Hoops Longwing Sample Data Visualization

Zane Billings

15 November 2019

In order to start analyzing the Hoops' Longwing sample data, we will first load the `tidyverse` package suite. After loading the packages we need, we can use `readr::read_csv()` to load in the data. But, notice the imported data frame has a useless column at the beginning, which we can easily remove manually.

```r
library(tidyverse)

butterfly <- read_csv("hoops_longwing_study.csv")
butterfly <- butterfly[ , -1]
```

Now that we have the data imported, we can go ahead and take a quick look at the summary and structure.

```r
summary(butterfly)
```

```
##   wing_length      wing_width          age        num_offspring
##  Min.   : 8.87   Min.   : 3.730   Min.   : 1.00   Min.   : 4.00
##  1st Qu.:14.45   1st Qu.: 6.750   1st Qu.:12.00   1st Qu.:24.00
##  Median :17.09   Median : 8.180   Median :19.00   Median :28.00
##  Mean   :20.32   Mean   : 8.727   Mean   :22.42   Mean   :27.78
##  3rd Qu.:26.91   3rd Qu.:10.832   3rd Qu.:31.00   3rd Qu.:32.00
##  Max.   :37.59   Max.   :15.330   Max.   :56.00   Max.   :38.00
##  feeding_range      color_peak      num_mates      avg_scale_size
##  Min.   : 0.790   Min.   :363.5   Min.   : 0.000   Min.   :19.90
##  1st Qu.: 2.620   1st Qu.:385.8   1st Qu.: 3.000   1st Qu.:28.11
##  Median : 3.490   Median :392.0   Median : 5.000   Median :33.03
##  Mean   : 5.856   Mean   :392.3   Mean   : 6.149   Mean   :38.36
##  3rd Qu.: 5.685   3rd Qu.:398.4   3rd Qu.: 9.000   3rd Qu.:48.82
##  Max.   :44.440   Max.   :423.6   Max.   :18.000   Max.   :76.10
##  antenna_length    num_spots       population        dispersal_distance
##  Min.   :1.140   Min.   : 3.000   Length:1000        Min.   :21.98
##  1st Qu.:3.210   1st Qu.: 4.000   Class :character   1st Qu.:24.10
##  Median :3.940   Median : 6.000   Mode  :character   Median :24.69
##  Mean   :4.406   Mean   : 5.737                      Mean   :24.70
##  3rd Qu.:5.812   3rd Qu.: 7.000                      3rd Qu.:25.27
##  Max.   :7.220   Max.   :13.000                      Max.   :27.90
##   body_length      sample_id
##  Min.   : 1.890   Length:1000
##  1st Qu.: 5.055   Class :character
##  Median : 6.390   Mode  :character
##  Mean   : 6.784
##  3rd Qu.: 8.543
##  Max.   :12.510
```

```r
str(butterfly)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1000 obs. of  14 variables:
##  $ wing_length       : num  28.2 20.4 12.8 16.2 15.5 ...
##  $ wing_width        : num  10.48 11.62 5.72 7.99 6.55 ...
##  $ age               : num  28 37 10 52 44 49 22 9 28 9 ...
##  $ num_offspring     : num  28 33 20 36 37 36 29 22 32 22 ...
##  $ feeding_range     : num  4.32 10.13 1.42 30.25 14.7 ...
##  $ color_peak        : num  408 416 391 382 391 ...
##  $ num_mates         : num  10 5 2 5 4 5 3 2 7 12 ...
##  $ avg_scale_size    : num  52.7 37.3 26.4 27.5 31.5 ...
##  $ antenna_length    : num  6.14 4.68 2.7 3.7 3.44 3.7 3.41 2.09 5.33 6.36 ...
##  $ num_spots         : num  4 4 8 6 7 6 6 12 5 3 ...
##  $ population        : chr  "Tidore" "Kayoa" "Ternate" "Ternate" ...
##  $ dispersal_distance: num  25.7 26.7 24.8 23.5 24.9 ...
##  $ body_length       : num  9.74 8.95 4.93 6.55 8.84 6.01 4.18 5.25 5.76 9.18 ...
##  $ sample_id         : chr  "Tid_0001_ZW" "Kay_0002_EM" "Ter_0003_ZW" "Ter_0004_EM" ...
```

