1. 실전 시각화

이번 장에서는 지금까지 살펴본 시각화 방법을 사용하여 실전에사 사용할 수 있는 시각화 응용 사례를 살펴보도록 하겠다. 앞의 실습에서도 ggplot2이외에 추가적인 패키지를 사용했지만 이번 장에서도 ggplot2를 확장 지원하는 몇가지 패키지를 추가적으로 사용한다.

# 비교 그룹간 분포 비교와 회귀 방정식 표기

* 목표 : 비교의 대상이 되는 두 개의 그룹에 대한 산점도와 추세선을 각각 그리고 추세선의 선형회귀 방정식을 표현하여 각각의 그룹의 분포를 서로 비교하고 통계적 유의성을 전달
* 활용 시각화 : 산점도, 회귀선, 클러스터 산점도, 러그 플롯, 선형회귀
* 사용 패키지 : ggplot2, ggalt

데이터 시각화는 데이터 자체를 표시하는 경우도 있지만 데이터 분석을 위해 생성한 데이터 분석 모델의 결과를 표시해야할 경우도 많다. 데이터 분석 모델에는 여러가지가 있고 데이터 모델을 지원하는 패키지에 해당 데이터 모델을 가장 잘 표현할 수 있는 데이터 시각화 함수를 포함하는 경우가 많지만 시각화 품질이 떨어져서 ggplot2를 사용하는 경우도 많다. 이번 절에서는 선형회귀 방정식을 ggplot2 시각화에 표현하는 방법을 알아보겠다.

이번 절에서 최종 생성되는 목표 시각화는 인문계열과 비교한 의약계열의 졸업자수 대비 취업자수 현황이다. 이를 위해 전체 학과의 졸업자수 대비 취업자수 산점도를 기본으로 의약계열 졸업자와 취업자의 선형회귀 모델을 생성한다. 이 그룹에 대조 그룹으로 인문계열 졸업자와 취업자의 선형회귀 모델을 생성하여 붉은 색으로 표현한 후 각각의 범례 라벨에 선형회귀 방정식을 표현한다. 마지막으로 의약계열의 졸업자와 취업자의 밀집도를 확인하기 위해 러그 플롯을 가장자리에 붙이도록 하겠다.

먼저 의약계열과 인문계열의 선형회귀 모델 생성을 위한 데이터 전처리를 해야한다. 선형회귀 모델 생성을 위해서는 R 기본 패키지 중에 하나인 stats패키지의 lm()을 사용한다. lm()으로 생성된 선형회귀 모델에서 필요한 회귀 계수를 산출하기 위해 broom 패키지의 glance()와 tidy() 패키지를 사용한다.

broom 패키지는 tidyverse 생태계에서 모델링 및 기계 학습을 위해 제공하는 Tidymodels 프레임워크에 포함된 패키지이다. broom 패키지는 tibble을 사용하여 생성된 모델에 대한 주요 정보를 요약하는 세개의 함수(glance(), tidy(), argument())를 제공한다. glance()는 전체 모델을 한 행으로 요약한 tibble을 돌려주는 함수로써 적합도 측정값(fit measures), 잔차에 대한 가설 검정을 위한 p-값, 또는 모델 수렴 정보를 제공한다.

glance(x, …)  
- x : stats:lm()으로 생성한 lm 객체  
- … : 추가적인 매개변수

tidy()는 모델에 사용된 각각의 독립 변수들에 주요 정보들을 제공하는 함수이다. 회귀 모델의 경우는 회귀 계수를 중심으로 표준 오차, p-값등을 제공한다.

tidy(x, conf.int = FALSE, conf.level = 0.95, …)  
- x : stats:lm()으로 생성한 lm 객체  
- conf.int : 신뢰구간을 포함할지 여부를 결정하는 논리값  
- conf.level : 신뢰구간을 포함할 경우 신뢰구간의 범위 설정

다음은 의약계열과 인문계열 학과의 졸업자 수와 취업자 수의 선형 회귀 모델을 만들고 glance()와 tidy()를 사용해 각각의 모델 정보와 회귀 계수 데이터를 생성하는 코드이다.

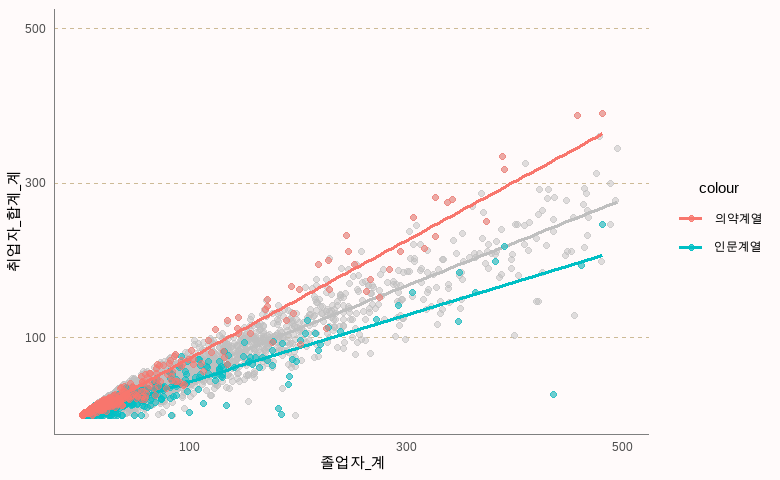
## broom 패키지 설치  
if(!require(broom)) {  
 install.packages('broom')  
 library(broom)  
}  
  
## 의약계열과 인문계열 데이터 생성   
df\_의약 <- df\_취업통계 |> filter(대계열 == '의약계열')  
df\_인문 <- df\_취업통계 |> filter(대계열 == '인문계열')  
  
## 의약계열과 인문계열의 선형회귀 모델 생성  
model\_lm\_의약 <- lm(df\_의약$취업자\_합계\_계 ~ df\_의약$졸업자\_계)  
model\_lm\_인문 <- lm(df\_인문$취업자\_합계\_계 ~ df\_인문$졸업자\_계)  
  
## 각각의 선형회귀 모델의 요약 정보 생성  
glance\_의약 <- broom::glance(model\_lm\_의약)  
glance\_인문 <- broom::glance(model\_lm\_인문)  
  
## 각각의 선형회귀 모델의 회귀 계수 생성  
tidy\_의약 <- broom::tidy(model\_lm\_의약)  
tidy\_인문 <- broom::tidy(model\_lm\_인문)

이제 그래프에 표현할 선형 회귀 방정식을 만든다. tidy()의 결과에서 회귀 계수를 산출하여 paste0()를 이용해 회귀방정식을 생성하고 R 스퀘어 값을 뒤에 붙여주는데 제곱에 해당하는 문자를 붙이기 위해 유니코드 문자(0B2)를 사용한다.

## 의약계열의 회귀방정식과 R^2값에 대한 문자열 생성  
equ\_의약 <- paste0('의약계열 : y = ', round(tidy\_의약$estimate[2], 2), 'x + ', round(tidy\_의약$estimate[1], 2), ', R\u00B2', ' = ',round(glance\_의약$r.squared, 3))  
  
## 인문계열의 회귀방정식과 R^2값에 대한 문자열 생성  
equ\_인문 <- paste0('인문계열 : y = ', round(tidy\_인문$estimate[2], 2), 'x + ', round(tidy\_인문$estimate[1], 2), ', R\u00B2', ' = ',round(glance\_인문$r.squared, 3))

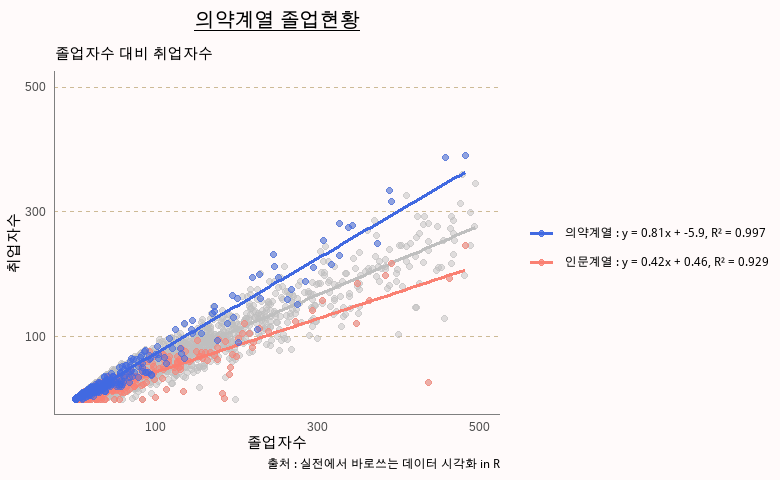
이제 본격적으로 그래프를 그리겠다. 우선 전체 산점도(geom\_point())를 그리고 그 위에 의약 계열 산점도와 회귀선(geom\_smooth()), 인문계열 산점도와 회귀선을 그리고 축의 스케일을 설정한다.

p\_lm\_equ <- df\_취업통계 |>   
 ggplot() +  
 ## 전체 계열의 산점도 생성  
 geom\_point(aes(x = 졸업자\_계, y = 취업자\_합계\_계), color = 'grey75', alpha = 0.5) +  
 ## 전체 계열의 회귀선 생성  
 geom\_smooth(aes(x = 졸업자\_계, y = 취업자\_합계\_계), color = 'grey75', se = F, method = 'lm') +   
 ## 인문계열의 산점도 생성  
 geom\_point(data = df\_인문,  
 aes(x = 졸업자\_계, y = 취업자\_합계\_계, color = '인문계열'), alpha = 0.5) +   
 ## 인문계열의 회귀선 생성  
 geom\_smooth(data = df\_인문,  
 aes(x = 졸업자\_계, y = 취업자\_합계\_계, color = '인문계열'), se = F, method = 'lm') +   
 ## 의약계열의 산점도 생성  
 geom\_point(data = df\_의약,  
 aes(x = 졸업자\_계, y = 취업자\_합계\_계, color = '의약계열'), alpha = 0.5) +   
 ## 의약계열의 회귀선 생성  
 geom\_smooth(data = df\_의약,  
 aes(x = 졸업자\_계, y = 취업자\_합계\_계, color = '의약계열'), se = F, method = 'lm') +   
 ## X축 스케일의 눈금과 라벨을 100, 300, 500로 설정하고 0부터 500까지 범위 설정  
 scale\_x\_continuous(breaks = c(100, 300, 500), labels = c(100, 300, 500), limits = c(0, 500)) +  
 ## Y축 스케일의 눈금과 라벨을 100, 300, 500로 설정하고 0부터 500까지 범위 설정  
 scale\_y\_continuous(breaks = c(100, 300, 500), labels = c(100, 300, 500), limits = c(0, 500))  
  
p\_lm\_equ



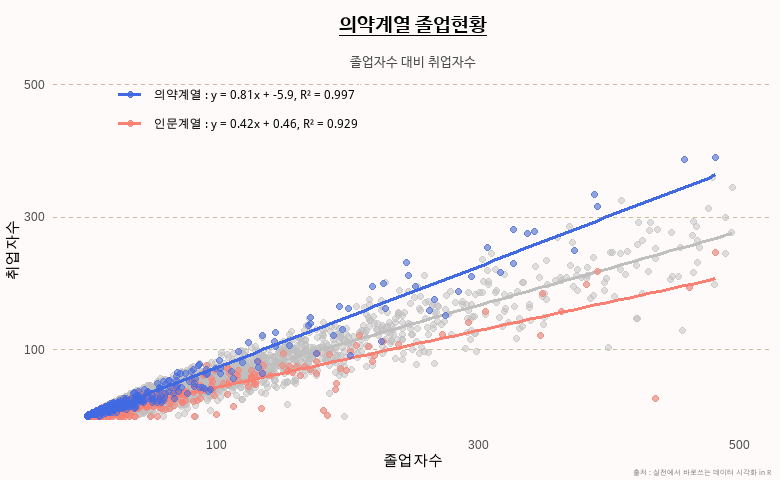
앞의 그래프에 미리 산출해 놓은 선형회귀 방정식을 범례 라벨로 사용하고 의약계열과 인문계열의 색을 설정하겠다. 그리고 그래프 제목, 부제목, 캡션을 설정하는데, 이 중 그래프 제목은 expression()과 underline()을 사용하여 밑줄을 그리도록 하겠다.

