Week 6: Relationships – Regression and Correlation

Ronnie Bailey-Steinitz

2025-11-03

Skills Learning – Lecture

This week, we will explore **relationships between variables** using **correlation** and **linear regression**. You'll learn to fit and interpret a simple linear model in R using the `lm()` function, visualize fitted trend lines with `geom_smooth(method = "lm")`, and compute correlation coefficients with `cor()`.

We'll use the **Palmer Penguins (raw)** dataset — a classic ecology dataset with measurements of penguin morphology, isotopes, and breeding information.

0. Load Packages

```
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   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_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 become errors
```

```
library(here)
```

```
## here() starts at /Users/rsteinitz/Documents/github/R Data Analysis Course
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
# install.packages("ggpmisc") # disable this after you've installed it once
library(ggpmisc)
```

```
## Loading required package: ggpp
## Registered S3 methods overwritten by 'ggpp':
##   method                  from
##   heightDetails.titleGrob ggplot2
##   widthDetails.titleGrob  ggplot2
##
## Attaching package: 'ggpp'
##
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

---

1. Import and Clean Data

```r
data <- read_csv(here("Week 1/Palmer Penguins Raw.csv")) %>%
  janitor::clean_names()
```

```
## Rows: 344 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): studyName, Species, Region, Island, Stage, Individual ID, Clutch C...
## dbl  (7): Sample Number, Bill Length (mm), Bill Depth (mm), Flipper Length (...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
dplyr::glimpse(data)
```

```
## Rows: 344
## Columns: 17
## $ study_name        <chr> "PAL0708", "PAL0708", "PAL0708", "PAL0708", "PAL0708~
## $ sample_number     <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ species           <chr> "Adelie Penguin (Pygoscelis adeliae)", "Adelie Pengu~
## $ region            <chr> "Anvers", "Anvers", "Anvers", "Anvers", "Anvers", "A~
## $ island            <chr> "Torgersen", "Torgersen", "Torgersen", "Torgersen", ~
## $ stage             <chr> "Adult, 1 Egg Stage", "Adult, 1 Egg Stage", "Adult, ~
## $ individual_id     <chr> "N1A1", "N1A2", "N2A1", "N2A2", "N3A1", "N3A2", "N4A~
## $ clutch_completion <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "No", "No"~
## $ date_egg          <chr> "11/11/07", "11/11/07", "11/16/07", "11/16/07", "11/~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
## $ flipper_length_mm <dbl> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g       <dbl> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex               <chr> "MALE", "FEMALE", "FEMALE", NA, "FEMALE", "MALE", "F~
```

```
## $ delta_15_n_o_oo   <dbl> NA, 8.94956, 8.36821, NA, 8.76651, 8.66496, 9.18718,~
## $ delta_13_c_o_oo   <dbl> NA, -24.69454, -25.33302, NA, -25.32426, -25.29805, ~
## $ comments          <chr> "Not enough blood for isotopes.", NA, NA, "Adult not~
```

