R4DS(2e) 1. 데이터 시각화 - v0.3

한상곤(sangkon@pusan.ac.kr / sigmadream@gmail.com)

2023.09.13, updated: 2023.12.01

R을 사용해서 데이터분석의 전반적인 과정을 진행하기 위해서 필요한 것은 tidyverse라는 패키지 입니다. 해당 패키지를 설치하신 후 관련 라이브러리를 불러와주세요. 이후 책에 소개된 패키지의 설치 및 활용 방법은 스터디 시간에 안내하였으며 관련하여 추가적인 정보가 필요하시면 언제든 디스코드에 질문해주시면 됩니다.

데이터 시각화

R4DS(2e)에서 사용하게 될 팔머펭귄 데이터를 제공하는 palmerpenguins 패키지와 색맹에 안전한 시각적 표현법을 제공하는 ggthemes 패키지도 설치 후 불러와주세요.

```
library(palmerpenguins)
library(ggthemes)
```

책에서 팔머펭귄 데이터를 간단하게 확인해보도록 하겠습니다. 출력 결과에서 확인할 수 있듯이, tidyverse를 사용하게 될 경우 R에서 주로 사용하는 data.frame이 아니라 tibble이 적용되어 있는 것을 확인할 수 있습니다. tibble에 대한 세부적인 사항은 스터디를 진행하면서 알아보도록 하겠습니다. 결론적으로, data.frame이 아닌 다른 형태의 데이터 구조를 사용한다는 것을 알아두세요.

```
penguins
```

```
## # A tibble: 344 x 8
##    species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##    <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
##  1 Adelie  Torgersen           39.1          18.7               181        3750
...
```

데이터에서 제공하는 변수(컬럼)에 대한 정보를 확인하도록 하겠습니다. 해당 정보가 당장에 중요한 정보는 아닐 수 있지만, 일반적으로 시각화를 위해서 데이터의 형태나 타입을 사전에 확인하는 작업은 꼭 필요한 절차입니다.

```
glimpse(penguins)
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
...
```

교재에서 시각화를 원하는 정보는 flipper lengths와 body masses의 관계입니다. 두 가지 변수에 대한 관계를 ggplot을 사용해서 시각적으로 표현하는 절차를 빠르게 학습해보도록 하겠습니다.

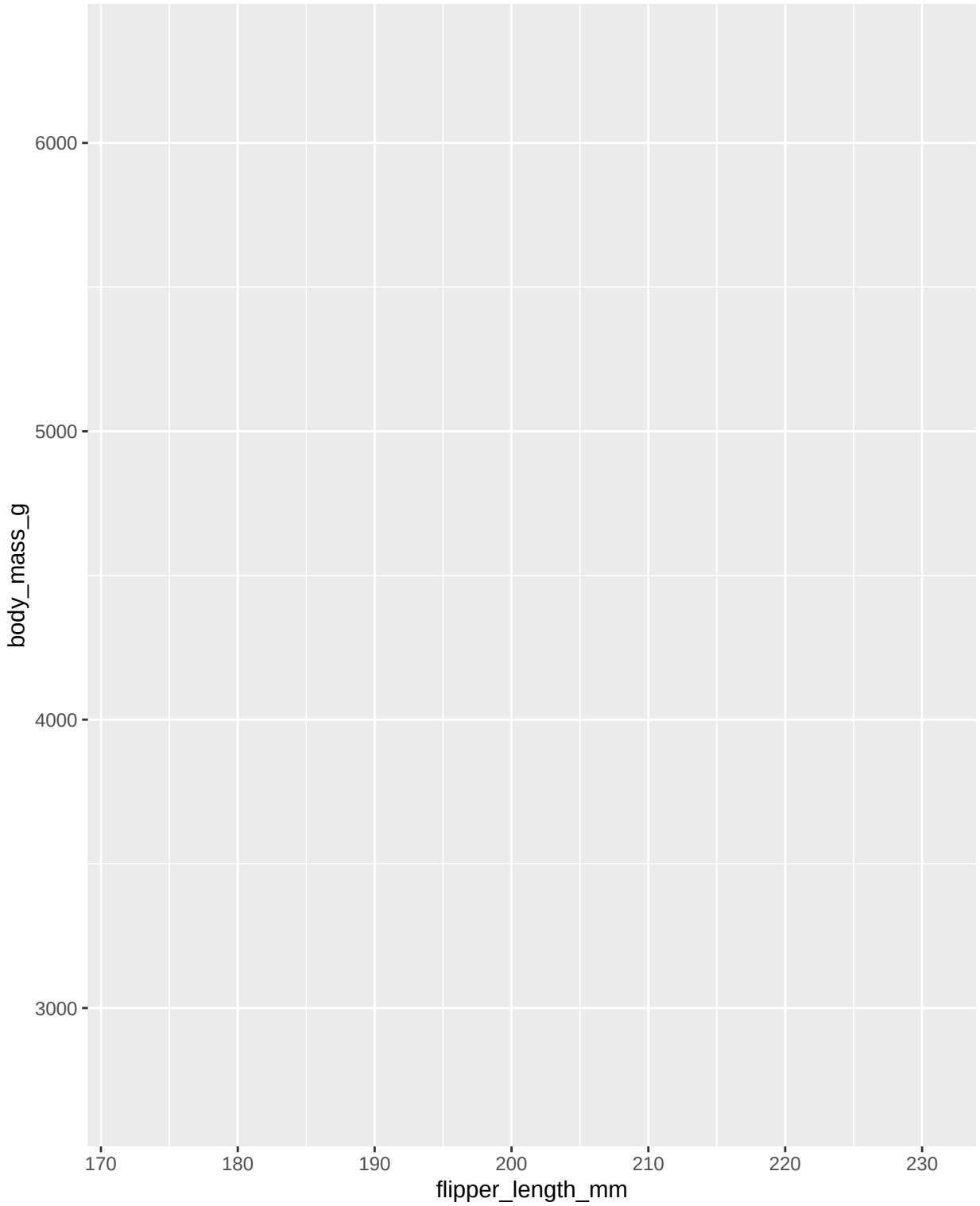
P.S : 기존에 파이썬에서 진행하던 시각화 기법과 차이점이 많기 때문에 ggplot에 대한 세부적인 지식보다 ggplot이 시각 정보를 구성하는 과정과 절차를 주의깊게 살펴보시길 권해드립니다.

ggplot2를 사용해서 시각정보를 표현할 수 있는 객체를 생성하도록 하겠습니다. 해당 객체를 생성할 때 가장 중요한 정보는 어떤 데이터를 시각적으로 표현할 것이니 전달해야 합니다. 아래 코드를 통해서 시각 정보를 전달하기 위한 첫번째 단계를 확인할 수 있습니다.

```
ggplot(data = penguins)
```


데이터의 세부 정보를 시각적으로 표현하기 위해서 ggplot에 어떤 변수를 활용해야 할지 전달해야 합니다. ggplot 함수의 `mapping` 매개변수를 사용해서 시각적 속성(aesthetics)을 전달합니다. 시각적 속성은 `aes` 함수를 사용해서 정의하며, aes 함수의 매개변수 인 x와 y는 x축과 y축에 해당하는 값과 연계됩니다. 여기서는 x는 `flipper_length_mm` 이고 y는 `body_mass_g`입니다.

