Effects of Different Factors on Income in Stardew Valley

Aidan Pierre-Louis

2024-05-10

INTRODUCTION AND DATA COLLECTION

The purpose of this experiment is to determine the optimal settings to obtain the maximum yield of income (integer value of gold) from the sale of crops in Stardew Valley. While there are many factors in Stardew Valley that presumably affect the sale price of crops, I have isolated 4 that I believe impact this response the most. The four factors in each run, each of which have two levels, are Crop: the type of crop being harvested and sold, Farming_Skill: the level of the player harvesting the crops, Fertilizer: the presence or absence of basic fertilizer in the soil the crops are grown in, and Number_of_Crops: the number of crops being harvested and sold. I chose these factors particularly because they are what come to mind when deciding what seeds to buy when I'm considering the general amount of profit I am striving for and am therefore curious to see just how impactful each factor is in magnitude on the response. In order to meaningfully record replicates of each run in this experiment I used the Stardew Valley Modding API (SMAPI–hyperlink) to rollback time to the morning of each day corresponding to a run to reharvest and sell the crops and modify my characters Farming_Level when and if it changes during the experimental process. To control crops dying or not growing I also used the SMAPI to rollback time if external factors such as debris, lightning, crows (scarecrows also used to control this) prevented me from being able to sell the specified Number_of_Crops. I also did all of the runs in the season of Summer in case season was a factor that needed controlling. Refer to the following table below for the treatment factors in this experiment and their assigned levels.

```
## Figure 1:
## # A tibble: 2 x 5
##   Levels `Crop(A)` `Farming_Skill(B)` `Fertilizer(C)` `Number_of_Crops(D)`
##   <chr>  <chr>                  <dbl> <chr>                           <dbl>
## 1 +      Corn                       5 Basic                              50
## 2 -      Wheat                      1 None                               30
```

In order to test these factors I designed the experiment as a 2^4 Full Factorial Design with 5 replications per run. I constructed the design as in the following table.

```
## Figure 2:
## # A tibble: 16 x 4
##    A     B     C     D
##    <chr> <chr> <chr> <chr>
##  1 +     +     +     +
##  2 +     +     +     -
##  3 +     +     -     +
##  4 +     +     -     -
##  5 +     -     +     +
##  6 +     -     +     -
##  7 +     -     -     +
```

```
##  8 +     -     -     -
##  9 -     +     +     +
## 10 -     +     +     -
## 11 -     +     -     +
## 12 -     +     -     -
## 13 -     -     +     +
## 14 -     -     +     -
## 15 -     -     -     +
## 16 -     -     -     -
```

