# Chapter 10

# Robust Parameter Design and Process Robustness Studies

# 10.1 Example 10.5, p. 511

## 10.1.1 Model mean and ln of variance separately

Read data and calculate mean, variance, log-variance, and SNR.

```
#### 10.5
fn.data <- "http://statacumen.com/teach/RSM/data/RSM_EX_10-05.txt"
df.10.5 <- read.table(fn.data, header=TRUE)
df.10.5
```

```
##   x1 x2    y1    y2      y3    y4
## 1 -1 -1 33.50 41.23 25.2683 31.99
## 2 -1  0 35.82 38.07 32.7928 34.04
## 3 -1  1 33.08 31.84 36.2500 34.02
## 4  0 -1 30.45 41.29 15.1493 23.99
## 5  0  0 34.87 40.23 27.7724 31.13
## 6  0  1 35.22 37.10 33.3280 35.21
## 7  1 -1 21.16 34.11  0.7917 15.74
## 8  1  0 27.67 38.15 15.5132 25.99
## 9  1  1 32.12 38.12 26.1673 32.16
```

```
# calculate some summary statistics, especially SNR
df.10.5a       <- df.10.5
df.10.5a$m     <- apply(df.10.5a[,c("y1","y2","y3","y4")], 1, mean)
df.10.5a$s     <- apply(df.10.5a[,c("y1","y2","y3","y4")], 1, sd)
df.10.5a$s2    <- apply(df.10.5a[,c("y1","y2","y3","y4")], 1, var)
df.10.5a$logs2 <- log(df.10.5a$s2)
df.10.5a$SNR   <- apply(df.10.5a[,c("y1","y2","y3","y4")], 1
                    , function(x) {
                        -10 * log10( sum(1 / x^2) / length(x) )
                      })

# average SNR over each condition, independently
library(plyr)
df.10.5a.SNR <- rbind(
  ddply(df.10.5a, .(x1), function(.X) {
    data.frame(
        var   = "x1"
      , level = .X$x1[1]
      , m.SNR = mean(.X$SNR)
      , s.SNR = sd(.X$SNR)
      , min.SNR = min(.X$SNR)
      , max.SNR = max(.X$SNR)
      )
    })[,-1]
    ,
  ddply(df.10.5a, .(x2), function(.X) {
    data.frame(
        var   = "x2"
      , level = .X$x2[1]
      , m.SNR = mean(.X$SNR)
      , s.SNR = sd(.X$SNR)
```

```
      , min.SNR = min(.X$SNR)
      , max.SNR = max(.X$SNR)
      )
    })[,-1]
  )
df.10.5a.SNR
```

```
##   var level m.SNR   s.SNR min.SNR max.SNR
## 1  x1    -1 30.47  0.4600  29.976   30.89
## 2  x1     0 29.43  2.0275  27.122   30.92
## 3  x1     1 20.36 14.2561   3.972   29.91
## 4  x2    -1 20.36 14.2607   3.972   29.98
## 5  x2     0 29.45  1.9742  27.196   30.89
## 6  x2     1 30.46  0.5092  29.909   30.92
```

First, model mean and ln of variance separately, as suggested on p. 519 of text.

