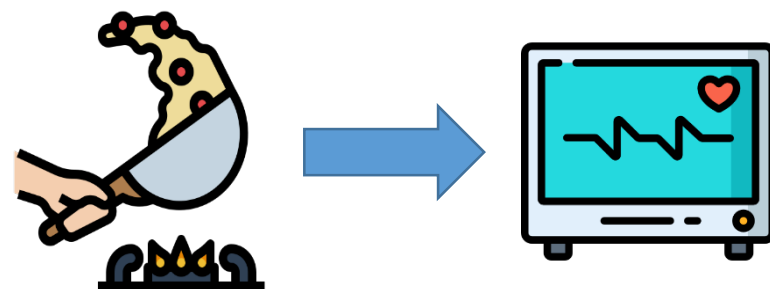




Data analysis on quantification of PM_{2.5} exposure-health evaluation

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:
Questionnai
re/time-
activity
diary (TAD)
data
processing

Part 3:
Generalized
Additive
Mixed Mod
el (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
 - To control autocorrelation


Notices

- All text after the pound sign "#" within the same line in R code
 - To be considered a comment
 - To be not run by R
- In this presentation, the R codes in the red box can be modify according your requirements (mostly the path of data file)
 - For example, you can modify the path of output file by yourself

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



Notices

- To use quotes (") to tell R to interpret something as a string.
 - For example,
`location <- "D:/"`  `D:/`
- The uppercase letters (strings) are different from the lowercase letters (strings)
 - For example,
 - `A` \neq `a`
 - `Apple` \neq `apple`

Part 1:

PM data processing

(matching the time of heart rate variability monitoring)



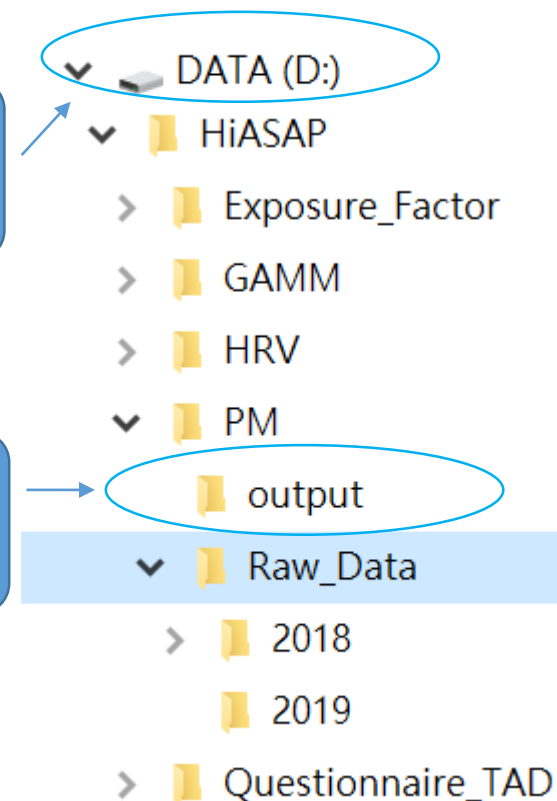
→ You can modify by yourself

1. To remove all objects from current workspace

```
1 # To remove previous memory in R.  
2 rm(list=ls())  
3  
4 location <- "D:/"  
5  
6 # To set the output file  
7 cmd1 <- paste0("setwd('",location,"HiASAP/PM/output'")  
8 eval(parse(text=cmd1))  
9
```