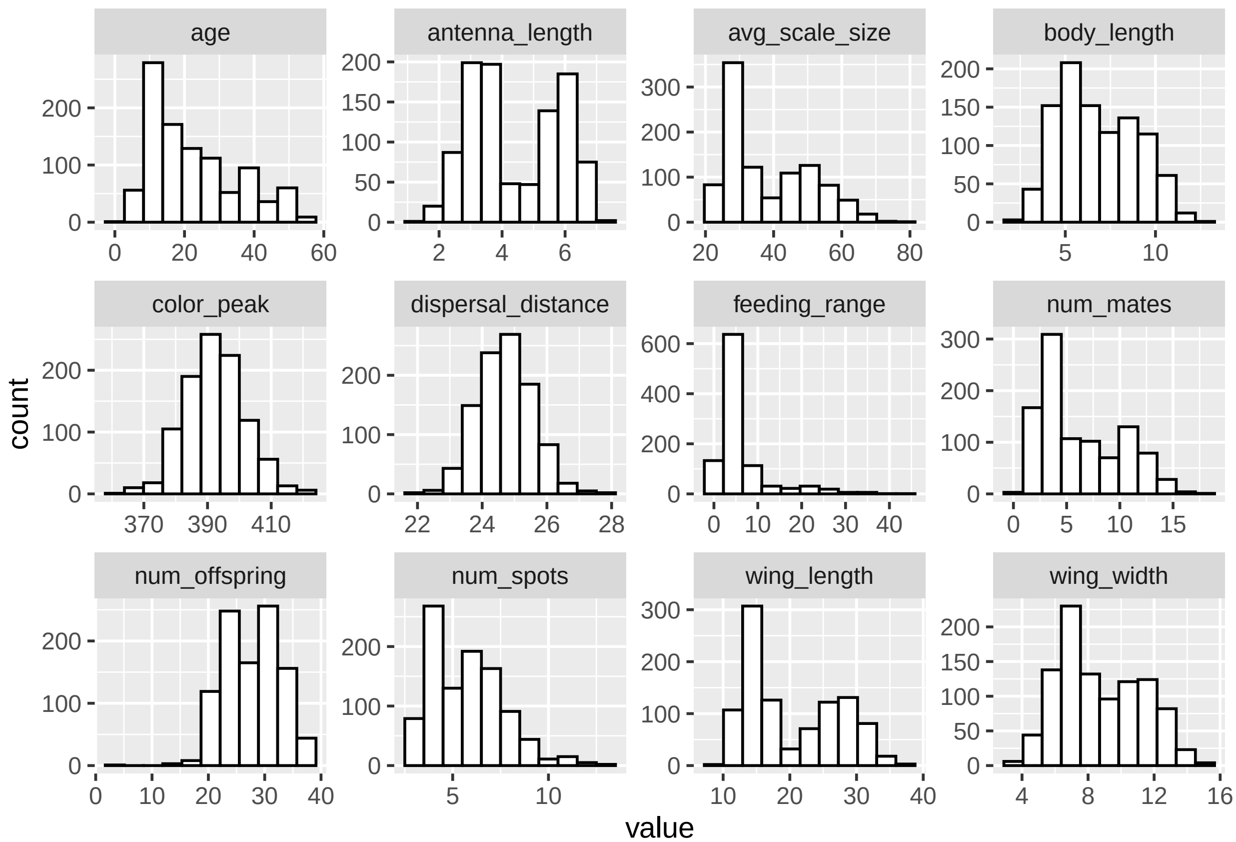
The only real change we need to make is to convert the `population` variable into a factor, since the functions provided in `readr` do not coerce strings to factors by default.

```r
butterfly$population <- as.factor(butterfly$population)
summary(butterfly$population)
```

```
##   Kayoa Ternate  Tidore
##     125     540     335
```

