STAT850 HW2

Stewart Kerr

February 4, 2019

Problem 1

a. First, I will calculate the confidence interval for the first sample (sample A):

$$CI_A = \bar{x}_A \pm t_{\alpha/2} \times SE_A = 8.92 \pm 2.23 \times \sqrt{\frac{1.40}{9}} = 8.92 \pm 0.91 = (8.01, 9.83)$$

Now I will calculate the confidence interval for the second sample (sample B):

$$CI_B = \bar{x}_B \pm t_{\alpha/2} \times SE_B = 7.41 \pm 2.31 \times \sqrt{\frac{1.30}{11}} = 7.41 \pm 0.77 = (6.64, 8.18)$$

With this procedure, because the two CIs overlap, we would fail to reject the null hypothesis that the two groups have different means.

Now, I will conduct a pooled-variance two-sample t-test. The null hypothesis is that the two groups have equal means while the alternative hypothesis is that the two means differ:

$$H_0 : \mu_A = \mu_B$$

$$H_A : \mu_A \neq \mu_B$$

The test-statistic, t, follows the t-distribution with $n_A + n_B - 2$ degrees of freedom.

$$t = \frac{\bar{X_a} - \bar{X_b}}{s_p \times \sqrt{1/n_1 + 1/n_2}}$$

$$t = \frac{8.92 - 7.41}{1.16 \times \sqrt{1/9 + 1/11}}$$

$$t = 2.90$$

$$p = 2 \times P(T_{18} < -2.90) = 2 \times 0.0047 = 0.0095$$

Because the p-value of 0.0095 is between 0.01 and 0.001, we conclude that there is strong evidence against the null hypothesis that the two groups have the same mean. Note that this is different from the conclusion resulting from the first procedure.

These two procedures lead to different conclusions because the second approach assumes, under the null hypothesis, that the two samples come from the same population while the CI approach does not make that claim. Assuming that the two samples come from the same population allows us to pool their variance for a better (that is lower variance) estimate of the sample variance, which in turn leads to a test with more power. Therefore, I would recommend we use the two independent sample hypothesis testing approach because we are more likely to accurately detect a difference in the population means.

b. A type I error is when we reject the null hypothesis, in this case that the two samples have different population means, when in reality they do not. Therefore, let's take two independent samples (A,B) of size n from the $N(\mu, \sigma^2)$ distribution. No matter what the means of A and B are, the margin of error will be the same for both samples - $1.96 \times \frac{\sigma}{\sqrt{n}}$. Therefore, this problem turns into finding the probability of taking two sample means from the same normal population that have an absolute difference of greater than two margin of errors. That is,

$$P(\text{Type I error}) = P(|\bar{X_A} - \bar{X_B}| > 2 \times 1.96 \times \frac{\sigma}{\sqrt{n}}) = P(|\bar{X_A} - \bar{X_B}| > \frac{3.92 \times \sigma}{\sqrt{n}})$$

We know that $D = \bar{X_A} - \bar{X_B} \sim N(0, \frac{2\sigma^2}{\sqrt{n}})$ because it's the difference of two iid normal variables. Thus, the problem reduces to calculating $p = 2 \times P(D > \frac{3.92\sigma}{\sqrt{n}})$ (where we used the symmetry of the normal distribution). After converting to the standard normal distribution by dividing by the standard deviation of D, we have that this probability is equal to $p = 2 \times P(Z > \frac{3.92}{\sqrt{2}}) = 2 * P(Z > 2.77)$. From R, we find this probability to be $2 \times 0.0028 = 0.0056$. Thus, the probability of a type I error using the overlapping confidence interval method is 0.0056. While this is a very low type I error rate, it reflects that this method has poor power.

