

# R tidyverse 마스터 클래스

---

## 7강 - ggplot2의 qplot

슬기로운통계생활

Issac Lee



# 학습목표

완벽하게 하나의 차트를 코드로 완성 시켜보자.

처음부터 끝까지 **완벽하게** 이해하는 것은 중요하다.

응용의 시작점.

# 파이차트



악명높은 파이차트

생각보다 그리기 어려움.

- 여러가지 기법들이 섞여있음
- 반복 숙달하면 점점 `ggplot`으로 그리는 속도가 빨라짐
- 데이터 로딩

```
library(tidyverse)
library(palmerpenguins)

pie_data <- penguins %>%
  group_by(species) %>%
  tally() %>%
  mutate(prop = n / sum(n))
```

```
## # A tibble: 3 x 3
##   species      n    prop
##   <fct>     <int>  <dbl>
## 1 Adelie     152  0.442
## 2 Chinstrap   68  0.198
## 3 Gentoo     124  0.360
```



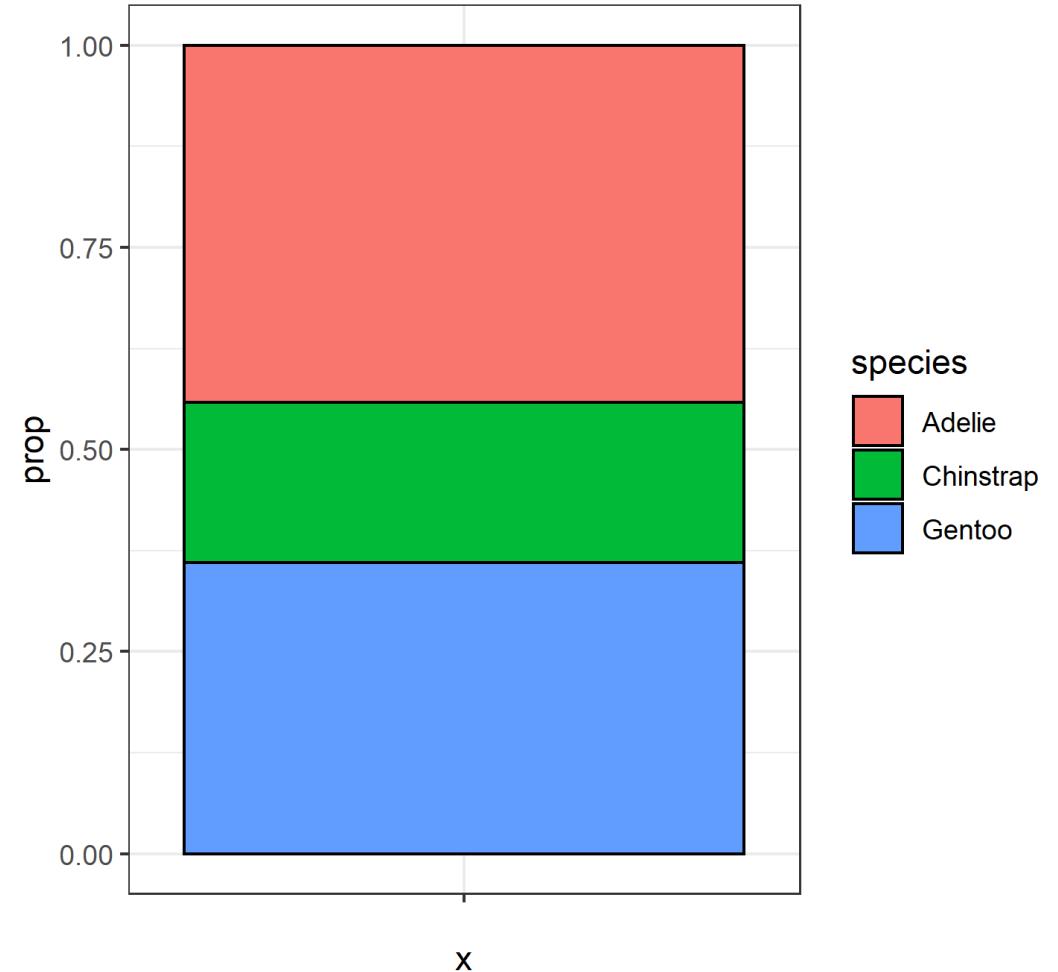
# Stacked Barchart

```
p <- pie_data %>%
  ggplot(aes(x = "", y = prop,
             fill = species)) +
  geom_bar(stat = "identity",
           width = 1,
           color = "black")
```

p

- `stat = "identity"` 개념 이해하기

- 기본옵션: `stat = "identity"`
- y값으로 실제 값이 들어왔기 때문



# stat 옵션 이해하기



## Bar chart의 기본 포맷

- X축에 factor를 맵핑
- 맵핑된 factor 변수를 geom\_bar 레이어에서 자동으로 count 해줌.
  - count
- Y축에 직접 Count 혹은 값 정보를 맵핑시키는 경우도 존재
  - identity

- stat = "count"

```
ggplot(penguins, aes(x = species)) +  
  geom_bar(stat = "count")
```

- stat = "identity"

```
ggplot(pie_data, aes(x = "", y = n)) +  
  geom_bar(stat = "identity")
```

