Atividade 2 - Lista de exercícios 2

▼ 1. Quantas linhas tem o penguins? E quantas colunas?

> Format: A tibble with 344 rows and 8 variables

Ou seja, o banco de dados “penguins” tem 344 linhas e 8 colunas.

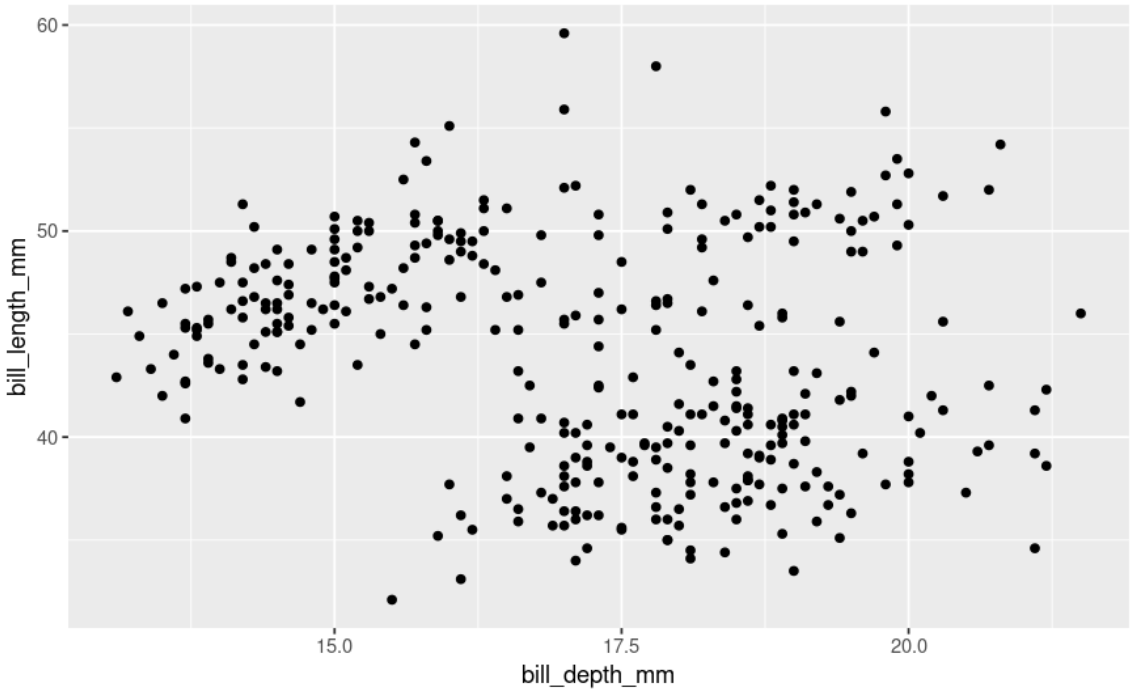
▼ 2. O que a variável bill_depth_mm no penguins descreve? (dica: use a função ? penguins)

> bill_depth_mm
>
> a number denoting bill depth (millimeters)

Descreve o valor em milímetros da “profundidade” do bico dos pinguins.