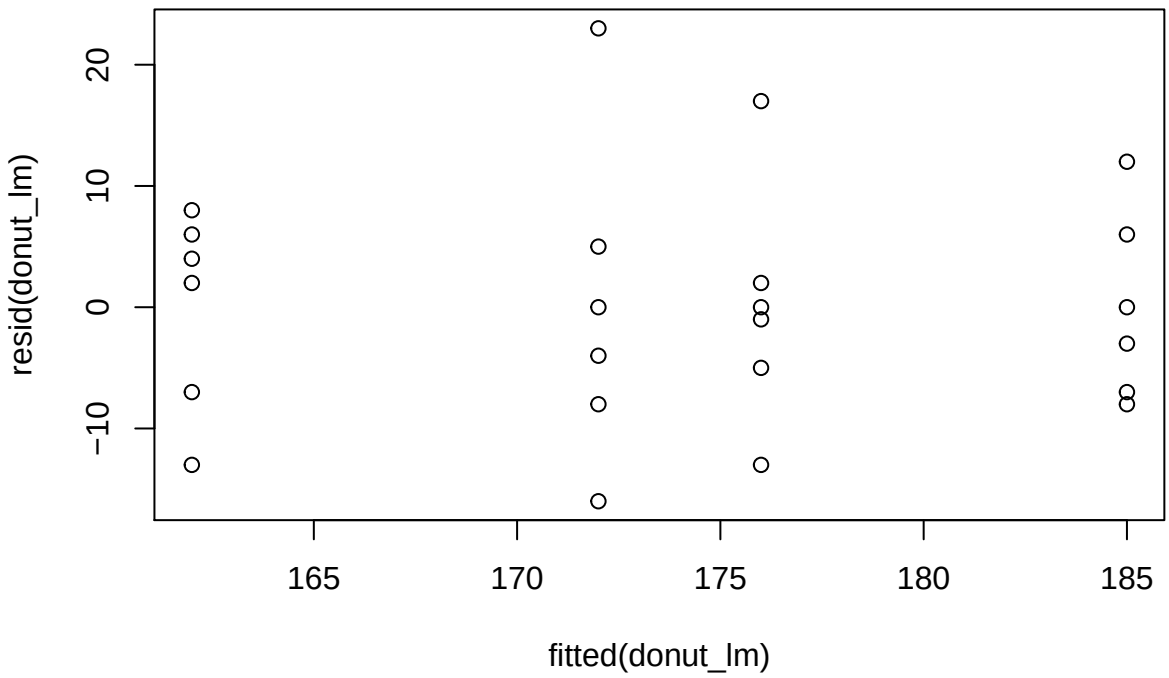
Problem 2

a.

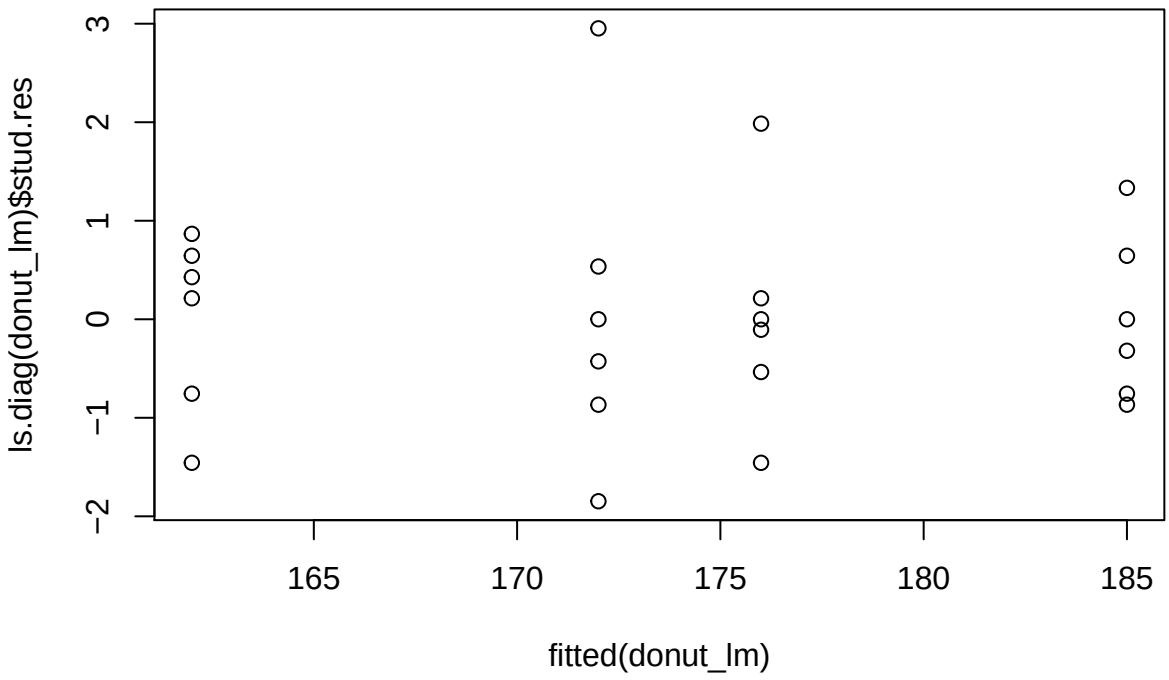
```
#Problem 2a
#Read in donut data set
donut <- read.csv("~/2019spring/STAT850/hw2/donut.txt", sep="")
donut$fat = factor(donut$fat)
#Perform F test for one-way anova
donut_lm <- lm(donut$gfa ~ donut$fat)
anova(donut_lm)
```

```
## Analysis of Variance Table
##
## Response: donut$gfa
##           Df Sum Sq Mean Sq F value   Pr(>F)
## donut$fat  3 1636.5   545.5  5.4063 0.006876 **
## Residuals 20 2018.0   100.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Plot raw/studentized residuals versus predicted values
plot(resid(donut_lm) ~ fitted(donut_lm))
```




```
plot(ls.diag(donut_lm)$stud.res ~ fitted(donut_lm))
```



