1. 구성(Composition)의 시각화

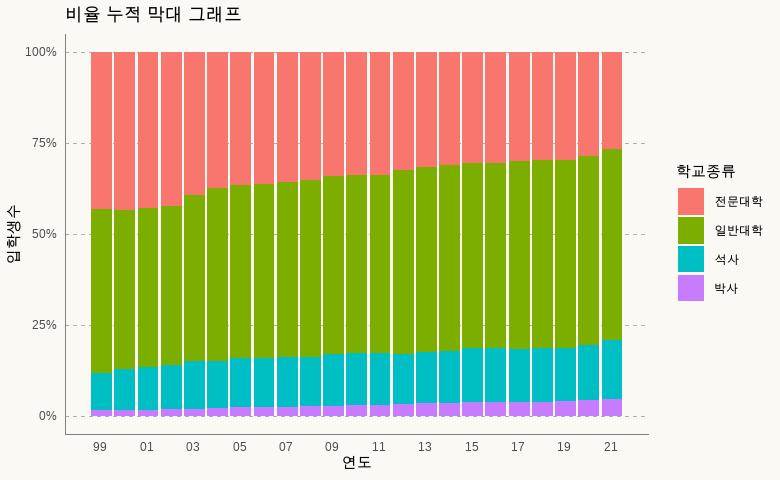
구성의 시각화는 특정 데이터를 구성하는 각 부분의 비율을 시각화하는 것이다. 보통의 경우 비율이나 백분율을 통해 시각화하며 막대나 원을 세분화하여 그 비율을 보여준다. 여러개의 세부 범주로 나누어진 많은 범주들의 구성 비율을 나타내는데 비율 누적 막대 그래프가 많이 사용된다. 반면 단 하나의 범주에 대한 세부 범주의 구성 비율을 나타낼 때는 누적 막대 그래프 보다는 동그란 파이 차트나 도넛 차트로 나타낼 수 있다. 하지만 시각화 전문가들은 파이 차트나 도넛 차트는 권장하지 않는다. 사람의 시각으로는 파이 차트나 도넛 차트의 그 비율을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



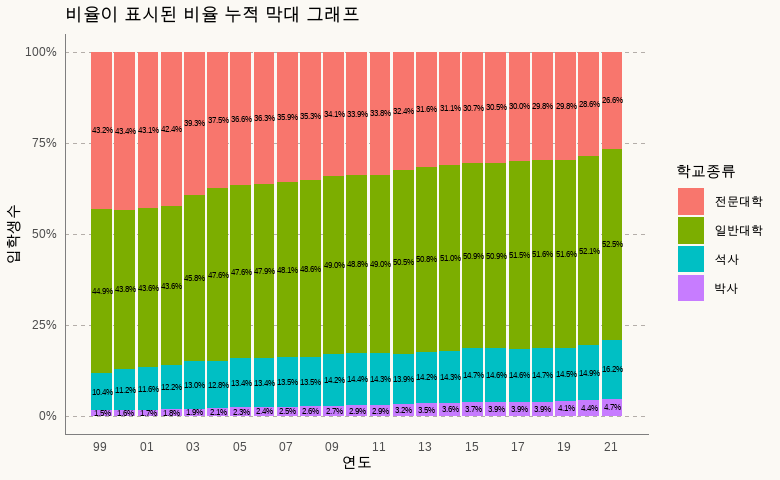
실행결과 9- . 비율 누적 막대 그래프

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 그루핑하고 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.

