Unit3 HW

**Rashmi Patel-05/17/2021**

## HW Instructions

The weekly HW assignments are designed to accomplish 2 goals for the MSDS student. The first is to provide a series of conceptual and analtical questions so the student can get a feel for their current understanding of the unit. The second goal is to introduce the students to standard functions and routines in R that effectively do the same things that the “Procs” do in SAS.

R and SAS are both wonderful tools and as we go through the assignments, students will begin to recognize very quickly that they both have pros and cons.

The formatting of the HW is as follows:  
1. A series of high level questions will be asked with either short answers or simple multiple choice responses.  
2. Analytical questions will be provided but a short vignette example of how R functions work for a given topic or method will be given. The student will then be asked a follow up question or two based on the output provided.  
3. Thirdly, a new data set will be given to allow the student to gain some experience with a new data set from start to finish.

Solutions to the HW will be provided a day or two after the HW is submitted. It is up to the student to “shore up” any confusion or missunderstanding of a topic. Grading will be based on a combination of correctness, completion, and overall conciseness.

The student may provide there answers in a seperate word document. Just make sure that it is easy to follow and that all questions have been addressed for the grader. You are welcome to use R markdown, but it is not required.

## ANOVA Conceptual questions

1. State the necessary assumptions for Two Way ANOVA analysis to beconducted. Note: That addative versus non additivie is not a component of the assumptions

**The assumptions of Two-Way ANOVA analysis are:**

* **Constant Variance**
* **Normality**
* **Linearity**
* **Independence**
* **Outliers**

1. State in words what it means for there to be an “interaction” between two explanatory variables. Note: Do not explain the meaning in terms of a graph with parallel lines.

**The interaction term comes in the regression analysis when an independent variable shows different behavior/effect on the response depending on the other independent variable.**

1. What is the family wise error rate? What is multiple testing and why is it an issue when conducting ANOVA type models such as Two-Way ANOVA?

**The family wise error rate occurs when a series of hypothesis tests performed on the data. The family wise error rate refers to the type I error in series of hypothesis test. A multiple testing is referred to a situation where we are trying to perform n number of hypothesis test where n>1, then the probability of making Type I error increases. The multiple testing can be an issue because it may lead to bias. To avoid this multiple testing issue, we use multiple comparison corrections. Few of them are Bonferroni, Tukey’s HSD etc.**

1. True or False? The overall Type-III sums of squares F-tests allow the analyst to determine where specific differences lie between levels of the factor. **True**

## Exercise #1 ACT Scores Revisited

The first step in any analysis is appropriately describing the data both numerically and visually. For a Two Way ANOVA analysis, one of the most helpful visual tools is the mean profile plot (with or without the raw data). The following code reads in the ACT data set from our pre live discussion and provides a handy, modifiable, function that can make a quick summary statistics table really quick.

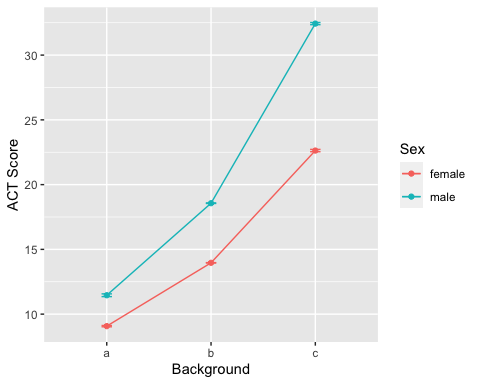
setwd("~/Desktop/MSDS\_NEW/ZippedFiles/Unit3/Unit3PreLive")  
  
ACT<-read.csv("MathACT\_2.csv")  
  
  
#Attaching the data set, creating a function, and creating a summary stats table. Note: In line 44 below, you can add other statistics like median, IQR,etc.  
  
attach(ACT)  
mysummary<-function(x){  
 result<-c(length(x),mean(x),sd(x),sd(x)/length(x))  
 names(result)<-c("N","Mean","SD","SE")  
 return(result)  
}  
sumstats<-aggregate(Score~Background\*Sex,data=ACT,mysummary)  
sumstats<-cbind(sumstats[,1:2],sumstats[,-(1:2)])  
sumstats

