PM2.5_Analysis, Laos

Sengdao Oudomsihn

2023-04-26

```
# install.packages("tidyverse")
# install.packages("lubridate")
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.2     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(lubridate)
```

```
myurl = 'http://berkeleyearth.lbl.gov/air-quality/maps/cities/Laos/Laos.txt'
data_laos <-  read_tsv(myurl, skip = 8, col_names = FALSE  )
```

```
## Rows: 43791 Columns: 7
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: "\t"
## dbl (7): X1, X2, X3, X4, X5, X6, X7
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data_laos
```

```
## # A tibble: 43,791 × 7
##       X1    X2    X3    X4    X5    X6    X7
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  2016     3    22     5  63.1  0.35     0
##  2  2016     3    22     6  60.4  0.35     0
##  3  2016     3    22     7  65.2  0.35     0
##  4  2016     3    23    22  70.0  0.35     0
##  5  2016     3    23    23  70.4  0.35     0
##  6  2016     3    24     1  57.7  0.35     0
##  7  2016     3    25    18  27.8  0.34     0
##  8  2016     3    25    19  33.8  0.34     0
##  9  2016     3    25    20  37.1  0.34     0
## 10  2016     3    25    21  37.0  0.35     0
## # ℹ 43,781 more rows
```

```
colnames(data_laos) <-  c('year', 'month', 'day', 'hour_UTC', 'pm2_5', 'X6', 'X7' )
data_laos
```

```
## # A tibble: 43,791 × 7
##     year month   day hour_UTC pm2_5    X6    X7
##    <dbl> <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl>
##  1  2016     3    22        5  63.1  0.35     0
##  2  2016     3    22        6  60.4  0.35     0
##  3  2016     3    22        7  65.2  0.35     0
##  4  2016     3    23       22  70.0  0.35     0
##  5  2016     3    23       23  70.4  0.35     0
##  6  2016     3    24        1  57.7  0.35     0
##  7  2016     3    25       18  27.8  0.34     0
##  8  2016     3    25       19  33.8  0.34     0
##  9  2016     3    25       20  37.1  0.34     0
## 10  2016     3    25       21  37.0  0.35     0
## # ℹ 43,781 more rows
```

```
data_laos <- data_laos %>%  select(year:pm2_5)
data_laos
```

```
## # A tibble: 43,791 × 5
##     year month   day hour_UTC pm2_5
##    <dbl> <dbl> <dbl>    <dbl> <dbl>
##  1  2016     3    22        5  63.1
##  2  2016     3    22        6  60.4
##  3  2016     3    22        7  65.2
##  4  2016     3    23       22  70.0
##  5  2016     3    23       23  70.4
##  6  2016     3    24        1  57.7
##  7  2016     3    25       18  27.8
##  8  2016     3    25       19  33.8
##  9  2016     3    25       20  37.1
## 10  2016     3    25       21  37.0
## # ℹ 43,781 more rows
```

