

# Hayes\_Metzger

2026-02-02

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.3.0
v lubridate  1.9.4     v tidyrr    1.3.1
v purrr     1.0.4
-- Conflicts -----
x dplyr::filter() masks stats::filter()
x dplyr::lag()   masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```
library(here)
```

```
here() starts at /Users/jacobmetzger/Documents/School/UCSC/Years/25to26/Courses/BIOE276 ds4ed
```

```
x <- 11

# Check if the value of x is greater than 10
if(x > 10)
{
  # Paste takes the value stored in x and combines that with a character string
  print(paste(x, "is greater than 10"))
} else
{
  print(paste(x, "is less than or equal to 10"))
}
```

```
[1] "11 is greater than 10"
```

## Q1.1

```
[1] "11 is greater than 10"

x <- "five"

# Check if the value of x is greater than 10
if(x > 10)
{
  # If x is > 10, multiple x by 2
  print(x*2)
} else
{
  # If x is not > 10, divide x by 2
  print(x/2)
}

# define a variable
x <- 11

# check the value of x using nested if-else statements
if (x < 10) {
  # if x is less than 10
  print("x is less than 10")
} else {
  # if x is exactly equal to 10
  if (x == 10) {
    print("x is 10!!!")
  } else {
    # if x is greater than 10
    print("x is greater than 10")
  }
}
```

```
[1] "x is greater than 10"
```

```
vec <- c(9, 10, 11, 12)

# For 1 through the length of the vector "vec"
for (i in 1:length(vec)) {
```

```

# check the value of using nested if-else statements
if (vec[i] < 10) {
  # if the element is less than 10
  print("value is less than 10")
} else {
  # if the element is exactly equal to 10
  if (vec[i] == 10) {
    # if the element equals 10
    print("value is 10!!!")
  } else {
    # if the element is greater than 10
    print("value is greater than 10")
  }
}

}

```

```

[1] "value is less than 10"
[1] "value is 10!!!"
[1] "value is greater than 10"
[1] "value is greater than 10"

```

### Q.1.2

```

#First we're going to make a vector = y
y <- c(-2, 42, 0, 10)

# For 1 through the length of the vector "vec"
for (i in 1:length(y)) {

  # check the value of using nested if-else statements
  if (y[i] < 0) {
    # if the element is less than 10
    print("value is Negative :(")
  } else {
    # if the element is exactly equal to 10
    if (y[i] == 0) {
      # if the element equals 10
      print("value is 0 my hero!!!")
    } else {
      # if the element is greater than 10

```

```
        print("value is 10 or the answer to life and the unviverse and everything #slay")
    }
}

}
```

```
[1] "value is Negative :("
[1] "value is 10 or the answer to life and the unviverse and everything #slay"
[1] "value is 0 my hero!!!"
[1] "value is 10 or the answer to life and the unviverse and everything #slay"
```