## Background Sex N Mean SD SE  
## 1 a female 82 9.073171 4.186340 0.05105293  
## 2 b female 387 13.963824 5.000905 0.01292224  
## 3 c female 54 22.629630 4.849806 0.08981122  
## 4 a male 48 11.458333 5.086312 0.10596483  
## 5 b male 223 18.565022 4.888305 0.02192065  
## 6 c male 67 32.432836 5.554752 0.08290675

With the three levels of background and two levels of sex status, the table provides the sample size, mean, standard deviation, and the means standard error for each of the 6 combinations of the two factors combined. This can be used to take a quick look at the data to see if things are making sense. Adding additional summaries like the max, min, and quartiles would be heplful as well.

The above table may not be too aesthetically pleasing. Luckily under the current format of the table, its quite easy to generate a means profile plot to visualize the data. This graphic was most likely a major point of discussion during live session.

library(ggplot2)  
ggplot(sumstats,aes(x=Background,y=Mean,group=Sex,colour=Sex))+  
 ylab("ACT Score")+  
 geom\_line()+  
 geom\_point()+  
 geom\_errorbar(aes(ymin=Mean-SE,ymax=Mean+SE),width=.1)



**HOMEWORK QUESTION**

1. Modify the previous R script so that the summary table also includeds the min, the max, and IQR. These functions are all self explanatory…min(x), max(x), IQR(x).

mysummary<-function(x){

result<-c(length(x),mean(x),sd(x),sd(x),min(x),max(x),IQR(x)/length(x))

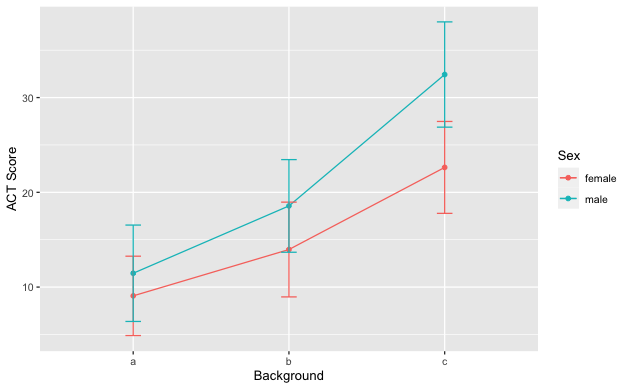
names(result)<-c("N","Mean","SD","SE","Minimum","Maximum","IQR")

return(result)

}

1. Create another means plot but rather than using the standard errors (SE) to make the error bars. Make it with the raw standard deviations (SD). Which graphic (compared to plot using SE) is more telling about the assumption of equal variances for the ANOVA model? Give a little explanation for your answer.

**The plot with raw standard deviations is giving more information about the equal standard deviations. Though the score are different and getting higher from a to b to c but the scores within each group for both male and females seems to have equal variances.**

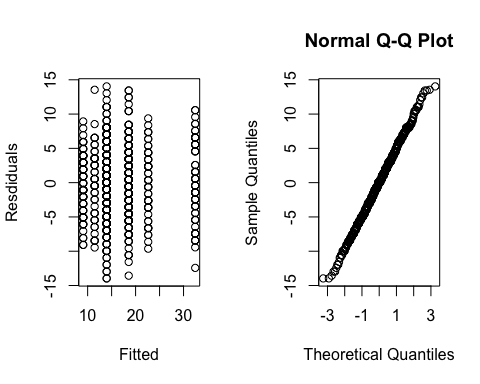


## Exercise #2 Conducting a Two Way ANOVA Analysis in R

Since Two Way ANOVA’s are techically just special cases of multiple linear regression, it’s not to suprising that the same function call is used to build the model. After viewing and exploring the data via Exercise 1. The next step would be to fit a full nonaddative model, check the assumptions of the model, and then examine the type III sums of squares F tables.

