Experimental Design and Optimization with Non-Linear Constraints in JMP

CASEY A. VOLINO
volinoca@corning.com
Technical Mgr. Data Analytics, Sr. Engineering Associate
Corning Incorporated

Currently JMP can incorporate linear constraints or non-linear constraints (via 'Disallowed Combinations') when designing experiments such that all design conditions to test in the experiment conform to specified design space limitations. Once a model is constructed optimization efforts using the Profiler platform in JMP are limited to conforming to only linear constraints. If non-linear constraints were specified, they are ignored for optimization. This can result in extrapolated "optimal" conditions if the fitted model does not have extrema within the constrained design space. A method is presented that allows for optimization with non-linear constraints to proceed and enforces the design-space restrictions they specify.

Key Words: DOE, constraints, non-linear, profiler, optimization, Boolean, logic, desirability, disallowed, combinations, design, experiments, space, filling

1. Overview

The design of experiments (DOE) package in JMP provides leading edge capabilities for modern industrial experimentation. Some notable examples are the Definitive Screening Design (DSD), Custom Design, Space-Filling Design, and the suite of tools for design diagnostics. JMP also makes these tools easy to use to analyze experiments correctly when the error structure should not be partitioned in the standard way. An example is when an experimenter specifies that a factor is "Hard to Change", the appropriate analysis (split-plot ANOVA, etc.) is performed on the resultant DOE results. JMP also allows for the factor space in DOE's to be constrained in ways described by linear combinations of the predictor variables (i.e. "Linear Constraints") and by non-linear combinations of the predictor variables (i.e. "Disallowed Combinations"). If linear constraints are specified, subsequent "What-If" and optimization work in JMP Profilers helps to keep prospective solutions, i.e. factor settings, from disobeying those linear constraints. Currently non-linear constraints are not obeyed in JMP Profilers, but a simple work-around can make this possible. To effectively use this work-around and to create Disallowed Combinations to constrain the design space requires a short explanation of available logic operators in JMP.

2. Review of Logic Operators

JMP offers several operators and functions that evaluate to a Boolean value of True (1) or False (0) that are relevant for constraining DOE's. A listing of some of these operators is shown in Table 1.

Operator	Meaning
=	Equal to
!= or < >	Not equal to
>	Greater than
<	Less Than
>=	Greater than or equal to
<=	Less than or equal to
!	Logical NOT
&	Logical AND (INTERSECTION)
|	Logical OR (UNION)
In Polygon( )	Does a point lie within a polygon?

Table 1 Selected Logic Operators/Functions in JMP

These logic operators can be used to constrain the factor space in useful ways when they are used in a 'Disallowed Combinations' script. As an example, let's construct Space-Filling designs to demonstrate how to constrain the factor space of a Venn diagram. Let A be the left circle and B be the right circle and S by the space the possibilities of A and B reside within.

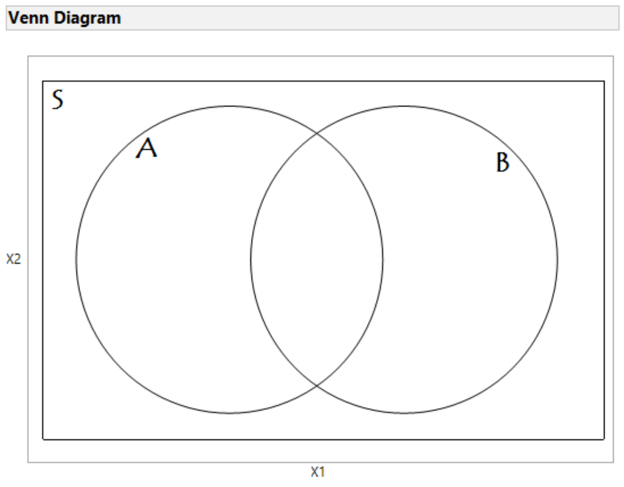


Figure 1 Venn Diagram Example

Now let's construct several logic scenarios. The complete code for the Venn diagram scenarios that follow is given in the appendix, but I'll show the Disallowed Combinations script by each graph for reference.

The following Disallowed Combinations script specifies that we only want region A. We'll populate each region with a 100-point Space Filling design. Our factors (or predictors or features) are X1 and X2.

```
Disallowed Combinations(
	deg = 0 :: 359 :: 1;
	deg = deg`;
	deg = deg * Pi() / 180;

	xCircleA = 15 + 6 * Cos( deg ); //x coordinates for points on circle A
	yCircleA = 17 + 6 * Sin( deg ); //y coordinates for points on circle A
	xCircleB = 22 + 6 * Cos( deg ); //x coordinates for points on circle B
	yCircleB = 17 + 6 * Sin( deg ); //y coordinates for points on circle B

	A = Eval Expr( In Polygon( X1, X2, xCircleA, yCircleA ) );
	B = Eval Expr( In Polygon( X1, X2, xCircleB, yCircleB ) );
	S = Eval Expr( In Polygon( X1, X2, [7.5, 7.5, 30, 30], [10, 24, 24, 10] );
	Eval( !(A) )
)
```

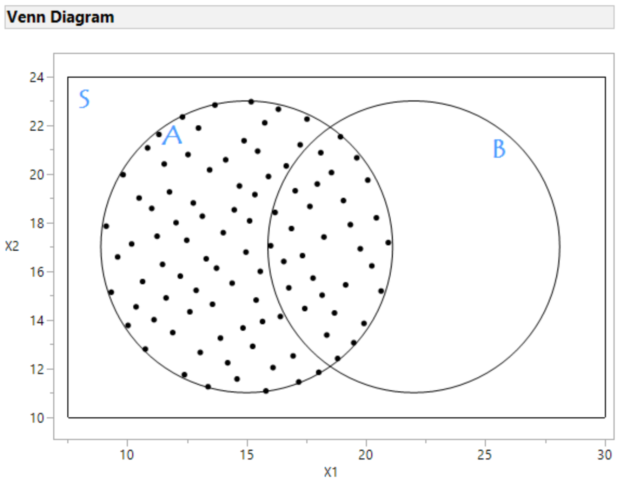


Figure 2 Region A Only

The approach I recommend is to specify the region of interest or Allowed Combinations, i.e. "(A)", and then use the NOT operator, "!", to specify the region that is not of interest or the Disallowed Combinations, i.e. "!(A)". The parentheses are not needed in this example, but I recommend *always* using the syntax "!(specified region of interest)". This will help greatly when the region of interest is complex to specify.

Figure 3 shows what a space filling design constrained to regions A OR B would look like. This would be the union of the area defined by A, with the area defined by B.

```
Disallowed Combinations(
        deg = 0 :: 359 :: 1;
        deg = deg`;
        deg = deg * Pi() / 180;

        xCircleA = 15 + 6 * Cos( deg ); //x coordinates for points on circle A
        yCircleA = 17 + 6 * Sin( deg ); //y coordinates for points on circle A
        xCircleB = 22 + 6 * Cos( deg ); //x coordinates for points on circle B
        yCircleB = 17 + 6 * Sin( deg ); //y coordinates for points on circle B

A = Eval Expr( In Polygon( X1, X2, xCircle1, yCircle1 ) );
        B = Eval Expr( In Polygon( X1, X2, xCircle2, yCircle2 ) );
       S = Eval Expr( In Polygon( X1, X2, [7.5, 7.5, 30, 30], [10, 24, 24, 10] );
        Eval( !(A | B );
)
```

Note that !(A | B) = !A & !B, so that the disallowed portion of the design space would be anything in S that is not A and not B. The NOT operator changes the OR, "|", to an AND, "&".

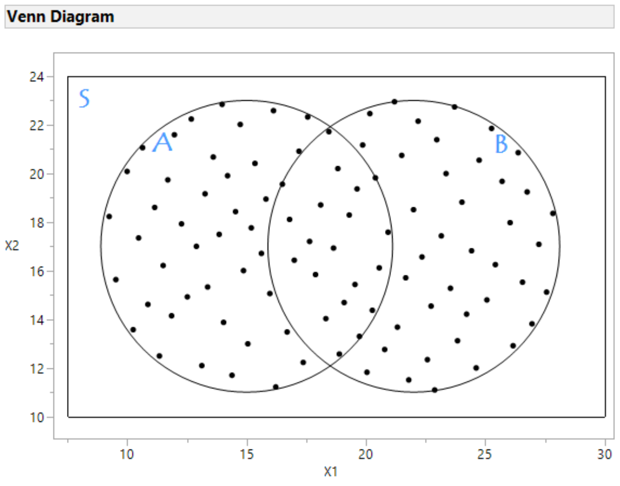
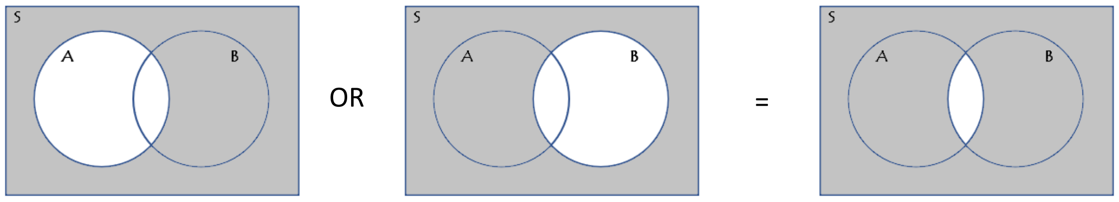


Figure 3 Region A or B

To specify that we only want the intersection of region A and region B, we start with what we want, (A & B) and then the Disallowed Combination would be !(A & B). Note that !(A & B) = !A or !B, which is the union of any areas that are not region A with any areas that are not region B. Graphically, the Disallowed Combinations region is found by forming the union of the regions !A with the region !B.



We could also specify a space filling design for A and B, which is the intersection of A and B.

```
Disallowed Combinations(
       deg = 0 :: 359 :: 1;
       deg = deg`;
       deg = deg * Pi() / 180;

       xCircle1 = 15 + 6 * Cos( deg ); //x coordinates for points on circle 1
       yCircle1 = 17 + 6 * Sin( deg ); //y coordinates for points on circle 1
       xCircle2 = 22 + 6 * Cos( deg ); //x coordinates for points on circle 2
       yCircle2 = 17 + 6 * Sin( deg ); //y coordinates for points on circle 2

       A = Eval Expr( In Polygon( X1, X2, xCircle1, yCircle1 ) );
       B = Eval Expr( In Polygon( X1, X2, xCircle2, yCircle2 ) );
       S = Eval Expr( In Polygon( X1, X2, [7.5, 7.5, 30, 30], [10, 24, 24, 10] ) );
       Eval( !(A & B) );
)
```

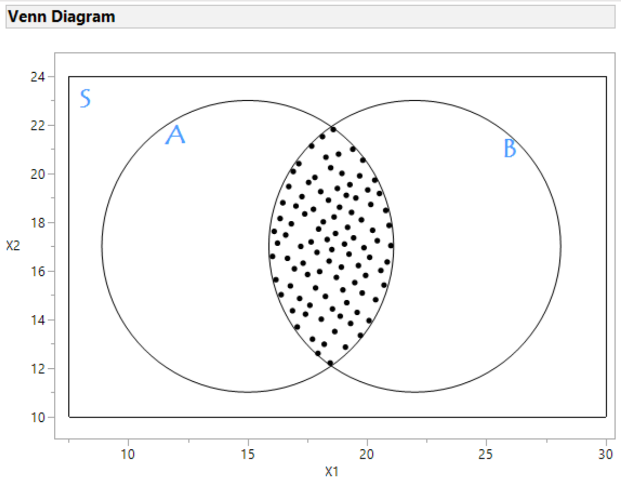


Figure 4 A & B

Furthermore, we could specify only the area in B that is not A.

```
Disallowed Combinations(
       deg = 0 :: 359 :: 1;
       deg = deg`;
       deg = deg * Pi() / 180;

       xCircle1 = 15 + 6 * Cos( deg ); //x coordinates for points on circle 1
       yCircle1 = 17 + 6 * Sin( deg ); //y coordinates for points on circle 1
       xCircle2 = 22 + 6 * Cos( deg ); //x coordinates for points on circle 2
       yCircle2 = 17 + 6 * Sin( deg ); //y coordinates for points on circle 2

       A = Eval Expr( In Polygon( X1, X2, xCircle1, yCircle1 ) );
       B = Eval Expr( In Polygon( X1, X2, xCircle2, yCircle2 ) );
       S = Eval Expr( In Polygon( X1, X2, [7.5, 7.5, 30, 30], [10, 24, 24, 10] ) );
       Eval( !(B & !A ) );
)
```

