

# HW5: Factorial Designs

MATH/STAT 571B

**DUE: 4/5/2024 11:59pm**

## Homework Guidelines

*Please submit your answers on Gradescope as a PDF with pages matched to question answers.*

One way to prepare your solutions to this homework is with R Markdown, which provides a way to include mathematical notation, text, code, and figures in a single document. A template .Rmd file is available through D2L.

Make sure all solutions are clearly labeled, and please utilize the question pairing tool on Gradescope. You are encouraged to work together, but your solutions, code, plots, and wording should always be your own. Come and see me during office hours or schedule an appointment when you get stuck and can't get unstuck.

## I. Mathematical Foundations [12 pts]

- (1) [12 pts] The goal of this question is to show that if a factorial design is  $D$ -optimal for the “effects” coding, it is also optimal for the 0-1 coding.

Consider an experiment designed to fit an additive model for two factors, A and B, each with two levels.

- (a) [4 pts] Let  $\mathbf{x}_i^{\text{eff}} = (1 \ x_{iA}^{\text{eff}} \ x_{iB}^{\text{eff}})$  represent the “effects” coding of factor levels for the  $i$ th observation such that “low” levels of each factor are assigned a value of -1 and “high” levels are assigned 1 (e.g., (1 -1 -1) corresponds to low levels of both factors). In addition, let  $\mathbf{x}_i$  represent the 0-1 coding of the same factor combinations but with 0 assigned for “low” levels instead of -1 (e.g., (1, 0, 0) corresponds to low levels of both factors). Find the values of a  $3 \times 3$  matrix  $\mathbf{A}$  such that  $\mathbf{x}_i^{\text{eff}} \mathbf{A} = \mathbf{x}_i$  for all possible  $\mathbf{x}_i^{\text{eff}}$  and  $\mathbf{x}_i$ .

The correct matrix is shown at the bottom of the following R chunk.

```
library(MASS)
library(dplyr)
library(lmerTest)

levels <- factor(c("-", "+"))
df <- expand.grid(A=levels,
                 B=levels,
                 C=levels)
formula <- ~ A + B + C
X_dm <- model.matrix(formula, data = df)
X_eff <- model.matrix(formula, data = df,
                      contrasts.arg = list(A = contr.helmert,
                                           B = contr.helmert,
                                           C = contr.helmert))
```

```
A <- round(ginv(X_eff)%*%X_dm,2)
X_eff%*%A # Correct output
```

```
##      (Intercept) A+ B+ C+
## 1          1  0  0  0
## 2          1  1  0  0
## 3          1  0  1  0
## 4          1  1  1  0
## 5          1  0  0  1
## 6          1  1  0  1
## 7          1  0  1  1
## 8          1  1  1  1
```

A

```
##      (Intercept) A+ B+ C+
## [1,]          1 0.5 0.5 0.5
## [2,]          0 0.5 0.0 0.0
## [3,]          0 0.0 0.5 0.0
## [4,]          0 0.0 0.0 0.5
```

- (b) [4 pts] Let  $\mathbf{X}^{\text{eff}}$  be the design matrix in the “effects” coding in which each row is  $\mathbf{x}_i^{\text{eff}}$  such that  $\mathbf{y} = \mathbf{X}^{\text{eff}}\boldsymbol{\beta}^{\text{eff}} + \boldsymbol{\epsilon}$ ,  $\boldsymbol{\epsilon} \sim N(0, \sigma^2 \mathbf{I})$ . Explain why  $\mathbf{X}^{\text{eff}}\mathbf{A} = \mathbf{X}$ , and then show that if a design minimizes  $|\mathbf{X}^{\text{eff}'}\mathbf{X}^{\text{eff}}|^{-1/2}$  it also minimizes  $|\mathbf{X}'\mathbf{X}|^{-1/2}$  (hint: you can use the following property of determinants:  $|\mathbf{BC}| = |\mathbf{B}||\mathbf{C}|$ ).

$\mathbf{X}^{\text{eff}}\mathbf{A} = \mathbf{X}$  holds because we know that the Helmert Matrix can convert  $\mathbf{X}$  to an orthogonal matrix ( $\mathbf{X}^{\text{eff}}$ ). Then, the reverse transformation would be converting  $\mathbf{X}^{\text{eff}}$  back to  $\mathbf{X}$  which requires the  $\mathbf{A}$  matrix we solve for in part (a). This can generalize for other transformations on the model matrix as well (beyond effects and dummy coding).

Also, you can see that there is a positive relationship between  $|\mathbf{X}'\mathbf{X}|$  and  $|\mathbf{X}_{\text{eff}}'\mathbf{X}_{\text{eff}}|$  from the following. The  $\mathbf{AA}'$  matrix is positive definite, so the determinant must be positive as well. Therefore, minimizing  $|\mathbf{X}_{\text{eff}}'\mathbf{X}_{\text{eff}}|^{-1/2}$  will minimize  $|\mathbf{X}'\mathbf{X}|^{-1/2}$ , too.

minimize  $|\mathbf{X}'\mathbf{X}|^{-1/2} \Leftrightarrow$  maximize  $|\mathbf{X}'\mathbf{X}|$

$$\begin{aligned} |\mathbf{X}'\mathbf{X}| &= |(\mathbf{A}'\mathbf{X}_{\text{eff}}')(\mathbf{X}_{\text{eff}}\mathbf{A})| \\ &= |\mathbf{A}'(\mathbf{X}_{\text{eff}}'\mathbf{X}_{\text{eff}})\mathbf{A}| \\ &= |\mathbf{A}||\mathbf{A}'||\mathbf{X}_{\text{eff}}'\mathbf{X}_{\text{eff}}| \\ &= |\mathbf{AA}'||\mathbf{X}_{\text{eff}}'\mathbf{X}_{\text{eff}}| \end{aligned}$$

Now consider an arbitrary experiment with  $k$  factors, each with its own number of levels, designed for any linear model with identifiable parameters.

- (c) [4 pts] It can be shown that any two design matrices,  $\mathbf{X}^*$ ,  $\mathbf{X}^\dagger$  for the same underlying model based on different codings can be related via some linear transformation,  $\mathbf{T}$ , such that  $\mathbf{X}^*\mathbf{T} = \mathbf{X}^\dagger$ . Explain why this means that  $D$ -optimality does not depend on the choice of coding for any linear model.

Since  $D$ -Optimality is proportional to  $|\mathbf{X}'\mathbf{X}|^{-1/2}$ , and minimizing this value for effects coding and dummy coding yields the same value, it would not matter what the encoding type is. This also generalizes to other types of encoding. As explained in part (b), having some matrix  $\mathbf{A}$  to transform between different types of encoding will produce a positive definite matrix. This means the determinant will be positive. This allows  $|\mathbf{X}'\mathbf{X}|^{-1/2}$  to be proportional to the variance of the coefficient estimates no matter what the encoding type is. Therefore, encoding will not change the  $D$ -optimal design.

## II. Applications [28 pts]

(2) [6 pts] (DAE 6.5)

- (a) [2 pts] At  $\alpha = 0.05$ , the factors Speed, Size, and Speed:Size interaction appear to be significant predictors of vibration. There also appears to be a positive relationship for all 3 factors.