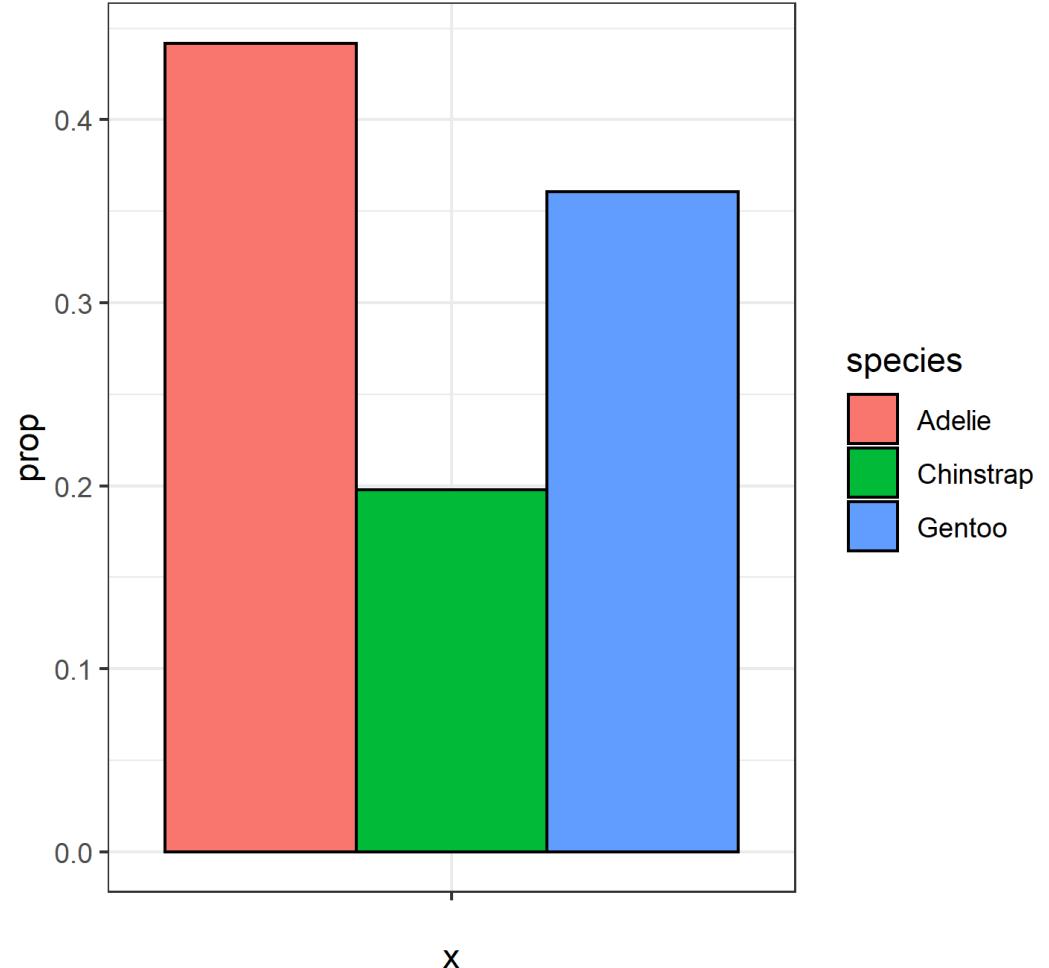
# position 옵션 이해하기



```
pie_data %>%  
  ggplot(aes(x = "", y = prop,  
             fill = species)) +  
  geom_bar(stat = "identity",  
           position = "dodge",  
           width = 1,  
           color = "black")
```

- **position = "dodge"** 개념 이해

- 기본옵션: **stat = "stack"**
- 쌓여져있는 것을 옆으로 펼침



# position 옵션 이해



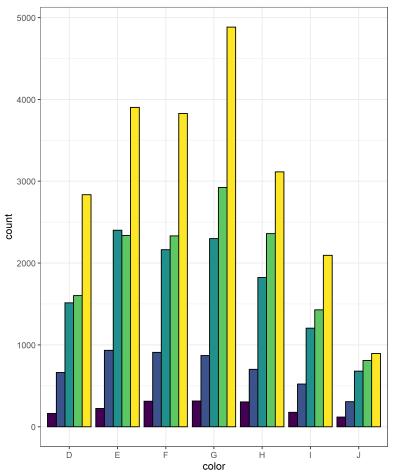
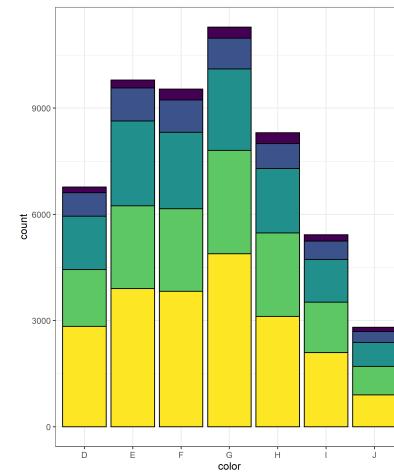
- position = "stack"

```
ggplot(diamonds, aes(color, fill = cut)) +  
  geom_bar(position = "stack",  
           color = "black")
```

- position = "dodge"

```
ggplot(diamonds, aes(color, fill = cut)) +  
  geom_bar(position = "dodge",  
           color = "black")
```

