미세먼지 예측

데이터 준비

(1) 8개 도시 월별 미세먼지 측정량 불러오기

```
library(tidyverse)
```

```
## ── Attaching packages ───────────────────────────────────────
─────────────────────────────────────────────────────────────
──────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.0     ✓ purrr   0.3.3
## ✓ tibble  3.0.4     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────────────
─────────────────────────────────────────────────────────────
─────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(fpp3)
```

```
## ── Attaching packages ───────────────────────────────────────
─────────────────────────────────────────────────────────────
──────────────────────────── fpp3 0.3 ──
```

```
## ✓ lubridate   1.7.4     ✓ feasts      0.1.5
## ✓ tsibble     0.9.3     ✓ fable       0.2.1
## ✓ tsibbledata 0.2.0
```

```
## ── Conflicts ────────────────────────────────────────────────
─────────────────────────────────────────────────────────────
─────────────────────────── fpp3_conflicts ──
## x lubridate::date()       masks base::date()
## x dplyr::filter()         masks stats::filter()
## x tsibble::interval()     masks lubridate::interval()
## x dplyr::lag()            masks stats::lag()
## x tsibble::new_interval() masks lubridate::new_interval()
```

```
df <- read_csv("PM10w.csv")
```

```
## Parsed with column specification:
## cols(
##   yymm = col_character(),
##   tot = col_double(),
##   seoul = col_double(),
##   busan = col_double(),
##   daegu = col_double(),
##   incheon = col_double(),
##   gwangju = col_double(),
##   daejeon = col_double(),
##   ulsan = col_double(),
##   sejong = col_double()
## )
```

```
df <- df[,-2]
```

```
df
```

```
## # A tibble: 123 x 9
##    yymm     seoul busan daegu incheon gwangju daejeon ulsan sejong
##    <chr>    <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl> <dbl>  <dbl>
##  1 2010. 01    59    47    56      64      46      48    46     NA
##  2 2010. 02    50    44    49      54      39      39    44     NA
##  3 2010. 03    61    64    69      67      65      52    60     NA
##  4 2010. 04    49    50    47      55      42      41    47     NA
##  5 2010. 05    56    56    55      62      62      52    54     NA
##  6 2010. 06    51    46    47      57      39      40    50     NA
##  7 2010. 07    33    41    36      37      27      26    39     NA
##  8 2010. 08    32    42    38      37      28      27    40     NA
##  9 2010. 09    25    38    36      34      29      28    36     NA
## 10 2010. 10    41    41    43      51      42      40    39     NA
## # … with 113 more rows
```

(2) tsibble로 변환

```
pm10w<- df %>%
  mutate(Month=yearmonth(yymm)) %>%
  select(-yymm) %>%
  as_tsibble(index=Month)
```

```
pm10w
```

```
## # A tsibble: 123 x 9 [1M]
##    seoul busan daegu incheon gwangju daejeon ulsan sejong    Month
##    <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl> <dbl>  <dbl>    <mth>
##  1    59    47    56      64      46      48    46     NA 2010 Jan
##  2    50    44    49      54      39      39    44     NA 2010 Feb
##  3    61    64    69      67      65      52    60     NA 2010 Mar
##  4    49    50    47      55      42      41    47     NA 2010 Apr
##  5    56    56    55      62      62      52    54     NA 2010 May
##  6    51    46    47      57      39      40    50     NA 2010 Jun
##  7    33    41    36      37      27      26    39     NA 2010 Jul
##  8    32    42    38      37      28      27    40     NA 2010 Aug
##  9    25    38    36      34      29      28    36     NA 2010 Sep
## 10    41    41    43      51      42      40    39     NA 2010 Oct
## # … with 113 more rows
```

```
class(pm10w)
```

```
## [1] "tbl_ts"     "tbl_df"     "tbl"        "data.frame"
```

```
index(pm10w)
```

```
## Month
```

(3) Wide format(pm10w)에서 Long format(pm10)으로 변환

```
pm10 <- pivot_longer(pm10w, cols=c(-Month, seoul, busan, daegu, incheon, gwangju, daejeon, ulsa
n, sejong), names_to='city', values_to='concentration')
```

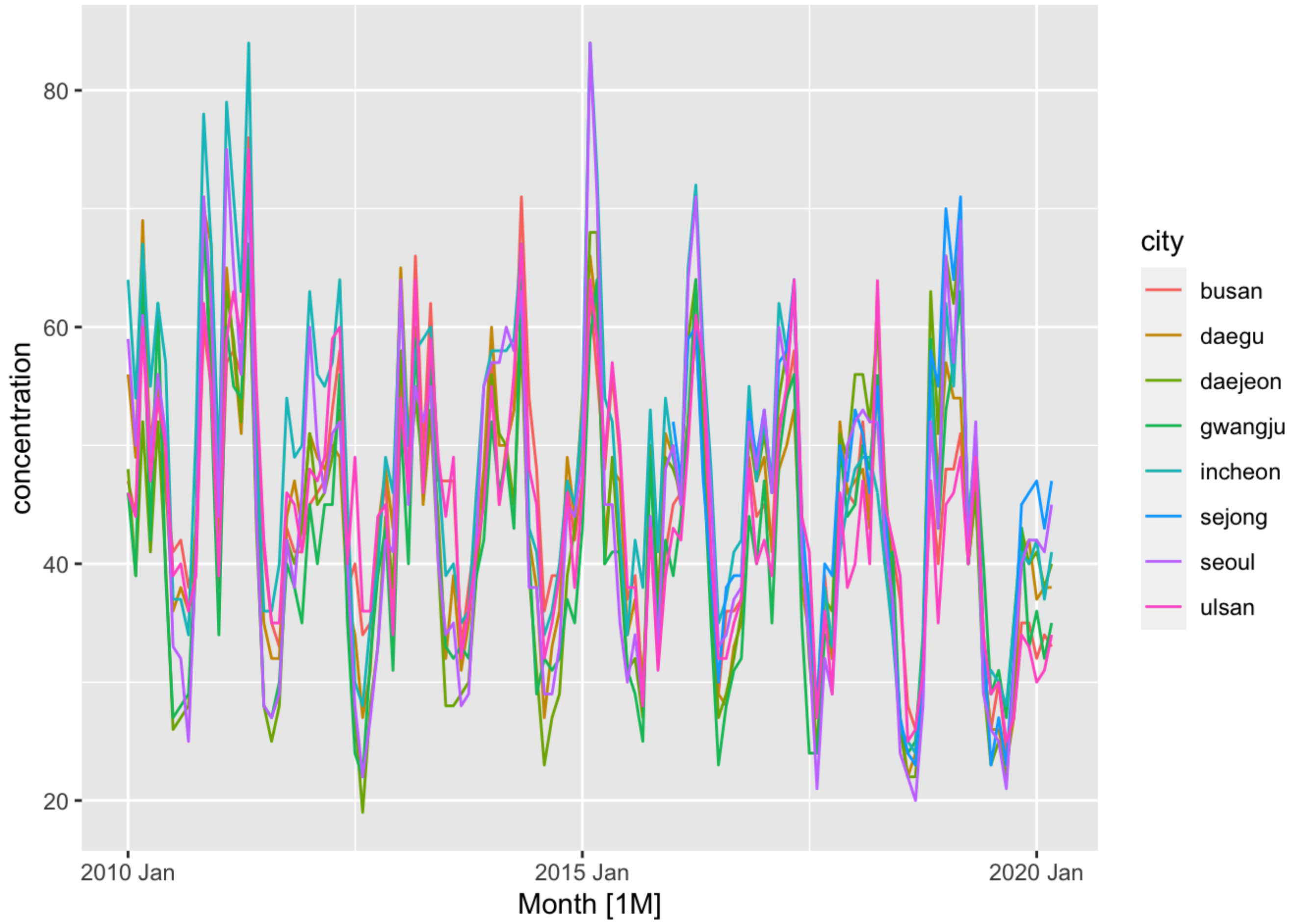
```
pm10
```

```
## # A tsibble: 984 x 3 [1M]
## # Key:       city [8]
##       Month city    concentration
##       <mth> <chr>           <dbl>
##  1 2010 Jan seoul              59
##  2 2010 Jan busan              47
##  3 2010 Jan daegu              56
##  4 2010 Jan incheon            64
##  5 2010 Jan gwangju            46
##  6 2010 Jan daejeon            48
##  7 2010 Jan ulsan              46
##  8 2010 Jan sejong             NA
##  9 2010 Feb seoul              50
## 10 2010 Feb busan              44
## # … with 974 more rows
```

시계열 그림

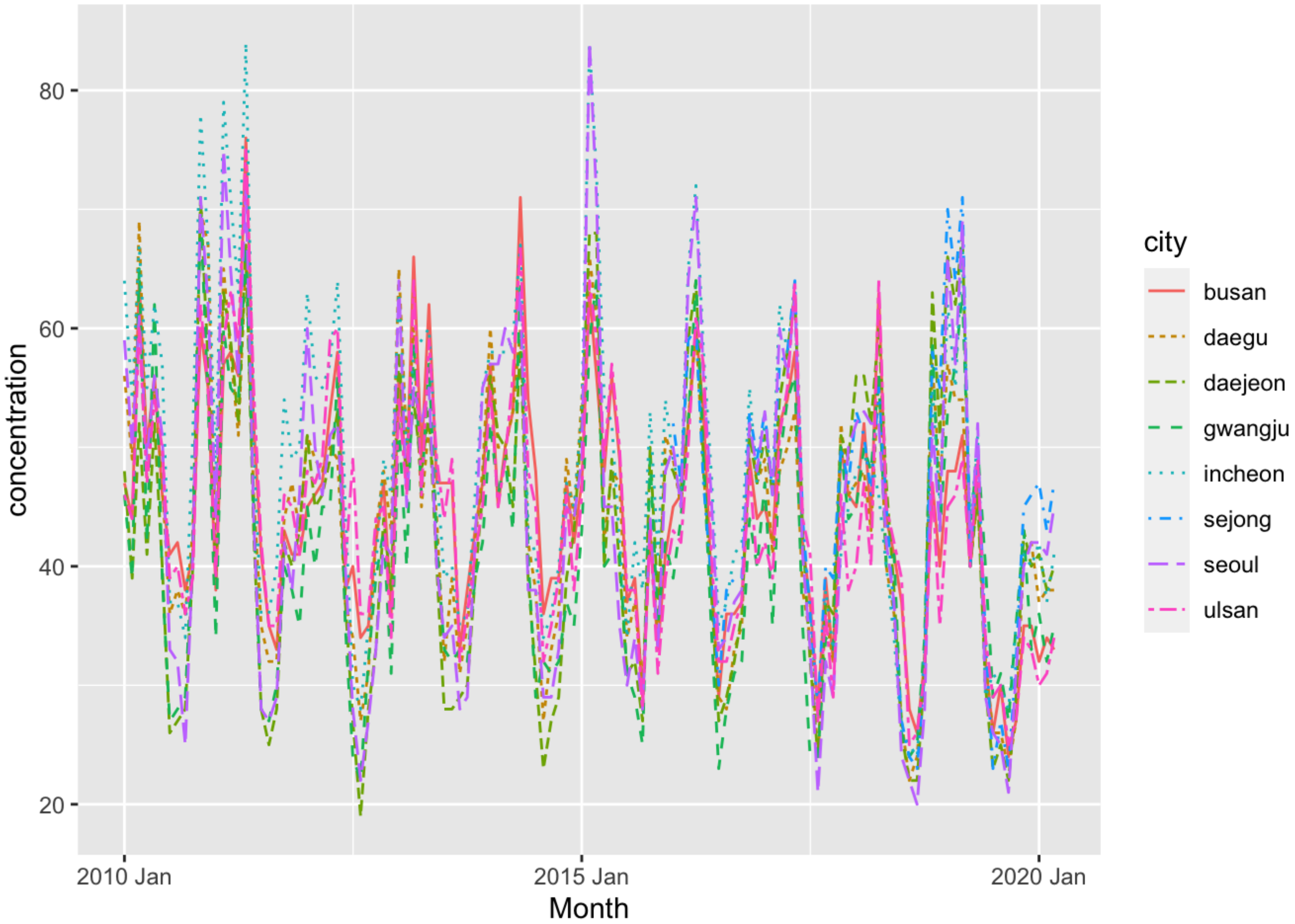
```
autoplot(pm10, concentration)
```

```
## Warning: Removed 72 row(s) containing missing values (geom_path).
```




```
# 위와 같은 그림을 가지는 다른 방법
ggplot(pm10, aes(x=Month, y=concentration, color=city)) +
  geom_line(aes(linetype=city))
```

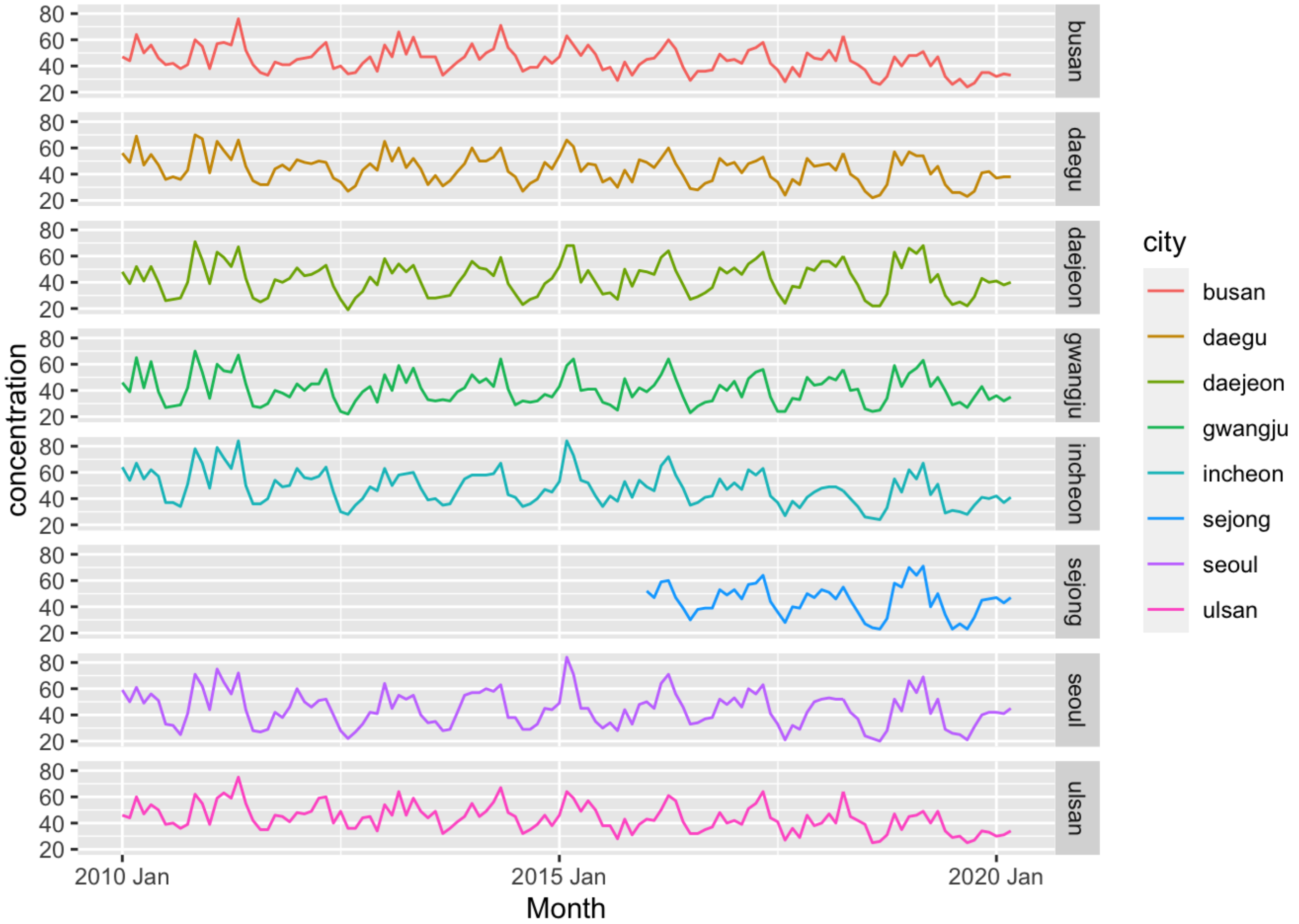
```
## Warning: Removed 72 row(s) containing missing values (geom_path).
```



