```r
####################################################
### Author: Dr. Wei-Wen Hsu          #############
### University of Cincinnati          #############
### Division of Biostatistics and Bioinformatics ###
### hsuwe@uc.edu                    #############
### Latest modification: 2/24/2024    #############
####################################################

####################################################
###   Update the FUE values  | R Code        #######
####################################################

rm(list=ls(all=TRUE))

#install.packages("basictabler")
#install.packages("survey")
#install.packages("jtools")
#install.packages("remotes")
#install.packages("writexl")
#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)



#### The location of the physical Excel file  ####

path_NHANES = "D:\\UC-OneDrive\\OneDrive - University of Cincinnati\\Desktop\\F\\Dr. Lynne - biomonitoring task\\NHANES";

#path_NHANES = "C:\\Users\\hsuwe\\OneDrive - University of Cincinnati\\Desktop\\F\\Dr. Lynne - biomonitoring task\\NHANES";


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";


##### Import NHANES datasets  #####


# NHANES 11-12

N_data_11_12=read_excel(paste0(path_NHANES,"\\NHANES 2011-2012\\Excel dataset\\all1112.xlsx"))



# NHANES 13-14

N_data_13_14=read_excel(paste0(path_NHANES,"\\NHANES 2013-2014\\Excel dataset\\all1314.xlsx"))
```

```
# NHANES 15-16
N_data_15_16=read_excel(paste0(path_NHANES,"\\NHANES 2015-2016\\Excel dataset\\all1516.xlsx"))


# NHANES 17-18
N_data_17_18=read_excel(paste0(path_NHANES,"\\NHANES 2017-2018\\Excel dataset\\all1718.xlsx"))


# NHANES 19-20
N_data_17_20=read_excel(paste0(path_NHANES,"\\NHANES 2017-2020\\Excel dataset\\all1720.xlsx"))



#########################################################
##########  Create the new variables  ##################
#########################################################
#### variable names ####
#names(N_data_11_12);
#names(N_data_13_14);
#names(N_data_15_16);
#names(N_data_17_18);
#names(N_data_17_20);

####  Groups variables ####
N_data_11_12 = mutate(N_data_11_12, RIAGENDR=as.factor(RIAGENDR));
N_data_11_12 = mutate(N_data_11_12, RIDEXPRG=as.factor(RIDEXPRG));
N_data_11_12 = mutate(N_data_11_12, RIDRETH3=as.factor(RIDRETH3));
N_data_11_12 = mutate(N_data_11_12, RIDEXAGE_yr=RIDAGEYR);
```

```
N_data_13_14 = mutate(N_data_13_14, RIAGENDR=as.factor(RIAGENDR));
N_data_13_14 = mutate(N_data_13_14, RIDEXPRG=as.factor(RIDEXPRG));
N_data_13_14 = mutate(N_data_13_14, RIDRETH3=as.factor(RIDRETH3));
N_data_13_14 = mutate(N_data_13_14, RIDEXAGE_yr=RIDAGEYR);

N_data_15_16 = mutate(N_data_15_16, RIAGENDR=as.factor(RIAGENDR));
N_data_15_16 = mutate(N_data_15_16, RIDEXPRG=as.factor(RIDEXPRG));
N_data_15_16 = mutate(N_data_15_16, RIDRETH3=as.factor(RIDRETH3));
N_data_15_16 = mutate(N_data_15_16, RIDEXAGE_yr=RIDAGEYR);

N_data_17_18 = mutate(N_data_17_18, RIAGENDR=as.factor(RIAGENDR));
N_data_17_18 = mutate(N_data_17_18, RIDEXPRG=as.factor(RIDEXPRG));
N_data_17_18 = mutate(N_data_17_18, RIDRETH3=as.factor(RIDRETH3));
N_data_17_18 = mutate(N_data_17_18, RIDEXAGE_yr=RIDAGEYR);

N_data_17_20 = mutate(N_data_17_20, RIAGENDR=as.factor(RIAGENDR));
N_data_17_20 = mutate(N_data_17_20, RIDEXPRG=as.factor(RIDEXPRG));
N_data_17_20 = mutate(N_data_17_20, RIDRETH3=as.factor(RIDRETH3));
N_data_17_20 = mutate(N_data_17_20, RIDEXAGE_yr=RIDAGEYR);


## Age groups ##
N_data_11_12 = mutate(N_data_11_12, age_grp=(ifelse(RIDEXAGE_yr>=3 & RIDEXAGE_yr<=5, 1,
                        ifelse(RIDEXAGE_yr>5 & RIDEXAGE_yr<=11, 2,
                        ifelse(RIDEXAGE_yr>11 & RIDEXAGE_yr<=17, 3,
                        ifelse(RIDEXAGE_yr>=18, 4, NA_real_ ))))));

N_data_13_14 = mutate(N_data_13_14, age_grp=(ifelse(RIDEXAGE_yr>=3 & RIDEXAGE_yr<=5, 1,
                        ifelse(RIDEXAGE_yr>5 & RIDEXAGE_yr<=11, 2,
```

```
                              ifelse(RIDEXAGE_yr>11 & RIDEXAGE_yr<=17, 3,
                              ifelse(RIDEXAGE_yr>=18, 4, NA_real_ ))))));

N_data_15_16 = mutate(N_data_15_16, age_grp=(ifelse(RIDEXAGE_yr>=3 & RIDEXAGE_yr<=5, 1,
                              ifelse(RIDEXAGE_yr>5 & RIDEXAGE_yr<=11, 2,
                              ifelse(RIDEXAGE_yr>11 & RIDEXAGE_yr<=17, 3,
                              ifelse(RIDEXAGE_yr>=18, 4, NA_real_ ))))));

N_data_17_18 = mutate(N_data_17_18, age_grp=(ifelse(RIDEXAGE_yr>=3 & RIDEXAGE_yr<=5, 1,
                              ifelse(RIDEXAGE_yr>5 & RIDEXAGE_yr<=11, 2,
                              ifelse(RIDEXAGE_yr>11 & RIDEXAGE_yr<=17, 3,
                              ifelse(RIDEXAGE_yr>=18, 4, NA_real_ ))))));

N_data_17_20 = mutate(N_data_17_20, age_grp=(ifelse(RIDEXAGE_yr>=3 & RIDEXAGE_yr<=5, 1,
                              ifelse(RIDEXAGE_yr>5 & RIDEXAGE_yr<=11, 2,
                              ifelse(RIDEXAGE_yr>11 & RIDEXAGE_yr<=17, 3,
                              ifelse(RIDEXAGE_yr>=18, 4, NA_real_ ))))));


## Poverty Ratio ##
N_data_11_12 = mutate(N_data_11_12, poverty_ratio=(ifelse(INDFMPIR <1 & is.na(INDFMPIR)<1, 1,
                              ifelse(INDFMPIR >=1 & INDFMPIR <3 & is.na(INDFMPIR)<1, 2,
                              ifelse(INDFMPIR >=3 & is.na(INDFMPIR)<1, 3, NA_real_ )))));

N_data_13_14 = mutate(N_data_13_14, poverty_ratio=(ifelse(INDFMPIR <1 & is.na(INDFMPIR)<1, 1,
                              ifelse(INDFMPIR >=1 & INDFMPIR <3 & is.na(INDFMPIR)<1, 2,
                              ifelse(INDFMPIR >=3 & is.na(INDFMPIR)<1, 3, NA_real_ )))));

N_data_15_16 = mutate(N_data_15_16, poverty_ratio=(ifelse(INDFMPIR <1 & is.na(INDFMPIR)<1, 1,
```

```
                                    ifelse(INDFMPIR >=1 & INDFMPIR <3 & is.na(INDFMPIR)<1, 2,
                                    ifelse(INDFMPIR >=3 & is.na(INDFMPIR)<1, 3, NA_real_ )))));

N_data_17_18 = mutate(N_data_17_18, poverty_ratio=(ifelse(INDFMPIR <1 & is.na(INDFMPIR)<1, 1,
                                    ifelse(INDFMPIR >=1 & INDFMPIR <3 & is.na(INDFMPIR)<1, 2,
                                    ifelse(INDFMPIR >=3 & is.na(INDFMPIR)<1, 3, NA_real_ )))));

N_data_17_20 = mutate(N_data_17_20, poverty_ratio=(ifelse(INDFMPIR <1 & is.na(INDFMPIR)<1, 1,
                                    ifelse(INDFMPIR >=1 & INDFMPIR <3 & is.na(INDFMPIR)<1, 2,
                                    ifelse(INDFMPIR >=3 & is.na(INDFMPIR)<1, 3, NA_real_ )))));

## Adult Education Levels ##

## 2011-2016 use variables of DMDHREDU DMDHSEDU
#1      Less Than 9th Grade
#2      9-11th Grade (Includes 12th grade with no diploma)
#3      High School Grad/GED or Equivalent
#4      Some College or AA degree
#5      College Graduate or above
#7      Refused
#9      Don't Know

## 2017-     use variables of DMDHREDZ DMDHSEDZ

#1      Less than high school degree
#2      High school grad/GED or some college/AA degree
#3      College graduate or above
#7      Refused
#9      Don't Know
```