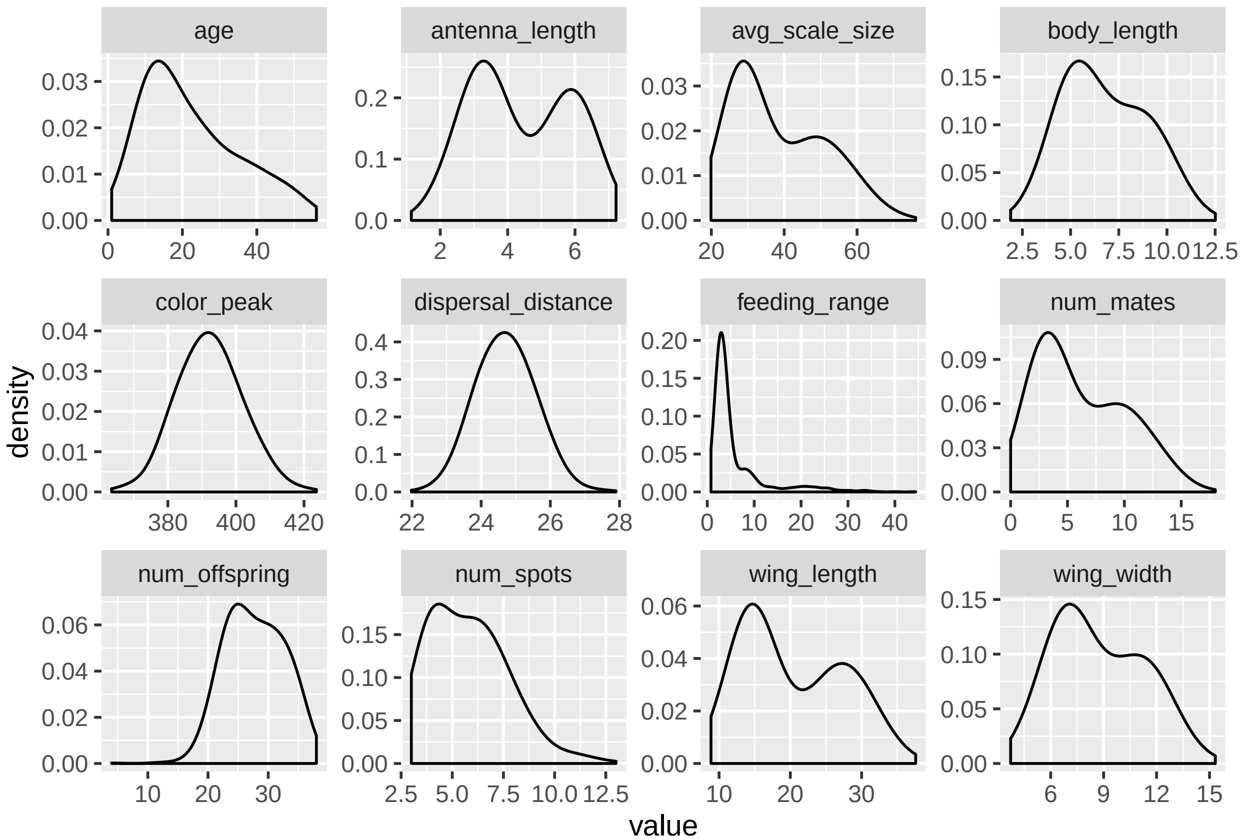
So, now we can start exploring our data. Let's start by making histograms of all the numeric data.

```
# Sturges Rule calculation for number of histogram bins.
num_bins <- ceiling(log2(nrow(butterfly))) + 1
butterfly %>%
  select(-population) %>%
  gather(key = "field", value = "value", -"sample_id") %>%
  ggplot(aes(x = value)) +
  geom_histogram(col = "black",
                 fill = "white",
                 position = "identity",
                 bins = num_bins) +
  facet_wrap(~field, scales = "free")
```



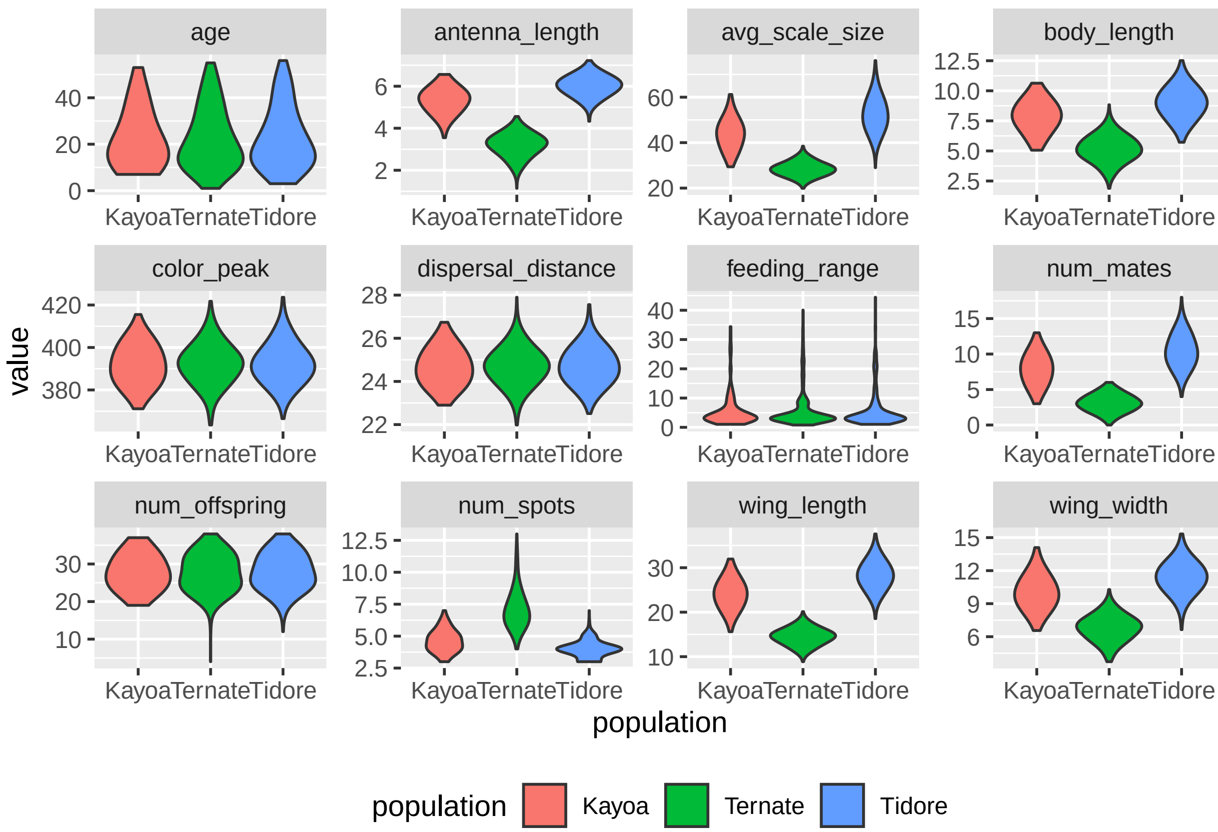
We can also visualize the distributions of the data using density curves, although we have to adjust the interpolation settings because part of our data is integers.

```
# An alternative way to view the data
butterfly %>%
  select(-population) %>%
  gather(key = "field", value = "value", -"sample_id") %>%
  ggplot(aes(x = value)) +
  geom_density(adjust = 2) +
  facet_wrap(~field, scales = "free")
```



