HW2Ascript.R

Frank

Mon Sep 19 14:43:10 2016

```
# ******>>>>> Dingjue Ji dj333<<<<<******
#
# I worked with * list other students here if applicable *
#
# STAT 230 Homework 2A
# Due Wednesday, September 21, in class, 1 PM
################################################################################
# Chunk of code that should clean up 2013.  Not perfect, needs some help!

filename <- "IndiaAQIraw/aqm2013.csv"

x <- read.csv(filename, as.is=TRUE, header=FALSE)
cinfo <- x[2:3,]

#lines with '24:00 AM' which are not well formatted.
ln<-grep('24:00', x[,1])
#Reformat them into the same way as others
x[ln,1]<-gsub('\\s*24:00\\s*AM\\s*', '', x[ln,1])
x[ln,2]<-'11:59 PM'


#Get rid of the summary at the end of this csv file
#Get the line number first
ln<-grep('[0-9]+/[0-9]+/[0-9]+', x[,1])


title<-sapply(1:ncol(x), function(i) paste(x[1:ln[1]-1,i], collapse = ';'))
col.date<-grep('[Dd][Aa][Tt][Ee]', title, perl = TRUE)
col.time<-grep('[Tt][Ii][Mm][Ee]', title, perl = TRUE)
places<-c("Chennai", "Delhi", "Hyderabad", "Kolkata", "Mumbai")
col.places<-sapply(places, function(x) grep(paste(x,';PM2\\.5_[^_]+;', sep=''),
                                            title, perl = TRUE))
cols<-c(col.date, col.time, col.places)

x<-x[ln,]
x <- x[,cols]
names(x) <- c("Date", "Time", "Chennai", "Delhi", "Hyderabad", "Kolkata",
              "Mumbai")
head(x)
```

```
##        Date    Time Chennai Delhi Hyderabad Kolkata Mumbai
## 5  1/1/2013 1:00 AM  NoData 324.4    NoData  NoData NoData
## 6  1/1/2013 2:00 AM  NoData 366.8    NoData  NoData NoData
## 7  1/1/2013 3:00 AM  NoData 290.7    NoData  NoData NoData
## 8  1/1/2013 4:00 AM  NoData 245.4    NoData  NoData NoData
## 9  1/1/2013 5:00 AM  NoData 220.3    NoData  NoData NoData
## 10 1/1/2013 6:00 AM  NoData 180.2    NoData  NoData NoData
```

```
str(x)
```

```
## 'data.frame':    8760 obs. of  7 variables:
##  $ Date     : chr  "1/1/2013" "1/1/2013" "1/1/2013" "1/1/2013" ...
##  $ Time     : chr  "1:00 AM" "2:00 AM" "3:00 AM" "4:00 AM" ...
##  $ Chennai  : chr  "NoData" "NoData" "NoData" "NoData" ...
##  $ Delhi    : chr  "324.4" "366.8" "290.7" "245.4" ...
##  $ Hyderabad: chr  "NoData" "NoData" "NoData" "NoData" ...
##  $ Kolkata  : chr  "NoData" "NoData" "NoData" "NoData" ...
##  $ Mumbai   : chr  "NoData" "NoData" "NoData" "NoData" ...
```

```
for (i in 3:7) {
  x[,i] <- suppressWarnings(as.numeric(x[,i]))
  x[which(x[,i] < 0), i] <- 0
}

# Now save the processed/cleaned file:
write.table(x, "IndiaPM25/India_PM25_2013.csv", sep=",",
            row.names=FALSE, col.names=TRUE)

# End block of code for 2013: Did you find and fix the problems?
#####################################################################

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

filename <- "IndiaAQIraw/aqm2014.csv"

x <- read.csv(filename, as.is=TRUE, header=FALSE)
cinfo <- x[2:3,]

#lines with '24:00 AM' which are not well formatted.
ln<-grep('24:00', x[,1])
#Reformat them into the same way as others
