

Section 12 – Statistical Details: Analysis

This section of the manual provides specific details on selected aspects of the statistical analyses you see in Design-Expert[®] software outputs. The primary source of information remains the on-line Help system. Always check Help first before the manual. In this section and throughout the manual, we've provided references for follow-up by those who really want to know where the numbers come from. See the Appendix for a comprehensive list, including the references noted below.

If you need more background on statistical modeling, we recommend:

- *Applied Regression Analysis* by Draper and Smith
- *Applied Linear Regression* by Weisberg.

As noted in earlier tutorial sections, for the basics of DOE we recommend:

- *Design and Analysis of Experiments* by Montgomery
- *Response Surface Methodology* by Myers and Montgomery
- *Experiments with Mixtures* by Cornell.

All the books cited above can be obtained from the publisher - John Wiley and Sons, Inc., New York. The latter three references can also be obtained via Stat-Ease.

We also recommend two paperbacks that you can purchase through Stat-Ease:

- *The Experimenter's Handbook* by Kraber, et al (free to registered users)
- *DOE Simplified* by Anderson and Whitcomb (Productivity Press).

These references are not as detailed, but they cover the basics very well.

If you need a thorough DOE education, consider attending one or more of Stat-Ease's computer-intensive workshops. Call us to get information on course content and schedule. We provide complete details on all outputs in these workshops. Also, don't be shy about calling us for statistical help. In many cases this will be provided at no charge, but if this can't be done, you will be advised on our consulting rates. You will find contact information at the end of the Introduction.

General Topics

Sequence of Analysis

With Design-Expert, you analyze one response at a time by following these steps:

1. Choose the model:
 - For one-factor (categorical) designs this step is skipped. Go to step 2: ANOVA.
 - For two-level factorial designs, choose the model via the full or half-normal plot of effects. For general factorials with pure error estimates, try the Select by Probability option under the View menu. Otherwise, you must manually designate highest order interactions as error and proceed to the next step.
 - For RSM and mixture designs (or a cross) via the sequential F-tests, lack-of-fit tests and other adequacy measures.
2. Do analysis of variance (ANOVA), post-ANOVA analysis of individual model coefficients, and case statistics for analysis of residuals and outlier detection. For factorial designs, if any coefficients have p-values above your critical threshold (we suggest 0.1), then go back to step 1, change them to error, and re-do the ANOVA. This step is optional for RSM and mixture designs.
3. Inspect various diagnostic plots to statistically validate the model.
4. If the model looks all right, generate model graphs for interpretation:
 - For one-factor design, look at the least significant differences (LSD) on a graph of treatment means
 - For factorial design, look at the main effect and interaction graphs and the cube plot.
 - For RSM and mixture designs (or a cross), look at the contours and 3D graphs.

After each response gets analyzed, you move on to multiple response optimization, either by inspection of the interpretation plots, or with the graphical and numerical tools provided for this purpose by Design-Expert.

Transformations

Design-Expert provides the user with a broad range of possible response transformations. The appropriate choice depends on subject matter and/or statistical considerations. The software provides extensive diagnostic capabilities to validate statistical assumptions. For further information on interpreting diagnostic data and choosing transformations, refer to Montgomery's *Design and Analysis of Experiments*. The available transformations are:

- Square root
- Log - base e or 10
- Reciprocal square root
- Inverse
- Power of your choice
- Logit (see below)
- Arcsine square root (see below).

The power transformation allows transformation to any power in the range -3 to +3, provided the data are positive. You may add a constant to the data to avoid powers of negative numbers. If the standard deviation associated with an observation is proportional to the mean raised to the alpha power, then transforming the observation by the one minus alpha power gives a scale satisfying the equal variance requirement of the statistical model. As a feature under Diagnostics, Design-Expert offers a helpful plot, called the Box-Cox, which recommends the appropriate power transformation (including the no transformation option). See the section on Two-Level Factorial Tutorials for a case study that demonstrates use of Box-Cox and provides a few details. For more information on this plot, refer to program Help and/or Montgomery's text.

