Stat453_Assigment02

Belina Jang

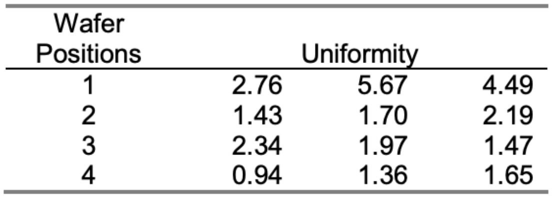
2023-03-25

Questions

Question 1. Show that for the one-way ANOVA we have E(MSE) = sigma^2.

Calculation by hand:

Question 2. An article in the Journal of the Electrochemical Society (Vol. 139, No. 2, 1992, pp. 524-532) describes an experiment to investigate low-pressure vapor deposition of polysilicon. The experiment was carried out in a large capacity reactor at Sematech in Austin, Texas. The reactor has several wafer positions, and four of these positions are selected at random. The response variable is film thickness uniformity. Three replicates of the experiment were run, and the data are as follows:



Wafer Positions	Uniformity		
1	2.76	5.67	4.49
2	1.43	1.70	2.19
3	2.34	1.97	1.47
4	0.94	1.36	1.65

```r
uniformity = c(2.76, 5.67, 4.49, 1.43, 1.70, 2.19, 2.34, 1.97, 1.47, 0.94, 1.36, 1.65)
WaterPosition = c(1,1,1,2,2,2,3,3,3,4,4,4)
Q2data = data.frame(uniformity, WaterPosition)
```

(a) Is there a difference in the wafer positions? Use a=0.05.

```r
group_by(Q2data, WaterPosition) %>%
    summarise(
        count = n(),
        mean = mean(uniformity, na.rm = TRUE),
        sd = sd(uniformity, na.rm = TRUE),
    )
```

```
## # A tibble: 4 x 4
##   WaterPosition count  mean    sd
##           <dbl> <int> <dbl> <dbl>
## 1             1     3  4.31 1.46
## 2             2     3  1.77 0.385
## 3             3     3  1.93 0.437
## 4             4     3  1.32 0.357
```

```
# is there a difference in the wafer positions?
# use alpha = 0.05
# H0: mu1 = mu2 = mu3 = mu4
# H1: at least one of the means is different

res.aov = aov(uniformity ~ factor(WaterPosition), data = Q2data)
summary(res.aov)
```

```
##                       Df Sum Sq Mean Sq F value  Pr(>F)
## factor(WaterPosition)  3 16.220   5.407    8.29 0.00775 **
## Residuals              8  5.217   0.652
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
F0 = 5.407/0.652 #F value: 8.292945
pf(F0, 3, 8, lower.tail = FALSE) # p value:0.00774
```

```
## [1] 0.007738504
```

∴ The p-value for the water position (0.00774) is less than 0.05, so we reject the null hypothesis. There is a difference in the wafer positions.

(b) Estimate the variability due to wafer position (sigma tau hat squared).

```
# estimate the variability due to wafer position
# estimating the parameters in the random model
a = 4 # number of levels
n = 3 # number of replicates
MStreat = 5.407 # sum of squares(treatment)/df(treatment)
MSerror = 0.652 # sum of squares(error)/df(error)
sigma_tau_hat_squared = (MStreat - MSerror)/n # ANSWER: 1.585
```

∴ 1.585

(c) Estimate the random error component (sigma hat squared).

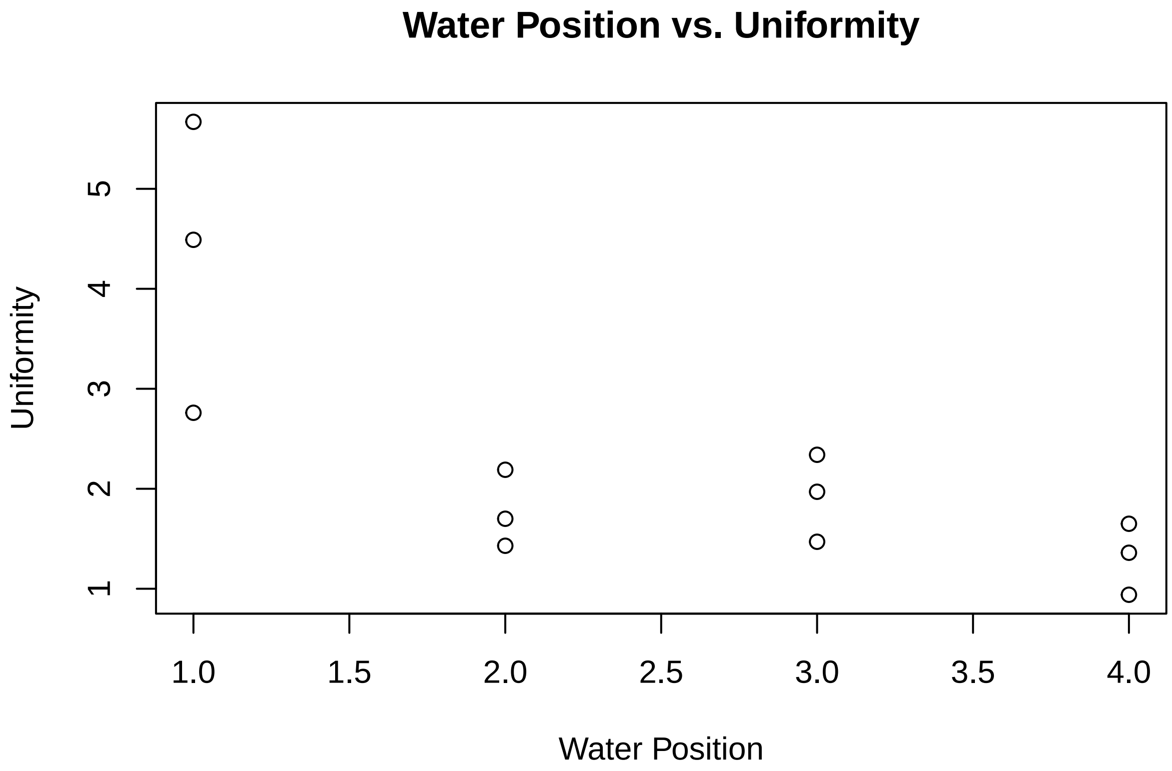
```
sigma_hat_squared = MSerror # ANSWER: 0.652

#double checking
remres = lmer(uniformity ~ 1|factor(WaterPosition))
summary(remres)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: uniformity ~ 1 | factor(WaterPosition)
##
## REML criterion at convergence: 35.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.62006 -0.53578 -0.05227  0.43737  1.98330
##
## Random effects:
##  Groups                Name        Variance Std.Dev.
##  factor(WaterPosition) (Intercept) 1.5848   1.2589
##  Residual                          0.6522   0.8076
## Number of obs: 12, groups:  factor(WaterPosition), 4
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   2.3308     0.6712   3.472
```

