Garden Sprinkler Experiment

[redacted]

Introduction

The goal of this project was to propose an experimental design for a research agency to build the best garden sprinkler. The characteristics that define a good sprinkler are low consumption and a wide spray range. Additionally, there are eight factors that we can adapt that could possibly have an impact on these quality characteristics. These are listed in Table 1 on the right.

Table 1: List of Factors	
Latin Letter	**Factor**
A	Vertical nozzle angle
B	Tangential nozzle angle
C	Nozzle profile
D	Diameter sprinkler head
E	Static friction moment
F	Dynamic friction moment
G	Entrance pressure
H	Diameter flow line

While we are interested in building the best garden sprinkler, we are restricted to 20 runs due to budgetary constraints. Consequently, this study has two primary goals: to be able to propose an experimental design that is able to identify the relevant factors that drive the water consumption and spray range, and to find the factors' settings that minimize the water consumption and maximize the spray range in just 20 runs. The overview and analysis provided below outlines how we managed to achieve these goals.

Methodology

Part I: Design of Experiment

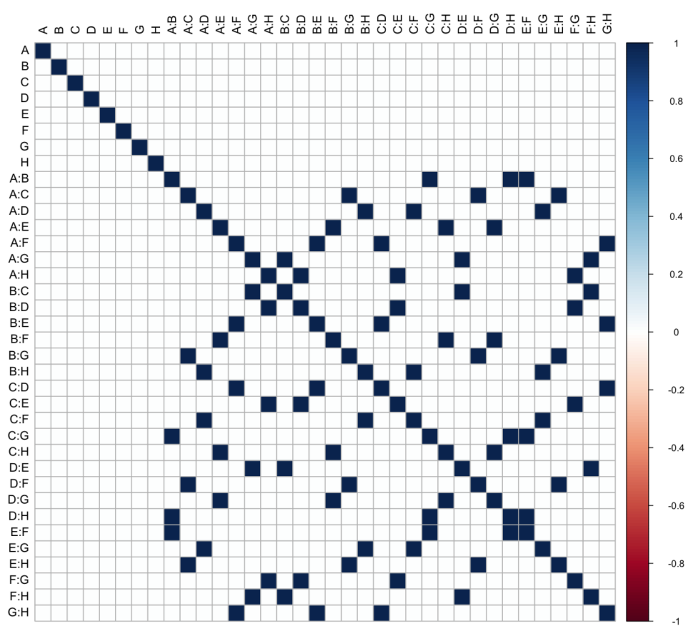
Question 1. Propose a cost-efficient experimental design. Motivate your decision in statistical and practical terms.

We propose that the research agency split the 20 runs into a 2^{8-4}_{IV} screening followed by 4 confirmation experiments. The screening will be done with the generators E = BCD, F = ACD, G = ABC, and H = ABD, since they allow for the highest possible resolution for this fraction (Montgomery, 2012, p.342). This produces the defining relation I = BCDE = ACDF, ABCG, ABDH. An overview of the runs can be

Table 2: 2^{8-4}_{IV} Fractional Factorial Design

Run	A	B	C	D	E = BCD	F = ACD	G = ABC	H = ABD
1	−	−	−	−	−	−	−	−
2	+	−	−	−	−	+	+	+
3	−	+	−	−	+	−	+	+
4	+	+	−	−	+	+	−	−
5	−	−	+	−	+	+	+	−
6	+	−	+	−	+	−	−	+
7	−	+	+	−	−	+	−	+
8	+	+	+	+	+	+	+	+
9	−	−	−	+	+	+	−	+
10	+	−	−	+	+	−	+	−
11	−	+	−	+	−	+	+	−
12	+	+	−	+	−	−	−	+
13	−	−	+	+	−	−	+	+
14	+	−	+	+	−	+	−	−
15	−	+	+	+	+	−	−	−
16	+	+	+	+	+	+	+	+

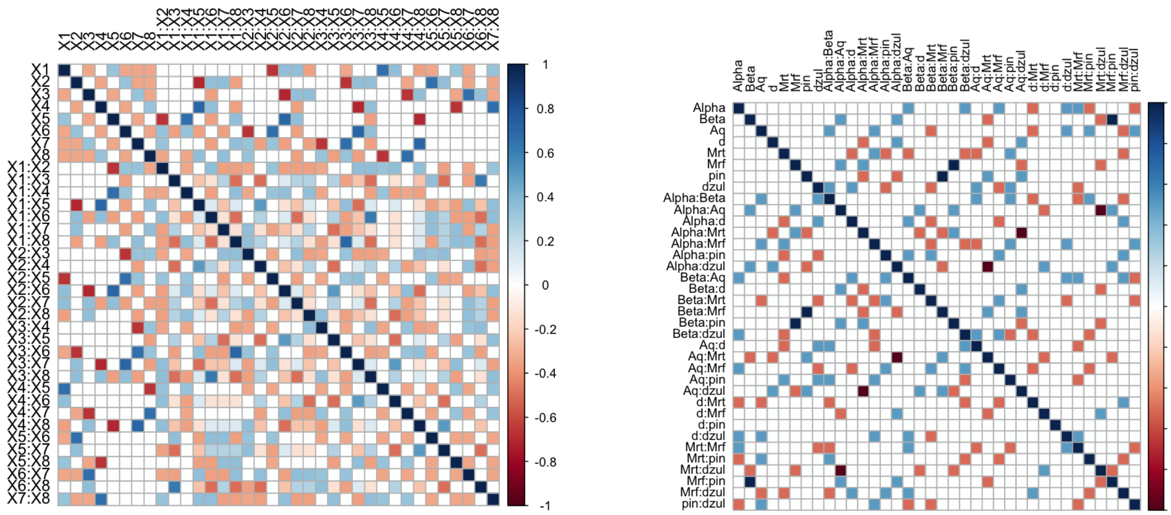
Figure 1: Alias Structure for 2^{8-4}_{IV} Design



found in Table 2. Additionally, the code used to create a preliminary dataframe in R can be found in Appendix A. While only 16 runs are being conducted, the alias structures that come with a resolution IV design ensure that the main effects are clear of two-factor interactions. This is shown in Figure 1 (code may be found in Appendix A). Additionally, we assume that three-factor and higher interactions are negligible. While the aliasing of two-factor interactions may be of some concern, the assumptions of effect hierarchy will hopefully resolve most of the issues that may come up. If not, we hope that the research agency may find the budget to conduct any follow-up experiments based on a projection of the fractional factorial design using the significant effects. As mentioned previously, following the 16 runs, we plan to complete 4 confirmation experiments.

Prior to designing this experiment, two other designs were considered. The Plackett-Burman Design with 12 runs was our first consideration. Since this is a non-geometric design, it has a complex partial aliasing structure with every two-factor interaction. This is not ideal, especially when we are under the assumption that two-factor interactions should be considered. Additionally, every main effect is also partially aliased with the two-factor interactions. This weighting of the two-factor interactions produces ambiguity that we do not need with the large number of factors that we are testing. Ultimately, this design was rejected because the partial aliasing means that in the long run, the experiment could potentially be more expensive depending on how much follow-up experimentation would be needed. In addition to the Plackett Burman Design, we also considered an experiment based on Optimal Screening with the D-optimality criterion. 16 runs followed by 4 confirmation experiments were considered because 16 is a multiple of 4, which means it is not only an orthogonal design, but it also allows for independent estimates or main effects. In the end, this design was also rejected because of concerns of partial aliasing, which could similarly make it more expensive in the long run. A correlation matrix for the Plackett-Burman Design and Optimal Screening Design can be found in Figure 2 (code may be found in Appendix A), in which partial aliasing is indicated.

Figure 2: Alias Structure for Plackett-Burman Design (Left) and Optimal Screening Design (Right)

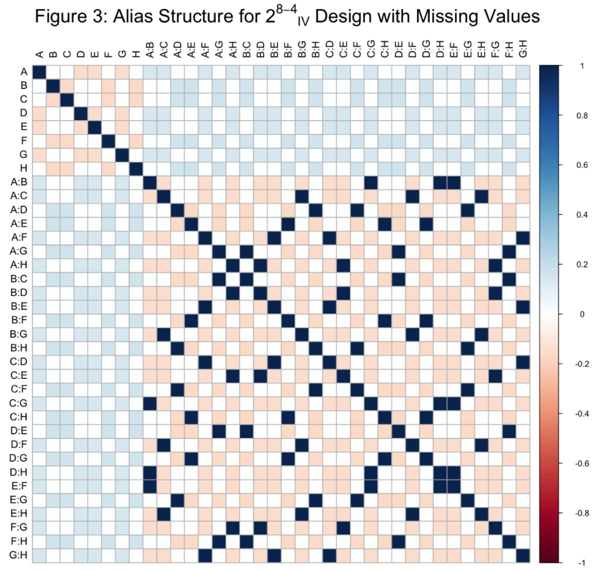


Question 2. What is the performance of your design for studying the main effects of the factors only? Can your design estimate all two-factor interactions? Why or why not?

Since our design is a 2^{8-4}_{IV} fractional factorial with resolution IV, the main effects are not aliased with any other main effect nor with any two-factor interaction. While the main effects will be fully aliased with three-factor interactions, we assume that under the principle of effect hierarchy, these three-factor interactions are not likely to be significant, and thus the effects we estimate can be fully attributed to the main effects. However, our design is unable to estimate all two-factor interactions, since all the two-factor interactions are fully aliased with each other; this alias structure can be seen in the correlation matrix displayed in Figure 1. A design of resolution V would be required in order to ensure that two-factor interactions are not aliased with any main effects or other two-factor interactions, but such a fractional design with eight factors would require far too many runs and is infeasible given the budgetary constraints.

Question 3. The production engineers are concerned about having some failed tests in the experiment, given by sprinklers which cannot spray water. If you remove two randomly chosen test combinations, what is the performance of the resulting design?

Figure 3: Alias Structure for 2^{8-4}_{IV} Design with Missing Values



By having some failed tests, the design is no longer orthogonal. This causes parameter estimates to be correlated and leads to inflation in the standard errors of the effects or regression model coefficients. This could also lead to partial aliasing. An example of a correlation matrix with some failed tests can be seen in Figure 3 (see Appendix A for code). This would not only lead to partial aliasing, but could cause some estimates of the main effects to be aliased. One potential solution to solving this problem would be to estimate the missing value with a number that makes the highest-order interaction contrast zero.

Question 4. The production engineers took an introductory course in experimental design. Using a commercial software, they came up with the experimental plan shown in Table 2. How does your full design compare with this one?

