Design of Experiments - A3

Francis Frégeau

Winter 2021

Q1

```
suppressMessages(library("data.table"))
suppressMessages(library("dplyr"))
suppressMessages(library("ggplot2"))
suppressMessages(library("gridExtra"))
suppressMessages(library("reshape2"))
suppressMessages(library("glmnet"))
suppressMessages(library("plotly"))

data_path <- "C:/Users/frank/OneDrive/Documents/Assignments/DoE 5.24.csv"
data <- fread(data_path) %>%
            .[, Unit := NULL] %>%
            .[, (c("CycleTime", "Operator", "Temperature")) := lapply(.SD, function(x){as.factor(x)}),
                                                                                          .SDcols = c("CycleTime"


linear_model <- lm(Score ~., data)
SUMMARY <- summary(linear_model)
ANOVA <- anova(linear_model)

print(SUMMARY)
```

```
##
## Call:
## lm(formula = Score ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8148 -2.6759  0.2222  2.8519  8.5926
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     26.4815     1.1882  22.287  < 2e-16 ***
## CycleTime50      5.6667     1.1882   4.769 1.76e-05 ***
## CycleTime60     -0.6667     1.1882  -0.561   0.5774
## Operator2        5.3333     1.1882   4.489 4.49e-05 ***
## Operator3        2.0000     1.1882   1.683   0.0988 .
## Temperature350   1.9259     0.9702   1.985   0.0529 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.565 on 48 degrees of freedom
## Multiple R-squared:  0.5506, Adjusted R-squared:  0.5038
## F-statistic: 11.76 on 5 and 48 DF,  p-value: 1.88e-07
```

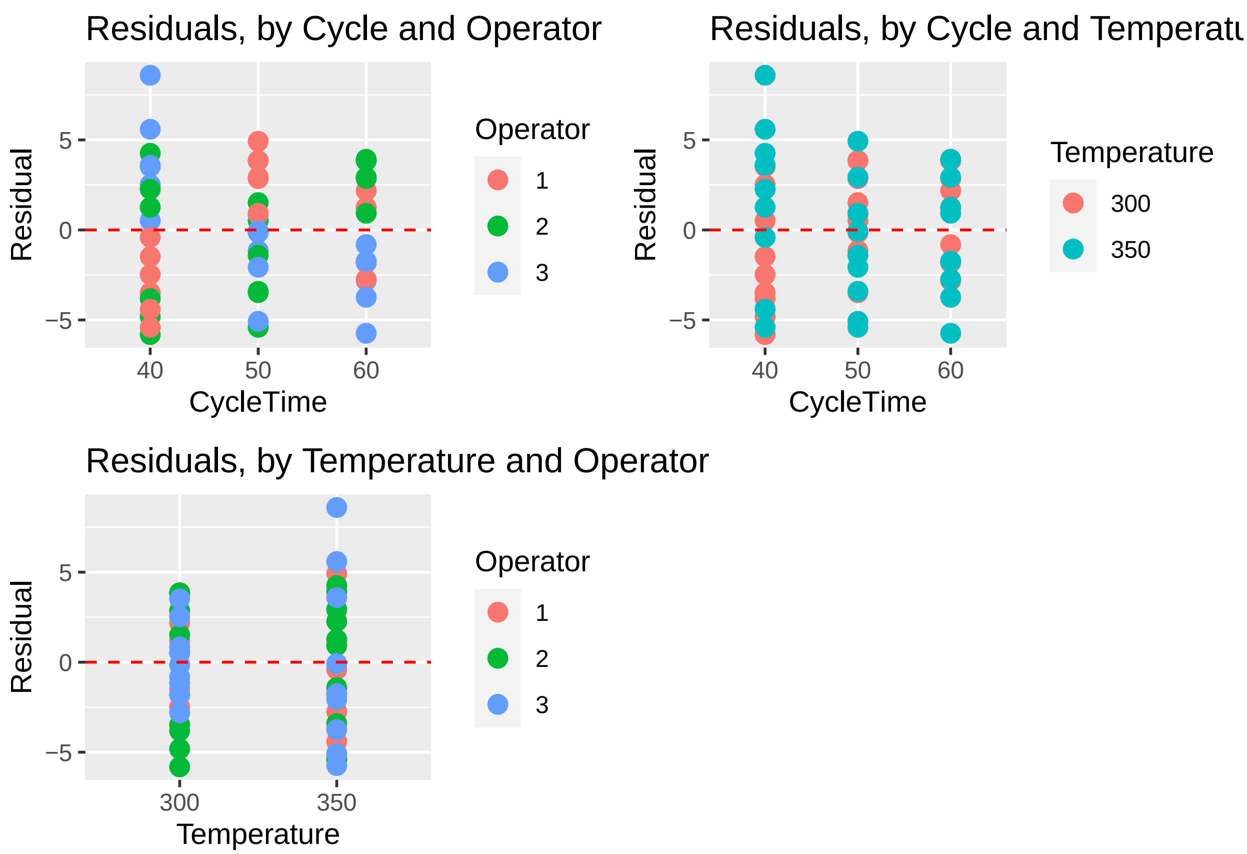
```r
print(ANOVA)
```

```
## Analysis of Variance Table
##
## Response: Score
##             Df Sum Sq Mean Sq F value    Pr(>F)
## CycleTime    2 436.00 218.000 17.1562 2.391e-06 ***
## Operator     2 261.33 130.667 10.2832 0.0001919 ***
## Temperature  1  50.07  50.074  3.9407 0.0528604 .
## Residuals   48 609.93  12.707
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Assumming $\alpha = 0.05$, the different temperature levels do not have a statistically significant effect on scores.

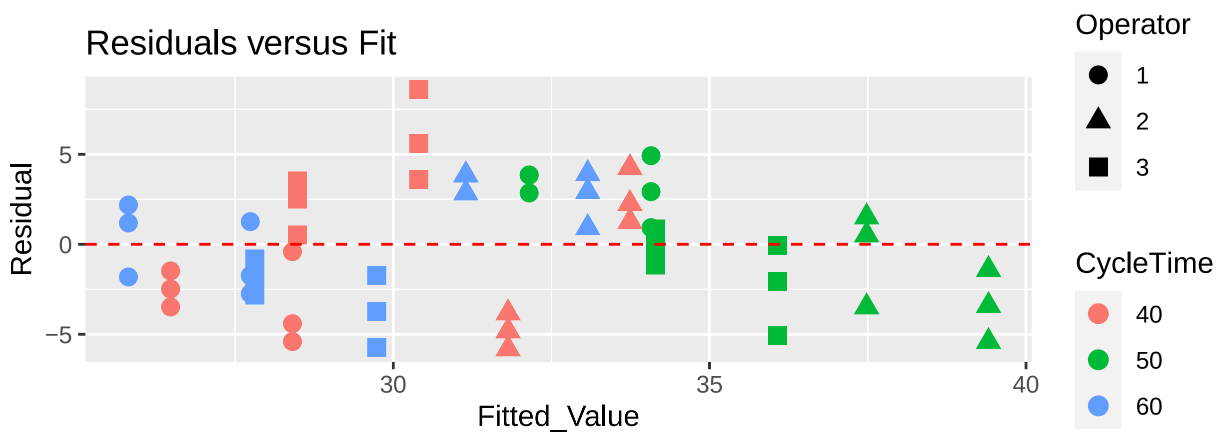
As far as relevant predictors are concerned, only $CycleTime = 50$ and $Operator = 2$ have a statistically significant effect on scores.

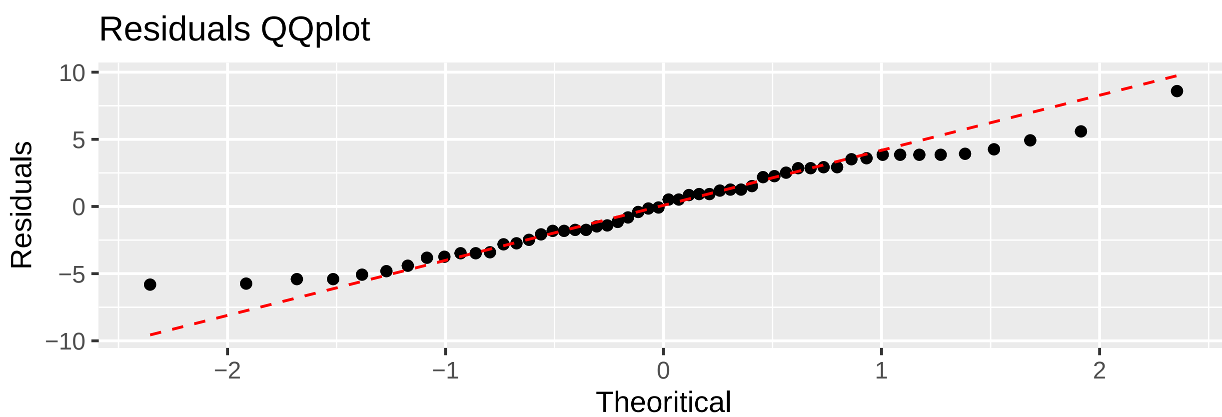
```r
#Residual plot
data[, Residual := linear_model$residuals] %>%
    .[, Fitted_Value := linear_model$fitted.values]