Use the logit transformation when your response varies within a finite range, such as 0 to 100 percent yield. Logit spreads out the values near the boundaries. The transform is:

$$\text{Logit}(Y) = \log_e[(Y - \text{lower limit})/(\text{upper limit} - Y)]$$

The response must be between the lower and upper limit.

The arcsine square root should be used for binomial data, for example, fraction defective. The transform is:

$$Y' = \sin^{-1} \sqrt{Y}$$

For this transformation to be valid, the response data must be in the form of a proportion between zero and one, from samples of equal size.

Design-Expert offers an option to plot the responses in terms of the original response data. It uses the model to calculate the surface matrix of data points and then applies the inverse transform prior to making the plot.

Model Fitting and Hierarchy

The model-fitting step in Design-Expert uses the QR decomposition algorithm (See Chap. 9, *Linpak User's Guide*, J.J. Dongarra, C.B. Moler, J.R. Bunch, G.W. Stewart, 1979, Siam) on the design matrix (X) to compute model coefficients. All computations are carried out on a standardized version of the design matrix to avoid numerical instabilities as much as possible.

Design-Expert checks the model hierarchy before you can run an ANOVA. For example, if you select an interaction, such as BD, for your model without selecting a main effect “parent” of the interaction, such as B, the resulting model would not be hierarchical. A well-formulated model must include all main effects present in the interactions. If the model fails the hierarchy check, you will be warned and offered a chance to correct this problem. (For details on model hierarchy see “A Property of Well-formulated Polynomial Regression Models,” Julio Peixoto, *The American Statistician*, Feb. 1990, Vol 44, No.1, and John A. Nelder, “The Selection of Terms in Response-Surface Models – How Strong is the Weak-Heredity Principle,” *The American Statistician*, Nov. 1998, V52, No. 4.)

Factorial Designs

Missing Data or Botched Levels

Missing data makes the design unbalanced and non-orthogonal, which introduces undesirable properties to the statistical analysis. However, Design-Expert will still provide a solution to the data that remain but some important effects may be lost to aliasing. The best advice regarding missing data is to run all experiments exactly as planned and get the response data. If you can’t do this, be sure to do a design evaluation and pay particular attention to the aliases.

For designs with missing data, or with lost orthogonality due to edited factor levels, Design-Expert uses the method of least squares in a hierarchical fashion to compute the coefficients, building up from a base that contains the intercept plus block effects (if any). Next the main effects are estimated and corrected for the intercept, block effects (if any) and other main effects. Then the coefficients for the two-factor interactions are generated from least squares estimates of the model containing the intercept, block effects (if any), all main effects and all two-factor interactions. If you edit the design to the degree that all the original factorial effects cannot be estimated, the program will estimate as many as possible. When all terms of a given order cannot be estimated, a subset is selected using forward stepwise regression. This aliasing process will slow down the computations.

For proper use of the normal probability plot, the effects must have a common error variance. Missing data, or altered independent factors, can cause the variance associated with the estimated effects to differ. Therefore the effects must be adjusted (standardized) to correct for this problem. Design-Expert computes standardized effects by multiplying each coefficient by two, and by the ratio of the standard error of the first coefficient computed (usually A) to that of the standard error of the current coefficient (i):

$$\text{Standardized Effect}_i = (\text{Coefficient}_i)(2)(\text{Std. Error}_A/\text{Std. Error}_i)$$

The standard error for effect A is a constant in this equation.

where the x^* refer to individual rows in the overall X matrix. The t-value is based on a risk level of 0.05 for a two-tailed test with the degrees of freedom for error associated with the standard deviation (Std. Dev.) on the ANOVA table. The comparison depends on the standard error of the mean for a given treatment versus the standard error for all other treatment means. Therefore you will see differences in the sizes of the bars for unbalanced designs. The bars narrow as sample size goes up. Look for overlap between pairs of LSD bars. If there is none, then you can say the associated means differ at the 95 percent confidence levels. See the model graph in the One Factor Tutorial for an example.

