Package Vignette

Introduction

Welcome to the stat5555sor package vignette! This package is designed to calculate the strength of record (SOR) for college basketball teams in a given season. SOR, or Strength of Record, is a measure of team performance that takes into account the quality of opponents a team has played and the outcomes of those games. By using data from both KenPom and Warren Nolan, the package calculates a custom SOR score for each team based on their game outcomes. The SOR score is calculated using a custom formula that is tailored specifically to college basketball and is designed to provide a more accurate assessment of team strength.

This package provides a comprehensive and flexible approach to calculating SOR scores for college basketball teams. Whether you're a sports analyst looking to evaluate team performance, a bettor making informed decisions, or a researcher conducting statistical analysis, stat5555sor offers a custom formula tailored to college basketball that provides a more accurate assessment of team strength. Unlike other SOR systems, this package is designed to be user-friendly and accessible to analysts of all skill levels. With clear and concise documentation, users can easily navigate the package and calculate SOR scores for a given season. Moreover, the package is highly transparent, providing users with insights into the underlying data and formula used to calculate the scores. By using stat5555sor, users can gain a deeper understanding of team performance and make informed decisions.

In this vignette, we will provide an overview of the package and demonstrate how to use it to calculate SOR scores for a given season. We will also discuss the underlying data and formula used to calculate the scores, as well as provide an example of how to use the output from this package in conjunction with the commonly used data visualization package, ggplot2, to create a visualization of the data. By the end of this vignette, users will have a better understanding of what an SOR score is, how it is calculated, and how to use the stat5555sor package to analyze college basketball team performance.

Using the sum_sor() function

In this section, we'll explore how to visualize NCAA basketball data using R. We'll be using data that includes information about every Division I basketball game played in the 2022-2023 seasons. This data is normally pulled from the two following websites, https://kenpom.com/cbbga23.txt and https://www.warrennolan.com/basketball/2023/net and then formatted into a more viewable data frame within R.

The 'sum_sor' function is the master function in this data analysis pipeline. Specifically, it calls all the necessary functions to compute the SOR score for every team. 'The sum_sor' function takes two arguments: year, which is the year for which the data is requested, and games_date, which is the cutoff date for the games played.

Upon calling the 'sum_sor' function, it first runs the 'get_combined_data' function to obtain the data frame with the necessary information about each game. Once that data frame is created and formatted correctly, two new, empty columns are created within that data frame for the SOR change for both the home and away teams.

In order to calculate the SOR score for each game, the function 'sor_points' is then called from within the 'get_combined_data' function and applied to each row twice. The 'sor_points' function takes in the team's own score, the opposing team's score, the opposing team's NET ranking, the number of teams in the college basketball season, and the game location as inputs. It returns the strength of record points gained or lost by the home or away team based on whether they won or lost the game.

At this point within the 'sum_sor' function, the data frame would look something like this based on the year and date given:

```
test <- get_combined_data(2023, "2023-03-12")
head(test)
```

```
#> # A tibble: 6 x 12
#> # Rowwise:
#>   Date       Away_team     Away_score away_NET Home_team Home_score home_NET score_diff
#>   <date>     <chr>              <int>    <int> <chr>          <int>    <int>      <int>
#> 1 2022-11-07 Jackson St.           56      305 Abilene ~         65      206          9
#> 2 2022-11-07 South Dakota~         80      155 Akron             81      102          1
#> 3 2022-11-07 Longwood              54      166 Alabama           75        2         21
#> 4 2022-11-07 Nicholls St.          75      263 Arizona          117       10         42
#> 5 2022-11-07 Tarleton St.          59      162 Arizona ~         62       66          3
#> 6 2022-11-07 North Dakota~         58      215 Arkansas          76       21         18
#> # i 4 more variables: site <chr>, neutral <dbl>, home_sor <dbl>, away_sor <dbl>
```

Finally, the sum_sor function will run two for loops to sum up the SOR scores for each team. The function then creates a new data frame that contains three columns, the name of the team, their total SOR score and their SOR ranking.

```
test_sor <- sum_sor(2023, "2023-03-12")
head(test_sor)
```

```
#>            Team      SOR ranking
#> 114     Houston 8.075916       1
#> 4       Alabama 7.806283       2
#> 133      Kansas 7.701571       3
#> 318        UCLA 7.557592       4
#> 238      Purdue 7.520942       5
#> 208 North Texas 6.767016       6
```

While this function was designed to calculate the Strength of Record for a specific team, the output can also be used to compare the performance of multiple teams over time or to identify trends in team performance. This can also be a useful tool for coaches, analysts, or fans looking to evaluate the success of their team or identify areas for improvement.

