Jiawei LiangAssignment 4: Data Wrangling

Jiawei Liang

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Wrangling

Directions

1. Rename this file `<FirstLast>_A03_DataExploration.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

The completed exercise is due on Friday, Oct7th @ 5:00pm.

Set up your session

1. Check your working directory, load the `tidyverse` and `lubridate` packages, and upload all four raw data files associated with the EPA Air dataset, being sure to set string columns to be read in a factors. See the README file for the EPA air datasets for more information (especially if you have not worked with air quality data previously).

2. Explore the dimensions, column names, and structure of the datasets.

```
#1
library(tidyverse)
library(lubridate)
library(dplyr)
getwd()
```

```
## [1] "C:/Users/Jiawei Liang/Documents/EDA-Fall2022/Assignments"
```

```
setwd('c:/Users/Jiawei Liang/Documents/EDA-Fall2022/Data/Raw/')
EPA_PM_2019 <-read.csv('EPAair_PM25_NC2019_raw.csv', stringsAsFactors =TRUE)
EPA_PM_2018 <-read.csv('EPAair_PM25_NC2018_raw.csv', stringsAsFactors =TRUE)
EPA_o3_2018 <-read.csv('EPAair_O3_NC2019_raw.csv', stringsAsFactors =TRUE)
EPA_o3_2018 <-read.csv('EPAair_O3_NC2018_raw.csv', stringsAsFactors =TRUE)
#2
dim(EPA_PM_2019)
```

```
## [1] 8581   20
```

```
dim(EPA_PM_2018)
```

```
## [1] 8983   20
```

```
dim(EPA_o3_2018)
```

```
## [1] 9737   20
```

```
dim(EPA_o3_2018)
```

```
## [1] 9737   20
```

```
str(EPA_PM_2019)
```

```
## ’data.frame':    8581 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2019","01/02/2019",..: 3 6 9 12 15 18
##  $ Source                        : Factor w/ 2 levels "AirNow","AQS": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Site.ID                       : int  370110002 370110002 370110002 370110002 370110002 370110002 3
##  $ POC                           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Mean.PM2.5.Concentration: num  1.6 1 1.3 6.3 2.6 1.2 1.5 1.5 3.7 1.6 ...
##  $ UNITS                         : Factor w/ 1 level "ug/m3 LC": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE               : int  7 4 5 26 11 5 6 6 15 7 ...
##  $ Site.Name                     : Factor w/ 25 levels "","Board Of Ed. Bldg.",..: 14 14 14 14 14 14
##  $ DAILY_OBS_COUNT               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ PERCENT_COMPLETE              : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE            : int  88502 88502 88502 88502 88502 88502 88502 88502 88502 88502
##  $ AQS_PARAMETER_DESC            : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1
##  $ CBSA_CODE                     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                     : Factor w/ 14 levels "","Asheville, NC",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ STATE_CODE                    : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                         : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                   : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                        : Factor w/ 21 levels "Avery","Buncombe",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ SITE_LATITUDE                 : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE                : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```
str(EPA_PM_2018)
```

```
## ’data.frame':    8983 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2018","01/02/2018",..: 2 5 8 11 14 17
##  $ Source                        : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                       : int  370110002 370110002 370110002 370110002 370110002 370110002 3
##  $ POC                           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Mean.PM2.5.Concentration: num  2.9 3.7 5.3 0.8 2.5 4.5 1.8 2.5 4.2 1.7 ...
##  $ UNITS                         : Factor w/ 1 level "ug/m3 LC": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE               : int  12 15 22 3 10 19 8 10 18 7 ...
##  $ Site.Name                     : Factor w/ 25 levels "","Blackstone",..: 15 15 15 15 15 15 15 15 1
##  $ DAILY_OBS_COUNT               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ PERCENT_COMPLETE              : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE            : int  88502 88502 88502 88502 88502 88502 88502 88502 88502 88502
##  $ AQS_PARAMETER_DESC          : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1
##  $ CBSA_CODE                   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                   : Factor w/ 14 levels "","Asheville, NC",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ STATE_CODE                  : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                       : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                 : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                      : Factor w/ 21 levels "Avery","Buncombe",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ SITE_LATITUDE               : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE              : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```
str(EPA_o3_2018)
```

```
