Instructions for using R script to merge testable data with demographic info

Jonathan Phillips

August 16, 2017

1 Downloading your testable data

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

2 R Script walk-through

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!

```
> setwd("C:/Users/Jonathan/Documents/R/testable")
```

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.

```
> files <- list.files()
> for(i in 1:length(files)) assign(files[i], read.csv(files[i],
+          header=F,stringsAsFactors=F,na.strings=c("", "NA")))
>
```

The next bit of code grabs the names of the demographic variables you collected and all of the variables that are recorded on each trial. We then combine these to determine the number of columns the final dataframe will have.

```
> descript <- get(files[1])[1,][!is.na(get(files[1])[1,])]
> variables <- get(files[1])[3,][!is.na(get(files[1])[3,])]
> colNum <- length(variables)+length(descript)
>
```

The next thing we do is to initalize the dataframe we are going to create. To do this, we simply use the first data file:

```
> d <- get(files[1])
> d <- d[-c(1:3),]
> colnames(d) <- variables
```

With this in hand, we grab the descriptives (for that first data file) and add them on:

```
> jd <- data.frame(get(files[1])[2,])
> colnames(jd) <- descript
> d[,descript] <- jd[1,descript]
>
```

Then we do this iteratively for each data file, while adding that data to the existing dataframe:

```
> for(i in 2:length(files)){
+
+   j <- get(files[i])
+   j <- j[-c(1:3),]
+   colnames(j) <- variables
+
+   jd <- data.frame(get(files[i])[2,])
+   colnames(jd) <- descript
+
+   j[,descript] <- jd[1,descript]
+
+   d <- rbind(d,j)
+ }
>
```

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 folderand then write the csv file.

```
> 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