```
library(lterdatasampler)
library(tidyverse)
```

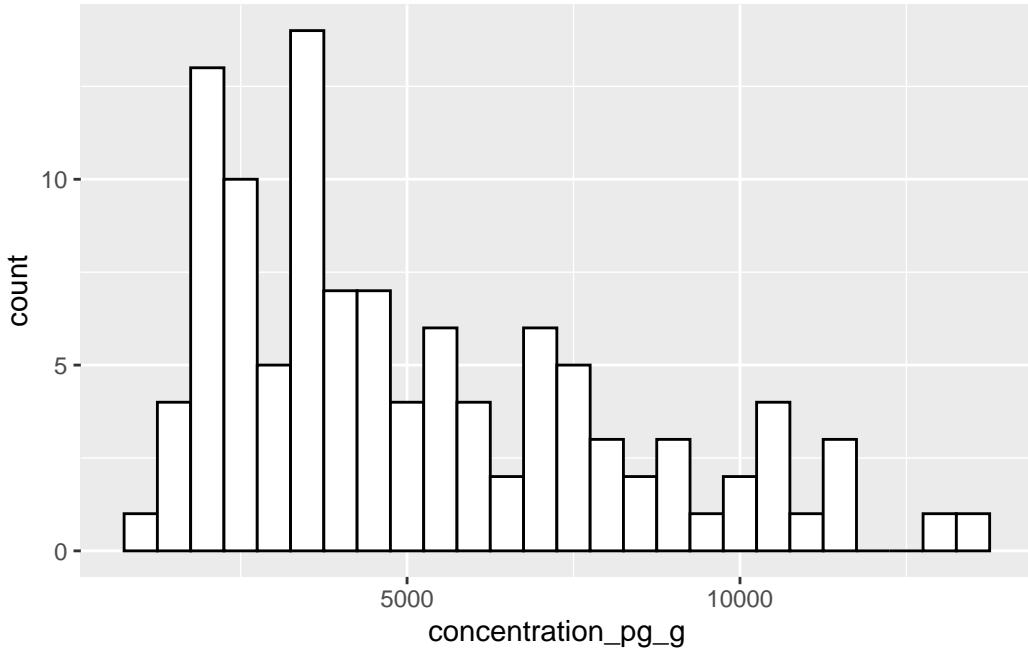
### Q1.3

Stress was measured by observing the amount of glucocorticoid metabolite present in pika feces. The stress variable is called “concentration\_pg\_m” and it’s in picograms GCM/gram of dry pika feces.

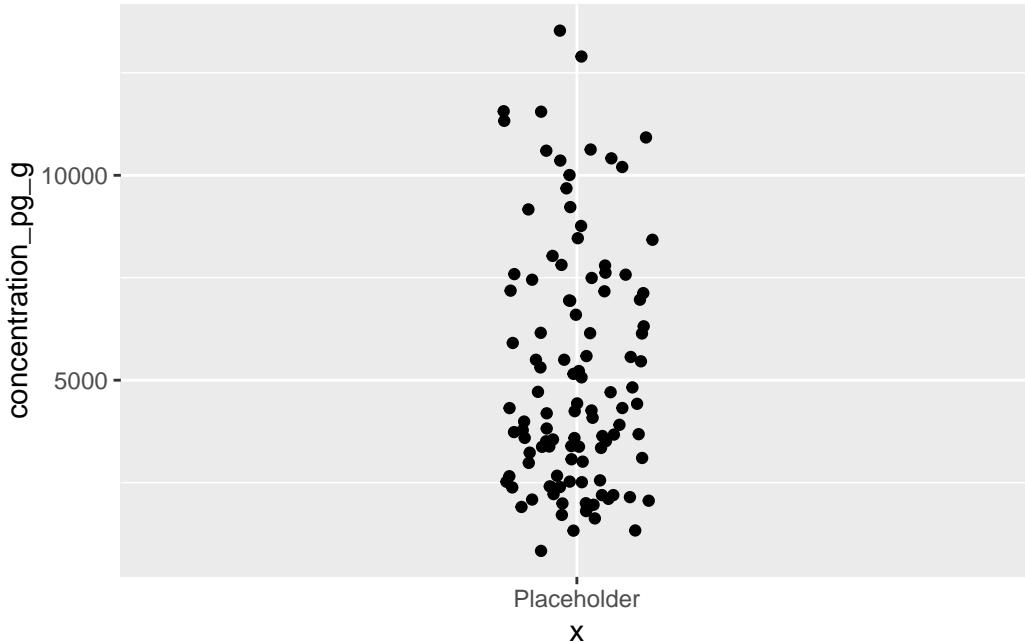
### Q1.4

A row represents a observation of individual pika poop, collected between June and September of 2018.

```
# Make a histogram
nwt_pikas %>%
  ggplot(aes(x = concentration_pg_g)) +
  # Add the histogram geom, which only needs an x-axis
  # Choose a binwidth of 500 picogram GCM/gram
  geom_histogram(binwidth = 500,
                 fill = "white",
                 color = "black")
```