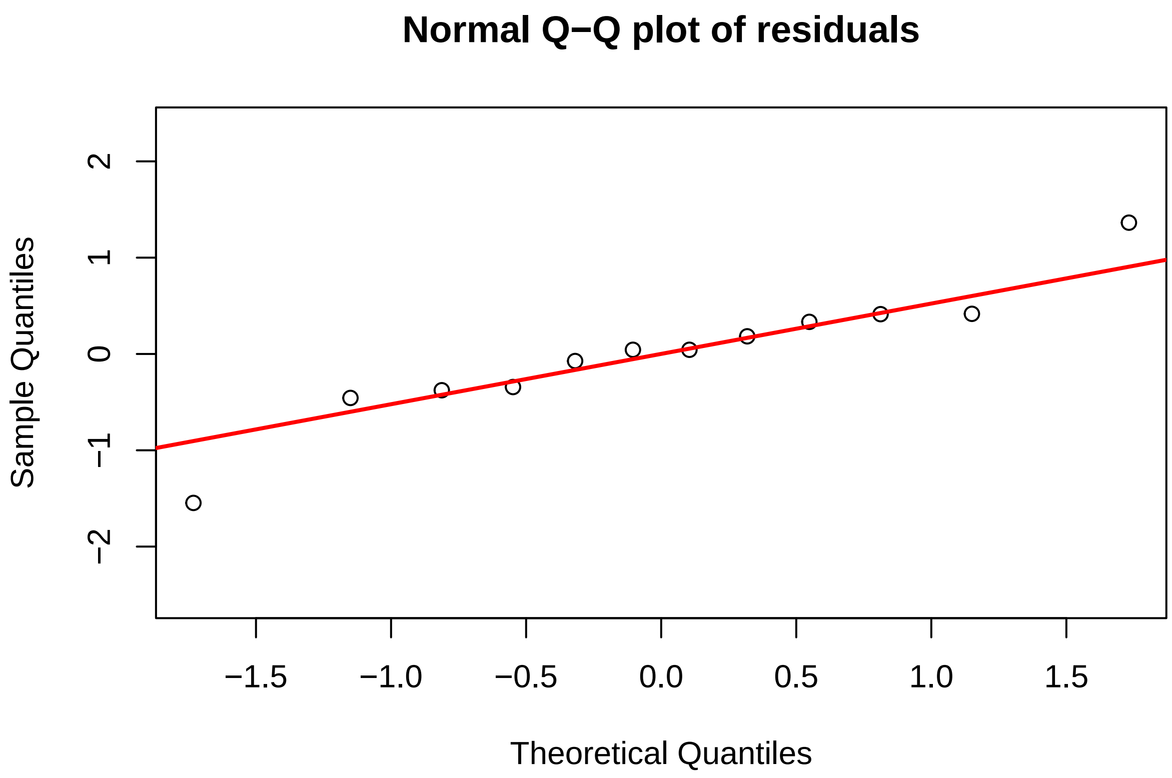
$\therefore 0.652$

(d) Analyze the residuals from this experiment and comment on model adequacy.

```
# Model adequacy checking
plot(WaterPosition, uniformity, main="Water Position vs. Uniformity", xlab="Water Position",
     ylab="Uniformity")
```



```
# Normality test
uniformity_residual = res.aov$residuals
qqnorm(uniformity_residual,ylim=c(min(uniformity_residual)-1,max(uniformity_residual)+1),
main="Normal Q-Q plot of residuals",
xlab="Theoretical Quantiles",
ylab="Sample Quantiles", plot.it=TRUE, datax=FALSE)

qqline(uniformity_residual, col="red", lwd=2, datax= FALSE)
```

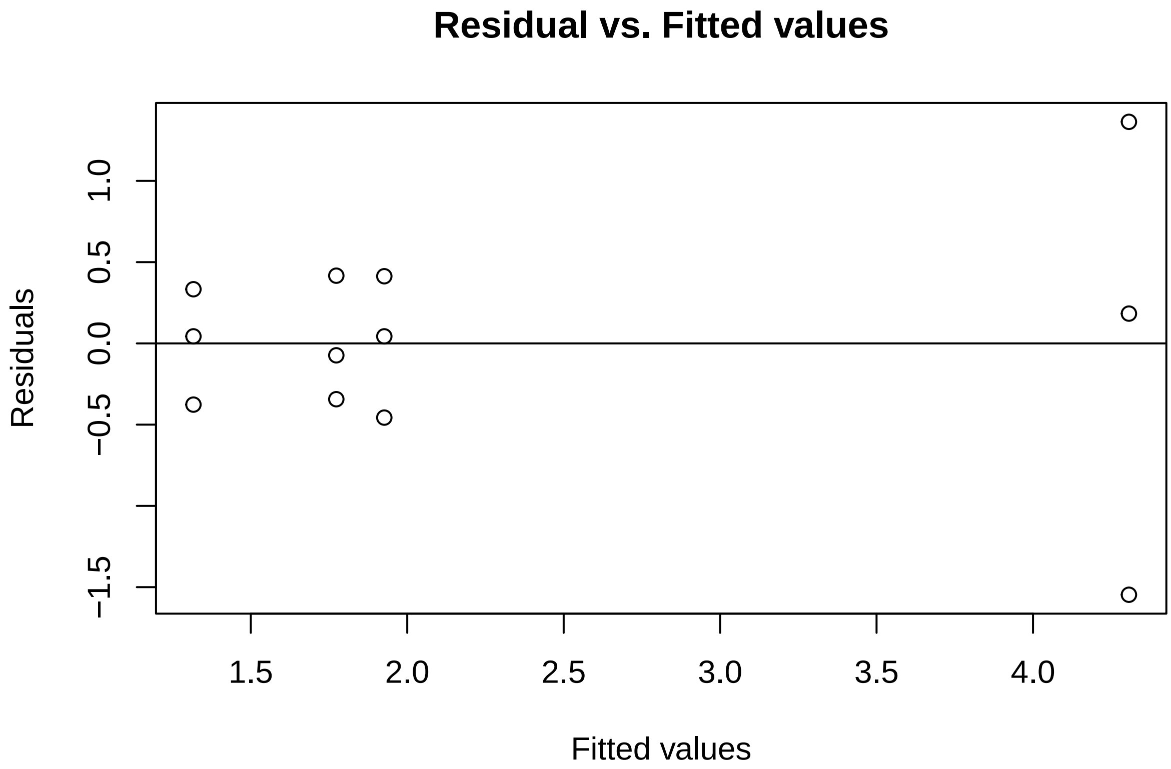


```
shapiro.test(uniformity_residual)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  uniformity_residual
## W = 0.92764, p-value = 0.3558
```

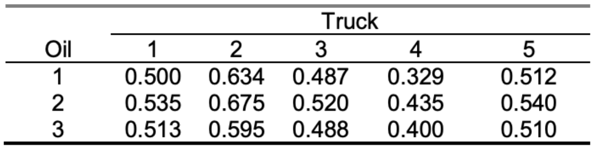
∴ The p value is 0.3558 so the data is likely normally distributed. There's no indication of violation of normality assumption. But there are some outliers.

```
# Variance test
Fitted_values = res.aov$fitted.values
plot(Fitted_values, uniformity_residual, main="Residual vs. Fitted values", xlab="Fitted values"
     , ylab="Residuals")
abline(h=0)
```



∴ Residuals are changing and variances seem different especially for the water position 1 level. Wafer position 1 (fitted value: 4.306667) seems to have a greater variance than othe positions.

Question 3. The effect of three different lubricating oils on fuel economy in diesel truck engines is being studied. Fuel economy is measured using brake-specific fuel consumption after the engine has been running for 15 minutes. Five different truck engines are available for the study, and the experimenters conduct the following randomized complete block design.



	Truck				
Oil	1	2	3	4	5
1	0.500	0.634	0.487	0.329	0.512
2	0.535	0.675	0.520	0.435	0.540
3	0.513	0.595	0.488	0.400	0.510

```
fuel = c(
    0.500, 0.634, 0.487, 0.329, 0.512,
    0.535, 0.675, 0.520, 0.435, 0.540,
    0.513, 0.595, 0.488, 0.400, 0.510)
Oil = c(
    1, 1, 1, 1, 1,
    2, 2, 2, 2, 2,
    3, 3, 3, 3, 3)
batch = c(
    1, 2, 3, 4, 5,
    1, 2, 3, 4, 5,
    1, 2, 3, 4, 5)
Q3_Yield_data = data.frame(fuel, as.factor(Oil), as.factor(batch))
```

