

This table is an example of the structure for the data frame 'JAMA\_trials', after which is the code to be entered into R. The columns are: trial (arbitrary identifier for each trial); measure (identify each baseline variable); Group (identify the group); number in group (identify the number of participants in each group reported for that variable); mean (enter as reported); sd (standard deviation, calculated if necessary from the SEM); decm (the number of decimal places to which the mean is reported; decsd (the number of decimal places to which the standard deviation is reported).

After the code has run call the outputs:

'1-MinP' lists the p values for separate variables in each trial;

'RCTmixnorm' lists the (combined) 1-sided p value for each trial (Stouffer method).

trial	measure	Group	number in group	mean	sd	decm	decsd
1	1	1	4645	66	7	0	0
1	1	2	4652	66	7	0	0
1	2	1	4645	139	20	0	0
1	2	2	4652	139	20	0	0
1	3	1	4645	79	11	0	0
1	3	2	4652	79	11	0	0
1	4	1	4645	69	11	0	0
1	4	2	4652	69	11	0	0
1	5	1	4645	28	4	0	0
1	5	2	4652	28	4	0	0
2	1	1	4761	66	7	0	0
2	1	2	4780	66	7	0	0
2	2	1	4761	139	20	0	0
2	2	2	4780	139	20	0	0
2	3	1	4761	79	11	0	0
2	3	2	4780	79	11	0	0
2	4	1	4761	69	11	0	0
2	4	2	4780	69	11	0	0
2	5	1	4761	28	4	0	0
2	5	2	4780	28	4	0	0

**#THIS IS THE BEGINNING OF THE CODE TO ENTER INTO R; you will need to upload the required packages**

```
require("MBESS")
require("binom")
require("rpsychi")
require("BSDA")
JAMA_trials <- as.data.frame(JAMA_trials)
MaxC1 <- NA
Maxtrial <- max(JAMA_trials$trial)
for(c in 1:Maxtrial){
  C1 <- which(JAMA_trials[,1]==c)
  Maxmeasure <- max(JAMA_trials[C1,2])
  MaxC1[c] <- Maxmeasure
}
```

```

}
PJAMATrial <- matrix(data=NA,nrow=Maxtrial,ncol=max(JAMA_trials$measure,na.rm=T))
ANOVAP <- matrix(data=NA,nrow=Maxtrial,ncol=max(JAMA_trials$measure,na.rm=T))
PJAMATrialsd <- matrix(data=NA,nrow=Maxtrial,ncol=max(JAMA_trials$measure,na.rm=T))
for(r in 1:Maxtrial){
  PPmean <- PPmeansd <- NA
  length(PPmean) <- max(JAMA_trials$measure,na.rm=T)
  length(PPmeansd) <- max(JAMA_trials$measure,na.rm=T)
  for(s in 1:MaxC1[r]){
    Row <- apply(JAMA_trials,1,function(row) any(row[1]==r & row[2]==s))
    m <- 25000
    Length <- length(which(Row))
    Lines <- which(Row)
    Sample <- Samplesd <- rep(1:Length)
    DiffSample <- DiffSamplesd <- rep(1:m)
    Value <- rep(1:Length)
    Participants <- rep(1:Length)
    Standard <- rep(1:Length)
    for(j in 1:Length){
      cum <- JAMA_trials[Lines[j],4]*JAMA_trials[Lines[j],5]
      people <- JAMA_trials[Lines[j],4]
      Value[j] <- cum
      SSDD <-
ifelse(JAMA_trials[Lines[j],4]>300,JAMA_trials[Lines[j],6],s.u(JAMA_trials[Lines[j],6],JAMA_trials[Line
s[j],4]))
      cumvar <- JAMA_trials[Lines[j],4]*SSDD^2
      Standard[j] <- cumvar
      Participants[j] <- people
    }
    Meanmean <- sum(Value)/sum(Participants)
    Meanvar <- sum(Standard)/sum(Participants)
    Meansd <- Meanvar^0.5
    MeanNo <- sum(Participants)/Length
    Diff <- sum((JAMA_trials[which(Row),5]-Meanmean)^2)
    Diffsd <- sum((JAMA_trials[which(Row),6]-Meansd)^2)
    SEMsample <- Meansd/sqrt(MeanNo)
    meansim <- NA
    for(z in 1:m){
      meansim <- rnorm(1,mean=Meanmean,sd=SEMsample)
      for(i in 1:Length){
        Sample[i] <-
round(sum(rnorm(JAMA_trials[Lines[i],4],mean=meansim,sd=Meansd)/JAMA_trials[Lines[i],4]),JAM
A_trials[Lines[i],7])
        Samplesd[i] <-
round(sd(rnorm(JAMA_trials[Lines[i],4],mean=meansim,sd=Meansd)),JAMA_trials[Lines[i],8])
      }
      MeanSample <- sum(Sample*Participants)/sum(Participants)
      MeanSamplesd <- sum(Samplesd*Participants)/sum(Participants)

