Stats 101B Homework #3

Anna Piskun

4/26/2020

Question 1

Researchers are interested in how different blends of milk in chocolate affect the amount of antioxidants each chocolate. They have collected data from three different types of chocolate bars (milk chocolate, dark chocolate, and a dark chocolate milk chocolate blend) with the response being antioxidant level. The data is posted on CCLE.

```
data <- read.csv("HW3 Q1 Data S2020.csv")
```

a) What are the results of the F-test and conclusion of the test for the differences in means for chocolate bars.

```
#anova model is the same as the F-test
model <- aov(antioxidant~factor(chocolate), data = data)
summary(model)
```

```
##                   Df Sum Sq Mean Sq F value   Pr(>F)
## factor(chocolate)  2 1952.6   976.3   93.58 2.52e-14 ***
## Residuals         33  344.3    10.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looking at the results of the F-tast, since the f value is large and p-value is less than 0.05, we reject the null hypothesis indicating that there is a difference in means for chocolate bars. Thus, we can conclude that there is a differerence in antioxidant levels in chocolate with different blends of milk.

b) What are the estimated effects for each of the different chocolate bars?

```
#overall mean
benchmark <- mean(data$antioxidant)

#estimated effects for each of the different chocolate bars
chocolate_effects <- by(data$antioxidant, data$chocolate, function(x) mean(x)) - benchmark
chocolate_effects
```

```
## data$chocolate: dc
## [1] 10.41111
## ------------------------------------------------------------
## data$chocolate: dcmk
## [1] -4.947222
## ------------------------------------------------------------
## data$chocolate: mc
## [1] -5.463889
```

Estimated Effects for:

Dark Chocolate = 10.41111

Dark Chocolate Milk Chocolate Blend = -4.947222

Milk Chocolate = -5.463889

c) Use the Bonferroni Correction to compute confidence intervals to compare the antioxidant levels in the chocolate bars. If you want to keep the experiment-wise error rate a 5% what is the confidence level for each of the individual intervals?

Since for our experiment K = 3, we use level 1 - 0.05/3 = 0.9833. Therefore, the confidence level for each of the individual intervals is 98.33%.

```
level <- 1 - 0.05/3
level
```

```
## [1] 0.9833333
```

```
library(DescTools)
PostHocTest(model, method = "bonferroni")
```

```
##
##   Posthoc multiple comparisons of means : Bonferroni
##     95% family-wise confidence level
##
## $`factor(chocolate)`
##                diff    lwr.ci     upr.ci    pval
## dcmk-dc -15.3583333 -18.68433 -12.032340 9.4e-13 ***
## mc-dc   -15.8750000 -19.20099 -12.549007 3.8e-13 ***
## mc-dcmk  -0.5166667  -3.84266   2.809326  1.0000
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

dcmk - dc = [-18.68433, -12.032340] mc - dc = [-19.20099, -12.549007] mc - dcmk = [-3.84266, 2.809326]

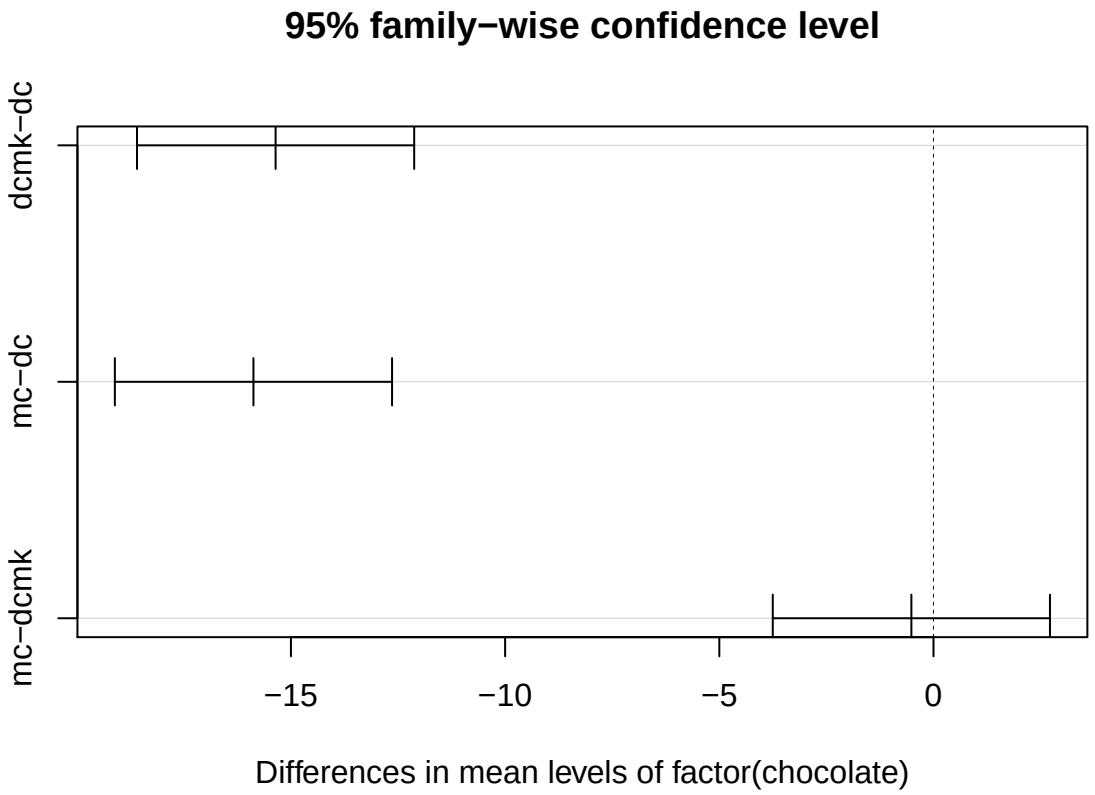
*allow for negligible rounding errors

d) Use Tukey Highly Significant Differences (HSD) intervals to compare confidence intervals. Present the plotted confidence intervals.

```
model_t <- TukeyHSD(model)
model_t
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = antioxidant ~ factor(chocolate), data = data)
##
## $`factor(chocolate)`
##                diff        lwr        upr     p adj
## dcmk-dc -15.3583333 -18.594104 -12.122562 0.0000000
## mc-dc   -15.8750000 -19.110771 -12.639229 0.0000000
## mc-dcmk  -0.5166667  -3.752438   2.719104 0.9190724
```

```
plot(model_t)
```



The only confidence interval that is not statistically significant is mc-dcmk because it includes 0 in its confidence interval and has a pvalue greater than 0.05.

2. Suppose our goal is to design a new study to measure the effect of coffee on memory that we performed on the Island. I conducted an initial study to determine whether coffee affects memory. The response variable was the subjects' score on the memory game. This test was based on two independent samples of 4 people in the treatment group and 4 people in the control group. For these questions, you'll need these data (posted on CCLE) to estimate the population standard deviation.

```
data1 <- read.csv("HW3 Q2 data.csv")
```

a) Find the sample size needed to detect an effect size of d=.2 using a power of 0.8. Put this effect size in context - at least how far apart must the mean memory scores be for there to be an 80% probability that we'll "see" an effect? (Use a two-sided alternative hypothesis. This doesn't make perfect sense, but does allow for the possibility that coffee makes memory worse.)

```
library(pwr)
pwr.t.test(d=0.2, power = 0.8, type = "two.sample")
```

```
##
##      Two-sample t test power calculation
##
##               n = 393.4057
##               d = 0.2
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

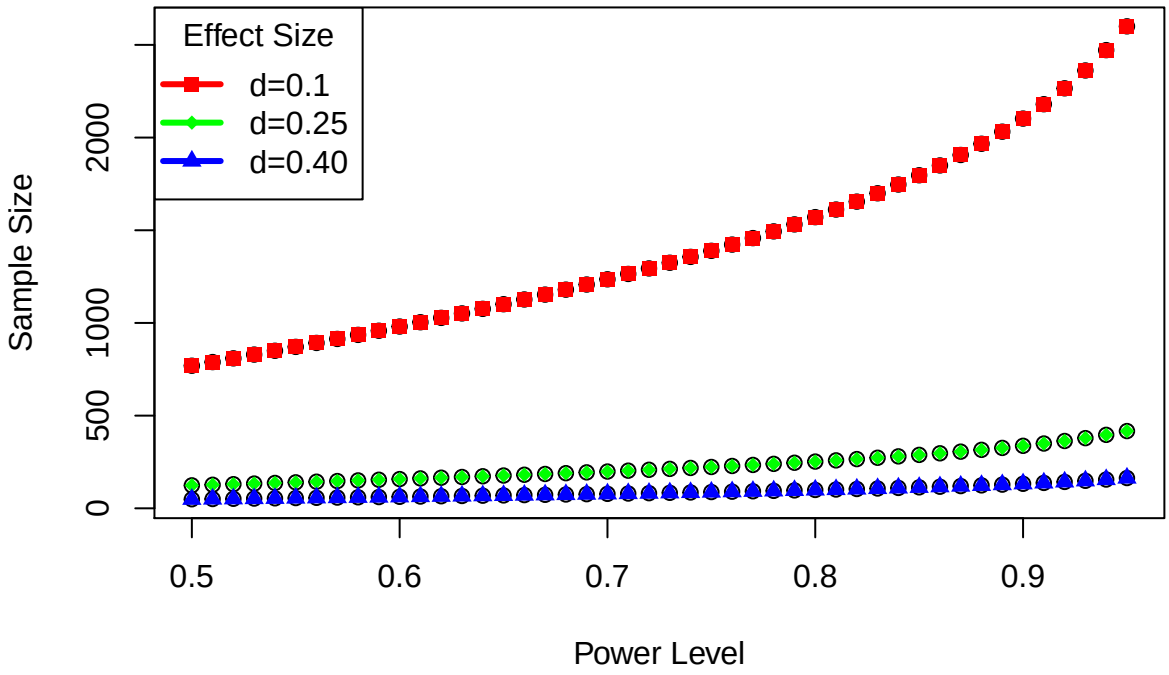
```r
#choose larger sigma to be more conservative
sd(data1$memorygame[data1$treatment == "T"])
```

```
## [1] 17.64133
```

We would need a sample size of approximately 394 (in both the treatment group and control group) so 788 total to detect and effect size of 0.2 using a power of 0.8. The sample standard deviation of memory game scores is approximately 17.64 so an effect size of 0.2 is about 3.53. This means that if we have only 394 people per group, the coffee would have to increase the mean memory scores by 3.53 points before we would have a 80% chance of "seeing" this effect.

b) Make a plot that plots power against sample size for three effect sizes: small (.1), medium (.25), and large (.4).

