

Chapter 17 시계열분석 v1.1

Sangkon Han(sangkon@pusan.ac.kr)

2023-07-02

Contents

날짜/시간 데이터 클래스	1
date 클래스	2
POSIXct, POSIXlt 클래스	2
yearmon, yearqtr 클래스	3
날짜, 시간 포맷	4
시계열 데이터 객체	4
ts	4
xts(extensible time-series)	4
tsibble	5
시계열 데이터 import	6
엑셀 파일	6
CSV 파일	7
시계열 데이터 시각화	7
실습 데이터 생성	7
xts 패키지	18
ts:forecast	27
tsibble: feasts 패키지	33
data.frame:timetk 패키지	38
시간 정보 추출	39
시간 연산	40
시간 반올림	41
주간, 월간 합계 및 평균	42
OHLC : 주식 시가, 고가, 저가, 종가	51
시간 롤링 : 3일 평균, 5일 합계	52
필터링 : 지난 달 데이는?	54
월별, 분기별, 연별 증감량	58
월 비중 백분율, 연 비중 백분율	61
월별, 분기별, 연별 누적 합계	63
동월별, 동분기별, 동년별 플롯	67

날짜/시간 데이터 클래스

```
library(zoo)
library(xts)
library(dplyr)
library(tsibble)
```

```
library(readxl)
library(ggplot2)
library(ggrepel)
library(forecast)
library(feasts)
library(timetk)
library(lubridate)
library(tibbletime)
```

date 클래스

```
(date <- as.Date(c('2023-07-02', '2023-07-03', '2023-07-04')))

## [1] "2023-07-02" "2023-07-03" "2023-07-04"
(date <- as.Date(c('23/07/02', '23/07/03', '23/07/04')), format = '%y/%m/%d'))

## [1] "2023-07-02" "2023-07-03" "2023-07-04"
unclass(date)

## [1] 19540 19541 19542
```

POSIXct, POSIXlt 클래스

```
# character를 POSIXct class로 변환
as.POSIXct('2023-07-02 12:34:56')

## [1] "2023-07-02 12:34:56 KST"

# POSIXct를 해제
unclass(as.POSIXct('2023-07-02 12:34:56'))

## [1] 1688268896
## attr(),"tzone"
## [1] ""

# character를 POSIXlt class로 변환
as.POSIXlt('2023-07-02 12:34:56')

## [1] "2023-07-02 12:34:56 KST"

# POSIXlt에서 1900년 이후 연도를 출력
unclass(as.POSIXlt('2023-07-02 12:34:56'))

## $sec
## [1] 56
##
## $min
## [1] 34
##
## $hour
## [1] 12
##
## $mday
## [1] 2
##
```

```

## $mon
## [1] 6
##
## $year
## [1] 123
##
## $wday
## [1] 0
##
## $yday
## [1] 182
##
## $isdst
## [1] 0
##
## $zone
## [1] "KST"
##
## $gmtoff
## [1] NA
##
## attr(),"tzone")
## [1] ""      "KST"  "KDT"
## attr(),"balanced")
## [1] TRUE

# POSIXlt에서 1900년 이후 연도를 추출
as.POSIXlt('2023-07-02 12:34:56')$year

```

```
## [1] 123
```

yearmon, yearqtr 클래스

```

# character를 yearmon class로 변환
as.yearmon("2023-07")

## [1] "Jul 2023"

# yearmon class를 해제하면 double
unclass(as.yearmon("2023-07"))

## [1] 2023.5

# 날짜가 있어도 yearmon은 연, 월까지만 인식
as.yearmon("2023-07-03")

## [1] "Jul 2023"

?as.yearmon
# character를 yearqtr class로 변환(1분기)
as.yearqtr("2023-01")

## [1] "2023 Q1"

# yearqtr class를 해제하면 double
unclass(as.yearqtr("2023-02"))

```

```

## [1] 2023.25

날짜, 시간 포맷

as.Date('07/03/2023', format = '%d/%m/%Y')

## [1] "2023-03-07"
Sys.setlocale("LC_ALL", "English")

## [1] "LC_COLLATE=English_United States.1252;LC_CTYPE=English_United States.1252;LC_MONETARY=English_U
as.Date('03July23', format = '%d%b%y')

## [1] "2023-07-03"
Sys.setlocale("LC_ALL", "Korean")

## [1] "LC_COLLATE=Korean_Korea.949;LC_CTYPE=Korean_Korea.949;LC_MONETARY=Korean_Korea.949;LC_NUMERIC=C
as.Date('037월23', format = '%d%b%y')

## [1] "2023-07-03"

```

시계열 데이터 객체

ts

```

ts(1:10, frequency = 4, start = c(2023, 2))

##      Qtr1 Qtr2 Qtr3 Qtr4
## 2023      1     2     3
## 2024      4     5     6     7
## 2025      8     9    10

```

xts(extensible time-series)

```

set.seed(42)
xts(rnorm(5), as.Date("2023-07-03") + 0:4)

## [,1]
## 2023-07-03  1.3709584
## 2023-07-04 -0.5646982
## 2023-07-05  0.3631284
## 2023-07-06  0.6328626
## 2023-07-07  0.4042683

ts <- ts(1:10, frequency = 4, start = c(1959, 2))
xts(ts) ## 오류가 나는게 정상입니다.

```

```

## Error in xts(ts): order.by requires an appropriate time-based object
as.xts(ts)

```

```

## [,1]
## 1959 Q2    1
## 1959 Q3    2
## 1959 Q4    3

```

```

## 1960 Q1      4
## 1960 Q2      5
## 1960 Q3      6
## 1960 Q4      7
## 1961 Q1      8
## 1961 Q2      9
## 1961 Q3     10

# 시계열 데이터 형태로 보이지 않음
head(ts)

```

```

##          Qtr1 Qtr2 Qtr3 Qtr4
## 1959            1     2     3
## 1960            4     5     6

# 시계열 형태로 보임
head(as.xts(ts))

```

```

##          [,1]
## 1959 Q2    1
## 1959 Q3    2
## 1959 Q4    3
## 1960 Q1    4
## 1960 Q2    5
## 1960 Q3    6

```

tsibble

```

set.seed(42)
x <- data.frame(date = as.Date('2008-01-01') + 0:9, id = 1:10, x1 = rnorm(10), x2= rep('a', 10))
as_tsibble(x, key = id, index = date)

## # A tsibble: 10 x 4 [1D]
## # Key:       id [10]
##   date       id     x1 x2
##   <date>     <int>  <dbl> <chr>
## 1 2008-01-01     1  1.37   a
## 2 2008-01-02     2 -0.565   a
## 3 2008-01-03     3  0.363   a
## 4 2008-01-04     4  0.633   a
## 5 2008-01-05     5  0.404   a
## 6 2008-01-06     6 -0.106   a
## 7 2008-01-07     7  1.51    a
## 8 2008-01-08     8 -0.0947  a
## 9 2008-01-09     9  2.02    a
## 10 2008-01-10    10 -0.0627 a

as_tsibble(x, index = date)

## # A tsibble: 10 x 4 [1D]
##   date       id     x1 x2
##   <date>     <int>  <dbl> <chr>
## 1 2008-01-01     1  1.37   a
## 2 2008-01-02     2 -0.565   a
## 3 2008-01-03     3  0.363   a
## 4 2008-01-04     4  0.633   a

```

```

## 5 2008-01-05      5  0.404  a
## 6 2008-01-06      6 -0.106  a
## 7 2008-01-07      7  1.51   a
## 8 2008-01-08      8 -0.0947 a
## 9 2008-01-09      9  2.02   a
## 10 2008-01-10     10 -0.0627 a

```

시계열 데이터 import

엑셀 파일

```

##  9 2007 Jan  8309932 541550  3829998 2063159      1841374  1347363  494011      0
## 10 2008 Jan  8187782 537822  3672207 2038611      1906978  1419486  487492      0
## # i 12 more rows
## # i 8 more variables: 특목고 <dbl>, 특성화고 <dbl>, 자율고 <dbl>,
## #   특수학교 <dbl>, 공민학교 <dbl>, 고등공민 <dbl>, 고등기술학교 <dbl>,
## #   각종학교 <dbl>

```

CSV 파일

```

Sys.getlocale()

## [1] "LC_COLLATE=English_United States.utf8;LC_CTYPE=English_United States.utf8;LC_MONETARY=English_United States.Dollar ;LC_NUMERIC=C;LC_TIME=English_United States.UTF-8"

Sys.setlocale("LC_ALL", "C")

## [1] "C"
Sys.setlocale("LC_ALL", "Korean")

## [1] "LC_COLLATE=Korean_Korea.949;LC_CTYPE=Korean_Korea.949;LC_MONETARY=Korean_Korea.949;LC_NUMERIC=C;LC_TIME=Korean_Korea.949"

employees <- read.csv('data/ts_산업별_취업자_20210206234505.csv',
                      header = TRUE,
                      na = '-',
                      strip.white = TRUE,
                      stringsAsFactors = TRUE)
colnames(employees)[1] <- 'time'
colnames(employees)[2] <- 'total'
colnames(employees)[3] <- 'employees.edu'
employees$time <- as.Date(paste0(employees$time, '. 01'), format = '%Y. %m. %d')
employees.ts <- ts(employees, start = c(2013, 01), frequency = 12)
employees.xts <- xts(employees[,2:3], order.by = employees[,1])
employees.tsibble <- as_tsibble(employees, index = time)
employees.tsibble

## # A tsibble: 96 x 3 [1D]
##       time     total employees.edu
##   <date>     <int>        <int>
## 1 2013-01-01     24287        1710
## 2 2013-02-01     24215        1681
## 3 2013-03-01     24736        1716
## 4 2013-04-01     25322        1745
## 5 2013-05-01     25610        1774
## 6 2013-06-01     25686        1786
## 7 2013-07-01     25681        1813
## 8 2013-08-01     25513        1811
## 9 2013-09-01     25701        1794
## 10 2013-10-01    25798        1790
## # i 86 more rows

```

시계열 데이터 시각화

실습 데이터 생성

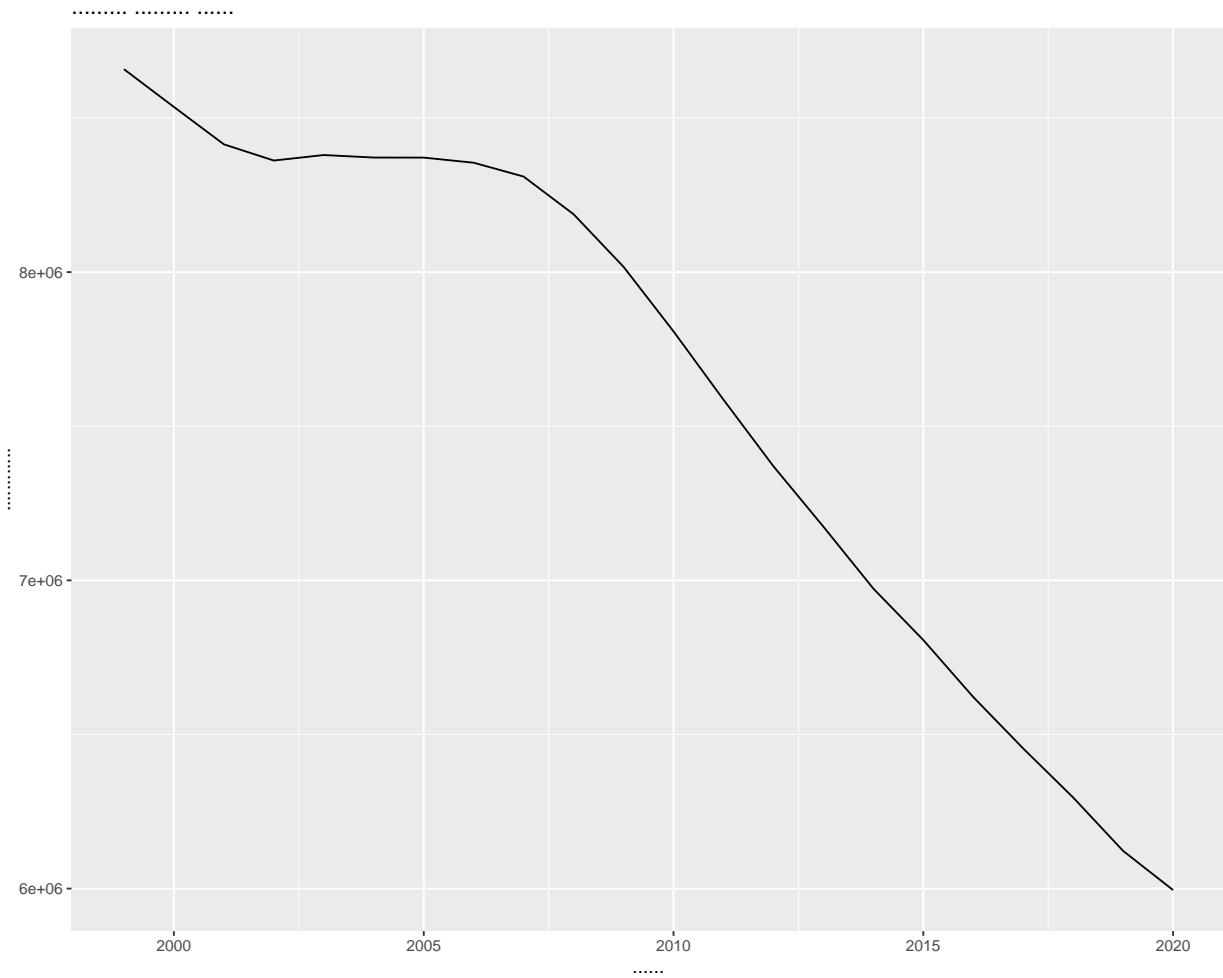
```

covid19 <- read.csv('data/ts_covid19.csv', header = TRUE, na = '-', strip.white = TRUE, stringsAsFactors = FALSE)
colnames(covid19) <- c('category', 'status', 'date', 'value')
covid19 <- covid19[, c(3, 1, 2, 4)]
covid19$date <- as.Date(covid19$date, "%Y. %m. %d")
covid19 <- covid19 %>%
  filter(grepl('세', category)) %>%
  filter(category != '세종')
covid19$value <- ifelse(is.na(covid19$value), 0, covid19$value)
covid19 <- tidyrr::spread(covid19, category, value)
covid19.ts <- ts(covid19[, 2:10], frequency = 365)
covid19.xts <- as.xts(covid19[, 3:10], order.by = covid19$date)
covid19.tsibble <- as_tsibble(covid19, index = date)
covid19.tsibble

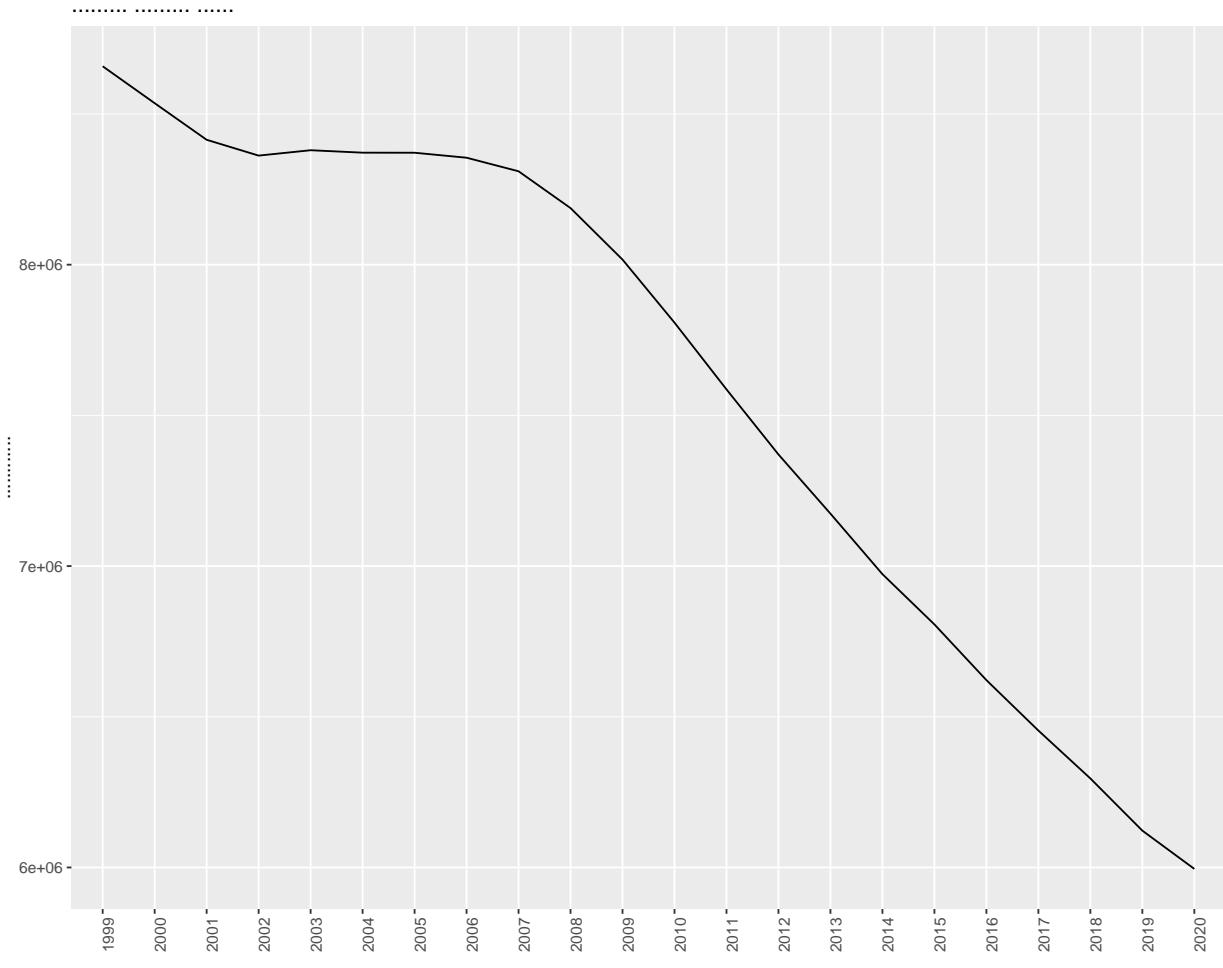
## # A tsibble: 302 x 11 [1D]
##   date      status `0-9세` `10-19세` `20-29세` `30-39세` `40-49세` `50-59세` `60-69세` `70-79세` `80세 이상`
##   <date>     <fct>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 2020-04-09 신규      2        4       12       7       7       2
## 2 2020-04-10 신규      1        1       7        4       2       3
## 3 2020-04-11 신규      1        5       5        2       3       6
## 4 2020-04-12 신규      0        3       13       5       1       4
## 5 2020-04-13 신규      2        1       10       2       1       2
## 6 2020-04-14 신규      0        3       7        4       4       3
## 7 2020-04-15 신규      0        4       9        5       3       2
## 8 2020-04-16 신규      3        3       5        1       2       3
## 9 2020-04-17 신규      1        4       9        3       1       2
## 10 2020-04-18 신규     2        0       9        1       1       2
## # i 292 more rows
## # i 3 more variables: `60-69세` <dbl>, `70-79세` <dbl>, `80세 이상` <dbl>

students %>%
  ggplot(aes(x = 연도, y = 학생수계)) +
  geom_line(aes(group = 1)) +
  labs(title = '연도별 학생수 추이')

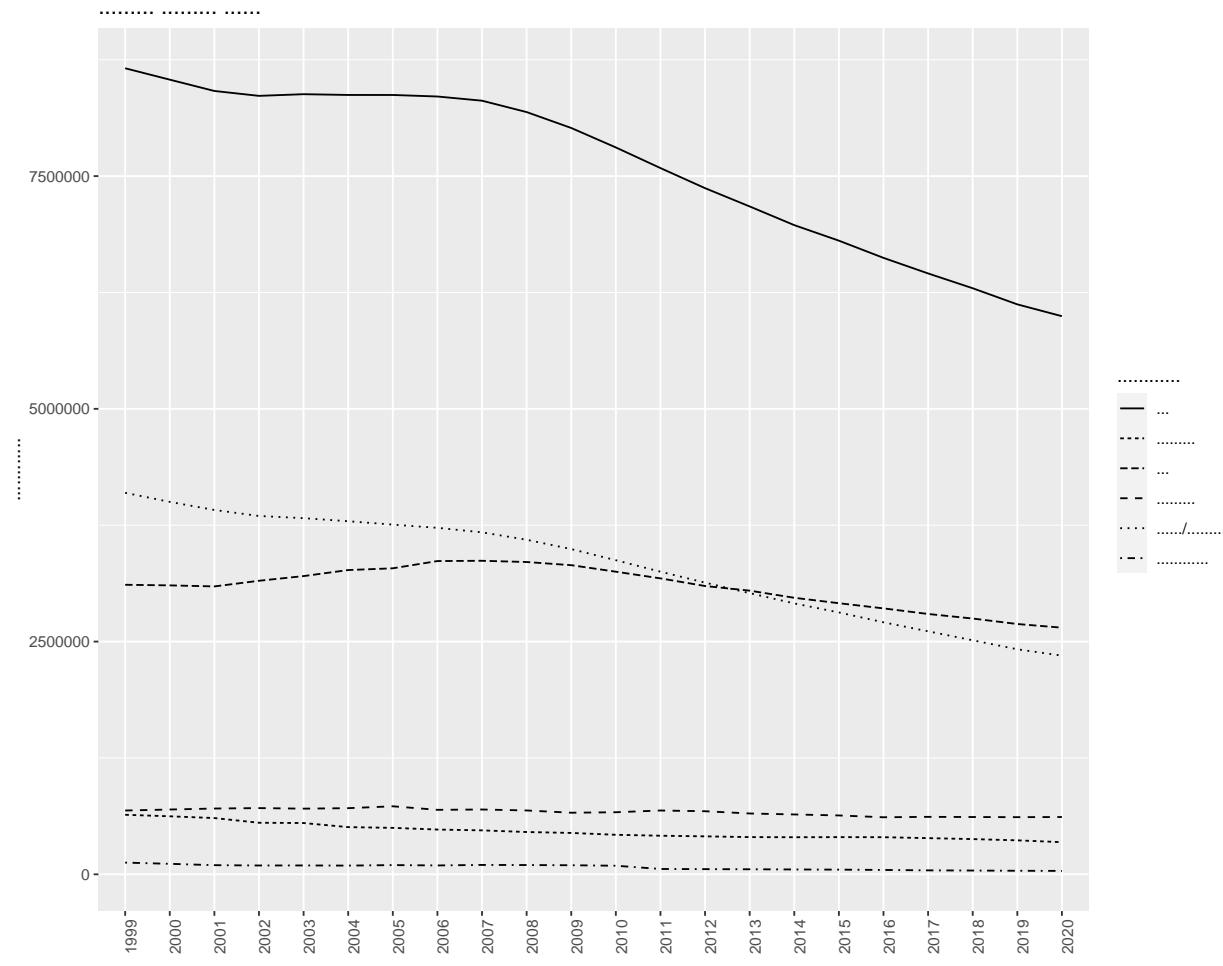
```



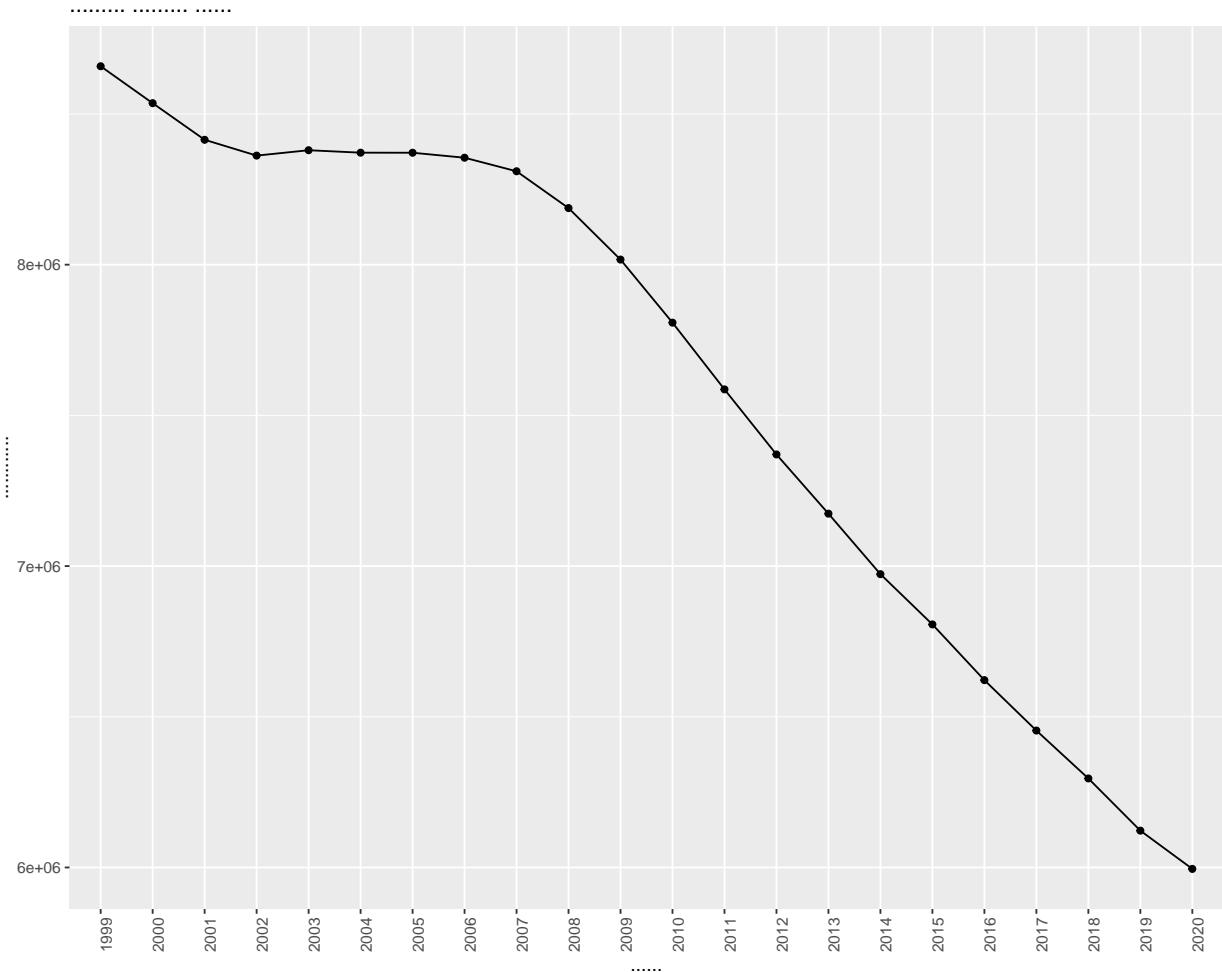
```
ggplot(data = students, aes(x = as.factor(lubridate::year(연도)), y = 학생수)) +  
  geom_line(aes(group = 1)) +  
  theme(axis.text.x=element_text(angle=90,hjust=1)) +  
  labs(title = '연도별 학생수 추이', x = '연도')
```



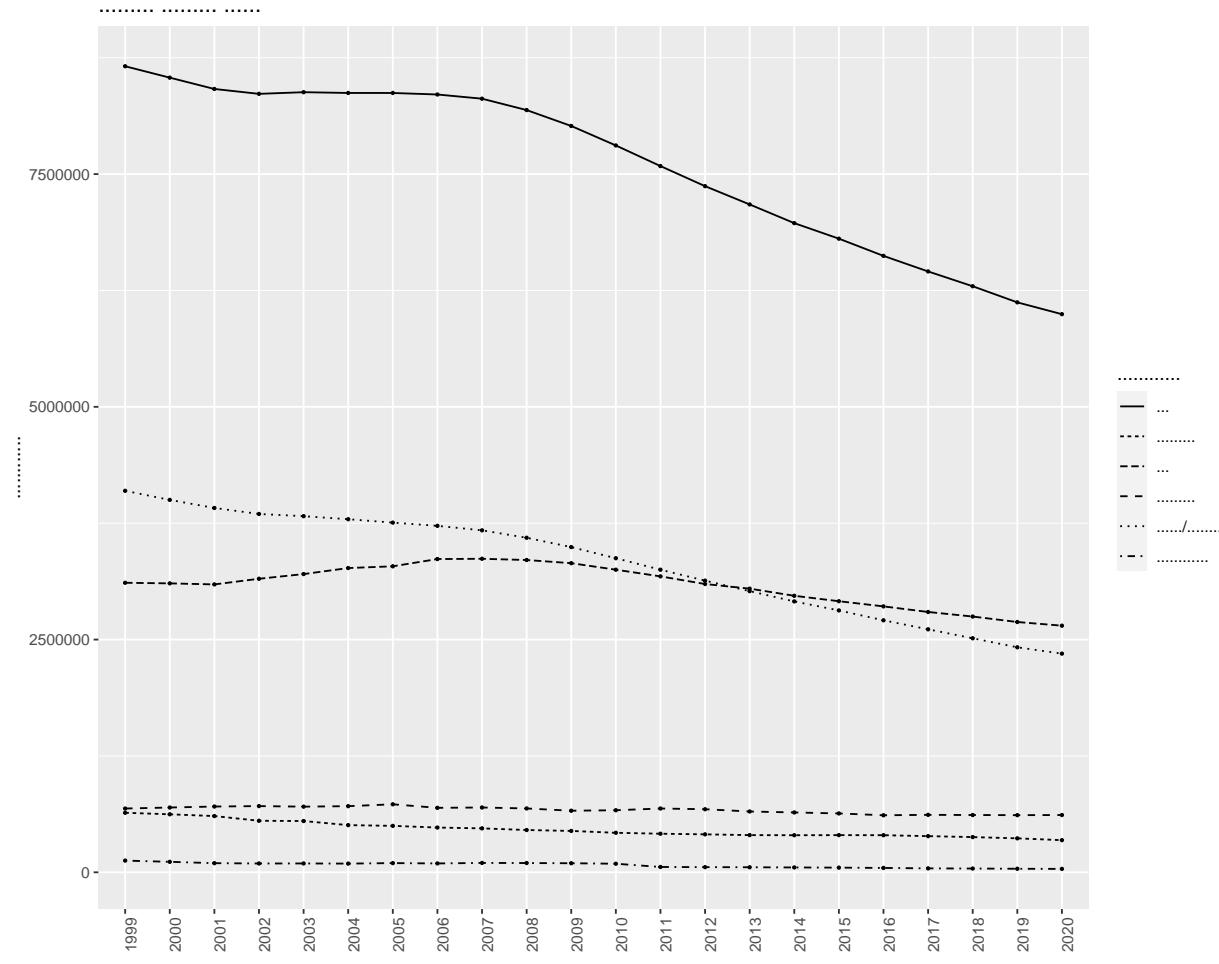
```
ggplot(data = students.all, aes(x = 연도, y = 학생수계)) +  
  geom_line(aes(group = 지역규모, linetype = 지역규모)) +  
  theme(axis.text.x=element_text(angle=90,hjust=1)) +  
  labs(title = '연도별 학생수 추이', x = '연도')
```



