```r
##########################################
### Author: Dr. Wei-Wen Hsu  #############
###        University of Cincinnati  ####
###        hsuwe@uc.edu            #####
### Latest modification: 2/27/2024   #####
##########################################
#### Updated the FUE values | R Code     ##########
##########################################

rm(list=ls(all=TRUE))

#install.packages("haven")
#install.packages("survey")
#install.packages("jtools")
#install.packages("remotes")
#install.packages("writexl")
#install.packages("readxl")
#install.packages("basictabler")
#install.packages("openxlsx")
#install.packages("dplyr")
#install.packages("tidyverse")

#remotes::install_github("carlganz/svrepmisc")

#After the packages are downloaded, they need to be loaded. This needs to be done at the beginning of each R session.

library("haven")
```

```
library("survey")
library("jtools")
library("remotes")
library("svrepmisc")
library("readxl")
library(dplyr)
library(tidyverse)
library("writexl")
library(basictabler)
library(openxlsx)

# Litstream Data
#("C:\\Users\\hsuwe\\OneDrive - University of Cincinnati\\Desktop\\F\\Dr. Lynne - biomonitoring task\\Documents for calculat\\Task 16 PHOP Full extraction results from Litstream_10.25.23 cleanup.xlsx", sheet="")
path_litstream = "D:\\UC-OneDrive\\OneDrive - University of Cincinnati\\Desktop\\F\\Dr. Lynne - biomonitoring task\\Documents for calculations\\Task 16 PHOP Litstream data_MB data check_v3_MB.xlsx";
#path_litstream = "C:\\Users\\hsuwe\\OneDrive - University of Cincinnati\\Desktop\\F\\Dr. Lynne - biomonitoring task\\Documents for calculations\\Task 16 PHOP Litstream data_MB data check_v3_MB.xlsx";

## Location for the results
path_results ="D:\\UC-OneDrive\\OneDrive - University of Cincinnati\\Desktop\\F\\Dr. Lynne - biomonitoring task\\Results";
#path_results ="C:\\Users\\hsuwe\\OneDrive - University of Cincinnati\\Desktop\\F\\Dr. Lynne - biomonitoring task\\Results";

## As the results in the file named:
exceldata_name = "\\2. Litstream_data_wide_results.xlsx";
```

```
##
litsream_urine_FUE=read_excel(paste0(path_litstream), sheet='Fue')
litsream_urine_ulimits=read_excel(paste0(path_litstream), sheet="Urine limits")
litsream_urine_HBM=read_excel(paste0(path_litstream), sheet="Urine HBM data_Clean Population")

#View(litsream_urine_FUE)
#View(litsream_urine_ulimits)
#View(litsream_urine_HBM)

names(litsream_urine_FUE)
names(litsream_urine_ulimits)
names(litsream_urine_HBM)

### Remove some letters from the variable names
names(litsream_urine_FUE) = gsub(pattern = "flexData:|initialQuestions:|matrix:|object_13:|object_17:|studyPopulations:|object_20:|Biomarker:", replacement = "", x = names(litsream_urine_FUE))
names(litsream_urine_ulimits) = gsub(pattern = "flexData:|initialQuestions:|matrix:|object_13:|object_17:|studyPopulations:|object_20:|Biomarker:|urine_limits:", replacement = "", x = names(litsream_urine_ulimits))
names(litsream_urine_HBM) = gsub(pattern = "flexData:|initialQuestions:|matrix:|object_13:|object_17:|studyPopulations:|object_20:|Biomarker:", replacement = "", x = names(litsream_urine_HBM))



#names(litsream_urine_HBM) = gsub(pattern = "IDX_matrix\\|object_13\\|$", replacement = "", x = names(litsream_urine_HBM))



### preprocessing data
```

```
litsream_urine_HBM0 = litsream_urine_HBM
%>%mutate(statistics=ifelse(concentration=="Other",concentrationoth,concentration))

litsream_urine_HBM1 = litsream_urine_HBM0 %>%

                    dplyr::select(studyId,studyLitstreamId,year,location,sampyear,

                         urine_samptype,urine_correction,biomarker,urine_concunits,

                         urine_concunitsoth,avgweightvalue,

                         sampsize, percentfreq, statistics,value, 'Value for urinary flow rate (L/Day)',

                         'UFRBW (L/kg-day, UFR adjusted by BW)', 'Basic population',

                         'Detailed Population (Added+Edited)')


litsream_urine_ulimits0 = litsream_urine_ulimits%>%

                    dplyr::select(studyId, biomarker, sampyear, location, limit, value, comment) %>%

