

```

# Structure of bf.all and gf.all objects
# 10 main list objects within each one is:

ind.smd.restructuring <- function(d1, d2, d3, omega = 0.05) {

  require("datawizard")
  require("dplyr")

  source("serlin.R")
  source("ncp.t.ci.R")

  # 1. Row combine SMD estimator results for all 11 delta list objects -----

  n.list <- length(d1)
  n.cell <- nrow(d1[[1]]$results$lsn.d1)

  fac.design <- d1[[1]]$design$fac.design

  fac.design <- cbind(seq(1:n.cell), fac.design)
  colnames(fac.design)[1] <- "cell.id"

  lsn.d1 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d1[[i]]$results$lsn.d1[, -1]) }))
  lsn.d2 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d2[[i]]$results$lsn.d2[, -1]) }))
  lsn.d3 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d3[[i]]$results$lsn.d3[, -1]) }))

  hc3.d1 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d1[[i]]$results$hc3.d1[, -1]) }))
  hc3.d2 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d2[[i]]$results$hc3.d2[, -1]) }))
  hc3.d3 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d3[[i]]$results$hc3.d3[, -1]) }))

  mse.d1 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d1[[i]]$results$mse.d1[, -1]) }))
  mse.d2 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d2[[i]]$results$mse.d2[, -1]) }))
  mse.d3 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d3[[i]]$results$mse.d3[, -1]) }))

  acv.d1 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d1[[i]]$results$acov.d1) }))
  acv.d2 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d2[[i]]$results$acov.d2) }))
  acv.d3 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d3[[i]]$results$acov.d3) }))

  # 2. Row combine data summary statistics for all 11 delta list objects -----

  sum.d1 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d1[[i]]$results$sum.data[, -1]) }))
  sum.d2 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d2[[i]]$results$sum.data[, -1]) }))
  sum.d3 <- do.call(rbind, lapply(seq_len(n.list),
    function(i) { cbind(fac.design, d3[[i]]$results$sum.data[, -1]) }))

  # 3. Create delta vector and add it and sample size info to all objects -----

  # Each data.frame has 21 columns at this point

  design <- do.call(rbind, lapply(seq_len(n.list), function(i) { fac.design } ))

```

```

delta <- factor(kronecker(1:9, rep(1, n.cell)),
               labels = c("-1.5", "-0.8", "-0.5", "-0.2",
                          "0", "0.2", "0.5", "0.8", "1.5"))

design <- cbind(design, delta)

# Cross-check factorial design against realised design in sum.d1 etc

sum.design <- sum.d1[, c(1, 14:18)]

colnames(sum.design) <- c("cell.id", "var", "n.size",
                        "bal", "dist", "delta")

lsn.d1 <- cbind(design, lsn.d1[, 6:15])
lsn.d2 <- cbind(design, lsn.d2[, 6:15])
lsn.d3 <- cbind(design, lsn.d3[, 6:15])

hc3.d1 <- cbind(design, hc3.d1[, 6:15])
hc3.d2 <- cbind(design, hc3.d2[, 6:15])
hc3.d3 <- cbind(design, hc3.d3[, 6:15])

mse.d1 <- cbind(design, mse.d1[, 6:19])
mse.d2 <- cbind(design, mse.d2[, 6:19])
mse.d3 <- cbind(design, mse.d3[, 6:19])

if (sum(names(lsn.d1) != names(hc3.d1)) != 0) {
  stop('Column names not the same in lsn.d1 & hc3.d1')
}

if (sum(names(lsn.d2) != names(hc3.d2)) != 0) {
  stop('Column names not the same in lsn.d2 & hc3.d2')
}

if (sum(names(lsn.d3) != names(hc3.d3)) != 0) {
  stop('Column names not the same in lsn.d3 & hc3.d3')
}

# 4. Add liberal and conservative robustness test results -----

# These become Columns 17-18... "rbl" & "rbc"

rs.l <- serlin(delta = 0.025, n.size = d1[[1]]$data$N.reps, omega = omega)$ci

lsn.d1$rbl <- lsn.d1$cap >= rs.l[1] & lsn.d1$cap <= rs.l[2]
lsn.d2$rbl <- lsn.d2$cap >= rs.l[1] & lsn.d2$cap <= rs.l[2]
lsn.d3$rbl <- lsn.d3$cap >= rs.l[1] & lsn.d3$cap <= rs.l[2]

hc3.d1$rbl <- hc3.d1$cap >= rs.l[1] & hc3.d1$cap <= rs.l[2]
hc3.d2$rbl <- hc3.d2$cap >= rs.l[1] & hc3.d2$cap <= rs.l[2]
hc3.d3$rbl <- hc3.d3$cap >= rs.l[1] & hc3.d3$cap <= rs.l[2]

rs.c <- serlin(delta = 0.0125, n.size = d1[[1]]$data$N.reps, omega = omega)$ci

lsn.d1$rbc <- lsn.d1$cap >= rs.c[1] & lsn.d1$cap <= rs.c[2]
lsn.d2$rbc <- lsn.d2$cap >= rs.c[1] & lsn.d2$cap <= rs.c[2]
lsn.d3$rbc <- lsn.d3$cap >= rs.c[1] & lsn.d3$cap <= rs.c[2]

hc3.d1$rbc <- hc3.d1$cap >= rs.c[1] & hc3.d1$cap <= rs.c[2]
hc3.d2$rbc <- hc3.d2$cap >= rs.c[1] & hc3.d2$cap <= rs.c[2]
hc3.d3$rbc <- hc3.d3$cap >= rs.c[1] & hc3.d3$cap <= rs.c[2]

if (sum(names(lsn.d1) != names(hc3.d1)) != 0) {

