AS2: Exploring Data via Visualization

By: Kevin Karnadi Kirmansjah 紀維鑫 - 109006241

1) Import and Preprocess Data

a) First, import the datasets using the following links 1) “https://bit.ly/3c4AHbL (https://bit.ly/3c4AHbL)” for 1999 data, and 2) “https://bit.ly/3nZicL2 (https://bit.ly/3nZicL2)” for 2012 data using the data.table package. p.s., set colClasses of the first 5 variables to “character” and the rest of it to “numeric.”

```
classes = c("character", "character", "character", "character", "character",
            "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric")
data1999 = fread("https://bit.ly/3c4AHbL", colClasses=classes)
data2012 = fread("https://bit.ly/3nZicL2", colClasses=classes)
```

b) Take a look at the 1999 data by (1) printing out the dimensions and (2) the first 3 rows.

```
print(dim(data1999))
```

```
## [1] 117421     12
```

```
head(data1999, 3)
```

```
##    X..RD Action.Code State.Code County.Code Site.ID Parameter POC
## 1:    RD           I         01         027    0001     88101   1
## 2:    RD           I         01         027    0001     88101   1
## 3:    RD           I         01         027    0001     88101   1
##    Sample.Duration Unit Method     Date Sample.Value
## 1:               7  105    120 19990103           NA
## 2:               7  105    120 19990106           NA
## 3:               7  105    120 19990109           NA
```

c) The variable of our interest is Sample.Value which contains the PM2.5 measurements. (3) Using the 1999 data, print the summary statistics of the variable with summary().

```
summary(data1999$Sample.Value)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    7.20   11.50   13.74   17.90  157.10   13217
```

i) We observe some missing values in the observations of the PM2.5 measurements (n = 13,217). Compute the number of NAs using table() and is.na(), then divide the numbers by the total number of observations in the data to calculate the proportions.

```
NA_count = table(is.na(data1999))
NA_proportion = NA_count["TRUE"] / nrow(data1999)
print(NA_count["TRUE"])
```

```
##  TRUE
## 13217
```

ii) (4) What is the percentage of the PM2.5 observations that are missing (round up to 3 decimal places)?

```
cat(round(NA_proportion*100, 3), "%", sep="")
```

```
## 11.256%
```

d) Bind the 1999 data and 2012 data and assign the aggregated data to an object called ‘pm’. Then, subset the years from the Date variable and convert it into a factor variable called ‘year’.

```
pm = rbind(data1999, data2012)
pm = pm[,':='(Year = as.factor(year(as.Date(as.character(Date), format="%Y%m%d"))))]
head(pm)
```

```
##    X..RD Action.Code State.Code County.Code Site.ID Parameter POC
## 1:    RD           I         01         027    0001     88101   1
## 2:    RD           I         01         027    0001     88101   1
## 3:    RD           I         01         027    0001     88101   1
## 4:    RD           I         01         027    0001     88101   1
## 5:    RD           I         01         027    0001     88101   1
## 6:    RD           I         01         027    0001     88101   1
##    Sample.Duration Unit Method     Date Sample.Value Year
## 1:               7  105    120 19990103           NA 1999
## 2:               7  105    120 19990106           NA 1999
## 3:               7  105    120 19990109           NA 1999
## 4:               7  105    120 19990112        8.841 1999
## 5:               7  105    120 19990115       14.920 1999
## 6:               7  105    120 19990118        3.878 1999
```

e) Next, rename the Sample.Value variable to PM which better expresses the values stored in the variable.

```
colnames(pm)[colnames(pm) == "Sample.Value"] = "PM"
head(pm)
```

```
##    X..RD Action.Code State.Code County.Code Site.ID Parameter POC
## 1:    RD           I         01         027    0001     88101   1
## 2:    RD           I         01         027    0001     88101   1
## 3:    RD           I         01         027    0001     88101   1
## 4:    RD           I         01         027    0001     88101   1
## 5:    RD           I         01         027    0001     88101   1
## 6:    RD           I         01         027    0001     88101   1
##    Sample.Duration Unit Method     Date     PM Year
## 1:               7  105    120 19990103     NA 1999
## 2:               7  105    120 19990106     NA 1999
## 3:               7  105    120 19990109     NA 1999
## 4:               7  105    120 19990112  8.841 1999
## 5:               7  105    120 19990115 14.920 1999
## 6:               7  105    120 19990118  3.878 1999
```

