Stats 158: Final Write-up

By: Andrey Mironyuk, Grace Deng, Anthony Cerna
Professor Purdom

I. Introduction

All of us have encountered a situation that demands we memorize a particular piece of information, but it's not always easy to do. We've all experienced this limit to memory. In fact, the number 7 has been documented as the magic limit for the number of items the average person can store in memory. That's sounds reasonable, until we need to memorize more. We decided to test several theories we heard and read about, to determine if they held any water. We used a 15 letter string and came up with two potential memory aids: music and grouping. Music refers to the presence of music, which people anecdotally cite as a great study aid. The other was blocking, which we applied to our experiment by grouping 3 letters or 5 letters together with a single color. Part of the rationale from this came from the fact that something similar is done to phone numbers. It's a technique known as chunking and we were excited to apply it to letters in an effort to improve memory retention. We hypothesized that either music, or grouping the letters in color blocks, would improve memory retention, but our experiment upheld the null hypothesis, that no condition had any significant effects.

II. Methodology

The experiment was run with a Replicated Latin squares design in order to control for two nuisance factors: subject-by-subject variability and variability in the difficulty between different strings of letters. Our **response** is the number of correctly remembered letters written down for each trial; a higher number of correct letters will correspond to better memorization. Our **factor of interest** was composed of the 6 levels of treatment combinations (Music **x** Color Grouping Aid). The **music factor** has 2 levels (no music or classical music) and the **grouping aid factor** has 3 levels (no grouping [black string of 15 letters], color-coded grouping by every 3 letters, or color-coded grouping by every 5 letters). For example:

XDTPSHOWQMBVCIL, DNXVHIPUZLTBFQY, KDISEZTBPRJOCNL

Our **two nuisance factors** were **subjects (rows of Latin Square)** and **strings (columns of Latin Square)**. At each trial, a subject will be given a string of 15 random letters, under one of 6 treatment combinations to memorize within 30 seconds for a total of six trials to cover six strings. A total of 12 subjects (72 trials) will be tested, which corresponds to two Latin Squares that was determined by our power calculations.

Randomization of treatment combinations was achieved by writing out an arbitrary 6x6 Latin Square, then using R to randomly permute the rows and columns of the Latin Square, the order of which was used to test subjects 1-6. A second Latin Square was created by using R to randomly permute just the rows of the first random Latin Square, since we were only changing the subjects, not the strings, for testing subjects 7-12. The letters A-F were then randomly assigned to correspond to one of the six treatment combinations.

After conducting trial runs, we decided to improvise by designing an answer sheet for each subject to write down as many letters as they can remember for each trial, which would save time and make it easier for testers to "grade" the number of correct letters. We will control for variability of strings by making all strings be capital letters with no repeating letters, with font size 30 and font type Bold in Times New Roman, no spacing between letters, and centered in the paper slip presented to the subject. We will control for the music condition by choosing the exact same 30 seconds of a single Youtube video of Fur Elise (from 0:07-0:37) by Ludwig van Beethoven, with the volume at 5 bars on an iPhone, and all subjects will be listening using the same Apple earbuds in both ears. For the factor of interest Color-coded Grouping, we will keep the colors consistent in the following order: Blue, Red, and Green for each x-number blocks, repeating as necessary.

For consistency, the proctor will start/stop the music, start/stop the timer, and flip over the paper that shows the string to memorize in each trial. Even if the subjects are doing a trial that consists of no music, they will still be required to keep the headphones on to keep the trails consistent.

In the actual experiment, each subject will be given six strips of paper, one at a time, with each strip of paper consisting of a string of letters randomly assigned to one of six treatment combinations to memorize for each trial. Each subject will be read the same set of instructions for consistency and given an answer sheet to write down their answers for each trial. The proctor with the timer will start the timer AND flip over the paper at the same time; the subject will begin memorizing. At 30 seconds, the timer will call time, and the paper will be removed from view. The proctor will start timing again and give the subject another 30 seconds to recall as many letters as possible by filling out the corresponding answer boxes.