Additional Uses for Package Output

Visualization of Utah Teams' SOR Scores Over the Last 3 Seasons

This section shows how to create a visualization of the Strength of Record (SOR) scores for all Utah college basketball teams over the past three seasons. The purpose of this visualization is to give users a better understanding of how the SOR scores of these teams have changed over time, and to identify any trends or patterns that may be present.

By running the sum_sor function once for every season, the user could then bind the data all together and add the year to each row to create a dataset that contains the SOR score for each team for the past 3 years.Note that this dataset is included within the package itself and can be called using the 'data' function once the package has been installed to the user's device.

```
## Getting data for each year
year_2023 <- sum_sor(2023, "2023-03-12")
year_2022 <- sum_sor(2022, "2022-03-13")
year_2021 <- sum_sor(2021, "2021-03-14")
```

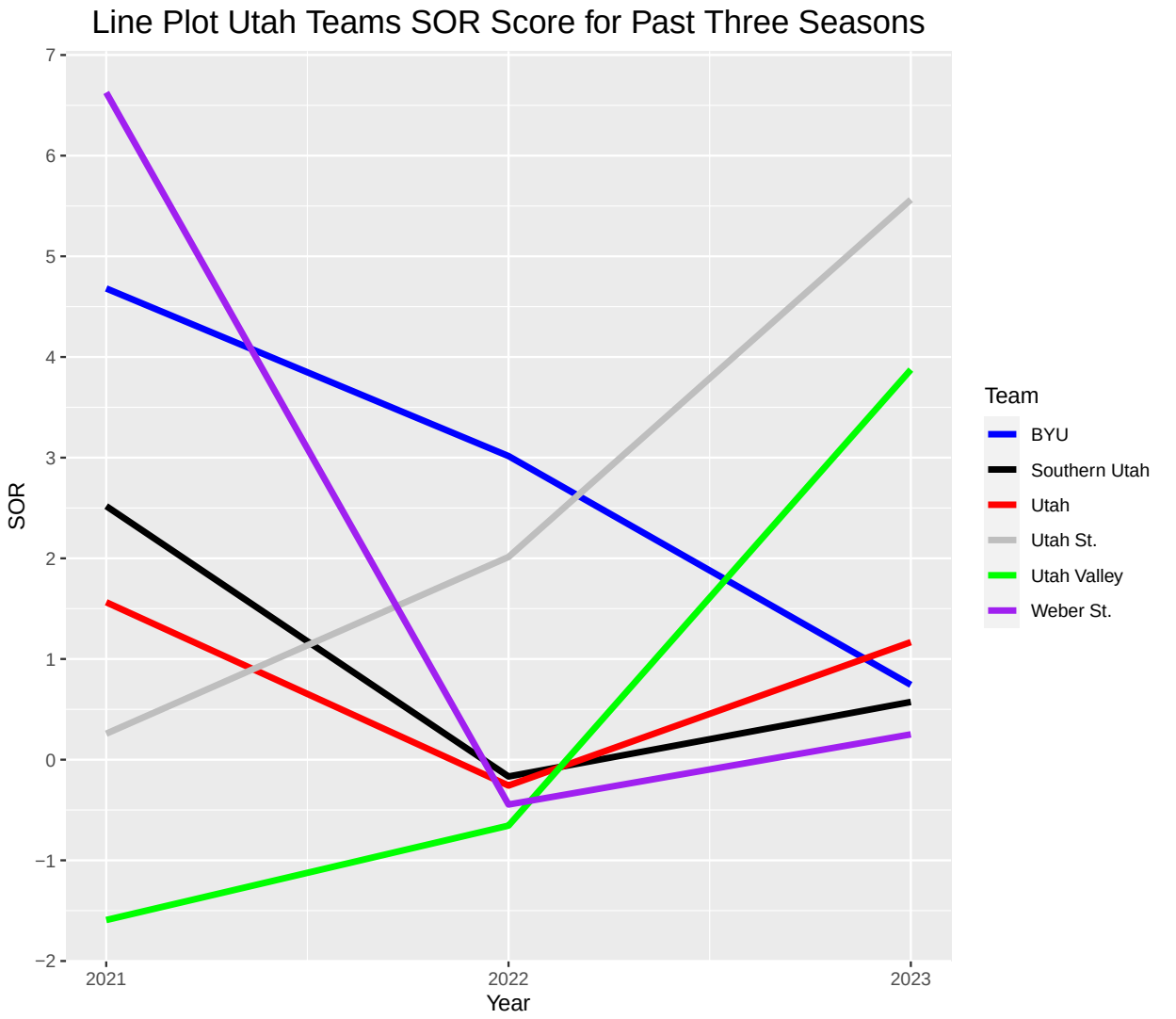
```r
## Adding year to data before combining
year_2023$year <- rep(2023, nrow(year_2023))
year_2022$year <- rep(2022, nrow(year_2022))
year_2021$year <- rep(2021, nrow(year_2021))

## binding data
combined_years <- rbind(year_2023, year_2022, year_2021)
```