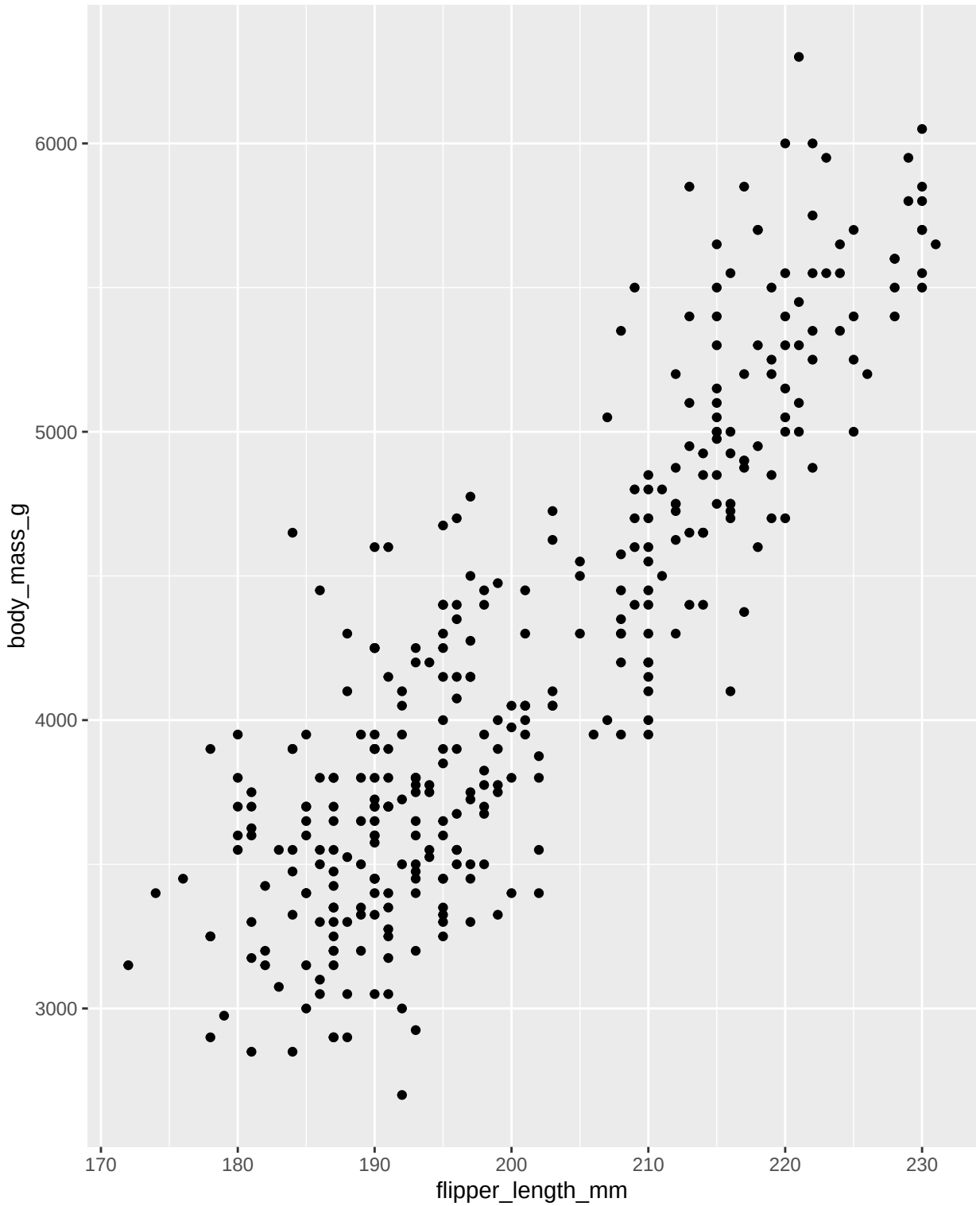
```
ggplot(data = penguins,
       mapping = aes(x = flipper_length_mm, y = body_mass_g))
```



기존에 존재하지 않던 격자가 생성되었다는 것을 통해서 x, y에 전달된 데이터가 적용되었음을 알 수 있씁니다. 하지만 해당 데이터를 어떤 형태로 표현해야 할 지 결정(혹은 정의하지) 않았기 때문에 관련 시각 정보가 표현되지 않습니다.

데이터를 표현하는 데 사용하는 기하학적 개체인 geom을 정의해야 합니다. 이러한 기하학적 개체는 geom_로 시작하는 함수를 통해 ggplot에 적용할 수 있습니다. 예를 들어 막대형 차트에는 geom_bar 함수이며, 꺾은선 차트에는 geom_line, 박스 플롯은 geom_boxplot, 산점도를 활용하는 차트는 geom_point() 등을 사용합니다. 저희는 산점도를 활용해보도록 하겠습니다.

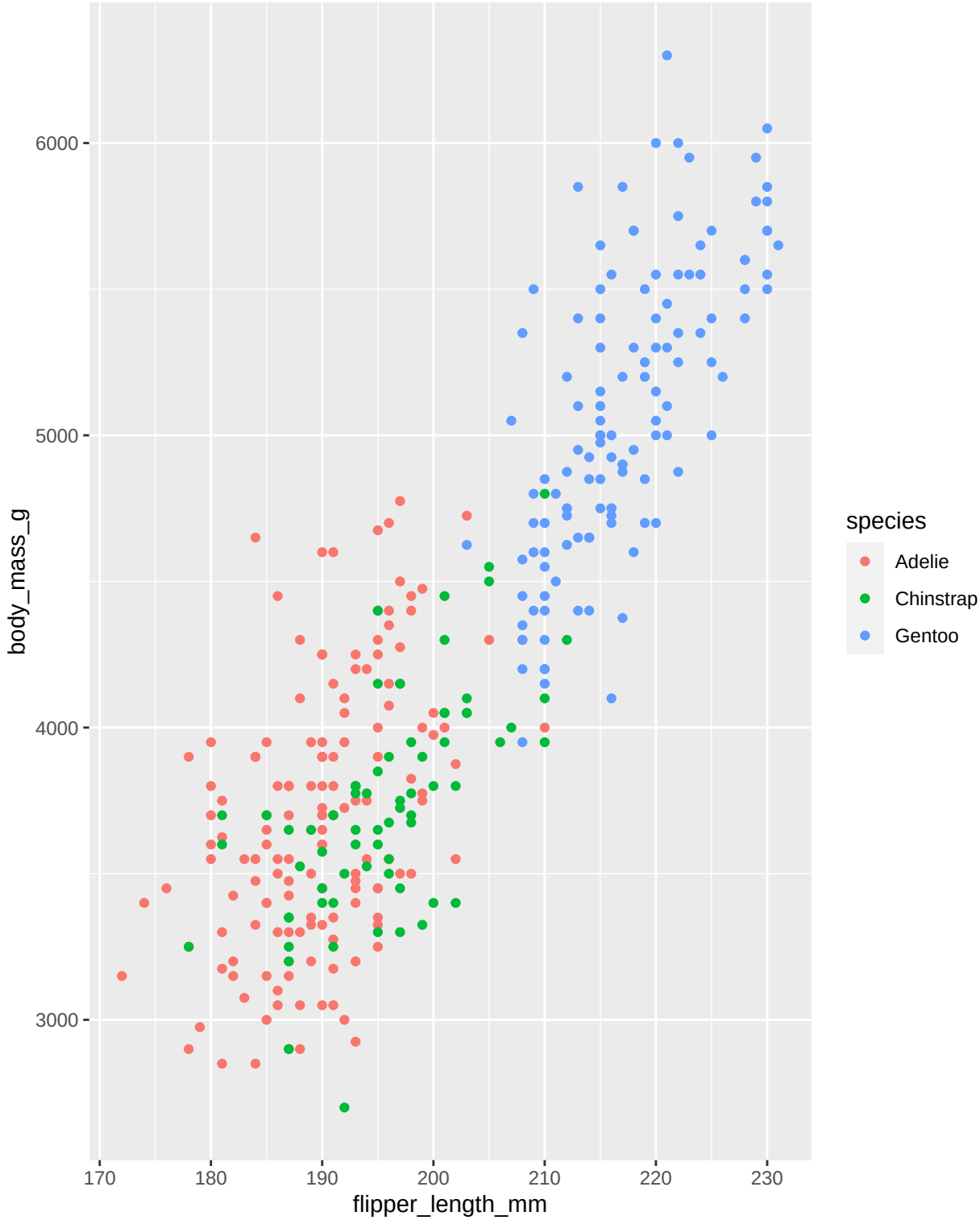
```
ggplot(data = penguins,
       mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point()
```



우리가 만든 산점도는 x,y에 대한 관계를 쉽게 이해하기 쉽지 않습니다. 변수에 대한 내용을 명확하게 이해가 쉽지 않기 때문에 색을 활용해서 추가 정보를 표현하도록 하겠습니다. 범주형 변수인 `species`를 적용해서 ggplot에서 적당한 색을 사용할 수 있도록 관련 정보를 전달하면, 범례도 추가됩니다.