Recoding

```
N_data_11_12$DMDHREDZ  = as.numeric(recode(N_data_11_12$DMDHREDU, '1' = '1', '2' = '1', '3' = '2', '4'='2', '5'='3'));

N_data_11_12$DMDHSEDZ  = as.numeric(recode(N_data_11_12$DMDHSEDU, '1' = '1', '2' = '1', '3' = '2', '4'='2', '5'='3'));

N_data_11_12 = mutate(N_data_11_12, adult_edu= ifelse(DMDHREDZ>DMDHSEDZ,DMDHREDZ,DMDHSEDZ));



N_data_13_14$DMDHREDZ  = as.numeric(recode(N_data_13_14$DMDHREDU, '1' = '1', '2' = '1', '3' = '2', '4'='2', '5'='3'));

N_data_13_14$DMDHSEDZ  = as.numeric(recode(N_data_13_14$DMDHSEDU, '1' = '1', '2' = '1', '3' = '2', '4'='2', '5'='3'));

N_data_13_14 = mutate(N_data_13_14, adult_edu= ifelse(DMDHREDZ>DMDHSEDZ,DMDHREDZ,DMDHSEDZ));



N_data_15_16$DMDHREDZ  = as.numeric(recode(N_data_15_16$DMDHREDU, '1' = '1', '2' = '1', '3' = '2', '4'='2', '5'='3'));

N_data_15_16$DMDHSEDZ  = as.numeric(recode(N_data_15_16$DMDHSEDU, '1' = '1', '2' = '1', '3' = '2', '4'='2', '5'='3'));

N_data_15_16 = mutate(N_data_15_16, adult_edu= ifelse(DMDHREDZ>DMDHSEDZ,DMDHREDZ,DMDHSEDZ));



N_data_17_18 = mutate(N_data_17_18, adult_edu= ifelse(DMDHREDZ>DMDHSEDZ,DMDHREDZ,DMDHSEDZ));
```

