1. 표의 시각화

데이터 시각화를 한다고 하면 대부분 떠올리는 작업이 그래프를 그리는 작업이다. 그래서 데이터 시각화를 시작할 때는 먼저 데이터를 대상으로 할지, 어떤 그래프를 써서 데이터를 예쁘고 직관적으로 표현할지를 고민하게 된다. 하지만 데이터 시각화 결과가 들어갈 보고문서에는 대부분 그래프와 함께 표가 들어가는 경우가 대부분이지만 데이터를 설명하는 데이터 시각화를 해야할 때 많은 사람들은 표까지 생각하지는 않는 듯하다. 인터넷 상의 표의 정의[[1]](#footnote-20)를 보면 ’시각적 의사소통과 자료의 정렬 양식’으로 나와 있다. 결국 표도 데이터 시각화의 일부인 것이다.

표는 데이터를 직접적으로 표현한다는 점에서 그래프나 플롯과는 다르다. 데이터를 직접 표현하기 때문에 데이터를 정확하게 표현하지만 데이터의 전체적인 흐름이나 분포를 알아보기에는 적절치 않다. 하지만 그래프나 플롯은 정확한 데이터를 알아보기 힘들다는 점에서 표와 그래프, 플롯은 상호 보완적으로 사용되어야 한다.

# 표(table)란?

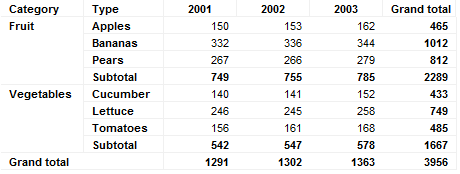
표는 행과 열로 이루어진 데이터의 모음이다. 행은 각각의 개별 사례2나 개별 사례를 적절히 그루핑한 요약된 데이터가 표현된다. 열은 각각의 사례를 설명하기 위해 필요한 속성이 나열된다. 행과 열은 이 만나는 곳이 사용자가 알고 싶어하는 데이터가 위치하고 이 곳을 칸, 혹은 셀이라고 한다.

표는 그 구성 방법에 따라 1차원 표와 다차원 표로 구성된다. 1차원 표는 단순히 데이터의 나열인 경우를 말한다. 이런 테이블을 ’단순 표(simple table)’이라고도 한다. 다차원 표는 1차원 표를 적절한 변환(추상화)하여 표에서 제공하는 몇가지 차원의 정보를 모두 가져야 데이터를 해석할 수 있는 표를 말한다. 다차원 표 중에서 가장 흔하게 보이는 표가 2차원 표, 교차표(cross table)이다. 2차원 표는 행(X축)과 열(Y축)의 정보를 가지고 해당 칸(셀)의 정보를 해석할 수 있다.

아래의 그림은 1차원 표와 2차원 표의 예를 보이고 있다.



<https://docs.tibco.com/pub/spotfire/6.5.2/doc/html/images/cross_example_table.png>