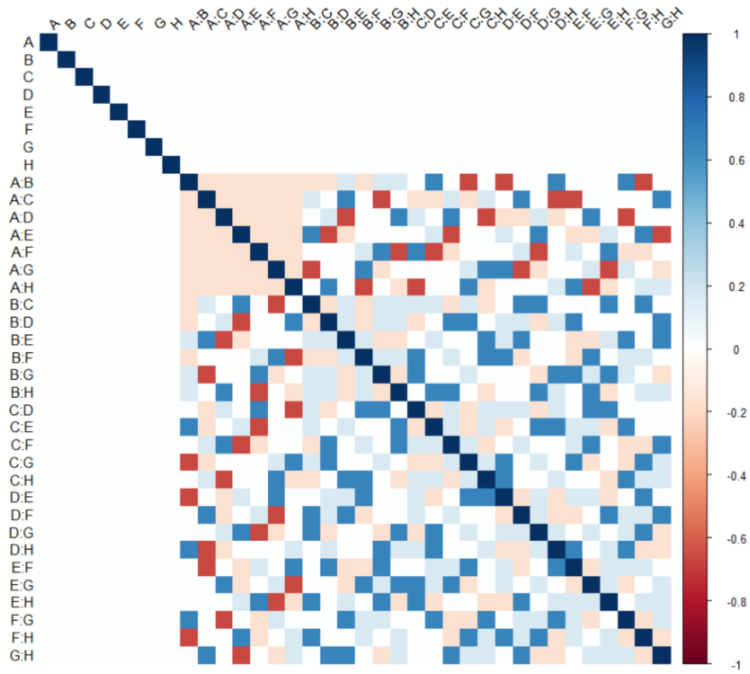
One main difference between our 2^{8-4}_{IV} fractional factorial design and the alternative experimental design shown in table 3 is that the alternative experimental design contains three levels and thus estimates a quadratic model, whereas our design only has two levels. Moreover, the alternative design has 17 runs, while our design only has 16.

Table 3: Alternative Experimental Design

Run	Alpha	Beta	A_q	d	M_t	M_f	p_in	d_zul
1	0	90	0.000002	0.2	0.02	0.015	1	5
2	45	45	0.000003	0.15	0.015	0.015	1.5	7.5
3	0	90	0.000004	0.15	0.01	0.01	2	5
4	0	0	0.000004	0.1	0.02	0.02	1.5	5
5	90	0	0.000002	0.15	0.02	0.02	1	10
6	0	45	0.000002	0.1	0.02	0.01	2	10
7	90	90	0.000002	0.1	0.015	0.02	2	5
8	0	0	0.000004	0.2	0.015	0.01	1	10
9	0	0	0.000002	0.2	0.01	0.02	2	7.5
10	90	0	0.000004	0.1	0.01	0.015	2	10
11	90	90	0.000004	0.1	0.02	0.01	1	7.5
12	90	90	0.000002	0.2	0.01	0.01	1.5	10
13	0	90	0.000003	0.1	0.01	0.02	1	10
14	90	0	0.000003	0.2	0.02	0.01	2	5
15	45	0	0.000002	0.1	0.01	0.01	1	5
16	45	90	0.000004	0.2	0.02	0.02	2	10
17	90	45	0.000004	0.2	0.01	0.02	1	5

As seen in the alias structure of our design in Figure 1 and the alias structure of the alternative design in Figure 4, both designs are able to estimate all main effects under the assumption of effect hierarchy, since the main effects are not aliased by any other main effect or any two-factor interactions and we can assume higher-order interactions to be negligible. In terms of the estimation of two-factor interactions, the alternative design has only partial aliasing for most of the two-factor interactions; in our design, all two-factor interactions are fully aliased with three other two-factor interactions, making it difficult to determine which interaction is the true effect. However, follow-up experiments can be conducted after our design to resolve any ambiguities resulting from the aliased two-factor interactions.

Figure 4: Alias Structure for the Alternative Experimental Design



Part II: Analysis of the Results

Question 5. Collect data using your recommended design in Question 1. Conduct a detailed data analysis.

Table 4: 2^{8-4}_{IV} Fractional Factorial Design

Run	A	B	C	D	E = BCD	F = ACD	G = ABC	H = ABD	Response 1	Response 2
1	−	−	−	−	−	−	−	−	0.218	3.488
2	+	−	−	−	−	+	+	+	0.028	4.849
3	−	+	−	−	+	−	+	+	0.388	4.792
4	+	+	−	−	+	+	−	−	0.000	3.487
5	−	−	+	−	+	+	+	−	0.321	8.624
6	+	−	+	−	+	−	−	+	0.046	6.566
7	−	+	+	−	−	+	−	+	0.288	6.756
8	+	+	+	+	+	+	+	+	0.000	8.716
9	−	−	−	+	+	+	−	+	0.129	3.492
10	+	−	−	+	+	−	+	−	0.021	4.618
11	−	+	−	+	−	+	+	−	0.391	4.733
12	+	+	−	+	−	−	−	+	0.025	3.262
13	−	−	+	+	−	−	+	+	0.241	11.178
14	+	−	+	+	−	+	−	−	0.000	6.324
15	−	+	+	+	+	−	−	−	0.285	6.095
16	+	+	+	+	+	+	+	+	0.010	9.751

Table 5: Summary of Effects

Response 1: Range		Response 2: Consumption	
Factor	Effects	Factor	Effects
A	-0.266	C	3.911
A:B	-0.063	G	2.224
B:C	-0.054	A:B	0.908
G	0.051	H	0.570
B	0.048	A:E	0.553
B:D	0.032	C:D	0.400
D	-0.023	D	0.272
A:D	0.019	B:D	-0.249
H	-0.010	E	-0.235
F	-0.007	A	-0.198
C:D	-0.006	B	-0.193
A:E	0.005	A:D	-0.188
A:C	-0.003	B:C	-0.150
C	-0.001	A:C	-0.126
E	0.001	F	-0.087

A detailed overview of our data analysis can be found in Appendix B. Table 4 and 5 show the results of the simulation as well as a summary of the main effects that were retrieved after analyzing the regression model of the experiment. The main effects for both response

Figure 5: Half-Normal Plots for Range (left) and Consumption (right)

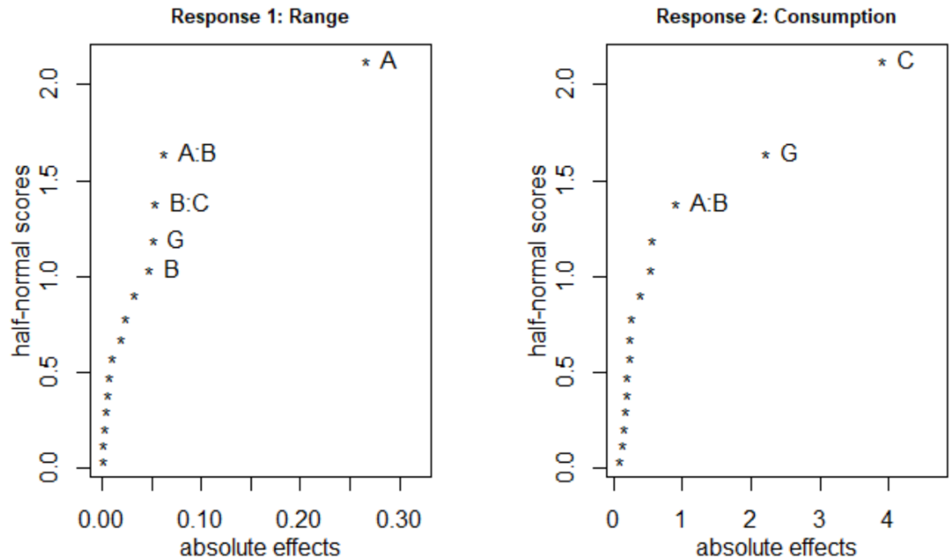
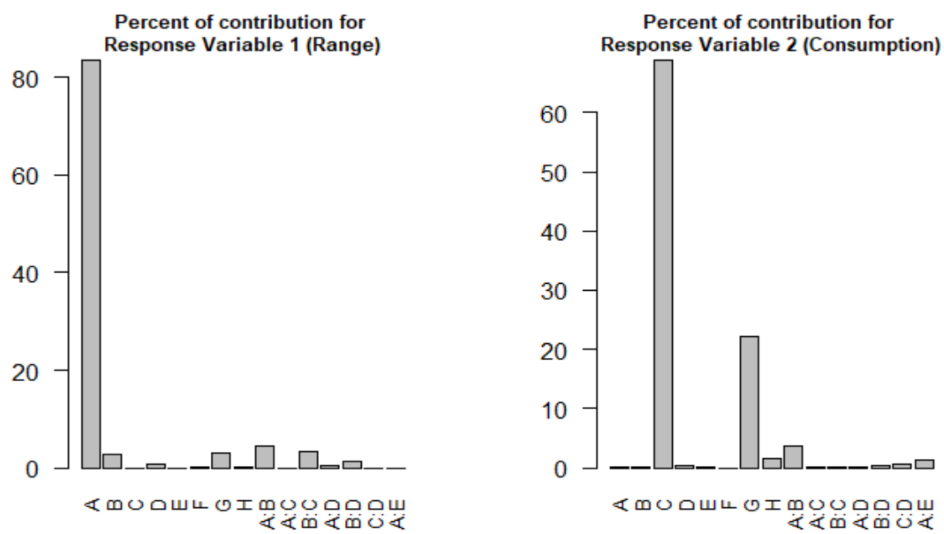


Figure 6: Contribution of Range (left) and Consumption (right)



variables are listed from largest to smallest in terms of absolute value. Additionally, a normal probability plot as well as barplots showing the percentage of contribution is presented in figures 5 and 6. From both the table and figures, we can observe that factor A, which is the vertical nozzle angle, is the only significant factor for the spray range of the sprinkler. On the other hand, factors C and G, the nozzle profile and entrance pressure, are both significant factors for water consumption. Additionally, both the plot and table indicate that the interaction between factors A and B is also significant. As we can see from Figure 1 in question 1, the interaction between A:B is aliased with C:G. Based on the assumptions of effect hierarchy, we can assume that the interaction between A and B indicated in our analysis is actually the interaction between C and G, which is the nozzle profile and entrance pressure.

Final Model

Based on the results found above, a final regression model was produced with factors A, C, and G. The results for both response variables can be found below in Table 6:

Table 6: Final Regression Model Results

	Response Variable 1: Range				Response Variable 2: Consumption			
	Estimate	Std. Error	T-value	Pr(>|t|)	Estimate	Std.Error	T value	Pr(>|t|)
Intercept	0.1494033	0.0129458	11.541	1.74e-07	6.04566	0.15817	38.223	4.75e-13
A	-0.1331465	0.0129458	-10.285	5.58e-07	-0.09918	0.15817	-0.627	0.5434
C	-0.0005295	0.0129458	-0.041	0.9681	1.95548	0.15817	12.363	8.56e-08
G	0.0256843	0.0129458	1.984	0.0728	1.11192	0.15817	7.030	2.18e-05
C:G	-0.0313919	0.0129458	-2.425	0.0337	0.45398	0.15817	2.870	0.0152

As we can observe from Table 6, most of the initial assumptions of the significant main effects hold: factor A is significant for range, and factors C, G, and the interaction between C and G is significant for consumption. It is interesting to note how under response variable 1, it appears that the interaction between C:G is also significant.