```r
#as explained by Professor Almohalwas' Notes
library(pwr)
a<-seq(0.5,0.95,0.01)
dd<-c(0.1,0.25,0.40)
samp <-NULL
pr1 <-NULL
pr2<-NULL
pr3<-NULL
pr<-c(pr1,pr2,pr3)
effectsize<-NULL
for (i in a) {
pwr1<-pwr.t.test(d=0.4,power=i,alternative='two.sided')
pwr2<-pwr.t.test(d=0.25,power=i,alternative='two.sided')
pwr3<-pwr.t.test(d=0.10,power=i,alternative='two.sided')
# samp1<-c(pwr1$n,pwr2$n,pwr3$n)
# power1<-c(pwr1$power,pwr2$power,pwr3$power)
samp<-c(samp,pwr1$n,pwr2$n,pwr3$n)
pr<-c(pr,pwr1$power,pwr2$power,pwr3$power)
effectsize<-c(effectsize,pwr1$d,pwr2$d,pwr3$d)}
data4<-cbind(effectsize,pr,samp)

plot(data4[,2],data4[,3],xlab="Power Level", ylab="Sample Size")
title("The Power vs. Sample size for three Effect Sizes")
points(data4[,2][data4[,1]==0.1],data4[,3][data4[,1]==0.1],col="red",pch=15)
points(data4[,2][data4[,1]==0.25],data4[,3][data4[,1]==0.25],col="green",pch=18)
points(data4[,2][data4[,1]==0.40],data4[,3][data4[,1]==0.40],col="blue",pch=17)
legend("topleft", legend=c("d=0.1","d=0.25","d=0.40"),
col=c("red","green","blue"), pch=c(15,18,17),lty=1,
lwd=3,title="Effect Size")
```



Thus we see as effect size increases, sample size decreases.

3) Question 4.11. An article in Communications of the ACM (Vol. 30, No. 5, 1987) studied different algorithms for estimating software development costs. Six algorithms were applied to several different software development projects and the percent error in estimating the development cost was observed. Some of the data from this experiment is shown in the table below.

```
data2 <- read.csv("HW3 Question 3 Data.csv")
```

a) Do the algorithms differ in their mean cost estimation accuracy?

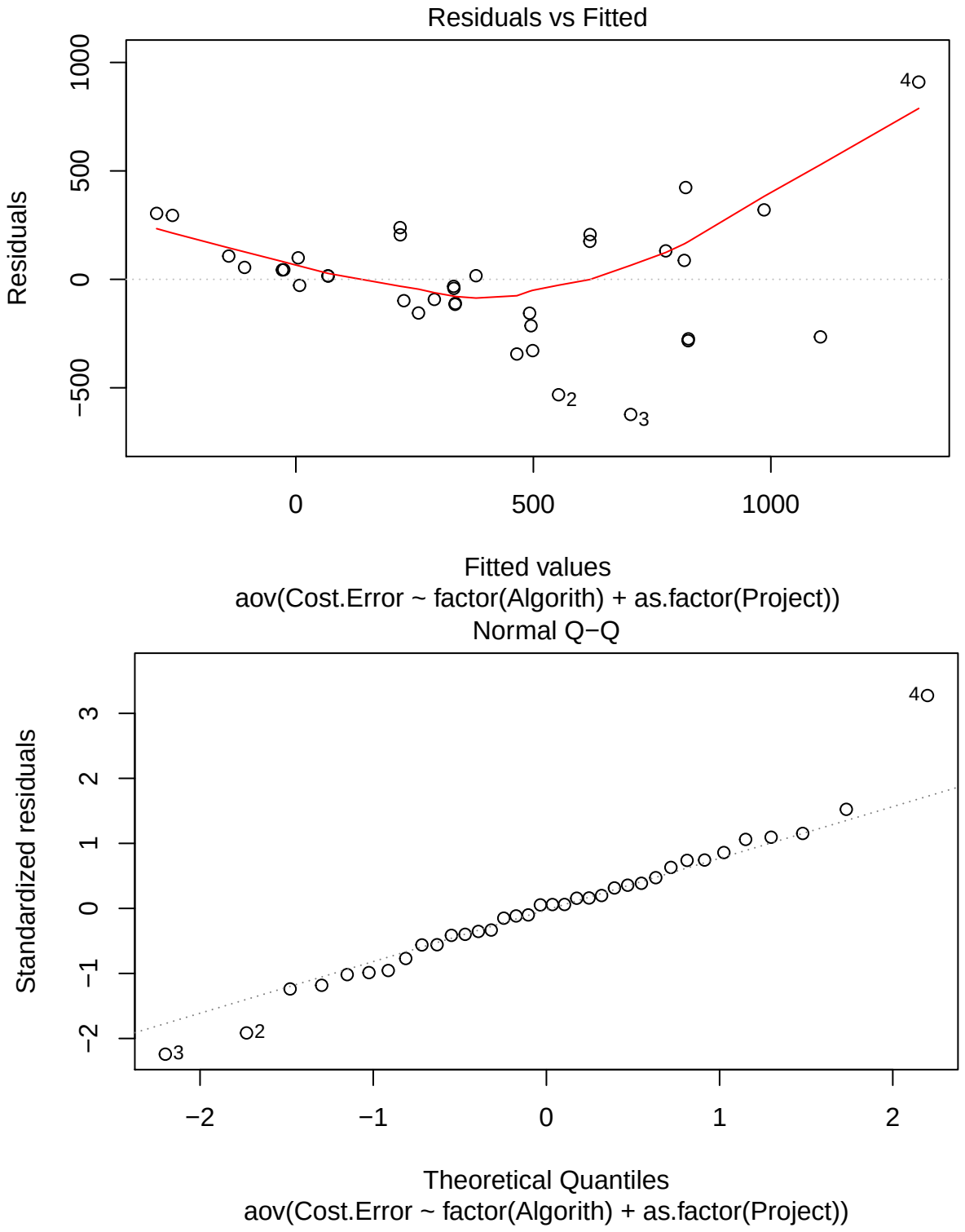
```
model1 <- aov(Cost.Error~factor(Algorith) + as.factor(Project), data = data2)
summary(model1)
```

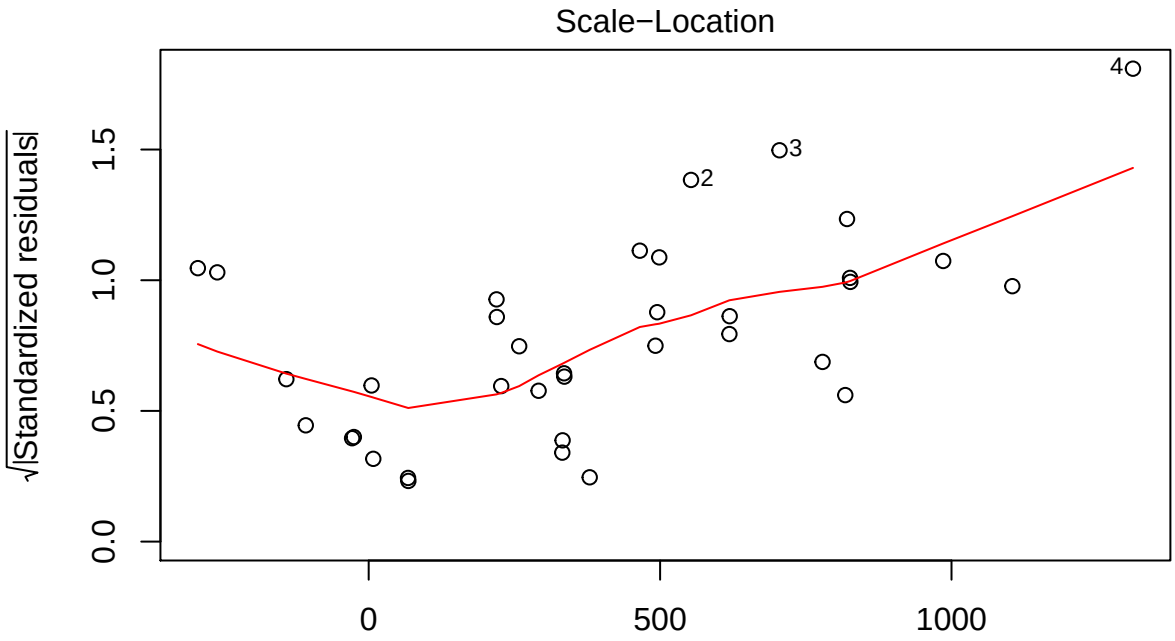
```
##                     Df  Sum Sq Mean Sq F value  Pr(>F)
## factor(Algorith)     5 2989130  597826   5.377 0.00172 **
## as.factor(Project)   5 2287339  457468   4.115 0.00730 **
## Residuals           25 2779574  111183
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