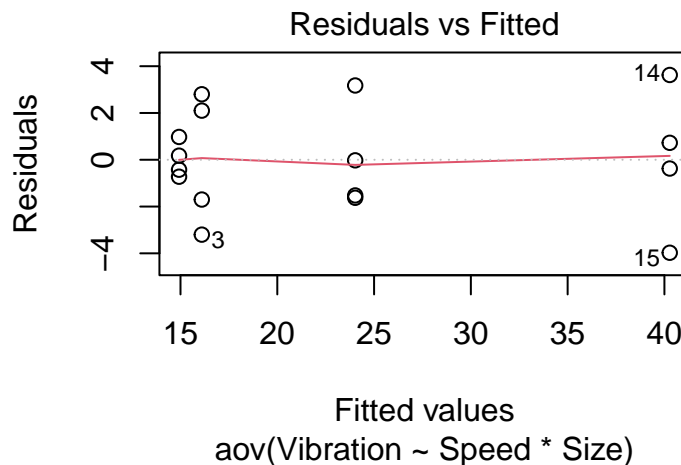
```
vibration <- read.csv("./Datasets/Q6-5.csv")
vibration <- vibration %>%
  mutate_at(vars(-Vibration), as.factor)

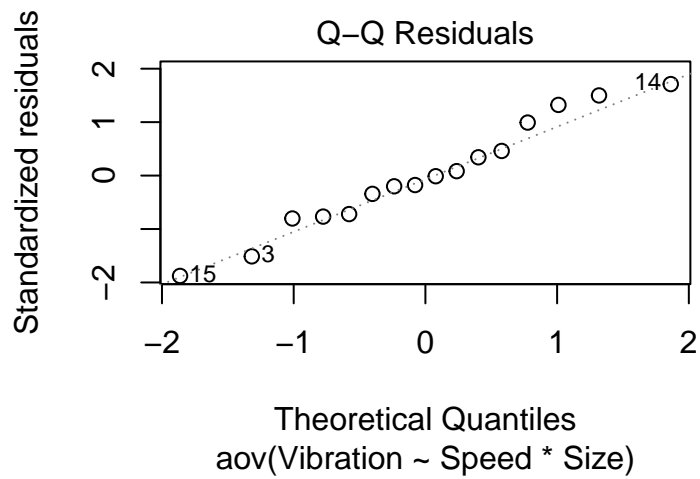
vibration_aov <- aov(Vibration ~ Speed * Size, data = vibration)
summary(vibration_aov)
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Speed      1  227.3    227.3    38.02 4.83e-05 ***
## Size       1 1107.2   1107.2   185.25 1.17e-08 ***
## Speed:Size  1  303.6    303.6    50.80 1.20e-05 ***
## Residuals 12    71.7     6.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

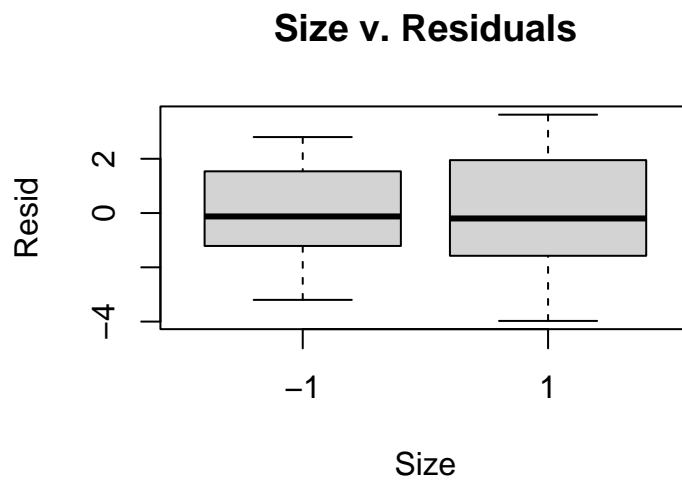
- (b) [2 pts] The qq plot shows that residuals appear to be normally distributed. The residuals v. fitted and residuals v. predictors also seem to have constant variance. There may be some heteroskedasticity in the Speed predictors (the -1 factor in Speed appears to possibly have lower variance in residuals.)

```
plot(vibration_aov, which = c(2,1))
```

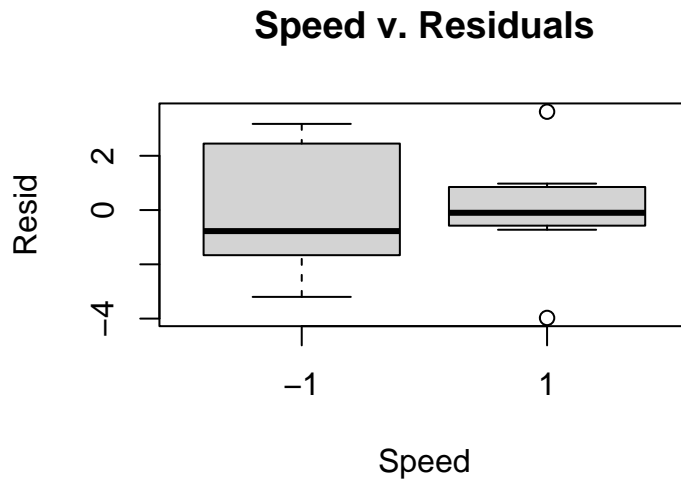




```
plot(vibration$Size, resid(vibration_aov), main = "Size v. Residuals", xlab="Size", ylab="Resid")
```

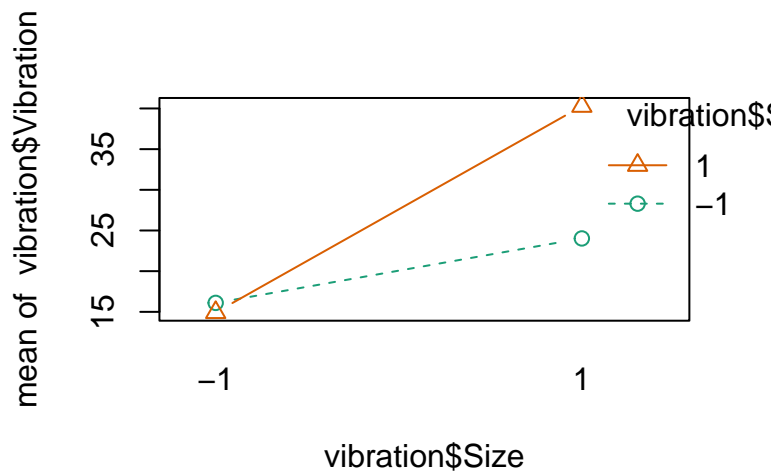


```
plot(vibration$Speed, resid(vibration_aov), main = "Speed v. Residuals", xlab="Speed", ylab="Resid")
```



- (c) [2 pts] The lines have extremely different slopes, indicating a likely interaction. It appears that size and speed have a reinforcing effect, where factor 1 of size and factor 1 of speed have a greater effect on the vibration. To reduce vibration, factor -1 speed and factor -1 size would be best for routine operation.

```
interaction.plot(vibration$Size, vibration$Speed, vibration$Vibration, type = "b",
                pch = 1:6, col = RColorBrewer::brewer.pal(6, "Dark2"))
```



- (3) [8 pts] (DAE 6.12)

- (a) [1 pts] Based on the full dataset, the average factor effects appear to be -0.31725, 0.58600, and 0.28150 for Rate1, Time1, and Rate1:Time1 respectively. Based on part (e), I decided to remove the outliers 2,15 which led to the average factor effects being 0.01920833, 0.92245833, 0.06970833 for Rate1, Time1, and Rate1:Time1 respectively.

```

thickness <- read.csv("./Datasets/Q6-12.csv")
thickness <- thickness %>%
  mutate_at(vars(-Thickness), as.factor)

# full data
thickness_aov <- aov(Thickness ~ Rate * Time, data = thickness,
  contrasts = list(Rate = "contr.helmert",
    Time = "contr.helmert"))
coef(thickness_aov) * 2

```

```

## (Intercept)      Rate1      Time1 Rate1:Time1
##      29.02775      -0.31725      0.58600      0.28150

```

```

# removing outlier
thickness_aov_rm <- aov(Thickness ~ Rate * Time, data = thickness[c(-2, -15),],
  contrasts = list(Rate = "contr.helmert",
    Time = "contr.helmert"))
coef(thickness_aov_rm) * 2

```

```

## (Intercept)      Rate1      Time1 Rate1:Time1
## 28.81595833 0.01920833 0.92245833 0.06970833

```

- (b) [2 pts] There are no significant factors when including outliers. However, when removing outliers, the additive factors seem to be significant (Rate and Time) at an  $\alpha = 0.05$ .

```
summary(thickness_aov)
```

```

##           Df Sum Sq Mean Sq F value Pr(>F)
## Rate       1  0.403   0.4026   1.262 0.2833
## Time       1  1.374   1.3736   4.305 0.0602 .
## Rate:Time   1  0.317   0.3170   0.994 0.3386
## Residuals  12  3.828   0.3190
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
summary(thickness_aov_rm)
```

```

##           Df Sum Sq Mean Sq F value    Pr(>F)
## Rate       1 0.0444   0.0444  12.657 0.0052 **
## Time       1 2.9175   2.9175 832.554 5.82e-11 ***
## Rate:Time   1 0.0167   0.0167   4.754 0.0542 .
## Residuals  10 0.0350   0.0035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