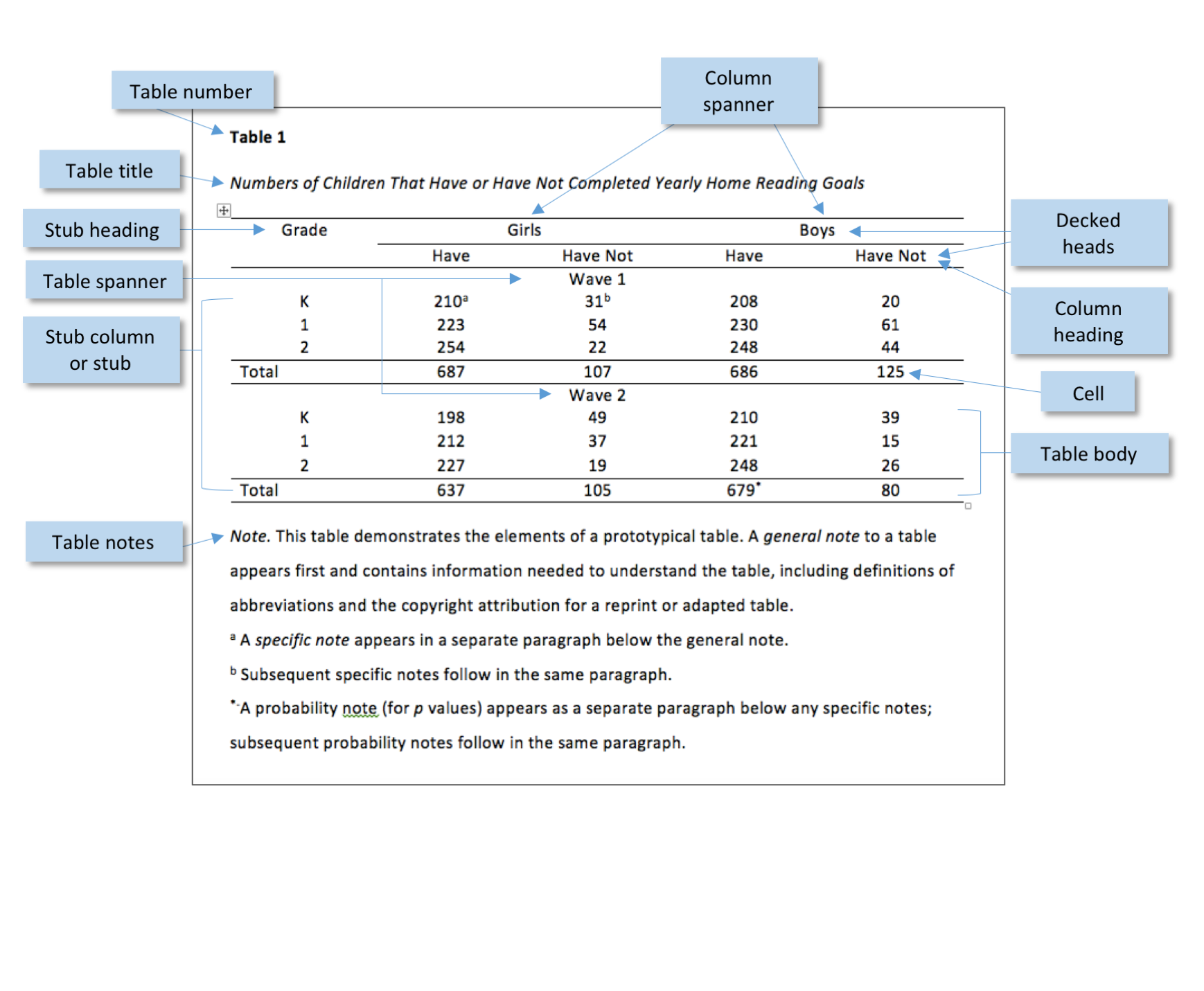


<https://docs.tibco.com/pub/spotfire/6.5.2/doc/html/images/cross_example_cross_table.png>

위의 그림에서 보면 1차원 표의 경우는 데이터를 설명하는 속성이 위쪽에만 설정되어 있다. 하지만 2차원 표의 경우는 데이터를 설명하는 속성이 위쪽과 왼쪽에 모두 설정되어 있다. 특정 칸의 데이터를 해석하기 위해서는 행과 열의 속성 정보를 모두 알아야 한다는 것이다.

1차원 표의 장점은 데이터를 원본 차원에서 확인 할 수 있다는 점이다. 반면 데이터의 행이 길어질 수 있어 보고서에 수록하는 것인 적절치 않을 때가 있다. 하지만 다차원 표의 가장 큰 장점은 데이터를 요약하고 구조화하였기 때문에 대량의 데이터의 특성을 간략히 표현했다는 점이다. 하지만 데이터를 요약하는 방식에 따라 전달되는 정보가 제한적일 수 있다는 단점이 있다.

표의 구성 방법과 요소는 사용자가 표현하고 싶은 데이터와 형태에 따라 표현방식이 매우 다르기 때문에 어느 하나로 정의하기가 어렵다. 다음 그림은 논문의 작성에서 대표적으로 사용되는 미국 심리학회의 표 구성 가이드라인이다.



<https://libapps.s3.amazonaws.com/customers/836/qu/e1d262992c7ea1e2aa109305faf24d59.png>

모든 표를 APA 가이드라인에 맞춰 그릴 수는 없겠지만 APA 가이드라인을 보면 표에 꼭 들어가야 할 몇가지 요소들을 알 수 있다.

