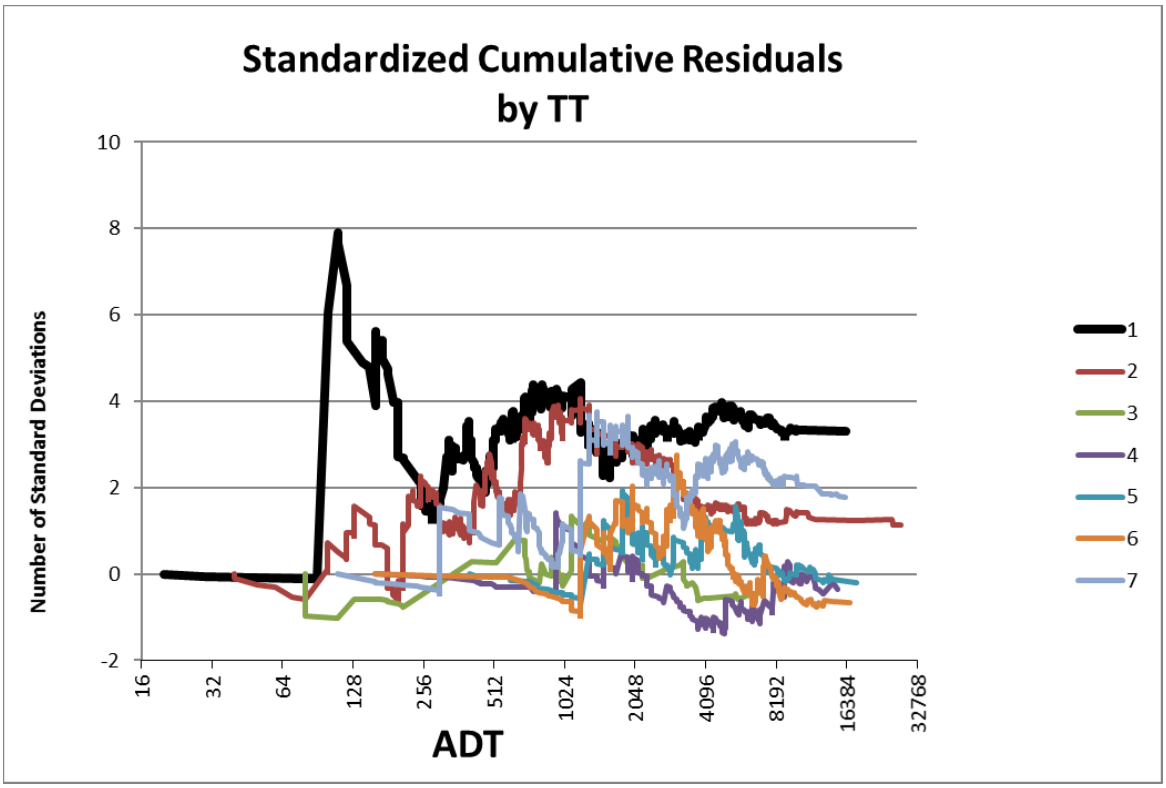


SPF Tool User Manual

Guide to Creating Safety Performance Functions Using SPF Tool 3.5.xlsm

Matt Warren, P.E.

Table of Contents

1.0 Introduction

The accompanying spreadsheet (currently distributed with the name "SPF Tool 3.5.xlsm" and hereafter called "the spreadsheet", "the tool" or "the program") is intended to fit parametric Safety Performance Functions to collision data. It is authored by Matt Warren, P.E., and is available free for use by qualified traffic safety analysts. Permission is granted to use, copy, and distribute both the spreadsheet and this user manual, provided that neither is altered from its original form except for changes of formatting and changes to the spreadsheet that naturally occur in the course of use, that no claims of authorship are made, that programmatic notices conveying this information are not altered, removed, or disabled, and that distribution of any copies, with or without variation, is free of charge and free of any stipulation or encumbrance except as given herein. The spreadsheet is supplied "as-is" and without any warranty or guarantee, express or implied. Persons unfamiliar with the nature and use of Safety Performance Functions should not use this spreadsheet. It is intended only for the use of traffic safety professionals.

The author assumes no duty to support this spreadsheet or to assist users in any way, including correction of critical errors or security flaws or providing relevant information. All users of this spreadsheet assume full responsibility for any and all direct or indirect outcomes of that use, including but not limited to costs, damages, or burdens therefrom arising. The author does not guarantee that this manual is free from error or that future changes to the spreadsheet will be reflected herein.

1.1 **System Requirements:** The spreadsheet has been tested with Excel 2010 and 2016 and may not be compatible with other versions of Excel. It is definitely not compatible with any version of Excel prior to Excel 2007. The Solver plug-in is required. The spreadsheet will not function with any operating system other than MS Windows, even if that operating system supports Excel. The spreadsheet uses macros and will not work unless macros are enabled. The Excel Solver must be installed before the spreadsheet is opened or it will not work (see 2.0).

1.2 **Contact**: Bug reports or requests for information may be sent to Matt Warren at matt.warren.oklahoma@gmail.com. Responses are not guaranteed. Please consult this manual, especially Part 5, before sending a bug report. When sending a report, please include a copy of the spreadsheet saved at the point of failure if possible. Bug fixes or upgrades might be sent to users who have requested them.

1.3 **Advantages:** This program is primarily a wrapper for the Excel Solver. Advantages over using the Solver alone include:

- Automation of various operations, such as the addition of parameters to functions and the creation of a variety of graphs.
- Quick access to graphical information about the fit of a model, with residuals plotted in various ways against any available variables and controllable smoothing of the graphs.
- Improved solutions. The Excel Solver does not work well for large data sets and equations with multiple parameters, because the gradient of the optimization function can be extremely small,

leading to premature halting and poor solutions. This program can force the Solver to find solutions closer to the true optimum.

- Functional parameters and independent variables are automatically and transparently scaled. The Solver's own scaling algorithms, as of Excel 2010, are generally ineffective, also leading to premature halting and poor solutions.
- Parameters may be constrained to the interval $(0,\infty)$, an option not available in the Solver as of Excel 2010.
- Initial guesses for parameter values are made automatically.
- Quick switching between alternative solution optimization methods and constraints.
- Several measures of performance are calculated automatically, including cumulative residual plots.

1.4 **Limitations:** Use of the spreadsheet requires a data set including a column of crash counts and at least one column of roadway data, with one row per site. This data must be prepared externally in some form which is capable of being copied into an Excel spreadsheet. The data may not exceed 65,535 rows plus one row for column headers, and may not exceed 98 columns of roadway data plus one column of crash data and one column for site identification.

1.5 **Assumptions:** The model assumes a negative binomial distribution of crash counts. It does not make any assumptions about the functional form of crash prediction models or about the form in which overdispersion is expressed.

1.6 **Cautions:** The spreadsheet is supplied "as-is" and without any warranty or guarantee, express or implied. The author assumes no obligation to provide support of any kind. The spreadsheet has limited data screening functions but cannot assure that data provided are accurate, meaningful, or negative binomially distributed. Solutions which are reached could be local optimums but not global optimums; solutions may also not be optimums at all due to premature halting of the Solver (see 4.5.5). **The acceptability of a solution for practical use must depend on the judgment of a person with knowledge of traffic accident data analysis**.

The spreadsheet is intended for use by persons **familiar with Excel and having knowledge of the basics of Empirical Bayesian traffic accident data analysis and the use of Safety Performance Functions** (as outlined in the Highway Safety Manual). The spreadsheet is not a substitute for this knowledge.

Users should not make changes to any of the hidden worksheets in the spreadsheet, nor to any cells except those designated for user input, nor to any chart. Minor formatting changes such as column width and cell color may be made but could be automatically reversed by the operation of macros. Any other changes should only be made with caution by an expert Excel user familiar with the spreadsheet. Changes for example to charts, to the number format of cells, to sheet protection, or to the merging of cells, could result in failure of the spreadsheet. Worksheets may not be renamed but new worksheets may be added. Deletion or renaming of user-added sheets is permissible but deletion or renaming of any other sheet, or changes to hidden sheets, is likely to result in irreversible failure of

the spreadsheet. **Cut-and-Paste or Drag-and-Drop operations should NEVER be performed** except in the Data worksheet.

The spreadsheet currently does not use sheet protections. Anyone with access to the spreadsheet must be familiar with Excel and also with these cautions; otherwise irreparable corruption of the spreadsheet is likely.

"Undo" will generally not be available within the spreadsheet, due to irreversible macro operations which occur on all button clicks and also on most changes to cells in the "Model" worksheet. External copies of the data used should be retained; if the data is accidentally cleared from the spreadsheet it might not be recoverable.

Always keep a backup copy of the original spreadsheet as you received it.

2.0 Installation

For Excel 2007 to 2016:

1. Open Excel and activate the Solver Add-In. For Excel 2010, the sequence is File/Options/Add-Ins/Go. Check the box next to "Solver Add-In" and OK.

2. Enable macros. For Excel 2010, the sequence is File/Options/Trust Center/Trust Center Settings/Macro Settings. Check "Enable all macros" and "OK".

   Note: Excel 2010 and later do not allow any macro security, unless the user is willing to re-enable macros every time a spreadsheet with macros is used and then disable them. If macros are left enabled, that computer could be hacked if a macro-enabled spreadsheet containing malware is opened. It is recommended to use virus detection software and to verify the origin of all macro-enabled (.xlsm) spreadsheets.

3. Open the spreadsheet. If you are asked for a password, cancel the requestor; do not enter a password (even an empty one). You should get the message "This software is available strictly "As-Is", with no express or implied warranty. OK to proceed?" If you do not get this message, macros are not correctly enabled.

SPF Tool 3.5 has been extensively tested only with Excel 2016. Earlier versions have been used with Excel 2007 and 2010 but backward compatibility of the current version has not been tested. The current version is definitely no longer compatible with Excel 2003.

3.0 Content Summary

The spreadsheet includes worksheets which are always visible, worksheets which are sometimes visible, and worksheets which are normally invisible, as well as numerous VBA macros including some automatic (event triggered) macros. It may also include sheets added by users, which could contain e.g. notes, links, data, or saved results of previous SPF runs.

The **Model** worksheet is the key work area. It includes:

1. an area at the top left (yellow) where equations are entered and coefficients for solutions are reported;
2. an area at the left (orange) where information on data variables is reported;
3. an area at left center (blue) where model parameter values are reported;
4. an area at center (green) where solution criteria can be chosen;
5. an area below center (purple) which contains various metrics of the current solution;
6. a chart at upper right showing any of several relations between actual data and the model prediction;
7. A box underneath the graph showing possible functional modifications associated with the trendlines on the graph;
8. Various buttons (gray) for project initialization, solver operation, and graphing options.

The **Data** worksheet is a structureless area for receiving site data from an external source.

The **Results** worksheet is not visible unless there are results available. It summarizes the results of the last successful Solver run.

The **Outliers** worksheet contains any outliers *deleted* from the data set last loaded – it is not a list of outliers detected.

The **Log** worksheet contains a record of the most recent solution process for diagnostic purposes.

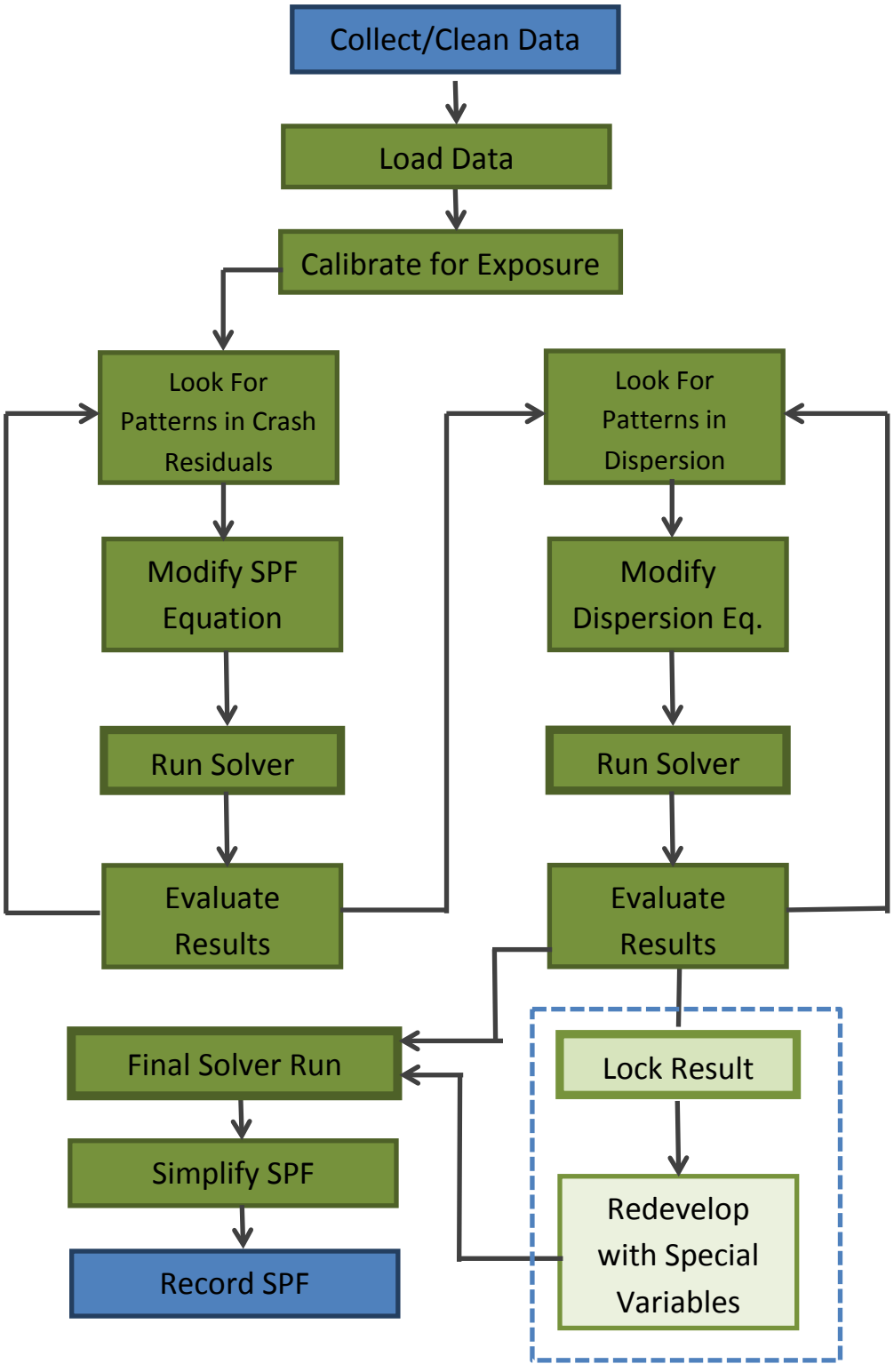
The **Example Data** worksheet contains a set of data that can be used to reproduce the example given in Appendix C (as of version 3.5).

The **Report** sheet gives a summary of information to help identify the problem when an equation fails (such as during a Solver run).

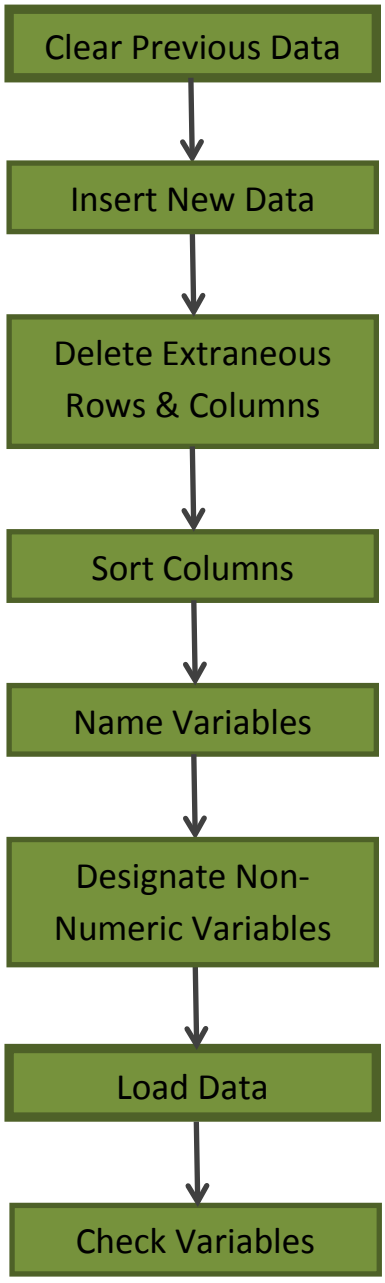
Other sheets are normally hidden but some may be useful in tracing the cause of failure of an equation or graph anomaly.

4.0 Operations

The basic SPF workflow structure is shown below. The first and last stages (collecting the data and recording the SPF) are not a function of the spreadsheet and are outside the scope of this manual; they are shown as indicators of where this spreadsheet fits in the overall analysis process. The stages are not rigid and not need be carried out exactly as shown. Steps shown in light green are an option for special purpose SPFs.



4.1 **Loading the Data (mostly "Data" worksheet)**



4.1.1 Preparing the Spreadsheet: If this is a new data set, click the "Clear Data" button in the Model sheet to prepare for new data entry. This does NOT remove existing data for calculation purposes, which is still available and unchanged in the (hidden) Calcs sheet. Use of the "Clear Data" button before entering a new data set is strongly recommended even if the Data worksheet appears empty, to prevent possible scaling errors caused by previous formatting.

This version of the spreadsheet will not recognize more than 65535 rows of site data, nor more than 98 columns of variable data (i.e. not including site identifier and crash counts), regardless of the version of Excel used.

4.1.2 Arranging the Data: The site data must be in a form which can be pasted into Excel. Each row of data represents one site, as defined by the user (e.g. an intersection or road segment). Each column represents one field describing that site.

The top row of the data must contain the column headers; every other row should contain data. Any additional rows used (e.g. for sub-headings or extended headings) must be deleted before loading.

The data must be placed in the "Data" worksheet. If it is being copied from another spreadsheet, use the "Paste Values" option to ensure that values, not formulae, are pasted. The data does not have to start at A1 – any rows or columns left blank will be automatically removed.

In the Data sheet **ONLY**, the user may freely move data, insert or delete rows or columns, change formats, etc. If formulae are used here, remember to use Paste Values to convert them to constants before attempting to load the data.

Except for the crash data column and the site identifier column (if any), each data column will result in the creation of a variable for use in equations. It is a good idea to delete unnecessary columns to avoid an excessively long variable list.

NOTE: Be sure to retain *only* data fields which will be usable when the SPF is deployed! Data fields may be available for development that have excellent explanatory power and can contribute to a SPF of very high quality, but this will be unimportant if the SPF cannot actually be used in practice.

It is highly advisable to insert a column for the **number of years** of data, even if this is the same for all data points, so that time duration may be explicitly included in formulae. Remember not to add this column to the right of the **crash** counts.

If your data includes a site identifier (primary key) it must be in the **leftmost column** of the table. Crash counts must be in the **rightmost column** of the table. *Failure to put the crash data in the rightmost data column will produce disastrous results even if no error message is generated.*

4.1.3 Naming the Variables: The other column headers should each be changed to a suitable **variable name** which can be used in formulae. Variable names may include numbers, letters, and/or any of the following special characters: ! # $ % @ \ _ ` ~ | [ ]. Variable names may not consist entirely of digits; they must have at least one letter or special character. Variable names are not case sensitive; they will be converted into uppercase automatically. Note that blank spaces are

not allowed in variable names. Duplicate variable names are not allowed. Short variable names are recommended because you may need to retype them many times into formulae.

In formulae entered by the user into the spreadsheet, the variable names will represent the site data from the corresponding column.

4.1.4 Data Format Control: Any columns which do not contain real numerical data should be formatted as text. For example, if the column "TER" is an integer 1-7 that is a look-up code for a terrain classification, and it is not formatted as text, it will be scaled by default, resulting in values ranging from (say) 0.48 to 3.36. In most cases the system will recognize which Excel functions (like CHOOSE) need to use the unscaled value of a variable, but at best scaling a non-scalar variable is wasting processor time and complicating the final function.

4.1.5 Loading the Data: When the preceding steps are complete, Go to the "Model" worksheet and left click the "Load Data" button. The data loading process may take several minutes depending on the amount of data.

You will get a message asking you whether to use field "*xxx*" as a location identifier instead of as a data field. If *xxx* (your leftmost column) is a site identifier, click Yes. Otherwise click No and the program will create a site identifier column for you. Cancel if you need to go back and rearrange your columns.

You may get warning messages concerning the content of certain fields (e.g. fields containing blanks), and you may be asked whether to continue the load or abort it. Choose the latter only if you think you need to check or correct your data; blank values may be acceptable for many fields. Blanks in a numerical data field will be converted into zeroes.

You will get a requestor asking you whether to clear formulae. If the last project was similar to the current one, you may wish to avoid clearing formulae so as to save retyping the equations. You can also clear the formulae at any time by using the "Clear Formulas" button.

Note: The data in the Data worksheet are not used directly in calculations. The spreadsheet will use whatever data were last loaded; *changing the data in the Data worksheet will have no effect until the data are loaded again* using the "Load Data" button.

4.1.6 Checking Variables: Check to be sure that your variables have loaded correctly. The variable list appears on the lower left of the "Model" worksheet (in the peach-highlighted portion). To the right of its name, each variable has a checkbox (in the column labeled "*θ") that determines whether the variable will be scaled.

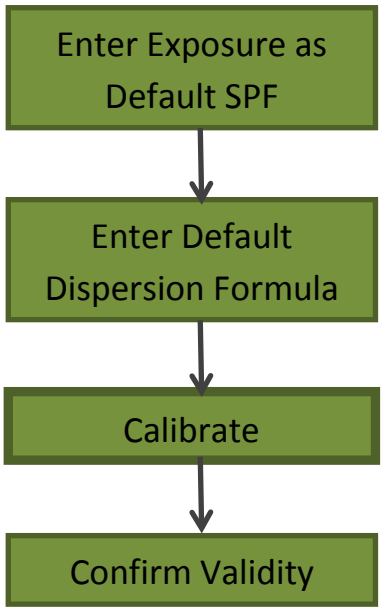
Scaling should be used only on variables which are truly numeric. If you forgot to designate a variable as non-numeric before loading, you should now uncheck the "*θ" for that variable. If the "*θ" box is highlighted brown, the variable cannot be scaled. If the range of values of a numeric variable is small, unscaling it will cause the final SPF results to initially appear in a

slightly simpler form. However, if a numeric variable has a wide range of variation, failure to scale it may cause the Solver to fail or to give an incorrect solution.

The third column in the variables area is the scaling coefficient which is applied to each variable before it is used in formulae. This is for user information only; changing these cells will *not* change the value of the coefficient. The SPF and Overdispersion Formula use the scaled values of variables, unless the program identifies them in the comparison argument of an IF or CHOOSE function; the other formulae always use the raw, unscaled values of variables. If a variable is not scaled, its Coefficient will be blank.

Variable names may be changed in the Model sheet, but if this is done there are no checks for invalid or duplicate variable names that would cause functions to fail or produce invalid results.

4.2 **Establishing a Baseline ("Model" worksheet)**



4.2.1 Entering Exposure SPF: The Crash Prediction Formula calculates an estimate of crash counts, based on known site data *other* than previous crash counts. Prima facie, crash counts can be expected to vary with exposure (traffic volume), and you should start with this as a baseline before looking for more subtle influences.

For highway segments, exposure would be equal to Length (Miles) × AADT × Time (Years). If you named these variables "L", "A", and "T", the exposure SPF would be "=L*A*T". Enter this (without the quote marks) in the box for "Safety Performance Function" on the "Model" worksheet (in the yellow area at upper left). Note that you do not have to include any parameter to allow for scaling; this is done automatically by the program.

For intersections, exposure might ideally be given as ($AADT_{major}$ + $AADT_{minor}$) × Time. If you named these variables "A1", "A2", and "T", you would enter "=(A1+A2)*T". If minor approach AADT is unavailable, you could use just $AADT_{major}$ × Time, or perhaps try some other variable as a surrogate. To do this you might have to introduce parameters into the function (see 4.4.5).

