

# STAT850 HW3

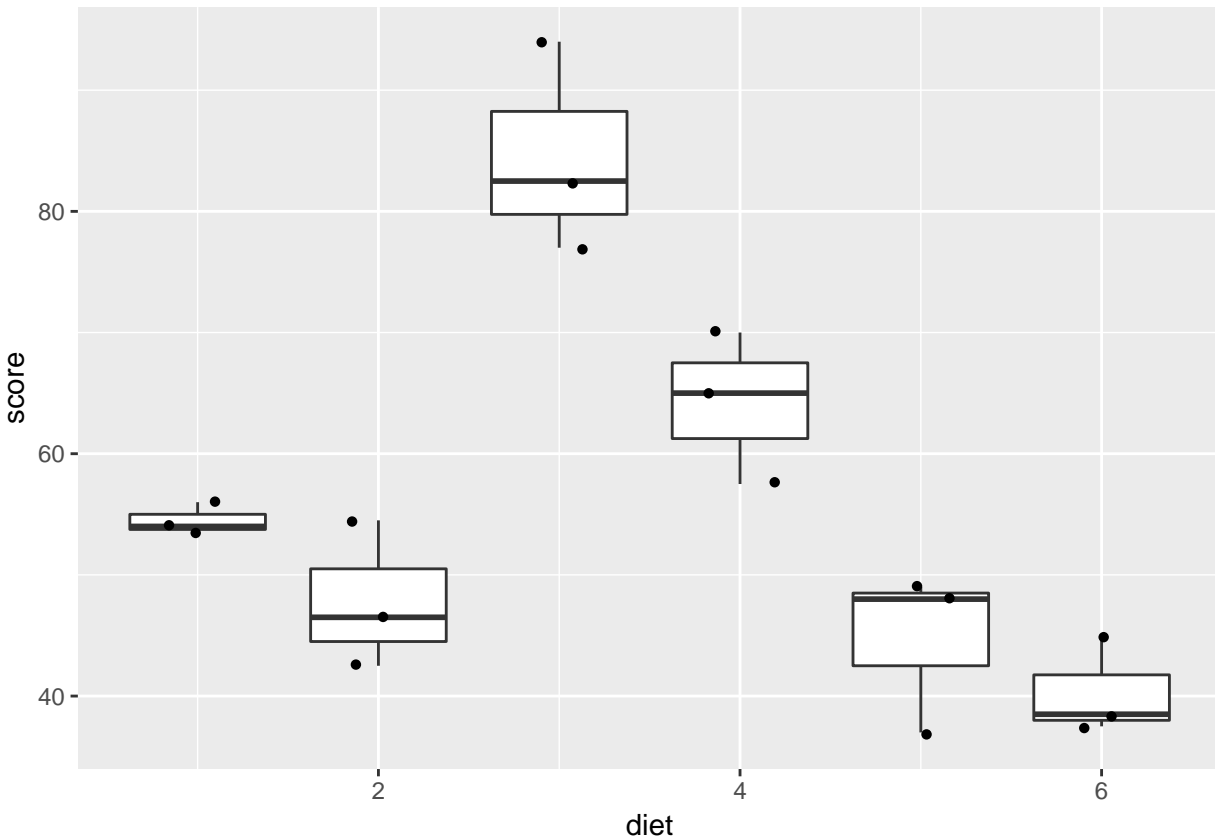
*Stewart Kerr*

*February 11, 2019*

## Problem 1

a.

```
#Problem 1a  
mouse <- read.csv("~/2019spring/STAT850/hw3/mouse.txt", sep="")  
  
#Create a grouped boxplot  
ggplot(mouse, aes(x=diet, y=score, group=diet)) +  
  geom_boxplot() +  
  geom_jitter(shape=16, position=position_jitter(0.2))
```

