The **desirability** Package

Max Kuhn
max.kuhn@pfizer.com

January 16, 2014

1 Introduction

The **desirability** package contains S3 classes for multivariate optimization using the desirability function approach of Harrington (1965) using functional forms described by Derringer and Suich (1980).

2 Basic Desirability Functions

The desirability function approach to simultaneously optimizing multiple equations was originally proposed by Harrington (1980). Essentially, the approach is to translate the functions to a common scale ([0, 1]), combine them using the geometric mean and optimize the overall metric. The equations may represent model predictions or other equations.

For example, desirability functions are popular in response surface methodology (Box and Wilson (1951), Myers and Montgomery (1995)) as a method to simultaneously optimize a series of quadratic models. A response surface experiment may use measurements on a set of outcomes. Instead of optimizing each outcome separately, settings for the predictor variables sought to satisfy all of the outcomes at once.

Also, in drug discovery, predictive models can be constructed to relate the molecular structures of compounds to characteristics of interest (such as absorption properties, potency and selectivity for the intended target). Given a set of predictive models built using existing compounds, predictions can be made on a large set of virtual compounds that have been designed but not necessarily synthesized. Using the model predictions, a virtual compound can be scored on how well the model results agree with required properties. In this case, ranking compounds on multiple endpoints may be sufficient to meet the scientist's needs.

Originally, Harrington used exponential functions to quantify desirability. In this package, the

simple discontinuous functions of Derringer and Suich (1984) are used. Suppose that there are R equations or function to simultaneously optimize, denoted $f_r(\mathbf{x})$ $(r = 1 \ldots R)$. For each of the R functions, an individual "desirability" function is constructed that is high when $f_r(\mathbf{x})$ is at the desirable level (such as a maximum, minimum, or target) and low when $f_r(\mathbf{x})$ is at an undesirable value. Derringer and Suich proposed three forms of these functions, corresponding to the type of optimization goal. For maximization of $f_r(\mathbf{x})$, the function

$$d_r^{max} = \begin{cases} 0 & \text{if } f_r(\mathbf{x}) < A \\ \left(\frac{f_r(\mathbf{X})-A}{B-A}\right)^s & \text{if } A \leq f_r(\mathbf{x}) \leq B \\ 1 & \text{if } f_r(\mathbf{x}) > B \end{cases} \tag{1}$$

can be used, where A, B, and s are chosen by the investigator. When the equation is to be minimized, they proposed the function

$$d_r^{min} = \begin{cases} 0 & \text{if } f_r(\mathbf{x}) > B \\ \left(\frac{f_r(\mathbf{X})-B}{A-B}\right)^s & \text{if } A \leq f_r(\mathbf{x}) \leq B \\ 1 & \text{if } f_r(\mathbf{x}) < A \end{cases}, \tag{2}$$

and for target is best situations,

$$d_r^{target} = \begin{cases} \left(\frac{f_r(\mathbf{X})-A}{t_0-A}\right)^{s_1} & \text{if } A \leq f_r(\mathbf{x}) \leq t_0 \\ \left(\frac{f_r(\mathbf{X})-B}{t_0-B}\right)^{s_2} & \text{if } t_0 \leq f_r(\mathbf{x}) \leq B \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

These functions are on the same scale and are discontinuous at the points A, B, and t_0. The values of s, s_1 or s_2 can be chosen so that the desirability criterion is easier or more difficult to satisfy. For example, if s is chosen to be less than 1 in (2), d_r^{Min} is near 1 even if the model $f_r(\mathbf{x})$ is not low. As values of s move closer to 0, the desirability reflected by (2) becomes higher. Likewise, values of s greater than 1 will make d_r^{Min} harder to satisfy in terms of desirability. These scaling factors are useful when one equation is of greater importance than the others. Examples of these functions are given in Figure 1. It should be noted that any function can be used to mirror the desirability of a model. For example, Del Castillo, Montgomery, and McCarville (1996) develop alternative desirability functions that can be used in conjunction with gradient based optimization routines.

For each of these three desirability functions (and the others discussed in Section 6.3), there are `print`, `plot` and `predict` methods.