If you are using a previously established function as a starting point, instead of starting from scratch, skip this step and the next one as well.

4.2.2 Entering Default Dispersion Formula: This formula expresses "Overdispersion" in the same form as the Highway Safety Manual; i.e., as the reciprocal of the Dispersion as typically characterized in other disciplines. For our purposes, a high value of Overdispersion thus indicates that our crash sites have important risk factors that are not in our data, and the Empirical Bayes formula will give more weight to crash counts than to other site factors (such as AADT). A low value of

Overdispersion indicates the opposite – but note that neither indication necessarily tells the whole story. The Overdispersion value does NOT indicate whether the SPF is correct, but a very high value does indicate that the SPF won't make much difference to crash projections.

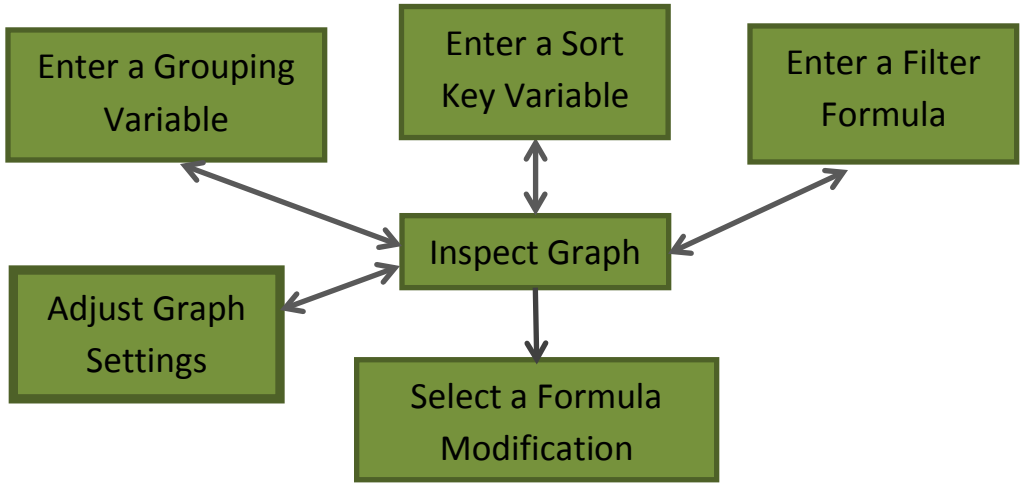
For highway segments, one method is to calculate Overdispersion as the reciprocal of segment length. If you have named the corresponding variable "L", you would enter "=1/L" or the equivalent in the Overdispersion Formula box. A better theoretical basis of Overdispersion is that it is inversely proportional to the SPF prediction; you would enter this as "=1/?" or the equivalent: ? is a special reserved character for the purpose of representing the output of the Crash Prediction Formula in other formulae. For intersections, Overdispersion can take this form or can be initially presumed constant, in which case you would not enter anything, or an intermediate form could be used such as 1/SQRT(?). As with the SPF, you do not have to enter a scale parameter for Overdispersion – the program creates one internally.

4.2.3 Calibrating the Model: For baseline calibration, use the "Calibrate Model" or "Calibrate SPF" button. The former calibrates Overdispersion as well, which may or may not be desirable – at this stage of operations, it is not unusual for calibration of the Overdispersion to fail, but without it most of the solution metrics will be meaningless.

Calibrating does not change the values of any parameters, just the scaling coefficients for the SPF and/or Overdispersion formulae. The SPF coefficient is set to make the total number of predicted collisions match the total number of observed collisions; the Overdispersion coefficient is set to minimize the sum of squared excess variance (see 4.5.1.2).

The Solver can also be used to calibrate the initial SPF, but calibration is much faster.

4.3 Looking for Patterns



4.3.1 Entering a Sort Key: The Sort Key comprises the X-axis of every graph. Enter the name of a variable into the Sort Key box to see how crash rates, dispersion, or cumulative residuals vary with that variable. It is also possible to enter a formula using any number of variables, but this is less likely to be helpful. The Sort Key equation may include variables, constants, and Excel functions/operators, but not parameters or cell references. The Sort Key may also be, or include, the special symbol ? (question mark). This symbol represents the calculated SPF value for each site. The user may thus sort the data and observe whether there is any pattern or bias relative to site crash predictions.

It is possible to use a non-scalar variable for the Sort Key, if the values are all numbers. In this case, points may be plotted which represent mixtures of incommensurable data points. For instance, if the Sort Key is a variable "RU" for which 1 represents Rural and 2 represents Urban, a point will be plotted which represents both Rural and Urban data points, and thus appears somewhere between 1 and 2 on the X-axis, even though this location on the X-axis has no meaning. If the values of "RU" are recorded as "R" and "U" instead of 1 and 2, it will not work as a Sort Key at all. Groups (4.3.2) are more useful for investigating non-numeric variables.

When the Sort Key is blank, the data sites are sorted randomly. The graph should then appear somewhat flat, no matter what the settings are. The varying appearance of the graph without a Sort Key will give the user some idea of what the graph should look like when the formula is free of bias with regard to the current Sort Key.

4.3.2 Grouping the Data: Grouping works similarly to the Sort Key, but instead of changing the X-axis of the graph, the grouping variable (or formula) is used to divide the data into groups and plots a separate line on the graph for each group. The program will attempt to determine the

appropriate number of groups; if it detects a continuous variable like AADT it will default to seven groups. The user can change the number of groups using the "# Groups" cell but cannot change the dividing points between groups. The maximum number of groups is twenty.

Grouping works with both scalar, non-scalar, and text data.

When data are grouped, the trendlines are removed from the graph to allow legibility. If it is desired to see the trendlines for a particular group, Filtering (4.3.3) may be used.

Grouping can be used to see the overall difference in crash residuals for different values of a variable by using that variable for grouping and entering a constant for the Sort Key. If the Grouping variable is numeric, it can be entered for the Sort Key as well and this will make the graph easier to read. If the Grouping variable is non-numeric, leaving the Sort Key blank will make the graph easier to see but will also add some noise to it.

4.3.3 Applying Filters: A Filter is a formula entered on the "Model" worksheet; it has nothing whatever to do with the Excel Autofilter. A filter causes only part of the data set to be used and displayed; it allows the user to select a subset of the data (e.g. rural two-lane highways only) to see if it behaves differently from the whole, or even to create a SPF for a subset of the data.

The Filter formula is entered in the "Filter" box in the yellow area at upper left of the "Model" worksheet. It should be an Excel formula returning a Boolean (True/False) value (otherwise it may not work as intended). The formula can include variables, constants, and Excel functions/operators, but not parameters or references to cells. Only sites for which the formula returns "True" will be included in the display or in any solution produced. The "# Data Points" box to the right of the Filter formula will show how many data points were included by the Filter. If the Filter is left blank, all sites in the data set are included in all calculations.

For example, if the intent is to look only at sites with two lanes, and the number of lanes variable is named "NL", the user would enter "NL=2" into the Filter box (the leading = sign is optional in all formula entry boxes).

The Filter always uses the raw values of variables, i.e. without scaling or conversion to positive-only values.

4.3.4 Understanding the Graphs: There are four available graphs (all of which appear alternately on the same chart, as selected by the user). Each of these graphs reveals something about the relation (or lack thereof) between the Sort Key variable and some pattern in the crash data.

In general, the ideal in creating an SPF would be for all four graphs to be *relatively* flat for all Sort Keys – this indicates that any systematic relation between that variable and crash counts or dispersion has been explained. In practice, this may not be achievable. However the user

should strive for a solution that at least has a reasonable Cumulative Residual (CURE) plot (see 4.3.4.3) for the major explanatory variables, including AADT. However, when using the "Weighting" function (4.4.3) the CURE plot may be dramatically biased (and wrong).

On the SPF and Dispersion graphs, ungrouped data are shown by a heavy black line or (if the Smoothness on the graph has been reduced enough) by blue squares. The thin colored lines are trendlines that may help highlight simple relationships in the data (see 4.3.5). If the data are grouped, each group is represented by a thin colored line, and the legend on the chart will show the range of each group.

NOTE: The scales on both axes of all graphs are adjusted automatically. Do not rely on visual appearance alone to indicate flatness of the curve! An improvement in fit will frequently result in a *loss* of visual flatness, because the scale on the Y-axis of the graph has contracted.

4.3.4.1 The SPF Graph: This graph option is activated by the "SPF" button to the left of the chart (upper right of the "Model" sheet); when it is active the chart will be titled "Actual Crashes vs. Predicted Crashes". What it displays is not the SPF itself but the *residuals* vs. the Sort Key, given the last *solved* SPF. If the current SPF entry has not had a solution run, this graph will NOT be correct.

For X-axis (Sort Key) regions of the graph where the Y-value is higher than 1 (in the standard ratio mode), the current SPF has underestimated the crash rate and an increase in the SPF might be appropriate. For regions where the Y-value is less than 1, the current SPF has overestimated the crash rate and a decrease might be appropriate. The box below this graph may provide helpful suggestions as to what terms might be added to the current SPF to improve its fit.

In "ratio" mode (which is the default), the graph shows the ratio between actual crashes and predicted (by the SPF) crashes. A new term based on this graph would be appended to the current SPF by multiplication. In "difference" mode, the graph shows the arithmetic difference between actual crashes and predicted crashes. A term based on this graph would be appended to the current SPF by addition.

When in "ratio" mode and with the Y-axis set to logarithmic, points may appear on the graph which have extremely low Y-values. This usually indicates that the actual crashes being represented at that X-value were zero. These points may cause serious distortion or (if graph smoothness is set low enough to show points) the graph may break into two bands. Increasing the graph smoothing may produce a more accurate picture of the actual situation.

The title of the X-axis includes the Pearson correlation (r) between the current Sort Key and the crash residuals. A large (positive or negative) value of r indicates that a potentially exploitable relation exists. A low value of r does not necessarily show the opposite; certain kinds of relationships between variables can be strong but have negligible Pearson correlation.

4.3.4.2 The Overdispersion Graph: This graph option is activated by the "1/Φ" button to the left of the chart. (Φ is a common symbol for dispersion; here we are dealing with its reciprocal, sometimes called "Overdispersion"). When this option is active, the chart will be titled "Actual vs. Predicted Overdispersion". This graph serves essentially the same purpose, in relation to the Overdispersion function, that the "SPF" graph serves in relation to the Safety Performance Function.

Overdispersion for each site is estimated as the squared residual minus the predicted mean, divided by the square of the mean. This method can produce negative estimates, but actual Overdispersion is always positive. When the graph is in difference mode, it simply shows the difference between the estimated Overdispersion and the result of the Overdispersion formula. When the graph is in ratio mode, it shows the ratio of estimated Overdispersion to the formula, with negative values being shown as extremely small positive values. This can cause the same kind of distortion in the graph that is caused in the SPF graph by points with zero crashes.

4.3.4.3 The CURE Plot (Cumulative Residuals): This graph option is activated by the "CURE" button to the left of the chart. It shows the cumulative residuals (of crashes compared to SPF), divided by the nominal standard deviation of those residuals at each point. It is a good tool for quickly revealing systematic bias of the SPF with respect to the Sort Key. The CURE plot is unlikely to ever be flat (which would probably indicated an over-fitted model), but (except perhaps for a few points at the low end of the X-axis) the scatter of the plotted values should ideally follow a standardized normal distribution. That is, about 95% of the points should lie within the Y-axis region from -2 to +2 (i.e., within 2 standard deviations of mean), and the majority should lie within -1 to +1. If any substantial number of points are at less than -3 or more than +3, a probable bias is indicated, and the nature of this bias is often quite obvious from the shape of the plot.

Note: The CURE plot in this program does not display the usual ±2σ envelope because the values are already standardized. That envelope, if plotted, would consist of straight horizontal lines at -2 and +2 "Number of Standard Deviations". Also unlike many implementations of CURE plots, the graph does not necessarily end at zero (the assumptions behind that behavior may not be true when using this tool).

For rare crash types where most of the crash counts are zero and few are greater than one, the CURE plot will typically be more jagged with more extreme points; this is because the basic assumptions underlying the calculations are no longer true, and the plot may still be acceptable.

At the extreme low end of the X axis, the CURE plot graph may show extreme values. This is typically because the plot is based on the assumption that the sum of residuals is normally distributed, which is only even approximately true when at least 31 data points (sites) have been added. A value of ±4 or even more at a point on the graph representing only the first (i.e. lowest) ten sites on the X axis may be perfectly acceptable.

The shape of the CURE plot can also be a clue to the presence of bias, data issues, and/or outliers even if none of the graph exceeds the ±2 standard deviation envelope. If the graph rises steadily to +2σ and then drops steadily back to zero, this is an obvious indication of a fairly simple bias which should be correctable. A sharp and perfectly vertical change in the CURE plot may suggest the presence of an outlier in the data, especially if the "cliff" is present for all Sort Key variables. A sharp, but not necessarily vertical, slope or hump in the graph may indicate the presence of an anomaly in the data set. For instance, if the Sort Key is Median Width, and Interstates have a minimum 40' median, there could be a sharp jump at the Median Width = 40' value since values below that do not include any Interstates.

When the Weighting function is in use, the CURE plot will be distorted because the underlying assumptions are violated. If there are any data points with high Weights, the CURE plot is entirely meaningless.

Note that unlike the other graphs, the CURE plot is not smoothed – every data point is represented, although some may be displayed one atop another due to having Sort Key values which are effectively the same.

The calculations that this tool uses for the standard deviation of the cumulative residuals are different from the usual general-purpose CURE assessment; equations specific to the negative binomial distribution are used. To the extent that the data are not negative binomially distributed, the CURE plot will be inaccurate.

4.3.4.4 The Odds Ratio Plot: This graph is activated by the "ODDS" button to the left of the chart. It shows how improbable the crash counts were, *supposing* that the current model is perfect. What is graphed is, approximately, the standardized logarithm of the odds against the observed crash count being as high it was; the graph should be approximately Normally distributed. Negative values indicate an improbably *low* crash count. The Odds Ratio plot can help detect outliers and also help determine whether an apparent bias in the SPF graph is significant or merely due to expected random variation.

4.3.5 Using the Graph Tools: In addition to the four basic graph modes, there are several tools that can modify the graph to help the user see underlying patterns in the data. These are activated by various buttons and selectors to the left of the chart on the "Model" worksheet:

4.3.5.1 Smoothness Control: This is a drop-down menu selector allowing the user to choose a number between 0 and 10. The number of points actually displayed on the chart is approximately equal to the total number of sites divided by 2 to the power of N where N is the user-selected number. At Smoothness 0, every site is plotted separately; otherwise, each plotted point is a weighted average of a varying number of sites.

The program will automatically choose a graph smoothness number when the data are loaded,

and if a Filter is applied (see 4.3.3) it will attempt to adjust this number to preserve the appearance of the graph. However, users may find it useful or necessary to change the graph smoothness to get a clearer picture of the underlying relationship.

If the data are Grouped, the actual smoothness applied to each group will be different but the Smoothness control will still affect all the groups.

If graph smoothness is relatively low, the chart will (except in CURE plot mode) show a disconnected cloud of points rather than a continuous line. If graph smoothness is excessively high, the graph may display a single straight line, or nothing at all.

The Smoothing of the graph can be unstable at the tails at high Smoothing settings. Certain data points near the low and high ends of the Sort Key may be represented more heavily at some levels of Smoothness than others.

4.3.5.2 Axis Controls: Six buttons control the scaling of X- and Y-axes on the graph. "XLog" and "YLog" set the corresponding axes to logarithmic scaling; "XLin" and "YLin" set them to linear scaling. For the SPF and Overdispersion graphs, the Y-axis is logarithmically scaled by default, and this is usually the most useful setting. For the CURE plot and ODDS graph, the Y-axis is always linear. The X-axis is usually set to logarithmic by default unless the Sort Key has zero or negative values for some sites.

If the X-axis is set to logarithmic and the data for the Sort Key includes zeroes (or negative values), these will be plotted anyway by setting them to a very low positive value. This can cause the graph to appear distorted. The same thing will happen on the Y-axis if the Smoothness is set low enough that points with a non-positive value (zero crashes or negative nominal Overdispersion at a site) are plotted.

The "Ratio" and "Diff" buttons set the Y-axis to display either the ratio of actual to predicted (crashes or dispersion) or the difference between actual and predicted. Generally the Ratio setting will be the most useful. The CURE plot is not affected by this setting.

4.3.5.3 Refresh button: In the upper left area of the Model sheet is the "Refresh" button. Clicking this will refresh the graph if it has failed to do so correctly, recheck for function errors, and reactivate event macros if they have become disabled due to a crash.

4.3.6 Choosing a Functional Modification: Changing the axis settings (and possibly the smoothness number) may help to clarify the kind of relationship that exists between the data and the current solution; i.e., it can help identify what terms might be applied to the current SPF or Overdispersion function in order to obtain a better fit (a flatter graph). Trendlines on the graph may help suggest modifications to the functions. If the smoothed line seems to follow one of the colored lines, that trendline might correlate with an underlying pattern in the data. The box below the graph suggests corresponding terms that could be appended to the function to

improve fit – if the red trendline fits the data closest, the term in red is most likely to be helpful, etc. In these suggestions, # and $ represent new parameters that would be created.

The multiplicative exponential (red and blue) terms are preferred in general. The red term will work for variables that are always positive and the blue term will work even for variables with zero or negative values (but it can result in functions with a limited range of applicability). The green and yellow terms, or any additive term (starts with "±"), are likely to be numerically problematic (see 4.4.6.6). A graph with a steady curve (regardless of graph settings) can often be well fitted by adding a term in the form "*A^X#*X$^A" but the solution may work very badly for data not very far outside the range that it was fitted to. More complex terms are possible, but each added term will increase the time required for the Solver to run, and also increase the risk that the final solution will be over-fitted.

Some trendlines will not appear on all graph forms because Excel cannot plot them correctly. When the Smoothness setting permits extremely low values (zero crashes or negative overdispersion) to be plotted on a logarithmic graph, or when the X axis is set to logarithmic but the Sort Key contains zeroes, the trendlines can be misleading.

4.4 Entering and Modifying Model Equations

Every SPF consists of two equations, found in the yellow portion (upper left) of the "Model" worksheet. The main one is the Crash Prediction Formula, i.e. the Safety Performance Function *per se*, which predicts crash counts. The other is the Overdispersion Formula, which calculates the Overdispersion value which determines the relative weights of SPF predicted crash counts, and actual recorded crash counts, in the Empirical Bayes regression equation. Both of these functions can include parameters, which represent the unknown numerical constants in the SPF model, and variables representing the data fields for each site. Otherwise these functions are written as Excel formulae (the = sign at the start of the formula is optional; the program will add it if it is omitted).

4.4.1 Safety Performance Function: This function estimates a crash risk for a site based on roadway data (not including reported crash counts); it is the heart of the model. The SPF may include variables, parameters, constants, and Excel functions/operators, but not cell references. It *must* return a positive value for every site; otherwise likelihood cannot be calculated and an error condition will result. An example of a very basic SPF would be "=L*A^#A*T", where "L"=segment length, "A"=AADT, "T"=years of crash data, and "#A" is a functional parameter (* and ^ are Excel operators). If the SPF is left blank, it will be treated as a constant (the same exact crash prediction for every site).

Note: If an explicit variable representing time has not been included, the user will have to modify the final result of the SPF accordingly in order to get it into a "per year" form. If a SPF calibrated to five years of original data were to be deployed using three years of crash data, the results would be embarrassing.

4.4.2 Overdispersion Function: This will typically be less complex than the main crash prediction function; in the minimal case it may be just a numerical constant (i.e., if the user leaves it blank). For highway segments, a minimally complex Overdispersion function is "=?^-0.5", i.e. an inverse proportion to the square root of the crash prediction (the program automatically adds a numerical coefficient). Any greater level of complexity in the Overdispersion function must depend on the needs and judgment of the modeler. This formula returns Overdispersion in the Highway Safety Manual (reciprocal) form, referenced therein as "k" but abbreviated herein as "1/Φ" for consistency with many other published sources.

The Overdispersion function may include variables, parameters, constants, and Excel functions/operators, but not cell references. It may also include the symbol ? (question mark) which represents the result of the SPF calculation. Overdispersion *must* be a positive value for every site. Otherwise, likelihood cannot be calculated and an error condition will result.

Typically the user will not seek much improvement of the Overdispersion function until the SPF has been refined into more or less its final form. See 4.4.6.3.

4.4.3 Weighting: A third equation that can affect the final parameter results, but is not part of the final SPF, is the "weighting" equation (found below the other two equations). This is an Excel formula which may contain variables, constants, and Excel functions/operators but not parameters or cell references. It *must* return a non-negative numerical value for every site. Every site is weighted in the solution proportionally to the value returned by this formula, exactly as if it represented that many different sites each with the same crash count. If it is left blank, all sites (except those excluded by a Filter) are equally weighted in the calculation of the solution, and this is normally the best choice. However, when a large number of sites with identical data (including crash count) are present, using weighting to represent all of them with a single line of data may dramatically reduce the Solver running time. In this case, the weighting formula would simply be a single variable representing the number of (identical) sites.

Note: Weighting always uses the raw values of variables, i.e. without scaling or conversion to positive-only values.

Weighting can interfere with the appearance of the graphs, especially the CURE plot.

4.4.4 Variables: All user-entered formulae can use the variables created when the data are loaded. The name of each variable is the column header of the corresponding data field on the Data worksheet. Whenever that name appears in a user formula, the program will convert it into a reference to the corresponding data field.

The SPF and Overdispersion equations will use the scaled values of variables if those variables' checkboxes have been marked for scaling ("*θ"), unless the variable name is accompanied in the formula by a comparison operator (e.g. ">") or is the first argument of an IF or CHOOSE function. All other equations use the raw, unmodified values of the variables.

4.4.5 Parameters: A parameter is a numerical constant, the value of which is to be estimated by the Excel Solver. The user creates parameters simply by including them in functions, but never gives them values, not even as initial guesses. Only the SPF and Overdispersion equations can include parameters. Parameters are created simply by including an acceptable parameter name in the right place in the equation, exactly as if it were a numerical constant. When a parameter is recognized by the program, it will be highlighted blue in the box containing its parent equation. Parameters are also shown in the blue lower left-central area of the "Model" worksheet.

Removing a parameter from a formula will not immediately remove it from the list, so previously estimated values of that parameter can still be seen. However, if a new solution is run, all unused parameters will be erased. Deleting parameter names from the list could cause the program to fail.

The maximum number of parameters is 100. However, the larger the number of parameters, the longer the Solver will take to run.

4.4.5.1 Naming Parameters: Parameter names may include only letters, digits, and/or the following special characters: ! # $ % @ \ _ ` ~ | [ ]. Parameter names may not consist entirely of digits; they must have at least one letter or special character. Parameter names are not case sensitive; they will be converted into uppercase automatically. Blank spaces are *not* allowed in parameter names. Parameter names may not duplicate variable names; if this is done accidentally, the program will parse the name as a variable. The characters "$" and "#" have a special function in parameter names (see below).

It is possible to include the same parameter in both the SPF and Overdispersion formulae; its value will be the same in both.

4.4.5.2 Constraining Parameters: Every parameter will have a checkbox next to its name in the parameter list, under the heading ">0". If this box is checked, the parameter will be constrained to positive values only. Otherwise, it can take on zero and negative values. A positive parameter may be required depending on how it is used in the equation; for instance in the term "X1^AADT", a zero value of X1 would cause the term (and probably the whole SPF) to equal zero, which would cause an error. A negative value for X1 would cause an Excel calculation error (because AADT is scaled and therefore contains non-integer values). Contrarily, if the ">0" checkbox is set for a parameter that needs to take on a negative value, the Solver will fail or produce unusable results. For instance in the term "MW^X2" (where "MW" = median width), it is nearly certain that the correct value for X2 will be negative. If ">0" is checked for X2, the Solver will try in vain to reach this value by pushing X2 to an infinitesimal positive number.

When a parameter is created, the program will initially decide whether to constrain that parameter to positive values. If the parameter name includes "$", the program will automatically check the ">0" box for that parameter; if the parameter name includes "#", it will automatically uncheck the box. Otherwise it will make a guess. If the program guesses incorrectly, the user must change the ">0" checkbox.

