Chapter 2
Response Surface Models to Analyze Nonlinear Group Phenomena

Andrew Pilny and Amanda R. Slone

2.1 Introduction to Response Surface Methodology

Using Response Surface Methodology (RSM) is a lot like being a chef, mixing together different combinations of ingredients to see which ones come together to make the best dish. In this situation, strict linear thinking no longer applies. For instance, adding just the right amount of salt to a dish can bring out the sweetness in desserts or bump up the taste in more savory dishes. But, too much salt can overwhelm the flavor of a dish, just as too little salt can leave it tasting bland and unsatisfying. Chefs must find that perfect amount of salt that takes their dish from acceptable to exceptional. In addition, chefs must consider how the salt will interact with other ingredients in the dish. For example, salt interacts with the yeast in bread to help create texture, and it helps sausage and other processed meats come together by gelatinizing the proteins. Likewise, RSM helps us find the optimal amount of an outcome variable based on two or more independent variables.

This chapter will provide an introduction on how to use RSM to analyze nonlinear group phenomenon. First, the chapter will outline a brief history and background of the approach. Then, the chapter will walk the reader through a tutorial demonstrating how to execute the second-order model using the PROC RSREG function in SAS. Data previously collected from virtual groups in the game *EverQuestII* (see Williams, Contractor, Poole, Srivastava, & Cai, 2011) will be provided as an example.

A. Pilny (✉) • A.R. Slone
University of Kentucky, Lexington, KY, USA
e-mail: andy.pilny@uky.edu; amanda.slone@uky.edu

© Springer International Publishing AG 2017

A. Pilny, M.S. Poole (eds.), *Group Processes*, Computational Social Sciences,
DOI 10.1007/978-3-319-48941-4_2

2.2 Brief Background of RSM

Box and Wilson's (1951) treatment on polynomial models provided the foundation for RSM, which evolved and developed significantly (e.g., different variations of designs) during the 1970s (Khuri, 2006). Like many statistical methods, RSM developed in the natural sciences, but has yet to be applied extensively within the social sciences given the amount of repeated observations needed for RSM. Indeed, given the complexity involved in running controlled social science experiments and the typically low rate of manipulations, though, it is no wonder that RSM has not taken hold. However, with the advent of Big Data providing virtual Petri dishes of human behavior, RSM has garnered new interest in the social sciences for its ability to answer questions about complex group interactions (Williams et al. 2011. For example, due to its emphasis on optimization (i.e., finding the right combination of independent variables that maximizes a dependent variable), RSM has primarily impacted the world of business and performance management.

2.3 Basic Processes Underlying RSM

RSM is a blend between least squares regression modeling and optimization methods. More formally, RSM can be defined as the "collection of statistical and mathematical techniques useful for developing, improving, and optimizing processes" (Myers, Montgomery, & Anderson-Cook, 2009, p. 1). Moreover, instead of trying to only explain *variance*, RSM also seeks to clarify *optimization*. In other word, it is not necessarily about *how* a set of independent variables explains a dependent variable, but rather *what combination* of independent variables will yield the highest (or lowest) response in a dependent variable. In order to do this, RSM requires at least three variations in each variable, measured on a ratio or interval level.

To conduct an RSM test, there are typically five consecutive steps to go through (SAS Institute, 2013): (1) the regression modeling, (2) lack of fit, (3) coding of variables, (4) canonical analysis, and (5) ridge analysis. Each of these steps is described in more detail below.

2.3.1 Step 1: Second-Order Regression Modeling

The most common and most useful RSM design is the *second-order model* because it is flexible (i.e., not limited to linear trends), easy (i.e., simple to estimate using least-squares), and practical (i.e., has been proven to solve real world problems; Myers et al., 2009). The general linear model formula is identical to that which is used when conducting a regression (Eq. 2.1):

$$y = f\left(x_1, x_2\right) + e \tag{2.1}$$

In this general linear equation, y equals a response variable, x_1 and x_2 represent predictor variables, and e equals the error term.

But, RSM uses the second-order model in order to fully determine the *response shape* (i.e., the observed nonlinear trend). The equation for the second-order model is as follows (Eq. 2.2):

$$\hat{y} = b_0 + x'b + x'\,\hat{\mathrm{B}}\,x \tag{2.2}$$

In this second-order matrix equation, "b_0, b, and $\hat{\mathrm{B}}$ are the estimates of the intercept, linear, and second-order coefficients" (Myers et al., 2009, p. 223) respectively. One thing to note is that, unlike most other second-order regression models used when conducting group research, the results provided in this model are preliminary. That is, the results are used to determine linear, quadratic, and interactional relationships between the independent variables, not to identify the response shape.

2.3.2 Step 2: Lack of Fit

Lack of fit is how well predicted repeated observations match the observed data. In other words, lack of fit of the second-order model indicates that the predicted values of the data do not look like the observed values (see Montgomery, 2005, p. 421–422). For example, though salt (independent variable) may be shown to influence taste (dependent variable) in a second-order model (i.e., statistically significant), when we compare the predicted responses to actual taste ratings (e.g., feedback from customers), there are major discrepancies. This indicates a poorly fitting model.

