Lab 4: Graphical interpretation of LMs

Eric Parajon

Today's lab will focus on graphical interpretation of LMs using a variety of R tools.

Don't forget to set your working directory!

```
setwd()
```

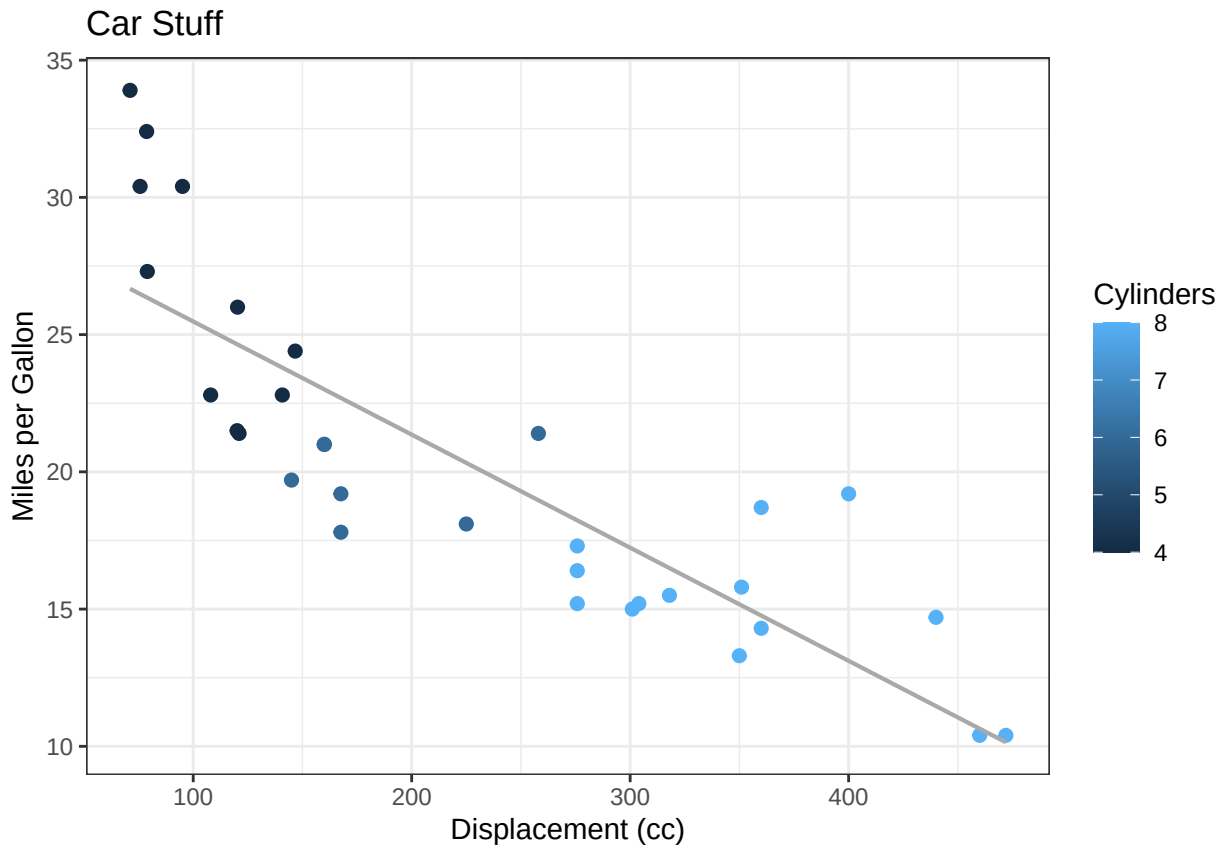
#Part 1: Ggplot visualizations Lets begin by returning to the mtcars data set and creating a basic plot of Displacement vs mpg.

```
carsplot <- ggplot(data = mtcars, aes(x = disp, y = mpg)) +
  geom_point(aes(colour = cyl), size = 2) +
  labs(title = "Car Stuff", x = "Displacement (cc)", y = "Miles per Gallon") +
  scale_colour_continuous(name = "Cylinders")+
  theme_bw()

#One method for saving plots:
#pdf("Plot1.pdf")
#carsplot
#dev.off()

#Where do these plots go?!

#Unique to ggplot:
#ggsave("Plot1a.pdf") #can edit image sizes as well
```

Okay great, but what if we wanted to plot the linear relationship between our x and y?

```
carsplot + geom_smooth(method = "lm", se = FALSE,
                       color = "darkgrey", size = 0.75)
```