Not present in 17-20 cycle

```
## New variable name of Biomarker in 2017-2018 and 2017-2020

N_data_17_18 = mutate(N_data_17_18, SSBDCPP= URXBDCP);

N_data_17_18 = mutate(N_data_17_18, SSBCPP = URXBCPP);

N_data_17_18 = mutate(N_data_17_18, SSBCEP = URXBCEP);

N_data_17_20 = mutate(N_data_17_20, SSBDCPP= URXBDCP);

N_data_17_20 = mutate(N_data_17_20, SSBCPP = URXBCPP);

N_data_17_20 = mutate(N_data_17_20, SSBCEP = URXBCEP);




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

#### Compute the Urine Flow Rate ####

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

# 1. number of urine measurements

N_data_11_12 = mutate(N_data_11_12, UFR_n=3-
(is.na(URDTIME1)*1+1*is.na(URDTIME2)+1*is.na(URDTIME3)))

N_data_13_14 = mutate(N_data_13_14, UFR_n=3-
(is.na(URDTIME1)*1+1*is.na(URDTIME2)+1*is.na(URDTIME3)))

N_data_15_16 = mutate(N_data_15_16, UFR_n=3-
(is.na(URDTIME1)*1+1*is.na(URDTIME2)+1*is.na(URDTIME3)))

N_data_17_18 = mutate(N_data_17_18, UFR_n=3-
(is.na(URDTIME1)*1+1*is.na(URDTIME2)+1*is.na(URDTIME3)))

N_data_17_20 = mutate(N_data_17_20, UFR_n=3-
(is.na(URDTIME1)*1+1*is.na(URDTIME2)+1*is.na(URDTIME3)))



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

##  Check the outliers for FLOW RATE 1  ###

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

# 2.use normal distribution to identify the outliers.
```

```r
N_data_11_12 = mutate(N_data_11_12,
URDFLOW1_out=ifelse(log(URDFLOW1)>quantile(log(URDFLOW1),0.975,na.rm = T) |
log(URDFLOW1)<quantile(log(URDFLOW1),0.025,na.rm = T),0,1))

N_data_13_14 = mutate(N_data_13_14,
URDFLOW1_out=ifelse(log(URDFLOW1)>quantile(log(URDFLOW1),0.975,na.rm = T) |
log(URDFLOW1)<quantile(log(URDFLOW1),0.025,na.rm = T),0,1))

N_data_15_16 = mutate(N_data_15_16,
URDFLOW1_out=ifelse(log(URDFLOW1)>quantile(log(URDFLOW1),0.975,na.rm = T) |
log(URDFLOW1)<quantile(log(URDFLOW1),0.025,na.rm = T),0,1))

N_data_17_18 = mutate(N_data_17_18,
URDFLOW1_out=ifelse(log(URDFLOW1)>quantile(log(URDFLOW1),0.975,na.rm = T) |
log(URDFLOW1)<quantile(log(URDFLOW1),0.025,na.rm = T),0,1))

N_data_17_20 = mutate(N_data_17_20,
URDFLOW1_out=ifelse(log(URDFLOW1)>quantile(log(URDFLOW1),0.975,na.rm = T) |
log(URDFLOW1)<quantile(log(URDFLOW1),0.025,na.rm = T),0,1))


#View(# see the outlier data of the variables

#N_data_13_14[,c("SEQN","URXVOL1","URDFLOW1","URDFLOW1_out","URDTIME1","URXVOL2","URDF
LOW2","URDTIME2", "URXVOL3", "URDFLOW3","URDTIME3","UFR_n")]

#)



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

## Compute FlowRate (mL/min) ##

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

# 3.compute FlowRate (mL/min)

N_data_11_12 = mutate(N_data_11_12,

 FlowRate=ifelse(UFR_n==3,(URXVOL1*URDFLOW1_out + URXVOL2 +
URXVOL3)/(URDTIME1*URDFLOW1_out + URDTIME2 + URDTIME3),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==0 &
1*is.na(URDTIME3)==1,(URXVOL1*URDFLOW1_out + URXVOL2 )/(URDTIME1*URDFLOW1_out +
URDTIME2 ),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==0 & 1*is.na(URDTIME3)==0,(URXVOL2 +
URXVOL3 )/(URDTIME2 + URDTIME3 ),
```

```
        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==1 &
1*is.na(URDTIME3)==0,(URXVOL1*URDFLOW1_out + URXVOL3 )/(URDTIME1*URDFLOW1_out +
URDTIME3 ),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==1 &
1*is.na(URDTIME3)==0,(URXVOL3)/(URDTIME3),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==0 &
1*is.na(URDTIME3)==1,(URXVOL2)/(URDTIME2),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==1 &
1*is.na(URDTIME3)==1,(URXVOL1*URDFLOW1_out)/(URDTIME1*URDFLOW1_out), NA_real_))))))));


N_data_13_14 = mutate(N_data_13_14,

 FlowRate=ifelse(UFR_n==3,(URXVOL1*URDFLOW1_out + URXVOL2 +
URXVOL3)/(URDTIME1*URDFLOW1_out + URDTIME2 + URDTIME3),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==0 &
1*is.na(URDTIME3)==1,(URXVOL1*URDFLOW1_out + URXVOL2 )/(URDTIME1*URDFLOW1_out +
URDTIME2 ),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==0 & 1*is.na(URDTIME3)==0,(URXVOL2 +
URXVOL3 )/(URDTIME2 + URDTIME3 ),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==1 &
1*is.na(URDTIME3)==0,(URXVOL1*URDFLOW1_out + URXVOL3 )/(URDTIME1*URDFLOW1_out +
URDTIME3 ),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==1 &
1*is.na(URDTIME3)==0,(URXVOL3)/(URDTIME3),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==0 &
1*is.na(URDTIME3)==1,(URXVOL2)/(URDTIME2),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==1 &
1*is.na(URDTIME3)==1,(URXVOL1*URDFLOW1_out)/(URDTIME1*URDFLOW1_out), NA_real_))))))));


N_data_15_16 = mutate(N_data_15_16,

 FlowRate=ifelse(UFR_n==3,(URXVOL1*URDFLOW1_out + URXVOL2 +
URXVOL3)/(URDTIME1*URDFLOW1_out + URDTIME2 + URDTIME3),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==0 &
1*is.na(URDTIME3)==1,(URXVOL1*URDFLOW1_out + URXVOL2 )/(URDTIME1*URDFLOW1_out +
URDTIME2 ),
```

```
        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==0 & 1*is.na(URDTIME3)==0,(URXVOL2 + 
URXVOL3 )/(URDTIME2 + URDTIME3 ),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==1 & 
1*is.na(URDTIME3)==0,(URXVOL1*URDFLOW1_out + URXVOL3 )/(URDTIME1*URDFLOW1_out + 
URDTIME3 ),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==1 & 
1*is.na(URDTIME3)==0,(URXVOL3)/(URDTIME3),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==0 & 
1*is.na(URDTIME3)==1,(URXVOL2)/(URDTIME2),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==1 & 
1*is.na(URDTIME3)==1,(URXVOL1*URDFLOW1_out)/(URDTIME1*URDFLOW1_out), NA_real_))))))));


N_data_17_18 = mutate(N_data_17_18,

 FlowRate=ifelse(UFR_n==3,(URXVOL1*URDFLOW1_out + URXVOL2 + 
URXVOL3)/(URDTIME1*URDFLOW1_out + URDTIME2 + URDTIME3),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==0 & 
1*is.na(URDTIME3)==1,(URXVOL1*URDFLOW1_out + URXVOL2 )/(URDTIME1*URDFLOW1_out + 
URDTIME2 ),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==0 & 1*is.na(URDTIME3)==0,(URXVOL2 + 
URXVOL3 )/(URDTIME2 + URDTIME3 ),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==1 & 
1*is.na(URDTIME3)==0,(URXVOL1*URDFLOW1_out + URXVOL3 )/(URDTIME1*URDFLOW1_out + 
URDTIME3 ),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==1 & 
1*is.na(URDTIME3)==0,(URXVOL3)/(URDTIME3),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==0 & 
1*is.na(URDTIME3)==1,(URXVOL2)/(URDTIME2),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==1 & 
1*is.na(URDTIME3)==1,(URXVOL1*URDFLOW1_out)/(URDTIME1*URDFLOW1_out), NA_real_))))))));


N_data_17_20 = mutate(N_data_17_20,

 FlowRate=ifelse(UFR_n==3,(URXVOL1*URDFLOW1_out + URXVOL2 + 
URXVOL3)/(URDTIME1*URDFLOW1_out + URDTIME2 + URDTIME3),
```

```
        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==0 &
1*is.na(URDTIME3)==1,(URXVOL1*URDFLOW1_out + URXVOL2 )/(URDTIME1*URDFLOW1_out +
URDTIME2 ),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==0 & 1*is.na(URDTIME3)==0,(URXVOL2 +
URXVOL3 )/(URDTIME2 + URDTIME3 ),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==1 &
1*is.na(URDTIME3)==0,(URXVOL1*URDFLOW1_out + URXVOL3 )/(URDTIME1*URDFLOW1_out +
URDTIME3 ),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==1 &
1*is.na(URDTIME3)==0,(URXVOL3)/(URDTIME3),

        ifelse(is.na(URDTIME1)*1==1 & 1*is.na(URDTIME2)==0 &
1*is.na(URDTIME3)==1,(URXVOL2)/(URDTIME2),

        ifelse(is.na(URDTIME1)*1==0 & 1*is.na(URDTIME2)==1 &
1*is.na(URDTIME3)==1,(URXVOL1*URDFLOW1_out)/(URDTIME1*URDFLOW1_out), NA_real_))))))));



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

## compute FlowRate (L/Day) ##

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

# 4. Convert to the right unit

N_data_11_12 = mutate(N_data_11_12, FlowRate_LD=FlowRate*60*24/1000);

N_data_13_14 = mutate(N_data_13_14, FlowRate_LD=FlowRate*60*24/1000);

N_data_15_16 = mutate(N_data_15_16, FlowRate_LD=FlowRate*60*24/1000);

N_data_17_18 = mutate(N_data_17_18, FlowRate_LD=FlowRate*60*24/1000);

N_data_17_20 = mutate(N_data_17_20, FlowRate_LD=FlowRate*60*24/1000);


#View(# see the data of the variables

#N_data_13_14[,c("SEQN","URXVOL1","URDFLOW1","URDTIME1","URDFLOW1_out","URXVOL2","URDF
LOW2","URDTIME2", "URXVOL3", "URDFLOW3","URDTIME3","UFR_n","FlowRate")]

#)
```

```
#############################
#### NHANSE STUDY DESIGN ####
#############################
## Use the sampling weight ##
nhc1112 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE, survey.lonely.psu = "adjust", data=N_data_11_12)
nhc1314 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE, survey.lonely.psu = "adjust", data=N_data_13_14)
nhc1516 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE, survey.lonely.psu = "adjust", data=N_data_15_16)
nhc1718 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE, survey.lonely.psu = "adjust", data=N_data_17_18)
#### Note: the variable name is different from the previous cycles. We use old variable name for computations. ####
N_data_17_20 = N_data_17_20 %>% mutate(WTMEC2YR=WTMECPRP);
nhc1720 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE, survey.lonely.psu = "adjust", data=N_data_17_20)


## Age: Average and SD
svymean(~RIDAGEYR, nhc1112 )