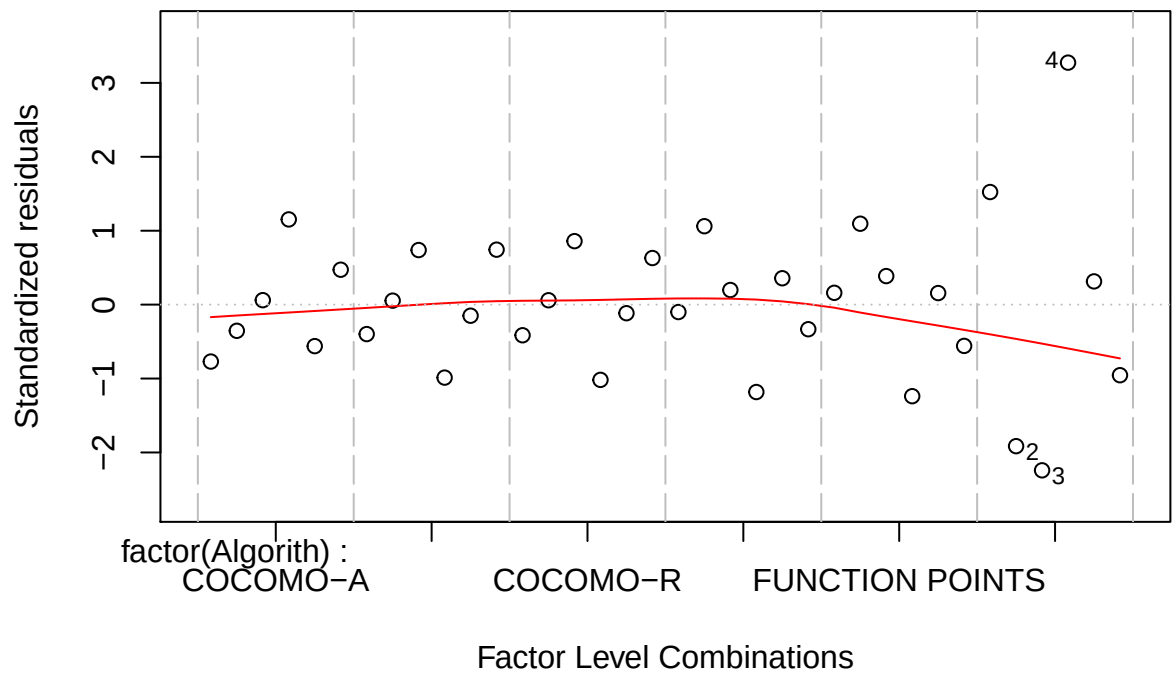
Since the p-value (0.00172) is less than 0.05, the type of algorithm is statistically significant and we reject the null hypothesis that there is no difference in mean cost estimation accuracy. Thus, we can conclude that algorithms do differ in their mean cost estimation accuracy. Additionally, the blocking factor Project is statistically significant meaning that across the different blocks the type of algorithm also differs.

b) Analyze the residuals from this experiment.

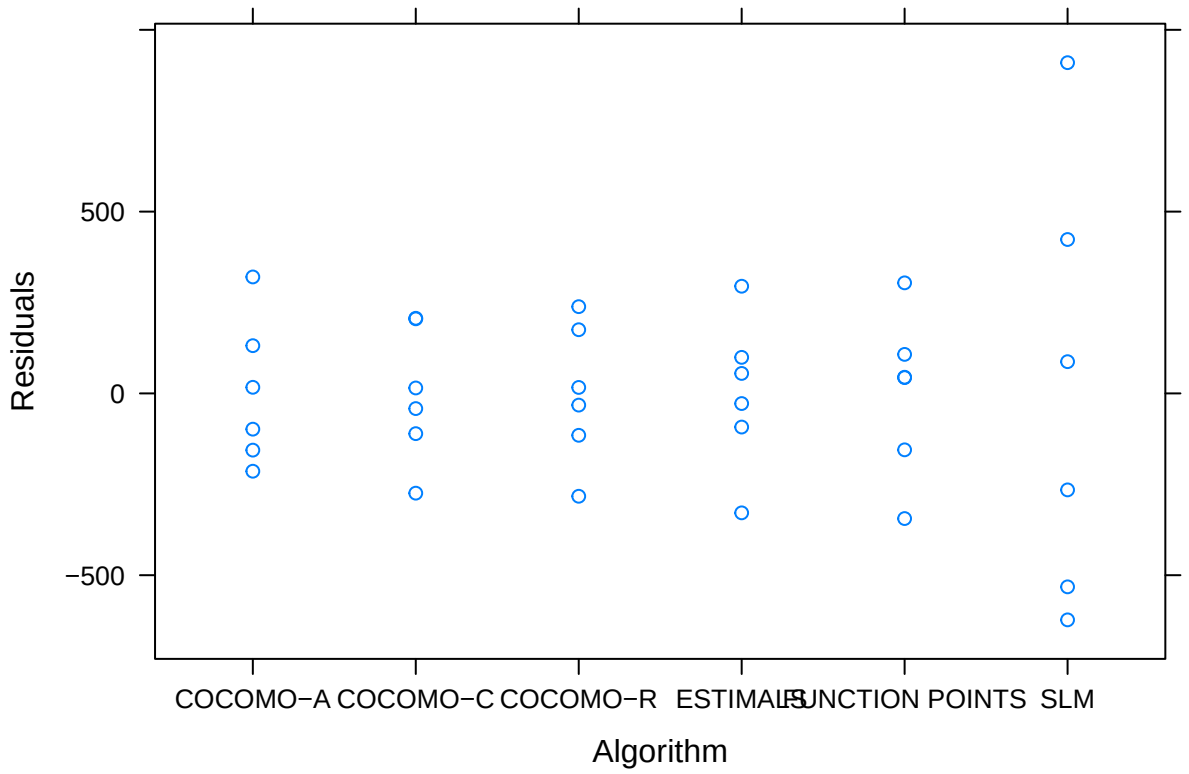
```
plot(model1)
```


Residuals vs Fitted
Residuals
1000
500
0
−500
4
2
3
0
500
1000
Fitted values
aov(Cost.Error ~ factor(Algorith) + as.factor(Project))
Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
4
2
3
−2
−1
0
1
2
Theoretical Quantiles
aov(Cost.Error ~ factor(Algorith) + as.factor(Project))






```
library(lattice)
df <- data.frame(x = as.factor(data2$Algorith), y = model1$residuals)
xyplot(y~x, data = df,
       xlab = "Algorithm",
       ylab = "Residuals")
```