Fit second-order linear model for $\bar{y}$ and $\log(s^2)$.

```
library(rsm)
rsm.10.5a.m.SOx1x2 <- rsm(m ~ SO(x1, x2), data = df.10.5a)
# externally Studentized residuals
rsm.10.5a.m.SOx1x2$studres <- rstudent(rsm.10.5a.m.SOx1x2)
summary(rsm.10.5a.m.SOx1x2)
```

```
##
## Call:
## rsm(formula = m ~ SO(x1, x2), data = df.10.5a)
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.3889     0.0717   465.5  2.2e-08 ***
## x1           -4.1752     0.0393  -106.3  1.8e-06 ***
## x2            3.7481     0.0393    95.4  2.5e-06 ***
## x1:x2         3.3485     0.0481    69.6  6.5e-06 ***
## x1^2         -2.3277     0.0680   -34.2  5.5e-05 ***
## x2^2         -1.8670     0.0680   -27.4  0.00011 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:     1,Adjusted R-squared:     1
## F-statistic: 5.43e+03 on 5 and 3 DF,  p-value: 3.94e-06
##
## Analysis of Variance Table
##
## Response: m
##             Df Sum Sq Mean Sq F value  Pr(>F)
## FO(x1, x2)   2  188.9    94.4   10198 1.8e-06
## TWI(x1, x2)  1   44.8    44.8    4843 6.5e-06
## PQ(x1, x2)   2   17.8     8.9     961 6.1e-05
## Residuals    3    0.0     0.0
## Lack of fit  3    0.0     0.0
## Pure error   0    0.0
##
```

```
## Stationary point of response surface:
##      x1      x2
## -0.4926  0.5620
##
## Eigenanalysis:
## $values
## [1] -0.4073 -3.7874
##
## $vectors
##       [,1]    [,2]
## x1 -0.6572 -0.7538
## x2 -0.7538  0.6572
```

```
library(rsm)
rsm.10.5a.logs2.SOx1x2 <- rsm(logs2 ~ SO(x1, x2), data = df.10.5a)
# externally Studentized residuals
rsm.10.5a.logs2.SOx1x2$studres <- rstudent(rsm.10.5a.logs2.SOx1x2)
summary(rsm.10.5a.logs2.SOx1x2)
```

```
##
## Call:
## rsm(formula = logs2 ~ SO(x1, x2), data = df.10.5a)
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.9863     0.5415    5.51    0.012 *
## x1            1.0350     0.2966    3.49    0.040 *
## x2           -1.4212     0.2966   -4.79    0.017 *
## x1:x2         0.1095     0.3633    0.30    0.783
## x1^2          0.2516     0.5137    0.49    0.658
## x2^2          0.0262     0.5137    0.05    0.963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.922,Adjusted R-squared:  0.792
## F-statistic: 7.09 on 5 and 3 DF,  p-value: 0.0688
##
## Analysis of Variance Table
##
## Response: logs2
##             Df Sum Sq Mean Sq F value Pr(>F)
## FO(x1, x2)   2  18.55    9.27   17.57  0.022
## TWI(x1, x2)  1   0.05    0.05    0.09  0.783
## PQ(x1, x2)   2   0.13    0.06    0.12  0.890
## Residuals    3   1.58    0.53
## Lack of fit  3   1.58    0.53
## Pure error   0   0.00
##
## Stationary point of response surface:
##     x1     x2
## -14.60  57.68
##
## Eigenanalysis:
## $values
```

```
## [1] 0.26416 0.01359
##
## $vectors
##       [,1]    [,2]
## x1 -0.9746  0.2241
## x2 -0.2241 -0.9746
```

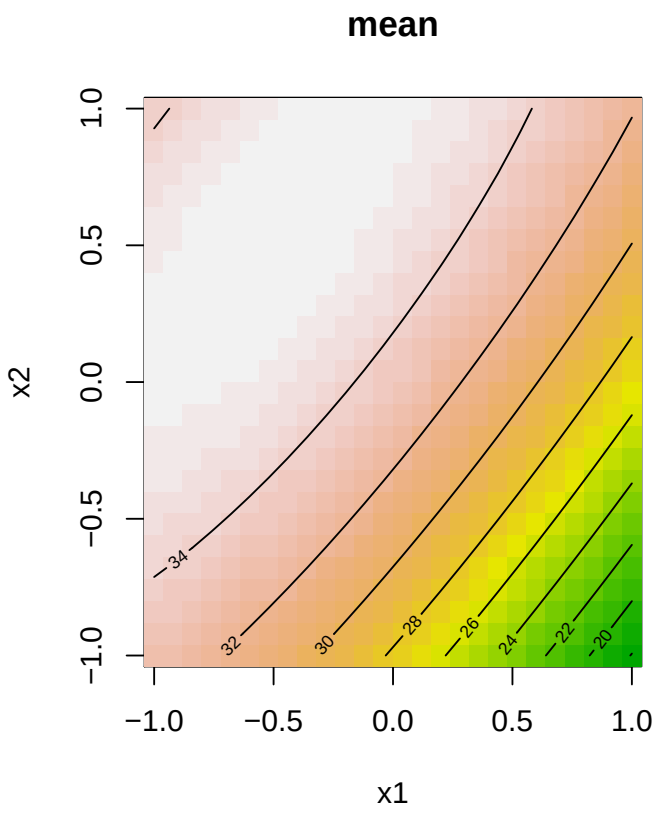
```
par(mfrow = c(1,2))
# this is the stationary point
canonical(rsm.10.5a.m.SOx1x2)$xs
```