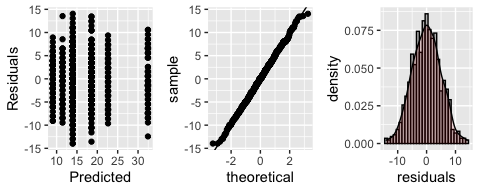
The following code fits the nonadditive two way anova model and then produces the first the main residual diagnostics for assumption checking. The syntax for including interaction terms is slightly different so please make note.

model.fit<-aov(Score~Background+Sex+Background:Sex,data=ACT)  
par(mfrow=c(1,2))  
plot(model.fit$fitted.values,model.fit$residuals,ylab="Resdiduals",xlab="Fitted")  
qqnorm(model.fit$residuals)



The previous graphics are not very pretty. We can use the ggplot2 package to jazz things up a bit.

library(gridExtra)  
myfits<-data.frame(fitted.values=model.fit$fitted.values,residuals=model.fit$residuals)  
  
#Residual vs Fitted  
plot1<-ggplot(myfits,aes(x=fitted.values,y=residuals))+ylab("Residuals")+  
 xlab("Predicted")+geom\_point()  
  
#QQ plot of residuals #Note the diagonal abline is only good for qqplots of normal data.  
plot2<-ggplot(myfits,aes(sample=residuals))+  
 stat\_qq()+geom\_abline(intercept=mean(myfits$residuals), slope = sd(myfits$residuals))  
  
#Histogram of residuals  
plot3<-ggplot(myfits, aes(x=residuals)) +   
 geom\_histogram(aes(y=..density..),binwidth=1,color="black", fill="gray")+  
 geom\_density(alpha=.1, fill="red")  
  
grid.arrange(plot1, plot2,plot3, ncol=3)



As discussed in class, the residual diagnostics do not provide any concern about the assumptions of a two way anova analysis. If there were, we would have to address those concerns via a transformation of the response or multiple analysis with and without outliers, etc. Examining the type-III sums of squares F table we have:

library(car)

## Loading required package: carData

Anova(model.fit,type=3)

## Anova Table (Type III tests)  
##   
## Response: Score  
## Sum Sq Df F value Pr(>F)   
## (Intercept) 6750.4 1 276.4610 < 2.2e-16 \*\*\*  
## Background 6007.9 2 123.0263 < 2.2e-16 \*\*\*  
## Sex 172.2 1 7.0542 0.008055 \*\*   
## Background:Sex 919.0 2 18.8192 1.004e-08 \*\*\*  
## Residuals 20876.8 855   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

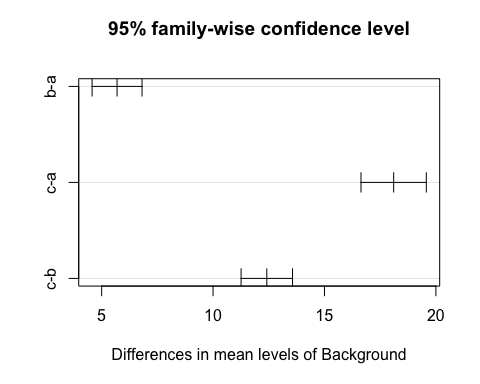
Writing contrasts are a little more cumbersome in R. To help you guys out and alleviate the need to keep track of all of the zero’s and one’s, I’ve wrote a little script that allows you to just specify the contrast that you want in a slightly simpler way. But first lets use some tools that provides a blanket lists of comparisons. Since there is no significant interaction, we just need to examine each factor one at a time. To examine all pairwise comparisons for say “background”, the following script provides the t-test results adjusted for multiple tests using Tukey’s procedure.

TukeyHSD(model.fit,"Background",conf.level=.95)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = Score ~ Background + Sex + Background:Sex, data = ACT)  
##   
## $Background  
## diff lwr upr p adj  
## b-a 5.692055 4.571357 6.812754 0  
## c-a 18.104005 16.638518 19.569492 0  
## c-b 12.411950 11.257405 13.566494 0

The table is helpful for quickly examining the results and getting the p-values and estimates. Its always helpful to visualize.

plot(TukeyHSD(model.fit,"Background",conf.level=.95))

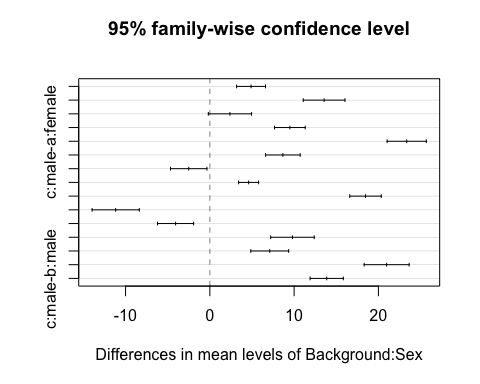


If an interaction is present, you can rinse and repeat the code just using the interaction term instead. This code below is for illustration, it makes no sense to do this on the ACT data set since the interaction F test is not significant.