This design is desirable because it provides a complete set of 16 runs which can be used to determine the significance of each factorial effect. Each run can be a bit time-consuming with 5 replicates, but the open-source repository Stardew Valley Profits Calculator–hyperlink– can be used to simulate experimental runs if one wishes to reproduce this experiment. The purpose of replicating each run 5 times is for consistency in response variable, so as to reduce the chances of some run being represented by a single response that is influenced by some underlying noise. The response variable is the sum of every crop's sale price in each replicate, and crops can sell for different prices depending upon their quality. The quality of a crop is determined by randomness as calculated within the game. Thus the noise in the data is derived from the game's intrinsic randomness in crop quality, and presumably the chances a crop is harvested at some quality is affected by any of these predictors. The following table details the values of the response under Rep1 through Rep5 respectively. The column y.bar is the mean response across all 5 replicates. The column s.sqr is the response variation across the 5 replicates. And lastly the column ln.s.sqr is the natural log of the response variation.

```
## Figure 3:
## # A tibble: 16 x 12
##    A     B     C     D      Rep1  Rep2  Rep3  Rep4  Rep5 y.bar  s.sqr ln.s.sqr
##    <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>    <dbl>
##  1 +     +     +     +      2992  3066  3044  2955  2953 3002  2642.      7.88
##  2 +     +     +     -      1867  1819  1768  1819  1733 1801. 2679.      7.89
##  3 +     +     -     +      2819  2805  2842  2795  2657 2784. 5320.      8.58
##  4 +     +     -     -      1660  1560  1572  1672  1621 1617  2541       7.84
##  5 +     -     +     +      2658  2759  2610  2696  2732 2691  3495       8.16
##  6 +     -     +     -      1637  1548  1648  1586  1634 1611. 1792.      7.49
##  7 +     -     -     +      2536  2549  2648  2537  2536 2561. 2385.      7.78
##  8 +     -     -     -      1500  1550  1550  1562  1525 1537.  619.      6.43
##  9 -     +     +     +      1520  1556  1484  1550  1460 1514  1728       7.45
## 10 -     +     +     -       960   894   882   888   888  902. 1055.      6.96
## 11 -     +     -     +      1376  1370  1376  1370  1376 1374.   10.8     2.38
## 12 -     +     -     -       834   810   828   804   786  812.  371.      5.92
## 13 -     -     +     +      1340  1358  1400  1316  1298 1342. 1559.      7.35
## 14 -     -     +     -       834   816   804   792   822  814.  263.      5.57
## 15 -     -     -     +      1280  1304  1274  1262  1292 1282.  263.      5.57
## 16 -     -     -     -       798   768   786   774   786  782.  137.      4.92
```

Refer to the appendix for information on how the values of each column was computed.

ANALYSIS

The following model

$$
\begin{aligned}
y =& \mu + \frac{A}{2} \cdot x_A + \frac{B}{2} \cdot x_B + \frac{C}{2} \cdot x_C + \frac{D}{2} \cdot x_D + \frac{AB}{2} \cdot x_A \cdot x_B \\
&+ \frac{AC}{2} \cdot x_A \cdot x_C + \frac{AD}{2} \cdot x_A \cdot x_D + \frac{BC}{2} \cdot x_B \cdot x_C + \frac{BD}{2} \cdot x_B \cdot x_D \\
&+ \frac{CD}{2} \cdot x_C \cdot x_D + \frac{ABC}{2} \cdot x_A \cdot x_B \cdot x_C + \frac{ABD}{2} \cdot x_A \cdot x_B \cdot x_D \\
&+ \frac{ACD}{2} \cdot x_A \cdot x_C \cdot x_D + \frac{BCD}{2} \cdot x_B \cdot x_C \cdot x_D + \frac{ABCD}{2} \cdot x_A \cdot x_B \cdot x_C \cdot x_D + \epsilon
\end{aligned} \tag{1}
$$

was used to analyze the results of the data using regression analysis. The letter terms refer to the factorial effects. For factor A:

$$\frac{A}{2}$$

is the estimated regression coefficient

$$\beta_A$$

for the predictor variable

$$x_A$$

$$\epsilon$$

represents the error term (residuals).
Normality in residuals need not be analyzed as can be seen from the following lm call summary. The model we have constructed has an R^2 value of 1, thus all the variation in the data is explained by our predictors, and our residuals have no degrees of freedom and are all 0.

```
##
## Call:
## lm(formula = y.bar ~ A * B * C * D, data = df)
##
## Residuals:
## ALL 16 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    782.4        NaN     NaN      NaN
## A+             755.0        NaN     NaN      NaN
## B+              30.0        NaN     NaN      NaN
## C+              31.2        NaN     NaN      NaN
## D+             500.0        NaN     NaN      NaN
## A+:B+           49.6        NaN     NaN      NaN
## A+:C+           42.0        NaN     NaN      NaN
## B+:C+           58.8        NaN     NaN      NaN
## A+:D+          523.8        NaN     NaN      NaN
## B+:D+           61.2        NaN     NaN      NaN
## C+:D+           28.8        NaN     NaN      NaN
## A+:B+:C+        52.2        NaN     NaN      NaN
## A+:B+:D+        81.6        NaN     NaN      NaN
## A+:C+:D+        27.8        NaN     NaN      NaN
## B+:C+:D+        21.6        NaN     NaN      NaN
## A+:B+:C+:D+    -44.0        NaN     NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 15 and 0 DF,  p-value: NA
```