2. To set the default drive of your data

3. To set the path of output file





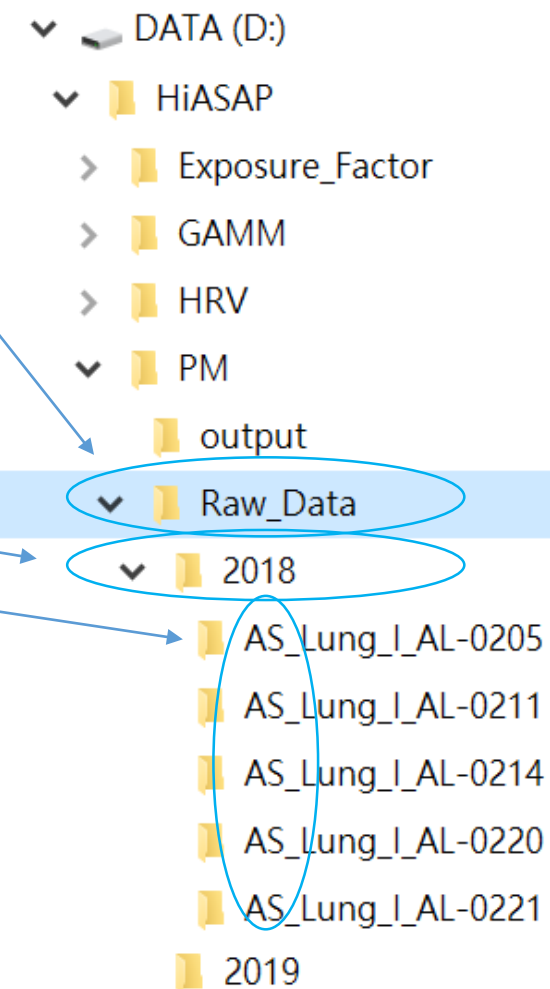
→ You can modify by yourself

To import the PM data by subjects/AS-LUNG

4. To set the path of PM data

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

To set the pattern
(keyword) to select files



To combine the PM data by subjects/AS-LUNG

```
16 ▾ for(p in 1:length(aa1)){  
17     ASLUNG <- data.frame()  
18     fileloc <- paste0(way2,"/",aa1[p])  
19     bb <- list.files(fileloc,pattern='csv')  
20     filename <- paste0(fileloc,"/",bb[1])  
21     cc <- read.csv(filename)  
22     ASLUNG <- cc  
23 ▾ for(k in 2:length(bb)){  
24     filename <- paste0(fileloc,"/",bb[k])  
25     cc <- read.csv(filename)  
26     ASLUNG <- rbind(ASLUNG,cc)  
27 ▴ }
```

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	datetime	id	date	time	sht_t	sht_h	pm1	pm25	pm10	co2	adc	acc_x	acc_y	acc_z	accx_int	accy
2	#####	0C9A4249	#####	00:00:00	26.6	67.1	13	17	18	552	1	-1	-66	-2	60	
3	#####	0C9A4249	#####	00:00:15	26.6	67.1	13	17	18	555	1	-1	-66	0	7	
4	#####	0C9A4249	#####	00:00:30	26.6	67.1	13	16	17	558	1	-2	-66	-2	81	
5	#####	0C9A4249	#####	00:00:45	26.6	67.1	13	16	17	558	1	-1	-66	-2	75	
6	#####	0C9A4249	#####	00:01:00	26.6	67.1	13	16	17	558	1	-2	-66	-2	67	
7	#####	0C9A4249	#####	00:01:15	26.6	67.1	13	15	16	558	1	-1	-68	-2	60	
8	#####	0C9A4249	#####	00:01:30	26.6	67.1	13	16	16	558	1	-1	-66	-2	80	
9	#####	0C9A4249	#####	00:01:45	26.6	67.1	13	16	17	558	1	1	-66	0	60	

```

29 # To select the variables of time, temperature, relative humidity, CO2, corrected PM1 and corrected PM2.5
30 ASLUNGt <- data.frame(subset(ASLUNG,select=c(datetime,date,time,sht_t,sht_h,co2,cPM1,cPM2.5)))
31

```

6. To select interested variables from original dataset
(time, temperature, relative humidity, corrected PM₁ and corrected PM_{2.5})

```

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

```

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

To create variables for the following analysis

```
# 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"

```
# 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
personal version = "3"

60
61
62
63
64
65

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

10. To combine the created variables with dataset

```
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="")
```

11. To export the dataset

	A	B	C	D	E	F	G	H	I	J	K
1	Season	AL_Type	datetime	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

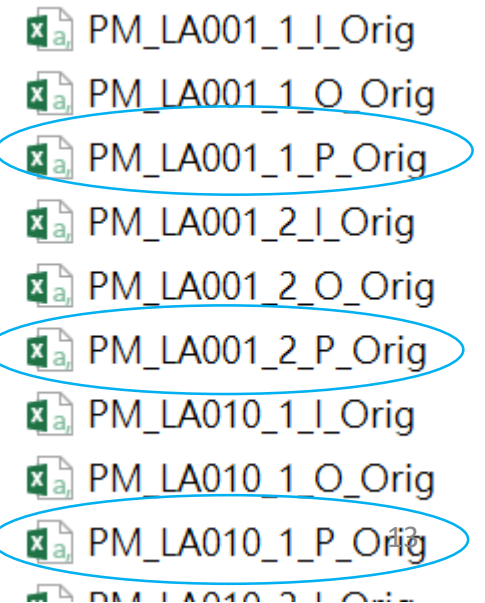
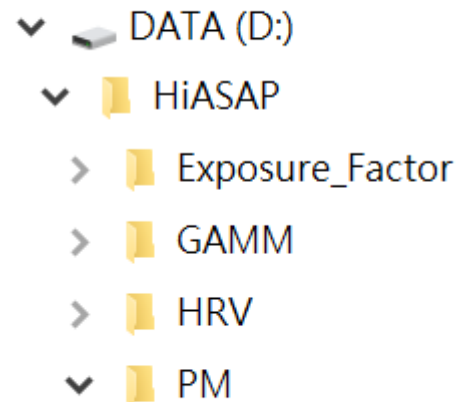
```
67 # To calculate 5-min average PM data for Generalized Additive Mixed Model (GAMM) (based on Rooti time)
68 way_Rooti <- paste0(location,"HiASAP/HRV/Raw_Data/")
```