Now we have one categorical variable, so let's look at all of our data stratified by the `population` value.

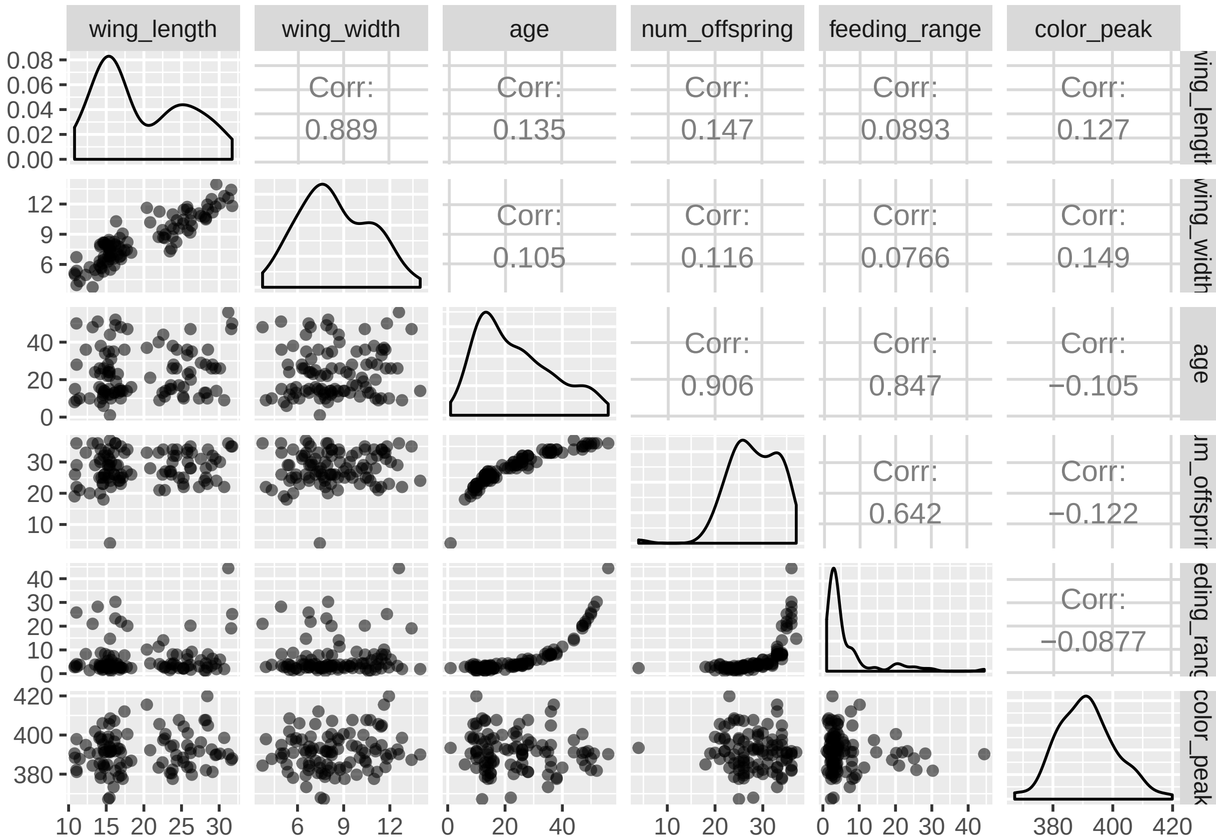
```
butterfly %>%
  gather(key = "field", value = "value", -c(sample_id, population)) %>%
  ggplot(aes(x = population, y = value, fill = population)) +
  geom_violin(adjust = 2) +
  facet_wrap(~field, scales = "free") +
  theme(legend.position = "bottom")
```