p\_lm\_equ <- p\_lm\_equ +   
 ## 그래프 제목을 설정하는데 밑줄을 그은 제목으로 설정  
 labs(title = expression(underline('의약계열 졸업현황')),  
 ## X축과 Y축의 제목 설정  
 x = '졸업자수', y = '취업자수',   
 ## 부제목 설정  
 subtitle = "졸업자수 대비 취업자수",   
 ## 캡션 설정  
 caption = '출처 : 실전에서 바로쓰는 데이터 시각화 in R') +  
 ## X축의 눈금과 라벨을 100, 300, 500으로 설정하고 범위를 0에서 500까지 설정   
 scale\_x\_continuous(breaks = c(100, 300, 500), labels = c(100, 300, 500), limits = c(0, 500)) +   
 ## Y축의 눈금과 라벨을 100, 300, 500으로 설정하고 범위를 0에서 500까지 설정  
 scale\_y\_continuous(breaks = c(100, 300, 500), labels = c(100, 300, 500), limits = c(0, 500)) +  
 ## 의약계열과 인문계열 색을 설정하고 범례에 표현되는 라벨을 방정식으로 대체  
 scale\_color\_manual(name = NULL, values = c('의약계열' = '#4169E1', '인문계열' = '#FA8072'), labels = c(equ\_의약, equ\_인문))  
  
p\_lm\_equ



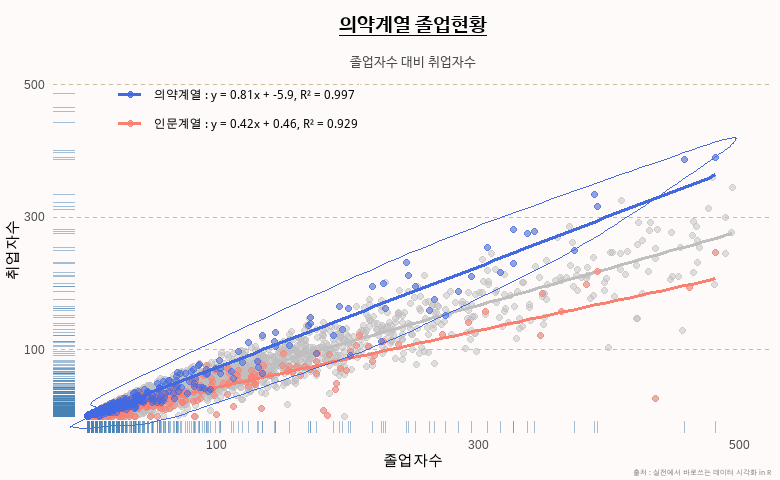
앞의 그래프를 보면 볌례가 너무 길기 때문에 상대적으로 그래프의 크기가 줄었고 이로 인해 중간에 표현된 그래프 제목이 왼쪽으로 치우쳐 보인다. 이를 정상적으로 보여주기 위해 그래프 밖에 표현된 범례를 그래프의 빈 공간인 왼쪽 상단에 위치하도록 옮겨주겠다. 그리고 그래프의 테마를 다음과 같이 설정한다.

font\_add('NanumBarunGothic', 'c:/windows/fonts/NanumBarunGothic.ttf')  
font\_add('NanumBarunGothicBold', 'c:/windows/fonts/NanumBarunGothicBold.ttf')  
  
p\_lm\_equ <- p\_lm\_equ +  
 theme(legend.position=c(.25,.9), ## 범례의 위치를 그래프 안쪽으로 조절  
 ## 범례의 배경을 제거  
 legend.background = element\_rect(fill = NA),  
 ## 범례 키의 배경색을 제거  
 legend.key = element\_rect(fill = NA),   
 ## 그래프 제목의 폰트, 크기 설정  
 plot.title = element\_text(size = 15, family = 'NanumBarunGothicBold'),  
 ## 부제목의 수평정렬, 크기, 폰트, 색 설정  
 plot.subtitle = element\_text(vjust = 0.5, size = 10, family = 'NanumBarunGothic', color = 'grey30'),  
 ## 축의 선을 제거  
 axis.line = element\_blank(),  
 ## 그래프 전체 배경색 설정  
 plot.background = element\_rect(fill = '#FFFAFA'),  
 ## 그래프 패널 배경색 설정  
 panel.background = element\_rect(fill = '#FFFAFA'),  
 ## 그래프의 여백 설정  
 plot.margin = margin(0.025, 0.01, 0.01, 0.01, "npc"),  
 ## 캡션 색, 크기, 수평 위치 설정  
 plot.caption = element\_text(color = 'grey50', size = 5, hjust = 1)  
 )  
   
p\_lm\_equ



앞의 그래프에 의약계열 학과의 밀도를 확인하기 위해 러그 플롯을 가장자리에 위치시키겠다. 그리고 의약계열 학과의 분포가 명확히 보이도록 의약계열 학과의 클러스터를 원으로 묶어주도록 한다.

## ggalt 패키지 설치  
if(!require(ggalt)) {  
 install.packages('ggalt')  
 library(ggalt)  
}  
  
p\_lm\_equ +  
 ## 의약계열 데이터의 전반적 위치를 표시하는 원 레이어 추가  
 geom\_encircle(data = df\_의약,   
 aes(x = 졸업자\_계, y = 취업자\_합계\_계, color='의약계열')) +  
 ## 의약계열 데이터의 밀도를 표시하는 가장자리 레이어 추가  
 geom\_rug(data = df\_의약,  
 aes(x = 졸업자\_계, y = 취업자\_합계\_계), col= "steelblue", alpha=0.5)



# 전체 그룹의 비교 시각화와 설명문 삽입

* 목표 : 전체 데이터의 그룹별 분포와 기술 통계를 시각화하여 전달하고 시각화를 통해 전달하고자 하는 인사이트를 삽입
* 활용 시각화 : 박스 플롯, 바이올린 플롯, 산점도(jitter)
* 사용 패키지 : ggplot2, ggtext

시각화는 독자들에게 데이터의 특성을 그림으로 표현하는 것이 최우선이지만 그림이 가지는 내재된 의미에 대해 간단한 설명을 달아주면 그 효과가 증폭된다. 설명을 달아주는 방법에 대해 알아본다.

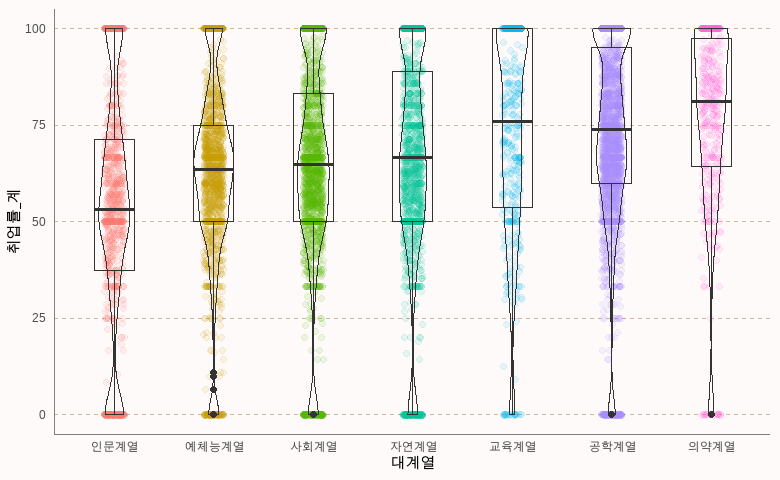
이번 시각화는 취업통계 결과를 계열별로 비교하는 박스플롯 위에 바이올린 플롯과 흩뿌려진 산점도를 그린다. 이 기본 그래프 위에 각각의 그룹간의 특성을 나타내기 위한 사례수, 평균, 중앙값 등을 텍스트로 표현하고 전체적인 시각화 그래프가 가지는 의미를 제목 아래 박스로 간략하게 표기한다.

먼저 사용하는 데이터는 앞서 사용했던 df\_취업통계 데이터이고 이 데이터에서 추가적 정보를 전처리하여 각 계열별 평균, 중간값, 사례수 등을 산출한다.

## 전체 취업률 평균과 중간값, 사례수를 산출  
df\_전체\_요약 <- df\_취업통계 |>  
 summarise(mean = mean(취업률\_계), median = median(취업률\_계), n = n())  
  
## 전체 사례수를 출력하기 위한 문자열 생성  
subtitle <- paste0('전체 사례수 : ', scales::comma(df\_전체\_요약$n, accuracy = 1))  
  
## 계열별 중간값과 사례수 산출  
df\_계열\_요약 <- df\_취업통계 |>  
 group\_by(대계열) |>  
 summarise(mean = mean(취업률\_계), median = median(취업률\_계), n = n())  
  
## 대계열의 순서를 평균의 순서로 설정  
df\_취업통계$대계열 = reorder(df\_취업통계$대계열, df\_취업통계$취업률\_계, mean)

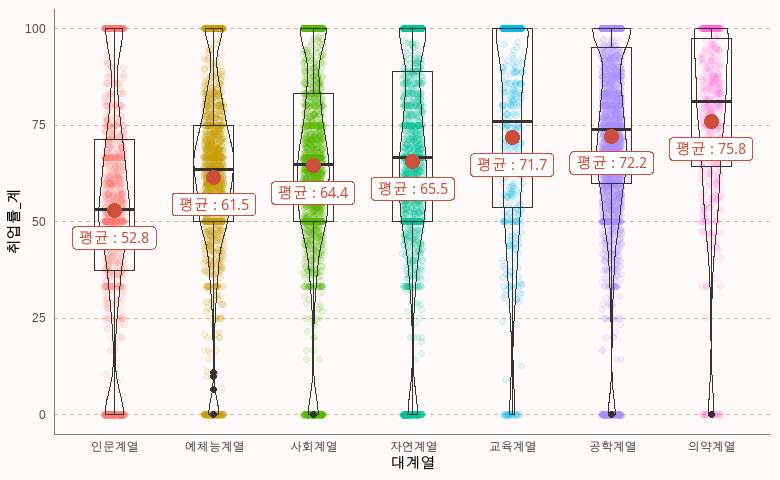
이제 먼저 기본 시각화 그래프를 그린다. 기본 시각화 그래프는 각 계열별 산점도 위에 박스 플롯과 바이올린 플롯을 사용하겠다.

p\_desc <- df\_취업통계 |>  
 ggplot() +  
 ## 대계열별 학과의 취업률 흩어진 산점도 레이어 생성  
 geom\_jitter(aes(x = 대계열, y = 취업률\_계, color = 대계열), width = 0.1, alpha = 0.1, show.legend = F) +  
 ## fill을 NA로 설정하여 투명한 색으로 설정된 대계열별 박스 플롯 레이어를 추가  
 geom\_boxplot(aes(x = 대계열, y = 취업률\_계), fill = NA, width = 0.4) +  
 ## fill을 NA로 설정하여 투명한 색으로 설정된 대계열별 바이롤린 플롯 레이어를 추가  
 geom\_violin(aes(x = 대계열, y = 취업률\_계), fill = NA, width = 0.4)  
  
p\_desc



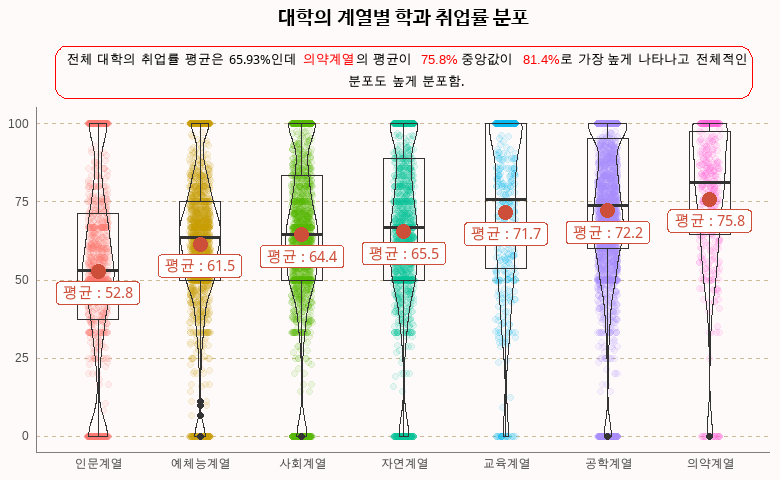
기본 그래프 위에 첫번째 추가적 정보인 각 계열별 평균값을 표시한다. 전처리 과정에서 산출한 계열별 평균값 점(geom\_point())과 이 점의 값을 표시하는 텍스트 라벨(geom\_label()) 레이어를 추가한다.

