

# Stats 101B Homework #6

Anna Piskun (505109399) - Discussion 1A

6/2/2020

## Question 1

Q1) A textile mill has a large number of looms. Each loom is supposed to provide the same output of cloth per minute. To investigate this assumption, five looms are chosen at random, and their output is noted at different times. The following data are obtained:

```
data.1 <- read.csv("HW6 S2020 Q1 Data.csv")
model <- aov(Output~factor(Loom), data = data.1)
summary(model)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## factor(Loom)   4  0.3416   0.0854     5.77 0.00296 **
## Residuals     20  0.2960   0.0148
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(a) Explain why this is a random effects experiment. Are the looms equal in output? Use ( $\alpha = 0.05$ ).

Given the above information, this is a random effects experiment since all of the looms (aka our treatment factor) used in the experiment are chosen at random to form a random sample of all looms in the manufacturing area. Looking at our anova analysis, the pvalue of 0.00296 is statistically significant. Since the pvalue is less than 0.05, we reject the null hypothesis that there is no difference in loom output. Thus, we find that the looms are NOT equal in output.

(b) Estimate the variability between looms.

```
a <- (anova(aov(Output~factor(Loom), data = data.1))[,3])
#the variability between looms
(a[1]-a[2])/5
```

```
## [1] 0.01412
```

(c) Estimate the experimental error variance.

```
#experimental error variance is just the MSerror
(anova(aov(Output~factor(Loom), data = data.1))[2,3])
```

```
## [1] 0.0148
```

(d) Find a 95 percent confidence interval for the ICC

```
#using the anova output we can calculate the confidence interval by hand
```

```
#Lower Bound
```

```
f.low <- qf(0.025,4,20, lower.tail = FALSE)
```

```
lb <- (1/5)*((0.0854/0.0148)*(1/f.low) - 1)
lb/(lb+1)

## [1] 0.113751
#Upper Bound

f.upper <- qf(0.975, 4, 20, lower.tail = FALSE)

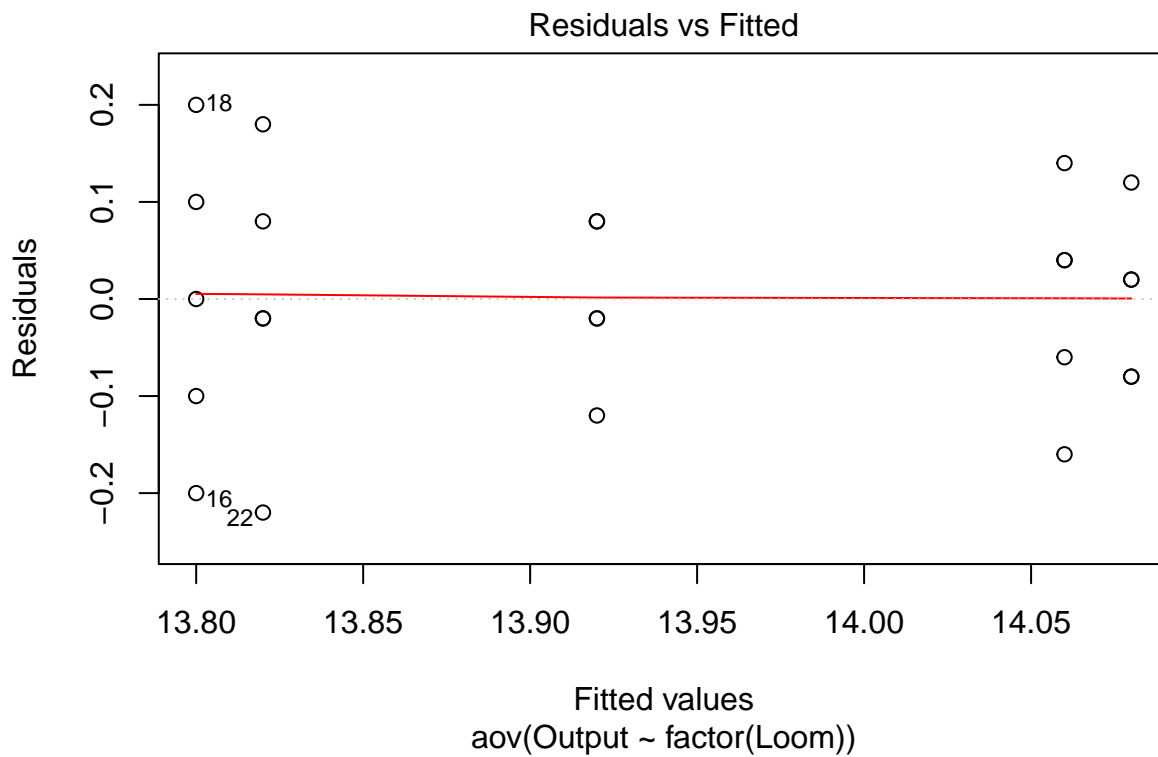
ub <- (1/5)*((0.0854/0.0148)*(1/f.upper) - 1)
ub/(ub+1)

