# Checking and Correcting Categorical Variable Trend Models

Jared L. Deutsch and Clayton V. Deutsch

*Categorical variable trend models are frequently used in the practice of geostatistics. They are often built using geological information that would otherwise be difficult to incorporate into a geostatistical model. Trend models may or may not be fair depending on how the trend model was built. A post-processing program, `posttrend`, is introduced which enforces fairness in the trend model by modifying it to respect available data. A second program, `plotfair`, is introduced which cross plots the actual observed proportions from the data against the proportions predicted by the trend model. This provides the modeler with a visualization tool to check if the trend model is fair.*

## Introduction

One of the most common methods for incorporating geological knowledge about an area is with a trend model. The trend model may be constructed by hand contouring, kriging, combining lower order vertical and areal trends or another methods. Regardless of how the trend is constructed, a good trend should be fair, informative and not over-fit. This note addresses the issue of categorical variable trend model fairness with the introduction of two GSLIB compatible programs that allow the user to check that the trend model respects the available data and corrects the trend model if needed. The first program, `plotfair`, provides a visual aid for the modeler to assist them in evaluating the fairness of their trend model. The second program, `posttrend`, adjusts the trend model in an iterative manner so that the probabilities align with the data. A consequence of updating the trend model probabilities independently is that the updated probabilities must be rescaled to fall in the range of [0,1] and sum to 1. A new method for rescaling these probabilities is introduced.

## Background

Consider a categorical variable, $k$, which can be described by a trend model $p_k(\mathbf{u})$ where $k$=1,..,$K$ for all $\mathbf{u}$ in the domain. $p_k(\mathbf{u})$ is interpreted as the expected value of category $k$ prevailing at location $\mathbf{u}$. If the trend model is fair, then Equation 1 will be satisfied for all $p$ and for all $k$ (McLennan, 2007 and Deutsch and Journel, 1998).

$$\mathrm{E}\{i(\mathbf{u};k) \mid p_k(\mathbf{u}) = p\} = p \tag{1}$$

This fairness condition is one of the three principle qualities of a good trend model; a good trend model is fair, informative and not over-fit. It is assumed that the geostatistician building the trend model will satisfy the latter criteria. This note is concerned only with enforcing the principle criterion.

## Plotting Fairness

Given data for $\underline{\mathbf{u}}$ (which is a subset of the domain $\mathbf{u}$) from well logs or other methods, the trend model probabilities can be compared to the actual proportions. Because of the continuous nature of trend model probabilities, a binning approach is suggested. Trend model probabilities at each location in $\underline{\mathbf{u}}$ are binned into 10 classes ($j$=1,…,10) as $p_{k,j}(\underline{\mathbf{u}})$ = 0.05±0.05, 0.10±0.05,…,0.95±0.05. For each of these classes, the proportion, $p$, of data at the trend model locations with an indicator $k$ is determined. The actual proportion of data with an indicator $k$ can be cross plotted against the expected fraction from the trend model $p_k$ (Figure 1).

The points shown in Figure 1 should lie on the 45° line within tolerances if the trend model is completely fair. The tolerances depend on the number of data used for each point. The trend model proportions follow a multinomial distribution, so classical statistics can be used to determine a 99% confidence interval. The probability of observing an actual proportion $p$ based on $n$ data points given that the true trend model proportion is $p_k$ is given by the binomial probability distribution: Equation 2, (Montgomery and Runger, 2008).

$$\Pr(P = p) = \binom{n}{np} \frac{p_k^{np}}{n} \left(1 - p_k\right)^{n(1-p)} \tag{2}$$

This can be approximated using the Gaussian distribution with B($n$, $p_k$) ~ N($np_k$, $np_k(1-p_k)$). The cumulative probabilities (0.005 and 0.995) necessary for constructing a 99% confidence interval can be calculated using the binomial probability distribution or, where appropriate, the Gaussian approximation.

The plotting of the actual proportions against expected trend model proportions is implemented in the GSLIB compatible program `plotfair`. `Plotfair` uses the output table from its partner program (detailed in this note), `posttrend`. Details on `plotfair` and the associated parameter file are contained in Appendix 1. An example plot is shown in Figure 2.

**Correcting the Trend Model**

To correct the trend model, an iterative scheme which slightly modifies the trend model probabilities so that they align better with the actual observed proportions is proposed. To calculate the deviation of the trend model proportions from the actual proportions, trend model probabilities are binned using the same bins as for plotting (0.05±0.05, 0.10±0.05,…,0.95±0.05). Instead of calculating the absolute deviation, we calculate a corrected deviation. The corrected deviation of the trend model probabilities from the actual observed proportions is given by Equation 3. This deviation, *d*, is dampened by a function $\omega(n)$ to prevent over-fitting the trend model (Equation 4). A sample plot of *d* values is given in Figure 3.

$$d = \left( p_k - p \right) \bullet \omega(n) \tag{3}$$

With larger amounts of data, less damping is desired. Equation 4 is proposed for $\omega(n)$ as a damping function. The parameters *a* and *b* can be adjusted by the user. To avoid over-fitting the data too quickly, we suggest that *a* and *b* are set to 0.5 and 1, respectively, providing a significant damping effect.

$$\omega(n) = a \cdot \left( 1 - \frac{b}{\sqrt{n}} \right) \tag{4}$$

Because the trend model probabilities are continuous on the interval of [0,1], a function is required to interpolate between calculated *d* values. The *d* values are fit with a trimmed quadratic equation. The quadratic equation is bounded at the top by the largest calculated *d* value and on the bottom by the smallest calculated *d* value. A schematic fit is shown in Figure 4. Using the trimmed quadratic fit of the *d* values, the trend model is corrected (Equation 5) to give *K* conditional probabilities that do not necessarily satisfy order relations.

