Exploring PSU Women's Volleyball and OSU Women's Volleyball Data

Lily Mayer, Ayla Orona, Kate Miller

2023-11-29

Introduction

The following document aims to provide details on Penn State Women's Volleyball data.

The General Rules of Collegiate Volleyball

Volleyball is a game that consists of two teams each with six players hitting a ball over a net to score points. Each play starts by a setter on one team hitting the ball over the net. After the ball is served, each team is allowed to touch the ball three times before the ball must go over the net, however, a single player may not hit the ball two times in a row. Throughout the game, when the ball hits the ground of an opposing teams side, points are given to the team that last hit the ball. Based on whichever team scored the point, they receive the ball to serve the next play.

Each game consists of five sets. In order to win a set a team must score 25 points. One score is equal to one point. To win the whole game, a team must win three out of the five sets.

Research Questions

What is the connection between amount of points a player scores versus their height?
What are the average points made for each position?
What are the average errors made for each position?
What is the relationship between a players height and their position?
What is the average number of sets played based on college year?
How does Penn State compare to Ohio State in: total attacks per set versus kills per set, kills per set versus aces per set, and total errors per player?

Data Origin and Data Wrangling Process

We gathered our data online from the PSU Women's volleyball team rosters. We used a general 2023 team roster that contained each players position, height, number, school year, along with a stats roster of each players stats for the 2023 season. The stats roster that we used did not have the position of each player, so in order to create a useful data table we merged the general team roster with the stats roster. This created a table so each player was represented with their name, number, position, school year, and the rest of their volleyball 2023 stats. With this table we could then group it how we needed based on each research question to create visualizations and summary tables.

The roster data's original purpose was to provide a women's volleyball roster to online viewers with generic information about the player's year and position. The statistics data's purpose was to provide an online

viewer with information a player's talents on the court in the areas of kills, errors, aces, etc. We are using the data to delve deeper into the relationships between some of these variables.

For Ohio States Women's Volleyball team, we also gathered data a similar way. We gathered their Women's volleyball stats from their OSU 2023 Womens's volleyball roster. In order to create visualizations and tables to directly compare PSU and OSU, both tables needed to be joined together. Once the tables were joined, direct comparisons between PSU and OSU could be conducted.

For the total attacks per set (TA/S) visualizations, we had to make the TA/S variable by dividing the value of the TA (total attack) variable and the sets played variable for each player. We also added the "school" variable in for each player to help us when comparing volleyball stats from OSU and PSU.

For this project, we specifically focused on the attributes of height, position, total attacks, sets played, grade (age), kills, points, and errors. Each of these variables was included in the original data.

Volleyball Terms Used in This Document

The following column headings were collected from the Penn State Women's Volleyball website and may be used in the data frames, tables, and visualizations. Each are based on individual volleyball players.

Player - Penn State Women's Volleyball player first and last name
SP - Total number of sets played
MP - Total number of matches played
MS - Total number of matches where the player started the game
PTS - Total points earned for the 2023 season
PTS/S - Average number of points earned per sets played
K - Total number of kills, a kill is when a player scores a point without the ball being returned from the opposite team
K/S - Average number of kills executed per sets played
E - Total number of errors
TA - Total attacks, an attack is when a player jumps and spikes the ball downward over the net in an attempt to score a kill
PCT - Attack percentage is calculated by subtracting kills minus errors then dividing by total number of attacks
A - Total number of assists where the player threw the ball to a teammate who then won the point
A/S - Average number of assists per sets played
SA - Total number of service aces, an ace means the serving team won the point before the receiving team could return the ball over the net
SA/S - Average number of service aces per sets played
SE - Total number of service errors, meaning the serve did not throw the ball over the net or threw it out of bounds

The following are the different positions on the volleyball team, which may be referenced in a data visualization or table.

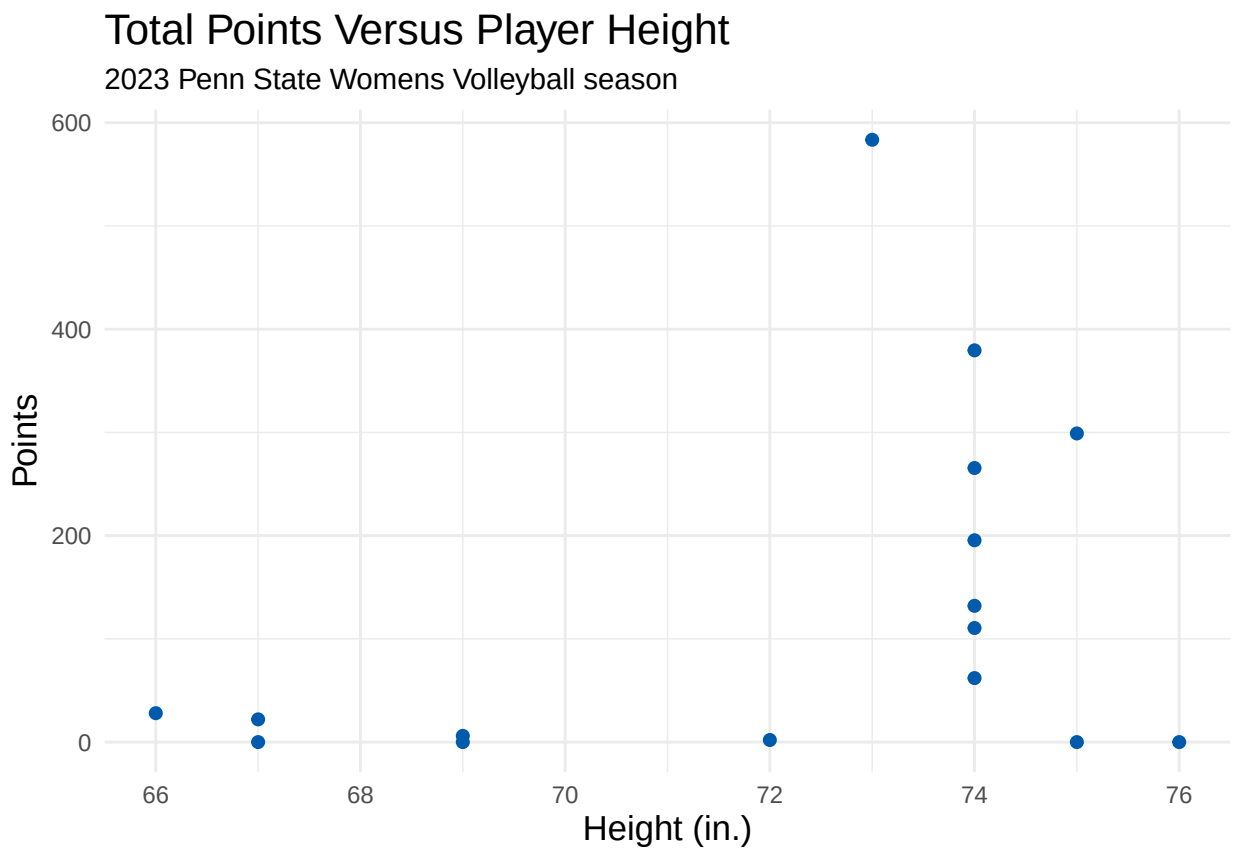
MB - Middle Blocker: The tallest person on the team. The middle blocker aims to block attacks from the other team.
L - Libero: An exclusive back-row player who is not allowed to approach the net. The libero specializes in getting digs by receiving attacks from the other team.
DS - Defensive Specialist: A substitute for any player on the court who focuses on ball passing and control.
OH - Outside Hitter: The hitter on the left side of the net. Both hitters aim to get attacks or kills.
S - Setter: A setter will 'set' the ball for a hitter to get an attack or a kill. The setter typically gets the second touch on their side
RS - Opposite Hitter: The hitter on the right side of the net. Both hitters aim to get attacks or kills.