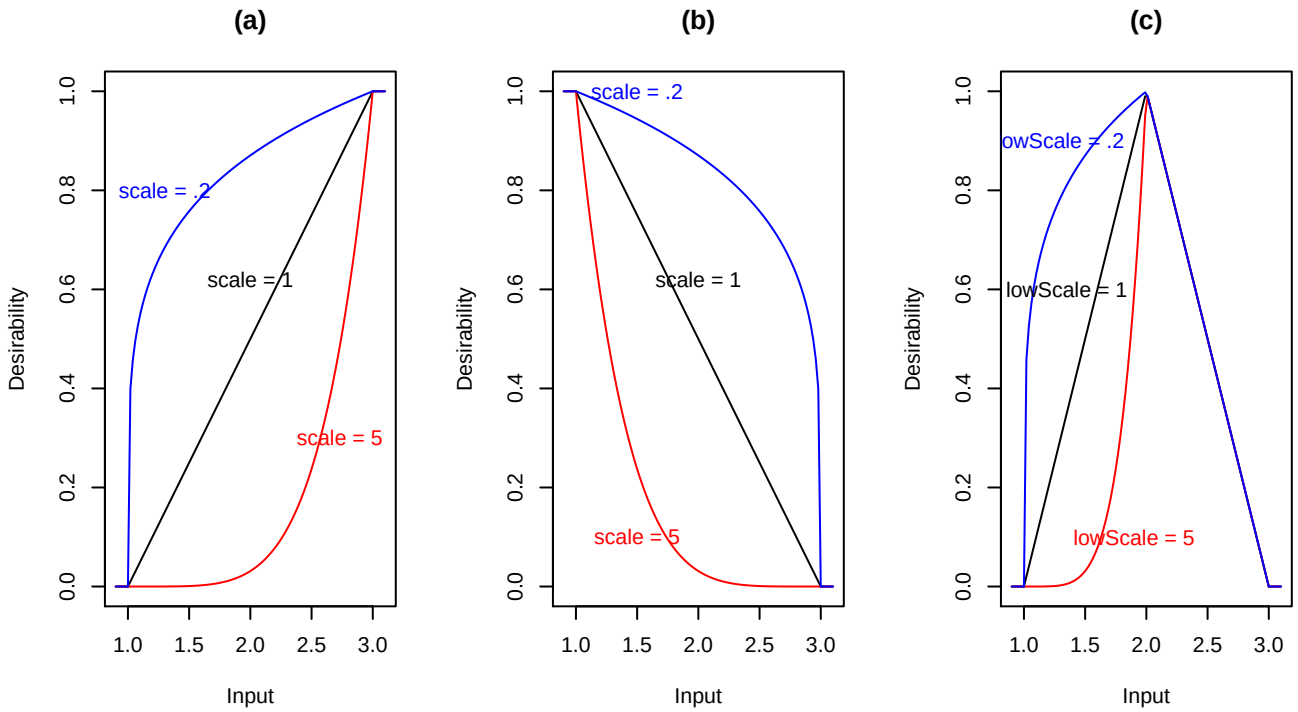

Figure 1: Examples of the three primary desirability functions. Panel (a) shows an example of a larger–is–better function, panel (b) shows a smaller–is–better desirability function and panel (c) shows a function where the optimal value corresponds to a target value. Not that increasing the scale parameter makes it more difficult to achieve higher desirability, while values smaller than 1 make it easier to achieve good results.

3 Overall Desirability

Given that the R desirability functions $d_1 \ldots d_r$ are on the [0,1] scale, they can be combined to achieve an overall desirability function, D. One method of doing this is by the geometric mean

$$D = \left(\prod_{r=1}^{R} d_r \right)^{1/R}.$$

The geometric mean has the property that if any one model is undesirable ($d_r = 0$), the overall desirability is also unacceptable ($D = 0$).

Once D has been defined and the prediction equations for each of the R equations have been computed, it can be use to optimize or rank the predictors.

4 An Example

Myers and Montgomery (1995) describe a response surface experiment where three factors (reaction time, reaction temperature and percent catalyst) were used to model two characteristics of the chemical reaction: percent conversion and thermal activity. They present two equations[1] for the fitted quadratic response surface models:

```
> conversionPred <- function(x) 81.09 + 1.0284 * x[1] + 4.043 * x[2] + 6.2037 * x[3] -
+    1.8366 * x[1]^2 + 2.9382 * x[2]^2 - 5.1915 * x[3]^2 +
+    2.2150 * x[1] * x[2] + 11.375 * x[1] * x[3] - 3.875 * x[2] * x[3]
> activityPred <- function(x) 59.85 + 3.583 * x[1] + 0.2546 * x[2] + 2.2298 * x[3] +
+    0.83479 * x[1]^2 + 0.07484 * x[2]^2 + 0.05716 * x[3]^2 -
+    0.3875 * x[1] * x[2] - 0.375 * x[1] * x[3] + 0.3125 * x[2] * x[3]
```

The goal of the analysis was to maximize conversion while keeping the thermal activity between 55 and 60 units. An activity target of 57.5 was used in the analysis. Plots of the response surface models are in Figures 2 and 3, where reaction time and percent catalyst are plotted while the reaction temperature was varied at four different levels. Both quadratic models are saddle surfaces and the stationary points are outside of the experimental region. To determine predictor settings these models, a constrained optimization can be used to stay inside of the experimental region.

Translating the experimental goals to desirability functions, a larger–is–better function (1) is used for percent conversion with values $A = 80$ and $B = 97$. A target oriented desirability function (3) was used for thermal activity with $t_0 = 57.5$, $A = 55$ and $B = 60$. Although the original

[1] In practice, we would just use the predict method for the linear model objects to get the prediction equation. Our results are slightly different from those given by Myers and Montgomery because they used prediction equations with full floating–point precision.

