Using Higher-Order Terms: Interaction

August 3, 2024

Table of contents

Where We Stand

In the last notebook, we saw how to model curvi-linear relationships. We added curvature to our models by feature engineering with `step_poly()` as a component of a `recipe()`. In that notebook, we saw that introducing curvature can improve model performance and can help us model more complex relationships, but results in models which are more difficult to interpret.

In this notebook, we'll consider another way that we can potentially improve model performance. That is, by allowing predictors to interact with one another. Interactions allow for the relationship between two variables (a predictor and a response) to depend on the value of a third variable (another predictor). The impact of an interaction term on a model depends on the types of variables which are interacting – we'll explore the possibilities in this notebook. Similar to models including polynomial terms, we'll sometimes observed improved *fit* when using models with interaction terms, but those resulting models become more difficult to interpret.

Objectives

- Use exploratory data analysis to identify visual evidence for interaction between predictors.
- Use `step_interact()` to create additional model terms corresponding to the interaction (product) between two or more predictors.

- Fit, assess, reduce, interpret, and utilize models including interactions terms.

Models Including Interaction (`step_interact()`)

We can use interaction terms when we expect that the association between our response and one predictor depends on the value of a second predictor. There are three types of interactions between two variables:

- Interactions between *two categorical predictors* result in a shift of intercept, associated with combinations of categories. This is very similar to what happened when we first introduced categorical predictors – there we got a different intercept for each level of the categorical predictor. Now, we'll get an adjustment to the intercept for each combination of levels of the corresponding predictors.
  - For example, if we allow `species` and `year` to interact in a model to predict penguin `body_mass_g`, we'll have potentially unique intercepts for each `species` and observed `year` combination.
- Interactions between *a categorical predictor and a numerical predictor* allow for different slopes/curvatures in the association between the response and corresponding predictor across different levels of the categorical variable.
  - For example, if we allow `species` and `bill_depth_mm` to interact in a model to predict penguin `body_mass_g`, we'll allow for the association between `bill_depth_mm` and `body_mass_g` to be different across the three `species` of penguin.
- Interactions between two numerical variables allow for the association between the response (y) and a predictor (x_1) to depend on the value of a second predictor (x_2).
  - There is not necessarily a nice *slope* or *intercept* interpretation here, however, interactions between pairs of numerical predictors can introduce curvature to your regression model. Simply put, if a model includes an interaction term $\beta x_1 x_2$, in order to know the expected impact of a unit increase of x_1 on the response variable (y), we must decide which level of the variable x_2 we are interested in first.
  - As an example, if we allow `bill_depth_mm` to interact with `bill_length_mm` in a model predicting penguin `body_mass_g`, then we are saying that the expected impact of a unit increase in `bill_length_mm` on `body_mass_g` will be different for penguins having different `bill_depth_mm` values.

Interactions in Action!

Let's see these different types of interactions in action with the penguins data. We'll look at the three scenarios above one-by-one and then we'll try pulling everything together.

Interactions Between Two Categorical Variables

Let's say that we propose a model which predicts penguin `body_mass_g` using `flipper_length_mm`, `species`, `year`, and an interaction between `species` and `year`. Such a model is of the following form:

$$\begin{aligned}\mathbb{E}\left[\text{body\_mass\_g}\right] =& \beta_0 + \beta_1\cdot\text{flipper\_length\_mm} +\\ & \beta_2\cdot\text{Chinstrap} + \beta_3\cdot\text{Gentoo} +\\ & \beta_4\cdot\text{year\_08} + \beta_5\cdot\text{year\_09} +\\ & \beta_6\cdot\left(\text{Chinstrap}\right)\left(\text{year\_08}\right) +\\ & \beta_7\cdot\left(\text{Chinstrap}\right)\left(\text{year\_09}\right) +\\ & \beta_8\cdot\left(\text{Gentoo}\right)\left(\text{year\_08}\right) +\\ & \beta_9\cdot\left(\text{Gentoo}\right)\left(\text{year\_09}\right)\end{aligned}$$

There's lots of coefficients in this model! Let's fit it and see what we actually end up with. In order to fit this model, we'll need to include some feature engineering `step_*()`s. First, we'll need to obtain *dummy* variables for the levels of the `species` and `island` predictors – we'll use `step_dummy()` for that. Then, we'll want to allow those dummy variables to interact – we'll use `step_interact()`, with a few tricks.

