1. 구성(Composition)의 시각화

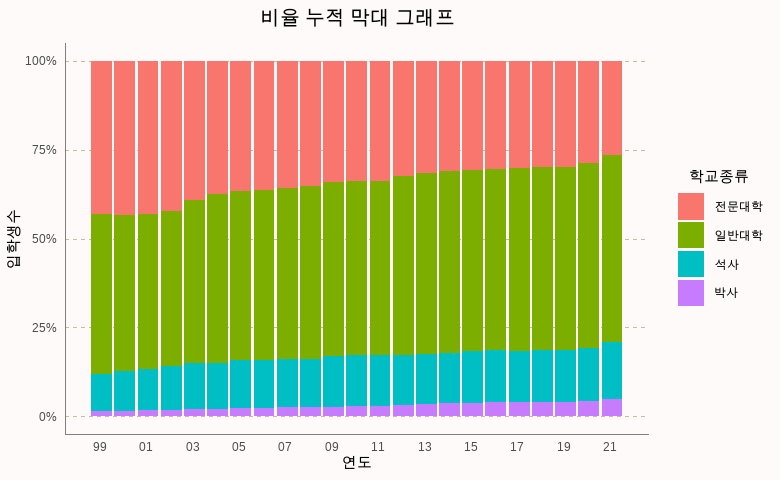
구성의 시각화는 특정 데이터를 구성하는 각 부분의 비율을 시각화하는 것이다. 보통의 경우 비율이나 백분율을 통해 시각화하며 막대나 원을 세분화하여 그 비율을 보여준다. 여러개의 세부 범주로 나누어진 많은 범주들의 구성 비율을 나타내는데 비율 누적 막대 그래프가 많이 사용된다. 반면 단 하나의 범주에 대한 세부 범주의 구성 비율을 나타낼 때는 누적 막대 그래프 보다는 동그란 파이 차트나 도넛 차트로 나타낼 수 있다. 하지만 시각화 전문가들은 파이 차트나 도넛 차트는 권장하지 않는다. 사람의 시각으로는 파이 차트나 도넛 차트의 그 비율을 정확히 인지하기가 어렵기 때문이다. 이에 대한 대안으로 트리맵이나 와플 차트가 이용된다.

# 비율 누적 막대 그래프

앞 장에서 막대 그래프의 타입에 stack, dodge, fill의 세가지가 있다고 설명하였다. 이중 stack과 dodge의 예는 이미 살펴보았지만 fill 타입은 설명하지 않았다. 이 fill 타입의 막대그래프가 비율 누적 막대 그래프이다. 앞선 두 타입의 막대 그래프와 다른 것은 stack과 dodge 타입의 막대 그래프는 데이터 값만큼 막대의 길이가 결정되지만 비율 누적 막대 그래프는 막대 길이가 모두 1로 같게 그려진다는 것이다. 결국 세부 분류의 비율의 합은 1이라는 의미이다. 따라서 전체 데이터 값의 비교가 불가능하고 단지 그 세부 분류의 비율만을 확인할 수 있다는 것이다.

비율 누적 막대 그래프는 geom\_col()이나 geom\_bar()에서 position 매개변수를 ’fill’로 설정하면 내부적으로 세부 구분들의 값이 비율로 변환되어 막대의 길이로 표시된다.

## df\_입학자\_long의 데이터 중 일부를 필터링  
df\_col\_fill <- df\_입학자\_long |>  
 filter(학교종류 %in% c('전문대학', '일반대학', '석사', '박사'), 지역 == '전체') |>  
 ### 표시 순서를 맞추기 위해 fct\_relevel로 레벨을 재조정  
 mutate(학교종류 = fct\_relevel(학교종류, '전문대학', '일반대학', '석사', '박사')) |>  
 ## X축 스케일을 날짜형으로 맞추기 위해 연도를 as.Date()를 사용해 날짜형으로 변환  
 mutate(연도 = as.Date(연도, format = '%Y'))  
  
p\_col\_fill <- df\_col\_fill |>  
 ggplot()  
  
p\_col\_fill1 <- p\_col\_fill +  
 ## X축을 연도, Y축을 입학생수, fill을 학교종류로 매핑하고 position을 fill로 설정  
 geom\_col(aes(x = 연도, y = 입학생수, fill = 학교종류), position = 'fill') +  
 ## X축 스케일을 날짜형으로 설정하고 눈금은 2년마다 라벨은 2자리 연도로 표기  
 scale\_x\_date(date\_breaks = '2 years', date\_labels = '%y') +  
 ## Y축 스케일을 퍼센트형으로 설정  
 scale\_y\_continuous(labels = scales::percent) +  
 labs(title = '비율 누적 막대 그래프')  
  
p\_col\_fill1



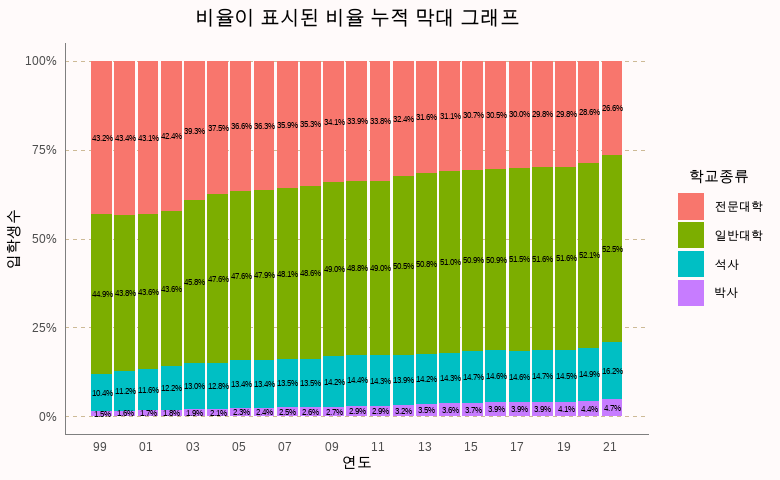
실행결과 9-비율 누적 막대 그래프

앞의 그래프는 각 연도별 입학생수의 학교구분별 비율을 표현한 막대 그래프이다. 이 그래프에 데이터 값이 표현되지 않으니 뭔가 허전하다. 다만 비율 누적 막대 그래프에서는 자체 데이터 값을 표시하는 것은 의미가 없고 전체 중의 비율을 표시하는 것이 의미가 있다. 이를 위해서 각각의 비율을 먼저 산출해 주어야 한다. 앞서 전처리한 데이터와 동일하게 필터링한 데이터에 연도별로 그룹화하고 mutate()를 사용하여 각각의 연도내에서의 비율을 산출한 데이터를 저장해 둔다.

df\_col\_fill\_rate <- df\_입학자\_long |>  
 filter(학교종류 %in% c('전문대학', '일반대학', '석사', '박사'), 지역 == '전체') |>  
 mutate(학교종류 = fct\_relevel(학교종류, '전문대학', '일반대학', '석사', '박사')) |>  
 group\_by(연도) |>  
 mutate(비율 = 입학생수 / sum(입학생수), 연도 = as.Date(연도, format = '%Y')) |>  
 ungroup()

산출한 비율을 geom\_text()를 사용하여 표시해주는데 position을 position\_stack()을 사용하여 위치를 설정해준다.

