1. 추세(trend)의 시각화

추세(Trend)란 데이터의 흐름에 따라 발생하는 데이터의 변화를 말한다. 보통 추세는 시간적 흐름에 따라 발생한 데이터를 말하는 경우가 많지만 추세가 꼭 시간의 흐름에 종속되지는 않는다. 예를 들자면 회차(물론 이 또한 시간의 흐름과 무관하지 않지만)나 이벤트의 발생과 같은 흐름도 추세에 속할 수 있다. 하지만 추세에 측정에 있어 하나 중요한 것은 그것이 시간이든 회차이든 특정 이벤트이던 그들의 흐름을 측정하는 간격이나 성질이 일정해야 한다는 것이다. 시간의 경우 추세를 측정하기위해서는 시간적 간격, 즉, 연도별, 월별, 일별 등의 간격이 동일해야 하고 회차의 경우 1회, 2회와 같이 연속된 회차로 기록되어야 유의미하다. 만약 시간의 간격이 어느 구간에서는 연도별, 어느 구간에서는 월별로 표현된다면 추세를 정확히 파악하기 어렵다. 따라서 추세에는 데이터의 흐름, 특히 흐름의 측정 간격이 매우 중요하다.

추세를 시각화할 때는 데이터의 포인트와 해당 데이터의 바로 전 데이터와 다음 데이터를 연결하는 선 그래프가 많이 사용되지만 막대그래프도 많이 사용된다.

# 시간의 흐름에 따른 변화는? - 선 그래프

선 그래프 (또는 꺾은 선형 차트)는 특정한 변량의 흐름에 따라 변화되는 데이터 값들을 선으로 연결하여 그 변화량을 보여주는 시각화 방법이다. 이 선그래프가 가장 효과적으로 사용되는 시각화가 시간의 흐름에 따라 변화하는 시계열 데이터에 대한 시각화 방법이다. 각각의 시간에 관측된 데이터 포인트들을 같은 변수이나 변량끼리 선으로 연결하였기 때문에 그 기본은 산점도에 있다고 할수도 있다.

ggplot2에서는 선 그래프를 위해 geom\_line()을 제공한다. 이 geom\_line()은 시계열 뿐만아닌 다양한 선 그래프를 그리는데 사용된다. 하지만 ggplot2에서 시계열을 위해 특별하게 제공하는 함수가 있는데 scale\_\*\_date()와 scale\_\*\_datetime(), scale\_\*\_time()이다. 이 세 함수는 축에 매핑되는 변수가 날짜형 포맷[[1]](#footnote-20)일때 사용이 가능한데 여러가지 시간에 관련된 연산이 가능하기 때문에 날짜형 포맷의 데이터를 이 스케일로 매핑하면 시간 데이터를 다루기가 쉬워진다. 데이터를 날짜형 포맷으로 만들기 위해서는 as.Date()를 사용할 수 있다.

as.Date(x, format, tryFormats = c(“%Y-%m-%d”, “%Y/%m/%d”), …)  
- x : 시간형 데이터 포맷으로 변경할 문자형 또는 수치형 벡터  
- format : x를 읽어들여 시간형으로 변환하기 위한 포맷(%Y : 4자리 연도, %y : 두자리 연도, %m : 두자리 월, %d : 두자리 일 등)  
- tryformat : format이 설정되지 않은 경우 적용할 포맷

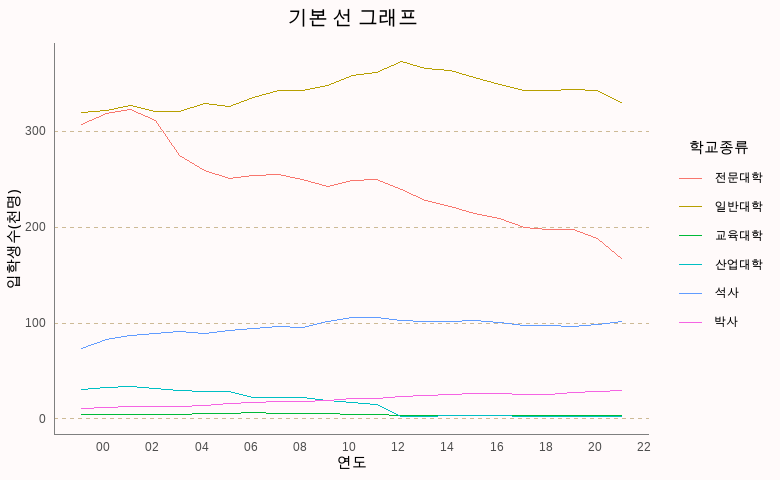
날짜형 포맷으로 설정된 변수를 X축과 Y축에 매핑시키면 scale\_\*\_date()와 scale\_\*\_datetime()을 사용하여 스케일을 날자형이나 시간형 스케일로 설정할 수 있다.

scale\_\*\_date(name = waiver(), breaks = waiver(), date\_breaks = waiver(), labels = waiver(), date\_labels = waiver(), minor\_breaks = waiver(), date\_minor\_breaks = waiver(), limits = NULL, expand = waiver(), oob = censor, guide = waiver(), position = “bottom”, sec.axis = waiver())

scale\_\*\_datetime(name = waiver(), breaks = waiver(), date\_breaks = waiver(), labels = waiver(), date\_labels = waiver(), minor\_breaks = waiver(), date\_minor\_breaks = waiver(), timezone = NULL, limits = NULL, expand = waiver(), oob = censor, guide = waiver(), position = “bottom”, sec.axis = waiver())  
- name : 축 제목 설정  
- braek : 축의 눈금 설정  
- date\_breaks : 축의 눈금을 만들 시간 간격 설정(ex. ‘2 years’, ‘3 months’ 등)  
- label : 축 눈금의 라벨 설정  
- date\_labels : 시간 간격에 따라 만들어진 눈금에 표기할 라벨  
- date\_minor\_breaks : 보조 눈금자를 만들 시간 간격 설정  
- oob : 한계를 넘어가는 데이터 값을 처리할 함수 설정  
- timezone : 국가별 시간대의 지정  
- sec.axis : 두번째 축에 대한 설정

다음은 geom\_line()으로 그린 시계열 선 그래프이다. 앞서 설명한 scale\_x\_date()를 사용하여 X축을 날짜형 스케일로 설정하였다.

## df\_입학자\_long의 연도열 데이터를 날짜형 데이터로 변환  
df\_입학자\_long$연도 <- as.Date(df\_입학자\_long$연도, '%Y')  
  
## 학교종류의 순서 설정을 위해 레벨 설정  
df\_입학자\_long$학교종류 <- fct\_relevel(df\_입학자\_long$학교종류, '전문대학', '일반대학', '교육대학', '산업대학', '방송통신대학', '원격및사이버대학', '석사', '박사')  
  
## df\_입학자\_long 데이터 중 일부를 필터링하여 ggplot 객체 생성  
p\_line <- df\_입학자\_long |> filter(지역 == '전체', 학교종류 %in% c('전문대학', '일반대학', '교육대학', '산업대학', '석사', '박사')) |>  
 ggplot() +   
 ## X축을 date 스케일로 설정하는데 2년마다 눈금자를 그리고 라벨은 2자리 연도로 표기  
 scale\_x\_date(date\_breaks = "2 years", date\_labels = '%y') +  
 labs(x = '연도', y = '입학생수(천명)')  
  