## Postion 옵션: Stack vs. Dodge

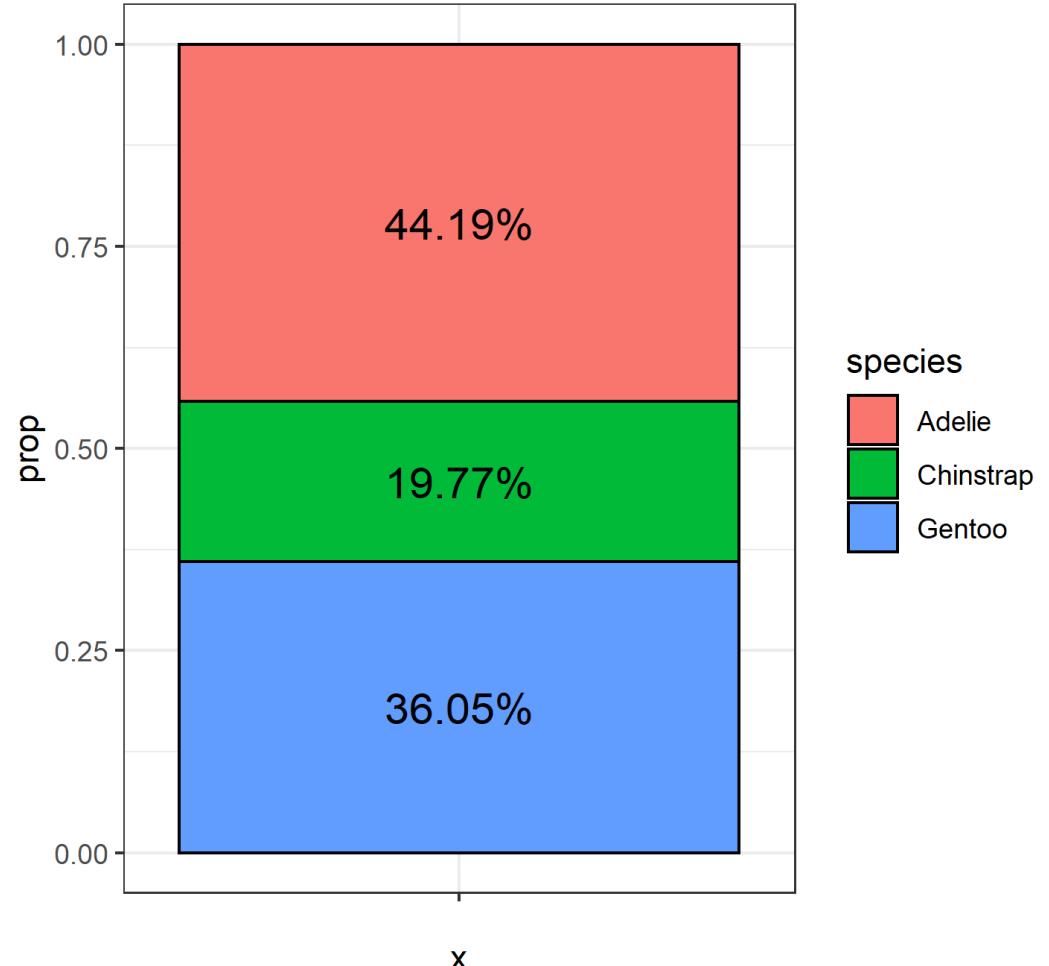




# geom\_text를 사용한 퍼센트 정보 넣기

```
p <- p + geom_text(  
  aes(label = paste0(  
    round(prop * 100,  
    digits = 2), "%"))  
,  
  position = position_stack(  
    vjust = 0.5  
)  
,  
  size = 5) # 글씨 크기  
  
p
```