* 제목(Title) : 표에서 표현하고 있는 데이터를 대표하는 제목
* 헤드(Heading) : 표에서 제시하고 있는 사례들의 속성값들에 대한 이름이 표현된 표 가로축의 맨 위줄
* 스텁(Stub) : 표에서 사례를 표현하는 세로축의 묶음
* 스패너(Spanner) : 표의 특성을 표현하는 가로축 중에 유사한 특성의 묶음
* 몸체(Body) : 표의 헤드와 스텁으로 감싸고 있는 데이터가 표현된 셀들이 표현된 부분

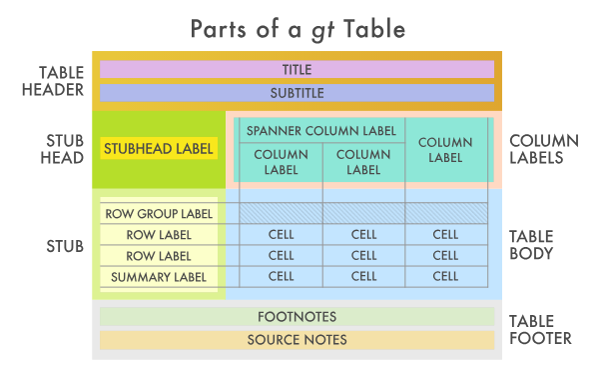
실무에서 표를 그릴 때 가장 많이 사용하는 툴은 아마도 MS-Excel일 것이다. SpreadSheet로 거의 유일하게(?) 살아남은 이 툴은 표를 그리는 WYSYWYG(What You See IS Wat You Get) 툴로 거의 모든 Spreadsheet를 없애버렸다고 해도 과언이 아닐 듯 하다. 어쨌던 엑셀은 표를 그리는데 매우 특화된 툴임에는 틀림없고 초보자들도 쉽게 사용할 수 있지만 사용하다보면 몇가지 단점을 만날 수 있다.

가장 큰 단점이 반복된 작업을 하기가 어렵다는 점이다. 이는 WYSIWIG 툴들이 가지는 공통된 단점이다. 표를 쉽게 만들 수 있지만 유사한 표를 다시 만들어야 하는 경우 반복된 작업을 수행해야하고 반복된 작업을 꼼꼼히 기록해두지 않으면 동일한 표를 만들기 어렵다. 또 하나의 단점이 기초 데이터의 구조가 업데이트 되면 표를 다시 그려야 하는 경우가 발생한다는 점이다. 기초 데이터의 열 이름이 바뀌거나 열이 추가, 삭제되면 이 데이터로부터 파생된 표가 정상적으로 표현되지 않는다는 것이다. 이러한 문제는 엑셀을 사용해 본 사용자라면 한번쯤, 아니 자주 겪는 문제일 것이다.

이제 엑셀에서 탈피해서 R에서 표를 그려보자. R에서도 아주 훌륭한, 그리고 예쁘게 표를 그릴 수 있는 다양한 방법을 제공한다. R로 표를 그려보면 반복된 작업을 할 필요가 없다는 점에서, 기초 데이터의 업데이트가 발생해도 바로 반영할 수 있다는 점, 표를 세세하게 다룰 수 있다는 점에서 엑셀로 다시 돌아갈 수 없을 수도 있다.

# gt 패키지

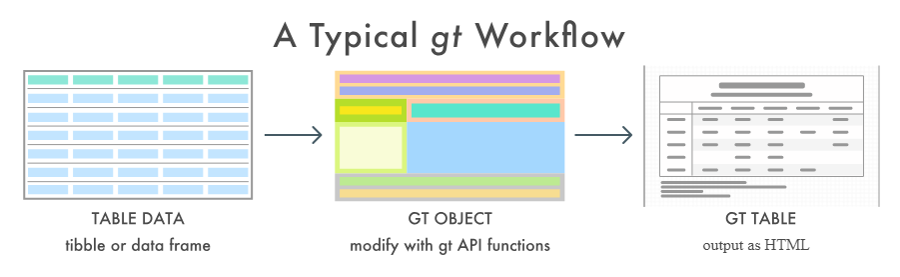
R에서 표를 만들기 위해 사용되는 패키지 중에 하나가 gt 패키지이다. gt 패키지는 표의 세부 구성들을 상세히 구분하고 이들을 구조적으로 조화롭게 구성시켜서 표를 만드는 다양한 함수와 매개 변수들을 제공한다. gt 패키지에서 사용하는 표의 세부 파트는 다음의 그림과 같다.