## X축을 연도, 입학생수를 천명단위로 넣기 위해 Y축을 입학생수/1000, group과 color을 학교종류로 매핑한 geom\_line() 레이어 생성  
p\_line +  
 geom\_line(aes(x = 연도, y = (입학생수)/1000, group = 학교종류, color = 학교종류)) +   
 labs(title = '기본 선 그래프')



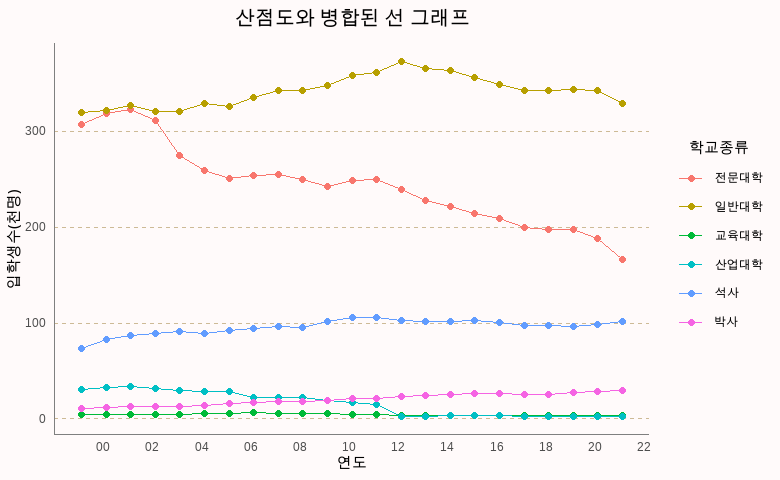
실행결과 7-기본 선 그래프

# 선만 있으면 심심해? 점도 찍고 숫자도 찍어 – 선 그래프의 변형

앞의 선 그래프는 데이터의 전체적인 흐름은 파악하기 쉬우나 데이터의 정확한 값을 알아낼 수 없다. 이런 경우 산점도(geom\_point())와 텍스트(geom\_text())를 사용하여 데이터의 정확한 위치와 값을 표현할 수 있다.

아래의 코드는 선 그래프에 산점도 레이어를 추가한 코드이다.

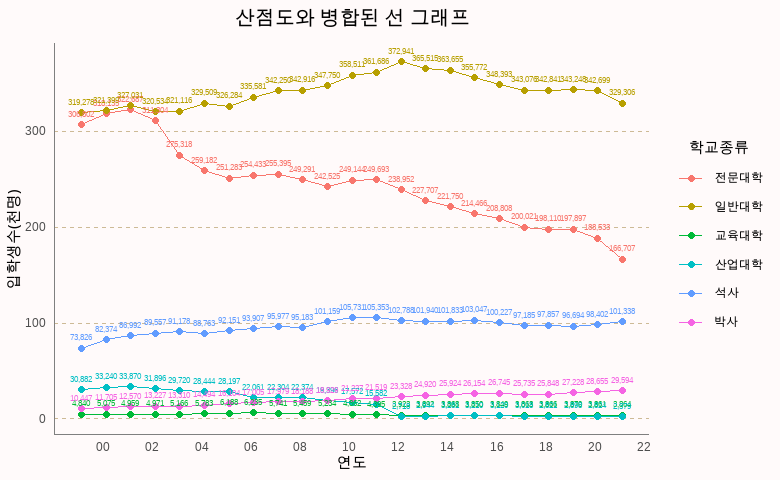
p\_line\_scatter <- p\_line +  
 ## X축을 연도, Y축을 입학생수/1000, group과 color를 학교종류로 매핑한 geom\_line 레이어 생성  
 geom\_line(aes(x = 연도, y = (입학생수)/1000, group = 학교종류, color = 학교종류)) +  
 ## X축을 연도, Y축을 입학생수/1000, color를 학교종류로 매핑한 geom\_point 레이어 추가  
 geom\_point(aes(x = 연도, y = (입학생수)/1000, color = 학교종류)) +   
 labs(title = '산점도와 병합된 선 그래프')  
  
p\_line\_scatter



실행결과 7-산점도와 병합된 선 그래프

여기에 정확한 데이터의 값을 보여주는 레이어를 다음과 같이 추가하겠다.

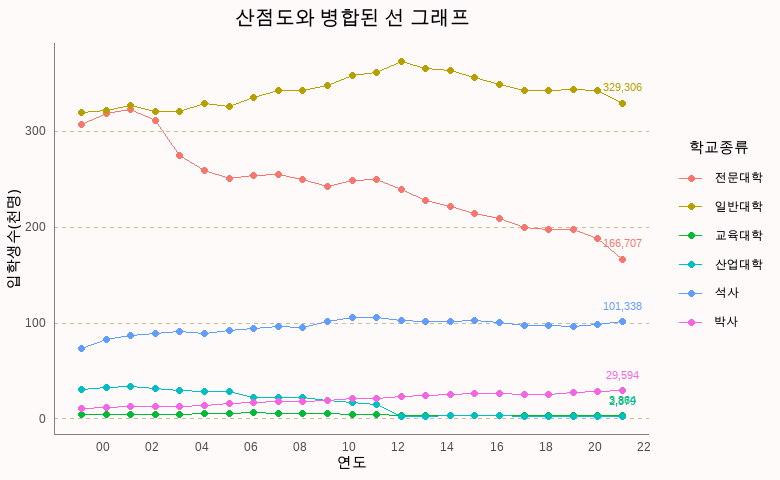
p\_line\_scatter +  
 ## X축을 연도, Y축을 입학생수/1000, color를 학교종류, label을 유효숫자가 1인 입학생수를 콤마형으로 매핑한 geom\_text 레이어 추가  
 geom\_text(aes(x = 연도, y = (입학생수)/1000, color = 학교종류, label = scales::comma(입학생수, accuracy = 1)), size = 2, vjust = -1.5)



실행결과 7-데이터 값이 표시된 선 그래프

하지만 수치가 너무 많아 오히려 그래프가 더 보기 어려워졌다. 데이터 값을 중요한 곳에만 표기해 주면 보기가 나아질 것이다. 다음의 코드는 마지막 연도의 데이터값만 표기하는 코드이다. 코드중에 lubridate 패키지에서 제공하는 year()가 사용되었다. lubridate패키지는 시계열 데이터의 가장 기본인 시간 데이터를 다루는 다양한 함수를 제공하는 패키지이다. lubridate의 year()는 날짜형 데이터 포맷에서 연도를 추출하는 함수이다.

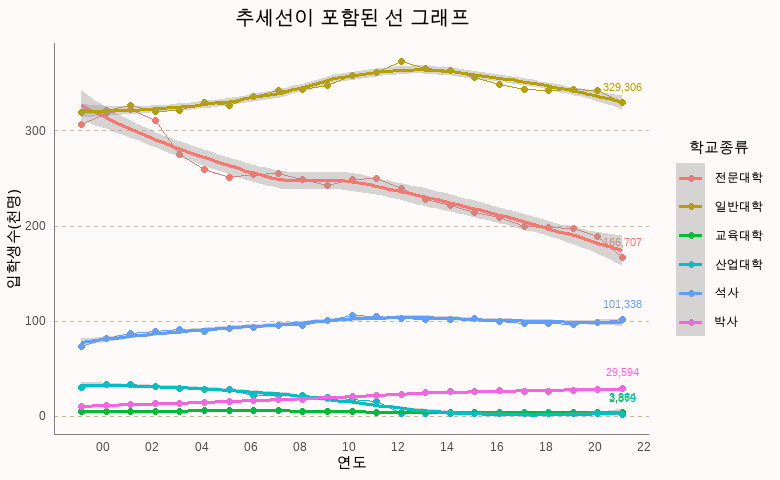
p\_line\_scatter1 <- p\_line\_scatter +  
 ## df\_입학자\_long에서 2021년만 필터링한 데이터를 사용하고   
 geom\_text(data = df\_입학자\_long |> filter(지역 == '전체', 학교종류 %in% c('전문대학', '일반대학', '교육대학', '산업대학', '석사', '박사'), lubridate::year(연도) == '2021'),  
 ## X축을 연도, Y축을 입학생수/1000, color를 학교종류, label을 유효숫자가 1인 입학생수를 콤마형으로 매핑한 geom\_text 레이어 추가  
 aes(x = 연도, y = (입학생수)/1000, color = 학교종류, label = scales::comma(입학생수, accuracy = 1)),   
 size = 3, vjust = -1.5, show.legend = FALSE)  
  
