Data Visualization with ggplot2

```
## you can set your working directory in this chunk

#setwd("/Users/namigabbasov/Desktop/R-Data-Carpentry")
```

```
## libraries

#install.packages("tidyverse")
#install.packages("magrittr")

library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
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 beco
```

```
library(magrittr)
```

```
Attaching package: 'magrittr'

The following object is masked from 'package:purrr':

    set_names
```

```
The following object is masked from 'package:tidyr':

    extract
```

```
## import data
surveys_complete<-read_csv("https://raw.githubusercontent.com/UnitForDataScience/RWorkshop/ma
```

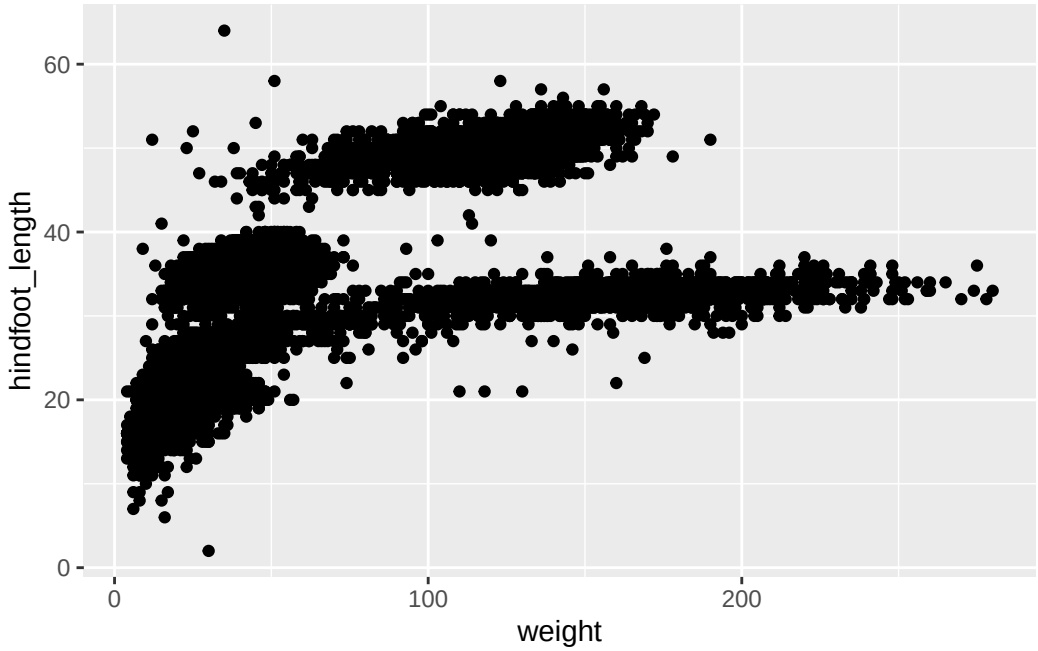
```
Rows: 30463 Columns: 13
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (6): species_id, sex, genus, species, taxa, plot_type
dbl (7): record_id, month, day, year, plot_id, hindfoot_length, weight

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.
```

Plotting with ggplot2: Format

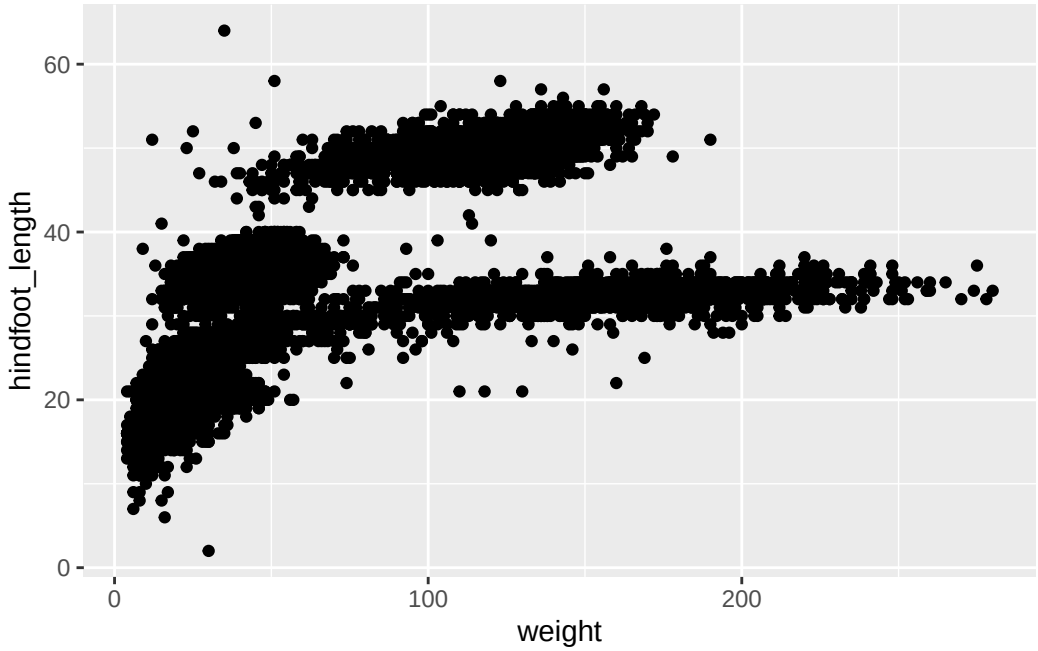
```
# ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) +  <GEOM_FUNCTION>()
```

Scatterplot

```
ggplot(data = surveys_complete, aes(x = weight, y = hindfoot_length)) +
  geom_point()
```

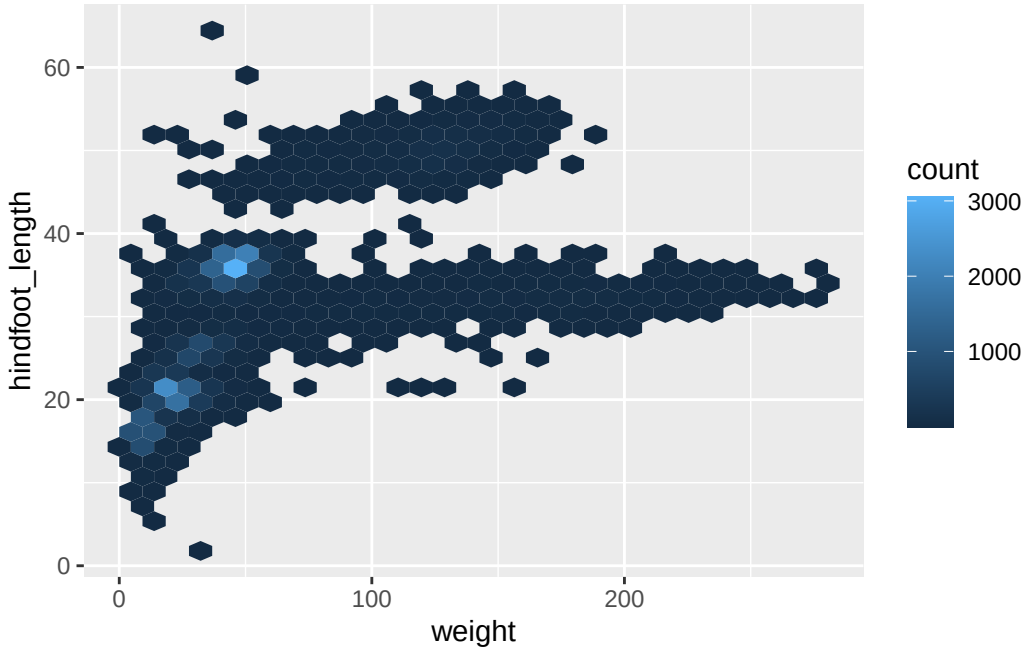



```
# Assign plot to a variable
surveys_plot <- ggplot(data = surveys_complete,
                       mapping = aes(x = weight, y = hindfoot_length))

# Draw the plot
surveys_plot +
    geom_point()
```



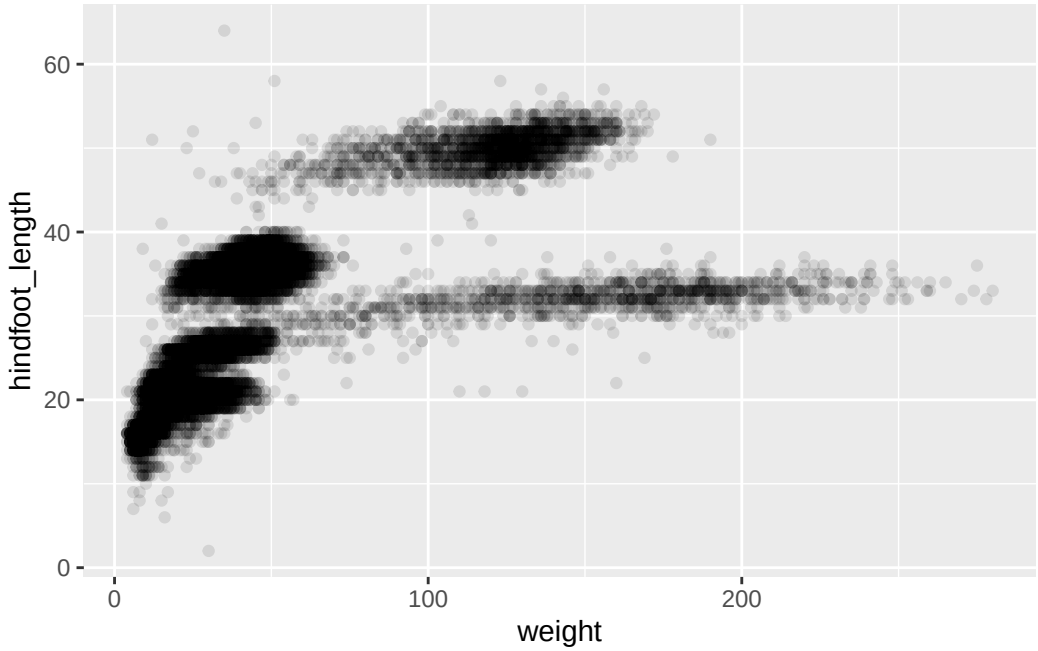
Hexbin

```
#install.packages("hexbin")
library(hexbin)

ggplot(data = surveys_complete,
                      mapping = aes(x = weight, y = hindfoot_length))+
  geom_hex()
```