Residuals Analysis

Following these results, we then analyzed the residual plots. The results are shown in Figures 7 and 8 below:

Figure 7: Response Variable 1 (Range)

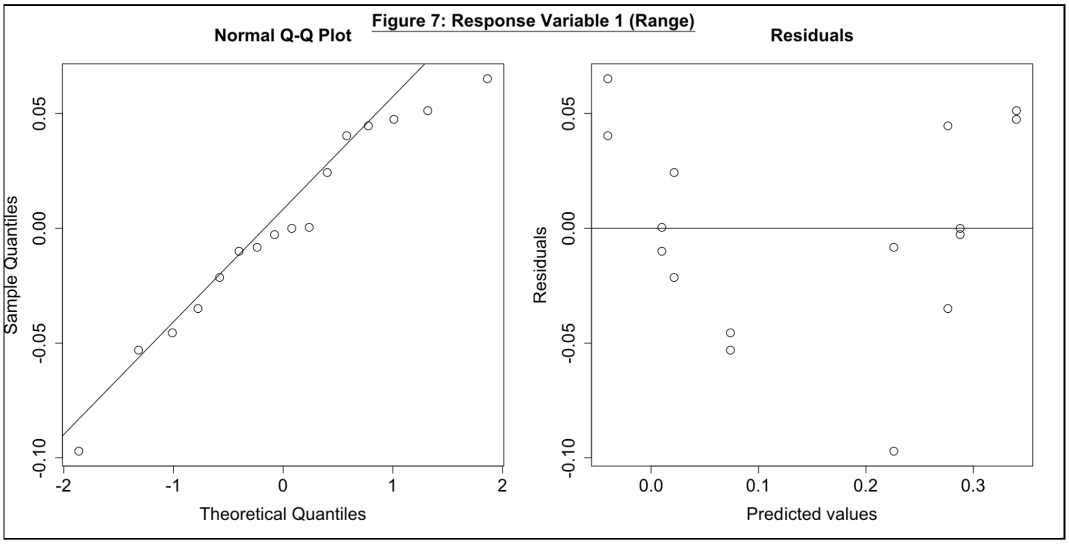
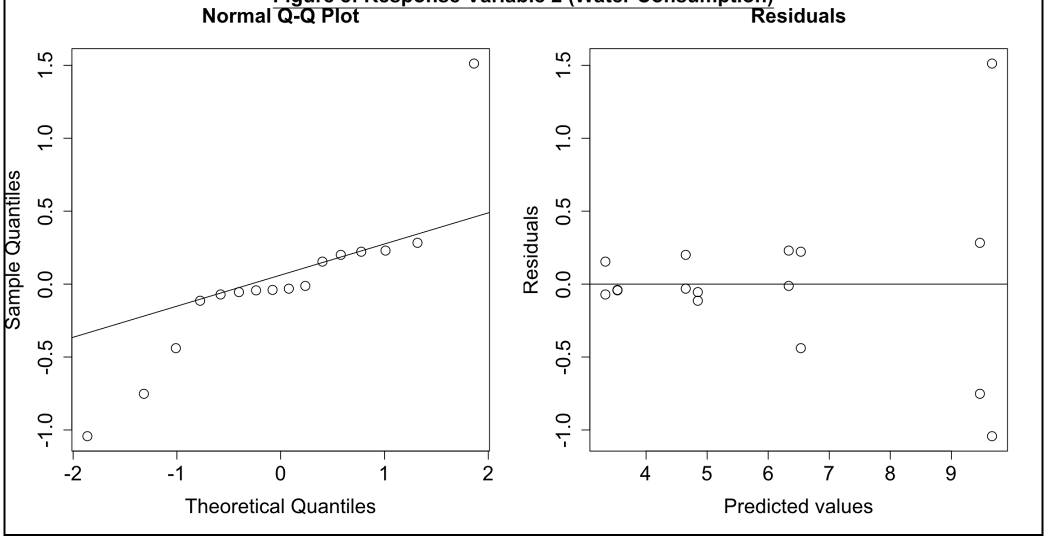


Figure 8: Response Variable 2 (Water Consumption)



The assumptions of normality and constant variance both appear to be violated in both response variables. This is something the research agency should consider when performing the experiments.

Question 6. What are the most influential factors?

As seen from the half-normal plots of Figure 5 and the summary of effects in Table 5, the most influential factor for range is the main effect of the vertical nozzle angle (alpha), and the most influential factors for consumption are the main effects of nozzle profile (A_q) and the entrance pressure (p_{in}), and the interaction between the vertical nozzle angle (alpha) and the tangential nozzle angle (beta). However, the interaction between alpha and beta is completely aliased with the interaction between A_q and p_{in}, and the assumption of effect heredity states that the interaction between two active main effects is also likely to be significant; thus, we assume that the significant interaction effect is between A_q and p_{in} rather than alpha and beta.

Question 7. Recommend the settings of the factors that optimize the water consumption and spray range simultaneously.

Table 7: Optimal Settings

Factor	Optimal Setting (N/A = Not significant)			
	Range (Maximizing)		Consumption (Minimizing)	
	Coded	Actual	Coded	Actual
A: Vertical Nozzle Angle	-1	0^o	N/A	N/a
C: Nozzle Profile	N/A	N/A	-1	0.000002 m^2
G: Entrance Pressure	N/A	N/A	-1	1 bar

Appendix C shows how we used the built in optim() function in R in order to identify the recommended settings for factors that optimize the water consumption and spray range simultaneously. A summary of our results can be found in table 7, as well as the interaction plots shown in Figures 9 and 10. As mentioned previously, the goal was to maximize the spray range and minimize the water consumption. The function was run twice, once with each response variable. Only the significant factors were considered in each algorithm when looking at the parameters. In this case, the algorithm indicates that to maximize range, the vertical nozzle angle should be at its minimum level, which is 0°. On the other hand, when it comes to minimizing consumption, factor C, or Nozzle profile, should be at the minimum level, and factor G, or Entrance Pressure, should also be at the minimum level. These correspond to 0.000002 m^2 and 1 bar for the two factors respectively.

Figure 9: Interaction Plot Matrix for Response 1 (Range)

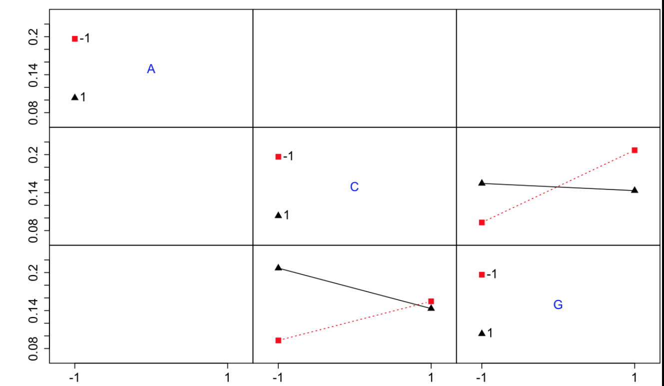
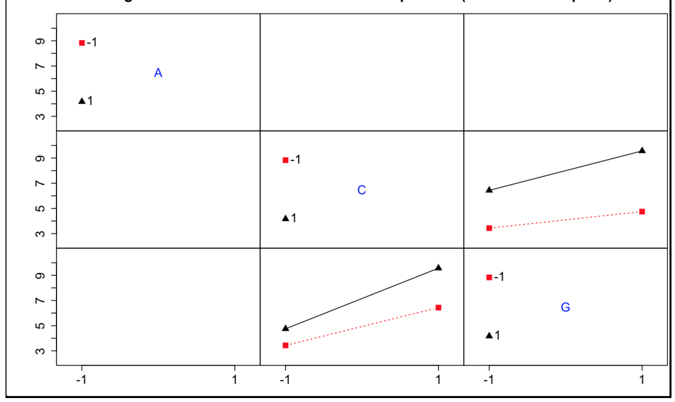


Figure 10: Interaction Plot Matrix for Response 2 (Water Consumption)



Question 8. Conduct confirmation experiments using your recommended settings. Are your predictions accurate?

Our confirmation experiments consisted of four runs, with factors A, C, and G at the following settings: A=1, C=-1, G=-1 to optimize consumption in one run, A=-1, C=-1, G=1 to optimize range in another run, and A=-1, C=-1, G=-1 to optimize both range and consumption simultaneously for another two runs. The other factors, which we had found to be insignificant, were set at their lowest settings. All four runs were randomized, and the predicted values for consumption and range were found using their respective separate models. The results obtained from the confirmation experiment are shown in Table 8.

Table 8: Confirmation Experiment Results and Predicted Values

A	B	C	D	E	F	G	H	Observed consumption	Predicted consumption	Observed range	Predicated range
1	-1	-1	-1	-1	-1	-1	-1	3.2037	3.3331	0.0188	-0.0403
-1	-1	-1	-1	-1	-1	-1	-1	3.2917	3.5314	0.2694	0.2260
-1	-1	-1	-1	-1	-1	1	-1	4.7967	4.8473	0.3665	0.3401
-1	-1	-1	-1	-1	-1	-1	-1	3.1929	3.5314	0.2615	0.2260

The predicted values are somewhat close to the actual observed values: the root mean squared error (RMSE) of prediction for the range was 0.04282091, and the root mean squared error of prediction for the consumption was 0.218731. However, it is important to note that the predicted value for range becomes negative under the recommended settings for minimum cost; this is physically impossible and indicates that the model used to optimize consumption is not suitable for predicting the range. Moreover, the runs that tested the recommended settings for the simultaneous optimization of both consumption and range resulted in values that were suboptimal in both response variables, suggesting that there is a tradeoff between maximizing range and minimizing water consumption.

Conclusions and Recommendations

To simultaneously maximize spray range and minimize water consumption, we recommend using a vertical nozzle angle of 0^o, a tangential nozzle angle of 0^o, a nozzle profile of 0.000002 m^2, a sprinkler head diameter of 0.10, a static friction moment of 0.01 nm, a dynamic friction moment of 0.01 nm/s, an entrance pressure of 1 bar, and a diameter flow line of 5 mm.

While our experiment was able to identify that the main effects of vertical nozzle angle, nozzle profile, and entrance pressure were significant to spray range and water consumption, some ambiguities still remain in terms of the true interaction effects, as the two-factor interaction effects are fully aliased in our design. As a result, follow-up experiments are recommended in order to resolve these ambiguities: we propose a 2^3 factorial design with vertical nozzle angle, nozzle profile, and entrance pressure as the factors under study to determine if the true interaction effect is from the interaction between nozzle profile and entrance pressure. Moreover, although our models for range and consumption were able to predict the spray range and water consumption with a RMSE of 0.0428 and 0.219 respectively, both models appear to violate the assumptions of normality and constant variance. The parabolic pattern of the residuals from the range model indicates that future experiments should consider three levels instead of two in order to account for non-linear relationships between the factors, and the near-symmetric funnel pattern of the residuals from the consumption model indicates that there may be another factor that was significant but not taken into account. These deficiencies should be taken into account when considering our model and when designing future experiments.

References

Montgomery, D.C. (2012). *Design and analysis of experiments* (8th ed.). John Wiley and Sons.