Response Surface (and Mixture) Designs

Scoring Models for Selection

Design-Expert uses the following system to score models. The data comes from the fit summary step in the analysis sequence.

1. Compute a score (M) from the sequential model sum of squares:
 - $M = 1$ if $p \leq .05$
 - $M = 0.05 / p$ if $p > .05$
 - $M = 0$ if model is aliased
2. Compute a score (L) from the lack-of-fit table:
 - $L = 1$ if $p \geq .10$
 - $L = p / .10$ if $p < .10$
3. Combine the first two scores with r-squared statistics to form an overall score:
 - Score-1 = $(M)(L)(R^2_{\text{predicted}})$
 - Score-2 = $(M)(L)(R^2_{\text{adjusted}})$
 - Select the model with the maximum score. If all model scores are less than or equal to zero, select the mean model.

Note that two scores are computed, based on which r-squared (multiple correlation coefficient) is used: predicted or adjusted. The adjusted r-squared should be familiar to most users. It's a measure of the amount of variation about the mean explained by the model, adjusted for the number of parameters. You obviously want to maximize it's value, which cannot exceed one. The more rigorous predicted r-squared, which may not be reported by all statistics programs, measures the amount of variation in new data explained by the model. The formula is:

$$R^2_{\text{predicted}} = 1 - \text{SSPRESS} / (\text{SSTotal} - \text{SSBlocks})$$

The PRESS (predicted residual sum of squares) a measure of how the model fits each point in the design. The PRESS is computed by first predicting where each point should

be from a model that contains all other points except the one in question. The squared residuals (difference between actual and predicted value) are then summed up. You want to minimize PRESS, which will maximize the predicted r-squared. Unlike other r-squared statistics, the predicted variety can go negative. In this case Design-Expert will revert back to a simpler model.

In most situations, there will be no discrepancy between the two model scores computed by the software. However, you might see two models suggested. Design-Expert then conservatively defaults to the model scored highest on the basis of predicted r-squared. Consider changing the selection to the highest-order model suggested in the fit summary, and then follow up by doing a model reduction. You might discover a few statistically significant higher-order model terms that would otherwise be overlooked.

The ultimate evaluation of the selected model occurs when Design-Expert does the ANOVA. We won't go over the detail of the ANOVA table itself, because it's all standard statistical procedure. However, in the output following the ANOVA table, after the afore-mentioned r-squared values, you will see a relatively little-used statistic called Adequate Precision. This is a measure of the contrast in predicted response relative to its associated error, in other words a signal to noise ratio. Its desired value is 4 or more. The formula is:

$$\text{Adeq Precision} = \left[\frac{\max(\hat{Y}) - \min(\hat{Y})}{\sqrt{\frac{p(\text{Std.Dev.})^2}{n}}} \right]$$

Where p denotes the number of model coefficients and n represents the number of experimental runs.

Linear Mixture Models

For modeling mixtures, Design-Expert uses polynomials that account for all ingredients summing to a fixed total. The mixture models, originally derived by Henri Scheffé, differ most notably from normal RSM models by their absence of an intercept term. Statisticians refer to this as a “canonical” form. (See the textbook by Cornell for details.) We will simply refer to these polynomials as “mixture models.” Linear mixture models present difficulties to the typical regression program because the coefficients cannot be tested in the usual manner. Instead of being compared to zero, the effect of each individual component (E_i) must be tested against the average effect of all other components. When the design space is not a simplex, the component effects must be adjusted for the differences in the individual ranges ($R_i = U_i - L_i$ in pseudo component scale, where U and L are upper and lower constraints, respectively.) The general formula for the “adjusted linear effect” is:

$$E_i = R_i \left[b_i - \sum_{j \neq i} b_j / (q - 1) \right]$$

where q represents the number of components in the mixture.

The linear and quadratic mixture models are subsets of the special cubic. See the reference by Cornell for a more complete discussion of these canonical polynomials.

