Instructions for using R script to merge testable data

Jonathan Phillips

September 15, 2015

1 Downloading your testable data

Download the zipped file to the folder where you want the aggegrated data file to appear. Then unzip that folder to the same location.

2 R Script walkthrough

After cleaning out the environment (this is optional but helps to keep things transparent), set the working directory to the unzipped folder that contains your testable data. Make sure that the folder does not contain any other files!

Next, you'll store the names of all of the files in a vector, and then use that vector to read each csv file into R. They will each be assigned a name that's the same as their filename. This code does that.

```
files <- list.files()
for(i in 1:length(files))
  assign(files[i], read.csv(files[i],header=F,stringsAsFactors=F))
```

The next bit of code grabs the dimensions and names for the data files you are working with and assigns them to a new dataframe.

```
colNum <- length(get(files[1]))

d <- data.frame(matrix(nrow=0,ncol=colNum))
colnames(d) <- get(files[1])[3,]
```

The next big for loop is the key portion of the code and it does a couple of things:

- Get each participant's demographic information and assigns it to its own column
- Delete the now redundant information testable stores at the top of each file
- Add each participant's data to the dataframe we created
- Assign the correct column names to each of the columns in the new dataframe

```
for(i in 1:length(files)){
   j <- get(files[i])

   j$turkID <- j[2,1]         # get Mturk ID
   j$age <- j[2,2]            # get age
   j$sex <- j[2,3]            # get sex
   j$education <- j[2,4]      # get education
   j$handedness <- j[2,5]     # get handedness
   j$other1 <- j[2,6]         # get other 1
   j$other2 <- j[2,7]         # get other 2
   j$other3 <- j[2,8]         # get other 3
   j$other4 <- j[2,9]         # get other 4
   j$other5 <- j[2,10]        # get other 5
   j$browser <- j[2,11]       # get browser
   j$version <- j[2,12]       # get version
   j$screenWidth <- j[2,13]   # get screen width
   j$screenHeight <- j[2,14]  # get screen height
   j$OS <- j[2,15]            # get operating system
   j$OS_lang <- j[2,16]       # get operating system language
   j$calibration <- j[2,17]   # get calibration
   j$ip <- j[2,18]            # get IP address
   j$code <- j[2,19]          # get completion code

   j <- j[-c(1:3),]
   d <- rbind(d,j)
  }

colnames(d) <- c(get(files[1])[3,],colnames(d[,tail(colnames(d),n=19)]))
```

Now all that's left to do is write the new dataframe to a .csv file. We first go up a level to the parent folder where you saved the zipped file and then write the csv.

```
setwd("../")
write.csv(d,file="TestData.csv", row.names=F)
```

That's it!

3 Questions, Comments, Suggestions?

Feel free to email me at phillips01@g.harvard.edu or suggest revisions on a branch of the github repo at https://github.com/phillipsjs/testable