Kristen Fukunaga, Karina Santoso, Steven Turnbull STATS 101B
8 June 2021 Professor Vazquez Alcocer

Optimal Garden Sprinkler Characteristics

Introduction

The efficiency and quality of a garden sprinkler are measured using two quantities: its consumption of water and its spray range. In order to do its job most effectively with minimal resources expended, we desire a garden sprinkler to have as low of a water consumption as possible and as high of a spray range as possible. There are eight factors that can be modified during the process of building a sprinkler: vertical nozzle angle (α), tangential nozzle angle (β), nozzle profile (Aq), diameter of sprinkler head (d), static friction moment (Mt), dynamic friction moment (Mf), entrance pressure (pin), and diameter flow line (dzul).

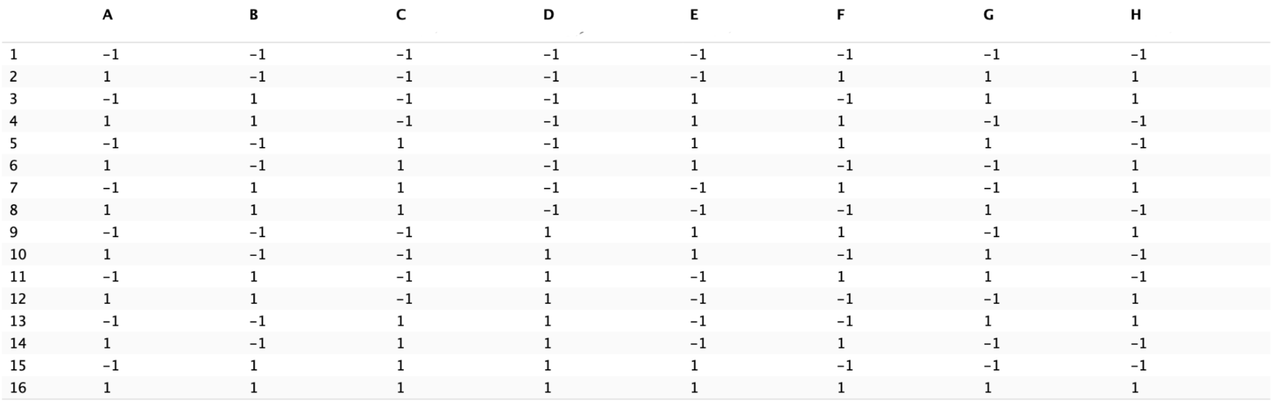
The problem we aim to solve in this project is to discover which of these eight factors have an effect on the consumption and range of a garden sprinkler and find the optimal settings of these factors to maximize the spray range and minimize the water consumption. We are provided with the lower and upper limits of these eight factors and will conduct experiments in order to accomplish the aforementioned goals. However, due to budgetary constraints, we are allotted a maximum of 20 test runs, all of which are conducted on the same day and by a single operator.

Methodology- Experimental Design

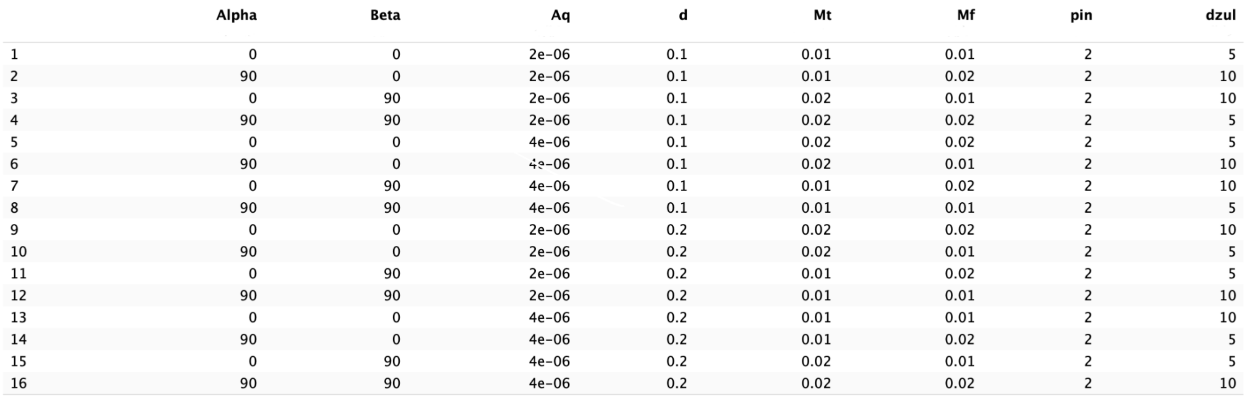
1. Propose a cost-efficient experimental design. Motivate your decision in statistical and practical terms.

In order to achieve this, the cost-efficient experimental design used was a 2^{8-4} fractional factorial design. Since we have eight factors to investigate and at most 20 runs in which to do that, the largest power of 2 below 20 is 16, so we include 16 runs in our fractional factorial design. We take the lower and upper limits of each factor as our two levels for the factor in order to do this.

For our 2^{8-4} fractional factorial design, which is a 1/16 fraction of 8 factors in 16 runs, our chosen design generators are: E = BCD, F = ACD, G = ABC, and H = ABD (See A.4 for full alias structure). This produces the design shown below in Figure 1, where the 8 factors A through H are the 8 factors listed above in order.



	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



	Alpha	Beta	Aq	d	Mt	Mf	pin	dzul
1	0	0	2e-06	0.1	0.01	0.01	2	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	2	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	2	10
7	0	90	4e-06	0.1	0.01	0.02	2	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	2	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	2	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	2	5
15	0	90	4e-06	0.2	0.02	0.01	2	5
16	90	90	4e-06	0.2	0.02	0.02	2	10

Figure 1: The Experimental Design Chosen- a 2^{8-4} Fractional Factorial Coded Design (top) Factor Settings (bottom)

The motivation for choosing this design is that fractional factorial designs allow us to study many factors using only a small number of test combinations when we cannot run a full 2^8factorial design. This design is a Resolution IV design, meaning that no main effect is aliased with any other main effect or any two-factor interaction, and two-factor interactions are aliased with each other. So, we will be able to estimate all the main effects of the 8 factors and gain some information on the interactions between main effects, which will help us determine which of the 8 have a significant effect on either the water consumption or spray range of a garden sprinkler.

We reasonably assume that high order interactions with more than two factors are negligible in order to obtain more information on our main and lower order interaction effects. We follow the sparsity of effects principle, which states that when there are several variables, the processes are likely to be driven primarily by only a few main and low order interaction effects. Additionally, we chose this design since it only utilizes 16 of our 20 total allotted experimental runs, so we will have 4 runs remaining for confirmation experiments after our results and recommended settings for the factors are discovered. Since we can learn about main effects and lower order interactions with enough runs left over for confirmation experiments, this design is both practically efficient and statistically effective. The benefits of this model are discussed more extensively in the next section.

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?

Below are correlation plots for our design, where we test 8 factors in 16 runs using a fractional factorial design. We choose to focus only on 1 and 2 factor interactions and assume higher order interactions have small effects. First, we discuss the performance of our model if studying main effects only. In this case, we see that our design has high performance. No main effects are aliased with each other, and we have maximal D-efficiency being equal to 1. This means that with a main effects design, we would be able to estimate regression coefficients with high precision.

However, since we are unfamiliar with the equipment, we cannot rule out the possibility of significant two-factor interactions. Thus, while our full main effects *and* two-factor interactions model has a low D-efficiency of 0.297, we *still* choose to include two-factor interactions in our model. We first and foremost need to be able to identify relevant factors and cannot dismiss two-factor interactions. Again, in this model no main effects are aliased with other main effects and no main effects are aliased with two-factor interaction effects. However, we see that 2-factor interactions are aliased with each other. Thus, our design cannot estimate all two factor interaction effects. We note, however, that it is impossible to create a design that estimates all main effects *and* two-factor interactions with a run size smaller than 20 (we have less than 20 runs for our experiment since we need to run confirmation tests). This is because we would need to estimate $1 + 8 + 7 \cdot 8/2 = 37$coefficients and we always need to have at least as many runs as coefficients we are trying to estimate. Thus, in order to estimate all main effects, two-factor interactions, and the intercept, we would require at least 37 runs, which is out of our budget. Therefore, given our overall goals of identifying important effects, our run size constraint, and the fact that no main effects are aliased with other (lower order) effects, we are content with the performance of our model.