Now, using the GGally package, we can also make a scatterplot matrix like we did with graphics::pairs(). In order to get this to be visible, we'll need to split up the data.

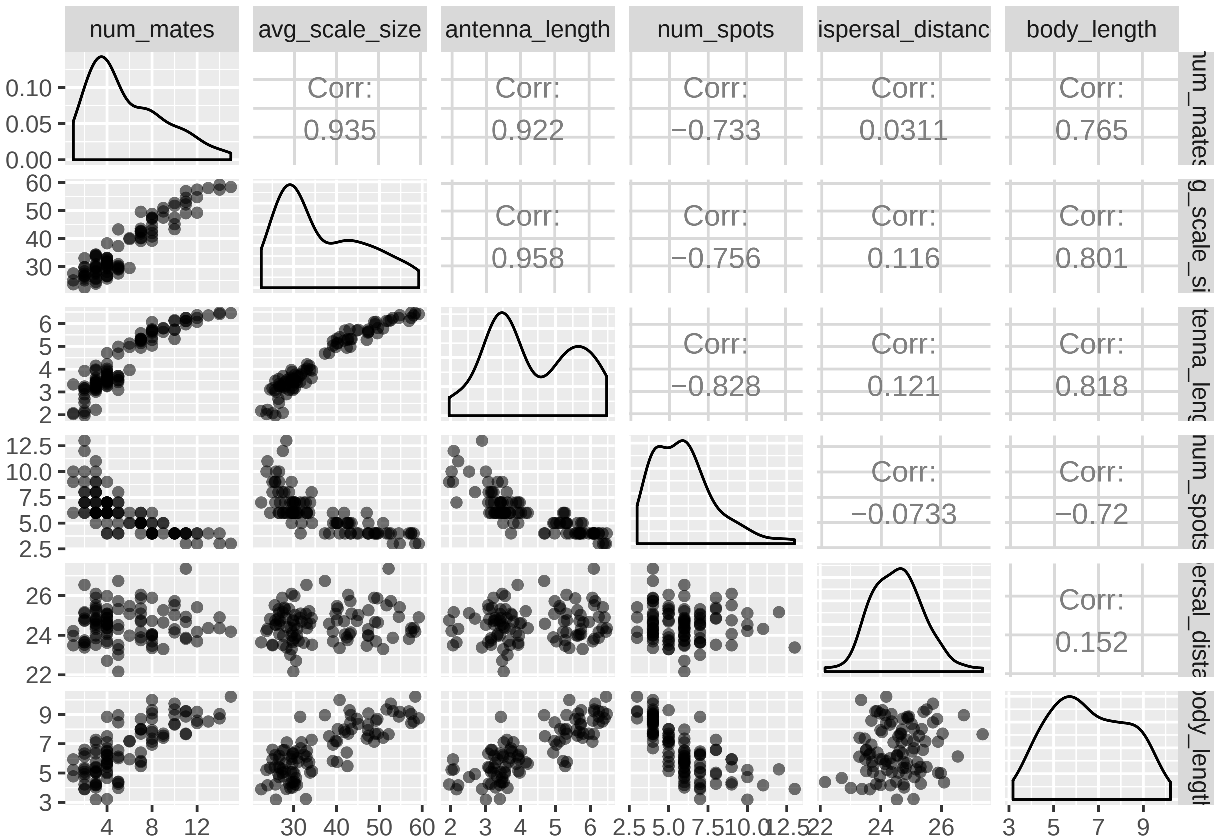
```
library(GGally)

butterfly[1:100, 1:6] %>%
  ggpairs(aes(alpha = 0.2))
```



For the second half of the data, we also need to exclude our non-numeric variables.

```
butterfly[1:100, 7:14] %>%
  select(-c(sample_id, population)) %>%
  ggpairs(aes(alpha = 0.2))
```



As you can probably see, this visualization is not ideal when we have a lot of data.