```
options(scipen = 999)
```

For today's examples, we'll focus on **bill length** and **flipper length**, two continuous variables that often show strong positive relationships in penguins.

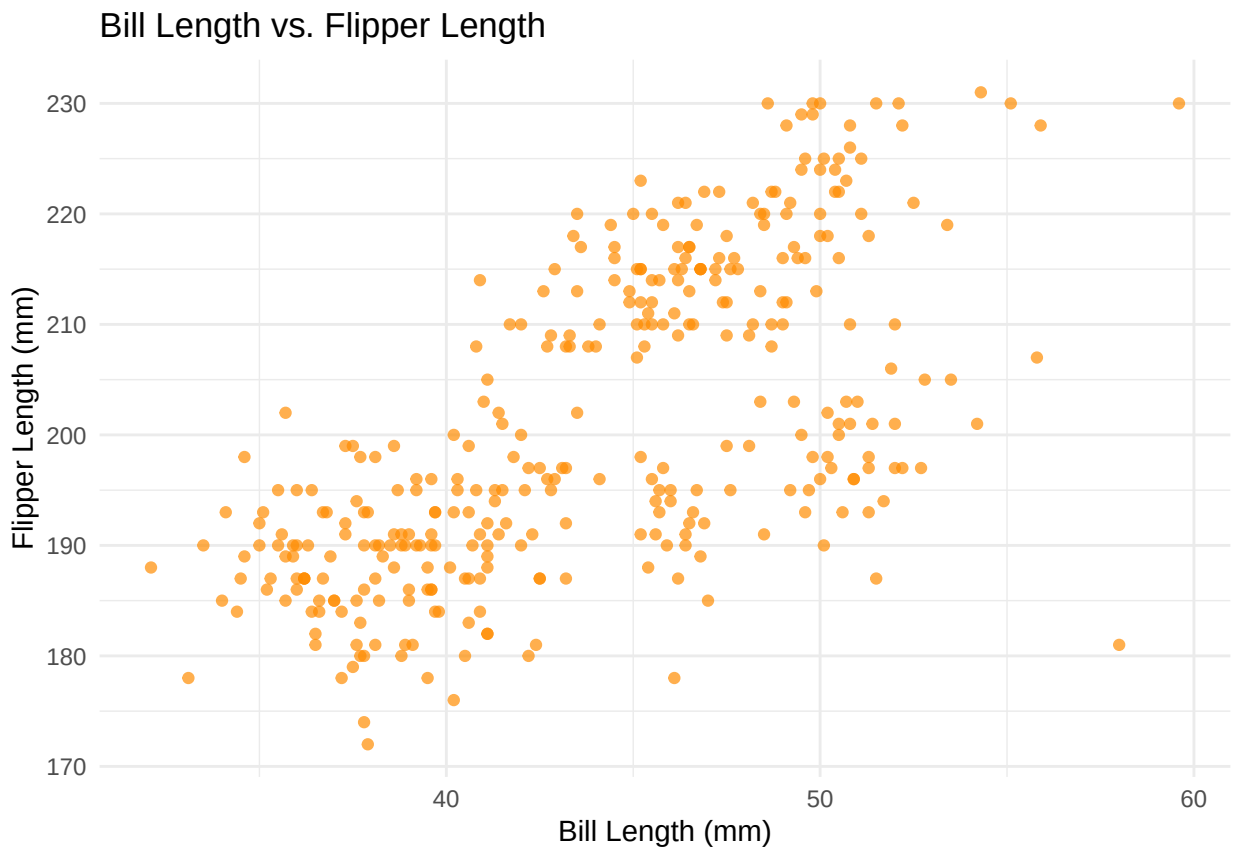
2. Visualizing Relationships

A scatterplot helps us see whether two numeric variables tend to increase or decrease together.

We can also fit and visualize a **linear model** using `geom_smooth(method = "lm")`.
This draws the best-fit straight line through the data.

```
ggplot(data, aes(x = bill_length_mm, y = flipper_length_mm)) +
  geom_point(color = "darkorange", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Bill Length vs. Flipper Length",
       x = "Bill Length (mm)",
       y = "Flipper Length (mm)")
```

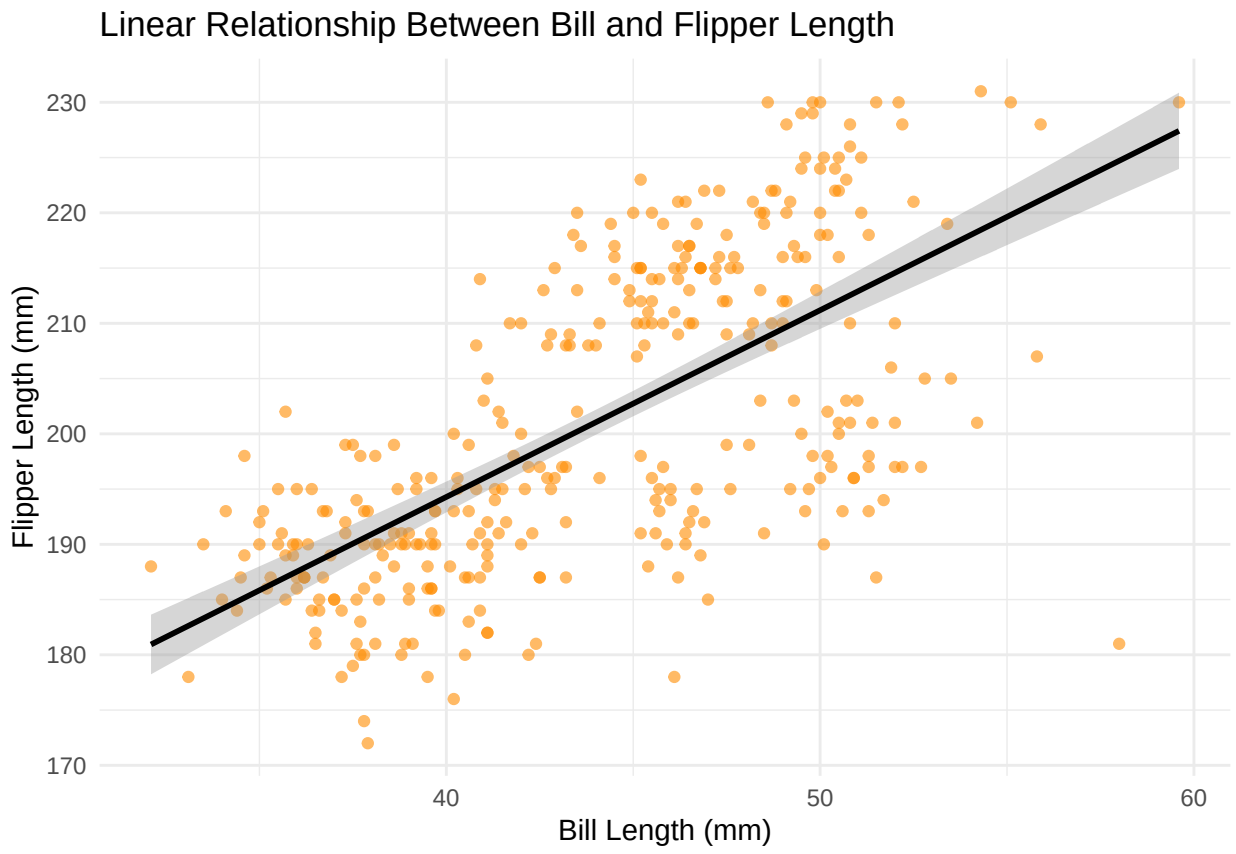
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```r
ggplot(data, aes(x = bill_length_mm, y = flipper_length_mm)) +
  geom_point(alpha = 0.6, color = "darkorange") +
  geom_smooth(method = "lm", se = TRUE, color = "black") +
  theme_minimal() +
  labs(title = "Linear Relationship Between Bill and Flipper Length",
       x = "Bill Length (mm)", y = "Flipper Length (mm)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range (`stat_smooth()`).
## Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Note: Each point is an individual penguin. If the points rise together, that suggests a positive correlation.

3. Fitting a Linear Model with `lm()`

Now let's fit the model explicitly using R's built-in **linear model** function: `lm()`.