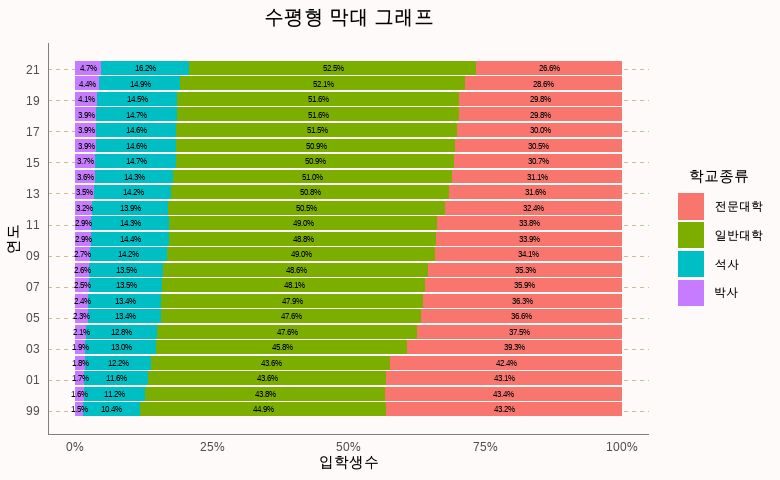
p\_col\_fill2 <- p\_col\_fill1 +   
 ## geom\_text()에 비율이 저장된 df\_col\_fill\_rate를 사용  
 geom\_text(data = df\_col\_fill\_rate,  
 ## X축은 연도, Y축은 비율, label은 비율을 퍼센트 형식, fill은 학교종류로 매핑  
 aes(x = 연도, y = 비율, label = scales::percent(비율, accuracy = 0.1), fill = 학교종류),   
 ## position은 position\_stack()을 사용하여 stack형 위치를 설정하고 수직위치(Vjust)를 중간(0.5), size를 2로 설정  
 position = position\_stack(vjust = 0.5), size = 2) +  
 labs(title = '비율이 표시된 비율 누적 막대 그래프')  
  
p\_col\_fill2



실행결과 9-비율이 표시된 비율 누적 막대 그래프

비율이 잘 들어간 것으로 보이는가? 글자가 너무 작아서 잘 눈에 띄이지 않는 듯하다. 하지만 글자를 키우면 글자가 막대 밖으로 빠져나와 보기가 안좋아진다. 보통 보고서의 종이는 가로보다는 세로가 길기 때문에 이를 세로로 길게 늘여뜨리면 보고서에 넣기가 좋아진다. 따라서 축을 다음과 같이 전환해보자.

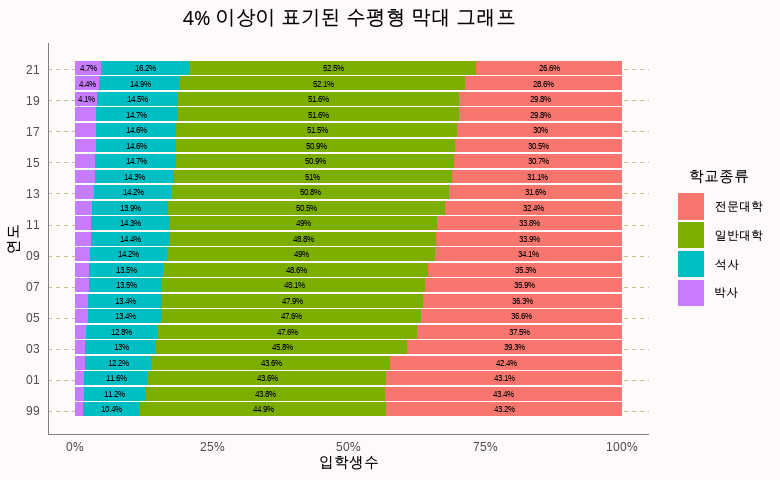
p\_col\_fill3 <- p\_col\_fill2 +   
 ## 축 전환  
 coord\_flip() +   
 labs(title = '수평형 막대 그래프')  
  
p\_col\_fill3



실행결과 9-수평형 막대 그래프

앞의 그래프에서 하나 눈에 거슬리는 부분이 좌측 하단의 1~4%정도 구간의 비율이 막대 범위를 벗어나고 있다. 이 부분을 수정하기 위해서는 geom\_text()의 label이 4를 넘는 곳에는 비율을 넣고 label이 4보다 작으면 label을 없애주도록 하려면 다음과 같이 할 수 있다.

p\_col\_fill3 <- p\_col\_fill1 +   
 ## geom\_text()에 비율이 저장된 df\_col\_fill\_rate를 사용  
 geom\_text(data = df\_col\_fill\_rate,  
 ## X축은 연도, Y축은 비율로 매핑  
 aes(x = 연도, y = 비율,   
 ## label은 0.04보다 크면 비율을 소수점 3자리에서 반올림한 후 100을 곱해서 '%'를 붙여주고 아니면 공란을 매핑  
 label = ifelse(비율 > 0.04, paste0(round(비율, 3) \* 100,"%"), ''),  
 ## fill은 학교종류로 매핑  
 fill = 학교종류),   
 ## position은 position\_stack()을 사용하여 stack형 위치를 설정하고 수직위치(Vjust)를 중간(0.5), size를 2로 설정  
 position = position\_stack(vjust = 0.5), size = 2) +  
 ## 축을 전환  
 coord\_flip() +   
 labs(title = '4% 이상이 표기된 수평형 막대 그래프')  
  
p\_col\_fill3



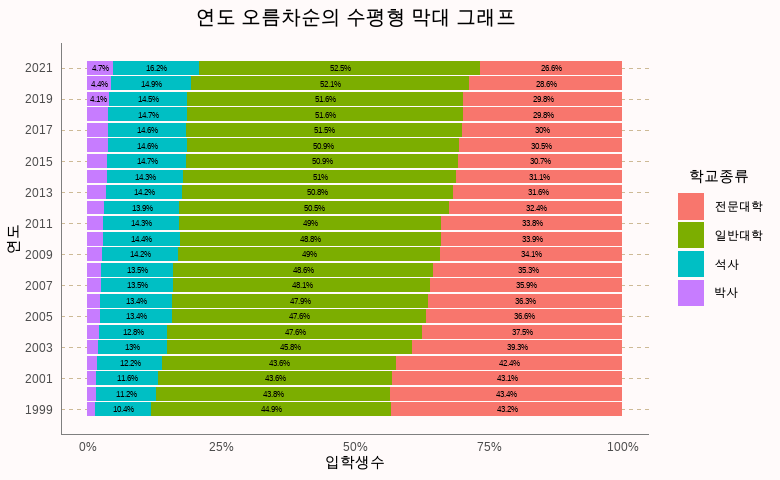
실행결과 9-4% 이상이 표기된 수평형 막대 그래프

앞의 그래프를 보면 연도의 배열이 내림차순으로 정렬되어 있다. 보통 연도는 오름차순으로 정렬된다. 이를 전환하기 위해서는 scale\_x\_date()를 다시 설정해야한다. 하지만 현재 ggplot2에서 제공하는 scale\_x\_date()에서는 세로축에 표현된 날짜형 타입을 오름차순으로 정렬해주는 옵션을 제공하지 않는다. 따라서 이를 위해서는 두가지 방법을 사용할 수 있다. 첫번쨰 방법은 연도를 다시 수치형 변수로 바꾸어 오름차순으로 정렬하는 방법이 있고 나머지 하나의 방법은 특별한 변환 객체를 사용하는 방법이다. 이 변환 객체는 ggplot2의 개발자인 Hadley Wickham이 직접 제공한 것으로 차후 ggplot2 패키지에 반영하겠다고 하였지만 아직 반영되지 않았다.[[1]](#footnote-20) 여기서는 첫번째 방법을 설며아겠다.

앞에서 생성한 ggplot 객체에 이미 scale\_x\_date()가 설정되어 있기 때문에 다시 X축에 대한 스케일 함수를 적용하면 중복적 스케일의 적용이라는 에러를 내게 된다. 따라서 처음부터 다시 만드는 것은 앞선 코드들을 되짚어서 만들여야 한다.[[2]](#footnote-21)

먼저 연도 정렬에 사용하기 위해 날짜형 변수에서 연도를 추출한 변수를 만들어준다. 이후 이 변수를 X축에 매핑한 후 X축의 스케일을 연속형으로 설정(scale\_x\_continuous())하고 2021부터 1999까지 -2마다 하나씩 눈금(breaks = seq(from = 1999, to = 2021, by = 2))을 만들어준다.

