Stat7350: Plotting with ggplot2 (lesson 2)

AC Gerstein

2019-03-07

Learning Objectives

- Produce histograms
- Produce bar charts
- Add summary statistics to plots
- Add model fits to plots
- Use different aesthetics purposefully

Pre-analysis workflow: packages & data prep

Packages

Note that `ggplot2` is include in the metapackage `tidyverse`. If you have not previously loaded these libraries you will need to use `install.packages()` to do so before running this code.

```
library(tidyverse)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

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

Load Data

Today we're going to continue using the dataset we used in the last class. If you remember, we already saved it a tidied .csv file, so we can go open that directly without having to reproduce the work.

```
surveys_complete <- read_csv("data_output/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()
## )
```

Challenges from last class

- How do you log10 transform weight? [what other transformations can we do?]
- Replace the box plot with a violin plot; see geom_violin().
- Try making a new plot to explore the distribution of another variable within each species:
  - Create a boxplot for hindfoot_length. Overlay the boxplot layer on a jitter layer to show actual measurements.
  - Add color to the data points on your boxplot according to the plot from which the sample was taken (plot_id).
- Create a plot that depicts how the average weight of each species changes through the years.
- Pick the plot you think is most informative and improve it: change the axes labels, add a title, change the font size or orientation on the x axis, change the theme.

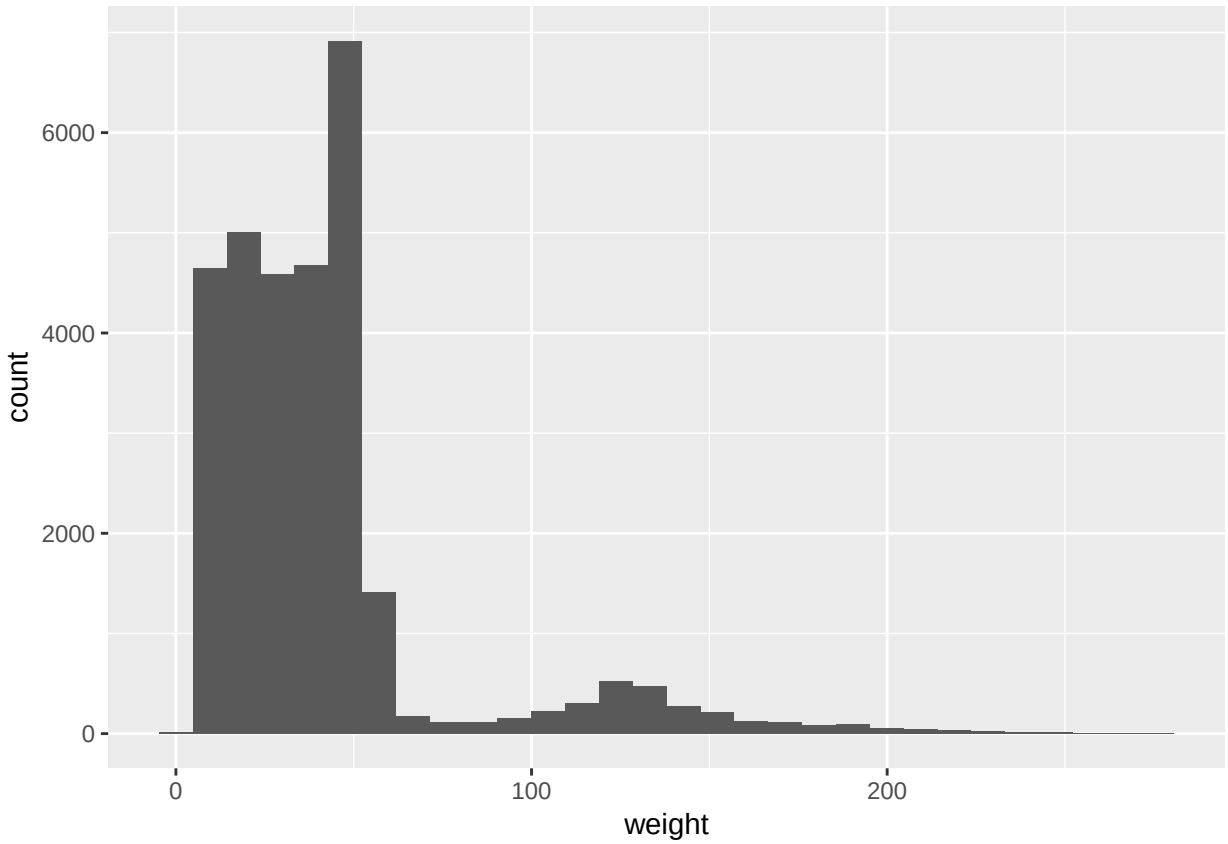
Plotting with ggplot, part 2

Histograms

To plot a histogram we require another geometric object `geom_histogram`, which requires a statistical transformation. Some plot types (such as scatterplots) do not require transformations, each point is