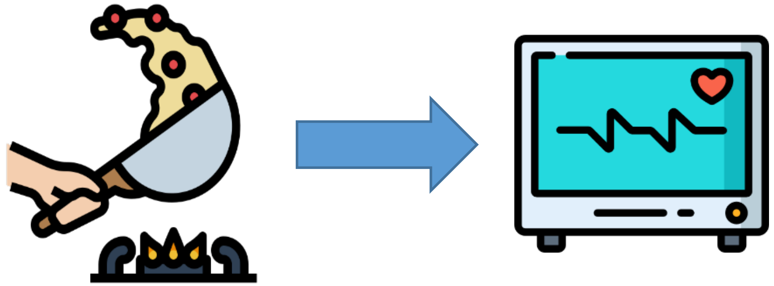
Data analysis on quantification of $PM_{2.5}$ exposure-health evaluation

Presenter: Dr. Ming-Chien Mark Tsou

Research Center for Environmental Changes

Academia Sinica



Outline

Part 1:
PM data processing (matching the time of heart rate variability monitoring)

Part 2:
Questionnaire/time-activity diary (TAD) data processing

Part 3:
Generalized Additive Mixed Model (GAMM)

Objectives

- To evaluate the effect of $PM_{2.5}$ exposure on heart rate variability (HRV) indices by Generalized Additive Mixed Model (GAMM)
  - To control fixed and random effects, including linear and non-linear parameter
  - To control autocorrelation

Part 1:
PM data processing

(matching the time of heart rate variability monitoring)

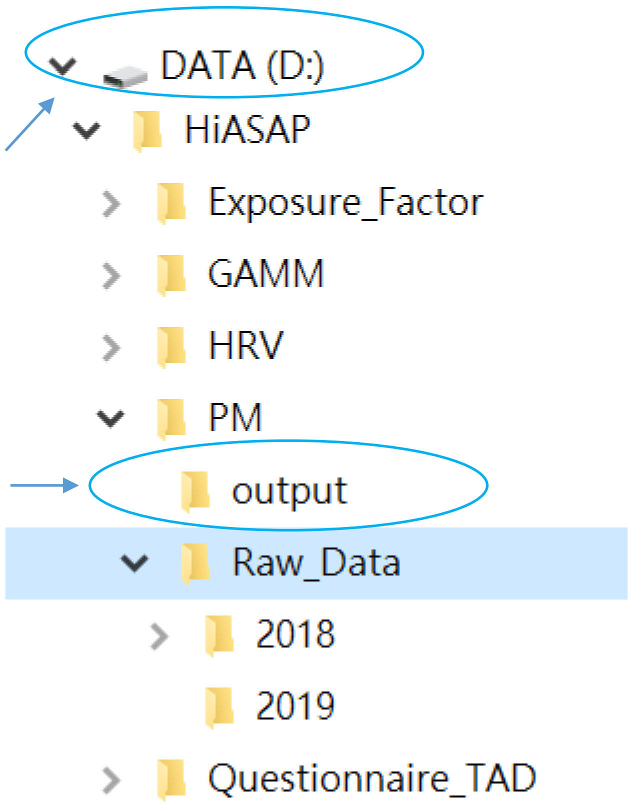
You can modify by yourself

```r
# To remove previous memory in R.
rm(list=ls())

location <- "D:/"

# To set the output file
cmd1 <- paste0("setwd('",location,"HiASAP/PM/output')")
eval(parse(text=cmd1))
```

1. To remove all objects from current workspace

2. To set the default drive of your data

3. To set the path of output file



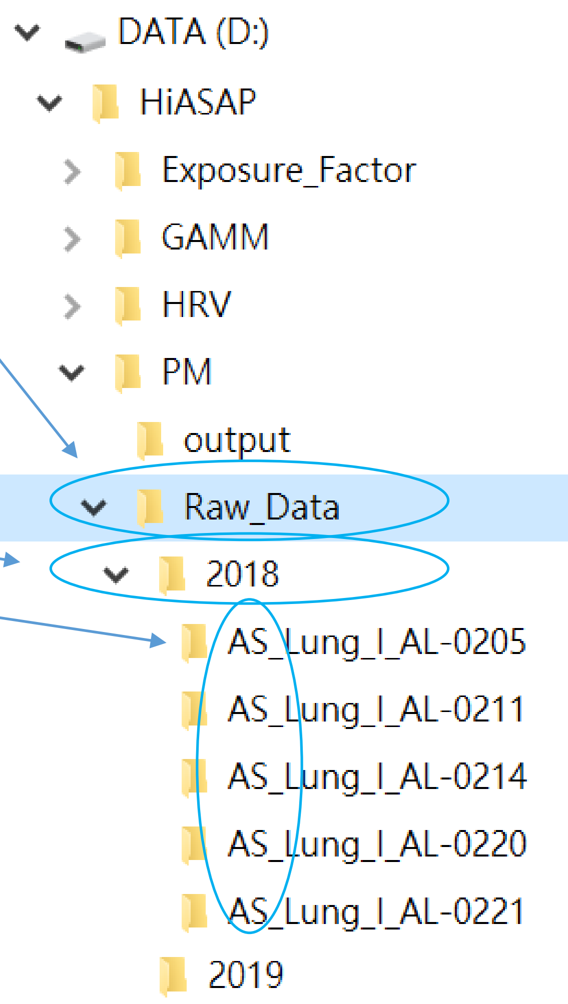
You can modify by yourself

To import the PM data by subjects/AS-LUNG

```
# To read the PM (AS-LUNG) data
way <- paste0(location,"HiASAP/PM/Raw_Data")
aa <- dir(path=way)
for(i in 1:length(aa)){
    way2 <- paste0(way,"/",aa[i])
    aa1 <- dir(path=way2,pattern="AS")
```

4. To set the path of PM data

To set the pattern (keyword) to select files



To combine the PM data by subjects/AS-LUNG

```r
for(p in 1:length(aa1)){
    ASLUNG <- data.frame()
    fileloc <- paste0(way2,"/",aa1[p])
    bb <- list.files(fileloc,pattern='csv')
    filename <- paste0(fileloc,"/",bb[1])
    cc <- read.csv(filename)
    ASLUNG <- cc
    for(k in 2:length(bb)){
        filename <- paste0(fileloc,"/",bb[k])
        cc <- read.csv(filename)
        ASLUNG <- rbind(ASLUNG,cc)
    }
```

5. To combine PM data for each subject/AS-LUNG

Dataset after QA/QC for PM (AS-LUNG) data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	datatime	id	date	time	sht_t	sht_h	pm1	pm25	pm10	co2	adc	acc_x	acc_y	acc_z	accx_int	accy
2	########	0C9A424(	########	00:00:00	26.6	67.1	13	17	18	552	1	-1	-66	-2	60	
3	########	0C9A424(	########	00:00:15	26.6	67.1	13	17	18	555	1	-1	-66	0	7	
4	########	0C9A424(	########	00:00:30	26.6	67.1	13	16	17	558	1	-2	-66	-2	81	
5	########	0C9A424(	########	00:00:45	26.6	67.1	13	16	17	558	1	-1	-66	-2	75	
6	########	0C9A424(	########	00:01:00	26.6	67.1	13	16	17	558	1	-2	-66	-2	67	
7	########	0C9A424(	########	00:01:15	26.6	67.1	13	15	16	558	1	-1	-68	-2	60	
8	########	0C9A424(	########	00:01:30	26.6	67.1	13	16	16	558	1	-1	-66	-2	80	

```
# To select the variables of time, temperature, relative humidity, CO2, corrected PM1 and corrected PM2.5
ASLUNGt <- data.frame(subset(ASLUNG,select=c(datatime,date,time,sht_t,sht_h,co2,cPM1,cPM2.5)))
```

6. To select interested variables from original dataset
(time, temperature, relative humidity, corrected PM_1 and corrected $PM_{2.5}$)

```
# To exclude the time without PM2.5 data
ASLUNGt<-ASLUNGt %>%
    filter(!(is.na(cPM2.5)))
```

