Week 5 Assignment

Ellen Bledsoe

2024-02-08

Week 5 Assignment

Assignment Exercises

Set-up

```r
download.file("https://ndownloader.figshare.com/files/5629542",
              "ACACIA_DREPANOLOBIUM_SURVEY.txt")
download.file("https://ndownloader.figshare.com/files/5629536",
              "TREE.txt")
download.file("https://esapubs.org/archive/ecol/E084/093/Mammal_lifehistories_v2.txt",
              "Mammal_lifehistories_v2.txt")
```

Read in the packages we will need. You can either load all of them individually (`readr`, `dplyr`, `ggplot2`) or load the `tidyverse` "meta" package.

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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 error
```

1. Acacia and Ants (20 pts)

Check to see if `ACACIA_DREPANOLOBIUM_SURVEY.txt` is in your workspace. If not, download it. Read it into R using the following command:

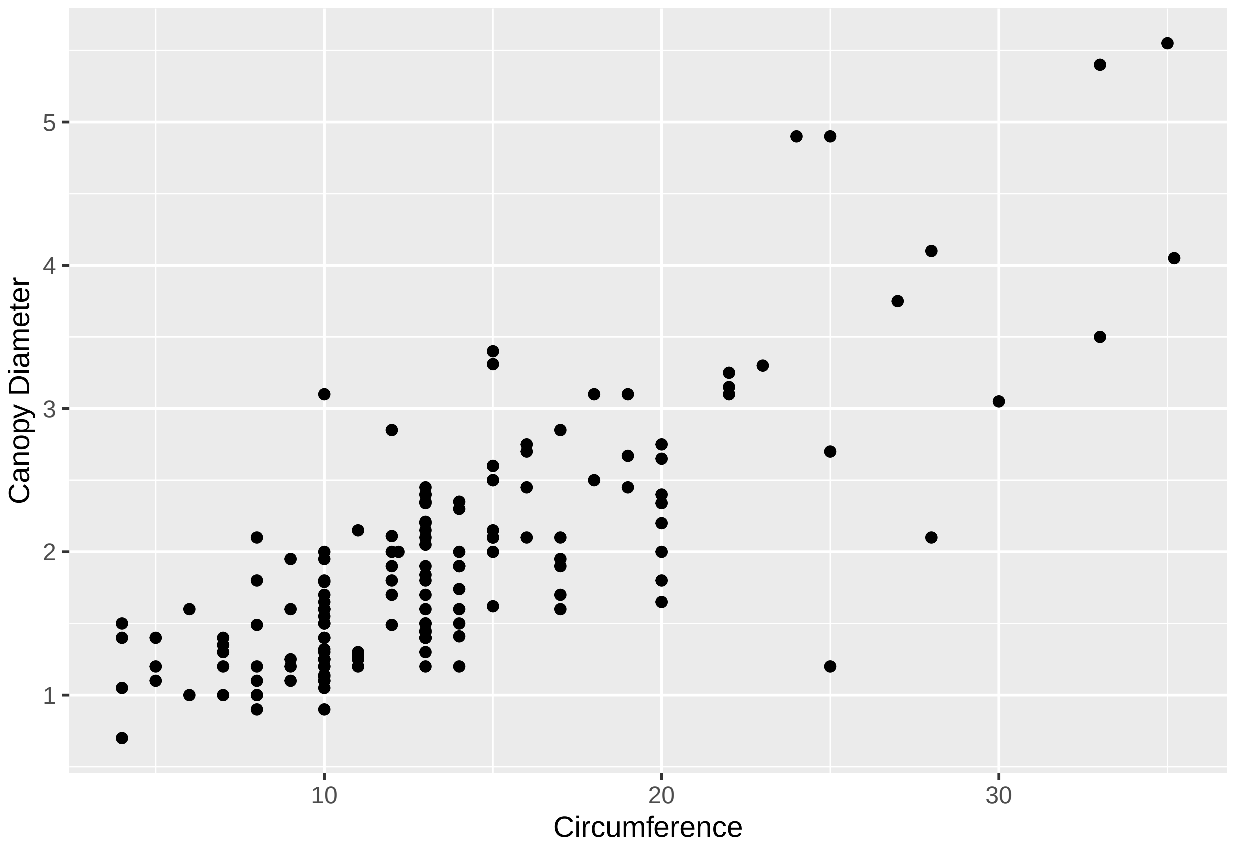
```r
acacia <- read_tsv("ACACIA_DREPANOLOBIUM_SURVEY.txt", na = c("", "dead"))
```

```
## Rows: 157 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr  (4): SITE, TREATMENT, PLOT, ANT
## dbl (11): SURVEY, YEAR, BLOCK, ID, HEIGHT, AXIS1, AXIS2, CIRC, FLOWERS, BUDS...
##
## 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.
```

1. Make a scatter plot with `CIRC` on the x axis and `AXIS1` (the maximum canopy width) on the y axis. Label the x axis "Circumference" and the y axis "Canopy Diameter".
2. The same plot as (1), but with both axes scaled logarithmically (using `scale_x_log10` and `scale_y_log10`).
3. The same plot as (1), but with points colored based on the `ANT` column (the species of ant symbiont living with the acacia)
4. The same plot as (3)), but instead of different colors show different species of ant (values of `ANT`) each in a separate subplot.
5. The same plot as (4) but add a simple model of the data by adding `geom_smooth`.

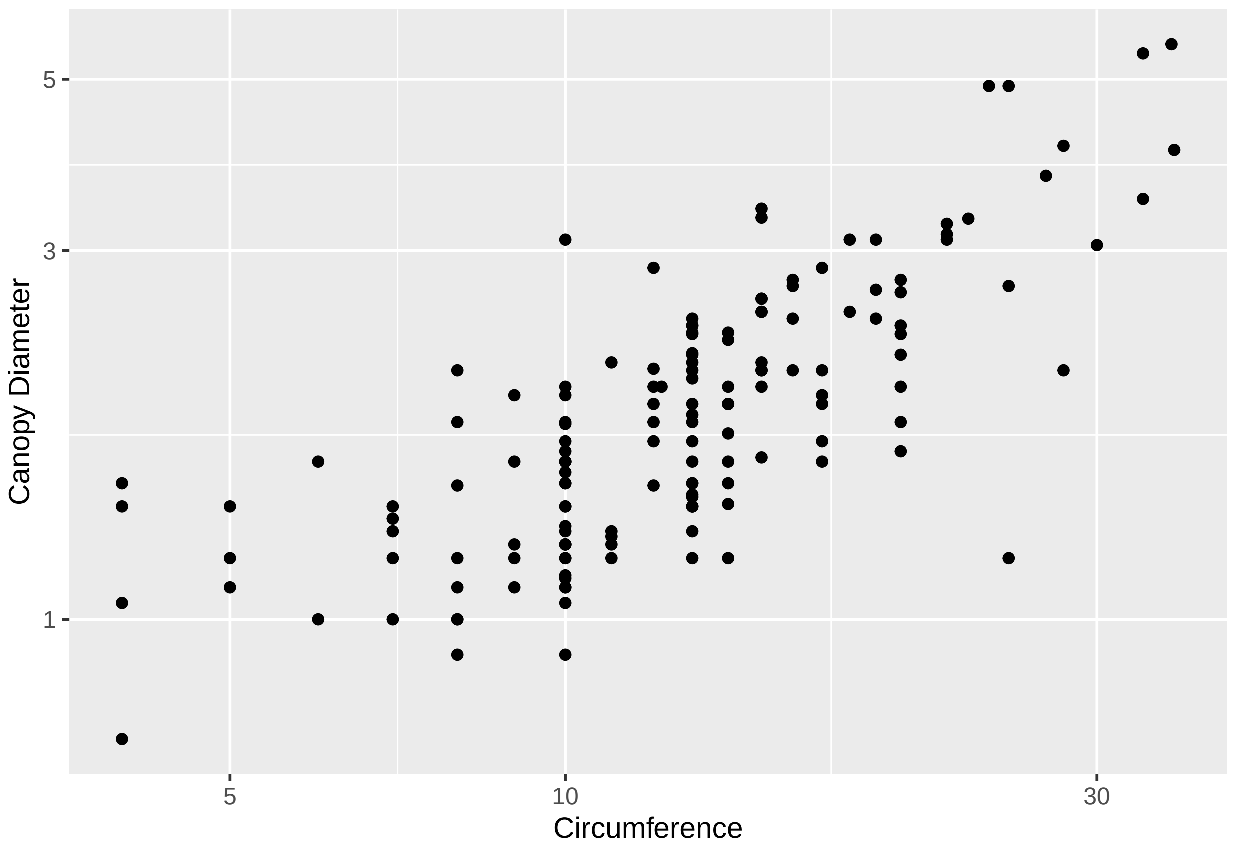
```
# 1

ggplot(data = acacia, mapping = aes(x = CIRC, y = AXIS1)) +
  geom_point() +
  labs(x = "Circumference", y = "Canopy Diameter")
```