```
# Make a scatterplot with jittered points
nwt_pikas %>%
  # We're adding a little placeholder axis just so we can see the point distribution
  ggplot(aes(x = "Placeholder",
             y = concentration_pg_g)) +
  # Add the geom_jitter geom
  geom_jitter(width = 0.1)
```

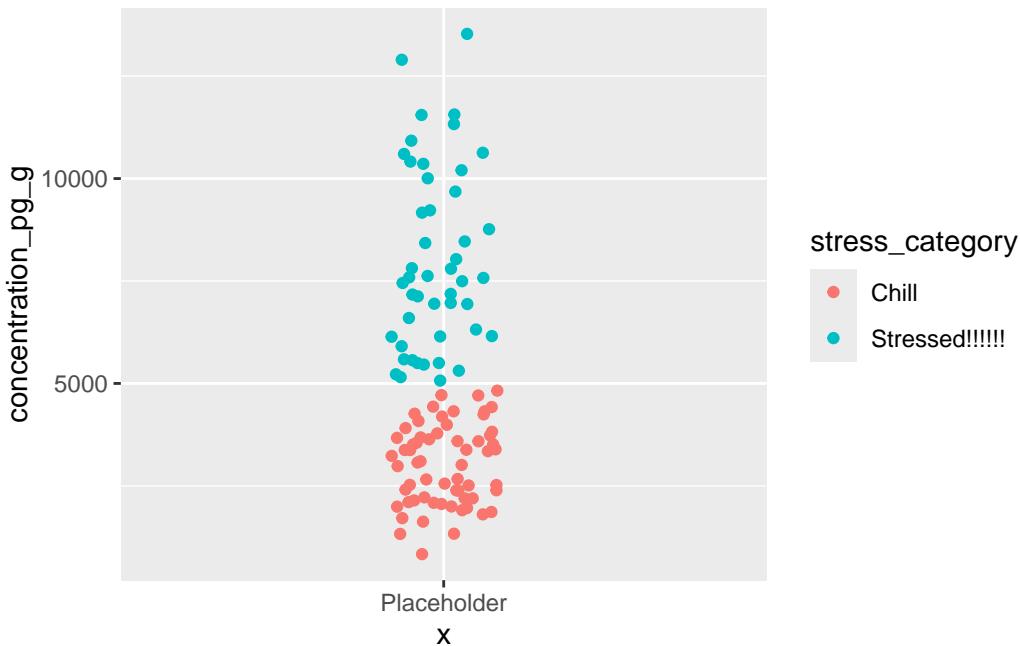


```
nwt_pikas_categ <- nwt_pikas %>%
  # Call the new column stress_category
  mutate(stress_category = case_when(
    # When the value is > 5000, make the new column's value "Stressed!!!!"
    concentration_pg_g > 5000 ~ "Stressed!!!!!!",
    # Otherwise, make the new column's value "Chill"
    .default = "Chill"
  ))
  # Check out the first 6 rows, but remove the utm columns just for visibility
  head(nwt_pikas_categ %>% select(-c(utm_easting, utm_northing)))
```

	date	site	station	sex	concentration_pg_g	elev_m	stress_category
	<date>	<fct>	<fct>	<fct>	<dbl>	<dbl>	<chr>
1	2018-06-08	Cable	Gate	Cable G~	male	11563.	3343. Stressed!!!!!!
2	2018-06-08	Cable	Gate	Cable G~	male	10629.	3353. Stressed!!!!!!
3	2018-06-08	Cable	Gate	Cable G~	male	10924.	3358. Stressed!!!!!!
4	2018-06-13	West	Knoll	West Kn~	male	10414.	3578. Stressed!!!!!!
5	2018-06-13	West	Knoll	West Kn~	male	13531.	3584. Stressed!!!!!!
6	2018-06-13	West	Knoll	West Kn~	<NA>	7799.	3595. Stressed!!!!!!

## Q1.5

```
# Make a scatterplot with jittered points
nwt_pikas_categ %>%
  # We're adding a little placeholder axis just so we can see the point distribution
  ggplot(aes(x = "Placeholder",
              y = concentration_pg_g,
              color = stress_category)) +
  # Add the geom_jitter geom
  geom_jitter(width = 0.1)
```



```
nwt_pikas_categ2 <- nwt_pikas_categ %>%
  # Create a new column called month
  # then, extract the month from the date using the month() function
  mutate(month = month(date)) %>%
  # Lastly, relocate the month column after the date column so it's more easily visible to us
  relocate(month, .after = date)

head(nwt_pikas_categ2)
```

```
# A tibble: 6 x 10
date      month site       station      utm_easting utm_northing sex

```