                     filter(!is.na(value))


### obtain the list of studies

study_id_list_HBM = names(table(litsream_urine_HBM1$studyId));

study_id_list_UL = names(table(litsream_urine_ulimits0 $studyId));


#####################################################

### Transform the data from long to wide      ######

#####################################################


### HBM data


wide_HBM_data = NULL;


for (study_id in study_id_list_HBM){

litsream_urine_HBM2= litsream_urine_HBM1 %>% filter(studyId==study_id)

litsream_urine_HBM3 = litsream_urine_HBM2 %>% spread(statistics, value)
```

```
wide_HBM_data = bind_rows(wide_HBM_data,litsream_urine_HBM3);

cat("study ID=", study_id,"\n")

}




############################

### Urine limits data   ####

############################


wide_UL_data =  NULL;

for (study_id in study_id_list_UL){

litsream_urine_ulimits1= litsream_urine_ulimits0 %>% filter(studyId==study_id)

litsream_urine_ulimits2 = litsream_urine_ulimits1 %>% spread(limit, value)

wide_UL_data = bind_rows(wide_UL_data,litsream_urine_ulimits2);

cat("study ID=", study_id,"\n")

}




#View(wide_HBM_data[88,])

names(wide_HBM_data)

#View(wide_UL_data[6,])

names(wide_UL_data)




##########################################

### merge HBM and Urine Limits data  ####

##########################################


## select the needed variables for merging
```

```
wide_UL_data1 = wide_UL_data %>% dplyr::select(studyId, biomarker, sampyear, location, MDL, LOD,
LOQ, max, min)

## merge and select the studies indicating "specific gravity"

urine_HBM_data =  wide_HBM_data %>% left_join(wide_UL_data1,
by=c("studyId","biomarker","sampyear","location")) %>%

                         filter(urine_correction == "Specific gravity")

#View(urine_HBM_data)

names(table(urine_HBM_data$studyId))  # the number of studies is 24.

table(urine_HBM_data$studyId, urine_HBM_data$biomarker) # a cross table to see how many studies
for each chemical

### Correct a typo in the raw dataset (studyId=3785967, sampsize=95 => sampsize=90
urine_HBM_data = urine_HBM_data  %>% mutate(sampsize = ifelse(studyId==3785967 & sampsize==95,
90, sampsize))

####################################################
####### Save as a physical data in excel here  ####
####################################################

write.xlsx(urine_HBM_data, file=paste0(path_results,exceldata_name),
```

```
        sheetName = "litstream_raw_data", colNames = TRUE, rowNames = F, append = FALSE)

####################################################




################################################
#### prepare the data for DI calculations  ####
################################################

# 1. rename variables
names(urine_HBM_data)
urine_HBM_data1 = urine_HBM_data %>%
                     rename("Geometric_mean"= "Geometric mean", "p95" = "95th percentile",
                           "p10" = "10th percentile", "p90"="90th percentile", "p25"="25th percentile",
"p75"="75th percentile",
                           "p50"="50th percentile/median","p5"="5th percentile",
                           "Arithmetic_mean"="Arithmetic mean", "Arithmetic_SD"="Arithmetic standard
deviation",
                           "Geometric_SD"="Geometric standard deviation", "UFRBW" = "UFRBW (L/kg-day, UFR
adjusted by BW)","Basic_population"="Basic population");

# 2. replace the value of "Geometric mean" if that indicated as NA, <LOD, Not calculated, low DF
#View(urine_HBM_data1%>%filter(Geometric_mean=="<LOD"))
#View(urine_HBM_data1%>%filter(Geometric_mean=="NA"))
#View(urine_HBM_data1%>%filter(Geometric_mean=="na"))
#View(urine_HBM_data1%>%filter(Geometric_mean=="Not calculated, low DF"))

urine_HBM_data2= urine_HBM_data1 %>%
  mutate(gmean = ifelse(is.na(Geometric_mean) & p50=="<LOD", NA,
```

```
                              ifelse(is.na(Geometric_mean) & !is.na(p50), p50,

                              ifelse(p50=="0.7" & Geometric_mean=="<LOD", p50,  ## p50=0.7 is from one
specific study. It is a shortcut for getting the correct value for p50 for that specific study.

