

PM25.R

tymon

Sat Feb 25 19:28:36 2017

```
# Load packages
library(curl)
library(ggplot2)
library(xlsx)

## Loading required package: rJava
## Loading required package: xlsxjars
library(XLConnect)

## Loading required package: XLConnectJars
## XLConnect 0.2-12 by Mirai Solutions GmbH [aut],
##   Martin Studer [cre],
##   The Apache Software Foundation [ctb, cph] (Apache POI, Apache Commons
##     Codec),
##   Stephen Colebourne [ctb, cph] (Joda-Time Java library),
##   Graph Builder [ctb, cph] (Curvesapi Java library)
## http://www.mirai-solutions.com ,
## http://miraisolutions.wordpress.com
##
## Attaching package: 'XLConnect'
## The following objects are masked from 'package:xlsx':
##
##   createFreezePane, createSheet, createSplitPane, getCellStyle,
##   getSheets, loadWorkbook, removeSheet, saveWorkbook,
##   setCellStyle, setColumnWidth, setRowHeight

# Download and extract data from GIOŚ
# http://powietrze.gios.gov.pl/pjp/archives

# unzip
#bla bla bla

# load files into data frames and data frames into a list
n <- c(2000:2015)
PM10 <- list()

for (i in n) {
  f <- paste('data/', i, '/', i, '_PM10_24g.xlsx', sep = '', collapse = NULL)
  x <- readWorksheetFromFile(f, sheet = 1, header = T, startCol = 1)
  name <- (paste('pm10_', i, sep = ''))
  PM10[[name]] <- x
}

summary(PM10)
```

```

##           Length Class      Mode
## pm10_2000    8    data.frame list
## pm10_2001   11    data.frame list
## pm10_2002   18    data.frame list
## pm10_2003   55    data.frame list
## pm10_2004  122    data.frame list
## pm10_2005  167    data.frame list
## pm10_2006  164    data.frame list
## pm10_2007  174    data.frame list
## pm10_2008  166    data.frame list
## pm10_2009  168    data.frame list
## pm10_2010  151    data.frame list
## pm10_2011  146    data.frame list
## pm10_2012  152    data.frame list
## pm10_2013  142    data.frame list
## pm10_2014  160    data.frame list
## pm10_2015  159    data.frame list

# merge tables #####
#
# Wroclaw station codes #####
#
## old:
# DsWrocWisA      Wroclaw - Wisniowa
# DsWrocBartA     Wroclaw - Bartnicza
# DsWrocGrobla    Wroclaw - Na Grobli
# DsWrocOrzech    Wroclaw - Orzechowa
# DsWrocKorza     Wroclaw - Korzeniowskiego
#
#
## new:
# DsWrocAlWisn    Wroclaw - Wisniowa
# DsWrocBartni    Wroclaw - Bartnicza
# DsWrocNaGrob    Wroclaw - Na Grobli
# DsWrocOrzech    Wroclaw - Orzechowa
# DsWrocWybCon    Wroclaw - Korzeniowskiego

## list all column names with Wroc ####
l <- c(1:length(PM10))

for (i in l) {
  colIndex <- grep("Wroc", colnames(PM10[[i]]))
  colNames <- colnames(PM10[[i]][colIndex])
  print(paste('*****Stacje dla', names(PM10[i])), sep = ' ', collapes = NULL)
  print(colIndex)
}

## [1] "*****Stacje dla pm10_2000"
## [1] 2
## [1] "*****Stacje dla pm10_2001"
## [1] 3
## [1] "*****Stacje dla pm10_2002"
## [1] 3
## [1] "*****Stacje dla pm10_2003"
## [1] 7