7. To exclude the time without $PM_{2.5}$ data

To create variables for the following analysis

```r
# To create the "Season" variable (fall=0 and winter=1)
Season <- c()
for(i in 1:dim(ASLUNGt)[1]){
    if(substr(ASLUNGt$date[i],1,4)==2018){
        Season[i]<-0
    }else{
        Season[i]<-1
    }
}
```

8. To create a “Season” variable

In this case, fall (2018) = “0” and spring (2019) = “1”

```r
# To create the variable of type of AS-LUNG (outdoor=1, indoor=2 and personal=3)
AL_Type <- c()
if(substr(aa1[p],9,9)=="O"){
    AL_Type<-1
}else{
    if(substr(aa1[p],9,9)=="I"){
        AL_Type<-2
    }else{
        AL_Type<-3
    }
}
```

9. To create a variable of type of AS-LUNG

In this case, outdoor version = “1”, indoor version = “2”, and portable version = “3”

```r
        ASLUNGt <- data.frame(Season,AL_Type,ASLUNGt)

        outputname<-paste0("PM_",substr(bb[k],10,14),"_",Season[i]+1,"_",substr(aa1[p],9,9),"_Orig.csv")
        write.csv(ASLUNGt,outputname,row.names=FALSE,na="")
      }
}
```

10. To combine the created variables with dataset

11. To export the dataset

	A	B	C	D	E	F	G	H	I	J	K
1	Season	AL_Type	datatime	date	time	sht_t	sht_h	co2	cPM1	cPM2.5	
2	0	2	2018/10/8 00:00	2018/10/8	00:00:00	26.7	71.4	420	14.307	15.182	
3	0	2	2018/10/8 00:00	2018/10/8	00:00:15	26.7	71.4	419	13.708	14.278	
4	0	2	2018/10/8 00:00	2018/10/8	00:00:30	26.7	71.3	417	13.708	14.278	
5	0	2	2018/10/8 00:00	2018/10/8	00:00:45	26.8	71.3	417	13.708	13.826	
6	0	2	2018/10/8 00:01	2018/10/8	00:01:00	26.8	71.2	415	13.708	13.826	
7	0	2	2018/10/8 00:01	2018/10/8	00:01:15	26.8	71.2	414	13.708	13.826	
8	0	2	2018/10/8 00:01	2018/10/8	00:01:30	26.8	71.2	414	13.708	13.826	
9	0	2	2018/10/8 00:01	2018/10/8	00:01:45	26.8	71.2	414	13.109	13.826	
10	0	2	2018/10/8 00:02	2018/10/8	00:02:00	26.8	71.1	414	13.109	13.826	
11	0	2	2018/10/8 00:02	2018/10/8	00:02:15	26.8	71	414	13.109	13.826	

To calculate 5-min PM data based on the time of heart rate variability monitoring

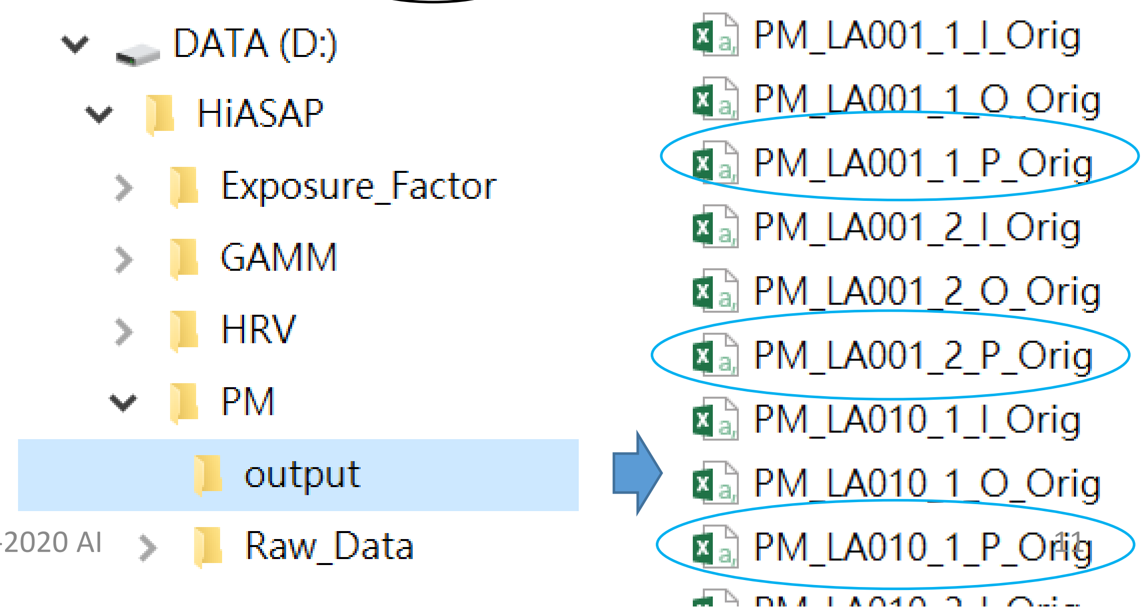
```r
# To calculate 5-min average PM data for Generalized Additive Mixed Model (GAMM) (based on Rooti time)
way_Rooti <- paste0(location,"HiASAP/HRV/Raw_Data/")
```

12. To set the path of HRV data to get the time of HRV monitoring

To set the files of personal data

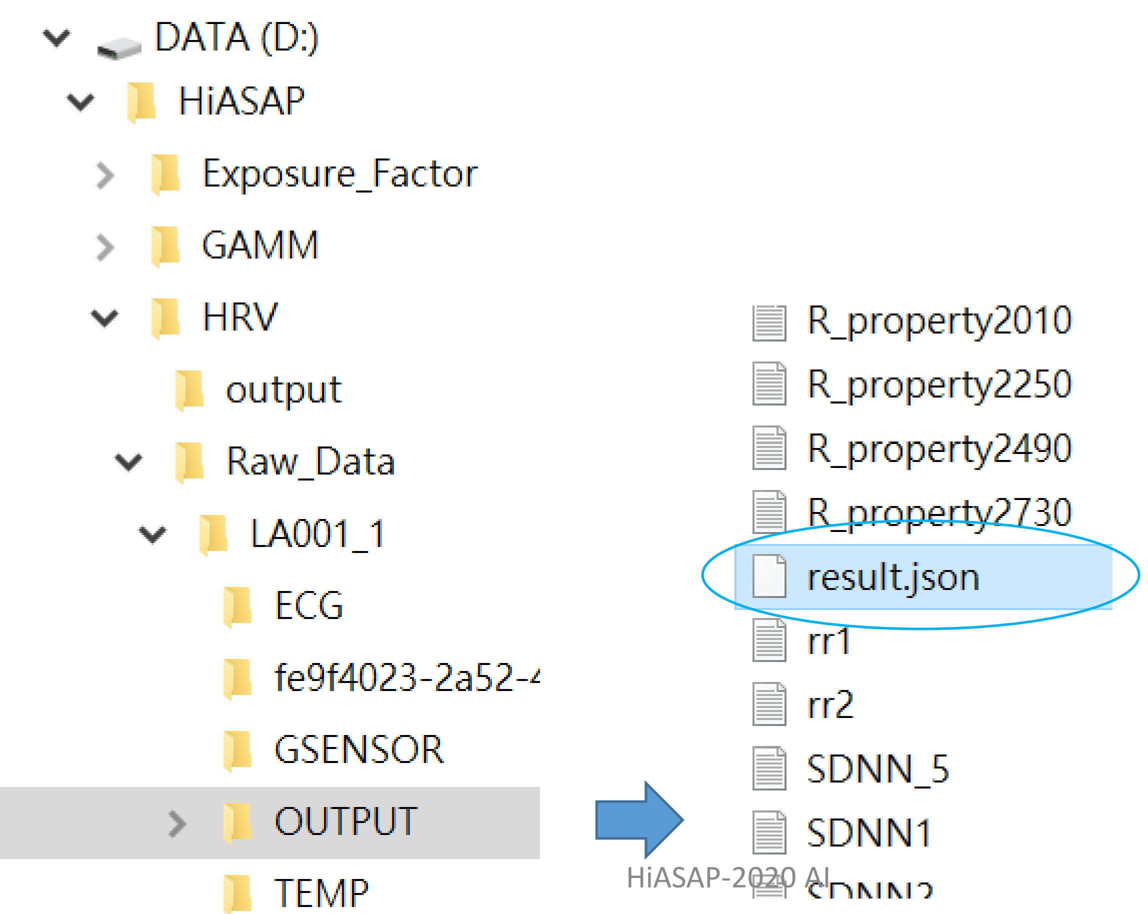
```r
aa2 <- list.files(paste0(location,"HiASAP/PM/output/"),pattern="P_Orig")
```