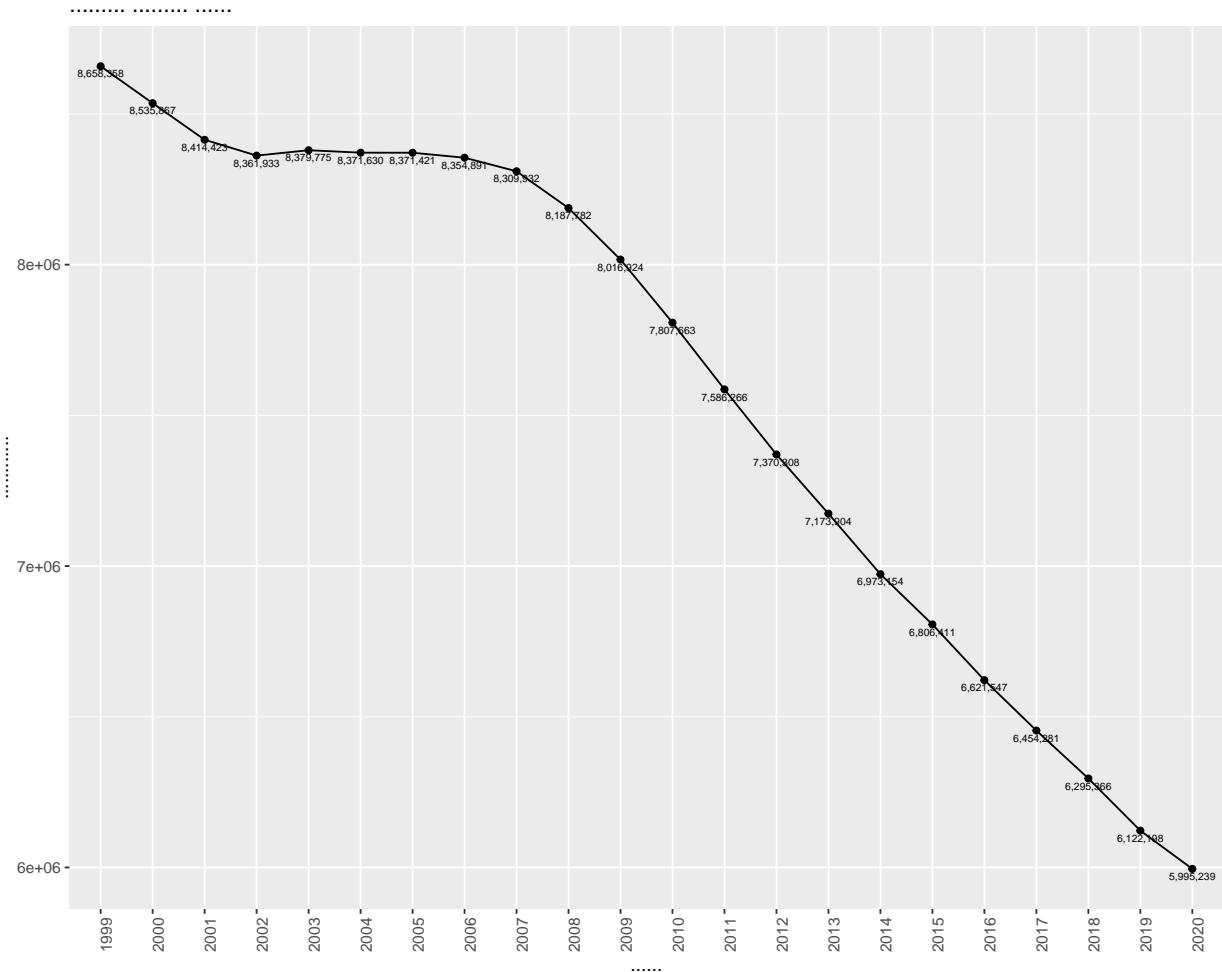
```
ggplot(data = students, aes(x = as.factor(lubridate::year(연도)), y = 학생수계)) +
  geom_line(aes(group = 1)) +
  geom_point(shape = 'circle') +
  theme(axis.text.x=element_text(angle=90,hjust=1)) +
  labs(title = '연도별 학생수 추이', x = '연도')
```



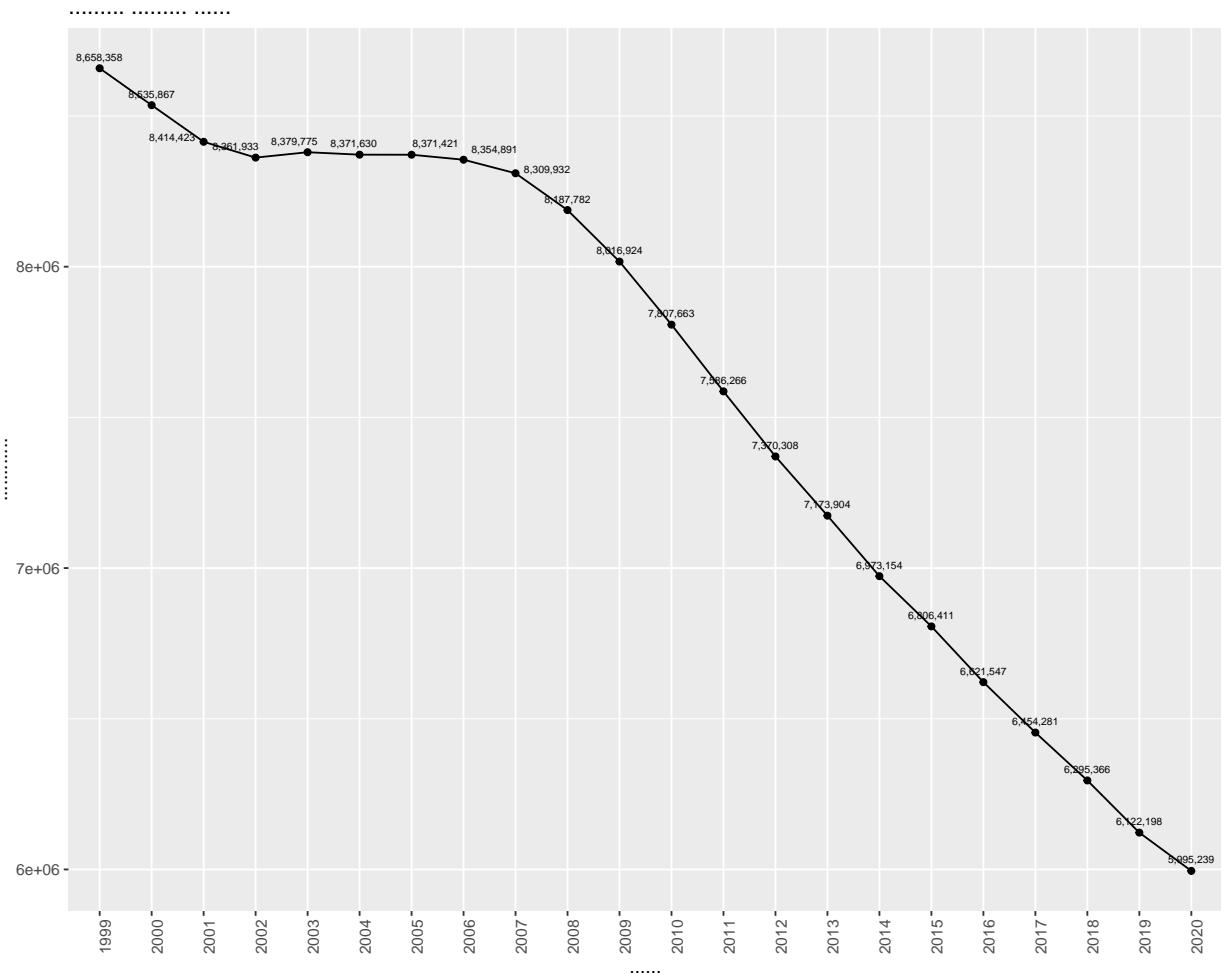
```
ggplot(data = students.all, aes(x = 연도, y = 학생수계)) +  
  geom_line(aes(group = 지역규모, linetype = 지역규모)) +  
  geom_point(shape = 'circle', size = 0.5) +  
  theme(axis.text.x=element_text(angle=90,hjust=1)) +  
  labs(title = '연도별 학생수 추이', x = '연도')
```



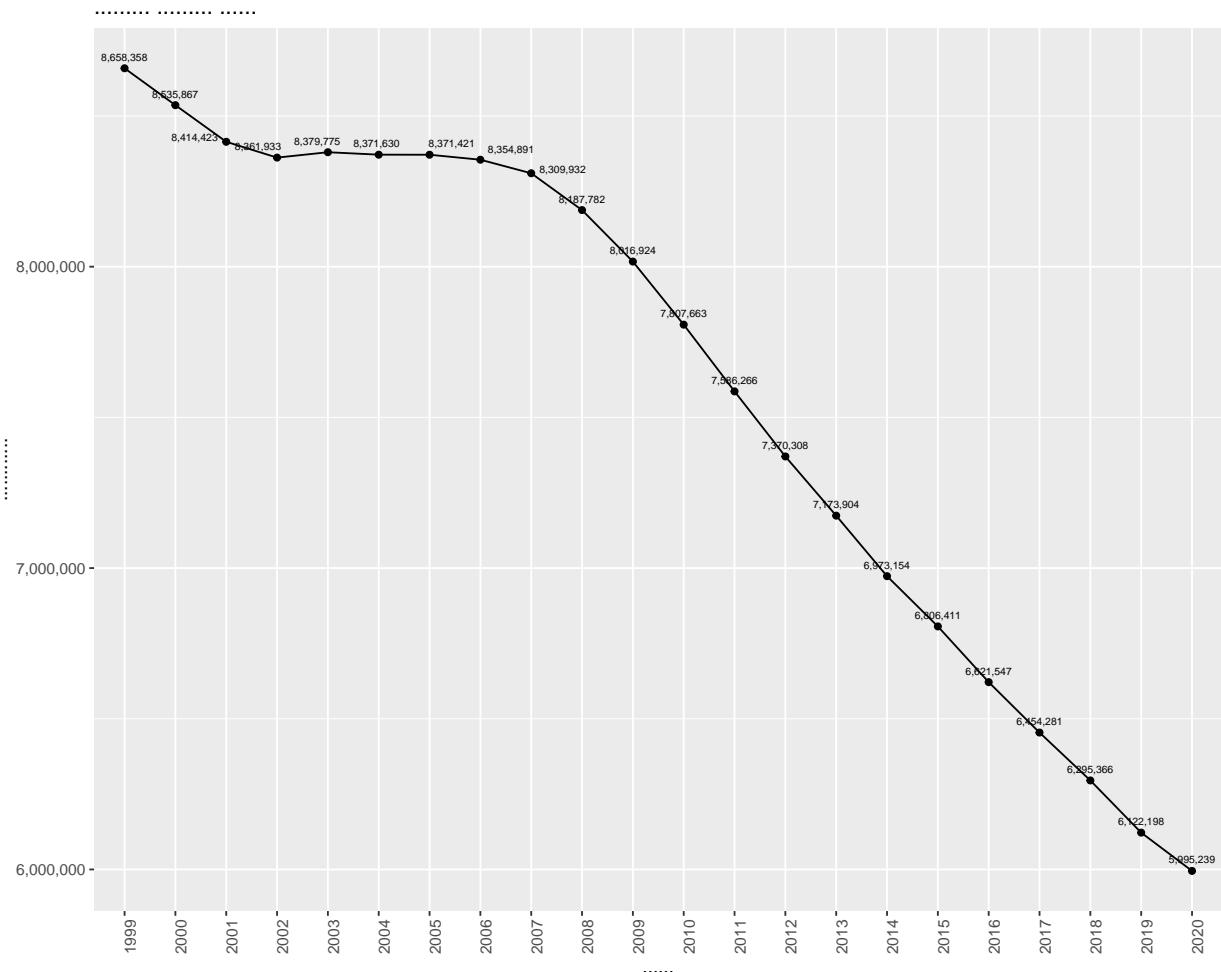
```
ggplot(data = students, aes(x = as.factor(lubridate::year(연도)), y = 학생수계)) +
  geom_line(aes(group = 1)) +
  geom_point(shape = 'circle') +
  geom_text(aes(label = scales::number(학생수계, big.mark = ',')), size = 2, vjust = 1.5) +
  theme(axis.text.x=element_text(angle=90,hjust=1)) +
  labs(title = '연도별 학생수 추이', x = '연도')
```



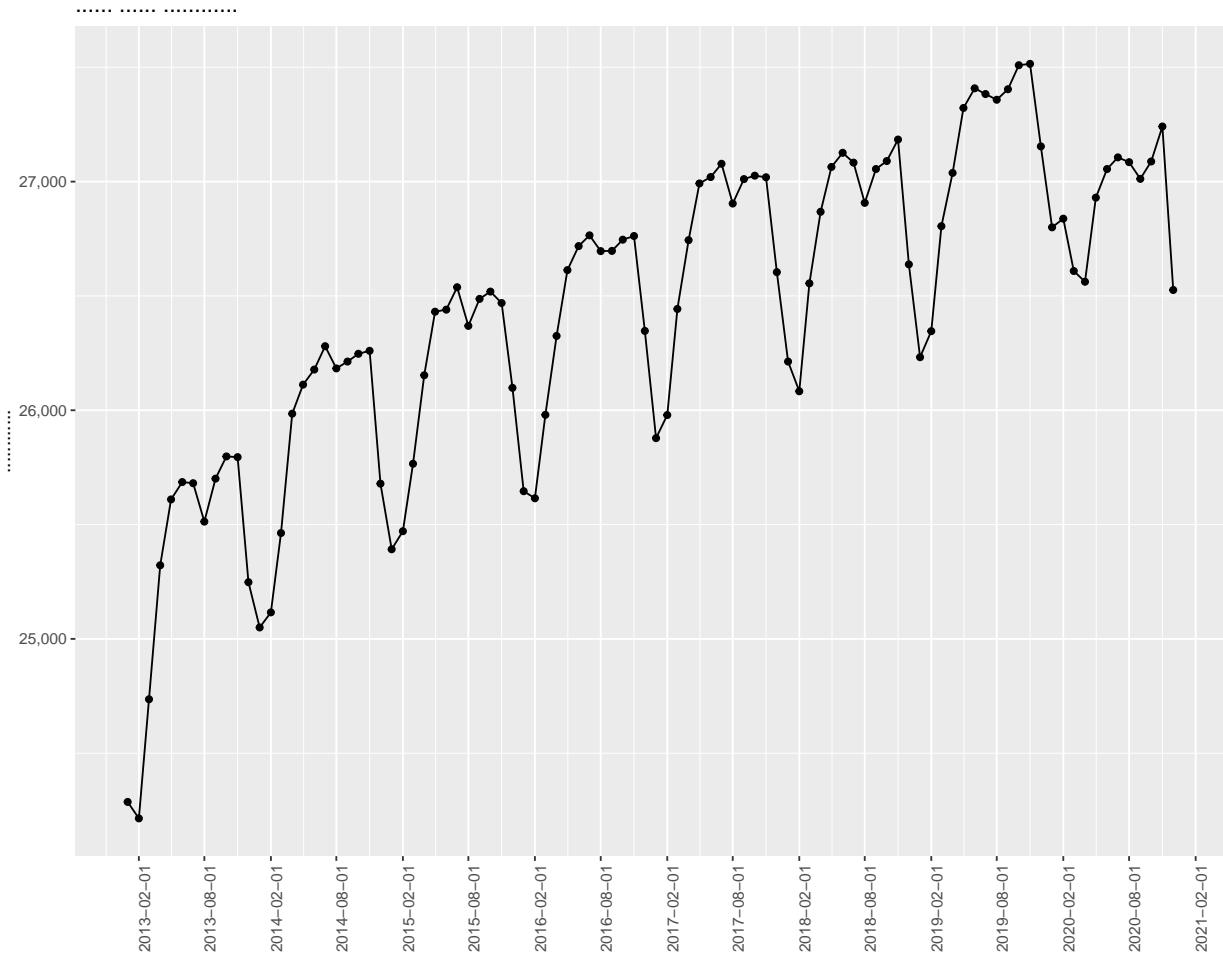
```
ggplot(data = students, aes(x = as.factor(lubridate::year(연도)), y = 학생수)) +
  geom_line(aes(group = 1)) +
  geom_point(shape = 'circle') +
  geom_text_repel(aes(label = scales::number(학생수), big.mark = ', ')), size = 2, vjust = 1.5) +
  theme(axis.text.x=element_text(angle=90,hjust=1)) +
  labs(title = '연도별 학생수 추이', x = '연도')
```



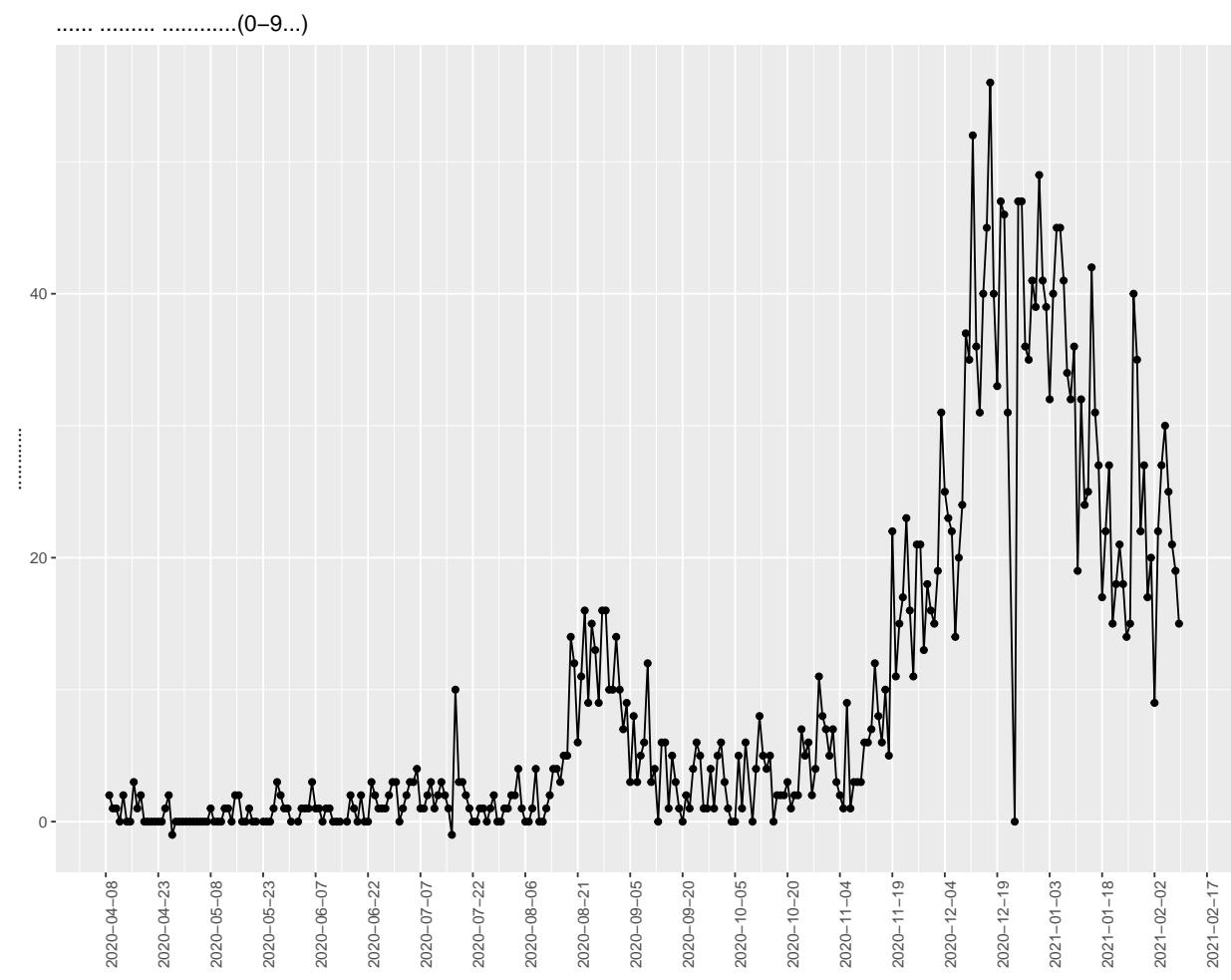
```
ggplot(data = students, aes(x = as.factor(lubridate::year(연도)), y = 학생수)) +
  geom_line(aes(group = 1)) +
  geom_point(shape = 'circle') +
  geom_text_repel(aes(label = scales::number(학생수), big.mark = ',', vjust = 1.5)) +
  theme(axis.text.x=element_text(angle=90,hjust=1)) +
  labs(title = '연도별 학생수 추이', x = '연도') +
  scale_y_continuous(labels = scales::number_format(big.mark = ','))
```



```
ggplot(data = employees, aes(x = time, y = total)) +
  geom_line(aes(group = 1)) +
  geom_point(shape = 'circle') +
  labs(title = '월별 신규 취업자수', x = '기간', y = '취업자수') +
  scale_y_continuous(labels = scales::number_format(big.mark = ',')) +
  scale_x_date(breaks = '6 month') +
  theme(axis.text.x=element_text(angle=90,hjust=1))
```



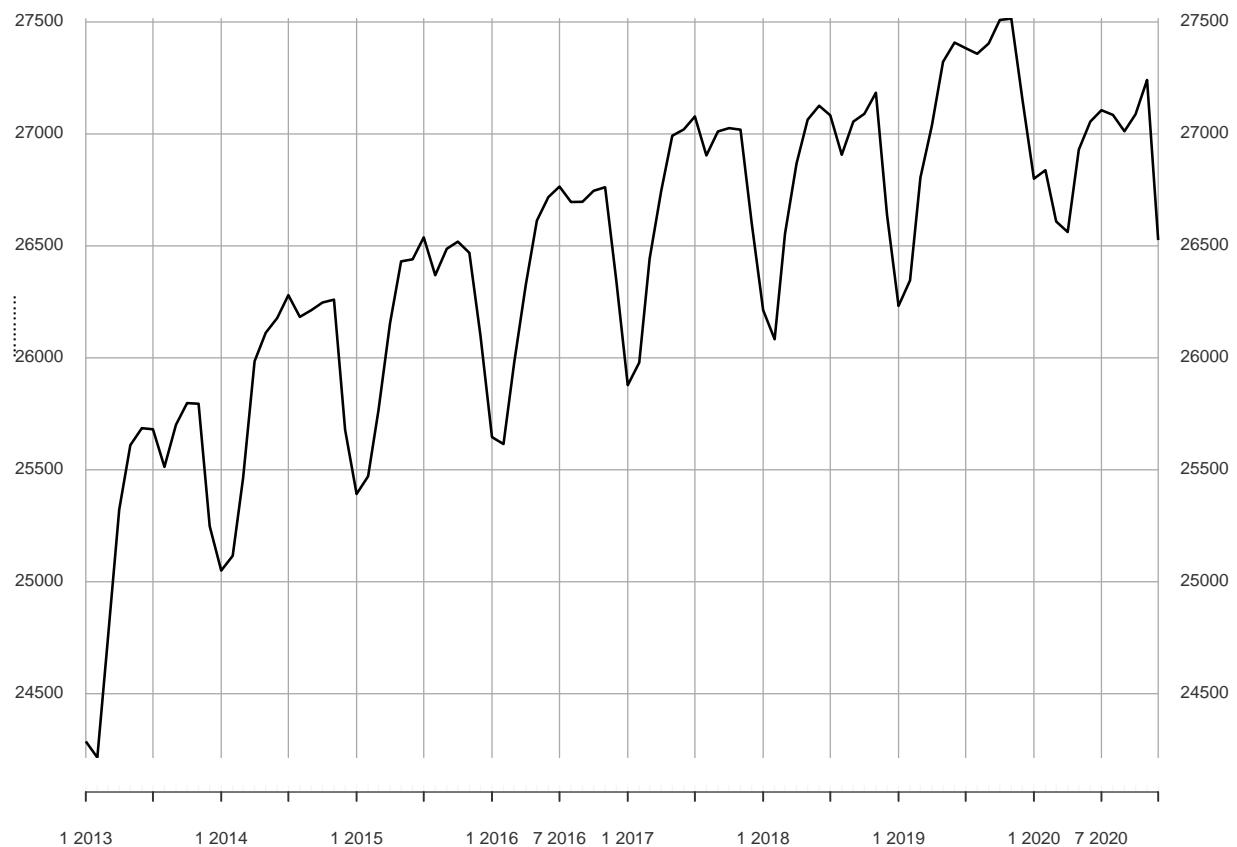
```
ggplot(data = covid19, aes(x = date, y = `0-9세`)) +
  geom_line(aes(group = 1)) +
  geom_point(shape = 'circle') +
  labs(title = '일별 코로나 확진자수(0-9세)', x = '시간', y = '확진자수') +
  scale_y_continuous(labels = scales::number_format(big.mark = ',')) +
  scale_x_date(breaks = '15 day') +
  theme(axis.text.x=element_text(angle=90,hjust=1))
```



xts 패키지

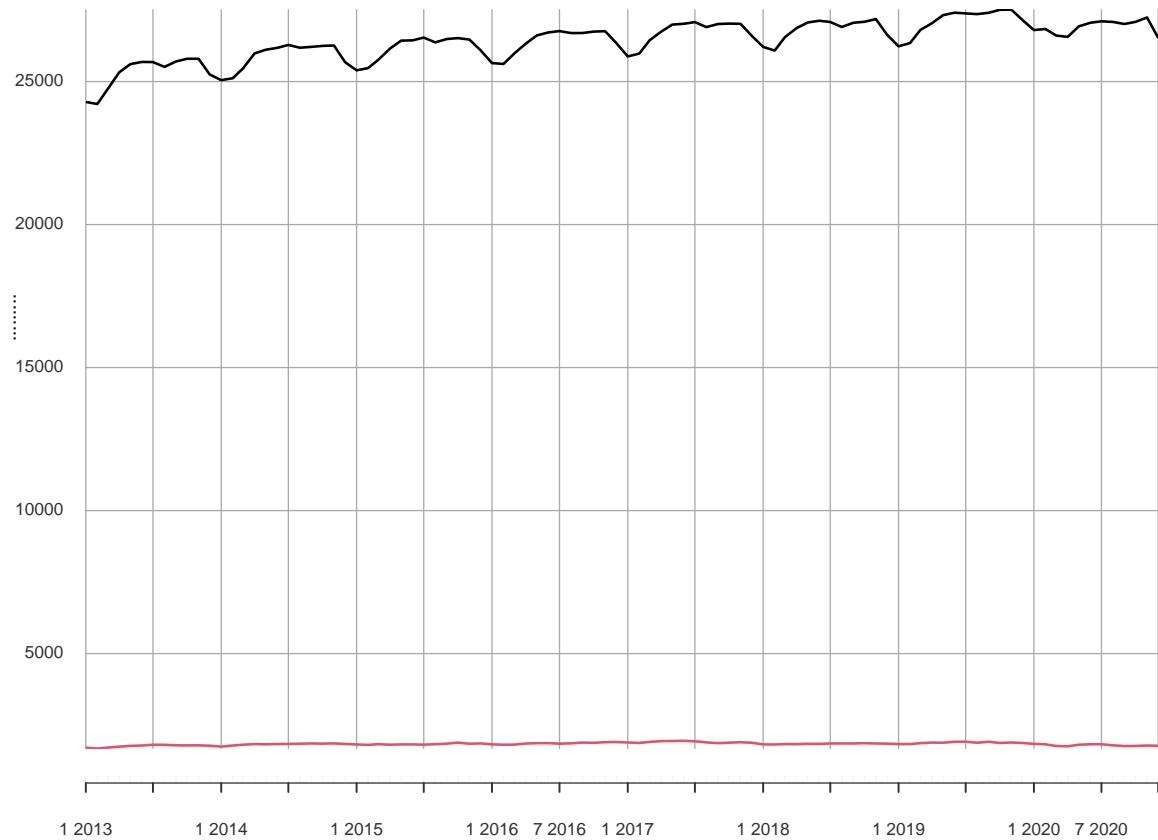
```
plot.xts(employees.xts$total, main = '월별 취업자수 추이', xlab = '월, 연', ylab = '취업자수')
```