x[ln,1]<-gsub('\\s*24:00\\s*AM\\s*', '', x[ln,1])
x[ln,2]<-'11:59 PM'


#Get rid of the summary at the end of this csv file
#Get the line number first
ln<-grep('[0-9]+/[0-9]+/[0-9]+', x[,1])


title<-sapply(1:ncol(x), function(i) paste(x[1:ln[1]-1,i], collapse = ';'))
col.date<-grep('[Dd][Aa][Tt][Ee]', title, perl = TRUE)
col.time<-grep('[Tt][Ii][Mm][Ee]', title, perl = TRUE)
places<-c("Chennai", "Delhi", "Hyderabad", "Kolkata", "Mumbai")
col.places<-sapply(places, function(x) grep(paste(x,';PM2\\.5_[^_]+;', sep=''),
                                            title, perl = TRUE))
cols<-c(col.date, col.time, col.places)

x<-x[ln,]
x <- x[,cols]
names(x) <- c("Date", "Time", "Chennai", "Delhi", "Hyderabad", "Kolkata",
              "Mumbai")
```

```
#Get rid of weird 'zero' in this dataset
for(i in 3:7){
  x[,i]<-gsub('zero', '0', x[,i])
}
head(x)
```

```
##        Date    Time Chennai Delhi Hyderabad Kolkata Mumbai
## 6  1/1/2014 1:00 AM      53   235        78   451.4  119.7
## 7  1/1/2014 2:00 AM  NoData   228    NoData   365.6  149.5
## 8  1/1/2014 3:00 AM      43   260    NoData   337.6  183.4
## 9  1/1/2014 4:00 AM      32   268    NoData   346.2    191
## 10 1/1/2014 5:00 AM      39   234    NoData   308.8  170.3
## 11 1/1/2014 6:00 AM  NoData   220    NoData   258.7  171.2
```

```
str(x)
```

```
## 'data.frame':    8736 obs. of  7 variables:
##  $ Date     : chr  "1/1/2014" "1/1/2014" "1/1/2014" "1/1/2014" ...
##  $ Time     : chr  "1:00 AM" "2:00 AM" "3:00 AM" "4:00 AM" ...
##  $ Chennai  : chr  "53" "NoData" "43" "32" ...
##  $ Delhi    : chr  "235" "228" "260" "268" ...
##  $ Hyderabad: chr  "78" "NoData" "NoData" "NoData" ...
##  $ Kolkata  : chr  "451.4" "365.6" "337.6" "346.2" ...
##  $ Mumbai   : chr  "119.7" "149.5" "183.4" "191" ...
```

```
for (i in 3:7) {
  x[,i] <- suppressWarnings(as.numeric(x[,i]))
  x[which(x[,i] < 0), i] <- 0
}

# Now save the processed/cleaned file:
write.table(x, "IndiaPM25/India_PM25_2014.csv", sep=",",
            row.names=FALSE, col.names=TRUE)
# End block of code for 2014: Did you find and fix the problems?
######################################################################

#############################################################################
# Chunk of code that should clean up 2015.  Ditto to the above comments.

filename <- "IndiaAQIraw/aqm_jan-nov2015.csv"

x <- read.csv(filename, as.is=TRUE, header=FALSE)
cinfo <- x[2:3,]

#lines with '24:00 AM' which are not well formatted.
ln<-grep('24:00', x[,1])
#Reformat them into the same way as others
x[ln,1]<-gsub('\\s*24:00\\s*AM\\s*', '', x[ln,1])
x[ln,2]<-'11:59 PM'


#Get rid of the summary at the end of this csv file
#Get the line number first
ln<-grep('[0-9]+/[0-9]+/[0-9]+', x[,1])
```

```
title<-sapply(1:ncol(x), function(i) paste(x[1:ln[1]-1,i], collapse = ';'))
col.date<-grep('[Dd][Aa][Tt][Ee]', title, perl = TRUE)
col.time<-grep('[Tt][Ii][Mm][Ee]', title, perl = TRUE)
