Sports Analytics Final Project

2025-03-31

To summarize here is a list of cleaned data sets merged with roster data for 2024 games:

- `master_df` contains every Texas attack recorded in the season
- `by_match_df` contains hitting percentage for each player by match
- `season_df` contains hitting percentage for each player for the entire season

```r
# import roster data
roster <- read_csv("~/Senior Year/Sports Analytics/Data/roster.csv")
```

```
## Rows: 20 Columns: 6
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (4): player_name, position, class, home_state
## dbl (2): player_number, height_inches
##
## 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.
```

```r
# View(roster)
```

Texas Attack Data

The columns that will remain in the data set are as follows: player_name, player_number,input_type, output_type, position, match_date, skill, match_id, evaluation, attack_code, start_coordinate_x, end_coordinate_x, start_coordinate_y, end_coordinate_y, winning_attack, video_time, opponent, epv_in, epv_out, epv_added, height_inches, home_state, and class.

```
# Filter to only Texas Attack Data
ncaa_ut <- filter(ncaa, team=="University of Texas at Austin", skill=="Attack") |>
  # Select relevant columns prior to joining data to conserve computing power
  select(player_name, player_number,input_type, output_type, position, match_date, skill,
         match_id, evaluation, attack_code, start_coordinate_x, end_coordinate_x,
         start_coordinate_y, end_coordinate_y, winning_attack, video_time, opponent, epv_in, epv
_out, epv_added, attack_description)
sec_ut <- filter(sec,team=="University of Texas at Austin", skill=="Attack") |>
  select(player_name, player_number,input_type, output_type, position, match_date, skill,
         match_id, evaluation, attack_code, start_coordinate_x, end_coordinate_x,
         start_coordinate_y, end_coordinate_y, winning_attack, video_time, opponent, epv_in, epv
_out, epv_added, attack_description)
b12_ut <- filter(b12, team=="University of Texas at Austin", skill=="Attack") |>
  select(player_name, player_number,input_type, output_type, position, match_date, skill, match_
id, evaluation, attack_code, start_coordinate_x, end_coordinate_x,
         start_coordinate_y, end_coordinate_y, winning_attack, video_time, opponent, epv_in, epv
_out, epv_added, attack_description)
```

```
# join datasets together add in player roster info
master_df <- full_join(ncaa_ut, sec_ut) |>
  full_join(b12_ut) |>
  mutate(player_name = case_when(player_name=="Player 10" ~ "Reagan Rutherford",
                                 .default = as.character(player_name))) |>
  inner_join(select(roster,player_name, height_inches,home_state),by='player_name')
```

```
## Joining with `by = join_by(player_name, player_number, input_type, output_type,
## position, match_date, skill, match_id, evaluation, attack_code,
## start_coordinate_x, end_coordinate_x, start_coordinate_y, end_coordinate_y,
## winning_attack, video_time, opponent, epv_in, epv_out, epv_added,
## attack_description)`
## Joining with `by = join_by(player_name, player_number, input_type, output_type,
## position, match_date, skill, match_id, evaluation, attack_code,
## start_coordinate_x, end_coordinate_x, start_coordinate_y, end_coordinate_y,
## winning_attack, video_time, opponent, epv_in, epv_out, epv_added,
## attack_description)`
```

Hitting Percentage of Players by Match

```r
# hitting percentage of players by match
by_match_df <- master_df |>

  mutate(error = ifelse(evaluation=="Error", 1, 0)) |>
  group_by(player_name, match_id, opponent) |>
  summarize(attempts = n(),
         errors = sum(error, na.rm=T),
         kills = sum(winning_attack, na.rm=T),
         hitting_pctg = (kills - errors)/attempts,
         med_epv_added = median(epv_added),
         mean_epv_added = mean(epv_added),
         match_date) |>

  inner_join(select(roster,player_name, height_inches,home_state),by='player_name')
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## ℹ Please use `reframe()` instead.
## ℹ When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'player_name', 'match_id', 'opponent'. You
## can override using the `.groups` argument.
```