```

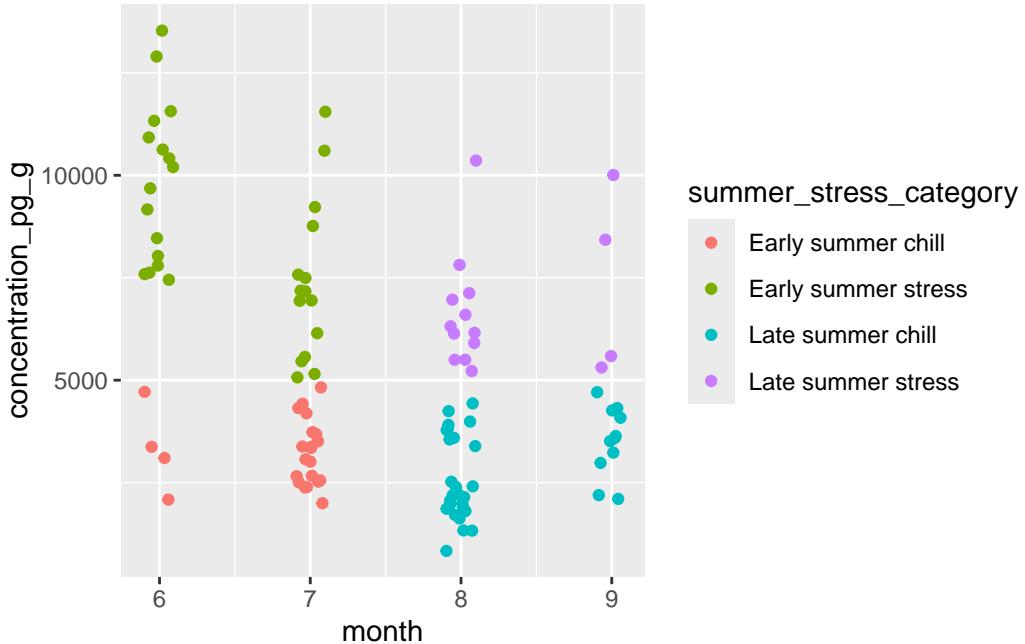
<date>      <dbl> <fct>      <fct>      <dbl>      <dbl> <fct>
1 2018-06-08     6 Cable Gate Cable Gate 1    451373    4432963 male
2 2018-06-08     6 Cable Gate Cable Gate 2    451411    4432985 male
3 2018-06-08     6 Cable Gate Cable Gate 3    451462    4432991 male
4 2018-06-13     6 West Knoll West Knoll 3   449317    4434093 male
5 2018-06-13     6 West Knoll West Knoll 4   449342    4434141 male
6 2018-06-13     6 West Knoll West Knoll 5   449323    4434273 <NA>
# i 3 more variables: concentration_pg_g <dbl>, elev_m <dbl>,
#   stress_category <chr>

nwt_pikas_summerstress <- nwt_pikas_categ2 %>%
  mutate(summer_stress_category = case_when(
    (month == 6 | month == 7) & concentration_pg_g > 5000 ~ "Early summer stress",
    (month == 6 | month == 7) & concentration_pg_g <= 5000 ~ "Early summer chill",
    (month == 8 | month == 9) & concentration_pg_g > 5000 ~ "Late summer stress",
    (month == 8 | month == 9) & concentration_pg_g <= 5000 ~ "Late summer chill",
    .default = "NA"
  ))
head(nwt_pikas_summerstress)

# A tibble: 6 x 11
  date      month site       station      utm_easting utm_northing sex
  <date>     <dbl> <fct>      <fct>          <dbl>          <dbl> <fct>
1 2018-06-08     6 Cable Gate Cable Gate 1    451373    4432963 male
2 2018-06-08     6 Cable Gate Cable Gate 2    451411    4432985 male
3 2018-06-08     6 Cable Gate Cable Gate 3    451462    4432991 male
4 2018-06-13     6 West Knoll West Knoll 3   449317    4434093 male
5 2018-06-13     6 West Knoll West Knoll 4   449342    4434141 male
6 2018-06-13     6 West Knoll West Knoll 5   449323    4434273 <NA>
# i 4 more variables: concentration_pg_g <dbl>, elev_m <dbl>,
#   stress_category <chr>, summer_stress_category <chr>

# Make a scatterplot with jittered points
nwt_pikas_summerstress %>%
  # We're adding a little placeholder axis just so we can see the point distribution
  ggplot(aes(x = month,
             y = concentration_pg_g,
             color = summer_stress_category)) +
  # Add the geom_jitter geom
  geom_jitter(width = 0.1)

```



## Q2.1

We're using the palmerpenguins data set.