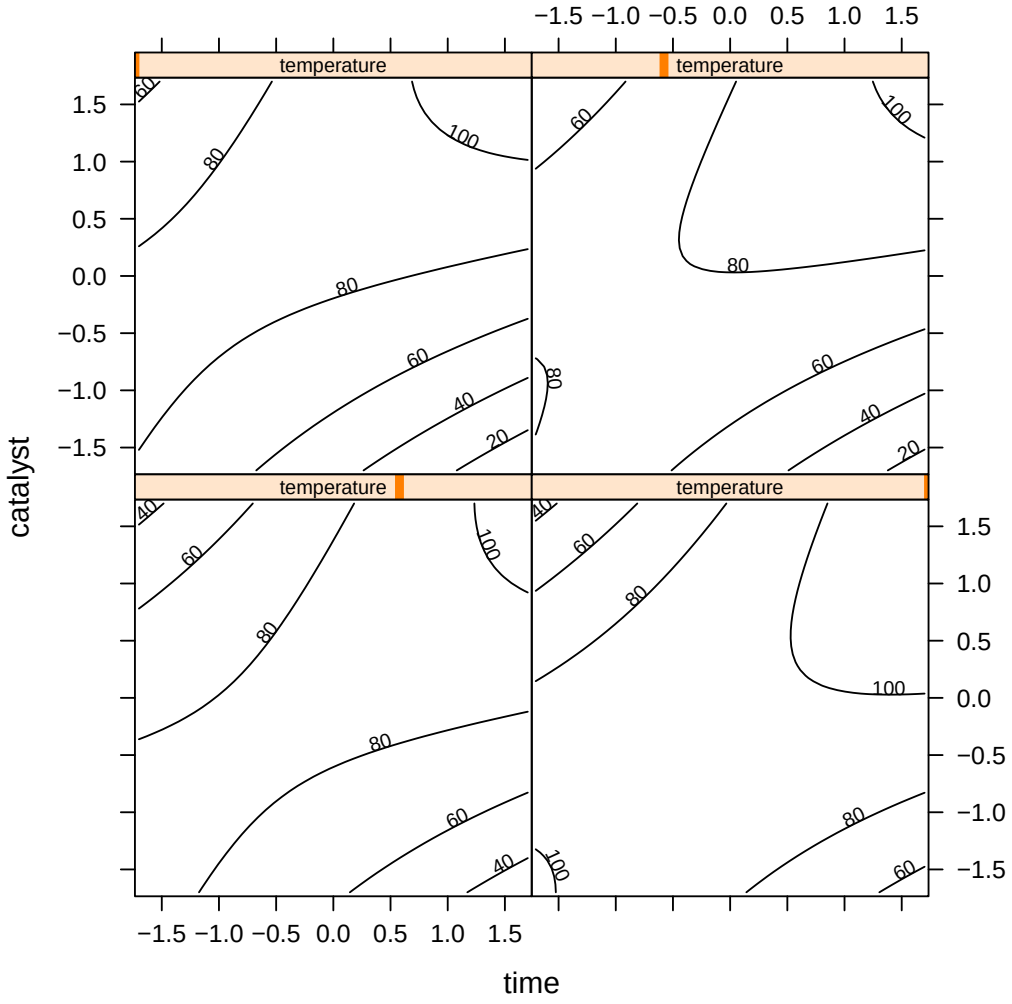

Figure 2: The response surface for the percent conversion model. To plot the model contours, the temperature variable was fixed at four diverse levels. The largest effects in the fitted model are due to the time×catalyst interaction and the linear and quadratic effects of catalyst.

analysis used numerous combinations of scaling parameters, we will only show analyses with the default scaling factor values. Figures 4, 4 and 6 show contour plots of the individual desirability function surfaces and the overall surface.

To construct the overall desirability functions, objects must be created for the individual functions. For example, the following code chunk creates the appropriate objects and uses the `predict` method to estimate desirability at the center point of the design:

```
> conversionD <- dMax(80, 97)
> activityD <- dTarget(55, 57.5, 60)
> predOutcomes <- c(conversionPred(c(0,0,0)), activityPred(c(0,0,0)))
> print(predOutcomes)

[1] 81.09 59.85

> predict(conversionD, predOutcomes[1])

[1] 0.06411765

> predict(activityD, predOutcomes[2])

[1] 0.06
```

To get the overall score for these settings of the experimental factors, the `dOverall` function is used to combine the objects and `predict` is used to get the final score:

```
> overallD <- dOverall(conversionD, activityD)
> print(overallD)

Combined desirability function

Call: dOverall.default(conversionD, activityD)

----
Larger-is-better desirability function

Call: dMax.default(low = 80, high = 97)

Non-informative value: 0.5
----
Target-is-best desirability function

Call: dTarget.default(low = 55, target = 57.5, high = 60)

Non-informative value: 0.4949

> predict(overallD, predOutcomes)

[1] 0 0
```