```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



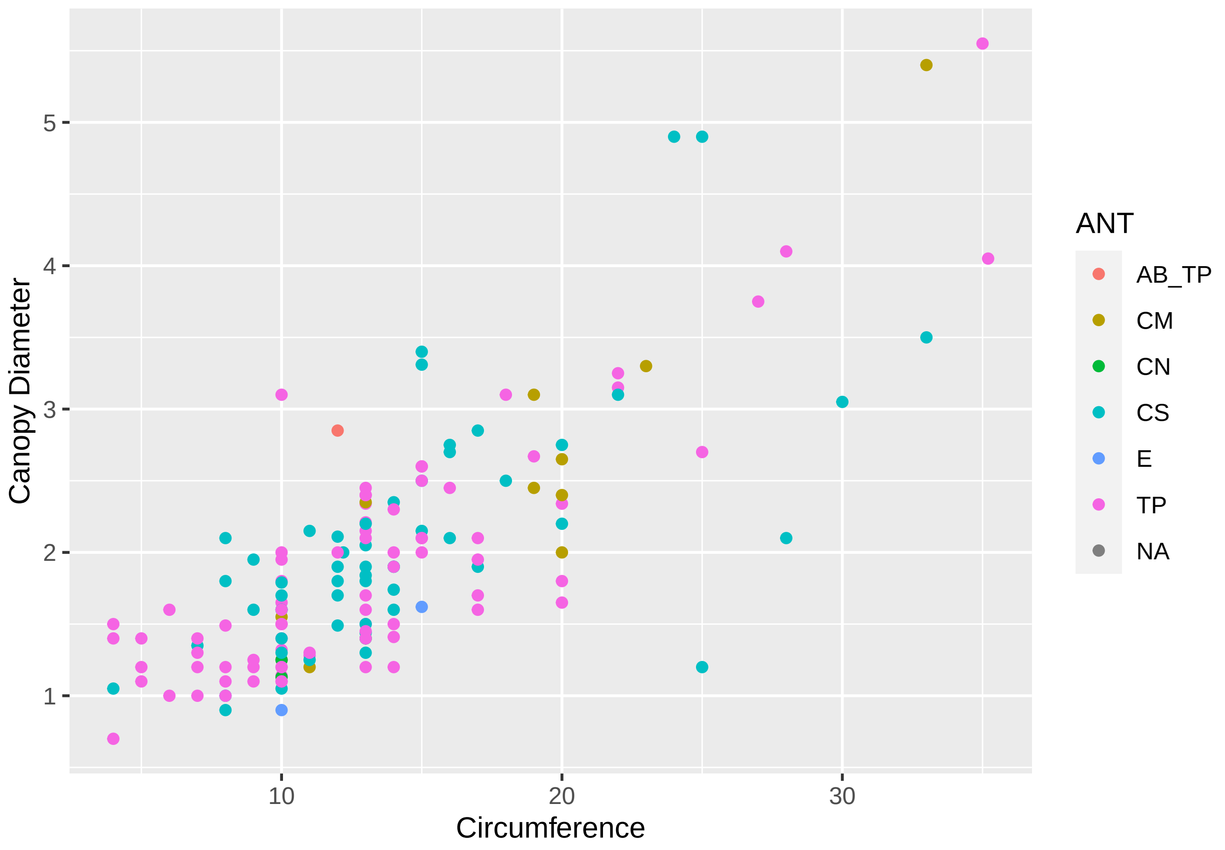
```
#ggsave("Graphing-acacia-ants-R-1.jpeg")

# 2

ggplot(data = acacia, mapping = aes(x = CIRC, y = AXIS1)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Circumference", y = "Canopy Diameter")
```

```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



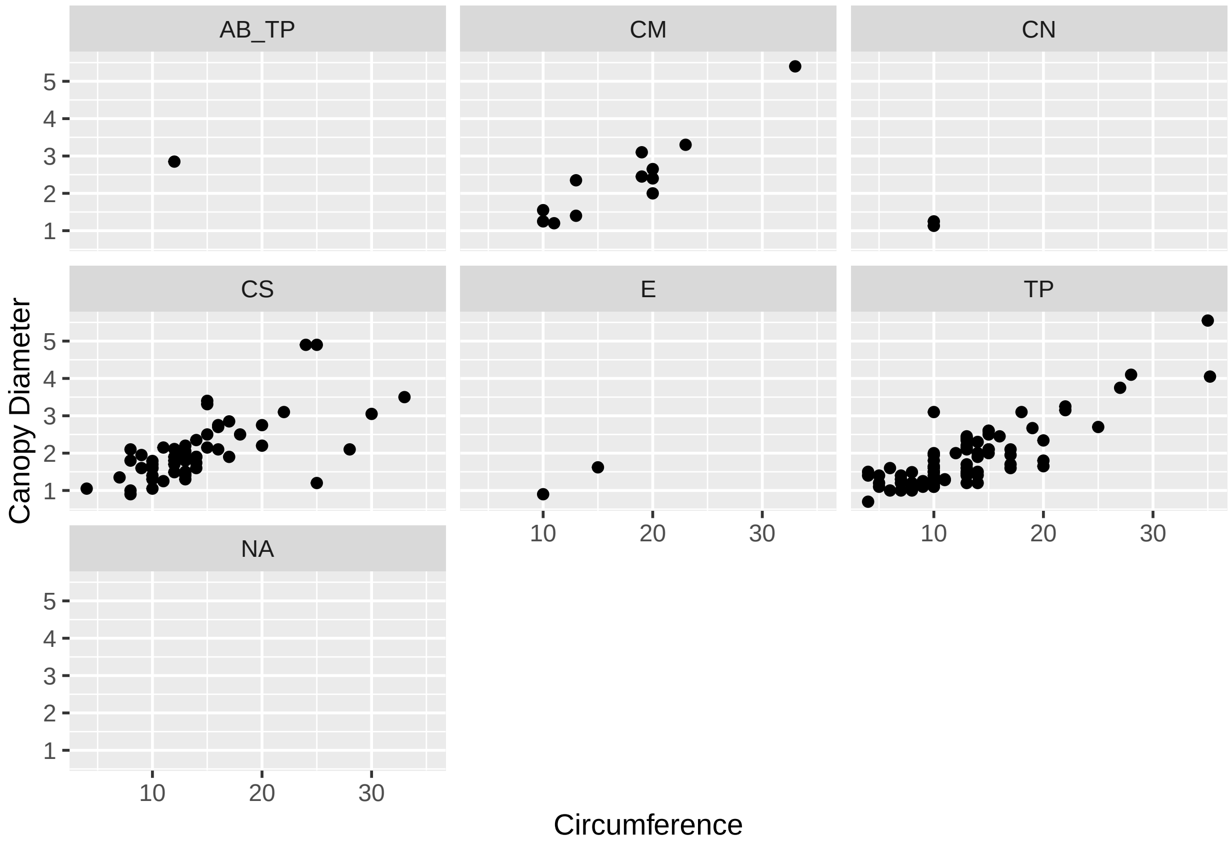
```
#ggsave("Graphing-acacia-ants-R-2.jpeg")

# 3

ggplot(data = acacia, mapping = aes(x = CIRC, y = AXIS1, color = ANT)) +
  geom_point() +
  labs(x = "Circumference", y = "Canopy Diameter")
```

```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



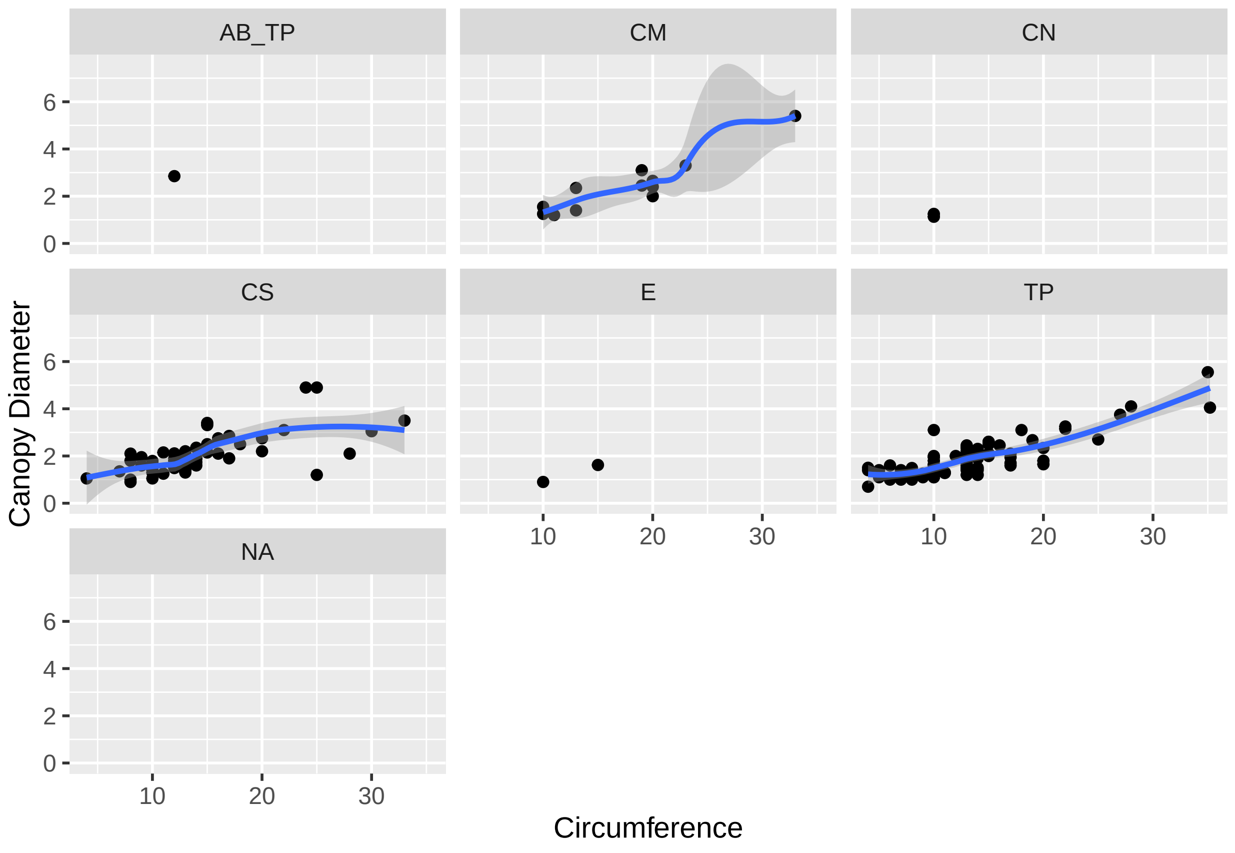
```
#ggsave("Graphing-acacia-ants-R-3.jpeg")

# 4

ggplot(data = acacia, mapping = aes(x = CIRC, y = AXIS1)) +
  geom_point() +
  labs(x = "Circumference", y = "Canopy Diameter") +
  facet_wrap(~ANT)
```

```
## Warning: Removed 4 rows containing missing values (‘geom_point()‘).
```



```r
#ggsave("Graphing-acacia-ants-R-4.jpeg")

# 5

ggplot(data = acacia, mapping = aes(x = CIRC, y = AXIS1)) +
  geom_point() +
  labs(x = "Circumference", y = "Canopy Diameter") +
  facet_wrap(~ANT) +
  geom_smooth()
```

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

## Warning: Removed 4 rows containing non-finite values (`stat_smooth()`).

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : span too small.  fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : at 9.975

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : radius 0.000625

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : all data on boundary of neighborhood. make span bigger
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 9.975

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 0.025

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : at 15.025

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : radius 0.000625

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : all data on boundary of neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 0.000625

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : zero-width neighborhood. make span bigger

## Warning: Computation failed in `stat_smooth()`
## Caused by error in `predLoess()`:
## ! NA/NaN/Inf in foreign function call (arg 5)

## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



```
#ggsave("Graphing-acacia-ants-R-5.jpeg")
```

