

Homework 5- DataVis-ggplot

Jeshurun Biney

DUE: Thursday, March 28

Necessary items:

- **libraries:** tidyverse, ggsci, ggforce, patchwork, Hmisc
- **files:** bloom_df.csv, hw_task2_blank.png

Within RStudio you can directly edit these blocks of code and turn in your homework that way with your code and figures printed.

I **HIGHLY** recommend getting your code to work in a normal R script then just copy and pasting the final code over to this document

First: import libraries, set your working directory, and read in bloom_df

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(tidyverse)
library(ggforce)
library(ggsci)
library(patchwork)
library(Hmisc)

setwd('C:/Users/Administrator/Desktop/BIOL_792/Data_Science_For_Biology_II/Part.4.DataVisualization/ggplot') #change to match your ggplot directory

bloom_df <- read.csv('bloom_df.csv')
```

bloom_df contents

- **taxa:** genus_species
- **genus**
- **species**
- **logbodysize**
- **trophic_position**
- **reg:** diadromous or non-diadromous

Let's take a peak and look at the structure

```
bloom_df[1:5,]
```

```
##           taxa genus      species logbodysize trophic_position
## 1      Alosa_alabamae Alosa      alabamae    1.707570      0.4313638
## 2      Alosa_alosa Alosa        alosa      1.778151      0.5563025
## 3      Alosa_fallax Alosa        fallax      1.778151      0.5563025
## 4      Alosa_mediocris Alosa      mediocris    1.778151      0.6127839
## 5 Alosa_pseudoharengus Alosa pseudoharengus    1.602060      0.5440680
##           reg
## 1 diadromous
## 2 diadromous
## 3 diadromous
## 4 diadromous
## 5 diadromous
```