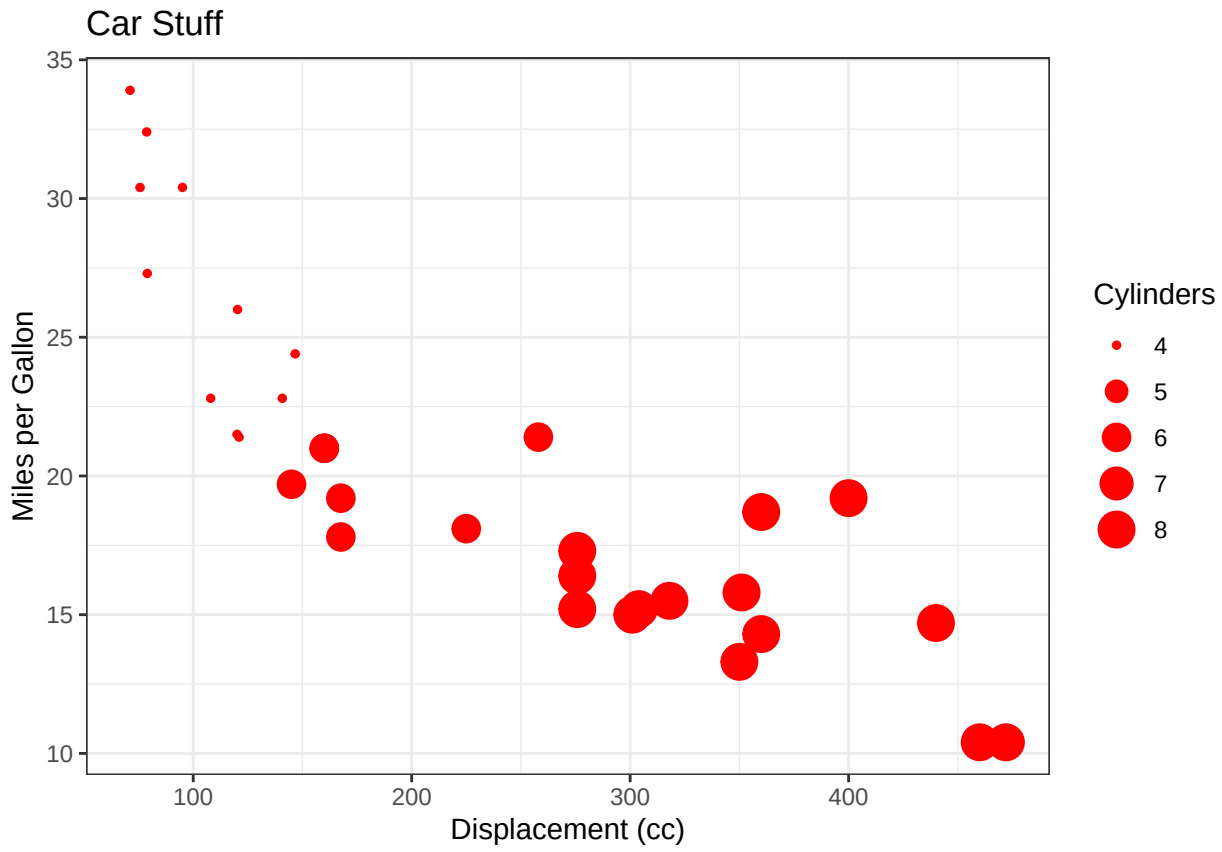
```
## `geom_smooth()` using formula 'y ~ x'
```