```

```

...stop('Column names not the same in lsn.d1 & hc3.d1')
...}

...if (sum(names(lsn.d2) != names(hc3.d2)) != 0) {
...stop('Column names not the same in lsn.d2 & hc3.d2')
...}

...if (sum(names(lsn.d3) != names(hc3.d3)) != 0) {
...stop('Column names not the same in lsn.d3 & hc3.d3')
...}

...# 5. Add CI width relative (rwd) to NCP interval -----
...# This becomes Column 19... "rdw"

...temp.ncp <- sum.d1[, c("cell.id", "n.1", "n.2", "delta")]

...ncp.t <- data.frame(unique(temp.ncp[, 2:4]))

...ncp.t$lb.d1 <- 0
...ncp.t$sub.d1 <- 0
...ncp.t$nwd.d1 <- 0

...temp.ncp$nwd.d1 <- NA

...# for d1 and d2 effect sizes
...for (i in 1:nrow(ncp.t)) {

...n1 <- as.numeric(ncp.t[i, 1])
...n2 <- as.numeric(ncp.t[i, 2])
...d <- as.numeric(ncp.t[i, 3])

...ncp.ci <- MBESS::ci.smd(smd = d, n.1 = n1, n.2 = n2)

...nwd.ci <- ncp.ci[[3]] - ncp.ci[[1]]

...ncp.t[i, 4:6] <- c(ncp.ci[[1]], ncp.ci[[3]], nwd.ci)

...tmp.rs <- (temp.ncp$n.1 == ncp.t[i, 1] &
...temp.ncp$n.2 == ncp.t[i, 2] &
...temp.ncp$delta == ncp.t[i, 3])

...temp.ncp$nwd.d1[tmp.rs] <- ncp.t[i, 6]
...}

...lsn.d1$rwd <- lsn.d1$wdt / temp.ncp$nwd.d1
...lsn.d2$rwd <- lsn.d2$wdt / temp.ncp$nwd.d1
...lsn.d3$rwd <- lsn.d3$wdt / temp.ncp$nwd.d1

...hc3.d1$rwd <- hc3.d1$wdt / temp.ncp$nwd.d1
...hc3.d2$rwd <- hc3.d2$wdt / temp.ncp$nwd.d1
...hc3.d3$rwd <- hc3.d3$wdt / temp.ncp$nwd.d1

...rm(tmp.rs, temp.ncp, ncp.t)

...if (sum(names(lsn.d1) != names(hc3.d1)) != 0) {
...stop('Column names not the same in lsn.d1 & hc3.d1')
...}

...if (sum(names(lsn.d2) != names(hc3.d2)) != 0) {
...stop('Column names not the same in lsn.d2 & hc3.d2')
...}

...if (sum(names(lsn.d3) != names(hc3.d3)) != 0) {