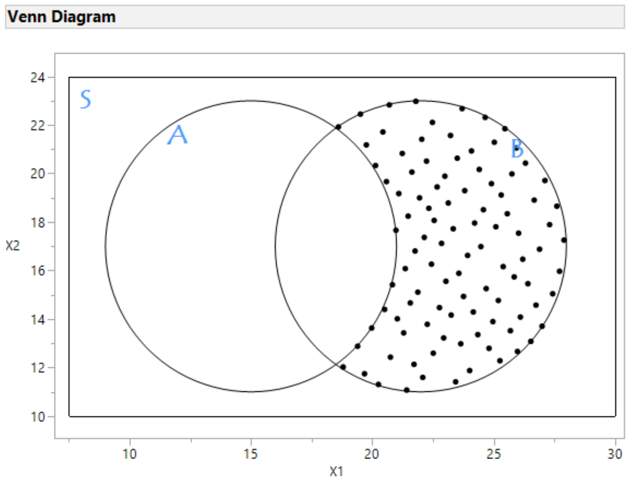


Figure 5 B and not A

Or we could specify anything in S that is not A or B.

```
Disallowed Combinations(
	deg = 0 :: 359 :: 1;
	deg = deg`;
	deg = deg * Pi() / 180;

	xCircle1 = 15 + 6 * Cos( deg ); //x coordinates for points on circle 1
	yCircle1 = 17 + 6 * Sin( deg ); //y coordinates for points on circle 1
	xCircle2 = 22 + 6 * Cos( deg ); //x coordinates for points on circle 2
	yCircle2 = 17 + 6 * Sin( deg ); //y coordinates for points on circle 2

A = Eval Expr( In Polygon( X1, X2, xCircle1, yCircle1 ) );
	B = Eval Expr( In Polygon( X1, X2, xCircle2, yCircle2 ) );
	S = Eval Expr( In Polygon( X1, X2, [7.5, 7.5, 30, 30], [10, 24, 24, 10] ) );
	Eval( !((!A & !B ) & S) );
)
```

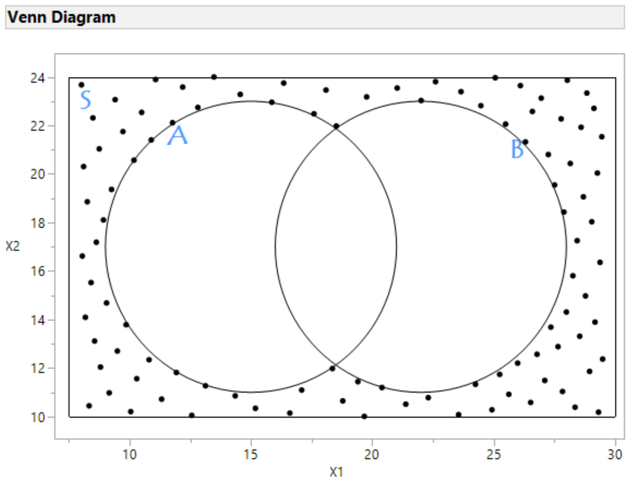


Figure 6 Not A and Not B and S

We had to specify that the points must lie with the region S in this example.

These simple examples demonstrate that we can lay out experimental points in any configuration which conform to specified non-linear arguments. This easily generalizes to more than two dimensions. It should be noted that a recommended technique for constructing Disallowed Combinations is to specify the region of interest (the allowed combinations) and then to negate that expression with the NOT operator, "!".Working with long logical operations can be tricky, so a review of logical operators will be useful.

Working with the NOT operator can sometimes be tricky, so it is important to graph your results to ensure the conditions for your experiment are in the specified regions. Platforms that will suit our needs well are the Scatterplot Matrix and the Contour Plot graphing platforms when we have more than two dimensions.

In summary, to construct Disallowed Combinations first start with the Allowed Combinations (what you want), and then use the NOT operator, "!", to disallow the negative of what you want. Then graph the results in some meaningful way to ensure you have disallowed the regions that are not of interest.

3. The Warmest Place in New York

Let's suppose we grab some weather data for New York from various readings across the state. We can fit various models to that data and then optimize a prediction model to find the warmest place. With some work we can get the data from a website such as http://www.usairnet.com/weather/maps/current/new-york/temperature/ and enter it in JMP along with the latitude and longitude of each site that reports temperature data. A example set of temperatures for New York in September 2018 is shown

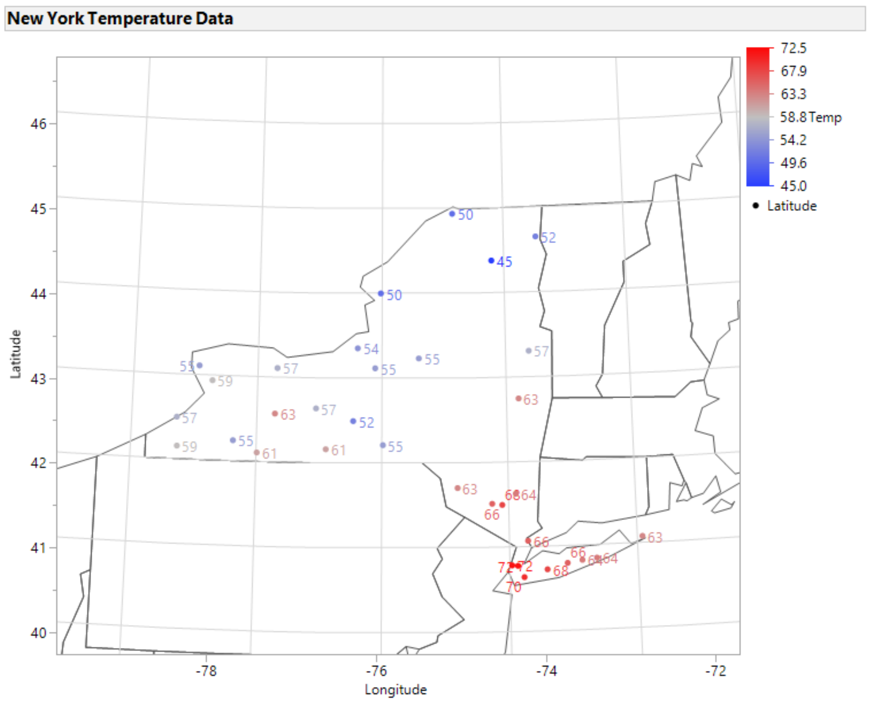


Figure 7 Map of New York Temperatures

We can fit a model to these temperature data using latitude and longitude as predictors. Figure 8 shows the contours from the fitted model overlain on the map of New York.

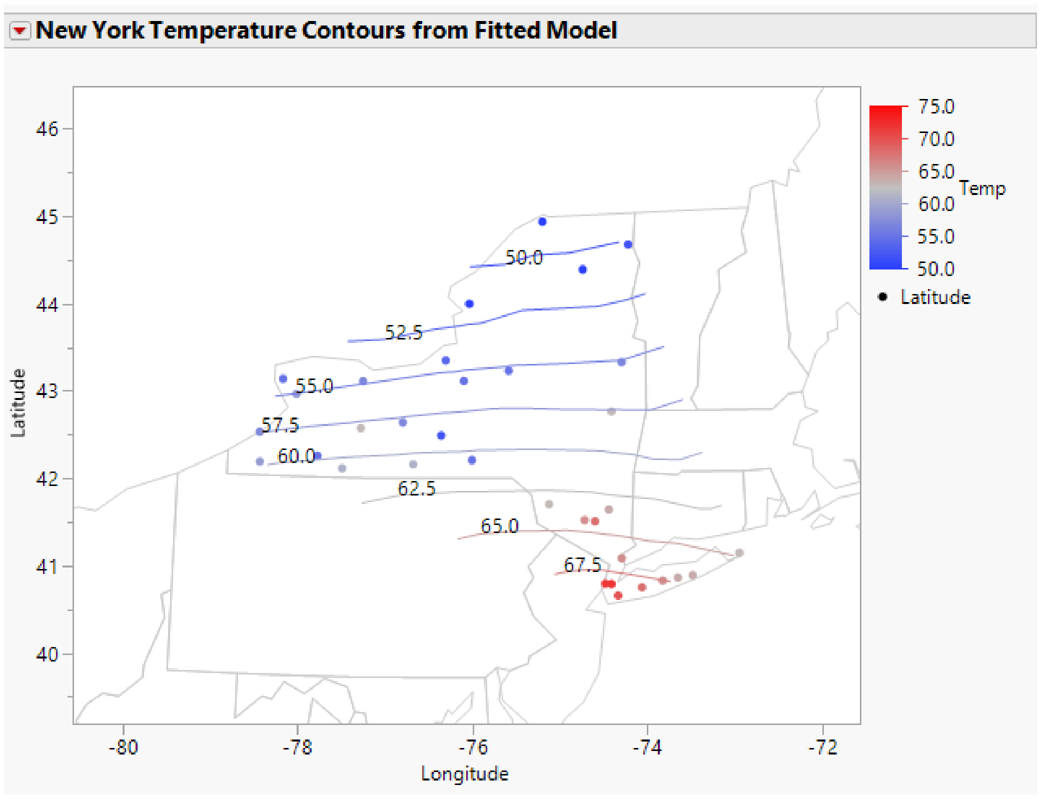


Figure 8 New York Temperature Contours from Fitted Model

To find the warmest place, we can use the fitted model for temperature in a Profiler, turn on Desirability Functions, specify that we want to Maximize the temperature, and then select Maximize Desirability. A solution is shown in the Profiler in Figure 9.

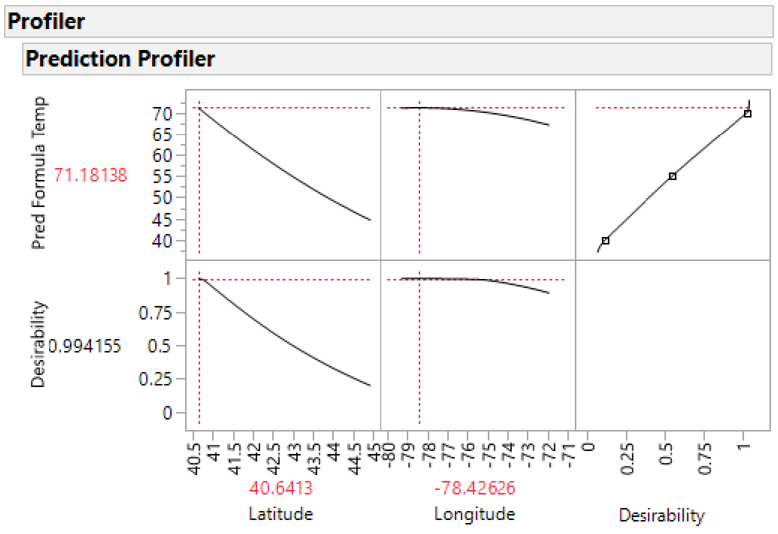


Figure 9 Prediction Profiler Optimization for Max Temperature in New York

Plotting the solution, latitude 40.6413° N, -78.42626 W, we discover that the warmest place in New York is apparently near Altoona, Pennsylvania! The problem is we have not told JMP yet to constrain the solution to the borders of New York state. Unfortunately, while we can specify linear constraints in a JMP Profiler, we cannot add non-linear constraints.

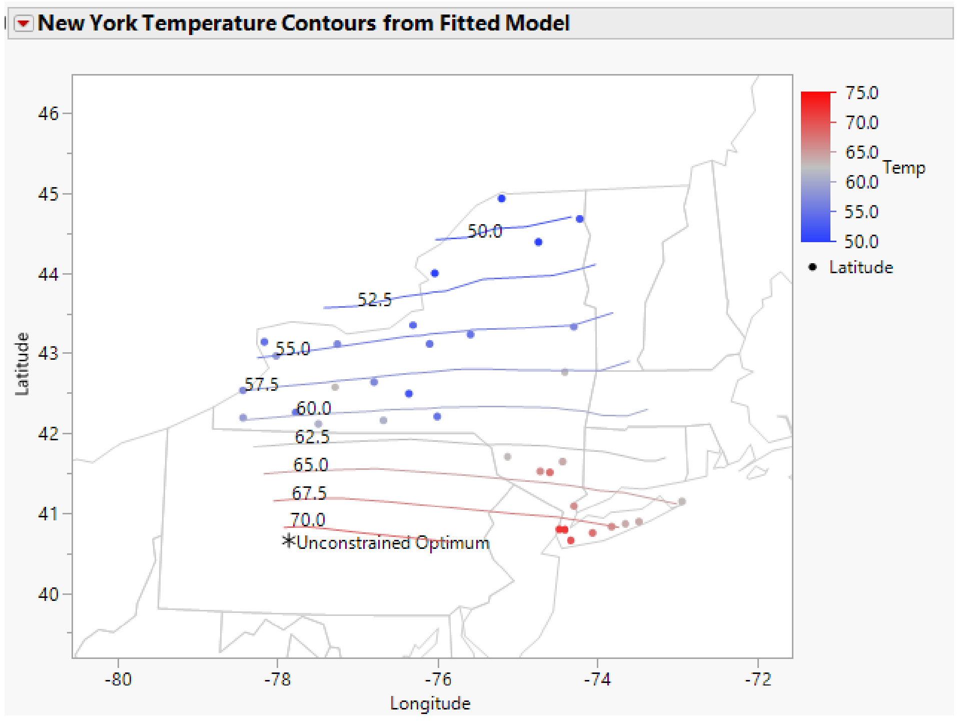


Figure 10 Unconstrained Optimum

The work-around is straight-forward and will be covered in the next section in detail.

