

Lab 4b

Jas Sur

3/1/2022

Question 1

- a. Set up the data frame, named “chem”, with “Bolt” and “Chemical” factors, “strength” as response.

```
setup_q_1 <- function()
{
  chemical.data <- rep(c(1,2,3,4), each=5)
  bolt.data <- rep(c(1,2,3,4,5), times=4)

  strength <- c(73,68,74,71,67,73,67,75,72,70,75,68,78,73,68,73,71,75,75,69)

  effect <- data.frame(strength, chemical.data, bolt.data)
  effect$chemical <- factor(chemical.data, levels=c(1,2,3,4),
                           labels = c("Chemical1", "Chemical2", "Chemical3",
                                       "Chemical4"))
  effect$bolt <- factor(bolt.data, levels=c(1,2,3,4,5),
                       labels=c("Bolt1", "Bolt2", "Bolt3", "Bolt4", "Bolt5"))
  return (effect)
}
effect <-setup_q_1()
str(effect)

## 'data.frame': 20 obs. of 5 variables:
## $ strength : num 73 68 74 71 67 73 67 75 72 70 ...
## $ chemical.data: num 1 1 1 1 1 2 2 2 2 2 ...
## $ bolt.data : num 1 2 3 4 5 1 2 3 4 5 ...
## $ chemical : Factor w/ 4 levels "Chemical1","Chemical2",...: 1 1 1 1 1 2 2 2 2 2 ...
## $ bolt : Factor w/ 5 levels "Bolt1","Bolt2",...: 1 2 3 4 5 1 2 3 4 5 ...

head(effect)

## strength chemical.data bolt.data chemical bolt
## 1 73 1 1 Chemical1 Bolt1
## 2 68 1 2 Chemical1 Bolt2
## 3 74 1 3 Chemical1 Bolt3
## 4 71 1 4 Chemical1 Bolt4
## 5 67 1 5 Chemical1 Bolt5
## 6 73 2 1 Chemical2 Bolt1
```

```
attach(effect)
```

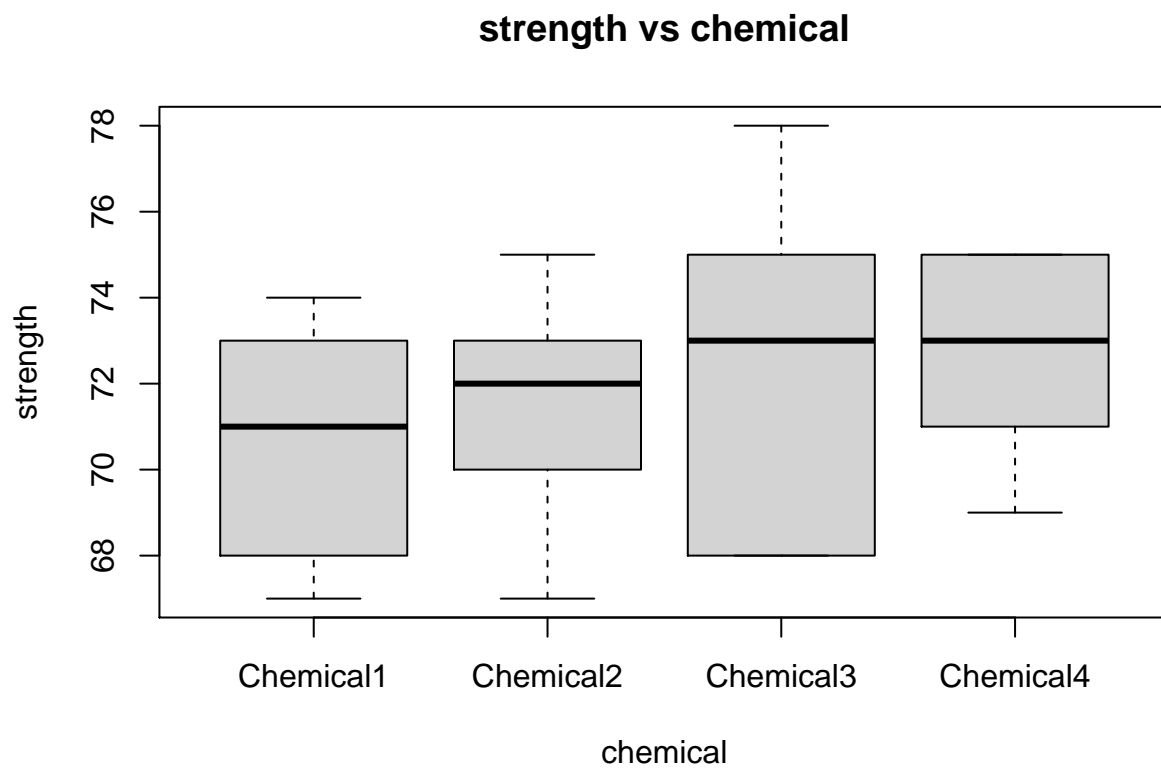
b.i. Any evidence that the Chemical affect Strength ?

Note: Chemical is the treatment factor.