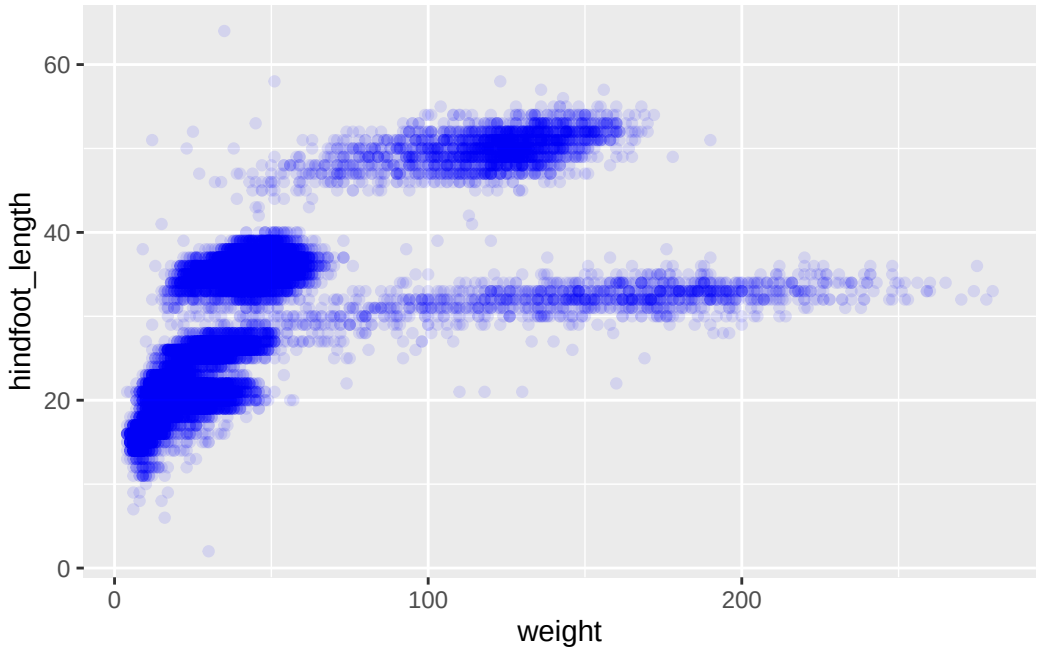


Building your plots iteratively

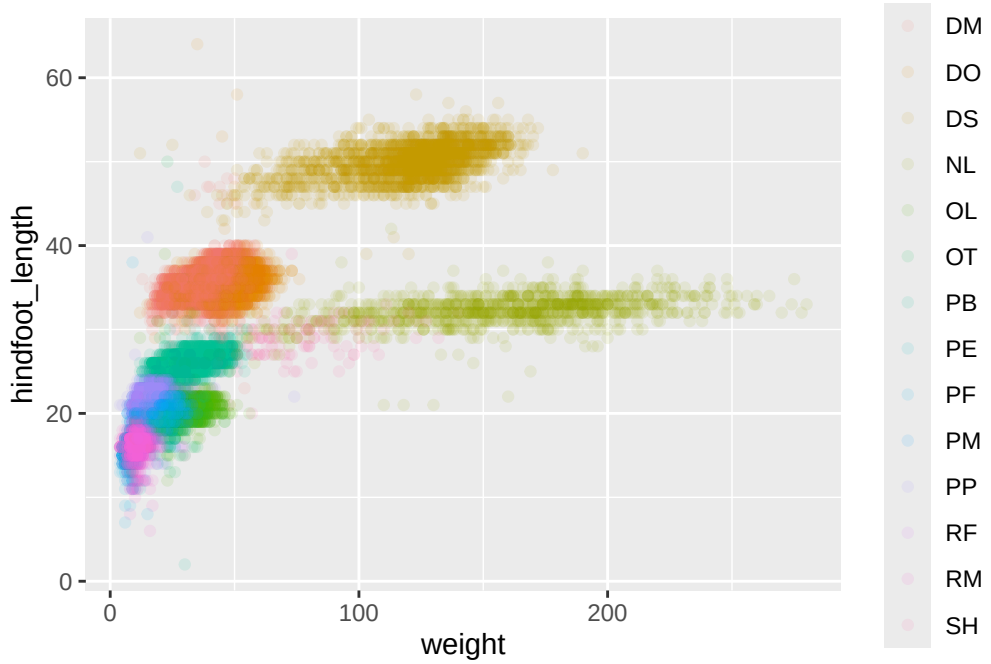
```
## add transparency (alpha) to avoid overplotting
ggplot(data = surveys_complete, aes(x = weight, y = hindfoot_length)) +
    geom_point(alpha = 0.1)
```



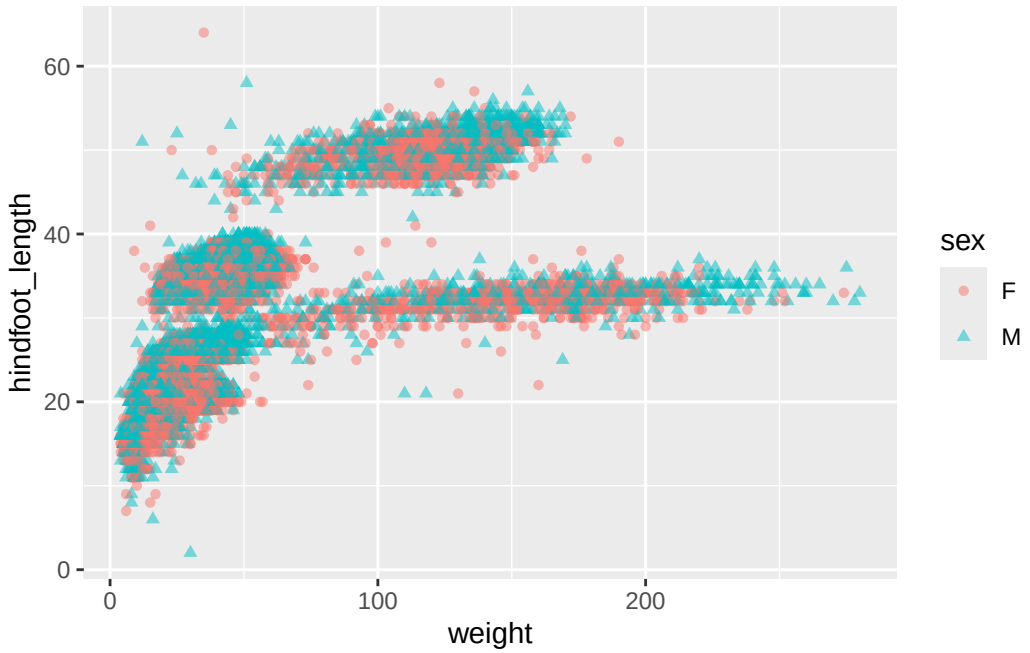

```
## add color
ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length)) +
    geom_point(alpha = 0.1, color = "blue")
```



```
## color by species_id
ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length)) +
    geom_point(alpha = 0.1, aes(color = species_id))
```

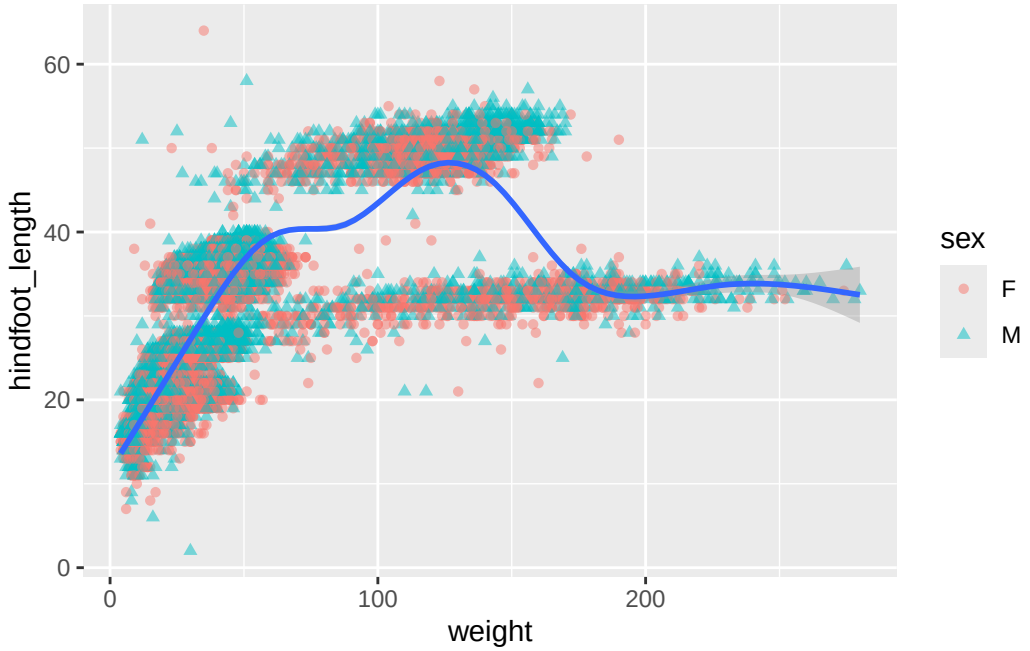



```
## shape by species_id
ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length, color =sex)) +
    geom_point(alpha = 0.5, aes(shape = sex))
```