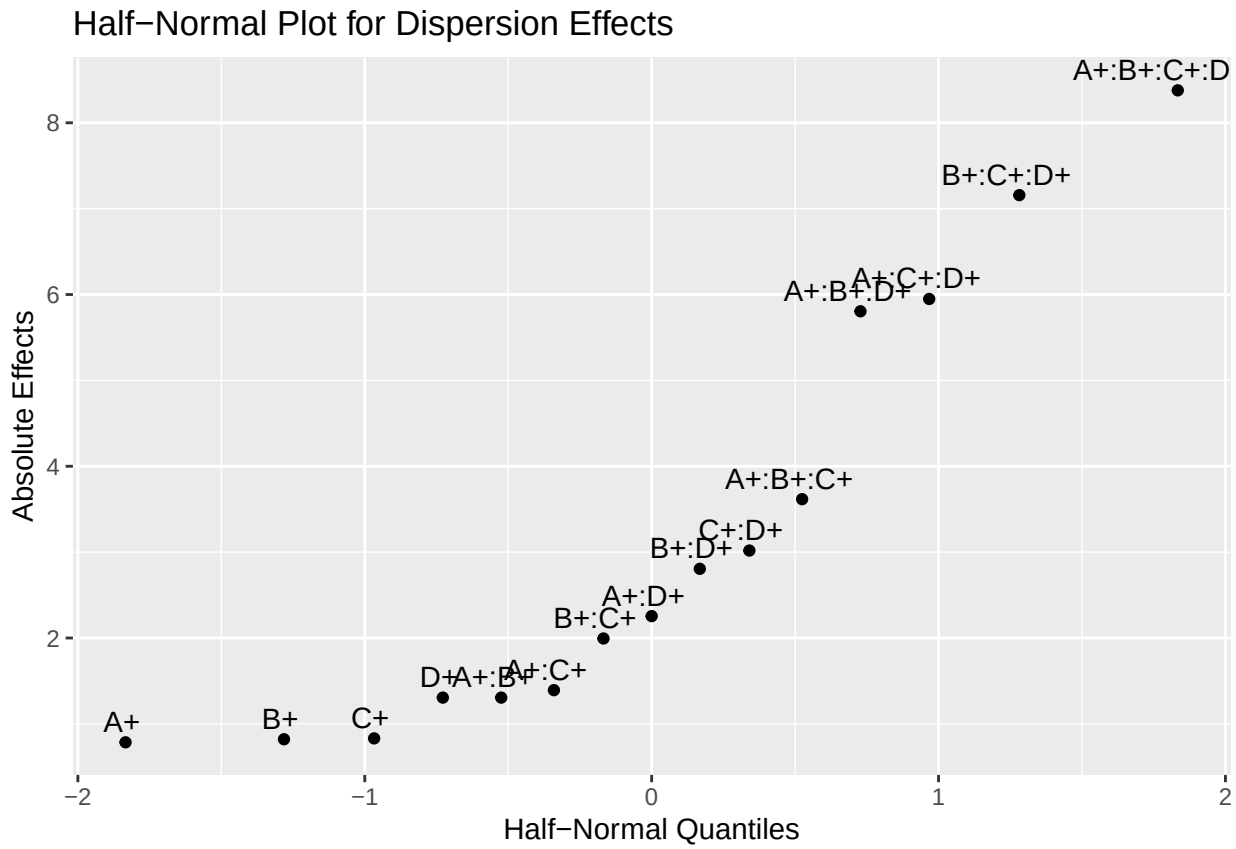
The following table lists the factorial effects for each term for both Location and Dispersion. The factorial effect is computed by doubling the estimated regression coefficient of the respective term. Refer to the appendix for code on how the values were computed.