Statement of Contribution

[redacted]

Appendix

Contents

Appendix A

Data frame for 2^{8-4}_{IV} Design

```
A <- rep(c(-1, 1), 8)
B <- rep(c(-1, 1), each = 2, times = 4)
C <- rep(c(-1, 1), each = 4, times = 2)
D <- rep(c(-1, 1), each = 8, times = 1)
E <- B * C * D
F <- A * C * D
G <- A * B * C
H <- A * B * D
design <- data.frame(A, B, C, D, E, F, G, H)
cat("Generators: E = BCD, F = ACD, G = ABC, H = ABD \n")
```

```
## Generators: E = BCD, F = ACD, G = ABC, H = ABD
```

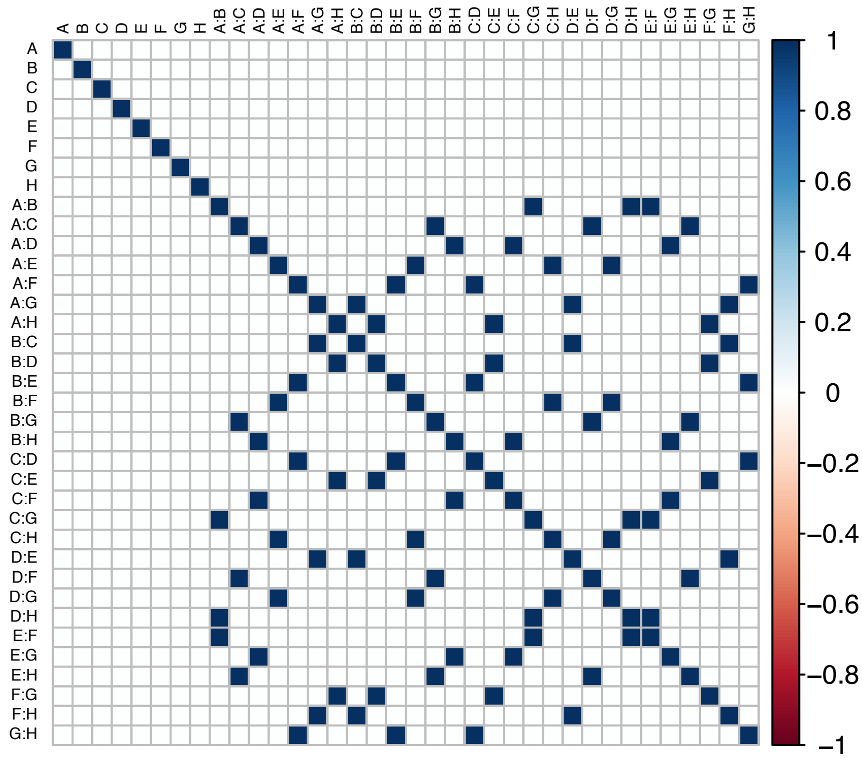
```
design
```

```
##     A  B  C  D  E  F  G  H
## 1  -1 -1 -1 -1 -1 -1 -1 -1
## 2   1 -1 -1 -1 -1  1  1  1
## 3  -1  1 -1 -1  1 -1  1  1
## 4   1  1 -1 -1  1  1 -1 -1
## 5  -1 -1  1 -1  1  1  1 -1
## 6   1 -1  1 -1  1 -1 -1  1
## 7  -1  1  1 -1 -1  1 -1  1
## 8   1  1  1 -1 -1 -1  1 -1
## 9  -1 -1 -1  1  1  1 -1  1
## 10  1 -1 -1  1  1 -1  1 -1
## 11 -1  1 -1  1 -1  1  1 -1
## 12  1  1 -1  1 -1 -1 -1  1
## 13 -1 -1  1  1 -1 -1  1  1
## 14  1 -1  1  1 -1  1 -1 -1
## 15 -1  1  1  1  1 -1 -1 -1
## 16  1  1  1  1  1  1  1  1
```

Alias Structure for 2^{8-4}_{IV} Design

```
## library(corrplot)
design_matrix <- design[,c("A", "B", "C", "D", "E", "F", "G", "H")]
X <- model.matrix(~(A + B + C + D + E + F + G + H)^2-1, design_matrix)
library(corrplot)
contrast.vectors.correlations <- cor(X)
par(mfrow = c(1,1))
corrplot(contrast.vectors.correlations, type = "full", addgrid.col = "gray",
         tl.col = "black", tl.srt = 90, method = "color",mar=c(0,0,3,0), tl.cex = 0.5)
mtext(expression('Figure 1: Alias Structure for 2'^{8-4}*''[IV]*' Design'),
      side = 3,line = -3, outer = TRUE, cex = 1)
```

Figure 1: Alias Structure for 2^{8-4}_{IV} Design



Alternate Considerations: Plackett-Burman Design and Optimal Screening

Dataframe for Plackett-Burman Design

```
X1 <- c(1, 1, -1, 1, 1, 1, -1, -1, -1, 1, -1, -1)
X2 <- c(X1[2:length(X1)], X1[1])
X3 <- c(X1[3:length(X1)], X1[1:2])
X4 <- c(X1[4:length(X1)], X1[1:3])
X5 <- c(X1[5:length(X1)], X1[1:4])
X6 <- c(X1[6:length(X1)], X1[1:5])
X7 <- c(X1[7:length(X1)], X1[1:6])
X8 <- c(X1[8:length(X1)], X1[1:7])
design_alt1 <- data.frame(X1, X2, X3, X4, X5, X6, X7, X8)
design_alt1
```

```
##    X1 X2 X3 X4 X5 X6 X7 X8
## 1   1  1 -1  1  1  1 -1 -1
## 2   1 -1  1  1  1 -1 -1 -1
## 3  -1  1  1  1 -1 -1 -1  1
## 4   1  1  1 -1 -1 -1  1 -1
## 5   1  1 -1 -1 -1  1 -1 -1
## 6   1 -1 -1 -1  1 -1 -1  1
## 7  -1 -1 -1  1 -1 -1  1  1
## 8  -1 -1  1 -1 -1  1  1 -1
```

```
## 9  -1  1 -1 -1  1  1 -1  1
## 10  1 -1 -1  1  1 -1  1  1
## 11 -1 -1  1  1 -1  1  1  1
## 12 -1  1  1 -1  1  1  1 -1
```

```
X <- model.matrix(~(X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8)^2-1, data.frame(design_alt1))
contrast.vectors.correlations.alt_1 <- cor(X)
```

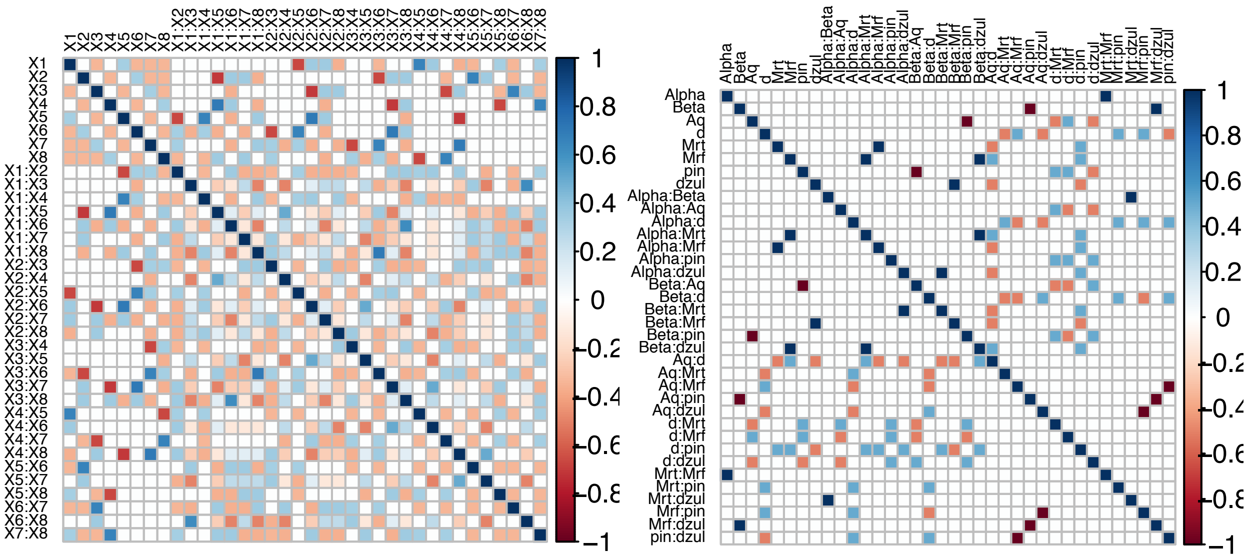
Dataframe for Optimal Screening

```
Alpha   <- c(0, 90)
Beta    <- c(0, 90)
Aq  <- c(0.000002, 0.000004)
d   <- c(0.10,  0.20)
Mrt <- c(0.01, 0.02)
Mrf <- c(0.01,  0.02)
pin <- c(1, 2)
dzul <- c(5, 10)
## library(AlgDesign)
candidate.set <- gen.factorial(levels=2, nVars = 8,
                               varNames = c("Alpha","Beta","Aq",
                                            "d", "Mrt", "Mrf", "pin", "dzul"))

opt.design <- optFederov(~Alpha + Beta + Aq + d + Mrt + Mrf + pin + dzul,
                         candidate.set, nTrials = 16, nRepeats = 1000)
D.opt <- opt.design$design
X.opt <- model.matrix(~(Alpha + Beta + Aq + d + Mrt + Mrf + pin + dzul)^2-1,
                      data.frame(D.opt))
contrast.vectors.correlations.opt <- cor(X.opt)
```

Corrplots

```
par(mfrow = c(1, 2))
corrplot(contrast.vectors.correlations.alt_1, type = "full",
         tl.col = "black", tl.srt = 90, method = "color",
         addgrid.col = "gray", tl.cex = 0.5)
corrplot(contrast.vectors.correlations.opt, type = "full", addgrid.col = "gray",
         tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.5)
mtext("Figure 2: Alias Structure for Plackett-Burman Design (Left)
      and Optimal Screening Design (Right)",
      side = 3,line = -3, outer = TRUE,
      cex = 1)
```

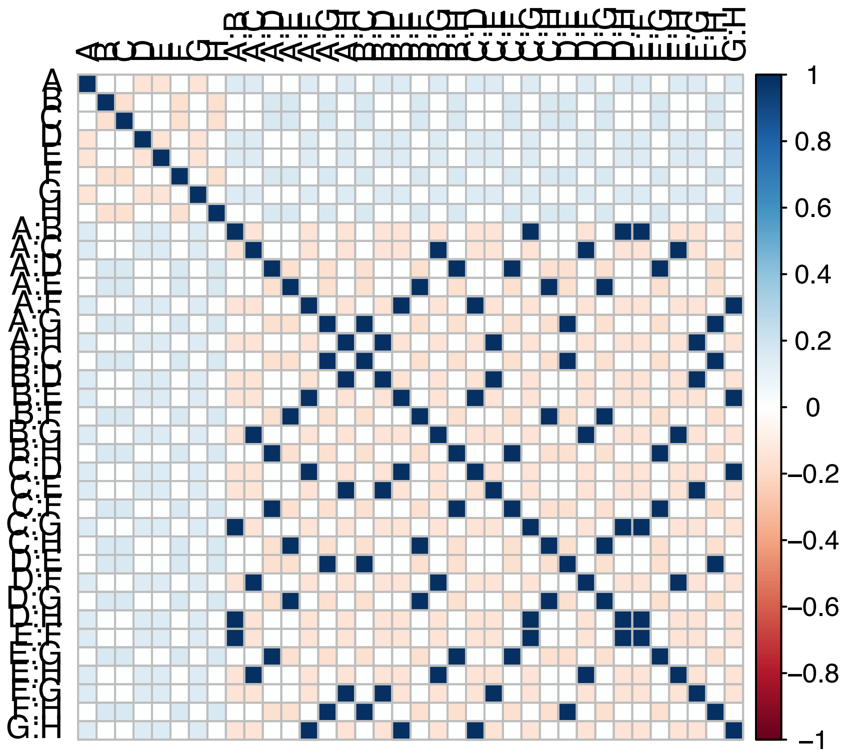
Figure 2: Alias Structure for Plackett–Burman Design (Left)
and Optimal Screening Design (Right)