- 분산이 일정하진 않지만 비교적 일정한 편으로 보인다.

```
ggplot(pm10, aes(x=Month, y=concentration, color=city, group=city)) +
  geom_line() +
  facet_grid(city~.)
```

```
## Warning: Removed 72 row(s) containing missing values (geom_path).
```

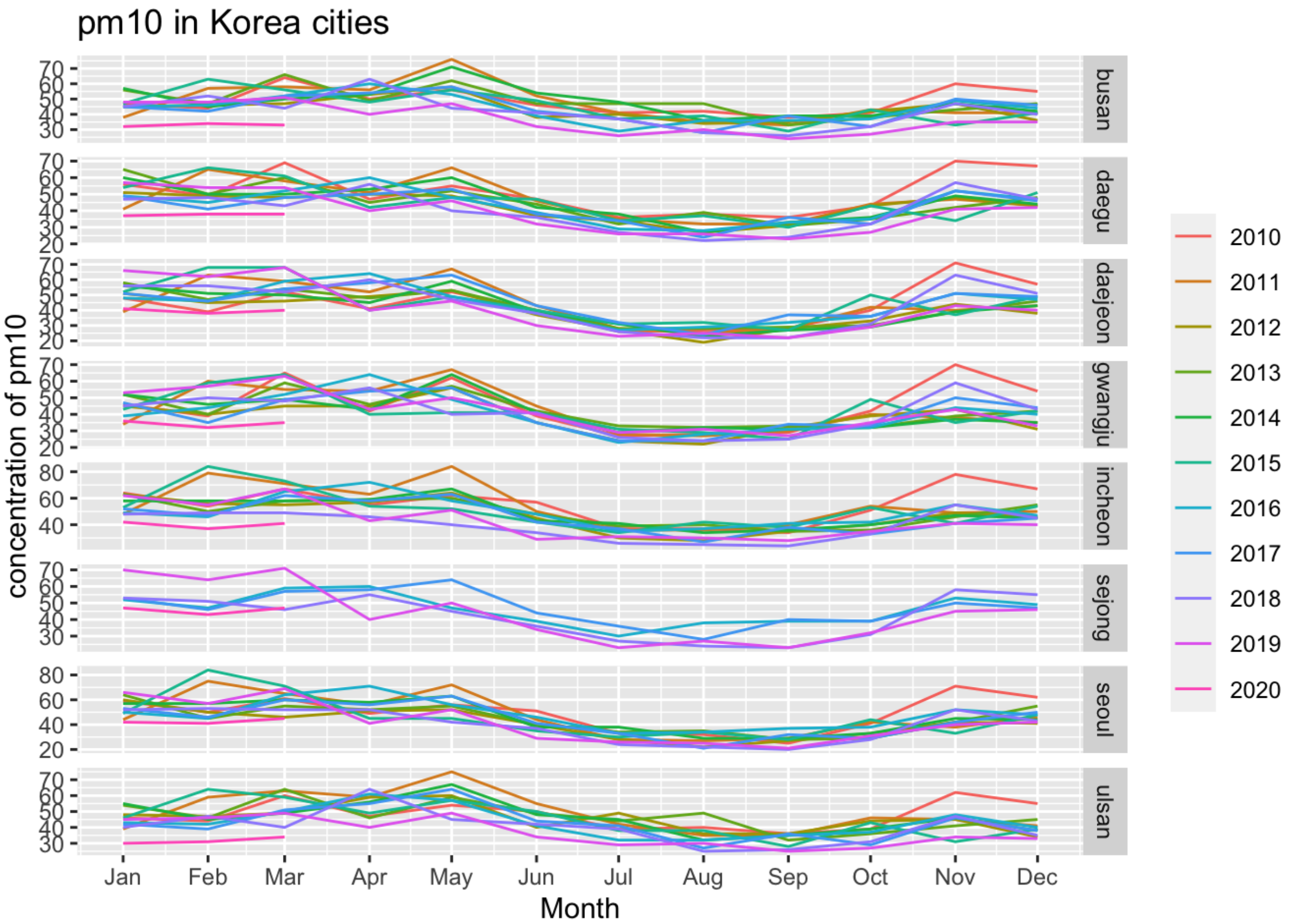


- 각 도시의 미세먼지 측정량 그래프가 비슷한 모양이다. 계절성이 있음을 짐작해볼 수 있다.

계절성 그림

gg_season

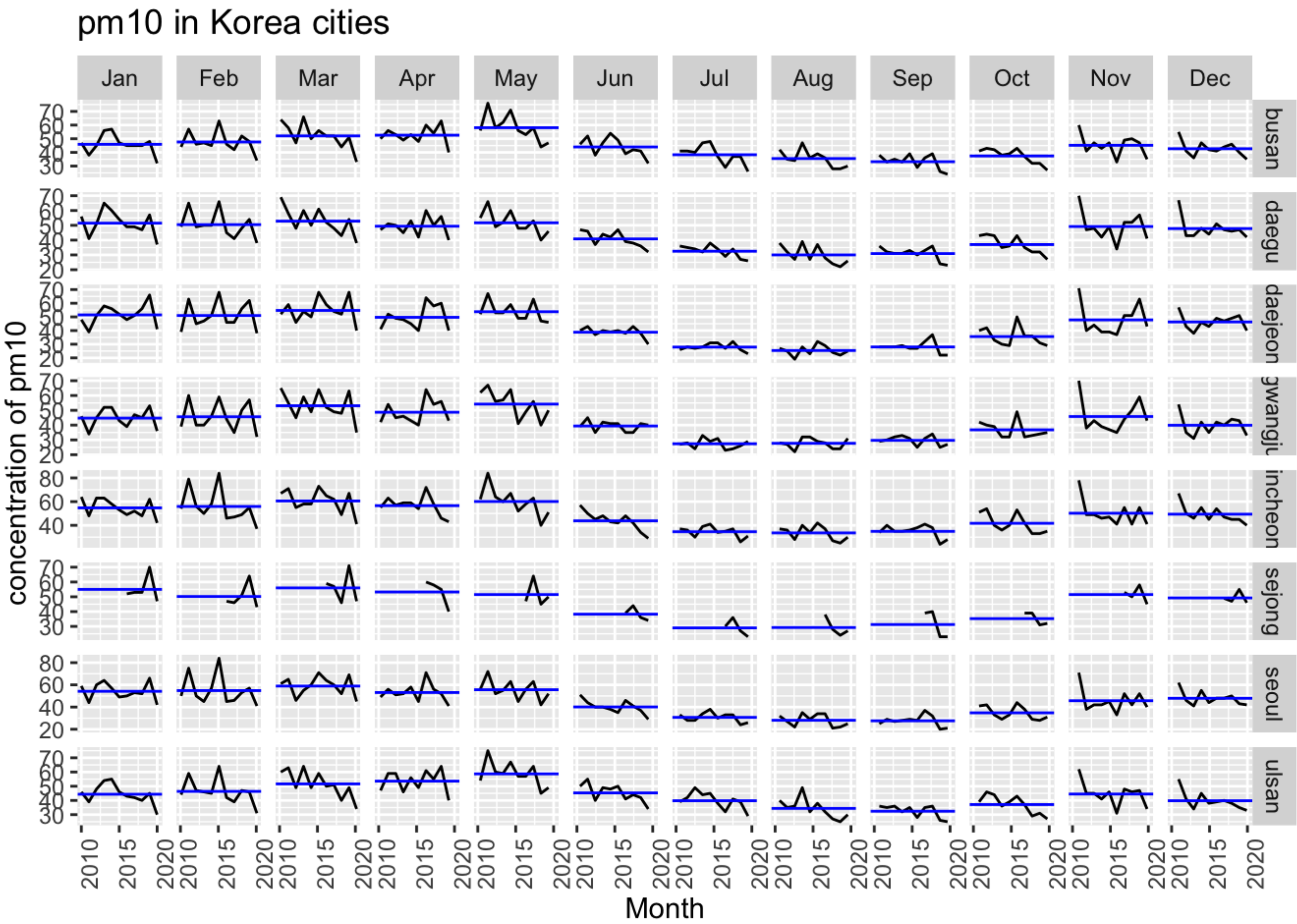
```
pm10 %>%
  gg_season(concentration) +
  ylab("concentration of pm10") +
  ggtitle("pm10 in Korea cities")
```



- 각 도시마다 시기별로 그래프의 모양이 비슷하고 도시별로 비교하여도 비슷하다. 계절성이 있다.

gg_subseries

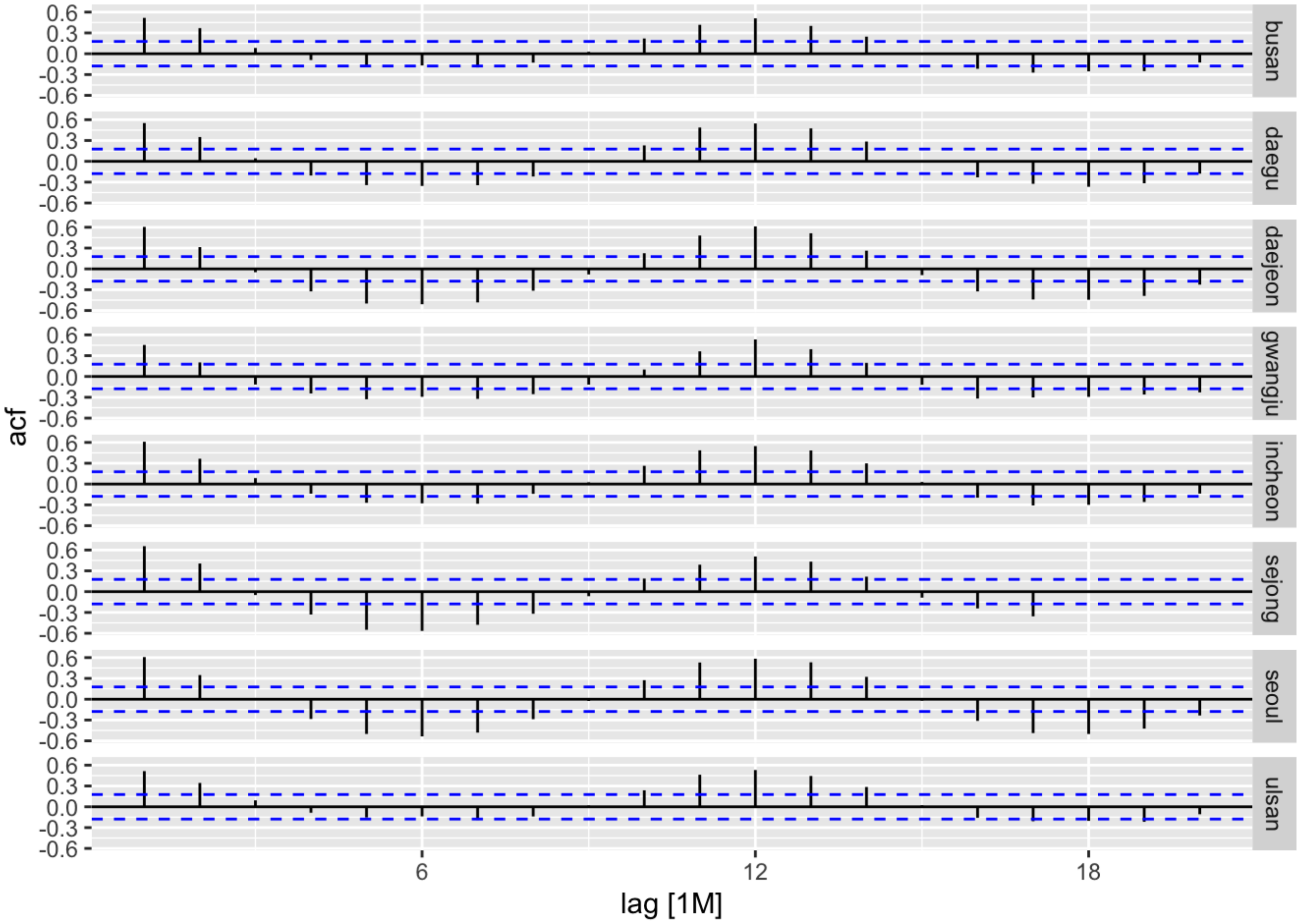
```
pm10 %>%
  gg_subseries(concentration) +
  ylab("concentration of pm10") +
  ggtitle("pm10 in Korea cities")
```



- 도시별 월별 그림을 그려보아도 여름에는 낮고 겨울,봄에 높은 계절성을 확인할 수 있다.

ACF의 특징 기술

```
pm10 %>% ACF(concentration) %>% autoplot()
```



- 낮은 lag 차수에도 기각역을 꽤 벗어나는 것을 볼 수 있다. 백색잡음이 아닐 것이라 추측해볼 수 있다. 이는 아직 남은 정보가 있다는 의미이다.

Ljung-Box 검정

```
Box.test(pm10$concentration, lag=12, type='L')
```

```
##
##  Box-Ljung test
##
## data:  pm10$concentration
## X-squared = 3747.5, df = 12, p-value < 2.2e-16
```

- ljung-box의 가설 H0: p1 = ... = p12 = 0 p-value가 0.05보다 작으므로 H0를 기각한다. 따라서 백색잡음이 아니다. 남은 정보가 있다.

서울의 미세먼지 측정량

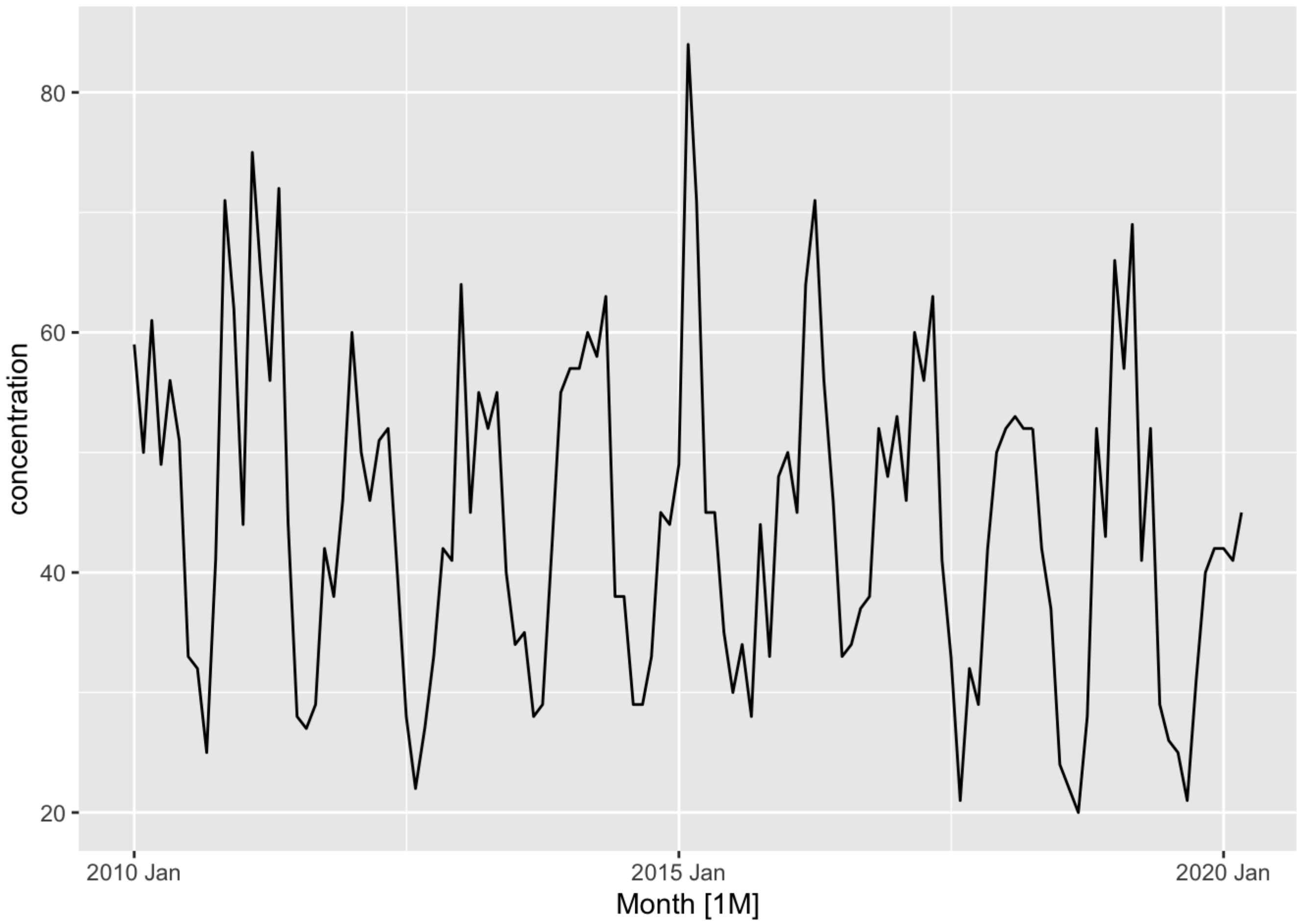
```
pm10s <- pm10 %>%
  filter(city=="seoul") %>%
  select(Month, concentration)

pm10s
```