2. Mass vs. Metabolism (20 pts)

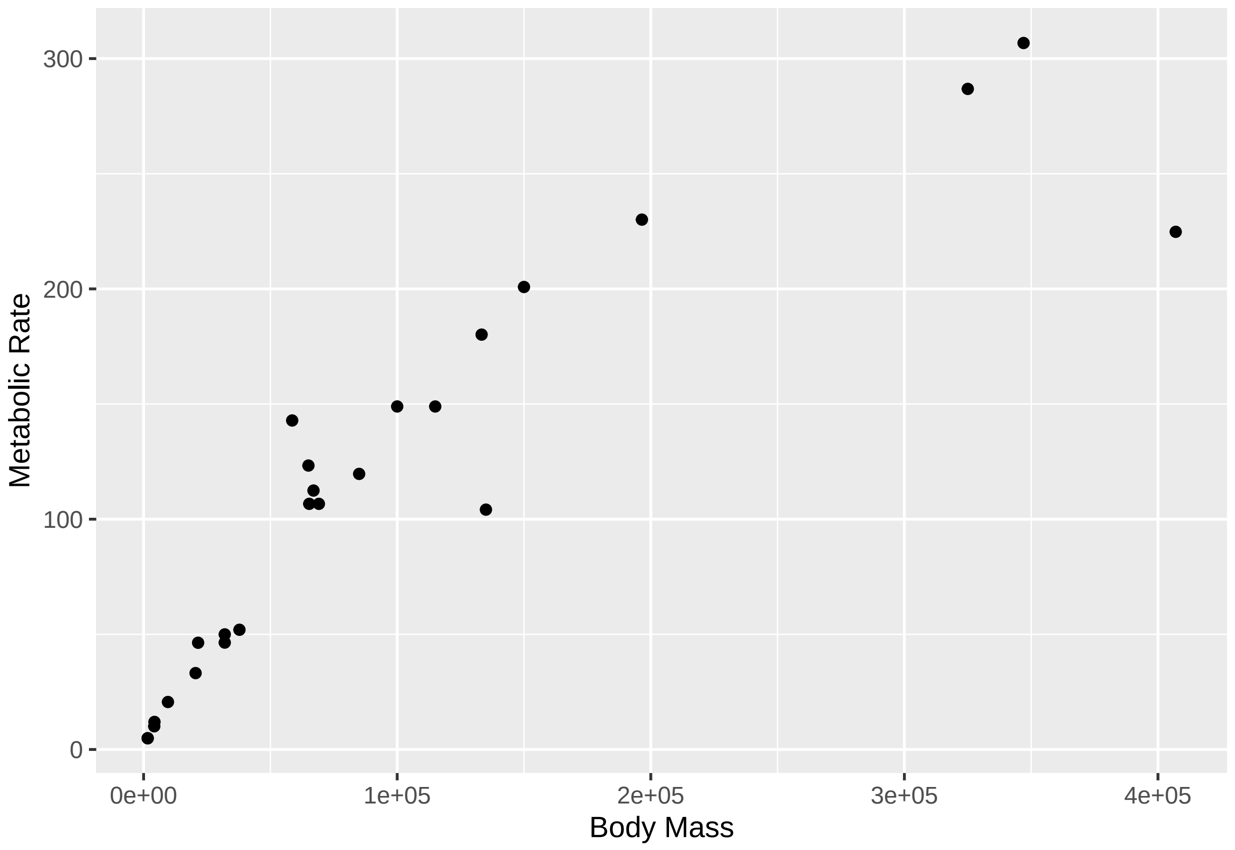
The relationship between the body size of an organism and its metabolic rate is one of the most well studied and still most controversial areas of organismal physiology. We want to graph this relationship in the Artiodactyla using a subset of data from a large compilation of body size data (Savage et al. 2004). You can copy and paste this data frame into your program:

```
size_mr_data <- data.frame(
  body_mass = c(32000, 37800, 347000, 4200, 196500, 100000,
    4290, 32000, 65000, 69125, 9600, 133300, 150000, 407000,
    115000, 67000,325000, 21500, 58588, 65320, 85000, 135000,
    20500, 1613, 1618),
  metabolic_rate = c(49.984, 51.981, 306.770, 10.075, 230.073,
    148.949, 11.966, 46.414, 123.287, 106.663, 20.619, 180.150,
    200.830, 224.779, 148.940, 112.430, 286.847, 46.347,
    142.863, 106.670, 119.660, 104.150, 33.165, 4.900, 4.865),
  family = c("Antilocapridae", "Antilocapridae", "Bovidae",
    "Bovidae", "Bovidae", "Bovidae", "Bovidae", "Bovidae",
    "Bovidae", "Bovidae", "Bovidae", "Bovidae", "Bovidae",
    "Camelidae", "Camelidae", "Canidae", "Cervidae",
    "Cervidae", "Cervidae", "Cervidae", "Cervidae", "Suidae",
    "Tayassuidae", "Tragulidae", "Tragulidae"))
```

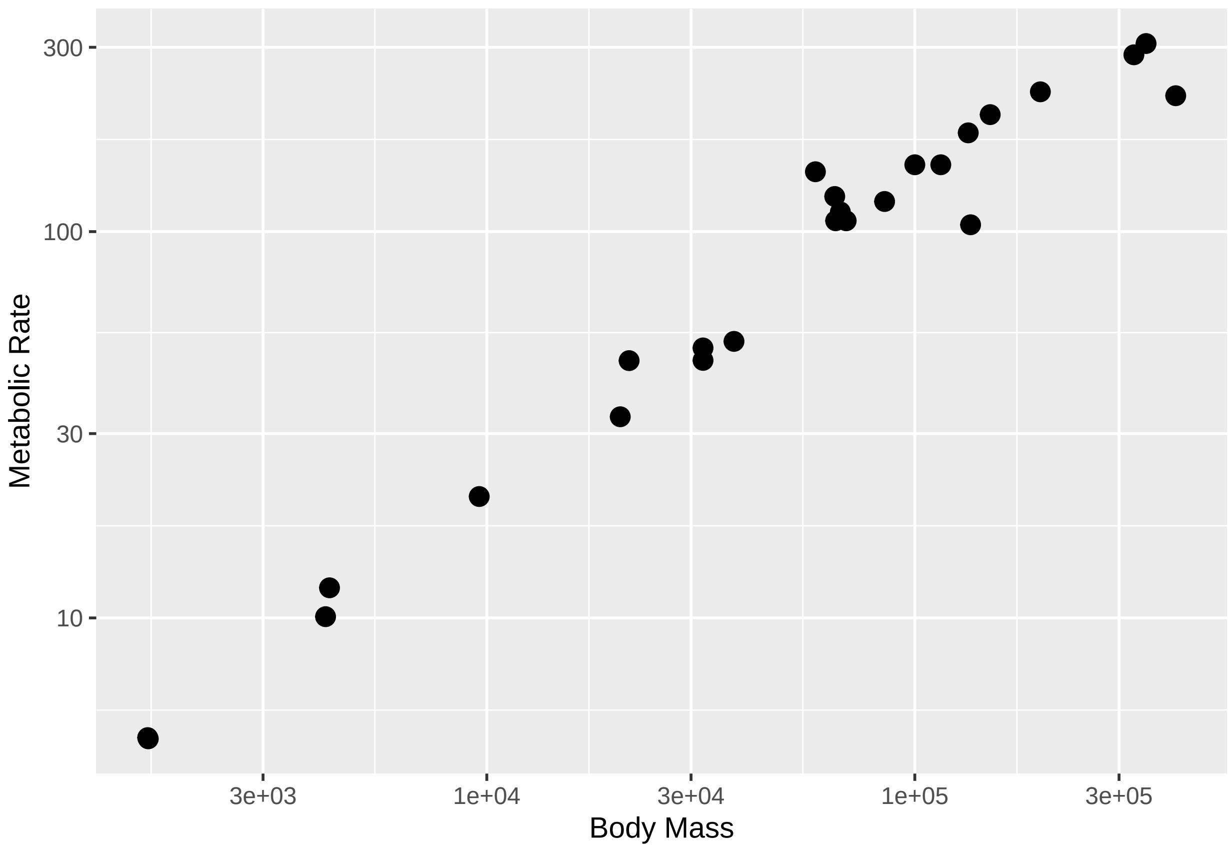
Make the following plots with appropriate axis labels:

1. A plot of body mass vs. metabolic rate
2. A plot of body mass vs. metabolic rate, with log10 scaled axes (this stretches the axis, but keeps the numbers on the original scale), and the point size set to 3.
3. The same plot as (2), but with the different families indicated using color.
4. The same plot as (2), but with the different families each in their own subplot.

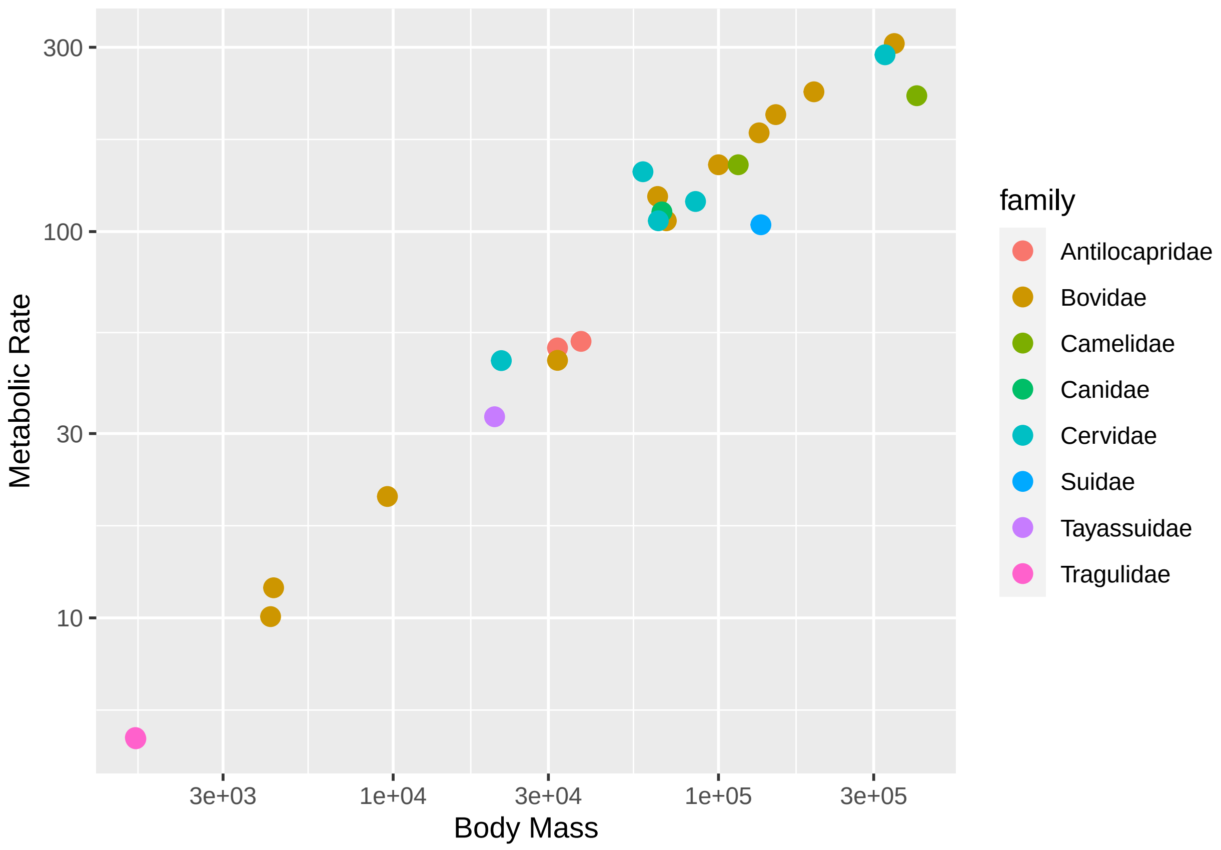
```
# 1.  A graph of body mass vs. metabolic rate
ggplot(size_mr_data, aes(body_mass, metabolic_rate)) + geom_point() +
  labs(x = "Body Mass", y = "Metabolic Rate")
```