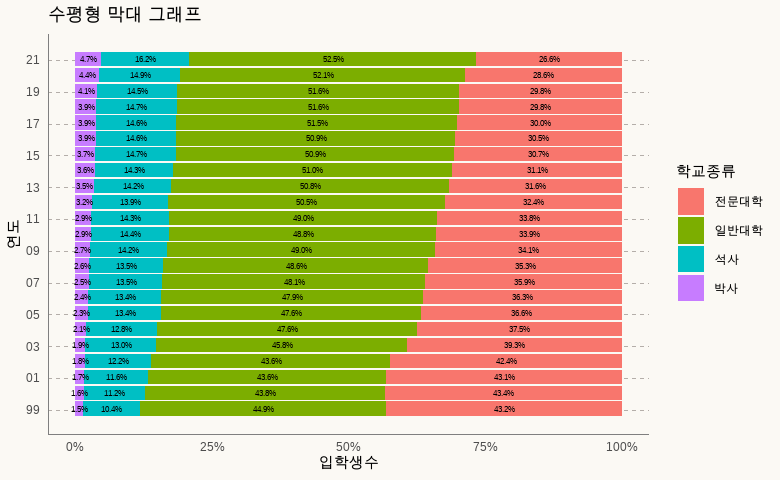
p\_col\_fill2 <- p\_col\_fill1 +   
 ## geom\_text()에 비율이 저장된 df\_col\_fill\_rate를 사용  
 geom\_text(data = df\_col\_fill\_rate,  
 ## X축은 연도, Y축은 비율, label은 비율을 퍼센트 형식, fill은 학교종류로 매핑  
 aes(x = 연도, y = 비율, label = scales::percent(비율, accuracy = 0.1), fill = 학교종류),   
 ## position은 position\_stack()을 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.

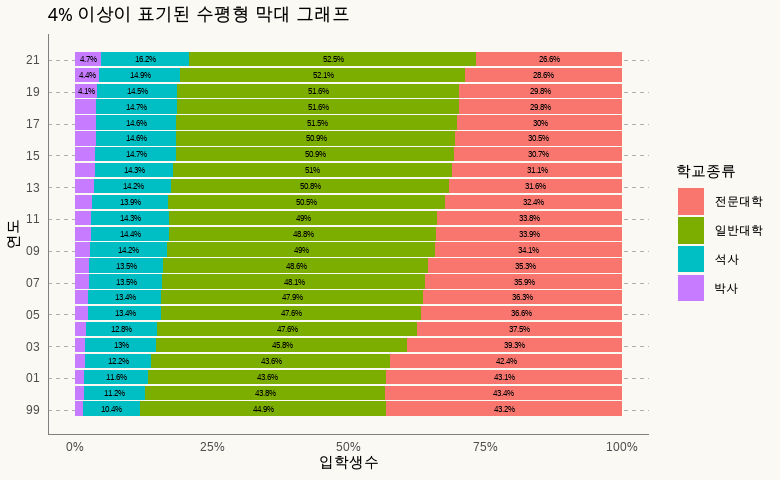
p\_col\_fill3 <- p\_col\_fill2 +   
 coord\_flip() +   
 labs(title = '수평형 막대 그래프')  
  
p\_col\_fill3



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



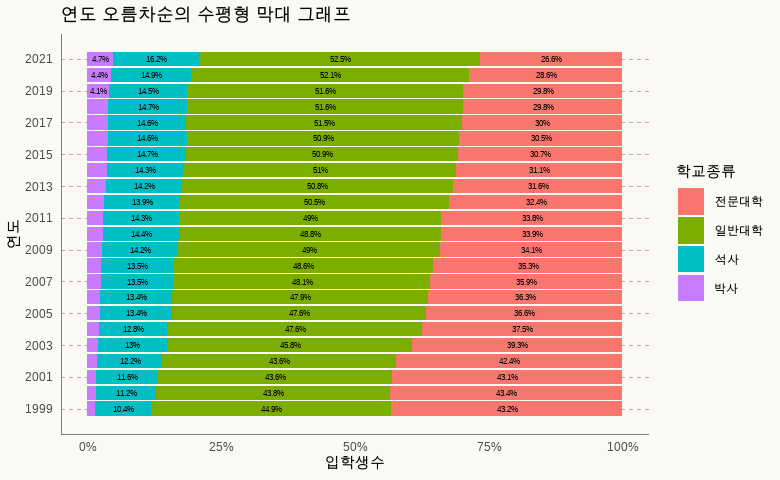
실행결과 9- . 4% 이상이 표기된 수평형 막대 그래프

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)

