Unit3 HW

Turner

12/16/2018

## 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 dependent variable is continuous
* Independent variable are categorical (if not, then this is a multiple linear regression problem)
* Independence - within and amongts groups
* Equal variance amongst groups
* Groups are normally distributed

# 2. 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.

* An interaction between variables accounts for their combined effect on the dependent variable. An example could be age and sex. If one had three categories of age (child, adult, elderly) and 2 for sex (M/F) - one could examine if the combination of age/sex has an influence on the dependent variable by including interaction terms for each categorical combination.

# 3. 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 familywise error rate is the likelihood of getting a false conclusion in a series of hypotheis tests. In an ANOVA test where there are no pre-planned comparisons (i.e all comparisons and contrasts are run) - the tests are run as a family and there is a corresponding increase in the type 1 error rate. Correction methods, such as Bonferroni must be used in order to account for the widening of confidence intervals.

# 4. True or False? The overall Type-III sums of squares F-test’s allow the analyst to determine where specific differences lie between levels of the factor.

- False - this test is for determining whether or not there is a signifcant effect from interaction terms. Contrasts should be performed to determine differences between specifical levels of a factor.

## 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("D:/MSDS6372/HWMark")  
  
ACT<-read.csv("MathACT.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), min(x), max(x), IQR(x))  
 names(result)<-c("N","Mean","SD","SE","Min", "Max", "IQR")  
 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 Min Max IQR  
## 1 a female 82 9.073171 4.186340 0.05105293 0 18 6.0  
## 2 b female 387 13.963824 5.000905 0.01292224 0 28 6.0  
## 3 c female 54 24.629630 4.849806 0.08981122 15 34 7.0  
## 4 a male 48 11.458333 5.086312 0.10596483 2 25 8.0  
## 5 b male 223 15.565022 4.888305 0.02192065 2 29 7.0  
## 6 c male 67 25.432836 5.554752 0.08290675 13 36 8.5

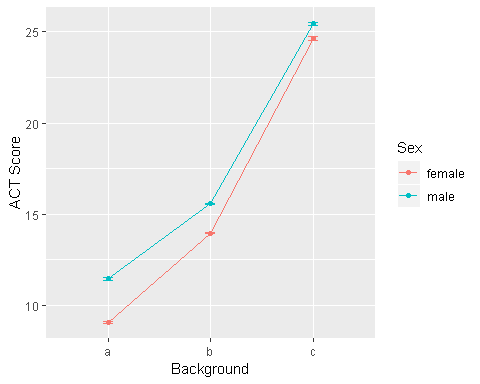
With the three levels of background and two levels of sex status, the table provides the sample size, mean, standard deviation, and the means standard error for each of the 6 combinations of the two factors combined. This can be used to take a quick look at the data to see if things are making sense. 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)

## Registered S3 methods overwritten by 'ggplot2':  
## method from   
## [.quosures rlang  
## c.quosures rlang  
## print.quosures rlang

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**

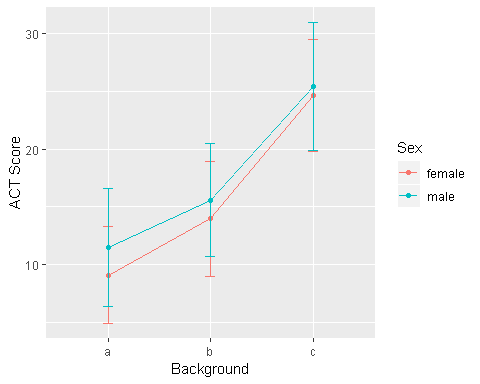
1. Modify the previous R script so that the summary table also includeds the min, the max, and IQR. These functions are all self explanatory…min(x), max(x), IQR(x).

mysummary<-function(x){  
 result<-c(length(x),mean(x),sd(x),sd(x)/length(x), min(x), max(x), IQR(x))  
 names(result)<-c("N","Mean","SD","SE","Min", "Max", "IQR")  
 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 Min Max IQR  
## 1 a female 82 9.073171 4.186340 0.05105293 0 18 6.0  
## 2 b female 387 13.963824 5.000905 0.01292224 0 28 6.0  
## 3 c female 54 24.629630 4.849806 0.08981122 15 34 7.0  
## 4 a male 48 11.458333 5.086312 0.10596483 2 25 8.0  
## 5 b male 223 15.565022 4.888305 0.02192065 2 29 7.0  
## 6 c male 67 25.432836 5.554752 0.08290675 13 36 8.5

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.