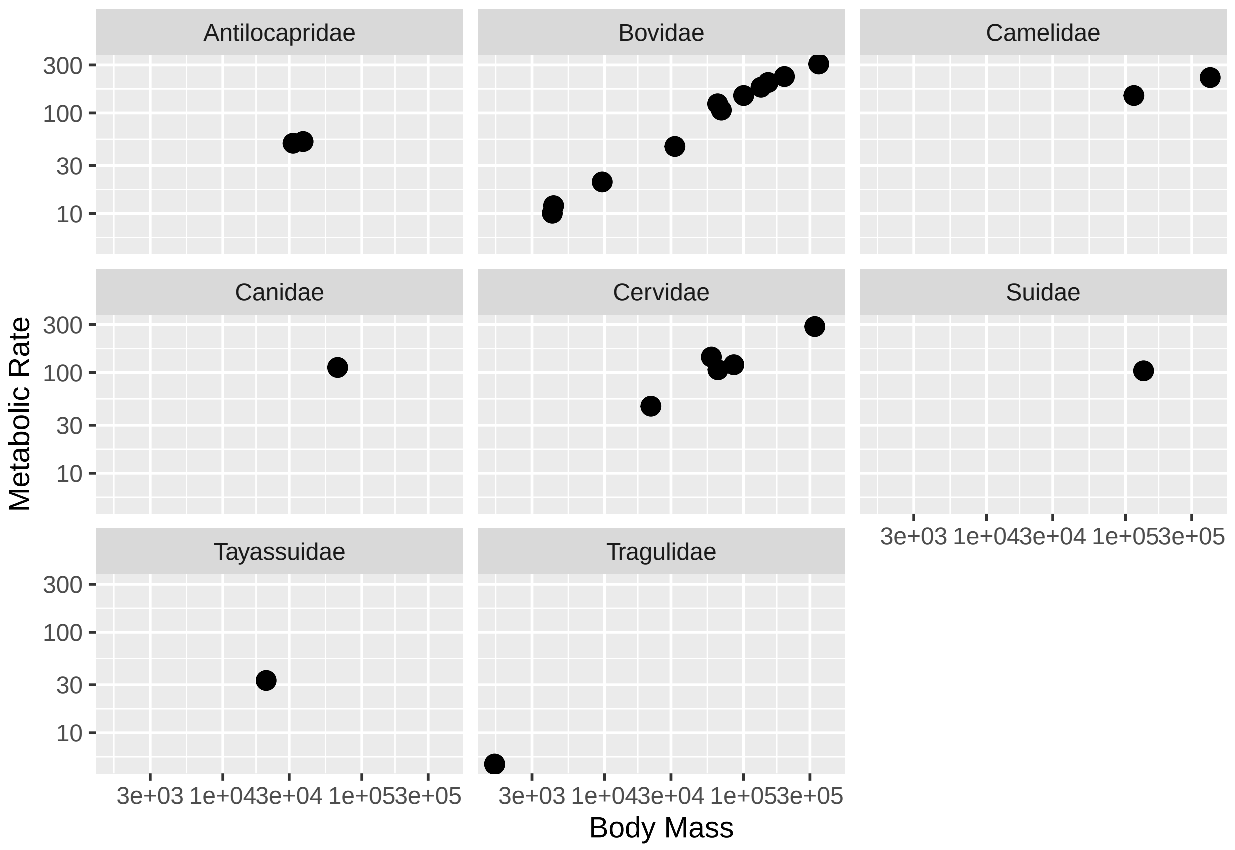
```
# 2.  A graph of body mass vs. metabolic rate, log scaled, with pt size 5.
ggplot(data = size_mr_data, mapping = aes(x = body_mass, y = metabolic_rate)) +
  geom_point(size=3) +
  labs(x = "Body Mass", y = "Metabolic Rate") +
  scale_x_log10() + scale_y_log10()
```



```r
# 3. The same plot as (2), but with the different families indicated using color.
ggplot(size_mr_data, aes(x = body_mass, y = metabolic_rate, color = family)) +
  geom_point(size = 3) +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Body Mass", y = "Metabolic Rate")
```



```
# 4. The same plot as (2), but with the different families each in their own subplot.
ggplot(size_mr_data, aes(x = body_mass, y = metabolic_rate)) +
  geom_point(size = 3) +
  scale_x_log10() +
  scale_y_log10() +
  facet_wrap(~family) +
  labs(x = "Body Mass", y = "Metabolic Rate")
```

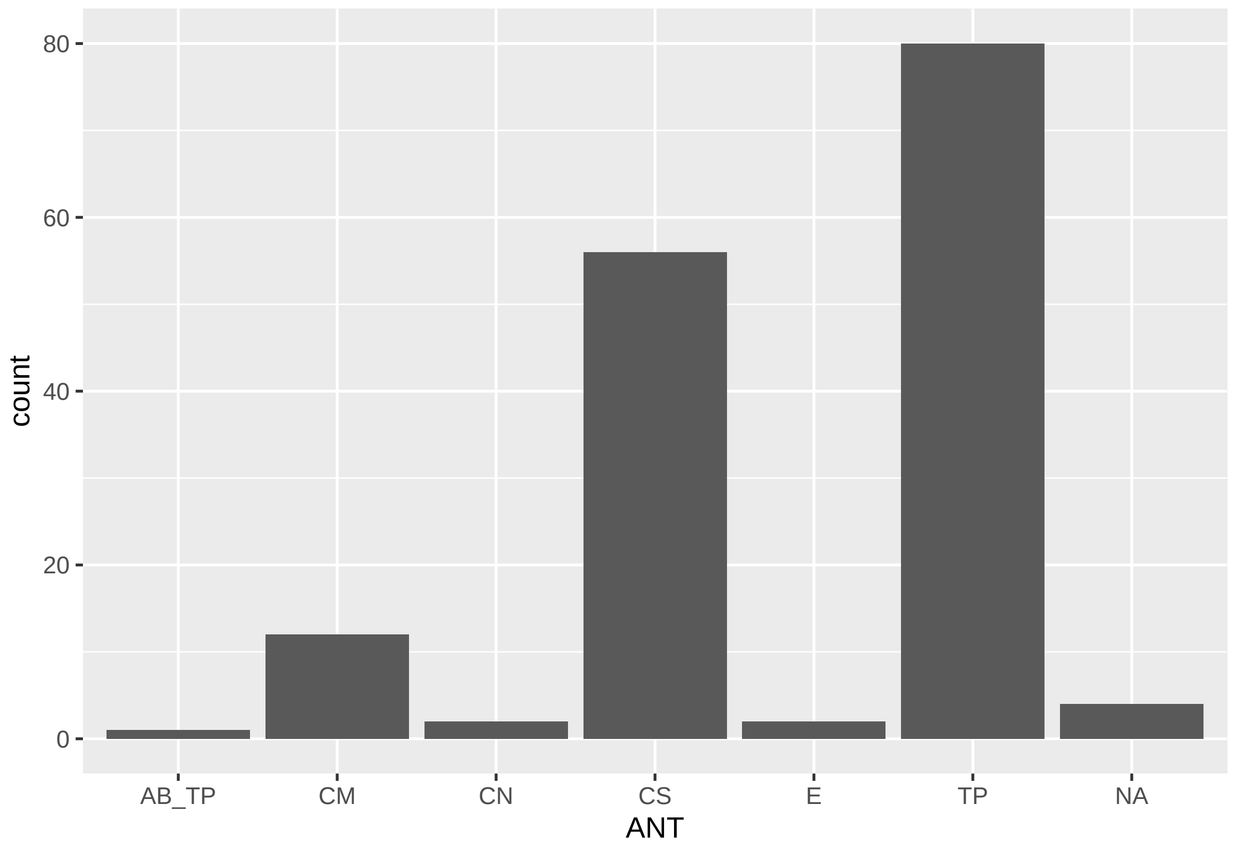


3. Acacia and Ants Histograms (20 pts)

In this exercise, we will be making a number of different histograms with the `acacia` dataset.

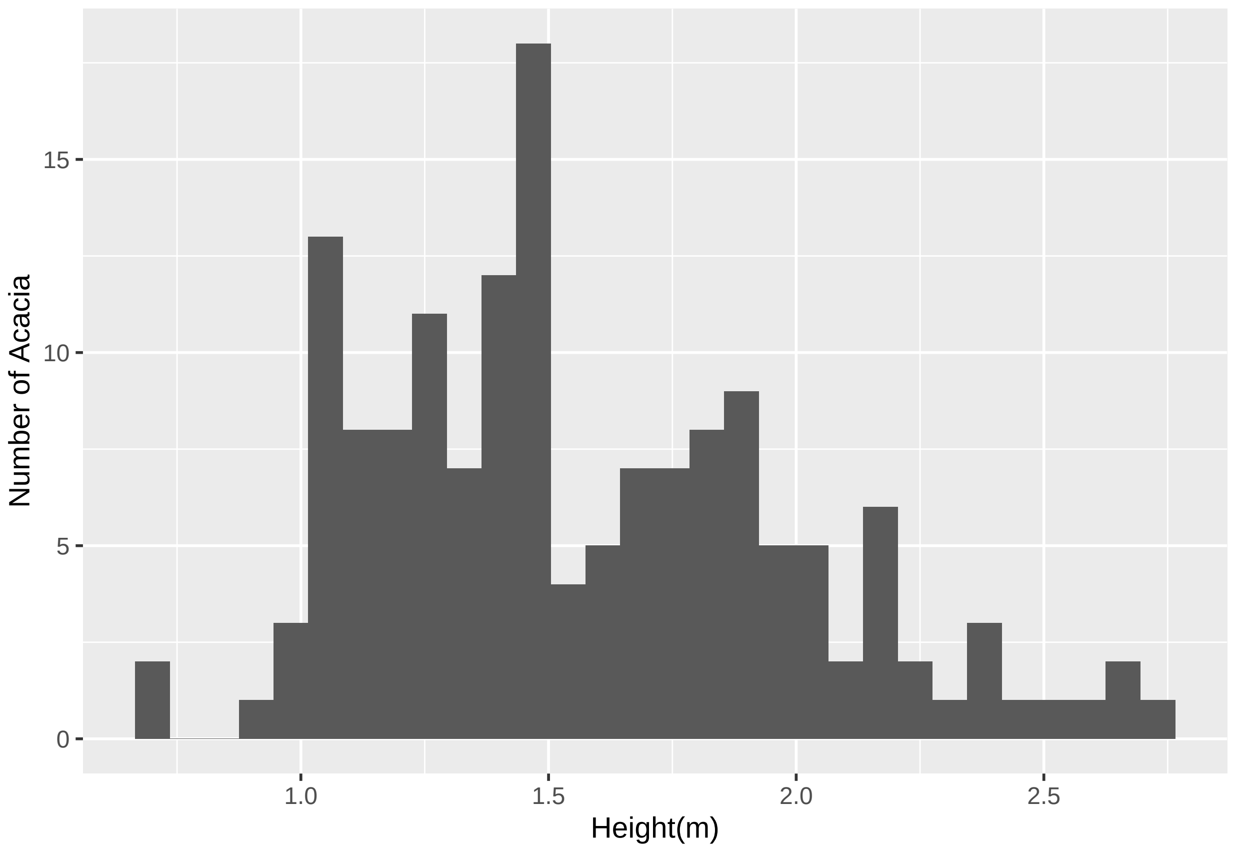
1. Make a bar plot of the number of acacia with each mutualist ant species (using the `ANT` column).
2. Make a histogram of the height of acacia (using the `HEIGHT` column). Label the x axis "Height (m)" and the y axis "Number of Acacia".
3. Make a plot that shows histograms of both `AXIS1` and `AXIS2`. Due to the way the data is structured you'll need to add a 2nd geom_histogram() layer that specifies a new aesthetic. To make it possible to see both sets of bars you'll need to make them transparent with the optional argument alpha = 0.3. Set the color for `AXIS1` to "red" and `AXIS2` to "black" using the `fill` argument. Label the x axis "Canopy Diameter(m)" and the y axis "Number of Acacia".
4. Use `facet_wrap()` to make the same plot as (3) but with one subplot for each treatment. Set the number of bins in the histogram to 10.

```
# 1

ggplot(data = acacia, mapping = aes(x = ANT)) +
  geom_bar()
```



