

Ch-05 R Codes

Ping-Yang Chen

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

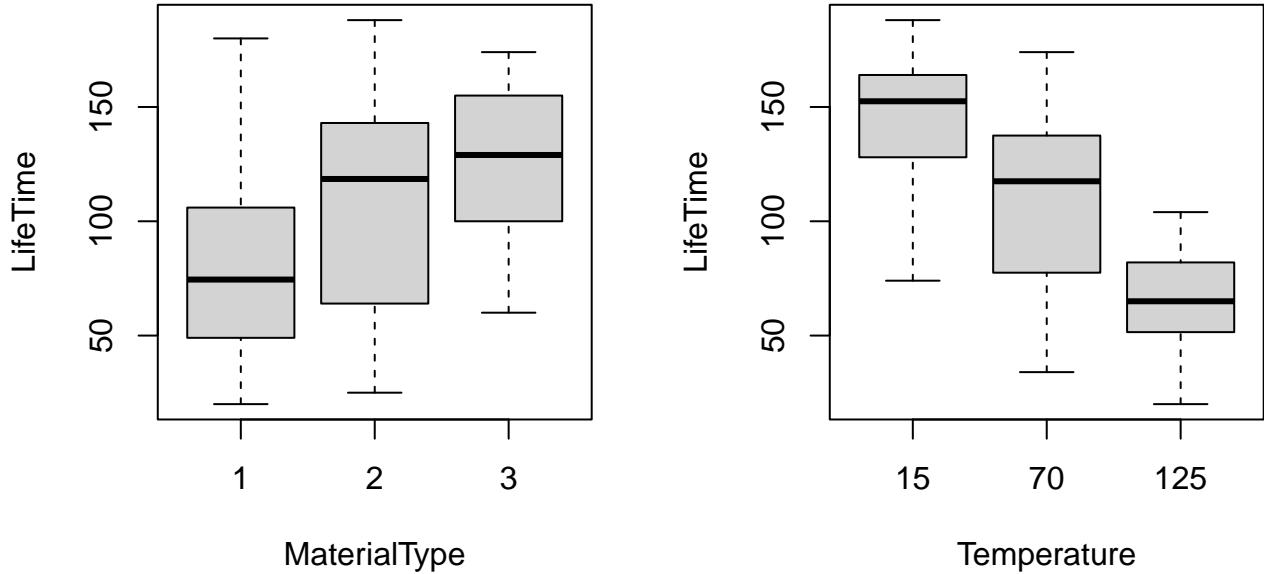
Example 5.1 The Battery Life Experiment

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`. The boxplots show that `MaterialType` affects `LifeTime` that materials 2 and 3 have higher median `LifeTime` than material 1, indicating better durability. For `Temperature`, `LifeTime` clearly decreases as temperature increases. The battery last the longest at $15^{\circ}C$ and the shortest at $125^{\circ}C$.

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



To analyze this battery lifetime data, we first establish the effect model

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

where

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

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

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()`. On 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 identical:

- 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

```

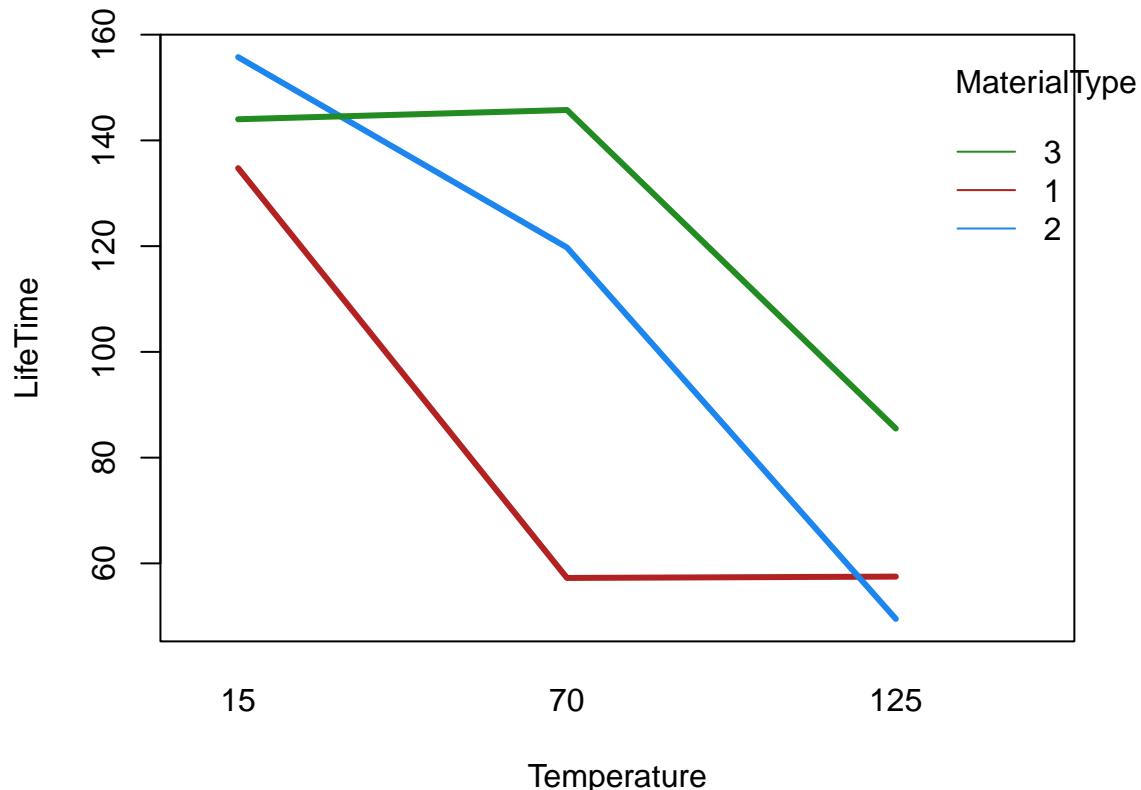
The p-values of both main effects and the interaction are less than the pre-specified significant level 0.05. That is, `MaterialType` and `Temperature` are both significantly related to the battery's `LifeTime`, and, the interaction of `MaterialType` and `Temperature` is also significant.

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"
)