<https://gt.rstudio.com/reference/figures/gt_parts_of_a_table.svg>

표의 구성을 보면 앞에서 살펴보았던 APA 가이드라인상의 표와 유사한 형태를 보인다. 이와 같이 표를 세부적인 파트로 구분하고 이들을 각각 설정함으로써 표를 만들 수 있다.

gt 패키지에서 권장하는 표 생성 방식은 다음의 그림과 같다



<https://gt.rstudio.com/reference/figures/gt_workflow_diagram.svg>

우선 표를 만들기 위한 기초 데이터를 생성해야 한다. gt 패키지로 생성되는 표는 tidyverse 방식을 준용하여 생성하는 것이 편하다. 따라서 gt 표를 생성하기 위해 만드는 기초 데이터도 tidyverse 의 tibble 이나 R의 기초 데이터 셋인 data.frame 으로 생성한다. 이렇게 생성된 기초 데이터 셋은 gt 객체로 변환되고 gt 패키지에서 제공되는 다양한 함수와 매개변수를 설정하여 사용자가 원하는 형태의 표로 만들게 된다. 마지막으로 이 표를 다른 문서에서 활용하기 위해 html, pdf 등의 문서로 변환하여 활용한다.

우선 gt 패키지를 사용하기 위해서는 gt 패키지를 설치하고 로딩하여야 한다.

if(!require('gt')) {  
 install.packages('gt')  
 library(gt)  
}

이번 장에서 사용할 데이터는 앞 장에서 생성한 df\_취업통계 데이터를 사용하겠다. 최종 생성될 표는 각각의 과정구분에 따라 편성된 대계열 학과들에 대한 취업 정보를 요약한 표이다. 표에서는 각각의 열을 과정구분로 구분(Stub)하고, 특성으로 표현되는 열은 졸업자, 취업자, 취업률 등 졸업 후 상황별 학생수를 표기하는데 취업 상세 학생과 비취업 상세 학생으로 열을 묶어(Spanner)하고 소계와 총계가 나타나는 표를 그리겠다.

## 데이터 전처리

우선 목표한 표를 만들기 위해 가진 데이터를 전처리하는 과정이 필요하다. 우선 추가적으로 필요한 열을 생성한다. 개별 학과별로 저장된 원본 데이터를 대계열별, 과정구분별로 요약(Summarise)하고 표현될 순서를 설정하기 위해 팩터로 레벨을 설정하도록 하겠다.

## df\_취업통계를 전처리하여 df\_gt로 저장  
df\_gt <- df\_취업통계 |>  
 ## 과정구분과 대계열로 그루핑  
 group\_by(과정구분, 대계열) |>  
 ## 각각의 열단위로 합계값으로 요약된 열 생성   
 summarise(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))) |>  
 ## 최종결과는 그룹 설정을 제거  
 ungroup() |>   
 arrange(과정구분)  
  
head(df\_gt, 10)

## # A tibble: 10 x 19  
## 과정구분 대계열 졸업자 취업자 교외취업자 교내취업자 해외취업자 농림어업종사자  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 대학과정 공학~ 83953 47789 44414 1016 179 5  
## 2 대학과정 교육~ 19733 9258 8156 239 6 1  
## 3 대학과정 사회~ 90304 47721 42979 1198 242 23  
## 4 대학과정 예체~ 36247 19104 13607 619 68 3  
## 5 대학과정 의약~ 25386 20014 19570 126 7 0  
## 6 대학과정 인문~ 37864 16681 13722 772 210 5  
## 7 대학과정 자연~ 37555 17648 15454 616 36 46  
## 8 대학원~ 공학~ 13978 9348 8766 396 35 0  
## 9 대학원~ 교육~ 2694 1763 1550 69 0 1  
## 10 대학원~ 사회~ 8178 4141 3724 90 2 3  
## # ... with 11 more variables: 개인창작활동종사자 <dbl>, 일인창사업자 <dbl>,  
## # 프리랜서 <dbl>, 진학자 <dbl>, 입대자 <dbl>, 취업불가능자 <dbl>,  
## # 외국인유학생 <dbl>, 제외인정자 <dbl>, 기타 <dbl>, 미상 <dbl>, 취업률 <dbl>

## 과정구분의 순서를 맞추기 위해 과정구분을 팩터로 설정하고 레벨의 순서를 설정  
df\_gt$과정구분 <- fct\_relevel(df\_gt$과정구분, '전문대학과정', '대학과정', '대학원과정')  
  
## 대계열의 순서를 맞추기 위해 과정구분을 팩터로 설정하고 레벨의 순서를 설정  
df\_gt$대계열 <- fct\_relevel(df\_gt$대계열, '인문계열', '사회계열', '교육계열', '자연계열', '공학계열', '의약계열', '예체능계열')  
  