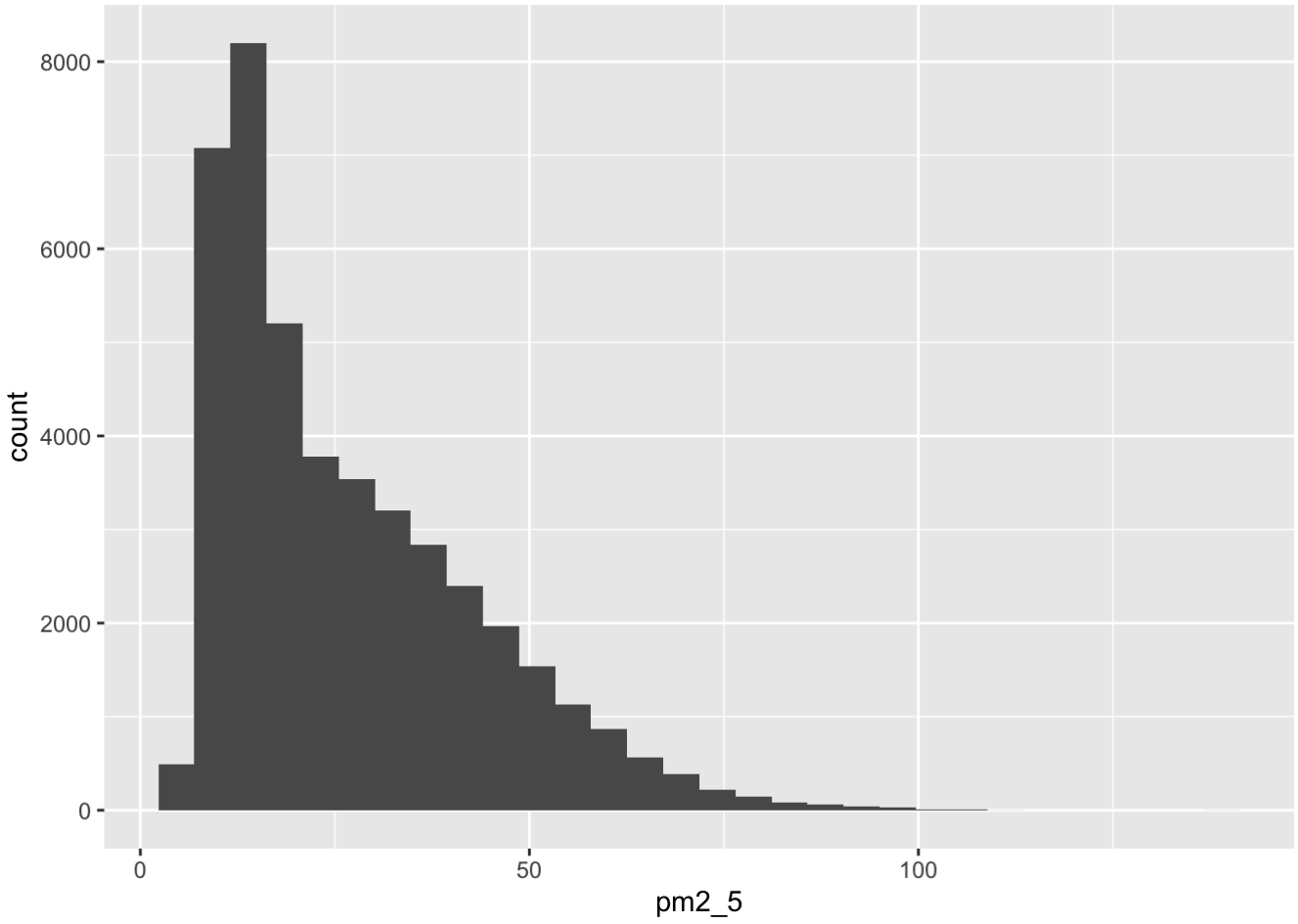
```
data_laos <-  data_laos %>%
  mutate(date_time = ISOdate(year, month, day, hour_UTC),
         local_date_time = date_time + hours(7),
         local_hour = hour(local_date_time))
data_laos
```

```
## # A tibble: 43,791 × 8
##     year month   day hour_UTC pm2_5 date_time           local_date_time
##    <dbl> <dbl> <dbl>    <dbl> <dbl> <dttm>              <dttm>
##  1  2016     3    22        5  63.1 2016-03-22 05:00:00 2016-03-22 12:00:00
##  2  2016     3    22        6  60.4 2016-03-22 06:00:00 2016-03-22 13:00:00
##  3  2016     3    22        7  65.2 2016-03-22 07:00:00 2016-03-22 14:00:00
##  4  2016     3    23       22  70.0 2016-03-23 22:00:00 2016-03-24 05:00:00
##  5  2016     3    23       23  70.4 2016-03-23 23:00:00 2016-03-24 06:00:00
##  6  2016     3    24        1  57.7 2016-03-24 01:00:00 2016-03-24 08:00:00
##  7  2016     3    25       18  27.8 2016-03-25 18:00:00 2016-03-26 01:00:00
##  8  2016     3    25       19  33.8 2016-03-25 19:00:00 2016-03-26 02:00:00
##  9  2016     3    25       20  37.1 2016-03-25 20:00:00 2016-03-26 03:00:00
## 10  2016     3    25       21  37.0 2016-03-25 21:00:00 2016-03-26 04:00:00
## # ℹ 43,781 more rows
## # ℹ 1 more variable: local_hour <int>
```

