Ch-05 & 06 R Codes

Ping-Yang Chen

2024-04-26

Textbook: Montgomery, D. C. (2012). *Design and analysis of experiments*, 8th Edition. John Wiley & Sons.

Online handouts: https://github.com/PingYangChen/ANOVA_Course_R_Code

Chapter 5: Factorial Experiments

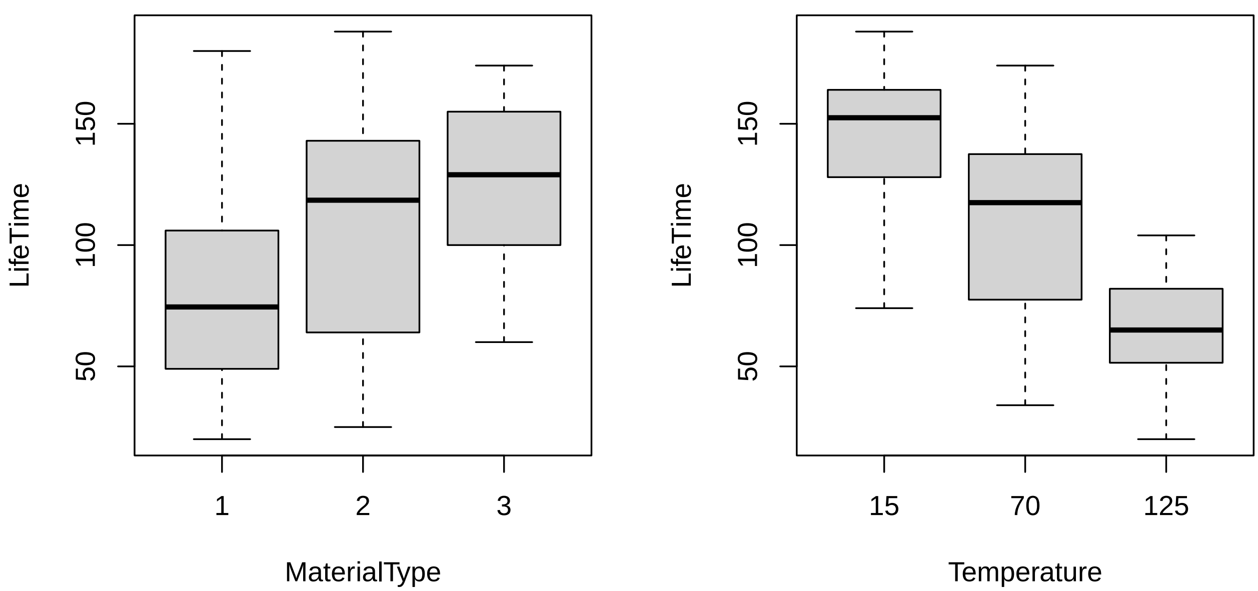
Example 5.1 The Battery Life Experiment

Read the csv file `5_BatteryLife.csv` in R. Make sure that in the `data.frame` the variables `MaterialType` and `Temperature` are the type of factor. If not sure, apply `as.factor()` on those variables after reading the dataset.

```
df1 <- read.csv(file.path("data", "5_BatteryLife.csv"))
df1$MaterialType <- as.factor(df1$MaterialType)
df1$Temperature <- as.factor(df1$Temperature)
```

Use boxplots to observe the differences of `LifeTime` among three levels of `MaterialType`, and, three levels of `Temperature`. We can observe that the average `LifeTime` tends to be lower for higher `Temperature`.

```
# Draw the grouped boxplot
par(mfrow = c(1, 2))
boxplot(LifeTime ~ MaterialType, data = df1)
boxplot(LifeTime ~ Temperature, data = df1)
```



The effect model

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk} \tag{1}$$

- τ_i is the effect of the ith `MaterialType` level, $i = 1, 2, 3$.
- β_j is the effect of the jth `Temperature` level, $j = 1, 2, 3$.
- $(\tau\beta)_{ij}$ is the interaction effect of the ith `MaterialType` level and the jth `Temperature` level.
- ε_{ijk} is the random error, $k = 1, 2, 3, 4$, satisfying

$$\varepsilon_{ijk} \overset{i.i.d.}{\sim} N(0, \sigma^2) \text{ where } \sigma^2 \text{ is the conatnt variance.}$$

Three statistical hypotheses of this problem are defined as

H_0: There is no effect on the choice of `MaterialType`.

H_0: There is no `Temperature` effect.

H_0: There is no interaction effect between `MaterialType` and `Temperature`

The function `aov()` fits the ANOVA model, and the ANOVA table is obtained by calling `summary()`. One the left-hand-side of the R model formula `Y ~ X`, input the name of the response variable, i.e. `LifeTime`. For factorial design, we test for the significance of the existence of the main effects as well as the the existence of the interaction effects. In R model formula, the syntax of the interaction term is `X1:X2`. In this battery life experiment, there are two factors, and hence the ANOVA model considers two main effects and one two-factor interaction. On the right-hand-side of the R model formula, the following two inputs are the same:

- Separately input main effects and two-factor interaction, `MaterialType + Temperature + MaterialType:Temperature`,
- Use multiplication `*` to include all interaction terms of the variables in the formula, `MaterialType * Temperature`.

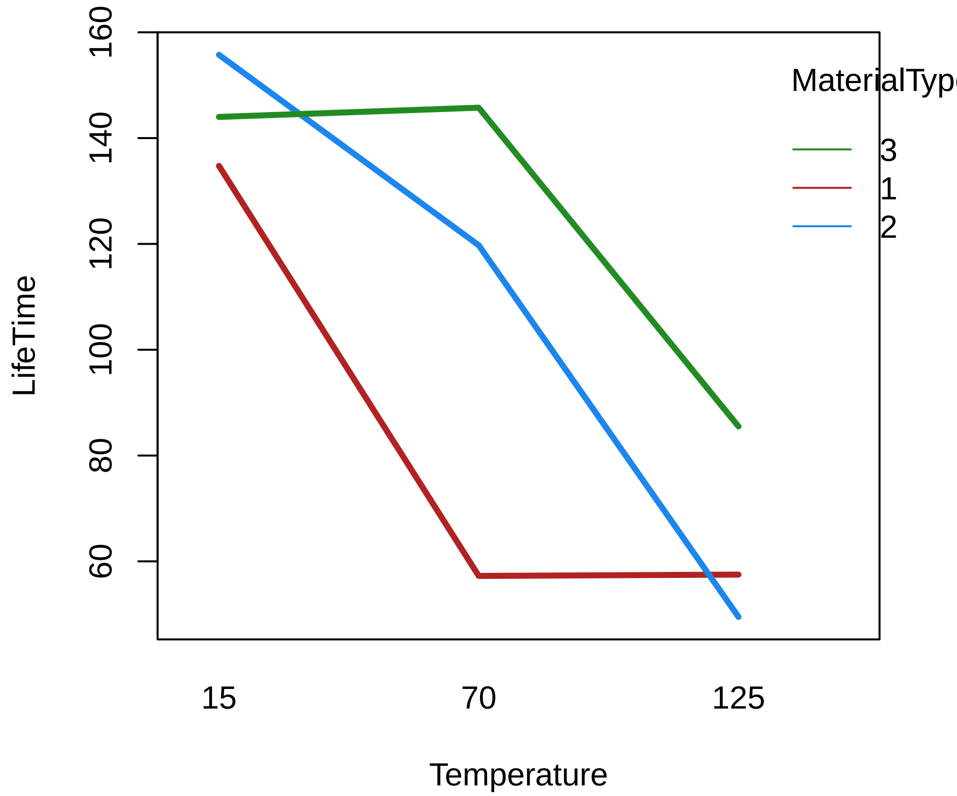
```
fit1 <- aov(LifeTime ~ MaterialType * Temperature, data = df1)
summary(fit1)
```

```
##                          Df Sum Sq Mean Sq F value   Pr(>F)
## MaterialType              2  10684    5342   7.911 0.00198 **
## Temperature               2  39119   19559  28.968 1.91e-07 ***
## MaterialType:Temperature  4   9614    2403   3.560 0.01861 *
## Residuals                27  18231     675
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-values of both main effects and the interaction are less than the pre-specified significant level 0.05. That is, `MaterialType` and `Temperature` are both significantly related to the battery's `LifeTime`, and, the interaction of `MaterialType` and `Temperature` is also significant.

To visualize the analysis result of the factorial experiment, interaction plot is commonly used tool.