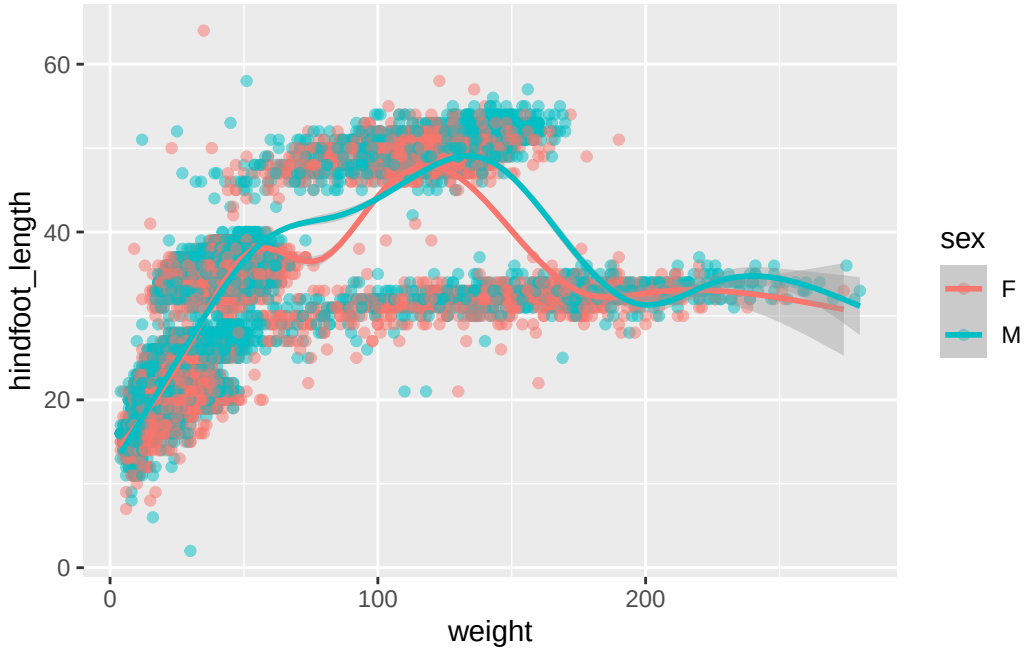

```
## add a line: global environment
ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length)) +
    geom_point(alpha = 0.5, aes(shape = sex, color =sex))+
  geom_smooth()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



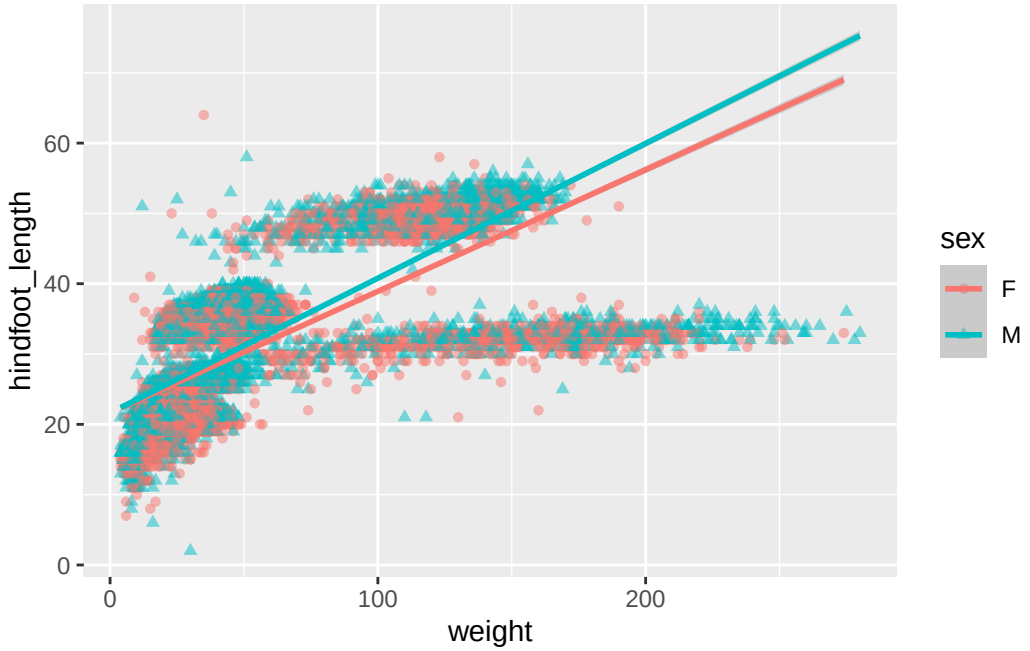

```
## add a line: local environment
library(ggplot2)

ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length, color = sex))
  geom_point(alpha = 0.5) +
  geom_smooth()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



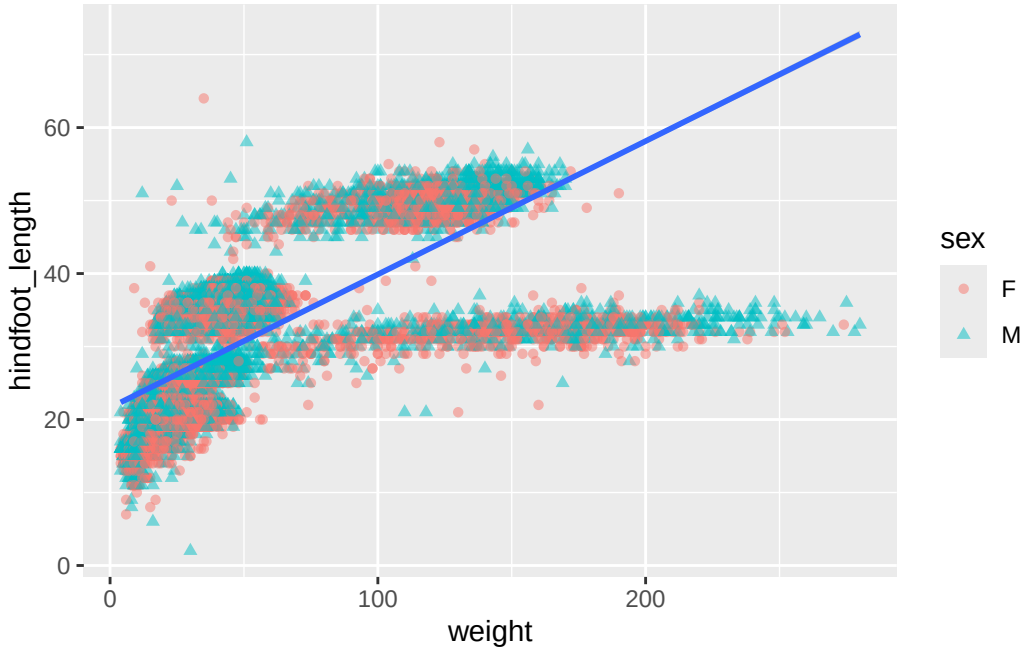

```
## Add a straight line
ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length, color = sex, s
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm")
```

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



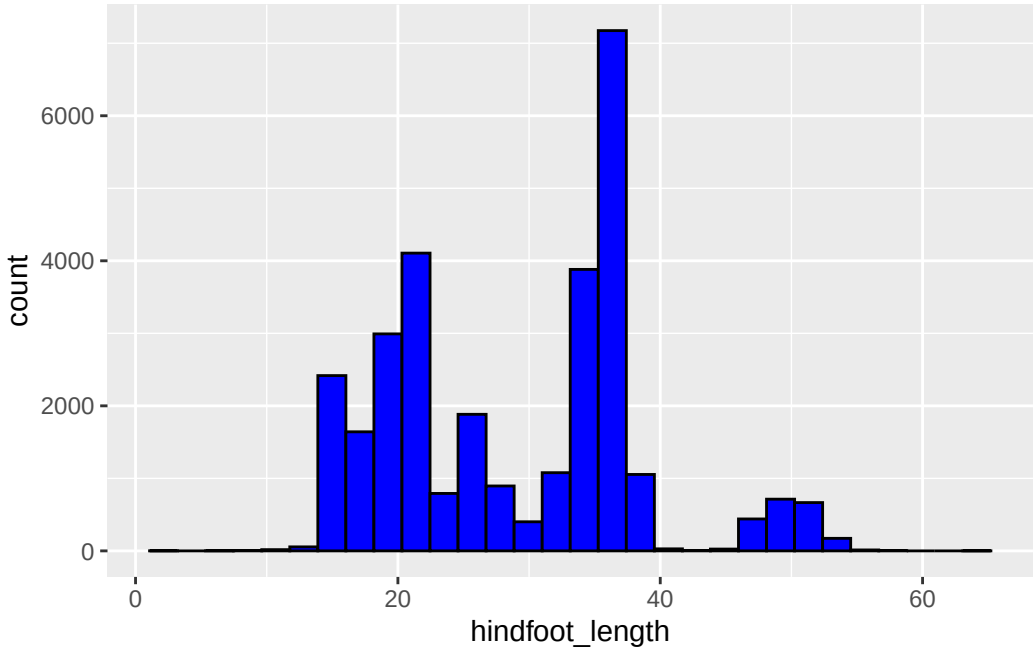

```
ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length)) +
  geom_point(alpha = 0.5, aes(color = sex, shape = sex)) +
  geom_smooth(method = "lm")
```

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



Distribution of a variable: Histogram

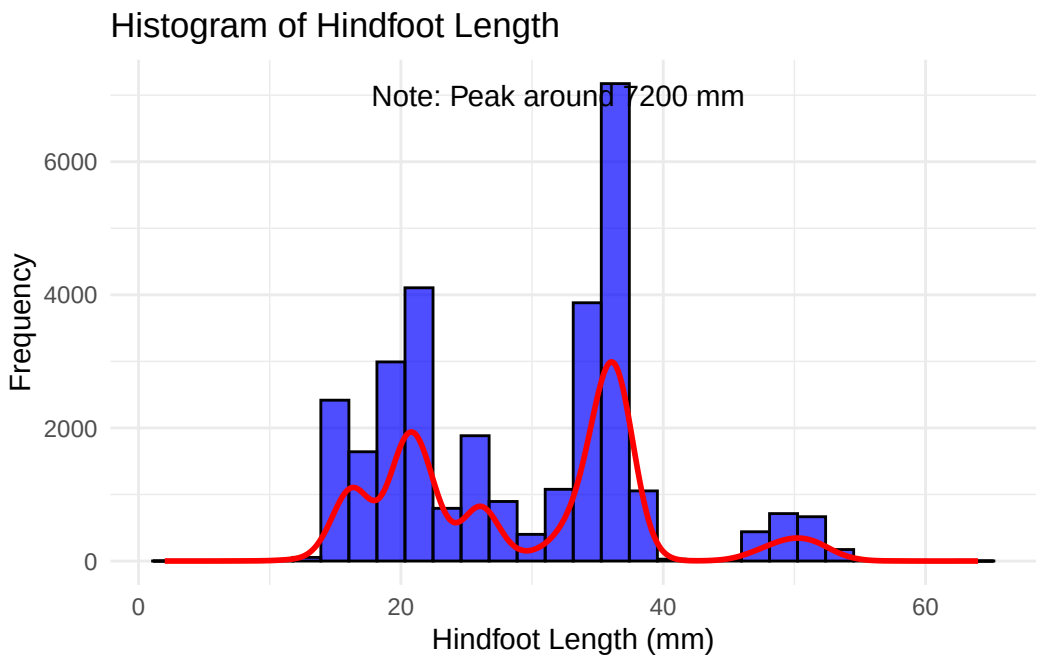
```
ggplot(data = surveys_complete, aes(x = hindfoot_length)) +
  geom_histogram(bins = 30, fill = "blue", color = "black")
```




```
## A publication level Histogram
ggplot(data = surveys_complete, aes(x = hindfoot_length)) +
  geom_histogram(bins = 30, fill = "blue", color = "black", alpha = 0.7) +
  geom_density(aes(y = ..count..), color = "red", size = 1) +
  labs(title = "Histogram of Hindfoot Length",
       x = "Hindfoot Length (mm)",
       y = "Frequency") +
  theme_minimal() +
  annotate("text", x = 32, y = 7000, label = "Note: Peak around 7200 mm", color = "black", si
```