2) Data Exploration with Visualization using ggplot2

Aggregate data analysis:

a) First, for better visibility and reproducibility, (5) set the seed at 2021 and draw 1,000 randomly selected samples from the data (i.e., pm) using the sampling function in dplyr package.

```
set.seed(2021)
samples = sample_n(pm, 1000)
head(samples)
```

```
##    X..RD Action.Code State.Code County.Code Site.ID Parameter POC
## 1:    RD           I         11         001    0043     88101   4
## 2:    RD           I         30         075    0001     88101   3
## 3:    RD           I         10         003    2004     88101   3
## 4:    RD           I         46         103    0020     88101   3
## 5:    RD           I         30         083    0001     88101   3
## 6:    RD           I         33         009    0010     88101   3
##    Sample.Duration Unit Method     Date   PM Year
## 1:               1  105    170 20120212 8.00 2012
## 2:               1  105    170 20120305 8.60 2012
## 3:               1  105    184 20120323 8.88 2012
## 4:               1  105    170 20120312 2.70 2012
## 5:               1  105    170 20120526 3.20 2012
## 6:               1  105    170 20120305 7.70 2012
```

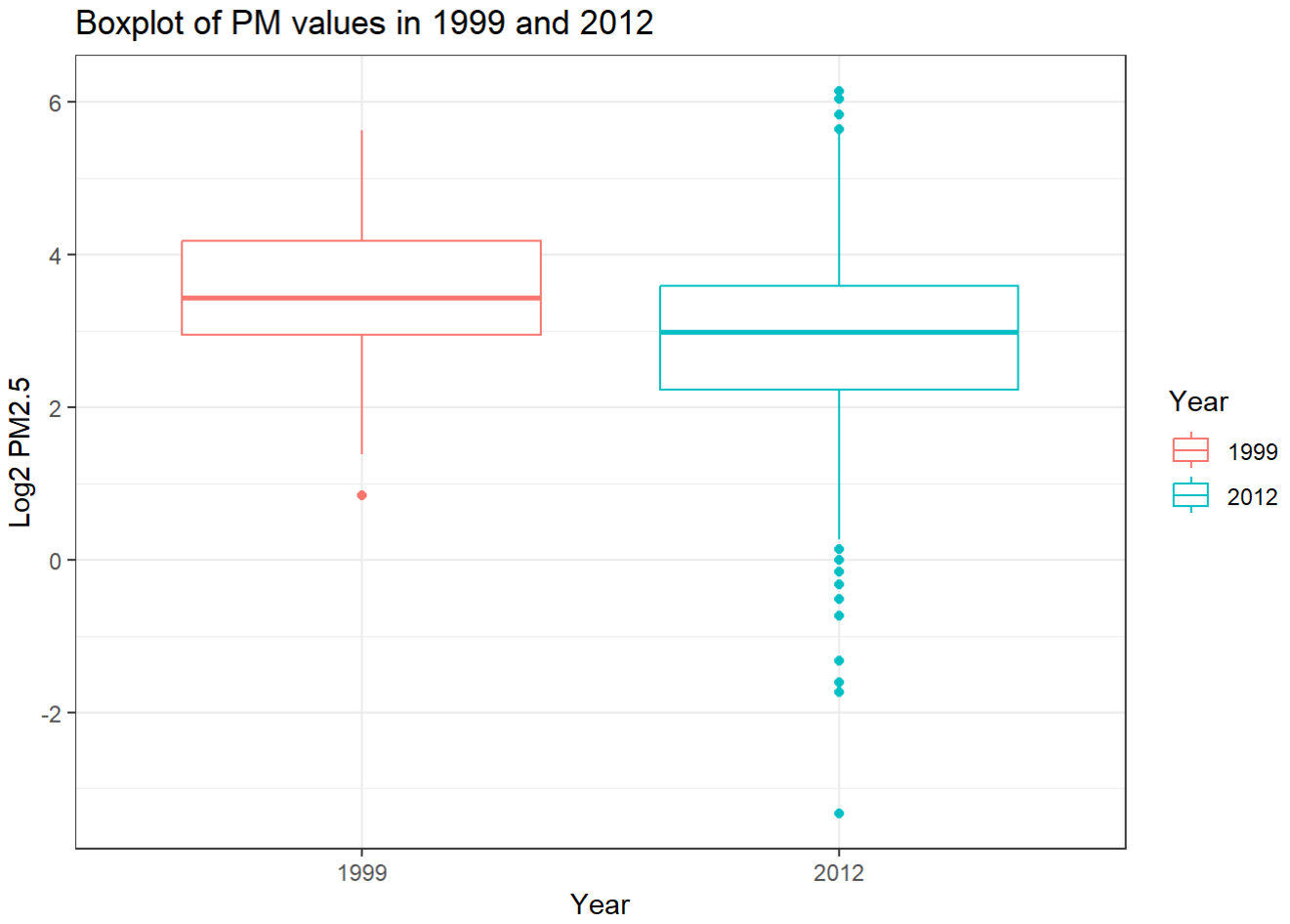
b) Then, create boxplots of all monitor values in 1999 and 2012 using the randomly sampled data as shown below. (6) Make sure to take the log of the PM values (with base 2; i.e., binary algorithm) to adjust for the skewness in the data, (7) label the title, x-axis & y-axis, and (8) use the base white theme to replicate the graphics.

```
ggplot(samples, aes(x=Year, y=log2(PM), color=Year)) +
  geom_boxplot() +
  labs(title="Boxplot of PM values in 1999 and 2012", x="Year", y="Log2 PM2.5") +
  theme_bw()
```

```
## Warning in FUN(X[[i]], ...): NaNs produced

## Warning in FUN(X[[i]], ...): NaNs produced
```

```
## Warning: Removed 89 rows containing non-finite values (`stat_boxplot()`).
```