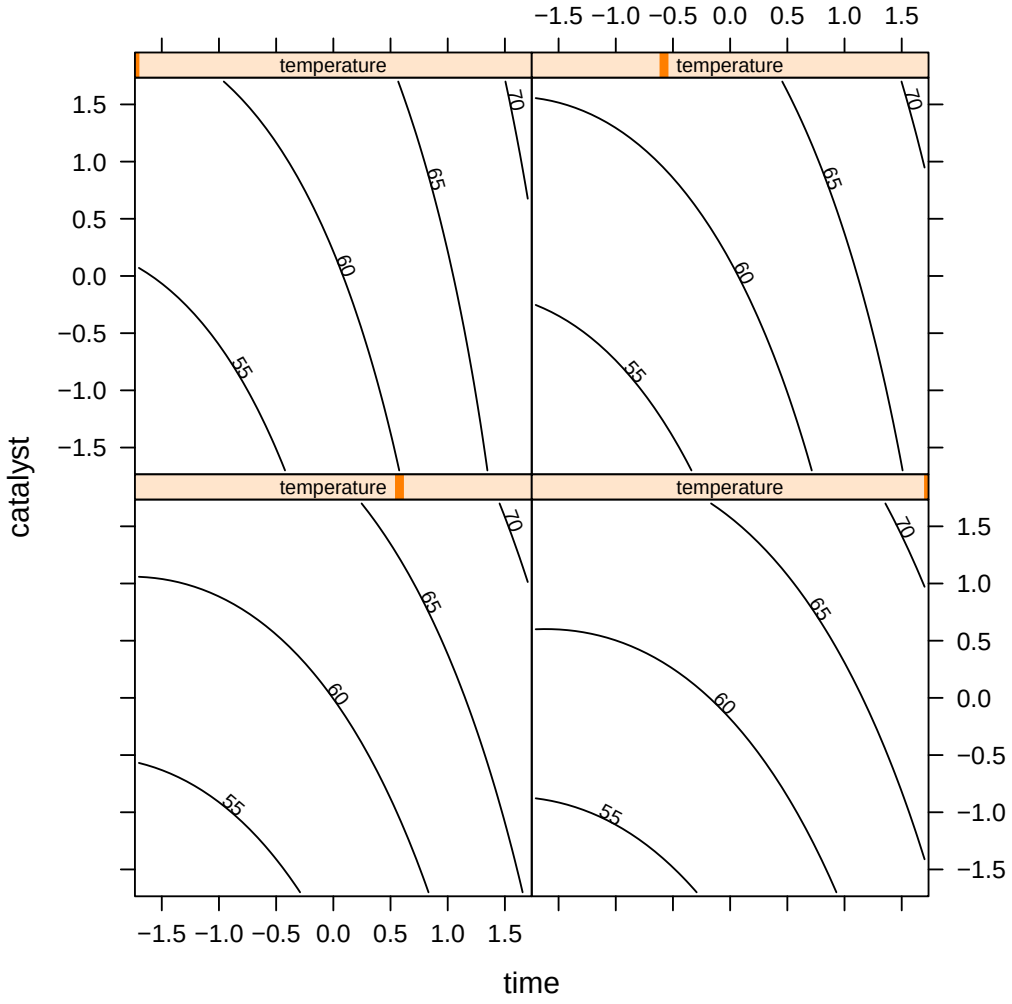

Figure 3: The response surface for the thermal activity model. To plot the model contours, the temperature variable was fixed at four diverse levels. The main effects of time and catalyst have the largest effect on the fitted model.

