Week 2 Assignment

Jessalyn Chuang

(1) Measures of location and data spread

```
# Load necessary data.
crab_data <- read_csv(file=here("Labs/week2/pie_crab.csv"))
```

```
## Rows: 392 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): site, name
## dbl  (6): latitude, size, air_temp, air_temp_sd, water_temp, water_temp_sd
## date (1): date
##
## 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.
```

```
mean(crab_data$size)
```

```
## [1] 14.65801
```

```
median(crab_data$size)
```

```
## [1] 14.435
```

```
sd(crab_data$size)
```

```
## [1] 3.503304
```

##a. What is the mean fiddler crab carapace size? The mean fiddler crab carapace size is 14.65801 mm.

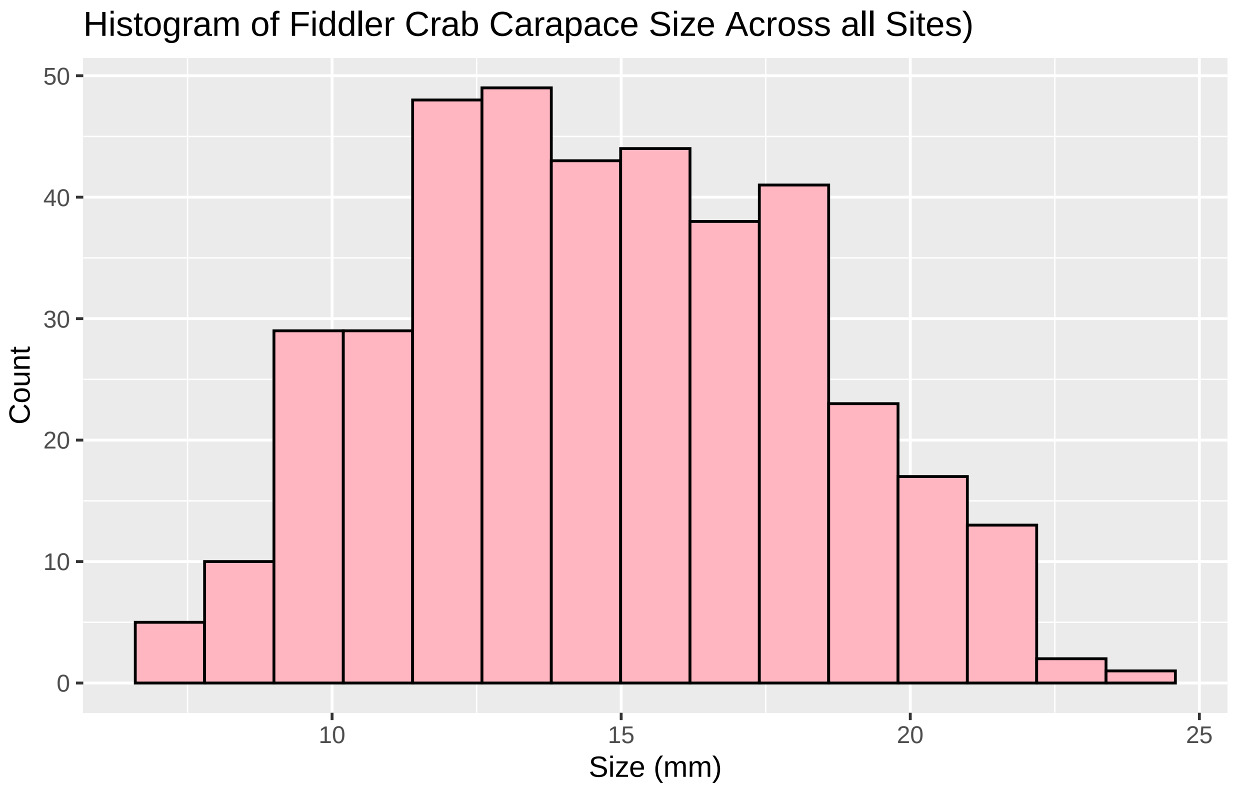
##b. What is the median fiddler crab carapace size? The median fiddler crab carapace size is 14.435 mm.

##c. What is the standard deviation in fiddler crab carapace size? The standard deviation in fiddler crab carapace size is 3.503304 mm.