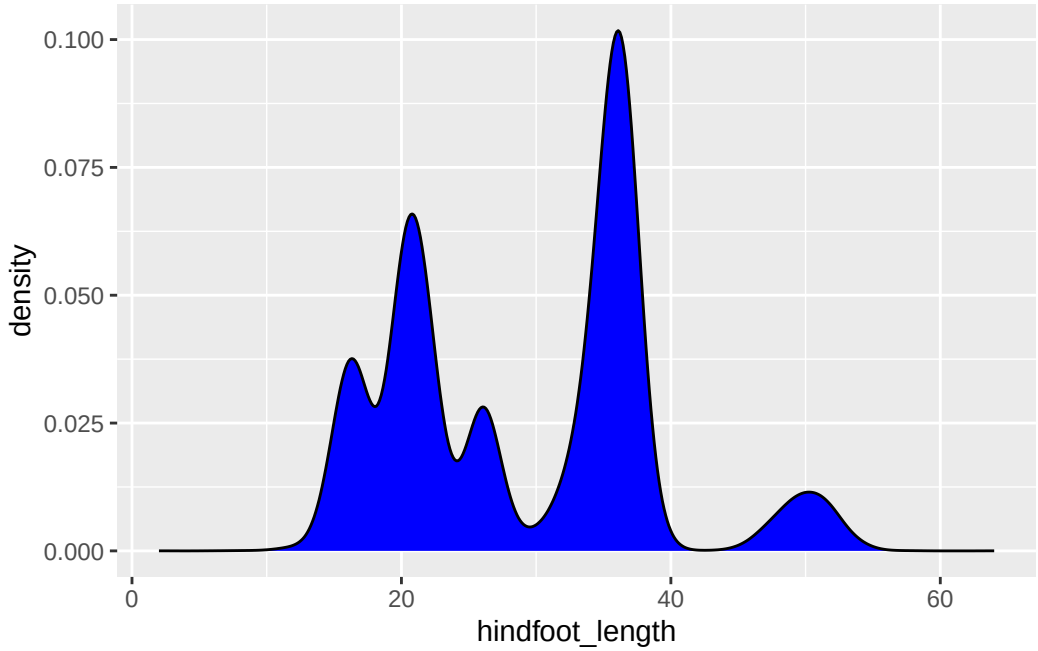
```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
i Please use `after_stat(count)` instead.
```

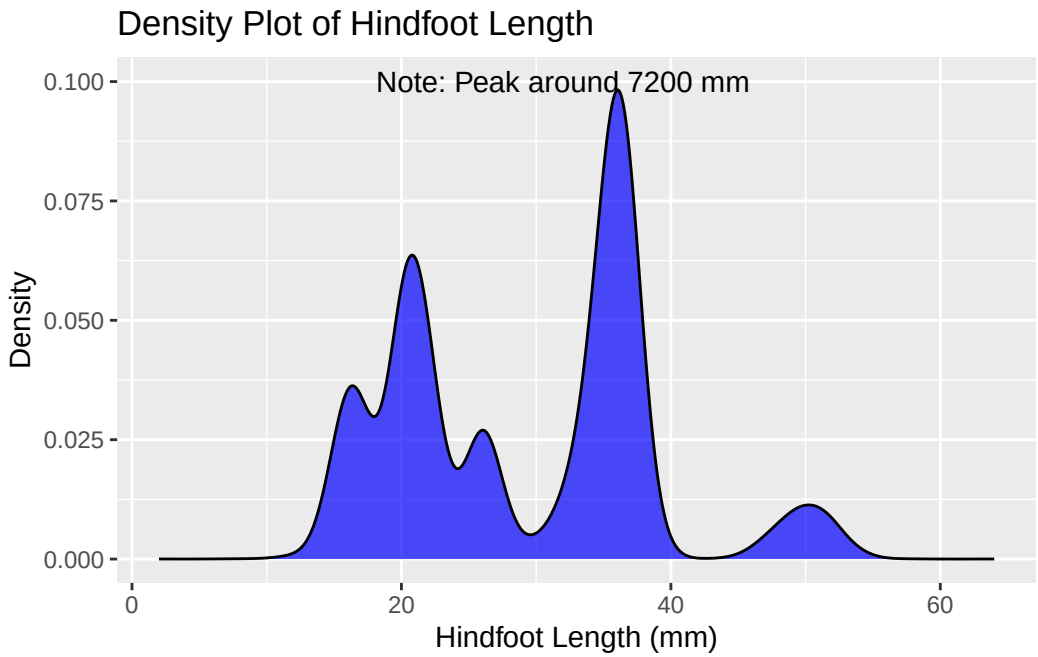


Distribution of a variable: Density Plot

```
ggplot(data = surveys_complete, aes(x = hindfoot_length)) +
  geom_density(fill = "blue", color = "black", bw=1)
```

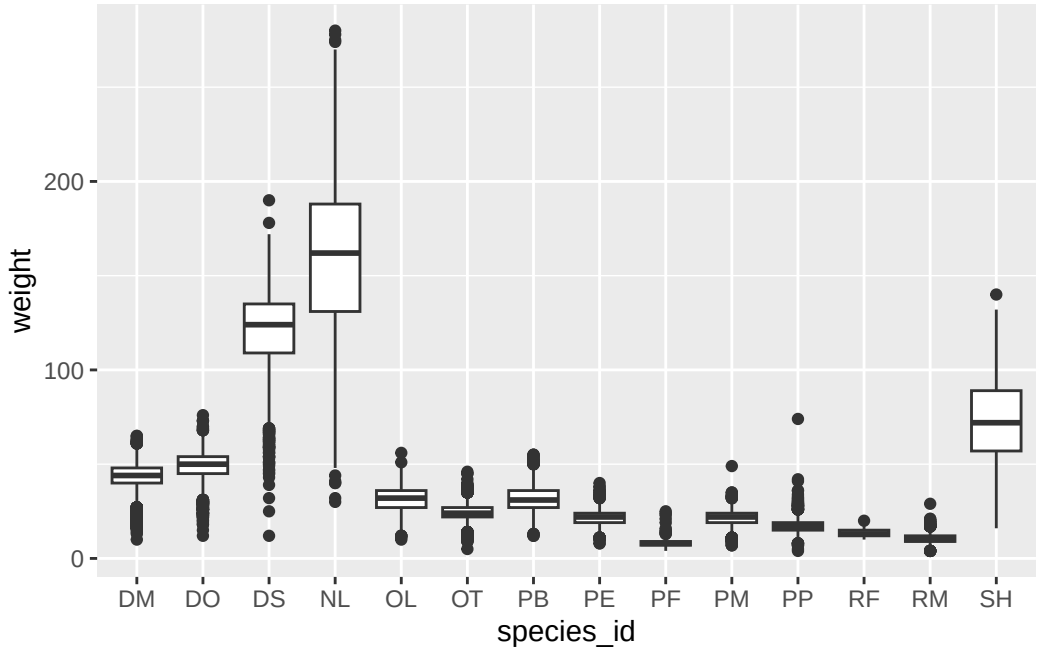



```
ggplot(data = surveys_complete, aes(x = hindfoot_length)) +
  geom_density(fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Density Plot of Hindfoot Length",
       x = "Hindfoot Length (mm)",
       y = "Density") +
  annotate("text", x = 32, y = 0.1, label = "Note: Peak around 7200 mm", color = "black", si
```

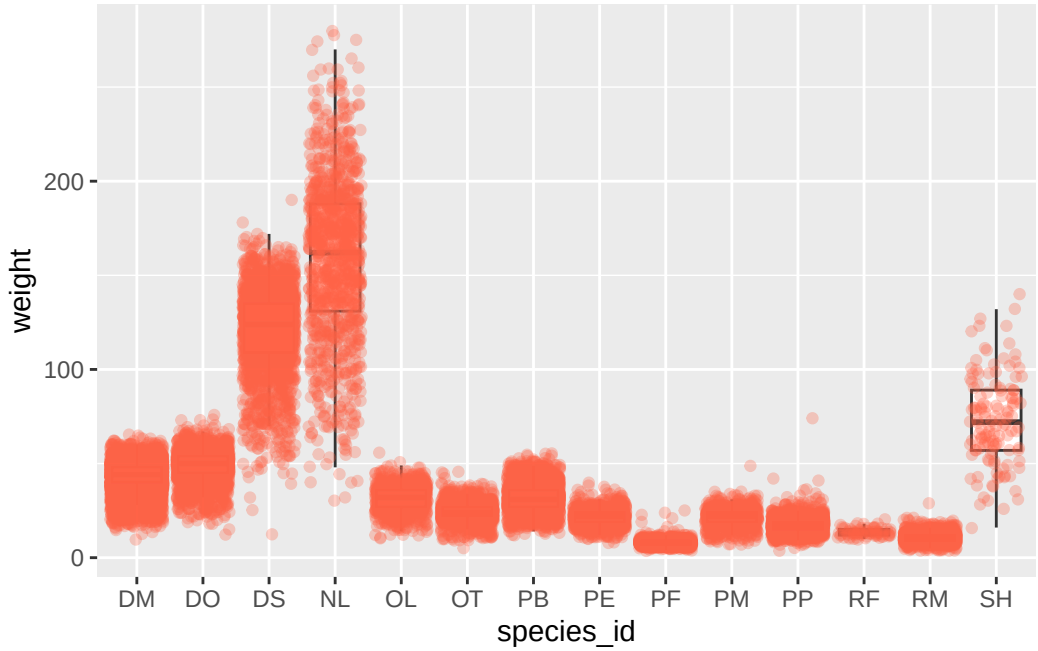


Boxplot: Continous variable and categorical variable

```
ggplot(data = surveys_complete, mapping = aes(x = species_id, y = weight)) +
    geom_boxplot()
```