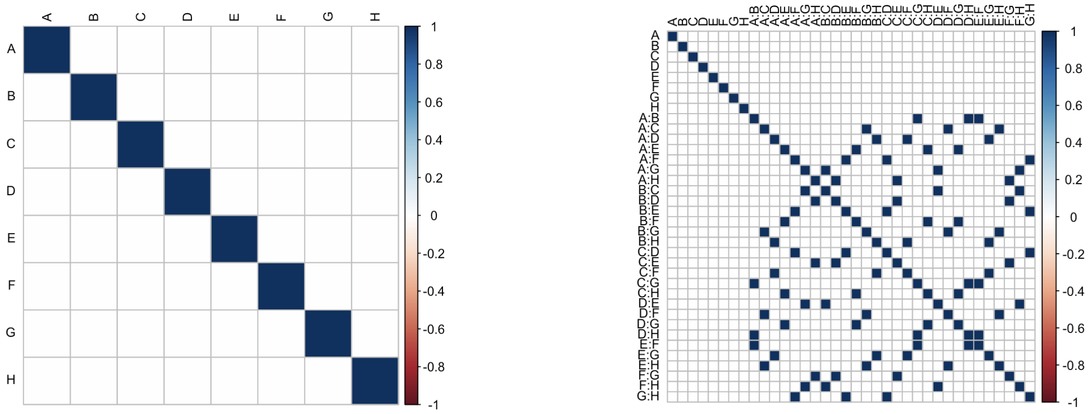
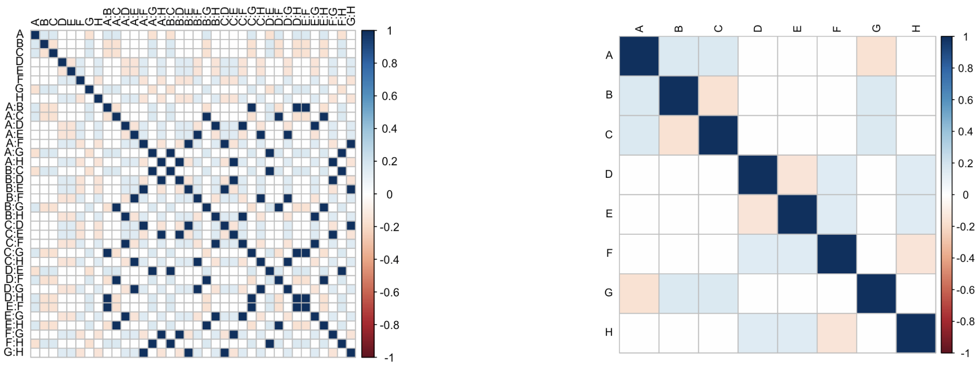


Figure 2: Correlation Plots Showing Aliasing Between Main Effects and 2-Factor Interactions for our Design

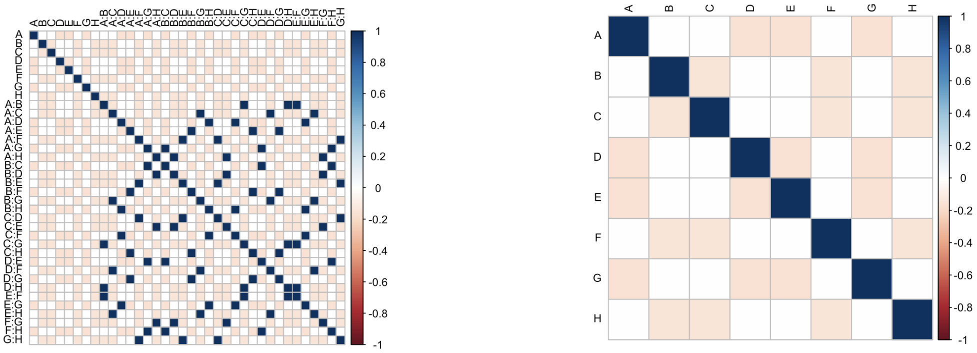
We make a final note that if discussions with the production engineers give us reason to believe that two-factor interactions are negligible, we would recommend the main effects only model to maximize precision. Otherwise, we encourage the 8 factors in 16 runs fractional factorial design, as it has relatively high performance and allows us to better understand main effects as well as low-order interactions.

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?

Since it is possible that the production engineers may encounter failed tests in the experiment, we now consider the performance of our design when two randomly chosen test combinations are removed. We first used R's built in random number generator to randomly select 2 observations to remove. We repeated this process 5 times and studied the performances of the resulting design each time. We will include some of the correlation plots for the resulting designs for illustrative purposes below:



a. Correlation Plots With Runs 2 and 10 Removed



b. Correlation Plots With Runs 10 and 16 Removed

Figure 3: Correlation Plots of Main Effects and 2-factor Interactions (left)
Correlation Plots of Main Effects Only (right)

In the selected correlation plots above, we see that the removal of 2 observations results in partial aliasing between main effects, between main effects and 2-factor interactions, and between 2-factor interactions. However, we do not see *full* aliasing between any two main effects and we do not see *full* aliasing between a main effect and a two-factor interaction. We cannot include all plots here, but it turns out that when we view the correlation plots of all 120 possible designs when two observations are removed (16 choose 2 runs to be removed = 120 possibilities), this is still the case. That is, there is no possible choice of two runs such that the removal of these two runs results in a design with full aliasing between two main effects or full aliasing between a main effect and a two-factor interaction term. This shows that if there are accidental failed tests, we will still be able to understand the most important main effects relatively well, as our design will avoid full aliasing between main effects and other (lower order) effects.

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?

Below are the correlation plots for the production engineers' design. We see that much like our design, there are no main effects aliased with other main effects and there are no main effects aliased with 2-factor interactions. However, we see that the alias structure is much more complex as there is more partial aliasing between 2-factor interactions in this model than in our model. Moreover, we also see that we would not really be able to improve the precision of our estimates with the production engineers' model, as it has a (slightly) lower D-efficiency score (0.249 compared to our 0.297). Because we are mainly focused on studying two levels per machine setting, we encourage the client to use our model as it requires one less run, it has a simpler alias structure making our design easier to interpret, and the precision of our estimates will be about the same. However, we note that if the client expresses a strong desire to learn about range and consumption for more than two levels per sprinkler setting, the production engineers' model may be of interest.

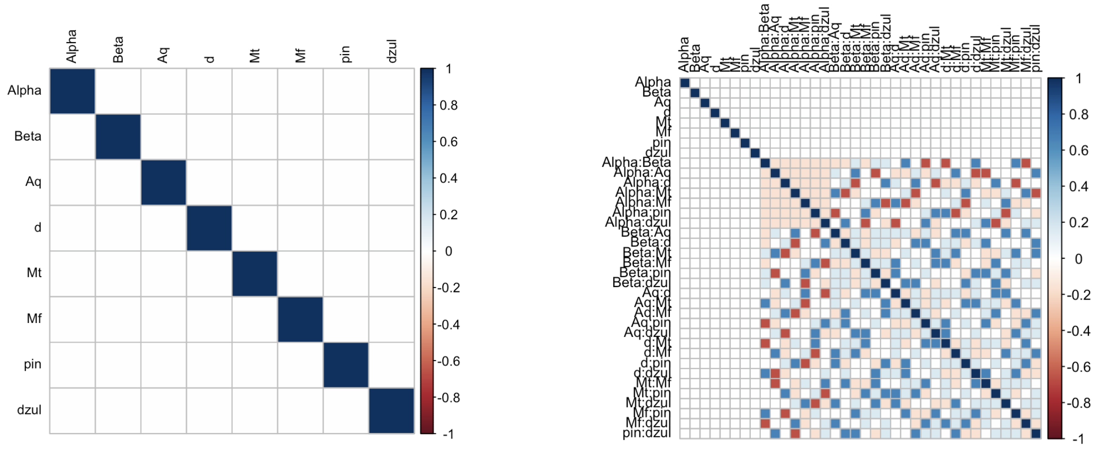


Figure 4: Correlation Plots Showing Aliasing Between Main Effects (left) and Between Main Effects/2-Factor Interactions (Right) for the Production Engineers' Model

Methodology- Data Analysis

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

Using the discussed 2^{8-4} fractional factorial model, the process engineers prototyped 16 different sprinklers and recorded the data (see appendix A.1 for the results of the experimental trials). We note that these runs were performed in randomized order, and the results are arranged and inputted into our code in a convenient viewing format).