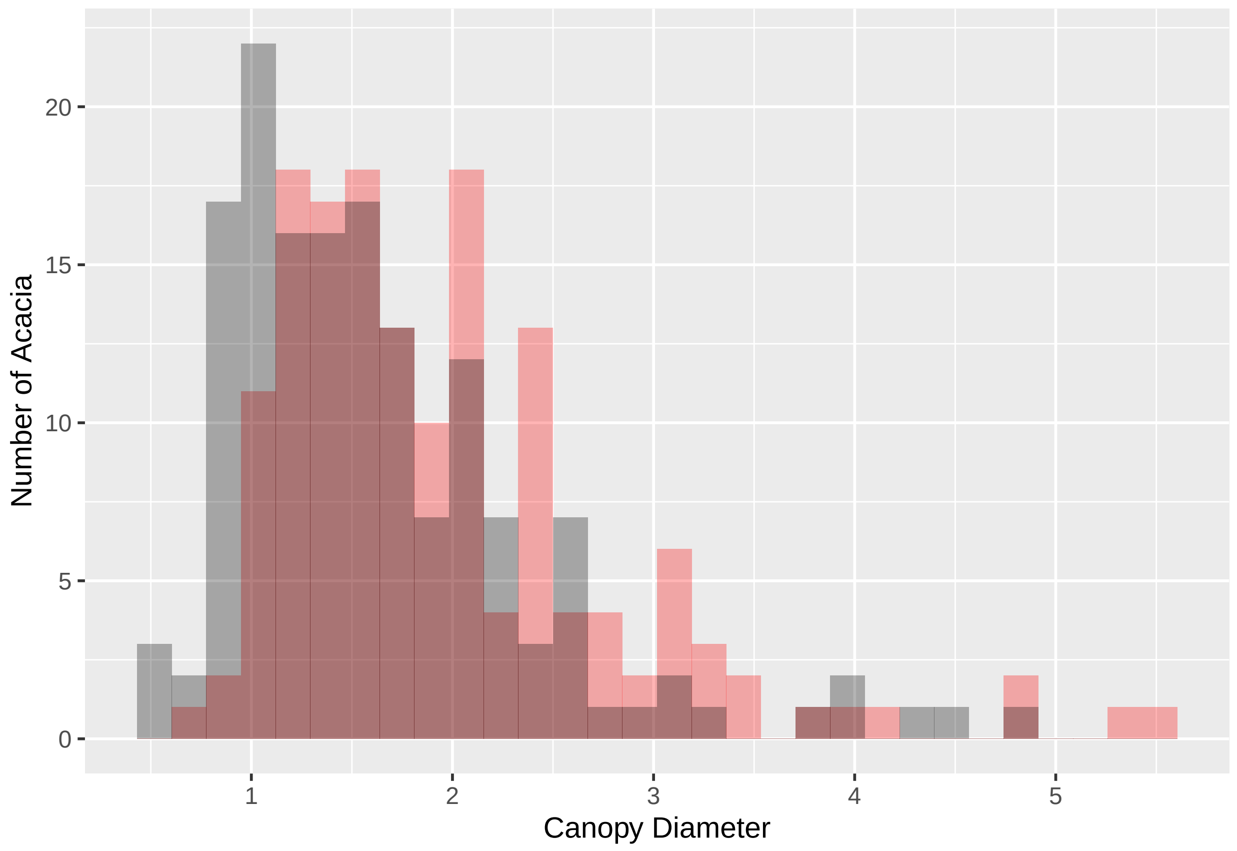
```r
#ggsave("Graphing-acacia-ants-histograms-R-1.jpeg")

# 2

ggplot(data = acacia, mapping = aes(x = HEIGHT)) +
  geom_histogram() +
  labs(x = "Height(m)", y = "Number of Acacia")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 4 rows containing non-finite values (`stat_bin()`).
```



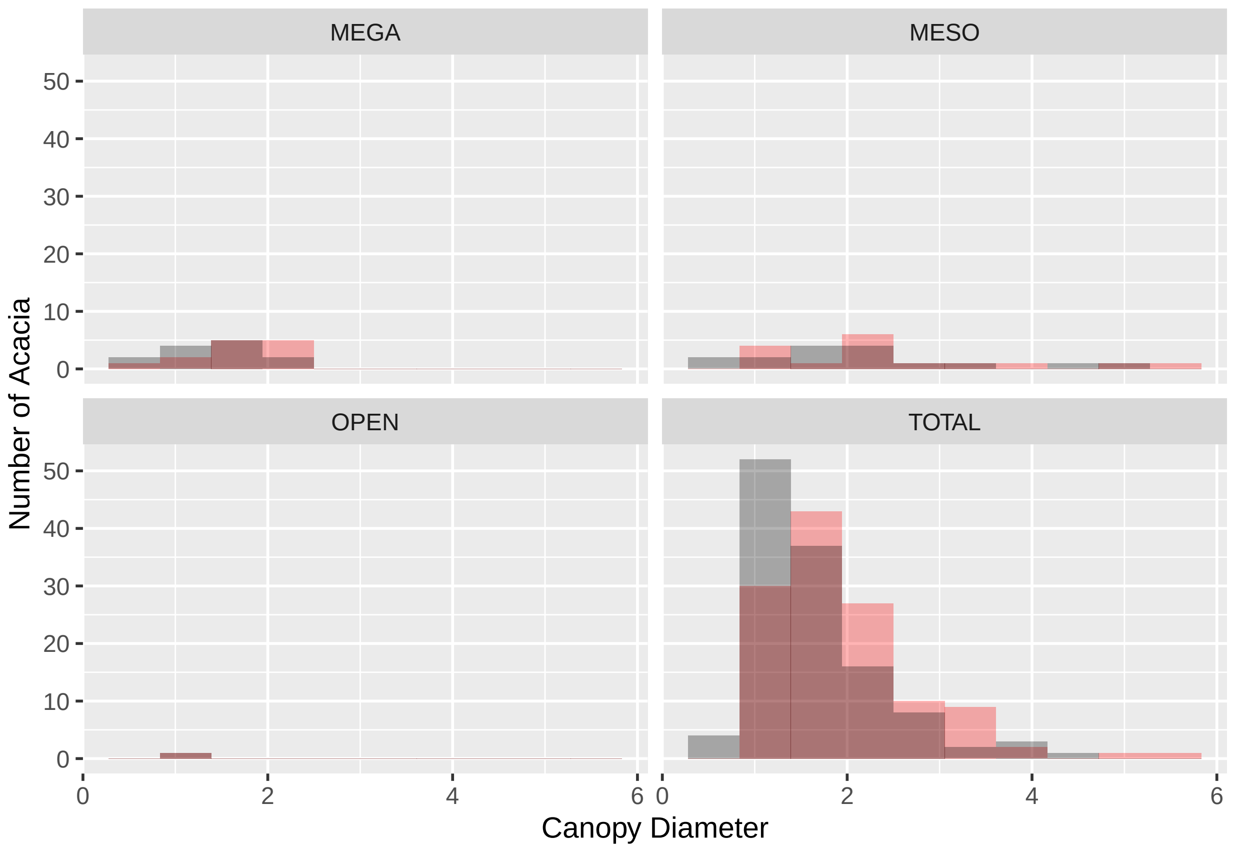
```r
#ggsave("Graphing-acacia-ants-histograms-R-2.jpeg")

# 3

ggplot(data = acacia) +
  geom_histogram(mapping = aes(x = AXIS1), fill = 'red', alpha = 0.3) +
  geom_histogram(mapping = aes(x = AXIS2), fill = 'black', alpha = 0.3) +
  labs(x = "Canopy Diameter", y = "Number of Acacia")
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.

## Warning: Removed 4 rows containing non-finite values (‘stat_bin()‘).

## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.

## Warning: Removed 4 rows containing non-finite values (‘stat_bin()‘).
```



```r
#ggsave("Graphing-acacia-ants-histograms-R-3.jpeg")

# 4

ggplot(data = acacia) +
  geom_histogram(mapping = aes(x = AXIS1), fill = 'red', alpha = 0.3, bins = 10) +
  geom_histogram(mapping = aes(x = AXIS2), fill = 'black', alpha = 0.3, bins = 10) +
  labs(x = "Canopy Diameter", y = "Number of Acacia") +
  facet_wrap(~TREATMENT)
```

```
## Warning: Removed 4 rows containing non-finite values (‘stat_bin()‘).
## Removed 4 rows containing non-finite values (‘stat_bin()‘).
```



```
#ggsave("Graphing-acacia-ants-histograms-R-4.jpeg")
```