(2) Creating Histograms

##a. One of the ways to visualize data that we have discussed in class is using histograms. Create a figure that is a histogram of fiddler crab carapace size (across all sites), and take care to make sure your figure uses an appropriate number of bins and includes a caption. (Hint: Learn how to use the geom_histogram() call to achieve this by typing ?geom_histogram() into your console or Google.)

```r
ggplot(data = crab_data, aes(x = size)) +
  geom_histogram(bins = 15, fill = "lightpink", color = "black") +
  labs(x = "Size (mm)",
       y = "Count",
       title = "Histogram of Fiddler Crab Carapace Size Across all Sites)",
       caption = "This histogram shows the distribution of Fiddler Crab carapace
       size across all sites with mean size of 14.658 mm and median of 14.435 mm.")
```

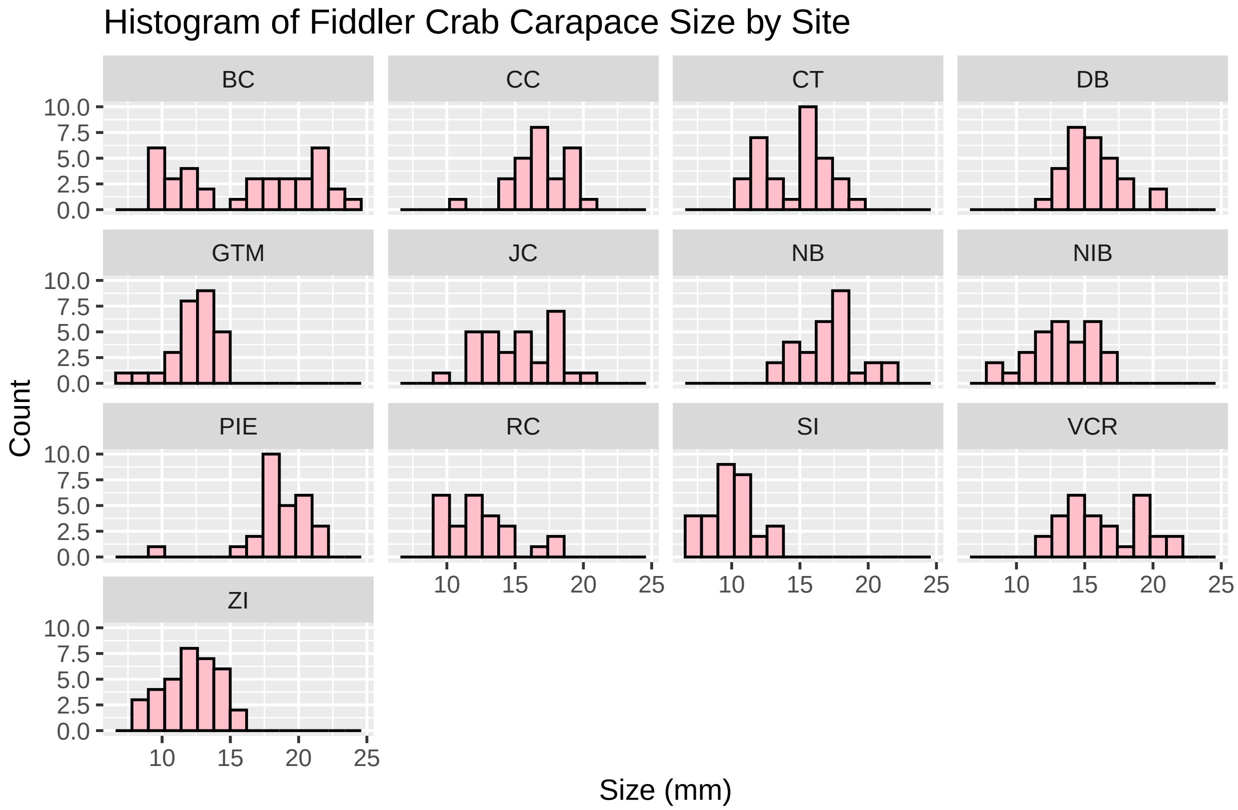


Histogram of Fiddler Crab Carapace Size Across all Sites)

This histogram shows the distribution of Fiddler Crab carapace
size across all sites with mean size of 14.658 mm and median of 14.435 mm.

##b. Create a second figure that now displays separate histograms for fiddler crab carapace size for each of the 13 sites. Again, make sure axes are appropriately labeled, and your figure includes a caption. (Hint: Use the facet_wrap() call and explore its functionality either in the console or via Google.)

```r
ggplot(data = crab_data, aes(x = size)) +
  geom_histogram(bins = 15, fill = "pink", color = "black") +
  facet_wrap(~ site) +  # Create separate histograms for each site
  labs(x = "Size (mm)",
       y = "Count",
       title = "Histogram of Fiddler Crab Carapace Size by Site",
       caption = "Each panel represents a separate site, showing the distribution of carapace sizes.")
```


Histogram of Fiddler Crab Carapace Size by Site

Each panel represents a separate site, showing the distribution of carapace sizes.

