

data-visualisation-with-ggplot2.R

RetailAdmin

2020-07-14

```
# install.packages('hexbin')

library('tidyverse')

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr    0.3.4
## v tibble   3.0.2     v dplyr    1.0.0
## v tidyr    1.1.0     v stringr  1.4.0
## v readr    1.3.1     vforcats  0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()

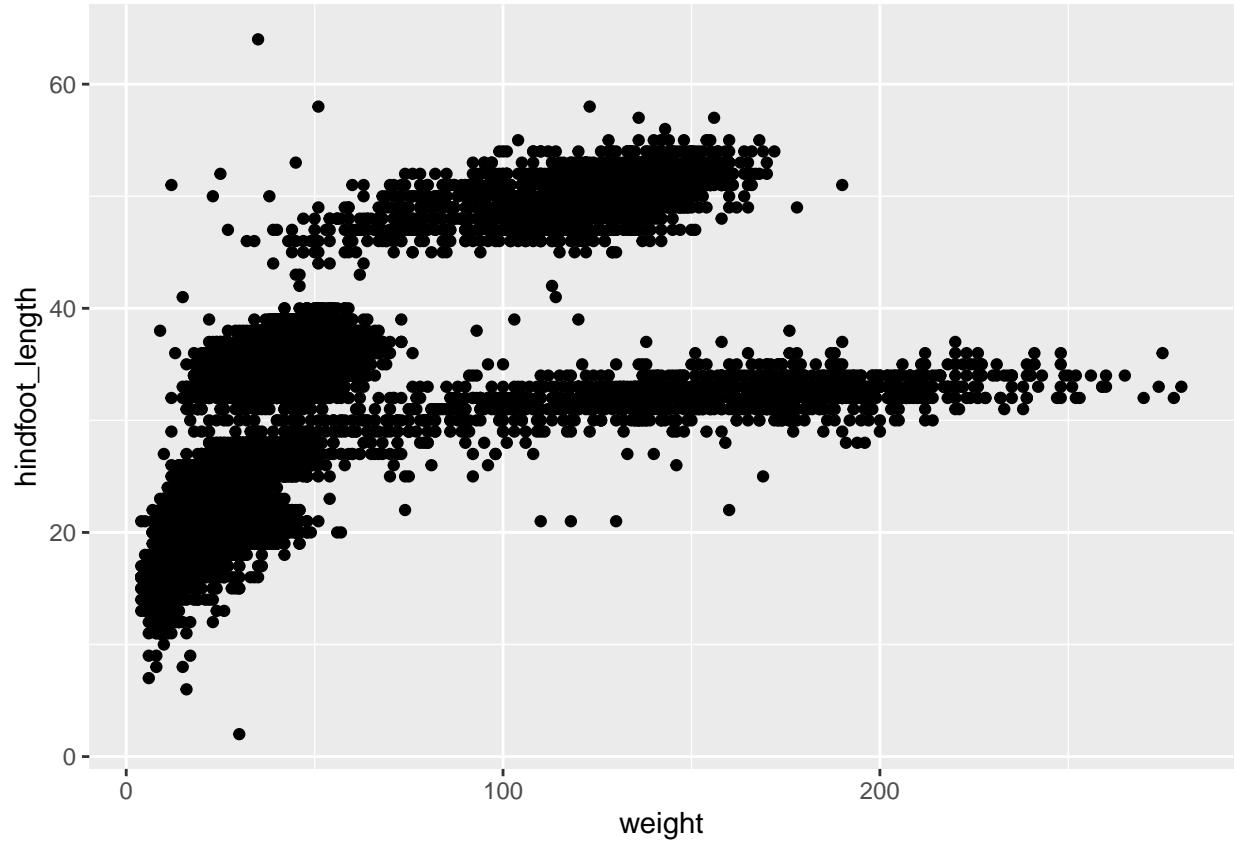
surveys_complete <- read_csv('data/surveys_complete.csv')

## Parsed with column specification:
## cols(
##   record_id = col_double(),
##   month = col_double(),
##   day = col_double(),
##   year = col_double(),
##   plot_id = col_double(),
##   species_id = col_character(),
##   sex = col_character(),
##   hindfoot_length = col_double(),
##   weight = col_double(),
##   genus = col_character(),
##   species = col_character(),
##   taxa = col_character(),
##   plot_type = col_character()
## )

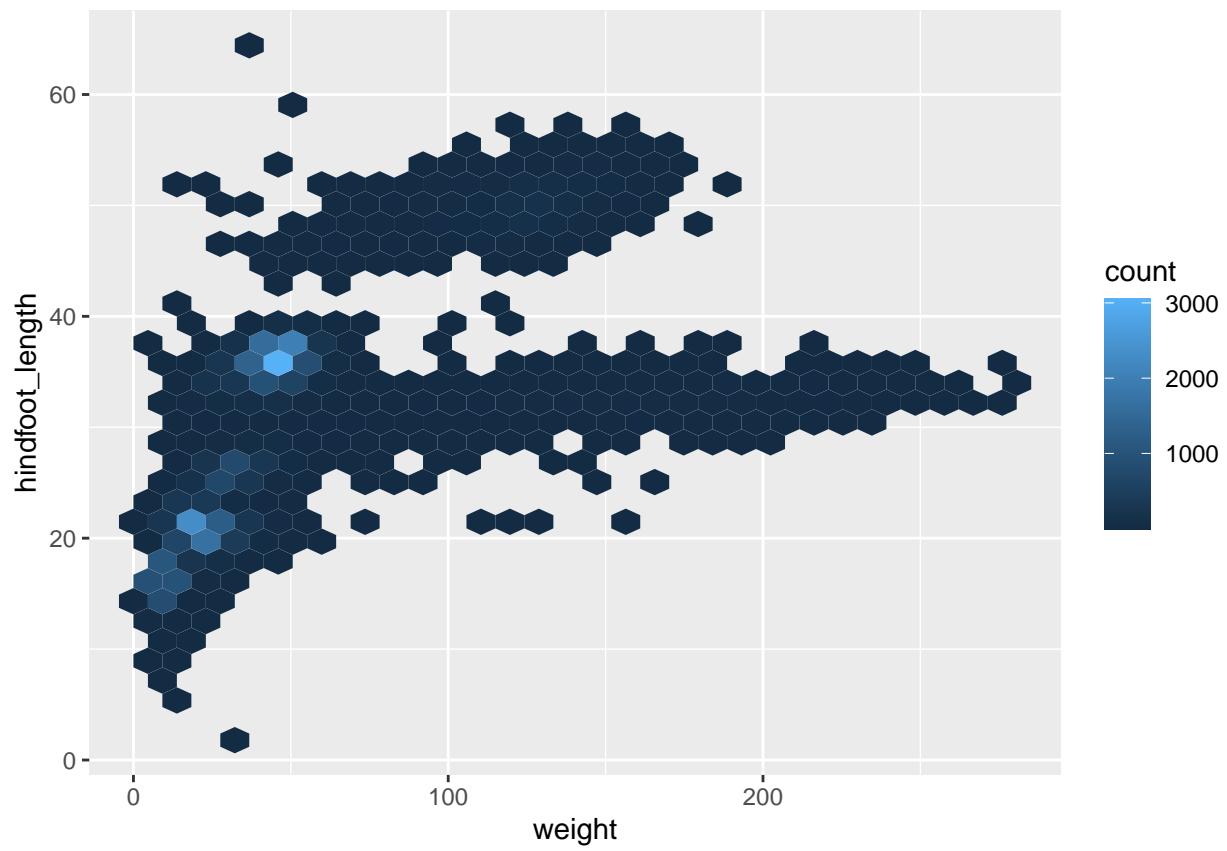
View(surveys_complete)

surveys_plot <- ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length))

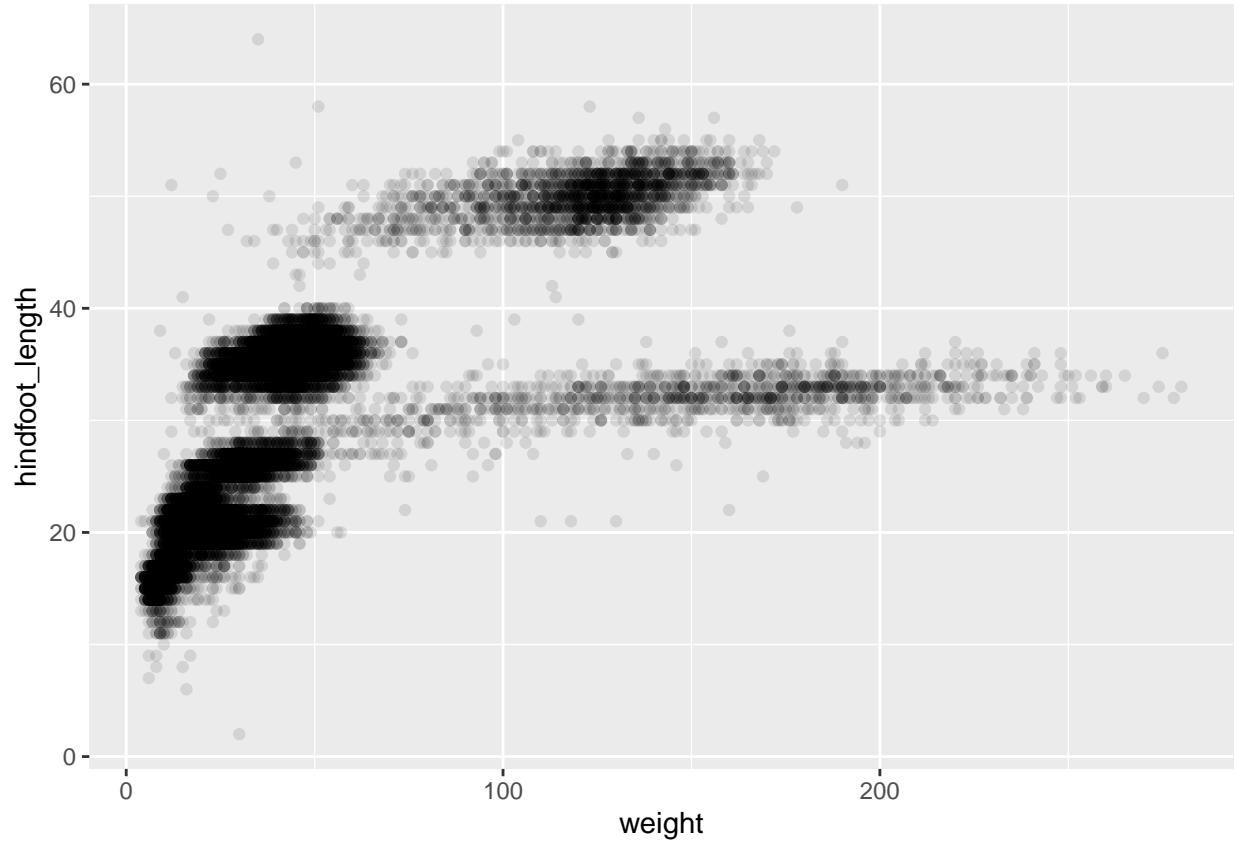
surveys_plot + geom_point()
```



