**Session 6: 2\_2 ANOVA.R**

(1) Another way to transform RT data is to use an inverse transform (i.e. 1/RT).

Take the RT data in Mydata, and create a new column with the inverse transformed data. Call this column InvRT  
   
Look at this data in a histogram and with quantile plots. Does it improve the distribution?

Mydata.corr$InvRT <- 1/Mydata.corr$RT

hist(Mydata.corr$InvRT)

qqnorm(Mydata.corr$InvRT)

qqline(Mydata.corr$InvRT)

(2) We will use the Mydata.corr.a (this the data with errors and outliers removed).  
Create summary data for each participant, with their mean RT for each level of HomLocation. Save this in ‘data.summary’.

data.summary <- ddply(Mydata.corr.a, c("SubNo","HomLocation"), summarise,

meanRT = mean(RT, na.rm=TRUE))

(3) What is the mean RT for the scalp location PZ? (across all subjects)  
Hint: use the ‘with’ and ‘tapply’ code.

with(data.summary, tapply(meanRT, HomLocation, mean))

*PZ Mean RT = 0.443 seconds*

(4) Complete a repeated measures ANOVA for HomLocation.  
HomLocation was manipulated within subjects.  
Use the log of mean RT. Hint: see code line 206.  
Is there a significant main effect of HomLocation (scalp location)?

aov.1 <-aov(log(meanRT) ~ HomLocation + Error(SubNo/HomLocation), data.summary)

*No significant main effect (F(23,432) = 0.17, p>1)*

(5) Complete follow up pairwise t-tests. Use the Holm method to control for false positives.  
Is there a significant difference between scalp locations ATL and CZ?  
Is there a significant difference between scalp locations CP1.2 and F7.8

pairwise.t.test(log(data.summary$meanRT), data.summary$HomLocation, p.adjust.method="holm", paired=T)

*ATL and CZ: p = 0.00079 Yes  
CP1.2 and F7.8 = 0.01030 Yes*

(8) To look at the data, create a boxplot of the RTs by each HomLocation

par(mfrow=c(1,1))

boxplot(data.summary$meanRT ~ data.summary$HomLocation)

**Session 7: 2\_3 More on ANOVA.R**

(1) Complete an ANOVA analysis with Axes, TwitchFactor and Task (2x1 Mixed ANOVA).  
In this analysis, look for the interaction and main effects for Axes x TwitchFactor, but only look at the main effect for Task.  
Store the result of this analysis in ‘aov.4’  
What are the results?  
  
data.summary <- ddply(Mydata.corr.a, c("SubNo", “Axes”, "Task", "TwitchFactor"), summarise,

meanRT = mean(RT, na.rm=TRUE))

aov.4 <- aov(meanRT ~ (Axes\*TwitchFactor) + Task

+ Error(SubNo/(Axes\*TwitchFactor)), data=data.summary)

*Marginal main effect of TwitchFactor  
Significant main effect of Task  
No effect of Axes  
No interaction between Axes and TwitchFactor*

(2) Use interaction.plot to look at the data above, specifically the interaction between Axes and TwitchFactor  
  
*with(data.summary, interaction.plot(Axes, TwitchFactor, meanRT, fun = mean))*

(3) Find the ‘revalue’ code where we change Hemisphere from 1 and 2 to Left and Right. Re-run this so that our current data has Hemisphere coded as Left and Right. Also change SubNo into a factor.  
Then run the ezANOVA again from line 180.  
How can you see the results from aov.1? What do you find?

data.summary$Hemisphere<-as.factor(data.summary$Hemisphere)

data.summary$Hemisphere <- revalue(data.summary$Hemisphere, c("1" = "Left", "2" = "Right"))

data.summary$SubNo <- as.factor(data.summary$SubNo)

aov.1 <- ezANOVA(data=data.summary, dv=meanRT, wid = SubNo, between = .(Task,Hemisphere))

aov.1

(4) Using ezANOVA, run a mixed ANOVA.  
Axes and TwitchFactor are within subjects  
Task is between subjects

Look at main effects for all variables (no interactions)  
You will need to create summary data first  
You can look at help("ezANOVA") to see how to enter between subjects and within subjects parts of the analysis.

Save it in aov.4

data.summary <- ddply(Mydata.corr.a, c("SubNo", "Task", “Axes”, "TwitchFactor"), summarise,

meanRT = mean(RT, na.rm=TRUE))

*aov.4 <-* ezANOVA(data=data.summary, dv=meanRT, wid = SubNo,   
between = Task, within = .(Axes,TwitchFactor))

(5) Create summary data from Mydata.corr.a

This time, generate average RTs for Task and Congruence  
Task is between subjects  
Congruence is within subjects  
Complete a mixed ANOVA for these two factors  
Look at the main effects and intereaction of Task and Congruence

data.summary <- ddply(Mydata.corr.a, c("SubNo", "Task", "Congruence"), summarise,

meanRT = mean(RT, na.rm=TRUE))

mix.anova <- aov\_car(meanRT ~ Task\*Congruence + Error(SubNo/Congruence),

data=data.summary)

(6) Create an interaction plot to look at the data  
What do the anova and the plot show?

with(data.summary, interaction.plot(Task, Congruence, meanRT, fun = mean))

(7) Do follow up comparisons so that you compare the Incongruent condition in Flanker to the Incongruent condition in CRT.

ref1 <- lsmeans(mix.anova, c("Task", "Congruence”))

c\_list <- list(

"Flanker.Incong – CRT.Incong" = c(0,0,1,-1))

summary(contrast(ref1, c\_list), adjust = "holm")

(8) Add a title and better y axis label to the graph

Can you work out how to change the key labels from ‘Cong’ and ‘Incong’ to

‘Congruent’ and ‘Incongruent’?

#label axes

plot<-plot + ylab("Mean RT (seconds)")

#add a title

plot<-plot + ggtitle("Mean RT by Task and Congruence")

**Session 8: 3\_1 Introduction to LMMs.R**

(1) Build an LMM that includes fixed effects for Trial, main effects for Twitches and main effects and interactions between Task and Congruence and main effect of Twitches.  
Include random slopes for Twitches varying across Subjects, and random slopes for the effect of Congruence varying across Subjects. Include correlations between intercepts and slopes.  
Save this in lmer.4

lmer.4<-lmer(log(RT) ~ Trial + Twitches + (Task \* Congruence)

+ (1 + Twitches|SubNo) + (1+Congruence|SubNo), data = Mydata.corr.a)

(2) Plot the fixed effects from this model.

eff<-allEffects(lmer.4)

plot(allEffects(lmer.4))

(3) Use update( ) to compare this model to a model that does not include the interaction between Task and Congruence. Save this model in lmer.4.NoInt  
What is the result?  
  
lmer.4<-lmer(log(RT) ~ Trial + Twitches + (Task \* Congruence)

+ (1 + Twitches|SubNo) + (1+Congruence|SubNo), data = Mydata.corr.a)

lmer.4.NoInt <- update(lmer.2,.~.-Task:Congruence)

(4) Plot the random effects for Congruence varying across Subjects. Make sure you update the axes labels. (hint – modify code on lines 155-157)

lattice::xyplot(fitted(lmer.4) ~ Congruence|SubNo, data=Mydata.corr.a,

main="Congruence by Subject Random Effects",

ylab="Fitted log RT",xlab="Congruence")