These equations, used for prediction, have no block effects. Blocking is a restriction on the randomization, used to reduce error in the experiment design. It is not a factor being studied. Blocks are only used to fit the observed experiments, not to make predictions.

Perturbation and Trace Plots

Design-Expert provides the usual contour plots and 3D graphs that you'd expect from a DOE program of this caliber. For examples of these outputs, see the Response Surface Methods (RSM) Tutorials and Mixture Design Tutorials sections. As the number of RSM factors or mixture components increases, it becomes difficult to visualize the response surface even with these sophisticated graphical tools. In this case it's helpful to view a special form of response plot called "perturbation" for RSM data, and the equivalent for mixtures – the "trace" plot.

Perturbation plots help you compare the effect of all the factors at a particular point in the RSM design space. The response is plotted by changing only one factor over its range, while holding all other factors constant. By default, Design-Expert sets the reference point at the midpoint (coded 0) of all of the factors, but this can be modified via the Factors tool.

The trace plot shows the effects of changing each mixture component while holding all others in a constant ratio. The response is plotted while moving along an imaginary line from a reference blend to the vertex of the component being incremented. The default reference blend is the centroid of the design. As the amount of this component increases, the amounts of all other components decrease, but their ratio to one another remains constant. For example, in a three component simplex design, a response trace can be generated as A increases along the line from the overall centroid ($A=1/3$, $B=1/3$, $C=1/3$) toward its pure component vertex (1,0,0). In this case, as the amount of A increases, the amounts of B and C decrease, but the ratio of B to C stays constant at a one-to-one ratio. The trace plots can be done either using the Piepel's direction (using a trace in pseudo units) or in Cox's direction (using a trace in real units). See Cornell's textbook for details.

On either the perturbation or trace plot, a steep slope or curvature in an input variable indicates a relatively high sensitivity of response. These influential variables are good ones to select for the axes on the 2D and 3D contour plots. If the response fits a linear model, you may not need the more sophisticated response plots, because the perturbation (or trace) plot tells the story.

Optimization

The optimization module in Design-Expert searches for a combination of factor levels that simultaneously satisfy the requirements placed on each of several responses. To use optimization, you must first analyze each response to establish the appropriate model.

Graphical Optimization

With multiple responses you need to find regions where requirements simultaneously meet the critical properties. By superimposing critical response contours on a contour plot you can visually search for the best compromise. If you are dealing with many input variables, you may want to do numerical optimization first. See the last parts of the Response Surface Methods (RSM) Tutorials and the Mixture Tutorials for optimization examples.

Design-Expert accepts up to 99 responses initially, but more can be added after creating the design by right-clicking on any response column and choosing Insert Response. Only those responses that have models selected can be optimized. Enter your desired minimums and/or maximums for each response. The program draws a contour corresponding to each constraint, if it appears within the plotting region. Design-Expert then shades the region where the response falls out of specification. The process is repeated for each response with its constraints. Each plotted shading is superimposed on top of the other responses. The un-shaded region that remains gives you the window or “sweet spot” where you can satisfy all response specifications. Set a flag in the window to provide useful details about the predicted outcome.

Numerical Optimization

Design-Expert bases its numerical optimization on an objective function called desirability. The overall desirability (D) is the geometric (multiplicative) mean of all individual desirabilities (d_i) that range from 0 (least) to 1 (most):

$$D = (d_1 \times d_2 \times \dots \times d_n)^{\frac{1}{n}} = \left(\prod_{i=1}^n d_i \right)^{\frac{1}{n}}$$

where n is the number of responses. If any of the responses fall outside their desirability range, the overall function becomes zero. See the Myer and Montgomery textbook for more details.

For simultaneous optimization each variable and response must have a low and high value assigned to each goal. The input variables (factors or components) will automatically default to “in range” goals that keep the solution within the experimental boundaries. (Note: the d_i from “in range” variables are included in the product of the desirability function “ D ,” but unlike all other goals, they do not cause the “ n ” to increase.) You can change the variable goals to “maximum,” “minimum,” “target,” or “is equal to.”