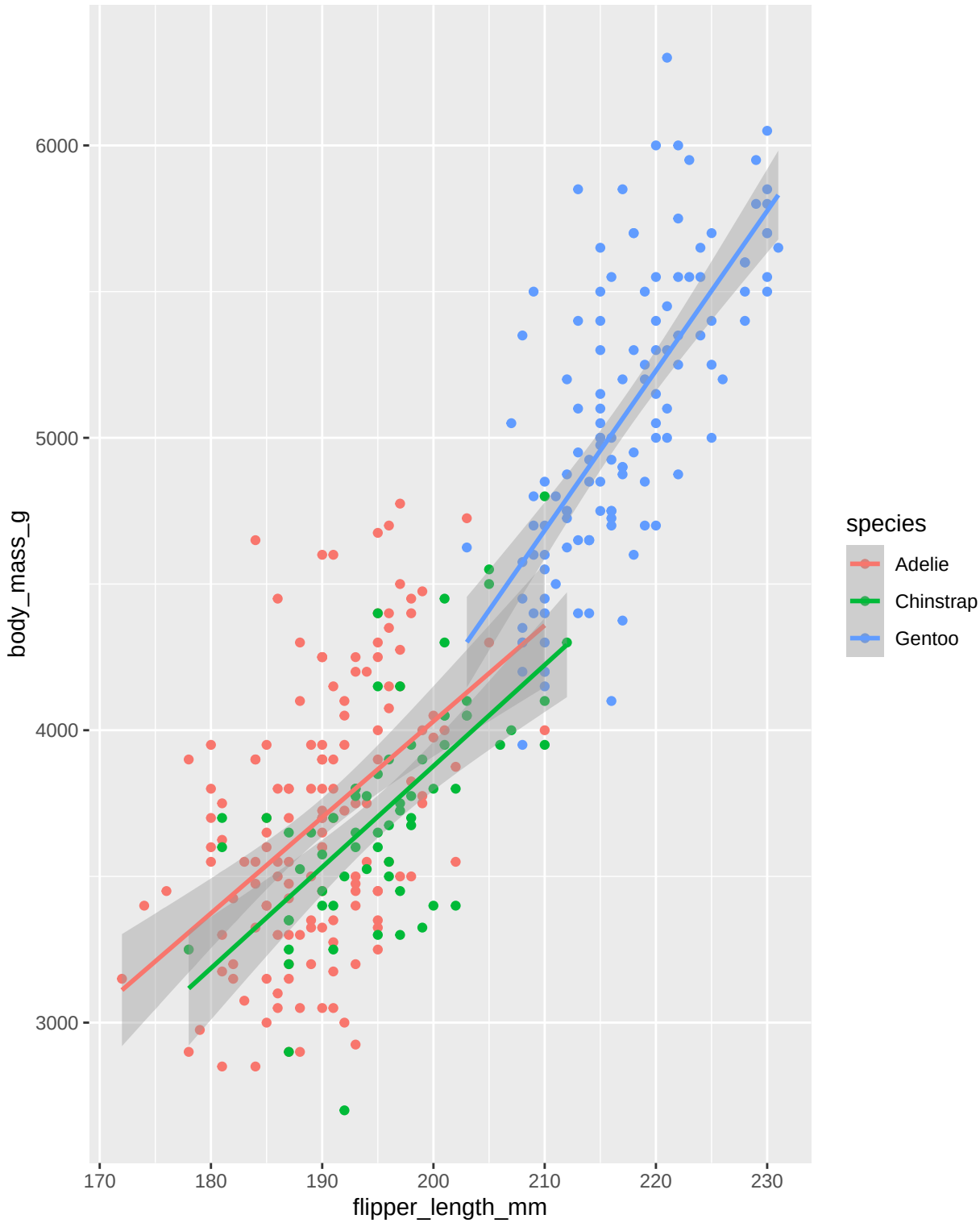
```
ggplot(
  data = penguins,
```

```
  mapping = aes(x = flipper_length_mm, y = body_mass_g, color = species)
) +
  geom_point()
```



geom_smooth 함수를 사용하면 선형 모델을 기반으로 한 추세선도 추가할 수 있습니다.

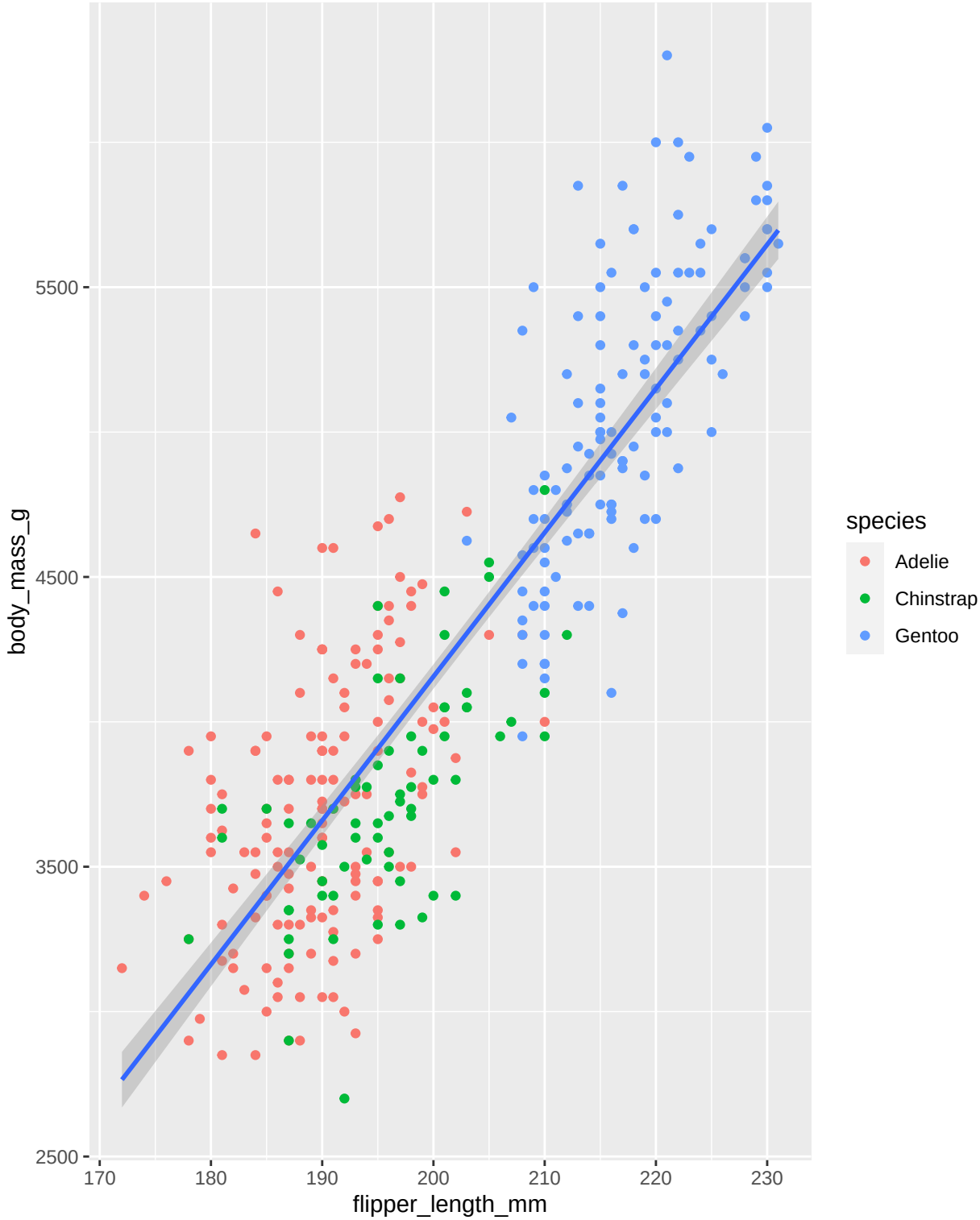
```r
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g, color = species)
) +
  geom_point() +
  geom_smooth(method = "lm")
```



각각의 펭귄 종에 대해 별도의 추세선을 추가하는 것 보다는, 전체 데이터에 추세선을 추가하고자 합니다. 이 경우 새로운 시각 정보를 추가하기 위해서 아래와 같이 코드를 수정합니다.

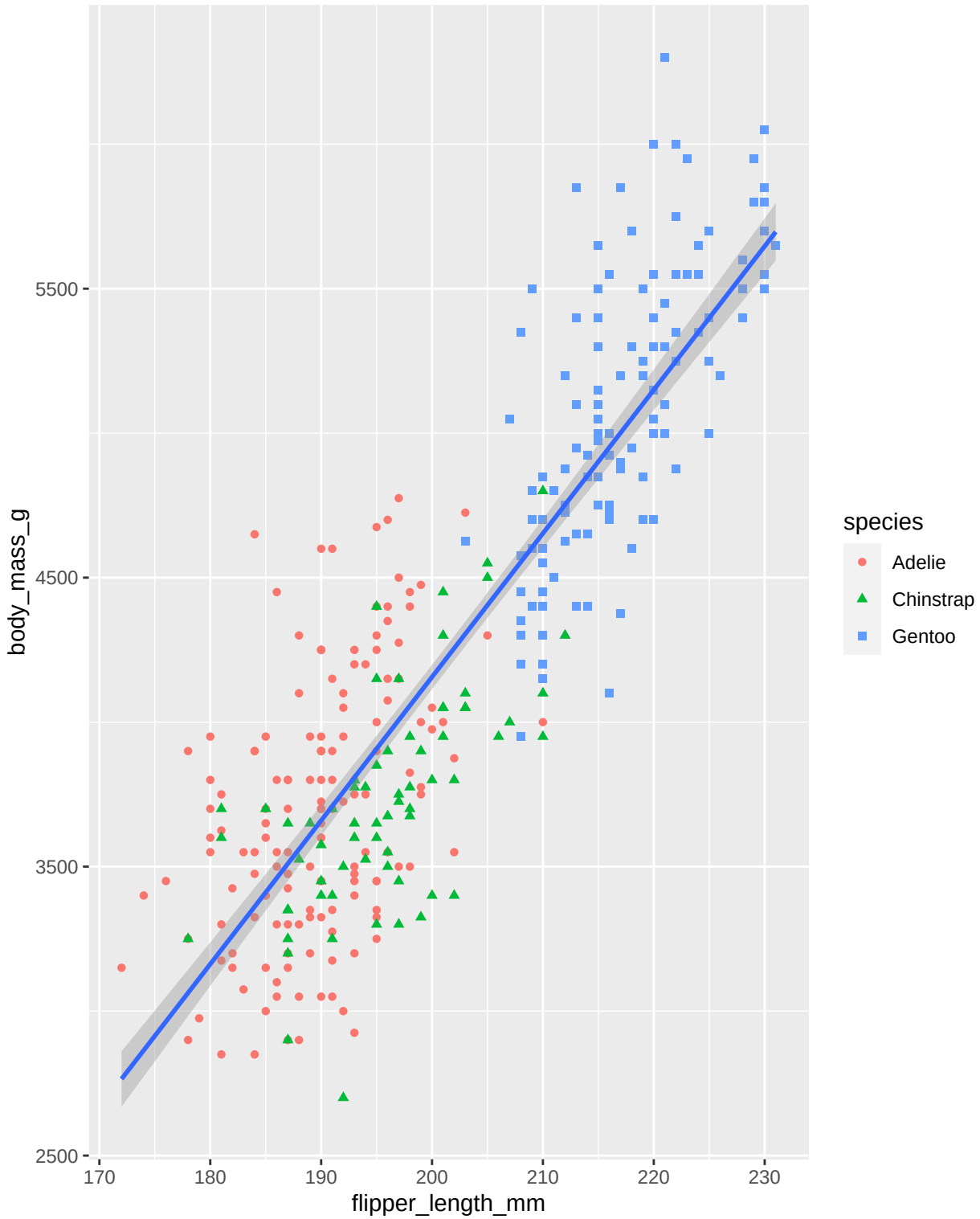
```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
```

```
) +
  geom_point(mapping = aes(color = species)) +
  geom_smooth(method = "lm")
```