(a) Analyze the data from this experiment.

```
# H0: tau1 = tau2 = tau3 = 0
# H1: At least two effect(tau) are not equal to 0

group_by(Q3_Yield_data, factor(Oil)) %>%
    summarise(
        count = n(),
        mean = mean(batch, na.rm = TRUE),
        sd = sd(batch, na.rm = TRUE),
    )
```

```
## # A tibble: 3 x 4
##   `factor(Oil)` count  mean    sd
##   <fct>         <int> <dbl> <dbl>
## 1 1                 5     3  1.46
## 2 2                 5     3  1.46
## 3 3                 5     3  1.46
```

```
# Using ANOVA
res.aov = aov(fuel ~ factor(Oil)+factor(batch), data = Q3_Yield_data)
summary(res.aov)
```

```
##                Df  Sum Sq  Mean Sq F value   Pr(>F)
## factor(Oil)     2 0.00671 0.003353   6.353   0.0223 *
## factor(batch)   4 0.09210 0.023025  43.626 1.78e-05 ***
## Residuals       8 0.00422 0.000528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

∴ The p-value for oil factor is 0.0223, which is between 0.05 and 0.1, so we reject H0. At least two treatment effects(tau i) are not 0.

(b) Use the Tukey method to make comparisons among the three lubricating oils to determine specifically which oils differ in break-specific fuel consumption.

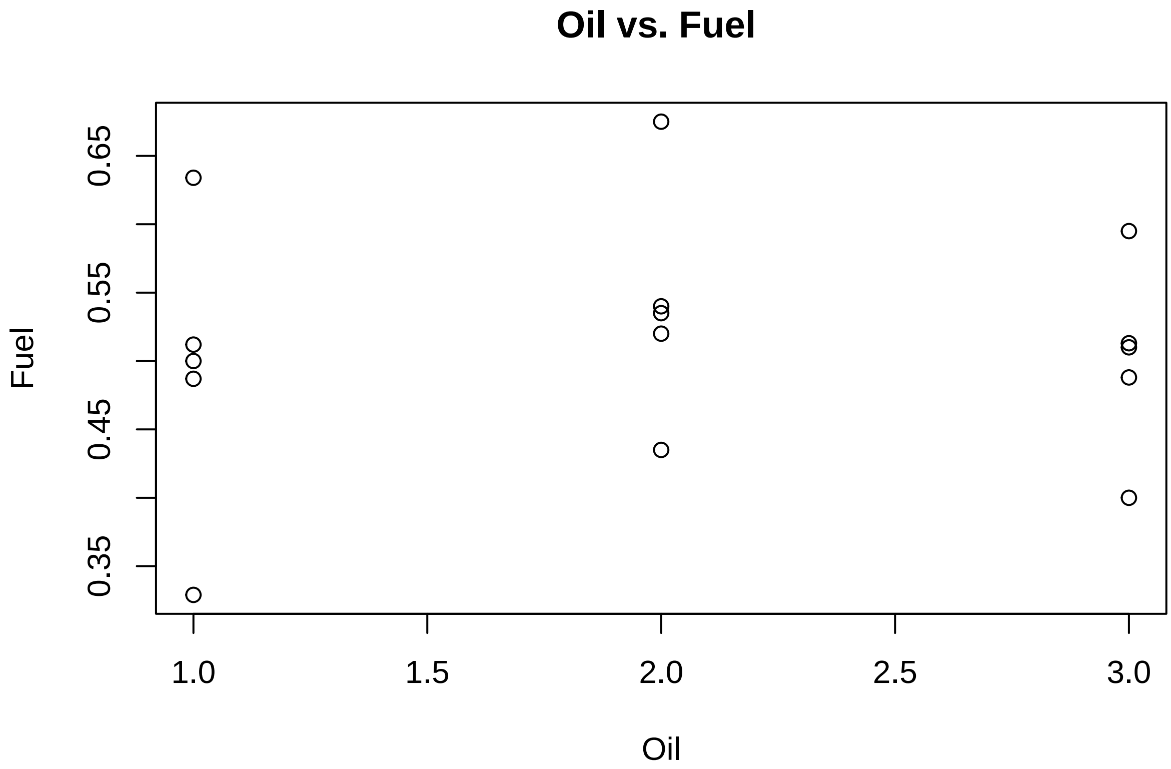
```
TukeyHSD(res.aov, which = "factor(Oil)")
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = fuel ~ factor(Oil) + factor(batch), data = Q3_Yield_data)
##
## $`factor(Oil)`
##        diff          lwr         upr     p adj
## 2-1  0.0486  0.007082078 0.090117922 0.0245809
## 3-1  0.0088 -0.032717922 0.050317922 0.8210970
## 3-2 -0.0398 -0.081317922 0.001717922 0.0594979
```

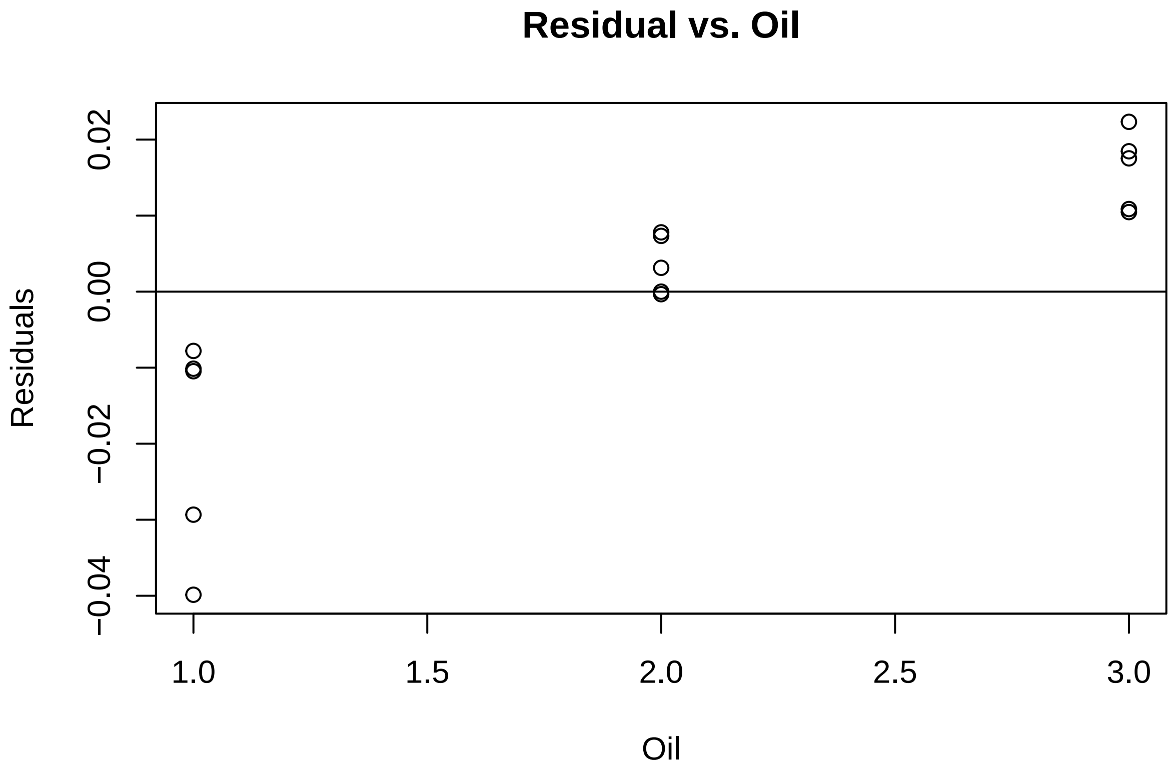
∴ By observing p-values, we can observe that Oil 3 and 1 do not differ (p-value: 0.821), but Oil 2 is different from Oil 1 and 3 in terms of break-specific fuel consumption(p-value: 0.0245 and 0.0595 respectively).

(c) Analyze the residuals from this experiment.

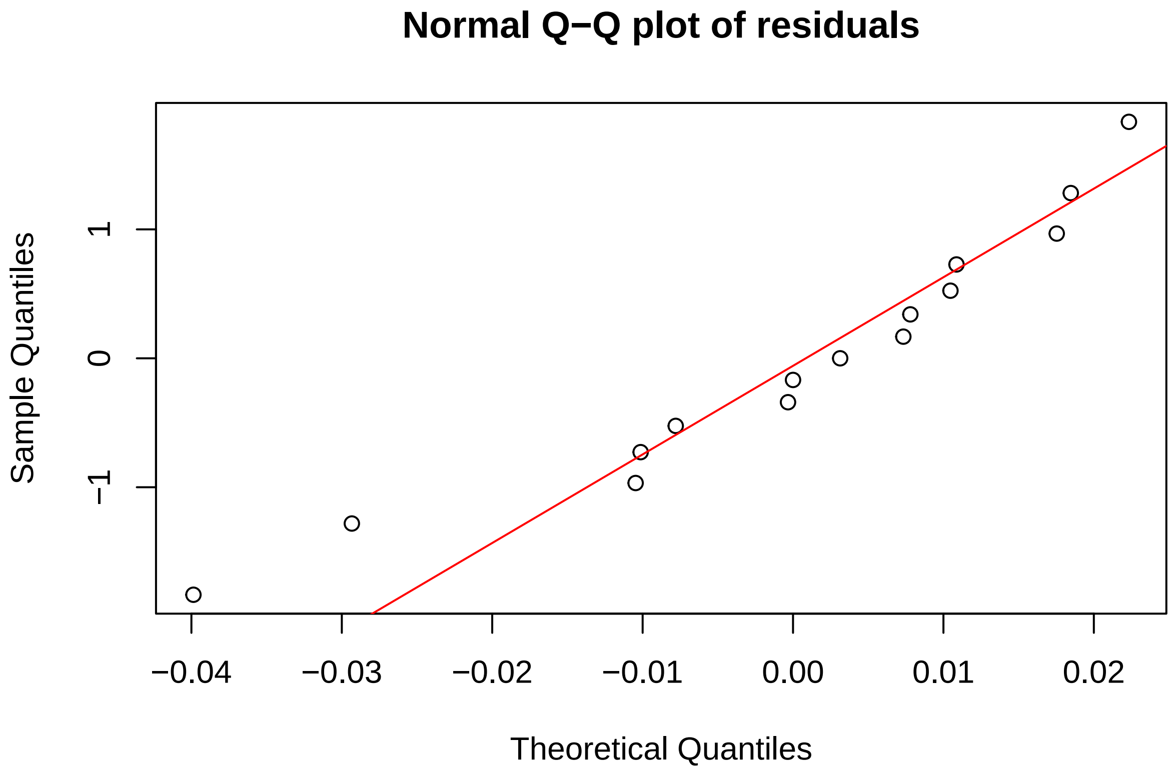
```r
# Oil vs fuel consumption plot
plot(Oil, fuel, main="Oil vs. Fuel ", xlab="Oil", ylab="Fuel")
```