Some players have more than one position. In these cases, the player's data will be included twice, once for each position, for comparisons that relate to the player's position.

Table of Contents for Visualizations and Tables

Conducting Exploratory Data Analysis

Figure 1



A player constitutes a case. The visualization shown above is the comparison of each players height versus their total amount of points for this season. This graph shows that there is a general positive relationship of a players height and the amount of points they have scorn, meaning as a players height increases, so does their amount of points. Also as seen in the graph, the player that has scored the most amount of points this season is 73 in (6'2 ft) with points ranging in the 500s. The players with the most amount of points range from 73 in (6'2 ft) to 75 in (6'4 ft). We can also see on the graph that the most popular height of the

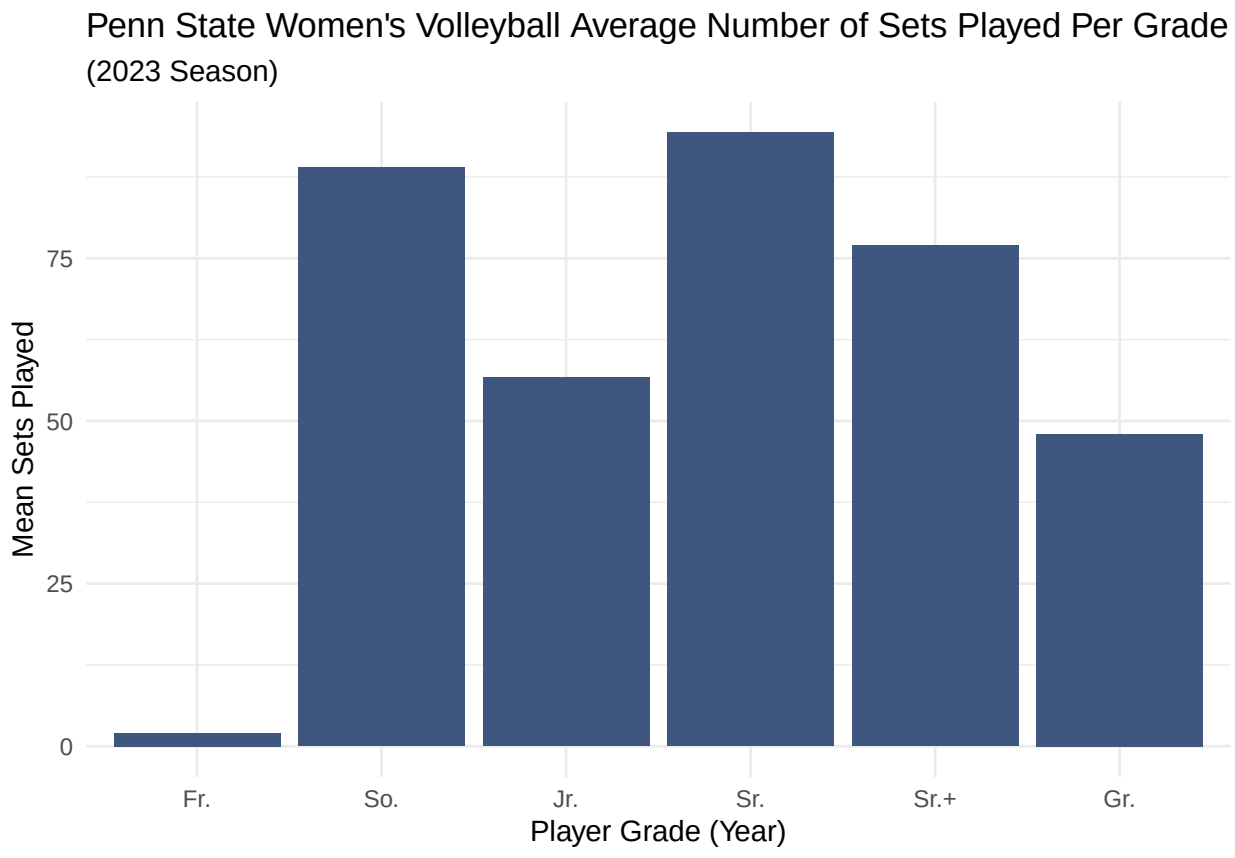
team is 74 in because they have the most amount of data points on 74 in line. Based on this graph alone, a viewer might conclude that height is a direct advantage of scoring more points, however, other variables need to be considered like, position. The closer a player is to the net the more likely they will score more points, compared to a player positioned farther from the net. This is why we will be taking many variables into consideration throughout this document.

Table 1: Summary Statistics of Average Sets Played Per Grade

Grade	Count	Minimum	First Quartile	Median	Third Quartile	Maximum	Arithmetic Mean	Arithmetic Standard Deviation
Fr.	1	2	2.00	2.0	2.0	2	2.00	NA
Gr.	3	11	12.50	14.0	66.5	119	48.00	61.51
Jr.	3	18	35.00	52.0	76.0	100	56.67	41.20
So.	2	58	73.50	89.0	104.5	120	89.00	43.84
Sr.	6	3	98.25	114.5	119.5	120	94.33	45.82
Sr.+	1	77	77.00	77.0	77.0	77	77.00	NA

Table 1 A player constitutes a case. The summary table shown above provides the 5 number summary statistics for number of sets played based on grade (year in school). It also displays the amount of players in each grade, as well as the mean and standard deviation for the number of sets played in each grade. From the summary table, we can see that the freshman had the lowest number of sets played, at 2 sets. The sophomores and seniors had the same maximum number of sets played at 116, though the seniors had a higher mean of total sets played. We can also see that the seniors have the greatest number of starters.

Figure 2



A player constitutes a case. From this data visualization, the viewer can see that the mean number of sets played was lowest for the freshman compared to the other class groups. The mean number of sets played was highest for the seniors, meaning that on average, the seniors played the most sets overall. An important item to note is that the mean number of sets played is higher for the sophomores than it is for the juniors, even though the juniors are older. In this case, the juniors don't play as much as the sophomores, on average, showing that the volleyball team does not choose who plays based on seniority.

