STAT 3280 Homework 2

Yingyi Zhu

September 21, 2022

.Rmd file can be found on Collab under Resources/Assigments

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

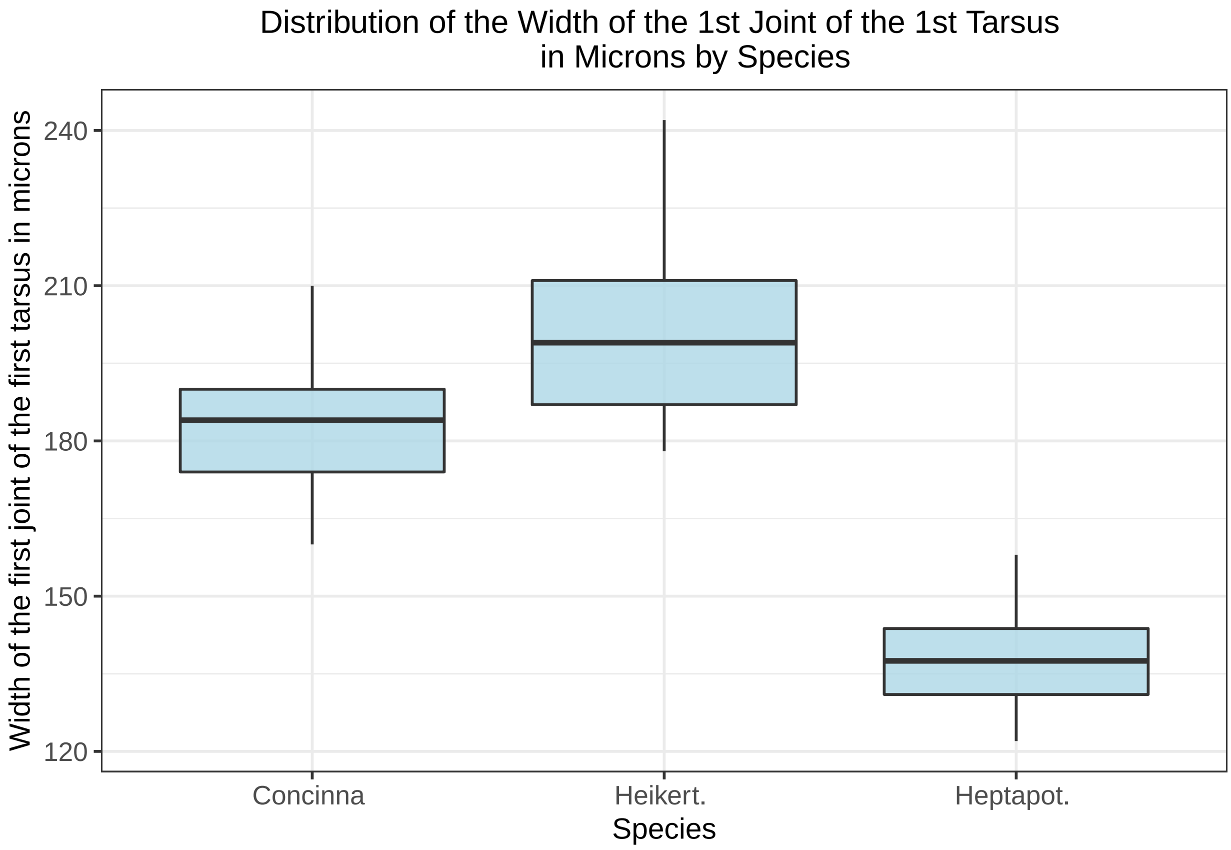
```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v readr   2.1.2

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

```
options(dplyr.summarise.inform = FALSE)
```

Q1: Using the `flea` dataset, create a plot that displays the distribution of the variable `tars1` by `species`. Ensure colors, labels, and themes make the plot easy to understand. Information about the dataset can be found at https://www.rdocumentation.org/packages/GGally/versions/1.5.0/topics/flea.