plotted at x and y coordinates equal to the original value. Other plots, such as boxplots, histograms, prediction lines etc. need to be transformed, and usually has a default statistic that can be changed via the `stat_bin` argument.

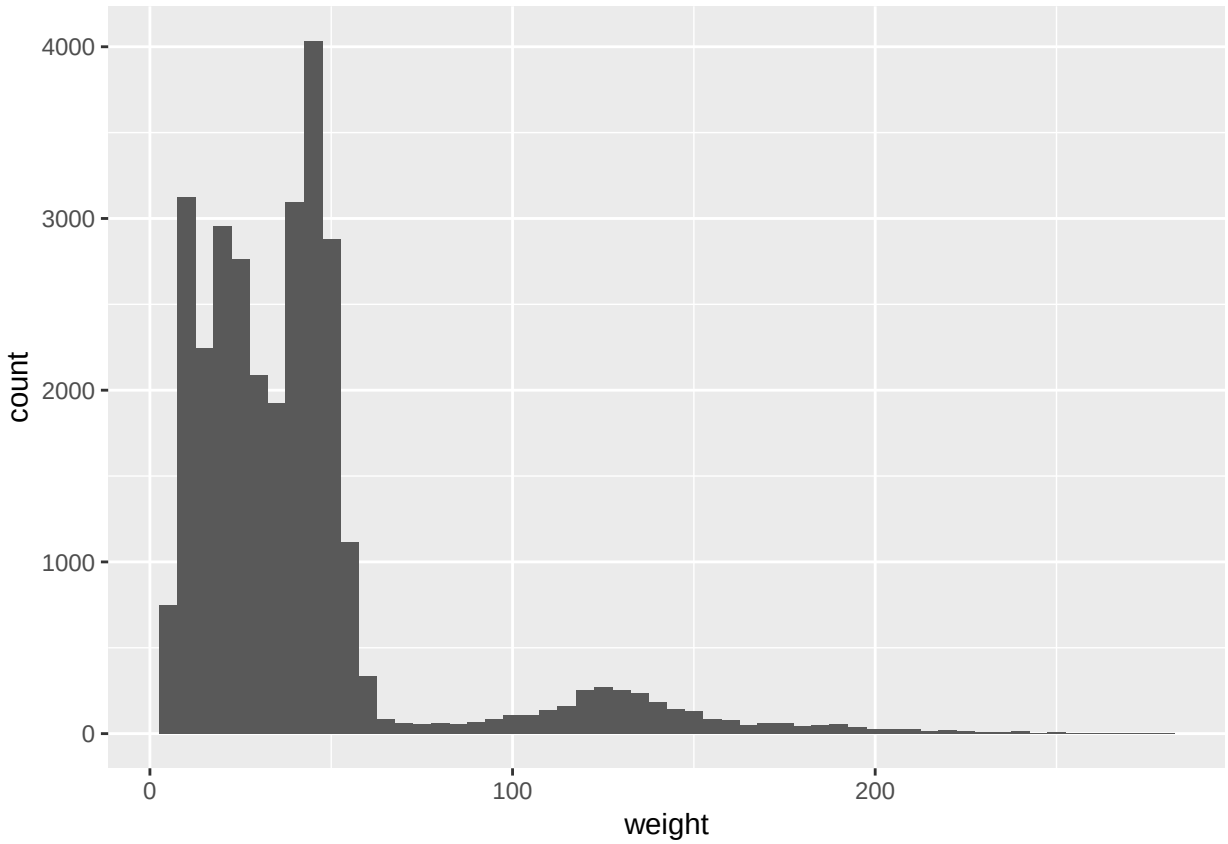
```
ggplot(surveys_complete, aes(weight)) +
  geom_histogram()
```

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



There are two ways to change the default bins if desired, within `geom_historam` you can specify either the number of bins or the binwidth. See `?geom_histogram` for more information.

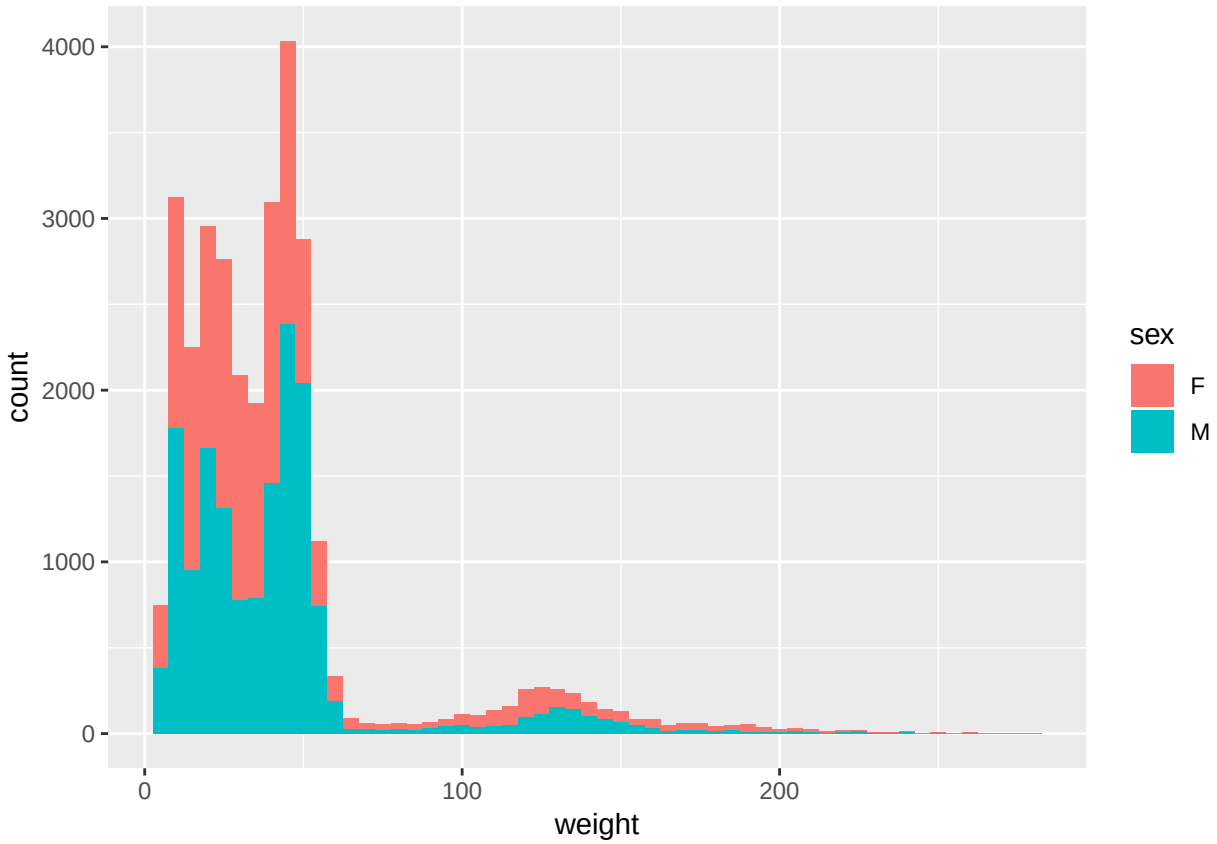
```
ggplot(surveys_complete, aes(weight)) +
  geom_histogram(binwidth = 5)
```



What is clear from plotting the data in this way is that we do not have normally-distributed data. So let's see if we can gain insight using the `fill` plotting option to overlay colour from a factor onto the histogram.

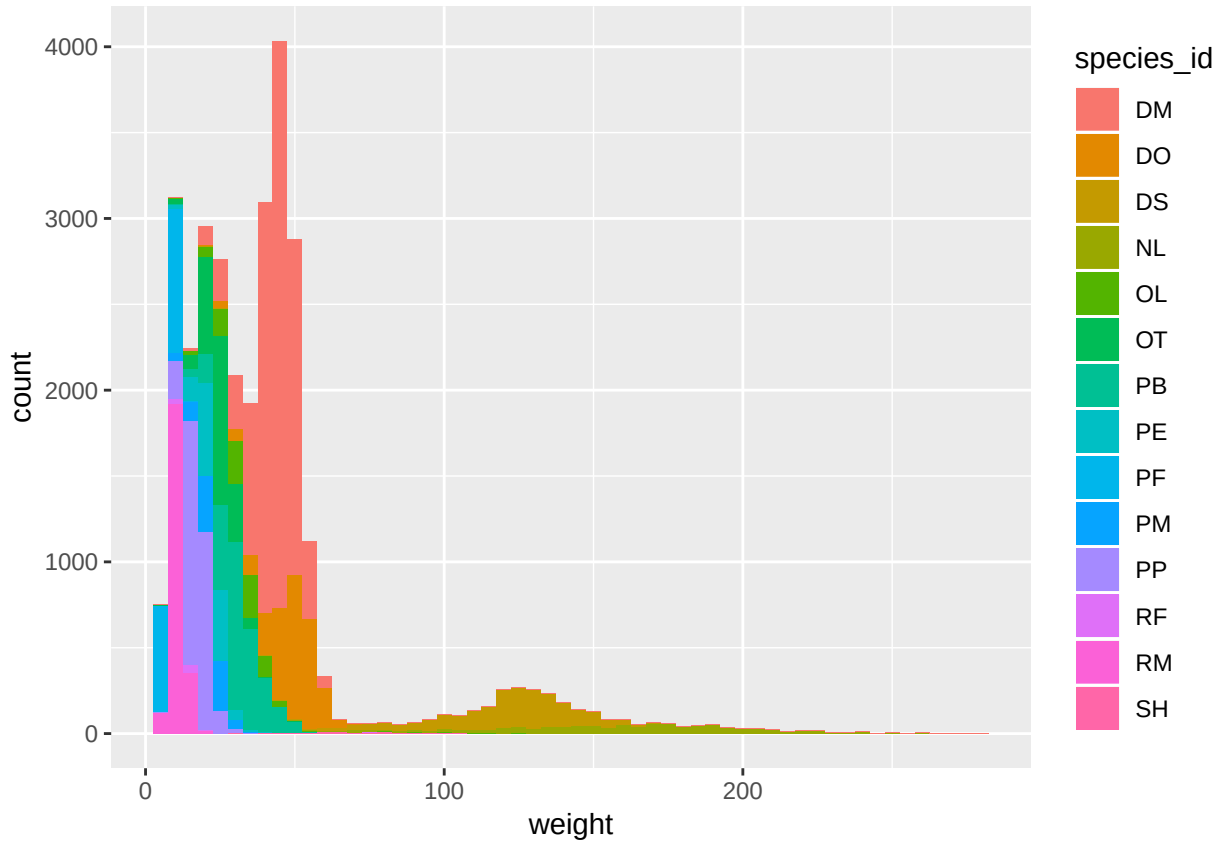
First let's try sex.