When we have more than one observation on an independent variable, there are several things to look out for when calculating lack of fit. First, it is important to differentiate *pure error* from *lack of fit error*. Pure error is more common in regression modeling is determined by looking at the sum of squares (Eq. 2.3) variability between each repeated observation of the independent variables (y_{ij}) and the average value of the response variable ($\overline{y}_i$):

$$SS_{PE} \sum_{i=1}^{m} \sum_{j=1}^{n_i} \left(y_{ij} - \overline{y}_i\right)^2 \tag{2.3}$$

Lack of fit error is different because it uses a weighted version of y_{ij} and looks at the actual observed value of the dependent variable, not the average. The Equation (2.4) can be calculated by taking the sum of the difference of the average value of

the response variable ($\overline{y}_i$) the fitted value of the response variable ($\hat{y}_i$), and weighting it by the number of observations at value of the independent variable (n_i):

$$SS_{LOF} \sum_{i=1}^{m} n_i \left(\overline{y}_i - \hat{y}_i \right)^2 \tag{2.4}$$

From there, an *F*-test (Eq. 2.5) can be derived using mean squares (MS) from both equations to determine whether or not a quadratic model is even necessary to replace a reduced first-order model. For instance, if the lack of fit test is not significant for a first-order model, then there could be a reasonable argument that a second-order model is not event needed:

$$F_0 = \frac{MS_{LOF}}{MS_{PE}} \tag{2.5}$$

Likewise, if the test is statistically significant for a second-order model, then by Occam's Razor (i.e., law of parsimony), we have evidence that a quadratic model might not be appropriate.

2.3.3 Step 3: Coding of Variables

Despite requiring variables to be measured at the interval or ratio level, RSM does not simply examine multiple sets of linear relationships. Instead, RSM conducts an experiment of sorts, and organizes variables into conditions to see which results in the optimal output. As such, to make it easier to conduct the canonical analysis (Step 4) and ridge analysis (Step 5), recoding values is a convenient way to examine the response shape at multiple values of the independent variables. As Lenth (2009) put it, "Using a coding method that makes all coded variables in the experiment vary over the same range is a way of giving each predictor an equal share in potentially determining the steepest-ascent path" (p. 3). In addition to simplifying the calculation, recoding the variables also produces results with respect to the original values of the independent variables. A common way to recode variables, as in the SAS package, is to do the following (Eq. 2.6):

$$\text{Coded value} = \frac{\text{Original value} - \text{M}}{\text{S}} \tag{2.6}$$

whereas "*M* is the average of the highest and lowest values for the variable in the design and *S* is half their difference" (SAS Institutive, 2013, p. 7323). For instance, if there were five observations of on salt, ranging from two ounces to ten ounces, then the data for salt are stored in coded form using the following (Eq. 2.7):

$$x_{salt} = \frac{\text{Salt value} - (10 - 2)}{4} \tag{2.7}$$

2.3.4 Step 4: Canonical Analysis of the Response System

The next step is to conduct a canonical analysis of each of the conditions. The purpose of the canonical analysis is to determine the overall shape of the data. For a first-order model, this is typically done through a method of *steepest ascent or descent*, wherein a linear shape determines which region of values creates an optimal response. However, for a second-order model, the shape can look more three-dimensional given the addition of interaction and polynomial terms. Here, we go back to our original Eq. (2.2) of a second-order response in matrix form (see Myers et al., 2009, p. 223):

$$\hat{y} = b_0 + x'b + x'\hat{\mathrm{B}}x$$

To optimize the response ($\hat{y}$) and locate the stationary point (x_s) (i.e., the point of highest response in the dependent variable) we can set the derivative of $\hat{y}$ equal to 0:

$$\frac{\partial \hat{y}}{\partial x} = b + 2\hat{B}x = 0 \tag{2.8}$$

and then solve for the stationary point:

$$x_s = -\frac{1}{2}\hat{\mathrm{B}}^{-1}\mathrm{b}$$

In these equations, *b* equals *a* vector of first-order beta coefficients:

$$\begin{bmatrix} \hat{\beta}_1 \\ . \\ . \\ . \\ \hat{\beta}_q \end{bmatrix} \tag{2.9}$$

And $\hat{\mathrm{B}}$ includes quadratic (diagonals) and interaction (off-diagonals) beta coefficients:

$$\begin{bmatrix} \hat{\beta}_{11,} & \hat{\beta}_{12}/2, & \ldots, & \hat{\beta}_{1q,}/2 \\ & \hat{\beta}_{22,} & \ldots, & \hat{\beta}_{2q,}/2 \\ & & \ddots & \\ sym. & & & \hat{\beta}_{qq,}/2 \end{bmatrix} \tag{2.10}$$

For instance, consider if we trying to maximize taste ($\hat{y}$) with salt (x_1) pepper (x_2). After running a clean second-order model (i.e., no lack of fit), we find that:

$$\mathbf{b} = \begin{bmatrix} 3.65 \\ 4.69 \end{bmatrix}, \hat{\mathrm{B}} = \begin{bmatrix} -1.22 & -0.25 \\ -0.25 & -2.66 \end{bmatrix}$$

then

$$\mathrm{x_s} = -\frac{1}{2}\hat{\mathbf{B}}^{-1}\mathbf{b}$$

$$= -\frac{1}{2}\begin{bmatrix} -0.84 & 0.08 \\ 0.08 & -0.38 \end{bmatrix}\begin{bmatrix} 3.65 \\ 4.69 \end{bmatrix}$$

$$= \begin{bmatrix} 1.34 \\ 0.75 \end{bmatrix}$$

To compute this equation using matrix algebra, the following R code can be used:

```
B = matrix(c( 1.22,-.25,-.25,-2.66),nrow = 2,ncol = 2,byrow = TRUE)
b = matrix(c(3.65,4.69),nrow = 2,ncol = 1,byrow = TRUE)
x = -.5*((solve(B))%*%b)
x
```

As such, the predicted stationary point for taste based on salt and pepper (x_1, x_2) is 1.34 and 0.75. If the hypothetical fitted second-order model is

$$\hat{y} = 69.65 + 12.22x_1 + 3.45x_2 - 9.33x_1^2 - 6.32x_2^2 - 4.65x_1x_2$$

then the predicted highest response of taste $\left(\hat{y}_s\right)$ would be 63.63 by plugging in the optimal values for salt and pepper. It can then be re-expressed in the canonical second order form (this will be useful for later, see Montgomery, 2005, p. 446):

$$\hat{y} = \hat{y}_s + \lambda_1 w_1^2 + \lambda_2 w_2^2$$

$$= 63.63 + 1.34w_1^2 + 0.75w_2^2$$

where w_1 and w_2 are canonical variables (i.e., latent variables in relationship with the original independent variables).