```
# We are loading the library and then reading in penguins as penguins
library(palmerpenguins)
penguins <- penguins
```

## Q2.2

For our for loop we want it to read the species column from penguins and return the unique species names in a vector. Then, the for loop is going to iterate across the species names in the vector giving us a unique plot for each species.

## Q2.3

```
# Store a vector of unique species names from the Species column of species in the penguins
peng_names <- unique(penguins$species)

# We are making a for loop across the species names in the peng_names vector
for (i in 1:length(peng_names)) {
  #this line filters data for the interated species
```

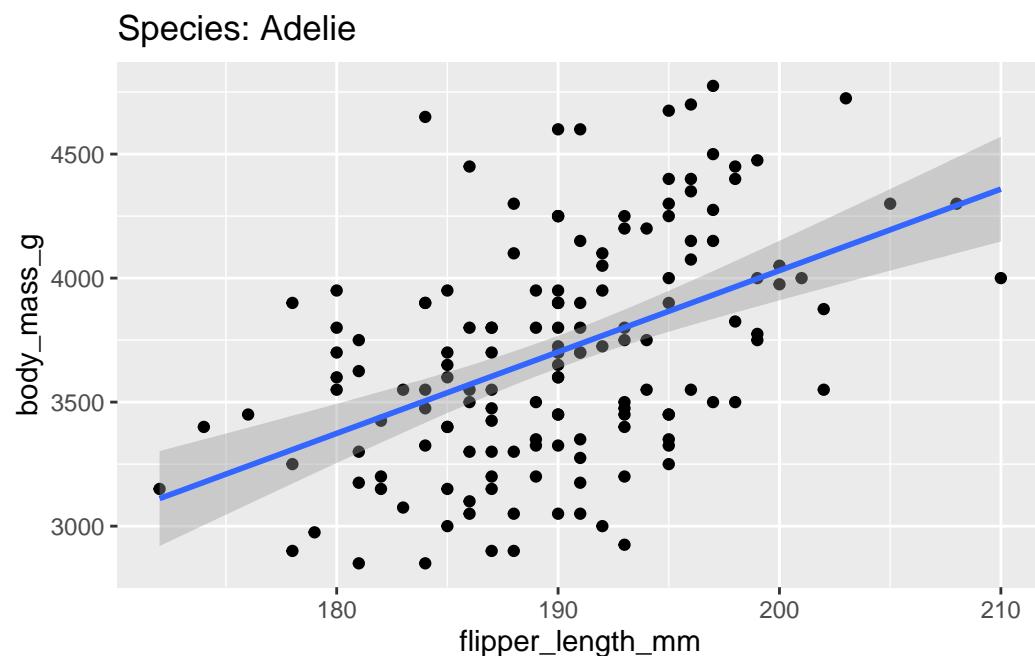
```

filt_data <- penguins %>%
  filter(species == peng_names[i])

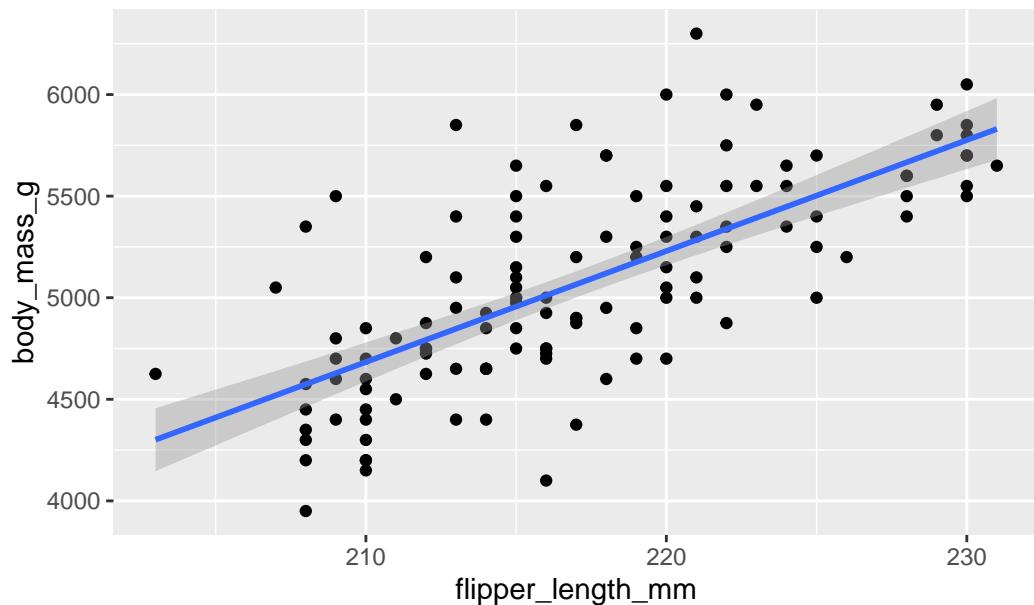
# These ggplot lines plot a scatter plot with a lm line.
plot <- filt_data %>%
  ggplot(aes(x = flipper_length_mm,
             y = body_mass_g)) +
  geom_point() +
  geom_smooth(method = "lm") +
  # This adds a title pasting in "Species and the interated species name
  ggttitle(paste("Species:", peng_names[i]))