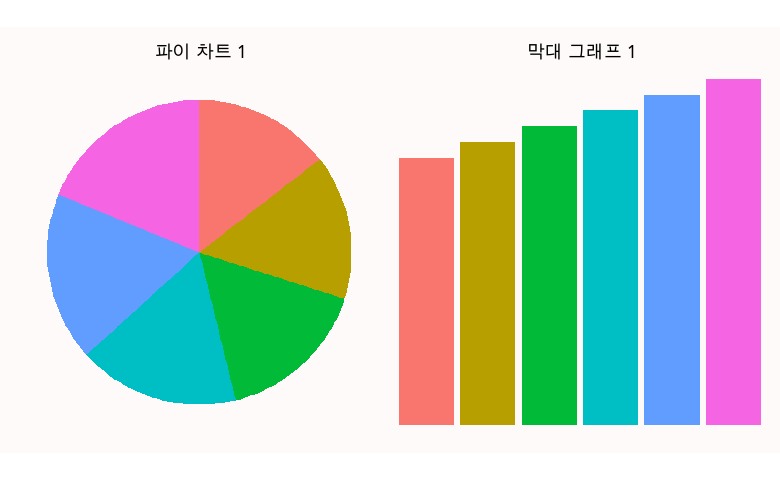
p\_col\_fill4 <- df\_col\_fill\_rate |>  
 ## 연도 정렬에 사용할 열을 생성  
 mutate(연도\_char = lubridate::year(연도)) |>  
 ggplot() +  
 ## X축을 연도\_char, Y축을 입학생수, fill을 학교종류로 매핑하고 position을 fill로 설정  
 geom\_col(aes(x = 연도\_char, y = 입학생수, fill = 학교종류), position = 'fill') +  
 geom\_text(  
 ## X축은 연도\_char, Y축은 비율로 매핑  
 aes(x = 연도\_char, y = 비율,   
 ## label은 0.04보다 크면 비율을 소수점 3자리에서 반올림한 후 100을 곱해서 '%'를 붙여주고 아니면 공란을 매핑  
 label = ifelse(비율 > 0.04, paste0(round(비율, 3) \* 100,"%"), ''),  
 ## fill은 학교종류로 매핑  
 fill = 학교종류),   
 ## position은 position\_stack()을 사용하여 stack형 위치를 설정하고 수직위치(Vjust)를 중간(0.5), size를 2로 설정  
 position = position\_stack(vjust = 0.5), size = 2) +  
 ## 축을 전환  
 coord\_flip() +   
 ## X축 스케일의 눈금을 1999부터 2021까지 2마다 한번씩 설정  
 scale\_x\_continuous(breaks = seq(from = 2021, to = 1999, by = -2)) +  
 ## Y축 스케일을 퍼센트형으로 설정  
 scale\_y\_continuous(labels = scales::percent) +  
 labs(title = '연도 오름차순의 수평형 막대 그래프', x = '연도')  
  
p\_col\_fill4



실행결과 9-연도 오름차순의 수평형 막대 그래프

# 파이 차트

파이 차트(원 그래프라고도 한다.)는 전체에서 차지하는 비중을 나타내는 아주 고전적인 방법이다. 이 방법은 과거에 많이 사용되었지만 데이터 시각화 전문가들은 권장하는 방법이 아니다. 표현되는 데이터의 양이 많지 않다. 하지만 더 큰 단점은 사람의 시각적 인지는 각도를 측정하는데 익숙치 않다는 것이다. 길이의 경우는 동일 축에 같은 시작점을 가진다면 쉽게 인지가 가능하지만 각도는 같은 축에 동일한 시작점을 가지더라도 옆에 나열되면 각도의 크기를 시각적으로 판단하기가 쉽지 않다. 다음의 예를 살펴보자.



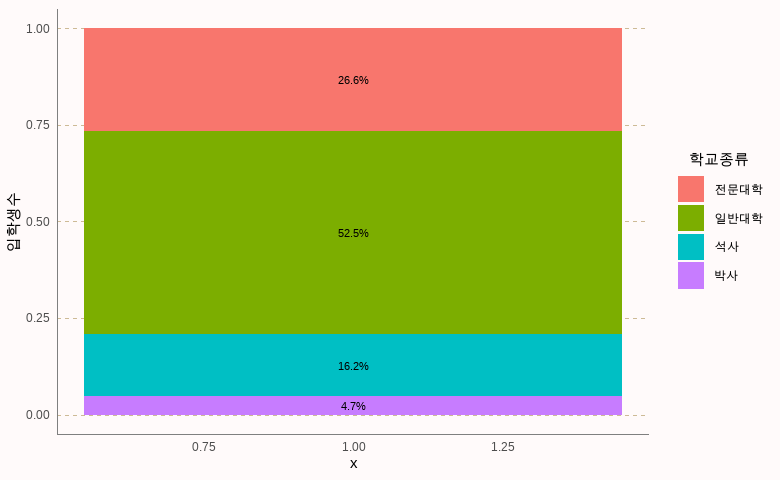
실행결과 9-파이 차트의 나쁜예

앞의 파이 차트에서는 각도가 거의 비슷해 보이지만 막대 그래프로 보면 그 차이가 명확히 보인다. 이를 보면 파이 차트를 왜 권장하지 않는지 알 수 있을 것이다. 따라서 파이 차트를 그릴때는 반드시 그 데이터 레이블과 데이터 값을 표기해 주는 것이 바람직하다. 이 떄문인지 모르지만 ggplot2에서는 파이 차트를 생성하는 전용 함수가 제공되지 않는다. 따라서 ggplot2를 사용해서 파이 차트를 생성하기 위해서는 하나의 막대 그래프를 그리고 Y축을 극 좌표계로 전환해서 돌려버리는 형태로 생성한다.

## 단일 파이 차트

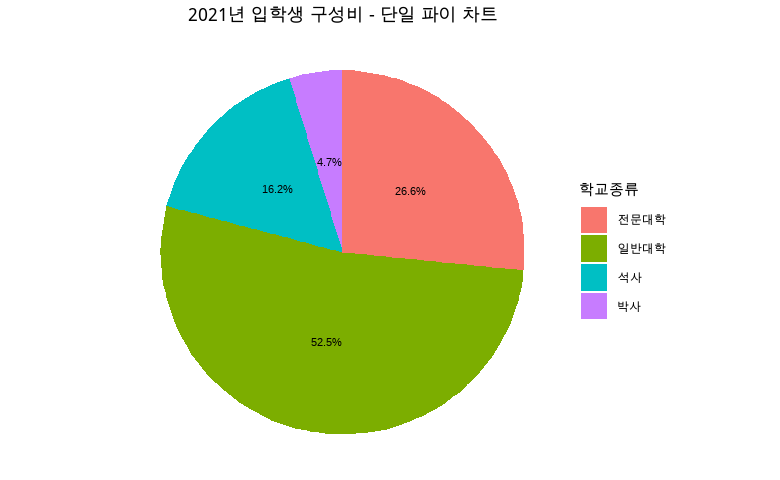
단일 파이 차트프는 우리가 지금까지 흔히 봐왔던 파이 차트의 형태이다. 단일 변수의 변량의 구성 비율을 하나의 원으로 나타낸 그래프를 말한다. 다음과 같이 막대 그래프를 그린 후 극 좌표계로 변환하면 파이 차트가 만들어 진다.