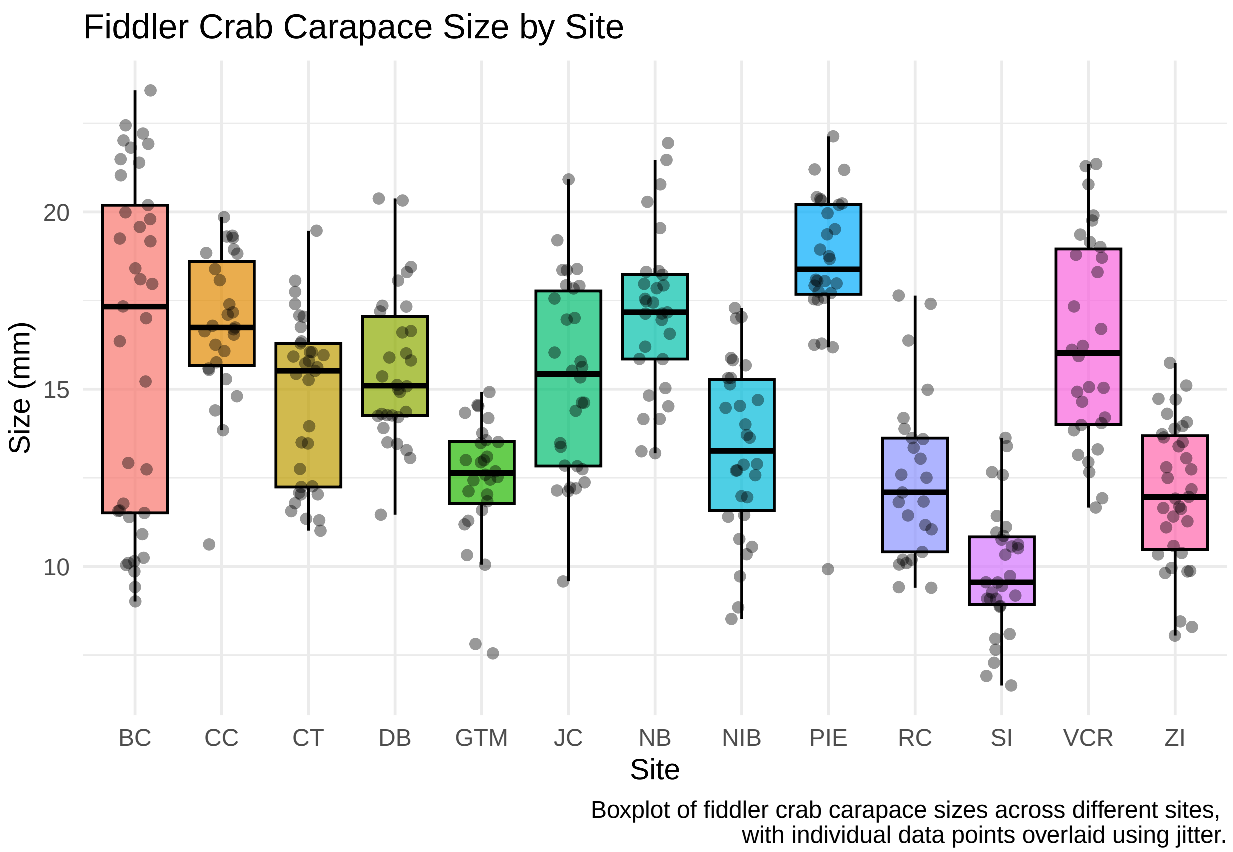
##c. How do the raw values of fiddler crab carapace size in your first figure compare to the mean and median values you calculated above? Based on your second figure, do any sites display strong differences in their data distributions?

Since the mean is slightly larger than the median, this suggests that the overall distribution of fiddler crab carapace size is slightly skewed to the right. The first figure shows the distribution. Meanwhile, the second figure that shows size distributions for individual sites reveals significant site-specific variations that deviate from this general trend. At sites such as ZI and SI, there is more of a concentration of crabs at smaller sizes and very few larger individuals. On the other hand, sites like GTM and PIE have more crabs grow to larger sizes compared to other locations. Some sites, such as DB and CT, show normal-like, symmetric distributions. To summarize, the overall dataset maintains a slight right-skewed distribution while the individual site breakdown indicate different distributions from the whole datset.

(3) Data Visualization - the Golden Rules

##a. Make a new plot examining any other relationship you are interested in visualizing using the dataset. It can be a scatterplot (geom_point()), a jitter plot (geom_jitter()), a boxplot (geom_boxplot()), a histogram (geom_histogram()), or any other data display you'd like. Take care to follow each of the golden rules of data visualization - Accuracy, Clarity, Aesthestics - and be sure to include a descriptive figure caption.