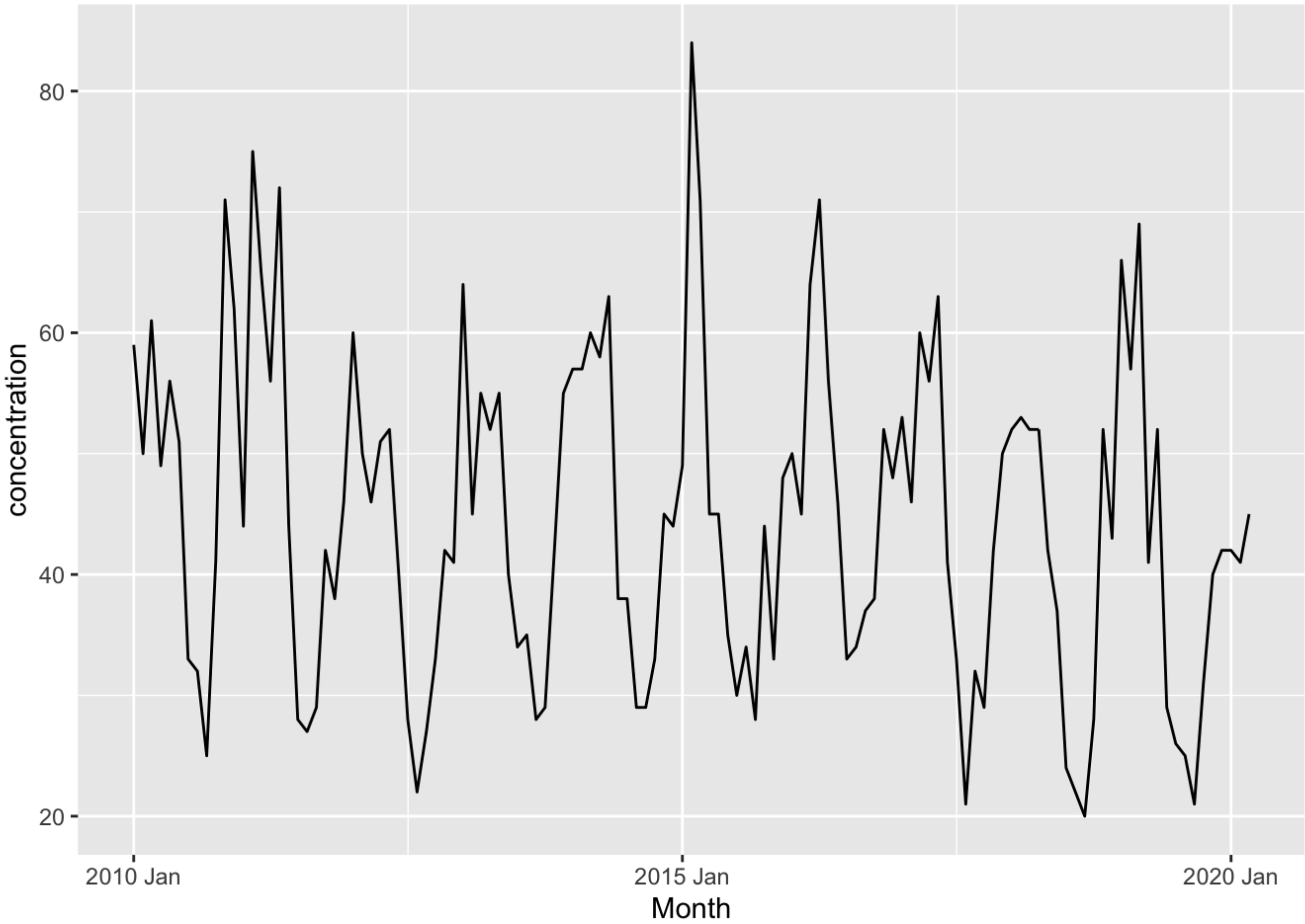
```
## # A tsibble: 123 x 2 [1M]
##       Month concentration
##       <mth>         <dbl>
##  1 2010 Jan            59
##  2 2010 Feb            50
##  3 2010 Mar            61
##  4 2010 Apr            49
##  5 2010 May            56
##  6 2010 Jun            51
##  7 2010 Jul            33
##  8 2010 Aug            32
##  9 2010 Sep            25
## 10 2010 Oct            41
## # … with 113 more rows
```

시계열 그림

```
autoplot(pm10s, concentration)
```




```
# 위와 같은 그림을 그리는 다른 방법
ggplot(pm10s, aes(x=Month, y=concentration)) +
  geom_line()
```

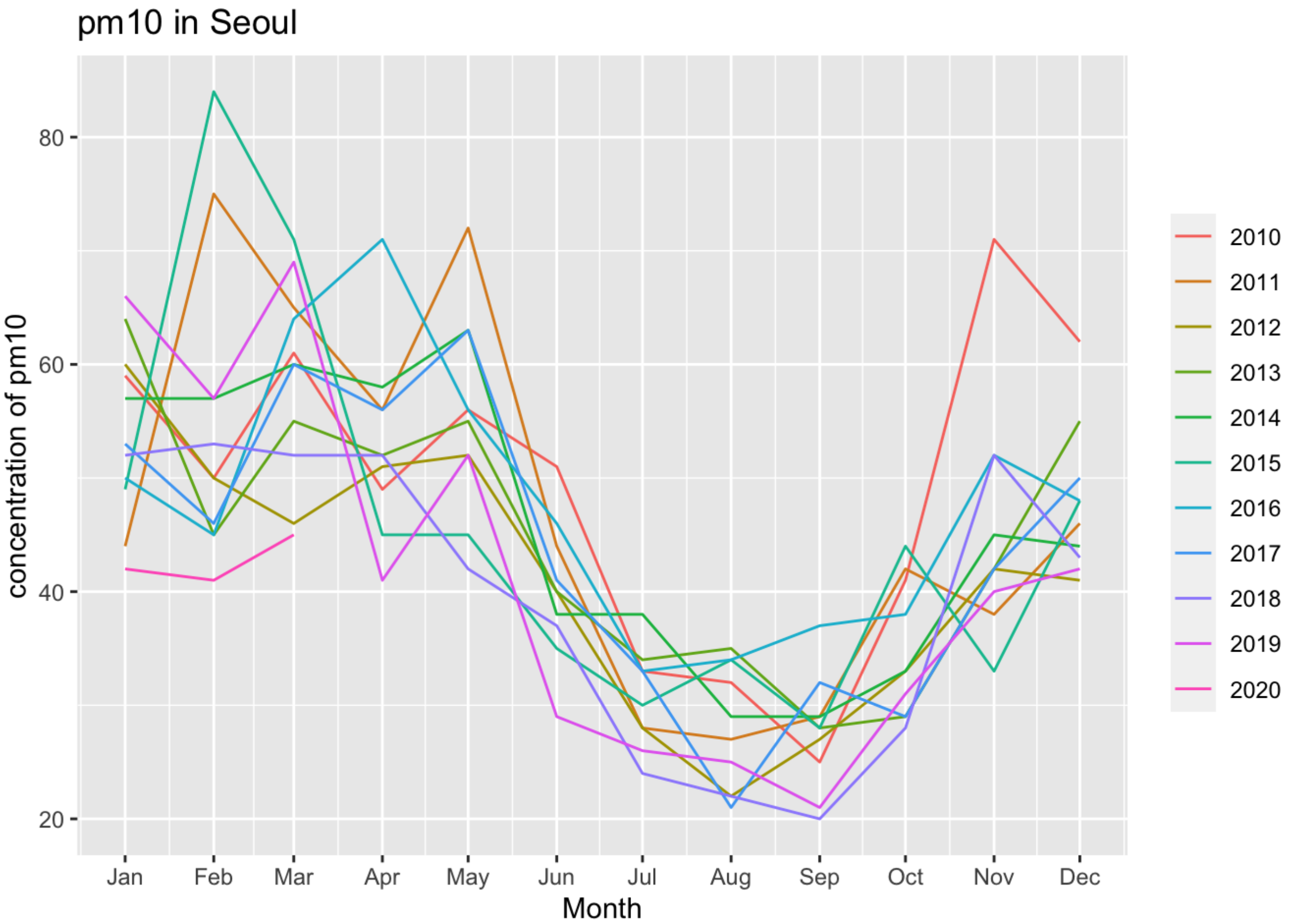


- 분산이 일정하지 않지만 비교적 일정한 것으로 보인다.

계절성 검토

gg_season

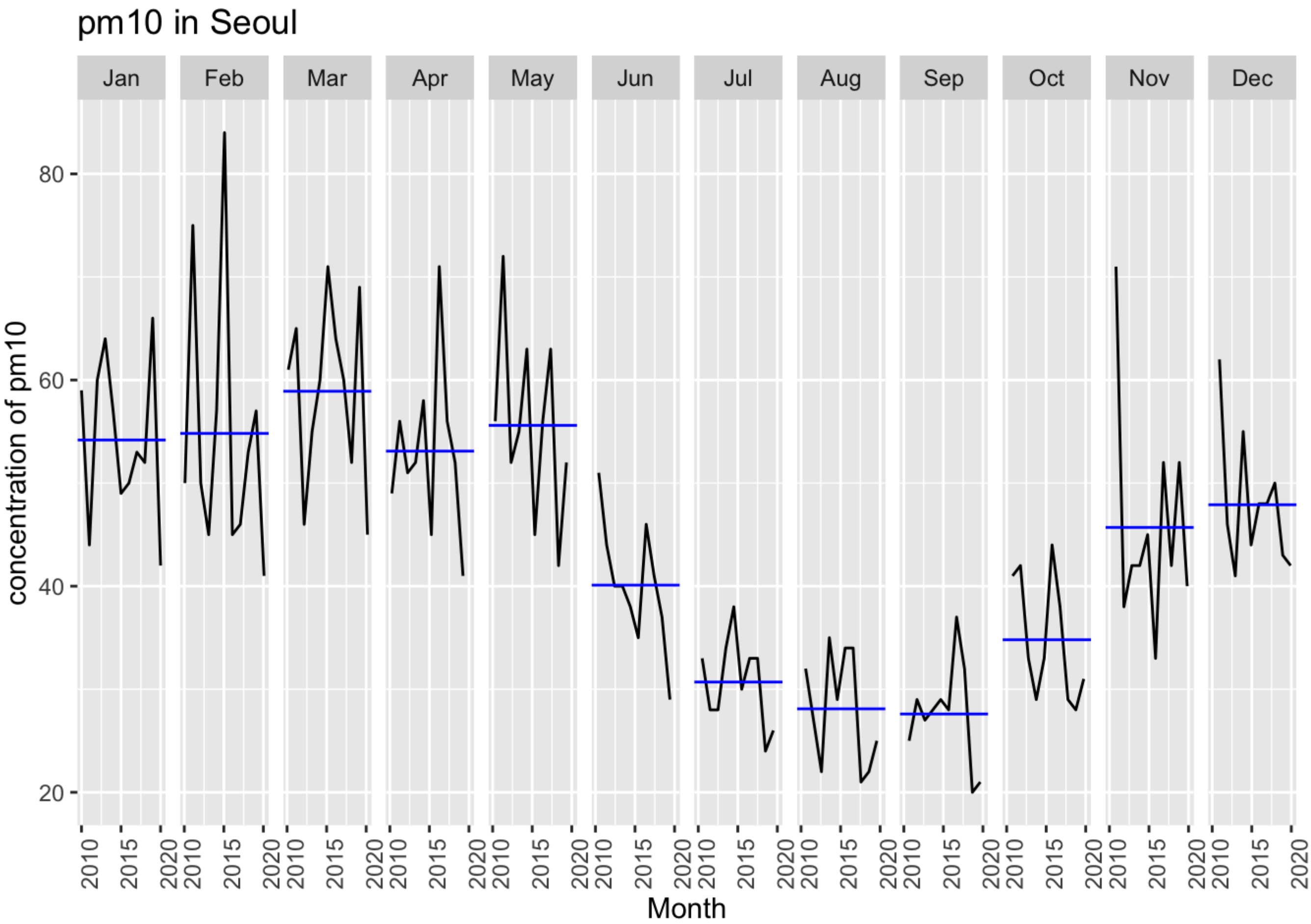
```
pm10s %>%
  gg_season(concentration) +
  ylab("concentration of pm10") +
  ggtitle("pm10 in Seoul")
```



- 모든 연도의 그래프가 겨울에서 봄에 가장 미세먼지 측정량이 높아지고 여름에 낮아지는 모양을 하고 있다. 계절성이 있음을 알 수 있다.

gg_subseries

```
pm10s %>%
  gg_subseries(concentration) +
  ylab("concentration of pm10") +
  ggtitle("pm10 in Seoul")
```



- 3월에 가장 높아지며 9월에 가장 낮아지는 것을 확인할 수 있다. 겨울에서 봄에 미세먼지 측정량이 높아지며 여름에 낮아지는 계절성이 있다.
- 정리: 계절성이 있고 추세는 불명확해보인다.

서울 미세먼지(pm10s) 분석

데이터 분할

```
TRN <- filter_index(pm10s, ~'2017 DEC')
TST <- filter_index(pm10s, '2018 JAN'~'2019 DEC')
```

```
TRN
```

```
## # A tsibble: 96 x 2 [1M]
##       Month concentration
##       <mth>         <dbl>
##  1 2010 Jan            59
##  2 2010 Feb            50
##  3 2010 Mar            61
##  4 2010 Apr            49
##  5 2010 May            56
##  6 2010 Jun            51
##  7 2010 Jul            33
##  8 2010 Aug            32
##  9 2010 Sep            25
## 10 2010 Oct            41
## # … with 86 more rows
```

```
TST
```

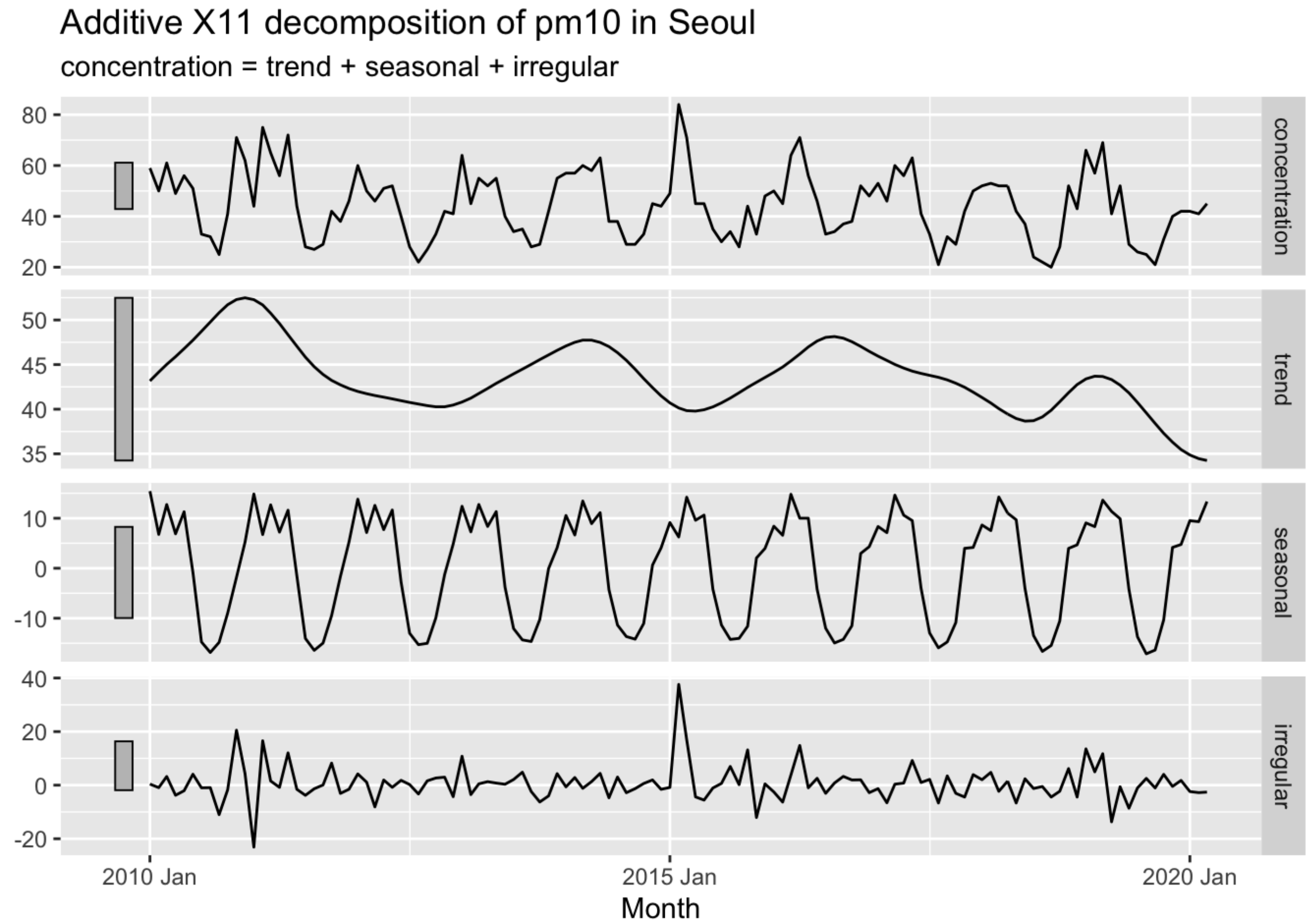
```
## # A tsibble: 24 x 2 [1M]
##       Month concentration
##       <mth>         <dbl>
##  1 2018 Jan            52
##  2 2018 Feb            53
##  3 2018 Mar            52
##  4 2018 Apr            52
##  5 2018 May            42
##  6 2018 Jun            37
##  7 2018 Jul            24
##  8 2018 Aug            22
##  9 2018 Sep            20
## 10 2018 Oct            28
## # … with 14 more rows
```