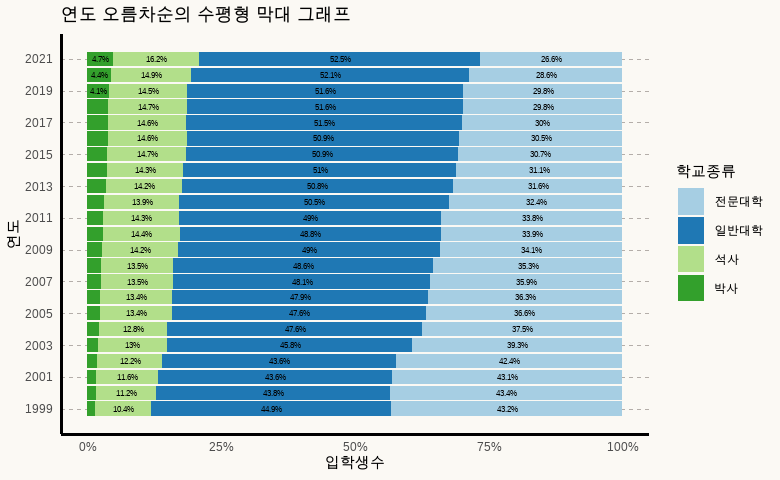
먼저 연도 정렬에 사용하기 위해 날짜형 변수에서 연도를 추출한 변수를 만들어준다. 이후 이 변수를 X축에 매핑한 후 X축의 스케일을 연속형으로 설정(scale\_x\_continuous())하고 2021부터 1999까지 -2마다 하나씩 눈금(breaks = seq(from = 1999, to = 2021, by = 2))을 만들어준다.

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

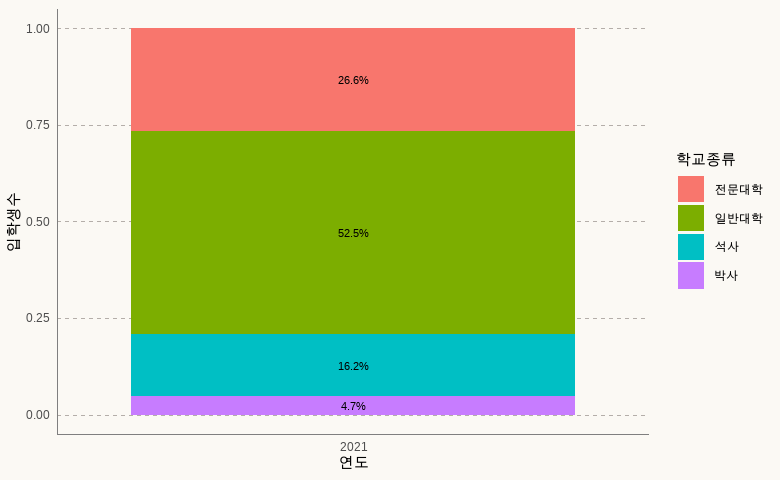
p\_col\_fill4 +  
 theme(axis.line = element\_line(size=1, colour = "black"),  
 panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(), panel.background = element\_blank()) + scale\_fill\_brewer(palette="Paired")



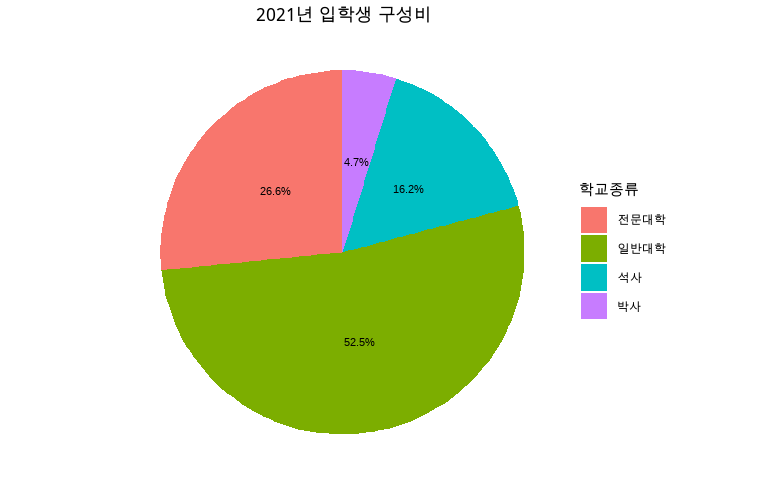
# 파이 차트

## 단일 파이 차트

df\_pie <- df\_입학자\_long |>  
 filter(학교종류 %in% c('전문대학', '일반대학', '석사', '박사'), 지역 == '전체', 연도 == 2021) |>  
 mutate(학교종류 = fct\_relevel(학교종류, '전문대학', '일반대학', '석사', '박사')) |>  
 mutate(비율 = 입학생수 / sum(입학생수))  
  