```

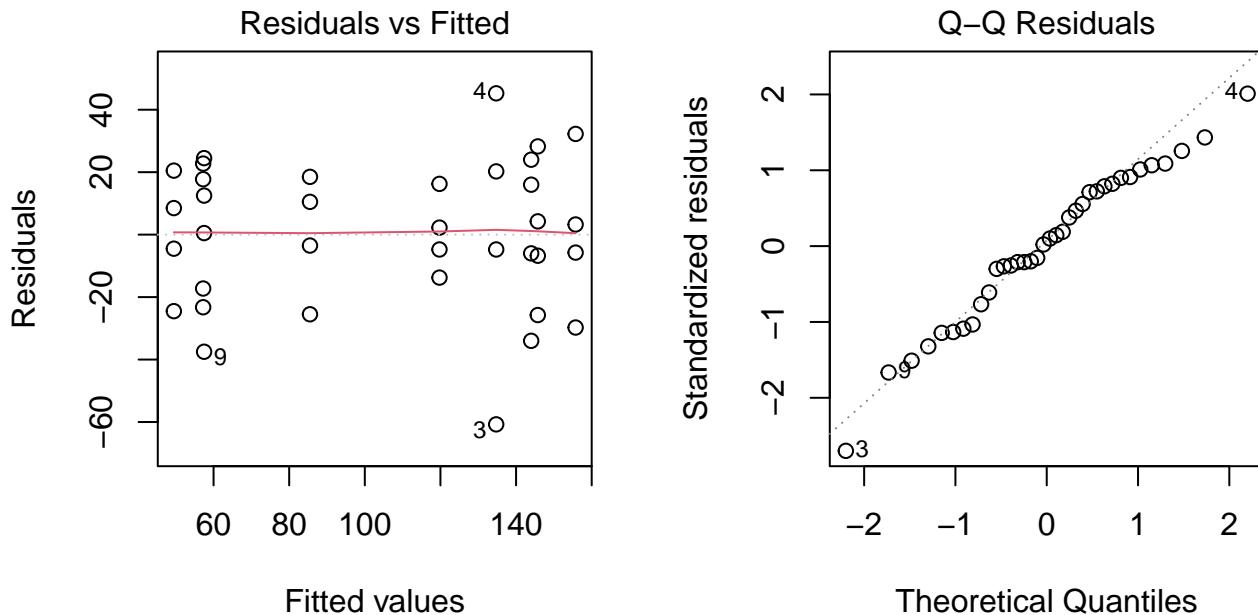


The plot shows two important conclusions:

- At low temperature, $15^{\circ}C$, the lifetime of the battery is generally longer than those battery's in $125^{\circ}C$ environment. Among all materials, the life time of battery of type 2 material is the longest.
- At middle temperature, $70^{\circ}C$, 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.

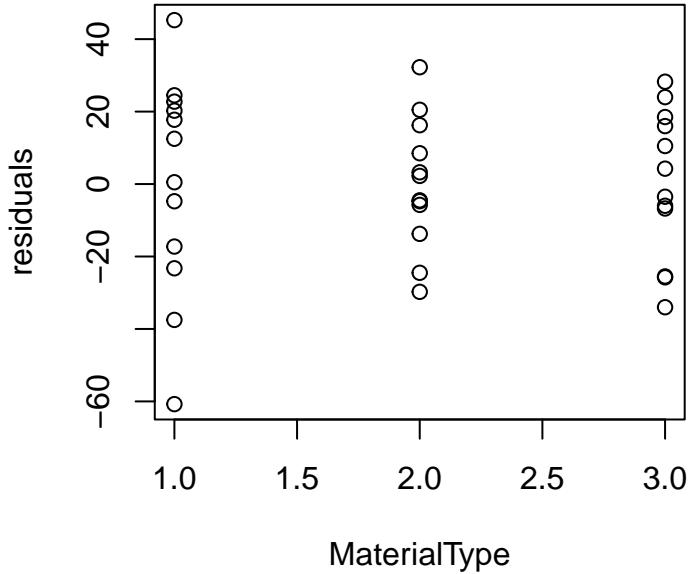
```
par(mfrow = c(1, 2))
plot(fit1, which = 1:2)
```



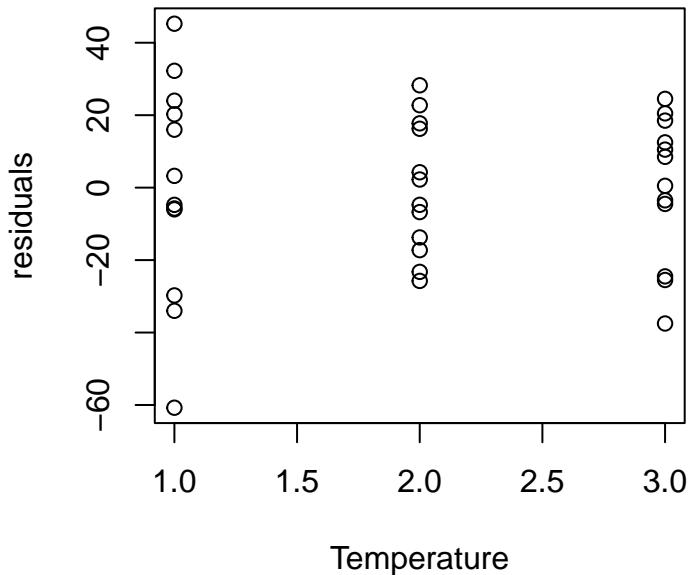
```
# par(mfrow = c(1, 1))
```

There are additional scatter plots showing 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"
)
```



```
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 show the test results of comparing differences among the `MaterialType` levels and `Temperature` 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 `MaterialType` levels and `Temperature` 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
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.

$$\begin{aligned}
y = & \beta_0 + \beta_{1a}x_{1a} + \beta_{1b}x_{1b} + \beta_2x_2 + \beta_{22}x_2^2 \\
& + \beta_{1a2}x_{1a}x_2 + \beta_{1b2}x_{1b}x_2 + \beta_{1a22}x_{1a}x_2^2 + \beta_{1b22}x_{1b}x_2^2 \\
& + \beta_{222}x_2^3 + \varepsilon
\end{aligned}$$

where x_1 . and x_2 are the value of `MaterialType` and `Temperature` respectively, and those 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 significance. 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

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

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

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

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.

```
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
# Fit the response surface
```

```

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

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

```

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.