분해법 적용

(1) X11

```
x11_dcmp <- pm10s %>%
  model(x11=feasts:::X11(concentration, type="additive")) %>%
  components()
```

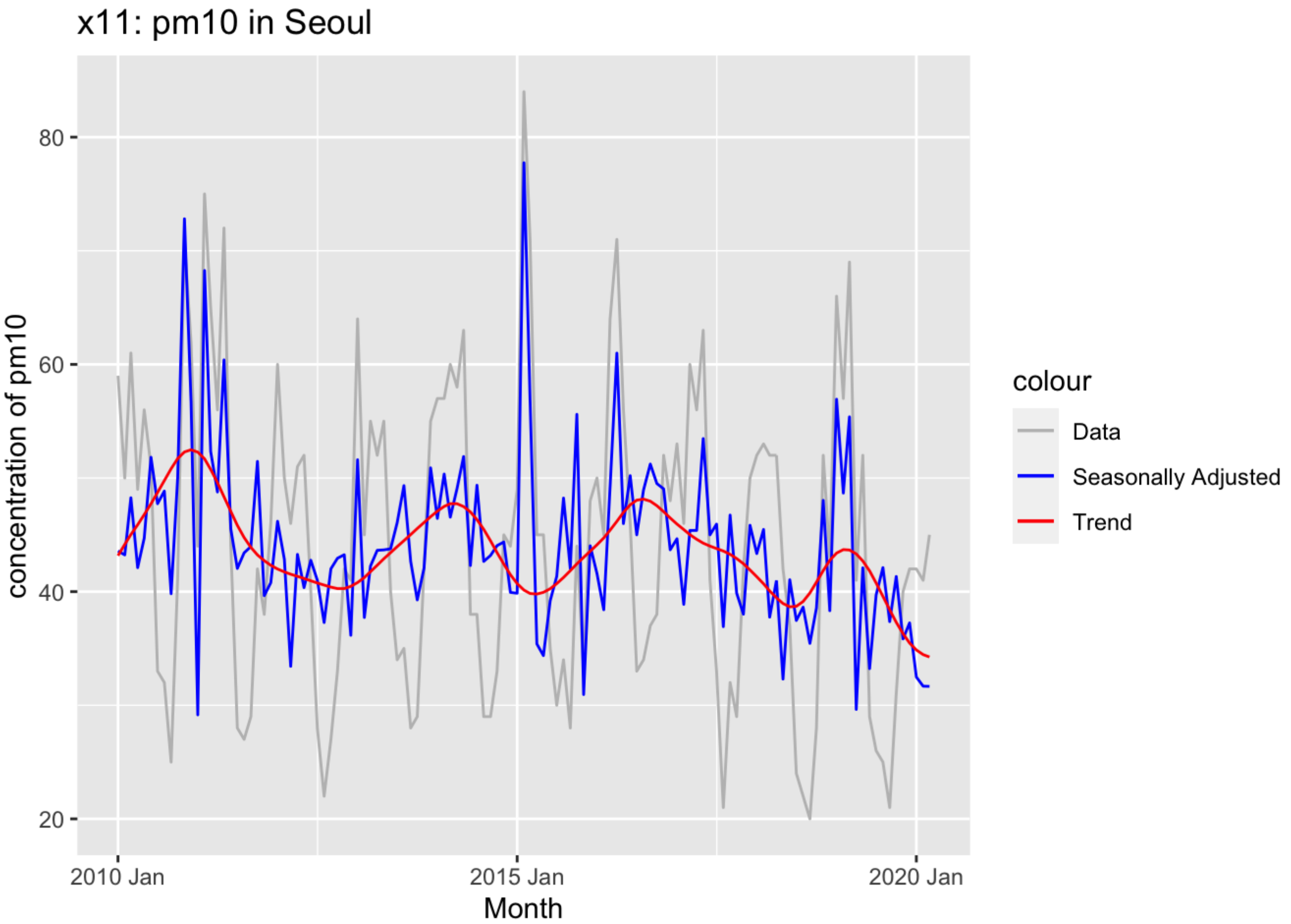
```
autoplot(x11_dcmp) + ggtitle("Additive X11 decomposition of pm10 in Seoul")
```



- 네모의 크기가 가장 작은 불규칙 성분의 영향이 가장 크다.

분해결과: trend, season_adjust

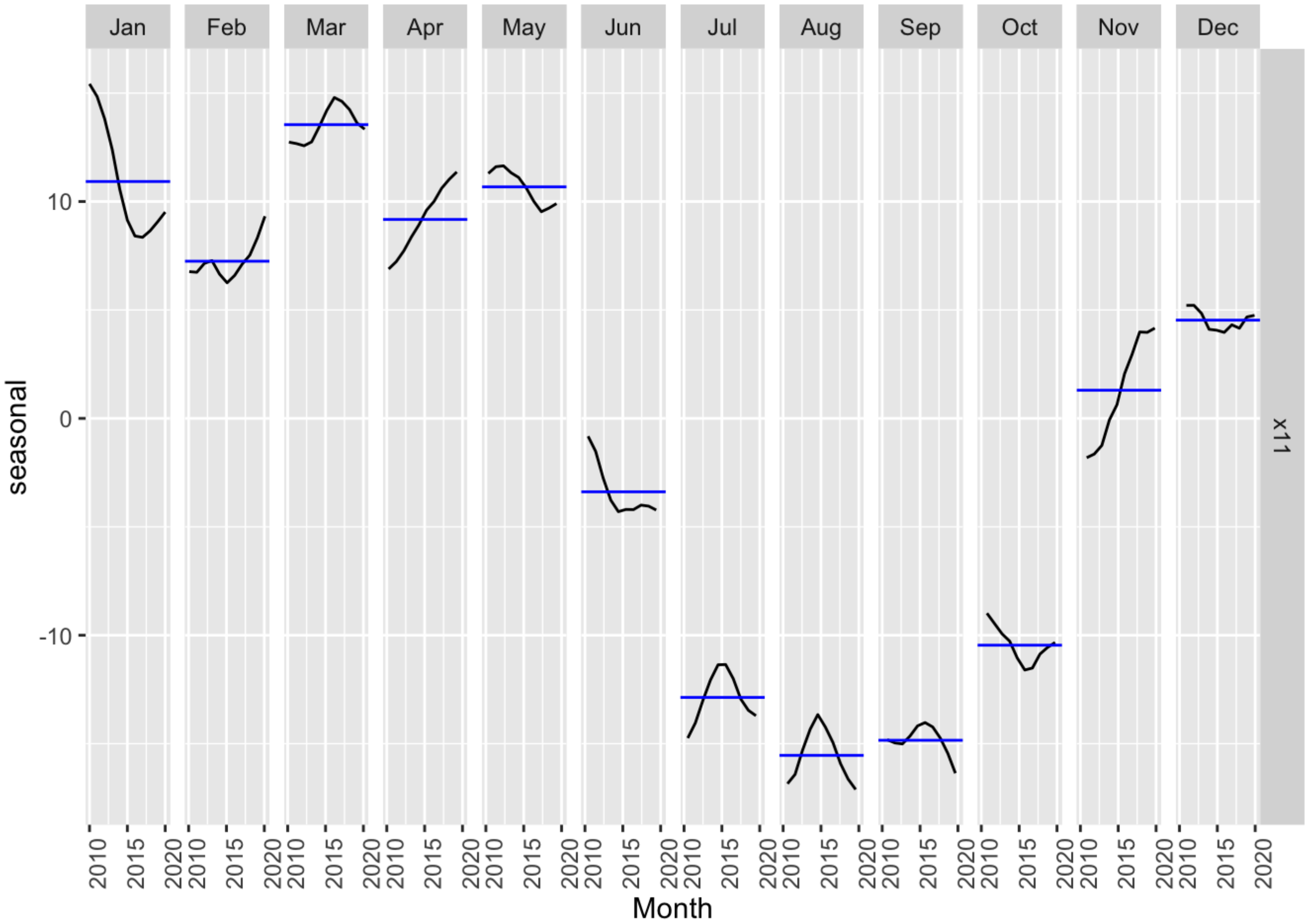
```
x11_dcmp %>%
  ggplot(aes(x = Month)) +
  geom_line(aes(y = concentration, colour = "Data")) +
  geom_line(aes(y =season_adjust, colour = "Seasonally Adjusted")) +
  geom_line(aes(y = trend, colour = "Trend")) +
  xlab("Month") + ylab("concentration of pm10") +
  ggtitle("x11: pm10 in Seoul") +
  scale_colour_manual(values=c("gray", "blue", "red"),
                      breaks=c("Data", "Seasonally Adjusted", "Trend"))
```



- 계절조정 시계열과 추세성분은 불규칙성분의 차이만 있지만 그래프가 크게 다른 것은 위에서 확인하였듯 불규칙 성분의 영향이 가장 크기 때문이다. 추세가 불명확하다.

계절성분의 시각화

```
x11_dcmp %>%
  gg_subseries(seasonal)
```

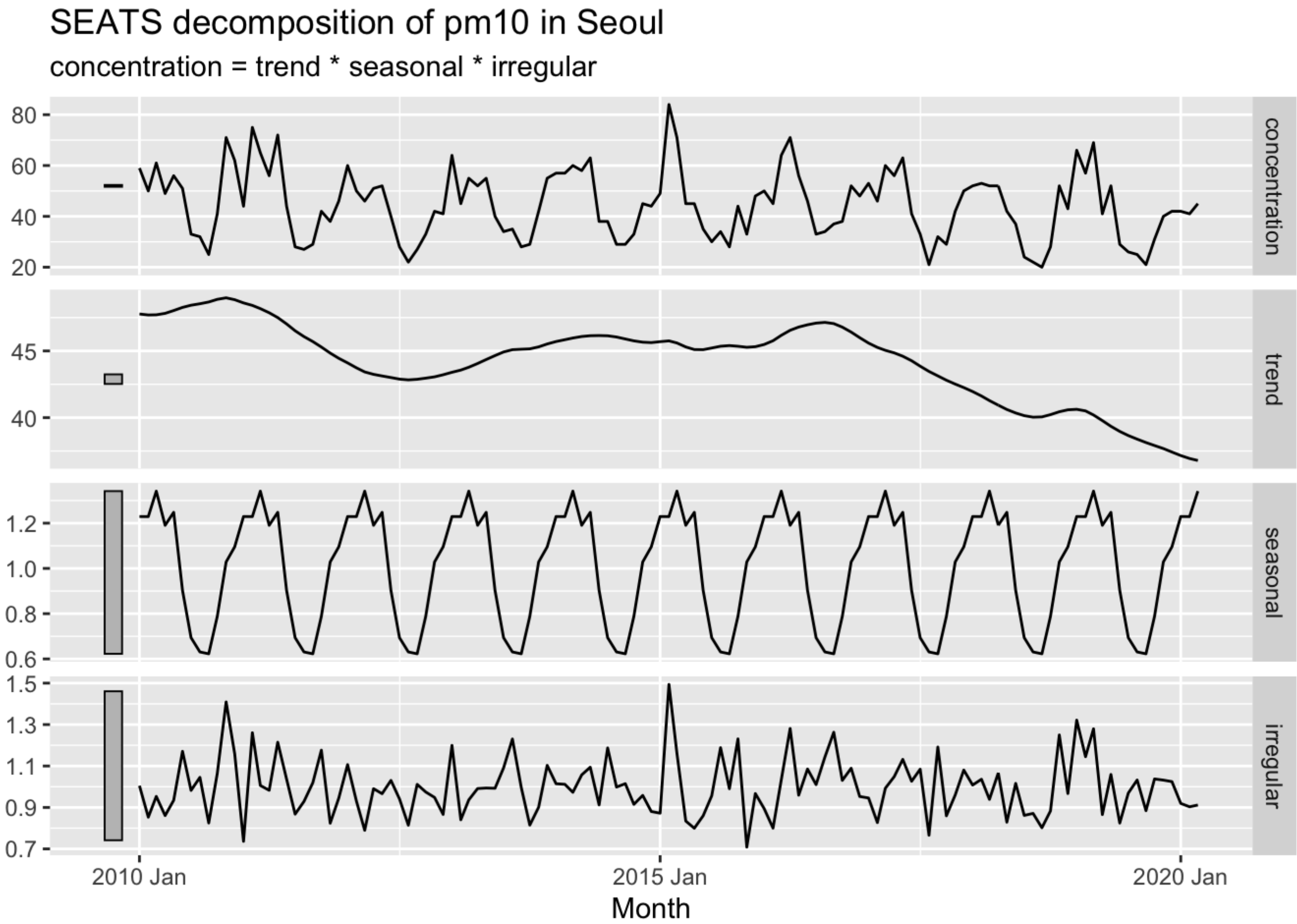


- 겨울에서 봄에 미세먼지 측정량이 높아지고 여름에는 낮아지는 뚜렷한 계절성을 가진다.
- 추세가 불명확하고 겨울에 미세먼지 측정량이 높아지는 뚜렷한 계절성이 있다.

