Ben Brill, Emma Chi, Cecile Wu
Professor Vazquez
Statistics 101B Lecture 3
5 June 2022

Final Project Report

1. Introduction

This project explores tuning parameters of random forest in order to collect data. Our goal is to find the most important tuning parameters of random forest that can best facilitate cross-validation accuracy, ideally a higher accuracy. Cross-validation splits a dataset into two subsets in order to compare the two. One is used to build a random forest and the other is used to evaluate the algorithm. Our auxiliary dataset is cardiovascular disease data – we will use this data for cross-validation. Before that, we will build a design for the tuning parameters using different techniques and deciding on one for cross-validation. We looked at both a fractional factorial design and an experimental design constructed using the optimal design approach before collecting data and creating a final model.

2. Methodology

Experimental Design

Question 1:

Since there are 7 total factors, we will conduct a 2^{7-2} fractional factorial design with a resolution of IV. This means that we will have 32 runs. The generators for this design are $I = ABC = ABDE$. As the vast majority of our factors are quantitative (ntree, nodesize, cutoff, maxnodes, classwt, cutoff) so we must devise some sort of way to classify the high (+) and low (-) levels for each of these factors. To this end, we will classify the low level as the minimum values of each factor and the high levels as the maximum values for each of these factors. For the factor "replace," the high level will be 1 and the low level will be 0. The values for each level are specified in **Table 1** located in the Appendix.

Some might propose a three level design for the quantitative factors, as to include a wider range of possibilities of each of the factors. However, with a 35 run limit, this would not be a feasible design to appropriately measure the main-effect and two-level interactions. Another possibility could be mixing in three level factors including low, high, and center points along the range of certain factors and include two level factors for others. However, these mixed level designs should be used with caution as it can be difficult to analyze aliasing and interaction effects (Montgomery 2013, p. 412).

As six of the seven factors are quantitative, we can use a two-level design with center point runs (Montgomery 2013, p. 412). Center-point runs are runs of an experiment where factors are set to the midpoint between their low and high values. It is generally recommended to add 3 to 5 center-point runs to a fractional factorial design in order to process instability (). We will use 3 center-point runs in addition to the 32 runs generated by our fractional factorial design. Center points will be generated for the six continuous factors. Two of these center-point runs will use a value of 0 for "replace" – the singular discrete factor – and one will use a value of 1. These values were generated using a random number generator to determine which "replace" value would be repeated.

The fractional factorial design was built with the FrF2 library in R, were 32 runs were generated on a two level basis. The fractional factorial design is shown in the Appendix in **Table 2**.

In addition, we generated an optimally designed experiment using the optFederov function from AlgDesign library in R. This algorithm will find the combination of factors that will best maximize the

D-optimality criterion for the model. We generated this design using all 35 runs and 100 trials. The optimal design can be found in the Appendix in **Table 3**.

Question 2:

The fractional factorial design is Resolution IV, which is good because resolutions below III are not useful and resolutions above V are wasteful and have no practical benefit in most cases. The motivation for fractional factorial design is that it can be difficult when the size of the design grows very quickly with more and more factors, and we can see that 7 factors can be a lot. While fractional factorial design can estimate main effects well because there is no aliasing between them and other main effects or two-parameter interaction, fractional factorial design cannot estimate all interactions because some are aliased with each other. Optimal design involves fractions that allow all main effects to be estimated and as many two-factor interactions as possible, so as to budget efficiently when a full factorial design is not feasible. Optimal design can estimate all main effects and two-parameter interactions but this comes with slight correlation and multicollinearity between main effects and two-parameter interactions. Therefore, optimal design has more multicollinearity than fractional factorial design. We looked at the correlation plots for each design (shown in the Appendix) and found that optimal design did not have aliasing in two-parameter interactions while fractional factorial did. However, the main effects had multicollinearity in optimal design while the main effects in fractional factorial did not have multicollinearity or aliasing, making fractional factorial more useful for analyzing main effects.