```

```

... stop ('Column names not the same in lsn.d33 & hc3.d3')
... }

... # 6. Create wide versions of LSN and HC3 and re-order columns -----
... # ...general order of columns is: cell.id then design factors then results

... # "wide" form has 45 variables

... tmp.d1 <- lsn.d1
... tmp.d2 <- lsn.d2
... tmp.d3 <- lsn.d3

... colnames(tmp.d1)[7:19] <- paste0(names(tmp.d1)[7:19], ".d1")
... colnames(tmp.d2)[7:19] <- paste0(names(tmp.d2)[7:19], ".d2")
... colnames(tmp.d3)[7:19] <- paste0(names(tmp.d3)[7:19], ".d3")

... lsn.wide <- data.frame(tmp.d1[, 1:19], tmp.d2[, 7:19], tmp.d3[, 7:19])

... rm(tmp.d1, tmp.d2, tmp.d3)

... # "wide" form has 45 variables

... tmp.d1 <- hc3.d1
... tmp.d2 <- hc3.d2
... tmp.d3 <- hc3.d3

... colnames(tmp.d1)[7:19] <- paste0(names(tmp.d1)[7:19], ".d1")
... colnames(tmp.d2)[7:19] <- paste0(names(tmp.d2)[7:19], ".d2")
... colnames(tmp.d3)[7:19] <- paste0(names(tmp.d3)[7:19], ".d3")

... hc3.wide <- data.frame(tmp.d1[, 1:19], tmp.d2[, 7:19], tmp.d3[, 7:19])

... rm(tmp.d1, tmp.d2, tmp.d3)

... if (sum(names(lsn.wide) != names(hc3.wide)) != 0) {
...   stop ('Column names not the same in lsn.wide & hc3.wide')
... }

... # 7. Create long versions of LSN and HC3 objects -----
... # This becomes Column 20... "ci"

... lsn.d1$smd <- 1
... lsn.d2$smd <- 3 ... # Swap order of Bonett's delta and Glass's Delta
... lsn.d3$smd <- 2

... # "long" form has 20 variables

... lsn.long <- data.frame(rbind(data.frame(lsn.d1),
...                               data.frame(lsn.d2),
...                               data.frame(lsn.d3)))

... hc3.d1$smd <- 1
... hc3.d2$smd <- 3 ... # Swap order of Bonett's delta and Glass's Delta
... hc3.d3$smd <- 2

... # "long" form has 20 variables

... hc3.long <- data.frame(rbind(data.frame(hc3.d1),
...                               data.frame(hc3.d2),
...                               data.frame(hc3.d3)))

