Basketball Plot Stars

Nicholas Brimhall, J Cook, Grace Russell, and Matthew White

https://github.com/Hoolahups/projectr5555

Overview

Basketball enthusiasts and analysts often seek robust and intuitive tools to visualize player and team performance data quickly. This package, projectr5555, offers a straightforward and flexible way to scrape and plot NCAA basketball team statistics. Whether the user is tracking player progress, comparing team statistics, or scouting opponents, projectr5555 provides the tools users need to make informed decisions and compelling presentations. Utilizing this package, users can effortlessly compare players across multiple metrics through visual representations, allowing for quick insights and deep analysis.

Data

The get_data function takes a URL as an argument and scrapes the corresponding webpage. Many NCAA basketball teams use a format for their website identical to USU's, and so this function works for any team using that format. The function checks that the HTML scraped from the website matches an expected format, but the user should check to make sure that the desired team statistics page looks visually similar to USU's. The URL can be found by visiting the desired NCAA basketball team's athletic page and navigating to the stats page.

```
url <- "https://utahstateaggies.com/sports/mens-basketball/stats/2023-24"
usu_t2023 <- get_data(url)
```

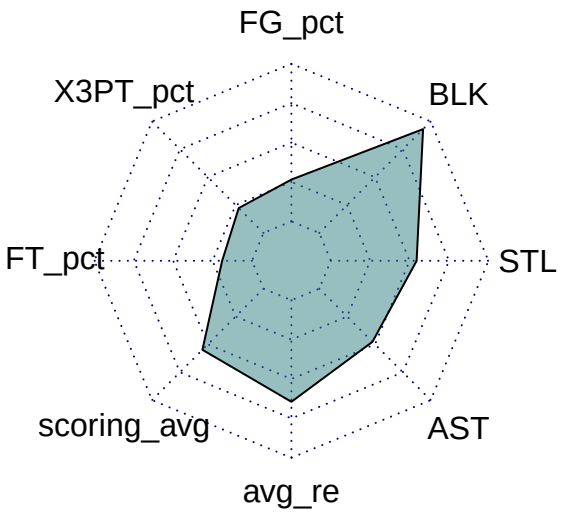
To obtain the URL for another team visit the desired NCAA basketball team's athletic page and navigating to the stats page. Select the desired season and copy the URL. The example below shows how the user can get the data for the University of Connecticut, the 2024 NCAA tournament champions, use the URL.

```
url <- "https://uconnhuskies.com/sports/mens-basketball/stats/2023-24"
uconn_t2023 <- get_data(url)
```

Plots

Star Plot This plot visualizes a specified player's statistics in a star chart, highlighting their performance across multiple areas like scoring, defense, and rebounding. There are eight default statistics. Each axis represents a different stat, providing a quick visual assessment of a player's skills. The function takes the team and players name as the first two arguments. Inside the function, every statistic value is divided by the average, so each star plot is easy to compare to one another.

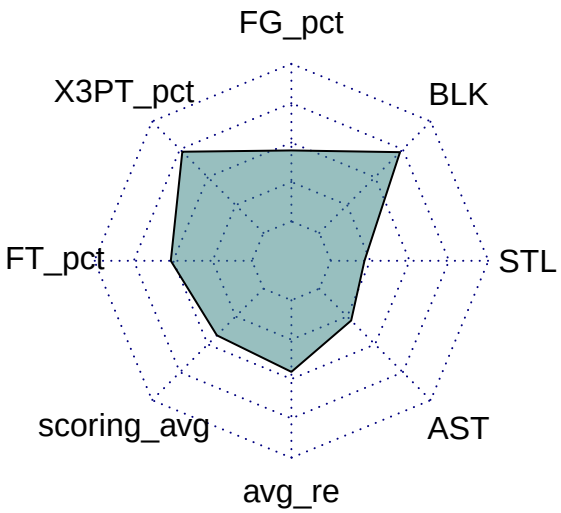
```
player_star(usu_2023, "osobor_great")
```



In the above plot, we can see that Great Osobor has a high number of blocks compared to the average value of blocks. X3PT_pct and FG_pct are closer towards the middle of the plot, so Osobor has relatively low values of number of three-pointers made and field goals made.

