```
# R Code for analyzing indoor dust data
# Written by WGG at ICF in August 2023
# Last updated 1/29/2024

folder <- "C:/CPSC/CO3"

library(data.table)
library(dplyr)
library(readxl)
library(tidyselect)

log_fit = function(var, pcts) {
  # var is data vector (any length >= 2) in regular space (not logs)
  # pcts is list of percentiles to fit (at least 2 percentiles)
  #      Range of pcts is 0-100 so enter median as 50, not 0.5
  n   <- length(pcts)             # Number of percentiles to use
  dat <- log(var)                 # Transform to log space
  Z   <- qnorm(pcts/100, 0, 1)      # Z-scores Z(P)
  L   <- dat                    # Data in log space L(P)
  SZ  <- sum(Z)                   # Sum of z-scores of percentiles
  SZZ <- sum(Z^2)                 # Sum of squares of z-scores
  SL  <- sum(L)                   # Sum of log(data)
  SLZ <- sum(L*Z)                 # Sum of log(data)*Z
  den <- n*SZZ - SZ*SZ            # Divisor in solution
  a   <- (SL*SZZ-SLZ*SZ)/den        # Fitted mean in log space
  b   <- (n*SLZ-SL*SZ)  /den        # Fitted sd in log space
  gm  <- exp(a)                   # Geometric mean
  gsd <- exp(b)                   # Geometric standard deviation
  am  <- exp(a+b^2/2)             # Arithmetic mean
```

```
  return(c(gm, gsd, am))
}

options("warn"=-1)
# Read in raw indoor dust data
raw  <- read_xlsx(paste0(folder,"/CO-03 Indoor Dust Extraction Results_2023-11-20.xlsx"))
ids <- unique(data.table(raw[,2],raw[,3],raw[,6], raw[,7], raw[,8], raw[,9], raw[,14],raw[,22]))
# [,2]  is Litstream ID
# [,3]  is Scenario Description
# [,6]  is Country
# [,7]  is Medium
# [,8]  is Medium submatrix
# [,9]  is Indoor Environment Type (from pulldown menu)
# [,14] is Location type (from pulldown menu)
# [,22] is Chemical

ids$index <- 1:nrow(ids)
rawid  <- as.data.table(left_join(raw, ids))
fwrite(rawid, paste0(folder, "/raw_data_with_ID.csv"))

# Create logfile and report studies with no valid stats
logfile <- data.table(index=NA_integer_, study="",Litstream_ID="",extract_notes="",reason="")[FALSE]
rows   <- is.na(rawid$`Stat Type (from pulldown list)`) | is.na(as.numeric(rawid$`Stat Estimate`)) |
          is.na(rawid$`Stat Units (from pulldown menu)`)
vars   <- c("index","Study Name (in Litstream)","Litstream ID","Overall Notes")
base   <- rawid[!rows]
badind <- unique(rawid$index[!rawid$index %in% base$index])
bad    <- rawid[rawid$index %in% badind, ..vars]
bad    <- bad[!duplicated(bad$index)]
```

```
bad$reason <- "No valid stats"
setnames(bad, names(logfile))
logfile <- rbindlist(list(logfile, bad))
# Base contains all data from raw dataset that have a defined stat type and value

# Check remaining stat units
cat ("\n")
plyr::count(base$`Stat Units (from pulldown menu)`)
cat ("\n")

# Remove records with loadings instead of concentrations
rows   <- base$`Final Stat Unit`=='ng/m2'
dat1   <- base[!rows]
badind <- unique(base$index[!base$index %in% dat1$index])
bad    <- base[rows, ..vars]
bad    <- bad[!duplicated(bad$index)]
bad$reason <- "Only loadings reported"
setnames(bad, names(logfile))
logfile <- rbindlist(list(logfile, bad))

# Convert values to numeric and all units to ng/g
dat1$statval <- as.numeric(dat1$`Stat Estimate`)