          ifelse(Maximum=="<LOD" & Geometric_mean=="<LOD", NA,

          ifelse(Geometric_mean=="<LOD" & LOD=="2.0",
as.character(exp((log(as.numeric(Maximum))+ log(as.numeric(LOD)/sqrt(2)))/2)),

          ifelse(Geometric_mean=="NA" & Minimum=="<0.08",
as.character(exp((log(as.numeric(Maximum)) + log(as.numeric(MDL)/2))/2)),

                              ifelse(Geometric_mean=="na" & Minimum=="<MDL",
as.character(exp((log(as.numeric(Maximum)) + log(as.numeric(MDL)/2))/2)),

                              ifelse(Geometric_mean=="Not calculated, low DF" & Minimum=="<MDL",
as.character(exp((log(as.numeric(Maximum)) + log(as.numeric(MDL)/2))/2)),

Geometric_mean))))))))




table(urine_HBM_data2$Geometric_mean)

table(urine_HBM_data2$gmean)




urine_HBM_data2$gmean = as.numeric(urine_HBM_data2$gmean)


#View(urine_HBM_data2[c("Maximum","p50","Minimum","p25","p75","p95","p5",

#                    "Arithmetic_mean","Arithmetic_SD","Geometric_SD","urine_concunitsoth.y",
"MDL","LOD","LOQ", "Geometric_mean","gmean")])


# 3. checking 95th percentile;


urine_HBM_data3= urine_HBM_data2 %>% filter(!is.na(gmean) &  gmean!="NA")  # There is no enough
info for estimating "gmean" no matter what, then remove them from analysis


names(urine_HBM_data3)
```

```
table(urine_HBM_data3$p95)

#View(urine_HBM_data3%>%filter(p95=="NA"))

#View(urine_HBM_data3%>%filter(is.na(p95)))


### Handle special cases in the dataset

urine_HBM_data3$p10 =
ifelse(urine_HBM_data3$p10=="<MDL",as.character(as.numeric(urine_HBM_data3$MDL)/2),
urine_HBM_data3$p10)

urine_HBM_data3$Maximum = ifelse(urine_HBM_data3$Maximum=="16,989", as.character(16989),
urine_HBM_data3$Maximum)

urine_HBM_data3$Minimum = ifelse(urine_HBM_data3$Minimum=="<0.08", as.character(0.08),
urine_HBM_data3$Minimum)


###############################################################

### Needed Z scores for the estimation of 95th percentile   ###

###############################################################

z_75=qnorm(0.75,0,1)

z_90=qnorm(0.9,0,1)

z_95=qnorm(0.95,0,1)

z_995=qnorm(0.995,0,1)


#################################

### Estimate 95th percentile  ###

#################################



urine_HBM_data4= urine_HBM_data3 %>%

  mutate(p95_est = case_when(

    is.na(p95) & is.na(Geometric_SD) & Minimum=="<LOQ"   & !is.na(Maximum) & is.na(p10)   &
is.na(p90)     & is.na(p25)   & is.na(p75) & !is.na(LOQ)              ~ as.character( exp( log(gmean) +
z_95*(log(as.numeric(Maximum)/(as.numeric(LOQ)/2))/(2*z_995)) )),
```

```
            is.na(p95) & is.na(Geometric_SD) & Minimum=="<LOD" & !is.na(Maximum) & is.na(p25)   & 
is.na(p75)     ~ as.character( exp( log(gmean) + 
z_95*(log(as.numeric(Maximum)/(as.numeric(LOD)/sqrt(2)))/(2*z_995)) )),

            is.na(p95) & is.na(Geometric_SD) & Minimum=="<LOD" & !is.na(Maximum) & p25=="<LOD"   & 
p50=="<LOD"     ~ as.character( exp( log(gmean) + 
z_95*(log(as.numeric(Maximum)/(as.numeric(LOD)/sqrt(2)))/(2*z_995)) )),

      is.na(p95) & is.na(Geometric_SD) & Minimum=="<MDL"   & !is.na(Maximum) & is.na(p10)   & 
is.na(p90)     & is.na(p25)   & is.na(p75) & !is.na(MDL)       ~ as.character( exp( log(gmean) + 
z_95*(log(as.numeric(Maximum)/(as.numeric(MDL)/2))/(2*z_995)) )),

            is.na(p95) & is.na(Geometric_SD) & Minimum=="<MDL"   & !is.na(Maximum) & is.na(p10)   & 
is.na(p90)     & is.na(p25)   & is.na(p75) & is.numeric(as.numeric(p50))  & is.na(MDL)       ~ as.character( 
exp( log(gmean) + z_95*(log(as.numeric(Maximum)/as.numeric(p50))/(z_995)) )),

      is.na(p95) & is.na(Geometric_SD) & is.na(Minimum)& !is.na(p10)       & !is.na(p90)     & is.na(p25)   & 
