Audio Data Task

Background

The data `audio.csv` contains data on 117 subjects with mild cognitive impairment (MCI). The subjects were volunteers in a longitudinal clinical trial of a new drug (LG-03812) that was investigated foa ny ability to improve or preserve attention in these patients. The focus of this trial was not the effectiveness of the drug as it pertains to MCI, but but on potential toxicity of the drug on the inner ear (ototoxicity). Subjects were randomized to one of three dosage groups: 0.50 mg/day, 0.25 mg/day, or a matching placebo. The planned duration of treatment was 12 months, after which time the primary effect of treatment on patients' attention would be assessed using the digit symbol substitution test (DSST). Secondary outcomes included other measures of cognitive function, as well as measures of treatment safety including hearing thresholds.

Randomization was stratified by baseline measures of attention (DSST < 35 vs DSST ≥ 35). At baseline and each follow-up visit, measures were made to assess the drug's ototoxicity. A tone was sounded near the patient's right and left ear, at frequencies of 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, 3000 Hz, and 4000 Hz. For each frequency, the administrators measured the decibel threshold required for the patient to discern the tone, and recorded the difference from baseline in the decibel threshold.

Research questions

Many of the questions relevant to this data set require more advanced methods for longitudinal data. However, we can simplify this data to answer questions at a level of a first or second course in statistics. Because of the longitudinal nature of these data, nontrivial data cleaning and manipulation needs to take place before the data are even in a form suitable for simple statistical tests. Some research questions are as follows:

1. Was the randomization successful? If not, we would detect an association between DSST threshold (<35 or ≥ 35) and dose group. Is there evidence of such an association?
2. Is there a difference in the average number of visits in the study across gender?
3. Is there a difference in the average length of time in the study across gender?
4. Is there a difference across dosage group in the average decibel threshold, averaged across frequency and time?

DSCI: Research Question #1

Clearly, we cannot answer any of these questions with the data as-is. We need to simplify it down to one row per patient.

Start with the original data by using the `read.csv()` command. We then want to reduce it, by getting rid of duplicated patients; create a new variable called DSSTThresh, and carry out the chi-squared test:

```
long <- read.csv('audio.csv')
short <- long[!duplicated(long$Subject),] #Investigate this command
head(short)
short$DSSTThresh <- ifelse(short$DSST < 35, 1, 0)
```

The data are now good to go! Create a stacked bar graph of DSST threshold by dosage group, and carry out the chi-squared test.

DSCI: Research Question #2

For Question 2 we want to add a new variable called `nvisits` that equals the number of visits for each patient. We can use the `by()` command for this:

```
nvisits <- by(long,long$Subject,nrow)
short$nvisits <- nvisits
head(long[,c('Subject','Sex','Race')])
head(short[,c('Subject','Sex','Race','nvisits')])
```

There's a problem with the data now! Can you identify it? To fix it, we need to back up a bit. Investigate the following:

```
short <- long[!duplicated(long$Subject),]
order(short$Subject) #Why is the first element "2"?  Why does "1" not show up until the 48th elem
ent?
short <- short[order(short$Subject),]
head(short) #That's better!
nvisits <- by(long,long$Subject,nrow)
short$nvisits <- nvisits
head(short[,c('Subject','Sex','Race','nvisits')])
short[short$Subject==2001,] #Verify that subject "2001" has 2 visits: yep!
```

We're good! Create a boxplot of number of visits by sex, and carry out the t-test.

DSCI: Research Question #3

Now we need to compuate the length of time on treatment for each individual. Intuitively, we want to compute `short$StopDate-short$StartDate`. But try to run this command: R doesn't like it! Why not?

To compute "time on treatment" we need to change the class of both `StopDate` and `StartDate` to class `Date`. What class are they currently?

To change the class, investigate the following code:

```
#Need to transform to "date" class:
as.Date(short$StopDate) #Why doesn't it work? See ?as.Date
as.Date(short$StartDate,format="%m/%d/%y") #What's wrong with this?
as.Date(short$StartDate,format="%m/%d/%Y") #Good to go!
short$TimeInTrial <- as.Date(short$StopDate,format="%m/%d/%Y") -
                     as.Date(short$StartDate,format="%m/%d/%Y")
head(short$TimeInTrial)
summary(short$TimeInTrial)  #Is of class "difftime"; should probably change to "numeric"
short$TimeInTrial <- as.numeric(short$TimeInTrial)
```

```
summary(short$TimeInTrial)
class(short$TimeInTrial)
```

We're good! Create a boxplot of time in trial by sex, and carry out the t-test.

DSCI: Research Question #4

Answering #4 is slightly more involved than the previous three, and will require use of `apply()`. We want to average across the number of visits, across frequency, for each patient. To do this we need to start from the original "long-form" data set. But there is a hiccup: some patients are missing measurements on some frequencies and visits. We will need to account for this when we calculate means.

```
apply(long[,12:ncol(long)],2,mean)
colMeans(long[,12:ncol(long)]) #Equivalent to above
by(long[,12:ncol(long)],long$Subject,colMeans,na.rm=TRUE) #Note use of na.rm=TRUE.  Need to simpl
ify this:
simplify2array(by(long[,12:ncol(long)],long$Subject,colMeans)) #Need to transpose:
head(t(simplify2array(by(long[,12:ncol(long)],long$Subject,colMeans)))) #Looks good!  Add it to "
short"
short[,12:27] <- t(simplify2array(by(long[,12:ncol(long)],long$Subject,colMeans,na.rm=T)))


#Now want to average across right and left ear
#One way which requires counting, which can be very error prone:
short$RightAvg <- apply(short[,12:19],1,mean)
#Alternative that searches for what we want in the variable names:
grep('R[0-9]',names(short))
short$RightAvg <- apply(short[,grep('R[0-9]',names(short))],1,mean)
short$LeftAvg <- apply(short[,grep('L[0-9]',names(short))],1,mean)
```

A lot went on in the above code! Make sure you understand what each line accomplishes.

Now that the data cleaning is done, create boxplots of average threshold on the right ear by dosage, and do the same for the left ear. We can also carry out the one-way ANOVAs as follows:

```
#Left ear:
left.anova <- aov(LeftAvg~factor(Dose),data=short)
summary(left.anova)
#No significant difference; but if there were we could carry out Tukey's HSD to correct the famil
ywise error rate:
TukeyHSD(left.anova)
```

This is a good chance to talk about the limitations of this simplified approach. We are throwing out a *lot* of information, by averaging not only over the number of visits but also the tone frequencies.