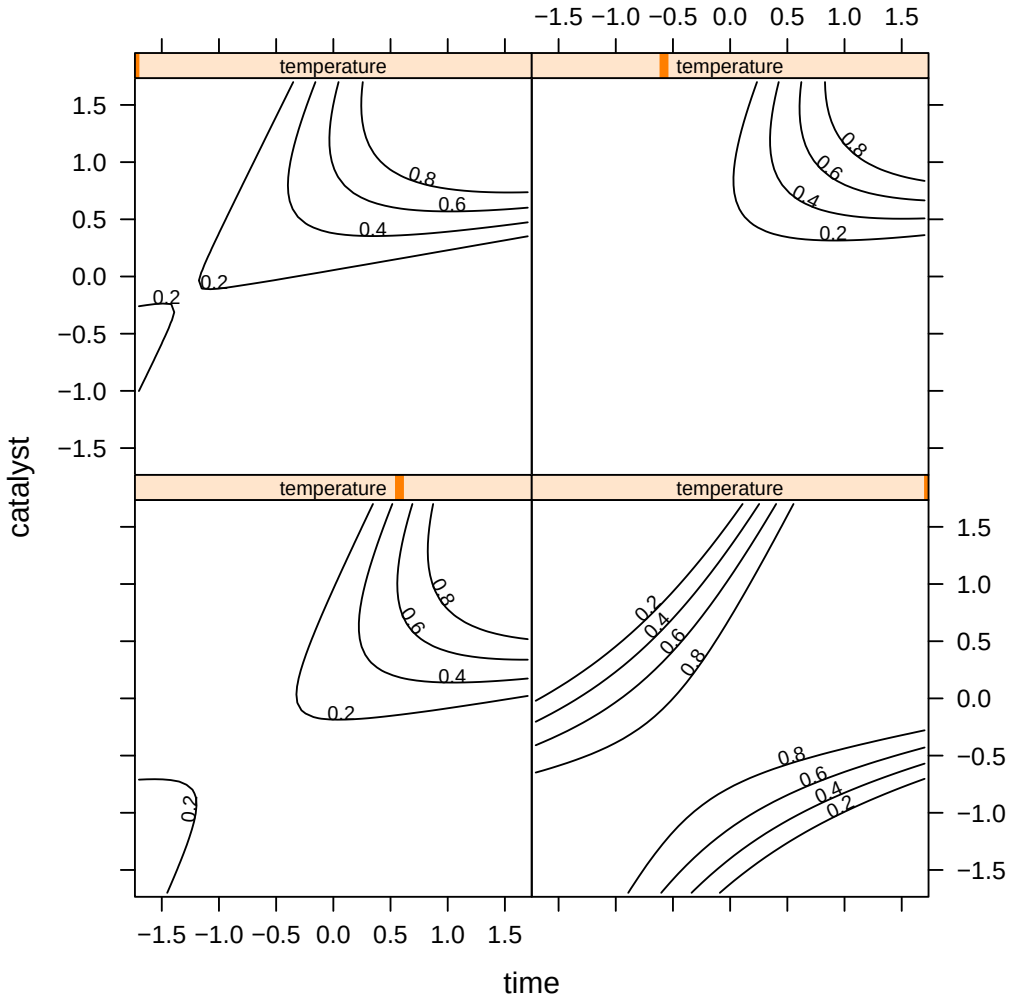

Figure 4: The individual desirability surface for the percent conversion outcome using `dMax(80, 97)`

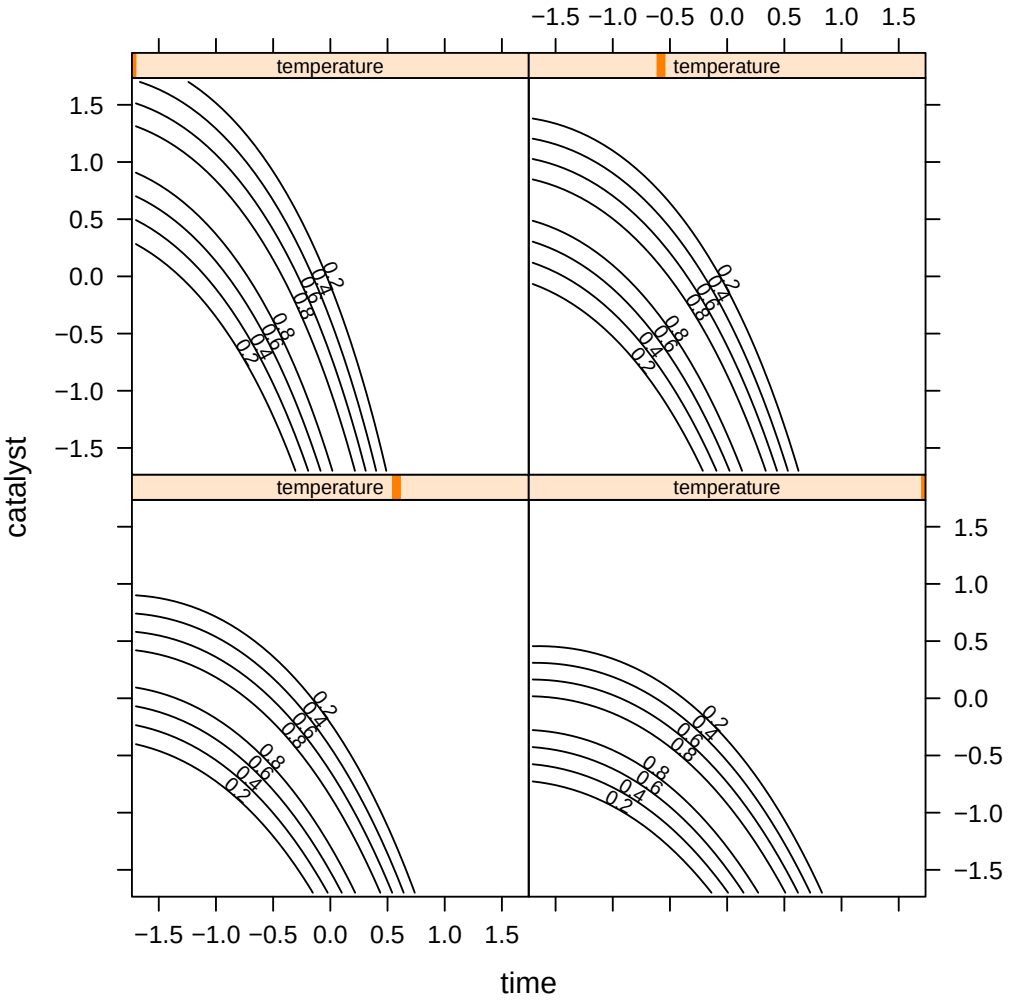

Figure 5: The individual desirability surface for the thermal activity outcome using `dTarget(55, 57.5, 60)`