펭귄 종에 대한 정보를 추가하기 위해서 산점도의 모양을 수정하도록 하겠습니다.

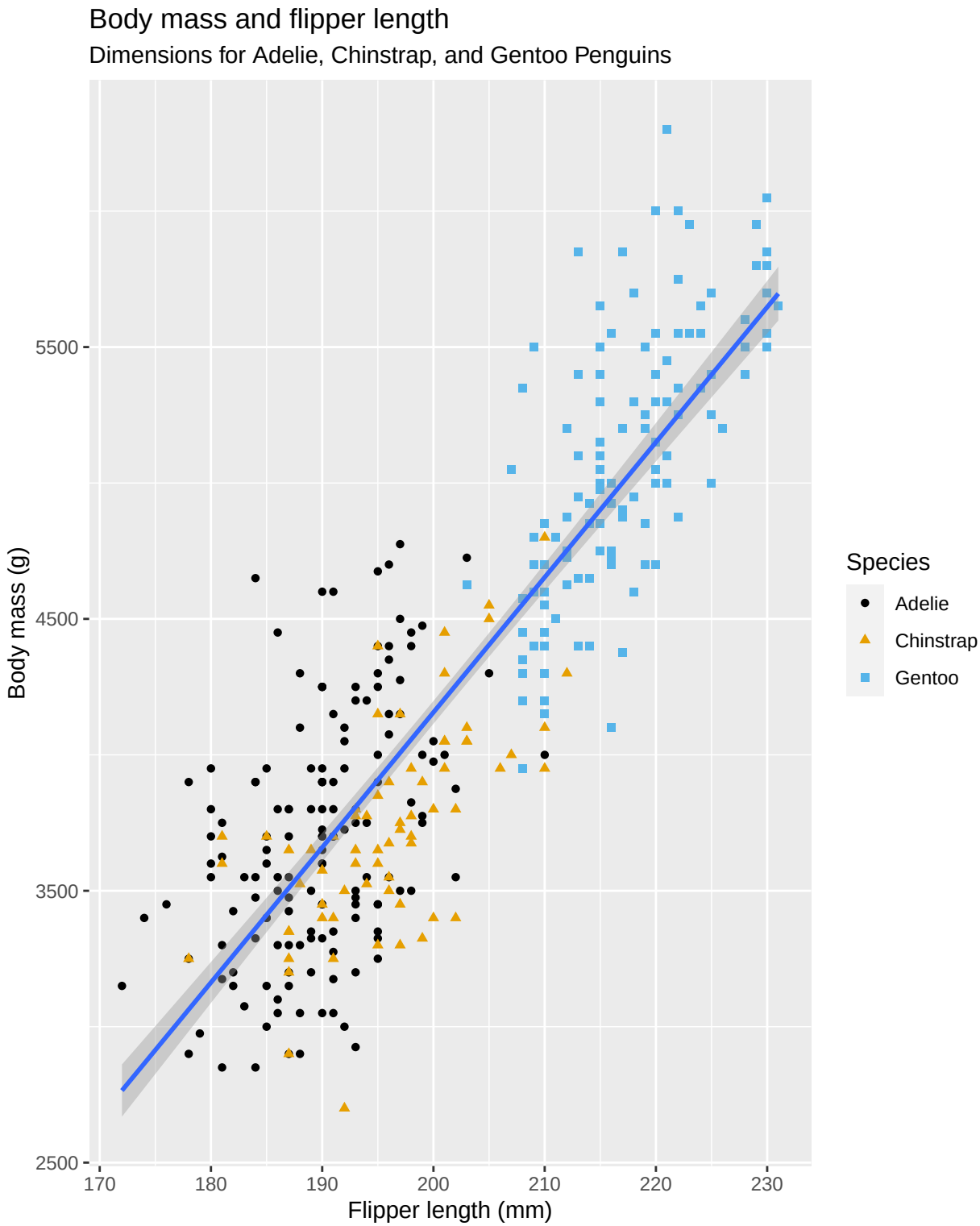
```r
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point(mapping = aes(color = species, shape = species)) +
  geom_smooth(method = "lm")
```



labs 함수를 사용하여 레이블을 수정 할 수 있습니다. labs에 대한 내용은 설명이 필요 없습니다. ggthemes 패키지의 scale_color_colorblind 함수를 사용하여 색상표가 색맹에 안전하도록 개선할 수 있습니다.

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
```

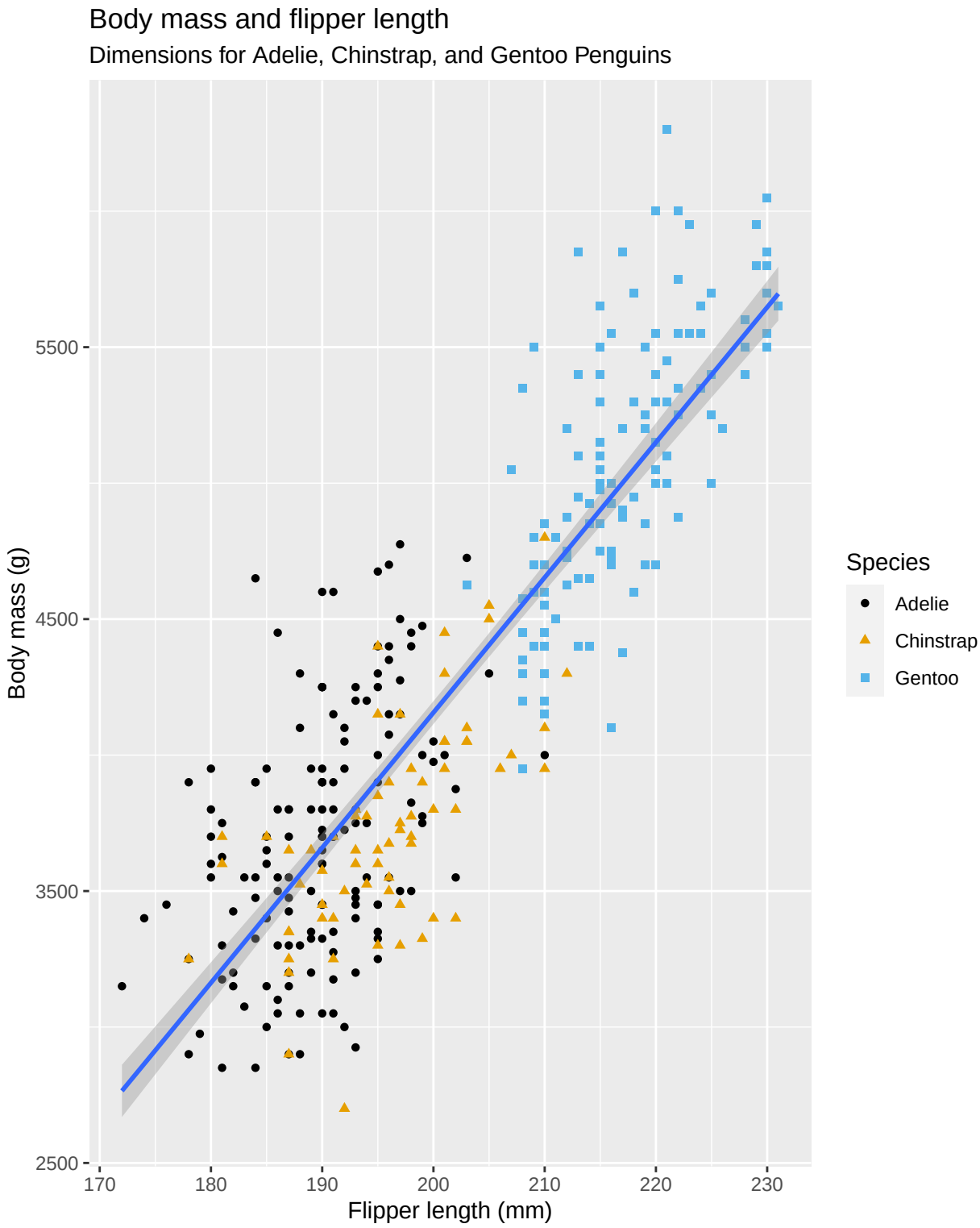
```
) +
  geom_point(aes(color = species, shape = species)) +
  geom_smooth(method = "lm") +
  labs(
    title = "Body mass and flipper length",
    subtitle = "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",
    x = "Flipper length (mm)",
    y = "Body mass (g)",
    color = "Species",
    shape = "Species"
  ) +
  scale_color_colorblind()
```



아래와 같이 %>% 또는 |>연산자를 사용할 수 있습니다. 하지만 이러한 형태의 코드를 작성하기 위해서 조금 다른 형태의 R 프로그래밍 구현에 대한 이해도가 필요합니다. R 사용에 관해서 혹시 많은 관심이 있으시다면 아래 코드 형태로 작성하는 방법을 연습하시는 것도 추천드립니다. 해당 내용은 25장에 소개되고 있습니다.