Based on our power calculations, we realized that we do not have enough power with only 2 Latin Squares, however we chose to keep 2 Latin Squares due to time and subject constraint during section. Our power calculations have changed from the protocol and are now accurate. That being said, our 2 latin squares give us a power of .57, while 3 and 4 would have given us .80 and .92, respectively. This should be taken into account when interpreting our results.

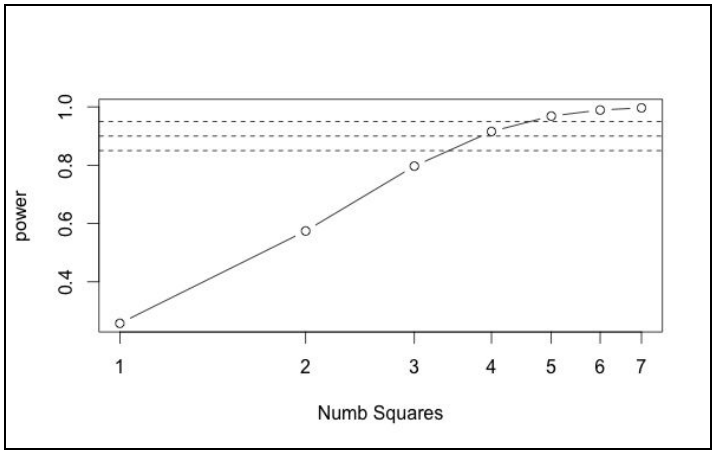

Figure 1

III. Results

A. Checking Assumptions

Before we analyze results, we will first check assumptions: Normality, Constant Variance, and Independence. Figures 4, 5, and 6 show a histogram of the responses, a QQ-plot, and the fitted residual plot, respectively. The histogram in Figure 4 appears unimodal, not symmetric, and left skewed due to many responses being the full score of 15. We have tried square root, log, and 1/y transformations, all of which were unable to correct the skewness. The QQ-plot shows data points that deviate from the designated quantiles and display a "bent" shape, further validating that the data is not normal. The fitted residual plot shows no obvious shape, appearing to be mostly random scatter symmetric about the horizontal line y=0, so the independence assumption is satisfied. From Figure 3's side by side boxplots, we observe lots of variation between different treatment combination groups, with two groups heavily skewed to the left, and a few groups with longer bars than others. This tells us that the constant variance assumption has not been satisfied.

Table 1 shows the shorthands for the conditions that will be used in the following plots.

no music, no grouping	no0
music, no grouping	yes0
no music, grouping by 3 letters	no3
music, grouping by 3 letters	yes3
no music, grouping by 5 letters	no5
music, grouping by 5 letters	yes5

Table 1

B. Informal Analysis

Figure 2 does not appear to show any discernable patterns, but does reveal a lot of interaction. It's difficult to see if there is a subject effect, a run effect, or what's going on with each treatment combination. In Figure 3, the subject vs response plot shows two things that stand out. The first is that there appears to a be a subject effect, since subjects 6-10 don't have as varied responses, but that is most likely due to random chance because we selected random students in the classroom. The second is that a lot of subjects achieved a perfect score in some of the runs, which suggests that we did not make the strings long enough to test a wider range of letters, which would have yielded a less skewed distribution of responses. This will negatively impact our findings because it suggests 2 treatments with the same response rate of 15 are similar, when in fact, a difference might have been noticed if we had tested 25 letters, for example. Looking at the Histogram of Correct Response Rate (Figure 4), it is clear that our data is left-skewed: most of our section had an excellent ability to accurately memorize the strings. Our preliminary sample, which was used to calculate our variance, did not do as well despite the fact that our preliminary sample was conducted on other Cal students .

The other plot, a boxplot (in Figure 3) for each treatment combination reveals that the variance for each condition might be similar (later plots will confirm this), and that the treatments themselves were not that different from each other, even though a plot of the condition averages does reveal differences.