The goal is to find the optimal levels of the factors of interest. Here the typical linear regression model for 2-factorial experiments will be assumed: $Y = X\beta + \epsilon$. Here the two models below are specified:

$$Water\ Consumption\ (consumption) = X\beta_1 + \epsilon$$
$$Range\ of\ sprinkler\ (range) = X\beta_2 + \epsilon$$

X is the matrix of the factors and their interactions (with an extra column of ones to represent the intercept), consumption and range are our response variables of interest, β_1, β_2 are the unknown regression coefficients, and ϵ is the noise. For each factor in *X*, the higher levels are coded as 1 and the lower limits are coded as -1.

Using R's built in linear model software, we fit preliminary full models containing all main effects and interactions and use half-normal plots to determine which of these factors have a significant effect on the response variables. For this analysis the significance level of$\alpha = 0.05$will be used. To succinctly display the information obtained from the linear models, half normal plots will be used to find potentially significant contributors to both variables of interest.

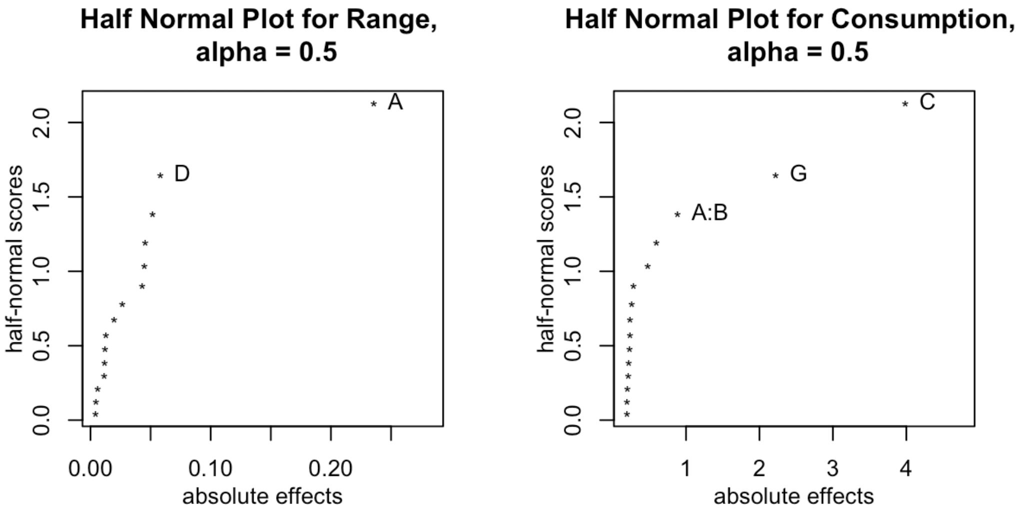


Figure 5: Daniel Plots for the Specified Models Above (D = d, A = alpha, B = beta, C = Aq, D = d, G = pin)

The half normal plots in Figure 5 give an indication of how to reliably reduce the models to continue analysing the data. The full water consumption model can be reduced to:

$$Water\ Consumption\ = \beta_0\ +\ \beta_{Aq}(Aq)\ +\ \beta_{pin}(pin)\ +\ \beta_{\alpha:\beta}(\alpha{:}\ \beta)\ +\ \epsilon$$

Here the full model has been reduced to only terms that the half normal plots suggest have a significant effect on water consumption of the sprinkler: nozzle profile (Aq), entrance pressure (pin) and the interaction between the vertical nozzle angle and Tangential nozzle angle (α: β). We note that since α: β is aliased with Mt:Mf, Aq:pin, and d:dzul, it may be the case that one of these interactions or a combination of these interactions are influential factors. That is, we cannot precisely conclude which of these two factor interactions have a significant effect on water consumption due to our alias structure. However, we can make the effect of heredity assumption. Since Aq and pin are determined to be significant, it is reasonable to assume that the interaction between pin and dzul has the largest effect among the four two-factor interactions listed above. Under this assumption, we now have:

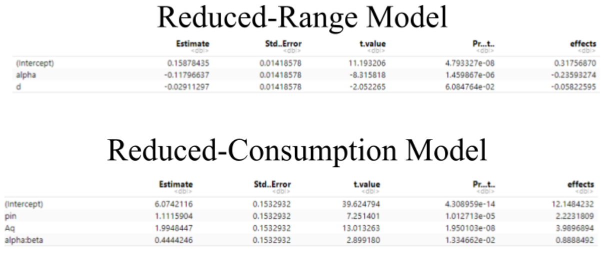
$$Water\ Consumption\ = \beta_0\ +\ \beta_{Aq}(Aq)\ +\ \beta_{pin}(pin)\ +\ \beta_{Aq:pin}(Aq{:}\ pin)\ +\ \epsilon$$

Similar to the way the water consumption model was reduced, the cumbersome full range model can be reduced:

$$range\ =\ \beta_0\ +\ \beta_\alpha(\alpha)\ +\ \beta_d(d)\ +\ \epsilon$$

In this case, there are only two significant terms: vertical nozzle angle (α) and the diameter of the sprinkler head (d).

The re-fitted models provide reliable t-statistics, standard errors and p-values further revealing the significance of the factors discussed above due to their strong absolute effects. From Figure 6, the significant factors of interest from the half normal plots are all found to be significant at α = 0.05 besides the diameter of the sprinkler head, but we will leave it in the model in case it provides some extra information later on in the analysis. The linear models specified by these regression outputs are:



Reduced-Range Model

	Estimate	Std..Error	t.value	Pr...t..	effects
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	0.15878435	0.01418578	11.193206	4.793327e-08	0.31756870
alpha	-0.11796637	0.01418578	-8.315818	1.459867e-06	-0.23593274
d	-0.02911297	0.01418578	-2.052265	6.084764e-02	-0.05822595

Reduced-Consumption Model

	Estimate	Std..Error	t.value	Pr...t..	effects
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	6.0742116	0.1532932	39.624794	4.308959e-14	12.1484232
pin	1.1115904	0.1532932	7.251401	1.012713e-05	2.2231809
Aq	1.9948447	0.1532932	13.013263	1.950103e-08	3.9896894
alpha:beta	0.4444246	0.1532932	2.899180	1.334662e-02	0.8888492

Figure 6: Results of Reduced Regression Models

$$\widehat{range} = 0.15878 - 0.11797(alpha) - 0.02911(d)$$

$$\widehat{Water\ Consumption} = 6.07421 + 1.11159(pin) + 1.99484(Aq) + 0.44442(Aq{:}pin)$$

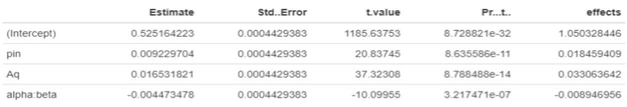
Before continuing the analysis, it is necessary to verify our assumptions made when fitting these linear models:

- **Constant variance condition**: Constant variance can be checked by noticing a fanning effect in the residuals vs. fitted plot; however, to be rigorous, the non-constant variance score test tests the null hypothesis that the variance is constant. The non-constant variance score test for the water consumption model rejects that null hypothesis, providing evidence of non-constant variance within our model. These findings are supported by a slight fanning effect located in the residuals vs. fitted plot. The range model does not appear to have any obvious inconsistencies in the residuals vs. fitted plot and the ncv test accepts the null hypothesis of constant variance. A strictly-increasing variance stabilizing transformation will be applied to the water consumption in an attempt to improve the adequacy of our model.(The results of these tests are located in Figure A.7)
- **Normality condition:**To check this condition we can use the qq plot or the shapiro-wilk normality test. We once again do both. There are large variations from the theoretical distribution line in the tails of the qq plot for the water consumption model. The shapiro-wilk test finds these to be grounds to reject the null hypothesis of normality. We will look for improvements in this normality after applying the transformation. The range model does not appear to have significant variations from the expected theoretical distribution (Backed by Shapiro-wilk test). (The results of these tests are displayed in Figure A.5)
- **Residuals and Run Order Correlation:** There is a slight sinusoidal correlation between run order and residual in the consumption model that is of concern but the range model appears to have no such correlation. We will once again look for improvements after transforming the water consumption model. (The results of these tests are displayed in Figure A.6)

As discussed, a variance-stabilizing transformation should now be applied to the response variable (Water Consumption) of the water consumption model. The chosen transformation is a box-cox transformation with $\lambda = -1.797$(This transformation is strictly increasing to the nature of its derivative). It will soon be shown that we see improvements in the adequacy of our model after applying this transformation. The new fitted model will be of the form (Say Y = Water Consumption):

$$\frac{(Y^{-1.797} - 1)}{-1.797} = \beta_0 + \beta_{Aq}(Aq) + \beta_{pin}(pin) + \beta_{Aq:pin}(Aq{:}pin) + \epsilon$$

The factors have the same meanings as before. Re-Fitting the model gives the resulting output in Figure 7 which implies the fitted equation:



	Estimate	Std..Error	t.value	Pr...t..	effects
(Intercept)	0.525164223	0.0004429383	1185.63753	8.728821e-32	1.050328446
pin	0.009229704	0.0004429383	20.83745	8.635586e-11	0.018459409
Aq	0.016531821	0.0004429383	37.32308	8.788488e-14	0.033063642
alpha:beta	-0.004473478	0.0004429383	-10.09955	3.217471e-07	-0.008946956

Figure 7:Linear Model results for Box Cox Transformed Water Consumption Model

$$\frac{(Y^{-1.797} - 1)}{-1.797} = 0.5252 + 0.0165\,(Aq) + 0.0092(pin) - 0.0044(Aq{:}pin) + \epsilon$$

Checking the adequacy we can see that there is improvement in our tests for each condition. The plots in Figure A.8 and test results in figure A.9 show improvements. Although it appears there is more variation in the qq plot, there are no large differences in the tails which improves the overall adequacy of our model (Supported by the Shapiro-Wilk Test). In terms of constant variance, the residuals appear evenly distributed around 0 and there appears to be little fanning effect (This is supported by the non constant variance test as well). The apparent correlation between run order and residuals also appears to have been corrected by our transformation. We will proceed with this model.

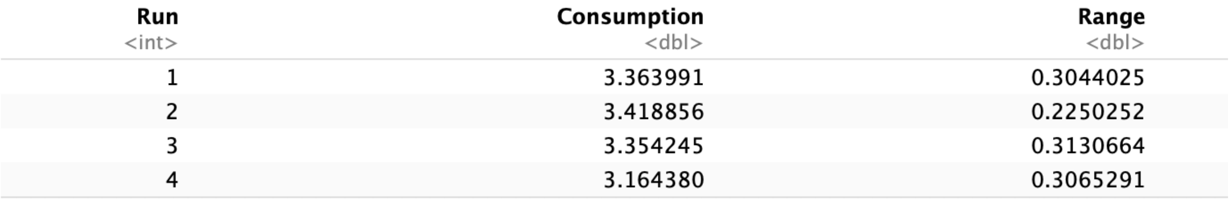
6. What are the most influential factors?

In summary, we find that the most influential factors on consumption rate are Aq, pin, the interaction between Aq and pin. The most influential factors on range are alpha and d. The estimated effects for each of these factors are listed in figure A.10 in the appendix (these are the estimated effects when we move from the low level to the high level).

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

Now that the reduced models are adequate, the next step would be to determine the optimal settings for water consumption and spray range. To determine the optimal levels, we calculated the estimated range and consumption for each combination of significant factors using the reduced and transformed models for range and consumption discussed above. The results are shown in Figures A.2 and A.3 in the appendix. Comparing the different choices for settings, we see that setting alpha, beta, d, Aq, and pin at the lowest levels result in lowest expected consumption and highest expected range. Thus, we recommend setting alpha to 0, beta to 0, d to 0.2, Aq to 0.000002, and pin to 1. The clients can choose anything for the remaining settings or we can just let the settings of the other (not significant) variables be at the low levels. The expected consumption rate for this combination of settings is 3.19 l/min and the expected range is 0.305 meters.

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



Run <int>	Consumption <dbl>	Range <dbl>
1	3.363991	0.3044025
2	3.418856	0.2250252
3	3.354245	0.3130664
4	3.164380	0.3065291

Figure 8: Confirmation Experiment Results With All Factors at Low Level

Using the settings recommended above, we conduct 4 runs of confirmation experiments to validate our findings. We set all factors to their lower limits and got the results above in Figure 8. The mean consumption rate of the four runs shown above is 3.325 l/min and the mean range is 0.287 meters. Since our model predicted a consumption rate of 3.19 l/min and a spray range of 0.305 meters (see figures A.2 and A.3 in the appendix) and both of these values are fairly close to the estimates we obtained with our model, we conclude that our predictions and model seem to be fairly accurate.

Moreover, comparing these values of consumption and range (observed when settings were set to our recommended levels) to our initial 16 experimental runs shows us that our recommended settings tend to result in a lower consumption and higher range. Only 4 of the original runs had a higher range than our average, and none of them had a lower consumption (Figure A.1 in the appendix). The settings of the factors that we keep constant between the 4 runs seem to be extremely effective and optimal in order to keep water consumption levels low and spray range levels high.

Conclusions

From the results of our study, we conclude that the optimal settings to construct a garden sprinkler in order to maximize its spray range and to minimize its water consumption are:

- A vertical nozzle angle (α) of 0.0°- lower limit of the factor
- A tangential nozzle angle (β) of 0.0°- lower limit of the factor
- A nozzle profile (Aq) of 2e-06 m2- lower limit of the factor
- A diameter of sprinkler head (d) of 0.1 m- lower limit of the factor
- Any static friction moment (Mt)
- Any dynamic friction moment (Mf)
- An entrance pressure (pin) of 1.0 Bar- lower limit of the factor
- Any diameter flow line (dzul)

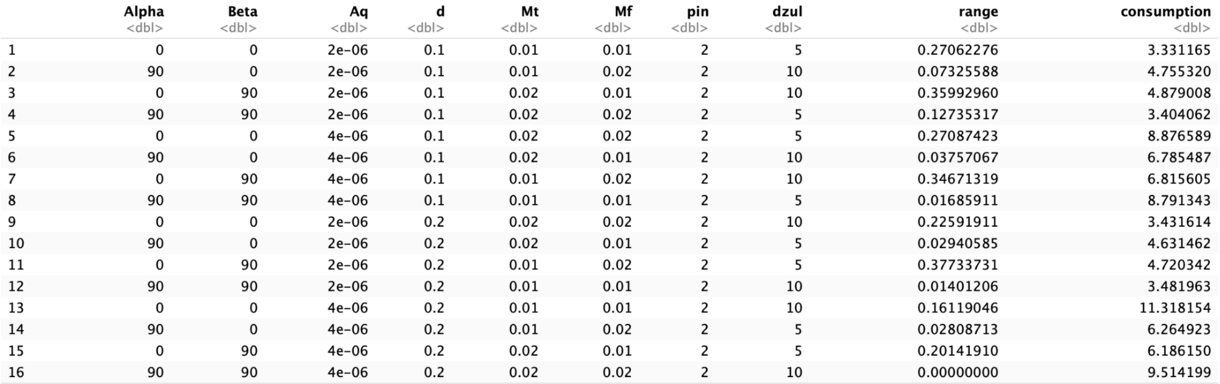
We recommend these settings for Aq and pin to minimize the water consumption, and these settings for d and α to maximize the water range. For the three factors that do not seem to significantly affect the water consumption or spray range, the cheapest or most convenient setting should be chosen.