svysd(~RIDAGEYR,design = nhc1112 , na = TRUE)

svymean(~RIDAGEYR, nhc1314 )
svysd(~RIDAGEYR,design = nhc1314 , na = TRUE)

svymean(~RIDAGEYR, nhc1516 )
svysd(~RIDAGEYR,design = nhc1516 , na = TRUE)
```

```
svymean(~RIDAGEYR, nhc1718 )

svysd(~RIDAGEYR,design = nhc1718 , na = TRUE)


svymean(~RIDAGEYR, nhc1720 )

svysd(~RIDAGEYR,design = nhc1720 , na = TRUE)



## Average of weight

svymean(~BMXWT,design = nhc1112 , na = TRUE)

svymean(~BMXWT,design = nhc1314 , na = TRUE)

svymean(~BMXWT,design = nhc1516 , na = TRUE)

svymean(~BMXWT,design = nhc1718 , na = TRUE)

svymean(~BMXWT,design = nhc1720 , na = TRUE)


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

#####    GM of biomarker concentrations   #####

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

GM_biomarker = function (survey_design, years){


# BDCPP

output_name_SSBDCPP_gmean=paste0("SSBDCPP_gmean_",years);

assign(output_name_SSBDCPP_gmean, exp( svymean(~log(SSBDCPP),design = survey_design, na =
TRUE) ), envir = parent.frame());


# BCPP

output_name_SSBCPP_gmean=paste0("SSBCPP_gmean_",years);

assign(output_name_SSBCPP_gmean, exp( svymean(~log(SSBCPP),design = survey_design, na = TRUE) ),
envir = parent.frame());
```

```
# BCEP

output_name_SSBCEP_gmean=paste0("SSBCEP_gmean_",years);

assign(output_name_SSBCEP_gmean, exp( svymean(~log(SSBCEP),design = survey_design, na = TRUE) ),
envir = parent.frame());


}

GM_biomarker(survey_design=nhc1112 , years=1112)

GM_biomarker(survey_design=nhc1314 , years=1314)

GM_biomarker(survey_design=nhc1516 , years=1516)

GM_biomarker(survey_design=nhc1718 , years=1718)

GM_biomarker(survey_design=nhc1720 , years=1720)



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

### Compute individual GM of daily intake  ###

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

# create a function for this for each cycle

GM_concentration_ind =  function (years, dataset){


getput_name_BDCPP =paste0("SSBDCPP_gmean_",years);

GM_BDCPP = get(getput_name_BDCPP)[1]

ICC_BDCPP = 0.48

y_BDCPP= 0.5


getput_name_BCPP =paste0("SSBCPP_gmean_",years);

GM_BCPP = get(getput_name_BCPP)[1]

ICC_BCPP = 0.54

y_BCPP = 0.5
```

```
getput_name_BCEP =paste0("SSBCEP_gmean_",years);

GM_BCEP = get(getput_name_BCEP)[1]

ICC_BCEP = 0.45

y_BCEP = 0.5


dataset = mutate(dataset, GM_ICC_SSBDCPP=((SSBDCPP/GM_BDCPP)^(ICC_BDCPP^y_BDCPP))*
GM_BDCPP);

dataset = mutate(dataset, GM_ICC_SSBCPP =((SSBCPP /GM_BCPP )^(ICC_BCPP^y_BCPP))  * GM_BCPP);

dataset = mutate(dataset, GM_ICC_SSBCEP =((SSBCEP /GM_BCEP )^(ICC_BCEP^y_BCEP))  * GM_BCEP);


output_name=paste0("N_data_",substring(years,1,2),"_",substring(years,3,4));

assign(output_name, dataset, envir = parent.frame());


} # the end of the function "GM_concentration_ind"


GM_concentration_ind(years=1112, dataset = N_data_11_12)

GM_concentration_ind(years=1314, dataset = N_data_13_14)

GM_concentration_ind(years=1516, dataset = N_data_15_16)

GM_concentration_ind(years=1718, dataset = N_data_17_18)

GM_concentration_ind(years=1720, dataset = N_data_17_20)




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

## Target PHOP concentrations : BCPP BCEtP BDCPP or TCPP, TCEtP or TCEP, TDCPP ##

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


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

## compute Excretion Rates  ##
```

```
##############################

N_data_11_12 = mutate(N_data_11_12, ER_SSBDCPP=GM_ICC_SSBDCPP*FlowRate_LD/BMXWT);

N_data_11_12 = mutate(N_data_11_12, ER_SSBCPP=GM_ICC_SSBCPP*FlowRate_LD/BMXWT);

N_data_11_12 = mutate(N_data_11_12, ER_SSBCEP=GM_ICC_SSBCEP*FlowRate_LD/BMXWT);


N_data_13_14 = mutate(N_data_13_14, ER_SSBDCPP=GM_ICC_SSBDCPP*FlowRate_LD/BMXWT);

N_data_13_14 = mutate(N_data_13_14, ER_SSBCPP=GM_ICC_SSBCPP*FlowRate_LD/BMXWT);

N_data_13_14 = mutate(N_data_13_14, ER_SSBCEP=GM_ICC_SSBCEP*FlowRate_LD/BMXWT);


N_data_15_16 = mutate(N_data_15_16, ER_SSBDCPP=GM_ICC_SSBDCPP*FlowRate_LD/BMXWT);

N_data_15_16 = mutate(N_data_15_16, ER_SSBCPP=GM_ICC_SSBCPP*FlowRate_LD/BMXWT);

N_data_15_16 = mutate(N_data_15_16, ER_SSBCEP=GM_ICC_SSBCEP*FlowRate_LD/BMXWT);


N_data_17_18 = mutate(N_data_17_18, ER_SSBDCPP=GM_ICC_SSBDCPP*FlowRate_LD/BMXWT);

N_data_17_18 = mutate(N_data_17_18, ER_SSBCPP=GM_ICC_SSBCPP*FlowRate_LD/BMXWT);

N_data_17_18 = mutate(N_data_17_18, ER_SSBCEP=GM_ICC_SSBCEP*FlowRate_LD/BMXWT);


N_data_17_20 = mutate(N_data_17_20, ER_SSBDCPP=GM_ICC_SSBDCPP*FlowRate_LD/BMXWT);

N_data_17_20 = mutate(N_data_17_20, ER_SSBCPP=GM_ICC_SSBCPP*FlowRate_LD/BMXWT);

N_data_17_20 = mutate(N_data_17_20, ER_SSBCEP=GM_ICC_SSBCEP*FlowRate_LD/BMXWT);




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

########### GM of GM-ICC-adjusted concentration for each chemical in 2017-18  ################

###########  Estimate distributional parameters for distributional analysis   ################

#####################################################################################
#########
```

```
#N_data_17_18 = mutate(N_data_17_18, age_grp2 =(ifelse(RIDEXAGE_yr>=6 & RIDEXAGE_yr<=11, 1,
#                                   ifelse(RIDEXAGE_yr>11 & RIDEXAGE_yr<=19, 2,
#                                   ifelse(RIDEXAGE_yr>19 & RIDEXAGE_yr<=39, 3,
#                                   ifelse(RIDEXAGE_yr>39 & RIDEXAGE_yr<=59, 4,
#                                   ifelse(RIDEXAGE_yr>59 , 5, NA_real_ )))))));

# Use age_grp
age_grp_list = c("3-5 yrs","6-11 yrs","12-17 yrs","18+ yrs")

## 2017-2018 UFRBW
N_data_17_18 = N_data_17_18 %>% mutate(UFRBW = FlowRate_LD/BMXWT)

## Update the survey design dataset
nhc1718 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data=N_data_17_18)

GM_UFRBW_tmp = svyby(~log(UFRBW), by=~age_grp ,design = nhc1718 , svymean , na = TRUE) ;

GM_UFRBW =data.frame(GM_UFRBW_tmp[[1]],exp(GM_UFRBW_tmp[[2]]) )
names(GM_UFRBW) = c("subgroups", "GM_UFRBW" )

datalist <- list()
for(i in levels(as.factor(N_data_17_18$age_grp)) ) {
  df_subset <- N_data_17_18[which(N_data_17_18$age_grp == i), ]

  design <- svydesign(id = ~SDMVPSU, weights=~WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data = df_subset)

  dat <- data.frame(as.data.frame(exp(sqrt(svyvar(~log(UFRBW), na.rm= TRUE, design = design )))[[1]]))
```

```
  colnames(dat) <- c("sd.variable")
  datalist[[i]] <- dat
}

GSD_UFRBW <- as.data.frame(do.call(rbind, datalist)) %>%
  rownames_to_column() %>%
  rename(subgroups = rowname)




### BCEP

gmean_tmp = NULL;
gmean_tmp = svyby(~log(GM_ICC_SSBCEP), by=~age_grp ,design = nhc1718 , svymean , na = TRUE) ;

GM_GM_ICC_BCEP =data.frame(gmean_tmp[[1]],exp(gmean_tmp[[2]]) )
names(GM_GM_ICC_BCEP ) = c("subgroups", "GM_BCEP" )
datalist <- list()
for(i in levels(as.factor(N_data_17_18$age_grp)) ) {
  df_subset <- N_data_17_18[which(N_data_17_18$age_grp == i), ]

  design <- svydesign(id = ~SDMVPSU, weights=~WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data = df_subset)

  dat <- data.frame(as.data.frame(exp(sqrt(svyvar(~log(GM_ICC_SSBCEP), na.rm= TRUE, design = design
)))[[1]]))

  colnames(dat) <- c("sd.variable")
```

```
  datalist[[i]] <- dat
}

GSD_GM_ICC_BCEP <- as.data.frame(do.call(rbind, datalist)) %>%
  rownames_to_column() %>%
  rename(subgroups = rowname)



### SSBCPP

gmean_tmp = NULL;
gmean_tmp = svyby(~log(GM_ICC_SSBCPP), by=~age_grp ,design = nhc1718 , svymean , na = TRUE) ;
GM_GM_ICC_SSBCPP = data.frame(gmean_tmp[[1]],exp(gmean_tmp[[2]]) )
 names(GM_GM_ICC_SSBCPP ) = c("subgroups", "GM_SSBCPP" )
datalist <- list()
for(i in levels(as.factor(N_data_17_18$age_grp)) ) {
  df_subset <- N_data_17_18[which(N_data_17_18$age_grp == i), ]

  design <- svydesign(id = ~SDMVPSU, weights=~WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data = df_subset)

  dat <- data.frame(as.data.frame(exp(sqrt(svyvar(~log(GM_ICC_SSBCPP), na.rm= TRUE, design = design
)))[[1]]))

  colnames(dat) <- c("sd.variable")
  datalist[[i]] <- dat
}

GSD_GM_ICC_SSBCPP<- as.data.frame(do.call(rbind, datalist)) %>%
```

```
  rownames_to_column() %>%
  rename(subgroups = rowname)

### SSBDCPP

gmean_tmp = NULL;
gmean_tmp = svyby(~log(GM_ICC_SSBDCPP ), by=~age_grp ,design = nhc1718 , svymean , na = TRUE) ;
GM_GM_ICC_SSBDCPP = data.frame(gmean_tmp[[1]],exp(gmean_tmp[[2]]) )
names(GM_GM_ICC_SSBDCPP) = c("subgroups", "GM_SSBDCPP" )

datalist <- list()
for(i in levels(as.factor(N_data_17_18$age_grp)) ) {
  df_subset <- N_data_17_18[which(N_data_17_18$age_grp == i), ]

  design <- svydesign(id = ~SDMVPSU, weights=~WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data = df_subset)

  dat <- data.frame(as.data.frame(exp(sqrt(svyvar(~log(GM_ICC_SSBDCPP), na.rm= TRUE, design = design
)))[[1]]))

  colnames(dat) <- c("sd.variable")
  datalist[[i]] <- dat
}

GSD_GM_ICC_SSBDCPP <- as.data.frame(do.call(rbind, datalist)) %>%
  rownames_to_column() %>%
  rename(subgroups = rowname)
```

```
###########################################################