From this point, it is necessary to determine the shape of the stationary point. The eigenvalues (λ) of the canonical analysis give indication to the nature of the shape (see Montogmery, 2005, p. 446):

$$|\mathrm{B} - \lambda I| = 0 \tag{2.11}$$

If the eigenvalues for each independent variable are negative, then a *maximum stationary point* has been reached. A maximum stationary point looks like a hill, meaning that there is a point that indicates a high response. For most research, this is good news because it means that some combinations of variables entered in the model to produce a maximum response in the dependent variable. On the other hand, if they are all positive, then this indicates a *minimum stationary point*, meaning that the data will look like a *valley*. For most research, this is bad news because it means that some combinations of variables entered in the model to produce a minimum response in the dependent variable, unless a decrease in the dependent variable was what was desired of course.

Finally, if the eigenvalues are mixed, this indicates a *saddle point*, meaning that maximum or minimum solutions are not found, but rather multiple regions of high and low variables exist. In other words, the data will look like a series of hills and valleys, or perhaps even a plateau. For instance, a high value of x_1 and low value of x_2 may produce the highest value of y, while at the same time, a low value of x_1 and low value of x_2 may also produce the same value in y. Moreover, if they are all very close, or are at zero, then there is a flat area, meaning that there was little to no relationship between the independent variables and the response variable. Beyond looking at the eigenvalues, a two-dimensional contour plot is also a visual that can easily determine the shape of the response surface.

From our current example,

$$|\mathrm{B} - \lambda \mathrm{I}| = 0$$

$$\begin{bmatrix} -1.22 - \lambda & -0.25 \\ -0.25 & -2.66 - \lambda \end{bmatrix} = 0$$

By taking the determinant of the matrix:

$$(-1.22 - \lambda)(-2.66 - \lambda) - (-0.25 * -0.25) = 0$$

$$\lambda^2 + 3.88\lambda + 3.183 = 0$$

The solution, using basic completing the square calculus, is $\lambda_1 = -1.177$ and $\lambda_2 = -2.70$. As such, because both eigenvalues were negative, it indicates a maximum stationary point. This means that the canonical values for salt and pepper would yield the highest value of taste based on the data the researcher has collected.

2.3.5 Step 5: Conduct Ridge Analysis if Needed

Often when a saddle point is found, or if the researcher wants additional information regarding a maximum or minimum point, a ridge analysis can be performed. The purpose of a ridge analysis is to "anchor the stationary point inside the

experimental region" and to give "some candidate locations for suggested improved operating conditions" (Myers et al., 2009, p. 236). In other words, the ridge analysis provides an estimated response value of y for each of the different values in the independent variables.

For instance, consider if the eigenvalue for pepper was essentially zero, but salt, as we found out, was significantly less than zero (see, Montgomery, 2005, p. 447). From this point, we would want to see what values of salt would yield a high amount of taste by analyzing the predicted response in taste from different values in salt. If the example formula was

$$\hat{y} = 69.65 - 14.87x_1 + 7.94x_2 - 0.33x_1^2 - 8.89x_2^2 + 13.65x_1x_2$$

and the resulting response in canonical form was

$$\hat{y} = \hat{y}_s - 13.56_1 w_1^2 - 0.02w_2^2$$

then we know we can pay more attention to salt because a single unit in the w_2 canonical variable would results in a 13.56 unit change rather than a small 0.02 unit change moved in the w_2 direction. In Table 2.1, a ridge analysis used this information to produce a line of predicted values that might indicate a trend:

Table 2.1 Example ridge analysis table

Order	Estimated response in taste	Un-coded value for salt	Un-coded value for pepper
1	65.24	2	1
2	59.17	3	0.9
3	49.53	4	0.8
4	36.32	5	0.7
5	19.55	6	0.6

From here, one can see how the decreasing levels of salt are related to a higher estimated response in taste, which could prove useful for future design of experiments.

2.4 RSM in Context

To demonstrate the usefulness of RSM in group research, this exemplar study employs data gathered from a download of data on 100,000 characters over 5 months in the Massive Multiplayer Online Game (MMOG) Everquest II (EQII).

2.4.1 About the Game

Commercially launched in November 2004, this game was estimated to have about 200,000 active subscribers in North America alone as of early 2008, the year in which the data was drawn from.[1] These players participate in thousands of teams over the 5 months, making it possible to draw much larger samples and making it possible to identify large samples of teams. Moreover, they incorporate precise metrics for the for team performance outcomes. As such, a random sample of 154 unique groups (i.e., no shared members) was analyzed for this tutorial.

As in most MMOGs, EQII players create a character and advance that character through challenges in a social milieu, typically banding together with other players for help and companionship. For each character, a class is chosen to fit some variation of the three basic archetypes found in nearly every fantasy MMO: damage-dealer, damage-taker and damage-healer. Each archetypal role has different capabilities, weaknesses and strengths, and the choice of class then determines how players develop their characters and how they will interact in the game environment and with other players. Players can communicate with others in the game through text messaging and voice chat.

Following a loose storyline, players use their characters to complete various tasks (quests) in order to earn virtual items such as currency and equipment. One important performance metric is number of "experience points" gained during a quest. Players must accumulate experience points to advance their character level. The character level is a fundamental indicator of players' success in the game. It not only represents a quantitative measure of players' skill and competence, but also determines whether players have access to certain quests and other game content, locations, and equipment. Until they attain the maximum level of 70, the accumulation of experience points is the only way for players to increase their character level. The amount of experience points associated with a given quest is associated with the difficulty of the quests and the value of the items won. Therefore, experience points can be used as a simple yet powerful indicator of players' performance at the common tasks in the game.