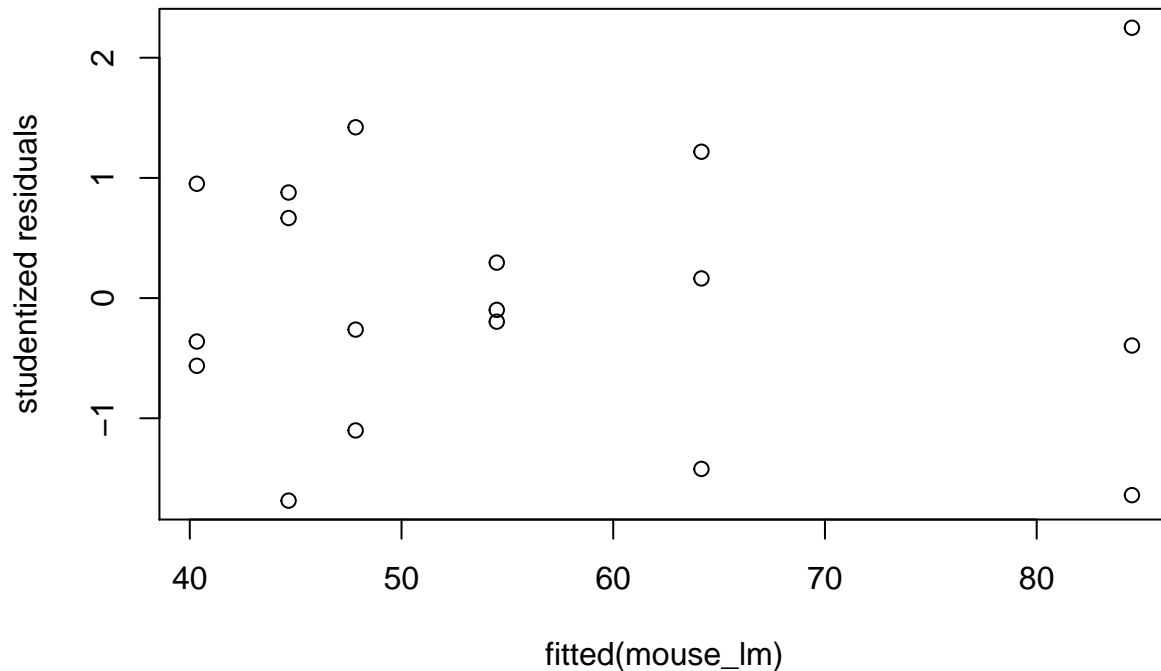


From this boxplot, it appears that diet 3, which is standard mouse chow plus vitamin C, may be the best diet for the development of optic nerves in mice babies. However, it's worth noting that a double dose of vitamin C (diet 5) yields negative effects in optic nerve development compared to the single dose. Compared to the standard diet (diet 1), vitamin E alone (diet 2) does not seem to offer much improvement.

b. As I mentioned above, compared to diet 1, there is good evidence based on the boxplots to suggest that vitamin C may have an effect on optic nerve development. The standard diet with Vitamin E, on the other hand, appears to yield optic nerve development that is similar to the standard diet alone. Vitamin B12 seems to have a negative effect on optic nerve development.

c.

```
#Problem 2c
#Construct a residuals plot
mouse$diet = factor(mouse$diet)
mouse_lm = lm(score ~ diet, data = mouse)
plot(ls.diag(mouse_lm)$stud.res ~ fitted(mouse_lm), ylab = "studentized residuals")
```



The studentized residual plot for this data is given above. The residuals for each group appear to be approximately normally distributed around 0. Most of the residuals between the groups appear to be roughly equal (equal variance) - however, the variance for diet 1 (with fitted values around 55) appears to be lower than the variance for the other groups. The assumption of equal variance for this diet may be violated, but because the variance is lower rather than higher and there is not an extremely large difference, I will assume equal variance.

d. For each of these procedures, we calculate a “least significant difference” that is compared with the differences between group means. If the difference in group means is larger than this “least significant difference,” then we say the groups differ significantly. Before I begin these procedures, I will do an F-test to determine if there is a significant difference among any of the group means thus validating our further exploration.

```
#Problem 1d
#Perform anova
anova(mouse_lm)
```