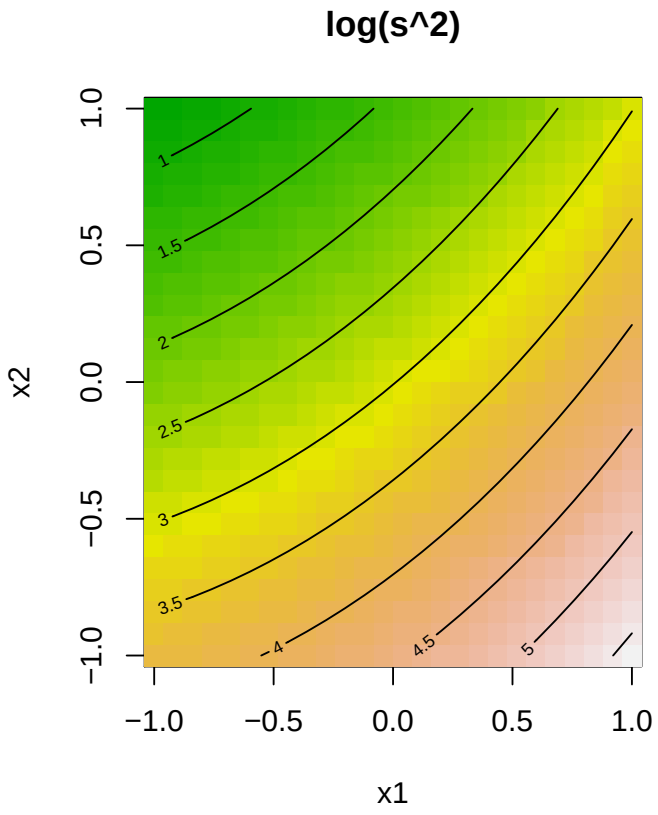
```
##      x1      x2
## -0.4926  0.5620
```

```
contour(rsm.10.5a.m.SOx1x2, ~ x1 + x2, image=TRUE
      , at = canonical(rsm.10.5a.m.SOx1x2)$xs, main = "mean")
# this is the stationary point
canonical(rsm.10.5a.logs2.SOx1x2)$xs
```

```
##     x1     x2
## -14.60  57.68
```

```
contour(rsm.10.5a.logs2.SOx1x2, ~ x1 + x2, image=TRUE
      , at = canonical(rsm.10.5a.logs2.SOx1x2)$xs, main = "log(s^2)")
```