residual_plots <- list()
residual_plots$plot1 <- ggplot(data, aes(y = Residual, x = CycleTime, color = Operator)) + geom_point(si
                                                           geom_hline(yintercept = 0, color = "red", li
                                                           ggtitle("Residuals, by Cycle and Operator")

residual_plots$plot2 <- ggplot(data, aes(y = Residual, x = CycleTime, color = Temperature)) + geom_poin
                                                           geom_hline(yintercept = 0, color = "red", li
                                                           ggtitle("Residuals, by Cycle and Temperatur

residual_plots$plot3 <- ggplot(data, aes(y = Residual, x = Temperature, color = Operator)) + geom_point
                                                           geom_hline(yintercept = 0, color = "red", li
                                                           ggtitle("Residuals, by Temperature and Opera

gridExtra::grid.arrange(grobs = residual_plots, ncol=2, nrow=2)
```



The distribution of the residuals seems to be approximately the same accross all predictor levels.

```
normal_plots <- list()
normal_plots$plot1 <- ggplot(data, aes(y = Residual, x = Fitted_Value, color = CycleTime, shape = Opera
                                                                  geom_hline(yintercept = 0, color = "red
                                                                  ggtitle("Residuals versus Fit")


normal_plots$plot2 <- ggplot(data, aes(sample = Residual)) + stat_qq() +
                                        stat_qq_line(color = "red", linetype = "dashed") +
                                        ggtitle("Residuals QQplot") +
                                        xlab("Theoritical") +
                                        ylab("Residuals")


gridExtra::grid.arrange(grobs = normal_plots, ncol=1, nrow=2)
```





Deviation from normality doesn't seem like a cause for concern, and the residuals do not seem to be correlated with their associated fitted values. (Save perhaps for the far right hand side which seems to be more often negative than not.)

Q2

a)

```
data_path <- "C:/Users/frank/OneDrive/Documents/Assignments/DoE 5.30.csv"
data <- fread(data_path) %>%
            .[, Unit := NULL]

backup <- data[, .SD, .SDcols = names(data)]