```
str(bloom_df)
```

```
## 'data.frame':   50 obs. of  6 variables:
## $ taxa          : chr  "Alosa_alabamae" "Alosa_alosa" "Alosa_fallax" "Alosa_mediocris" ...
## $ genus         : chr  "Alosa" "Alosa" "Alosa" "Alosa" ...
## $ species       : chr  "alabamae" "alosa" "fallax" "mediocris" ...
## $ logbodysize   : num  1.71 1.78 1.78 1.78 1.6 ...
## $ trophic_position: num  0.431 0.556 0.556 0.613 0.544 ...
## $ reg           : chr  "diadromous" "diadromous" "diadromous" "diadromous" ...
```

Graphical tasks

This next tasks are going to ask you to make a series of figures. For each one, I would like you to make all of them *look good*. Change the theme around, change the axis labels, etc. Additionally, each of these will use the **reg** column which has 2 factors. Change the color and/or shape of them both.

Task 1: Continuous vs. continuous plot

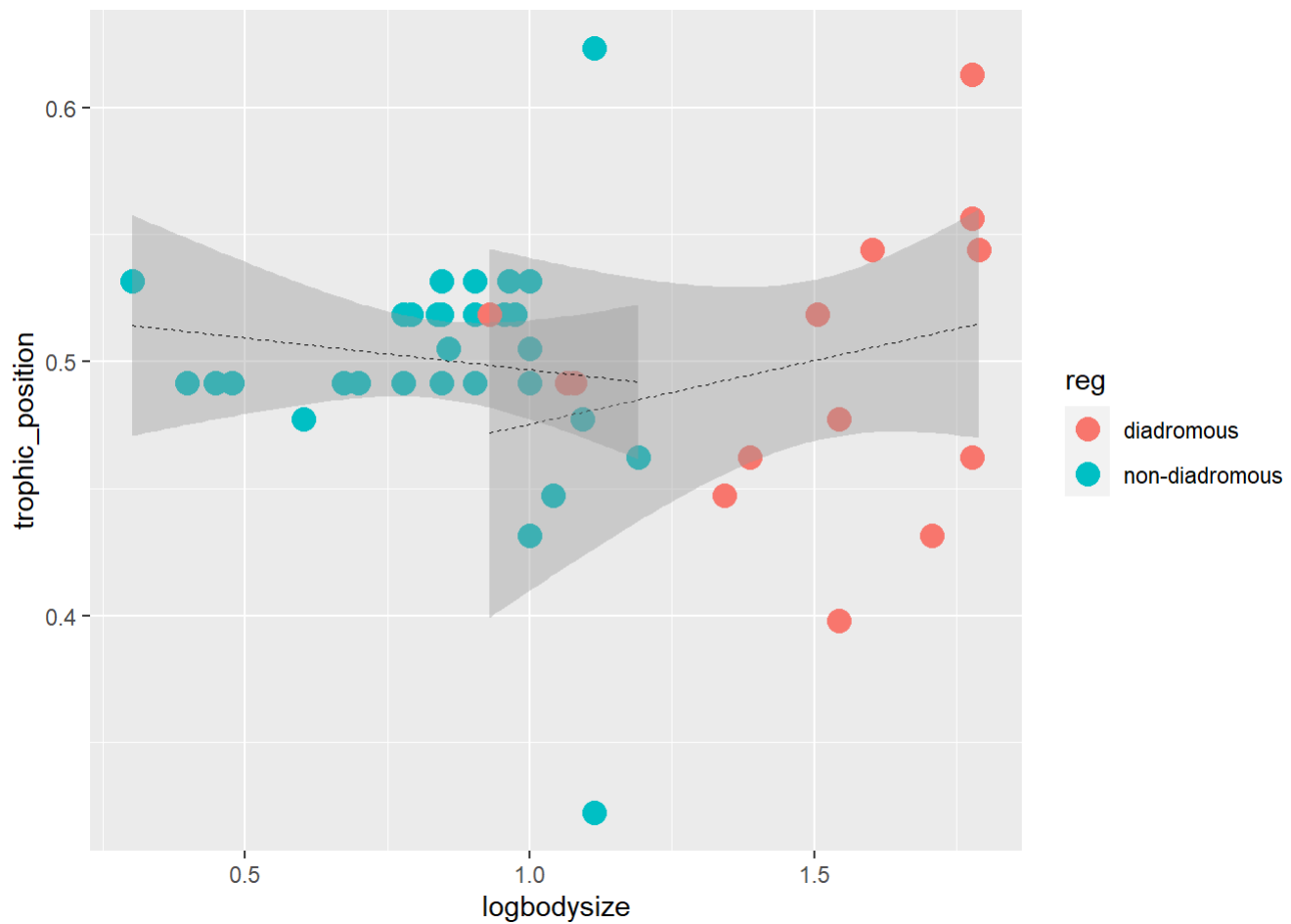
Create two scatterplots of logbodysize vs. trophic_position grouped by reg

First plot:

- a 2 trend lines (method = 'lm'), one for each reg variable

```
#plot 1.1
ggplot(data = bloom_df, aes(x = logbodysize, y = trophic_position, color = reg)) +
  geom_point(size = 4) +
  stat_smooth(method = 'lm', formula = y ~ x, aes(group = reg),
             linetype = 'dashed', color = 'black', size = 0.25)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