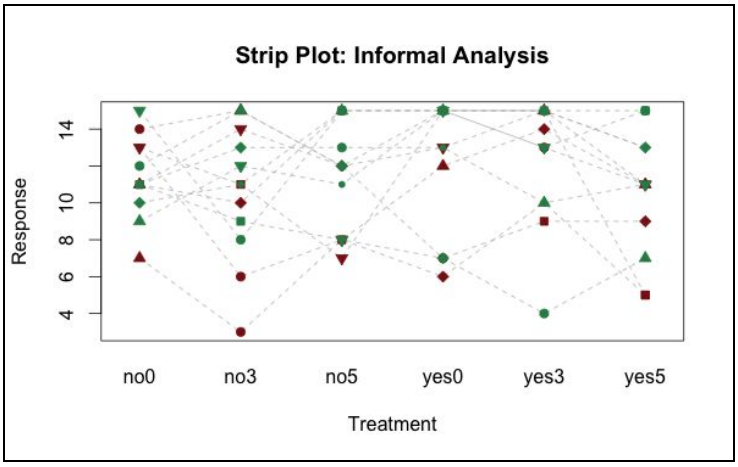

Figure 2

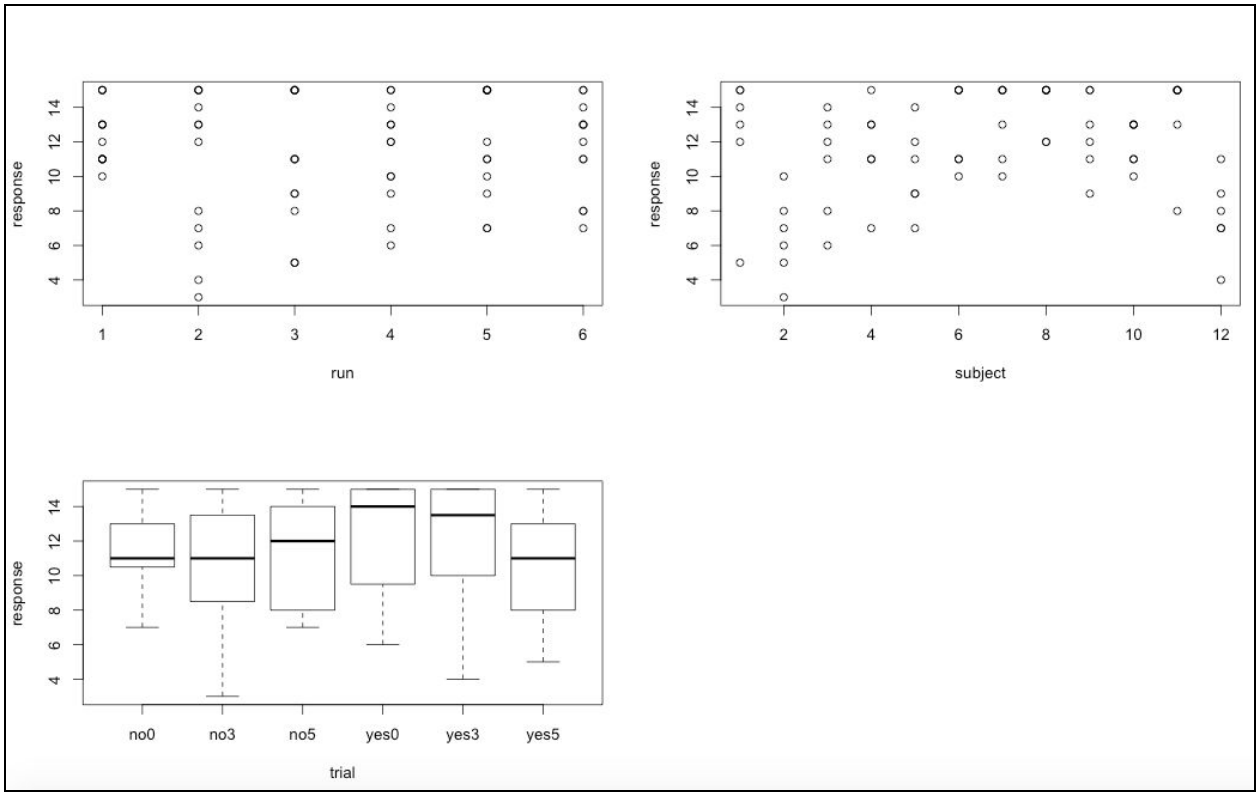

Figure 3

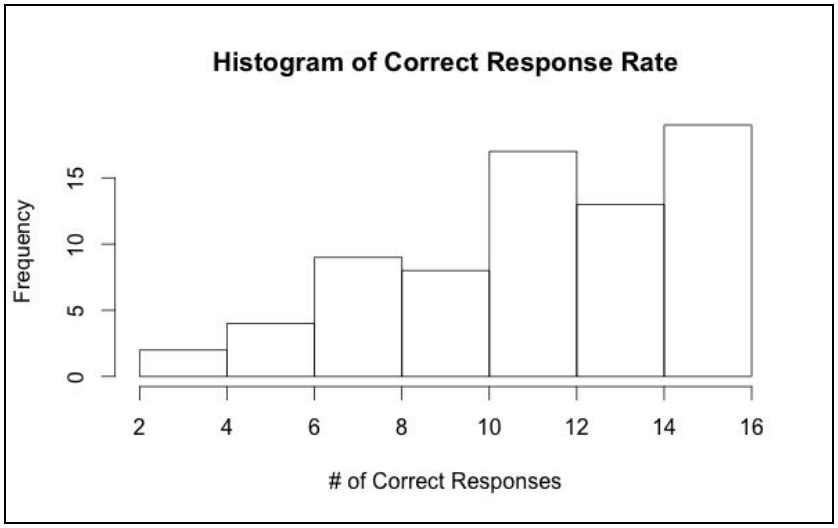

Figure 4

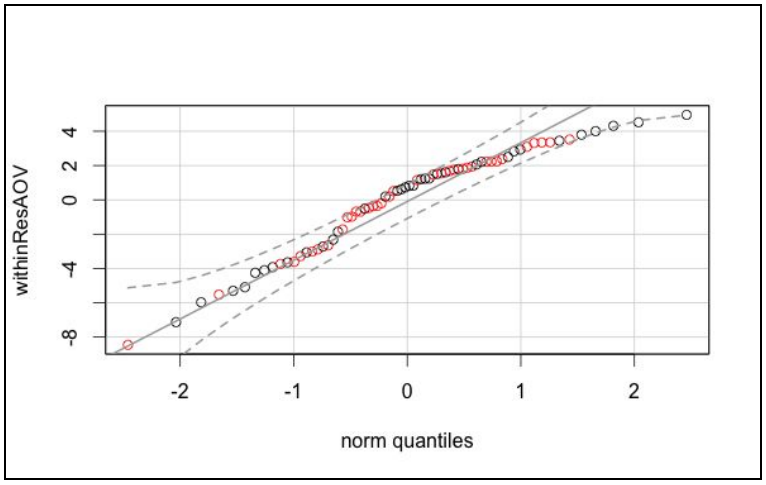

Figure 5

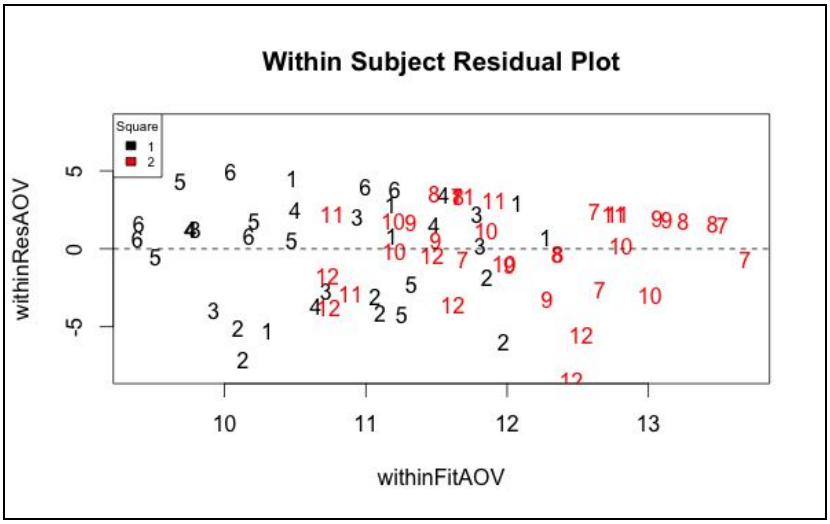

Figure 6

C. ANOVA Analysis

After our informal analysis, we did an ANOVA analysis on our data. We treated our nuisance factor, 'subject,' as a random effect because our subjects were sampled from a population of subjects. We set our α=0.05. Our null hypothesis was that there is no significant difference in correct response rate between different treatment combinations. Our p-value for our treatment combination turned out to be 0.5866. Therefore, we were unable to reject our null hypothesis.

ANOVA TABLE					
Error: Subject					
	DF	**SS**	**MS**		
Square	1	13.1	13.1		
Error: Within					
	DF	**SS**	**MS**	**F-statistic**	**p-value**
Square	1	30.8	30.846	2.978	0.0893
String	1	0.1	0.076	0.007	0.9319
Trial (i.e. Treatment Combination)	5	39	7.805	0.754	0.5866
Residuals	63	652.5	10.356	N/A	N/A