$$p_{k,i+1} = p_{k,i} + d(p_{k,i}) \tag{5}$$

The set of conditional probabilities for categories $k$=1,…,$K$ must satisfy order relations, that is;

$$p_k \geq 0 \ \forall \ k = 1,...,K \ \text{ and } \sum p_k = 1$$

The conventional method for enforcing the non-negativity constraint is by resetting all negative probabilities to zero:

$$p_k{}^{a1} = \left( p_k \right)_{\min=0} \ \ \forall \ k = 1,...,K$$

The sum to unity constraint is then met by dividing the non-negative probabilities by the sum:

$$p_k{}^{a2} = \frac{p_k{}^{a1}}{\sum p_k{}^{a1}}$$

The result of this two step correction satisfies order relations provided that $\sum p_k{}^{a1} > 0$. This conventional correction treats values less than zero and greater than one differently. One could imagine correcting the complimentary probabilities in a binary case, that is;

$$\left. \begin{aligned} p_k{}^{a1} &= \left( p_k \right)_{\min=0} \\ p_k{}^{a2} &= \frac{p_k{}^{a1}}{p_1{}^{a1} + p_2{}^{a1}} \end{aligned} \right\} k = 1,2$$

$$\left.\begin{aligned} 1-p_k^{\,b1} &= \left(1-p_k\right)_{\min=0} \\ 1-p_k^{\,b2} &= \frac{1-p_k^{\,b1}}{\left(1-p_1^{\,b1}\right)+\left(1-p_2^{\,b1}\right)} \end{aligned}\right\} k=1,2$$

The two alternative corrections are equally valid, but lead to different results. Consider the following numerical example:

$$p_1 = -0.1\ ,\ p_2 = 0.8$$
$$p_1^{\,a2} = 0.0\ ,\ p_2^{\,a2} = 1.0$$
$$p_1^{\,b2} = 0.154\ ,\ p_2^{\,b2} = 0.846$$

The results for *a* arguably have $p_1^{a2}$ too low and $p_2^{a2}$ too high. The results for *b* are the reverse, that is, $p_1^{b2}$ too high (much higher than the initial -0.1) and $p_2^{b2}$ is too low (just a little higher than the initial). One approach to reconcile these two alternatives is to average them:

$$p_k^{\,c} = \frac{p_k^{\,a2} + p_k^{\,b2}}{2}$$

The final corrected values will always satisfy order relations in the binary case. The correct values for this small example would be $p_1^c$ = 0.077 and $p_2^c$ = 0.923. The key idea of this approach is to correct the probability of the categories and the probability of *not* the categories. These are equally valid so an equal weighted average of the two is therefore reasonable.

The correction for *K*>2 is only slightly more complex. The complimentary probabilities to a particular category must consider all others, that is, *j=1,…K* (*j≠k*). Then, the binary probability of *k* being present and *k* not being presenty must satisfy order relations. The correction would be applied *k=1,…,K* times (Equations 6-8).

$$p_k^a = \frac{\left(p_k\right)_{\min=0}}{\left(p_k\right)_{\min=0} + \left(\sum_{j,j\neq k} p_j\right)_{\min=0}} \tag{6}$$

$$p_k^b = 1 - \frac{\left(1-p_k\right)_{\min=0}}{\left(1-p_k\right)_{\min=0} + \left(\sum_{j,j\neq k} \left(1-p_j\right)\right)_{\min=0}} \tag{7}$$

$$p_k^c = \frac{p_k^a + p_k^b}{2} \tag{8}$$

The results $p_k^c$ are each correct considering *k* and not *k* in a series of *K* binary evaluations; however they need not sum to unity. A final restandardization may be required (Equation 9).

$$p_k^* = \frac{p_k^c}{\sum_j p_j^c} \tag{9}$$

To illustrate this correction for a ternary case, let's consider $p_1$ =-0.1, $p_2$ =0.6 and $p_3$ =0.3. The conventional correction would lead to $p_1^c$ =0.0, $p_2^c$ =0.667 and $p_3^c$ =0.333. As before, we would expect the corrected $p_1$ to be greater than 0.0 since the sum of all probabilities is less than one. Correcting $p_1$ with Equations 6-9 leads to $p_1^c$ =0.250. Similarly, $p_2^c$ =0.784 and $p_3^c$ =0.528. Restandarizations leads to final values of $p_1^*$ =0.160, $p_2^*$ =0.502 and $p_3^*$ =0.338. These appear more reasonable and do not treat probabilities less than zero in an unfair manner.

Of course, corrected probability values could be zero; whenever the sum of original uncorrected probabilities is greater than one, then a negative probability will always be corrected to zero. This is reasonable. This is one method, similar to a naive Bayes approach or permanence of ratios (Journel, 2002 and Ortiz, 2003), of avoiding order relation deviations; however other methods have been explored by Ortiz (2001, 2002).

The process of correcting the trend model using Equations 3-5 is iterated a set number of times which is dictated by balancing fairness in the trend model with the other criteria that the trend model be informative and not over-fit. Due to the subjective nature of these criteria, the number of times that this process is iterated can be changed, however 3-5 times is suggested. This entire process is implemented in the GSLIB compatible program *posttrend* (see Appendix 1). After correcting the trend model, a fairness table which gives the probability values for *plotfair* is generated for assessing the corrected and original trend models.

**Case Studies**

The test case for this study was built in an unconditional simulation using the CCG program `BlockSIS` (Deutsch, 2005). An area, shown in Figure 5, measuring 5000m x 5000m x 50m was unconditionally simulated using 100m x 100m x 1m grid cells. The categorical variable of interest consists of 3 categories: 0, 1 and 2. Complete details on the parameters used to generate this test case are included in Appendix 2. The simulated global means for the three categories were 0.4074, 0.3023 and 0.2904 respectively. Data in the form of vertical wells was extracted for a regular grid of wells spaced 200 m apart.