```
interaction.plot(
  x.factor = df1$Temperature, # x-axis variable
  trace.factor = df1$MaterialType, # variable for lines
  response = df1$LifeTime, # y-axis variable
  ylab = "LifeTime", xlab = "Temperature",
  col = c("firebrick", "dodgerblue2", "forestgreen"),
  lty = 1, lwd = 3, trace.label = "MaterialType"
)
```

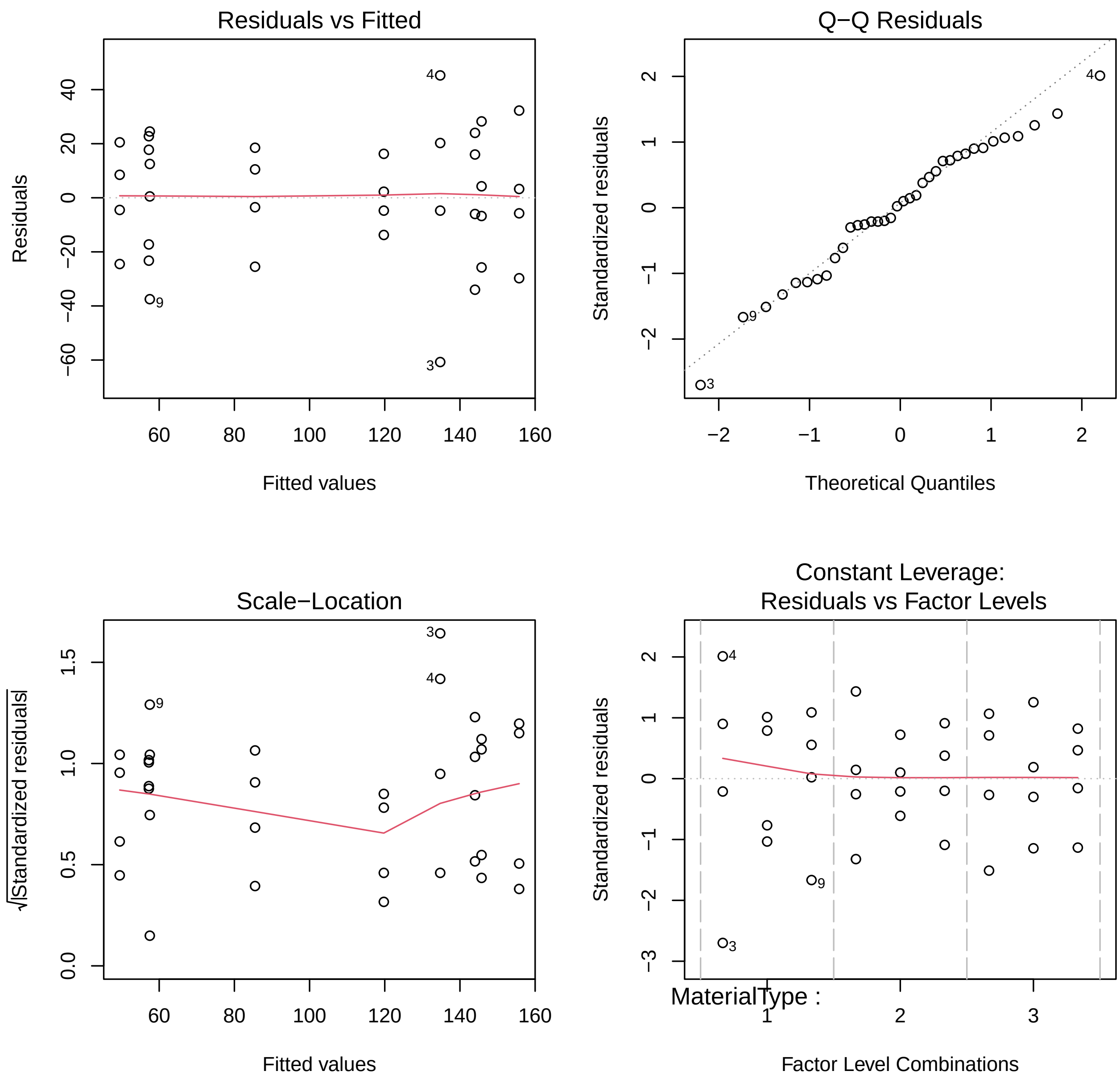


The plot shows two important conclusions:

- AT low temperature, $15°C$, the lifetime of the battery is generally longer than those battery's in $125°C$ environment. Among all materials, the life time of battery of type 2 material is the longest.
- AT middle temperature, $70°C$, the lifetime of the battery of type 3 material is the longest.

The procedure of diagnosing the residual is similar to that for the one-way ANOVA model. Please refer to the handout of R codes in Chapter 3 for more details of interpreting the residual plots.

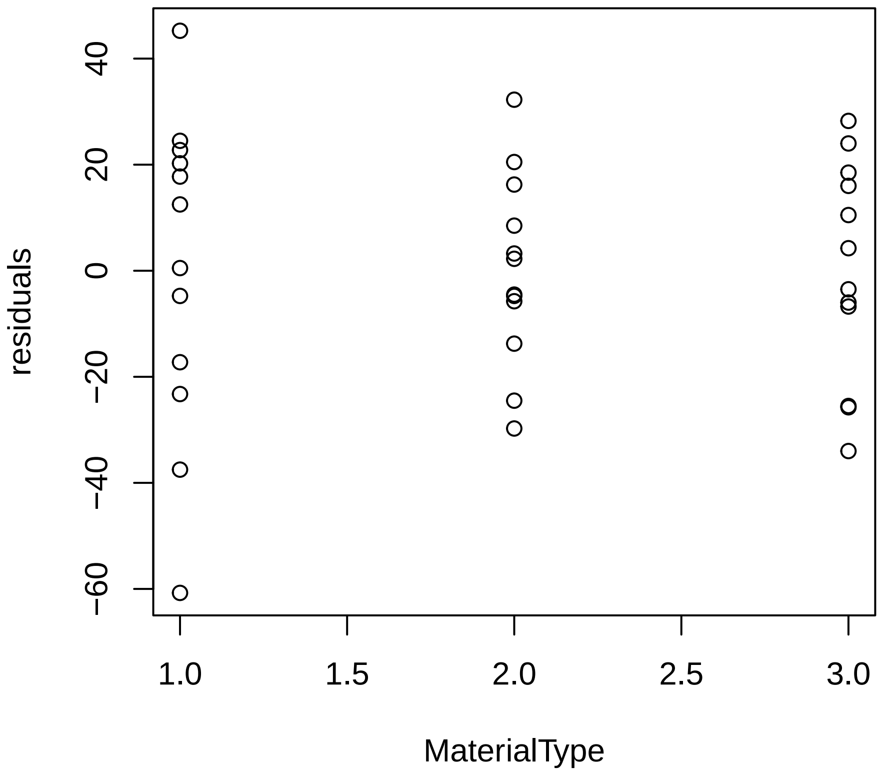
```
par(mfrow = c(2, 2))
plot(fit1)
```



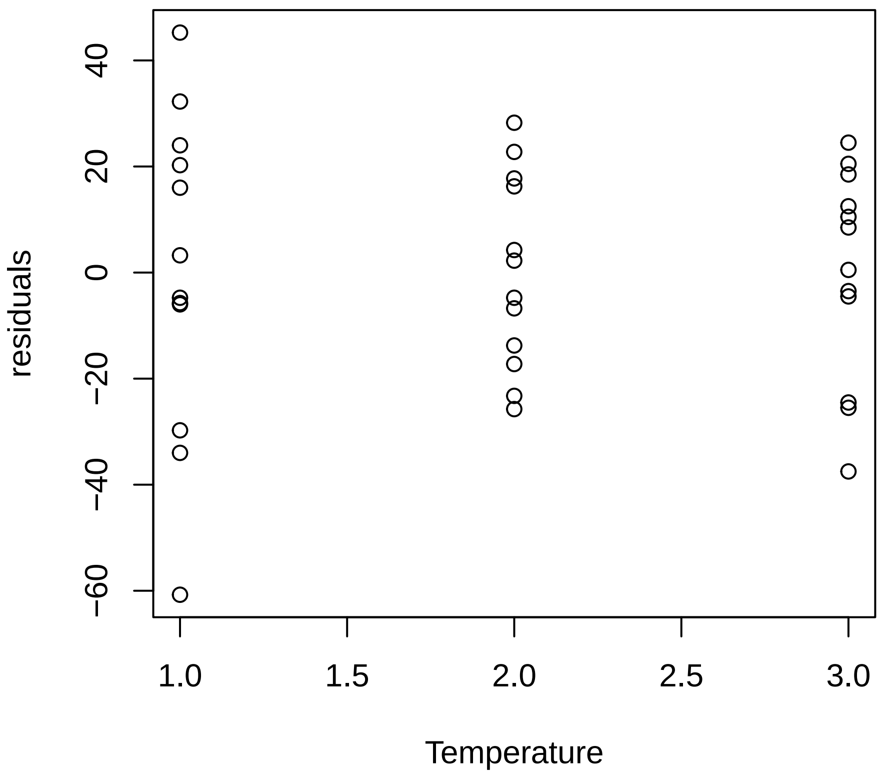