At the opposite end of the spectrum, a player can die during a quest. When a player dies in the game, they are not gone forever, but do pay a cost. For instance, for several minutes, the character is very vulnerable and cannot use many of their capabilities until they have had time to refresh many of their spells, buffs, and item effects. Moreover, their armor takes a significant amount of damage and if completely destroyed, the character will have to find a shop to get new armor or get it repaired. Finally, unless they are revived from a teammate, they will likely revive at a location far away from where the quest was being performed. As such, it is in the team's interest to avoid death because it can hinder their progress in the quest.

[1] There is no definitive evidence for the exact size of the population on Everquest II. The number 200,00 is estimated from multiple professional and fan sites such as http://www.mmogchart.com and http://gamespot.com

This study focused on group (i.e., heroic) quests and teams of three to six members. Generally, groups undertaking heroic quests include characters with different capabilities and skills. As discussed earlier, experience level is an important indicator of players' capabilities and competence, and groups often have members with different experience levels. The diversity in experience levels in a group can influence team processes substantially (Valenti & Rockett, 2008). Groups also typically are composed of members with different archetypal roles (i.e., damage-dealers, damage-takers, and damage-healers).

The groups in EQII that undertake heroic quests resemble the action teams described by Sundstrom, De Meuse, and Futrell (1990) in that they have short-term projects with clear goals and standards for evaluation, and members take on specific highly-interdependent roles. Their projects are the quests in the game, which require players to complete certain activities, such as finding objects or information, or killing a monster. Success or failure is clearly indicated by whether the quest is completed or not and whether or not members are killed during the quest. Analogous real world teams include military units, emergency medical response teams, and surgical teams.

2.5 Dependent Variable

2.5.1 Team Performance

Team performance was measured using two metrics. The first was the amount of *experience points* each player earned during the quest. These were obtained through the back-end database. Throughout the quest, characters earn points for successfully completing required tasks (i.e., defeating a monster, finding hidden objects). Likewise, *death* was the second and separate indicator of team performance. The total amount of deaths was calculated and the lower the number of group deaths, the better the performance.

2.6 Independent Variables

2.6.1 Complexity

Task complexity scores for each group were obtained through individually coding each quest. Detailed descriptions of each quest were obtained through *ZAM EverQuest II*, the largest EQII online information database. *ZAM* also features EQII wikis, strategy guides, forums, and chat rooms. Graduate and undergraduate researchers independently coded each quest based on the general definition of

task complexity given by Wood (1986). According to Wood (1986), complexity entails three aspects: (1) component complexity (i.e., the number of acts and information cues in the quests), (2) coordinative complexity (i.e., the type and number of relationships among acts and cues), and (3) dynamic complexity (i.e., the changes in acts and cues, and the relationships among them). These features were used to code the complexity of each quest (*Mean* = 21.11, *SD* = 17.35, *Min* = 4, *Max* = 81).

2.6.2 Difficulty

Difficulty scores for each quest were obtained through Sony Online Entertainment. Each quest is given a static difficulty score ranging from 1 (least difficult) to 70 (most difficult). To create a variable that most closely resembled how difficult it was for the group attempting it, we subtracted the difficulty of the level of the quest from the highest player's character level. Thus, a negative number indicates that the group has at least one player that has a character level much higher that the quest they are attempting, meaning that it will likely be quite easy. On the other hand, a positive number indicates that everybody in the group has a character level below the quest difficulty level, meaning that it will likely be quite difficult to complete (*Mean* = –2.52, *SD* = 6.76, *Min* = –31, *Max* = 13).

2.7 Control Variables

2.7.1 Group Size

The more group members, the more likely there are opportunities for groups to both earn experience points and die. As such, to account for group size, we used group size as a covariate. Groups ranged from three (67.3 %), to four (22.2 %), five (5.2 %), and six (5.2 %) members. Since most groups has three members, the group size of three was used a reference point.

2.8 Data Analysis

The current example carries out RSM in SAS, through the proc. rsreg procedure. SAS is used here because it has perhaps the simplest code, though other programs can easily implement RSM like R and JMP.

2.8.1 Controlling for Group Size

Another benefit of using SAS is that the procedure, including contour plots and ridge analysis, are all done through specifying a few lines of code:

```
data rsm;
set rsm;
g6 = (groupsize = 6);
g5 = (groupsize = 5);
g4 = (groupsize = 4);
g3 = (groupsize = 3);
```

In the above line of code, the first thing that we must do is create the covariate variable. Since we want qualitative variable for each group size, we create four different variables and call them g6, g5, g4, and g3.

2.8.2 Experience Points: A Minimum Stationary Point

The next line of code runs the RSM procedure:

```
ods graphics;
proc rsreg data = rsm plots = (ridge surface);
model experience _ pts = g6 g4 g5 g3 difficulty complexity / covar = 4 lackfit;
ridge max min;
run;
(odsgraphicsoff;)
```

The first line (ods graphics on;), simply tells SAS to turn on the ODS Statistical Graphics (Rodriguez, 2011). These graphics are necessary to produce the contour plots that show the predicted response based on different values of the independent variables. The second line of code does two things. First, it specified the data, which we have named "rsm" (**proc rsreg** data = rsm). Second, it tells the program which types of plots we want form the output. In this case, we want a ridge and surface plot (plots = (ridge surface)). The third line of code specifies the model variables. In model one, we are analyzing experience points as a function of quest difficulty and complexity while treating group size as a covariate. When reading this line of code, the dependent variable should come directly after the model term followed by an equal sign (model experience_pts=). The independent variables should come next (g6 g4 g5 g3 Difficulty Complexity), making sure to have the covariates come first. The covariate command lets the program know that the first four variables are to be treated as covariates and not included in the canonical and ridge analysis (covar = **4**).