p\_line\_scatter1



# 20년간 추세를 알려줘 – 추세 선이 추가된 선 그래프

앞에서 그린 선 그래프에 데이터가 전반적으로 증가하는 추세인지, 감소하는 추세인지를 확인하기 위해서는 추세선을 추가하면 그 추세를 쉽게 확인할 수 있다. 추세선은 다음과 같이 추가할 수 있다.

p\_line\_scatter1 +  
 ## 추세선을 추가하기 위해 geom\_smooth 레이어 추가  
 geom\_smooth(aes(x = 연도, y = (입학생수)/1000, group = 학교종류, color = 학교종류)) +   
 labs(title = '추세선이 포함된 선 그래프')



실행결과 7-추세선이 포함된 선 그래프

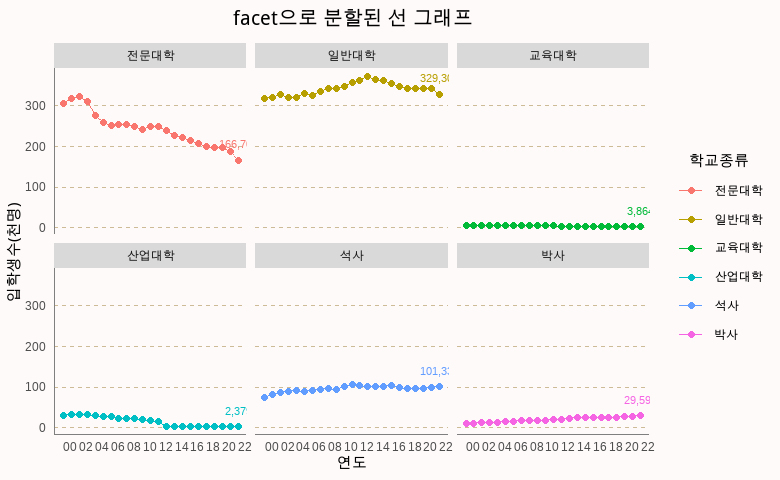
# 선 그래프가 너무 꼬였어. 어떻게 해? - 스파게티 선 그래프

선 그래프는 하나의 선 그래프에 하나의 변수만이 표현되는 일변량과 여러 변수를 표현하는 다변량에 모두 사용할 수 있다. 다변량 선 그래프는 같은 X, Y좌표에서 변수의 분류에 따라 여러 개의 선을 동시에 표현하는 선 그래프를 말한다. 하지만 한 그래프에서 너무 많은 선이 표현되면 선들이 겹쳐져서 데이터의 흐름을 알아내기가 어렵다. 이와 같은 선 그래프를 스파게티(spaghetti) 선 그래프라고 한다. 하지만 여러 범주를 가진 변수에 대한 선 그래프를 그리기 위해서는 어쩔수 없이 스파게티 선 그래프를 그려야 할 때가 있다. 이런 경우 어떻게 효과적으로 시각화할 지에 대해 알아보자.

### 분할(facet) 사용

하나의 그래프에 여러개의 범주를 가지는 변수를 표현할 때 많이 사용하는 방법이 각각의 범주별로 분할하여 그래프를 표현하는 방법이다. ggplot2에서는 facet\_wrap()과 facet\_grid()를 사용하여 그래프를 변수의 범주별로 분할한다. 다음은 앞선 선 그래프를 학교종류별로 분할한 코드이다.

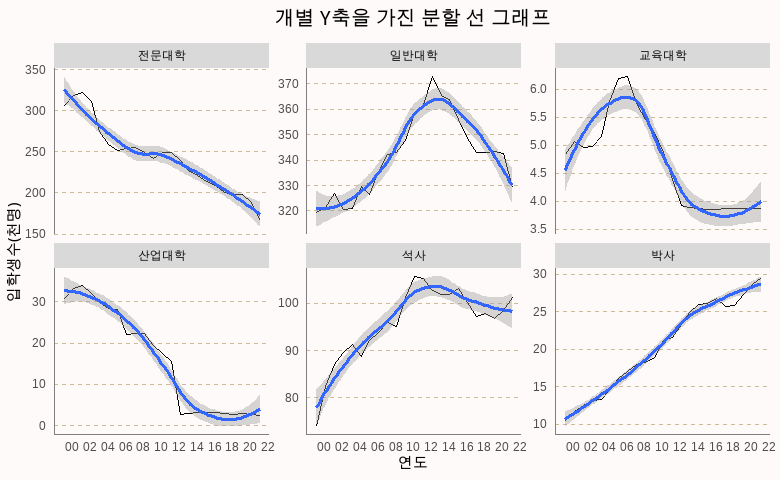
p\_line\_scatter1 +  
 ## facet\_wrap()으로 학교종류별로 그래프를 분할  
 facet\_wrap(~ 학교종류) +  
 labs(title = 'facet으로 분할된 선 그래프')



실행결과 7-facet으로 분할된 선 그래프

앞서 분할한 선 그래프를 보면 Y축이 고정되어 있기 때문에 각각의 분할된 그래프들의 시간적 데이터의 차이가 명확히 보이지 않는다. 이런 경우 facet\_\*()의 scales 매개변수를 ’free\_y’로 설정하면 각 분할 그래프에 적합한 Y축의 범위가 설정된다. 하지만 이는 매우 주의깊게 설정해야 한다. Y축의 범위가 넓은 분할 그래프의 1단위 변화와 Y축의 범위가 좁은 분할 그래프의 1단위의 변화는 그 값의 크기가 다르기 때문이다. 다음의 예를 보자.

p\_line +  
 geom\_line(aes(x = 연도, y = (입학생수)/1000, group = 학교종류)) +  
 ## facet\_wrap()에 scales를 'free\_y'로 설정  
 facet\_wrap(~ 학교종류, scales = 'free\_y') +  
 ## geom\_smooth()를 사용하여 추세선 레이어 추가  
 geom\_smooth(aes(x = 연도, y = (입학생수)/1000, group = 학교종류)) +   
 labs(title = '개별 Y축을 가진 분할 선 그래프')



실행결과 7-개별 Y축을 가진 분할 선 그래프

앞의 분할된 그래프 중 전문대학과 박사는 그 형태가 완전히 반비례하는 것을 볼 수 있다. 하지만 전문대학은 20년간 약 150~170만명 정도가 감소한 반면 박사는 2만명정도 증가하였다. 기울기가 유사해 보이기 떄문에 증가폭도 비슷할 것으로 보이지만 두 데이터의 증감량 차이는 매우 크다.