The overall F-test for one-way ANOVA for this data tests the null hypothesis that the mean grams of fat absorbed is the same for each type of fat versus that at least one of the types of fat has a different mean. That is,

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$$

$$H_A : \text{At least one of } (\mu_1, \mu_2, \mu_3, \mu_4) \text{ differ}$$

From the R output, we calculate an F-test statistic of 5.41 and a p-value of 0.006876, indicating strong evidence against the null hypothesis. Next, a plot of raw residuals versus fitted values was constructed. Without a standardized y-axis, it's hard to draw many conclusions from this graph. Therefore, a plot of externally studentized residuals vs fitted values was constructed. From this plot, it appears that we likely have an outlier in the fat = 2 group. There do not appear to be any clear patterns in the residual plot and the residual for each type of fat appears roughly equal - so there does not seem to be any reason to seriously question our ANOVA assumptions.

The main difference between the raw residual plot and the studentized residual plot is in the y-axis. In the studentized plot, the residuals run from about 3 to -2 while in the raw plot they have a much larger range. Other than that, they have the same shape, which is a bit odd. The reason for this is that removing an observation affects only the regression coefficient for that type of fat. Thus the overall shape of the regression curve is not changed, only the relative distance to the mean for that particular fat type.

b. As defined in the donut dataset, x1 = 1 if the observation is from the first fat type, x2 = 2 if the observation is from the second fat type, etc. Thus, the model $gfa = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \epsilon$ is no different from $Y_{ij} = \mu + \alpha_i + \epsilon_{ij}$ where $\mu = \beta_0$ and $\alpha_i = \beta_i x_i$. However, when I try to fit this model in R, it is not able to estimate the last parameter β_4 (see below). This is because the model is overparametrized.

```
#Problem 2b
#Attempt to fit the overparametrized model
summary(lm(gfa ~ x1 + x2 + x3 + x4, data = donut))
```

```
##
## Call:
## lm(formula = gfa ~ x1 + x2 + x3 + x4, data = donut)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -16.00  -7.00   0.00   5.25  23.00
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  162.000      4.101  39.504  < 2e-16 ***
## x1            10.000      5.799   1.724 0.100075
## x2            23.000      5.799   3.966 0.000762 ***
## x3            14.000      5.799   2.414 0.025484 *
## x4                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.04 on 20 degrees of freedom
## Multiple R-squared:  0.4478, Adjusted R-squared:  0.365
## F-statistic: 5.406 on 3 and 20 DF,  p-value: 0.006876
```

c.

```
#Problem 2c
#Fit the models
lm1 = lm(gfa ~ x1 + x2 + x3, data = donut)
lm2 = lm(gfa ~ x3 + x2 + x1, data = donut)
lm3 = lm(gfa ~ z1 + z2 + z3, data = donut)
lm4 = lm(gfa ~ z3 + z2 + z1, data = donut)
#Perform anova for each model
anova1 = anova(lm1)
anova2 = anova(lm2)
anova3 = anova(lm3)
anova4 = anova(lm4)
#Demonstrate how to calculate F statistic
#This shows fitting order does not matter for overall F test
sum(anova1$`Mean Sq`[1:3]/3)/anova1$`Mean Sq`[4]
```

```
## [1] 5.406343
```

```
sum(anova2$`Mean Sq`[1:3]/3)/anova2$`Mean Sq`[4]
```

```
## [1] 5.406343
```

```
sum(anova3$`Mean Sq`[1:3]/3)/anova3$`Mean Sq`[4]
```

```
## [1] 5.406343
```

```
sum(anova4$`Mean Sq`[1:3]/3)/anova4$`Mean Sq`[4]
```

```
## [1] 5.406343
```