Using this simulated data, three trend models were generated. The first trend model can be considered a correct trend model calculated using a moving window average with a radius of 15 cells. The second trend model was again built using a moving window average but biased so that the proportion of category 0 was overestimated and categories 1 and 2 were both underestimated. The third trend model was generated using a moving window average but from a different simulated set of data so is completely incorrect. Parameters used for calculating each trend model are detailed in Appendix 2.

**Correcting the Correct Model**

The correct trend model was post-processed using `posttrend` with three iterations. Other parameters including *a* and *b* were left as the default settings. Horizontal slices of trend model before and after running `posttrend` for category 1 are shown in Figure 6. It can be seen that the probability values were not dramatically changed. The trend model was sharpened as available data confirmed the trend.

Plots of the actual proportions against the predicted trend model proportions are shown in Figure 7, along with histograms of the trend model probabilities before and after post-processing. Although Figure 6 showed us that no major sweeping changes were made to the trend model, the overall fairness was significantly improved as shown in Figure 7. Points from the original trend model (shown as small dots) deviate significantly more from the 45° line than do points from the corrected trend model (shown as large dots). The spread of the probabilities was increased for all categories as a result of the post-processing as shown in the histograms. After post-processing there was a much more even distribution of probabilities. This effect is most evident for category 2.

**Correcting the Biased Model**

The biased trend model was constructed by increasing the trend proportions for category 0 and decreasing the trend proportions for categories 1 and 2. The biased trend model was processed with `posttrend`, again with three iterations and default parameters. Figure 8 shows slices of the trend model before and after processing were taken at the same location as Figure 6. Comparing the initial correct and biased trend models shows that the trend probabilities for the biased trend model are considerably lower. The post-processed trend model has probabilities that are very similar to the correct trend model indicating that the model is mostly corrected by `posttrend`.

Plots of the probabilities and histograms before and after processing are shown in Figure 9. The deviation of the initial trend model from the 45° line is immediately evident indicating that something is wrong with the model. The corrected model falls almost exactly on the 45° line for all categories. A spreading of probabilities similar to that for the correct trend model is observed as a result of the post-processing.

**Correcting the Incorrect Model**

The incorrect trend model was built from an entirely different `BlockSIS` realization, although the same variogram ranges and proportions were used. Slices from the same location as before are shown in Figure

10. The initial trend model looks nothing like the correct trend model while the post-processed model converged to the global mean. The converged trend model is completely fair and not over-fit, but is not informative.

Fairness plots and histograms (Figure 11) show the complete incorrectness of the initial trend model. The convergence to the global means is striking in the histograms of the post-processed trend model. This is because `posttrend` cannot add information to the model, only slightly modify it to respect available data. The plots in Figure 11 show that the initial trend model did not respect the data at all.

**Conclusions**

This note introduced two new programs, `plotfair` and `posttrend`, for plotting fairness diagrams and correcting trend models respectively. `Posttrend` was successfully applied to three very different trend models in a small case study. These three trend models were all corrected to fairly represent the available data. It was shown that a small bias in the trend model could be eliminated to give a more reasonable trend model. A trend model that was completely incorrect converged to the global mean.

The probability rescaling and updating procedure introduced in this note seems to be an effective method for avoiding order relation deviations as 0 and 1 are treated the same. By considering the opposite probabilities, this method avoids treating 0 and 1 differently.

Currently, only categorical variable trend models are considered for correction and only primary category data is considered for the correction. This technique could be extended to incorporate secondary data and possibly continuous variables. The corrections made are highly sensitive to user input so care must be taken to avoid over-fitting the available data.

**References**

Deutsch, C.V. and Journel, A.G., 1998, *GSLIB: Geostatistical Software Library and User's Guide,* Oxford University Press, New York, 2nd Ed., pp 142-143.

Deutsch, C.V., 2005, A Sequential Indicator Simulation Program for Categorical Variables with Point and Block Data: *BlockSIS*, *Centre for Computational Geostatistics,* Volume 7, 402.

Journel, A.G., 2002, Combining knowledge from diverse sources: An alternative to traditional data independence hypotheses, *Mathematical Geology*, Volume 34, 5, pp 573-596.

McLennan, J.A., 2007, *The Decision of Stationarity*, PhD Thesis, University of Alberta

Montgomery, D.C. and Runger, G.C., 2008, *Applied statistics and probability for engineers*, John Wiley & Sons, 4th Ed.

Ortiz, J.C., 2001, Research Note HISIM: Hierarchical Indicator Simulation, *Centre for Computational Geostatistics*, Volume 3, 8.

Ortiz, J.C., 2002, Follow Up on HISIM: Hierarchical Indicator Simulation, *Centre for Computational Geostatistics*, Volume 4, 64.

Ortiz, J.C., 2003, Short Note: Naive Bayes Classifiers and Permanence of Ratios, *Centre for Computational Geostatistics*, Volume 5, 84.

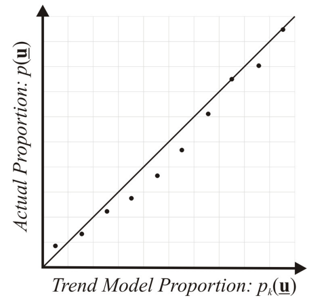


**Figure 1:** Schematic of actual proportions from data plotted against the binned trend model proportions.

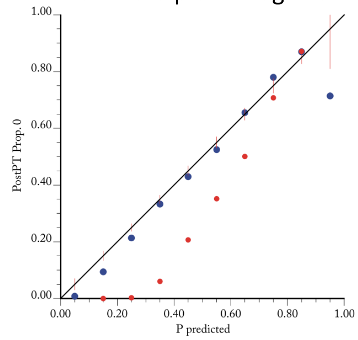


**Figure 2:** Example fairness plot from *plotfair* plotting the actual observed proportion against the trend model proportion and the 99% confidence interval for each point.