p\_pie <- df\_pie |>  
 ggplot()  
  
  
p\_pie1 <- p\_pie +  
 geom\_col(aes(x = 연도, y = 입학생수, fill = 학교종류), position = 'fill') +  
 geom\_text(aes(x = 연도, y = 비율, label = scales::percent(비율, accuracy = 0.1), fill = 학교종류), position = position\_stack(vjust = 0.5), size = 3)  
  
p\_pie1

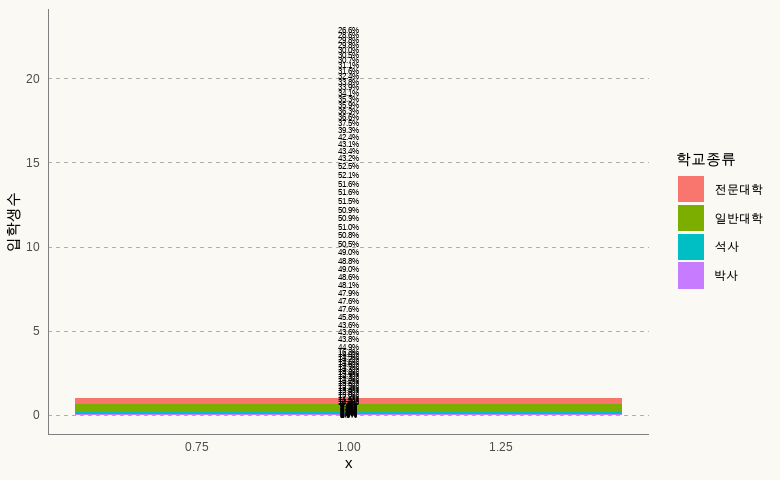


p\_pie1 +  
 coord\_polar(theta = "y") +  
 theme\_void() +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 labs(title = '2021년 입학생 구성비')

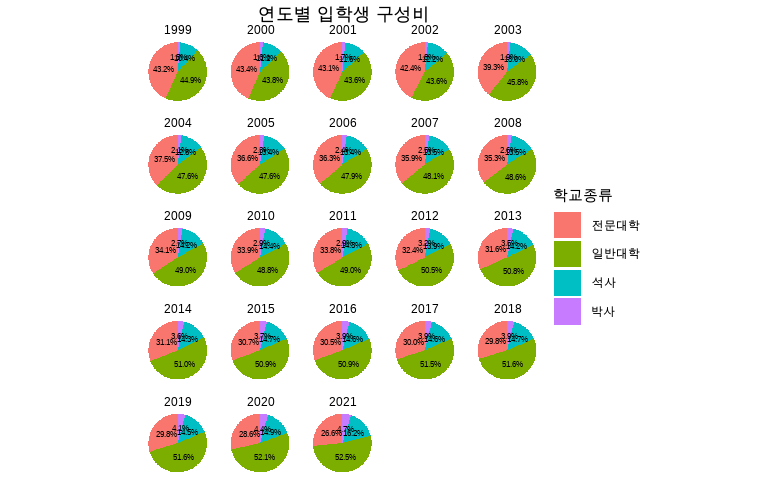


## 분할 파이 차트

df\_pie\_all <- 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)) |>  
 ungroup()  
  
  
p\_pie\_all <- df\_pie\_all |>  
 ggplot()  
  
  
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)  
  
p\_pie\_all1



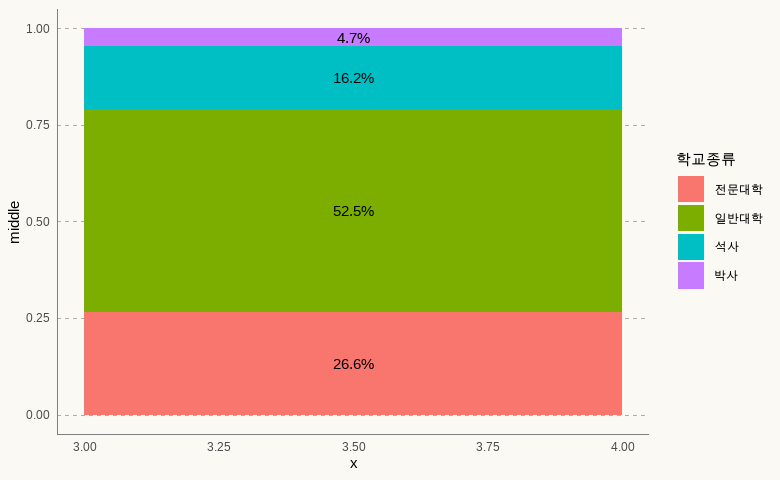
p\_pie\_all1 +  
 facet\_wrap(~연도) +   
 coord\_polar(theta = "y") +  
 theme\_void() +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 labs(title = '연도별 입학생 구성비')