Question 3: Failed Tests

```
design_missing <- design
design_missing[10,] <- rep(NA, 8)
design_missing[1,] <- rep(NA, 8)
design.2 <- design_missing[,c("A", "B", "C", "D", "E", "F", "G", "H")]
X.2 <- model.matrix(~(A + B + C + D + E + F + G + H)^2-1, design.2)
contrast.vectors.correlations.2 <- cor(X.2)
par(mfrow = c(1,1))
corrplot(contrast.vectors.correlations.2, type = "full", addgrid.col = "gray",
         tl.col = "black", tl.srt = 90, method = "color",mar=c(0,0,3,0))
mtext(expression(
  'Figure 3: Alias Structure for 2'^{8-4}*''[IV]*' Design with Missing Values'),
      side = 3,line = -3, outer = TRUE, cex = 1)
```

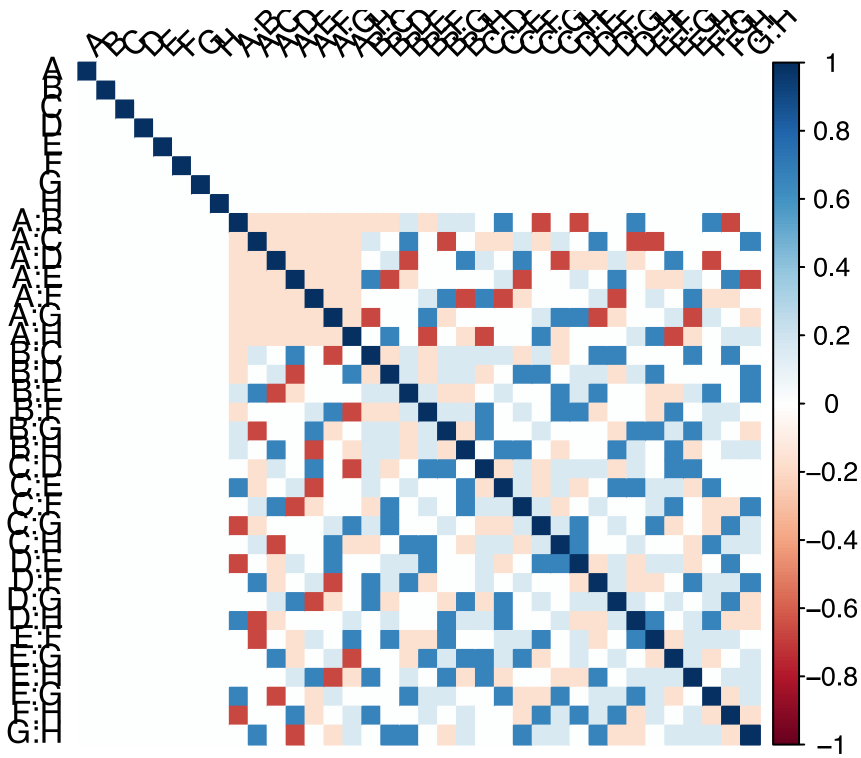
Figure 3: Alias Structure for 2^{8-4}_{IV} Design with Missing Values



Question 4: Alternative Experimental Design

```r
# alternative experimental design

A <- c(-1, 0, -1, -1, 1, -1, 1, -1, -1, 1, 1, 1, -1, 1, 0, 0, 1)
B <- c(1, 0, 1, -1, -1, 0, 1, -1, -1, -1, 1, 1, 1, -1, -1, 1, 0)
C <- c(-1, 0, 1, 1, -1, -1, -1, 1, -1, 1, 1, -1, 0, 0, -1, 1, 1)
D <- c(1, 0, 0, -1, 0, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1)
E <- c(1, 0, -1, 1, 1, 1, 0, 0, -1, -1, 1, -1, -1, 1, -1, 1, -1)
F <- c(0, 0, -1, 1, 1, -1, 1, -1, 1, 0, -1, -1, 1, -1, -1, 1, 1)
G <- c(-1, 0, 1, 0, -1, 1, 1, -1, 1, 1, -1, 0, -1, 1, -1, 1, -1)
H <- c(-1, 0, -1, -1, 1, 1, -1, 1, 0, 1, 0, 1, 1, -1, -1, 1, -1)

design <- matrix(c(A, B, C, D, E, F, G, H), ncol=8)

X.alt <- model.matrix(~(A + B + C + D + E + F + G + H)^2-1, data.frame(design))

corrplot(cor(X.alt), type = "full",
         tl.col = "black", tl.srt = 45, method = "color",
         main="Figure 4: Alias Structure for the Alternative Experimental Design",
         mar=c(0,0,3,0))
```

Figure 4: Alias Structure for the Alternative Experimental Design



Appendix B

Retrieving the results

Creating a data frame with the actual values:

Coded Data Frame:

```
A <- rep(c(-1, 1), 8)
B <- rep(c(-1, 1), each = 2, times = 4)
C <- rep(c(-1, 1), each = 4, times = 2)
D <- rep(c(-1, 1), each = 8, times = 1)
E <- B * C * D
F <- A * C * D
G <- A * B * C
H <- A * B * D
design <- data.frame(A, B, C, D, E, F, G, H)
cat("Generators: E = BCD, F = ACD, G = ABC, H = ABD \n")
```

```
## Generators: E = BCD, F = ACD, G = ABC, H = ABD
```

```
design
```

```
##     A  B  C  D  E  F  G  H
## 1  -1 -1 -1 -1 -1 -1 -1 -1
## 2   1 -1 -1 -1 -1  1  1  1
## 3  -1  1 -1 -1  1 -1  1  1
## 4   1  1 -1 -1  1  1 -1 -1
## 5  -1 -1  1 -1  1  1  1 -1
## 6   1 -1  1 -1  1 -1 -1  1
## 7  -1  1  1 -1 -1  1 -1  1
## 8   1  1  1 -1 -1 -1  1 -1
## 9  -1 -1 -1  1  1  1 -1  1
## 10  1 -1 -1  1  1 -1  1 -1
## 11 -1  1 -1  1 -1  1  1 -1
## 12  1  1 -1  1 -1 -1 -1  1
## 13 -1 -1  1  1 -1 -1  1  1
## 14  1 -1  1  1 -1  1 -1 -1
## 15 -1  1  1  1  1 -1 -1 -1
## 16  1  1  1  1  1  1  1  1
```

Actual Values:

```
Alpha   <- c(0, 90)
Beta    <- c(0, 90)
Aq  <- c(0.000002, 0.000004)
d   <- c(0.10,  0.20)
Mrt <- c(0.01, 0.02)
Mrf <- c(0.01,  0.02)
pin <- c(1, 2)
dzul <- c(5, 10)
design_actual <- data.frame(design)
```

```r
design_actual$A <- ifelse(design_actual$A == -1, Alpha[1], Alpha[2])
design_actual$B <- ifelse(design_actual$B == -1, Beta[1], Beta[2])
design_actual$C <- ifelse(design_actual$C == -1, Aq[1], Aq[2])
design_actual$D <- ifelse(design_actual$D == -1, d[1], d[2])
design_actual$E <- ifelse(design_actual$E == -1, Mrt[1], Mrt[2])
design_actual$F <- ifelse(design_actual$F == -1, Mrf[1], Mrf[2])
design_actual$G <- ifelse(design_actual$G == -1, pin[1], pin[2])
design_actual$H <- ifelse(design_actual$H == -1, dzul[1], dzul[2])
design_actual
```

```
##     A  B      C   D    E    F G  H
## 1   0  0 2e-06 0.1 0.01 0.01 1  5
## 2  90  0 2e-06 0.1 0.01 0.02 2 10
## 3   0 90 2e-06 0.1 0.02 0.01 2 10
## 4  90 90 2e-06 0.1 0.02 0.02 1  5
## 5   0  0 4e-06 0.1 0.02 0.02 2  5
## 6  90  0 4e-06 0.1 0.02 0.01 1 10
## 7   0 90 4e-06 0.1 0.01 0.02 1 10
## 8  90 90 4e-06 0.1 0.01 0.01 2  5
## 9   0  0 2e-06 0.2 0.02 0.02 1 10
## 10 90  0 2e-06 0.2 0.02 0.01 2  5
## 11  0 90 2e-06 0.2 0.01 0.02 2  5
## 12 90 90 2e-06 0.2 0.01 0.01 1 10
## 13  0  0 4e-06 0.2 0.01 0.01 2 10
## 14 90  0 4e-06 0.2 0.01 0.02 1  5
## 15  0 90 4e-06 0.2 0.02 0.01 1  5
## 16 90 90 4e-06 0.2 0.02 0.02 2 10
```

Uploading the data into the simulation and reading the results:

```r
write.table(design_actual, "design_actual.txt", sep="\t", quote=FALSE, dec=".",
            row.names=FALSE)
result <- read.delim("result.txt", header = TRUE, sep = ",")
result
```

```
##    alpha beta    Aq   d   mt   mf pin dzul consumption      range
## 1      0    0 2e-06 0.1 0.01 0.01   1    5    3.488302 0.21768389
## 2     90    0 2e-06 0.1 0.01 0.02   2   10    4.849373 0.02831968
## 3      0   90 2e-06 0.1 0.02 0.01   2   10    4.792286 0.38757945
## 4     90   90 2e-06 0.1 0.02 0.02   1    5    3.487137 0.00000000
## 5      0    0 4e-06 0.1 0.02 0.02   2    5    8.623523 0.32090498
## 6     90    0 4e-06 0.1 0.02 0.01   1   10    6.565773 0.04572039
## 7      0   90 4e-06 0.1 0.01 0.02   1   10    6.756439 0.28765625
## 8     90   90 4e-06 0.1 0.01 0.01   2    5    8.715883 0.00000000
## 9      0    0 2e-06 0.2 0.02 0.02   1   10    3.491602 0.12890120
## 10    90    0 2e-06 0.2 0.02 0.01   2    5    4.617540 0.02076775
## 11     0   90 2e-06 0.2 0.01 0.02   2    5    4.733259 0.39136924
## 12    90   90 2e-06 0.2 0.01 0.01   1   10    3.261907 0.02484154
## 13     0    0 4e-06 0.2 0.01 0.01   2   10   11.178094 0.24135439
## 14    90    0 4e-06 0.2 0.01 0.02   1    5    6.323551 0.00000000
## 15     0   90 4e-06 0.2 0.02 0.01   1    5    6.095192 0.28494879
## 16    90   90 4e-06 0.2 0.02 0.02   2   10    9.750682 0.01040519
```

Inserting results into the coded data frame:

```
design$Y_1 <- result$range
design$Y_2 <- result$consumption
design
```

```
##     A  B  C  D  E  F  G  H        Y_1       Y_2
## 1  -1 -1 -1 -1 -1 -1 -1 -1 0.21768389  3.488302
## 2   1 -1 -1 -1 -1  1  1  1 0.02831968  4.849373
## 3  -1  1 -1 -1  1 -1  1  1 0.38757945  4.792286
## 4   1  1 -1 -1  1  1 -1 -1 0.00000000  3.487137
## 5  -1 -1  1 -1  1  1  1 -1 0.32090498  8.623523
## 6   1 -1  1 -1  1 -1 -1  1 0.04572039  6.565773
## 7  -1  1  1 -1 -1  1 -1  1 0.28765625  6.756439
## 8   1  1  1 -1 -1 -1  1 -1 0.00000000  8.715883
## 9  -1 -1 -1  1  1  1 -1  1 0.12890120  3.491602
## 10  1 -1 -1  1  1 -1  1 -1 0.02076775  4.617540
## 11 -1  1 -1  1 -1  1  1 -1 0.39136924  4.733259
## 12  1  1 -1  1 -1 -1 -1  1 0.02484154  3.261907
## 13 -1 -1  1  1 -1 -1  1  1 0.24135439 11.178094
## 14  1 -1  1  1 -1  1 -1 -1 0.00000000  6.323551
## 15 -1  1  1  1  1 -1 -1 -1 0.28494879  6.095192
## 16  1  1  1  1  1  1  1  1 0.01040519  9.750682
```