p\_desc <- p\_desc +  
 geom\_point(data = df\_계열\_요약, aes(x = 대계열, y = mean), color = 'tomato3', size = 4) +  
 geom\_label(data = df\_계열\_요약, aes(x = 대계열, y = mean, label = paste0('평균 : ', round(mean, 1))), color = 'tomato3', size = 4, nudge\_y = -7)  
  
p\_desc



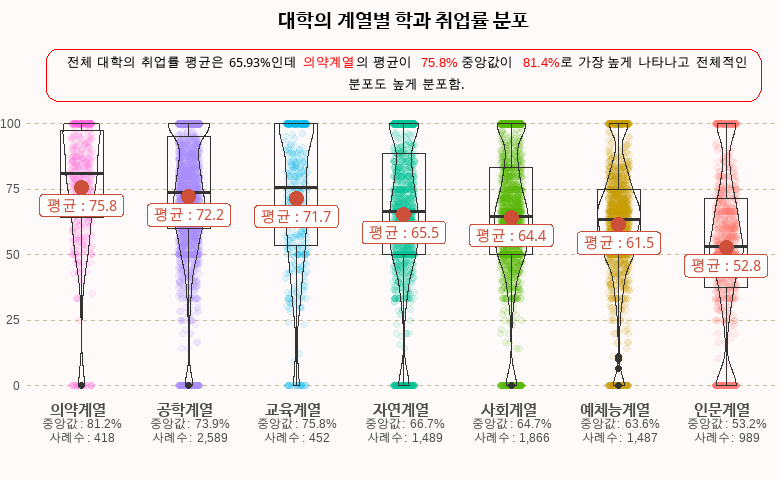
두번째 정보는 제목 밑에 이 데이터에 대한 간략한 설명을 추가하는 것이다. 이를 위해 ggtext 패키지의 element\_textbox\_simple()을 사용하여 테마요소의 plot.title과 plot.subtitle의 속성 값을 설정하겠다. element\_textbox\_simple()은 문자열에 관련한 테마 요소의 속성 설정에 사용하는 함수로써 마크다운 문법으로 기술된 문자열을 상자에 담아 표기하는 함수이다.

if(!require(ggtext)) {  
 install.packages('ggtext')  
 library(ggtext)  
}  
  
font\_add('NanumBarunGothic', 'c:/windows/fonts/NanumBarunGothic.ttf')  
font\_add('NanumBarunGothicBold', 'c:/windows/fonts/NanumBarunGothicBold.ttf')  
  
## 제목으로 사용할 마크다운 문자열 생성  
title <- "<b><span style= 'font-family: NanumBarunGothicBold;font-size: 15pt'>대학의 계열별 학과 취업률 분포</b>"  
  
## 데이터 설명문으로 사용할 마크다운 문자열 생성  
subtitle <- "<span style = 'font-size:10pt'>전체 대학의 취업률 평균은 65.93%인데 <span style = 'color:red;'>의약계열</span>의 평균이 <span style = 'color:red;'> 75.8%</span> 중앙값이 <span style = 'color:red;'> 81.4%</span>로 가장 높게 나타나고 전체적인 분포도 높게 분포함.</span>"  
  
p\_desc <- p\_desc +   
 ## 제목과 부제목을 앞에서 생성한 마크다운 문자열로 설정  
 labs(title = title, x = NULL, y = NULL, subtitle = subtitle) +  
 ## plot.title을 element\_textbox\_simple로 세부 속성(수평 정렬) 설정  
 theme(plot.title = element\_textbox\_simple(halign = 0.5),  
 plot.subtitle = element\_textbox\_simple(  
 width = unit(0.95, "npc"), ## 텍스트 박스의 너비를 전체 플롯 너비의 0.95로 설정  
 size = 13, ## 텍스트의 크기를 13으로 설정  
 lineheight = 1, linetype = 1, ## 텍스트 박스의 타입과 높이 설정  
 padding = margin(5.5, 0.5, 5.5, 5.5), ## 텍스트 박스 내부의 여백 설정  
 margin = margin(10, 0, 5.5, 0), ## 텍스트 박스 외부의 여백 설정  
 halign = 0.5, box.color = 'red', ## 텍스트 박스의 수평 정렬과 박스 색 설정  
 r = grid::unit(8, "pt")) ## 텍스트 박스의 모서리 설정  
 )  
  
p\_desc



다음으로 전달할 정보는 각 계열별 중간값과 사례수이다. 중간값은 박스 플롯에서 선으로 표현되지만 정확한 값을 알아보기는 힘들다. 하지만 평균값이 표기되기 있기 때문에 그래프 내에 표기하면 오히려 혼잡해질 가능성이 있어서 사례수와 함께 X축 계열명에 표기하는 방법을 선택했다.

## 각 계열의 특성을 X축의 라벨에 표현하기 위한 문자열 생성  
labels <- paste0('<span style= "font-family: NanumBarunGothicBold;font-size: 12pt" >', df\_계열\_요약$대계열, '</span> <br>', '중앙값 : ', scales::percent(df\_계열\_요약$median, scale = 1), '<br>', '사례수 : ', scales::comma(df\_계열\_요약$n, accuracy = 1))  
  
## 계열 요약 데이터를 데이터프레임이아닌 네임드 벡터로 변환  
labels\_name <- pull(df\_계열\_요약, 대계열)  
  
p\_desc +   
 ## X축의 여백을 설정하고 마크다운으로 설정한 라벨을 설정하고 순서를 역순으로 설정   
 scale\_x\_discrete(expand = expansion(add = c(0.5, 0.5)), labels = setNames(labels, labels\_name), limits = rev) +   
 theme(axis.line = element\_blank(), ## 축 선을 제거   
 axis.text.x = element\_markdown(), ## X축 라벨을 마크다운으로 해석  
 plot.margin = margin(0.025, 0, 0.075, 0, "npc") ## 플롯의 여백을 설정  
 )



# ggplot 이어 붙이기와 영국 이코노미스트지 스타일의 시각화

목표 : 선 그래프와 영역 그래프를 붙여 데이터를 보는 두가지 관점을 한번에 볼 수 있도록 시각화를 그리고 영국 이코노미스트지에서 사용하는 스타일로 시각화를 꾸며줌  
활용 시각화 : 박스 플롯, 바이올린 플롯, 산점도  
사용 패키지 : ggplot2, ggtext, grid. patchwork

데이터를 시각화할 때 독자의 눈에 잘 띄도록 디자인하기 위해서는 데이터 자체의 표현과 함께 제목의 표현, 축의 형태, 전체 그래프의 외관 설정 등에도 신경써야 한다. ggplot2에서는 다양한 테마요소를 사용하여 이를 설정하는데 하나 하나 설정해서 사용자의 눈에 띄는 디자인을 만드는 것은 상당히 어려운 일임에 틀림없다. 그래서 ggplot2, ggthemes 패키지 등에서 미리 만들어진 테마들을 제공한다. 이들 테마 중에는 워싱턴 포스트와 같이 유명한 언론지에서 사용하는 테마를 제공하지만 영국의 이코노미스트에서 사용하는 테마는 제공하지 않는다. 물론 영국 이코노미스트 스타일의 시각화를 선호하지 않을 수도 있지만 이 스타일을 만들어 보면 이를 응용하여 자신만의 디자인을 할 수 있을 것이라는 점에서 만들어 보고자 한다.

영국 이코노미스트지의 시각화는 크게 세가지 점에서 특징이 있다. 첫 번째는 Y 축 라벨을 플롯 안에 배치하고 눈금선 위에 옅은 회색으로 표기한다. 두 번째는 Y축 라벨을 플롯 안쪽으로 배치했기 때문에 Y축의 선을 제가한다. 마지막으로 그래프의 전체 모양을 꾸며주기 위해 전체 그래프의 맨 위에 붉은 줄을 그리고 왼쪽 상단 귀퉁이에 작은 붉은 네모를 만들어 준다.

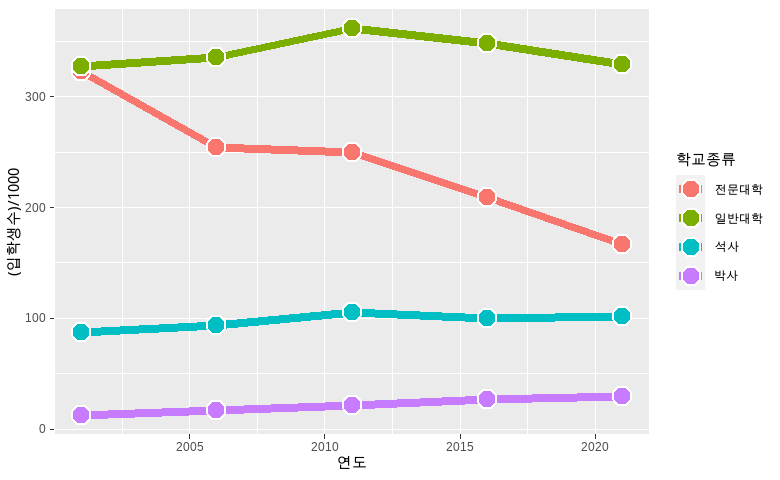
이코노미스트 스타일의 그래프를 그리기 위해 두개의 ggplot 객체를 생성하고 patchwork로 이어 붙이겠다. 이어 붙일 ggplot 객체는 5년 단위의 대학 종류별 입학생 선 그래프와 면적 그래프를 그리는데 선 그래프의 경우 꺽어지는 곳을 힌지 스타일로 꾸며주였다.

먼저 선 그래프를 그리기 위한 데이터를 전처리한다. 데이터는 긴 형태의 입학생수 데이터프레임에서 2001년부터 2021년까지 5년 간격으로 ‘전문대학’, ‘일반대학’, ‘석사’, ’박사’의 입학생수 사용한다.

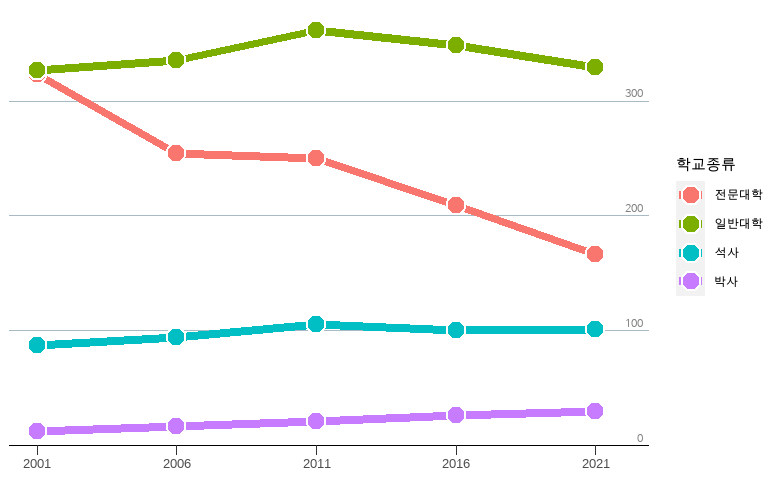
## df\_입학자\_long에서 지역이 '전체'이고 연도가 2001년부터 2021년까지 5년단위인 데이터, 학교종류가 '전문대학', '일반대학', '석사', '박사'인 데이터를 필터링  
df\_total\_line <- df\_입학자\_long |>   
 filter(지역 == '전체', lubridate::year(연도) %in% c(seq(from = 2001, to = 2021, by = 5)), 학교종류 %in% c('전문대학', '일반대학', '석사', '박사'))   
  
## 연도의 점 위치를 정확히 맞추기 위해 연도를 각 년도 1월 1일로 설정   
df\_total\_line <- df\_total\_line |>  
 mutate(연도 = as.Date(paste0(lubridate::year(연도), '-01-01'), format = '%Y-%m-%d'))

첫 번째 선 그래프를 다음과 같이 그린다. 여기서 살펴보아야 할 것이 geom\_point() 레이어에 설정한 shape 미적요소이다. shape 미적요소를 21에서 25번까지를 설정하면 원의 fill(내부 색)과 color(선 색)을 다르게 설정할 수 있다.