Let's try a correlation table as well. However, note that while a correlation table can give us a good sense of linaer relationships, we lose any information we had about nonlinear relationships, which we have to examine visually if we don't have a hypothesis about their existence.

```
library(pander)
butterfly %>%
  select(-c(sample_id, population)) %>%
  cor() %>%
  pander()
```

Table 1: Table continues below

	wing_length	wing_width	age	num_offspring
wing_length	1	0.9174	-0.0116	0.0008764
wing_width	0.9174	1	-0.007997	0.001432
age	-0.0116	-0.007997	1	0.9377
num_offspring	0.0008764	0.001432	0.9377	1
feeding_range	0.002284	0.007923	0.8505	0.6989
color_peak	-0.002	-0.005368	0.005909	0.001609
num_mates	0.9625	0.8808	-0.01837	-0.002842
avg_scale_size	0.9796	0.8996	-0.01501	-0.002943
antenna_length	0.9914	0.9102	-0.01246	0.0008284
num_spots	-0.8369	-0.9412	0.01844	0.0129
dispersal_distance	0.01152	0.01272	0.00128	-0.0009606
body_length	0.8831	0.8536	-0.003725	-0.001351

Table 2: Table continues below

	feeding_range	color_peak	num_mates
wing_length	0.002284	-0.002	0.9625
wing_width	0.007923	-0.005368	0.8808
age	0.8505	0.005909	-0.01837
num_offspring	0.6989	0.001609	-0.002842
feeding_range	1	0.003986	-0.003715
color_peak	0.003986	1	-0.009782
num_mates	-0.003715	-0.009782	1
avg_scale_size	0.0001692	0.009505	0.9522
antenna_length	0.0002521	0.0005114	0.9427
num_spots	-0.002536	0.009333	-0.7871
dispersal_distance	0.001011	0.9546	0.005073
body_length	0.004159	-0.005353	0.8472

Table 3: Table continues below

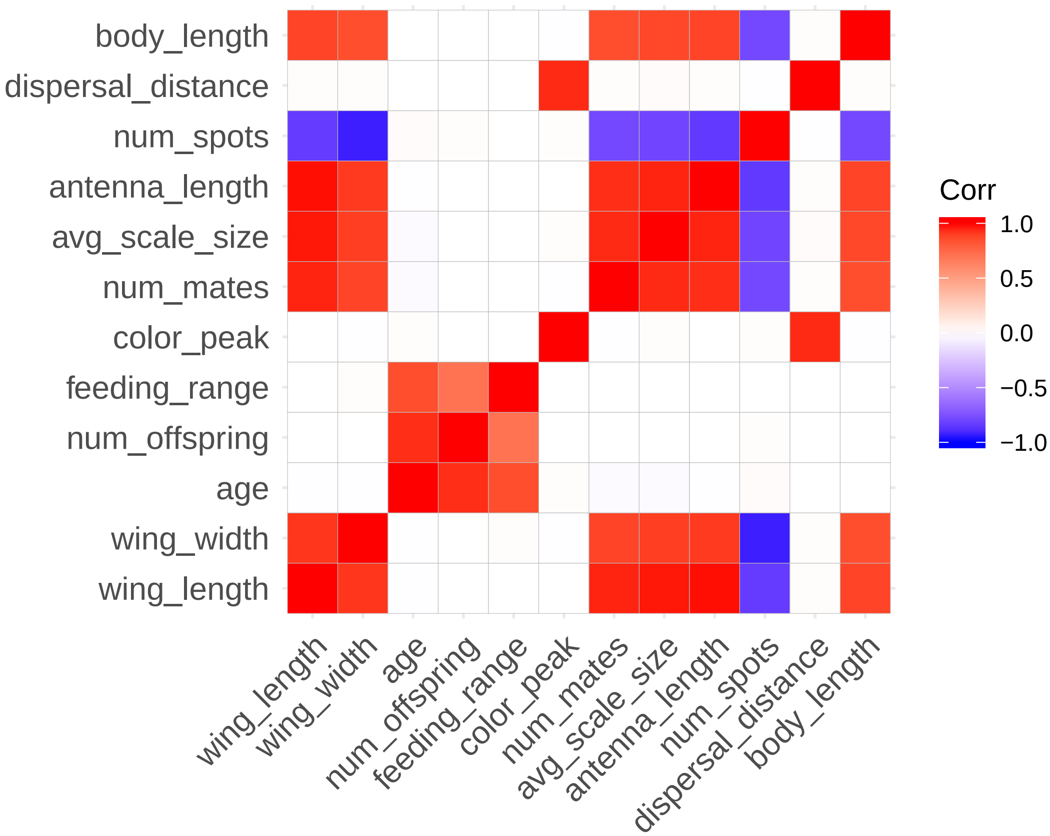
	avg_scale_size	antenna_length	num_spots
wing_length	0.9796	0.9914	-0.8369
wing_width	0.8996	0.9102	-0.9412
age	-0.01501	-0.01246	0.01844
num_offspring	-0.002943	0.0008284	0.0129
feeding_range	0.0001692	0.0002521	-0.002536

