~~~
---
title: "ExposureDA"
output: html_document
---

# read-in DailyPM data

```{r}
DailyPM_Date = readRDS("/Volumes/My Passport for Mac/WD passport/Columbia-Ghana Project/MicroPEM_Data/DailyPM.rds")
```

# read-in McirоPEM log data

```{r}
require(readstata13)
require(stringr)
require(lubridate)
MicroPEM = read.dta13("/Volumes/My Passport for Mac/WD passport/Columbia-Ghana
Project/Data/Survey_Data/MicroPem.dta")
#add leading 0 to time variables if hour is a single digit (e.g. 825 to 0825)
MicroPEM$labsetdtt = str_pad(MicroPEM$labsetdtt, 4, pad = "0")
MicroPEM$fieldsetdt = str_pad(MicroPEM$fieldsetdt, 4, pad = "0")
MicroPEM$thepaon1 = str_pad(MicroPEM$thepaon1, 4, pad = "0")
MicroPEM$pickupdtt = str_pad(MicroPEM$pickupdtt, 4, pad = "0")
MicroPEM$thepaon2 = str_pad(MicroPEM$thepaon2, 4, pad = "0")
MicroPEM$thepaoff2 = str_pad(MicroPEM$thepaoff2, 4, pad = "0")
MicroPEM$tupemoff = str_pad(MicroPEM$tupemoff, 4, pad = "0")

#assign NA to HEPA end times if Micorpem was not running when retrieving the instrument
MicroPEM$thepaon2[MicroPEM$thepaon2=="0000"|MicroPEM$thepaon2=="9999"]=NA
MicroPEM$thepaoff2[MicroPEM$thepaoff2=="0000"|MicroPEM$thepaoff2=="9999"]=NA

#HEPA start Datetime
MicroPEM$HEPA1St = paste(MicroPEM$datevisit, MicroPEM$thepaon1)
MicroPEM$HEPA1St = dmy_hm(as.character(MicroPEM$HEPA1St), tz="GMT")
range(MicroPEM$HEPA1St)

#HEPA end Datetime
MicroPEM$HEPA2St = paste(MicroPEM$pickupdtd, MicroPEM$thepaon2)
MicroPEM$HEPA2St = dmy_hm(as.character(MicroPEM$HEPA2St), tz="GMT")
range(MicroPEM$HEPA2St, na.rm=T)

MicroPEM$HEPA2End = paste(MicroPEM$pickupdtd, MicroPEM$thepaoff2)
MicroPEM$HEPA2End = dmy_hm(as.character(MicroPEM$HEPA2End), tz="GMT")
range(MicroPEM$HEPA2End, na.rm=T)

#correct a filterid typo
MicroPEM$filterid[which(is.na(MicroPEM$mstudyid))]
which(MicroPEM$filterid=="KHC031B")
MicroPEM$filterid[MicroPEM$filterid=="KHC031B"] = "KHCD31B"
```

# Merge DailyPM with MicroPEM log data

```{r}
DailyPM = merge(DailyPM_Date, MicroPEM, by.x="filterID", by.y="filterid", all.x=T)
DailyPM[which(is.na(DailyPM$mstudyid)),]
```

missing three MircoPEM logsheet  KHC0729, KHC1015, KHCD48C

# read-in DailyCO data

```{r}
COdata =
readRDS("/Users/zhengzhou/Dropbox/Ghana_exposure_data_SHARED_2014/CO_files_processed/FINAL_CO_parameters_withvalidation

COdata$Startdate = as.Date(COdata$firstdate)                                #get the start date of CO measurements
COdata1 = COdata[is.na(COdata$cstudyid),]                                   #exclude child CO measurements
```

# Merge PM and CO data
```{r}
PMCO = merge(DailyPM, COdata1, by=c("mstudyid", "Startdate"), all.x=T)
```

# Wide to long format

```{r}
# change co colnames
colnames(PMCO)[colnames(PMCO)=="co_day1_mean"] = "OldCO_1"
~~~