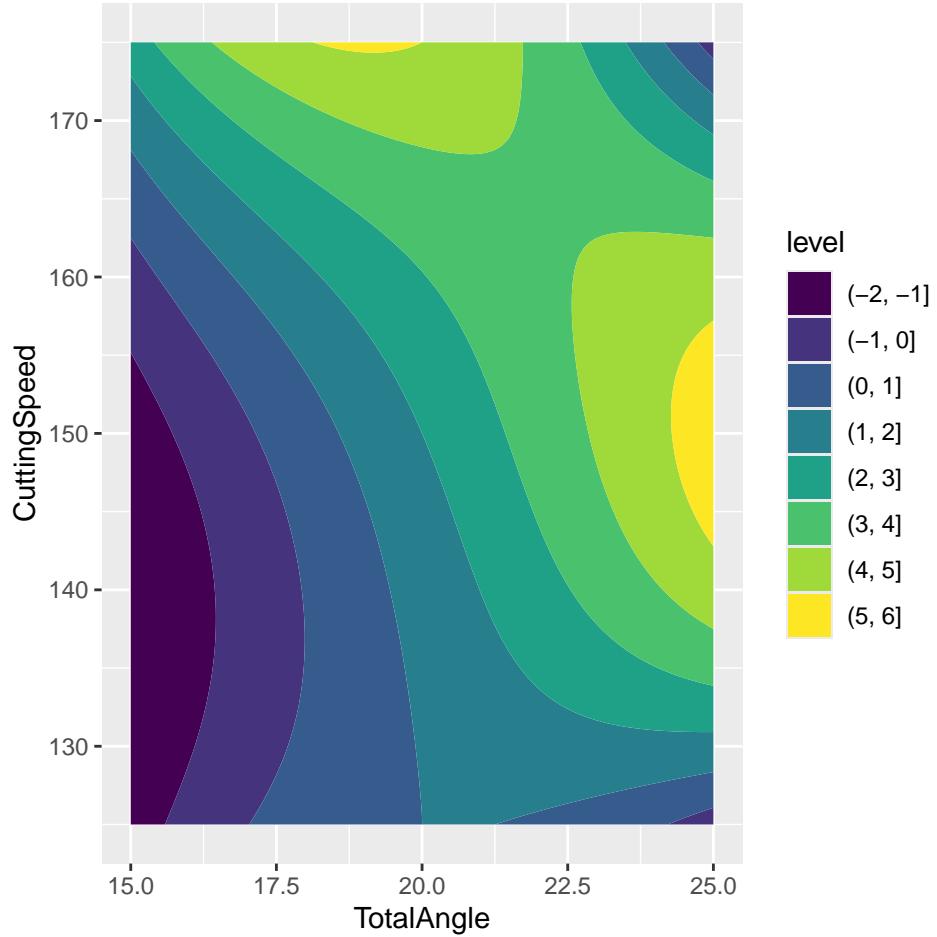
```

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))

```



We can also draw the interacting 3D plot of the RSM for better visualization.

```
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"
    )
  ),
```

```

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.

```

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

```

The effect model

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

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

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

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()`.

```

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.

```

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

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

- τ_i is the effect of the i th Carbonation level, $i = 1, 2, 3$.
- β_j is the effect of the j th Pressure level, $j = 1, 2$.
- γ_k is the effect of the k th 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 interaction.
- ε_{ijkl} is the random error, $k = 1, 2$, satisfying

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

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

```
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.

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

```
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