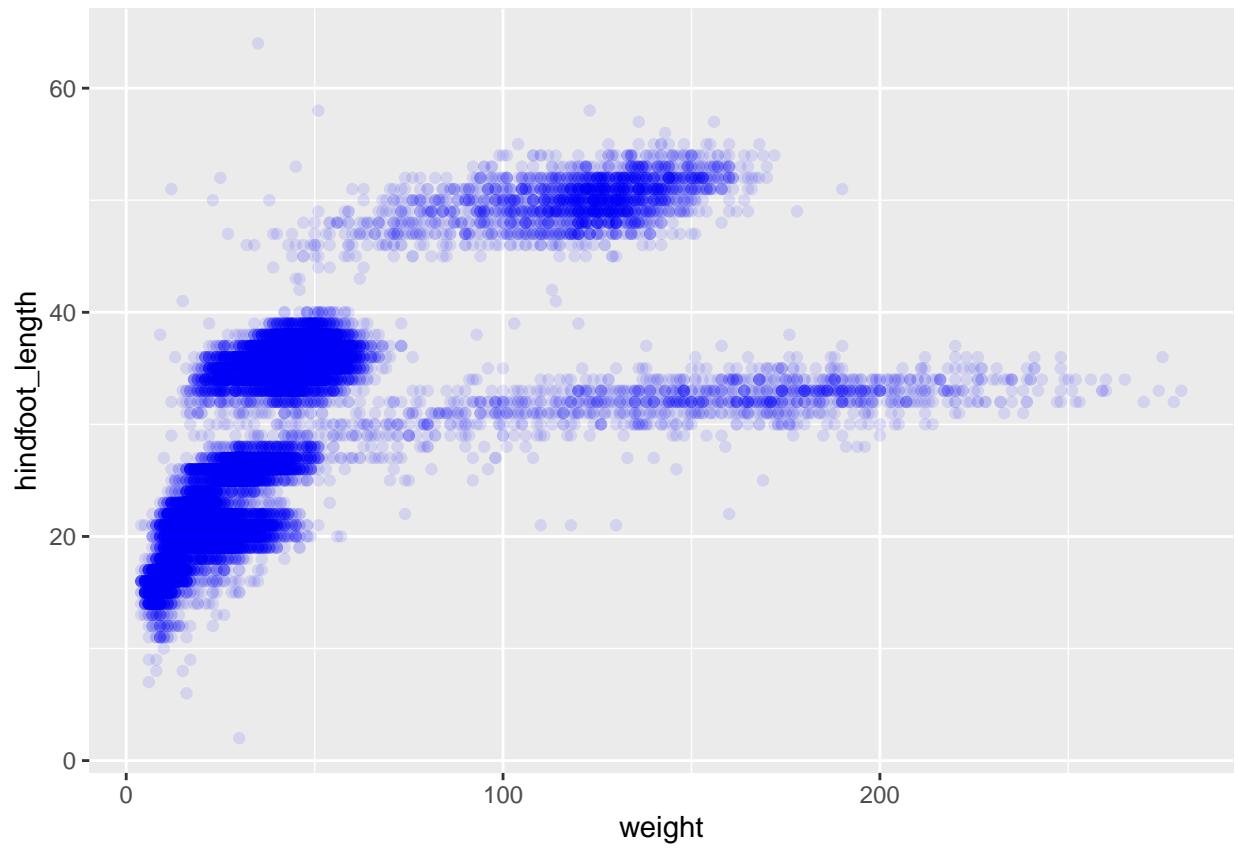
```
library('hexbin')  
surveys_plot + geom_hex()
```



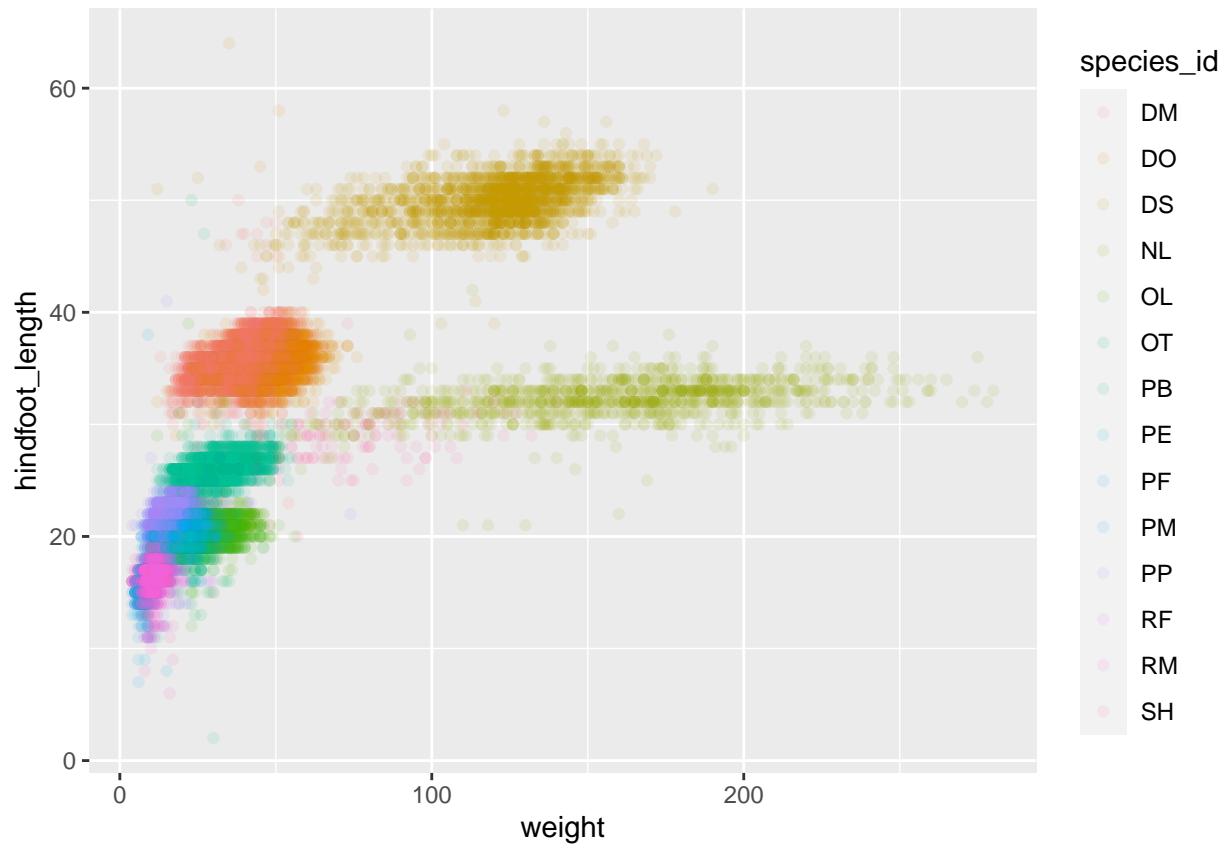
```
surveys_plot + geom_point(alpha = 0.1)
```



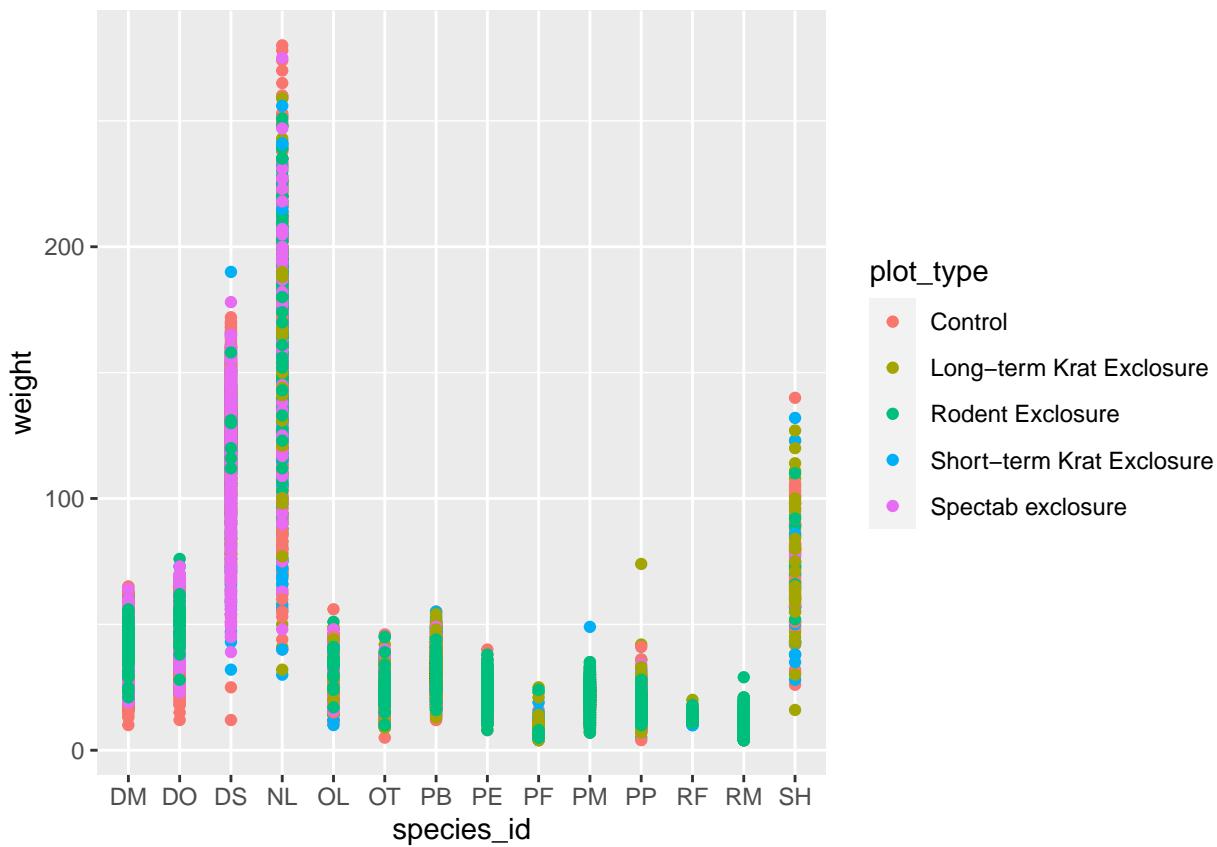
```
surveys_plot + geom_point(alpha = 0.1, colour = 'blue')
```



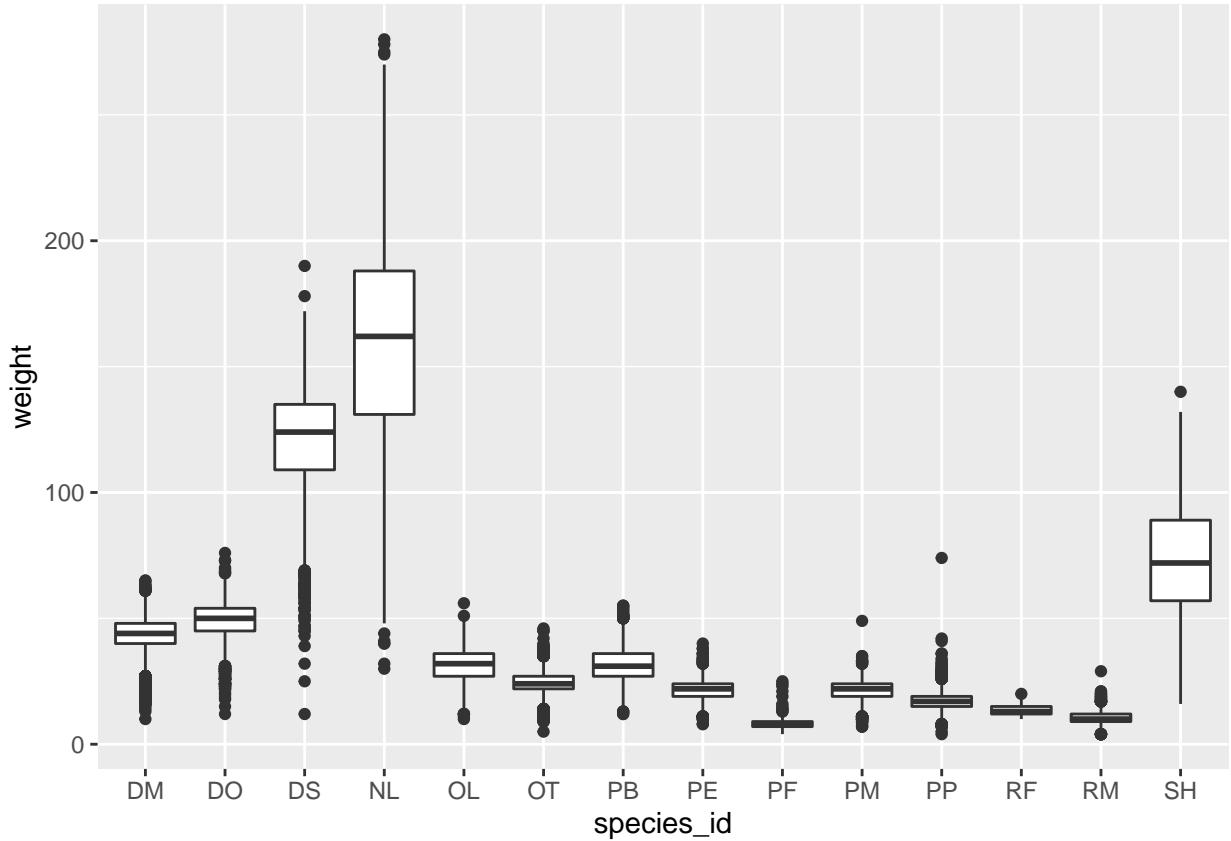
```
surveys_plot + geom_point(alpha = 0.1, aes(colour = species_id))
```