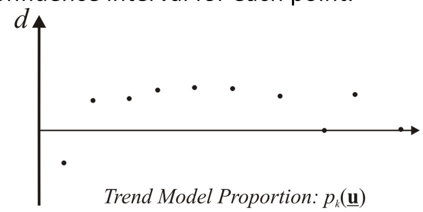


**Figure 3:** Schematic of *d* values from Figure 1 with exaggerated vertical scale.

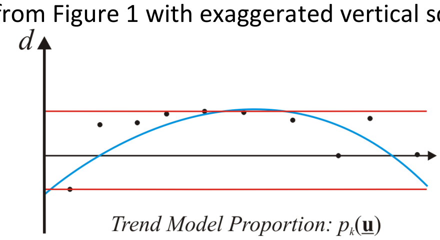


**Figure 4:** Schematic of trimmed quadratic fit to *d* values from Figure 3.

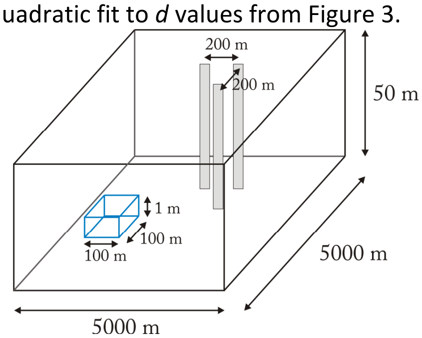


**Figure 5:** Schematic of test case domain

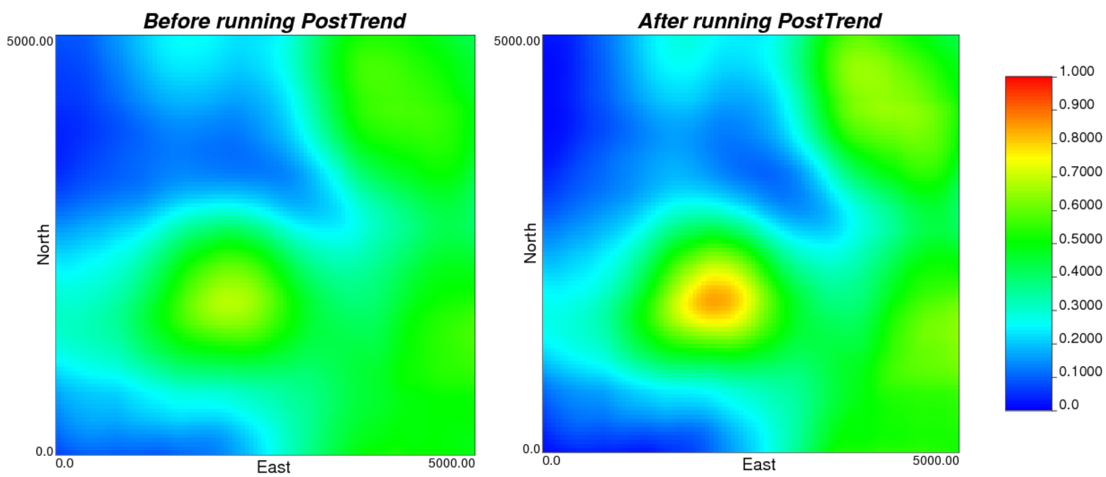


**Figure 6:** Slices of the correct trend model before and after running *posttrend* for category 1

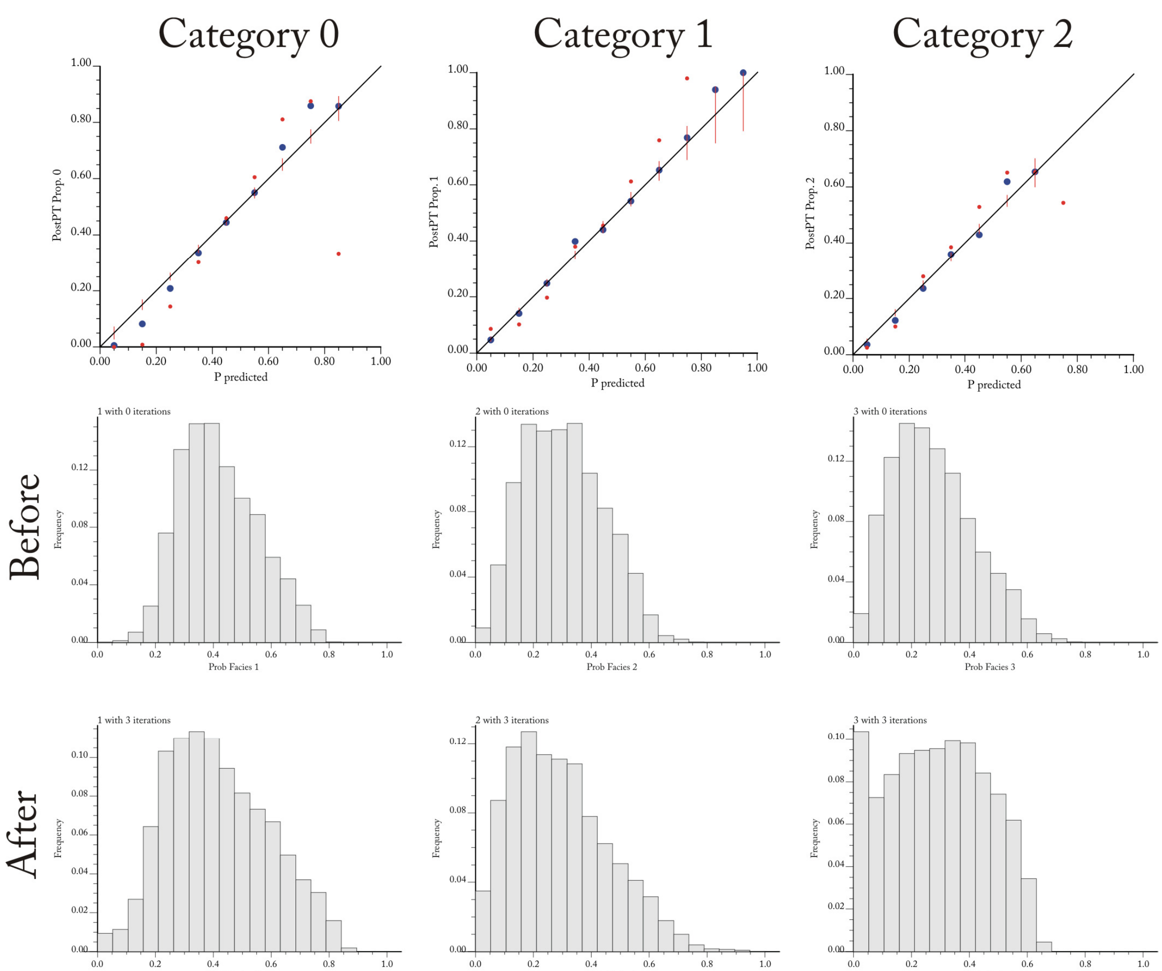


**Figure 7:** Fairness plots from *plotfair* and histograms of the trend model probabilities for the correct trend model

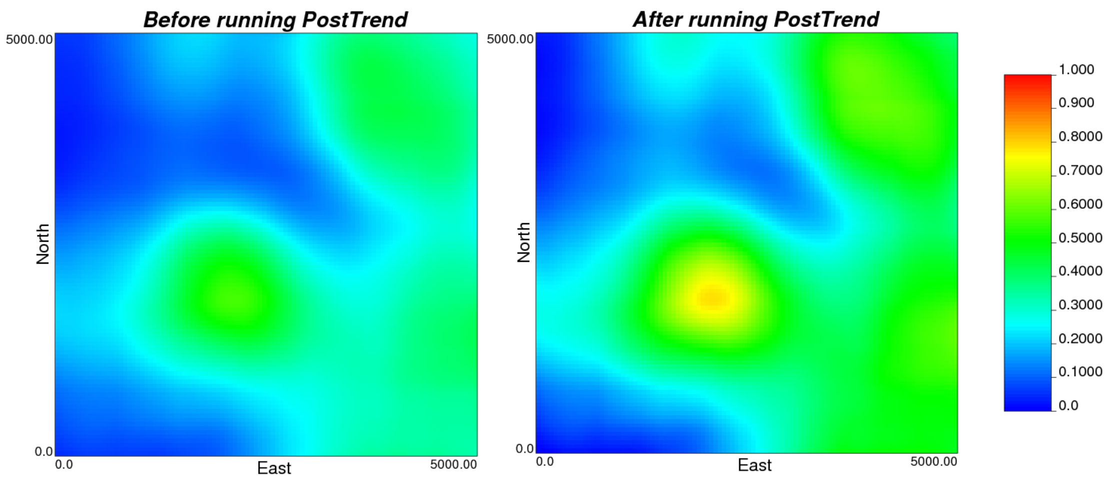


**Figure 8:** Slices of the biased trend model before and after running *posttrend* for category 1

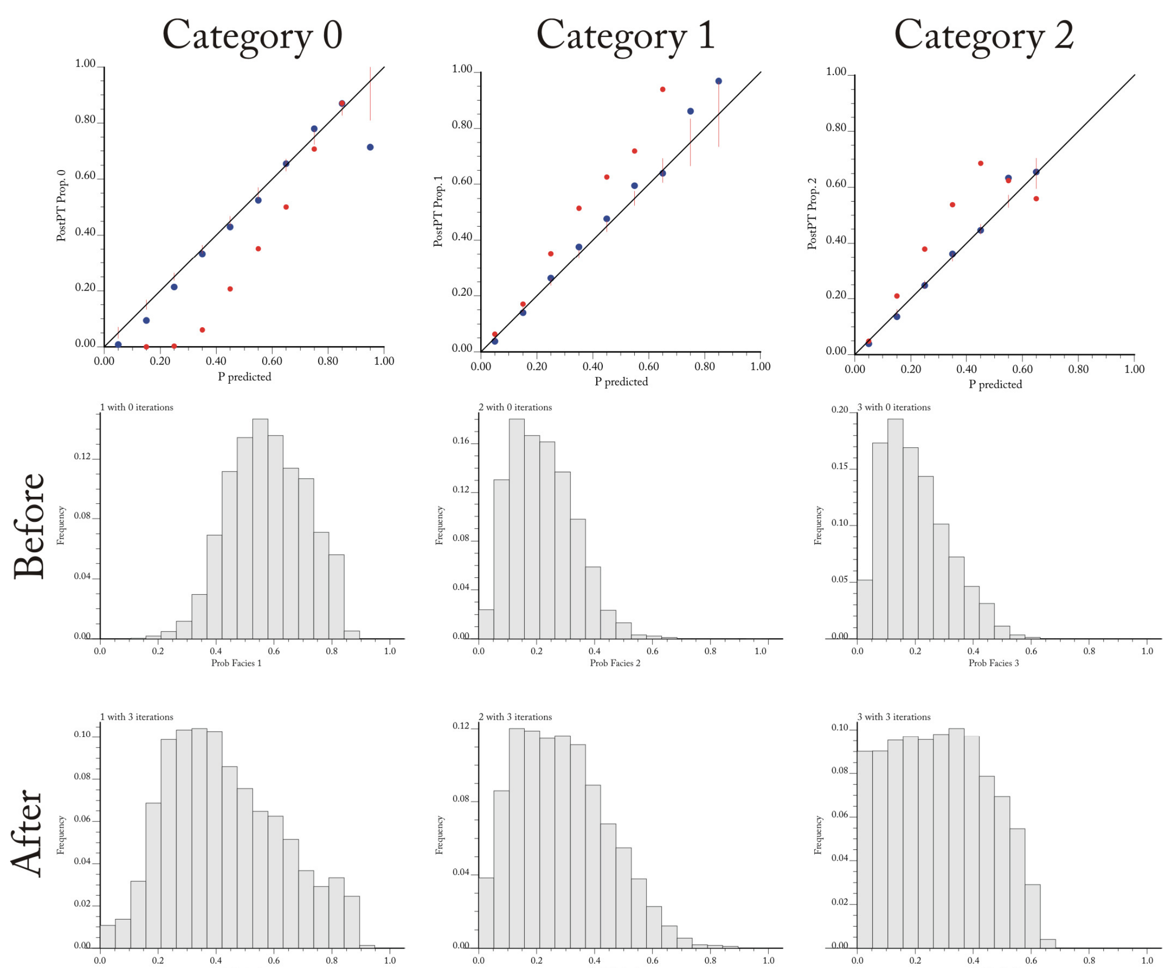


**Figure 9:** Fairness plots from *plotfair* and histograms of the trend model probabilities for the biased trend model

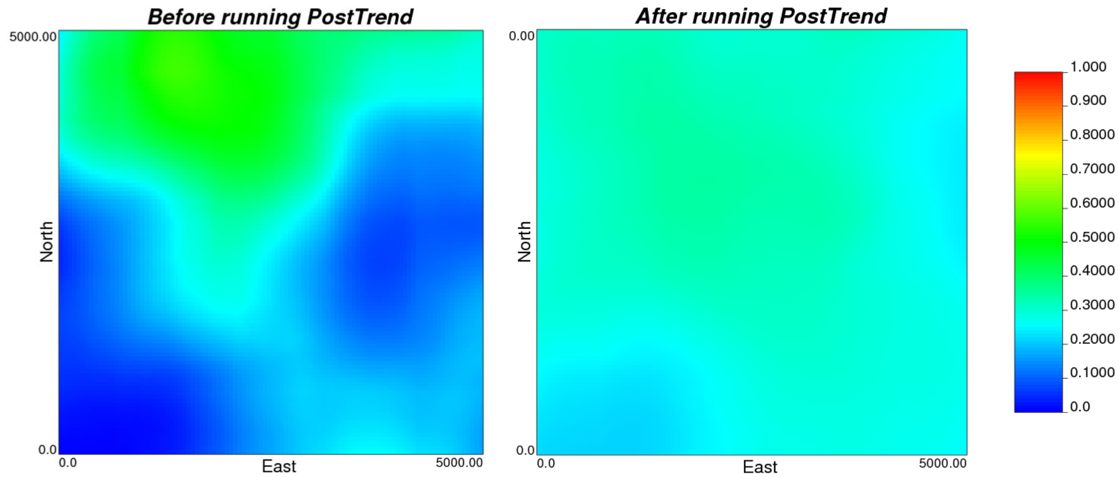


**Figure 10:** Slices of the incorrect trend model before and after running *posttrend* for category 1

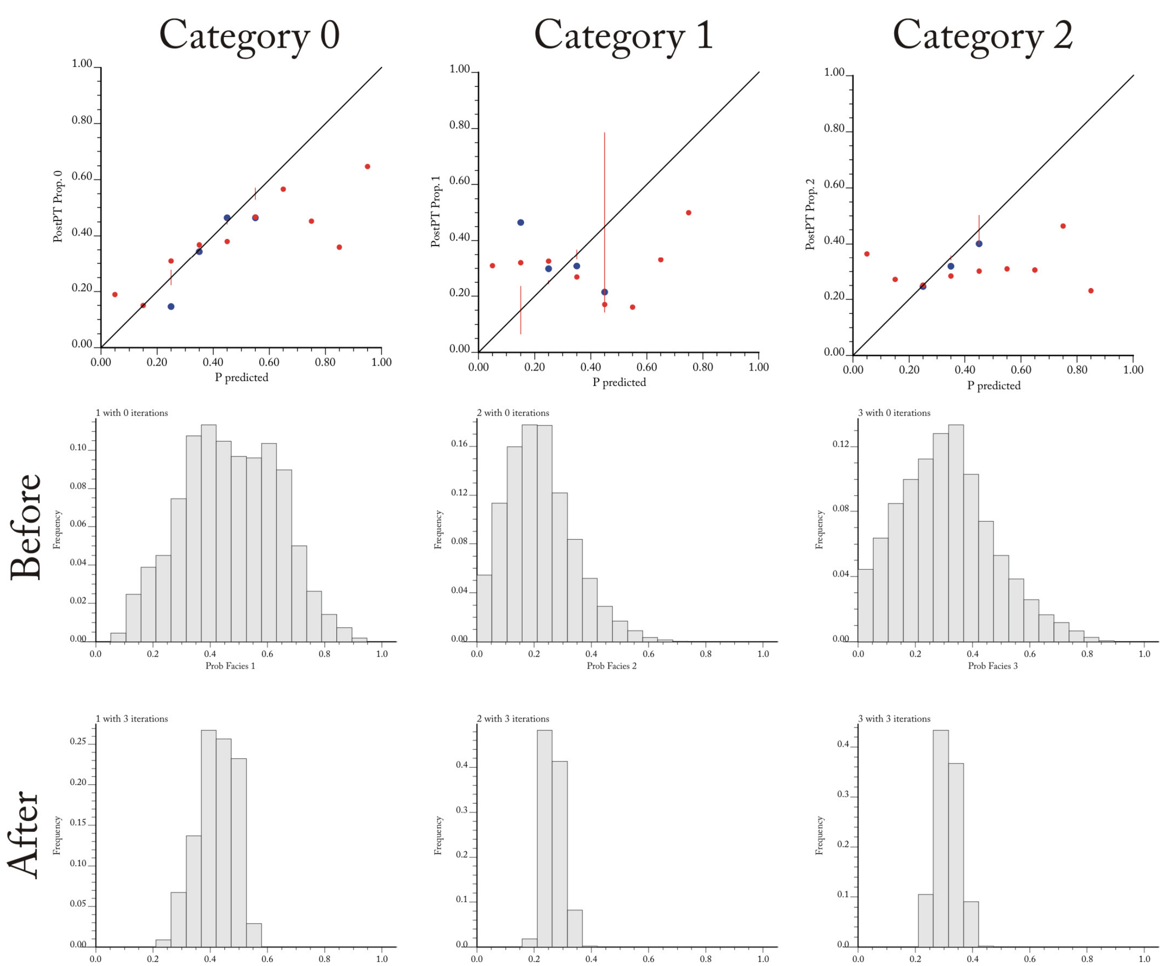