13. To set the path of PM data which we export in step 11



14. To read the results of HRV monitoring for getting the start time and end time

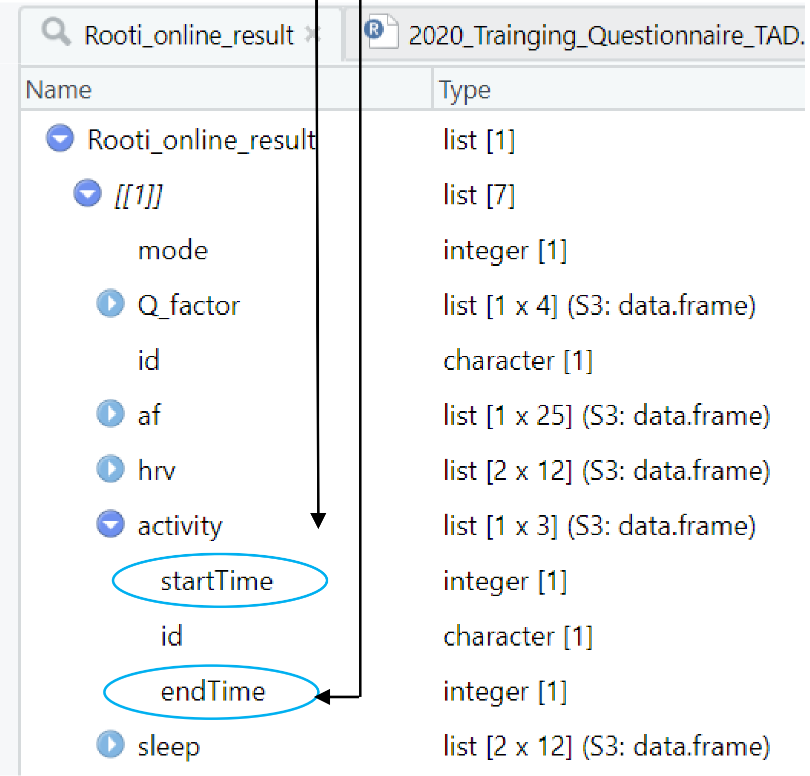
```r
for (p in 1:length(aa2)) {
    # To read the time of heart rate variability monitoring form Rooti
    filename <- paste0(location,"HiASAP/PM/output/",aa2[p])
    ASLUNGt <- read.csv(filename)
    Rooti_online_result<- fromJSON(paste0(way_Rooti, substr(aa2[p],4,10), "/OUTPUT/result.json"))
```



15. To get the start time and end time of HRV monitoring

```r
start_time<-Rooti_online_result$activity$startTime
start_time<-as.POSIXct(start_time, origin="1970-01-01",tz='Asia/Taipei')
end_time<-Rooti_online_result$activity$endTime
end_time<-as.POSIXct(end_time, origin="1970-01-01",tz='Asia/Taipei')
```

The time is present as how many seconds has passed since Jan 1, 1970



16. To format the time variables expressed as YYYY-MN-DD hh:mm:ss

Example:

"2018-10-08 13:11:56 CST"

```
from <- as.POSIXct(substr(start_time,1,16),tz="Asia/Taipei")
to <- as.POSIXct(substr(end_time,1,16),tz="Asia/Taipei")
sort_out_time<-data.frame(date=seq.POSIXt(from, to, by = "15 secs",tz="Asia/Taipei"))
Date_AL<-seq.POSIXt(from, to, by = "15 secs",tz="Asia/Taipei")
Date_AL2<-Date_AL[1:(length(Date_AL)-1)]
```

The time interval is set as 15 secs

17. To get the time of PM data and format the time to be consistent with HRV data

```
date_1<-c(ymd(as.character(ASLUNGt$date)))
time<-substr(ASLUNGt$time,1,8)
date<-paste(date_1,time)
ASLUNGt2<-data.frame(date,ASLUNGt)

ASLUNGt2$date <- as.POSIXct(ASLUNGt2$date,tz="Asia/Taipei")
```

18. To add the start time of HRV monitoring to PM data to have the same 5-min intervals

	PM data	HRV data
Start time	10:00:00	10:02:00
5-min intervals	10:00:00 to 10:04:59 10:05:00 to 10:09:59 … and so on	10:02:00 to 10:06:59 10:07:00 to 10:11:59 … and so on

```r
# If the start time of PM monitoring was different from the start time of HRV monitoring,
# we add the start time of HRV monitoring to PM data to have the consistent 5-min interval
if((substr(ASLUNGt2$date[1],1,16))!=(substr(Date_AL2[1],1,16))){
    dd <- as.POSIXct(Date_AL2[1],tz="Asia/Taipei")
    Add_Row <- data.frame()
    Add_Row <- data.frame(date=as.factor(dd),Season="",AL_Type="",datatime="",date.1="",time="",sht_t="",sht_h="",co2="",cPM1="",cPM2.5="")
    ASLUNGt3<- rbind(ASLUNGt2,Add_Row)
}else{
    ASLUNGt3<- ASLUNGt2
}
ASLUNGt3 <- merge(ASLUNGt3,sort_out_time,by="date")
```

	date	Season	AL_Type	datatime	date.1	time	sht_t	sht_h	co2	cPM1	cPM2.5
1	2018-10-08 13:11:00										
2	2018-10-08 13:13:00	0	3	2018-10-08 13:13:00	2018-10-08	13:13:00	26.3	78.9	445	2.573	2.887
3	2018-10-08 13:13:15	0	3	2018-10-08 13:13:15	2018-10-08	13:13:15	26.3	78.9	445	3.084	3.615
4	2018-10-08 13:13:30	0	3	2018-10-08 13:13:30	2018-10-08	13:13:30	26.3	78.9	445	2.573	3.251
5	2018-10-08 13:13:45	0	3	2018-10-08 13:13:45	2018-10-08	13:13:45	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

19. To format the time variables again after step 18

Notice:
The format of time variables may be changed after you run a command

```
Date_AL<-seq.POSIXt(ASLUNGt3$date[1], ASLUNGt3$date[dim(ASLUNGt3)[1]], by = "15 secs",tz="Asia/Taipei")
Date_AL2<-c(ASLUNGt3$date)
```

20. To calculate the 5-min average PM data

```
# To calculate 5-min average PM data
ALFinal <-ASLUNGt3 %>%
    group_by(date = cut(Date_AL2, breaks="300 secs")) %>%
    summarize(
        TEM = mean(as.numeric(sht_t), na.rm = TRUE),
        HUM = mean(as.numeric(sht_h), na.rm = TRUE),
        PM1 = mean(as.numeric(cPM1), na.rm = TRUE),
        PM2.5 = mean(as.numeric(cPM2.5), na.rm = TRUE),
        CO2 = mean(as.numeric(co2), na.rm = TRUE),
        Freq = length(as.numeric(cPM2.5)))
```

The time interval is set as 5 minutes

To count the number of data in each 5-min interval for excluding the intervals with insufficient data

	date	TEM	HUM	PM1	PM2.5	CO2	Freq
1	2018-10-08 13:11:00	26.30500	78.89500	2.956250	3.378400	449.9500	21
2	2018-10-08 13:16:00	26.71000	78.17000	2.981800	3.487600	448.0500	20
3	2018-10-08 13:21:00	27.01000	77.31500	3.135100	3.633200	440.6500	20
4	2018-10-08 13:26:00	27.16500	76.05000	3.186200	3.633200	428.6500	20