The response goals default to “none,” in which case the response will not be used for the optimization, but any one or more of these responses can be changed to the other goals listed above, except for the “is equal to.” You can optimize only one response if you like, or a single response subject to upper and/or lower constraints on other responses.

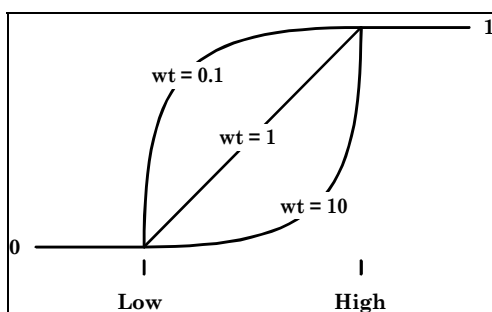
Weights

You can change the shape of the desirability for each goal by the application of “weight.” Weights greater than 1 (maximum 10) give more emphasis to the goal. Weights less than 1 (minimum 0.1) have the opposite effect. The formulas and outcomes for various options can be seen below. These shapes will be displayed by Design-Expert. You can drag the curves with your mouse and change their shape.

For goal of maximum, the desirability curve is defined by the following formula:

$$d_i = \left[\frac{Y_i - \text{Low}_i}{\text{High}_i - \text{Low}_i} \right]^{wt_i} \quad \text{Low}_i < Y_i < \text{High}_i$$

As shown below, the desirability below the low limit falls to 0 and above the high limit plateaus at 1.



Desirability Curves for Goal is Maximum

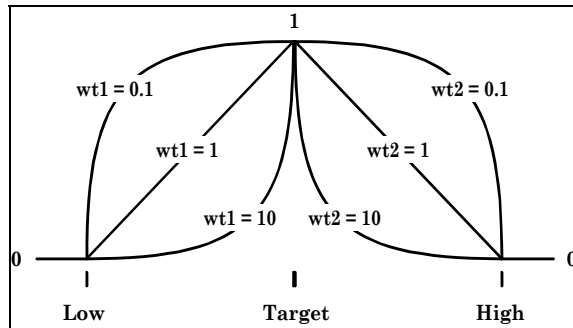
In order to get an appreciable desirability at the heaviest weight of 10, the response must approach the high level. On the other hand, when the weight is reduced to a minimum (0.1), a small increase of response above the low limit is all that’s needed to get a decent desirability.

For goal of minimum, the desirability ramp is defined as:

$$d_i = \left[\frac{\text{High}_i - Y_i}{\text{High}_i - \text{Low}_i} \right]^{wt_i} \quad \text{Low}_i < Y_i < \text{High}_i$$

The shapes at various weights are the mirror image of those shown for the goal as maximum.

For goal as a target, the desirability ramps are created like a maximum on the way up, and a minimum on the way down.



Desirability Curves for Goal is Target

By creative use of weights, you can capture the needs of your clients. For example, it's very common to hear someone say that they want you to hit a specified target, but if you must err, do it on the low or high side. For a goal within range (a constraint), desirability will be defined by the following formulas:

$$d_i = 0 \text{ for } Y_i \leq \text{Low}_i$$

$$d_i = 1 \text{ for } \text{Low}_i < Y_i < \text{High}_i$$

$$d_i = 0 \text{ for } Y_i \geq \text{High}_i$$



Desirability Curves for Goal as Range

The software modifies the desirability function by adding a “tail” as the function evaluation approaches zero. The tail adjustment is reduced with each iteration, thus driving the function towards its pure value. This is a technicality that makes little difference.

Importance

In the desirability objective function $D(X)$, each response can be assigned an importance relative to the other responses. Importance (r_i) varies from the least important (+), a value of 1, to the most important (+++++), a value of 5. If varying degrees of importance are assigned to the different responses, the objective function is:

$$D = (d_1^{r_1} \times d_2^{r_2} \times \dots \times d_n^{r_n})^{\frac{1}{\sum r_i}} = \left(\prod_{i=1}^n d_i^{r_i} \right)^{\frac{1}{\sum r_i}}$$

where n is the number of responses in the measure. If all the importance values are the same, the simultaneous objective function reduces to the normal form for desirability.