#### Save the results for distributional analysis   #######
###########################################################
result_distri =as.data.frame(cbind(GM_UFRBW,GSD_UFRBW[[2]],GM_GM_ICC_BCEP[[2]],
GSD_GM_ICC_BCEP[[2]], GM_GM_ICC_SSBCPP[[2]],
                                GSD_GM_ICC_SSBCPP[[2]], GM_GM_ICC_SSBDCPP[[2]],
GSD_GM_ICC_SSBDCPP[[2]]));
result_distri$subgroups= age_grp_list;


write.xlsx(result_distri, file=paste0(path_results,"\\NHANES_parameters_for_distri.xlsx"),
        sheetName = "Parameters from 17-18", colNames = TRUE, rowNames = TRUE, append = FALSE)


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


##############################
## Compute the daily intake ##
##############################


## Best estimated numbers of FUE as of 11/6/2023 ##
#FUE_SSBDCPP= 0.42;
#FUE_SSBCPP = 0.39;
#FUE_SSBCEP = 0.2;



## New estimated values of FUE as of 2/24/2024 ##
FUE_SSBDCPP= 0.23;
FUE_SSBCPP = 0.23;
FUE_SSBCEP = 0.13;
```

```
bioavilability=1;

N_data_11_12 = mutate(N_data_11_12, DI_SSBDCPP =  ER_SSBDCPP * (1/FUE_SSBDCPP) * 1/bioavilability)

N_data_11_12 = mutate(N_data_11_12, DI_SSBCPP  =  ER_SSBCPP  * (1/FUE_SSBCPP) * 1/bioavilability)

N_data_11_12 = mutate(N_data_11_12, DI_SSBCEP  =  ER_SSBCEP  * (1/FUE_SSBCEP) * 1/bioavilability)

N_data_13_14 = mutate(N_data_13_14, DI_SSBDCPP =  ER_SSBDCPP * (1/FUE_SSBDCPP) * 1/bioavilability)

N_data_13_14 = mutate(N_data_13_14, DI_SSBCPP  =  ER_SSBCPP  * (1/FUE_SSBCPP) * 1/bioavilability)

N_data_13_14 = mutate(N_data_13_14, DI_SSBCEP  =  ER_SSBCEP  * (1/FUE_SSBCEP) * 1/bioavilability)

N_data_15_16 = mutate(N_data_15_16, DI_SSBDCPP =  ER_SSBDCPP * (1/FUE_SSBDCPP) * 1/bioavilability)

N_data_15_16 = mutate(N_data_15_16, DI_SSBCPP  =  ER_SSBCPP  * (1/FUE_SSBCPP) * 1/bioavilability)

N_data_15_16 = mutate(N_data_15_16, DI_SSBCEP  =  ER_SSBCEP  * (1/FUE_SSBCEP) * 1/bioavilability)

N_data_17_18 = mutate(N_data_17_18, DI_SSBDCPP =  ER_SSBDCPP * (1/FUE_SSBDCPP) * 1/bioavilability)

N_data_17_18 = mutate(N_data_17_18, DI_SSBCPP  =  ER_SSBCPP  * (1/FUE_SSBCPP) * 1/bioavilability)

N_data_17_18 = mutate(N_data_17_18, DI_SSBCEP  =  ER_SSBCEP  * (1/FUE_SSBCEP) * 1/bioavilability)

N_data_17_20 = mutate(N_data_17_20, DI_SSBDCPP =  ER_SSBDCPP * (1/FUE_SSBDCPP) * 1/bioavilability)

N_data_17_20 = mutate(N_data_17_20, DI_SSBCPP  =  ER_SSBCPP  * (1/FUE_SSBCPP) * 1/bioavilability)

N_data_17_20 = mutate(N_data_17_20, DI_SSBCEP  =  ER_SSBCEP  * (1/FUE_SSBCEP) * 1/bioavilability)
```

```
# See whether the log(chemical concentration) is normally distributed

#hist(log(N_data_13_14$ER_SSBDCPP))

hist(log(N_data_13_14$DI_SSBDCPP))



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

##  Update survey design with new N_data  ###

##  Use the sampling weight             ####

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

nhc1112 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data=N_data_11_12)

nhc1314 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data=N_data_13_14)

nhc1516 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data=N_data_15_16)

nhc1718 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data=N_data_17_18)

#### Note: the variable name is different from the previous cycles. We use old variable name for
computations. ####

N_data_17_20 = N_data_17_20 %>% mutate(WTMEC2YR=WTMECPRP);

nhc1720 <- svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data=N_data_17_20)



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

## Overall Population ##

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


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

###  Quantile settings   ###
```

```
###########################

quantile_list = c(.25,.5,.75,.9,.95);

q_list = as.character(paste(100*quantile_list,"th percentile"))


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

# Define a function to compute all needed statistics  ##

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

statistics_all = function (survey_design, years){


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

### Arithmetic mean and SD ###

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

# BDCPP

output_name_SSBDCPP_mean=paste0("DI_SSBDCPP_mean_",years);

assign(output_name_SSBDCPP_mean, svymean(~DI_SSBDCPP,design = survey_design, na = TRUE), envir
= parent.frame());


output_name_SSBDCPP_sd=paste0("DI_SSBDCPP_sd_",years);

assign(output_name_SSBDCPP_sd, svysd(~DI_SSBDCPP, design = survey_design, na = TRUE), envir =
parent.frame());


# BCPP

output_name_SSBCPP_mean=paste0("DI_SSBCPP_mean_",years);

assign(output_name_SSBCPP_mean, svymean(~DI_SSBCPP,design = survey_design, na = TRUE), envir =
parent.frame());


output_name_SSBCPP_sd=paste0("DI_SSBCPP_sd_",years);

assign(output_name_SSBCPP_sd, svysd(~DI_SSBCPP, design = survey_design, na = TRUE), envir =
parent.frame());
```

```
# BCEP

output_name_SSBCEP_mean=paste0("DI_SSBCEP_mean_",years);

assign(output_name_SSBCEP_mean, svymean(~DI_SSBCEP,design = survey_design, na = TRUE), envir =
parent.frame());


output_name_SSBCEP_sd=paste0("DI_SSBCEP_sd_",years);

assign(output_name_SSBCEP_sd, svysd(~DI_SSBCEP, design = survey_design, na = TRUE), envir =
parent.frame());



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

### Geometric mean and sd  ###

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

# BDCPP

output_name_SSBDCPP_gmean=paste0("DI_SSBDCPP_gmean_",years);

assign(output_name_SSBDCPP_gmean, exp( svymean(~log(DI_SSBDCPP),design = survey_design, na =
TRUE) ), envir = parent.frame());


output_name_SSBDCPP_gsd=paste0("DI_SSBDCPP_gsd_",years);

assign(output_name_SSBDCPP_gsd, exp( svysd(~log(DI_SSBDCPP), design = survey_design, na = TRUE) ),
envir = parent.frame());


# BCPP

output_name_SSBCPP_gmean=paste0("DI_SSBCPP_gmean_",years);

assign(output_name_SSBCPP_gmean, exp( svymean(~log(DI_SSBCPP),design = survey_design, na =
TRUE) ), envir = parent.frame());


output_name_SSBCPP_gsd=paste0("DI_SSBCPP_gsd_",years);

assign(output_name_SSBCPP_gsd, exp( svysd(~log(DI_SSBCPP), design = survey_design, na = TRUE) ),
envir = parent.frame());
```

```
# BCEP

output_name_SSBCEP_gmean=paste0("DI_SSBCEP_gmean_",years);

assign(output_name_SSBCEP_gmean, exp( svymean(~log(DI_SSBCEP),design = survey_design, na = TRUE) ), envir = parent.frame());


output_name_SSBCEP_gsd=paste0("DI_SSBCEP_gsd_",years);

assign(output_name_SSBCEP_gsd, exp( svysd(~log(DI_SSBCEP), design = survey_design, na = TRUE) ), envir = parent.frame());




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

### Quantiles      ###

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

#quantile_list = c(.25,.5,.75,.9,.95);

#q_list = as.character(paste(100*quantile_list,"th percentile"))


#assign("quantile_list", quantile_list)

#assign("q_list", q_list)


# BDCPP

output_name_SSBDCPP_quantile=paste0("DI_SSBDCPP_quantile_",years);

assign(output_name_SSBDCPP_quantile, data.frame(svyquantile(~DI_SSBDCPP, design = survey_design, na = TRUE, quantile_list,ci=F)[1]), envir = parent.frame());


# BCPP

output_name_SSBCPP_quantile=paste0("DI_SSBCPP_quantile_",years);

assign(output_name_SSBCPP_quantile, data.frame(svyquantile(~DI_SSBCPP, design = survey_design, na = TRUE, quantile_list,ci=F)[1]), envir = parent.frame());
```

```
# BCEP

output_name_SSBCEP_quantile=paste0("DI_SSBCEP_quantile_",years);

assign(output_name_SSBCEP_quantile, data.frame(svyquantile(~DI_SSBCEP, design = survey_design, na = TRUE, quantile_list,ci=F)[1]), envir = parent.frame());



} # the end of the function "statistics_all"


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

## Use the function to compute statistics for overall population for each cycle

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

statistics_all(survey_design=nhc1112 , years=1112)

statistics_all(survey_design=nhc1314 , years=1314)

statistics_all(survey_design=nhc1516 , years=1516)

statistics_all(survey_design=nhc1718 , years=1718)

statistics_all(survey_design=nhc1720 , years=1720)





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

## Export the data to Excel ##

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

write_xlsx(N_data_11_12, paste0(path_results,"\\N_data_11_12.xlsx"))

write_xlsx(N_data_13_14, paste0(path_results,"\\N_data_13_14.xlsx"))

write_xlsx(N_data_15_16, paste0(path_results,"\\N_data_15_16.xlsx"))

write_xlsx(N_data_17_18, paste0(path_results,"\\N_data_17_18.xlsx"))

write_xlsx(N_data_17_20, paste0(path_results,"\\N_data_17_20.xlsx"))
```

```
################################################
#### Create an Excel file for result tables  ####
################################################
readme="Readme";
write.xlsx(readme, file=paste0(path_results,"\\N_result_tables.xlsx"),
        sheetName = "Readme", colNames = TRUE, rowNames = TRUE, append = FALSE)




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

### list of the cycles ###
cycle_list =c("2011-2012", "2013-2014","2015-2016", "2017-2018", "2017-2020");