Table 2: Summary Statistics of Average Errors Made Per Position

Position	Count	Minimum	First Quartile	Median	Third Quartile	Maximum	Arithmetic Mean	Arithmetic Standard Deviation
DS	5	0	0.00	0.0	0.00	0	0.00	0.00
L	3	0	0.00	0.0	0.00	0	0.00	0.00
MB	2	45	50.25	55.5	60.75	66	55.50	14.85
OH	5	0	29.00	53.0	86.00	178	69.20	68.53
RS	4	0	39.75	69.5	96.50	128	66.75	54.06
S	3	0	0.50	1.0	9.00	17	6.00	9.54

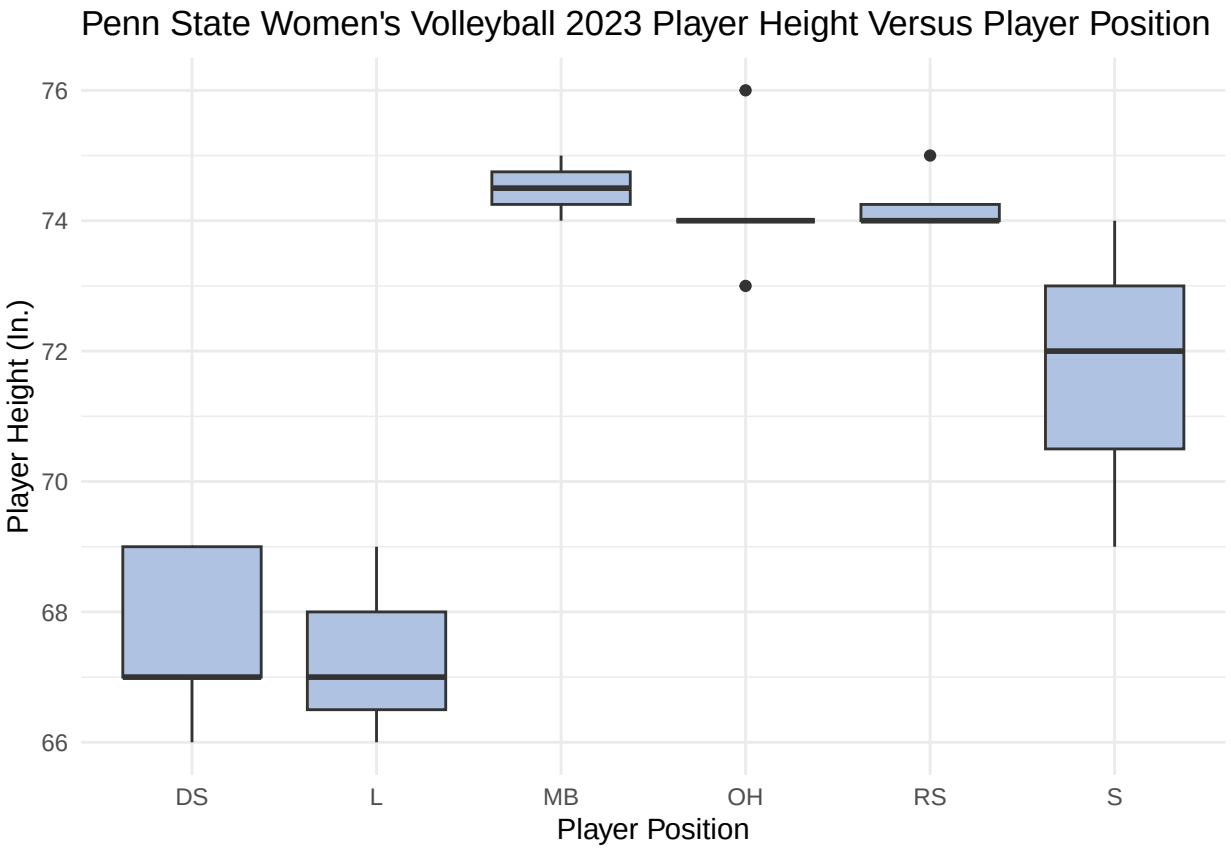
Table 2 A player constitutes a case. From this visualization, we can see that position does play a part in who makes the most errors. The Defensive Specialist (DS) and Libero (L) positions don't have any errors, and the setters (S) have minimal errors, with the maximum being 17. The Outside Hitter (OH), often the one spiking the ball and attempting to win the point, have the most errors and the highest mean number of errors. Right Side (RS) players and Middle Blockers (MB) also have many errors, since they are also right at the net. The players who are not at the net, such as DS and L, are going to have less errors since they are not trying to win the point, merely keep the ball in play.

Table 3: Summary Statistics of Average Points Earned Per Position

Position	Count	Minimum	First Quartile	Median	Third Quartile	Maximum	Arithmetic Mean	Arithmetic Standard Deviation
DS	5	0.0	0.00	6.00	22.00	28.0	11.20	13.01
L	3	6.0	14.00	22.00	25.00	28.0	18.67	11.37
MB	2	265.5	273.88	282.25	290.62	299.0	282.25	23.69
OH	5	0.0	62.00	132.00	195.50	583.5	194.60	229.46
RS	4	0.0	99.00	163.75	241.50	379.5	176.75	157.80
S	3	0.0	1.00	2.00	56.25	110.5	37.50	63.23

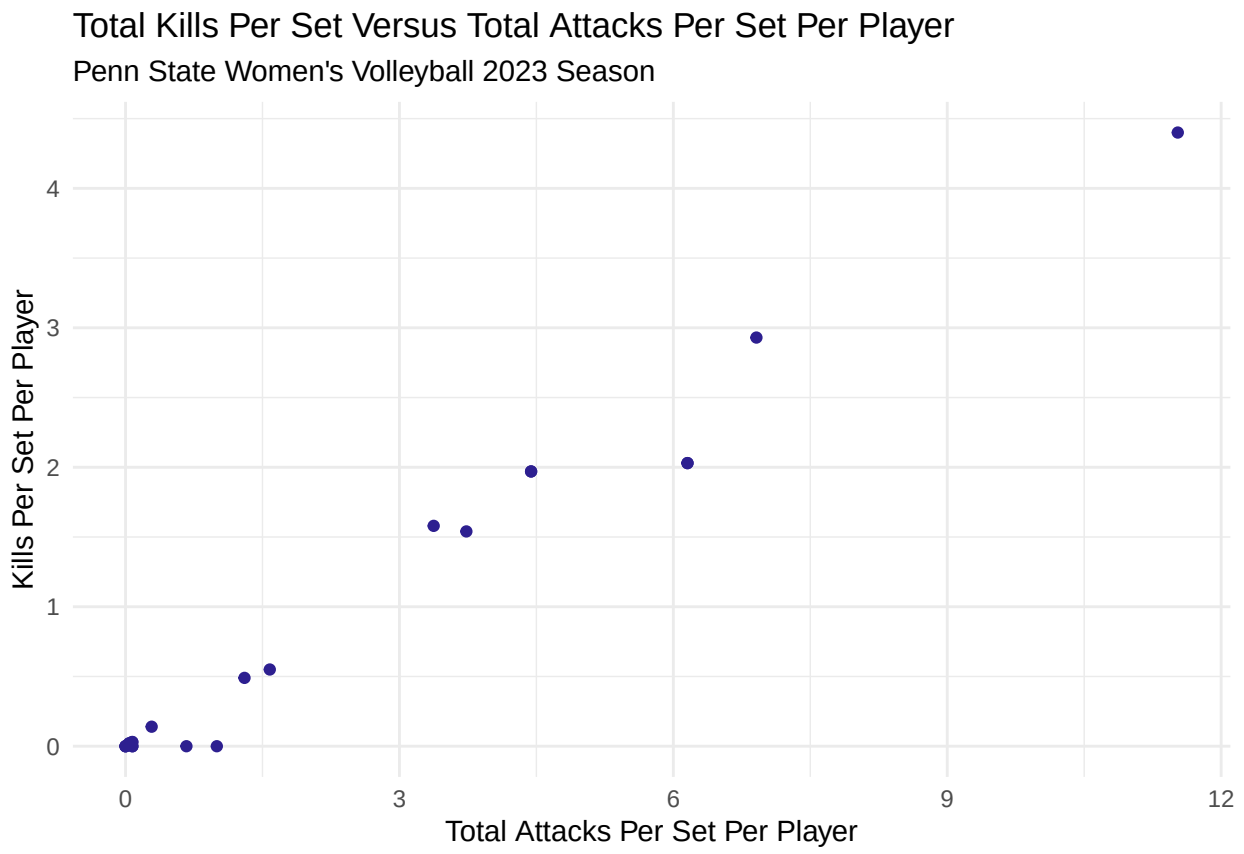
Table 3 A player constitutes a case. From the visualization above, one can gather that the number of points depends on where the player is on the court, which depends on their position. For example, while there's only two players who have middle blocker (MB) listed on their position, they have scored the highest average number of points. However, the individual who has the most number of points belongs to the outside hitter (OH), with a maximum of 567.5 points scored. The right side (RS) players also have a relatively high average of points scored, making the three positions with the highest average scores to be the MB, OH, and RS positions. We can see that the Liberos (L) and Defensive Specialist (DS) positions, as well as the setter (S) often don't score. Liberos and Defensive Specialists are often in the rear of the court, which is why they do not score points as much as the right side, outside hitter, and middle blocker, since those are the positions right at the net, where the ball is often spiked. This is depicted in the summary table.