Optimization Algorithm

All of these cases can be reduced to a general non-linear algorithm with constraints.

1. Let X , a vector of x_i for $i=1 \dots n$, represent design variables over the optimization space, which is a subset of the design space.
2. Let y_j, U_j, L_j for $j=1 \dots m$ be responses with upper and/or lower bounds serving as constraints.
3. Let $y(X)$ be the response to be optimized. Then $f(X) = y(X)$ for minimization, $f(X) = -y(X)$ for maximization. Define the constraints as a series of discontinuous functions:

- $g_j(X) = y_j(X) - U_j$ for $y_j > U_j$
- $g_j(X) = 0$ for $L_j \leq y_j \leq U_j$
- $g_j(X) = L_j - y_j(X)$ for $y_j < L_j$

This produces a system of m constraints, which can be solved as an unconstrained problem via a penalty function approach:

$$\text{Minimize } \{f(X) + p \sum g_j(X)\}$$

where p is a penalty parameter greater than 0 for $j = 1$ to m . (The penalty parameter p starts at 1 and increases with each iteration by a factor of 100. The number of iterations is limited to 15, which gives a penalty factor of 10^{30} maximum.)

Finding an initial feasible region is one problem with multiple constraints. We start with a small value of a penalty function in a downhill simplex (Nelder-Mead) multi-dimensional pattern search (*Numerical Recipes in Pascal* by William H. Press, et. al., p.326), which converges at either a stationary point or a design space boundary. Limits of the design space are maintained by evaluating the $f(X)$ to +1010 at the design boundaries. The search around the initial convergence point is restarted using a larger penalty function. Convergence is achieved when the distance moved or objective function change is less than a 10^{-6} ratio.

The starting $N+1$ simplex points are constructed by adding or subtracting a fraction of each of the N factor ranges to the initial starting point. The decision to add or subtract is made to maintain a maximum distance from the factor limits.

Let's look at an example of a starting simplex for three factors: factor A goes from -1 to +1 with a range of 2 in coded units, factor B goes from -1.5 to +1.5 with a range of 3 and factor C goes from -0.5 to +0.5 with a range of 1. Assume the randomly selected starting point is (.5, 0, -0.5) in coded units. The starting simplex size defaults to 10%, but it can be changed to any value between 1% to 20% of the factor range. Using the default starting point, the additional simplex points (defining a tetrahedron) are:

- $[.5 - .1(2), 0, -.5] = [.3, 0, -.5]$
- $[.5, 0 + .1(3), -.5] = [.5, .3, -.5]$
- $[.5, 0, -.5 + .1(1)] = [.5, 0, -.4]$

Propagation of Error

Propagation of Error (POE) calculations in Design-Expert are done using matrix algebra. The following section provides the mathematical details. The first step is a big one: Run your DOE and fit an appropriate model to the response, $Y = f(X_1, \dots, X_k, e)$. If the response is non-linear than we can calculate POE. The general form of these calculations are:

1. Enter the standard deviation associated with each factor:

$$\text{Var}(X_i) = \sigma_{ii}^2$$

$$\Sigma_{\text{actual}} = \begin{pmatrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{kk}^2 \end{pmatrix}$$

2. Define the covariance structure:

$$\text{Grad}_{\text{adj}} = \begin{pmatrix} r_{11} & r_{12} & r_{1k} \\ r_{21} & r_{22} & r_{2k} \\ r_{k1} & r_{k2} & r_{kk} \end{pmatrix}$$

$$\Sigma_{\text{adj}} = \text{Grad}_{\text{adj}} \Sigma_{\text{actual}} (\text{Grad}_{\text{adj}})^T$$

$$\Sigma_{\text{adj}} = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \sigma_{1k}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 & \sigma_{2k}^2 \\ \sigma_{1k}^2 & \sigma_{2k}^2 & \sigma_{kk}^2 \end{pmatrix}$$