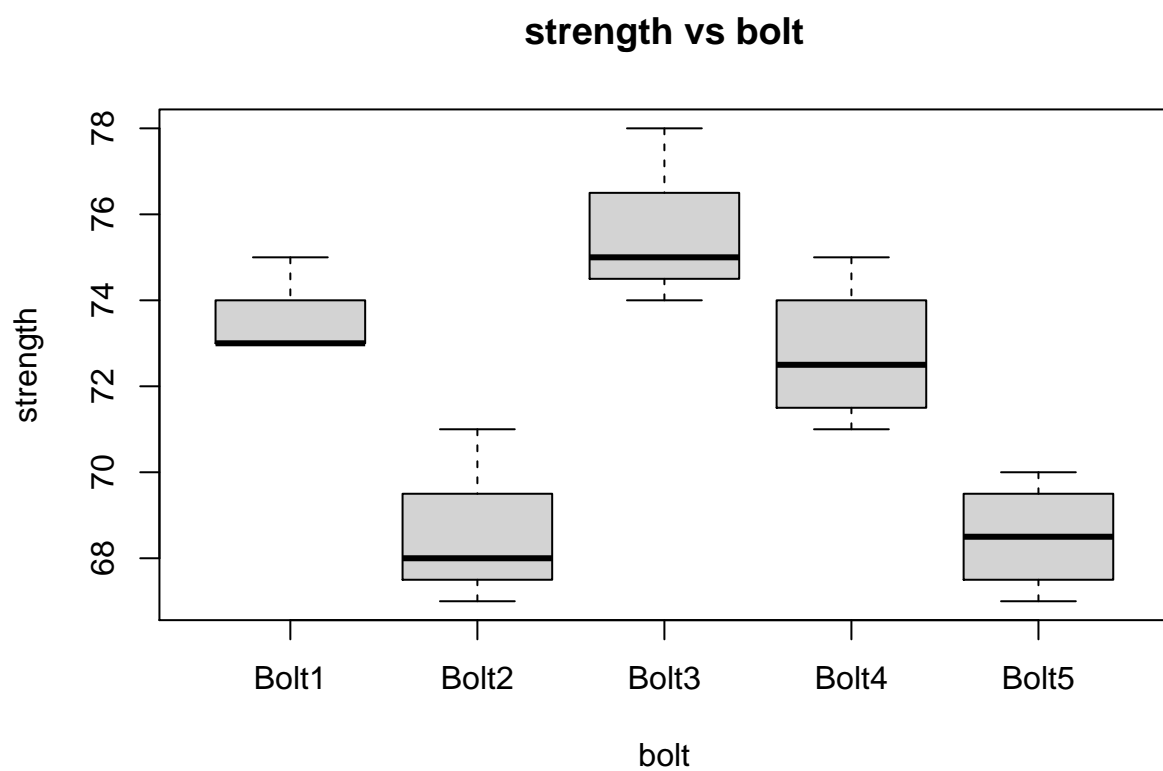
```
tapply(strength, list(chemical, bolt), mean)
```

```
##           Bolt1 Bolt2 Bolt3 Bolt4 Bolt5
## Chemical1    73    68    74    71    67
## Chemical2    73    67    75    72    70
## Chemical3    75    68    78    73    68
## Chemical4    73    71    75    75    69
```

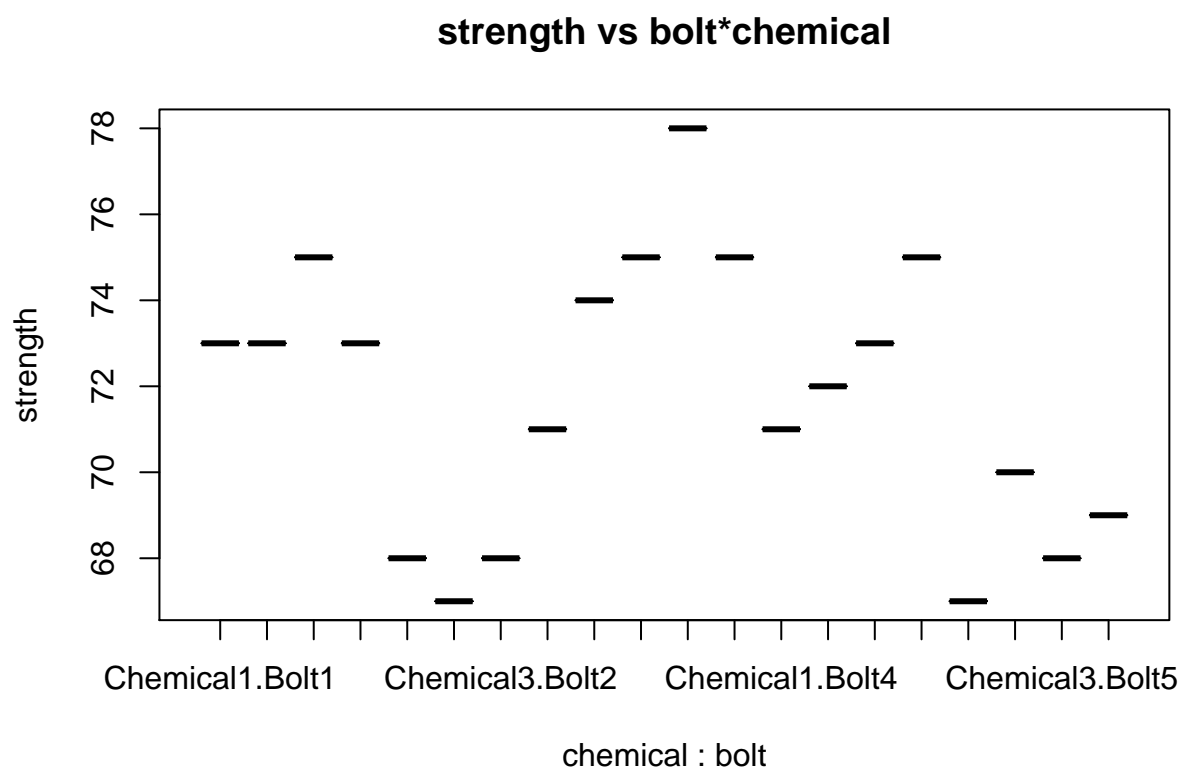
```
boxplot(strength~chemical, main="strength vs chemical")
```