## ’data.frame':    9737 obs. of  20 variables:
##  $ Date                                : Factor w/ 364 levels "01/01/2018","01/02/2018",..: 60 61 62
##  $ Source                              : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                             : int  370030005 370030005 370030005 370030005 370030005 37003
##  $ POC                                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.043 0.046 0.047 0.049 0.047 0.03 0.036 0.044 0.049 0
##  $ UNITS                               : Factor w/ 1 level "ppm": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE                     : int  40 43 44 45 44 28 33 41 45 40 ...
##  $ Site.Name                           : Factor w/ 40 levels "","Beaufort",..: 35 35 35 35 35 35 35 3
##  $ DAILY_OBS_COUNT                     : int  17 17 17 17 17 17 17 17 17 17 ...
##  $ PERCENT_COMPLETE                    : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44201 44201 44201 44201 4
##  $ AQS_PARAMETER_DESC                  : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25860 25860 25860 25860 2
##  $ CBSA_NAME                           : Factor w/ 17 levels "","Asheville, NC",..: 9 9 9 9 9 9 9 9 9
##  $ STATE_CODE                          : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                               : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                         : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                              : Factor w/ 32 levels "Alexander","Avery",..: 1 1 1 1 1 1 1
##  $ SITE_LATITUDE                       : num  35.9 35.9 35.9 35.9 35.9 ...
##  $ SITE_LONGITUDE                      : num  -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
str(EPA_o3_2018)
```

```
## ’data.frame':    9737 obs. of  20 variables:
##  $ Date                                : Factor w/ 364 levels "01/01/2018","01/02/2018",..: 60 61 62
##  $ Source                              : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                             : int  370030005 370030005 370030005 370030005 370030005 37003
##  $ POC                                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.043 0.046 0.047 0.049 0.047 0.03 0.036 0.044 0.049 0
##  $ UNITS                               : Factor w/ 1 level "ppm": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE                     : int  40 43 44 45 44 28 33 41 45 40 ...
##  $ Site.Name                           : Factor w/ 40 levels "","Beaufort",..: 35 35 35 35 35 35 35 3
##  $ DAILY_OBS_COUNT                     : int  17 17 17 17 17 17 17 17 17 17 ...
##  $ PERCENT_COMPLETE                    : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44201 44201 44201 44201 4
##  $ AQS_PARAMETER_DESC                  : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25860 25860 25860 25860 2
##  $ CBSA_NAME                           : Factor w/ 17 levels "","Asheville, NC",..: 9 9 9 9 9 9 9 9 9
##  $ STATE_CODE                          : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                               : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                         : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                              : Factor w/ 32 levels "Alexander","Avery",..: 1 1 1 1 1 1 1
##  $ SITE_LATITUDE                       : num  35.9 35.9 35.9 35.9 35.9 ...
##  $ SITE_LONGITUDE                      : num  -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
colnames(EPA_PM_2019)
```

```
##  [1] "Date"                           "Source"
##  [3] "Site.ID"                        "POC"
##  [5] "Daily.Mean.PM2.5.Concentration" "UNITS"
##  [7] "DAILY_AQI_VALUE"                "Site.Name"
##  [9] "DAILY_OBS_COUNT"                "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"             "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"                      "CBSA_NAME"
## [15] "STATE_CODE"                     "STATE"
## [17] "COUNTY_CODE"                    "COUNTY"
## [19] "SITE_LATITUDE"                  "SITE_LONGITUDE"
```

```
colnames(EPA_PM_2018)
```

```
##  [1] "Date"                           "Source"
##  [3] "Site.ID"                        "POC"
##  [5] "Daily.Mean.PM2.5.Concentration" "UNITS"
##  [7] "DAILY_AQI_VALUE"                "Site.Name"
##  [9] "DAILY_OBS_COUNT"                "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"             "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"                      "CBSA_NAME"
## [15] "STATE_CODE"                     "STATE"
## [17] "COUNTY_CODE"                    "COUNTY"
## [19] "SITE_LATITUDE"                  "SITE_LONGITUDE"
```

```
colnames(EPA_o3_2018)
```

```
##  [1] "Date"
##  [2] "Source"
##  [3] "Site.ID"
##  [4] "POC"
##  [5] "Daily.Max.8.hour.Ozone.Concentration"
##  [6] "UNITS"
##  [7] "DAILY_AQI_VALUE"
##  [8] "Site.Name"
##  [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
```

```
colnames(EPA_o3_2018)
```

```
##  [1] "Date"
##  [2] "Source"
##  [3] "Site.ID"
##  [4] "POC"
##  [5] "Daily.Max.8.hour.Ozone.Concentration"
##  [6] "UNITS"
##  [7] "DAILY_AQI_VALUE"
##  [8] "Site.Name"
##  [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
```