is.na(p75)     ~ as.character( exp( log(gmean) + z_95*(log(as.numeric(p90)/as.numeric(p10))/(2*z_90)) )),

            is.na(p95) & is.na(Geometric_SD) & is.na(Minimum)  & is.na(p10)   & is.na(p90)     & is.na(p25)   
& is.na(p75)  & !is.na(p50)     & !is.na(Maximum)              ~ as.character( exp( log(gmean) + 
z_95*(log(as.numeric(Maximum)/as.numeric(p50))/(z_995)) )),

            is.na(p95) & is.na(Geometric_SD) & is.na(Minimum)  & is.na(p10)   & is.na(p90)     & 
p25=="<LOD"  &is.numeric(as.numeric(p75))   & !is.na(p50)     & !is.na(Maximum)              ~ as.character( 
exp( log(gmean) + z_95*(log(as.numeric(Maximum)/as.numeric(p50))/(z_995)) )),

      is.na(p95) & is.na(Geometric_SD) & is.numeric(as.numeric(Minimum))   & !is.na(Maximum) & 
is.na(p25)   & is.na(p75)     ~ as.character( exp( log(gmean) + 
z_95*(log(as.numeric(Maximum)/(as.numeric(Minimum)))/(2*z_995)) )),

            is.na(p95) & is.na(Geometric_SD) & is.numeric(as.numeric(p25))       & 
is.numeric(as.numeric(p75))                              ~ as.character( exp( log(gmean) + 
z_95*(log(as.numeric(p75)/as.numeric(p25))/(2*z_75)) )),

      is.na(p95) & is.na(Geometric_SD) & p50=="Not calculated, low DF"     & !is.na(Maximum)       ~ 
as.character( exp( log(gmean) + z_95*( log(as.numeric(Maximum)/gmean)/z_995 ) )),

            is.na(p95) & !is.na(Geometric_SD) & !is.na(gmean)                        ~ as.character( exp( log(gmean) 
+ z_95*(log(as.numeric(Geometric_SD))) )),

      TRUE ~ p95

    )

)



urine_HBM_data4$p95_est = as.numeric(urine_HBM_data4$p95_est);
```

```
### Estimate p95 for a special case

urine_HBM_data4 = urine_HBM_data4 %>% mutate(p95_est = ifelse(studyId==3785966 &
percentfreq==9  ,
exp(log(gmean)+z_95*log(exp(log(as.numeric(Maximum)/as.numeric(Minimum))/z_995))), p95_est));

urine_HBM_data4 = urine_HBM_data4 %>% mutate(p95_est = ifelse(studyId==3785966 &
percentfreq==11 ,
exp(log(gmean)+z_95*log(exp(log(as.numeric(Maximum)/as.numeric(Minimum))/z_995))), p95_est));



# Take a quick look

#View(urine_HBM_data4%>%filter(is.na(p95) )%>%
select("Minimum","p5","p10","p25","p50","p75","p90","p95","p95_est","Maximum","Geometric_SD","
MDL","LOD","LOQ","Geometric_mean","gmean"))



### Convert the unit to ug/L

## NOTE: 1. ng/mL = ug/L

##       2. pg/mL * 0.001 = ug/L


urine_HBM_data_final = urine_HBM_data4 %>% mutate(con_units = ifelse(urine_concunits=="Other",
urine_concunitsoth, urine_concunits) )

#View(urine_HBM_data_final[,c("urine_concunits","urine_concunitsoth","con_units")])

table(urine_HBM_data_final$con_units)



### Convert the unit of final gmean and final p95

urine_HBM_data_final = urine_HBM_data_final %>% mutate(final_gmean = ifelse(con_units == "pg/mL",
(gmean) * 0.001, gmean) ,

                                   final_p95   = ifelse(con_units == "pg/mL", (p95_est) * 0.001, p95_est) );
```

```
# take a quick look

#View(urine_HBM_data_final[,c("con_units","Geometric_mean","gmean","final_gmean","p95_est","final_p95")])


## Estimate Geometric SD for chemical concentration

urine_HBM_data_final = urine_HBM_data_final %>% mutate(final_GSD = ifelse(!is.na(Geometric_SD),Geometric_SD, exp(log(final_p95/final_gmean)/z_95)) )



### Estimate the Geometric SD for a special case

urine_HBM_data_final = urine_HBM_data_final %>% mutate(final_GSD = ifelse(studyId==3785966 & percentfreq==9 , exp(log(as.numeric(Maximum)/as.numeric(Minimum))/z_995), final_GSD));


### Make final_GSD numeric

urine_HBM_data_final$final_GSD=as.numeric(urine_HBM_data_final$final_GSD)