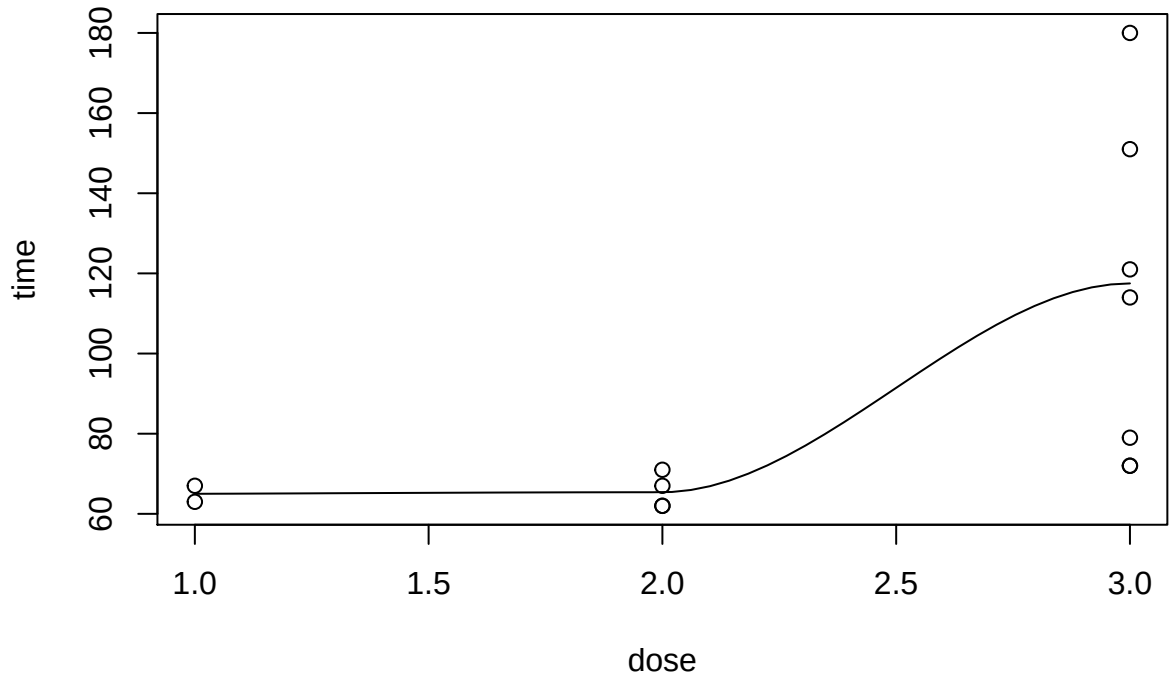
Using the anova output for each of these different models, the F-test statistic for the overall F test for $H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$ can be calculated by summing the mean square residuals for each of the fat types (x1,x2,x3), dividing by the degrees of freedom (3), and then dividing by the mean square error. As shown above, this gives you the same F-test statistic, 5.41, for each of the different fitting orders. Thus, for the overall F-test, fitting order does not matter. When fitting order is changed, it changes the individual mean

square residuals for each parameter. That is, the mean square residual is calculated *given* the previous parameters in the model. Fitting x3 will capture a different amount of variation in Y if it is fitted first than if it is fitted after x1 and x2 are already in the model.

Problem 3

a.

```
#Problem 3a
#Load the data into R
g1 = c(15,15) %>% cbind(c(67,63))
g2 = rep(25,4) %>% cbind(c(62,71,62,67))
g3 = rep(100,7) %>% cbind(c(121,114,79,151,72,180,72))
pain = data.frame(rbind(g1,g2,g3))
pain = rename(pain,dose = `.`, time = V2)
pain$dose = factor(pain$dose)
#Make plot of data
scatter.smooth(pain)
```