- `vjust` 값에 따른 정렬위치
  - 0 - bottom
  - 0.5 - middle
  - 1 - top

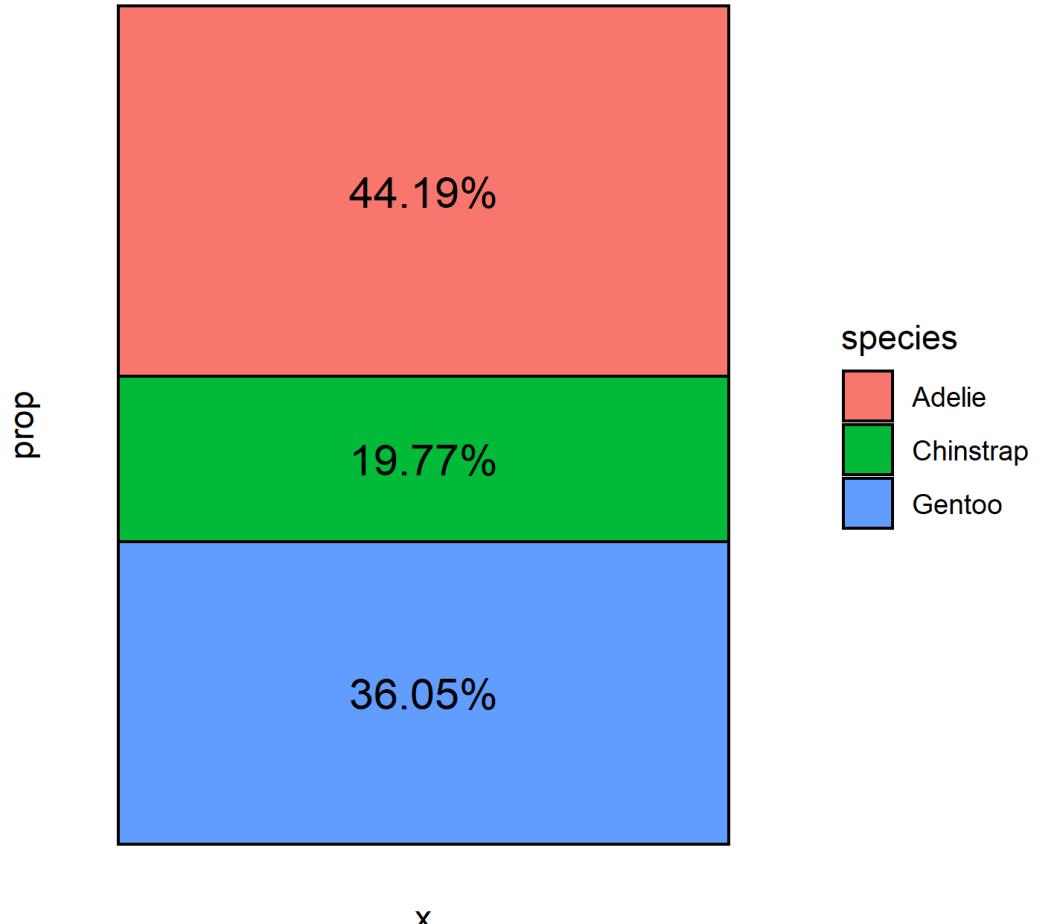




# 테마 레이어를 이용한 세부 조정

```
theme_set(theme_bw())
p <- p + theme(
  panel.border = element_blank(),
  panel.grid   = element_blank(),
  axis.ticks   = element_blank(),
  axis.text.y  = element_blank(),
  axis.text.x  = element_blank(),
  plot.title   = element_text(size=24,
                               face="bold"))
p
```

- 하나씩 캐다가 끄면서 이해해보자.



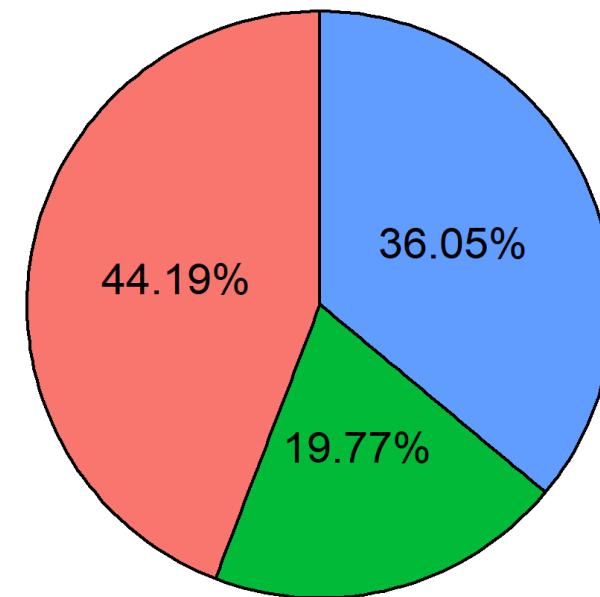


# coord\_polar를 사용한 파이 만들기

```
p + guides(fill = guide_legend(  
    title = "펭귄 종류")) +  
  labs(x = "", y = "",  
    title = "팔머 펭귄 종별 분포도") +  
  coord_polar("y", start = 0)
```

- coord\_polar() 축을 중심으로 회전
  - x축과 y축을 선택해서 돌릴 수 있음
  - start 옵션을 사용해서 돌릴 수 있음

팔머 펭귄 종별 분포도



펭귄 종류
Adelie
Chinstrap
Gentoo

# 학습목표

`qplot()` 사용하기

ggplot을 빠르게 그리고 싶다면?

난 베이스 plot이 더 좋던데?



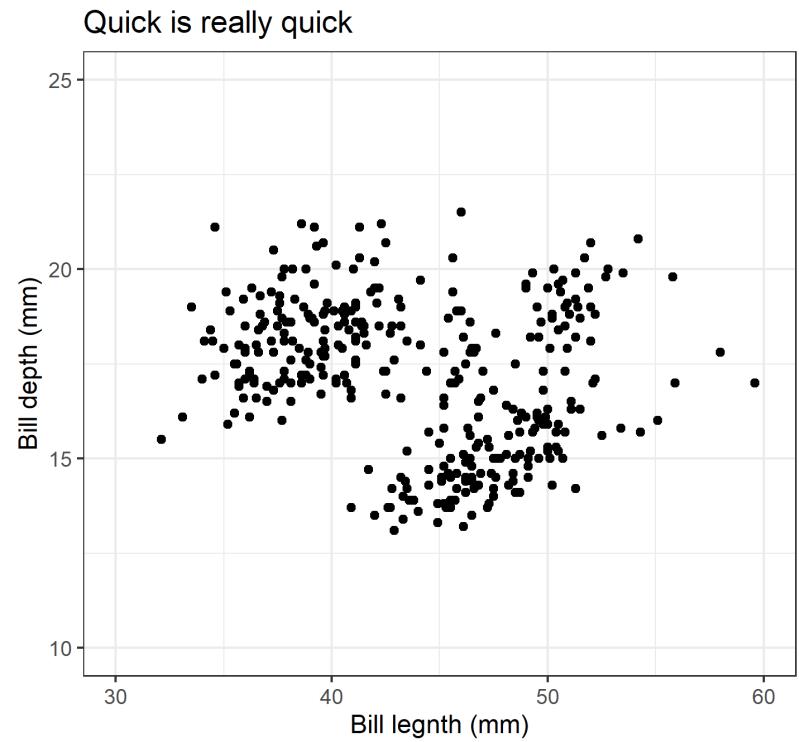
# qplot()