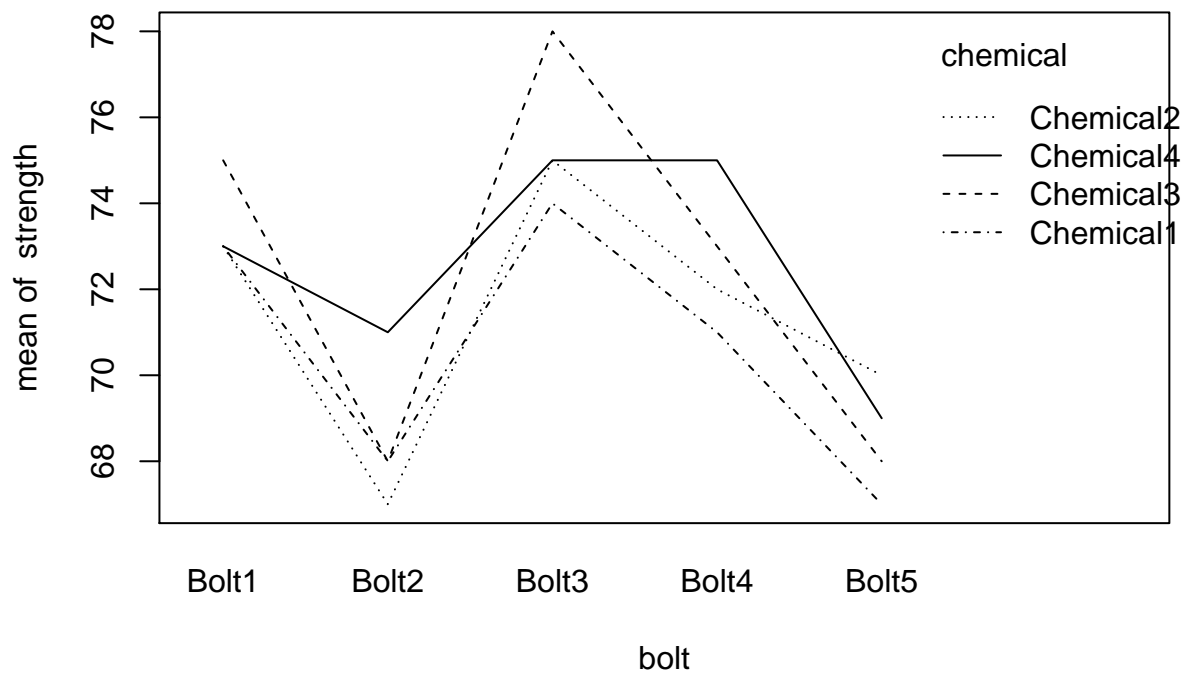
```
boxplot(strength~bolt, main="strength vs bolt")
```



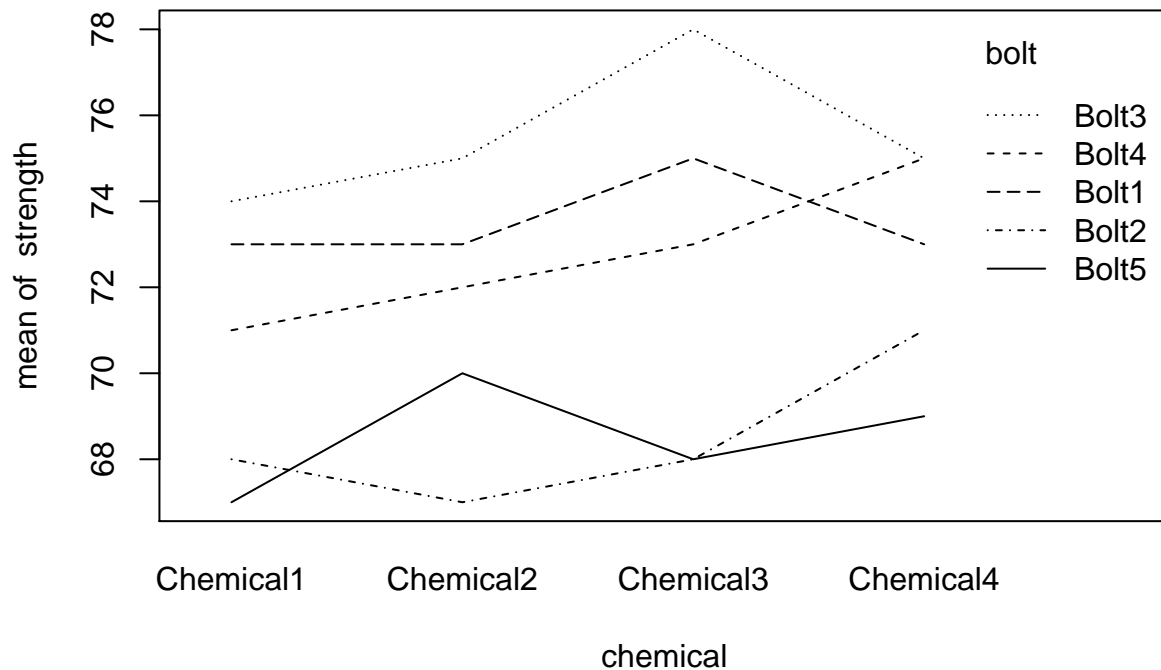
```
boxplot(strength~chemical*bolt, main="strength vs bolt*chemical")
```



```
interaction.plot(bolt, chemical, strength)
```



```
interaction.plot(chemical, bolt, strength)
```



There appears to be strong interaction between bolt and chemical on it appears to be causing a varying response

```
effect.interaction.mod <- aov(strength~chemical*bolt, data=effect)
summary.aov(effect.interaction.mod)
```

```
##           Df Sum Sq Mean Sq
## chemical    3  12.95    4.32
## bolt        4 157.00   39.25
## chemical:bolt 12  21.80    1.82
```

Since $n=1$ (single replicate) ; $MSE = SSE/ab(n-1)$, MSE can't be calculated! We can assume no significant between chemical and bolt

We need to run Tukey1df to check if the interaction term is significant

```
library(daewr)
```

```
## Warning: package 'daewr' was built under R version 4.0.5
```

```
## Registered S3 method overwritten by 'DoE.base':
##   method          from
##   factorize.factor conf.design
```

```
tukey.interaction <- data.frame(strength, chemical, bolt)
Tukey1df(tukey.interaction)
```

```
## Source      df      SS      MS      F      Pr>F
## A           3    12.95    4.3167
## B           4    157      39.25
## Error       12    21.8     29.0667
## NonAdditivity 1    0.2081    0.2081    0.11    0.7508
## Residual    11    21.5919   1.9629
```

NonAdditivity is 0.7508, so interaction term is not significant

```
effect.mod <- aov(strength~chemical+bolt, data=effect)
summary.aov(effect.mod)
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## chemical    3  12.95    4.32    2.376    0.121
## bolt        4 157.00   39.25   21.606 2.06e-05 ***
## Residuals   12  21.80    1.82
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The Chemical has pvalue >0.05, so the null hypothesis that the chemical doesn't affect strength is accepted and the alternative hypothesis that the chemical affect strength is rejected

The bolt has pvalue <0.05, so the null hypothesis that the chemical doesn't affect strength is rejected and the alternative hypothesis that the chemical affect strength is accepted

b.ii. Is Bolt is an effective blocking factor?

Since it's pvalue is significant, the blocking factor could be effective. We need to check if the MSE increases or decreases without blocking factor. If it increases, then we know that Blocking factor is effective.