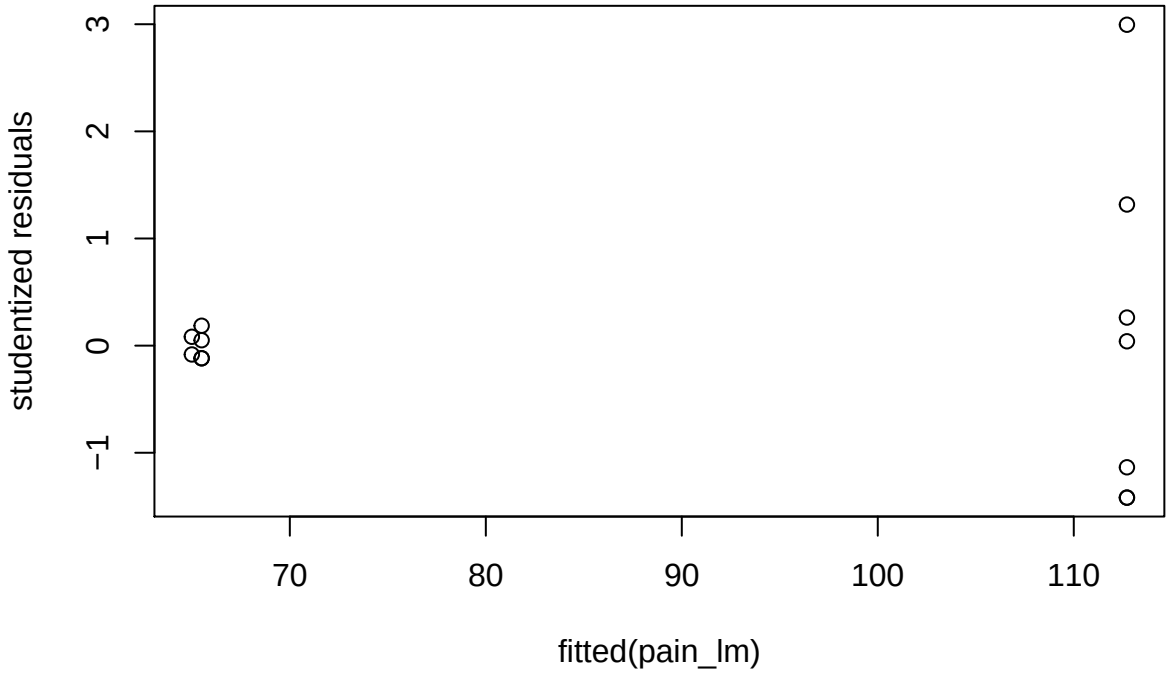



```
#Perform anova
pain_lm = lm(time ~ dose, data = pain)
anova(pain_lm)
```

```
## Analysis of Variance Table
##
## Response: time
##           Df  Sum Sq Mean Sq F value Pr(>F)
```

```
## dose       2  7253.3  3626.6  3.4277 0.0735 .
## Residuals 10 10580.4  1058.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Plot residuals
plot(ls.diag(pain_lm)$stud.res ~ fitted(pain_lm), ylab = "studentized residuals")
```



From the R ANOVA output, we would not conclude that there's a significant difference in mean pain times for each of these doses. However, in looking at the scatter plot of the data, it is clear that there is, in fact, a difference in mean response time. From the plot of residuals, it's clear that the assumption of equal variance has been violated for this data set.

b.

```
#Problem 3b
#Perform Levene's test with Brown-Forsythe mod
leveneTest(pain_lm, center=median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value  Pr(>F)
## group  2  4.6852 0.03667 *
##       10
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Levene's test assesses equal variance between each of the different dose levels. The null hypothesis is that each dose has equal variance $H_0 : \sigma^2_{15} = \sigma^2_{25} = \sigma^2_{100}$. After using leveneTest from the car package, the test statistics was calculated to be $F = 4.69$ with a p-value of $p = 0.037$. Therefore, there is moderate evidence to

suggest that these three dose levels **do not** have equal variance.

c.