Wrangle individual datasets to create processed files.

3. Change date to date
4. Select the following columns: Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE
5. For the PM2.5 datasets, fill all cells in AQS_PARAMETER_DESC with "PM2.5" (all cells in this column should be identical).
6. Save all four processed datasets in the Processed folder. Use the same file names as the raw files but replace "raw" with "processed".

```
#3
class(EPA_PM_2019$Date)
```

```
## [1] "factor"
```

```
EPA_PM_2019$Date <- as.Date(EPA_PM_2019$Date, format = "%m/%d/%y")
EPA_PM_2018$Date <- as.Date(EPA_PM_2018$Date, format = "%m/%d/%y")
EPA_o3_2018$Date <- as.Date(EPA_o3_2018$Date, format = "%m/%d/%y")
EPA_o3_2018$Date <- as.Date(EPA_o3_2018$Date, format = "%m/%d/%y")
#4
EPA_PM_2019_1 <- select(EPA_PM_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_
EPA_PM_2018_1 <- select(EPA_PM_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_
EPA_o3_2018_1 <- select(EPA_o3_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_
EPA_o3_2018_1 <- select(EPA_o3_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_
#5
EPA_PM_2019$AQS_PARAMETER_DESC <-"PM2.5"
EPA_PM_2018$AQS_PARAMETER_DESC <-"PM2.5"
#6
```

```
write.csv(EPA_PM_2019, file = "c:/Users/Jiawei Liang/Documents/EDA-Fall2022/Data/Processed/EPAair_PM25_
write.csv(EPA_PM_2018, file = "c:/Users/Jiawei Liang/Documents/EDA-Fall2022/Data/Processed/EPAair_PM25_
write.csv(EPA_o3_2018, file = "c:/Users/Jiawei Liang/Documents/EDA-Fall2022/Data/Processed/EPAair_o3_NC
write.csv(EPA_o3_2018, file = "c:/Users/Jiawei Liang/Documents/EDA-Fall2022/Data/Processed/EPAair_o3_NC
```

Combine datasets

7. Combine the four datasets with `rbind`. Make sure your column names are identical prior to running this code.
8. Wrangle your new dataset with a pipe function (%>%) so that it fills the following conditions:

- Include all sites that the four data frames have in common: “Linville Falls”, “Durham Armory”, “Leggett”, “Hattie Avenue”, “Clemmons Middle”, “Mendenhall School”, “Frying Pan Mountain”, “West Johnston Co.”, “Garinger High School”, “Castle Hayne”, “Pitt Agri. Center”, “Bryson City”, “Millbrook School” (the function `intersect` can figure out common factor levels)
- Some sites have multiple measurements per day. Use the split-apply-combine strategy to generate daily means: group by date, site, aqs parameter, and county. Take the mean of the AQI value, latitude, and longitude.
- Add columns for “Month” and “Year” by parsing your “Date” column (hint: `lubridate` package)
- Hint: the dimensions of this dataset should be 14,752 x 9.

