

```

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

df = read.csv ('data/data.csv')

# re-code correct as T/F
df$correct = ifelse(df$correct=='C', T, F)

# repeat indiv measurements for each row with NA
vs = 13:16 #
c('yearsMusic', 'ageMusic', 'highLowDiscrim', 'durDiscrim')
nonNA = which(!is.na(df$yearsMusic))
diff(nonNA) # 1, 193, ... +192 rows each time
length(nonNA) == length(unique(df$subject)) # yes: for each subject

for (i in nonNA){
  df[(i : (i+191)), vs] = df[i,vs]
}

# same for memTask
nonNA_memTask = which(!is.na(df$memTask))
diff(nonNA_memTask)
for (i in nonNA_memTask){
  df[(i : (i+11)), 'memTask'] = df[i, 'memTask']
}

# get a short sound ID and emotion from mixed block-sound variable
soundName
length(unique(df$soundName[df$type=='target'])) # 80 target sounds
length(unique(df$soundName[df$type=='distractor'])) # 112 distractor
sounds - extract their emotion!!!
colnames(df)[6] = 'emotionBlock'
df$emotion = NA
for (i in 1:nrow(df)){
  a = as.character(df$soundName[i])
  b = strsplit(a, ' ')[[1]][3]
  c = substr(b, 1, nchar(b)-5)
  d = unlist(strsplit(c, '_'))
  # df$sound[i] = paste(d[(length(d)-2) : length(d)], collapse='_')
# the last three parts
  df$emotion[i] = d[length(d)-2]
}
length(unique(df$sound))
length(unique(df$emotion))
table(df$emotion); table(df$emotionBlock)
df$emotion = ifelse(df$emotion=='ACH', 'Achievement', df$emotion)
df$emotion = ifelse(df$emotion=='AM', 'Amusement', df$emotion)
df$emotion = ifelse(df$emotion=='ANG', 'Anger', df$emotion)
df$emotion = ifelse(df$emotion=='DG', 'Disgust', df$emotion)
df$emotion = ifelse(df$emotion=='FEAR', 'Fear', df$emotion)
df$emotion = ifelse(df$emotion=='PL', 'Pleasure', df$emotion)
df$emotion = ifelse(df$emotion=='RE', 'Relief', df$emotion)
df$emotion = ifelse(df$emotion=='SAD', 'Sadness', df$emotion)

```

```

df$condition = as.character(df$condition)
df$condition = ifelse(df$condition=='baseline','Fast',df$condition)
df$condition =
ifelse(df$condition=='deliberated','Deliberated',df$condition)
df$condition = ifelse(df$condition=='cogload6','Load
1',df$condition)
df$condition = ifelse(df$condition=='cogload8','Load
2',df$condition)
# unique(df$condition)

for (i in 1:nrow(df)){
  if (df$correct[i]==T & df$type[i]=='target'){
    df$response[i] = 'hit'
  } else if (df$correct[i]==T & df$type[i]=='distractor'){
    df$response[i] = 'correctRejection'
  } else if (df$correct[i]==F & df$type[i]=='target'){
    df$response[i] = 'miss'
  } else if (df$correct[i]==F & df$type[i]=='distractor'){
    df$response[i] = 'falseAlarm'
  }
}
# table(df$response, df$condition)
pos = c('Achievement','Amusement','Pleasure','Relief')
neg = c('Anger','Disgust','Fear','Sadness')

df$sameVal = ifelse ( ((df$emotionBlock %in% pos) & (df$emotion %in%
pos)) | ((df$emotionBlock %in% neg) & (df$emotion %in% neg)), T,
F )

df$hit = ifelse(df$response=='hit', T, F)
df$hit[df$type=='distractor'] = NA
df>falseAlarm = ifelse(df$response=='falseAlarm', T, F)
df>falseAlarm[df$type=='target'] = NA

# df_copy = df
# adding time and trial number to the complete dataset
folder = '/home/allgoodguys/Documents/Studying/Lund_PhD/epistles/
005_with-Lima/analysis/data/FullData/pooled_csv'
files = dir(folder)
df$time2 = NA
df$trialNum = NA

for (f in files){
  p <- read.csv(paste0(folder,'/',f)) # open file
  partNum = as.numeric(substr(f, 1, 3)) # subject number as in full
dataset
  idx = which(df$subject==partNum)
  m = match(df$soundName[idx], p$EventName)
  df$time2[idx] = p$ReactionTime[m] # check: plot (df$time2[idx],
df$time[idx])
  df$trialNum[idx] = p$TrialNo[m]
}
# plot (df$time2, df$time) # should be perfect... In fact, one