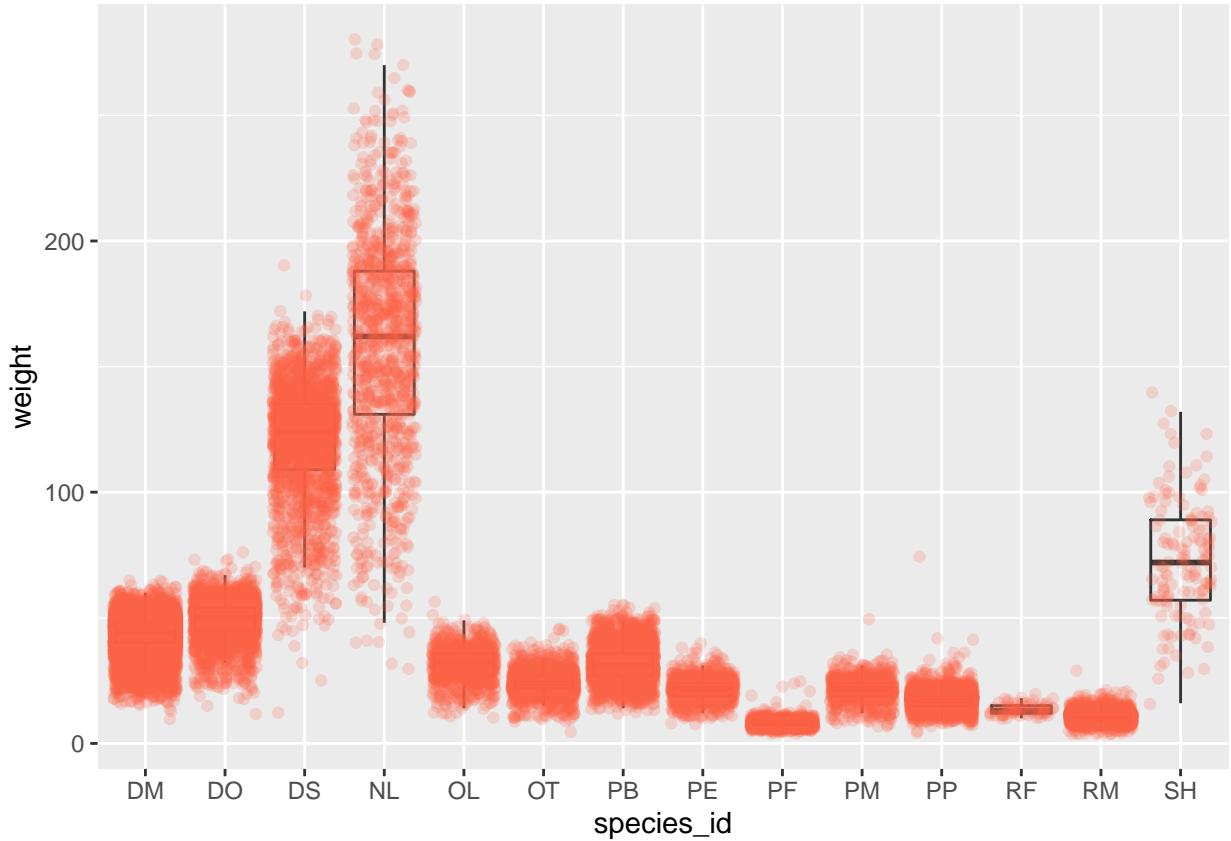
```
ggplot(data = surveys_complete,
       mapping = aes(x = species_id, y = weight)) +
  geom_point(aes(colour = plot_type))
```



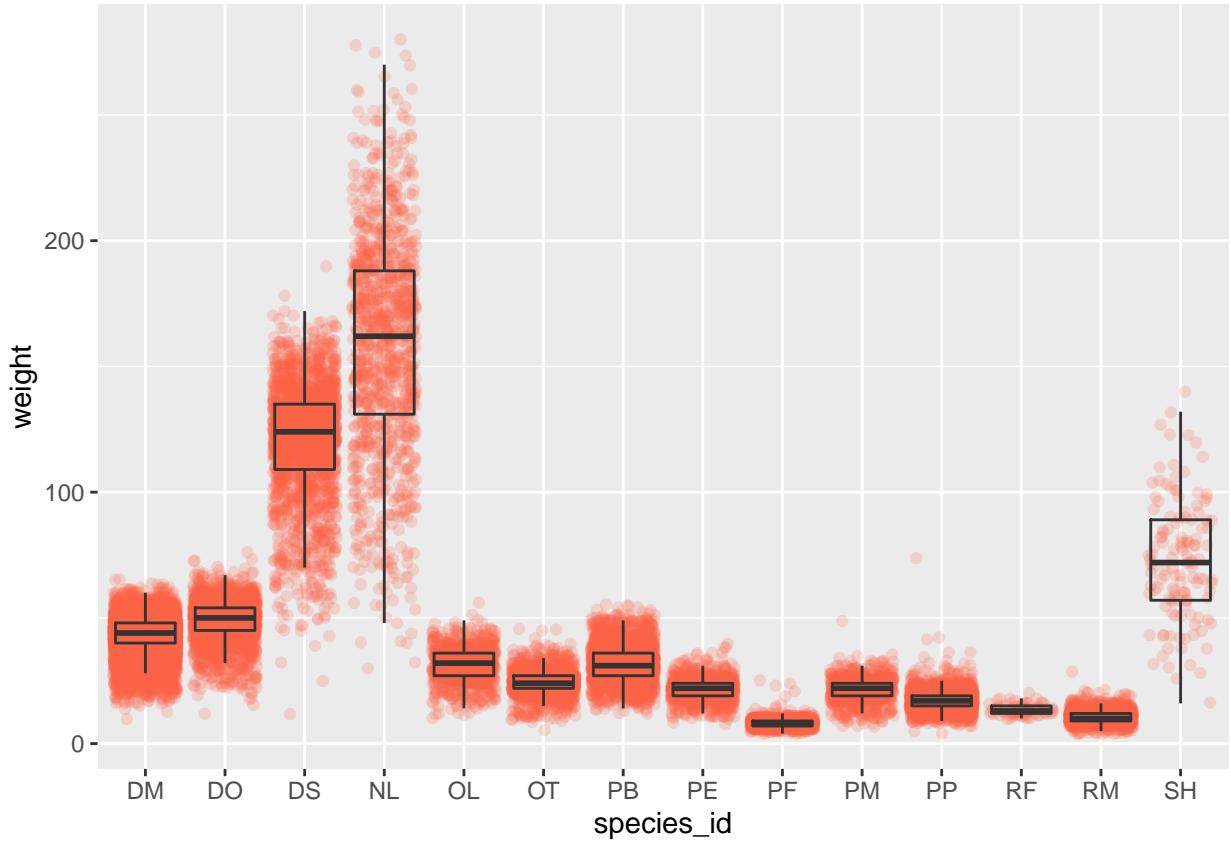
```
ggplot(data = surveys_complete,
       mapping = aes(x = species_id, y = weight)) +
  geom_boxplot()
```



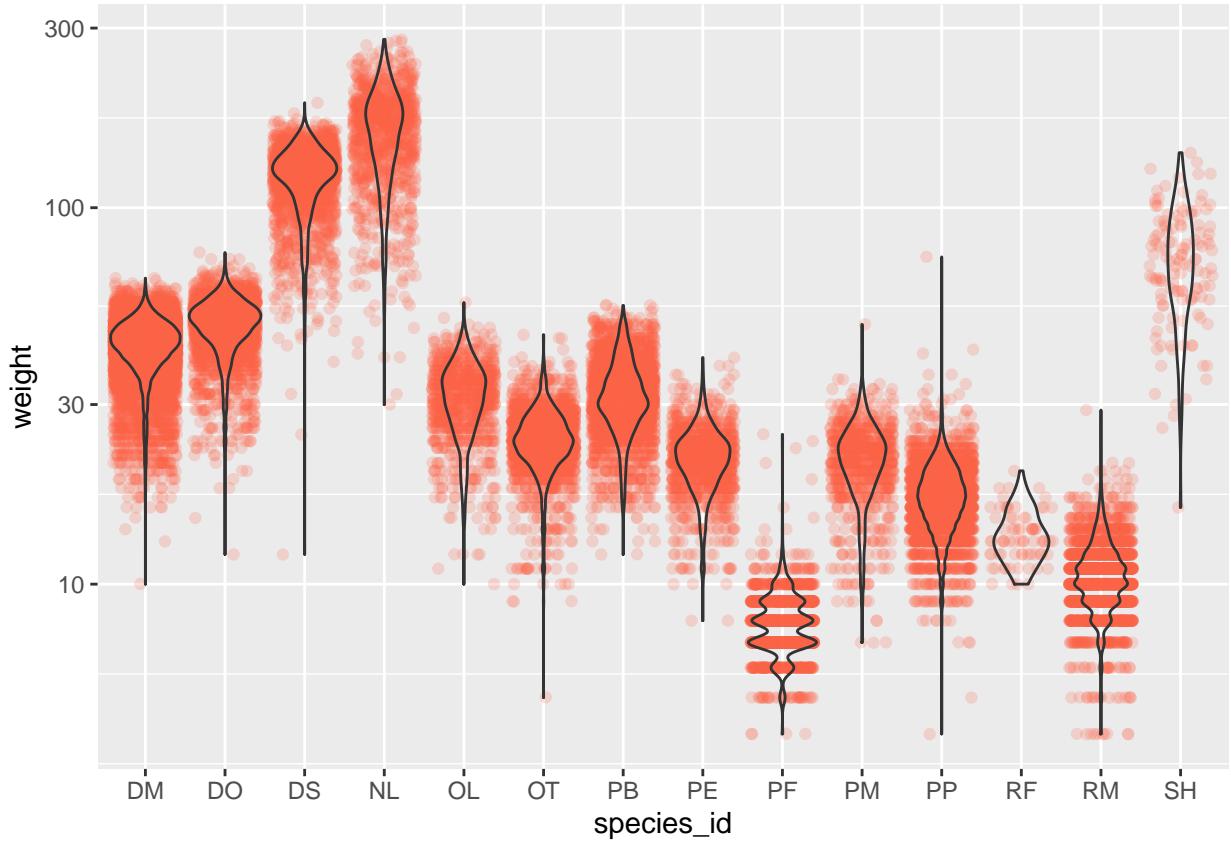
```
ggplot(data = surveys_complete,
       mapping = aes(x = species_id, y = weight)) +
  geom_boxplot(alpha = 0) + # alpha = 0 means transparent, the outliers have been invisible
  geom_jitter(alpha = 0.2, colour = 'tomato')
```