4. Acacia and Ants Data Manipulation (20 pts)

Check to see if `TREE_SURVEYS.txt` is in your workspace. If not, download `TREE_SURVEYS.txt`. Use `read_tsv` from the `readr` package to read in the data using the following command:

```
trees <- read_tsv("TREE.txt",
                  col_types = list(HEIGHT = col_double(),
                                   AXIS_2 = col_double()))
```

```
## Warning: One or more parsing issues, call ‘problems()‘ on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

1. Update the `trees` data frame with a new column named `canopy_area` that contains the estimated canopy area calculated as the value in the `AXIS_1` column times the value in the `AXIS_2` column. Show output of the `trees` data frame with just the `SURVEY`, `YEAR`, `SITE`, and `canopy_area` columns.
2. Make a scatter plot with `canopy_area` on the x axis and `HEIGHT` on the y axis. Color the points by `TREATMENT` and plot the points for each value in the `SPECIES` column in a separate subplot. Label the x axis “Canopy Area (m)” and the y axis “Height (m)”. Make the point size 2.
3. That’s a big outlier in the plot from (2). 50 by 50 meters is a little too big for a real Acacia, so filter the data to remove any values for `AXIS_1` and `AXIS_2` that are over 20 and update the data frame. Then remake the graph.

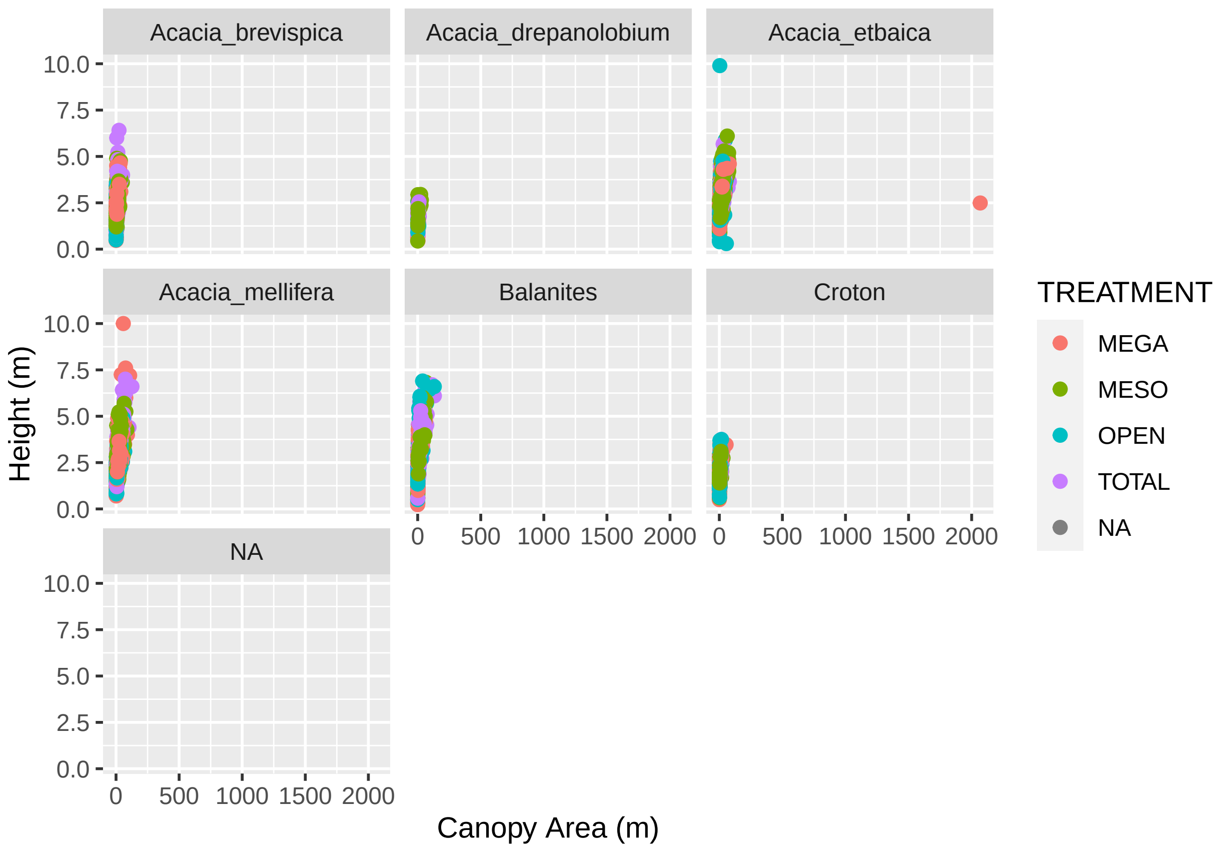
4. Using the data without the outlier (i.e., the data generated in (3)), find out how the abundance of each species has been changing through time. Use `group_by`, `summarize`, and `n` to make a data frame with `YEAR`, `SPECIES`, and an `abundance` column that has the number of individuals in each species in each year. Print out this data frame.
5. Using the data frame generated in (4), make a line plot with points (by using `geom_line` in addition to `geom_point`) with `YEAR` on the x axis and `abundance` on the y axis with one subplot per species. To let you seen each trend clearly let the scale for the y axis vary among plots by adding `scales = "free_y"` as an optional argument to `facet_wrap`.

```r
# 1

trees <- mutate(trees, canopy_area = AXIS_1 * AXIS_2)
select(trees, SURVEY, YEAR, SITE, canopy_area)
```

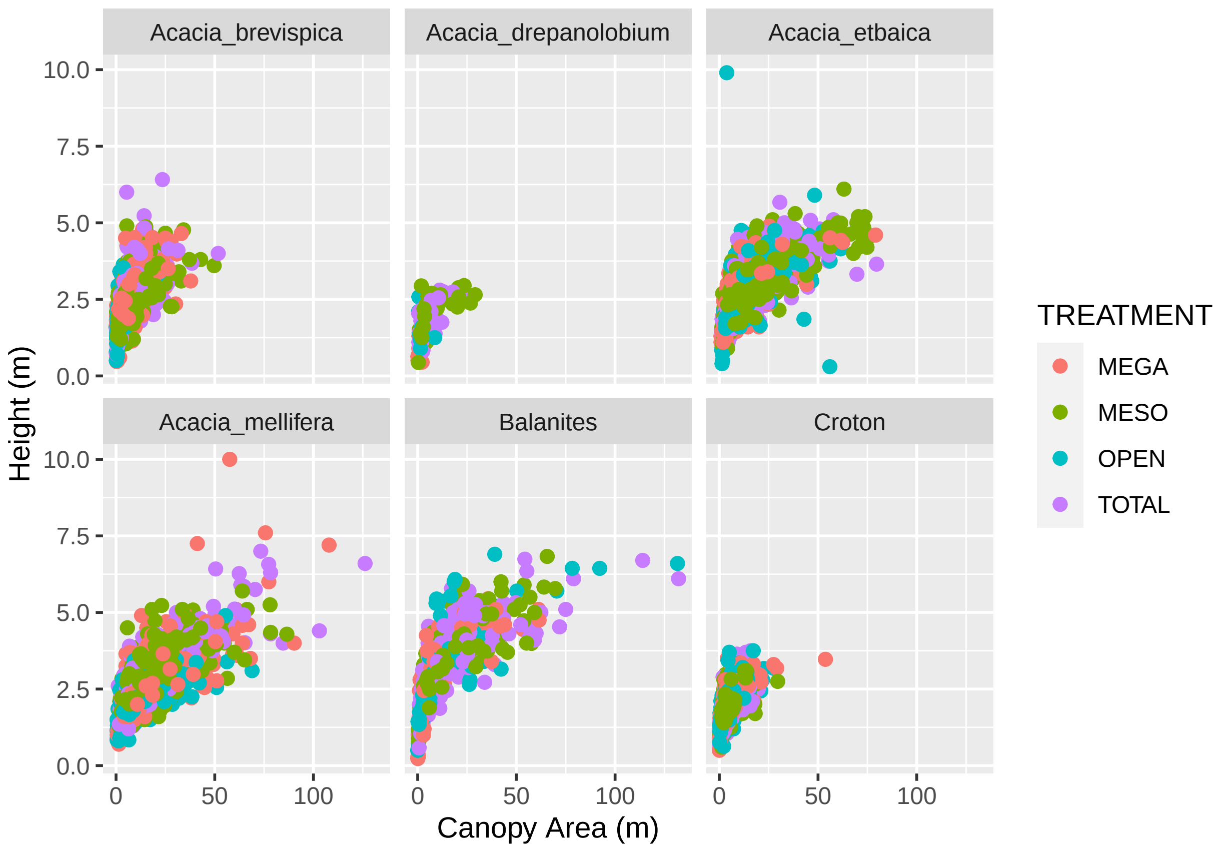
```
## # A tibble: 7,508 x 4
##    SURVEY  YEAR SITE  canopy_area
##     <dbl> <dbl> <chr>       <dbl>
##  1      1  2009 SOUTH       30.5
##  2      2  2010 SOUTH       69.7
##  3      3  2011 SOUTH       79.6
##  4      4  2012 SOUTH       39.0
##  5      5  2013 SOUTH       40.8
##  6      1  2009 SOUTH        6.16
##  7      2  2010 SOUTH        7.29
##  8      3  2011 SOUTH       12.5
##  9      4  2012 SOUTH       NA
## 10      5  2013 SOUTH        9.62
## # i 7,498 more rows
```

```r
# 2

ggplot(data = trees, mapping = aes(x = canopy_area, y = HEIGHT, color = TREATMENT)) +
  geom_point(size = 2) +
  labs(x = "Canopy Area (m)", y = "Height (m)") +
  facet_wrap(~SPECIES)
```

```
## Warning: Removed 215 rows containing missing values (‘geom_point()‘).
```



```
#ggsave("Graphing-acacia-ants-data-manip-R-2.jpeg")

# 3

trees <- filter(trees, AXIS_1 < 20, AXIS_2 < 20)
ggplot(data = trees, mapping = aes(x = canopy_area, y = HEIGHT, color = TREATMENT)) +
  geom_point(size = 2) +
  labs(x = "Canopy Area (m)", y = "Height (m)") +
  facet_wrap(~SPECIES)
```



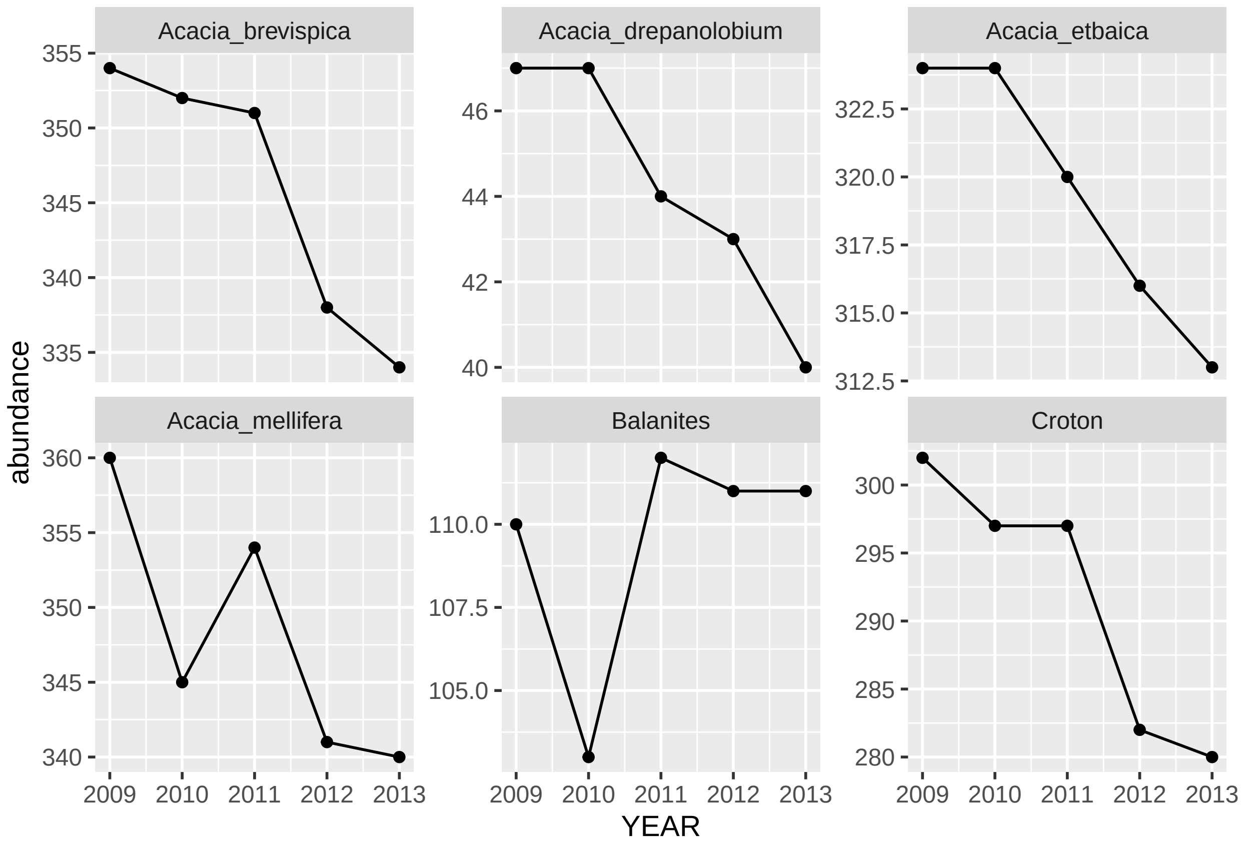
```r
#ggsave("Graphing-acacia-ants-data-manip-R-3.jpeg")

# 4

abundance_time <- trees %>%
  group_by(YEAR, SPECIES) %>%
  summarize(abundance = n())
```

```
## `summarise()` has grouped output by 'YEAR'. You can override using the
## `.groups` argument.
```

```r
abundance_time
```

```
## # A tibble: 30 x 3
## # Groups:   YEAR [5]
##     YEAR SPECIES              abundance
##    <dbl> <chr>                    <int>
##  1  2009 Acacia_brevispica          354
##  2  2009 Acacia_drepanolobium        47
##  3  2009 Acacia_etbaica             324
##  4  2009 Acacia_mellifera           360
##  5  2009 Balanites                  110
##  6  2009 Croton                     302
##  7  2010 Acacia_brevispica          352
##  8  2010 Acacia_drepanolobium        47
##  9  2010 Acacia_etbaica             324
## 10  2010 Acacia_mellifera           345
## # i 20 more rows
```

```
# 5

ggplot(data = abundance_time, mapping = aes(x = YEAR, y = abundance)) +
  geom_point() +
  geom_line() +
  facet_wrap(~SPECIES, scales = "free_y")
```



```
#ggsave("Graphing-acacia-ants-data-manip-R-5.jpeg")
```

5. Graphing Data from Multiple Tables (optional)

We want to compare the circumference to height relationship in acacia and to the same relationship for trees in the region. These data are stored in two different tables. Make a graph with the relationship between CIRC and HEIGHT for the trees as gray circles in the background and the same relationship for acacia as red circles plotted on top of the graph circles. Scale the both axes logarithmically. Inlude linear models for both sets of data. Provide clear labels for the axes.

6. Adult vs. Newborn Size (20 pts)

Larger organisms have larger offspring. We want to explore the form of this relationship in mammals.