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

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

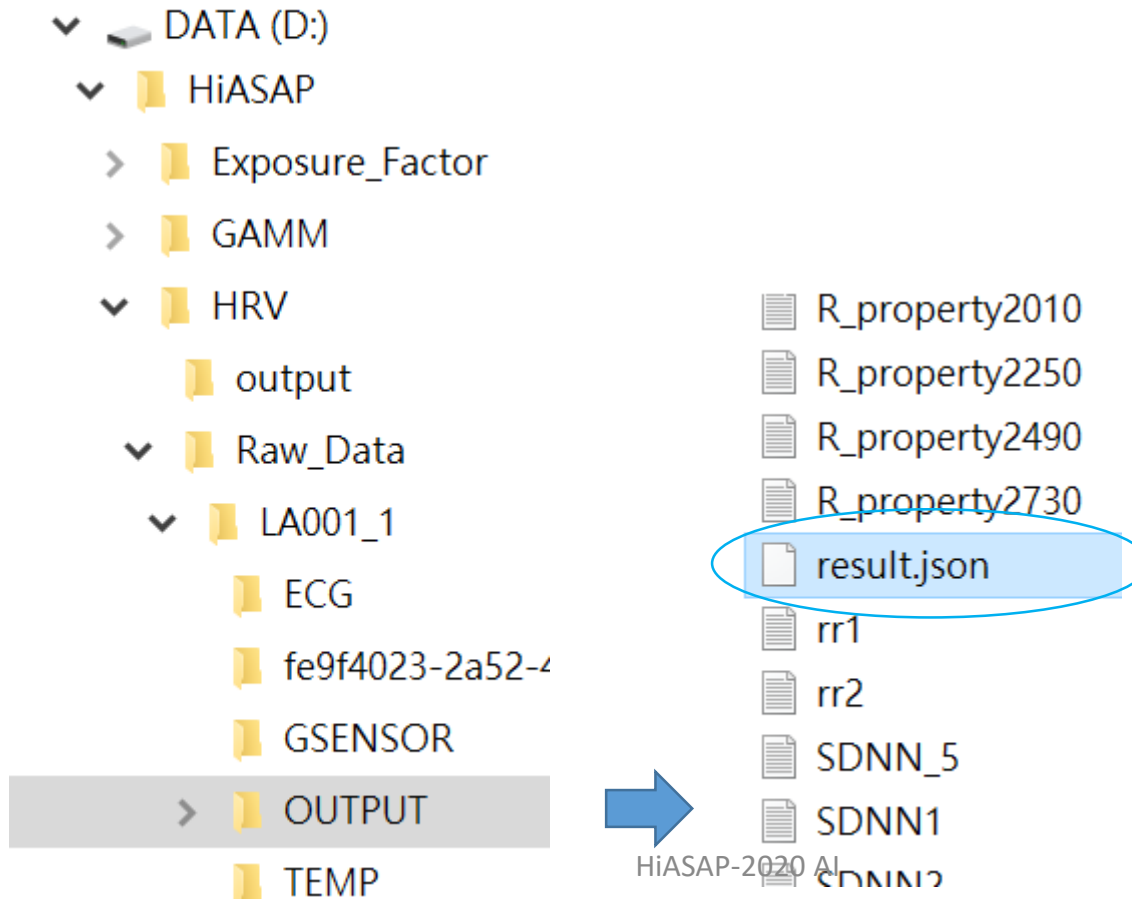
13. To set the path of PM data which we export in step 11

To set the files of personal data



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

```
70 for (p in 1:length(aa2)) {  
71   # To read the time of heart rate variability monitoring form Rooti  
72   filename <- paste0(location,"HiASAP/PM/output/",aa2[p])  
73   ASLUNGt <- read.csv(filename)  
74   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

```
75 start_time<-Rooti_online_result$activity$startTime
76 start_time<-as.POSIXct(start_time, origin="1970-01-01",tz='Asia/Taipei')
77 end_time<-Rooti_online_result$activity$endTime
78 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