TukeyHSD(model.fit,"Background:Sex",conf.level=.95)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = Score ~ Background + Sex + Background:Sex, data = ACT)  
##   
## $`Background:Sex`  
## diff lwr upr p adj  
## b:female-a:female 4.890654 3.1748785 6.606429 0.0000000  
## c:female-a:female 13.556459 11.0830154 16.029902 0.0000000  
## a:male-a:female 2.385163 -0.1797967 4.950122 0.0854058  
## b:male-a:female 9.491852 7.6691025 11.314601 0.0000000  
## c:male-a:female 23.359665 21.0354027 25.683927 0.0000000  
## c:female-b:female 8.665805 6.6155720 10.716039 0.0000000  
## a:male-b:female -2.505491 -4.6652479 -0.345734 0.0123022  
## b:male-b:female 4.601198 3.4146282 5.787768 0.0000000  
## c:male-b:female 18.469012 16.6014652 20.336558 0.0000000  
## a:male-c:female -11.171296 -13.9710443 -8.371548 0.0000000  
## b:male-c:female -4.064607 -6.2051648 -1.924050 0.0000011  
## c:male-c:female 9.803206 7.2221661 12.384246 0.0000000  
## b:male-a:male 7.106689 4.8610087 9.352369 0.0000000  
## c:male-a:male 20.974502 18.3056335 23.643371 0.0000000  
## c:male-b:male 13.867813 11.9015327 15.834094 0.0000000

plot(TukeyHSD(model.fit,"Background:Sex",conf.level=.95))



As discussed in class, including all possible combinations of comparisons may be too much and of little interest to the actual study at hand. We can manually create the comparisons of interest and manual adjust the p-values through writing contrasts. To help streamline this for you guys, I’ve included a little R script that makes the process a little more automated for you.

The following script allow you to write out your contrasts in a more verbal syntax. I’ll run you through the most tedious scenario. The script can be easily modified to handle simpler situations. First things first, all you need to do is provide some details as to what comparisons you’d like to make. Suppose, that if the interaction was significant, the only meaningful comparisons to make in the analysis comparing males versus females for each level of background.

library(lsmeans) #maybe need eemeans package

## Loading required package: emmeans

## The 'lsmeans' package is now basically a front end for 'emmeans'.  
## Users are encouraged to switch the rest of the way.  
## See help('transition') for more information, including how to  
## convert old 'lsmeans' objects and scripts to work with 'emmeans'.

contrast.factor<-~Background\*Sex  
mycontrast<-c("amale-afemale","bmale-bfemale","cmale-cfemale")  
dat<-ACT

The above piece of code provides no output, but formats things for the following code to run. The key player here is the “contrast.factor” and the “mycontrast” objects. The contrast.factor piece is just specifiying what types of comparisons you would like to make. For example, if we only wanted to compare the background levels we would have just specified “~Background”. The “mycontrast” object is where you get to specify what comparisons you would like to make. For a single factor, you just simply write out the factor levels you want to compare with a subtration between them. For an interaction type comparison the syntax depends on what was used in the contrast.factor object. In our example, background is listed first, so when making comparisons the levels of background are concatenated to the levels of Sex before subtracting which combinations you want to compare.

The following code is something I wrote that takes the information you specified above and creates a clean table of resutls with bonferroni adjusted p-values. This script can be reused over and over, just changing the initial starting script is all that is required.