visualize data

```
data_laos %>%
  ggplot(aes(pm2_5))+
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

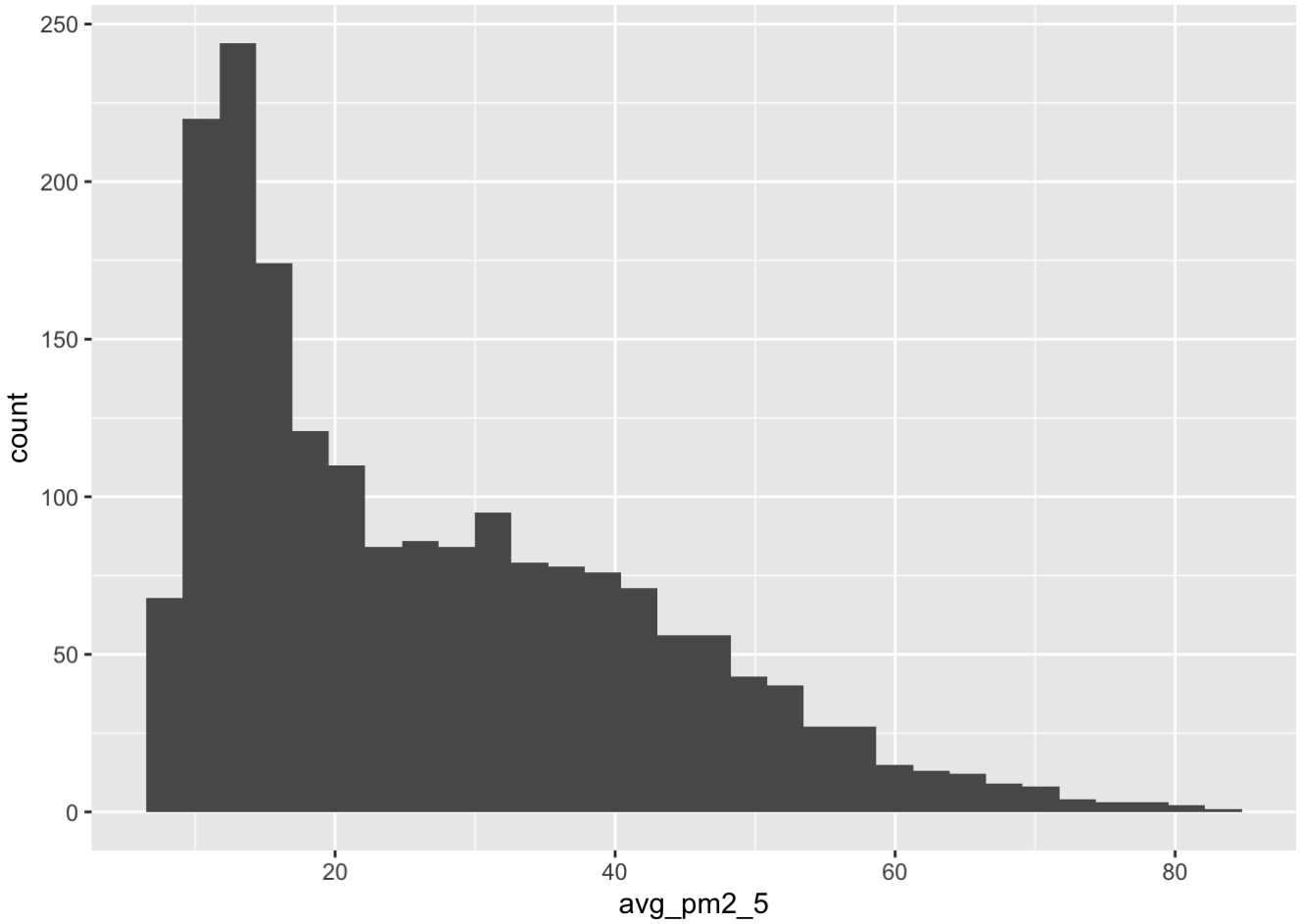