4.4.5.3 Redundant Parameters: The user should be careful not to introduce redundant parameters. One example would a scaling coefficient (because the program automatically adds one internally for the whole function). For instance, if the SPF is Length × Time × AADT^X1 × X2, the final parameter (X2) is redundant. In this case it would probably just hold a value of 1 when the Solver is run (i.e., it would be ignored), but other cases are less obvious and can be more problematic. Suppose for instance that the SPF is Length × Time × (AADT^X1 + AADT^X2 × X3). Here, X3 relates the proportions of the two different AADT terms, but if the AADT^X2 term is overwhelmingly dominant during the Solver run, X3 could drift toward infinity while the internal scaling coefficient drifts toward zero, resulting in a failure.

4.4.6 Common Numerical Issues: Some challenges that not infrequently arise in SPF models include:

4.4.6.1 Segment Length presents a difficult problem for the modeler, in that it is often desirable or necessary for a highway SPF to be directly proportional to segment length, but this may be impossible to achieve a good model fit for, especially if there are few usable variables. This is because segment length (often assigned for planning or inventory purposes) tends to be highly correlated with variables (like driveway density) that are unavailable. The modeler will have to make a decision whether to allow the SPF to vary with segment length other than in a directly proportional manner ("*L").

4.4.6.2 Years of Data can be an issue in a similar way that segments can, in cases where sites have varying numbers of years of data for use. The crash rate "should" be proportional to the number of years but might not be, and the expected relation between Overdispersion and timespan of data is uncertain.

4.4.6.3 Overdispersion cannot be directly fitted because there is no practical way to calculate the Overdispersion at a single point. Accordingly the graph uses point estimates that can easily be negative (and are not plotted correctly on logarithmic graphs), but actual Overdispersion is always positive. Another issue is that the method of solution automatically prefers high Overdispersion at data points where the SPF fits poorly. A solution may tend to sacrifice SPF fit by allowing higher dispersion, especially if the Overdispersion equation is complex enough to allow such distortion. Overdispersion is a very important part of the model but typically it is has poor stability and is quite sensitive to the form of the SPF equation (the reverse is less true). The general recommendation is to fit the SPF first using a basic Overdispersion function, and then introduce only moderate (if any) complexity to the Overdispersion function, avoiding making any changes to the form of the SPF equation while fitting Overdispersion. Using Minimum Squares Regression (instead of likelihood) to fit Overdispersion may produce more stable results.

4.4.6.4 Variables with Zero Values can result in certain equation forms failing even if these appear, from the graph, to provide the best potential fit. The modeler may be tempted to add a value to these variables in order to be able to use the preferred functional form. This may be acceptable, but the amount added should always include a parameter. If the modeler chooses to just add +1 (e.g. LOG(X+1)), they have introduced a parameter which is not properly accounted for in the calculation of BIC and even a value of exactly 1 is arbitrary because all of the variables are always scaled, either internally or externally.

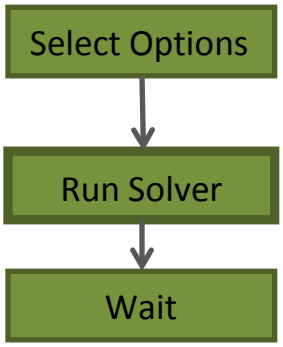
4.4.6.5 Underdispersion occurs when the amount of random variation in at least part of the data is less than ought to be possible assuming that the current model is correct. Underdispersion can occur (through random variation) when the data set is too small; when the data set deviates too much from a negative binomial distribution *given the form of the model*, and therefore when the SPF fits the data poorly; when the model is overfitted (trying to explain purely random variation); or when the Overdispersion equation fits the data poorly (too simple, for example).

The Solver will often fail in these cases because it is trying to push Overdispersion values below zero which the program does not allow.

4.4.6.6 Additive terms may be useful in equations but create some problems:

1. If direct proportionality to time of exposure and/or segment length is desired, appropriate terms must be applied to the added term, and this may contradict the underlying pattern that made the term useful in the first place.
2. Such terms will not work as expected unless they include a coefficient. Usually each such term will need at least two new parameters.
3. If the term needs to take on a negative value to fit the data, it could cause a Solver failure. Constraining it to be positive via parameter constraints will not help.

4.5 Running the Solver



4.5.1 Solution Parameters: Several user-controlled parameters are available to manipulate the optimization of the model. These are found in the green central area of the Model sheet and must be set before the Solver is run.

4.5.1.1 SPF Optimization Method can be set to Maximum Likelihood or Minimum Squares Regression. Running the Solver will then set the parameter values to optimize the chosen metric. Minimum Squares Regression is somewhat faster but can produce inferior results.

4.5.1.2 Overdispersion Method can likewise be set to Maximum Likelihood or Minimum Squares Regression. If both formulae are set to Maximum Likelihood optimization, they will be solved simultaneously. Otherwise, Overdispersion will be solved last.

For Overdispersion, "Minimum Squares Regression" minimizes the sum of the squares of the differences between "observed" and expected variance. "Observed" variance for each site is estimated as the square of the difference between the recorded crash count and the mean crash count predicted by the SPF. This method may produce more stable results than Maximum Likelihood.

4.5.1.3 Solver Intensity determines how persistent the Solver will be in optimizing the model on a scale of 1 to 5. The higher it is set, the longer the Solver will run and the more reliable the results will be. The more complex the model, the more difference there will be between levels of Solver Intensity.

4.5.1.4 The "∑ excess sqr errors ≥ 0" checkbox keeps the Solver from finding a solution with less variance between predicted and actual crashes than would be expected from a negative binomial model. It is off by default. Using it reduces the possibility that a model could be overfitted but can cause the Solver to fail.

4.5.1.5 The "∑ residual x sort key = 0" checkbox is off by default. When it is checked, the Solver solution will be constrained so that there is zero Pearson correlation between the residuals (arithmetic difference between actual crashes and SPF predicted crashes) and the Sort Key (whatever it may be when the Solver is run). This option may be used to reduce serious bias in cases where the

CURE plot is distinctly lopsided (the graph is mostly positive or mostly negative). It will only improve the CURE plot relative to that Sort Key, and otherwise will reduce the quality of fit. It will also increase the chance of Solver failure.

This constraint is most likely to be of use when it is impossible or disallowed to model the effect of a variable properly, e.g. when the SPF is required to vary in direct proportion to the segment length variable.

4.5.1.6 The "∑ residuals = 0" checkbox forces the total number of predicted collisions to match the total number of reported collisions (like a Calibrated model). It is off by default, in favor of the "∑ expected residuals = 0" option.

4.5.1.7 The "∑ expected residuals = 0" checkboxes forces the total number of expected collisions (regression estimate) to match the total number of reported collisions, so that the latter may be considered an unbiased estimate of the former. It is on by default.

Checking both the "∑ residuals = 0" and "∑ expected residuals = 0" boxes can cause the Solver to fail or produce a poor model. In general, using more than one of the four constraint options simultaneously leads to a substantial chance of Solver failure.

4.5.2 Automatic Elements: The spreadsheet handles certain things automatically:

Initial values of parameters are estimated.

Scaling coefficients are added to both the SPF and Overdispersion formulae.

All parameters are internally scaled based on the sensitivity of the Object Function (likelihood or squared residuals) to changes in that parameter. This is more effective than the Solver's autoscaling option but can result in a parameter scaling error if the sensitivity is extremely high or low. The user should only ever see the true (unscaled) values of the parameters unless they inspect the Solver sheet.

Numerical variables, unless the user selects otherwise, are also internally scaled. This means that the numbers the Solver sees are a multiple of the real numbers.

Parameters for which the ">0" constraint has been set can appear negative to the Solver but are converted to positive values for all calculations by the Excel function EXP().

4.5.3 Running the Solver: The "Solve" button initiates model optimization. When a run is initiated, the user is asked whether to save the current Results (i.e., those of the last Solver run). If the user chooses to save these, they will be saved within this spreadsheet using a sheet name that the user will provide.

If there is a Filter in place, the user will be warned. If the user proceeds anyway, the resulting SPF will be valid only for sites that pass the Filter.

Parameters are assigned initial values (using the Solver) before the principal series of Solver runs begins. The user never supplies initial values.

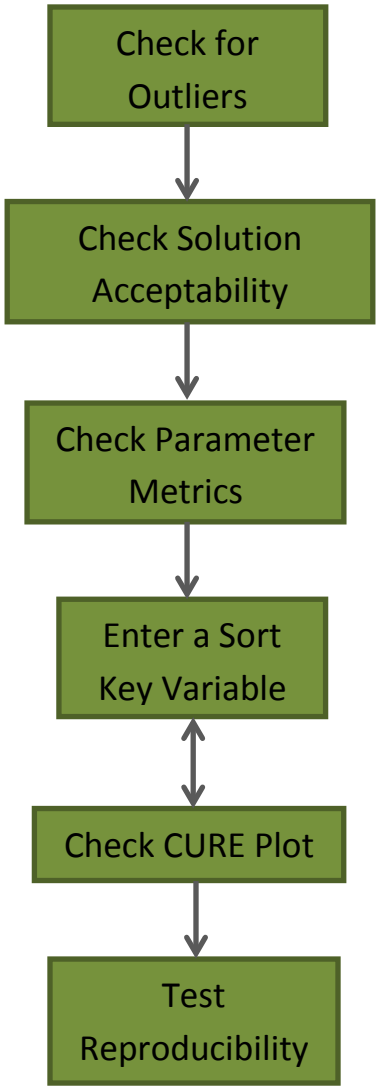
4.5.4 When the Solver Finishes: The user will receive a message. If the solution was completed successfully, the message will be, "Valid Solution Found." Any other message indicates some sort of problem with the results; see 5.5 and 4.4.6. Even if the run is successful, the solution may not be optimal or reasonable; it is "valid" in the sense that it can be used to compute results – the results themselves are not necessarily good.

4.5.5 Limitations of the Excel Solver:

For complex models, Solver could find a solution which is stable but not the ideal solution (i.e. a local minimum); this may be very different from the ideal solution.

Due to very low likelihood gradients for most functions, the Solver may never reach the ideal solution even using Solver Intensity 5. Factors that tend to increase the difference between the final solution and the ideal solution are low Solver Intensity, complexity of the functions, and size of the data set.

4.6 Evaluating Solver Results



Several tools are provided to evaluate the acceptability of a solution and to suggest possible revisions. Ultimately the user must decide whether further refinement of the functions is necessary, desirable, possible, or worthwhile. In some cases, especially when limited to a single explanatory variable (such as AADT), a "good" fit may not be obtainable except by overfitting.

4.6.1 "Outliers" are available at the bottom of the "Results" worksheet. Each line in the "Outliers Detected" results indicates a site which does not fit the SPF very well. Appearance of a site in this list does not necessarily mean that the data for that site are invalid, nor does the failure of a site to appear in the list guarantee that it should not be excluded as an outlier. If the functional form of the SPF is incapable of modeling the data, many sites will necessarily appear as "outliers" because they cannot be modeled. If the functional form of the SPF has been distorted in order to achieve a better fit by accommodating one or two outliers, those outliers may then fit tolerably well within the model.

The "odds ratio" column for the outliers gives the odds against even a single site with such an extreme crash count (normally a high crash count) appearing in the *entire data set* – not the odds against that site having that crash count. Sites with an odds ratio of 2 or 3 may well appear from time to time without being real outliers. An odds ratio of 100 or more virtually guarantees that there is a problem.

Clicking the "Remove Selected Outliers and Reload Data" button will remove any outliers in the list of which at least one cell is selected when the button is clicked. The outlier and its data are removed from the "Data" worksheet and saved in the "Outliers" worksheet, then the data are reloaded.

The ODDS and CURE graphs may also help to find outliers, but it is up to the user to find out what site is causing an anomaly. Then it is necessary to manually remove the outlier from the data and Load Data again.

4.6.2 The user should inspect the results to be sure that the solution is plausible and acceptable. Various metrics are available to aid the user in evaluating models; most are found in the purple lower-central area of the Model sheet.

4.6.2.1 If the Solver ended with the message "Questionable Solution Found", at least one parameter has reached an extreme value. That value should be highlighted blue-green in the parameter list. The SPF *might* still be valid, but probably not; it is a good idea to determine the cause of the overrun and try again. Typical causes could be a parameter set to ">0" when it shouldn't be, redundant parameters, or a functional form which does not match the data.

4.6.2.2 "SPF Minimum Coefficient Of Error" returns a *lower bound* on the error in the model, based on the comparison of actual to expected variance and expressed as the ratio of (minimum) standard error to the SPF prediction for the model as a whole. Actual error in the model could be much higher than this. If the value returned is "N/A", the model has no error that can be proven by variance alone, but it is not necessarily a perfect (or even plausible) model.

This measure is heavily and inversely dependent on Overdispersion, since both reflect variance that is not captured by the SPF.

4.6.2.3 The "Log Mean Likelihood" value is the geometric mean probability, for the sites in the data set, of observing the actual recorded crash counts, assuming that the SPF and Overdispersion equations are correct. If this value is unreasonable, the SPF may be poorly fitted or overfitted. The "Target" value underneath the Objective Value is an approximation of the highest Mean Likelihood that could be achieved by a perfect model. Actual values will usually be somewhat less than this, but can be higher in cases where crash data are very sparse.

4.6.2.4 "Mean Residual" is the average difference between recorded and predicted crashes. It should always be close to zero, but will not usually be exactly zero even if the relevant constraint has been set.

4.6.2.5 "Mean Expected Residual" is the average difference between recorded and expected crashes (regression estimate). It should always be close to zero and will usually be very close to the Mean Residual.

4.6.2.6 The "Mean Distribution Error" is the arithmetic mean of the deviation of the data from a perfect negative binomial distribution, *assuming that the SPF is correct*. Values around 5-10% seem to be typical. This measure will *not* be accurate if Weighting is being used. It will also be distorted if the data have a large proportion of sites with zero crashes.

4.6.2.7 The weighted number of data points (sites) is shown as "Total Weight of Sites". If the number of sites is small, the results may have no validity even if all other metrics are good.

4.6.2.8 The "Log Mean Overdispersion" value is the geometric (i.e. logarithmic) mean Overdispersion of the sites. Moderate or low values are more desirable in general. A value greater than 1.0 could be considered high; a value less than 0.5 could be considered low. High Overdispersion does not necessarily indicate that the SPF is badly fitted; this can also happen because the data do not match the negative binomial distribution very well or because the crash risks are heavily influenced by variables that are not available to the modeler. Likewise, a low value of Overdispersion does not guarantee that the SPF is of good quality; this can happen because the SPF is overfitted, because risk factors not available to or used by the modeler happen to emulate a Poisson distribution, or (in small data sets) by sheer chance.

A SPF with a high mean Overdispersion may still be valid and usable. However, the effect of EB regression will be relatively small for most or all sites. If mean Overdispersion is extremely high, the modeler may consider the possibility that a useful negative binomial model simply cannot be fitted given the available data. If mean Overdispersion is extremely low, especially with only one or two explanatory variables, the modeler should consider the SPF with caution and look carefully at all other measures of SPF quality. Such a model effectively ignores crash history in favor of its own predictions; it had better be an accurate model!

4.6.2.9 The Bayesian Information Criterion (BIC) is an estimate of the model's information loss. The lower it is, the better. A better model fit (higher Objective Value) will reduce the BIC, but adding parameters will raise it. If the net effect of adding a parameter is to increase the BIC, the usefulness of that parameter is questionable. BIC is *not* useful for comparing models developed from different sets of data.

4.6.2.10 The Coefficient of Linear Determination is simply the square of the Pearson correlation between recorded and predicted crash counts; it measures how much of the variation in the recorded crash counts is explained by the SPF.

4.6.2.11 A SPF may be unacceptable if the functional form does not make any sense given the parameter values. This is especially true if the modeler cannot be sure that the SPF will not be used for sites of a somewhat different nature and even more so for sites with characteristics outside (or at the fringes of) the range found in the data set.

For example, it is often useful to introduce a Hoerl term into a SPF; e.g. "*ADT^X1×X2^ADT". If the values for X1 and X2 turn out to be 0.54 and 1.21 respectively (remember that ADT is scaled before these parameter values are applied; the function is NOT taking 1.21 to the power of 56,000!) the resulting influence of AADT on crash projections will be a reasonable curve over a very wide range of AADT. But if the values for X1 and X2 are found for instance as 5.36 and 0.017, the SPF may fit the data very well within most of the range but poorly at the fringes, while failing absurdly for values not very far outside the range of the data set.

The modeler should also be suspicious of relations that make no sense. For instance, it may be found that crash risk increases with increasing median width on divided highways. Such a relationship could exist because wider medians are correlated with higher speeds and/or the absence of median barriers. A SPF embodying such a relationship could be acceptable for some purposes but would be very inappropriate for many others (e.g. comparing one high-speed highway segment with no median barrier to a similar segment with a narrower median).

4.6.2.12 The "±σ/E" values for the SPF and Overdispersion functions (found at the right hand side of the yellow area on the "Model" worksheet which contains these functions) should always be fairly low, since these are coefficients of variation for simple scaling parameters. If either value exceeds 10%, it is likely that something has gone wrong. This can happen when the functional form of the equations is not capable of matching the data, or when the data lack a negative binomial distribution, or when the Solver has failed. NOTE: These coefficients do NOT represent the whole error of the equations.

4.6.3 The modeler may consider the elimination of certain parameters, based on the results, or may wish to try a different functional form. The spreadsheet provides estimates both of the coefficient of error of the parameter values and of the parameter "significance". The modeler may also consider the elimination of outliers.

4.6.3.1 The values of the parameters, found by the Solver, are given in the parameters area under the heading "value". Note that while they are given to several digits of precision, their accuracy is much less than that. If a parameter has taken an unexpected or unreasonable value, especially a value tending toward infinity (e.g. 4.103E98), it may be necessary to alter the entire function. An infinitesimal value (e.g. 0.308E-54) may indicate the same thing, although it is possible in some cases that the problem could be fixed by unchecking the ">0" option for that parameter.

4.6.3.2 The coefficient of error for each parameter is given in the parameters lists under the heading "±σ/E" (standard deviation divided by expected value). It is an approximation only (based on the gradient of the likelihood function). Typical values range from a few percent to perhaps

twenty or thirty percent. High values indicate that changes in the parameter value make little difference to the overall likelihood of the data set. Such a parameter might still be used if it improves the CURE plot substantially, but the modeler should be aware that the value of that parameter is somewhat arbitrary. This situation typically happens when there is a small cluster of sites with extreme values for some roadway attribute that have (among them) an exceptionally high total of recorded crashes. The user should at least examine the Results page for outliers before accepting a parameter with an extremely high coefficient of error.

If the coefficient of error is given as "?", it is likely that the gradient is non-zero at the solution found (this is more likely the more Solver constraints are in place), or that the gradient is discontinuous (this can happen if you are using potentially discontinuous Excel functions like "IF" or "MAX").

4.6.3.3 "Significance" is shown in the parameter areas under the heading "Sig." It is a measure of how much difference the parameter makes to the SPF calculations (not to the likelihood itself). It shows the typical change (see Appendix D for definition) made to each site's calculation (compared to setting the parameter value at default value, which is zero or one depending on whether the parameter has ">0" checked).

A low "Significance" does not always mean that the parameter should be dropped. A parameter with a large effect on a small number of sites can have a low percent "significance" but still have a big impact on the CURE plot (bias). A high "Significance" does not always mean that the parameter is really significant. Two parameters that tend to cancel each other out can each have extremely high "Significance" yet, taken together, have minimal net effect.

A parameter that is used only in the Overdispersion function will always have zero significance. The effect of parameters on Overdispersion is not reported.

"Significance" is by comparison with the case where the parameter takes its default value – i.e. zero or one. This can lead to another kind of case where a parameter shows a misleading high "Significance"; e.g. the terms "*ADT" and "*ADT^A#" are identical when A# = 1, but the program would compare "*ADT^1" to the case "*ADT^0" because zero is the default value for a parameter with "#" in its name.

4.6.4 The modeler should check the functions for bias against all available variables, using the CURE plot graph option. See 4.3.4.3.

If the CURE plot shows a lack of bias, the SPF graph will always show a reasonable fit. However, when multiple variables are used, it may be impossible or impractical to avoid serious bias against some of them. The user may then wish to examine the SPF graph for such a variable before making a decision as to whether the bias is acceptable.

4.6.5 It may also be a good idea to check the Overdispersion graph for extreme variation against at least the most important Sort Keys (e.g. AADT), although some variation in this graph is inevitable.

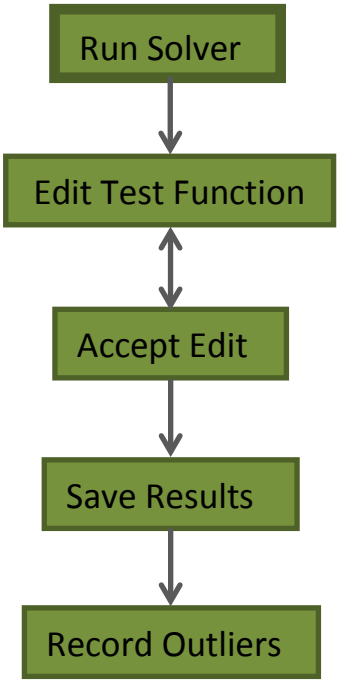
4.6.5.1 The “ODDS” graph may help to determine whether an apparent deviation in the SPF or CURE plot is really significant, to identify regions of poor fit, or to detect outliers. It shows the odds against the observed results, assuming the model is correct, expressed as a standardized score (i.e. 5 on the Y-axis represents 5 standard deviations higher than average). This is an approximation, as the odds do not actually follow a Gaussian distribution.

4.6.6 The “Test Reproducibility” button runs a Monte Carlo type simulation to help evaluate the stability of the parameter values. For reach run, a new set of crash data is generated based on the assumption that the “expected” crashes calculated by the model for each site is an exact measure of the mean of Poisson distributed crash counts at that site, and the model is then solved using the same function form and the same solution settings. The average root mean square error is shown in the Results sheet as “RMS Error on Replication”, expressed as a coefficient of error for each parameter and coefficient.

The user can specify any number of runs. If the number is large enough, the results may be taken as an accurate reflection of the true standard error of each parameter and coefficient. However, this may take an extremely long time to run. Even a single run, however, may help identify parameters with unstable values.

4.6.7 The “Parameter σ Estimate” button uses a more accurate method to estimate the standard errors of the parameter values than the default method used for Solver runs. However, it is much slower.

4.7 Finalizing the Results



When the form of the model has been definitely established, the user will probably want to run a solution with "Solver Intensity" set to 5 (or as high as time permits). The results of the last Solver run are summarized in the "Results" worksheet. Information about the run process is recorded in the "Log" worksheet. Outliers which were deleted by the user using the "Remove Selected Outliers" button are saved in the "Outliers" worksheet.

High Solver Intensity can sometimes cause a previously successful model to fail. This is because the original solution was not an optimum; the Solver halted prematurely. A mathematically optimized solution isn't always a better solution.

4.7.1 The user may wish to remove low-signficance parameters before making the final Solver run, and/or to freeze the value of parameters (especially in the Overdispersion equation). When the run is complete, it may be desirable to calibrate the model (or just the Overdispersion).

4.7.2 The Results sheet includes information about any Filtering or Weighting applied, the final form of the SPF and Overdispersion function, the parameters, and the performance metrics, which is the same as on the Model sheet. It also records the Solver convergence criterion used and the number of times the Solver ran to reach a stable result (the program does this automatically). The Results sheet also has an area (yellow cells labeled "Trial SPF" and "Trial 1/ϕ") where the user can test arithmetic simplifications of these equations.

The values, standard deviation, coefficient of error, and significance of all parameters are listed

below the equations section. "Cspf" is the scaling coefficient for the SPF and "Cdsp" is the scaling coefficient for the Overdispersion function.

4.7.2.1 The functions in the light yellow cells in “Results” are given with the parameter names replaced by their true numeric values and the scaling constants applied to the variables. These functions can be edited by the user to simplify them (mainly by combining the scaling constants with the parameter values) and to reduce the number of digits of precision.