## 백분률인 취업률은 그 자체로 합계나 평균을 낼 수 없으니 각 소계 그룹별로 재계산하여 dt\_gt\_summary에 저장  
df\_gt\_summary <- df\_gt |> group\_by(대계열) |>   
 summarise(취업률 = sum(취업자) / (sum(졸업자) - (sum(진학자)+sum(입대자)+sum(취업불가능자)+sum(외국인유학생)+sum(제외인정자))))  
  
## 백분률인 취업률은 그 자체로 합계나 평균을 낼 수 없으니 각 총계로 재계산하여 dt\_gt\_grand\_summary에 저장  
df\_gt\_grand\_summary <- df\_gt |>   
 summarise(취업률 = sum(취업자) / (sum(졸업자) - (sum(진학자)+sum(입대자)+sum(취업불가능자)+sum(외국인유학생)+sum(제외인정자))))

## gt 객체 생성하기

gt 패키지를 사용하여 표를 그리려면 먼저 gt 패키지의 gt() 함수를 사용하여 gt 객체를 생성한다.

gt(data, rowname\_col = "rowname", groupname\_col = dplyr::group\_vars(data), ...)  
 - data : 표에 사용할 데이터프레임이나 tibble  
 - rowname\_col : 표 스텁으로 사용할 열 이름  
 - groupname\_col : 행 그룹으로 그루핑할 열 이름

매개변수 없이 gt()만으로도 gt 객체가 생성된다. 하지만 행을 그루핑하여 행 그룹으로 사용할 groupname\_col과 표의 스텁(Stub)으로 사용할 열인 rowname\_col을 설정해주면 표의 전반적인 기본 형태를 볼 수 있기 때문에 가급적이면 gt 객체를 생성할 때 설정해 주는 것이 좋다.

## df\_gt 데이터를 사용해 생성된 gt 객체를 gt\_table1에 저장  
gt\_table1 <- df\_gt |>  
 ## gt()로 rowname\_col을 과정구분, groupname\_col을 대계열로 설정한 gt 객체 생성  
 gt(rowname\_col = '과정구분',   
 groupname\_col = '대계열')  
  
gt\_table1



## 표제 생성하기

이제 표의 각 부분을 설정하는데 먼저 표의 제목부터 설정한다. 표 제목을 설정하기 위해서는 tab\_header()를 사용한다. tab\_header()의 title 매개변수로 표 제목을 설정하고 subtitle을 사용하여 표 부제목을 설정할 수 있다.

gt\_table2 <- gt\_table1 |>  
 ## tab\_header()를 사용하여 제목과 부제목을 설정  
 tab\_header(title = '고등교육기관 취업통계',   
 subtitle = '2021년 전체 고등교육기관 대상')  
  
gt\_table2



## 데이터 포맷 설정

다음으로 각각의 셀에 표현되는 수치값의 형태를 설정한다. gt 객체의 각 셀의 포맷을 설정할 떄는 다음과 같은 fmt\_\*()를 사용한다.

| 함수명 | 설명 |
| --- | --- |
| [fmt\_number()](https://gt.rstudio.com/reference/fmt_number.html) | 숫자형 포맷을 설정 |
| [fmt\_integer()](https://gt.rstudio.com/reference/fmt_integer.html) | 정수형 포맷을 설정 |
| [fmt\_scientific()](https://gt.rstudio.com/reference/fmt_scientific.html) | 과학 표기 포맷을 설정 |
| [fmt\_engineering()](https://gt.rstudio.com/reference/fmt_engineering.html) | 공학 표기 포맷을 설정 |
| [fmt\_percent()](https://gt.rstudio.com/reference/fmt_percent.html) | 백분률 포맷을 설정 |
| [fmt\_currency()](https://gt.rstudio.com/reference/fmt_currency.html) | 통화기호 포맷을 설정 |
| [fmt()](https://gt.rstudio.com/reference/fmt.html) | 사용자 정의형 포맷을 설정 |

fmt\_\*() 의 주요 매개변수는 다음과 같다.

fmt\_\*(data, columns, rows = everything(), decimals = 2, sep\_mark = ",", dec\_mark = ".", use\_seps = TRUE, ...)  
 - data : 표에 사용할 데이터프레임이나 tibble  
 - columns : 포맷을 설정할 열 이름, `c()`를 사용하여 열 이름 벡터나 열 번호를 사용할 수 있고 헬퍼 함수인 `starts\_with()`, `ends\_with()`, `contains()`, `matches()`, `one\_of()`, `num\_range()`, `everything()`으로 열을 선택할 수 있음   
 - decimals : 소수점 아래 표기 자리수 설정  
 - sep\_mark : 천단위 구분자로 사용할 기호 설정  
 - dec\_mark : 소수점 구분자로 사용할 기호 설정  
 - use\_seps : 천단위 구분자를 사용할지를 결정하는 논리값