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

```
## diet      5 3965.3  793.07  22.149 1.123e-05 ***
## Residuals 12 429.7   35.81
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Collecting variables for computation
alpha = 0.05
dfError = 12
MSE = anova(mouse_lm)$`Mean Sq`[2]
SE = sqrt(MSE)
n = 3
k = 6

#LSD
LSD = LSD = qt(1-alpha/2,dfError)*SE*sqrt(2/n)

#Tukey's
Tukey = qtukey(1-alpha,k,dfError)*SE*sqrt(1/n)

#Hayter
Hayter = (1/sqrt(2))*qtukey(1-alpha,k-1,dfError)*SE*sqrt(2/n)

#Scheffe
Scheffe = SE*sqrt(2/n)*sqrt((k-1)*qf(1-alpha,k-1,dfError))

#Now make a table of the "yardsticks"
yardsticks = c(LSD,Tukey,Hayter,Scheffe)
names(yardsticks) <- c("LSD","Tukey's","Hayter","Scheffe")
yardsticks

##      LSD  Tukey's  Hayter  Scheffe
## 10.64509 16.41078 15.57294 19.25336

#Make a table of means for each group
labels <- c("Std","Std + E", "Std + C","Std + E + C","Std + 2C","Std + B6") %>% cbind(c(1,2,3,4,5,6))
labels <- data.frame(labels) %>% rename(diet = V2)
scoreMeans <- summarise(group_by(mouse,diet), mean = mean(score))
scoreMeans <- inner_join(labels,scoreMeans, by = c('diet')) %>% select(-diet)
kable(scoreMeans)
```

	mean
Std	54.50000
Std + E	47.83333
Std + C	84.50000
Std + E + C	64.16667
Std + 2C	44.66667
Std + B6	40.33333

The “yardsticks” and means are given in the R output above. See the attached sheet for the comparison of means. From the results, we observe that LSD is the most liberal of the techniques - that is, it found the most differences. Both Tukey’s and Scheffe’s procedures were the most conservative finding the fewest differences among the groups. In all of the methods, the standard diet + vitamin C appeared to differ significantly compared to all of the other diets. Therefore, I will conclude that the standard + vitamin C diet is the best for optic nerve development. It’s also worth noting that the standard diet was not found to vary significantly

with the 2nd highest scored diet - standard diet + vitamin E + vitamin C - thus, the standard diet also performs well in optic nerve development.

e. When looking at the boxplots from part A, it seems rather obvious that there is no linear relationship between vitamin C and response. Compared to the standard diet, one dose of vitamin C clearly improves the score while a double dose clearly reduces the score. However, to be rigorous, I will test the linear pattern using a linear hypothesis test. Specifically, we want to test the hypothesis of whether  $H_0 : 0.5 \times (\mu_1 + \mu_5) - \mu_3 = 0$ .

```
#Problem 1e
lht(lm(score~diet-1, data = mouse),hypothesis.matrix = c(1,0,-2,0,1,0))

## Linear hypothesis test
##
## Hypothesis:
## diet1 - 2 diet3 + diet5 = 0
##
## Model 1: restricted model
## Model 2: score ~ diet - 1
##
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1      13 2868.01
## 2      12  429.67  1    2438.3 68.1 2.732e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the test, we see an F-test statistic of 68.1 and a p-value of  $2.732 \times 10^{-6}$ , which indicates very strong evidence against the null hypothesis. That is, there is no evidence of a linear pattern in the response as a function of vitamin C.

f. The first contrast tests whether the negative interaction between vitamin C and vitamin E is equal to (or offsets) two times the main effect of vitamin C. The second contrast tests whether the two times the main effect of vitamin E is equal to the interaction effect between vitamin C and vitamin E. The third contrast tests whether there is no interaction between vitamin C and vitamin E.

```
#Problem 1f
#Create each of the contrasts
contr = contrasts(mouse$diet)
contr[,1] = c(1,1,-1,-1,0,0)
contr[,2] = c(1,-1,1,-1,0,0)
contr[,3] = c(1,-1,-1,1,0,0)

#Perform the linear regression with contrasts
mouse_lm2 <- lm(score ~ diet, data = mouse, contrasts = list(diet = contr))
summary(mouse_lm2)

##
## Call:
## lm(formula = score ~ diet, data = mouse, contrasts = list(diet = contr))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.667 -2.625 -0.750  4.083  9.500
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    62.750      1.727  36.327 1.21e-13 ***
## diet2          -11.583      1.727  -6.706 2.18e-05 ***
```

```
## diet3          6.750      1.727   3.908 0.002081 **
## diet4          -3.417      1.727  -1.978 0.071357 .
## diet5          -18.083      3.863  -4.682 0.000531 ***
## diet6          -22.417      3.863  -5.804 8.43e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.984 on 12 degrees of freedom
## Multiple R-squared:  0.9022, Adjusted R-squared:  0.8615
## F-statistic: 22.15 on 5 and 12 DF,  p-value: 1.123e-05
```

We see that the first and second contrasts (represented by diet2 and diet3 respectively) are significantly different from zero. While there is weak to no evidence to believe that the third contrast (represented by diet4) is zero.

## Problem 2

First, before doing Fisher's LSD, we need to check if F is significant.