Question 3:

We decided to go with the fractional factorial design. Based on the correlation plots, we are left with two options. The fractional factorial design (**Figure 1** in the Appendix) has aliasing between some multi-factor interactions but not between the main effects themselves; since there are colors on the non diagonals, there is correlation between the main effects and interactions, and among interactions, but the aliasing structure of the design for the model involving only the main effects is excellent, as these elements are all white and correspond to zero correlations. Since the interactions among main effects have no correlation, there is no multicollinearity. The optimal design, however, has multicollinearity in almost all main effects and interactions (**Figure 2** in the Appendix). In addition, when we calculated the VIF values for each model, several of the VIF values for the optimal design model were close to 5, further indicating the presence of multicollinearity in this model. Since the main effects are what we want to focus on, we opted for the fractional factorial design where the main effects are not aliased and do not have multicollinearity.

Question 4:

Our experimental design is not as robust as the one created by commercial software. The commercial software has 3 levels for all factors except one factor, “replace.” This is known as mixed-level; some factors are at different levels than other factors. This is seen to be done when there’s a mix of qualitative and quantitative factors – replace is the one qualitative factor in this case (Montgomery 2013, p. 412). In this way, their experimental design is more advanced than ours. The commercial software design has 22 runs, while we opted for more runs (32). In regards to multicollinearity, our recommended design has no severe multicollinearity since the elements on the correlation plot corresponding to the interactions between main effects are white and thus indicate zero correlation. The correlation plot for the commercial experimental design is an optimal design, because the number of runs

is 22 (not a power of 2), so there is multicollinearity. We ran an optimal design with 6 factors of 3 levels (**Figure 3** in the Appendix) and the correlation plot shows colors on the non-diagonals and also a few darker colors, indicating more multicollinearity and aliasing between main effects and interactions. Overall, our design is different from the commercial software because it is fractional factorial, does not include 3 levels and uses center points instead, and has more runs.

Data Analysis

First, we created an argument design that swapped the 1 and -1 with the actual tuning parameter levels. We used the cv.rf function to collect data from the design, with 35 runs. 32 runs were generated from the fractional factorial design and the additional 3 runs were the center-point runs.

We ran a Half Normal Plot (**Figure 4** in the Appendix) to determine which parameters have a significant influence on our model created with a fractional factorial design; looking at these plots, we can see that the factors classwt, mtry:maxnodes, nodesize:classwt, nodesize, nodesize:cutoff, mtry, maxnodes, cutoff:maxnodes, mtry:nodesize, classwt:cutoff, cutoff, classwt:maxnodes are significant and thus influential tuning parameters. However, we wanted to cut down to the most significant factors. We used the plot to distinguish the factors furthest from the line and tested combinations, eliminating those that were not significant in the summary function. We ended up with a model that included classwt, cutoff, classwt:maxnodes, and classwt:cutoff.

The final model is: CV = 0.9319 - 0.5456(classwt) - 0.2270(cutoff) - 6.867x 10^{-5}(classwt:maxnodes) - 0.4642(classwt:cutoff)

After creating our final model using only these influential parameters, we then conducted residual analysis to assess the validity of our model (**Figure 5** in the Appendix). Looking at the normal QQ-Plot, the points appear to be on or near the line with no significant departures from the line, and so the assumption of normality is satisfied since the residuals are normally distributed. We can also say that the assumption of constant variance is satisfied since there do not appear to be any significant patterns in the Residuals vs. Fitted plot. Thus, since the assumptions of normality and constant variance are satisfied, the model is adequate and provides a good fit to the data.

3. Conclusion

Our model is strong because we investigated center points in our experiment in order to address some of the larger quantitative ranges, since six of the seven factors we started with were quantitative. In addition, we also took several measures, including correlation plots and calculating VIFs for each model, to investigate multicollinearity and ensure that the model we chose had no multicollinearity. After conducting these investigations to strengthen our model, we made sure that the Normal QQ Plot and the Residuals vs. Fitted Plots satisfied the assumptions of normality and constant variance, respectively. However, the half normal plot produced by our model showed that many of the factors were significant and that the two-factor interactions included all main effects. We chose to only include the top four most influential factors, however, so this may have weakened our model compared to if we were to include each of the factors that appeared in the half normal plot. Furthermore, we could have made some factors three levels to account for larger ranges of potential factors.