9. Spread your datasets such that AQI values for ozone and PM2.5 are in separate columns. Each location on a specific date should now occupy only one row.
10. Call up the dimensions of your new tidy dataset.
11. Save your processed dataset with the following file name: “EPAair_O3_PM25_NC1718_Processed.csv”

```
setwd('C:/Users/Jiawei Liang/Documents/WeChat Files/wxid_ob6tgcp1ldju22/FileStorage/File/2022-10')
library(tidyverse)
library(lubridate)

# read in all csv files
EPA_PM_2018 <- read.csv("EPAair_PM25_NC2018_raw.csv", header = TRUE, sep = ",")
EPA_PM_2019 <- read.csv("EPAair_PM25_NC2019_raw.csv", header = TRUE, sep = ",")
EPA_o3_2018 <- read.csv("EPAair_O3_NC2018_raw.csv", header = TRUE, sep = ",")
EPA_o3_2019 <- read.csv("EPAair_O3_NC2019_raw.csv", header = TRUE, sep = ",")

EPA_PM_2018$Date = as.Date(EPA_PM_2018$Date, format = "%m/%d/%Y")
EPA_PM_2019$Date = as.Date(EPA_PM_2019$Date, format = "%m/%d/%Y")
EPA_o3_2018$Date = as.Date(EPA_o3_2018$Date, format = "%m/%d/%Y")
EPA_o3_2019$Date = as.Date(EPA_o3_2019$Date, format = "%m/%d/%Y")

EPA_PM_2018$AQS_PARAMETER_DESC <- "PM2.5"
EPA_PM_2019$AQS_PARAMETER_DESC <- "PM2.5"

#7
colnames(EPA_PM_2019)<-colnames(EPA_PM_2018)<-colnames(EPA_o3_2019)<-colnames(EPA_o3_2018)
All_Four_data <-rbind(EPA_PM_2019,EPA_PM_2018,EPA_o3_2019,EPA_o3_2018)
#8
common_sites <- c("Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue", "Clemmons Middle", "Me

EPA_o3PM25_1819 <- All_Four_data[All_Four_data$Site.Name %in% common_sites,]%>%
group_by(Date, Site.Name, COUNTY, AQS_PARAMETER_DESC)%>%
summarise(AQI = mean(DAILY_AQI_VALUE), Latitude = mean(SITE_LATITUDE), Longitude = mean(SITE_LONGITUDE)
```

```
## ‘summarise()‘ has grouped output by ’Date’, ’Site.Name’, ’COUNTY’. You can
## override using the ‘.groups‘ argument.
```

```
EPA_o3PM25_1819$Month <- month(EPA_o3PM25_1819$Date)
EPA_o3PM25_1819$Year <- year(EPA_o3PM25_1819$Date)
print(EPA_o3PM25_1819)
```

```
## # A tibble: 14,752 x 9
## # Groups:   Date, Site.Name, COUNTY [8,976]
##    Date       Site.Name         COUNTY AQS_P~1   AQI Latit~2 Longi~3 Month  Year
##    <date>     <chr>             <chr>  <chr>   <dbl>   <dbl>   <dbl> <dbl> <dbl>
##  1 2018-01-01 Bryson City       Swain  PM2.5      35    35.4   -83.4     1  2018
##  2 2018-01-01 Castle Hayne      New H~ PM2.5      13    34.4   -77.8     1  2018
##  3 2018-01-01 Clemmons Middle   Forsy~ PM2.5      24    36.0   -80.3     1  2018
##  4 2018-01-01 Durham Armory     Durham PM2.5      31    36.0   -78.9     1  2018
##  5 2018-01-01 Garinger High Sc~ Meckl~ Ozone      32    35.2   -80.8     1  2018
##  6 2018-01-01 Garinger High Sc~ Meckl~ PM2.5      20    35.2   -80.8     1  2018
##  7 2018-01-01 Hattie Avenue     Forsy~ PM2.5      22    36.1   -80.2     1  2018
##  8 2018-01-01 Leggett           Edgec~ PM2.5      14    36.0   -77.6     1  2018
##  9 2018-01-01 Millbrook School  Wake   Ozone      34    35.9   -78.6     1  2018
## 10 2018-01-01 Millbrook School  Wake   PM2.5      28    35.9   -78.6     1  2018
## # ... with 14,742 more rows, and abbreviated variable names
## #   1: AQS_PARAMETER_DESC, 2: Latitude, 3: Longitude
```

```
#9
EPA_o3PM25_1819.Name.gathered <- gather(EPA_o3PM25_1819, "PM2.5", "Ozone", AQI)
EPA_o3PM25_1819.Name.spread <- spread(EPA_o3PM25_1819.Name.gathered, PM2.5, Ozone)
#10
dim(EPA_o3PM25_1819)
```

```
## [1] 14752     9
```

```
#11
write.csv(EPA_o3PM25_1819, file = "c:/Users/Jiawei Liang/Documents/EDA-Fall2022/Data/Processed/EPAair_O3
```

Generate summary tables

12. Use the split-apply-combine strategy to generate a summary data frame. Data should be grouped by site, month, and year. Generate the mean AQI values for ozone and PM2.5 for each group. Then, add a pipe to remove instances where a month and year are not available (use the function `drop_na` in your pipe).

13. Call up the dimensions of the summary dataset.

```
#12a
EPA_data_summary <- EPA_o3PM25_1819 %>%
  group_by(Site.Name, Month, Year) %>%
```

```r
  summarise(meanaqi_pm = mean("PM2.5"),
            meanaqi_o3 = mean("Ozone"),
            .groups = "keep")
```

```
## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("PM2.5"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA

## Warning in mean.default("Ozone"): argument is not numeric or logical: returning
## NA
```

```
print(EPA_data_summary)
```

```
## # A tibble: 308 x 5
## # Groups:   Site.Name, Month, Year [308]
##    Site.Name   Month  Year meanaqi_pm meanaqi_o3
##    <chr>       <dbl> <dbl>      <dbl>      <dbl>
##  1 Bryson City     1  2018         NA         NA
##  2 Bryson City     1  2019         NA         NA
##  3 Bryson City     2  2018         NA         NA
##  4 Bryson City     2  2019         NA         NA
##  5 Bryson City     3  2018         NA         NA
##  6 Bryson City     3  2019         NA         NA
##  7 Bryson City     4  2018         NA         NA
##  8 Bryson City     4  2019         NA         NA
##  9 Bryson City     5  2018         NA         NA
## 10 Bryson City     5  2019         NA         NA
## # ... with 298 more rows
```

```
#12b
EPA_data_summary_1 <- drop_na(EPA_data_summary)
print(EPA_data_summary_1)
```

```
## # A tibble: 0 x 5
## # Groups:   Site.Name, Month, Year [0]
## # ... with 5 variables: Site.Name <chr>, Month <dbl>, Year <dbl>,
## #   meanaqi_pm <dbl>, meanaqi_o3 <dbl>
```

```
#13
dim(EPA_data_summary_1)
```

```
## [1] 0 5
```

14. Why did we use the function `drop_na` rather than `na.omit`?

    Answer: If we use the 'na.omit', we will remove the whole row which NA is in. Thus, we would like to use drop_na.