```
ggplot(surveys_complete, aes(weight,  fill = sex)) +
  geom_histogram(binwidth = 5)
```



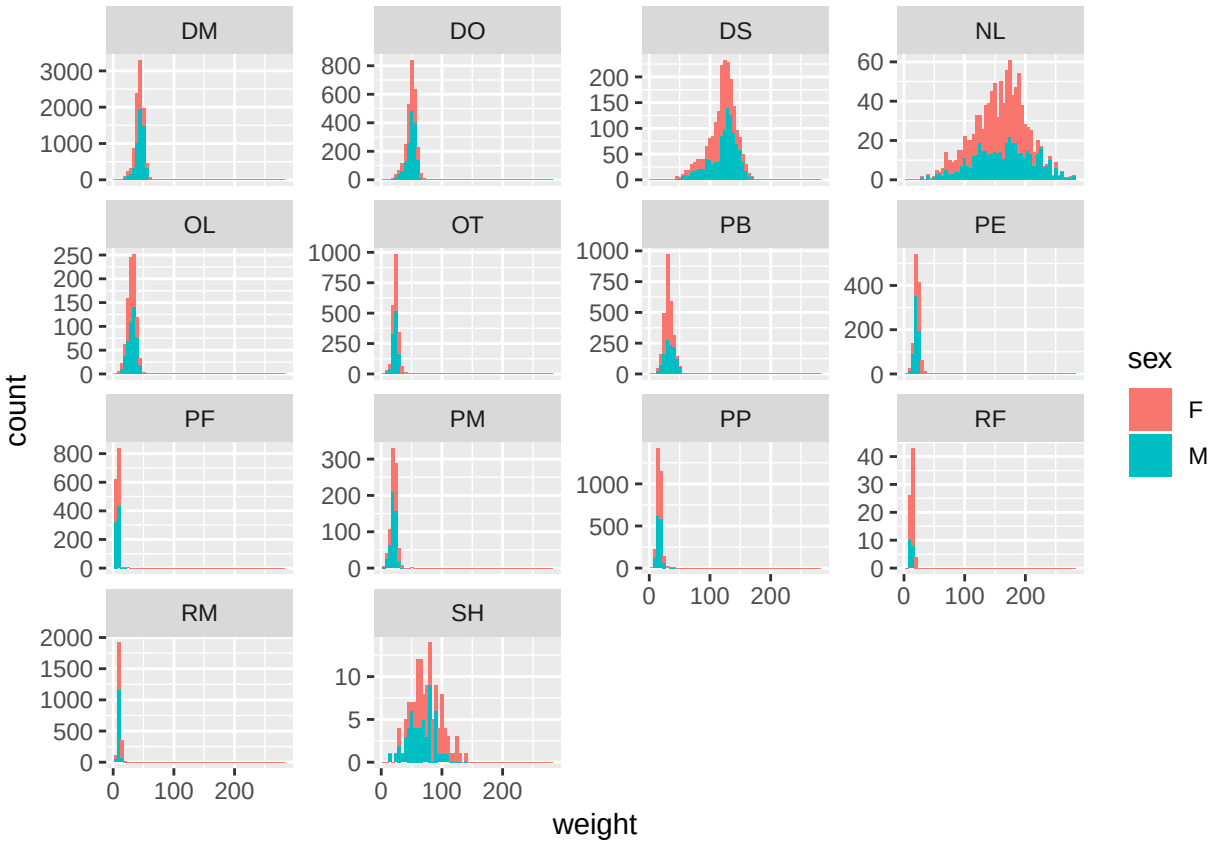
Hmm, perhaps males are a bit skewed to the rightm but that doesn't seem to be the major factor. If we remember back to last week we saw that hindfoot_length was really influenced by species. Perhaps weight is as well?