(2) SEATS

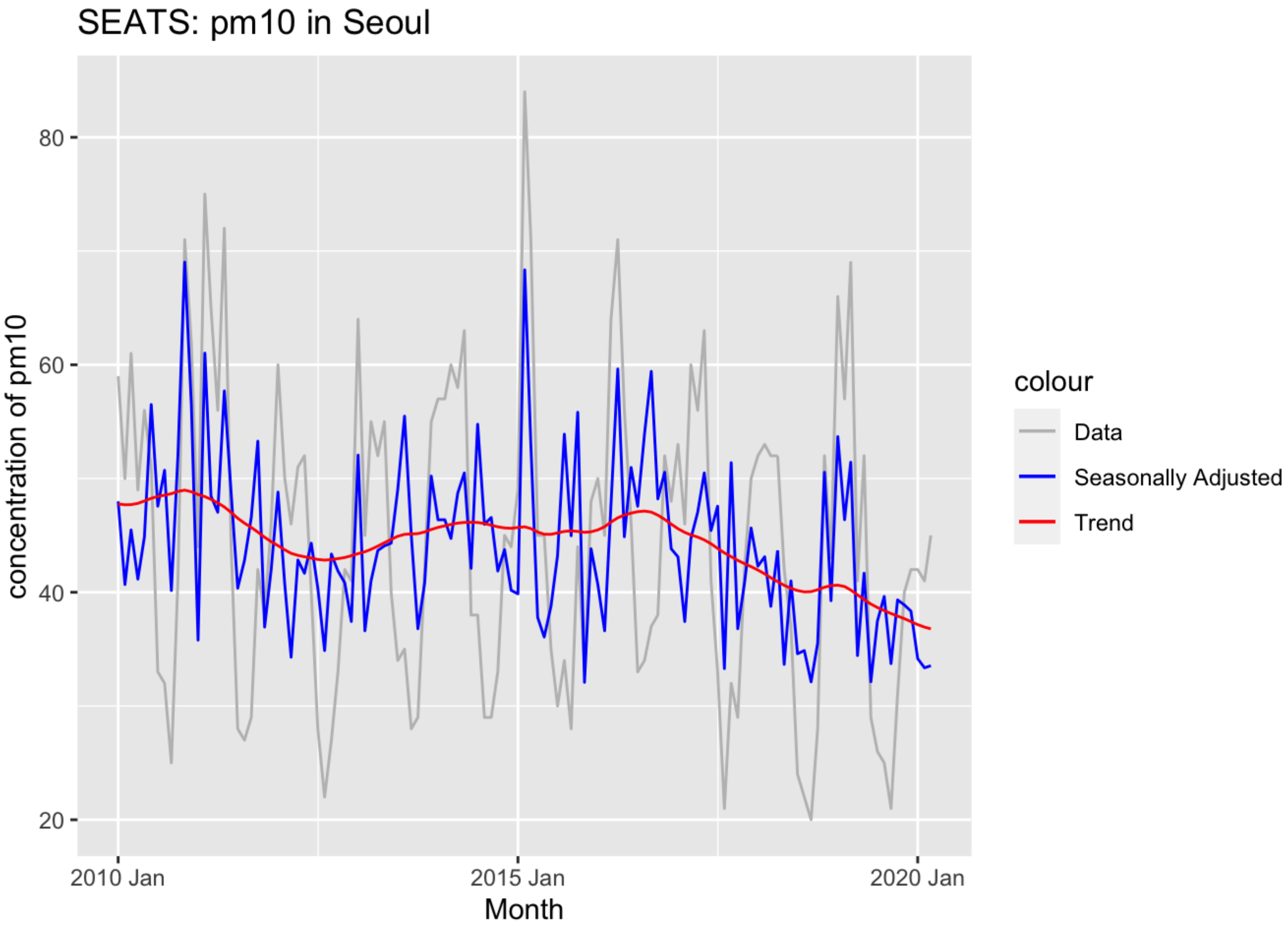
```
seats_dcmp <- pm10s %>%
  model(seats=feasts:::SEATS(concentration)) %>%
  components()
```

```
autoplot(seats_dcmp) + ggtitle("SEATS decomposition of pm10 in Seoul")
```



분해결과: trend, season_adjust

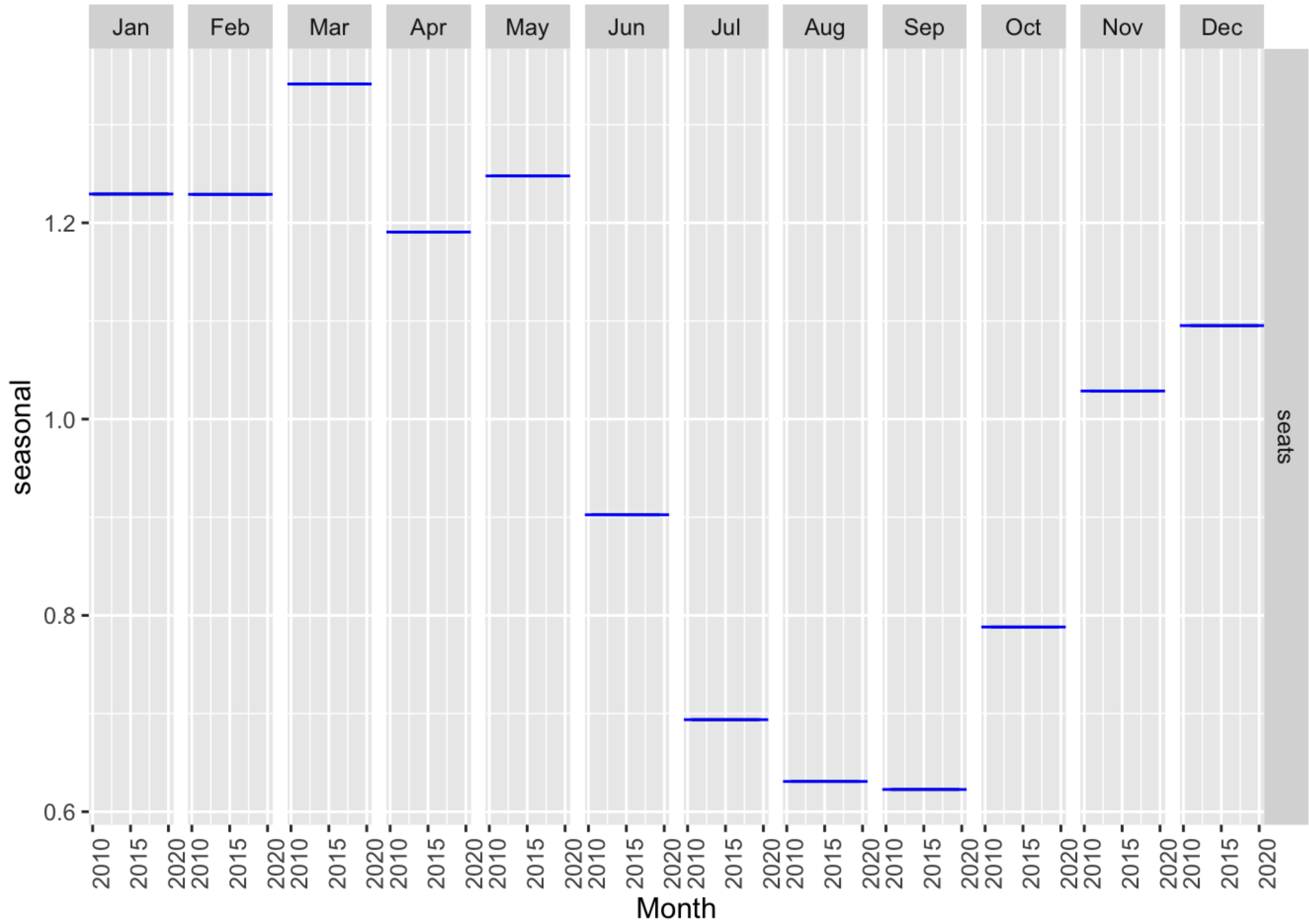
```
seats_dcmp %>%
  ggplot(aes(x = Month)) +
  geom_line(aes(y = concentration, colour = "Data")) +
  geom_line(aes(y =season_adjust, colour = "Seasonally Adjusted")) +
  geom_line(aes(y = trend, colour = "Trend")) +
  xlab("Month") + ylab("concentration of pm10") +
  ggtitle("SEATS: pm10 in Seoul") +
  scale_colour_manual(values=c("gray", "blue", "red"),
                      breaks=c("Data", "Seasonally Adjusted", "Trend"))
```



- 추세가 떨어지는 것처럼 보이나 불명확하다.

계절성분의 시각화

```
seats_dcmp %>%
  gg_subseries(seasonal)
```

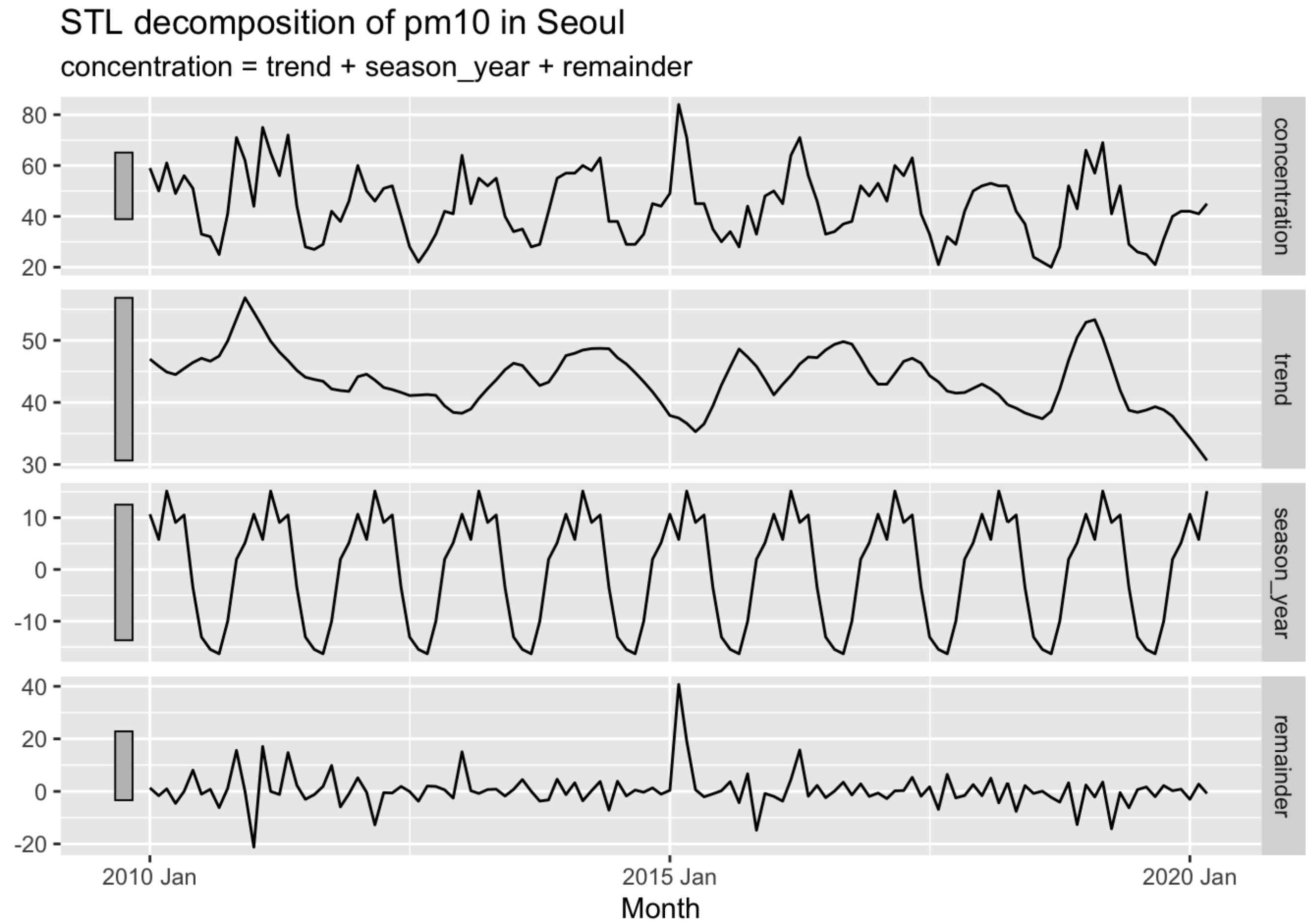


- 겨울에서 봄에 미세먼지 측정량이 높아지고 여름에 떨어지는 뚜렷한 계절성이 있다.
- 추세가 떨어지는 것처럼 보이나 불명확하고 겨울에 pm10의 측정량이 높아지는 뚜렷한 계절성을 볼 수 있다.

(3) STL

```
stl_dcmp <- pm10s %>%
  model(STL(concentration~trend(window=7) + season(window='periodic'), robust=TRUE)) %>%
  components()
```

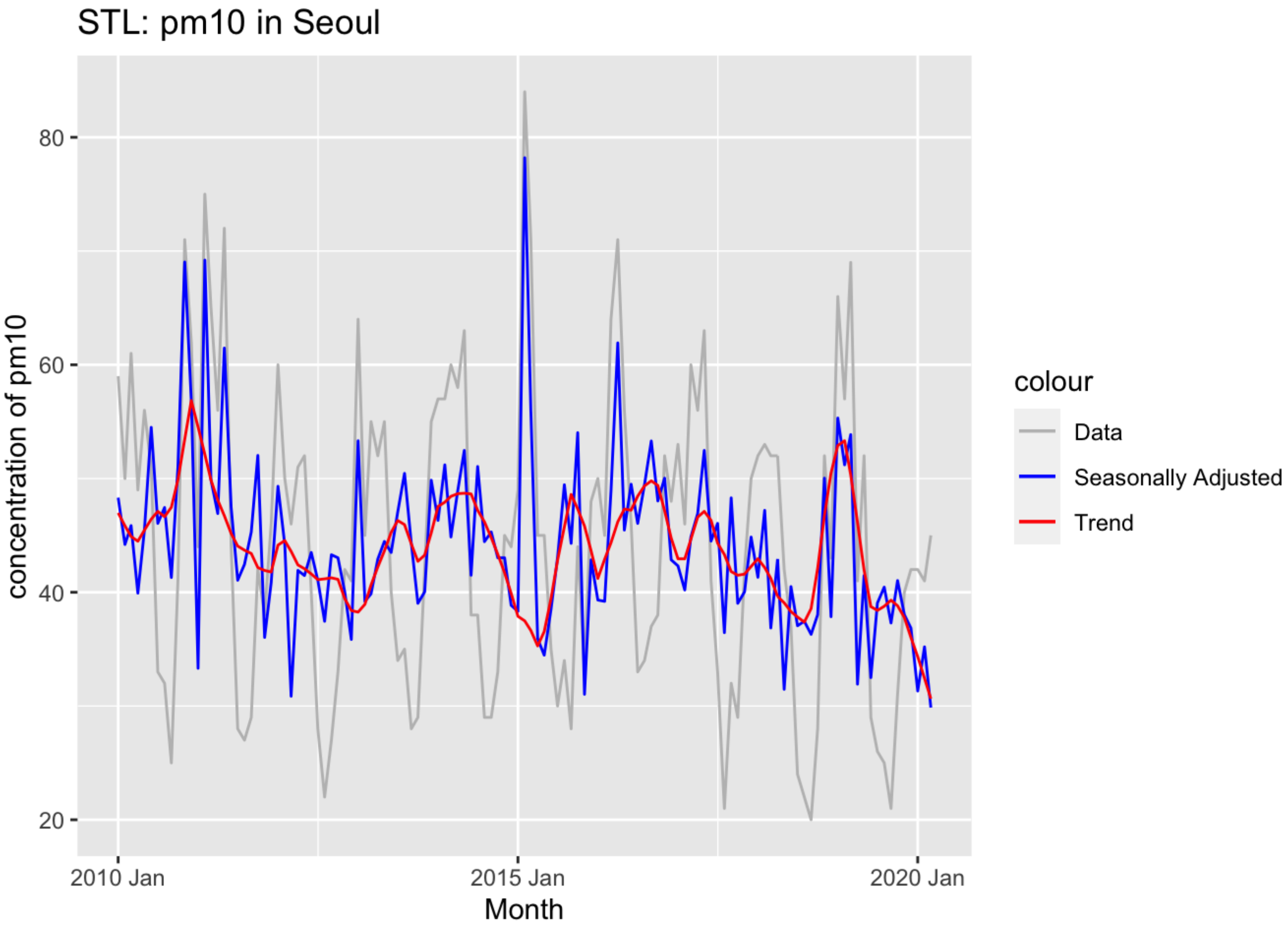
```
autoplot(stl_dcmp) + ggtitle("STL decomposition of pm10 in Seoul")
```



- 뚜렷한 계절성을 확인할 수 있다.

분해결과: trend, season_adjust

```
stl_dcmp %>%
  ggplot(aes(x = Month)) +
  geom_line(aes(y = concentration, colour = "Data")) +
  geom_line(aes(y =season_adjust, colour = "Seasonally Adjusted")) +
  geom_line(aes(y = trend, colour = "Trend")) +
  xlab("Month") + ylab("concentration of pm10") +
  ggtitle("STL: pm10 in Seoul") +
  scale_colour_manual(values=c("gray", "blue", "red"),
                     breaks=c("Data", "Seasonally Adjusted", "Trend"))
```



- 추세가 불명확하고 뚜렷한 계절성을 가진다.