2013-01-01 / 2020-12-01



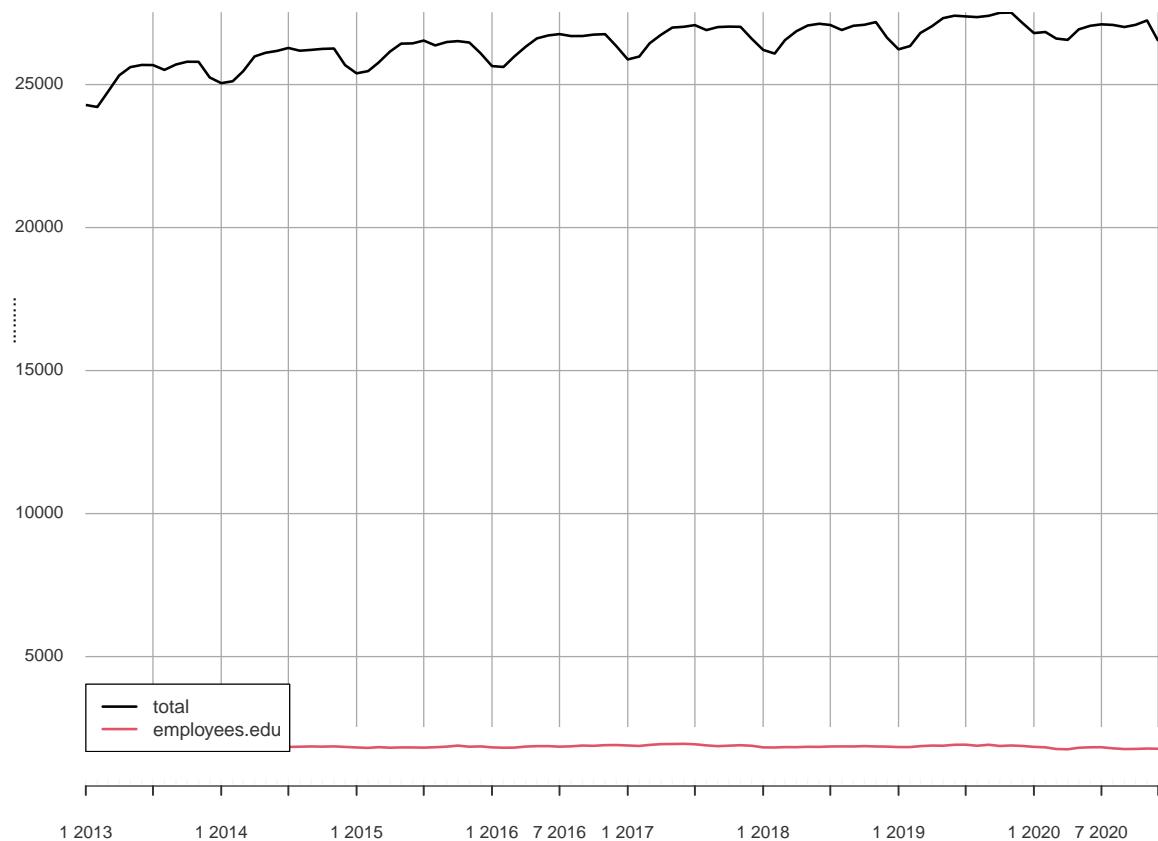
```
plot.xts(employees.xts, main = '연도별 학생수 추이', xlab = '연', ylab = '학생수', yaxis.right=FALSE)
```

2013-01-01 / 2020-12-01



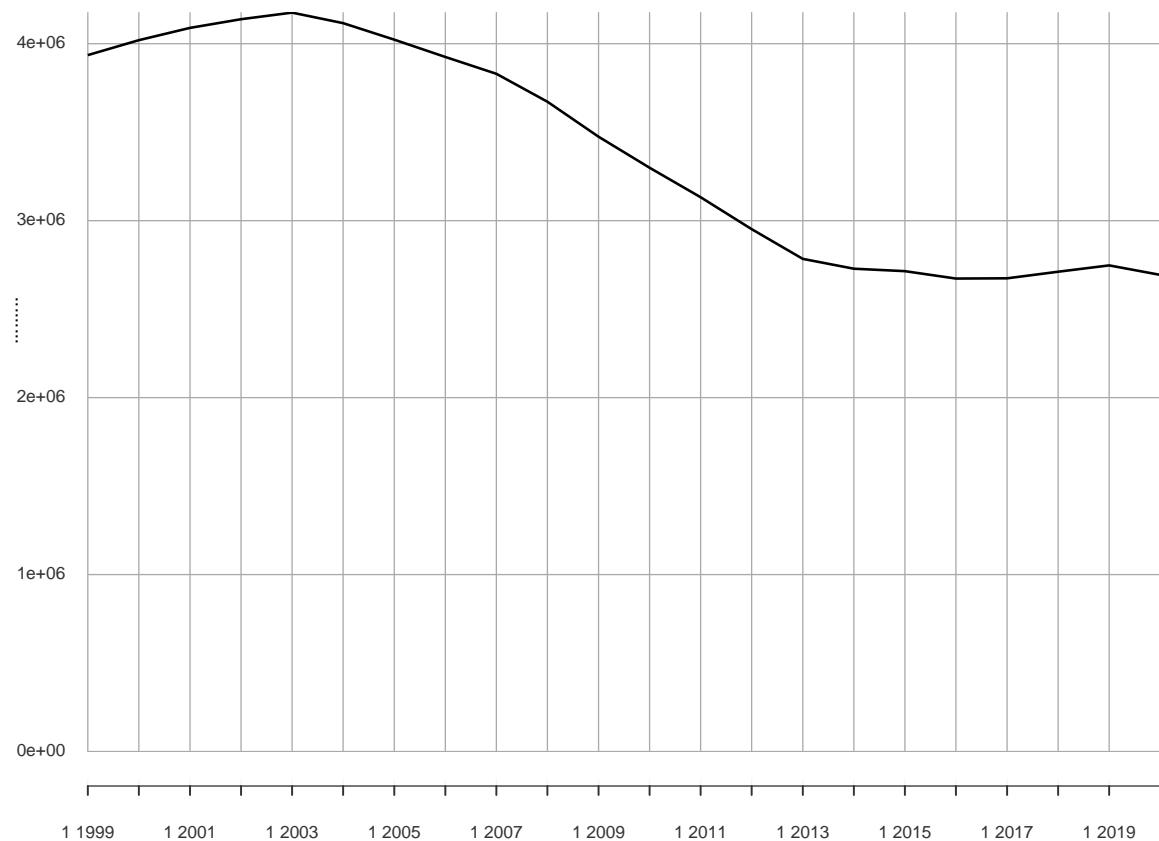
```
addLegend('bottomleft', ncol = 1, bg = 'white', lty=c(rep(1, 12)), lwd=c(rep(2, 12)), bty="o")
```

2013-01-01 / 2020-12-01



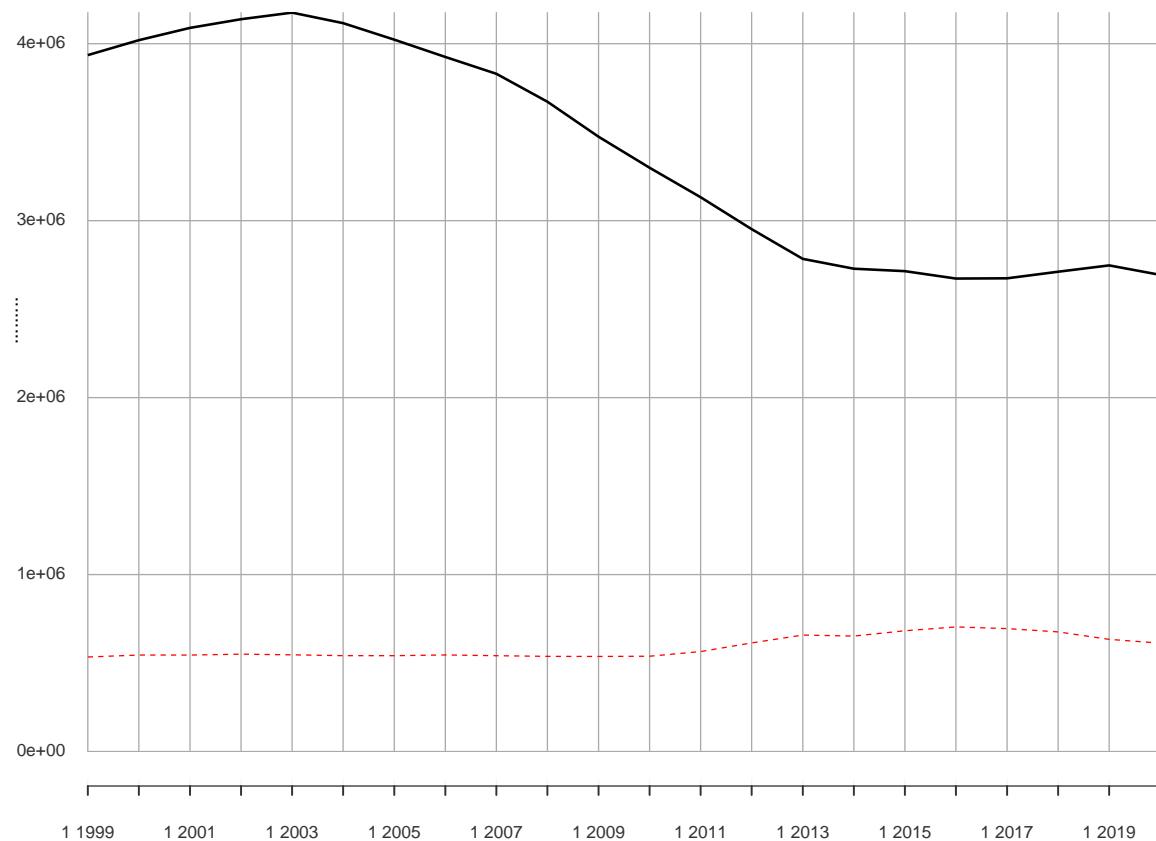
```
plot.xts(students.xts$초등학교, main = '연도별 학생수 추세', xlab = '연', ylab = '학생수', yaxis.right=FALSE)
```

1999-01-01 / 2020-01-01

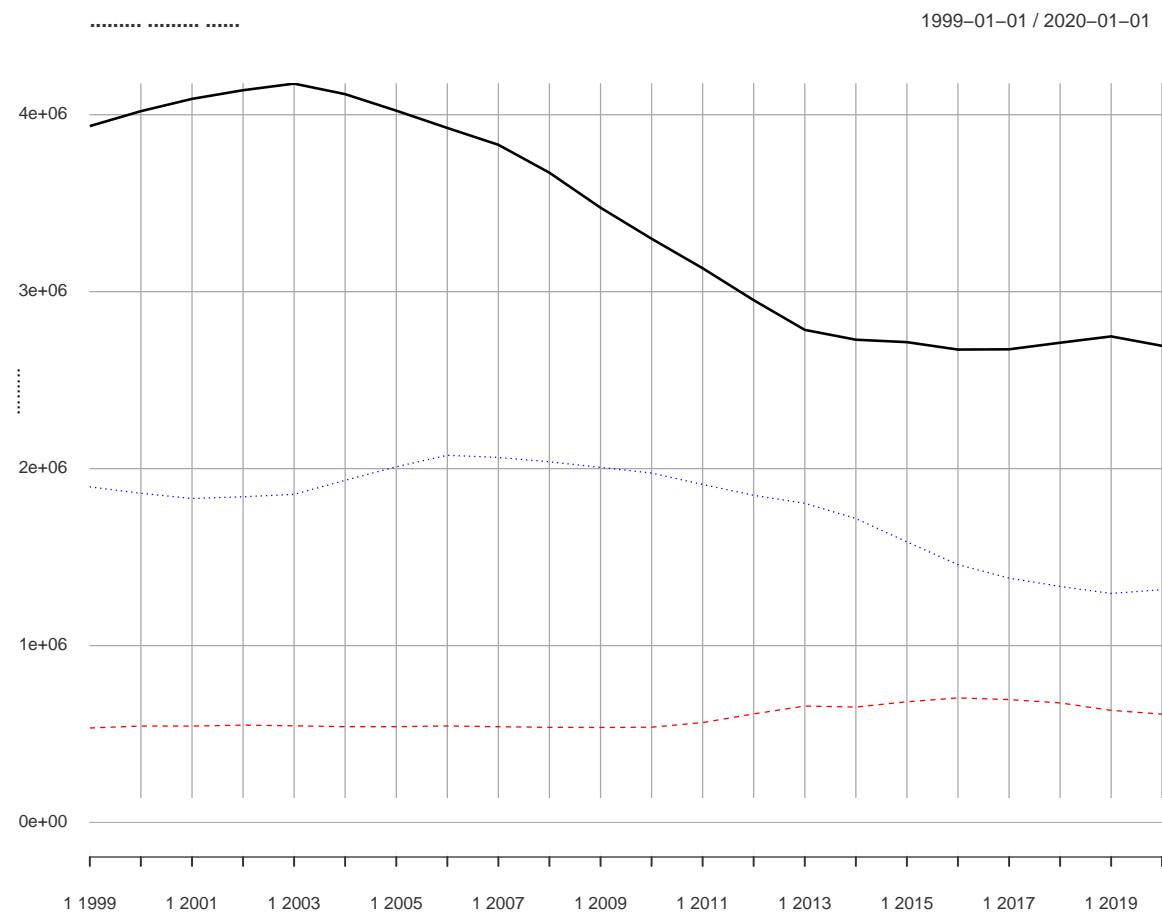


```
lines(students.xts$유치원, lty = 2, col = 'red')
```

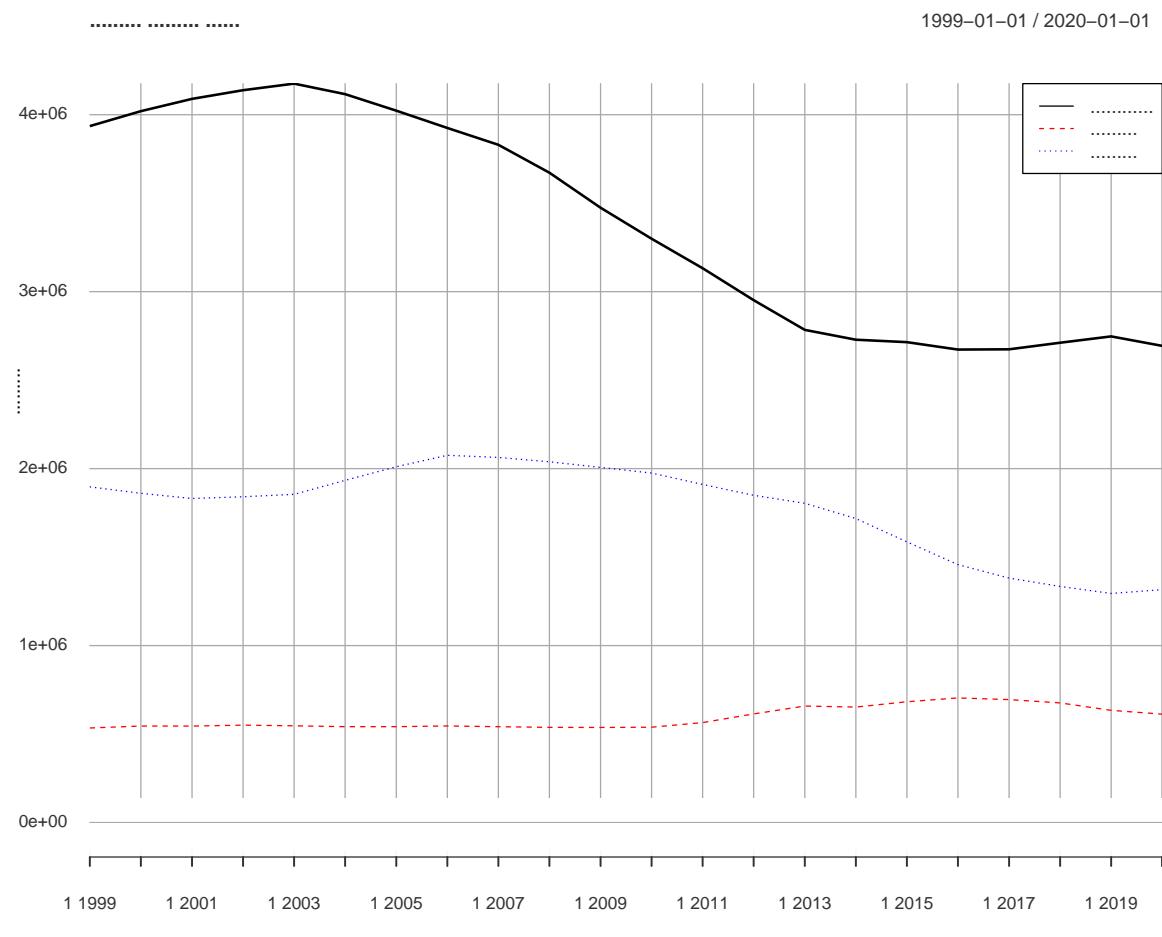
1999-01-01 / 2020-01-01



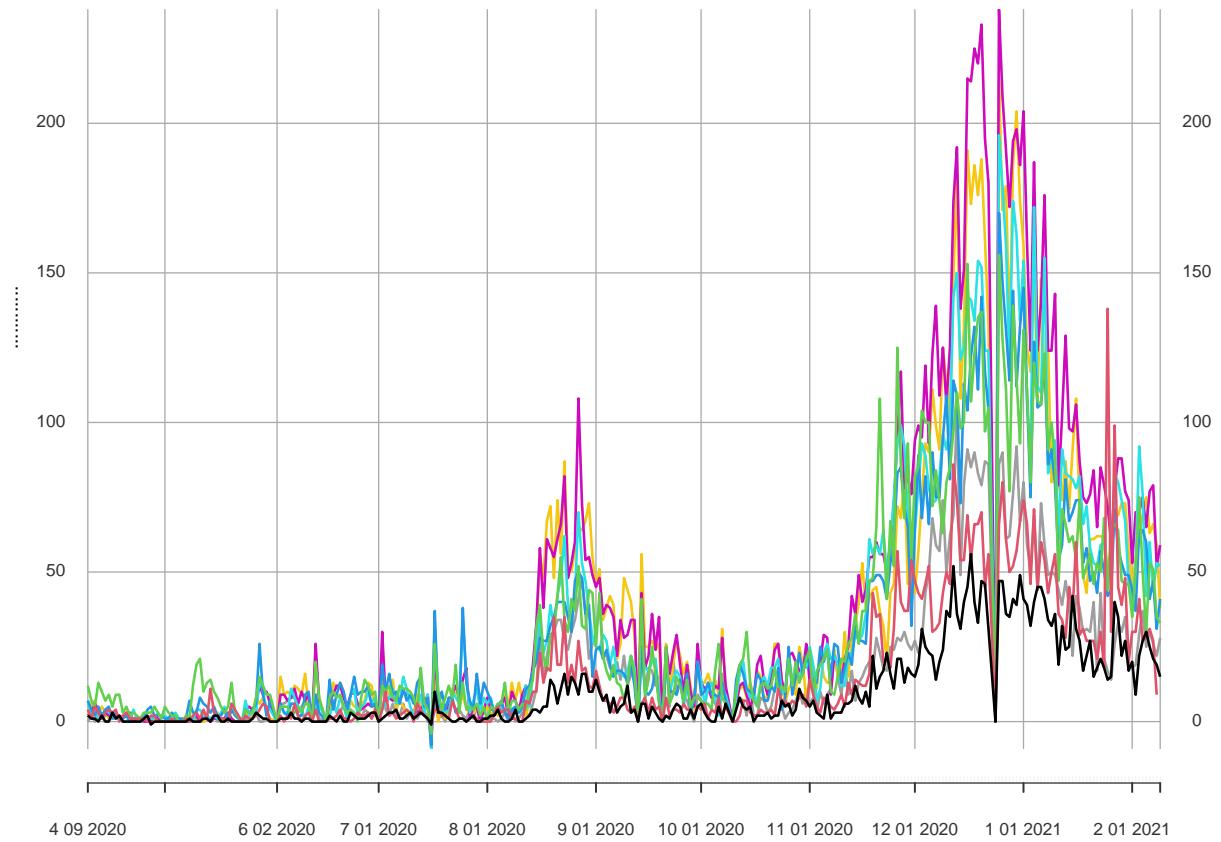
```
lines(students.xts$중학교, lty = 3, col = 'blue')
```



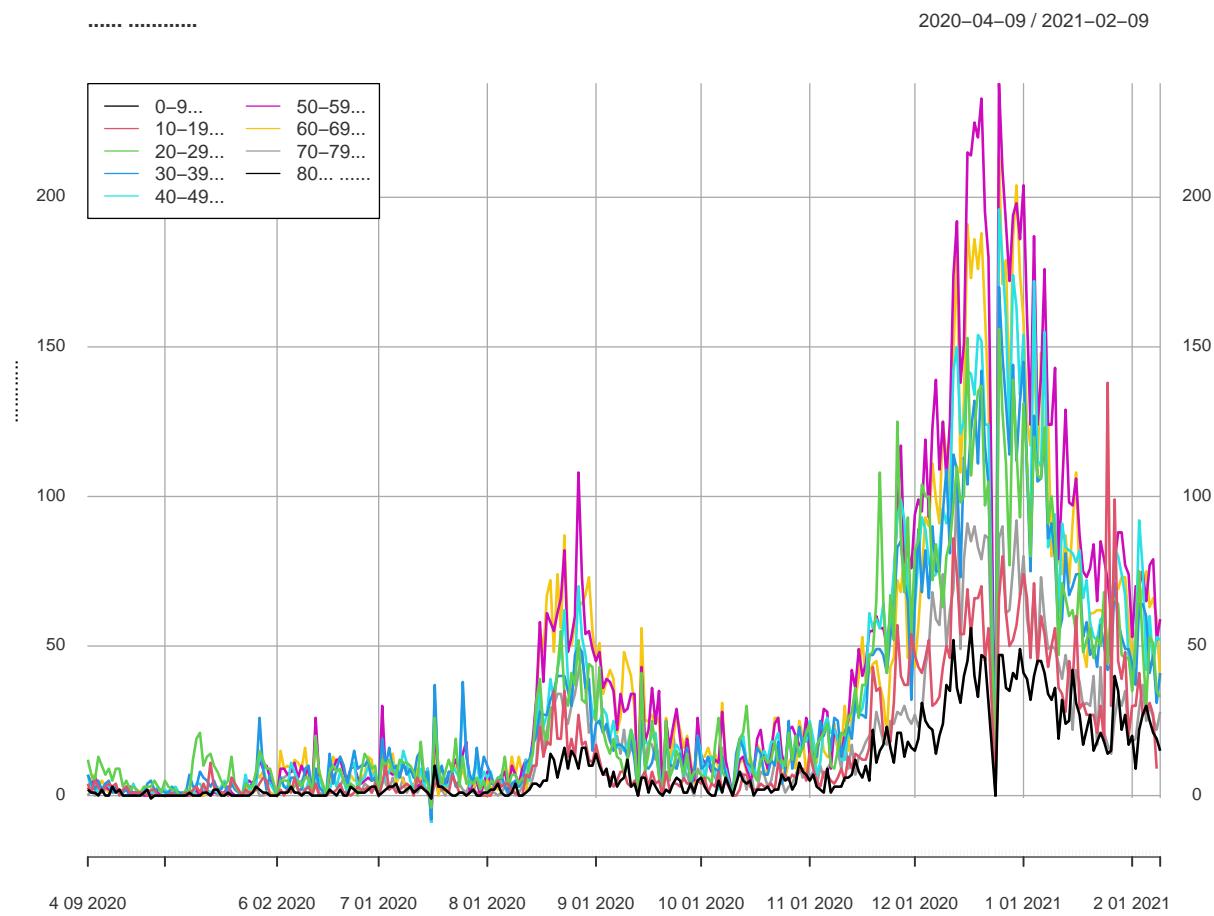
```
addLegend('topright', ncol = 1, legend.names = c('초등학교', '유치원', '중학교'), col = c('black', 'red', 'blue'))
```



2020-04-09 / 2021-02-09

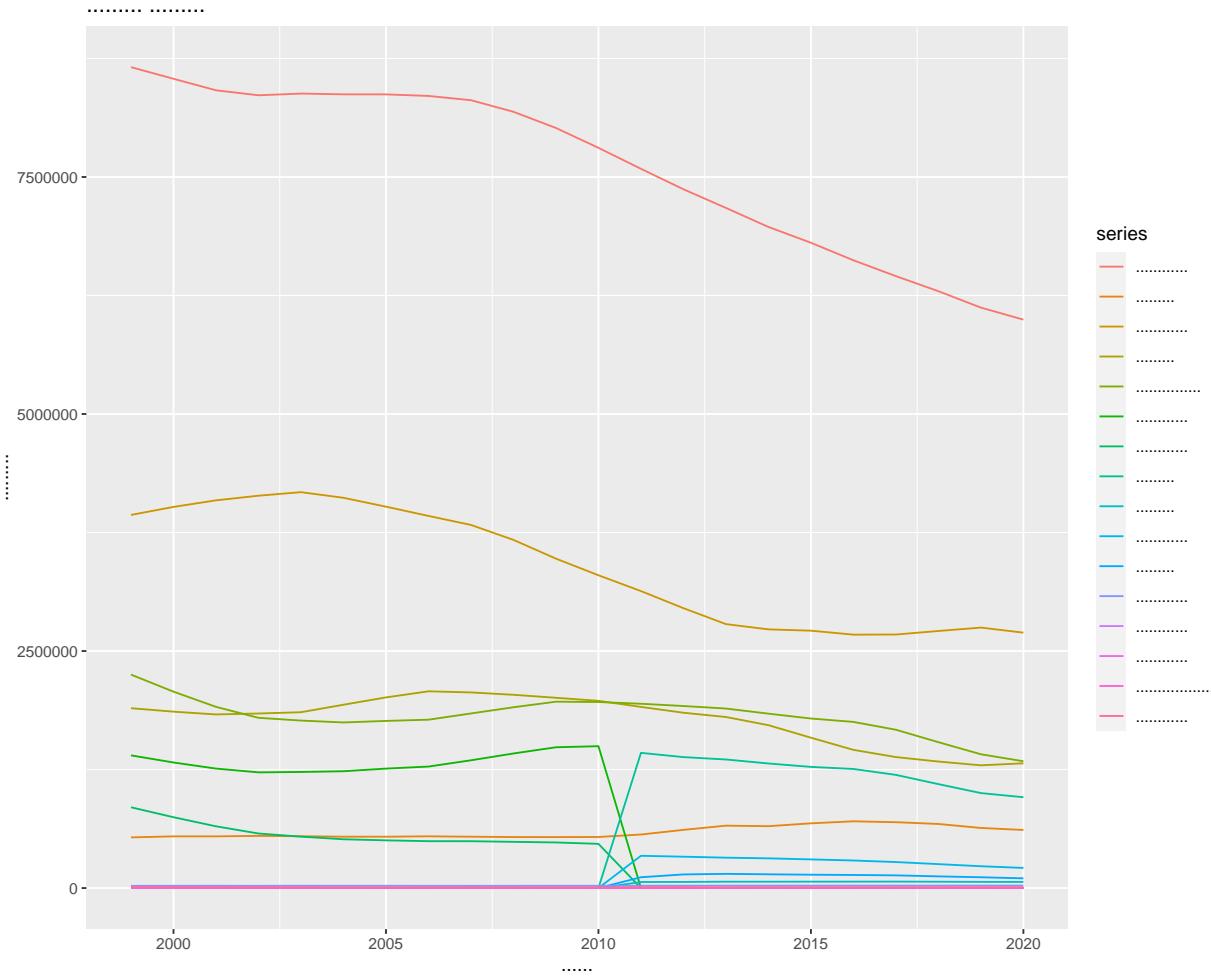


```
addLegend('topleft', ncol = 2, legend.names = c('0-9세', '10-19세', '20-29세', '30-39세', '40-49세', '50-59세'))
```