D. Contrasts

In order to test for contrasts, we set our α=0.05. Our null hypothesis is that our contrast vector w(μ)=0 (i.e. the mean for each of the 6 treatment combinations are all equal to each other). We decided to test 2 contrasts, one to compare music to no music, and the other to compare grouping by 3 letters vs. grouping by 5 letters. We got p-values of .42 and .56, respectively, so we failed to reject the null hypothesis that there is no effect for grouping or music. The confidence intervals are in Figure 7.

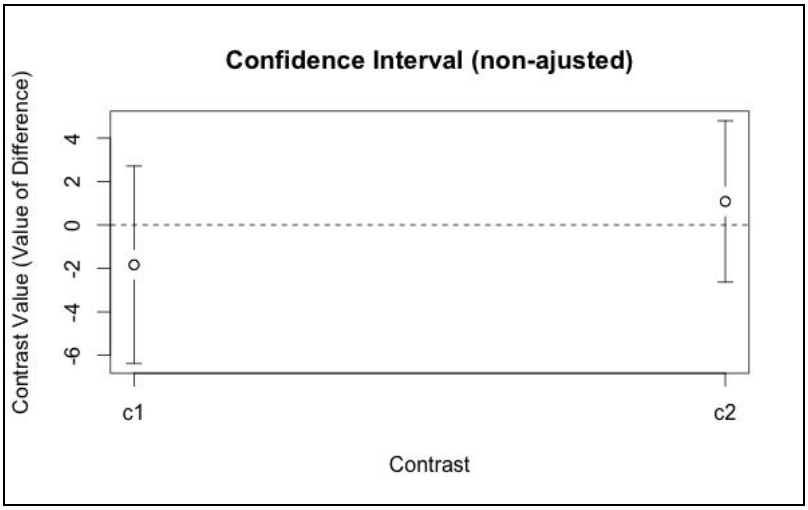

Figure 7

In order to account for multiple testing, we decided to do a Bonferroni correction with K=2. This correction was quite small, however, and did not change our results from above. The confidence intervals for the Bonferroni correction are in Figure 8 below.

Figure 8

VIII. Discussion

This experiment, although unsuccessful in demonstrating the efficacy of several memory aids, does show promise. Two reasons can be attributed to our lack of significance. One is that our experiment relied on 2 latin squares, which made our power rather weak, and thus, our probability of correctly rejecting a false null hypothesis decreased. A higher power test means that we would have a greater probability of correctly rejecting the null.The second is that our trial runs before the actual experiment underestimated the memory of our subjects. This created a ceiling effect and further hampered analysis. We believe that once these factors are accounted for, the experiment will yield positive results. Either way, this experiment raised several interesting questions that could prove to form the base for new studies. For instance, maybe grouping by itself isn't a memory aid, but rather chunks that have meaning to the individual. The grouping factor could be expanded on to include spaces between the chunks, or maybe entire blank lines between the chunks. It may also be worthwhile to consider that there are significant subject effects, and some methods may work better for certain individuals than others.

