations.

```
Matlab> X = sqrt(10) *  X  * sqrtm(cov(Xo))
   + ones(11,1)*mean(Xo);
```

where Xo is the original data matrix.

With these steps, we can create a gene that satisfies the constraint of our mathematical formulation, that is, the aforementioned summary statistics of our new gene are identical to our original gene. We independently generated 1,000 such random genes to create our initial population P.

3.3 Fitness Values

At each generation, a fitness value needs to be calculated for each population member. For our problem a gene's fitness is proportional to its graphical difference from the given data matrix $\mathbf{X}^*$. We used the graphical difference functions mentioned in the problem description section in different runs of experiments.

3.4 Creating The Next Generation

3.4.1 Selection

Once the fitness values are calculated, parents are selected for the next generation based on their fitness. We use the "stochastic uniform" selection procedure, which is the default method in Matlab Genetic Algorithm Toolbox (Matlab 2006). This selection algorithm first lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled fitness value. Then the algorithm moves along the line in steps of equal size. At each step, the algorithm allocates a parent from the section it lands on.

3.4.2 New Children

Three types of children are created for the next generation:

(i) *Elite Children*—Individuals in the current generation with the top two best fitness values are called elite children, and automatically survive in the next generation.

(ii) *Crossover Children*—These are children obtained by combining two parent genes. A child is obtained by splitting two selected parent genes at a random point, and combining the head of one with the tail of the other and vice versa as illustrated in Figure 2. Eighty percent of the individuals in the new generation are created this way.

(iii) *Mutation Children*—Mutation children make up the remaining members of the new generation. A parent gene is modified by adding a random number, or mutation, chosen from a

$$\mathbf{X}_1' = \begin{bmatrix} 6.09 & 7.12 & 14.73 & 12.19 & 6.25 & 7.27 & 6.25 & 5.92 & 8.06 & 13.34 & 11.76 \\ 5.86 & 1.84 & 4.03 & 5.15 & 1.50 & 0.12 & 6.52 & 4.45 & 2.48 & 7.06 & 6.00 \end{bmatrix}$$

$$\mathbf{X}_2' = \begin{bmatrix} 14.34 & 12.53 & 9.37 & 12.67 & 11.09 & 7.46 & 6.95 & 6.00 & 3.40 & 7.82 & 7.36 \\ 5.21 & 7.60 & 2.94 & 5.28 & 5.22 & 6.13 & 1.27 & 3.61 & 5.84 & 1.26 & 0.64 \end{bmatrix}$$

$$\mathbf{X}_c' = \begin{bmatrix} 6.09 & 7.12 & 14.73 & 12.19 & 6.25 & 7.46 & 6.95 & 6.00 & 3.40 & 7.82 & 7.36 \\ 5.86 & 1.84 & 4.03 & 5.15 & 1.50 & 6.13 & 1.27 & 3.61 & 5.84 & 1.26 & 0.64 \end{bmatrix}$$

Figure 2. Crossover operation.