```r
setwd("/Users/zach0422/Desktop/STAT3280/data/")
load("flea.rdata")
my_theme <- theme_bw() +
  theme(axis.text = element_text(size = 10),
        axis.title = element_text(size = 11),
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 11),
        plot.title = element_text(size = 12)) +
  theme(plot.title = element_text(hjust = 0.5))

plot_1 <- ggplot(flea) +
  geom_boxplot(aes(x = species, y = tars1), fill = "lightblue", alpha = 0.8) +
  labs(x = "Species",
       y = "Width of the first joint of the first tarsus in microns",
       title = "Distribution of the Width of the 1st Joint of the 1st Tarsus
       in Microns by Species") +
  my_theme
plot_1
```

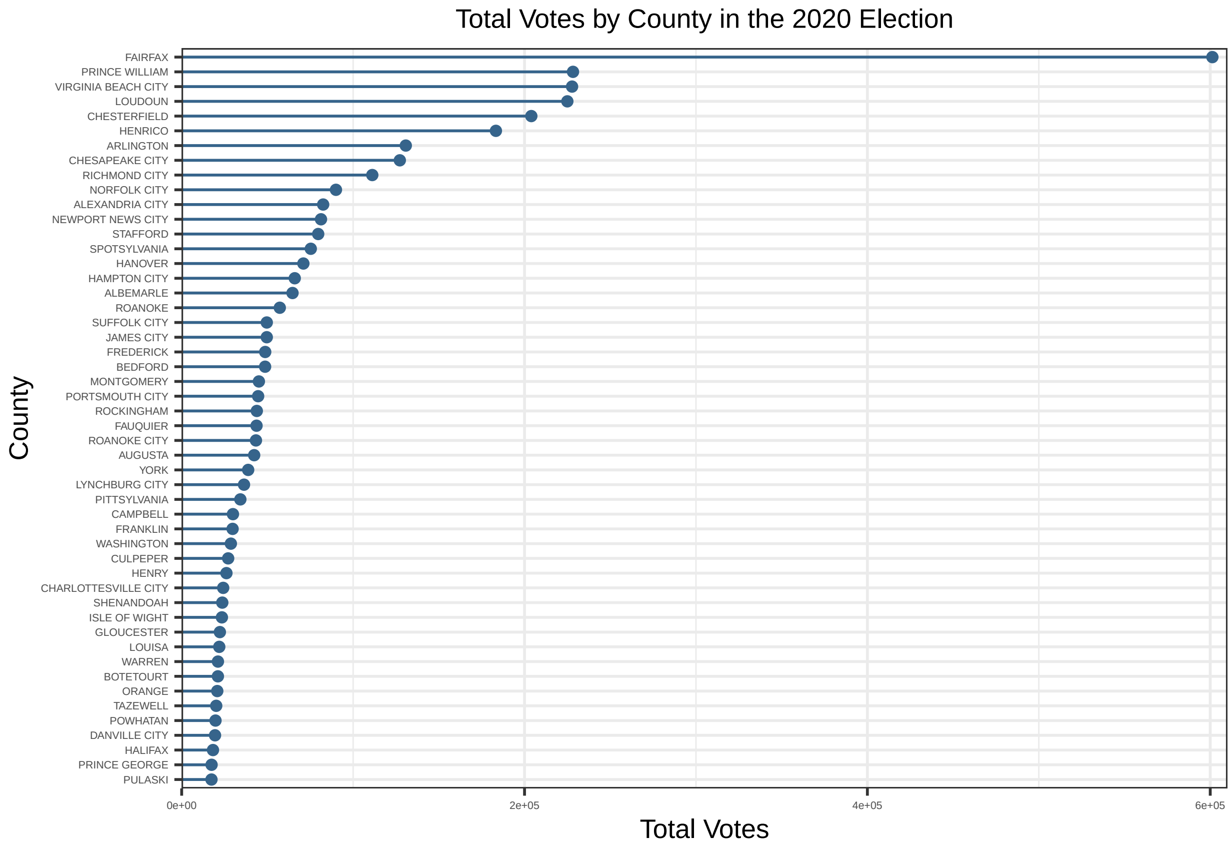


Q2: Using the `RecentVAElections` data set, create a dot & bar plot (for a large number of groups) the total votes (`totalvotes`) cast by county (`county_name`) in the 2020 election. Only include the largest 50 counties. Ensure colors, labels, and themes make the plot easy to understand.