theme\_set(  
 theme\_grey()  
)  
  
  
p\_line\_1 <- df\_total\_line |>  
 ggplot() +   
 ## geom\_line 레이어를 생성  
 geom\_line(aes(x = 연도, y = (입학생수)/1000, group = 학교종류, color = 학교종류), size = 2.4) +  
 ## geom\_point 레이어 추가  
 geom\_point(aes(x = 연도, y = (입학생수)/1000, fill = 학교종류), size = 5,  
 shape = 21, # fill과 color를 따로 설정할 수 있는 모양 설정  
 color = "white", ## 점의 color를 흰색으로 설정하여 힌지 효과를 설정  
 stroke = 1 # 점의 경계선 두께 설정   
 )  
  
p\_line\_1

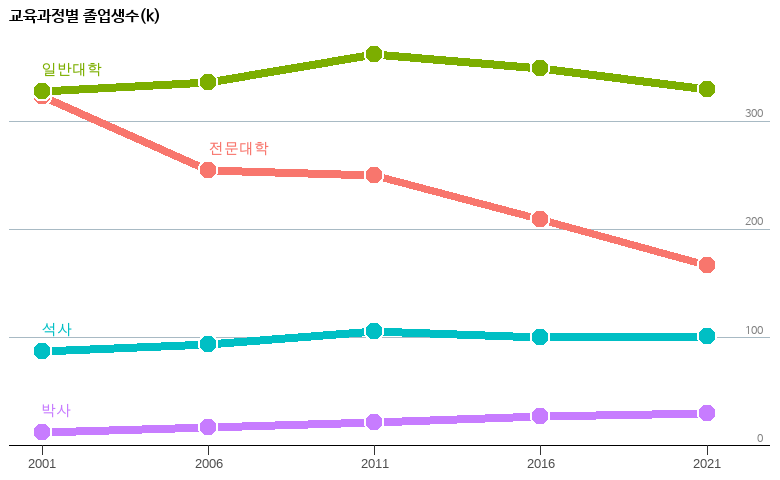


p\_line\_2 <- p\_line\_1 +   
 ##  
 geom\_text(  
 data = data.frame(x = as.Date('2022-09-01'), y = seq(0, 300, by = 100)),  
 aes(x, y, label = y),  
 hjust = 1, # Align to the right  
 vjust = -0.5, # Align to the bottom  
 family = "NanumBarunGothic",  
 size = 3,   
 color = 'grey50'  
 ) +   
 scale\_x\_date(  
 expand = c(0, 0), # 수평 축의 확장 여백이 없이 설정  
 limits = c(as.Date('2000-01-01'), as.Date('2022-12-01')), ## 여백을 없애는 대신 범위에서 여백을 추가  
 ## 축 눈금을 5년마다 하나씩 설정  
 breaks = seq(from = as.Date("2001-01-01"), to = as.Date("2021-01-01"), by = "5 years"),   
 ## 축 라벨을 5년마다 하나씩 설정  
 labels = lubridate::year(seq(from = as.Date("2001-01-01"), to = as.Date("2021-01-01"), by = "5 years"))) +  
 ## Y축의 범위와 확장 여백을 설정  
 scale\_y\_continuous(limits = c(0, 380), expand = c(0, 0)) +   
 theme(  
 # 패널 배경색을 흰색으로 설정  
 panel.background = element\_rect(fill = "white"),  
 # 눈금선을 모두 제거  
 panel.grid = element\_blank(),  
 # Y축의 주 눈금선 색과 두꼐를 설정   
 panel.grid.major.y = element\_line(color = "#A8BAC4", size = 0.3),  
 # Y축의 tick을 제거  
 axis.ticks.length.y = unit(0, "mm"),   
 # X축의 tick을 2mm로 설정  
 axis.ticks.length.x = unit(2, "mm"),  
 # 축 제목을 제거  
 axis.title = element\_blank(),  
 # X축 선색을 검정으로 설정  
 axis.line.x.bottom = element\_line(color = "black"),  
 # X축의 문자 크기를 16으로 설정  
 axis.text.x = element\_text(size = 10),  
 # Y축 선을 제거  
 axis.text.y = element\_blank()  
 )  
  
p\_line\_2



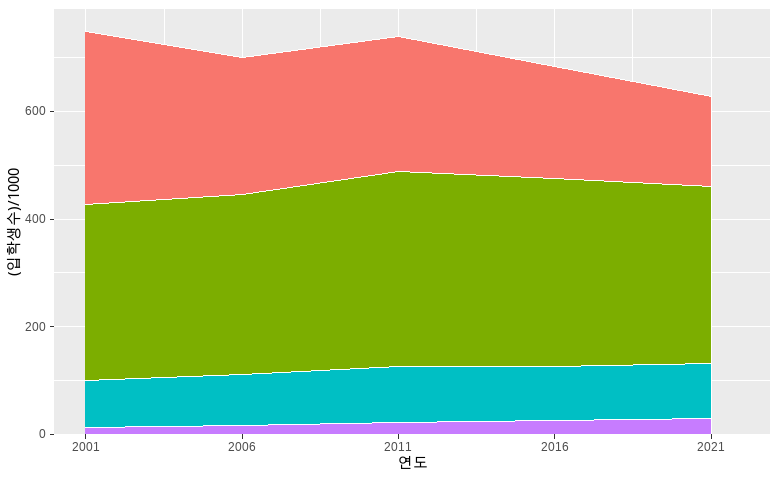
앞의 그래프에 범례를 없애고 범례 내용을 그래프의 시작점에 직접 표기한다. 다만 전문대학과 일반대학의 시작점이 거의 같기 때문에 전문대학은 2006년에 표기하여 데이터를 구분하도록 해준다.

font\_add('NanumBarunGothic', 'c:/windows/fonts/NanumBarunGothic.ttf')  
font\_add('NanumBarunGothicBold', 'c:/windows/fonts/NanumBarunGothicBold.ttf')  
  
## 데이터 라벨을 위한 데이터 산출  
## 일반대학, 석사, 박사는 2001년 데이터, 전문대학은 2006년 데이터를 추출하여 bind\_rows()로 하나의 데이터프레임으로 만들어 줌  
data\_labels <- bind\_rows(  
 df\_total\_line |> filter(lubridate::year(연도) == 2001, 학교종류 %in% c('일반대학', '석사', '박사')),   
 df\_total\_line |> filter(lubridate::year(연도) == 2006, 학교종류 %in% c('전문대학')))  
  
p\_line\_3 <- p\_line\_2 +   
 ## data\_labels 데이터로 각각의 선에 대한 범례를 표기하는 geom\_text 레이어 추가  
 geom\_text(data = data\_labels, aes(x = 연도, y = (입학생수/1000) + 20, label = 학교종류, color = 학교종류), hjust = 0, show.legend = F) +   
 theme(legend.position = 'none') +  
 labs(  
 title = "교육과정별 졸업생수(k)",  
 ) +   
 theme(  
 # theme\_markdown() is provided by ggtext and means the title contains   
 # Markdown that should be parsed as such (the '\*\*' symbols)  
 plot.title = element\_text(family = 'NanumBarunGothicBold', size = 12)  
 )  
  
p\_line\_3



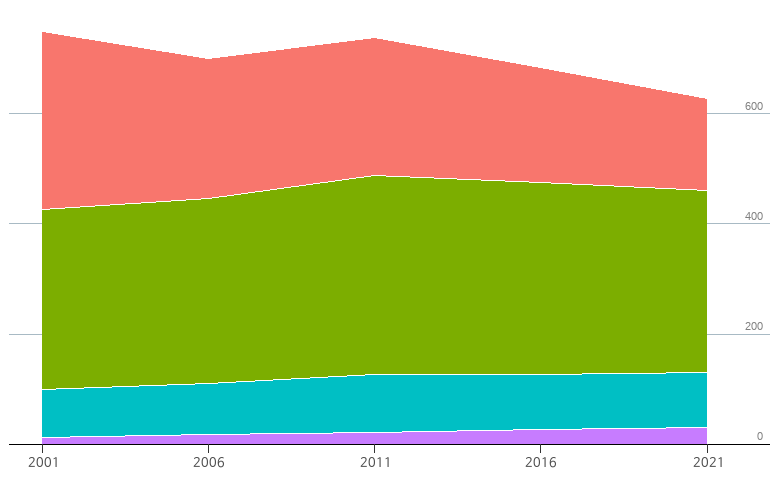
이제 두 번째 면적 그래프를 그려보겠다. 기본 그래프는 앞의 선 그래프와 유사하나 geom\_line() 대신 geom\_area()를 사용하여 면적 그래프로 그려준다.

p\_area\_1 <- df\_total\_line |> ggplot() +  
 # color = "white" indicates the color of the lines between the areas  
 geom\_area(aes(x = 연도, y = (입학생수)/1000, group = 학교종류, fill = 학교종류), color = "white") +  
# scale\_fill\_manual(values = c('grey', 'brown', 'green', 'blue')) +  
 theme(legend.position = "None") + # no legend +  
 scale\_x\_date(  
 expand = c(0, 0), # The horizontal axis does not extend to either side  
 limits = c(as.Date('2000-01-01'), as.Date('2022-12-01')),   
 breaks = seq(from = as.Date("2001-01-01"), to = as.Date("2021-01-01"),  
 by = "5 years"),   
 labels = lubridate::year(seq(from = as.Date("2001-01-01"), to = as.Date("2021-01-01"),  
 by = "5 years")) # Set custom break locations  
 # Set custom break locations  
 # labels = c("2008", "12", "16", "20") # And custom labels on those breaks!  
 ) +   
 scale\_y\_continuous(  
 limits = c(0, 790),  
 expand = c(0, 0)  
 )  
  
p\_area\_1



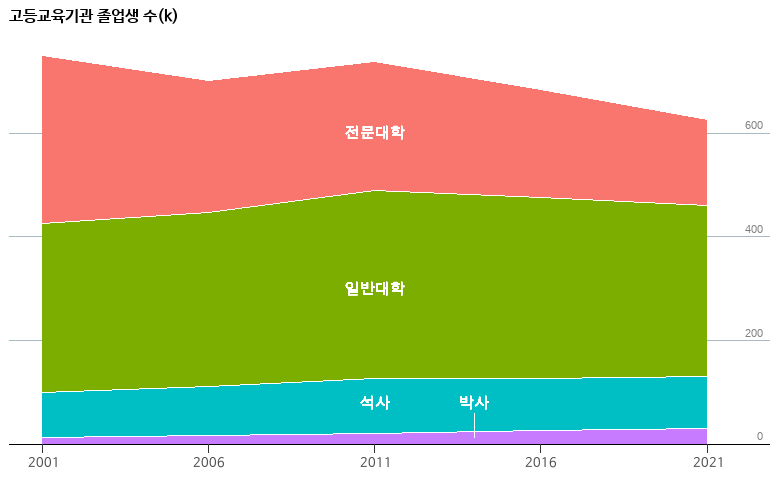
이제 그래프에 이코노미스트지 형태의 눈금선과 Y축 라벨에 대한 스타일을 적용하도록 하겠다.

p\_area\_2 <- p\_area\_1 +   
 geom\_text(  
 data = data.frame(x = as.Date('2022-09-01'), y = seq(0, 800, by = 200)),  
 aes(x, y, label = y),  
 hjust = 1, # Align to the right  
 vjust = -0.5, # Align to the bottom  
 size = 3,   
 color = 'grey50'  
 ) +   
 theme(  
 # Set background color to white  
 panel.background = element\_rect(fill = "white"),  
 # Remove all grid lines  
 panel.grid = element\_blank(),  
 # But add grid lines for the vertical axis, customizing color and size   
 panel.grid.major.y = element\_line(color = "#A8BAC4", size = 0.3),  
 # Remove tick marks on the vertical axis by setting their length to 0  
 axis.ticks.length.y = unit(0, "mm"),   
 # But keep tick marks on horizontal axis  
 axis.ticks.length.x = unit(2, "mm"),  
 # Remove the title for both axes  
 axis.title = element\_blank(),  
 # Only the bottom line of the vertical axis is painted in black  
 axis.line.x.bottom = element\_line(color = "black"),  
 # Remove labels from the vertical axis  
 axis.text.y = element\_blank(),  
 # But customize labels for the horizontal axis  
 axis.text.x = element\_text(family = "NanumBarunGothic", size = 10)  
 )  
  
p\_area\_2



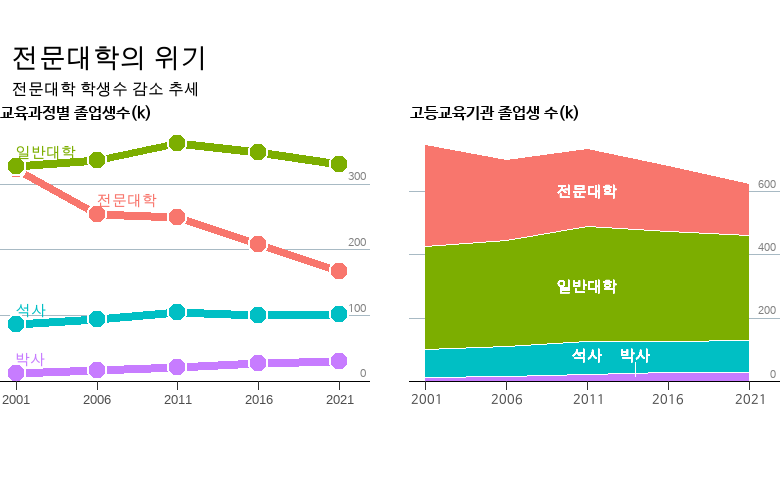
앞서 그린 면적 그래프에서 범례를 제거했기 때문에 각각의 데이터에 데이터 라벨을 표현해주도록 하겠다. 그런데 맨 아래의 박사 영역은 범례를 표현하기에 너무 영역이 좁기 때문에 이 부분은 조금 위에 데이터 라벨을 표기하고 선으로 데이터 영역을 연결해주어 두 번째 그래프를 완성한다.