```
# par(mfrow = c(1, 1))
```

One additional plot is to draw the scatter plot of the residual against the levels of each factor. A lack of any visually obvious pattern in the dots on the plot is desired.

```
plot(
  as.numeric(df1$MaterialType), fit1$residuals,
  xlab = "MaterialType", ylab = "residuals"
)
```




```r
plot(
  as.numeric(df1$Temperature), fit1$residuals,
  xlab = "Temperature", ylab = "residuals"
)
```



Multiple comparison is performed for the treatment effect. The following codes demonstrate the use of Tukey's test and Fisher's LSD method.

For Tukey's test, add the input argument `which = c("MaterialType", "Temperature")` to only show the test results of comparing differences among the `ExtPressure` levels.

For Fisher's LSD method, specify `trt = c("MaterialType", "Temperature")` as the input argument to the `LSD.test()` function to show the comparison results among the `ExtPressure` levels. For information of interpreting the results, please refer to the handout of R codes in Chapter 3.

```
TukeyHSD(fit1, which = c("MaterialType", "Temperature"))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = LifeTime ~ MaterialType * Temperature, data = df1)
##
## $MaterialType
##         diff       lwr      upr     p adj
## 2-1 25.16667 -1.135677 51.46901 0.0627571
## 3-1 41.91667 15.614323 68.21901 0.0014162
## 3-2 16.75000 -9.552344 43.05234 0.2717815
##
## $Temperature
##             diff        lwr       upr     p adj
## 70-15  -37.25000  -63.55234 -10.94766 0.0043788
## 125-15 -80.66667 -106.96901 -54.36432 0.0000001
## 125-70 -43.41667  -69.71901 -17.11432 0.0009787
```

```
if (!("agricolae" %in% rownames(installed.packages()))) {
  install.packages("agricolae")
}
library(agricolae)
out <- LSD.test(fit1, trt = c("MaterialType", "Temperature"), p.adj = "bonferroni")
out$group
```

```
##       LifeTime groups
## 2:15    155.75      a
## 3:70    145.75     ab
## 3:15    144.00     ab
## 1:15    134.75     ab
## 2:70    119.75    abc
## 3:125    85.50    bcd
## 1:125    57.50     cd
## 1:70     57.25     cd
## 2:125    49.50      d
```

To fit the response surface model (RSM), the quantitative factor `Temperature` should be changed as of numeric type.

```
df1q <- read.csv(file.path("data", "5_BatteryLife.csv"))
df1q$MaterialType <- as.factor(df1q$MaterialType)
# Set MaterialType's dummy variable to use values -1, 0, 1
```

```r
contrasts(df1q$MaterialType) <- contr.sum(3)
# Check the result of the model matrix of main effects
model.matrix( ~ MaterialType + Temperature, data = df1q)
```

```
##    (Intercept) MaterialType1 MaterialType2 Temperature
## 1            1             1             0          15
## 2            1             1             0          15
## 3            1             1             0          15
## 4            1             1             0          15
## 5            1             1             0          70
## 6            1             1             0          70
## 7            1             1             0          70
## 8            1             1             0          70
## 9            1             1             0         125
## 10           1             1             0         125
## 11           1             1             0         125
## 12           1             1             0         125
## 13           1             0             1          15
## 14           1             0             1          15
## 15           1             0             1          15
## 16           1             0             1          15
## 17           1             0             1          70
## 18           1             0             1          70
## 19           1             0             1          70
## 20           1             0             1          70
## 21           1             0             1         125
## 22           1             0             1         125
## 23           1             0             1         125
## 24           1             0             1         125
## 25           1            -1            -1          15
## 26           1            -1            -1          15
## 27           1            -1            -1          15
## 28           1            -1            -1          15
## 29           1            -1            -1          70
## 30           1            -1            -1          70
## 31           1            -1            -1          70
## 32           1            -1            -1          70
## 33           1            -1            -1         125
## 34           1            -1            -1         125
## 35           1            -1            -1         125
## 36           1            -1            -1         125
## attr(,"assign")
## [1] 0 1 1 2
## attr(,"contrasts")
## attr(,"contrasts")$MaterialType
##   [,1] [,2]
## 1    1    0
## 2    0    1
## 3   -1   -1
```

The `lm()` function is used to fit the response surface model.

$$
\begin{aligned}
y = \beta_0 &+ \beta_{1a}x_{1a} + \beta_{1b}x_{1b} + \beta_2 x_2 + \beta_{22}x_2^2 \\
&+ \beta_{1a2}x_{1a}x_2 + \beta_{1b2}x_{1b}x_2 + \beta_{1a22}x_{1a}x_2^2 + \beta_{1b22}x_{1b}x_2^2 \\
&+ \beta_{222}x_2^3 + \varepsilon
\end{aligned}
$$

where $x_{1\cdot}$ and x_2 are the value of `MaterialType` and `Temperature` respectively, and β's are model coefficients.

In R model formula, the syntax indicating the higher order of the explanatory variable is `I(X^p)` where p is the power. The RSM is

```
ols1 <- lm(
  LifeTime ~ (Temperature + I(Temperature^2)) * MaterialType + I(Temperature^3),
  data = df1q
)
```

The `summary()` function for `lm` object is used to show the estimate of the coefficients and their significances. The coefficient estimate of $I(Temperature^3)$ is `NA` value because this cubic effect is aliased to the main effect.

```
summary(ols1)
```

```
##
## Call:
## lm(formula = LifeTime ~ (Temperature + I(Temperature^2)) * MaterialType +
##     I(Temperature^3), data = df1q)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -60.750 -14.625   1.375  17.938  45.250
##
## Coefficients: (1 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  153.922176  11.874742  12.962 4.17e-13 ***
## Temperature                   -0.590634   0.435985  -1.355  0.18674
## I(Temperature^2)              -0.001019   0.003037  -0.336  0.73975
## MaterialType1                 15.457989  16.793421   0.920  0.36547
## MaterialType2                  5.701791  16.793421   0.340  0.73684
## I(Temperature^3)                     NA         NA      NA       NA
## Temperature:MaterialType1     -1.910813   0.616576  -3.099  0.00450 **
## Temperature:MaterialType2      0.417287   0.616576   0.677  0.50430
## I(Temperature^2):MaterialType1   0.013871   0.004295   3.229  0.00325 **
## I(Temperature^2):MaterialType2  -0.004642   0.004295  -1.081  0.28936
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.98 on 27 degrees of freedom
## Multiple R-squared:  0.7652, Adjusted R-squared:  0.6956
## F-statistic:    11 on 8 and 27 DF,  p-value: 9.426e-07
```

The Result of the RSM:

For Material Type 1

$$\text{Life} = +169.380 - 2.489 \times \text{Temp} + 0.0129 \times \text{Temp}^2$$

For Material Type 2

$$\text{Life} = +159.624 - 0.179 \times \text{Temp} + 0.4163 \times \text{Temp}^2$$

For Material Type 3

$$\text{Life} = +132.762 + 0.893 \times \text{Temp} - 0.4322 \times \text{Temp}^2$$

Example 5.2 Tool Life Experiment (Two Quantitative Factors)

Read the csv file `5_ToolLife.csv` in R. Make variables `TotalAngle` and `CuttingSpeed` to be the type of factor.