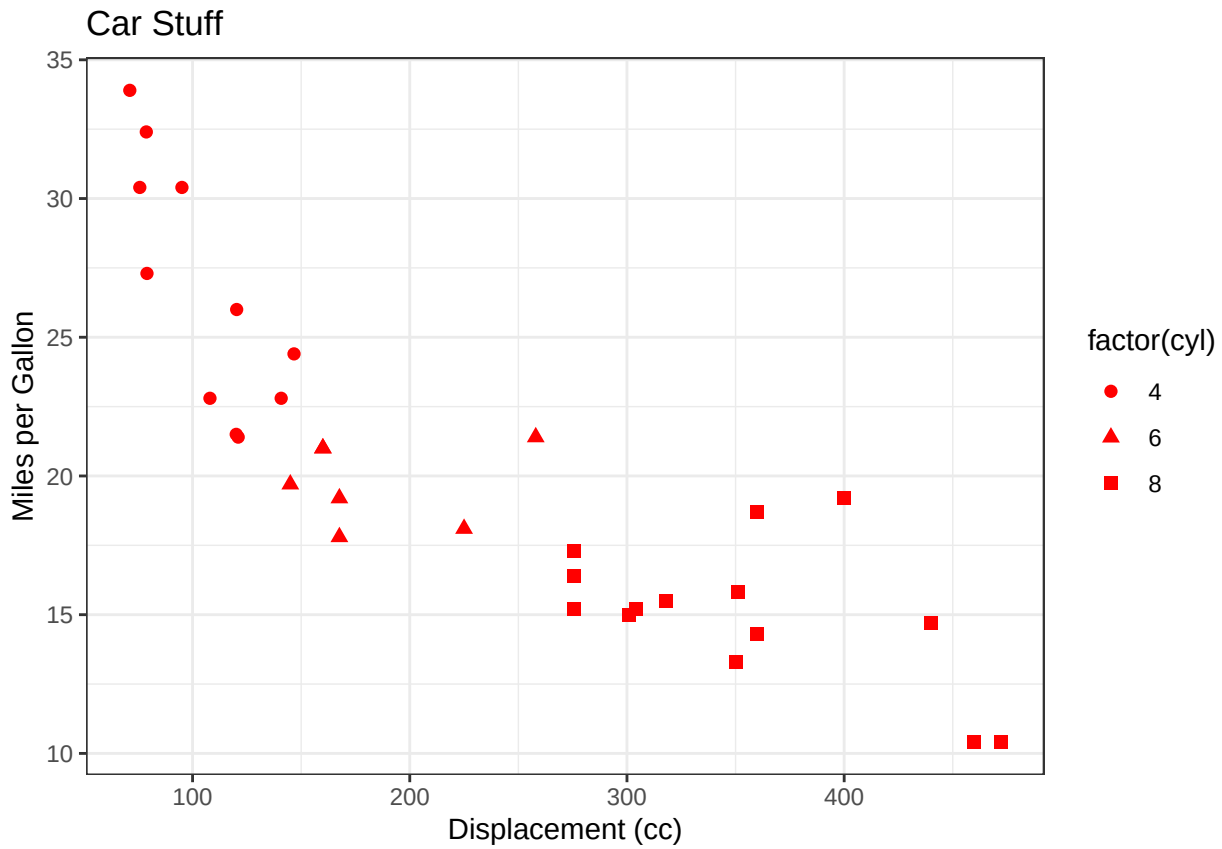

```
#geom_smooth is for fitted lines.
#se=TRUE plots confidence intervals
#We can adjust the color and size of the line plotted.
#Also, note that we just added a layer over carsplot - did we change the carsplot object itself?
```

Right now, we're using color as the aesthetic mapped to a specific explanatory variable (color represents the number of cylinders). We can play around with this:

```
ggplot(data = mtcars, aes(x = disp, y = mpg)) +
  geom_point(aes(size = cyl), colour = "red") +
  labs(title = "Car Stuff", x = "Displacement (cc)", y = "Miles per Gallon") +
  scale_size_continuous(name = "Cylinders") +
  theme_bw()
```




```
ggplot(data=mtcars, aes(x = disp, y = mpg)) +
  geom_point(aes(shape = factor(cyl)), colour = "red", size = 2) +
  labs(title = "Car Stuff", x = "Displacement (cc)", y = "Miles per Gallon") +
  scale_size_continuous(name = "Cylinders") +
  theme_bw()
```



What are the benefits/drawbacks of each of these strategies? Think about how well the plot communicates information about the types of variables plotted as well as the relationship between them.

The above are great places to start out visualizing any potential relationships between your variables.

Now let's jump back into the palmerpenguins dataset to perform some more in-depth visualization of regression models.

Here we will evaluate the association between penguin body mass (measured in grams) and flipper length (measured in mm).

Let's suppose our main research question is to determine whether body mass was associated with flipper length We will set the body mass as our dependent variable and flipper length as our independent variable (or predictor of interest).