## df\_입학자\_long의 데이터를 필터링하여 파이 차트를 위한 데이터를 생성  
df\_pie <- df\_입학자\_long |>  
 ## 학교종류에서 일부 데이터만 필터링하고  
 filter(학교종류 %in% c('전문대학', '일반대학', '석사', '박사'), 지역 == '전체', 연도 == 2021) |>  
 ## fct\_level로 학교종류 순서를 맞추고  
 mutate(학교종류 = fct\_relevel(학교종류, '전문대학', '일반대학', '석사', '박사')) |>  
 ## 비율을 산출  
 mutate(비율 = 입학생수 / sum(입학생수))  
  
## df\_pie 데이터를 사용하여 ggplot객체를 생성  
p\_pie <- df\_pie |>  
 ggplot()  
  
## P\_pie 객체에서  
p\_pie1 <- p\_pie +  
 ## X축을 1(연도를 매핑해야하나 하나만 필터링해서 데이터가 하나밖에 없음), Y축을 입학생수, fill을 학교종류로 매핑하고 비율로 표현하기 위해 position을 fill로 설정한 geom\_col 레이어를 생성  
 geom\_col(aes(x = 1, y = 입학생수, fill = 학교종류), position = 'fill') +  
 ## X축을 연도(연도를 매핑해야하나 하나만 필터링해서 데이터가 하나밖에 없음), Y축을 비율, label을 백분률로 만들되 정확성을 소수점 한자리, fill을 학교종류로 매핑하고 텍스트 위치를 설정하기 위해 position을 position\_stack()으로 수직정렬을 0.5로 설정하고 크기를 3으로 설정한 geom\_text 레이어 추가  
 geom\_text(aes(x = 1, y = 비율, label = scales::percent(비율, accuracy = 0.1), fill = 학교종류), position = position\_stack(vjust = 0.5), size = 3)  
  
p\_pie1



실행결과 9-단일 파이 차트를 위한 단일 막대 그래프

## 앞서 만든 단일 막대 그래프를 사용  
p\_pie1 +  
 ## 극 좌표계로 Y축을 변환하는데 데이터를 시계방향으로 설정  
 coord\_polar(theta = "y", direction = -1) +  
 ## theme\_void()로 배경등을 모두 없애주고  
 theme\_void() +  
 ## 표 제목의 수평 정렬을 중간으로 설정  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 labs(title = '2021년 입학생 구성비 - 단일 파이 차트')



실행결과 9-단일 파이 차트

## 다중 파이 차트

다중 파이 차트는 앞서 만들었던 단일 파이 차트를 facet\_\*()을 사용하여 여러 파이 차트로 분할한 차트이다. 다중 파이 차트를 생성하는 방법은 앞서 생성했던 단일 파이 차트에 분할을 위한 변수로 facet\_\*()을 적용함으로써 생성할 수 있다.

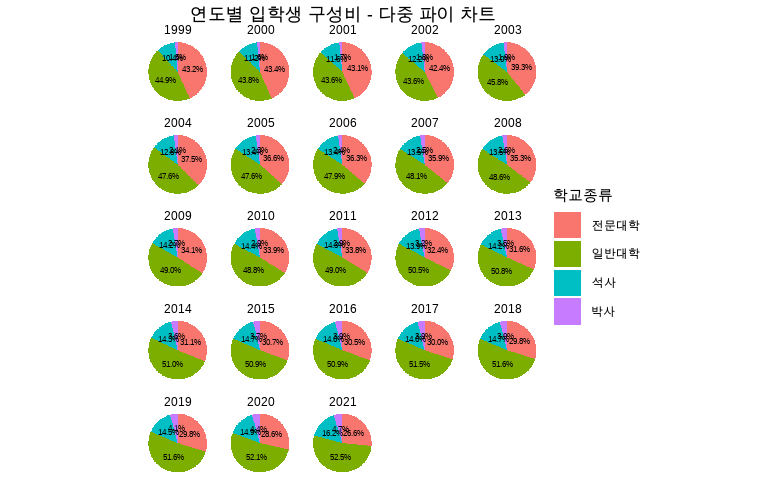
## df\_입학자\_long의 데이터를 필터링하여 파이 차트를 위한 데이터를 생성  
df\_pie\_all <- df\_입학자\_long |>  
 ## 학교종류에서 일부 데이터만 필터링하고  
 filter(학교종류 %in% c('전문대학', '일반대학', '석사', '박사'), 지역 == '전체') |>  
 ## fct\_level로 학교종류 순서를 맞추고  
 mutate(학교종류 = fct\_relevel(학교종류, '전문대학', '일반대학', '석사', '박사')) |>  
 ## facet을 적용하기 위해 연도로 그룹화해주고  
 group\_by(연도) |>  
 ## 각 그룹마다의 비율을 산출하고  
 mutate(비율 = 입학생수 / sum(입학생수)) |>  
 ## 하나의 데이터 프레임으로 다시 변환  
 ungroup()  
  
## df\_pie\_all 데이터를 사용하여 ggplot객체를 생성  
p\_pie\_all <- df\_pie\_all |>  
 ggplot()  
  
## 앞서 그린 막대 그래프와 동일한 막대 그래프를 연도별로 분할하여 생성  
p\_pie\_all1 <- p\_pie\_all +  
 geom\_col(aes(x = 1, y = 입학생수, fill = 학교종류), position = 'fill') +  
 geom\_text(aes(x = 1, y = 비율, label = scales::percent(비율, accuracy = 0.1), fill = 학교종류), position = position\_stack(vjust = 0.5), size = 2) +  
 ## 연도별로 분할  
 facet\_wrap(~연도)  
  
p\_pie\_all1



실행결과 9-다중 파이 차트를 위한 다중 막대 그래프 생성

앞서 그린 다중 막대 그래프의 Y축을 극좌표계로 적용하는 코드는 다음과 같다.

## 앞서 생성한 막대 그래프를   
p\_pie\_all1 +  
 ## 극 좌표계로 Y축을 변환  
 coord\_polar(theta = "y", direction = -1) +  
 ## theme\_void()로 배경등을 모두 없애주고  
 theme\_void() +  
 ## 표 제목의 수평 정렬을 중간으로 설정  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 labs(title = '연도별 입학생 구성비 - 다중 파이 차트')



실행결과 9-다중 파이 차트

# 도넛 차트

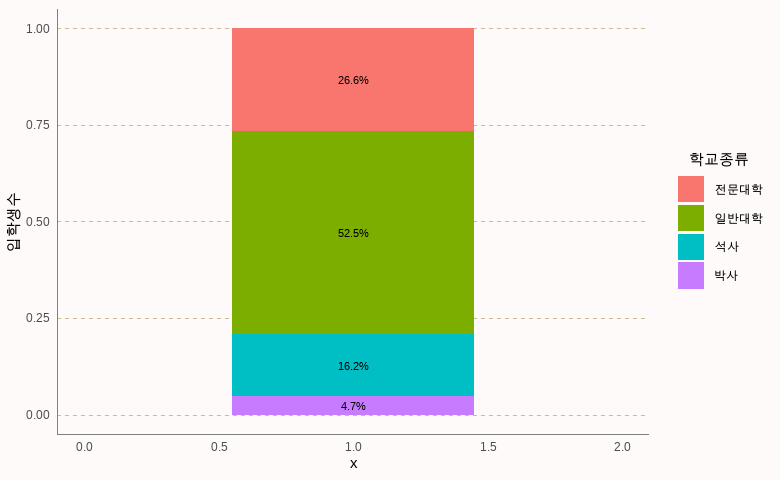
도넛 차트는 파이 차트와 유사한 시각화 방법이지만 파이 차트는 원을 사용하는데 반해 도넛 차트는 링의 형태로 표현한다. 도넛 차트도 파이 차트와 마찬가지 단점이 존재한다. 하지만 단 하나의 차이는 링의 중간에 공간이 발생하기 때문에 이 공간을 이용해 추가적인 데이터를 표현할 수 있다는 것이다. 표의 제목을 넣을수도 있고 요약 데이터 값을 넣을 수 있다.

도넛 차트는 단일 파이 차트와 같이 막대를 극좌표계롤 돌린다는 점에서 동일하다. 막대를 그리는 방법이 조금 다르다. 파이 차트에서는 막대 그래프를 그린 다음에 극좌표계를 사용해 둥글게 돌려주었다. 하지만 도넛 차트에서는 중간에 공간을 두어야 하기 때문에 막대와 X축의 0점간의 사이에 약간의 간격을 두어야 한다. 간격을 두려면 xlim()를 사용해 X축의 범위를 조절한다.