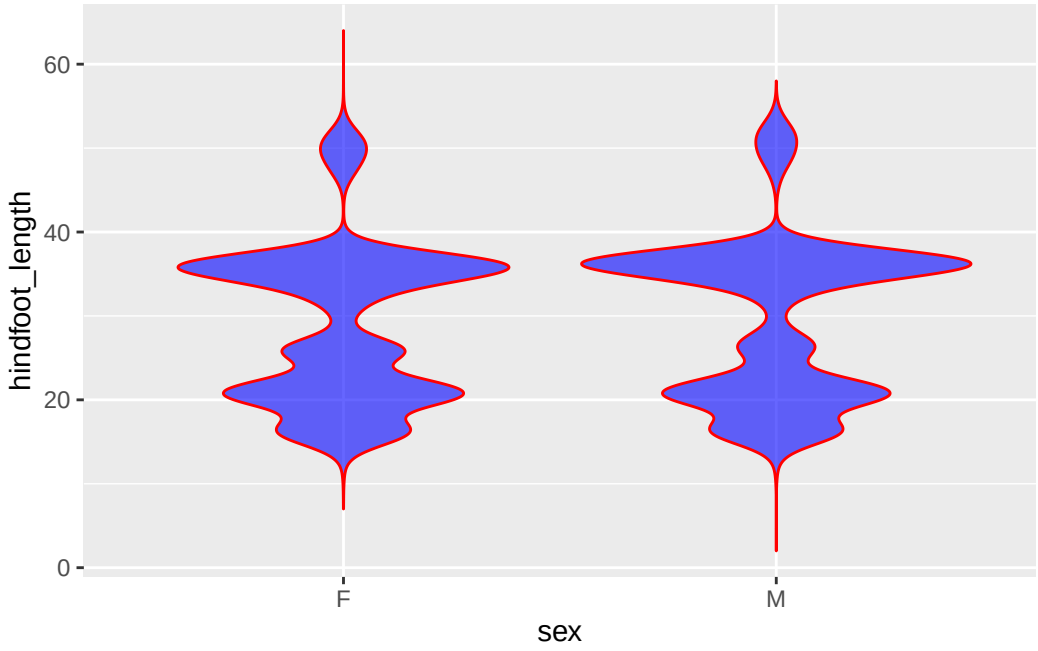



```
# add color, jitters
ggplot(data = surveys_complete, mapping = aes(x = species_id, y = weight)) +
    geom_boxplot(outlier.shape = NA) +
    geom_jitter(alpha = 0.3, color = "tomato")
```



violin: Continuous variable and categorical variable

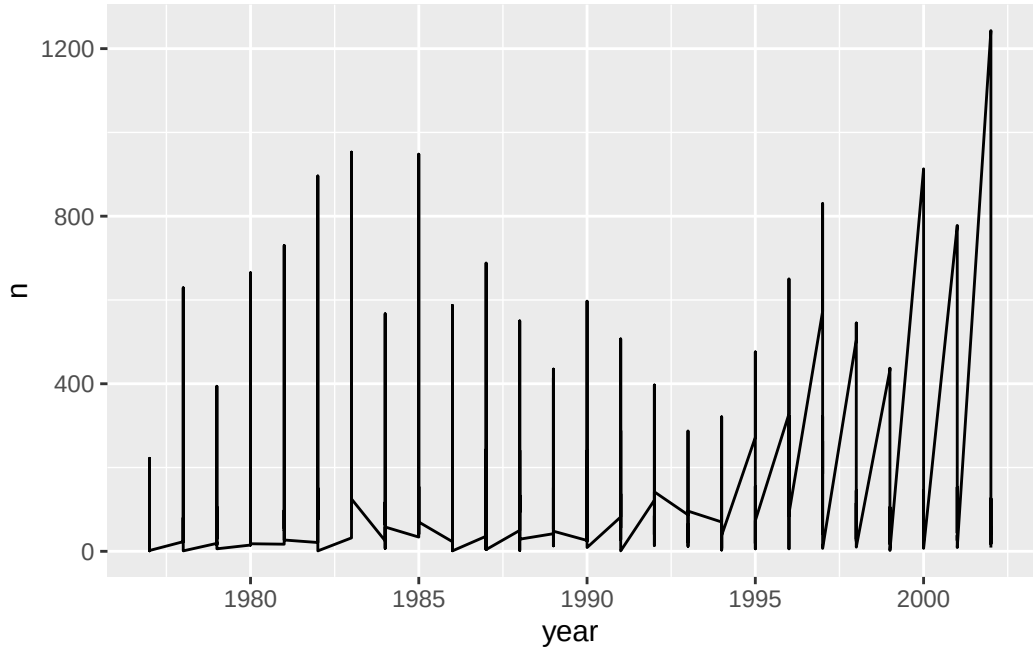
```
ggplot(data = surveys_complete, mapping = aes(x = sex, y = hindfoot_length)) +
   geom_violin(color = "red", fill = "blue", alpha =.6)
```



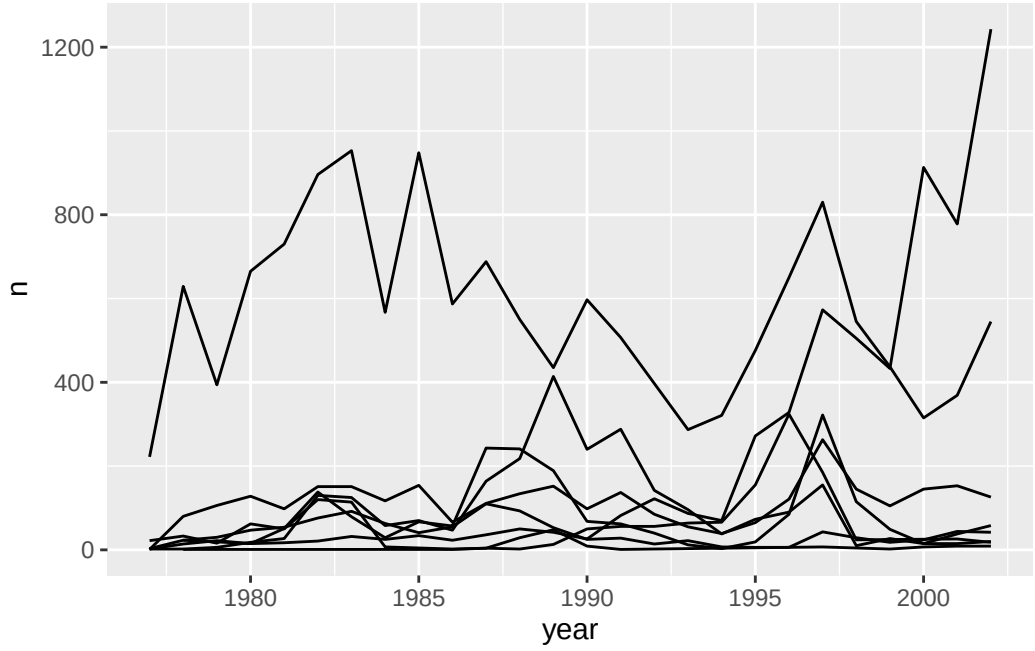
Plotting time series data

```
## prepare data first
yearly_counts <- surveys_complete %>%
  count(year, genus)
```

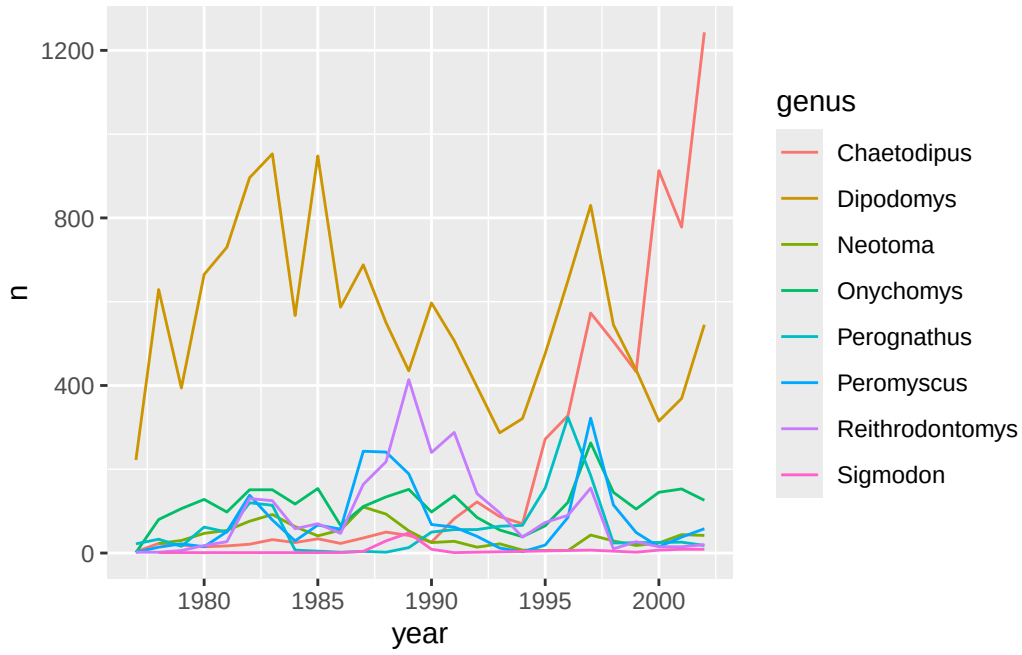
```
## Now plot it with line plot
ggplot(data = yearly_counts, aes(x = year, y = n)) +
     geom_line()
```




```
## modifying the aesthetic function, group = genus
ggplot(data = yearly_counts, aes(x = year, y = n, group = genus)) +
    geom_line()
```