$E[Body_mass_i|Flipper_length_i] = \beta_0 + \beta_1 Flipper_length_i + \epsilon,$

where $E[Body_mass_i|Flipper_length_i]$ denotes the expected body mass for penguin i given the flipper length of penguin i.

```
#We could start by plotting the association between the penguin's body mass and flipper length

mass_flipper <- ggplot(data = penguins,
                       aes(x = flipper_length_mm,
                           y = body_mass_g))+
  geom_point()

#Does there seem to be an association?
```

```
#Yes! as flipper length increases, the body mass also increases. There appears to be a positive relatio
```

Now, we can construct a linear regression model with body mass as the dependent variable and flipper length. as the independent variable (or predictor of interest).

```
linear.model1 <- lm(body_mass_g ~ flipper_length_mm, data = penguins)
summary(linear.model1)
```

```
##
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm, data = penguins)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1058.80  -259.27   -26.88   247.33  1288.69
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -5780.831    305.815  -18.90   <2e-16 ***
## flipper_length_mm    49.686      1.518   32.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 394.3 on 340 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.759,  Adjusted R-squared:  0.7583
## F-statistic:  1071 on 1 and 340 DF,  p-value: < 2.2e-16
```

```
#Using the broom package to tidy the model data and format into a more easily usable form.
temp_lm = broom::tidy(linear.model1, se = 'standard', conf.int = TRUE,conf.level = 0.95) %>%
  filter(term=="flipper_length_mm")


#Visualize the results in a coefficient plot
Penguin_results <- ggplot(data=temp_lm, aes(x=term, y=estimate)) +
    geom_hline(yintercept=0, color="red", size=.5) +
    geom_errorbar(aes(ymin=conf.low, ymax=conf.high, width=0), size=.5, position=position_dodge(width=0
    geom_point(aes(y=estimate), size=1.75, position = position_dodge(width=0.5)) +
    ylab("Effect of Flipper Length on Penguin Body Mass") +
    xlab("")+
    coord_flip()+
  geom_text(aes(y=estimate,label=round(estimate, digits=2)), size=2.5, vjust=-1.5, position = position_


#Okay great so huge effect! But what about confounders? Perhaps the penguins species is also important.
```

Lets again plot it but this time break out the results by species.

```
mass_flipper_species <- ggplot(data = penguins,
                       aes(x = flipper_length_mm,
                           y = body_mass_g)) +
  geom_point(aes(color = species,
                 shape = species),
             size = 3,
             alpha = 0.8) +
```

```
  scale_color_manual(values = c("darkorange","purple","cyan4")) +
  labs(title = "Flipper length and body mass",
       x = "Flipper length (mm)",
       y = "Body mass (g)",
       color = "Penguin species",
       shape = "Penguin species")

#Definitely seems like species could be important to capture in our model. So let's add to our current
```

Now our model looks like this.

$E[Body\_mass_i|Flipper\_length_i], Species_i] = \beta_0 + \beta_1 Flipper\_length_i + \beta_2 Species_i + \epsilon$

where $E[Body\_mass_i|Flipper\_length_i], Speciesi]$ denotes the expected body mass for penguin i given the flipper length of penguin i and controlling for species of the penguin i.