#Running a loop that determines the appropriate 0's and 1's for each   
#contrast specified above.  
library(limma)  
final.result<-c()  
for( j in 1:length(mycontrast)){  
contrast.factor.names<-gsub(" ", "", unlist(strsplit(as.character(contrast.factor),split = "\*", fixed = T))[-1])  
contrast.factor.2 <- vector("list", length(contrast.factor.names))  
for (i in 1:length(contrast.factor.names)) {  
 contrast.factor.2[[i]] <- levels(dat[, contrast.factor.names[i]])  
}  
new.factor.levels <- do.call(paste, c(do.call(expand.grid,   
 contrast.factor.2), sep = ""))  
temp.cont<-mycontrast[j]  
contrast2 <- list(comparison = as.vector(do.call(makeContrasts,   
 list(contrasts = temp.cont, levels = new.factor.levels))))  
  
contrast.result <- summary(contrast(lsmeans(model.fit,   
 contrast.factor), contrast2, by = NULL))  
  
final.result<-rbind(final.result,contrast.result)  
}  
#Cleaning up and applying bonferroni correction to the number  
#of total comparisons investigated.  
final.result$contrast<-mycontrast  
final.result$bonf<-length(mycontrast)\*final.result$p.value  
final.result$bonf[final.result$bonf>1]<-1  
  
final.result

## contrast estimate SE df t.ratio p.value bonf  
## amale-afemale 2.39 0.898 855 2.656 0.0081 0.0242  
## bmale-bfemale 4.60 0.415 855 11.076 <.0001 0.0000  
## cmale-cfemale 9.80 0.904 855 10.848 <.0001 0.0000

**HOMEWORK QUESTION**

1. Consider comparing the mean ACT scores of males versus females specifically for background A. Compare the outputs from the Tukey comparison result table to that of the output generated from my manual contrast maker. Is the estimated differences the same? Can you explain why are the adjusted p-values different for the two result tables? One would suggest that we reject the null while the other would have us to fail to reject. (This is just a conceptual thinking question. The interaction term is not significant for this data analysis.)

**Yes, the estimated differences are the same. It may be possible that by considering specific interaction from the model results in change of the p-values but remain significant/or insignificant.**

## Exercise #3

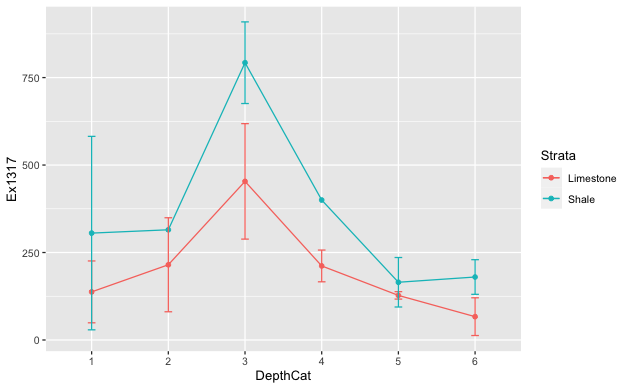
Lets examine the dta Exercise 13.17 from the statistical sleuth book. The data set is easily accesable in R via the following package.

library(Sleuth3)  
head(ex1317)

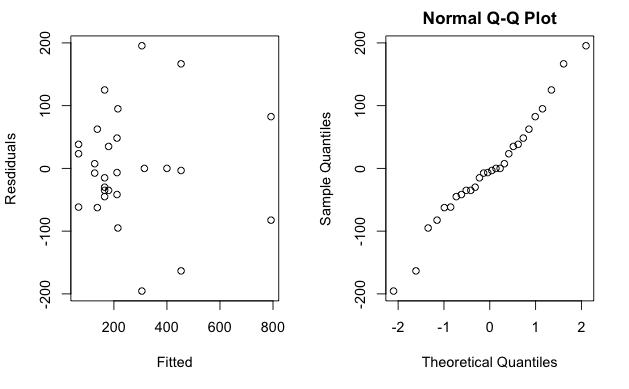
## Iridium Strata DepthCat  
## 1 75 Limestone 1  
## 2 200 Limestone 1  
## 3 120 Limestone 2  
## 4 310 Limestone 2  
## 5 290 Limestone 3  
## 6 450 Limestone 3

1. Provide a means plot of the data. Use this along with any additional information to comment on whether an addative or nonadditive model is probably the most appropriated. If it is not obvious that is okay just do your best.