The darker yellow cells to the right of each function editing area will show the *maximum* error introduced to the calculation for any site by any editing of the formulae. If the error exceeds 1% the cell will be highlighted orange. If the error exceeds 10% or the editing produced an error, the cell will be highlighted red. Neither of these limits constitutes a recommendation as to the amount of error that is acceptable. The intrinsic error of an SPF is not calculated by this program but will not be less than indicated by the “SPF Minimum Coefficient of Variation” (4.6.2.2) and may very well exceed 10% in all cases.

Note that if an edit to a function needs to be reversed, “undo” is not available. If the user cannot remember what the previous state was, it will be necessary to copy the original function from the area above and paste it into the light yellow cells, and all edits will be lost. The user should make edits in stages and save the spreadsheet between stages to avoid losing work.

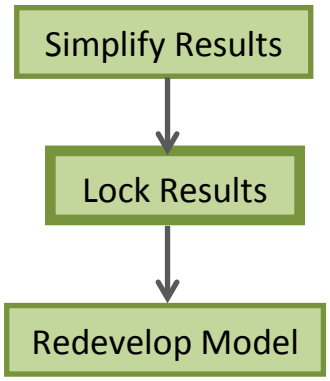
4.7.3 The “Save These Results” button in the Results worksheet will save the information as a new user-named sheet in the workbook. As of version 3.5, the results *cannot* be restored for any further work, including using the “Trial” function area in the Results sheet to simplify the equations.

4.7.4 If any outliers were removed using the “Remove Selected Outliers…” button, they will be saved in the Outliers worksheet until the Clear Data function is used. Users may wish to document the removal of outliers from their data sets.

Outliers may also be detected by other methods, such as the presence of sudden vertical “cliffs” in a CURE plot, and directly deleted from the data set. Such removal of outliers is not recorded in the Outliers sheet.

4.7.5 If any record is to be made of the CURE plots, it must be done manually one by one.

4.8 Multi-Stage Safety Performance Functions (Optional)



4.8.1 The tool has two built-in functions to assist with developing SPFs in stages. The usual application would be a case where crash counts of non-target crash types are being used as a surrogate for unknown roadway attributes; for example, using the total counts of non-fatality crashes may improve the prediction of fatality crashes at intersections where minor approach ADT is unknown.

The method works by locking in the current parameter values and using one of the outputs of the model (Predicted Crashes or Expected Crashes) as a variable in a second model. Since non-target crash counts would also correlate with the known roadway attributes, if a model was developed in a single stage the non-target crash counts could displace known roadway attributes in the optimized model and result in a form that, while superficially having better metrics, will perform poorly.

If any simplification of the model is to be done (4.7.1.1) this must be done before using either locking function.

There are two implementations of this method:

4.8.1.1 The Lock Current Npred button creates a new variable that, for each site, represents the "predicted" crashes of the current (primary) model (which should use only roadway variables). This variable is scalable, but unscaled by default. The secondary model then developed should not use any variables other than the new one created and variables representing inferior, indirect measurements (i.e. counts of non-target crashes). Ideally the secondary model will take the form *Crash Prediction = X*f(Z)* where *X* is the crash prediction of the first model and *Z* is a count of non-target crashes. Otherwise the application of the model will be ambiguous for most cases other than network screening.

The variable(s) representing non-target crash counts must be present from the beginning, but not used until the secondary model is developed. The target crash counts are the same in both modeling stages.

4.8.1.2 The Lock Current Nexp button creates a new variable that, for each site, represents the "expected" crashes of the current model (i.e. the regression estimate). This variable is also scalable but unscaled by default. With this method, any roadway variables can be used in both stages of the model. However, the result will be substantially more complicated to apply since it requires two regression estimates each time the model is used in practice.

For this method, the primary model must use the non-target crash counts as the crash count, not a variable (i.e., this data should start out in the right-most column of the Data sheet). The target crash counts must be included as a variable, but not used in the primary model. When the secondary model is initiated, the user will tell the system which variable is to become the new crash counts.

4.8.2 Concerns and Limitations:

When a model is developed using the first method, the number of years of data used for the non-target crashes must be fixed, i.e. any application of that model would require the same number of years of crash data. Even if the non-target crashes are expressed as a frequency, the change in relative variability of this measure with changing numbers of years could invalidate the model.

Non-target crashes are only a random reflection of the unknown roadway attributes. The usual implementation of the negative binomial model neglects errors in the model itself for estimation both of the regression estimate (expected crashes) and the standard error of that estimate. Models using this method may produce stronger regression to the mean than would be optimal, and will definitely underestimate the standard error of regression estimates (expected crashes). Solution metrics for models that use this method are not necessarily comparable to solution metrics for conventional models.

It is possible to use more than two stages to develop a model, but this is unlikely to be of any utility.

5.0 Troubleshooting

The following error messages are Excel artifacts and should be ignored:

- **"Excel found a problem with one or more formula references in this worksheet"**
- **"Some trendlines cannot be calculated…"**
- **"Negative or zero values cannot be plotted correctly on log charts"**

5.1 **Pink cells**: If a cell containing an equation is highlighted dark pink, that equation contains or has caused an error. Some possible sources of error:

- Typos and parsing problems such as…
  - Misplaced, missing, or superfluous parentheses
  - Missing operators (+, *, ^, etc.)
  - Mistyped or misused Excel functions
  - Variables used that were not included in the data
  - Mistyped variable names
  - Illegal parameter names
  - Trying to give a parameter the same name as a variable
  - Missing "" marks for text variables
  - Using the letter "E" in a numerical constant (scientific notation)
- Illegal values caused by…
  - SPF or Overdispersion formulae that produce negative or zero values for any site for any possible values of the parameters. This is typically caused by using a multiplicative term capable of having a non-positive value (*LOG(ADT) for example).
  - Allowing zero or negative terms where positive terms are required, for instance in the expressions LOG(ADT*X) or X^ADT, the parameter X must be set ">0" or the formula will return an error value.
  - Operations that are invalid for some sites, e.g. Shoulder Width ^ parameter where some sites have Shoulder Width zero.
  - Terms tending to infinity, usually detectable by the fact that one or parameters have extreme values. This typically means that redundant parameters have been included or that the functional form of the SPF equation is not capable of representing the data.
  - Using numerical operators on text data, or vice versa.
  - Accidentally re-using the same parameter name in different parts of an equation.
  - Unscaled variables resulting in extreme calculations, e.g. 1.46^AADT is being calculated when (internally) it should be 1.46^(AADT*0.00023).
- If the Filter has failed to pass any sites, the Filter will be highlighted dark pink (and there will be other error messages).
- Failure in one equation can cause other equations to fail.
- Defects in the data, such as missing or inappropriate values, can cause equation failure.

- Excel 2016 contains a bug that can cause macros to randomly lose the ability to read and write checkboxes. This can cause parameters to be incorrectly flagged as to whether they are constrained to be positive, even if they were correctly constrained before. If resetting the parameter flags becomes a nuisance, rebooting Windows may stop the problem (for a while).

Pink cell coloring can sometimes persist even after the cause has been corrected. Possible solutions: Click the "Refresh" button to recheck for errors. Click the "Clear Values" button to reset the values of any parameters which have gone toward infinity. If that fails, use the "RESET" button to rewrite all internal formulae. Note: Both "Clear Values" and "RESET" will obliterate the current solution.

5.2 **Orange Cells:** The "Filter" title cell will be highlighted orange if the Filter formula is not blank. This is not an error but is intended to help the user remember that only part of the data set is being considered.

5.3 **Gray Cells**: Certain results in the "Solution Metrics" area are calculated by programs rather than formulae and so do not always represent the current state of the model. When these cells are colored gray, their contents are accurate for the most recent solution only.

5.4 **Question Marks**: If the coefficient of error ("±σ/E" for a parameter or equation is shown as "?", it means the programs was not able to estimate that coefficient. It does not necessarily mean that the results are invalid or that the parameter is not useful. Usually this happens because the likelihood gradient at the parameter value found is not close enough to zero; i.e., the solution found is not a true local likelihood maximum (this is likely when the optimization method was not Max Likelihood) or when multiple solution constraints are used. It can also happen when a discontinuous function is used, or when a parameter has no effect on the solution. Using Maximum Likelihood optimization, running the model with higher Solver Intensity, unchecking one or more solution constraints, and/or removing functions that use comparisons (MAX, >, <, etc.) may result in a "±σ/E" value being calculated. This may or may not be desirable.

5.5 **Error Messages**: There are three likely sources of error messages – this program, Excel VBA, and Excel Solver.

5.5.1 VBA and Excel error messages are usually caused by an overflow, and this is usually caused by a parameter trending toward infinity.

- VBA runtime errors may be reported by the program as "Run Error…" giving the error number, error description, and possibly the subroutine in which the error occurred. Bugs in the program could result in VBA errors, but the most likely cause is an overflow caused by a parameter nearing infinity or zero. The reported error type in this case may be a "type mismatch" as VBA attempts to perform a calculation on an error code returned by an Excel formula.
- **"Negative or zero values cannot be plotted correctly on log charts," "Excel found a problem with one or more formula references in this worksheet," and "Some trendlines cannot be calculated…"** are caused by defects in Excel during the process of updating the graph, and **should be ignored**.

5.5.2 Error messages thrown by this program include:

- "Column Overflow: Data Load Aborted" More than 97 columns of data were found in the Data sheet, or more than 96 if you did not designate a column as the site identifier.
- "Data Load Aborted" By itself means that the data load was terminated by the user.
- "Data Points are Underdispersed" The program can still be run and may produce a solution, but its value may be questionable as the data are unlikely to be negative binomially distributed. This error is most likely when the number of sites passing the Filter is too small to be statistically meaningful.
- "Defective Field Label: Data Load Aborted" An error code was found in the header for the first column, which by default is used as the Site Identifier column.
- "Defective Variable Name…" Either the data variable contained illegal characters, or was an error code, or the program misread something as a variable name that shouldn't have been. In the former cases, this error is recoverable. See 4.1.3 for legal variable names.
- "Defective… Function" The function could not be understood, or returned an error code, text, or otherwise unacceptable value for at least one site. Unacceptable values include zero or negative numbers for SPF and Overdispersion, negative values for Weighting, and any Filter that does not pass at least one site.
- "Duplicate Variable Names Detected, Data Load Aborted." More than one column has the same header. This can happen even if the columns started out with different names, because unusable characters are stripped out of them before they are used and variable names are not case sensitive. For example "g" and "G+" will both be turned into "G".
- "Field… Contains Blanks…" This is a warning to the user that data *might* be missing, but in many cases blank data are acceptable. The user may choose to continue.
- "Field… Contains Errors. Data Load Aborted". Error codes have been found in a field (data column). This might be caused by pasting formulae into the Data sheet instead of values.
- "Field… Contains Negative Values…"
- "Field… Contains Non-Numerical Data…" This is a warning to the user that data is not in the expected format. It could be numbers formatted as text, or a column with text data that the user forgot to format the whole column as text. The user may choose to continue, but in most cases it will be necessary to abort and fix the data.
- "…Function cannot be understood…" The parsing engine was not able to translate the function into an acceptable Excel formula. The most likely causes are typos (e.g. missing parentheses, illegal parameter names).
- "…Function has generated errors… or other unacceptable values" The formula was understood, but didn't work. See "Illegal values caused by …" in 5.1.
- "Insufficient Data: Data Load Aborted" An attempt was made to load data but there are none in the Data sheet, or no headers, or just one column.
- "Invalid Results" The Solver run has failed to produce a meaningful result. The reason should have already been given by previous error messages.
- "Missing Field Label: Data Load Aborted" Similar to "Defective Field Label".

- "No Data Found" Could occur if an attempt was made to run a solution without first loading the data.
- "No data to graph" Most likely the Filter has failed to pass any sites.
- "No Sites Selected: Check Filter" Make sure the Filter is not blocking every site in the data, then click "RESET".
- "Over 1,000 unique non-numerical values were detected in…" Probably, the named field consists of numerical values which were accidentally formatted as text. If you really do have a text field with that many different values in it, ignore this message.
- "Questionable Solution…" The Solver run technically produced a solution, but it may be worthless. See 4.6.2.1.
- "Solver Failure… on Parameter Initialization" The Solver has failed while trying to set the initial values of the parameters before a main solution run. This almost always means that the main Solver run will fail as well. See Solver Error Messages (5.5.3) and User Tools (Appendix A).
- "SPF/Dispersion Parameter…/Coefficient reached an unexpected low/high value" If this is a *low* value, it is usually because a parameter is constrained to be >0 but the solution gets better as the parameter value is pushed infinitely close to zero. If the ">0" constraint was not accidental, it will be necessary to choose a different functional form for the equation. If the error message indicates a *high* value, it usually means the functional form is not working. This error will almost always indicate that the results should not be used.
- "Underdispersion Encountered" The coefficient of the Overdispersion equation was found to be less than 3.72×10^{-44} on a Solver Failure 9. See 4.4.6.5.

5.5.3 Solver Error Messages are returned by the program along with the Solver return code. The more likely Solver failure modes and causes are:

- Solver Failure 5: Solver could not find a feasible solution. This error is most likely if one or more constraint options is checked, but can also happen with an SPF or Overdispersion function that does not match the data well.
- Solver Failure 6: Solver stopped at user's request.
- Solver Failure 7 is returned by this program, not the Solver (an actual Solver failure 7 is impossible using this spreadsheet). The solution was found to be diverging on several consecutive runs and terminated instead of running until it produced an error due to overflow.
- Solver Failure 8: The problem is too large for the Solver to handle. Possibly caused by having an excessive number of parameters.
- Solver Failure 9: Solver encountered an error value in a target or constraint cell. This can be caused by a formula that results in division by zero, taking the logarithm of a negative number, etc. but is most often caused by underdispersion or by a parameter tending toward infinity.
- Solver Failure 10: Stop chosen when the maximum time limit was reached. This might conceivably appear using Excel 2007.
- Solver Failure 11: There is not enough memory available to solve the problem.

- Solver Failure 13: Error in model. This would probably indicate a breakdown in this program, such as a change made to the "Controls" sheet.

See also 4.4.6 for a description of common computational issues, some of which can be associated with Solver failures.

5.6 Program Crash: If normal functions become unresponsive (probably after a program crash), click the "Refresh" button. If the program is completely locked up, hold down the Escape key to halt the program and (if this works) then use RESET. Otherwise the Windows Task Manager must be used to stop Excel.

In versions of Excel prior to 2016, it is possible to run this spreadsheet simultaneously with another, by starting a different instance of Excel. This will not work in Excel 2016.

At certain points in the solution process, the status bar will say "Ready" but Excel remains unresponsive. Usually this does not mean that Excel has crashed but that the Solver has failed to take over the status bar even though it is running.

5.7 Parameter Failure: If an error was caused by a parameter going to infinity (e.g. Solver Failure 9, Parameter Scaling Failure, program crash) the parameter(s) that failed may have their names highlighted in blue-green on the Model sheet.

Parameter failure usually requires a change in the functional form of the SPF to correct. In rare cases, the automatic scaling of a parameter will be inadequate; putting a numerical constant into the function should fix this.

5.8 Graphing Problems:

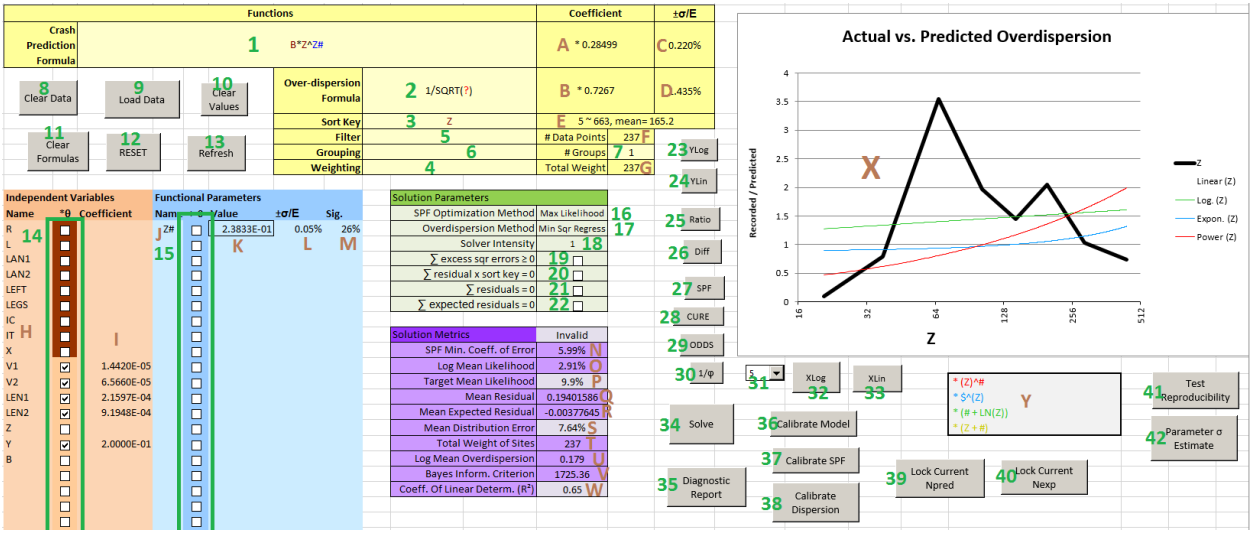
- Graph is grayed out or X-axis labels are smeared:
  - If the graph turns gray, but none of the functions are producing errors, click the Refresh button. If this does not work, try changing the graph settings.
- Little or nothing appears on graph:
  - The smoothing value is set too high.
  - Too few sites are passing the Filter.
  - Sort key has produced an error (Sort Key cell is pink).
  - Sort key is producing unexpected results without throwing an error.
  - Axis settings are not appropriate for the data.
  - Graph is not responding to changes, e.g. axes have failed to re-scale.
- Graph fails to respond to changes:
  - Click the "Refresh" button. If that fails, click "RESET".
- Graph appears distorted:

- Using a logarithmic X-axis ("XLog") with a Sort Key that includes zeroes (such as, in many cases, shoulder width) will cause the values at Sort Key = 0 to be stretched over a large part of the graph. Note that the X-axis may or may not be changed from logarithmic to linear automatically in this case.
- A Sort Key that reflects groupings (as opposed to a continuous scale of numbers) can cause points to appear that are between groups on the X-axis, as the smoothing algorithm takes the numerical average of data that aren't really numerical.
- A defective SPF or Overdispersion Function can cause extreme (or extremely flat) results to be graphed, because the graph shows the relation between the data and the function.

5.8 The "**Log**" Worksheet contains a record of the most recent solution process which may occasionally be helpful. Cells A1 and B1 are used to tell the program where to write the next line to the Log and are not useful to the user. Cell A2 should contain the date and time the run was initiated, and the last cell should contain the date and time the run was completed. Other events in the process are recorded line by line. The Solver itself does not log anything and the line "Solver Return Code = …" indicates that the program ran the Solver, which returned the indicated code. The return code for a normal successful solver run is 0.

5.9 **Hidden Sheets:** Users who are confident Excel experts may benefit from viewing certain hidden sheets to diagnose problems (usually Solver failures). See Appendix B.

Appendix A: Quick Reference to Items in Model sheet



Controls and Inputs

1. **Safety Performance Function:** The user-created predictive crash model.
2. **Overdispersion Function:** The user-created Overdispersion equation corresponding to (1).
3. **Sort Key:** The variable against which comparisons and evaluations are to be made.
4. **Weighting:** (Optional) A function that can treat some data lines as representing multiple sites.
5. **Filter:** (Optional) A Boolean formula determining which sites will be included in analysis.
6. **Grouping:** (Optional) A variable by which data are divided into groups on the graph.
7. **# Groups:** Controls the number of groups to be plotted on the graph.
8. **Clear Data:** Empties the Data sheet and prepares it for a new data set.
9. **Load Data:** Copies a data set from the Data sheet into the calculation area.
10. **Clear Values:** Clears all parameter values to clear possible errors.
11. **Clear Formulas:** Clears formulae in (1) – (6).
12. **RESET:** Redraws the graph and rewrites internal calculations to clear possible errors.
13. **Refresh:** Corrects some error conditions relating to parameters and functions and reactivates macros after a crash.
14. ***θ**: Determines if variable values are to be scaled.
15. **>0:** Determines if parameters are to be constrained to take positive values only.
16. **SPF Optimization Method:** Selects the object function for the Solver.
17. **Overdispersion Method:** Selects the object function for optimizing Overdispersion.
18. **Solver Intensity:** Determines how hard the Solver will work to optimize the solution.
19. **∑ excess sqr errors ≥ 0:** An optional constraint that prevents solutions from having less variance than the theoretical optimum.
20. **∑ residual x sort key = 0:** An optional constraint that forces solutions to have zero Pearson correlation between the residuals and the sort key.
21. **∑ residuals = 0:** An optional constraint that forces the sum of predicted crashes to match the sum of actual crashes.

22. **∑ expected residuals = 0:** An optional constraint that forces the sum of expected crashes to match the sum of actual crashes.
23. **YLog:** Sets the graph Y axis to logarithmic scale.
24. **YLin:** Sets the graph Y axis to linear scale.
25. **Ratio:** Sets the graph to display residuals as the ratio of actual to predicted.
26. **Diff:** Sets the graph to display simple residuals.
27. **SPF:** Makes the graph display the crash residuals.
28. **CURE:** Makes the graph display the standardized cumulative crash residuals.
29. **ODDS:** Makes the graph display the standardized odds ratio against the crash counts.
30. **1/φ:** Makes the graph display the Overdispersion residuals.
31. **Smoothness:** Controls the smoothing of the graph.
32. **XLog:** Sets the graph X axis to logarithmic scale.
33. **XLin:** Sets the graph X axis to linear scale.
34. **Solve:** Runs the Solver to optimize the model.
35. **Diagnostic Report:** Shows a diagnostic page to help track down errors.
36. **Calibrate Model:** Sets the SPF and Overdispersion coefficients without changing any user parameters.
37. **Calibrate SPF:** Sets the SPF coefficient only.
38. **Calibrate Dispersion:** Sets the Overdispersion coefficient only.
39. **Lock Current Npred:** Turns the current crash predictions into a variable for use in a second modeling stage.
40. **Lock Current Nexp:** Turns the current crash regression estimates into a variable for use in a second modeling stage.
41. **Test Reproducibility:** Uses a Monte Carlo method to assess the stability of model parameters.
42. **Parameter σ estimate:** Attempts a more accurate calculation of the standard error of parameter values.

Information and Outputs

A. **SPF Coefficient**: The value of the scaling coefficient that is automatically applied to the SPF to make total predicted crashes equal total recorded crashes.
B. **1/φ Coefficient**: The value of the scaling coefficient automatically applied to the Overdispersion formula to achieve maximum likelihood.
C. **±σ/E:** The coefficient of error of the estimated value of output (A).
D. **±σ/E:** The coefficient of error of the estimated value of output (B).
E. **Sort Key Range**: The range of the current Sort Key values.
F. **# Data Points**: The number of data points that passed the filter, ignoring weights.
G. **Total Weight**: The sum of the Weighting formula for all sites that passed the filter.
H. **Name**: Names of data variables, taken from the Data sheet during loading.
I. **Coefficient**: The scaling coefficient being applied to the values of that variable.
J. **Name**: Names of parameters identified in the SPF and Overdispersion functions.
K. **Value**: Values of parameters found by Solver.
L. **±σ/E:** Coefficient of variation of the parameter values found.
M. **Sig.**: Root-mean-square of the proportional difference that the presence of this parameter makes to predictions for sites.
N. **SPF Minimum Coefficient of Variation**: The minimum possible coefficient of error of the current model, based on excess variance.
O. **Log Mean Likelihood:** The geometric mean of the likelihood for all sites.

P. **Target Mean Likelihood:** Estimated log mean likelihood of a perfect model.
Q. **Mean Residual:** Average difference between actual and predicted crashes.
R. **Mean Expected Residual:** Average different between actual crashes and regression estimates.
S. **Mean Distribution Error:** The arithmetic mean of the deviation of crash counts from what they would be if the data perfectly fit a negative binomial distribution given the assumed functional model.
T. **Total Weight of Sites:** Same as Total Weight (G).
U. **Log Mean Overdispersion:** The geometric mean of the calculated Overdispersion value for all sites.
V. **Bayes Information Criterion:** The BIC for the current solution.
W. **Coefficient of Linear Determination:** R^2 value for SPF relative to crash counts.
X. **Graph:** Shows relationships between actual data and the model predictions. May show Pearson correlation between current Sort Key and residuals, and/or trendlines.
Y. **Possible Terms:** Shows terms that might be added to model if they fit the graph.