Figure 3



A player constitutes a case. The boxplot above describes the different heights of the different players grouped by their position on Penn State's Women's Volleyball team. The shorter players on the team (depicted by a smaller inch size) are seen to be primarily the Defensive Specialist and the Libero. The tallest players are seen to be Middle Blockers, Outside Hitters, and Right Side. In this way, the tallest players are the ones closer to the net. The taller players have added height that aids them in achieving success in their specific position. Contrarily, the positions of Defensive Specialist and Libero do not necessarily require great height, since they are farther back on the court. The range in height of the Setters (S) varies greatly, showing that many different women, regardless of height, can be successful at that position.

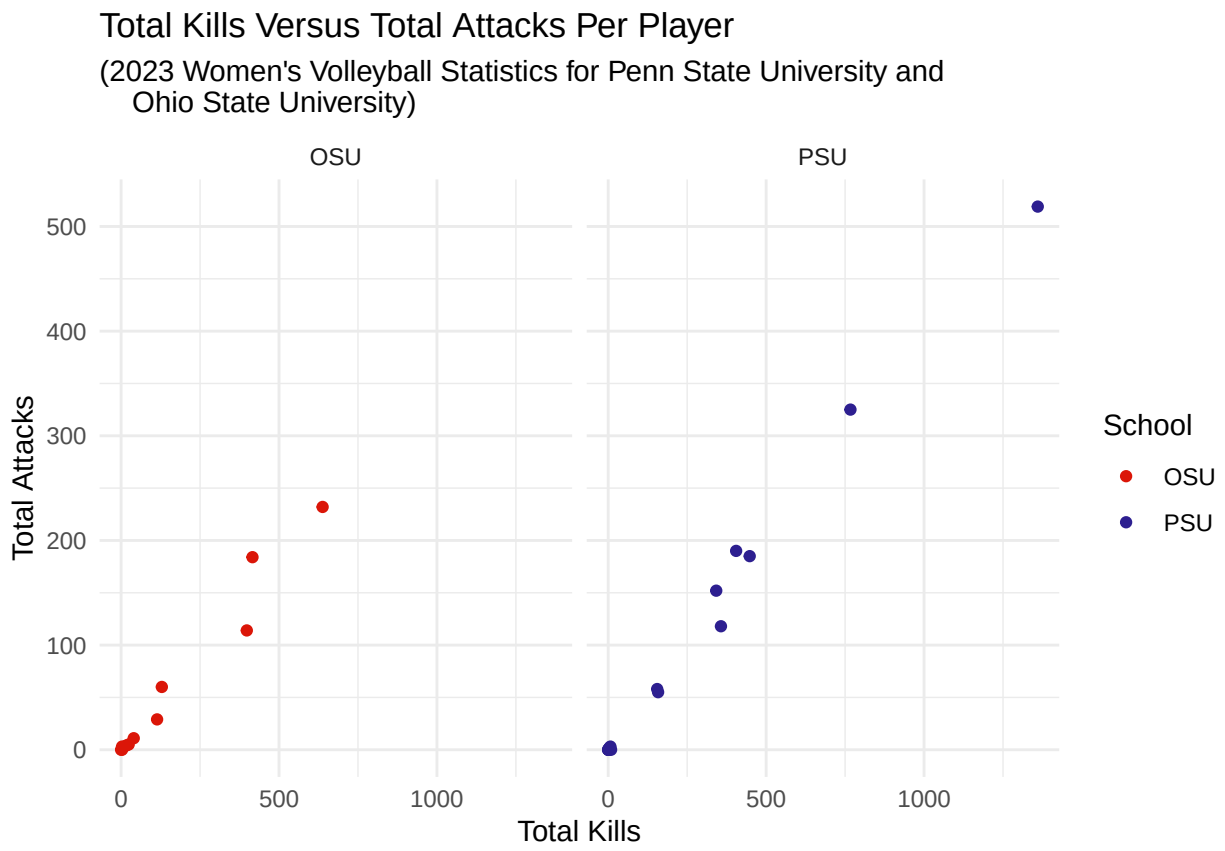
Figure 4



A player constitutes a case. The above data visualization depicts the potential correlation between kills per set per player and total attacks per set per player for Penn State's Women's Volleyball team. There is a general positive correlation between the two variables, showing that the players who have more attacks per set also tend to have more kills per set, and those who have less attacks are prone to less kills. This could be based on position, with those closer to the net often having a higher number of kills.