- `step_interact()` requires the use of a tilde (`~`) prior to defining the terms that should interact with one another.
- We'll denote terms that should interact with one another via the colon (`:`).
- We replaced `species` and `island` with corresponding dummy variables when we used `step_dummy()`, so those columns will not be available in the transformed data being passed to `step_interact()`. Instead, we'll have a series of columns like `species_X`, `species_Y`, ... , and `island_A`, `island_B`,... – rather than defining all of these individual interactions, we'll use the `starts_with()` selector function to obtain the groups of columns we want interacting with one another.

```
mass_cat_inter_spec <- linear_reg() %>%
  set_engine("lm")

mass_cat_inter_rec <- recipe(body_mass_g ~ flipper_length_mm + species + year, data = pengui
  step_dummy(species) %>%
  step_mutate(year = as.factor(year)) %>%
  step_dummy(year) %>%
  step_interact(~ starts_with("species"):starts_with("year"))

mass_cat_wf <- workflow() %>%
  add_model(mass_cat_inter_spec) %>%
  add_recipe(mass_cat_inter_rec)

mass_cat_fit <- mass_cat_wf %>%
  fit(penguins_train)

mass_cat_fit %>%
  glance() %>%
  kable() %>%
  kable_styling()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
0.80644	0.7993586	367.5427	113.8804	0	9	-1870.299	3762.598	3801.595	33231550