The successful outcomes of our confirmation experiments demonstrates that the design chosen and corresponding models worked well for our statistical analysis. The chosen 2^{8-4} fractional factorial design was effective in allowing us to estimate the main effects of the 8 factors under investigation as well as some of the two-factor interaction effects relatively well (see correlation plots).

Additionally, the models used to analyze the main effects of the factors are valid ones as proven by the residual analyses that show that the assumptions of constant and normal variance necessary for the model to hold are met, for the final transformed model. The transformation used for the model is an appropriate one as seen by the comparisons of the residual plots before and after the transformation. The assumptions are met much more adequately after the transformation.

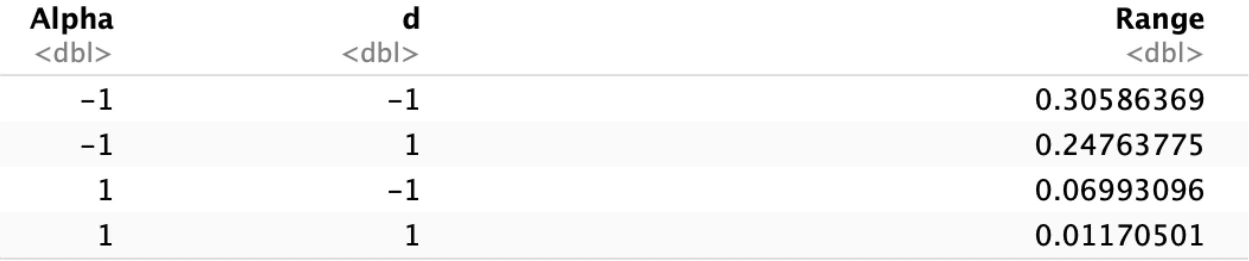
For future experimentation, testing out a middle level of each factor right in between the lower and upper bounds may prove beneficial as the relationship between the response variables and the different factors may not necessarily be linear and a setting for the factor somewhere in the middle may actually be the most optimal option for our desired outcomes. Additionally, running a 2^{8-3} fractional factorial design with 32 runs or a fractional design with an even larger number of runs if budget and resources allow could be beneficial in order to get more accurate estimates of the main and two-factor interaction effects.

Appendix



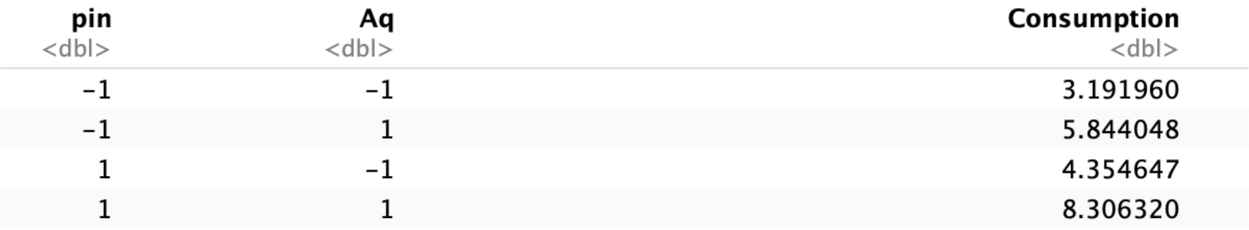
	Alpha	Beta	Aq	d	Mt	Mf	pin	dzul	range	consumption
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0	0	2e−06	0.1	0.01	0.01	2	5	0.27062276	3.331165
2	90	0	2e−06	0.1	0.01	0.02	2	10	0.07325588	4.755320
3	0	90	2e−06	0.1	0.02	0.01	2	10	0.35992960	4.879008
4	90	90	2e−06	0.1	0.02	0.02	2	5	0.12735317	3.404062
5	0	0	4e−06	0.1	0.02	0.02	2	5	0.27087423	8.876589
6	90	0	4e−06	0.1	0.02	0.01	2	10	0.03757067	6.785487
7	0	90	4e−06	0.1	0.01	0.02	2	10	0.34671319	6.815605
8	90	90	4e−06	0.1	0.01	0.01	2	5	0.01685911	8.791343
9	0	0	2e−06	0.2	0.02	0.02	2	10	0.22591911	3.431614
10	90	0	2e−06	0.2	0.02	0.01	2	5	0.02940585	4.631462
11	0	90	2e−06	0.2	0.01	0.02	2	5	0.37733731	4.720342
12	90	90	2e−06	0.2	0.01	0.01	2	10	0.01401206	3.481963
13	0	0	4e−06	0.2	0.01	0.01	2	10	0.16119046	11.318154
14	90	0	4e−06	0.2	0.01	0.02	2	5	0.02808713	6.264923
15	0	90	4e−06	0.2	0.02	0.01	2	5	0.20141910	6.186150
16	90	90	4e−06	0.2	0.02	0.02	2	10	0.00000000	9.514199

Figure A.1: The Experimental Results for Range and Consumption



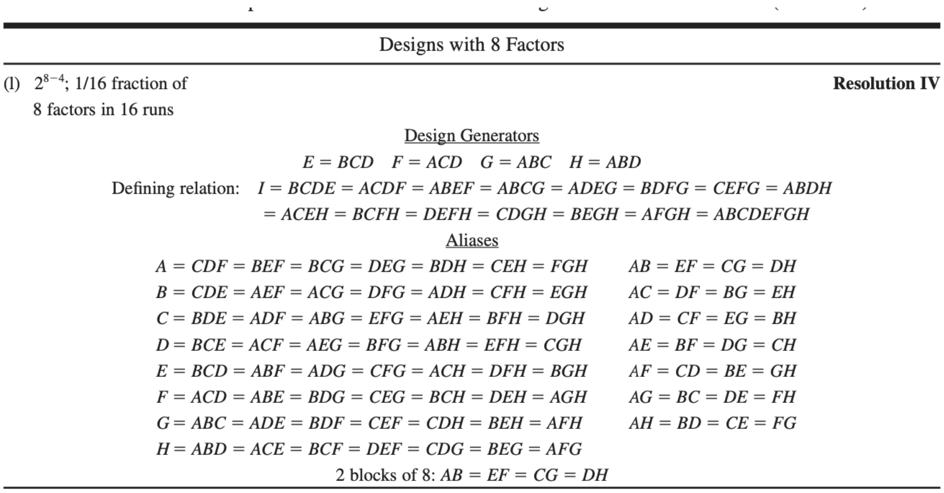
Alpha	d	Range
<dbl>	<dbl>	<dbl>
−1	−1	0.30586369
−1	1	0.24763775
1	−1	0.06993096
1	1	0.01170501

Figure A.2: Estimated Range for each Factor Level Combination (using our final model)



pin	Aq	Consumption
<dbl>	<dbl>	<dbl>
−1	−1	3.191960
−1	1	5.844048
1	−1	4.354647
1	1	8.306320

Figure A.3: Estimated Consumption for Each Factor Level Combination (using our final model)



Designs with 8 Factors

(l) 2^{8-4}; 1/16 fraction of 8 factors in 16 runs **Resolution IV**

Design Generators

$E = BCD \quad F = ACD \quad G = ABC \quad H = ABD$

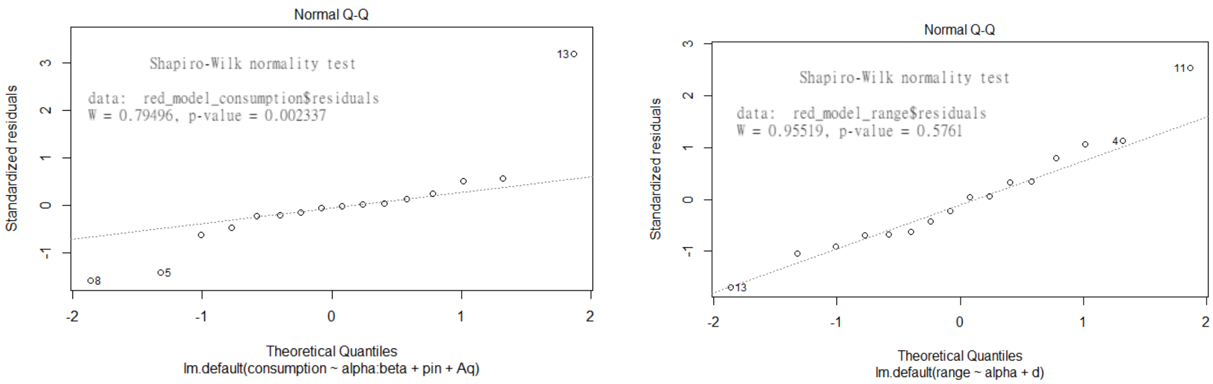
Defining relation: $I = BCDE = ACDF = ABEF = ABCG = ADEG = BDFG = CEFG = ABDH$
$= ACEH = BCFH = DEFH = CDGH = BEGH = AFGH = ABCDEFGH$