```r
model <- lm(flipper_length_mm ~ bill_length_mm, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = flipper_length_mm ~ bill_length_mm, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.708  -7.896   0.664   8.650  21.179
##
## Coefficients:
##                Estimate Std. Error t value            Pr(>|t|)
## (Intercept)    126.6844     4.6651   27.16 <0.0000000000000002 ***
## bill_length_mm   1.6901     0.1054   16.03 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.63 on 340 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.4306, Adjusted R-squared:  0.4289
## F-statistic: 257.1 on 1 and 340 DF,  p-value: < 0.00000000000000022
```

Interpreting Model Output

Call
- `lm(formula = flipper_length_mm ~ bill_length_mm, data = data)` - Shows what model was fitted.
- Here, `flipper_length_mm` is the *response* (dependent variable) and `bill_length_mm` is the *predictor* (independent variable).

Coefficients Table
Each row is part of the model equation:

$$\text{Flipper Length} = \text{Intercept} + (\text{Slope} \times \text{Bill Length})$$

- **`Estimate`**
  - `(Intercept)` = `126.6844`: predicted flipper length when bill = 0 (not biologically meaningful).
  - `bill_length_mm` = `1.6901`: for each 1 mm increase in bill length, flipper length increases by ~1.69 mm.
- **`Std. Error`**
  - Uncertainty of the estimate; smaller = more precise.
- **`t value`**
  - Tests if each coefficient differs from 0: `t = Estimate / Std.Error`.
  - Large |t| = stronger effect.
- **`Pr(>|t|)` (p-value)**
  - Probability that effect occurred by chance.
  - Very small values (< 0.001) = strong evidence of a real effect.
- **Significance codes (`***`, `**`, `*`, `.`)**
  - Quick visual indicators:

- *** p < 0.001
- ** p < 0.01
- * p < 0.05

Residuals
- Differences between observed and predicted `flipper_length_mm`. - Summarize how far predictions deviate from actual values. - The five-number summary (`Min`, `1Q`, `Median`, `3Q`, `Max`) shows spread and balance of errors. - `Min` / `Max` – largest under/over-predictions. - `1Q` / `3Q` – middle 50% of residuals. - `Median` – should be near 0 if model is unbiased.

- `10.63` → average deviation (mm) between observed and fitted values.
- Smaller RSE = tighter fit of data to the line.

Degrees of Freedom
- `340` → number of observations minus number of estimated parameters (`n - 2`). - Reflects information used to estimate residual variation.

Multiple R-squared (`R²`)
- `0.4306` → about 43% of variation in flipper length explained by bill length. - Closer to 1 = stronger linear relationship.

Adjusted R-squared
- `0.4289` → adjusts for sample size and number of predictors. - Nearly same here since only one predictor.

F-statistic
- `257.1 on 1 and 340 DF, p-value < 2.2e-16` - Tests if at least one predictor explains a significant portion of variance. - Large F + very small p-value = model fits significantly better than a null model.

What this means:
- Penguins with longer bills tend to have longer flippers. - Relationship is **positive**, **strong**, and **statistically significant**, explaining ~43% of variation in flipper length.

You can access specific components of the model like this:

```
model$coefficients     # slope and intercept
```

```
##    (Intercept) bill_length_mm
##     126.684427       1.690062
```

```
summary(model)$r.squared   # R² value
```

```
## [1] 0.430574
```

Interpretation: `flipper_length_mm = 126.6844 + 1.6901 * bill_length_mm`

- `Intercept (126.6844)`: This is the model's predicted flipper length (mm) when `bill_length_mm = 0`. It defines where the regression line crosses the y-axis.

- `Slope (1.6901)`: For every `1 mm` increase in bill length, the model predicts an average increase of `1.69 mm` in flipper length. The positive sign indicates a positive relationship: penguins with longer bills tend to have longer flippers.

4. Correlation Between Two Variables

The **correlation coefficient (r)** measures the strength and direction of a linear relationship.
It ranges from **-1 (perfect negative)** to **+1 (perfect positive)**.

```
# Correlation coefficient (r)
data %>%
  summarize(cor(bill_length_mm, flipper_length_mm, use = "complete.obs"))
```

```
## # A tibble: 1 x 1
##   `cor(bill_length_mm, flipper_length_mm, use = "complete.obs")`
##                                                            <dbl>
## 1                                                          0.656
```

> The correlation coefficient squared (r^2) is directly related to the R^2 from the regression model. r tells you how the two variables move together, while R^2 explains how much of one variable's changes can be explained by the other.

5. Visualizing Relationships by Group

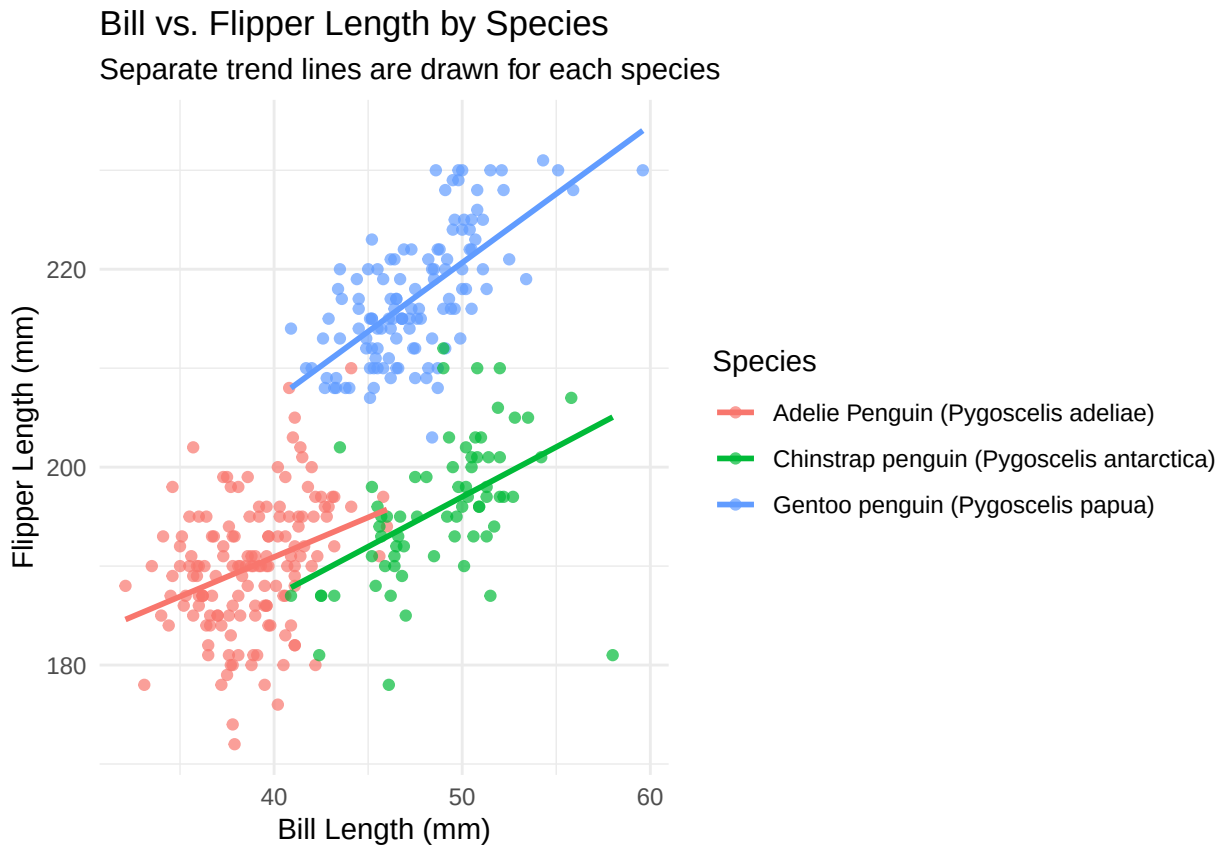
We can color points and lines by a grouping variable to show how relationships differ among groups (e.g., species).