```
mass_cat_fit %>%
  extract_fit_engine() %>%
  tidy() %>%
  kable() %>%
  kable_styling()
```

term	estimate	std.error	statistic	p.value
Intercept	-4933.59608	691.592270	-7.1336773	0.0000000
flipper_length_mm	46.42457	3.704239	12.5328209	0.0000000
species_Chinstrap	-315.26370	104.431973	-3.0188427	0.0028044
species_Gentoo	35.38641	144.099729	0.2455689	0.8062207
year_X2008	-186.53149	84.902008	-2.1970209	0.0289520
year_X2009	-330.22761	88.313278	-3.7392747	0.0002296
species_Chinstrap_x_year_X2008	54.03892	158.852806	0.3401823	0.7340095
species_Chinstrap_x_year_X2009	104.68961	146.791942	0.7131836	0.4764081
species_Gentoo_x_year_X2008	-41.65877	126.758321	-0.3286472	0.7427019
species_Gentoo_x_year_X2009	267.92841	129.193714	2.0738502	0.0391339

There are some interesting things happening here. Several of our model terms have large `p.value`s, indicating that they may not be significant predictors of penguin `body_mass_g`. However, if we look more closely, *some* of the terms are significant.

- The `species_Gentoo` term is not significantly different from the base level (`species_Adelie`), but the `species_Chinstrap` term is significantly different from the base level.
- Both year categories are significantly different from the base year (`year_2007`).
- The interaction between `species` and `year` is significant for Gentoo penguins in the year 2009, but not for any of the other combinations of `species` and `year`.

If all levels of a categorical predictor (or interaction) show insignificant `p.value`s, then we would drop that predictor (or the corresponding interaction) from the model. Since *some* of the terms resulting from the categorical predictor (or interaction) are statistically significant, then we'll keep all of the corresponding model terms.

- ($\star$) There are some other things we *could* do here – we can address some of them, as well as their benefits and drawbacks, in class.

Okay, let's look at the model. The model including estimated β-coefficients is:

$$\begin{aligned}\mathbb{E}\left[\text{body\_mass\_g}\right] =& -4933.6 + 46.42\cdot\text{flipper\_length\_mm}+\\ &(-315.26)\cdot\text{Chinstrap} + 35.39\cdot\text{Gentoo}+\\ &(-186.53)\cdot\text{year\_08} + (-330.23)\cdot\text{year\_09}+\\ &54.04\cdot\left(\text{Chinstrap}\right)\left(\text{year\_08}\right)+\\ &104.69\cdot\left(\text{Chinstrap}\right)\left(\text{year\_09}\right)+\\ &(-41.66)\cdot\left(\text{Gentoo}\right)\left(\text{year\_08}\right) +\\ 267.93\cdot\left(\text{Gentoo}\right)\left(\text{year\_09}\right)\end{aligned}$$

From this, we can make interpretations as follows:

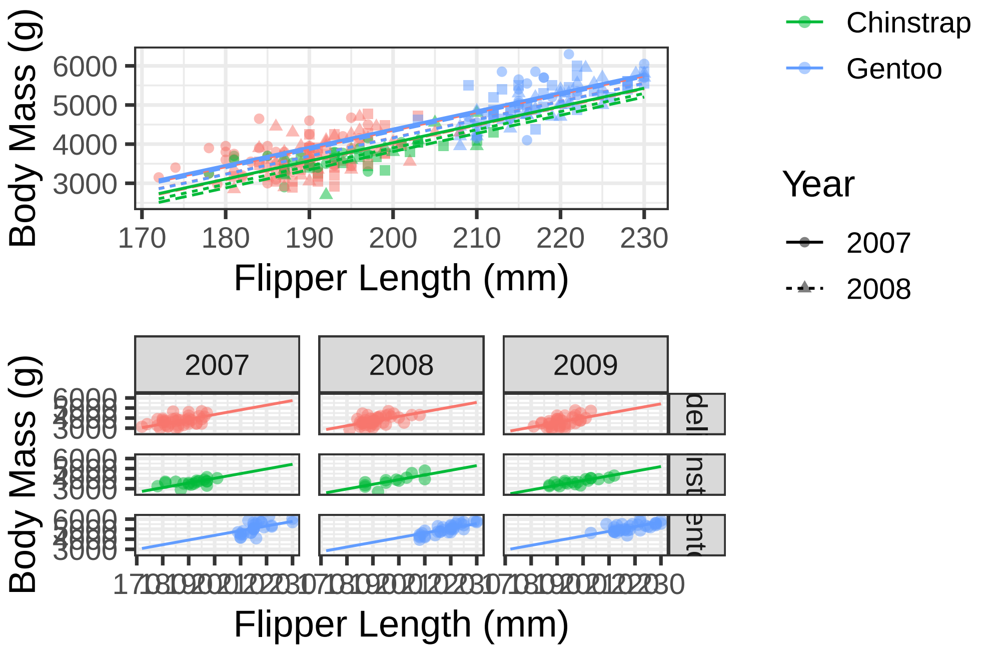
- On average, controlling for `species` and observation `year`, a unit increase in penguin `flipper_length_mm` is associated with a penguin `body_mass_g` increase of about 46.42 grams.
- Controlling for `flipper_length_mm`, Chinstrap penguins are about 315.26 grams less massive than Adelies, on average. The data used to fit the model doesn't seem to suggest significant changes in Chinstrap mass year over year.
- Controlling for `flipper_length_mm`, the data used to fit the model doesn't seem to suggest that Gentoo penguins have significantly different `body_mass_g` than Adelie penguins. However, there does seem to be a change in the year 2009 – Gentoos observed during that year seem to be much more massive than Adelies in that year (by about $267.92 - (-330.23) \approx 590$ grams).

- Controlling for `flipper_length_mm`, Adelie penguins seem to have less mass year-over-year. Approximately 186.5 grams lower than their average 2007 `body_mass_g` in 2008 and approximately 330.23 grams lower than their average 2007 `body_mass_g` in 2009.

Now, let's look at our model graphically!