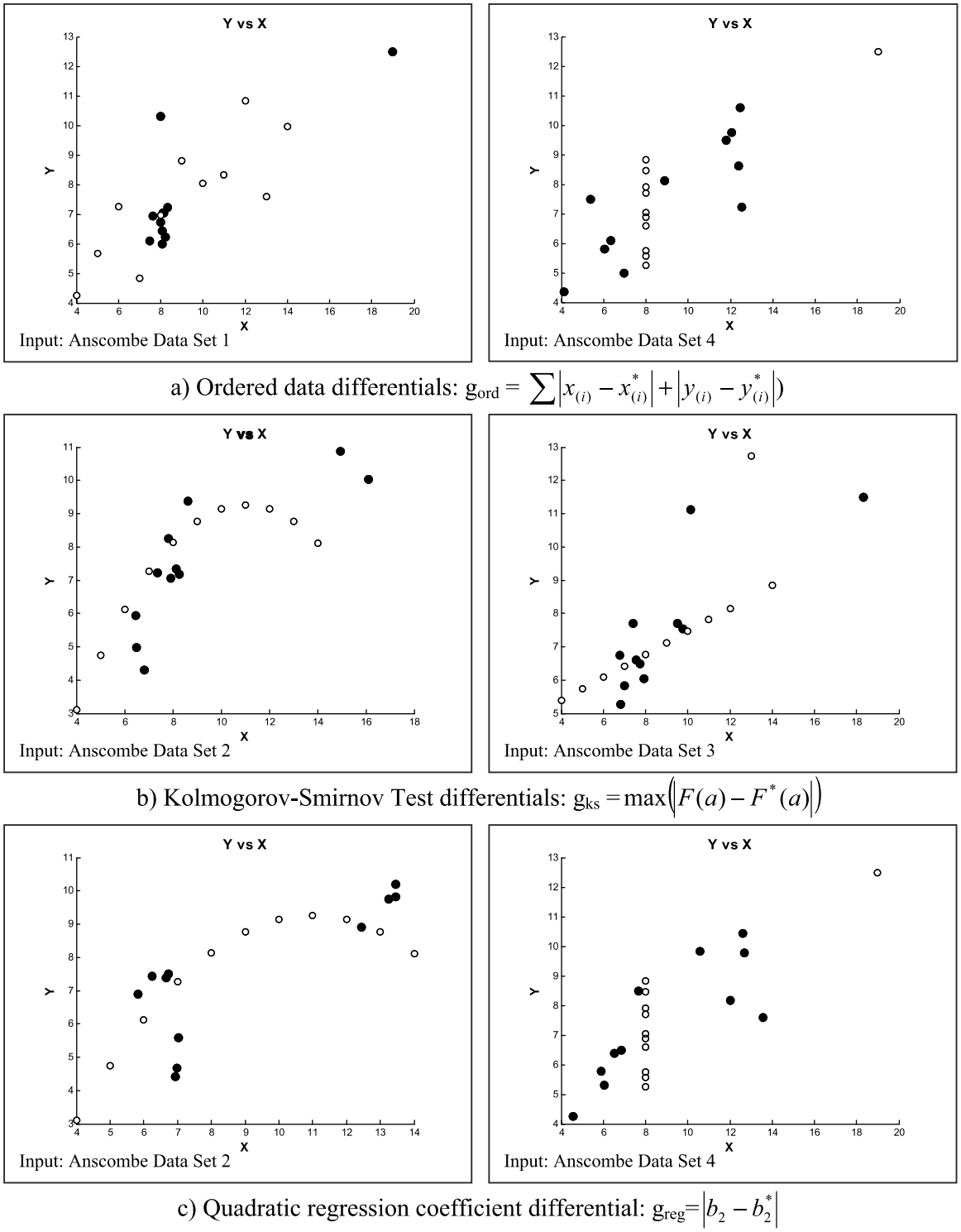
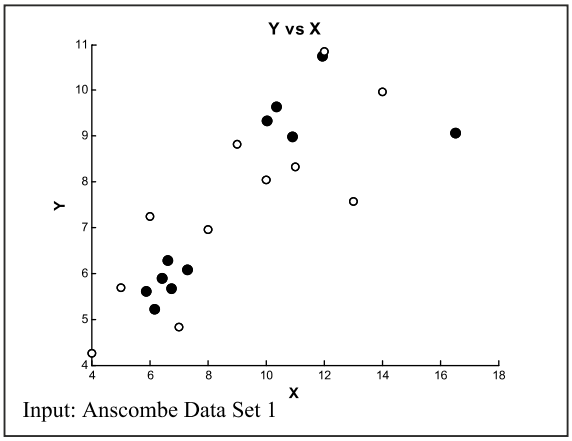
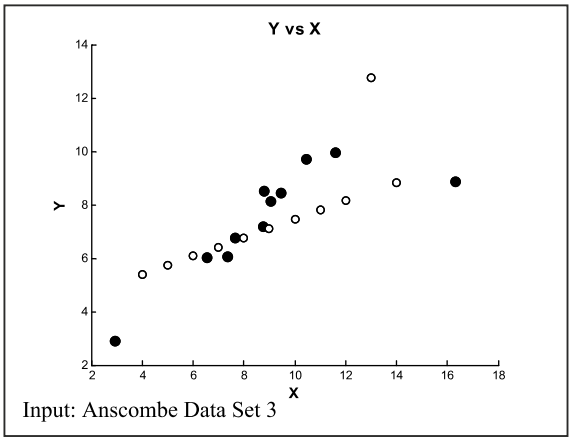
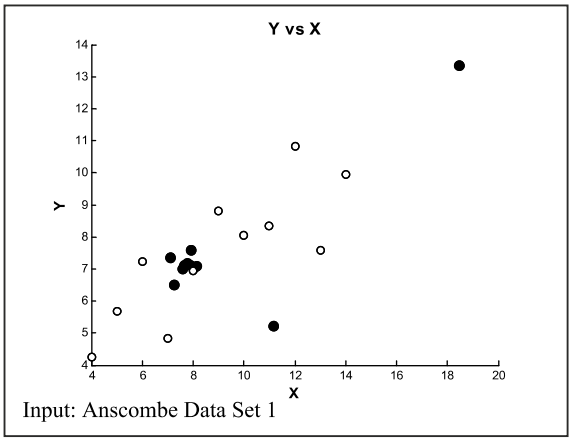