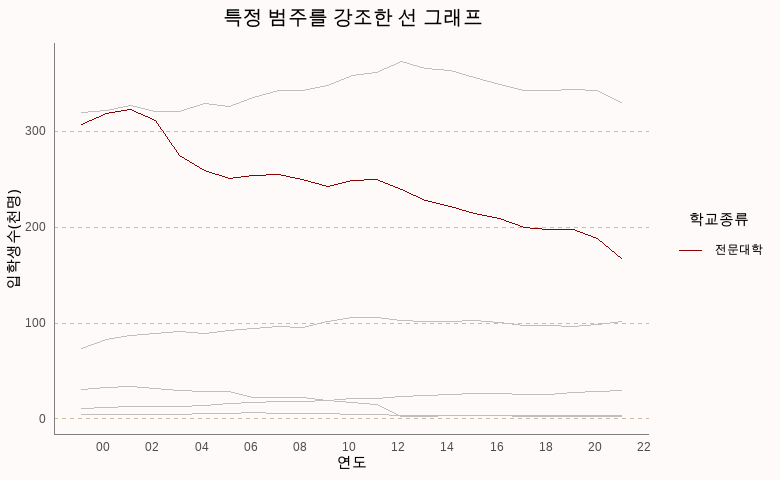
### 특정 범주만 강조

스파게티 선 그래프를 해결하는 방법으로 특정 범주만을 강조하는 방법이 있다. 이 방법은 앞서 박스플롯을 설명할 때 사용했던 방법과 유사하다.

보통 스파게티 선 그래프를 그리는 이유는 동일 변수의 여러 범주에 대비하여 특정 범주의 흐름을 파악하는 경우가 많다. 이 경우 강조하고 싶은 범주를 강조하고 나머지 비교 대상이 되는 범주를 흐릿하게 표현하면 효과적으로 스파게티 선 그래프를 활용할 수 있다. 여기에서는 전문대학 입학생을 강조해보도록 한다.

우선 흐릿한 회색(grey80) 선으로 전체 스파게티 선 그래프를 그린다. 이후 강조하고자 하는 데이터만 필터링한 데이터를 사용하여 강조색을 사용한 선 그래프를 그리면 해당 부분이 눈에 도드라지게 된다.

p\_line +  
 ## color를 'grey75'로 설정한 geom\_line 레이어를 생성  
 geom\_line(aes(x = 연도, y = (입학생수)/1000, group = 학교종류), color = 'grey75') +   
 ## 강조할 데이터만 필터링한 데이터를 사용하여  
 geom\_line(data = df\_입학자\_long |> filter(지역 == '전체', 학교종류 %in% c('전문대학')),   
 ## color를 학교종류로 매핑한 geom\_line 레이어 추가  
 aes(x = 연도, y = (입학생수)/1000, group = 학교종류, color = 학교종류)) +   
 ## 색 스케일을 설정  
 scale\_color\_manual(values = c('전문대학' = 'darkred')) +   
 labs(title = '특정 범주를 강조한 선 그래프')



실행결과 7-특정 범주를 강조한 선 그래프

### 분할(facet)과 특정 범주 강조(범프 차트)

강조를 사용한 스파게티 선 그래프는 변수의 전체 범주에서 특정 범주를 확인하는데 적절하다. 하지만 한번에 하나의 강조만을 사용할 수 밖에 없다는 단점이 있는데 모든 범주에 대해 각각의 강조 선 그래프를 그리려면 어떻게 해야할까?

이것을 해결하기 위해 facet\_wrap()과 강조 방법을 사용하여 표현해보겠다.

이를 위해서는 먼저 임시로 사용될 데이터프레임이 하나 필요하다. 이 데이터프레임은 원본 데이터 프레임과 동일한 구조를 지니지만 하나 다른점은 facet\_wrap()에서 사용되는 분할 변수 열 이름을 달리 저장한다는 점이다. 여기서 하나 주의해야할 것이 임시로 만든 데이터프레임에 분할 변수로 사용할 열과 동일한 열을 제거해야 한다는 것이다. 만약 분할 변수와 동일한 열이 있으면 정상적으로 생성되지 않는다.[^2]

이 코드의 원리는 facet\_wrap()의 분할 변수를 통해 그래프를 분할 시키고 동일한 범주를 가졌지만 열 이름이 다른 데이터로 흐릿한 선들을 그려준다. 흐릿한 선들은 facet\_wrap()에 지정되는 분할 열과 동일한 열이름이 아니기 때문에 분할되어 그려지지 않고 전체 분할 그래프에 공통적으로 그려지게 된다.

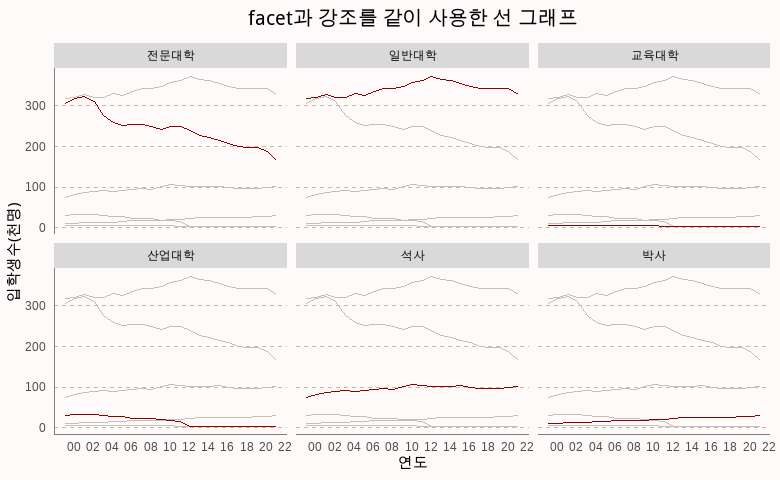
이러한 형태의 그래프를 범프 차트(Bump Chart)라고 한다. 범프 차트의 장점은 전체 변량 사이에서의 모든 변량의 확인이 가능하다는 것이다. 하지만 일반적 선 그래프는 선들이 모여있다보니 강조되는 특정 범주의 데이터를 확인하기 쉽지만 전체 범주의 변화를 각각 확인하기 어렵다.

[^2] : 만약 이 부분이 이해되지 않는다면 df\_입학자\_long\_temp 생성 코드에서 select(-학교종류)를 제외하고 실행해보라.

## 임시 데이터프레임을 만드는데  
df\_입학자\_long\_temp <- df\_입학자\_long |>   
 filter(지역 == '전체', 학교종류 %in% c('전문대학', '일반대학', '교육대학', '산업대학', '석사', '박사')) |>  
 ## 학교종류1 열은 학교종류 열과 동일한 열  
 mutate(학교종류1 = 학교종류) |>  
 ## 학교종류 열을 삭제  
 select(-학교종류)  
  
head(df\_입학자\_long\_temp)

| 연도 | 지역 | 입학생수 | 학교종류1 |
| --- | --- | --- | --- |
| 1999-02-14 | 전체 | 306802 | 전문대학 |
| 1999-02-14 | 전체 | 4840 | 교육대학 |
| 1999-02-14 | 전체 | 319278 | 일반대학 |
| 1999-02-14 | 전체 | 30882 | 산업대학 |
| 1999-02-14 | 전체 | 73826 | 석사 |
| 1999-02-14 | 전체 | 10447 | 박사 |

p\_line +  
 ## 임시데이터프레임 데이터로 X축을 연도, Y축을 입학생수/1000, group을 학교종류1으로 매핑하고 color를 'grey75'로 설정한 geom\_line 레이어 생성  
 geom\_line(data = df\_입학자\_long\_temp, aes(x = 연도, y = (입학생수)/1000, group = 학교종류1), color = 'grey75') +   
 ## 원 데이터를 사용하여 각각의 그룹에 color를 darkred로 설정한 geom\_line 레이어 추가  
 geom\_line(aes(x = 연도, y = (입학생수)/1000, group = 1), color = 'darkred') +  
 ## 추가된 레이어를 학교종류열의 데이터로 분할  
 facet\_wrap(~ 학교종류) +   
 labs(title = 'facet과 강조를 같이 사용한 선 그래프')