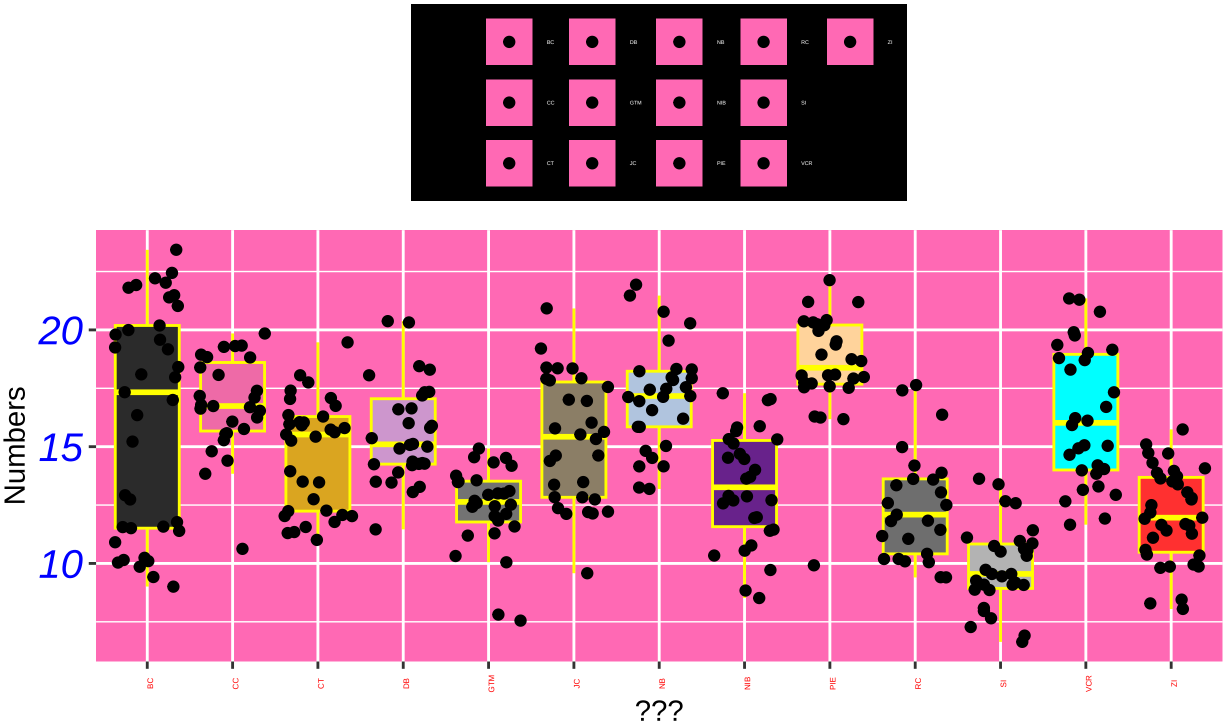
```
ggplot(data = crab_data, aes(x = site, y = size, fill = site)) +
  geom_boxplot(outlier.shape = NA, color = "black", alpha = 0.7) +  # Fill with different colors by sit
  geom_jitter(width = 0.2, alpha = 0.4, color = "black") +
  labs(x = "Site",
       y = "Size (mm)",
       title = "Fiddler Crab Carapace Size by Site",
       caption = "Boxplot of fiddler crab carapace sizes across different sites,
       with individual data points overlaid using jitter.") +
  theme_minimal() +
  theme(legend.position = "none")
```



##b. Now, using the same base figure you created above in part a, create the worst version of that same figure that you possibly can. Think color, font size, replacing points with emojis, etc. - the uglier and more confusing the better! You must also include a figure caption, but it, too, should be unhelpful.

```
ggplot(data = crab_data, aes(x = site, y = size, fill = site)) +
  geom_boxplot(outlier.shape = NA, color = "yellow", fill = sample(colors(), 13)) +  # Random ugly colo
  geom_jitter() +
  labs(x = "???",
       y = "Numbers",
       title = "Crabby Sizes",
       caption = "Data is fun!") +
  theme(
    axis.text.x = element_text(size = 3, angle = 90, color = "red"),  # Unreadable x-axis
    axis.text.y = element_text(size = 15, face = "italic", color = "blue"),  # Huge y-axis
    plot.caption = element_text(size = 20, color = "green"),  # Oversized, distracting caption
    panel.background = element_rect(fill = "hotpink"),
    legend.position = "top",  # Moves legend to a useless place
    legend.background = element_rect(fill = "black"),
    legend.text = element_text(size = 2, color = "white")  # Impossible to read legend text
  )
```

Crabby Sizes



Data is fun!