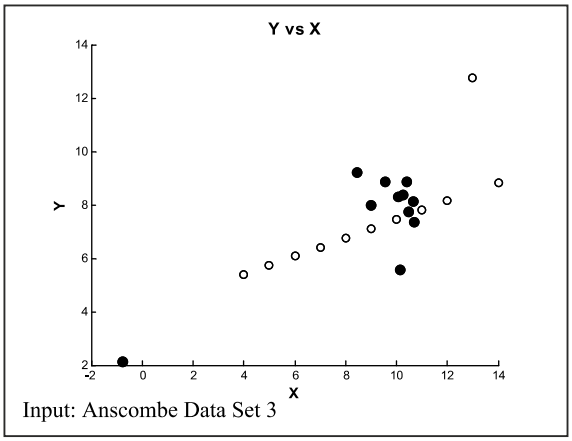
Figure 3. Scatterplots of *Y* versus *X* with different graphical dissimilarity functions (a to c). The solid circles correspond to output, and the empty circles correspond to input datasets.



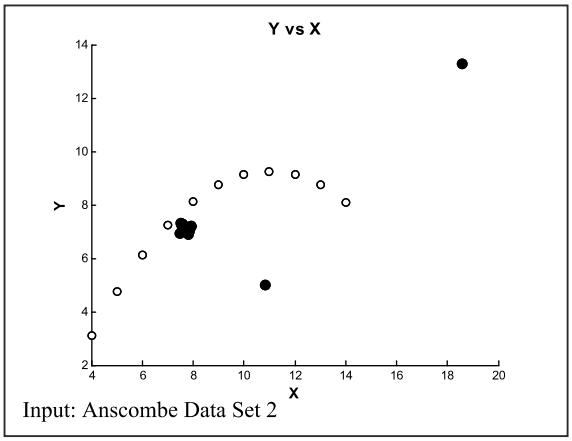


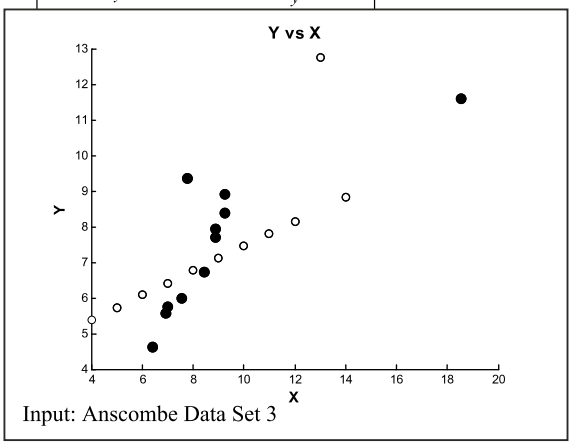
d) Breusch-Pagan heteroscedasticity test differential: $g_{LM}=\left|LM - LM^{*}\right|$