실행결과 7-facet과 강조를 같이 사용한 선 그래프

# 축이 더 필요해? 아님 잘라? – 축 다루기

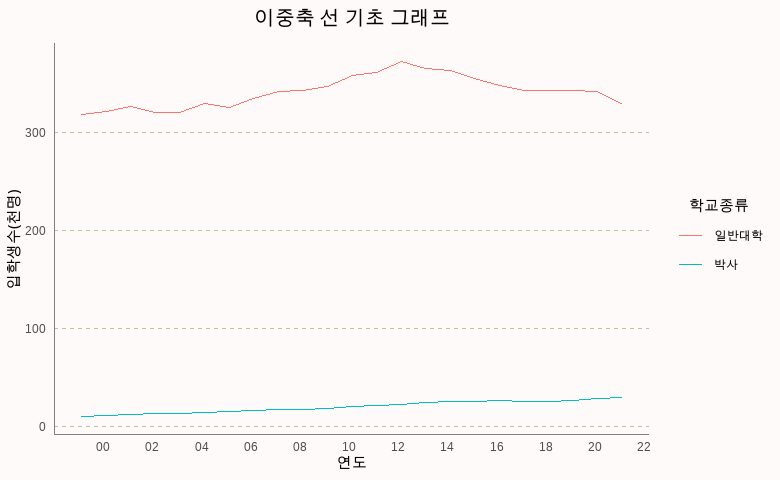
ggplot에서는 특별한 설정을 하지 않아도 자동적으로 X축과 Y축의 눈금과 범위를 자동적으로 설정해준다. 하지만 다변량(Multi-variated) 선 그래프를 그릴때 맞닥치는 가장 흔한 문제는 한쪽 값이 너무 크거나 작으면 그래프가 효과적이지 않다는 것이다. 이럴경우는 축을 적절히 조절함으로써 데이터를 효과적으로 시각화할 수 있다.

## 이중 축 선 그래프

이중 축 선 그래프는 보통 왼쪽과 오른쪽에 서로 다른 축척을 지닌 Y축을 배치함으로써 하나의 선은 왼쪽 Y축에, 다른 하나의 선은 오른쪽 Y축에 대응키는 방법이다. 사실 오른쪽에 위치하는 두번째 Y축은 스케일로 매핑되는 것이 아니고 그저 선과 라벨의 집합일 뿐이다. 따라서 두번째 축에 매핑되는 것이 아니고 첫번째 축에 매핑하기 위해 데이터 값을 일정 비율로 곱해서 스케일을 맞춰주는 것이다.

이를 위해 데이터를 다음과 같이 전처리 하겠다. 입학생의 차이가 큰 ’일반대학’과 ’박사’의 선 그래프를 그리면 다음과 같다.

p\_dual\_axis <- df\_입학자\_long |> filter(지역 == '전체', 학교종류 %in% c('일반대학', '박사')) |>  
 ggplot(aes(x = 연도, y = (입학생수)/1000)) +   
 scale\_x\_date(date\_breaks = "2 years", date\_labels = '%y') +  
 labs(x = '연도', y = '입학생수(천명)')  
  
p\_dual\_axis +  
 ## X축을 연도, Y축을 입학생수/1000, group을 학교종류로 매핑한 geom\_line 레이어 생성  
 geom\_line(aes(x = 연도, y = (입학생수)/1000, group = 학교종류, color = 학교종류)) +   
 labs(title = '이중축 선 기초 그래프')



실행결과 7-이중축 선 기초 그래프

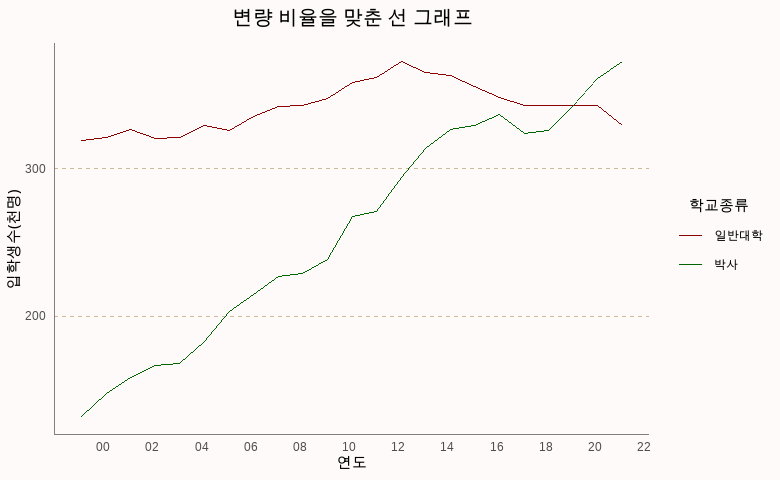
앞의 그래프를 보면 양쪽 변량간의 차이가 커서 효과적으로 그려지지 않았다. 그렇다면 먼저 첫번째 변량인 ’일반대학’과 두번째 변량인 ’박사’의 비율을 구해야 한다. 이 비율은 두 변량이 하나의 플롯에 적절하게 표현되기 위해서는 필요한 비율로써, 양쪽 변량의 최대값 혹은 최소값의 비율이다. 이 비율을 사용하여 한 쪽 변량의 데이터 값을 전체적으로 변환해서 그려야 한다는 것이다. 여기서는 ’일반대학’의 최대값과 ’박사’의 최대값의 비율을 산출했다.

## 일반대학 입학생의 최대값 산출  
first\_max <- df\_입학자\_long |>  
 filter(지역 == '전체', 학교종류 %in% c('일반대학')) |>  
 summarise(max = max(입학생수)) |>  
 ## summarise()의 결과는 데이터프레임이기 때문에 pull()로 벡터로 변환  
 pull()  
  
## 박사 입학생의 최대값 산출  
second\_max <- df\_입학자\_long |>  
 filter(지역 == '전체', 학교종류 %in% c('박사')) |>  
 summarise(max = max(입학생수)) |>  
 pull()  
  
## 두값을 나누어 비율 산출   
ratio <- first\_max/second\_max  
  
ratio

## [1] 12.60191

비율은 12.6으로 산출되었다. 일반대학의 입학생 최대값은 박사 입학생의 최대값의 12.6배라는 것이다. 그럼 박사 입학생의 12.6배를 하거나 일반대학 입학생을 12.6으로 나누면 두 변량들의 범위가 비슷해 질 것이다.

p\_dual\_axis1 <- p\_dual\_axis +  
 ## 일반대학 데이터만 필터링  
 geom\_line(data = df\_입학자\_long |> filter(지역 == '전체', 학교종류 %in% c('일반대학')),   
 ## X축을 연도, Y축을 입학생수/1000, 단일그룹, color를 학교종류로 매핑한 geom\_line 레이어 생성  
 aes(x = 연도, y = 입학생수 / 1000, group = 1, color = 학교종류)) +  
 ## 박사 데이터만 필터링  
 geom\_line(data = df\_입학자\_long |> filter(지역 == '전체', 학교종류 %in% c('박사')),   
 ## X축을 연도, Y축을 입학생수/1000 \* ratio, 단일그룹, color를 학교종류로 매핑한 geom\_line 레이어 생성  
 aes(x = 연도, y = (입학생수 / 1000) \* ratio, group = 학교종류, color = 학교종류)) +  
 scale\_color\_manual(values = c('일반대학' = 'darkred', '박사' = 'darkgreen')) +   
 labs(title = '변량 비율을 맞춘 선 그래프')  
  
p\_dual\_axis1



실행결과 7-변량 비율을 맞춘 선 그래프

앞의 플롯에서 보듯이 이제 대략 두 변량이 효과적으로 표현된다. 하지만 이 그래프에서 데이터는 박사 입학생수의 수치를 측정할 축이 없기 때문에 데이터 값을 측정할 수 없다. 따라서 박사 입학생 데이터를 측정할 두 번째 Y축을 설정해 주어야 한다.