```

```

DiffS <- (Sample-MeanSample)^2
DiffSsd <- (Samplesd-MeanSamplesd)^2
DiffSample[z] <- sum(DiffS)
DiffSamplesd[z] <- sum(DiffSsd)
}
PP1 <- sum(DiffSample < Diff)/m
PPsd1 <- sum(DiffSamplesd < Diffsd)/m
PP2 <- sum(DiffSample <= Diff)/m
PPsd2 <- sum(DiffSamplesd <= Diffsd)/m
PPmean[s] <- (PP1+PP2)/2
PPmeansd[s] <- (PPsd1+PPsd2)/2
}
PJAMATrial[r,] <- PPmean
PJAMATrialsd[r,] <- PPmeansd
}

#p value from summary data ANOVA
require("CarletonStats")
ANOVAtable2 <- NA
ANOVAP2 <- matrix(data=NA,nrow=Maxtrial,ncol=max(JAMA_trials$measure,na.rm=T))
for(r in 1:Maxtrial){
  for(s in 1:MaxC1[r]){
    Row <- apply(JAMA_trials,1,function(row) any(row[1]==r & row[2]==s))
    Lines <- which(Row)
    ANOVAtable2 <-
invisible(anovaSummarized(JAMA_trials[Lines,4],JAMA_trials[Lines,5],JAMA_trials[Lines,6]))
    ANOVAP2[r,s] <- 1-(as.numeric(ANOVAtable2[8]))
  }
}

#p value from summary data t test
require("BSDA")
MaxC1 <- NA
Maxtrial <- max(JAMA_trials$trial)
for(c in 1:Maxtrial){
  C1 <- which(JAMA_trials[,1]==c)
  Maxmeasure <- max(JAMA_trials[C1,2])
  MaxC1[c] <- Maxmeasure
}
TTESTP <- matrix(data=NA,nrow=Maxtrial,ncol=max(JAMA_trials$measure,na.rm=T))
for(r in 1:Maxtrial){
  for(s in 1:MaxC1[r]){
    Row <- apply(JAMA_trials,1,function(row) any(row[1]==r & row[2]==s))
    Lines <- which(Row)
    ifelse(length(Lines)>2,TTESTP[r,s]<-NA,TTESTP[r,s]<-1-
(tsum.test(JAMA_trials[Lines[1],5],JAMA_trials[Lines[1],6],JAMA_trials[Lines[1],4],JAMA_trials[Lines[
2],5],JAMA_trials[Lines[2],6],JAMA_trials[Lines[2],4])$p.value))
  }
}

```

```

}

MinP <- matrix(data=NA,nrow=Maxtrial,ncol=max(JAMA_trials$measure,na.rm=T))
choice <- NA
for(r in 1:Maxtrial){
  for(s in 1:MaxC1[r]){
    choice <- c(PJAMAtrial[r,s],ANOVAP2[r,s],TTESTP[r,s])
    MCP <- (PJAMAtrial[r,s]-0.5)^2
    ANP <- (ANOVAP2[r,s]-0.5)^2
    TTP <- (TTESTP[r,s]-0.5)^2
    near0.5 <- which.min(c(MCP,ANP,TTP))
    MinP[r,s] <- choice[near0.5]
  }
}

#Stouffer's method which is sumz
RCTmixpnorm <- NA
MinP <- as.data.frame(MinP)
for(k in 1:length(MaxC1)){
  p <- NA
  z <- NA
  z <- which(!is.na(MinP[k,]))
  p <- as.numeric(MinP[k,z])
  erf <- function(x) 2 * pnorm(2 * x/ sqrt(2)) - 1
  erfinv <- function(x) qnorm( (x+1)/2 ) / sqrt(2)
  pcomb <- function(p) (1-erf(sum(sqrt(2) * erfinv(1-2*p))/sqrt(2*length(p))))/2
  pl <- NA
  pl <- length(p)
  RCTmixpnorm[k] <- pcomb(p)
}

#logit mean Wilkinson sumlog sum and sum z methods
require("metap")
JAMApvar <- matrix(data=NA,nrow=Maxtrial,ncol=6)
measures2 <- which(MaxC1>1)
measures4 <- which(MaxC1>3)
for(k in 1:Maxtrial){
  z <- which(!is.na(MinP[k,]))
  p <- as.numeric(MinP[k,z])
  JAMApvar[k,1] <- logitp(p)$p
  JAMApvar[k,3] <- minimump(p)$p
  JAMApvar[k,4] <- sumlog(p)$p
  JAMApvar[k,5] <- sump(p)$p
  JAMApvar[k,6] <- sumz(p)$p
}
for(k in measures4){
  z <- which(!is.na(MinP[k,]))
  p <- as.numeric(MinP[k,z])
}

```

```
JAMApvar[k,2] <- meanp(p)$p  
}
```