places<-c("Chennai", "Delhi", "Hyderabad", "Kolkata", "Mumbai")
col.places<-sapply(places, function(x) grep(paste(x,';PM2\\.5_[^_]+;', sep=''),
                                            title, perl = TRUE))
cols<-c(col.date, col.time, col.places)

x<-x[ln,]
x <- x[,cols]
names(x) <- c("Date", "Time", "Chennai", "Delhi", "Hyderabad", "Kolkata",
              "Mumbai")
head(x)
```

```
##        Date    Time Chennai Delhi Hyderabad Kolkata Mumbai
## 6  1/1/2015 1:00 AM      75   ---       ---   199.2   75.7
## 7  1/1/2015 2:00 AM     113   ---       ---   219.4  107.4
## 8  1/1/2015 3:00 AM      93   ---       ---   196.5  134.7
## 9  1/1/2015 4:00 AM      71   ---       ---   220.5  128.4
## 10 1/1/2015 5:00 AM      59   ---       ---   193.8   99.2
## 11 1/1/2015 6:00 AM      56   ---       ---   187.4  103.8
```

```
str(x)
```

```
## 'data.frame':    8016 obs. of  7 variables:
##  $ Date     : chr  "1/1/2015" "1/1/2015" "1/1/2015" "1/1/2015" ...
##  $ Time     : chr  "1:00 AM" "2:00 AM" "3:00 AM" "4:00 AM" ...
##  $ Chennai  : chr  "75" "113" "93" "71" ...
##  $ Delhi    : chr  "---" "---" "---" "---" ...
##  $ Hyderabad: chr  "---" "---" "---" "---" ...
##  $ Kolkata  : chr  "199.2" "219.4" "196.5" "220.5" ...
##  $ Mumbai   : chr  "75.7" "107.4" "134.7" "128.4" ...
```

```
for (i in 3:7) {
  x[,i] <- suppressWarnings(as.numeric(x[,i]))
  x[which(x[,i] < 0), i] <- 0
}

# Now save the processed/cleaned file:
write.table(x, "IndiaPM25/India_PM25_2015.csv", sep=",",
            row.names=FALSE, col.names=TRUE)
# End block of code for 2015: Did you find and fix the problems?
######################################################################

z <- NULL
for (year in 2013:2015) {
  filename <- paste("IndiaPM25/India_PM25_", year, ".csv", sep="")
  cat("Reading", filename, "-------------\n")
  x <- read.csv(filename, as.is=TRUE)
  print(dim(x))
  print(summary(x$Delhi))
  z <- rbind(z, x)
}
```

```
## Reading IndiaPM25/India_PM25_2013.csv -------------
```

```
## [1] 8760    7
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    45.0    80.0   115.6   161.0   889.0     723
## Reading IndiaPM25/India_PM25_2014.csv -------------
## [1] 8736    7
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0      48      87     131     179     981     759
## Reading IndiaPM25/India_PM25_2015.csv -------------
## [1] 8016    7
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    35.0    65.0    95.2   128.0   976.0     333
```

```
dim(z)
```

```
## [1] 25512     7
```

```
z <- z[,-2]      # We're going to drop the time variable and just use date
z$Date <- as.Date(z$Date, format="%e/%m/%Y")

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

table(is.na(z$Date))
```

```
##
## FALSE
## 25512
```

```
summary(z$Delhi)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    43.0    77.0   114.2   156.0   981.0    1815
```

```
summary(z$Hyderabad)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   30.00   43.30   53.14   61.90  985.00    8114
```