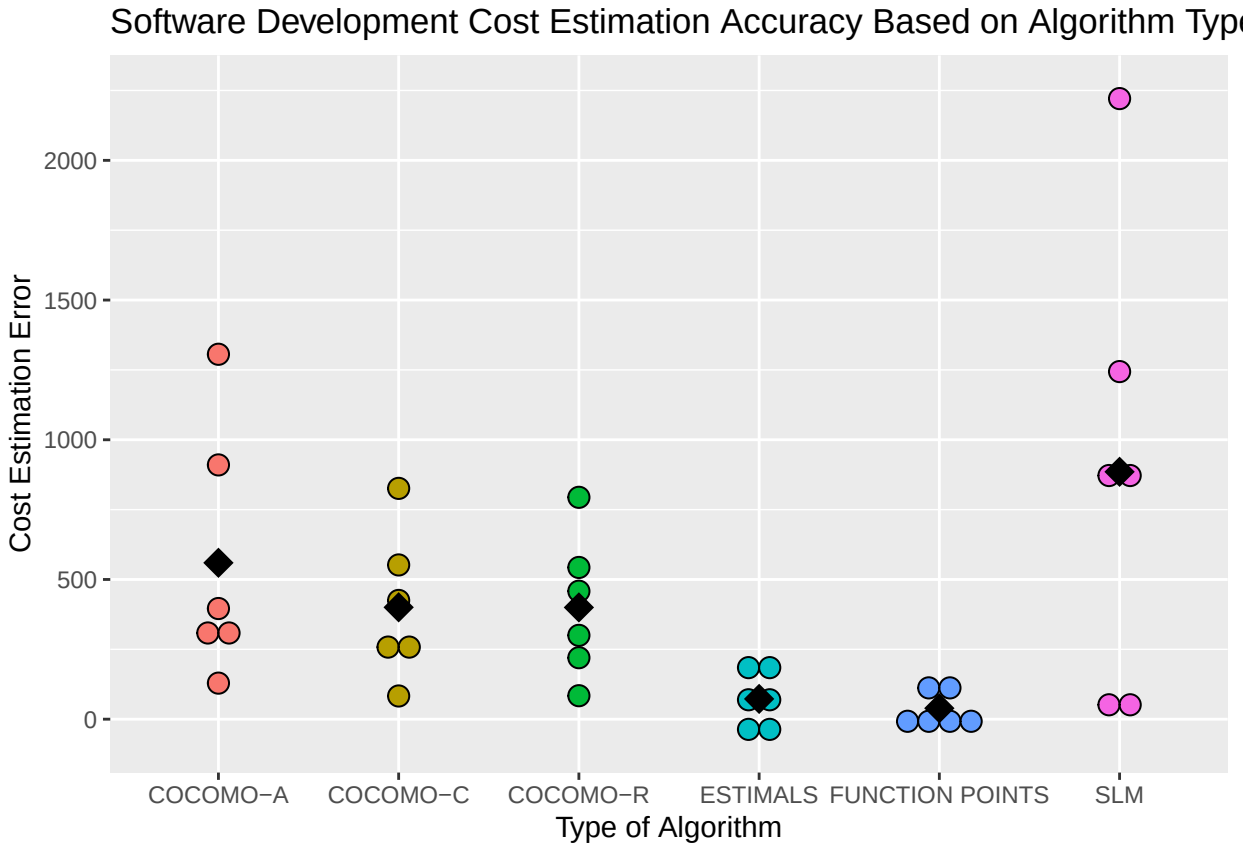


Looking at the residual plots of our model (using fitted values and just the type of algorithm) we see a clear increasing fan-shape and curve indicating that the constant variance assumption is not satisfied. The normal probability plot shows no signifiant deviation from a straight line indicating that the errors (residuals) are normally distributed. However, the Scale-Location plot again shows an increasing trend, again confirming non-constant variance in the residuals. Therefore, the model is not valid due to the curved residual plot and increasing scale location plot.

c) Which algorithm would you recommend for use in practice?

```
library(ggplot2)
ggplot(data2, aes(x=factor(Algorith), y=Cost.Error, fill = factor(Algorith))) +
  geom_dotplot(binaxis='y', stackdir='center') +
  stat_summary(fun.y=mean, geom="point", shape=18,
                 size=5, color="black") + guides(fill = FALSE) +
  xlab("Type of Algorithm") +
  ylab("Cost Estimation Error") +
  ggtitle("Software Development Cost Estimation Accuracy Based on Algorithm Type")
```

```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```



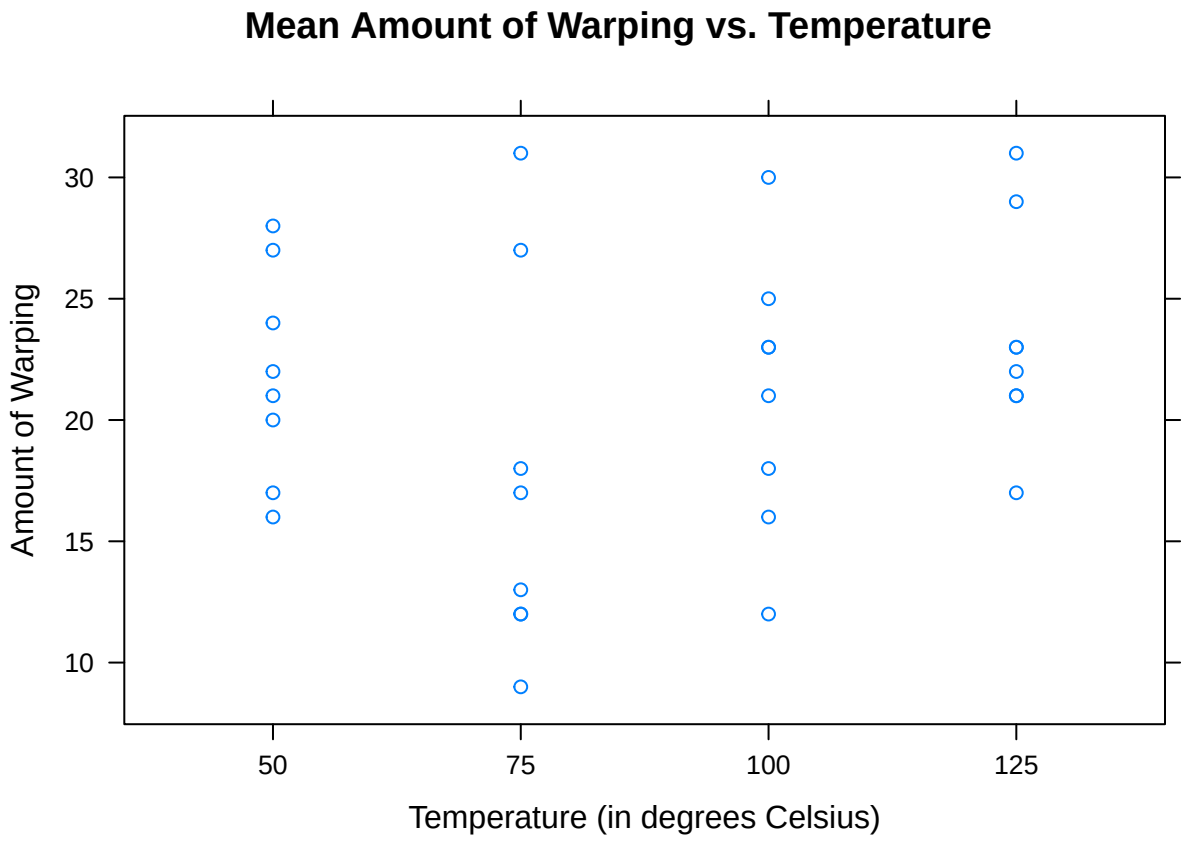
Based off of the above graphical display, we see that the FUNCTION POINTS algorithm has the lowest mean cost estimation error making it the most accurate in comparison to the other algorithms. Therefore, I would recommend using the FUNCTION POINTS algorithm in practice.

4. Question 5.7. Johnson and Leone (Statistics and Experimental Design in Engineering and the Physical Sciences, Wiley, 1977) describe an experiment to investigate warping of copper plates. The two factors studied were the temperature and the copper content of the plates. The response variable was a measure of the amount of warping. The data were as follows:

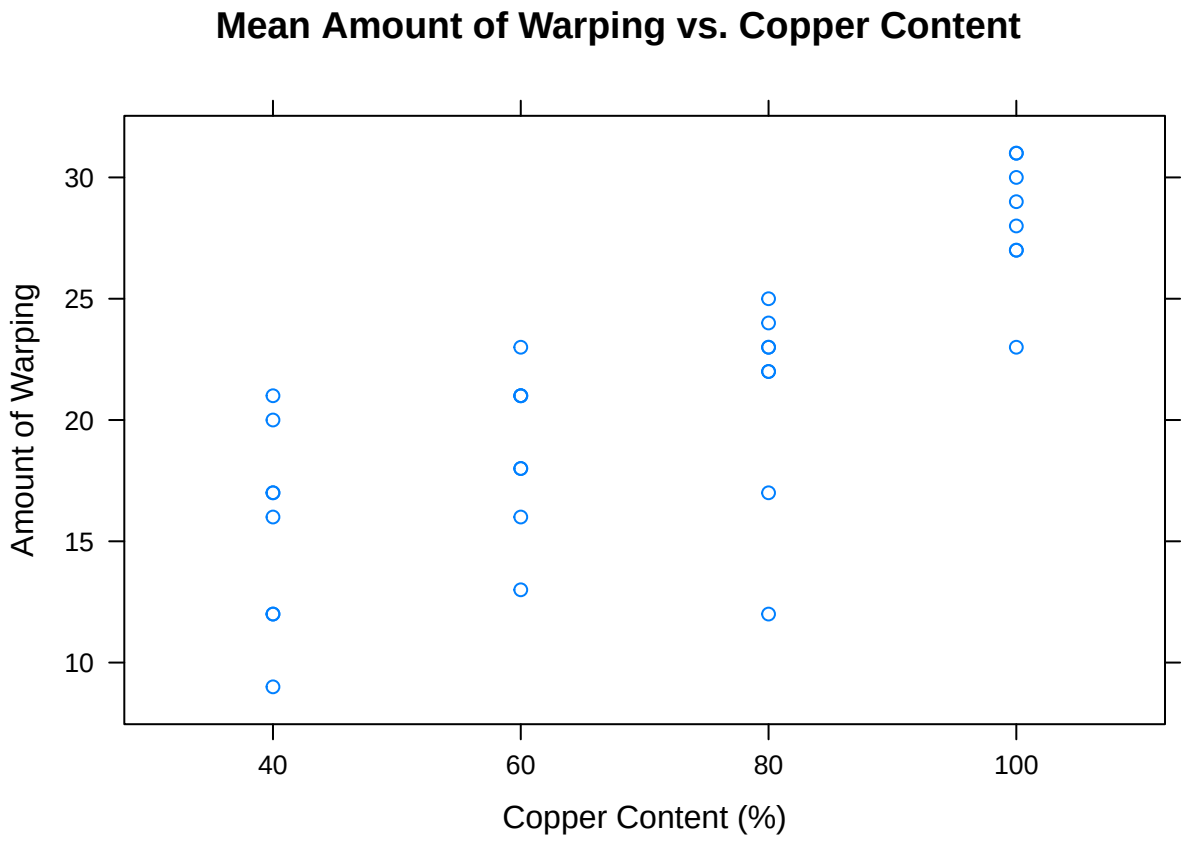
```
data3 <- read.csv("HW3 Question 4 data.csv")
```

(a) Is there any indication that either factor affects the amount of warping? Is there any interaction between the factors? Use (alpha= 0.05). Provide a graphical evedice for your answer.