```
effect.1.mod <- aov(strength~chemical, data=effect)
summary.aov(effect.1.mod)
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## chemical    3  12.95    4.317    0.386    0.764
## Residuals   16 178.80   11.175
```

MSE increases significantly without bolt, so yes the bolt act as an effective blocking factor

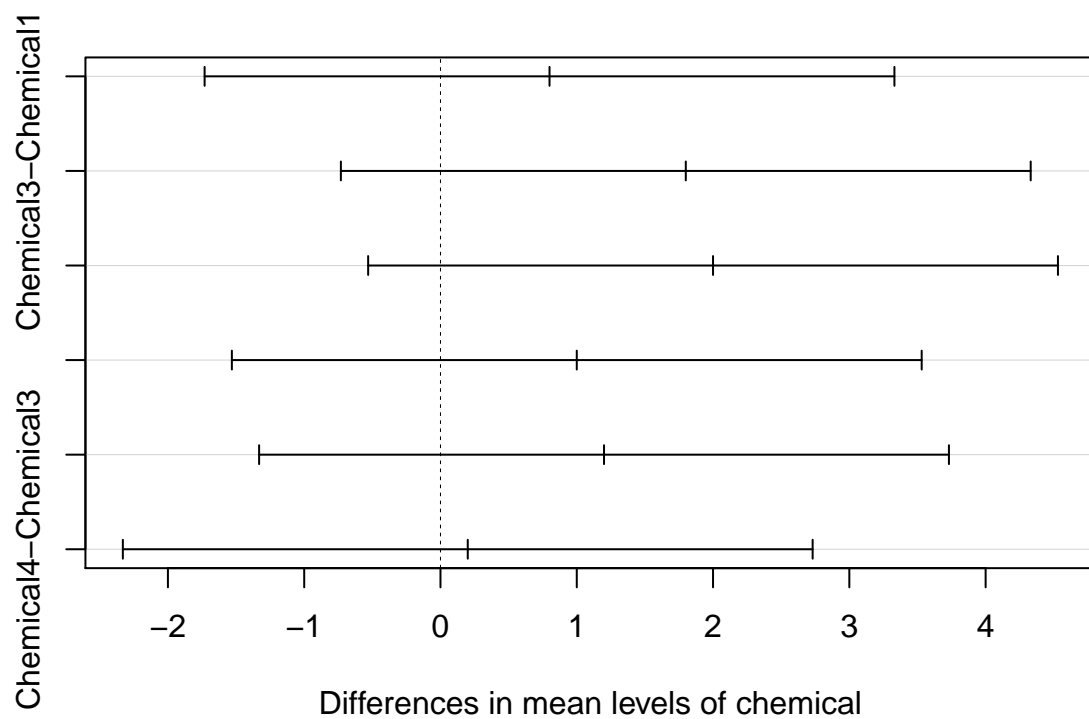
c. Perform a TukeyHSD to compare the treatment means. Which Chemical is the preferred(bring the highest strength)

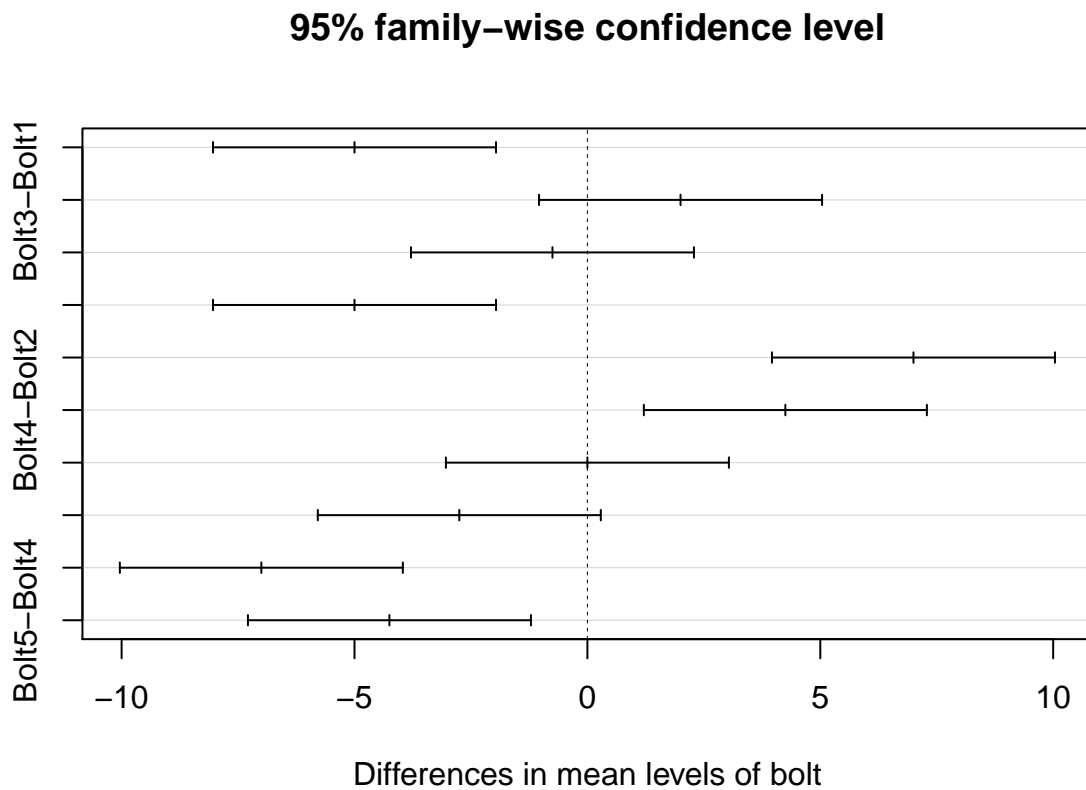
```
tukey <- TukeyHSD(effect.mod)
tukey
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = strength ~ chemical + bolt, data = effect)
##
## $chemical
##          diff          lwr          upr          p adj
## Chemical2-Chemical1  0.8 -1.7308322  3.330832  0.7852734
## Chemical3-Chemical1  1.8 -0.7308322  4.330832  0.2042593
## Chemical4-Chemical1  2.0 -0.5308322  4.530832  0.1417326
## Chemical3-Chemical2  1.0 -1.5308322  3.530832  0.6540138
## Chemical4-Chemical2  1.2 -1.3308322  3.730832  0.5182726
## Chemical4-Chemical3  0.2 -2.3308322  2.730832  0.9952030
##
## $bolt
##          diff          lwr          upr          p adj
## Bolt2-Bolt1 -5.00 -8.037831 -1.9621691  0.0015656
## Bolt3-Bolt1  2.00 -1.037831  5.0378309  0.2814173
## Bolt4-Bolt1 -0.75 -3.787831  2.2878309  0.9295872
## Bolt5-Bolt1 -5.00 -8.037831 -1.9621691  0.0015656
## Bolt3-Bolt2  7.00  3.962169 10.0378309  0.0000717
## Bolt4-Bolt2  4.25  1.212169  7.2878309  0.0056966
## Bolt5-Bolt2  0.00 -3.037831  3.0378309  1.0000000
## Bolt4-Bolt3 -2.75 -5.787831  0.2878309  0.0830636
## Bolt5-Bolt3 -7.00 -10.037831 -3.9621691  0.0000717
## Bolt5-Bolt4 -4.25 -7.287831 -1.2121691  0.0056966
```