```
ggplot(surveys_complete, aes(weight,  fill = species_id)) +
  geom_histogram(binwidth = 5)
```



Let's go back to our facetting and pull this all together.

```
ggplot(surveys_complete, aes(weight,  fill = sex)) +
  geom_histogram(binwidth = 5) +
  facet_wrap(~ species_id, scale = "free_y")
```

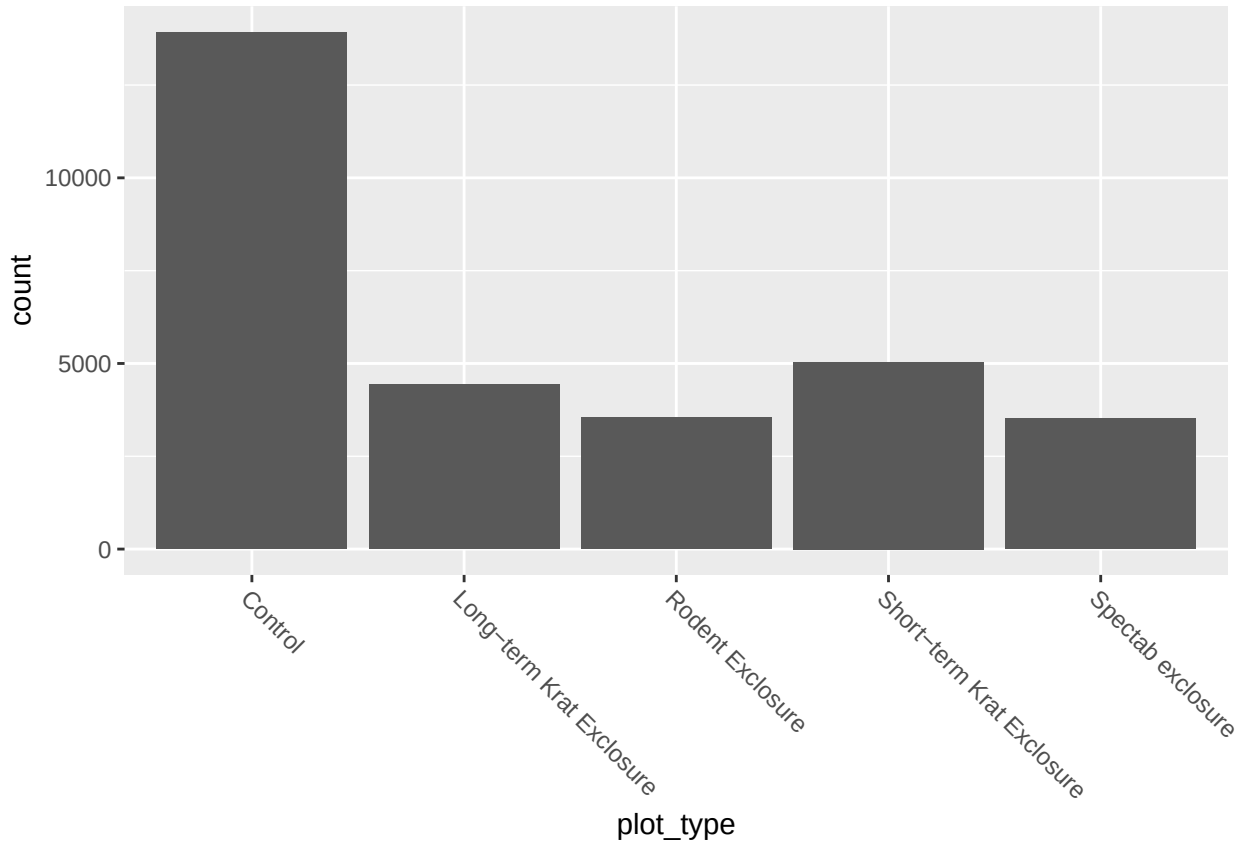


Bar charts

partial source: https://ggplot2.tidyverse.org/reference/geom_bar.html
There are two types of bar charts: `geom_bar()` and `geom_col()`. `geom_bar()` makes the height of the bar proportional to the number of cases in each group (or if the weight aesthetic is supplied, the sum of the weights). If you want the heights of the bars to represent values in the data, use `geom_col()` instead. `geom_bar()` uses `stat_count()` by default: it counts the number of cases at each x position. `geom_col()` uses `stat_identity()`: it leaves the data as is. Essentially this is based on the data you are using. Here we are using a data set (surveys_complete) where each row is an individual, so we will use `geom_bar`. If we had a different tibble where we had already summarized this data we would use `geom_col()`.

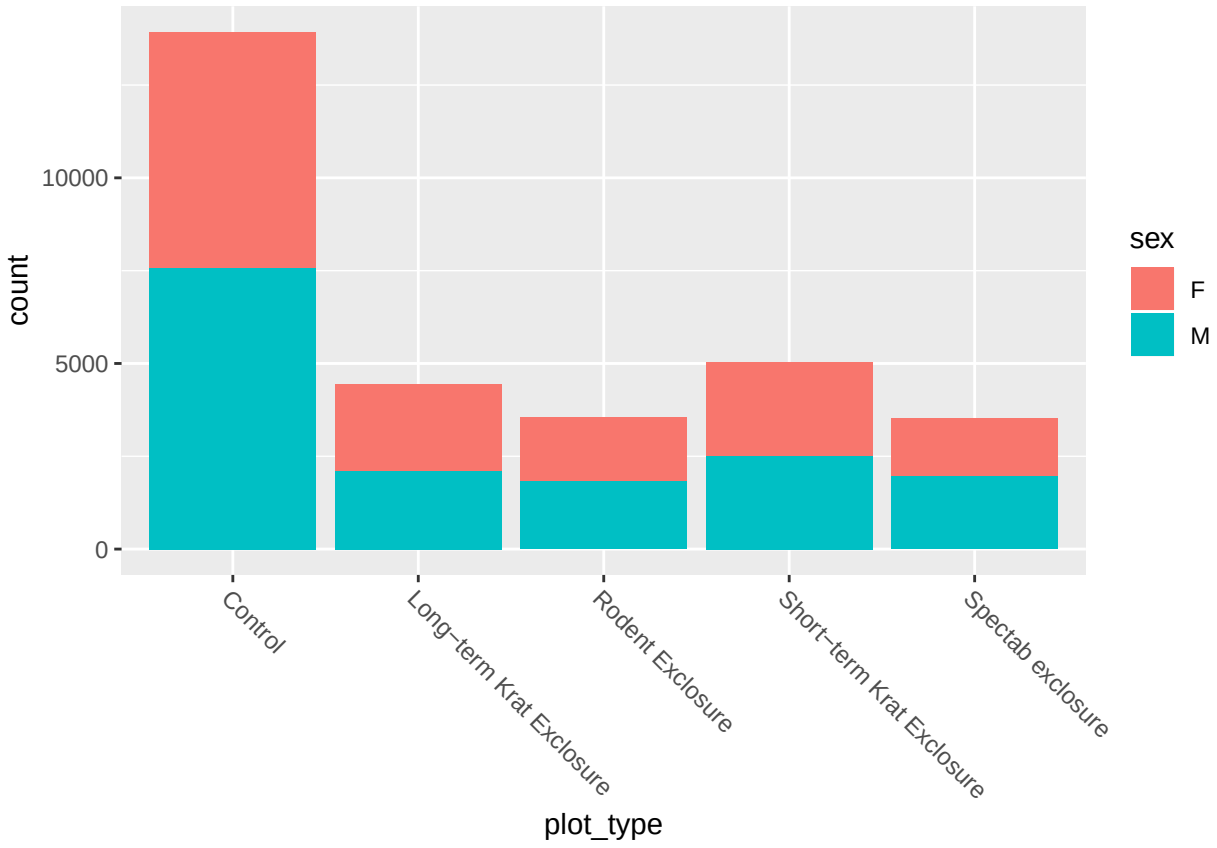
Let's use a bar chart to investigate how many individuals were captured in each type of enclosure (`plot_type`).