```r
# Model adequacy checking
fuel_residual = res.aov$residuals

# residuals vs. oil plot
qqplot(Oil, fuel_residual, main="Residual vs. Oil", xlab="Oil", ylab="Residuals")
abline(h=0)
```



```
# Normality test
qqnorm(fuel_residual,ylim=c(min(fuel_residual),max(fuel_residual)),
main="Normal Q-Q plot of residuals",
xlab="Sample Quantiles",
ylab="Theoretical Quantiles", plot.it=TRUE, datax=TRUE)

qqline(fuel_residual, col="red", lwd=1, datax= TRUE)
```

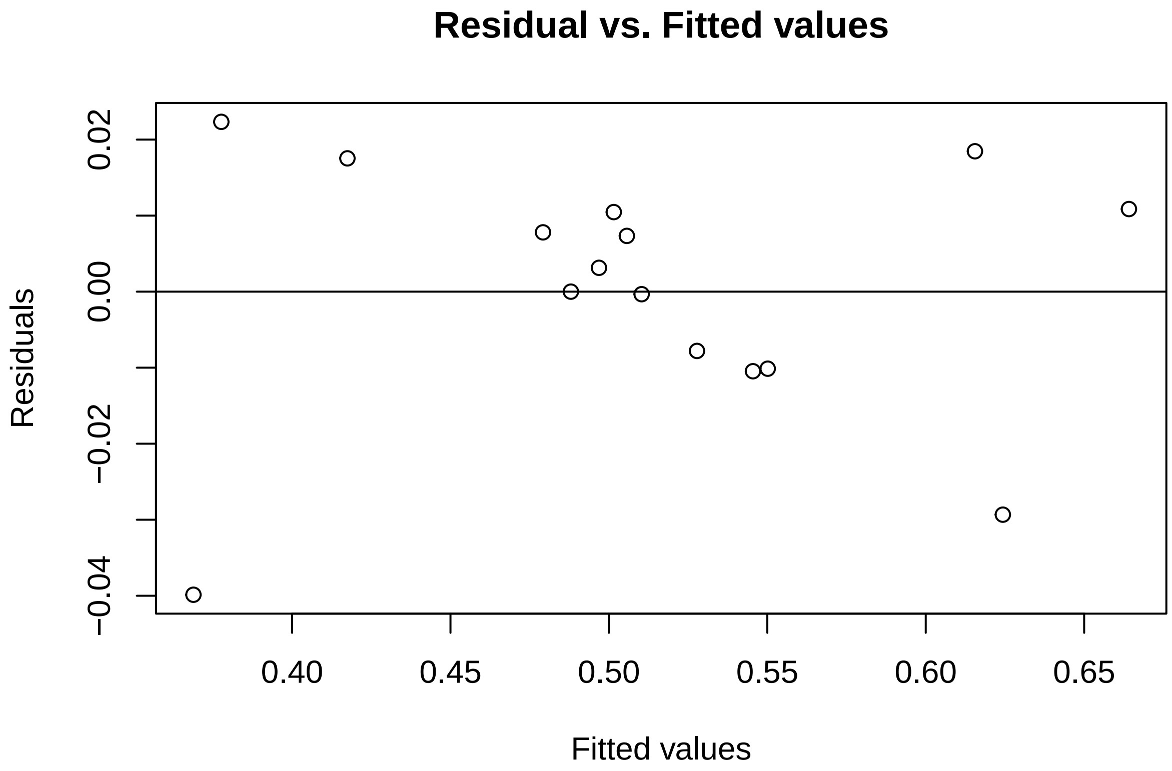


```
shapiro.test(fuel_residual)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fuel_residual
## W = 0.91855, p-value = 0.183
```

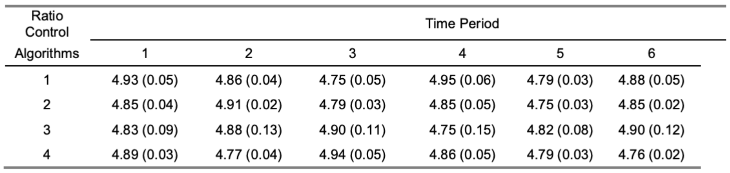
∴ The p value is $0.1836 > 0.05$ so the data is normally distributed. There's no indication of violation of normality assumption.

```
# Variance test
Fitted_values = res.aov$fitted.values
plot(Fitted_values, fuel_residual, main="Residual vs. Fitted values", xlab="Fitted values",
     ylab="Residuals")
abline(h=0)
```



∴ The plot seems to be randomly spread out. There's no indication of violation of variance assumption.

Question 4. Consider the ratio control algorithm experiment described in Section 3.8 of your textbook. The experiment was actually conducted as a randomized block design, where six time periods were selected as the blocks, and all four ratio control algorithms were tested in each time period. The average cell voltage and the standard deviation of voltage (shown in parentheses) for each cell are as follows:



Ratio Control Algorithms	Time Period					
	1	2	3	4	5	6
1	4.93 (0.05)	4.86 (0.04)	4.75 (0.05)	4.95 (0.06)	4.79 (0.03)	4.88 (0.05)
2	4.85 (0.04)	4.91 (0.02)	4.79 (0.03)	4.85 (0.05)	4.75 (0.03)	4.85 (0.02)
3	4.83 (0.09)	4.88 (0.13)	4.90 (0.11)	4.75 (0.15)	4.82 (0.08)	4.90 (0.12)
4	4.89 (0.03)	4.77 (0.04)	4.94 (0.05)	4.86 (0.05)	4.79 (0.03)	4.76 (0.02)

```
voltage = c(
    4.93, 4.86, 4.75, 4.95, 4.79, 4.88,
    4.85, 4.91, 4.79, 4.85, 4.75, 4.85,
    4.83, 4.88, 4.90, 4.75, 4.82, 4.90,
    4.89, 4.77, 4.94, 4.86, 4.79, 4.76
)
algorithm = c(
    1, 1, 1, 1, 1, 1,
    2, 2, 2, 2, 2, 2,
    3, 3, 3, 3, 3, 3,
    4, 4, 4, 4, 4, 4
)
time_block = c(
    1, 2, 3, 4, 5, 6,
    1, 2, 3, 4, 5, 6,
    1, 2, 3, 4, 5, 6,
    1, 2, 3, 4, 5, 6
)
Q4_Yield_data = data.frame(voltage, as.factor(algorithm), as.factor(time_block))
```

(a) Analyze the average cell voltage data. (Use a = 0.05.) Does the choice of ratio control algorithm affect the cell voltage? Justify

```
group_by(Q4_Yield_data, factor(algorithm)) %>%
     summarise(
         count = n(),
         mean = mean(voltage, na.rm = TRUE),
         sd = sd(voltage, na.rm = TRUE),
     )
```

```
## # A tibble: 4 x 4
##   `factor(algorithm)` count  mean     sd
##   <fct>               <int> <dbl>  <dbl>
## 1 1                       6  4.86 0.0780
## 2 2                       6  4.83 0.0557
## 3 3                       6  4.85 0.0585
## 4 4                       6  4.84 0.0729
```