Appendix B: Normally Hidden Sheets

The **Calcs** worksheet contains most of the working calculations. The first 99 columns are reserved for scaled site data, the second 100 columns for raw input data. The Safety Performance Function and Overdispersion calculations are in columns GR and GS, and the scaled values in columns GT and GU. Error links from the Report sheet will usually lead to one of these columns. The following block of columns is for calculations of likelihood, cumulative residuals, and other values dependent on the model. Column IP is the weight of each data point, IQ is the sort key, IR is the filter, IS holds random numbers for sorting, IT is the Sort Key, IU is the Group number, and IV is the fixed part of the likelihood for each row. Columns IF to IO may be used temporarily by various utilities. Columns past IV are not used because this tool was originally developed for Excel 2003.

The **Solver** sheet is where the Solver actually runs from, and contains both scaled and unscaled parameter values, coefficients, metrics, object functions, and other values for secondary operations.

The **Controls** sheet contains a variety of constants and miscellaneous calculations. This is where the final scaling coefficients for the SPF and Overdispersion equations are located, as well as the formulae that detect errors in user input functions.

The **Scale** sheet handles calculations during the scaling of variables and parameters and is used to test input functions for certain errors.

Sheets **G1**, **G2**, etc. sheet apply the smoothing algorithm and feed the graph. Each sheet processes data for one data group, i.e. one series on the graph. The graph reads from columns HL to HR which in return read from other columns in the sheet depending on the graph settings. In general each block of columns in each sheet except for the first and last blocks represents some particular measure, at smoothing levels 0 – 10.

The **Smooth** sheet is used to calculate the number of points plotted on the graph for each data group.

Sheet **FFF** and **Temp** are used for certain transient internal calculations.

Appendix C: Example of SPF generation

The following example can be recreated using the data in the "Example Data" sheet (version 2.14). The results may differ slightly from what is shown here.

For this example, the data set consists of sub-sections of 2-lane rural highway, with 5 years of non-intersection injury crashes. The following has been pasted into the "Data" worksheet after using "Clear Data" in the "Model" worksheet (the columns have been resized to show the whole column headers).

	A	B	C	D	E	F	G	H	I	J
1										
2		Segment Code	Length	ADT	Terrain	Shoulder Type	Shoulder Width	KAB		
3		01020000	0.44	1700	2	1	8	0		
4		01020044	7.01	1800	2	1	4	9		
5		01020745	4.11	4300	2	1	4	10		
6		01040000	0.26	5500	1	3	5	0		
7		01040059	5.86	4500	2	3	5	32		
8		01040645	0.53	4000	2	1	8	1		
9		01040698	2.18	3500	2	3	4	5		
10		01040916	2.35	4100	2	3	4	3		
11		01041156	0.2	5000	2	3	4	0		
12		01060200	1.6	4700	3	3	5	5		
13		01060360	4.87	4900	3	3	4	10		
14		01060928	0.62	6500	3	3	4	2		
15		01080000	1.45	920	3	3	3	1		
16		01080145	0.56	1100	3	1	8	1		
17		01080201	0.99	1100	3	3	3	1		
18		01080300	2.84	1100	3	3	3	5		

We decide to replace the Shoulder fields with one field which includes only the width of Type 1 (paved) shoulders (zero for other types). Because this is a calculated field, we then copy the column and "paste special" back over itself as "values", so there are no formulae in the "Data" worksheet.

	A	B	C	D	E	F	G	H	I
1									
2		Segment Code	Length	ADT	Terrain	Paved Shoulder	KAB		
3		01020000	0.44	1700	2	8	0		
4		01020044	7.01	1800	2	4	9		
5		01020745	4.11	4300	2	4	10		
6		01040000	0.26	5500	1	0	0		
7		01040059	5.86	4500	2	0	32		
8		01040645	0.53	4000	2	8	1		
9		01040698	2.18	3500	2	0	5		
10		01040916	2.35	4100	2	0	3		
11		01041156	0.2	5000	2	0	0		
12		01060200	1.6	4700	3	0	5		
13		01060360	4.87	4900	3	0	10		
14		01060928	0.62	6500	3	0	2		
15		01080000	1.45	920	3	0	1		
16		01080145	0.56	1100	3	8	1		
17		01080201	0.99	1100	3	0	1		

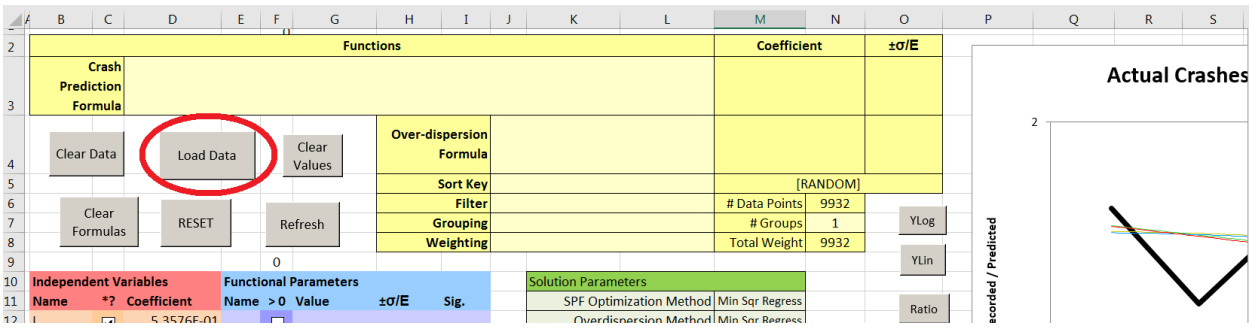
Next we insert a column for Years. The number of years of data is 5 for every site. Note that we do not put the inserted column to the right of the crash counts!

	A	B	C	D	E	F	G	H	I	J
1										
2		Segment Code	Length	ADT	Terrain	Paved Shoulder	Years	KAB		
3		01020000	0.44	1700	2	8	5	0		
4		01020044	7.01	1800	2	4	5	9		
5		01020745	4.11	4300	2	4	5	10		
6		01040000	0.26	5500	1	0	5	0		
7		01040059	5.86	4500	2	0	5	32		
8		01040645	0.53	4000	2	8	5	1		
9		01040698	2.18	3500	2	0	5	5		
10		01040916	2.35	4100	2	0	5	3		
11		01041156	0.2	5000	2	0	5	0		
12		01060200	1.6	4700	3	0	5	5		
13		01060360	4.87	4900	3	0	5	10		
14		01060928	0.62	6500	3	0	5	2		
15		01080000	1.45	920	3	0	5	1		

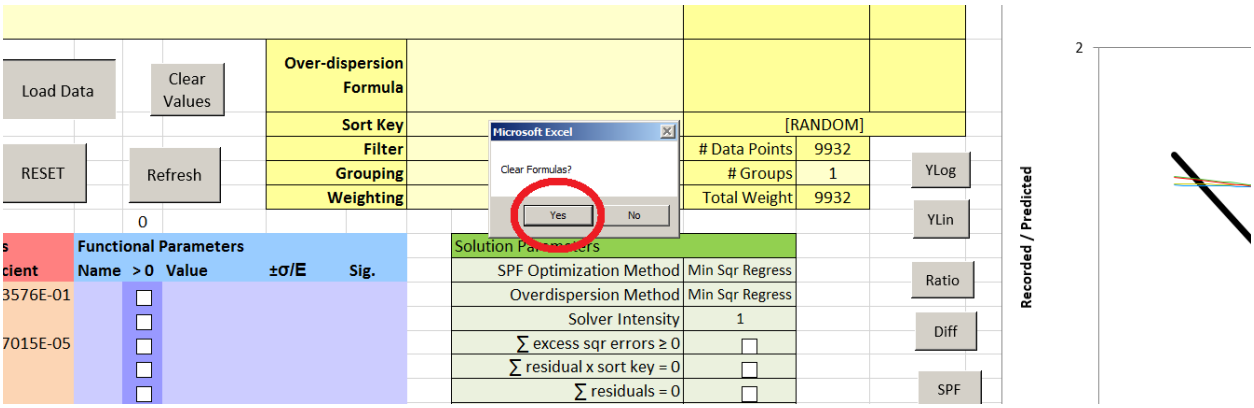
Now we rename all of the columns to short, legal variable names. We don't need to rename the crash counts column or the Segment Code column because they are not variables. We also format column E as text, because Terrain is a look-up code instead of a scalar variable.

	A	B	C	D	E	F	G	H	I
1									
2		Segment Code	L	A	T	S	Y	KAB	
3		01020000	0.44	1700	2	8	5	0	
4		01020044	7.01	1800	2	4	5	9	
5		01020745	4.11	4300	2	4	5	10	
6		01040000	0.26	5500	1	0	5	0	
7		01040059	5.86	4500	2	0	5	32	
8		01040645	0.53	4000	2	8	5	1	
9		01040698	2.18	3500	2	0	5	5	
10		01040916	2.35	4100	2	0	5	3	
11		01041156	0.2	5000	2	0	5	0	
12		01060200	1.6	4700	3	0	5	5	
13		01060360	4.87	4900	3	0	5	10	

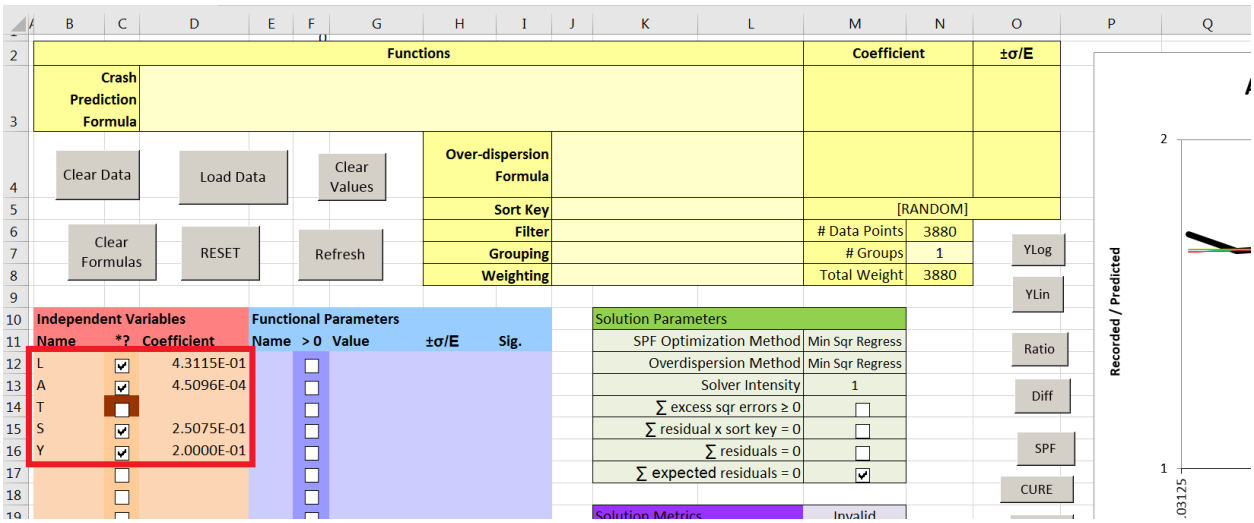
After returning to the "Model" sheet, we click "Load Data".



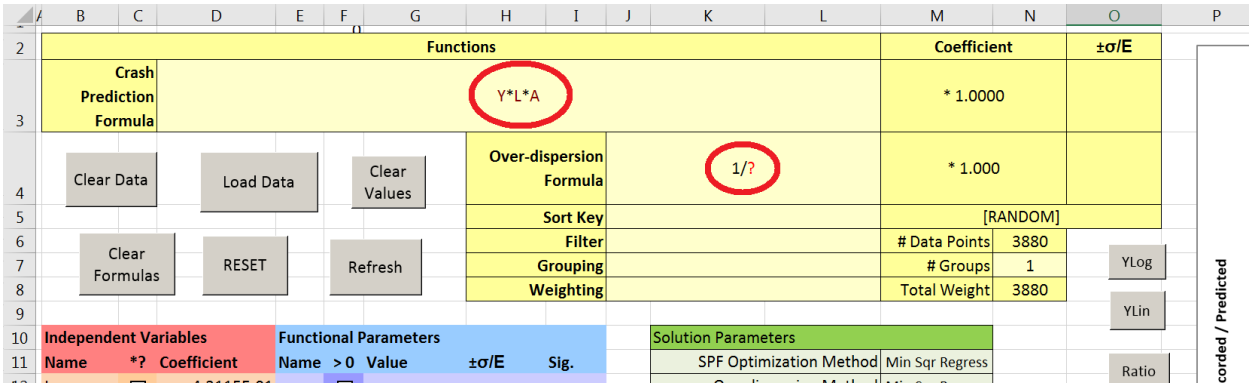
For the “Use Segment Code as Location Identifier…” prompt we click “Yes” because Segment Code uniquely identifies each site in the data and is not relevant as a variable.



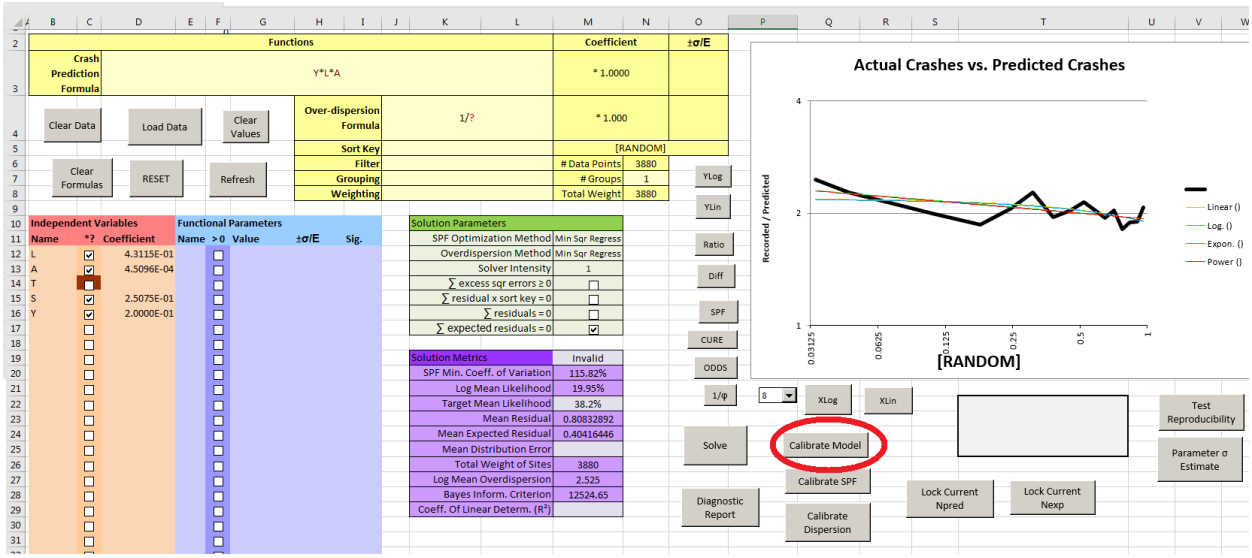
We have the option of clearing any formulae that might be in the Model sheet from earlier work. Note that those variables which are scalable have been automatically checked for scaling. Terrain (T) is marked in brown because it is not scalable (if we had not formatted it as text, it would be scalable because the text is all numerals).



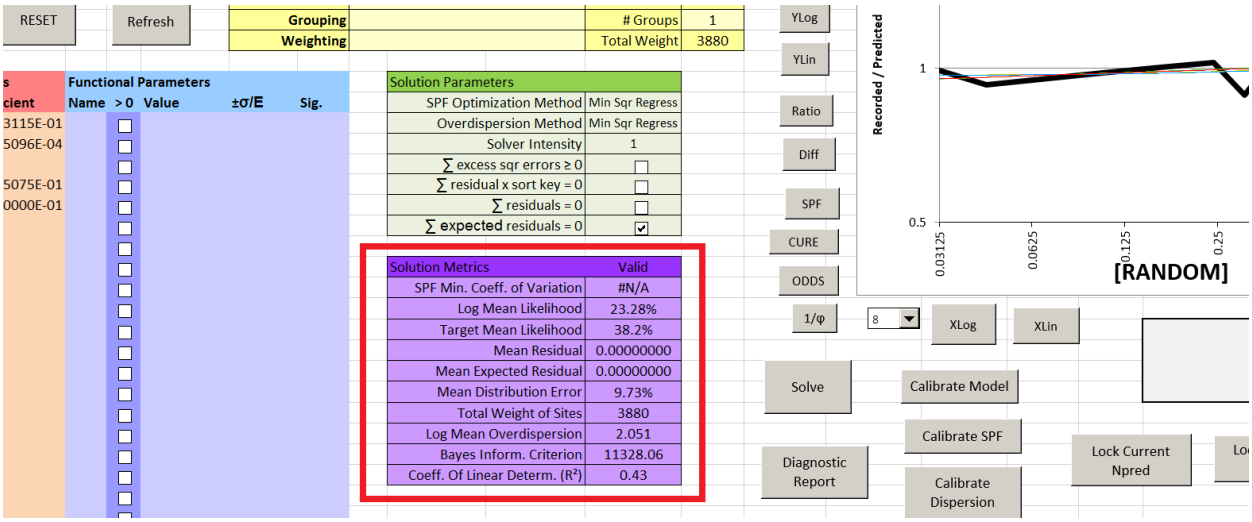
We are now ready to analyze the data. A simple starting point is to assume that crashes are directly proportional to exposure, and then observe the ways in which this assumption does *not* match the data. Exposure is proportional to Time x Length x AADT, so in the SPF box we enter "Y*L*A". (Note: putting "Y" first in the formula prevents Excel autocomplete from annoying us later on when we enter "L" or "A" elsewhere.) Because the data are segments, we enter "1/?" for the default Overdispersion equation. This will set Overdispersion as the reciprocal of the crash prediction, parallel with the Highway Safety Manual method.



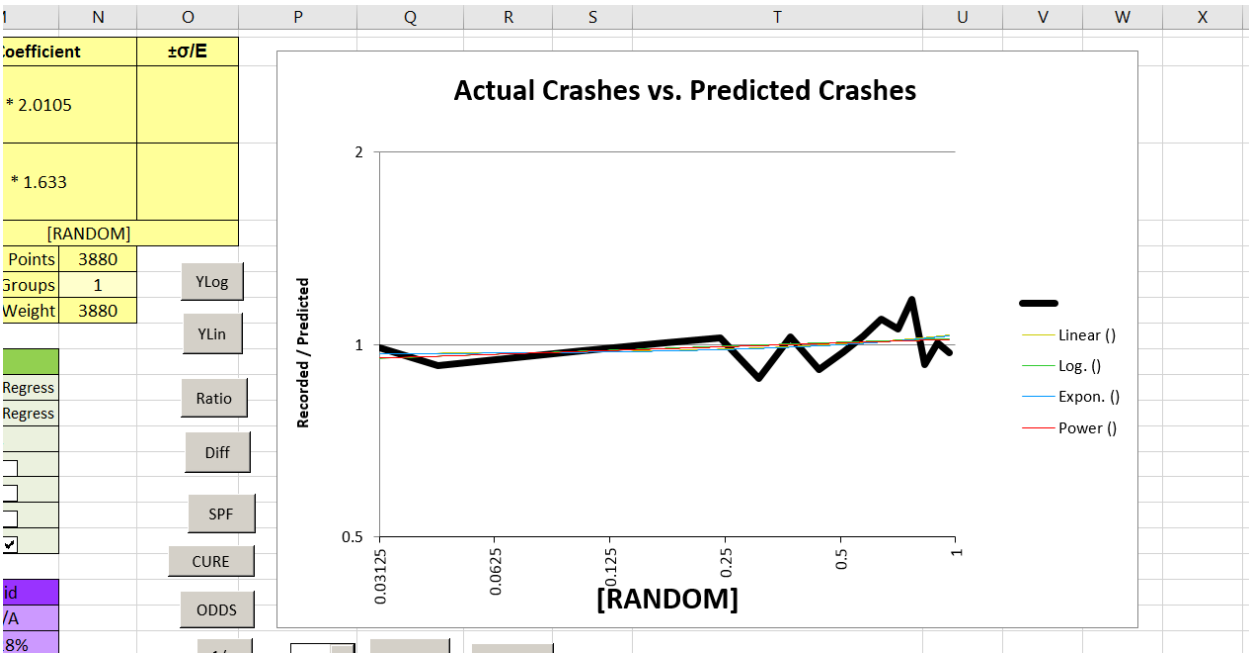
We use the Calibrate Model button to get the initial coefficients for the SPF and Overdispersion formulae:



Some of the solution metrics are already tolerable at this stage. Log Mean Likelihood is not terrible, R^2 is 0.42, and the Mean Distribution Error is acceptable (<10%) meaning that this simplistic fixed-rate model fits the negative binomial distribution fairly well. However, the Log Mean Overdispersion is quite high (>2), indicating that this model is relying heavily on crash counts and making little use of the roadway variables. The SPF Minimum Coefficient of Variation is "#N/A", meaning that the variance in crash counts is smaller than expected. This is because the high Overdispersion means a high expected variance; it doesn't mean the fixed-rate model is a good one.

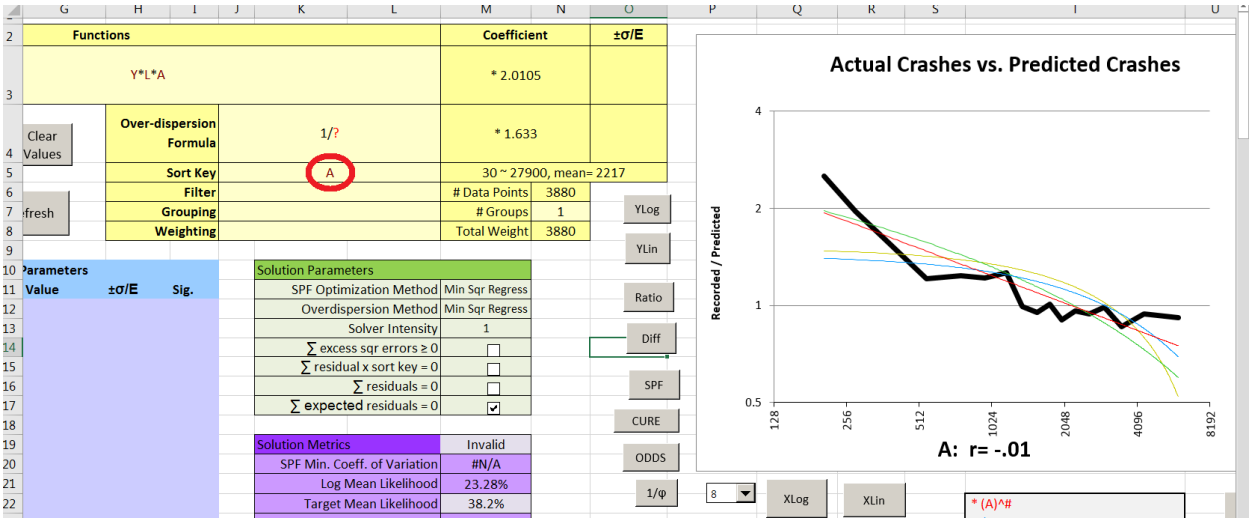


Note also that the graph is quite flat:



This is because the data are randomly sorted (even if they weren't to begin with, the program randomizes them to avoid the appearance of spurious trends).