### 10.1.2 Model mean and ln of variance as on p. 512 (eq. 10.20)

Reshape data and create $z_1$ and $z_2$ variables.

```
#### 10.5b
fn.data <- "http://statacumen.com/teach/RSM/data/RSM_EX_10-05.txt"
df.10.5b <- read.table(fn.data, header=TRUE)
df.10.5b
```

```
##   x1 x2    y1    y2      y3    y4
## 1 -1 -1 33.50 41.23 25.2683 31.99
## 2 -1  0 35.82 38.07 32.7928 34.04
## 3 -1  1 33.08 31.84 36.2500 34.02
## 4  0 -1 30.45 41.29 15.1493 23.99
## 5  0  0 34.87 40.23 27.7724 31.13
## 6  0  1 35.22 37.10 33.3280 35.21
## 7  1 -1 21.16 34.11  0.7917 15.74
## 8  1  0 27.67 38.15 15.5132 25.99
## 9  1  1 32.12 38.12 26.1673 32.16
```

```
# reshape data into long format
library(reshape2)
df.10.5b.long <- melt(df.10.5b, id.vars = c("x1", "x2"), variable.name = "rep", value.name = "

# create z1 and z2 variables
df.10.5b.long$z1 <- NA
df.10.5b.long$z2 <- NA
df.10.5b.long[(df.10.5b.long$rep == "y1"), c("z1", "z2")] <- data.frame(z1 = -1, z2 = -1)
df.10.5b.long[(df.10.5b.long$rep == "y2"), c("z1", "z2")] <- data.frame(z1 = -1, z2 =  1)
df.10.5b.long[(df.10.5b.long$rep == "y3"), c("z1", "z2")] <- data.frame(z1 =  1, z2 = -1)
df.10.5b.long[(df.10.5b.long$rep == "y4"), c("z1", "z2")] <- data.frame(z1 =  1, z2 =  1)
df.10.5b.long
```

```
##    x1 x2 rep       y z1 z2
## 1  -1 -1  y1 33.5021 -1 -1
## 2  -1  0  y1 35.8234 -1 -1
## 3  -1  1  y1 33.0773 -1 -1
## 4   0 -1  y1 30.4481 -1 -1
## 5   0  0  y1 34.8679 -1 -1
## 6   0  1  y1 35.2202 -1 -1
## 7   1 -1  y1 21.1553 -1 -1
## 8   1  0  y1 27.6736 -1 -1
## 9   1  1  y1 32.1245 -1 -1
## 10 -1 -1  y2 41.2268 -1  1
## 11 -1  0  y2 38.0689 -1  1
## 12 -1  1  y2 31.8435 -1  1
## 13  0 -1  y2 41.2870 -1  1
## 14  0  0  y2 40.2276 -1  1
## 15  0  1  y2 37.1008 -1  1
## 16  1 -1  y2 34.1086 -1  1
## 17  1  0  y2 38.1477 -1  1
## 18  1  1  y2 38.1193 -1  1
## 19 -1 -1  y3 25.2683  1 -1
```

```
## 20 -1  0  y3 32.7928  1 -1
## 21 -1  1  y3 36.2500  1 -1
## 22  0 -1  y3 15.1493  1 -1
## 23  0  0  y3 27.7724  1 -1
## 24  0  1  y3 33.3280  1 -1
## 25  1 -1  y3  0.7917  1 -1
## 26  1  0  y3 15.5132  1 -1
## 27  1  1  y3 26.1673  1 -1
## 28 -1 -1  y4 31.9930  1  1
## 29 -1  0  y4 34.0383  1  1
## 30 -1  1  y4 34.0162  1  1
## 31  0 -1  y4 23.9883  1  1
## 32  0  0  y4 31.1321  1  1
## 33  0  1  y4 35.2085  1  1
## 34  1 -1  y4 15.7450  1  1
## 35  1  0  y4 25.9873  1  1
## 36  1  1  y4 32.1622  1  1
```