```
res.aov = aov(voltage ~ factor(algorithm)+factor(time_block), data = Q4_Yield_data)
summary(res.aov)
```

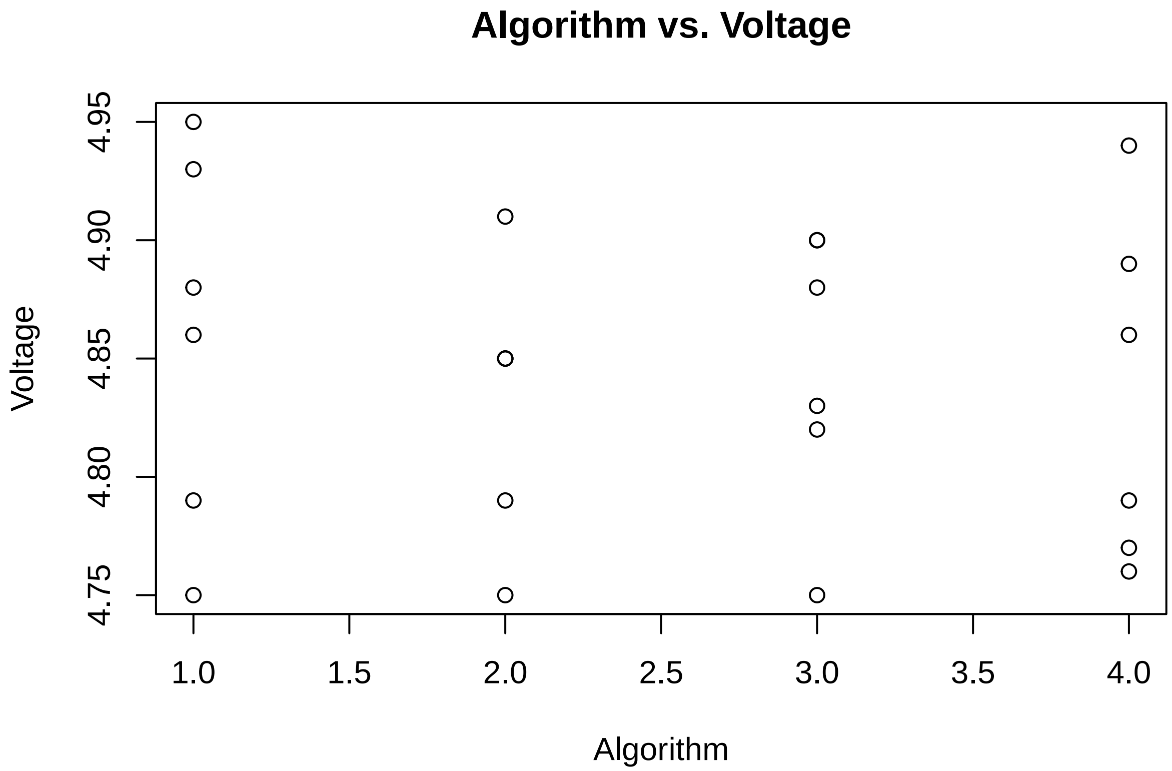
```
##                    Df  Sum Sq  Mean Sq F value Pr(>F)
## factor(algorithm)   3 0.00275 0.000915   0.190  0.901
## factor(time_block)  5 0.01744 0.003488   0.725  0.615
## Residuals          15 0.07218 0.004812
```

```
ta = qtukey(0.05, 3, 15, lower.tail = FALSE) #3.673378
```

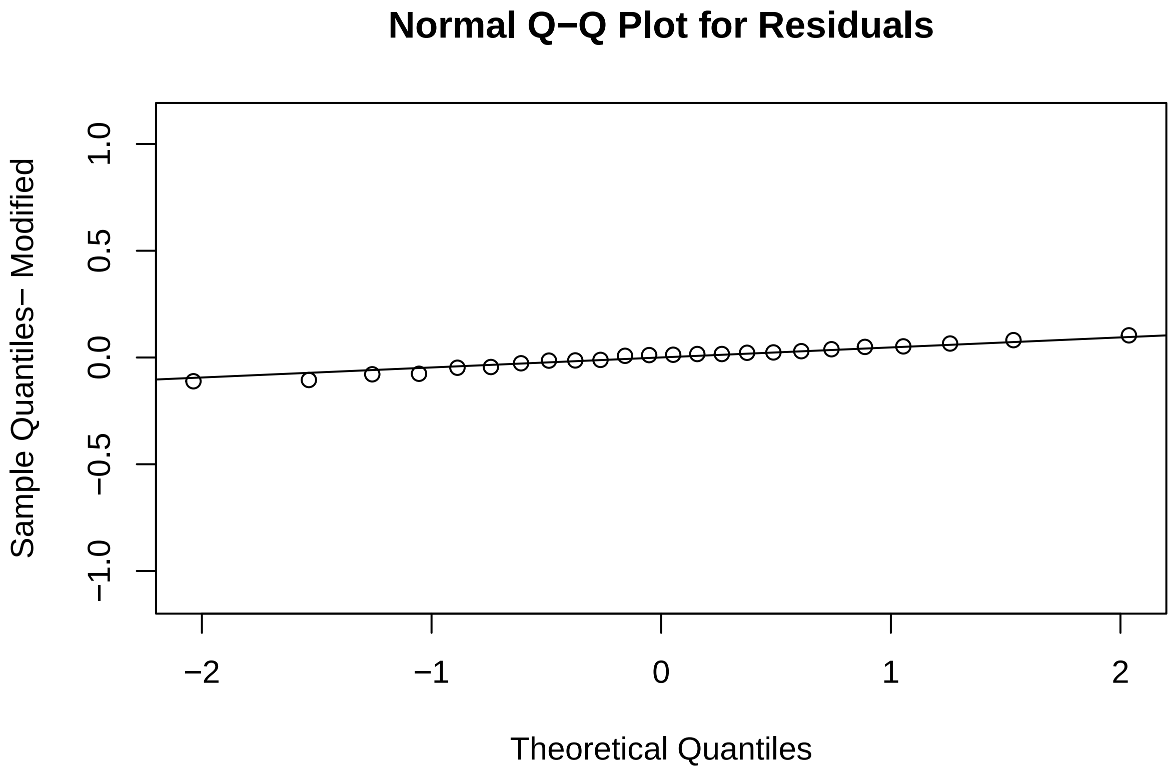
∴ The p-value for algorithm factor is 0.901 so we fail to reject H0 so there's no significant difference between algorithms on mean value of cell voltage. The choice of algorithm does not affect the mean cell voltage.

(b) Conduct any residual analyses that seem appropriate.

```
# algorithm vs. voltage plot
plot(algorithm, voltage, main="Algorithm vs. Voltage", xlab="Algorithm", ylab="Voltage")
```



```
# Model adequacy test
# Normality
voltage_residual = res.aov$residuals
qqnorm(voltage_residual, ylim=c(min(voltage_residual)-1,max(voltage_residual)+1),
       main = "Normal Q-Q Plot for Residuals",
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles- Modified",
       plot.it = TRUE, datax = FALSE)

qqline(voltage_residual, datax = FALSE, distribution = qnorm)
```

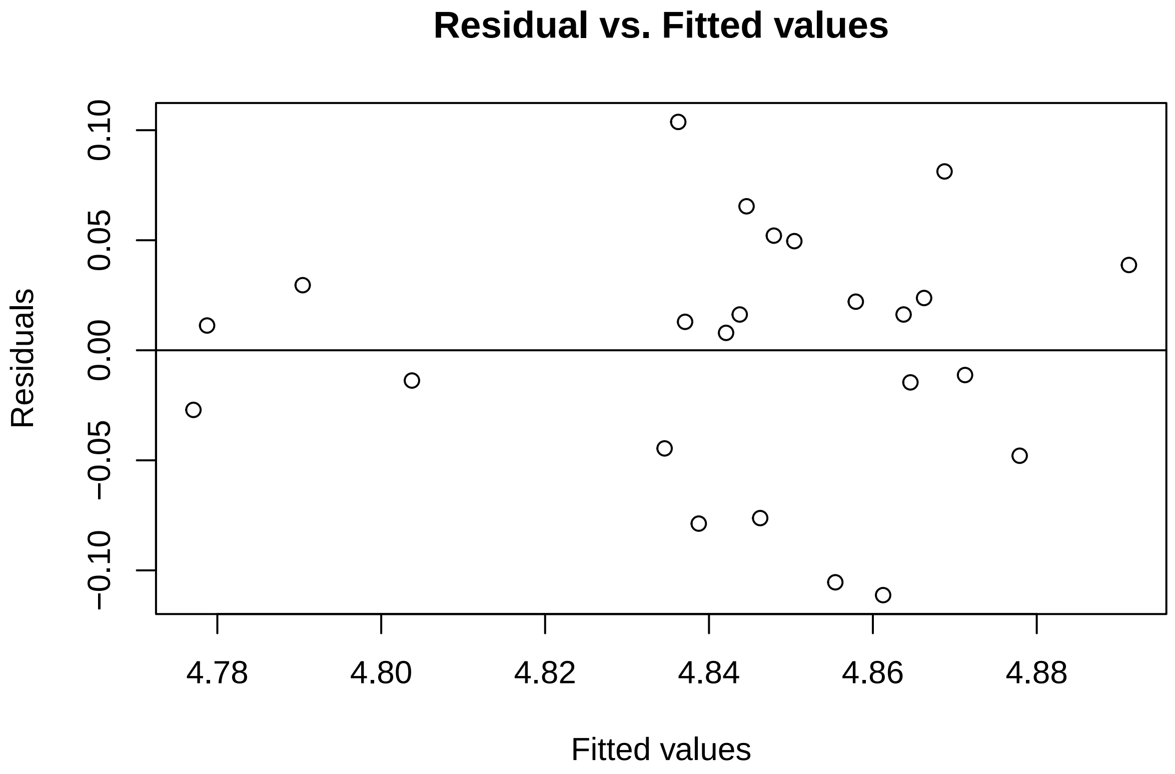


```
# Test normality using Shapiro Wilks
shapiro.test(voltage_residual)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  voltage_residual
## W = 0.96926, p-value = 0.6487
```

∴ The p-value is 0.6487, which is greater than 0.05, so we fail to reject the null hypothesis that the data is normally distributed. Therefore, the normality assumption is reasonable.

