Welcome to {midsprint}

Aaron Pearson, Dani Chu, and Patrick Ward

2021-02-15

The midsprint Package

This package provides coaches with a low barrier of entry into athlete profiling and positional tracking data. This file provides you with a simple walkthrough that allows you to model athlete:

- Speed-Acceleration
- Speed and acceleration over time from game data
- Speed and acceleration over time from Combine data

Getting Started

This section walks you through installing midsprint, loading it into R/RStudio, and preparing your data for analysis.

Installing and Loading midsprint

To install midsprint, you'll need to copy and paste the following code into your 'Console'.

```
devtools::install_github("aaronzpearson/midsprint")
```

Loading in the package allows you to make use of the functions that are kept within. Without loading them, they are much more difficult to reach and utilize. Loading midsprint into your current session is simply:

```
library(midsprint)
```

Loading Data

The package comes with sample GPS and Combine data. If you are unsure of how to clean and analyze your current tracking data, I suggest following this walkthrough. You can always call upon the sample data to try things out before working on your own data set. The sample data will always be at your disposal and you cannot affect it permanently.

```
# To load the sample data:
data(player_a)
data(player_b)
data(player_40yd)
```

To see what they look like, run the following line:

```
head(player_a)
#>   speed accel
```

```
#> 1  0.02  0.03
#> 2  0.03  0.03
#> 3  0.02  0.03
#> 4  0.02  0.02
#> 5  0.02  0.02
#> 6  0.01  0.02
```

```
head(player_40yd)
#>   Distance Split
#> 1        0  0.00
#> 2       10  1.50
#> 3       20  2.59
#> 4       40  4.47
```

When loading your own data, it will look something like this:

```
player_a <- read_csv("~/folder/player_a.csv")
player_a_combine <- read_csv("~/folder/player_a_combine.csv")
```

The function `read_csv()` is part of the readr package. It is part of the Tidyverse, and it is suggested that inexperienced R users download the Tidyverse. It is intuitive and easy to learn.

Working with Game Data

midsprint does the heavy lifting for you. Once your data is loaded, run `game_data()` to clean the data and keep what is important.

```
player_a_data <- game_data(player_a$speed, player_a$accel)
player_b_data <- game_data(player_b$speed, player_b$accel)
```

Once the data is loaded, there are two methods you can progress. Simple modeling doesn't require further processing.

Speed-Acceleration Modeling

The following functions are modeled based on JB Morin's in-situ speed-acceleration modeling. Please allow 2-3 seconds for these functions to run. There is a lot that goes on behind-the-scenes. Larger data sets take longer to run.