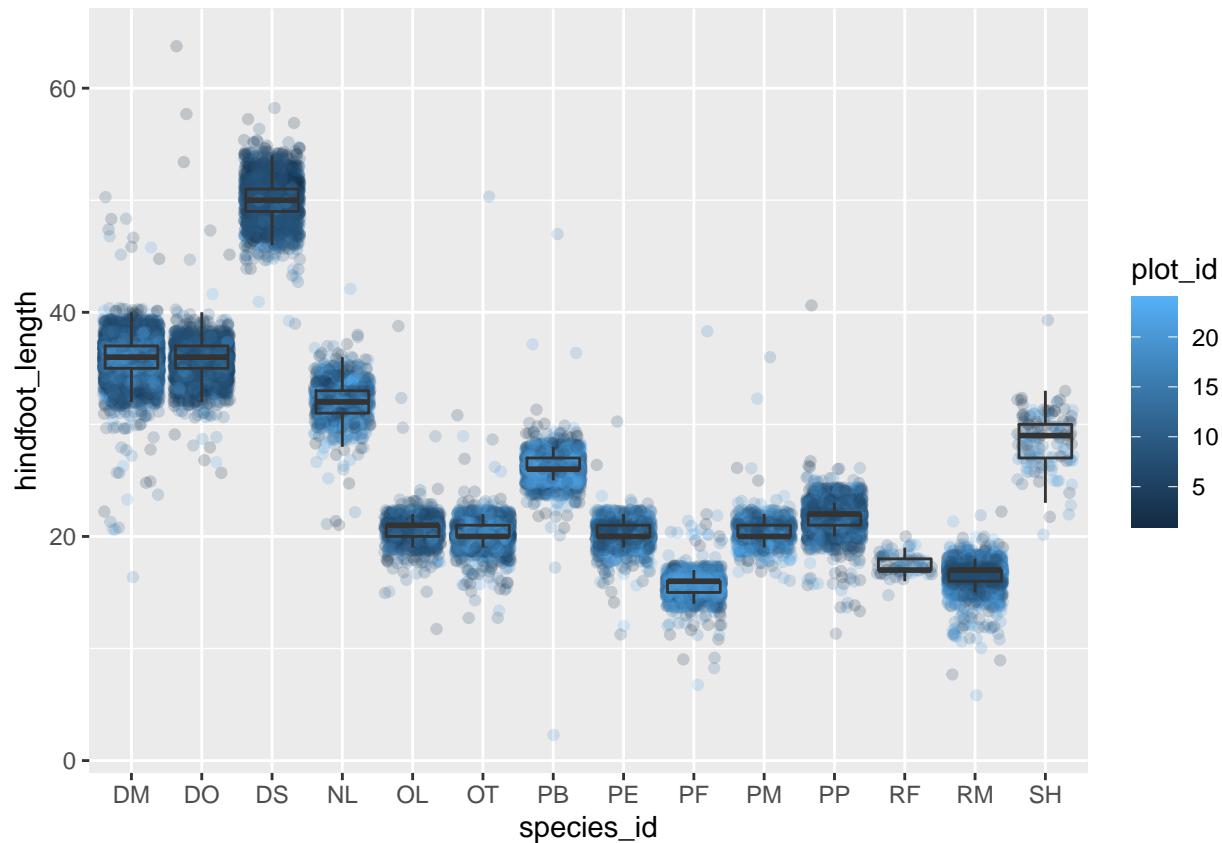
```
ggplot(data = surveys_complete,
       mapping = aes(x = species_id, y = weight)) +
  geom_jitter(alpha = 0.2, colour = 'tomato') +
  geom_boxplot(alpha = 0)
```



```
ggplot(data = surveys_complete,
       mapping = aes(x = species_id, y = weight)) +
  geom_jitter(alpha = 0.2, colour = 'tomato') +
  geom_violin(alpha = 0) +
  scale_y_log10()
```



```
ggplot(data = surveys_complete,
       mapping = aes(x = species_id, y = hindfoot_length)) +
  geom_jitter(alpha = 0.2, aes(colour = plot_id)) +
  geom_boxplot(alpha = 0)
```



```

class(surveys_complete$plot_id)

## [1] "numeric"

str(surveys_complete$plot_id)

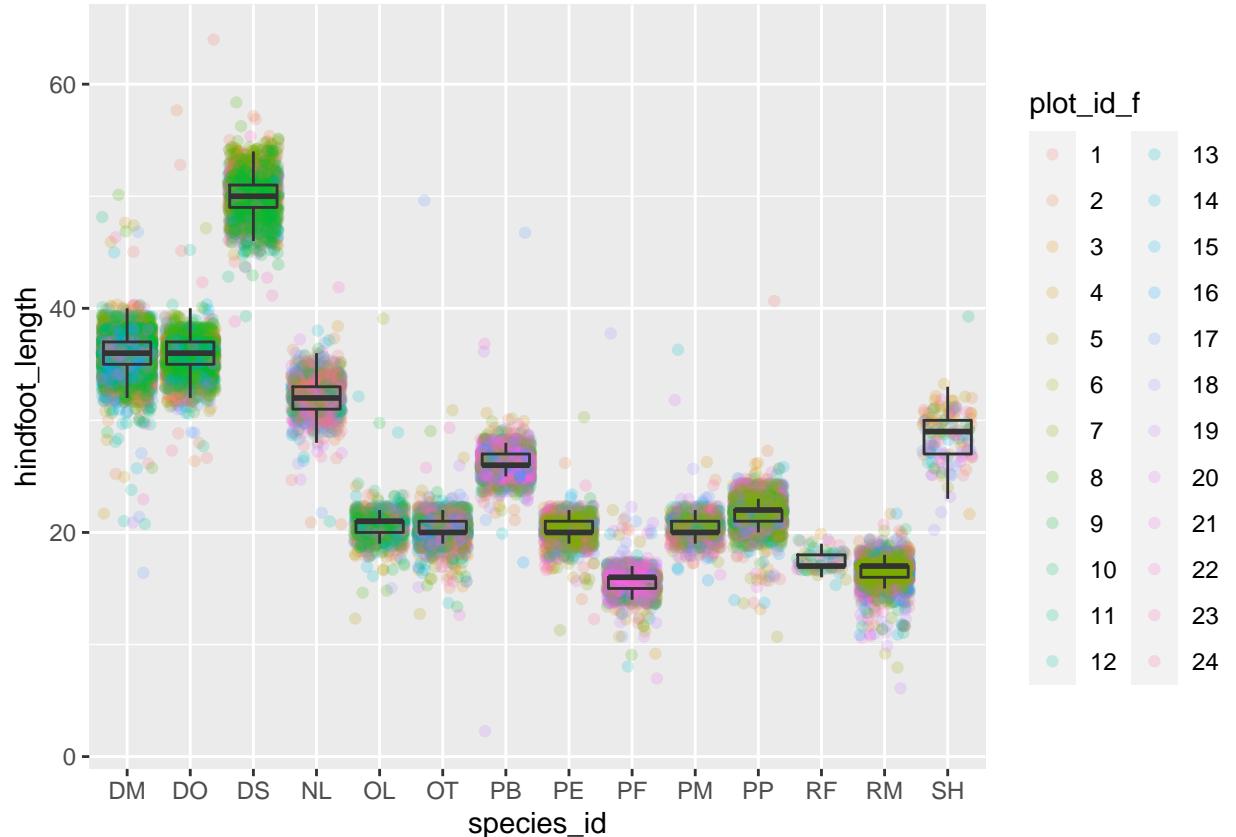
##  num [1:30463] 2 2 2 2 2 2 2 2 2 2 ...

surveys_complete$plot_id_f <- as.factor(surveys_complete$plot_id)
str(surveys_complete$plot_id_f)

##  Factor w/ 24 levels "1","2","3","4",...: 2 2 2 2 2 2 2 2 2 2 ...

ggplot(data = surveys_complete,
       mapping = aes(x = species_id, y = hindfoot_length)) +
  geom_jitter(alpha = 0.2, aes(colour = plot_id_f)) +
  geom_boxplot(alpha = 0)

```

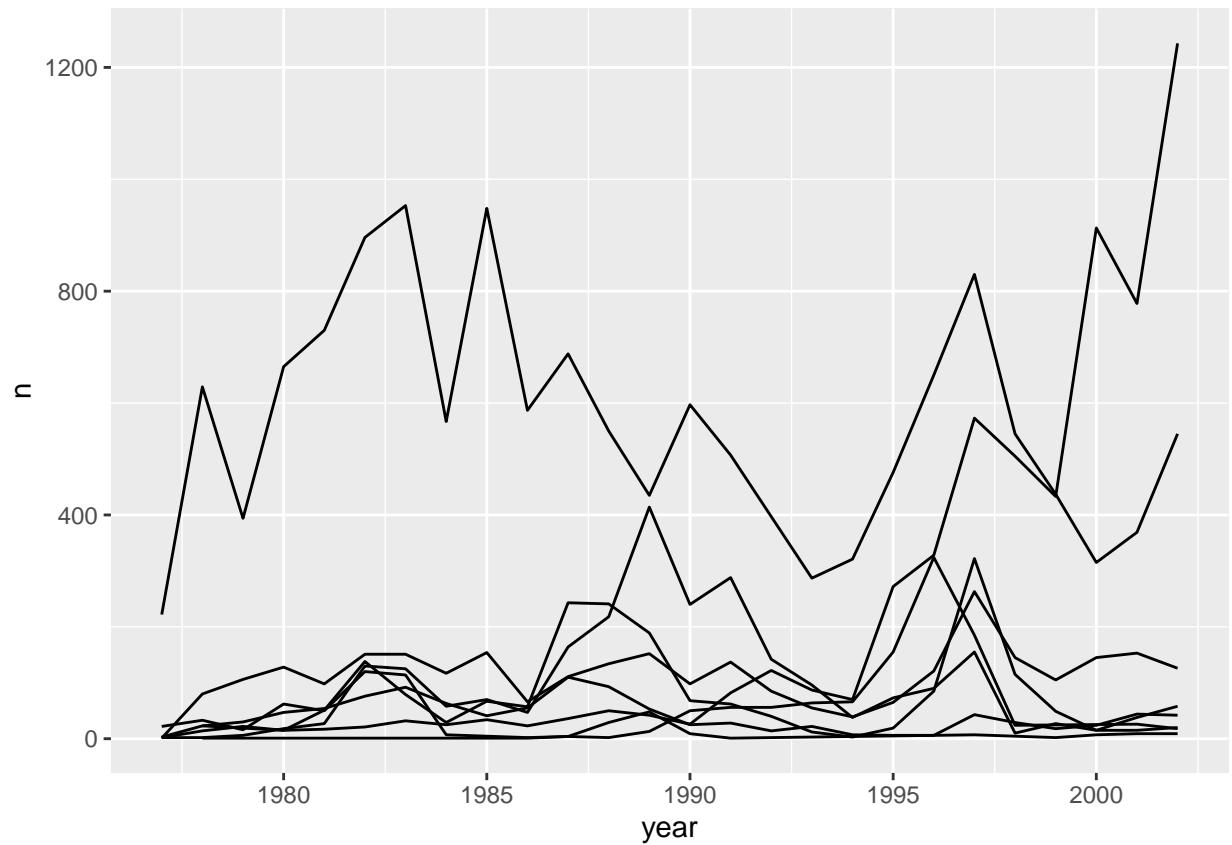


```

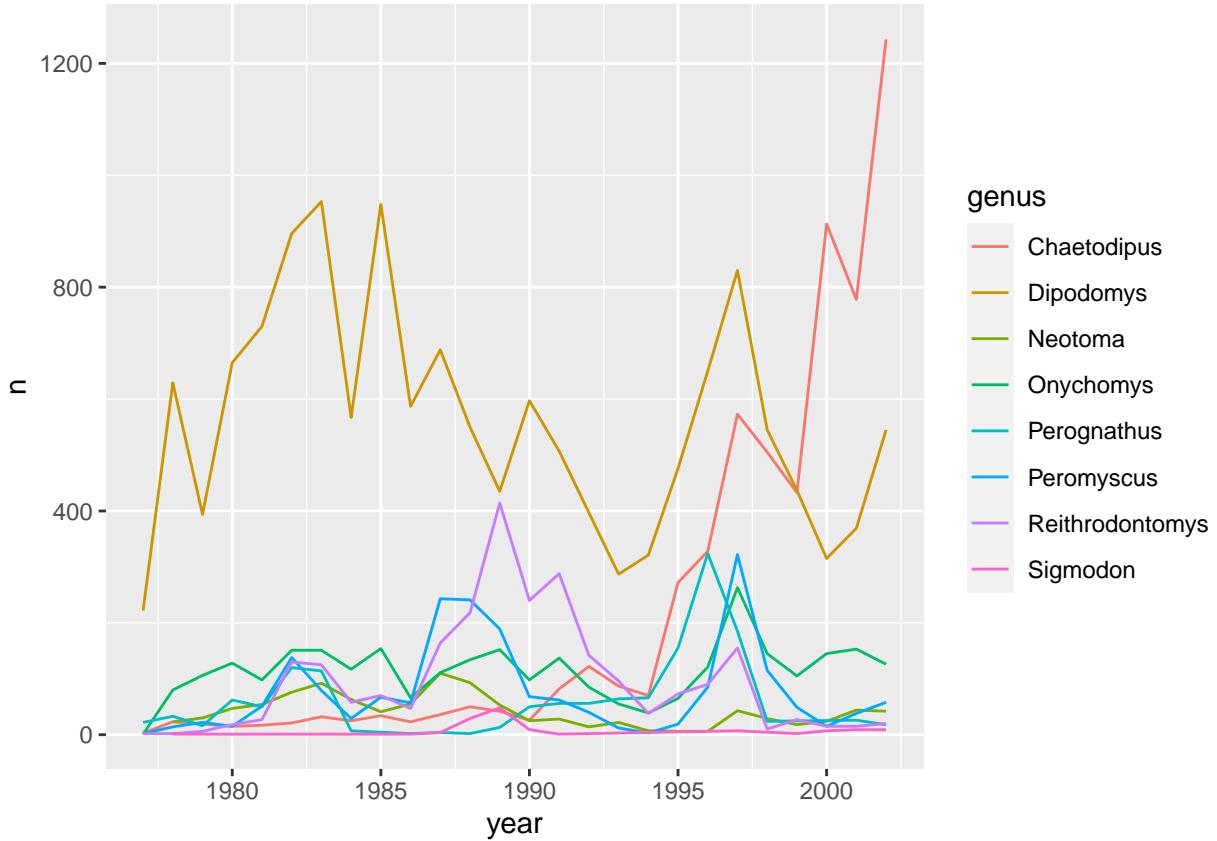
yearly_counts <- surveys_complete %>%
  count(year, genus)
View(yearly_counts)

ggplot(data = yearly_counts,
       mapping = aes(x = year, y = n, group = genus)) +
  geom_line()