```
# Variance test
Fitted_values = res.aov$fitted.values
plot(Fitted_values, voltage_residual, main="Residual vs. Fitted values", xlab="Fitted values", ylab="Re
abline(h=0)
```



∴ The plots seem to be spread out. There's no indication of violation of constant variance assumption.

(c) Which ratio control algorithm would you select if your objective is to reduce the average cell voltage?

```
Tukey = TukeyHSD(x=res.aov, "factor(algorithm)", conf.level = 0.95)
Tukey
```

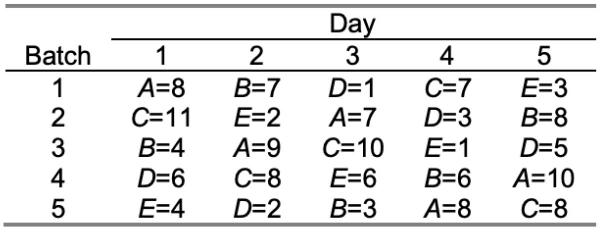
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = voltage ~ factor(algorithm) + factor(time_block), data = Q4_Yield_data)
##
## $`factor(algorithm)`
##             diff        lwr        upr     p adj
## 2-1 -0.026666667 -0.1420960 0.08876263 0.9082750
## 3-1 -0.013333333 -0.1287626 0.10209596 0.9867674
## 4-1 -0.025000000 -0.1404293 0.09042930 0.9226897
## 3-2  0.013333333 -0.1020960 0.12876263 0.9867674
## 4-2  0.001666667 -0.1137626 0.11709596 0.9999729
## 4-3 -0.011666667 -0.1270960 0.10376263 0.9910326
```

```
group_by(Q4_Yield_data, factor(algorithm)) %>%
     summarise(
         count = n(),
         mean = mean(voltage, na.rm = TRUE),
         sd = sd(voltage, na.rm = TRUE),
     )
```

```
## # A tibble: 4 x 4
##   ‘factor(algorithm)‘ count  mean     sd
##   <fct>               <int> <dbl>  <dbl>
## 1 1                       6  4.86 0.0780
## 2 2                       6  4.83 0.0557
## 3 3                       6  4.85 0.0585
## 4 4                       6  4.84 0.0729
```

∴ Since the p-value for difference between any pair is large, the ratio control algorithm does not have a significant effect on mean cell voltage, but since the algorithm 2 has a smaller standard deviation, I would select algorithm 2.

Question 5. The effect of five different ingredients (A, B, C, D, E) on reaction time of a chemical process is being studied. Each batch of new material is only large enough to permit five runs to be made. Furthermore, each run requires approximately 1 1/2 hours, so only five runs can be made in one day. The experimenter decides to run the experiment as a Latin square so that day and batch effects can be systematically controlled. She obtains the data that follow. Analyze the data from this experiment (use a = 0.05) and draw conclusions.



	Day				
Batch	1	2	3	4	5
1	*A*=8	*B*=7	*D*=1	*C*=7	*E*=3
2	*C*=11	*E*=2	*A*=7	*D*=3	*B*=8
3	*B*=4	*A*=9	*C*=10	*E*=1	*D*=5
4	*D*=6	*C*=8	*E*=6	*B*=6	*A*=10
5	*E*=4	*D*=2	*B*=3	*A*=8	*C*=8

```
rm(list = ls())
Day<-c(rep(1,5), rep(2,5),rep(3,5), rep(4,5),rep(5,5))
Batch<-c(c(1,3,2,4,5),c(3,1,4,5,2),c(2,5,3,1,4),c(5,4,1,2,3),c(4,2,5,3,1))
Ingredient<-rep(c(rep('A',1), rep('B',1),rep('C',1), rep('D',1),rep('E',1)),5)
Reaction_time<-c(8,4,11,6,4,
                9,7,8,2,2,
                7,3,10,1,6,
                8,6,7,3,1,
                10,8,8,5,3)

Q5_data <- data.frame(Reaction_time, Ingredient,Day,Batch)
```

```
# H0: tau1 = tau2 = tau3 = tau4 = tau5 = 0 (taui is the effect of ith ingredient)
# H1: At least two effect(tau) are not equal to 0

res.aov <-aov(Reaction_time~factor(Ingredient)+factor(Day)+factor(Batch),data=Q5_data)
summary(res.aov)
```

```
##                     Df Sum Sq Mean Sq F value   Pr(>F)    
## factor(Ingredient)  4 141.44   35.36  11.309 0.000488 ***
## factor(Day)         4  12.24    3.06   0.979 0.455014    
## factor(Batch)       4  15.44    3.86   1.235 0.347618    
## Residuals          12  37.52    3.13                     
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

∴ Since the p-value for Ingredient effect (0.000488) is much smaller than 0.05, so we reject H0. Then at least two effects are not equal to 0.

```
# multiple comparison test
Tukey = TukeyHSD(x=res.aov, "factor(Ingredient)", conf.level = 0.95)
Tukey
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
## 
## Fit: aov(formula = Reaction_time ~ factor(Ingredient) + factor(Day) + factor(Batch), data = Q5_data)
## 
## $‘factor(Ingredient)‘
##     diff        lwr        upr     p adj
## B-A -2.8 -6.3646078  0.7646078 0.1539433
## C-A  0.4 -3.1646078  3.9646078 0.9960012
## D-A -5.0 -8.5646078 -1.4353922 0.0055862
## E-A -5.2 -8.7646078 -1.6353922 0.0041431
## C-B  3.2 -0.3646078  6.7646078 0.0864353
## D-B -2.2 -5.7646078  1.3646078 0.3365811
## E-B -2.4 -5.9646078  1.1646078 0.2631551
## D-C -5.4 -8.9646078 -1.8353922 0.0030822
## E-C -5.6 -9.1646078 -2.0353922 0.0023007
## E-D -0.2 -3.7646078  3.3646078 0.9997349
```

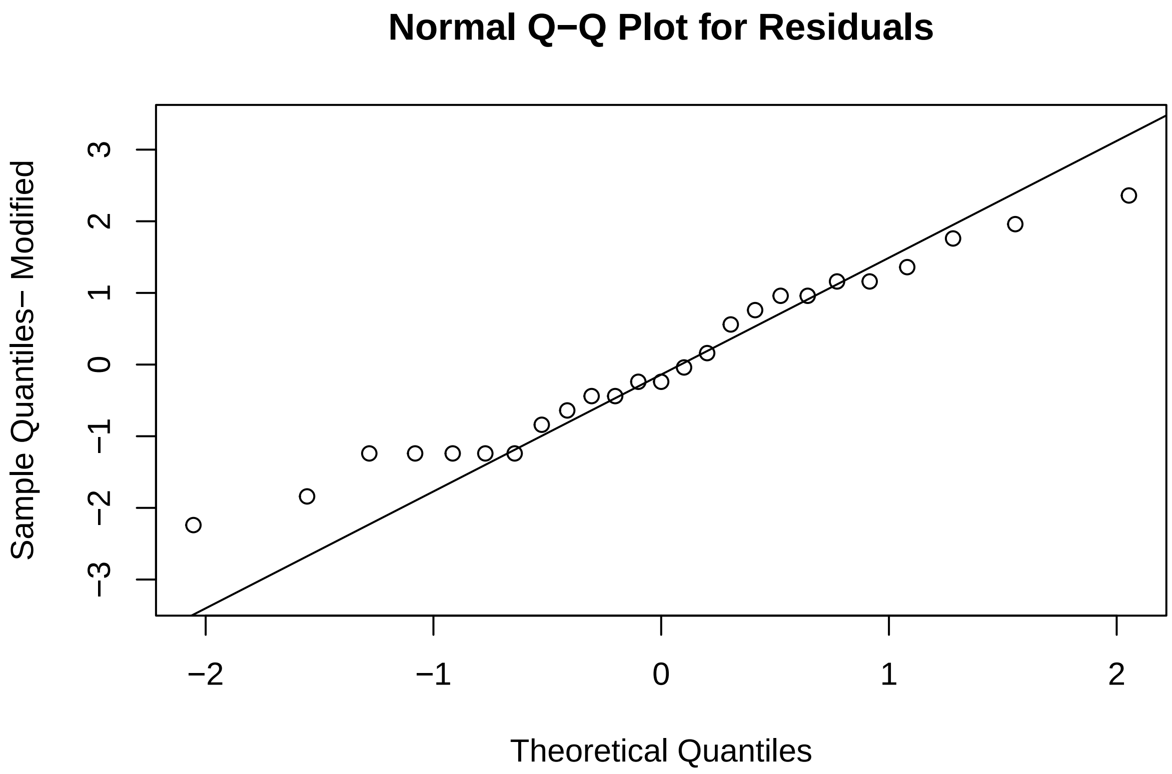
```
# treatment(ingredient) effect values
model.tables(res.aov,se=TRUE)
```

```
## Tables of effects
## 
##  factor(Ingredient) 
## factor(Ingredient)
##     A     B     C     D     E 
##  2.52 -0.28  2.92 -2.48 -2.68 
## 
##  factor(Day) 
## factor(Day)
##     1     2     3     4     5 
##  0.72 -0.28 -0.48 -0.88  0.92 
## 
##  factor(Batch) 
## factor(Batch)
##     1     2     3     4     5 
## -0.68  0.32 -0.08  1.32 -0.88 
## 
```

```
## Standard errors of effects
##          factor(Ingredient) factor(Day) factor(Batch)
##                      0.7908      0.7908        0.7908
## replic.                   5           5             5
```

∴ The pairs D-A, E-A, D-C, and E-C are significantly different within the pair. Therefore, other than ingredient B, the other ingredients have a significant effect on reaction time. Ingredients A and C has positive treatment effect on reaction time, while ingredients D and E has negative treatment effect on reaction time.