```
ggplot(data, aes(x = bill_length_mm, y = flipper_length_mm, color = species)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_minimal() +
  labs(title = "Bill vs. Flipper Length by Species",
       subtitle = "Separate trend lines are drawn for each species",
       x = "Bill Length (mm)", y = "Flipper Length (mm)",
       color = "Species")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

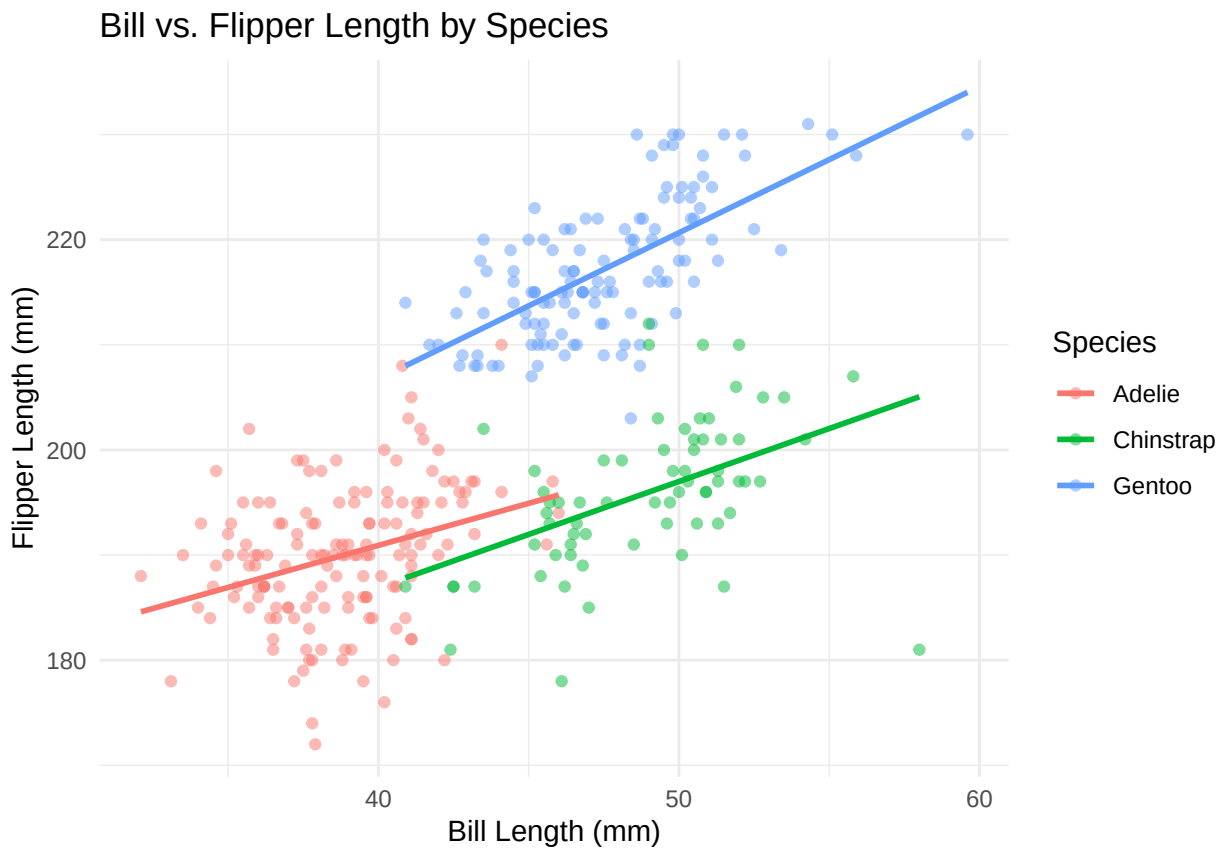
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# Notice how the species names are so long?
# remember that we can extract a single word from a string of words:
data <- data %>%
  mutate(spp = word(species, 1))

# then use that as your grouping variable!
ggplot(data, aes(x = bill_length_mm, y = flipper_length_mm, color = spp)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_minimal() +
  labs(title = "Bill vs. Flipper Length by Species",
       x = "Bill Length (mm)", y = "Flipper Length (mm)",
       color = "Species")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range (`stat_smooth()`).
## Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



