Two-Factor Design: Estimating Model Parameters

Recall the model:

$$y_{i,j,k} = \mu + \tau_i + \beta_j + (\tau\beta)_{i,j} + \epsilon_{i,j,k}$$

For balanced data, with the *natural* constraints:

$$\begin{aligned} \mathrm{E}\left(\bar{y}_{\cdots}\right) &= \mu \\ \mathrm{E}\left(\bar{y}_{i\cdot\cdot}\right) &= \mu + \tau_i \\ \mathrm{E}\left(\bar{y}_{\cdot j\cdot}\right) &= \mu + \beta_j \\ \mathrm{E}\left(\bar{y}_{i,j\cdot}\right) &= \mu + \tau_i + \beta_j + (\tau\beta)_{i,j} \end{aligned}$$

So the natural parameter estimates are

$$
\begin{aligned}
\hat{\mu} &= \bar{y}_{\cdots} \\
\hat{\tau}_i &= \bar{y}_{i\cdot\cdot} - \hat{\mu} \\
&= \bar{y}_{i\cdot\cdot} - \bar{y}_{\cdots} \\
\hat{\beta}_j &= \bar{y}_{\cdot j\cdot} - \hat{\mu} \\
&= \bar{y}_{\cdot j\cdot} - \bar{y}_{\cdots} \\
\widehat{(\tau\beta)}_{i,j} &= \bar{y}_{i,j\cdot} - \left(\hat{\mu} + \hat{\tau}_i + \hat{\beta}_j\right) \\
&= \bar{y}_{i,j\cdot} - \bar{y}_{i\cdot\cdot} - \bar{y}_{\cdot j\cdot} + \bar{y}_{\cdots}
\end{aligned}
$$

These are also least squares estimates.

Standard packages use the "reference level" constraints.

E.g., the battery life data

R commands

```
# for more compact output:
batteryLife$M <- factor(batteryLife$Material)
batteryLife$T <- factor(batteryLife$Temperature)
summary(lm(Life ~ M * T, batteryLife))
```

Output

```
Call:
lm(formula = Life ~ M * T, data = batteryLife)

Residuals:
    Min      1Q  Median      3Q     Max
-60.750 -14.625   1.375  17.938  45.250
```

Output, continued

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   134.75      12.99  10.371 6.46e-11 ***
M2             21.00      18.37   1.143 0.263107
M3              9.25      18.37   0.503 0.618747
T70           -77.50      18.37  -4.218 0.000248 ***
T125          -77.25      18.37  -4.204 0.000257 ***
M2:T70         41.50      25.98   1.597 0.121886
M3:T70         79.25      25.98   3.050 0.005083 **
M2:T125       -29.00      25.98  -1.116 0.274242
M3:T125        18.75      25.98   0.722 0.476759
---
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
```

R note

For convenience, you can get the same output using:

```
with(batteryLife, {M <- factor(Material);
                   T <- factor(Temperature);
                   summary(lm(Life ~ M * T))})
```

But this way does not add the variables `M` and `T` to `batteryLife`.

If you use `within()` instead of `with()`, the new variables are added:

```
batteryLife <- within(batteryLife, {M <- factor(Material);
                                    T <- factor(Temperature)})
summary(lm(Life ~ M * T, batteryLife))
```

Additive Model: No Interactions

Model is

$$y_{i,j,k} = \mu + \tau_i + \beta_j + \epsilon_{i,j,k}$$

Use with care–first test significance of interactions;

In ANOVA table *without interactions*, "Error" line results from pooling Df and SS for "Interactions" and "Error" from the table *with* interactions.

With balanced data, main effect sums of squares and mean squares do not change, but F-ratios and P-values generally do change.

Example: Battery life

```
# Interaction model
summary(aov(Life ~ M * T, batteryLife))
            Df Sum Sq Mean Sq F value    Pr(>F)
M            2  10684  5341.9  7.9114  0.001976 **
T            2  39119 19559.4 28.9677 1.909e-07 ***
M:T          4   9614  2403.4  3.5595  0.018611 *
Residuals   27  18231   675.2
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1


# Additive model
summary(aov(Life ~ M + T, batteryLife))
            Df Sum Sq Mean Sq F value    Pr(>F)
M            2  10684  5341.9  5.9472  0.006515 **
T            2  39119 19559.4 21.7759 1.239e-06 ***
Residuals   31  27845   898.2
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

One Observation per Cell

Only a *single* replicate:

$$y_{i,j} = \mu + \tau_i + \beta_j + (\tau\beta)_{i,j} + \epsilon_{i,j}$$

Degrees of freedom for error $= ab(n-1) = 0$, so we cannot test the usual hypotheses about main effects and interactions.