###############################################

######### Compute DI for each chemical  #########

###############################################


# Daily Intake (ug/day) = Urine concentration(ug/L) * urine volume (24-hour) (L/day) / FUE

# Daily Intake (ug/kg-day) = Urine concentration(ug/L) * urine volume (24-hour) (L/day) / (FUE * Body weight(kg))


################

### FUE    #####

################

#FUE_BCEP   = 0.2
```

```
#FUE_BDCIPP = 0.42
#FUE_BCIPP  = 0.39
#FUE_BCIPHIPP = 0.39
#FUE_BCIPP_BCIPHIPP = 0.78

############################################
### Updated FUE values as of 2/27/2024 ###
############################################
FUE_BCEP   = 0.13
FUE_BDCIPP = 0.23
FUE_BCIPP  = 0.23
FUE_BCIPHIPP = 0.35
FUE_BCIPP_BCIPHIPP = 0.58




# Convert "UFRBW" from character to numeric
# Note: UFRBW = (L/kg-day, UFR adjusted by BW)

urine_HBM_data_final$UFRBW = as.numeric(urine_HBM_data_final$UFRBW)


###############################################################
### IF esle sentence - identify studies by chemicals     ###
###############################################################
find_chemical <- function(x, chemical) {
grepl(paste0('\\b', chemical, '\\b', collapse = '|'), x)
}
```

```
urine_HBM_data_final$Find_BCEP   = 1*find_chemical(urine_HBM_data_final$biomarker, "BCEP")

urine_HBM_data_final$Find_BDCIPP = 1*find_chemical(urine_HBM_data_final$biomarker, "BDCIPP")

urine_HBM_data_final$Find_BCIPP  = 1*find_chemical(urine_HBM_data_final$biomarker, "BCIPP")

urine_HBM_data_final$Find_BCIPHIPP = 1*find_chemical(urine_HBM_data_final$biomarker,
"BCIPHIPP")

urine_HBM_data_final$Find_BCIPP_BCIPHIPP = 1*find_chemical(urine_HBM_data_final$biomarker,
c("BCIPP_BCIPHIPP"))


### select the studies that have BCIPP and BCIPHIPP

### NOTE: Five studies reported both chemicals


urine_HBM_data_final_two_chemicals = urine_HBM_data_final %>% filter(studyId %in%
c('3750768','3785966','3785967','3785972','3785974') & (Find_BCIPP==1 | Find_BCIPHIPP==1))


# checking:

# View(urine_HBM_data_final_two_chemicals)


# merge two chemical concentrations by sum

dat.merged <- urine_HBM_data_final_two_chemicals %>%

  dplyr::group_by(studyId,studyLitstreamId, Basic_population, UFRBW,sampsize, urine_concunits,
con_units, year, location,            sampyear, urine_samptype, urine_correction) %>%

  dplyr::summarise( gmean = sum(final_gmean), p95_est = sum(final_p95)) %>%

  mutate(Find_BCIPP_BCIPHIPP=1,biomarker = "BCIPP & BCIPHIPP",

        Basic_population = ifelse(studyId==3785974 & UFRBW < 0.036,"12-month Infants",
Basic_population))


dat.merged = dat.merged %>% mutate(final_GSD = exp( ( log(p95_est)-log(gmean) ) /z_95 ),

                              final_p95 = p95_est, final_gmean = gmean )
```

```
# View(dat.merged)


### select studies that have other chemicals

# urine_HBM_data_final_other_chemicals = urine_HBM_data_final %>% filter(!(studyId %in%
c('3750768','3785966','3785967','3785972','3785974') & (Find_BCIPP==1 | Find_BCIPHIPP==1)))

urine_HBM_data_final_other_chemicals = urine_HBM_data_final ;


# checking:

# View(urine_HBM_data_final_two_chemicals[,c("studyId","biomarker","Basic_population",
"UFRBW","sampsize","urine_concunits", "Find_BCEP" ,"Find_BDCIPP", "Find_BCIPP","Find_BCIPHIPP")])


urine_HBM_data_final0 = bind_rows(urine_HBM_data_final_other_chemicals,dat.merged)


# checking:

# View(urine_HBM_data_final0)


# Assign FUE based on chemical

urine_HBM_data_final = urine_HBM_data_final0 %>% mutate(FUE_chem =
ifelse(Find_BCIPP_BCIPHIPP==1, FUE_BCIPP_BCIPHIPP,

                                        ifelse(Find_BCEP==1, FUE_BCEP,