4. The Warmest Place in New York - Revisited

To constrain our solution for the warmest place in New York, we need start by defining the region of interest, everything within the borders of New York. Defining the border of New York is straight-forward using an approach published by Jones and Leikivetz, in Issue 29 Summer 2014 edition of **JMPer Cable** (https://community.jmp.com/t5/JMPer-Cable/Fast-Flexible-Filling-in-Space-Filling-Designs/ba-p/21185). I present their instructions here with updated directory info.

Get the Map Shape Coordinates:

JMP installs a set of data tables that contain geographical map data. Each map consists of two JMP data tables named with a common prefix:

- The **-Name.jmp** data table contains the unique names for the different regions.
- The **-XY.jmp** data table contains the latitude and longitude coordinates for the regions.

Map files are installed in Maps directory for your version of JMP. On Windows, look in C:/Program Files/SAS/JMP/14/ Maps (or in the JMPPro/14/Maps folder). On Macintosh, look in /Library/ Application Support/JMP/14/Maps.

Step 1.1: Open **US-State-Name.jmp** from the Maps folder. Look for New York and notice the shape ID is 33.

Step 1.2. Open **US-State-XY.jmp**. Scroll down to row 1819 and copy all rows for shape ID 33 into a new data table. You can do this by highlighting all the rows from 1819 to row 1868, then from the Tables menu select Subset and then click the OK button.

Step 1.3. Run the following script to get the coordinates in a format that the Space Filling Design platform can use.

```
Names Default to Here(1);

dt = Current Data Table();
mymap = dt << Get as Matrix( {X, Y} );
xx = mymap[0,1];
yy = mymap[0,2];

Show(min(xx), max(xx));
Show(min(yy), max(yy));

Show(xx, yy);
```

In the Log you should see the following output:

```
Min(xx) = -79.761951;
Max(xx) = -71.87391;
Min(yy) = 40.545175;
Max(yy) = 45.01585;
```

```
xx = [-73.343124, -73.390231, -73.293613, -73.43774, -73.350593, -73.431229,
      -73.241589, -73.264957, -73.487314, -73.550961, -73.482709, -73.727775,
      -73.657336, -73.7813692570467, -73.485365, -73.229285, -73.110368, -72.585327,
      -72.278789, -72.10216, -71.87391, -72.469996, -73.23914, -73.934512,
      -74.024543, -73.893979, -74.694914, -74.983341, -75.072172, -75.359579,
      -79.761374, -79.761951, -79.148723, -78.853455, -79.074467, -79.070469,
      -78.488857, -77.760231, -77.534184, -76.794708, -76.417581, -76.217958,
      -76.282719906663, -76.125023, -76.360306, -76.312647, -75.912985, -75.283136,
      -74.826578, -74.731301];

yy = [45.01084, 44.618353, 44.440559, 44.045006, 43.771939, 43.588285, 43.534973,
      42.74594, 42.049638, 41.295422, 41.21276, 41.100696, 40.985171,
      40.7949069300726, 40.946397, 40.905121, 40.971938, 40.997587, 41.158722,
      40.991509, 41.052278, 40.84274, 40.6251, 40.545175, 40.709436, 40.997197,
      41.357423, 41.480894, 41.813732, 41.999445, 41.999067, 42.26986, 42.553672,
      42.783958, 43.077855, 43.262454, 43.374763, 43.341161, 43.234569, 43.309632,
      43.521285, 43.545156, 43.858600621629, 43.912773, 44.070907, 44.199044,
      44.368084, 44.849156, 45.01585, 44.990422];
```

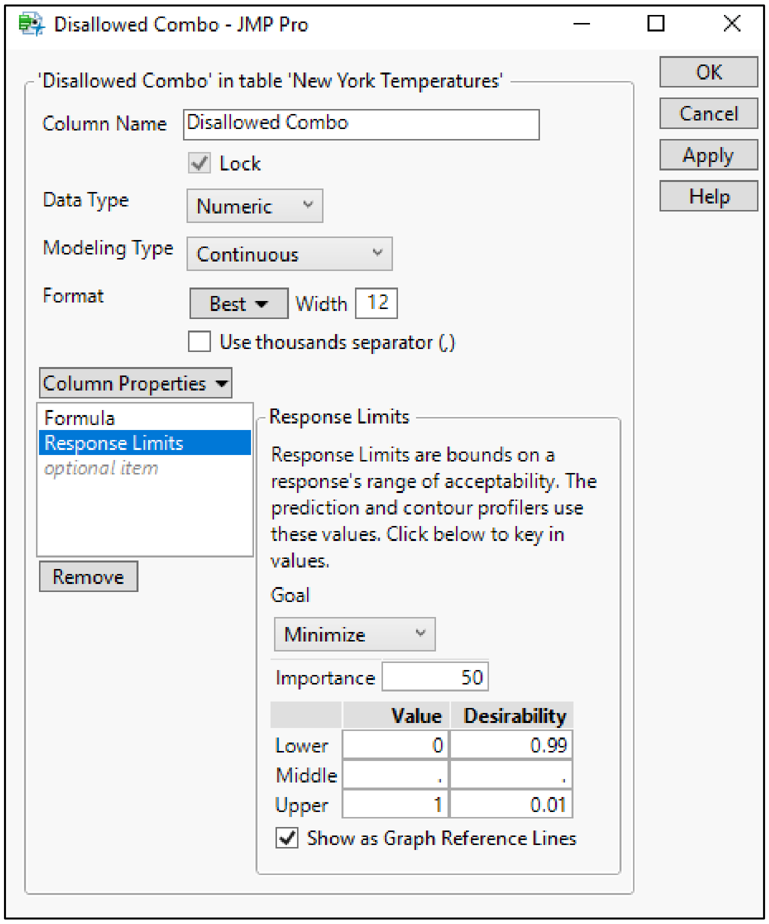
Next we will construct a new response column in the data table with our temperature data and call it ‘Disallowed Combinations’. To this column we’ll add a formula that evaluates to a Boolean value, 0 if the latitude and longitude coordinates are within the state of New York or 1 if they are not. The formula script is given below:

```
xx = [-73.343124, -73.390231, -73.293613, -73.43774, -73.350593,
-73.431229, -73.241589, -73.264957, -73.487314, -73.550961,
-73.482709, -73.727775, -73.657336, -73.7813692570467, -73.485365,
-73.229285, -73.110368, -72.585327, -72.278789, -72.10216, -71.87391,
-72.469996, -73.23914, -73.934512, -74.024543, -73.893979, -74.694914,
-74.983341, -75.072172, -75.359579, -79.761374, -79.761951,
-79.148723, -78.853455, -79.074467, -79.070469, -78.488857,
-77.760231, -77.534184, -76.794708, -76.417581, -76.217958,
-76.282719906663, -76.125023, -76.360306, -76.312647, -75.912985,
-75.283136, -74.826578, -74.731301];

yy = [45.01084, 44.618353, 44.440559, 44.045006, 43.771939, 43.588285,
43.534973, 42.74594, 42.049638, 41.295422, 41.21276, 41.100696,
40.985171, 40.7949069300726, 40.946397, 40.905121, 40.971938,
40.997587, 41.158722, 40.991509, 41.052278, 40.84274, 40.6251,
40.545175, 40.709436, 40.997197, 41.357423, 41.480894, 41.813732,
41.999445, 41.999067, 42.26986, 42.553672, 42.783958, 43.077855,
43.262454, 43.374763, 43.341161, 43.234569, 43.309632, 43.521285,
43.545156, 43.858600621629, 43.912773, 44.070907, 44.199044,
44.368084, 44.849156, 45.01585, 44.990422];

!In Polygon( :Longitude, :Latitude, xx, yy );
```

Next we all a Response Limits column property with the settings shown in Figure 11. We must set the importance very high, say 50, with a ‘Minimize’ goal. The desirability setting for a lower bound of 0 we’ll set to 0.99 (on a scale of 0 to 1) and the desirability setting for an upper bound of 1 we’ll set to 0.01. Click the OK button to continue to the next step.


Disallowed Combo - JMP Pro

'Disallowed Combo' in table 'New York Temperatures'

Column Name: Disallowed Combo

✓ Lock

Data Type: Numeric

Modeling Type: Continuous

Format: Best Width 12

☐ Use thousands separator (,)

Column Properties: Formula, Response Limits

Response Limits

Response Limits are bounds on a response's range of acceptability. The prediction and contour profilers use these values. Click below to key in values.

Goal: Minimize

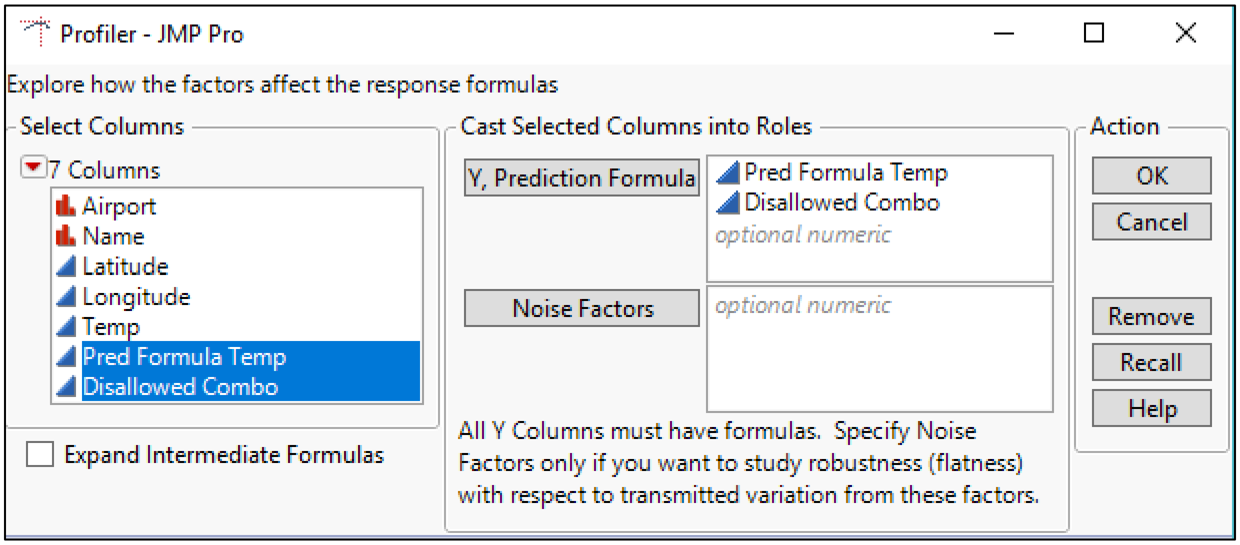
Importance: 50

	Value	Desirability
Lower	0	0.99
Middle	.	.
Upper	1	0.01

✓ Show as Graph Reference Lines

Figure 11 Setting Column Property for Disallowed Combo Column

We proceed by relaunching the Profiler platform and this time we'll add two responses, 'Pred Formula Temp' and 'Disallowed Combo'. Click the OK button to launch the Profiler.


Profiler - JMP Pro

Explore how the factors affect the response formulas

Select Columns: 7 Columns — Airport, Name, Latitude, Longitude, Temp, Pred Formula Temp, Disallowed Combo

Cast Selected Columns into Roles

Y, Prediction Formula: Pred Formula Temp, Disallowed Combo

Noise Factors: optional numeric

☐ Expand Intermediate Formulas

All Y Columns must have formulas. Specify Noise Factors only if you want to study robustness (flatness) with respect to transmitted variation from these factors.