	avg_scale_size	antenna_length	num_spots
color_peak	0.009505	0.0005114	0.009333
num_mates	0.9522	0.9427	-0.7871
avg_scale_size	1	0.9578	-0.8022
antenna_length	0.9578	1	-0.8523
num_spots	-0.8022	-0.8523	1
dispersal_distance	0.02136	0.01339	-0.01233
body_length	0.8661	0.8755	-0.787

	dispersal_distance	body_length
wing_length	0.01152	0.8831
wing_width	0.01272	0.8536
age	0.00128	-0.003725
num_offspring	-0.0009606	-0.001351
feeding_range	0.001011	0.004159
color_peak	0.9546	-0.005353
num_mates	0.005073	0.8472
avg_scale_size	0.02136	0.8661
antenna_length	0.01339	0.8755
num_spots	-0.01233	-0.787
dispersal_distance	1	0.005216
body_length	0.005216	1

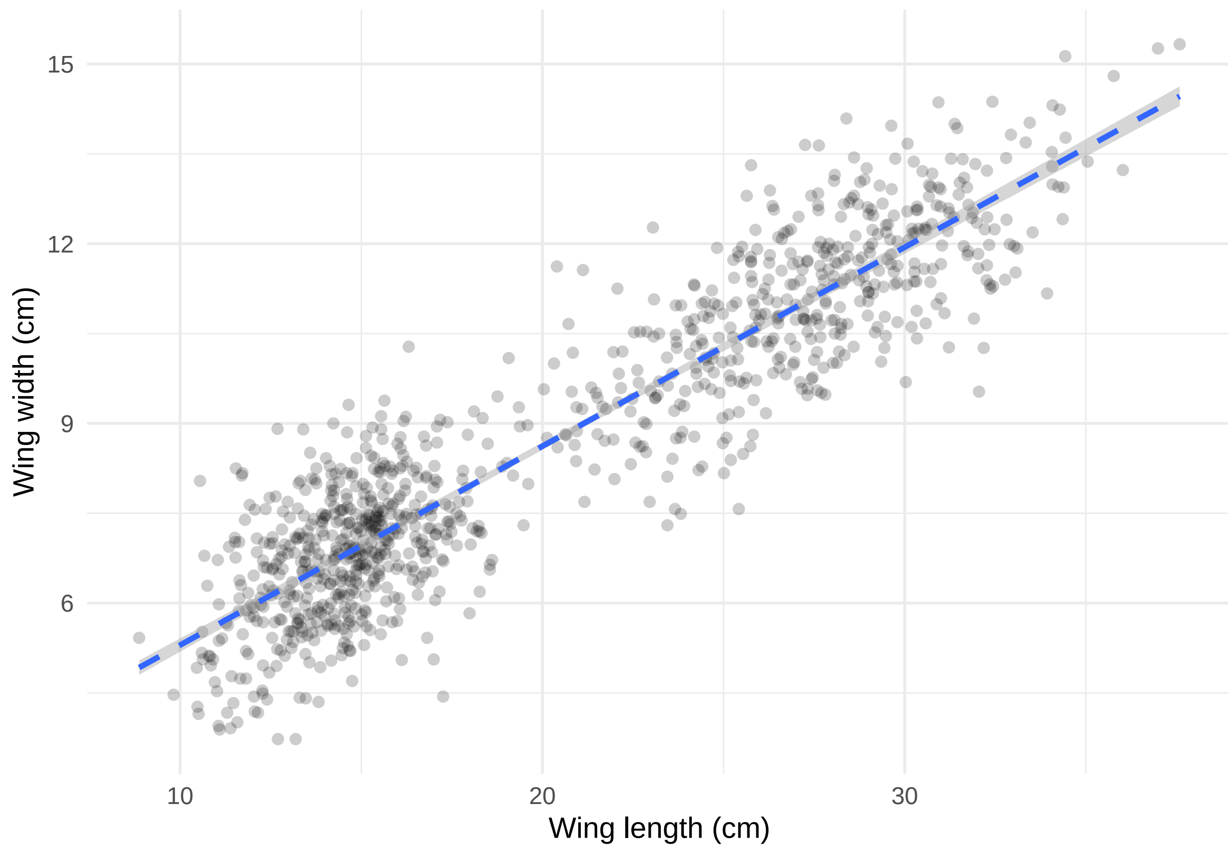
Or, we can use a function from the ggcorrplot library to make a visual representation of the same data.

```
library(ggcorrplot)
butterfly %>%
  select(-c(sample_id, population)) %>%
  cor() %>%
  ggcorrplot()
```