Data Analysis

Linear Regression Model and Effects

```
m1 <- lm(Y_1 ~ A * B * C * D * E * F * G * H, data = design)
#summary(m1)
m2 <- lm(Y_2 ~ A * B * C * D * E * F * G * H, data = design)
#summary(m2)
effects_1 <- 2*(coef(m1)[-1])
cat("Estimated effects: \n")
```

```
## Estimated effects:
```

```
effects_1[order(abs(effects_1[which(!is.na(effects_1))]), decreasing = TRUE)]
```

```
##            A          A:B          B:C            G            B          B:D
## -0.266292955 -0.062783796 -0.054135904  0.051368576  0.047893523  0.032241831
##            D          A:D            H            F          C:D          A:E
## -0.023159570  0.018653171 -0.010112069 -0.006917457 -0.006233744  0.004932682
##          A:C            C            E
## -0.003391752 -0.001059096  0.001000343
```

```
effects_2 <- 2*(coef(m2)[-1])
cat("Estimated effects: \n")
```

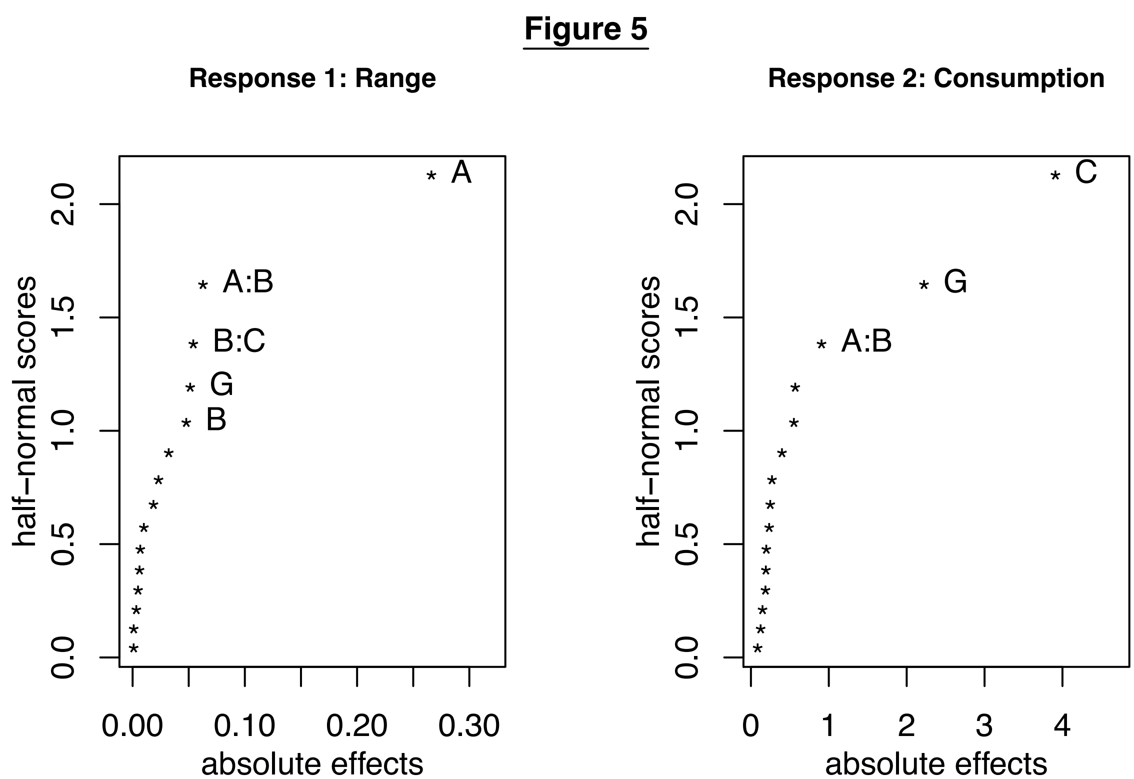
```
## Estimated effects:
```

```r
effects_2[order(abs(effects_2[which(!is.na(effects_2))]), decreasing = TRUE)]
```

```
##           C           G         A:B           H         A:E         C:D
##  3.91096608  2.22384215  0.90796458  0.57022119  0.55298875  0.39983640
##           D         B:D           E           A           B         A:D
##  0.27163888 -0.24931486 -0.23538409 -0.19835632 -0.19312141 -0.18776036
##         B:C         A:C           F
## -0.15006460 -0.12598344 -0.08742632
```

Daniel Plot

```r
# library(FrF2)
par(mfrow=c(1,2))
DanielPlot(m1, half = T, main = "Response 1: Range", cex.main = 0.8)
DanielPlot(m2, half = T,main = "Response 2: Consumption", cex.main = 0.8)
mtext(expression(bold(paste(underline("Figure 5")))),
      side = 3,line = -1.5, outer = TRUE,
      cex = 1)
```



Barplots Showing Percentage Contribution

```
m1_anova <- anova(m1)
```

```
## Warning in anova.lm(m1): ANOVA F-tests on an essentially perfect fit are
## unreliable
```

```
cat('\n ANOVA summary table: \n')
```

```
##
##  ANOVA summary table:
```

```
print(m1_anova)
```

```
## Analysis of Variance Table
##
## Response: Y_1
##           Df   Sum Sq  Mean Sq F value Pr(>F)
## A          1 0.283648 0.283648
## B          1 0.009175 0.009175
## C          1 0.000004 0.000004
## D          1 0.002145 0.002145
## E          1 0.000004 0.000004
## F          1 0.000191 0.000191
## G          1 0.010555 0.010555
## H          1 0.000409 0.000409
## A:B        1 0.015767 0.015767
## A:C        1 0.000046 0.000046
## B:C        1 0.011723 0.011723
## A:D        1 0.001392 0.001392
## B:D        1 0.004158 0.004158
## C:D        1 0.000155 0.000155
## A:E        1 0.000097 0.000097
## Residuals  0 0.000000
```

```
total.sums.of.squares_1 <- sum(m1_anova$`Sum Sq`)
percent.of.contribution_1 <- (m1_anova$`Sum Sq`/(total.sums.of.squares_1))*100
effect_names_1 <- names(effects_1[which(!is.na(effects_1))])
neffects_1 <- length(effect_names_1)
m2_anova <- anova(m2)
```

```
## Warning in anova.lm(m2): ANOVA F-tests on an essentially perfect fit are
## unreliable
```

```
cat('\n ANOVA summary table: \n')
```

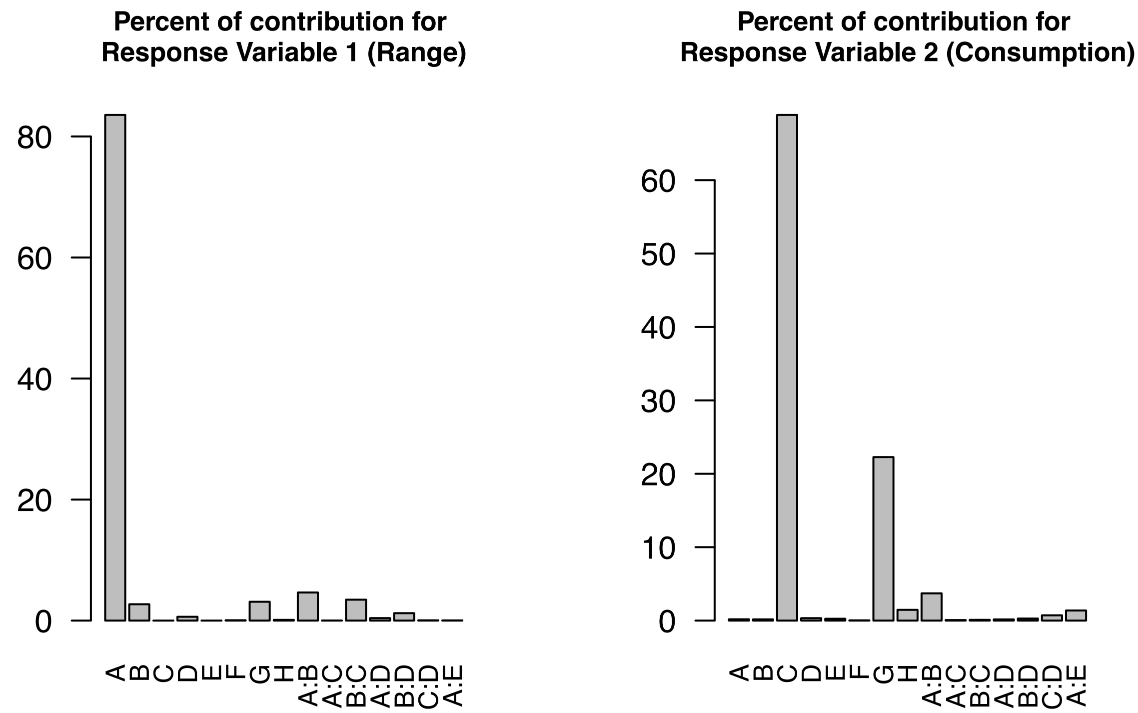
```
##
##  ANOVA summary table:
```

```r
print(m2_anova)
```

```
## Analysis of Variance Table
##
## Response: Y_2
##           Df Sum Sq Mean Sq F value Pr(>F)
## A          1  0.157   0.157
## B          1  0.149   0.149
## C          1 61.183  61.183
## D          1  0.295   0.295
## E          1  0.222   0.222
## F          1  0.031   0.031
## G          1 19.782  19.782
## H          1  1.301   1.301
## A:B        1  3.298   3.298
## A:C        1  0.063   0.063
## B:C        1  0.090   0.090
## A:D        1  0.141   0.141
## B:D        1  0.249   0.249
## C:D        1  0.639   0.639
## A:E        1  1.223   1.223
## Residuals  0  0.000
```

```r
total.sums.of.squares_2 <- sum(m2_anova$`Sum Sq`)
percent.of.contribution_2 <- (m2_anova$`Sum Sq`/(total.sums.of.squares_2))*100
effect_names_2 <- names(effects_2[which(!is.na(effects_2))])
neffects_2 <- length(effect_names_2)
par(mfrow = c(1,2))
barplot(height=percent.of.contribution_1[-length(percent.of.contribution_1)],
        names=effect_names_1,
        main = "Percent of contribution for \nResponse Variable 1 (Range)",
        las=2,cex.names=0.8, cex.main = 0.8)
barplot(height=percent.of.contribution_2[-length(percent.of.contribution_2)],
        names=effect_names_2,
        main = "Percent of contribution for \nResponse Variable 2 (Consumption)",
        las=2,cex.names=0.8, cex.main = 0.8)
mtext(expression(bold(paste(underline("Figure 6")))),
      side = 3,line = -1.2, outer = TRUE,
      cex = 1)
```

Figure 6



It is important to note that the interaction A:B is aliased with C:G.