                                        ifelse(Find_BDCIPP==1, FUE_BDCIPP,

                                        ifelse(Find_BCIPP==1, FUE_BCIPP,

                                        ifelse(Find_BCIPHIPP==1, FUE_BCIPHIPP, NA_real_)))))) %>%

                              filter(!is.na(FUE_chem));  ## Remove other chemicals if we don't have their
FUE



# checking

# View(urine_HBM_data_final[,c("studyId","biomarker", "Find_BCEP" ,"Find_BDCIPP",
"Find_BCIPP","Find_BCIPHIPP","Find_BCIPP_BCIPHIPP","FUE_chem")])
```

```
## compute GM and p95


urine_HBM_data_final = urine_HBM_data_final %>% mutate(DI_GM = final_gmean * UFRBW /
FUE_chem,
                                              DI_p95 = final_p95 * UFRBW / FUE_chem);


urine_HBM_data_final = urine_HBM_data_final %>% mutate( DI_GSD = exp(log(DI_p95/DI_GM)/z_95) )


# checking

# View(urine_HBM_data_final[,c("biomarker", "Find_BCEP" ,"Find_BDCIPP",
"Find_BCIPP","Find_BCIPHIPP","DI_GM","DI_p95")])




##### By chemicals  #####

#View(urine_HBM_data_final[,c("studyId","biomarker", "Arithmetic_mean","Arithmetic_SD",
"Geometric_mean",
"Geometric_SD","urine_concunits","urine_concunitsoth","sampsize","percentfreq","final_gmean","final
_p95","UFRBW","DI_GM","DI_p95")])


by_chemicals = urine_HBM_data_final %>%

                 group_by(biomarker) %>%

                 summarise(n_DI_GM = n(), ave_DI_GM = mean(DI_GM,na.rm = TRUE), min_DI_GM =
min(DI_GM,na.rm = TRUE), max_DI_GM = max(DI_GM,na.rm = TRUE), n_DI_p95=sum(!is.na(DI_p95)),
ave_DI_p95 = mean(DI_p95,na.rm = TRUE), min_DI_p95 = min(DI_p95,na.rm = TRUE), max_DI_p95 =
max(DI_p95,na.rm = TRUE))




#### By biomarker and population  ####

by_chemicals_population = urine_HBM_data_final %>%

                      group_by(biomarker, Basic_population) %>%
```

```
                    summarise(n_DI_GM = n(), ave_DI_GM = mean(DI_GM,na.rm = TRUE), min_DI_GM = min(DI_GM,na.rm = TRUE), max_DI_GM = max(DI_GM,na.rm = TRUE), n_DI_p95=sum(!is.na(DI_p95)), ave_DI_p95 = mean(DI_p95,na.rm = TRUE), min_DI_p95 = min(DI_p95,na.rm = TRUE), max_DI_p95 = max(DI_p95,na.rm = TRUE))


### Correct NaN values in the table

by_chemicals_population$ave_DI_p95 = ifelse(is.na(by_chemicals_population$ave_DI_p95),NA,by_chemicals_population$ave_DI_p95)

by_chemicals_population$min_DI_p95 = ifelse(by_chemicals_population$min_DI_p95==Inf,NA,by_chemicals_population$min_DI_p95)

by_chemicals_population$max_DI_p95 = ifelse(by_chemicals_population$max_DI_p95==-Inf,NA,by_chemicals_population$max_DI_p95)




#####################################################

### Save results as a worksheet in the Excel file  ###

#####################################################


## Tab = By Chemicals ##


wb <- loadWorkbook(paste0(path_results,exceldata_name));


urine_HBM_data_final = urine_HBM_data_final%>% mutate(Estimated_con_GM = ifelse(is.na(as.numeric(Geometric_mean)), "Estimated", "Raw"),

                                    Estimated_con_p95 = ifelse(is.na(as.numeric(p95))& !is.na(p95_est), "Estimated",

                                              ifelse(is.na(as.numeric(p95))& is.na(p95_est), NA_real_,"Raw")),

                                    Estimated_con_GSD = ifelse(is.na(as.numeric(Geometric_SD))& !is.na(final_GSD), "Estimated",

                                              ifelse(as.numeric(Geometric_SD)==final_GSD, "Raw", NA_real_))
)
```

```
urine_HBM_data_final1 = urine_HBM_data_final %>%
dplyr::select("studyId","studyLitstreamId","year","location","sampyear","urine_samptype",

