```
#############################
# this is for children between 2 and 19.9 years of age (24 to 239.9 months)
# computes z-score, modified z-score, extended BMIz (and percentiles)
# and other BMI metrics based on the CDC growth charts

# Note - need sex coded as 1 (boys) or 2 (girls)
# Also need weight (kg), height (cm), and months of age
# The names of weight, height and age can be anything in your data
# Also need the reference data for the LMS files

# About age in months:
# If there are decimal place in months of age, keep all digits
# If only completed months of age is known (as in NHANES data),
#   specify 'integer=T' (default is F) when calling the function
#   In this case, 0.5 will be added to the number of completed months
# if age is in years w decimal places, divide by 365.25
# if age is in days, convert to months =  age_days/(365.25/12)

# About the CDC reference values for the L M and S parameters:
# refdata_dir is folder that contains 'CDCref_d.csv' - this file can be download from
# https://www.cdc.gov/nccdphp/dnpao/growthcharts/resources/sas.htm
# (4th paragraph, before 'Instructions for SAS Users)

# specify the folder/directory of CDCref_d.csv in the ref_dir argument
# For example, ref_dir = '~/R/Growth_Charts/Data/'

# To use function:
# First, source the function. For example, source('~/Sync/R/Functions/ext_bmiz.R)
# Examples:
# For completed (integer) age in months (as in NHANES):
#    new <- ext_bmiz(yourdata_name, age=ridageex, integer=TRUE,
#                          wt=bmxwt,ht=bmxht, bmi=bmxbmi,
#                          ref_dir='~/R/Growth_Charts/Data/')
# Or for age in months with decimals omit the 'integer=T' or use 'integer=F':
#    d <- ext_bmiz(yourdata_name, age=agemos,
#                         wt=weight,ht=height, bmi=bmi,
#                         ref_dir='~/R/Growth_Charts/Data/')
# Or, d <- ext)bmiz(yourdata,agemos,wt,ht,bmi, ref_dir= 'whatever')

# agemos, weight, height and bmi are the names of these variables in your data
# When you run the function, do not put the variable names in quotation marks

# requires 5 packages:
require(Hmisc)
require(data.table)
require(magrittr)
require(dplyr)
require(labelled)


# Function:
ahb_ext_bmiz <- function(data,
              age='agemos',
              integer=FALSE, # if agemos is given as 'complete' months set to TRUE
              wt='weight', ht='height', bmi='bmi',
              ref_dir='~/Sync/R/Anal/Growth_Charts/Data/'
```

```
          )
{

# functions used in code:
set_cols_first <- function (DT, cols, intersection = TRUE) # thanks to hutils
  {
    if (intersection) {
      return(setcolorder(DT, c(intersect(cols, names(DT)),
        setdiff(names(DT), cols))))
    }
    else {
      return(setcolorder(DT, c(cols, setdiff(names(DT), cols))))
    }
  }

z_score=function(var, l, m, s){ # LMS formula with modified (m) z-scores
    ls=l*s; invl=1/l
    z = (((var/m) ^ l) -1) / (ls) # z-score forumula
    sdp2 = (m * (1 + 2*ls) ^ (invl)) - m; # modified z-score (+2)
    sdm2 = m - (m * (1 - 2*ls) ^ (invl));
    mz=fifelse(var < m, (var - m)/(0.5*sdm2), (var - m)/(sdp2*0.5) )
    list(z, mz)
  }

  setDT(data)

  data$seq_ <- 1L:nrow(data) # needed for merging back with original data
  dorig <- copy(data)
```

```
age <- deparse(substitute(age)); data$age <- data[[age]]
   if (integer)  data$age <- data$age + 0.5;
wt <- deparse(substitute(wt)); data$wt <- data[[wt]]
ht <- deparse(substitute(ht)); data$ht <- data[[ht]]
bmi <- deparse(substitute(bmi)); data$bmi <- data[[bmi]]

data <- data[between(age,24,240) & !(is.na(wt) & is.na(ht)),
                .(seq_, sex,age,wt,ht,bmi)];
v1 <- Cs(seq_,id,sex,age,wt,ht,bmi)

dref <- fread(paste0(ref_dir,'CDCref_d.csv'))[`_AGEMOS1`>23 & denom=='age']
names(dref) <- tolower(names(dref))
names(dref) <- gsub('^_', '', names(dref))

d20 <- dref[agemos2==240,
        .(sex,agemos2,lwt2,mwt2,swt2,lbmi2,mbmi2,sbmi2,lht2,mht2,sht2)]
names(d20) <- gsub('2','',names(d20));

dref <- dref[,.(sex,agemos1,lwt1,mwt1,swt1,lbmi1,mbmi1,sbmi1,lht1,mht1,sht1)]
names(dref) <- gsub('1','',names(dref));

dref=rbindlist(list(dref,d20))
adj_bmi_met <- dref[agemos==240,.(sex,mbmi,sbmi)] %>% setnames(.,Cs(sex,mref,sref))

dref <- dref[adj_bmi_met, on='sex']
v=Cs(sex,age,wl,wm,ws,bl,bm,bs,hl,hm,hs,mref,sref);  setnames(dref,v)

# interpolate reference data to match each agemos in input data
if (length(setdiff(data$age,dref$age))>0) {
```

```
uages=unique(data$age)
fapprox <- function(i){
  .d <- dref[sex==i]
     fapp <- function(vars,...)approx(.d$age,vars,xout=uages)$y
     data.frame(sapply(.d[,..v],fapp))
   }
dref <- rbindlist(lapply(1:2,fapprox))
}