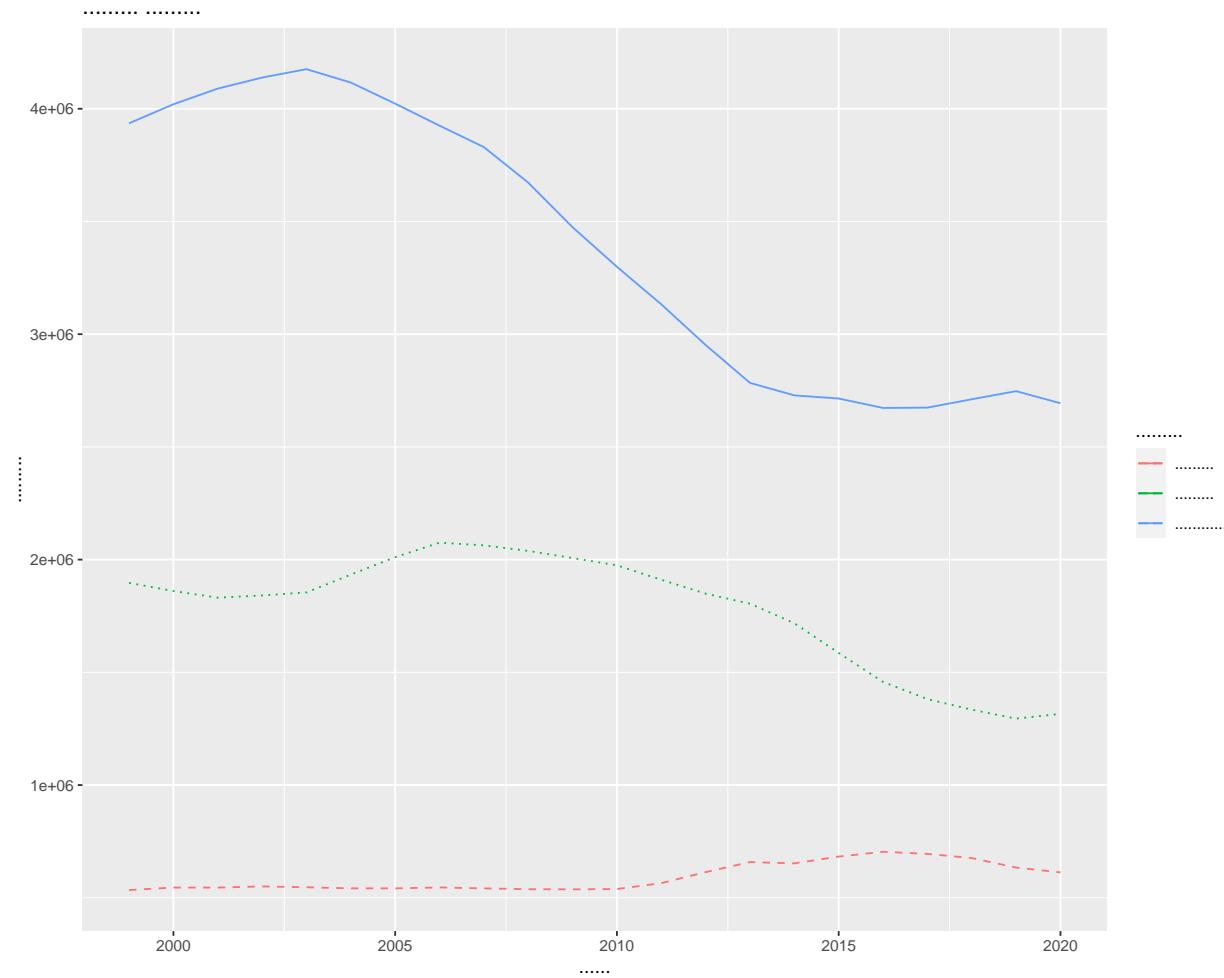


ts:forecast

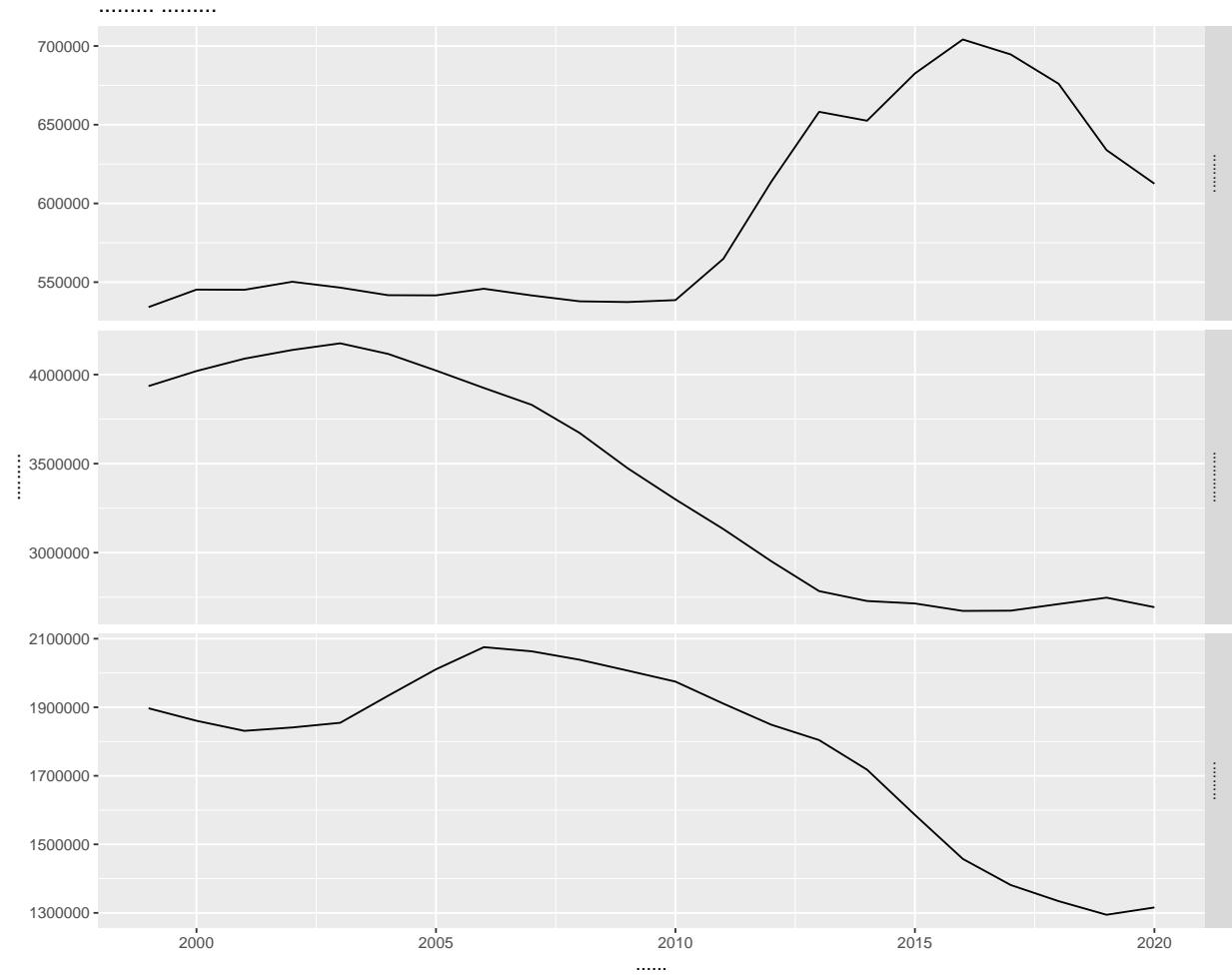
```
autoplot(students.ts[,-1], main = '연도별 학생수', xlab = '연도', ylab = '학생수')
```



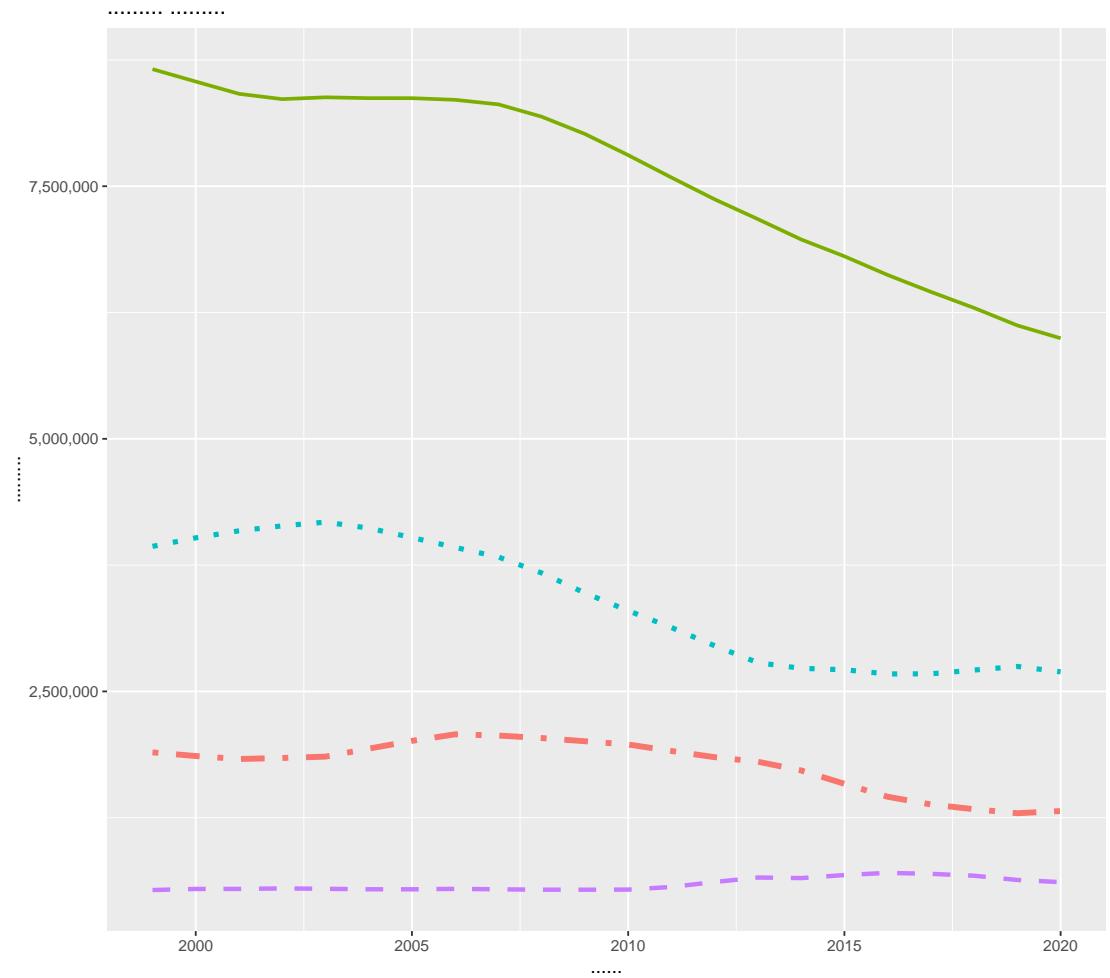
```
autoplot(students.ts[, 4], main = '연도별 학생수', xlab = '연도', ylab = '학생수', series = '초등학교', lty = 1)
autolayer(students.ts[, 3], series = '유치원', lty = 2) +
autolayer(students.ts[, 5], series = '중학교', lty = 3) +
labs(colour = "학교급")
```



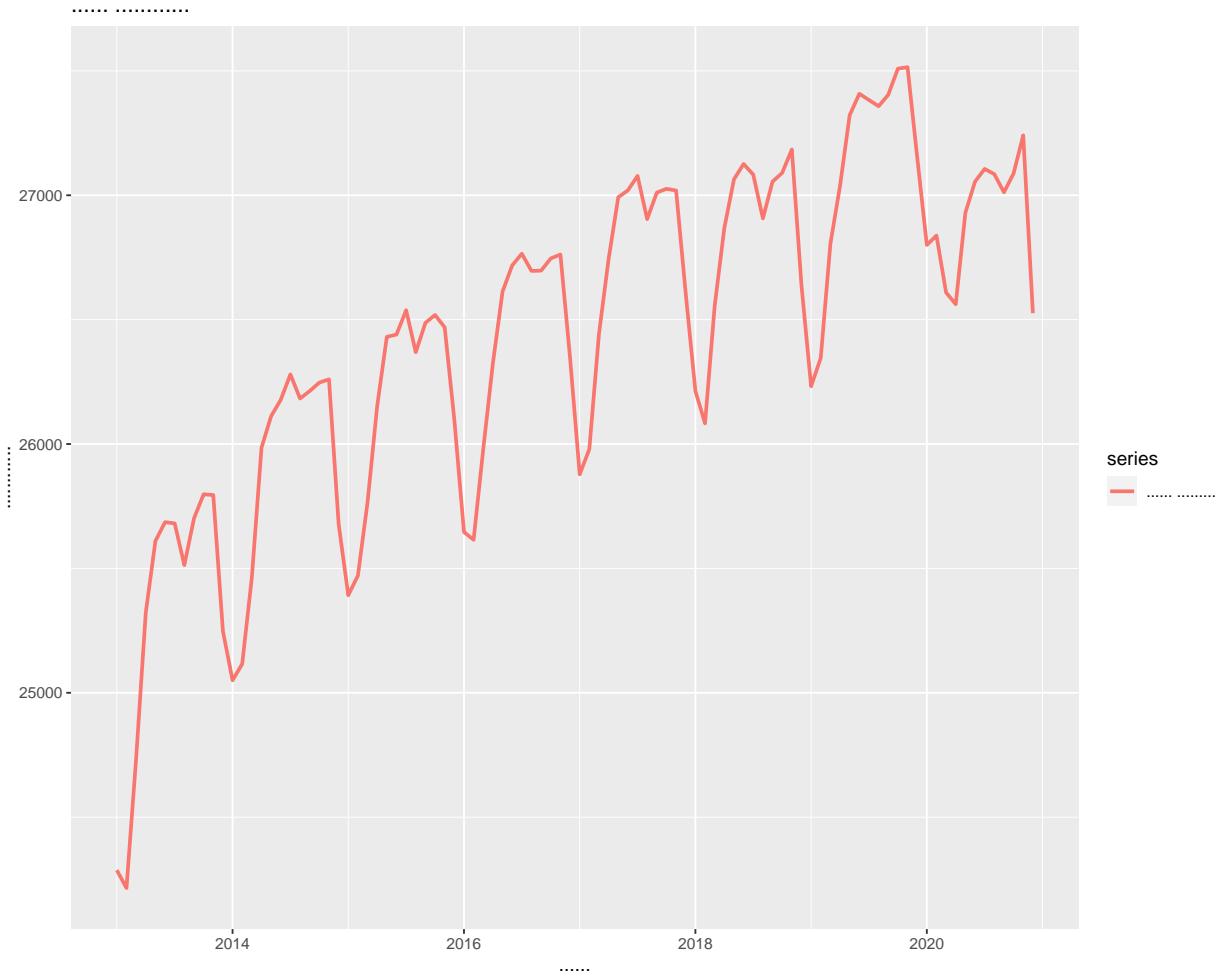
```
autoplot(students.ts[, 3:5], main = '연도별 학생수', xlab = '연도', ylab = '학생수', facet = TRUE)
```

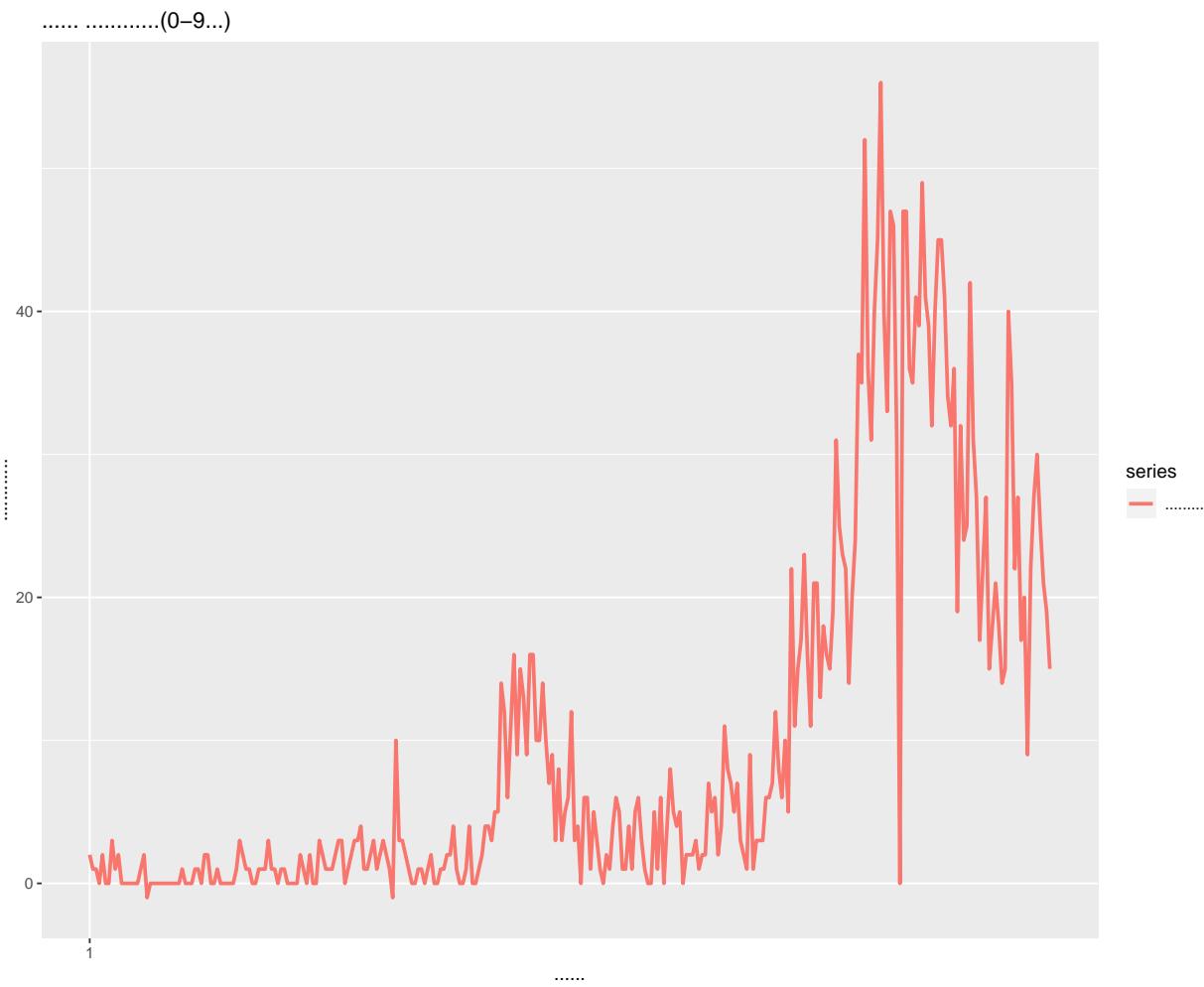


```
autoplot(students.ts[,2], main = '연도별 학생수', xlab = '연도', ylab = '학생수', series = '유치원', lty = 1)
autolayer(students.ts[,3], series = '초등학교', lty = 2, lwd = 1.2) +
autolayer(students.ts[,4], series = '중학교', lty = 3, lwd = 1.4) +
autolayer(students.ts[,5], series = '고등학교', lty = 4, lwd = 1.6) +
scale_y_continuous(labels=scales::number_format(big.mark = ',',')))
```



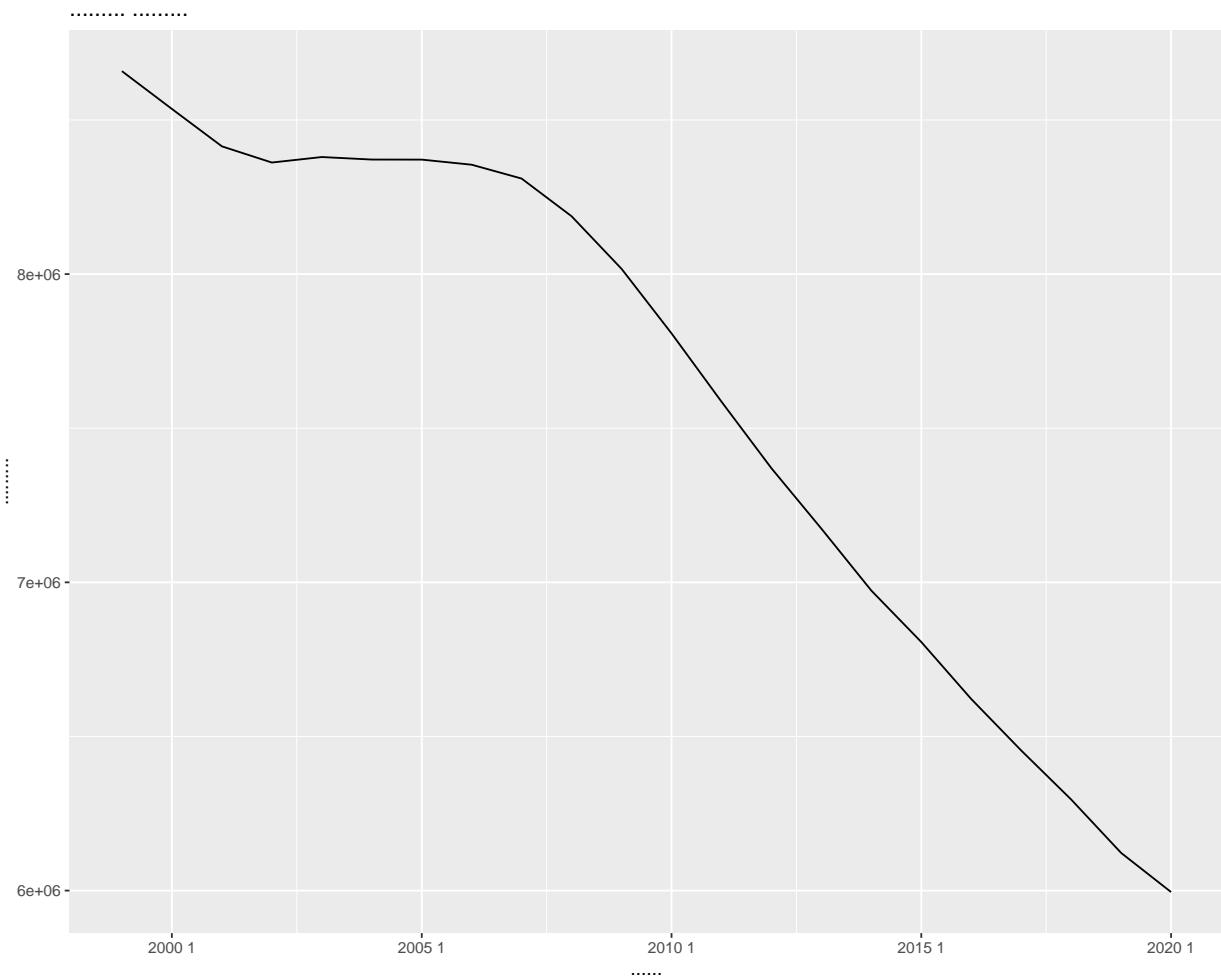
```
autoplot(employees.ts[,2], main = '월별 취업자수', xlab = '연도', ylab = '취업자수', series = '전체 취업자수')
```



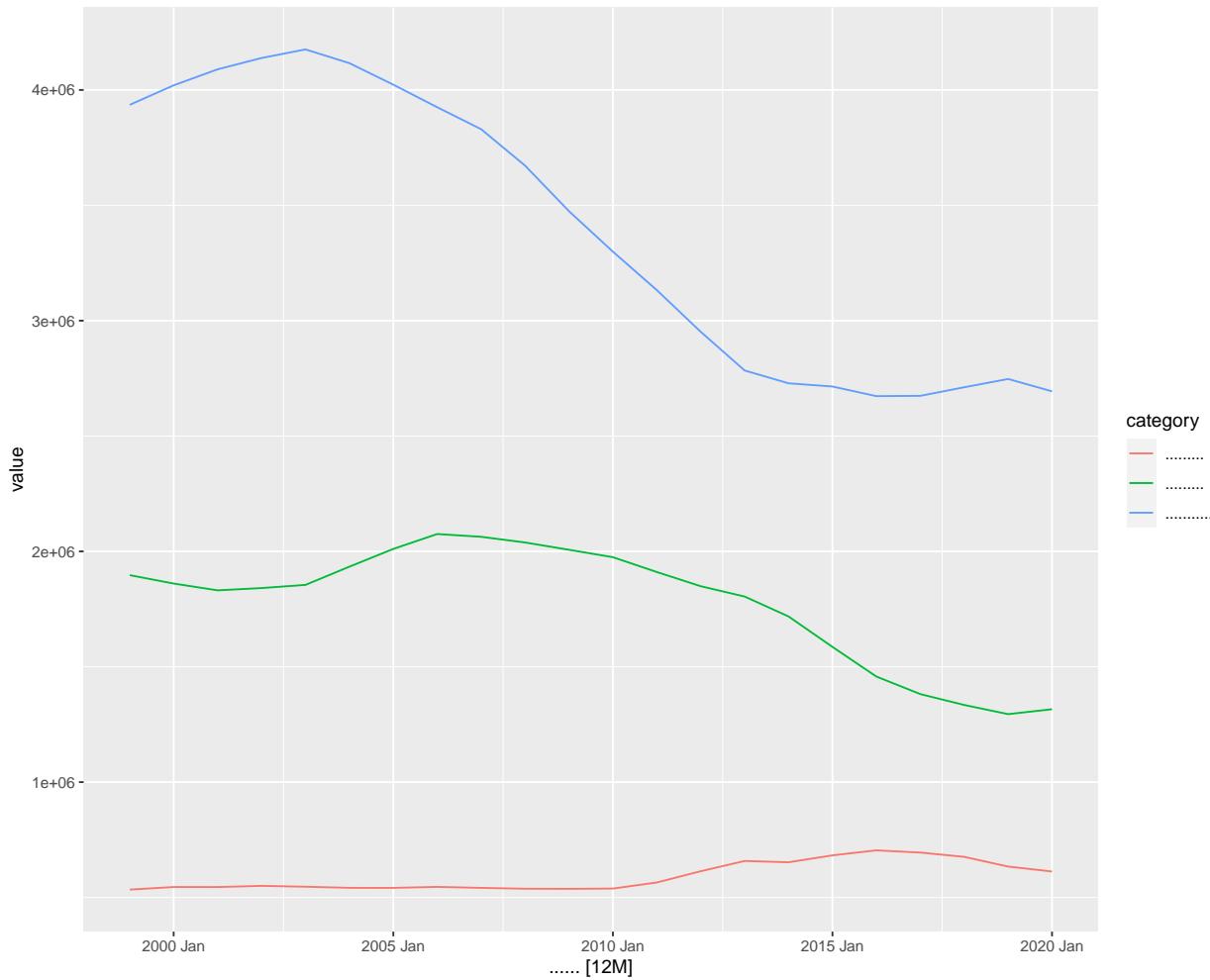


tsibble: feasts 패키지

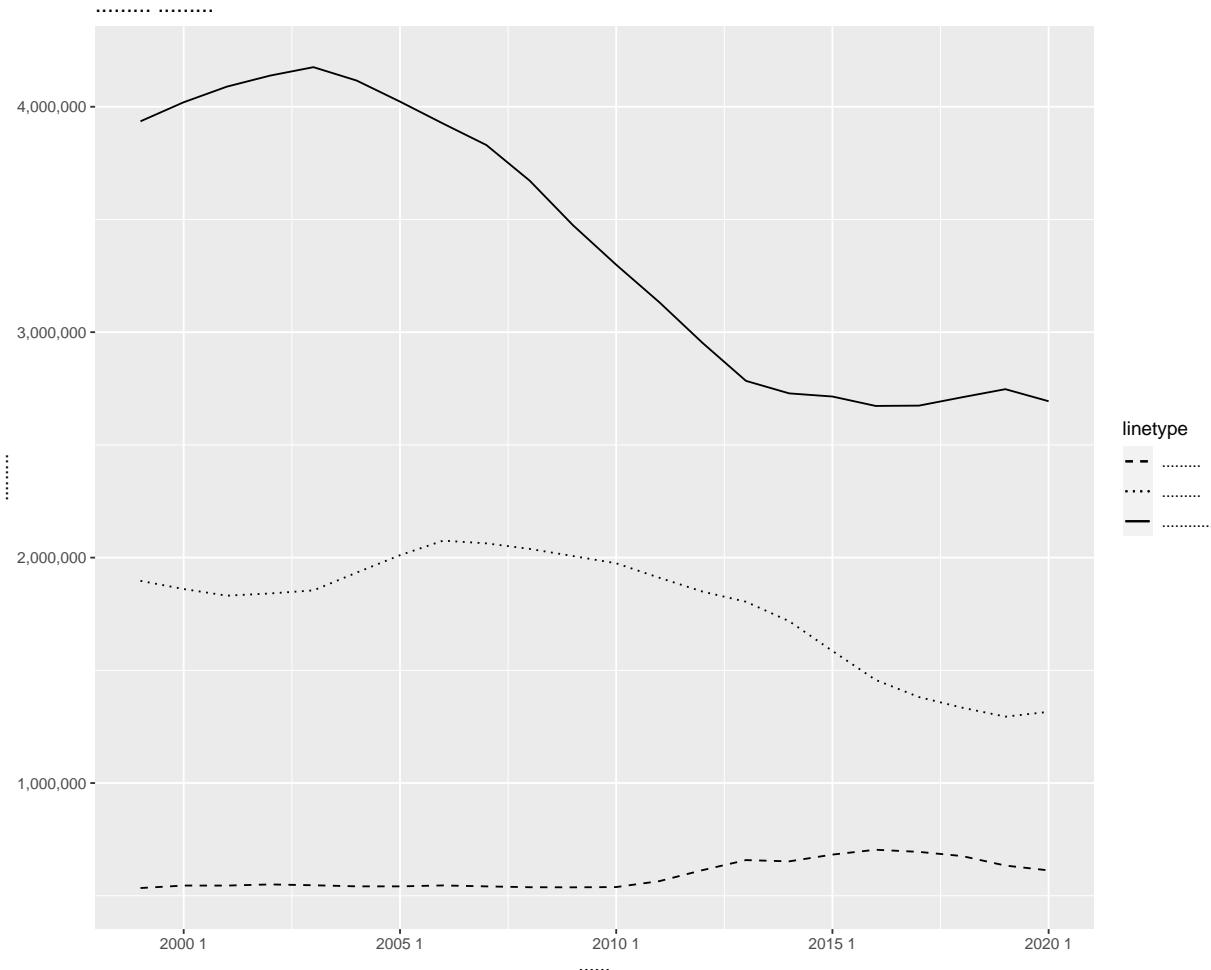
```
students.tsibble %>% autoplot(학생수계) +
  labs(title = '연도별 학생수', x = '연도', y = '학생수')
```