Appendix B. R Code

```
################
# Code Chunk 0  : Read in and format Data
################
#Put this file in the working directory
mem <- read.csv("memory.csv")

#adds new columns for each factor
mem$music <- 1:72
mem$group <- 1:72
mem$square <- c(rep(1,36),rep(2,36))
mem$string <- c(rep(1:6,12))

#rewrites factor column values to the correct ones
for (i in 1:72) {
  ifelse(mem$trial[i]=="yes0" || mem$trial[i]=="yes3" || mem$trial[i]=="yes5", mem$music[i] <- 1,
mem$music[i] <- 0)
  ifelse(mem$trial[i]=="yes0" || mem$trial[i]=="no0", mem$group[i] <- 0, 3+3)
  ifelse(mem$trial[i]=="yes3" || mem$trial[i]=="no3", mem$group[i] <- 3,3+3)
  ifelse(mem$trial[i]=="yes5" || mem$trial[i]=="no5", mem$group[i] <- 5,3+3)
}

################
# Code Chunk 0.5  : Anova Calculations
```

```
################

####ANOVA Calculations
mem.aov <- aov(response~square+string+trial +Error(subject),data=mem)
summary(mem.aov)

#Extracting Fitted and Residual values by using proj function
aovProj.mem<-proj(mem.aov)
names(aovProj.mem)
#get all the effects:
allTerms.mem<-do.call("cbind",lapply(aovProj.mem[1:3],function(x){x}))
colnames(allTerms.mem)[3]<-"subject"
allTerms.mem

################
# Code Chunk 1  : Strip Plot
################

interaction.plot(x.factor=mem$trial,trace.factor=mem$subject,response=mem$response,
            col=c("grey", "seagreen")[mem$square],lty=2,lwd=1,legend=FALSE,
            type="l",ylab="Response",xlab="Treatment", main="Strip Plot: Informal Analysis")
points(x=c(1:6)[mem$trial],y=mem$response,pch=c(20:25)[mem$string],col=c("firebrick4",
"seagreen")[mem$square],bg=c("firebrick4", "seagreen")[mem$square])



################
# Code Chunk 2  : Informal Analysis
################

par(mfrow=c(2,2))
plot(response ~ run+subject+trial, mem)

################
# Code Chunk 3  : Redone Power Calculations
################

grand_avg = mean(mem$response)

par(mfrow=c(1,1))
gpVar = grand_avg - tapply(mem$response,mem$trial,mean)
t<-6
sigmasq<-4.25
```

```
mseq<-1:7
power<-sapply(mseq,function(m){ #m is number of squares -> 3m is number of cows
  npertreatment<- t*m # #per level, n because complete block
  ncpseq<-npertreatment*sum(gpVar^2)/sigmasq #nc factor
  dfDenom<- (t-1)*(t*m-2) #9m is total number of observations
  dfNum<-length(gpVar)-1
  fcutoff<-qf(1-0.05,df=dfNum,df2=dfDenom) #cutoff under the null
  1-pf(fcutoff,df1=dfNum,df2=dfDenom,ncp=ncpseq)
})
plot(mseq,power,log="x",type="b",xlab="Numb Squares")
abline(h=c(0.85,0.9,0.95),lty=2)

################
# Code Chunk 4  : Group averages by condition
################
grp_avgs = tapply(mem$response,mem$trial,mean)
plot(grp_avgs)



################
# Code Chunk 5  : Run Order Plots
################

run_avgs = tapply(mem$response,mem$run,mean)
plot(run_avgs)

#finally the residual plot for Run Order
square_avgs = tapply(mem$response,mem$square,mean)
plot(jitter(mem$run, .3),mem$res, xlab="Run", ylab="Res", main="Run Order vs. Residual")

################
# Code Chunk 6  : Variance Calculations
################