# this displays the plot(s)
print(plot)
}

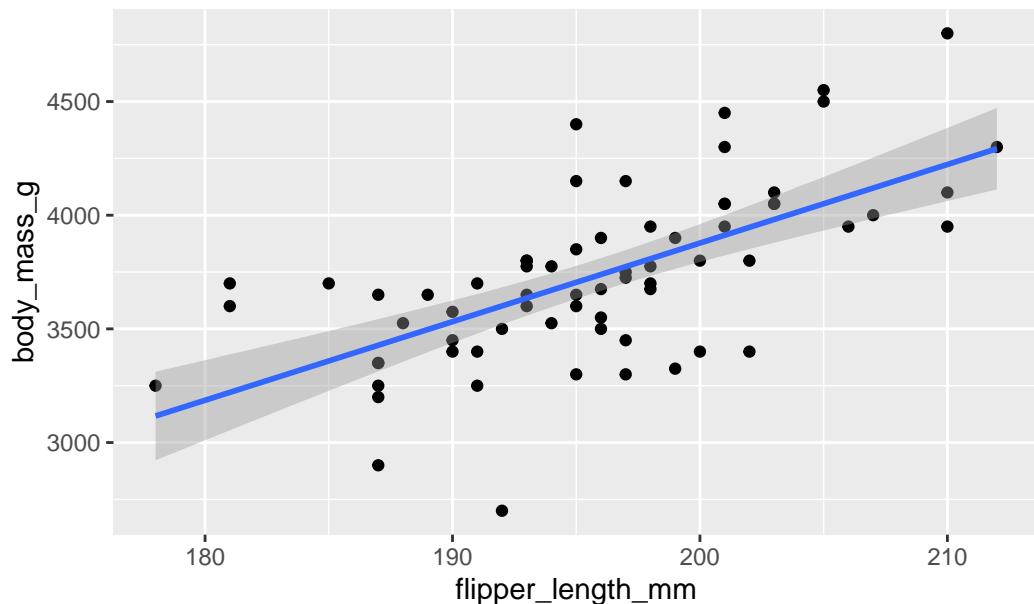
```



Species: Gentoo



Species: Chinstrap



#### Q 2.4

If flipper length is  $\geq 200$ , label as Big Ass Penguin and display points as different color in the

ggplots

## Q 2.5

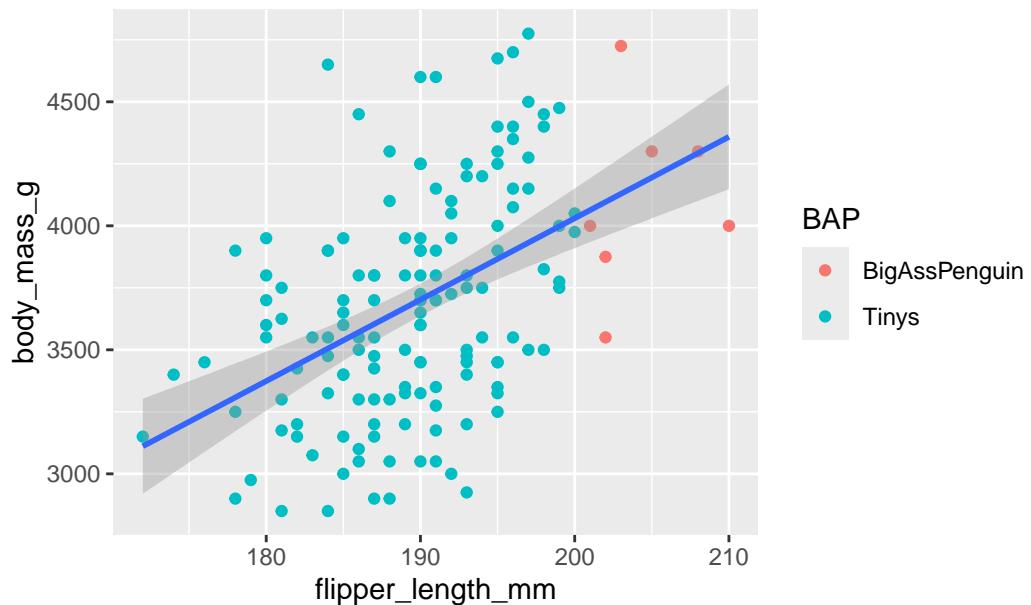
```
penguins <- penguins %>%
  # Call the new column BAP
  mutate(BAP = case_when(
    # When the value is > 200, make the new column's value "BigAssPenguin"
    flipper_length_mm > 200 ~ "BigAssPenguin",
    # Otherwise, make the new column's value "Tinys"
    .default = "Tinys"
  ))
}