```r
setwd("/Users/zach0422/Desktop/STAT3280/data/")
load("RecentVAElections.rdata")
RecentVAElections_1 <- RecentVAElections%>%
  filter(year == 2020)%>%
  arrange(desc(totalvotes))%>%
  distinct(totalvotes, .keep_all=TRUE)%>%
  top_n(50)
```

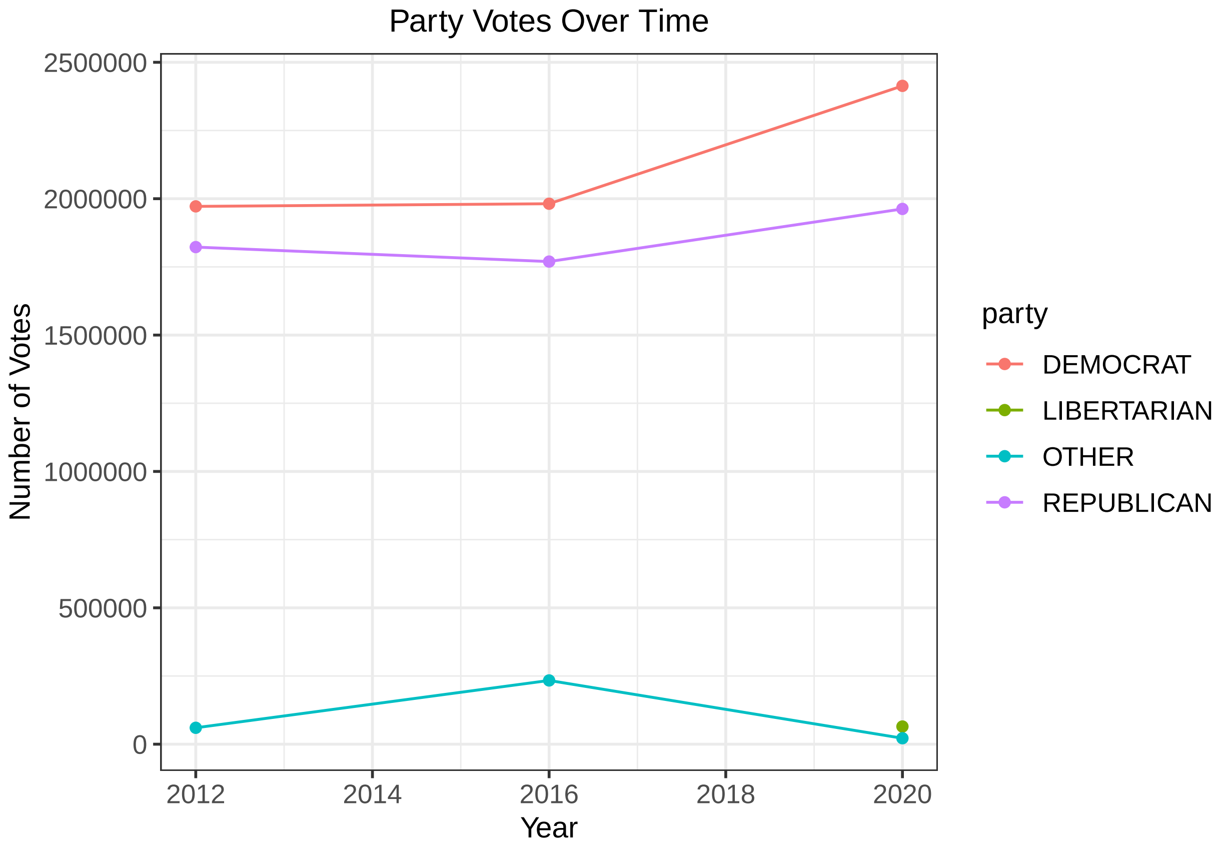
```
## Selecting by totalvotes
```

```r
my_theme1 <- theme_bw() +
  theme(axis.text = element_text(size = 4),
        axis.title = element_text(size = 10),
        legend.text = element_text(size = 7),
        legend.title = element_text(size = 8),
        plot.title = element_text(size = 10)) +
  theme(plot.title = element_text(hjust = 0.5))