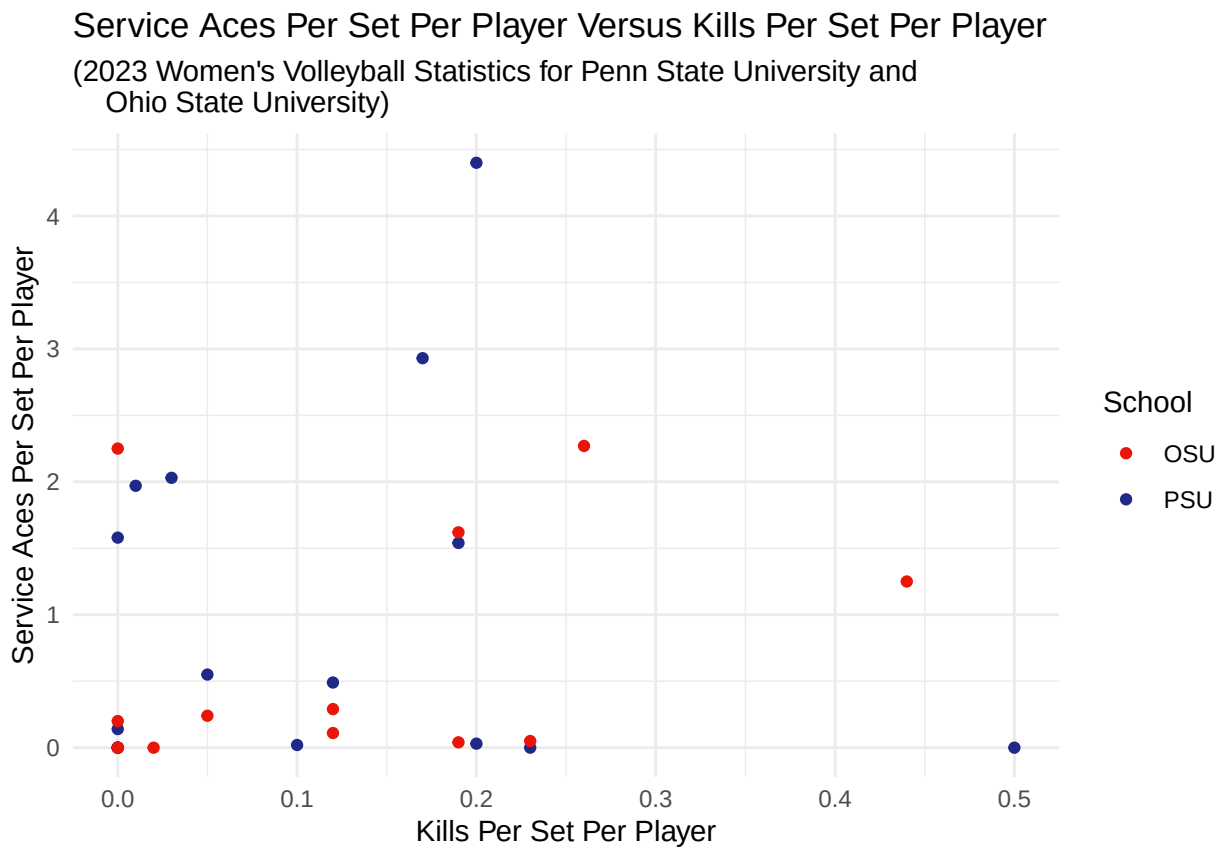
How does this data compare to other collegiate volleyball teams, specifically in Division 1, such as Ohio State University? The visualization below depicts the Total Kills Per Set Versus the Total Attacks Per Set Per Player for both teams (OSU and PSU). For this, we used the total attacks and total kills instead of the rate per set for both to see a data visualization that encompassed the conclusion of the season by providing the totals from the end of the season.

Figure 5



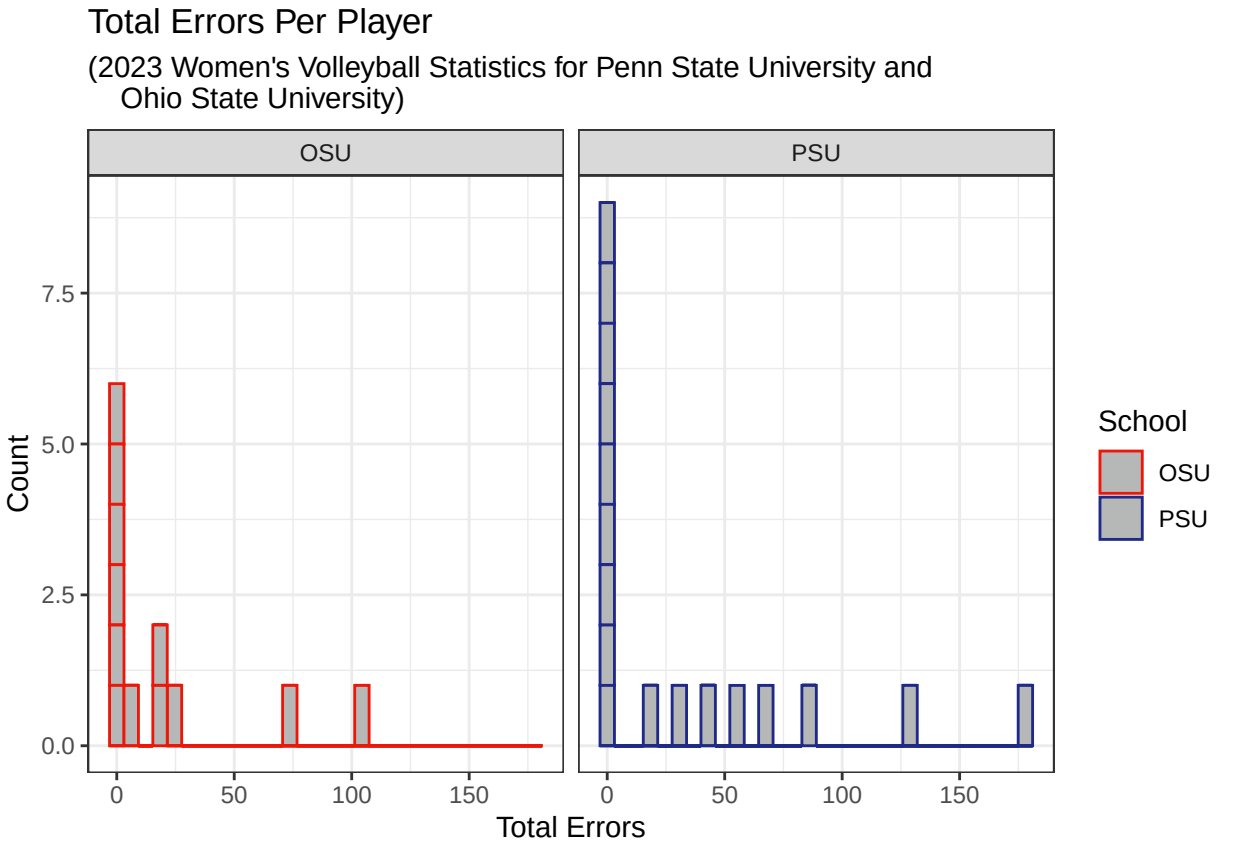
A player constitutes a case. From this visualization, the viewer can see that both teams have a general positive correlation between total attacks and total kills per player. This provides an inference that there may be strong correlation between these two variables for numerous other collegiate teams as well. As we can see, Penn State has two players who have higher total kills and higher total attacks than any Ohio State players. However, the other Penn State players appear to have very similar attacks and kills to the Ohio State players. When creating the visualization, we made sure to have the same scale for both teams.

Figure 6



A player constitutes a case. The above data visualization depicts an interesting comparison between kills per set per player and aces per set per player, showing if whether those who have more kills also have more aces, and vice versa. A kill is when the player ends the point by hitting the ball into an area that the opposing team cannot get. An ace is when the player serves the ball, but no one on the opposing team is able to receive it. We were aiming to see whether those who had more kills, who are usually more aggressive, since points often come from speed and force of the ball, also used that aggression when serving. Aces would come about from high speed of the ball, though position on the court (where the ball lands/is aimed) is also a contributing factor. As seen in the scatter plot, there is no apparent correlation between aces per set and kills per set per player. Some individuals have a high number of aces, with very few kills, and others have a high number of kills with very few aces, for both teams. Many individuals are clustered around the lower left hand corner of the visualization, showing that they didn't have many aces or kills. When comparing the two different teams (OSU and PSU), one can see that Penn State has two individuals with a higher ace per set rate than the Ohio State players, whereas Ohio State has two individuals with a higher kill per set rate than the majority of Penn State players.