```
#Problem 2
F = 4.90
alpha = 0.05
df(F,3,50) < alpha
```

```
## [1] TRUE
```

As we see from the R output, the F-test does indicate that at least one of the group means is significantly different. Next, I will calculate the critical value that our computed difference in group means will be compared to.

```
#Problem 2
#Calculate the pooled variance - we will use for the t-tests
grandMean = sum((2/54)*(39.3), (25/54)*(40.1), (25/54)*(42.0), (2/54)*(43.0))
MSR = sum((2/3)*(39.3-grandMean)^2,
          (25/3)*(40.1-grandMean)^2,
          (25/3)*(42.0-grandMean)^2,
          (2/3)*(43.0-grandMean)^2)
MSE = MSR/F
SE = sqrt(MSE)
harmonicN = 1/4*((1/2) + (1/25) + (1/25) + (1/2))
LSD = qt(0.975,50)*SE*sqrt(2*harmonicN)
paste("The Fisher's LSD to compare treatment mean differences against is ",round(LSD,2))
```

```
## [1] "The Fisher's LSD to compare treatment mean differences against is  2.95"
```

From my calculations, the Fisher's Least Significant Difference (using the harmonic sample size) is about 2.95. Therefore, if the difference in means between any of the pairwise treatments is at least 2.95 then we will conclude that the two treatments have a significantly different population mean. After performing pairwise comparisons, the resulting conclusion is that we essentially have two groups - treatments 1,2,3 and treatments 2,3,4. The only difference is between treatments 1 and 4. I don't necessarily think there's anything illogical about this result, but we did have a huge difference in sample sizes between the treatment groups.

However, if I don't use the harmonic sample size, I get the result that 2 & 3 are significantly different but 1 & 4 are not - which is illogical. This shows the importance of using harmonic mean for an unbalanced design.

### Problem 3

a. For the first column of  $\mathbf{X}$ , it cannot take a value other than 1. For the other columns, we must have one and only one of the columns  $(x_1, x_2, x_3, x_4) = 1$ . Thus, we have four linearly independent rows (rank 4):

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In order to be rank 5, we would have to have a row  $[1 \ 0 \ 0 \ 0 \ 0]$ , however, this is impossible because it implies that the observation did not come from any of the groups. Thus, the maximum rank for  $\mathbf{X}$  is 4.

b. A generalized inverse of  $\mathbf{A}$  is:

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

c. I will verify that each of these are generalized inverse using the R code below.

```
#Problem 3c
X = matrix(byrow = TRUE, nrow = 6, ncol = 4, data = c(1,1,0,0,
1,1,0,0,
1,1,0,0,
1,0,1,0,
1,0,1,0,
1,0,0,1))
G1 = matrix(byrow = TRUE, nrow = 4, ncol = 4, data = c(0,0,0,0,
0,1/3,0,0,
0,0,1/2,0,
0,0,0,1))
G2 = matrix(byrow = TRUE, nrow = 4, ncol = 4, data = c(1,-1,-1,0,
-1,4/3,1,0,
-1,1,3/2,0,
0,0,0,0))
G3 = (1/18)*matrix(byrow = TRUE, nrow = 4, ncol = 4,
data = c(0,2,3,6,
0,4,-3,-6,
0,-2,6,-6,
0,-2,-3,12))
G4 = (1/54)*matrix(byrow = TRUE, nrow = 4, ncol = 4,
data = c(17,-11,-8,1,
-11,23,2,-7,
-8,2,26,-10,
1,-7,-10,35))

#Check to see if the matrices are is generalized inverses
paste("G1 is a generalized inverse: ", (t(X)%*%X)%*%G1%*(t(X)%*%X) %>% identical(t(X)%*%X))

## [1] "G1 is a generalized inverse: TRUE"
paste("G2 is a generalized inverse: ", (t(X)%*%X)%*%G2%*(t(X)%*%X) %>% identical(t(X)%*%X))

## [1] "G2 is a generalized inverse: TRUE"
paste("G3 is a generalized inverse: ", (t(X)%*%X)%*%G3%*(t(X)%*%X) %>% all.equal(t(X)%*%X))

## [1] "G3 is a generalized inverse: TRUE"
```

```
paste("G4 is a generalized inverse: ", (t(X)%*%X)%*%G4%*(t(X)%*%X) %>% all.equal(t(X)%*%X))
```