## [1] 0.9063551
```

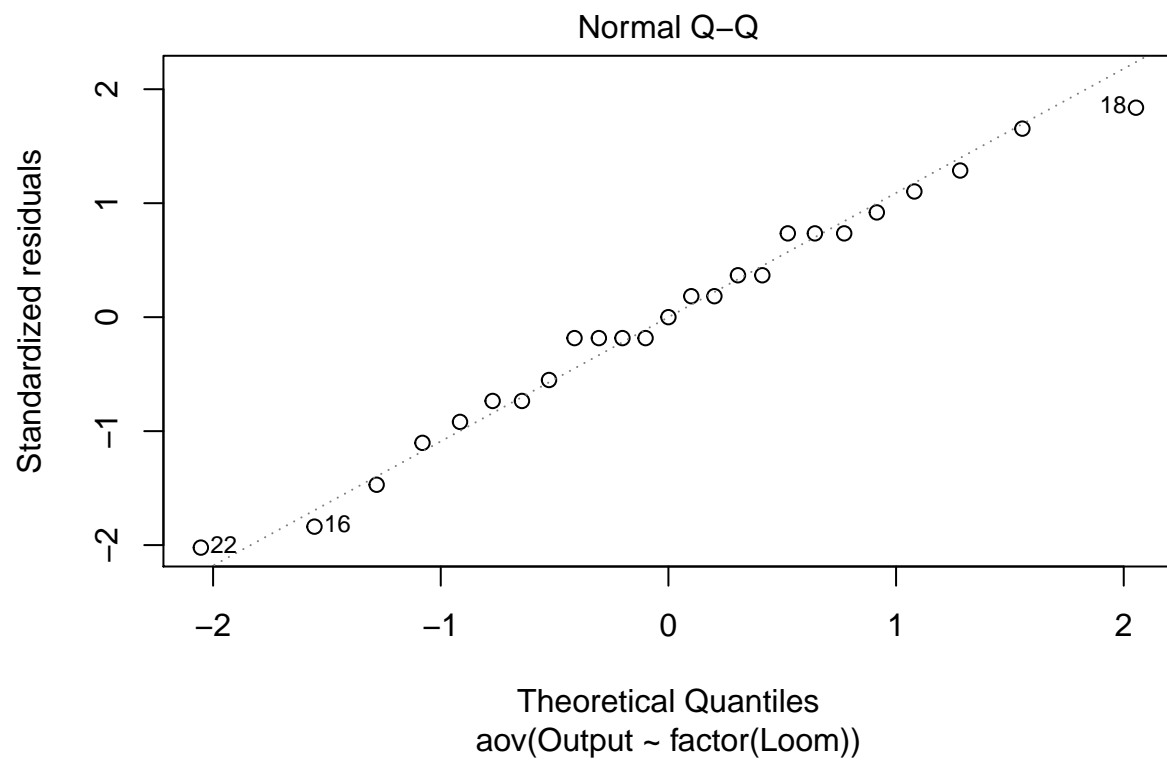
Therefore, a 95% CI is [0.114, 0.9064].

- (e) Analyze the residuals from this experiment. Do you think that the analysis of variance assumptions are satisfied?

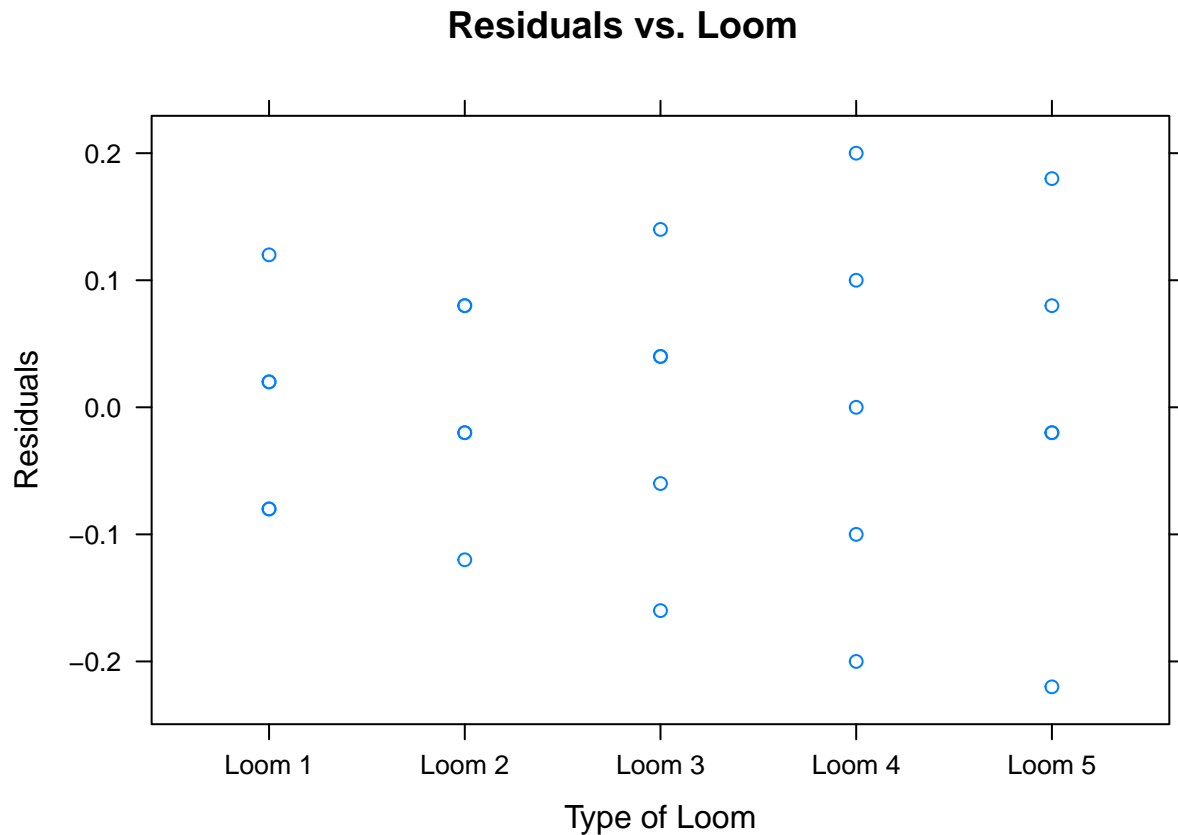
```
plot(model, which = 1:2)
```



```
#Residuals vs. Loom
library(lattice)
```



```
df1 <- data.frame(x = as.factor(data.1$Loom), y = model$residuals)
xyplot(y~x, data = df1,
       xlab = "Type of Loom",
       ylab = "Residuals",
       main = "Residuals vs. Loom")
```



Looking at the residuals vs. fitted values plot we see evidence of a slight fanshape which may indicate non constant variance. Specifically looking at the residuals vs. loom plot, we that there are about equal points above and below 0 at each loom which indicates good model fit, however, the fluctuations of the residuals themselves within each factor are not similar across all 5 levels indicating that the constant variance assumption is not satisfied. Likewise, analyzing the normal probability plot it is quite curvy and there is some deviation from a straight line indicating that the errors are not normally distributed and the normality assumption is not satisfied.

- (f) Use the REML method to analyze this data. Compare the 95 percent confidence interval on the error variance from REML with the exact chi-square confidence interval.

```
library(lme4)

## Loading required package: Matrix

m2 <- lmer(Output ~ (1|Loom), data=data.1)
m2reduced <- lm(Output~factor(Loom), data = data.1)
summary(m2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Output ~ (1 | Loom)
## Data: data.1
##
## REML criterion at convergence: -22.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.9736 -0.4525 -0.1872  0.6283  1.4503
##
```

```
## Random effects:
## Groups Name Variance Std.Dev.
## Loom (Intercept) 0.01412 0.1188
## Residual 0.01480 0.1217
## Number of obs: 25, groups: Loom, 5
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 13.93600 0.05845 238.4
```

```
logLik(m2reduced)
```

```
## 'log Lik.' 19.97993 (df=6)
```

```
logLik(m2)
```

```
## 'log Lik.' 11.38814 (df=3)
```

```
chisquare.test <- 2*(19.97993-11.38814)
chisquare.test
```

```
## [1] 17.18358
```

```
pvalue <- 1- pchisq(17.18358,3)
pvalue
```

```
## [1] 0.0006478813
```

Using the log-likelihood ratio test to complete our chi-square distribution, we get a pvalue of 0.0006 which is less than 0.05 meaning that we reject the null hypothesis and conclude that the Looms are truly different from each other.

Q2) An engineer is interested in the effects of cutting speed (A), tool geometry (B), and cutting angle (C) on the life (in hours) of a machine tool. Two levels of each factor are chosen, and three replicates of a 23 factorial design are run. The results are as follows:

```
data.2 <- read.csv("HW6 S2020 Q2 Data.csv")
```

(a) Estimate the factor effects. Which effects appear to be large?

```
summary(lm(Life.Hours~Cutting.Speed*Tool.Geometry*Cutting.Angle, data = data.2))
```

```
##
## Call:
## lm(formula = Life.Hours ~ Cutting.Speed * Tool.Geometry * Cutting.Angle,
## data = data.2)
##
## Residuals:
## Min 1Q Median 3Q Max
## -5.667 -3.500 -1.167 3.167 10.333
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 40.8333 1.1211 36.421 < 2e-16
## Cutting.Speed 0.1667 1.1211 0.149 0.883680
## Tool.Geometry 5.6667 1.1211 5.054 0.000117
## Cutting.Angle 3.4167 1.1211 3.048 0.007679
## Cutting.Speed:Tool.Geometry -0.8333 1.1211 -0.743 0.468078
## Cutting.Speed:Cutting.Angle -4.4167 1.1211 -3.939 0.001172
```

```
## Tool.Geometry:Cutting.Angle          -1.4167      1.1211  -1.264  0.224475
## Cutting.Speed:Tool.Geometry:Cutting.Angle -1.0833      1.1211  -0.966  0.348282
##
## (Intercept)                        ***
## Cutting.Speed
## Tool.Geometry                      ***
## Cutting.Angle                      **
## Cutting.Speed:Tool.Geometry
## Cutting.Speed:Cutting.Angle        **
## Tool.Geometry:Cutting.Angle
## Cutting.Speed:Tool.Geometry:Cutting.Angle
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.492 on 16 degrees of freedom
## Multiple R-squared:  0.7696, Adjusted R-squared:  0.6689
## F-statistic: 7.637 on 7 and 16 DF,  p-value: 0.0003977
```

The effects that appear to be large are Tool.Geometry (B), Cutting Angle (C), and the interaction between Cutting Speed (A) and Cutting Angle (C).

(b) Use the analysis of variance to confirm your conclusions for part (a).