- quickplot()

base 함수를 옮겨왔다?

기본 베이스 함수의  
문법을 사용함

```
qplot(  
    bill_length_mm,  
    bill_depth_mm,  
    data = penguins,  
    main = "Quick is really quick",  
    xlim = c(30, 60),  
    ylim = c(10, 25),  
    xlab = "Bill length (mm)",  
    ylab = "Bill depth (mm)"  
)
```

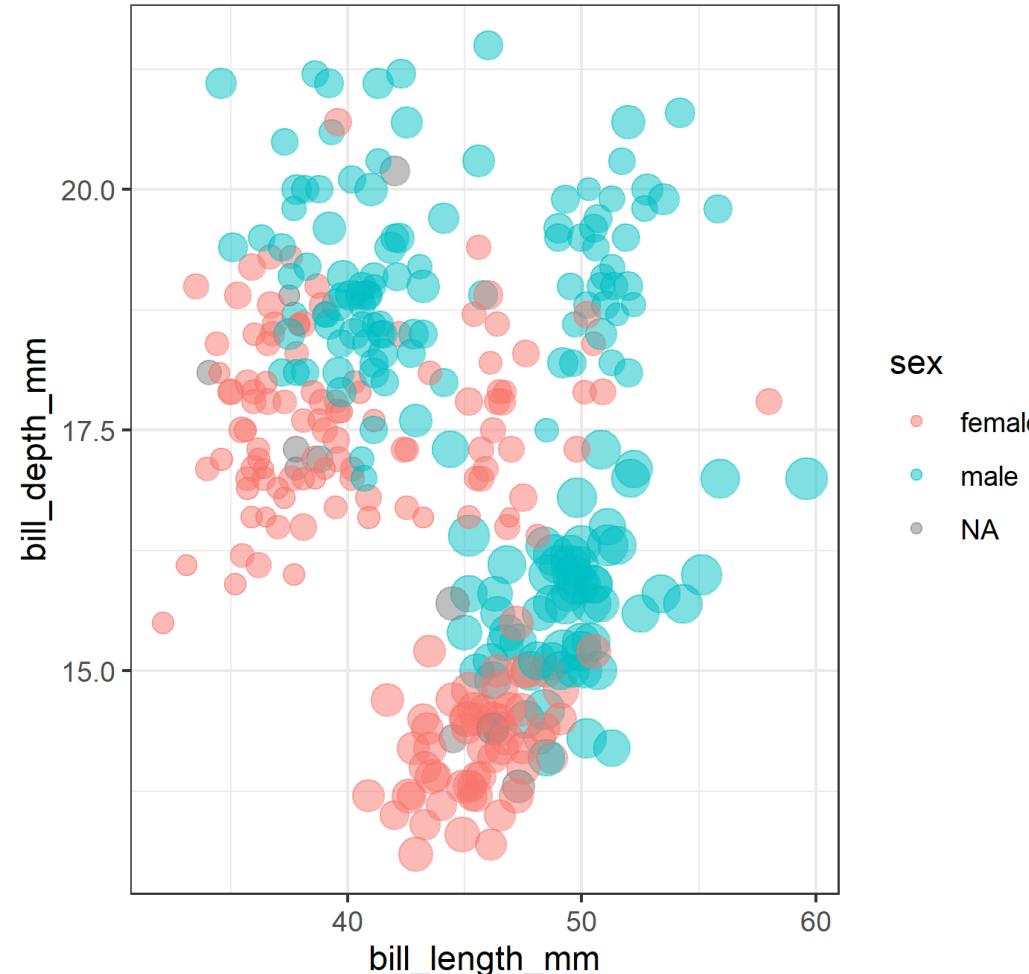


# Aesthetic 맵핑



```
qplot(  
  bill_length_mm,  
  bill_depth_mm,  
  data = penguins,  
  color = sex,  
  size = I(penguins$body_mass_g/1000),  
  alpha = I(0.5)  
)
```