#  fapprox <- function(df){
#      fapp <- function(vars,...)approx(df$age,vars,xout=uages)$y
#      data.frame(sapply(df,fapp))
#    }
# v=Cs(sex,age,wl,wm,ws,bl,bm,bs,hl,hm,hs,mref,sref)
# dlply(dref[,..v],~sex,fapprox) %>% rbindlist(.)

setkey(data,sex,age); setkey(dref,sex,age)
dt <- dref[data];
dt[,Cs(waz, mwaz):= z_score(dt$wt, dt$wl, dt$wm, dt$ws)]
dt[,Cs(haz, mhaz):= z_score(dt$ht, dt$hl, dt$hm, dt$hs)]
dt[,Cs(bz, mbz):= z_score(dt$bmi, dt$bl, dt$bm, dt$bs)]
# d[age<24.5,.(seq_,age,wt,ht,bz,bl,bm,bs,bz)]; d[age>239.5,.(seq_,age,wt,ht,bz,bl,bm,bs,bz)]

setDT(dt);  setnames(dt,Cs(bl,bm,bs),Cs(l,m,s))
dt[,Cs(wl,wm,ws,hl,hm,hs):=NULL]

dt=mutate(dt, bp=100*pnorm(bz),
   p95= m * (1 + l*s*qnorm(0.95))^(1 / l),
   p97= m * (1 + l*s*qnorm(0.97))^(1 / l),
   bmip95=100*(bmi/p95),
   wp=100*pnorm(waz),  hp=100*pnorm(haz),

      # other BMI metrics -- PMID 31439056
   z1=((bmi/m) - 1) / s,  # LMS formula when L=1: ((BMI/M)-1)/S
   dist1 = z1 * m * s, # unadjusted distance from median
   adist1 = z1 * sref * mref, # Adjusted (to age 20y) dist from median
   perc1 = z1 * 100 * s, # unadjusted %distance from median
   aperc1 = z1 * 100*sref, # adj %distance from median

   obese=1L*(bmi>=p95),
   sev_obese=1L*(bmip95>=120)
  ) %>% setDT()

## now create Extended z-score for BMI >=95th P
  dt[,':=' (ebz=bz, ebp=bp, agey=age/12)]
  dt[, sigma:=fifelse(sex==1, 0.3728 + 0.5196*agey - 0.0091*agey^2,
                 0.8334 + 0.3712*agey - 0.0011*agey^2)]
  dt[bp>=95, ebp:=90 + 10*pnorm((bmi - p95) / sigma)]
  dt[bp>=95, ebz:=qnorm(ebp/100)]
  dt[bp>99 & is.infinite(ebz), ebz:=8.21] # highest poss value is 8.20945

x <- Cs(agey,mref,sref,sex,wt,ht,bmi); dt[,(x):=NULL]
setnames(dt,Cs(adist1,  aperc1,  bp,   bz,  mbz,     mwaz,   mhaz,
          ebp,    ebz,    l,    m,    s),
      Cs(adj_dist1,adj_perc1,bmip,bmiz,mod_bmiz,mod_waz,mod_haz,
         ext_bmip,ext_bmiz, bmi_l,bmi_m,bmi_s)
)
```

```
# names(dt)

dt <- dt %>% labelled::set_variable_labels(
        bmiz = 'LMS BMI-for-sex/age z-score',
        bmip = 'LMS BMI-for-sex/age percentile',
        waz =  'LMS Weight-for-sex/age z-score',
        wp =  'LMS Weight-for-sex/age percentile',
        haz =  'LMS Height-for-sex/age z-score',
        hp =  'LMS Height-for-sex/age percentile',
        p95 = '95th percentile of BMI in growth charts',
        p97 = '97th percentile of BMI in growth charts',
        bmip95 = 'BMI as a percentage of the 95th percentile',
        mod_bmiz = 'Modified BMI-for-age z-score',
        mod_waz = 'Modified Weight-for-age z-score',
        mod_haz = 'Modified Height-for-age z-score',
        # bmi_l = 'BMI L parameter',
        # bmi_m = 'BMI M parameter',
        # bmi_s = 'BMI S parameter',
        sigma = 'Scale parameter for half-normal distribution',
        ext_bmip = 'Extended BMI percentile',
        ext_bmiz = 'Extended BMI z-score',
        sev_obese = 'BMI >= 120% of 95th percentile (0/1)',
        obese = 'BMI >= 95th percentile (0/1)'
)

v=Cs(seq_,bmiz,bmip,waz,wp,haz,hp,p95,p97,bmip95,mod_bmiz,mod_waz,mod_haz,
    # bmi_l,bmi_m,bmi_s,
    sigma,ext_bmip,ext_bmiz,sev_obese,obese)
dt <- dt[,..v]
```

```
  setkey(dt,seq_); setkey(dorig,seq_)

  dtot <- dt[dorig]

  set_cols_first(dtot,names(dorig))

  dtot[,Cs(seq_):=NULL]


  dtot[]

}
```