```
summary(aov(Life.Hours~Cutting.Speed*Tool.Geometry*Cutting.Angle, data = data.2))
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Cutting.Speed      1      0.7      0.7    0.022 0.883680
## Tool.Geometry       1    770.7    770.7   25.547 0.000117
## Cutting.Angle       1    280.2    280.2    9.287 0.007679
## Cutting.Speed:Tool.Geometry      1     16.7     16.7    0.552 0.468078
## Cutting.Speed:Cutting.Angle      1    468.2    468.2   15.519 0.001172
## Tool.Geometry:Cutting.Angle       1     48.2     48.2    1.597 0.224475
## Cutting.Speed:Tool.Geometry:Cutting.Angle  1     28.2     28.2    0.934 0.348282
## Residuals          16    482.7     30.2
##
## Cutting.Speed
## Tool.Geometry      ***
## Cutting.Angle      **
## Cutting.Speed:Tool.Geometry
## Cutting.Speed:Cutting.Angle      **
## Tool.Geometry:Cutting.Angle
## Cutting.Speed:Tool.Geometry:Cutting.Angle
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looking at our anova table above, we see that Tool.Geometry (B), Cutting.Angle (C), and the interaction between Cutting.Speed and Cutting.Angle (AC) are all statistically significant (less than 0.05), which confirms our conclusions for part a.

(c) Write down a regression model for predicting tool life (in hours) based on the results of this experiment.

Using our above analysis we can write a regression model for predicting tool life only using the statistically significant factors. Thus we get the following model:

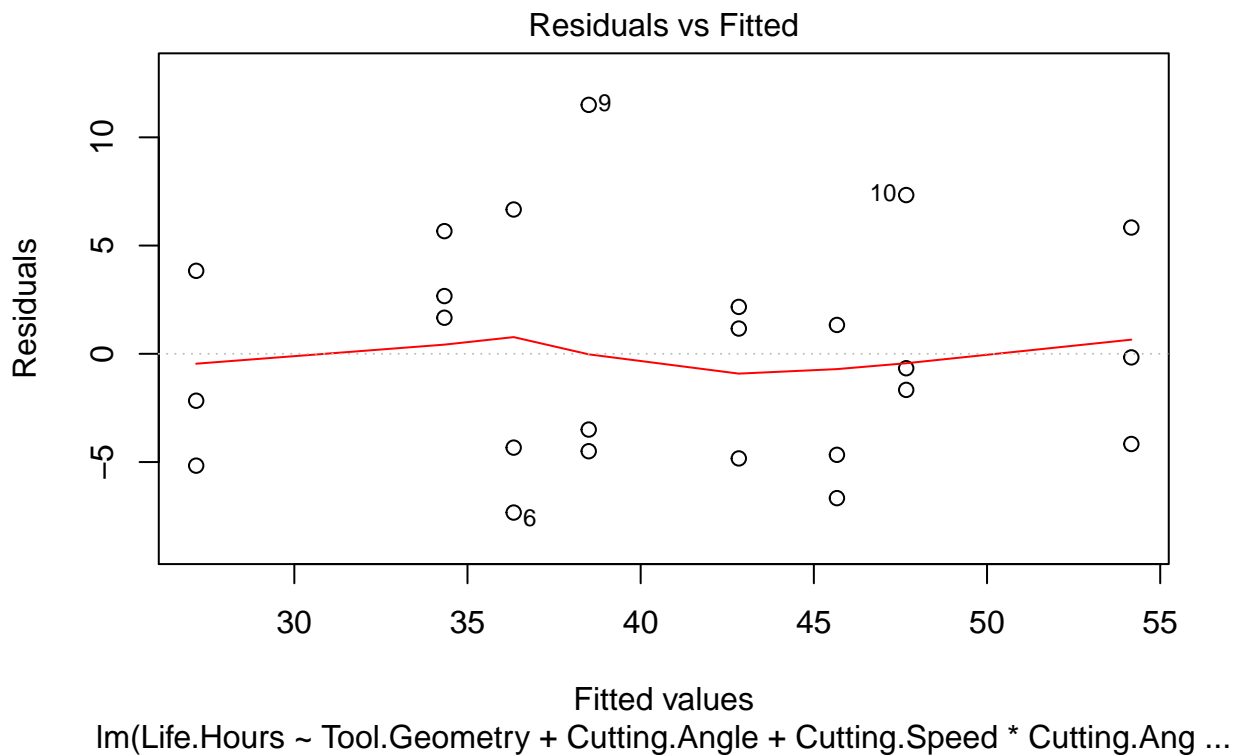
$$Y = 40.833 + 5.6667(B) + 3.4167(C) - 4.4167(AC)$$

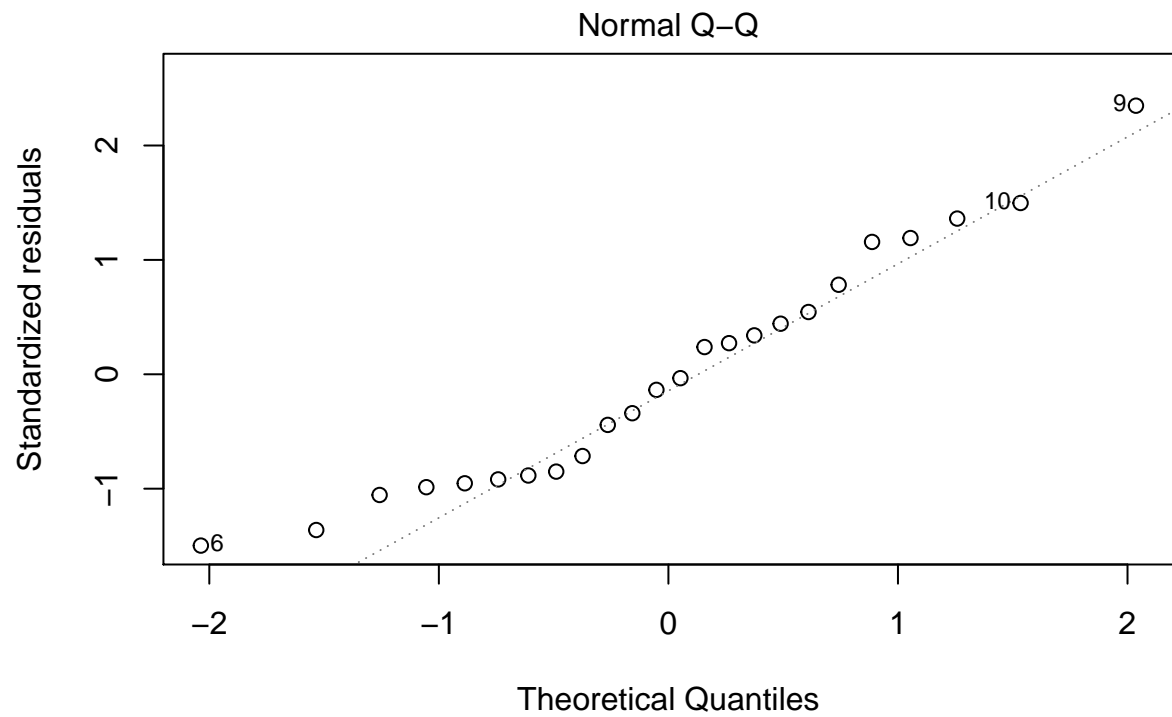
```
model.2 <- lm(Life.Hours~Tool.Geometry+Cutting.Angle+Cutting.Speed*Cutting.Angle, data = data.2)
summary(model.2)
```

```
##
## Call:
## lm(formula = Life.Hours ~ Tool.Geometry + Cutting.Angle + Cutting.Speed *
##     Cutting.Angle, data = data.2)
##
## 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)      40.8333     1.1236  36.342 < 2e-16 ***
## Tool.Geometry       5.6667     1.1236   5.043 7.22e-05 ***
## Cutting.Angle       3.4167     1.1236   3.041 0.006724 **
## Cutting.Speed       0.1667     1.1236   0.148 0.883641
## Cutting.Angle:Cutting.Speed -4.4167     1.1236  -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
```

