Introduction to 2^2 Factorial Design

Dr Austin R Brown

Kennesaw State University

Introduction

- In the last section, we learned about factorial designs.
- Such a design is appropriate when we have multiple treatment factors (each with 2+ levels) and we want to study these main effects and the potential interactions between them.
- But let's consider some other situations where a full factorial design might initially make sense, but some alternatives may prove more effective and useful.

Introduction

- Remember in the Instagram Ads example, we had two treatment factors, Ad Content and Ad Format, each with two levels (funny/informational and image/video).
- This gave us a total of four treatment combinations.
- Suppose we wanted to rerun the experiment, but now we have a third treatment factor, Ad Color Palette, with three levels (blue, red, and green).
- This would give us a total of 12 treatment combinations.

Introduction

- Supposing we would want 10 replications of each treatment combination, this would require a total of 120 experimental units.
- This is an increase of 80 experimental units from the original design (supposing $r = 10$).
- In some cases, the cost of running these additional experiments and/or collecting these additional observations may not be overly high.
- But in other cases, it may be prohibitive.
  - So what do we do?

Introduction

- In experimental situations where we may be interested in screening a large number of factors <u>or</u> where the cost of running all factor combinations is too great in terms of time or money, a 2^k *factorial design* provides and efficient way to explore our main effects and interactions with fewer experiments.

- A 2^k design is used to study the effects of k factors, each at two levels (e.g., high and low), on some outcome/response.
  - It allows researchers to efficiently explore the main effects and interactions of multiple factors with a **manageable number of experimental runs**

Introduction

- The key here is taking a potentially complex design and simplifying it by having every treatment effect simply be a pairwise comparison.
- As we will see, it also simplifies computations quite a bit.
- Before we get into the general case, let's talk about the simplest case, the 2^2 factorial design.

Experimental Example

- A popular hot wings chain called **Buster's Blazing Wings** wants to develop a new mobile app to enhance customer engagement and boost online orders.

- The development team has identified two key factors that may impact **monthly purchases**
  1. Push Notifications (we'll call this factor A) - Enabled (+) or Disabled (-)
  2. Loyalty Program (we'll call this factor B) - Basic (points-based +) or Enhanced (includes exclusive discounts -)
     - Note, we will use $+/-$ notation to denote the levels of the main effects

- For each of the $2^2 = 4$ treatment combinations, suppose we randomly recruit $r = 3$ customers to pilot the app. After a month of using the app, we record their monthly purchases.
  - The data are stored in the `2^2 Wings Example.xlsx`

Considering the Context

- As we've done before, we can work out the roadmap of our experiment:
- We are trying to quantify the effect main effects (Push Notifications and Loyalty Program) as well as their interaction have on our outcome, monthly restaurant sales.
- You can imagine there likely being some lurking variables working with human participants, so hopefully we've done a good job sampling app users/customers who are relatively homogeneous.

Considering the Context

- Our hypotheses in this case would be the same as our hypotheses for the two-factor factorial design.
- We will discuss how the computations simplify in the 2^2 case.

Exploring the Data

- With such a small dataset, we can visually examine the structure in a tabular format:

A	B	Treatment Combo	Rep 1	Rep 2	Rep 3	Total
-	-	Enabled/Enhanced	151.80	179.99	154.92	486.71
-	+	Enabled/Points	136.15	141.49	152.41	430.05
+	-	Disabled/Enhanced	129.30	119.98	131.79	381.07
+	+	Disabled/Points	107.22	44.72	77.85	229.79

Table 1: 2^2 Factorial Design with Treatment Combinations and Replicates

Exploring the Data

- For the sake of notation, let:
  - Capital "A" denote the effect of factor A (Push Notifications in our case)
  - Capital "B" denote the effect of factor B (Loyalty Program in our case)
  - Capital "AB" denotes the interaction between the main effects.

- Additionally, let:
  - a represent the treatment combination of A at the $+$ level and B at the - level
  - b represent the treatment combination of A at the - level and B at the $+$ level
  - ab represents both factors at the $+$ level
  - (1) is used to denote both factors at the - level
  - We can use these symbols to denote the values in the *Total* column of Table 1

Exploring the Data: The Effect of A

- Our goal in a factorial design, 2^2 or otherwise, is to quantify the effect of a given factor or factors on the outcome.