dat2 <- dat1[!is.na(dat1$statval)]
rows <- dat2$`Stat Units (from pulldown menu)`=="ug/g"
dat2$statval[rows] <- 1000*dat2$statval[rows]

# Drop DF and LOD if LOD units are not reported
dat2$lodvalue  <- as.numeric(dat2$`LOD Value`)
dat2$lodvalue[dat2$`LOD Units (from pulldown menu)`=="NR"] <- NA
```

```
rows <- dat2$`LOD Units (from pulldown menu)`=="ug/g"
rows[is.na(rows)] <- FALSE
dat2$lodvalue[rows] <- 1000*dat2$lodvalue[rows]
dat2$detecfrac  <- as.numeric(dat2$`DF Value`)
dat2$detecfrac[dat2$`LOD Units (from pulldown menu)`=="NR"] <- NA

# Data from datasets that report point values
psets  <- unique(dat2$index[dat2$`Stat Type (from pulldown list)`=="Point"])
pdat   <- dat2[dat2$index %in% psets]

# Datasets without any Point values
dat3          <- dat2[!dat2$index %in% psets]
dat3$stattype  <- tolower(dat3$`Stat Type (from pulldown list)`)
rows <- dat3$stattype=="percentile"
dat3$stattype[rows] <- tolower(dat3$`Stat Name`[rows])

# Grab the first record from each dataset
first <- dat3[!duplicated(dat3$index)]

# Drop DF and LOD if LOD units are not reported
first$lodvalue  <- as.numeric(first$`LOD Value`)
first$lodvalue[first$`LOD Units (from pulldown menu)`=="NR"] <- NA
rows <- first$`LOD Units (from pulldown menu)`=="ug/g"
rows[is.na(rows)] <- FALSE
first$lodvalue[rows] <- 1000*first$lodvalue[rows]
first$detecfrac  <- as.numeric(first$`DF Value`)
first$detecfrac[first$`LOD Units (from pulldown menu)`=="NR"] <- NA

# Create a matrix to contain the numerical data from each dataset on one line
```

```
x    <- plyr::count(dat3$stattype)
xx   <- x[!x$x %in% c("point", "standard error", "std. deviation"), ]
nxx  <- nrow(x)
dat4 <- as.data.frame(matrix(NA_real_, nrow=uniqueN(dat3$index), ncol=nxx+14))
dat4[, 1] <- first$`Study Reported Chemical Abbreviation`
dat4[, 2] <- first$`Standardized Chemical Name (from pulldown menu)`
dat4[, 3] <- first$`Indoor Environment Type (from pulldown menu)`
dat4[, 4] <- first$`Study Name (in Litstream)`
dat4[, 5] <- first$`Litstream ID`
dat4[, 6] <- first$`Scenario Description`
dat4[, 7] <- first$Country
dat4[, 8] <- first$`Medium  (from pulldown menu)`
dat4[, 9] <- first$`Medium Submatrix`
dat4[,10] <- unique(dat3$index)
dat4[,11] <- first$`Pooling Bin`
dat4[,12] <- first$`Sample n`
dat4[,13] <- first$detecfrac
dat4[,14] <- first$lodvalue
newcols   <- x$x[1:nxx]
setnames(dat4,c("chem","chemname","envir","study","litstream_id","scenario",
       "country","medium","submatrix","index","bin","n","df","lod", newcols))
fwrite(dat4, paste0(folder, "/Transposed dataset_A.csv"))

# copy in the stats from dat3 for the correct index number
for (i in 1:nrow(dat3)) {
  one <- dat3[i]
  row <- which(dat4$index==one$index)
  dat4[row, one$stattype] <- one$statval
}
```

```
# Get stats into percentiles (and median) where possible
dat4 <- as.data.table(dat4)
setorder(dat4, "chem", "envir")
rows <- is.na(dat4$median & !is.na(dat4$`geometric mean`))
dat4[rows]$median <- dat4[rows]$`geometric mean`

# convert DF and LOD to percentile values
rows <- !is.na(dat4$df) & dat4$df>0 & dat4$df<1
dat4$lodperc <- NA_real_
dat4[rows]$lodperc <- 100-100*dat4[rows]$df
dat4$lodval <- NA_real_