The final line of the model command is the lack of fit test, telling the command to include it in the output (lackfit;). Finally, we want to include the ridge analysis to find values of the independent variables that predict a maximum or minimum response in experience points (ridge max min;). After these commands are properly arranged, we must tell the program to run it (**run**;). Turning the ODS Graphics off is useful because it might make future commands run a bit slower, even if they are not using the ODS Graphics.

2.9 Results

The following figures contain screenshots from the actual SAS output to ease in initial interpretation. Figure 2.1 contains the results from the least squares regression, including the interaction and polynomial terms. Before the results, however, are some descriptive information, including how the two independent variables were re-coded

The SAS System

The RSREG Procedure

Coding Coefficients for the Independent Variables		
Factor	**Subtracted off**	**Divided by**
difficulty	-9.000000	22.000000
complexity	42.500000	38.500000

Response Surface for Variable experience_pts: experience_pts	
Response Mean	3232.605656
Root MSE	1359.462094
R-Square	0.1533
Coefficient of Variation	42.0547

Regression	DF	Type I Sum of Squares	R-Square	F Value	Pr > F
Covariates	3	10950853	0.0348	1.98	0.1204
Linear	2	23611455	0.0751	6.39	0.0022
Quadratic	2	9855992	0.0314	2.67	0.0729
Crossproduct	1	3758993	0.0120	2.03	0.1560
Total Model	8	48177294	0.1533	3.26	0.0019

Fig. 2.1 Coding coefficients and ANOVA (experience points)

Residual	DF	Sum of Squares	Mean Square	F Value	Pr > F
Lack of Fit	120	235646022	1963717	1.55	0.1084
Pure Error	24	30485732	1270239		
Total Error	144	266131755	1848137		

Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|	Parameter Estimate from Coded Data
Intercept	1	3661.765063	361.579354	10.13	<.0001	2326.341716
difficulty	1	-24.793149	29.353031	-0.84	0.3997	-587.744282
complexity	1	-53.178658	25.183288	-2.11	0.0364	-792.042760
difficulty*difficulty	1	3.050587	1.336332	2.28	0.0239	1476.484094
complexity*difficulty	1	1.246778	0.874219	1.43	0.1560	1056.020971
complexity*complexity	1	0.515613	0.342138	1.51	0.1340	764.267643
g6	1	1088.103762	514.943437	2.11	0.0363	1088.103762
g4	1	354.285863	276.326599	1.28	0.2019	354.285863
g5	1	292.070086	511.576337	0.57	0.5689	292.070086
g3	0	0	.	.	.	0

Factor	DF	Sum of Squares	Mean Square	F Value	Pr > F	Label
difficulty	3	13954485	4651495	2.52	0.0606	difficulty
complexity	3	27768937	9256312	5.01	0.0025	complexity

Fig. 2.2 Model coefficients and lack of fit (experience points)

for the canonical and ridge analysis, and descriptives for the dependent variable, which in this case is experience points ($M = 3232.61$). The omnibus analysis of variance table compares the different models (e.g., linear, quadratic, cross-product, covariate) to an intercept-only model in order to determine how much of an effect they add. For instance, because the quadratic terms by themselves ($F = 2.67$, $p = 0.07$), more or less, provide a better fit than an intercept-only model, it means they will likely be influential predicting an optimal or minimal response surface.

Figure 2.2 displays information on the lack of fit test and individual estimates for each independent variable. Overall, the lack of fit test was just above a 0.10 threshold for significance ($p = 0.11$). While this is generally acceptable as a rule of thumb, it points to some concern about how well the model predicted the actual response of experience points. Nevertheless, there were both linear and nonlinear effects in the model. For instance, there was a negative linear relationship with complexity ($t = -2.11$, $p = 0.03$), meaning that groups earned more experience points with less complex tasks. On the other hand, while there was not a linear relationship with difficulty ($t = -0.84$, $p = 0.40$), there was a quadratic effect ($t = 2.28$, $p = 0.02$),

Fig. 2.3 Canonical analysis (experience points)

The SAS System

The RSREG Procedure
Canonical Analysis of Response Surface Based on Coded Data

Factor	Critical Value		Label
	Coded	Uncoded	
difficulty	0.018236	-8.598818	difficulty
complexity	0.505573	61.964548	complexity
Predicted value at stationary point: 2271.661362			

Eigenvalues	Eigenvectors	
	difficulty	complexity
1757.249593	0.882936	0.469494
483.502144	-0.469494	0.882936
Stationary point is a minimum.		

meaning that there is a certain difficulty peak where groups tend to earn more experience points. To further investigate that point, a canonical analysis is useful here.

Figure 2.3 shows the results of the canonical analysis. Because the eigenvalues for difficulty ($\lambda_1 = 1757.25$) and complexity ($\lambda_2 = 483.50$) were both positive, this means the unique solution is a minimum. In other words, a unique combination of difficulty and class can yield a solution in which groups earned the *least* amount of experience points. As such, these two variables cannot tell us much about high performing groups, but do tell us a lot about *low* performing groups. Moreover, because difficulty was over three times the value of complexity, it means that experience points changes more rapidly along changes in difficulty compared to complexity. Finally, the table below gives the solution for the predicted minimum stationary point of 2271.66 experience points at a value of −8.59 for difficulty and 61.96 for complexity. Because the mean values for each variable is −2.52 and 21.11, this means that groups perform the worst when they choose quests that are about 40 units higher in complexity than average and when the groups highest member is about 8 units less than the quest value, which is higher than average.