Name	Type
Rooti_online_result	list [1]
[[1]]	list [7]
mode	integer [1]
Q_factor	list [1 x 4] (S3: data.frame)
id	character [1]
af	list [1 x 25] (S3: data.frame)
hrv	list [2 x 12] (S3: data.frame)
activity	list [1 x 3] (S3: data.frame)
startTime	integer [1]
id	character [1]
endTime	integer [1]
sleep	list [2 x 12] (S3: data.frame)

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

Example:

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

```
80 from <- as.POSIXct(substr(start_time,1,16),tz="Asia/Taipei")
81 to <- as.POSIXct(substr(end_time,1,16),tz="Asia/Taipei")
82 sort_out_time<-data.frame(date=seq.POSIXt(from, to, by = "15 secs",tz="Asia/Taipei"))
83 Date_AL<-seq.POSIXt(from, to, by = "15 secs",tz="Asia/Taipei")
84 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

```
86 date_1<-c(ymd(as.character(ASLUNGt$date)))
87 time<-substr(ASLUNGt$time,1,8)
88 date<-paste(date_1,time)
89 ASLUNGt2<-data.frame(date,ASLUNGt)
90
91 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

```

93 # If the start time of PM monitoring was different from the start time of HRV monitoring,
94 # we add the start time of HRV monitoring to PM data to have the consistent 5-min interval
95 if((substr(ASLUNGt2$date[1],1,16))!=(substr(Date_AL2[1],1,16))){
96     dd <- as.POSIXct(Date_AL2[1],tz="Asia/Taipei")
97     Add_Row <- data.frame()
98     Add_Row <- data.frame(date=as.factor(dd),Season="",AL_Type="",datetime="",date.1="",time="",sht_t="",sht_h="",co2="",cPM1="",cPM2.5="")
99     ASLUNGt3<- rbind(ASLUNGt2,Add_Row)
100 }else{
101     ASLUNGt3<- ASLUNGt2
102 }
103 ASLUNGt3 <- merge(ASLUNGt3,sort_out_time,by="date")

```

	date	Season	AL_Type	datetime	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	26.3	78.9	445	2.573	3.251

19. To format the time variables again after step 18

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

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

20. To calculate the 5-min average PM data

```
108 # To calculate 5-min average PM data
109 ALFinal <-ASLUNGt3 %>%
110   group_by(date = cut(Date_AL2, breaks="300 secs")) %>%
111   summarize(
112     TEM = mean(as.numeric(sht_t), na.rm = TRUE),
113     HUM = mean(as.numeric(sht_h), na.rm = TRUE),
114     PM1 = mean(as.numeric(cPM1), na.rm = TRUE),
115     PM2.5 = mean(as.numeric(cPM2.5), na.rm = TRUE),
116     CO2 = mean(as.numeric(co2), na.rm = TRUE),
117     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

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

```

118 ALFinal$date <-ymd_hms(ALFinal$date,tz="Asia/Taipei")
119 ALFinal2 <- ALFinal[which(ALFinal$Freq>=10),]
120 colnames(ALFinal2)<-c("Date","TEM","HUM","PM1","PM2.5","CO2","Freq")
121
122 S_no<-substr(aa2[p],4,8) → The variable for ID of subjects (S_no)
123
124 Season <- as.numeric(substr(aa2[p],10,10))-1 → The variable for sampling season (Season)
125
126 AL_Type<-3 → 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

```

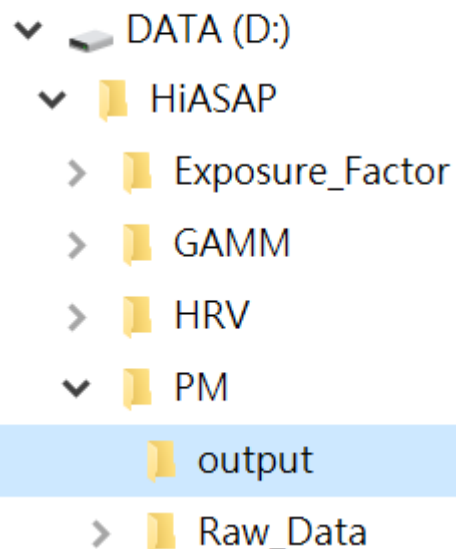
128 ALFinal2 <- data.frame(S_no,Season,AL_Type,subset(ALFinal2,select=c(Date,TEM,HUM,PM1,PM2.5,CO2)))
129 outputname<-paste0(substr(aa2[p],1,13),"5 min_Rooti_Time.csv")
130 write.csv(ALFinal2,outputname,row.names=FALSE,na="")
131 }