```
plot(tukey)
```


95% family-wise confidence level





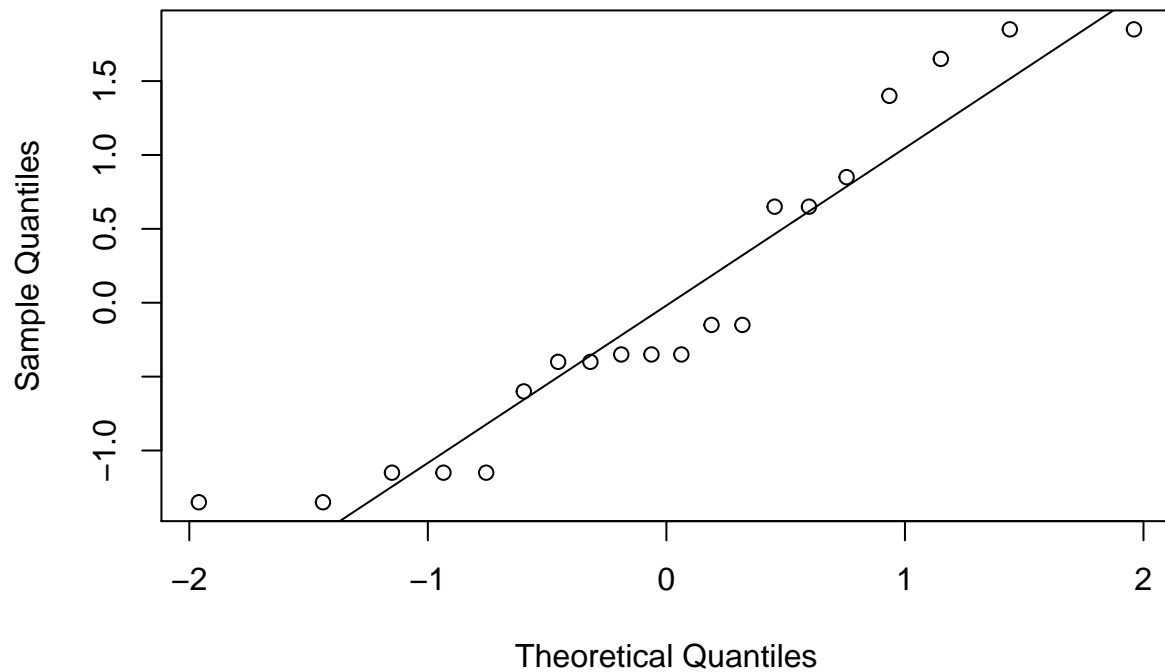
Chemical difference of mean is not significant, so it's clear chemical doesn't affect the strength. Some of the bolt values have pvalue < 0.05 , so it's clearly affect the strength.

d. Check the assumptions of the residuals.

```
res <- residuals(effect.mod)

qqnorm(res)
qqline(res)
```

Normal Q-Q Plot



```
shapiro.test(res)
```

```
##  
##  Shapiro-Wilk normality test  
##  
## data:  res  
## W = 0.8996, p-value = 0.04054
```

```
bartlett.test(res~bolt)
```

```
##  
##  Bartlett test of homogeneity of variances  
##  
## data:  res by bolt  
## Bartlett's K-squared = 0.65699, df = 4, p-value = 0.9565
```

```
bartlett.test(res~chemical)
```

```
##  
##  Bartlett test of homogeneity of variances  
##  
## data:  res by chemical  
## Bartlett's K-squared = 2.6757, df = 3, p-value = 0.4444
```

The data doesn't appear to be normal, but it has const variance.

Question 2

a. set up the data frame named `eye`, `subject`, `distance` as factors and `time` as response

```
setup_q_2 <-function()
{
  subject.data <-rep(c(1,2,3,4,5), time=4)
  distance.data <- rep(c(4,6,8,10), each=5)
  time <-c(10,6,6,6,6,7,6,6,1,6,5,3,3,2,5,6,4,4,2,3)
  eye <- data.frame(time, subject.data, distance.data)

  eye$subject <- factor(subject.data, c(1,2,3,4,5), label=c("Sub1","Sub2", "Sub3","Sub4", "Sub5"))

  eye$distance <- factor(distance.data, c(4,6,8,10), label=c("Dist4","Dist6", "Dist8","Dist10"))
  return (eye)
}
eye <- setup_q_2()
head(eye)
```

```
##   time subject.data distance.data subject distance
## 1   10           1           4     Sub1     Dist4
## 2    6           2           4     Sub2     Dist4
## 3    6           3           4     Sub3     Dist4
## 4    6           4           4     Sub4     Dist4
## 5    6           5           4     Sub5     Dist4
## 6    7           1           6     Sub1     Dist6
```

```
str(eye)
```

```
## 'data.frame':   20 obs. of  5 variables:
##  $ time          : num  10 6 6 6 6 7 6 6 1 6 ...
##  $ subject.data  : num   1 2 3 4 5 1 2 3 4 5 ...
##  $ distance.data : num   4 4 4 4 4 6 6 6 6 6 ...
##  $ subject       : Factor w/ 5 levels "Sub1","Sub2",...: 1 2 3 4 5 1 2 3 4 5 ...
##  $ distance      : Factor w/ 4 levels "Dist4","Dist6",...: 1 1 1 1 1 2 2 2 2 2 ...
```

b. build a regression model named `eye.mod`. Is “subject” an effecting blocking factor?