(d) Analyze the residuals. Are there any obvious problems?

```
plot(model.2, which = 1:2)
```

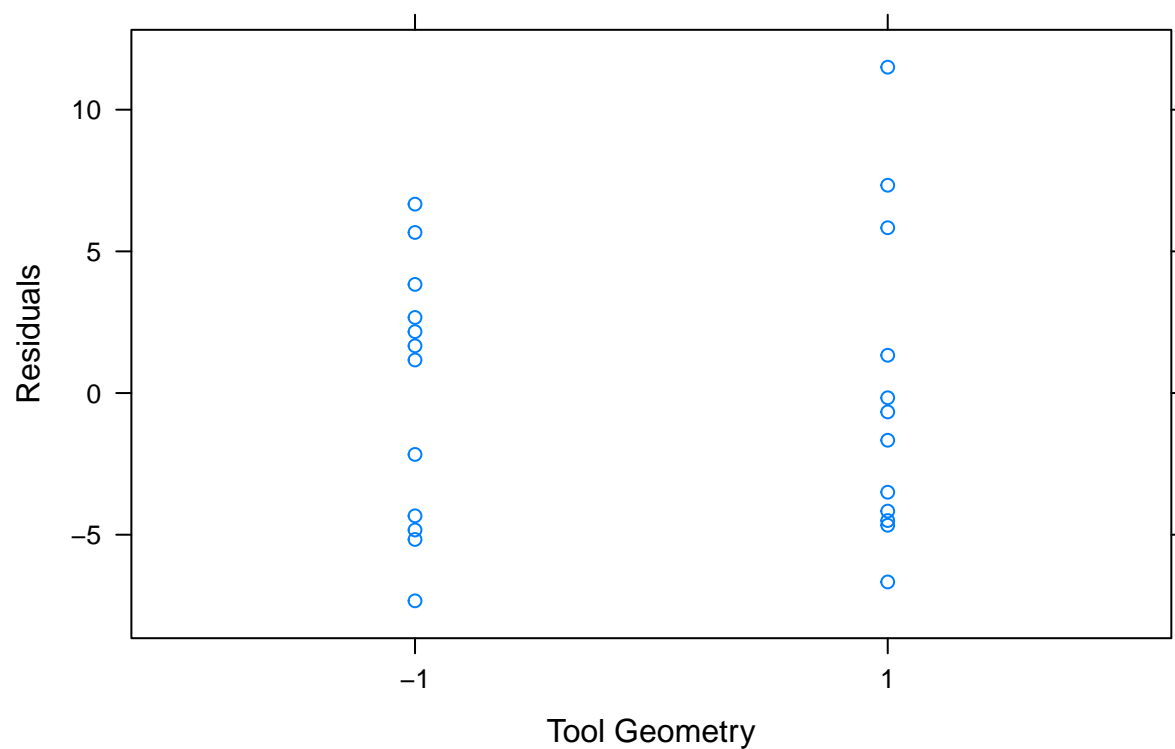




```
library(lattice)
df2 <- data.frame(x = as.factor(data.2$Tool.Geometry), y = model.2$residuals)
xyplot(y~x, data = df2,
       xlab = "Tool Geometry",
       ylab = "Residuals",
       main = "Residuals vs. Tool Geometry")
```

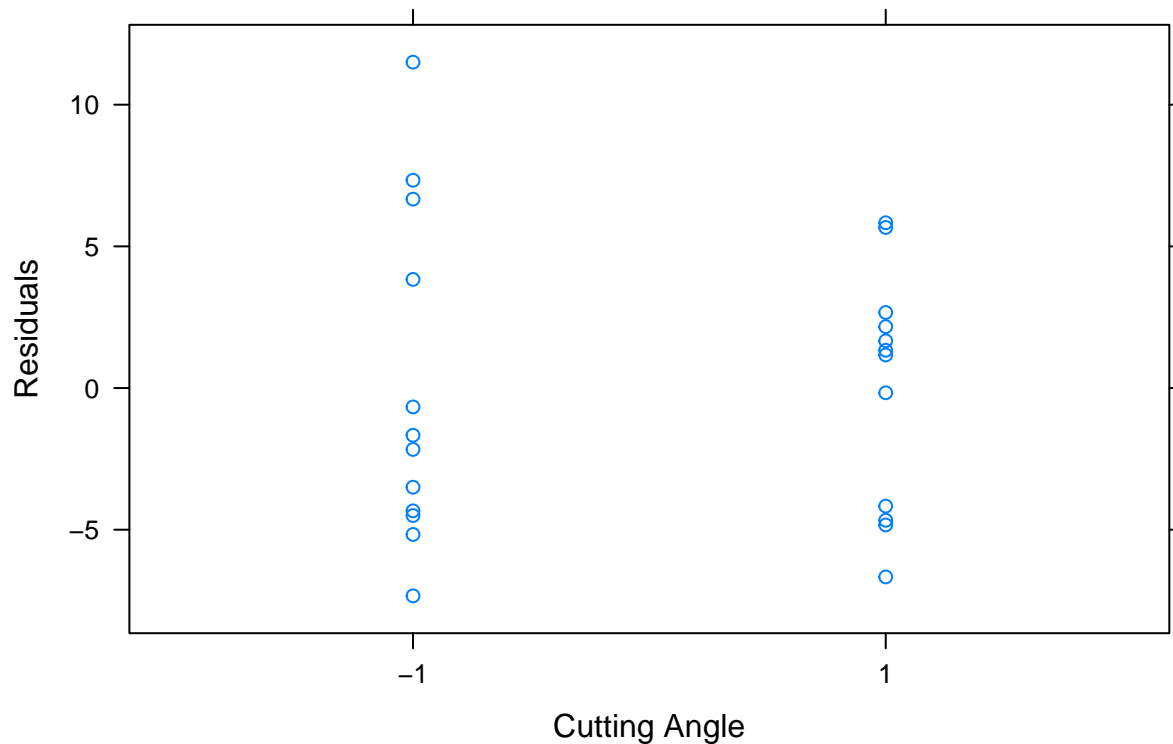


## Residuals vs. Tool Geometry



```
df3 <- data.frame(x = as.factor(data.2$Cutting.Angle), y = model.2$residuals)
xyplot(y~x, data = df3,
       xlab = "Cutting Angle",
       ylab = "Residuals",
       main = "Residuals vs. Cutting Angle")
```