#Compute conditional means before converting to factor
by_clauses <- list(a = "Doping",
                    b = "Anneal",
                    c = c("Doping", "Anneal"))

conditional_means <- lapply(by_clauses, function(x){data[, lapply(.SD, mean), by = x, .SDcols = "BaseCu

g <- function(x){

    copy <- x[, .SD, .SDcols = names(x)]

    if(ncol(x) > 2){

        copy[, Group := paste("D:", Doping, "A:", Anneal, sep = "")] %>%
            .[, Doping := NULL] %>%
            .[, Anneal := NULL]

    } else {

        names(copy)[which(names(copy) != "BaseCurrent")] <- "Group"

    }

    return(copy)

}

conditional_means <- lapply(conditional_means, g)

for(i in 1:3){conditional_means[[i]] <- g(conditional_means[[i]])}
names(conditional_means) <- c("Doping", "Anneal", "Doping and Anneal")


#Convert to factors, and run anova
data[, (c("Doping", "Anneal")) := lapply(.SD, function(x){as.factor(x)}), .SDcols = c("Doping", "Anneal

linear_model <- lm(BaseCurrent ~., data)
SUMMARY <- summary(linear_model)
ANOVA <- anova(linear_model)

print(SUMMARY)
```

```
##
## Call:
## lm(formula = BaseCurrent ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45583 -0.17229 -0.02458  0.22833  0.38917
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.2108     0.2001  21.039 2.74e-08 ***
## Doping2e+20  -0.5717     0.2001  -2.856   0.0213 *
## Anneal950     6.0125     0.2451  24.528 8.15e-09 ***
## Anneal1000    6.8250     0.2451  27.843 2.99e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3467 on 8 degrees of freedom
## Multiple R-squared:  0.9915, Adjusted R-squared:  0.9883
## F-statistic: 311.1 on 3 and 8 DF,  p-value: 1.279e-08
```

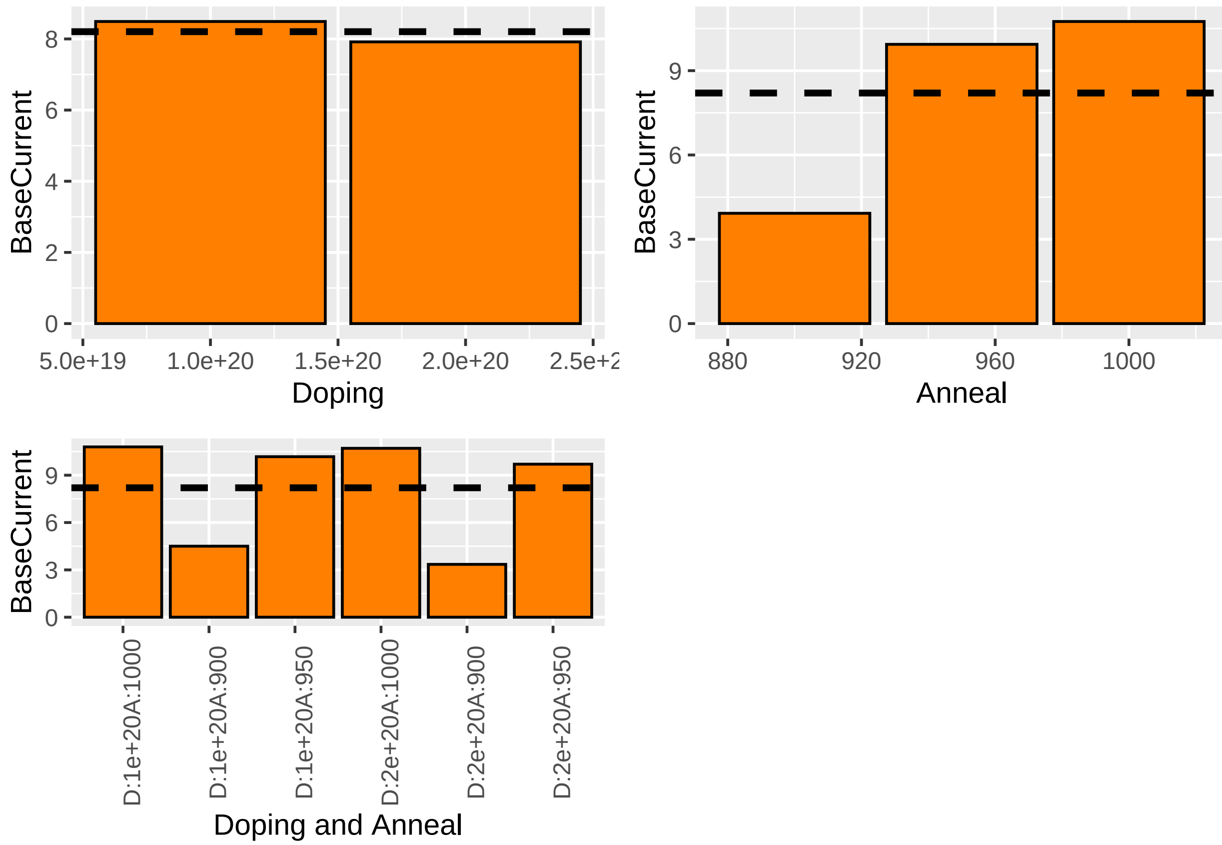
```
print(ANOVA)
```