- In a 2^2 factorial design, the average effect of a given factor (say A) is the change in the outcome/response when that factor's level changes ($+$ to - or vice versa), averaged across all levels of the other factor (say B).

Exploring the Data: The Effect of A

- The effect of A at the - level of B is (A at its + level less A at its - level while B is fixed at its - level averaged across the replicates):

$$\frac{a - (1)}{r}$$

- The effect of A at the + level of B is (A at its + level less A at its - level while B is fixed at its + level averaged across the replicates):

$$\frac{ab - b}{r}$$

Exploring the Data: The Effect of A

- Averaging these two quantifies yields the **main effect of A**:

$$A = \frac{1}{2}\left(\frac{a - (1)}{r} + \frac{ab - b}{r}\right)$$

$$A = \frac{1}{2r}(ab + a - b - (1))$$

Exploring the Data: The Effect of A

- So in our case:

$$A = \frac{1}{2(3)}(229.79+381.07-430.05-486.71) = \frac{-305.90}{6} = -50.98$$

- The effect of A is negatively signed suggesting that moving from the - level (Enabled Notifications) to the $+$ level (Disabled Notifications) is associated with a decrease in monthly spend.

Exploring the Data: The Effect of B

- We can do the exact same thing to quantify the effect of B. The effect of B at the - level of A is:

$$\frac{b-(1)}{r}$$

- The effect of B at the $+$ level of A is:

$$\frac{ab-a}{r}$$

Exploring the Data: The Effect of B

- Averaging these two quantifies yields the **main effect of B**:

$$B = \frac{1}{2r}(ab + b - a - (1))$$

- In our case, this works out to be:

$$B = \frac{1}{2(3)}(229.79 + 430.05 - 381.07 - 486.71) = \frac{-207.94}{6} = -34.66$$

Exploring the Data: The Effect of B

- Again, since the effect of B is negatively signed, this suggests that moving from the - level (Enhanced) to the $+$ level (Points) is associated with a decrease in monthly spend.
- Now, what about the interaction effect?

Exploring the Data: The Interaction Effect, AB

- We define the interaction effect, AB, as the average difference between the effect of A at the $+$ level of B and the effect of A at the - level of B:

$$AB = \frac{1}{2r}((ab - b) - (a - (1)))$$

- In our case:

$$AB = 12(3)(229.79 - 430.05 - 381.07 + 486.71) = \frac{-97.62}{6} = -15.77$$

Exploring the Data: The Interaction Effect, AB

- Here, the interpretation is a little more involved.
- Since the sign is negative, this implies that as we move from the - level (Enabled Notifications) of A to the $+$ level of A (Disabled Notifications) while simultaneously moving from the - level (Enhanced) level of B to the $+$ level of B (Points), we would expect to see a decrease in monthly spend.
- This is best observed in our boxplot/interaction combination plot:

Exploring the Data: The Interaction Effect, AB

```
library(tidyverse)
library(readxl)
library(rstatix)
## Read in the Data ##
wings <- read_excel("2^2 Wings Example.xlsx")
## Boxplot/Interaction Combo Plot ##
wings |>
  ggplot(aes(x = Push_Notifications, y = Sales, fill = Loyalty_Program)) +
  geom_boxplot() +
  geom_point(data = wings |>
               group_by(Push_Notifications,Loyalty_Program) |>
               get_summary_stats(Sales,type='mean_sd'),
             aes(x=Push_Notifications,y=mean,group=Loyalty_Program),
             shape=4,size=5) +
  geom_line(data = wings |>
              group_by(Push_Notifications,Loyalty_Program) |>
              get_summary_stats(Sales,type='mean_sd'),
            aes(x=Push_Notifications,y=mean,color=Loyalty_Program,
                group=Loyalty_Program)) +
  labs(title = "Sales by Push Notifications and Loyalty Program",
       x = "Push Notifications",
       y = "Sales",
       fill = "Loyalty Program") +
  theme_classic()
```