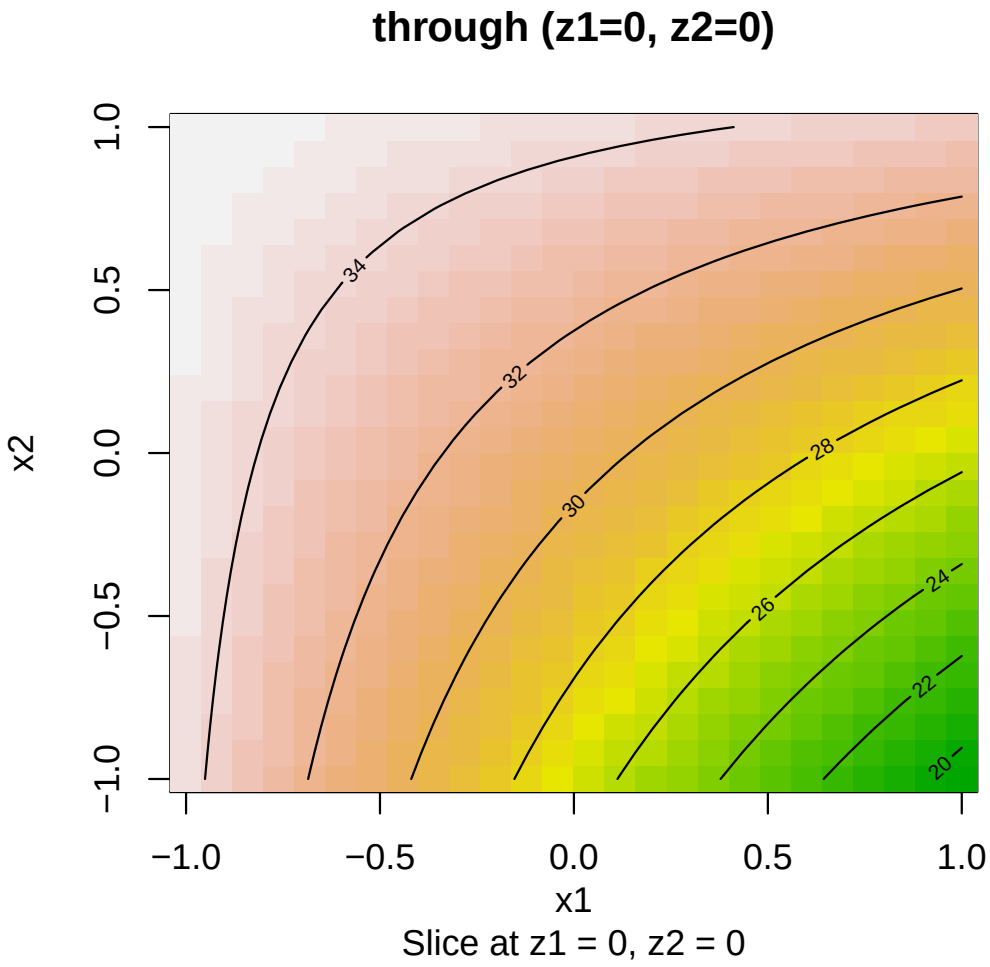
Model response using Equation 10.20 on p. 512.

```
library(rsm)
#rsm.10.5b.y.SPECx1x2z1z2 <- rsm(y ~ SO(x1, x2) + FO(z1, z2)
#                                + x1:z1 + x1:z2 + x2:z1 + x2:z2, data = df.10.5b.long)
lm.10.5b.y.SPECx1x2z1z2 <- lm(y ~ x1 + x2 + x1:x2 + x1:x1 + x2:x2
                                  + x1:x2 + x1:z1 + x1:z2 + x2:z1 + x2:z2
                              , data = df.10.5b.long)
# externally Studentized residuals
lm.10.5b.y.SPECx1x2z1z2$studres <- rstudent(lm.10.5b.y.SPECx1x2z1z2)
summary(lm.10.5b.y.SPECx1x2z1z2)
```