```

	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

```
133 # To combine PM data for all subjects
134 way <- paste0(location,"HiASAP/PM/output")
135 bb <- list.files(way,pattern='Rooti_Time')
136 filename <- paste0(way,"/",bb[1])
137 cc <- read.csv(filename)
138 ASLUNG <- cc
139 for(k in 2:length(bb)){
140     filename <- paste0(way,"/",bb[k])
141     cc <- read.csv(filename)
142     ASLUNG <- rbind(ASLUNG,cc)
143 }
```



PM_LA001_1_P_5 min_Rooti_Time
PM_LA001_2_P_5 min_Rooti_Time
PM_LA010_1_P_5 min_Rooti_Time
PM_LA010_2_P_5 min_Rooti_Time
PM_LA014_1_P_5 min_Rooti_Time
PM_LA014_2_P_5 min_Rooti_Time
PM_LA020_1_P_5 min_Rooti_Time
PM_LA020_2_P_5 min_Rooti_Time
PM_LA021_1_P_5 min_Rooti_Time
PM_LA021_2_P_5 min_Rooti_Time
PM_LA026_1_P_5 min_Rooti_Time
PM_LA026_2_P_5 min_Rooti_Time

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

```

145 # To create the time variables (year, month, day, hour and minute) for the following data matching
146 library('lubridate')
147 date_1 <- substr(ASLUNG$Date,1,10)
148 date_2 <- substr(ASLUNG$Date,12,16)
149 date_3 <- c(ymd_hm(paste(date_1,date_2)))
150 yy <- c(substr(date_3,1,4))
151 mn <- c(substr(date_3,6,7))
152 dd <- c(substr(date_3,9,10))
153 hh <- c(substr(date_3,12,13))
154 mm <- c(substr(date_3,15,16))
155 mm_30 <- c()
156 for (l in 1:length(mm)) {
157   if(mm[l] < 30){
158     mm_30[l] <- 1
159   }else{
160     mm_30[l] <- 2
161   }
162 }

```

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_30
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

```

163 ALfinal<-data.frame()
164 for(j in 1:length(date_3)[1]){
165     ALfinal[j,1]<-date_3[j]
166     ALfinal[j,2]<-yy[j]
167     ALfinal[j,3]<-mn[j]
168     ALfinal[j,4]<-dd[j]
169     ALfinal[j,5]<-hh[j]
170     ALfinal[j,6]<-mm[j]
171     ALfinal[j,7]<-mm_30[j]
172     ALfinal[j,8]<-ASLUNG$S_no[j]
173     ALfinal[j,9]<-ASLUNG$Season[j]
174     ALfinal[j,10]<-ASLUNG$AL_Type[j]
175     ALfinal[j,11]<-ASLUNG$TEM[j]
176     ALfinal[j,12]<-ASLUNG$HUM[j]
177     ALfinal[j,13]<-ASLUNG$PM1[j]
178     ALfinal[j,14]<-ASLUNG$PM2.5[j]
179     ALfinal[j,15]<-ASLUNG$CO2[j]
180 }
181 colnames(ALfinal)<-c("Date","Year","Month","Day","Hour","Minute","Minute_30","S_no","Season","AL_Type","TEM","HUM","PM1","PM2.5","CO2")
182
183 outputname<-paste0("PM_5 min_Rooti_Time_All.csv")
184 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_30	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	2020/10/5 12:26	2020	10	5	12	26	HiASAP-2020 AI		0	3	27.18880	75.78880	2.528220	2.870	507.27722

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



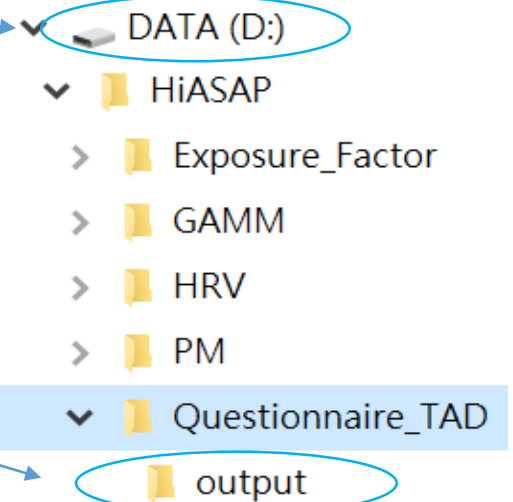
→ You can modify by yourself

```
1 ##Questionnaire data processing
2 #remove previous memory in R.
3 rm(list=ls())
4
5 location <- "D:/"
6
7 cmd1 <- paste0("setwd('",location,"HiASAP/Questionnaire_TAD/output'")
8 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

```
10 way <- paste0(location,"HiASAP/Questionnaire_TAD")
11
12 Q <- read.csv(paste0(way,"/Questionnaire_Raw.csv"))
```