#histogram of responses
hist(mem$response, xlab="# of Correct Responses", main="Histogram of Correct Response
Rate")

#qqPlot
library(car) #Package required for qqPlot function below
withinFitAOV<-rowSums(allTerms.mem[,-ncol(allTerms.mem)])
withinResAOV<-allTerms.mem[,"Residuals"]
qqPlot(withinResAOV,col.lines="darkgrey",col=palette()[mem$square][order(withinResAOV)])
```

```
###############
# Code Chunk 7  : Within Subject Residual Plot
###############

withinFitAOV<-rowSums(allTerms.mem[,-ncol(allTerms.mem)])
withinResAOV<-allTerms.mem[,"Residuals"]
plot(x=withinFitAOV,y=withinResAOV,type="n",main="Within Subject Residual Plot",
        ylim = c(-8, 8))
text(x=withinFitAOV,y=withinResAOV,as.character(mem$subject),col=palette()[mem$square])
abline(h=0,lty=2)
legend("topleft",c("1","2"),title="Square",fill=palette()[1:2], cex=.6)

#adds residual column to data, for contrast calculations in the next section
mem$res <- withinResAOV

###############
# Code Chunk 8  : Contrast Calculations
###############

#The order of our treatments, used to come up with correct contrasts
#   no0      no3      no5     yes0     yes3     yes5
#11.41667 10.58333 11.33333 12.33333 12.33333 10.50000

#The hypothesis we are testing
#music vs no music = c(1,1,1,-1,-1,-1)
#3 vs 5 grouping = c(0,1,-1,0,1,-1)

c1 = c(1,1,1,-1,-1,-1)
c2 = c(0,1,-1,0,1,-1)
means = tapply(mem$response,mem$trial,mean)

anova.table <- summary(mem.aov)
est.var = 10.356 #from anova table, σ.hat^2
df_res =  63 #from table

#Contrast Estimate: w1(μ)-hat, w2(μ)-hat
contEstc1 = sum(c1*means)
contEstc2 = sum(c2*means)

#Standard Error for each contrast estimate from above
seEstc1 = sqrt(est.var * sum(c1^2)/12)
seEstc2 = sqrt(est.var * sum(c2^2)/12)
```

```
#Two t-statistics: one for each contrast
tStatc1 <- contEstc1/seEstc1
tStatc2 <- contEstc2/seEstc2

#2 corresponding p-values (non-corrected)
pval.nonc1 <- 2 * (1-pt(abs(tStatc1),df=df_res))
round(pval.nonc1, 5)

pval.nonc2 <- 2 * (1-pt(abs(tStatc2),df=df_res))
round(pval.nonc2, 5)


#Confidence intervals
memC <- rbind(c1 = c(1,1,1,-1,-1,-1), c2 = c(0,1,-1,0,1,-1))
memC
contEst <- as.vector(memC %*% means)
names(contEst) <- row.names(memC)
seEst <- sqrt(est.var * rowSums(memC^2)/12)
tStat <- contEst/seEst

tfactor <- abs(qt(0.05/2,df=df_res))
lower <- contEst - tfactor*seEst
upper <- contEst + tfactor*seEst
cbind(Lower=lower,Estimate=contEst,Upper=upper)

library(gplots)
plotCI(contEst,ui=upper,li=lower,main="Confidence Interval (non-ajusted)",ylab="Contrast Value
(Value of Difference)",xaxt="n",xlab="Contrast")
axis(1,1:2,names(contEst))
abline(h=0,lty=2)

#Confidence Intervals (Bonferroni correction)
bfactor <- abs(qt((0.05/2)/2,df=df_res)) #Adjusted t-factor: K=2
lowerB <- contEst - bfactor*seEst
upperB <- contEst + bfactor*seEst
cbind(LowerB=lowerB,Estimate=contEst,UpperB=upperB)

plotCI(contEst,ui=upperB,li=lowerB,main="Bonferroni-adjusted",ylab="Contrast Value (Value of
Difference)",xaxt="n",xlab="Contrast")
axis(1,1:3,names(contEst))
abline(h=0,lty=2)
```