- `I()`를 사용한 aes. identity 효과 부여
- `I()`함수 안의 속성 멘션 시 원래 데이터 표시

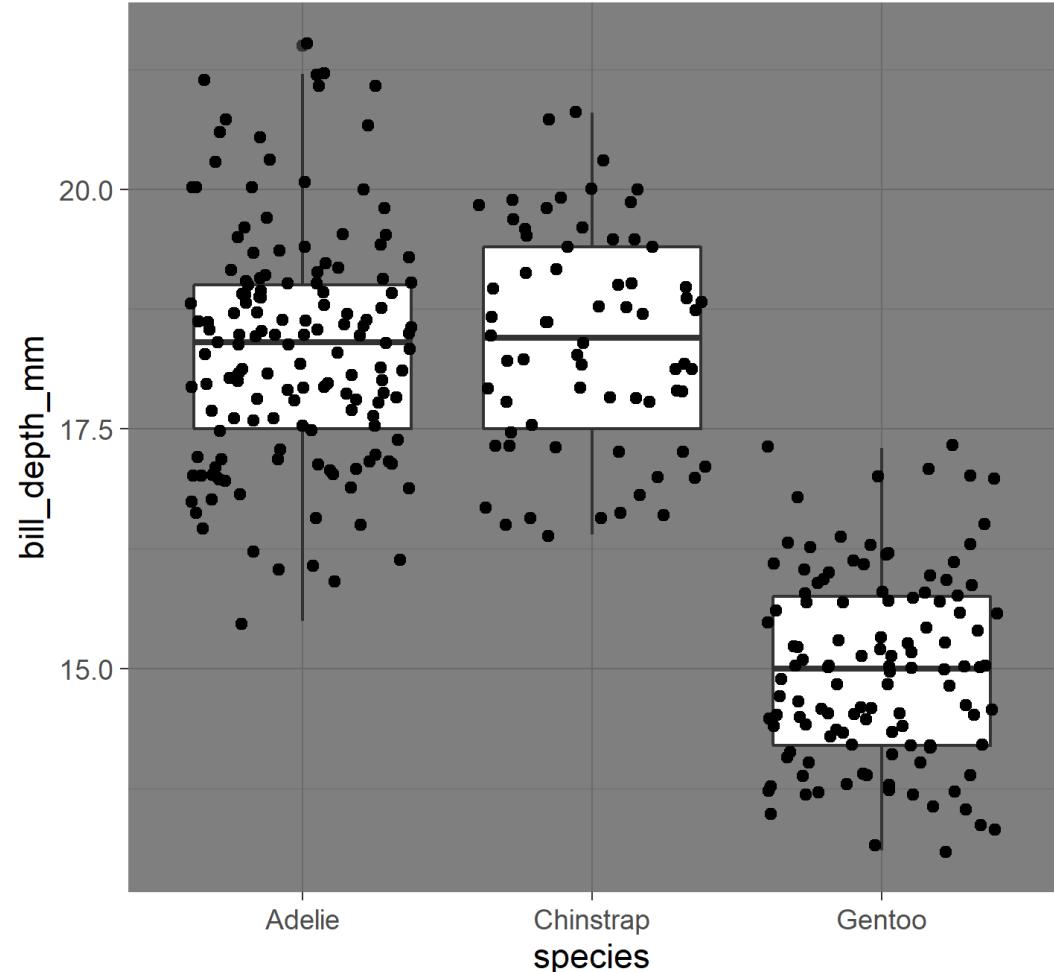


# geom 레이어 선택하기



```
q1 <- qplot(  
  species,  
  bill_depth_mm,  
  data = penguins %>% drop_na(),  
  geom = c("boxplot", "jitter"))  
)  
q1 + theme_dark()
```