Figure 2.4 shows the ridge analysis for a minimum solution. The quadratic effect for difficulty is clearly evidence here as the values fluctuate from moving higher from −9 to −7.87, then decreasing from −7.87 to −13.07. This is important because the relationship as demonstrated by the regression model is not linear, suggesting that the difficulty of the quest compared to the highest-level character in the group has a tipping point (~−8.59). On the other hand, though there is an overall negative linear trend with complexity according to the regression model, the ridge solution paints a more complicated picture. For instance, almost equal predicted responses are obtained with a complexity value of 42.50 and 80.33. These results are in line with

The SAS System

The RSREG Procedure

Estimated Ridge of Minimum Response for Variable experience_pts: experience_pts				
			Uncoded Factor Values	
Coded Radius	Estimated Response	Standard Error	difficulty	complexity
0.0	2477.237861	305.414405	-9.000000	42.500000
0.1	2393.253525	298.196900	-7.903294	45.837519
0.2	2335.568630	295.048816	-7.364132	49.648045
0.3	2298.871725	300.511618	-7.412100	53.710737
0.4	2278.533530	318.542869	-7.869005	57.772282
0.5	2271.681982	351.784851	-8.553852	61.734160
0.6	2276.769309	401.333098	-9.361888	65.591317
0.7	2292.959761	466.932201	-10.239946	69.362502
0.8	2319.777660	547.615192	-11.160508	73.067055
0.9	2356.935769	642.273088	-12.108464	76.720329
1.0	2404.251190	749.935315	-13.074990	80.333788

Fig. 2.4 Ridge analysis of minimum response (experience points)

the critical value threshold demonstrating that a complexity value near 60 is where groups are predicted to perform the least, much higher than average ($M_{compelxity} = 21.11$).

Although no maximum solution was found, a ridge analysis for maximum ascent tends to demonstrate simple linear effects for both difficulty and complexity (see Fig. 2.5). More specifically, groups are predicted to perform better when the character level of the highest character approaches the same level of the quest and the complexity of the quest increases. This makes sense because it means that the quest should not be as challenging for the group if they have at least one character in the group that is close to the quest difficulty level. The quest is still complex enough for group members to do activities that will give them a chance to earn points. However, no solid conclusions should be drawn from this. Instead, it may serve as an impetus to collect more data for future analysis.

Finally, Fig. 2.6 is a visualization of the response surface analysis as a contour plot, with covariates fixed at their average values. This means that this plot is most relevant for groups of three, which were the majority of groups playing this game. The minimum solution can be easily visualized by looking at the large ring representing values below 3000. Values closer to the center of that ring are the lowest predicted values of experience points. If you cross the intersection between the two critical values of −8.59 for difficulty and 61.96 for complexity, one can pinpoint to the center of the ring. The circles represent the predicted values for each observation.

The SAS System

The RSREG Procedure

Estimated Ridge of Maximum Response for Variable experience_pts: experience_pts				
			Uncoded Factor Values	
Coded Radius	Estimated Response	Standard Error	difficulty	complexity
0.0	2477.237861	305.414405	-9.000000	42.500000
0.1	2591.514039	313.627581	-10.456265	39.614201
0.2	2738.304958	322.498612	-12.106776	37.047424
0.3	2918.733427	334.045714	-13.861558	34.688401
0.4	3133.381678	351.890047	-15.675575	32.465815
0.5	3382.571431	380.356503	-17.525517	30.335855
0.6	3666.492302	423.364871	-19.398557	28.271090
0.7	3985.262567	483.522055	-21.287199	26.253747
0.8	4338.959535	561.909924	-23.186815	24.271897
0.9	4727.635805	658.499613	-25.094415	22.317271
1.0	5151.328370	772.719727	-27.007990	20.383968

Fig. 2.5 Ridge analysis maximum response (experience points)

2.9.1 Model for Deaths: A Saddle Point

For the model with deaths as the response surface, we use the same code except switch the dependent variable form experience points to deaths:

```
ods graphics on;
proc rsreg data = rsm plots = (ridge surface);
model deaths = g6 g4 g5 g3 difficulty complexity / covar = 4 lackfit;
ridge max min;
run;
odsgraphics off;
```

The initial outputs in Fig. 2.7 details similar information about the coded variables and analysis of variance.

As you can see in Fig. 2.7, there is a significant difference between an intercept only model and the linear, quadratic and cross-product models, suggesting that the

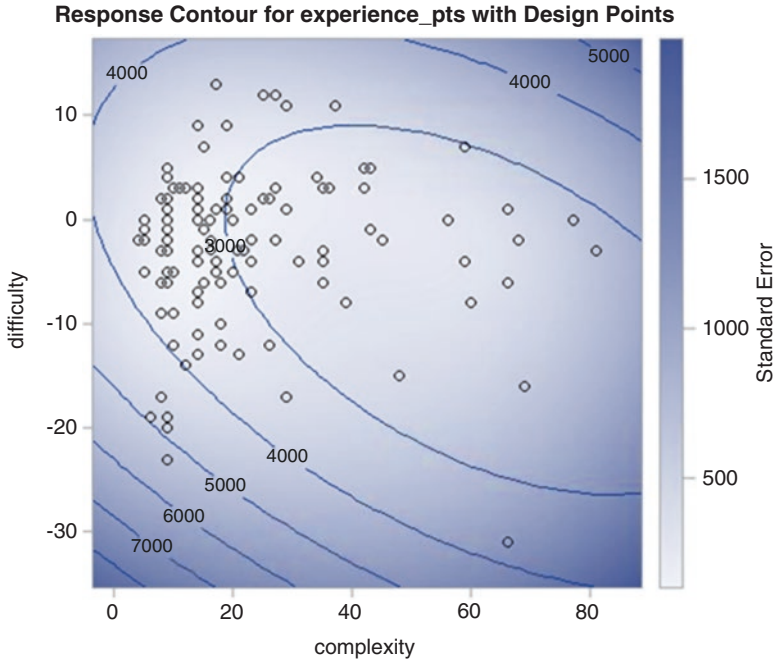

Fig. 2.6 Contour plot (experience points)

variables have considerable influence on deaths. However, as demonstrated by Fig. 2.8, the full model has a significant lack of fit, meaning that the average values of death deviate more than we would expect by chance from the predicted responses of deaths.