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

### CODE FOR PLOTS HERE:

summary(z[,-1])
```

```
##     Chennai           Delhi         Hyderabad         Kolkata
##  Min.   :  0.00   Min.   :  0.0   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 22.00   1st Qu.: 43.0   1st Qu.: 30.00   1st Qu.: 26.00
##  Median : 32.00   Median : 77.0   Median : 43.30   Median : 50.00
##  Mean   : 36.75   Mean   :114.2   Mean   : 53.14   Mean   : 77.99
##  3rd Qu.: 45.00   3rd Qu.:156.0   3rd Qu.: 61.90   3rd Qu.:106.20
##  Max.   :493.00   Max.   :981.0   Max.   :985.00   Max.   :997.00
##  NA's   :8900     NA's   :1815    NA's   :8114     NA's   :7688
##      Mumbai
##  Min.   :  0.00
##  1st Qu.: 23.90
##  Median : 40.20
##  Mean   : 54.81
##  3rd Qu.: 74.50
##  Max.   :966.00
```

```
##  NA's   :7991
```

```
par(mfrow=c(1,1))
par(mgp = c(2, 1, 0), font.axis=2)
boxplot(log(z[,-1]), use.col=TRUE,
        xlab = 'Cities', ylab = 'log PM2.5',
        cex.lab=1.2, cex.axis=1)
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 1 is not drawn
```

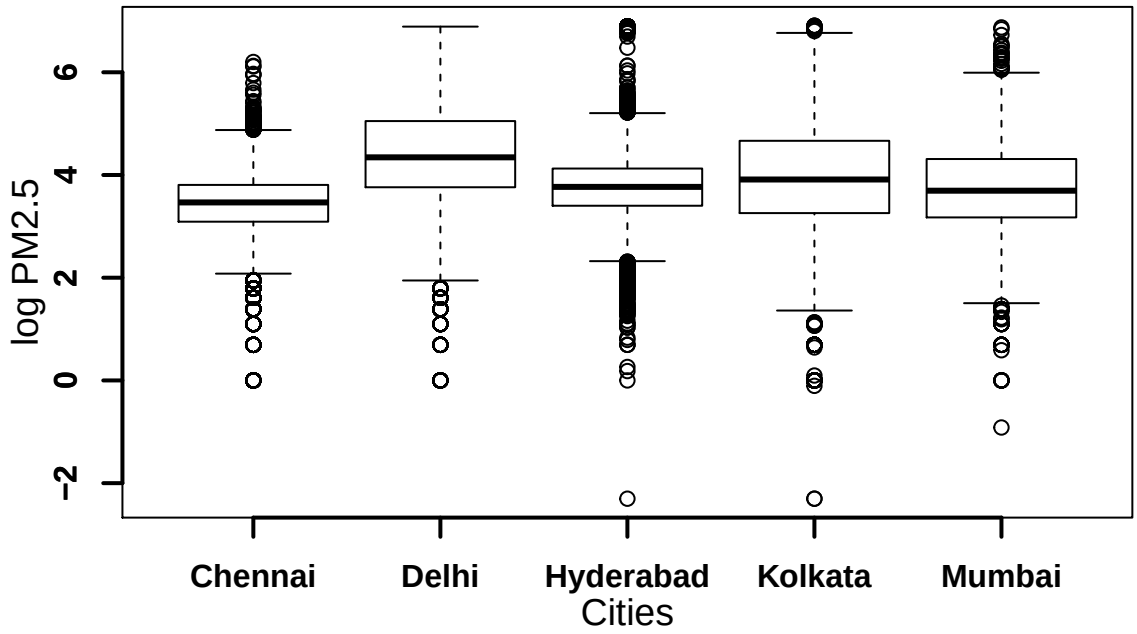
```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 2 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 3 is not drawn
```

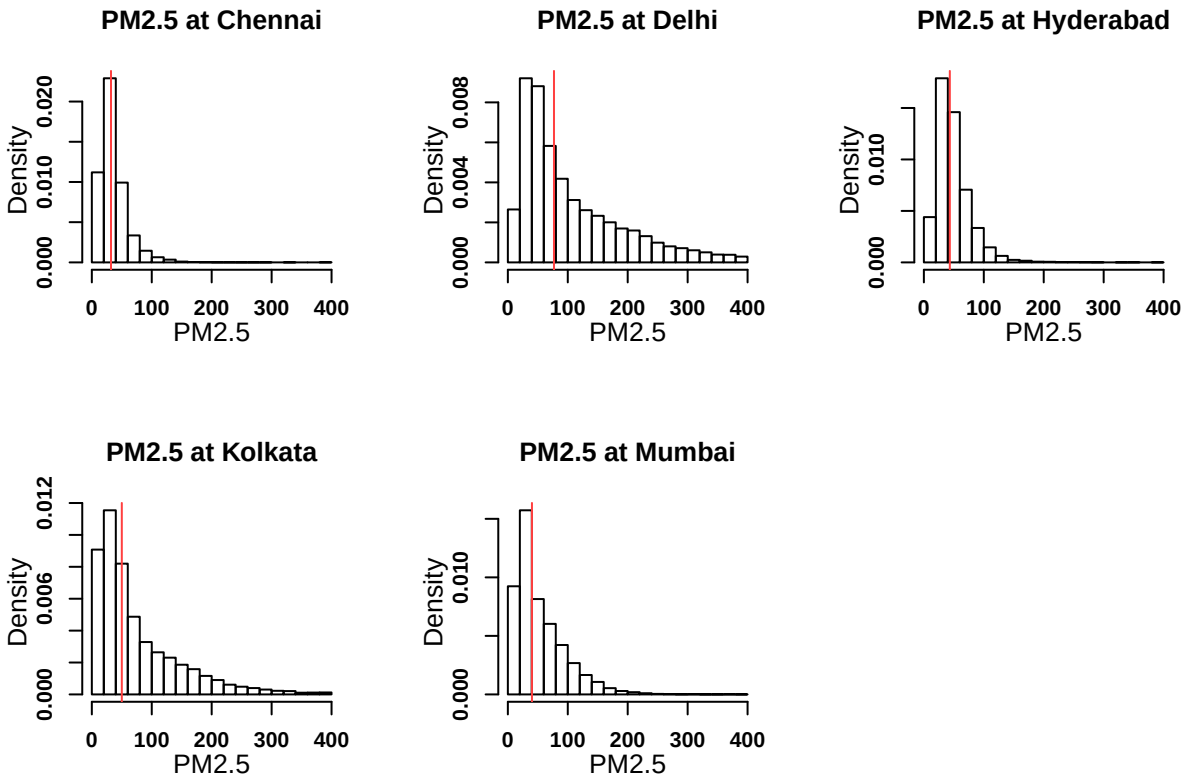
```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 4 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 5 is not drawn
```