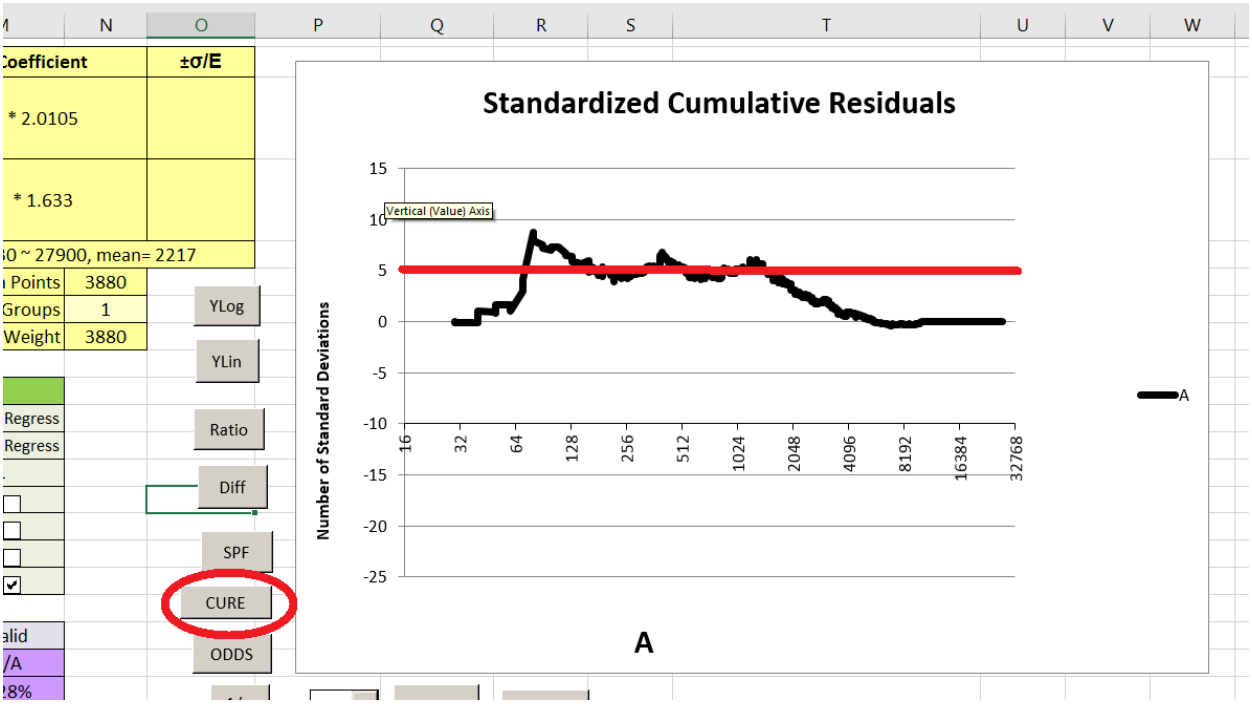
The effects of ADT are usually non-linear (not only total crashes, but rates, vary with ADT); to see the relationship we enter "A" for the Sort Key and get:



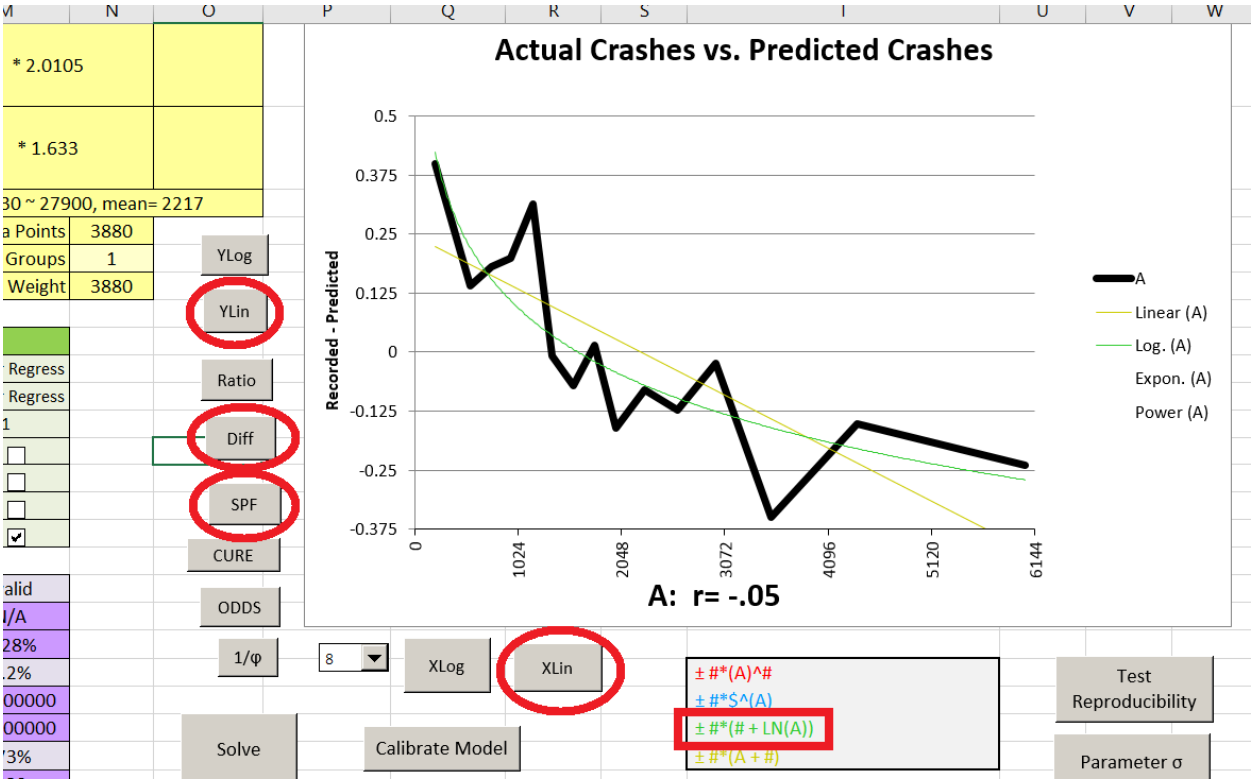
The graph shows a definite decline in crash *rate* with increasing ADT, which seems to taper off at about 2000 ADT. A straight line on this log-log graph (the red trendline) would indicate a power relationship of

the form Y = X^constant. The red trendline does not match the data perfectly, but is closer than any of the other trendlines and we can use a power relationship for further investigation. Note that certain Solution Metrics have been marked as Invalid because we have changed the SPF and they have not yet been recalculated.

By using the CURE button we confirm that this model is indeed extremely biased with regard to AADT; the CURE plot is about five standard deviations from mean for most of the data and is lopsided as well.



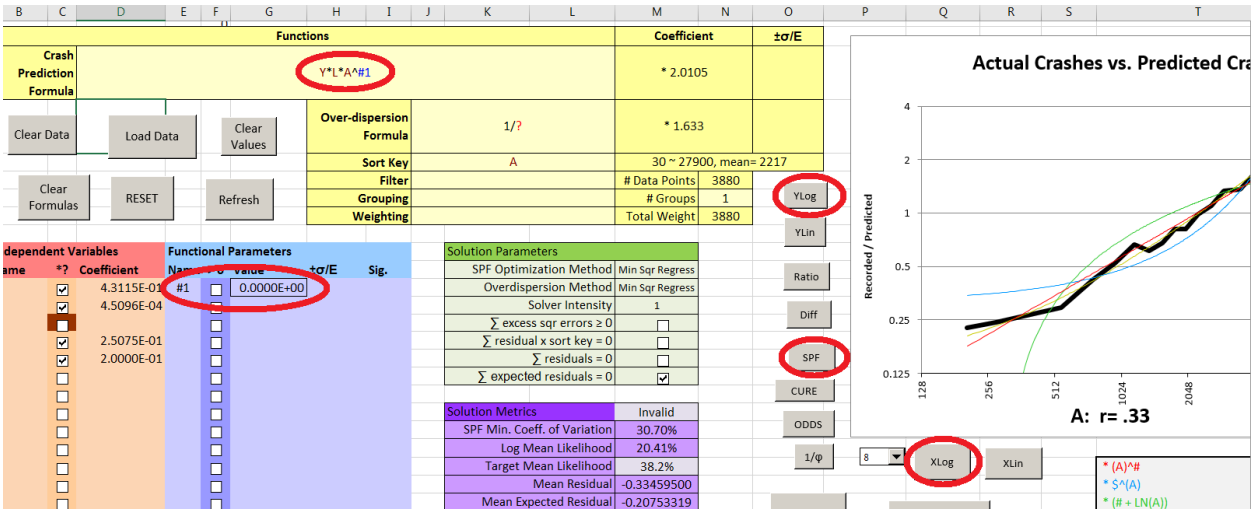
By using the "SPF" button to get back to the SPF graph and the "Diff", "YLin", and "XLin" controls, we can get the following:



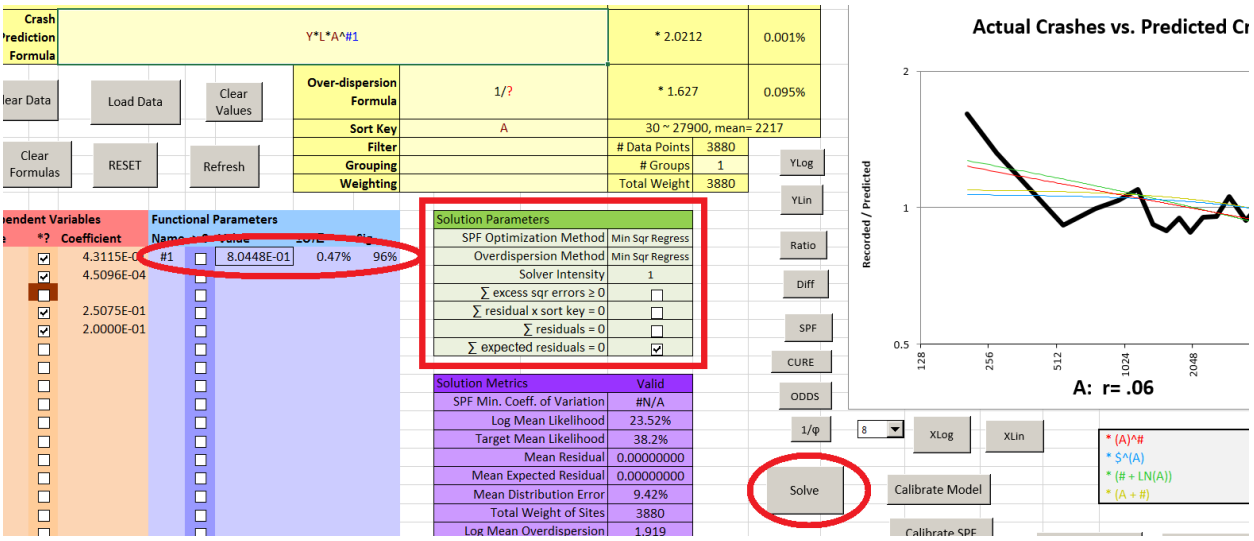
The green (logarithmic) trendline seems to match the data quite closely. Adding the appropriate term to our SPF would cause problems, however. The SPF would then be in the form Y*L*A + #1*(#2+LN(A)) (where #1 and #2 are parameters) which is no longer directly proportional to L (segment length). This may or may not be acceptable. Also, the equation might very well result in negative predicted crashes at high ADT, which could cause the results to be unusable or the program to crash (the graphed data have been smoothed into a kind of rolling average – the highest AADT in the data set is not 6,144 but 27,900!)

Accordingly we use the "Ratio", "YLog", and "XLog" controls to return the graph to its default settings, and enter the modified SPF: Y*L*A^#1. Note that:

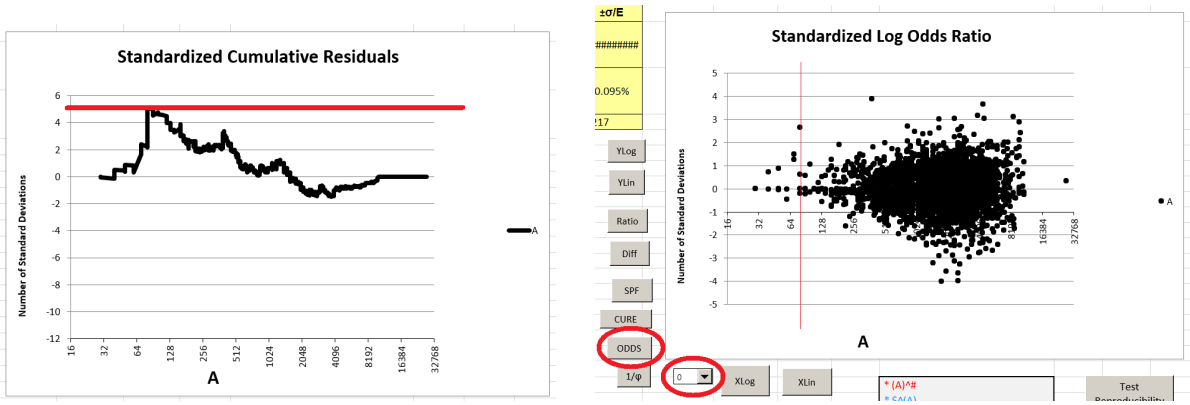
- #1 is automatically created as a new parameter which can take any real numerical value in the model because it has a "#" in its name. It can be seen in the list but has no value yet.
- Y*L*A^#1 is the same as Y*L*A*(A)^#1: A*A^#1 = A^(1+#1) is equivalent to A^#1 because the value of #1 has yet to be determined and it can take any numerical value.
- The graph has changed because #1 has been set to a default value of zero. Until #1 is evaluated, the equation is actually Y*L*A^0 = Y*L.



To evaluate the parameter "#1" we need to use the Solver; calibration does not affect parameter values. For the quickest preliminary solutions, we set the optimization method to Minimum Squares Regression for both the SPF and Overdispersion, set Solver Intensity to 1, allow only the sole default constraint (sum of expected residuals = 0), and click "Solve". After a few minutes the value of #1 has been determined:

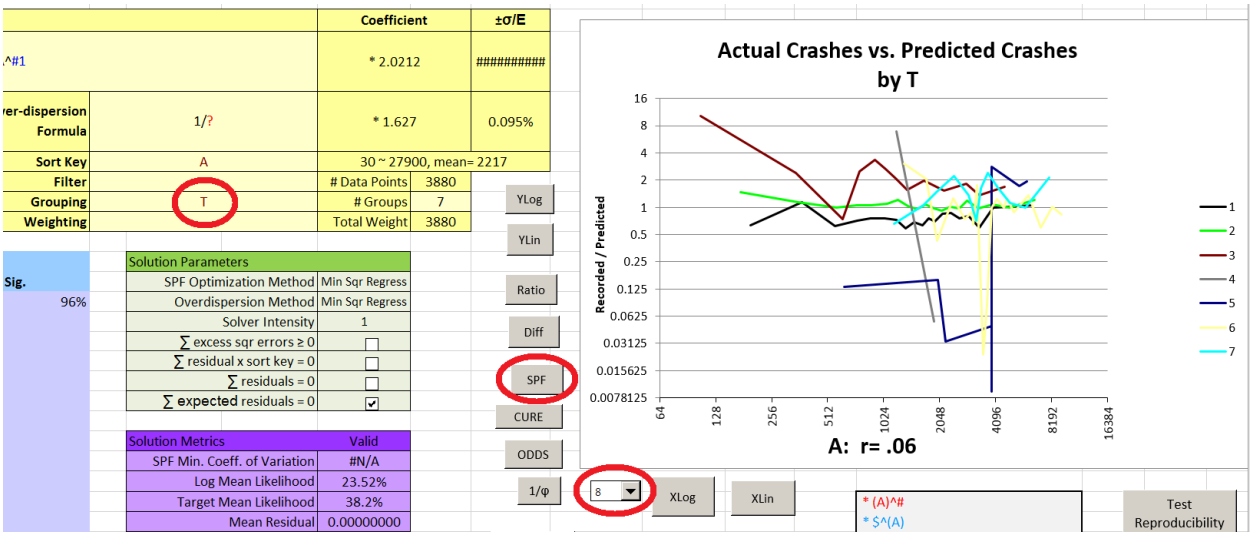


Note that the graph has flattened more than it might seem, because the Y axis scale now shows 0.5-2 instead of the 0.5-4 when we first sorted by ADT. Switching to the CURE plot, we see that the bias has been reduced but it is still dubious, reaching 5 standard deviations, and other metrics are barely improved. The value for the parameter #1 is somewhat less than 1, as expected; its coefficient of error is very small, and its significance is high; none of these indicate any problem. The sharp increase in the CURE plot at about 80 ADT might indicate an outlier, but there are none listed in the "Results" sheet.



Switching to the ODDS plot and reducing the graph smoothness to zero, we confirm that the most extreme site at 80 ADT is less than 3 standard deviations above 1:1 odds. There are only 16 sites in the data with ADT ≤ 80, so the 5 standard deviation excursion in the CURE plot is at least somewhat exaggerated (a t-test shows that its probability is the equivalent of less than 4 standard deviations for a normal distribution). Nonetheless, the CURE and ODDS plot both show that this deviation is the result of a very strong tendency to underestimate crashes on sites below about 150 ADT. This could be due to the low ADT itself, or perhaps correlation between low ADT and some variable that is available to us.

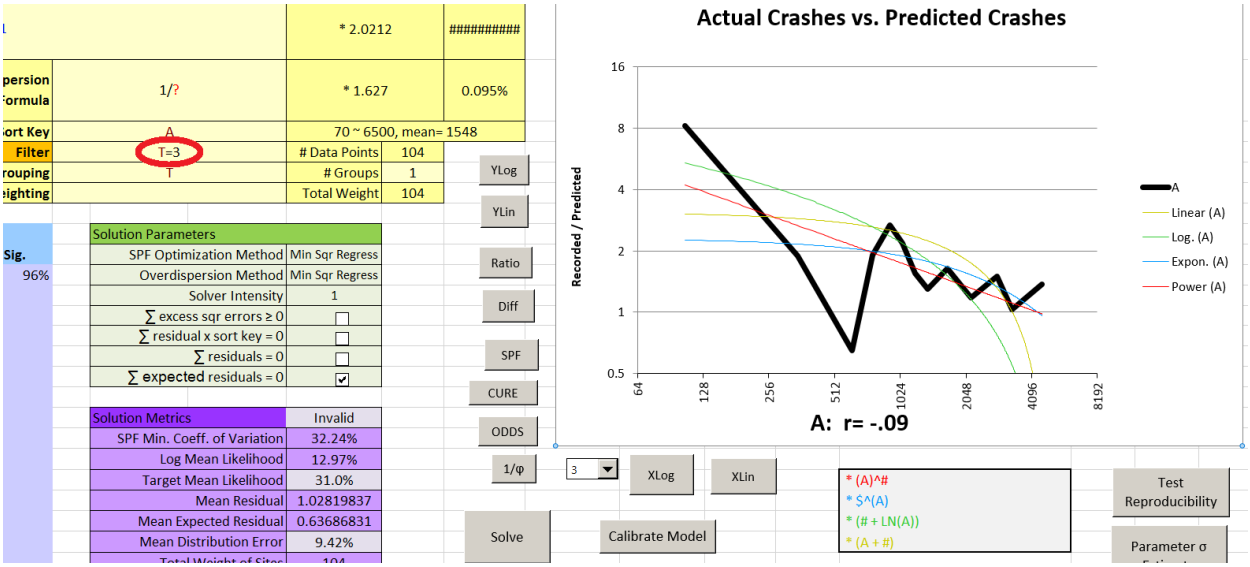
Of the few variables we have, Terrain seems likely to have an interaction with ADT. By changing our graph settings back to SPF and Smoothness 8, and entering "T" (Terrain) as our grouping function, we get



It is immediately apparent that crashes are extremely high (compared to the current model) for very low ADT in mountainous terrain (type 3). Another effect of interest is that crashes are low at high ADTs for T = 4,5,6 (these are various non-residential developed areas).

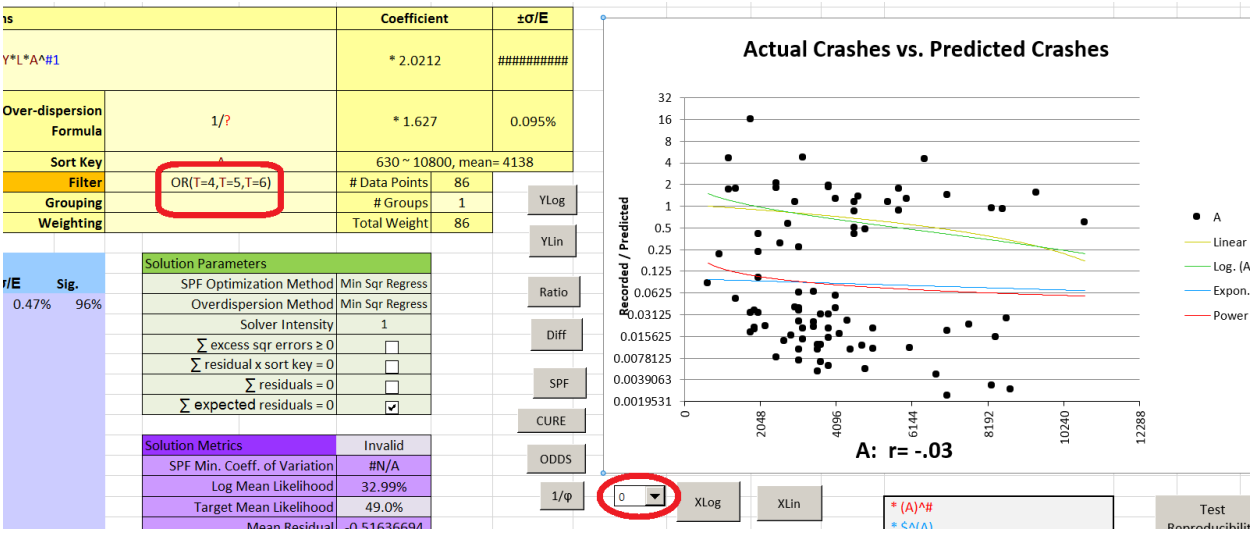
Using an IF function to segregate the mountainous, low ADT sites might be acceptable, but we would have to make an arbitrary choice where to set the ADT cutoff. The Solver cannot do this, even if we were to create a parameter for it (i.e., *IF(AND(T=3,ADT<MIN),ADT^M#*M$,1)). A better solution

would be a continuous function; to see what might be appropriate we can try looking at the mountainous sites alone by entering "T=3" as a Filter:



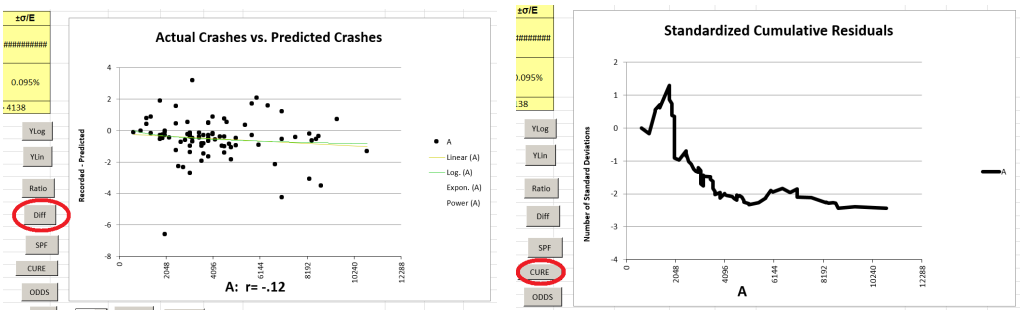
The functional modification that is immediately suggested is "* (A)^#" which can be implemented in our current model as "*IF(T=3,A^M#*M$,1)". The scaling constant M$ is necessary because applying M# is certain to change the average crash prediction for mountainous sites, probably downward.