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



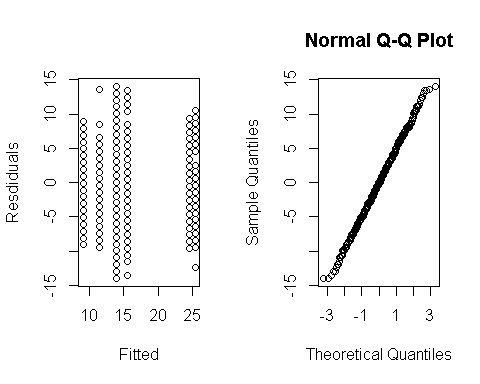
### The standard error variant is impossible to interpret because the values are adjusted for sample size. Standard deviation is not adusted for sample size and hence is on an order of magnitude similar to the response variable (ACT Score)

## 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)

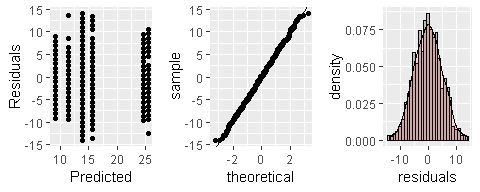


summary(model.fit)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Background 2 15619 7809 319.826 < 2e-16 \*\*\*  
## Sex 1 517 517 21.158 4.87e-06 \*\*\*  
## Background:Sex 2 38 19 0.771 0.463   
## Residuals 855 20877 24   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

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 8045.8 2 164.7564 < 2.2e-16 \*\*\*  
## Sex 172.2 1 7.0542 0.008055 \*\*   
## Background:Sex 37.6 2 0.7709 0.462898   
## Residuals 20876.8 855   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

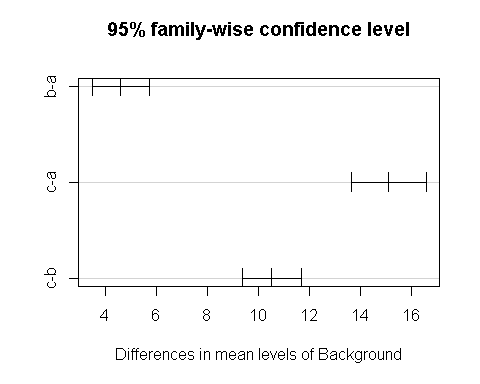
Writing contrasts are a little more cumbersome in R. To help you guys out and alleviate the need to keep track of all of the zero’s and one’s, I’ve wrote a little script that allows you to just specify the contrast that you want in a slightly simpler way. But first lets use some tools that provides a 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 4.595334 3.474636 5.716033 0  
## c-a 15.120534 13.655047 16.586021 0  
## c-b 10.525200 9.370655 11.679744 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))

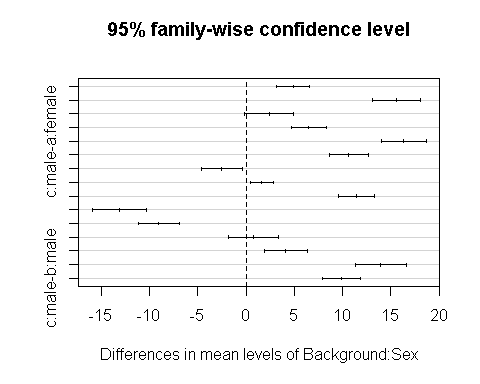


If an interaction is present, you can rinse and repeat the code just using the interaction term instead. This code below is for illustration, it makes no sense to do this on the ACT data set since the interaction F test is not significant.