## 단일 도넛 차트

단일 도넛 차트는 도넛 차트 하나로 데이터를 시각화하는 방법을 말한다. geom\_bar()나 geom\_col()을 사용하여 막대 그래프 하나를 그리는데 xlim()을 이용해서 X축 원점과의 간격을 만든 후 이를 Y축을 중심으로 극좌표계로 만들면 생성된다. 차근차근 도넛 차트를 만들어 보자. 우선 하나의 막대를 가지는 막대 그래프를 그리고 xlim()을 사용해 X축과의 간격을 만드는 코드는 다음과 같다.

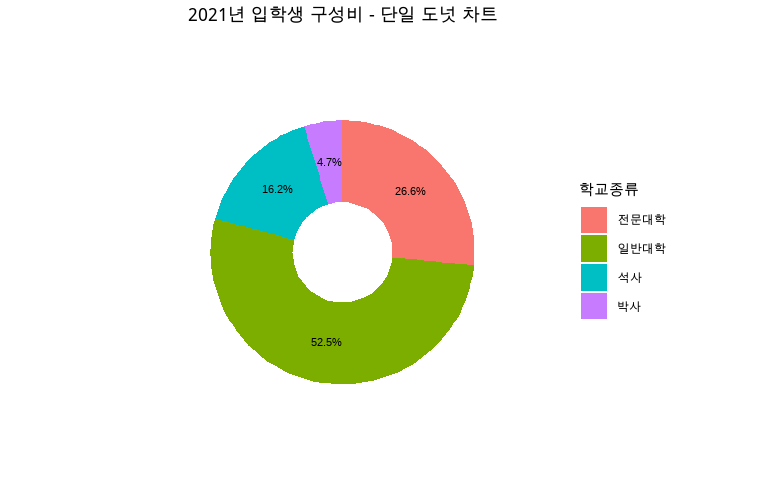
## p\_pie 객체에서  
p\_donut <- p\_pie +  
 ## X축을 1, Y축을 입학생수, fill을 학교종류로 매핑하고 비율로 표현하기 위해 position을 fill로 설정한 geom\_col 레이어를 생성  
 geom\_col(aes(x = 1, y = 입학생수, fill = 학교종류), position = 'fill') +  
 ## X축을 연도, Y축을 비율, label을 백분률로 만들되 정확성을 소수점 한자리, fill을 학교종류로 매핑하고 텍스트 위치를 설정하기 위해 position을 position\_stack()으로 수직정렬을 0.5로 설정하고 크기를 3으로 설정한 geom\_text 레이어 추가  
 geom\_text(aes(x = 1, y = 비율, label = scales::percent(비율, accuracy = 0.1), fill = 학교종류), position = position\_stack(vjust = 0.5), size = 3) +   
 ## X축을 0부터 2까지로 한정  
 xlim(0,2)  
  
p\_donut



실행결과 9-단일 도넛 차트를 위한 기본 막대 그래프

앞의 막대 그래프는 하나의 막대만 표현되었고 X축이 0부터 시작되는데 막대의 X축 값은 1이기 떄문에 다소 여유가 보인다. 이를 Y축을 기준으로 극과표계를 적용하는데 시계방향으로 적용하면 다음과 같다. 별로 시각화에 도움이 되지 않는 요소들은 theme\_void()로 제거한다.

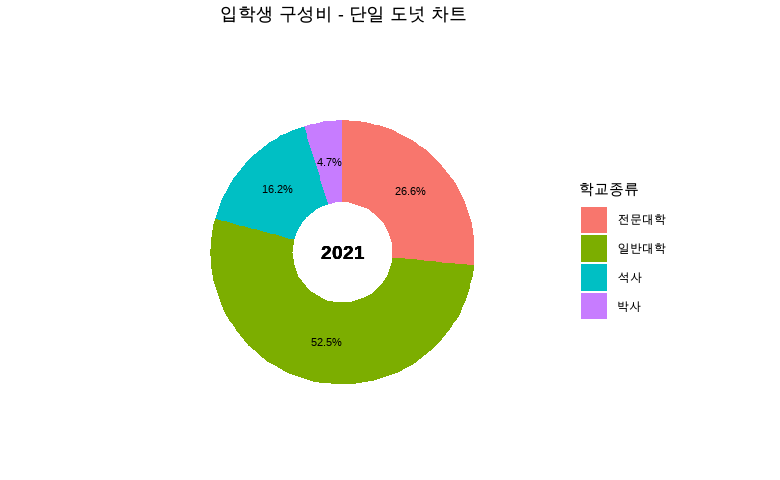
## p\_pie 객체에서  
p\_donut1 <- p\_donut +  
 ## 극 좌표계로 Y축을 변환하는데 데이터를 시계방향으로 설정  
 coord\_polar(theta = "y", direction = -1) +  
 theme\_void() +  
 ## 표 제목의 수평 정렬을 중간으로 설정  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 labs(title = '2021년 입학생 구성비 - 단일 도넛 차트')  
  
p\_donut1



실행결과 9-단일 도넛 차트

앞서 설명한 바와 같이 파이 차트와 다른 점을 부각하기 위해 가운데 빈 곳을 활용해보자 다음과 같이 geom\_text()를 사용하면 강조할 문구를 넣을 수 있다.

p\_donut2 <- p\_donut1 +  
 ## label이 연도, X, Y가 0으로 매핑되고 size가 5, fontface는 굵게 설정한 geom\_text 레이어 추가  
 geom\_text(aes(label = 연도, x = 0, y = 0), size = 5, fontface='bold') +  
 labs(title = '입학생 구성비 - 단일 도넛 차트')  
  
p\_donut2

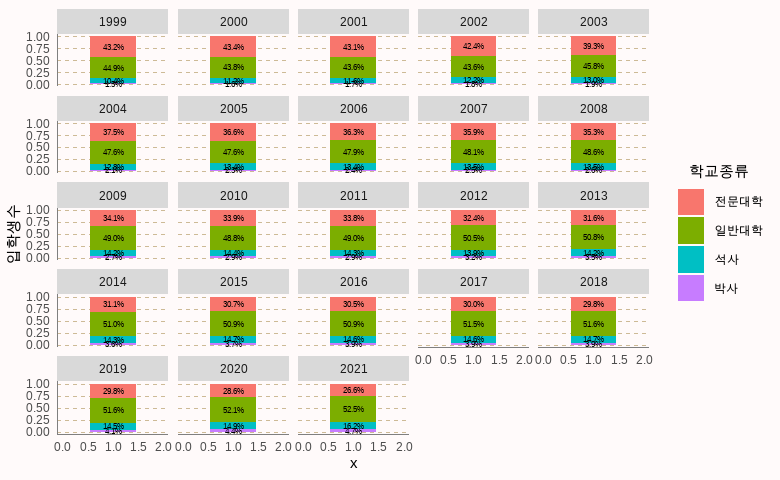


실행결과 9-주석이 추가된 단일 도넛 차트

## 다중 도넛 차트

다중 도넛 차트는 다중 파이 차트와 동일한 방식으로 생성된다. 각각의 그룹으로 막대 그래프를 분할하여 그리고 이를 Y축을 중심으로 극좌표계를 적용한다. 여기에 앞에서 추가했던 연도 주석을 가운데에 추가하는 코드는 다음과 같다 .