```
# Model adequacy test
# Normality
Reaction_time_residual = res.aov$residuals
qqnorm(Reaction_time_residual, ylim=c(min(Reaction_time_residual)-1,max(Reaction_time_residual)+1), mai
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles- Modified",
       plot.it = TRUE, datax = FALSE)

qqline(Reaction_time_residual, datax = FALSE, distribution = qnorm)
```

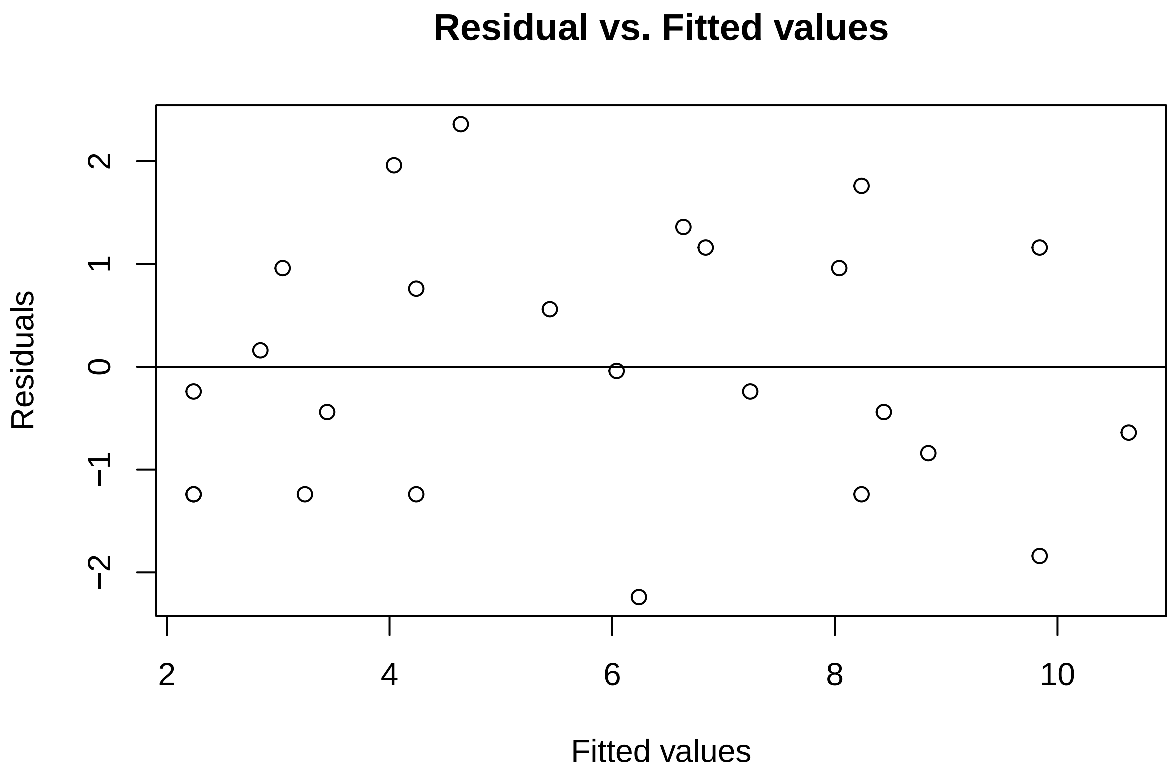


```
# Test normality using Shapiro Wilks
shapiro.test(Reaction_time_residual)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Reaction_time_residual
## W = 0.96606, p-value = 0.5476
```

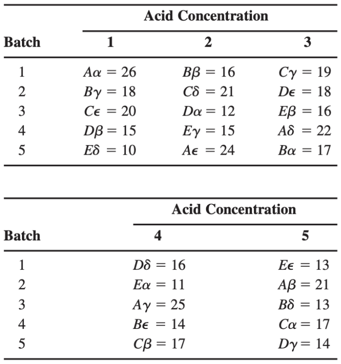
∴ The p-value is 0.5476, which is greater than 0.05, so we fail to reject the null hypothesis that the data is normally distributed. Therefore, the normality assumption is reasonable.

```
# Variance test
Fitted_values = res.aov$fitted.values
plot(Fitted_values, Reaction_time_residual, main="Residual vs. Fitted values", xlab="Fitted values"
     , ylab="Residuals")
abline(h=0)
```



∴ The plots seem to be ramdomly spread out. There's no indication of violation of constant variance assumption.

Question 6. The yield of a chemical process was measured using five batches of raw material, five acid concentrations, five standing times, (A, B, C, D, E) and five catalyst concentrations (a, b, g, d, e). The Graeco-Latin square that follows was used. Analyze the data from this experiment (use a = 0.05) and draw conclusions.



	Acid Concentration				
Batch	1	2	3	4	5
1	$A\alpha = 26$	$B\beta = 16$	$C\gamma = 19$	$D\delta = 16$	$E\epsilon = 13$
2	$B\gamma = 18$	$C\delta = 21$	$D\epsilon = 18$	$E\alpha = 11$	$A\beta = 21$
3	$C\epsilon = 20$	$D\alpha = 12$	$E\beta = 16$	$A\gamma = 25$	$B\delta = 13$
4	$D\beta = 15$	$E\gamma = 15$	$A\delta = 22$	$B\epsilon = 14$	$C\alpha = 17$
5	$E\delta = 10$	$A\epsilon = 24$	$B\alpha = 17$	$C\beta = 17$	$D\gamma = 14$

```
Yield = c(26,18,20,15,10,
          16,21,12,15,24,
          19,18,16,22,17,
          16,11,25,14,17,
          13,21,13,17,14)
Acid_conc = c(rep(1,5), rep(2,5), rep(3,5), rep(4,5), rep(5,5))
Batch = c(rep(c(1,2,3,4,5),5))
Standing_time = c("A","B","C","D","E",
                  "B","C","D","E","A",
                  "C","D","E","A","B",
                  "D","E","A","B","C",
                  "E","A","B","C","D") #Treatment
Catalyst_conc = c("a","g","e","b","d",
                  "b","d","a","g","e",
                  "g","e","b","d","a",
                  "d","a","g","e","b",
                  "e","b","d","a","g") #Treatment
Q6_data <- data.frame(Yield, Acid_conc, Batch, Standing_time, Catalyst_conc)
```

```
res.aov <-aov(Yield~factor(Standing_time)+factor(Catalyst_conc)+factor(Acid_conc)+factor(Batch),
              data=Q6_data)
summary(res.aov)
```

```
##                         Df Sum Sq Mean Sq F value   Pr(>F)
## factor(Standing_time)  4  342.8   85.70  14.650 0.000941 ***
## factor(Catalyst_conc)  4   12.0    3.00   0.513 0.728900
## factor(Acid_conc)      4   24.4    6.10   1.043 0.442543
## factor(Batch)          4   10.0    2.50   0.427 0.785447
## Residuals              8   46.8    5.85
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

∴ Since only the p-value for Standing_time effect (0.00941) is smaller than 0.05, at least two standing effects are different and the choice of Catalyst concentrations does not have significant effect on the yield of a chemical process.

```
# multiple comparison test
Tukey = TukeyHSD(x=res.aov, "factor(Standing_time)", conf.level = 0.95)
Tukey
```