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 4.8906536 3.1748785 6.606429 0.0000000  
## c:female-a:female 15.5564589 13.0830154 18.029902 0.0000000  
## a:male-a:female 2.3851626 -0.1797967 4.950122 0.0854058  
## b:male-a:female 6.4918517 4.6691025 8.314601 0.0000000  
## c:male-a:female 16.3596651 14.0354027 18.683927 0.0000000  
## c:female-b:female 10.6658053 8.6155720 12.716039 0.0000000  
## a:male-b:female -2.5054910 -4.6652479 -0.345734 0.0123022  
## b:male-b:female 1.6011981 0.4146282 2.787768 0.0017335  
## c:male-b:female 11.4690115 9.6014652 13.336558 0.0000000  
## a:male-c:female -13.1712963 -15.9710443 -10.371548 0.0000000  
## b:male-c:female -9.0646072 -11.2051648 -6.924050 0.0000000  
## c:male-c:female 0.8032062 -1.7778339 3.384246 0.9492008  
## b:male-a:male 4.1066891 1.8610087 6.352369 0.0000033  
## c:male-a:male 13.9745025 11.3056335 16.643371 0.0000000  
## c:male-b:male 9.8678134 7.9015327 11.834094 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(emmeans) #maybe need eemeans package

## Warning: package 'emmeans' was built under R version 3.6.1

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 2.385 0.898 855 2.656 0.0081 0.024165  
## bmale-bfemale 1.601 0.415 855 3.854 0.0001 0.000374  
## cmale-cfemale 0.803 0.904 855 0.889 0.3743 1.000000

**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.)

# According to the TukeHSD - the adjusted p-value is 0.0854 (not signifcant at an alpha level of 0.05) and is alread adjusted for the family-wise error rate. The contrast - on the other hand - is not adjusted for multiple comparisons though the bonferroni adjustment would account for the multiple comparison error rate (0.0081 non-adjusted pvalue vs 0.024165 bf adjusted)

# Tukey (Fail to Reject)

a:male-a:female 2.3851626 -0.1797967 4.950122 0.0854058

# Contrast

amale-afemale 2.385 0.898 855 2.656 0.0081 0.024165

## 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)

## Warning: package 'Sleuth3' was built under R version 3.6.1

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

attach(ex1317)

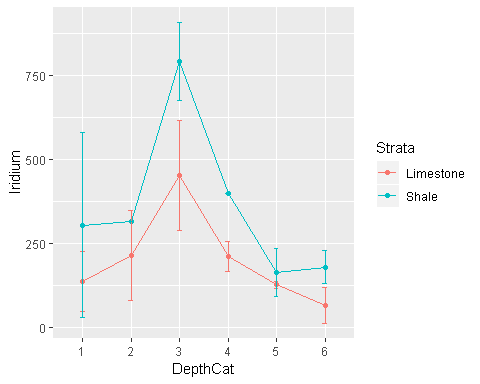
1. Provide a means plot of the data. Use this along with any additional information to comment on whether an addative or nonadditive model is probably the most appropriated. If it is not obvious that is okay just do your best.

# The dataset appears to be non-linear. Means for both strata show a spike at a DepthCat of 3 and a tail off after that value. Despite the fact that both strata appear are nonlinear in their reltaionship to to depth - they both also appear to share the same slopes, hence an additive model seems most appropriate (i.e interaction effects should be ingnored)

ex1317.summary<-aggregate(Iridium~Strata\*DepthCat,data=ex1317,mysummary)  
ex1317.summary<-cbind(ex1317.summary[,1:2],ex1317.summary[,-(1:2)])  
ex1317.summary

## Strata DepthCat N Mean SD SE Min Max IQR  
## 1 Limestone 1 2 137.50000 88.38835 44.194174 75 200 62.5  
## 2 Shale 1 2 305.50000 276.47875 138.239376 110 501 195.5  
## 3 Limestone 2 2 215.00000 134.35029 67.175144 120 310 95.0  
## 4 Shale 2 1 315.00000 NA NA 315 315 0.0  
## 5 Limestone 3 3 453.33333 165.02525 55.008417 290 620 165.0  
## 6 Shale 3 2 792.50000 116.67262 58.336309 710 875 82.5  
## 7 Limestone 4 3 211.66667 45.36886 15.122953 170 260 45.0  
## 8 Shale 4 1 400.00000 NA NA 400 400 0.0  
## 9 Limestone 5 2 127.50000 10.60660 5.303301 120 135 7.5  
## 10 Shale 5 5 165.00000 70.71068 14.142136 120 290 20.0  
## 11 Limestone 6 3 66.66667 53.92897 17.976322 5 105 50.0  
## 12 Shale 6 2 180.00000 49.49747 24.748737 145 215 35.0