6. Customizing Your Visualization

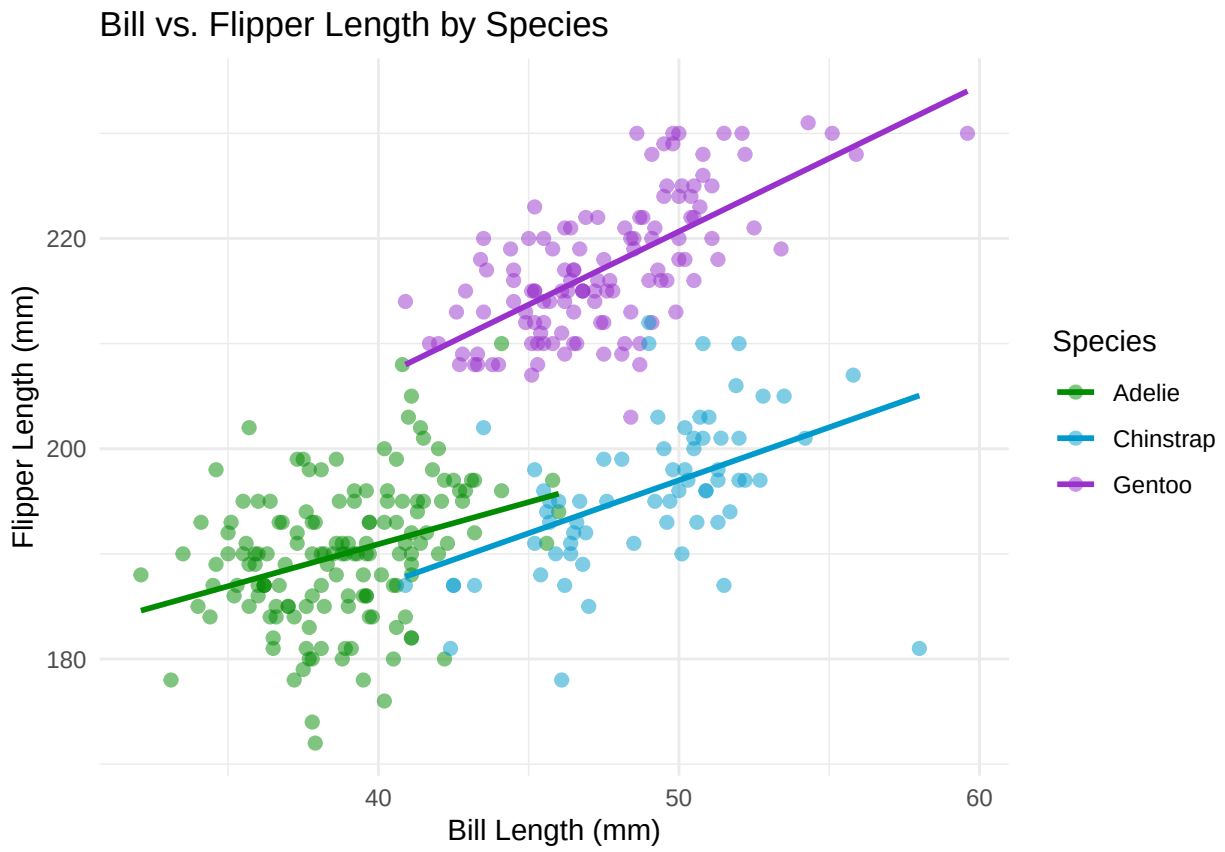
Let's make the plot cleaner and more publication-ready.

```
# define your own color palette
ggplot(data, aes(x = bill_length_mm, y = flipper_length_mm, color = spp)) +
  geom_point(alpha = 0.5, size = 2) +
  geom_smooth(method = "lm", se = FALSE, linewidth = 1) +
  scale_color_manual(
    values = c(
      "Adelie" = "green4",       # green
      "Chinstrap" = "deepskyblue3",    # orange
      "Gentoo" = "darkorchid"        # purple
    )
  ) +
  theme_minimal() +
  labs(
    title = "Bill vs. Flipper Length by Species",
    x = "Bill Length (mm)",
    y = "Flipper Length (mm)",
    color = "Species"
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

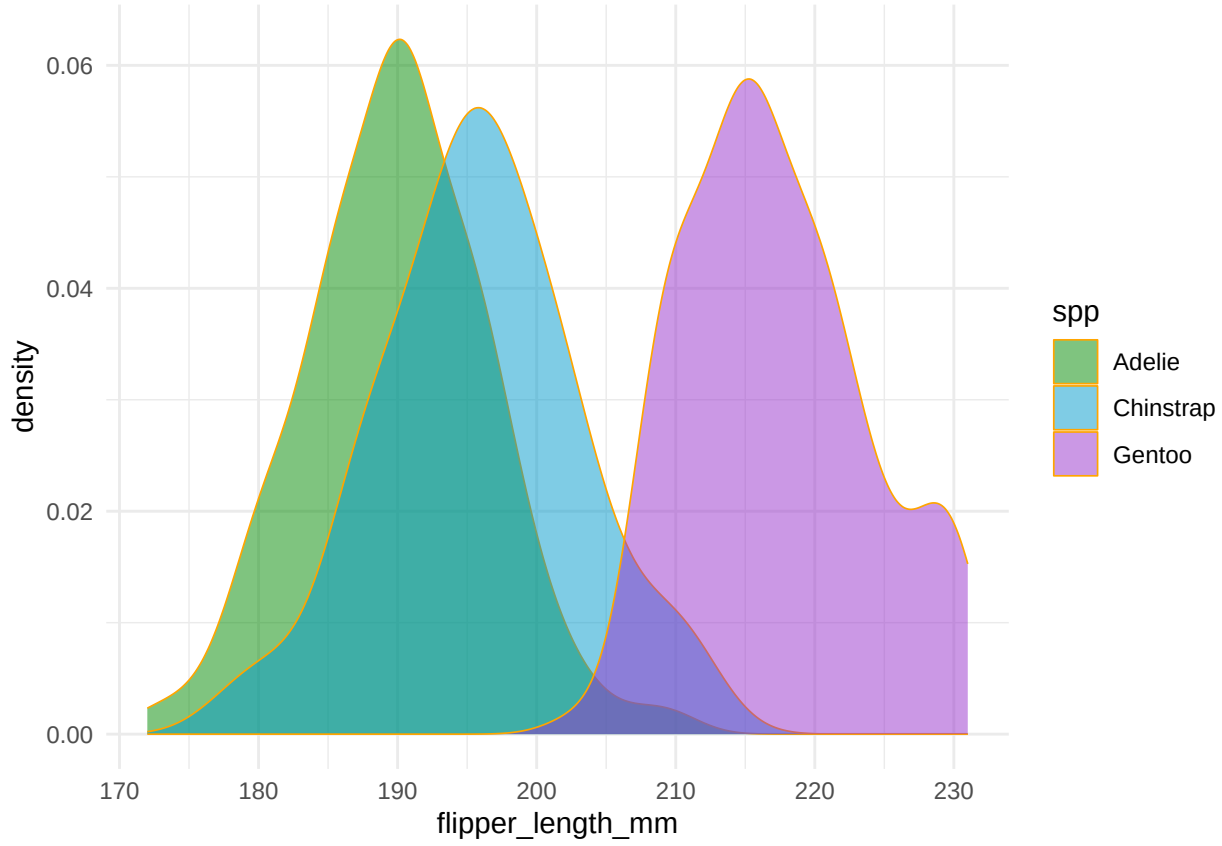
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
ggplot(data, aes(x = flipper_length_mm, fill = spp)) +
  geom_density(alpha = 0.5, color = "orange", linewidth = 0.3) +
  scale_fill_manual(
    values = c(
      "Adelie" = "green4",       # green
      "Chinstrap" = "deepskyblue3",    # orange
      "Gentoo" = "darkorchid"        # purple
    )) +
  theme_minimal()
```