###############################
######  Table Settings ########
###############################
hs1 <- createStyle(
  fgFill = "#DCE6F1", halign = "CENTER", textDecoration = "italic",
  border = "Bottom"
)
hs2 <- createStyle(
  fontColour = "#ffffff", fgFill = "#4F80BD",
  halign = "center", valign = "center", textDecoration = "bold",
  border = "TopBottomLeftRight"
```

```
)

### Load the worksheet in the Excel file  ###

wb <- loadWorkbook( paste0(path_results,"\\N_result_tables.xlsx") );

removeWorksheet(wb, "Overall population")

addWorksheet(wb,"Overall population")

## Table Caption

writeData(wb, sheet = "Overall population", x = "Table XX. Estimated daily intakes (ug/kg-day) for the overall US population, by NHANES cycle", startCol = 1, startRow = 1)

# TCEP//BCEP

writeData(wb, sheet = "Overall population", x = "TCEP // BCEP",borders = "columns", headerStyle = hs2, startCol = 1, startRow = 3)

tbl1 <- BasicTable$new()

### loops for filling in the rest of year data

my_complete_table1=NULL;

for (loop_i in cycle_list) {

my_complete_table1=rbind(

my_complete_table1,cbind(
get(paste0("DI_SSBCEP_mean_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],

        get(paste0("DI_SSBCEP_sd_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],

        get(paste0("DI_SSBCEP_gmean_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],

        get(paste0("DI_SSBCEP_gsd_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],
```

```
        get(paste0("DI_SSBCEP_quantile_",substring(loop_i,3,4),substring(loop_i,8,9))) )
)
}


tbl1$addData(my_complete_table1, firstColumnAsRowHeaders=TRUE,
    explicitColumnHeaders=c("Arithmetic mean","Arithmetic standard deviation","Geometric
mean","Geometric standard deviation",q_list),
    explicitRowHeaders=cycle_list);


# set the styling on the count cells
# the arguments are (rFrom, cFrom, rTo, cTo, declarations)
tbl1$setStyling(2, 2, 6, 10, declarations=list("xl-value-format"="##0.000"))
tbl1$writeToExcelWorksheet(wb=wb, wsName="Overall population",
        topRowNumber=4, leftMostColumnNumber=1,
        applyStyles=T, mapStylesFromCSS=F, outputValuesAs="rawValue")




# TDCIPP//BDCIPP


writeData(wb, sheet = "Overall population", x = "TDCIPP // BDCIPP",borders = "columns", headerStyle =
hs1 , startCol = 1, startRow = 11)
tbl2 <- BasicTable$new()


### loops for filling in the rest of year data
my_complete_table2=NULL;
for (loop_i in cycle_list) {
my_complete_table2=rbind(
my_complete_table2,cbind(
get(paste0("DI_SSBDCPP_mean_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],
```

```
        get(paste0("DI_SSBDCPP_sd_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],
        get(paste0("DI_SSBDCPP_gmean_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],
        get(paste0("DI_SSBDCPP_gsd_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],
        get(paste0("DI_SSBDCPP_quantile_",substring(loop_i,3,4),substring(loop_i,8,9))) )
)
}

tbl2$addData(my_complete_table2, firstColumnAsRowHeaders=TRUE,
    explicitColumnHeaders=c("Arithmetic mean","Arithmetic standard deviation","Geometric
mean","Geometric standard deviation",q_list),
    explicitRowHeaders=cycle_list);

tbl2$setStyling(2, 2, 6, 10, declarations=list("xl-value-format"="##0.000"))
tbl2$writeToExcelWorksheet(wb=wb, wsName="Overall population",
        topRowNumber=12, leftMostColumnNumber=1,
        applyStyles=T, mapStylesFromCSS=F, outputValuesAs="rawValue")


# TCIPP//BCIPP

writeData(wb, sheet = "Overall population", x = "TCIPP // BCIPP",borders = "columns", startCol = 1,
startRow = 19)
tbl3 <- BasicTable$new()

### loops for filling in the rest of year data
my_complete_table3=NULL;
for (loop_i in cycle_list) {
my_complete_table3=rbind(
my_complete_table3,cbind(
get(paste0("DI_SSBCPP_mean_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],
```

```
                get(paste0("DI_SSBCPP_sd_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],
                get(paste0("DI_SSBCPP_gmean_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],
                get(paste0("DI_SSBCPP_gsd_",substring(loop_i,3,4),substring(loop_i,8,9)))[1],
                get(paste0("DI_SSBCPP_quantile_",substring(loop_i,3,4),substring(loop_i,8,9))) )
)
}


tbl3$addData(my_complete_table3, firstColumnAsRowHeaders=TRUE,
        explicitColumnHeaders=c("Arithmetic mean","Arithmetic standard deviation","Geometric
mean","Geometric standard deviation",q_list),
        explicitRowHeaders=cycle_list);
tbl3$setStyling(2, 2, 6, 10, declarations=list("xl-value-format"="##0.000"))
tbl3$writeToExcelWorksheet(wb=wb, wsName="Overall population",
                topRowNumber=20, leftMostColumnNumber=1,
                applyStyles=T, mapStylesFromCSS=F, outputValuesAs="rawValue")



#### Save the statistics for the overall population  ####
saveWorkbook(wb, file=paste0(path_results,"\\N_result_tables.xlsx"), overwrite = TRUE)









##################################################
####  Define a function of "statistics_by"   ####
##################################################
```

```
statistics_by = function (df=df, DI_PHOP=DI_PHOP, years=years, by_var=by_var){

df1=df
DI_PHOP=DI_PHOP
years=years
by_var=by_var

DI_PHOP_na=df1[,DI_PHOP]
by_what_na=df1[, by_var]

df2 = df1[which(!is.na(DI_PHOP_na)==1&!is.na(by_what_na)==1),]

## some groups only have one observation => remove
if ( length(names(which(table(df2[,by_var])==1)))>0 ){
df2 = df2[which(df2[[by_var]]!= names(which(table(df2[[by_var]])==1))),];
df2[[by_var]] = droplevels(factor(df2[[by_var]]))
}

## some groups have no observation => remove
if ( length(levels(factor(df2[[by_var]])))!=length(levels(factor(df1[[by_var]]))) ){
df2[[by_var]]= droplevels(factor(df2[[by_var]]) )
}

by_what=df2[, by_var]

df=df2

suveydesign = svydesign(id=~SDMVPSU, weights=~ WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data=df)
```

```
## Arithmetic mean and SD

DI_mean=svyby(as.formula(paste0("~",DI_PHOP)), by=as.formula(paste0("~",by_var)), design =
suveydesign , svymean, na = TRUE);


options(survey.adjust.domain.lonely=TRUE)

options(survey.lonely.psu="adjust")


datalist <- list()

for(i in levels(as.factor(by_what[!is.na(by_what)])) ) {

  df_subset <- df[which(by_what == i), ]


  design <- svydesign(id = ~SDMVPSU, weights=~WTMEC2YR, strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data = df_subset)


  dat <- data.frame(as.data.frame(sqrt(svyvar(as.formula(paste0("~",DI_PHOP)), na.rm= TRUE, design =
design)))[,"variance"])


  colnames(dat) <- c("sd.variable")

  datalist[[i]] <- dat

}


DI_sd<- as.data.frame(do.call(rbind, datalist)) %>%

  rownames_to_column() %>%

  rename(subgroups = rowname)




## Geometric mean and sd

gmean_out=svyby(as.formula(paste0("~log(",DI_PHOP,")")), by=as.formula(paste0("~",by_var)) ,design =
suveydesign , svymean , na = TRUE) ;
```

```
DI_gmean=data.frame(gmean_out[[1]],exp(gmean_out[[2]]) )

datalist <- list()
for(i in levels(as.factor(by_what[!is.na(by_what)])) ) {
  df_subset <- df[which(by_what == i), ]

  design <- svydesign(id = ~SDMVPSU, weights=~WTMEC2YR,strata=~SDMVSTRA, nest=TRUE,
survey.lonely.psu = "adjust", data = df_subset)

  dat <- data.frame(as.data.frame(exp(sqrt(svyvar(as.formula(paste0("~log(",DI_PHOP,")")), na.rm= TRUE,
design = design)))[[1]]))

  colnames(dat) <- c("sd.variable")
  datalist[[i]] <- dat
}

DI_gsd<- as.data.frame(do.call(rbind, datalist)) %>%
  rownames_to_column() %>%
  rename(subgroups = rowname)

## Quantiles
quantile_list=c(.25,.5,.75,.9,.95);

DI_quantile=data.frame(svyby(as.formula(paste("~",DI_PHOP)), as.formula(paste0("~",by_var)) , design =
suveydesign , na = TRUE, svyquantile, quantiles=quantile_list)[,1:length(quantile_list)+1])

output_name=paste(DI_PHOP,"_",by_var,years, sep = "");
assign(output_name, cbind(DI_mean[,1:2], DI_sd[,2], DI_gmean[,2], DI_gsd[,2], DI_quantile), envir =
parent.frame());
}
```

```
#############################################
### End the of definition of the function ###
#############################################



###########################################################
##### Statistics for sub-groups for each year #############
###########################################################

for (loop_i in cycle_list){# the start of the loop to generate stratified tables by variables for each year
yrs=NULL;
dfs=NULL;

yrs=paste0(substring(loop_i,3,4),substring(loop_i,8,9))
dfs=get(paste0("N_data_",substring(loop_i,3,4),"_",substring(loop_i,8,9)))

#######################
#### Gender      ####
#######################
statistics_by(df=dfs, DI_PHOP="DI_SSBCEP",  years=yrs, by_var="RIAGENDR")
statistics_by(df=dfs, DI_PHOP="DI_SSBDCPP", years=yrs, by_var="RIAGENDR")
statistics_by(df=dfs, DI_PHOP="DI_SSBCPP",  years=yrs, by_var="RIAGENDR")

#df=N_data_13_14 DI_PHOP="DI_SSBCEP" suveydesign = nhc1314 years="1314" by_var="RIAGENDR"; #
for code testing and checking

## the results by gender
#DI_SSBCEP_RIAGENDR1314
```