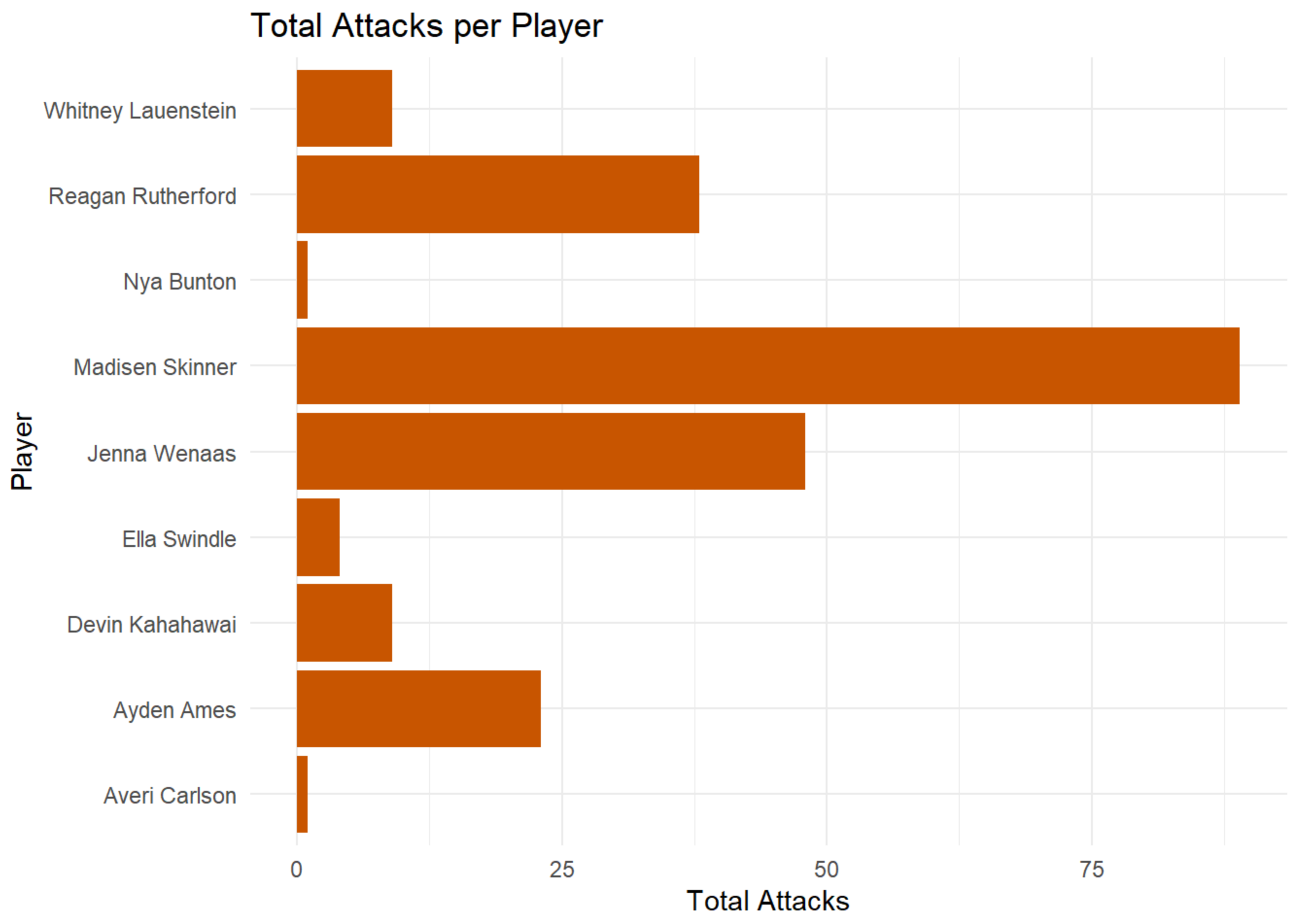
Season Hitting Percentages by Player

```r
# season hitting percentages with roster info
season_df <- master_df |>
  # select(player_name, player_number,match_id, opponent,evaluation, output_type,winning_attack,
epv_in, epv_out, epv_added) |>
  mutate(error = ifelse(evaluation=="Error", 1, 0)) |>
  group_by(player_name) |>
  summarize(attempts = n(),
            errors = sum(error, na.rm=T),
            kills = sum(winning_attack, na.rm=T),
            hitting_pctg = (kills - errors)/attempts,
            med_epv_added = median(epv_added),
            mean_epv_added = mean(epv_added)) |>
  inner_join(select(roster,player_name, height_inches,home_state),by='player_name')
```