```
df2 <- read.csv(file.path("data", "5_ToolLife.csv"))
df2$TotalAngle <- as.factor(df2$TotalAngle)
df2$CuttingSpeed <- as.factor(df2$CuttingSpeed)
```

The effect model

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk} \tag{2}$$

- τ_i is the effect of the ith `TotalAngle` level, $i = 1, 2, 3$.
- β_j is the effect of the jth `CuttingSpeed` level, $j = 1, 2, 3$.
- $(\tau\beta)_{ij}$ is the interaction effect of the ith `TotalAngle` level and the jth `CuttingSpeed` level.
- ε_{ijk} is the random error, $k = 1, 2$, satisfying

$$\varepsilon_{ijk} \overset{i.i.d.}{\sim} N(0, \sigma^2) \text{ where } \sigma^2 \text{ is the conatnt variance.}$$

Three statistical hypotheses of this problem are defined as

H_0: There is no `TotalAngle` effect.

H_0: There is no `CuttingSpeed` effect.

H_0: There is no interaction effect between `TotalAngle` and `CuttingSpeed`

Fit the ANOVA model by `aov()` function, and then print the ANOVA table by calling `summary()`.

```
fit2 <- aov(ToolLife ~ TotalAngle * CuttingSpeed, data = df2)
summary(fit2)
```

```
##                         Df Sum Sq Mean Sq F value  Pr(>F)
## TotalAngle               2  24.33  12.167   8.423 0.00868 **
## CuttingSpeed             2  25.33  12.667   8.769 0.00770 **
## TotalAngle:CuttingSpeed  4  61.33  15.333  10.615 0.00184 **
## Residuals                9  13.00   1.444
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-values of both main effects and the interaction are less than the pre-specified significant level 0.05. That is, `TotalAngle` and `CuttingSpeed` are both significantly related to the battery's `ToolLife`, and, the interaction of `TotalAngle` and `CuttingSpeed` is also significant.

Hereafter, the residual checking and multiple comparison processes are left for practice.

The `lm()` function is used to fit the response model.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \varepsilon$$

where x_1 and x_2 are the value of `TotalAngle` and `CuttingSpeed` respectively, and β's are model coefficients.

```r
df2q <- read.csv(file.path("data", "5_ToolLife.csv"))
x1m <- mean(df2q$TotalAngle)
x2m <- mean(df2q$CuttingSpeed)
# Centralize the variables
df2q$TotalAngle <- df2q$TotalAngle - x1m
df2q$CuttingSpeed <- df2q$CuttingSpeed - x2m
ols2 <- lm(
  ToolLife ~ TotalAngle*CuttingSpeed + I(TotalAngle^2) + I(CuttingSpeed^2),
  data = df2q
)
summary(ols2)
```

```
##
## Call:
## lm(formula = ToolLife ~ TotalAngle * CuttingSpeed + I(TotalAngle^2) +
##     I(CuttingSpeed^2), data = df2q)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5000 -1.3750 -0.0833  1.1250  3.8333
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.333333   1.239150   2.690   0.0197 *
## TotalAngle               0.166667   0.135742   1.228   0.2431
## CuttingSpeed             0.053333   0.027148   1.965   0.0731 .
## I(TotalAngle^2)         -0.080000   0.047022  -1.701   0.1146
## I(CuttingSpeed^2)       -0.001600   0.001881  -0.851   0.4116
## TotalAngle:CuttingSpeed -0.008000   0.006650  -1.203   0.2522
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.351 on 12 degrees of freedom
## Multiple R-squared:  0.4651, Adjusted R-squared:  0.2422
## F-statistic: 2.086 on 5 and 12 DF,  p-value: 0.1377
```

Another choice of the response surface model for two factors is to include all possible interactions of all the second-order terms

$$
\begin{aligned}
y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 \\
+ \beta_{112} x_1^2 x_2 + \beta_{122} x_1 x_2^2 + \beta_{1122} x_1^2 x_2^2 + \varepsilon
\end{aligned}
$$

```r
ols2_full <- lm(
  ToolLife ~ (TotalAngle + I(TotalAngle^2))*(CuttingSpeed + I(CuttingSpeed^2)),
  data = df2q
)
summary(ols2_full)
```

```
##
## Call:
```

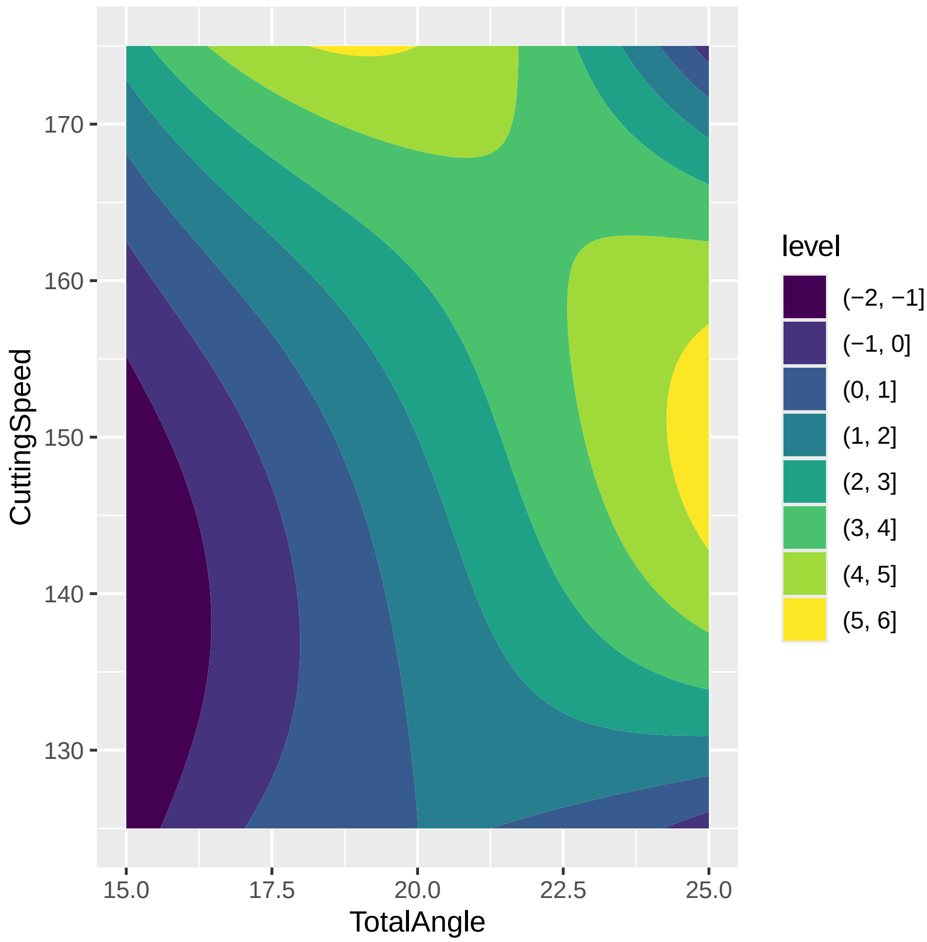
```
## lm(formula = ToolLife ~ (TotalAngle + I(TotalAngle^2)) * (CuttingSpeed +
##     I(CuttingSpeed^2)), data = df2q)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##   -1.5   -0.5    0.0    0.5    1.5
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         2.000e+00  8.498e-01   2.353 0.043065 *
## TotalAngle                          7.000e-01  1.202e-01   5.824 0.000252 ***
## I(TotalAngle^2)                     1.863e-17  4.163e-02   0.000 1.000000
## CuttingSpeed                        8.000e-02  2.404e-02   3.328 0.008824 **
## I(CuttingSpeed^2)                   1.600e-03  1.665e-03   0.961 0.361768
## TotalAngle:CuttingSpeed            -8.000e-03  3.399e-03  -2.353 0.043065 *
## TotalAngle:I(CuttingSpeed^2)       -1.280e-03  2.355e-04  -5.435 0.000414 ***
## I(TotalAngle^2):CuttingSpeed       -1.600e-03  1.178e-03  -1.359 0.207306
## I(TotalAngle^2):I(CuttingSpeed^2) -1.920e-04  8.158e-05  -2.353 0.043065 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.202 on 9 degrees of freedom
## Multiple R-squared:  0.8952, Adjusted R-squared:  0.802
## F-statistic: 9.606 on 8 and 9 DF,  p-value: 0.001337
```