~~~
colnames(PMCO)[colnames(PMCO)=="co_day2_mean"] = "OldCO_2"
colnames(PMCO)[colnames(PMCO)=="co_day3_mean"] = "OldCO_3"
colnames(PMCO)[colnames(PMCO)=="co_day1_mean_corr"] = "CorCO_1"
colnames(PMCO)[colnames(PMCO)=="co_day2_mean_corr"] = "CorCO_2"
colnames(PMCO)[colnames(PMCO)=="co_day3_mean_corr"] = "CorCO_3"

PMCO1 <-reshape(PMCO,
                     varying=c(grep("PMday_",colnames(PMCO)),
                               grep("OldPM_",colnames(PMCO)),
                               grep("CorPM_",colnames(PMCO)),
                               grep("compliance_",colnames(PMCO)),
                               grep("complianceWake_",colnames(PMCO)),
                               grep("PMn_",colnames(PMCO)),
                               grep("OldCO_",colnames(PMCO)),
                               grep("CorCO_",colnames(PMCO)),
                               grep("Day_",colnames(PMCO))),
                     idvar="id",
                     direction="long",  sep="_")
```

# Data cleaning of the merged PMCO data

```{r}
PMCO2 = PMCO1[PMCO1$visually_valid!=3,]    # exclude samples with invalid CO readings
PMCO2 = PMCO2[PMCO2$PMn>1320,]      #exclude PM sample-day < 22hrs
PMCO2 = PMCO2[!is.na(PMCO2$CorCO),]  #exclude CO sample-day < 24hrs
summary(PMCO2$CorPM)       # check the range of PM
summary(PMCO2$CorCO)       # check the range of CO
PMCO2 = PMCO2[PMCO2$CorPM>0,]       #exclude PM <0
#calculate compliance measure and categorize the measure into 7 buckets
PMCO2$complianceWakePct = PMCO2$complianceWake/PMCO2$PMn
PMCO2$complianceWakePctGP = cut(PMCO2$complianceWakePct, seq(0, 0.7, 0.1), labels=c(0.1, 0.2, 0.3, 0.4, 0.5, 0.6,
0.7), right=FALSE)
```

# Plot PM vs CO

```{r}
require(ggplot2)
par(mar=c(4,4,1,1))
par(las=3)
ggplot(PMCO2, aes(CorCO, CorPM)) + geom_point() + scale_x_log10() + scale_y_log10()
```

# Plot PM vs CO by compliance bin

```{r, echo=FALSE}
require(ggplot2)
par(mar=c(4,4,1,1))
par(las=3)
ggplot(PMCO2, aes(CorCO, CorPM)) + geom_point() + facet_grid(complianceWakePctGP ~ .) + scale_x_log10() +
scale_y_log10()
```

# Correlation between PM and CO by compliance bin

```{r}
cor(PMCO2$CorCO[PMCO2$complianceWakePctGP==0.1], PMCO2$CorPM[PMCO2$complianceWakePctGP==0.1])
cor(PMCO2$CorCO[PMCO2$complianceWakePctGP==0.2], PMCO2$CorPM[PMCO2$complianceWakePctGP==0.2])
cor(PMCO2$CorCO[PMCO2$complianceWakePctGP==0.3], PMCO2$CorPM[PMCO2$complianceWakePctGP==0.3])
cor(PMCO2$CorCO[PMCO2$complianceWakePctGP==0.4], PMCO2$CorPM[PMCO2$complianceWakePctGP==0.4])
cor(PMCO2$CorCO[PMCO2$complianceWakePctGP==0.5], PMCO2$CorPM[PMCO2$complianceWakePctGP==0.5])
cor(PMCO2$CorCO[PMCO2$complianceWakePctGP==0.6], PMCO2$CorPM[PMCO2$complianceWakePctGP==0.6])
cor(PMCO2$CorCO[PMCO2$complianceWakePctGP==0.7], PMCO2$CorPM[PMCO2$complianceWakePctGP==0.7])
```

# Boxplot of PM by compliance bin