4. To set the path of questionnaire data

5. To read the questionnaire raw data

DATA (D:)

HiASAP

Exposure_Factor

GAMM

HRV

PM

Questionnaire_TAD

output

output

Questionnaire_Raw

TAD_Raw

	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_FBirth	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

```
14 A_Age<-c()  
15 for(i in 1:dim(Q)[1]){  
16   A_Age[i]<-107-Q$A_Birth[i]  
17 }
```

6. To calculate the age of subjects

```
18  
19 A_Gender2 <- c()  
20 for(i in 1:dim(Q)[1]){  
21   if(Q$A_Gender[i]==1){  
22     A_Gender2[i]<-1  
23   }else{  
24     A_Gender2[i]<-0  
25   }  
26 }
```

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

```
27  
28 C_BMI <- c()  
29 for(i in 1:dim(Q)[1]){  
30   C_BMI[i]<-Q$C_Weight[i]/((Q$C_Height[i]/100)^2)  
31 }
```

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

```
33 Qfinal<-data.frame(subset(Q,select=c(S_no,Season)),A_Age,A_Gender2,C_BMI)
34
35 colnames(Qfinal)<-c("S_no","Season","Age","Gender","BMI")
36 outputname<-"2020_Training_Course_Questionnaire.csv"
37 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
7	LA026	0	50	0	20.2408

Part 3:

Generalized Additive Mixed Model (GAMM)



→ You can modify by yourself

```
1 # To remove previous memory in R.  
2 rm(list=ls())  
3  
4 # To install the R package (only for the first time to run)  
5 #install.packages("mgcv")  
6  
7 # To load the R package  
8 library(mgcv)  
9  
10 location <- "D:/"  
11  
12 # To set the output file  
13 cmd1 <- paste0("setwd('",location,"HiASAP/GAMM/output'")  
14 eval(parse(text=cmd1))  
15  
16 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

```
18 # To combine PM, questionnaire, TAD and meteorological data with HRV data (1) PM data
19 PM <- read.csv(paste0(location,"HiASAP/PM/output/PM_5 min_Rooti_Time_All.csv"))
20 HRV <- read.csv(paste0(location,"HiASAP/HRV/output/HRV_5 minute_All.csv")) (2) HRV data
21 HRV <- HRV[,c(2:25)]
22
23 PMA11 <- data.frame()
24 PMA11 <- merge(PM,HRV, by=c("Year","Month","Day","Month","Hour","Minute","Minute_30","S_no"))
25
26 PMA11_2 <- data.frame()
27 QA <- read.csv(paste0(location,"HiASAP/Questionnaire_TAD/output/2020_Training_Course_Questionnaire.csv"))
28 PMA11_2 <- merge(PMA11,QA, by=c("S_no","Season")) (3) Questionnaire data
29
30 PMA11_3 <- data.frame()
31 TAD <- read.csv(paste0(location,"HiASAP/Questionnaire_TAD/output/2020_Training_Course_TAD.csv"))
32 PMA11_3 <- merge(PMA11_2,TAD, by=c("Year","Month","Day","Month","Hour","Minute_30","S_no")) (4) TAD data
33
34 Meteor <- read.csv(paste0(location,"HiASAP/Meteor/Meteor_Hourly_All.csv")) (5) Meteorological data
35 Meteor <- Meteor[,c(1:4,14)]
36 PMA11_4 <- merge(PMA11_3,Meteor, by=c("Year","Month","Day","Month","Hour"))
```

To merge data by ID of subjects and date

```

38 ## To create a subject-day variable for autocorrelation adjustment
39 library(lubridate)
40 Time_1 <- ymd(paste0(PMa11_4$Year, '-', PMa11_4$Month, '-', PMa11_4$Day))
41
42 S_no_Day <- c()
43 S_no_Day <- paste0(PMa11_4$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


```

45 ## Log-transformed HRV
46 lg_HRsum5 <- log10(PMa11_4$HRsum5)
47 lg_HRmean5 <- log10(PMa11_4$HRmean5)
48 lg_SDNN5 <- log10(PMa11_4$SDNN5)
49 lg_RMSSD5 <- log10(PMa11_4$RMSSD5)
50 lg_LFHF5 <- log10(PMa11_4$LFHF5)
51 lg_LF5 <- log10(PMa11_4$LF5)
52 lg_HF5 <- log10(PMa11_4$HF5)
53 lg_VLF5 <- log10(PMa11_4$VLF5)
54 lg_TP5 <- log10(PMa11_4$TP5)
55

```

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