```
data_laos %>%
  group_by(date(local_date_time)) %>%
  summarise(avg_pm2_5 = mean(pm2_5))
```

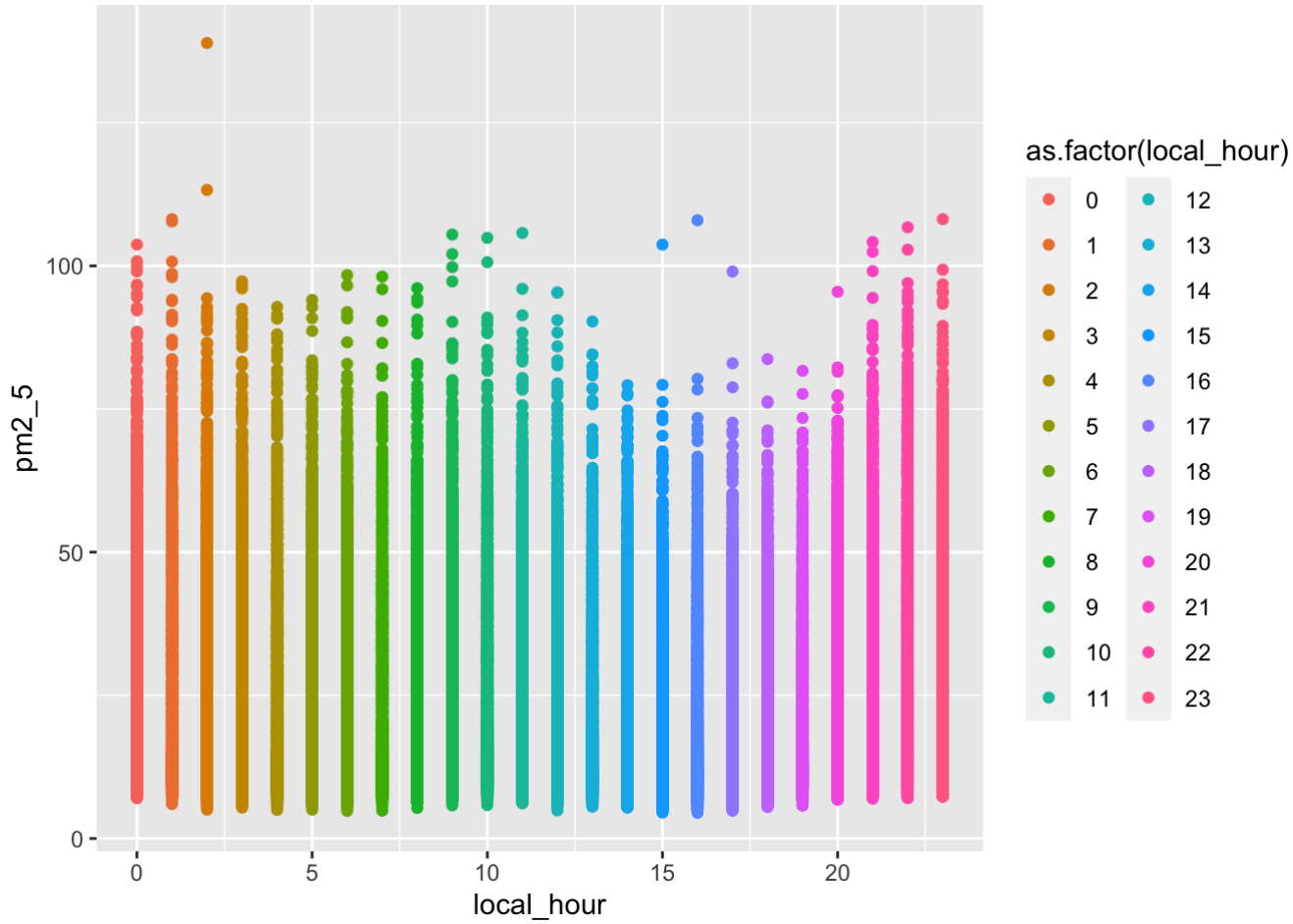
```
## # A tibble: 1,909 × 2
##    `date(local_date_time)` avg_pm2_5
##    <date>                      <dbl>
##  1 2016-03-22                   62.9
##  2 2016-03-24                   66.0
##  3 2016-03-26                   34.5
##  4 2016-03-27                   44.3
##  5 2016-03-28                   45.6
##  6 2016-03-29                   69.8
##  7 2016-03-30                   69.4
##  8 2016-03-31                   79.9
##  9 2016-04-01                   66.5
## 10 2016-04-02                   54.9
## # ℹ 1,899 more rows
```