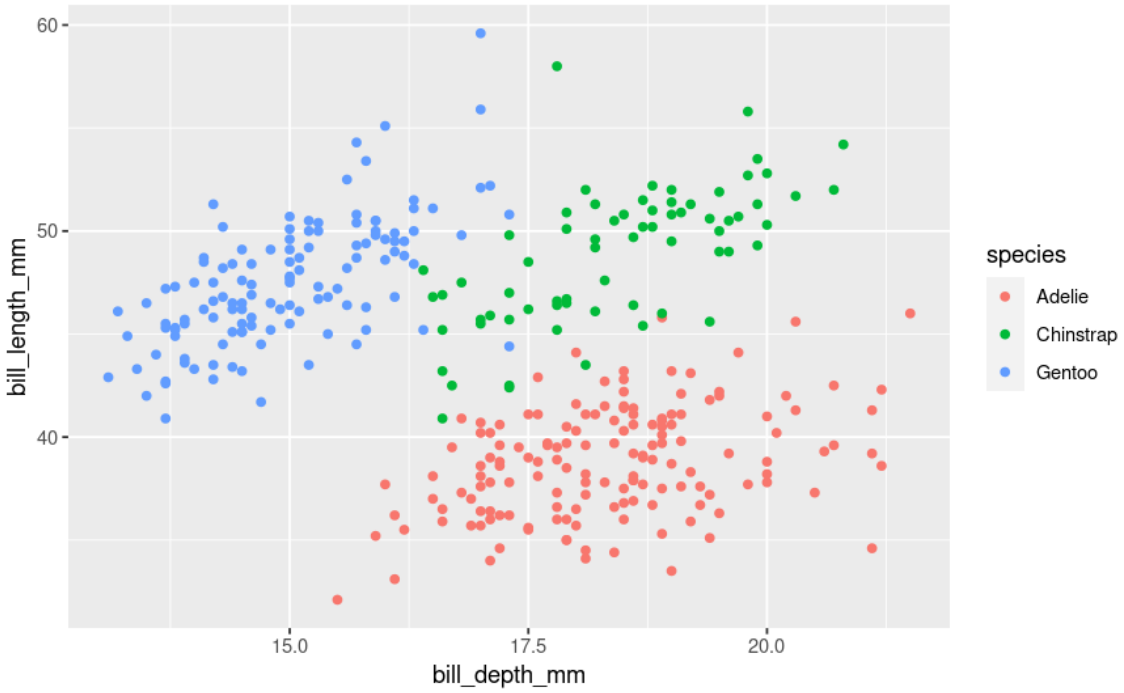
▼ 3. Faça um gráfico de dispersão (scatterplot) do bill_depth_mm vs bill_length_mm. Descreva a relação entre essas duas variáveis

```
ggplot(
  data=penguins,
  mapping=aes(
    x=bill_depth_mm,
    y=bill_length_mm,
    # color=species
    )
  ) +
geom_point()
```



Inicialmente, os dois valores não parecem estar correlacionados por se apresentarem de forma muito dispersa. Entretanto, percebe-se alguns agrupamentos específicos. No canto superior esquerdo há um grupo de pontos mais denso, na parte central inferior há um outro grupo mais espalhado, já no canto superior direito há um grupo menor mas também com algum grau de proximidade entre eles.

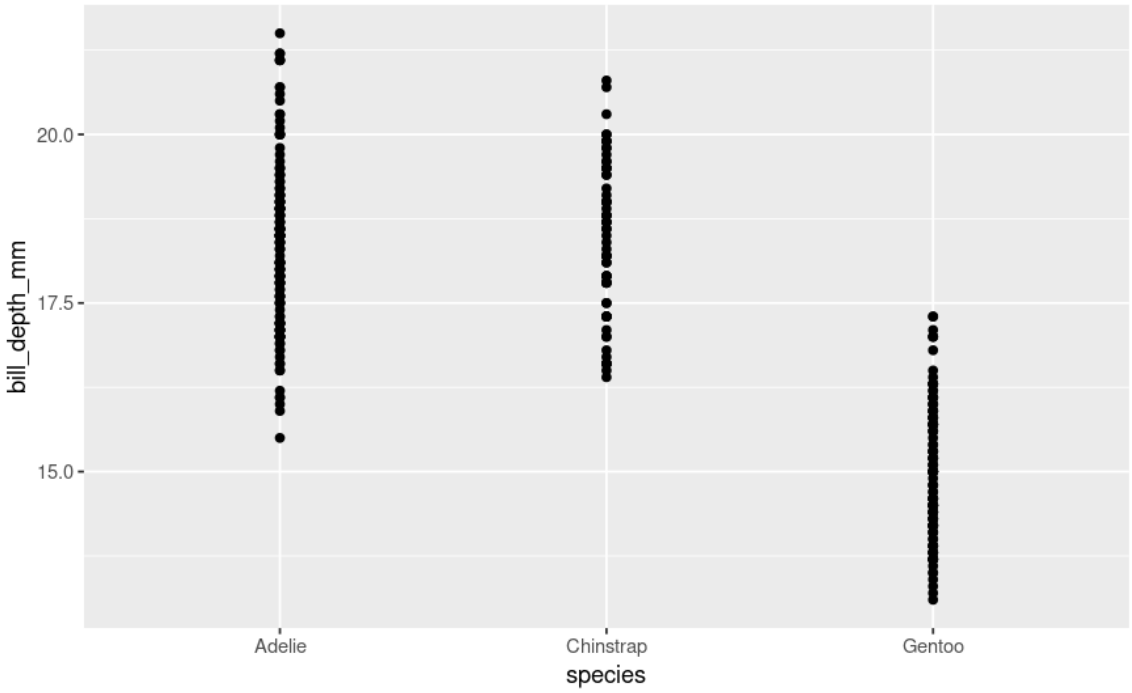
Ao colorí-los por espécie, conseguimos melhor o que representam esses grupos:



O primeiro grupo é constituído principalmente pelos Gentoo, o segundo pelos Adelie e o terceiro pelos Chinstrap.

▼ 4. O que acontece se você fizer um gráfico de dispersão (scatterplot) de species vs bill_depth_mm? Esse gráfico pode ser útil?

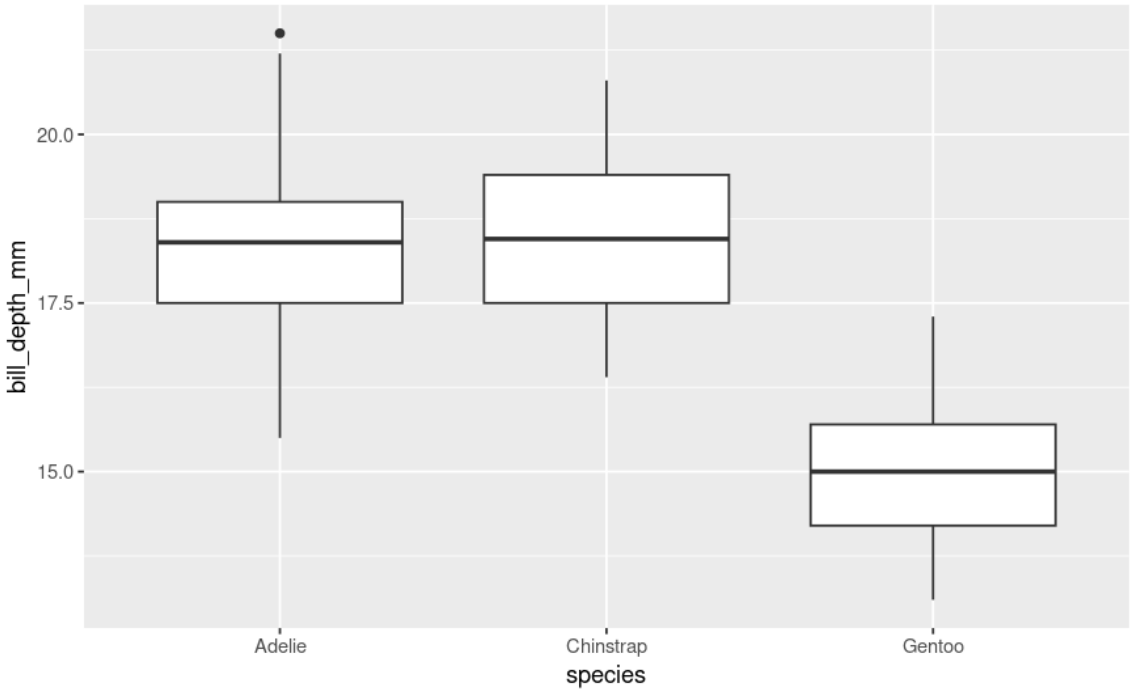
```
ggplot(
  data=penguins,
  mapping=aes(
    x=species,
    y=bill_depth_mm
  )
) +
geom_point()
```



O que acontece é que as espécies ficam distribuídas no eixo X e o “bill depth” fica no eixo y. Entretanto, os pontos “plotados” ficam muito próximos uns dos outros, assim dificultando a visualização.

Este gráfico pode sim ser útil, mas não parece ser o ideal para analisarmos esses dados. Um “boxplot” pode vir a ser mais útil para analisarmos melhor a distribuição referente a estes dados.

```
ggplot(
  data=penguins,
  mapping=aes(
    x=species,
    y=bill_depth_mm
  )
) +
geom_boxplot()
```



▼ 5. Por que o código dá erro e como poderíamos resolver?