plot_2 <- ggplot(RecentVAElections_1) +
  geom_point(aes(x = totalvotes, y = reorder(county_name, totalvotes)),
             color = "steelblue4") +
  geom_segment(aes(x = 0, xend = totalvotes,
                   y = reorder(county_name, totalvotes),
                   yend = reorder(county_name, totalvotes)),
               color = "steelblue4") +
  scale_x_continuous(limits = c(0, 610000), expand = c(0, 0)) +
  labs(x = "Total Votes", y = "County",
       title = "Total Votes by County in the 2020 Election") +
  my_theme1
plot_2
```



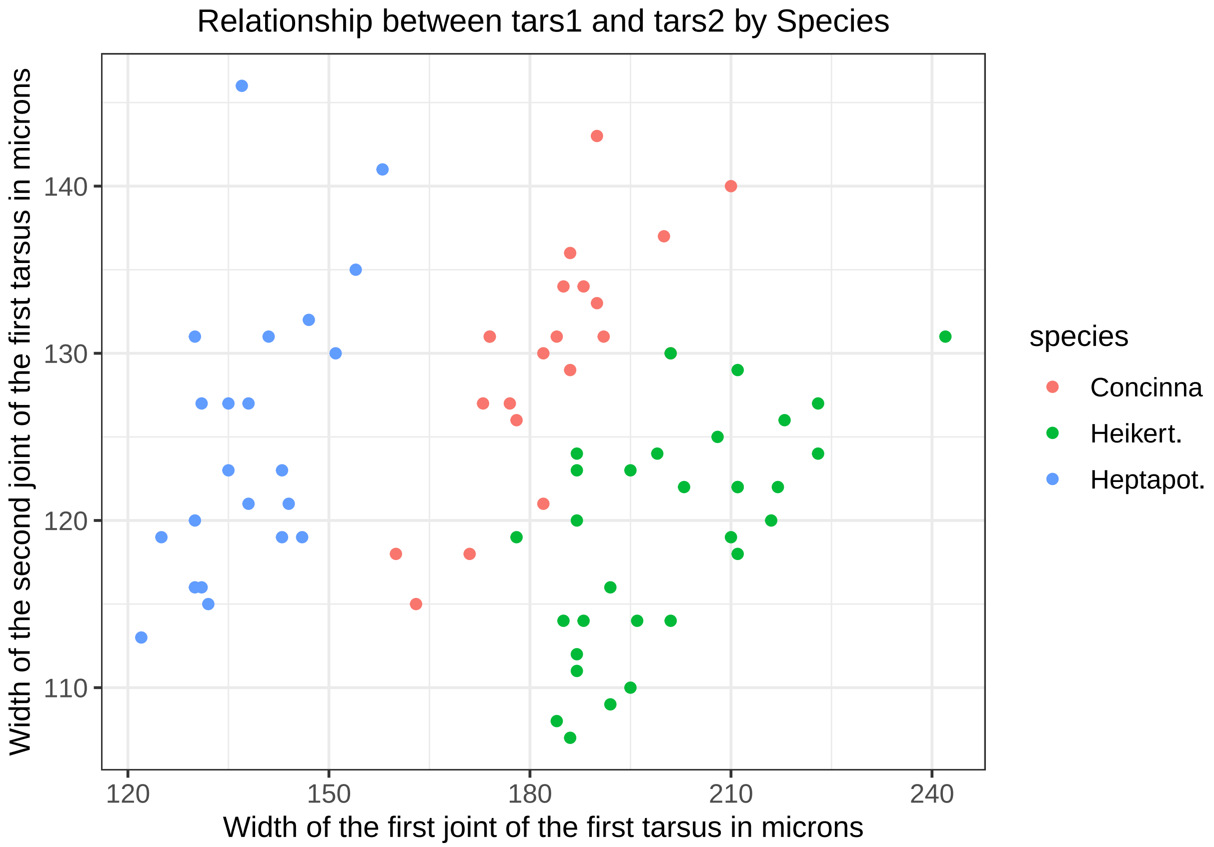
Q3: Using the RecentVAElections data set, create a plot that describes the party vote (candidatevotes, party) over time (year), **aggregated (sum)** for all Virginia counties. Ensure colors, labels, and themes make the plot easy to understand.

```
RecentVAElections_2 <- RecentVAElections%>%
  group_by(year, party)%>%
  summarise(partyvotes = sum(candidatevotes))

plot_3 <- ggplot(data = RecentVAElections_2, aes(x = year,y = partyvotes,
                                                 color = party)) +
  geom_line() +
  geom_point() +
  labs(x = "Year", y = "Number of Votes", title = "Party Votes Over Time") +
  my_theme