```
data_laos %>%
  group_by(date(local_date_time)) %>%
  summarise(avg_pm2_5 = mean(pm2_5)) %>%
  ggplot(aes(avg_pm2_5))+
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
data_laos %>%
  ggplot(aes(x =local_hour, y = pm2_5, color = as.factor(local_hour)))+
  geom_point()
```



top five best days (lowest PM2.5)

```
data_laos %>%
  mutate(month = month(local_date_time)) %>%
  group_by(month) %>%
  summarise(avg_pm2_5 = mean(pm2_5))
```

```
## # A tibble: 12 × 2
##    month avg_pm2_5
##    <dbl>     <dbl>
##  1     1      36.7
##  2     2      37.6
##  3     3      45.8
##  4     4      39.8
##  5     5      21.5
##  6     6      13.0
##  7     7      11.4
##  8     8      12.2
##  9     9      14.8
## 10    10      18.3
## 11    11      26.1
## 12    12      34.1
```

```
data_laos %>%
  mutate(date = date(local_date_time)) %>%
  group_by(date) %>%
  summarise(avg_pm2_5 = mean(pm2_5)) %>%
  arrange(avg_pm2_5) %>%
  top_n(-5)
```

```
## Selecting by avg_pm2_5
```

```
## # A tibble: 5 × 2
##   date       avg_pm2_5
##   <date>         <dbl>
## 1 2021-06-13      6.97
## 2 2020-08-02      7.14
## 3 2021-09-17      7.37
## 4 2021-07-09      7.37
## 5 2020-09-20      7.62
```

```
data_laos %>%
  mutate(date = date(local_date_time)) %>%
  group_by(date) %>%
  summarise(min_pm2_5 = min(pm2_5)) %>%
  arrange(min_pm2_5) %>%
  top_n(-5)
```

```
## Selecting by min_pm2_5
```

```
## # A tibble: 5 × 2
##   date       min_pm2_5
##   <date>         <dbl>
## 1 2021-07-21      4.49
## 2 2021-07-09      4.53
## 3 2021-08-27      4.61
## 4 2021-06-15      4.82
## 5 2021-08-18      4.86
```

top five worst days (highest PM2.5)

```
data_laos %>%
  mutate(month = month(local_date_time)) %>%
  group_by(year,  month) %>%
  summarise(avg_pm2_5 = mean(pm2_5))%>%
  arrange(desc(avg_pm2_5))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 67 × 3
## # Groups:   year [7]
##     year month avg_pm2_5
##    <dbl> <dbl>     <dbl>
##  1  2016     3      61.0
##  2  2016     4      56.9
##  3  2019     3      55.4
##  4  2020     3      53.6
##  5  2018     3      53.5
##  6  2021     1      48.3
##  7  2023     4      47.3
##  8  2018     2      43.8
##  9  2021     3      42.8
## 10  2020     4      42.7
## # ℹ 57 more rows
```

```
data_laos %>%
  mutate(date = date(local_date_time)) %>%
  group_by(date) %>%
  summarise(avg_pm2_5 = mean(pm2_5)) %>%
  arrange(desc(avg_pm2_5)) %>%
  top_n(5)
```

```
## Selecting by avg_pm2_5
```

```
## # A tibble: 5 × 2
##   date       avg_pm2_5
##   <date>         <dbl>
## 1 2020-04-01      82.6
## 2 2020-03-29      81.2
## 3 2016-03-31      79.9
## 4 2019-03-14      79.2
## 5 2018-03-26      77.2
```