- `qplot()`의 결과물은 당연하게도 ggplot 오브젝트
- 이후 다른 레이어 입히기 가능

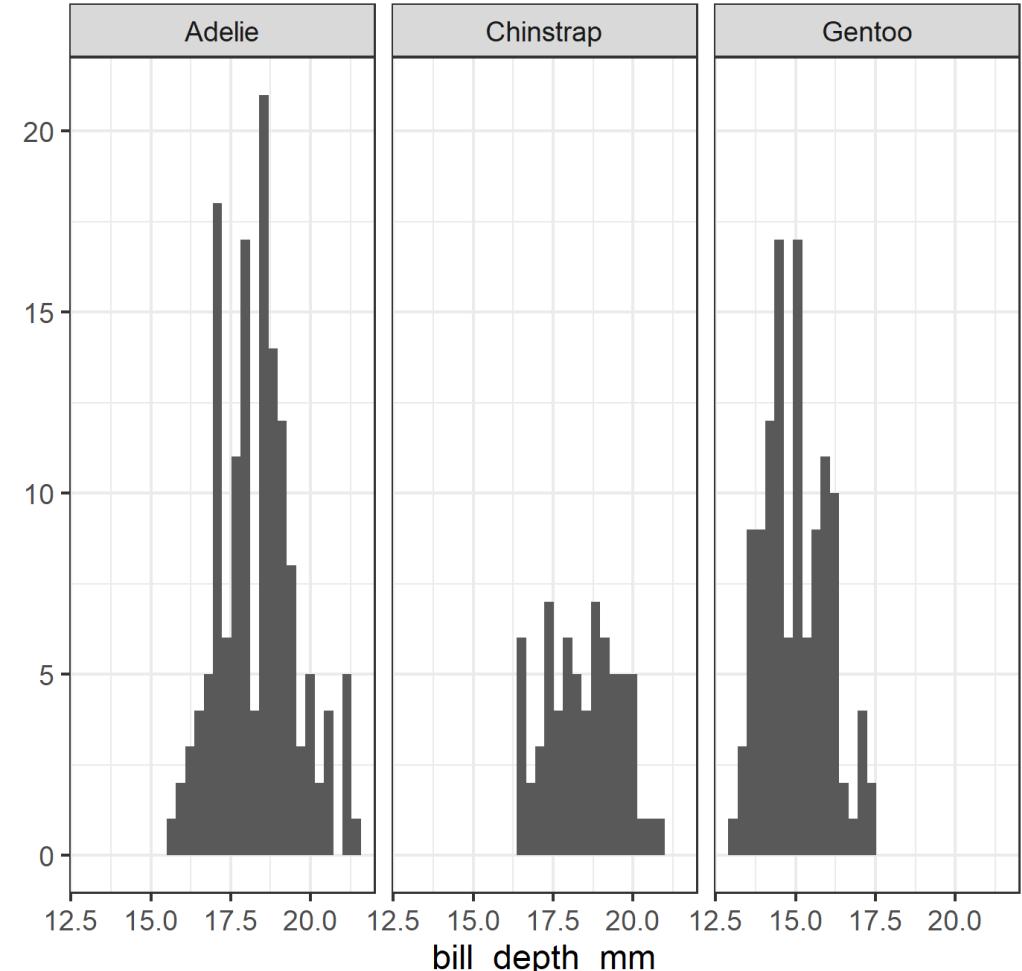


# facet 적용하기



```
q2 <- qplot(  
  bill_depth_mm,  
  data = penguins %>% drop_na(),  
  geom = "histogram",  
  facets = .~species  
)  
q2
```

- qplot()의 결과물은 당연하게도 ggplot 오브젝트
- 이후 다른 레이어 입히기 가능



# 학습목표

여러 패키지를 이용한 그래프 그리기

<https://r-charts.com/ggplot2/>

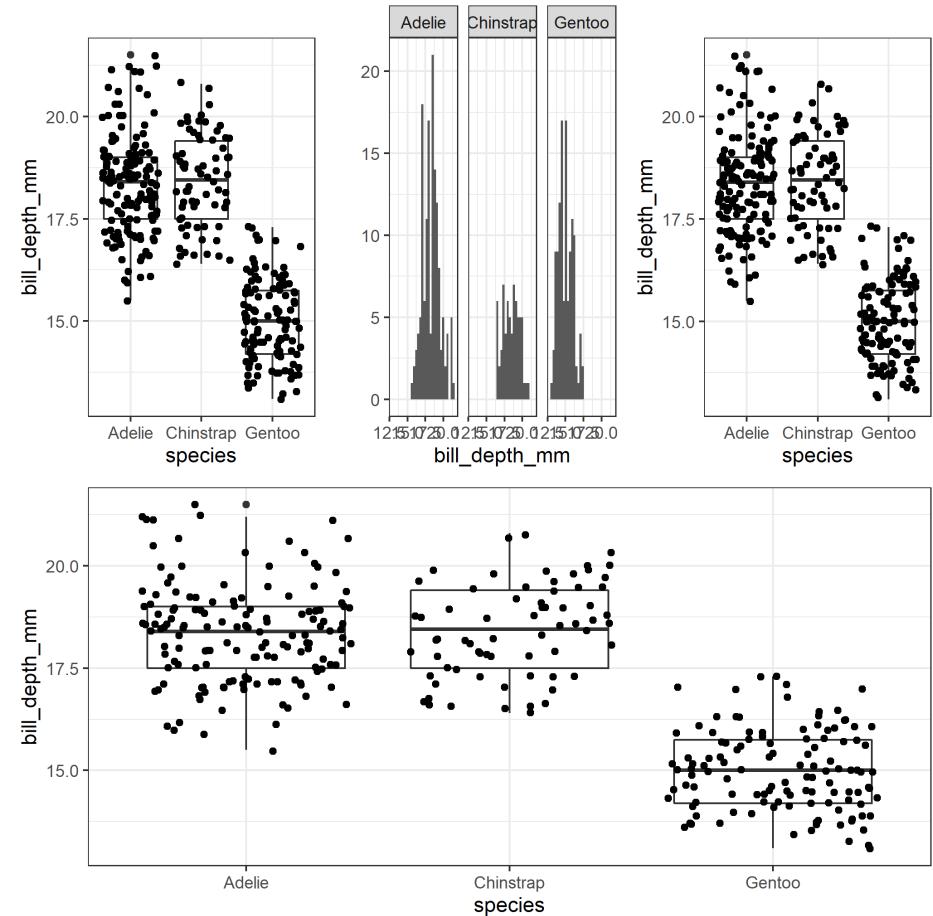
<https://exts.ggplot2.tidyverse.org/gallery/>