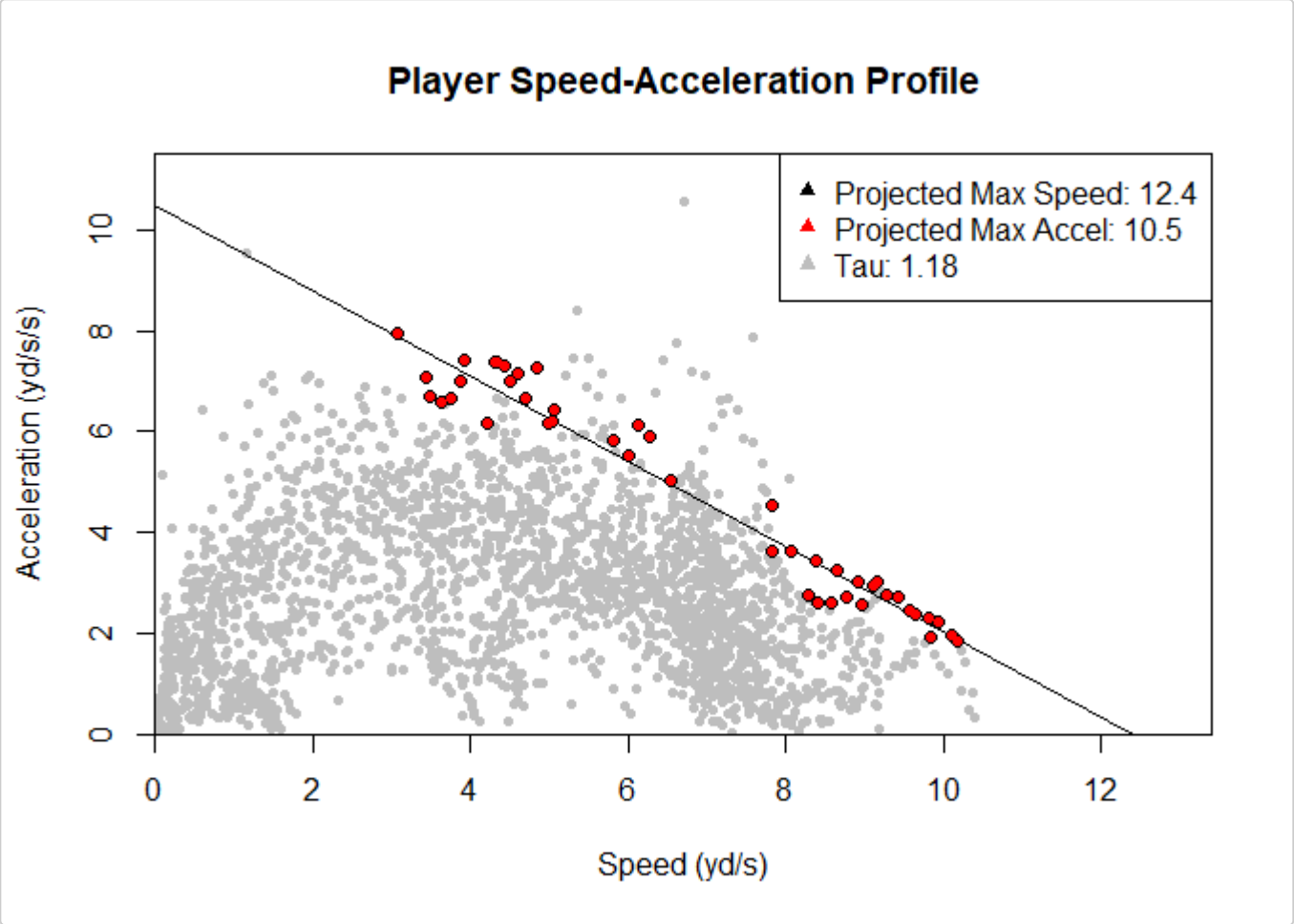
```
player_a_speed_accel <- speed_accel(player_a_data)
```

The output should look something like this:

```
player_a_speed_accel
#>   theoretical_max_accel theoretical_max_speed r_square n_obervation
#> 1                  10.5                  12.4     0.95           45
```

For coaches who are more visual, you can run the plotting function below.

```
player_a_speed_accel_plot <- speed_accel_plot(player_a_data)
```



In-Depth Analyses

To further your analyses, you'll need to create a player profile. These profiles are unique for each player and can tell you a lot about their sprint abilities on the field.