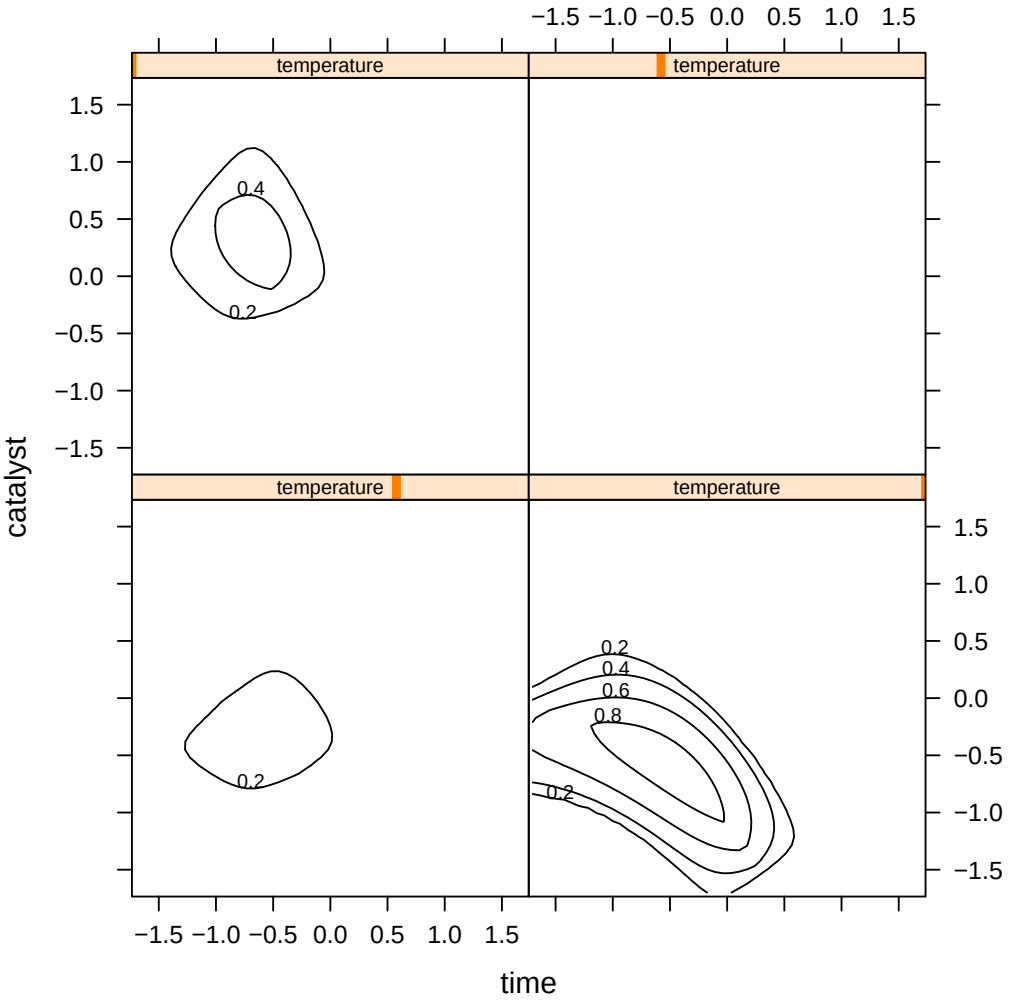

Figure 6: The overall desirability surface that combines the model for percent conversion and thermal activity

5 Maximizing Desirability

Following Myers and Montgomery, we can maximize desirability within a cuboidal region bounded by the value of the axial points. To do this, the objective function (`rsmOpt`) uses a penalty approach; if a candidate point falls outside of the cuboidal design region, the desirability is set to zero.

```
> rsmOpt <- function(x, dObject, space = "square")
+ {
+   conv <- conversionPred(x)
+   acty <- activityPred(x)
+
+   out <- predict(dObject, data.frame(conv = conv, acty = acty))
+
+   if(space == "circular")
+     {
+       if(sqrt(sum(x^2)) > 1.682) out <- 0
+     } else if(space == "square") if(any(abs(x) > 1.682)) out <- 0
+   out
+ }
```

The Nelder–Mean simplex method (Nelder and Mead (1965), Olsson and Nelson (1975)) can be used to maximize desirability using the `optim` function. This is a direct search method that uses only function calls and does not use gradient information. This optimization technique can handle discontinuous or non–smooth functions that are commonly produced by desirability functions. However, this method has the propensity to converge to a local optimum. Since the technique is fast when we have an efficient function to optimize, we can restart the algorithm in multiple locations in the feasible space and use the best outcome. As an alternative, other direct search methods are more likely to yield a global optimum, such as simulated annealing (Bohachevsky *et al*, 1986). This particular routine is also available in `optim`, but tuning the annealing parameters may be needed to ensure that the technique is performing well.