p\_area\_3 <- p\_area\_2 +   
 geom\_text(aes(x = as.Date('2011-01-01'), y = 600), label = '전문대학', color = 'white') +   
 geom\_text(aes(x = as.Date('2011-01-01'), y = 300), label = '일반대학', color = 'white') +   
 geom\_text(aes(x = as.Date('2011-01-01'), y = 80), label = '석사', color = 'white') +   
 geom\_text(aes(x = as.Date('2014-01-01'), y = 80), label = '박사', color = 'white') +   
 geom\_segment(aes(x = as.Date('2014-01-01'), xend = as.Date('2014-01-01'), y = 12, yend = 60), color = 'white') +   
 theme(legend.position = 'none') +   
 labs(title = "고등교육기관 졸업생 수(k)") +   
 theme(plot.title = element\_text(family = 'NanumBarunGothicBold', size = 12))  
  
p\_area\_3



두개의 그래프가 완성되었으니 이제 patchwork 패키지를 사용하여 그래프를 수평 방향으로 이어 붙이고 이어 붙인 그래프의 전체 제목과 테마요소를 설정한다.

if(!require(patchwork)) {  
 install.packages('patchwork')  
 library(patchwork)  
}  
  
## 첫 번째 그래프의 플롯 여백을 조절  
plt1 <- p\_line\_3 + theme(plot.margin = margin(0, 0.05, 0, 0, "npc"))  
  
## 두 번째 그래프의 플롯 여백을 조절  
plt2 <- p\_area\_3 + theme(plot.margin = margin(0, 0, 0.05, 0, "npc"))  
  
## 두개의 그래프를   
plt <- plt1 | plt2  
  
title\_theme <- theme(  
 plot.title = element\_text(hjust = 0.02, size = 20, margin = margin(0.5, 0, 0.3, 0, "npc")),  
 plot.subtitle = element\_text(hjust = 0.02, size = 12, margin = margin(0.4, 0, 0.5, 0, "npc")),   
 plot.margin = margin(0.075, 0, 0.1, 0, "npc")  
)  
  
plt <- plt + plot\_annotation(  
 title = "전문대학의 위기",  
 subtitle = "전문대학 학생수 감소 추세",  
 theme = title\_theme  
)  
  
plt



전반적으로 그래프가 완성되었다. 이제 이코노미스트지에서 그래프를 꾸미기 위해 사용하는 상단의 붉은색 선과 왼쪽 상단의 네모를 그리도록 한다. 이를 위해 grid 패키지의 기능을 이용한다.

grid 패키지는 저수준(low-level) 플로팅 요소들을 그리는데 사용되는 각종 함수들을 묶어놓은 패키지이다. 보통 R base 패키지에서 기본적으로 그래프나 플롯을 만들기 위해 제공하는 플로팅 함수들을 저수준 플로팅 함수라고 한다.

| 함수 | 설명 |
| --- | --- |
| plot() | 플로팅 함수를 사용하기 위한 플로팅 공간을 만드는 함수 |
| point(x, y) | 점을 찍는 플로팅 함수 |
| abline(), segment() | 선을 그리는 플로팅 함수 |
| arrows() | 선의 끝을 화살표로 그리는 함수 |
| curve() | 곡선을 그리는 플로팅 함수 |
| rect(), polygon() | 사각형과 다각형을 그리는 플로팅 함수 |
| text() | 문자를 그리는 플로팅 함수 |
| legend() | 범례를 만드는 플로팅 함수 |
| axis() | 축을 만드는 플로팅 함수 |

grid 패키지는 완성된 ggplot2 플롯에 저수준 플로팅 요소들을 추가하는데 사용하는 패키지이다. grid 패키지에서 기본적 그래픽 요소들(Primitive Graphical Element)를 그리기 위해 다양한 함수를 제공하지만 단독으로는 사용되지 않고 완성된 플롯에 그래픽 요소를 추가하는데 사용된다. 사실상 ggplot2의 그래픽 시스템은 grid 패키지의 그래픽 시스템위에서 구축되었기 때문에 grid 패키지에서 제공하는 저수준의 함수들이 ggplot2 객체, 그리고 lattice 패키지를 통해 만들어진 플롯에 잘 작동한다. grid 패키지는 R의 설치시에 자동적으로 설치되는 패키지 중 하나이다. 따라서 install.packages()를 사용해 설치가 따로 필요하지는 않지만 library()를 사용하여 패키지 로딩은 필요하다.

이코노미스트지 스타일로 그래프를 꾸며주기 위해 사용하는 grid 패키지의 함수는 다음과 같다.

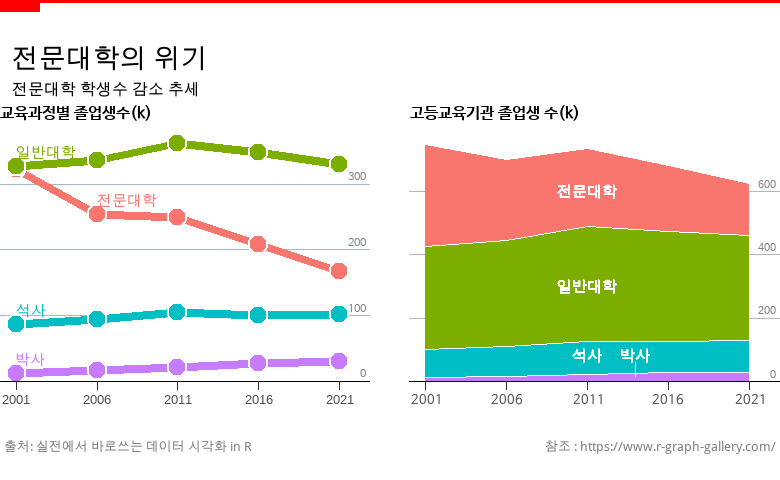
grid.lines(x = unit(c(0, 1), “npc”), y = unit(c(0, 1), “npc”), default.units = “npc”, arrow = NULL, name = NULL, gp=gpar(), draw = TRUE, vp = NULL)  
- x : X값으로 사용될 수치 벡터나 unit 객체  
- y : Y값으로 사용될 수치 벡터나 unit 객체  
- default.units : X, Y에 수치벡터가 설정될 때 사용될 단위 설정  
- arrow : 선의 끝을 화살표로 만들지에 대한 설정  
- name : 문자열 ID  
- gp : 그래픽 파라메터 설정  
- draw : 그래픽 출력을 할지를 설정하는 논리값  
- vp : Grid Viewpoint 객체 설정

grid.rect(x = unit(0.5, “npc”), y = unit(0.5, “npc”), width = unit(1, “npc”), height = unit(1, “npc”), just = “centre”, hjust = NULL, vjust = NULL, default.units = “npc”, name = NULL, gp=gpar(), draw = TRUE, vp = NULL)  
- width : 사각형의 너비를 설정하는 수치 벡터  
- height : 사각형의 높이를 설정하는 수치 벡터  
- just : x, y에 상대적인 정렬 설정  
- hjust : 수평 정렬 설정  
- vjust : 수직 정렬 설정

grid.text(label, x = unit(0.5, “npc”), y = unit(0.5, “npc”), just = “centre”, hjust = NULL, vjust = NULL, rot = 0, check.overlap = FALSE, default.units = “npc”, name = NULL, gp = gpar(), draw = TRUE, vp = NULL)  
- label : 표시할 문자열  
- rot : 문자열의 표시 각도 설정  
- check.overlap : 문자열이 겹쳐서 표현될지를 설정하는 논리값

grid 패키지의 함수를 사용하여 다음과 같이 그려주면 이코노미스트지 스타일의 그래프가 만들어 진다.

library(grid)  
  
## 그래프 상단의 붉은 선을 그림  
grid.lines(x = c(0, 1), y = c(1, 1), gp = gpar(col = "red", lwd = 4))  
  
# 그래프 왼쪽 상단의 붉은 상자를 그림  
grid.rect(x = c(0, 0), y = c(1, 1), width = 0.05, height = 0.025, just = c("left", "top"),  
 gp = gpar(fill = "red", col = "red", lwd = 0))  
  
# 그래프 왼쪽 하단에 캡션을 삽입  
grid.text('출처: 실전에서 바로쓰는 데이터 시각화 in R', x = 0.005, y = 0.06, just = c("left", "bottom"),  
 gp = gpar(col = "grey50", fontsize = 10))  
  
# 그래프 오른쪽 하단에 캡션을 삽입  
grid.text("참조 : https://www.r-graph-gallery.com/", x = 0.995, y = 0.06, just = c("right", "bottom"),  
 gp = gpar(col = "grey50", fontsize = 10))



# 둥근 막대 그래프

목표 : 막대 그래프를 극 자료계를 사용하여 둥근 형태의 둥근 막대 그래프로 변환 활용 시각화 : 막대 그래프, 극 좌표계 사용 패키지 : ggplot2, geomtextpath

원형 막대 그래프는 막대 그래프를 둥글게 표현한 그래프이다. 사실 둥글게 데이터를 표현하는 시각화 방식은 데이터가 원의 중심으로 고르게 표현되다 보니 한 눈에 데이터를 살펴보기가 좋지만 최근 선호되지는 않는 방식이다. 특히 파이 차트의 경우 그 내부의 구성 비율을 나타내기 위해 많이 사용되지만 사람의 눈으로 그 비율을 정확하게 비교하기란 쉬운 일이 아니어서 데이터를 정확히 파악할 수 없다는 단점이 이다. 그래서 파이 차트는 내부에 표시되는 데이터를 가급적 줄여서 중요한 몇 개의 데이터의 비율을 표현하는데 사용된다. 반면 막대 그래프의 경우 가로로 배열된 수평형 막대 그래프는 대부분 공간의 제약으로 많은 막대를 표현하지 못한다. 그래서 세로로 배열된 수직형 막대 그래프를 사용하는데 이 경우도 데이터가 많아질 경우 한눈에 비교하기가 어렵다. 그래서 원형 그래프와 막대 그래프의 장점을 사용한 것이 원형 막대 그래프이다.

원형 막대 그래프는 수평형 막대 그래프보다 더 많은 데이터의 표현이 가능하고 원의 중심으로 균등하게 데이터가 배치되기 때문에 전체 데이터를 한눈에 파악하기 쉽다는 장점이 있다. 하지만 막대의 길이를 정확하게 비교하기가 어렵기 때문에 멀리 떨어진 데이터의 비교가 어렵다는 단점이 있다. 따라서 원형 막대 그래프는 여러 그룹으로 구분된 막대 그래프를 원형으로 그려줌으로써 그룹간의 비교, 그룹내의 비교를 한눈에 파악하기 쉽게 그려주는게 바람직하다.

원형 막대 그래프는 막대 그래프를 극 좌표계를 사용하여 둥글게 말아줌으로써 만들 수 있다. 따라서 먼저 막대 그래프를 그려야 한다. 원형 막대 그래프에는 막대의 길이의 차이로 데이터를 비교할 수있는 dodge형 막대 그래프가 적절하지만 fill형 막대 그래프를 사용할 수도 있다.

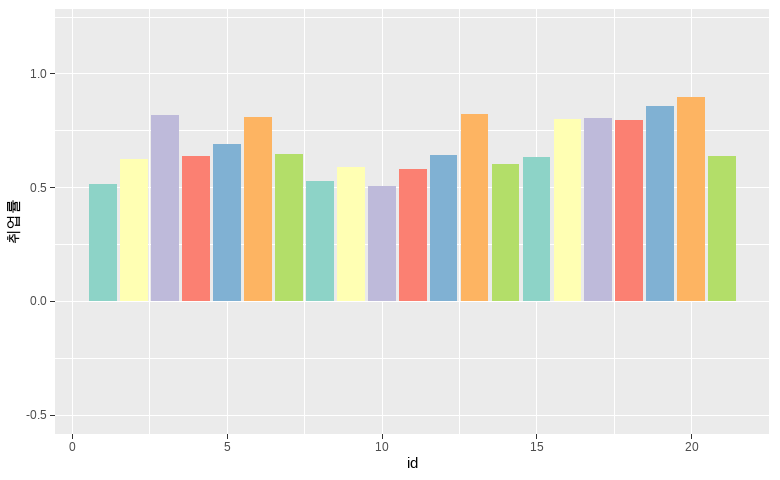
원형 막대 그래프를 그리기 위해 사용할 데이터는 대학 과정별 대계열의 취업률을 사용한다. 다만 각 막대에 데이터 계열과 데이터 값을 표현하기 위해 각각의 위치를 설정할 필요가 있기 때문에 그룹화한 dodge형 막대 그래프를 사용하지 않고 그룹이 없이 일련번호를 사용한 개별 막대 그래프를 사용하였다. 또 각 계열의 평균값을 텍스트로 표현하되 둥근 형태로 표현되는 텍스트를 사용하기 위해 geomtextpath 패키지의 coord\_curvedpolar()를 사용하여 극 좌표계를 설정하고 geom\_textpath()를 사용하여 극 좌표계 중심으로 둥글게 표현되는 문자열을 사용했다.