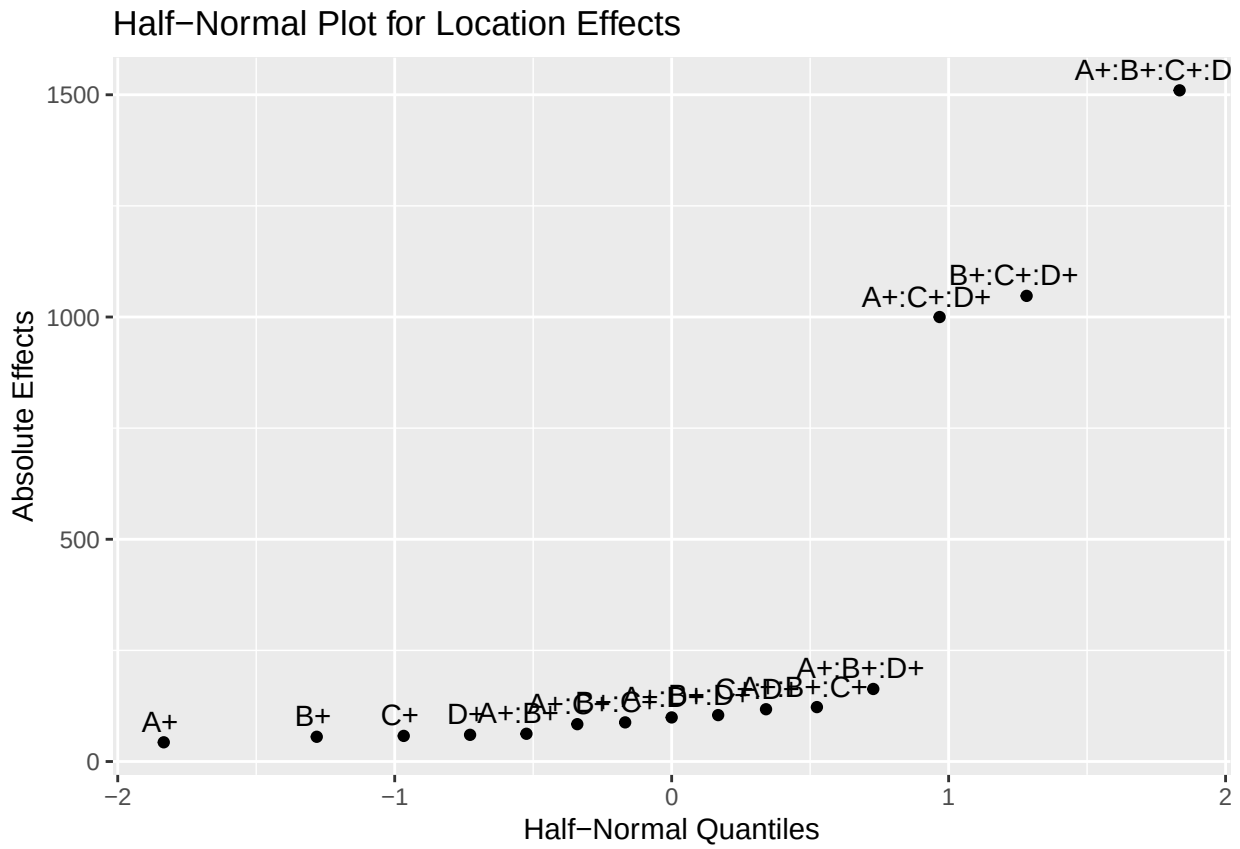
```
## Figure 4:
##             Location.Effects_y.bar Dispersion.Effects_ln.s.sqr
## A+                          1510.0                   3.0185242
## B+                            60.0                   1.9942857
## C+                            62.4                   1.3057466
## D+                          1000.0                   1.3057466
## A+:B+                         99.2                   0.8307761
## A+:C+                         84.0                   0.8206412
## B+:C+                        117.6                   0.7851407
## A+:D+                       1047.6                   1.3923464
## B+:D+                        122.4                  -8.3779800
## C+:D+                         57.6                   2.2548100
## A+:B+:C+                     104.4                  -2.8056073
## A+:B+:D+                     163.2                   7.1576430
## A+:C+:D+                      55.6                  -3.6166776
## B+:C+:D+                      43.2                   5.8046504
## A+:B+:C+:D+                  -88.0                  -5.9481244

## Intercept Term mu for Location: 782.4

## Intercept Term mu for Dispersion: 4.91852
```