단순예측법 적용

한 번에 예측

```
Mpm10s <- model(TRN,
                mn = MEAN(concentration),
                rw = NAIVE(concentration),
                rwd = RW(concentration~drift()),
                srw = SNAIVE(concentration))
Mpm10s
```

```
## # A mable: 1 x 4
##        mn      rw          rwd      srw
##   <model> <model>      <model>  <model>
## 1  <MEAN> <NAIVE> <RW w/ drift> <SNAIVE>
```

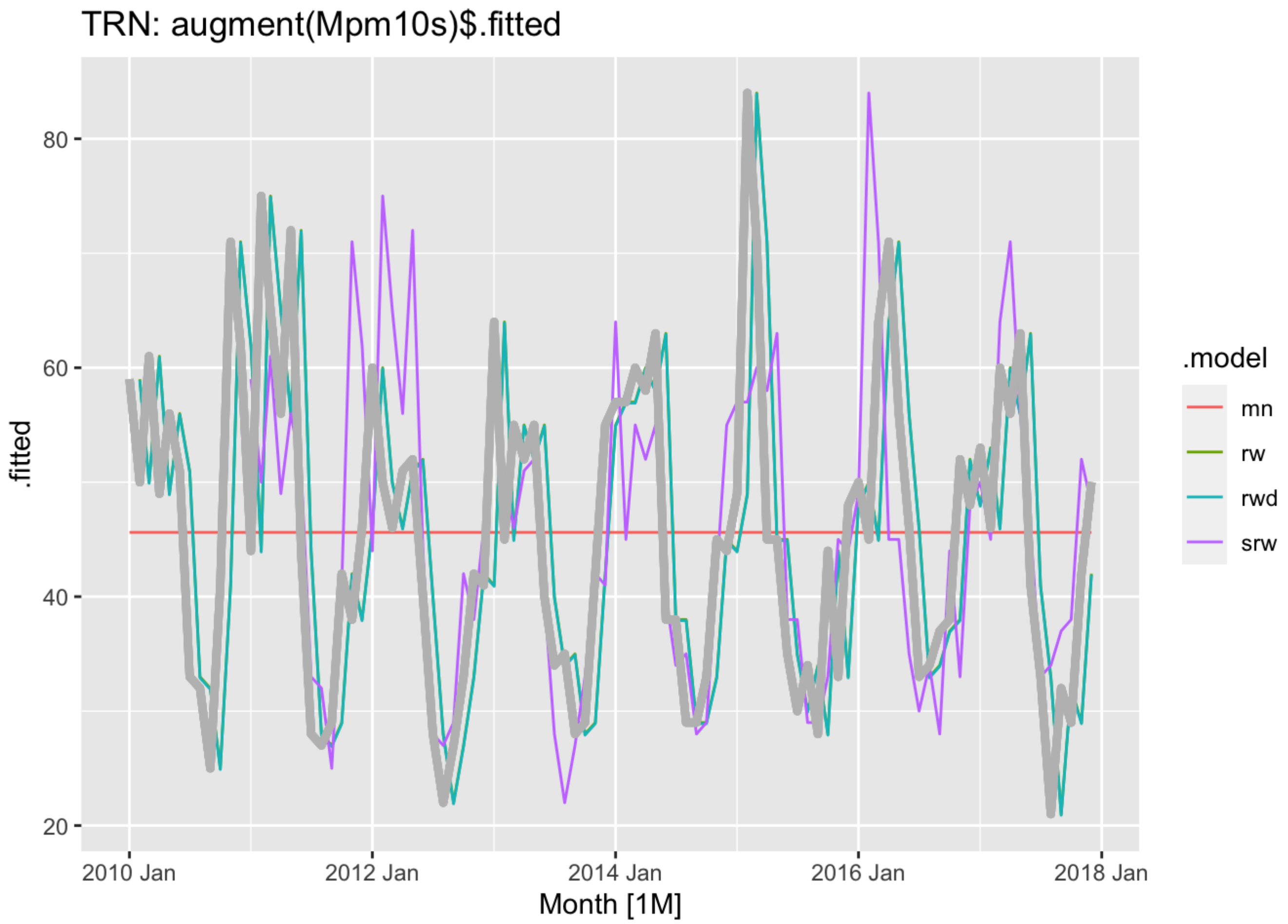
```
Apm10s <- augment(Mpm10s)
Apm10s
```

```
## # A tsibble: 384 x 6 [1M]
## # Key:       .model [4]
##    .model    Month concentration .fitted .resid .innov
##    <chr>     <mth>         <dbl>   <dbl>  <dbl>  <dbl>
##  1 mn     2010 Jan            59    45.6  13.4   13.4
##  2 mn     2010 Feb            50    45.6   4.39   4.39
##  3 mn     2010 Mar            61    45.6  15.4   15.4
##  4 mn     2010 Apr            49    45.6   3.39   3.39
##  5 mn     2010 May            56    45.6  10.4   10.4
##  6 mn     2010 Jun            51    45.6   5.39   5.39
##  7 mn     2010 Jul            33    45.6 -12.6  -12.6
##  8 mn     2010 Aug            32    45.6 -13.6  -13.6
##  9 mn     2010 Sep            25    45.6 -20.6  -20.6
## 10 mn     2010 Oct            41    45.6  -4.61  -4.61
## # … with 374 more rows
```

예측값 그림

```
autoplot(Apm10s, .fitted) +
  autolayer(Apm10s, concentration, color = 'gray', size = 1.5) +
  ggtitle('TRN: augment(Mpm10s)$.fitted')
```

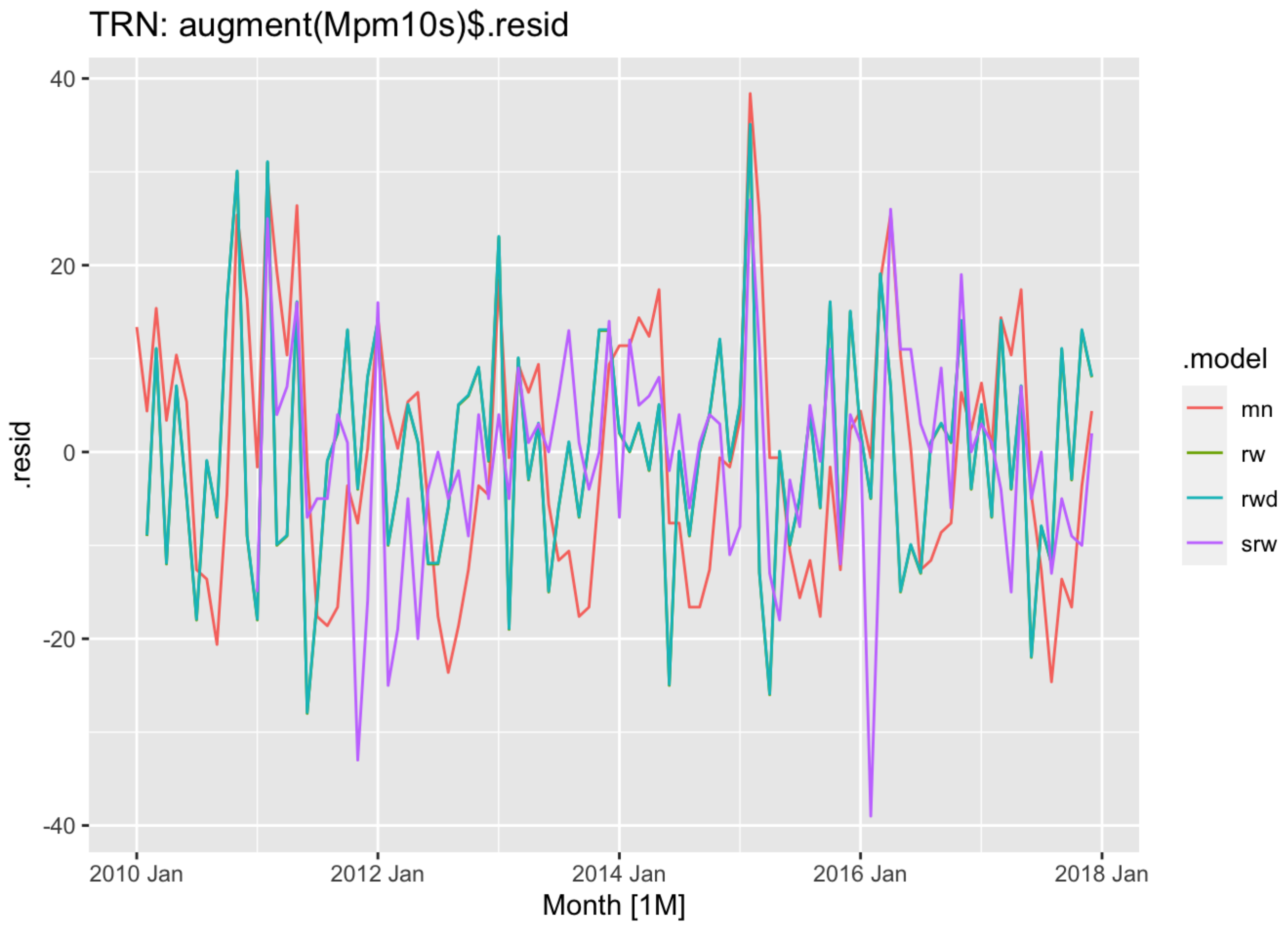
```
## Warning: Removed 14 row(s) containing missing values (geom_path).
```



잔차 그림

```
autoplot(Apm10s, .resid) +
  ggtitle('TRN: augment(Mpm10s)$.resid')
```

```
## Warning: Removed 14 row(s) containing missing values (geom_path).
```

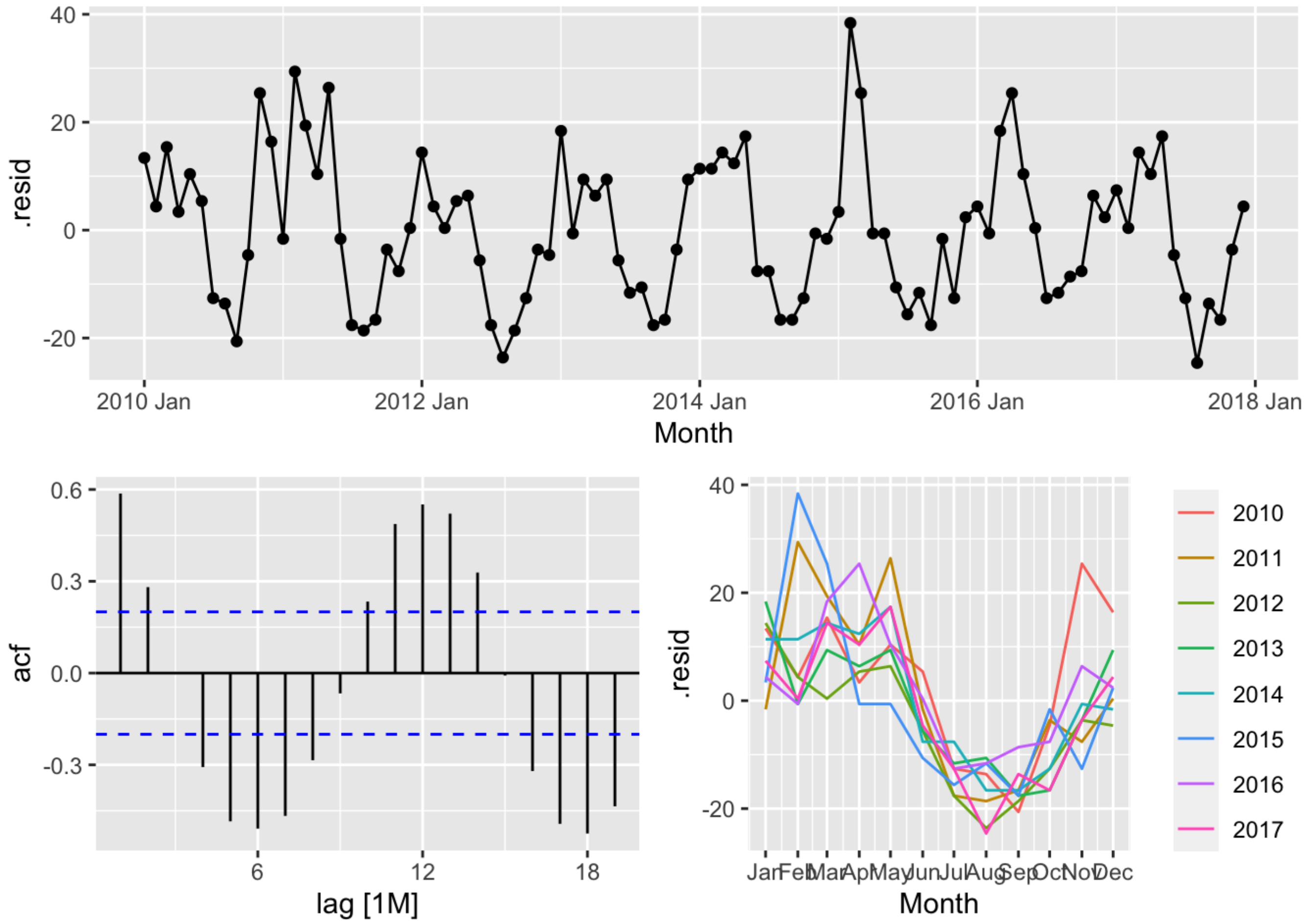


한 모델씩 뽑아 잔차 그림 보기

(1) MEAN

```
gg_tsdisplay(filter(Apm10s, .model=='mn') %>% select(.resid))
```

```
## Plot variable not specified, automatically selected `y = .resid`
```



(2) Random Walk

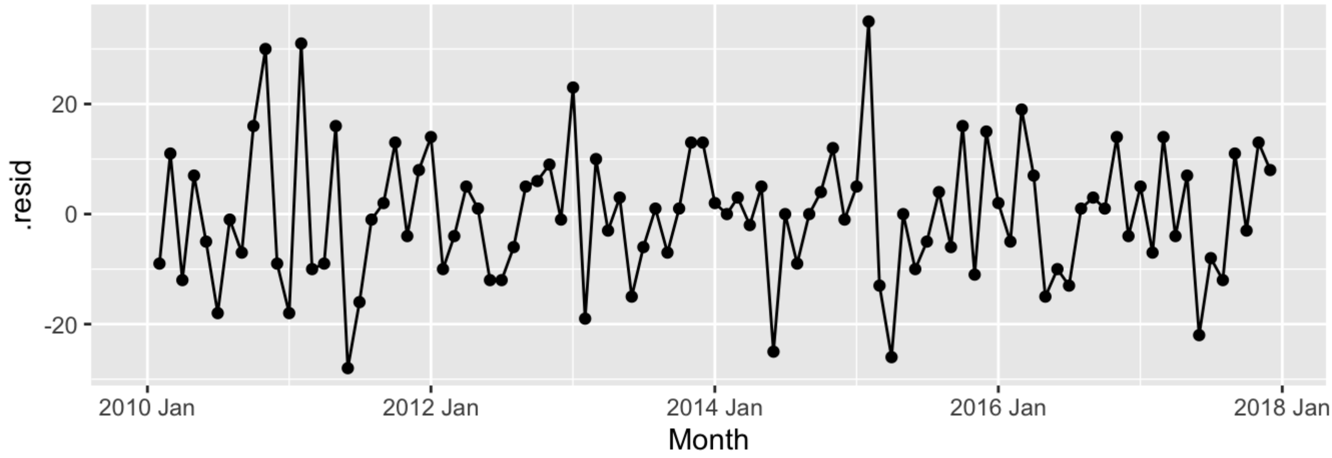
```
gg_tsdisplay(filter(Apm10s, .model=='rw') %>% select(.resid))
```

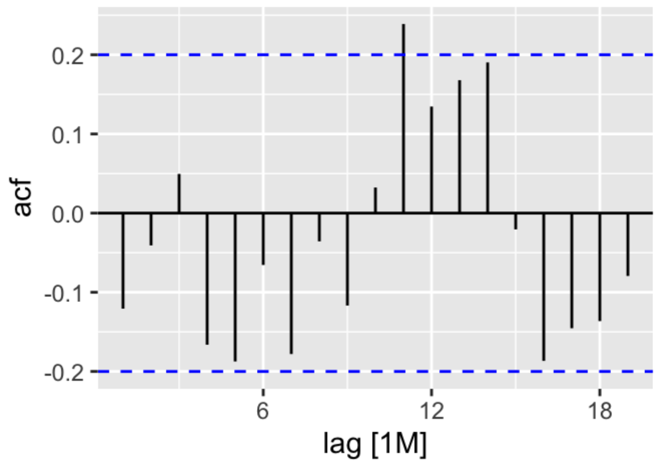
```
## Plot variable not specified, automatically selected `y = .resid`
```