Further subsetting the data to only include any teams that contain the phrases "Utah", "BYU" or "Weber St" will produce a dataset that only contains the Utah team's SOR scores over the last 3 seasons.

```r
  library(stat5555sor)
  # Load dataset from within package
  data("combined_years")
  # filter dataset to only include rows where the state column contains "Utah"
  df_utah <- combined_years[grepl("Utah|BYU|Weber St", combined_years$Team), ]
  # exclude Utah Tech
  df_utah <- df_utah[!grepl("Utah Tech", df_utah$Team), ]
  head(df_utah)
#>              Team       SOR ranking year
#> 331      Utah St. 5.5628272       8 2023
#> 333   Utah Valley 3.8743455      32 2023
#> 330          Utah 1.1675393      99 2023
#> 35            BYU 0.7434555     120 2023
#> 275 Southern Utah 0.5732984     126 2023
#> 348     Weber St. 0.2513089     138 2023
```

From here, producing a line plot to see the change over time could be done using the ggplot2 library.

```r
library(ggplot2)
  ggplot(data = df_utah, aes(x = year, y = SOR, group = Team, color = Team)) +
    geom_line(size = 1.5) +
    scale_x_continuous(breaks = seq(min(df_utah$year), max(df_utah$year),
                                    by = 1)) +
    scale_y_continuous(breaks = seq(round(min(df_utah$SOR)),
                                    round(max(df_utah$SOR))),
                       labels = function(x) format(x, scientific = FALSE)) +
    scale_color_manual(values = c("blue", "black", "red", "grey",
                                  "green", "purple")) +
    labs(title = "Line Plot Utah Teams SOR Score for Past Three Seasons",
         x = "Year", y = "SOR") +
    theme(plot.title = element_text(size = 16, hjust = 0.5))
```



In the graph, we notice an uptick in performance ever since the hiring of Ryan Odom at Utah State as a replacement for Craig Smith who left to join Utah. This suggests that coaching changes can have a significant impact on a team's SOR over time. Interestingly, Ryan Odom recently left Utah State in 2023 to join VCU. If we were to perform this same analysis 2-3 years from then, it would be interesting to see how the SOR of Utah State evolves under a new coach.

Conclusion

In this vignette, we have presented a package that can be used to calculate the Strength of Record (SOR) scores for college basketball teams. The main function of the package, sum_sor, allows users to easily obtain the SOR scores for any given season and date.

Additionally, we have demonstrated several potential uses for the SOR scores, including comparing team performance across seasons, identifying top-performing teams, and analyzing the performance of local Utah teams.

We have also included a custom function, sor_points, which can be used to calculate the SOR points for individual games. This function may be of particular interest to those who want to dive deeper into the SOR methodology or who want to create their own SOR-related analyses.

Overall, this package provides a useful tool for anyone interested in analyzing college basketball team performance. With its simple and intuitive interface, users can quickly and easily obtain SOR scores for any season and date. We hope that this package will be of use to researchers, analysts, and basketball enthusiasts alike.

Link to Package

https://github.com/Gideonparry/stat5555sor