**Figure 11:** Fairness plots from *plotfair* and histograms of the trend model probabilities for the incorrect trend model

**Appendix 1: Program Descriptions**

The first program *plotfair* follows standard GSLIB conventions for the parameter file and input data file from *posttrend*. An example parameter file is shown below. *Plotfair* also requires the data file *plotfair.dat* to use error values from the binomial distribution; otherwise a Gaussian approximation is used. This approximation is reasonable for larger amounts of data or for probability values near the center of the spectrum.

```
1                    Parameters for PLOTFAIR
2                    ***********************
3
4  START OF PARAMETERS:
5  posttrendft.out             -PostTrend fairness table output file
6  plotfair.ps                 -file for Postscript output
7  1.5                         -bullet size: 0.1(sml)-1(reg)-10(big)
8  1                           -plot error bars (1 = yes,0 = no)
```

The *plotfair* program requires the location and name of the output fairness table from *posttrend* (**Line 5**), an output file name (**Line 6**). The bullet size (**Line 7**) can be specified and the user can turn off error bar plotting if desired (**Line 8**).

The second program, *posttrend*, also requires a parameter file and gridded input trend model and input data file. The trend model follows standard GSLIB convention, accepting trends created by *BlockSIS* or any other technique so long as the resulting trend model is gridded (see Deutsch and Journel, 1998). The data file specifies the category at location *x, y, z* in standard GSLIB format.