3. Calculate variance by: $\text{Var}(Y) = \nabla^T \Sigma_{\text{adj}} \nabla + \sigma_e^2$

where:

$$\nabla = \begin{pmatrix} \frac{\partial f}{\partial X_1} \\ \frac{\partial f}{\partial X_2} \\ \frac{\partial f}{\partial X_k} \end{pmatrix}$$

which yields: $\text{Var}(Y) = \sum_{i=1}^k \left(\frac{\partial f}{\partial X_i} \right)^2 \sigma_{ii}^2 + \sum_{i < j}^k 2 \frac{\partial f}{\partial X_i} \frac{\partial f}{\partial X_j} \sigma_{ij}^2 + \sigma_e^2$

$$\sigma_e^2 = \text{MS}_{\text{residual}} \text{ from the ANOVA}$$

4. POE is the square root of the variance: $\text{POE} = \sqrt{\text{Var}(Y)}$

Calculating POE for Process Factors

1. Enter the standard deviation associated with each factor:

$$\Sigma_{\text{actual}} = \begin{pmatrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{kk}^2 \end{pmatrix}$$

2. Define the covariance structure. For process designs we assume the factors are independent:

$$\text{Grad}_{\text{adj}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{I}$$

$$\Sigma_{\text{adj}} = \text{Grad}_{\text{adj}} \Sigma_{\text{actual}} (\text{Grad}_{\text{adj}})^T$$

$$\Sigma_{\text{adj}} = \begin{pmatrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{kk}^2 \end{pmatrix}$$

The model, σ and ∇ are in actual values.

3. Calculate variance:

$$\Sigma_{\text{adj}} = \begin{pmatrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{kk}^2 \end{pmatrix} \quad \nabla = \begin{pmatrix} \frac{\partial f}{\partial X_1} \\ \frac{\partial f}{\partial X_2} \\ \frac{\partial f}{\partial X_k} \end{pmatrix}$$

$$\text{Var}(Y) = \nabla^T \Sigma_{\text{adj}} \nabla + \sigma_e^2$$

$$\text{Var}(Y) = \begin{pmatrix} \frac{\partial f}{\partial X_1} & \frac{\partial f}{\partial X_2} & \frac{\partial f}{\partial X_k} \end{pmatrix} \begin{pmatrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{kk}^2 \end{pmatrix} \begin{pmatrix} \frac{\partial f}{\partial X_1} \\ \frac{\partial f}{\partial X_2} \\ \frac{\partial f}{\partial X_k} \end{pmatrix} + \sigma_e^2$$

$$\text{Var}(Y) = \left(\frac{\partial f}{\partial X_1} \right)^2 \sigma_{11}^2 + \dots + \left(\frac{\partial f}{\partial X_k} \right)^2 \sigma_{kk}^2 + \sigma_e^2$$

4. POE is the square root of the variance: $\text{POE} = \sqrt{\text{Var}(Y)}$

Calculating POE for Mixture Components

1. Enter the standard deviation associated with each factor:

$$\Sigma_{\text{actual}} = \begin{pmatrix} \sigma_{11}^2 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 \\ 0 & 0 & \sigma_{kk}^2 \end{pmatrix}$$

2. Define the covariance structure. For mixture designs the components (use real values) add to one and we assume each component is weighed independently:

$$\text{Actual values: } X_1 + X_2 + X_3 = \text{Total}$$

$$\text{Real values: } X'_1 + X'_2 + X'_3 = 1$$

$$X'_1 = \frac{X_1}{X_1 + X_2 + X_3} \quad X'_2 = \frac{X_2}{X_1 + X_2 + X_3} \quad X'_3 = \frac{X_3}{X_1 + X_2 + X_3}$$

$$\left(\frac{\partial}{\partial x} \right) \left(\frac{u}{v} \right) = \frac{v \frac{\partial u}{\partial x} - u \frac{\partial v}{\partial x}}{v^2}$$