All factorial terms with the exception of ABCD have positive effects on the response as it concerns location. It is less so clear the relationship between the terms and dispersion from the table, so the following half-normal plots were made to hopefully reveal any significant information from the factorial effects for location and dispersion.


Half−Normal Plot for Dispersion Effects
Absolute Effects
Half−Normal Quantiles
8
6
4
2
−2
−1
0
1
2
A+:B+:C+:D
B+:C+:D+
A+:C+:D+
A+:B+:D+
A+:B+:C+
C+:D+
B+:D+
A+:D+
B+:C+
A+:C+
D+
A+:B+
A+
B+
C+



The only abnormality present seems to be the effects ACD, BCD, and ABCD in the location half-normal plot, though because of the effect heredity principle, it would be a stretch to assume that these interaction terms are statistically significant when their main effects do not obviously deviate in the above plots.

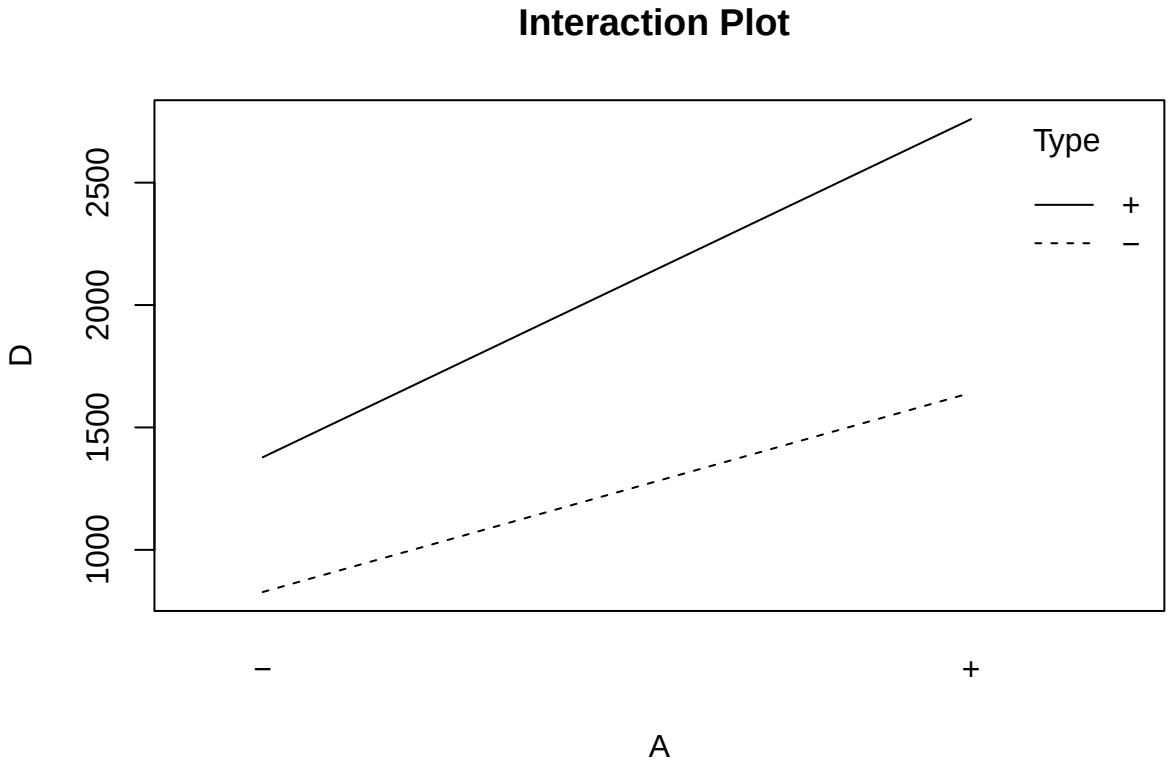
From here I decided to use Lenth's Method to determine effect significance. The following table lists the t_PSE statistics for each location and dispersion effect and the relevant IER critical value. Refer to appendix for computations.

```
## Figure 5:
##              Location.t_PSE Dispersion.t_PSE
## A+               11.7054264        0.8924696
## B+                0.4651163        0.5896389
## C+                0.4837209        0.3860626
## D+                7.7519380        0.3860626
## A+:B+             0.7689922        0.2456308
## A+:C+             0.6511628        0.2426342
## B+:C+             0.9116279        0.2321380
## A+:D+             8.1209302        0.4116670
## B+:D+             0.9488372        2.4770690
## C+:D+             0.4465116        0.6666667
## A+:B+:C+          0.8093023        0.8295177
## A+:B+:D+          1.2651163        2.1162590
## A+:C+:D+          0.4310078        1.0693222
## B+:C+:D+          0.3348837        1.7162275
## A+:B+:C+:D+       0.6821705        1.7586476

## IER Critical Value at alpha 0.05 and I = 15: 2.16
```