```r
new_data <- crossing(flipper_length_mm = seq(min(penguins_train$flipper_length_mm,
                                                 na.rm = TRUE),
                                             max(penguins_train$flipper_length_mm,
                                                 na.rm = TRUE),
                                             by = 1),
                     species = c("Adelie", "Chinstrap", "Gentoo"),
                     year = c(2007, 2008, 2009)
)

new_data <- mass_cat_fit %>%
  augment(new_data)

p1 <- ggplot() +
  geom_point(data = penguins_train,
             aes(x = flipper_length_mm,
                 y = body_mass_g,
                 color = species,
                 shape = as.factor(year)),
             alpha = 0.5) +
  geom_line(data = new_data,
            aes(x = flipper_length_mm,
                y = .pred,
                color = species,
                linetype = as.factor(year))) +
  labs(x = "Flipper Length (mm)",
       y = "Body Mass (g)",
       linetype = "Year",
       shape = "Year")

p2 <- ggplot() +
  geom_point(data = penguins_train,
             aes(x = flipper_length_mm,
                 y = body_mass_g,
                 color = species),
             alpha = 0.5,
             show.legend = FALSE) +
  geom_line(data = new_data,
            aes(x = flipper_length_mm,
                y = .pred,
                color = species),
            show.legend = FALSE) +
  facet_grid(species ~ year) +
  labs(x = "Flipper Length (mm)",
       y = "Body Mass (g)")

(p1 / p2)
```

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



The single plot above which shows the model with all of the varied intercepts may be difficult to read, but it is clear that the slope of each line is the same. That is, the estimated association between `flipper_length_mm` and `body_mass_g` is constant across each of the combinations of `species`/`year` category. The intercepts are all that were varied! The plot on the right is much more readable, but the impact of the categorical variables and the corresponding interactions is less obvious from that plot.

Interaction Between a Categorical and Numerical Predictor

Let's say that we propose a model which predicts penguin `body_mass_g` using `flipper_length_mm`, `species`, and an interaction between `species` and `flipper_length_mm`. Such a model is of the following form:

$$\begin{aligned}\mathbb{E}\left[\text{body\_mass\_g}\right] =& \beta_0 + \beta_1\cdot \text{flipper\_length\_mm} + \beta_2\cdot \text{Chinstrap} + \beta_3\cdot \text{Gentoo} +\\ & \beta_4\cdot\left(\text{flipper\_length\_mm}\right)\left(\text{Chinstrap}\right) + \beta_5\cdot\left(\text{flipper\_length\_mm}\right)\left(\text{Gentoo}\right)\end{aligned}$$

We'll build this model similar to the previous one. We'll use `step_dummy()` to convert `species` to corresponding dummy variables and then we'll use `step_interact()` to obtain interactions between each level of the `species` variable and the `flipper_length_mm`.

```
mass_catnum_spec <- linear_reg() %>%
  set_engine("lm")

mass_catnum_rec <- recipe(body_mass_g ~ flipper_length_mm + species, data = penguins_train) 
  step_dummy(species) %>%
  step_interact(~ starts_with("species"):flipper_length_mm)

mass_catnum_wf <- workflow() %>%
  add_model(mass_catnum_spec) %>%
  add_recipe(mass_catnum_rec)

mass_catnum_fit <- mass_catnum_wf %>%
  fit(penguins_train)

mass_catnum_fit %>%
  glance() %>%
  kable() %>%
  kable_styling()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
0.7952182	0.7911226	375.0103	194.1623	0	5	-1877.513	3769.025	3793.842	35158182

```
mass_catnum_fit %>%
  extract_fit_engine() %>%
  tidy() %>%
  kable() %>%
  kable_styling()
```

term	estimate	std.error	statistic	p.value
Intercept	-2797.759952	1059.375697	-2.6409516	0.0087881
flipper_length_mm	34.282823	5.582906	6.1406767	0.0000000
species_Chinstrap	-129.708846	1708.747729	-0.0759087	0.9395524
species_Gentoo	-4216.630301	1702.555778	-2.4766474	0.0139247
species_Chinstrap_x_flipper_length_mm	-0.366684	8.853089	-0.0414188	0.9669951
species_Gentoo_x_flipper_length_mm	21.389349	8.284884	2.5817319	0.0104008

