Section 1

MANE 6313

Subsection 1

Week 13, Module D

Student Learning Outcome

- Select an appropriate experimental design with one or more factors,
- Select an appropriate model with one or more factors,
- Evaluate statistical analyses of experimental designs,
- Assess the model adequacy of any experimental design, and
- Interpret model results.

Module Learning Outcome

Employ rsm() to design and analyze RSM models.

RSM Design in Package rsm()

- rsm() supports two designs:
  - Box-Behnken design with 2 to 7 factors
  - Center Composite Design (ccd)
- Details and examples of using both designs will be provided in separate modules

Model Types

- The model formula for designs in rsm() is different
  - FO() specifies first-order model
  - TWI() is used to generate two-way interations
  - PQ() is used to add pure quadratic terms to model
  - SO() creates all terms (FO(),TWI(),PQ()) in a model

Coded Variables

- When analyzing RSM models, variables should be coded
- The R chunk for creating coded variables in a model is shown below

```
library(rsm)
bb3.design <- bbd(3,n0=2,coding=list(x1~(Force-20)/3,x2~(Rate-
print(bb3.design)
```

Coded Variables Output

```r
library(rsm)
bb3.design <- bbd(3,n0=2,coding=list(x1~(Force-20)/3,x2~(Rate-50)/10,x3~Polish-4))
print(bb3.design)
```

```
   run.order std.order Force Rate Polish
1          1         7    17   50      5
2          2        10    20   60      3
3          3         9    20   40      3
4          4        14    20   50      4
5          5         3    17   60      4
6          6         5    17   50      3
7          7         4    23   60      4
8          8         6    23   50      3
9          9        13    20   50      4
10        10        12    20   60      5
11        11         8    23   50      5
12        12         1    17   40      4
13        13        11    20   40      5
14        14         2    23   40      4

Data are stored in coded form using these coding formulas ...
x1 ~ (Force - 20)/3
x2 ~ (Rate - 50)/10
x3 ~ Polish - 4
```

Adding a Response Variable

```
y <- rnorm(14)
bb3.design$y <- y
print(bb3.design)
```

Adding a Response Variable Output

```r
y <- rnorm(14)
bb3.design$y <- y
print(bb3.design)
```

```
   run.order std.order Force Rate Polish          y
1          1         7    17   50      5  1.3724421
2          2        10    20   60      3  1.7480628
3          3         9    20   40      3 -0.5099157
4          4        14    20   50      4  0.8571949
5          5         3    17   60      4 -1.5198610
6          6         5    17   50      3  1.2546819
7          7         4    23   60      4  0.1785562
8          8         6    23   50      3 -0.2568965
9          9        13    20   50      4 -1.8243286
10        10        12    20   60      5  1.2297702
11        11         8    23   50      5  0.3593844
12        12         1    17   40      4  1.0656732
13        13        11    20   40      5  1.1265194
14        14         2    23   40      4  0.9289255

Data are stored in coded form using these coding formulas ...
x1 ~ (Force - 20)/3
x2 ~ (Rate - 50)/10
```

Model Fitting

```
bb3.fitted <- rsm(y~SO(x1,x2,x3),data=bb3.design)
summary(bb3.fitted)
```

Model Fitting Output

```r
bb3.fitted <- rsm(y~SO(x1,x2,x3),data=bb3.design)
summary(bb3.fitted)
```

```
Call:
rsm(formula = y ~ SO(x1, x2, x3), data = bb3.design)

            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.48357    1.10443 -0.4378   0.6841
x1          -0.12037    0.55222 -0.2180   0.8381
x2          -0.12183    0.55222 -0.2206   0.8362
x3           0.23152    0.55222  0.4193   0.6966
x1:x2        0.45879    0.78095  0.5875   0.5884
x1:x3        0.12463    0.78095  0.1596   0.8809
x2:x3       -0.53868    0.78095 -0.6898   0.5283
x1^2         0.21534    0.87313  0.2466   0.8173
x2^2         0.43155    0.87313  0.4943   0.6470
x3^2         0.95063    0.87313  1.0888   0.3375

Multiple R-squared:  0.3734,    Adjusted R-squared:  -1.036
F-statistic: 0.2649 on 9 and 4 DF,  p-value: 0.9546

Analysis of Variance Table

Response: y
                Df Sum Sq Mean Sq F value Pr(>F)
FO(x1, x2, x3)   3 0.6635  0.2212  0.0907 0.9614
TWI(x1, x2, x3)  3 2.0648  0.6883  0.2821 0.8367
PQ(x1, x2, x3)   3 3.0869  1.0290  0.4218 0.7479
Residuals        4 9.7581  2.4395
Lack of fit      3 6.1628  2.0543  0.5714 0.7223
Pure error       1 3.5953  3.5953

Stationary point of response surface:
        x1         x2         x3
 1.2224666 -0.7710184 -0.4203606

Stationary point in original units:
    Force      Rate    Polish
23.667400 42.289816  3.579639

Eigenanalysis:
eigen() decomposition
$values
[1] 1.06647794 0.50637516 0.02466496

$vectors
          [,1]       [,2]       [,3]
x1 -0.04164836 -0.6261646  0.7785777
```