EDA

Number of Attacks for each Texas Player

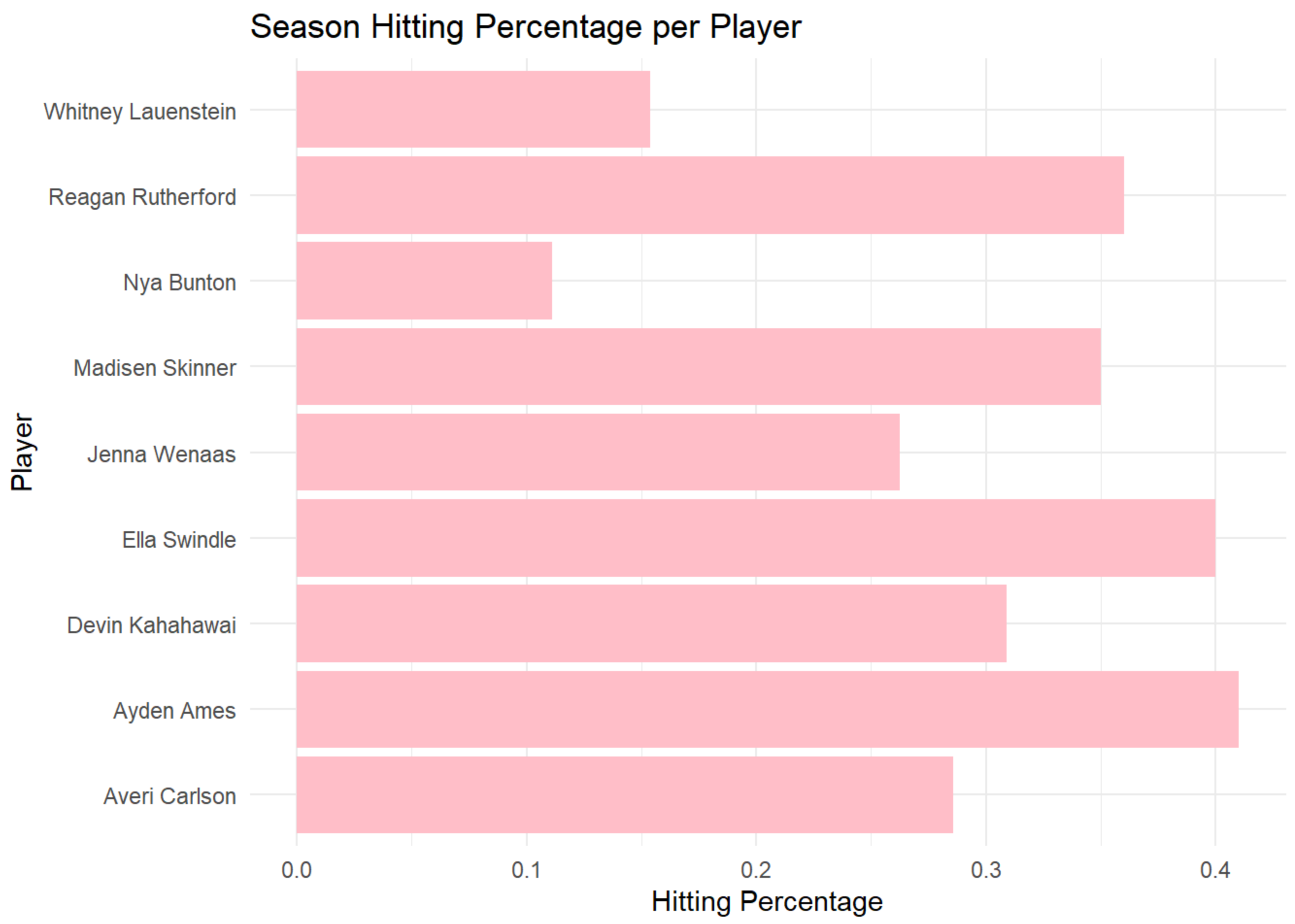
```
master_df |>
  # filter for games on 9/7/24 and 9/15/24
  filter(match_date == "2024-09-07" | match_date == "2024-09-15") |>
  group_by(player_name) |>
  summarize(n_attacks = n()) |>
  ggplot() +
    geom_bar(aes(y = player_name, x = n_attacks), stat = "identity",fill="#CC5500") +
    labs(y = "Player", x = "Total Attacks", title = "Total Attacks per Player") +
    theme_minimal()
```



For the first visualization we looked into the number of plays that were recorded for each Texas player during games that occurred on September 7, 2024 and September 15, 2024. These were games played against University of Miami and Stanford University, respectively. There were 9 athletes recorded as having attack attempts during either of these 2 games. The athlete who had the most attacks in these two games was Madisen Skinner (89) and the athletes with the least attacks who were still swinging were Nya Bunton and Averi Carlson (1).