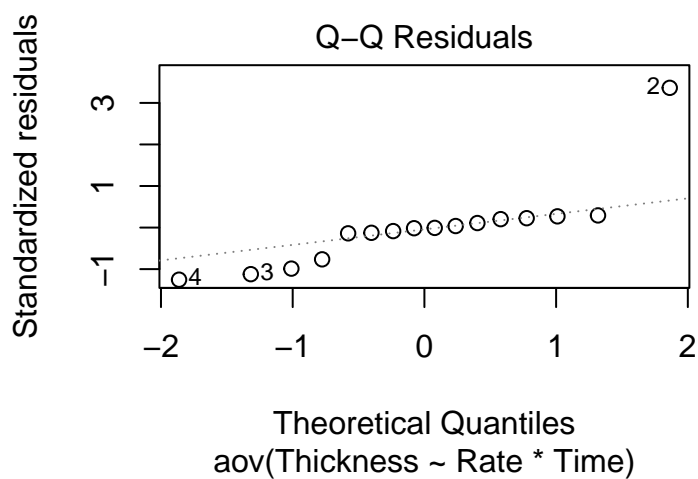
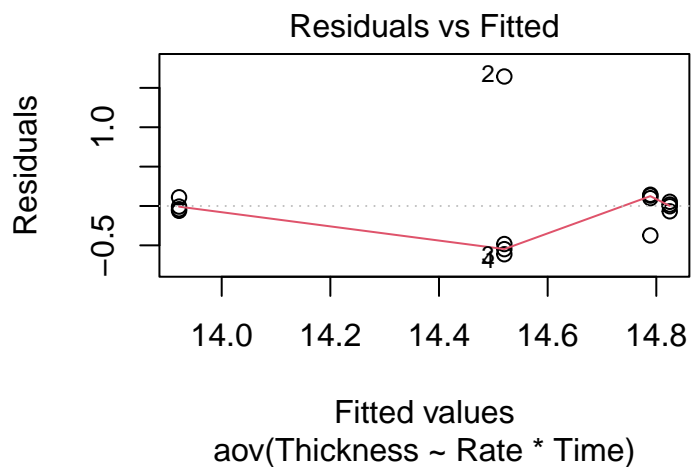
- (c) [1 pts] There were no significant factors using the data containing outliers. However, the regression line from the data with outliers removed is:  $\hat{y}_i = 14.407979167 + 0.009604167 * x_{iA} + 0.461229167 * x_{iB}$  using -1 1 encoding.

```
coef(thickness_aov_rm)
```

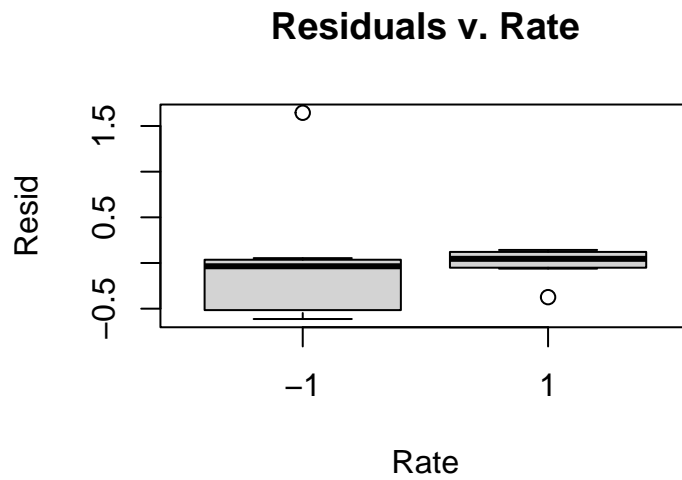
```
## (Intercept)      Rate1      Time1 Rate1:Time1
## 14.407979167  0.009604167  0.461229167  0.034854167
```

- (d) [2 pts] The QQ plot, Residuals v. fitted, and Residuals v. predictor plots show that point 2 is an outlier. The residuals v. predicted plots show that point 15 under the factor 1 conditions show point 15 could be a potential outlier, as well. After removing these outliers, the QQ plot appears normal, and the residual plots show constant variance.

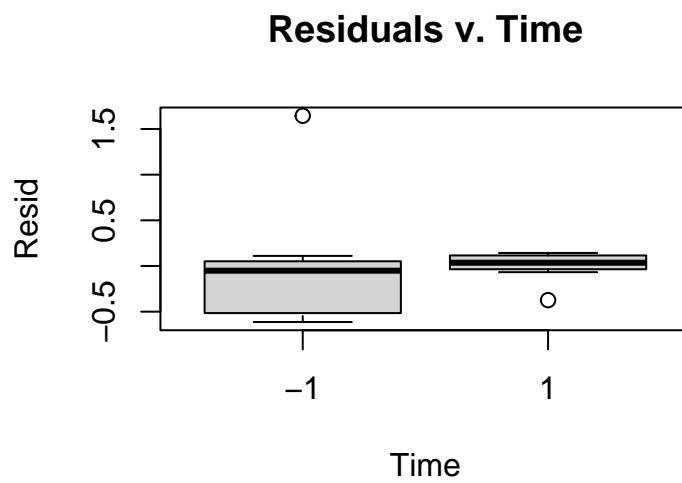
```
plot(thickness_aov, which = c(1,2))
```



```
plot(thickness$Rate, resid(thickness_aov), main = "Residuals v. Rate", xlab = "Rate", ylab = "Resid")
```

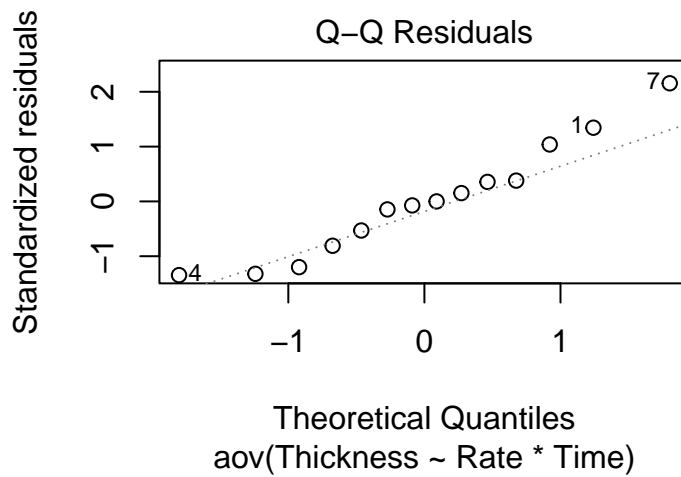
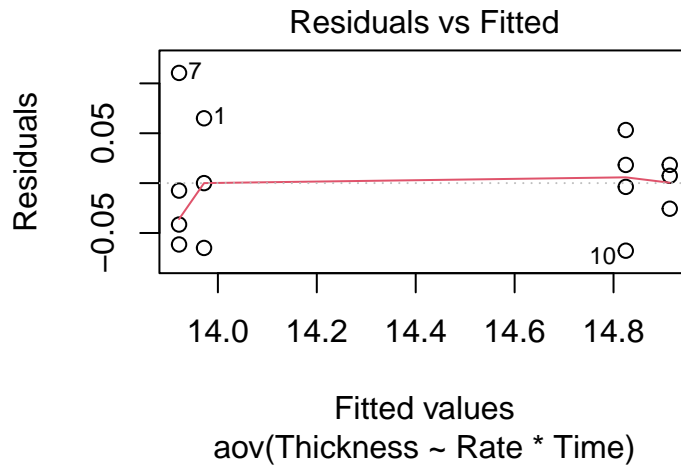


```
plot(thickness$Time, resid(thickness_aov), main = "Residuals v. Time", xlab = "Time", ylab = "Resid")
```



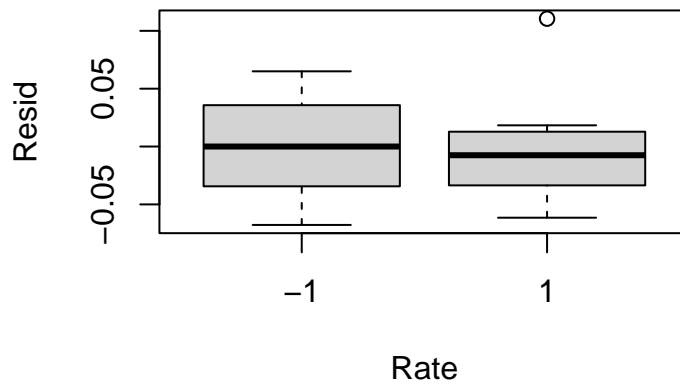
```
plot(thickness_aov_rm, which = c(1,2))
```





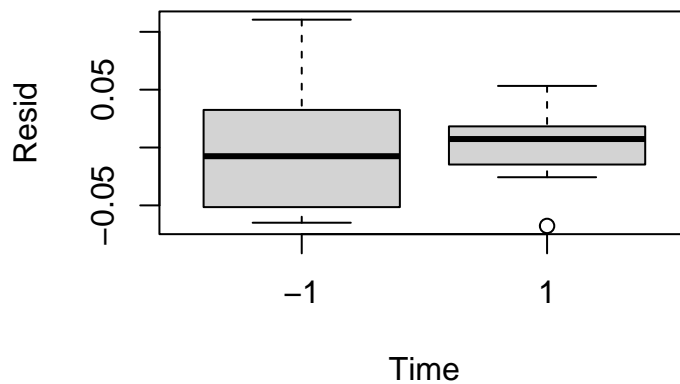
```
plot(thickness$Rate[c(-2,-15)], resid(thickness_aov_rm), main = "Residuals v. Rate (Outlier Removed)",
```