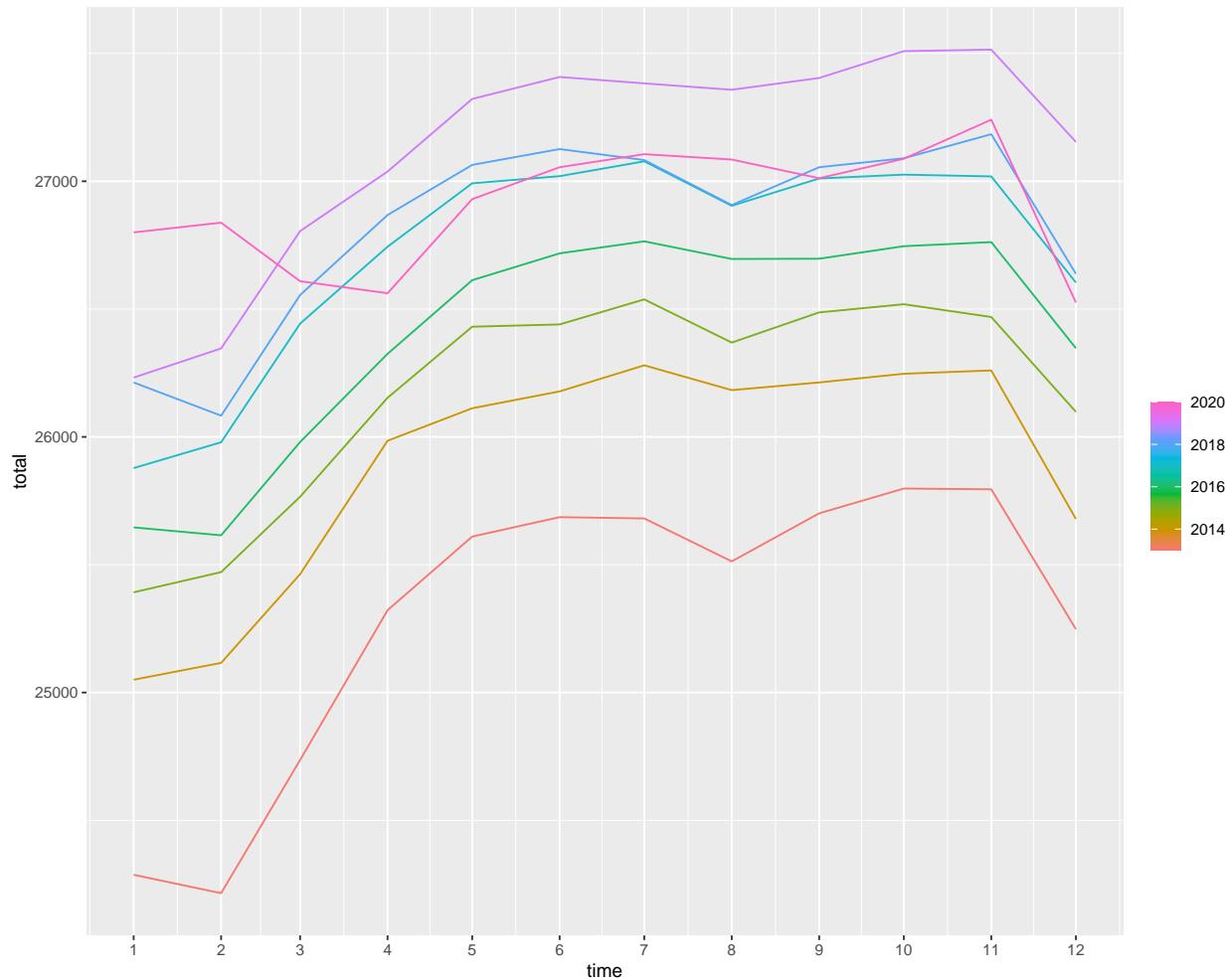
```
students.tsibble %>%
  select(1, 3, 4, 5) %>%
  tidyr::gather(category, value, 2:4) %>%
  autoplot()
```



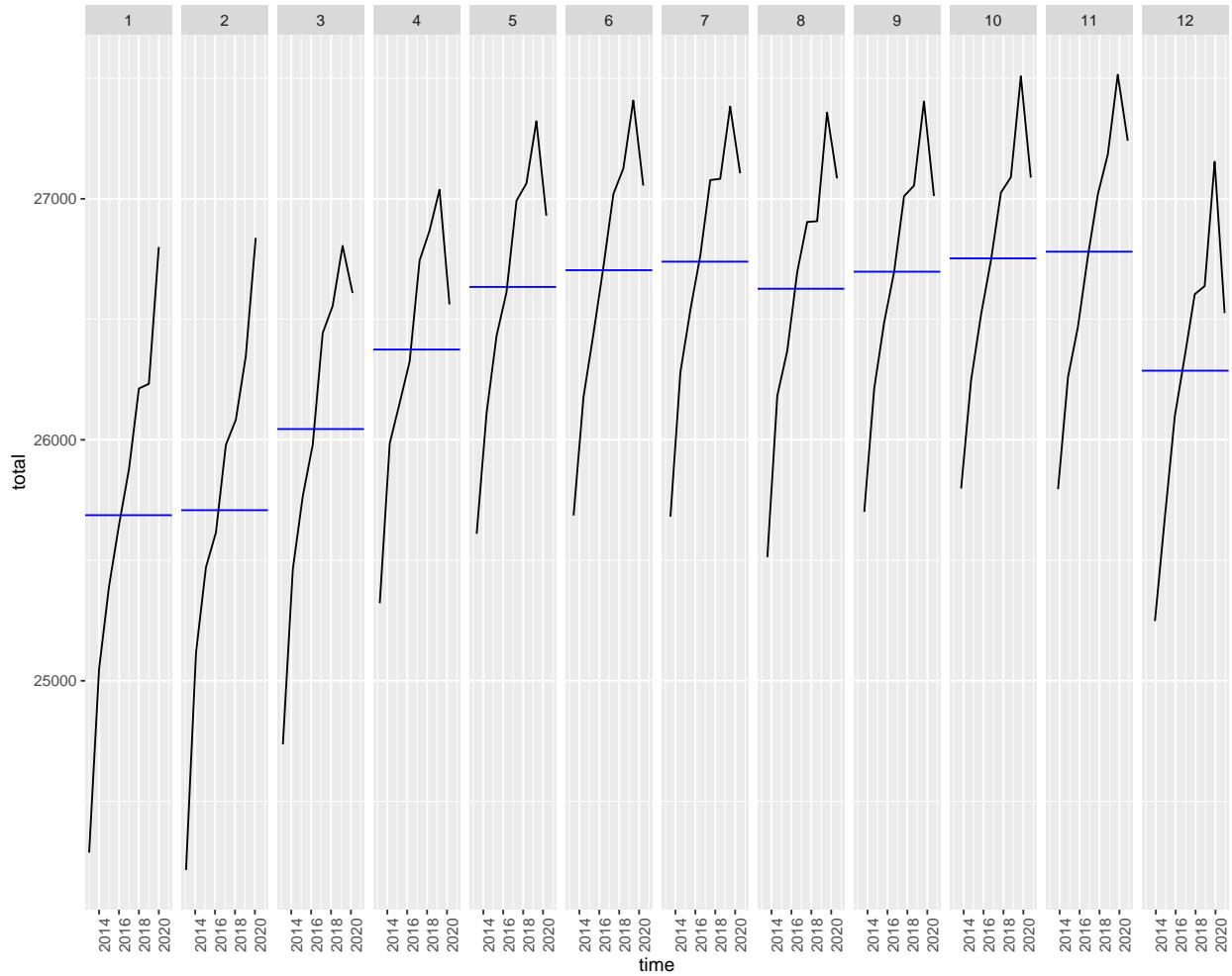
```
ggplot(students.tsibble, aes(x = 연도)) +
  geom_line(aes(y = 초등학교, group = 1, linetype = '초등학교')) +
  geom_line(aes(y = 유치원, group = 1, linetype = '유치원')) +
  geom_line(aes(y = 중학교, group = 1, linetype = '중학교')) +
  labs(title = '연도별 학생수', x = '연도', y = '학생수', color = '학교급') +
  scale_y_continuous(labels = scales::number_format(big.mark = ',')) +
  scale_linetype_manual(values = c('초등학교' = 1, '유치원' = 2, '중학교' = 3))
```



```
employees.tsibble %>% mutate(time = yearmonth(employees.tsibble$time)) %>%
  gg_season(total)
```



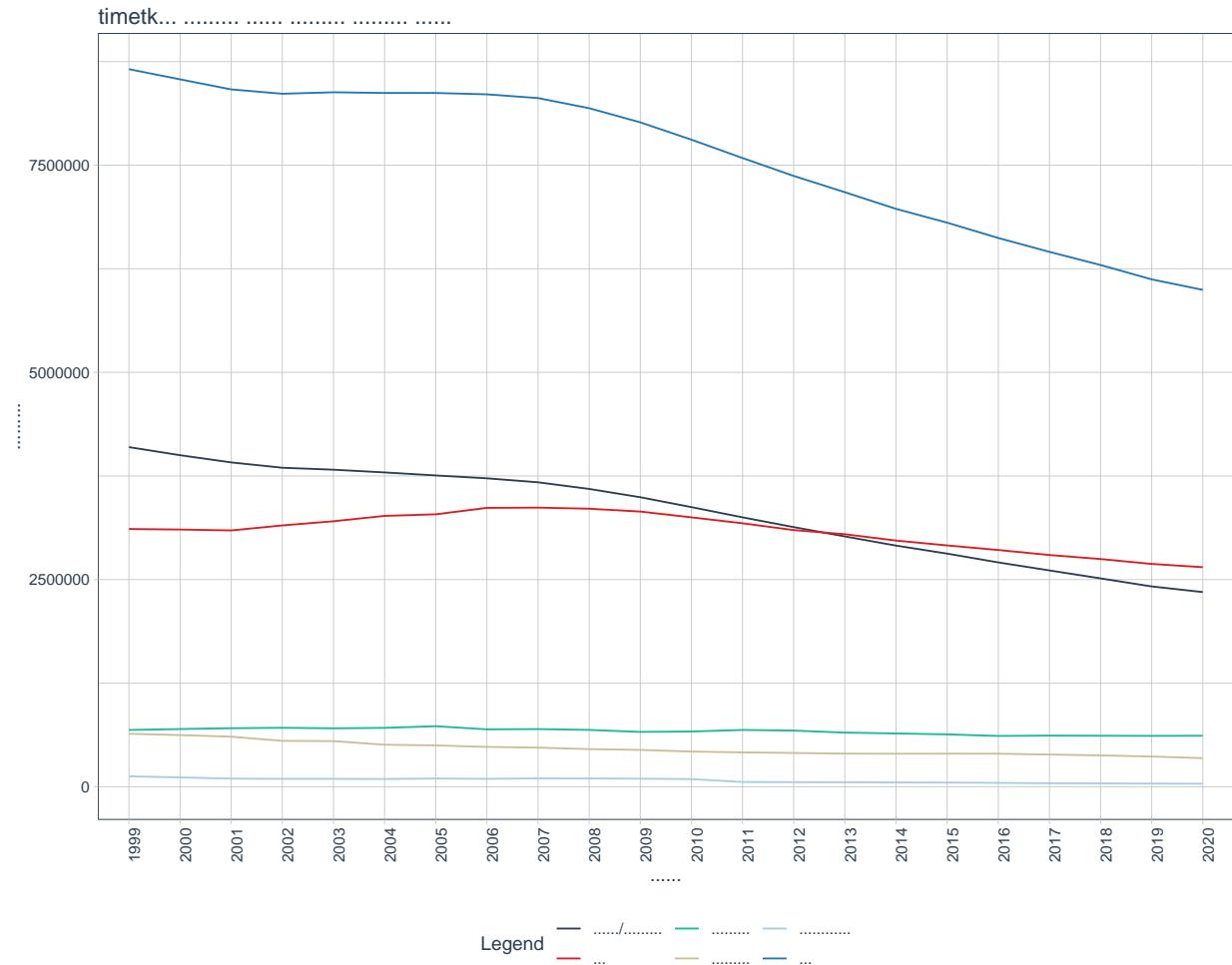
```
employees.tsibble %>% mutate(time = yearmonth(employees.tsibble$time)) %>%
  gg_subseries(total)
```



data.frame:timetk 패키지

```
students %>%
  plot_time_series(.date_var = 연도, .value = 학생수계, .smooth = T, .line_type = 2, .smooth_size = 0.5)

students.all %>%
  plot_time_series(.date_var = 연도, .value = 학생수계, .color_var = 지역규모, .smooth = F, .title= 'time'
```



```
students %>% select(1, 3, 4, 5) %>%
  tidyr::gather(category, value, 2:4) %>%
  plot_time_series(.date_var = 연도, .value = value, .color_var = category, .smooth = F, .title = 'timet'

employees %>%
  plot_time_series(.date_var = time, .value = total, .smooth = F, .title = '월별 신규 취업자수', .x_lab = 

covid19 %>%
  plot_time_series(.date_var = date, .value = `0~9세`, .smooth = F, .title = '일별 코로나 확진자수(0~9세)' )
```

시간 정보 추출

```
(now.date <- Sys.time())
## [1] "2023-07-02 22:29:31 KST"
(now.char <- as.character(Sys.time()))
## [1] "2023-07-02 22:29:31.980972"
paste0('오늘은 ', year(now.date), '년 ', month(now.char), '월 ', day(now.date), '일입니다')
## [1] "오늘은 2023년 7월 2일입니다"
```

```

paste0('1월 1일부터 오늘까지 ', yday(now.date), '일 지났습니다')

## [1] "1월 1일부터 오늘까지 183일 지났습니다"
paste0('이번 분기 시작일부터 오늘까지 ', qday(now.date), '일 지났습니다')

## [1] "이번 분기 시작일부터 오늘까지 2일 지났습니다"
paste0('오늘은 ', wday(now.date, label = T, abbr = T), '요일입니다')

## [1] "오늘은 Sun요일입니다"
paste0('지금은 ', hour(now.date), '시 ', minute(now.char), '분 ', second(now.date), '초입니다')

## [1] "지금은 22시 29분 31.9802129268646초입니다"
paste0('이번 주는 올해의 ', week(now.date), '번째 주입니다')

## [1] "이번 주는 올해의 27번째 주입니다"

```

시간 연산

```

# 1980년 1월 1일부터 2021년 1월 1일까지의 날짜 수
as.Date('2021-01-01') - as.Date('1980-01-01')

## Time difference of 14976 days

# 오늘 날짜를 today에 저장
today <- today()
# 오늘부터 100일 후
today + 100

## [1] "2023-10-10"

# 오늘부터 2개월 전
today - months(2)

## [1] "2023-05-02"

# 오늘부터 1년 전
today - years(1)

## [1] "2022-07-02"

# 1980.1.1부터 2021.12.31까지의 interval을 int에 저장
# 결과값을 보면 우리가 생각하는 형태가 아님
(int <- lubridate::interval(as.Date('1980-01-01'), as.Date('2021-12-31')))

## [1] 1980-01-01 UTC--2021-12-31 UTC

# 연월일 형태로 interval 출력
lubridate::as.period(int)

## [1] "41y 11m 30d 0H 0M 0S"

# 경과 초 형태로 interval 출력
lubridate::as.duration(int)

## [1] "1325376000s (~42 years)"

```

```

# 1980.1.1부터 2021.12.31까지의 interval 클래스를 int1에 저장
int1 <- '1980-01-01' %--% '2021-12-31'
# 연월일 형태로 interval 출력
lubridate::as.period(int1)

## [1] "41y 11m 30d 0H 0M 0S"

# 2020년은 윤년
leap_year(2020)

## [1] TRUE

# 2020-01-01부터 기간상 1년 후(period)는 우리의 상식대로 2021-01-01
as.Date('2020-01-01') + years(1)

## [1] "2021-01-01"

# 2020-01-01부터 시간상 1년 후(duration)는 2020년은 윤년이므로 2020년은 366일임. 그래서 365일 후인 2020-12-31
as.Date('2020-01-01') + dyears(1)

## [1] "2020-12-31 06:00:00 UTC"

# 2020-02-01부터 한 달 후(period)는 2020년 3월 1일
as.Date('2020-02-01') + months(1)

## [1] "2020-03-01"

# 2020-02-01부터 한 달 후(duration)는 30일 후인 2020년 3월 2일
as.Date('2020-02-01') + dmonths(1)

## [1] "2020-03-02 10:30:00 UTC"

# 2021-02-01부터 한 달 후(period)는 2021년 3월 1일
as.Date('2021-02-01') + months(1)

## [1] "2021-03-01"

# 2020-01-01부터 한 달 후(duration)는 30일 후인 3월 2일
as.Date('2021-02-01') + dmonths(1)

## [1] "2021-03-03 10:30:00 UTC"

```

시간 반올림

```

(x <- as.Date("2020-11-12 13:45:40"))

## [1] "2020-11-12"

# 주 단위로 반올림
round_date(x, "week")

## [1] "2020-11-15"

# 주 단위로 내림
floor_date(x, "week")

## [1] "2020-11-08"

# 주 단위로 올림
ceiling_date(x, "week")

```

```

## [1] "2020-11-15"
# 월 단위로 반올림
round_date(x, "month")

## [1] "2020-11-01"
# 월 단위로 내림
floor_date(x, "month")

## [1] "2020-11-01"
# 월 단위로 올림
ceiling_date(x, "month")

## [1] "2020-12-01"
# 연 단위로 반올림
round_date(x, "year")

## [1] "2021-01-01"
# 연 단위로 내림
floor_date(x, "year")

## [1] "2020-01-01"
# 연 단위로 올림
ceiling_date(x, "year")

## [1] "2021-01-01"
# 말일을 구하는 코드
days_in_month(as.Date('2012-03-01'))

## Mar
## 31

```

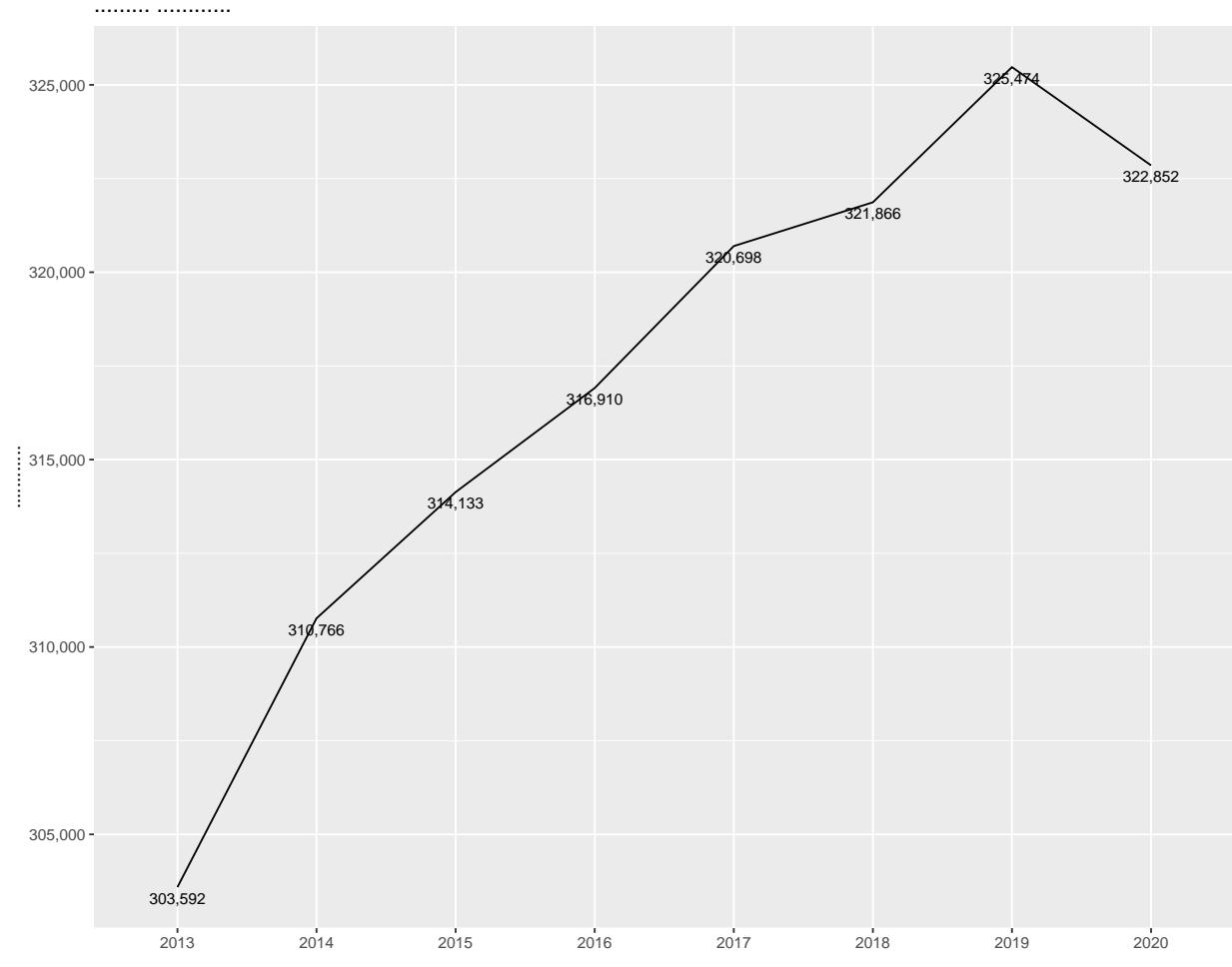
주간, 월간 합계 및 평균

```

# 월별 취업자수를 연별 취업자수로 그루핑
employees_by_year <-
  employees %>%
    mutate(year = year(time)) %>%
    group_by(year) %>%
    summarise(total.year = sum(total), employees.edu = sum(employees.edu))

employees_by_year %>%
  ggplot(aes(as.factor(year), total.year)) +
  geom_line(aes(group = 1)) +
  geom_text(aes(label = scales::number(total.year, big.mark = ',')), size = 3, vjust = 1.5) +
  labs(title = '연도별 취업자수', x = '연도', y = '취업자수') +
  scale_y_continuous(labels = scales::number_format(big.mark = ','))

```



```
# 일별 평균 확진자수를 산출
```

```
(mean.covid19.by.age <- covid19 %>%
  mutate(yearmon = yearmonth(date)) %>%
  group_by(yearmon) %>%
  summarise(`01대` = mean(`0-9세`),
            `10대` = mean(`10-19세`),
            `20대` = mean(`20-29세`),
            `30대` = mean(`30-39세`),
            `40대` = mean(`40-49세`),
            `50대` = mean(`50-59세`),
            `60대` = mean(`60-69세`),
            `70대` = mean(`70-79세`),
            `80대` = mean(`80세 이상`)))
```

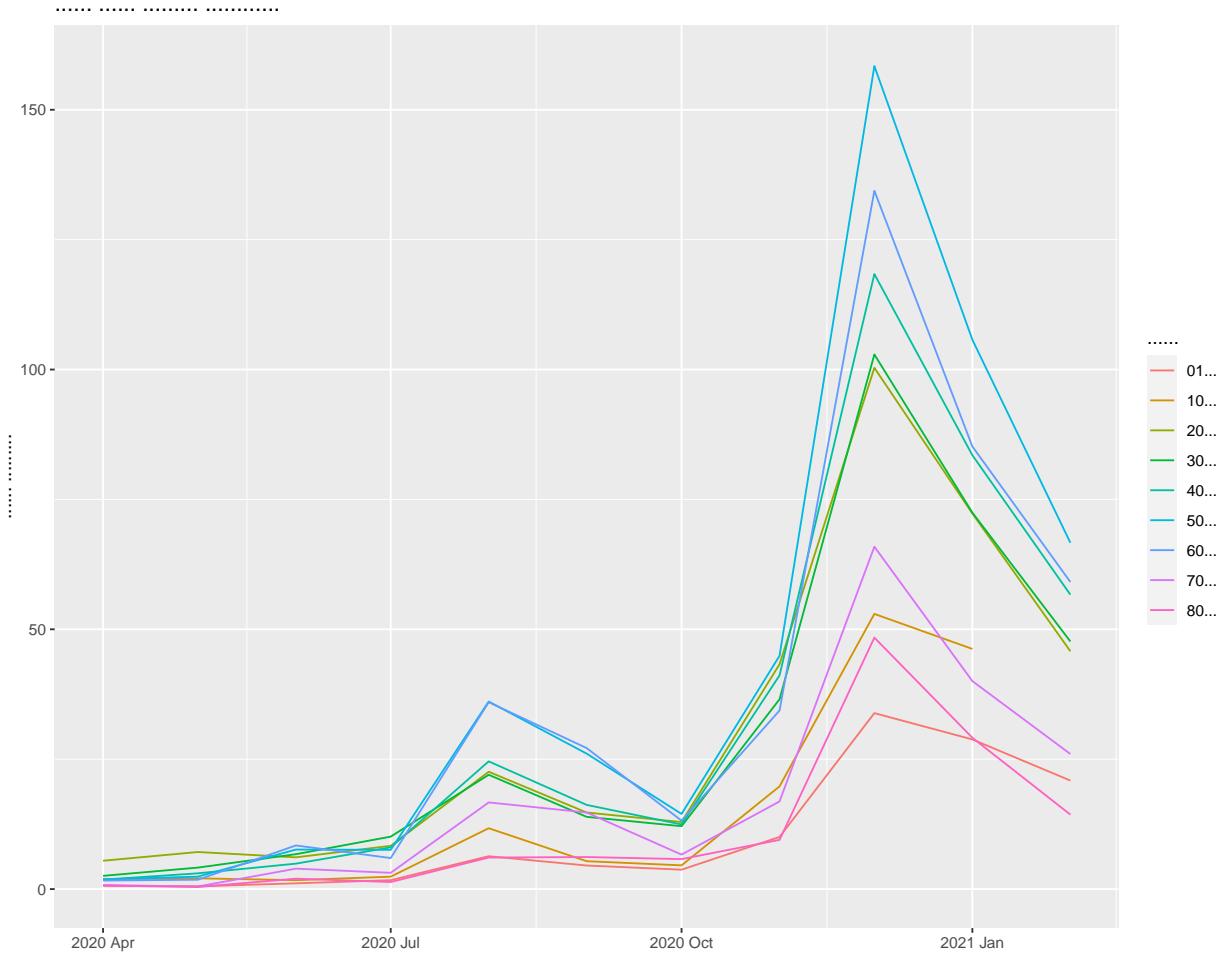
```
## # A tibble: 11 x 10
##   yearmon `01대` `10대` `20대` `30대` `40대` `50대` `60대` `70대` `80대`
##   <mth>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 2020 Apr  0.636   1.91   5.45   2.55   1.82   1.86   1.64  0.773  0.682
## 2 2020 May  0.533   2.07   7.13   4.13   3.03   2.4    1.8   0.5    0.4
## 3 2020 Jun  1.11    1.71   6.11   6.71   4.89   7.61   8.39  3.93   2
## 4 2020 Jul  1.71    2.39   8.32  10.1    8.03   7.55   5.97  3.13   1.35
## 5 2020 Aug  6.32   11.7   22.6    22     24.6   36.1   36.0  16.7   6.06
```

```

## 6 2020 Sep 4.53    5.37 14.7   13.9   16.2   26.1   27.1   14.7   6.17
## 7 2020 Oct 3.73    4.57 12.9   12.1   12.4   14.5   13.2   6.63   5.77
## 8 2020 Nov 10.0    19.8 43.3   36.5   41.1   44.9   34.4   16.9   9.47
## 9 2020 Dec 33.9    53.0 100.    103.    118.    158.    134.    65.9   48.4
## 10 2021 Jan 28.8    46.2 72.3   72.5   83.5   106.    85.2   40.0   29.1
## 11 2021 Feb 20.9    NA     45.8 47.7   56.7   66.7   59.1   26     14.3

mean.covid19.by.age %>%
  tidyr::gather(category, value, 2:10) %>%
  ggplot(aes(x = yearmon, y = value)) +
  geom_line(aes(group = category, color = category)) +
  labs(title = '월간 평균 코로나 확진자수', x = '시간', y = '평균 확진자', color = '세대')

```



```

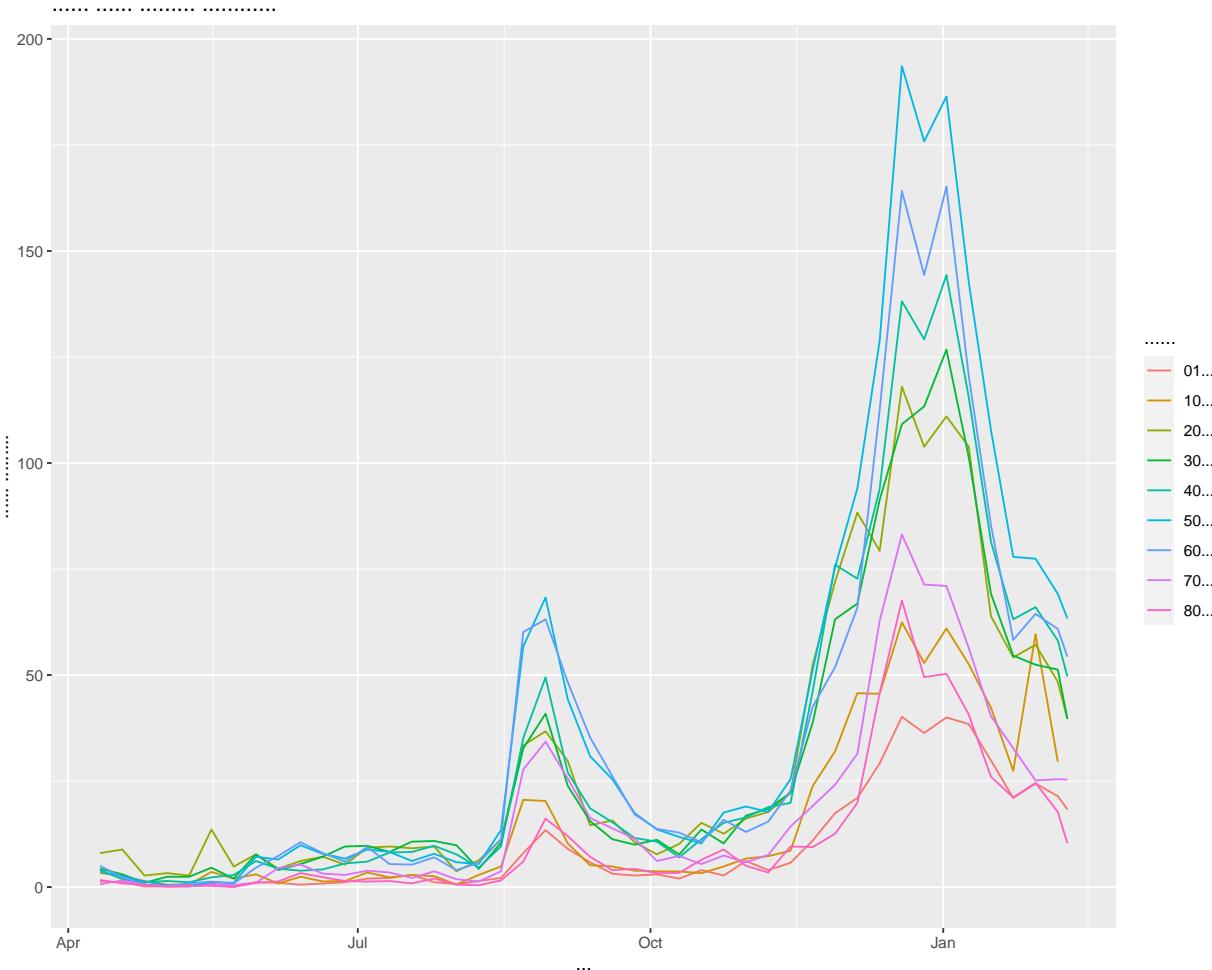
as_tbl_time(covid19, index = date) %>%
  collapse_by('weekly') %>%
  group_by(date) %>%
  summarise(`01대` = mean(`0-9세`),
            `10대` = mean(`10-19세`),
            `20대` = mean(`20-29세`),
            `30대` = mean(`30-39세`),
            `40대` = mean(`40-49세`),
            `50대` = mean(`50-59세`),
            ...