```
 1                   Parameters for PostTrend
 2                   ************************
 3
 4  START OF PARAMETERS:
 5  3                            -number of categories
 6  0     1      2               -   categories
 7  0.500    0.250    0.250      -   global proportions
 8  wells-condSIS.out            -file with local data
 9  1   2   3   4                -   columns for X,Y,Z, and category
10  lvm-uncond.out               -file with input gridded prior mean values
11   1    2    3                      -   columns with input gridded prior mean
values
12  PostTrend.out                -file for output gridded prior mean values
13  PostTrendft.out              -file for output fairness table
14  100.00   0.00   50.00        -nx,xmn,xsiz
15  100.00   0.00   50.00        -ny,ymn,ysiz
16   50.00   0.00    1.00        -nz,zmn,zsiz
17  3                            -number of iterations
18  0.500    1.000               -a,b where omega = a(1-b/n)
```

The parameters for *posttrend* include the number of categories (**Line 5**) and integer category identifiers (**Line 6**) as well as the global proportions of each category (**Line 7**). The local data file (**Line 8, 9**) and trend model (**Line 10, 11**) are specified along with the output corrected trend model and fairness table for *plotfair* (**Line 12, 13**). The size of the study area is specified using the format given by Deutsch and Journel (1998) in **Lines 14-16**. Finally, the user can specify the number of iterations (**Line 17**) and change the values for *a* and *b* (**Line 18**, Equation 4).

The corrected trend model is output as a gridded prior mean file in the same format as the input trend model. The fairness table is output in a format readable by *plotfair* for making the fairness plots. Part of an example fairness table is shown below. A value of -1 indicates that there was not enough data for that bin.

| Ptrnd | n 1 | Pin 1 | Pfn 1 | n 2 | Pin 2 | Pfn 2 | n 3 | Pin 3 | Pfn 3 |
|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 713 | -1.0000 | 0.0084 | 3499 | 0.0630 | 0.0380 | 5358 | 0.0477 | 0.0386 |
| 0.15 | 2667 | 0.0000 | 0.0934 | 7094 | 0.1701 | 0.1397 | 5733 | 0.2103 | 0.1355 |
| 0.25 | 5958 | 0.0024 | 0.2127 | 6824 | 0.3516 | 0.2629 | 5709 | 0.3792 | 0.2480 |
| 0.35 | 6172 | 0.0598 | 0.3331 | 6285 | 0.5147 | 0.3758 | 5916 | 0.5382 | 0.3621 |
| 0.45 | 5168 | 0.2061 | 0.4294 | 4137 | 0.6267 | 0.4774 | 4751 | 0.6858 | 0.4464 |
| 0.55 | 3925 | 0.3510 | 0.5243 | 2385 | 0.7183 | 0.5945 | 3290 | 0.6245 | 0.6331 |