```

```

if (sum(names(lsn.long) != names(hc3.long)) != 0) {
  stop("Column names not the same in lsn.long & hc3.long")
}

# 8. Recode and re-label some factors -----

lsn.long$smd <- factor(lsn.long$smd,
  levels = c(1, 2, 3), labels = c("d1", "d3", "d2"))

hc3.long$smd <- factor(hc3.long$smd,
  levels = c(1, 2, 3), labels = c("d1", "d3", "d2"))

# 9. Create complete wide and long versions of data -----

# ...Becomes Column 21 in "wide" form
# ...Becomes Column 46 on "long" form

lsn.wide$ci <- 1
hc3.wide$ci <- 2

data.wide <- rbind(lsn.wide, hc3.wide)
data.wide$ci <- factor(data.wide$ci, labels = c("LSN", "HC3"))

lsn.long$ci <- 1
hc3.long$ci <- 2

data.long <- rbind(lsn.long, hc3.long)
data.long$ci <- factor(data.long$ci, labels = c("LSN", "HC3"))

# 10. Assumptions, Distributions & Skewness -----

# These become Column 47 to 49 in wide

data.wide <- data.wide %>%
  mutate(ass = case_when(
    distrib == "Nrm" & sigma == "EQ" & balance == "Bal" ~ 1,
    distrib == "Nrm" & sigma == "EQ" & balance != "Bal" ~ 2,
    distrib == "Nrm" & sigma != "EQ" & balance == "Bal" ~ 3,
    distrib == "Nrm" & sigma != "EQ" & balance != "Bal" ~ 4,
    distrib != "Nrm" & sigma == "EQ" & balance == "Bal" ~ 5,
    distrib != "Nrm" & sigma == "EQ" & balance != "Bal" ~ 6,
    distrib != "Nrm" & sigma != "EQ" & balance == "Bal" ~ 7,
    distrib != "Nrm" & sigma != "EQ" & balance != "Bal" ~ 8))

data.wide$ass <- factor(data.wide$ass, labels =
  c("No & Eq & Ba",
    "No & Eq & Un",
    "No & Un & Ba",
    "No & Un & Un",
    "Nn & Eq & Ba",
    "Nn & Eq & Un",
    "Nn & Un & Ba",
    "Nn & Un & Un"))

data.wide <- data.wide %>%
  mutate(dist = case_when(
    distrib == "Nrm" ~ 1,
    distrib == "Plt" ~ 2,
    distrib == "Lep" ~ 3,
    distrib == "n50" | distrib == "p50" ~ 4,

```

```

.....distrib == "n20" | distrib == "p20" ~ 5,
.....distrib == "n05" | distrib == "p05" ~ 6,
.....distrib == "n01" | distrib == "p01" ~ 7))

data.wide$dist <- factor(data.wide$dist,
.....levels = 1:7,
.....labels =
.....c("Norm",
....."Plat",
....."Lept",
....."50%",
....."20%",
....."5%",
....."1%"))

data.wide <- data.wide %>%
mutate(skew = case_when(
.....distrib == "Nrm" ~ 1,
.....distrib == "Plt" | distrib == "Lep" ~ 2,
.....distrib == "n50" | distrib == "n20" |
.....distrib == "n05" | distrib == "n01" ~ 3,
.....distrib == "p50" | distrib == "p20" |
.....distrib == "p05" | distrib == "p01" ~ 4))

data.wide$skew <- factor(data.wide$skew,
.....levels = 1:4,
.....labels =
.....c("Norm",
....."Kurt",
....."Neg.",
....."Pos."))

# These become Column 22 to 24 in long

data.long <- data.long %>%
mutate(ass = case_when(
.....distrib == "Nrm" & sigma == "EQ" & balance == "Bal" ~ 1,
.....distrib == "Nrm" & sigma == "EQ" & balance != "Bal" ~ 2,
.....distrib == "Nrm" & sigma != "EQ" & balance == "Bal" ~ 3,
.....distrib == "Nrm" & sigma != "EQ" & balance != "Bal" ~ 4,
.....distrib != "Nrm" & sigma == "EQ" & balance == "Bal" ~ 5,
.....distrib != "Nrm" & sigma == "EQ" & balance != "Bal" ~ 6,
.....distrib != "Nrm" & sigma != "EQ" & balance == "Bal" ~ 7,
.....distrib != "Nrm" & sigma != "EQ" & balance != "Bal" ~ 8))

data.long$ass <- factor(data.long$ass, labels =
.....c("No & Eq & Ba",
....."No & Eq & Un",
....."No & Un & Ba",
....."No & Un & Un",
....."Nn & Eq & Ba",
....."Nn & Eq & Un",
....."Nn & Un & Ba",
....."Nn & Un & Un"))

data.long <- data.long %>%
mutate(dist = case_when(
.....distrib == "Nrm" ~ 1,
.....distrib == "Plt" ~ 2,
.....distrib == "Lep" ~ 3,
.....distrib == "n50" | distrib == "p50" ~ 4,
.....distrib == "n20" | distrib == "p20" ~ 5,
.....distrib == "n05" | distrib == "p05" ~ 6,
.....distrib == "n01" | distrib == "p01" ~ 7))

