```
# subsetting: selecting 40 sounds with lowest recognition accuracy

setwd('/home/allgoodguys/Documents/Studying/Lund_PhD/epistles/
005_with-Lima/analysis')

### To try and avoid a ceiling effect, let's re-analyze the data
using only the least well-recognized sounds
df = read.csv ('data/data_preprocessed.csv')[,-1]
df = df[df$type=='target',] # targets only

aggregate (item~emotionBlock, df, function(x)length(unique(x))) # 10
items per emotion
aggregate(item~condition, df, function(x)length(unique(x))) # all
192 sounds tested in each condition

a = aggregate (hit~item, df, mean) # table(a$emotionBlock)
# hist (a$hit)
a = a[order(a$hit),]

for (i in 1:nrow(a)){
  itemIdx = which(df$item %in% a$item[1:i])
  a$num[i] = length(itemIdx) / nrow(df) # prop. of data left
  a$accuracy[i] = mean(df$hit[itemIdx]) # accuracy for all df when
including sounds 1:i
  a$em[i] = as.character (df$emotionBlock[df$item==a$item[i]][1]) #
emotion of sound i
}
plot (1:nrow(a), a$accuracy, type='p', ylim=c(0,1), xlab='No.
items', ylab='')
lines (1:nrow(a), a$num, col='red')

# a$accuracy[40] # 40 sounds give us 85% accuracy (all 80 sounds ->
91%)

# to make sure we get a certain number of sounds from each emotional
category, say 5 targets per emotion:
a = a[order(a$em, a$hit),]
idx = c()
for (i in unique(a$em)){
  start = min(which(a$em==i))
  idx = c(idx, (start:(start+4)))
}
df40 = df[df$item %in% a$item[idx],]
mean(df40$hit) # 85%, but this time we have nice, equal numbers of
targets and distractors from each emotional category:
aggregate (item~emotionBlock, df40, function(x)length(unique(x))) #
now we have 5 targets per emotion
# aggregate(hit~emotionBlock, df40, mean)

write.csv (df40, 'data_preprocessed_40sounds.csv')
```