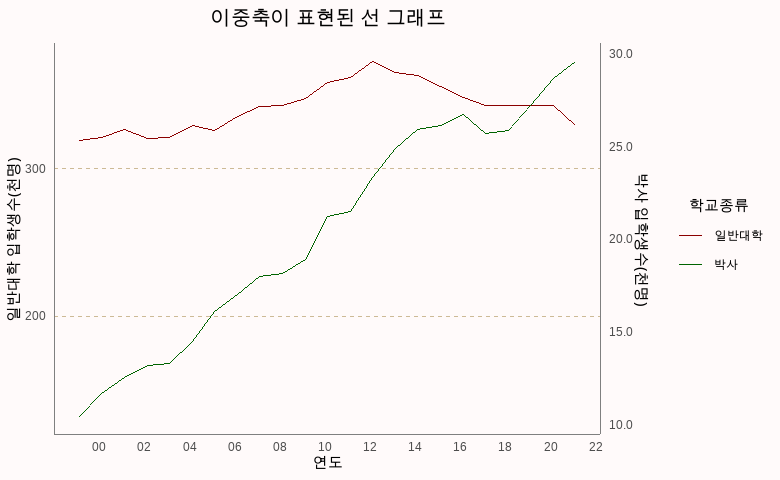
두 번째 Y축은 scale\_y\_continuous()의 sec.axis 매개변수를 통해 설정할 수 있는데 이 매개변수는 다시 sec\_axis()를 사용하여 설정할 수 있다.

sec\_axis()에 사용되는 사용법과 주요 매개변수는 다음과 같다.

sec\_axis(trans = NULL, name = waiver(), breaks = waiver(), labels = waiver(), guide = waiver())  
- trans : 첫 번째 축에 대비하여 두 번째 축을 설정할 변환식 설정  
- name : 두 번째 축의 이름 설정  
- breaks : 두 번째 축의 눈금 설정  
- labels : 두 번째 축의 눈금 라벨 설정  
- guide : 범례 설정

trans는 설정된 Y축의 값을 어떻게 변환할 지에 대한 함수식을 전달하는 매개변수이다. 여기서 중요하게 사용되는 키워드는 ~.인데 두번째 Y축값에 대응되는 첫번째 Y축의 값이라고 생각하면 쉽다.

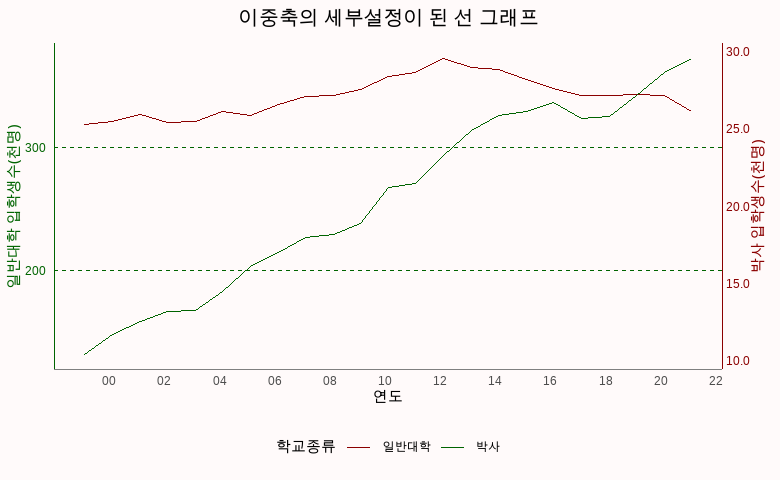
p\_dual\_axis2 <- p\_dual\_axis1 +  
 scale\_y\_continuous(name = "일반대학 입학생수(천명)", ## 첫 번째 Y축 이름 설정  
 labels = scales::comma,   
 ## 두 번째 Y축 변환식과 이름 설정  
 sec.axis = sec\_axis(trans=~ ./ratio,   
 ##두 번째 축의 제목 설정  
 name="박사 입학생수(천명)",  
 ## 라벨은 콤마형  
 labels = scales::comma)   
 ) +  
 labs(title = '이중축이 표현된 선 그래프')  
  
p\_dual\_axis2



실행결과 7-이중축이 표현된 선 그래프

범례가 오른쪽에 있다보니 그래프 표현 공간이 줄어든다. 또 범례로 선과 학교종류를 설명했는데 축의 색도 선의 색과 맞춰주면 그래프를 이해하기가 더 쉬워질 듯 하다.

p\_dual\_axis2 +  
 theme(legend.position = 'bottom', ## 범례의 위치는 아래  
 ## 왼쪽 축의 선 색은 다크 그린  
 axis.line.y.left = element\_line(color = 'darkgreen'),   
 ## 왼쪽 축의 글자 색은 다크 그린  
 axis.text.y.left = element\_text(color = 'darkgreen'),   
 ## 왼쪽 축의 제목 색은 다크 그린  
 axis.title.y.left = element\_text(color = 'darkgreen'),  
 ## 패널의 Y축 주 눈금선 색은 다크 그린  
 panel.grid.major.y = element\_line(color = 'darkgreen'),   
 ## 오른쪽 축의 선 색은 다크 그린  
 axis.line.y.right = element\_line(color = 'darkred'),   
 ## 오른쪽 축의 글자 색은 다크 그린  
 axis.text.y.right = element\_text(color = 'darkred'),   
 ## 오른쪽 축의 제목 색은 다크 그린  
 axis.title.y.right = element\_text(color = 'darkred')  
 ) +   
 labs(title = '이중축의 세부설정이 된 선 그래프')



실행결과 7-이중축의 세부설정이 된 선 그래프

이중 축 선 그래프를 사용할 떄도 앞서 선 그래프 분할에서 설명했듯이 데이터의 왜곡에 신경써야한다. 앞의 선 그래프에서도 보듯이 박사 입학생이 매우 크게 증가한 것으로 보이지만 실질적인 데이터 증가량은 일반대학에 비해 크지 않다. 이러한 데이터 변화량의 차이를 분명히 설명해야 한다.