**Looking at the graph, I think a non-additive model will be most appropriate.**



1. Fit a nonadditive 2 way anova model to the data set and provide the residual diagnostics. Comment on the appropriateness of the current anova fit.



**Looking at the residuals plot, it seems that data has good normality with some right skewness.**

1. Provide the type 3 ANOVA F-tests. Answer the following question using the table. Do the potential changes in mean Iridium by strata depend on the depth?

**It seems the mean Iridium change depend on depth since the table shows that the depth is significant.**

Anova Table (Type III tests)

Response: Iridium

Sum Sq Df F value Pr(>F)

(Intercept) 37812 1 2.9821 0.10344

DepthCat 263448 5 4.1554 0.01306 \*

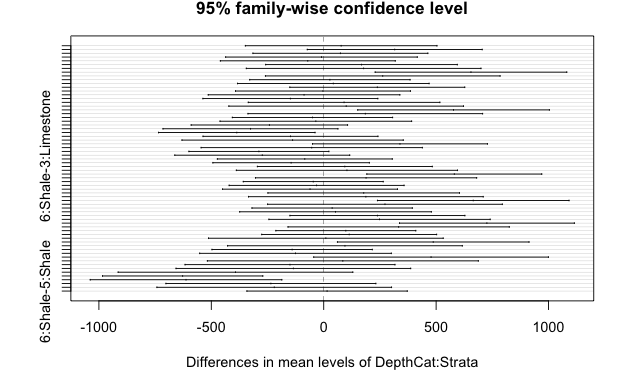
Strata 28224 1 2.2259 0.15517

DepthCat:Strata 65523 5 1.0335 0.43144

Residuals 202878 16

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

1. Using multple testing techniques, determine what factors (or combinations) contribute to changes in mean iridium.

Tukey multiple comparisons of means

95% family-wise confidence level

Fit: aov(formula = Iridium ~ DepthCat + Strata + DepthCat:Strata, data = ex1317)