```{r, echo=FALSE}
require(plyr)
myboxplot <- ddply(PMCO2,
                   .(complianceWakePctGP),
                   summarise,
                   min = min(CorPM),
                   q1 = quantile(CorPM,0.25),
                   med = median(CorPM),
                   q3 = quantile(CorPM,0.75),
                   max= max(CorPM),
                   count = length(complianceWakePctGP))

ggplot(myboxplot, aes(x = complianceWakePctGP)) + geom_boxplot(aes(lower = q1, upper = q3, middle = med, ymin = 0,
~~~

~~~
ymax = 200),
       stat = "identity") + xlab("Complicance") + ylab(expression(paste("PM level"))) +
       theme(axis.text=element_text(size=20),axis.title=element_text(size=20,face="bold")) +
       geom_text(aes(y = q3,label = count),size= 8, vjust = -5)
```

# Boxplot of CO by compliance bin

```{r, echo=FALSE}
require(plyr)
myboxplot <- ddply(PMCO2,
                    .(complianceWakePctGP),
                    summarise,
                    min = min(CorCO),
                    q1 = quantile(CorCO,0.25),
                    med = median(CorCO),
                    q3 = quantile(CorCO,0.75),
                    max= max(CorCO),
                    count = length(complianceWakePctGP))

ggplot(myboxplot, aes(x = complianceWakePctGP)) + geom_boxplot(aes(lower = q1, upper = q3, middle = med, ymin = 0,
ymax = 3),
      stat = "identity") + xlab("Complicance") + ylab(expression(paste("CO level (ppm)"))) +
      theme(axis.text=element_text(size=20),axis.title=element_text(size=20,face="bold")) +
      geom_text(aes(y = q3,label = count),size= 8, vjust = -5)
```

# Boxplot of PM by arm

```{r, echo=FALSE}
boxplot(CorPM~cluster,data=PMCO2, ylim=c(0,250), xlab="cluster", ylab="PM(ug/m3)")
```

# Boxplot of PM by arm (compliance > 40%)

```{r, echo=FALSE}
boxplot(CorPM~cluster,data=PMCO2[PMCO2$complianceWakePct>0.4,], ylim=c(0,250), xlab="cluster", ylab="PM(ug/m3)")
```

# Boxplot of CO by arm

```{r, echo=FALSE}
boxplot(CorCO~cluster,data=PMCO2, ylim=c(0,5), xlab="cluster", ylab="CO(ppm)")
```

# Boxplot of CO by arm (compliance > 40%)

```{r, echo=FALSE}
boxplot(CorCO~cluster,data=PMCO2[PMCO2$complianceWakePct>0.4,], ylim=c(0,5), xlab="cluster", ylab="CO(ppm)")
```

# read in daily exposure data

```{r}
Exposure = read.dta13("/Volumes/My Passport for Mac/WD passport/Columbia-Ghana
Project/Data/Survey_Data/Exposure.dta", generate.factors=T)
Exposure$Startdate = as.Date(dmy(Exposure$datevisit, tz="GMT")) -1     # create a varoable for the startdate of daily
exposure
which(duplicated(Exposure[c("mstudyid","Startdate")]))        # check duplicate daily exposure data
```

some duplicated daily exposure data

# Merge PMCO with daily exposure data

```{r}
PMCO3 = merge(PMCO2, Exposure, by = c("mstudyid","Startdate"), all.x= T )
```

# Plot PM by arm and cookfood

```{r, echo=FALSE}
a = aggregate(CorPM~ cluster.x + cookfood, data = PMCO3, function(x) c(m = round(mean(x), digits=1),  gm =
round(exp(mean(log(x))), digits=1),
                                                          md= round(median(x), digits=1),  min = min(x), max = max(x),
counts = length(x)))

a$cluster = c("3-stone", "Biolite", "LPG", "3-stone", "Biolite", "LPG")
a$cookfood = c("COOK", "COOK", "COOK", "NO COOK", "NO COOK", "NO COOK")
b = as.data.frame(cbind(a$cluster, a$cookfood, a$CorPM), stringsAsFactors=FALSE)
colnames(b) = c("cluster", "COOK", colnames(a$CorPM))
~~~