# patchwork 패키지를 통한 여러개 그래프



```
library(patchwork)  
(q1 + q2 + q3) / q1
```

- <https://patchwork.data-imaginist.com/>
- 그래프 여러개 이어 붙이는 방법을 사칙연산 기호로 만들어 놓음



# patchwork 패키지를 통한 여러개 그래프

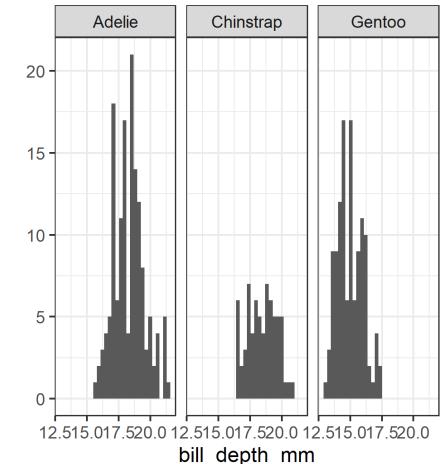
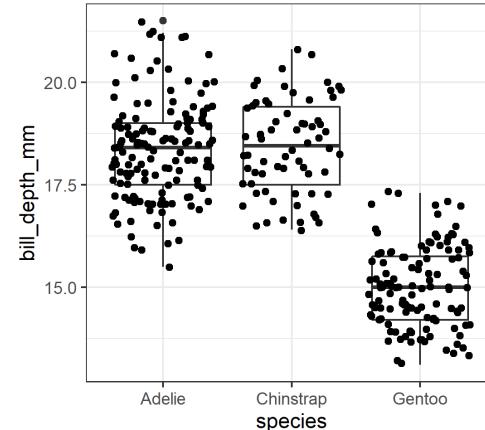


```
library(patchwork)  
(q1 + plot_spacer())/(plot_spacer() + q2)
```

## 섬세한 레이아웃 조정도 가능

- `plot_spacer()` 그래프 공백 넣기
- `plot_layout()`
  - `ncol = 3`: 전체 그래프 열 3개로 설정
  - `widths = c(2, 1)`: 가로비율 2:1
  - `design = custom_layout`

```
layout <- "##BB  
AACC"
```



# Custom design 적용하기

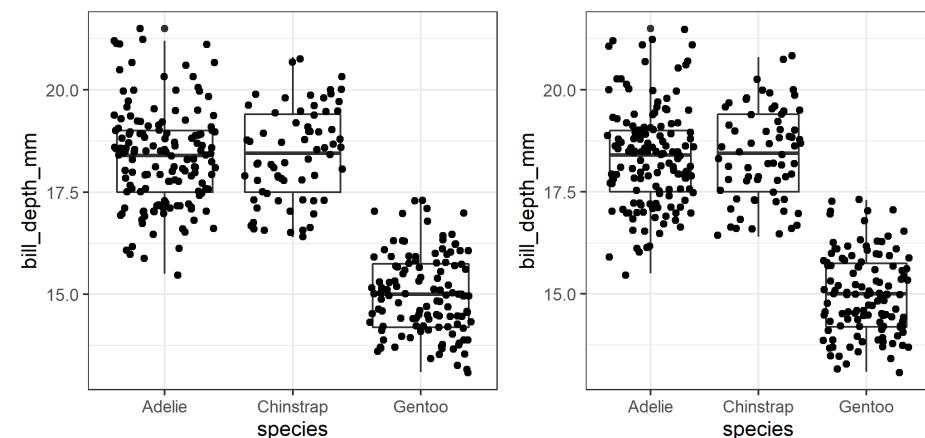
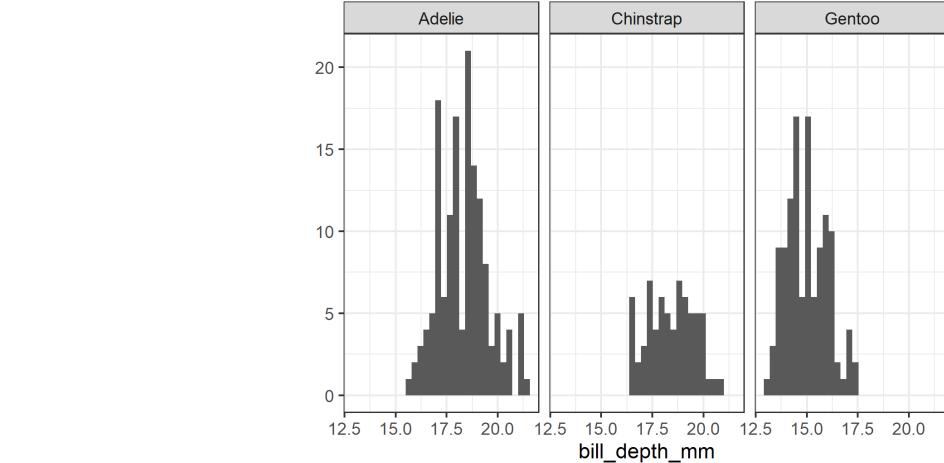


```
custom_layout <- "  
# BBB  
AACC  
"  
q1 + q2 + q1 +  
    plot_layout(design = custom_layout)
```

`area()`를 통한 그래프 컨테이너 생성 가능

- 그래프 겹쳐그리기

```
custom_layout <- c(  
    area(t = 2, l = 1, b = 5, r = 4),  
    area(t = 1, l = 3, b = 3, r = 5)  
)
```





## 같이 보면 좋은 책 추천

### [1] [R for Data Science](#)

- 웹 상에 무료 공개된 책입니다.
- 위 교재의 한글 번역본 [R](#) 을 활용한 [데이터과학](#)도 있습니다.
- 도서 제목 클릭하셔서 구매하시면 저의 [사리사욕](#)을 충당하는데 도움이 됩니다.

## 참고자료

- [stat identity 이유](#)
- [맥에서 폰트사용 자료참고](#)
- [외국 폰트 사이트 - dafont.com](#)