Draw the contour plot of the RSM. From the plot, we can conclude that setting mid-level of cutting speed and large angle could achieve higher tool life.

```r
x1_grid <- seq(min(df2q$TotalAngle), max(df2q$TotalAngle), length = 100)
x2_grid <- seq(min(df2q$CuttingSpeed), max(df2q$CuttingSpeed), length = 100)
newx <- data.frame(
  TotalAngle = rep(x1_grid, each = 100),
  CuttingSpeed = rep(x2_grid, time = 100)
)
rs <- predict(ols2_full, newx)

rsplot_data <- data.frame(newx, rs = rs)
rsplot_data$TotalAngle <- rsplot_data$TotalAngle + x1m
rsplot_data$CuttingSpeed <- rsplot_data$CuttingSpeed + x2m

library(ggplot2)
#- Add color to the contour plot
ggplot(rsplot_data) +
  geom_contour(aes(TotalAngle, CuttingSpeed, z = rs), colour = "white") +
  geom_contour_filled(aes(TotalAngle, CuttingSpeed, z = rs))
```



As a complementation, here are the codes for 3D plot of the RSM

```
library(plotly)
library(htmlwidgets)
rsplot_matrix <- matrix(rsplot_data$rs, 100, 100)
p <- plot_ly(z = rsplot_matrix, type = "surface") %>%
  layout(scene = list(
      xaxis = list(
        title = 'TotalAngle',
        ticktext = lapply(seq(0, 100, 20), function(i) {
          diff(range(rsplot_data$TotalAngle))*i/100 + min(rsplot_data$TotalAngle)
        }),
        tickvals = list(0, 20, 40, 60, 80, 100),
        tickmode = "array"
      ),
      yaxis = list(
        title = 'CuttingSpeed',
        ticktext = lapply(seq(0, 100, 20), function(i) {
          diff(range(rsplot_data$CuttingSpeed))*i/100 + min(rsplot_data$CuttingSpeed)
        }),
        tickvals = list(0, 20, 40, 60, 80, 100),
        tickmode = "array"
      ),
```

```r
      zaxis = list(title = 'hat(ToolLife)')))

htmlwidgets::saveWidget(as_widget(p), "plotly_rsm_ch5.html")
```

One Observation per Cell

Read the csv file `5_Impurity.csv` in R. Make variables `Temperature` and `Pressure` to be the type of factor.

```r
df3 <- read.csv(file.path("data", "5_Impurity.csv"))
df3$Temperature <- as.factor(df3$Temperature)
df3$Pressure <- as.factor(df3$Pressure)
```

The effect model

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk} \tag{3}$$

- τ_i is the effect of the ith `Temperature` level, $i = 1, 2, 3$.
- β_j is the effect of the jth `Pressure` level, $j = 1, 2, 3$.
- $(\tau\beta)_{ij}$ is the interaction effect of the ith `Temperature` level and the jth `Pressure` level.
- ε_{ijk} is the random error, $k = 1, 2$, satisfying

$$\varepsilon_{ijk} \overset{i.i.d.}{\sim} N(0, \sigma^2) \text{ where } \sigma^2 \text{ is the conatnt variance.}$$

Three statistical hypotheses of this problem are defined as

H_0: There is no `Temperature` effect.

H_0: There is no `Pressure` effect.

H_0: There is no interaction effect between `Temperature` and `Pressure`

Fit the ANOVA model by `aov()` function, and then print the ANOVA table by calling `summary()`.

```r
fit3 <- aov(Impurity ~ Temperature * Pressure, data = df3)
summary(fit3)
```

```
##                      Df Sum Sq Mean Sq
## Temperature           2  23.33   11.67
## Pressure              4  11.60    2.90
## Temperature:Pressure  8   2.00    0.25
```

(Important) Because there is no replicates for each treatment combination, the ANOVA table does not exist given that the error variance σ^2 cannot be estimated.

Thus, for no-replicate scenario, we can only test for the two main effects.

H_0: There is no `Temperature` effect.

H_0: There is no `Pressure` effect.

```r
fit3_m <- aov(Impurity ~ Temperature + Pressure, data = df3)
summary(fit3_m)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Temperature  2  23.33   11.67   46.67 3.88e-05 ***
## Pressure     4  11.60    2.90   11.60  0.00206 **
## Residuals    8   2.00    0.25
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Then the significance of the interaction effect is verified using Tukey's test of nonadditivity. To implement Tukey's test of nonadditivity, the function `nonadditivity()` in the **agricolae** package is used. The resulting ANOVA for nonadditivity is shown below.

```
library(agricolae)
naddtest <- nonadditivity(
  df3$Impurity, df3$Temperature, df3$Pressure,
  df = df.residual(fit3_m), MSerror = deviance(fit3_m)/df.residual(fit3_m)
)
```

```
##
## Tukey's test of nonadditivity
## df3$Impurity
##
## P : 2.666667
## Q : 72.17778
##
## Analysis of Variance Table
##
## Response: residual
##               Df  Sum Sq  Mean Sq F value Pr(>F)
## Nonadditivity  1 0.09852 0.098522  0.3627  0.566
## Residuals      7 1.90148 0.271640
```

```
naddtest$ANOVA
```

```
## Analysis of Variance Table
##
## Response: residual
##               Df  Sum Sq  Mean Sq F value Pr(>F)
## Nonadditivity  1 0.09852 0.098522  0.3627  0.566
## Residuals      7 1.90148 0.271640
```

The p-value of the nonadditivity is larger than the significance level 0.05 suggesting that there is no two-factor interaction of `Temperature` and `Pressure`. 05

Three-Factor Factorial Experiment

Read the csv file `5_Impurity.csv` in R. Make variables `Carbonation`, `Pressure` and `LineSpeed` to be the type of factor.

```
df4 <- read.csv(file.path("data", "5_SoftDrinkBottling.csv"))
df4$Carbonation <- as.factor(df4$Carbonation)
df4$Pressure <- as.factor(df4$Pressure)
df4$LineSpeed <- as.factor(df4$LineSpeed)
```

The effect model

$$y_{ijkl} = \mu + \tau_i + \beta_j + \gamma_k + (\tau\beta)_{ij} + (\tau\gamma)_{ik} + (\beta\gamma)_{jk} + (\tau\beta\gamma)_{ijk} + \varepsilon_{ijkl} \tag{4}$$

- τ_i is the effect of the ith `Carbonation` level, $i = 1, 2, 3$.
- β_j is the effect of the jth `Pressure` level, $j = 1, 2$.
- γ_k is the effect of the kth `LineSpeed` level, $j = 1, 2$.
- $(\tau\beta)_{ij}$ are two-factor interactions.
- $(\tau\beta)_{ij}$, $(\tau\gamma)_{ik}$ and $(\beta\gamma)_{jk}$ are two-factor interactions.
- $(\tau\beta\gamma)_{ijk}$ is the three-factor interatoion.
- ε_{ijkl} is the random error, $k = 1, 2$, satisfying

$$\varepsilon_{ijkl} \overset{i.i.d.}{\sim} N(0, \sigma^2) \text{ where } \sigma^2 \text{ is the conatnt variance.}$$

Totally, there are 7 statistical hypotheses of this problems Fit the ANOVA model by `aov()` function, and then print the ANOVA table by calling `summary()`.