~~~
ggplot(b, aes(x = cluster, y=as.numeric(m), fill=factor(COOK), color=factor(COOK))) + geom_bar(stat="identity",
position=position_dodge()) + xlab("cluster") + ylab(expression(paste("PM level (", mu,"g/", m^{3},')'))) +
  theme(axis.text=element_text(size=20),axis.title=element_text(size=20,face="bold")) +
  geom_text(aes(y=as.numeric(m), ymax=as.numeric(m), label=counts), size = 8, position= position_dodge(width=0.9),
vjust=-.5, color="black")
```

# Plot PM by arm and cookfood (compliance > 40%)

```{r, echo=FALSE}
a = aggregate(CorPM~ cluster.x + cookfood, data = PMCO3[PMCO3$complianceWakePct>0.4,], function(x) c(m =
round(mean(x), digits=1),  gm =                     round(exp(mean(log(x))), digits=1), md= round(median(x),
digits=1),  min = min(x), max = max(x), counts = length(x)))

a$cluster = c("3-stone", "Biolite", "LPG", "3-stone", "Biolite", "LPG")
a$cookfood = c("COOK", "COOK", "COOK", "NO COOK", "NO COOK", "NO COOK")
b = as.data.frame(cbind(a$cluster, a$cookfood, a$CorPM), stringsAsFactors=FALSE)
colnames(b) = c("cluster", "COOK", colnames(a$CorPM))

ggplot(b, aes(x = cluster, y=as.numeric(m), fill=factor(COOK), color=factor(COOK))) + geom_bar(stat="identity",
position=position_dodge()) + xlab("cluster") + ylab(expression(paste("PM level (", mu,"g/", m^{3},')'))) +
  theme(axis.text=element_text(size=20),axis.title=element_text(size=20,face="bold")) +
  geom_text(aes(y=as.numeric(m), ymax=as.numeric(m), label=counts), size = 8, position= position_dodge(width=0.9),
vjust=-.5, color="black")
```

# Plot CO by arm and cookfood (compliance > 40%)

```{r, echo=FALSE}
a = aggregate(CorCO~ cluster.x + cookfood, data = PMCO3[PMCO3$complianceWakePct>0.4,], function(x) c(m =
round(mean(x), digits=1),  gm =                     round(exp(mean(log(x+0.01))), digits=1), md= round(median(x),
digits=1),  min = min(x), max = max(x), counts = length(x)))

a$cluster = c("3-stone", "Biolite", "LPG", "3-stone", "Biolite", "LPG")
a$cookfood = c("COOK", "COOK", "COOK", "NO COOK", "NO COOK", "NO COOK")
b = as.data.frame(cbind(a$cluster, a$cookfood, a$CorCO), stringsAsFactors=FALSE)
colnames(b) = c("cluster", "COOK", colnames(a$CorCO))

ggplot(b, aes(x = cluster, y=as.numeric(m), fill=factor(COOK), color=factor(COOK))) + geom_bar(stat="identity",
position=position_dodge()) + xlab("cluster") + ylab(expression(paste("CO level (ppm)"))) +
  theme(axis.text=element_text(size=20),axis.title=element_text(size=20,face="bold")) +
  geom_text(aes(y=as.numeric(m), ymax=as.numeric(m), label=counts), size = 8, position= position_dodge(width=0.9),
vjust=-.5, color="black")
```

# Plot PM by fuel (compliance > 40%)

```{r, echo=FALSE}
PMCO3$MIFUEL <- "Wood"
PMCO3$MIFUEL[PMCO3$mipmeal==99] <- "No Cook"
PMCO3$MIFUEL[PMCO3$mipmeal!=99 & PMCO3$miusecoal=="Yes"] <- "Coal"
PMCO3$MIFUEL[PMCO3$mipmeal!=99 & PMCO3$miusecoal!="Yes" & PMCO3$mipstove==18 ] <- "LPG"
table(PMCO3$MIFUEL)

