# Supplement to Hughes et al. manuscript: #
# Alternatives to Switch-Cost Scoring in the Task-Switching Paradigm: #
# Their Reliability and Increase Validity #

# SAMPLE Task-Switching data scoring R code #
# Computes ACC Switch-Cost, RT Switch-Cost, Rate Residual, Bin, and Inverse Efficiency Scores #
# Meredith M. Hughes, August 2013 #
# NOTE: Some parameters (e.g., the overall accuracy cut-off) should vary by task design #
# Required R package (including dependencies): library(ggplot2) #

# Data frame "data" description: #
# - Includes trial data from Mixed condition only #
# - Variables: #
# * Subject = Participant identifier #
# * Condition = Response task (Odd/Even or High/Low) #
# * SwitchTrial = Trial type (Switch or NonSwitch) #
# * trial = Trial number (1 - total number of trials) #
# * rt = Reaction time (ms) #
# * acc = Response accuracy (0 or 1) #

# Read in data frame #
data <- read.delim("FileName.txt")

# Data Preparation #

# Melt data (Creates two rows per trial: one with rt, one with acc. Trials are identified by #
# Subject, Condition, SwitchTrial, and trial.) #
data.melt <- melt(data, id = c("Subject","Condition","SwitchTrial","trial"), measure = c("rt","acc"))

# Participant exclusion: Remove people with less than 80% overall accuracy #
data.rm.lowacc <- cast(data.melt, Subject ~ variable, mean)
okacc <- data.rm.lowacc[data.rm.lowacc$acc >= .80, "Subject"]
data.melt <- data.melt[data.melt$Subject %in% okacc,]

# RT cleaning #
data.rtmeans <- cast(data.melt, Subject + Condition + SwitchTrial ~ variable, c(mean, function(x) mean(x) + 3*sd(x)))[c(1:5)]
colnames(data.rtmeans) <- c("Subject", "Condition", "SwitchTrial", "rtmean", "rtcutoff")
data.rtcleanup <- merge(data.melt, data.rtmeans, by = c("rt","acc"))
data.rtcleanup2 <- merge(data.rtcleanup, data.rtmeans, by = c("Subject", "Condition", "SwitchTrial"))
data.rtcleanup2$rt <- data.rtcleanup2$value
data.rtcleanup2$rt[data.rtcleanup2$value < 200] <- data.rtcleanup2$rtmean[data.rtcleanup2$value < 200]
data.rtcleanup2$rt[data.rtcleanup2$value > data.rtcleanup2$rtcutoff] <-
data.rtcleanup2$rtcutoff[data.rtcleanup2$value > data.rtcleanup2$rtcutoff]
data.clean <- merge(data.rtcleanup2[,c(1:4,9)], data.melt[data.melt$variable == "acc"], by = c("Subject", "Condition", "SwitchTrial","trial"))[c(1:5,7)]
colnames(data.clean)[6] <- "acc"

# RT Switch-Cost Scores #

# RT Switch-Cost scoring function #
sc.scoring <- function(df, label){
  df.condmeans <- cast(melt(df, id = c("Subject","Condition","SwitchTrial","trial")),
Subject ~ SwitchTrial + variable, mean)
df.condmeans$SwitchCost <- df.condmeans$Switch_rt - df.condmeans$NonSwitch_rt
write.table(df.condmeans[,c("Subject","SwitchCost")], paste("SwitchCost_scores_", label, ".txt", sep = ""), sep = "\t", row.names = FALSE)
# Remove trials after error trial

data.clean.sort <- data.clean[order(data.clean$Subject, data.clean$trial),]
data.clean.sort$PrevAcc <- 1

# For the next line, specify trial numbers based on your task design. PrevAcc is the accuracy of the previous trial.
data.clean.sort[data.clean.sort$trial %in% c(2:72, 73:144), ]$PrevAcc <- 1
data.clean.sort[data.clean.sort$trial %in% c(1:71, 72:143), ]$acc

data.clean.final <- data.clean.sort[data.clean.sort$acc == 1 & data.clean.sort$PrevAcc == 1,]
sc.scoring(data.clean.final, "RTSwitchCost")

###################################
# Accuracy Switch-Cost Scores #
###################################

data.acc.cast <- cast(data.melt[data.melt$variable == "acc",], Subject ~ SwitchTrial, mean)
data.acc.cast$sc_acc <- data.acc.cast$NonSwitch - data.acc.cast$Switch
write.table(data.acc.cast[, c("Subject", "sc_acc")], "SwitchCost_Acc_scores.txt", sep = "\t", row.names = FALSE)