plot_3
```



Q4: Using the `flea` data set, create a scatter plot of the `tars1` and `tars2` variables. If you think there exists an association between the variables, highlight this in your visualization. If you think these quantities differ by species, highlight this in your visualization as well. Ensure colors, labels, and themes make the plot easy to understand.

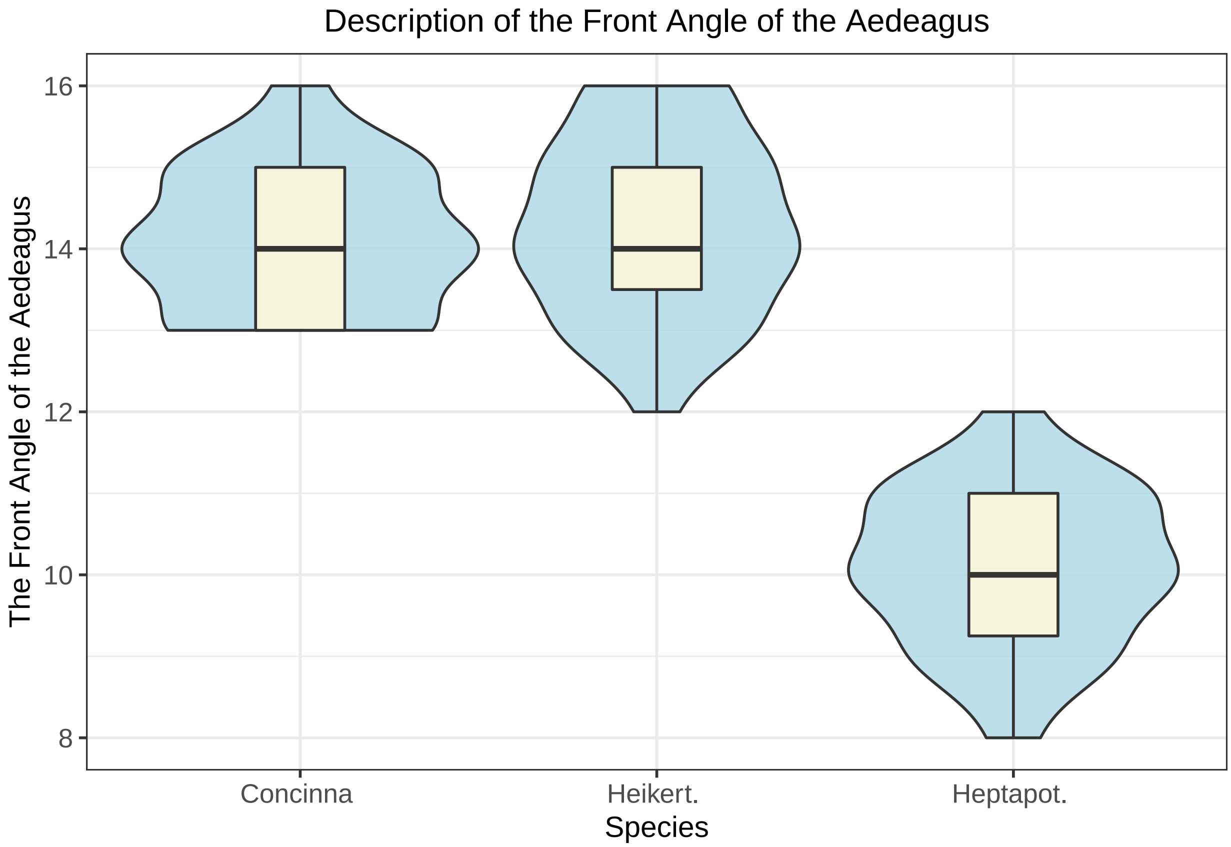
```
plot_4 <- ggplot(flea) +
  geom_point(aes(x = tars1, y = tars2, color = species)) +
  labs(x = "Width of the first joint of the first tarsus in microns",
       y = "Width of the second joint of the first tarsus in microns",
       title = "Relationship between tars1 and tars2 by Species") +
  my_theme
plot_4
```