우선 다음과 같이 데이터를 전처리한다.

## 데이터 표현 순서를 설정하기 위해 과정구분 팩터의 순서 설정  
df\_취업통계$과정구분 = fct\_relevel(df\_취업통계$과정구분, '전문대학과정', '대학과정', '대학원과정')  
  
## 데이터 표현 순서를 설정하기 위해 대계열 팩터의 순서 설정  
df\_취업통계$대계열 = fct\_relevel(df\_취업통계$대계열, '인문계열', '사회계열', '교육계열', '자연계열', '공학계열', '의약계열', '예체능계열')  
  
## 과정구분별 대계열별 데어터와 취업률 산출  
df\_취업통계\_계열별 <- df\_취업통계 |>  
 group\_by(과정구분, 대계열) |>   
 summarise(졸업자 = sum(졸업자\_계),   
 취업자 = sum(취업자\_합계\_계),   
 진학자 = sum(진학자\_계),   
 입대자 = sum(입대자),  
 취업불가능자 = sum(취업불가능자\_계),   
 외국인유학생 = sum(외국인유학생\_계),   
 제외인정자 = sum(제외인정자\_계),  
 ## 백분률인 취업률은 그 자체로 합계나 평균을 낼 수 없으니 각 그룹별로 재계산  
 취업률 = 취업자 / (졸업자 - (진학자+입대자+취업불가능자+외국인유학생+제외인정자))) |>  
 ## 계열의 표시 순서를 설정하기 위해 레벨을 재조정  
 arrange(과정구분, 대계열) |>  
 ungroup() |>  
 ## 막대 순서 설정을 위한 일련번호 생성  
 mutate(id = seq(1:n())) |>  
 ## 대계열 이름과 데이터 값을 표현하기 위한 텍스트 각도 계산  
 mutate(angle = 90 - (id-0.5)/n() \* 360) |>  
 ## 180도 넘어가는 막대의 각도를 180도 돌려 거꾸로 문자열이 표시되는 것을 방지  
 mutate(angle1 = case\_when(  
 id > n()/2 ~ angle + 180,   
 id <= n()/2 ~ angle  
 ))  
  
## 과정별 취업률 평균 산출  
df\_취업통계\_과정별 <- df\_취업통계 |>   
 group\_by(과정구분) |>   
 summarise(졸업자 = sum(졸업자\_계),   
 취업자 = sum(취업자\_합계\_계),   
 진학자 = sum(진학자\_계),   
 입대자 = sum(입대자),  
 취업불가능자 = sum(취업불가능자\_계),   
 외국인유학생 = sum(외국인유학생\_계),   
 제외인정자 = sum(제외인정자\_계),  
 ## 백분률인 취업률은 그 자체로 합계나 평균을 낼 수 없으니 각 그룹별로 재계산  
 취업률 = 취업자 / (졸업자 - (진학자+입대자+취업불가능자+외국인유학생+제외인정자))) |>  
 ## 계열의 표시 순서를 설정하기 위해 레벨을 재조정  
 arrange(과정구분) |>  
 ungroup() |>  
 ## 각각의 막대를 분리해 주기 위해 일련번호를 붙여줌  
 mutate(id = seq(1:n())) |>  
 ## 각각의 막대에 표시되는 데이터 계열과 데이터 값은 수직형 문자열 표기(90도 회전)하고 이 문자열은 전체 원(360도)에 균등하게 돌아가면서 배치되기 때문에 각각의 막대 각도를 90도에서 빼주어 막대와 동일한 각도로 표현하는 각도를 산출  
 mutate(angle = 90 - (id-0.5)/n() \* 360) |>  
 ## 전체 막대중 반의 문자열은 거꾸로 표기되는 것을 방지하기 위해 다시 180도를 돌려줌   
 mutate(angle1 = case\_when(  
 id >= n()/2 ~ angle + 270,   
 id < n()/2 ~ angle -90,   
 ))  
  
df\_취업통계\_전체 <- df\_취업통계 |>   
 summarise(## 백분률인 취업률은 그 자체로 합계나 평균을 낼 수 없으니 각 그룹별로 재계산  
 취업률 = sum(취업자\_합계\_계) / (sum(졸업자\_계) - (sum(진학자\_계)+sum(입대자)+sum(취업불가능자\_계)+sum(외국인유학생\_계)+sum(제외인정자\_계)))) |>  
 select(취업률) |>  
 pull()

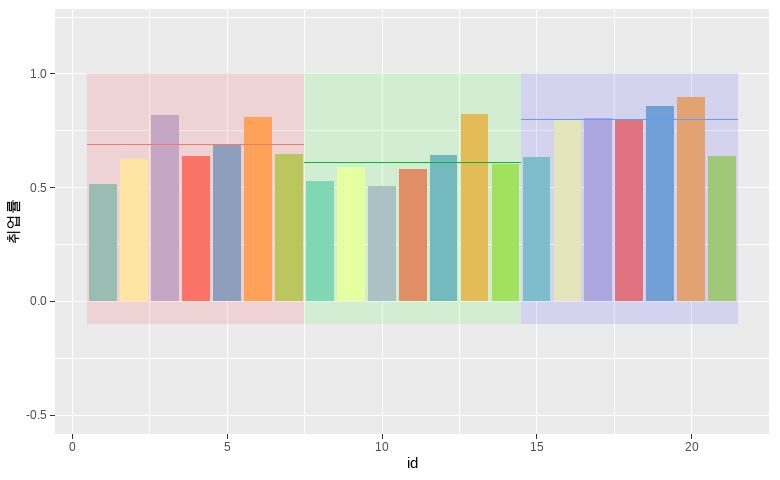
데이터가 준비됐다면 기본 막대 그래프를 그려보겠다. 앞서 설명한 바와 같이 기본 그래프는 마지막에 계열 이름과 데이터 값을 geom\_text()로 표현하기 위해 dodge형 막대 그래프를 사용하지 않고 일련 번호로 각각 개별로 구분된 막대 그래프를 그려준다. 그리고 도넛 차트의 형태로 원의 가운데가 뚫린 형태로 그려주어야 하기 때문에 그래프의 하단에 약간의 여백을 준다.

## 그래프에 사용할 폰트를 설정  
font\_add('NanumBarunGothic', 'c:/windows/fonts/NanumBarunGothic.ttf')  
  
p\_circular <- df\_취업통계\_계열별 |>  
 ggplot() +   
 ## X축을 id, Y축을 취업률, fill을 대계열로 매핑한 geom\_col 레이어 생성  
 geom\_col(aes(x = id, y = 취업률, fill = 대계열), position = 'dodge', show.legend = F)+   
 ## 도넛 차트의 형태로 생성할 수 있도록 X축과 Y축의 범위를 설정  
 scale\_y\_continuous(limits = c(-0.5, 1.2)) +  
 scale\_x\_continuous(limits = c(0.5, 21.5)) +  
 ## 색 팔레트를 설정  
 scale\_fill\_brewer(palette = 'Set3')  
  
p\_circular



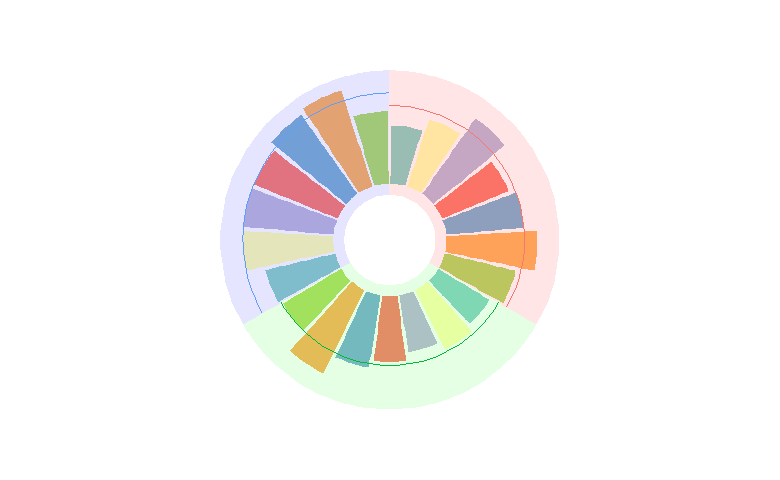
앞의 그래프를 dodge 형 막대 그래프로 그리지 않았가 때문에 각 학위 과정별 그룹간의 구분이 되지 않는다. 이를 구분하기 위해 각 그룹별 사각형으로 구분해주도록 하겠다. 이는 annotate()를 사용해 주석 레이어로 다음과 같이 그려주도록 하겠다. 그리고 추가적으로 각 학위과정별 평균값은 선으로 각각 그려준다.

p\_circular <- p\_circular +  
 annotate(xmin = 0.5, xmax = 7.5, ymin = -0.1, ymax = 1, alpha = 0.1, geom = 'rect', fill = 'red') +  
 annotate(xmin = 7.5, xmax = 14.5, ymin = -0.1, ymax = 1, alpha = 0.1, geom = 'rect', fill = 'green') +  
 annotate(xmin = 14.5, xmax = 21.5, ymin = -0.1, ymax = 1, alpha = 0.1, geom = 'rect', fill = 'blue') +  
 geom\_segment(data = df\_취업통계\_과정별, aes(x = 0.5+((id-1)\*7), xend = 7.5+((id-1)\*7), y = 취업률, yend = 취업률, color = as.factor(id)), inherit.aes = F, show.legend = F)  
  
p\_circular



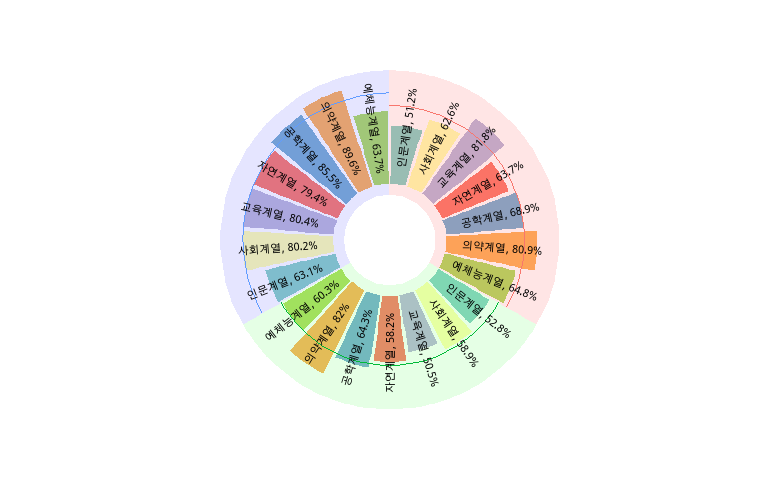
이제 기본 그래프가 완성되었으니 둥글게 말아주도록 하겠다. 앞의 도넛 차트에서 사용했던 coord\_polar()를 사용해도 되겠지만 마지막에 둥근 형태의 문자열을 사용하기 위해 geomtextpath 패키지의 coord\_curveploar()를 사용하여 극 좌표계를 적용하였다. 그리고 전체적인 테마를 없애기 위해 theme\_void()를 적용하였다.

library(geomtextpath)  
  
p\_circular <- p\_circular +  
 coord\_curvedpolar() +  
 theme\_void()  
  
p\_circular



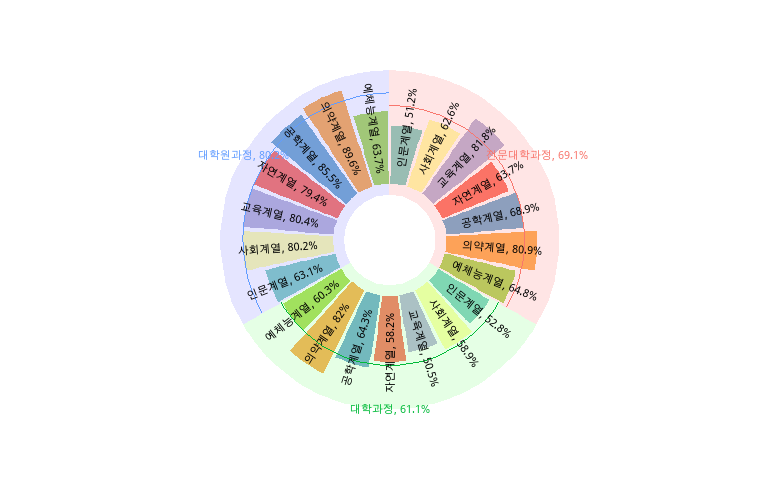
둥글게 그려진 막대 그래프에 데이터 계열과 데이터 값을 표기하도록 하겠다. 앞서 전처리 과정에서 계산한 문자열 각도를 angle에 매핑하여 각각의 문자열의 각도를 설정한다.