```

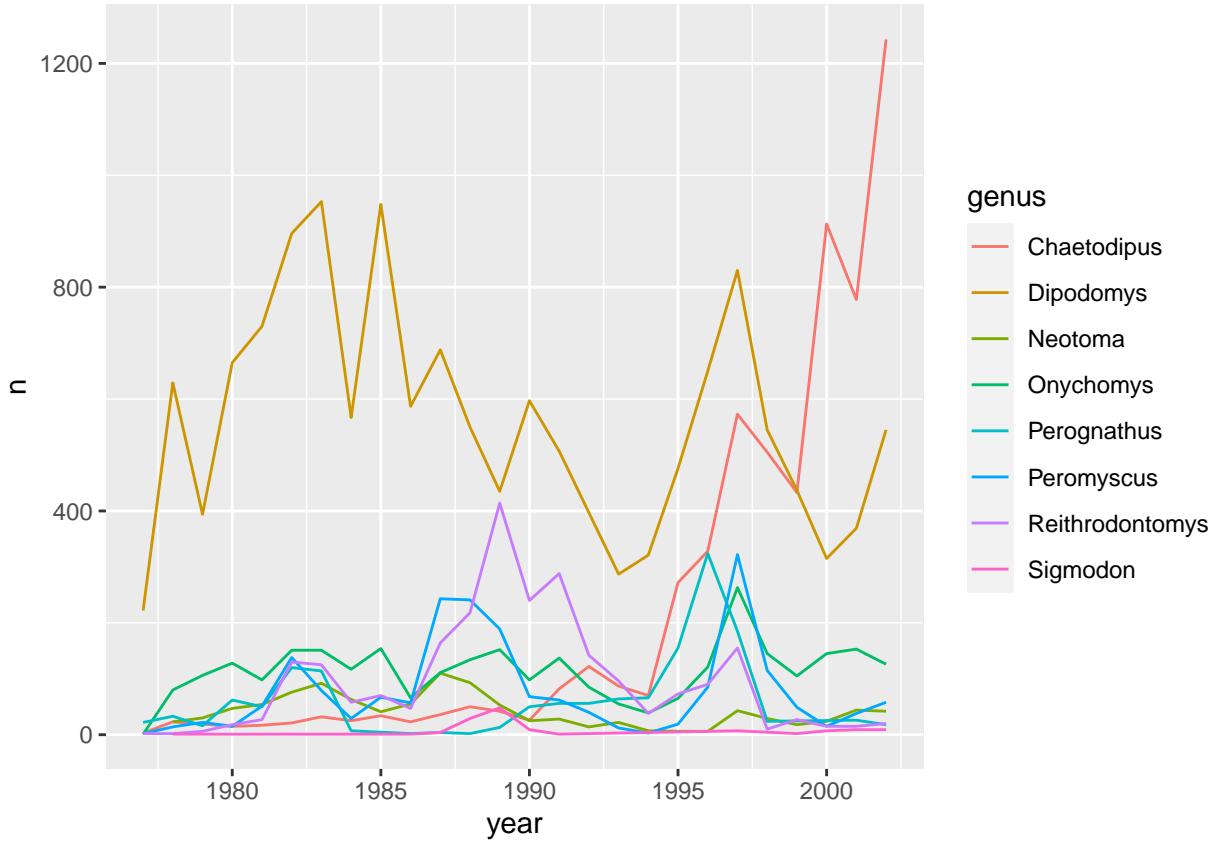


```
ggplot(data = yearly_counts,
       mapping = aes(x = year, y = n, colour = genus)) +
  geom_line()
```

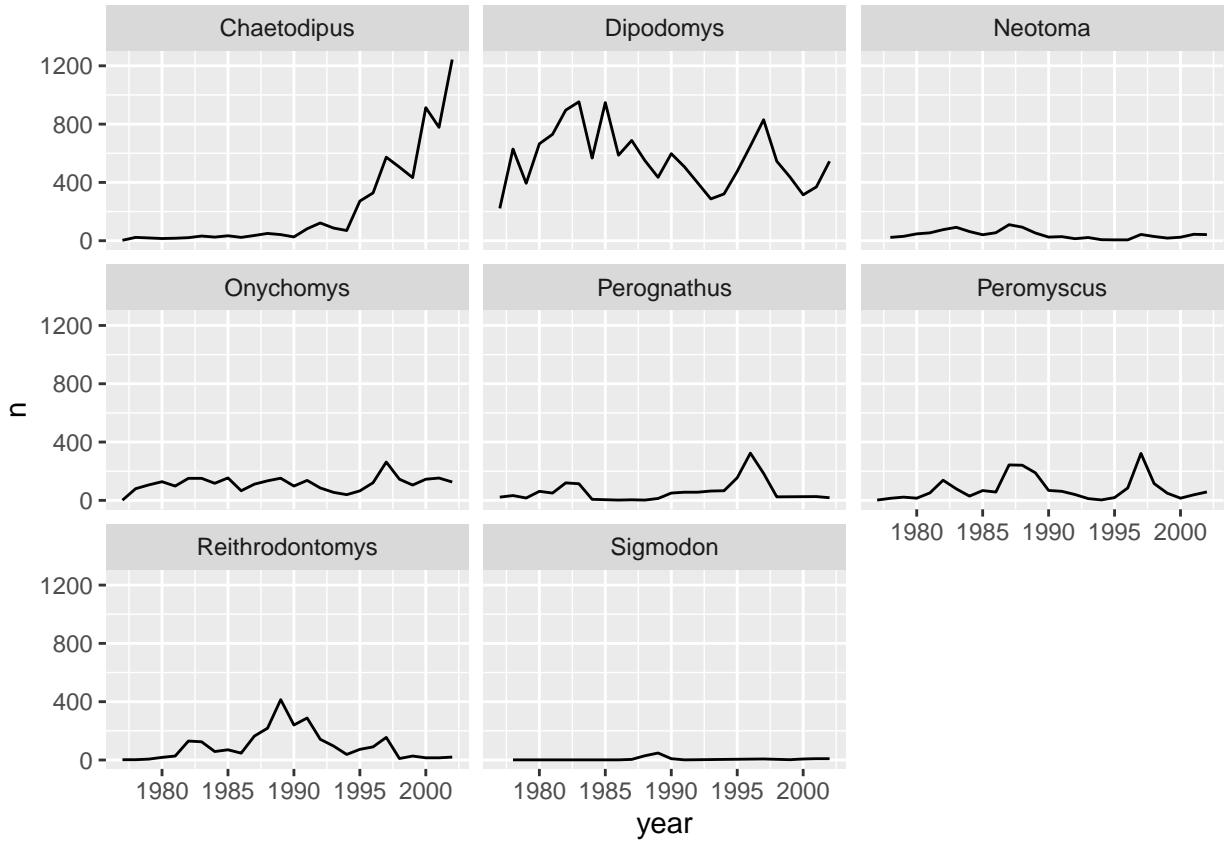


```
# Pipe can be used to produce plots
yearly_counts_graph <- surveys_complete %>%
  count(year, genus) %>%
  ggplot(mapping = aes(x = year, y = n, colour = genus)) +
  geom_line()

yearly_counts_graph
```