21. To exclude the time with insufficient data and create the variables for the following analysis

```r
ALFinal$date <-ymd_hms(ALFinal$date,tz="Asia/Taipei")
ALFinal2 <- ALFinal[which(ALFinal$Freq>=10),]
colnames(ALFinal2)<-c("Date","TEM","HUM","PM1","PM2.5","CO2","Freq")

S_no<-substr(aa2[p],4,8)

Season <- as.numeric(substr(aa2[p],10,10))-1

AL_Type<-3
```

There should be 4x5=20 data in each 5-min interval for personal PM data

The variable for ID of subjects (S_no)

The variable for sampling season (Season)

The variable for the type of AS-LUNG (AL_Type)

22. To export the dataset for 5-min average data for each subject

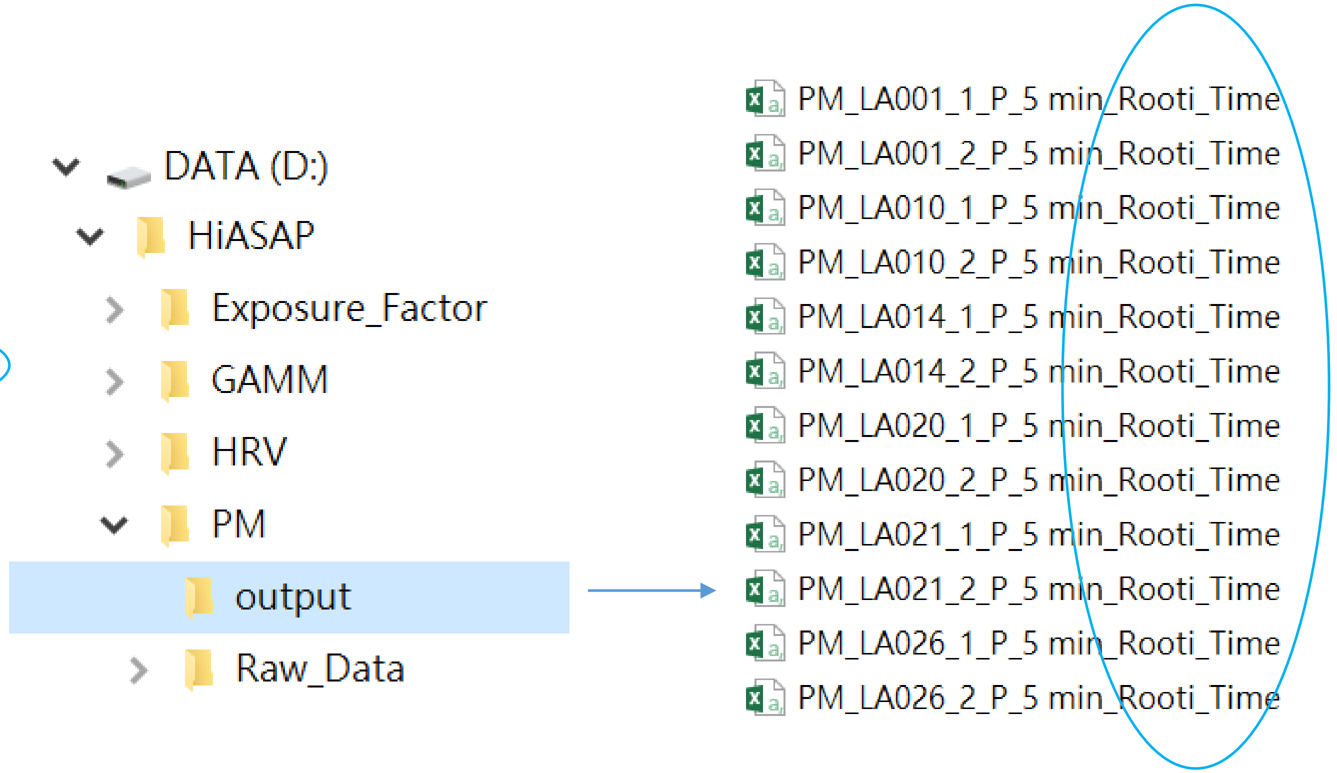
To combine “S_no”, “Season” and “AL_Type” variables with 5-min average data

```r
ALFinal2 <- data.frame(S_no,Season,AL_Type,subset(ALFinal2,select=c(Date,TEM,HUM,PM1,PM2.5,CO2)))
outputname<-paste0(substr(aa2[p],1,13),"5 min_Rooti_Time.csv")
write.csv(ALFinal2,outputname,row.names=FALSE,na="")
}
```

	S_no	Season	AL_Type	Date	TEM	HUM	PM1	PM2.5	CO2
1	LA001	0	3	2018-10-08 13:11:00	26.30500	78.89500	2.956250	3.378400	449.9500
2	LA001	0	3	2018-10-08 13:16:00	26.71000	78.17000	2.981800	3.487600	448.0500
3	LA001	0	3	2018-10-08 13:21:00	27.01000	77.31500	3.135100	3.633200	440.6500

23. To combine 5-min average data for all subjects

```r
# To combine PM data for all subjects
way <- paste0(location,"HiASAP/PM/output")
bb <- list.files(way,pattern='Rooti_Time')
filename <- paste0(way,"/",bb[1])
cc <- read.csv(filename)
ASLUNG <- cc
for(k in 2:length(bb)){
    filename <- paste0(way,"/",bb[k])
    cc <- read.csv(filename)
    ASLUNG <- rbind(ASLUNG,cc)
}
```



24. To create the variables of the year, month, day, hour and minute of the date

```r
# To create the time variables (year, month, day, hour and minute) for the following data matching
library('lubridate')
date_1 <- substr(ASLUNG$Date,1,10)
date_2 <- substr(ASLUNG$Date,12,16)
date_3 <- c(ymd_hm(paste(date_1,date_2)))
yy <- c(substr(date_3,1,4))
mn <- c(substr(date_3,6,7))
dd <- c(substr(date_3,9,10))
hh <- c(substr(date_3,12,13))
mm <- c(substr(date_3,15,16))
mm_30 <- c()
for (l in 1:length(mm)) {
    if(mm[l] < 30){
        mm_30[l] <- 1
    }else{
        mm_30[l] <- 2
    }
}
```

To get the year, day, hour and minute of the date

To create a variable of 30-minute interval of each hour for merging data with TAD

Year	Month	Day	Hour	Minute	Minute_3(
2018	10	8	13	46	2
2018	10	8	13	51	2
2018	10	8	13	56	2
2018	10	8	14	1	1
2018	10	8	14	6	1
2018	10	8	14	11	1

Time between 30 to 59 minutes -> 2

Time between 0 to 29 minutes -> 1

```r
ALfinal<-data.frame()
for(j in 1:length(date_3)[1]){
    ALfinal[j,1]<-date_3[j]
    ALfinal[j,2]<-yy[j]
    ALfinal[j,3]<-mn[j]
    ALfinal[j,4]<-dd[j]
    ALfinal[j,5]<-hh[j]
    ALfinal[j,6]<-mm[j]
    ALfinal[j,7]<-mm_30[j]
    ALfinal[j,8]<-ASLUNG$S_no[j]
    ALfinal[j,9]<-ASLUNG$Season[j]
    ALfinal[j,10]<-ASLUNG$AL_Type[j]
    ALfinal[j,11]<-ASLUNG$TEM[j]
    ALfinal[j,12]<-ASLUNG$HUM[j]
    ALfinal[j,13]<-ASLUNG$PM1[j]
    ALfinal[j,14]<-ASLUNG$PM2.5[j]
    ALfinal[j,15]<-ASLUNG$CO2[j]
}
colnames(ALfinal)<-c("Date","Year","Month","Day","Hour","Minute","Minute_30","S_no","Season","AL_Type","TEM","HUM","PM1","PM2.5","CO2")

outputname<-paste0("PM_5 min_Rooti_Time_All.csv")
write.csv(ALfinal,outputname,row.names=FALSE)
```