```
library(lattice)
xyplot(Warping~factor(Temperature), data = data3,
       xlab = "Temperature (in degrees Celsius)",
       ylab = "Amount of Warping",
       main = "Mean Amount of Warping vs. Temperature")
```

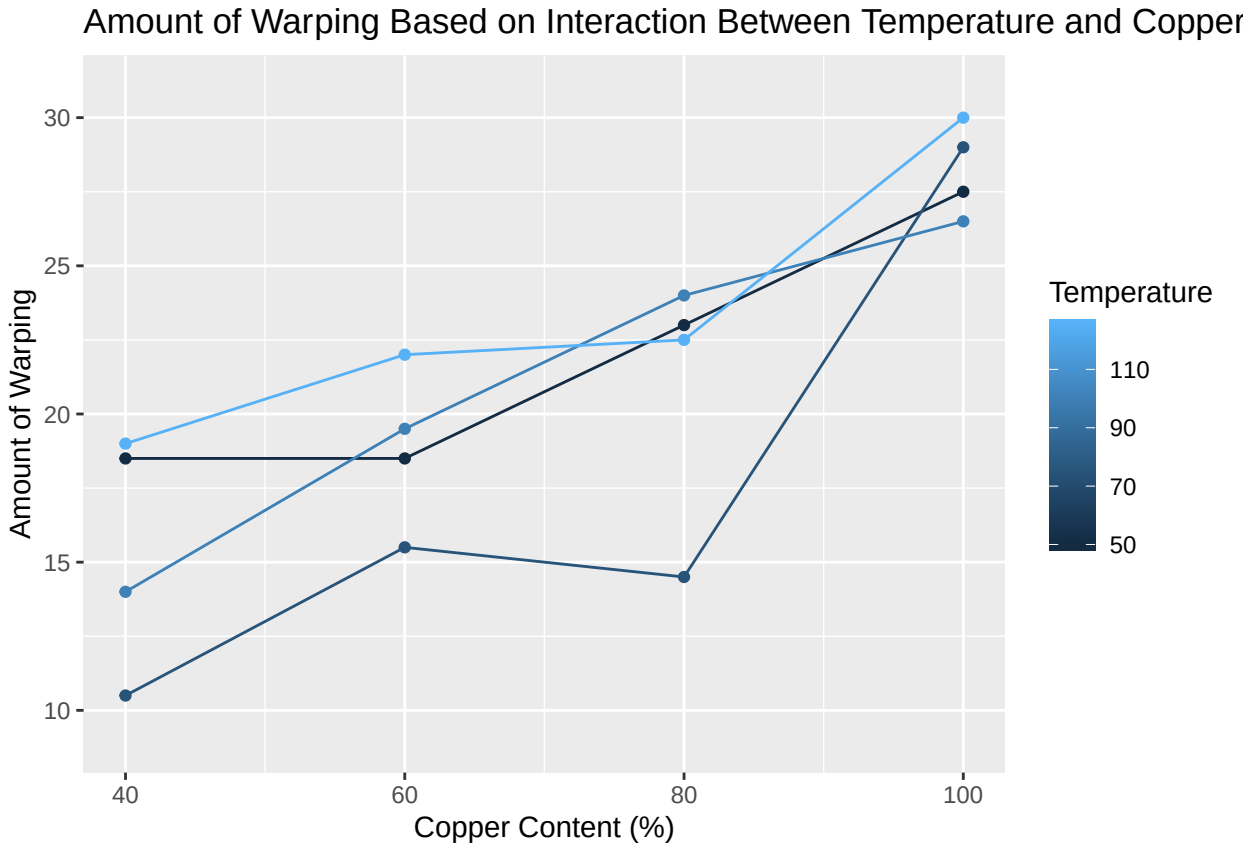



```
xyplot(Warping~factor(Copper.Content), data = data3,
       xlab = "Copper Content (%)",
       ylab = "Amount of Warping",
       main = "Mean Amount of Warping vs. Copper Content")
```




```
library(ggplot2)
ggplot() +
  aes(x = data3$Copper.Content, color = data3$Temperature, group = data3$Temperature, y = data3$Warping
  stat_summary(fun = mean, geom = "point") +
  stat_summary(fun = mean, geom = "line") +
  xlab("Copper Content (%)") +
  ylab("Amount of Warping") +
  ggtitle("Amount of Warping Based on Interaction Between Temperature and Copper Content") +
  labs(color = "Temperature")
```

```
## Warning: Ignoring unknown parameters: fun

## Warning: Ignoring unknown parameters: fun

## No summary function supplied, defaulting to `mean_se()
## No summary function supplied, defaulting to `mean_se()
```



Looking at our above plot of mean amount of warping vs. temperature, eventhough it is not linear the differences in marginal averages indicates that the temperature factor does affect warping. Looking at the plot for mean amount of warping vs. copper content, there is a clear increasing trend indicating that the copper content factor does affect warping. The third plot above, indicates that there is some interaction between temperature and copper content because they dont have equal rates of change. Thus, we must investigate further to understand if these interactions are significant.

b) Create an anova table (effect and interaction model), comment on the results.

```
model2 <- aov(Warping~as.factor(Copper.Content) * as.factor(Temperature), data = data3)
summary(model2)
```

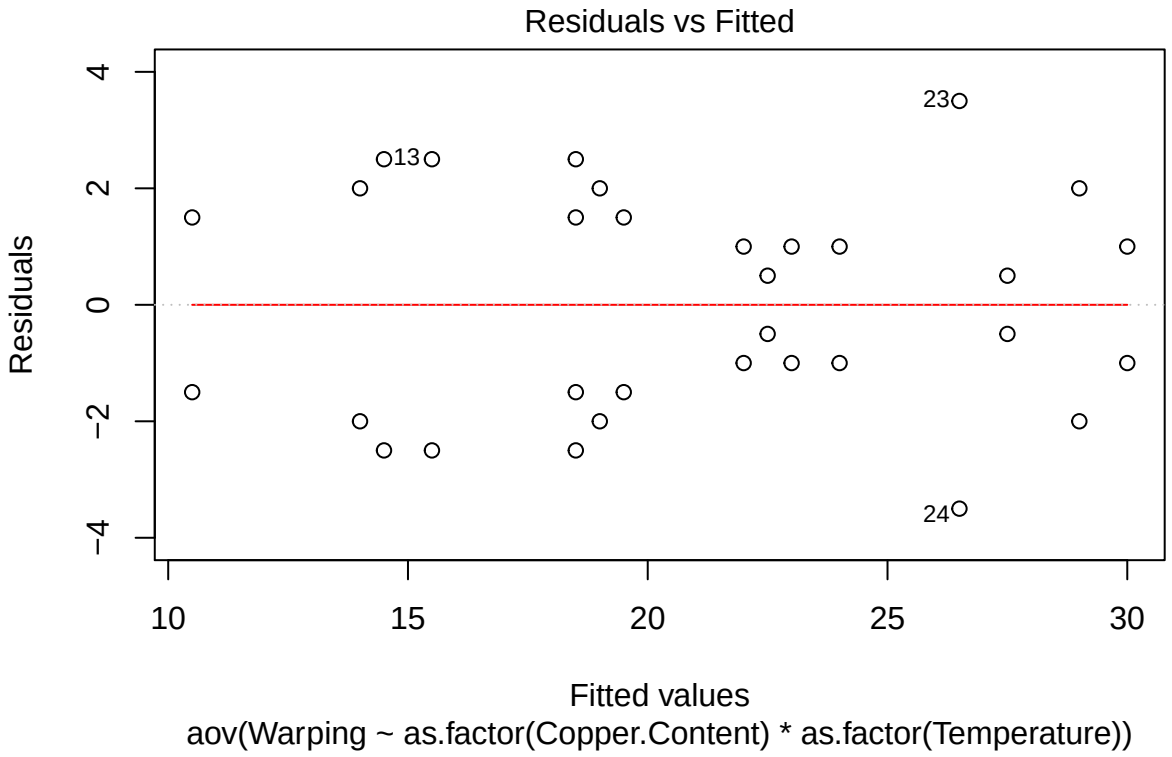
```
##                                              Df Sum Sq Mean Sq F value
## as.factor(Copper.Content)                     3  698.3  232.78  34.327
## as.factor(Temperature)                        3  156.1   52.03   7.673
## as.factor(Copper.Content):as.factor(Temperature)  9  113.8   12.64   1.864
## Residuals                                    16  108.5    6.78
##                                                Pr(>F)
## as.factor(Copper.Content)                    3.35e-07 ***
## as.factor(Temperature)                        0.00213 **
## as.factor(Copper.Content):as.factor(Temperature)  0.13275
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