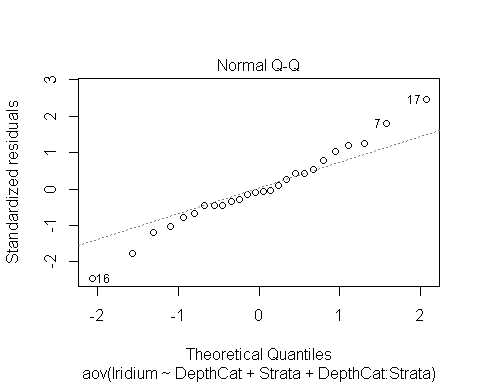
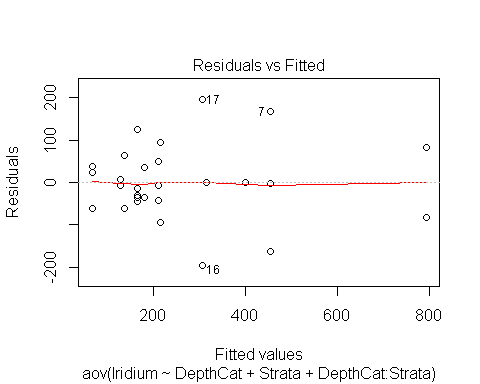
ex1317.summary[is.na(ex1317.summary)] = 0  
ggplot(ex1317.summary,aes(x=DepthCat,y=Mean,group=Strata,colour=Strata))+  
 ylab("Iridium")+  
 geom\_line()+  
 geom\_point()+  
 geom\_errorbar(aes(ymin=Mean-SD,ymax=Mean+SD),width=.1)



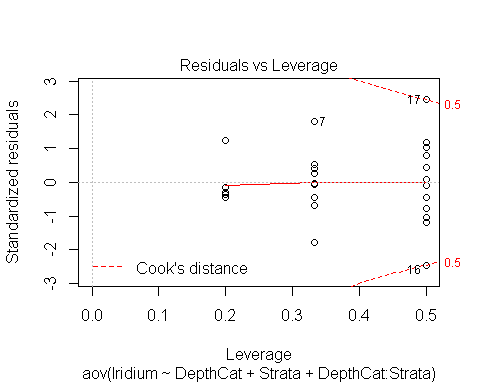
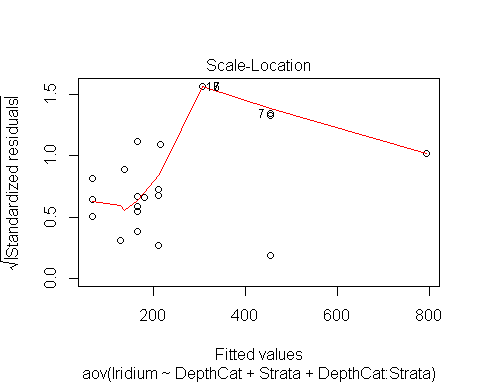
1. Fit a nonadditive 2 way anova model to the data set and provide the residual diagnostics. Comment on the appropriateness of the current anova fit.

anova.model = aov(Iridium~DepthCat+Strata+DepthCat:Strata, data=ex1317)  
plot(anova.model)