학생수는 정수로 모두 소수점이하가 필요없고 취업률은 백분율이기 때문에 소수점 한자리까지 표현되는 백분률 포맷으로 설정한다.

gt\_table3 <- gt\_table2 |>   
 fmt\_number(columns = 3:18, decimals = 0, use\_seps = TRUE) |>  
 fmt\_percent(columns = 19, decimals = 1, use\_seps = FALSE)  
  
gt\_table3



gt\_table4 <- gt\_table3 |>   
 tab\_spanner(columns = 5:11, label = '취업 상세') |>  
 tab\_spanner(columns = 12:18, label = '비취업 상세')   
  
gt\_table4



gt\_table5 <- gt\_table4 |>   
 cols\_label(일인창사업자 = '1인창(사)업자'  
 )  
  
gt\_table5



gt\_table6 <- gt\_table5 |>   
 row\_group\_order(  
 groups = c('인문계열', '사회계열', '교육계열', '자연계열', '공학계열', '의약계열', '예체능계열')  
 )  
  
gt\_table6



gt\_table7 <- gt\_table6 |>   
 summary\_rows(  
 groups = T,  
 columns = 3:18,  
 fns = list(  
 소계 = ~sum(.)),  
 formatter = fmt\_number,   
 decimals = 0,  
 use\_seps = TRUE  
 ) |>  
 summary\_rows(  
 groups = '인문계열',  
 columns = 19,  
 fns = list(  
 소계 = ~pull(df\_gt\_summary |> filter(대계열 == '인문계열') |> select(취업률))  
 ),  
 formatter = fmt\_percent,   
 decimals = 1,  
 use\_seps = TRUE  
 ) |>  
 summary\_rows(  
 groups = '사회계열',  
 columns = 19,  
 fns = list(  
 소계 = ~pull(df\_gt\_summary |> filter(대계열 == '사회계열') |> select(취업률))  
 ),  
 formatter = fmt\_percent,   
 decimals = 1,  
 use\_seps = TRUE  
 ) |>  
 summary\_rows(  
 groups = '교육계열',  
 columns = 19,  
 fns = list(  
 소계 = ~pull(df\_gt\_summary |> filter(대계열 == '교육계열') |> select(취업률))  
 ),  
 formatter = fmt\_percent,   
 decimals = 1,  
 use\_seps = TRUE  
 ) |>  
 summary\_rows(  
 groups = '자연계열',  
 columns = 19,  
 fns = list(  
 소계 = ~pull(df\_gt\_summary |> filter(대계열 == '자연계열') |> select(취업률))  
 ),  
 formatter = fmt\_percent,   
 decimals = 1,  
 use\_seps = TRUE  
 ) |>  
 summary\_rows(  
 groups = '공학계열',  
 columns = 19,  
 fns = list(  
 소계 = ~pull(df\_gt\_summary |> filter(대계열 == '공학계열') |> select(취업률))  
 ),  
 formatter = fmt\_percent,   
 decimals = 1,  
 use\_seps = TRUE  
 ) |>  
 summary\_rows(  
 groups = '의약계열',  
 columns = 19,  
 fns = list(  
 소계 = ~pull(df\_gt\_summary |> filter(대계열 == '의약계열') |> select(취업률))  
 ),  
 formatter = fmt\_percent,   
 decimals = 1,  
 use\_seps = TRUE  
 ) |>  
 summary\_rows(  
 groups = '예체능계열',  
 columns = 19,  
 fns = list(  
 소계 = ~pull(df\_gt\_summary |> filter(대계열 == '예체능계열') |> select(취업률))  
 ),  
 formatter = fmt\_percent,   
 decimals = 1,  
 use\_seps = TRUE  
 )  
  
gt\_table7



gt\_table8 <- gt\_table7 |>  
 grand\_summary\_rows(  
 columns = 3:18,  
 fns = list(  
 총계 = ~sum(.)),   
 formatter = fmt\_number,   
 decimals = 0,  
 use\_seps = TRUE  
 ) |>  
 grand\_summary\_rows(  
 columns = 19,  
 fns = list(  
 총계 = ~pull(df\_gt\_grand\_summary |> select(취업률))),   
 formatter = fmt\_percent,   
 decimals = 1,  
 use\_seps = TRUE  
 )  
  