PMCO3$EVFUEL <- "Wood"
PMCO3$EVFUEL[PMCO3$evpmeal==99] <- "No Cook"
PMCO3$EVFUEL[PMCO3$evpmeal!=99 & PMCO3$evusecoal=="Yes"] <- "Coal"
PMCO3$EVFUEL[PMCO3$evpmeal!=99 & PMCO3$evusecoal!="Yes" & PMCO3$evprstove==18 ] <- "LPG"
table(PMCO3$EVFUEL)

PMCO3$MOFUEL <- "Wood"
PMCO3$MOFUEL[PMCO3$mopmeal==99] <- "No Cook"
PMCO3$MOFUEL[PMCO3$mopmeal!=99 & PMCO3$mousecoal=="Yes"] <- "Coal"
PMCO3$MOFUEL[PMCO3$mopmeal!=99 & PMCO3$mousecoal!="Yes" & PMCO3$moprstove==18 ] <- "LPG"
table(PMCO3$MOFUEL)

PMCO3$WOOD <- as.numeric(PMCO3$MIFUEL=="Wood") + as.numeric(PMCO3$EVFUEL=="Wood") + as.numeric(PMCO3$MOFUEL=="Wood")
PMCO3$RAIN <- as.numeric(PMCO3$mirain=="Yes") + as.numeric(PMCO3$evrain=="Yes") + as.numeric(PMCO3$morain=="Yes")
table(PMCO3$WOOD)
table(PMCO3$RAIN)

PMCO4 = PMCO3[PMCO3$complianceWakePct>0.4,]
a = aggregate(CorPM~ WOOD, data = PMCO4, function(x) c(m = round(mean(x), digits=1),  gm = round(exp(mean(log(x))),
digits=1), md= round(median(x), digits=1),  min = min(x), max = max(x), counts = length(x)))
a$WOOD = c("0", "1", "2", "3")
b = as.data.frame(cbind(a$WOOD, a$CorPM), stringsAsFactors=FALSE)
colnames(b) = c("WOOD", colnames(a$CorPM))

ggplot(b, aes(x = WOOD, y=as.numeric(gm))) + geom_bar(stat="identity") + xlab("WOOD USE") + ylab(expression(paste("PM
level (", mu,"g/", m^{3},')')))  +
~~~

~~~
  theme(axis.text=element_text(size=20),axis.title=element_text(size=20,face="bold")) +
  geom_text(aes(y=as.numeric(gm), ymax=as.numeric(gm), label=counts), size = 8, position= position_dodge(width=0.9),
vjust=-.5, color="black")
```

# Plot CO by fuel (compliance > 40%)

```{r, echo=FALSE}
a = aggregate(CorCO~ WOOD, data = PMCO4, function(x) c(m = round(mean(x), digits=1),  gm =
round(exp(mean(log(x+0.01))), digits=1), md= round(median(x), digits=1),  min = min(x), max = max(x), counts =
length(x)))
a$WOOD = c("0", "1", "2", "3")
b = as.data.frame(cbind(a$WOOD, a$CorCO), stringsAsFactors=FALSE)
colnames(b) = c("WOOD", colnames(a$CorCO))
b
ggplot(b, aes(x = WOOD, y=as.numeric(gm))) + geom_bar(stat="identity") + xlab("WOOD USE") + ylab(expression(paste("CO
level")))  +
  theme(axis.text=element_text(size=20),axis.title=element_text(size=20,face="bold")) +
  geom_text(aes(y=as.numeric(gm), ymax=as.numeric(gm), label=counts), size = 8, position= position_dodge(width=0.9),
vjust=-.5, color="black")
```

# PM-CO
```{r}
reg1 = lm(log(CorPM) ~ log(CorCO+0.01) + WOOD + RAIN + complianceWakePctGP, data = PMCO3)
summary(reg1)
```
~~~