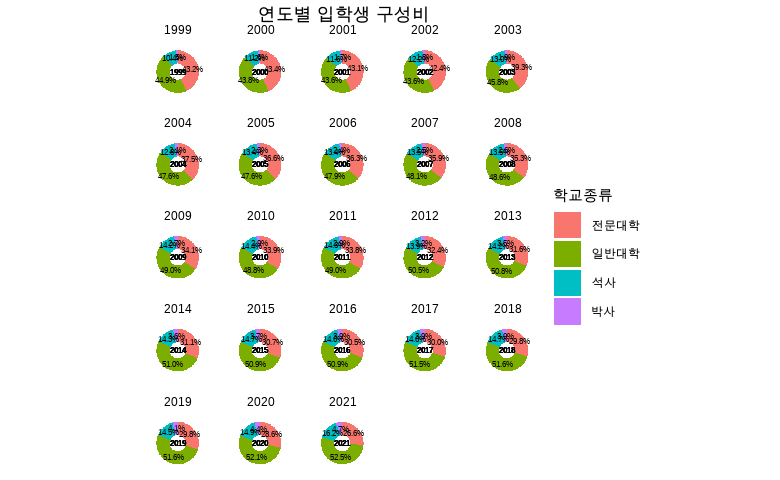
## df\_입학자\_long 데이터에서  
df\_donut\_all <- df\_입학자\_long |>  
 ## 일부 데이터만 필터링  
 filter(학교종류 %in% c('전문대학', '일반대학', '석사', '박사'), 지역 == '전체') |>  
 ## 표시 순서를 설정  
 mutate(학교종류 = fct\_relevel(학교종류, '전문대학', '일반대학', '석사', '박사')) |>   
 ## 연도별로 그룹화  
 group\_by(연도) |>  
 ## 각 연도별로 학교 종류별 비율을 계산  
 mutate(비율 = 입학생수 / sum(입학생수)) |>  
 ungroup()  
  
p\_donut\_all <- df\_donut\_all |>  
 ggplot()  
  
p\_donut\_all1 <- p\_donut\_all +  
 ## X축을 1, Y축을 입학생수, fill을 학교종류로 매핑하고 비율로 표현하기 위해 position을 fill로 설정한 geom\_col 레이어를 생성  
 geom\_col(aes(x = 1, y = 입학생수, fill = 학교종류), position = 'fill') +  
 ## X축을 1, Y축을 비율, label을 비율을 소수점 첫째자리의 정확성을 가진 퍼센트 형식, fill을 학교종류로 매핑하고 position을 position\_stack()으로 위치, size를 2로 설정한 geom\_text() 레이어 생성  
 geom\_text(aes(x = 1, y = 비율, label = scales::percent(비율, accuracy = 0.1), fill = 학교종류), position = position\_stack(vjust = 0.5), size = 2) +   
 ## X축을 0부터 2까지로 한정  
 xlim(0,2) +  
 facet\_wrap(~연도)  
  
p\_donut\_all1



실행결과 9-다중 도넛 차트를 위한 분할 막대 그래프

앞서 그린 분할 막대 그래프를 도넛 차트로 만들기 위해 Y축을 기준으로 극좌표계를 적용하고 시각화에 방해되는 요소들은 theme\_void()로 제거한다. 그 가운데에 각각의 연도를 표현하기 위해 geom\_text() 레이어를 추가한다 .이를 구현하는 코드는 다음과 같다.

p\_donut\_all1 +  
 ## Y축을 중심으로 시계방향 극좌표계 적용  
 coord\_polar(theta = "y", direction = -1) +  
 ## label을 연도, X축을 0, Y축을 0에 매핑한 geom\_text 레이어 추가  
 geom\_text(aes(label = 연도, x = 0, y = 0), size = 2, fontface='bold') +  
 ## 테마 요소 제거  
 theme\_void() +  
 ## 전체 제목을 중간으로 설정  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 labs(title = '연도별 입학생 구성비')



실행결과 9-주석이 추가된 다중 도넛 차트

# 트리맵

대부분의 보고문서들은 네모난 종이위에 표현되거나 네모난 화면위에 표현된다. 앞서 기술한 파이 차트나 도넛 차트는 원형으로 표현되기 떄문에 네모난 종이나 화면에 표현하면 시각화에 사용되지 못하는 공간이 많아져서 전체 영역을 호과적으로 사용하지 못한다. 이러한 단점을 극복하기 위해 네모난 사각형을 사용하여 전체의 비율을 표현한 시각화 방법이 트리맵이다. 트리맵은 treemapify 패키지나 ggplotify 패키지를 사용하여 생성할 수 있다. 여기서는 treemapify 패키지를 사용하는 방법에 대해 설명하겠다. 그러려면 먼저 treemapity 패키지를 설치하여야 한다.

## treemapify 패키지 설치   
if(!require(treemapify)) {  
 install.packages('treemapify')  
 library(treemapify)  
}

## 단일 트리맵

단일 트리맵은 앞서 설명했던 단일 파이 차트나 단일 도넛 차트와 같이 단일 그룹에 대한 비율을 산출하기 위해 사용되는 트리맵이다. 트리맵을 산출하기 위해서는 treemapify 패키지의 geom\_treemap()을 사용한다.

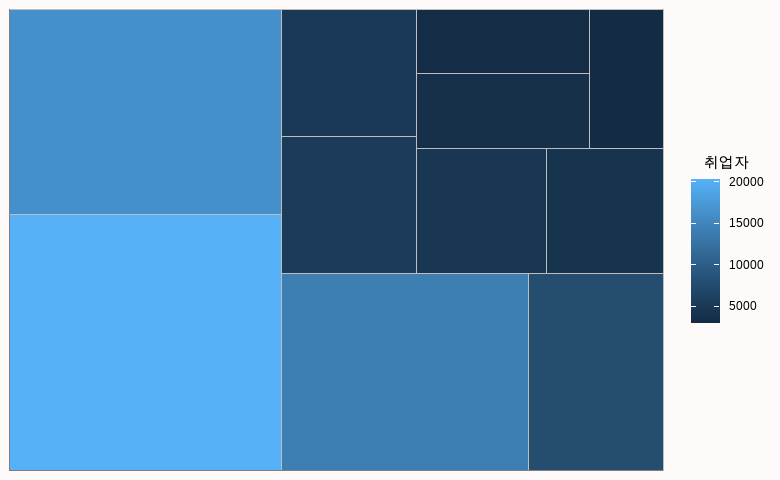
geom\_treemap(mapping = NULL, data = NULL, stat = "identity", position = "identity", na.rm = FALSE, show.legend = NA, inherit.aes = TRUE, layout = "squarified", start = "bottomleft", radius = grid::unit(0, "pt"),...)  
 - mapping, data, stat, position, na.rm, show.legend, inherit.aes, ... : geom\_rect()와 동일한 설정  
 - layout : 트리맵의 레이아웃을 설정, 'squarified', 'scol', 'srow','fixed' 중에 하나 설정  
 - start : 트리맵의 시작위치 설정  
 - radius : 사각형 귀퉁이를 둥글게 설정

geom\_treemap()에서 사용할 수 있는 미적요소는 area, alpha, color, fill, linetype, subgroup, subgroup2, subgroup3이다. 이중 area는 필수적 미적 요소로써 사각형의 크기를 결정하는데 사용할 변수를 매핑하는 미적요소이고 subgroup, subgroup2, subgroup3는 다중 트리맵을 설정하기 위한 세부 그룹을 매핑하는 미적 요소이다.

공학계열의 중계열에 해당하는 취업자의 분포를 트리맵으로 시각화하는 코드는 다음과 같다.