```
#Problem 3c
#Perform kruskal test
kruskal.test(time ~ dose, data = pain)
```

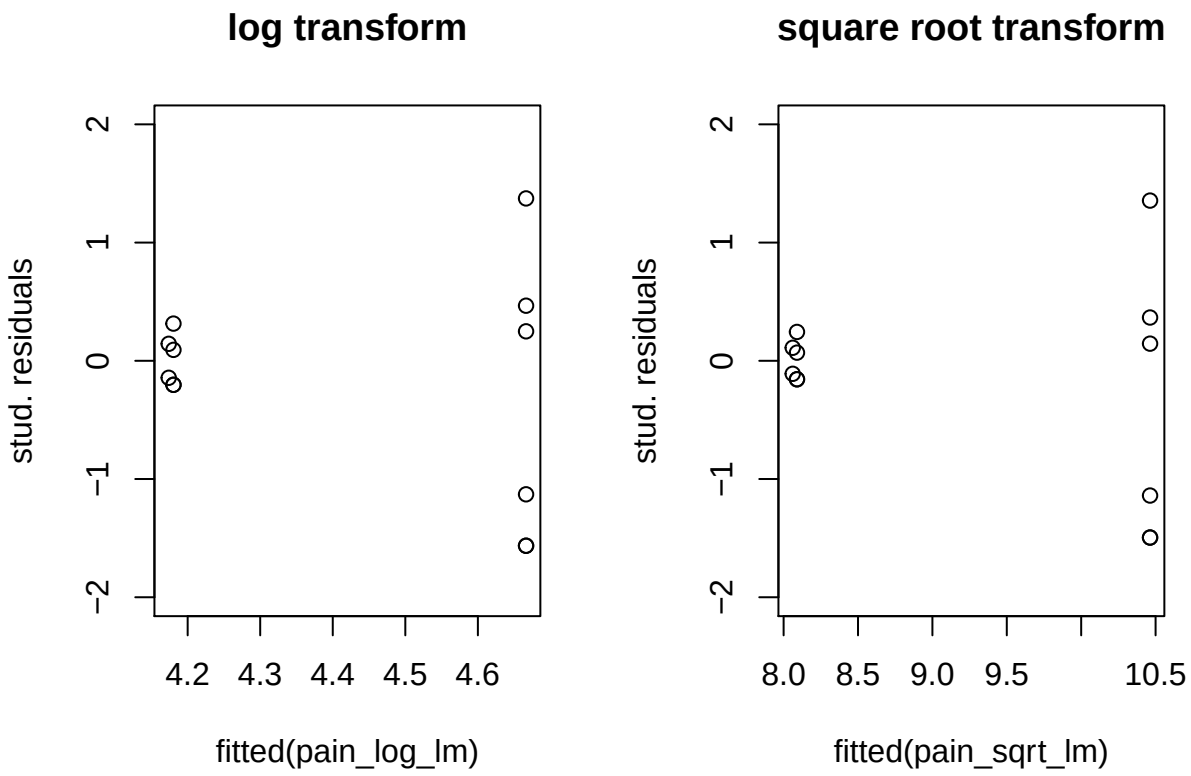
```
##
##  Kruskal-Wallis rank sum test
##
## data:  time by dose
## Kruskal-Wallis chi-squared = 9.0873, df = 2, p-value = 0.01063
```

The Kruskal-Wallis test is a non-parametric method to test if independent samples originate from the same distribution. Specifically, the null hypothesis is H_0 : The pain duration for the three doses have the same distribution. From R, we see a chi-square test statistic of 9.09 with 2 degrees of freedom. This corresponds to a p-value of 0.01063. Therefore, we conclude that there is strong evidence against the null hypothesis. It is likely that the pain duration is distributed differently for at least two of the dose levels.

d. Whatever transform we perform should make the data have variances that are more equal. With that in mind, I will perform both transformations and then look at the plot of residuals.

```
#Problem 3d
#Perform the transforms
pain_log <- mutate(pain, time = log(time))
pain_sqrt <- mutate(pain, time = sqrt(time))

#Look at the residuals
pain_log_lm <- lm(time ~ dose, data = pain_log)
pain_sqrt_lm <- lm(time ~ dose, data = pain_sqrt)
par(mfrow = c(1,2))
plot(ls.diag(pain_log_lm)$stud.res ~ fitted(pain_log_lm), ylab = "stud. residuals", main = "log transfo
plot(ls.diag(pain_sqrt_lm)$stud.res ~ fitted(pain_sqrt_lm), ylab = "stud. residuals", main = "square ro
```



From the plots, there does not appear to be a huge difference. However, the log transform looks to equalize variance a little better than the square root transform. Therefore, between the two, I would conclude that the square root transform is better. However, the assumption of equal variance is still violated and so I think the non-parametric Kruskal-Wallis test in part c is best.

e. For this data set, we seek to determine if different "doses" of capsaicin lead to different durations of pain. An unbalanced sample of subjects at dosage levels 15, 25, and 100 μL was taken and their pain time in minutes was recorded. In part a, we saw from the residual plot that the that the ANOVA assumption of equal variance was likely violated - this was confirmed using a Levene's test in part b. Then, it part c we performed a nonparametric ANOVA test (Kruskal-Wallis) and concluded that there is strong evidence that at least one of the dosage levels yields a different pain response. From the plot of the original data, it's fairly clear that the $100\mu L$ dose of capsaicin leads to a longer pain duration. However, this pain response has a large variance as some subjects experience a reaction similar to subjects at the lower dose level. Finally, in part d we saw that performing a transform did not "fix" the data such that we could assume equal variance. Thus, the test performed in part c was most appropriate.

Code

