Factorial Designs II

Sean Hellingman ©

Design for Data Science (ADSC2030)

shellingman@tru.ca

Winter 2025

THOMPSON RIVERS UNIVERSITY

Topics

Introduction

- When many factors are under study, it is more efficient to study them together.
- There is more power in detecting main effects.
- Higher order interactions may also be detected.

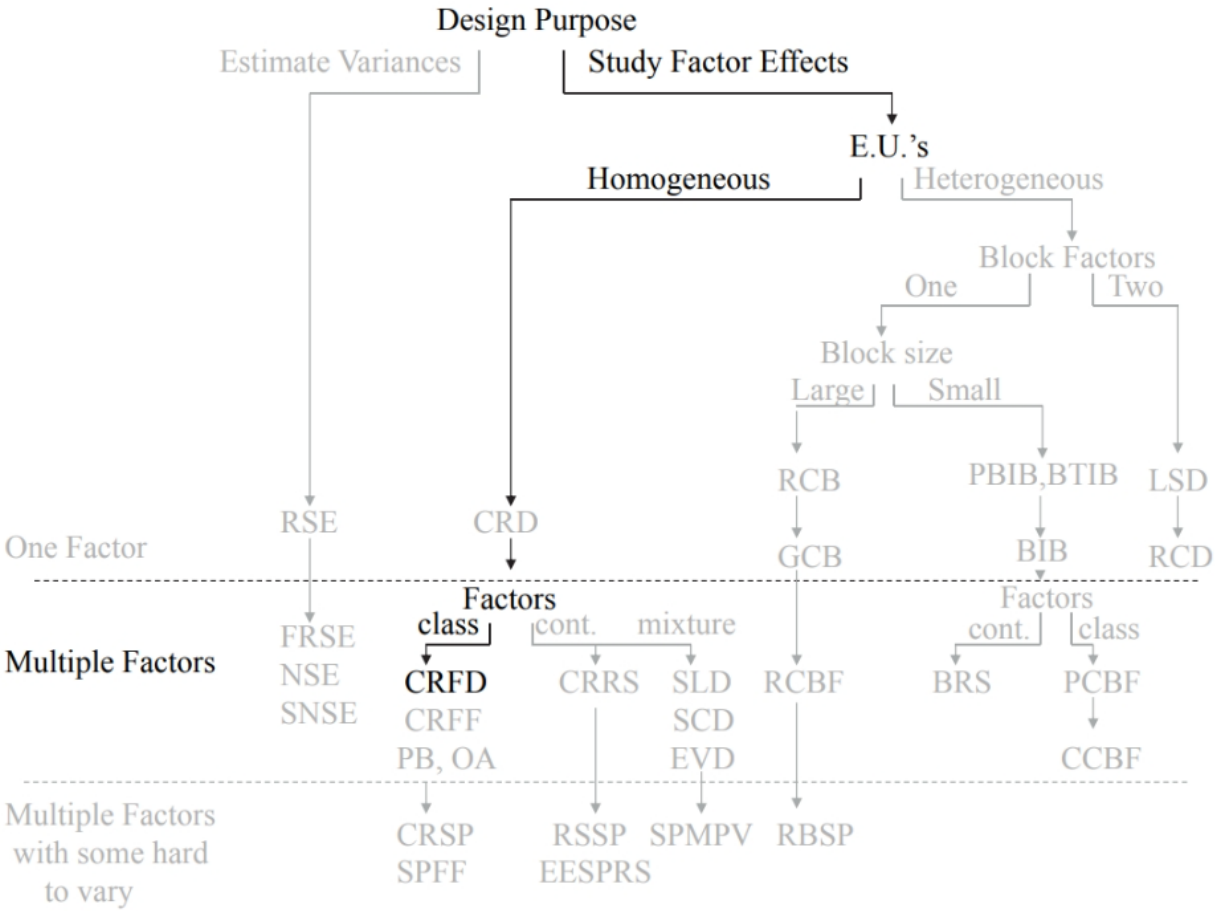


Figure: Source: (1)

Factorial Designs

- Factorial Designs examine all possible combinations of factor levels.
- The number of replicates of a specific level of one factor is increased by the product of the number of levels of all other factors in the design.
- The same power or precision can be obtained with fewer replicates.