### Residuals v. Rate (Outlier Removed)



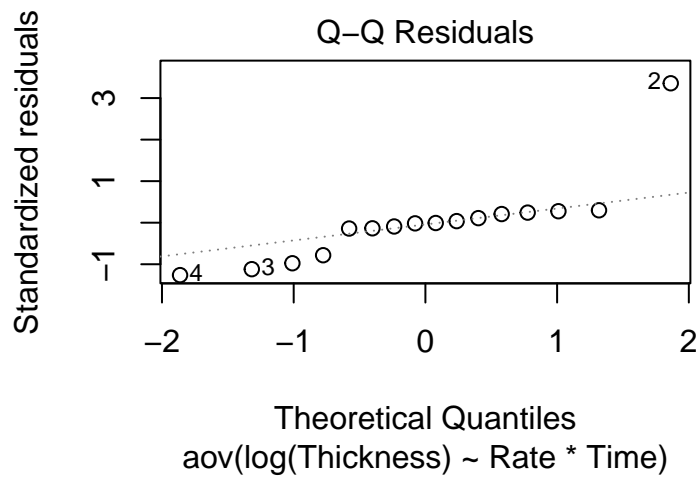
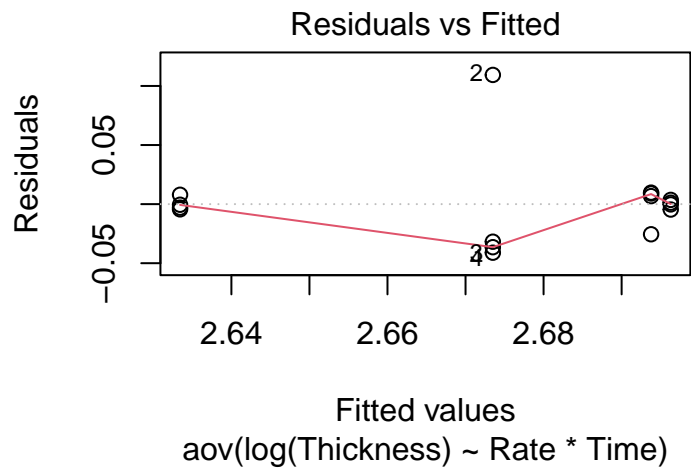
```
plot(thickness$Time[c(-2,-15)], resid(thickness_aov_rm), main = "Residuals v. Time (Outlier Removed)",
```

### Residuals v. Time (Outlier Removed)

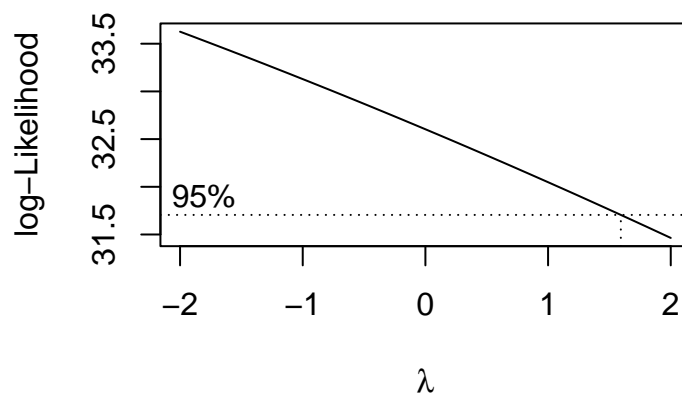


- (e) [2 pts] As seen in parts (a)-(d), I have found the 2nd and 15th data points to be outliers. But, because of the symmetry of the outliers (2nd data point and 2nd to last data point), there is the possibility that this is a systematic error. Therefore, I decided to conduct the analysis without them. Other possibilities that were explored were the log transform and the box cox transformation on Y to address the issues of the residuals. However, they were shown not to have much of an effect on fixing these outliers.

```
# log
plot(aov(log(Thickness) ~ Rate*Time, data = thickness), 1:2)
```



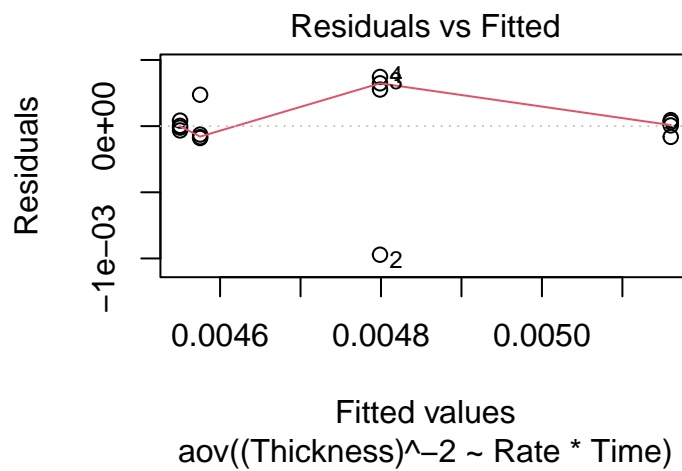
```
# box-cox
bc <- boxcox(thickness_aov)
```

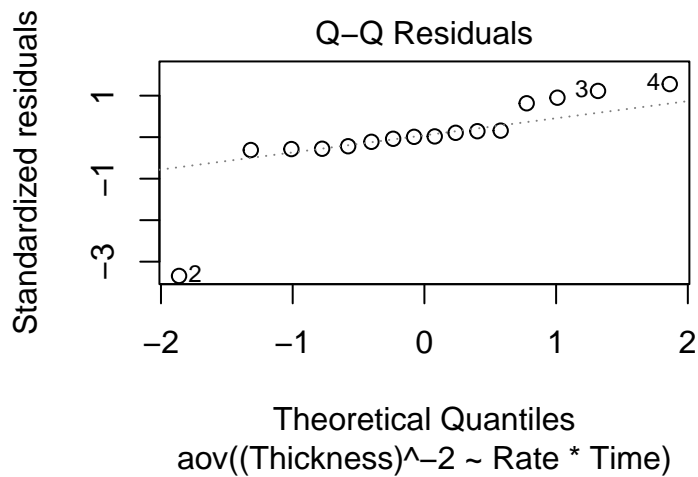


```
bc$x[which.max(bc$y)]
```

```
## [1] -2
```

```
plot(aov((Thickness)^-2 ~ Rate*Time, data = thickness), 1:2)
```





(4) [8 pts] (DAE 6.28)

(a) [1 pts] B seems to have the least effect out of the 4 factors according to the normal probability plot. A, C, D appear to have the largest effects. AC also appears to have a large effect.

```
yield <- read.csv("./Datasets/Q6-28.csv")
yield <- yield %>%
  mutate_at(vars(-Yield), as.factor)

yield_aov <- aov(Yield ~ A*B*C*D, data = yield,
  contrasts = list(A="contr.helmert",
    B="contr.helmert",
    C="contr.helmert",
    D="contr.helmert"))

anova(yield_aov)
```

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

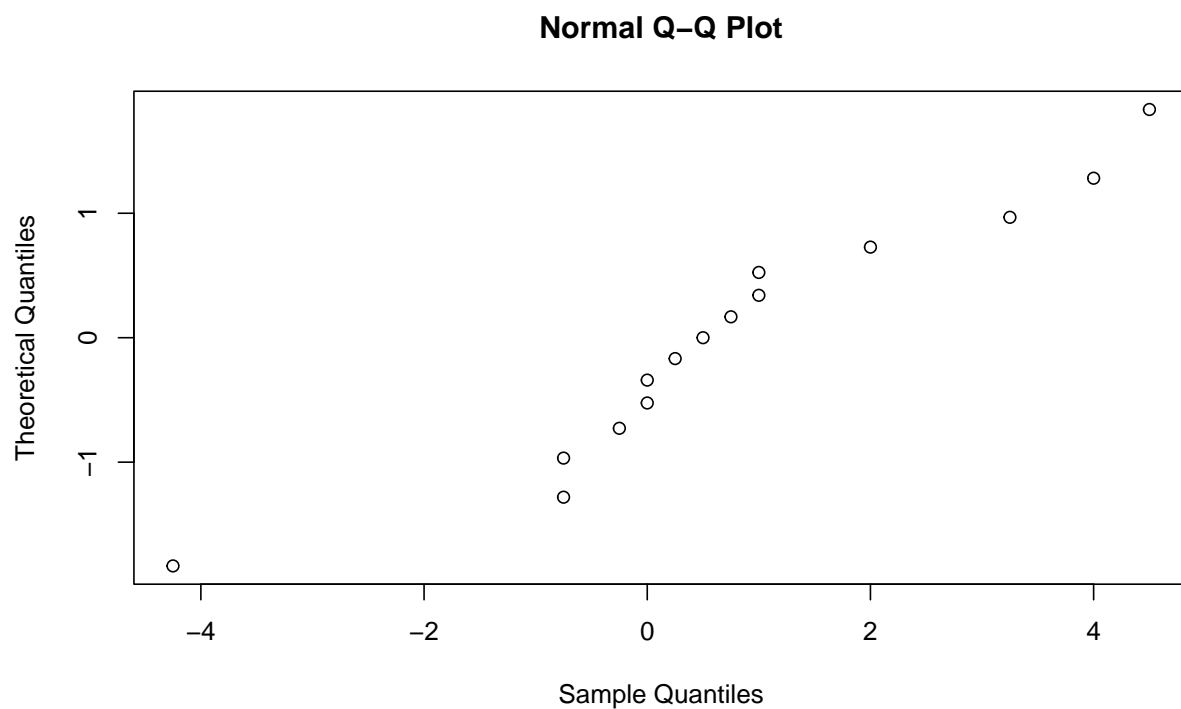
```
## Analysis of Variance Table
##
## Response: Yield
##
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	1	81.00	81.00	NaN	NaN
B	1	1.00	1.00	NaN	NaN
C	1	16.00	16.00	NaN	NaN
D	1	42.25	42.25	NaN	NaN
A:B	1	2.25	2.25	NaN	NaN
A:C	1	72.25	72.25	NaN	NaN
B:C	1	0.25	0.25	NaN	NaN
A:D	1	64.00	64.00	NaN	NaN
B:D	1	0.00	0.00	NaN	NaN