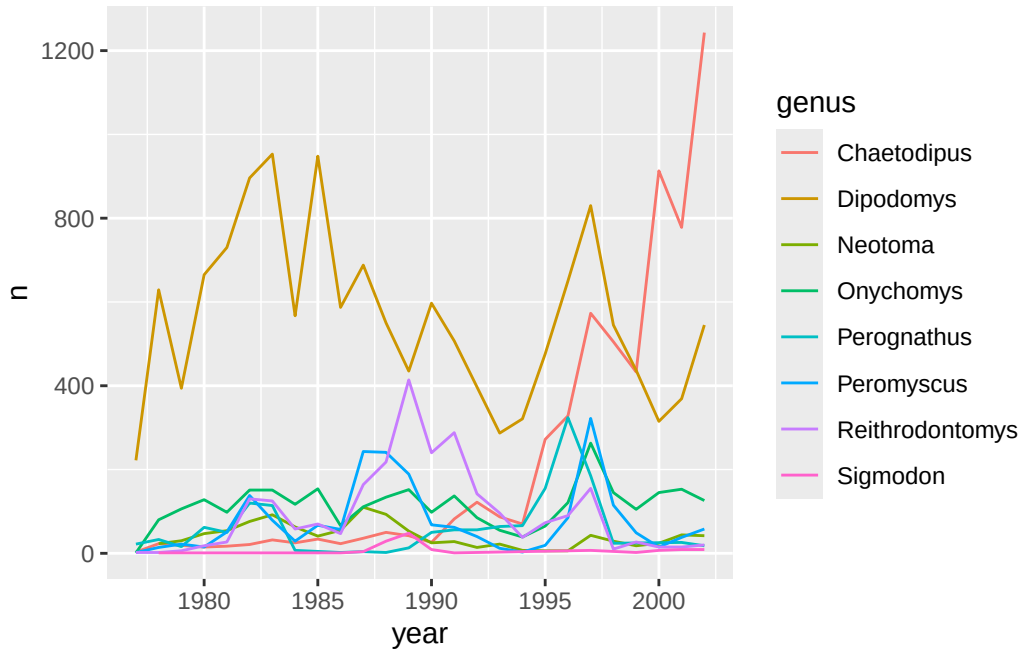


```
## add lines to distinguish colors
ggplot(data = yearly_counts, aes(x = year, y = n, color = genus)) +
    geom_line()
```

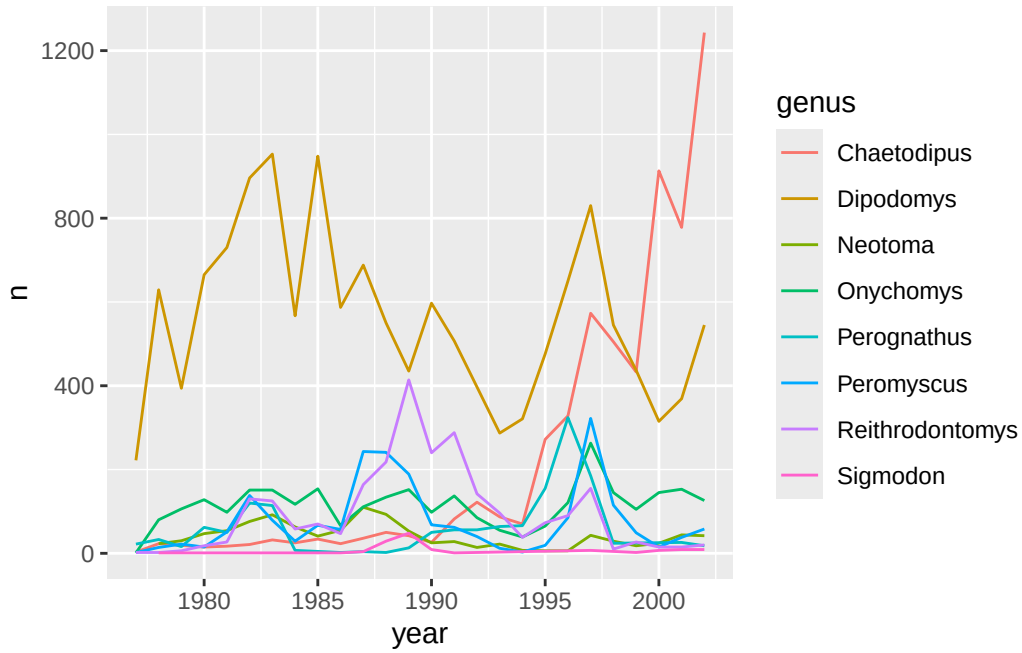


Integrating the pipe operator with ggplot2

```
yearly_counts %>%
    ggplot(mapping = aes(x = year, y = n, color = genus)) +
    geom_line()
```

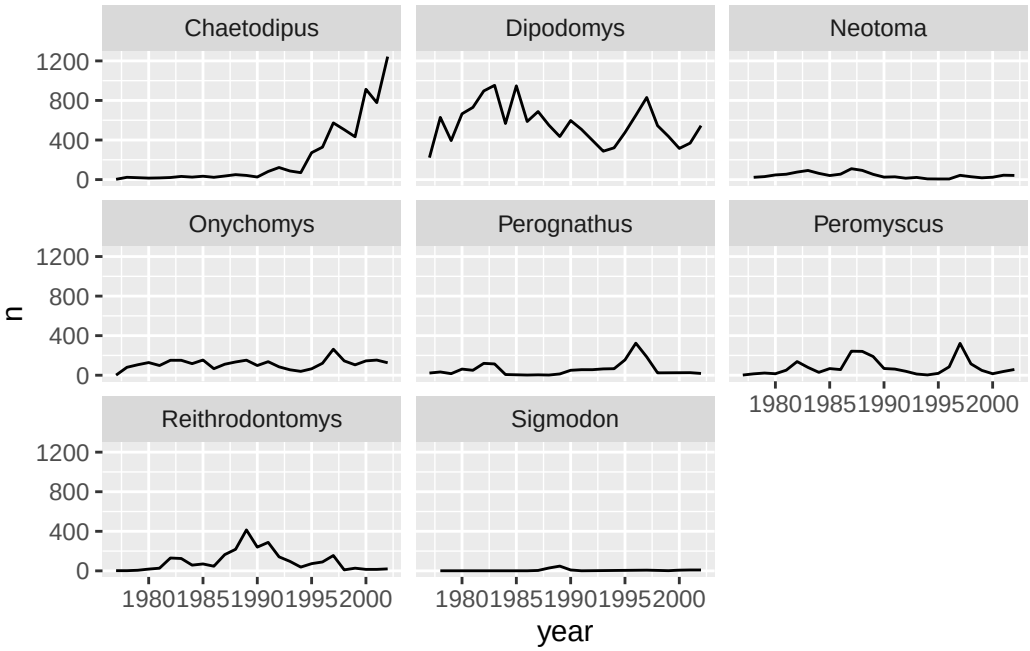



```
## use data manipulation before ggplot with pipe
yearly_counts_graph <- surveys_complete %>%
    count(year, genus) %>%
    ggplot(mapping = aes(x = year, y = n, color = genus)) +
    geom_line()

yearly_counts_graph
```



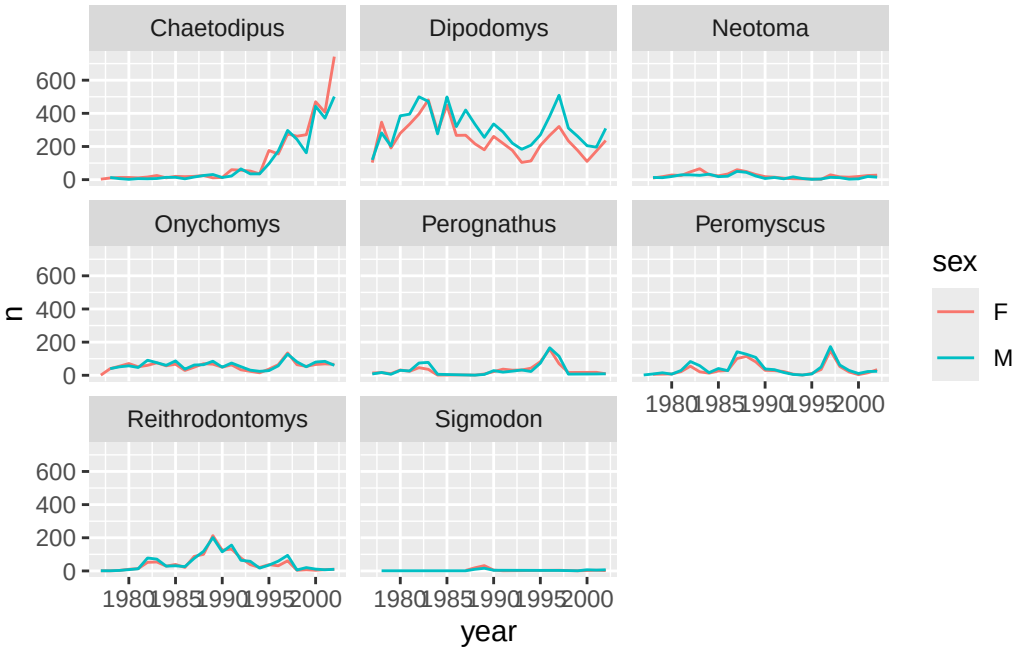
Faceting

```
ggplot(data = yearly_counts, aes(x = year, y = n)) +
    geom_line() +
    facet_wrap(facets = vars(genus))
```