Figure 12 Launching the Profiler Platform with an Added 'Disallowed Combo' Response Column

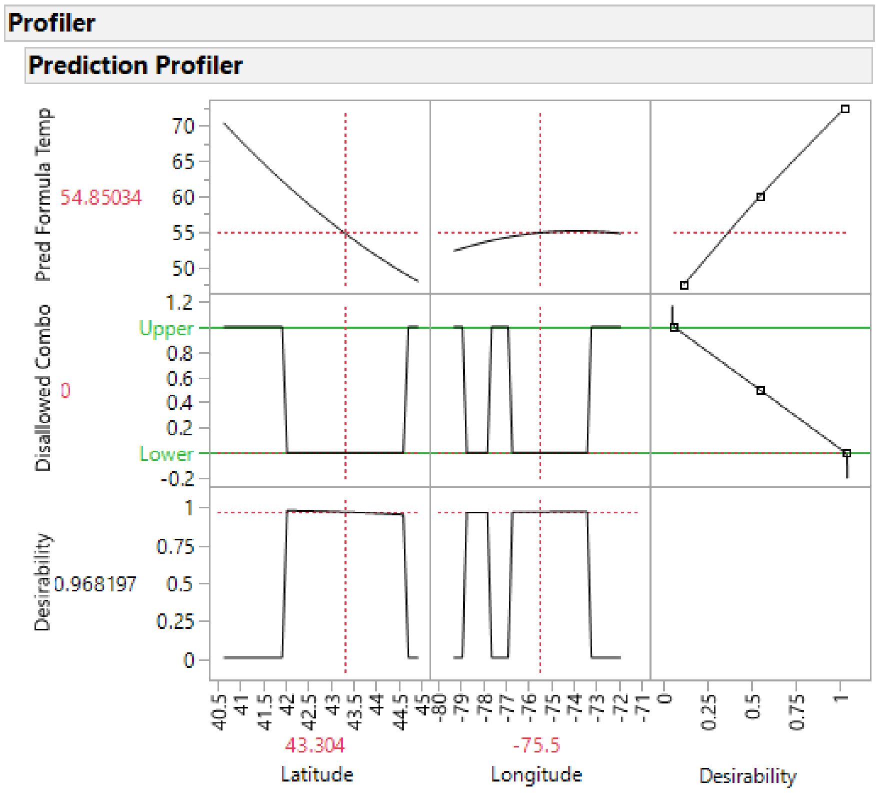


Figure 13 Profiler Platform with Disallowed Combo Response Included

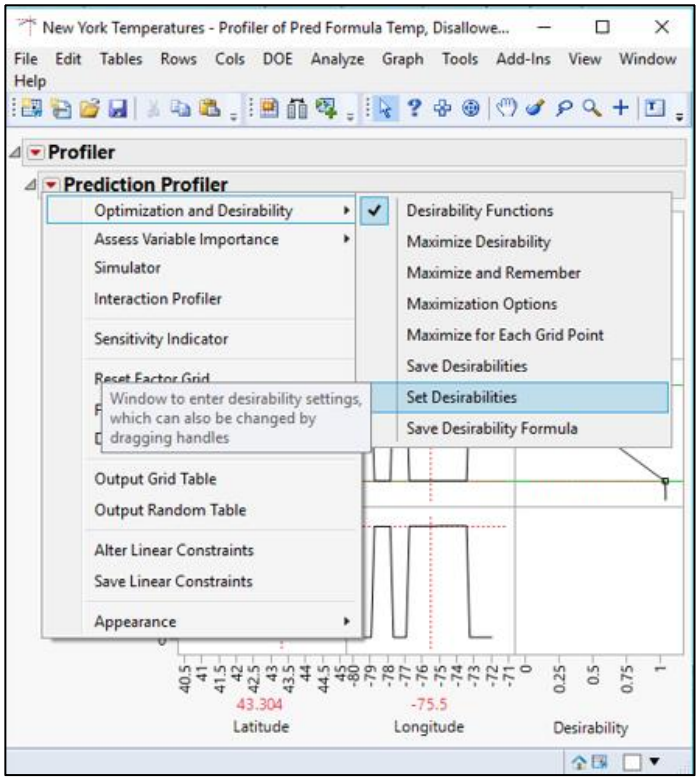
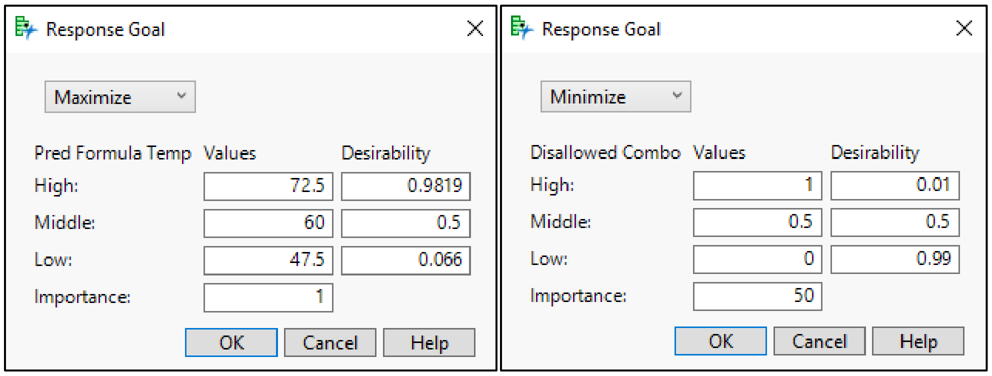


Figure 14 Launching the Set Desirabilities Dialog

We can access the ‘Set Desirabilities’ dialog by clicking the red arrow next to ‘Prediction Profiler’ and selecting ‘Optimization and Desirability’ and then ‘Set Desirabilities’. In Figure 15 note that the goal is to maximize “Pred Formula Temp” and that the Importance of that goal is set to 1, while the goal is to minimize “Disallowed Combo” and that the importance is set to 50. Essentially these response goals are saying, “Once you’re absolutely sure you are within New York (Importance = 50), find the location that maximizes the temperature (Importance = 1).”



Response Goal: Maximize

Pred Formula Temp	Values	Desirability
High:	72.5	0.9819
Middle:	60	0.5
Low:	47.5	0.066
Importance:	1	

Response Goal: Minimize

Disallowed Combo	Values	Desirability
High:	1	0.01
Middle:	0.5	0.5
Low:	0	0.99
Importance:	50	

Figure 15 Desirability Settings for ‘Pred Formula Temp’ and ‘Disallowed Combo’

Once these values are set for the response goals, we are finally ready to optimize, i.e. find the warmest place in New York. From the red arrow next to Prediction Profiler select ‘Optimization and Desirability’ and then ‘Maximize Desirability’.

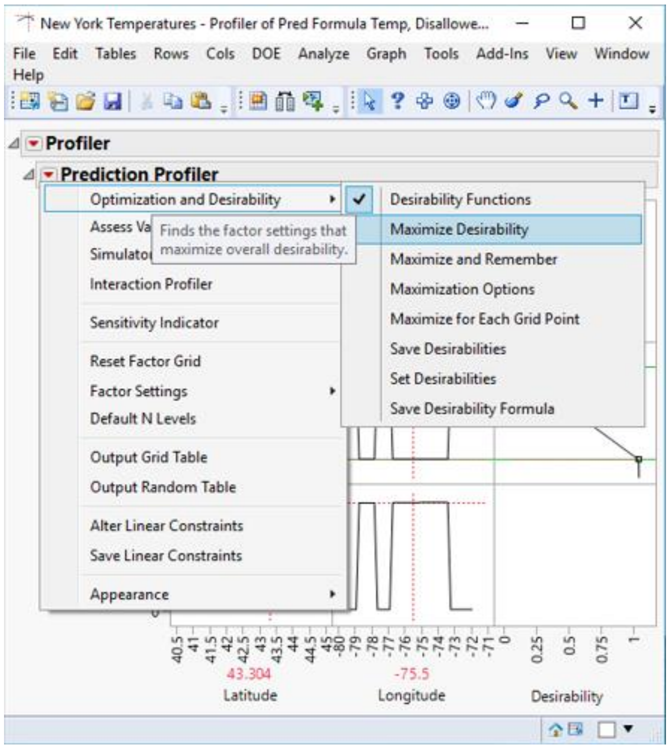


Figure 16 Maximizing the Desirability

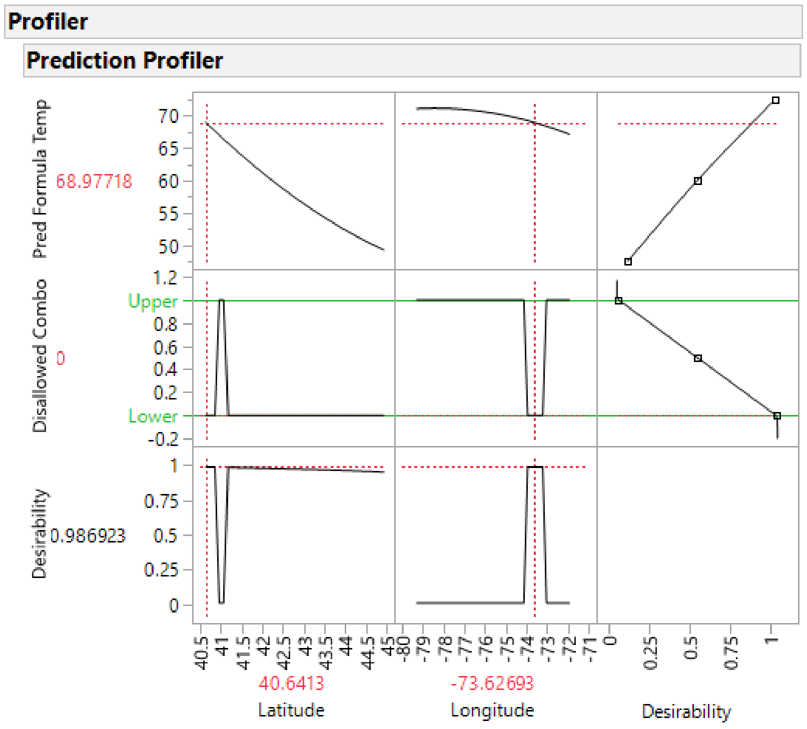


Figure 17 Profiler After 'Maximize Desirability'

To see where the predicted location of the warmest temperature is within the state of New York we can save this solution to the data table by selecting ‘Append Settings to Table’ from the ‘Factor Settings’ menu item.