> ggplot(data = penguins) + geom_point()

O erro mostrado é:

> Caused by error in `compute_geom_1()`: ! `geom_point()` requires the following missing aesthetics: x and y

Que nos informa que para realizar o comando `geom_point()` é necessário que haja o argumento “aes” (thetics) contendo os valores que deverão estar nos eixos x e y. E é esse o motivo pelo qual o código dá erro.

Para resolver, o código precisaria ser algo como:

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

Ou seja, adicionando à função `ggplot` o parâmetro `mapping` que receberá o valor da função `aes()` contendo os parâmetros `x` e `y`, cada um deles tendo como valor a listagem de valores de uma das colunas do banco de dados `penguins`.

▼ 6. O que o argumento na.rm faz no geom_point()? Qual é o valor padrão desse argumento? Crie um gráfico de dispersão (scatterplot) onde você usa com sucesso este argumento definido como TRUE.

▼ O que o argumento na.rm faz no geom_point()?

> **na.rm**
> If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

NA.RM significa a remoção (Removal(RM)) de dados não disponíveis/nulos (Not Available). Dessa forma o argumento na.rm serve para “remover silenciosamente”/”não mostrar” os avisos de valores faltantes/nulos se seu valor estiver definido para `TRUE` senão, ele mostrará um aviso sempre que houverem observações removidas por estarem nulas.

▼ Qual é o valor padrão desse argumento?

O seu valor padrão é FALSE

▼ Crie um gráfico de dispersão (scatterplot) onde você usa com sucesso este argumento definido como TRUE

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

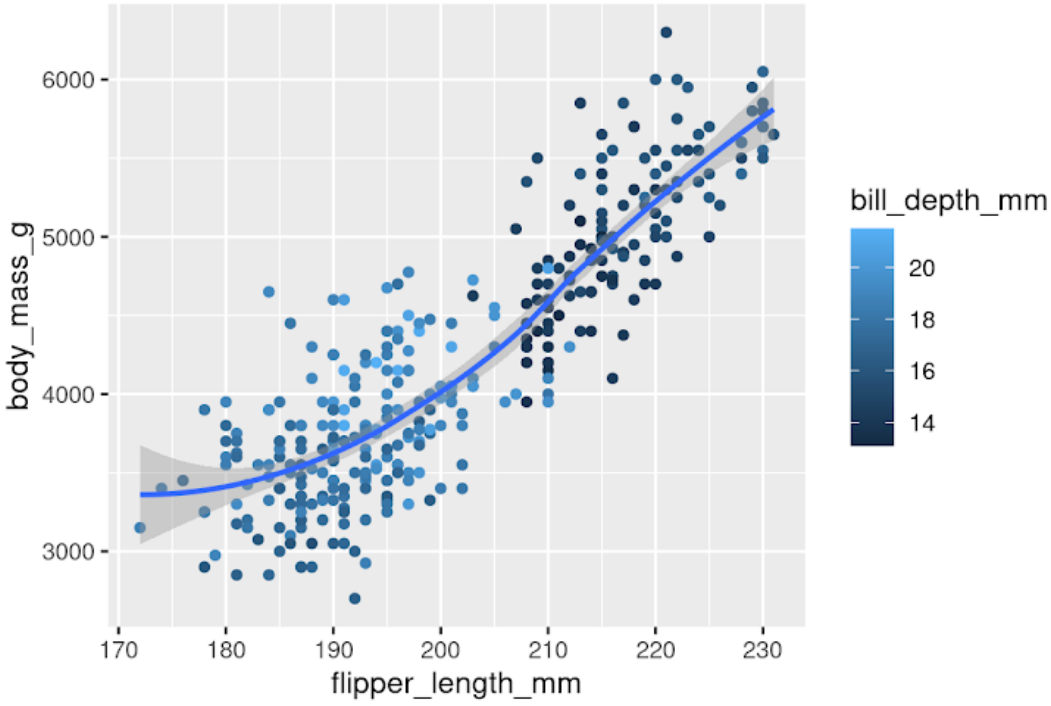
▼ 7. Adicione a seguinte legenda ao gráfico que você fez no exercício anterior: “Os dados vêm do pacote palmerpenguins”. Dica: dê uma olhada na documentação de labs(). [código]