The data for terrain types 4 and 5 are very sparse – if we enter T=4 and then T=5 into the Filter we discover there are only 2 and 8 sites, respectively (and the Smoothing gets progressively distorted as we try more Filters, so we will eventually have to reset it). Therefore we combine Terrain 4 and 5 with Terrain 6 (because all are business areas) and treat them together. Setting the Filter accordingly to OR(T=4,T=5,T=6), and erasing Grouping, we get a different picture of the behavior of these Terrain types. At Smoothness zero, we see:

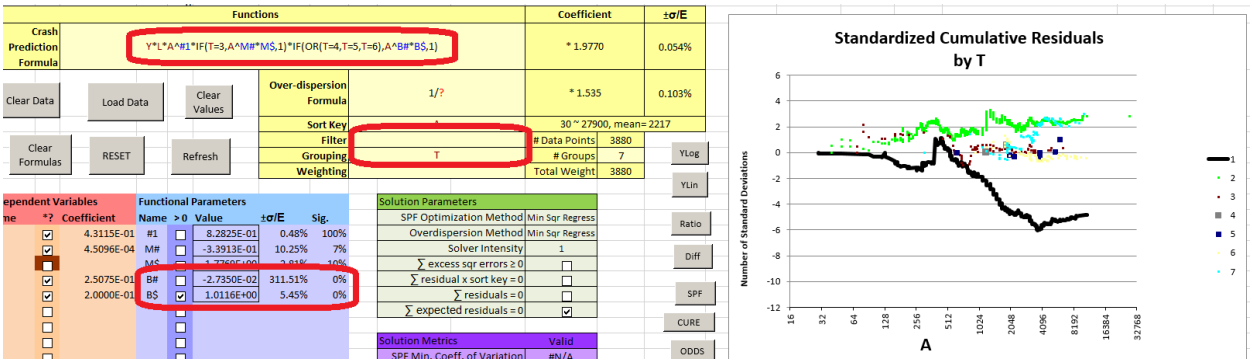


The sites are divided into two horizontal bands, with some extreme values in the lower band. This tells us that the sites in the lower band probably all have zero crashes, and this is making the ratio of actual to predicted crashes look deceptively low (because zero values are plotted on the logarithmic chart as being extremely low positive values).

Using the Diff button (which automatically also sets the Y axis of the graph to linear) we get a very different picture; which is confirmed by the CURE plot.



There is a slight but consistent downward drift over much of the ADT range. We may as well use the same kind of functional modification that we did for Terrain type 3, so our next SPF will be Y*L*A^#1*IF(T=3,A^M#*M$,1)*IF(OR(T=4,T=5,T=6),A^B#*B$,1). After entering this, *deleting the Filter*, and setting the Grouping back to "T", we run the Solver again (the Grouping won't affect the solution, but we still want to see how different Terrain types fit the model.

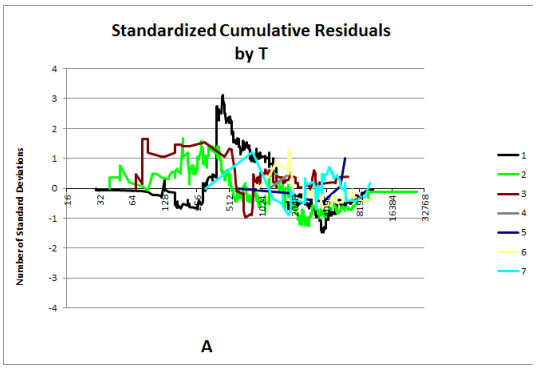


The CURE plots for Terrains 3-6 are now optimum, but for other Terrain types the CURE plots range from dubious to worse. Also, the parameters used to evaluate the business areas seem to have made little difference to the model – their significance is nil and their values make the relevant term nearly equal to one (ADT^-0.027*1.01). We will retain them for now though, because they are insignificant only in the current model. When we add new terms for terrain types 2 and 7, hoping to improve the CURE plots, we may as well rewrite the whole function:

Functions	
Crash Prediction Formula	Y*L*CHOOSE(T,A^#1,A^#2*$2,A^#3*$3,A^B#*B$,A^B#*B$,A^B#*B$,A^#7*$7)

Note that there is no scaling coefficient for T=1 because that is treated as the default Terrain; i.e. it is scaled by the system scaling coefficient and everything else is relative to T=1. The parameter "#1" becomes the ADT exponent for T=1; at this point having an exponential ADT term for the entire function would be redundant. The B# and B$ parameters are repeated for Terrains 4-6.

After solving, the Solution Metrics are still unimproved, but the CURE plots are decidedly better and the business area parameters have become significant:

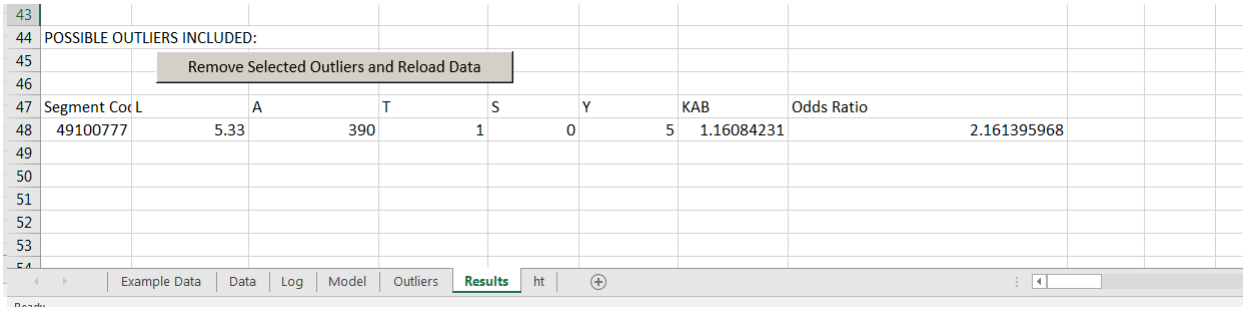


...efficient	Name	> 0	Value	±σ/E	Sig.
4.3115E-01	#1	☐	9.2422E-01	1.44%	58%
4.5096E-04	#2	☐	7.7108E-01	0.71%	66%
	$2	☑	1.2983E+00	0.16%	14%
2.5075E-01	#3	☐	4.8993E-01	7.11%	11%
2.0000E-01	$3	☑	2.1456E+00	2.82%	13%
	B#	☐	8.0582E-01	10.67%	9%
	B$	☑	1.2166E+00	5.48%	3%
	#7	☐	8.8070E-01	8.78%	9%
	$7	☑	1.7242E+00	4.10%	9%

Solution Parameters	
SPF Optimization Method	Min Sqr Regress
Overdispersion Method	Min Sqr Regress
Solver Intensity	1
∑ excess sqr errors ≥ 0	☐
∑ residual x sort key = 0	☐
∑ residuals = 0	☐
∑ expected residuals = 0	☑

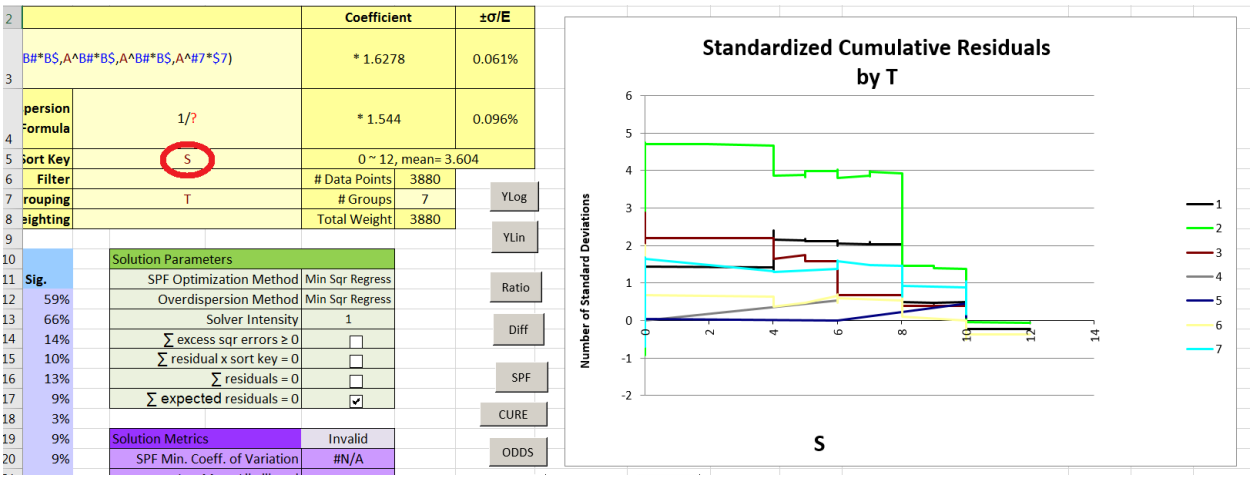
Solution Metrics	Valid
SPF Min. Coeff. of Variation	#N/A
Log Mean Likelihood	23.93%
Target Mean Likelihood	38.2%
Mean Residual	-0.00000343
Mean Expected Residual	-0.00000207
Mean Distribution Error	9.38%
Total Weight of Sites	3880
Log Mean Overdispersion	1.815
Bayes Inform. Criterion	11188.42
Coeff. Of Linear Determ. (R^2)	0.45

Terrain Type 1 appears to have a possible outlier at around ADT 350. Checking the Results sheet confirms this:

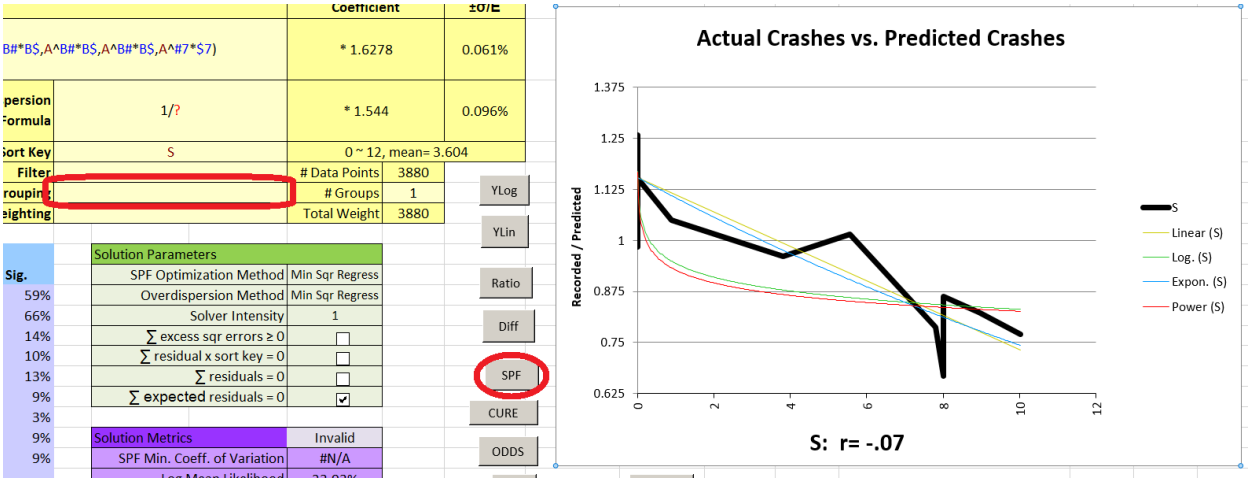


Investigation shows that this highway segment is actually in mountainous terrain (T=3). We could either select row 48 in Results and click "Remove Selected Outliers and Reload Data", or correct the data in the Data sheet and then use Load Data in the Model sheet. For this example we take the latter option, making sure to click "no" when asked to Clear Formulas. Nonetheless the values in the model are reset, as it is not valid for the re-loaded data. When the Solver is run again, segment 49100777 now causes a large jump in the CURE plot for T=3, but it no longer appears on the Results sheet, is only the 15th site in sequence for Terrain 3, and is not the most unlikely site by Odds Ratio, so we will accept it.

Setting aside segment length for now (because we want our SPF to vary directly with segment length), our only remaining variable is "S" (Paved Shoulder Width). Setting the Sort Key to S but leaving Grouping set to "T", we get an interesting CURE plot:

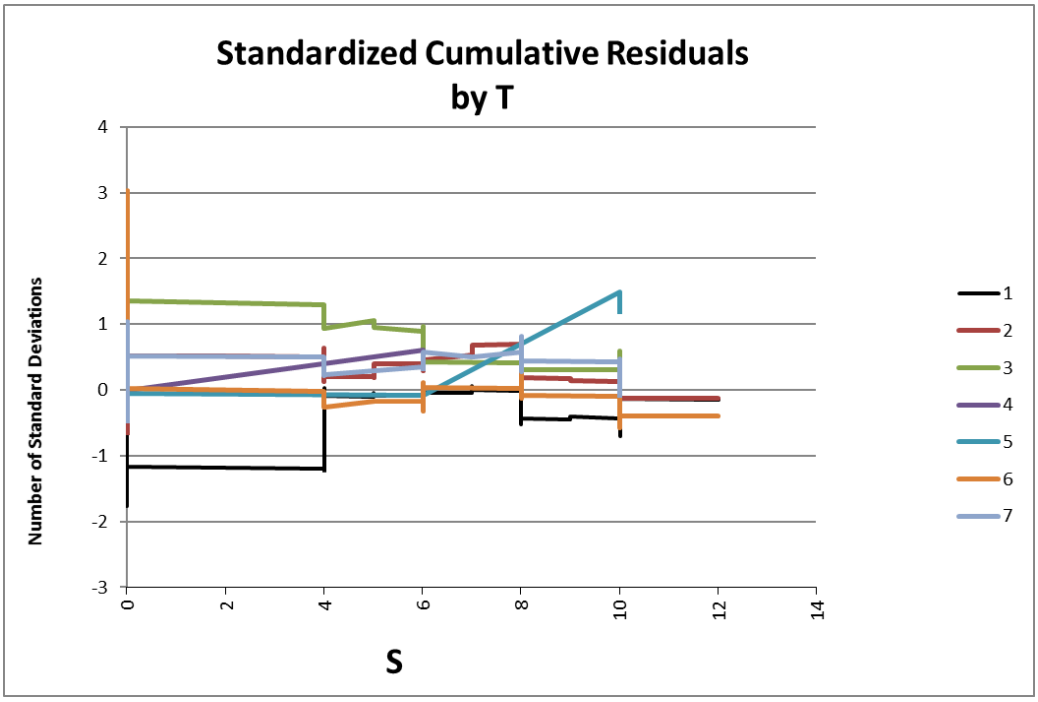


Except for T=2 (rolling terrain), shoulder width seems unimportant. However, the SPF graph (of actual crashes vs. predicted crashes) shows no obvious strong relationship for T=2 either, and the amount of data is small: for all terrain types other than 1 or 2, there are only 141 segments with shoulders of any width. Therefore we will start by disregarding terrain. Removing the grouping and setting the graph back to SPF, we get

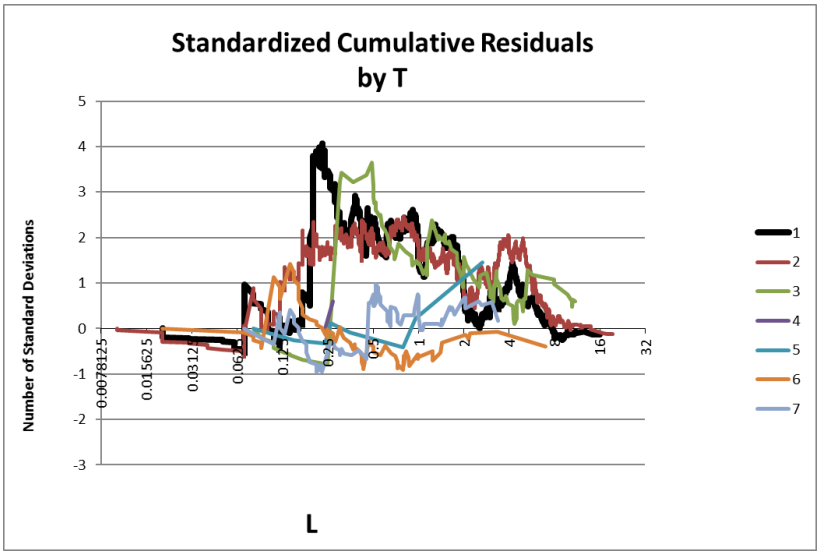


In this case, the X axis was automatically switched from logarithmic to linear scale when we changed the Sort Key, because Shoulder Width is zero for some sites. If for some reason the X axis has been switched to logarithmic, the graph will be quite distorted. It can be switched back using "XLin".

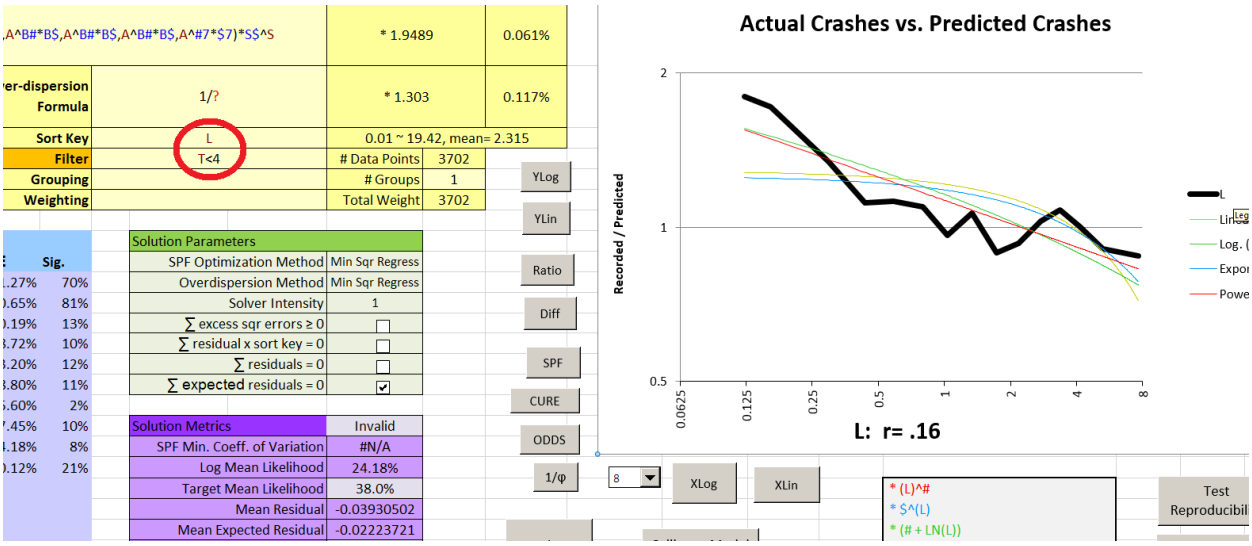
Switching the Y axis to linear scale using "YLin" does not show enough difference to help choose between the blue and yellow trendlines. We choose the functional modification for the blue trendline, which avoids any numerical difficulties. Our new SPF will be Y*L*CHOOSE(T,A^#1,A^#2*$2,A^#3*$3,A^B#*B$,A^B#*B$,A^B#*B$,A^#7*$7)*S$^S. The "$" in the new parameter's name forces it to be positive, reflected in the fact that its ">0" checkbox is automatically checked. (Taking a zero or negative number to the power of Shoulder Width would cause a numerical error). Restoring the Terrain grouping and running the Solver again gets us the CURE plot



Which is quite satisfactory. The only variable we have left to consider is L (segment length). If we are willing to accept that crash *rate* will vary with segment length, we enter L for the Sort Key. The CURE plot grouped by T (terrain) shows an anomaly for types 1-3 (rural terrain types) in the mid-range of segment lengths.

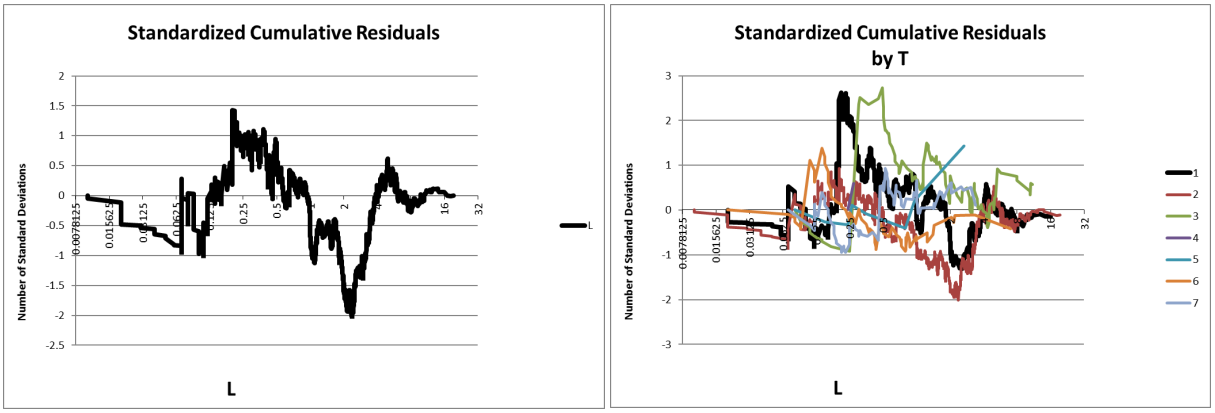


We switch the graph back to showing SPF residuals, and use a Filter (T<4) to see only terrain types 1-3:



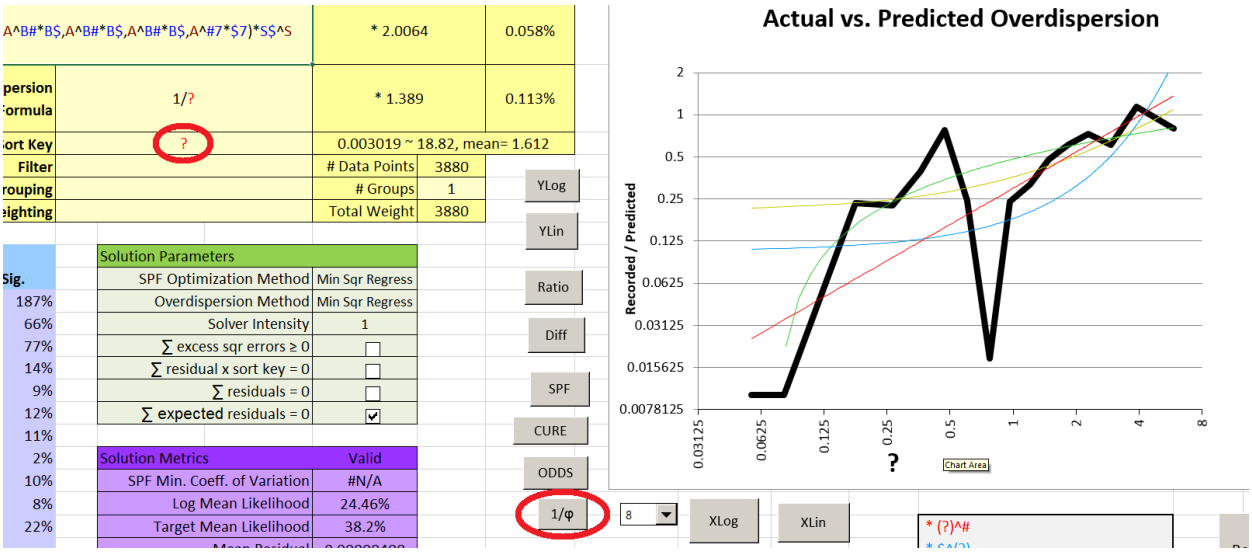
From the graph, the best SPF term to add appears to be *L^L#. Since we have a Filter on, however, it needs to be a conditional term: *IF(T<4,L^L#,1). Since L already exists in our SPF, we can simplify this to get Y*L^IF(T<4,L#,1)… at the beginning of the SPF. Notice that in this case the "1" in the final argument of the IF function is an exponent, in the first suggested form it was a multiple – the fact that it stayed the same is mere coincidence.