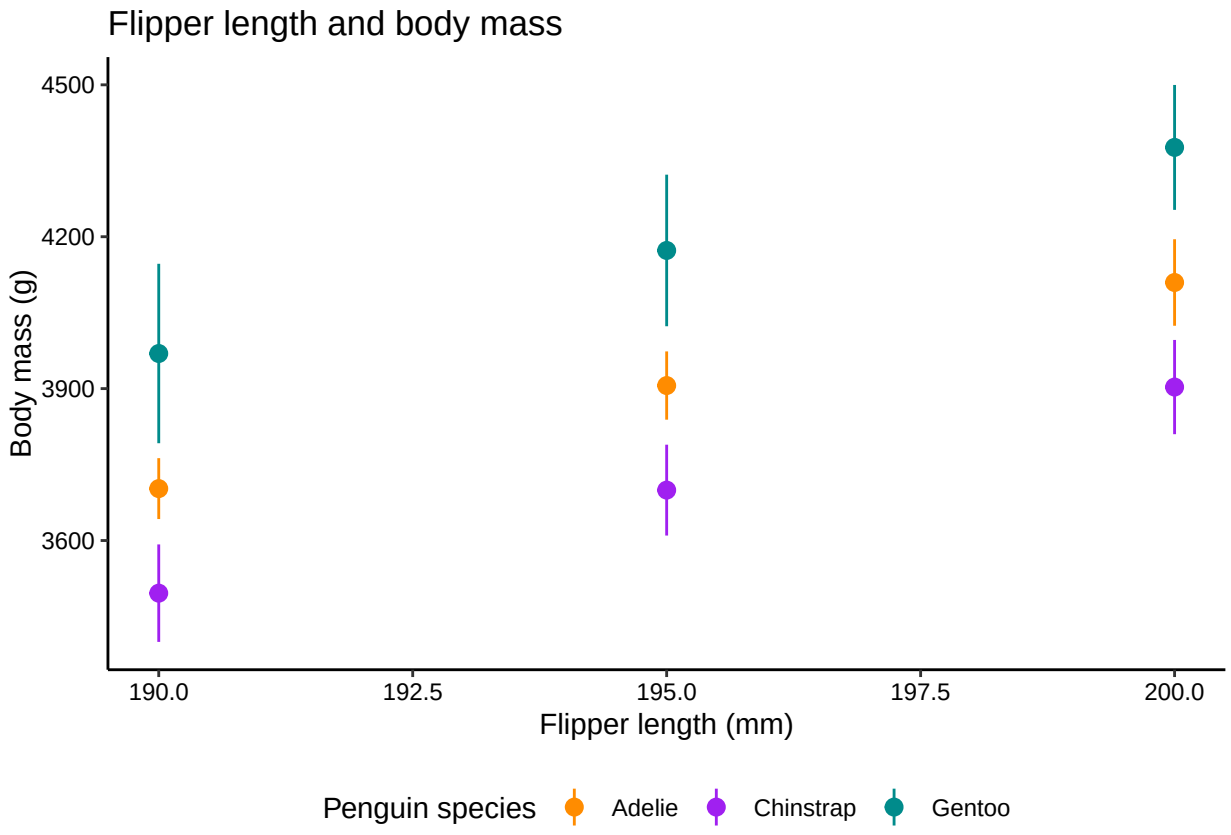
```
linear.model2 <- lm(body_mass_g ~ flipper_length_mm+species, data = penguins)
summary(linear.model2)
```

```
##
## Call:
## lm(formula = body_mass_g ~ flipper_length_mm + species, data = penguins)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -927.70 -254.82  -23.92  241.16 1191.68
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -4031.477    584.151  -6.901 2.55e-11 ***
## flipper_length_mm    40.705      3.071  13.255  < 2e-16 ***
## speciesChinstrap   -206.510     57.731  -3.577 0.000398 ***
## speciesGentoo       266.810     95.264   2.801 0.005392 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 375.5 on 338 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.7826, Adjusted R-squared:  0.7807
## F-statistic: 405.7 on 3 and 338 DF,  p-value: < 2.2e-16
```

```
#Using the broom package to tidy the model data and format into a more easily usable form.
temp_lm_2 = broom::tidy(linear.model2, se = 'standard', conf.int = TRUE,conf.level = 0.95) %>%
  filter(term!="(Intercept)")


#Visualize the results in a coefficient plot
Penguin_results_2 <- ggplot(data=temp_lm_2, aes(x=term, y=estimate)) +
    geom_hline(yintercept=0, color="red", size=.5) +
    geom_errorbar(aes(ymin=conf.low, ymax=conf.high, width=0), size=.5, position=position_dodge(width=0
    geom_point(aes(y=estimate), size=1.75, position = position_dodge(width=0.5)) +
    ylab("Effect of Flipper Length and Species on Penguin Body Mass") +
    xlab("")+
    coord_flip()+
  geom_text(aes(y=estimate,label=round(estimate, digits=2)), size=2.5, vjust=-1.5, position = position_
  plot_theme
```

Okay great, but what if we want to visualize substantive effects? For this we can use our old friend predict()!

```
newData <- expand.grid(flipper_length_mm = seq(190, 200, by = 5),
                       species=c("Adelie","Chinstrap","Gentoo"))

plotData <- as.data.frame(predict(linear.model2, newdata=newData, interval='confidence'))

#Pulling it all together
int_dat <- data.frame(newData, plotData)

subeffects_plot <- ggplot(int_dat, aes(x=flipper_length_mm, y=fit, ymin=lwr, ymax=upr,color=species)) +
        geom_pointrange() +
 scale_color_manual(values = c("darkorange","purple","cyan4")) +
  labs(title = "Flipper length and body mass",
       x = "Flipper length (mm)",
       y = "Body mass (g)",
       color = "Penguin species",
       shape = "Penguin species")
subeffects_plot+
  plot_theme
```



Interaction analysis

Additionally, in other models we may be interested in exploring the interaction between variables. Often these interaction effects are difficult to interpret without graphic aids.