```r
fit4 <- aov(FillHeightsDev ~ Carbonation * Pressure * LineSpeed, data = df4)
summary(fit4)
```

```
##                                Df Sum Sq Mean Sq F value   Pr(>F)    
## Carbonation                     2 252.75  126.38 178.412 1.19e-09 ***
## Pressure                        1  45.37   45.37  64.059 3.74e-06 ***
## LineSpeed                       1  22.04   22.04  31.118  0.00012 ***
## Carbonation:Pressure            2   5.25    2.62   3.706  0.05581 .  
## Carbonation:LineSpeed           2   0.58    0.29   0.412  0.67149    
## Pressure:LineSpeed              1   1.04    1.04   1.471  0.24859    
## Carbonation:Pressure:LineSpeed  2   1.08    0.54   0.765  0.48687    
## Residuals                      12   8.50    0.71                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interpretation of these results are left for practice.

We can further remove all the terms with large p-value and then fit a reduced ANOVA model.

$$y_{ijkl} = \mu + \tau_i + \beta_j + \gamma_k + (\tau\beta)_{ij} + \varepsilon_{ijkl} \tag{5}$$

```r
fit4_r <- aov(
  FillHeightsDev ~ Carbonation + Pressure + LineSpeed + Carbonation:Pressure,
  data = df4
)
summary(fit4_r)
```

```
##                      Df Sum Sq Mean Sq F value   Pr(>F)    
## Carbonation           2 252.75  126.38 191.677 2.18e-12 ***
## Pressure              1  45.37   45.37  68.822 2.22e-07 ***
## LineSpeed             1  22.04   22.04  33.431 2.21e-05 ***
## Carbonation:Pressure  2   5.25    2.62   3.981   0.0382 *  
## Residuals            17  11.21    0.66                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Chapter 6: Two-Level Factorial Designs

2^2-Design: Chemical Process Example

Read the csv file `6_ChemicalRecovery.csv` in R.

```
df5 <- read.csv(file.path("data", "6_ChemicalRecovery.csv"))
```

The estimation of factor effects are:

```
# Effect of factor A (reactant concentration)
mean(df5$Recovery[df5$ReactConc == 1] - df5$Recovery[df5$ReactConc == -1])
# or 2*mean(df5$Recovery*df5$ReactConc)

# Effect of factor A (catalyst amount)
mean(df5$Recovery[df5$CataAmo == 1] - df5$Recovery[df5$CataAmo == -1])

# Effect of two-factor interaction AB
twofi <- as.numeric(df5$ReactConc)*df5$CataAmo
mean(df5$Recovery[twofi == 1] - df5$Recovery[twofi == -1])
```

```
## Effect of factor A (reactant concentration): 8.3333

## Effect of factor A (catalyst amount): -5.0000

## Effect of two-factor interaction AB: 1.6667
```

Use `aov()` to fit the ANOVA model (model description in math is omitted). Here, if the columns of the factor in the `data.frame` are not of the `factor` type. We can also specify the type of the factors as `factor` in the R model formula.

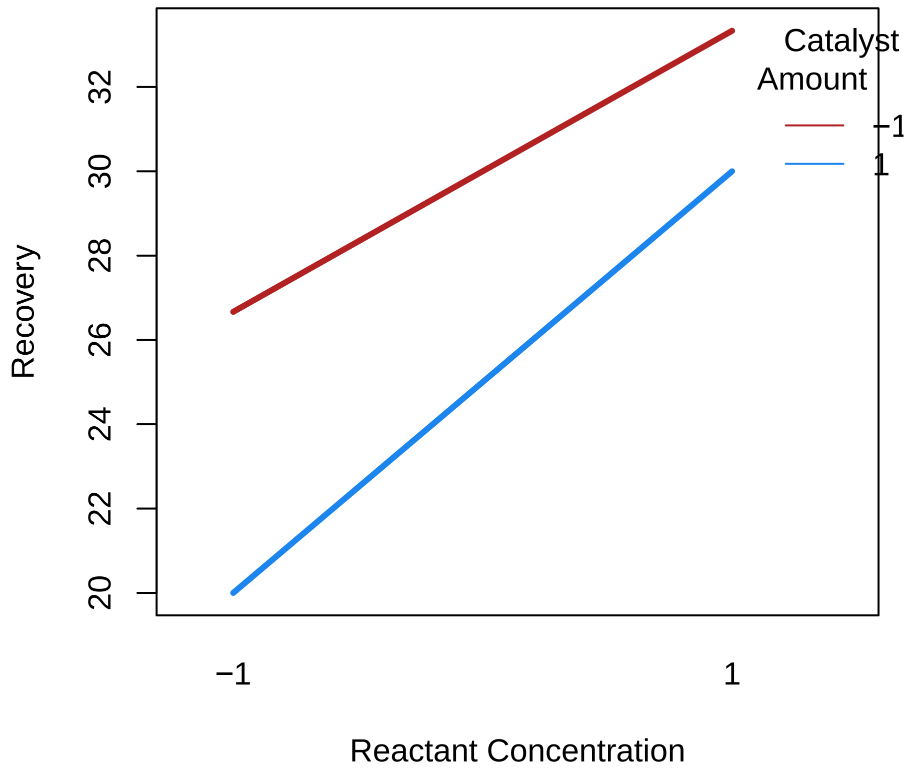
```
fit5 <- aov(Recovery ~ factor(ReactConc)*factor(CataAmo), data = df5)
summary(fit5)
```

```
##                                   Df Sum Sq Mean Sq F value   Pr(>F)    
## factor(ReactConc)                  1 208.33  208.33  53.191 8.44e-05 ***
## factor(CataAmo)                    1  75.00   75.00  19.149  0.00236 ** 
## factor(ReactConc):factor(CataAmo)  1   8.33    8.33   2.128  0.18278    
## Residuals                          8  31.33    3.92                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The two main effects are significant and their two-factor interaction may not exist.

To visualize the analysis result of the factorial experiment, interaction plot is commonly used tool.

```
interaction.plot(
  x.factor = df5$ReactConc, # x-axis variable
  trace.factor = df5$CataAmo, # variable for lines
  response = df5$Recovery, # y-axis variable
  ylab = "Recovery", xlab = "Reactant Concentration",
  col = c("firebrick", "dodgerblue2"),
  lty = 1, lwd = 3, trace.label = "Catalyst\nAmount"
)
```



2^3-Design: Plasma Etching Example

Read the csv file `6_PlasmaEtching_2^3.csv` in R.

```
df6 <- read.csv(file.path("data", "6_PlasmaEtching_2^3.csv"))
```

The estimation of factor effects are:

```
# Effect of main effects
2*mean(df6$EachRate*df6$Gap)
2*mean(df6$EachRate*df6$GasFlow)
2*mean(df6$EachRate*df6$Power)

# Effect of two-factor interactions
GapGasFlow <- df6$Gap * df6$GasFlow
GapPower <- df6$Gap * df6$Power
GasFlowPower <- df6$GasFlow * df6$Power
2*mean(df6$EachRate*GapGasFlow)
2*mean(df6$EachRate*GapPower)
2*mean(df6$EachRate*GasFlowPower)

# Effect of three-factor interaction
GapGasFlowPower <- df6$Gap * df6$GasFlow * df6$Power
2*mean(df6$EachRate*GapGasFlowPower)
```

```
##   Factor Est.Effect
## 1      A   -101.625
```

```
## 2      B      7.375
## 3      C    306.125
## 4     AB    -24.875
## 5     AC   -153.625
## 6     BC     -2.125
## 7    ABC      5.625
```

Use `aov()` to fit the ANOVA model (model description in math is omitted). Here, if the columns of the factor in the `data.frame` are not of the `factor` type. We can also specify the type of the factors as `factor` in the R model formula.

```r
fit6 <- aov(
  EachRate ~ factor(Gap) * factor(GasFlow) * factor(Power),
  data = df6
)
summary(fit6)
```

```
##                                           Df Sum Sq Mean Sq F value   Pr(>F)
## factor(Gap)                                1  41311   41311  18.339 0.002679
## factor(GasFlow)                            1    218     218   0.097 0.763911
## factor(Power)                              1 374850  374850 166.411 1.23e-06
## factor(Gap):factor(GasFlow)                1   2475    2475   1.099 0.325168
## factor(Gap):factor(Power)                  1  94403   94403  41.909 0.000193
## factor(GasFlow):factor(Power)              1     18      18   0.008 0.930849
## factor(Gap):factor(GasFlow):factor(Power)  1    127     127   0.056 0.818586
## Residuals                                  8  18020    2253
##
## factor(Gap)                               **
## factor(GasFlow)
## factor(Power)                             ***
## factor(Gap):factor(GasFlow)
## factor(Gap):factor(Power)                 ***
## factor(GasFlow):factor(Power)
## factor(Gap):factor(GasFlow):factor(Power)
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The significant effects are main effects `Gap` and `Power`, and, the two-factor interaction `Gap:Power`.

WE can also fit a response surface model and obtain the same conslusion.