```
## Analysis of Variance Table
##
## Response: BaseCurrent
##           Df  Sum Sq Mean Sq  F value  Pr(>F)
## Doping     1   0.980   0.980   8.1585 0.02127 *
## Anneal     2 111.188  55.594 462.6244 5.4e-09 ***
## Residuals  8   0.961   0.120
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both variables are significant at the $\alpha = 0.05$ level.

```
#Plot the conditional means
mu <- mean(data$BaseCurrent)
p <- function(x){

    out <- ggplot(x, aes(x = Group, y = BaseCurrent)) + geom_bar(stat = "identity", fill = "darkorange1
                                                        geom_hline(yintercept = mu, color = "black", li

}
conditional_means_plots <- lapply(conditional_means, p)
for(i in 1:length(conditional_means_plots)){

    conditional_means_plots[[i]] <- conditional_means_plots[[i]] + xlab(names(conditional_means)[[i]])
```

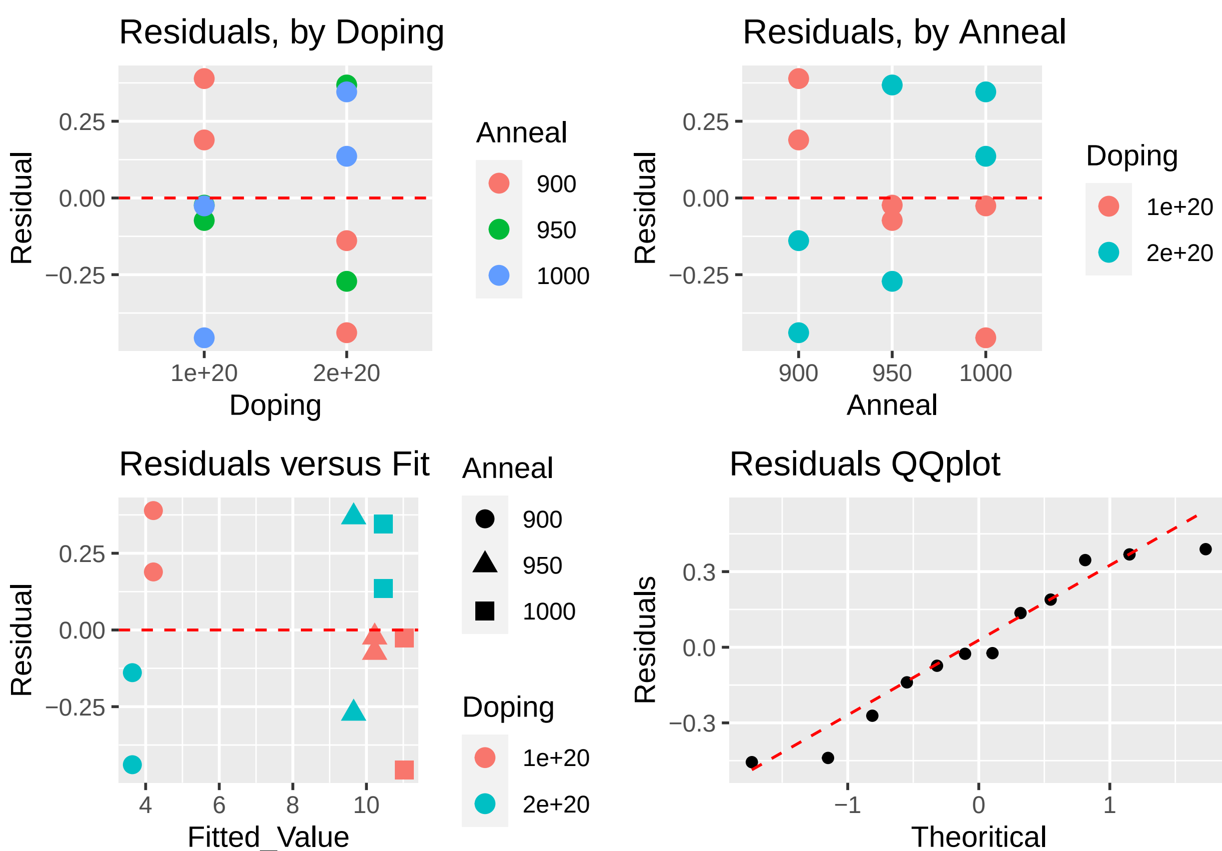
```
}
conditional_means_plots[[3]] <- conditional_means_plots[[3]] + theme(axis.text.x = element_text(angle =

gridExtra::grid.arrange(grobs = conditional_means_plots, ncol=2, nrow=2)
```



A lower level of anneal seems to be associated with a lower base current, while doping levels seem to bear almost no effect despite their statistical significance.

```
data[, Residual := linear_model$residuals] %>%
    .[, Fitted_Value := linear_model$fitted.values]


plots <- list()
plots$plot1 <- ggplot(data, aes(y = Residual, x = Doping, color = Anneal)) + geom_point(size = 3) +
                                                     geom_hline(yintercept = 0, color = "red", l
                                                     ggtitle("Residuals, by Doping")

plots$plot2 <- ggplot(data, aes(y = Residual, x = Anneal, color = Doping)) + geom_point(size = 3) +
                                                     geom_hline(yintercept = 0, color = "red", l
                                                     ggtitle("Residuals, by Anneal")
```

```
plots$plot3 <- ggplot(data, aes(y = Residual, x = Fitted_Value, color = Doping, shape = Anneal)) + geom_
                                                             geom_hline(yintercept = 0, color = "red
                                                             ggtitle("Residuals versus Fit")


plots$plot4 <- ggplot(data, aes(sample = Residual)) + stat_qq() +
                                       stat_qq_line(color = "red", linetype = "dashed") +
                                       ggtitle("Residuals QQplot") +
                                       xlab("Theoritical") +
                                       ylab("Residuals")

gridExtra::grid.arrange(grobs = plots, ncol=2, nrow=2)
```



The distribution of residuals looks invariant with respect to anneal and doping level, whilst deviation from normality doesn't look like a cause for concern. Also, the residuals do not appear to be correlated with their respective fitted values.