## df\_취업통계\_계열별 데이터프레임에서 x축은 id, y축은 0.5, 라벨은 대계열이름과 취업률을 붙이고 anble은 앞에서 설정한 angle1을 사용하여 텍스트를 표기해주고 다른 미적요소를 설정  
p\_circular <- p\_circular +  
 geom\_text(data = df\_취업통계\_계열별, aes(x=id, y=0.5, label=paste0(대계열, ', ', round(취업률\*100, 1), '%'), angle= angle1), color="black", hjust = 0.5, size = 3) +   
 theme(text = element\_text(family = 'NanumBarunGothic'))  
  
p\_circular



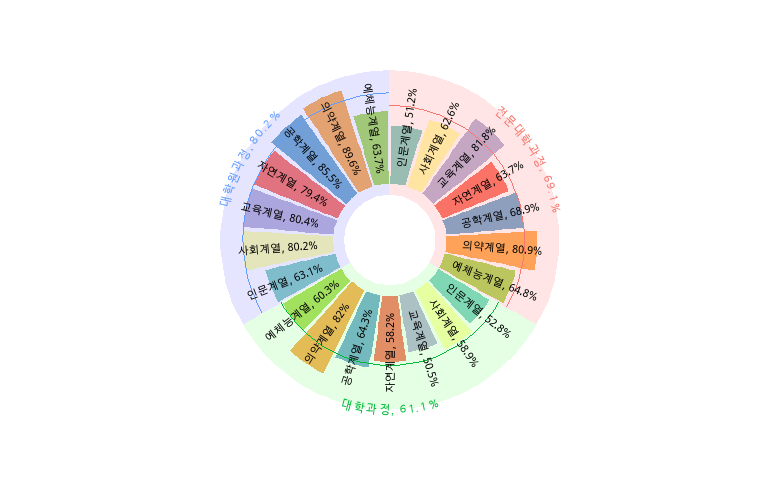
이제 전체 학위과정 이름과 평균값을 표시해주도록 하겠다. geom\_text()를 사용하면 다음의 그림과 같이 수평으로 길게 표시되는 문자열이 추가된다.

p\_circular +  
 geom\_text(data = df\_취업통계\_과정별,aes(x = 4+((id-1)\*7), y = 1.0, label = paste0(과정구분, ', ', round(취업률\*100, 1), '%'), color = as.factor(id)), inherit.aes = F, show.legend = F, size = 3)



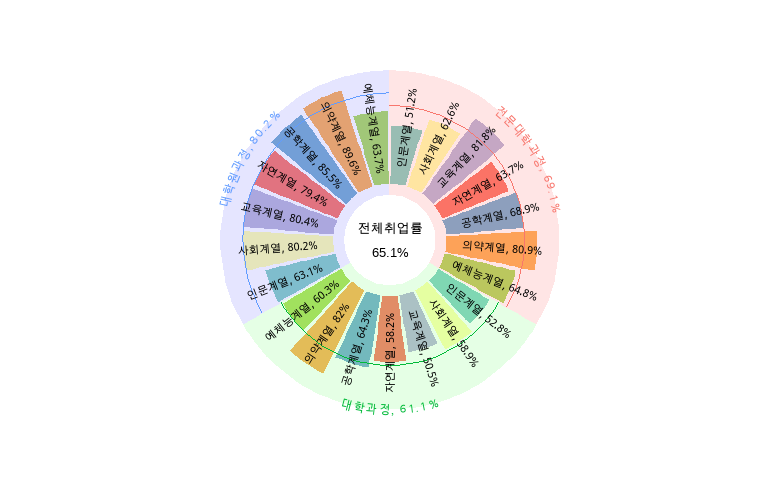
수평으로 길게 표시되는 문자열을 사용할 수 있겠지만 일부 문자열이 겹치고 원형 막대 그래프이기 때문에 이 보다는 둥글게 그려지는 문자열을 사용하는 편이 더 보기가 좋을 듯 하다. 이를 위해서는 geomtextpath패키지의 geom\_textpath()를 사용하도록 하겠다. 이 함수는 앞서 극 좌표계를 설정할때 사용했던 coord\_curvedpolar() 좌표계 위에서 원점을 중심으로 둥글게 문자열을 생성하는 함수이다.

p\_circular <- p\_circular +  
 geom\_textpath(data = df\_취업통계\_과정별,aes(x = 4+((id-1)\*7), y = 1.0, label = paste0(과정구분, ', ', round(취업률\*100, 1), '%'), color = as.factor(id)), show.legend = F, rich = TRUE, family = 'NanumBarunGothic', size = 3)  
  
p\_circular



마지막으로 전체 취업률을 원형 막대 그래프의 중간에 표기해 준다. 원형 막대 그래프는 극좌표계를 사용했기 때문에 그래프의 중앙 위치를 찾을 수 없다. 따라서 ggplot2의 geom\_text()를 사용할 수 없어 grid 패키지의 grid.text()를 사용한다.

p\_circular  
  
grid.text('전체취업률\n65.1%', x = 0.5, y = 0.5, just = c("center", "center"),  
 gp = gpar(col = "black", fontsize = 10))



# ggplot2와 지도의 병합

목표 : 사각형으로 간략화한 지도의 각각 사각형 안에 지역별 ggplot2 분할 그래프를 위치하여 ggplot2와 지도를 병합함  
활용 시각화 : 지도 시각화, ggplot2  
분할 그래프 사용 패키지 : ggplot2, geofacet

지도를 시각화할 때 지형의 원형 그대로를 사용하는 경우가 대부분이다. 하지만 앞 장에서 본 바와 같이 지도를 육각형이나 사각형으로 단순화하여 데이터를 표현하는 경우도 있다. 이렇게 표현하는 시각화는 대부분 지역을 표현하는 다각형의 내부 색을 사용하여 데이터를 표현하는 경우가 대부분이지만 ggplot2의 facet\_\*()으로 분할된 그래프를 지도의 형태로 표현할 수 있다. 이러한 시각화는 geofacet 패키지를 사용하면 생각보다 쉽게 생성할 수 있다.

geofacet 패키지는 ggplot2에서 사용하는 facet\_\*()을 지리적 정보를 기반하여 표현하는 facet\_geo()를 제공한다. facet\_geo()는 ggplot2의 facet\_\*()과 동일한 용도로 사용되기 떄문에 ggplot2의 facet\_\*()의 사용법과 동일하다.

facet\_geo(facets, …, grid = “us\_state\_grid1”, label = NULL, move\_axes = TRUE)  
- facets : 분할 그래프를 만들 분할 변수  
- grid : 지형 레이아웃으로 사용할 그리드 데이터프레임  
- label : 분할 그래프 라벨로 사용할 문자열 벡터  
- … : facet\_wrap()에서 사용하는 매개변수

facet\_geo()를 사용할 때 가장 중요한 것이 바로 grid 매개변수에 사용할 지형 레이아웃 데이터프레임이다. 이 데이터프레임은 두 가지 방법을 사용할 수 있다.

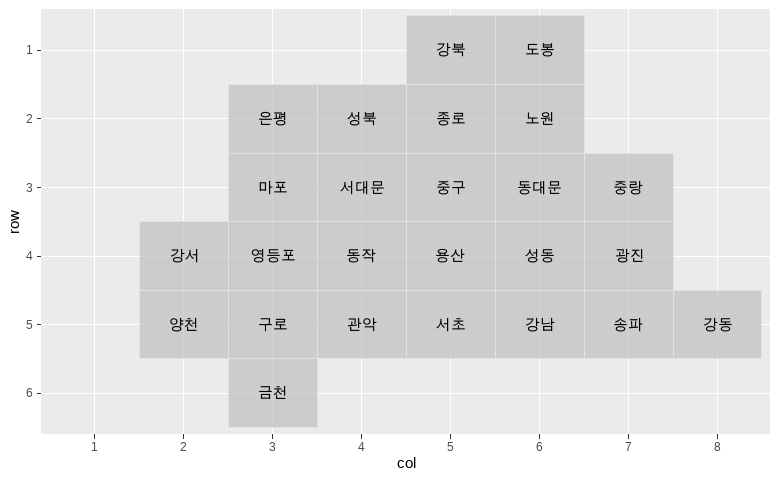
첫 번째 방법은 사전에 정의된 그리드 데이터프레임을 사용하는 것이다. geofacet 패키지의 개발 깃헙에서는 다양한 지형에 대한 그리드 데이터 프레임을 제공하고 있다. 이 중 우리나라 지형에 대한 그리드 데이터프레임도 제공되고 있는데 get\_grid\_names()을 실행하면 현재 사용가능한 그리드 데이터프레임을 확인할 수 있다.

if(!require(geofacet)) {  
 install.packages('geofacet')  
 library(geofacet)  
}  
  
grid\_name <- get\_grid\_names()  
  
grid\_name[str\_detect(grid\_name, 'seoul')]

## [1] "kr\_seoul\_district\_grid1"

geofacet에서 제공하는 서울의 구에 대한 그리드 시각화는 다음과 같다.

grid\_preview("kr\_seoul\_district\_grid1")



두 번째는 자신이 원하는 그리드 데이터프레임을 만들어서 사용할 수 있다. 그리드 데이터프레임을 만들기 위해서는 다음 네 개의 열을 가진 데이터프레임이 필요하다.

| 열 이름 | 내용 |
| --- | --- |
| row | facet이 위치할 행의 번호 |
| col | facet이 위치할 열의 번호 |
| code | 해당 facet의 고유코드 |
| name | 해당 facet의 이름 |

자신이 원하는 그리드 데이터프레임을 생성하기 위해서 각 셀을 하나하나 설정해서 만들수 있지만 geofacet에서 그리드 데이터프레임을 쉽게 만들수 있는 도구를 제공한다. 이 도구는 자바 스크립트로 작성된 프로그램으로 grid\_design()을 호출하여 사용할 수 있다. 이 프로그램을 사용하여 그리드의 위치를 변경할 수 있고 셀을 추가하거나 삭제할 수 있다. 그리드의 편집아 완성되면 왼쪽 하단의 R코드를 복사하여 그리드 데이터프레임을 만들어 줄 수 있다.

grid\_design(kr\_seoul\_district\_grid1)

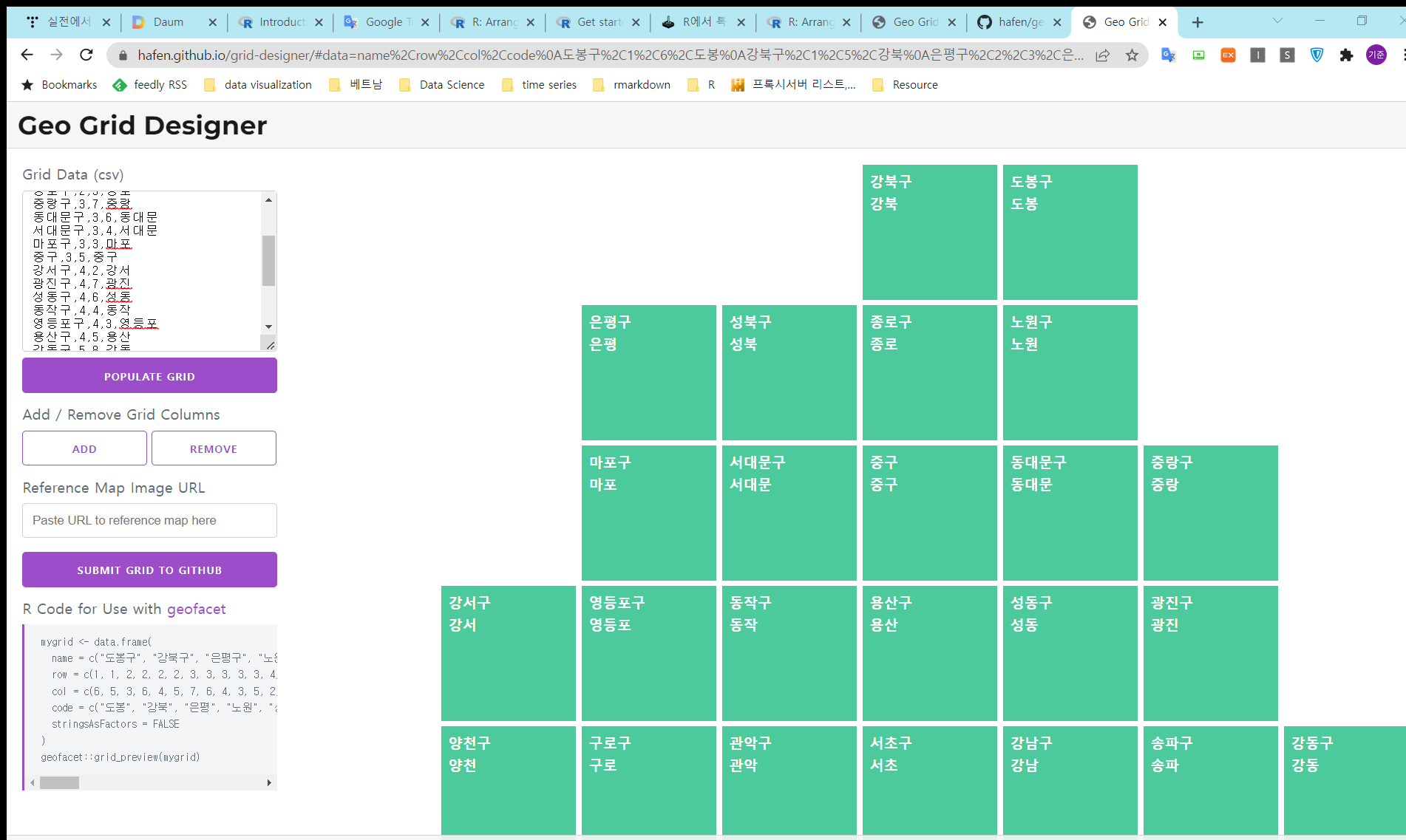


그림 12-1. grid\_design 실행화면

다음은 geofacet에서 제공하는 ’kr\_seoul\_district\_grid1`의 그리드 데이터프레임을 생성하는 코드이다.