```
penguins |>
  ggplot(aes(x = flipper_length_mm, y = body_mass_g)) +
```

```
  geom_point(aes(color = species, shape = species)) +
  geom_smooth(method = "lm") +
  labs(
    title = "Body mass and flipper length",
    subtitle = "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",
    x = "Flipper length (mm)",
    y = "Body mass (g)",
    color = "Species",
    shape = "Species"
  ) +
  scale_color_colorblind()
```

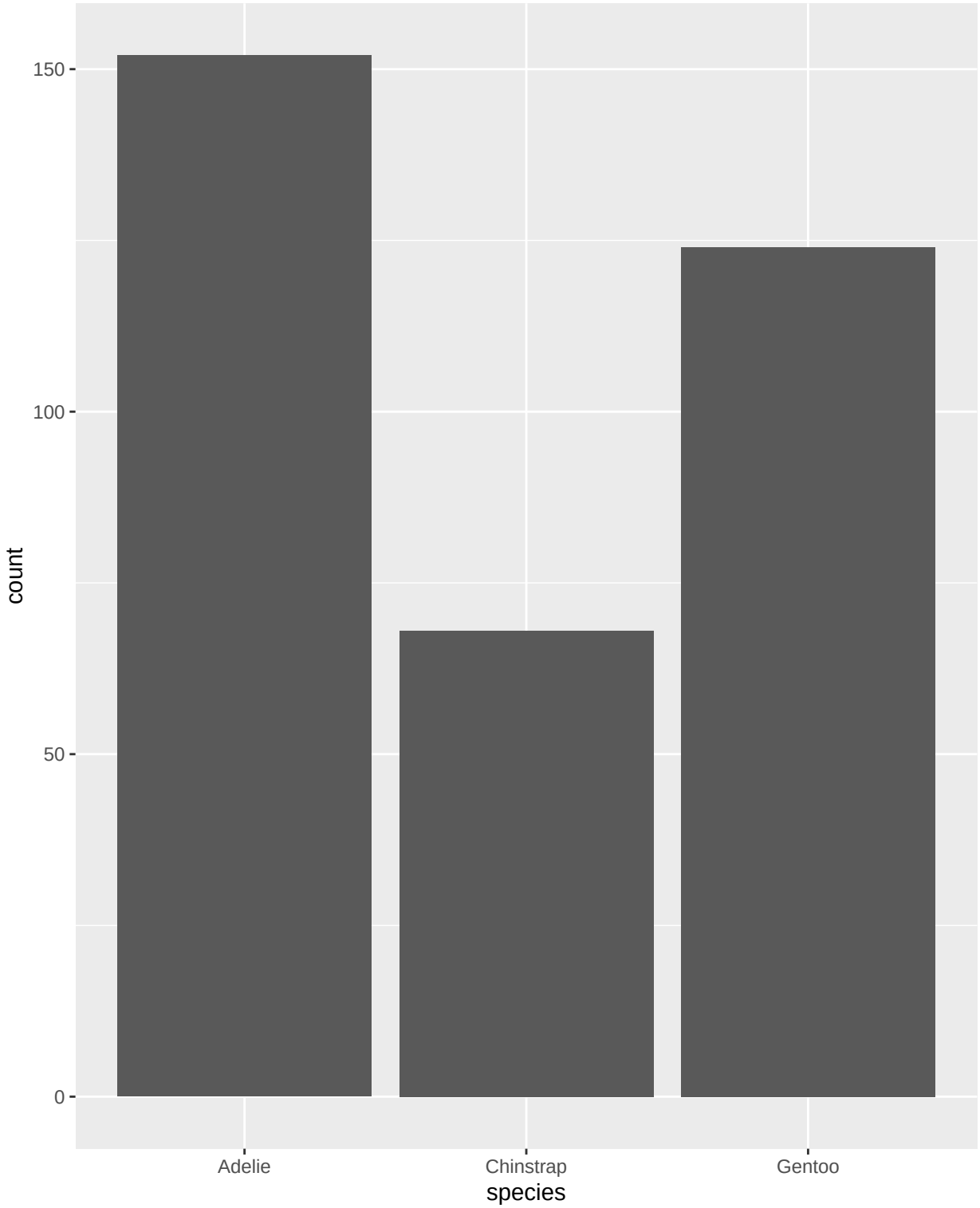


Visualizing distributions

- bar
- histogram
- density

막대 그래프를 사용해서 범주형 데이터를 표현할 수 있습니다. 만약 정렬이 필요하다면 `fct_infreq` 함수를 사용하시면 됩니다.

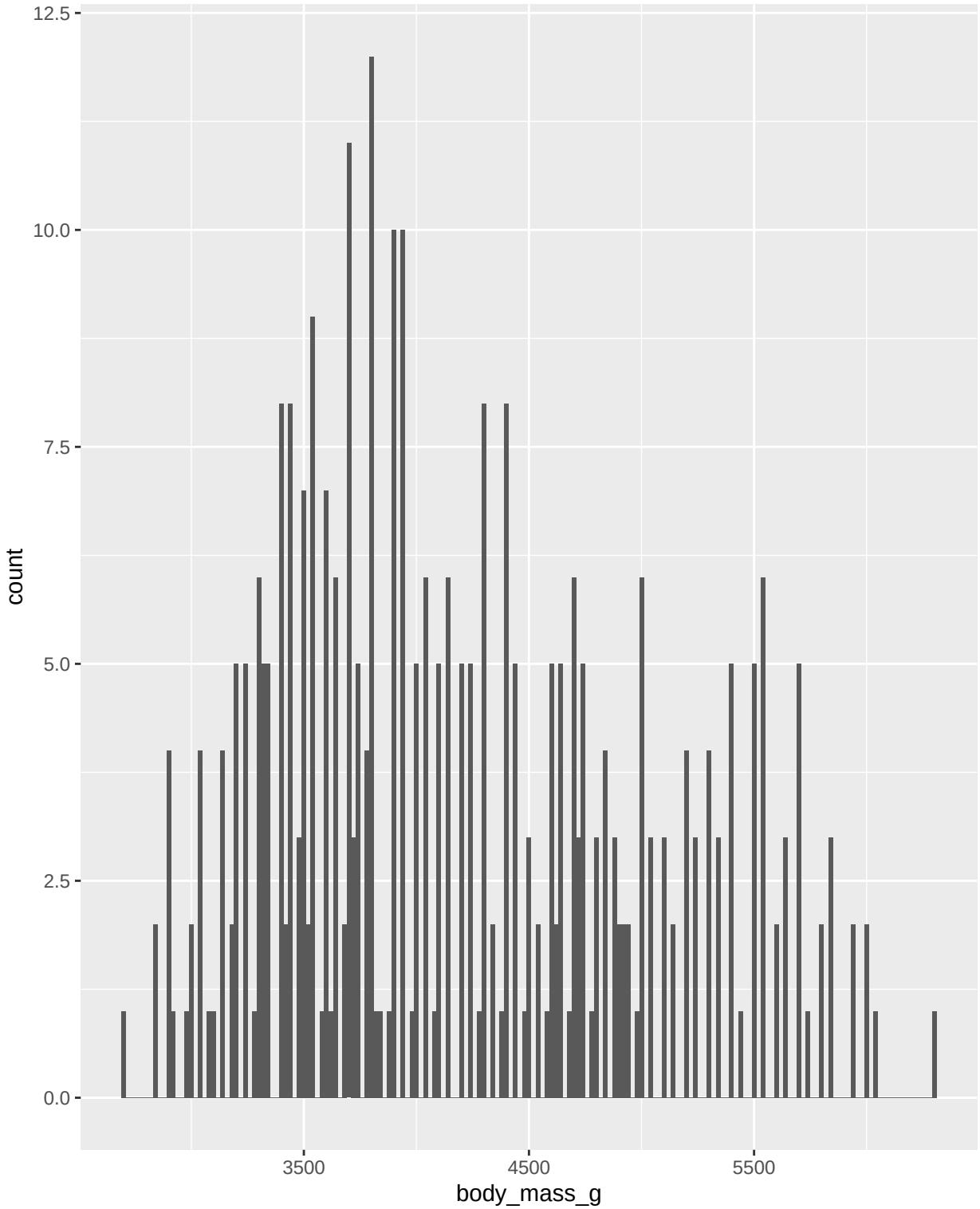
```
ggplot(penguins, aes(x = species)) +
  geom_bar()
```



변수가 숫자로 되었있고, 해당 값에 더하기, 빼기 또는 평균을 구하는 것이 합리적이라면 해당 변수를 수치형

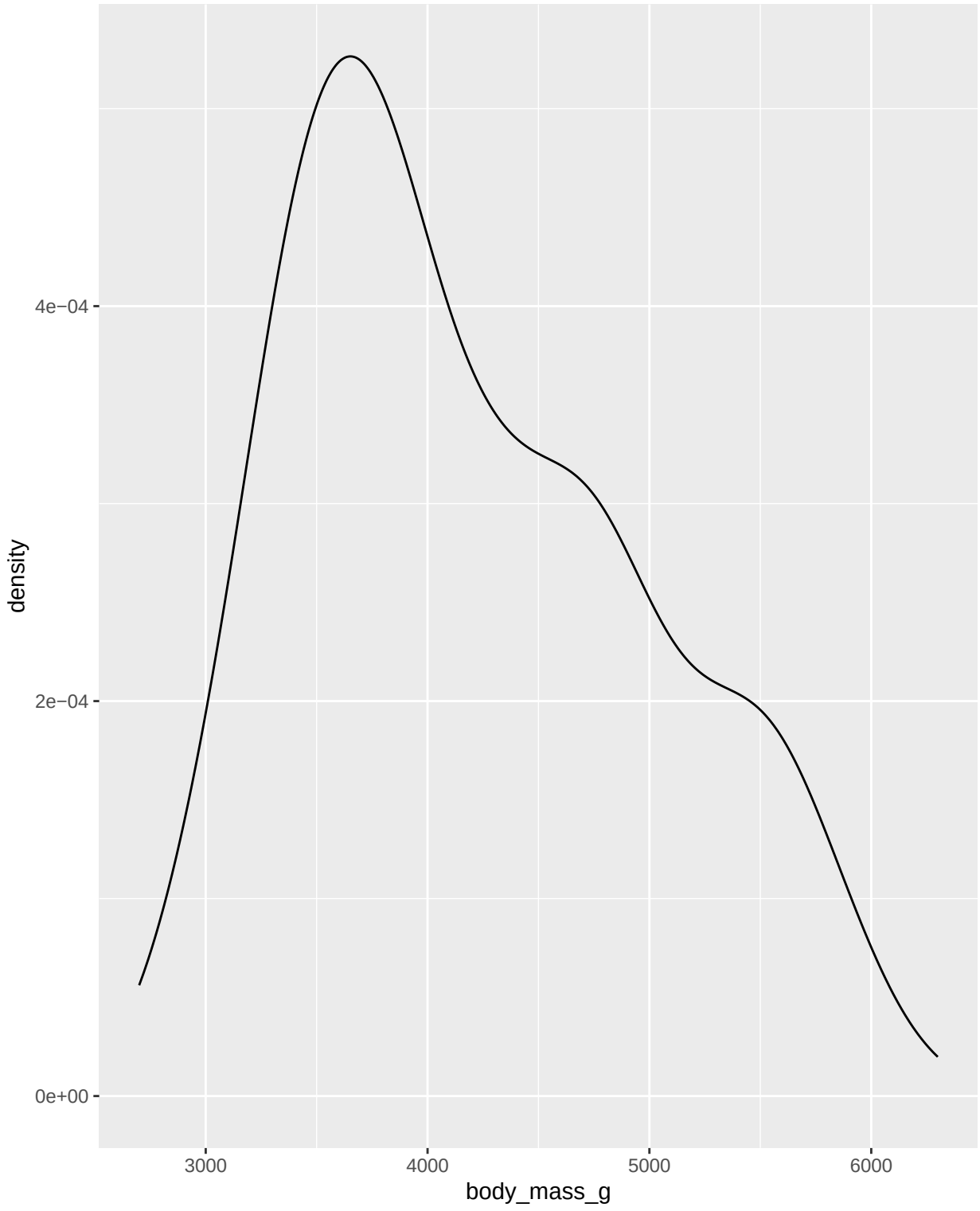
변수라 할 수 있습니다. 이 경우 히스토그램도 활용할 수 있습니다.