```r
ols6_full <- lm(EachRate ~ Gap * GasFlow * Power, data = df6)
summary(ols6_full)
```

```
##
## Call:
## lm(formula = EachRate ~ Gap * GasFlow * Power, data = df6)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -65.50 -11.12   0.00  11.12  65.50
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         776.062     11.865  65.406 3.32e-12 ***
## Gap                 -50.812     11.865  -4.282 0.002679 **
## GasFlow               3.688     11.865   0.311 0.763911
## Power               153.062     11.865  12.900 1.23e-06 ***
## Gap:GasFlow         -12.438     11.865  -1.048 0.325168
## Gap:Power           -76.812     11.865  -6.474 0.000193 ***
## GasFlow:Power        -1.062     11.865  -0.090 0.930849
## Gap:GasFlow:Power     2.813     11.865   0.237 0.818586
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.46 on 8 degrees of freedom
## Multiple R-squared:  0.9661, Adjusted R-squared:  0.9364
## F-statistic: 32.56 on 7 and 8 DF,  p-value: 2.896e-05
```

The final RSM is the reduced model that those insignificant terms are removed from the full model. Now we can use the real value of the factot levels as the predictors' values fro this reduced model for better interpretation.

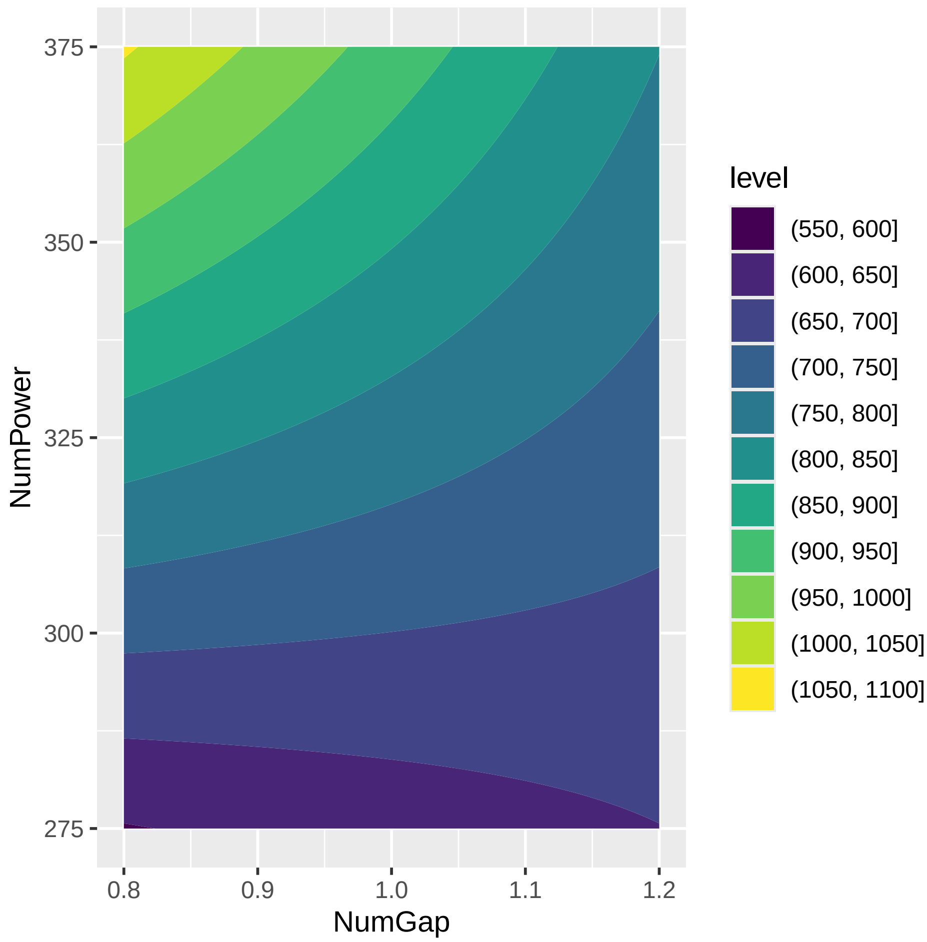
```r
ols6_reduced <- lm(EachRate ~ NumGap * NumPower, data = df6)
summary(ols6_reduced)
```

```
##
## Call:
## lm(formula = EachRate ~ NumGap * NumPower, data = df6)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -72.50 -15.44   2.50  18.69  66.50
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2461.188    349.513  -7.042 1.35e-05 ***
## NumGap           2242.344    342.725   6.543 2.76e-05 ***
## NumPower           10.743      1.063  10.107 3.19e-07 ***
## NumGap:NumPower    -7.681      1.042  -7.370 8.62e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.69 on 12 degrees of freedom
## Multiple R-squared:  0.9608, Adjusted R-squared:  0.9509
## F-statistic: 97.91 on 3 and 12 DF,  p-value: 1.054e-08
```

```r
x1_grid <- seq(min(df6$NumGap), max(df6$NumGap), length = 100)
x2_grid <- seq(min(df6$NumPower), max(df6$NumPower), length = 100)
newx <- data.frame(
  NumGap = rep(x1_grid, each = 100),
  NumPower = rep(x2_grid, time = 100)
)

rsplot_data6 <- data.frame(newx, rs = predict(ols6_reduced, newx))
```

```
library(ggplot2)
#- Add color to the contour plot
ggplot(rsplot_data6) +
  geom_contour(aes(NumGap, NumPower, z = rs), colour = "white") +
  geom_contour_filled(aes(NumGap, NumPower, z = rs))
```



Unreplicated 2^k Factorial Designs: The Resin Plant Experiment

Read the csv file `6_PilotPlant.csv` in R.

```
df7 <- read.csv(file.path("data", "6_PilotPlant.csv"))
```

The estimation of factor effects are:

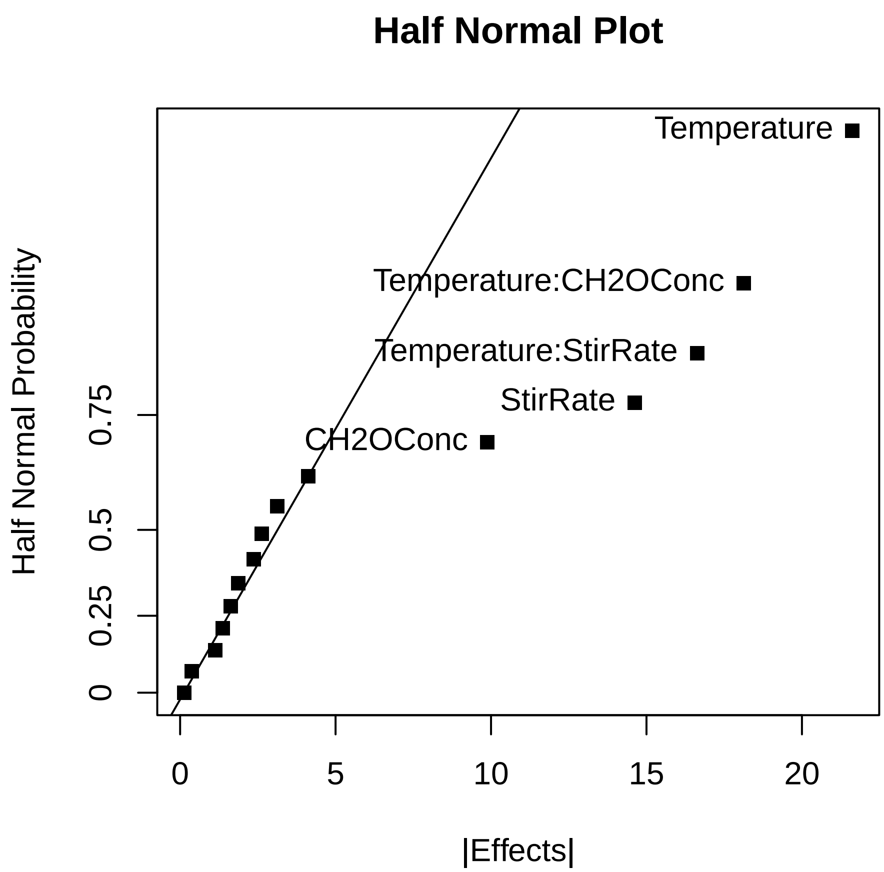
```
# Compute the model matrix of all effect terms without intercept
mmat7 <- model.matrix( ~ Temperature*Pressure*CH2OConc*StirRate - 1, data = df7)
# Calculate the effect sizes using the +/- signs of the model matrix
eff7 <- numeric(ncol(mmat7))
for (i in 1:ncol(mmat7)) {
  eff7[i] <- 2*mean(df7$FiltrationRate*mmat7[,i])
}
names(eff7) <- colnames(mmat7)
```