Before running a new solution, we delete the Filter so that our SPF will be valid for the whole data set. The resulting CURE plot for segment length is significantly improved, even when grouped by Terrain:

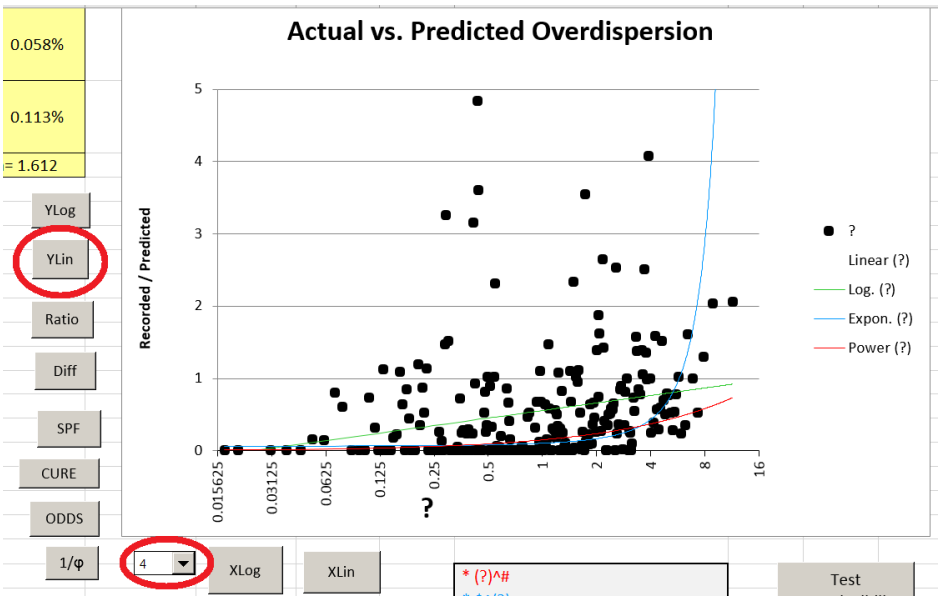


This exhausts the variables that can be used for crash prediction (Y is 5 years for every site). We have not yet considered Overdispersion, however. What is shown here is by no means the only way to fit Overdispersion, but it is the method preferred by the author.

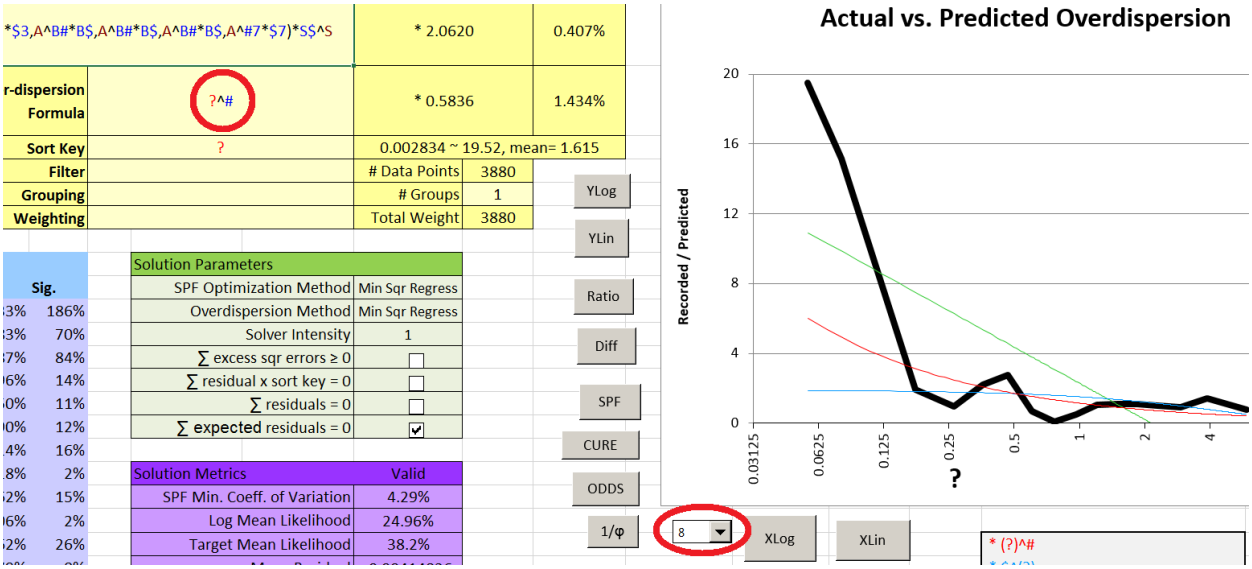
First we use the "1/ϕ" button to see the graph of Overdispersion. Then we set the Sort Key to "?" to see Overdispersion graphed relative to the current crash prediction – this is the only variable we will use in our Overdispersion equation.



Note that the graph shows overall very little Overdispersion compared to what our formula (1/?) predicts; also the current solution has a high mean (predicted) Overdispersion of 1.568. By playing with the graph Smoothness we can see why: there are points with enormous ratios of estimated to predicted Overdispersion (up to about 120:1) as well as large numbers of points with zero or negative Overdispersion (these show up as a solid line at the bottom of the graph at lower Smoothness settings). Setting the Y axis to linear may also help. Then at Smoothness 4 we see:



Based on this, the best Overdispersion function to use might be based on the logarithm of the crash prediction, but basing it on a power of the crash prediction presents no numerical difficulties and has been established as a way of getting reasonably stable results for most models. Accordingly we will try the function ?^# - i.e., predicted crashes to the power of some parameter between negative and positive infinity. Solving for this and setting Smoothness back to 8 we get

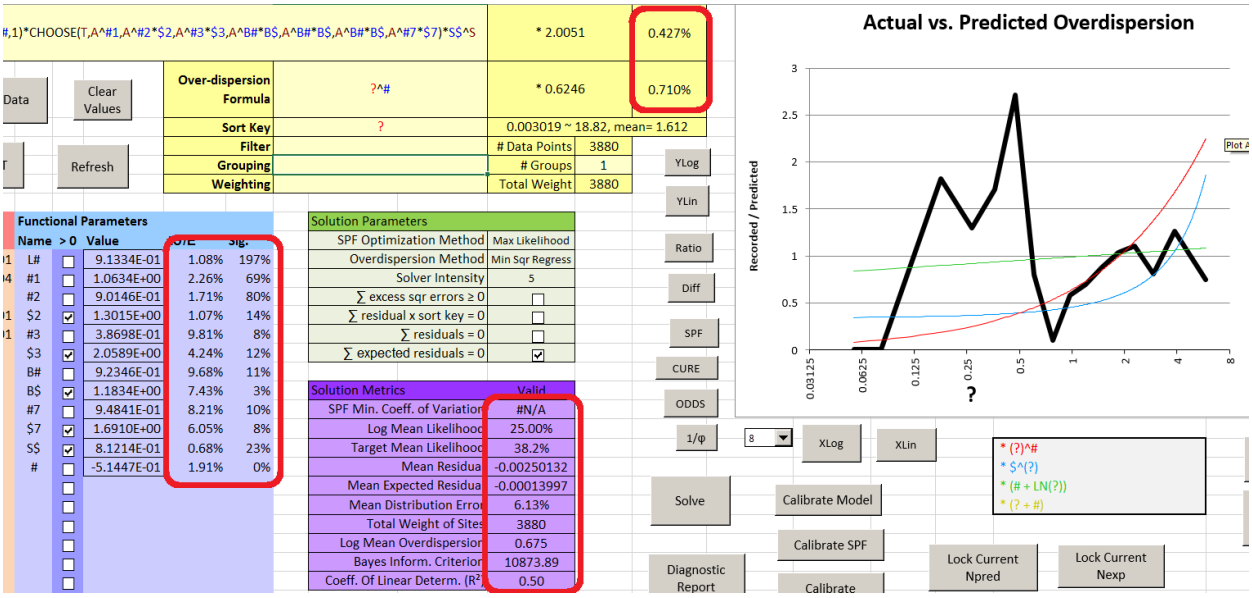


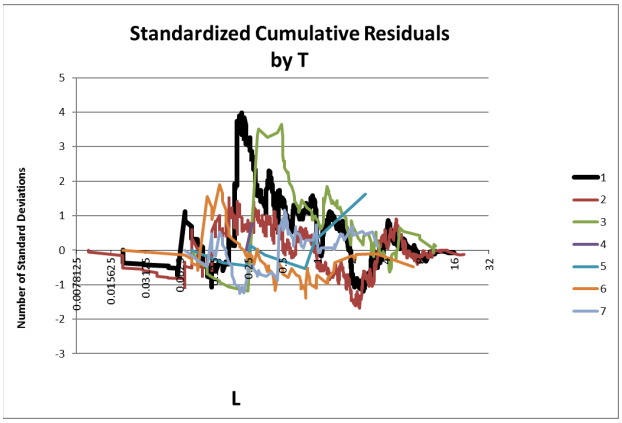
There is still a discrepancy at very low crash predictions, but the fit is much better and the mean Overdispersion is a much more acceptable 0.627. The value of the parameter "#" is -0.515, which from experience is typical for well-fitted SPFs.

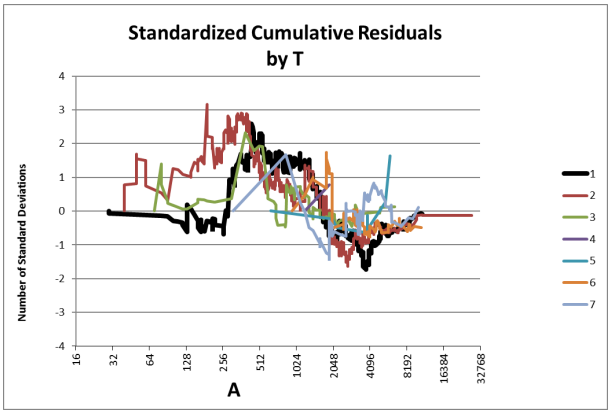
An option at this point would be to eliminate parameters of low significance before doing a final Solver run. However, the two parameters with 2% significance – B$ and $7 – are scaling factors. Eliminating them might very well lower the BIC, but this would be superficial – to eliminate a parameter whose value happened to be close to 1 isn't the same as never having that parameter in the first place. Also, their significance is low mostly because they affect a relatively small number of sites. Eliminating them would probably ruin the CURE plots for ADT for terrain types 4-7. So we will make our final Solver run at Intensity 5, setting the SPF Optimization Method to Max Likelihood. For Overdispersion, we will stick with Minimum Squares for better stability, but instead of using the parameter "#" we will set the Overdispersion as 1/SQRT(?). This form has proven to be common and useful in the past, and fits the current value of # very closely.

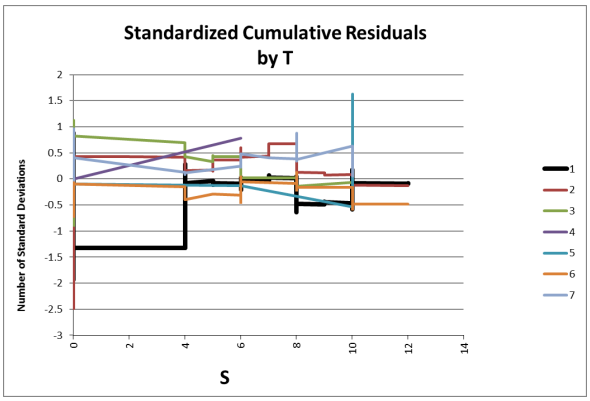
The final result is not too different; the mean Overdispersion is slightly increased and the least significant parameters have changed and become more significant. The Overdispersion vs. "?" graph (plotted as a ratio with a linear Y axis) is fairly flat and all the CURE plots (grouped by terrain) are acceptable. All of the parameter coefficients of variation are under 10%, which is good, and the Mean Distribution Error is under 10%. The "SPF Minimum Coefficient of Variation" is still below zero – but Overdispersion is now low enough (mean 0.656) that we can say this is because the model is a reasonable fit. With only four explanatory variables and a simple functional form, it is unlikely that the

model is overfitted. Even though some of the metrics show little improvement from the first calibration, the CURE plots and moderate Overdispersion show that this is a much better model. Improvement might be possible (note the CURE plot for segment length for terrain types 1 and 3 was made worse by the final solution), but a more complex function might not provide a significant improvement in actual predictive power – it might even be overfitted.









There is one more test we can do to confirm the stability of the model. First we set the Solver Intensity back to 1, to save time. Then we use the "Test Reproducibility" button and select ten trials. The Results tab shows that the stability of the parameters is rather poor for Overdispersion (which is typical) and for terrain types 3-7 (probably because the amount of data is small).

	A	B	C	D	E	F	G	H
0								
1		Parameter	Value	±σ	±σ/E	Significance	RMS Error on Replication	
2		L#	9.13642646E-01	9.8571E-03	1.079%	197%	1.87%	
3		#1	1.06370784E+00	2.3997E-02	2.256%	69%	4.26%	
4		#2	9.01773343E-01	1.5393E-02	1.707%	80%	2.32%	
5		$2	1.30134466E+00	1.4084E-02	1.082%	14%	3.39%	
6		#3	3.85613925E-01	3.7839E-02	9.813%	8%	16.15%	
7		$3	2.05983857E+00	8.7180E-02	4.232%	12%	9.44%	
8		B#	9.21141040E-01	8.9078E-02	9.670%	10%	39.17%	
9		B$	1.18544800E+00	8.7750E-02	7.402%	3%	26.32%	
0		#7	9.47664280E-01	7.7801E-02	8.210%	10%	16.21%	
1		$7	1.69215068E+00	1.0218E-01	6.038%	8%	22.98%	
2		S$	8.12018389E-01	5.5470E-03	0.683%	23%	1.09%	
3		<Cspf>	2.00520000E+00	8.7174E-03	0.435%		2.81%	
4		<Cdsp>	6.08100000E-01	4.8993E-03	0.806%		45.14%	
5								
6	POSSIBLE OUTLIERS INCLUDED:							
7		Remove Selected Outliers and Reload Data						
8								
9	Segment Cod	L	A	T	S	Y	KAB	Odds Ratio
0								

Suppose we decide to keep this model. We will probably want to simplify the final mathematical form using the tool in the Results sheet.

The formulae in the light yellow cells are the working part of this sheet. Here the model functions are inserted along with all the scaling coefficients. We will simplify these equations before recording or using them. Above the yellow cells the original formulae are reiterated; if we make an irreversible mistake in the simplification process, we can copy and paste the content of these cells into the appropriate yellow "Trial" cells. Below this, the functional forms, parameter values, and other parameter related outputs are given.

We start the simplification process by changing the values in both Trial cells out of scientific notation, rounding off digits that we know won't matter, and removing redundant parentheses. Then we factor out all the coefficients, rearranging terms to make this easier, and finally round off numbers by trial and error until we get a result with an error that is acceptable and numbers of significant digits that we like. In this case, the greatest error caused by the round is 0.77% for the SPF and 0.31% for the Overdispersion.

Trial SPF	Y*IF(T<4,L^0.914,L*0.93)*CHOOSE(T,A^1.064,A^0.902*4.53,A^0.386*383,A^0.921*3.56,A^0.921*3.56,A^0.921*3.56,A^0.948*4.14)*0.949^S*0.000051	0.77%
Trial 1/φ	0.61/sqrt(?)	0.31%

The final results can be stored anywhere; they can be readily saved as a sheet within the Spreadsheet using the "Save These Results" button on the "Results" sheet and assigning an appropriate name.

Solver Precision	1E-10
SPF Optimization	Max Likelihood
Dispersion Estima	Min Sqr Regress
# Trials	2
# Data Points	3880
Total Weight	3880

Save These Results

Filter	
Weight	

Mean Likelihood	25.00%
Target Likelihood	38.23%
Coeff. Of Var.	#N/A
Mean 1/φ	0.656
BIC	10864.68
Mean Residual	-0.002491
Mean Exp. Res.	-0.000125
Fitting Error	6.10%
Sim. Nexp Z^2	0.663096092

The saved model can be inspected later but as of this time there is no utility to reload it, and hence no way to use the simplification tool once new data has been loaded or anything in the Model sheet has been changed. Saving the results does *not* save any record of outliers removed, nor does it save any record of the CURE plots that were achieved.

Appendix D: Selected Equations

Relative Likelihood:

$$\ln(L_i') = ln\Gamma(N_i + \phi_i) - ln\Gamma(\phi_i) + \phi_i ln(\phi_i) + N_i ln(\mu_i) - (N_i + \phi_i) ln(\mu_i + \phi_i)$$

L_i' = relative likelihood (the variable part of the likelihood of N for each site *i*)
lnΓ = natural log of gamma function (i.e. Excel GAMMALN function)
N_i = recorded crash count for site *i*
$\emptyset_i$ = reciprocal of computed Overdispersion for site *i*
μ_i = computed SPF prediction for site *i*

Mean Likelihood (Default Objective Value):

$$n\ln(L) = \sum_{i=1}^{n} [\ln(L_i') - \ln\Gamma(N_i + 1)]$$

L = geometric mean likelihood
n = total number of sites

Bayesian Information Criterion:

$$BIC = K\ln(n) - 2n\ln(L)$$

BIC = Bayesian Information Criterion
K = total number of free parameters (including SPF and Overdispersion coefficients)

Target Likelihood:

$$\hat{L} = \exp\left\{\frac{1}{n}\sum_{i=1}^{n} \ln\left[\frac{{\lambda_i}^{N_i} e^{-\lambda_i}}{N_i!}\right]\right\}$$

$$\lambda_i = N_i \quad \text{if } N_i > 0$$

$$\lambda_i = \delta \quad \text{if } N_i = 0$$

$\hat{L}$ = Estimated Likelihood for perfect model
δ = Fraction of sites with non-zero crash counts, or 0.5, whichever is less

Mean Overdispersion

$$mean\ 1/\emptyset = \exp\left\{\frac{1}{n}\sum_{i=1}^{n} \ln\left(\frac{1}{\emptyset_i}\right)\right\}$$

Mean Fitting Error

$$P_i = \left[\sum_{x=0}^{N_i} \frac{\emptyset_i{}^{\emptyset_i}\mu_i{}^{x}\Gamma(x+\phi_i)}{x!\,(\mu_i+\emptyset_i)^{(x+\emptyset_i)}\Gamma(\phi_i)}\right] - \frac{\emptyset_i{}^{\emptyset_i}\mu_i{}^{N_i}\Gamma(N_i+\phi_i)}{2N_i!\,(\mu_i+\emptyset_i)^{(N_i+\emptyset_i)}\Gamma(\phi_i)}$$

$$P_i' = \frac{\sum_{j=1}^{n}(P_i > P_j) + 0.5}{n}$$

$$P_i{}' \leq \sum_{x=0}^{\widehat{N}_i} \frac{\emptyset_i{}^{\emptyset_i}\mu_i{}^{x}\Gamma(x+\phi_i)}{x!\,(\mu_i+\emptyset_i)^{(x+\emptyset_i)}\Gamma(\phi_i)}$$

$$Mean\ Fitting\ Error = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|N_i - \widehat{N}_i\right|}{\widehat{N}_i + 0.5}$$

P_i = Midpoint of cumulative probability range of N_i

P_i' = Cumulative probability site *i* should have if the negative binomial model is a perfect fit.

$\widehat{N}_i$ = Expected number of crashes for site *i* if the negative binomial model is a perfect fit.

Standardized Cumulative Residual

This tool uses a different method for calculating the variance of the cumulative residuals, since a known distribution (negative binomial) is assumed. Cumulative variance is adjusted for the number of data points. No adjustment is made for the degrees of freedom in the model. Note that the correction term, though appearing to be a Bessel correction, is just a crude attempt to approximate a t-distribution without using too much processor time.

$$Z_j = \frac{\sum_{i=1}^{j}(N_i - \mu_i)}{\sqrt{\frac{j}{j-1}\left\{\sum_{i=1}^{j}\left(\frac{\mu^2}{\varphi} + \mu\right)\right\}}}$$

Z_j = standardized cumulative residual at data point *j*

Parameter Significance:

$$Significance = \left\{\exp\sqrt{\mathrm{VAR}\left(ln\left(\frac{\mu_i}{\mu_{i,0}}\right)\right)}\right\} - 1$$

$\mu_{i,0}$ = computed SPF prediction for the site when the parameter in question is set to its default value (0 or 1).

Parameter Coefficient of Variation (±σ/E) (Long Method):

For parameters constrained to be greater than zero:

$$\pm\sigma/\mathrm{E} = ln\sqrt{AB}$$

Where A>1, B>1 are solutions to:

$$L'\left(\frac{V_0}{A}\right) = \frac{L'(V_0)}{e^{0.5}} \quad and \quad L'(V_0 B) = \frac{L'(V_0)}{e^{0.5}}$$

V_0 = Value of parameter at maximum likelihood.

For other parameters:

$$\pm\sigma/\mathrm{E} = \frac{A+B}{2V_0}$$

Where A>0, B>0 are solutions to:

$$L'(V_a - A) = \frac{L'(V_0)}{e^{0.5}} \quad and \quad L'(V_a + B) = \frac{L'(V_0)}{e^{0.5}}$$

L' = Variable part of total likelihood as a function of one parameter (other parameters held constant).

SPF Calibration:

The existing SPF coefficient is multiplied by a factor C:

$$C = \frac{\sum N_i}{\sum \mu_i}$$

Overdispersion Calibration:

The existing Overdispersion coefficient is multiplied by a factor C:

$$C = \frac{\sum[(N_i - \mu_i)^2 - \mu_i][{\mu_i}^2 K_i]}{\sum {\mu_i}^4 {K_i}^2}$$

Where: N_i = observed crashes, μ = predicted crashes (SPF), K = overdispersion function

Minimum Squares Regression for Overdispersion:

The statistic that is minimized is the sum of the squares of the excess variance:

$$\sum \left[(N_i - \mu_i)^2 - \mu_i - \frac{{\mu_i}^2}{\varphi_i}\right]^2$$

Glossary of Terms

Bayesian Information Criterion: A measure of how much information loss a given model has. The lower the BIC, the better the model – given certain assumptions. BIC depends on the model's likelihood and on the number of its parameters. Adding parameters to a model generally increases its likelihood, but can also increase its BIC, meaning it is more likely that the model is now overfitted.

Calibration: Modifying an SPF by adjusting only the SPF and/or Overdispersion coefficients.

Coefficient: A number that something else is multiplied by, like the scaling factors for variables.

Coefficient of Error: The standard error of an estimate divided by the absolute value of the estimate.

Data point: A single line of site data in a worksheet, usually comprising data for a single site.

Discontinuous: A function that can have abrupt changes in its output, or even in the way its output is changing (upward to downward, slow to fast) as the values of its parameters change gradually.

Filter: A function with a true/false output that determines which Data Points will be included for calculations and solutions. Not related in any way to Excel's built-in filter.

Function: An equation that calculates a single output number from various parameters, variables, and possibly constants. Functions in this context can include almost anything that might appear in an Excel formula, *except* for cell/range references. They can also include variables and parameters that are named by the user. For each function, a different output is calculated for every site, but the user does not normally see any of these outputs.

Functional form: The structure of a function – the parameters, variables, operators, Excel functions, parentheses, etc. used to describe the actual calculation, ignoring the values of the parameters.

Likelihood: The probability of getting a known result, given certain hypothetical model characteristics (and other assumptions).

Model: A system of equations used to make predictions; in this case, the SPF and Overdispersion equations are two unknown parts of a model that the user is trying to construct.

Objective Value: A number that is a function of all the data, the SPF and Overdispersion equations, and the values of their parameters. The Solver tries to set the parameter values so as to maximize or minimize the objective value.

Operator: A symbol for a mathematical operation, e.g. +, -, *, /, ^ etc.

Overdispersion: A measure of how much more scattered the crash counts are compared to what the SPF alone predicts. High Overdispersion means there are risk factors the model does not account for so predictions will rely more on crash history (or that the model is a poor fit).

Overfitted: A model which appears to fit the data better than should be possible, but is likely to be a *worse* fit to any future data.

Parameter: A value in a model that is unknown when starting (cannot be directly calculated from the data) but is a constant once the model has been finished.

Safety Performance Function: An equation for predicting crashes based on factors other than previous crash history.

Scaling: Multiplying a variable, parameter, or a whole function by a numerical constant, usually in order to get a number which is in a range that is suitable for the Solver to use.

Site: A location for which a single crash count and a single value for each variable is used.

Smoothness: The spreadsheet condenses data into a sort of rolling average so that its patterns are more visible. The amount of smoothing is set automatically but can be altered by the user. Each point of smoothing cuts the number of points displayed on the graph by about half.

Solver: An Excel add-in that finds solutions to complex numerical problems, especially optimization problems. In this context it is used to find the optimum values of parameters as measured by maximum likelihood.

Sort Key: A function of which the output is used to sort the Data Points so that relationships are more easily seen between the data and whatever the Sort Key is. Typically the Sort Key is just a single variable.

Underdispersion: When the crash counts are less scattered than the model predicts, i.e. Overdispersion is negative. A model with underdispersion is defective.

Variable: A datum, other than crash counts, that is available for each site.

Weighting: A function that can give some sites extra influence on the outcome of calculations, particularly in order to allow one line of data to represent multiple identical sites.