## Warning: not plotting observations with leverage one:  
## 18, 21



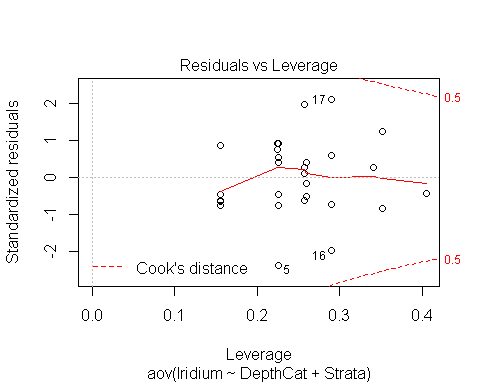
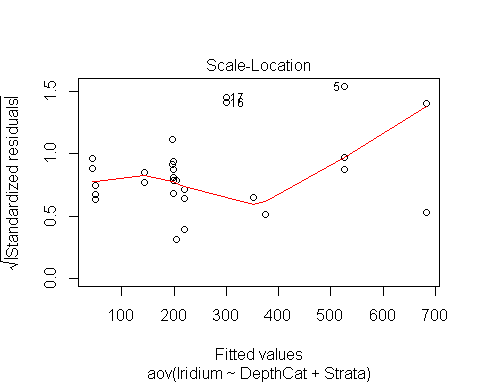
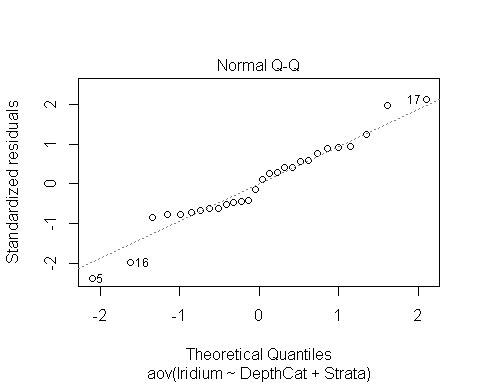
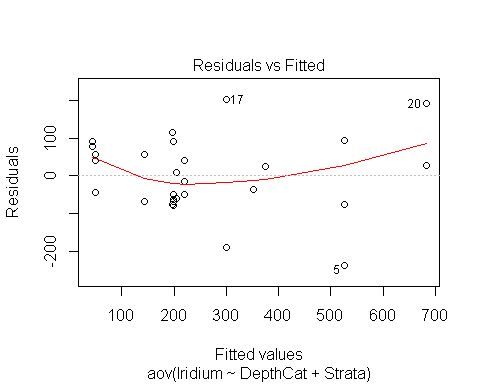
## Warning: not plotting observations with leverage one:  
## 18, 21



summary(anova.model)

## Df Sum Sq Mean Sq F value Pr(>F)   
## DepthCat 5 735267 147053 11.597 7.34e-05 \*\*\*  
## Strata 1 151433 151433 11.943 0.00325 \*\*   
## DepthCat:Strata 5 65523 13105 1.033 0.43144   
## Residuals 16 202878 12680   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

anova.model\_add = aov(Iridium~DepthCat+Strata, data=ex1317)  
plot(anova.model\_add)



summary(anova.model\_add)

## Df Sum Sq Mean Sq F value Pr(>F)   
## DepthCat 5 735267 147053 11.51 1.93e-05 \*\*\*  
## Strata 1 151433 151433 11.85 0.00244 \*\*   
## Residuals 21 268401 12781   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# The additive two way anova model provides an adequate fit (R-squared of 0.7676 on 21 df) - despite having a lower R-squared than the non-additive model, the adjusted r-squared values are nearly identical which indicates that the added variables are not significanlty contributing to the predictive power of the more complex model. The simplified model seems to capture the trends shown in the mean plots - Shale is a signficant indicator of Iridium level - as is DepthCat = 3. Both paramaters have p-values below 0.05.

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?

# Quite clearly, depth is an important factor, but so is Strata according to the F-tests. The p-value for DepthCat is extremely low - orders of magnitude lower than Strata. The interaction between the two variables is not significant though. Each Strata has a similar relationship w/r/t Iridium and DepthCat

model.anova = aov(Iridium~Strata+DepthCat, data=ex1317)  
Anova(model.anova, type = 3)

## Anova Table (Type III tests)  
##   
## Response: Iridium  
## Sum Sq Df F value Pr(>F)   
## (Intercept) 71141 1 5.5661 0.028073 \*   
## Strata 151433 1 11.8483 0.002444 \*\*   
## DepthCat 810293 5 12.6796 9.337e-06 \*\*\*  
## Residuals 268401 21   
## ---  
## 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.

## Multiple Contrasts

# TukeyHSD

TukeyHSD(model.anova,"DepthCat",conf.level=.95)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = Iridium ~ Strata + DepthCat, data = ex1317)  
##   
## $DepthCat  
## diff lwr upr p adj  
## 2-1 44.29060 -225.84123 314.42242 0.9950744  
## 3-1 377.97436 140.71486 615.23386 0.0007750  
## 4-1 63.43590 -186.65757 313.52937 0.9655520  
## 5-1 -89.65934 -311.34351 132.02483 0.7998163  
## 6-1 -99.02564 -336.28514 138.23386 0.7787278  
## 3-2 333.68376 75.38833 591.97919 0.0067517  
## 4-2 19.14530 -250.98652 289.27712 0.9999142  
## 5-2 -133.94994 -378.01618 110.11630 0.5362396  
## 6-2 -143.31624 -401.61167 114.97919 0.5248810  
## 4-3 -314.53846 -551.79796 -77.27896 0.0053046  
## 5-3 -467.63370 -674.73077 -260.53663 0.0000076  
## 6-3 -477.00000 -700.69040 -253.30960 0.0000177  
## 5-4 -153.09524 -374.77941 68.58893 0.2964359  
## 6-4 -162.46154 -399.72104 74.79796 0.3048292  
## 6-5 -9.36630 -216.46337 197.73077 0.9999907