Aliases

$A = CDF = BEF = BCG = DEG = BDH = CEH = FGH$ $AB = EF = CG = DH$
$B = CDE = AEF = ACG = DFG = ADH = CFH = EGH$ $AC = DF = BG = EH$
$C = BDE = ADF = ABG = EFG = AEH = BFH = DGH$ $AD = CF = EG = BH$
$D = BCE = ACF = AEG = BFG = ABH = EFH = CGH$ $AE = BF = DG = CH$
$E = BCD = ABF = ADG = CFG = ACH = DFH = BGH$ $AF = CD = BE = GH$
$F = ACD = ABE = BDG = CEG = BCH = DEH = AGH$ $AG = BC = DE = FH$
$G = ABC = ADE = BDF = CEF = CDH = BEH = AFH$ $AH = BD = CE = FG$
$H = ABD = ACE = BCF = DEF = CDG = BEG = AFG$

2 blocks of 8: $AB = EF = CG = DH$

Figure A.4: Full Alias Structure for our Design (Montgomery, pg. 709)



```
Shapiro-Wilk normality test

data:  red_model_consumption$residuals
W = 0.79496, p-value = 0.002337
```

```
Shapiro-Wilk normality test

data:  red_model_range$residuals
W = 0.95519, p-value = 0.5761
```

Figure A.5: Reduced Model QQ-Plots. Left:Consumption Model(Note large variations is both tails, significant Shapiro-Wilk normality test stat), Right: Range Model

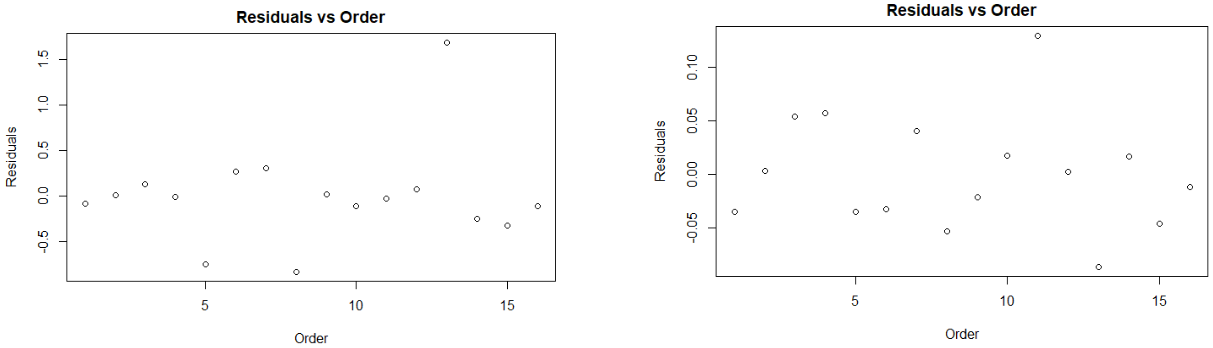
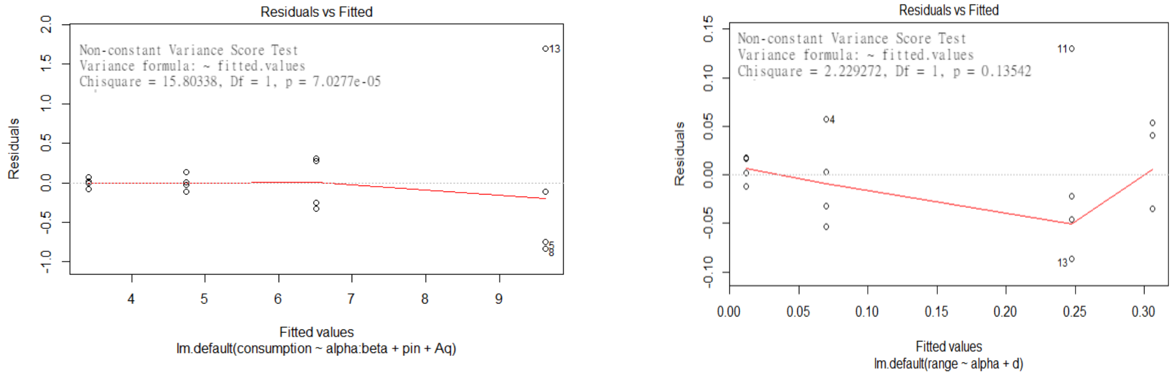


Figure A.6: Reduced Model Residuals vs Order Plot. Left:Consumption Model (Note sinusoidal pattern in data), Right: Range Model



```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 15.80338, Df = 1, p = 7.0277e-05
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 2.229272, Df = 1, p = 0.13542
```

Figure A.7: Constant Variance Tests w/ residuals vs fitted and ncvTest. Left: Consumption Model (Note fanning pattern in data), Right: Range Model

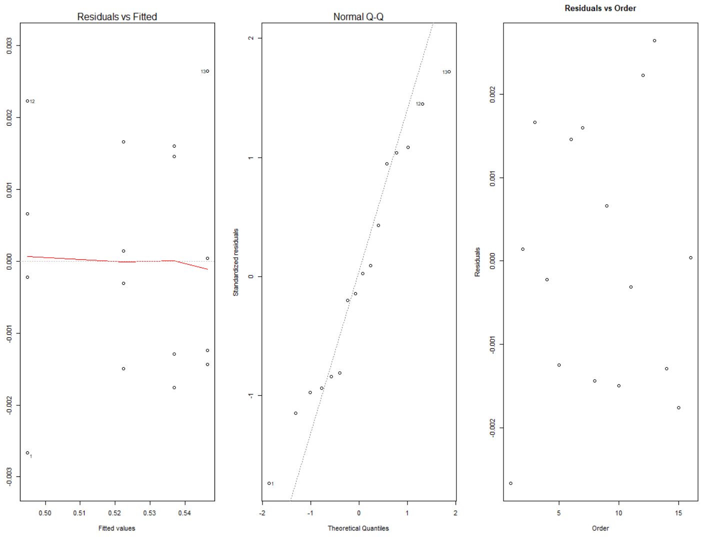
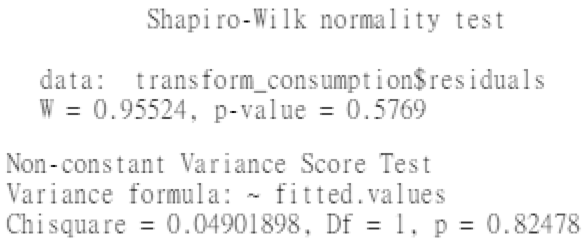


Figure A.8: Plots to check adequacy of transformed water consumption model



```
Shapiro-Wilk normality test

data:  transform_consumption$residuals
W = 0.95524, p-value = 0.5769

Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.04901898, Df = 1, p = 0.82478
```

Figure A.9: Tests to check adequacy of transformed water consumption model

Estimated Effects				
Alpha	d	pin	Aq	pin:Aq
-0.235	-0.058	1.017	1.032	0.991

Figure A.10: Estimated Effects of Significant Factors

Statement of Contribution

Steven contributed the data analysis code for question 5,6,7 (code was verified by Kristen and Karina) and wrote the corresponding sections of the report. Kristen wrote questions 2, 3, and 4 and wrote the code for these sections. Karina wrote the introduction, conclusion, and questions 1 and 8. We all worked together to clean up the code, discuss ideas, and organize the final report.