```

```

`60대` = mean(`60-69세`),
`70대` = mean(`70-79세`),
`80대` = mean(`80세 이상`)) %>%
tidyrr::gather(category, value, 2:10) %>%
ggplot(aes(x = date, y = value)) +
geom_line(aes(group = category, color = category)) +
labs(title = '주간 평균 코로나 확진자수', x = '월', y = '평균 확진자', color = '세대')

```

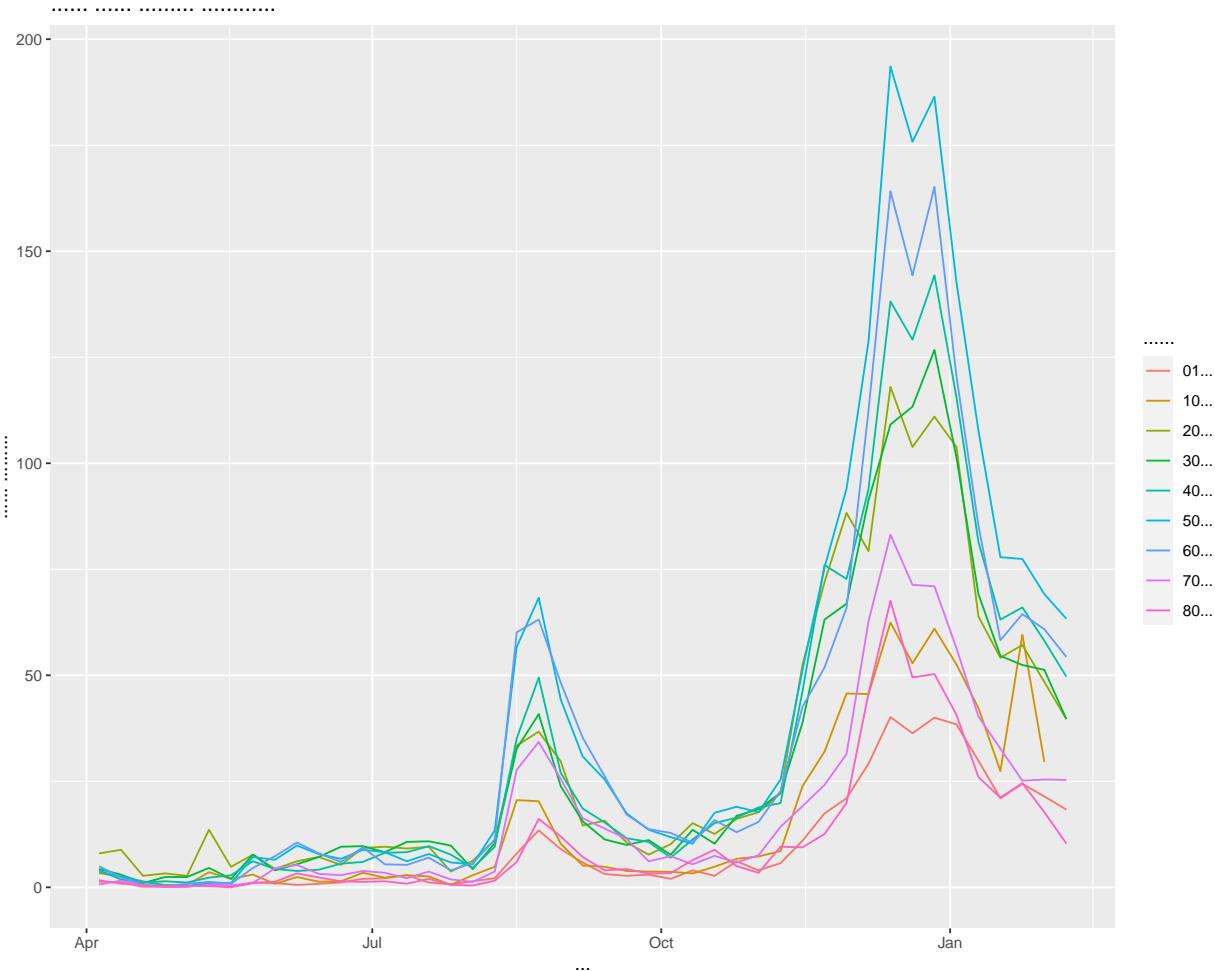


```

covid19 %>%
summarise_by_time(.date_var = date, .by = 'week',
`01대` = mean(`0-9세`),
`10대` = mean(`10-19세`),
`20대` = mean(`20-29세`),
`30대` = mean(`30-39세`),
`40대` = mean(`40-49세`),
`50대` = mean(`50-59세`),
`60대` = mean(`60-69세`),
`70대` = mean(`70-79세`),
`80대` = mean(`80세 이상`)) %>%
tidyrr::gather(category, value, 2:10) %>%
ggplot(aes(x = date, y = value)) +

```

```
geom_line(aes(group = category, color = category)) +
  labs(title = '주간 평균 코로나 확진자수', x = '월', y = '평균 확진자', color = '세대')
```



```
employees %>%
  summarise_by_time(.date_var = time, .by = 'month',
                     total.year = sum(total),
                     employees.edu = sum(employees.edu)) %>%
  head(10)
```

```
## # A tibble: 10 x 3
##   time      total.year employees.edu
##   <date>        <int>         <int>
## 1 2013-01-01     24287        1710
## 2 2013-02-01     24215        1681
## 3 2013-03-01     24736        1716
## 4 2013-04-01     25322        1745
## 5 2013-05-01     25610        1774
## 6 2013-06-01     25686        1786
## 7 2013-07-01     25681        1813
## 8 2013-08-01     25513        1811
## 9 2013-09-01     25701        1794
```

```

## 10 2013-10-01      25798      1790
employees.tsibble%>%
  index_by(yearqtr = ~ yearquarter(.)) %>%
  summarise(sum.qtrly = sum(total)) %>%
  head(10)

## # A tsibble: 10 x 2 [1Q]
##   yearqtr sum.qtrly
##   <qtr>     <int>
## 1 2013 Q1    73238
## 2 2013 Q2    76618
## 3 2013 Q3    76895
## 4 2013 Q4    76841
## 5 2014 Q1    75629
## 6 2014 Q2    78275
## 7 2014 Q3    78676
## 8 2014 Q4    78186
## 9 2015 Q1    76629
## 10 2015 Q2   79024

covid19.tsibble[, c(1,3)]%>%
  index_by(yearweek = ~ yearweek(.)) %>%
  summarise(sum.weekly = sum(`0-9세`)) %>%
  head(10)

## # A tsibble: 10 x 2 [1W]
##   yearweek sum.weekly
##   <week>     <dbl>
## 1 2020 W15      4
## 2 2020 W16      8
## 3 2020 W17      3
## 4 2020 W18     -1
## 5 2020 W19      1
## 6 2020 W20      6
## 7 2020 W21      1
## 8 2020 W22      8
## 9 2020 W23      7
## 10 2020 W24     3

covid19.tsibble[, c(1,3)]%>%
  index_by(twomonth = ~ lubridate::floor_date(., "2 month")) %>%
  summarise(sum.2month = sum(`0-9세`)) %>%
  head(10)

## # A tsibble: 6 x 2 [1D]
##   twomonth  sum.2month
##   <date>     <dbl>
## 1 2020-03-01     14
## 2 2020-05-01     47
## 3 2020-07-01    249
## 4 2020-09-01    248
## 5 2020-11-01   1317
## 6 2021-01-01   1081

covid19.tsibble[, c(1,3)]%>%
  index_by(fourday = ~ lubridate::floor_date(., "4 day")) %>%

```

```

summarise(sum.4days = sum(`0-9세`)) %>%
head(10)

## # A tsibble: 10 x 2 [1D]
##   fourday    sum.4days
##   <date>      <dbl>
## 1 2020-04-09      4
## 2 2020-04-13      5
## 3 2020-04-17      3
## 4 2020-04-21      0
## 5 2020-04-25      2
## 6 2020-04-29      0
## 7 2020-05-01      0
## 8 2020-05-05      1
## 9 2020-05-09      1
## 10 2020-05-13     5

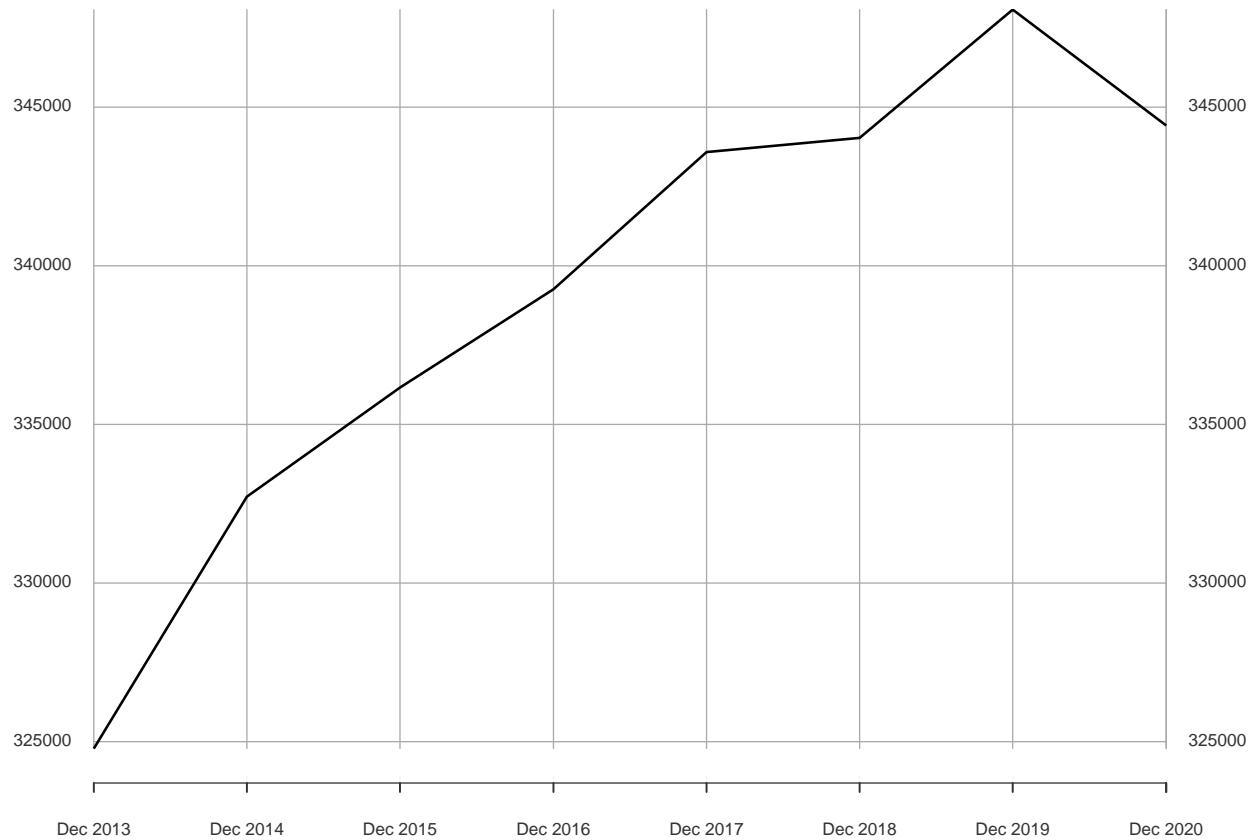
apply.quarterly(employees.xts, sum) %>%
head(10)

##          [,1]
## 2013-03-01 78345
## 2013-06-01 81923
## 2013-09-01 82313
## 2013-12-01 82203
## 2014-03-01 80977
## 2014-06-01 83779
## 2014-09-01 84226
## 2014-12-01 83741
## 2015-03-01 82095
## 2015-06-01 84488

apply.yearly(employees.xts, sum) %>%
plot.xts()

```

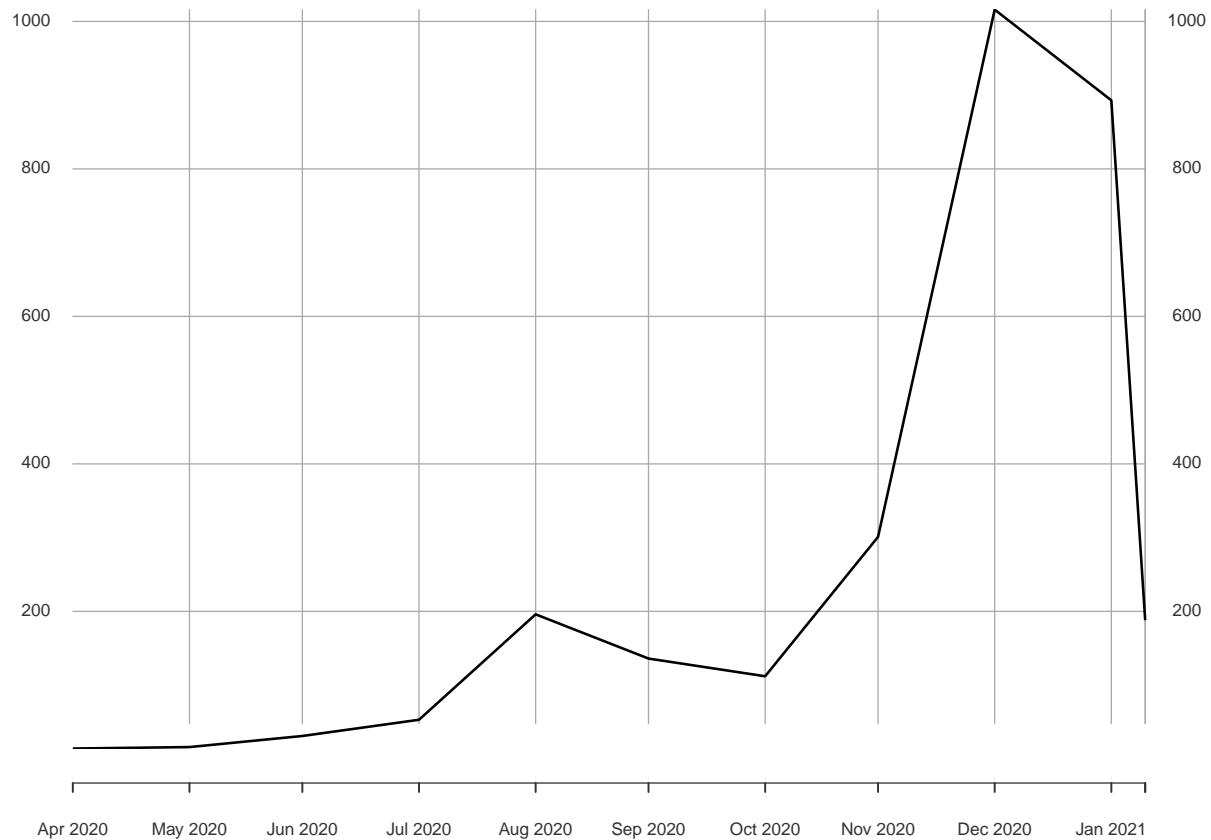
2013-12-01 / 2020-12-01



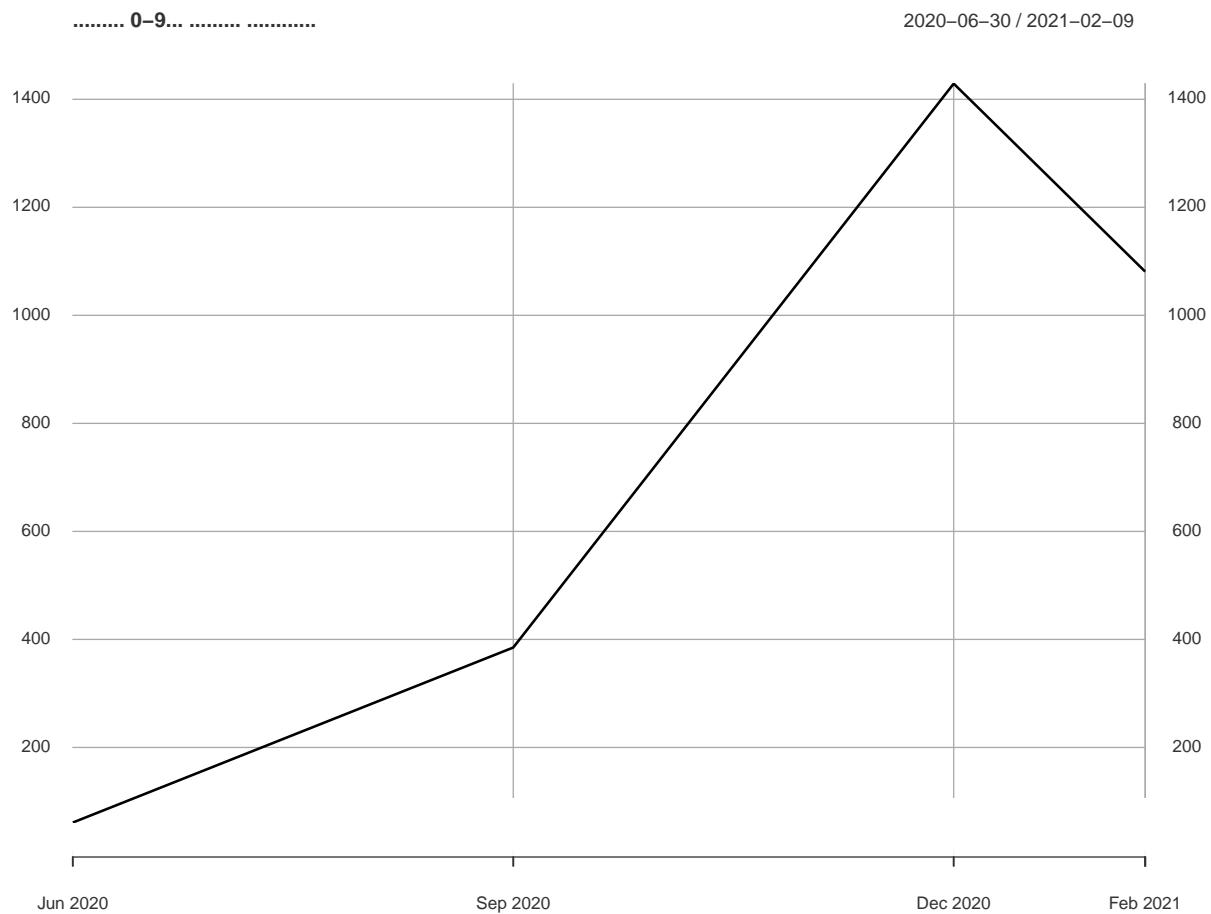
```
apply.monthly(covid19.xts[,1], sum) %>%
  plot.xts(main = '월별 0-9세 코로나 확진자수')
```

..... 0-9...

2020-04-30 / 2021-02-09



```
apply.quarterly(covid19.xts[,1], sum) %>%
  plot.xts(main = '분기별 0-9세 코로나 확진자수')
```



OHLC : 주식 시가, 고가, 저가, 종가

```
as_tbl_time(covid19, index = date) %>%
  collapse_by('weekly') %>%
  group_by(date) %>%
  summarise(Open = first(`0-9세`),
            High = max(`0-9세`),
            Low = min(`0-9세`),
            Close = last(`0-9세`)) %>%
  head(10)

## # A time tibble: 10 x 5
## # Index:           date
## #   date      Open  High  Low Close
## #   <date>    <dbl> <dbl> <dbl> <dbl>
## 1 2020-04-11     2     2     1     1
## 2 2020-04-18     0     3     0     2
## 3 2020-04-25     0     1     0     1
## 4 2020-05-02     2     2    -1     0
## 5 2020-05-09     0     1     0     0
## 6 2020-05-16     0     2     0     2
```

```

## 7 2020-05-23 0 1 0 0
## 8 2020-05-30 0 3 0 1
## 9 2020-06-06 0 3 0 3
## 10 2020-06-13 1 1 0 0
to.period(covid19.xts, method = 'months', OHLC = TRUE)

##               covid19.xts.Open covid19.xts.High covid19.xts.Low covid19.xts.Close
## 2020-04-30                  2                  5                  1                  2
## 2020-05-31                  0                 11                  0                  7
## 2020-06-30                  0                  6                  0                  6
## 2020-07-31                  0                 10                 -4                 13
## 2020-08-31                  1                 35                  0                 15
## 2020-09-30                 14                 17                  0                 20
## 2020-10-31                  6                 11                  0                 22
## 2020-11-30                  5                 57                  9                 32
## 2020-12-31                 15                 86                  0                132
## 2021-01-31                 41                138                 44                  49
## 2021-02-08                 20                 41                 30                  31

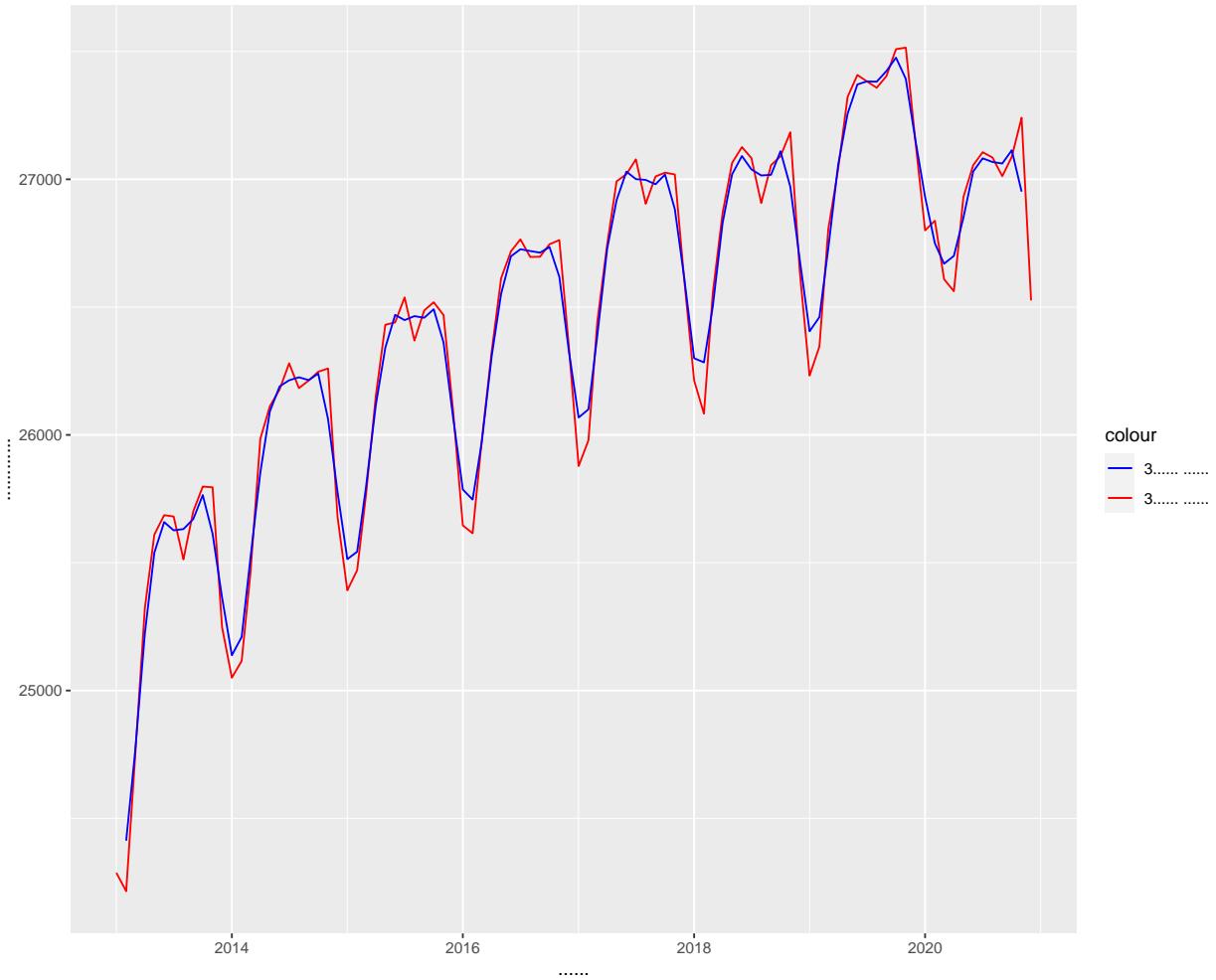
```

시간 롤링 : 3일 평균, 5일 합계

```

employees %>%
  mutate(ma3 = rollmean(total, k = 3, fill = NA),
        sum3 = rollapply(total, 3, sum, fill = NA)) %>%
  select(time, total, ma3, sum3) %>%
  ggplot(aes(x = time)) +
  geom_line(aes(y = total, group = 1, color = '3개월 합계')) +
  geom_line(aes(y = ma3, group = 1, color = '3개월 평균')) +
  labs(y = '취업자수', x = '연도') +
  scale_color_manual(values = c('3개월 합계' = 'red', '3개월 평균' = 'blue'))