dat4[rows]$lodval <- dat4[rows]$lod

# Convert max to percentile using n
dat4$maxperc <- NA_real_
rows <- !is.na(dat4$max)
dat4[rows]$maxperc <- 100*(1-1/(dat4[rows]$n+1))

# Convert min to percentile using n (if df=1 or min>lod)
dat4$minperc <- NA_real_
rows <- !is.na(dat4$min) & (is.na(dat4$df) | dat4$df==1)
dat4[rows]$minperc  <- 100/(dat4[rows]$n+1)
dat4[!rows]$minperc <- NA_real_
dat4[!rows]$min     <- NA_real_

# Count number of percentiles per row
dat4$nperc <- 0
rows <- !is.na(dat4$`10th percentile`)
```

```
dat4[rows]$nperc <- dat4[rows]$nperc + 1
rows <- !is.na(dat4$`25th percentile`)
dat4[rows]$nperc <- dat4[rows]$nperc + 1
rows <- !is.na(dat4$`5th percentile`)
dat4[rows]$nperc <- dat4[rows]$nperc + 1
rows <- !is.na(dat4$`75th percentile`)
dat4[rows]$nperc <- dat4[rows]$nperc + 1
rows <- !is.na(dat4$`90th percentile`)
dat4[rows]$nperc <- dat4[rows]$nperc + 1
rows <- !is.na(dat4$`95th percentile`)
dat4[rows]$nperc <- dat4[rows]$nperc + 1
rows <- !is.na(dat4$median)
dat4[rows]$nperc <- dat4[rows]$nperc + 1
rows <- !is.na(dat4$lodperc)
dat4[rows]$nperc <- dat4[rows]$nperc + 1
rows <- !is.na(dat4$minperc)
dat4[rows]$nperc <- dat4[rows]$nperc + 1
rows <- !is.na(dat4$maxperc)
dat4[rows]$nperc <- dat4[rows]$nperc + 1
plyr::count(dat4$nperc)
fwrite(dat4, paste0(folder, "/Transposed dataset_B.csv"))

# Remove studies with fewer than 2 percentiles
dat5   <- dat4[dat4$nperc>=2]
badind <- dat4[dat4$nperc<2]$index
bad    <- base[base$index %in% badind, ..vars]
bad    <- bad[!duplicated(bad$index)]
bad$reason <- "Fewer than two percentiles"
setnames(bad, names(logfile))
```

```
logfile <- rbindlist(list(logfile, bad))


# Create vectors of percentiles and values and then sort
dat5$vvec <- list()
dat5$pvec <- list()
dat6 <- copy(dat5)
dat6$gm  <- NA_real_
dat6$gsd <- NA_real_
dat6$am  <- NA_real_
badind   <- list()
for (i in 1:nrow(dat5)) {
  x    <- dat5[i]
  j    <- x$index
  k    <- which(dat6$index==j)
  vtemp <- c(x$lodval, x$min, x$`5th percentile`, x$`10th percentile`, x$`25th percentile`,
         x$median, x$`75th percentile`, x$`90th percentile`, x$`95th percentile`, x$max)
  ptemp <- c(x$lodperc, x$minperc, 5, 10, 25, 50, 75, 90, 95, x$maxperc)
  ptemp[is.na(vtemp)] <- NA_real_
  badmin <- x$minperc!=min(ptemp, na.rm=TRUE)
  if(!is.na(badmin) & badmin==TRUE) {
   vtemp <- vtemp[-2]
   ptemp <- ptemp[-2]
  }
  badmax <- x$maxperc!=max(ptemp, na.rm=TRUE)
  if(!is.na(badmax) & badmax==TRUE) {
   vtemp <- vtemp[-length(vtemp)]
   ptemp <- ptemp[-length(ptemp)]
  }
  keep  <- !is.na(vtemp)
```

```
  vvec  <- vtemp[keep]
  pvec  <- ptemp[keep]
  dat5[i]$vvec[[1]] <- vvec
  dat5[i]$pvec[[1]] <- pvec
  if (all(order(pvec)==order(vvec))) {
    cat(" Dat5 row",i,", Dat6 row",k,", Index ",j," OK \n")
    dat6[k]$vvec[[1]] <- vvec
    dat6[k]$pvec[[1]] <- pvec
    y <- log_fit(vvec, pvec)
    dat6[k]$gm  <- y[1]
    dat6[k]$gsd <- y[2]
    dat6[k]$am  <- y[3]
  } else {
    cat(" Dat5 row",i,", Dat6 row",k,", Index ",j," inconsistent \n")
    badind <- c(badind, j)
    dat6 <- dat6[-k]
  }
}
bad    <- base[base$index %in% badind, ..vars]
bad    <- bad[!duplicated(bad$index)]
bad$reason <- "Inconsistent percentile order"