"urine_correction", "biomarker", "urine_concunits", "avgweightvalue", "sampsize", "percentfreq","Value
for urinary flow rate (L/Day)", "UFRBW",

"Basic_population", "Detailed Population (Added+Edited)", "p10", "p90", "Maximum" , "p50" ,
"Minimum","p25",

"p75", "p5", "p95",  "Arithmetic_mean", "Arithmetic_SD", "Geometric_mean",
"Geometric_SD","gmean", "p95_est","con_units",

"final_gmean", "Estimated_con_GM", "final_GSD",
"Estimated_con_GSD","final_p95","Estimated_con_p95","DI_GM", "DI_GSD", "DI_p95")



addWorksheet(wb,"LitStream_Daily_Intake")

writeData(wb, sheet = "LitStream_Daily_Intake", x = urine_HBM_data_final1, borders = "columns",
startCol = 1, startRow = 1)



sheet_name = "By Chemicals"


addWorksheet(wb,sheet_name)


## Table Caption

writeData(wb, sheet = sheet_name, x = "Unit: ug/kg-day", borders = "columns", startCol = 1, startRow =
1)


row_headers = paste0(by_chemicals[[1]])


tbl <- BasicTable$new()

tbl$addData(by_chemicals[,1:9], firstColumnAsRowHeaders=T,

        explicitColumnHeaders=c("Biomarker", "Valid rows for GM","Ave. Geometric mean","Min. GM",
"Max. GM","Valid rows for P95","Ave. 95th percentile", "Min. 95th p.", "Max. 95th p."));
```

```
# the arguments are (rFrom, cFrom, rTo, cTo, declarations)
tbl$setStyling(2, 2, 9, 2, declarations=list("xl-value-format"="###0"))
tbl$setStyling(2, 6, 9, 6, declarations=list("xl-value-format"="###0"))

tbl$setStyling(2, 3, 9, 5, declarations=list("xl-value-format"="##0.00000"))
tbl$setStyling(2, 7, 9, 9, declarations=list("xl-value-format"="##0.00000"))

tbl$writeToExcelWorksheet(wb=wb, wsName=sheet_name,
                     topRowNumber=2, leftMostColumnNumber=1,
                     applyStyles=T, mapStylesFromCSS=F, outputValuesAs="rawValue")


##############################################################

# by_chemicals_population

sheet_name1 = "By Chemicals_Populations";
addWorksheet(wb,sheet_name1)
writeData(wb, sheet = sheet_name1, x = "Unit: ug/kg-day", borders = "columns", startCol = 1, startRow = 1)


row_headers1 = paste0(by_chemicals_population[[1]])

tbl1 <- BasicTable$new()
tbl1$addData(by_chemicals_population[,1:10], firstColumnAsRowHeaders=T,
        explicitColumnHeaders=c("Biomarker","Basic Population","Valid rows for GM","Ave. Geometric mean","Min. GM", "Max. GM","Valid rows for P95","Ave. 95th percentile", "Min. 95th p.", "Max. 95th p."));
```

```
# the arguments are (rFrom, cFrom, rTo, cTo, declarations)
tbl1$setStyling(2, 3, 77, 3, declarations=list("xl-value-format"="###0"))
tbl1$setStyling(2, 7, 77, 7, declarations=list("xl-value-format"="###0"))

tbl1$setStyling(2, 4, 77,  6, declarations=list("xl-value-format"="##0.00000"))
tbl1$setStyling(2, 8, 77, 10, declarations=list("xl-value-format"="##0.00000"))

tbl1$writeToExcelWorksheet(wb=wb, wsName=sheet_name1,
                topRowNumber=2, leftMostColumnNumber=1,
                applyStyles=T, mapStylesFromCSS=F, outputValuesAs="rawValue")

saveWorkbook(wb, file=paste0(path_results,exceldata_name), overwrite = TRUE);



####################################
#########  The END       ##########
####################################




#################################################################
### Additional calculations for BCIPP + BCIPHIPP       #########
### For checking purpose                              #########
#################################################################

#View(urine_HBM_data_final_two_chemicals)
```

```
urine_HBM_data_final_two_chemicals= urine_HBM_data_final_two_chemicals%>%
mutate(Basic_population = ifelse(studyId==3785974 & UFRBW < 0.036,"12-month Infants",
Basic_population))