```



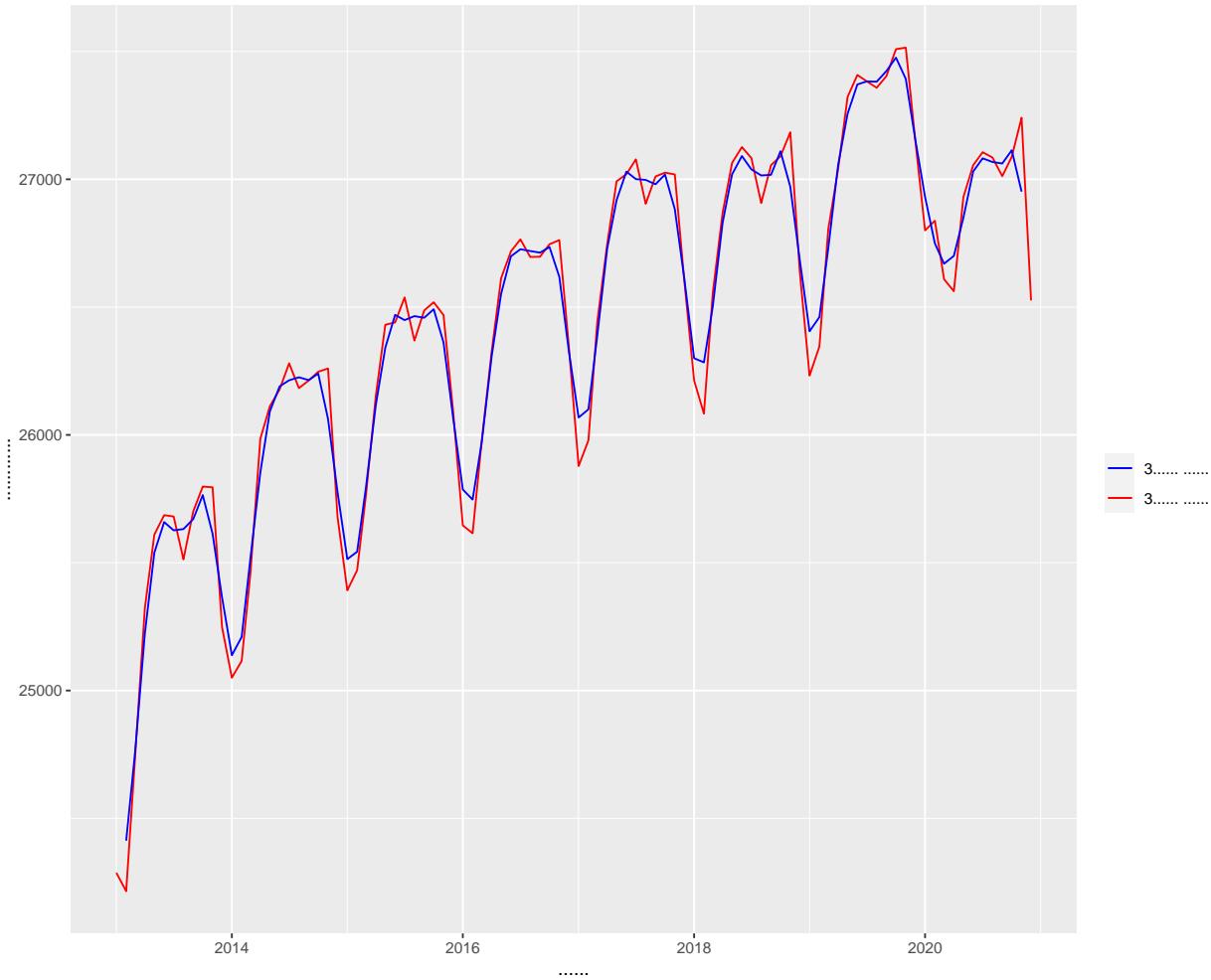
```

ma3 <- slidify(mean, .period = 3, .align = "center")
sum3 <- slidify(sum, .period = 3, .align = "center")
class(ma3)

## [1] "function"
class(sum3)

## [1] "function"
employees %>%
  mutate(ma3 = ma3(total), sum3 = sum3(total)) %>%
  select(time, total, ma3, sum3) %>%
  ggplot(aes(x = time)) +
  geom_line(aes(y = total, group = 1, color = '3개월 합계')) +
  geom_line(aes(y = ma3, group = 1, color = '3개월 평균')) +
  labs(y = '취업자수', x = '연도') +
  scale_color_manual(' ', values = c('3개월 합계' = 'red', '3개월 평균' = 'blue'))

```



```
rollapply(employees.xts, width = 3, FUN = mean) %>%
  head(10)
```

```
##          total employees.edu
## 2013-01-01      NA         NA
## 2013-02-01      NA         NA
## 2013-03-01 24412.67    1702.333
## 2013-04-01 24757.67    1714.000
## 2013-05-01 25222.67    1745.000
## 2013-06-01 25539.33    1768.333
## 2013-07-01 25659.00    1791.000
## 2013-08-01 25626.67    1803.333
## 2013-09-01 25631.67    1806.000
## 2013-10-01 25670.67    1798.333
```

필터링 : 지난 달 데이는?

```
covid19 %>%
  filter(date >= as.Date('2020-10-01') & date <= as.Date('2020-10-10'))
```

	date	status	0-9세	10-19세	20-29세	30-39세	40-49세	50-59세	60-69세
## 1	2020-10-01	신규	6	4	7	9	14	13	13

```

## 2 2020-10-02 신규 3 6 6 10 6 11 13
## 3 2020-10-03 신규 1 1 6 13 11 11 16
## 4 2020-10-04 신규 0 4 4 9 13 8 12
## 5 2020-10-05 신규 0 3 17 9 8 12 14
## 6 2020-10-06 신규 5 8 26 8 2 11 9
## 7 2020-10-07 신규 1 2 8 13 10 28 31
## 8 2020-10-08 신규 6 5 6 7 9 12 11
## 9 2020-10-10 신규 0 0 0 0 0 0 0
## 70-79세 80세 이상
## 1 7 4
## 2 4 4
## 3 13 3
## 4 8 6
## 5 6 4
## 6 4 2
## 7 16 5
## 8 10 3
## 9 0 0

```

covid19 %>%

```
filter(between(date, as.Date('2021-01-01'), as.Date('2021-01-15')))
```

	date	status	0-9세	10-19세	20-29세	30-39세	40-49세	50-59세	60-69세
## 1	2021-01-01	신규	41	74	131	145	154	204	159
## 2	2021-01-02	신규	39	66	111	109	106	160	125
## 3	2021-01-03	신규	32	46	80	75	88	124	117
## 4	2021-01-04	신규	40	71	120	127	172	187	153
## 5	2021-01-05	신규	45	44	107	105	112	124	108
## 6	2021-01-06	신규	45	60	106	106	110	140	148
## 7	2021-01-07	신규	41	54	123	120	155	176	113
## 8	2021-01-08	신규	34	43	91	86	83	124	125
## 9	2021-01-09	신규	32	50	100	91	89	124	80
## 10	2021-01-10	신규	36	56	87	79	94	143	87
## 11	2021-01-11	신규	19	36	47	54	61	79	79
## 12	2021-01-12	신규	32	34	71	59	91	102	74
## 13	2021-01-13	신규	24	28	66	81	83	129	87
## 14	2021-01-14	신규	25	45	60	67	82	98	71
## 15	2021-01-15	신규	42	37	62	70	81	97	91
		70-79세 80세 이상							
## 1			80	41					
## 2			59	49					
## 3			51	38					
## 4			69	81					
## 5			44	26					
## 6			73	51					
## 7			59	27					
## 8			49	37					
## 9			50	25					
## 10			49	34					
## 11			44	31					
## 12			39	35					
## 13			47	16					
## 14			39	29					
## 15			22	11					

```

employees %>%
  filter(year(time) == 2019 & month(time) == 5)

##           time total employees.edu
## 1 2019-05-01    27322          1884

# 매월 3일부터 7일까지 필터링
covid19 %>%
  filter(between(day(date), 3, 7)) %>%
  head(15)

##           date status 0-9세 10-19세 20-29세 30-39세 40-49세 50-59세 60-69세
## 1 2020-05-03 신규     0      1      2      3      2      0      2
## 2 2020-05-04 신규     0      0      2      1      3      0      2
## 3 2020-05-05 신규     0      0      0      1      1      1      0
## 4 2020-05-06 신규     0      0      0      1      0      0      1
## 5 2020-05-07 신규     0      1      2      1      0      0      0
## 6 2020-06-03 신규     1      2      5      6      5      9      15
## 7 2020-06-04 신규     1      0      5      3      3      9      9
## 8 2020-06-05 신규     1      2      5      2      3      7      10
## 9 2020-06-06 신규     3      1      2      6      11      8      9
## 10 2020-06-07 신규     1      6      3      6      8      11      12
## 11 2020-07-03 신규     2     10     10     10      6      7      11
## 12 2020-07-04 신규     3      2     12     16      8      8      9
## 13 2020-07-05 신규     3      3      8      9      11      7      14
## 14 2020-07-06 신규     4      1      7     10      6      9      4
## 15 2020-07-07 신규     1      2     12      6      7      7      7
##           70-79세 80세 이상
## 1           1      2
## 2           0      0
## 3           0      0
## 4           0      0
## 5           0      0
## 6           6      0
## 7           6      3
## 8           7      2
## 9           8      3
## 10          8      2
## 11          4      3
## 12          2      3
## 13          6      0
## 14          2      3
## 15          2      0

covid19 %>%
  filter_by_time(.date_var = date, .start = '2020-10-01', .end = '2020-10-05')

##           date status 0-9세 10-19세 20-29세 30-39세 40-49세 50-59세 60-69세
## 1 2020-10-01 신규     6      4      7      9     14     13     13
## 2 2020-10-02 신규     3      6      6     10      6     11     13
## 3 2020-10-03 신규     1      1      6     13     11     11     16
## 4 2020-10-04 신규     0      4      4      9     13      8     12
## 5 2020-10-05 신규     0      3     17      9      8     12     14
##           70-79세 80세 이상
## 1           7      4

```

```

## 2      4      4
## 3     13     3
## 4      8      6
## 5      6      4

covid19 %>%
  filter(`0-9세` != 0) %>%
  filter_period(.date_var = date, .period = '1 month', `0-9세` == max(`0-9세`)) %>%
  head(10)

## # A tibble: 10 x 11
##   date      status `0-9세` `10-19세` `20-29세` `30-39세` `40-49세` `50-59세` `60-69세` `70-79세` `80세 이상`
##   <date>    <fct>   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 2020-04-16 신규      3        3        5        1        2        3
## 2 2020-05-27 신규      3        2        8        6        3        6
## 3 2020-06-06 신규      3        1        2        6       11        8
## 4 2020-06-23 신규      3        0        4        9        9        9
## 5 2020-06-29 신규      3        1        9        5        6        5
## 6 2020-06-30 신규      3        4       10        6        8        6
## 7 2020-07-17 신규     10        4       26       37       32       23
## 8 2020-08-23 신규     16       35       34       40       62       82
## 9 2020-08-28 신규     16       17       32       48       53       73
## 10 2020-08-29 신규    16       18       31       36       47       54
## # i 3 more variables: `60-69세` <dbl>, `70-79세` <dbl>, `80세 이상` <dbl>

# 2020-10-02에 해당하는 데이터 필터링
covid19.xts['2020-10-02']

##          0-9세 10-19세 20-29세 30-39세 40-49세 50-59세 60-69세 70-79세
## 2020-10-02    3      6      6     10      6     11     13      4

# 2020-10-01에서부터 2020-10-10까지 데이터 필터링
covid19.xts['2020-10-01/2020-10-10']

##          0-9세 10-19세 20-29세 30-39세 40-49세 50-59세 60-69세 70-79세
## 2020-10-01    6      4      7      9     14     13     13      7
## 2020-10-02    3      6      6     10      6     11     13      4
## 2020-10-03    1      1      6     13     11     11     16     13
## 2020-10-04    0      4      4      9     13      8     12      8
## 2020-10-05    0      3     17      9      8     12     14      6
## 2020-10-06    5      8     26      8      2     11      9      4
## 2020-10-07    1      2      8     13     10     28     31     16
## 2020-10-08    6      5      6      7      9     12     11     10
## 2020-10-10    0      0      0      0      0      0      0      0

# 2021-02-05일부터 끝까지 데이터 필터링
covid19.xts['2021-02-05/']

##          0-9세 10-19세 20-29세 30-39세 40-49세 50-59세 60-69세 70-79세
## 2021-02-05   30     28     30     60     42     65     75     25
## 2021-02-06   25     31     53     41     60     77     63     30
## 2021-02-07   21     27     51     47     44     79     66     26
## 2021-02-08   19      9     35     31     53     52     59     22
## 2021-02-09   15     NA     33     41     52     59     38     28

# 처음부터 2020-04-11까지의 필터링
covid19.xts['/2020-04-11']

```

```

##          0-9세| 10-19세| 20-29세| 30-39세| 40-49세| 50-59세| 60-69세| 70-79세|
## 2020-04-09     2        4       12        7        7        2        2        0
## 2020-04-10     1        1        7        4        2        3        6        2
## 2020-04-11     1        5        5        2        3        6        7        0

```

월별, 분기별, 연별 증감량

```

students_lag <- cbind(연도 = students$연도,
                      학생수계 = students$학생수계,
                      전년 = students %>%
                        lag(1) %>%
                        select(학생수계) %>%
                        rename(전년 = 학생수계)) %>%
  mutate(증감 = 학생수계 - 전년, 증감률 = round((학생수계/전년)-1, 3) * 100)

students_lag %>% head()

```

```

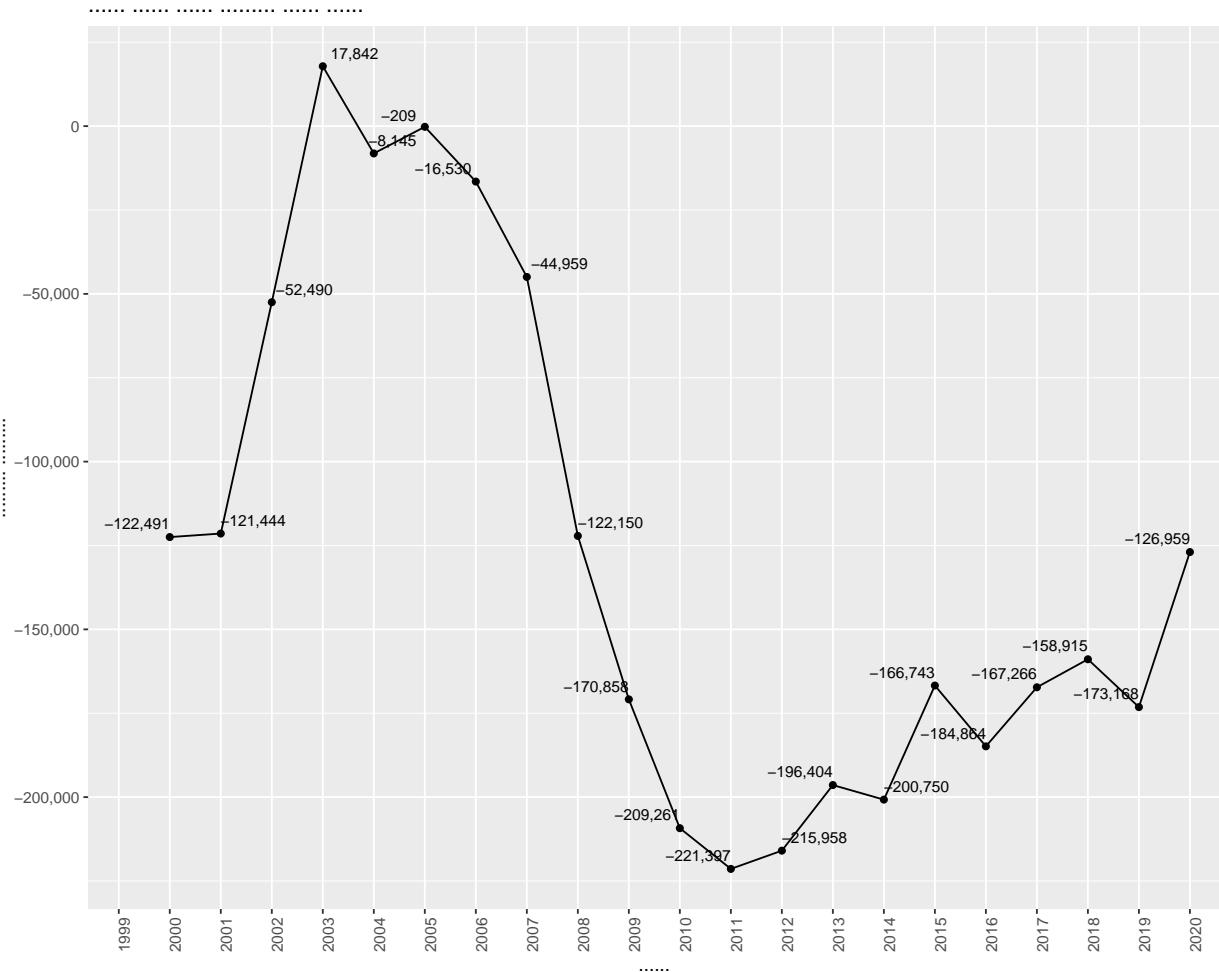
##      연도 학생수계   전년   증감 증감률
## 1 1999-01-01 8658358     NA     NA     NA
## 2 2000-01-01 8535867 8658358 -122491  -1.4
## 3 2001-01-01 8414423 8535867 -121444  -1.4
## 4 2002-01-01 8361933 8414423  -52490  -0.6
## 5 2003-01-01 8379775 8361933   17842   0.2
## 6 2004-01-01 8371630 8379775  -8145  -0.1

```

```

students_lag %>%
  ggplot(aes(as.factor(year(연도)), 증감)) +
  geom_line(aes(group = 1)) +
  geom_point() +
  ggrepel::geom_text_repel(aes(label = scales::comma(증감)), vjust = 1, size = 3) +
  # ggrepel::geom_text_repel() 함수로 숫자들이 겹치지 않게 시각화
  labs(title = '전년 대비 전체 학생수 증감 추이', x = '연도', y = '학생수 증감량') +
  scale_y_continuous(labels = scales::number_format(big.mark = ',')) +
  theme(axis.text.x=element_text(angle=90,hjust=1))

```



```
students.tsibble %>%
  select(1, 2) %>%
  mutate(증감 = difference(. $학생수계, lag = 1)) %>%
  mutate(증감률 = round((증감/학생수계), 3) * 100) %>% head(10)
```

```
## # A tsibble: 10 x 4 [12M]
##      연도 학생수계    증감 증감률
##   <mth>     <dbl>   <dbl>   <dbl>
## 1 1999 Jan  8658358     NA     NA
## 2 2000 Jan  8535867 -122491  -1.4
## 3 2001 Jan  8414423 -121444  -1.4
## 4 2002 Jan  8361933 -52490  -0.6
## 5 2003 Jan  8379775  17842   0.2
## 6 2004 Jan  8371630 -8145   -0.1
## 7 2005 Jan  8371421 -209    0
## 8 2006 Jan  8354891 -16530  -0.2
## 9 2007 Jan  8309932 -44959  -0.5
## 10 2008 Jan  8187782 -122150 -1.5
```

```
employees %>%
  mutate(증감 = difference(employees.tsibble$total, lag = 1)) %>%
  mutate(증감률 = round((증감/total), 3) * 100) %>% select(1, 2, 4, 5) %>% head(10)
```

```
##       time total 증감 증감률
```

```

## 1 2013-01-01 24287 NA NA
## 2 2013-02-01 24215 -72 -0.3
## 3 2013-03-01 24736 521 2.1
## 4 2013-04-01 25322 586 2.3
## 5 2013-05-01 25610 288 1.1
## 6 2013-06-01 25686 76 0.3
## 7 2013-07-01 25681 -5 0.0
## 8 2013-08-01 25513 -168 -0.7
## 9 2013-09-01 25701 188 0.7
## 10 2013-10-01 25798 97 0.4

students.xts$증감 <- diff(students.xts[,2])
students.xts$증감률 <- round((students.xts$증감/students.xts$학생수계), 3) * 100
students.xts[, c('유치원', '증감', '증감률')] %>% head(10)

##          유치원 증감 증감률
## 1999-01-01 534166 NA NA
## 2000-01-01 545263 11097 0.1
## 2001-01-01 545142 -121 0.0
## 2002-01-01 550256 5114 0.1
## 2003-01-01 546531 -3725 0.0
## 2004-01-01 541713 -4818 -0.1
## 2005-01-01 541603 -110 0.0
## 2006-01-01 545812 4209 0.1
## 2007-01-01 541550 -4262 -0.1
## 2008-01-01 537822 -3728 0.0

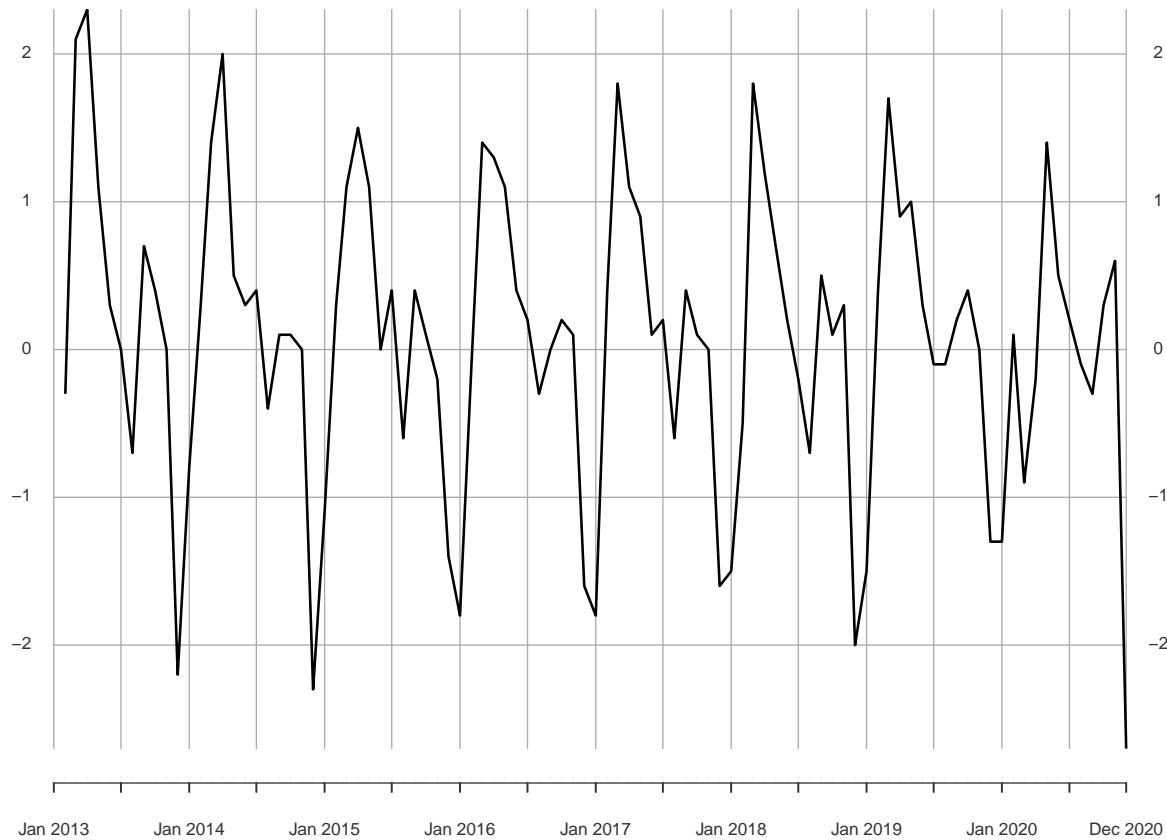
employees.xts$증감 <- diff(employees.xts$total)
employees.xts$증감률 <- round((employees.xts$증감/employees.xts$total), 3) * 100
employees.xts[, c('total', '증감', '증감률')] %>% head(10)

##          total 증감 증감률
## 2013-01-01 24287 NA NA
## 2013-02-01 24215 -72 -0.3
## 2013-03-01 24736 521 2.1
## 2013-04-01 25322 586 2.3
## 2013-05-01 25610 288 1.1
## 2013-06-01 25686 76 0.3
## 2013-07-01 25681 -5 0.0
## 2013-08-01 25513 -168 -0.7
## 2013-09-01 25701 188 0.7
## 2013-10-01 25798 97 0.4

plot.xts(employees.xts[, c('증감률')], main = '전월 대비 전체 취업자 증감률')

```

2013-01-01 / 2020-12-01



월 비중 백분율, 연 비중 백분율

```
employees %>%
  group_by(year(time)) %>%
  mutate(sum.by.year = sum(total)) %>%
  ungroup() %>%
  mutate(rate.by.year = round(total/sum.sum.year, 3) * 100) %>%
  head(15)
```

```
## # A tibble: 15 x 6
##   time      total employees.edu `year(time)` sum.sum.year rate.sum.year
##   <date>    <int>        <int>           <dbl>       <int>        <dbl>
## 1 2013-01-01 24287        1710        2013     303592         8
## 2 2013-02-01 24215        1681        2013     303592         8
## 3 2013-03-01 24736        1716        2013     303592        8.1
## 4 2013-04-01 25322        1745        2013     303592        8.3
## 5 2013-05-01 25610        1774        2013     303592        8.4
## 6 2013-06-01 25686        1786        2013     303592        8.5
## 7 2013-07-01 25681        1813        2013     303592        8.5
## 8 2013-08-01 25513        1811        2013     303592        8.4
## 9 2013-09-01 25701        1794        2013     303592        8.5
```

```

## 10 2013-10-01 25798      1790      2013    303592     8.5
## 11 2013-11-01 25795      1793      2013    303592     8.5
## 12 2013-12-01 25248      1779      2013    303592     8.3
## 13 2014-01-01 25050      1748      2014    310766     8.1
## 14 2014-02-01 25116      1786      2014    310766     8.1
## 15 2014-03-01 25463      1814      2014    310766     8.2

```

```

covid19 %>%
  group_by(yeарmonth(date)) %>%
  mutate(sum.by.month = sum(`0-9세`)) %>%
  ungroup() %>%
  mutate(rate.by.month = round(`0-9세`/sum.sum.month, 3) * 100) %>%
  select(date, `0-9세`, sum.sum.month, rate.sum.month)

```

```

## # A tibble: 302 x 4
##   date      `0-9세` sum.sum.month rate.sum.month
##   <date>     <dbl>        <dbl>          <dbl>
## 1 2020-04-09     2           14          14.3
## 2 2020-04-10     1           14           7.1
## 3 2020-04-11     1           14           7.1
## 4 2020-04-12     0           14            0
## 5 2020-04-13     2           14          14.3
## 6 2020-04-14     0           14            0
## 7 2020-04-15     0           14            0
## 8 2020-04-16     3           14          21.4
## 9 2020-04-17     1           14           7.1
## 10 2020-04-18    2           14          14.3
## # i 292 more rows

```

```

covid19 %>%
  group_by(year(date), month(date), week(date)) %>%
  mutate(sum.sum.week = sum(`0-9세`)) %>%
  ungroup() %>%
  mutate(rate.sum.week = round(`0-9세`/sum.sum.week, 3) * 100) %>%
  select(date, `0-9세`, sum.sum.week, rate.sum.week)

```

```

## # A tibble: 302 x 4
##   date      `0-9세` sum.sum.week rate.sum.week
##   <date>     <dbl>        <dbl>          <dbl>
## 1 2020-04-09     2           6          33.3
## 2 2020-04-10     1           6          16.7
## 3 2020-04-11     1           6          16.7
## 4 2020-04-12     0           6            0
## 5 2020-04-13     2           6          33.3
## 6 2020-04-14     0           6            0
## 7 2020-04-15     0           6            0
## 8 2020-04-16     3           6          50
## 9 2020-04-17     1           6          16.7
## 10 2020-04-18    2           6          33.3
## # i 292 more rows

```

```

# 취업자수의 분기별 비율
employees.tsibble %>%
  index_by(yearqtr = ~ yearquarter(..)) %>%
  mutate(sum.qtrly = sum(total)) %>%
  mutate(rate.qtrly = total/sum.qtrly) %>%

```

```

head(15)

## # A tsibble: 15 x 6 [1D]
## # Groups:   @ yearqtr [5]
##   time      total employees.edu yearqtr sum.qtrly rate.qtrly
##   <date>     <int>        <int>   <qtr>    <int>     <dbl>
## 1 2013-01-01 24287        1710 2013 Q1    73238    0.332
## 2 2013-02-01 24215        1681 2013 Q1    73238    0.331
## 3 2013-03-01 24736        1716 2013 Q1    73238    0.338
## 4 2013-04-01 25322        1745 2013 Q2    76618    0.330
## 5 2013-05-01 25610        1774 2013 Q2    76618    0.334
## 6 2013-06-01 25686        1786 2013 Q2    76618    0.335
## 7 2013-07-01 25681        1813 2013 Q3    76895    0.334
## 8 2013-08-01 25513        1811 2013 Q3    76895    0.332
## 9 2013-09-01 25701        1794 2013 Q3    76895    0.334
## 10 2013-10-01 25798       1790 2013 Q4    76841    0.336
## 11 2013-11-01 25795       1793 2013 Q4    76841    0.336
## 12 2013-12-01 25248       1779 2013 Q4    76841    0.329
## 13 2014-01-01 25050       1748 2014 Q1    75629    0.331
## 14 2014-02-01 25116       1786 2014 Q1    75629    0.332
## 15 2014-03-01 25463       1814 2014 Q1    75629    0.337

employees.tsibble%>%
  index_by(yearqtr = ~ year(.)) %>%
  mutate(sum.qtrly = sum(total)) %>%
  mutate(rate.qtrly = (total/sum.qtrly)*100) %>%
  head(15)

## # A tsibble: 15 x 6 [1D]
## # Groups:   @ yearqtr [2]
##   time      total employees.edu yearqtr sum.qtrly rate.qtrly
##   <date>     <int>        <int>   <dbl>    <int>     <dbl>
## 1 2013-01-01 24287        1710 2013 303592    8.00
## 2 2013-02-01 24215        1681 2013 303592    7.98
## 3 2013-03-01 24736        1716 2013 303592    8.15
## 4 2013-04-01 25322        1745 2013 303592    8.34
## 5 2013-05-01 25610        1774 2013 303592    8.44
## 6 2013-06-01 25686        1786 2013 303592    8.46
## 7 2013-07-01 25681        1813 2013 303592    8.46
## 8 2013-08-01 25513        1811 2013 303592    8.40
## 9 2013-09-01 25701        1794 2013 303592    8.47
## 10 2013-10-01 25798       1790 2013 303592    8.50
## 11 2013-11-01 25795       1793 2013 303592    8.50
## 12 2013-12-01 25248       1779 2013 303592    8.32
## 13 2014-01-01 25050       1748 2014 310766    8.06
## 14 2014-02-01 25116       1786 2014 310766    8.08
## 15 2014-03-01 25463       1814 2014 310766    8.19