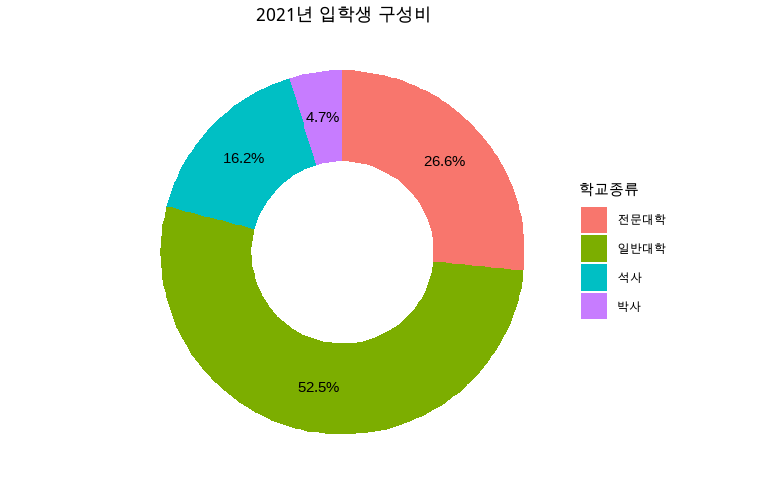
# 도넛 차트

## 단일 도넛 차트

df\_donut <- df\_입학자\_long |>  
 filter(학교종류 %in% c('전문대학', '일반대학', '석사', '박사'), 지역 == '전체', 연도 == 2021) |>  
 mutate(학교종류 = fct\_relevel(학교종류, '전문대학', '일반대학', '석사', '박사')) |>  
 mutate(비율 = round(입학생수 / sum(입학생수), 3))  
  
  
df\_donut$ymax = cumsum(df\_donut$비율)  
  
df\_donut$ymin = lag(df\_donut$ymax, 1, default = 0)  
  
df\_donut$middle = (df\_donut$ymin + df\_donut$ymax) / 2   
  
p\_donut <- df\_donut |>  
 ggplot()  
  
p\_donut1 <- p\_donut +  
 geom\_rect(aes(xmin = 3, xmax = 4, ymin = ymin, ymax = ymax, fill = 학교종류)) +   
 geom\_text(aes(x = 3.5, y = middle, label = scales::percent(비율, accuracy = 0.1)))  
  
p\_donut1

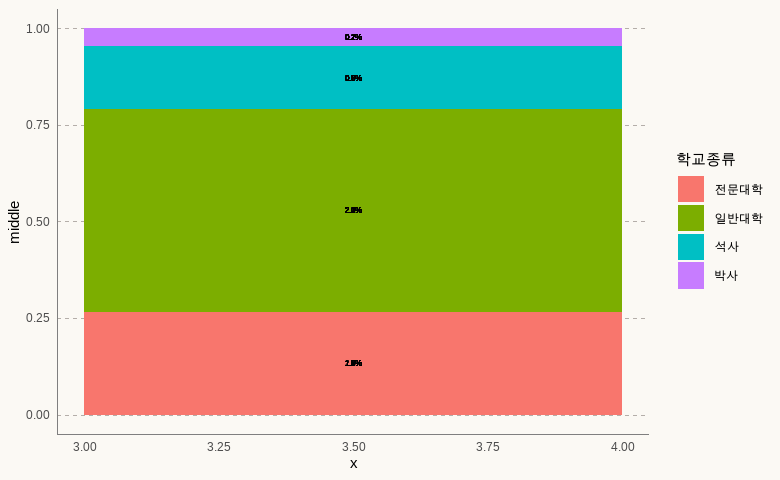


p\_donut2 <- p\_donut1 +  
 coord\_polar(theta = "y") +  
 lims(x = c(2, 4)) +  
 theme\_void() +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 labs(title = '2021년 입학생 구성비')  
  