```

```

divergent number, for participant 20: df[which(df$time2-df$time>0),]
# table (is.na(df$time), df$condition)
# table (is.na(df$time2), df$condition) # should be no NA's left
# table (is.na(df$trialNum)) # 2 participants with missing (30 and
20)

## extract short stimulus names for the entire long dataframe
for (i in 1:nrow(df)) {
  s = as.character(df$soundName[i])
  s1 = strsplit(s, ' ')[[1]][3]
  if (substr(s1, 1, 4) == 'EXP_') {
    start = 5
  } else {
    start = 1
  }
  df$soundName_short[i] = substr(s1, start, nchar(s1)-5)
}
# length(unique(df$soundName_short)) # 192

## extract pretest results
pretest = read.csv('data/data_pretest.csv')
idx = match(df$soundName_short, pretest$soundName_short)
mean(is.na(idx)) # 112 / 192 - NA for 112 distractors, since there
is no info on them
# cor(df$subjValence, pretest$valence[idx],
use='pairwise.complete.obs') # same
df$subjIntensity = pretest$intensity[idx]
df$pretestAccuracy = pretest$accuracy[idx]
df$pretestAccuracy_derived = pretest$accuracy_derived[idx]
# plot(df$pretestAccuracy, df$pretestAccuracy_derived)

## add sound duration for both target and distractors
dur = read.csv('data/data_duration.csv')
df$duration = dur$duration[match(df$soundName_short, dur$sound)]
# aggregate(duration ~ soundName_short, df, mean)

### mark outliers for latency ( $\pm 3$  SD per participant or  $\leq 250$  ms)
df$outlier = F
for (p in 1:length(unique(df$subject))){
  temp = df[df$subject==unique(df$subject)[p],]
  mean_per_subject = mean(temp$time2, na.rm=T)
  sd_per_subject = sd(temp$time2, na.rm=T)
  lwr = max(250, mean_per_subject - 3*sd_per_subject)
  upr = mean_per_subject + 3*sd_per_subject
  idx_lwr = which(df$subject==unique(df$subject)[p] & df$time2 <=
lwr)
  idx_upr = which(df$subject==unique(df$subject)[p] & df$time2 >=
upr)
  df$outlier[c(idx_lwr,idx_upr)] = T
}
# table(df$outlier); aggregate(time2~outlier, df, range)

```

```

## add text to item and subject to make sure they are immediately
recognized as factors
df$item = paste0('item',sprintf("%03d",df$item))
df$subject = paste0('subject',sprintf("%03d",df$subject))

write.csv(df, 'data/data_preprocessed.csv')

## least accurate 40 sounds, targets only
df = read.csv ('data/data_preprocessed.csv')[,-1]
df_target = df[df$type=='target',] # targets only
a = aggregate(correct ~ soundName_short + emotionBlock, df_target,
mean)
a = a[order(a$emotionBlock, a$correct), ]
idx = vector()
for (i in 0:7) {
  idx = c(idx, 1:5 + i * 10)
}
low = as.character(a$soundName_short[idx])
high = as.character(a$soundName_short[!a$soundName_short %in% low])

df_low = df_target[df_target$soundName_short %in% low, ]
df_high = df_target[df_target$soundName_short %in% high, ]
mean(df_low$correct)
mean(df_high$correct)
# write.csv(df_low, 'data/data_preprocessed_40sounds.csv')

```