Multi-Factor Designs in R I

- *There are many ways to create this kind of design in R.*
- We can use the `expand.grid()` function to create a data frame of all possible combinations:
  - ```
    D <- expand.grid(Factor.1 =
        as.factor(c(level.1,...,level.i)),
        Factor.2 = as.factor(c(level.1,...,level.j)), ...,
        Factor.k = as.factor(c(level.1,...,level.m)) )
    ```
- To replicate you can use the `rbind()` function:
  - `D <- rbind(D,D)`
    ```

Multi-Factor Designs in R II

- Randomization:
  - `set.seed(2030) # Reproducible`
  - `D <-D[order(sample(1:nrow(D))), ] # Randomization`
  - `write.csv(D, file="Design.csv") # save optional`

Example 1

- Assume that we examine four levels for each of the two previous factors from the helicopter experiment and also want to examine two drop heights:
  - Wing Length: 4, 4.75, 5.5, 6 (inches)
  - Body Width: 3.25, 3.75, 4, 4.25 (inches)
  - Drop Height: 1.5, 2 (metres)

- Use R to come up with a randomized three-factor design with two replications for each combination.

Mathematical Model (Multiple Factors)

- The mathematical model for a completely randomized multi-factor factorial design (m factors) can be written as:

$$Y_{ij...mk} = \mu_{ij...m} + \epsilon_{ij...mk}. \tag{1}$$

- i represents the level of the first factor.
- m represents the level of the z^{th} factor.
- k represents the replicate number.

- This model is called a *cell means model* and $\mu_{ij...m}$ represents the expected response in the $ij...m^{th}$ cell.

Alternative Mathematical Model

- The effects model:

$$Y_{ij...mk} = \mu + \alpha_i + \beta_j + ... + \gamma_m + \alpha\beta_{ij} + ... + \alpha\beta \cdots \gamma_{ij...m} + \epsilon_{ij..mk}. \quad (2)$$

- α_i, β_j, ... γ_m are the main effects:
  - α_i represents the difference between the marginal average of all experiments at the i^{th} level of the first factor and the overall average.
  - γ_j represents the difference between the marginal average at the m^{th} level of the z^{th} factor and the overall average.
- **Now we consider all possible interactions.**

Illustrative Example 1

- Using the *web* data from the *daewr* package and the description in the example code (changes in the webpage configuration) we have the following model:

$$\text{prop}_{ijkl} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \gamma_k + \alpha\gamma_{ik} + \beta\gamma_{jk} + \alpha\beta\gamma_{ijk} + \delta_l + \alpha\delta_{il} + \beta\delta_{jl} + \alpha\beta\delta_{kl} + \gamma\delta_{kl} + \alpha\gamma\delta_{ikl} + \beta\gamma\delta_{ikl} + \alpha\beta\gamma\delta_{ijkl}$$

- *Because we have a different number of replicates we will use the* `contr.sum` *and* `type = "III"` *arguments in* R.

Effects Model in R

- Use the `lm()` function to estimate the linear model with all interactions.
- **If there is only one observation per group you will need to remove a term**
  - Otherwise, include all of the terms.
- Next use the `Anova()` function from the *car* package with the `type = "III"` argument to examine the significance of the overall interactions.

Example 2

- Estimate the effects (linear) model with the `prop` variable as the response variable.
- Omit the four-way interaction `A:B:C:D` as we only have one replication for each group.