```

```
## [1] "*****Stacje dla pm10_2004"
## [1] 17
## [1] "*****Stacje dla pm10_2005"
## [1] 20 21 22 23 24
## [1] "*****Stacje dla pm10_2006"
## [1] 19 20 21 22
## [1] "*****Stacje dla pm10_2007"
## [1] 19 20 21 22
## [1] "*****Stacje dla pm10_2008"
## [1] 18 19
## [1] "*****Stacje dla pm10_2009"
## [1] 19
## [1] "*****Stacje dla pm10_2010"
## [1] 10 11
## [1] "*****Stacje dla pm10_2011"
## [1] 13 14
## [1] "*****Stacje dla pm10_2012"
## [1] 17 18
## [1] "*****Stacje dla pm10_2013"
## [1] 15 16
## [1] "*****Stacje dla pm10_2014"
## [1] 14 15
## [1] "*****Stacje dla pm10_2015"
## [1] 13 14
```

```
# extract columns for Wroc and store in a new list
PM10Wroc_temp <- list()
l <- c(1:length(PM10))
for (i in l) {
  colIndex <- grep("Wroc", colnames(PM10[[i]]))
  name <- names(PM10[i])
  colIndex <- append(colIndex, 1, 0)
  PM10Wroc_temp[[name]] <- PM10[[i]][colIndex]
}

summary(PM10Wroc_temp)
```

```
##           Length Class      Mode
## pm10_2000  2      data.frame list
## pm10_2001  2      data.frame list
## pm10_2002  2      data.frame list
## pm10_2003  2      data.frame list
## pm10_2004  2      data.frame list
## pm10_2005  6      data.frame list
## pm10_2006  5      data.frame list
## pm10_2007  5      data.frame list
## pm10_2008  3      data.frame list
## pm10_2009  2      data.frame list
## pm10_2010  3      data.frame list
## pm10_2011  3      data.frame list
## pm10_2012  3      data.frame list
## pm10_2013  3      data.frame list
## pm10_2014  3      data.frame list
## pm10_2015  3      data.frame list
```

```

# remove top two rows with attribute names
# head(PM10[[1]], 10)
l <- c(1:length(PM10))
for (i in l) {
  PM10Wroc_temp[[i]] <- PM10Wroc_temp[[i]][c(-1:-2), ]
}

# convert first column to date and change header ####
PM10Wroc <- PM10Wroc_temp
l <- c(1:length(PM10Wroc_temp))
for (i in l) {
  PM10Wroc[[i]][, 1] <- as.Date(PM10Wroc_temp[[i]][, 1])
  colnames(PM10Wroc[[i]])[1] <- "DataObserwacji"
}
remove(PM10Wroc_temp)

# convert measures from char to numeric and calculate mean ####
# convert
l <- c(1:length(PM10Wroc))
for (i in l) {
  nc <- c(2:ncol(PM10Wroc[[i]]))
  for (j in nc) {
    PM10Wroc[[i]][, j] <- as.numeric(PM10Wroc[[i]][, j])
  }
}

# do the means
l <- c(1:length(PM10Wroc))
for (i in l) {
  nc <- c(2:ncol(PM10Wroc[[i]]))
  if (length(nc) > 1) {
    PM10Wroc[[i]]$Srednia <- apply(PM10Wroc[[i]][, nc], 1, mean, na.rm = T)
  } else {
    PM10Wroc[[i]]$Srednia <- mean(PM10Wroc[[i]], na.rm = T)
  }
}

## Warning in mean.default(PM10Wroc[[i]], na.rm = T): argument is not numeric
## or logical: returning NA

## Warning in mean.default(PM10Wroc[[i]], na.rm = T): argument is not numeric
## or logical: returning NA

## Warning in mean.default(PM10Wroc[[i]], na.rm = T): argument is not numeric
## or logical: returning NA

## Warning in mean.default(PM10Wroc[[i]], na.rm = T): argument is not numeric
## or logical: returning NA

## Warning in mean.default(PM10Wroc[[i]], na.rm = T): argument is not numeric
## or logical: returning NA

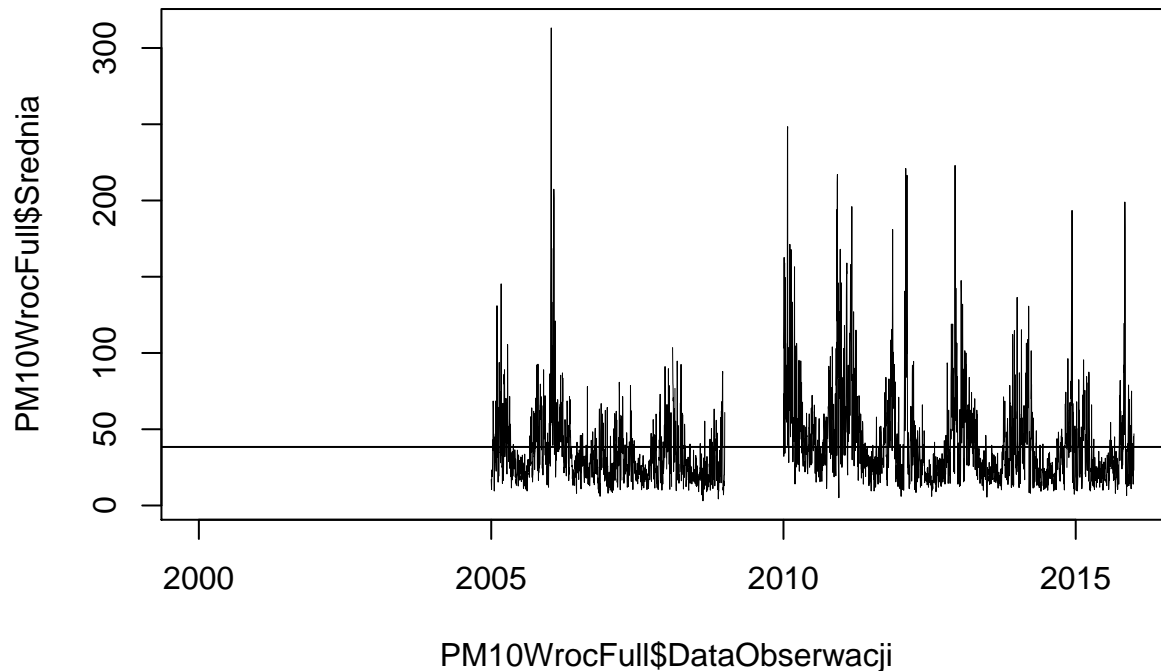
## Warning in mean.default(PM10Wroc[[i]], na.rm = T): argument is not numeric
## or logical: returning NA

```

```
## or logical: returning NA
# bind the means from all data frames
l <- c(1:length(PM10Wroc))
PM10WrocFull <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(PM10WrocFull) <- c("DataObserwacji", "Srednia")

for (i in l) {
  nc <- ncol(PM10Wroc[[i]])
  PM10WrocFull <- rbind(PM10WrocFull, PM10Wroc[[i]][, c(1,nc)])
}

# plots #####
plot(PM10WrocFull$DataObserwacji, PM10WrocFull$Srednia,
     type = "l",
     col = "black",
     lwd = 0.5)
abline(h = mean(PM10WrocFull$Srednia, na.rm = T))
ma <- filter(PM10WrocFull$Srednia, filter = rep(1/20, 20))
lines(ma, lwd = 2)
```



```
# handling missing data #####
# https://www.r-bloggers.com/imputing-missing-data-with-r-mice-package/
# https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/
```