

ggplot2 시각화의 기본 구조

ggplot2는 항상 아래와 같은 기본 공식을 따릅니다.

```
ggplot(data=<DATA>, mapping=aes(<MAPPINGS>))+<GEOMFUNCTION>
```

1. **ggplot(data)**: 시각화할 데이터 프레임을 지정합니다.
2. **aes()** (Aesthetics): 데이터를 그래프의 시각적 요소(x축, y축, 색상, 모양)에 매핑합니다.
3. **geom_X()** (Geometric objects): 그래프의 형태(산점도, 막대, 선 등)를 결정합니다.

1. 필수 레이어(Layer) 가이드

① 데이터와 축 설정 (The Base)

```
library(ggplot2)
# 도화지를 준비하고 축을 설정합니다. (아직 그래프는 그려지지 않음)
p <- ggplot(data = mtcars, aes(x = wt, y = mpg))
```

② 기하학적 객체 추가 (Geoms)

+ 기호를 사용하여 그래프 모양을 더합니다.

- 산점도: + `geom_point()`
- 막대 그래프: + `geom_bar()` (빈도) 또는 + `geom_col()` (값)
- 선 그래프: + `geom_line()`
- 박스 플롯: + `geom_boxplot()`

2. 시각적 요소 커스터마이징 (aes vs setting)

`aes()` 안에 쓰는 것과 밖에 쓰는 것의 차이를 이해하는 것이 중요합니다.

- **aes() 내부**: 데이터의 값에 따라 색상이나 크기가 **변하게** 하고 싶을 때 (예: `color = species`)
- **aes() 외부**: 데이터와 상관없이 **일괄적으로** 적용하고 싶을 때 (예: `color = "blue"`)

```
# 데이터 변수(cyl)에 따라 색상이 달라지는 경우
ggplot(mtcars, aes(x = wt, y = mpg, color = as.factor(cyl))) +
  geom_point()
```

```
# 모든 점을 일괄적으로 크기 3, 파란색으로 고정하는 경우
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point(size = 3, color = "blue")
```

3. 유용한 부가 기능

③ 분할 그래프 (Faceting)

하나의 그래프를 특정 변수 기준으로 여러 개의 작은 그래프로 나눕니다.

```
# 실린더(cyl) 개수별로 그래프를 따로 그림
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  facet_wrap(~cyl)
```

④ 테마와 레이블 (Themes & Labels)

그래프의 제목과 배경 스타일을 지정합니다.

```
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  labs(title = "자동차 무게와 연비", x = "무게", y = "연비") +
  theme_minimal() # 깔끔한 흰색 배경 테마
```

ggplot2 주요 함수 요약표

구분	함수	설명
Geom	geom_histogram()	연속형 변수의 분포 확인
	geom_smooth()	추세선(회귀선) 추가
Scale	scale_color_manual()	색상을 수동으로 지정
	scale_y_log10()	y축을 로그 스케일로 변환
Coord	coord_flip()	x축과 y축을 뒤집음 (가로 막대 그래프 등)
Guide	guides(color = "none")	특정 범례를 제거

ggplot2 사용 팁

- 파이프 연산자(%>% 또는 |>) 활용:** 데이터를 가공한 뒤 바로 ggplot으로 넘기면 코드가 훨씬 깔끔해집니다.
 - df %>% filter(val > 10) %>% ggplot(aes(x, y)) + geom_point()
 - 데이터 타입 확인:** x축에 범주형 데이터를 넣을 때는 반드시 as.factor() 처리가 되어 있는지 확인하세요.
 - 레이어 순서:** 아래에 깔리는 레이어를 먼저 작성합니다. (예: geom_boxplot()을 먼저 쓰고 그 위에 geom_jitter()로 점을 찍음)
- 이 기본 문법만 익히면 어떤 복잡한 그래프도 층층이 쌓아서 만들어낼 수 있습니다.

혹시 특정 데이터를 활용해 실제 ggplot2 그래프를 하나 완성해보는 실습을 해보고 싶으신가요? 혹은 데이터의 분포를 보여주는 고급 그래프(Violin plot 등) 그리는 법이 궁금하신가요?

여러 개의 그래프를 한 페이지에 레이아웃(Patchwork)

1. Patchwork: 여러 그래프를 자유자재로 배치하기

patchwork는 ggplot2 객체들을 마치 산술 연산을 하듯 +, /, | 기호로 더하고 나누어 배치할 수 있게 해주는 혁신적인 패키지입니다.

💡 주요 연산자

- + 또는 | : 그래프를 **옆으로(가로)** 배치
- / : 그래프를 **위아래로(세로)** 배치
- () : 그래프들을 그룹화하여 배치 구조를 계층화

예제 코드

```
# 패키지 로드
library(ggplot2)
library(patchwork)

# 1. 개별 그래프 생성
p1 <- ggplot(mtcars, aes(wt, mpg)) + geom_point() +
  ggtitle("산점도")
p2 <- ggplot(mtcars, aes(as.factor(gear))) + geom_bar() +
  ggtitle("막대 그래프")
p3 <- ggplot(mtcars, aes(as.factor(cyl), mpg)) + geom_boxplot() +
  ggtitle("박스 플롯")

# 2. 레이아웃 배치
# p1과 p2를 나란히 배치하고, 그 아래에 p3를 배치
layout <- (p1 | p2) / p3

# 3. 전체 레이아웃 서식 지정
layout + plot_annotation(title = "자동차 데이터 종합 분석 리포트")
```

2. Plotly: 정적 그래프를 인터랙티브하게 변환

plotly 패키지의 ggplotly() 함수를 사용하면, 기존에 만든 ggplot2 그래프에 마우스 오버(Hover), 확대/축소(Zoom), 클릭 시 데이터 표시 기능을 즉시 추가할 수 있습니다.

💡 주요 특징

- **Hover**: 마우스를 올리면 해당 포인트의 상세 값이 나타남.
- **Zoom & Pan**: 특정 영역을 드래그하여 확대 가능.

- **HTML 저장:** 인터랙티브 기능이 살아있는 채로 웹페이지나 보고서에 삽입 가능.

예제 코드

```
library(plotly)

# 1. ggplot2로 기본 그래프 생성
p <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point(aes(text = paste("꽃잎 길이:", Petal.Length))) + #
  # 툴팁에 커스텀 텍스트 추가
  theme_minimal()

# 2. 인터랙티브 그래프로 변환
ggplotly(p)
```

3. 두 패키지 비교 및 활용 시나리오

기능	Patchwork	Plotly
주 목적	여러 그래프의 레이아웃 구성	그래프에 동적인 기능 부여
출력 형태	정적 이미지 (PNG, PDF 등)	동적 웹 객체 (HTML, Widget)
주요 용도	학술 논문, 인쇄용 보고서	대시보드, 웹 프리젠테이션
학습 난이도	매우 쉬움 (+, / 연산)	쉬움 (기본 ggplot 객체 변환)

실전 팁: 레이아웃 고도화하기

patchwork를 사용할 때 특정 그래프의 크기를 조절하고 싶다면 `plot_layout()` 함수를 추가해 보세요.

```
p1 + p2 + plot_layout(widths = c(2, 1)) # p1을 p2보다 2배 넓게 배치
```