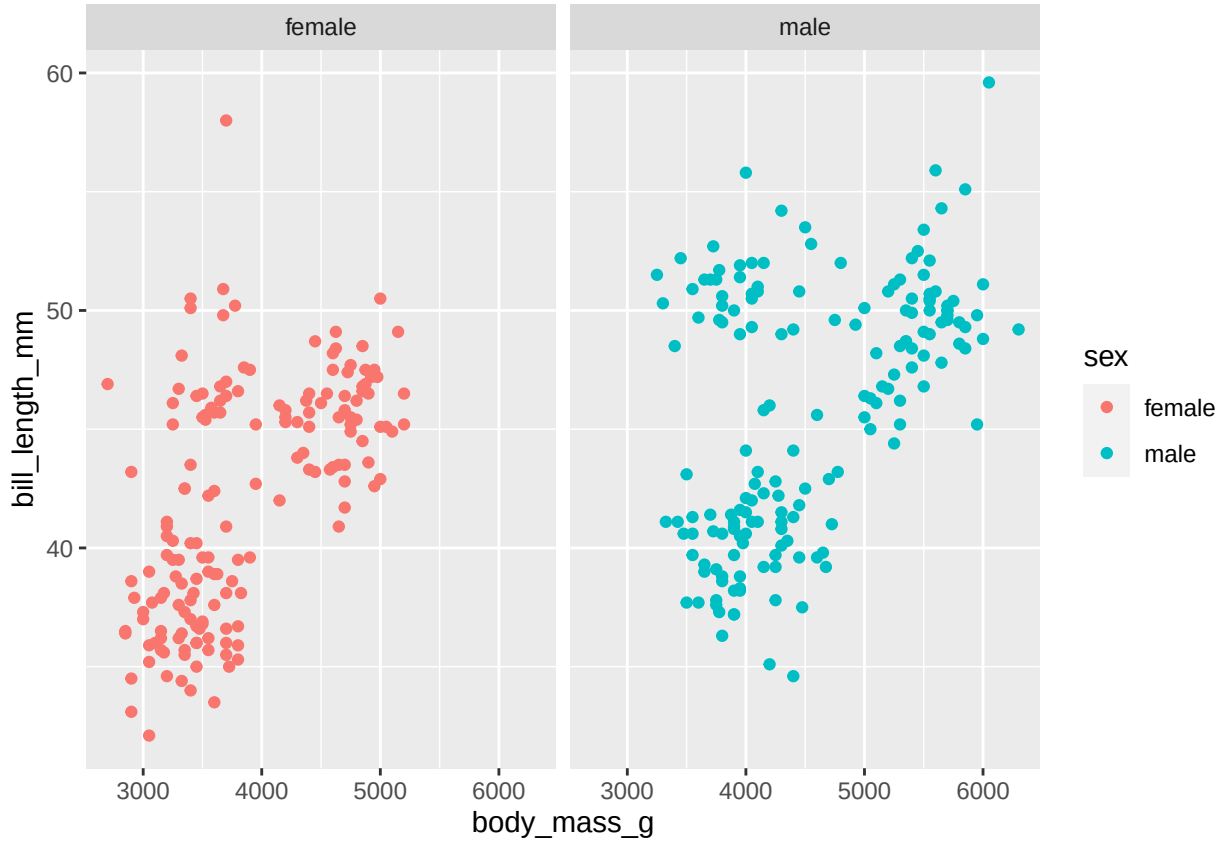
In this example, suppose we are interested in knowing if the effect of body mass on bill length is attenuated

by the sex of the penguin. We begin by plotting the data and then perform interaction analysis.

Our regression equation looks like this:

$body\_mass = B0 + B1 * bill\_length + B2 * sex + B3 * bill\_length * sex + \epsilon$

```
#First let's plot the raw data to see if there's anything there.
penguins %>%
  # drop rows with missing data
  drop_na(sex) %>%
  ggplot(aes(x = body_mass_g, y = bill_length_mm, color = sex)) +
  geom_point() +
  facet_wrap(~ sex)
```