```
ggplot(penguins, aes(x = body_mass_g)) +
  geom_histogram(binwidth = 20)
```



만약, 해당 수치형 변수가 연속형이라면 밀도 그래프를 활용하세요.

```r
ggplot(penguins, aes(x = body_mass_g)) +
  geom_density()
```

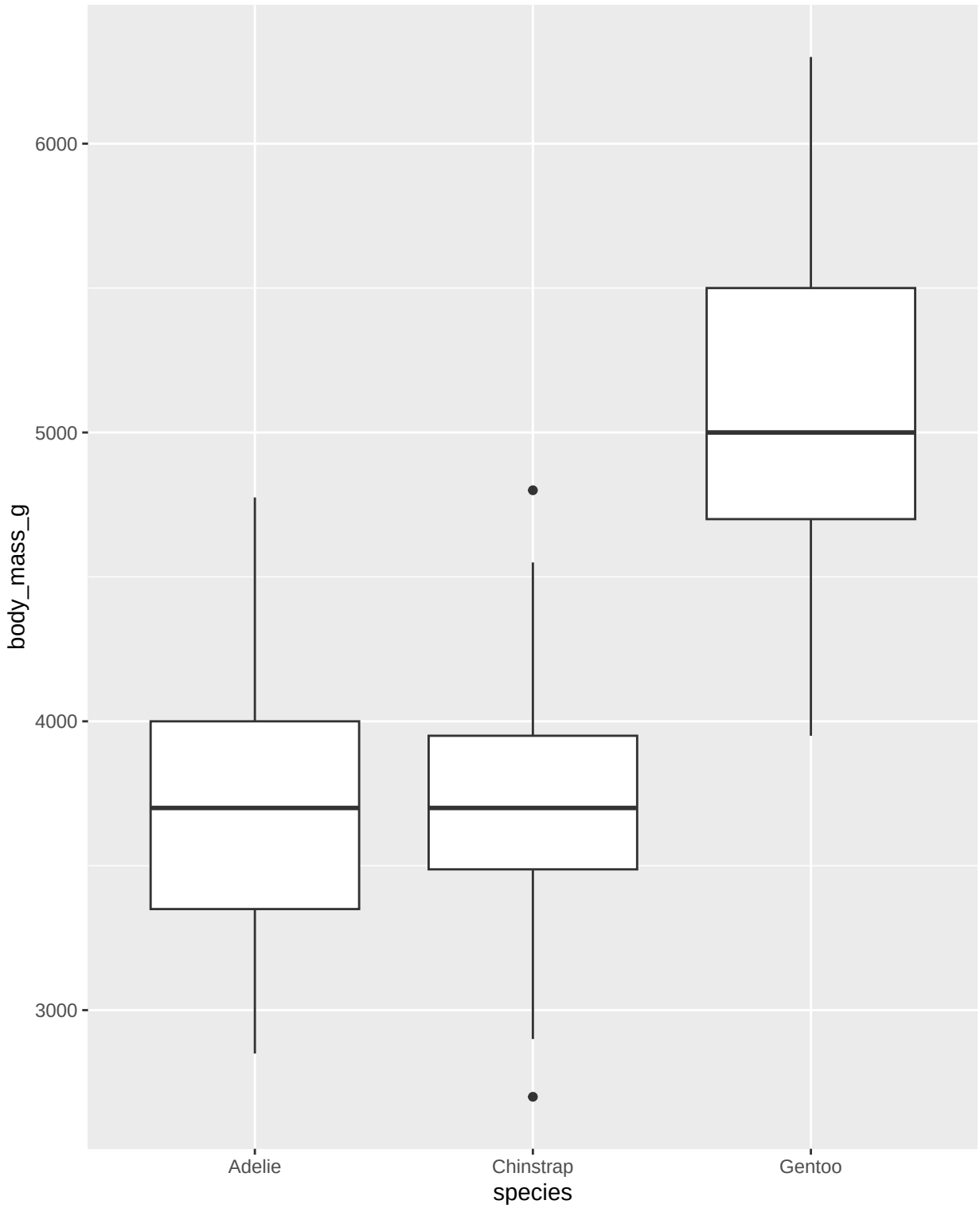


Visualizing relationships

- box
- density
- point

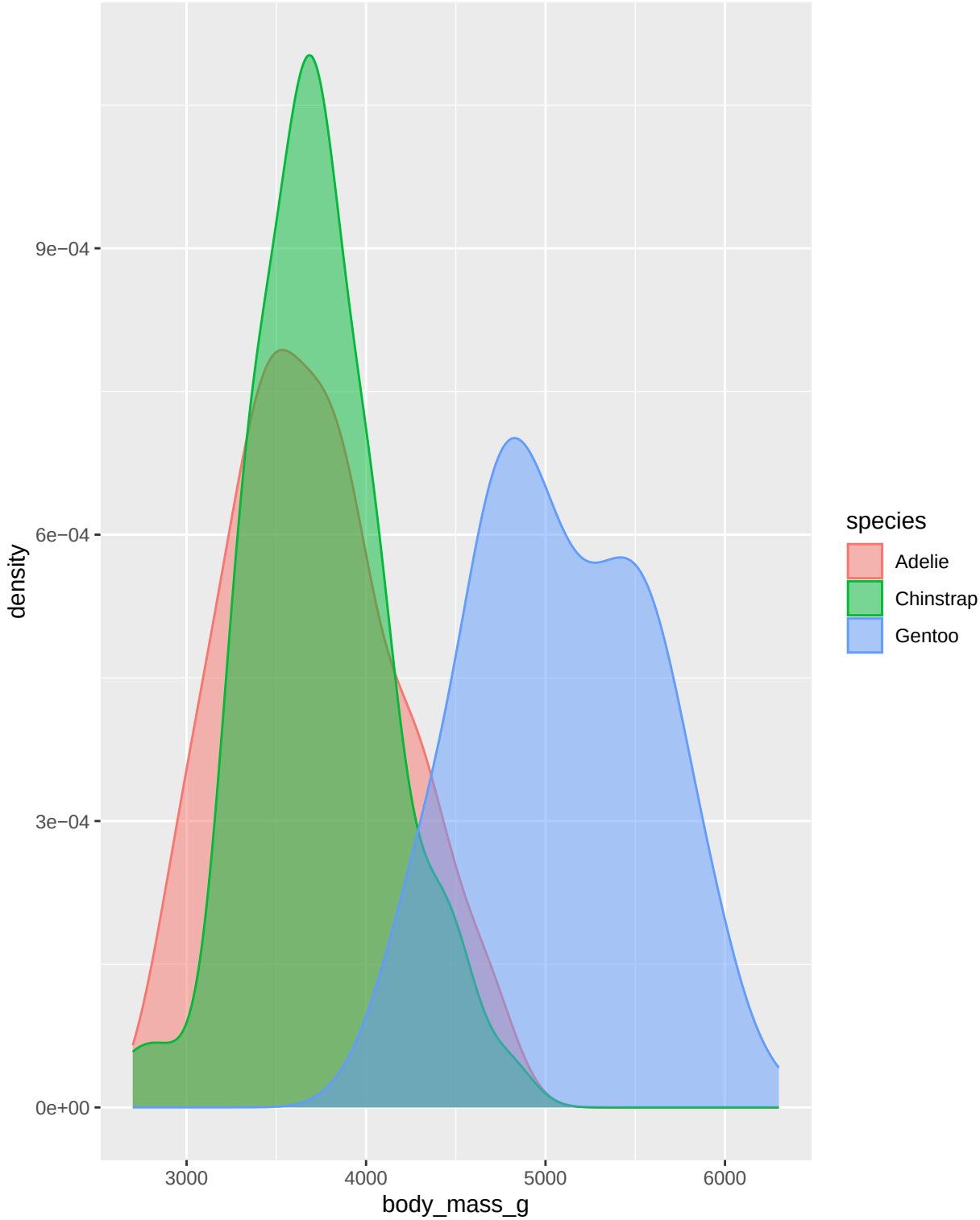
수치형 변수와 범주형 변수 간의 관계를 시각화하기 위해 나란히 배치된 박스 플롯을 사용할 수 있습니다. 박스 플롯의 경우 해석을 위해서 별도의 연습이 필요합니다. 사용 후 해석에 주의하세요.

```
ggplot(penguins, aes(x = species, y = body_mass_g)) +
  geom_boxplot()
```



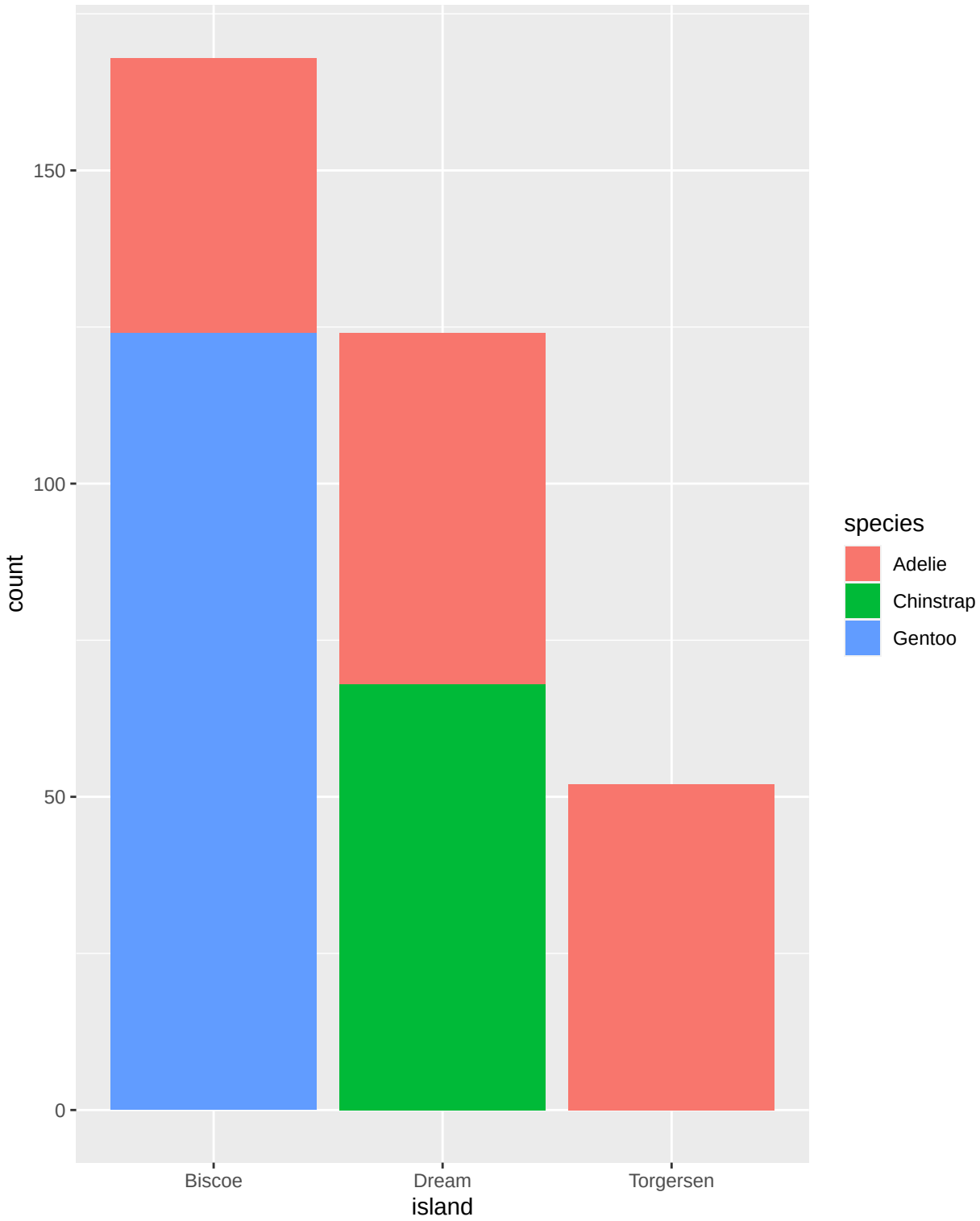
밀도 그래프를 활용하는 방법도 있습니다. 이 경우 비교를 위해서 색을 활용하세요.