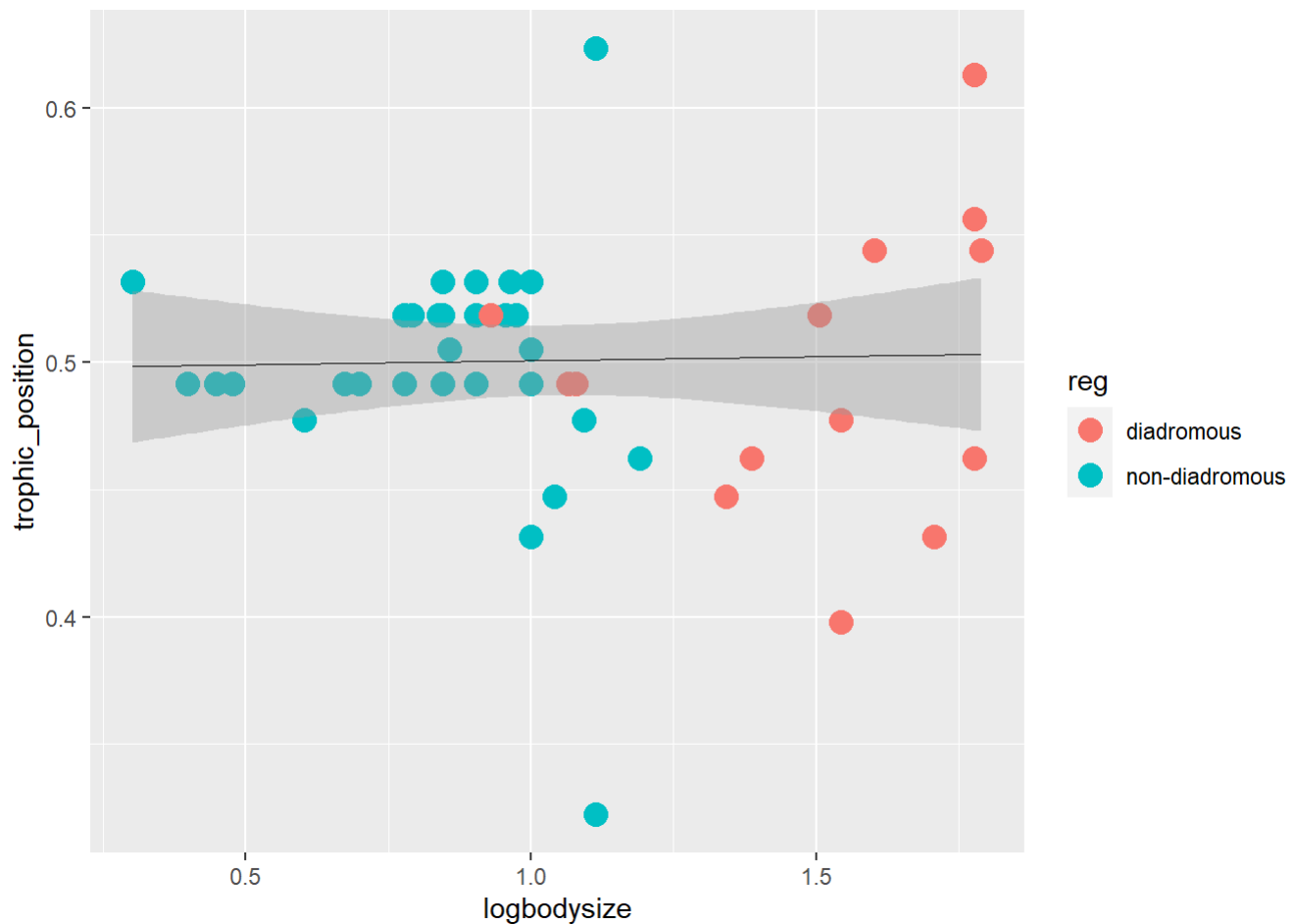


Second plot

- a single trend line for the whole model

#plot 1.2

```
ggplot(data = bloom_df, aes(x = logbodysize, y = trophic_position, color = reg)) +
  geom_point(size = 4) +
  stat_smooth(method = 'lm', formula = y ~ x, color = 'black', size = 0.25)
```



Task 2: Continuous vs. categorical

The layout will be:

- 2 panels: trophic_position and logbodysize
- x axis: reg
- y axis: value of trophic_position and logbodysize

Create 4 plots:

- bar and error bars (mean and 95% conf. int.)
- point and error bars (mean and 95% conf. int.)
- box plots
- raw data + point and error bars (mean and 95% conf. int.)

All these plots should have the same general layout but change the type of plot to visualize the data.