Figure 7



A player constitutes a case. The above visualization shows that there are numerous Penn State players that don't have any errors, which perhaps stems from Penn State having more benched players than Ohio State, since we can see that Penn State has a higher total number of players than OSU. However, Ohio State does have more individuals with minimal errors (in comparison with PSU), with only two individuals with errors exceeding 50. Contrarily, Penn State's players are more spread out, with 5 players exceeding 50 errors and with a couple more players exceeding 25 errors than OSU's team. This is further developed with the summary table of errors for both teams below.

Table 4: Summary Statistics of Total Errors Made Per School Per Player

School	Count	Minimum	First Quartile	Median	Third Quartile	Maximum	Arithmetic Mean	Arithmetic Standard Deviation
OSU	12	0	0.75	5	22.5	107	21.92	34.51
PSU	17	0	0.00	1	53.0	178	35.47	52.68

Table 4 A player constitutes a case. The summary table displayed above aids the viewer in seeing the numerical values associated with the errors data visualization (scatter plot) from before. Penn State's player with the maximum number of errors had 178 errors, where Ohio State's player with the maximum number of errors had 107. Furthermore, Penn State had a higher arithmetic mean of errors, showing that on average, they had a higher number of errors than Ohio State did. The data for OSU is only for 12 individuals, whereas the data for PSU is for 17, as also described in the scatter plot. This is how Penn State came to have a median of 1, since many players do not play and therefore have 0 errors.

Conclusion

In total, Penn State Women's Volleyball had 2,086.5 points. From the Ohio State Women's Volleyball data, we gathered that they had 876.5 points total this season. From both data sets, we were able to compare numerous variables to help draw conclusions about collegiate women's volleyball, specifically about OSU and PSU. We learned about the correlation between position and height, kills per set and attacks per set, and numerous other relationships from the scraped data. Utilizing our skills in R, we were able to create data visualizations that aided in our and the user's understanding of the women's volleyball data.

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE, fig.align = 'left', dpi = 300)
# We used the BOAST style guide
library(rvest)
library(ggplot2)
library(knitr)
library(kableExtra)
library(stringr)
library(tidyr)
library(pdftools)
library(tidyverse)
library(esquisse)
library(janitor)
psuVBRoster <- read_html(
  x = "https://gopsusports.com/sports/womens-volleyball/roster"
) %>%
  html_elements(css = "table") %>%
  html_table()
psuVBRoster <- bind_cols(psuVBRoster[[3]])
#saves correct data frame from harvest

psuVBRoster <- psuVBRoster %>%
  tidyr::separate_wider_delim(
    cols = `Ht.`,
    delim = '-',
    names = c('Feet', 'Inches')
  ) %>%
  #separates height column into feet and inches columns
  mutate_at(vars('Feet', 'Inches'), as.numeric) %>%
  #turns feet and inches into numeric types
  mutate('Height (in.)' = Inches + (Feet * 12)) %>%
  #creates a height column in inches
  select('#', 'Full Name', 'Pos.', 'Elig.', 'Height (in.)', 'Hometown / High School')
#reorders column and removes feet and inches columns

psuRosterStats <- read_html(
  x = 'https://gopsusports.com/sports/womens-volleyball/stats'
) %>%
  html_elements(css = 'table') %>%
  html_table()
#harvests the player data from the gopsusports website
```

```r
psuRosterOffensiveHeading <- bind_cols(psuRosterStats[[2]][1, ])
#saves the column headers as a single row

psuRosterOffensiveStats <- bind_cols(psuRosterStats[[2]][2:nrow(psuRosterStats[[2]]) , ])
#saves the rest of the player data as a dataframe


View(psuRosterOffensiveStats)
colnames(psuRosterOffensiveStats) <- c(psuRosterOffensiveHeading[1, ])
#inserts column headers into player data

psuOffensive <- psuRosterOffensiveStats %>%
  select(!contains('Link'))
#removes the 'Bio Link' column

psuOffensive <- psuOffensive[1:17, ]
#removes the 'Total' and 'Opponent' rows

psuOffensive$Player <- str_sub(psuOffensive$Player, end = -25)
#removes the repeated names from the player column

psuOffensive <- psuOffensive %>%
  tidyr::separate_wider_delim(
    cols = Player,
    delim = ',',
    names = c('Last', 'First')
  ) %>%
  #Separates the player names into separate columns for first and last name
  mutate(
    'Player' = paste(First, Last, sep = ' '),
    .before = First
  ) %>%
  #concatenates the first and last names together
  mutate(
    `Player` = stringr::str_squish(`Player`)) %>%
  select(!'First') %>%
  select(!'Last')
  #removes excess whitespace and removes individual name columns

psuOffensive <- psuOffensive %>%
  mutate_at(vars(!Player), as.numeric)
#turns all columns except Player into numeric types

psuOffensiveRoster <- inner_join(
  x = psuOffensive,
  y = psuVBRoster,
  by = join_by('Player' == 'Full Name', '#' == '#')
)
#joins the roster data into the offensive statistics

psuSepPos <- separate_longer_delim(psuOffensiveRoster, 'Pos.', delim = '/')
#creates a second row for players with two positions so each row has only one position
```

```r
psuSepPos <- psuSepPos %>%
  mutate(`TA/S` = TA / SP) %>%
  mutate(across(22, round, 3)) %>%
  relocate('TA/S', .after='TA')

url <- c(
'https://s3.us-east-2.amazonaws.com/sidearm.nextgen.sites/ohiostatebuckeyes.com/stats/wvb/2023/pdf/cume
raw_text <- pdf_text(url)

#the following code was found on the internet to harvest the osu data from a pdf
clean_table <- function(raw) {
  raw <- map(raw, ~ str_split(.x, '\\n') %>% unlist())
  raw <- reduce(raw, c)

  table_start <- stringr::str_which(tolower(raw), '#')
  table_end <- stringr::str_which(tolower(raw), '31')
  table_end <- table_end[min(which(table_end > table_start))]

  table <- raw[(table_start):(table_end)]
  table <- str_replace_all(table, '\\s{2,}', '|')
  text_con <- textConnection(table)
  data_table <- read.csv(text_con, sep = '|')

  data_table
}
osuRawStats <- map_df(raw_text, clean_table)
#the osu data is very dirty, with some columns holding data from more than one column

osuFixOne <- osuRawStats[1, ] %>%
  #the first row has the name and 'SP' in the 'Player' column
  mutate('missing' = paste(Player))
#creates a new column to duplicate the broken cell
osuFixOne$missing <- str_sub(osuFixOne$missing, start = 22)
#removes the name and leaves just the number
osuFixOne$Player <- str_sub(osuFixOne$Player, end = -4)
#removes the number and leaves just the name
osuFixOne <- osuFixOne %>%
  select(!'PTS')
#removes the 'PTS' column
colnames(osuFixOne) <- c('#', 'Player', 'K', 'K.S', 'E', 'TA', 'Pct', 'A', 'A.S', 'SA',
  'SE', 'SA.S', 'RE', 'DIG', 'DIG.S', 'BS', 'BA', 'BLK', 'BLK.S', 'BE', 'BHE',
  'PTS', 'SP')
#shifts the column headers to reflect the correct data
osuFixOne <- osuFixOne %>%
  select('#', 'Player', 'SP', 'K', 'K.S', 'E', 'TA', 'Pct', 'A', 'A.S', 'SA', 'SE',
  'SA.S', 'RE', 'DIG', 'DIG.S', 'BS', 'BA', 'BLK', 'BLK.S', 'BE', 'BHE', 'PTS')
#changes the order of the columns to the original order

osuFixOthers <- osuRawStats[7:12, ] %>%
#rows 7 to 12 include both player number and name in the 'X.' column
  mutate('missing' = paste(X.))
#copies the broken cell into a new column
osuFixOthers$missing <- str_sub(osuFixOthers$missing, start = 4)
```