setnames(bad, names(logfile))
logfile <- rbindlist(list(logfile, bad))

# Analysis of datasets with point values
p1     <- pdat[pdat$`Stat Type (from pulldown list)`=="Point"]
pf     <- p1[!duplicated(p1$index)]
pn     <- dat6[1:nrow(pf), ]
pn[,  ] <- NA
```

```
pn[, 1] <- pf$`Study Reported Chemical Abbreviation`
pn[, 2] <- pf$`Standardized Chemical Name (from pulldown menu)`
pn[, 3] <- pf$`Indoor Environment Type (from pulldown menu)`
pn[, 4] <- pf$`Study Name (in Litstream)`
pn[, 5] <- pf$`Litstream ID`
pn[, 6] <- pf$`Scenario Description`
pn[, 7] <- pf$Country
pn[, 8] <- pf$`Medium  (from pulldown menu)`
pn[, 9] <- pf$`Medium Submatrix`
pn[,10] <- pf$index
pn[,11] <- pf$`Pooling Bin`
pn[,12] <- pf$`Sample n`
pn[,13] <- pf$detecfrac
pn[,14] <- pf$lodvalue
k     <- 0
for (j in pf$index) {
  k      <- k+1
  dataset <- p1[p1$index==j]
  vals    <- dataset$statval
  vals[is.na(vals)] <- dataset[1]$lodvalue/2
  logvals   <- log(vals)
  pn[k]$gm  <- exp(mean(logvals))
  pn[k]$gsd <- exp(sd(logvals))
  pn[k]$am  <- pn[k]$gm * exp(0.5*sd(logvals)^2)
}
dat6 <- rbind(dat6, pn)
setorder(dat6, "chem", "envir")
plyr::count(dat6$chemname)
fwrite(logfile, paste0(folder, "/logfile.csv"))
```

```
fwrite(dat6, paste0(folder, "/Lognormal_fits.csv"))



#  Read chemical data
chem <- read_xlsx(paste0(folder, "/PHOP P-CHEM Properties for Modeling_2023-08-21.xlsx"))
setnames(chem, 2, "chemname")
chem$fabs_dd <- 0.23
dat7    <- inner_join(dat6, chem, by="chemname")
dat7$koa <- 10^dat7$logkoa
dat7$kow <- 10^dat7$logkow
dat7$kaw <- dat7$kow / dat7$Koa
# Henry's law constant is in (atm-m3/mole) but is needed in (Pa-m3/mole)
dat7$h     <- dat7$h * 101325
dat7$h_unit <- "Pa-m3/mole"

# Function for approach #3 (exposure based on dust concentrations)
# Input Variables:
# tsp     = total suspended particulate (default = 20 ug/m3)
# fomd    = fraction organic matter in dust (default = 0.2)
# rhod    = density of dust particles (default = 2 g/cm3)
# fomp    = fraction organic matter in airborne particulates (default = 0.4)
# rhop    = density of airborne particles (default = 1 g/cm3)
# bw      = human body weight (kg)
# inh_vol  = volume of air inhaled (m3/day)
# ing_dust = dust ingestion rate (g/day)
# fabs_inh = absorption fraction for inhalation (-)
# fabs_ing = absorption fraction for ingestion (-)
# fhome   = fraction of time spent in the home (-)
# bsa     = body surface area (possibly adjusted for clothing, m2)
```

```
# vd      = air to skin mass transfer coefficient (default = 600 cm/hr)
# cfd     = chemical fraction (concentration) in dust (ng/g)
# mw      = molecular weight of chemical (g/mol)
# kow     = octanol-water partition coefficient (-) (not logKow)
# kaw     = air-water partition coefficient (-)    (not logKaw)
# koa     = octanol-air partition coefficient (-)   (not logKoa)
# h       = Henry's law coefficient (Pa m3/mol)
# f_inghand   = fraction of ingested dust coming from hands (-)
# f_handing   = fraction of dust on hands that gets ingested (-)
# fabs_dd  = absorption fraction for dermal dust (-)
#
dust_exp = function(tsp, fomd, rhod, fomp, rhop, bw, inh_vol, ing_dust,
                fabs_inh, fabs_ing, fabs_dd, fhome, bsa, vd, f_inghand,