```
## C:D      1  0.00  0.00   NaN   NaN
## A:B:C     1  4.00  4.00   NaN   NaN
## A:B:D     1  2.25  2.25   NaN   NaN
## A:C:D     1  0.25  0.25   NaN   NaN
## B:C:D     1  2.25  2.25   NaN   NaN
## A:B:C:D    1  4.00  4.00   NaN   NaN
## Residuals 0  0.00   NaN
```

```
coefs <- coef(yield_aov)

# QQ plots of the coefficients.
qqnorm(coefs[-1] * 2, datax = T)
```



```
sort(coefs[-1]) # B seems to have the least effect
```

```
##          A1:C1      B1:C1:D1      A1:B1      A1:C1:D1      B1:D1
## -2.125000e+00 -3.750000e-01 -3.750000e-01 -1.250000e-01 -6.595890e-16
##          C1:D1      B1:C1      B1      A1:B1:D1      A1:B1:C1
## -2.984437e-16  1.250000e-01  2.500000e-01  3.750000e-01  5.000000e-01
##   A1:B1:C1:D1      C1      D1      A1:D1      A1
##  5.000000e-01  1.000000e+00  1.625000e+00  2.000000e+00  2.250000e+00
```

- (b) [2 pts] Performing the analysis on factors A,C, and D (removing factor B), I found tht the main effects A,C,D, as well as pairwise interactions AC,AD appeared significant at  $\alpha=0.05$ .

```
yield_aov_ACD <- aov(Yield ~ A*C*D, data = yield,
                     contrasts = list(A="contr.helmert",
                                     C="contr.helmert",
                                     D="contr.helmert"))
anova(yield_aov_ACD)
```

```
## Analysis of Variance Table
##
## Response: Yield
##          Df Sum Sq Mean Sq F value    Pr(>F)
## A           1  81.00   81.00  40.500 0.0002173 ***
## C           1  16.00   16.00   8.000 0.0222039 *
## D           1  42.25   42.25  21.125 0.0017642 **
## A:C          1  72.25   72.25  36.125 0.0003197 ***
## A:D          1  64.00   64.00  32.000 0.0004776 ***
## C:D          1   0.00    0.00   0.000 1.0000000
## A:C:D        1   0.25    0.25   0.125 0.7328099
## Residuals    8  16.00    2.00
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

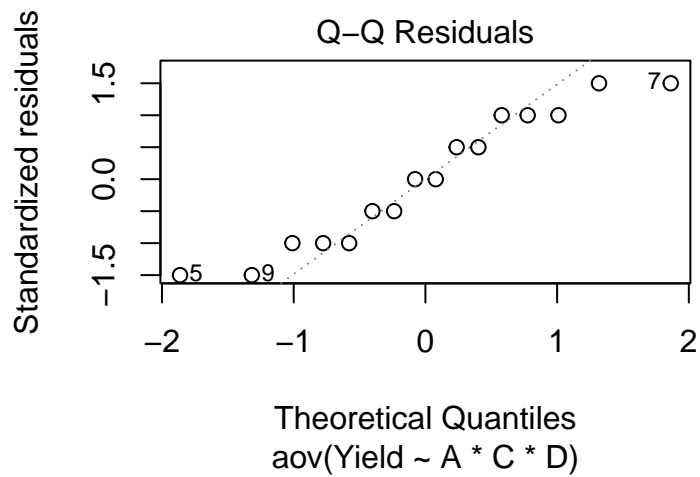
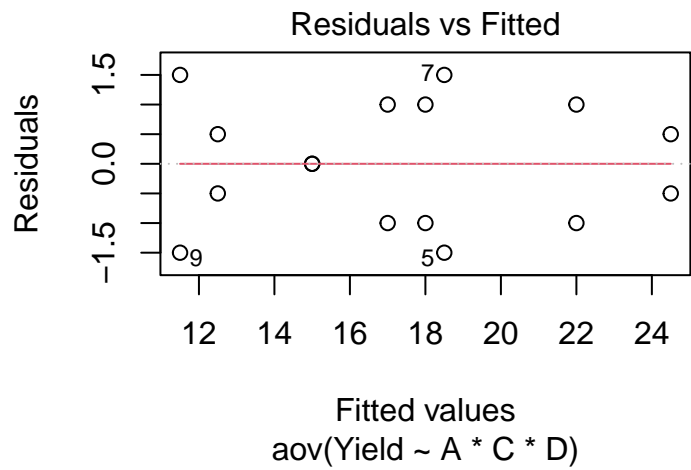
(c) [2 pts] The regression model is:  $\hat{y}_i = 17.375 + 2.25 * x_{iA} + 1 * x_{iC} + 1.625 * x_{iD} + (-2.125) * x_{iA} * x_{iC} + 2 * x_{iA} * x_{iD}$  using -1 1 encoding.

```
coef(yield_aov_ACD)
```

```
##      (Intercept)           A1           C1           D1           A1:C1
## 1.737500e+01  2.250000e+00  1.000000e+00  1.625000e+00 -2.125000e+00
##           A1:D1           C1:D1           A1:C1:D1
## 2.000000e+00  1.552447e-16 -1.250000e-01
```

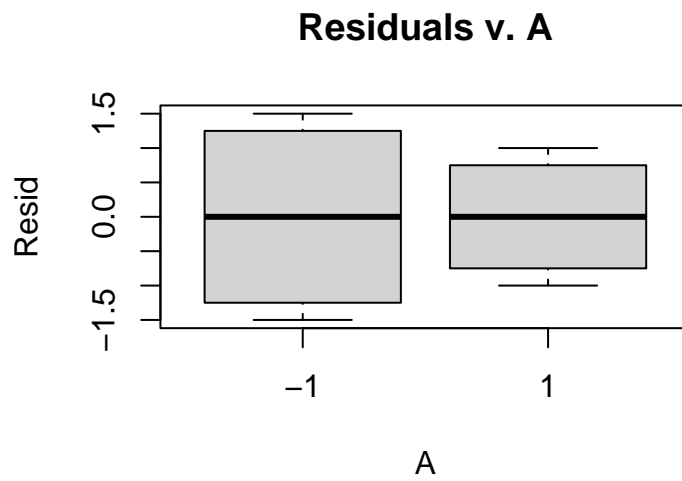
(d) [1 pts] The QQ plot shows normal residuals. The residuals v. fitted and residuals v. predictors all show constant variance. There do not appear to be any problems in the diagnostics.

```
plot(yield_aov_ACD, 1:2)
```

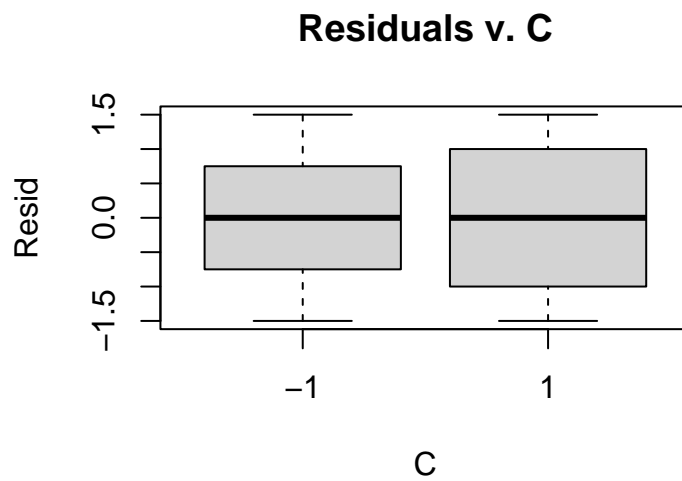


```
plot(yield$A, resid(yield_aov_ACD), main = "Residuals v. A", xlab="A", ylab="Resid")
```