References

Montgomery, Douglas C. Design and Analysis of Experiments. Eighth Edition, John Wiley & Sons, Inc., 2013.

Code

~~~
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(FrF2)
library(corrplot)
library(AlgDesign)
library(car)
library(MASS)
```

```{r General Information about our Design}
###here, A = alpha, B = beta, C = Aq, D = d, E = Mt, F = Mf, G = pin, and H = dzul

my.design.by.hand <- FrF2(nruns = 16, nfactors = 8, factor.names = c("A", "B", "C", "D", "E",
"F", "G","H"),generators = c("BCD", "ACD", "ABC","ABD"), randomize = F)
cat("Generators of the design \n")
generators(my.design.by.hand)
cat("Alias structure \n")
design.info(my.design.by.hand)$aliased


des1 = data.frame(apply(my.design.by.hand,2,as.numeric))
des1[des1[,1]==-1,1] <- 0
des1[des1[,1]==1,1] <- 90

des1[des1[,2]==-1,2] <- 0
des1[des1[,2]==1,2] <- 90

des1[des1[,3]==-1,3] <- 2e-06
des1[des1[,3]==1,3] <- 4e-06

des1[des1[,4]==-1,4] <- 0.10
des1[des1[,4]==1,4] <- 0.20

des1[des1[,5]==-1,5] <- 0.010
des1[des1[,5]==1,5] <- 0.020

des1[des1[,6]==-1,6] <- 0.010
des1[des1[,6]==1,6] <- 0.020

des1[des1[,7]==-1,7] <- 1
des1[des1[,7]==1,7] <- 2

des1[des1[,8]==-1,8] <- 5
des1[des1[,8]==1,8] <- 10
~~~

~~~
colnames(des1) = c("Alpha", "Beta", "Aq", "d", "Mt", "Mf", "pin", "dzul")
rownames(des1) = 1:16
print(des1)

##coded design
my.design.by.hand
```

```{r Intro / Question 1 + 2 Our Coded Design and D-efficiency Score}
D.one <- desnum(my.design.by.hand) # our coded design.
print(D.one)

des2 = as.matrix(D.one)
des3 = matrix(c(rep(1,16), des2[1:16,]),nrow=16)

(det(t(des3)%*%des3)/(16^16))^(1/16)#D-efficiency for our model
(det(t(des3)%*%des3)/(16^9))^(1/9) #D-efficiency for main effects model (not the one we are using)
```
```{r Question 2 Aliasing in Our Model}
X.one <- model.matrix(~(A + B + C + D + E + F + G + H)^2-1, data.frame(D.one))

# Create color map on pairwise correlations.
contrast.vectors.correlations.one <- cor(X.one)
corrplot(contrast.vectors.correlations.one, type = "full", addgrid.col = "gray",
        tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.8)

# create color map only for main effects
X.one.main <- model.matrix(~(A + B + C + D + E + F + G + H)-1, data.frame(D.one))
contrast.vectors.correlations.one <- cor(X.one.main)
corrplot(contrast.vectors.correlations.one, type = "full", addgrid.col = "gray",
        tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.8)
```

```{r Question 3}
#seed was set so this could be reproduced with same randomly chosen rows and cols to remove
set.seed(1234)

##First we consider randomly selecting combinations of 2 runs to remove and study the aliases

#randomly choosing run pairs
random_selections = matrix(rep(0,20),nrow=2)
for (i in 1:10){
  n = sample(1:16,2)
  random_selections[,i] = n
~~~

~~~
}
colnames(random_selections) = 1:10
rownames(random_selections) = (c("first obs","second obs"))
random_selections #randomly chosen rows and cols to remove

#looking at the correlation plots for each of these experiments where the runs are removed
#this looks at the first five resulting experiments' alias structures
for (i in 1:5){
  X.one <- model.matrix(~(A + B + C + D + E + F + G + H)^2-1,
data.frame(D.one[-random_selections[,i],]))

    # Create color map on pairwise correlations.
    contrast.vectors.correlations.one <- cor(X.one)
    corrplot(contrast.vectors.correlations.one, type = "full", addgrid.col = "gray",
             tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.8)

    X.one.main <- model.matrix(~(A + B + C + D + E + F + G + H)-1, data.frame(X.one))
    contrast.vectors.correlations.one <- cor(X.one.main)
    corrplot(contrast.vectors.correlations.one, type = "full", addgrid.col = "gray",
             tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.8)
}

###this code will find corr plots for all possibilities
#I do not recommend running this code! Will take a long time

#a = combn(1:16,2) #matrix containing all 16C2 combinations
#for (i in 1:120){
#  ind = a[,i]
#  X.one <- model.matrix(~(A + B + C + D + E + F + G + H)^2-1, data.frame(D.one[-ind,]))

    # Create color map on pairwise correlations.
#    contrast.vectors.correlations.one <- cor(X.one)
#    corrplot(contrast.vectors.correlations.one, type = "full", addgrid.col = "gray",
#             tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.8)

#    X.one.main <- model.matrix(~(A + B + C + D + E + F + G + H)-1, data.frame(X.one))
#    contrast.vectors.correlations.one <- cor(X.one.main)
#    corrplot(contrast.vectors.correlations.one, type = "full", addgrid.col = "gray",
#             tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.8)
#}