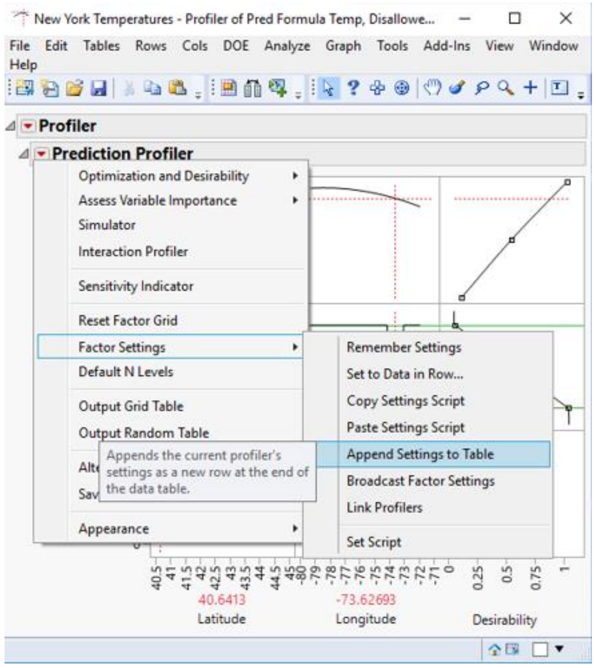


Figure 18 Append the Solution Settings to the Data Table

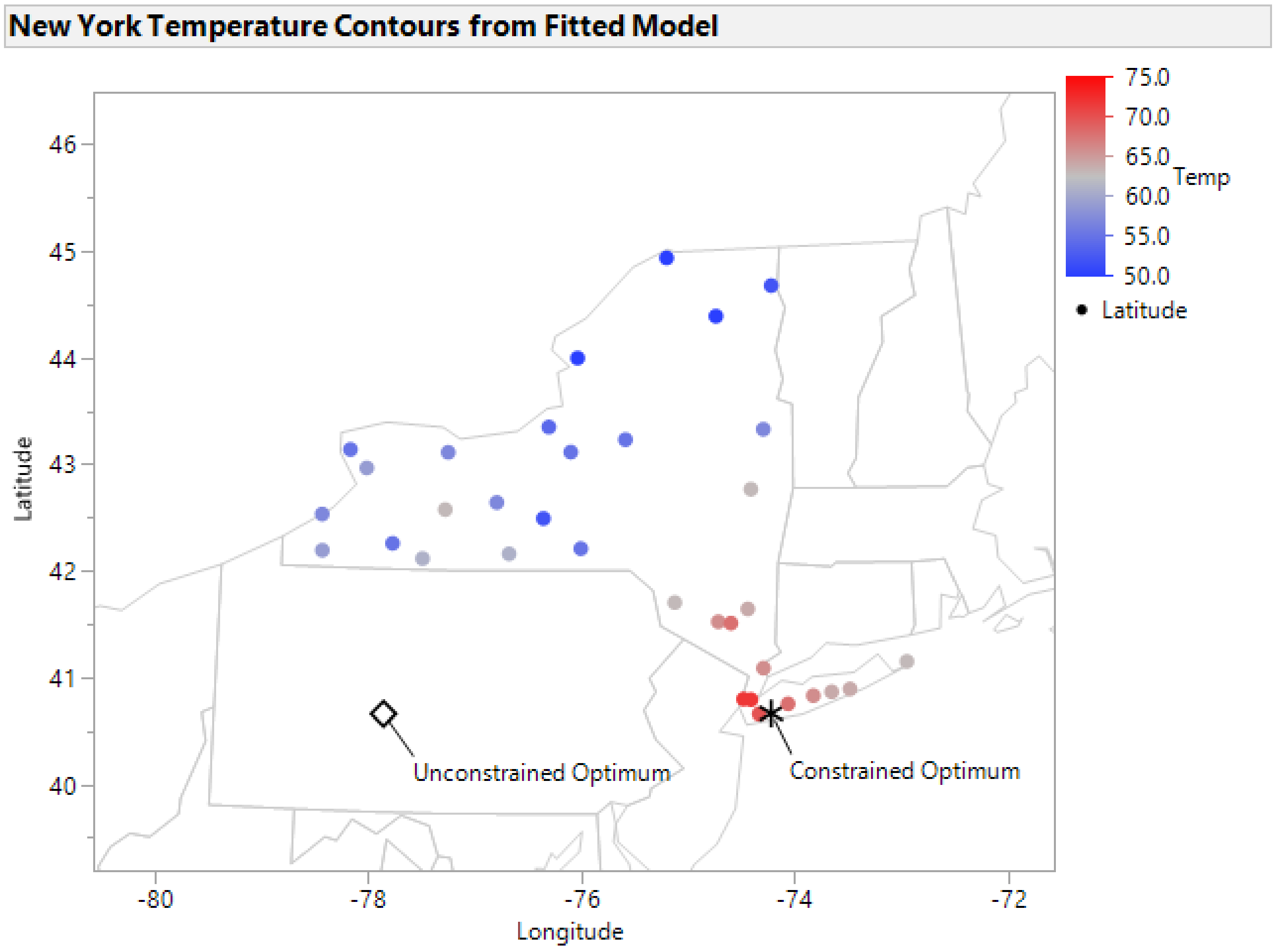


Figure 19 Constrained and Unconstrained Optimum

Finally, the constrained solution for the warmest place in New York state lies within the state boundaries and makes sense relative to the surrounding temperature data.

5. Space-Filling Design with Non-Linear Constraints

Space-Filling designs can be constructed which conform to non-linear constraints. Let's suppose we have a two-dimensional design space with the following non-linear constraints:

$$Y > -8 + 2X + 0.5X^2 + 0.7X^3, \text{ and}$$

$$Y < 30 + 0.5X - X^2$$

These constraints cannot be specified as linear constraints because of the higher-order terms. In words Y must be greater than some curve, AND Y must be less than another curve. Constraints such as these come into play for a variety of reasons such as avoiding excess vibrations, avoiding excess vibrations, avoiding undesirable process conditions due to machine or material wear out rates, avoiding high-expense conditions, etc. Let's further suppose that the true but unknown response surface for the response Z takes the form:

$$Z = (X - 3)^2 + (Y + 11)^2$$

If we wanted to minimize Z, clearly X = 3 and Y = -11 would suffice. However, it is apparent from Figure 20 that the global minimum at (3, -11) is outside the region between the two non-linear constraints. Generally, we would not know this ahead of time and would need to find an empirical equation to form contours from the data.

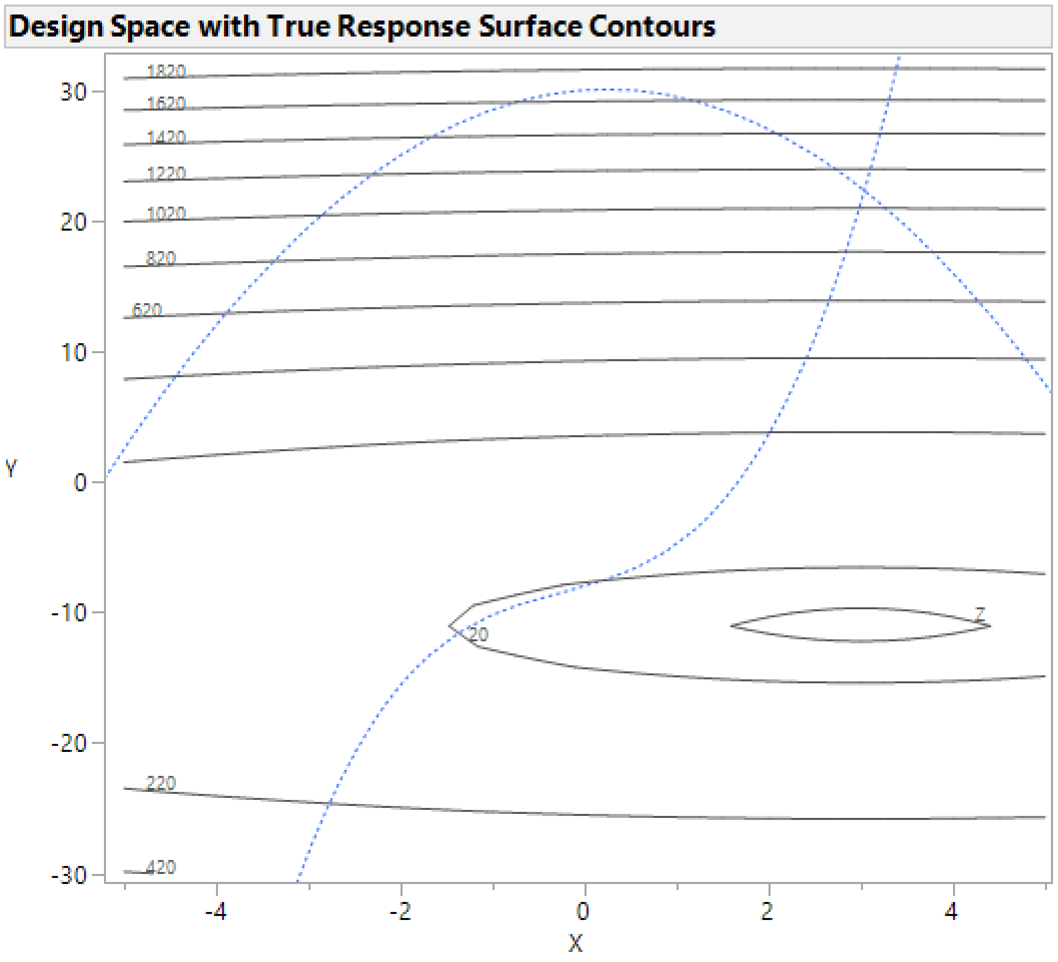
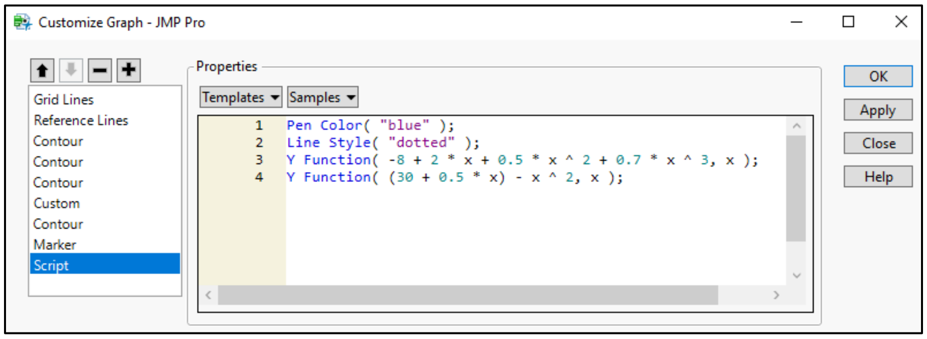


Figure 20 Constrained Design Space with True Contours

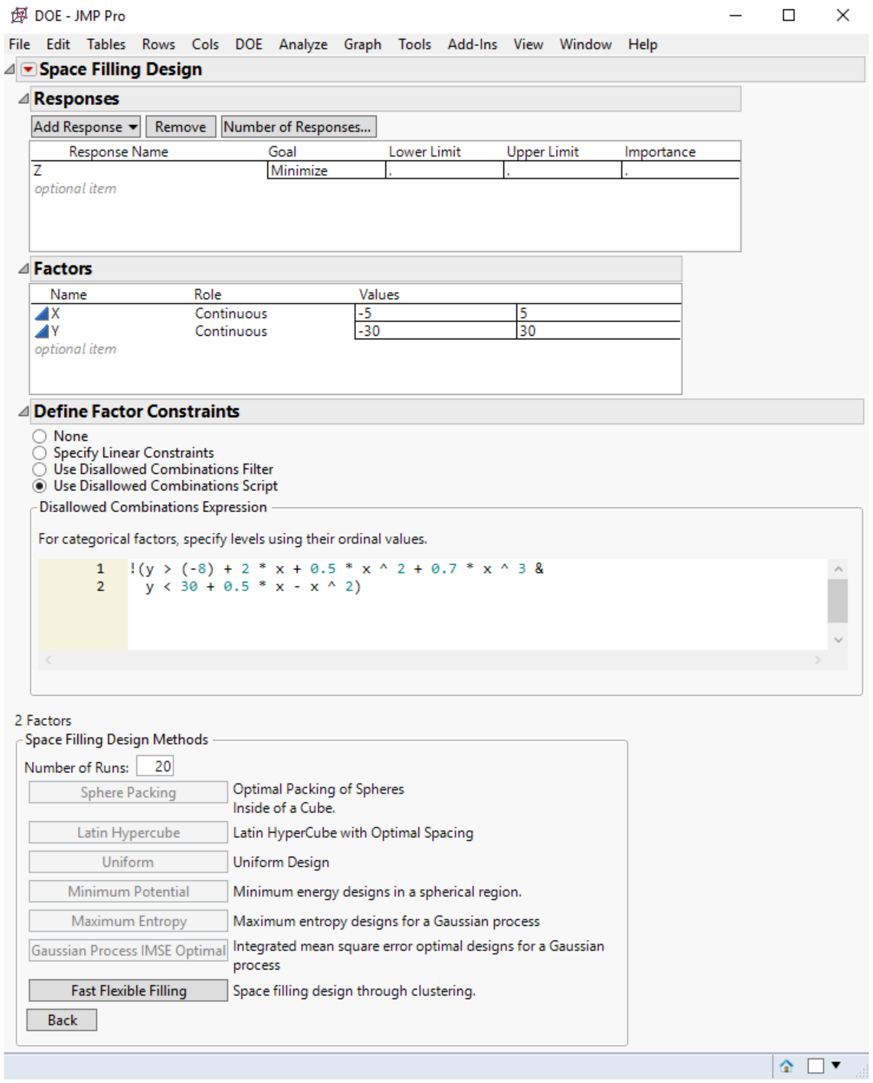
We can add the constraints, shown as dotted lines, to the graph by right-clicking in the body of the graph and choosing Customize. The following script shown in Figure 21 plots the constraints as dotted lines.



```
Pen Color( "blue" );
Line Style( "dotted" );
Y Function( -8 + 2 * x + 0.5 * x ^ 2 + 0.7 * x ^ 3, x );
Y Function( (30 + 0.5 * x) - x ^ 2, x );
```

Figure 21 Script to Plot Non-Linear Constraints

First, we'll set up a 20-run space-filling design in X and Y with the goal of minimizing Z and with the non-linear constraints defined as disallowed combinations.