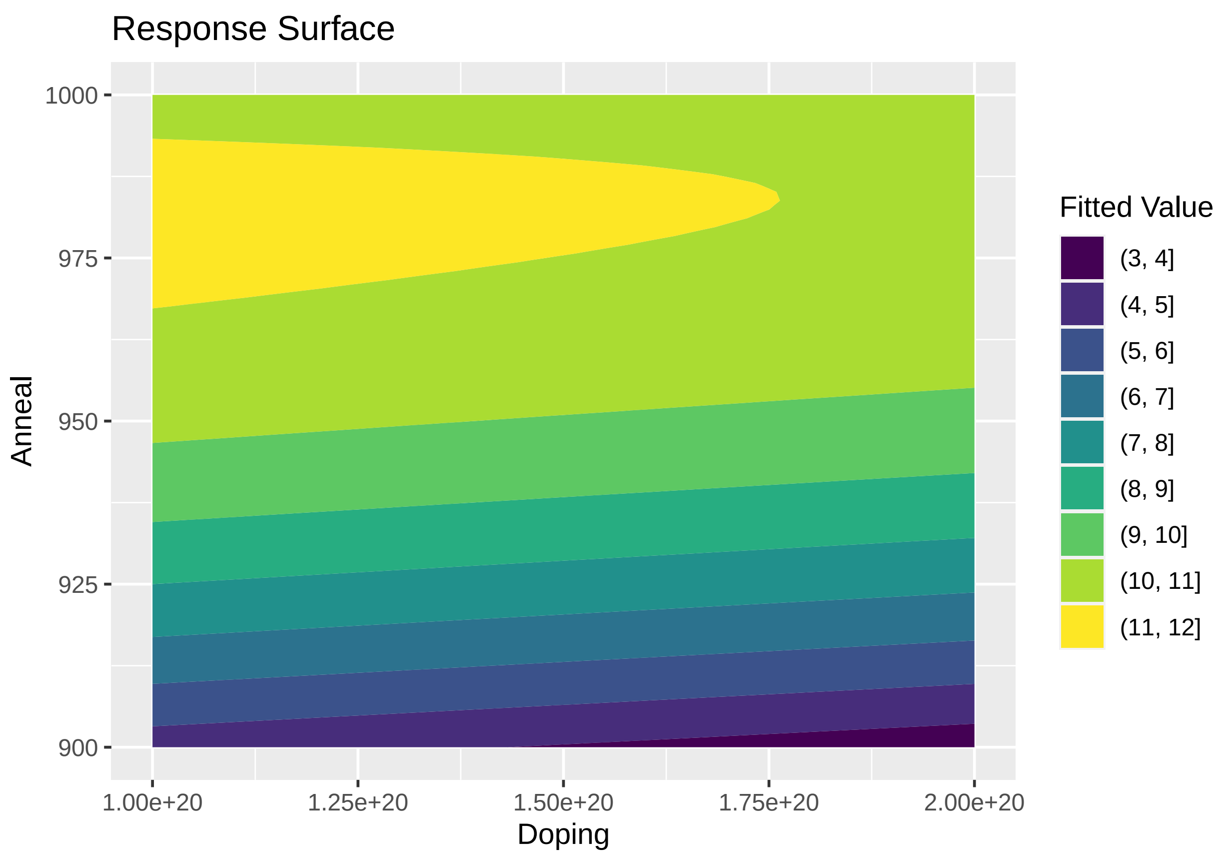
```
backup[, Anneal_Sq := Anneal^2] %>%
        .[, Doping_times_Anneal := Doping * Anneal]

linear_model <- lm(BaseCurrent ~., backup)
```

```r
SUMMARY <- summary(linear_model)

print(SUMMARY)
```

```
##
## Call:
## lm(formula = BaseCurrent ~ ., data = backup)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27167 -0.14042 -0.04833  0.12458  0.36833
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -9.775e+02  5.296e+01 -18.457 3.40e-07 ***
## Doping              -1.064e-19  3.213e-20  -3.312   0.0129 *
## Anneal               2.028e+00  1.113e-01  18.221 3.71e-07 ***
## Anneal_Sq           -1.040e-03  5.852e-05 -17.771 4.41e-07 ***
## Doping_times_Anneal  1.060e-22  3.379e-23   3.137   0.0164 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2389 on 7 degrees of freedom
## Multiple R-squared:  0.9965, Adjusted R-squared:  0.9944
## F-statistic: 493.7 on 4 and 7 DF,  p-value: 1.175e-08
```

All terms are significant at the $\alpha = 0.05$ level.

```r
#Draw response surface
b <- coef(linear_model)
x <- seq(min(backup$Doping), max(backup$Doping), length.out = 75)
y <- seq(min(backup$Anneal), max(backup$Anneal), length.out = 75)

xy <- expand.grid(x = x, y = y)
g <- function(i){as.numeric(b[1] + b[2]*xy[i,1] + b[3]*xy[i,2] + b[4]*xy[i,2]^2 + b[5]*xy[i,1]*xy[i,2])
contour_plot_frame <- as.data.table(cbind(xy, sapply(c(1:nrow(xy)), g)))

names(contour_plot_frame) <- c("Doping", "Anneal", "Fitted_Value")

ggplot(contour_plot_frame, aes(x = Doping, y = Anneal, z = Fitted_Value)) + geom_contour_filled() +
                                                                            ggtitle("Response Surface")
                                                                            labs(fill = "Fitted Value")
```