Indeed, although there are significant effects regarding the difficulty term ($t = 3.45$, $p < .01$) and overall interaction term (i.e., difficulty*complexity, $t = 2.38$, $p = .02$), the lack of fit finding puts a hitch into the entire analysis because it means that we cannot generalize much of the subsequent canonical and ridge analysis. From here, this usually means the researcher might look into some additional reasons for the lack of fit. For instance, there may not be enough variability in deaths and it might be useful to transform it to make it look more normally distributed (e.g., log linear transformation). Alternatively, the researchers might attempt to add more data or additional explanatory variables. Nevertheless, for demonstration, we will carry on with the canonical and ridge analysis.

As expected, there was no unique solution because of the saddle point response shape as demonstrated by the mixed signs of the eigenvalues (see Fig. 2.9). Nevertheless, the eigenvalue for difficulty ($\lambda_1 = 12.51$) is quite larger for complexity ($\lambda_2 = -1.03$), suggesting that there was more variability regarding changes in difficulty. Because there was a significant quadratic interaction, it is useful to look

The SAS System

The RSREG Procedure

Coding Coefficients for the Independent Variables		
Factor	Subtracted off	Divided by
difficulty	-9.000000	22.000000
complexity	42.500000	38.500000

Response Surface for Variable deaths: deaths	
Response Mean	3.718954
Root MSE	6.680704
R.Square	0.1725
Coefficient of Variation	179.6393

Regression	DF	Type I Sum of Squares	R-Square	F Value	Pr > F
Covariates	3	322.150612	0.0415	2.41	0.0698
Linear	2	394.317475	0.0508	4.42	0.0137
Quadratic	2	371.536805	0.0478	4.16	0.0175
Crossproduct	1	251.930606	0.0324	5.64	0.0188
Total Model	8	1339.935498	0.1725	3.75	0.0005

Fig. 2.7 Coding coefficients and ANOVA (deaths)

at a maximum ridge analysis to see what exact levels of difficulty were more associated with more deaths.

The ridge analysis complicates things even further because although the regression model suggested a nonlinear effect on difficulty, the ridge analysis does suggest a linear relationship (see Fig. 2.10). In other words, the more groups attempt quests that have difficulty levels higher than their highest level character, they are more likely to die in that attempt. Again, however, this might be due to a lack of fit.

Finally, the contour plot in Fig. 2.11 visually demonstrates the relationship between difficulty and complexity as it relates to the number of deaths incurred on a question. The wide open space in the middle indicates the least amount of deaths, but does not reveal a solution because those groups varied too widely on complexity and difficulty. Moreover, the bottom left and top right corners specify very high predicted values of deaths, meaning that no maximum solution could be found either because the existence of these high values occurs at seemingly opposite ends of the spectrum. That is, a high number of deaths can occur at a combination of either high complexity and low difficulty, or high difficulty and low complexity.

Residual	DF	Sum of Squares	Mean Square	F Value	Pr > F
Lack of Fit	120	6147.812868	51.231774	4.40	<.0001
Pure Error	24	279.166667	11.631944		
Total Error	144	6426.979535	44.631802		

Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|	Parameter Estimate from Coded Data
Intercept	1	1.085448	1.776883	0.61	0.5422	1.055083
difficulty	1	0.128346	0.144247	0.89	0.3751	3.402473
complexity	1	0.060029	0.123756	0.49	0.6284	-0.058813
difficulty*difficulty	1	0.022638	0.006567	3.45	0.0007	10.956709
complexity*difficulty	1	0.010207	0.004296	2.38	0.0188	8.645243
complexity*complexity	1	0.000357	0.001681	0.21	0.8324	0.528477
g6	1	0.142864	2.530548	0.06	0.9551	0.142864
g4	1	1.764094	1.357931	1.30	0.1960	1.764094
g5	1	6.893767	2.514002	2.74	0.0069	6.893767
g3	0	0	.	.	.	0

Factor	DF	Sum of Squares	Mean Square	F Value	Pr > F	Label
difficulty	3	799.986242	266.662081	5.97	0.0007	difficulty
complexity	3	343.854037	114.618012	2.57	0.0568	complexity

Fig. 2.8 Model coefficients and lack of fit (deaths)

2.10 Conclusion

With the advent of mass amounts of data (e.g., trace data), it is possible to extract a bulk amount of information on how groups face different environments, process information, and perform. RSM inherently requires multiple observations on similar values of variables and is in a unique position to exploit such data. The main contribution of RSM is optimization. That is, through enough data collection, RSM can specify the conditions that are most likely to lead towards a certain outcome.

For instance, in the current example, traditional methods like regression and ANOVA would have been able to detect nonlinear relationships between difficulty and complexity, but they would not have been able to detect the specific values that can yield a certain outcome. The canonical analysis that RSM provides is more practical because it adds specific values and a contour plot that demonstrates how an outcome fluctuates based on different values of the independent variables. In this sense, the contour plot is a lot like a road map, guiding the researcher towards optimal paths that can yield insightful suggestions for practical implications.

Fig. 2.9 Canonical analysis (deaths)

The SAS System

The RSREG Procedure
Canonical Analysis of Response Surface Based on Coded Data

Factor	Critical Value		Label
	Coded	Uncoded	
difficulty	0.079581	-7.249208	difficulty
complexity	-0.595284	19.581569	complexity
Predicted value at stationary point: 1.967925			

Eigenvalues	Eigenvectors	
	difficulty	complexity
12.515485	0.940705	0.339227
-1.030299	-0.339227	0.940705
Stationary point is a saddle point.		