```
ggplot(data = surveys_complete, mapping = aes(x = plot_type)) +
    geom_bar() +
    theme(axis.text.x=element_text(angle = -45, hjust = 0))
```



We can use stacked bar charts to add another layer of information onto the bar chart. Let's see if a similar number of male and female animals were caught in each plot type.

```
ggplot(data = surveys_complete, mapping = aes(x = plot_type, fill = sex)) +
  geom_bar()  +
  theme(axis.text.x=element_text(angle = -45, hjust = 0))
```



CHALLENGE

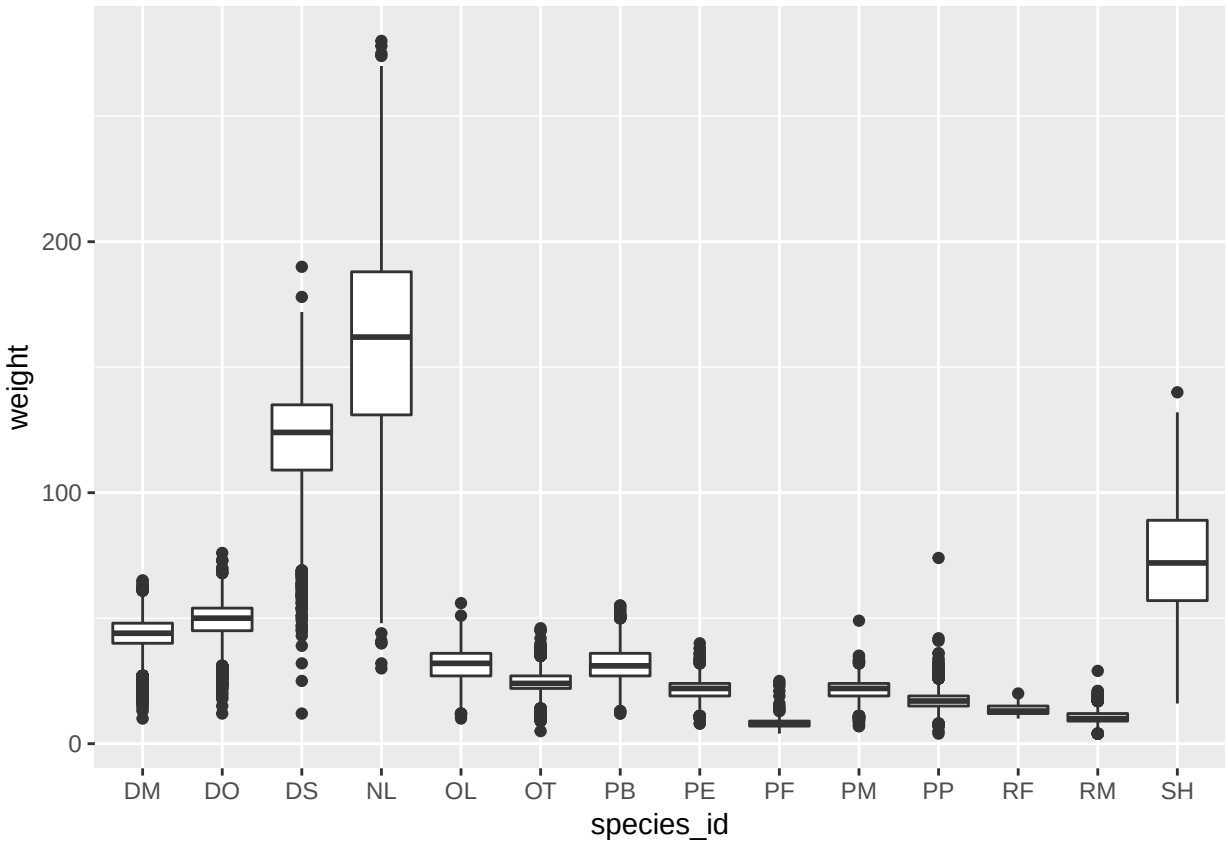
Let's pull some of what we now know how to do to produce a plot that asks a biological question:

What is the influence of plot_type on the species that are captured?

Add summary statistics

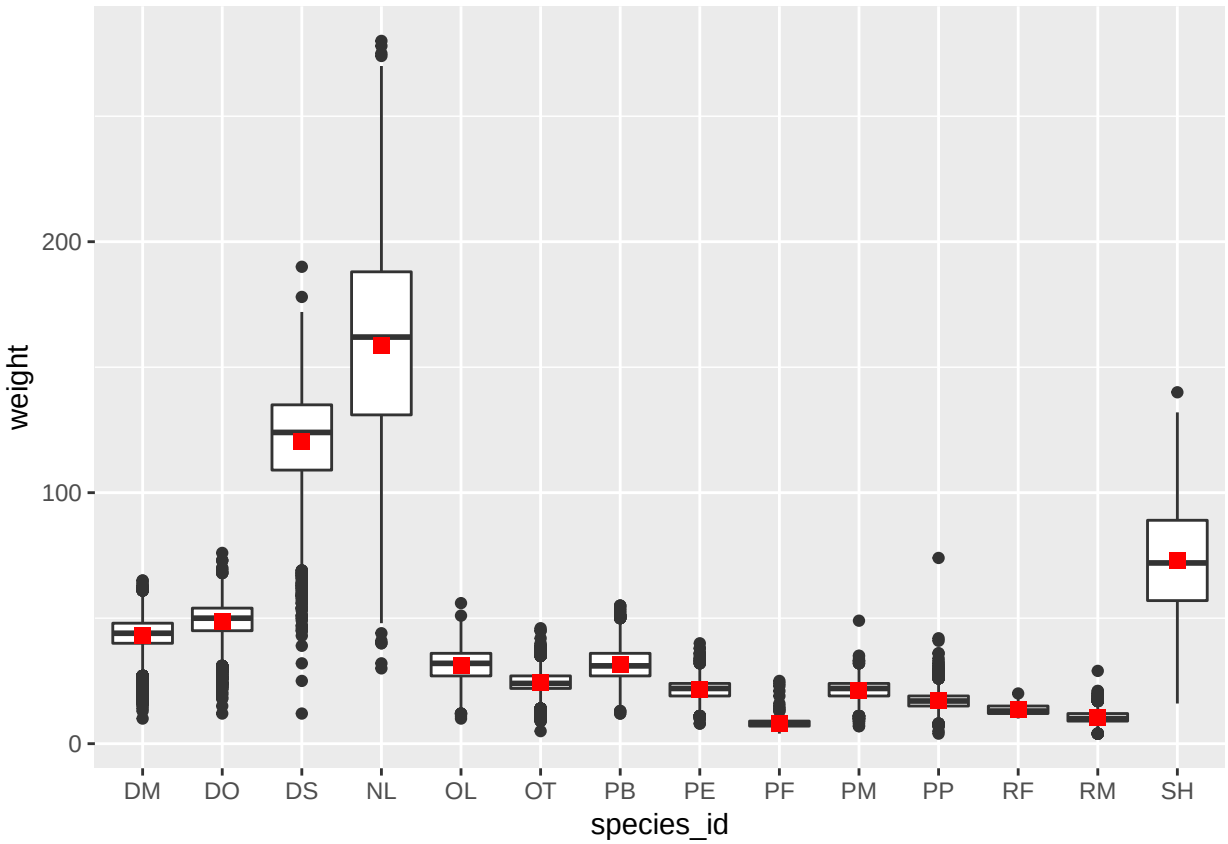
Let's go back to our boxplot from last class. As above, the nature of a boxplot is that there has already been a statistical transformation done "under the hood" to produce the figure.

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



There are a lot of built-in options that we can specify to add to this plot. See `?geom_boxplot` for the full list. For now, let's add the mean.