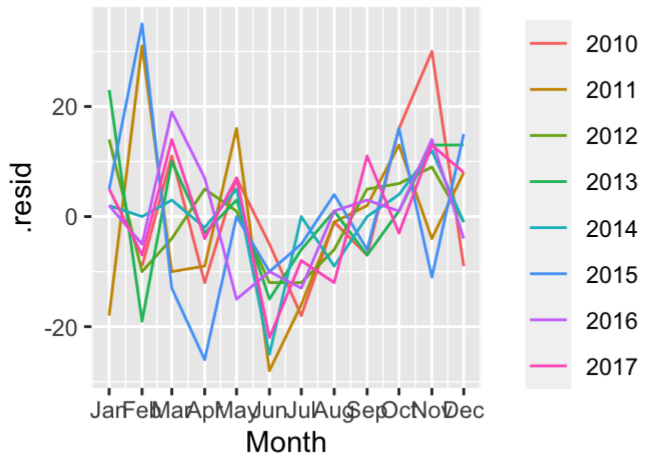
```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```







(3) Random Walk with Drift

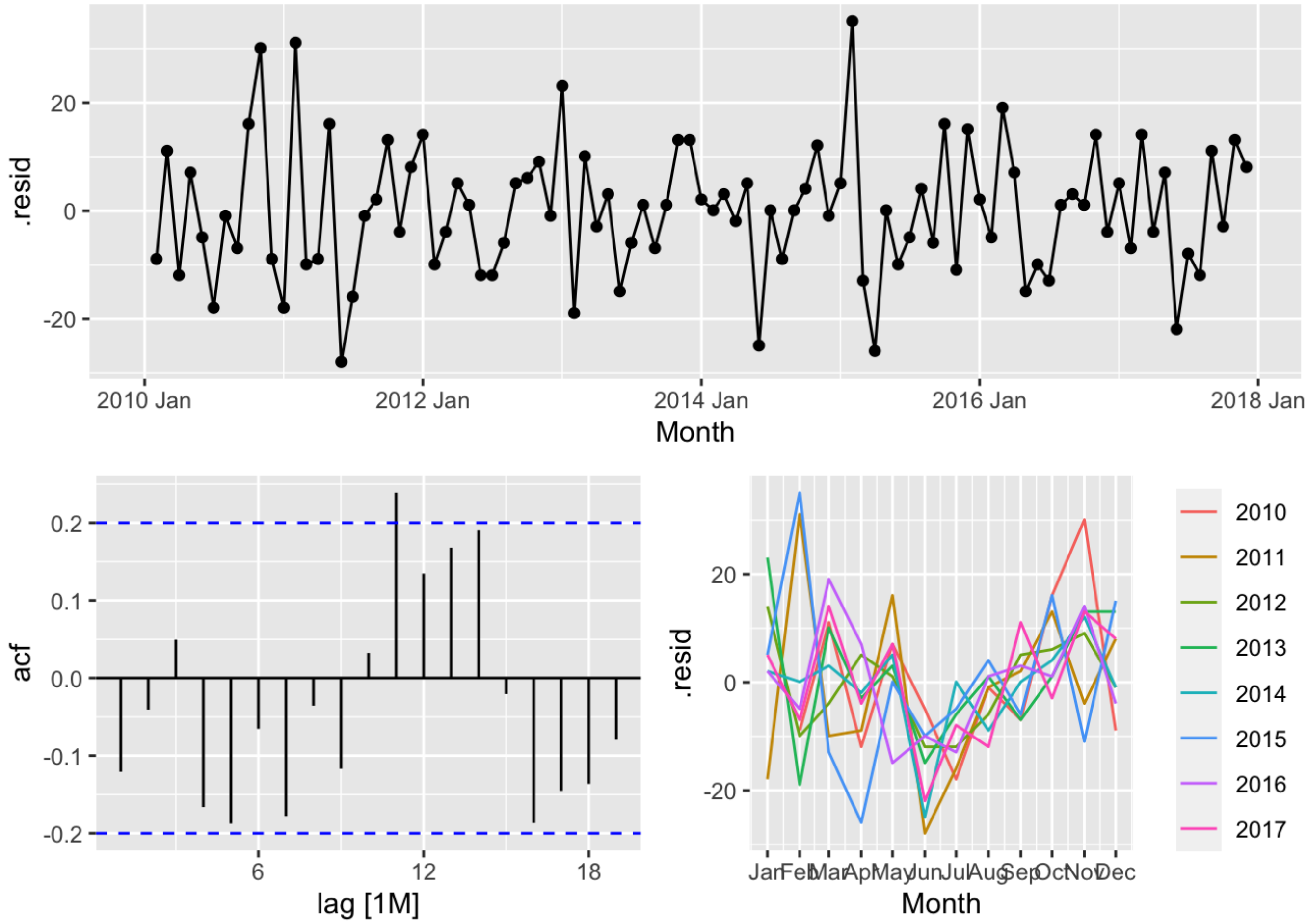
```
gg_tsdisplay(filter(Apm10s, .model=='rwd') %>% select(.resid))
```

```
## Plot variable not specified, automatically selected `y = .resid`
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



(4) Seosnal Random Walk

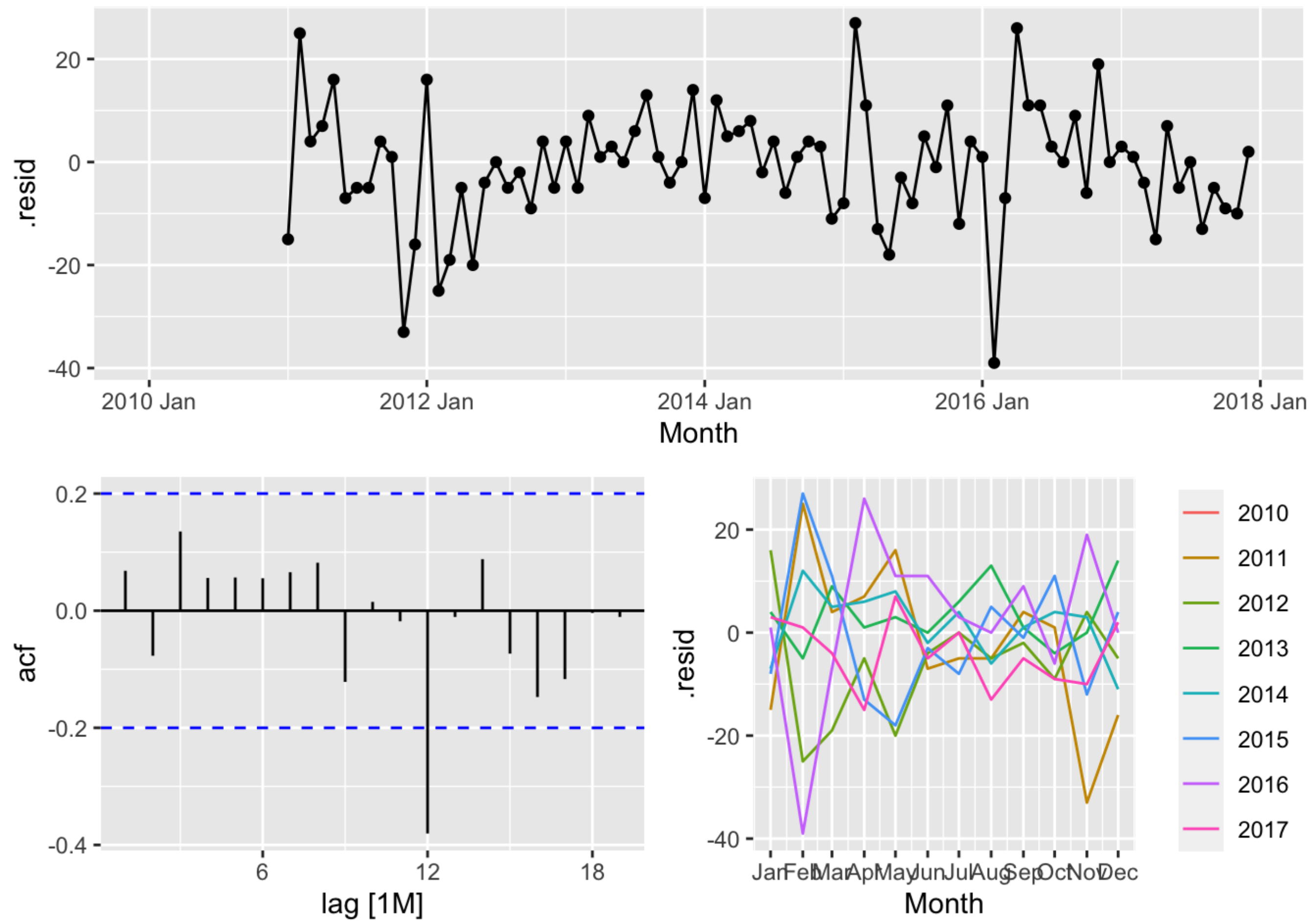
```
gg_tsdisplay(filter(Apm10s, .model=='srw') %>% select(.resid))
```

```
## Plot variable not specified, automatically selected `y = .resid`
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 12 rows containing missing values (geom_point).
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```



model별 백색잡음 검정

```
features(Apm10s, .resid, ljung_box, lag=4, dof=0)
```

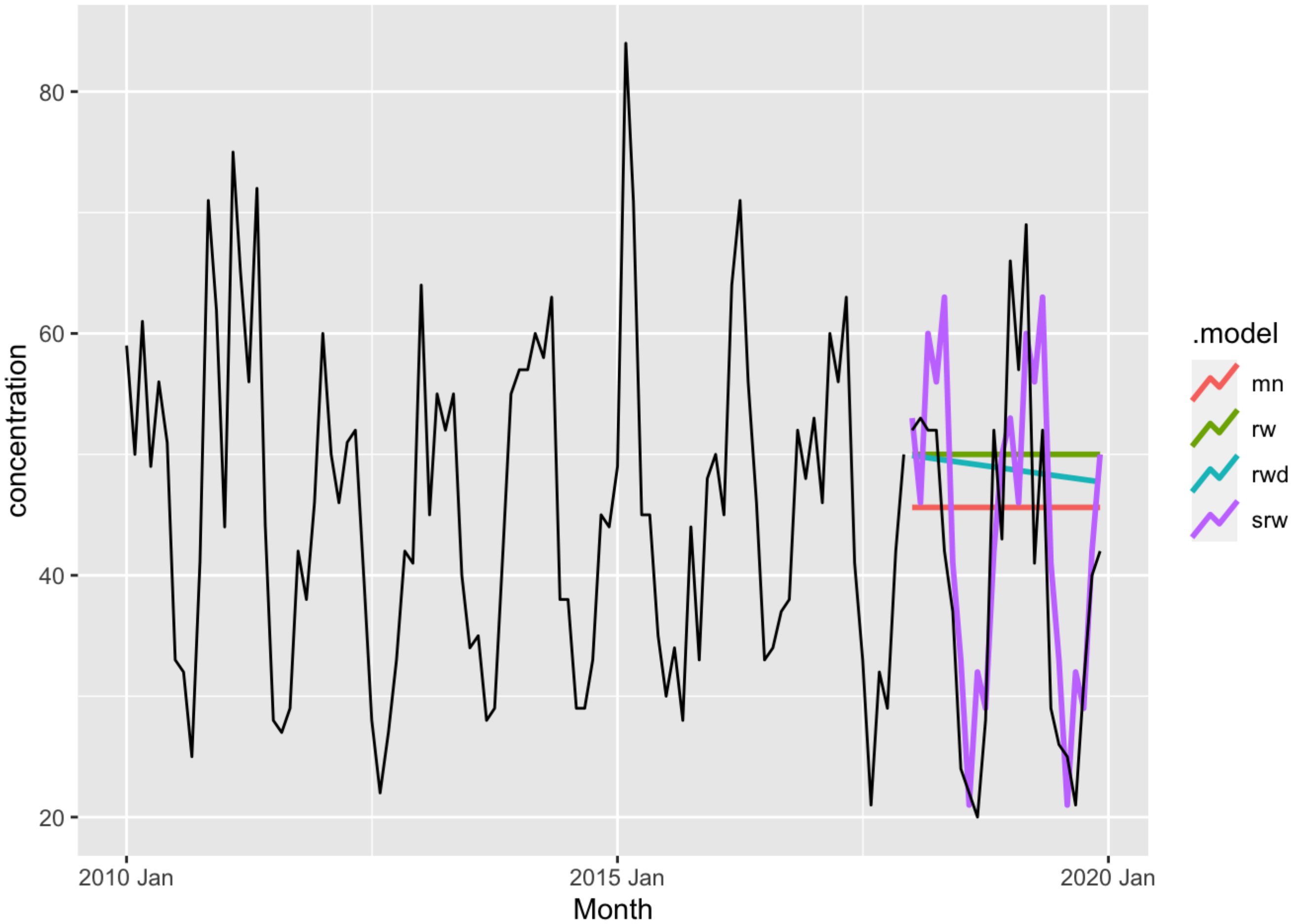
```
## # A tibble: 4 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 mn       51.6   1.65e-10
## 2 rw        4.64  3.27e- 1
## 3 rwd       4.64  3.27e- 1
## 4 srw       2.84  5.85e- 1
```

성능 평가

```
Fpm10s <- forecast(Mpm10s, h=24)
Fpm10s
```

```
## # A fable: 96 x 4 [1M]
## # Key:     .model [4]
##    .model    Month concentration .mean
##    <chr>     <mth>        <dist> <dbl>
##  1 mn     2018 Jan    N(46, 181)  45.6
##  2 mn     2018 Feb    N(46, 181)  45.6
##  3 mn     2018 Mar    N(46, 181)  45.6
##  4 mn     2018 Apr    N(46, 181)  45.6
##  5 mn     2018 May    N(46, 181)  45.6
##  6 mn     2018 Jun    N(46, 181)  45.6
##  7 mn     2018 Jul    N(46, 181)  45.6
##  8 mn     2018 Aug    N(46, 181)  45.6
##  9 mn     2018 Sep    N(46, 181)  45.6
## 10 mn     2018 Oct    N(46, 181)  45.6
## # … with 86 more rows
```

```
autoplot(Fpm10s, TRN, level=NULL, size=1) +
  autolayer(TST, concentration)
```



TRN에 대한 평가

```
as.data.frame(accuracy(Mpm10s))
```

```
##   .model    .type             ME     RMSE       MAE       MPE     MAPE      MASE
## 1     mn Training -2.368688e-15 13.30051 10.882161 -9.328597 26.79283 1.291104
## 2     rw Training -9.473684e-02 12.06954  9.442105 -3.504211 21.19058 1.120250
## 3    rwd Training -1.498984e-16 12.06916  9.445097 -3.276447 21.17628 1.120605
## 4    srw Training -7.619048e-01 11.44552  8.428571 -4.414184 18.61234 1.000000
##          ACF1
## 1  0.58671564
## 2 -0.12066405
## 3 -0.12066405
## 4  0.06814661
```

TST에 대한 평가

```
accuracy(Fpm10s, data=pm10s)
```

```
## # A tibble: 4 x 9
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE   ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 mn     Test  -4.95 15.0   12.8  -28.3  41.7 1.52  0.631
## 2 rw     Test  -9.33 17.0   13.9  -40.7  48.1 1.65  0.631
## 3 rwd    Test  -8.15 16.2   13.4  -37.2  45.7 1.59  0.626
## 4 srw    Test  -3.17  9.31   7.92 -12.3  21.5 0.939 0.0960
```

- 최종 모형은 TST에 대한 평가에서 MAPE, RMSE, MAE 모두 가장 낮게 나온 srw 모델이다.

최종 모형의 예측값 그리기

최종 모형: srw (Seasonal Random Walk) 모델