```r
mammal_histories <- read_tsv("Mammal_lifehistories_v2.txt",
                             na = c("-999", "-999.00")) %>%
  rename(mass_g = `mass(g)`,
         gestation_mo = `gestation(mo)`,
         newborn_g = `newborn(g)`,
         weaning_mo = `weaning(mo)`,
         wean_mass_g = `wean mass(g)`,
         AFR_mo = `AFR(mo)`,
         max_life_mo = `max. life(mo)`,
         litter_size = `litter size`,
         litters_per_year = `litters/year`)
```

```
## Rows: 1440 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (4): order, family, Genus, species
## dbl (9): mass(g), gestation(mo), newborn(g), weaning(mo), wean mass(g), AFR(...
## num (1): refs
##
## 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.
```

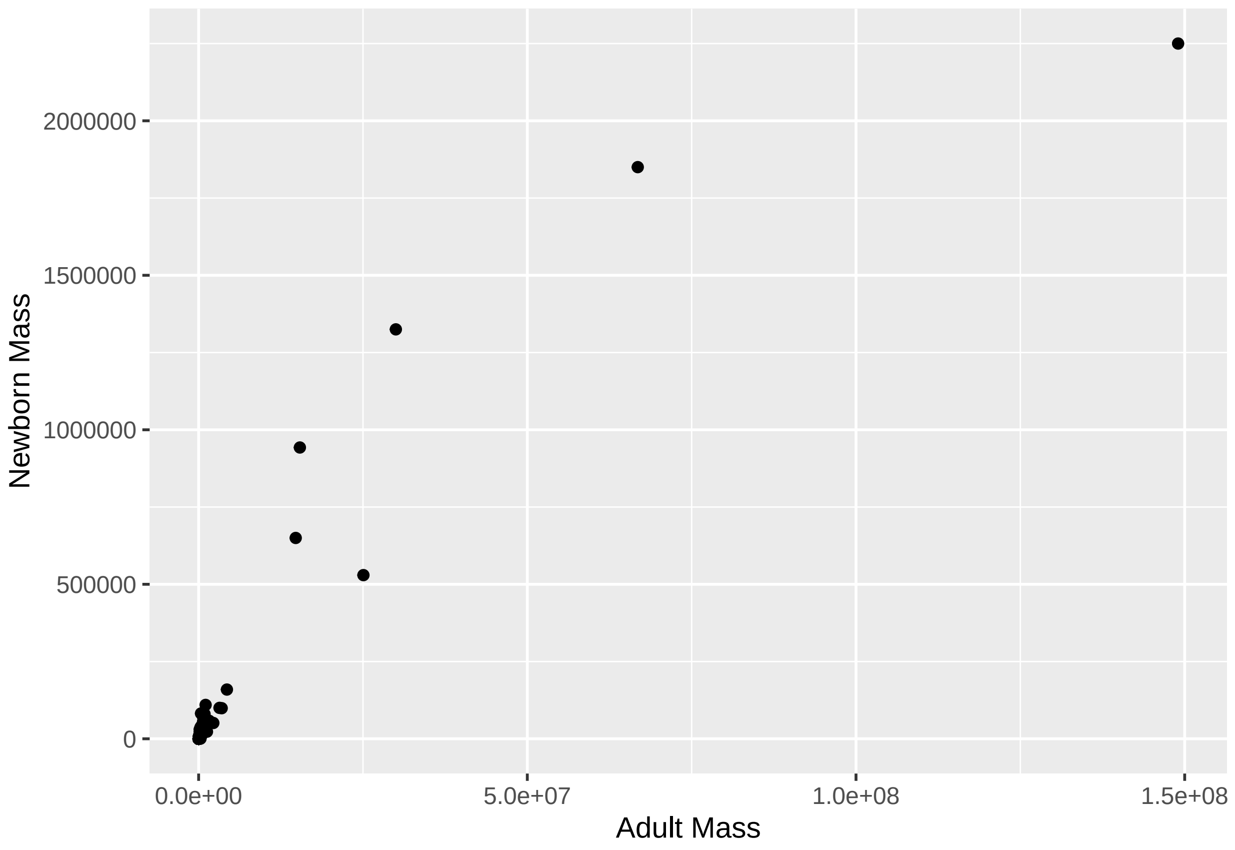
a. Write 2-3 sentences explaining what the code above is doing.

Answer:

b. Graph adult mass vs. newborn mass. Label the axes with clearer labels than the column names.
c. It looks like there's a regular pattern here, but it's definitely not linear. Let's see if log-transformation straightens it out. Graph adult mass vs. newborn mass, with both axes scaled logarithmically. Label the axes.
d. This looks like a pretty regular pattern, so you wonder if it varies among different groups. Graph adult mass vs. newborn mass, with both axes scaled logarithmically, and the data points colored by order. Label the axes.
e. Coloring the points was useful, but there are a lot of points and it's kind of hard to see what's going on with all of the orders. Use `facet_wrap` to create a subplot for each order.
f. Now let's visualize the relationships between the variables using a simple linear model. Create a new graph like your faceted plot, but using `geom_smooth` to fit a linear model to each order. You can do this using the optional argument `method = "lm"` in `geom_smooth`.

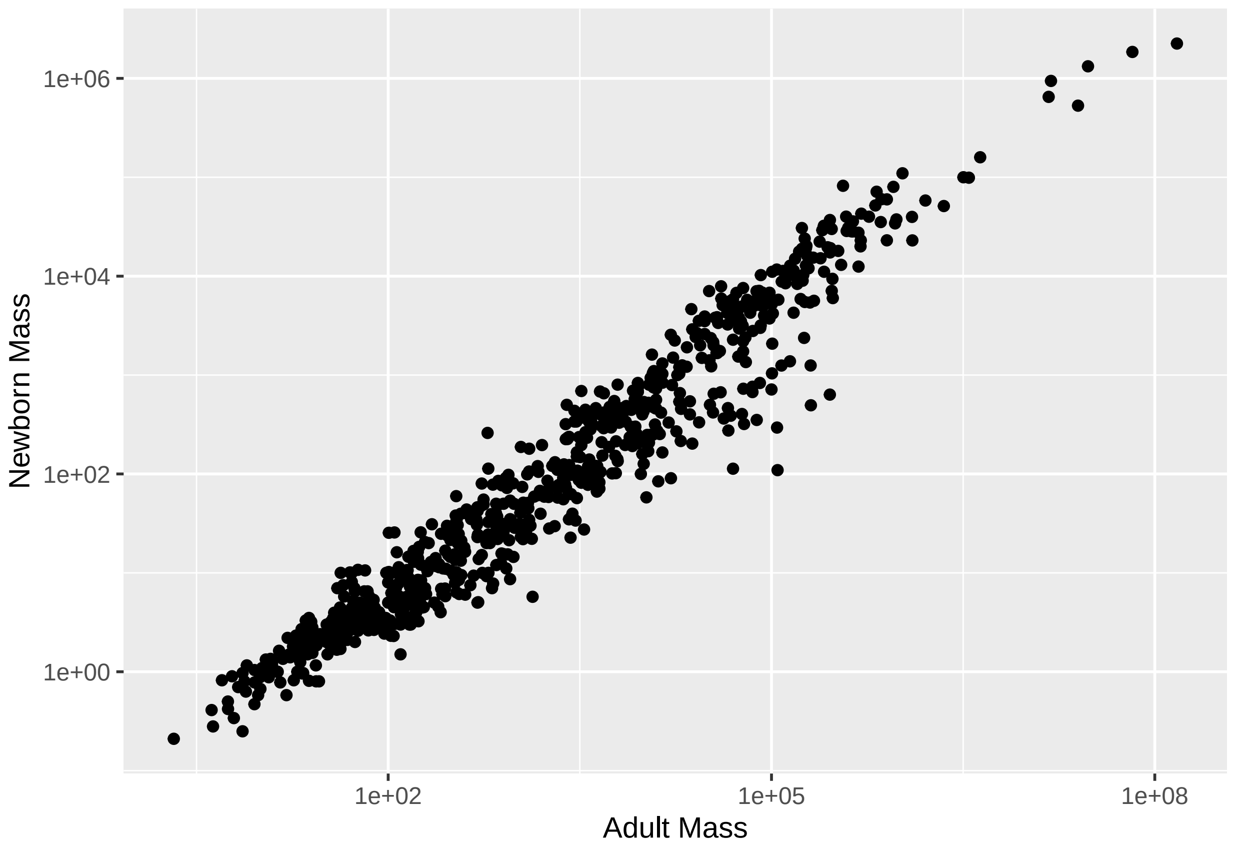
```r
# b. Graph adult mass vs. newborn mass.
ggplot(mammal_histories, aes(x = mass_g, y = newborn_g)) +
  geom_point() +
  labs(x = "Adult Mass", y = "Newborn Mass")
```