4. Appendix

Parameter	**+1**	**-1**
ntree	100	1000
replace	0	1
mtry	2	6
nodesize	1	11
maxnodes	10	1000
classwt	0.5	0.9
cutoff	0.2	0.8

Table 1. The corresponding values for parameters for low and high levels

run	*ntree*	*mtry*	*replace*	*nodesize*	*classwt*	*cutoff*	*maxnodes*
1	−1	−1	−1	−1	−1	−1	1
2	1	−1	−1	−1	−1	1	−1
3	−1	1	−1	−1	−1	1	−1
4	1	1	−1	−1	−1	−1	1
5	−1	−1	1	−1	−1	1	1
6	1	−1	1	−1	−1	−1	−1
7	−1	1	1	−1	−1	−1	−1
8	1	1	1	−1	−1	1	1
9	−1	−1	−1	1	−1	−1	−1
10	1	−1	−1	1	−1	1	1
11	−1	1	−1	1	−1	1	1
12	1	1	−1	1	−1	−1	−1
13	−1	−1	1	1	−1	1	−1
14	1	−1	1	1	−1	−1	1
15	−1	1	1	1	−1	−1	1
16	1	1	1	1	−1	1	−1
17	−1	−1	−1	−1	1	−1	−1
18	1	−1	−1	−1	1	1	1
19	−1	1	−1	−1	1	1	1
20	1	1	−1	−1	1	−1	−1
21	−1	−1	1	−1	1	1	−1
22	1	−1	1	−1	1	−1	1
23	−1	1	1	−1	1	−1	1
24	1	1	1	−1	1	1	−1
25	−1	−1	−1	1	1	−1	1
26	1	−1	−1	1	1	1	−1
27	−1	1	−1	1	1	1	−1
28	1	1	−1	1	1	−1	1
29	−1	−1	1	1	1	1	1
30	1	−1	1	1	1	−1	−1
31	−1	1	1	1	1	−1	−1
32	1	1	1	1	1	1	1

Table 2. A summary of the proposed Fractional Factorial Design

run	*ntree*	*mtry*	*replace*	*nodesize*	*classwt*	*cutoff*	*maxnodes*
2	1	−1	−1	−1	−1	−1	−1
4	1	1	−1	−1	−1	−1	−1
8	1	1	1	−1	−1	−1	−1
9	−1	−1	−1	1	−1	−1	−1
11	−1	1	−1	1	−1	−1	−1
15	−1	1	1	1	−1	−1	−1
17	−1	−1	−1	−1	1	−1	−1
22	1	−1	1	−1	1	−1	−1
30	1	−1	1	1	1	−1	−1
39	−1	1	1	−1	−1	1	−1
41	−1	−1	−1	1	−1	1	−1
44	1	1	−1	1	−1	1	−1
49	−1	−1	−1	−1	1	1	−1
54	1	−1	1	−1	1	1	−1
55	−1	1	1	−1	1	1	−1
60	1	1	−1	1	1	1	−1
62	1	−1	1	1	1	1	−1
63	−1	1	1	1	1	1	−1
69	−1	−1	1	−1	−1	−1	1
72	1	1	1	−1	−1	−1	1
77	−1	−1	1	1	−1	−1	1
81	−1	−1	−1	−1	1	−1	1
83	−1	1	−1	−1	1	−1	1
90	1	−1	−1	1	1	−1	1
92	1	1	−1	1	1	−1	1
95	−1	1	1	1	1	−1	1
98	1	−1	−1	−1	−1	1	1
99	−1	1	−1	−1	−1	1	1
102	1	−1	1	−1	−1	1	1
108	1	1	−1	1	−1	1	1
109	−1	−1	1	1	−1	1	1
110	1	−1	1	1	−1	1	1
116	1	1	−1	−1	1	1	1
119	−1	1	1	−1	1	1	1
121	−1	−1	−1	1	1	1	1