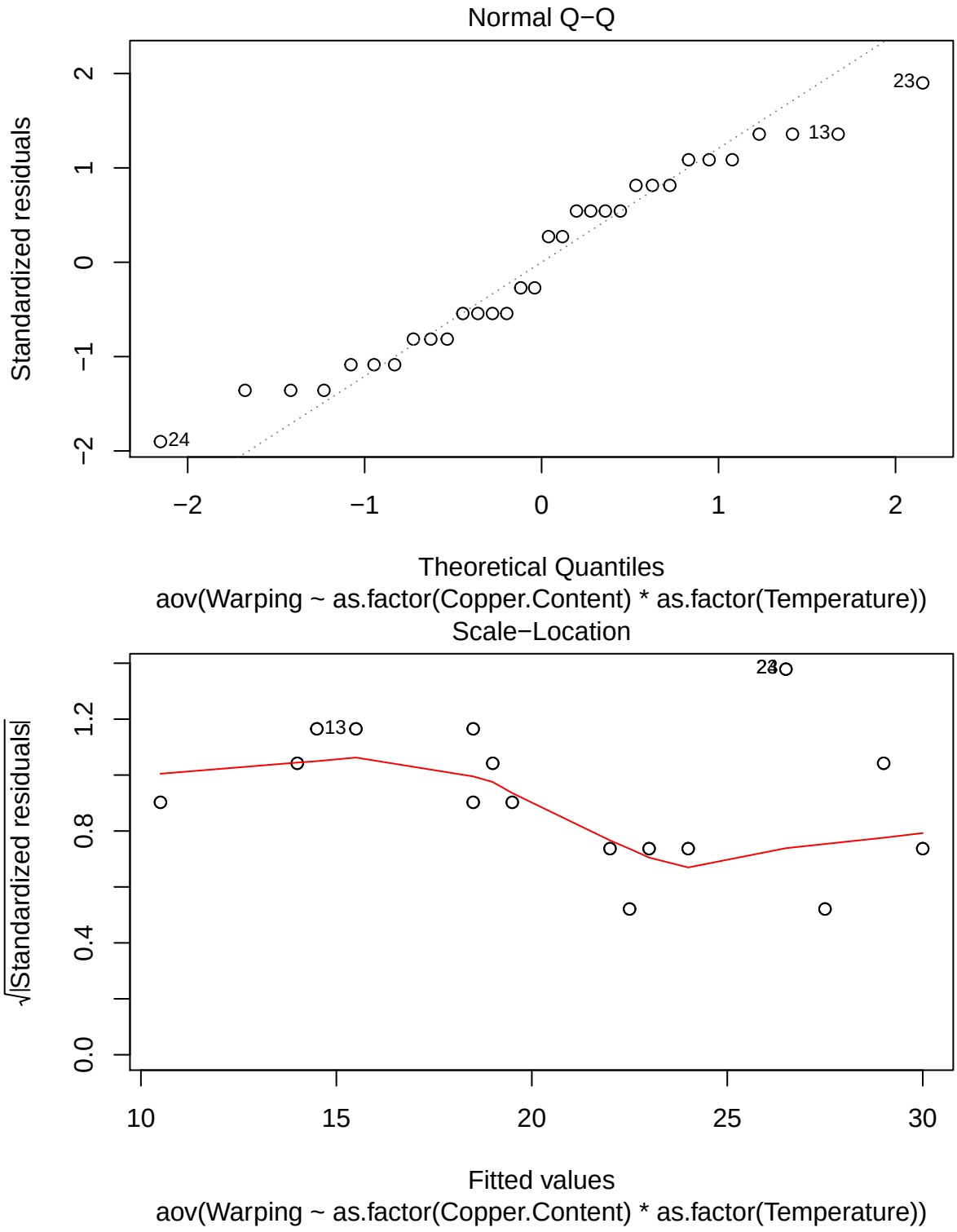
The first row of our anova table tells us that the amount of variability of the amount of warping between copper content percentages is very big, relative to the error residuals. Because the p-value is less than 0.05, we reject the null hypothesis and conclude that copper content does produce different amounts of warping. The second row tells us that varying temperature degrees is statistically significant and does result in different

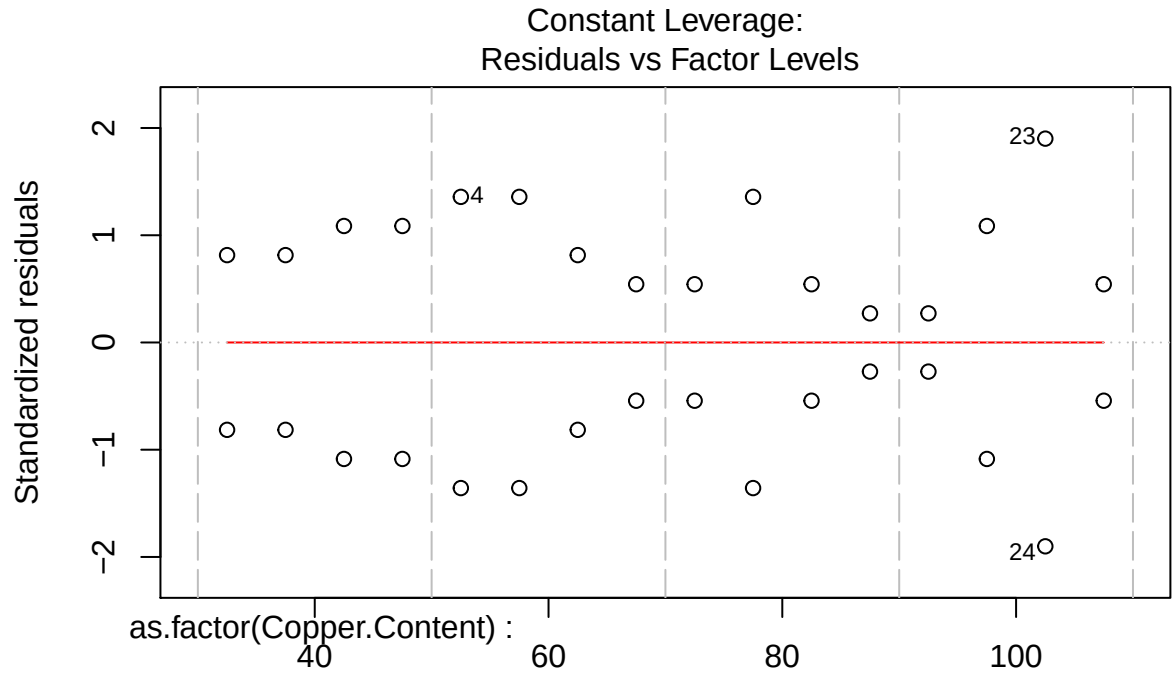
amounts of warping. Since the p-value is less than 0.05, we reject the null hypothesis and conclude that temperature does produce different amounts of warping. The third row tells us that the interaction between Copper Content and Temperature is not significant, thus the copper content (%) does not depend on the temperature and vice versa.