The following code chunk uses the Nelder–Mean simplex method and a cubiodal design region:

```
> searchGrid <- expand.grid(time = seq(-1.5, 1.5, length = 5),
+                           temperature = seq(-1.5, 1.5, length = 5),
+                           catalyst = seq(-1.5, 1.5, length = 5))
> for(i in 1:dim(searchGrid)[1])
+ {
+   tmp <- optim(as.vector(searchGrid[i,]),
+                rsmOpt,
+                dObject = overallD,
+                space = "square",
+                control = list(fnscale = -1))
+   if(i == 1)
+     {
+       best <- tmp
```

```
+     } else {
+       if(tmp$value > best$value) best <- tmp
+     }
+ }
> print(best)

$par
       time temperature    catalyst
 -0.5117132   1.6820000  -0.5863863

$value
[1] 0.9425094

$counts
function gradient
     226       NA

$convergence
[1] 0

$message
NULL
```

From this optimization, the predicted value of conversion was 95.1 and activity was predicted to be 57.5.

Alternatively we can try to maximize desirability such that the experimental factors are constrained to be within a spherical design region with a radius equal to the axial point distance:

```
> for(i in 1:dim(searchGrid)[1])
+ {
+   tmp <- optim(as.vector(searchGrid[i,]),
+                rsmOpt,
+                space = "circular",
+                dObject = overallD,
+                control = list(fnscale = -1))
+   if(i == 1)
+     {
+       best <- tmp
+     } else {
+       if(tmp$value > best$value) best <- tmp
+     }
+ }
> print(best)

$par
       time temperature    catalyst
 -0.5094851   1.5034146  -0.5561415
```

```
$value
[1] 0.8581525

$counts
function gradient
     308       NA

$convergence
[1] 0

$message
NULL
```

The process converges to relative sub–optimum values (conversion = 92.52 and activity = 57.5). Using a radius of 2 produces overall desirability equal to one, although the solution extrapolates slightly outside of the design region.

6 Non–Standard Features

The preceding approach has been faithful to the process described in Derringer and Suich (1980) and is consistent with current implementations of desirability functions. This package also contains a few non–standard features.

6.1 Non–Informative Desirability and Missing Values

In some cases, the inputs to the desirability functions cannot be computed. When individual desirability functions are defined, a non–informative value is estimated by computing the desirabilities over the possible range and taking the mean value. By default, if the input to this desirability function is `NA`, it is replaced by the non–informative value. In order to have the calculation return an `NA` value, the value of `object$missing` can be changed to `NA`, where `object` is the result of a call to one of the desirability R functions. The non–informative value is plotted as a broken line in the default `plot` methods (see Figure 7 for an example).

6.2 Zero–Desirability Tolerances

In some cases where the dimensionality of the outcomes is large, it may be difficult to find feasible solutions where every individual desirability value is acceptable. Each desirability R function has a `tol` argument that can be set to a number on [0, 1] (the default value is `NULL`). If this value is not null, desirability values equal to zero are replaced by the value of `tol`. This computation is applied after the missing value imputation step.

6.3 Non–Standard Desirability Functions

In some cases, the three R desirability functions previously discussed may not cover all of the possible forms of the desirability function required by the user. The function `dArb` takes as input a numeric vector of input values and their matching desirabilities and can approximate many other functional forms.

For example, if a symmetric, sinusoidal curve was needed to translate a real–valued equation to the desirability scale, the logistic function could be used:

$$d(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x})}$$

Outside of the range ±5, the desirability values are close to zero and one, so we can define 20 points on this range, compute the logistic model and use these values to define the desirability function:

```
> foo <- function(u) 1/(1+exp(-u))
> xInput <- seq(-5,5, length = 20)
> logisticD <- dArb(xInput, foo(xInput))
```

Inputs in–between the grid points are linearly interpolated.

Using this approach, the extreme values are used outside of the input range. For example, an input value of –10 would be assigned to desirability value of `foo(-5)`. Similarly, on the high side, the last desirability value (when the inputs are ordered) is carried forward. Figure 7 shows an example of the `plot` method applied to the object `logisticD`.

Similarly, there is another desirability function to implement box constraints on an equation. For example, instead of using a penalty approach to constraining the experimental factors to be within the design region, we could coerce values outside of ±1.682 to have zero desirability. Figure 8 shows an example function. In our previous optimization example, we could create a desirability function for each of the predictors and add them to the overall desirability object.

Another non–standard application of desirability functions uses categorical inputs. Desirabilities can be assigned for all possible values. For example:

```
> values <- c("value1" = .1, "value2" = .9, "value3" = .2)
> groupedDesirabilities <- dCategorical(values)
> groupedDesirabilities

Desirability function for categorical data

Call: dCategorical.default(values = values)

Non-informative value: 0.4
```

Figure 9 shows a plot of the desirability profiles for this configuration.