```r
#removes the number
osuFixOthers$X. <- str_sub(osuFixOthers$X., end = 3)
#removes the name
osuFixOthers <- osuFixOthers %>%
  select(!'PTS')
#removes the 'PTS' column
colnames(osuFixOthers) <- c('#', 'SP', 'K', 'K.S', 'E', 'TA', 'Pct', 'A', 'A.S', 'SA',
  'SE', 'SA.S', 'RE', 'DIG', 'DIG.S', 'BS', 'BA', 'BLK', 'BLK.S', 'BE', 'BHE',
  'PTS', 'Player')
#shifts the column names to reflect the correct data


osuGoodRows <- osuRawStats[2:6, ] %>%
  #selects the rows which were formatted correctly
  rename('#' = 'X.')
#changes the name of the player number column

osuStats <- rbind(osuFixOne, osuGoodRows, osuFixOthers)
#vertically joins the osu data back together

osuStats <- osuStats %>%
  tidyr::separate_wider_delim(
    cols = Player,
    delim = ',',
    names = c('Last', 'First')
  ) %>%
  #Separates the player names into separate columns for first and last name
  mutate(
    'Player' = paste(First, Last, sep = ' '),
    .before = First
  ) %>%
  #concatenates the first and last names together
  mutate(
    `Player` = stringr::str_squish(`Player`)) %>%
  select(!'First') %>%
  select(!'Last') %>%
  #removes the first and last name columns
  mutate_at(vars(!Player), as.numeric)
#saves all columns except for Player as a numeric type

osuJoin <- osuStats %>%
  select('#', 'Player', 'SP', 'K', 'K.S', 'E', 'TA', 'Pct', 'A', 'A.S', 'SA', 'SE',
  'SA.S', 'PTS') %>%
  #selects the columns which are in common with the psu data
  mutate('School' = 'OSU')
#adds a column to save the school of each player
colnames(osuJoin) <- c('#', 'Player', 'SP', 'K', 'K/S', 'E', 'TA', 'PCT', 'A', 'A/S',
  'SA', 'SE', 'SA/S', 'PTS', 'School')
#changes the names of the columns to match the psu data

psuJoin <- psuOffensive %>%
  select('#', 'Player', 'SP', 'K', 'K/S', 'E', 'TA', 'PCT', 'A', 'A/S', 'SA', 'SE',
  'SA/S', 'PTS') %>%
```

```
  #selects the columns which are in common with the osu data
  mutate('School' = 'PSU')
#adds a column to save the school of each player

psuAndOsu <- rbind(psuJoin, osuJoin)
#vertically joins the psu and osu data together
View(psuAndOsu)
#Hide the code until the appendix
ggplot(psuOffensiveRoster) + #Plot based on the PSU offensive roster
  aes(x = `Height (in.)`, y = PTS) +
  geom_point(shape = "circle", size = 1.8, colour = "#005BAC") +
  labs(
    x = "Height (in.)",
    y = "Points",
    title = "Total Points Versus Player Height",
    subtitle = "2023 Penn State Womens Volleyball season"
    #Addition of a subtitle allows the reader to understand exactly
    #what data is being measured
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16L),
    axis.title.y = element_text(size = 13L),
    axis.title.x = element_text(size = 13L)
  )
fig.align = "left"
setsSummaryTable <- psuOffensiveRoster %>%
  # Creates a summary table based on the wrangled and tidied data
  # from the previous code chunk.
  group_by(Elig.) %>%
  # We are grouping by eligibility (grade), meaning that the summary
  # statistics will be given for each grade.
  summarize(
  count = n(),
  minimum = min(SP),
  # Since we are looking specifically at the data for sets played, we select
  # the variable "SP" for all the statistics.
  firstQuintile = quantile(SP, 0.25),
  median = median(SP),
  thirdQuintile = quantile(SP, 0.75),
  maximum = max(SP),
  mean = mean(SP),
  standardDeviation = sd(SP)
    )

setsSummaryTable2 <- setsSummaryTable %>%
  mutate(
    across(where(is.numeric), round, 2)
    )
  # This section utilizes the mutate function to round any variable that
  # is a number to 2 decimal places.

setsSummaryTable2 %>%
```

```
  kable(
  col.names = c("Grade","Count", "Minimum", "First Quartile", "Median", "Third Quartile",
  "Maximum", "Arithmetic Mean", "Arithmetic Standard Deviation"),
  # Changes the column names to those listed above.
  caption = "Summary Statistics of Average Sets Played Per Grade",
  # Changes the title of the summary table.
  booktabs = TRUE,
  align = c("l", rep("c", 6)),
  font_size = 14
  ) %>%
  kableExtra::kable_styling(
  latex_options = c('HOLD_position', 'scale_down')) %>%
  #scales table to stay on the page
  kableExtra::kable_classic()
  # This section utilizes the kableExtra package to create a clear summary
  # table for the viewer to look at.

fig.align = "left"


ggplot(setsSummaryTable2) +
  # Utilizes ggplot to create a data visualization based on the
  # summary table created previously.
  aes(x = Elig., y = mean, group = count) +
  # Elig. (grade) will be on the horizontal axis, mean will be on the
  # vertical axis, and the data will overall be grouped by count.
  scale_x_discrete(limits = c('Fr.', 'So.', 'Jr.', 'Sr.', 'Sr.+', 'Gr.')) +
  #the bars will be ordered by increasing year
  geom_col(fill = "#3F567E") +
  # geom_col will display a bar chart.
  labs(
    x = "Player Grade (Year)",
    y = "Mean Sets Played",
    title = "Penn State Women's Volleyball Average Number of Sets Played Per Grade",
    subtitle = "(2023 Season)"
  # Sets the axes and title of the data visualization.
  ) +
  theme_minimal()
fig.align = "left"



PosErrorsSummaryTable <- psuSepPos %>%
  # Creates a summary table based on the wrangled and tidied data with object
  # name "psuSepPos" from a previous code chunk.
  group_by(Pos.) %>%
  # We are grouping by position, meaning that the summary
  # statistics will be given for each position.
  summarize(
  count = n(),
  minimum = min(E),
  # We are evaluating the number of errors made for each position,
  # which uses the variable E.
```

```
  firstQuintile = quantile(E, 0.25),
  median = median(E),
  thirdQuintile = quantile(E, 0.75),
  maximum = max(E),
  mean = mean(E),
  standardDeviation = sd(E)
    )

PosErrorsSummaryTable2 <- PosErrorsSummaryTable %>%
  mutate(
    across(where(is.numeric), round, 2)
    )
  # This section utilizes the mutate function to round any variable that
  # is a number to 2 decimal places.