                f_handing, cfd, mw, kow, kaw, koa, h) {
 if(is.na(kow)) {cat ("\n Kow required"); return(NULL) }
 if(is.na(kaw)) {
  if(!is.na(koa)) { kaw <- kow / koa
  } else if(!is.na(h)) { kaw <- h / (8.314 * 295)
  } else {cat("\n One of Kaw, Koa, or H required"); return(NULL)}
  # 8.314 = gas constant (Pa m3/mol K), 295 = temperature (K)
 }
 koa   <- kow / kaw
 # Now kow, kaw, koa should all be defined
 cg    <- 10^6 * rhod * cfd / (fomd * koa)
 # cg = air concentration as vapor
 # (ng/m3) = (cm3/m3)*(g/cm3)*(ng/g)/(-)/(-)
 cp    <- 10^-12 * cg * tsp * fomp * koa / rhop
 # cp = air concentration in particulates
 # (ng/m3) = ((m3/cm3)*(g/ug))*(ng/m3)*(ug/m3)*(-)/(g/cm3)
```

```
ca    <- cg + cp
# air concentration is sum of gas and particulate phases
adinh <- 10^-6 * ca * inh_vol * fhome * fabs_inh / bw
# adinh = absorbed dose from inhalation
# (mg/kg/day) = (mg/ng)*(ng/m3)*(m3/day)*(-)*(-)/(kg)
ading <- 10^-6 * cfd * ing_dust * fhome * fabs_ing / bw
# ading = absorbed dose from direct ingestion
# (mg/kg/day) = (mg/ng)*(ng/g)*(g/day)*(-)*(-)/(kg)
kpcw  <- 10^(0.7*log10(kow) - 0.0722*mw^(2/3) - 5.252) * 3600
# Kpcw equation taken from Mitro et al. 2016 page S10
b     <- kpcw * mw^(1/2) / 2.6
# b equation taken from Mitro et al. 2016 page S10
kpw   <- kpcw / (1+b)
# kpw equation taken from Mitro et al. 2016 page S10
kpb   <- kpw / kaw
# kpb is a variation of Mitro at al. 2026 (who used logKaw instead)
kpg   <- 1 / (1/vd + 1/kpb)
# kpg equation taken from Mitro et al. 2016 page S10 (cm/hr)
adats <- 10^-6 * cg * (kpg/100) * bsa * fhome * 24 / bw
# adats is dermal absorbed dose from air-to-skin deposition
# (mg/kg/day) = (mg/ng)*(ng/m3)*(m/hr)*(m2)*(-)*(hr/day)/(kg)
hand_ing <- ing_dust * f_inghand
# hand_ing = mass of dust (per day) ingested from mouthing hands
# (g/day) = (g/day)*(-)
dust_hands <- hand_ing / f_handing
# dust_hands = mass of dust getting on hands per day
# (g/day) = (g/day)/(-)
addd <- 10^-6 * (dust_hands - hand_ing) * cfd * fabs_dd / bw
# addd = absorbed dose from dermal dust (not air-to-skin)
```

```
  # (mg/kg/day) = (mg/ng)*(g/day)*(ng/g)*(-)/(kg)
  adtot <- adinh + ading + adats + addd
  # adtot is total absorbed dose across pathways in (mg/kg/day)

  return(c(cfd, cg, cp, ca, adinh, ading, adats, addd, adtot))
}

# Data defaults and assumptions
tsp      <- 20                              # total suspended particulate (default = 20 ug/m3)
fomd     <- 0.2                               # fraction organic matter in dust (default = 0.2)
rhod     <- 2                               # density of dust particles on floor (default = 2 g/cm3)
fomp     <- 0.4                               # fraction organic matter in airborne particulates (default)
rhop     <- 1                               # density of airborne particles (default = 1 g/cm3)
bw       <- c(7.8, 12.6, 18.6, 31.8, 56.8, 71.6, 80)      # human body weight (kg)
inh_vol  <- c(5.52, 8.4, 10.08, 12, 15.12, 16.32, 14.64)  # Daily inhalation volume while at home (m3)
ing_dust <- c(0.03, 0.04, 0.03, 0.03, 0.02, 0.02, 0.02)   # dust ingestion rate (g/day)
fabs_inh <- 0.5                                 # absorption fraction for inhalation (-) (assumption)