- Perform an ANOVA on your resulting model.
- What do you think your results imply?

Visualizing Interactions

- We can use the code from the previous slides to examine interactions or we can use the `plot_model(lm, type = "int")` function from the *sjPlot* package.

Example 3

- Estimate the effects model on the *COdata* that we covered in Example 3 of Factorial Designs I.
- Use the `plot_model()` function to examine the interaction term.
- Use the `plot_model()` function to examine the interaction terms from Example 2.

Two-Level Factorials

- As we increase the number of factors in a design, the treatment combinations increases exponentially.
  - Four factors with 5 levels: $4^5 = 1024$ runs needed
- Very popular approach is to design experiments with two-level factors (2^k).
- The two levels are often denoted (-) and (+) for lowest and highest respectively.

Two-Level Factorials Notation

- We can replace the i notation by $+$ or $-$.
- For Example: $\alpha_- = -\alpha_+$
- We can define the effects of the main effects of a two-level factorial:

$$E_A = \bar{y}_{+\ldots} - \bar{y}_{-\ldots}$$

- Represents the change in the average response caused by going from low (-) to high $(+)$ in factor A.
- β_A is one half of the effect E_A.

Effects

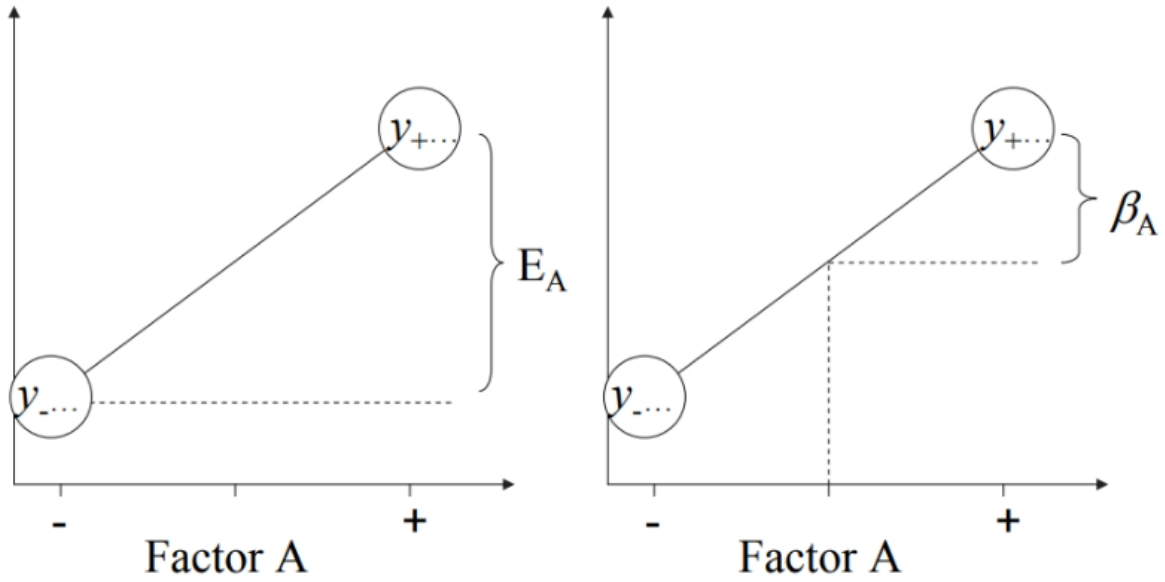
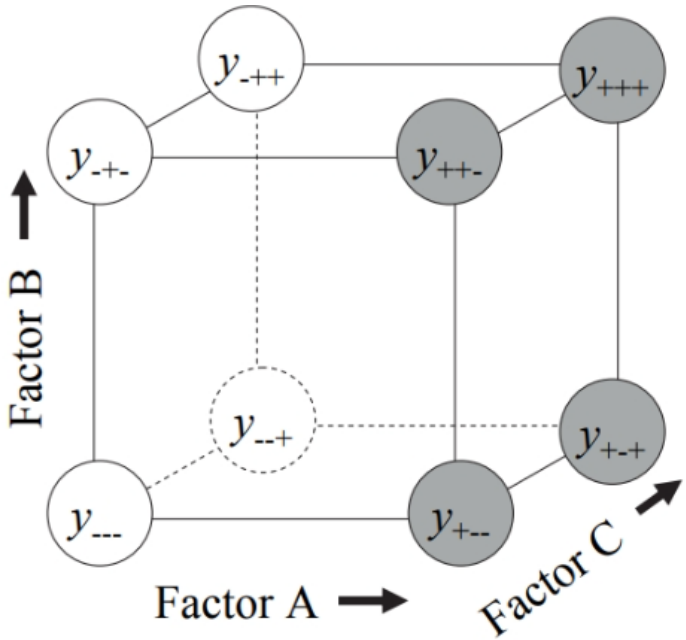


Figure: Source: (1)

Geometric Representation (2^3)

Run	X_A	X_B	X_C	Response
1	-	-	-	y_{---}
2	+	-	-	y_{+--}
3	-	+	-	y_{-+-}
4	+	+	-	y_{++-}
5	-	-	+	y_{--+}
6	+	-	+	y_{+-+}
7	-	+	+	y_{-++}
8	+	+	+	y_{+++}



$$E_A = (y_{+--} + y_{++-} + y_{+-+} + y_{+++})/4 - (y_{---} + y_{-+-} + y_{--+} + y_{-++})/4$$

Figure: Source: (1)

Two-Level Factorials Notation

- We can examine the terms including interactions in R using the `lm()` function.

- To include the *half effects* we need to include the `contr.FrF2` from the *DoE.base* package argument to the estimation process.

$$X_A = \frac{ActualFactorSetting - FactorMidPoint}{HalfTheRange}$$

- R Code:
  - `model <- lm(response ~ Factor.1*Factor.2*...*Factor.m, data = data, contrast = list(Factor.1=contr.FrF2, Factor.2=contr.FrF2,..., Factor.m=contr.FrF2))`
  - `summary(model)`

Example 4

- Load the *volt* dataset from the *daewr* package into R.
- Take some time to examine the data.
- Estimate a linear model using the method covered on the previous slide.
- **Remember, these estimates are half of the main effects values.**
- Are the main effects meaningful?

Example 5

- Visualize any significant interactions from Example 4.

Example 4 Comments

- Removing any insignificant terms we may obtain the following model formula:

$$y = 668.563 - 16.813\left(\frac{Temp - 27}{5}\right) - 6.688\left(\frac{CWarm - 2.75}{2.25}\right)\left(\frac{Temp - 27}{5}\right)$$

- *We have a significant interaction.*

Number of Replicates Shortcut I

- Assume we are interested in a power equal to 0.95 for a two-level factorial design with $\alpha = 0.05$.
- The formula to obtain the number of **runs** N:

$$N = ((8\sigma)/\Delta)^2. \tag{3}$$

- σ is the standard deviation of the experimental error.
- $N = r \cdot 2^k$
- **This formula only works for two-level factorial designs.**

Example 6