```
## Warning: Removed 624 rows containing missing values (`geom_point()`).
```



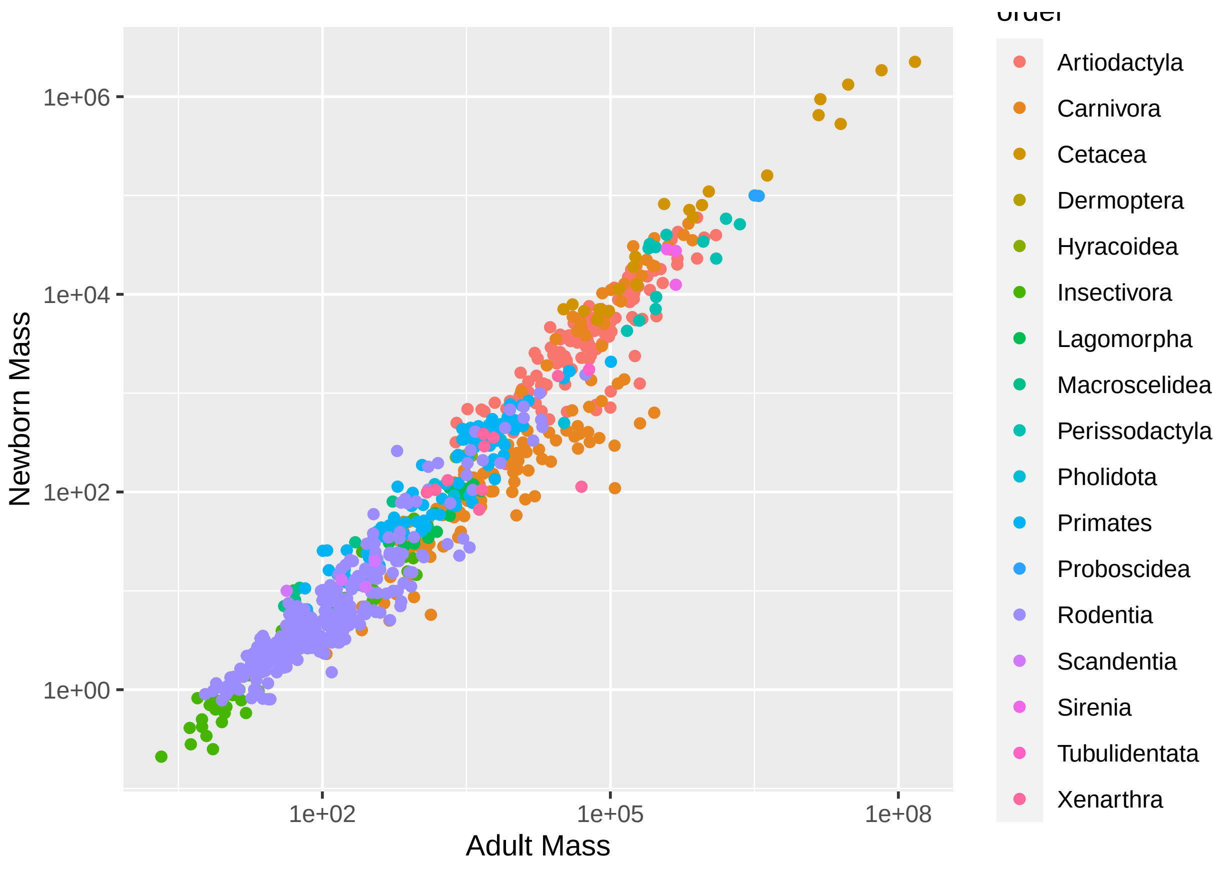
```r
# c. Graph adult mass vs. newborn mass, with both axes scaled logarithmically.
ggplot(mammal_histories, aes(x = mass_g, y = newborn_g)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Adult Mass", y = "Newborn Mass")
```

```
## Warning: Removed 624 rows containing missing values (‘geom_point()‘).
```



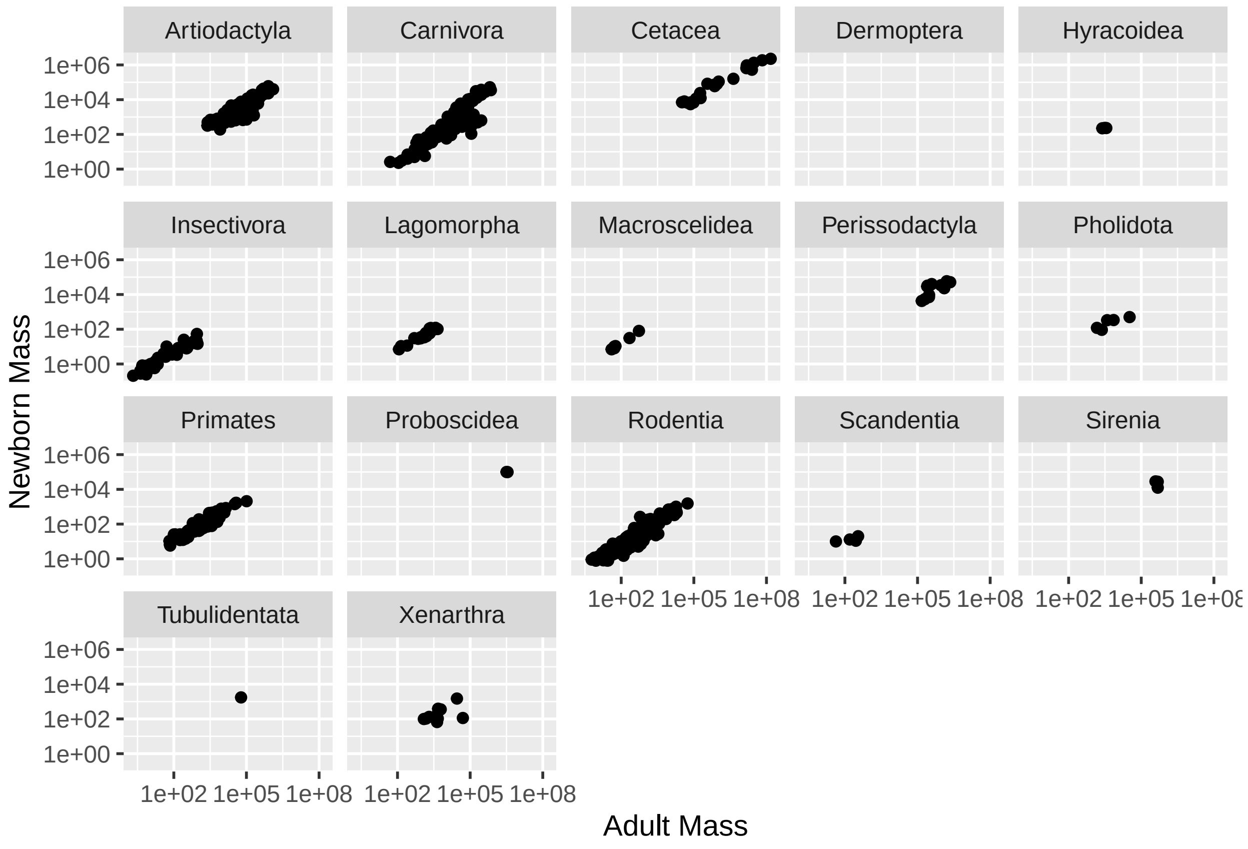
```r
# d. Graph adult mass vs. newborn mass, log-scaled, with data colored by order.
ggplot(mammal_histories, aes(x = mass_g, y = newborn_g)) +
  geom_point(aes(color=order)) +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Adult Mass", y = "Newborn Mass")
```

```
## Warning: Removed 624 rows containing missing values (‘geom_point()‘).
```



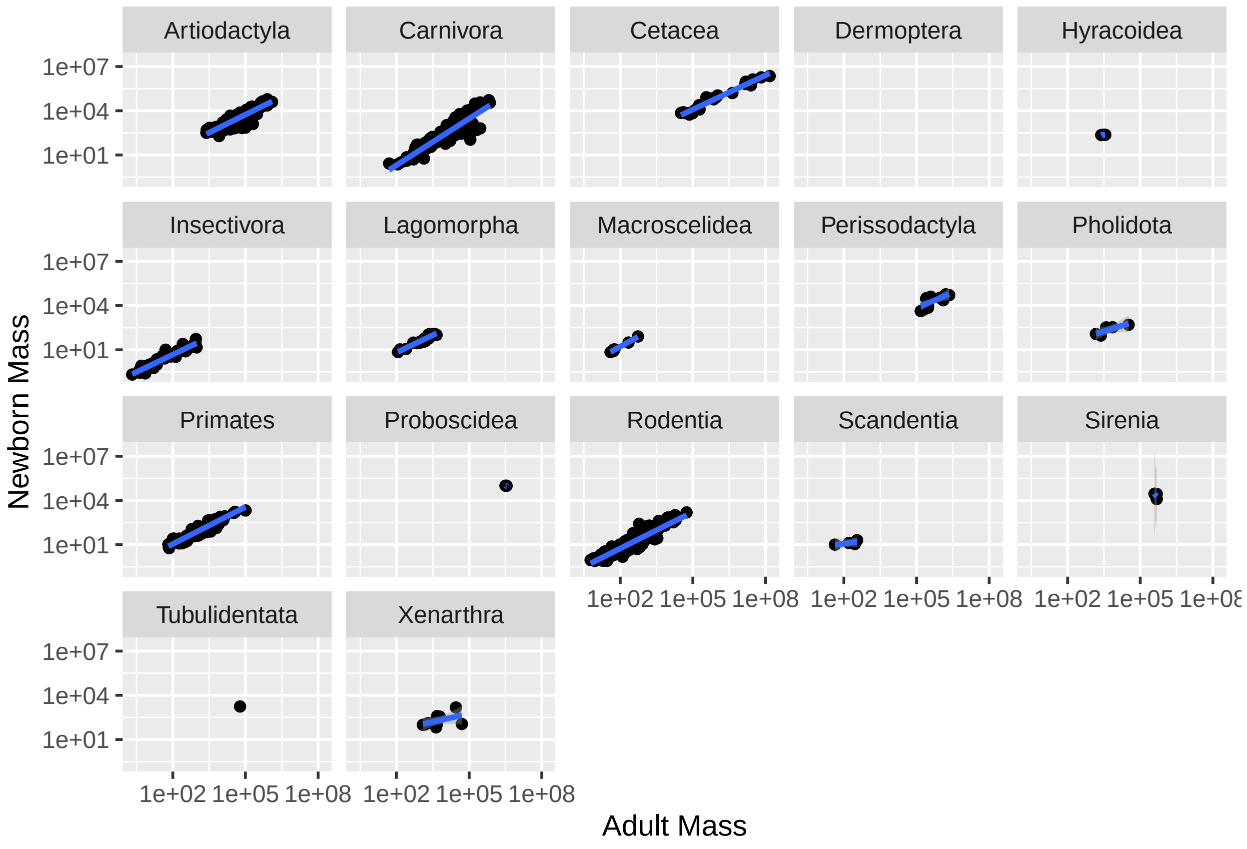
```r
# e. Use `facet_wrap` to create subplot for each order.
ggplot(mammal_histories, aes(x = mass_g, y = newborn_g)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10() +
  facet_wrap(~ order) +
  labs(x = "Adult Mass", y = "Newborn Mass")
```

```
## Warning: Removed 624 rows containing missing values (‘geom_point()‘).
```



```r
# f. use `geom_smooth` to fit a linear model to each order.
ggplot(mammal_histories, aes(x = mass_g, y = newborn_g)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_x_log10() +
  scale_y_log10() +
  facet_wrap(~ order) +
  labs(x = "Adult Mass", y = "Newborn Mass")
```

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

## Warning: Removed 624 rows containing non-finite values (‘stat_smooth()‘).

## Warning in qt((1 - level)/2, df): NaNs produced

## Warning: Removed 624 rows containing missing values (‘geom_point()‘).

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```



Artiodactyla
Carnivora
Cetacea
Dermoptera
Hyracoidea
Insectivora
Lagomorpha
Macroscelidea
Perissodactyla
Pholidota
Primates
Proboscidea
Rodentia
Scandentia
Sirenia
Tubulidentata
Xenarthra
1e+07
1e+04
1e+01
1e+02
1e+05
1e+08
Newborn Mass
Adult Mass