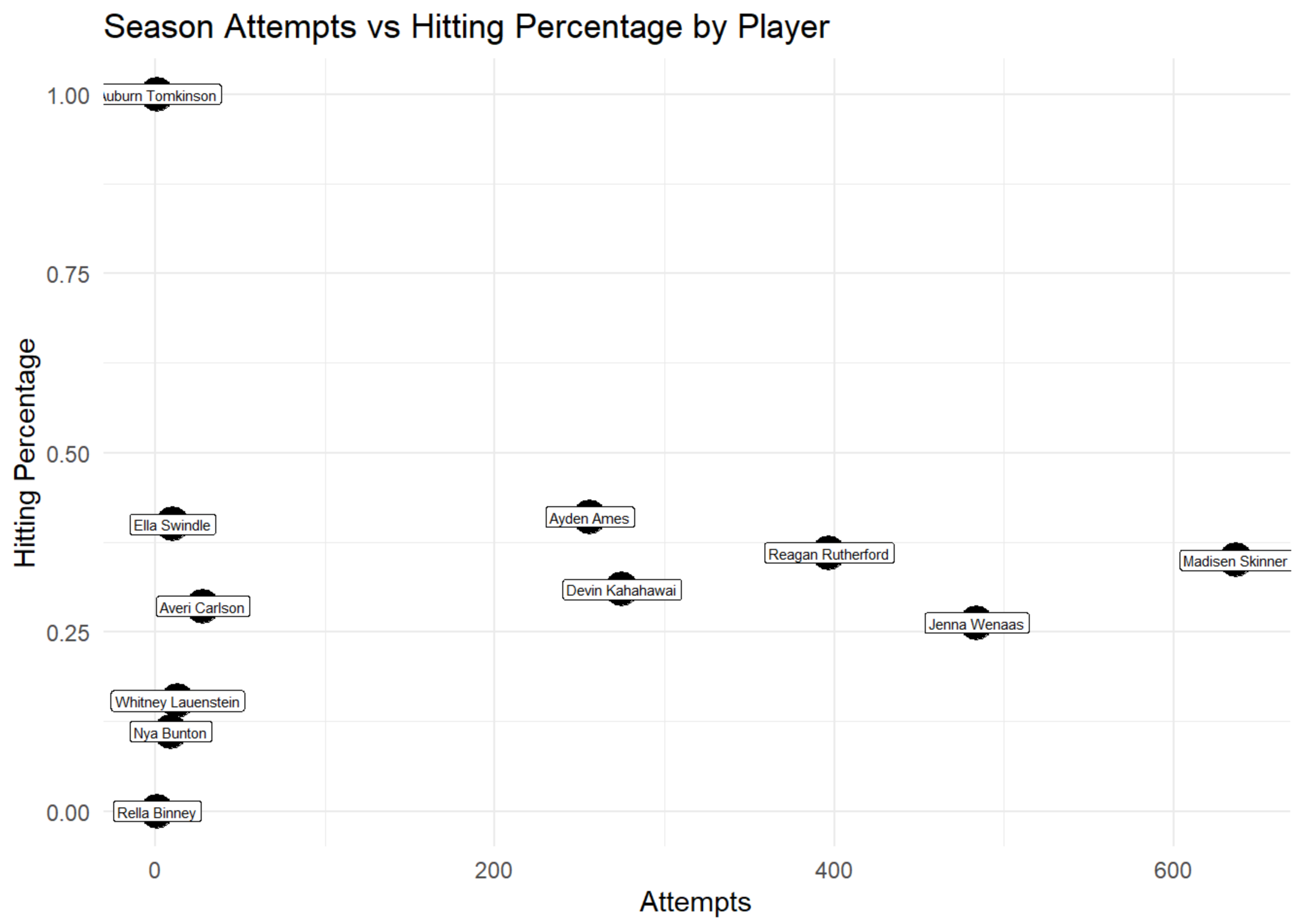
Distribution of Hitting Percentages

```
# plot hitting percentages for each player
season_df |>
  filter(attempts >= 4) |>
  ggplot() +
    geom_bar(aes(x=hitting_pctg, y=player_name), stat="identity", fill="pink") +
  labs(y = "Player", x = "Hitting Percentage", title = "Season Hitting Percentage per Player") +
    theme_minimal()
```



The athletes with the greatest season hitting percentage are Ayden Ames, Ella Swindle, and Reagan Rutherford. To make more sense of this, we will plot season hitting percentage against attempts.

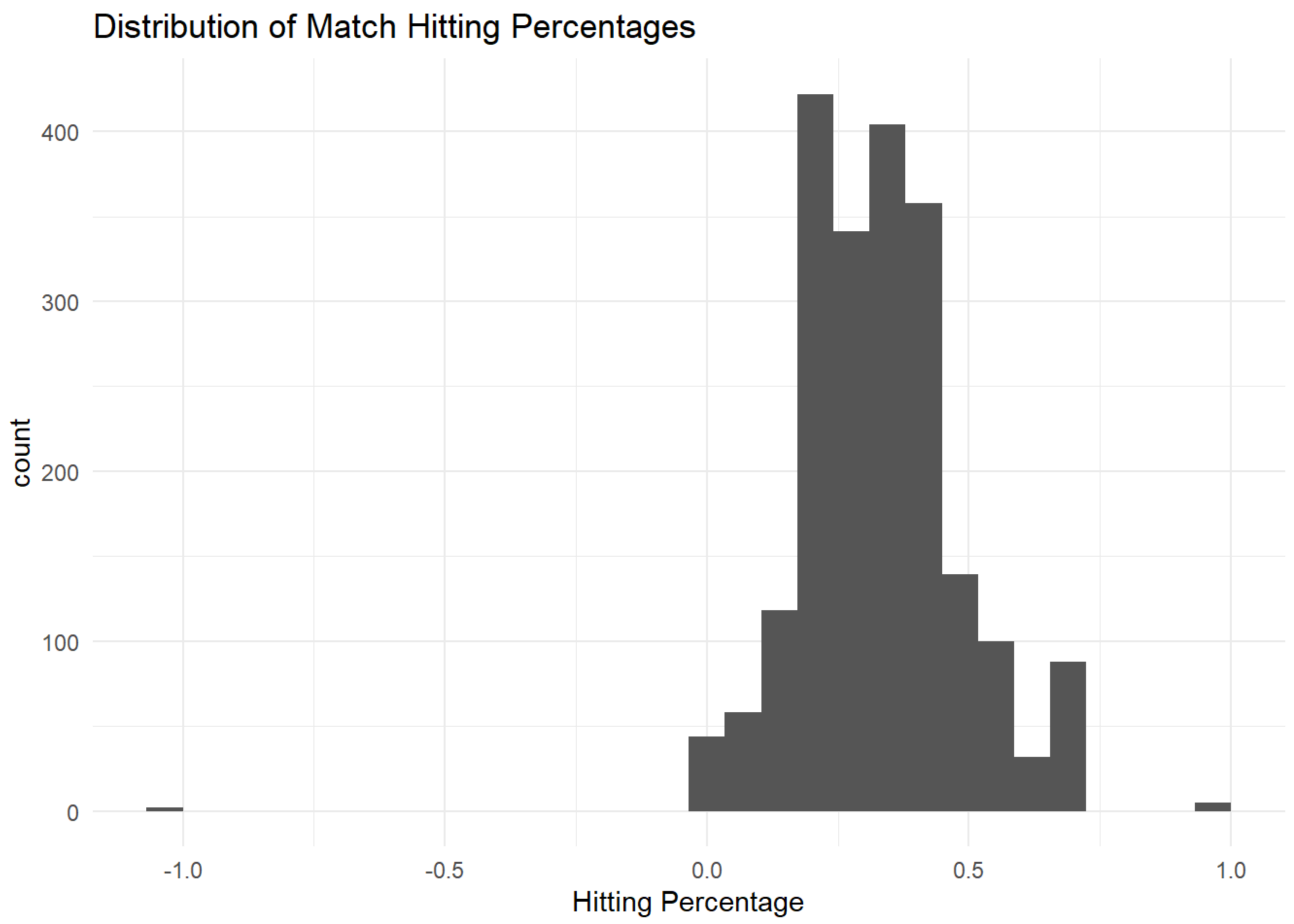
```
# Plot season hitting percentage against total attacks
season_df |>
  ggplot(aes(x = attempts, y = hitting_pctg, label = player_name), vjust = -0.5) +
  geom_point(size=6) +
  geom_label(size=2) +
  labs(y = "Hitting Percentage", x = "Attempts", title = "Season Attempts vs Hitting Percentage 
by Player") +
    theme_minimal()
```