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

```
  mapping = aes(
    x = body_mass_g,
    y = flipper_length_mm
  )
) +
geom_point(
  na.rm = TRUE,
) +
labs (
  caption = "Os dados vêm do pacote palmerpenguins",
)
```

▼ 8. Recrie a seguinte visualização. Para qual estética o bill_depth_mm deve ser mapeado? E deve ser mapeado no nível global ou no nível geom?

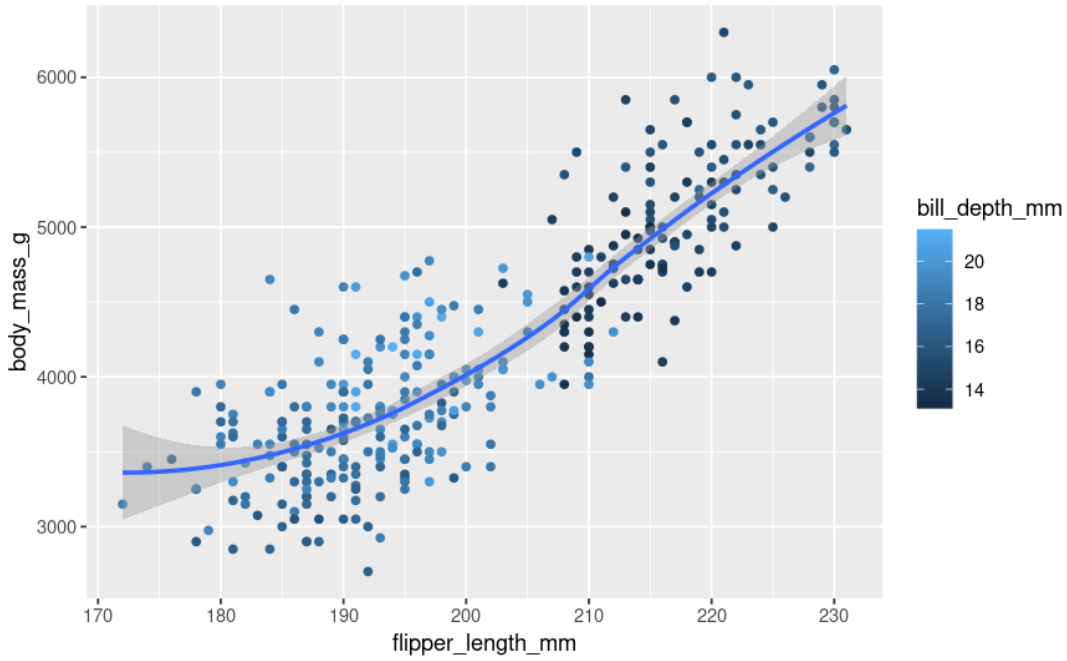
▼ Visualização alvo



▼ Visualização atingida

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

```
  na.rm = TRUE,
)
```



▼ Para qual estética o bill_depth_mm deve ser mapeado?

O bill_depth_mm deve ser mapeado para a estética color no ggplot.

▼ E deve ser mapeado no nível global ou no nível geom?

Deve ser mapeado no nível global

▼ 9. Execute esse código em sua cabeça e preveja como será a saída. Em seguida, execute o código em R e verifique suas previsões. Foi o que você esperava?