HINT: logbodysize and trophic_position need to be turned into long format

Example layout:

sample layout for task2

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(ggpubr)
#plot 2.1
# Transform data to long format
bloom_df_long <- bloom_df %>%
  select(reg, trophic_position, logbodysize) %>%
  pivot_longer(cols = c(trophic_position, logbodysize), names_to = "variable", values_to = "value")

# Create bar plot
plot_bar <- ggplot(data = bloom_df_long, aes(x = reg, y = value, fill = variable)) +
  geom_bar(stat = "summary", fun = "mean", position = position_dodge(), color = "black") +
  geom_errorbar(stat = "summary", fun.data = mean_cl_normal, position = position_dodge(width =
0.9), width = 0.2) +
  facet_wrap(vars(variable), scales = "free_y", ncol = 2) +
  labs(x = "reg", y = "Value", fill = NULL) +
  theme_bw()
```

#plot 2.2

```
# Create point plot
plot_point <- ggplot(data = bloom_df_long, aes(x = reg, y = value, color = variable)) +
  geom_point(size = 2, position = position_dodge(width = 0.5)) +
  stat_summary(fun = "mean", geom = "point", size = 3, color = "black", position = position_dodge(width = 0.5)) +
  geom_errorbar(stat = "summary", fun.data = mean_cl_normal, width = 0.2, position = position_dodge(width = 0.5)) +
  facet_wrap(vars(variable), scales = "free_y", ncol = 2) +
  labs(x = "reg", y = "Value", color = NULL) +
  theme_bw()
```

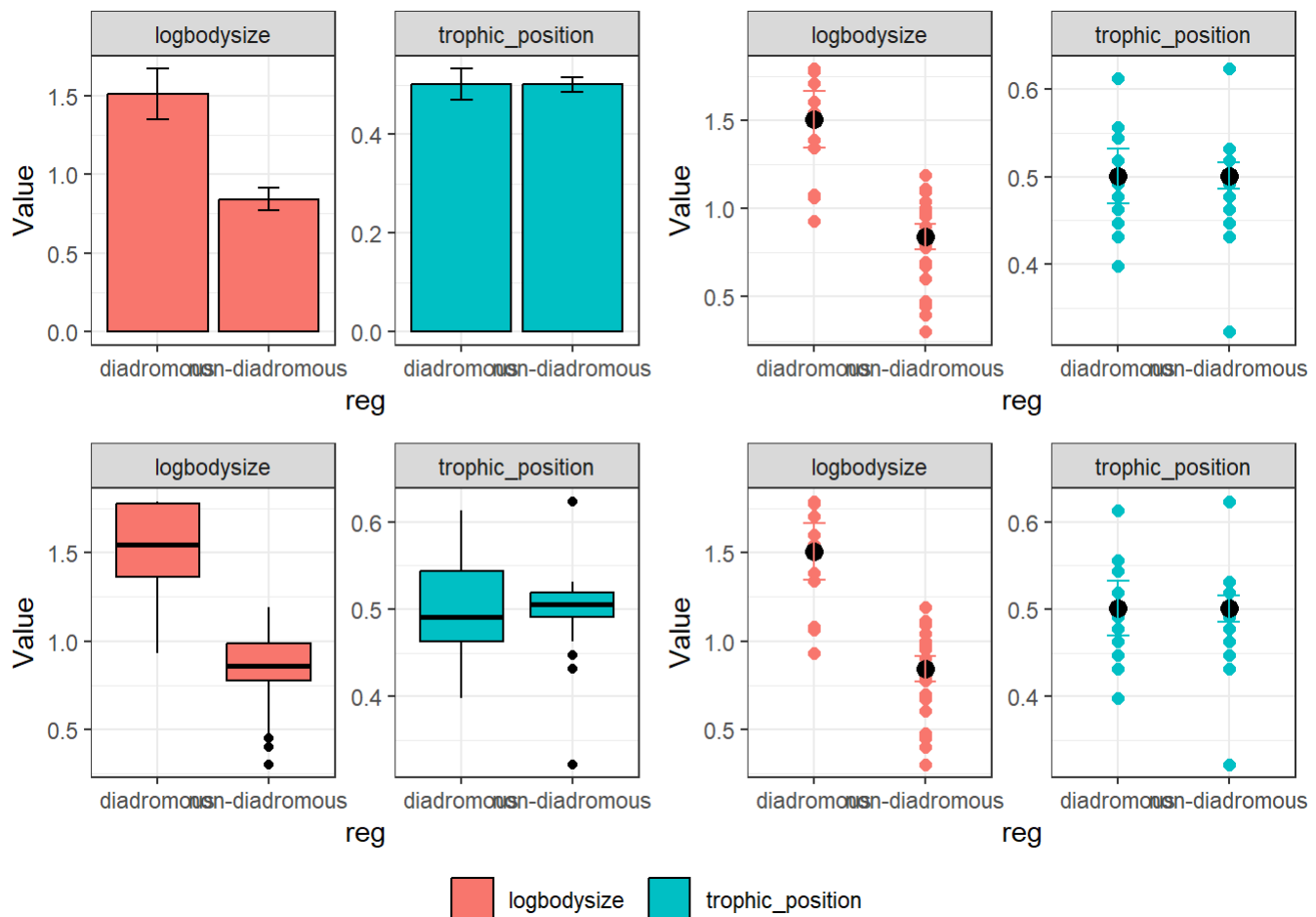
#plot 2.3