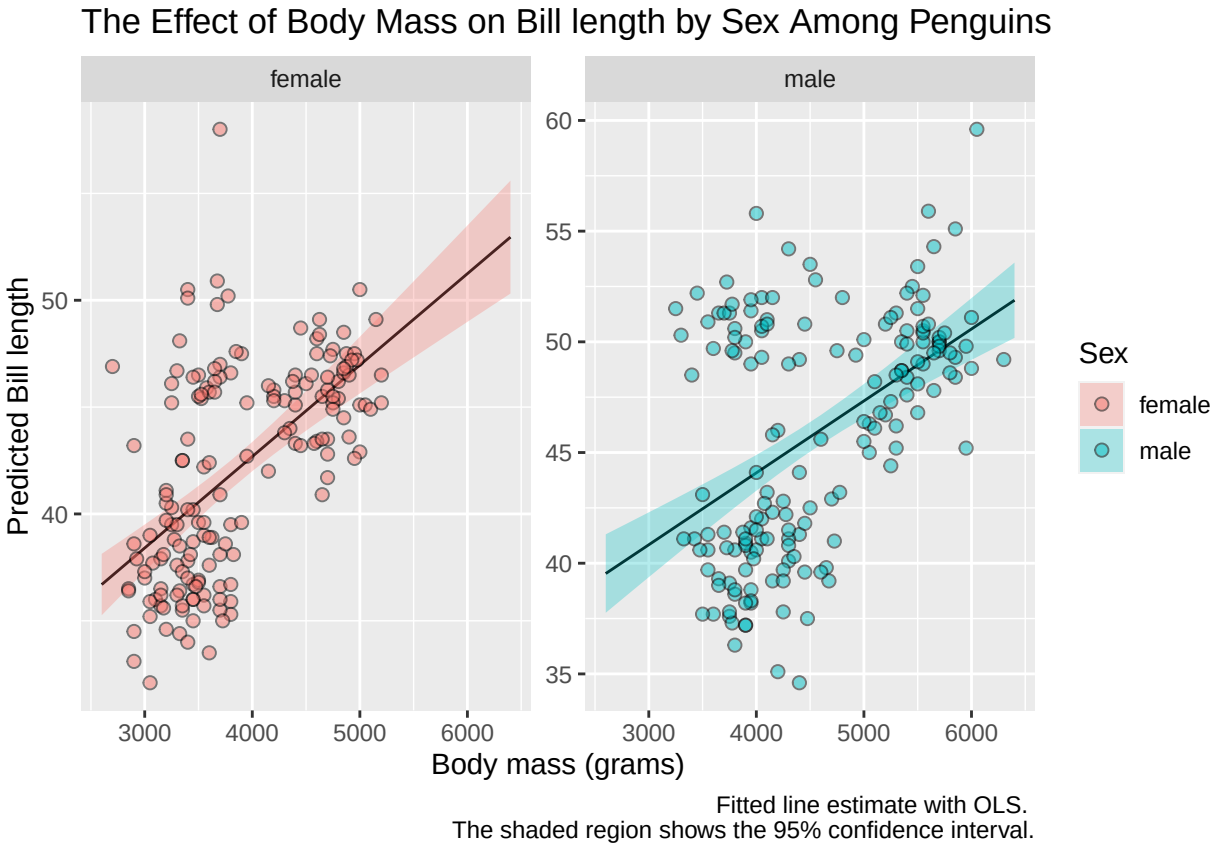



```
penguin_interact<-lm(bill_length_mm~body_mass_g*sex,data=penguins)

tidy_penguin_interact = broom::tidy(penguin_interact, se = 'standard', conf.int = TRUE,conf.level = 0.95

#Next we'll use ggpredict to get the marginal effects for the main effects and the interaction term.
#There are a ton of different packages that do similar things make_predictions from jtools is also grea

y_hat <- ggpredict(penguin_interact, terms = c("body_mass_g", "sex"))

#This returns predicted values of the dv
y_hat
```

```
## # Predicted values of bill_length_mm
##
```

```
## # sex = female
##
## body_mass_g | Predicted |         95% CI
## ----------------------------------------
##        2600 |     36.70 | [35.26, 38.13]
##        3200 |     39.26 | [38.32, 40.21]
##        3800 |     41.83 | [41.16, 42.50]
##        4400 |     44.40 | [43.54, 45.26]
##        5200 |     47.82 | [46.32, 49.33]
##        6400 |     52.96 | [50.31, 55.60]
##
## # sex = male
##
## body_mass_g | Predicted |         95% CI
## ----------------------------------------
##        2600 |     39.53 | [37.76, 41.31]
##        3200 |     41.48 | [40.17, 42.80]
##        3800 |     43.43 | [42.52, 44.35]
##        4400 |     45.38 | [44.71, 46.06]
##        5200 |     47.98 | [47.12, 48.84]
##        6400 |     51.88 | [50.18, 53.58]
```

```r
#Plotting interaction effects

  #First extracting data from the model
model_data <- penguin_interact$model
  #Next, rename group to sex
y_hat <-
  y_hat %>%
  rename(sex = group)


#Plotting
ggplot(data = y_hat, aes(x = x, y = predicted, fill = sex)) +
  #plot the fitted line
  geom_line() +
  #plot the confidence intervals
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.3) +
  #plot the model data
  geom_point(data = model_data, aes(x = body_mass_g, y = bill_length_mm),
             size = 2, shape = 21, alpha = 0.5) +
  facet_wrap(~ sex, scales = "free") +
  labs(title = "The Effect of Body Mass on Bill length by Sex Among Penguins",
       x = "Body mass (grams)",
       y = "Predicted Bill length",
       fill = "Sex",
       caption = "Fitted line estimate with OLS. \n The shaded region shows the 95% confidence interval
```