#DI_SSBDCPP_RIAGENDR1314

#DI_SSBCPP_RIAGENDR1314

#1 = Male;

#2 = Femal;

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

Pregnancy status

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

statistics_by(df=dfs, DI_PHOP="DI_SSBCEP", years=yrs, by_var="RIDEXPRG")

statistics_by(df=dfs, DI_PHOP="DI_SSBDCPP", years=yrs, by_var="RIDEXPRG")

statistics_by(df=dfs, DI_PHOP="DI_SSBCPP", years=yrs, by_var="RIDEXPRG")

#df=N_data_13_14 DI_PHOP="DI_SSBCEP" suveydesign = nhc1314 years="1314" by_var="RIDEXPRG"; # for code testing and checking

the results by pregnancy status

#DI_SSBCEP_RIDEXPRG1314

#DI_SSBDCPP_RIDEXPRG1314

#DI_SSBCPP_RIDEXPRG1314

#1 = yes, positive lab test or self-reported pregnant;

#2 = not pregnant at exam;

#3 = cannot ascertain if pregnant at exam

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

```
#### Age groups     ####
##########################
statistics_by(df=dfs, DI_PHOP="DI_SSBCEP",  years=yrs, by_var="age_grp")
statistics_by(df=dfs, DI_PHOP="DI_SSBDCPP", years=yrs, by_var="age_grp")
statistics_by(df=dfs, DI_PHOP="DI_SSBCPP",  years=yrs, by_var="age_grp")

## the results by age groups
#DI_SSBCEP_age_grp1314
#DI_SSBDCPP_age_grp1314
#DI_SSBCPP_age_grp1314

#1 3-5
#2 6-11
#3 12-17
#4 18+


##########################
#### Race groups     ####
##########################
statistics_by(df=dfs, DI_PHOP="DI_SSBCEP",  years=yrs, by_var="RIDRETH3")
statistics_by(df=dfs, DI_PHOP="DI_SSBDCPP", years=yrs, by_var="RIDRETH3")
statistics_by(df=dfs, DI_PHOP="DI_SSBCPP",  years=yrs, by_var="RIDRETH3")

## the results by race groups
#DI_SSBCEP_RIDRETH31314
#DI_SSBDCPP_RIDRETH31314
#DI_SSBCPP_RIDRETH31314
```

```
#1	Mexican American
#2	Other Hispanic
#3	Non-Hispanic White
#4	Non-Hispanic Black
#6	Non-Hispanic Asian
#7	Other Race - Including Multi-Racial


#################################
#### Income to poverty Ratio ####
#################################
statistics_by(df=dfs, DI_PHOP="DI_SSBCEP",  years=yrs, by_var="poverty_ratio")
statistics_by(df=dfs, DI_PHOP="DI_SSBDCPP", years=yrs, by_var="poverty_ratio")
statistics_by(df=dfs, DI_PHOP="DI_SSBCPP",  years=yrs, by_var="poverty_ratio")

## the results by poverty ration
#DI_SSBCEP_poverty_ratio1314
#DI_SSBDCPP_poverty_ratio1314
#DI_SSBCPP_poverty_ratio1720

#1	<1
#2	1 to 3
#3	>3


###############################
#### Adult education level ####
###############################
if (yrs !="1720"){ # no data in 2017-2020
```

```
statistics_by(df=dfs, DI_PHOP="DI_SSBCEP",  years=yrs, by_var="adult_edu")

statistics_by(df=dfs, DI_PHOP="DI_SSBDCPP", years=yrs, by_var="adult_edu")

statistics_by(df=dfs, DI_PHOP="DI_SSBCPP",  years=yrs, by_var="adult_edu")

}

## the results by education level

#DI_SSBCEP_adult_edu1314

#DI_SSBDCPP_adult_edu1314

#DI_SSBCPP_adult_edu1314


#1	Less than high school degree

#2	High school grad/GED or some college/AA degree

#3	College graduate or above



#yrs=NULL;

#dfs=NULL;


} # the end of the loop

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




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

#### Define a function to write the result tables into the excel  #####

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


save_worksheet_excel = function(path,exceldata,sheet_name,table_caption,
by_var,group_list,recode_list) {
```

```
##########################################################

#table_caption = "Table XX. Estimated daily intakes (ug/kg-day), by age group and NHANES cycle"

#sheet_name = "Age groups"

#by_var="age_grp";

#group_list = c("3-5 yrs","6-11 yrs","12-17 yrs","18+ yrs")

#group_list = c("6-11 yrs","12-17 yrs","18+ yrs")

#recode_list=expression({result_data1[,1]=recode(result_data1[,1], '1'="3-5 yrs", '2'="6-11 yrs", '3'="12-
17 yrs",'4'="18+ yrs");})

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


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

wb <- loadWorkbook(paste(path,"\\",exceldata,sep=""));


addWorksheet(wb,sheet_name)


## Table Caption

writeData(wb, sheet = sheet_name, x = table_caption,borders = "columns", startCol = 1, startRow = 1)


# 3 Chemicals


position=3

chemical_list=c( "TCEP // BCEP", "TDCIPP // BDCIPP", "TCIPP // BCIPP");


for (chemical_name in chemical_list){


        if (chemical_name=="TCEP // BCEP"){DI_PHOP="SSBCEP"}

        if (chemical_name=="TDCIPP // BDCIPP"){DI_PHOP="SSBDCPP"}

        if (chemical_name=="TCIPP // BCIPP"){DI_PHOP="SSBCPP"}
```

```
for (grp_name in group_list){

            writeData(wb, sheet = sheet_name, x = chemical_name, borders = "columns", startCol = 1,
startRow = position )
            writeData(wb, sheet = sheet_name, x = grp_name, borders = "columns", startCol = 1, startRow =
(position+1) )


  ### loops for filling in the rest of year data
            my_complete_table=NULL;

for (loop_i in cycle_list) {

dataset_name=paste0("DI_",DI_PHOP,"_",by_var, substring(loop_i,3,4),substring(loop_i,8,9));

if (exists(dataset_name)){
result_data1 = get(dataset_name);
#########################################################################
eval(recode_list)
#########################################################################
result_data = result_data1;

if (nrow(result_data[which(result_data[,1]==grp_name),])>0){

my_complete_table=rbind(
my_complete_table,matrix(c(result_data[which(result_data[,1]==grp_name),2],
                    result_data[which(result_data[,1]==grp_name),3],
                    result_data[which(result_data[,1]==grp_name),4],
```

```
                    result_data[which(result_data[,1]==grp_name),5],
                    result_data[which(result_data[,1]==grp_name),6:10]),1,9)
)
} else {

cat("No Data:",loop_i,by_var, grp_name,"\n");
my_complete_table=rbind(my_complete_table, rep("NoData",9))
}

} else {

cat("No Data:",loop_i,by_var,"\n");
my_complete_table=rbind(my_complete_table, rep("NoData",9))

} # exclude the cycle of 2017-2020 if no data
} # end of the cycles

my_complete_table=as.data.frame(my_complete_table);

tblg1 <- BasicTable$new()
tblg1$addData(my_complete_table, firstColumnAsRowHeaders=TRUE,
        explicitColumnHeaders=c("Arithmetic mean","Arithmetic standard deviation","Geometric
mean","Geometric standard deviation",q_list),
        explicitRowHeaders=cycle_list);
tblg1$setStyling(2, 2, 6, 10, declarations=list("xl-value-format"="##0.000"))
tblg1$writeToExcelWorksheet(wb=wb, wsName=sheet_name,
                    topRowNumber=(position+2), leftMostColumnNumber=1,
                    applyStyles=T, mapStylesFromCSS=F, outputValuesAs="rawValue")
```

```
position=position+10

saveWorkbook(wb, file=paste(path,"\\",exceldata,sep=""), overwrite = TRUE);

} # end of the subgroup names

} # end of the chemical group


} # end of the function





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

##### Settings for saving the result tables ###################

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


#### The location of the physical Excel file  ####

path = path_results;


#### The name of the Excel file of all tables ###

exceldata = "N_result_tables.xlsx";




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

###### Save All the tables in the same Excel file  ######

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


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

#### Gender ####

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

table_caption = "Table XX. Estimated daily intakes (ug/kg-day), by gender and NHANES cycle"

sheet_name = "Gender"
```

```
by_var="RIAGENDR";
group_list = c("Female","Male")

#### Need to be updated based on the group variable used here  ######
recode_list=expression({result_data1[,1]=recode(result_data1[,1], '1'="Male", '2'="Female");})

save_worksheet_excel(path,exceldata,sheet_name,table_caption,by_var,group_list,recode_list)

#1 = Male;
#2 = Female;

###################
#### Pregnancy ####
###################
table_caption = "Table XX. Estimated daily intakes (ug/kg-day), by pregnancy status and NHANES cycle"
sheet_name = "Pregnancy status"
by_var="RIDEXPRG";
group_list = c("Pregnant","Not pregnant")

#### Need to be updated based on the group-variable used here  ######
recode_list=expression({result_data1[,1]=recode(result_data1[,1], '1'="Pregnant", '2'="Not pregnant");})

save_worksheet_excel(path,exceldata,sheet_name,table_caption,by_var,group_list,recode_list)

#1 = yes, positive lab test or self-reported pregnant;
#2 = not pregnant at exam;


####################
```

```
#### Age Groups ####

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

table_caption = "Table XX. Estimated daily intakes (ug/kg-day), by age group and NHANES cycle"

sheet_name = "Age groups"

by_var="age_grp";

group_list = c("3-5 yrs","6-11 yrs","12-17 yrs","18+ yrs")

#group_list = c("6-11 yrs","12-17 yrs","18+ yrs")


#### Need to be updated based on the group variable used here  ######


recode_list=expression({result_data1[,1]=recode(result_data1[,1], '1'="3-5 yrs", '2'="6-11 yrs", '3'="12-17 yrs",'4'="18+ yrs");})


save_worksheet_excel(path,exceldata,sheet_name,table_caption,by_var,group_list,recode_list)


#1 3-5

#2 6-11

#3 12-17

#4 18+




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

#### Race-ethnicity Groups ####

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

table_caption = "Table XX. Estimated daily intakes (ug/kg-day), by race/ethnicity and NHANES cycle"

sheet_name = "Race-ethnicity groups"

by_var="RIDRETH3";
```

group_list = c("Mexican American","Other Hispanic","Non-Hispanic White","Non-Hispanic Black", "Non-Hispanic Asian","Other Race - Including Multi-Racial")