```
ggplot(data = surveys_complete, mapping = aes(x = species_id, y = weight)) +
    geom_boxplot() +
    stat_summary(fun.y = "mean", geom = "point", size = 2.5, color = "red", shape = 15)
```

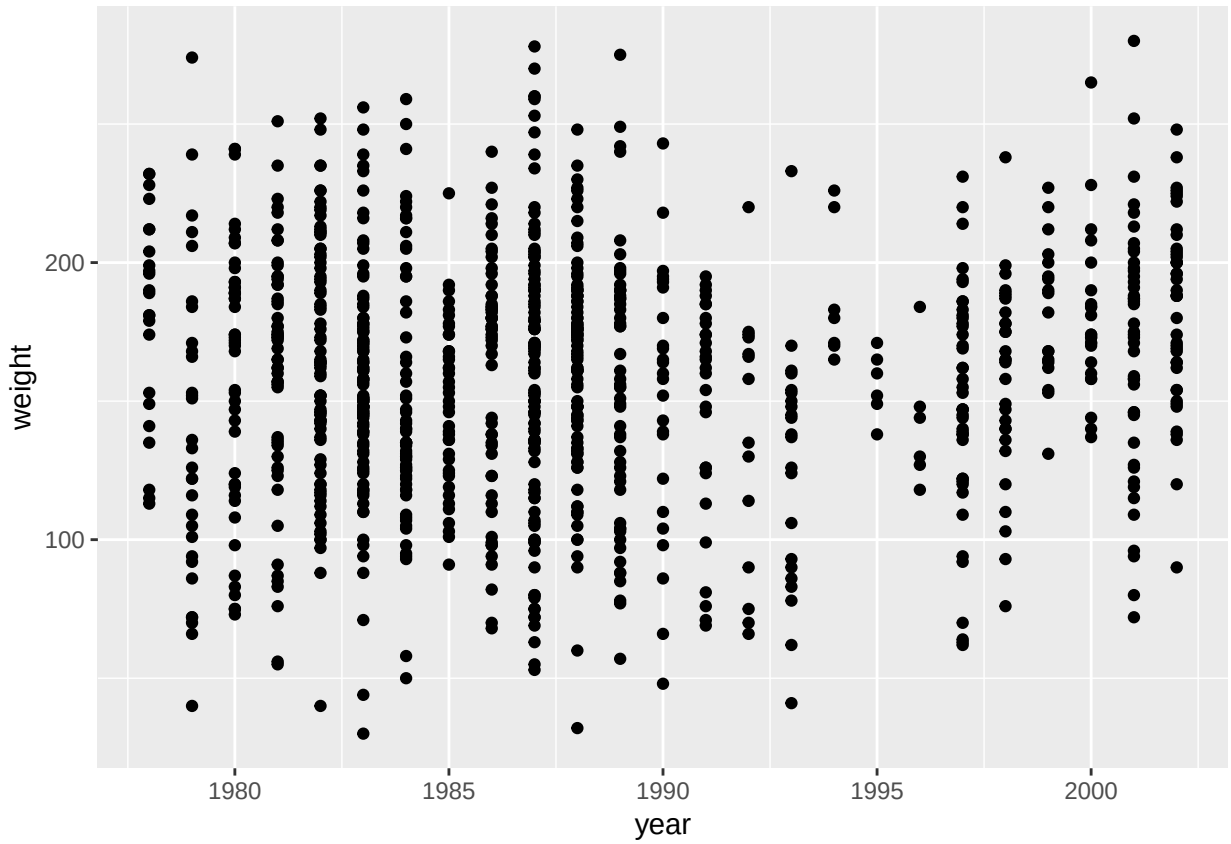


The summary statistics are added through the various `stat_` functions. It is probably the case that you will want to decide which `stat_` function is appropriate after deciding which `geom_` is appropriate, which is based on the data and the message you are trying to convey. Here are some examples (from a perhaps underwhelmingly titled blog post: https://www.dummies.com/programming/r/how-to-plot-summarized-data-in-a-ggplot2-in-r/)

Default_Geom	Stat	Description
geom_bar()	stat_bin()	Counts the number of observations in bins.
geom_line()	stat_smooth()	Creates a smooth line.
geom_point()	stat_sum()	Adds values.
geom_point()	stat_identity()	No summary. Plots data as is.
geom_boxplot()	stat_boxplot()	Summarizes data for a box-and-whisker plot.

Let's try it out. Let's first create a plot to see if there's a relationship between weight and year for the genus Neotoma.

```
p <- surveys_complete %>%
  filter(genus == "Neotoma") %>%
  ggplot(aes(x = year, y = weight )) +
  geom_point()
p
```

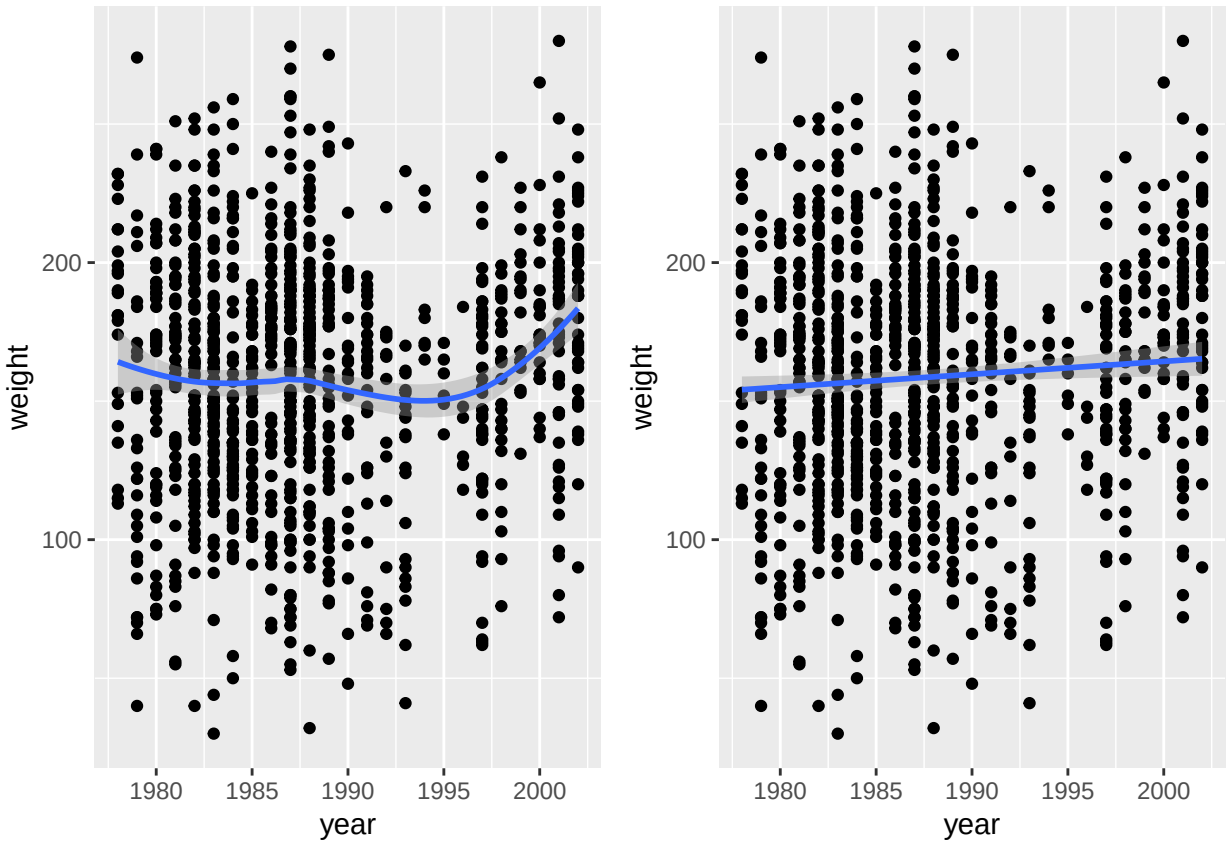


Now we add a line of best fit from model on top of the raw data. To help our eyes decide if there is a pattern in the data (our eyes are really good at this!). There's a few options (as always `?` is your friend) but here we'll look at two, 'loess' and 'lm' (linear model).