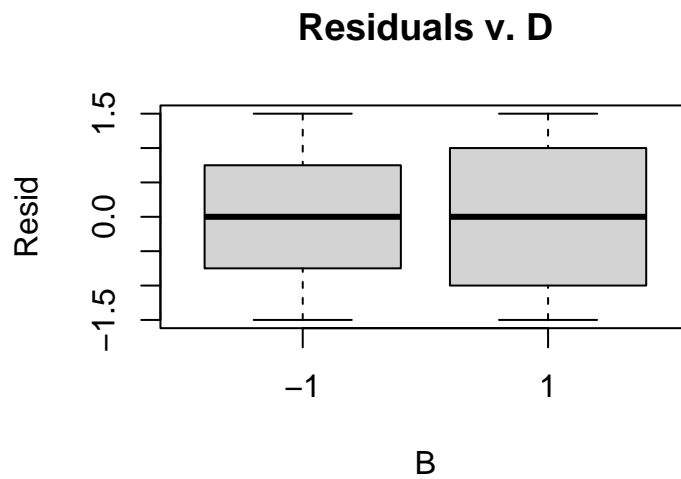




```
plot(yield$C, resid(yield_aov_ACD), main = "Residuals v. C", xlab="C", ylab="Resid")
```



```
plot(yield$D, resid(yield_aov_ACD), main = "Residuals v. D", xlab="B", ylab="Resid")
```



(e) [2 pts] Yes, this design can be collapsed by removing factor B. The figure below shows the average and range of yields for each factor combinations.

```
perform_mean_range <- function(x) return(c(mean(x), diff(x)))
perform_mean_range(yield[yield$A==1 & yield$B==1 & yield$C==1,"Yield"])
```

```
## [1] 19 8
```

```
perform_mean_range(yield[yield$A==1 & yield$B==1 & yield$C==1,"Yield"])
```

```
## [1] 20 8
```

```
perform_mean_range(yield[yield$A==1 & yield$B==1 & yield$C==1,"Yield"])
```

```
## [1] 18 6
```

```
perform_mean_range(yield[yield$A==1 & yield$B==1 & yield$C==1,"Yield"])
```

```
## [1] 21.5 7.0
```

```
perform_mean_range(yield[yield$A==1 & yield$B==1 & yield$C==1,"Yield"])
```

```
## [1] 18.5 -3.0
```

```
perform_mean_range(yield[yield$A==1 & yield$B==1 & yield$C==1,"Yield"])
```

```
## [1] 13 0
```

```
perform_mean_range(yield[yield$A==1 & yield$B==1 & yield$C==1,"Yield"])
```

```
## [1] 18 2
```

```
perform_mean_range(yield[yield$A==1 & yield$B==1 & yield$C==1,"Yield"])
```

```
## [1] 11 -2
```

(5) [6 pts] (DAE 6.30)

(a) [2 pts] The results show that A1 (glass) has a significant and positive effect on the scoring of the cake. The other factors do not appear to be significant.

```
brownies <- read.csv("./Datasets/brownies.csv")
brownies <- brownies %>%
  mutate_at(vars(-c(X,score)), as.factor)

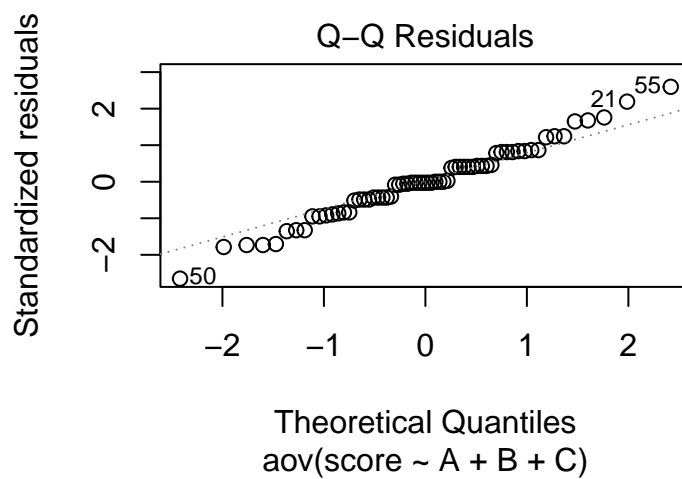
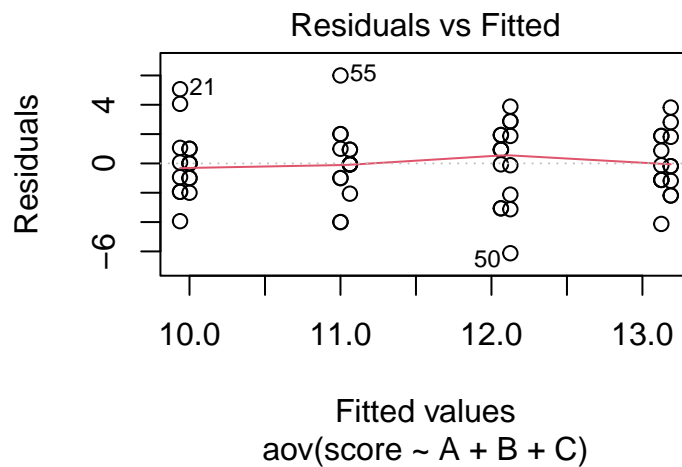
brownies_aov <- aov(score~ A + B + C, data = brownies)
summary(brownies_aov)
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## A           1   72.3    72.25  12.699 0.000725 ***
## B           1   18.1    18.06   3.175 0.079849 .
## C           1    0.1     0.06   0.011 0.916877
## Residuals   60  341.4     5.69
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

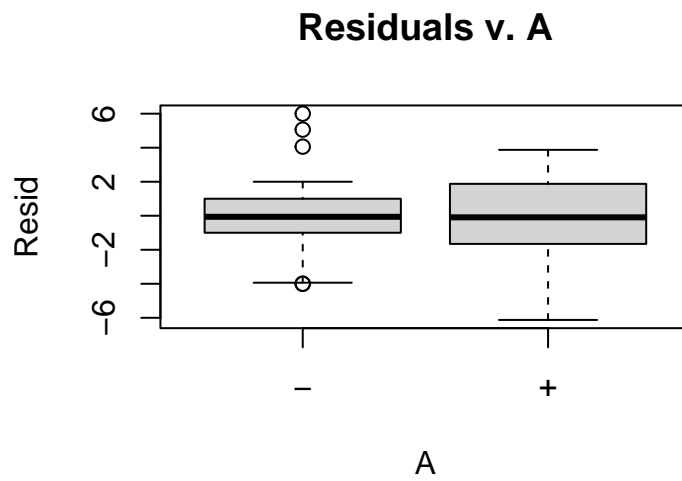
```
coef(brownies_aov)
```

```
## (Intercept)      A+      B+      C+  
##    10.0000     2.1250     1.0625    -0.0625
```

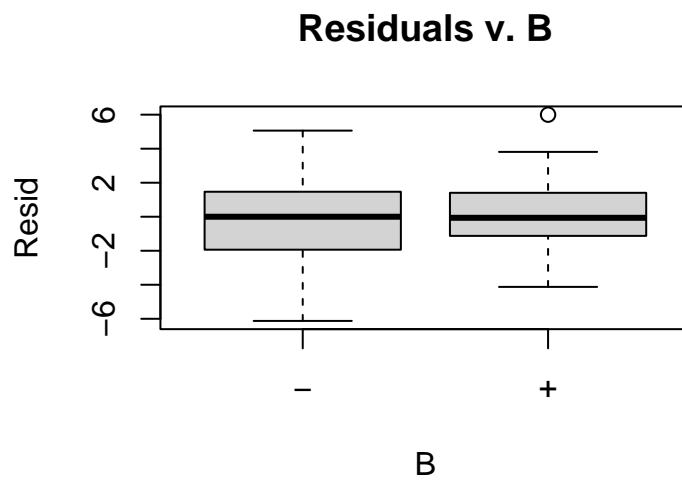
```
plot(brownies_aov, 1:2)
```



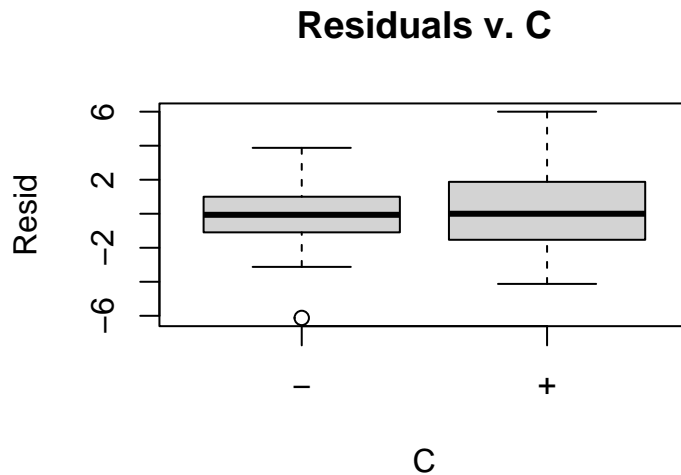
```
plot(brownies$A, resid(brownies_aov), main = "Residuals v. A", xlab="A", ylab="Resid")
```