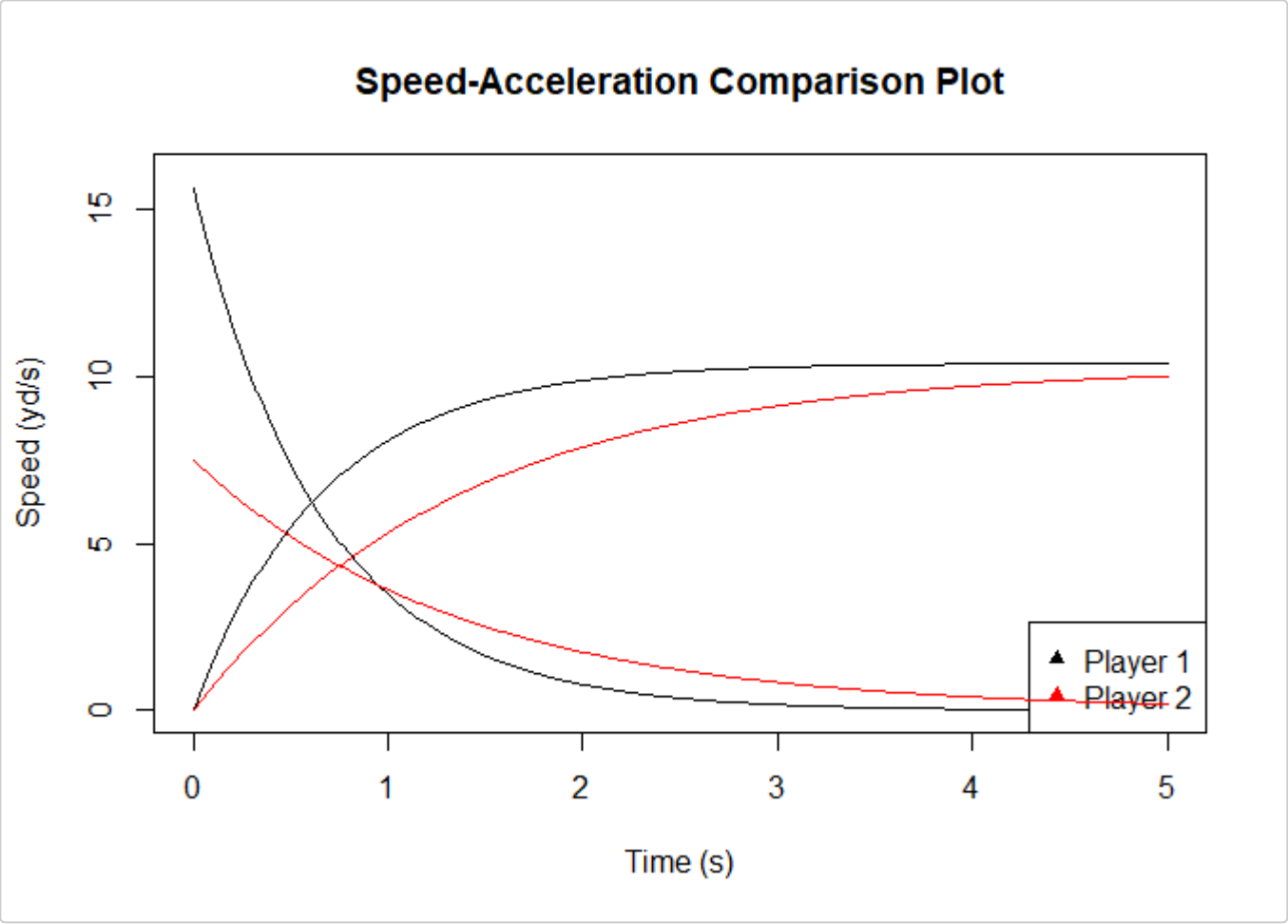
```
player_a_profile <- game_profile(player_a_data)
player_b_profile <- game_profile(player_b_data)
```

Resulting in:

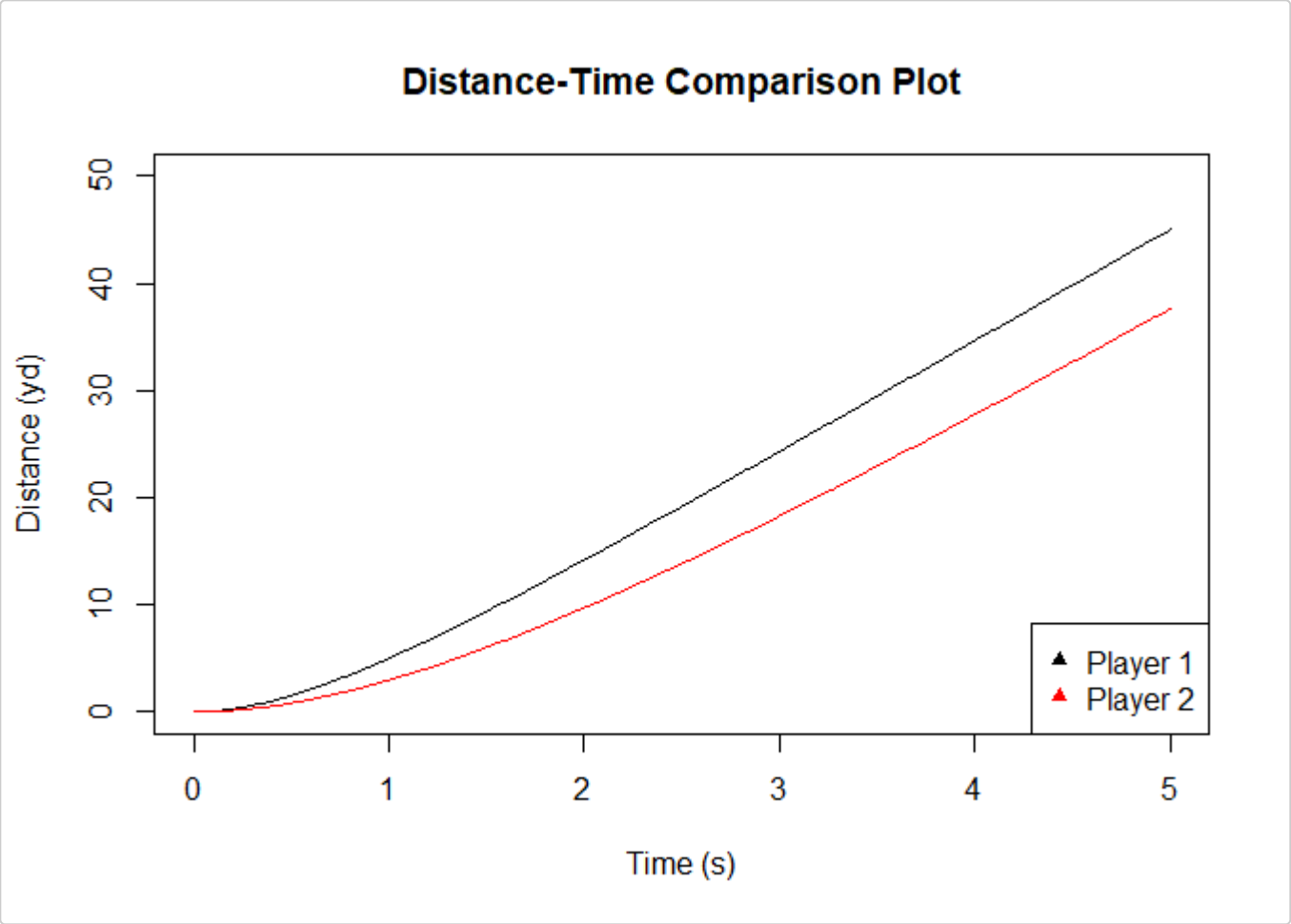
```
player_a_profile
#>   Max.Observed.Player.Speed Max.Observed.Player.Acceleration Player.Tau.Value
#> 1                     10.39                             15.6        0.6660256
```

You can then plot athletes separately or compare two on the same plot. Below are some examples that you can run.

```
compare_player_profile_plot(player_a_profile, player_b_profile)
```




```
compare_player_distance_time_plot(player_a_profile, player_b_profile)
```



Working with Combine Data

Combine data are data that include time splits and distances traveled. Algorithms are then used to model player speed over time. FIrstly, you'll need to load Combine data. You did this in the first section and currently have `player_40yd` ready to go. The data should look something like this:

```
head(player_40yd)
#>   Distance Split
#> 1        0  0.00
#> 2       10  1.50
#> 3       20  2.59
#> 4       40  4.47
```