$$\text{Prob}(t_{\text{PSE},i} > \text{IER}_\alpha | H_0) = \alpha \tag{2}$$

The formula in (2) denotes that effect 'i' is statistically significant if its t_PSE value exceeds some threshold, that being the IER value, given the null hypothesis H_0 that all effects are zero. By examining the table, the only Dispersion term that exceeds the IER value of 2.16 is BD. Though, the effect heredity principles suggests at least one of the main effects of the interaction term be statistically significant to make any meaningful inferences on the interaction term itself. As for location effects both A and D exceed the IER critical value, as well as their interaction term AD. The following interaction plot displays the relationship between factors A and D on the response (income).



As can be seen from D+ increasing from A- to A+ and D- increasing from A- to A+, A and D have a synergistic relationship. With this information we can interpret the combined effect of A and D as greater than the sum of their individual effects.

RESULTS

Returning to Figure 4, we recall that every location effect is positive except ABCD. Keeping all 4 factors at their positive levels, especially the statistically significant A and D, maximizes the yield in our response. We can likely disregard the negative effect ABCD has on location because of the higher order nature of the term, the effect hierarchy principle suggests that it is less likely to be important compared to the lower-order effects. As for minimizing model dispersion, because there are no statistically significant main dispersion effects, we might like to only change B's and/or C's level to '-' so as to avoid changing the '+' levels of A and D for maximizing yield. We use the following regression equation for estimating the change in factor levels:

$$\hat{y} = \mu + \beta_A x_A + \beta_B x_B + \ldots + \beta_{ABCD} x_A x_B x_C x_D \tag{3}$$

where beta_factor is half the factorial effect for the respective term, and x_factor is 1 or -1 to represent the level it is set to. The target y.hat for maximizing yield is the maximum value attainable with the regression equation using the location coefficients (half location effects) optimally, and as for minimizing dispersion the optimal settings of the dispersion coefficients (half dispersion effects) achieve a value of y.hat as close to zero.

```
## Location (Yield) y.hat from levels (ABCD, ++++): 3002

## Location (Yield) y.hat from levels (ABCD, +--+): 2297.6

## Dispersion y.hat from levels (ABCD, ++++): 7.879481

## Dispersion y.hat from levels (ABCD, +-++): 8.438697

## Dispersion y.hat from levels (ABCD, ++-+): 9.278901

## Dispersion y.hat from levels (ABCD, +--+): 5.510236
```