e) Skewness differential: $g_{skewness} = \left| \frac{\sum (y_i - \bar{y})^3}{s_y^3} - \frac{\sum (y_i^* - \bar{y}^*)^3}{s_{y^*}^3} \right|$





f) Kurtosis differential: $g_{kurtosis} = \left| \frac{\sum (y_i - \bar{y})^4}{s_y^4} - \frac{\sum (y_i^* - \bar{y}^*)^4}{s_{y^*}^4} \right|$

Figure 4. Scatterplots of *Y* versus *X* with different graphical dissimilarity functions (d to f). The solid circles correspond to output, and the empty circles correspond to input datasets.

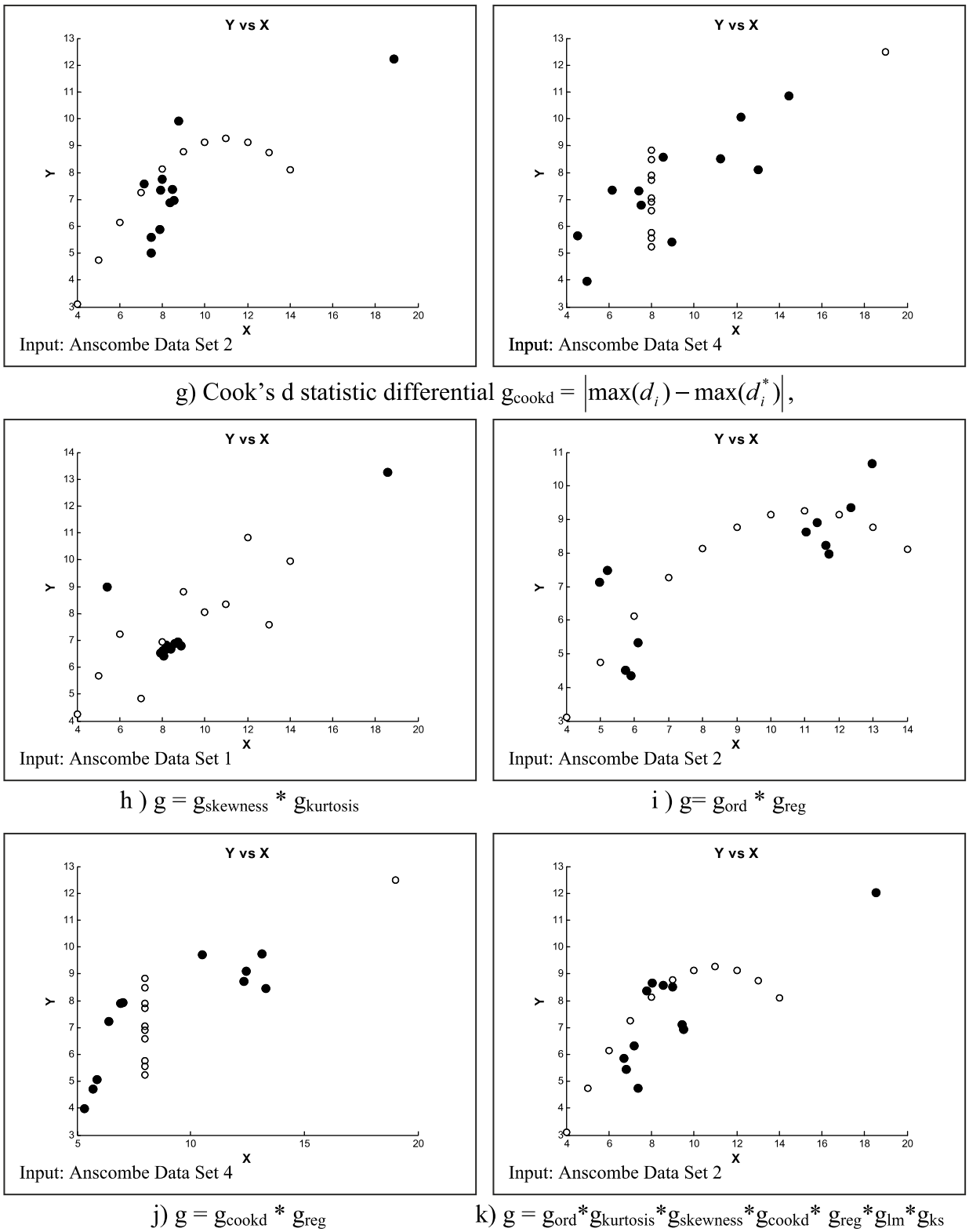


Figure 5. Scatterplots of Y versus X with different graphical dissimilarity functions (g to k). The solid circles correspond to output, and the empty circles correspond to input datasets.

Gaussian distribution (with mean 0, and variance 0.5), to each entry of the parent vector. Mutation amount decreases at each generation according to the following formula supplied by Matlab:

$$\text{var}_k = \text{var}_{k-1}\left(1 - 0.75 * \frac{k}{\text{Generations}}\right),$$

where var_k is the variance of the Gaussian distribution at the current generation k, and Generations is the number of total generations.

3.4.3 Ortho-normalization and Transformation

After the children for the new generation are created, they are ortho-normalized and transformed so that they also satisfy the constraint like the initial population. This is accomplished as explained in Section 3.2, Steps ii–iv.

3.5 Final Generation

The algorithm runs for 2,500 generations or until there is no improvement in the objective function during an interval of 20 seconds. Within the final generation, genes with large fitness values are obvious solutions to our problem.

4. RESULTS

In our experiments we used Anscombe's four initial datasets as input data to our algorithm. The scatterplots of our generated datasets are shown in Figures 3–5. Our experiments reveal that ordered data, kurtosis, skewness, and maximum of Cook's D statistic differentials consistently performed well and produced dissimilar graphics. Kolmogorov–Smirnov test, quadratic regression coefficient, and