```

56 # To create the activity indexes
57 Activitymean <- (PMa11_4$MeanX5^2+PMa11_4$MeanY5^2+PMa11_4$MeanZ5^2)^0.5
58 Activitymax <- (PMa11_4$MaxX5^2+PMa11_4$MaxY5^2+PMa11_4$MaxZ5^2)^0.5

```

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

```

60 # To create the variable of the time of day
61 Time<-PMa11_4$Hour*12+PMa11_4$Minute/5+1

```

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

```

63 Age_G<-c()
64 for(i in 1:dim(PMall_4)[1]){
65     ifelse(PMall_4$Age[i]<60, Age_G[i]<-0, Age_G[i]<-1)
66 }

```

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

```

67
68 BMI_G<-c()
69 for(i in 1:dim(PMall_4)[1]){
70     if(PMall_4$BMI[i]>=24){
71         BMI_G[i]<-1
72     }else{
73         BMI_G[i]<-0
74     }
75 }

```

11. To group BMI into <=24 (normal-weight) and > 24 (overweight and obese) kg/m²

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

```

77 PMfinal <- data.frame(PMall_4,1g_HRsum5,1g_HRmean5,1g_SDNN5,1g_RMSSD5,1g_LFHF5,1g_LF5,1g_HF5,1g_VLF5,
78 1g_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

```
80 # select no-raining and awake period
81 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

```
83 # To run the GAMM for each HRV indices
84 lg_SDNN<-gamm(lg_SDNN5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,
85 lg_LFHF<-gamm(lg_LFHF5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,
86 lg_HRsum<-gamm(lg_HRsum5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Tim
87 lg_HRmean<-gamm(lg_HRmean5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(T
88 lg_RMSSD<-gamm(lg_RMSSD5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Tim
89 lg_LF<-gamm(lg_LF5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,bs=c
90 lg_HF<-gamm(lg_HF5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,bs=c
91 lg_VLF<-gamm(lg_VLF5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,bs
92 lg_TP<-gamm(lg_TP5~PM2.5+Loc_Out+Season+Age_G+BMI_G+s(Activitymean,bs=c("tp"))+Gender+TEM+s(Time,bs=c
```

GAMM

$s(\text{Time}, \text{bs}=\text{c}(\text{"cc"}))$
 $s(\text{R_Activitymean}, \text{bs}=\text{c}(\text{"tp"}))$

Smoothing terms

①

②

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

$$Y = x_1 + x_2 + \dots + x_i$$

```
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))
```

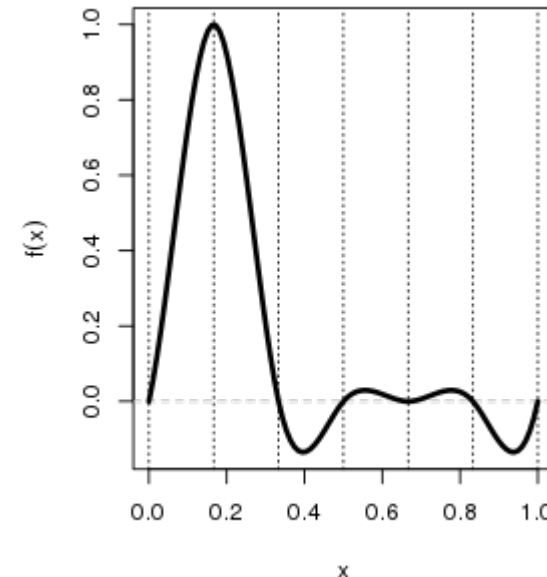
Dataset

Random effects

Autocorrelations

Smoothing 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
 - 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)