```
## ----include = FALSE-----------------------------------------------------------
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(knitr)
library(car)

set.seed(1104)             # make random results reproducible
```

```
this_file <- "kerr_stat850_hw02.Rmd"  # used to automatically generate code appendix

## ------------------------------------------------------------------------
#Problem 2a
#Read in donut data set
donut <- read.csv("~/2019spring/STAT850/hw2/donut.txt", sep="")
donut$fat = factor(donut$fat)
#Perform F test for one-way anova
donut_lm <- lm(donut$gfa ~ donut$fat)
anova(donut_lm)
#Plot raw/studentized residuals versus predicted values
plot(resid(donut_lm) ~ fitted(donut_lm))
plot(ls.diag(donut_lm)$stud.res ~ fitted(donut_lm))

## ------------------------------------------------------------------------
#Problem 2b
#Attempt to fit the overparametrized model
summary(lm(gfa ~ x1 + x2 + x3 + x4, data = donut))

## ------------------------------------------------------------------------
#Problem 2c
#Fit the models
lm1 = lm(gfa ~ x1 + x2 + x3, data = donut)
lm2 = lm(gfa ~ x3 + x2 + x1, data = donut)
lm3 = lm(gfa ~ z1 + z2 + z3, data = donut)
lm4 = lm(gfa ~ z3 + z2 + z1, data = donut)
#Perform anova for each model
anova1 = anova(lm1)
anova2 = anova(lm2)
anova3 = anova(lm3)
anova4 = anova(lm4)
#Demonstrate how to calculate F statistic
#This shows fitting order does not matter for overall F test
sum(anova1$`Mean Sq`[1:3]/3)/anova1$`Mean Sq`[4]
sum(anova2$`Mean Sq`[1:3]/3)/anova2$`Mean Sq`[4]
sum(anova3$`Mean Sq`[1:3]/3)/anova3$`Mean Sq`[4]
sum(anova4$`Mean Sq`[1:3]/3)/anova4$`Mean Sq`[4]

## ---- warning=FALSE-----------------------------------------------------
#Problem 3a
#Load the data into R
g1 = c(15,15) %>% cbind(c(67,63))
g2 = rep(25,4) %>% cbind(c(62,71,62,67))
g3 = rep(100,7) %>% cbind(c(121,114,79,151,72,180,72))
pain = data.frame(rbind(g1,g2,g3))
pain = rename(pain,dose = `.`, time = V2)
pain$dose = factor(pain$dose)
#Make plot of data
scatter.smooth(pain)
#Perform anova
pain_lm = lm(time ~ dose, data = pain)
anova(pain_lm)
#Plot residuals
```

```r
plot(ls.diag(pain_lm)$stud.res ~ fitted(pain_lm), ylab = "studentized residuals")

## ------------------------------------------------------------------------
#Problem 3b
#Perform Levene's test with Broum-Forsythe mod
leveneTest(pain_lm, center=median)

## ------------------------------------------------------------------------
#Problem 3c
#Perform kruskal test
kruskal.test(time ~ dose, data = pain)

## ------------------------------------------------------------------------
#Problem 3d
#Perform the transforms
pain_log <- mutate(pain, time = log(time))
pain_sqrt <- mutate(pain, time = sqrt(time))

#Look at the residuals
pain_log_lm <- lm(time ~ dose, data = pain_log)
pain_sqrt_lm <- lm(time ~ dose, data = pain_sqrt)
par(mfrow = c(1,2))
plot(ls.diag(pain_log_lm)$stud.res ~ fitted(pain_log_lm), ylab = "stud. residuals", main = "log transfo
plot(ls.diag(pain_sqrt_lm)$stud.res ~ fitted(pain_sqrt_lm), ylab = "stud. residuals", main = "square ro

## ----code = readLines(purl(this_file, documentation = 1)), echo = T, eval = F----
## # this R markdown chunk generates a code appendix
```