Response Name	Goal	Lower Limit	Upper Limit	Importance
Z	Minimize	.	.	.
optional item				

Factors

Name	Role	Values	
X	Continuous	-5	5
Y	Continuous	-30	30
optional item			

Define Factor Constraints

( ) None
( ) Specify Linear Constraints
( ) Use Disallowed Combinations Filter
(●) Use Disallowed Combinations Script

Disallowed Combinations Expression

For categorical factors, specify levels using their ordinal values.

```
!(y > (-8) + 2 * x + 0.5 * x ^ 2 + 0.7 * x ^ 3 &
  y < 30 + 0.5 * x - x ^ 2)
```

2 Factors

Space Filling Design Methods

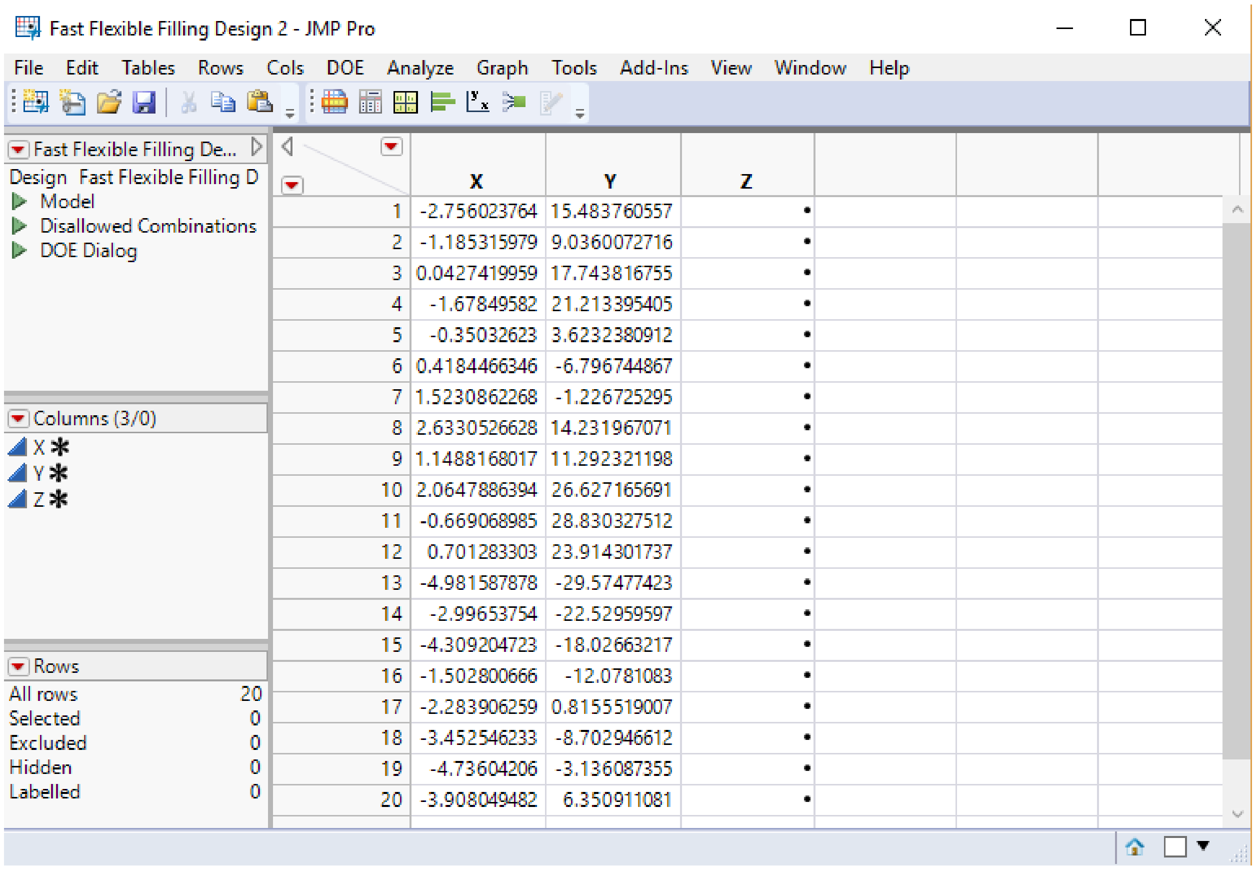
Number of Runs: 20

Sphere Packing	Optimal Packing of Spheres Inside of a Cube.
Latin Hypercube	Latin HyperCube with Optimal Spacing
Uniform	Uniform Design
Minimum Potential	Minimum energy designs in a spherical region.
Maximum Entropy	Maximum entropy designs for a Gaussian process
Gaussian Process IMSE Optimal	Integrated mean square error optimal designs for a Gaussian process
Fast Flexible Filling	Space filling design through clustering.

Figure 22 Space Filling DOE Dialog with Disallowed Combinations Script

As before, for the Disallowed Combinations script we first specify the region we want, and then use the NOT operator to specify regions we do not want. Once we click the Make Table button to create the design, three scripts are automatically added to the new table.

1. Model – Script to create a Gaussian Process interpolation model.
2. Disallowed Combinations – The script for the Disallowed Combinations
3. DOE Dialog – Allows you to recreate the design in the future with all initial options.


Fast Flexible Filling Design 2 - JMP Pro
File Edit Tables Rows Cols DOE Analyze Graph Tools Add-Ins View Window Help
Fast Flexible Filling De...
Design Fast Flexible Filling D
Model
Disallowed Combinations
DOE Dialog
Columns (3/0)
X
Y
Z

Rows	
All rows	20
Selected	0
Excluded	0
Hidden	0
Labelled	0

	X	Y	Z
1	-2.756023764	15.483760557	•
2	-1.185315979	9.0360072716	•
3	0.0427419959	17.743816755	•
4	-1.67849582	21.213395405	•
5	-0.35032623	3.6232380912	•
6	0.4184466346	-6.796744867	•
7	1.5230862268	-1.226725295	•
8	2.6330526628	14.231967071	•
9	1.1488168017	11.292321198	•
10	2.0647886394	26.627165691	•
11	-0.669068985	28.830327512	•
12	0.701283303	23.914301737	•
13	-4.981587878	-29.57477423	•
14	-2.99653754	-22.52959597	•
15	-4.309204723	-18.02663217	•
16	-1.502800666	-12.0781083	•
17	-2.283906259	0.8155519007	•
18	-3.452546233	-8.702946612	•
19	-4.73604206	-3.136087355	•
20	-3.908049482	6.350911081	•

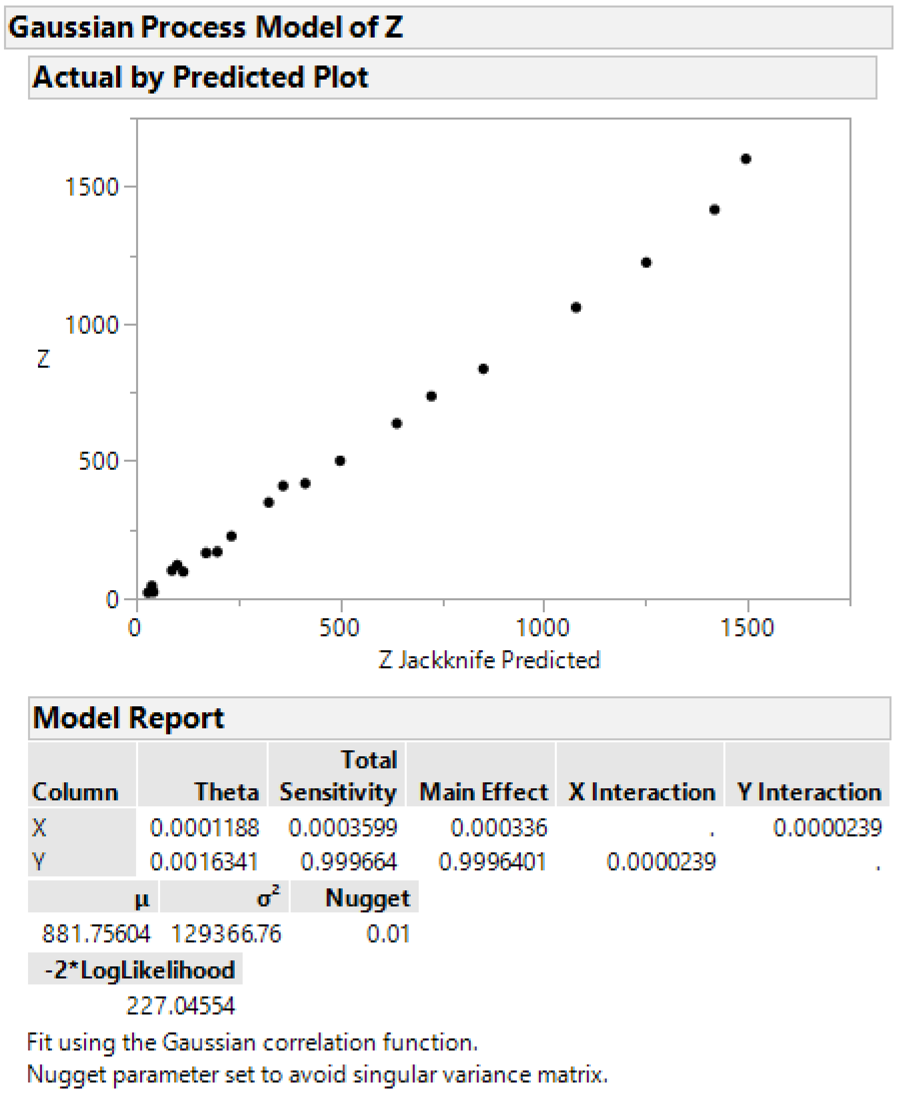
Figure 23 Space-Filling Design

For the purposes of this demonstration Z was specified as:

$$Z = (X - 3)^2 + (Y + 11)^2) + \text{Random Normal}()$$

The **Random Normal()** term adds a bit of noise to the response from a Normal distribution with mean 0 and variance 1.

While we can fit any type of model we'd like, let's use the default Gaussian Process model script, 'Model', that is generated with the space filling design. Figure 24 shows the results.



Model Report

Column	Theta	Total Sensitivity	Main Effect	X Interaction	Y Interaction
X	0.0001188	0.0003599	0.000336	.	0.0000239
Y	0.0016341	0.999664	0.9996401	0.0000239	.

μ	σ^2	Nugget
881.75604	129366.76	0.01

-2*LogLikelihood
227.04554

Fit using the Gaussian correlation function.
Nugget parameter set to avoid singular variance matrix.

Figure 24 Result of Running the 'Model' Script

After fitting the Gaussian Process model, the prediction formula must be stored in a new column in the data table. Finally, we can copy the “Disallowed Combinations” script from the data table, and add a new column called “Disallowed Combo” and paste the script into the Formula property of that column. Now that we have the prediction formula for Z and the disallowed combinations formula for ‘Disallowed Combo’ we can launch add both responses to a profiler.