```
p_loess  <- p +
  stat_smooth(method = "loess")

p_lm <- p +
  stat_smooth(method = "lm")

grid.arrange(p_loess, p_lm, ncol = 2)
```



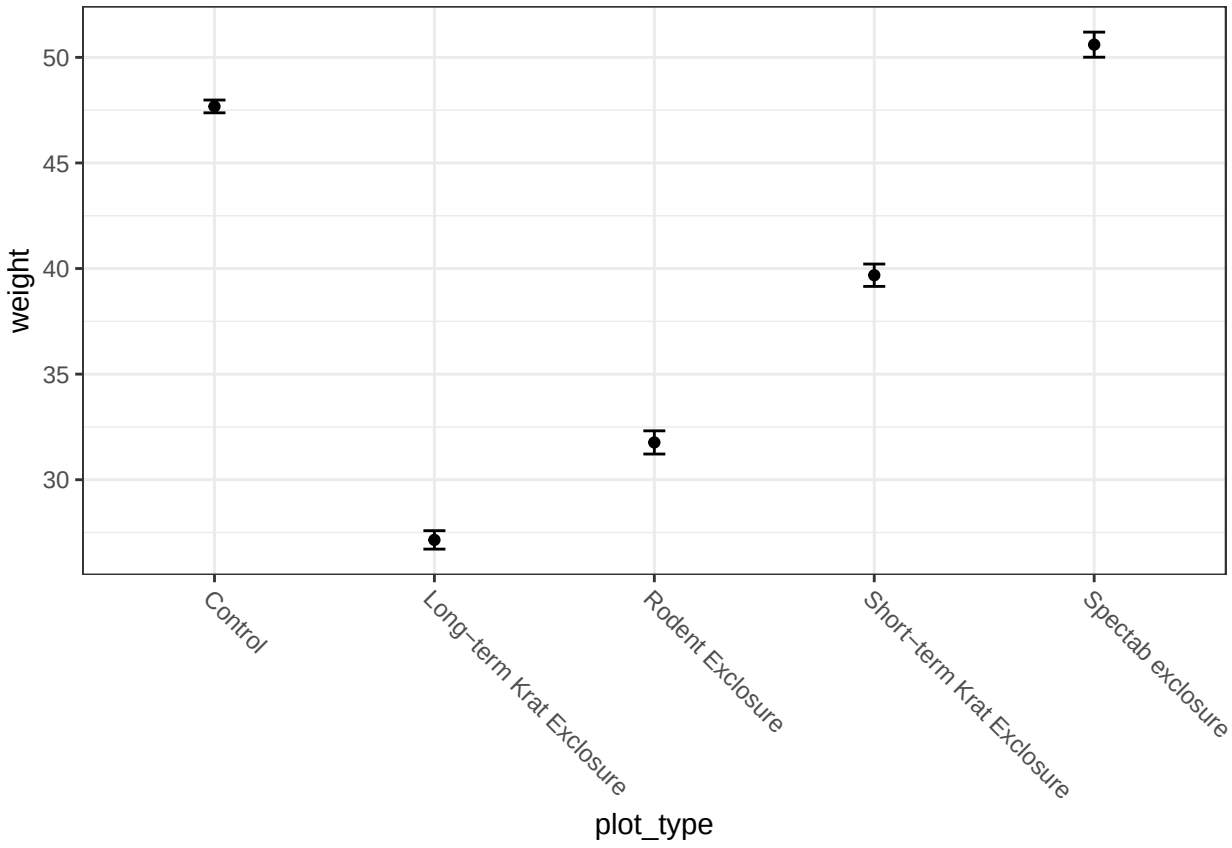
CHALLENGE

How do you make the loess line less smooth?

Okay, one more. A really common thing that we often want to do is to add error bars onto our figures. Let's test whether the average weight of animals caught waas influenced by plot type. You'll notice that the way we create this plot is a bit different – we're not using a `geom_` layer, because right away we need to take the mean of all observations which is a `stat_` layer. Then we add a second `stat_` layer for the standard error bars.

```
bar <- ggplot(data = surveys_complete, mapping = aes(x = plot_type, y = weight)) +
  stat_summary(fun.y = mean, geom = "point", fill = "White", colour = "Black") +
  stat_summary(fun.data = mean_se, geom = "errorbar", width=0.1) +
  theme_bw() +
  theme(axis.text.x=element_text(angle = -45, hjust = 0))

bar
```



CHALLENGE

How do you add a 95% confidence interval instead of a standard error bar to the figure above?

Homework

1. Make your project into a github repo. It's actually easier to do github first then make a new project in R Studio second, but if you follow the instructions here (https://happygitwithr.com/existing-github-first.html) it shouldn't be too hard to do the reverse. When you've done this please email Aleeza the URL of your github repo.

Assigned Readings

1. We've now done a fairly quick tour through `ggplot` basics. We've covered more or less what is typically covered in an introductory workshop. There is a lot more that can be said and learned about graphics though, and at this point I want us to start learning something about the principals of design. Luckily for us, there is a new, free, online textbook on *Data Analysis and Visualization* available here: https://serialmentor.com/dataviz/balance-data-context.html
   Before class on March 14 I'd like you to read at minimum chapters 1-5 and 13-26. I don't think this will actually be as difficult as it perhaps sounds. Many chapters are short and the writing is very accessible.