In applying the aforementioned interpretations of location and dispersion effects to compute regression estimates for the proposed factor level combinations, we can see that whilst the optimal settings for dispersion are (+,-,-,+) for ABCD == 5.510236, estimated location suffers for those same levels == 2297.6. In other words we fail to maximize yield for ABCD (+,-,-,+). Preferably a slightly higher estimate of dispersion for ABCD (+,+,+,+) == 7.879481 is worth the much higher location estimate at ABCD (+,+,+,+) == 3002. Thus the optimal settings for maximixing the yield of our response variable 'income' are in the table as follows.

```
## Figure 6:
##                     Levels
## Crop(A)                  +
## Farming_Skill(B)         +
## Fertilizer(C)            +
## Number_of_Crops(D)       +
```

CONCLUSION

Having found the optimal settings to maximize income in Stardew Valley given the factors at their respective levels, the design for the data essentially reinforced a priori thought as to how the predictors affected the response as it pertains to location. Though interesting is their effects on variance in the data. We can interpret the data as our income in gold being optimally maximized by harvesting corn instead of wheat, having a higher farming skill when harvesting, using basic fertilizer instead of none, and harvesting a greater number of crops. I think the magnitude of difference between the response values for factors A and D (Crop and Number of Crops) somewhat overshadowed the effects of the other factors on the response because of their magnitude. A future experiment replacing these factors with ones that have underlying randomness in Stardew Valley's code may be more enlightening: such as a factpr which represents the presence of a scarecrow which drives random events that may kill a crop, a factor which represents the cost spent on the seeds of the crops (which detracts from income/profit) which may interact with the factor of how many crops were planted, or even standardizing/transforming the income based on the number of days a crop takes to grow and be harvested based on the different levels of a 'crop factor'. As far as these ideas relate to this experiment, the degree of complexity would be higher and possibly not properly 'modelable' using linear regression or a simple Full Factorial Design.

APPENDIX: CODE TO PRODUCE OUTPUTS

Code for computing Response Mean, Variation, and log(Variation) and Displaying Figure 3

```
df <- df[, 2:ncol(df)]
df$y.bar = rowMeans(df[, 5:9], na.rm = TRUE)
df$s.sqr = apply(df[, 5:9], 1, var, na.rm = TRUE)
df$ln.s.sqr = log(df$s.sqr)
print.my_df(3, df)
```

Code for computing Location and Dispersion Effects and Displaying Figure 4

```
lm.loc = lm(y.bar ~ A * B * C * D, data = df)
# summary(lm.loc)
location.effects = 2*lm.loc$coef[-1]
# print(location.effects)
mu.loc = lm.loc$coef[1]

lm.disp = lm(ln.s.sqr ~ A * B * C * D, data = df)
# summary(lm.disp)
dispersion.effects = 2*lm.disp$coef[-1]
# print(dispersion.effects)
mu.disp = lm.disp$coef[1]

effects.df = data.frame(Factor = names(location.effects), Location.Effects_y.bar = location.effects, Di
print.my_df(4, effects.df[-1])

cat('Intercept Term mu for Location:', mu.loc)
cat('Intercept Term mu for Dispersion:', mu.disp)
```

Code for Half-Normal Plots

```
library(ggplot2)
# half-normal plot for dispersion effects
halfnorm_data <- data.frame(Theoretical = qnorm(ppoints(length(abs(dispersion.effects)))),
                            Sample = sort(abs(dispersion.effects)),
                            Labels = names(dispersion.effects))

ggplot(halfnorm_data, aes(x = Theoretical, y = Sample, label = Labels)) +
    geom_point() +
    geom_text(aes(label = Labels), hjust = 0.6, vjust = -0.5) +
    labs(title = "Half-Normal Plot for Dispersion Effects",
         x = "Half-Normal Quantiles", y = "Absolute Effects")

halfnorm_data <- data.frame(Theoretical = qnorm(ppoints(length(abs(location.effects)))),
                            Sample = sort(abs(location.effects)),
                            Labels = names(location.effects))

ggplot(halfnorm_data, aes(x = Theoretical, y = Sample)) +
  geom_point() +
  geom_text(aes(label = Labels), hjust = 0.6, vjust = -0.5) +
```

```
  labs(title = "Half-Normal Plot for Location Effects",
       x = "Half-Normal Quantiles", y = "Absolute Effects")
```