## 축 자르기

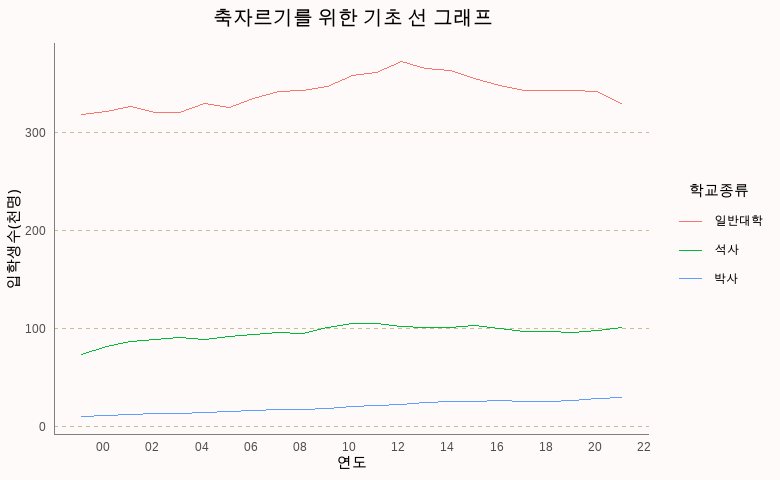
다변량을 표현하는 플롯에서 변량의 값 범위의 차이가 너무 커서 플롯이 효율적으로 그려지지 않는 경우를 처리하기 위해 이중축을 사용하는 방법을 설명하였다. 하지만 세 개 이상의 다변량을 표현할 때는 이중축으로는 해결이 어렵다.

이런 경우 간혹 축의 일부를 잘라서 표현하는 경우가 있다. 축의 일부 값을 잘라서 데이터가 표현되지 않는 부분을 최소화함으로써 플롯의 공간을 최대한 활용하는 방법이다.

이에 대한 해답으로 ggbreak 패키지를 소개한다. ggbreak 패키지는 축의 일부를 잘라서 플롯의 빈공간을 최소화함으로써 플롯에 표현되는 데이터들이 그 특성을 잘 나타낼수 있도록 만드는 함수를 제공한다.

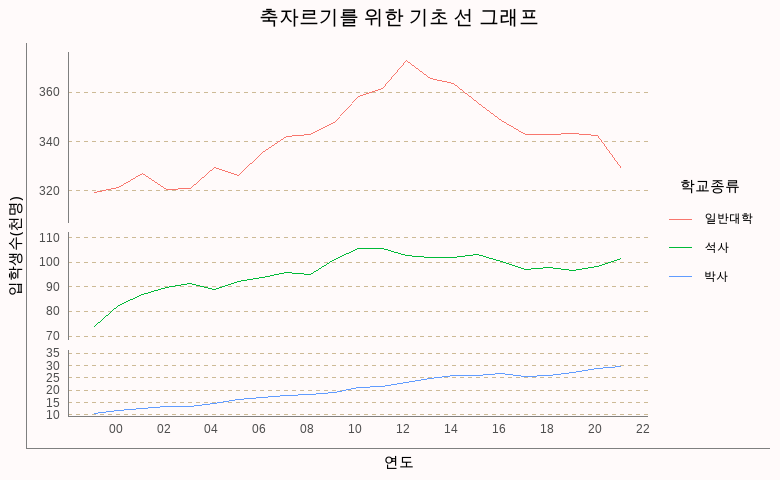
우선 앞의 이중 축 선 그래프에서 사용된 데이터에 석사 입학생을 추가한다.

p\_cut\_axis <- df\_입학자\_long |> filter(지역 == '전체', 학교종류 %in% c('일반대학', '석사', '박사')) |>  
 ggplot(aes(x = 연도, y = (입학생수)/1000)) +   
 geom\_line(aes(group = 학교종류, color = 학교종류)) +  
 scale\_x\_date(date\_breaks = "2 years", date\_labels = '%y') +  
 labs(title = '축자르기를 위한 기초 선 그래프', x = '연도', y = '입학생수(천명)')  
  
p\_cut\_axis



실행결과 7-축 자르기를 위한 기초 선 그래프

## ggbreak 패키지 설치  
if(!require(ggbreak)) {  
 install.packages('ggbreak')  
 library(ggbreak)  
}  
  
p\_cut\_axis +   
 ## ggbreak 패키지의 scale\_y\_break()를 사용하여 Y축의 110~310까지를 자름  
 scale\_y\_break(c(110, 310)) +   
 ## ggbreak 패키지의 scale\_y\_break()를 사용하여 Y축의 35~70까지를 자름  
 scale\_y\_break(c(35, 70)) +   
 theme(axis.title.y = element\_text(angle = 90))



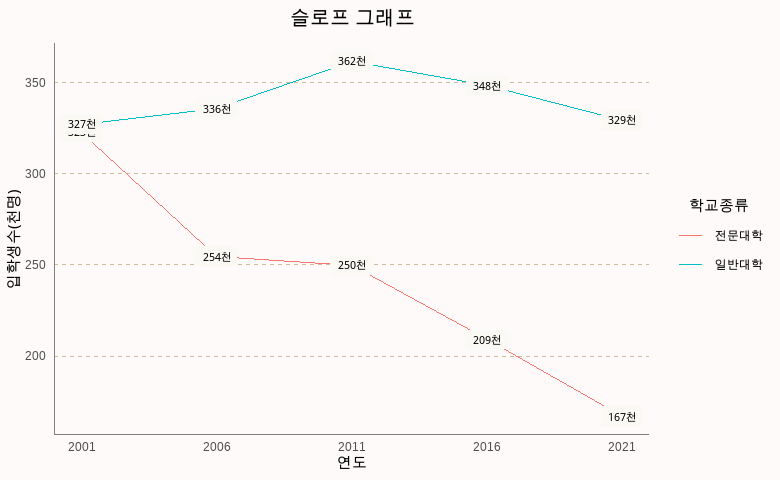
실행결과 7-축을 자른 선 그래프

# 20년은 너무 많아. 단순화해 보자 – 기울기 그래프

기울기 차트는 시간의 흐름에 따라 데이터를 선으로 표현한다는 점에서 선 그래프와 유사하지만 주요 변곡점이 되는 두개 혹은 세개의 시간을 선정하여 해당 기간동안의 값의 변화를 시각화하는 방법이다. 보통 두개의 시간을 두고 이 두 시간의 값을 선으로 이어주면 이 선의 기울기가 표현되는데 이 기울기를 위주로 데이터를 시각화하는 방법이다. 기울기 혹은 시작점의 데이터, 마지막점의 데이터에 따른 순위 변화도 표현이 가능한 시각화 방법이다.

기울기 차트는 geom\_line()으로 구현한다. 아래의 코드는 전문대학과 일반대학의 2001년부터 2021년까지의 입학생의 변화량을 5년 단위의 기울기 차트로 표현한 코드이다.

## df\_입학자\_long에서 2001년부터 2021년까지 5년단위의 데이터만 필터링  
df\_입학자\_long |> filter(지역 == '전체', lubridate::year(연도) %in% c(seq(from = 2001, to = 2021, by = 5)), 학교종류 %in% c('전문대학', '일반대학')) |>  
 ggplot(aes(x = 연도, y = (입학생수)/1000)) +   
 ## geom\_line 레이어를 생성  
 geom\_line(aes(group = 학교종류, color = 학교종류)) +  
 ## 데이터 값을 표현하는 geom\_label 레이어 생성  
 geom\_label(aes(group = 학교종류, label = paste0(round(입학생수/1000, 0), '천')), label.size = NA, size = 3, fill = '#fbf9f4') +   
 scale\_x\_date(breaks = as.Date(as.character(seq(from = 2001, to = 2021, by = 5)), format = '%Y'), date\_labels = '%Y') +  
 labs(title = '슬로프 그래프', x = '연도', y = '입학생수(천명)')