```
data_laos %>%
  mutate(date = date(local_date_time)) %>%
  group_by(date) %>%
  summarise(max_pm2_5 = max(pm2_5)) %>%
  arrange(desc(max_pm2_5)) %>%
  top_n(5)
```

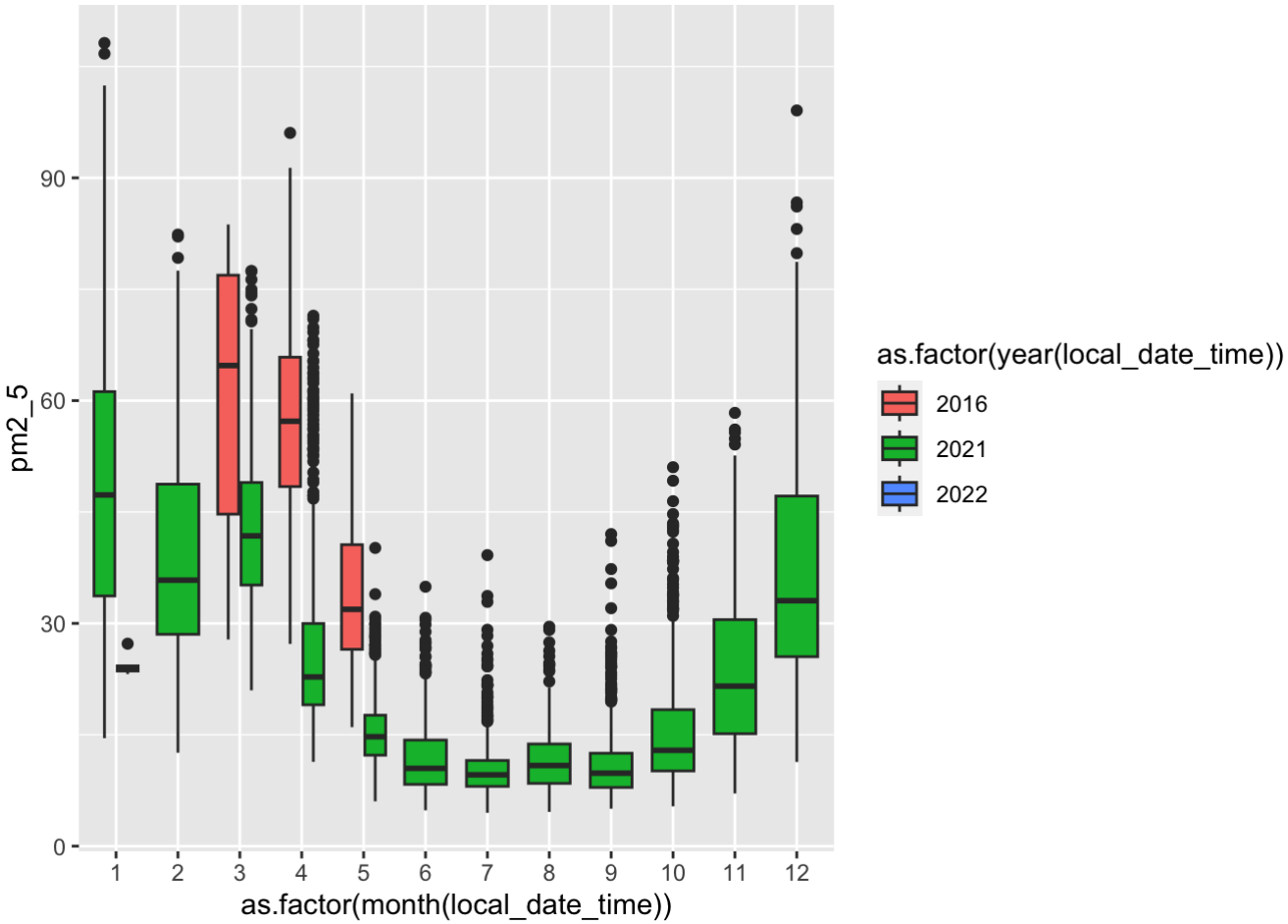
```
## Selecting by max_pm2_5
```

```
## # A tibble: 5 × 2
##   date       max_pm2_5
##   <date>         <dbl>
## 1 2018-02-16      139.
## 2 2020-04-01      113.
## 3 2021-01-20      108.
## 4 2019-03-31      108.
## 5 2021-01-21      107.
```

```
data_laos %>%
  mutate(weekdays = weekdays(local_date_time)) %>%
  group_by(weekdays) %>%
  summarise(avg_pm2_5 = mean(pm2_5))
```