  
gt\_table8



gt\_table9 <- gt\_table8 |>  
 tab\_options(  
 table.font.size = 12,  
 ## 행 그룹 요약 행의 배경색 설정  
 summary\_row.background.color = "#009acd",  
 ## 전체 요약 행의 배경색 설정  
 grand\_summary\_row.background.color = "#00688B",  
 ## 구분(Stub) 행의 헤더 외곽선 스타일 설정  
 stub.border.style = 'solid',   
 ## 구분(Stub) 행의 배경색 설정  
 stub.background.color = '#00BFFF',  
 ## 행 그룹 이름 표현 셀 배경색 설정  
 row\_group.background.color = '#F0F8FF',  
 ## 열 제목(Heading) 배경색 설정  
 heading.background.color = '#104E8B',  
 ## 표 몸체(Body) 수평선 스타일 설정  
 table\_body.hlines.style = 'dashed',   
 ## 표 몸체(Body) 수직선 색깔 설정   
 table\_body.vlines.color = 'grey',  
 ## 표 몸체(BOdy) 수직선 스타일 설정  
 table\_body.vlines.style = 'dashed'  
 )  
  
# gt\_table9 <- gt\_table8 |>  
# tab\_options(  
# table.font.size = 12,  
# ## 행 그룹 요약 행의 배경색 설정  
# summary\_row.background.color = "deepskyblue3",  
# ## 전체 요약 행의 배경색 설정  
# grand\_summary\_row.background.color = "deepskyblue4",  
# ## 구분(Stub) 행의 헤더 외곽선 스타일 설정  
# stub.border.style = 'solid',   
# ## 구분(Stub) 행의 배경색 설정  
# stub.background.color = 'lightskyblue1',  
# ## 행 그룹 이름 표현 셀 배경색 설정  
# row\_group.background.color = 'aliceblue',  
# ## 열 제목(Heading) 배경색 설정  
# heading.background.color = 'dodgerblue4',  
# ## 표 몸체(Body) 수평선 스타일 설정  
# table\_body.hlines.style = 'dashed',   
# ## 표 몸체(Body) 수직선 색깔 설정   
# table\_body.vlines.color = 'grey',  
# ## 표 몸체(BOdy) 수직선 스타일 설정  
# table\_body.vlines.style = 'dashed'  
# )  
  
  
gt\_table9



gt\_table10 <- gt\_table9 |>  
 tab\_style(  
 locations = cells\_title(groups = "title"),  
 style = list(  
 cell\_text(weight = "bold", size = px(25))  
 )  
 ) |>  
 tab\_style(  
 locations = cells\_title(groups = "subtitle"),  
 style = list(  
 cell\_text(size = px(15))  
 )  
 )  
  
  
gt\_table10



gt\_table11 <- gt\_table10 |>  
 tab\_style(  
 locations = cells\_column\_spanners(spanners = "취업 상세"),  
 style = list(  
 cell\_text(weight = "bold", color = 'grey20', align = 'center')  
 )  
 ) %>%   
 tab\_style(  
 locations = cells\_column\_labels(columns = 5:11),  
 style = list(  
 cell\_text(weight = "lighter", color = 'grey50', align = 'center'),   
 "vertical-align:middle" ## css attribute  
 )  
 ) %>%   
 tab\_style(  
 locations = cells\_column\_spanners(spanners = "비취업 상세"),  
 style = list(  
 cell\_text(weight = "bold",color = "grey20", align = 'center')  
 )  
 ) |>  
 tab\_style(  
 locations = cells\_column\_labels(columns = 12:18),  
 style = list(  
 cell\_text(weight = "lighter", color = 'grey50', align = 'center'),   
 "vertical-align:middle" ## css attribute  
 )  
 ) |>  
 tab\_style(  
 locations = cells\_column\_labels(columns = c(3:4, 19)),  
 style = list(  
 cell\_text(weight = "bold", color = 'grey20', align = 'center'),   
 "vertical-align:middle" ## css attribute  
 )  
 )  
  