fhome    <- c(0.89, 0.82, 0.77, 0.74, 0.74, 0.71, 0.73)   # fraction of time spent in the home (-)  (CHAD)
bsa      <- c(0.199, 0.285, 0.38, 0.54, 0.795, 0.92, 0.98) # body surface area (possibly adjusted for
clothing, m2)
vd       <- 600                             # air to skin mass transfer coefficient (default = 600 cm/hr)
f_inghand <- 0.75                                # fraction of ingested dust coming from hands (-)  (assumption)
f_handing <- 0.05                                # fraction of dust on hands that gets ingested (-) (assumption)
nages    <- length(bw)

dat8     <- dat7[FALSE]
for (j in 1:nrow(dat7)) {
  for (a in 1:nages) {
    dat8    <- rbind(dat8, dat7[j])
```

```
 }
}
dat8$age   <- rep(1:nages, nrow(dat7))
dat8$cg    <- NA_real_    # air concentration (gas)
dat8$cp    <- NA_real_    # air concentration (particle)
dat8$ca    <- NA_real_    # air concentration (total)
dat8$adinh <- NA_real_    # absorbed dose (inhalation)
dat8$ading <- NA_real_    # absorbed dose (ingestion)
dat8$adats <- NA_real_    # absorbed dose (air-to-skin)
dat8$addd  <- NA_real_    # absorbed dose (dermal dust)
dat8$adtot <- NA_real_    # adsorbed dose (total)
dat8$agerange  <- NA_character_
dat8$lod_unit  <- "ng/g"
dat8$stat_unit <- "ng/g"
dat8$ad_unit   <- "mg/kg/day"

for (j in 1:nrow(dat7)) {
 fabs_dd  <- dat7[j]$fabs_dd
 cfd      <- dat7[j]$gm
 mw       <- dat7[j]$mw
 kow      <- dat7[j]$kow
 kaw      <- dat7[j]$kaw
 koa      <- dat7[j]$koa
 h        <- dat7[j]$h
 # Fit for GI tract absorption based on Fang and Stapleton, 2014
 fabs_ing <- min(max(1.7918 - 0.176*log10(kow), 0.3), 0.8)
 out      <- dust_exp(tsp, fomd, rhod, fomp, rhop, bw, inh_vol, ing_dust,
              fabs_inh, fabs_ing, fabs_dd, fhome, bsa, vd, f_inghand,
              f_handing, cfd, mw, kow, kaw, koa, h)
```

```
  for (a in 1:nages) {
    k            <- (j-1)*nages + a
    dat8[k]$cg    <- out[2]
    dat8[k]$cp    <- out[3]
    dat8[k]$ca    <- out[4]
    dat8[k]$adinh  <- out[4+a]
    dat8[k]$ading  <- out[4+nages+a]
    dat8[k]$adats  <- out[4+2*nages+a]
    dat8[k]$addd   <- out[4+3*nages+a]
    dat8[k]$adtot  <- out[4+4*nages+a]
  }
}

dat8$agerange[dat8$age==1] <- "under 1"
dat8$agerange[dat8$age==2] <- "1 - 2"
dat8$agerange[dat8$age==3] <- "3 - 5"
dat8$agerange[dat8$age==4] <- "6 - 10"
dat8$agerange[dat8$age==5] <- "11 - 15"
dat8$agerange[dat8$age==6] <- "16 - 20"
dat8$agerange[dat8$age==7] <- "21 and over"

# Results for each dataset (no pooling)
fwrite(dat8, paste0(folder, "/singles.csv"))

# Pooling of datasets
bins  <- unique(dat8$bin)
chems <- unique(dat8$chemname)
r     <- 0
adinh <- ading <- adats <- addd <- adtot <- rep(NA_real_, nages)
```

```
for (j in bins) {
  for (k in chems) {
    for (a in 1:nages) {
      dat  <- dat8[dat8$bin==j & dat8$chemname==k & dat8$age==a]
      if (nrow(dat)>0) {
        nrows <- nrow(dat)
        n     <- sum(dat$n)
        gm    <- exp(sum(dat$n*log(dat$gm))/n)
        cg    <- exp(sum(dat$n*log(dat$cg))/n)
        cp    <- exp(sum(dat$n*log(dat$cp))/n)
        ca    <- exp(sum(dat$n*log(dat$ca))/n)
        adinh <- exp(sum(dat$n*log(dat$adinh))/n)
        ading <- exp(sum(dat$n*log(dat$ading))/n)