```
attach(eye)
eye.mod <- aov(time~subject*distance, data=effect)
summary.aov(eye.mod)
```

```
##              Df Sum Sq Mean Sq
## subject       4  36.30   9.075
## distance      3  32.95  10.983
## subject:distance 12  15.30   1.275
```

Since $n=1$, so there are no p-values. We need to check it using Tukeys1df

```
library(daewr)
```

```
## Warning: package 'daewr' was built under R version 4.0.5
```

```
tukey.interaction <- data.frame(time, subject, distance)
Tukey1df(tukey.interaction)
```

## Source	df	SS	MS	F	Pr>F
## A	4	36.3	9.075		
## B	3	32.95	10.9833		
## Error	12	15.3	11.475		
## NonAdditivity	1	0.2749	0.2749	0.2	0.6624
## Residual	11	15.0251	1.3659		

NonAdditivity is 0.6624, so interaction is not significant.

```
eye.mod <- aov(time~subject+distance, data=effect)
summary.aov(eye.mod)
```

##	Df	Sum Sq	Mean Sq	F value	Pr(>F)			
## subject	4	36.30	9.075	7.118	0.00355 **			
## distance	3	32.95	10.983	8.614	0.00254 **			
## Residuals	12	15.30	1.275					
## ---								
## Signif. codes:	0	'***'	0.001	'**'	0.01	'*' 0.05	'.' 0.1	' ' 1

It's evident that both subject and distance are both significant.

```
eye.distance.mod <- aov(time~distance, data=effect)
summary.aov(eye.distance.mod)
```

##	Df	Sum Sq	Mean Sq	F value	Pr(>F)			
## distance	3	32.95	10.983	3.406	0.0434 *			
## Residuals	16	51.60	3.225					
## ---								
## Signif. codes:	0	'***'	0.001	'**'	0.01	'*' 0.05	'.' 0.1	' ' 1

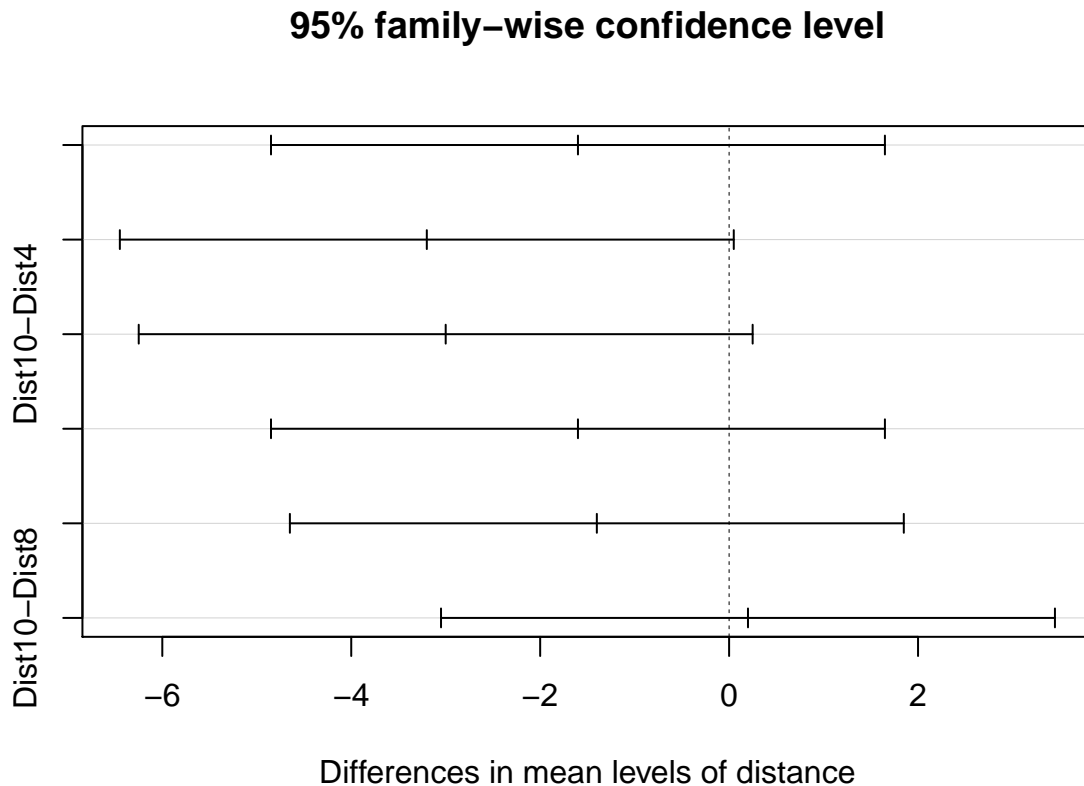
Blocking factor decreases the MSE value, so it is effective

c. Perform TukeyHSD on “Distance”. Which Distances bring the longest/ shortest focus time

```
tukey.distance <- TukeyHSD(eye.distance.mod)
tukey.distance
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = time ~ distance, data = effect)
##
## $distance
##      diff      lwr      upr    p adj
## Dist6-Dist4 -1.6 -4.849494 1.64949387 0.5121274
## Dist8-Dist4 -3.2 -6.449494 0.04949387 0.0543435
## Dist10-Dist4 -3.0 -6.249494 0.24949387 0.0756883
## Dist8-Dist6 -1.6 -4.849494 1.64949387 0.5121274
## Dist10-Dist6 -1.4 -4.649494 1.84949387 0.6161363
## Dist10-Dist8 0.2 -3.049494 3.44949387 0.9979766
```

```
plot(tukey.distance)
```



d. Calculate the sample size for power > .90, use `power.anova.test()`.