```
plot(brownies$B, resid(brownies_aov), main = "Residuals v. B", xlab="B", ylab="Resid")
```



```
plot(brownies$C, resid(brownies_aov), main = "Residuals v. C", xlab="C", ylab="Resid")
```



- (b) [2 pts] The analysis in part (a) would not be the correct approach because there is only one batch per factor combination. There are 8 people trying each batch, so there are no true replicates of each factor combination. The test panel should really be treated as a random effect and nuisance factor in the design.
- (c) [2 pts] The average and standard deviation analyses show that none of the factors are significant. This is a more appropriate approach because this treats each batch as having one replicate with some average rating and some standard deviation. The standard deviation analysis can give some evidence that the random effect of Panel\_ID may not be significant if it were to be included. This means that while the analysis with mean/standard deviation may be more appropriate, the analysis in part (a) may be a better test as a follow-up. It has more power and could possibly be treated as replicates since there is not a significant difference in the panel's scoring.

```
head(brownies)
```

```
##   X Panel_ID A B C score
## 1 1         1 - - -   11
## 2 2         1 + - -   15
## 3 3         1 - + -    9
## 4 4         1 + + -   16
## 5 5         1 - - +   10
## 6 6         1 + - +   12
```

```
brownies_summary <- brownies[1:8, c("A", "B", "C")]
brownies_summary$mean <- tapply(brownies$score, brownies$Panel_ID, mean)
brownies_summary$sd <- tapply(brownies$score, brownies$Panel_ID, sd)

brownies_mean_aov <- aov(mean ~ A + B + C, data = brownies_summary)
brownies_sd_aov <- aov(sd ~ A + B + C, data = brownies_summary)
summary(brownies_mean_aov)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## A           1  0.070   0.070   0.072 0.8023
## B           1  1.125   1.125   1.145 0.3448
## C           1  4.500   4.500   4.581 0.0991 .
```

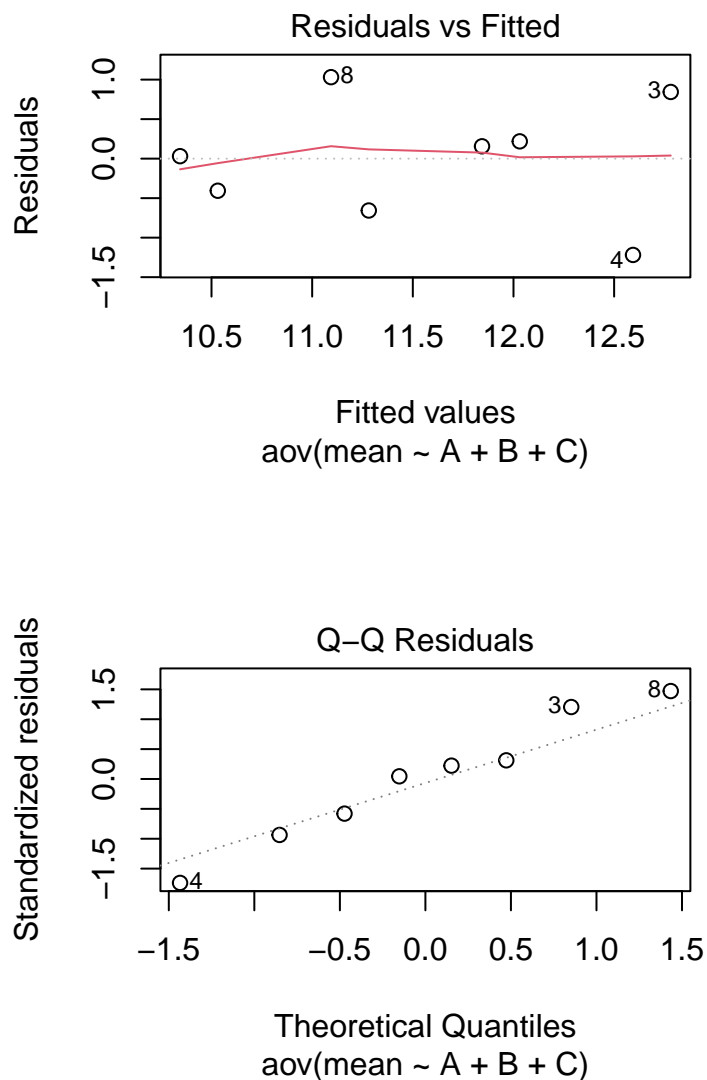
```
## Residuals    4  3.930    0.982
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(brownies_sd_aov)
```

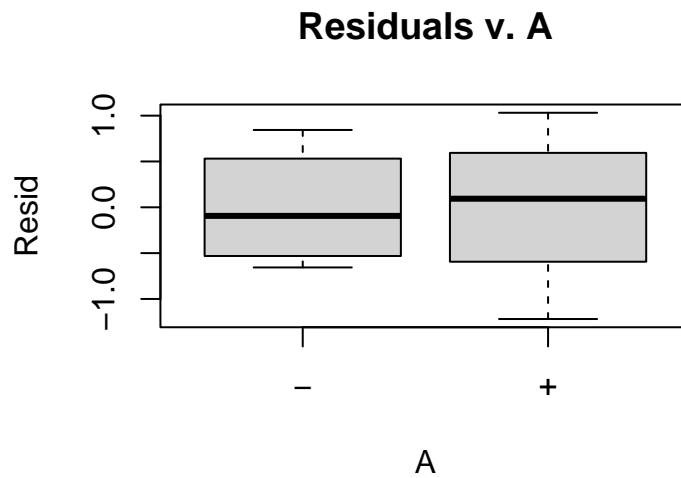
```
##           Df Sum Sq Mean Sq F value Pr(>F)
## A           1  0.4684   0.4684   2.216  0.211
## B           1  0.0001   0.0001   0.000  0.984
## C           1  0.2316   0.2316   1.096  0.354
## Residuals    4  0.8456   0.2114
```

Diagnostic plots shown below reveal normal residuals and constant variance.

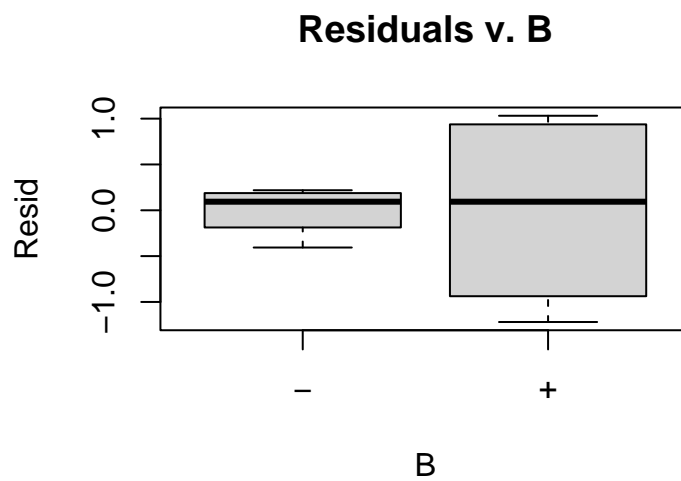
```
plot(brownies_mean_aov, 1:2)
```



```
plot(brownies_summary$A, resid(brownies_mean_aov), main = "Residuals v. A", xlab="A", ylab="Resid")
```

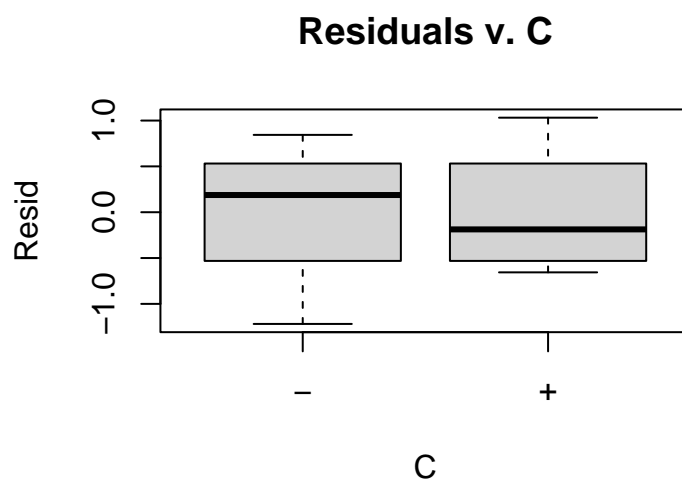