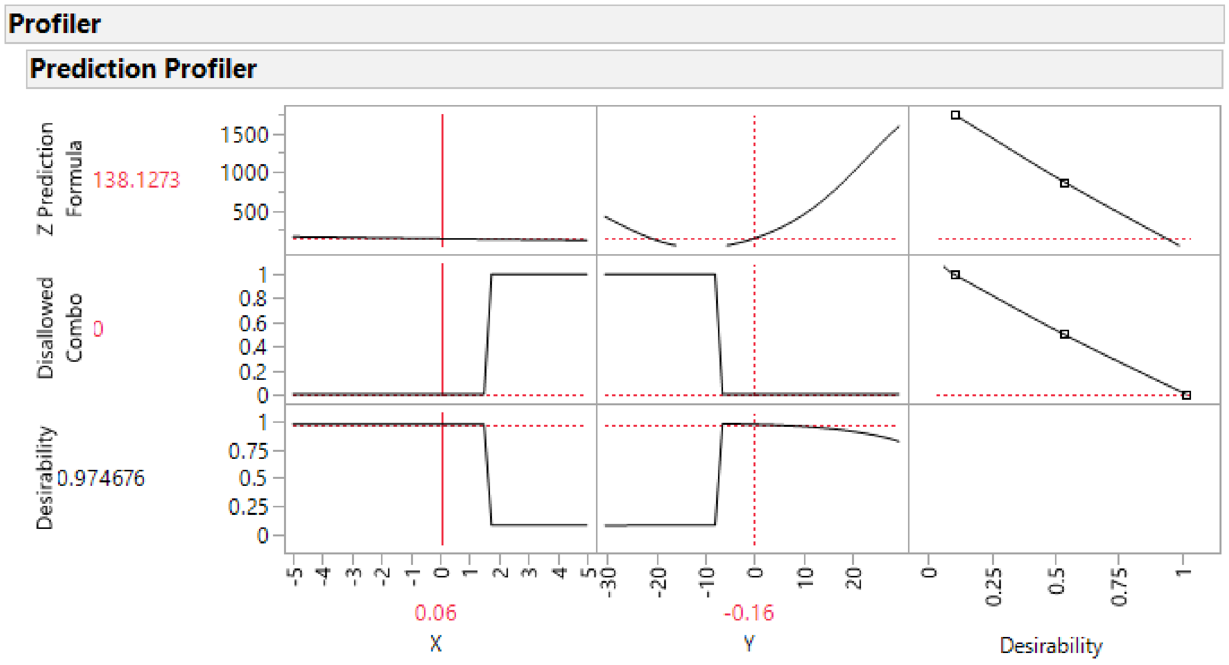
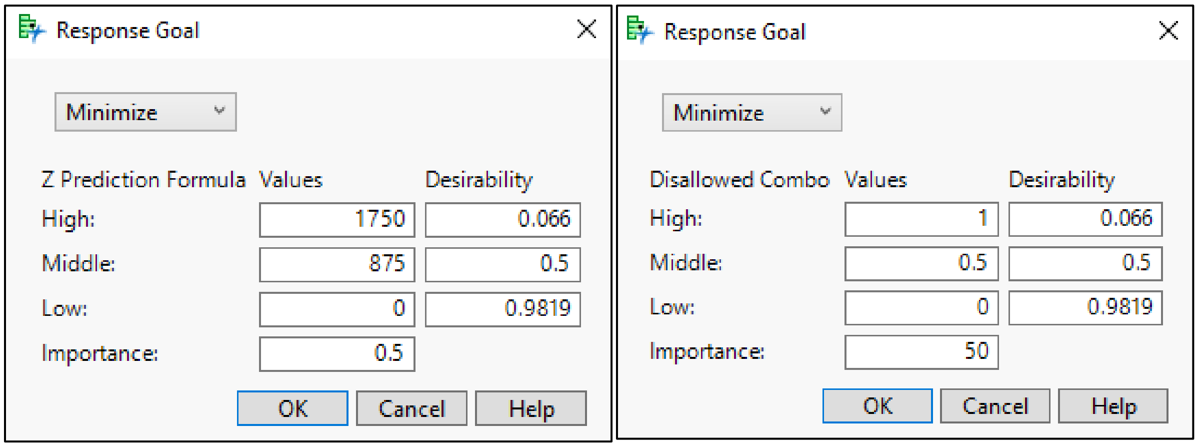


Figure 25 Prediction Profiler with Prediction Formula Response and Disallowed Combination Formula Response

As before we must turn on and define the Desirability functions.



Minimize

Z Prediction Formula	Values	Desirability
High:	1750	0.066
Middle:	875	0.5
Low:	0	0.9819
Importance:	0.5	

Minimize

Disallowed Combo	Values	Desirability
High:	1	0.066
Middle:	0.5	0.5
Low:	0	0.9819
Importance:	50	

Figure 26 Settings in 'Set Desirabilities' Dialogs

And finally, we can optimize the response by selecting ‘Maximize Desirability’ from the ‘Optimize and Desirability’ menu item. Figure 27 shows both the constrained optimal value and the unconstrained optimal value (steps not shown). Since the global optimum is outside the constrained factor space, the constrained optimum lies along the border of the cubic constraint. The unconstrained optimum does not coincide with the global optimum, but this is due mostly to extrapolation from an empirical fit, since there are no design points near the global optimum.

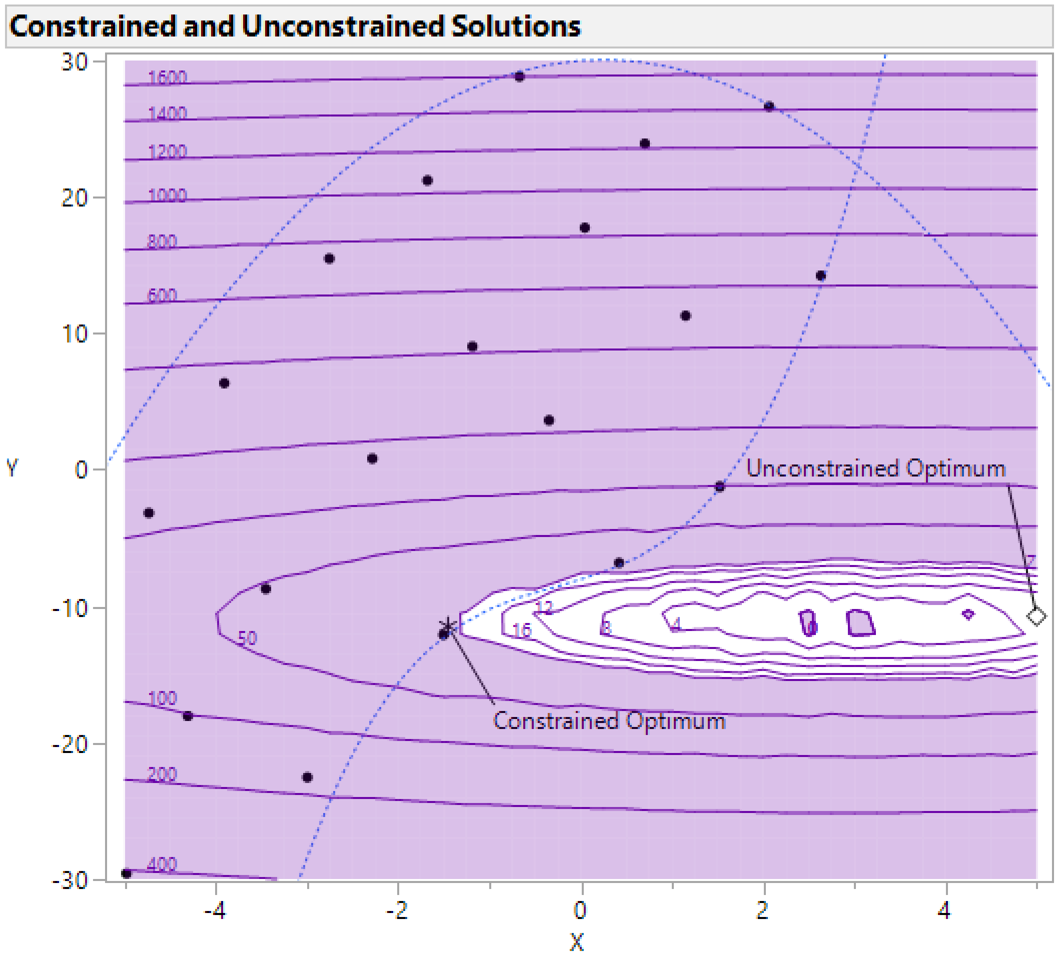


Figure 27 Constrained and Unconstrained Optimal Values

6. Conclusions

Space-Filling designs can be constructed which conform to non-linear constraints. By constructing a new response variable that represents our disallowed combinations, and then performing multiple response optimization we can find optimal values within our factor space that conform to non-linear constraints. We must set the importance of our disallowed combinations response to be high relative to our response variable of interest to ensure optimization obeys the constraint. An easy method to create disallowed combinations is to first specify the region we want, and then use the logical NOT operator to identify the regions we do not want. This analysis can be done in any package that performs multiple response optimization, but JMP Profilers make this easy.