$`DepthCat:Strata`

diff lwr upr p adj

2:Limestone-1:Limestone 77.500000 -348.62306 503.62306 0.9998289

3:Limestone-1:Limestone 315.833333 -73.16202 704.82869 0.1731727

4:Limestone-1:Limestone 74.166667 -314.82869 463.16202 0.9997334

5:Limestone-1:Limestone -10.000000 -436.12306 416.12306 1.0000000

6:Limestone-1:Limestone -70.833333 -459.82869 318.16202 0.9998270

1:Shale-1:Limestone 168.000000 -258.12306 594.12306 0.9227263

2:Shale-1:Limestone 177.500000 -344.39203 699.39203 0.9694283

3:Shale-1:Limestone 655.000000 228.87694 1081.12306 0.0011042

4:Shale-1:Limestone 262.500000 -259.39203 784.39203 0.7427456

5:Shale-1:Limestone 27.500000 -329.02013 384.02013 1.0000000

6:Shale-1:Limestone 42.500000 -383.62306 468.62306 0.9999996

3:Limestone-2:Limestone 238.333333 -150.66202 627.32869 0.5014783

4:Limestone-2:Limestone -3.333333 -392.32869 385.66202 1.0000000

5:Limestone-2:Limestone -87.500000 -513.62306 338.62306 0.9994717

6:Limestone-2:Limestone -148.333333 -537.32869 240.66202 0.9365358

1:Shale-2:Limestone 90.500000 -335.62306 516.62306 0.9992833

2:Shale-2:Limestone 100.000000 -421.89203 621.89203 0.9997208

3:Shale-2:Limestone 577.500000 151.37694 1003.62306 0.0039761

4:Shale-2:Limestone 185.000000 -336.89203 706.89203 0.9597980

5:Shale-2:Limestone -50.000000 -406.52013 306.52013 0.9999865

6:Shale-2:Limestone -35.000000 -461.12306 391.12306 0.9999999

4:Limestone-3:Limestone -241.666667 -589.59469 106.26135 0.3388347

5:Limestone-3:Limestone -325.833333 -714.82869 63.16202 0.1475457

6:Limestone-3:Limestone -386.666667 -734.59469 -38.73865 0.0228190

1:Shale-3:Limestone -147.833333 -536.82869 241.16202 0.9378144

2:Shale-3:Limestone -138.333333 -630.37786 353.71119 0.9923714

3:Shale-3:Limestone 339.166667 -49.82869 728.16202 0.1184980

4:Shale-3:Limestone -53.333333 -545.37786 438.71119 0.9999990

5:Shale-3:Limestone -288.333333 -599.52962 22.86295 0.0826092

6:Shale-3:Limestone -273.333333 -662.32869 115.66202 0.3247402

5:Limestone-4:Limestone -84.166667 -473.16202 304.82869 0.9991533

6:Limestone-4:Limestone -145.000000 -492.92802 202.92802 0.8945388

1:Shale-4:Limestone 93.833333 -295.16202 482.82869 0.9978151

2:Shale-4:Limestone 103.333333 -388.71119 595.37786 0.9993523

3:Shale-4:Limestone 580.833333 191.83798 969.82869 0.0014998

4:Shale-4:Limestone 188.333333 -303.71119 680.37786 0.9350909

5:Shale-4:Limestone -46.666667 -357.86295 264.52962 0.9999737

6:Shale-4:Limestone -31.666667 -420.66202 357.32869 1.0000000

6:Limestone-5:Limestone -60.833333 -449.82869 328.16202 0.9999602

1:Shale-5:Limestone 178.000000 -248.12306 604.12306 0.8932100

2:Shale-5:Limestone 187.500000 -334.39203 709.39203 0.9561647

3:Shale-5:Limestone 665.000000 238.87694 1091.12306 0.0009386

4:Shale-5:Limestone 272.500000 -249.39203 794.39203 0.7020689

5:Shale-5:Limestone 37.500000 -319.02013 394.02013 0.9999993

6:Shale-5:Limestone 52.500000 -373.62306 478.62306 0.9999964

1:Shale-6:Limestone 238.833333 -150.16202 627.82869 0.4986989

2:Shale-6:Limestone 248.333333 -243.71119 740.37786 0.7391657

3:Shale-6:Limestone 725.833333 336.83798 1114.82869 0.0001221

4:Shale-6:Limestone 333.333333 -158.71119 825.37786 0.3700239

5:Shale-6:Limestone 98.333333 -212.86295 409.52962 0.9816836

6:Shale-6:Limestone 113.333333 -275.66202 502.32869 0.9899559

2:Shale-1:Shale 9.500000 -512.39203 531.39203 1.0000000

3:Shale-1:Shale 487.000000 60.87694 913.12306 0.0182200

4:Shale-1:Shale 94.500000 -427.39203 616.39203 0.9998359

5:Shale-1:Shale -140.500000 -497.02013 216.02013 0.9229136

6:Shale-1:Shale -125.500000 -551.62306 300.62306 0.9890968

3:Shale-2:Shale 477.500000 -44.39203 999.39203 0.0892737

4:Shale-2:Shale 85.000000 -517.62901 687.62901 0.9999857

5:Shale-2:Shale -150.000000 -616.79442 316.79442 0.9793521

6:Shale-2:Shale -135.000000 -656.89203 386.89203 0.9960834

4:Shale-3:Shale -392.500000 -914.39203 129.39203 0.2472265

5:Shale-3:Shale -627.500000 -984.02013 -270.97987 0.0002436

6:Shale-3:Shale -612.500000 -1038.62306 -186.37694 0.0022197

5:Shale-4:Shale -235.000000 -701.79442 231.79442 0.7417976

6:Shale-4:Shale -220.000000 -741.89203 301.89203 0.8878615

6:Shale-5:Shale 15.000000 -341.52013 371.52013 1.0000000

**Looking at the table obtained by implementing TukeyHSD multiplecomparison, I found that below interaction(combimation) are contributing to changes in mean Iridium.**

**3:Shale-6:Limestone**

**5:Shale-3:Shale**

**6:Shale-3:Shale**

**3:Shale-1:Shale**

**3:Shale-5:Limestone**

**3:Shale-4:Limestone**

**3:Shale-2:Limestone**

**3:Shale-1:Limestone**