p\_donut2

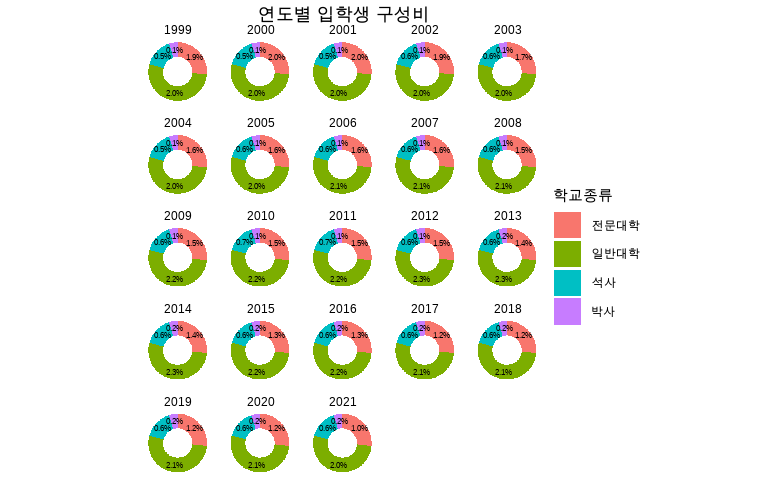


## 분할 도넛 차트

df\_donut\_all <- 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), ymax = cumsum(df\_donut$비율), ymin = lag(df\_donut$ymax, 1, default = 0), middle = (df\_donut$ymin + df\_donut$ymax) / 2) |>  
 ungroup() |>  
 mutate(비율 = round(입학생수 / sum(입학생수), 3))  
  
p\_donut\_all <- df\_donut\_all |>  
 ggplot()  
  
p\_donut\_all1 <- p\_donut\_all +  
 geom\_rect(aes(xmin = 3, xmax = 4, ymin = ymin, ymax = ymax, fill = 학교종류)) +   
 geom\_text(aes(x = 3.5, y = middle, label = scales::percent(비율, accuracy = 0.1)), size = 2)  
  
  
p\_donut\_all1



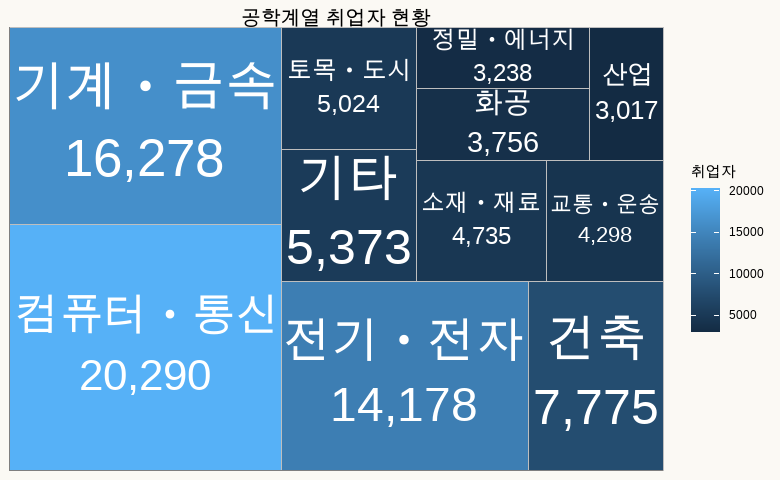
p\_donut\_all1 +  
 facet\_wrap(~연도) +   
 lims(x = c(2, 4)) +  
 coord\_polar(theta = "y") +  
 theme\_void() +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 labs(title = '연도별 입학생 구성비')