As when attempts are greater than 200, the hitting percentage for the team ranges from 0.26 (Jenna Wenaas) to 0.41 (Ayden Ames). Madisen Skinner has the most attempts (637) and a hitting percentage of 0.35.

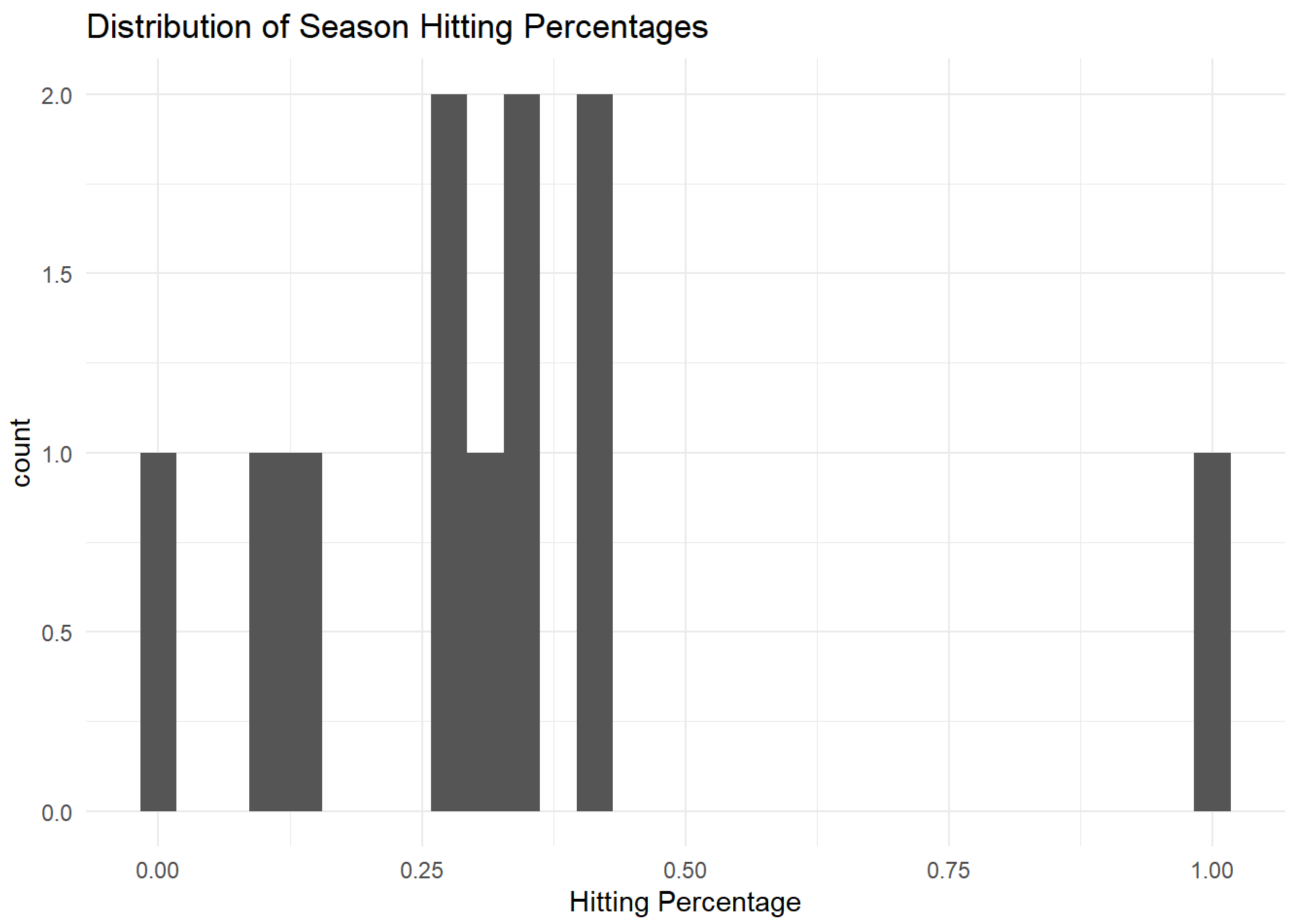
```
# Distribution of hitting percentages by match
by_match_df |>
  ggplot() +
    geom_histogram(aes(x=hitting_pctg)) +
   labs(x = "Hitting Percentage", title = "Distribution of Match Hitting Percentages") +
    theme_minimal()
```