The SAS System

The RSREG Procedure

Estimated Ridge of Maximum Response for Variable deaths: deaths

Coded Radius	Estimated Response	Standard Error	Uncoded Factor Values	
			difficulty	complexity
0.0	1.815033	1.500875	-9.000000	42.500000
0.1	2.270613	1.406639	-6.822034	43.043526
0.2	2.968788	1.320360	-4.692571	44.071160
0.3	3.915048	1.248370	-2.587764	45.235399
0.4	5.110684	1.214818	-0.495253	46.455653
0.5	6.556150	1.256220	1.590387	47.704213
0.6	8.251648	1.405920	3.671825	48.969051
0.7	10.197280	1.676481	5.750511	50.244105
0.8	12.393103	2.060827	7.827300	51.525993
0.9	14.839153	2.545683	9.902726	52.812677
1.0	17.535451	3.119825	11.977140	54.102855

Fig. 2.10 Ridge analysis for maximum response (deaths)

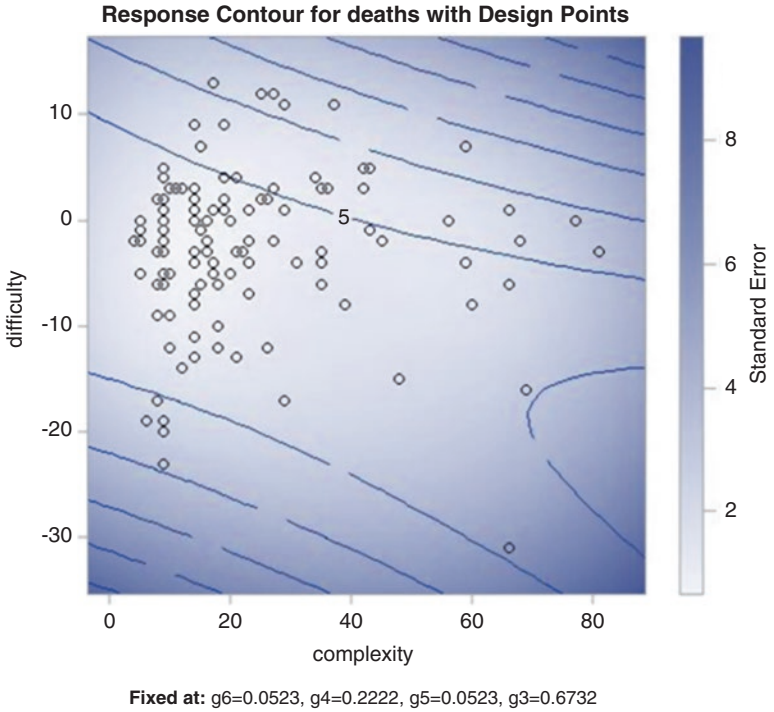

Fig. 2.11 Contour plot (deaths)

For instance, in EQII, groups are faced with decisions on which quests to attempt. Although traditional methods can detect relationships, they do provide an easy go-to guide that can be useful for actual decision-making. RSM, on the other hand, provides a very useful heuristic to help groups make decisions. For example, before a group attempts a quest, they can locate the values of the current quest and group (e.g., its difficulty) and pinpoint via the contour plot where their performance is predicted to land. If it lands on a very low performance spectrum, then this could be used as an important piece of information on whether or not that group should attempt to take on the quest.

Theoretically, RSM has the ability to test and examine a number of theoretical perspective. Notably, however, RSM has a unique opportunity to examine the basic tenets of chaos theory (see Tutzauer, 1996, for an application to organizations and groups), which highlights notions of unpredictability and unstableness. For instance, canonical and ridge analysis might not be very clean at times. That is, results that yield saddle points do not necessarily mean null findings. Instead, they have the ability to show how even small fluctions in the independent variables could cause dramatic changes in an outcome variable. Indeed, chaos theory would predict that in many contexts, a simple unique solution is not possible.

References

Box, G. E. P., & Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society*, *B13*(1), 1–45.

Khuri, A. I. (2006). *Response surface methodology and related topics*. Singapore: World Scientific Publishing.

Lenth, R. V. (2009). Response-surface methods in R, using rsm. *Journal of Statistical Software*, *32*(7), 1–17.

Montgomery, D. (2005). *Design and analysis of experiments*. Hoboken, NJ: Willey.

Myers, R. H., Montgomery, D. C., & Anderson-Cook, C. (2009). *Response surface methodology*. Hoboken, NJ: Wiley.

SAS Institute Inc. (2013). SAS/STAT 9.3 User's Guide. Cary, NC: SAS Institute Inc.

Sundstrom, E., De Meuse, K. P., & Futrell, D. (1990). Work teams: Applications and effectiveness. *American Psychologist*, *45*(2), 120.

Rodriguez, R. N. (2011), An Overview of ODS Statistical Graphics in SAS 9.3, Technical report, SAS Institute Inc.

Tutzauer, F. (1996). Chaos and organization. In G. Barnett, & L. Thayler (Eds.), *Organization—Communication: The renaissance in systems thinking* (vol. 5, pp. 255–273). Greenwich, CN: Ablex Publishing.

Valenti, M. A., & Rockett, T. (2008). The effects of demographic differences on forming intragroup relationships. *Small Group Research*, *39*(2), 179–202.

Williams, D., Contractor, N., Poole, M. S., Srivastava, J., & Cai, D. (2011). The Virtual Worlds Exploratorium: Using large-scale data and computational techniques for communication research. *Communication Methods and Measures*, *5*(2), 163–180.

Wood, R. E. (1986). Task complexity: Definition of the construct. *Organizational Behavior and Human Decision Processes*, *37*(1), 60–82.