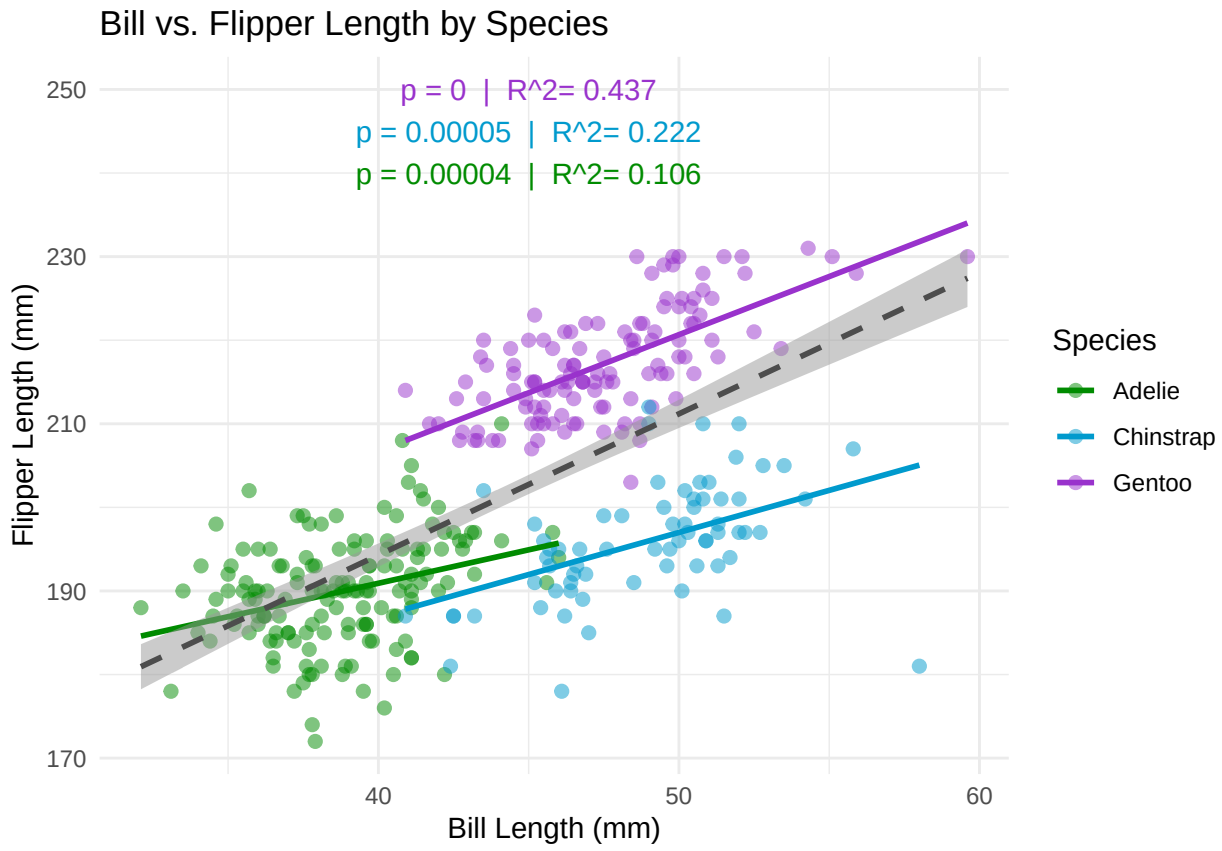
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_density()`).
```



. . . and much more advanced:

```
ggplot(data, aes(x = bill_length_mm, y = flipper_length_mm, color = spp)) +
  geom_point(alpha = 0.5, size = 2) +
  geom_smooth(method = "lm", se = FALSE, linewidth = 1) +
  geom_smooth(data = data, method = "lm", aes(x = bill_length_mm, y = flipper_length_mm), color = "grey
  scale_color_manual(
    values = c(
      "Adelie" = "green4",       # green
      "Chinstrap" = "deepskyblue3",    # orange
      "Gentoo" = "darkorchid"        # purple
    )
  ) +
  stat_fit_glance(method = "lm",
                  geom = "text",
                  aes(label = paste("p = ", round(signif(..p.value.., digits = 8), 5),
                                    "  |  R^2= ", signif(..r.squared.., digits = 3),
                                    sep = "")),
                  label.x = 45, label.y = c(240, 245, 250)) +
  theme_minimal() +
  labs(
    title = "Bill vs. Flipper Length by Species",
    x = "Bill Length (mm)",
    y = "Flipper Length (mm)",
    color = "Species"
  )
```

```
## Warning: The dot-dot notation (‘..p.value..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(p.value)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## ‘geom_smooth()‘ using formula = ’y ~ x’

## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).

## ‘geom_smooth()‘ using formula = ’y ~ x’

## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).

## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_fit_glance()‘).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



You can find **additional color palettes here**.* <- click here*

Skills Application – Practice Prompts

Try the following short exercises using the **Palmer Penguins Raw** dataset:

1. **Fit your own model:**
   - Choose two new numeric variables.
   - Fit a linear model using `lm()` and inspect the output with `summary()`.
   - What do the slope and R^2 tell you about the relationship?
2. **Check correlation:**
   - Use `cor()` to calculate the correlation between the same two variables.
   - How does it compare to the R^2 from your model?
3. **Visualize grouped relationships:**
   - Make a scatterplot using `geom_point()` + `geom_smooth(method = "lm")`.
   - Add a grouping variable inside your `aes()` call (e.g., `aes(color = species)`).
   - How do the slopes differ among the groups?
4. **Interpret your model:**
   - In one or two sentences, describe what your regression model suggests.
   - Are the relationships positive or negative? Strong or weak?
5. **Challenge:**
   - Fit a model predicting `body_mass_g` from two predictors (e.g., `flipper_length_mm` and `bill_length_mm`).
   - Compare the R^2 from this two-variable model to the one-variable version. What improved?

By the end of Week 6, you should be able to:

- Fit and interpret a simple linear regression with `lm()`.
- Understand key parts of a regression summary (slope, intercept, R^2, p-value).
- Calculate correlations with `cor()`.
- Visualize relationships and add regression lines with `geom_smooth(method = "lm")`.
- Customize scatterplots with grouping colors and improved labeling.