

Package Vignette

```
library(stat555sor)
```

Workflow

Web scrape data and format

Everything starts with the `sum_sor` function, which will in turn call all the rest of the helper functions. It is not the only function exported though as some users may want to see the changes in data in between steps.

To start off, the data is 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. This function was exported as there is a parameter of date that the function will include all games up until that date. So, a user could find the sor score for half a season if they wanted. In this example below, we chose the date of selection sunday for this year.

```
test <- get_combined_data(2023, "2023-03-12")
#> No encoding supplied: defaulting to UTF-8.
head(test)
#> # A tibble: 6 x 12
#> # Rowwise:
#>   Date       Away_team      Away_score away_NET Home_team      Home_score home_NET score_diff site
#>   <date>      <chr>          <int>    <int> <chr>          <int>    <int>    <int> <chr>
#> 1 2022-11-07 Jackson St.         56      305 Abilene Chris~        65      206         9 Abil
#> 2 2022-11-07 South Dakota St.      80      155 Akron              81      102         1 Akro
#> 3 2022-11-07 Longwood             54      166 Alabama             75         2      21 Alab
#> 4 2022-11-07 Nicholls St.         75      263 Arizona            117        10      42 Ariz
#> 5 2022-11-07 Tarleton St.         59      162 Arizona St.         62        66         3 Ariz
#> 6 2022-11-07 North Dakota St.      58      215 Arkansas            76        21      18 Arka
```

sor_points function

This is the function that contains the algorithm we created to find the change in sor score for one game. It was exported as well so that users have the ability to see the sor change for any specific game. As an example we will take the data from the first row of the test dataset and run the function to see the change in sor for Jackson State.

```
sor_points(56, 65, 206, 363, FALSE, FALSE)
#> [1] -0.5366492
```

sum_sor results

Finally, the `sum_sor` function will sum up the sor changes for each row and each team. It will then create a new data frame with two columns, the team name and the total sor score.

```
test_sor <- sum_sor(2023, "2023-03-12")
#> No encoding supplied: defaulting to UTF-8.
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
```

Additional Package Uses

If I wanted to look at the past seasons and make a visualization for their sor scores over that time period, here is how I could do it using this package (note that this is an example dataset that is included within the package already):

```
## 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")

## 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)

usethis::use_data(combined_years, overwrite = TRUE)
```

This gives me a dataset with every team's total sor score over the past three seasons. I could now filter it to only contain Utah teams by running this code:

```
library(stat555sor)
# 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
```

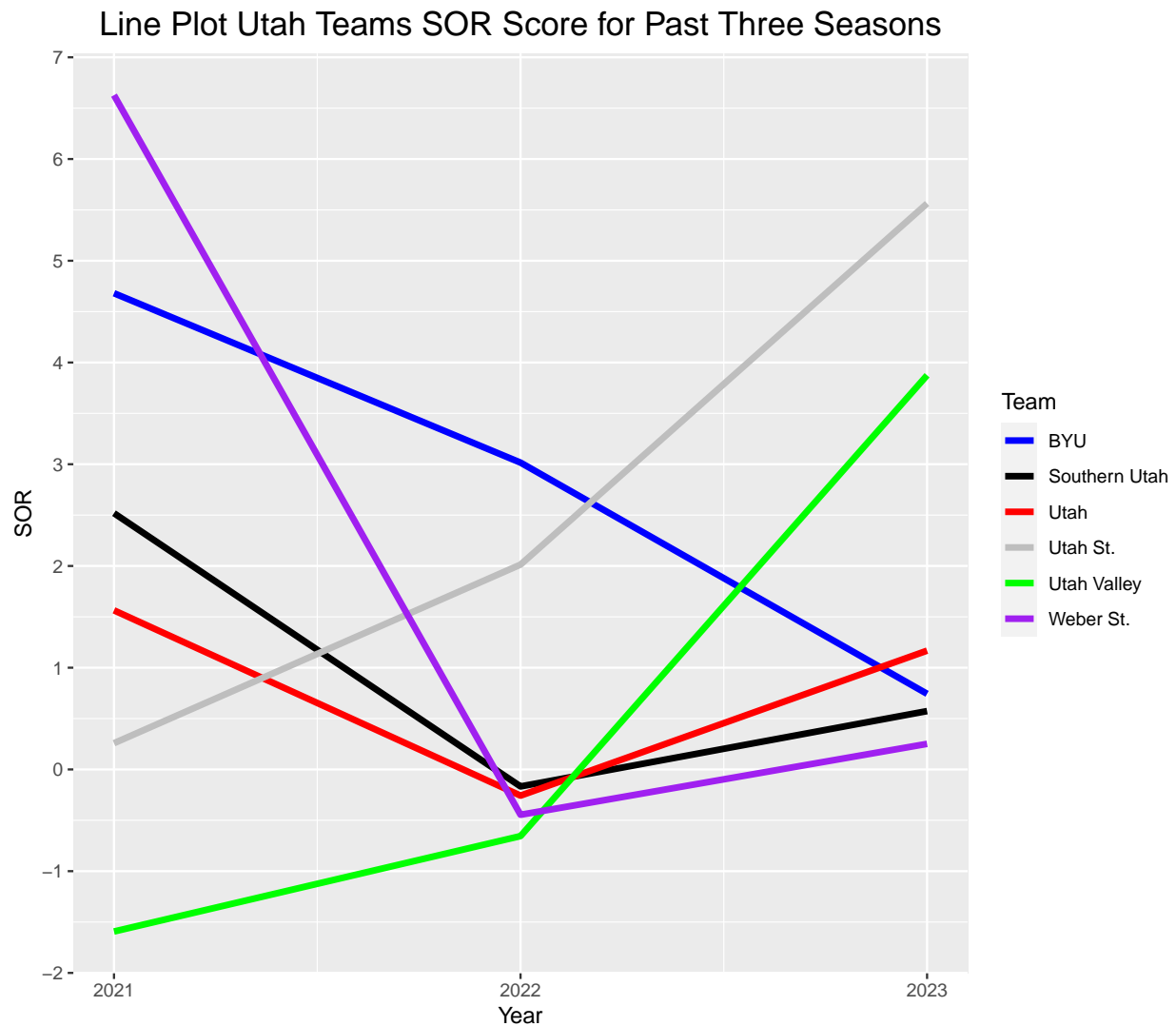
From here, to create a line plot to be able to better visualize the change in total sor over the past three seasons, I could run this code:

```
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))

```



Link to Package

<https://github.com/Gideonparry/stat555sor>