# We are making a for loop across the species names in the peng_names vector
for (i in 1:length(peng_names)) {
  #this line filters data for the interated species
  filt_data <- penguins %>%
    filter(species == peng_names[i])

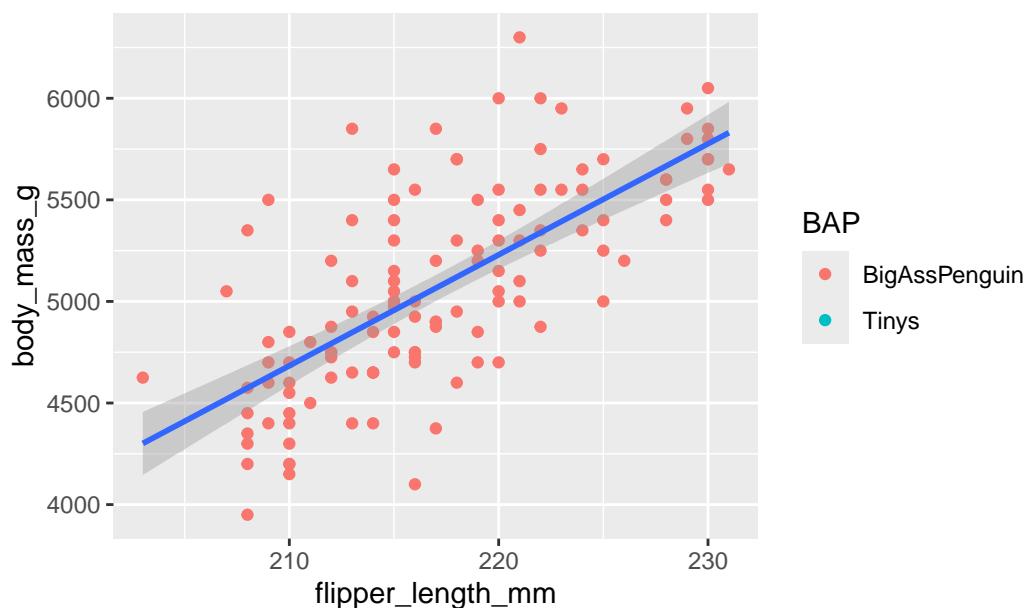
  # These ggplot lines plot a scatter plot with a lm line.
  plot <- filt_data %>%
    ggplot(aes(x = flipper_length_mm,
               y = body_mass_g)) +
    #coloring points by BAP category
    geom_point(aes(color=BAP)) +
    geom_smooth(method = "lm") +
    # This adds a title pasting in "Species and the interated species name
    ggttitle(paste("Species:", peng_names[i]))

  # this displays the plot(s)
  print(plot)
}
```

Species: Adelie



Species: Gentoo



### Species: Chinstrap