Table 3. A summary of the proposed optimal design

run	*ntree*	*mtry*	*replace*	*nodesize*	*classwt*	*cutoff*	*maxnodes*	**CV**
1	100.00	2.00	0.00	11.00	0.50	0.80	10.00	0.69
2	1000.00	2.00	0.00	11.00	0.90	0.20	10.00	0.50
3	100.00	2.00	1.00	11.00	0.90	0.20	10.00	0.50
4	1000.00	2.00	1.00	11.00	0.50	0.80	10.00	0.70
5	100.00	6.00	0.00	11.00	0.90	0.80	10.00	0.55
6	1000.00	6.00	0.00	11.00	0.50	0.20	10.00	0.71
7	100.00	6.00	1.00	11.00	0.50	0.20	10.00	0.71
8	1000.00	6.00	1.00	11.00	0.90	0.80	10.00	0.57
9	100.00	2.00	0.00	11.00	0.50	0.20	10.00	0.66
10	1000.00	2.00	0.00	11.00	0.90	0.80	10.00	0.51
11	100.00	2.00	1.00	11.00	0.90	0.80	10.00	0.51
12	1000.00	2.00	1.00	11.00	0.50	0.20	10.00	0.67
13	100.00	6.00	0.00	11.00	0.90	0.20	10.00	0.50
14	1000.00	6.00	0.00	11.00	0.50	0.80	10.00	0.72
15	100.00	6.00	1.00	11.00	0.50	0.80	10.00	0.72
16	1000.00	6.00	1.00	11.00	0.90	0.20	10.00	0.50
17	100.00	2.00	0.00	11.00	0.50	0.20	1000.00	0.68
18	1000.00	2.00	0.00	11.00	0.90	0.80	1000.00	0.72
19	100.00	2.00	1.00	11.00	0.90	0.80	1000.00	0.72
20	1000.00	2.00	1.00	11.00	0.50	0.20	1000.00	0.67
21	100.00	6.00	0.00	11.00	0.90	0.20	1000.00	0.54
22	1000.00	6.00	0.00	11.00	0.50	0.80	1000.00	0.64
23	100.00	6.00	1.00	11.00	0.50	0.80	1000.00	0.63
24	1000.00	6.00	1.00	11.00	0.90	0.20	1000.00	0.54
25	100.00	2.00	0.00	11.00	0.50	0.80	1000.00	0.68
26	1000.00	2.00	0.00	11.00	0.90	0.20	1000.00	0.50
27	100.00	2.00	1.00	11.00	0.90	0.20	1000.00	0.50
28	1000.00	2.00	1.00	11.00	0.50	0.80	1000.00	0.67
29	100.00	6.00	0.00	11.00	0.90	0.80	1000.00	0.71
30	1000.00	6.00	0.00	11.00	0.50	0.20	1000.00	0.67
31	100.00	6.00	1.00	11.00	0.50	0.20	1000.00	0.66
32	1000.00	6.00	1.00	11.00	0.90	0.80	1000.00	0.71
33	550.00	4.00	1.00	6.00	0.70	0.50	505.00	0.72
34	550.00	4.00	0.00	6.00	0.70	0.50	505.00	0.72
35	550.00	4.00	1.00	6.00	0.70	0.50	505.00	0.72

Table 4. The resulting cross validation accuracies for given parameters of the random forest function