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

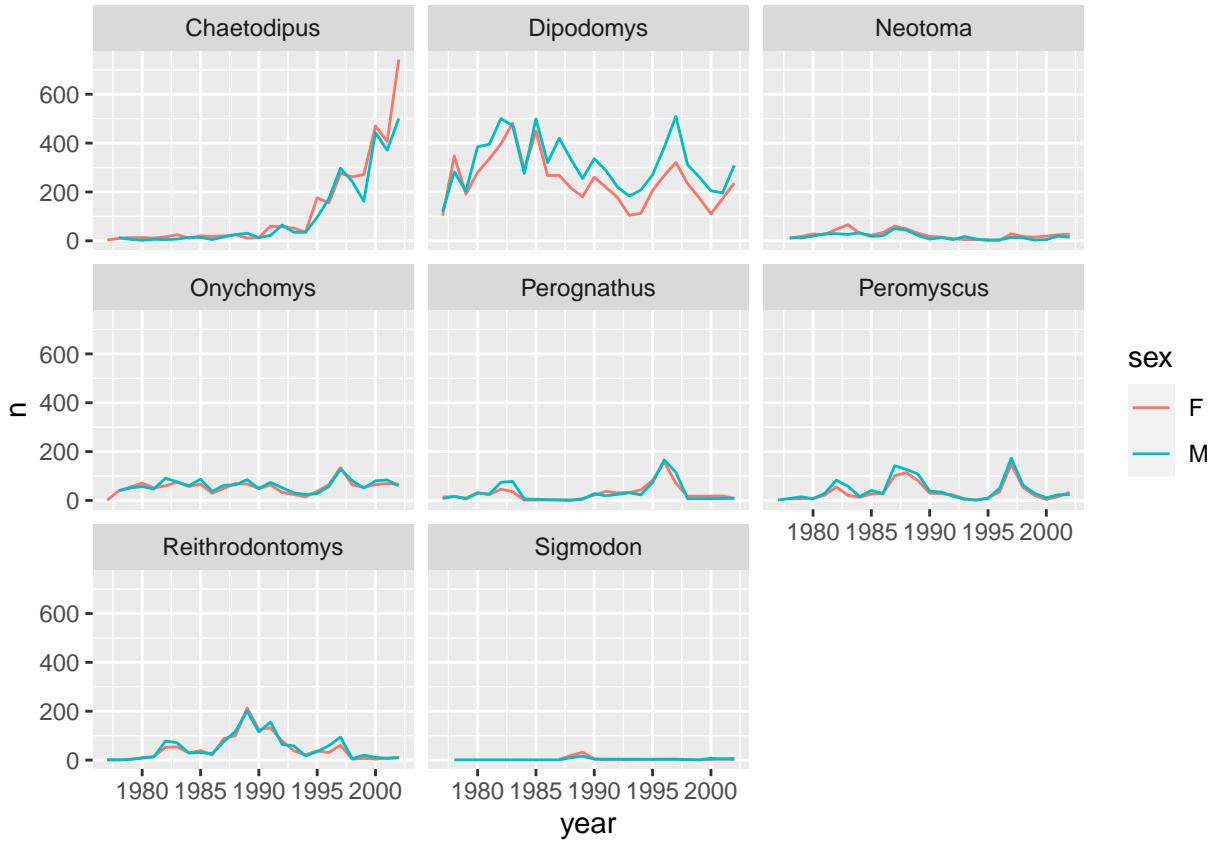


```

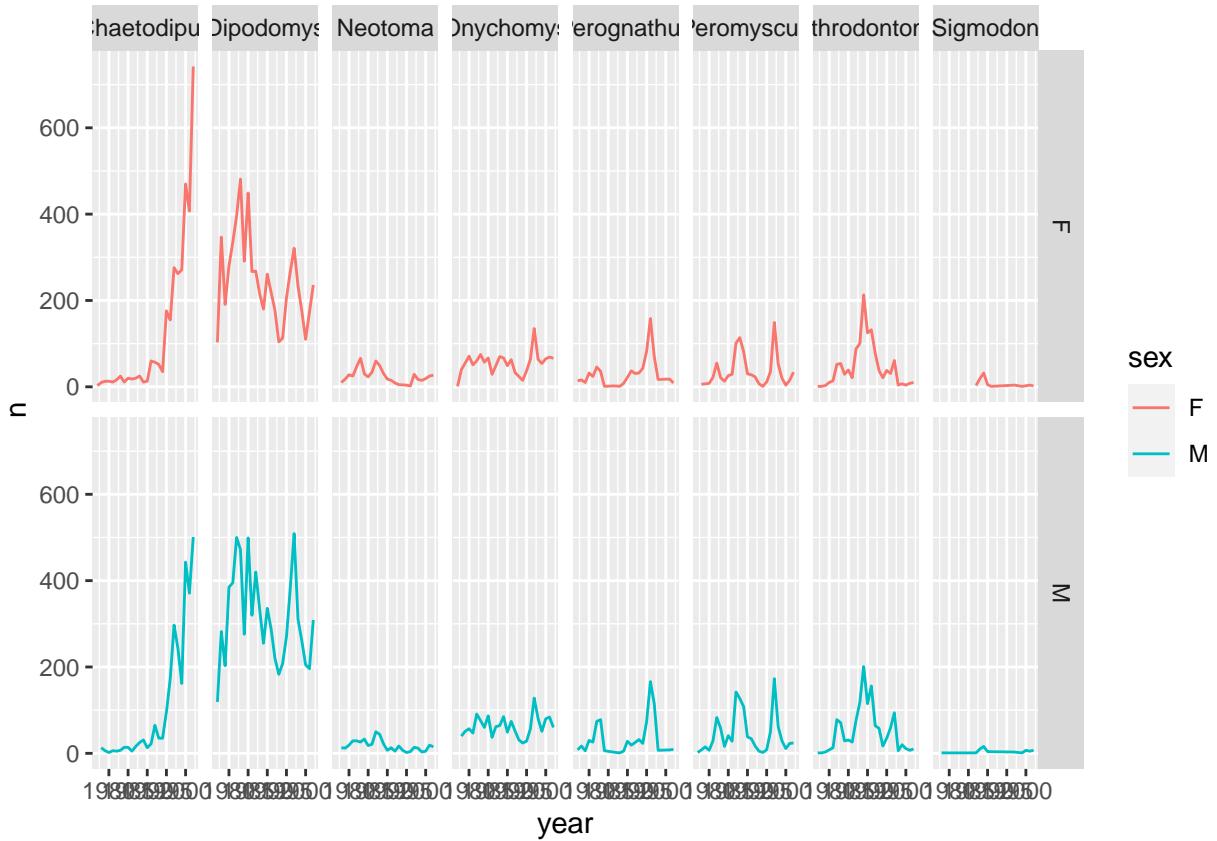
yearly_sex_counts <- surveys_complete %>%
  count(year, genus, sex)

ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, colour = sex)) +
  geom_line() +
  facet_wrap(facets = vars(genus))

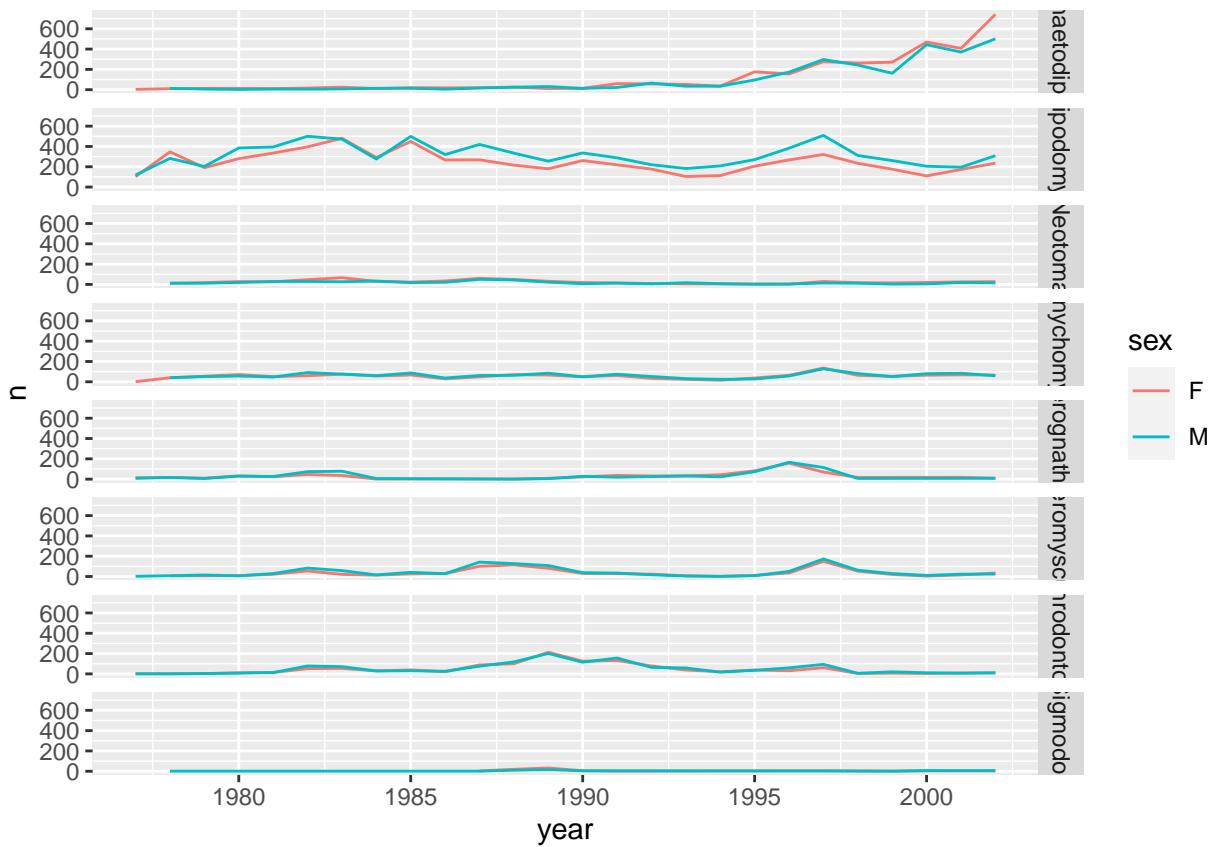
```



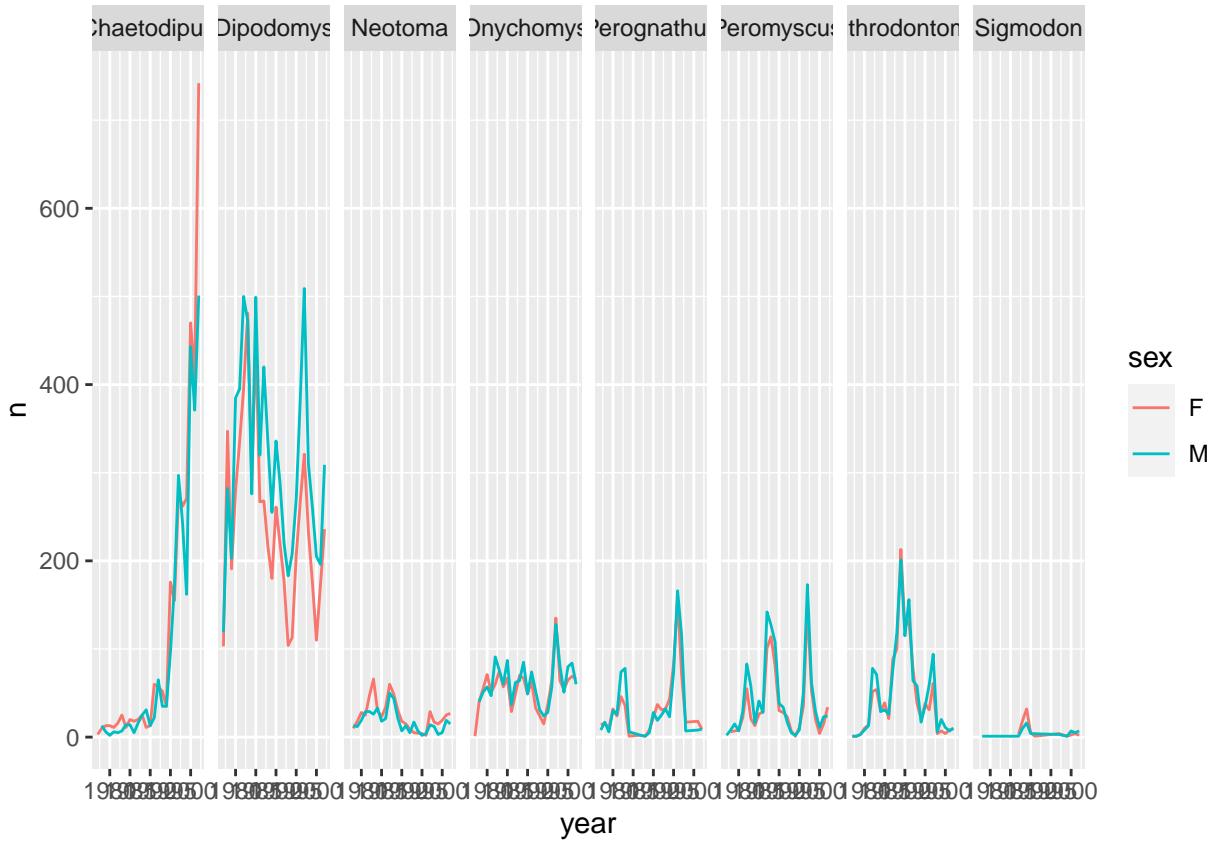
```
ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, colour = sex)) +
  geom_line() +
  facet_grid(rows = vars(sex), cols = vars(genus))
```