  
gt\_table11



gt\_table12 <- gt\_table11 |>   
 tab\_style(  
 locations = cells\_row\_groups(groups = everything()),  
 style = list(  
 cell\_text(color = "grey25", size = 24, align = 'center', weight = 'bold'),   
 cell\_fill(color = 'aliceblue')  
 )  
 ) |>  
 tab\_style(  
 locations = cells\_summary(groups = everything()),  
 style = list(  
 cell\_text(color = "white", size = 24, align = 'center', weight = 'bold')  
 )  
 ) |>  
 tab\_style(  
 locations = cells\_grand\_summary(),  
 style = list(  
 cell\_text(color = "white", size = 24, align = 'center', weight = 'bold')  
 )  
 ) |>  
 tab\_style(  
 locations = cells\_stub\_summary(),  
 style = list(  
 cell\_text(color = "white", size = 24, align = 'center', weight = 'bold')  
 )  
 ) |>  
 tab\_style(  
 locations = cells\_stub\_grand\_summary(),  
 style = list(  
 cell\_text(color = "white", size = 24, align = 'center', weight = 'bold')  
 )  
 )  
  
gt\_table12



gt\_table13 <- gt\_table12 |>   
 cols\_width(everything() ~ px(70))  
  
gt\_table13



gt\_table14 <- gt\_table13 |>  
 data\_color(  
 columns = c(19),  
 colors = scales::col\_numeric(  
 palette = paletteer::paletteer\_d(  
 palette = "ggsci::red\_material"  
 ) |> as.character(),  
 domain = NULL  
 )  
 )  
  
gt\_table14



if(!require('gtExtras')) {  
 install.packages('gtExtras')  
 library(gtExtras)  
}  
  
gtextras\_table1 <- gt\_table14 |>  
 gt\_duplicate\_column(취업률, dupe\_name = '취업률\_막대그래프', after = 취업률) |>  
 cols\_width(  
 취업률\_막대그래프 ~ px(100)   
 ) |>  
 gt\_plt\_percentile(column = 취업률\_막대그래프, scale = 100)  
  
gtextras\_table1



gtextras\_table2 <- gtextras\_table1 |>  
 gt\_color\_box(columns = 3, domain = c(min(df\_gt$졸업자), max(df\_gt$졸업자)), palette = "ggsci::blue\_material") |>  
 cols\_width(  
 졸업자 ~ px(80)   
 )  
  
gtextras\_table2



gtextras\_table3 <- gtextras\_table2 |>  
 gt\_duplicate\_column(취업자, dupe\_name = '취업자\_비율', after = 취업자) |>  
 fmt(columns = 취업자\_비율,  
 fns = function(x) {  
 paste0(round(prop.table(x) \* 100, 1), '%')  
 }) |>  
 cols\_merge\_n\_pct(취업자, 취업자\_비율, autohide = TRUE) |>  
 tab\_spanner(columns = 19:20, label = '취업 결과') |>  
 cols\_label(취업률\_막대그래프 = '그래프') |>  
 tab\_style(  
 locations = cells\_column\_spanners(spanners = "취업 결과"),  
 style = list(  
 cell\_text(weight = "bold",color = "grey20", align = 'center')  
 )  
 ) |>  
 tab\_style(  
 locations = cells\_column\_labels(columns = 19:20),  
 style = list(  
 cell\_text(weight = "lighter", color = 'grey50', align = 'center'),   
 "vertical-align:middle" ## css attribute  
 )  
 )  
  
gtextras\_table3



gtextras\_table4 <- gtextras\_table3 |>  
 gt\_duplicate\_column(column = 교외취업자) |>  
 gt\_duplicate\_column(column = 교내취업자) |>  
 gt\_duplicate\_column(column = 해외취업자) |>  
 gt\_duplicate\_column(column = 농림어업종사자) |>  
 gt\_duplicate\_column(column = 개인창작활동종사자) |>  
 gt\_duplicate\_column(column = 일인창사업자) |>  
 gt\_duplicate\_column(column = 프리랜서) |>  
 gt\_duplicate\_column(column = 진학자) |>  
 gt\_duplicate\_column(column = 입대자) |>  
 gt\_duplicate\_column(column = 취업불가능자) |>  
 gt\_duplicate\_column(column = 외국인유학생) |>  
 gt\_duplicate\_column(column = 제외인정자) |>  
 gt\_duplicate\_column(column = 기타) |>  
 gt\_duplicate\_column(column = 미상) |>  
 fmt(columns = ends\_with('dupe'),  
 fns = function(x) {  
 paste0(round(prop.table(x) \* 100, 1), '%')  
 })  
  
  
gtextras\_table4



temp <- temp1 <- gtextras\_table4  
  
for(i in 5:18) {  
 temp1 <- temp |>  
 cols\_merge\_n\_pct(i, i+17, autohide = TRUE)  
 temp <- temp1  
}  
  
gtextras\_table5 <- temp  
  
gtextras\_table5



1. <https://ko.wikipedia.org/wiki/%ED%91%9C> [↑](#footnote-ref-20)