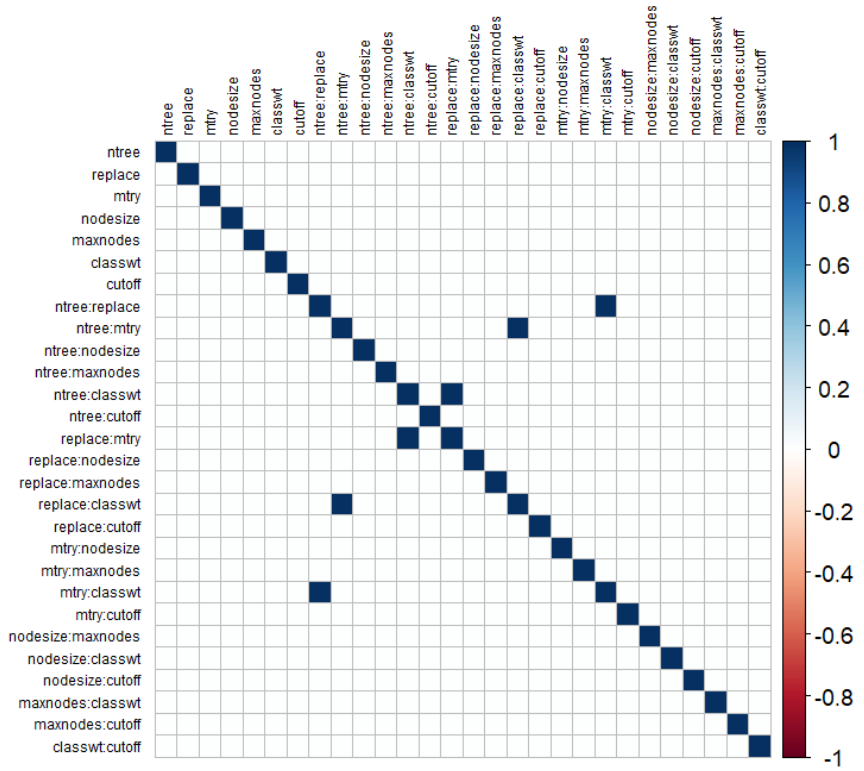

Figure 1. Correlation plot for the Fractional Factorial Design

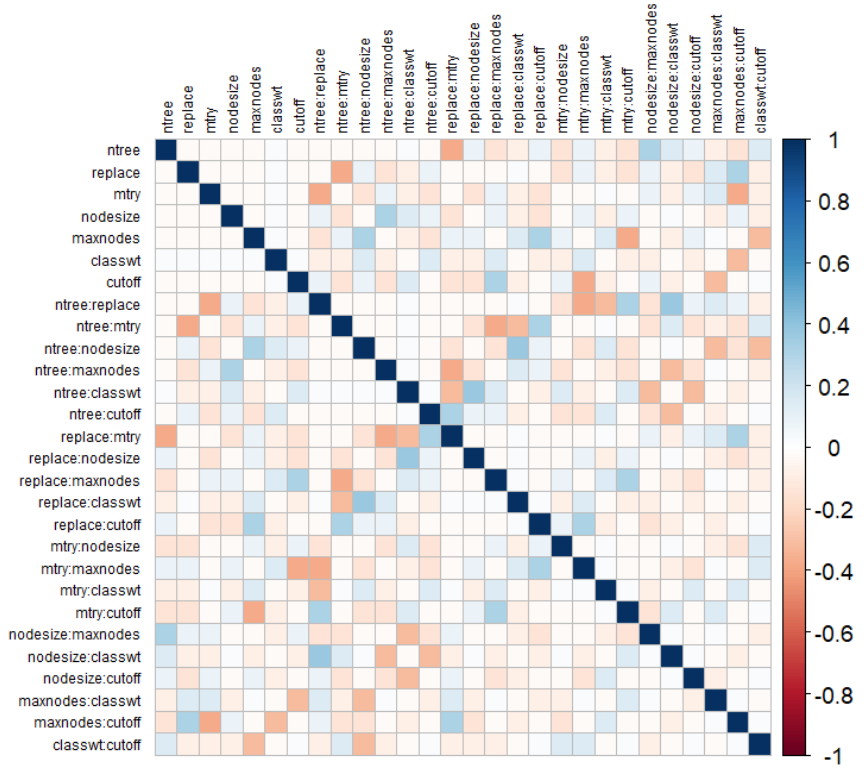

Figure 2. Correlation plot for the Optimal Design

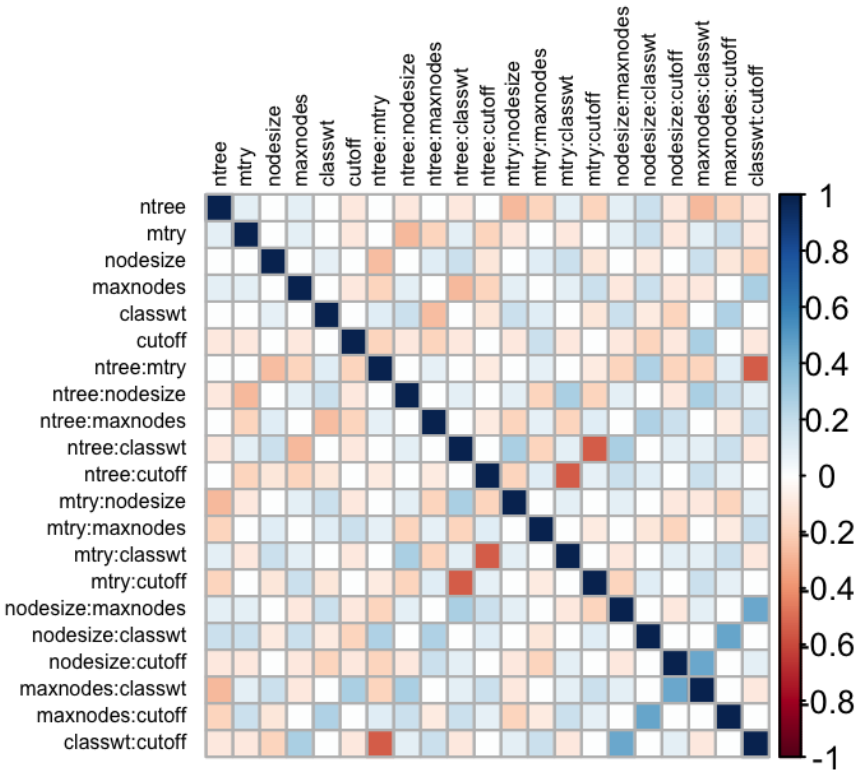

Figure 3. Correlation plot for optimal design with 6 factors at 3 levels

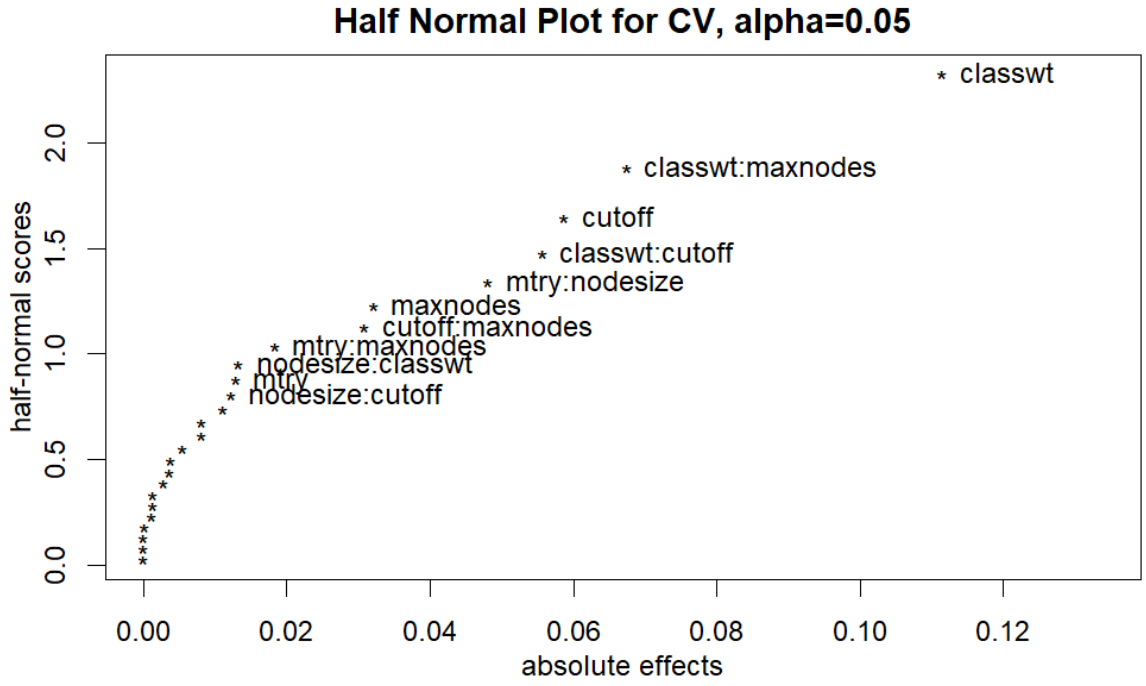

Figure 4. Half Normal Probability Plot for Fractional Factorial Design

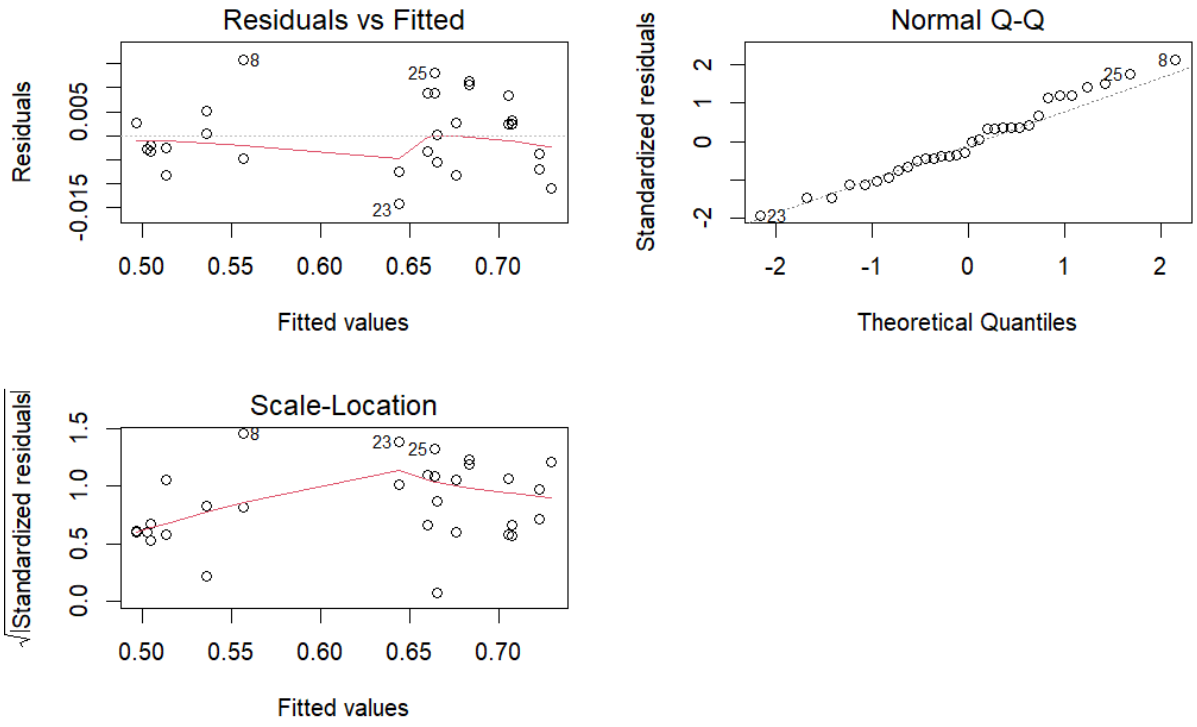

Figure 5. Residual plots for Fractional Factorial Design

Code Appendix

```
```{r}
# load data and functions
load("cardiovascular.Rdata")
source("CrossValidation_RF.R")
```

Fractional Factorial Design

```{r}
library(FrF2)
frfact.design <- FrF2(nruns = 32, nfactors = 7, randomize = F, factor.names =
c("ntree", "replace", "mtry", "nodesize", "maxnodes", "classwt", "cutoff"))