```
MSE <-1.275
trt.means <-tapply(time, list(subject, distance), mean)
power.anova.test(groups=5, n=2, between.var = var(trt.means), within.var = MSE,
  sig.level = 0.05)
```

```
##
```

```
##      Balanced one-way analysis of variance power calculation
##
##      groups = 5
##      n = 2
##      between.var = 3.20, 1.80, 1.40, 2.20, 1.80, 5.70, 2.35, 2.80, 1.40, 2.35, 1.80, 1.15, 2.20, 2.80
##      within.var = 1.275
##      sig.level = 0.05
##      power = 0.6209504, 0.3863123, 0.3087149, 0.4602038, 0.3863123, 0.8639282, 0.4866474, 0.561
##
## NOTE: n is number in each group
```

Question 3

a. Setup a dataframe manually to use the Latin square design

```
setup_q_3<- function ()
{
operator <- factor(rep(c(1,2,3,4), time=4), level=c(1,2,3,4),
                    label=c("Op1", "Op2", "Op3", "Op4"))
order <- factor(rep(c(1,2,3,4), each=4), level=c(1,2,3,4),
                 label=c("Order1","Order2","Order3","Order4" ))
r1 <- c('C','D','A','B')
r2 <- c('B','C','D','A')
r3 <- c('A','B','C','D')
r4 <- c('D','A','B','C')
treat <- c(r1,r2,r3,r4)
time <- c(10,14,7,8,7,18,11,8,5,10,11,9,10,10,12,14)

experiment <-data.frame(operator, order, treat, time)
return (experiment)
}
experiment <- setup_q_3()
str(experiment)
```

```
## 'data.frame': 16 obs. of 4 variables:
## $ operator: Factor w/ 4 levels "Op1","Op2","Op3",...: 1 2 3 4 1 2 3 4 1 2 ...
## $ order : Factor w/ 4 levels "Order1","Order2",...: 1 1 1 1 2 2 2 2 3 3 ...
## $ treat : chr "C" "D" "A" "B" ...
## $ time : num 10 14 7 8 7 18 11 8 5 10 ...
```

```
head(experiment)
```

```
## operator order treat time
## 1      Op1 Order1    C   10
## 2      Op2 Order1    D   14
## 3      Op3 Order1    A    7
## 4      Op4 Order1    B    8
## 5      Op1 Order2    B    7
## 6      Op2 Order2    C   18
```

```
attach(experiment)
```

```
## The following object is masked from eye:
##
##      time
```

b. Build a regression model, using `aov()`. Do the treatment affect the assembly time?

```
time.aov <- aov(time~treat+order+operator)
summary(time.aov)
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## treat      3   72.5   24.167   13.810 0.00421 **
## order      3   18.5    6.167    3.524 0.08852 .
## operator   3   51.5   17.167    9.810 0.00993 **
## Residuals  6   10.5    1.750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The pvalue for treatment is significant, so it does affect the assembly time.

c. Find the lowest assembly time.

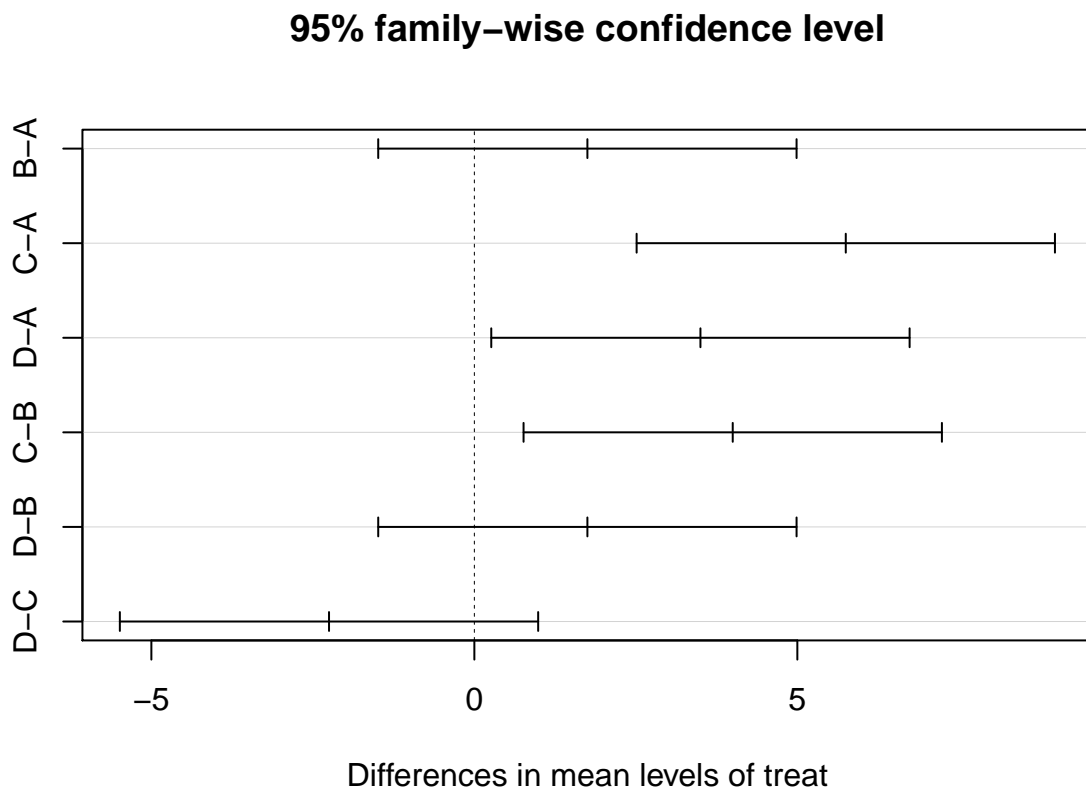
```
tukey <- TukeyHSD(time.aov)
tukey
```

```
##      Tukey multiple comparisons of means
##      95% family-wise confidence level
##
## Fit: aov(formula = time ~ treat + order + operator)
##
## $treat
##      diff      lwr      upr      p adj
## B-A   1.75 -1.4881345 4.9881345 0.3304308
## C-A   5.75  2.5118655 8.9881345 0.0034505
## D-A   3.50  0.2618655 6.7381345 0.0363534
## C-B   4.00  0.7618655 7.2381345 0.0202927
## D-B   1.75 -1.4881345 4.9881345 0.3304308
## D-C  -2.25 -5.4881345 0.9881345 0.1761447
##
## $order
##      diff      lwr      upr      p adj
## Order2-Order1  1.25 -1.9881345 4.4881345 0.5756823
## Order3-Order1 -1.00 -4.2381345 2.2381345 0.7191024
## Order4-Order1  1.75 -1.4881345 4.9881345 0.3304308
## Order3-Order2 -2.25 -5.4881345 0.9881345 0.1761447
## Order4-Order2  0.50 -2.7381345 3.7381345 0.9474067
## Order4-Order3  2.75 -0.4881345 5.9881345 0.0924529
```