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



```
# Distribution of hitting percentages by season
season_df |>
  ggplot() +
    geom_histogram(aes(x=hitting_pctg)) +
   labs(x = "Hitting Percentage", title = "Distribution of Season Hitting Percentages") +
    theme_minimal()
```

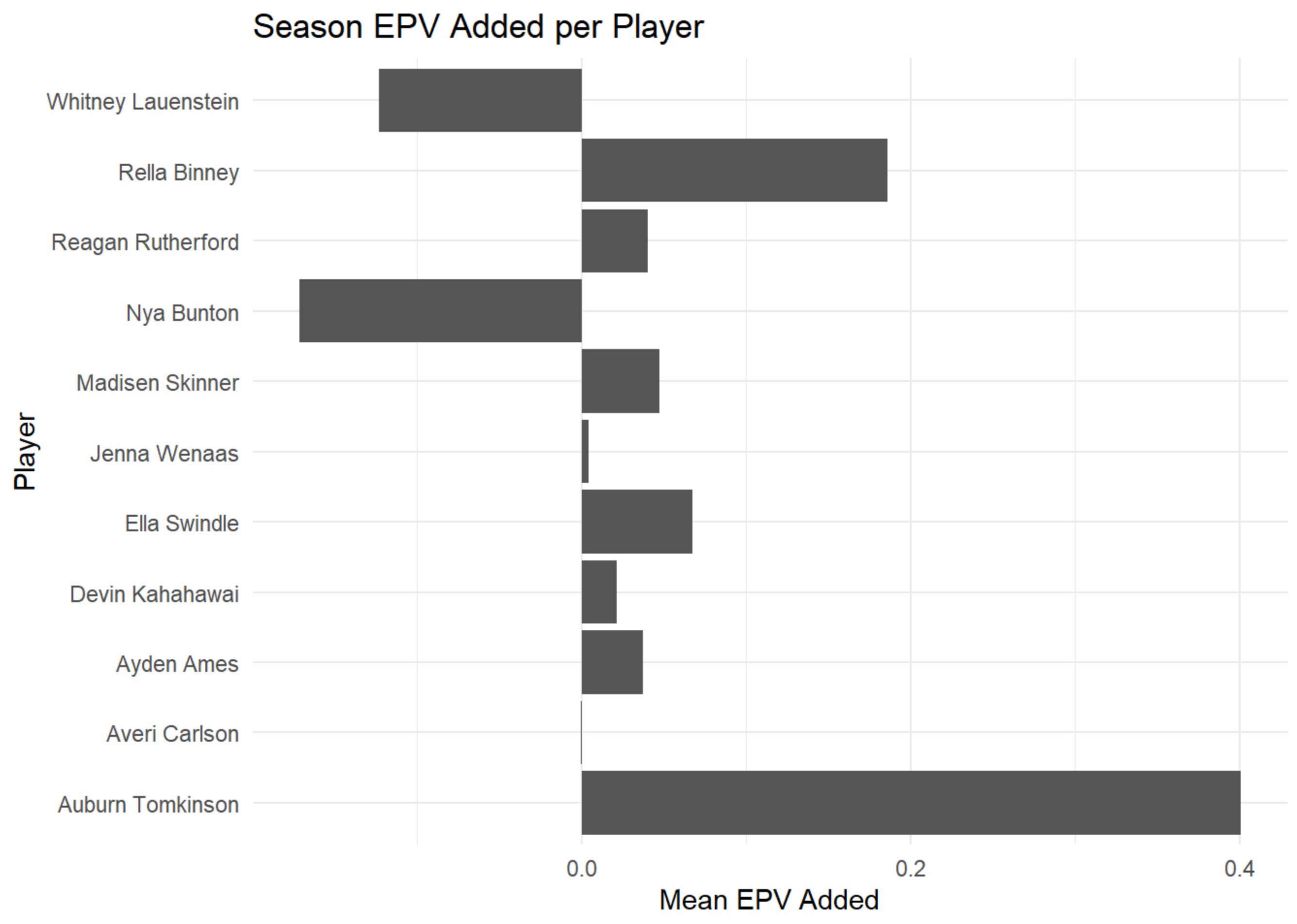
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