D.two <- desnum(frfact.design) # Extract the design.
print(D.two)
```

```{r}
cat("Generators of the design \n")
generators(frfact.design)
cat("Alias structure \n")
design.info(frfact.design)$aliased
```
```

```
cat("Resolution and word length pattern \n")
design.info(frfact.design)$catlg.entry
```

```{r}
# generate correlation plots
library(corrplot)
X.two <- model.matrix(~(.)^2-1, data.frame(D.two))

# Create color map on pairwise correlations.
contrast.vectors.correlations.two <- cor(X.two)
corrplot(contrast.vectors.correlations.two, type = "full", addgrid.col = "gray",
         tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.3)
```

Optimal Design

```{r}
library(AlgDesign)
candidate.set <- gen.factorial(levels=2, nVars = 7,
                               varNames = c("ntree", "replace", "mtry", "nodesize",
"maxnodes", "classwt", "cutoff"))
alternative.design <- optFederov(~ntree + replace + mtry+ nodesize + maxnodes +
classwt + cutoff, candidate.set, nTrials = 35, nRepeats = 100)
print.data.frame(alternative.design$design)
```

```{r}
X.two <- model.matrix(~(.)^2-1, data.frame(alternative.design$design))

# Create color map on pairwise correlations.
contrast.vectors.correlations.two <- cor(X.two)
corrplot(contrast.vectors.correlations.two, type = "full", addgrid.col = "gray",
         tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.5)
```

```{r}
# find VIFs
X.opt.me <- model.matrix(~(.)^2, data.frame(data.frame(alternative.design$design)))
XtX <- t(X.opt.me) %*% X.opt.me
inv.XtX <- solve(XtX)
var.eff <- diag(inv.XtX)
print(nrow(data.frame(alternative.design$design))*var.eff)
```