## df\_취업통계 데이터에서  
df\_treemap <- df\_취업통계 |>  
 ## 공학계열만 필터링  
 filter(대계열 == '공학계열') |>   
 ## 중계열 별로 그룹화  
 group\_by(중계열) |>  
 ## 취업자 합계를 산출  
 summarise(취업자 = sum(취업자\_합계\_계))

p\_treemap <- df\_treemap |>   
 ggplot() +  
 ## area, fill을 취업자, label을 중계열와 콤마형 취업자를 붙여 매핑한 geom\_treemap 레이어 생성  
 geom\_treemap(aes(area = 취업자, fill = 취업자, label = paste0(중계열, "\n", scales::comma(취업자))))  
  
p\_treemap



실행결과 9-기본 트리맵

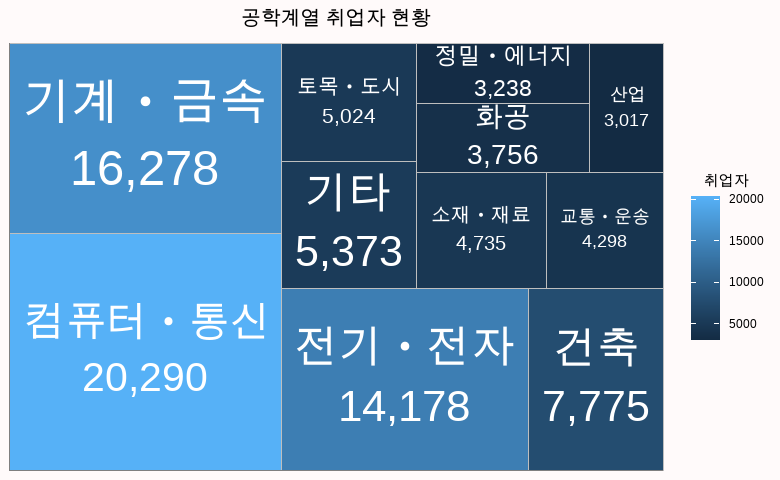
앞의 트리맵은 전체에서의 크기를 사각형으로만 표현했고 그 사각형이 무엇을 표현하는지 크기가 얼마나되는지가 전혀 나타나지 않아 데이터를 인식할 수 없다. 이 트리맵에 해당 사각형의 변량과 크기에 해당하는 전체의 비율을 표시하기 위해서는 geom\_treemap\_text()를 사용한다.

geom\_treemap\_text(mapping = NULL, data = NULL, stat = "identity", position = "identity", na.rm = FALSE, show.legend = FALSE, inherit.aes = TRUE, padding.x = grid::unit(1, "mm"), padding.y = grid::unit(1, "mm"), place = "topleft", min.size = 4, grow = FALSE, reflow = FALSE, layout = "squarified", start = "bottomleft", ...)  
 - mapping, data, stat, position, na.rm, show.legend, inherit.aes, ... : geom\_text와 동일한 설정   
 - padding.x, padding.y : 수평, 수직 방향으로 문자와 경계선과의 거리 설정  
 - place : 문자가 사각형 안에서 위치하는 장소  
 - min.size : 문자의 최소 크기  
 - grow : 문자의 크기를 사각형의 크기에 따라 조절할 지를 결정하는 논리값  
 - reflow : 문자를 사각형에 더 잘 맞춰줄지를 결정하는 논리값

geom\_treemap\_text()에서 매핑이 가능한 미적요소는 area, label, subgroup, (subgroup2 또는 subgroup3), color, size, alpha, family, fontface, angle 등이 있다. 이중 area와 label은 필수 미적요소이다.

앞서 그린 트리맵에 변량과 취업자수를 표기하는 코드는 다음과 같다.

p\_treemap +  
 ## area는 취업자수, label은 중계열과 취업자수를 매핑  
 geom\_treemap\_text(aes(area = 취업자, label = paste0(중계열, "\n", scales::comma(취업자))),   
 ## 문자색은 하얀색   
 colour = "white",  
 ## 위치는 중간  
 place = "centre",  
 ## 사각형의 크기에 따라 문자 크기 조절  
 grow =TRUE,   
 ## 가로방향으로 경계선과 문자 간격을 3mm 설정  
 padding.x = grid::unit(3, "mm")) +  
 labs(title = '공학계열 취업자 현황') +  
 theme(plot.title = element\_text(hjust = 0.5, size = 15))



실행결과 9-변량이 포기된 트리맵

## 다중 트리맵

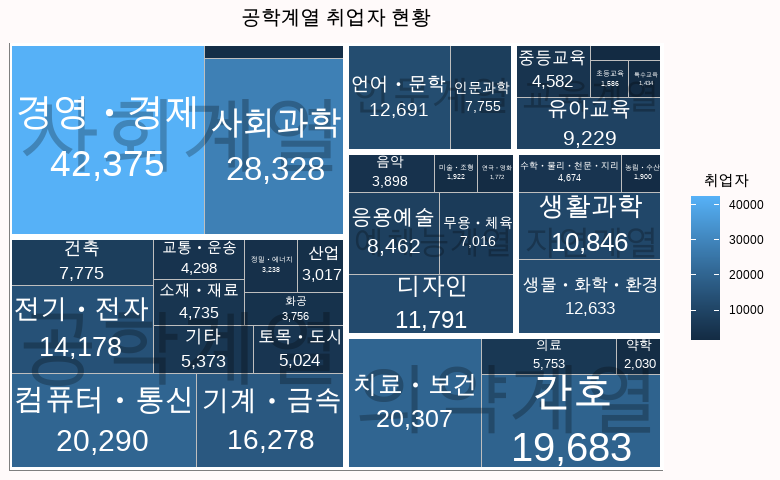
다중 파이 차트나 다중 도넛 차트는 facet\_\*()을 사용하여 그룹 변수를 사용해 차트를 분할함으로서 다중 차트를 만들었다. 하지만 트리맵은 facet\_\*()을 사용할 수 없다. 대신 treemapify 패키지에서는 geom\_treemap()의 subgroup, subgroup2, subgroup3 미적요소에 변수를 매핑함으로써 다중 트리맵을 생성할 수 있고 서브 그룹의 경계선 설정은 geom\_tree\_subgroup\_border(), 서브그룹의 문자 설정은 geom\_tree\_subgroup\_text()를 사용한다.

geom\_tree\_subgroup\_border(mapping = NULL, data = NULL, stat = "identity", position = "identity", na.rm = FALSE, show.legend = NA, inherit.aes = TRUE, layout = "squarified", start = "bottomleft", level = "subgroup", ...)  
 - level : 어떤 서브그룹에 대한 설정일지를 설정, 'subgroup', 'subgroup2', 'subgroup3' 중에 하나 설정  
  
geom\_treemap\_subgroup\_text(mapping = NULL, data = NULL, stat = "identity", position = "identity", na.rm = FALSE, show.legend = FALSE, inherit.aes = TRUE, padding.x = grid::unit(1, "mm"), padding.y = grid::unit(1, "mm"), place = "bottom", min.size = 4, grow = FALSE, reflow = FALSE, layout = "squarified", start = "bottomleft", level = "subgroup", ...)  
 - mapping, data, stat, position, na.rm, show.legend, inherit.aes, ... : geom\_text와 동일한 설정   
 - padding.x, padding.y : 수평, 수직 방향으로 문자와 경계선과의 거리 설정  
 - place : 문자가 사각형 안에서 위치하는 장소  
 - min.size : 문자의 최소 크기  
 - grow : 문자의 크기를 사각형의 크기에 따라 조절할 지를 결정하는 논리값  
 - reflow : 문자를 사각형에 더 잘 맞춰줄지를 결정하는 논리값

geom\_tree\_subgroup\_border()에서 매핑이 가능한 미적요소는 area, subgroup, (subgroup2 or subgroup3), color, size, linetype, alpha 등이고 geom\_tree\_subgroup\_text()에서 매핑이 가능한 미적요소는 area, subgroup(subgroup2 or subgroup3), colour, size, alpha, family, fontface, angle 등이다. 이 중 area와 서브그룹 설정(subgroup, subgroup2, subgroup3)은 필수 미적요소이다.