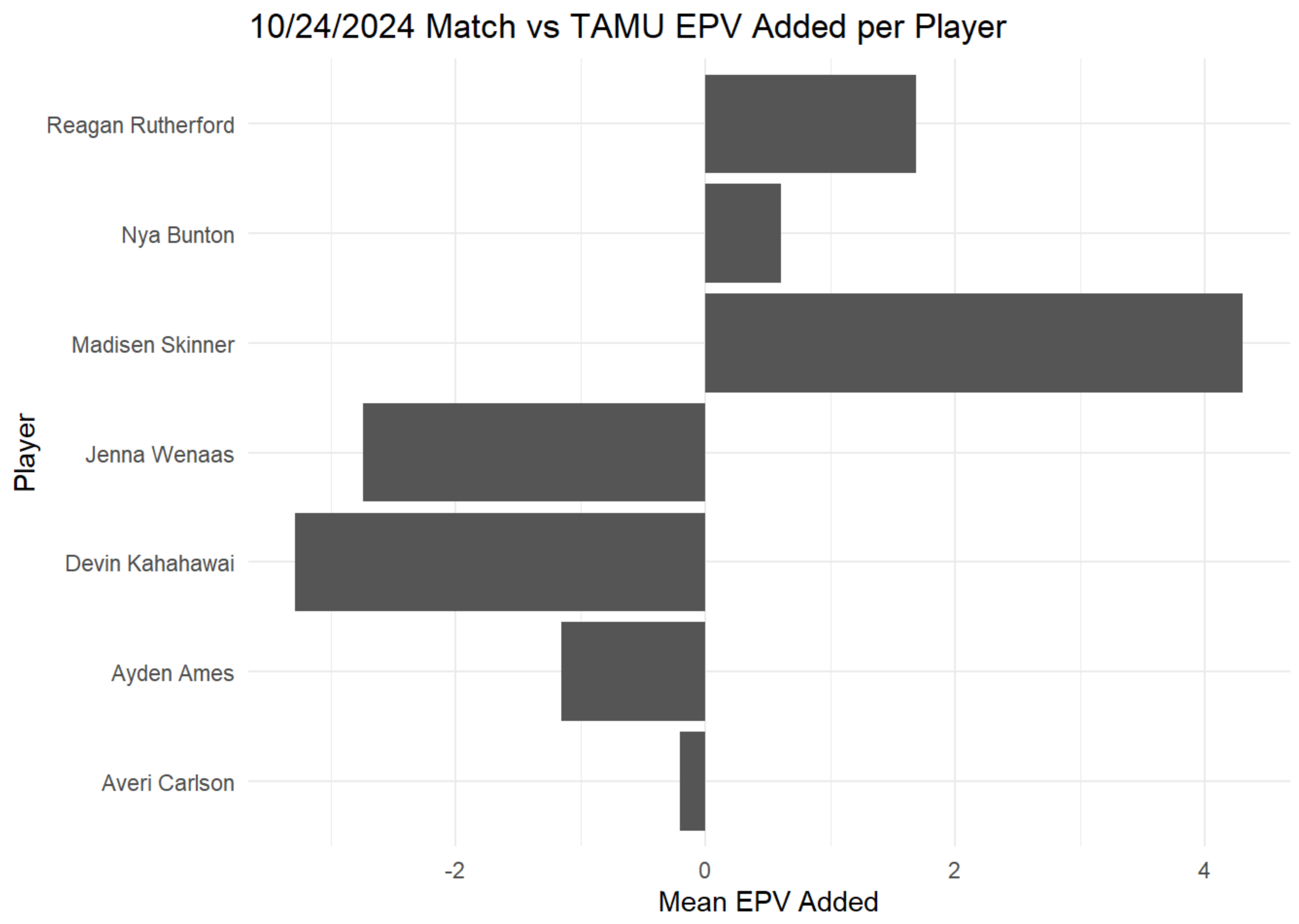
The hitting percentage by match follows a relatively normal distribution while the distribution of season hitting percentages does not.