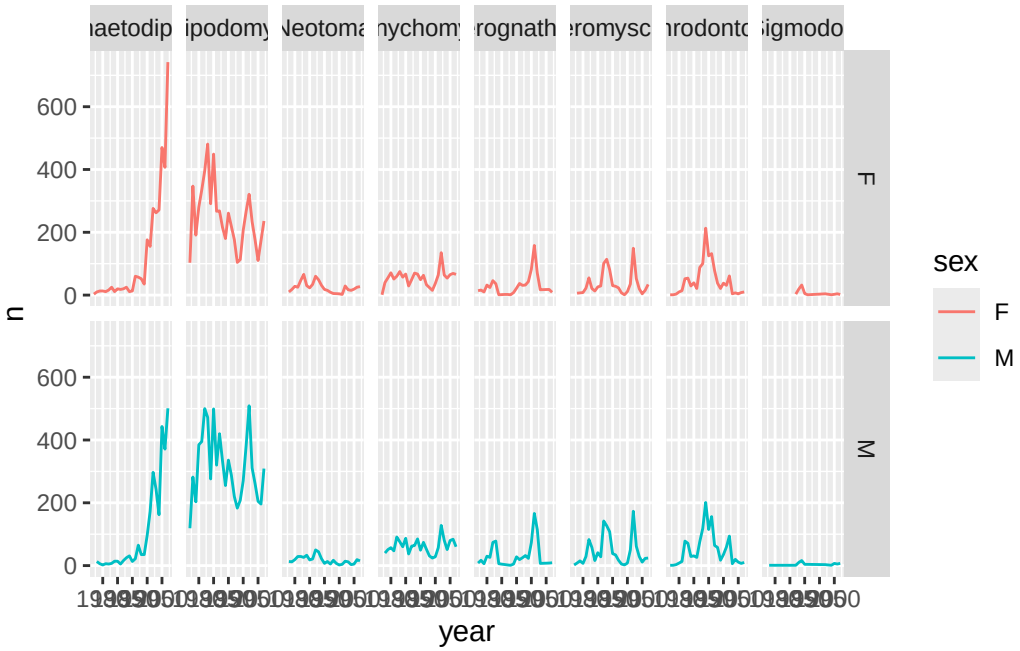



```
## split the line in each plot by the sex of each individual measured
yearly_sex_counts <- surveys_complete %>%
                      count(year, genus, sex)
```

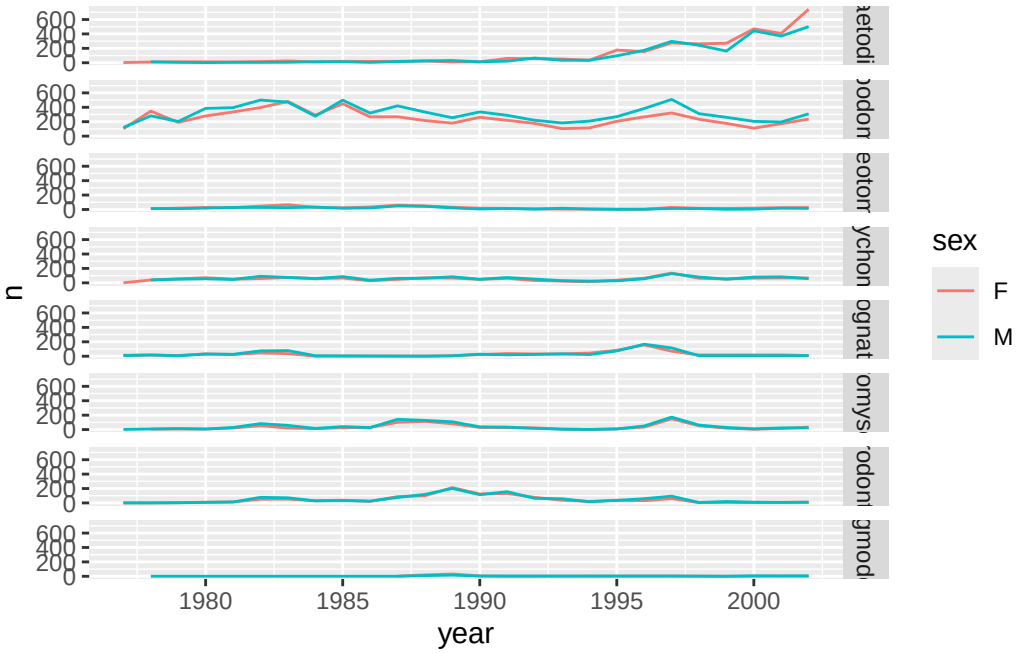
```
## now make the faceted plot by splitting further by sex using color (within a single plot)
ggplot(data = yearly_sex_counts, mapping = aes(x = year, y = n, color = sex)) +
  geom_line() +
  facet_wrap(facets =  vars(genus))
```



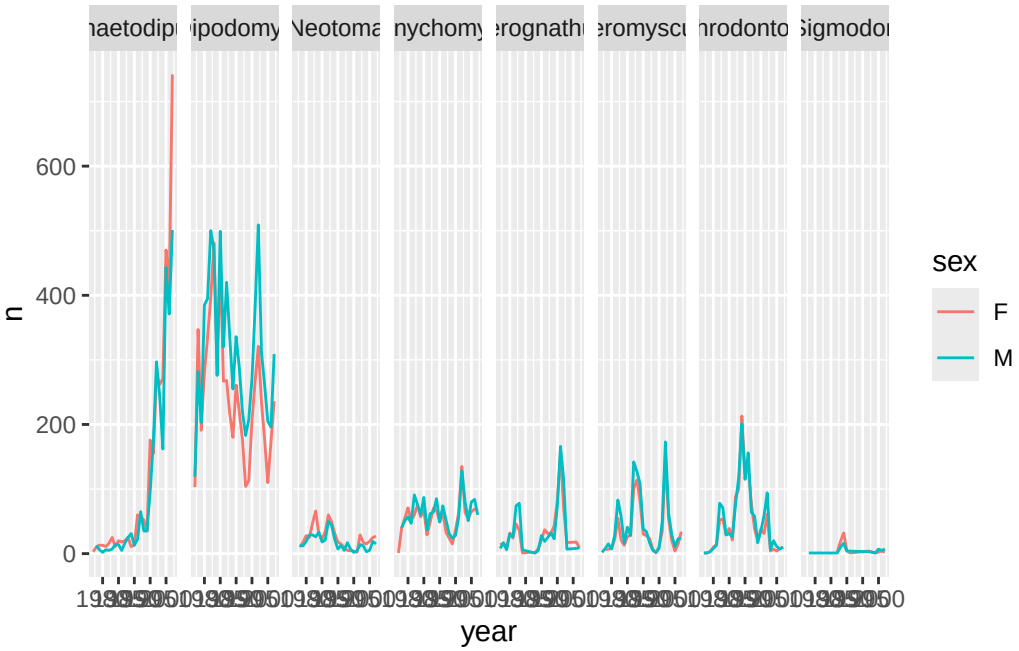

```
## We can also facet both by sex and genus
ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, color = sex)) +
  geom_line() +
  facet_grid(rows = vars(sex), cols =  vars(genus))
```




```
## organise the panels only by rows (or only by columns)

# One column, facet by rows
ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, color = sex)) +
  geom_line() +
  facet_grid(rows = vars(genus))
```

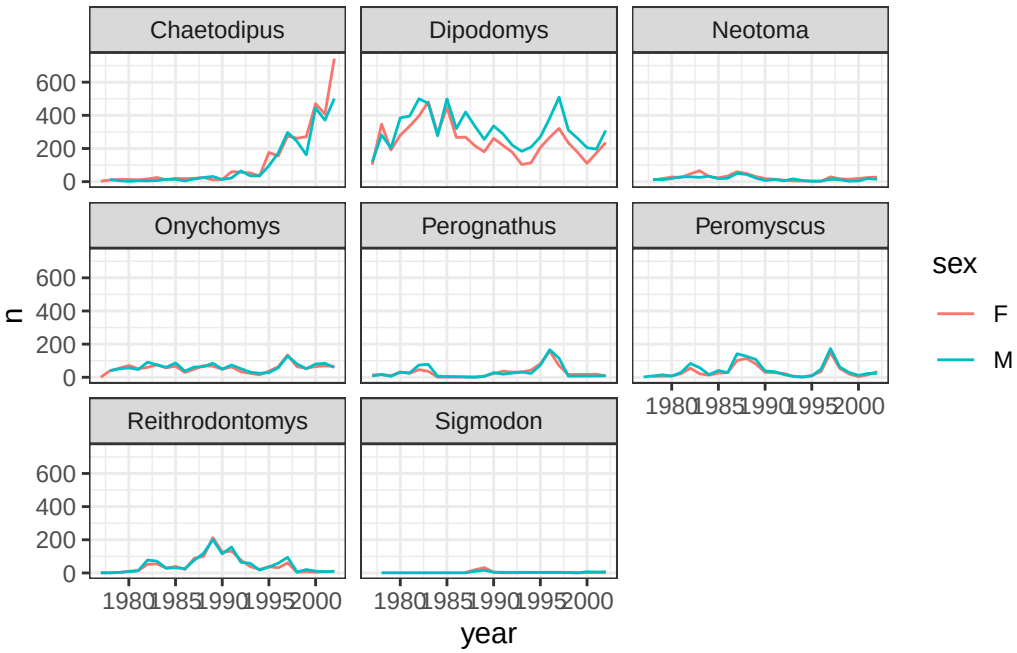



```
# One row, facet by column
ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, color = sex)) +
  geom_line() +
  facet_grid(cols = vars(genus))
```