As indicated in the graph, it does not look like there is an association between tars1 and tars2 since the dots are randomly distributed without a certain pattern. However, it seems that the quantities differ by species since Heikert tends to have greater tars1 and slightly less tars2 compared to Concinna and Heptapot. In addition, Concinna seems to have higher tars1 compared to Heptapot, while their tars2 values are roughly the same.

Q5: Using the `flea` data set, describe the `aede2` variable by species using either a boxplot, violin plot, strip plot, beeswarm plot, or some combination. Choose the visual you think best presents the data, and ensure colors, labels, and themes make the plot easy to understand.

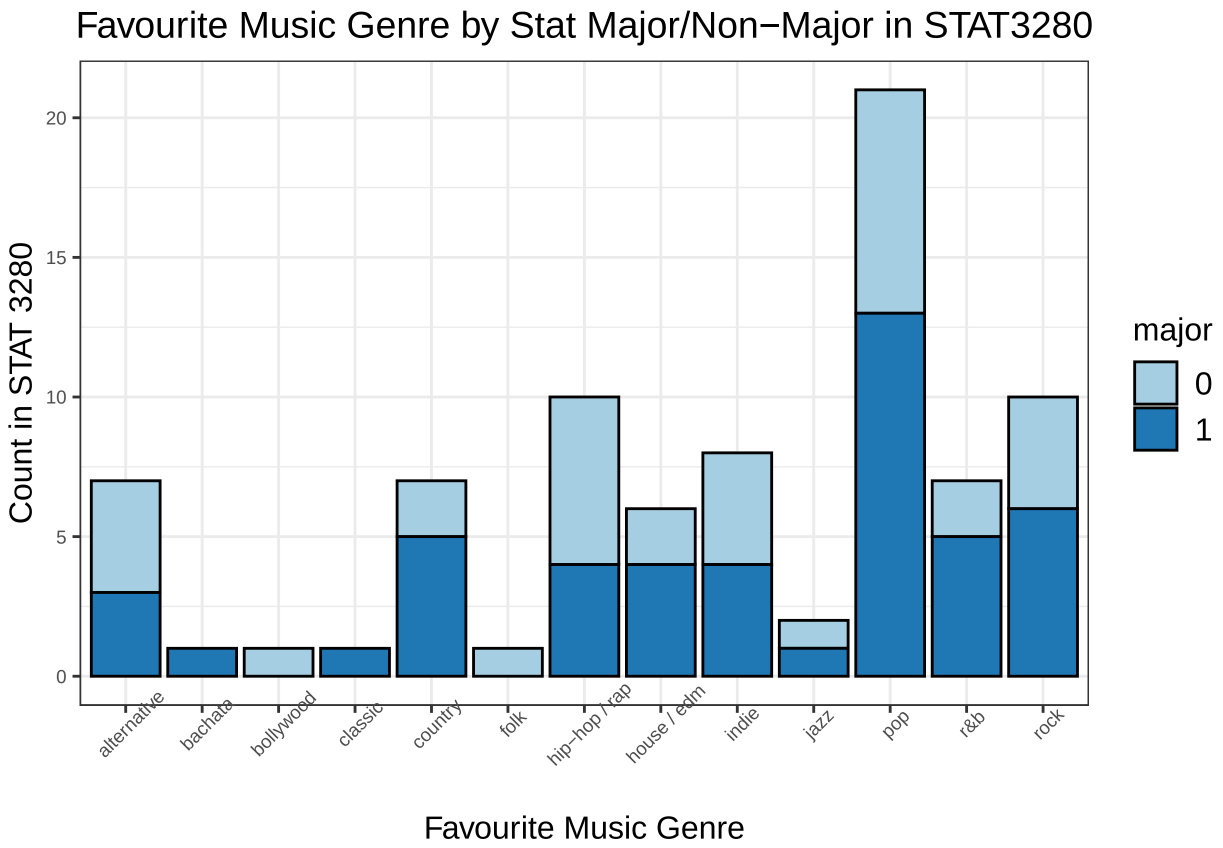
```
plot_5 <- ggplot(flea) +
  geom_violin(aes(x = species, y = aede2), fill = "lightblue", alpha = 0.8,
              width = 1) +