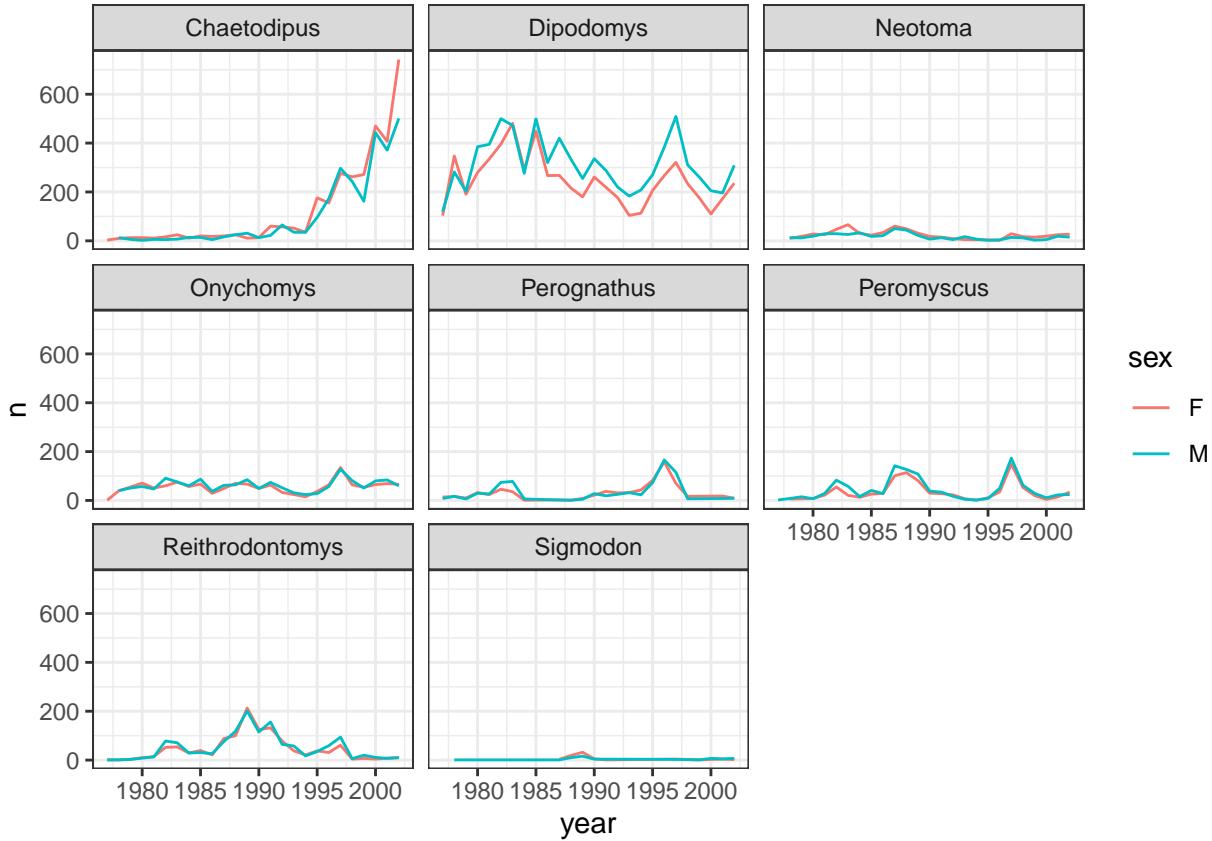
```
ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, colour = sex)) +
  geom_line() +
  facet_grid(rows = vars(genus))
```



```
ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, colour = sex)) +
  geom_line() +
  facet_grid(cols = vars(genus))
```



```
ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, colour = sex)) +
  geom_line() +
  facet_wrap(facets = vars(genus)) +
  theme_bw()
```

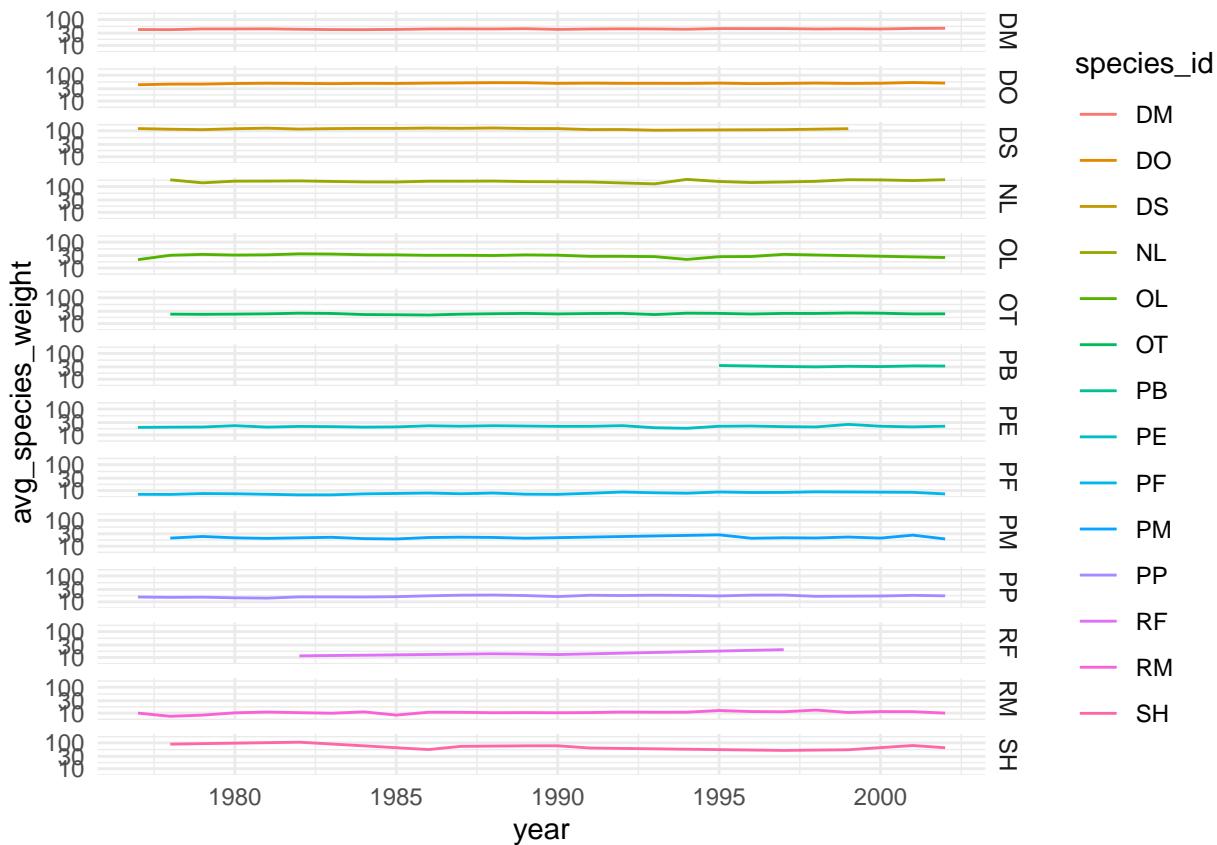


```

surveys_complete %>%
  group_by(year, species_id) %>%
  summarise(avg_species_weight = mean(weight)) %>%
  ggplot(mapping = aes(x = year, y = avg_species_weight, colour = species_id)) +
  geom_line() +
  facet_grid(rows = vars(species_id)) +
  theme_minimal() +
  scale_y_log10()

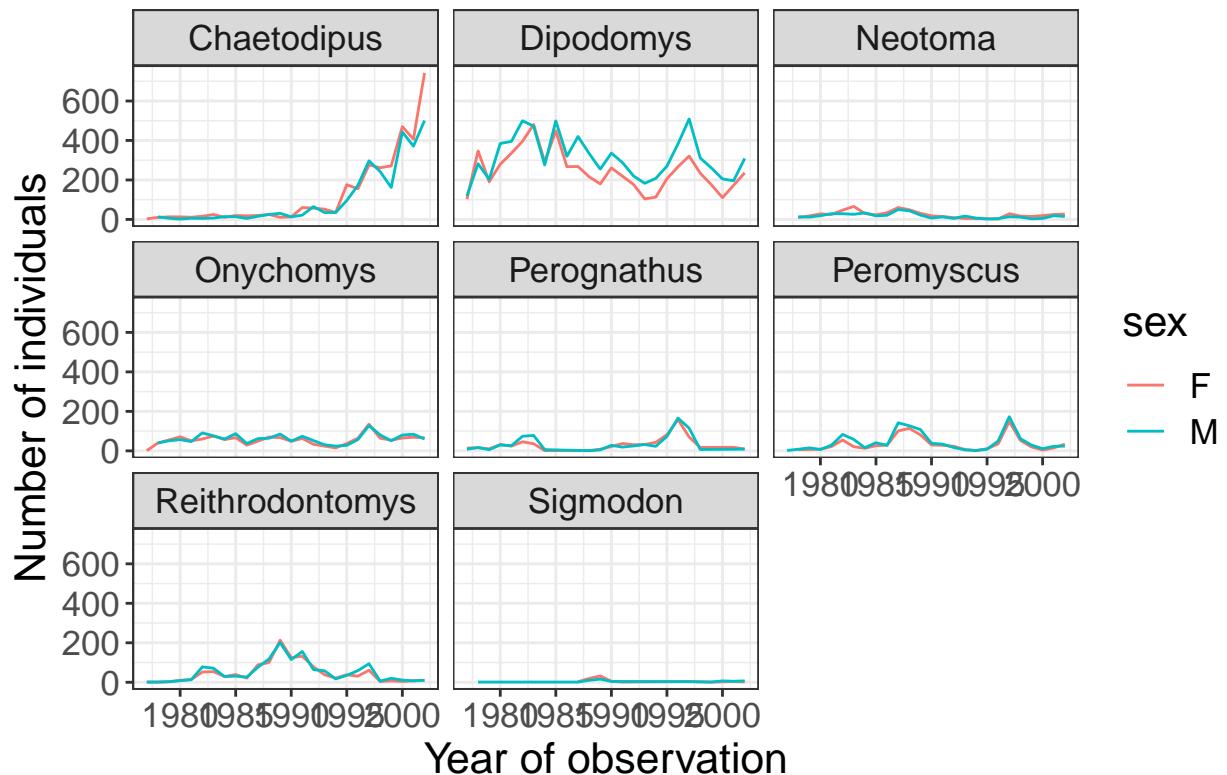
## `summarise()` regrouping output by 'year' (override with '.groups' argument)

```



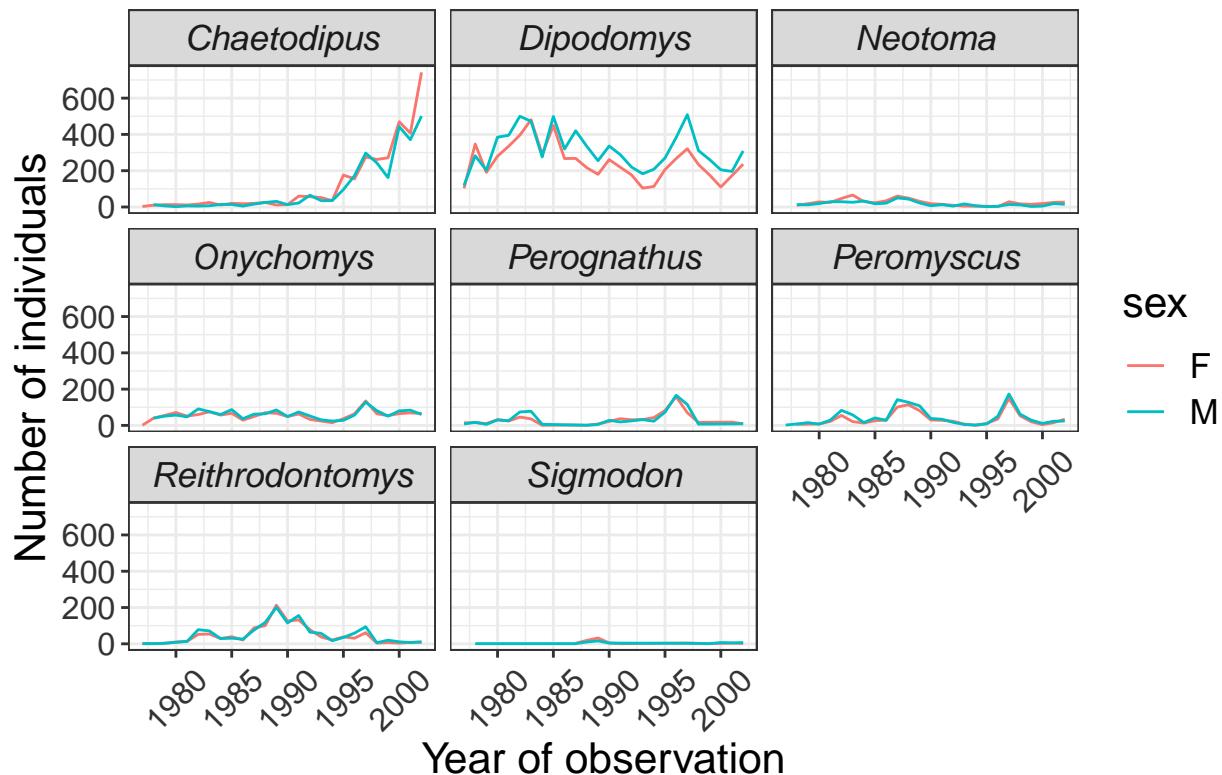
```
ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, colour = sex)) +
  geom_line() +
  facet_wrap(facets = vars(genus)) +
  labs(x = 'Year of observation',
       y = 'Number of individuals',
       title = 'Observed genera through time') +
  theme_bw() +
  theme(text = element_text(size = 16))
```