We've obtained our estimated model,

$$\begin{aligned}\mathbb{E}\left[\text{body\_mass\_g}\right] \approx & -2797.76 + 34.38\cdot\text{flipper\_length\_mm} + (-129.71)\cdot\text{Chinstrap} + (-4216.6)\cdot\text{Gentoo}+\\ & (-0.37)\cdot(\text{flipper\_length\_mm})\,(\text{Chinstrap}) + 21.39\cdot(\text{flipper\_length\_mm})\,(\text{Gentoo})\end{aligned}$$

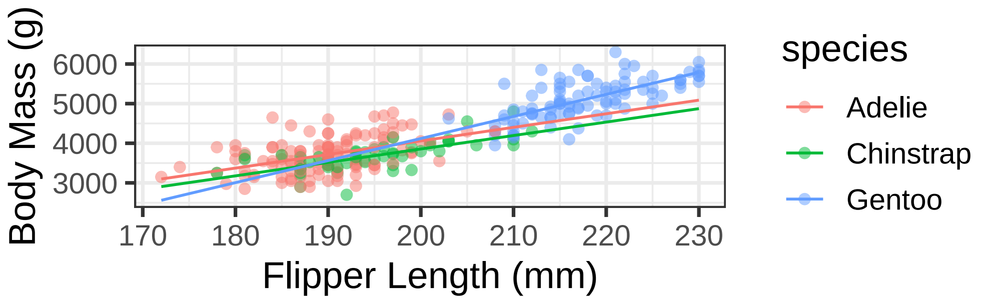
Notice that this model allows for different slopes and intercepts for each species. For example:

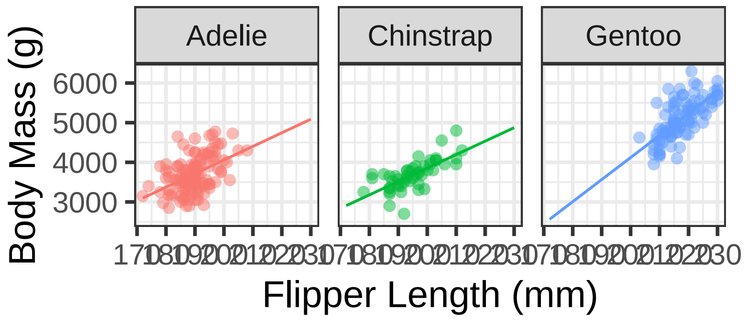
$$\begin{array}{lcl}\text{Adelie} & : & \mathbb{E}\left[\text{body\_mass\_g}\right] \approx -2797.76 + 34.28\cdot\text{flipper\_length\_mm}\\ \text{Chinstrap} & : & \mathbb{E}\left[\text{body\_mass\_g}\right] \approx (-2797.76 - 129.71) + (34.28 - 0.37)\cdot\text{flipper\_length\_mm}\\ \text{Gentoo} & : & \mathbb{E}\left[\text{body\_mass\_g}\right] \approx (-2797.76 - 4216.63) + (34.28 + 21.39)\cdot\text{flipper\_length\_mm}\end{array}$$