  geom_boxplot(aes(x = species, y = aede2), fill = "beige", width = 0.25,
               outlier.shape = 25, outlier.size = 10) +
  labs(x = "Species", y = "The Front Angle of the Aedeagus",
       title = "Description of the Front Angle of the Aedeagus") +
  my_theme
plot_5
```



I adopted the Violin&Box plot since it shows almost all characteristics of the variable "aede2", including the median, range, and data distribution. Moreover, it is easier to understand visually.

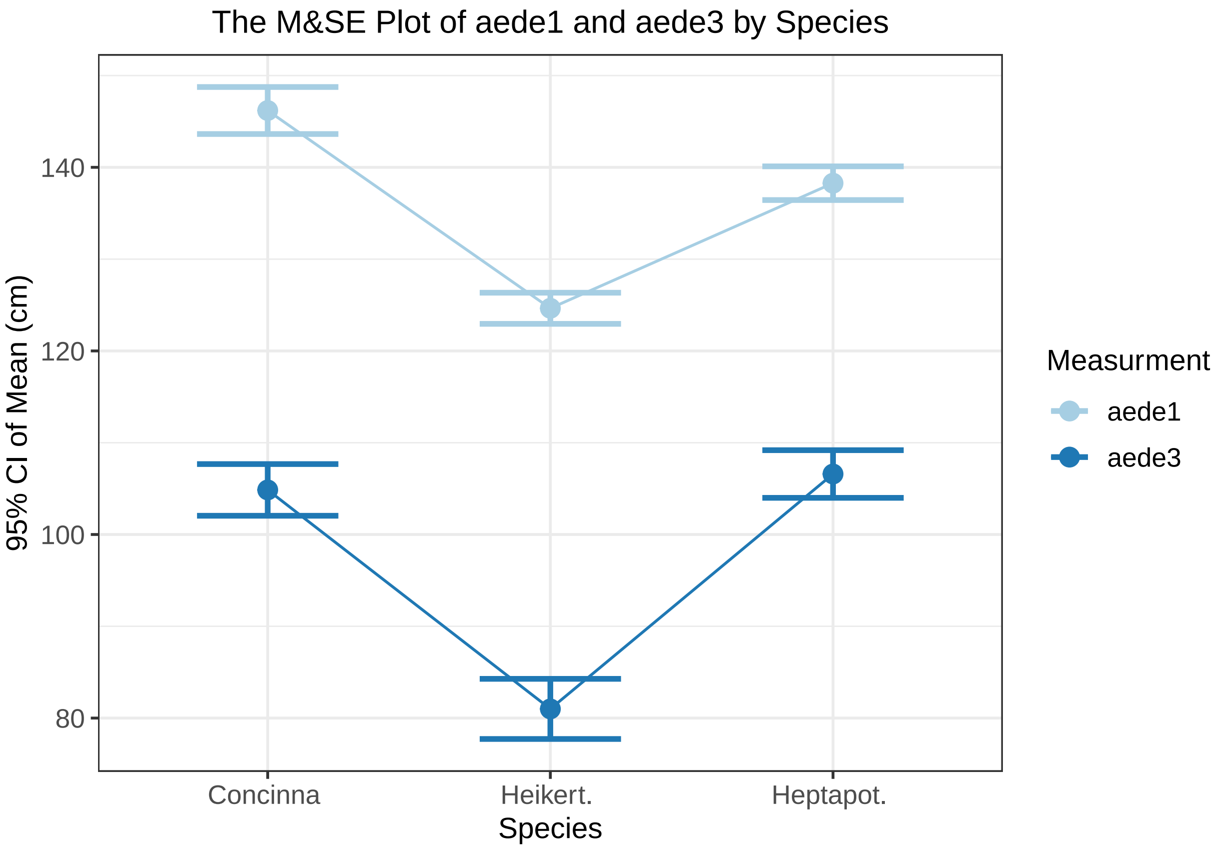
Q6: Using the `class` data set, create a bar plot (either stacked, grouped, or segmented) of favorite music genre (`music`) grouped by if the student is a statistics major (`major`). Choose the visual you think best presents the data, and ensure colors, labels, and themes make the plot easy to understand.