Mean EPV added

```
# Season Mean EPV Added
season_df |>
  ggplot() +
    geom_bar(aes(x=mean_epv_added, y=player_name), stat="identity") +
  labs(y = "Player", x = "Mean EPV Added", title = "Season EPV Added per Player") +
    theme_minimal()
```

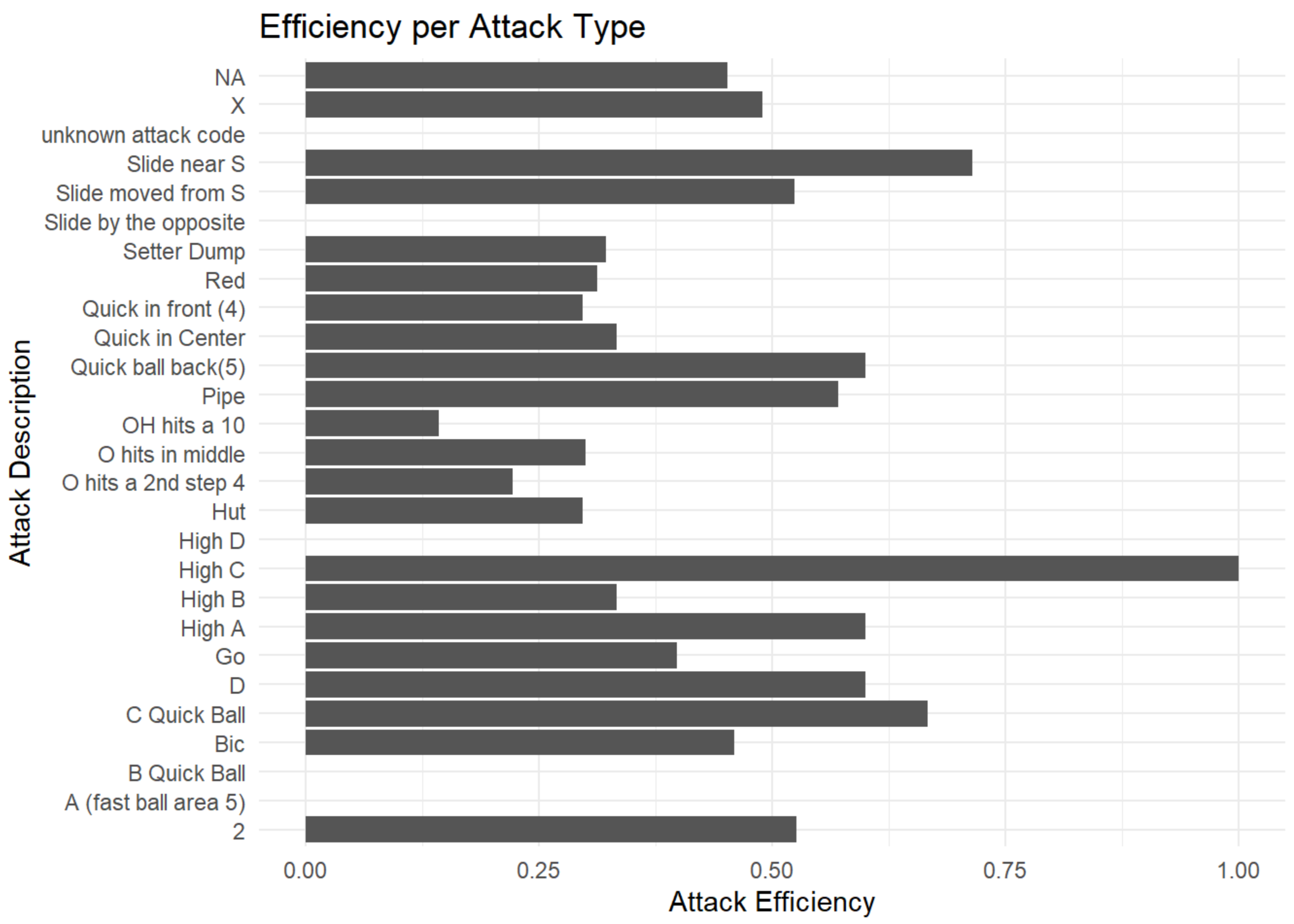


```
# Match Mean EPV Added for Game vs Texas A&M on 2024-10-24
by_match_df |>
  filter(match_date=="2024-10-24") |>
  ggplot() +
    geom_bar(aes(x=mean_epv_added, y=player_name), stat="identity") +
  labs(y = "Player", x = "Mean EPV Added", title = "10/24/2024 Match vs TAMU EPV Added per Playe
r") +
    theme_minimal()
```



Kills by Type of Attack

```
master_df |>
  mutate(kill = ifelse(evaluation=="Winning attack", 1, 0)) |>
  group_by(attack_description) |>
  summarize(efficiency = mean(kill)) |>
  ggplot() +
    geom_bar(aes(x=efficiency, y=attack_description), stat="identity") +
  labs(y = "Attack Description", x = "Attack Efficiency", title = "Efficiency per Attack Type")
+
    theme_minimal()
```



Efficiency in this context is defined as attacks that have the highest proportion of kills for attempts. The high C has an efficiency of 100% which leads to some uncertainty in this evaluation. However, we see that the slide moving toward the setter as well as the C quick ball have the greatest attack efficiency of nearly 70%. The least efficient attack is when the outside hitter hits a 10 which is a hit from the outside at the ten foot line. This means that for whatever reason, the outside hitter could not take their full approach and had to attack the ball from the 10-foot line.