Final Model

Regression Model

```
final_model_1 <- lm(Y_1 ~ A + C + G +C:G, data = design)
summary(final_model_1)
```

```
##
## Call:
## lm.default(formula = Y_1 ~ A + C + G + C:G, data = design)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.097102 -0.024816 -0.001425  0.041365  0.065131
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.1494033  0.0129458  11.541 1.74e-07 ***
## A           -0.1331465  0.0129458 -10.285 5.58e-07 ***
## C           -0.0005295  0.0129458  -0.041   0.9681
## G            0.0256843  0.0129458   1.984   0.0728 .
## C:G         -0.0313919  0.0129458  -2.425   0.0337 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05178 on 11 degrees of freedom
## Multiple R-squared:  0.9131, Adjusted R-squared:  0.8815
## F-statistic:  28.9 on 4 and 11 DF,  p-value: 8.792e-06
```

```
final_model_2 <- lm(Y_2 ~ A + C + G +C:G, data = design)
summary(final_model_2)
```

```
##
## Call:
## lm.default(formula = Y_2 ~ A + C + G + C:G, data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04270 -0.08187 -0.03560  0.20583  1.51187
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.04566    0.15817  38.223 4.75e-13 ***
## A           -0.09918    0.15817  -0.627   0.5434
## C            1.95548    0.15817  12.363 8.56e-08 ***
## G            1.11192    0.15817   7.030 2.18e-05 ***
## C:G          0.45398    0.15817   2.870   0.0152 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6327 on 11 degrees of freedom
## Multiple R-squared:  0.9504, Adjusted R-squared:  0.9324
## F-statistic: 52.73 on 4 and 11 DF,  p-value: 4.15e-07
```

Effects

```
effects_final_1 <- 2*(coef(final_model_1)[-1])
cat("Estimated effects: \n")
```

```
## Estimated effects:
```

```
effects_final_1[order(abs(effects_final_1), decreasing = TRUE)]
```

```
##            A          C:G            G            C
## -0.266292955 -0.062783796  0.051368576 -0.001059096
```

```
effects_final_2 <- 2*(coef(final_model_2)[-1])
cat("Estimated effects: \n")
```

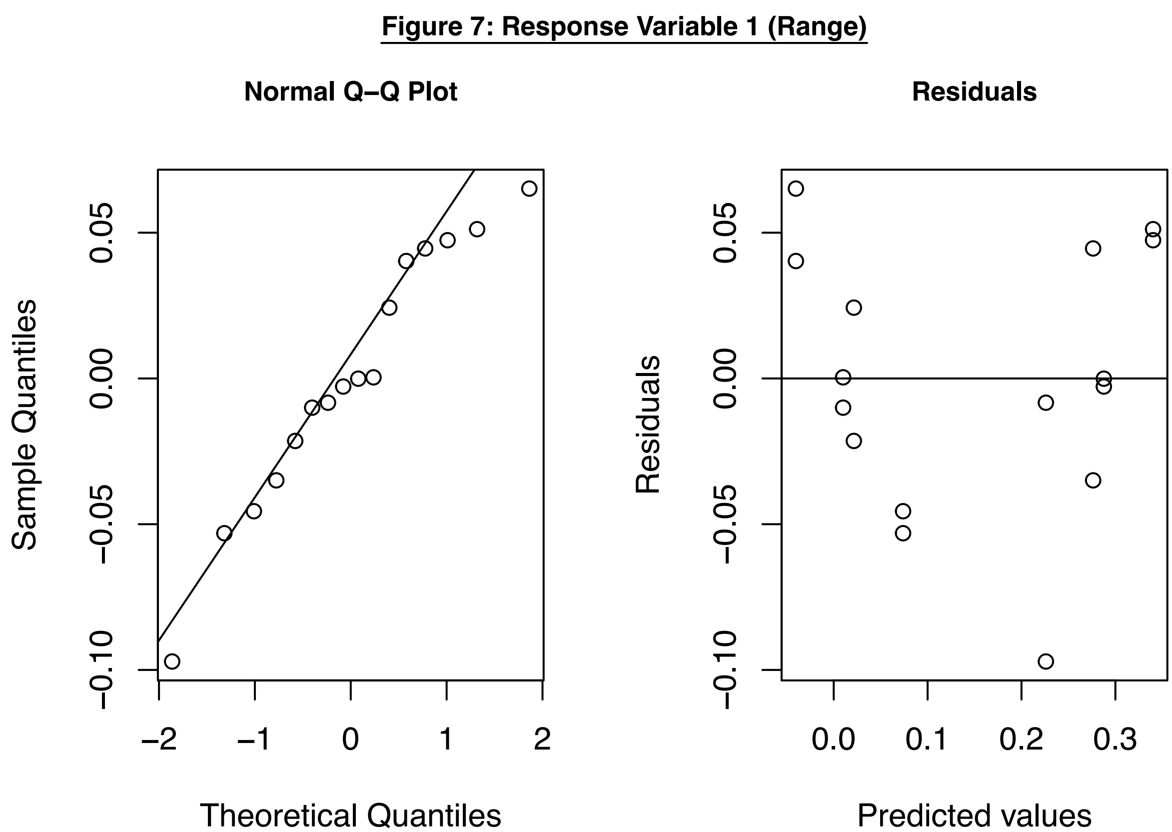
```
## Estimated effects:
```

```
effects_final_2[order(abs(effects_final_2), decreasing = TRUE)]
```

```
##          C           G         C:G           A
##  3.9109661   2.2238421   0.9079646 -0.1983563
```

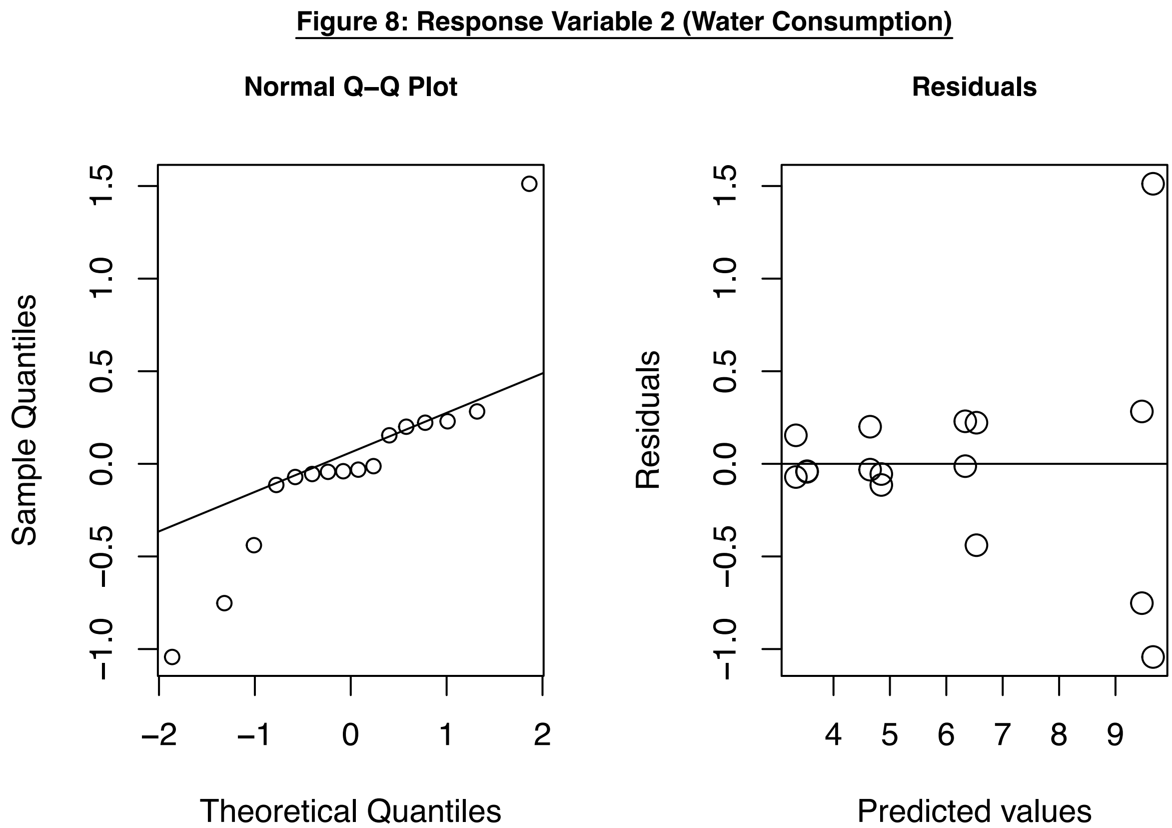
Residual Plots

```
# for range
res_1 <- final_model_1$residuals
par(mfrow=c(1,2))
qqnorm(res_1, cex.main = 0.8);
qqline(res_1)
plot(final_model_1$fitted.values, res_1, xlab = 'Predicted values',
     ylab = 'Residuals', main = "Residuals", cex.main = 0.8)
abline(h = 0)
mtext(expression(bold(paste(underline("Figure 7: Response Variable 1 (Range)")))),
      side = 3,line = -1, outer = TRUE,
      cex = 0.8)
```




```
# for consumption
res_2 <- final_model_2$residuals
par(mfrow=c(1,2))
qqnorm(res_2, cex.main = 0.8);
qqline(res_2)
plot(final_model_2$fitted.values, res_2, xlab = 'Predicted values',
     ylab = 'Residuals', main = "Residuals", cex = 1.5, cex.main = 0.8)
abline(h = 0)
```

```
mtext(expression(bold(paste(underline("Figure 8: Response Variable 2 (Water Consumption)")))),
      side = 3,line = -1, outer = TRUE,
      cex = 0.8)
```



Appendix C

Optim() Function

For Response Variable 1 (Range)

```
# x[1]=A, x[2]=C, x[3]=G
obj_func_1<- function(x){
    pred.y <- final_model_1$coefficients[1] + final_model_1$coefficients["A"] * x[1] +
      final_model_1$coefficients["C"] * x[2] +
      final_model_1$coefficients["G"] * x[3] +
      final_model_1$coefficients["C:G"] * x[2] * x[3]
    return(-1*pred.y) # Because the 'optim' function minimizes.
}
optim(par = c(0, 0, 0), fn = obj_func_1, lower = -1, upper = 1, method = "L-BFGS-B")
```

```
## $par
## [1] -1 -1  1
##
## $value
## [1] -0.3401555
##
## $counts
## function gradient
##       13       13
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: NORM OF PROJECTED GRADIENT <= PGTOL"
```

For Response Variable 2 (Water Consumption)