Multiple star plots can be to created in order to compare multiple players.

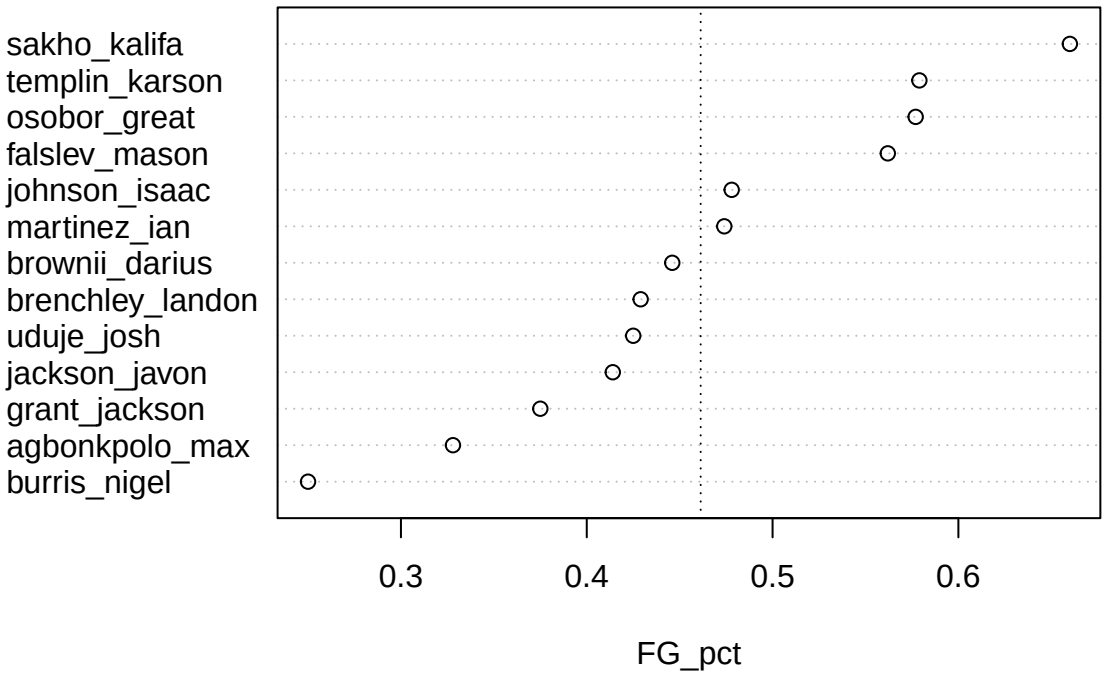
```
player_star(usu_2023, "johnson_isaac")
```



In the above plot, we can visualize a few of Isaac Johnson's statistic. Compared to Osobor we can quickly see that Johnson does not have as many blocks and rebounds as Osobor. Instead Johnson seems to do much better at making 3's and free throws by his 3 point percentage and his free throw percentage.

Dot Plot This horizontal bar plot compares a single statistic, such as field goal percentage, across all team members. It quickly identifies top performers and those needing improvement in specific areas.