25. To combine variables created at step 24 with 5-min average data

26. To export final dataset of 5-min average data for all subjects

1	Date	Year	Month	Day	Hour	Minute	Minute_3(	S_no	Season	AL_Type	TEM	HUM	PM1	PM2.5	CO2
2	2018/10/8 13:11	2018	10	8	13	11	1	LA001	0	3	26.305	78.895	2.95625	3.3784	449.95
3	2018/10/8 13:16	2018	10	8	13	16	1	LA001	0	3	26.71	78.17	2.9818	3.4876	448.05
4	2018/10/8 13:21	2018	10	8	13	21	1	LA001	0	3	27.01	77.315	3.1351	3.6332	440.65
5	2018/10/8 13:26	2018	10	8	13	26	1	LA001	0	3	27.165	76.05	3.1862	3.6332	428.65
6	2018/10/8 13:31	2018	10	8	13	31	2	LA001	0	3	27.38	75.31	3.31395	3.797	426.5
7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Part 2: Questionnaire/time-activity diary (TAD) data processing

For $PM_{2.5}$ exposure-health evaluation, we only the location (microenvironment) information in TAD

Most of TAD data were used for assessing the $PM_{2.5}$ exposure factors, so data processing for TAD will present tomorrow

Questionnaire raw data

- Including basic information about subjects, life style, living environments, and etc.
- In this case, we only use the age, gender and BMI data obtained from the questionnaire

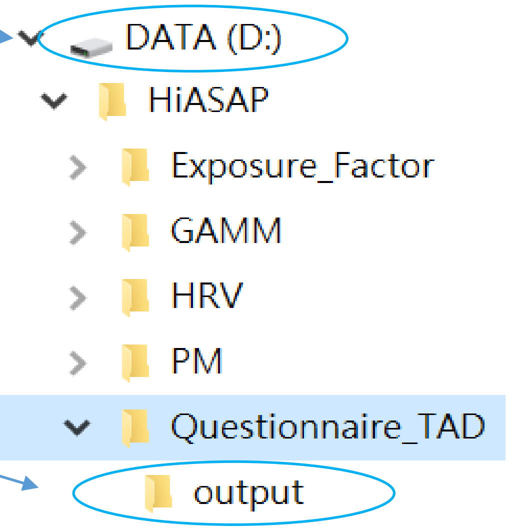
You can modify by yourself

```r
##Questionnaire data processing
#remove previous memory in R.
rm(list=ls())

location <- "D:/"

cmd1 <- paste0("setwd('",location,"HiASAP/Questionnaire_TAD/output')")
eval(parse(text=cmd1))
```

1. To remove all objects from current workspace

2. To set the default drive of your data

3. To set the path of output file

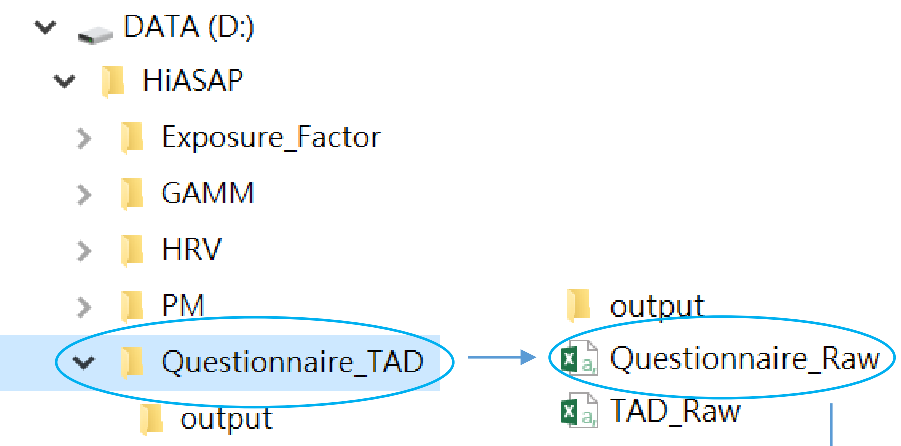


You can modify by yourself

```r
way <- paste0(location,"HiASAP/Questionnaire_TAD")

Q <- read.csv(paste0(way,"/Questionnaire_Raw.csv"))
```

4. To set the path of questionnaire data

5. To read the questionnaire raw data



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	S_no	Season	A_Gender	A_Birth	A_FBirthI	A_Educ	A_Marria;	A_Religio	A_Smoke	A_Smoke	A_Smoke	A_Smoke	A_Smoke	A_WD_C	A_WD_C	A_WD_C	A_WD_C(	A_WD_C	A_WD_C	A
2	LA001	0	2	48	1	4	3	8	1	999	999	999	999	40	0	30	20	0	0	
3	LA010	0	2	58	1	6	3	1	1	999	999	999	999	0	0	0	60	0	0	
4	LA014	0	1	38	1	2	3	1	3	30	60	1	60	0	0	12	0	20	0	
5	LA020	0	2	37	1	2	3	7	1	999	999	999	999	0	0	30	0	0	0	
6	LA021	0	2	44	1	2	6	1	2	997	997	997	997	60	0	0	0	0	2	
7	LA026	0	2	55	1	5	3	3	1	999	999	999	999	995	7.5	17.5	0	0	65	

Questionnaire data processing

```r
A_Age<-c()
for(i in 1:dim(Q)[1]){
    A_Age[i]<-107-Q$A_Birth[i]
}

A_Gender2 <- c()
for(i in 1:dim(Q)[1]){
    if(Q$A_Gender[i]==1){
        A_Gender2[i]<-1
    }else{
        A_Gender2[i]<-0
    }
}

C_BMI <- c()
for(i in 1:dim(Q)[1]){
    C_BMI[i]<-Q$C_Weight[i]/((Q$C_Height[i]/100)^2)
}
```

6. To calculate the age of subjects

7. To re-code the “gender” variable (male=“1” and female=“0”)

8. To calculate the body mass index (BMI) of subjects

9. To select variables which are used in the following analysis and combine the age, gender and BMI variables with the data

```r
Qfinal<-data.frame(subset(Q,select=c(S_no,Season)),A_Age,A_Gender2,C_BMI)

colnames(Qfinal)<-c("S_no","Season","Age","Gender","BMI")
outputname<-"2020_Training_Course_Questionnaire.csv"
write.csv(Qfinal,outputname,row.names=FALSE,na="")
```

10. To export the final questionnaire data

	A	B	C	D	E
1	S_no	Season	Age	Gender	BMI
2	LA001	0	59	0	30.04326
3	LA010	0	49	0	18.77834
4	LA014	0	69	1	25.82645
5	LA020	0	70	0	22.60026
6	LA021	0	63	0	21.77844

Part 3:
Generalized Additive Mixed Model (GAMM)

You can modify by yourself

```r
# To remove previous memory in R.
rm(list=ls())

# To install the R package (only for the first time to run)
#install.packages("mgcv")

# To load the R package
library(mgcv)

location <- "D:/"

# To set the output file
cmd1 <- paste0("setwd('",location,"HiASAP/GAMM/output')")
eval(parse(text=cmd1))

outputname <- "GAMM"
```

1. To remove all objects from current workspace

2. To install (only for the first time to load) and load the R package

3. To set the default drive of your data

4. To set the path and filename of output file

5. To merge (1) PM, (2) HRV, (3) questionnaire, (4) TAD and (5) meteorological data for GAMM

```r
# To combine PM, questionnaire, TAD and meteorological data with HRV data
PM <- read.csv(paste0(location,"HiASAP/PM/output/PM_5 min_Rooti_Time_All.csv"))
HRV <- read.csv(paste0(location,"HiASAP/HRV/output/HRV_5 minute_All.csv"))
HRV <- HRV[,c(2:25)]

PMall <- data.frame()
PMall <- merge(PM,HRV, by=c("Year","Month","Day","Month","Hour","Minute","Minute_30","S_no"))

PMall_2 <- data.frame()
QA <- read.csv(paste0(location,"HiASAP/Questionnaire_TAD/output/2020_Training_Course_Questionnaire.csv"))
PMall_2 <- merge(PMall,QA, by=c("S_no","Season"))

PMall_3 <- data.frame()
TAD <- read.csv(paste0(location,"HiASAP/Questionnaire_TAD/output/2020_Training_Course_TAD.csv"))
PMall_3 <- merge(PMall_2,TAD, by=c("Year","Month","Day","Month","Hour","Minute_30","S_no"))

Meteor <- read.csv(paste0(location,"HiASAP/Meteor/Meteor_Hourly_All.csv"))
Meteor <- Meteor[,c(1:4,14)]
PMall_4 <- merge(PMall_3,Meteor, by=c("Year","Month","Day","Month","Hour"))
```