```
94 # Directly show results in the console window
95 summary(lg_SDNN$gam)
96 summary(lg_LFHF$gam)
97 summary(lg_HRsum$gam)
98 summary(lg_HRmean$gam)
99 summary(lg_RMSSD$gam)
100 summary(lg_LF$gam)
101 summary(lg_HF$gam)
102 summary(lg_VLF$gam)
103 summary(lg_TP$gam)
```

→ To print out the results in the Console window

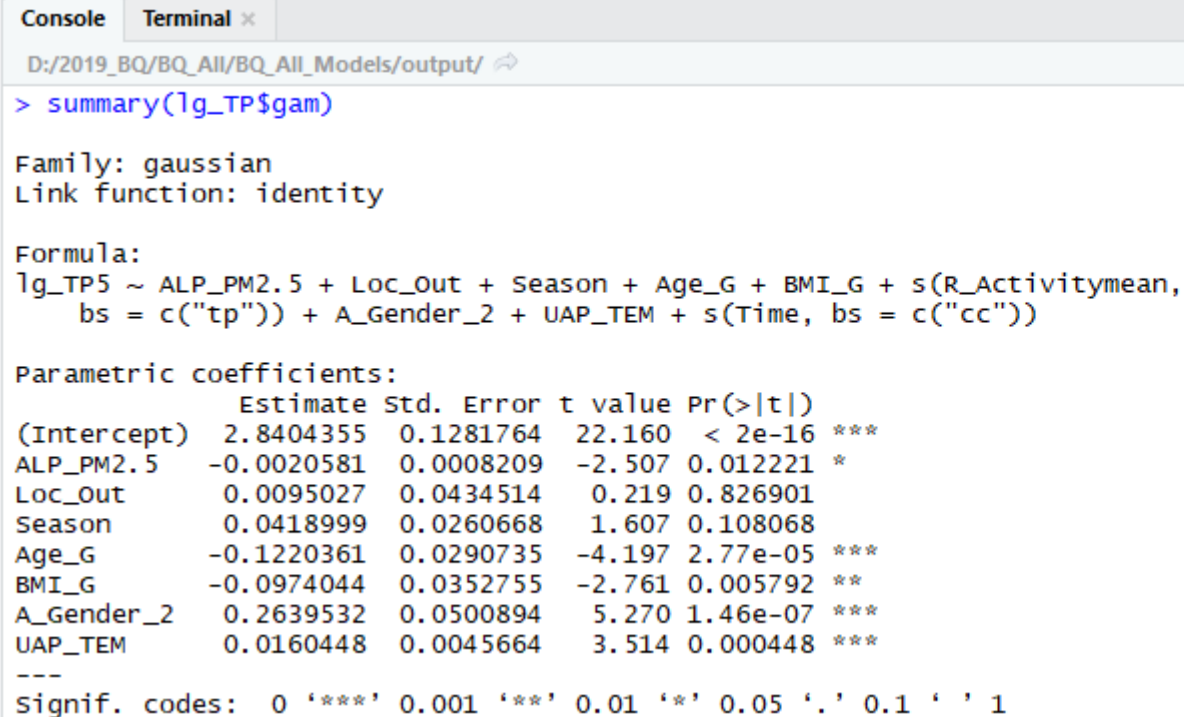
```
104
105 # To print out GAMM results to txt file
106 sink("GAMM_Results.txt") # redirect console output to a file
107 print(summary(lg_SDNN$gam))
108 print(summary(lg_LFHF$gam))
109 print(summary(lg_HRsum$gam))
110 print(summary(lg_HRmean$gam))
111 print(summary(lg_RMSSD$gam))
112 print(summary(lg_LF$gam))
113 print(summary(lg_HF$gam))
114 print(summary(lg_VLF$gam))
115 print(summary(lg_TP$gam))
116 sink() # close connection to file
```

→ To export the results to the txt file

GAMM results

```
94 # Directly show results in the Console window
95 summary(lg_SDNN$gam)
96 summary(lg_LFHF$gam)
97 summary(lg_HRsum$gam)
98 summary(lg_HRmean$gam)
99 summary(lg_RMSSD$gam)
100 summary(lg_LF$gam)
101 summ
102 summ
103 summ
```

→ To print out the results in the Console window



```
> 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
```

```

105 # To print out GAMM results to txt file
106 sink("GAMM_Results.txt") # redirect console output to a file
107 print(summary(lg_SDNN$gam))
108 print(summary(lg_LFHF$gam))
109 print(summary(lg_HRsum$gam))
110 print(summary(lg_HRmean$gam))
111 print(summary(lg_RMSSD$gam))
112 print(summary(lg_LF$gam))
113 print(summary(lg_HF$gam))
114 print(summary(lg_VLF$gam))
115 print(summary(lg_TP$gam))
116 sink() # close connection

```

→ 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 smoothing terms

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

④ Adjusted R² 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	$10^{(\beta * IQR)} - 1 * 100$	$((10^{((B + 1.96 * SE) * IQR)} - 1)) * 100$	$((10^{((B - 1.96 * SE) * IQR)} - 1)) * 100$
2	11.4	-0.00224	0.000114	-5.73	-5.18	-6.28

IQR = 75th percentile – 25th percentile

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

	PM _{2.5} (µg/m ³)		
	Percentage change ^a	95% CI	<i>p</i> -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/m³) was associated with a change of -5.73% SDNN.

GAMM results – ③ Results of smoothing 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

R-sq.(adj) = 0.539

Scale est. = 0.028199

n = 3084

Sample size

Thank you for your attention