An R Function for Calculating Mood Entropy from a Series of Likert-Type Scale Items

This short document describes an R function for calculating the normalized mood entropy measure described in detail in:

Glynn, L.M, Howland, M.A., Sandman, C.A., Davis, E.P., Phelan, M., Baram, T.Z., & Stern, H. (2018). Prenatal maternal mood patterns predict child temperament and adolescent mental health. *Journal of Affective Disorders 228,* 83-90.

Mood entropy can be calculated from a series of Likert-type scale items using the following R functions, which were developed by Michael Phelan for the Conte Center @ UCI. This document assumes some familiarity with using the R software package. Note that reverse-scoring of items must be completed before calculating mood entropy.

```
entropy <- function(p){
     iz <- which(p==0)
     z <- -p * log2(p)
     if (length(iz))
         z[iz] <- 0
     sum(z) }
norm_entropy_E <-
     function(x,E=1:4,minnum=floor(length(x)/2)){
         if(sum(is.na(x)) > minnum) return(NA)
         xf <- factor(x,levels=E)
         p <- prop.table(table(xf))
         k <- length(E)
         nent <- 100*entropy(p)/log2(k)
         nent
     }
```

The highlighted portion of the norm_entropy_E function needs to be altered to reflect the range of responses on the Likert-type scale to which we are applying the entropy function. As an example, we will use scores for hypothetical subjects on the Center for Epidemiologic Studies-Depression Scale-Short Form (CESD-SF; Santor & Coyne, 1997). This scale consists of 9 Likert-type items with responses ranging from 0 to 3. Therefore, the highlighted portion of the R function above is changed to `E=0:3`. The norm_entropy_E function is designed to take a vector of responses for an individual subject, determine if at least half the items have responses, create a vector of the proportions for each response, and then compute the entropy for that vector of proportions. The entropy is normalized by expressing it as a percentage of the maximum possible entropy for the items on the scale.

Once the appropriate responses for the scale are specified in E, then the function can be run in R.

Here is an example data set, called datafile, containing the item-by-item responses on the CESD-SF scale for 6 subjects that we will use to demonstrate the function:

```
datafile
 subject     cesd1   cesd2   cesd3   cesd4   cesd5   cesd6   cesd7R   cesd8R   cesd9
1    100         2       1       3       2       3       2        2        3       1
2    101         1       1       1       1       1       1        2        2       1
3    102         2       2       2       1       3       2        1        2       1
4    103         2       1       2       1       3       1        1        1       1
5    104         1       1       4       1       1       1        1        1       1
6    105         2       3       3       3       3       2        2        1       4
```

The above entropy function (norm_entropy_E) will only calculate an entropy score for a single vector of item responses. To apply the entropy function to responses for multiple subjects within a data frame, we will utilize the `apply` function in R to calculate entropy scores for each row of data (i.e., for each subject).

```
cesdentropy <- apply(datafile[,2:10],1,FUN=norm_entropy_E)
```

In the above example line of code, `cesdentropy` is the name we are giving to the vector which will contain entropy values for each row/subject. We must also specify the data file which contains the item-by-item responses (within the `apply` function, `datafile`). We must also define the columns in our data file which contain the item-by-item responses (i.e., which columns correspond to the 9 items on the CESD-SF scale). In our example, columns 2 through 10 contain the 9 CESD-SF items, so within the `apply` function, `[,2:10]`. The second argument in the apply function (the 1) is indicating that we want to apply our function to the rows of the dataset. Finally, `FUN=norm_entropy_E` will loop the entropy function to calculate an entropy score for each row/subject. After running this line of code, we can locate the entropy scores for each of the 6 subjects in the vector we created:

```
cesdentropy
  [1] 0.7652465 0.3821023 0.6758221 0.6121972 0.2516292 0.8763576
```

Often we want to take the normalized entropy values and integrate them into the data frame. This can be accomplished through the following steps. First, we convert the vector of entropy scores into its own data frame:

```
cesdentropy <-data.frame(cesdentropy)
```

Then we can add the entropy scores into our item-by-item datafile:

```
datafile["cesdentropy"] <- NA
datafile$cesdentropy <- cesdentropy$cesdentropy
```

```
datafile
   subject   cesd1   cesd2   cesd3   cesd4   cesd5   cesd6   cesd7R   cesd8R   cesd9  cesdentropy
1      100       2       1       3       2       3       2        2        3       1     76.52465
2      101       1       1       1       1       1       1        2        2       1     38.21023
3      102       2       2       2       1       3       2        1        2       1     67.58221
4      103       2       1       2       1       3       1        1        1       1     61.21972
5      104       1       1       4       1       1       1        1        1       1     25.16292
6      105       2       3       3       3       3       2        2        1       4     87.63576
```