c) (9) Describe what you observe in terms of the average and variance (in general terms—where it is centered & how much it is spread out) of the observations in 1999 and 2012?

> Ans: The average PM2.5 value in 1999 is higher than in 2012, while the variance of the PM2.5 value in 2012 is higher than in 1999.

Changes in PM levels at an individual monitor:

d) Our first task is to identify a monitor in New York State that has data in 1999 and 2012 (not all monitors operated during both time periods). (10) Subset the data to include only the observations from New York (i.e., State.Code == 36) and only include the County.Code and the Site.ID (i.e. monitor number) variables using filter(), select(), and unique().

```
data_NY = filter(pm, State.Code == 36)
data_NY = select(data_NY, County.Code, Site.ID)
data_NY = unique(data_NY)
head(data_NY)
```

```
##    County.Code Site.ID
## 1:         001    0005
## 2:         001    0012
## 3:         005    0073
## 4:         005    0080
## 5:         005    0083
## 6:         005    0110
```

e) (11) Create a new variable called Site.Code that combines the county code and the site ID into a single string by using paste() with “.” as the separator.

```
pm$`Site.Code` = paste(pm$County.Code, pm$Site.ID, sep=".")
head(pm$`Site.Code`)
```

```
## [1] "027.0001" "027.0001" "027.0001" "027.0001" "027.0001" "027.0001"
```

f) (12) Find the intersection of the sites (i.e., monitors) in between 1999 and 2012 which gives us the list of monitors in New York that operated both in 1999 and 2012 using split() and intersect().

```
pm_NY = filter(pm, State.Code == 36)
pm_split = split(pm_NY, pm_NY$Year)
monitors = intersect(pm_split$`1999`$Site.Code, pm_split$`2012`$Site.Code)
monitors
```

```
##  [1] "001.0005" "001.0012" "005.0080" "013.0011" "029.0005" "031.0003"
##  [7] "063.2008" "067.1015" "085.0055" "101.0003"
```

g) We observe that the list contains 10 monitors. Rather than choosing a monitor at random, it would make more sense to choose one that had the most observations. (13) Write a block of code to identify the monitor in the original data (i.e., pm) that had the most data using mutate(), filter(), group_by(), summarize(), and arrange().

```
monitor_count = mutate(pm, Site.Code = Site.Code)
monitor_count =
pm %>%
  filter(Site.Code %in% monitors) %>%
  group_by(Site.Code) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count))
monitor_count
```

```
## # A tibble: 10 × 2
##    Site.Code Count
##    <chr>     <int>
##  1 101.0003    527
##  2 013.0011    213
##  3 031.0003    198
##  4 001.0005    186
##  5 067.1015    153
##  6 063.2008    152
##  7 029.0005     94
##  8 001.0012     92
##  9 005.0080     92
## 10 085.0055     38
```

h) It seems that monitor 101.0003 had collected the most data in the U.S. (i.e., pm) during 1999 and 2012 (n = 527). (14) Subset the data (i.e., pm) that contains observations from the monitor we just identified (State.Code = 36 & County.Code = 101 & Site.ID = 0003) and assign the subset data to an obj. called ‘pmsub’.