        adats <- exp(sum(dat$n*log(dat$adats))/n)
        addd  <- exp(sum(dat$n*log(dat$addd ))/n)
        adtot <- exp(sum(dat$n*log(dat$adtot))/n)
      } else {
        gm <- cg <- cp <- ca <- adinh <- adats <- ading <- addd <- adtot <- NA_real_
        ndata <- 0
      }
      r <- r+1
      z <- data.table(bin=j, chem=k, age=a, gm=gm, cg=cg, cp=cp, ca=ca, ndata=nrows,
            adinh=adinh, ading=ading, adats=adats, addd=addd, adtot=adtot)
      if (r==1) dat9 <- z  else dat9 <- rbind(dat9, z)
      # After last age is done in each group, compute LADD
      if (a==nages) {
        y     <- dat9[dat9$bin==j & dat9$chem==k]
        x     <- y$adinh
        adinh <- (x[1]+2*x[2]+3*x[3]+5*x[4]+5*x[5]+5*x[6]+57*x[7]) / 78
```

```
      x    <- y$ading
      ading <- (x[1]+2*x[2]+3*x[3]+5*x[4]+5*x[5]+5*x[6]+57*x[7]) / 78
      x    <- y$adats
      adats <- (x[1]+2*x[2]+3*x[3]+5*x[4]+5*x[5]+5*x[6]+57*x[7]) / 78
      x    <- y$addd
      addd  <- (x[1]+2*x[2]+3*x[3]+5*x[4]+5*x[5]+5*x[6]+57*x[7]) / 78
      x    <- y$adtot
      adtot <- (x[1]+2*x[2]+3*x[3]+5*x[4]+5*x[5]+5*x[6]+57*x[7]) / 78
      z    <- data.table(bin=j,chem=k,age=8,gm=gm,cg=cg,cp=cp,ca=ca,ndata=nrows,
             adinh=adinh, ading=ading, adats=adats, addd=addd, adtot=adtot)
      dat9  <- rbind(dat9, z)
    }
   }
  }
}

dat9$agerange  <- NA_character_
dat9$agerange[dat9$age==1] <- "under 1"
dat9$agerange[dat9$age==2] <- "1 - 2"
dat9$agerange[dat9$age==3] <- "3 - 5"
dat9$agerange[dat9$age==4] <- "6 - 10"
dat9$agerange[dat9$age==5] <- "11 - 15"
dat9$agerange[dat9$age==6] <- "16 - 20"
dat9$agerange[dat9$age==7] <- "21 and over"
dat9$agerange[dat9$age==8] <- "Lifetime"
dat9$c_unit  <- "ng/m3"
dat9$ad_unit <- "mg/kg/day"
dat9$ncount  <-
```

```
fwrite(dat9, paste0(folder, "/pooled.csv"))


dat10 <- dat9[age==1 & bin!="Do not use"]

chems <- unique(dat10$chem)

dat11 <- as.data.table(data.frame(chem="hi",ce_cg=0, ce_rg=0, re_cg=0, re_rg=0, sc_cg=0, sc_rg=0,
ve_cg=0, ve_rg=0))[0]

for (j in 1:length(chems)) {

  chemj <- chems[[j]]

  dat   <- dat10[dat10$chem==chemj]

  ce_cg <- log(dat[bin=="Commercial Elevated Source Strength"]$gm/dat[bin=="Commercial General
Population"]$gm)

  ce_rg <- log(dat[bin=="Commercial Elevated Source Strength"]$gm/dat[bin=="Residential General
Population"]$gm)

  re_cg <- log(dat[bin=="Residential Elevated Source Strength"]$gm/dat[bin=="Commercial General
Population"]$gm)

  re_rg <- log(dat[bin=="Residential Elevated Source Strength"]$gm/dat[bin=="Residential General
Population"]$gm)

  sc_cg <- log(dat[bin=="School General Population"]$gm/dat[bin=="Commercial General
Population"]$gm)

  sc_rg <- log(dat[bin=="School General Population"]$gm/dat[bin=="Residential General
Population"]$gm)

  ve_cg <- log(dat[bin=="Vehicle"]$gm/dat[bin=="Commercial General Population"]$gm)

  ve_rg <- log(dat[bin=="Vehicle"]$gm/dat[bin=="Residential General Population"]$gm)

  rec   <- data.table(chem=chemj, ce_cg=ce_cg, ce_rg=ce_rg, re_cg=re_cg, re_rg=re_rg, sc_cg=sc_cg,

                  sc_rg=sc_rg, ve_cg=ve_cg, ve_rg=ve_rg)

  dat11 <- rbind(dat11, rec)

}

res1 <- colSums(dat11[, 2:9])/7

res2 <- colSums(abs(dat11[, 2:9]))/7
```