Code for Lenth's Method t_PSE values and IER Critical Value and Figure 5

```
# t_PSE values for Location Effects
median.theta = median(abs(location.effects))
s0 = 1.5 * median.theta
trim.constant = 2.5 * s0
median.theta.trim = median(abs(location.effects[abs(location.effects) < trim.constant]))
PSE.loc = 1.5 * median.theta.trim
t_PSE.loc = abs(location.effects/PSE.loc)

# t_PSE values for Dispersion Effects
median.theta = median(abs(dispersion.effects))
s0 = 1.5 * median.theta
trim.constant = 2.5 * s0
median.theta.trim = median(abs(dispersion.effects[abs(dispersion.effects) < trim.constant]))
PSE.disp = 1.5 * median.theta.trim
t_PSE.disp = abs(dispersion.effects/PSE.disp)

# IER Critical values for alpha 0.1 0.05 and 0.01  respectively sourced from textbook appendix
IER_0.1 = 1.70
IER_0.05 = 2.16
IER_0.01 = 3.63

lenth.df = data.frame(Location.t_PSE = t_PSE.loc, Dispersion.t_PSE = t_PSE.disp)
print.my_df(5, lenth.df)

cat('IER Critical Value at alpha 0.05 and I = 15:', IER_0.05)
# cat('IER Critical Value at alpha 0.1 and I = 15:', IER_0.1)
```

Code for Interaction Plot

```
interaction.plot(df$A, df$D, df$y.bar, type = "l", legend = TRUE, trace.label = 'Type',
                 xlab = "A", ylab = "D",
                 main = "Interaction Plot")
```

Code for optimal factor settings' regression estimates

```
loc.coefs = lm.loc$coef
cat('Location (Yield) from levels (ABCD, ++++):', sum(loc.coefs))

loc.coefs = lm.loc$coef
# Identify indices where the name contains "B+" and "C+"
indices <- grep("B\\+", names(loc.coefs))
loc.coefs[indices] <- loc.coefs[indices] * -1
indices <- grep("C\\+", names(loc.coefs))
loc.coefs[indices] <- loc.coefs[indices] * -1
```

```r
cat('Location (Yield) from levels (ABCD, +--+):', sum(loc.coefs))

disp.coefs = lm.disp$coef
cat('Dispersion from levels (ABCD, ++++):', sum(disp.coefs))

disp.coefs = lm.disp$coef
# Identify indices where the name contains "B+"
# Multiply corresponding elements by -1
indices <- grep("B\\+", names(disp.coefs))
disp.coefs[indices] <- disp.coefs[indices] * -1
cat('Dispersion from levels (ABCD, +-++):', sum(disp.coefs))

disp.coefs = lm.disp$coef
# Identify indices where the name contains "C+"
indices <- grep("C\\+", names(disp.coefs))
disp.coefs[indices] <- disp.coefs[indices] * -1
cat('Dispersion from levels (ABCD, ++-+):', sum(disp.coefs))

disp.coefs = lm.disp$coef
# Identify indices where the name contains "B+" and "C+"
indices <- grep("B\\+", names(disp.coefs))
disp.coefs[indices] <- disp.coefs[indices] * -1
indices <- grep("C\\+", names(disp.coefs))
disp.coefs[indices] <- disp.coefs[indices] * -1
cat('Dispersion from levels (ABCD, +--+):', sum(disp.coefs))
```