All data after this code (until further subsets) will be using the subsetted data from the code below

```
pmsub = pm[State.Code == "36" & County.Code == "101" & Site.ID == "0003", ]
```

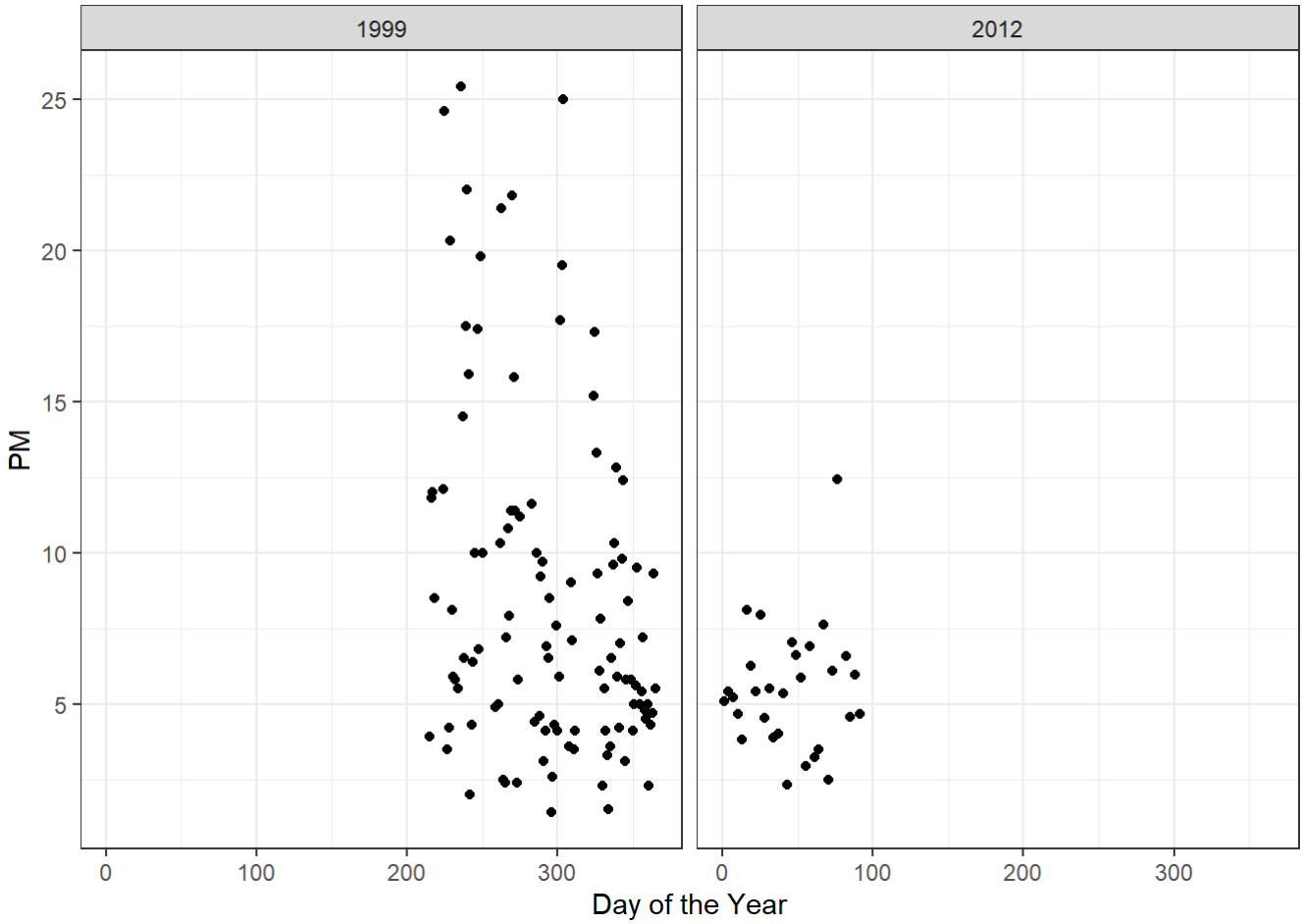
i) Next, using the lubridate package, (15) convert the Date variable into a date obj. and then create a variable called ‘yday’ containing info. on day of the year using yday().

```
pmsub$Date = as.Date(as.character(pmsub$Date), format = "%Y%m%d")
pmsub$yday = yday(pmsub$Date)
```

j) Draw a scatter plot by mapping the year-day variable on the x-axis, PM2.5 level on the y-axis separately for 1999 and 2012. (16) Make sure to label the x-axis, (17) separate the plots using the facet function and (18) use the base white theme to replicate the graphics shown below.

```
ggplot(pmsub, aes(x=yday, y=PM)) +
  geom_point() +
  facet_wrap(. ~ Year) +
  labs(x="Day of the Year", y="PM") +
  theme_bw()
```

```
## Warning: Removed 45 rows containing missing values (`geom_point()`).
```



k) Interesting pattern observed is that the variation (spread) in the PM values in 2012 is much smaller (vs. larger in aggregate) than it was in 1999. The plot shows that not only are the average levels of PM lower in 2012, but that there are fewer large spikes from day to day in 2012.