```
# Create box plot
plot_box <- ggplot(data = bloom_df_long, aes(x = reg, y = value, fill = variable)) +
  geom_boxplot(color = "black", position = position_dodge(width = 0.9)) +
  facet_wrap(vars(variable), scales = "free_y", ncol = 2) +
  labs(x = "reg", y = "Value", fill = NULL) +
  theme_bw()
```

#plot 2.4

Create raw data plot

```
plot_raw <- ggplot(data = bloom_df_long, aes(x = reg, y = value, color = variable)) +
  geom_point(size = 2) +
  stat_summary(fun = "mean", geom = "point", size = 3, color = "black") +
  geom_errorbar(stat = "summary", fun.data = mean_cl_normal, width = 0.2) +
  facet_wrap(vars(variable), scales = "free_y", ncol = 2) +
  labs(x = "reg", y = "Value", color = NULL) +
  theme_bw()
# Combine plots using ggarrange
ggarrange(plot_bar, plot_point, plot_box, plot_raw, nrow = 2, ncol = 2, common.legend = TRUE, legend = "bottom")
```



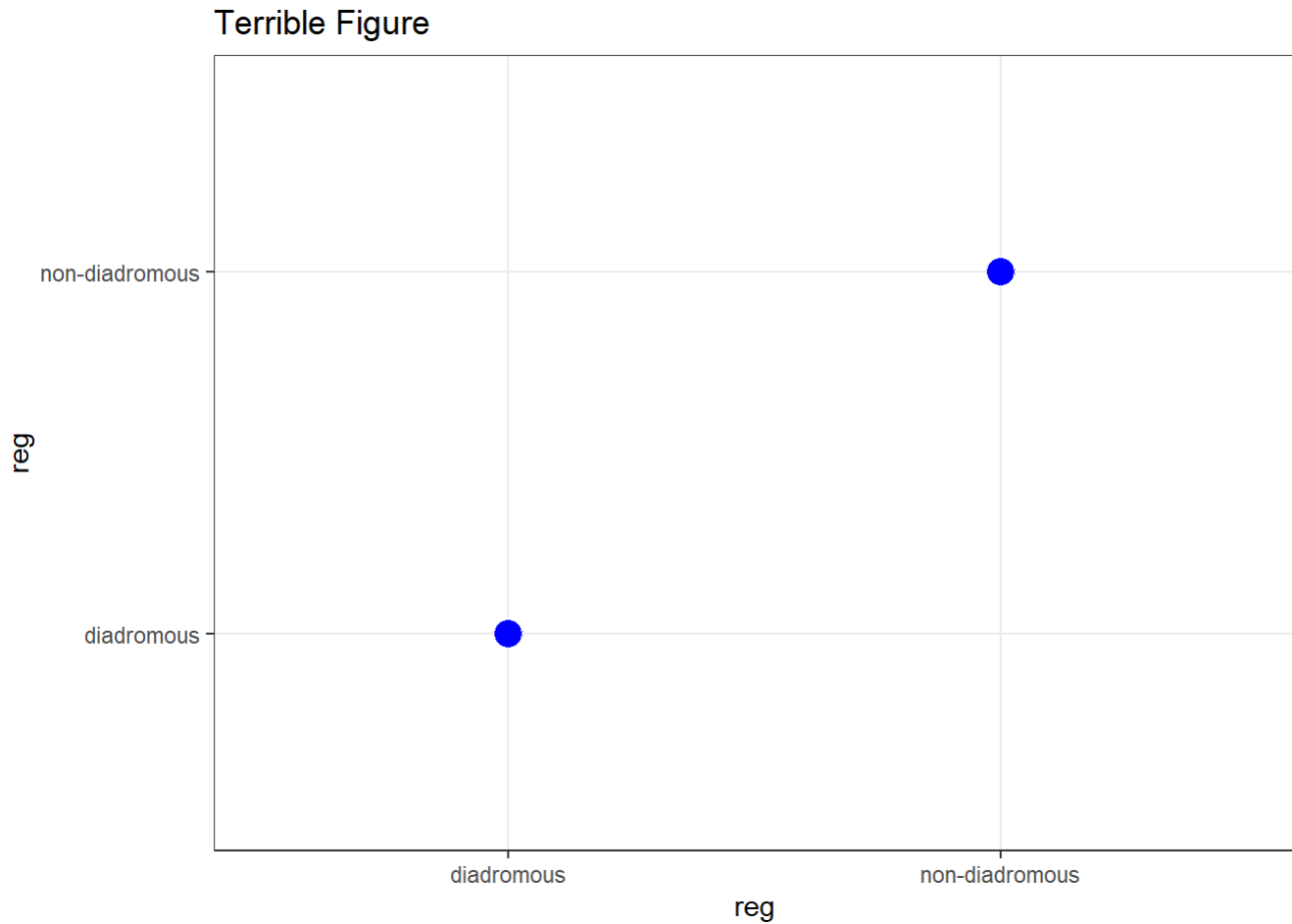
Task 3: layering your plots

Make any two figures with the bloom_df! Can be something we've done or something completely different. Make one TERRIBLE figure and one GREAT figure.

Put them together using *patchwork* or *ggarrange*

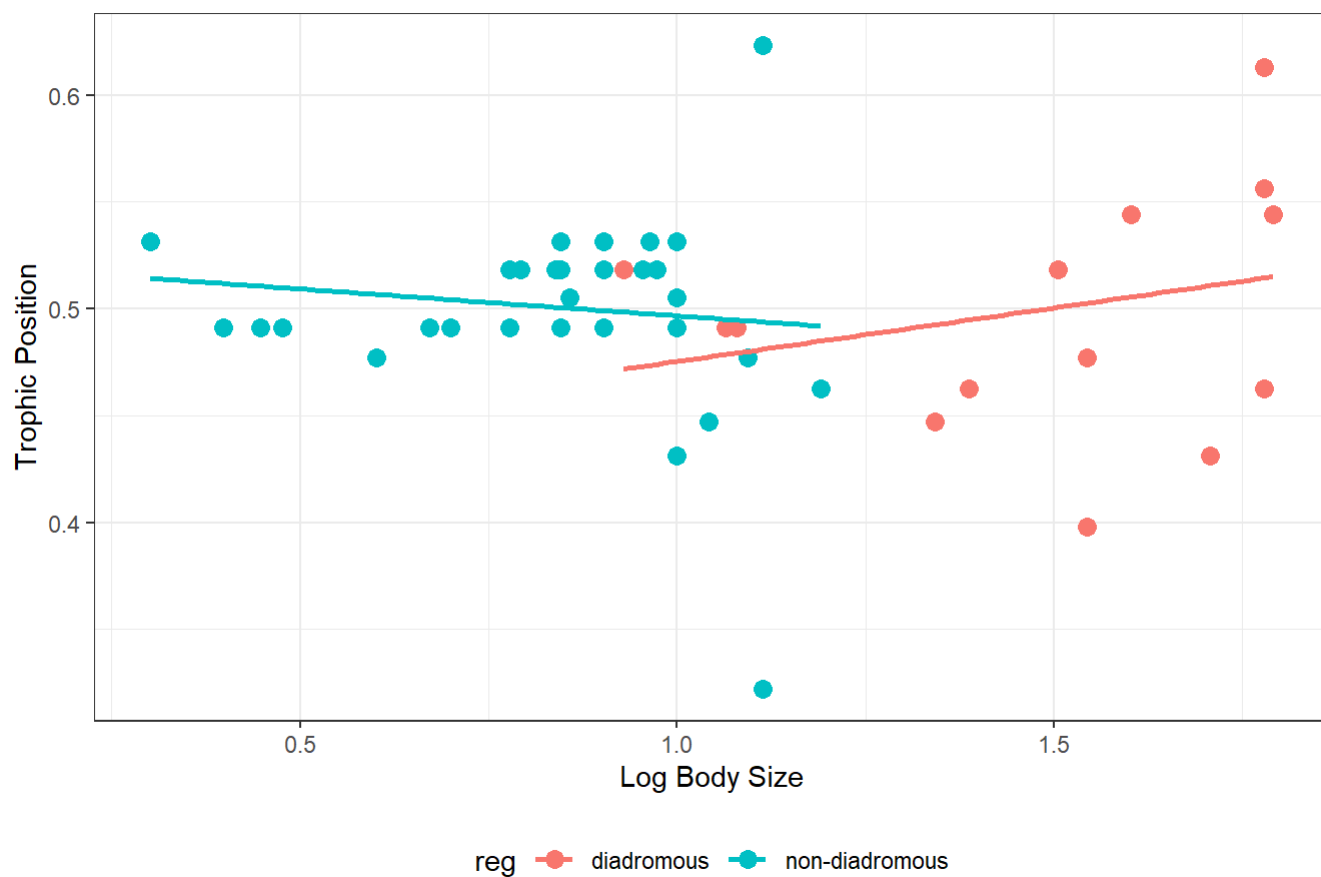
#plot 3

```
ggplot(data = bloom_df, aes(x = reg, y = reg)) +
  geom_point(size = 5, shape = 21, fill = "blue", colour = "white") +
  theme_bw() +
  labs(x = "reg", y = "reg", title = "Terrible Figure")
```



```
ggplot(data = bloom_df, aes(x = logbodysize, y = trophic_position, colour = reg)) +
  geom_point(size = 3) +
  stat_smooth(method = "lm", se = FALSE, size = 1, formula = y ~ x, aes(group = reg, colour = reg)) +
  labs(x = "Log Body Size", y = "Trophic Position", title = "Great Figure") +
  theme_bw() +
  theme(legend.position = "bottom")
```

Great Figure



```
library(patchwork)
```

```
# Combine the terrible and great figures
#terrible_fig + great_fig + plot_layout(ncol = 2)
```

Bonus: make a map (not required)

In the github for this course, there is an Rmarkdown called *ggmap_ex*. You can use this as a reference to make a map using any data you want. Try using your own data so it is useful.

```
#bonus map
```