$$\frac{\partial X'_1}{\partial X_1} = \frac{(X_1 + X_2 + X_3)(1) - (X_1)(1)}{(X_1 + X_2 + X_3)^2} = \frac{\frac{X_1 + X_2 + X_3}{X_1 + X_2 + X_3} - \frac{X_1}{X_1 + X_2 + X_3}}{X_1 + X_2 + X_3} = \frac{1 - p_1}{\text{Total}}$$

$$\frac{\partial X'_2}{\partial X_1} = \frac{(X_1 + X_2 + X_3)(0) - (X_2)(1)}{(X_1 + X_2 + X_3)^2} = \frac{-\frac{X_2}{X_1 + X_2 + X_3}}{X_1 + X_2 + X_3} = \frac{-p_2}{\text{Total}}$$

$$\frac{\partial X'_3}{\partial X_1} = \frac{(X_1 + X_2 + X_3)(0) - (X_3)(1)}{(X_1 + X_2 + X_3)^2} = \frac{-\frac{X_3}{X_1 + X_2 + X_3}}{X_1 + X_2 + X_3} = \frac{-p_3}{\text{Total}}$$

$$\text{Grad}_{\text{adj}} = \begin{pmatrix} \frac{1 - p_1}{\text{Total}} & \frac{-p_2}{\text{Total}} & \frac{-p_k}{\text{Total}} \\ \frac{-p_1}{\text{Total}} & \frac{1 - p_2}{\text{Total}} & \frac{-p_k}{\text{Total}} \\ \frac{-p_1}{\text{Total}} & \frac{-p_2}{\text{Total}} & \frac{1 - p_k}{\text{Total}} \end{pmatrix}$$

$$\Sigma_{\text{adj}} = \text{Grad}_{\text{adj}}^T \Sigma_{\text{actual}} (\text{Grad}_{\text{adj}})$$

$$\Sigma_{\text{adj}} = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \sigma_{1k}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 & \sigma_{2k}^2 \\ \sigma_{1k}^2 & \sigma_{2k}^2 & \sigma_{kk}^2 \end{pmatrix}$$

3. Calculate variance by:

$$\text{Var}(Y) = \nabla^T \Sigma_{\text{adj}} \nabla + \sigma_e^2$$

where:

$$\nabla = \begin{pmatrix} \frac{\partial f}{\partial X'_1} \\ \frac{\partial f}{\partial X'_2} \\ \frac{\partial f}{\partial X'_k} \end{pmatrix}$$

which yields:

$$\text{Var}(Y) = \sum_{i=1}^k \left(\frac{\partial f}{\partial X'_i} \right)^2 \sigma_{ii}^2 + \sum_{i < j}^k 2 \frac{\partial f}{\partial X'_i} \frac{\partial f}{\partial X'_j} \sigma_{ij}^2 + \sigma_e^2$$

$$\sigma_e^2 = \text{MS}_{\text{residual}} \text{ from the ANOVA}$$

4. POE is the square root of the variance:

$$\text{POE} = \sqrt{\text{Var}(Y)}$$

The proportions are in real values. The model and ∇ are in actual values. The σ are in actual values.

Calculating POE for a Transformed Response

Y' is the transformed response (Y):

$$Y' = t(Y)$$

The transformed response (Y') is a function of the X s:

$$Y' = g(X)$$

POE in the transformed scale:

$$\sigma_{Y'}^2 = \sum_{i=1}^k \left(\frac{\partial Y'}{\partial X_i} \right)^2 \sigma_{X_i}^2 + \sigma_{\text{resid}}^2$$

Apply the inverse transformation (t^{-1}) to get the response in the original scale:

$$Y = t^{-1}(Y')$$

Calculate POE in the original scale by applying the Chain Rule

$$\frac{\partial Y}{\partial X} = \frac{\partial Y}{\partial Y'} \cdot \frac{\partial Y'}{\partial X}$$

$$\sigma_Y^2 = \sum_{i=1}^k \left(\frac{\partial Y}{\partial X} \right)^2 \sigma_{X_i}^2 + \sigma_{\text{resid}}^2$$