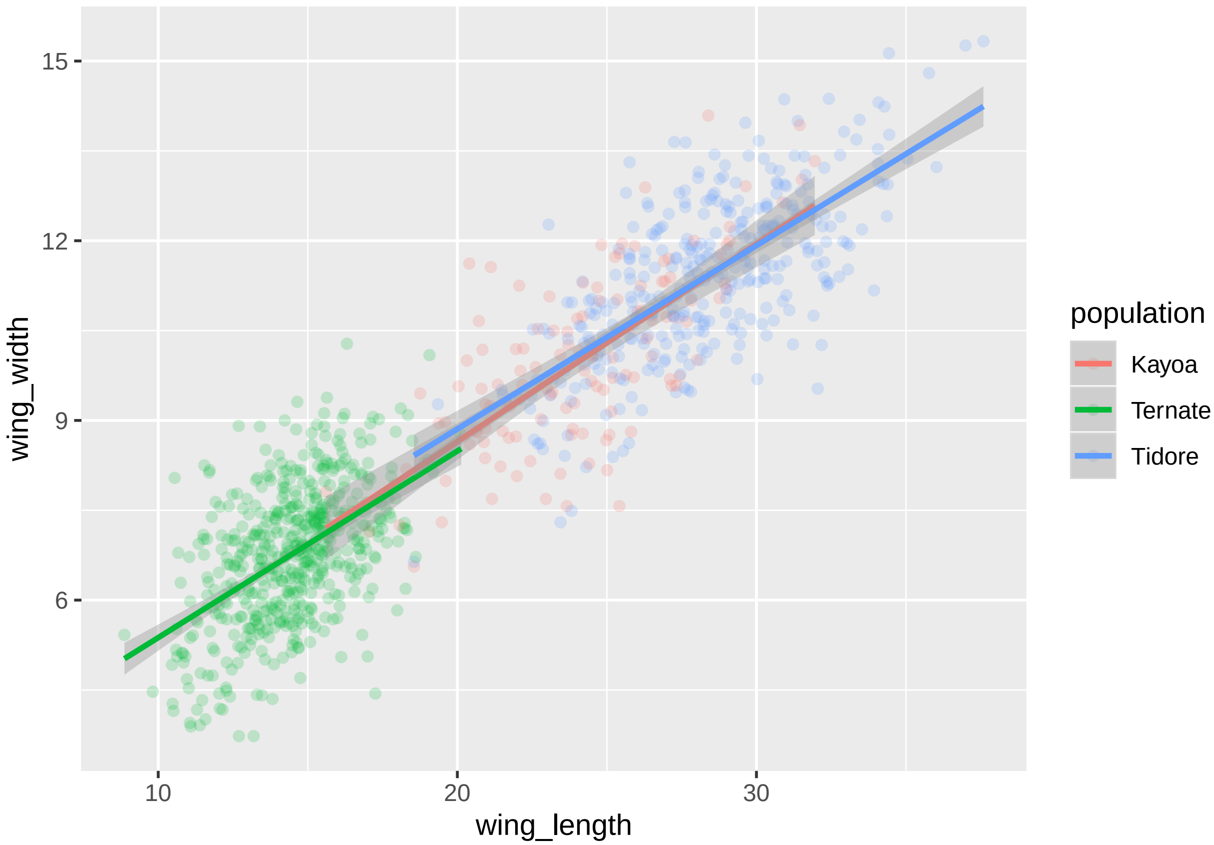
Just an example of a scatterplot.

```
butterfly %>%
  select(wing_length, wing_width) %>%
  ggplot(aes(x = wing_length, y = wing_width)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", lty = 2) +
  labs(x = "Wing length (cm)", y = "Wing width (cm)") +
  theme_minimal()
```



We can also do a LOT with scatterplots.

```
butterfly %>%
  select(wing_length, wing_width, population) %>%
  ggplot(aes(x = wing_length,
             y = wing_width,
             col = population)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm")
```



There's literally so many aesthetic arguments you can use for a scatterplot (I contend that not all of them are useful).

```
butterfly %>%
  select(wing_length, wing_width, population, antenna_length, body_length) %>%
  ggplot(aes(x = wing_length,
             y = wing_width,
             col = antenna_length,
             size = body_length)) +
  geom_point(alpha = 0.2) +
  facet_wrap(~population)
```