From the three models above, we can make our usual interpretations. Let's see those models graphically.

```r
new_data <- crossing(flipper_length_mm = seq(min(penguins_train$flipper_length_mm,
                                                 na.rm = TRUE),
                                             max(penguins_train$flipper_length_mm,
                                                 na.rm = TRUE),
                                             by = 1),
                     species = c("Adelie", "Chinstrap", "Gentoo")
                     )

new_data <- mass_catnum_fit %>%
  augment(new_data)

p1 <- ggplot() +
  geom_point(data = penguins_train,
             aes(x = flipper_length_mm,
                 y = body_mass_g,
                 color = species),
             alpha = 0.5) +
  geom_line(data = new_data,
            aes(x = flipper_length_mm,
                y = .pred,
                color = species)) +
  labs(x = "Flipper Length (mm)",
       y = "Body Mass (g)")

p2 <- ggplot() +
  geom_point(data = penguins_train,
             aes(x = flipper_length_mm,
                 y = body_mass_g,
                 color = species),
             alpha = 0.5,
             show.legend = FALSE) +
  geom_line(data = new_data,
            aes(x = flipper_length_mm,
                y = .pred,
                color = species),
            show.legend = FALSE) +
  facet_wrap(~ species) +
  labs(x = "Flipper Length (mm)",
       y = "Body Mass (g)")

(p1 / p2)
```

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





The visual shows what we see numerically in the model summary output – the data do not provide convincing evidence that the association between body mass and flipper length is different (in either intercept or slope) between Adelie and Chinstrap penguins. The data do suggest, however, that both the slope and intercept in this model is different between Adelie and Gentoo penguins.

Interaction Between Two Numerical Predictors

Let's say that we propose a model which predicts penguin `body_mass_g` using `flipper_length_mm`, `bill_length_mm`, and an interaction between these two predictors. Such a model is of the following form:

$$\mathbb{E}\left[\text{body\_mass\_g}\right] = \beta_0 + \beta_1\cdot\text{flipper\_length\_mm} + \beta_2\cdot\text{bill\_length\_mm} + \beta_3\cdot\left(\text{flipper\_length\_mm}\right)\left(\text{bill\_length\_m}\right.$$

We'll build this model similar to the previous one. Since both of our variables are numerical, we no longer need `step_dummy()` and we'll go straight to `step_interact()`.

```
mass_num_spec <- linear_reg() %>%
  set_engine("lm")

mass_num_rec <- recipe(body_mass_g ~ flipper_length_mm + bill_length_mm, data = penguins_tra
  step_interact(~ flipper_length_mm:bill_length_mm)
```

```r
mass_num_wf <- workflow() %>%
  add_model(mass_num_spec) %>%
  add_recipe(mass_num_rec)

mass_num_fit <- mass_num_wf %>%
  fit(penguins_train)

mass_num_fit %>%
  glance() %>%
  kable() %>%
  kable_styling()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	d
0.7755339	0.7728617	391.0593	290.2213	0	3	-1889.26	3788.521	3806.247	38537702	