```
setwd("/Users/zach0422/Desktop/STAT3280/data/")
load("class.rdata")
class1 <- class%>%
  filter(major == "0" | major == "1")%>% # since there is one observation in
  #which major = 2, I deleted it in order to keep the data consistent and avoid
  #misunderstanding in the plot
  mutate(major = as.character(major), na.rm = T)

my_theme2 <- theme_bw() +
  theme(axis.text = element_text(size = 7),
        axis.title = element_text(size = 12),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 12),
        plot.title = element_text(size = 14, hjust = 0.5),
        axis.text.x = element_text(angle = 45))

plot_6 <- ggplot(class1) +
  geom_bar(aes(x = music, fill = major), position = "stack", color = "black") +
  scale_fill_brewer("major", palette = "Paired") +
  labs(x = "Favourite Music Genre", y = "Count in STAT 3280",
       title = "Favourite Music Genre by Stat Major/Non-Major in STAT3280") +
  my_theme2

plot_6
```



Q7: Using the `flea` data set, create a mean and standard error plot of `aede1` and `aede3` by `species`. The error bars should represent the standard deviation of each variable. Ensure colors, labels, and themes make the plot easy to understand, and avoid plots with overlapping error bars.

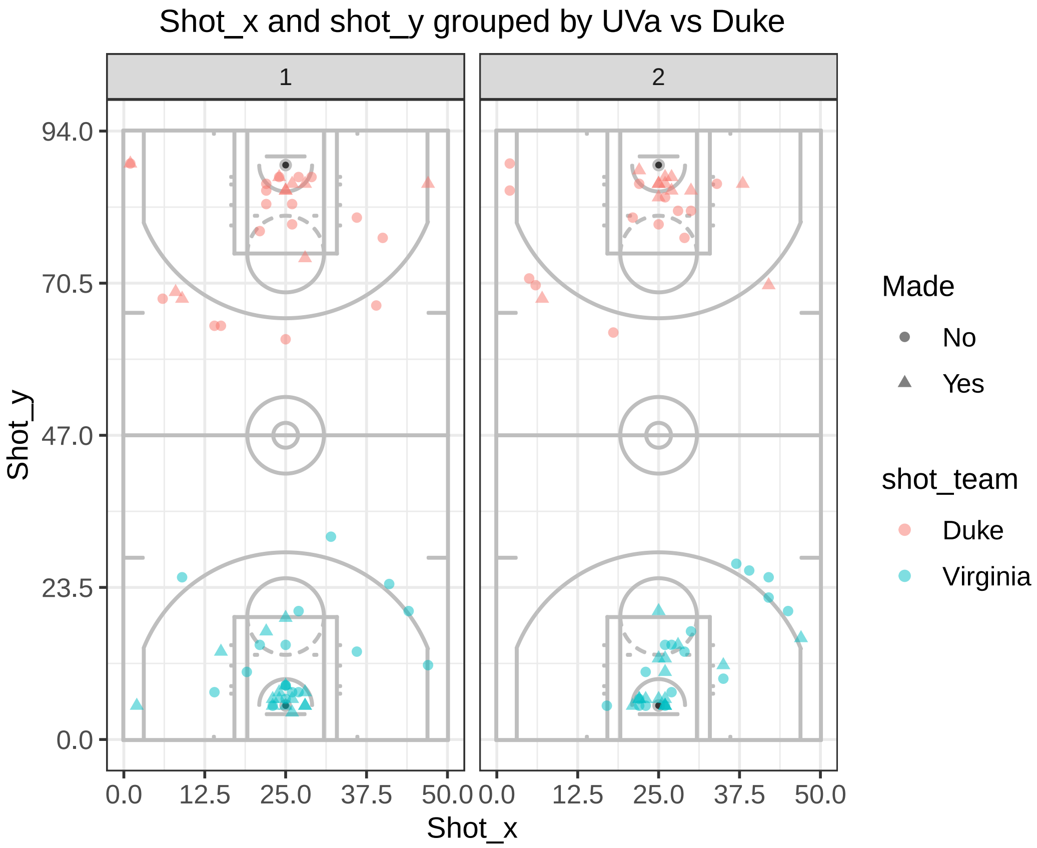
```
mse_flea <- flea %>%
  pivot_longer(cols = c("aede1", "aede3"), names_to = "Measurement",
               values_to = "Value") %>%
  group_by(species, Measurement) %>%
  summarise(n = n(), mean = mean(Value), se = sd(Value) / sqrt(n),
            ci95lwr = mean - se * qt(0.975, n - 1),
            ci95upr = mean + se * qt(0.975, n - 1))

plot_7 <- ggplot(mse_flea) +
  geom_point(aes(x = species, y = mean, color = Measurement), size = 3) +
  geom_errorbar(aes(x = species, ymin = ci95lwr, ymax = ci95upr,
                    color = Measurement), width = 0.5, size = 1) +
  geom_line(aes(x = species, y = mean, group = Measurement,
                color = Measurement)) +
  scale_color_brewer("Measurment", palette = "Paired") +
  labs(x = "Species", y = "95% CI of Mean (cm)",
       title = "The M&SE Plot of aede1 and aede3 by Species") +
  my_theme

plot_7
```



Q8: Using the `UVA_Duke_020722` data set, create a scatter plot of `shot_x` and `shot_y` grouped by `shot_team`. Indicate the `shot_outcome` by changing the point shape, and facet the plot by the half of the game, `half`. Overlay your scatter plot on the `draw_court` function, which should take the place of the original `ggplot()` command. The initial code is written below.

```
# Include your code for Q8 here

# First line is my personal directory, but you will need to change it
#   to source the draw_court() function from the saved location.
#   The code can be found on Collab.

setwd("/Users/zach0422/Desktop/STAT3280/data/")
source("draw_court.R")
load("UVA_Duke_020722.rdata")

UVA_Duke_020722 <- UVA_Duke_020722%>%
  mutate(Made = ifelse(shot_outcome == "made", "Yes", "No"))
```

```
plot_8 <- draw_court() +
  geom_point(data = UVA_Duke_020722, aes(x = shot_x,
                                         y = shot_y,
                                         color = shot_team,
                                         shape = Made),
                                         alpha = 0.5) +
  facet_wrap(~half) +
  labs(x = "Shot_x",
       y = "Shot_y",
       title = "Shot_x and shot_y grouped by UVa vs Duke") +
  my_theme

plot_8
```


Shot_x and shot_y grouped by UVa vs Duke
Shot_y
Shot_x
Made
● No
▲ Yes
shot_team
● Duke
● Virginia