```
axis(1, lwd=2, at = seq(1,5), labels=FALSE)
axis(2, lwd=2)
```

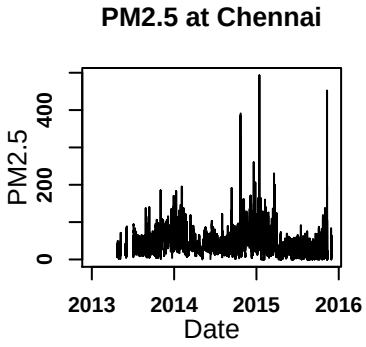


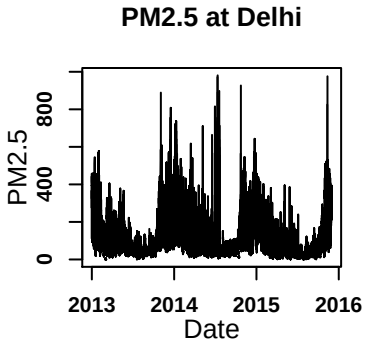

```
par(mfrow=c(2,3))
for(i in 2:6){
  hist(z[z[,i]<400,i], probability = TRUE,
       xlab='PM2.5', ylab='Density',
       cex.lab=1.2, cex.axis=1, main = paste(
         'PM2.5 at ', colnames(z)[i], sep=''
       ))
  abline(v=median(z[,i], na.rm = TRUE), col='brown1')
}

par(mfrow=c(2,3))
```

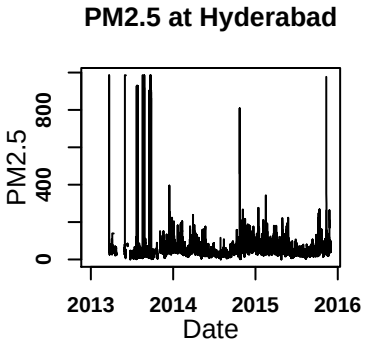


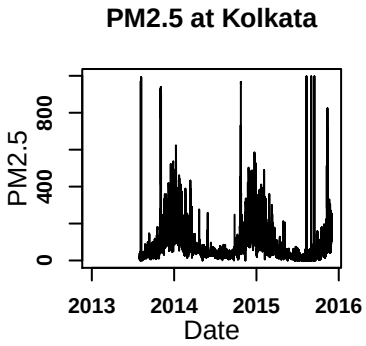

```
for(i in 2:6){
  plot(z[,i]~z$Date,
       xlab='Date', ylab='PM2.5', type='l',
       cex.lab=1.2, cex.axis=1, main = paste(
         'PM2.5 at ', colnames(z)[i], sep=''
       ))
}



### END YOUR CODE FOR PLOTS
###
### (1) Which cities appear to have the strongest and weakest seasonal
###     fluctuations?  Give at least one city in the STRONGER and WEAKER
###     category, but list all five below.  That is, you can't call all of them
###     STRONGER, for example!
###
###     STRONGER: Delhi Kolkata
###     WEAKER: Chennal Hyderabad Mumbai
###
### (2) Ask a question about one thing that puzzles you about R thus far.
###     Feel free to provide a specific code example, though realize you
###     may have to present it "commented out" so it doesn't run when you
###     do "compile notebook".  If you don't want to ask a question,
###     just say, "I'm fine with R so far."
###
###     I'm totally cool with R so far.
###
###
```

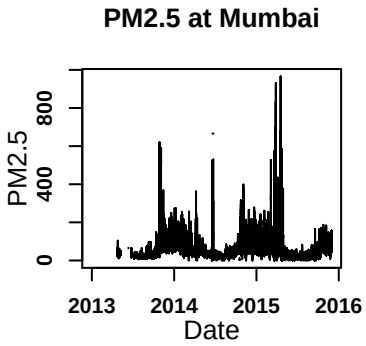
```
################################################################################
```


PM2.5 at Chennai
PM2.5
0
200
400
2013
2014
2015
2016
Date


PM2.5 at Delhi
PM2.5
0
400
800
2013
2014
2015
2016
Date


PM2.5 at Hyderabad
PM2.5
0
400
800
2013
2014
2015
2016
Date


PM2.5 at Kolkata
PM2.5
0
400
800
2013
2014
2015
2016
Date


PM2.5 at Mumbai
PM2.5
0
400
800
2013
2014
2015
2016
Date