```
ggplot(penguins, aes(x = body_mass_g, color = species, fill = species)) +
  geom_density(alpha = 0.5)
```



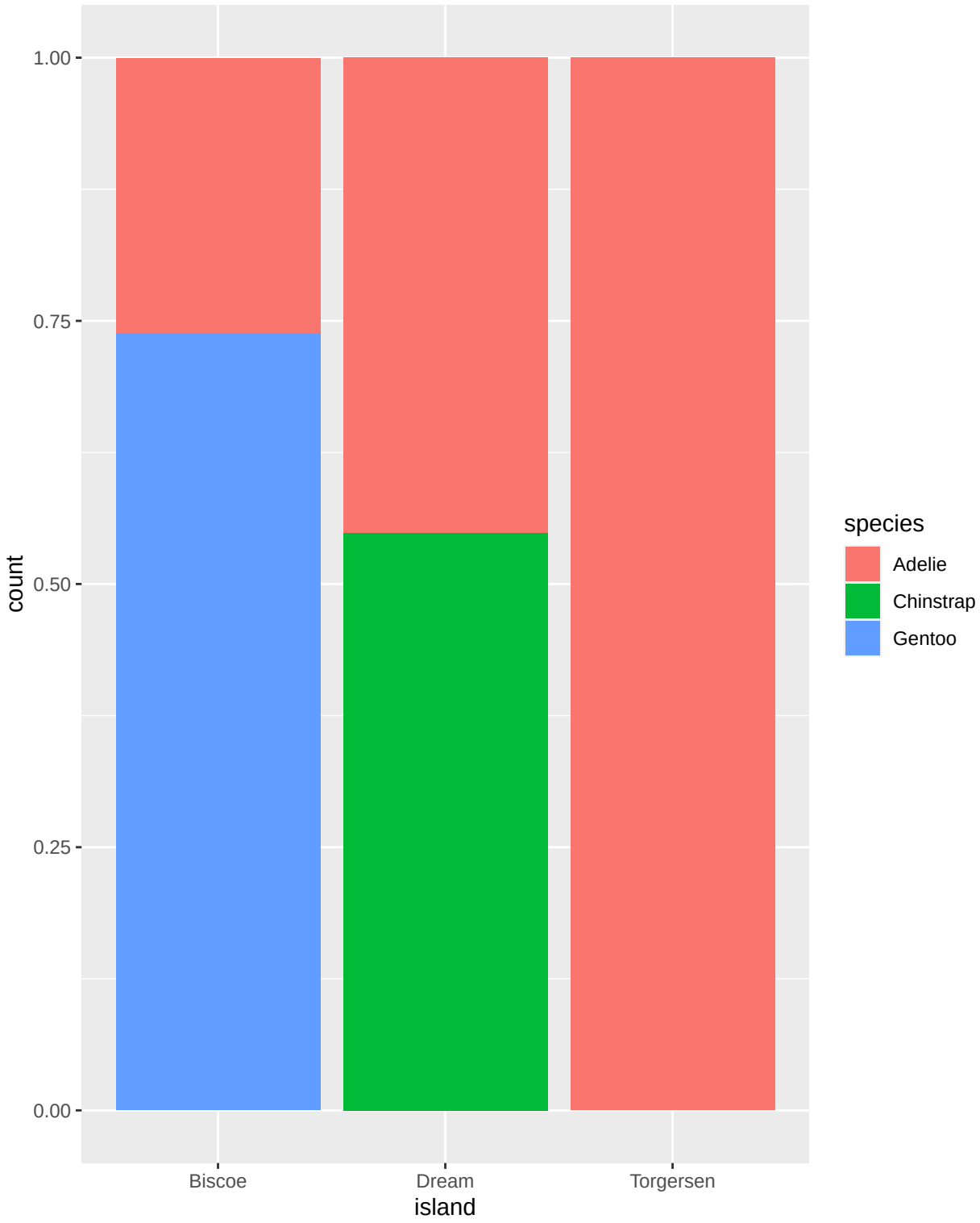
누적 막대형 차트를 사용하여 두 범주형 변수 간의 관계를 시각화할 수 있습니다.

```
ggplot(penguins, aes(x = island, fill = species)) +
  geom_bar()
```



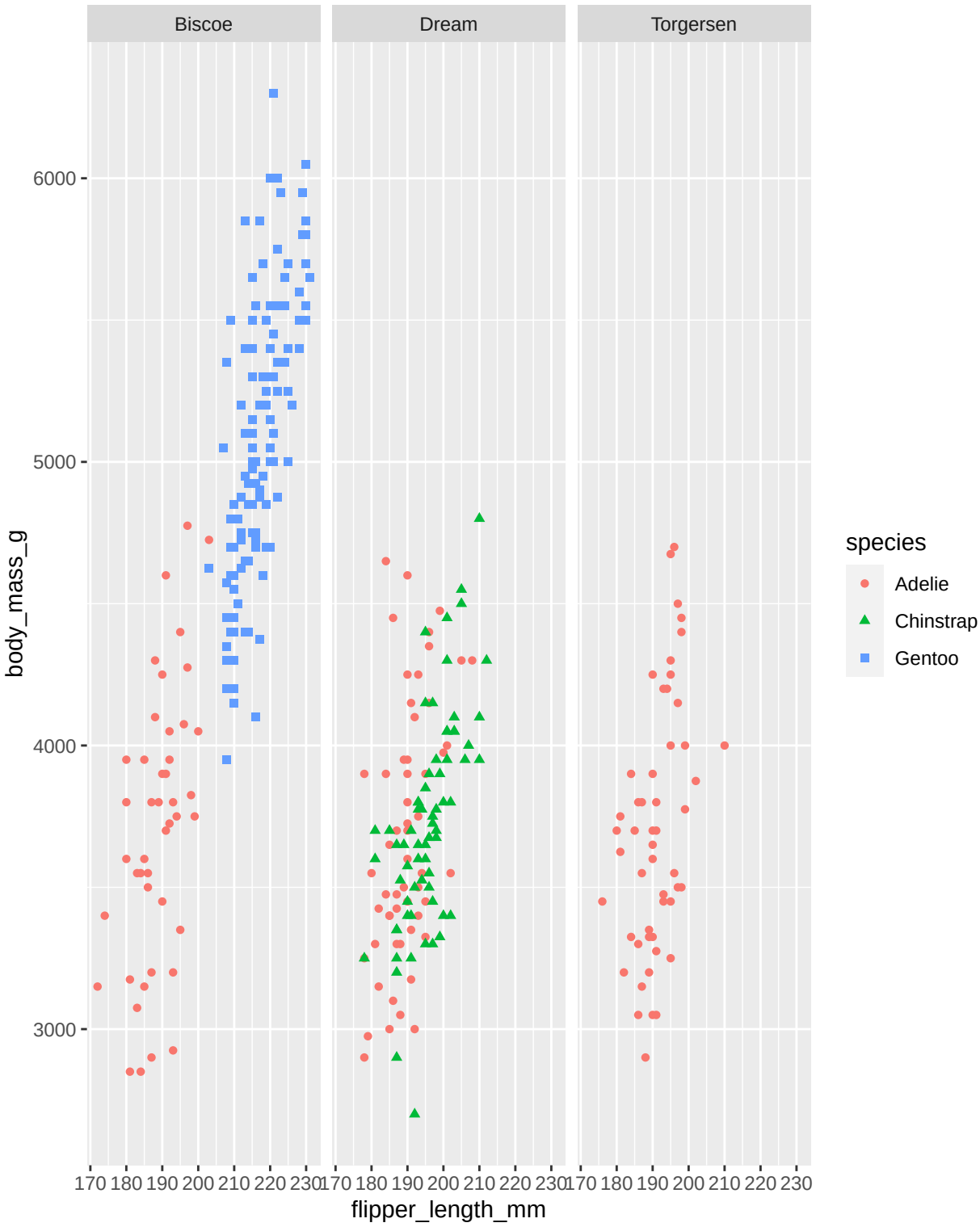
누적 막대형 그래프를 활용하면 아래와 같이 분포를 확인할 수 있습니다.

```
ggplot(penguins, aes(x = island, fill = species)) +
  geom_bar(position = "fill")
```



3가지 이상의 수치형 변수를 비교해야 된다면 앞서 소개한 산점도를 활용하세요.

```
ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point(aes(color = species, shape = species)) +
  facet_wrap(~island)
```



Saving your plots

ggsave 함수를 사용하면 이미지를 저장할 수 있습니다.

```r
ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point()
ggsave(filename = "penguin-plot.png")
```

소감

데이터가 주어졌을 때 간단한 그래프를 먼저 그릴 수 있도록 하는 과정입니다. R 문법에 대한 소개없이 곧바로 그래프를 그려서 조금 어렵게 느껴지긴 하지만, 그래프와 관련된 간단한 문법을 먼저 익혀서 빠르게 데이터를 활용하는 방법으로 나쁘지 않은 듯 합니다. ggplot에 대한 호기심이 다들 많아서 즐거운 스터디였습니다. ggplot은 R4DS가 끝나고 여유가 되면 진행보면 좋을 듯 합니다.