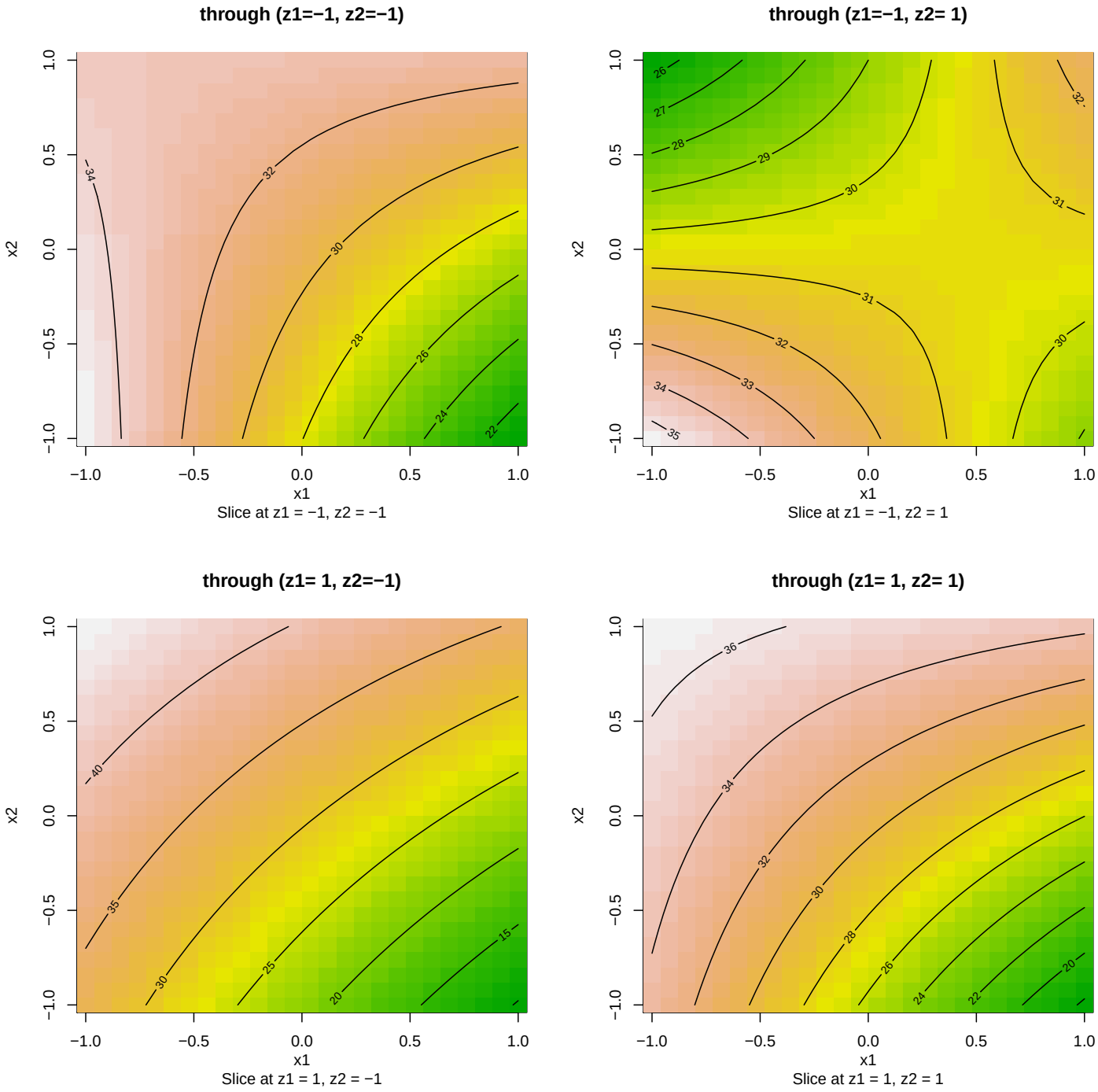
```
##
## Call:
## lm(formula = y ~ x1 + x2 + x1:x2 + x1:x1 + x2:x2 + x1:x2 + x1:z1 +
##     x1:z2 + x2:z1 + x2:z2, data = df.10.5b.long)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.931 -2.759 -0.415  4.504  9.635
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   30.592      0.998   30.66   <2e-16 ***
## x1            -4.175      1.222   -3.42   0.0020 **
## x2             3.748      1.222    3.07   0.0048 **
## x1:x2          3.348      1.496    2.24   0.0334 *
## x1:z1         -2.324      1.222   -1.90   0.0675 .
## x1:z2          1.932      1.222    1.58   0.1250
## x2:z1          3.268      1.222    2.67   0.0123 *
## x2:z2         -2.073      1.222   -1.70   0.1009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.99 on 28 degrees of freedom
```

```
## Multiple R-squared:  0.601,Adjusted R-squared:  0.502
## F-statistic: 6.03 on 7 and 28 DF,  p-value: 0.000237
```

```
par(mfrow = c(1,1))
contour(lm.10.5b.y.SPECx1x2z1z2, ~ x1 + x2, at = data.frame(z1 =  0, z2 =  0), image=TRUE, mai
```




```
par(mfrow = c(2,2))
contour(lm.10.5b.y.SPECx1x2z1z2, ~ x1 + x2, at = data.frame(z1 =  -1, z2 =  -1), image=TRUE, m
contour(lm.10.5b.y.SPECx1x2z1z2, ~ x1 + x2, at = data.frame(z1 =  -1, z2 =   1), image=TRUE, m
contour(lm.10.5b.y.SPECx1x2z1z2, ~ x1 + x2, at = data.frame(z1 =   1, z2 =  -1), image=TRUE, m
contour(lm.10.5b.y.SPECx1x2z1z2, ~ x1 + x2, at = data.frame(z1 =   1, z2 =   1), image=TRUE, m
```


through (z1=−1, z2=−1)
through (z1=−1, z2= 1)
through (z1= 1, z2=−1)
through (z1= 1, z2= 1)
x1
x2
−1.0
−0.5
0.0
0.5
1.0
Slice at z1 = −1, z2 = −1
Slice at z1 = −1, z2 = 1
Slice at z1 = 1, z2 = −1
Slice at z1 = 1, z2 = 1