```
##                                     Factor Est.Effect
## 1                              Temperature     21.625
## 2                                 Pressure      3.125
## 3                                 CH2OConc      9.875
## 4                                 StirRate     14.625
## 5                     Temperature:Pressure      0.125
## 6                     Temperature:CH2OConc    -18.125
## 7                        Pressure:CH2OConc      2.375
## 8                     Temperature:StirRate     16.625
## 9                        Pressure:StirRate     -0.375
## 10                       CH2OConc:StirRate     -1.125
## 11           Temperature:Pressure:CH2OConc      1.875
## 12           Temperature:Pressure:StirRate      4.125
## 13           Temperature:CH2OConc:StirRate     -1.625
## 14              Pressure:CH2OConc:StirRate     -2.625
## 15 Temperature:Pressure:CH2OConc:StirRate      1.375
```

Because there is no replicates in each treatment combination, the estimate of the random error σ^2 does not exist and the ANOVA table is not available. Of all effect terms, we try of eliminate some of them before analyzing the data. According to the model assumption, the effects that are negligible should be similar to random error which is normally distributed with zero mean and constant variance. Therefore, a QQ-plot or half-Normal plot is helpful to identify effective effects. Belows are the codes of half-Normal plot.

```r
# Half Normal Plot
halfqqnorm <- function(input, tol = 0.5) {
    y <- sort(abs(input))
    nq <- qnorm(seq(0.5, 0.99, length = length(y)))
    plot(y, nq, yaxt = "n", pch = 15,
         xlab = "|Effects|", ylab = "Half Normal Probability")
    title("Half Normal Plot")
    # choose anchor point to draw a straight line
    s <- min(which(diff(y)/diff(range(y)) > 1/(length(y)-1)))
    abline(a = -y[1]*(nq[s]-nq[1])/(y[s]-y[1]), b = (nq[s]-nq[1])/(y[s]-y[1]))
    axis(2, at = qnorm(seq(0.5, 0.9999, length = 5)),
         labels = round(seq(0, 1, length = 5),2))
    loc <- sqrt((nq - (y - y[1])*(nq[s]-nq[1])/(y[s]-y[1]))^2) > tol
    if (is.null(names(y))) {
        text(y[loc], nq[loc], order(abs(input))[loc], pos = 2)
    } else {
        text(y[loc], nq[loc], names(abs(input))[order(abs(input))[loc]], pos = 2)
    }
}
```

```r
halfqqnorm(eff7)
```



By the half Normal plot, we find out that the main effects `Temperature`, `CH2OConc`and `StirRate` and interactions `Temperature:CH2OConc`, `Temperature:StirRate` appear to be large.

Based on the observation from the half Normal plot, now the ANOVA model is

$$y_{ijkl} = \mu + \tau_i + \beta_j + \gamma_k + (\tau\beta)_{ij} + (\tau\gamma)_{ik} + \varepsilon_{ijkl} \tag{6}$$

- τ_i is the effect of the ith `Temperature` level, $i = 1, 2$.
- β_j is the effect of the jth `CH2OConc` level, $j = 1, 2$.
- γ_k is the effect of the kth `StirRate` level, $j = 1, 2$.
- $(\tau\beta)_{ij}$ is the interaction effect of the ith `Temperature` level and the jth `CH2OConc` level.
- $(\tau\gamma)_{ij}$ is the interaction effect of the ith `Temperature` level and the kth `StirRate` level.
- ε_{ijkl} is the random error, $l = 1, 2$, satisfying

$$\varepsilon_{ijkl} \overset{i.i.d.}{\sim} N(0, \sigma^2) \text{ where } \sigma^2 \text{ is the conatnt variance.}$$

Use `aov()` to fit the ANOVA model.

```
fit7 <- aov(
  FiltrationRate ~ factor(Temperature) * (factor(CH2OConc) + factor(StirRate)),
  data = df7
)
summary(fit7)
```

```
##                                     Df Sum Sq Mean Sq F value    Pr(>F)
```

```
## factor(Temperature)                      1 1870.6  1870.6   95.86 1.93e-06 ***
## factor(CH2OConc)                         1  390.1   390.1   19.99   0.0012 **
## factor(StirRate)                         1  855.6   855.6   43.85 5.92e-05 ***
## factor(Temperature):factor(CH2OConc)     1 1314.1  1314.1   67.34 9.41e-06 ***
## factor(Temperature):factor(StirRate)     1 1105.6  1105.6   56.66 2.00e-05 ***
## Residuals                               10  195.1    19.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here, we left the interpretation of the response surface model for practice.

Addition of Center Points to a 2^k Designs

Recall the Resin Plant Experiment. Read the csv file `6_PilotPlant.csv` in R.

```
df7 <- read.csv(file.path("data", "6_PilotPlant.csv"))
```

```
df7_C <- data.frame(
  Temperature = rep(0, 4),
  Pressure = rep(0, 4),
  CH2OConc = rep(0, 4),
  StirRate = rep(0, 4),
  FiltrationRate = c(73, 75, 66, 69)
)
```

The calculation of $SS_{Purequadratic}$ with degree of freedom 1.

```
Yf_bar <- mean(df7$FiltrationRate)
Yc_bar <- mean(df7_C$FiltrationRate)
nf <- nrow(df7)
nc <- nrow(df7_C)
SS_pureQuad <- nf*nc*(Yf_bar - Yc_bar)^2/(nf + nc)
```

The calculation of SS_E with degree of freedom $n_c - 1$.

```
SS_E <- sum((df7_C$FiltrationRate - mean(df7_C$FiltrationRate))^2)
```

To test the significance of the Curvature, we compute the ratio of $MS_{Purequadratic}$ and MS_E as the F-statistic which follows the F distribution with degrees of freedom 1 and $n_c - 1$.

```
MS_pureQuad <- SS_pureQuad/1
MS_E <- SS_E/(nc - 1)
fval <- MS_pureQuad/MS_E
pval <- 1 - pf(fval, 1, nc - 1) # p-value
```

The part of the testing the significance of the Curvature of the ANOVA table.

```
print(data.frame(
  Source = c("Curvature", "Residual"),
  SS = c(SS_pureQuad, SS_E),
```

```r
  DF = c(1, nc - 1),
  MS = c(MS_pureQuad, MS_E),
  "F" = c(sprintf("%.3f", fval), ""),
  "Pr(>F)" = c(sprintf("%.3f", pval), "")
))
```

```
##      Source      SS DF      MS     F Pr..F.
## 1 Curvature  1.5125  1  1.5125 0.093  0.780
## 2  Residual 48.7500  3 16.2500
```