mygrid <- data.frame(  
 name = c("도봉구", "강북구", "은평구", "노원구", "성북구", "종로구", "중랑구", "동대문구", "서대문구", "마포구", "중구", "강서구", "광진구", "성동구", "동작구", "영등포구", "용산구", "강동구", "양천구", "송파구", "강남구", "관악구", "구로구", "서초구", "금천구"),  
 row = c(1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 6),  
 col = c(6, 5, 3, 6, 4, 5, 7, 6, 4, 3, 5, 2, 7, 6, 4, 3, 5, 8, 2, 7, 6, 4, 3, 5, 3),  
 code = c("도봉", "강북", "은평", "노원", "성북", "종로", "중랑", "동대문", "서대문", "마포", "중구", "강서", "광진", "성동", "동작", "영등포", "용산", "강동", "양천", "송파", "강남", "관악", "구로", "서초", "금천"),  
 stringsAsFactors = FALSE  
)

그리드 데이터프레임이 완성되었다면 facet\_geo()를 사용하여 서울 지역의 지형에 따른 분할 그래프를 그릴 수 있다. 분할 그래프를 그리기 위해 서울시의 구별 대학 재적 학생수 데이터를 사용하도록 하겠다.

df\_행정구역 <- read\_excel('c:/R/git/datavisualization/chap10/2021\_행정구역별 학과수 및 학년별 재적학생수.xlsx',   
 ## 'sheet0' 시트의 데이터를 불러오는데,  
 sheet = 'Sheet0',  
 ## 앞의 3행을 제외하고  
 skip = 3,   
 ## 첫번째 행은 열 이름이 아님을 설정  
 col\_names = FALSE,   
 ## 열의 타입을 설정, 처음 4개는 문자형으로 다음 39개는 수치형으로 설정  
 col\_types = c(rep('text', 4), rep('numeric', 39)))  
  
## 읽어온 데이터에서 '소계', '전체'에 해당하는 데이터는 제거하고 '서울'데이터만 필터링해서 필요한 열만 남김  
df\_행정구역 <- df\_행정구역 |> filter(...1 == '서울', ...2 != '소계', ...3 != '전체', ...4 == '소계') |>  
 select(1, 2, 3, 4, 5, 10)  
  
## 열 이름을 적절히 설정  
names(df\_행정구역) <- c('시도', 'code', '과정구분', '학제구분', '대학수', '재적학생수')  
  
## 구이음 열에서 '서울 ' 문자열을 ''으로 치환  
df\_행정구역$code <- gsub('서울 ', '', df\_행정구역$code)

font\_add('NanumBarunGothic', 'c:/windows/fonts/NanumBarunGothic.ttf')  
font\_add('NanumBarunGothicBold', 'c:/windows/fonts/NanumBarunGothicBold.ttf')  
  
df\_행정구역 |>  
 ggplot() +  
 geom\_col(aes(x = 과정구분, y = 재적학생수, fill = 과정구분)) +  
 facet\_geo(~code, grid = mygrid) +   
 theme(  
 strip.background = element\_blank(),   
## axis.line = element\_blank(),   
 axis.text = element\_blank(),  
 axis.ticks = element\_blank(),  
 strip.text = element\_textbox\_simple(family = 'NanumBarunGothicBold',   
 size = 10,  
 color = "white", fill = "dodgerblue4", box.color = "#4A618C",  
 halign = 0.5, linetype = 1, r = unit(5, "pt"), width = unit(1, "npc"),  
 padding = margin(2, 0, 1, 0), margin = margin(3, 3, 3, 3)),   
# panel.border = element\_rect(color = '#4A618C', fill = NA),   
# panel.background = element\_blank()  
 ) +  
 labs(x = NULL, y = NULL)



# 각 분야별 Top 3

library(tidyverse)  
library(showtext)  
showtext\_auto()  
  
font\_add('NanumBarunGothic', 'c:/windows/fonts/NanumBarunGothic.ttf')  
font\_add('NanumBarunGothicBold', 'c:/windows/fonts/NanumBarunGothicBold.ttf')  
  
  
  
df\_취업통계\_계열별 <- df\_취업통계 |>   
 group\_by(과정구분, 대계열, 중계열, 소계열) |>   
 summarise(학과수 = n(), 졸업자 = sum(졸업자\_계),   
 취업자 = sum(취업자\_합계\_계),   
 교외취업자 = sum(취업자\_교외취업자\_계),   
 교내취업자 = sum(취업자\_교내취업자\_계),   
 해외취업자 = sum(취업자\_해외취업자\_계),   
 농림어업종사자 = sum(취업자\_농림어업종사자\_계),   
 개인창작활동종사자 = sum(취업자\_개인창작활동종사자\_계),   
 일인창사업자 = sum(`취업자\_1인창(사)업자\_계`),   
 프리랜서 = sum(취업자\_프리랜서\_계),   
 진학자 = sum(진학자\_계),   
 입대자 = sum(입대자),  
 취업불가능자 = sum(취업불가능자\_계),   
 외국인유학생 = sum(외국인유학생\_계),   
 제외인정자 = sum(제외인정자\_계),   
 기타 = sum(기타\_계),   
 미상 = sum(미상\_계),   
 ## 백분률인 취업률은 그 자체로 합계나 평균을 낼 수 없으니 각 그룹별로 재계산  
 취업률 = 취업자 / (졸업자 - (진학자+입대자+취업불가능자+외국인유학생+제외인정자))) |>  
 ## 계열의 표시 순서를 설정하기 위해 레벨을 재조정  
 mutate(대계열 = fct\_relevel(대계열, '인문계열', '사회계열', '교육계열', '자연계열', '공학계열', '의약계열', '예체능계열'))  
  
df\_취업통계\_과정별 <- df\_취업통계 |>   
 group\_by(과정구분) |>   
 summarise(학과수 = n(), 졸업자 = sum(졸업자\_계),   
 취업자 = sum(취업자\_합계\_계),   
 교외취업자 = sum(취업자\_교외취업자\_계),   
 교내취업자 = sum(취업자\_교내취업자\_계),   
 해외취업자 = sum(취업자\_해외취업자\_계),   
 농림어업종사자 = sum(취업자\_농림어업종사자\_계),   
 개인창작활동종사자 = sum(취업자\_개인창작활동종사자\_계),   
 일인창사업자 = sum(`취업자\_1인창(사)업자\_계`),   
 프리랜서 = sum(취업자\_프리랜서\_계),   
 진학자 = sum(진학자\_계),   
 입대자 = sum(입대자),  
 취업불가능자 = sum(취업불가능자\_계),   
 외국인유학생 = sum(외국인유학생\_계),   
 제외인정자 = sum(제외인정자\_계),   
 기타 = sum(기타\_계),   
 미상 = sum(미상\_계),   
 ## 백분률인 취업률은 그 자체로 합계나 평균을 낼 수 없으니 각 그룹별로 재계산  
 취업률 = 취업자 / (졸업자 - (진학자+입대자+취업불가능자+외국인유학생+제외인정자))) |>  
 mutate(rank = min\_rank(-취업률),   
 pos = case\_when(  
 과정구분 == '대학원과정' ~ 1,  
 과정구분 == '대학과정' ~ 2,  
 과정구분 == '전문대학과정' ~ 3  
 ))  
   
  
  
  
  
df\_취업통계\_소계열별 <- df\_취업통계\_계열별 |> filter(학과수 >= 3) |>  
 group\_by(과정구분) |>  
 top\_n(n = 10, wt = 취업률) |>  
 mutate(소계열순위 = min\_rank(-취업률)) |>  
 mutate(과정구분 = fct\_relevel(과정구분, '대학원과정', '대학과정', '전문대학과정'))  
   
  
  
df\_취업통계\_소계열별 |>  
 ggplot() +   
 geom\_col(aes(x = 과정구분, y = 취업률, fill = as.factor(소계열순위)), position = position\_dodge2(width = 0.9, padding = 0.1, reverse = TRUE), show.legend = FALSE, size = 0) +   
 geom\_text(aes(x = 과정구분, y = 0.01, fill = as.factor(소계열순위), label = 소계열), position = position\_dodge2(width = 0.9, padding = 0.1, reverse = TRUE), hjust = 0, show.legend = FALSE) +  
 geom\_text(aes(x = 과정구분, y = 취업률, fill = as.factor(소계열순위), label = paste0(round(취업률, 3)\*100, '%')), position = position\_dodge2(width = 0.9, padding = 0.1, reverse = TRUE), hjust = 1.1, show.legend = FALSE) +  
 geom\_segment(aes(x = 0.55, xend = 1.45, y = -0.01, yend = -0.01), color = 'goldenrod1') +  
 geom\_segment(aes(x = 1.55, xend = 2.45, y = -0.01, yend = -0.01), color = 'coral2') +  
 geom\_segment(aes(x = 2.55, xend = 3.45, y = -0.01, yend = -0.01), color = 'mediumpurple1') +  
 scale\_fill\_manual(values = c("goldenrod1", "mediumpurple1", "coral2", rep("gray70", 7))) +  
 scale\_x\_discrete(labels = c('대\n학\n원\n과\n정', '대\n학\n과\n정', '전\n문\n대\n학\n과\n정'),   
 expand = expansion(add = c(0, 0))) +  
 scale\_y\_continuous(expand = expansion(add = c(0.005, 0.05)), label = scales::percent) +  
 theme(text = element\_text(family = 'NanumBarunGothic', size = rel(4)),   
 axis.text.y = element\_text(hjust = 0, size = 15),   
 axis.ticks.y = element\_line(unit(0, 'mm')),   
 panel.background = element\_blank()) +  
 labs(x = NULL, y = NULL) +   
 geom\_segment(data = df\_취업통계\_과정별 |> filter(과정구분 == '대학원과정'), aes(x = 0.55, xend = 1.45, y = 취업률, yend = 취업률), color = 'goldenrod1') +   
 geom\_segment(data = df\_취업통계\_과정별 |> filter(과정구분 == '대학과정'), aes(x = 1.55, xend = 2.45, y = 취업률, yend = 취업률), color = 'coral2') +   
 geom\_segment(data = df\_취업통계\_과정별 |> filter(과정구분 == '전문대학과정'), aes(x = 2.55, xend = 3.45, y = 취업률, yend = 취업률), color = 'mediumpurple1') +   
 geom\_label(data = df\_취업통계\_과정별 |> filter(과정구분 == '대학원과정'), aes(x = 1, y = 취업률, label = round(취업률, 3)\*100), color = 'goldenrod1', fill = 'white') +   
 geom\_label(data = df\_취업통계\_과정별 |> filter(과정구분 == '대학과정'), aes(x = 2, y = 취업률, label = round(취업률, 3)\*100), color = 'coral2', fill = 'white') +   
 geom\_label(data = df\_취업통계\_과정별 |> filter(과정구분 == '전문대학과정'), aes(x = 3, y = 취업률, label = round(취업률, 3)\*100), color = 'mediumpurple1', fill = 'white') +   
 coord\_flip()