## Residuals vs. Cutting Angle

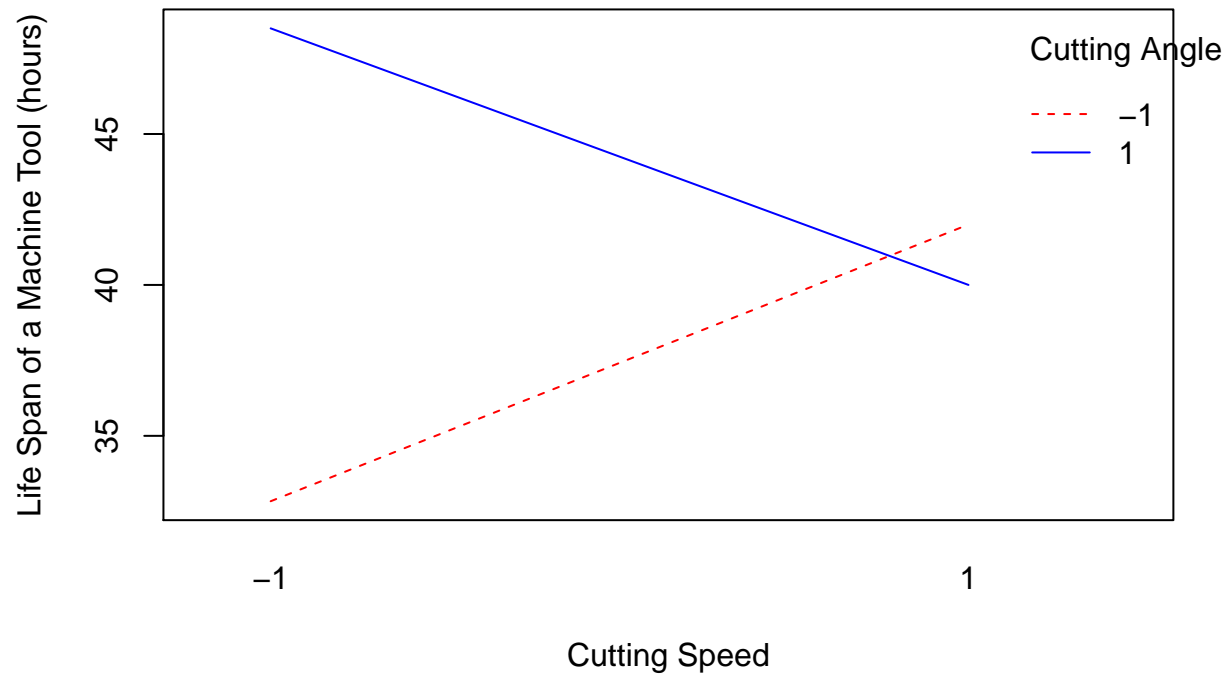


Looking at the above residual plots, there are no clear patterns or fanshapes and there are about equal points above and below 0 within the factor specific plots which indicates good model fit. Likewise, the fluctuations of the residuals themselves within each factor are similar across all levels indicating that the constant variance assumption is met. However, the normal probability plot is slightly curvy which may indicate that the errors are not normally distributed. Aside from that, there are no obvious problems with the model.

- (e) On the basis of an analysis of main effect and interaction plots, what coded factor levels of A, B, and C would you recommend using?

```
interaction.plot(x.factor = data.2$Cutting.Speed, trace.factor = data.2$Cutting.Angle,
               response = data.2$Life.Hours, fun = base::mean, trace.label = "Cutting Angle",
               col = c("red", "blue"),
               xlab = "Cutting Speed", ylab = "Life Span of a Machine Tool (hours)",
               main = "Interaction b/w Cutting Speed and Cutting Angle")
```

## Interaction b/w Cutting Speed and Cutting Angle

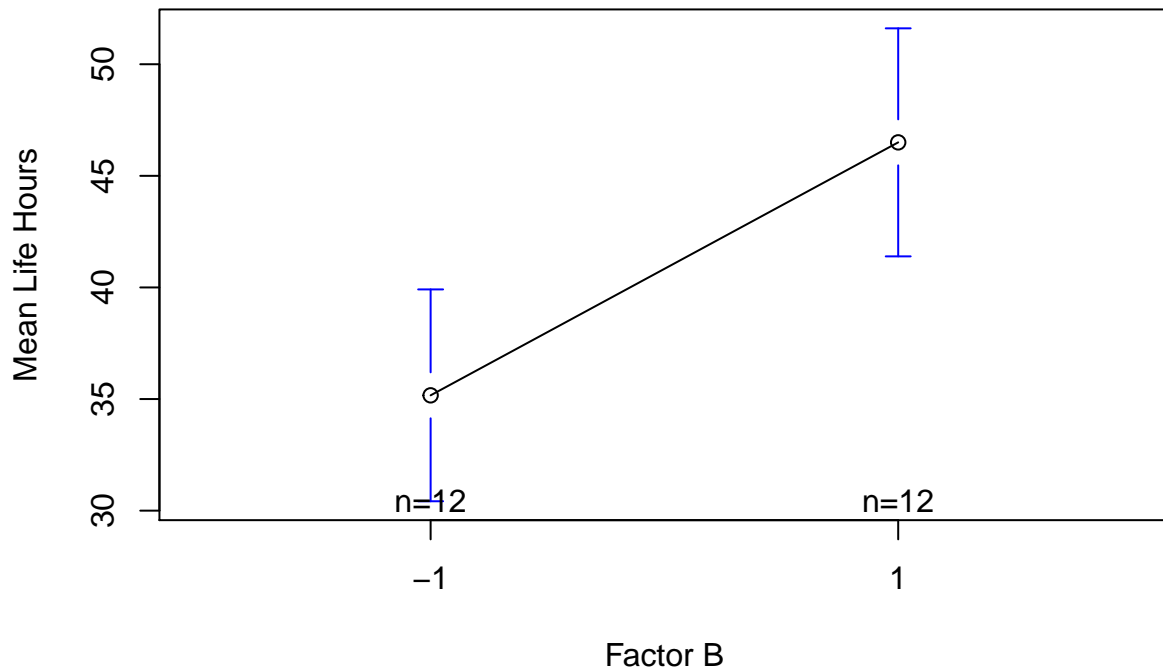


```
library(gplots)

##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##
##      lowess

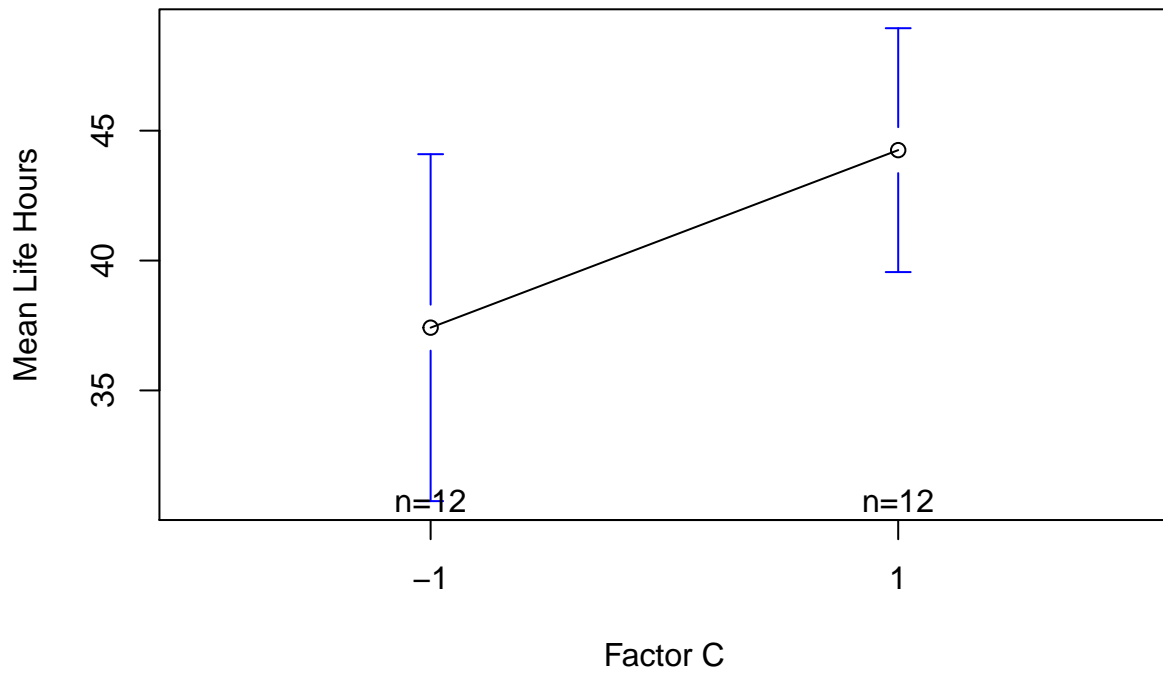
plotmeans(Life.Hours~Tool.Geometry, data = data.2,
          xlab="Factor B", ylab="Mean Life Hours",
          main = "Main Effect Plot for B")
```