Response Surface
1000
975
950
925
900
Anneal
1.00e+20
1.25e+20
1.50e+20
1.75e+20
2.00e+20
Doping
Fitted Value
(3, 4]
(4, 5]
(5, 6]
(6, 7]
(7, 8]
(8, 9]
(9, 10]
(10, 11]
(11, 12]

Q3

a)

$118.667 - 10 - 12.167 - 96.333 = 0.167$

b)

$0.167/0.0833 \approx 2$

c)

$11 - 6 - 2 - 2 = 1$

d)

$10/6 \approx 1.667$

e)

```r
print(round(1 - pf(3.65, 2, 6), 4))
```

```
## [1] 0.0918
```

f)

$2 + 1 = 3$

g)

$1 + 1 = 2$

h)

$\frac{12}{2^2} = 3$

i)

Not at the $\alpha = 0.05$ level.

j)

$\sqrt{\frac{5}{3}} \approx 1.291$

Q4

By the bias-variance decomposition:

$$\mathbb{E}\left[bn\sum_{i=1}^{a}\left(\overline{y}_{i\cdot\cdot}-\overline{y}_{\cdots}\right)^2\right]=bn\sum_{i=1}^{a}\mathrm{Var}\left(\overline{y}_{i\cdot\cdot}-\overline{y}_{\cdots}\right)+bn\sum_{i=1}^{a}\mathbb{E}\left[\overline{y}_{i\cdot\cdot}-\overline{y}_{\cdots}\right]^2$$

First:

$$\mathrm{Var}\left(\overline{y}_{i\cdot\cdot}-\overline{y}_{\cdots}\right)=\mathrm{Var}\left(\overline{y}_{i\cdot\cdot}-\frac{1}{a}\sum_{u=1}^{a}\overline{y}_{u\cdot\cdot}\right)=\mathrm{Var}\left(\frac{a-1}{a}\overline{y}_{i\cdot\cdot}-\frac{1}{a}\sum_{u\neq i}^{a}\overline{y}_{u\cdot\cdot}\right)$$

$$=\frac{(a-1)^2}{a^2}\frac{\sigma^2}{bn}+\frac{a-1}{a^2}\frac{\sigma^2}{bn}=\frac{(a-1)\sigma^2}{abn}$$

$$\Rightarrow bn\sum_{i=1}^{a}\mathrm{Var}\left(\overline{y}_{i\cdot\cdot}-\overline{y}_{\cdots}\right)=abn\cdot\frac{(a-1)\sigma^2}{abn}=(a-1)\,\sigma^2$$

Then, assuming $\overline{y}_{i\cdot\cdot}\neq\overline{y}_{\cdots}$:

$$\mathbb{E}\left[\overline{y}_{i\cdot\cdot}-\overline{y}_{\cdots}\right]^2=\tau_i^2$$

Which completes the proof:

$$\mathbb{E}\left[bn\sum_{i=1}^{a}\left(\overline{y}_{i\cdot\cdot}-\overline{y}_{\cdots}\right)^2\right]=(a-1)\,\sigma^2+bn\sum_{i=1}^{a}\tau_i^2\ \Rightarrow\ \mathrm{MS}_a=\sigma^2+\frac{bn\sum_{i=1}^{a}\tau_i^2}{a-1}$$

Q5

a)

```
data_path <- "C:/Users/frank/OneDrive/Documents/Assignments/DoE 6.5.csv"
data <- fread(data_path) %>%
            .[, Unit := NULL]

#Convert to factor
f <- function(i){

    cname <- names(data)[i]
    levs <- unique(data[, cname, with = FALSE][[1]])
    return(as.factor(paste(cname, match(data[, cname, with = FALSE][[1]], levs), sep = "")))

}

for(i in 1:3){data[, (names(data)[i]) := f(i)]}

#Add interaction terms
variables <- names(data)[1:3]
for(i in 1:3){

    for(j in i:3){

        if(i == j){next}
        vals <- names(data)[c(i,j)]
        interactions_2way <- paste(data[, vals[1], with = FALSE][[1]],
                                   data[, vals[2], with = FALSE][[1]], sep = "")
        index <- which(interactions_2way == paste(vals[1], 2, vals[2], 2, sep = ""))
        interactions_2way[-index] <- paste("0_", vals[1], vals[2], sep = "")
        valname <- paste(vals[1], vals[2], sep = "")
        data[, (valname) := as.factor(interactions_2way)]


    }

    if(i == 3){

        vals <- names(data)[1:3]
        interactions_3way <- paste(data[, vals[1], with = FALSE][[1]],
                                   data[, vals[2], with = FALSE][[1]],
                                   data[, vals[3], with = FALSE][[1]], sep = "")
        index <- which(interactions_3way == paste(vals[1], 2, vals[2], 2, vals[3], 2, sep = ""))
        interactions_3way[-index] <- paste("0_", vals[1], vals[2], vals[3], sep = "")
        valname <- paste(vals[1], vals[2], vals[3], sep = "")
        data[, (valname) := as.factor(interactions_3way)]

    }

}
```

```r
linear_model <- lm(LifeHour ~., data)
linear_model_all <- linear_model
SUMMARY <- summary(linear_model)
ANOVA <- anova(linear_model)

print(round(SUMMARY$coefficients, 4))
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   26.0000     3.1710  8.1992   0.0000
## SS2            8.6667     4.4845  1.9326   0.0712
## GG2           13.6667     4.4845  3.0475   0.0077
## CC2           16.3333     4.4845  3.6421   0.0022
## SGS2G2         1.0000     6.3421  0.1577   0.8767
## SCS2C2       -13.3333     6.3421 -2.1024   0.0517
## GCG2C2        -1.3333     6.3421 -0.2102   0.8361
## SGCS2G2C2     -8.6667     8.9691 -0.9663   0.3483
```