**Appendix 2: Test Case Construction**

The test case was built using a four stage process to create a trend model created using a moving window average and a well data file. These files could then be read by *posttrend* and the resulting changes studied. An area, measuring 5000m x 5000m x 50m was unconditionally simulated using 500,000 100m x 100m x 1m grid cells. For this, *BlockSIS* was used with the parameter file shown below. For the three categories, 0, 1 and 2, the input global proportions were 0.500, 0.250 and 0.250 respectively. The simulated global proportions were 0.4074, 0.3023 and 0.2904.

```
                          Parameters for BLOCKSIS
                          ***********************

START OF PARAMETERS:
0                              -0=SK,1=OK,2=L1,3=L2,4=CC,5=BU,6=PR,7=BK,8=BC
0                              -Clean: 0=none, 1=light, 2=heavy, 3=super
3                              -number of categories
0     1      2                 -   categories
0.500   0.250   0.250          -   global proportions
0.333 0.333  0.333             -   correlation coefficients for soft data
./well.dat                     -file with local data
1   2   3   4                  -   columns for X,Y,Z, and category
./lvm3d.dat                    -file with gridded prior mean values
1   2    3                     -   columns for each category
3                              -   2-D areal map (2) or 3-D cube (3)
./keyout.dat                   -file with keyout array
1                              -   column for keyout indicator
0                              -debugging level: 0,1,2,3,4
BlockSIS.dbg                   -file for debugging output
BlockSIS-uncond.out            -file for simulation output
1                              -number of realizations
100 0.00  50.0                 -nx,xmn,xsiz
100 0.00  50.0                 -ny,ymn,ysiz
 50 0.00   1.0                 -nz,zmn,zsiz
69069                          -random number seed
12                             -maximum original data  for each kriging
12                             -maximum previous nodes for each kriging
1                              -assign data to nodes? (0=no,1=yes)
0                              -maximum per octant    (0=not used)
5000. 5000.  10.               -maximum search radii
  0.    0.   0.                -angles for search ellipsoid
 100   100   50                -size of covariance lookup table
1 0.0                          -Cat 1: nst, nugget effect
1 1.0 0.0 0.0 0.0              -       it,cc,ang1,ang2,ang3
  1500.  1500.  10.            -       a_hmax, a_hmin, a_vert
1 0.0                          -Cat 2: nst, nugget effect
1 1.0 0.0 0.0 0.0              -       it,cc,ang1,ang2,ang3
  1500.  1500.  10.            -       a_hmax, a_hmin, a_vert
1 0.0                          -Cat 3: nst, nugget effect
1 1.0 0.0 0.0 0.0              -       it,cc,ang1,ang2,ang3
  1500.  1500.  10.            -       a_hmax, a_hmin, a_vert
```

From these values, a locally varying mean (LVM) was extracted using a moving window average. This moving window average used a search cube with a radius of 15 grid cells. This locally varying mean was then used to generate the "truth". Conditional simulation using *BlockSIS* was done with the LVM to construct the true data set. For this operation, the parameter file shown below was used.

```
                          Parameters for BLOCKSIS
                          ***********************

START OF PARAMETERS:
2                              -0=SK,1=OK,2=L1,3=L2,4=CC,5=BU,6=PR,7=BK,8=BC
0                              -Clean: 0=none, 1=light, 2=heavy, 3=super
3                              -number of categories
0     1      2                 -   categories
0.500   0.250   0.250          -   global proportions
0.333 0.333  0.333             -   correlation coefficients for soft data
./well.dat                     -file with local data
1   2   3   4                  -   columns for X,Y,Z, and category
./lvm-uncond.out                    -file with gridded prior mean values
1   2    3                     -   columns for each category
3                              -   2-D areal map (2) or 3-D cube (3)
```

```
./keyout.dat                      -file with keyout array
1                                 -   column for keyout indicator
0                                 -debugging level: 0,1,2,3,4
BlockSIS.dbg                      -file for debugging output
BlockSIS-lvm.out                  -file for simulation output
1                                 -number of realizations
100 0.00  50.0                    -nx,xmn,xsiz
100 0.00  50.0                    -ny,ymn,ysiz
 50 0.00   1.0                    -nz,zmn,zsiz
69069                             -random number seed
12                                -maximum original data  for each kriging
12                                -maximum previous nodes for each kriging
1                                 -assign data to nodes? (0=no,1=yes)
0                                 -maximum per octant    (0=not used)
5000. 5000.  10.                  -maximum search radii
  0.    0.   0.                   -angles for search ellipsoid
 100   100   50                   -size of covariance lookup table
1 0.0                             -Cat 1: nst, nugget effect
1 1.0 0.0 0.0 0.0                 -       it,cc,ang1,ang2,ang3
  1500.  1500.  10.               -       a_hmax, a_hmin, a_vert
1 0.0                             -Cat 2: nst, nugget effect
1 1.0 0.0 0.0 0.0                 -       it,cc,ang1,ang2,ang3
  1500.  1500.  10.               -       a_hmax, a_hmin, a_vert
1 0.0                             -Cat 3: nst, nugget effect
1 1.0 0.0 0.0 0.0                 -       it,cc,ang1,ang2,ang3
  1500.  1500.  10.               -       a_hmax, a_hmin, a_vert
```

Finally, the last step was to extract well data. For this, a 200m well spacing was chosen with the first well at location (100,100). The resulting well file was used for all test cases; only the trend model was modified between test cases.

The biased test case was created by permuting the trend model values for each category by 1.5, 0.75 and 0.75 respectively. These values were then rescaled so that they summed to 1 and were in the domain of [0,1]. The incorrect test case used a seed of 690693 for the first *BlockSIS* parameter file resulting in a completely different realization which was used to build the trend model.