Need to be updated based on the group variable used here

recode_list=expression({result_data1[,1]=recode(result_data1[,1], '1'="Mexican American", '2'="Other Hispanic", '3'="Non-Hispanic White",'4'="Non-Hispanic Black", '6'="Non-Hispanic Asian",'7'="Other Race - Including Multi-Racial");})

save_worksheet_excel(path,exceldata,sheet_name,table_caption,by_var,group_list,recode_list)

#1 Mexican American

#2 Other Hispanic

#3 Non-Hispanic White

#4 Non-Hispanic Black

#6 Non-Hispanic Asian

#7 Other Race - Including Multi-Racial

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

Adult Education Levels

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

table_caption = "Table XX. Estimated daily intakes (ug/kg-day), by highest education level for an adult in the household and NHANES cycle"

sheet_name = "Adult education level"

by_var="adult_edu";

group_list = c("Less than high school degree","High school grad/GED or some college/AA degree","College graduate or above")

Need to be updated based on the group variable used here

```
recode_list=expression({result_data1[,1]=recode(result_data1[,1], '1'="Less than high school degree", '2'="High school grad/GED or some college/AA degree", '3'="College graduate or above");})

save_worksheet_excel(path,exceldata,sheet_name,table_caption,by_var,group_list,recode_list)

#1	Less than high school degree
#2	High school grad/GED or some college/AA degree
#3	College graduate or above



##################################
#### Income to poverty Ratio ####
##################################
table_caption = "Table XX. Estimated daily intakes (ug/kg-day), by ratio of income to poverty and NHANES cycle"
sheet_name = "Income to poverty Ratio"
by_var="poverty_ratio";
group_list = c("<1","1 to 3",">3")

#### Need to be updated based on the group variable used here  ######
recode_list=expression({result_data1[,1]=recode(result_data1[,1], '1'="<1", '2'="1 to 3", '3'=">3");})

save_worksheet_excel(path,exceldata,sheet_name,table_caption,by_var,group_list,recode_list)

#1	<1
#2	1 to 3
#3	>3
```

```
######################################
####   High-exposed Individuals   ####
######################################
recent_cycle = "2017-2018";


getput_name_BDCPP
=paste0("DI_SSBDCPP_gmean_",substring(recent_cycle,3,4),substring(recent_cycle,8,9));
GM_BDCPP = get(getput_name_BDCPP)[1]


getput_name_BCPP
=paste0("DI_SSBCPP_gmean_",substring(recent_cycle,3,4),substring(recent_cycle,8,9));
GM_BCPP = get(getput_name_BCPP)[1]


getput_name_BCEP
=paste0("DI_SSBCEP_gmean_",substring(recent_cycle,3,4),substring(recent_cycle,8,9));
GM_BCEP = get(getput_name_BCEP)[1]


working_data =
get(paste0("N_data_",substring(recent_cycle,3,4),"_",substring(recent_cycle,8,9)))[,c("SEQN","RIDAGEYR
","DI_SSBDCPP",
"DI_SSBCPP", "DI_SSBCEP")]


# the total number of individuals that were evaluated for the high-exposure analysis.
dim(working_data)
# View(working_data)
# complete cases for DI_SSBDCPP
dim(working_data[complete.cases(working_data[,c("DI_SSBDCPP")]),])
# complete cases for DI_SSBCPP
dim(working_data[complete.cases(working_data[,c("DI_SSBCPP")]),])
# complete cases for DI_SSBCEP
```

```
dim(working_data[complete.cases(working_data[,c("DI_SSBCEP")]),])

# complete cases for "DI_SSBDCPP","DI_SSBCPP"

dim(working_data[complete.cases(working_data[,c("DI_SSBDCPP","DI_SSBCPP")]),])

# complete cases for "DI_SSBCPP","DI_SSBCEP"

dim(working_data[complete.cases(working_data[,c("DI_SSBCPP","DI_SSBCEP")]),])

# complete cases for "DI_SSBDCPP","DI_SSBCEP"

dim(working_data[complete.cases(working_data[,c("DI_SSBDCPP","DI_SSBCEP")]),])

# complete cases for ALL

dim(working_data[complete.cases(working_data[,c("DI_SSBDCPP","DI_SSBCPP","DI_SSBCEP")]),])


##

working_data = working_data %>%

        mutate(hi_exp_1 = 1*(DI_SSBCPP > GM_BCPP)  ) %>%

        mutate(hi_exp_2 = 1*(DI_SSBCEP > GM_BCEP)  ) %>%

        mutate(hi_exp_3 = 1*(DI_SSBDCPP> GM_BDCPP) )%>%

        mutate(hi_exp_4 = 1*(DI_SSBCPP > GM_BCPP)*(DI_SSBCEP > GM_BCEP) )%>%

        mutate(hi_exp_5 = 1*(DI_SSBCPP > GM_BCPP)*(DI_SSBDCPP> GM_BDCPP))%>%

        mutate(hi_exp_6 = 1*(DI_SSBCEP > GM_BCEP)*(DI_SSBDCPP> GM_BDCPP))%>%

        mutate(hi_exp_7 = 1*(DI_SSBCPP > GM_BCPP)*(DI_SSBCEP > GM_BCEP)*(DI_SSBDCPP>
GM_BDCPP))


hiexp_groups = c("TCPP > GM",

      "TCEP > GM",

      "TDCPP > GM",

      "TCPP & TCEP > GM",

      "TCPP & TDCPP > GM",

      "TCEP & TDCPP > GM",

      "TCPP & TCEP & TDCPP > GM");
```

```
data_hi_exp_1= working_data %>% filter(hi_exp_1==1)

data_hi_exp_2= working_data %>% filter(hi_exp_2==1)

data_hi_exp_3= working_data %>% filter(hi_exp_3==1)

data_hi_exp_4= working_data %>% filter(hi_exp_4==1)

data_hi_exp_5= working_data %>% filter(hi_exp_5==1)

data_hi_exp_6= working_data %>% filter(hi_exp_6==1)

data_hi_exp_7= working_data %>% filter(hi_exp_7==1)


#table(working_data$hi_exp_1)

#table(working_data$hi_exp_2)

#table(working_data$hi_exp_3)

#table(working_data$hi_exp_4)

#table(working_data$hi_exp_5)

#table(working_data$hi_exp_6)

#table(working_data$hi_exp_7)


GM_value = c("GM",round(GM_BCEP,3), round(GM_BCPP,3), round(GM_BDCPP,3), "AGE");


sheet_name = "High-Exposure"


table_caption = "Table 2: 2017-2018 Individual daily intake (ug/kg/day) derived from step 2C"


hiexp_group="HiExp1"


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

wb <- loadWorkbook(paste(path,"\\",exceldata,sep=""));


#removeWorksheet(wb,sheet_name)

addWorksheet(wb,sheet_name)
```

```
## Table Caption

writeData(wb, sheet = sheet_name, x = table_caption,borders = "columns", startCol = 1, startRow = 1)


chemical_list=c( "TCEP // BCEP", "TDCIPP // BDCIPP", "TCIPP // BCIPP");

hiexp_group_list = c("HiExp1","HiExp2","HiExp3","HiExp4","HiExp5","HiExp6","HiExp7")


position=1

jj=1

for (hiexp_group in hiexp_group_list){


dataset_name=paste0("data_hi_exp_",substring(hiexp_group,6,6));

result_data1 = get(dataset_name);

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

result_data= result_data1 %>% select(SEQN, DI_SSBCEP, DI_SSBCPP, DI_SSBDCPP, RIDAGEYR)


my_complete_table=result_data


writeData(wb, sheet = sheet_name, x =hiexp_groups[[jj]],borders = "columns", headerStyle = hs1,
startCol = position, startRow = 3)

jj=jj+1;

writeData(wb, sheet = sheet_name, x = GM_value[1] ,borders = "columns", borderColour ="red",
startCol = position, startRow = 4)

writeData(wb, sheet = sheet_name, x = GM_value[2] ,borders = "columns", borderColour ="red",
startCol = position+1, startRow = 4)

writeData(wb, sheet = sheet_name, x = GM_value[3] ,borders = "columns", borderColour ="red",
startCol = position+2, startRow = 4)

writeData(wb, sheet = sheet_name, x = GM_value[4] ,borders = "columns", borderColour ="red",
startCol = position+3, startRow = 4)

writeData(wb, sheet = sheet_name, x = GM_value[5] ,borders = "columns", borderColour ="red",
startCol = position+4, startRow = 4)
```

```
writeData(wb, sheet = sheet_name, x = "SEQN" ,borders = "columns", startCol = position, startRow = 5)


tblg1 <- BasicTable$new()

tblg1$addData(my_complete_table[,2:5], firstColumnAsRowHeaders=TRUE,

      explicitColumnHeaders=c("TCEP","TCPP","TDCPP","RIDAGEYR"),

      explicitRowHeaders=as.character(my_complete_table[[1]]));

tblg1$setStyling(2, 2, length(my_complete_table[[1]])+ 5 , 4, declarations=list("xl-value-
format"="##0.000"))

tblg1$writeToExcelWorksheet(wb=wb, wsName=sheet_name,

            topRowNumber=5, leftMostColumnNumber=(position),

            applyStyles=T, mapStylesFromCSS=F, outputValuesAs="rawValue")


position=position+6


saveWorkbook(wb, file=paste(path,"\\",exceldata,sep=""), overwrite = TRUE);

} # end of the subgroup names





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

#########   THE END    #############

####################################
```