Comparing with Commercial Software

```{r}
```

```
candidate.set2 <- gen.factorial(levels=3, nVars = 6,
                                varNames = c("ntree", "mtry", "nodesize", "maxnodes",
"classwt", "cutoff"))
comm.opt <- optFederov(~ntree + mtry + nodesize + maxnodes + classwt + cutoff,
candidate.set2, nTrials = 22, nRepeats = 100)
print.data.frame(comm.opt$design)

library(corrplot)
X.three <- model.matrix(~(.)^2-1, data.frame(comm.opt$design))

# Create color map on pairwise correlations.
contrast.vectors.correlations.three <- cor(X.three)
corrplot(contrast.vectors.correlations.three, type = "full", addgrid.col = "gray",
         tl.col = "black", tl.srt = 90, method = "color", tl.cex=0.5)
```


Creating Argument Design
```{r}
# Creating argument design from fractional factorial design
argumentdesign <- data.frame(D.two)
argumentdesign$ntree[argumentdesign$ntree == -1] <- 100
argumentdesign$ntree[argumentdesign$ntree == 1] <- 1000

argumentdesign$nodesize[argumentdesign$nodesize == -1] <- 1
argumentdesign$nodesize[argumentdesign$nodesize == 1] <- 11

argumentdesign$cutoff[argumentdesign$cutoff == -1] <- 0.2
argumentdesign$cutoff[argumentdesign$cutoff == 1] <- 0.8

argumentdesign$replace[argumentdesign$replace == -1] <- 0
argumentdesign$replace[argumentdesign$replace == 1] <- 1

argumentdesign$mtry[argumentdesign$mtry == -1] <- 2
argumentdesign$mtry[argumentdesign$mtry == 1] <- 6

argumentdesign$maxnodes[argumentdesign$maxnodes == -1] <- 10
argumentdesign$maxnodes[argumentdesign$maxnodes == 1] <- 1000

argumentdesign$classwt[argumentdesign$classwt == -1] <- 0.5
argumentdesign$classwt[argumentdesign$classwt == 1] <- 0.9

argumentdesign <- data.frame("ntree" = argumentdesign$ntree, "mtry" =
argumentdesign$mtry, "replace" = argumentdesign$replace, "nodesize" = rep(c(rep(1,
8), rep(11, 8)), 2), "classwt" = argumentdesign$classwt, "cutoff" =
argumentdesign$cutoff, "maxnodes" = argumentdesign$maxnodes)
```

```{r}
# add central points
argumentdesign[nrow(argumentdesign) + 1,] <- c(550, 4, 1, 6, 0.7, 0.5, 505.0)
argumentdesign[nrow(argumentdesign) + 1,] <- c(550, 4, 0, 6, 0.7, 0.5, 505.0)
argumentdesign[nrow(argumentdesign) + 1,] <- c(550, 4, 1, 6, 0.7, 0.5, 505.0)
```

Run Experiment

```{r}
results <- cv.rf(argumentdesign, y, X)
```

Create Model

```{r}
df <- data.frame(D.two)
df <- argumentdesign[-c(33,34,35),]
df$CV <- results$CV[-c(33,34,35)] # remove central points for initial fitting
model1 <- lm(CV~ (ntree + mtry + replace + nodesize + classwt + cutoff + maxnodes)^2,
data = df, na.action = "na.omit")

par(mfrow=c(1,2))
DanielPlot(model1, half = T)
DanielPlot(model1, half = F)
summary(model1)
```

```{r}
df <- data.frame(D.two)
df$CV <- results$CV # re add central points for fitting
model_final <- lm(CV~ classwt + cutoff + classwt:maxnodes + classwt:cutoff , data =
df)
par(mfrow = c(2,2))
plot(model_final)
```

```{r}
summary(model_final)
```

5. Statement of Contribution

Developed code for experimental design and data analysis: Ben, Cecile, Emma
Wrote the report: Ben, Cecile, Emma
Made the appendix: Ben
Created presentation slides: Ben, Cecile, Emma
Presented: Ben, Cecile, Emma

6. References

Montgomery, D. C. (2013). Additional Design and Analysis Topics for Factorial and Fractional Factorial Designs. In Design and Analysis of Experiments (8th ed., pp. 394–444). John Wiley & Sons.