```r
mass_num_fit %>%
  extract_fit_engine() %>%
  tidy() %>%
  kable() %>%
  kable_styling()
```

term	estimate	std.error	statistic	p.value
Intercept	5701.959158	3379.8251437	1.6870574	0.0928297
flipper_length_mm	-10.287765	17.4109650	-0.5908785	0.5551314
bill_length_mm	-242.479461	72.9847879	-3.3223288	0.0010251
flipper_length_mm_x_bill_length_mm	1.264581	0.3718553	3.4007338	0.0007813

We've obtained our estimated model and the interaction between `flipper_length_mm` and `bill_length_mm` is statistically significant. The estimated mode is

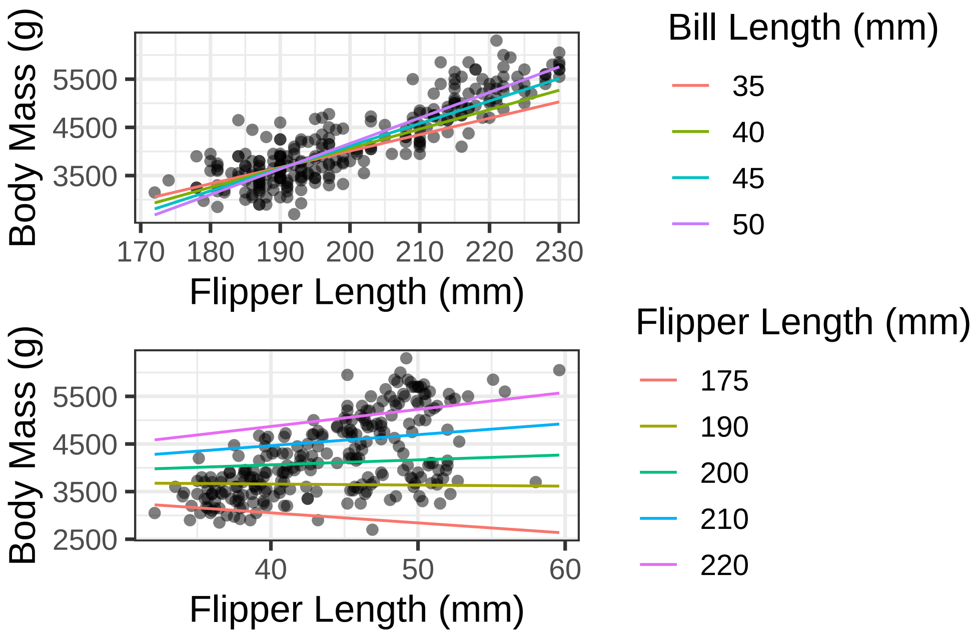
$$\mathbb{E}\left[\text{body\_mass\_g}\right] \approx 5701.96+(-10.29)\cdot\text{flipper\_length\_mm}+(-242.48)\cdot\text{bill\_length\_mm}+1.26\cdot(\text{flipper\_length\_}$$

Notice that this model allows for the expected increase in body mass due to an increased flipper length to depend on the bill length. Similarly, the expected increase in body mass due to an increased bill length depends on the flipper length in this model.

It will be easiest to understand this phenomenon by calculating a few values or by plotting our models using various assumed levels for our predictors.

```r
new_data_flipper <- crossing(flipper_length_mm = seq(min(penguins_train$flipper_length_mm,
                                                         na.rm = TRUE),
                                                     max(penguins_train$flipper_length_mm,
                                                         na.rm = TRUE),
                                                     by = 1),
                             bill_length_mm = c(35, 40, 45, 50)
                             )

new_data_bill <- crossing(bill_length_mm = seq(min(penguins_train$bill_length_mm,
                                                   na.rm = TRUE),
                                               max(penguins_train$bill_length_mm,
                                                   na.rm = TRUE),
                                               length.out = 100),
                          flipper_length_mm = c(175, 190, 200, 210, 220)
                          )

new_data_flipper <- mass_num_fit %>%
  augment(new_data_flipper)

new_data_bill <- mass_num_fit %>%
  augment(new_data_bill)

p1 <- ggplot() +
  geom_point(data = penguins_train,
             aes(x = flipper_length_mm,
                 y = body_mass_g),
             alpha = 0.5) +
  geom_line(data = new_data_flipper,
            aes(x = flipper_length_mm,
                y = .pred,
                color = as.factor(bill_length_mm))) +
  labs(x = "Flipper Length (mm)",
       y = "Body Mass (g)",
       color = "Bill Length (mm)")

p2 <- ggplot() +
  geom_point(data = penguins_train,
             aes(x = bill_length_mm,
                 y = body_mass_g),
             alpha = 0.5,
             show.legend = FALSE) +
  geom_line(data = new_data_bill,
            aes(x = bill_length_mm,
                y = .pred,
                color = as.factor(flipper_length_mm))) +
  labs(x = "Flipper Length (mm)",
       y = "Body Mass (g)",
       color = "Flipper Length (mm)")

(p1 / p2)
```

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



In each case, we can see that the slope of the modeled association between penguin body mass and the plotted predictor depends on the *level* (value) of the third variable.

Your Task

1. Use what you've learned to build a model to predict penguin `body_mass_g` using `bill_depth_mm`.

2. Update your model to include a *main effects* term for `species`.
3. Update your model with an additional term allowing for interaction between `species` and `bill_depth_mm`.
4. Interpret what you just discovered!