다음은 서브 그룹을 대계열과 중계열로 설정한 다중 트리맵을 생성하는 코드이다.

df\_treemap\_multi <- df\_취업통계 |>   
 ## 대계열과 중계열로 그룹화  
 group\_by(대계열, 중계열) |>  
 ## 취업자\_합계\_계 열을 합계로 요약  
 summarise(취업자 = sum(취업자\_합계\_계))  
  
df\_treemap\_multi |>   
 ggplot() +  
 ## area, fill을 취업자, label을 중계열명과 취업자, subgroup을 대계열로 매핑한 geom\_treemap 레이어 생성  
 geom\_treemap(aes(area = 취업자, fill = 취업자, label = paste0(중계열, "\n", scales::comma(취업자)), subgroup = 대계열)) +   
 ## area를 취업자, label을 중계열명과 취업자, subgroup을 대계열로 매핑한 geom\_treemap\_border 레이어 생성  
 geom\_treemap\_subgroup\_border(aes(area = 취업자, label = paste0(중계열, "\n", scales::comma(취업자)), subgroup = 대계열), colour = "white", size = 5) +  
 ## area를 취업자, label을 중계열명과 취업자, subgroup을 대계열로 매핑한 geom\_treemap\_text 레이어 생성  
 geom\_treemap\_subgroup\_text(aes(area = 취업자, label = paste0(중계열, "\n", scales::comma(취업자)), subgroup = 대계열), place = "centre", grow = TRUE, alpha = 0.25, colour = "black") +  
 ## area를 취업자, label을 중계열명과 취업자, subgroup을 대계열로 매핑한 geom\_treemap\_text 레이어 생성  
 geom\_treemap\_text(aes(area = 취업자, label = paste0(중계열, "\n", scales::comma(취업자)), subgroup = 대계열), colour = "white", place = "centre",  
 size = 10, grow = TRUE) +  
 labs(title = '공학계열 취업자 현황') +  
 theme(plot.title = element\_text(hjust = 0.5, size = 15))



실행결과 9-변량이 표기된 다중 트리맵

# 와플 차트(Waffle Chart)

와플은 아마도 한번 정도는 먹어본 음식일 것이다. 아이스크림이나 달콤한 시럽이 얹혀진 이 파이인듯, 빵인듯, 케이크인 둣한 음식의 표면은 작은 네모들로 가득하다. 여기에서 영감을 얻은 것같은 와플 차트는 작은 사각형으로 전체 영역을 나누고 각 변량이 차지하는 비율만큼 사각형을 각 변량에 포함시킴으로써 데이터를 시각화한다.

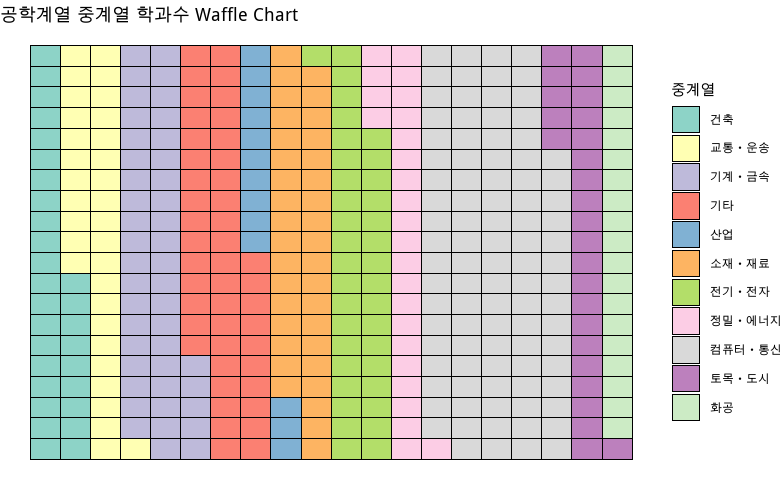
이를 위해서는 다른 시각화에는 수행하지 않았던 전처리 작업이 필요하다. 먼저 와플 차트에서 사용할 사각형의 갯수를 결정해야 하고 전체 비율을 이 사각형의 갯수로 변환하는 작업이다. 예를 들어 전체 사각형이 100개인 와플 차트라면 비율을 소수점 한자리에서 반올림해 소수점을 없애고 각각의 백분률만큼 사각형을 배치하면 되겠지만 150개인 와플차트라면 각각의 비율에 1.5를 곱해야 할 것이다.

ggplot2에서는 와플 차트를 위한 전용 함수를 제공하지 않는다. 대신 geom\_tile()을 사용해서 만들 수 있는데 어쩌면 geom\_tile()이 와플 차트를 위한 함수로 특화된 것은 아닌지 생각도 든다.

geom\_tile(mapping = NULL, data = NULL, stat = "identity", position = "identity", ..., linejoin = "mitre", na.rm = FALSE, show.legend = NA, inherit.aes = TRUE)  
 - mapping : aes()를 사용하여 매핑할 미적요소, 생략되면 ggplot()에 정의된 미적매핑 사용  
 - data : 시각화를 위해 사용될 데이터, 생략되면 ggplot()에 정의된 데이터 사용  
 - stat : 시각화에 적용될 통계요소, 기본값은 'identity'  
 - position : 시각화에 적용될 위치요소, 기본값은 'identity'  
 - ... : 미적요소의 설정  
 - linejoin : 사각형의 꼭지점의 처리 방법 설정(round, mitre, bevel)  
 - na.rm : NA 값을 생략할 것인지를 설정하는 논리값  
 - show.legend : 범례를 사용할 것인지를 설정하는 논리값  
 - inherit.aes : ggplot()에서 설정한 매핑값을 상속받을지 결정하는 논리값

geom\_tile()의 매개변수는 대부분 다른 기하요소의 설명에 사용되었던 매개변수이기 때문에 특별히 설명하지 않았다. 또 매핑이 가능한 미적요소도 다른 기하요소들과 유사한데 x, y, alpha, color, fill, group, height, linetype, size, width 등이 있다.

vec\_waffle <- df\_취업통계 |>   
 filter(대계열 == '공학계열') |>   
 ## 와플 차트에 사용할 변량 하나를 선택  
 select(중계열) |>   
 ## 필터링한 데이터를 벡터로 변환  
 pull() # the categorical data   
  
## 와플 차트의 사각형 행수 설정  
nrows <- 20  
  
## expand.grid는 모든 조합에 대한 데이터 프레임을 만드는 함수로 X, Y 두 변수 20개씩 총 400개의 조합을 가진 데이터 프레임 생성  
df\_waffle <- expand.grid(y = 1:nrows, x = 1:nrows)  
  
## 중계열로 생성된 벡터들의 비율 데이터프레임 생성  
## table(vec\_waffle)로 vec\_waffle의 각각의 데이터에 대한 빈도 산출  
## nrows\*nrows는 전체 사각형의 갯수, length(vec\_waffle)은 vec\_waffle의 데이터 갯수로 여기서는 nrows\*nrows는 400, vec\_waffle 2589로 사각형 하나당 0.15의 비중을 가짐  
freq\_table <- round(table(vec\_waffle) \* ((nrows\*nrows)/(length(vec\_waffle))))  
  
## freq\_table에서 산출된 중계열별 빈도수만큼 반복한 행을 만듦  
df\_waffle$중계열 <- factor(rep(names(freq\_table), freq\_table))  
  
## X축을 x, Y축을 y, fill을 category로 매핑한 ggplot 객체 생성  
df\_waffle |> ggplot(aes(x = x, y = y, fill = 중계열)) +   
 ## color를 black, size를 0.5로 설정  
 geom\_tile(color = "black", size = 0.5) +  
 ## 색 팔레트를 'Set3'로 설정  
 scale\_fill\_brewer(palette = "Set3") +  
 ## 모든 미적요소를 제거  
 theme\_void() +  
 labs(title="공학계열 중계열 학과수 Waffle Chart", fill = '중계열')



1. Hadley Wickham이 제안한 방법은 필자의 블로그를 참조하라 [↑](#footnote-ref-20)
2. ggplot 객체에서 미적요소를 제거하는 방법도 있다. 이 방법은 필자의 블로그를 참조하라. [↑](#footnote-ref-21)