```

월별, 분기별, 연별 누적 합계

```

employees %>%
  mutate(cumsum = cumsum(total)) %>%
  select(time, total, cumsum) %>%
  head(15)

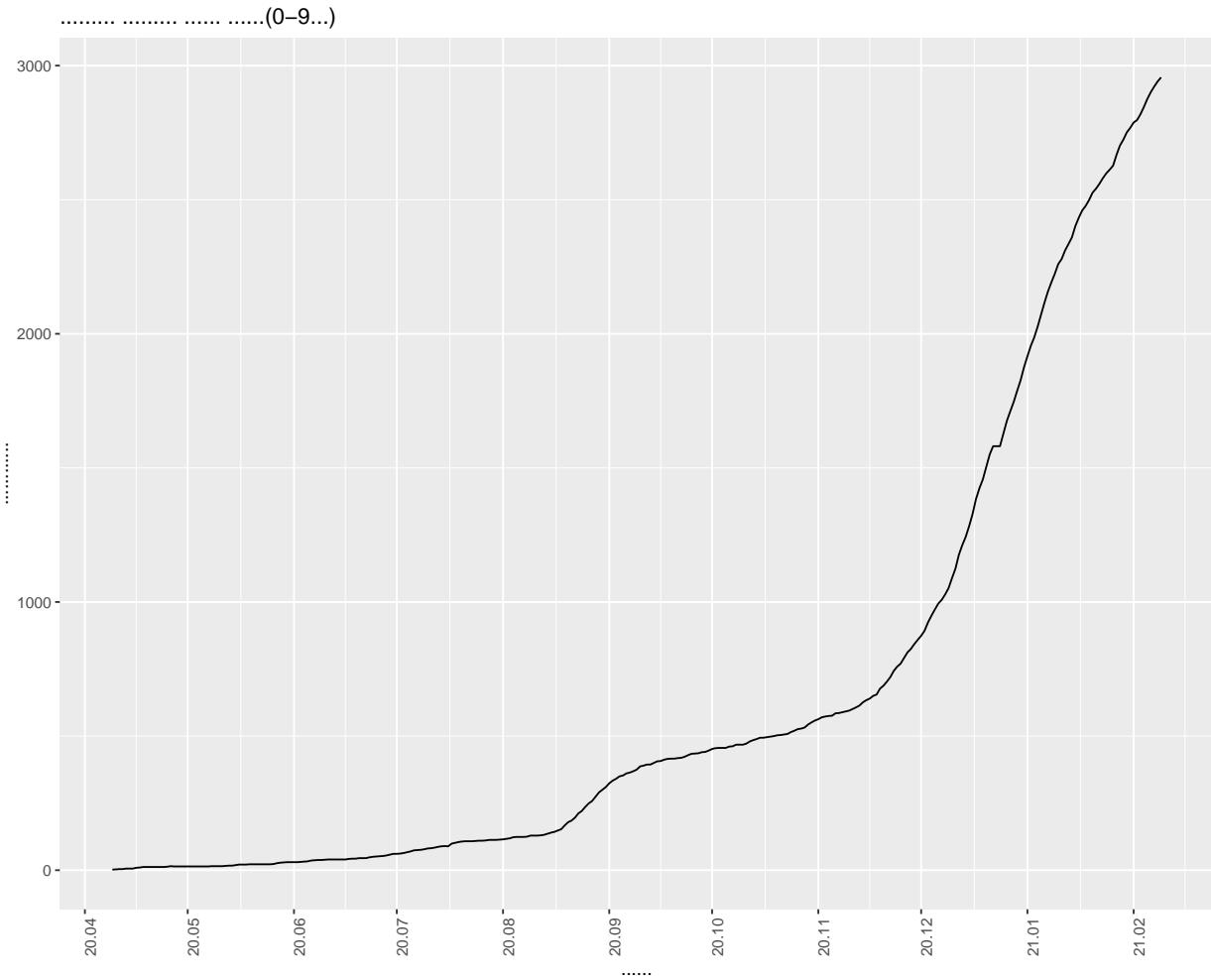
```

```

##           time total cumsum
## 1 2013-01-01 24287 24287
## 2 2013-02-01 24215 48502
## 3 2013-03-01 24736 73238
## 4 2013-04-01 25322 98560
## 5 2013-05-01 25610 124170
## 6 2013-06-01 25686 149856
## 7 2013-07-01 25681 175537
## 8 2013-08-01 25513 201050
## 9 2013-09-01 25701 226751
## 10 2013-10-01 25798 252549
## 11 2013-11-01 25795 278344
## 12 2013-12-01 25248 303592
## 13 2014-01-01 25050 328642
## 14 2014-02-01 25116 353758
## 15 2014-03-01 25463 379221

# 0-9세 코로나 확진자의 누적 플롯
covid19 %>%
  mutate(cumsum = cumsum(`0-9세`)) %>%
  select(date, `0-9세`, cumsum) %>%
  ggplot(aes(date, cumsum)) +
  geom_line(aes(group = 1)) +
  labs(title = '코로나 확진자 누적 합계(0-9세)', x = '날짜', y = '누적합계') +
  scale_x_date(date_breaks = "1 month", date_labels = "%y.%m") +
  theme(axis.text.x=element_text(angle=90,hjust=1))

```



```
employees %>%
  group_by(year(time)) %>%
  mutate(cumsum.total = cumsum(total),
        cumsum.edu = cumsum(employees.edu)) %>%
  select(time, total, cumsum.total, employees.edu, cumsum.edu) %>%
  head(15)
```

```
## # A tibble: 15 x 6
## # Groups:   year(time) [2]
##   `year(time)` time     total cumsum.total employees.edu cumsum.edu
##   <dbl> <date>    <int>      <int>       <int>       <int>
## 1 2013 2013-01-01  24287     24287      1710      1710
## 2 2013 2013-02-01  24215     48502      1681      3391
## 3 2013 2013-03-01  24736     73238      1716      5107
## 4 2013 2013-04-01  25322     98560      1745      6852
## 5 2013 2013-05-01  25610    124170      1774      8626
## 6 2013 2013-06-01  25686    149856      1786     10412
## 7 2013 2013-07-01  25681    175537      1813     12225
## 8 2013 2013-08-01  25513    201050      1811     14036
## 9 2013 2013-09-01  25701    226751      1794     15830
## 10 2013 2013-10-01  25798    252549      1790     17620
## 11 2013 2013-11-01  25795    278344      1793     19413
```

```

## 12      2013 2013-12-01 25248      303592      1779      21192
## 13      2014 2014-01-01 25050      25050       1748      1748
## 14      2014 2014-02-01 25116      50166       1786      3534
## 15      2014 2014-03-01 25463      75629       1814      5348

employees.tsibble%>%
  index_by(yearqtr = ~ yearquarter(.)) %>%
  mutate(cumsum.qtrly = cumsum(total)) %>%
  select(yearqtr, cumsum.qtrly) %>%
  head(10)

## # A tsibble: 10 x 3 [1D]
## # Groups:   @ yearqtr [4]
##   yearqtr cumsum.qtrly time
##   <qtr>     <int> <date>
## 1 1 2013 Q1        24287 2013-01-01
## 2 2 2013 Q1        48502 2013-02-01
## 3 3 2013 Q1        73238 2013-03-01
## 4 4 2013 Q2        25322 2013-04-01
## 5 5 2013 Q2        50932 2013-05-01
## 6 6 2013 Q2        76618 2013-06-01
## 7 7 2013 Q3        25681 2013-07-01
## 8 8 2013 Q3        51194 2013-08-01
## 9 9 2013 Q3        76895 2013-09-01
## 10 10 2013 Q4        25798 2013-10-01

covid19.tsibble[, c(1,3)]%>%
  index_by(yearweek = ~ yearweek(.)) %>%
  mutate(cumsum.weekly = cumsum(`0-9세`)) %>%
  head(10)

## # A tsibble: 10 x 4 [1D]
## # Groups:   @ yearweek [2]
##   date      `0-9세` yearweek cumsum.weekly
##   <date>    <dbl>    <week>     <dbl>
## 1 2020-04-09      2 2020 W15         2
## 2 2020-04-10      1 2020 W15         3
## 3 2020-04-11      1 2020 W15         4
## 4 2020-04-12      0 2020 W15         4
## 5 2020-04-13      2 2020 W16         2
## 6 2020-04-14      0 2020 W16         2
## 7 2020-04-15      0 2020 W16         2
## 8 2020-04-16      3 2020 W16         5
## 9 2020-04-17      1 2020 W16         6
## 10 2020-04-18     2 2020 W16         8

do.call(rbind, lapply(split(employees.xts, f = 'year'), cumsum)) %>%
  head(15)

##      total employees.edu 증감 증감률
## 2013-01-01 24287      1710    NA    NA
## 2013-02-01 48502      3391    NA    NA
## 2013-03-01 73238      5107    NA    NA
## 2013-04-01 98560      6852    NA    NA
## 2013-05-01 124170     8626    NA    NA
## 2013-06-01 149856     10412   NA    NA

```

```

## 2013-07-01 175537      12225    NA    NA
## 2013-08-01 201050      14036    NA    NA
## 2013-09-01 226751      15830    NA    NA
## 2013-10-01 252549      17620    NA    NA
## 2013-11-01 278344      19413    NA    NA
## 2013-12-01 303592      21192    NA    NA
## 2014-01-01 25050       1748    -198   -0.8
## 2014-02-01 50166       3534    -132   -0.5
## 2014-03-01 75629       5348    215    0.9

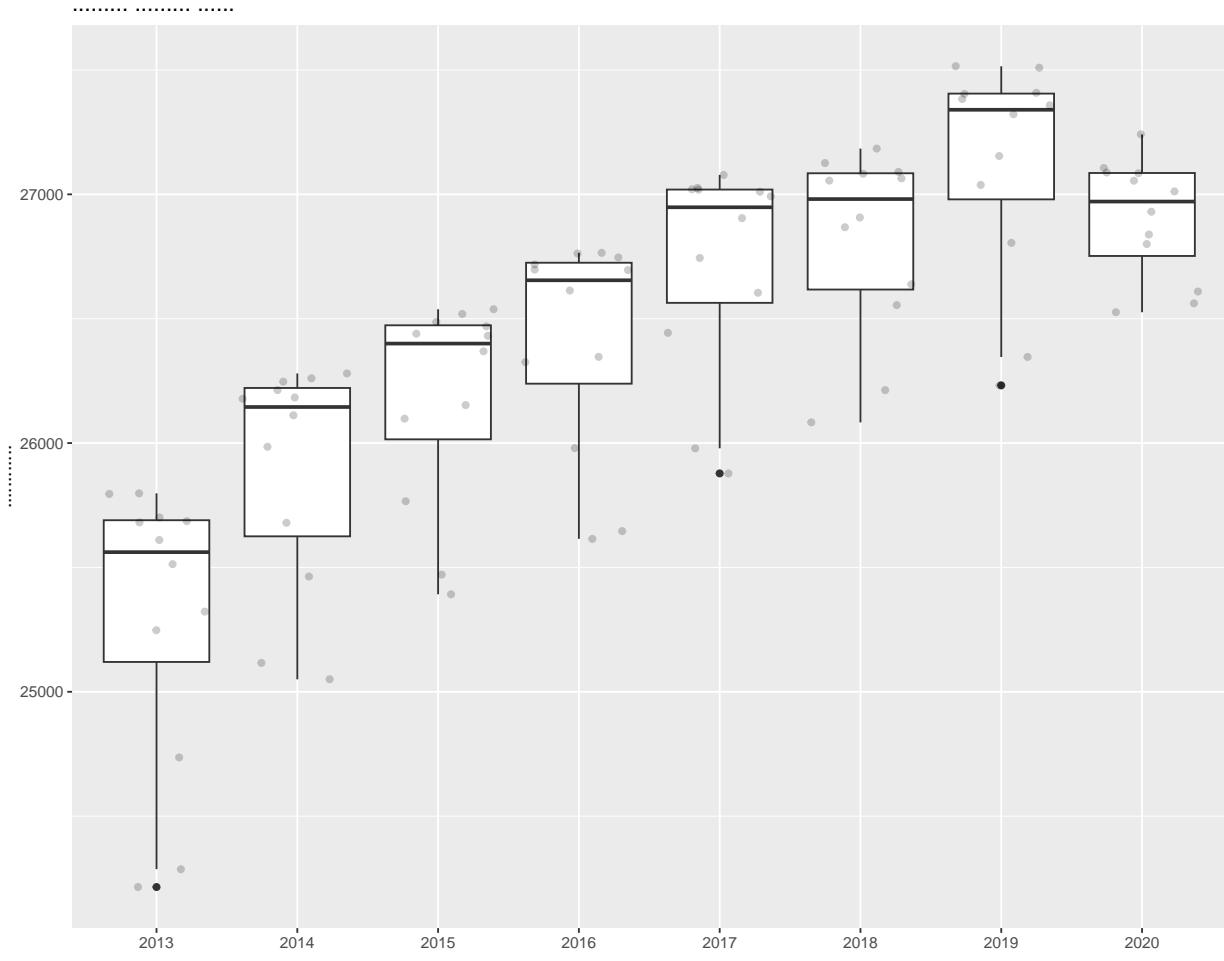
```

동월별, 동분기별, 동년별 플롯

```

employees %>%
  mutate(year = lubridate::year(employees$time)) %>%
  ggplot(aes(as.factor(year), total)) +
  geom_boxplot() +
  geom_jitter(alpha = 0.2) +
  labs(title = '동년별 취업자 분포', x = '연도', y = '취업자수')

```



```

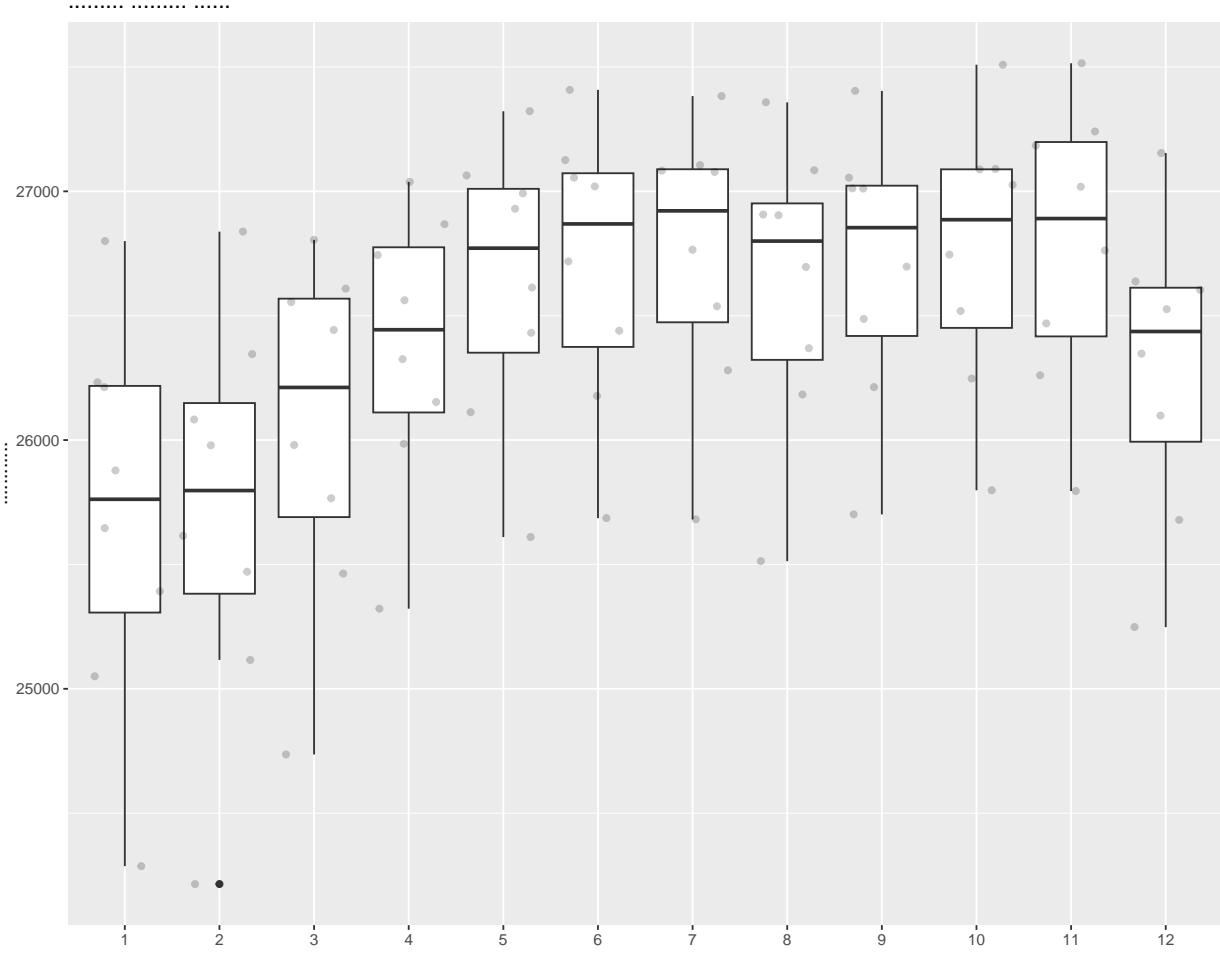
employees %>%
  mutate(month = lubridate::month(employees$time)) %>%

```

```

ggplot(aes(as.factor(month), total)) +
  geom_boxplot() +
  geom_jitter(alpha = 0.2) +
  labs(title = '동월별 취업자 분포', x = '월', y = '취업자수')

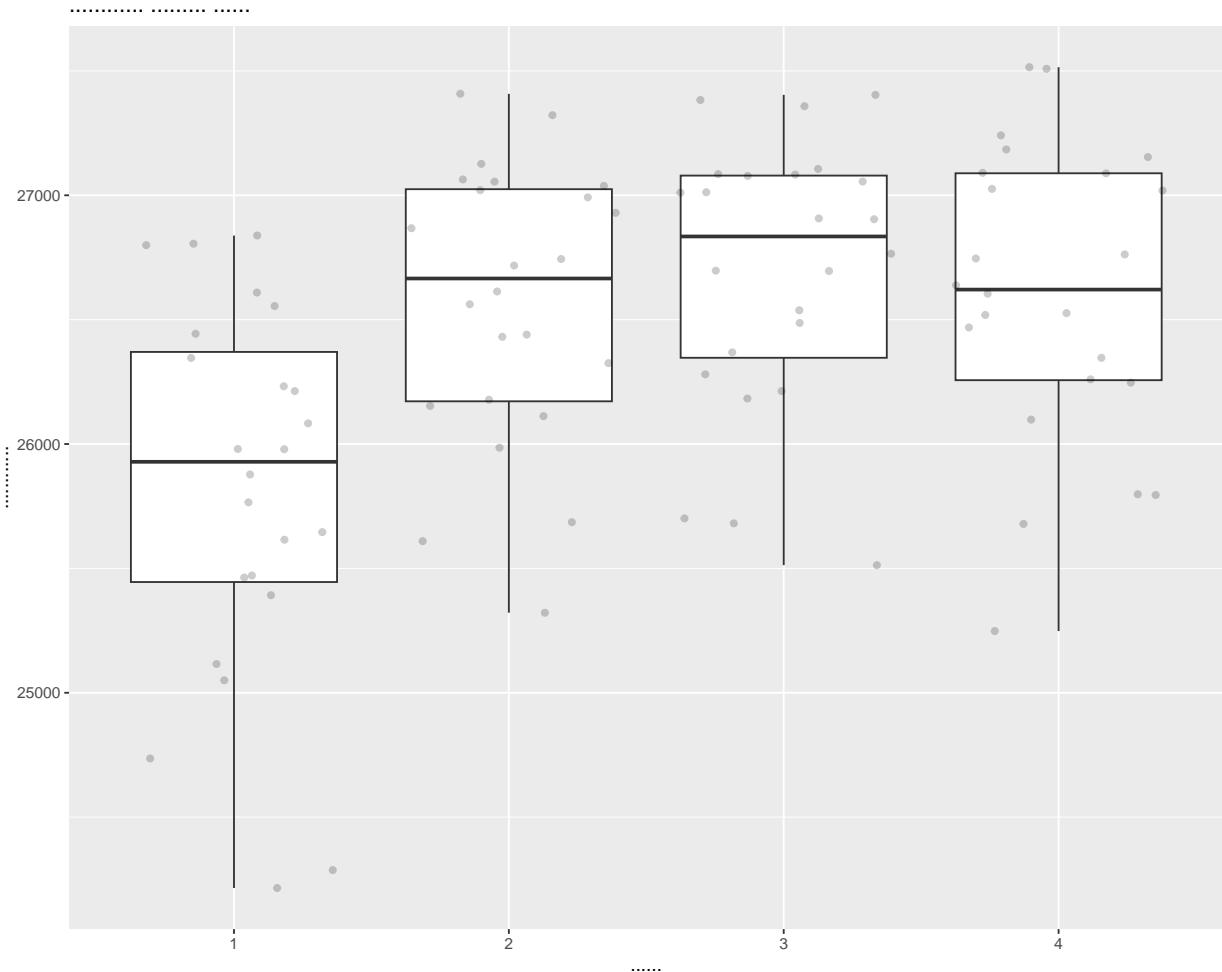
```



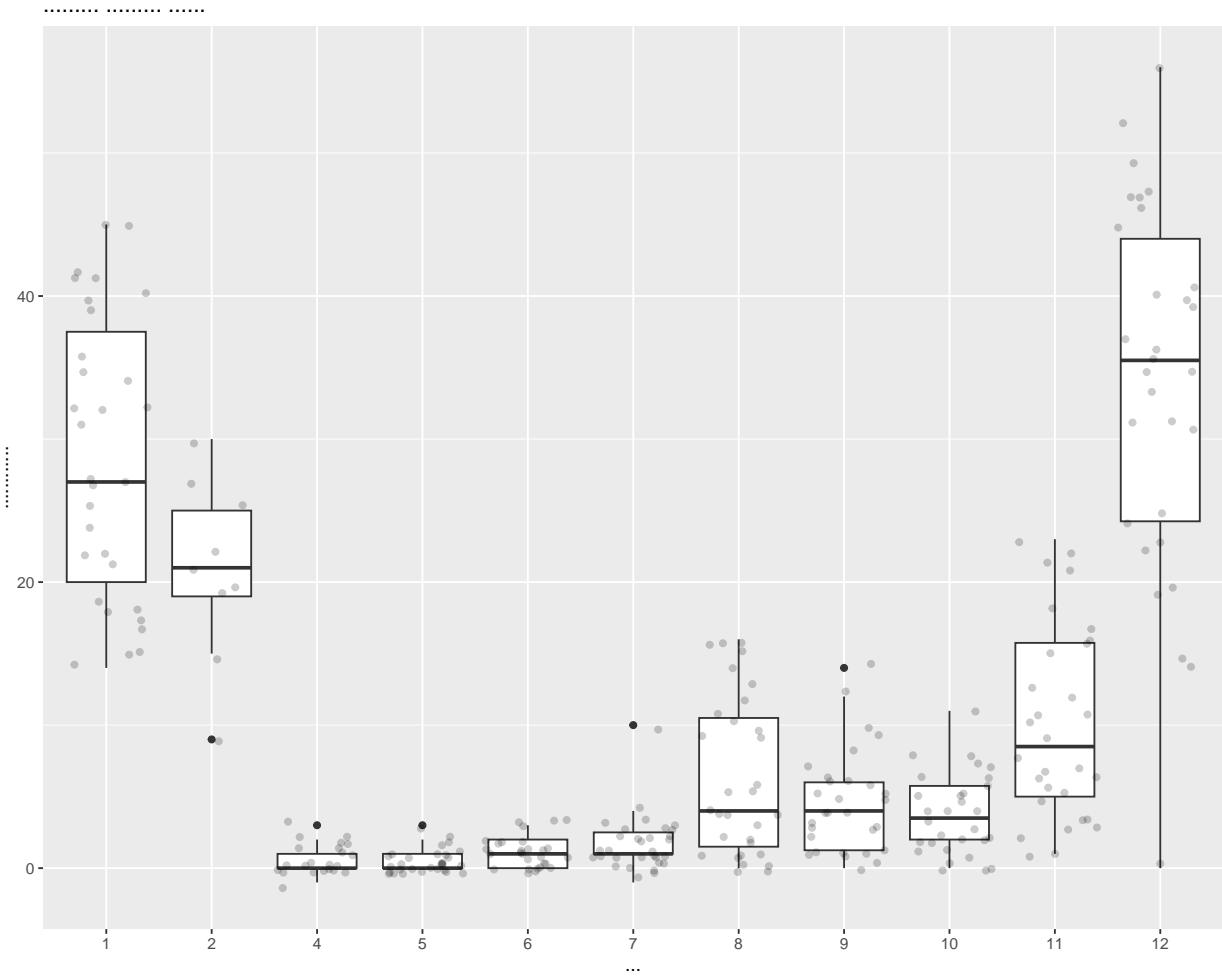
```

employees %>%
  mutate(quarter = lubridate::quarter(employees$time)) %>%
  ggplot(aes(as.factor(quarter), total)) +
  geom_boxplot() +
  geom_jitter(alpha = 0.2) +
  labs(title = '동분기별 취업자 분포', x = '분기', y = '취업자수')

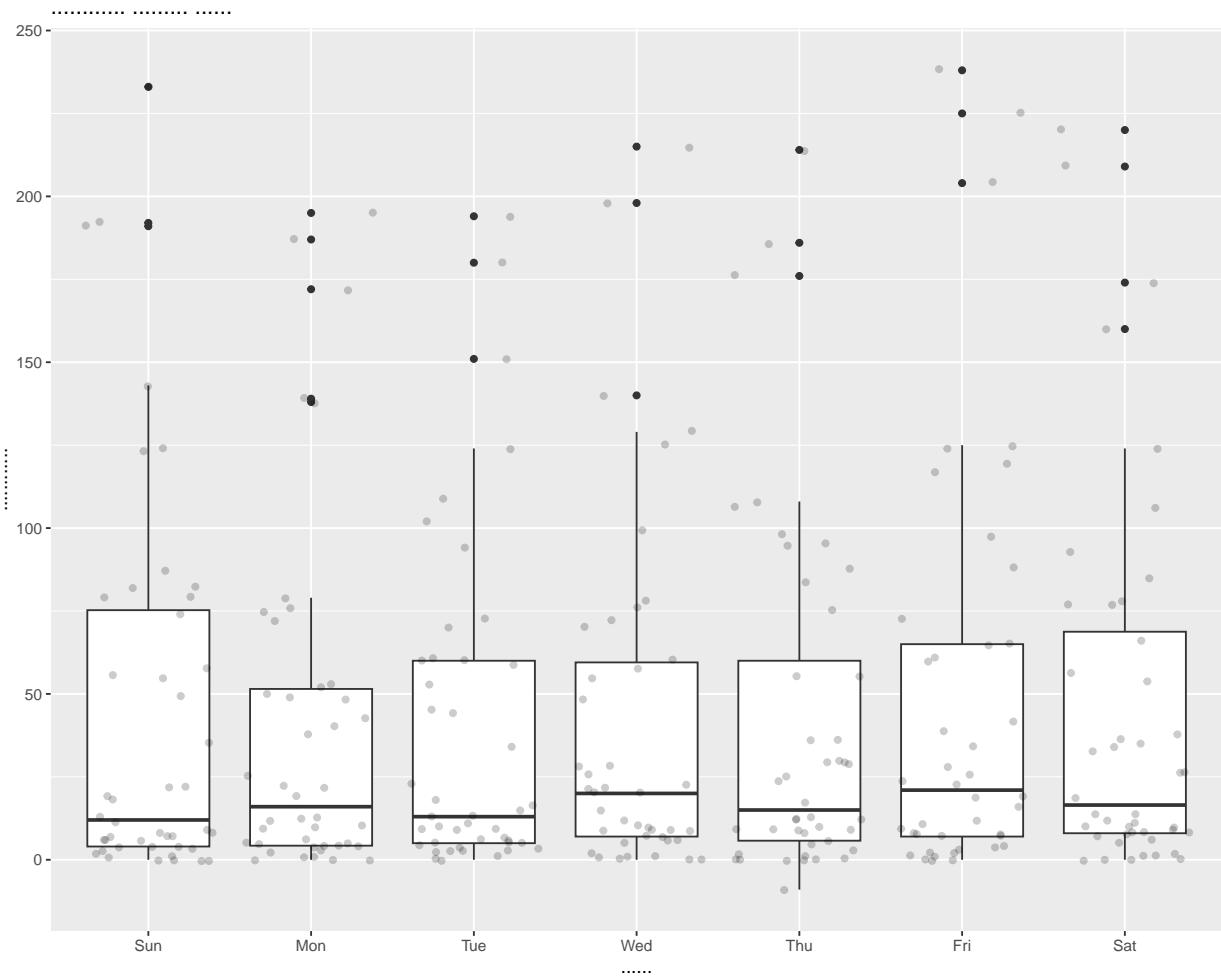
```



```
covid19 %>%
  mutate(month = lubridate::month(covid19$date)) %>%
  ggplot(aes(as.factor(month), `0-9세`)) +
  geom_boxplot() +
  geom_jitter(alpha = 0.2) +
  labs(title = '동월별 확진자 분포', x = '월', y = '확진자수')
```



```
covid19 %>%
  mutate(wday = lubridate::wday(covid19$date, label = TRUE)) %>%
  ggplot(aes(as.factor(wday), `50-59세`)) +
  geom_boxplot() +
  geom_jitter(alpha = 0.2) +
  labs(title = '동요일별 확진자 분포', x = '요일', y = '확진자수')
```



```
employees %>%
  timetk::plot_seasonal_diagnostics(.date_var = time, .value = total, .title = '전체
취업자의 주기별 플롯')
```

```
covid19 %>%
  timetk::plot_seasonal_diagnostics(.date_var = date, .value = `0-9세`, .title = '코로나
확진자(0-9세)의 주기별 플롯')
```