```

```{r Question 4}

#inputting
~~~

```
A = c(0,45,0,0,90,0,90,0,0,90,90,90,0,90,45,45,90)
B = c(90,45,90,0,0,45,90,0,0,0,90,90,90,0,0,90,45)
C =
c(0.000002,0.000003,0.000004,0.000004,0.000002,0.000002,0.000002,0.000004,0.000002,0.00
0004,0.000004,0.000002,0.000003,0.000003,0.000002,0.000004,0.000004)
D = c(0.2,0.15,0.15,0.1,0.15,0.1,0.1,0.2,0.2,0.1,0.1,0.2,0.1,0.2,0.1,0.2,0.2)
E = c(0.02,0.015,0.01,0.02,0.02,0.02,0.015,0.015,0.01,0.01,0.02,0.01,0.01,0.02,0.01,0.02,0.01)
F1 = c(0.015,0.015,0.01,0.02,0.02,0.01,0.02,0.01,0.02,0.015,0.01,0.01,0.02,0.01,0.01,0.02,0.02)
G = c(1,1.5,2,1.5,1,2,2,1,2,2,1,1.5,1,2,1,2,1)
H = c(5,7.5,5,5,10,10,5,10,7.5,10,7.5,10,10,5,5,10,5)

des.alt = data.frame("Alpha"=A,"Beta"=B,"Aq"=C,"d"=D,"Mt"=E,"Mf"=F1,"pin"=G,"dzul"=H)
print(des.alt)


######## recoding the design
des.alt.coded = des.alt
des.alt.coded[des.alt.coded[,1]==0,1] <- -1
des.alt.coded[des.alt.coded[,1]==45,1] <- 0
des.alt.coded[des.alt.coded[,1]==90,1] <- 1

des.alt.coded[des.alt.coded[,2]==0,2] <- -1
des.alt.coded[des.alt.coded[,2]==45,2] <- 0
des.alt.coded[des.alt.coded[,2]==90,2] <- 1

des.alt.coded[des.alt.coded[,3]==2e-06,3] <- -1
des.alt.coded[des.alt.coded[,3]==3e-06,3] <- 0
des.alt.coded[des.alt.coded[,3]==4e-06,3] <- 1

des.alt.coded[des.alt.coded[,4]==0.10,4] <- -1
des.alt.coded[des.alt.coded[,4]==0.15,4] <- 0
des.alt.coded[des.alt.coded[,4]==0.20,4] <- 1

des.alt.coded[des.alt.coded[,5]==0.010,5] <- -1
des.alt.coded[des.alt.coded[,5]==0.015,5] <- 0
des.alt.coded[des.alt.coded[,5]==0.020,5] <- 1

des.alt.coded[des.alt.coded[,6]==0.010,6] <- -1
des.alt.coded[des.alt.coded[,6]==0.015,6] <- 0
des.alt.coded[des.alt.coded[,6]==0.020,6] <- 1

des.alt.coded[des.alt.coded[,7]==1,7] <- -1
des.alt.coded[des.alt.coded[,7]==1.5,7] <- 0
des.alt.coded[des.alt.coded[,7]==2.0,7] <- 1

des.alt.coded[des.alt.coded[,8]==5,8] <- -1
```

~~~
des.alt.coded[des.alt.coded[,8]==7.5,8] <- 0
des.alt.coded[des.alt.coded[,8]==10,8] <- 1

des.alt.coded

#looking at alias structures
contrast.vectors.correlations.opt <- cor(des.alt.coded)
corrplot(contrast.vectors.correlations.opt, type = "full", addgrid.col = "gray",
        tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.8)

# Create color map to see pairwise correlations
X.opt <- model.matrix(~.^2-1, data.frame(des.alt.coded))

contrast.vectors.correlations.opt <- cor(X.opt)
corrplot(contrast.vectors.correlations.opt, type = "full", addgrid.col = "gray",
        tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.8)

#D-efficiency for production engineers' model
des = as.matrix(des.alt.coded)
des1 = matrix(c(rep(1,16), des[1:16,]),nrow=16)

(det(t(des1)%*%des1)/(17^16))^(1/16)
```

```{r Question 5 -- Inputting data}
#experimental range values
range =
c(0.27062275839259337,0.07325588430992591,0.3599295998745906,0.12735317006412863,0
.27087422748167445,0.037570673875332604,0.34671319303962284,0.016859109137131 73,0.
2259191119591298,0.029405852052446384,0.37733730867527276,0.014012056607738575,0.1
611904564560182,0.02808712965950742,0.20141910408882613,0.0)


#experimental consumption values
cons =
c(3.3311653428157535,4.755319793890016,4.879007616438037,3.4040622672572223,8.87658
9397505471,6.785486701253629,6.815605125654621,8.791343328088601,3.431613734890872
,4.631461789301221,4.720341610439475,3.481962986552131,11.318153625715299,6.2649232
85654086,6.186149738223111,9.514198884803207)

### recall our fractional factorial design 8 factors in 16 runs from above
#my.design.by.hand
D.one <- desnum(my.design.by.hand) # Extract the design.
data1 = data.frame(D.one,"range" = range)
data1 = data.frame(D.one, "range" = range, "consumption" = cons)
~~~

~~~
###Again, we are using A = alpha, B = beta, C = Aq, D = d, E = Mt, F = Mf, G = pin, and H =
dzul

data1

data.frame(des1, "range" = range, "consumption" = cons)
```

```{r Question 5 -- Building the Models and Analysis for Range}

############# full model for range
m1.full.range = lm(range~A*B*C*D*E*F*G*H,data=data1)
#summary(m1.full)
DanielPlot(m1.full.range,half=TRUE)

############ reduced model with the significant effects
m1.red.range = lm(range~A+D,data=data1)
summary(m1.red.range)
par(mfrow=c(2,2))
plot(m1.red.range, which = c(1,2))

#estimated effects for transformed reduced model
2*coef(m1.red.range)[-1]
```
```{r Question 5 -- Building the Models and Analysis for Consumption}
############# full model
m1.full.cons = lm(cons~A*B*C*D*E*F*G*H,data=data1)
#summary(m1.full.cons)
DanielPlot(m1.full.cons,half=TRUE)

############ reduced model with important effects
m1.red.cons = lm(cons~A:B+G+C,data=data1)
summary(m1.red.cons)
par(mfrow=c(2,2))
plot(m1.red.cons, which = c(1,2))

2*coef(m1.red.cons)[-1]

```



```{r Question 5 Checkin Assumptions and Applying transformations}
#Variance
ncvTest(m1.red.range)
ncvTest(m1.red.cons)
#Normality
~~~

~~~~
shapiro.test(m1.red.range$residuals)
shapiro.test(m1.red.cons$residuals)
#Residual/Run Order correlation
plot(1:16,m1.red.cons$residuals,
        main = "Residuals vs Order",
        xlab = "Order",
        ylab = "Residuals")
  plot(1:16,m1.red.range$residuals,
        main = "Residuals vs Order",
        xlab = "Order",
        ylab = "Residuals")

bc <- boxcox(m1.red.cons)
lambda <- bc$x[which.max(bc$y)]
lambda
boxcoxTrans <- function(x, lam) {
  if (lam == 0L) {
        log(x)
  } else {
        (((x)^lam) - 1) / lam
  }
}

data2 <- cbind(data1, trans_cons = boxcoxTrans(data1$consumption, lambda))
m1.red.cons.t <- lm(trans_cons ~ A:B + G + C, data = data2)
summary(m1.red.cons.t)
ncvTest(m1.red.cons.t)
shapiro.test(m1.red.cons.t$residuals)
plot(m1.red.cons.t, which  = 1:2)
plot(1:16,m1.red.cons.t$residuals,
        main = "Residuals vs Order",
        xlab = "Order",
        ylab = "Residuals")
```
```{r Question 7 -- Optimal Settings}
## here, we use our regression formulas to estimate range and consumption for each combination
of alpha, beta, dzul, pin, etc.

#high and low settings for each factor
poss = c(-1,1)

#range estimate function
y.range = function(A,D){
  co = as.numeric(coef(m1.red.range))
  est = co[1] + co[2]*A + co[3]*D
  est
~~~~

~~~~
}

#consumption estimate function
y.cons = function(G,C,A,B){
  co = as.numeric(coef(m1.red.cons.t))
  est = co[1]+co[2]*G+co[3]*C+co[4]*C*G
  est = (-1.797*est + 1)^(-1/1.897)
}

### estimates for range at each combination

est.range = c()
for (i in poss){
  for (j in poss){
    est = y.range(i,j)
    cat(" with A = ",i, " and D = ",j, " range = ",est,"\n")
    est.range = c(est.range,est)
  }
}

est.cons = c()
temp.g=c()
temp.c=c()
temp.a=c()
temp.b=c()
# estimates for consumption at each combination
for (i in poss){
  for (j in poss){
        est= y.cons(i,j)
        cat("with pin = ",i, " Aq = ", j," consumption estimate = ",est,"\n")
        est.cons = c(est.cons,est)
        temp.g=c(temp.g,i)
        temp.c=c(temp.c,j)
      }
    }

est.cons.df = data.frame("pin"=temp.g,"Aq"=temp.c,"Consumption"=est.cons)
est.range.df = data.frame("Alpha"=c(-1,-1,1,1),"d"=c(-1,1,-1,1),"Range"=est.range)

#we see that all levels set to minimum maximize range and reduce consumption
```

```{r Confirmation Experiments}
conf.run =
matrix(c(0.0,0.0,2e-06,0.1,0.01,0.01,1.0,5.0,3.363990935607565,0.3044025394019873,0.0,0.0,2
e-06,0.1,0.01,0.01,1.0,5.0,3.418856323264561,0.22502517325678695,0.0,0.0,2e-06,0.1,0.01,0.0
~~~~

~~~
1,1.0,5.0,3.3542448630890838,0.3130664040495763,0.0,0.0,2e-06,0.1,0.01,0.01,1.0,5.0,3.1643
79667763956,0.3065291467899396), byrow=TRUE, nrow = 4)

conf.exp = data.frame("Run"=1:4,"Consumption"=conf.run[,9],"Range"=conf.run[,10])
colMeans(conf.exp)
# generate design for confirmation experiments
my.design <- FrF2(4, 3, generators = c("AB"), randomize = F)
cat("Generators of the design \n")
generators(my.design)
cat("Alias structure \n")
design.info(my.design)$aliased
```
```{r final estimated effects}
est.eff.range = coef(m1.red.range)[-1]*2
cons.est = coef(m1.red.cons.t)[-1]*2
est.eff.cons = (-1.797*cons.est+1)^(-1/1.897)

est.eff.range
est.eff.cons
```






```
~~~