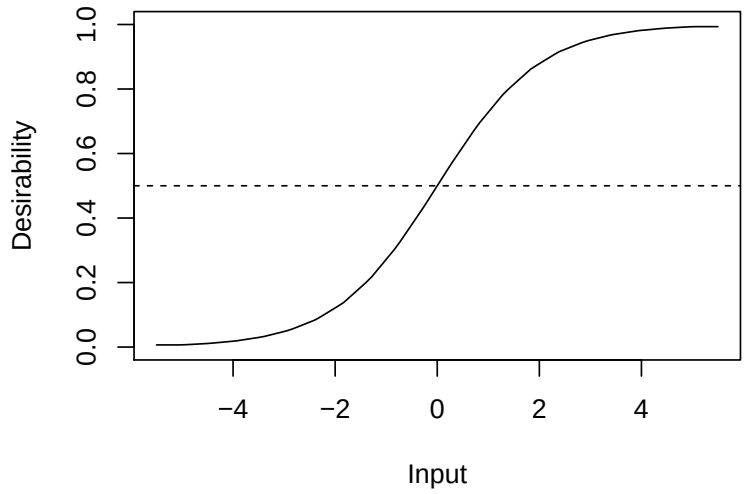

Figure 7: An example of using a logistic function to translate inputs to desirability using the `dArb` function.

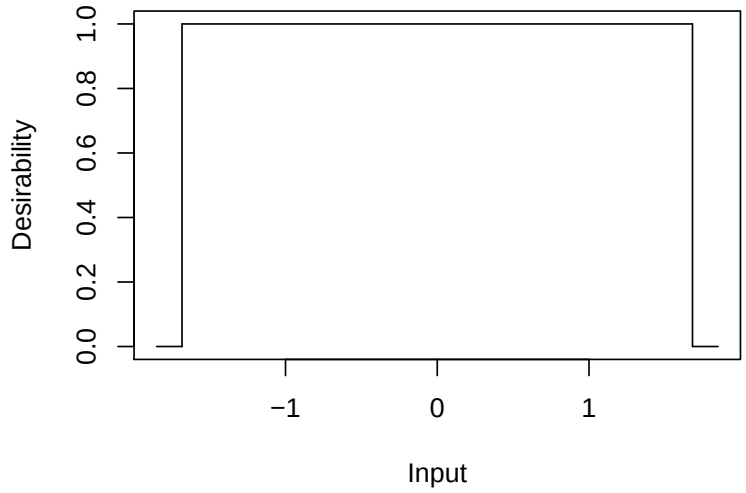

Figure 8: An example of using a box–like desirability function.

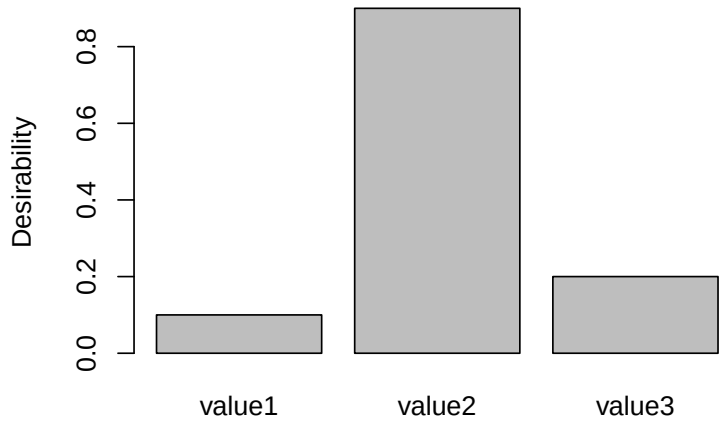

Figure 9: An example of using a desirability function for categorical values.

7 References

Bohachevsky, I. O., Johnson, M. E. and Stein, M. L. (1986), Generalized Simulated Annealing for Function Optimization. *Technometrics* **28**, 209–217.

Box, G. E. P. and Wilson, K. B. (1951), On the Experimental Attainment of Optimum Conditions. *Journal of the Royal Statistical Society, Series B* **13**, 1–45.

Del Castillo, E., Montgomery, D. C., and McCarville, D. R. (1996), Modified Desirability Functions for Multiple Response Optimization. *Journal of Quality Technology* **28**, 337–345.

Derringer, G. and Suich, R. (1980), Simultaneous Optimization of Several Response Variables. *Journal of Quality Technology* **12**, 214–219.

Harington, J. (1965), The Desirability Function. *Industrial Quality Control* **21**, 494–498.

Myers, R. H. and Montgomery, C. M. (1995), *Response Surfaces Methodology: Process and Product Optimization Using Designed Experiments*. New York: Wiley.

Nelder, J. A. and Mead, R. (1965), A Simplex Method for Function Minimization. *Computer Journal* **7**, 308–313.

Olsson, D. M. and Nelson, L. S. (1975), The Nelder–Mead Simplex Procedure for function minimization. *Technometrics* **17**, 45–51.