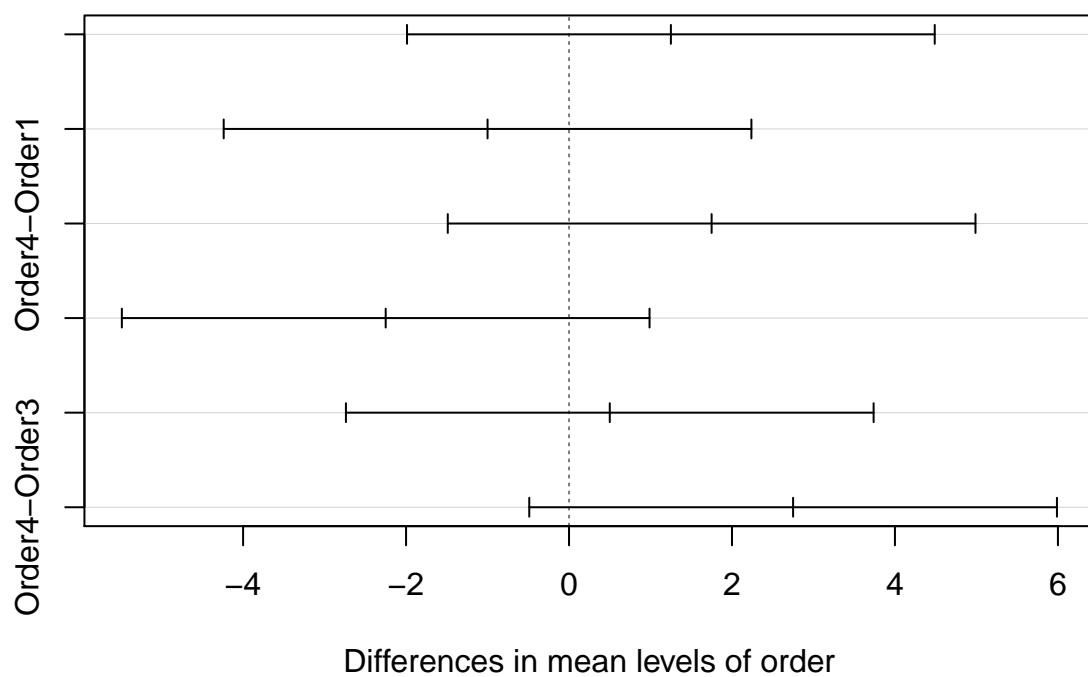


```
##
## $operator
##      diff      lwr      upr    p adj
## Op2-Op1  5.00  1.7618655  8.23813449 0.0070204
## Op3-Op1  2.25 -0.9881345  5.48813449 0.1761447
## Op4-Op1  1.75 -1.4881345  4.98813449 0.3304308
## Op3-Op2 -2.75 -5.9881345  0.48813449 0.0924529
## Op4-Op2 -3.25 -6.4881345 -0.01186551 0.0492740
## Op4-Op3 -0.50 -3.7381345  2.73813449 0.9474067
```

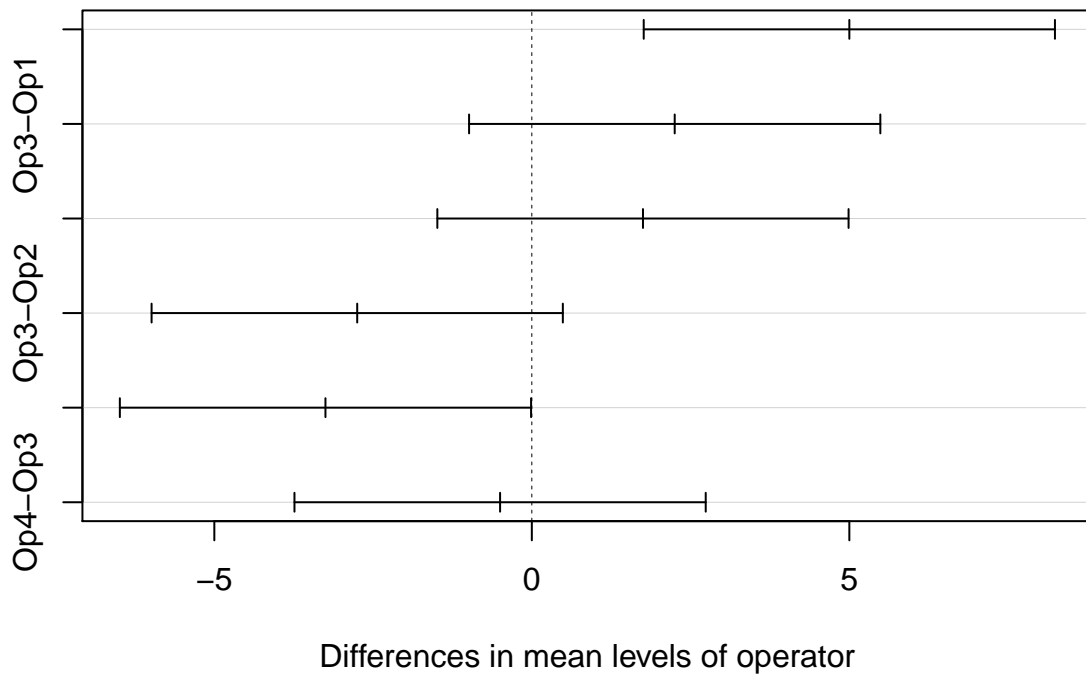
```
plot(tukey)
```



95% family-wise confidence level



95% family-wise confidence level



A has smallest assembly time