```
plot(brownies_summary$B, resid(brownies_mean_aov), main = "Residuals v. B", xlab="B", ylab="Resid")
```

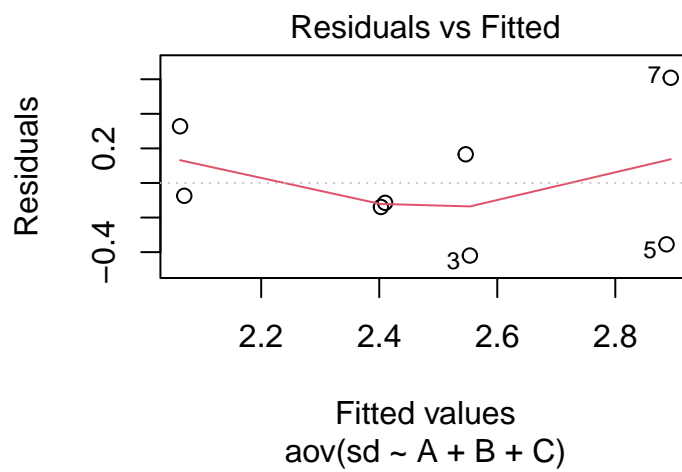


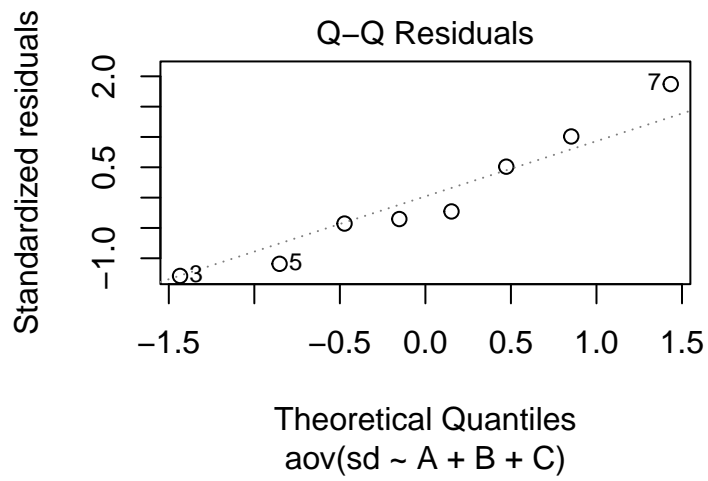
```
plot(brownies_summary$C, resid(brownies_mean_aov), main = "Residuals v. C", xlab="C", ylab="Resid")
```



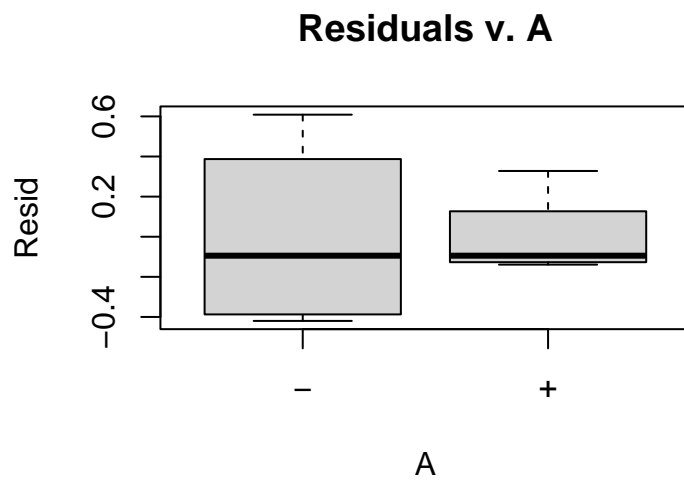


```
plot(brownies_sd_aov, 1:2)
```

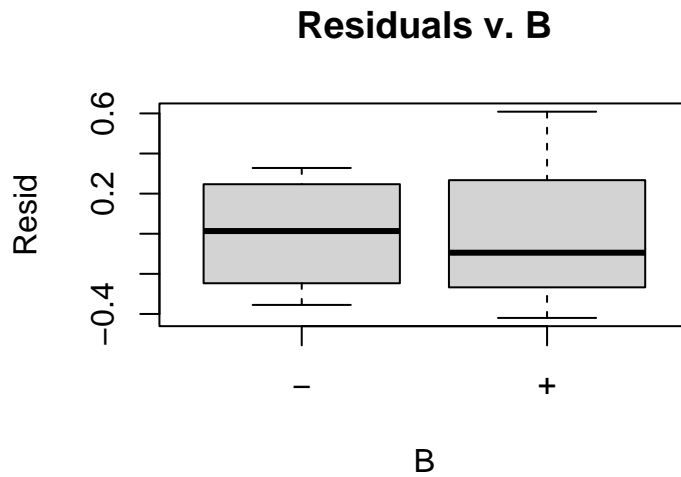




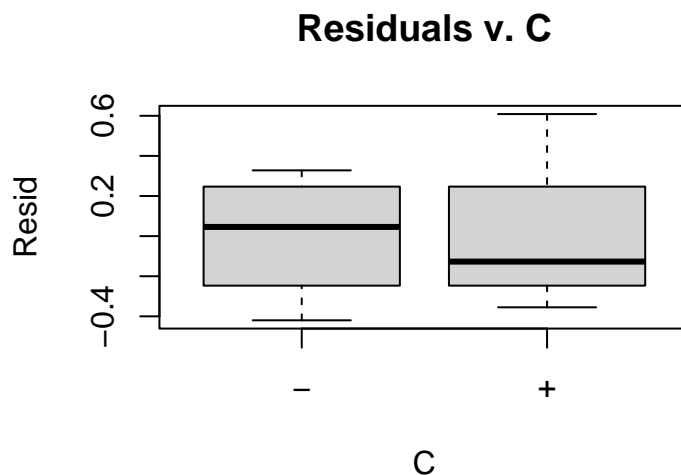
```
plot(brownies_summary$A, resid(brownies_sd_aov), main = "Residuals v. A", xlab="A", ylab="Resid")
```



```
plot(brownies_summary$B, resid(brownies_sd_aov), main = "Residuals v. B", xlab="B", ylab="Resid")
```



```
plot(brownies_summary$C, resid(brownies_sd_aov), main = "Residuals v. C", xlab="C", ylab="Resid")
```



- (d) [BONUS 2 pts] *Fit a model for the raw data treating Panel ID as an additive random effect. How do your conclusions compare? Which of the approaches in (c) and (d) do you think it most appropriate? Why?* By introducing a random effect, factor A is shown to have a significant effect on scoring. Factors B,C, and Panel\_ID do not appear to be significant. The random effect model is the most accurate out of all 3 methods since in part (a) the batches are not true replicates. Each batch is only tested by one member of the panel. The method used in part (c) loses a lot of power because the original data is reduced to a mean and standard deviation value for each batch. Therefore, the random effect model stays true to the data and while retaining the highest level of power.

```
brownies_random_aov <- lmer(score ~ A + B + C + (1 | Panel_ID), data = brownies)
anova(brownies_random_aov)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
```

```
##      Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## A 72.250  72.250      1    53 14.4842 0.0003682 ***
## B 18.062  18.062      1    53  3.6210 0.0624914 .
## C  0.062   0.062      1    53  0.0125 0.9112974
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ranova(brownies_random_aov)
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## score ~ A + B + C + (1 | Panel_ID)
##           npar logLik   AIC   LRT Df Pr(>Chisq)
## <none>         6 -142.35 296.71
## (1 | Panel_ID)   5 -143.53 297.07 2.3579 1    0.1247
```

```
coef(brownies_random_aov)
```

```
## $Panel_ID
##      (Intercept)      A+      B+      C+
## 1  10.375737 2.125 1.0625 -0.0625
## 2  10.239105 2.125 1.0625 -0.0625
## 3  11.127211 2.125 1.0625 -0.0625
## 4   9.897526 2.125 1.0625 -0.0625
## 5   9.214368 2.125 1.0625 -0.0625
## 6   9.351000 2.125 1.0625 -0.0625
## 7   9.487631 2.125 1.0625 -0.0625
## 8  10.307421 2.125 1.0625 -0.0625
##
## attr(,"class")
## [1] "coef.mer"
```