#Part 3: LM practice Use the msleep dataset to model how body weight (bodywt) and brain weight (brainwt) influences total REM sleep (sleep_rem). Then create a model including an additional predictor (of your choice). Create a coefficient plot of the results of the two models and compare them.

```
glimpse(msleep)
```

```
## Rows: 83
## Columns: 11
## $ name         <chr> "Cheetah", "Owl monkey", "Mountain beaver", "Greater shor~
## $ genus        <chr> "Acinonyx", "Aotus", "Aplodontia", "Blarina", "Bos", "Bra~
## $ vore         <chr> "carni", "omni", "herbi", "omni", "herbi", "herbi", "carn~
## $ order        <chr> "Carnivora", "Primates", "Rodentia", "Soricomorpha", "Art~
## $ conservation <chr> "lc", NA, "nt", "lc", "domesticated", NA, "vu", NA, "dome~
## $ sleep_total  <dbl> 12.1, 17.0, 14.4, 14.9, 4.0, 14.4, 8.7, 7.0, 10.1, 3.0, 5~
## $ sleep_rem    <dbl> NA, 1.8, 2.4, 2.3, 0.7, 2.2, 1.4, NA, 2.9, NA, 0.6, 0.8, ~
## $ sleep_cycle  <dbl> NA, NA, NA, 0.1333333, 0.6666667, 0.7666667, 0.3833333, N~
## $ awake        <dbl> 11.9, 7.0, 9.6, 9.1, 20.0, 9.6, 15.3, 17.0, 13.9, 21.0, 1~
## $ brainwt      <dbl> NA, 0.01550, NA, 0.00029, 0.42300, NA, NA, NA, 0.07000, 0~
## $ bodywt       <dbl> 50.000, 0.480, 1.350, 0.019, 600.000, 3.850, 20.490, 0.04~
```

```
#Creating model 1
model1 <- lm(sleep_rem ~ bodywt + brainwt, data = msleep)

#Tidying model 1
model1_tidy<-broom::tidy(model1, se = 'standard', conf.int = TRUE,conf.level = 0.95) %>%
  mutate(model = "Model 1")

#Creating model 2
```

```r
model2 <- lm(formula = sleep_rem ~ bodywt + brainwt + sleep_total, data = msleep)

#Tidying model 2
model2_tidy<-broom::tidy(model2, se = 'standard', conf.int = TRUE,conf.level = 0.95) %>%
  mutate(model = "Model 2")

#Combine results into a single dataframe
sleep_results <- bind_rows(model1_tidy, model2_tidy) %>%
  filter(term!="(Intercept)")


#Plotting (make sure to assign color and shape to the model variable in the ggplot call)
sleep_results_plot <- ggplot(data=sleep_results, aes(x=term, y=estimate,
           color = model, shape = model)) +
    geom_hline(yintercept=0, color="red", size=.5) +
    geom_errorbar(aes(ymin=conf.low, ymax=conf.high, width=0), size=.5, position=position_dodge(width=0
    geom_point(aes(y=estimate), size=1.75, position = position_dodge(width=0.5)) +
  labs(title = "Model Estimates of Brain and Body Weight (and total sleep) on REM Sleep",
       x = "Predictor",
       y = "Coefficient Estimate",
       caption = "Models fit with OLS. Error bars show the 95% confidence interval.",
       color = "Model:",
       shape = "Model:") +
   scale_x_discrete(labels = c("Body Weight", "Brain Weight", "Total Sleep"))+
  coord_flip()+
  plot_theme
```