```
## [1] "G4 is a generalized inverse: TRUE"
```

Now that we've shown  $G_1$ ,  $G_2$ ,  $G_3$ , and  $G_4$  are all generalized inverses, I will calculate  $\hat{\beta}$  for each:

```
#Problem 3c
```

```
Y = matrix(nrow = 6, ncol = 1, data = c(72,36,12,48,12,36))
B1 = G1%*%t(X)%*%Y
B2 = G2%*%t(X)%*%Y
B3 = G3%*%t(X)%*%Y
B4 = G4%*%t(X)%*%Y
paste("mu: ",round(B1,2),
      " ,alpha1: ", round(B2,2),
      " ,alpha2: ", round(B3,2),
      " ,alpha3: ", round(B4,2))
```

```
## [1] "mu: 0 ,alpha1: 36 ,alpha2: 35.33 ,alpha3: 35.33"
## [2] "mu: 40 ,alpha1: 4 ,alpha2: 4.67 ,alpha3: 4.67"
## [3] "mu: 30 ,alpha1: -6 ,alpha2: -5.33 ,alpha3: -5.33"
## [4] "mu: 36 ,alpha1: 0 ,alpha2: 0.67 ,alpha3: 0.67"
```

From the R output above, we see that  $\hat{\beta}$  is not the same for all four generalized inverses. Furthermore, the estimates of  $\mu$  are also different for the generalized inverses (aside from  $G_3$  and  $G_4$ ). We can also see that  $\mu + \alpha_i$  and  $\sum \alpha_i$  are also different for each  $i$  and each generalized inverse.

d. I will repeat the experiment with a different  $Y$ .

```
#Problem 3d
```

```
Y = matrix(nrow = 6, ncol = 1, data = c(39,52,99,12,36,48))
B1 = G1%*%t(X)%*%Y
B2 = G2%*%t(X)%*%Y
B3 = G3%*%t(X)%*%Y
B4 = G4%*%t(X)%*%Y
paste("mu: ",round(B1,2),
      " ,alpha1: ", round(B2,2),
      " ,alpha2: ", round(B3,2),
      " ,alpha3: ", round(B4,2))
```

```
## [1] "mu: 0 ,alpha1: 48 ,alpha2: 45.11 ,alpha3: 45.11"
## [2] "mu: 63.33 ,alpha1: 15.33 ,alpha2: 18.22 ,alpha3: 18.22"
## [3] "mu: 24 ,alpha1: -24 ,alpha2: -21.11 ,alpha3: -21.11"
## [4] "mu: 48 ,alpha1: 0 ,alpha2: 2.89 ,alpha3: 2.89"
```

After comparing the output from the original experiment with the the results of our modified  $Y$  there appear to be a few tentative conclusions that we can draw. The first generalized inverse predicts 0 for  $\mu$  in both cases. The second generalized inverse predicts 0 for  $\alpha_3$  in both cases. The  $\hat{\beta}$  for  $G_3$  and  $G_4$  is the same in both cases and includes no zeroes.

e. The predicted  $\hat{\beta}$  is the same for both  $G_3$  and  $G_4$  even though the matrices look nothing alike. I'm not quite sure why this happens but I'm looking into it.

## Problem 4

We can express  $A_{main}$  in the following way:

$$A_{main} = \frac{1}{2}((\bar{a}\bar{b} - \bar{b}) + (\bar{a} - \bar{1}))$$

We can then rewrite each of the terms in this sum:

$$\bar{a}\bar{b} = \mu + \alpha_2 + \beta_2 + (\alpha\beta)_{22}$$

$$\bar{a} = \mu + \alpha_2 + \beta_1 + (\alpha\beta)_{21}$$

$$\bar{b} = \mu + \alpha_1 + \beta_2 + (\alpha\beta)_{12}$$

$$\bar{1} = \mu + \alpha_1 + \beta_1 + (\alpha\beta)_{11}$$

After plugging in these terms above and simplifying, we have:

$$A_{main} = \frac{1}{2}(2(\alpha_2 - \alpha_1) + (\alpha\beta)_{22} + (\alpha\beta)_{21} - (\alpha\beta)_{11} - (\alpha\beta)_{12})$$

Now, we consider the constraints placed on our parameters, that is:

$$\alpha_1 + \alpha_2 = 0 \implies \alpha_1 = -\alpha_2$$

$$(\alpha\beta)_{12} + (\alpha\beta)_{22} = 0 \implies (\alpha\beta)_{12} = -(\alpha\beta)_{22}$$

$$(\alpha\beta)_{21} + (\alpha\beta)_{22} = 0 \implies (\alpha\beta)_{21} = -(\alpha\beta)_{22}$$

$$(\alpha\beta)_{11} + (\alpha\beta)_{12} = 0 \implies (\alpha\beta)_{11} = -(\alpha\beta)_{12} = (\alpha\beta)_{22}$$

Now, after considering constraints, we have:

$$A_{main} = \frac{1}{2}(4\alpha_2) = 2\alpha_2$$

Of course we don't know the true parameters, thus we observe:

$$\hat{A}_{main} = 2\hat{\alpha}_2$$

Now, I we will explore the relationship between AB and these parameters. We can express AB in the following way:

$$AB = \frac{1}{2}((\bar{a}\bar{b} - \bar{b}) - (\bar{a} - \bar{1}))$$

After plugging in these terms and simplifying, we have:

$$AB = \frac{1}{2}((\alpha\beta)_{22} - (\alpha\beta)_{21} - (\alpha\beta)_{12} + (\alpha\beta)_{11})$$

Again, after considering the constraints outlined above, we have:

$$AB = \frac{1}{2}(4(\alpha\beta)_{22}) = 2(\alpha\beta)_{22}$$

$$\implies \hat{AB} = 2\widehat{(\alpha\beta)_{22}}$$

## Code