## 29-Phillips-2018 , Children


study_3750768 = urine_HBM_data_final_two_chemicals %>% filter(studyId==3750768)
#View(study_3750768 )
study_3750768_p10 = exp( ( log(sum(as.numeric(study_3750768$p10))) ))
study_3750768_p90 = exp( ( log(sum(as.numeric(study_3750768$p90))) ))
study_3750768_GM = sum(study_3750768$final_gmean)
study_3750768_p95 = sum(study_3750768$final_p95)
study_3750768_GSD = exp( ( log(study_3750768_p95)-log(study_3750768_GM) ) /z_95 )


## Results
study_3750768_p10
study_3750768_p90
study_3750768_GM
study_3750768_GSD
study_3750768_p95



## 72-Butt-2016 , Children


study_3785966c= urine_HBM_data_final_two_chemicals %>% filter(studyId==3785966 &
Basic_population == "Children")
#View(study_3785966c)


study_3785966c_GM = sum(study_3785966c$final_gmean)
study_3785966c_p95 = sum(study_3785966c$final_p95)
```

```
study_3785966c_GSD = exp( ( log(study_3785966c_p95)-log(study_3785966c_GM) ) /z_95 )


## Results
study_3785966c_GM
study_3785966c_GSD
study_3785966c_p95



## 72-Butt-2016 , Women


study_3785966w= urine_HBM_data_final_two_chemicals %>% filter(studyId==3785966 & Basic_population == "Women")
#View(study_3785966w)


study_3785966w_GM = sum(study_3785966w$final_gmean)
study_3785966w_p95 = sum(study_3785966w$final_p95)
study_3785966w_GSD = exp( ( log(study_3785966w_p95)-log(study_3785966w_GM) ) /z_95 )


## Results
study_3785966w_GM
study_3785966w_GSD
study_3785966w_p95


## 89-Gibson-2019 , Children


study_3785967c= urine_HBM_data_final_two_chemicals %>% filter(studyId==3785967 & Basic_population == "Children")
#View(study_3785967c)
```

```
study_3785967c_GM = sum(study_3785967c$final_gmean)

study_3785967c_p95 = sum(study_3785967c$final_p95)

study_3785967c_GSD = exp( ( log(study_3785967c_p95)-log(study_3785967c_GM) ) /z_95 )


## Results

study_3785967c_GM

study_3785967c_GSD

study_3785967c_p95



## 89-Gibson-2019 , Women


study_3785967w= urine_HBM_data_final_two_chemicals %>% filter(studyId==3785967 & Basic_population == "Women")

#View(study_3785967w)


study_3785967w_GM = sum(study_3785967w$final_gmean)

study_3785967w_p95 = sum(study_3785967w$final_p95)

study_3785967w_GSD = exp( ( log(study_3785967w_p95)-log(study_3785967w_GM) ) /z_95 )


## Results

study_3785967w_GM

study_3785967w_GSD

study_3785967w_p95



# 144-Hoffman-2017, Pregnant women

study_3785972= urine_HBM_data_final_two_chemicals %>% filter(studyId==3785972)

#View(study_3785972)
```

```r
study_3785972_GM = sum(study_3785972$final_gmean)
study_3785972_p95 = sum(study_3785972$final_p95)
study_3785972_GSD = exp( ( log(study_3785972_p95)-log(study_3785972_GM) ) /z_95 )

## Results
study_3785972_GM
study_3785972_GSD
study_3785972_p95



## 175-Hammel-2020 , 12-month Infants

study_3785974i12= urine_HBM_data_final_two_chemicals %>% filter(studyId==3785974 & Basic_population == "12-month Infants")
#View(study_3785974i12)

study_3785974i12_GM = sum(study_3785974i12$final_gmean)
study_3785974i12_p95 = sum(study_3785974i12$final_p95)
study_3785974i12_GSD = exp( ( log(study_3785974i12_p95)-log(study_3785974i12_GM) ) /z_95 )

## Results
study_3785974i12_GM
study_3785974i12_GSD
study_3785974i12_p95


## 175-Hammel-2020 , Infants
```

```
study_3785974i= urine_HBM_data_final_two_chemicals %>% filter(studyId==3785974 & Basic_population == "Infants")
#View(study_3785974i)


study_3785974i_GM = sum(study_3785974i$final_gmean)
study_3785974i_p95 = sum(study_3785974i$final_p95)
study_3785974i_GSD = exp( ( log(study_3785974i_p95)-log(study_3785974i_GM) ) /z_95 )


## Results
study_3785974i_GM
study_3785974i_GSD
study_3785974i_p95
```