Breusch-Pagan heteroscedasticity test differentials did not consistently produce dissimilar scatterplots. These measures, however, were still useful when combined with other measures. For example, as shown in Figure 5(j), the combination of quadratic regression coefficient with the maximum of Cook's D statistic produced a very similar scatterplot to Anscombe's second dataset when the input was his fourth dataset. In Figure 5(g), we also see a dataset similar to Anscombe's first dataset when the input was fourth dataset, and in Figure 1(a) we see a dataset similar to Anscombe's fourth dataset when the input was his first dataset. These results indicate that datasets with characteristics similar to the Anscombe's datasets could have been created using the procedure outlined in this article.

5. CONCLUSION

Anscombe data retain their well-earned place in statistics and serve as a starting point for a more general approach to generate other datasets with identical summary statistics but very different scatter plots. With this work, we provided a general procedure to generate similar data sets with an arbitrary number of independent variables that can be used for instructional and experimental purposes.

[Received October 2005. Revised November 2006.]

REFERENCES

Anscombe, F. J. (1973), "Graphs in Statistical Analysis," *The American Statistician*, 27, 17–21.

Arfken, G. (1985), "Gram-Schmidt Orthogonalization," *Mathematical Methods for Physicists*, 3, 516–520.

Breusch, T. S., and Pagan, A. R. (1979), "A Simple Test for Heteroscedasticity and Random Coefficient Variation," *Econometrica*, 47, 1287–1294.

Chatterjee, S., Laudato, M., and Lynch, L. A. (1996), "Genetic Algorithms and their Statistical Applications: An Introduction," *Computational Statistics and Data Analysis*, 22, 633–651.

Cook, R. D. (1977), "Detection of Influential Observation in Linear Regression," *Technometrics*, 19, 15–18.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA: Addision-Wesley.

Matlab (2006), *Matlab Reference Guide*, Natick, MA: Mathworks Inc.