```
## ## ----include = FALSE-----
## knitr::opts_chunk$set(echo = TRUE)
## library(dplyr)
library(knitr)
## library(car)
## library(ggplot2)
##
## set.seed(1104)           # make random results reproducible
```



```

##
## this_file <- "kerr_stat850_hw03.Rmd" # used to automatically generate code appendix
##
## ## -----
## #Problem 1a
## mouse <- read.csv("~/2019spring/STAT850/hw3/mouse.txt", sep="")
##
## #Create a grouped boxplot
## ggplot(mouse, aes(x=diet, y=score, group=diet)) +
##   geom_boxplot() +
##   geom_jitter(shape=16, position=position_jitter(0.2))
##
## ## ---- warning = FALSE-----
## #Problem 2c
## #Construct a residuals plot
## mouse$diet = factor(mouse$diet)
## mouse_lm = lm(score ~ diet, data = mouse)
## plot(ls.diag(mouse_lm)$stud.res ~ fitted(mouse_lm), ylab = "studentized residuals")
##
## ## -----
## #Problem 1d
## #Perform anova
## anova(mouse_lm)
##
## #Collecting variables for computation
## alpha = 0.05
## dfError = 12
## MSE = anova(mouse_lm)$`Mean Sq`[2]
## SE = sqrt(MSE)
## n = 3
## k = 6
##
## #LSD
## LSD = LSD = qt(1-alpha/2,dfError)*SE*sqrt(2/n)
##
## #Tukey's
## Tukey = qtkey(1-alpha,k,dfError)*SE*sqrt(1/n)
##
## #Hayter
## Hayter = (1/sqrt(2))*qtkey(1-alpha,k-1,dfError)*SE*sqrt(2/n)
##
## #Scheffe
## Scheffe = SE*sqrt(2/n)*sqrt((k-1)*qf(1-alpha,k-1,dfError))
##
## #Now make a table of the "yardsticks"
## yardsticks = c(LSD,Tukey,Hayter,Scheffe)
## names(yardsticks) <- c("LSD","Tukey's","Hayter","Scheffe")
## yardsticks
##
## #Make a table of means for each group
## labels <- c("Std","Std + E", "Std + C","Std + E + C","Std + 2C","Std + B6") %>% cbind(c(1,2,3,4,5,6))
## labels <- data.frame(labels) %>% rename(diet = V2)
## scoreMeans <- summarise(group_by(mouse,diet), mean = mean(score))

```