c) Analyze the residuals from this experiment

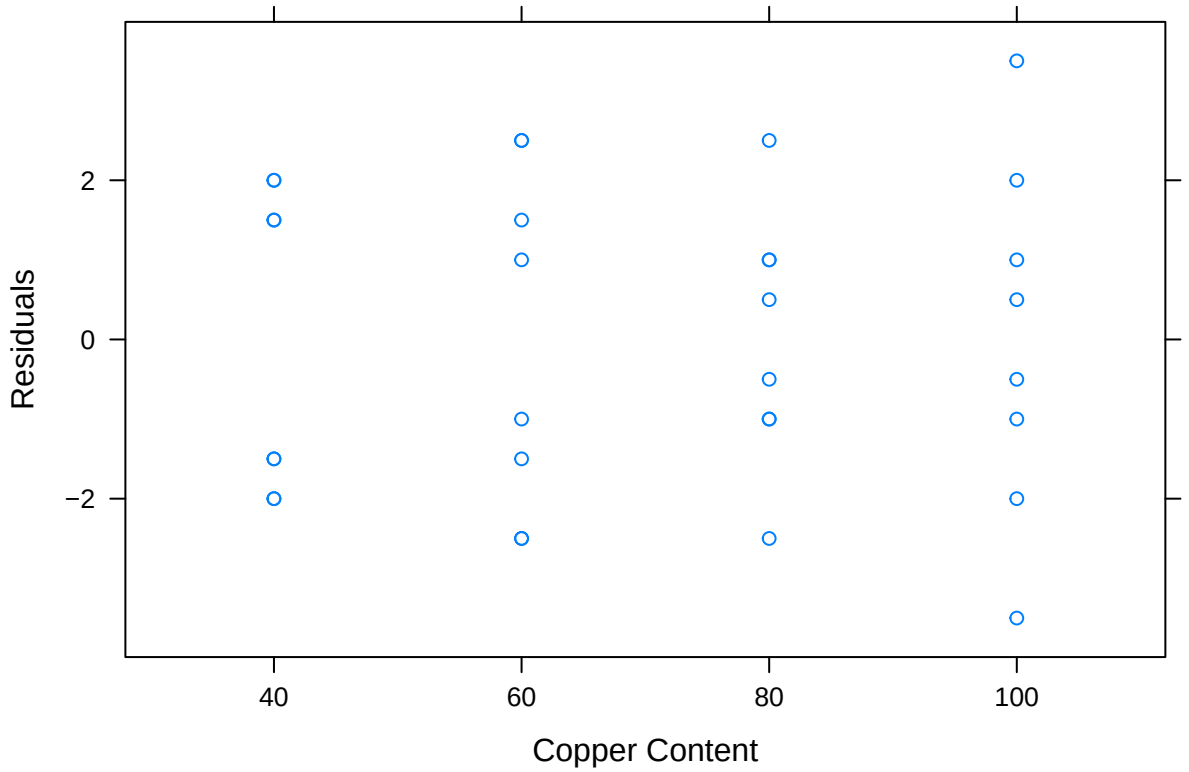
```
plot(model2)
```



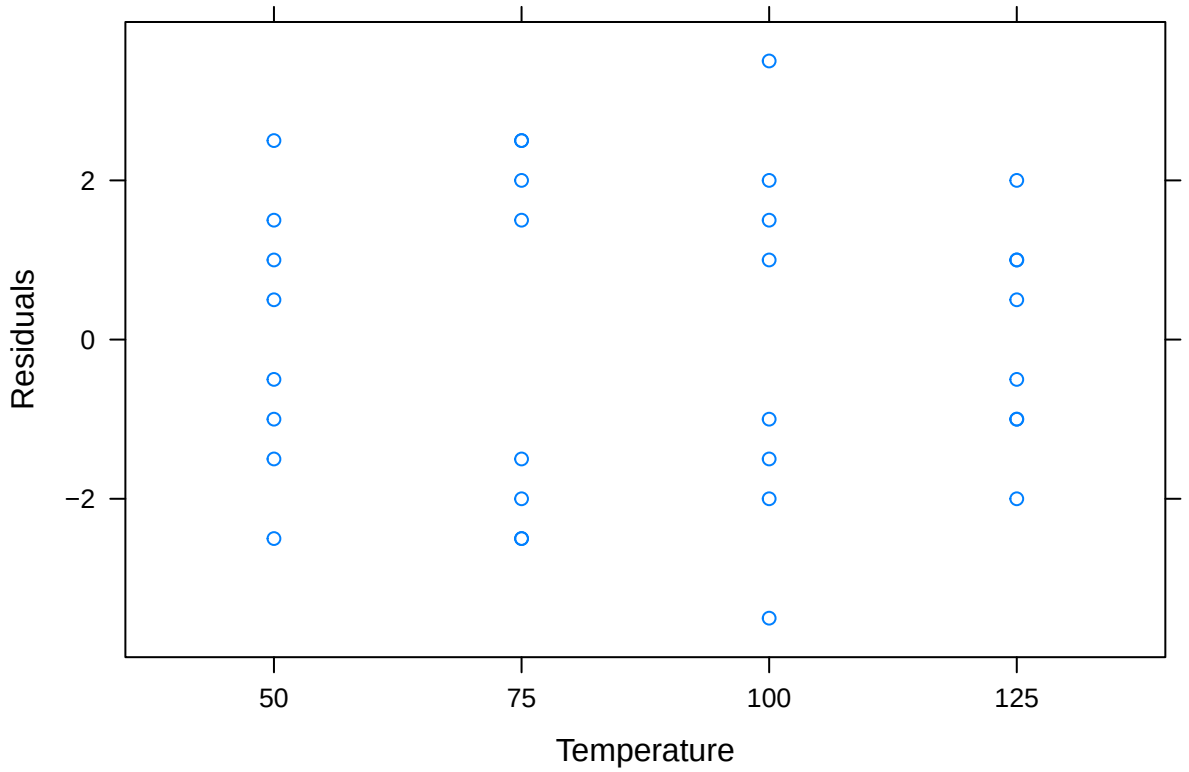

Normal Q−Q
Standardized residuals
Theoretical Quantiles
aov(Warping ~ as.factor(Copper.Content) * as.factor(Temperature))
Scale−Location
√|Standardized residuals|
Fitted values
aov(Warping ~ as.factor(Copper.Content) * as.factor(Temperature))




```
library(lattice)
df1 <- data.frame(x = as.factor(data3$Copper.Content), y = model2$residuals)
xyplot(y~x, data = df1,
       xlab = "Copper Content",
       ylab = "Residuals")
```



```
df2 <- data.frame(x = as.factor(data3$Temperature), y = model2$residuals)
xyplot(y~x, data = df2,
       xlab = "Temperature",
       ylab = "Residuals")
```



Looking at the residual plot for our model, there is no clear pattern or trend indicating that the constant variance assumption is satisfied. However, our Scale - Location plot shows a decreasing trend indicating non-constance variance. The normal probability does not follow a relatively straight line, therefore indicating that the residuals are not normally distributed. Looking at the individual residual plots for Copper Content and Temperature, there are equal amounts of points above and below 0 indicating constant variance as well.

(d) Plot the average warping at each level of copper content and compare them to an appropriately scaled t distribution. Describe the differences in the effects of the different levels of copper content on warping. If low warping is desirable, what level of copper content would you specify?

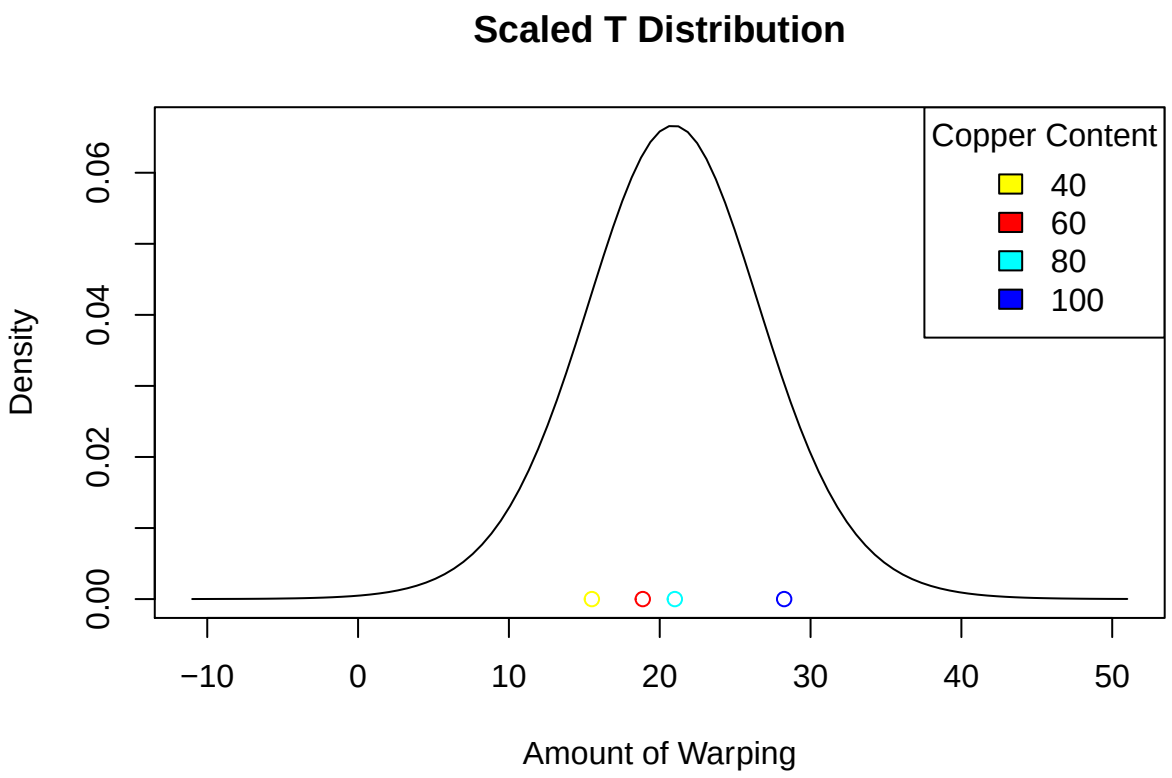
```
means <- c((mean(data3$Warping[data3$Copper.Content == "40"])), (mean(data3$Warping[data3$Copper.Conten

y <- c(0,0,0,0)

library(metRology)
```

```
##
## Attaching package: 'metRology'
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
curve(dt.scaled(x, df = 16, mean = mean(data3$Warping), sd = sd(data3$Warping)), min(data3$Warping) - 2
points(x = means, y = y, type = "p", col = means)
legend("topright", title = "Copper Content", legend = c("40", "60", "80", "100"), fill = means,
```

```
       col = means)
```



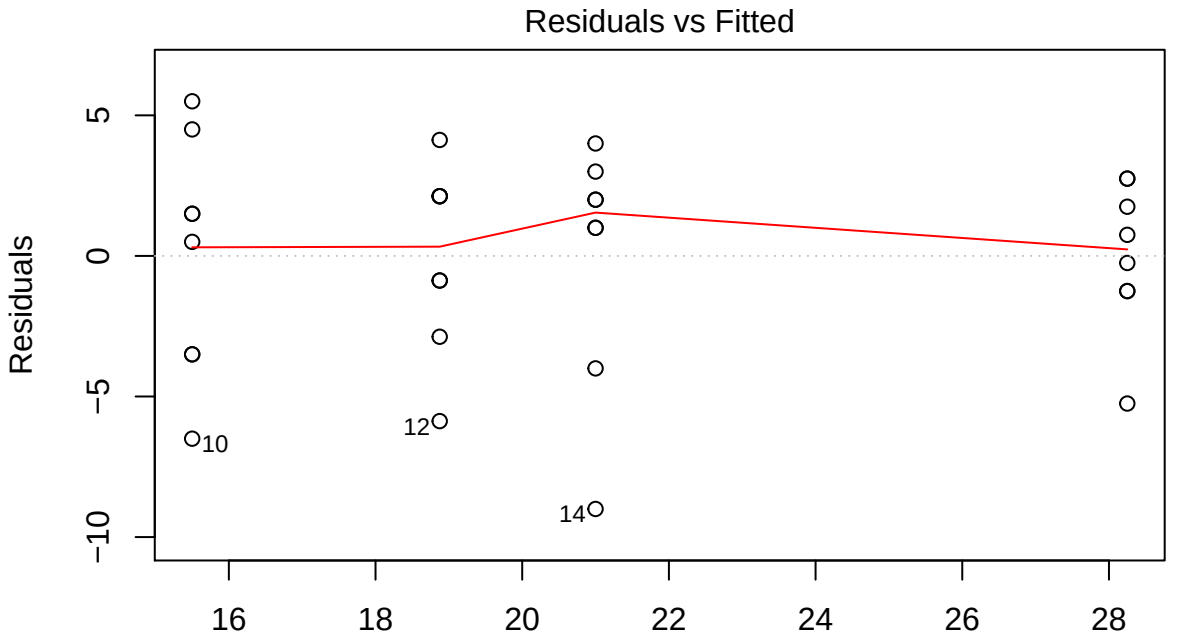
Plotting the average warping at each level of copper content we see that warping increases as copper content percentage increases. Thus, if low warping is desirable I would recommend a copper content level of 40%.