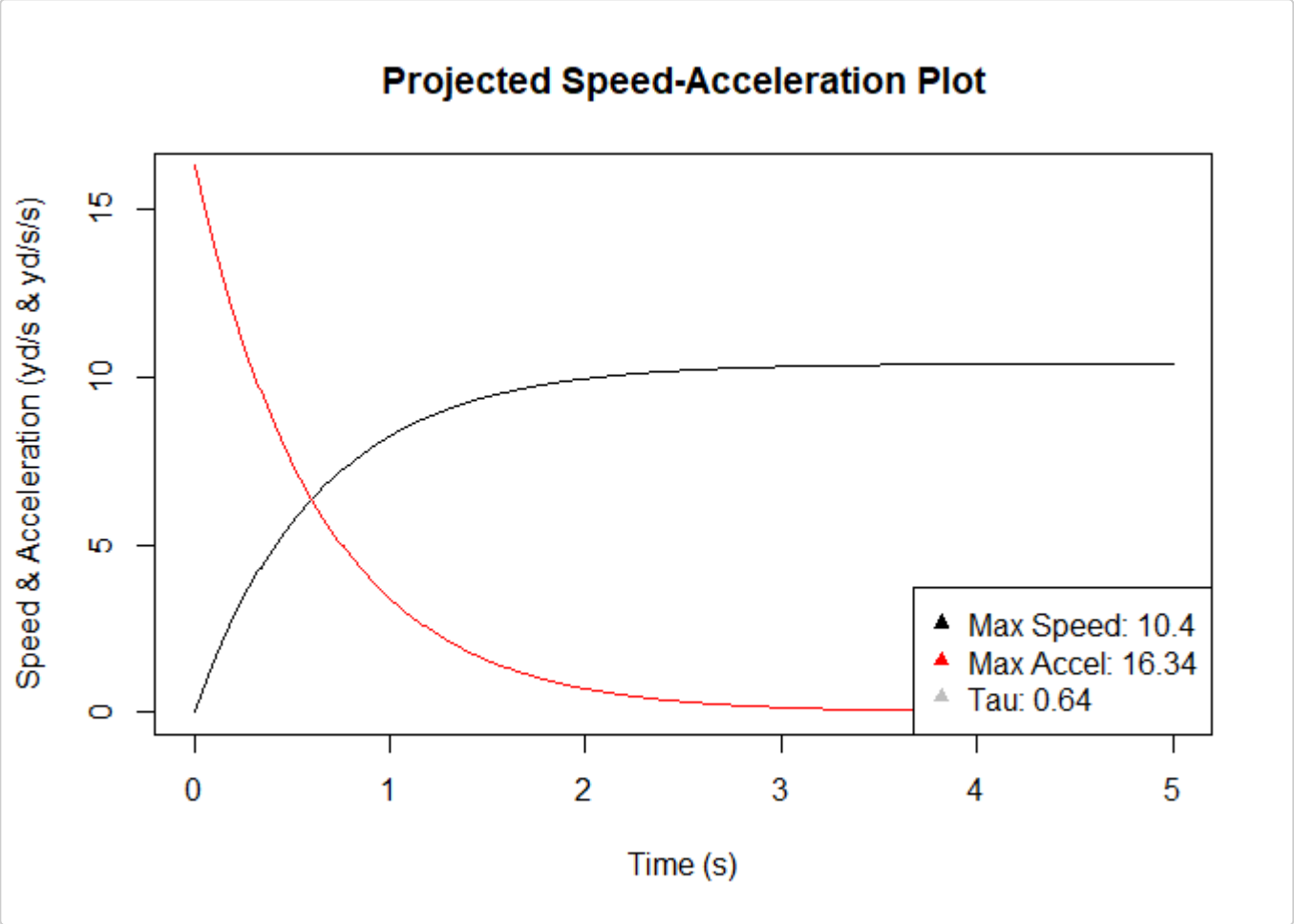
You will then need to create a player profile.

```
optim_player_a_profile <- optim_player_profile(player_40yd$Distance, player_40yd$Split)

optim_player_a_profile
#>   Max.Modeled.Player.Speed Max.Modeled.Player.Acceleration
#> 1                 10.40031                        16.33889
#>   Modeled.Player.Tau.Value
#> 1                0.6365367
```