▼ Código

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g, color = island)
) +
  geom_point() +
  geom_smooth(se = FALSE)
```

Prevejo que será feito um gráfico baseado nos dados `penguins` onde o eixo X representará `flipper_length_mm` e o eixo Y representará `body_mass_g`, e os pontos serão coloridos baseado em sua localização por ilha. O gráfico será o

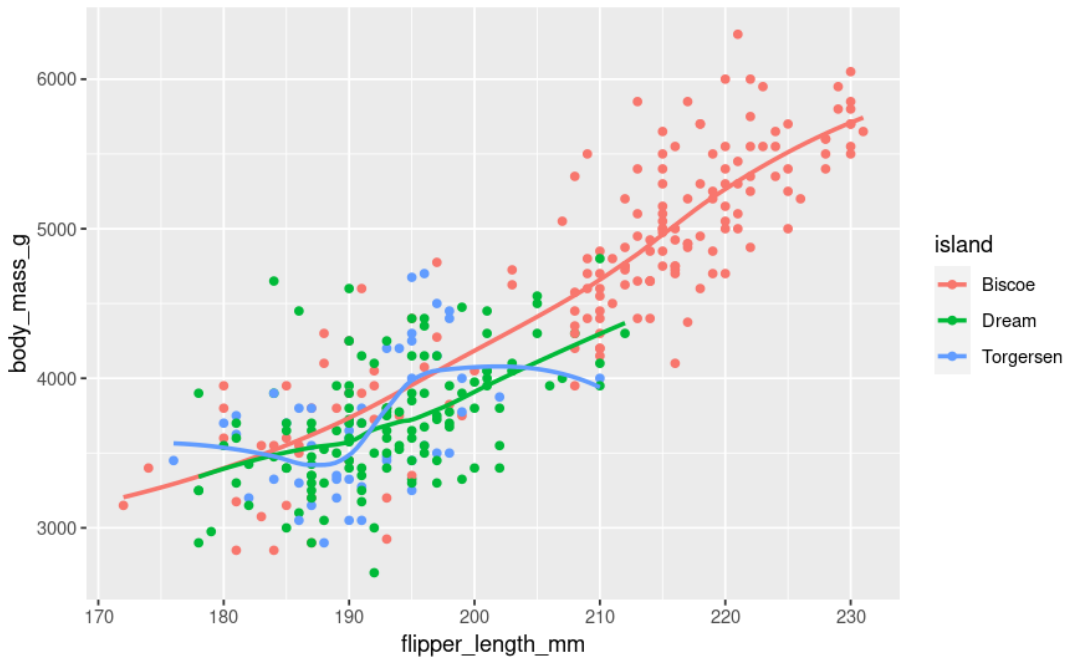
distribuição de pontos. Mas não sei o que o valor de “se” sendo falso causaria na exibição do gráfico. Vou pesquisar.

> **se**
>
> Display confidence interval around smooth? (`TRUE` by default, see `level` to control.)

Nesse caso, suponho que ele não mais apresentará a sombra escura que representa o intervalo de confiança. Restando apenas a linha azul.

Não consegui encontrar o que o “SE” significa de fato.

▼ Resultado da execução



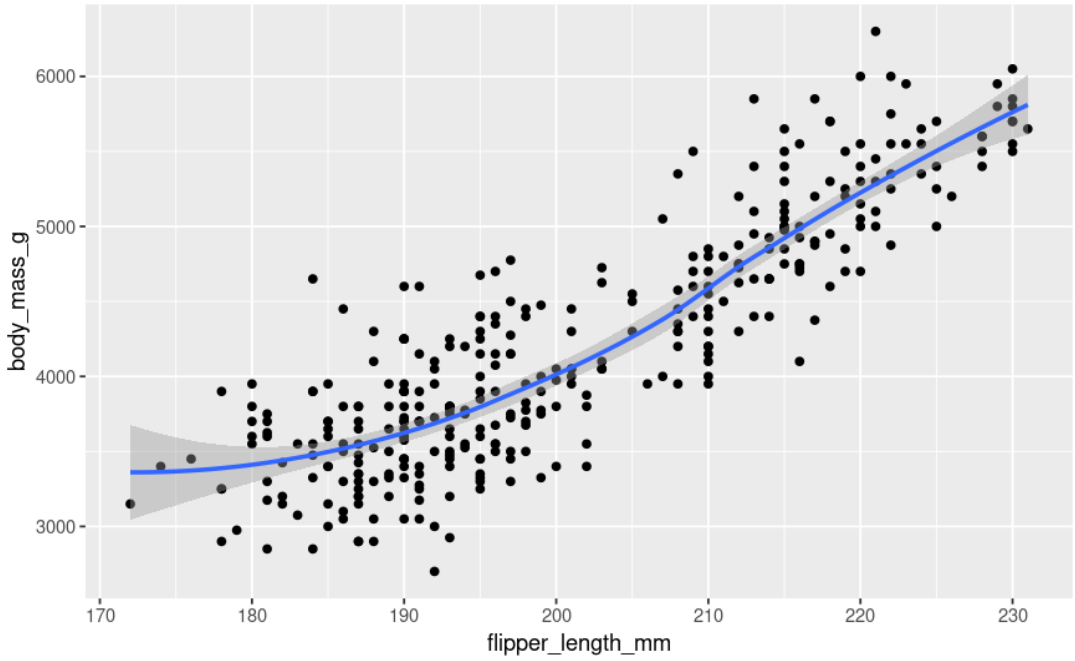