(1) PM data

(2) HRV data

To merge data by ID of subjects and date

(3) Questionnaire data

(4) TAD data

(5) Meteorological data

```r
## To create a subject-day variable for autocorrelation adjustment
library(lubridate)
Time_1 <- ymd(paste0(PMall_4$Year,'-',PMall_4$Month,'-',PMall_4$Day))

S_no_Day <- c()
S_no_Day <- paste0(PMall_4$S_no,"_",Time_1)
```

6. To create a subject-day variable for autocorrelation in GAMM

- For example:
- To control the autocorrelation between 10:00 and 10:05

Subject-day	Time	$PM_{2.5}$
S_01_10/1	10:00	11.3
	10:05	12.7
	10:10	11.9
S_01_10/2	10:00	12.5
	10:05	11.2
	10:10	12.4
S_02_10/1	10:00	13.5
	10:05	12.4
	10:10	16.3
S_02_10/2	10:00	13.2
	10:05	10.9
	10:10	11.6

Autocorrelation adjustment

Autocorrelation adjustment

Autocorrelation adjustment

Autocorrelation adjustment

```r
## Log-transformed HRV
lg_HRsum5 <- log10(PMall_4$HRsum5)
lg_HRmean5 <- log10(PMall_4$HRmean5)
lg_SDNN5 <- log10(PMall_4$SDNN5)
lg_RMSSD5 <- log10(PMall_4$RMSSD5)
lg_LFHF5 <- log10(PMall_4$LFHF5)
lg_LF5 <- log10(PMall_4$LF5)
lg_HF5 <- log10(PMall_4$HF5)
lg_VLF5 <- log10(PMall_4$VLF5)
lg_TP5 <- log10(PMall_4$TP5)
```

7. To take base-10 logarithms of HRV indices due to the skewed distributions

```r
# To create the activity indexes
Activitymean <- (PMall_4$MeanX5^2+PMall_4$MeanY5^2+PMall_4$MeanZ5^2)^0.5
Activitymax <- (PMall_4$MaxX5^2+PMall_4$MaxY5^2+PMall_4$MaxZ5^2)^0.5
```

8. To create the activity indexes by calculating the Sum Vector (SV) of accelerations for three-axis

```r
# To create the variable of the time of day
Time<-PMall_4$Hour*12+PMall_4$Minute/5+1
```

9. To create a variable for controlling the time of the day

```r
Age_G<-c()
for(i in 1:dim(PMall_4)[1]){
    ifelse(PMall_4$Age[i]<60,Age_G[i]<-0,Age_G[i]<-1)
}
```

10. To group ages into 40 to 59 and 60 to 75 years old

```r
BMI_G<-c()
for(i in 1:dim(PMall_4)[1]){
    if(PMall_4$BMI[i]>=24){
        BMI_G[i]<-1
    }else{
        BMI_G[i]<-0
    }
}
```

11. To group BMI into <=24 (normal-weight) and > 24 (overweight and obese) kg/m^2

Based on the definition proposed by the Health Promotion Administration, Ministry of Health and Welfare in Taiwan

12. To combine the variables created at step 6-11 with dataset

```r
PMfinal <- data.frame(PMall_4,lg_HRsum5,lg_HRmean5,lg_SDNN5,lg_RMSSD5,lg_LFHF5,lg_LF5,lg_HF5,lg_VLF5,
 lg_TP5,Activitymean,Activitymax,Time,S_no_Day,Age_G,BMI_G)
write.csv(PMfinal,file=paste0(outputname,".csv"),row.names=FALSE)
```

13. To export the final dataset

14. To select data during no-raining and awake period

```r
# Select no-raining and awake period
PMfinal_A_NR <- PMfinal[which(PMfinal$Precp==0 & PMfinal$Sleep5==4),]
```

15. To run the GAMM for evaluating the effects of $PM_{2.5}$ on each HRV indices

```r
# To run the GAMM for each HRV indices
lg_SDNN<-gamm(lg_SDNN5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,
lg_LFHF<-gamm(lg_LFHF5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,
lg_HRsum<-gamm(lg_HRsum5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Ti
lg_HRmean<-gamm(lg_HRmean5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(
lg_RMSSD<-gamm(lg_RMSSD5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Ti
lg_LF<-gamm(lg_LF5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,bs=
lg_HF<-gamm(lg_HF5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,bs=
lg_VLF<-gamm(lg_VLF5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,bs
lg_TP<-gamm(lg_TP5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,bs=
```

GAMM

$$\begin{aligned} log(y) = & \beta_0 \longrightarrow \text{Intercept} \\ & \left.\begin{aligned} +\beta_1 x_{\text{PM2.5}} +\beta_2 x_{\text{Loc}} +\beta_3 x_{\text{Season}} +\beta_4 x_{\text{Age}} \\ +\beta_5 x_{\text{BMI}} +\beta_6 x_{\text{Gender}} +\beta_7 x_{\text{Temperature}} \end{aligned}\right\} \text{Linear terms} \\ & +f(x_{\text{Activity}}) + f(x_{\text{Time}}) \longrightarrow \text{Smooth terms} \\ & +\gamma_{\text{subject}} \longrightarrow \text{Random effect} \\ & +\varepsilon \longrightarrow \text{Error term} \end{aligned}$$

```
s(Time,bs=c("cc"))
s(R_Activitymean,bs=c("tp"))
```

Smooth terms

R code for GAMM:

① Dependent variable = ② independent variable (fixed effect, including linear and non-linear variables)

Y = x1 + x2 + …… +xi

```
lg_SDNN5~PM2.5+Loc_In+Season+A_Age+C_BMI+s(R_Activitymean,bs=c("tp"))+A_Gender_2+TEM+s(Time,bs=c("cc"))
,data=PMfinal_A_NR,random=list(S_no=~1),correlation=corCAR1(form=~Time|S_no_Day))
```

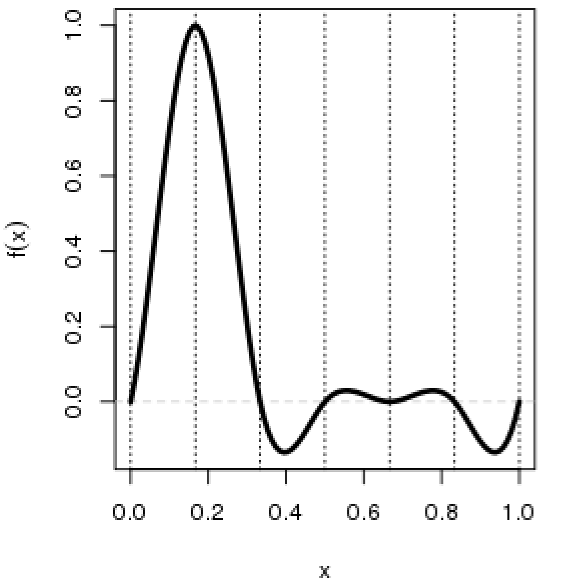
Dataset

③ Random effects

④ Autocorrelations

Smooth terms in GAMM

- TP
  - Thin plate regression splines
  - Default smooth for s terms because there is a defined sense in which they are the optimal smoother of any given basis dimension/rank (Wood, 2003)
- CC
  - One of the cubic regression splines
  - A cyclic cubic regression splines
  - A penalized cubic regression splines whose ends match, up to second derivative.
- More information can be found on the following website:
- https://stat.ethz.ch/R-manual/R-devel/library/mgcv/html/smooth.terms.html