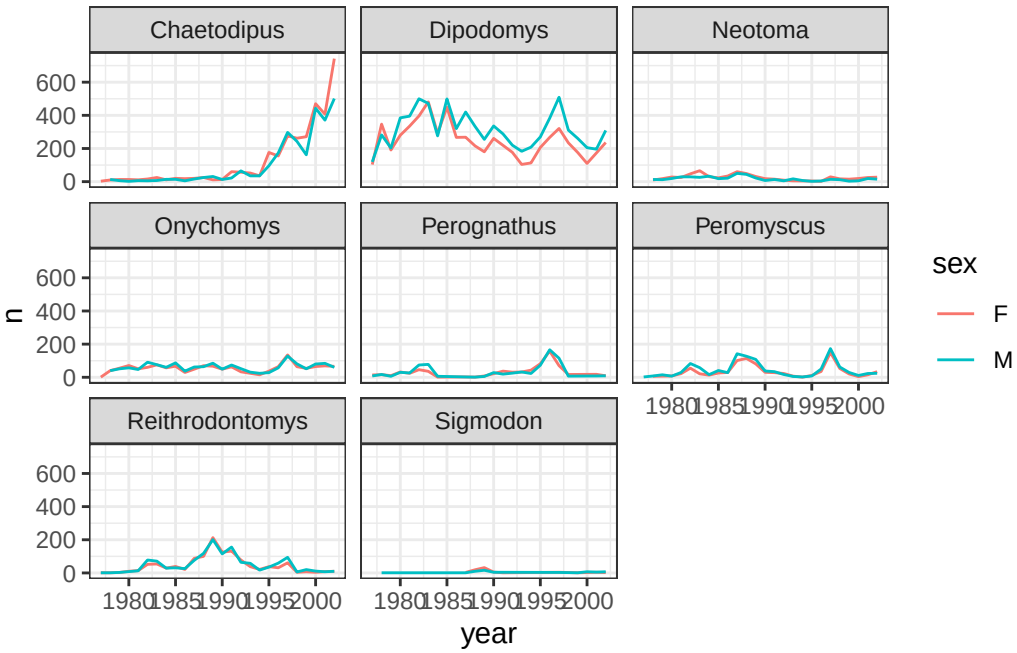


ggplot2 themes

```
## a simpler white background using the theme_bw() function
ggplot(data = yearly_sex_counts,
        mapping = aes(x = year, y = n, color = sex)) +
    geom_line() +
    facet_wrap(vars(genus)) +
    theme_bw()
```

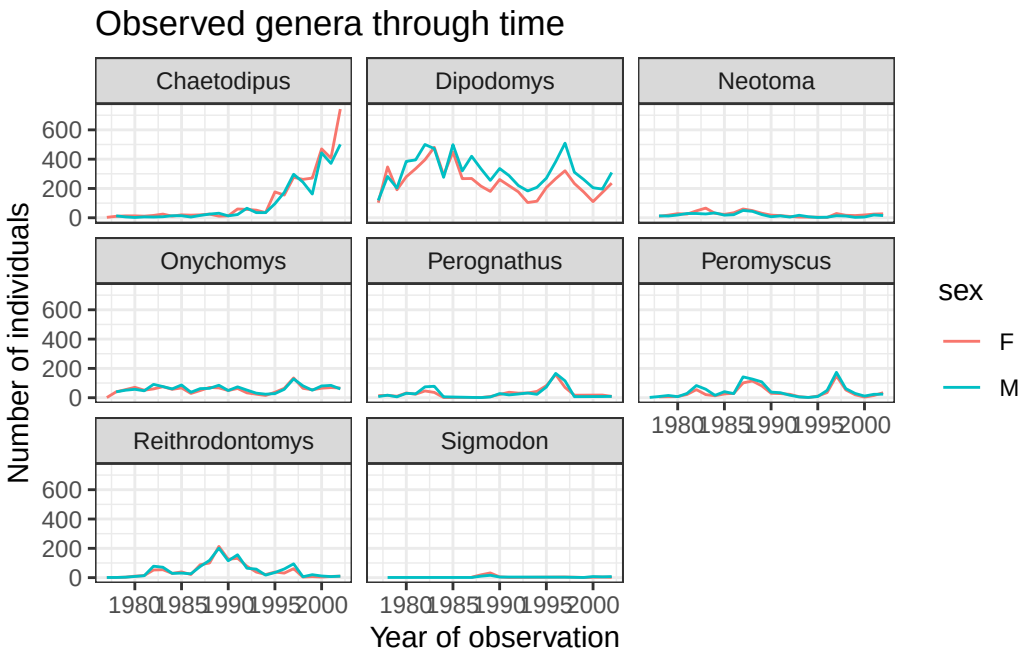



```
## Classic theme
ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, color = sex)) +
    geom_line() +
    facet_wrap(vars(genus)) +
    theme_bw()
```

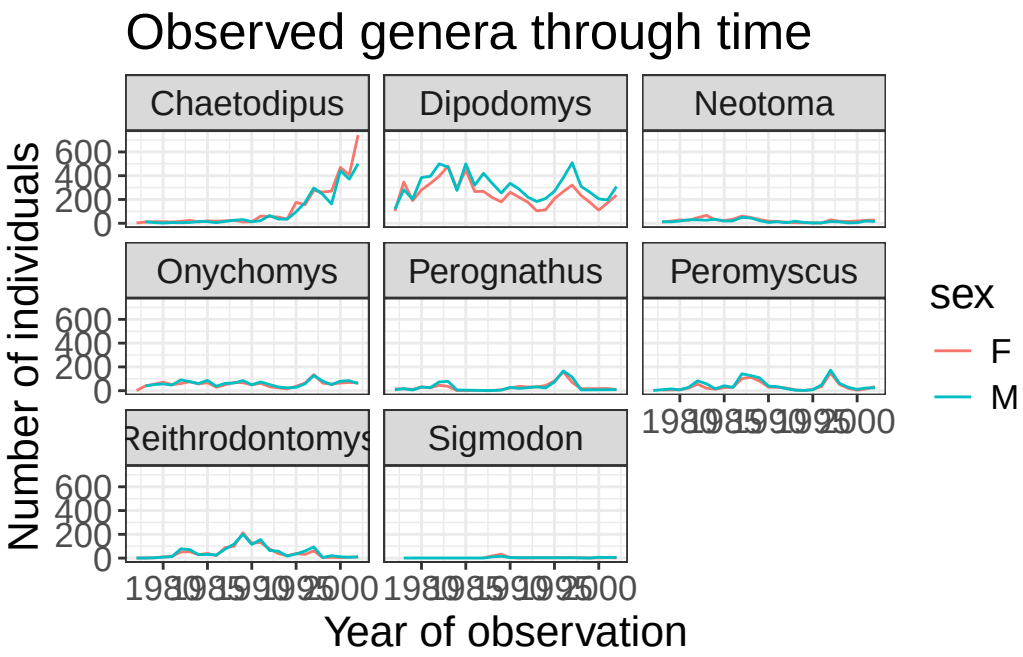


Customization

```
## add titles, x-axis name, y-axis name
ggplot(data = yearly_sex_counts, aes(x = year, y = n, color = sex)) +
    geom_line() +
    facet_wrap(vars(genus)) +
    labs(title = "Observed genera through time",
         x = "Year of observation",
         y = "Number of individuals") +
    theme_bw()
```



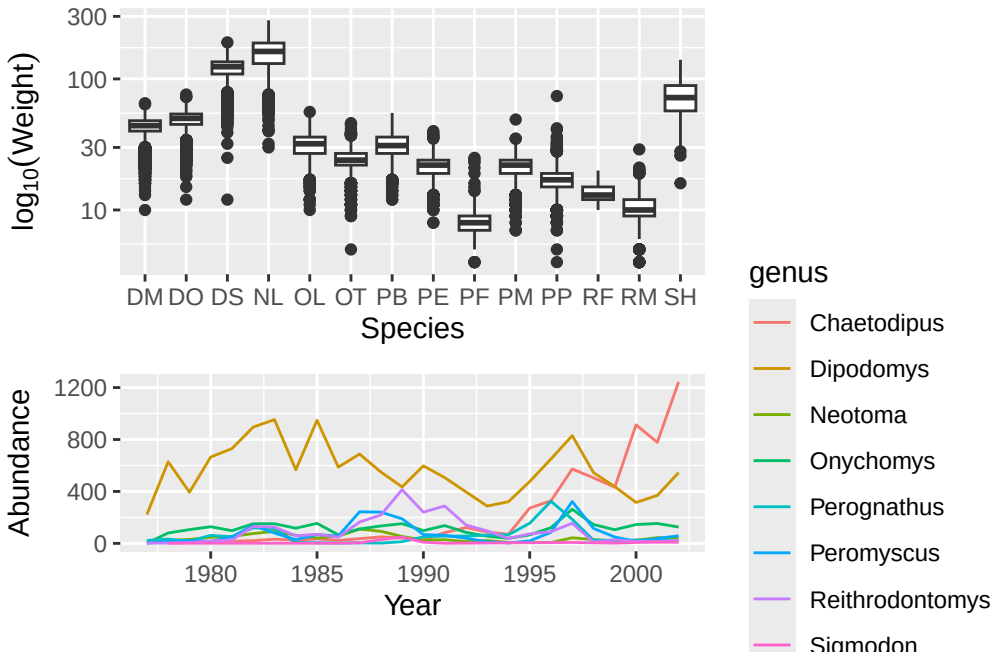

```
## Increase font size by generic theme() function
ggplot(data = yearly_sex_counts, mapping = aes(x = year, y = n, color = sex)) +
    geom_line() +
    facet_wrap(vars(genus)) +
    labs(title = "Observed genera through time",
        x = "Year of observation",
        y = "Number of individuals") +
    theme_bw() +
    theme(text=element_text(size = 16))
```



Arranging plots

```
#install.packages("patchwork")
library(patchwork)
```

```
plot_weight <- ggplot(data = surveys_complete, aes(x = species_id, y = weight)) +
  geom_boxplot() +
  labs(x = "Species", y = expression(log[10](Weight))) +
  scale_y_log10()

plot_count <- ggplot(data = yearly_counts, aes(x = year, y = n, color = genus)) +
  geom_line() +
  labs(x = "Year", y = "Abundance")

plot_weight / plot_count + plot_layout(heights = c(3, 2))
```



Exporting Plots

```
my_plot <- ggplot(data = yearly_sex_counts,
                  aes(x = year, y = n, color = sex)) +
    geom_line() +
    facet_wrap(vars(genus)) +
    labs(title = "Observed genera through time",
        x = "Year of observation",
        y = "Number of individuals") +
    theme_bw() +
    theme(axis.text.x = element_text(colour = "grey20", size = 12, angle = 90,
                                     hjust = 0.5, vjust = 0.5),
          axis.text.y = element_text(colour = "grey20", size = 12),
          text = element_text(size = 16))

ggsave("name_of_file.png", my_plot, width = 15, height = 10)
```

```
## This also works for plots combined with patchwork
plot_combined <- plot_weight / plot_count + plot_layout(heights = c(3, 2))
ggsave("plot_combined.png", plot_combined, width = 10, dpi = 300)
```

```
Saving 10 x 3.5 in image
```