```

```

data.long$distrib <- factor(data.long$distrib,
  levels = 1:7,
  labels =
    c("Norm",
      "Plat",
      "Lept",
      "50%",
      "20%",
      "5%",
      "1%"))

data.long <- data.long %>%
  mutate(skew = case_when(
    distrib == "Nrm" ~ 1,
    distrib == "Plt" | distrib == "Lep" ~ 2,
    distrib == "n50" | distrib == "n20" |
    distrib == "n05" | distrib == "n01" ~ 3,
    distrib == "p50" | distrib == "p20" |
    distrib == "p05" | distrib == "p01" ~ 4))

data.long$skew <- factor(data.long$skew,
  levels = 1:4,
  labels =
    c("Norm",
      "Kurt",
      "Neg.",
      "Pos."))

# 11. Reorder columns so that all factors are before results -----

data.wide <- data.wide[, c(1:6, 46:49, 7:45)] # Currently has 49 columns

colnames(data.wide)[c(3:5, 9)] <- c("n.size", "bal", "dist", "dist.sk")

seqn <- seq(1, nrow(data.wide))

data.wide <- data.frame(seqn, data.wide) # Now has 50 columns

# long format has SMD in Column 20 (reordered to Column 9)
data.long <- data.long[, c(1:6, 20:24, 7:19)] # Currently has 24 columns

colnames(data.long)[c(3:5, 10)] <- c("n.size", "bal", "dist", "dist.sk")

seqn <- seq(1, nrow(data.long))
data.long <- data.frame(seqn, data.long) # Now has 25 columns

# 12. Create long version of MSE -----

n.long <- 1:(nrow(data.long)/2) # No longer 3 because GLS removed completely

mse.d1$smd <- 1
mse.d2$smd <- 3 # Swap order of Bonett's delta and Glass's Delta
mse.d3$smd <- 2

colnames(mse.d1)[11] <- "d.est"
colnames(mse.d2)[11] <- "d.est"
colnames(mse.d3)[11] <- "d.est"

mse.long <- rbind(mse.d1, mse.d2, mse.d3)

mse.long <- data.frame(cbind(data.long[n.long, 1:12], mse.long[, 7:20]))

```

```

mse.long <- mse.long[, -9] # Remove "ci" column

# Now has 17 columns

# 13. Create long version of SUM -----

sum.d1$smd <- 1
sum.d2$smd <- 3 # Swap order of Bonett's delta and Glass's Delta
sum.d3$smd <- 2

sum.long <- rbind(sum.d1, sum.d2, sum.d3)

sum.long <- data.frame(cbind(data.long[n.long, 1:12], sum.long[, 6:13]))

sum.long <- sum.long[, -9] # Remove "ci" column

# 14. Return wide and long versions of combined and updated data objects -----

colnames(lsn.d1)[3:5] <- c("n.size", "bal", "dist")
colnames(lsn.d2)[3:5] <- c("n.size", "bal", "dist")
colnames(lsn.d3)[3:5] <- c("n.size", "bal", "dist")

colnames(hc3.d1)[3:5] <- c("n.size", "bal", "dist")
colnames(hc3.d2)[3:5] <- c("n.size", "bal", "dist")
colnames(hc3.d3)[3:5] <- c("n.size", "bal", "dist")

return(list(all.wide = data.wide,
            all.long = data.long,
            all.mse = mse.long,
            all.sum = sum.long,
            lsn.d1 = lsn.d1,
            lsn.d2 = lsn.d2,
            lsn.d3 = lsn.d3,
            hc3.d1 = hc3.d1,
            hc3.d2 = hc3.d2,
            hc3.d3 = hc3.d3,
            mse.d1 = mse.d1,
            mse.d2 = mse.d2,
            mse.d3 = mse.d3,
            sum.d1 = sum.d1,
            sum.d2 = sum.d2,
            sum.d3 = sum.d3,
            rs.l = rs.l,
            rs.c = rs.c))
}

# Mean square error -----
# obs = empirical value
# exp = expected value
# order = order of magnitude to increase results

MSE <- function(obs, exp, order = 0) {
  e <- (obs - exp)^2
  return(10^order * mean(e))
}

bias <- function(obs, exp, order = 0) {
  b <- mean(obs) - exp
  return(10^order * b^2)
}