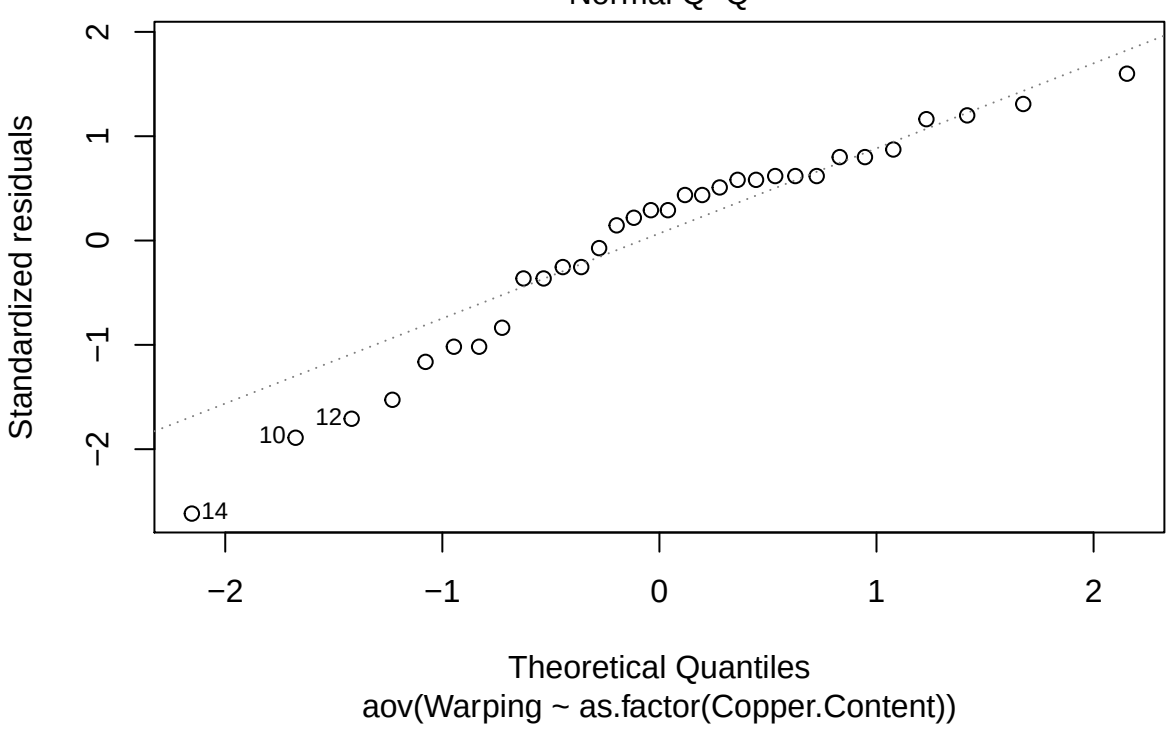
e) Suppose that temperature cannot be easily controlled in the environment in which the copper plates are to be used. Does this change your answer for part (c)?

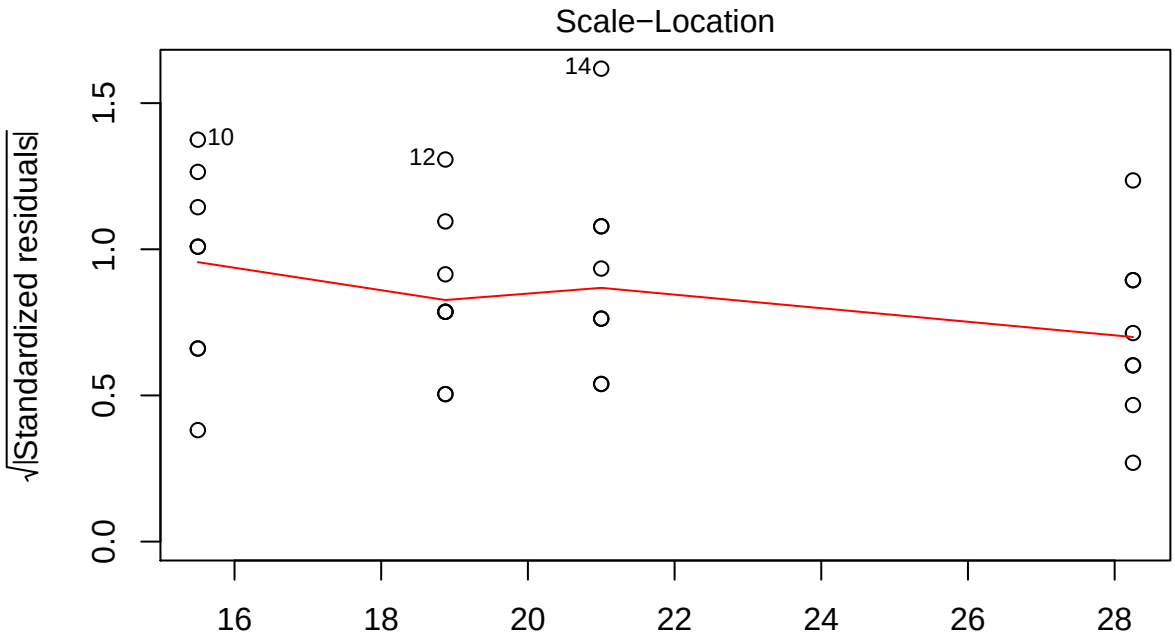
```
model3 <- aov(Warping~as.factor(Copper.Content), data = data3)
summary(model3)
```

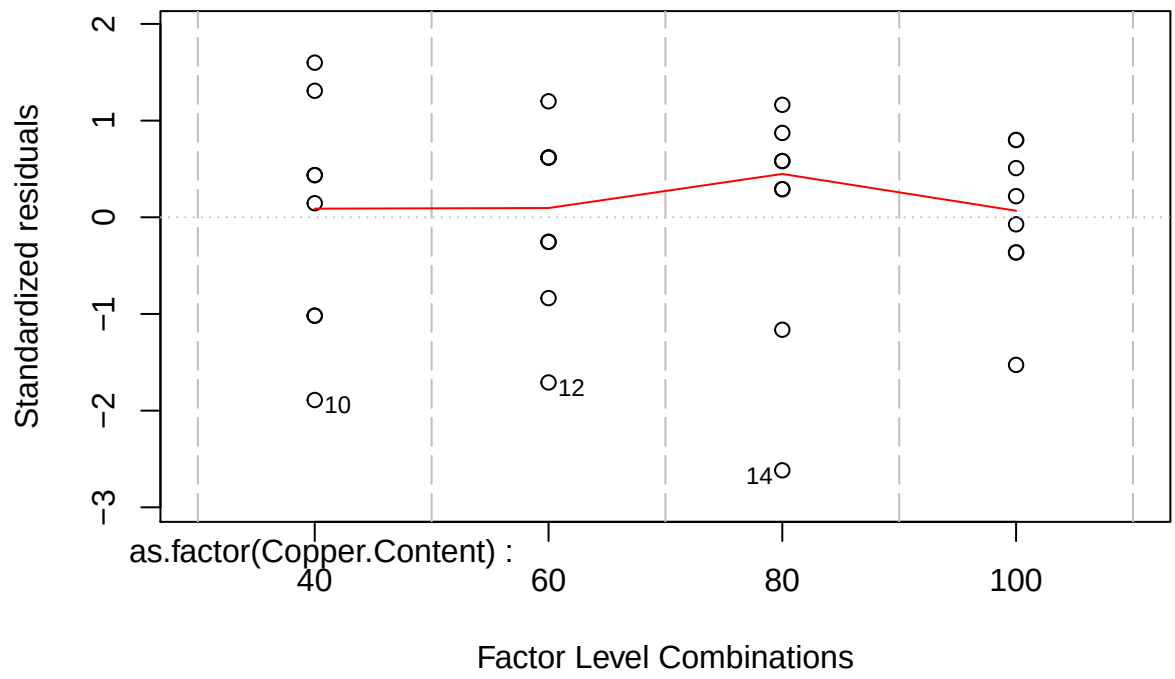
```
##                          Df Sum Sq Mean Sq F value   Pr(>F)
## as.factor(Copper.Content)  3  698.3  232.78   17.23 1.56e-06 ***
## Residuals                 28  378.4   13.51
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(model3)
```


Residuals vs Fitted
Residuals
5
0
−5
−10
10
12
14
16
18
20
22
24
26
28
Fitted values
aov(Warping ~ as.factor(Copper.Content))


Normal Q−Q
Standardized residuals
2
1
0
−1
−2
10
12
14
−2
−1
0
1
2
Theoretical Quantiles
aov(Warping ~ as.factor(Copper.Content))





Looking at the residual plot, there is no clear trend or fanshape pattern. However, the scale location plot shows a downward trend indicating non constant variance. Likewise there is significant curvature in the normal probability plot indicating that the errors not normally distributed. Thus, both models have the same issues, however the residual plot in c) is more scattered and patternless and has a less steep downward trend in its scale location plot than that of the plot in e).