- Use the formula on the previous slide to write a function in R to estimate the number of replicates needed.

- Use your function and the following information to approximate how many replicates will we need to obtain a power of 0.95 for the experiment from Example 4.
  - $\sigma = 15.0$ (Known by lab technician)
  - $\Delta = 30.0$ (Suggested by students)

Number of Replicates Shortcut II

- Assume we have a budget for an experiment.
- The formula can be rearranged to find Δ (the size of effect we are likely to detect):

$$\Delta = 8 \cdot \sigma / \sqrt{N} \tag{4}$$

- σ is the standard deviation of the experimental error.
- $N = r \cdot 2^k$
- **This formula only works for two-level factorial designs.**

One Replicate Per Cell

- It is very possible that only one replicate per cell is *needed*.
- In the case that there is only one replicate per cell, we cannot estimate the variance of the error term to conduct any of the hypothesis tests.
- There are visualisation methods that we can use to identify possible significant effects.

One Replicate Per Cell in R

Estimate a model (`contr.FrF2` not needed):

- `model <- lm(response ~ A*B*..., data = data)`

Off diagonal elements from this plot are *significant*:

- `library(daewr)`
- `fullnormal(coef(model)[-1],alpha=.025)`

- *There are other methods to detect significant terms.*
  - `LGB(coef(model)[-1], rpt = FALSE)`

Example 7

- Import the *chem* dataset into R.
- Take a moment to understand the data.
- Are there any significant factors in this experiment?

Model Assumptions with Replication

- When **there are replicates for each cell** (combination of factor levels) we can use the previous methods discussed to check for normality and a constant variance.

- Methods:
  - Scatterplots of residuals
  - Q-Q plot
  - Shapiro-Wilk Test
  - `ncvTest()`

Model Assumptions WITHOUT Replication

- When we do not have replication it is more difficult to test for violations.
- Generally, violations of normality are driven by an *outlier.*
  - Will bias the estimated effects away from 0. (recall the plots used to detect significance for one replicate)
- In R:
  - `Gaptest(data)` *Daniel's Method to find an outlier in an unreplicated* 2^k *design*
  - *The response is in the last column of the data frame.*

Comments on Outliers

- When a detected *outlier* is removed or corrected, results should be interpreted with caution.
- When we have multiple replicates at factor setting where an outlier is detected, it may be okay to interpret the results.
- **If there are two or fewer replicates and an outlier is detected, it may be advisable to rerun the experiment.**

Example 8

- Import the *BoxM* dataset into R.
- Use the techniques we used in Example 7 to detect any significant factors with regards to the response y.
- Use the `Gaptest()` function to detect any possible outliers.
- What conclusions can we draw from the results?

Exercise 1

- Take some time to work through the examples in these slides on your own.
  - See if you can solve them and interpret the results without the *Filled* example code.

References & Resources

1. Lawson, J. (2014). *Design and Analysis of Experiments with R (Vol. 115)*. CRC press.

- plot_model()
- contr.FrF2
- fullnormal()
- LGB()
- Anova
- Gaptest()