# 트리맵

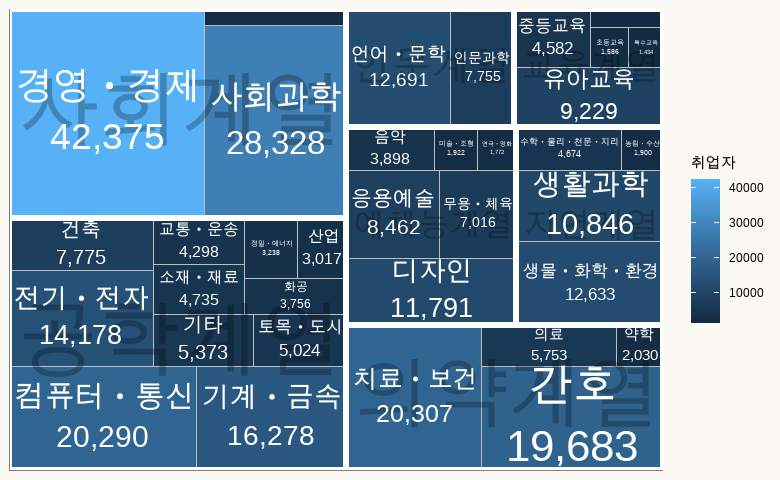
## 단일 그룹 트리맵

if(!require(treemapify)) {  
 install.packages('treemapify')  
 library(treemapify)  
}  
  
df\_treemap <- df\_취업통계 |> filter(대계열 == '공학계열') |> group\_by(중계열) |>  
 summarise(취업자 = sum(취업자\_합계\_계))  
  
df\_treemap |> ggplot(aes(area = 취업자, fill = 취업자, label = paste0(중계열, "\n", scales::comma(취업자)))) +  
 geom\_treemap() +  
 geom\_treemap\_text(colour = "white",  
 place = "centre",  
 size = 10, grow =TRUE) +  
 labs(title = '공학계열 취업자 현황') +  
 theme(plot.title = element\_text(hjust = 0.5, size = 15))



## 다중 그룹 트리맵

df\_treemap\_multi <- df\_취업통계 |> group\_by(대계열, 중계열) |>  
 summarise(취업자 = sum(취업자\_합계\_계))  
  
df\_treemap\_multi |> ggplot(aes(area = 취업자, fill = 취업자, label = paste0(중계열, "\n", scales::comma(취업자)), subgroup = 대계열)) +  
 geom\_treemap() +   
 geom\_treemap\_subgroup\_border(colour = "white", size = 5) +  
 geom\_treemap\_subgroup\_text(place = "centre", grow = TRUE,  
 alpha = 0.25, colour = "black") +  
 geom\_treemap\_text(colour = "white", place = "centre",  
 size = 10, grow = TRUE)

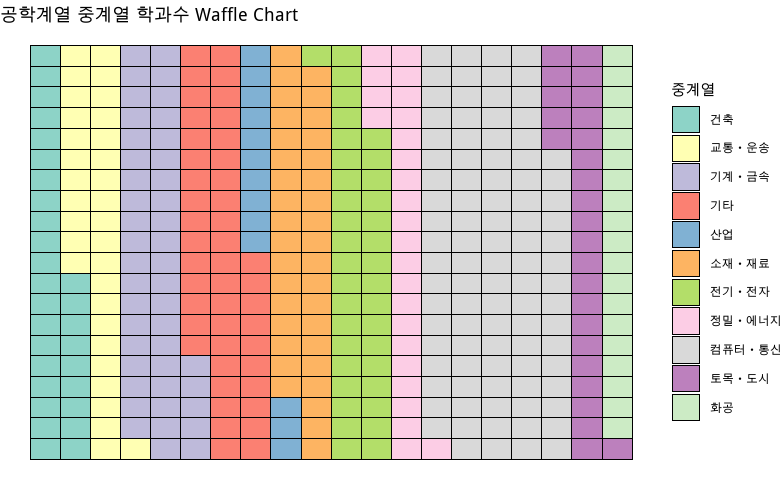


labs(title = '공학계열 취업자 현황') +  
 theme(plot.title = element\_text(hjust = 0.5, size = 15))

## NULL

# 와플차트

var <- df\_취업통계 |> filter(대계열 == '공학계열') |> select(중계열) |> pull() # the categorical data   
  
nrows <- 20  
  
df\_waffle <- expand.grid(y = 1:nrows, x = 1:nrows)  
  
categ\_table <- round(table(var) \* ((nrows\*nrows)/(length(var))))  
  
df\_waffle$category <- factor(rep(names(categ\_table), categ\_table))  
  
ggplot(df\_waffle, aes(x = x, y = y, fill = category)) +   
 geom\_tile(color = "black", size = 0.5) +  
 theme\_void() +  
 labs(title="공학계열 중계열 학과수 Waffle Chart", fill = '중계열') +  
 scale\_fill\_brewer(palette = "Set3")



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)