```



```

}

prec <- function(obs, exp, order = 0) {
  v <- (obs - mean(obs))^2
  return(10^order * mean(v))
}

mse.all <- function(obs, exp, order = 0) {
  mse <- MSE(obs, exp, order)
  bias <- bias(obs, exp, order)
  prec <- prec(obs, exp, order)

  return(data.frame(mse, bias, prec))
}

# Testing restructuring -----

test.restructures <- function(d1.org, d1.rst,
  ..... d2.org, d2.rst,
  ..... d3.org, d3.rst, type) {

  type <- tolower(type)

  # d1.lsn -----

  if (type == "lsn") {

    d1.lsn.org <- colMeans(d1.org[[1]]$results$lsn.d1[, 2:9])

    for (i in 2:length(d1.org)) {
      d1.lsn.org <- rbind(d1.lsn.org,
        ..... colMeans(d1.org[[i]]$results$lsn.d1[, 2:9]))
    }

    rownames(d1.lsn.org) <- NULL

    d1.lsn.rst <- d1.rst ..... %>%
      select(c(delta, est:wtd)) ..... %>%
      group_by(delta) ..... %>%
      summarise(est = mean(est),
        ..... se = mean(se),
        ..... pH0 = mean(pH0),
        ..... pH1 = mean(pH1),
        ..... cap = mean(cap),
        ..... lbv = mean(lbv),
        ..... ubv = mean(ubv),
        ..... wtd = mean(wtd))

    expect_equal(data.frame(d1.lsn.org), data.frame(d1.lsn.rst[, 2:9]))

    # d2.lsn -----

    d2.lsn.org <- colMeans(d2.org[[1]]$results$lsn.d2[, 2:9])

    for (i in 2:length(d2.org)) {
      d2.lsn.org <- rbind(d2.lsn.org,
        ..... colMeans(d2.org[[i]]$results$lsn.d2[, 2:9]))
    }

    rownames(d2.lsn.org) <- NULL

```

```

d2.lsn.rst <- d2.rst %>%
  select(c(delta, est:wtd)) %>%
  group_by(delta) %>%
  summarise(est = mean(est),
            se = mean(se),
            pH0 = mean(pH0),
            pH1 = mean(pH1),
            cap = mean(cap),
            lbv = mean(lbv),
            ubv = mean(ubv),
            wtd = mean(wtd))

expect_equal(data.frame(d2.lsn.org), data.frame(d2.lsn.rst[, 2:9]))

```

d3.lsn -----

```

d3.lsn.org <- colMeans(d3.org[[1]]$results$lsn.d3[, 2:9])

for (i in 2:length(d3.org)) {
  d3.lsn.org <- rbind(d3.lsn.org,
                    colMeans(d3.org[[i]]$results$lsn.d3[, 2:9]))
}

rownames(d3.lsn.org) <- NULL

d3.lsn.rst <- d3.rst %>%
  select(c(delta, est:wtd)) %>%
  group_by(delta) %>%
  summarise(est = mean(est),
            se = mean(se),
            pH0 = mean(pH0),
            pH1 = mean(pH1),
            cap = mean(cap),
            lbv = mean(lbv),
            ubv = mean(ubv),
            wtd = mean(wtd))

expect_equal(data.frame(d3.lsn.org), data.frame(d3.lsn.rst[, 2:9]))
}