Cyclic cubic spline basis functions

16. To export results of GAMM (in two ways)

```r
# Directly show results in the Console window
summary(lg_SDNN$gam)
summary(lg_LFHF$gam)
summary(lg_HRsum$gam)
summary(lg_HRmean$gam)
summary(lg_RMSSD$gam)
summary(lg_LF$gam)
summary(lg_HF$gam)
summary(lg_VLF$gam)
summary(lg_TP$gam)
```

To print out the results in the Console window

```r
# To print out GAMM results to txt file
sink("GAMM_Results.txt") # redirect console output to a file
print(summary(lg_SDNN$gam))
print(summary(lg_LFHF$gam))
print(summary(lg_HRsum$gam))
print(summary(lg_HRmean$gam))
print(summary(lg_RMSSD$gam))
print(summary(lg_LF$gam))
print(summary(lg_HF$gam))
print(summary(lg_VLF$gam))
print(summary(lg_TP$gam))
sink()   # close connection to file
```

To export the results to the txt file

GAMM results

```
# Directly show results in the Console window
summary(lg_SDNN$gam)
summary(lg_LFHF$gam)
summary(lg_HRsum$gam)
summary(lg_HRmean$gam)
summary(lg_RMSSD$gam)
summary(lg_LF$gam)
summ
summ
summ
```

To print out the results in the Console window

Console Terminal

D:/2019_BQ/BQ_All/BQ_All_Models/output/

```
> summary(lg_TP$gam)

Family: gaussian
Link function: identity

Formula:
lg_TP5 ~ ALP_PM2.5 + Loc_Out + Season + Age_G + BMI_G + s(R_Activitymean,
    bs = c("tp")) + A_Gender_2 + UAP_TEM + s(Time, bs = c("cc"))

Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.8404355  0.1281764  22.160  < 2e-16 ***
ALP_PM2.5   -0.0020581  0.0008209  -2.507 0.012221 *
Loc_Out      0.0095027  0.0434514   0.219 0.826901
Season       0.0418999  0.0260668   1.607 0.108068
Age_G       -0.1220361  0.0290735  -4.197 2.77e-05 ***
BMI_G       -0.0974044  0.0352755  -2.761 0.005792 **
A_Gender_2   0.2639532  0.0500894   5.270 1.46e-07 ***
UAP_TEM      0.0160448  0.0045664   3.514 0.000448 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```r
# To print out GAMM results to txt file
sink("GAMM_Results.txt") # redirect console output to a file
print(summary(lg_SDNN$gam))
print(summary(lg_LFHF$gam))
print(summary(lg_HRsum$gam))
print(summary(lg_HRmean$gam))
print(summary(lg_RMSSD$gam))
print(summary(lg_LF$gam))
print(summary(lg_HF$gam))
print(summary(lg_VLF$gam))
print(summary(lg_TP$gam))
sink()   # close connection t
```

To export the results to the txt file

名稱	修改日期	類型	大小
GAMM_Results	2020/9/28 上午 09:11	文字文件	11 KB

GAMM_Results - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
Family: gaussian
Link function: identity

Formula:
lg_SDNN5 ~ ALP_PM2.5 + Loc_Out + Season + Age_G + BMI_G + s(R_Activitymean,
    bs = c("tp")) + A_Gender_2 + UAP_TEM + s(Time, bs = c("cc"))

Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.4626672  0.0733606  19.938  < 2e-16 ***
ALP_PM2.5   -0.0011442  0.0004618  -2.478   0.0133 *
Loc_Out     -0.0060102  0.0248617  -0.242   0.8090
Season       0.0098366  0.0149795   0.657   0.5114
Age_G       -0.0815214  0.0121410  -6.715 2.23e-11 ***
BMI_G       -0.0776092  0.0147042  -5.278 1.40e-07 ***
A_Gender_2   0.1838930  0.0209695   8.770  < 2e-16 ***
UAP_TEM      0.0061343  0.0026278   2.334   0.0196 *
---
Signif. codes:  0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1

Approximate significance of smooth terms:
                  edf Ref.df     F  p-value
s(R_Activitymean) 1.000      1 0.055    0.815
s(Time)           3.383      8 3.474 7.32e-07 ***
---
Signif. codes:  0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1

R-sq.(adj) =  0.0558
  Scale est. = 0.042158   n = 3132
```

GAMM results

```
Family: gaussian
Link function: identity
```

① Formula (Equation) of GAMM

```
Formula:
lg_SDNN5 ~ PM2.5 + Loc_Out + Season + Age_G + BMI_G + s(Activitymean,
    bs = c("tp")) + Gender + TEM + s(Time, bs = c("cc"))
```

```
Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.997170   0.098426  10.131  < 2e-16 ***
PM2.5       -0.002243   0.000114 -19.663  < 2e-16 ***
Loc_Out     -0.063581   0.027778  -2.289 0.022152 *
Season       0.068763   0.019058   3.608 0.000314 ***
Age_G       -0.095829   0.032788  -2.923 0.003496 **
BMI_G       -0.374163   0.040184  -9.311  < 2e-16 ***
Gender       0.263251   0.057101   4.610 4.18e-06 ***
TEM          0.015707   0.003397   4.624 3.92e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

② Results of linear parameters

Effects can be quantified

```
Approximate significance of smooth terms:
                  edf Ref.df     F  p-value
s(Activitymean) 1.000      1 5.828   0.0158 *
s(Time)         4.089      8 6.791  8.7e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

③ Results of smooth terms

```
R-sq.(adj) =  0.539
  Scale est. = 0.028199   n = 3084
```

④ Adjusted R^2 and sample size (n)

GAMM results – ② Results of linear parameters (Take SDNN as an example)

β (estimated coefficients)

SE (standard error)

p value

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	0.997170	0.098426	10.131	< 2e-16	***
PM2.5	-0.002243	0.000114	-19.663	< 2e-16	***
Loc_Out	-0.063581	0.027778	-2.289	0.022152	*
Season	0.068763	0.019058	3.608	0.000314	***
Age_G	-0.095829	0.032788	-2.923	0.003496	**
BMI_G	-0.374163	0.040184	-9.311	< 2e-16	***
Gender	0.263251	0.057101	4.610	4.18e-06	***
TEM	0.015707	0.003397	4.624	3.92e-06	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05

$PM_{2.5}$ effects were expressed as percent changes by interquartile range (IQR) changes as:

$$[10^{(\beta * IQR)} - 1] * 100\%$$

and with 95% confidence intervals (CI) as:

$$[10^{((\beta \pm 1.96 * \text{standard error}) * IQR)} - 1] * 100\%$$

for HRV indices

	A	B	C	D	E	F
1	IQR	β	SE	10^(β*IQR)-1*100	((10^((B+1.96*SE)*IQR)-1))*100	((10^((B-1.96*SE)*IQR)-1))*100
2	11.4	-0.00224	0.000114	-5.73	-5.18	-6.28

IQR = 75^{th} percentile – 25^{th} percentile

GAMM results – ② Results of linear parameters (Take SDNN as an example)

	$PM_{2.5}$ (μg/m^3)		
	Percentage change [a]	95% CI	*p*-value
SDNN	-5.73	-6.28, -5.18	<0.001

[a] Percentage change in HRV indices for interquartile range (IQR) increases in $PM_{2.5}$ exposure in models adjusted for subject, age, gender, body mass index (BMI), location, season, temperature, activity, and time of day. CI, confidence interval.

- Increase in $PM_{2.5}$ concentration of one interquartile range (IQR) (11.4 μg/m^3) was associated with a change of -5.73% SDNN.

GAMM results – ③ Results of smooth terms and ④ Adjusted R^2 and sample size (Take SDNN as an example)

reference number of degrees of freedom

estimated degrees of freedom

p value

```
Approximate significance of smooth terms:
                   edf Ref.df     F  p-value
s(Activitymean) 1.000      1 5.828  0.0158 *
s(Time)         4.089      8 6.791 8.7e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Adjusted R^2

This model could explains 53.9% of variance in the dependent variable

```
R-sq.(adj) =  0.539
  Scale est. = 0.028199  n = 3084
```