**Main Effect Plot for B**



```
plotmeans(Life.Hours~Cutting.Angle, data = data.2,  
          xlab="Factor C",ylab="Mean Life Hours",  
          main = "Main Effect Plot for C")
```

**Main Effect Plot for C**



Using the above main effect plots and interaction plot, we see that mean tool life is highest when (A,B,C)

= (-, +, +). Thus I would recommend the coded factor levels of -1 for A, 1 for B, and 1 for C.

Q3) I am always interested in improving my golf scores. Since a typical golfer uses the putter for about 35–45 percent of his or her strokes, it seems reasonable that improving one’s putting is a logical and perhaps simple way to improve a golf score (“The man who can putt is a match for any man.”— Willie Parks, 1864–1925, two-time winner of the British Open). An experiment was conducted to study the effects of four factors on putting accuracy. The design factors are length of putt, type of putter, breaking putt versus straight putt, and level versus downhill putt. The response variable is distance from the ball to the center of the cup after the ball comes to rest. One golfer performs the experiment, a 24 factorial design with seven replicates was used, and all putts are made in random order. The results are shown in Table P6.4.

```
data.3 <- read.csv("HW6 S2020 Q3 Data.csv")
```

(a) Analyze the data from this experiment. Which factors significantly affect putting performance?

```
summary(aov(Distance.from.Cup~factor(Length.of.Putt)*factor(Type.of.Putter)*factor(Break.of.Putt)*factor(Slope.of.Putt)))
```

##	Df
## factor(Length.of.Putt)	1
## factor(Type.of.Putter)	1
## factor(Break.of.Putt)	1
## factor(Slope.of.Putt)	1
## factor(Length.of.Putt):factor(Type.of.Putter)	1
## factor(Length.of.Putt):factor(Break.of.Putt)	1
## factor(Type.of.Putter):factor(Break.of.Putt)	1
## factor(Length.of.Putt):factor(Slope.of.Putt)	1
## factor(Type.of.Putter):factor(Slope.of.Putt)	1
## factor(Break.of.Putt):factor(Slope.of.Putt)	1
## factor(Length.of.Putt):factor(Type.of.Putter):factor(Break.of.Putt)	1
## factor(Length.of.Putt):factor(Type.of.Putter):factor(Slope.of.Putt)	1
## factor(Length.of.Putt):factor(Break.of.Putt):factor(Slope.of.Putt)	1
## factor(Type.of.Putter):factor(Break.of.Putt):factor(Slope.of.Putt)	1
## factor(Length.of.Putt):factor(Type.of.Putter):factor(Break.of.Putt):factor(Slope.of.Putt)	1
## Residuals	96
##	Sum Sq
## factor(Length.of.Putt)	917
## factor(Type.of.Putter)	388
## factor(Break.of.Putt)	145
## factor(Slope.of.Putt)	1
## factor(Length.of.Putt):factor(Type.of.Putter)	219
## factor(Length.of.Putt):factor(Break.of.Putt)	12
## factor(Type.of.Putter):factor(Break.of.Putt)	115
## factor(Length.of.Putt):factor(Slope.of.Putt)	94
## factor(Type.of.Putter):factor(Slope.of.Putt)	56
## factor(Break.of.Putt):factor(Slope.of.Putt)	2
## factor(Length.of.Putt):factor(Type.of.Putter):factor(Break.of.Putt)	7
## factor(Length.of.Putt):factor(Type.of.Putter):factor(Slope.of.Putt)	113
## factor(Length.of.Putt):factor(Break.of.Putt):factor(Slope.of.Putt)	39
## factor(Type.of.Putter):factor(Break.of.Putt):factor(Slope.of.Putt)	34
## factor(Length.of.Putt):factor(Type.of.Putter):factor(Break.of.Putt):factor(Slope.of.Putt)	96
## Residuals	8316
##	Mean Sq
## factor(Length.of.Putt)	917.1
## factor(Type.of.Putter)	388.1
## factor(Break.of.Putt)	145.1
## factor(Slope.of.Putt)	1.4

```

## factor(Length.of.Put):factor(Type.of.Putter) 218.7
## factor(Length.of.Put):factor(Break.of.Put) 11.9
## factor(Type.of.Putter):factor(Break.of.Put) 115.0
## factor(Length.of.Put):factor(Slope.of.Put) 93.8
## factor(Type.of.Putter):factor(Slope.of.Put) 56.4
## factor(Break.of.Put):factor(Slope.of.Put) 1.6
## factor(Length.of.Put):factor(Type.of.Putter):factor(Break.of.Put) 7.3
## factor(Length.of.Put):factor(Type.of.Putter):factor(Slope.of.Put) 113.0
## factor(Length.of.Put):factor(Break.of.Put):factor(Slope.of.Put) 39.5
## factor(Type.of.Putter):factor(Break.of.Put):factor(Slope.of.Put) 33.8
## factor(Length.of.Put):factor(Type.of.Putter):factor(Break.of.Put):factor(Slope.of.Put) 95.6
## Residuals 86.6
## F value
## factor(Length.of.Put) 10.588
## factor(Type.of.Putter) 4.481
## factor(Break.of.Put) 1.676
## factor(Slope.of.Put) 0.016
## factor(Length.of.Put):factor(Type.of.Putter) 2.525
## factor(Length.of.Put):factor(Break.of.Put) 0.137
## factor(Type.of.Putter):factor(Break.of.Put) 1.328
## factor(Length.of.Put):factor(Slope.of.Put) 1.083
## factor(Type.of.Putter):factor(Slope.of.Put) 0.651
## factor(Break.of.Put):factor(Slope.of.Put) 0.019
## factor(Length.of.Put):factor(Type.of.Putter):factor(Break.of.Put) 0.084
## factor(Length.of.Put):factor(Type.of.Putter):factor(Slope.of.Put) 1.305
## factor(Length.of.Put):factor(Break.of.Put):factor(Slope.of.Put) 0.456
## factor(Type.of.Putter):factor(Break.of.Put):factor(Slope.of.Put) 0.390
## factor(Length.of.Put):factor(Type.of.Putter):factor(Break.of.Put):factor(Slope.of.Put) 1.104
## Residuals
## Pr(>F)
## factor(Length.of.Put) 0.00157
## factor(Type.of.Putter) 0.03686
## factor(Break.of.Put) 0.19862
## factor(Slope.of.Put) 0.89928
## factor(Length.of.Put):factor(Type.of.Putter) 0.11538
## factor(Length.of.Put):factor(Break.of.Put) 0.71178
## factor(Type.of.Putter):factor(Break.of.Put) 0.25205
## factor(Length.of.Put):factor(Slope.of.Put) 0.30066
## factor(Type.of.Putter):factor(Slope.of.Put) 0.42159
## factor(Break.of.Put):factor(Slope.of.Put) 0.89127
## factor(Length.of.Put):factor(Type.of.Putter):factor(Break.of.Put) 0.77294
## factor(Length.of.Put):factor(Type.of.Putter):factor(Slope.of.Put) 0.25623
## factor(Length.of.Put):factor(Break.of.Put):factor(Slope.of.Put) 0.50121
## factor(Type.of.Putter):factor(Break.of.Put):factor(Slope.of.Put) 0.53386
## factor(Length.of.Put):factor(Type.of.Putter):factor(Break.of.Put):factor(Slope.of.Put) 0.29599
## Residuals
##
## factor(Length.of.Put) **
## factor(Type.of.Putter) *
## factor(Break.of.Put)
## factor(Slope.of.Put)
## factor(Length.of.Put):factor(Type.of.Putter)
## factor(Length.of.Put):factor(Break.of.Put)
## factor(Type.of.Putter):factor(Break.of.Put)