Não foi que eu esperava. Não imaginava que o gráfico fosse se comportar de maneira tão diferente do anterior visto que as únicas mudanças relevantes seriam a definição do SE como `FALSE` e a mudança de variável a ser colorida. Aparentemente, por causa da quantidade de dados ilustrados pela cor deixando de ser uma variável numérica para ser uma categórica, o gráfico passou a gerar uma linha para cada uma das categorias. Situação não prevista por mim.

▼ 10. Esses códigos geram gráficos iguais ou diferentes? Por que?

▼ Código A

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

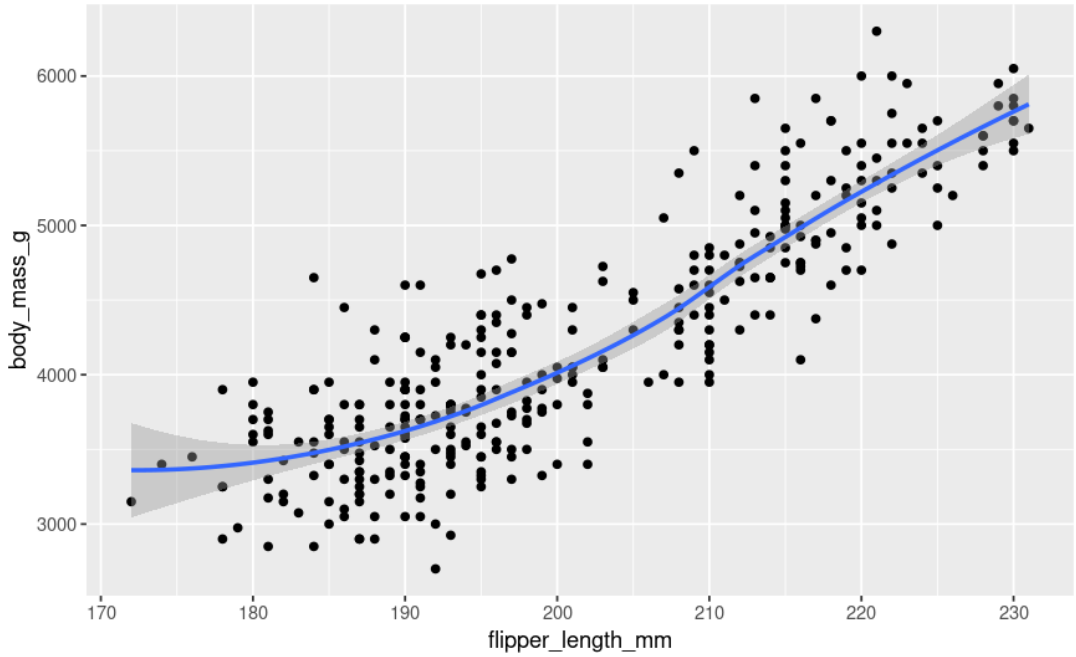
▼ Gráfico Gerado A



▼ Código B

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

▼ Gráfico Gerado



Os códigos geram gráficos iguais. Isso se dá porque no primeiro caso, um gráfico base é definido usando um certo conjunto de dados e uma certa distribuição de suas colunas nos eixos. Essas propriedades são utilizadas também nas alterações que são feitas em seguida ao adicionar a distribuição dos pontos e a curva. Já no segundo caso, como o gráfico base não define esses valores base, eles precisam ser replicados em cada uma das alterações adicionadas.