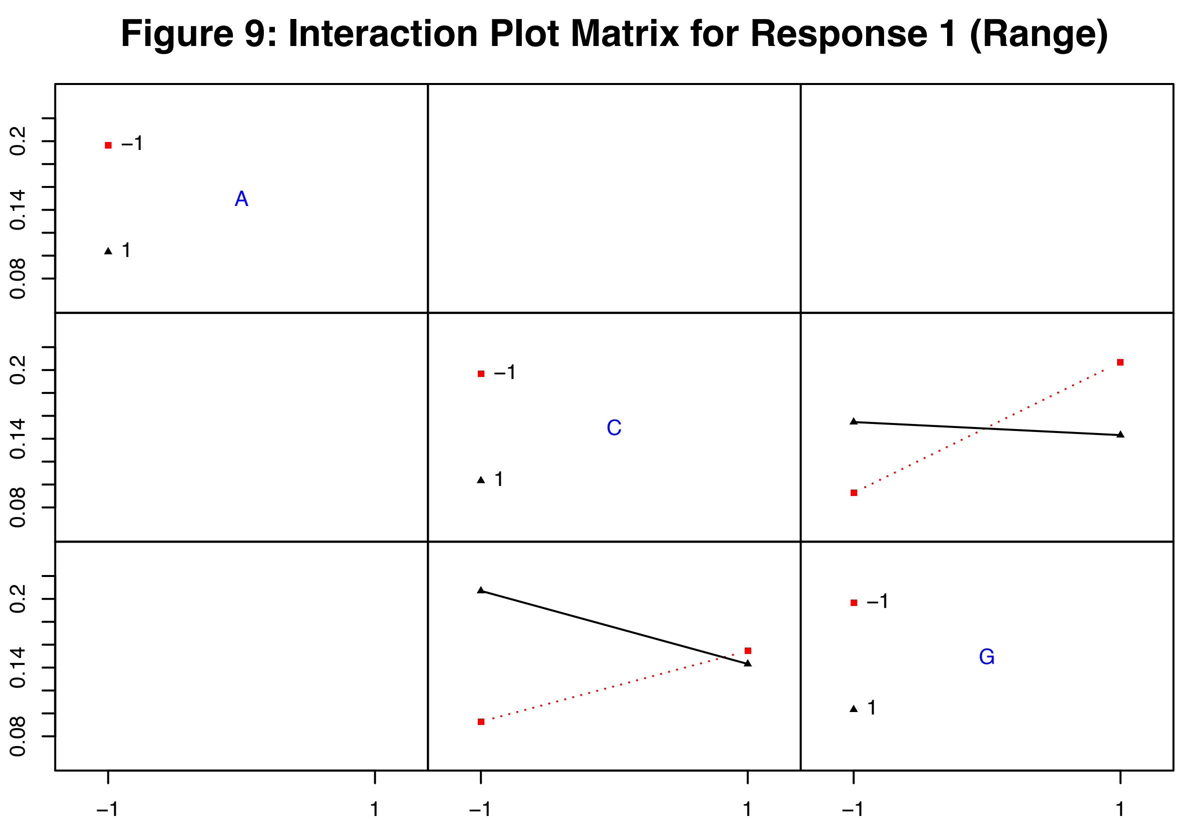
```
# x[1]=A, x[2]=C, x[3]=G
obj_func_2<- function(x){
    pred.y <- final_model_2$coefficients[1] + final_model_2$coefficients["A"] * x[1] +
      final_model_2$coefficients["C"] * x[2] +
      final_model_2$coefficients["G"] * x[3] +
      final_model_2$coefficients["C:G"] * x[2] * x[3]
    return(1*pred.y) # Because the 'optim' function minimizes.
}
optim(par = c(0, 0, 0), fn = obj_func_2, lower = -1, upper = 1, method = "L-BFGS-B")
```

```
## $par
## [1]  1 -1 -1
##
## $value
## [1] 3.333059
##
```

```
## $counts
## function gradient
##        4        4
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: NORM OF PROJECTED GRADIENT <= PGTOL"
```

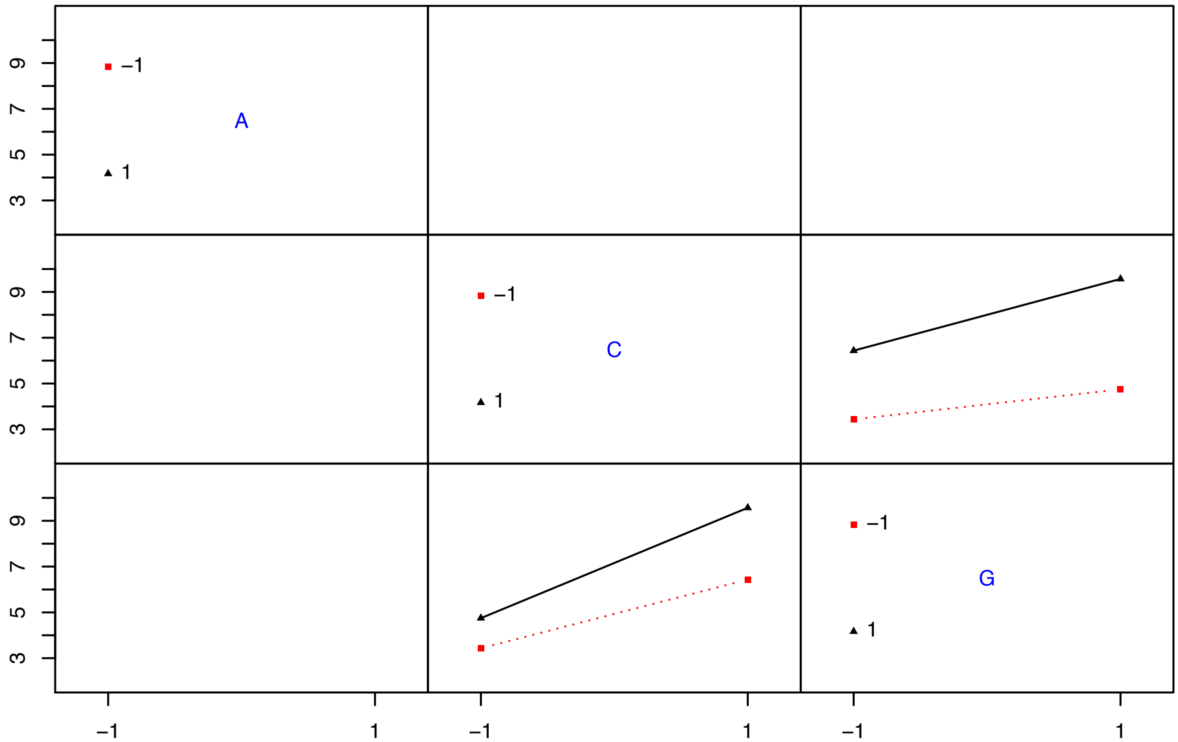
Interaction Plots

```r
## library(phia)
IAPlot(final_model_1,
       main = "Figure 9: Interaction Plot Matrix for Response 1 (Range)")
```




```r
IAPlot(final_model_2,
       main = "Figure 10: Interaction Plot Matrix for Response 2 (Water Consumption)")
```

Figure 10: Interaction Plot Matrix for Response 2 (Water Consumpti



Appendix D

Question 8: Confirmation Experiments

Predicted Values

```
# predicted values of y1
# to optimize y1
y1_optim_pred <- obj_func_1(c(-1,-1,1)) * -1
y1_optim_pred
```

```
## (Intercept)
##   0.3401555
```

```
# to optimize y2
y1_optim_pred <- obj_func_1(c(1,-1,-1)) * -1
y1_optim_pred
```

```
## (Intercept)
## -0.04028982
```

```
# for both
y1_optim_pred <- obj_func_1(c(-1,-1,-1)) * -1
y1_optim_pred
```

```
## (Intercept)
##   0.2260031
```

```
# predicted values of y2
# to optimize y1
y2_optim_pred <- obj_func_2(c(-1,-1,1))
y2_optim_pred
```

```
## (Intercept)
##    4.847293
```

```
# to optimize y2
y2_optim_pred <- obj_func_2(c(-1,-1,1))
y2_optim_pred
```

```
## (Intercept)
##    4.847293
```

```
# for both
y2_optim_pred <- obj_func_2(c(-1,-1,-1))
y2_optim_pred
```

```
## (Intercept)
##    3.531415
```

Design of the Confirmation Experiment

```r
# make dataframe for confirmation experiments
A <- c(-1, 1, -1, -1)
B <- rep(-1, 4)
C <- c(-1, -1, -1, -1)
D <-  rep(-1, 4)
E <-  rep(-1, 4)
F <-  rep(-1, 4)
G <- c(1, -1, -1, -1)
H <-  rep(-1, 4)

conf <- data.frame(A, B, C, D, E, F, G, H)
conf
```

```
##    A  B  C  D  E  F  G  H
## 1 -1 -1 -1 -1 -1 -1  1 -1
## 2  1 -1 -1 -1 -1 -1 -1 -1
## 3 -1 -1 -1 -1 -1 -1 -1 -1
## 4 -1 -1 -1 -1 -1 -1 -1 -1
```

```r
# randomize the rows
set.seed(10000)
rows <- sample(nrow(conf))
conf <- conf[rows,]
conf
```

```
##    A  B  C  D  E  F  G  H
## 2  1 -1 -1 -1 -1 -1 -1 -1
## 3 -1 -1 -1 -1 -1 -1 -1 -1
## 1 -1 -1 -1 -1 -1 -1  1 -1
## 4 -1 -1 -1 -1 -1 -1 -1 -1
```

```r
# New data frame with actual values
conf_actual <- conf
conf_actual$A <- ifelse(conf_actual$A == -1, Alpha[1], Alpha[2])
conf_actual$B <- ifelse(conf_actual$B == -1, Beta[1], Beta[2])
conf_actual$C <- ifelse(conf_actual$C == -1, Aq[1], Aq[2])
conf_actual$D <- ifelse(conf_actual$D == -1, d[1], d[2])
conf_actual$E <- ifelse(conf_actual$E == -1, Mrt[1], Mrt[2])
conf_actual$F <- ifelse(conf_actual$F == -1, Mrf[1], Mrf[2])
conf_actual$G <- ifelse(conf_actual$G == -1, pin[1], pin[2])
conf_actual$H <- ifelse(conf_actual$H == -1, dzul[1], dzul[2])
conf_actual
```

```
##    A B     C   D    E    F G H
## 2 90 0 2e-06 0.1 0.01 0.01 1 5
## 3  0 0 2e-06 0.1 0.01 0.01 1 5
## 1  0 0 2e-06 0.1 0.01 0.01 2 5
## 4  0 0 2e-06 0.1 0.01 0.01 1 5
```

Reading in Results of the Confirmation Experiment

```
# Writing and Results
write.table(conf_actual, "conf.txt", sep="\t", quote=FALSE, dec=".",row.names=FALSE)
conf_result <- read.delim("conf_result.txt", header = TRUE, sep = ",")
conf_result
```

```
##   alpha beta    Aq   d   mt   mf pin dzul consumption      range
## 1    90    0 2e-06 0.1 0.01 0.01   1    5    3.203725 0.01879764
## 2     0    0 2e-06 0.1 0.01 0.01   1    5    3.291689 0.26943327
## 3     0    0 2e-06 0.1 0.01 0.01   2    5    4.796737 0.36649784
## 4     0    0 2e-06 0.1 0.01 0.01   1    5    3.192858 0.26154320
```

Calculating Root Mean Squared Error of Prediction

```
range_obs <- conf_result$range
consumption_obs <- conf_result$consumption

range_pred <- c(obj_func_1(c(1,-1,-1))*-1, obj_func_1(c(-1,-1,-1))*-1,
                obj_func_1(c(-1,-1,1))*-1, obj_func_1(c(-1,-1,-1))*-1)
consumption_pred <- c(obj_func_2(c(1,-1,-1)), obj_func_2(c(-1,-1,-1)),
                      obj_func_2(c(-1,-1,1)), obj_func_2(c(-1,-1,-1)))

#rmse of range
sqrt(1/4 * sum((range_pred - range_obs)^2))
```

```
## [1] 0.04282091
```

```
#rmse of consumption
sqrt(1/4 * sum((consumption_pred - consumption_obs)^2))
```

```
## [1] 0.218731
```