```

```
## factor(Length.of.Putt):factor(Slope.of.Putt)
## factor(Type.of.Putter):factor(Slope.of.Putt)
## factor(Break.of.Putt):factor(Slope.of.Putt)
## factor(Length.of.Putt):factor(Type.of.Putter):factor(Break.of.Putt)
## factor(Length.of.Putt):factor(Type.of.Putter):factor(Slope.of.Putt)
## factor(Length.of.Putt):factor(Break.of.Putt):factor(Slope.of.Putt)
## factor(Type.of.Putter):factor(Break.of.Putt):factor(Slope.of.Putt)
## factor(Length.of.Putt):factor(Type.of.Putter):factor(Break.of.Putt):factor(Slope.of.Putt)
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

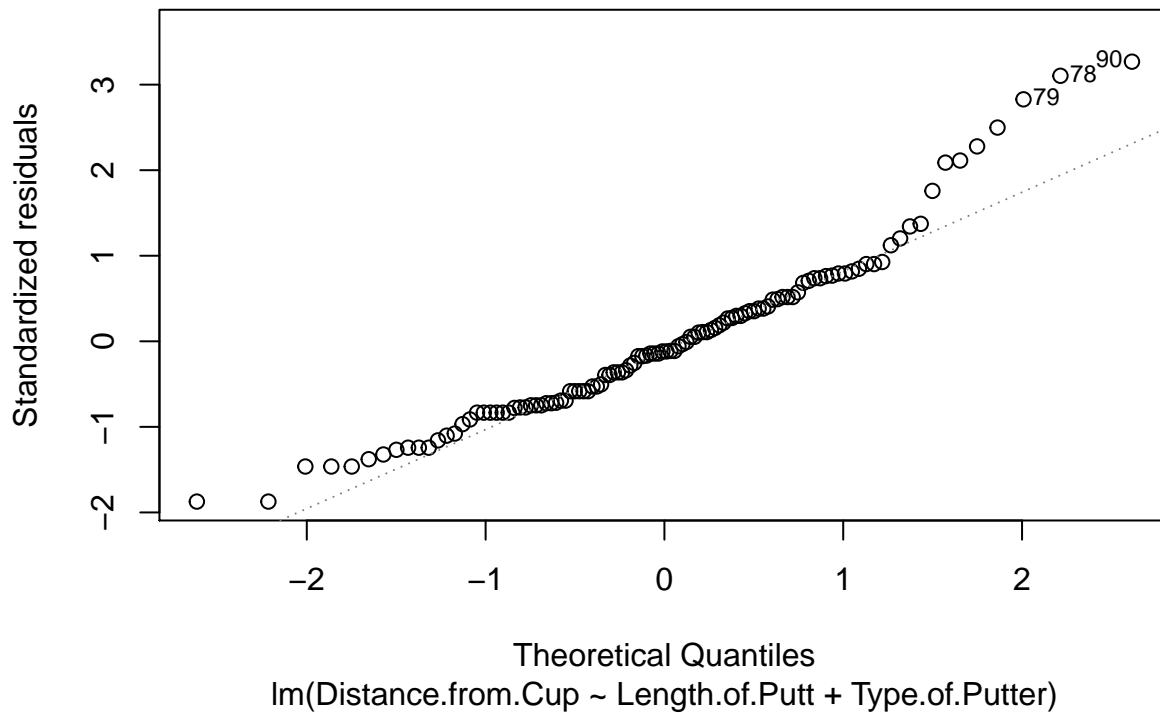
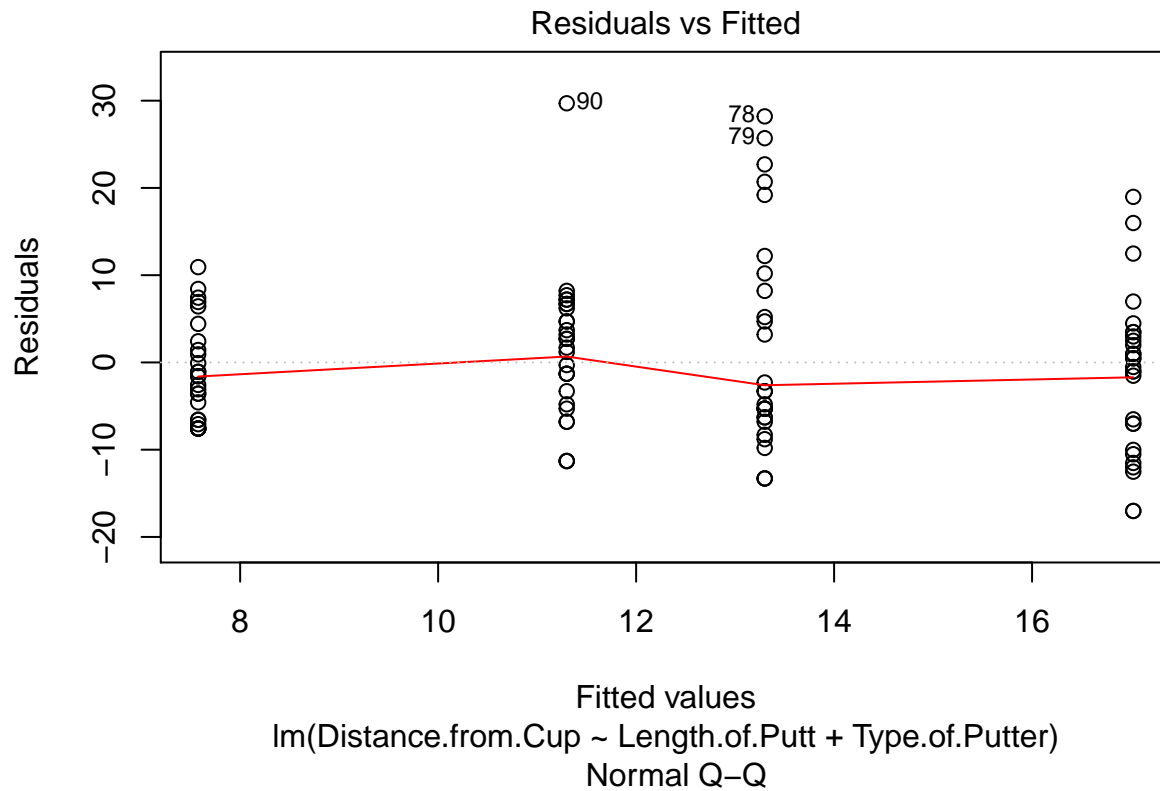
Using our analysis of variance, we find that only two factors, Length of Putt (A) and Type of Putter (B), significantly affect putting performance. Thus, we can create a model using only statistically significant factors which can be seen below:

```
model.3 <- lm(Distance.from.Cup~Length.of.Putt+Type.of.Putter, data = data.3)
summary(model.3)
```