Sample size

Thank you for your attention

Appendix

GAMM

Main effect: $PM_{2.5}$, in epidemiology, we only care about the coefficient of the main effect

$log(y)=\beta_0$ ⟶ Intercept

$+\beta_1 X_{PM2.5}$

$+\beta_2 X_{Loc} + \beta_3 X_{Season} + \beta_4 X_{Age}$

$+\beta_5 X_{BMI} + \beta_6 X_{Gender} + \beta_7 X_{Temperature}$

Linear terms

$+f(X_{Activity}) + f(X_{Time})$ ⟶ Smooth terms

$+\gamma_{subject}$ ⟶ Random effect

$+\varepsilon$ ⟶ Error term

- The other variables are adjustment variables, which means these variables also have impacts on Y (HRV in our case). Thus, we need to "adjust for" these variables in order to estimate accurately the impact (β_1) of the main effect ($PM_{2.5}$) on HRV. [ex. season, ag, BMI, gender, activity, time, subject…] We don't care about their coefficients.
- for meteorological parameters, since temperature and humidity have high correlations, we only put temperature in this case; for future Hi-ASAP studies, we may consider to adjust for humidity not temperatur

$log(y)=\beta_0$ ⟶ Intercept

$+\beta_1 X_{PM2.5} + \beta_8 X^2_{PM2.5} + \beta_9 X^3_{PM2.5}$….

$+\beta_2 X_{Loc} + \beta_3 X_{Season} + \beta_4 X_{Age}$

$+\beta_5 X_{BMI} + \beta_6 X_{Gender} + \beta_7 X_{Temperature}$

Linear terms

$+f(X_{Activity}) + f(X_{Time})$ ⟶ Smooth terms

$+\gamma_{subject}$ ⟶ Random effect

$+\varepsilon$ ⟶ Error term

Main effect: if your main variable $PM_{2.5}$, has non-linear relationship with Y (HRV in this case), you may put in the second or third orders of the main variable (polynomial) into this GAMM to get their coefficients. However, epidemiologists seldom did that

```r
## To create a subject-day variable for autocorrelation adjustment
library(lubridate)
Time_1 <- ymd(paste0(PMall_4$Year,'-',PMall_4$Month,'-',PMall_4$Day))

S_no_Day <- c()
S_no_Day <- paste0(PMall_4$S_no,"_",Time_1)
```

6. To create a subject-day variable for autocorrelation in GAMM

- For example:
- To control the autocorrelation between 10:00 and 10:05

Subject-day	Time	$PM_{2.5}$
S_01_10/1	10:00	11.3
	10:05	12.7
	10:10	11.9
S_01_10/2	10:00	12.5
	10:05	11.2
	10:10	12.4
S_02_10/1	10:00	13.5
	10:05	12.4
	10:10	16.3
S_02_10/2	10:00	13.2
	10:05	10.9
	10:10	11.6

Autocorrelation adjustment

Autocorrelation adjustment

Autocorrelation adjustment

Autocorrelation adjustment

GAMM results – ② Results of linear parameters (Take SDNN as an example)

SE (standard error)

β (estimated coefficients)

p value

```
Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.997170   0.098426  10.131  < 2e-16 ***
PM2.5       -0.002243   0.000114 -19.663  < 2e-16 ***
Loc_Out     -0.063581   0.027778  -2.289 0.022152 *
Season       0.068763   0.019058   3.608 0.000314 ***
Age_G       -0.095829   0.032788  -2.923 0.003496 **
BMI_G       -0.374163   0.040184  -9.311  < 2e-16 ***
Gender       0.263251   0.057101   4.610 4.18e-06 ***
TEM          0.015707   0.003397   4.624 3.92e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
```

$PM_{2.5}$ effects were expressed as percent changes by interquartile range (IQR) changes as:

$$[10^{(\beta*IQR)}-1]*100\%$$

and with 95% confidence intervals (CI) as:

$$[10^{((\beta\pm1.96*\text{standard error})*IQR)}-1]*100\%$$

for HRV indices

	A	B	C	D	E	F
1	IQR	β	SE	10^(β*IQR)-1*100	((10^((B+1.96*SE)*IQR)-1))*100	((10^((B-1.96*SE)*IQR)-1))*100
2	11.4	-0.00224	0.000114	-5.73	-5.18	-6.28

IQR = 75^{th} percentile – 25^{th} percentile

GAMM results – ③ Results of smooth terms and ④ Adjusted R^2 and sample size (Take SDNN as an example)

reference number of degrees of freedom

estimated degrees of freedom

p value

```
Approximate significance of smooth terms:
                  edf Ref.df     F  p-value
s(Activitymean) 1.000      1 5.828  0.0158 *
s(Time)         4.089      8 6.791 8.7e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Adjusted R^2

This model could explains 53.9% of variance in the dependent variable

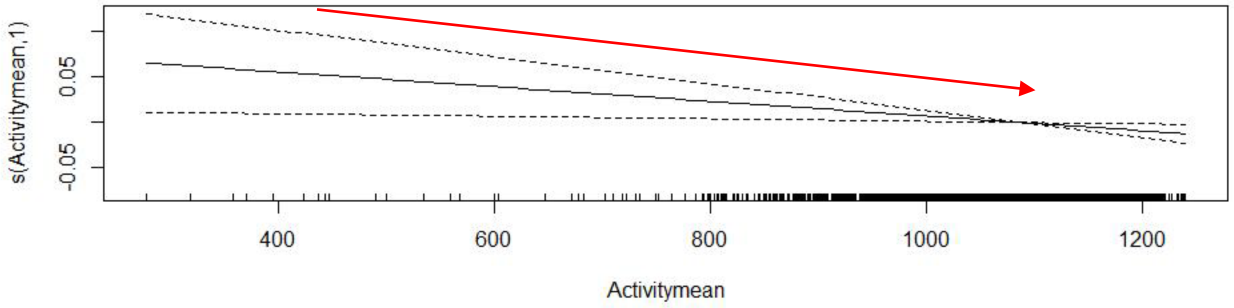
```
R-sq.(adj) =  0.539
  Scale est. = 0.028199  n = 3084
```

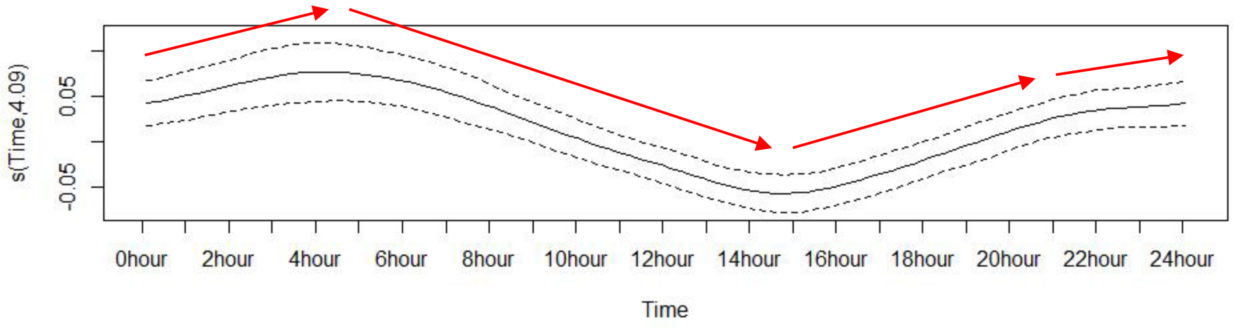
Sample size

Plots for smooth terms - SDNN

- Results of smooth terms of GAMM for SDNN

```
Approximate significance of smooth terms:
                  edf Ref.df     F  p-value
s(Activitymean) 1.000      1 5.828   0.0158 *
s(Time)         4.089      8 6.791  8.7e-13 ***
```





Plots for smooth terms – LF/HF ratio

- Results of smooth terms of GAMM for LF/HF ratio

```
Approximate significance of smooth terms:
                  edf Ref.df     F  p-value
s(Activitymean) 4.997  4.997 5.939 1.99e-05 ***
s(Time)         3.734  8.000 5.203 5.09e-10 ***
```