Assumming $\alpha = 0.05$, only the interaction between S (speed) and G (geometry) seems to be statistically significant.

b)

```r
print(ANOVA)
```

```
## Analysis of Variance Table
##
## Response: LifeHour
##           Df Sum Sq Mean Sq F value    Pr(>F)
## S          1   0.67    0.67  0.0221 0.8836803
## G          1 770.67  770.67 25.5470 0.0001173 ***
## C          1 280.17  280.17  9.2873 0.0076787 **
## SG         1  16.67   16.67  0.5525 0.4680784
## SC         1 468.17  468.17 15.5193 0.0011722 **
## GC         1  48.17   48.17  1.5967 0.2244753
## SGC        1  28.17   28.17  0.9337 0.3482825
## Residuals 16 482.67   30.17
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

c)

```
data <- data[, c("S", "G", "C", "SC", "LifeHour"), with = FALSE]
linear_model <- lm(LifeHour ~., data)
print(summary(linear_model))
```

```
##
## Call:
## lm(formula = LifeHour ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3333 -4.3750 -0.4167  2.9583 11.5000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   27.167      2.512  10.813 1.47e-09 ***
## SS2            9.167      3.178   2.884 0.009497 **
## GG2           11.333      2.247   5.043 7.22e-05 ***
## CC2           15.667      3.178   4.930 9.30e-05 ***
## SCS2C2       -17.667      4.494  -3.931 0.000897 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.504 on 19 degrees of freedom
## Multiple R-squared:  0.7253, Adjusted R-squared:  0.6674
## F-statistic: 12.54 on 4 and 19 DF,  p-value: 3.688e-05
```

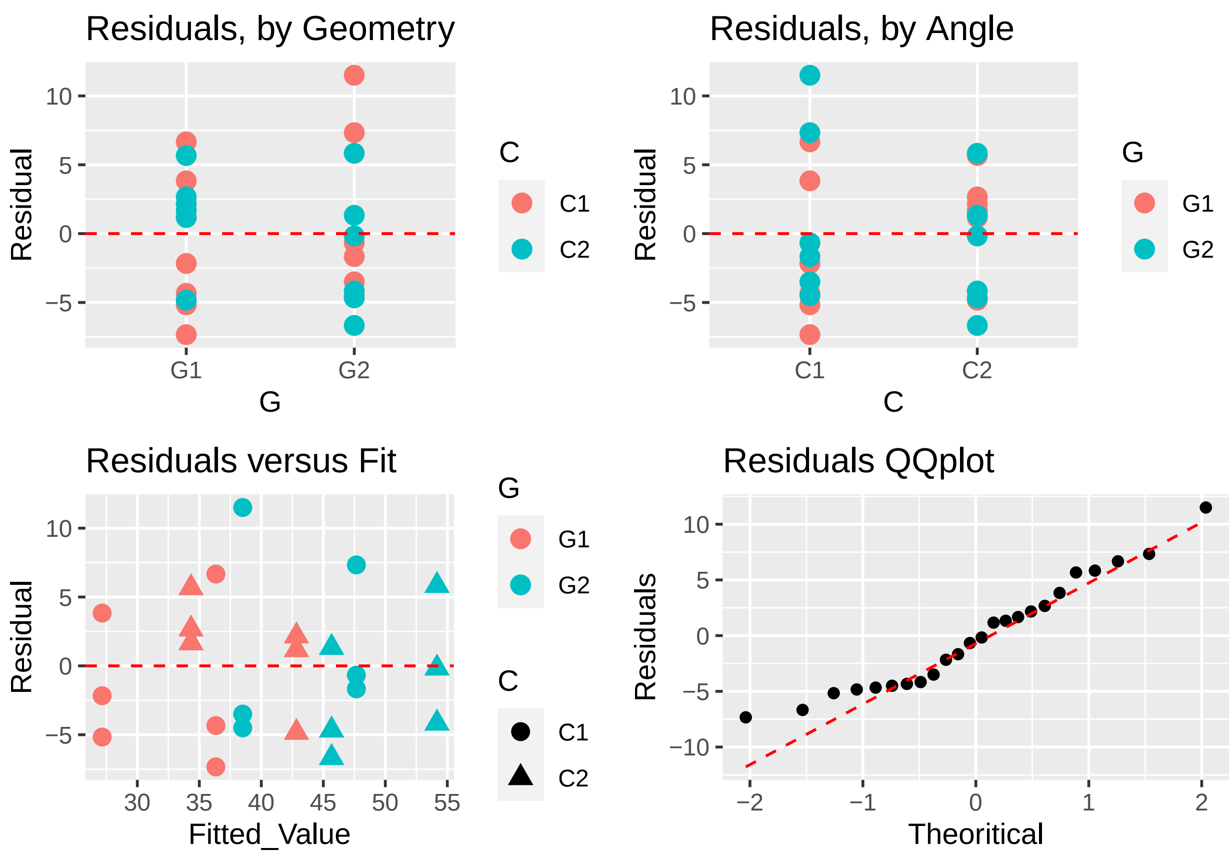
```
print(anova(linear_model))
```

```
## Analysis of Variance Table
##
## Response: LifeHour
##           Df Sum Sq Mean Sq F value    Pr(>F)
## S          1   0.67    0.67   0.022 0.8836408
## G          1 770.67  770.67  25.436 7.216e-05 ***
## C          1 280.17  280.17   9.247 0.0067238 **
## SC         1 468.17  468.17  15.452 0.0008972 ***
## Residuals 19 575.67   30.30
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Only Speed appears to be statistically insignificant.