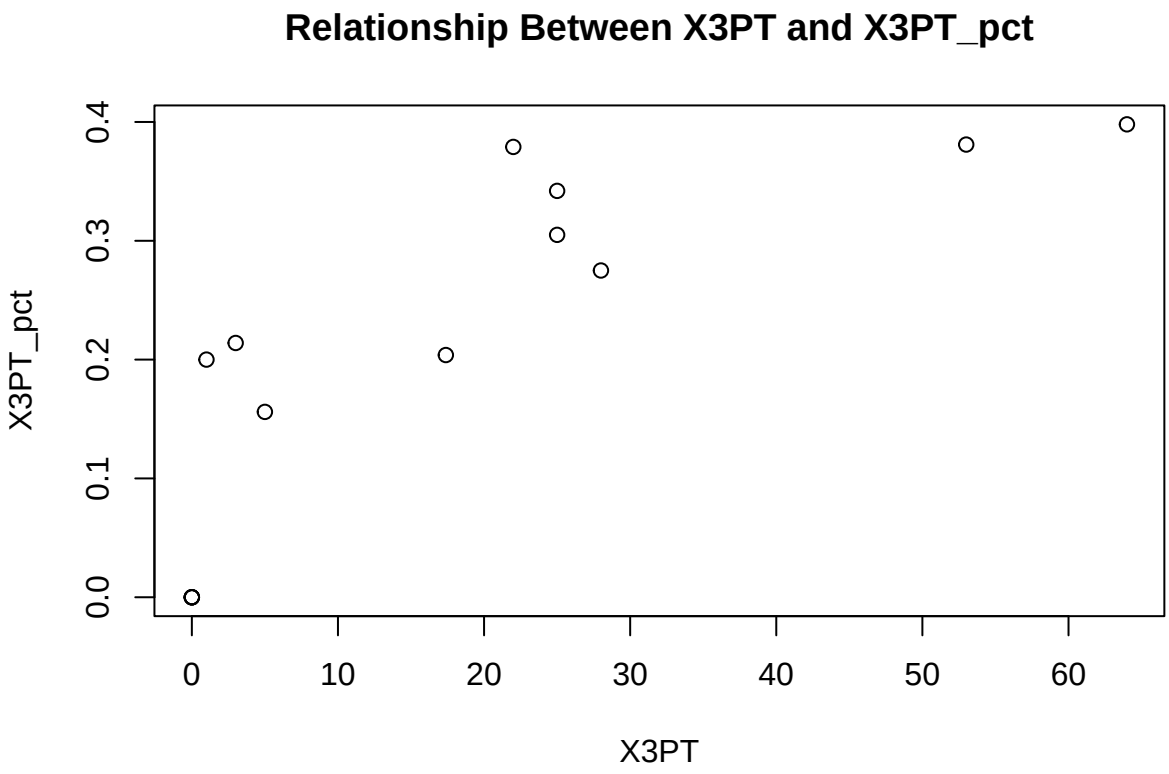
```
team_dot(usu_2023, "FG_pct")
```



In the above plot, we can see that Kalifa Sakho has the highest field goal percentage, while Nigel Burris has the lowest. The average value for the team is displayed by the dotted vertical line.

Scatter Plot This scatter plot compares two statistics to explore their relationship, such as the correlation between overall shooting efficiency and three-point shooting. It helps identify patterns that can inform coaching strategies.

```
compare_stats(usu_2023, "X3PT", "X3PT_pct")
```



In the above plot, we can see a clear positive relationship between the number of three-pointers and the percentage of three-pointers made. This may imply that players that attempt more three-pointers make a higher percentage of these shots.

Make Your Own Variables

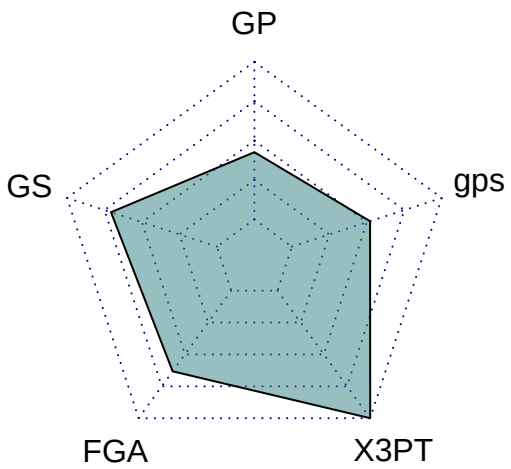
The variable_maker function takes two existing variables and combines them in a way that the user specifies. It then returns the entire list with the variables and average added to it.

```
usu_t2023 <- variable_maker(usu_2023, "GP", "GS", "sum", "gps")
```

The above line of code adds a "gps" varialbe, which is the sum of games started and games played.

Once you have your new variable, you can use it to make your own plots. Whether or not its useful is up to you, but make sure to have fun with it!

```
player_star(usu_t2023, "martinez_ian", vars = c("GP", "GS", "FGA", "X3PT", "gps"
))
```



In the above plot, we can see a star plot with the variables that we passed into the star plot function. The new "gps" variable is included.

Summary

The plotting capabilities provided by our package are invaluable for visualizing complex basketball statistics in a clear and accessible manner. These plots not only enhance understanding and presentation of player and team metrics but also aid in strategic decision-making. Whether for fan engagement, coaching strategies, or academic research, our tools offer significant benefits to diverse users.