```

d1.hc3 -----

```

if (type == "hc3") {

  d1.hc3.org <- colMeans(d1.org[[1]]$results$hc3.d1[, 2:9])

  for (i in 2:length(d1.org)) {
    d1.hc3.org <- rbind(d1.hc3.org,
                      colMeans(d1.org[[i]]$results$hc3.d1[, 2:9]))
  }

  rownames(d1.hc3.org) <- NULL

  d1.hc3.rst <- d1.rst %>%
    select(c(delta, est:wtd)) %>%
    group_by(delta) %>%
    summarise(est = mean(est),
              se = mean(se),
              pH0 = mean(pH0),
              pH1 = mean(pH1),
              cap = mean(cap),
              lbv = mean(lbv),
              ubv = mean(ubv),

```

```

.....wdt = mean(wdt))

....expect_equal(data.frame(d1.hc3.org), data.frame(d1.hc3.rst[, 2:9]))

## d2.hc3 -----

....d2.hc3.org <- colMeans(d2.org[[1]]$results$hc3.d2[, 2:9])

....for (i in 2:length(d2.org)) {
.....d2.hc3.org <- rbind(d2.hc3.org,
.....colMeans(d2.org[[i]]$results$hc3.d2[, 2:9]))
....}

....rownames(d2.hc3.org) <- NULL

....d2.hc3.rst <- d2.rst %>%
.....select(c(delta, est:wdt)) %>%
.....group_by(delta) %>%
.....summarise(est = mean(est),
.....se = mean(se),
.....pH0 = mean(pH0),
.....pH1 = mean(pH1),
.....cap = mean(cap),
.....lbv = mean(lbv),
.....ubv = mean(ubv),
.....wdt = mean(wdt))

....expect_equal(data.frame(d2.hc3.org), data.frame(d2.hc3.rst[, 2:9]))

## d3.hc3 -----

....d3.hc3.org <- colMeans(d3.org[[1]]$results$hc3.d3[, 2:9])

....for (i in 2:length(d3.org)) {
.....d3.hc3.org <- rbind(d3.hc3.org,
.....colMeans(d3.org[[i]]$results$hc3.d3[, 2:9]))
....}

....rownames(d3.hc3.org) <- NULL

....d3.hc3.rst <- d3.rst %>%
.....select(c(delta, est:wdt)) %>%
.....group_by(delta) %>%
.....summarise(est = mean(est),
.....se = mean(se),
.....pH0 = mean(pH0),
.....pH1 = mean(pH1),
.....cap = mean(cap),
.....lbv = mean(lbv),
.....ubv = mean(ubv),
.....wdt = mean(wdt))

....expect_equal(data.frame(d3.hc3.org), data.frame(d3.hc3.rst[, 2:9]))

}

}

ind.mse.restructuring <- function(all.d) {

  all.d$mse.d1$smd <- 1
  all.d$mse.d2$smd <- 3 # Swap order of Bonett's delta and Glass's Delta

```

```

all.d$mse.d3$smd <- 2

colnames(all.d$mse.d1)[17] <- "d.est"
colnames(all.d$mse.d2)[17] <- "d.est"
colnames(all.d$mse.d3)[17] <- "d.est"

all.mse <- data.frame(rbind(all.d$mse.d1, all.d$mse.d2, all.d$mse.d3))

all.mse$smd <- factor(all.mse$smd,
  labels = c("d1", "d3", "d2"))

all.mse$d.type <- NA
all.mse$d.type[all.mse$dist.1 == "Nm" &
  all.mse$dist.2 == "Nm"] <- 1
all.mse$d.type[is.na(all.mse$d.type)] <- 2

all.mse$d.type <- factor(all.mse$d.type,
  labels = c("Normal", "Non-normal"))

all.mse$delta.n <- as.numeric(all.mse$delta)

all.mse$delta.n <- plyr::mapvalues(all.mse$delta.n,
  from = 1:9,
  to = c(1.5, 0.8, 0.5, 0.2,
    0, 0.2, 0.5, 0.8, 1.5))
all.mse$delta.n <- factor(all.mse$delta.n,
  labels = c("0", "0.2", "0.5", "0.8", "1.5"))

all.mse$r.bias <- sqrt(all.mse$bias)

colnames(all.mse)[19:20] <- c("bal", "var")

return(all.mse)
}

n.list <- length(d1)

```