```
## # A tibble: 7 × 2
##   weekdays  avg_pm2_5
##   <chr>         <dbl>
## 1 Friday         26.8
## 2 Monday         26.2
## 3 Saturday       26.5
## 4 Sunday         26.5
## 5 Thursday       27.3
## 6 Tuesday        26.8
## 7 Wednesday      26.9
```

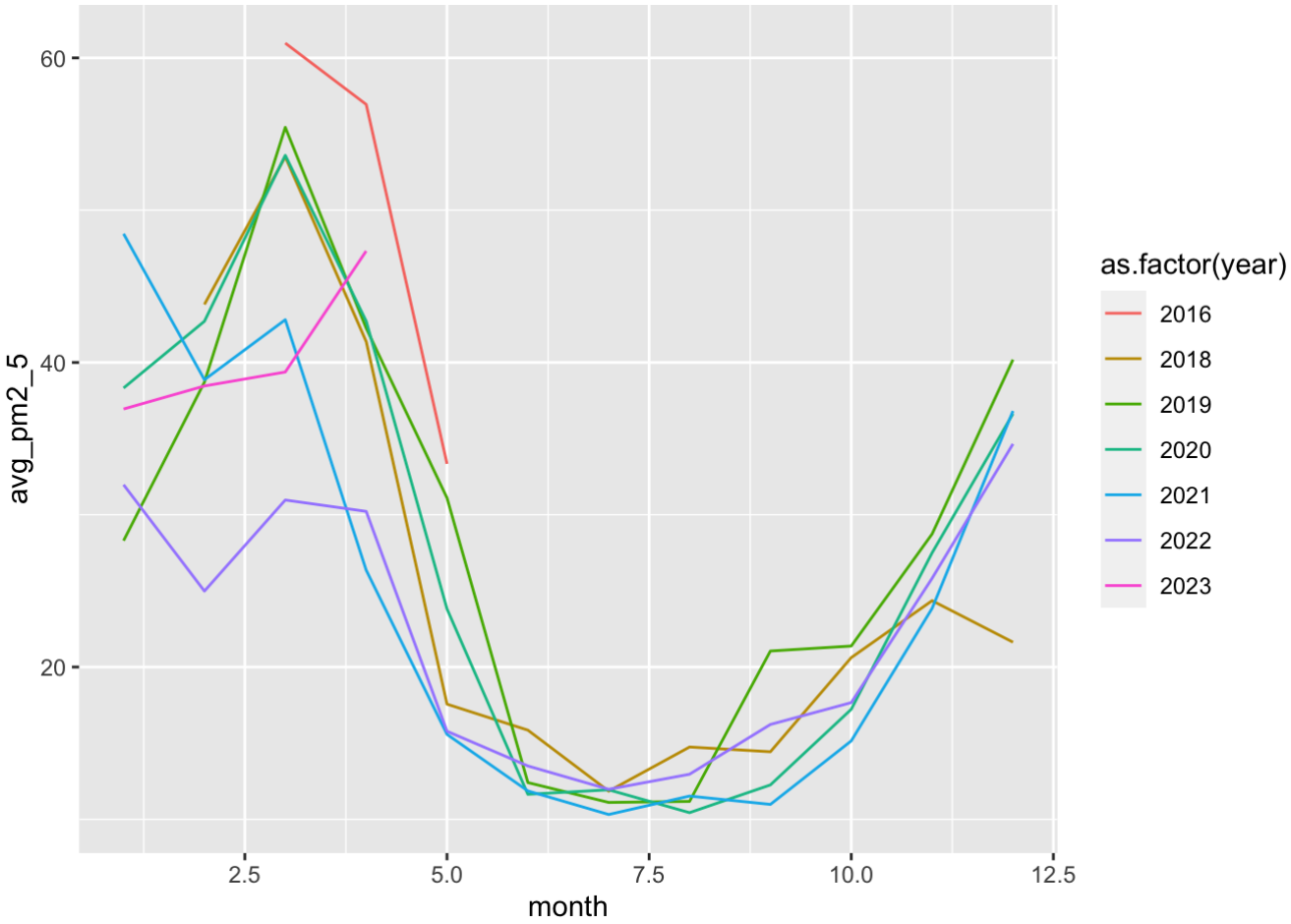
```
data_laos %>%
  filter(year == 2016 | year == 2021) %>%
  ggplot(aes(x = as.factor(month(local_date_time)), y =pm2_5,
             fill = as.factor(year(local_date_time))))+
  geom_boxplot()
```



#time series

```
data_laos %>%
  mutate(month = month(local_date_time),
         year = year(local_date_time)) %>%
  group_by(year, month) %>%
  summarise(avg_pm2_5 = mean(pm2_5)) %>%
  ggplot(aes(x = month, y =avg_pm2_5, color = as.factor(year)))+
  geom_line()
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```


60
40
20
avg_pm2_5
2.5
5.0
7.5
10.0
12.5
month
as.factor(year)
2016
2018
2019
2020
2021
2022
2023