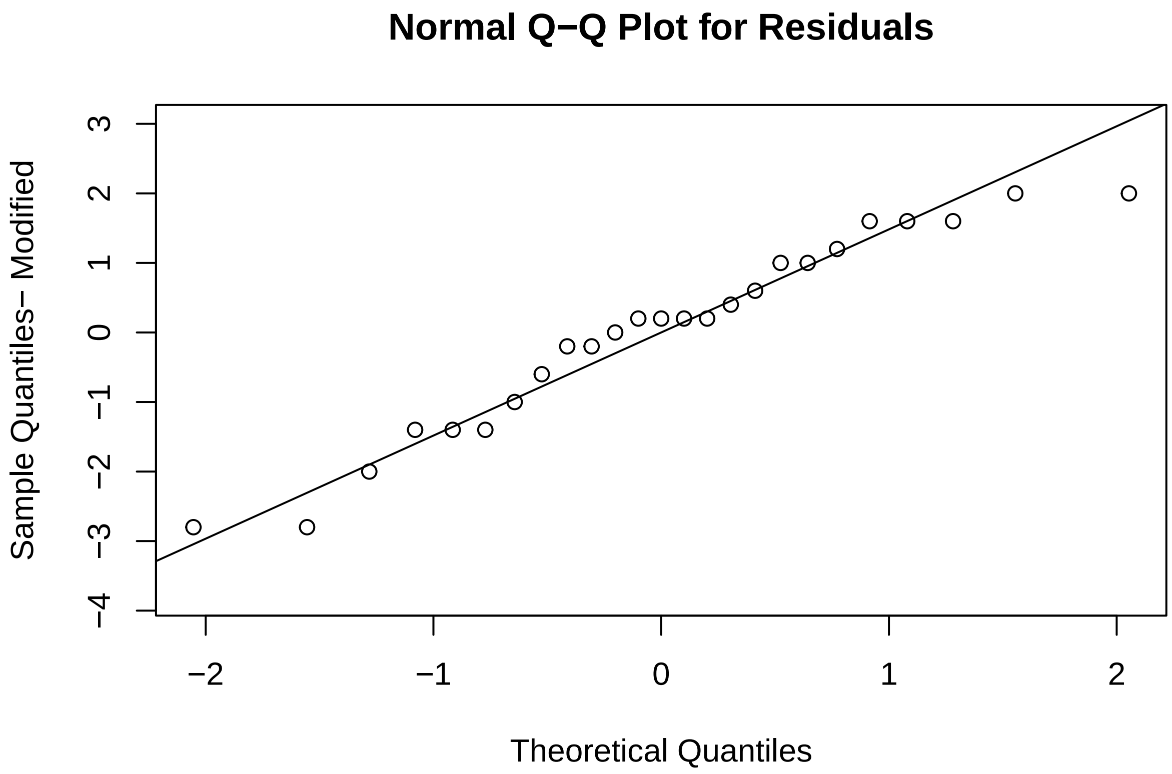
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Yield ~ factor(Standing_time) + factor(Catalyst_conc) + factor(Acid_conc) + facto
##
## $`factor(Standing_time)`
##      diff        lwr        upr     p adj
## B-A  -8.0 -13.284751 -2.7152488 0.0051639
## C-A  -4.8 -10.084751  0.4847512 0.0770797
## D-A  -8.6 -13.884751 -3.3152488 0.0032815
## E-A -10.6 -15.884751 -5.3152488 0.0008219
## C-B   3.2  -2.084751  8.4847512 0.3087034
## D-B  -0.6  -5.884751  4.6847512 0.9939694
## E-B  -2.6  -7.884751  2.6847512 0.4837165
## D-C  -3.8  -9.084751  1.4847512 0.1869031
## E-C  -5.8 -11.084751 -0.5152488 0.0317351
## E-D  -2.0  -7.284751  3.2847512 0.6948188
```

```
# treatment(ingredient) effect values
model.tables(res.aov,se=TRUE)
```

```
## Tables of effects
##
##  factor(Standing_time)
## factor(Standing_time)
##    A    B    C    D    E
##  6.4 -1.6  1.6 -2.2 -4.2
##
##  factor(Catalyst_conc)
## factor(Catalyst_conc)
##    a    b    d    e    g
## -0.6 -0.2 -0.8  0.6  1.0
##
##  factor(Acid_conc)
## factor(Acid_conc)
##    1    2    3    4    5
##  0.6  0.4  1.2 -0.6 -1.6
##
##  factor(Batch)
## factor(Batch)
##    1    2    3    4    5
##  0.8  0.6  0.0 -0.6 -0.8
##
## Standard errors of effects
##         factor(Standing_time) factor(Catalyst_conc) factor(Acid_conc)
##                         1.082                 1.082             1.082
## replic.                      5                      5                  5
##          factor(Batch)
##                  1.082
## replic.              5
```

∴ The pairs, B-A, D-A,E-A, and E-C are significantly different within each pair. Also from the model.tables, we can see that the standing time A and C has positive treatment effect on yield, while standing time B, D and E has negative treatment effect on yield. But the effect of standing time B is not significant.

```
# Model adequacy test
# Normality
Yield_residual = res.aov$residuals
qqnorm(Yield_residual, ylim=c(min(Yield_residual)-1,max(Yield_residual)+1), main = "Normal Q-Q Plot for
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles- Modified",
       plot.it = TRUE, datax = FALSE)

qqline(Yield_residual, datax = FALSE, distribution = qnorm)
```

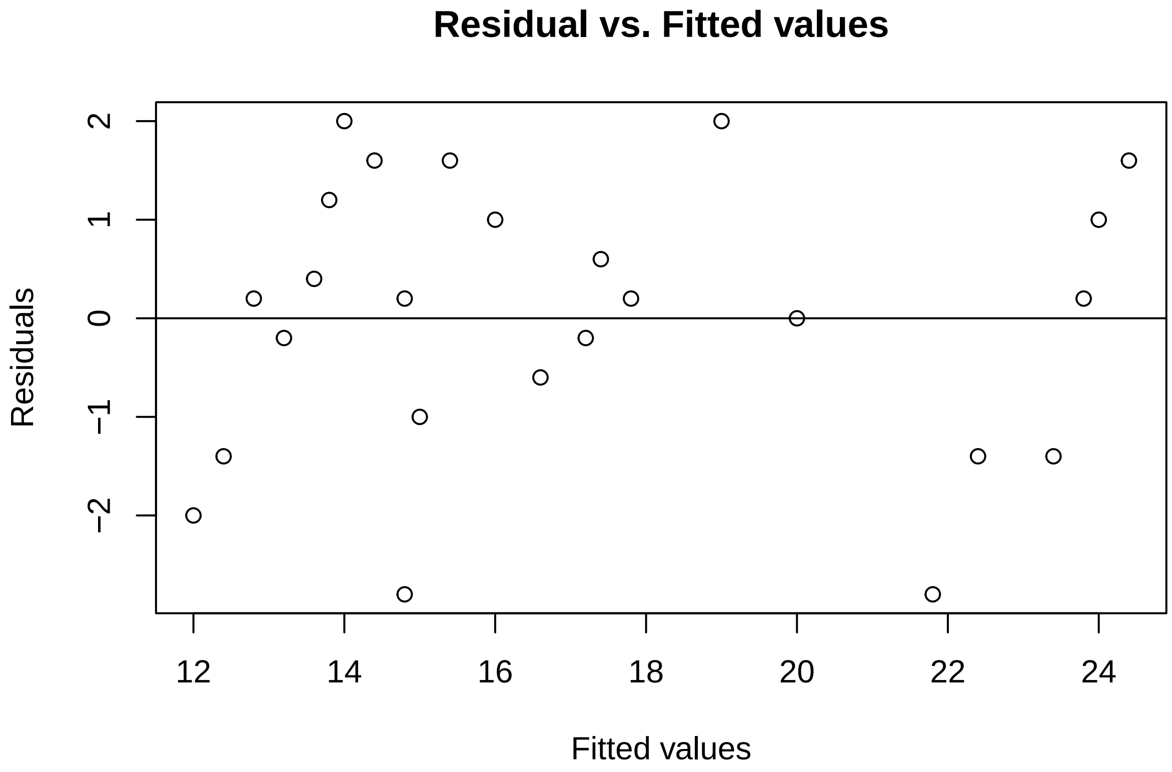


```
# Test normality using Shapiro Wilks
shapiro.test(Yield_residual)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Yield_residual
## W = 0.94721, p-value = 0.2167
```

∴ The p-value is 0.2167, which is greater than 0.05, so we fail to reject the null hypothesis that the data is normally distributed. Therefore, the normality assumption is reasonable.

```
# Variance test
Fitted_values = res.aov$fitted.values
plot(Fitted_values, Yield_residual, main="Residual vs. Fitted values", xlab="Fitted values", ylab="Resi
abline(h=0)
```



∴ The plots seem to be ramdomly spread out. There's no indication of violation of constant variance assumption.