```

## scoreMeans <- inner_join(labels,scoreMeans, by = c('diet')) %>% select(-diet)
## kable(scoreMeans)
##
## ## -----
## #Problem 1e
## lht(lm(score~diet-1, data = mouse),hypothesis.matrix = c(1,0,-2,0,1,0))
##
## ## -----
## #Problem 1f
## #Create each of the contrasts
## contr = contrasts(mouse$diet)
## contr[,1] = c(1,1,-1,-1,0,0)
## contr[,2] = c(1,-1,1,-1,0,0)
## contr[,3] = c(1,-1,-1,1,0,0)
##
## #Perform the linear regression with contrasts
## mouse_lm2 <- lm(score ~ diet, data = mouse, contrasts = list(diet = contr))
## summary(mouse_lm2)
##
## ## -----
## #Problem 2
## F = 4.90
## alpha = 0.05
## df(F,3,50) < alpha
##
## ## -----
## #Problem 2
## #Calculate the pooled variance - we will use for the t-tests
## grandMean = sum((2/54)*(39.3),(25/54)*(40.1),(25/54)*(42.0),(2/54)*(43.0))
## MSR = sum((2/3)*(39.3-grandMean)^2,
##           (25/3)*(40.1-grandMean)^2,
##           (25/3)*(42.0-grandMean)^2,
##           (2/3)*(43.0-grandMean)^2)
## MSE = MSR/F
## SE = sqrt(MSE)
## harmonicN = 1/4*((1/2) + (1/25) + (1/25) + (1/2))
## LSD = qt(0.975,50)*SE*sqrt(2*harmonicN)
## paste("The Fisher's LSD to compare treatment mean differences against is ",round(LSD,2))
##
## ## -----
## #Problem 3c
## X = matrix(byrow = TRUE, nrow = 6, ncol = 4, data = c(1,1,0,0,
##               1,1,0,0,
##               1,1,0,0,
##               1,0,1,0,
##               1,0,1,0,
##               1,0,0,1))
## G1 = matrix(byrow = TRUE, nrow = 4, ncol = 4, data = c(0,0,0,0,
##               0,1/3,0,0,
##               0,0,1/2,0,
##               0,0,0,1))
## G2 = matrix(byrow = TRUE, nrow = 4, ncol = 4, data = c(1,-1,-1,0,
##               -1,4/3,1,0,

```

```

##              -1,1,3/2,0,
##              0,0,0,0))
## G3 = (1/18)*matrix(byrow = TRUE, nrow = 4, ncol = 4,
##                   data = c(0,2,3,6,
##                             0,4,-3,-6,
##                             0,-2,6,-6,
##                             0,-2,-3,12))
## G4 = (1/54)*matrix(byrow = TRUE, nrow = 4, ncol = 4,
##                   data = c(17,-11,-8,1,
##                             -11,23,2,-7,
##                             -8,2,26,-10,
##                             1,-7,-10,35))
##
## #Check to see if the matrices are is generalized inverses
## paste("G1 is a generalized inverse: ",(t(X)%*%X)%*%G1*(t(X)%*%X) %>% identical(t(X)%*%X))
## paste("G2 is a generalized inverse: ",(t(X)%*%X)%*%G2*(t(X)%*%X) %>% identical(t(X)%*%X))
## paste("G3 is a generalized inverse: ",(t(X)%*%X)%*%G3*(t(X)%*%X) %>% all.equal(t(X)%*%X))
## paste("G4 is a generalized inverse: ",(t(X)%*%X)%*%G4*(t(X)%*%X) %>% all.equal(t(X)%*%X))
##
## ## -----
## #Problem 3c
## Y = matrix(nrow = 6, ncol = 1, data = c(72,36,12,48,12,36))
## B1 = G1%*%t(X)%*%Y
## B2 = G2%*%t(X)%*%Y
## B3 = G3%*%t(X)%*%Y
## B4 = G4%*%t(X)%*%Y
## paste("mu: ",round(B1,2),
##       " ,alpha1: ", round(B2,2),
##       " ,alpha2: ", round(B3,2),
##       " ,alpha3: ", round(B4,2))
##
## ## -----
## #Problem 3d
## Y = matrix(nrow = 6, ncol = 1, data = c(39,52,99,12,36,48))
## B1 = G1%*%t(X)%*%Y
## B2 = G2%*%t(X)%*%Y
## B3 = G3%*%t(X)%*%Y
## B4 = G4%*%t(X)%*%Y
## paste("mu: ",round(B1,2),
##       " ,alpha1: ", round(B2,2),
##       " ,alpha2: ", round(B3,2),
##       " ,alpha3: ", round(B4,2))
##
## ## ----code = readLines(purl(this_file, documentation = 1)), echo = T, eval = F----
## ## # this R markdown chunk generates a code appendix
##

```