####################
# Rate Residual Scores #
####################

data.cleanrates <- data.clean.sort[data.clean.sort$PrevAcc == 1,]
data.cleanrates$trial <- as.numeric(data.cleanrates$trial)
data.cleanrates$Subject <- as.numeric(as.character(data.cleanrates$Subject))
data.cleanrates <- data.cleanrates[, c("Subject", "Condition", "SwitchTrial", "trial", "rt", "acc")]
data.cleanrates <- melt(data.cleanrates, id = c("Subject", "Condition", "SwitchTrial", "trial"))

# Divide trials into 8 sets (or your preferred number of sets).
subsets <- data.frame(set = 1:8, lower = c(3, 21, 39, 57, 75, 93, 111, 129), upper = c(22, 40, 58, 73, 94, 112, 130, 145))
data.cleanrates$set <- NA
ratesdfs <- list()
for(s in 1:8){
data.cleanrates$set[!is.na(data.cleanrates$trial) & data.cleanrates$trial > subsets[s, "lower"] & data.cleanrates$trial < subsets[s, "upper"]] <- s
ratedfs[[paste("data.rates", s, sep = "")]] <- data.cleanrates[!is.na(data.cleanrates$set) & data.cleanrates$set == s, ]
}
residfinal <- as.data.frame(mat.or.vec(nr = length(unique(data.cleanrates$Subject)), nc = 9))
residfinal[, 1] <- unique(data.cleanrates$Subject)
colnames(residfinal)[1] <- "Subject"

# Calculate residuals within each subset
for(s in 1:8){
  recast <- cast(ratedfs[[s]], Subject ~ SwitchTrial + variable, sum)
  recast$nsrate <- (recast$NonSwitch_acc)/(1.001*(recast$NonSwitch_rt))
  recast$srate <- (recast$Switch_acc)/(1.001*(recast$Switch_rt))
  comdf <- na.omit(recast[, c("Subject", "nsrate", "srate")])
  comlm <- lm(comdf[, "srate"] ~ comdf[, "nsrate"], comdf)
  comresid <- as.data.frame(cbind(comdf$Subject, as.data.frame(comlm$residuals)))
  colnames(comresid)[1] <- "Subject"
  residfinal[, (1+s)] <- comresid[,2]
}

# Sum subset residuals
residual$RateResid <- residfinal$RateResid <- rowSums(residfinal[, c(2:9)])
write.table(residual[, c("Subject", "RateResid")], "RateResidual_scores.txt", row.names = F, sep = "\t")

####################
# Bin Scores #
####################
Data cleaning and analysis procedures:

1. From cleaned RTs, find mean non-switch RT (by condition) for each person:
   ```r
   data.clean.ns.mean <- cast(data.clean, Subject + Condition ~ SwitchTrial + trial, mean)
   data.clean.ns.mean <- data.clean.ns.mean[,c("Subject","Condition","NonSwitch Rt")]
   ```

2. Merge back with Switch Trial RTs and ACCs:
   ```r
   data.bin.setup <- merge(data[data$SwitchTrial == "Switch",], data.clean.ns.mean, by = c("Subject","Condition"))
   data.bin.setup$rt.diff <- data.bin.setup$rt - data.bin.setup$NonSwitch_rt
   quants <- quantile(data.bin.setup[data.bin.setup$acc == 1, "rt.diff"], probs = seq(0,1,.1), names = FALSE)
   data.bin.setup$bin.raw <- cut(data.bin.setup$rt.diff, breaks = quants, include.lowest = TRUE, labels = FALSE)
   data.bin.setup$binscore <- data.bin.setup$bin.raw
   data.bin.setup[data.bin.setup$acc == 0, "binscore"] <- 20
   ```

3. Compute bin scores:
   ```r
   bin.scores <- cast(melt(data.bin.setup[,c("Subject","Condition","trial","binscore")], id = c("Subject","Condition","trial")), Subject ~ variable, sum)
   write.table(bin.scores,"Bin_scores.txt",row.names = F, sep = "\t")
   ```

4. Inverse Efficiency Scores:
   ```r
   ies.data <- data.clean.sort
   ies.rts <- cast(ies.data, Subject ~ SwitchTrial, value = "rt", mean)
   colnames(ies.rts)[2:ncol(ies.rts)] <- paste(colnames(ies.rts)[2:ncol(ies.rts)], "_RT", sep = "")
   ies.acc<- cast(ies.data, Subject ~ SwitchTrial, value = "acc", mean)
   colnames(ies.acc)[2:ncol(ies.acc)] <- paste(colnames(ies.acc)[2:ncol(ies.acc)], "_ACC", sep = "")
   ies.all <- merge(ies.rts, ies.acc, by = "Subject")
   ies.IES <- (ies.all$NonSwitch_RT / ies.all$NonSwitch_ACC)
   ies.IES <- ies.IES / ies.all$NS_IES
   write.table(ies.all[, c("Subject","IES")], "IES_scores.txt", row.names = FALSE, sep = "\t")
   ```