PosErrorsSummaryTable2 %>%
  kable(
  col.names = c("Position","Count", "Minimum", "First Quartile", "Median",
  "Third Quartile", "Maximum", "Arithmetic Mean", "Arithmetic Standard Deviation"),
  # Changes the column names to those listed above.
  caption = "Summary Statistics of Average Errors Made Per Position",
  # Changes the title of the summary table.
  booktabs = TRUE,
  align = c("l", rep("c", 6)),
  font_size = 14
  ) %>%
  kableExtra::kable_styling(
  latex_options = c('HOLD_position', 'scale_down')) %>%
  #scales table to stay on the page
  kableExtra::kable_classic()
  # This section utilizes the kableExtra package to create a clear summary
  # table for the viewer to look at.
fig.align = "left"

statsSummaryTable <- psuSepPos %>%
  # Creates a summary table based on the wrangled and tidied data with object
  # name "psuSepPos" from a previous code chunk.
  group_by(Pos.) %>%
  # We are grouping by position, meaning that the summary
  # statistics will be given for each position.
  summarize(
  count = n(),
  minimum = min(PTS),
  # We are evaluating the number of points scored for each position,
  # which uses the variable PTS.
  firstQuintile = quantile(PTS, 0.25),
  median = median(PTS),
  thirdQuintile = quantile(PTS, 0.75),
  maximum = max(PTS),
  mean = mean(PTS),
  standardDeviation = sd(PTS)
    )
```

```r
statsSummaryTable2 <- statsSummaryTable %>%
  mutate(
    across(where(is.numeric), round, 2)
    )
  # This section utilizes the mutate function to round any variable that
  # is a number to 2 decimal places.

statsSummaryTable2 %>%
  kable(
  col.names = c("Position","Count", "Minimum", "First Quartile", "Median",
  "Third Quartile", "Maximum", "Arithmetic Mean", "Arithmetic Standard Deviation"),
  # Changes the column names to those listed above.
  caption = "Summary Statistics of Average Points Earned Per Position",
  # Changes the title of the summary table.
  booktabs = TRUE,
  align = c("l", rep("c", 6)),
  font_size = 14
  ) %>%
  kableExtra::kable_styling(
  latex_options = c('HOLD_position', 'scale_down')) %>%
  #scales table to stay on the page
  kableExtra::kable_classic()
  # This section utilizes the kableExtra package to create a clear summary
  # table for the viewer to look at.
fig.align = "left"


ggplot(psuSepPos) +
  # Utilizes ggplot to make a boxplot for the heights compared to the
  # players' positions.
  aes(x = Pos., y = `Height (in.)`) +
  geom_boxplot(fill = "#B0C2E2") +
  labs(
    x = "Player Position",
    y = "Player Height (In.)",
    title = "Penn State Women's Volleyball 2023 Player Height Versus Player Position"
    # Gives the axes labels and title.
  ) +
  theme_minimal()
fig.align = "left"


# Utilizes the ggplot package to make a scatter plot of the data.
ggplot(psuSepPos) +
  # The two variables are Total Attacks Per Set and Total Kills Per Set.
  aes(x = `TA/S`, y = `K/S`)+
  geom_point(shape = "circle", size = 1.5, colour = "#2D1F90") +
  labs(
    x = "Total Attacks Per Set Per Player",
    y = "Kills Per Set Per Player",
    title = "Total Kills Per Set Versus Total Attacks Per Set Per Player",
    subtitle = "Penn State Women's Volleyball 2023 Season"
  ) +
```

```
  theme_minimal()
fig.align = "left"

# Utilizes ggplot to create a data visualization from the psuAndOsu combined data.
ggplot(psuAndOsu) +
  aes(x = TA, y = `K`, colour = School) +
  geom_point(shape = "circle", size = 1.5) +
  scale_color_manual(
    values = c(OSU = "#DB1608",
    PSU = "#2D1F90")
  ) +
  labs(
    # Includes the axis and label titles.
    x = "Total Kills",
    y = "Total Attacks",
    title = "Total Kills Versus Total Attacks Per Player",
    subtitle = "(2023 Women's Volleyball Statistics for Penn State University and
    Ohio State University)",
    color = "School"
  ) +
  theme_minimal()+
  facet_wrap("School")
fig.align = "left"
# The facet was school so that the viewer could see the visualizations side by side
# for proper comparison.

ggplot(psuAndOsu) +
  aes(x = `SA/S`, y = `K/S`, colour = School) +
  geom_point(shape = "circle", size = 1.5) +
  scale_color_manual(
    values = c(OSU = "#E91506",
    PSU = "#1F2987")
  ) +
  labs(
    x = "Kills Per Set Per Player",
    y = "Service Aces Per Set Per Player",
    title = "Service Aces Per Set Per Player Versus Kills Per Set Per Player",
    subtitle = "(2023 Women's Volleyball Statistics for Penn State University and
    Ohio State University)",
    color = "School"
  ) +
  theme_minimal()
fig.align = "left"


# Utilizing ggplot to compare errors, grouped by player, for both schools.
ggplot(psuAndOsu) +
  aes(x = E, colour = School, group = Player) +
  # Creates a histogram with specified bin width and fill.
  geom_histogram(bins = 30L, fill = "#B5B8B7") +
  scale_color_manual(
    values = c(OSU = "#EC1708",
    PSU = "#1F2987")
```

```
  ) +
  labs(
    x = "Total Errors",
    y = "Count",
    title = "Total Errors Per Player",
    subtitle = "(2023 Women's Volleyball Statistics for Penn State University and
    Ohio State University)"
  ) +
  # Details the theme chosen (black and white) and once again the facet is school to see
  # the visualizations side by side.
  # If there was no facet, it would be hard to tell which players were from which school.
  theme_bw() +
  facet_wrap(vars(School))
fig.align = "left"
errorsSummaryTable <- psuAndOsu %>%
  # Utilizes the joint data frame of PSU and OSU data.
  group_by(School) %>%
  summarize(
  count = n(),
  minimum = min(E),
  # We are evaluating the number of errors made for each team,
  # which uses the variable E.
  firstQuintile = quantile(E, 0.25),
  median = median(E),
  thirdQuintile = quantile(E, 0.75),
  maximum = max(E),
  mean = mean(E),
  standardDeviation = sd(E)
    )

errorsSummaryTable2 <- errorsSummaryTable %>%
  mutate(
    across(where(is.numeric), round, 2)
    )
  # This section utilizes the mutate function to round any variable that
  # is a number to 2 decimal places.

errorsSummaryTable2 %>%
  kable(
  col.names = c("School","Count", "Minimum", "First Quartile", "Median",
  "Third Quartile", "Maximum", "Arithmetic Mean", "Arithmetic Standard Deviation"),
  # Changes the column names to those listed above.
  caption = "Summary Statistics of Total Errors Made Per School Per Player",
  # Changes the title of the summary table.
  booktabs = TRUE,
  align = c("l", rep("c", 6)),
  font_size = 14
  ) %>%
  kableExtra::kable_styling(
  latex_options = c('HOLD_position', 'scale_down')) %>%
  #scales table to stay on the page
  kableExtra::kable_classic()
  # This section utilizes the kableExtra package to create a clear summary
```

```
  # table for the viewer to look at.
fig.align = "left"
# This is the code chunk for the code appendix so viewers can view all the code
# at the end of the RMD file
```