7. Appendix

Code for Venn Diagrams:

```
Names Default To Here( 1 );

//-----------------------------------------------------------------------
//      REGION A Space Filling Design
//-----------------------------------------------------------------------

dtDOE = DOE(
      Space Filling Design,
      {Add Response( None, "Response", ., ., . ), Change Factor Settings( 1, 7.5,
29.5, "X1" ),
      Change Factor Settings( 2, 7.5, 29.5, "X2" ), Disallowed Combinations(
            deg = 0 :: 359 :: 1;
            deg = deg`;
            deg = deg * Pi() / 180;
            xCircle1 = 15 + 6 * Cos( deg );
            yCircle1 = 17 + 6 * Sin( deg );
            xCircle2 = 22 + 6 * Cos( deg );
            yCircle2 = 17 + 6 * Sin( deg );
            A = Eval Expr( In Polygon( X1, X2, xCircle1, yCircle1 ) );
            B = Eval Expr( In Polygon( X1, X2, xCircle2, yCircle2 ) );
            S = Eval Expr( In Polygon( X1, X2, [7.5, 7.5, 30, 30], [10, 24, 24, 10]
) );
            Eval( !A );
      ), FFF Optimality Criterion( MaxPro ), Space Filling Design Type( Fast
Flexible Filling, 100 ),
      Set Run Order( Randomize ), Make Table}
);
wait(0);
dtDOE << Close window;
Current Data Table() << New Script(
      "Points Selected",
      Graph Builder(
            Size( 573, 419 ),
            Show Control Panel( 0 ),
            Show Legend( 0 ),
            Variables( X( :X1 ), Y( :X2 ) ),
            Elements( Points( X, Y, Legend( 3 ) ) ),
            SendToReport(
                  Dispatch( {}, "Graph Builder", OutlineBox, {Set Title( "Venn
Diagram" )} ),
                  Dispatch(
                        {},
                        "X1",
                        ScaleBox,
                        {Format( "Best", 12 ), Min( 6.89531952900519 ), Max(
30.3742869283079 ), Inc( 5 ),
                        Minor Ticks( 1 )}
                ),
                Dispatch(
                        {},
                        "X2",
                        ScaleBox,
                        {Min( 9.09183673469388 ), Max( 24.9690325109636 ), Inc( 2
), Minor Ticks( 1 )}
                ),
                Dispatch( {}, "graph title", TextEditBox, {Set Text( "" )} ),
                Dispatch(
                        {},
                        "Graph Builder",
                        FrameBox,
                        {Add Graphics Script(
                                4,
                                Description( "Script" ),
                                circ1 = Eval Expr( Circle( {15, 17}, radius = 6 ) );
                                circ2 = Eval Expr( Circle( {22, 17}, radius = 6 ) );
                                rect1 = Eval Expr( Rect( 7.5, 24, 30, 10, 0 ) );
                                Eval( rect1 );
                                Eval( circ1 );
                                Eval( circ2 );
                        ), {Add Text Annotation(
                                Text( "S" ),
                                Text Box( {19, 28, 56, 79} ),
                                Filled( 0 ),
                                Text Color( "Medium Light Blue" ),
                                Font( "Tempus Sans ITC", 18, "Bold" )
                        ), Add Text Annotation(
                                Text( "A" ),
                                Text Box( {98, 60, 143, 111} ),
                                Filled( 0 ),
                                Text Color( "Medium Light Blue" ),
                                Font( "Tempus Sans ITC", 18, "Bold" )
                        ), Add Text Annotation(
                                Text( "B" ),
                                Text Box( {410, 74, 450, 125} ),
                                Filled( 0 ),
                                Text Color( "Medium Light Blue" ),
                                Font( "Tempus Sans ITC", 18, "Bold" )
                        )}}
                )
        )
    )
);


//----------------------------------------------------------------------------
//      REGION A or B Space Filling Design
//----------------------------------------------------------------------------
dtDOE = DOE(
    Space Filling Design,
    {Add Response( None, "Response", ., ., . ), Change Factor Settings( 1, 7.5,
29.5, "X1" ),
    Change Factor Settings( 2, 7.5, 29.5, "X2" ), Disallowed Combinations(
        deg = 0 :: 359 :: 1;
        deg = deg`;
        deg = deg * Pi() / 180;
        xCircle1 = 15 + 6 * Cos( deg );
        yCircle1 = 17 + 6 * Sin( deg );
        xCircle2 = 22 + 6 * Cos( deg );
        yCircle2 = 17 + 6 * Sin( deg );
        A = Eval Expr( In Polygon( X1, X2, xCircle1, yCircle1 ) );
        B = Eval Expr( In Polygon( X1, X2, xCircle2, yCircle2 ) );
        S = Eval Expr( In Polygon( X1, X2, [7.5, 7.5, 30, 30], [10, 24, 24, 10]
) );
        Eval( !(A | B ));
    ), FFF Optimality Criterion( MaxPro ), Space Filling Design Type( Fast
Flexible Filling, 100 ),
    Set Run Order( Randomize ), Make Table}
);
wait(0);
dtDOE << Close window;
Current Data Table() << New Script(
    "Points Selected",
    Graph Builder(
        Size( 573, 419 ),
        Show Control Panel( 0 ),
        Show Legend( 0 ),
        Variables( X( :X1 ), Y( :X2 ) ),
        Elements( Points( X, Y, Legend( 3 ) ) ),
        SendToReport(
                Dispatch( {}, "Graph Builder", OutlineBox, {Set Title( "Venn
Diagram" )} ),
                Dispatch(
                        {},
                        "X1",
                        ScaleBox,
                        {Format( "Best", 12 ), Min( 6.89531952900519 ), Max(
30.3742869283079 ), Inc( 5 ),
                        Minor Ticks( 1 )}
                ),
                Dispatch(
                        {},
                        "X2",
                        ScaleBox,
                        {Min( 9.09183673469388 ), Max( 24.9690325109636 ), Inc( 2
), Minor Ticks( 1 )}
                ),
                Dispatch( {}, "graph title", TextEditBox, {Set Text( "" )} ),
                Dispatch(
                        {},
                        "Graph Builder",
                        FrameBox,
                        {Add Graphics Script(
                                4,
                                Description( "Script" ),
                                circ1 = Eval Expr( Circle( {15, 17}, radius = 6 ) );
                                circ2 = Eval Expr( Circle( {22, 17}, radius = 6 ) );
```

```
                                rect1 = Eval Expr( Rect( 7.5, 24, 30, 10, 0 ) );
                                Eval( rect1 );
                                Eval( circ1 );
                                Eval( circ2 );
                          ), {Add Text Annotation(
                                Text( "S" ),
                                Text Box( {19, 28, 56, 79} ),
                                Filled( 0 ),
                                Text Color( "Medium Light Blue" ),
                                Font( "Tempus Sans ITC", 18, "Bold" )
                          ), Add Text Annotation(
                                Text( "A" ),
                                Text Box( {98, 60, 143, 111} ),
                                Filled( 0 ),
                                Text Color( "Medium Light Blue" ),
                                Font( "Tempus Sans ITC", 18, "Bold" )
                          ), Add Text Annotation(
                                Text( "B" ),
                                Text Box( {410, 74, 450, 125} ),
                                Filled( 0 ),
                                Text Color( "Medium Light Blue" ),
                                Font( "Tempus Sans ITC", 18, "Bold" )
                          )}}
                    )
              )
      )
);


//----------------------------------------------------------------------------
//      REGION A & B Space Filling Design
//----------------------------------------------------------------------------

dtDOE = DOE(
      Space Filling Design,
      {Add Response( None, "Response", ., ., . ), Change Factor Settings( 1, 8,
29.5, "X1" ),
      Change Factor Settings( 2, 8, 29.5, "X2" ), Disallowed Combinations(
            deg = 0 :: 359 :: 1;
            deg = deg`;
            deg = deg * Pi() / 180;
            xCircle1 = 15 + 6 * Cos( deg );
            yCircle1 = 17 + 6 * Sin( deg );
            xCircle2 = 22 + 6 * Cos( deg );
            yCircle2 = 17 + 6 * Sin( deg );
            A = Eval Expr( In Polygon( X1, X2, xCircle1, yCircle1 ) );
            B = Eval Expr( In Polygon( X1, X2, xCircle2, yCircle2 ) );
            S = Eval Expr( In Polygon( X1, X2, [7.5, 7.5, 30, 30], [10, 24, 24, 10]
) );
            Eval( !(A & B) );
      ), FFF Optimality Criterion( MaxPro ), Space Filling Design Type( Fast
Flexible Filling, 100 ),
      Set Run Order( Randomize ), Make Table}
);
wait(0);
```

```
dtDOE << Close window;

Current Data Table() << New Script(
	"Points Selected",
	Graph Builder(
		Size( 573, 419 ),
		Show Control Panel( 0 ),
		Show Legend( 0 ),
		Variables( X( :X1 ), Y( :X2 ) ),
		Elements( Points( X, Y, Legend( 3 ) ) ),
		SendToReport(
			Dispatch( {}, "Graph Builder", OutlineBox, {Set Title( "Venn
Diagram" )} ),
			Dispatch(
				{},
				"X1",
				ScaleBox,
				{Format( "Best", 12 ), Min( 6.89531952900519 ), Max(
30.3742869283079 ), Inc( 5 ),
				Minor Ticks( 1 )}
			),
			Dispatch(
				{},
				"X2",
				ScaleBox,
				{Min( 9.09183673469388 ), Max( 24.9690325109636 ), Inc( 2
), Minor Ticks( 1 )}
			),
			Dispatch( {}, "graph title", TextEditBox, {Set Text( "" )} ),
			Dispatch(
				{},
				"Graph Builder",
				FrameBox,
				{Add Graphics Script(
					4,
					Description( "Script" ),
					circ1 = Eval Expr( Circle( {15, 17}, radius = 6 ) );
					circ2 = Eval Expr( Circle( {22, 17}, radius = 6 ) );
					rect1 = Eval Expr( Rect( 7.5, 24, 30, 10, 0 ) );
					Eval( rect1 );
					Eval( circ1 );
					Eval( circ2 );
				), {Add Text Annotation(
					Text( "S" ),
					Text Box( {19, 28, 56, 79} ),
					Filled( 0 ),
					Text Color( "Medium Light Blue" ),
					Font( "Tempus Sans ITC", 18, "Bold" )
				), Add Text Annotation(
					Text( "A" ),
					Text Box( {98, 60, 143, 111} ),
					Filled( 0 ),
					Text Color( "Medium Light Blue" ),
					Font( "Tempus Sans ITC", 18, "Bold" )
				), Add Text Annotation(
                                Text( "B" ),
                                Text Box( {410, 74, 450, 125} ),
                                Filled( 0 ),
                                Text Color( "Medium Light Blue" ),
                                Font( "Tempus Sans ITC", 18, "Bold" )
                        )}}
                    )
                )
            )
);


//----------------------------------------------------------------------------
//      REGION B and not A Space Filling Design
//----------------------------------------------------------------------------

dtDOE = DOE(
        Space Filling Design,
        {Add Response( None, "Response", ., ., . ), Change Factor Settings( 1, 8,
29.5, "X1" ),
        Change Factor Settings( 2, 8, 29.5, "X2" ), Disallowed Combinations(
                deg = 0 :: 359 :: 1;
                deg = deg`;
                deg = deg * Pi() / 180;
                xCircle1 = 15 + 6 * Cos( deg );
                yCircle1 = 17 + 6 * Sin( deg );
                xCircle2 = 22 + 6 * Cos( deg );
                yCircle2 = 17 + 6 * Sin( deg );
                A = Eval Expr( In Polygon( X1, X2, xCircle1, yCircle1 ) );
                B = Eval Expr( In Polygon( X1, X2, xCircle2, yCircle2 ) );
                S = Eval Expr( In Polygon( X1, X2, [7.5, 7.5, 30, 30], [10, 24, 24, 10]
) );
                Eval( !(B & !A ) );
        ), FFF Optimality Criterion( MaxPro ), Space Filling Design Type( Fast
Flexible Filling, 100 ),
        Set Run Order( Randomize ), Make Table}
);
wait(0);
dtDOE << Close window;

Current Data Table() << New Script(
        "Points Selected",
        Graph Builder(
                Size( 573, 419 ),
                Show Control Panel( 0 ),
                Show Legend( 0 ),
                Variables( X( :X1 ), Y( :X2 ) ),
                Elements( Points( X, Y, Legend( 3 ) ) ),
                SendToReport(
                        Dispatch( {}, "Graph Builder", OutlineBox, {Set Title( "Venn
Diagram" )} ),
                        Dispatch(
                                {},
                                "X1",
                                ScaleBox,
                                {Format( "Best", 12 ), Min( 6.89531952900519 ), Max(
30.3742869283079 ), Inc( 5 ),
                                Minor Ticks( 1 )}
                        ),
                        Dispatch(
                                {},
                                "X2",
                                ScaleBox,
                                {Min( 9.09183673469388 ), Max( 24.9690325109636 ), Inc( 2
), Minor Ticks( 1 )}
                        ),
                        Dispatch( {}, "graph title", TextEditBox, {Set Text( "" )} ),
                        Dispatch(
                                {},
                                "Graph Builder",
                                FrameBox,
                                {Add Graphics Script(
                                        4,
                                        Description( "Script" ),
                                        circ1 = Eval Expr( Circle( {15, 17}, radius = 6 ) );
                                        circ2 = Eval Expr( Circle( {22, 17}, radius = 6 ) );
                                        rect1 = Eval Expr( Rect( 7.5, 24, 30, 10, 0 ) );
                                        Eval( rect1 );
                                        Eval( circ1 );
                                        Eval( circ2 );
                                ), {Add Text Annotation(
                                        Text( "S" ),
                                        Text Box( {19, 28, 56, 79} ),
                                        Filled( 0 ),
                                        Text Color( "Medium Light Blue" ),
                                        Font( "Tempus Sans ITC", 18, "Bold" )
                                ), Add Text Annotation(
                                        Text( "A" ),
                                        Text Box( {98, 60, 143, 111} ),
                                        Filled( 0 ),
                                        Text Color( "Medium Light Blue" ),
                                        Font( "Tempus Sans ITC", 18, "Bold" )
                                ), Add Text Annotation(
                                        Text( "B" ),
                                        Text Box( {410, 74, 450, 125} ),
                                        Filled( 0 ),
                                        Text Color( "Medium Light Blue" ),
                                        Font( "Tempus Sans ITC", 18, "Bold" )
                                )}}
                        )
                )
        )
);

//-----------------------------------------------------------------------------
//      REGION that is not A or B but is still in S
//-----------------------------------------------------------------------------

dtDOE = DOE(
        Space Filling Design,
	{Add Response( None, "Response", ., ., . ), Change Factor Settings( 1, 8, 
29.5, "X1" ),
	Change Factor Settings( 2, 8, 29.5, "X2" ), Disallowed Combinations(
		deg = 0 :: 359 :: 1;
		deg = deg`;
		deg = deg * Pi() / 180;
		xCircle1 = 15 + 6 * Cos( deg );
		yCircle1 = 17 + 6 * Sin( deg );
		xCircle2 = 22 + 6 * Cos( deg );
		yCircle2 = 17 + 6 * Sin( deg );		A = Eval Expr( In Polygon( X1, 
X2, xCircle1, yCircle1 ) );
		B = Eval Expr( In Polygon( X1, X2, xCircle2, yCircle2 ) );
		S = Eval Expr( In Polygon( X1, X2, [7.5, 7.5, 30, 30], [10, 24, 24, 10] 
) );
		Eval( !((!A & !B ) & S) );
	), FFF Optimality Criterion( MaxPro ), Space Filling Design Type( Fast 
Flexible Filling, 100 ),
	Set Run Order( Randomize ), Make Table}
);
wait(0);
dtDOE << Close window;

Current Data Table() << New Script(
	"Points Selected",
	Graph Builder(
		Size( 573, 419 ),
		Show Control Panel( 0 ),
		Show Legend( 0 ),
		Variables( X( :X1 ), Y( :X2 ) ),
		Elements( Points( X, Y, Legend( 3 ) ) ),
		SendToReport(
			Dispatch( {}, "Graph Builder", OutlineBox, {Set Title( "Venn 
Diagram" )} ),
			Dispatch(
				{},
				"X1",
				ScaleBox,
				{Format( "Best", 12 ), Min( 6.89531952900519 ), Max( 
30.3742869283079 ), Inc( 5 ),
				Minor Ticks( 1 )}
			),
			Dispatch(
				{},
				"X2",
				ScaleBox,
				{Min( 9.09183673469388 ), Max( 24.9690325109636 ), Inc( 2 
), Minor Ticks( 1 )}
			),
			Dispatch( {}, "graph title", TextEditBox, {Set Text( "" )} ),
			Dispatch(
				{},
				"Graph Builder",
				FrameBox,
				{Add Graphics Script(
					4,
                                Description( "Script" ),
                                circ1 = Eval Expr( Circle( {15, 17}, radius = 6 ) );
                                circ2 = Eval Expr( Circle( {22, 17}, radius = 6 ) );
                                rect1 = Eval Expr( Rect( 7.5, 24, 30, 10, 0 ) );
                                Eval( rect1 );
                                Eval( circ1 );
                                Eval( circ2 );
                        ), {Add Text Annotation(
                                Text( "S" ),
                                Text Box( {19, 28, 56, 79} ),
                                Filled( 0 ),
                                Text Color( "Medium Light Blue" ),
                                Font( "Tempus Sans ITC", 18, "Bold" )
                        ), Add Text Annotation(
                                Text( "A" ),
                                Text Box( {98, 60, 143, 111} ),
                                Filled( 0 ),
                                Text Color( "Medium Light Blue" ),
                                Font( "Tempus Sans ITC", 18, "Bold" )
                        ), Add Text Annotation(
                                Text( "B" ),
                                Text Box( {410, 74, 450, 125} ),
                                Filled( 0 ),
                                Text Color( "Medium Light Blue" ),
                                Font( "Tempus Sans ITC", 18, "Bold" )
                        )}}
                )
        )
    )
);
```