Observed genera through time



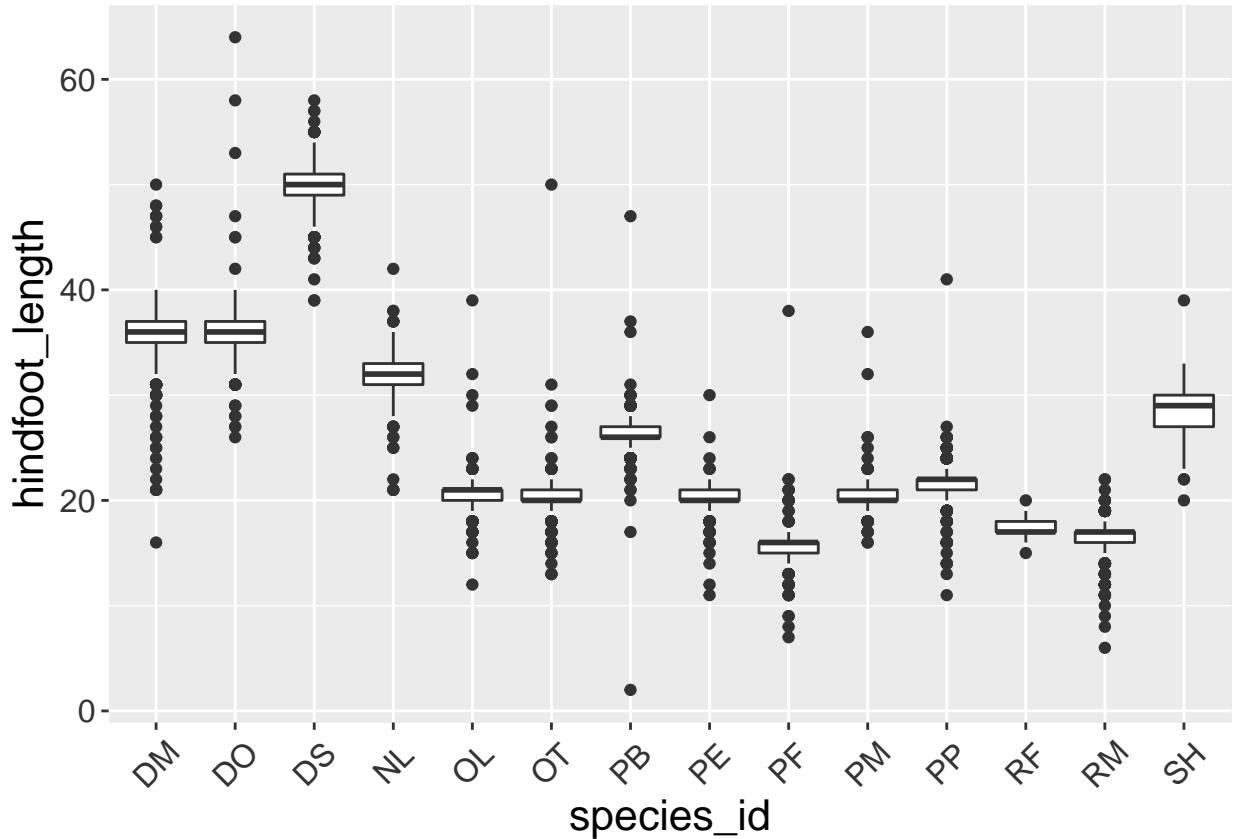
```
ggplot(data = yearly_sex_counts,
       mapping = aes(x = year, y = n, colour = sex)) +
  geom_line() +
  facet_wrap(facets = vars(genus)) +
  labs(x = 'Year of observation',
       y = 'Number of individuals',
       title = 'Observed genera through time') +
  theme_bw() +
  theme(axis.text.x = element_text(colour = 'grey20', size = 12, angle = 45, hjust = 0.5, vjust = 0.5),
        axis.text.y = element_text(colour = 'grey20', size = 12),
        strip.text = element_text(face = 'italic'),
        text = element_text(size = 16))
```

Observed genera through time



```
# if you like the theme you've created, you can save the theme:
grey_theme <- theme(axis.text.x = element_text(colour = 'grey20', size = 12,
                                              angle = 45, hjust = 0.5, vjust = 0.5),
                      axis.text.y = element_text(colour = 'grey20', size = 12),
                      strip.text = element_text(face = 'italic'),
                      text = element_text(size = 16))

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



```
# Combining different plots into a single figure
library(gridExtra)

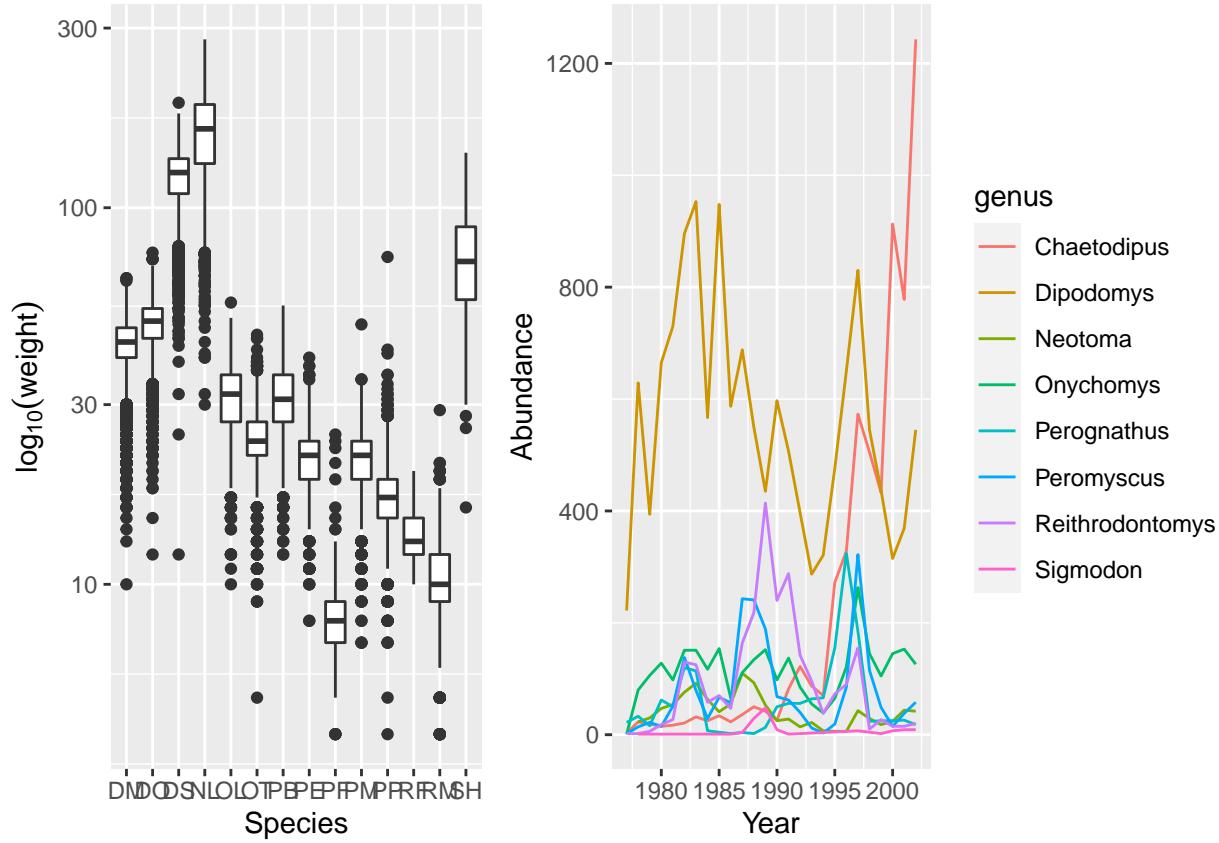
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##      combine

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

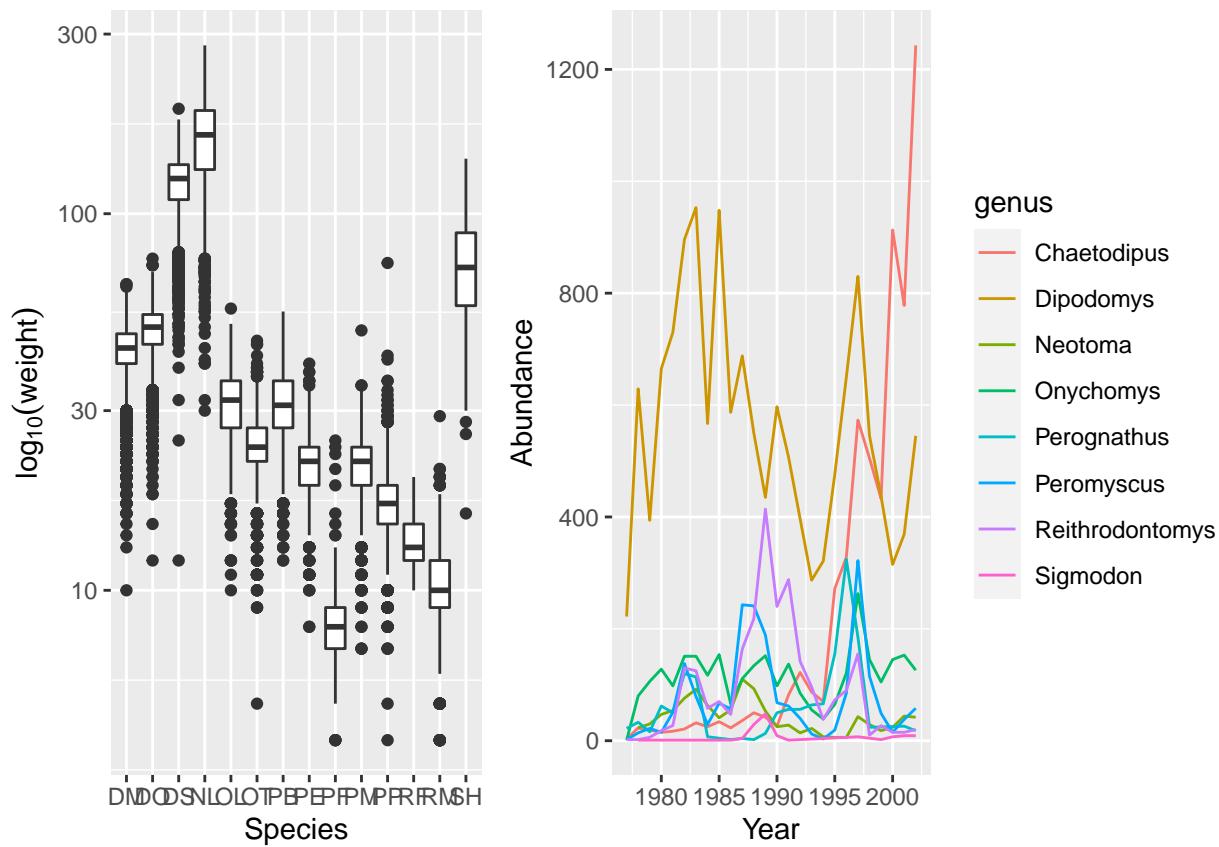
spp_count_plot <- ggplot(data = yearly_counts,
                           aes(x = year, y = n, colour = genus)) +
  geom_line() +
  labs(x = 'Year', y = 'Abundance')

grid.arrange(spp_weight_boxplot, spp_count_plot, ncol = 2, widths = c(4, 6))
```



```
# save plots
my_plot <- ggplot(data = yearly_sex_counts,
                   mapping = aes(x = year, y = n, colour = sex)) +
  geom_line() +
  facet_wrap(facets = vars(genus)) +
  labs(x = 'Year of observation',
       y = 'Number of individuals',
       title = 'Observed genera through time') +
  theme_bw() +
  theme(axis.text.x = element_text(colour = 'grey20', size = 12, angle = 45, hjust = 0.5, vjust = 0.5),
        axis.text.y = element_text(colour = 'grey20', size = 12),
        strip.text = element_text(face = 'italic'),
        text = element_text(size = 16))
ggsave('Genera through time.png', my_plot, width = 15, height = 10)

combo_plot <- grid.arrange(spp_weight_boxplot, spp_count_plot,
                           ncol = 2, widths = c(4, 6))
```



```
ggsave('Weight & Abundance.png', combo_plot, width = 15, height = 10, dpi = 300)
```