```
MSRW <- model(TRN, SNAIVE(concentration))
MSRW
```

```
## # A mable: 1 x 1
##   `SNAIVE(concentration)`
##                   <model>
## 1                <SNAIVE>
```

예측값 생성

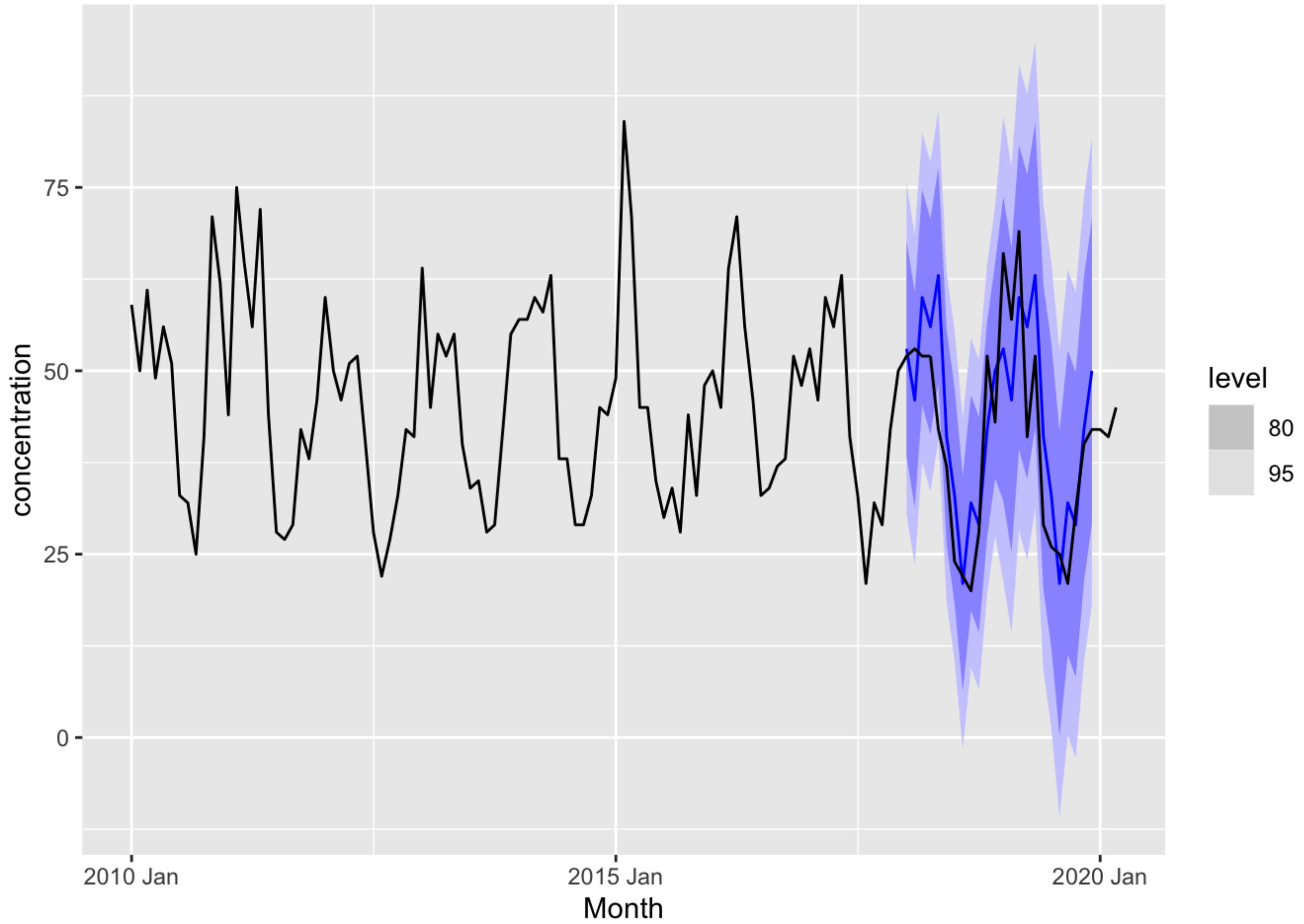
```
FSRW <- forecast(MSRW, h='2 years')
FSRW
```

```
## # A fable: 24 x 4 [1M]
## # Key:     .model [1]
##    .model                  Month concentration .mean
##    <chr>                   <mth>        <dist> <dbl>
##  1 SNAIVE(concentration) 2018 Jan    N(53, 131)    53
##  2 SNAIVE(concentration) 2018 Feb    N(46, 131)    46
##  3 SNAIVE(concentration) 2018 Mar    N(60, 131)    60
##  4 SNAIVE(concentration) 2018 Apr    N(56, 131)    56
##  5 SNAIVE(concentration) 2018 May    N(63, 131)    63
##  6 SNAIVE(concentration) 2018 Jun    N(41, 131)    41
##  7 SNAIVE(concentration) 2018 Jul    N(33, 131)    33
##  8 SNAIVE(concentration) 2018 Aug    N(21, 131)    21
##  9 SNAIVE(concentration) 2018 Sep    N(32, 131)    32
## 10 SNAIVE(concentration) 2018 Oct    N(29, 131)    29
## # … with 14 more rows
```

```
hilo(FSRW)
```

```
## # A tsibble: 24 x 6 [1M]
## # Key:       .model [1]
##    .model    Month concentration .mean                   `80%`
##    <chr>     <mth>        <dist> <dbl>                  <hilo>
##  1 SNAIV… 2018 Jan    N(53, 131)    53 [38.331972, 67.66803]80
##  2 SNAIV… 2018 Feb    N(46, 131)    46 [31.331972, 60.66803]80
##  3 SNAIV… 2018 Mar    N(60, 131)    60 [45.331972, 74.66803]80
##  4 SNAIV… 2018 Apr    N(56, 131)    56 [41.331972, 70.66803]80
##  5 SNAIV… 2018 May    N(63, 131)    63 [48.331972, 77.66803]80
##  6 SNAIV… 2018 Jun    N(41, 131)    41 [26.331972, 55.66803]80
##  7 SNAIV… 2018 Jul    N(33, 131)    33 [18.331972, 47.66803]80
##  8 SNAIV… 2018 Aug    N(21, 131)    21 [ 6.331972, 35.66803]80
##  9 SNAIV… 2018 Sep    N(32, 131)    32 [17.331972, 46.66803]80
## 10 SNAIV… 2018 Oct    N(29, 131)    29 [14.331972, 43.66803]80
## # … with 14 more rows, and 1 more variable: `95%` <hilo>
```

```
autoplot(FSRW, pm10s)
```



잔차 검토

```
ASRW <- augment(MSRW)
ASRW
```

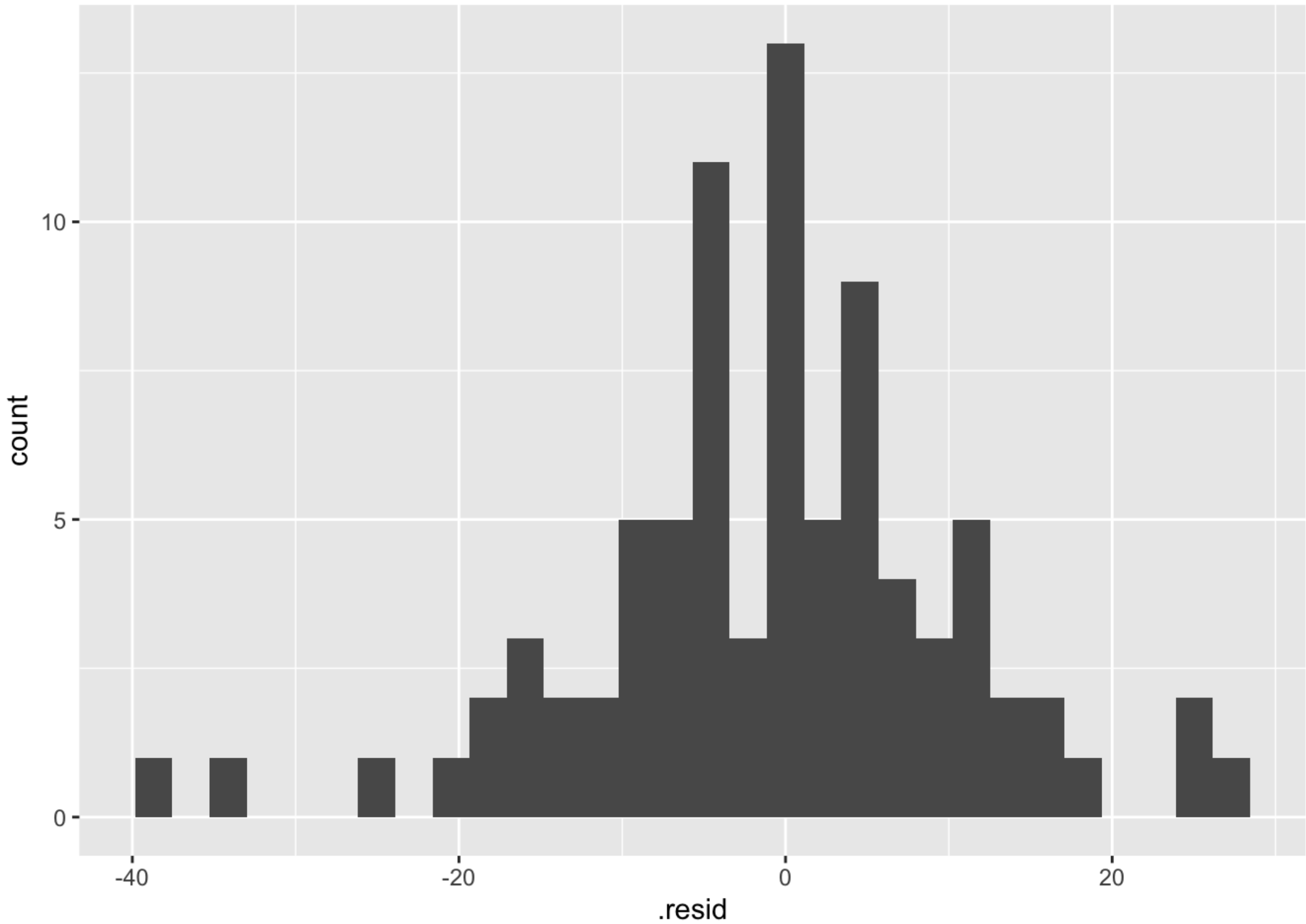
```
## # A tsibble: 96 x 6 [1M]
## # Key:       .model [1]
##    .model                  Month concentration .fitted .resid .innov
##    <chr>                   <mth>         <dbl>   <dbl>  <dbl>  <dbl>
##  1 SNAIVE(concentration) 2010 Jan            59      NA     NA     NA
##  2 SNAIVE(concentration) 2010 Feb            50      NA     NA     NA
##  3 SNAIVE(concentration) 2010 Mar            61      NA     NA     NA
##  4 SNAIVE(concentration) 2010 Apr            49      NA     NA     NA
##  5 SNAIVE(concentration) 2010 May            56      NA     NA     NA
##  6 SNAIVE(concentration) 2010 Jun            51      NA     NA     NA
##  7 SNAIVE(concentration) 2010 Jul            33      NA     NA     NA
##  8 SNAIVE(concentration) 2010 Aug            32      NA     NA     NA
##  9 SNAIVE(concentration) 2010 Sep            25      NA     NA     NA
## 10 SNAIVE(concentration) 2010 Oct            41      NA     NA     NA
## # … with 86 more rows
```

잔차의 히스토그램

```
ggplot(ASRW, aes(x=.resid)) + geom_histogram()
```

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

```
## Warning: Removed 12 rows containing non-finite values (stat_bin).
```

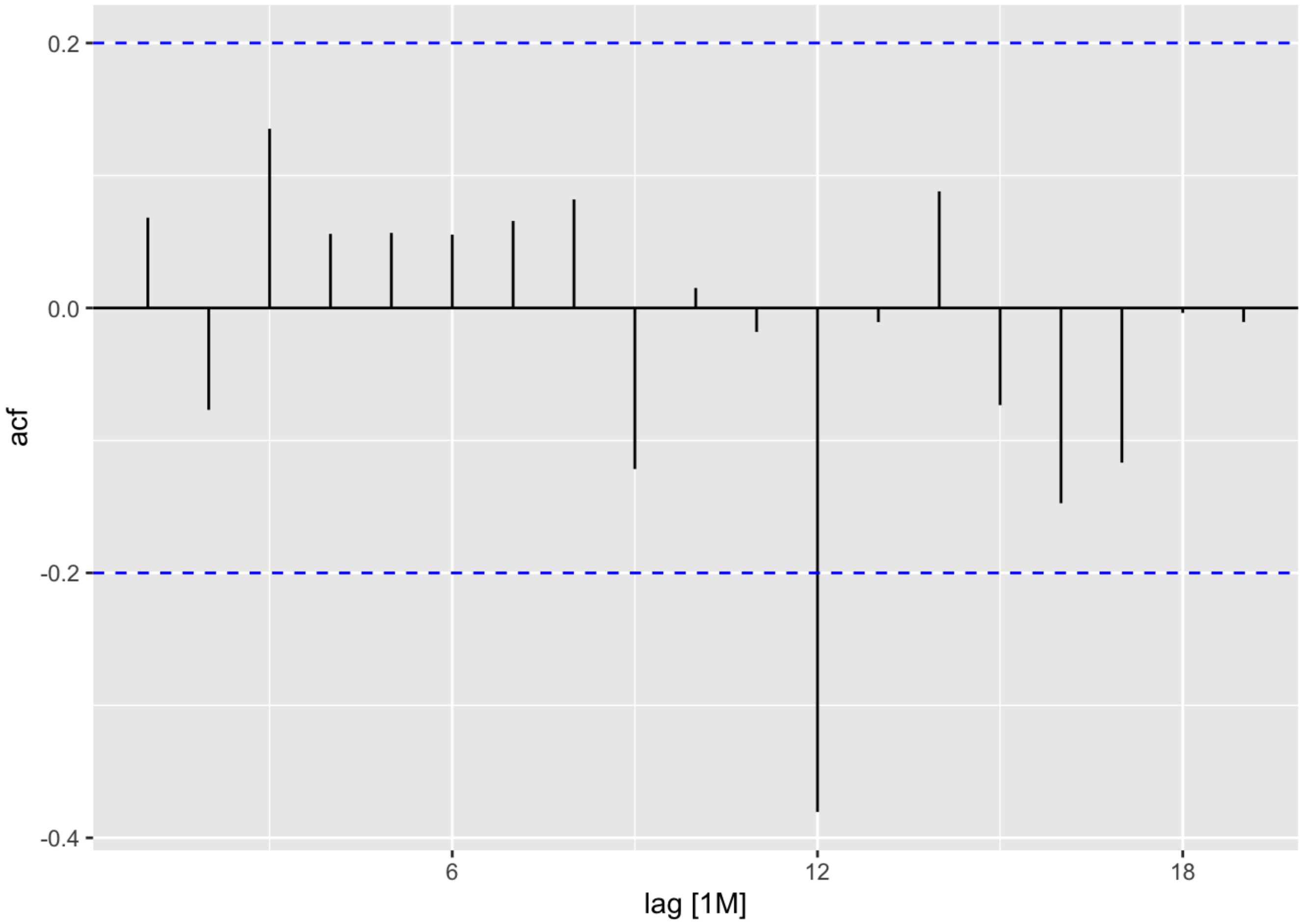


잔차의 ACF

```
ACF(ASRW, .resid)
```

```
## # A tsibble: 19 x 3 [1M]
## # Key:       .model [1]
##    .model                  lag      acf
##    <chr>                 <lag>    <dbl>
##  1 SNAIVE(concentration)    1M  0.0681
##  2 SNAIVE(concentration)    2M -0.0769
##  3 SNAIVE(concentration)    3M  0.135
##  4 SNAIVE(concentration)    4M  0.0559
##  5 SNAIVE(concentration)    5M  0.0567
##  6 SNAIVE(concentration)    6M  0.0554
##  7 SNAIVE(concentration)    7M  0.0657
##  8 SNAIVE(concentration)    8M  0.0819
##  9 SNAIVE(concentration)    9M -0.122
## 10 SNAIVE(concentration)   10M  0.0151
## 11 SNAIVE(concentration)   11M -0.0181
## 12 SNAIVE(concentration)   12M -0.380
## 13 SNAIVE(concentration)   13M -0.0106
## 14 SNAIVE(concentration)   14M  0.0880
## 15 SNAIVE(concentration)   15M -0.0733
## 16 SNAIVE(concentration)   16M -0.147
## 17 SNAIVE(concentration)   17M -0.117
## 18 SNAIVE(concentration)   18M -0.00374
## 19 SNAIVE(concentration)   19M -0.0106
```

```
autoplot(ACF(ASRW, .resid))
```



- ACF 그림을 보면 12차에서만 기각역을 벗어난 것을 확인할 수 있다. 하지만 제 1종 오류의 가능성이 있으므로 백색잡음 검정을 통해 다시 확인한다.

잔차의 백색잡음

```
# 잔차에 대한 Ljung_box 검정
features(ASRW, .resid, ljung_box, lag=10, dof=0)
```

```
## # A tibble: 1 x 3
##   .model                lb_stat lb_pvalue
##   <chr>                   <dbl>     <dbl>
## 1 SNAIVE(concentration)    5.90     0.823
```

- 10차까지 검정한 결과 p-value가 0.05보다 크므로 ljung-box의 가설 H0를 기각하지 못한다. 따라서 더 이상 남은 정보가 없다는 것을 알 수 있고, SNAIVE 모델로 설명이 된다고 할 수 있다.