d)

```
data[, Residual := linear_model$residuals] %>%
    .[, Fitted_Value := linear_model$fitted.values]


plots <- list()
plots$plot1 <- ggplot(data, aes(y = Residual, x = G, color = C)) + geom_point(size = 3) +
                                                        geom_hline(yintercept = 0, color = "red", li
                                                        ggtitle("Residuals, by Geometry")

plots$plot2 <- ggplot(data, aes(y = Residual, x = C, color = G)) + geom_point(size = 3) +
                                                        geom_hline(yintercept = 0, color = "red", li
                                                        ggtitle("Residuals, by Angle")

plots$plot3 <- ggplot(data, aes(y = Residual, x = Fitted_Value, color = G, shape = C)) + geom_point(size
                                                            geom_hline(yintercept = 0, color = "red
                                                            ggtitle("Residuals versus Fit")


plots$plot4 <- ggplot(data, aes(sample = Residual)) + stat_qq() +
                                        stat_qq_line(color = "red", linetype = "dashed") +
                                        ggtitle("Residuals QQplot") +
                                        xlab("Theoritical") +
                                        ylab("Residuals")

gridExtra::grid.arrange(grobs = plots, ncol=2, nrow=2)
```



Residuals do not deviate too much from normality, and their distribution seem to be the same accross groups. Additionnaly, the residuals do no appear to be correlated with the model's output.

e)

Speed doesn't seem to be statistically significant, and it interacts negatively with the cutting angle. Additionnaly, it is the variable whose coefficient is of the least magnitude. Given the interaction term is the only variable with a negative coefficient, it would be wise to set the cutting angle high, as well as the geometry, whilst keeping speed low. This way, the negative impact of a low speed on Life Hour will be overriden by the expected effect of the interaction.

Q7

a)

```
data_path <- "C:/Users/frank/OneDrive/Documents/Assignments/DoE 6.5.csv"
data <- fread(data_path) %>%
            .[, Unit := NULL]

#order the frame
yates_design <- matrix(nrow = 2^3, ncol = 3)
yates_design[1, ] <- c(-1,-1,-1)
yates_design[2, ] <- c(1,-1,-1)
yates_design[3, ] <- c(-1,1,-1)
yates_design[4, ] <- c(1,1,-1)
yates_design[5, ] <- c(-1,-1,1)
yates_design[6, ] <- c(1,-1,1)
yates_design[7, ] <- c(-1,1,1)
yates_design[8, ] <- c(1,1,1)

#Compute the means
g <- function(i){sum(data[S == yates_design[i, 1] & G == yates_design[i, 2] & C == yates_design[i, 3], 
responses <- sapply(c(1:nrow(yates_design)), g)

yates_output <- matrix(0, 2^3, 3)
#Fill columbns
for(j in 1:3){

    for(i in 1:4){

        if(j == 1){

            a <- responses[2*(i)-1]
            b <- responses[2*i]

        } else {

            a <- yates_output[2*(i)-1, j-1]
            b <- yates_output[2*i, j-1]

        }

        yates_output[i, j] <- a + b
        yates_output[i+4, j] <- b - a

    }

}

yates_estimates <- as.matrix(yates_output[, 3] / (3*2^3))
rnames <- rep("", 8)
rnames[1] <- "(Intercept)"
for(i in 2:8){
```

```r
    index <- which(yates_design[i, ] == 1)
    nm <- ""
    for(j in index){nm <- paste(nm, names(data)[j], sep = "")}
    rnames[i] <- nm

}
rownames(yates_estimates) <- rnames
rownames(yates_output) <- rnames
linear_model_coefs <- as.matrix(coef(linear_model_all)[c(1, 2, 3, 5, 4, 6, 7, 8)])
rownames(linear_model_coefs) <- rnames

print("Yate's matrix:", quote = FALSE)
```

```
## [1] Yate's matrix:
```

```r
print(yates_output)
```

```
##             [,1] [,2] [,3]
## (Intercept)  182  449  980
## S            267  531    4
## G            240   55  136
## SG           291  -51  -20
## C             26   85   82
## SC            29   51 -106
## GC           -14    3  -34
## SGC          -37  -23  -26
```

```r
print("--------------------------------------", quote = FALSE)
```

```
## [1] --------------------------------------
```

```r
print("Yate's coefficients:", quote = FALSE)
```

```
## [1] Yate's coefficients:
```

```r
print(yates_estimates)
```

```
##                   [,1]
## (Intercept) 40.8333333
## S            0.1666667
## G            5.6666667
## SG          -0.8333333
## C            3.4166667
## SC          -4.4166667
## GC          -1.4166667
## SGC         -1.0833333
```

```r
print("--------------------------------------", quote = FALSE)
```

```
## [1] --------------------------------------
```

```r
print("Linear model coefficients:", quote = FALSE)
```

```
## [1] Linear model coefficients:
```

```r
print(linear_model_coefs)
```

```
##                   [,1]
## (Intercept)  26.000000
## S             8.666667
## G            13.666667
## SG            1.000000
## C            16.333333
## SC          -13.333333
## GC           -1.333333
## SGC          -8.666667
```

The coefficients aren't the same as those of a linear model using indicator variables as columns.

b)

The contrast would be the expansion of: $(a-1)(b-1)(c-1)(d+1)(e-1)$

Which I don't really feel like expanding since it has 32 terms.

Let C_{abce} denote the appropriatedly weighted sum of orthogonal residuals stemming from the contrast and the estimated effects. (I.e.: ϵ_a, ϵ_b, ... , ϵ_{abcde} weighted by either -1 or 1, based on the contrast).

Then the effect is: $\frac{2}{n2^k} \cdot C_{abce}$, while the associated sum of square is $\frac{1}{n2^k} \cdot C^2_{abce}$, where n denotes the number of replicates.