Additive (no-interaction) model may still be fitted, and we can test for a less general form of interactions.

Tukey's One Degree of Freedom for Non-Additivity

Assumes *structured* interactions $(\tau\beta)_{i,j} = \gamma\tau_i\beta_j$.

Can fit with one observation per cell, and test $H_0 : \gamma = 0$.

Sum of squares is

$$\mathrm{SS}_N = \frac{\left[\sum_{i=1}^{a}\sum_{j=1}^{b} y_{i,j}y_{i\cdot}y_{\cdot j} - y_{\cdot\cdot}\left(\mathrm{SS}_A + \mathrm{SS}_B + \frac{y_{\cdot\cdot}^2}{ab}\right)\right]^2}{ab\mathrm{SS}_A\mathrm{SS}_B}$$

Break this out as a separate line in the ANOVA table.

Tukey's ODOFNA is not implemented in some packages; ANOVA table can be found by including *squared fitted values* in model, *after* the other effects.

Example: impurity data

impurity.txt; the ANOVA line for `I(fitted(a)^2)` is the same as that for "Nonadditivity" in Example 5.2:

```
impurity <- read.table("data/impurity.txt", header = TRUE)
a <- aov(Impurity ~ factor(Temperature) + factor(Pressure), impurity)
summary(a)

                    Df Sum Sq Mean Sq F value    Pr(>F)
factor(Temperature)  2 23.333  11.667  46.667 3.885e-05 ***
factor(Pressure)     4 11.600   2.900  11.600  0.002063 **
Residuals            8  2.000   0.250

summary(aov(Impurity ~ factor(Temperature) + factor(Pressure)
                       + I(fitted(a)^2), impurity))

                    Df  Sum Sq Mean Sq F value    Pr(>F)
factor(Temperature)  2 23.3333 11.6667 42.9491 0.0001174 ***
factor(Pressure)     4 11.6000  2.9000 10.6759 0.0042006 **
I(fitted(a)^2)       1  0.0985  0.0985  0.3627 0.5660026
Residuals            7  1.9015  0.2716
```

General Factorial Design

More than two factors.

Terms in model:

- main effects A, B, C, ...;
- two-factor interactions AB, AC, ...;
- three-factor interactions ABC, ...;
- and so on.

E.g. three factor statistical model:

$$y_{i,j,k,l} = \mu + \tau_i + \beta_j + \gamma_k + (\tau\beta)_{i,j} + (\tau\gamma)_{i,k} + (\beta\gamma)_{j,k} + (\tau\beta\gamma)_{i,j,k} + \epsilon_{i,j,k,l}$$

Example: Soft drink bottling (soft-drink-bottling.txt),

```
Carbonation Pressure Speed Height
10          25       200   -3
10          25       200   -1
10          25       250   -1
10          25       250    0
10          30       200   -1
10          30       200    0
10          30       250    1
10          30       250    1
12          25       200    0
12          25       200    1
12          25       250    2
...         ...      ...   ...
```

R commands

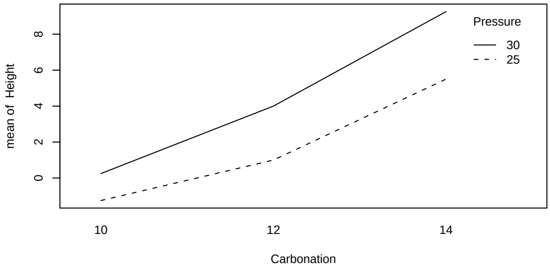
```
softDrinkBottling <- read.table("data/soft-drink-bottling.txt", header = TRUE)
softDrinkBottling <- within(softDrinkBottling,
                            {C <- factor(Carbonation);
                             P <- factor(Pressure);
                             S <- factor(Speed)})
summary(aov(Height ~ C * P * S, softDrinkBottling))
```