```
##
## Call:
## lm(formula = Distance.from.Cup ~ Length.of.Putt + Type.of.Putter,
##     data = data.3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.022  -6.632  -1.076   4.701  29.701
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.71429     2.13206   2.211  0.02911 *
## Length.of.Putt    0.28616     0.08704   3.288  0.00136 **
## Type.of.PutterMallet 3.72321     1.74082   2.139  0.03469 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.212 on 109 degrees of freedom
## Multiple R-squared:  0.1237, Adjusted R-squared:  0.1076
## F-statistic: 7.692 on 2 and 109 DF,  p-value: 0.0007504
```

(b) Analyze the residuals from this experiment. Are there any indications of model inadequacy?

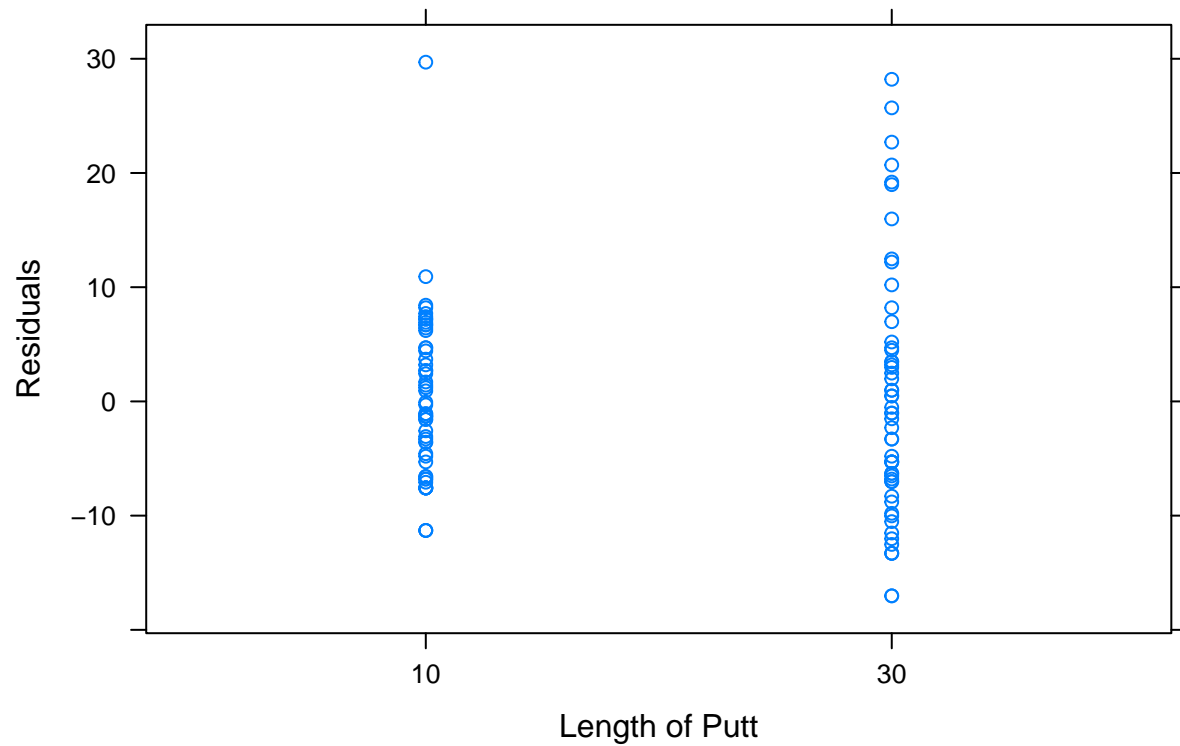
```
plot(model.3, which = 1:2)
```



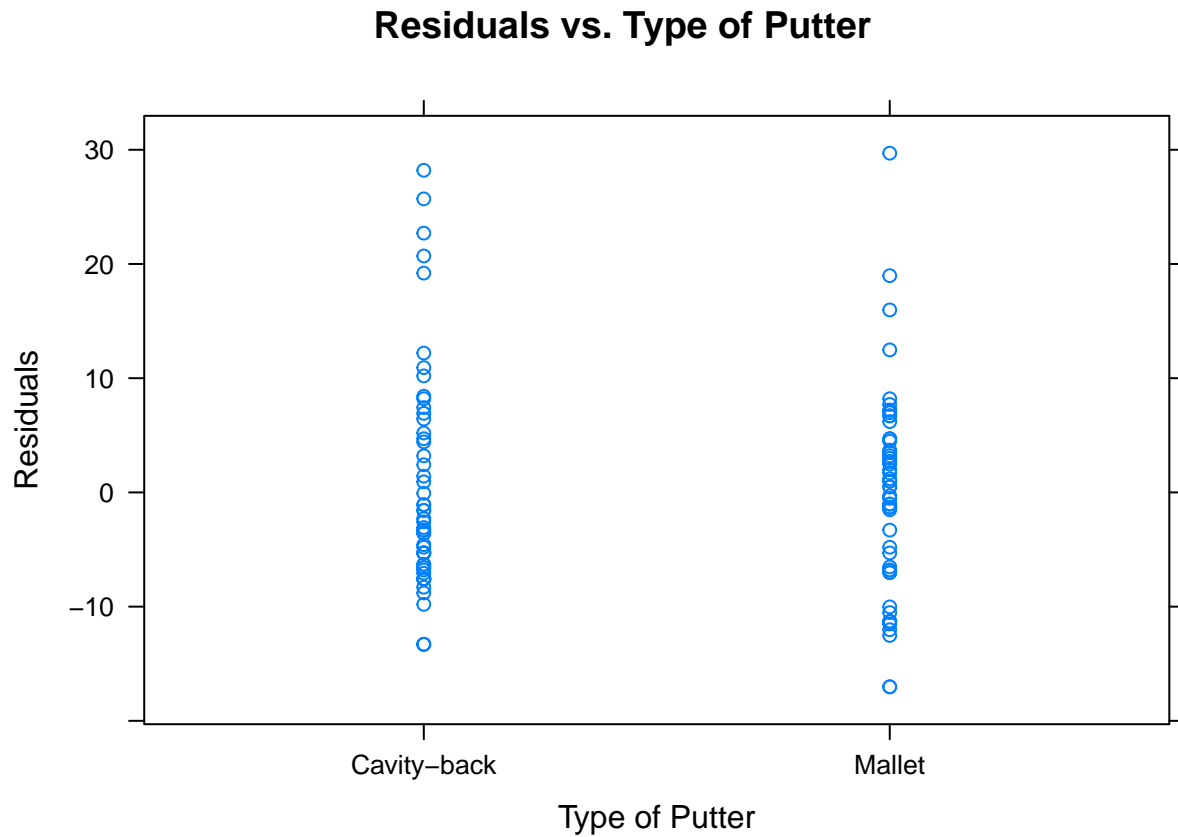
```
library(lattice)
df5 <- data.frame(x = as.factor(data.3$Length.of.Putt), y = model.3$residuals)
xyplot(y~x, data = df5,
       xlab = "Length of Putt",
       ylab = "Residuals",
       main = "Residuals vs. Length of Putt")
```



## Residuals vs. Length of Putt



```
df6 <- data.frame(x = as.factor(data.3$Type.of.Putter), y = model.3$residuals)
xyplot(y~x, data = df6,
       xlab = "Type of Putter",
       ylab = "Residuals",
       main = "Residuals vs. Type of Putter")
```



Analyzing the above residual plots we see a fanshape in the residuals vs. fitted plot which indicates non constant variance. This is further confirmed with the individual residual plots for the two significant factors A and B which show unequal fluctuations of the residuals within each factor. Additionally, there is significant curvature in the normal probability plot which indicates that the normality assumption is not satisfied. Therefore, since both the normality and constant variance conditions are not met this model is not valid.