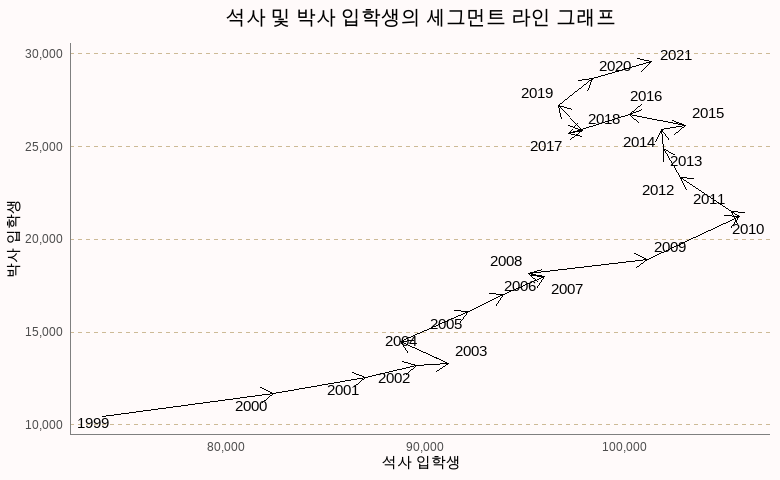
실행결과 7-슬로프 그래프

# 데이터들은 어떻게 움직이고 있는 거야? - 세그먼트 그래프

세그먼트 라인 그래프는 산점도와 같이 연속형 X축과 Y축으로 설정된 공간에서 순차적으로 기록된 데이터를 세그먼트 라인으로 연결한 시각화 방법이다. 이 세그먼트 라인 그래프는 연결된 산점도(Connected Scatter Plot)라고도 불리는데, 산점도와 같이 연속형 X축과 Y축으로 매핑된 두 변수 간의 관계를 데이터의 순차적 순서에 의해 보여준다.

세그먼트 라인 그래프는 데이터의 발생 순서에 따른 데이터 2개의 변수의 변화를 보여주는데 데이터의 사이즈가 커지면 산점도에서 발생한 오버플로팅 문제가 발생하기 때문에 가급적 데이터를 대표할 수 있는 시간적이나 이벤트를 잘 설정하여야 한다. 다음은 시간의 흐름에 따른 석사 입학생과 박사 입학생의 변화를 세그먼트 라인 으로 시각화한 것이다.

df\_입학자 |> filter(지역 == '전체') |>  
 ggplot(aes(x = 석사, y = 박사)) +   
 ggrepel::geom\_text\_repel(aes(label = 연도)) +  
 ## 연도별 데이터를 geom\_segment()를 이용하여 연결  
 geom\_segment(aes(  
 xend=c(tail(석사, n=-1), NA),   
 yend=c(tail(박사, n=-1), NA)  
 ),  
 arrow=arrow(length=unit(0.3,"cm"))  
 ) +   
 ## X축의 라벨을 콤마형으로 설정  
 scale\_x\_continuous(labels = scales::comma) +   
 ## Y축의 라벨을 콤마형으로 설정  
 scale\_y\_continuous(labels = scales::comma) +  
 labs(title = '석사 및 박사 입학생의 세그먼트 라인 그래프', x = '석사 입학생', y = '박사 입학생')



실행결과 7-석사 및 박사의 입학생 세그먼트 라인 그래프

# 늘은 거야? 줄은 거야? - 덤벨 차트

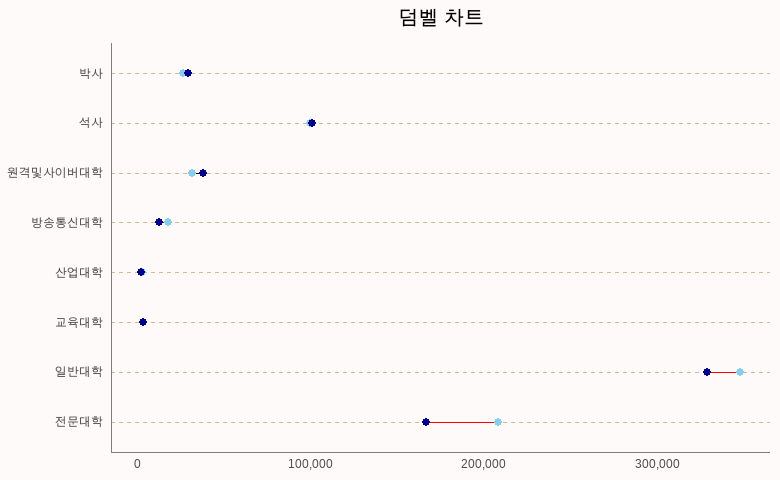
덤벨 차트는 양쪽 끝이 동그란 포인트로 이루어진 세그먼트 차트를 말한다. 생긴 모양이 운동할 떄 사용하는 아령(Dumbbell)과 닮았다고 해서 붙여진 이름이다. 덤델 차트는 두 시점간의 데이터 양의 차이의 크기를 시각화하거나 두 지점 사이의 상대적 위치(증가 혹은 하락)을 표현할 때 효과적인 시각화 방법이다.

덤벨 차트가 잘 표현되기 위해서는 X축이나 Y축에 매핑된 두 변수 중 하나가 이산형 변수이어야 한다.

ggplot2를 사용하여 덤벨 차트를 만들때는 보통 세가지 방법이 있는데 첫번째는 ggalt 패키지의 geom\_dumbbell()을 이용하는 방법, 두번째는 geom\_line()을 사용하는 방법, 세번째는 geom\_segment()를 사용하는 방법이다.

다음의 코드는 첫번째 방법인 ggalt 패키지의 geom\_dumbbell()을 이용하는 방법이다. 먼저 덤벨 차트로 비교할 연도 데이터인 2016년과 2021년 데이터를 필터링하고 2016년과 2021년 데이터를 열로 변환한 넓은 형태의 데이터프레임으로 변환한다. 이후 2016년과 2021년 데이터를 geom\_point()로 그려주고 이 두 점사이를 geom\_segment()로 이어주면 덤벨 차트가 만들어진다.

## df\_입학자\_long에서 연도가 2016과 2021만 필터링  
df\_dumbbell <- df\_입학자\_long |> filter(지역 == '전체', lubridate::year(연도) %in% c(2016, 2021)) |>  
 ## 열이름으로 만들기 위해 연도만 추출  
 mutate(연도 = lubridate::year(연도)) |>  
 ## 사용할 열만 선택  
 select(1, 3, 4) |>   
 ## 연도를 열 이름으로 하고 데이터를 입학생수로 설정한 넓은 형태의 데이터프레임 변환  
 pivot\_wider(names\_from = 연도, values\_from = 입학생수) |>  
 ## 성장률 산출  
 mutate(growth = `2021` - `2016`)  
  
df\_dumbbell |>  
 ggplot() +  
 ## X 시작점이 2016, X 끝점이 2021, y축이 학교종류로 매핑된 geom\_dumbbell 레이어 생성  
 ggalt::geom\_dumbbell(aes(x = `2016`, xend = `2021`, y = 학교종류),  
 ## 시작점 색은 'skyblue' 끝점 색은 'darkblue'로 설정  
 colour\_x="skyblue", colour\_xend = 'darkblue',   
 ## 시작점 크기는 2, 끝점 크기는 2  
 size\_x = 2, size\_xend = 2,   
 ## 선 색은 성장률이 음수이면 'red', 아니면 'black'  
 color = ifelse(df\_dumbbell$growth < 0, 'red', 'black')) +   
 scale\_x\_continuous(label=scales::comma) +   
 scale\_color\_manual(values = c('2016' = 'skyblue', '2021' = 'darkblue')) +  
 labs(x=NULL, y=NULL, title="덤벨 차트")



실행결과 7-덤벨 차트

1. 날짜형 포맷은 시간형 포맷을 포함한다. [↑](#footnote-ref-20)