Output

```
            Df  Sum Sq Mean Sq  F value    Pr(>F)
C            2 252.750 126.375 178.4118 1.186e-09 ***
P            1  45.375  45.375  64.0588 3.742e-06 ***
S            1  22.042  22.042  31.1176 0.0001202 ***
C:P          2   5.250   2.625   3.7059 0.0558081 .
C:S          2   0.583   0.292   0.4118 0.6714939
P:S          1   1.042   1.042   1.4706 0.2485867
C:P:S        2   1.083   0.542   0.7647 0.4868711
Residuals   12   8.500   0.708
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Interaction plot

```
with(softDrinkBottling, interaction.plot(Carbonation, Pressure, Height))
```



Response Curves

When one or more factors is *quantitative*, a regression model can help in understanding the relationship.

Example: battery life

Temperature is quantitative; fit quadratic equations, separately by material:

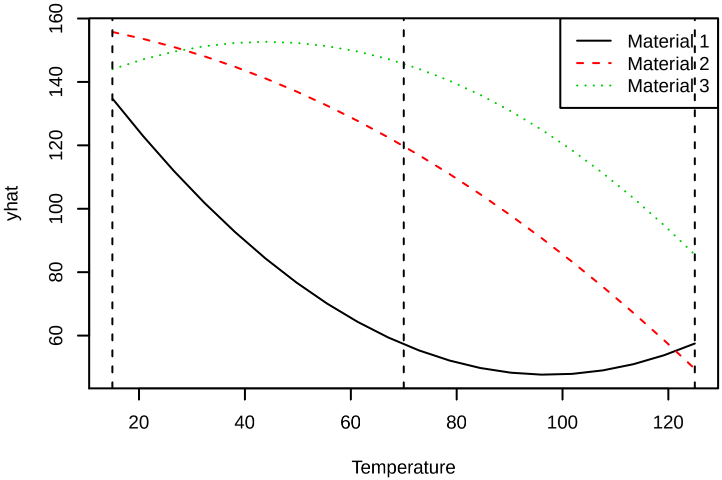
```
l <- lm(Life ~ M * (Temperature + I(Temperature^2)), batteryLife)
summary(l)
```

```
Call:
lm(formula = Life ~ M * (Temperature + I(Temperature^2)), data = batteryLife)

Residuals:
    Min      1Q  Median      3Q     Max
-60.750 -14.625   1.375  17.938  45.250

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         169.380165  20.567656   8.235 7.66e-09 ***
M2                   -9.756198  29.087058  -0.335  0.73991
M3                  -36.617769  29.087058  -1.259  0.21884
Temperature          -2.501446   0.755148  -3.313  0.00264 **
I(Temperature^2)      0.012851   0.005260   2.443  0.02139 *
M2:Temperature        2.328099   1.067941   2.180  0.03815 *
M3:Temperature        3.404339   1.067941   3.188  0.00361 **
M2:I(Temperature^2)  -0.018512   0.007439  -2.488  0.01929 *
M3:I(Temperature^2)  -0.023099   0.007439  -3.105  0.00443 **
---
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
```

Plot the three curves:

```
ngrid <- 20
tg <- with(batteryLife, seq(min(Temperature), max(Temperature), length = ngrid))
grid <- expand.grid(Temperature = tg, M = levels(batteryLife$M))
yhat <- predict(l, grid)
yhat <- matrix(yhat, nrow = length(tg))
matplot(tg, yhat, type = "l", xlab = "Temperature")
abline(v = c(15, 70, 125), lty = 2)
legend("topright", legend = paste("Material", levels(batteryLife$M)),
       lty = 1:3, col = 1:3)
```

In this case, because Temperature has only 3 levels, and a quadratic can interpolate any 3 points, the response curves are just smoother versions of the interaction plot.


Material 1
Material 2
Material 3
yhat
160
140
120
100
80
60
20
40
60
80
100
120
Temperature

Response Surface

When two factors are quantitative, the regression model can include polynomial functions of both.

Example: cutting tool lifetime

The lifetime of a cutting tool is affected by two factors:

- Total tool angle;
- Cutting speed.

Data: tool-life.csv (Lifetimes are coded)

Interaction plots show that the effects of angle and speed are complicated:

```
toolLife <- read.csv("data/tool-life.csv")

with(toolLife, interaction.plot(Angle, Speed, Life))
with(toolLife, interaction.plot(Speed, Angle, Life))
```

Try a complete second-order model:

```
l <- lm(Life ~ Angle + Speed + Angle:Speed + I(Angle^2) + I(Speed^2),
                 toolLife)
summary(l)
```

Output

```
Call:
lm(formula = Life ~ Angle + Speed + Angle:Speed + I(Angle^2) +
    I(Speed^2), data = toolLife)

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) -1.000e+02  4.972e+01  -2.011   0.0673 .
Angle        4.567e+00  2.133e+00   2.141   0.0535 .
Speed        6.933e-01  5.804e-01   1.195   0.2553
I(Angle^2)  -8.000e-02  4.702e-02  -1.701   0.1146
I(Speed^2)  -1.600e-03  1.881e-03  -0.851   0.4116
Angle:Speed -8.000e-03  6.650e-03  -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
```

Lack of fit

The model does not fit well:

- No parameter is really significant;
- R^2 is low.

Test for lack of fit in the ANOVA table:

```
summary(aov(Life ~ Angle + Speed + Angle : Speed + I(Angle^2) + I(Speed^2)
            + factor(Angle) * factor(Speed), toolLife))
```

The last term (`factor(Angle) * factor(Speed)`) breaks up the 12 degrees of freedom for Residuals into:

- 9 d.f. for "pure error";
- 3 d.f. for "lack of fit".

Output

```
Analysis of Variance Table

Response: Life
                             Df Sum Sq Mean Sq F value   Pr(>F)
Angle                         1  8.333  8.3333  5.7692 0.039772 *
Speed                         1 21.333 21.3333 14.7692 0.003948 **
I(Angle^2)                    1 16.000 16.0000 11.0769 0.008824 **
I(Speed^2)                    1  4.000  4.0000  2.7692 0.130451
Angle:Speed                   1  8.000  8.0000  5.5385 0.043065 *
factor(Angle):factor(Speed)   3 53.333 17.7778 12.3077 0.001548 **
Residuals                     9 13.000  1.4444
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

The `Residuals` line is now pure error, and the `factor(Angle) * factor(Speed)` line is lack of fit; it is highly significant.

Response surface

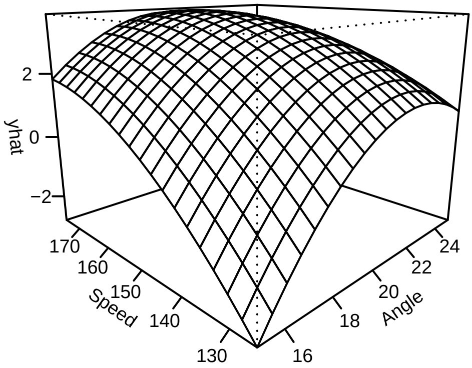
Even though the model does not fit well, we can use it as an example of a response surface:

```
ngrid <- 20
Angle <- with(toolLife, seq(min(Angle), max(Angle), length = ngrid))
Speed <- with(toolLife, seq(min(Speed), max(Speed), length = ngrid))
grid <- expand.grid(Angle = Angle, Speed = Speed)
yhat <- predict(l, grid)
yhat <- matrix(yhat, length(Angle), length(Speed))
persp(Angle, Speed, yhat, theta = -45, expand = 0.75, ticktype = "detailed")
```

We could use the same steps to plot the response surface for other models, such as:

```
l <- lm(Life ~ (Angle + I(Angle^2)) * (Speed + I(Speed^2)), toolLife)
```

Response surface for the complete second-order model:



Blocking in a Factorial Design

For a single nuisance factor, use a *blocked* design:

- The design has $abc\dots$ treatments;
- A *randomized complete block design* has all $abc\dots$ treatments in each block;
- The RCBD may be infeasible: too many treatments per block;
- *Incomplete* block designs provide the alternative.

Statistical model for blocked two-factor design (no interaction between block and experimental factors):

$$y_{i,j,k} = \mu + \tau_i + \beta_j + (\tau\beta)_{i,j} + \delta_k + \epsilon_{i,j,k}$$

With two (or more) nuisance factors, use *Latin Square* (or hyper-square) design:

- E.g. for a 3×2 factorial design, use a 6×6 Latin Square.
- Statistical model:

$$y_{i,j,k,l} = \mu + \tau_i + \beta_j + (\tau\beta)_{i,j} + \delta_k + \theta_l + \epsilon_{i,j,k,l}$$