TukeyHSD(model.anova,"Strata",conf.level=.95)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = Iridium ~ Strata + DepthCat, data = ex1317)  
##   
## $Strata  
## diff lwr upr p adj  
## Shale-Limestone 104.7436 15.654 193.8332 0.0233877

# Bonferroni Contrasts

contrast.factor<-~Strata\*DepthCat  
mycontrast<-c("Limestone1-Shale1","Limestone2-Shale2","Limestone3-Shale3", "Limestone4-Shale4", "Limestone5-Shale5", "Limestone6-Shale6")  
dat<-ex1317  
  
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.anova,   
 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  
## Limestone1-Shale1 -156 45.2 21 -3.442 0.0024 0.0147  
## Limestone2-Shale2 -156 45.2 21 -3.442 0.0024 0.0147  
## Limestone3-Shale3 -156 45.2 21 -3.442 0.0024 0.0147  
## Limestone4-Shale4 -156 45.2 21 -3.442 0.0024 0.0147  
## Limestone5-Shale5 -156 45.2 21 -3.442 0.0024 0.0147  
## Limestone6-Shale6 -156 45.2 21 -3.442 0.0024 0.0147

The differences in depth are all stastically significant for contrasts with level 3 indicating a strong correlation between a DepthCat = 3 and higher levels of Iridium.

There is also a difference by strata though not as pronounced. Shale appears to be associated with higher levels of Iridium.

contrast.factor<-~Strata\*DepthCat  
mycontrast<-c("Limestone1-Limestone3","Limestone2-Limestone3","Limestone4-Limestone3", "Limestone5-Limestone3", "Limestone6-Limestone3")  
dat<-ex1317  
  
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.anova,   
 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  
## Limestone1-Limestone3 -383 76.0 21 -5.042 0.0001 2.71e-04  
## Limestone2-Limestone3 -330 82.6 21 -3.998 0.0007 3.27e-03  
## Limestone4-Limestone3 -307 76.1 21 -4.031 0.0006 3.02e-03  
## Limestone5-Limestone3 -484 67.7 21 -7.143 <.0001 2.41e-06  
## Limestone6-Limestone3 -477 71.5 21 -6.671 <.0001 6.64e-06

contrast.factor<-~Strata\*DepthCat  
mycontrast<-c("Shale1-Shale3","Shale2-Shale3","Shale4-Shale3", "Shale5-Shale3", "Shale6-Shale3")  
dat<-ex1317  
  
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.anova,   
 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  
## Shale1-Shale3 -383 76.0 21 -5.042 0.0001 2.71e-04  
## Shale2-Shale3 -330 82.6 21 -3.998 0.0007 3.27e-03  
## Shale4-Shale3 -307 76.1 21 -4.031 0.0006 3.02e-03  
## Shale5-Shale3 -484 67.7 21 -7.143 <.0001 2.41e-06  
## Shale6-Shale3 -477 71.5 21 -6.671 <.0001 6.64e-06

# check against DepthCat2 - just to be sure

contrast.factor<-~Strata\*DepthCat  
mycontrast<-c("Limestone1-Limestone2","Limestone3-Limestone2","Limestone4-Limestone2", "Limestone5-Limestone2", "Limestone6-Limestone2")  
dat<-ex1317  
  
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.anova,   
 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  
## Limestone1-Limestone2 -52.8 86.7 21 -0.609 0.5490 1.00000  
## Limestone3-Limestone2 330.3 82.6 21 3.998 0.0007 0.00327  
## Limestone4-Limestone2 23.4 86.4 21 0.271 0.7893 1.00000  
## Limestone5-Limestone2 -153.4 79.9 21 -1.920 0.0686 0.34297  
## Limestone6-Limestone2 -146.7 82.6 21 -1.776 0.0903 0.45128

# As expected - only sig diff is 2 vs 3