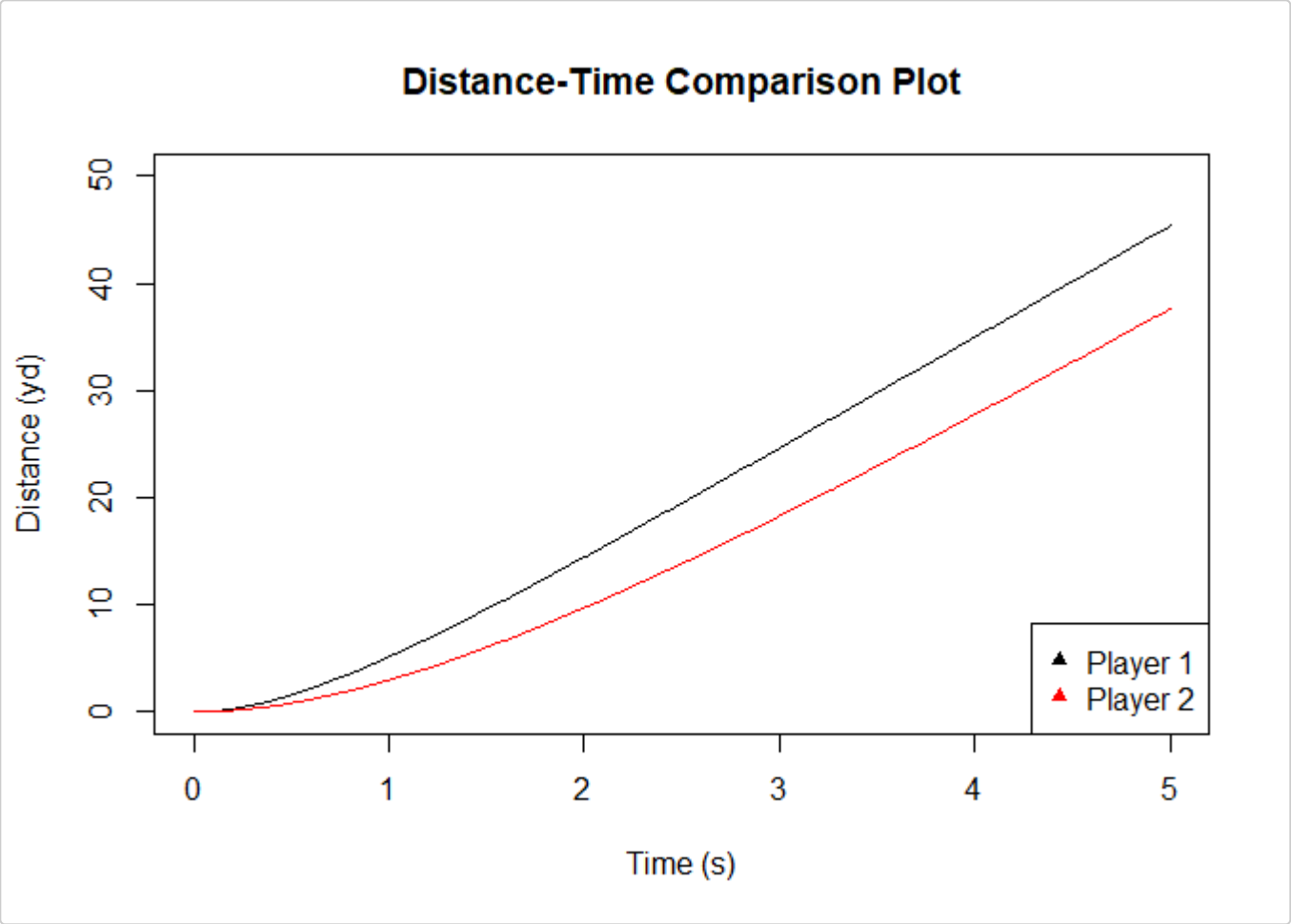
Once the data is loaded, you can perform the same analyses.

```
optim_player_plot(optim_player_a_profile)
```



With two profiles, you can compare player abilities like you did above. This does not need to be data originating from the same source.

```
compare_player_distance_time_plot(optim_player_a_profile, player_b_profile)
```



Other Functions

Below are some other functions included in the package. These functions return player values that correspond with the graphs you produced in the previous sections.

When starting from zero velocity, this function calculates the separation distance after traveling a given distance:

```
separation_distance(player_a_profile, player_b_profile, 40)
#> [1] "Player 1 is ahead of Player 2 by 7.2 yards."
```

Similarly, this function calculates separation distance while players have any positive velocity. Also, you can input different distances each player must travel. This allows practitioners to better understand player strengths and improve match-up efficacy.

```
separation_in_motion(player_a_profile, 6, 17,
                     player_b_profile, 2.5, 22)
#> [1] "Player A trails Player B by 3.8 yards."
```

To find a player's final velocity:

```
final_speed(player_a_profile, 6, 28)
#> [1] 10.34
```

Citation

When publishing work utilizing the midsprint package, please cite the package using the function `cite_midsprint()`.

```
cite_midsprint()
#>
#> To cite package 'midsprint' in publications use:
#>
#>   "Aaron Pearson, Dani Chu and Patrick Ward" (2021). midsprint:
#>   midsprint. R package version 0.1.1.
#>
#> A BibTeX entry for LaTeX users is
#>
#>   @Manual{,
#>     title = {midsprint: midsprint},
#>     author = {"Aaron Pearson and Dani Chu and Patrick Ward"},
#>     year = {2021},
#>     note = {R package version 0.1.1},
#>   }
#>
#> ATTENTION: This citation information has been auto-generated from the
#> package DESCRIPTION file and may need manual editing, see
#> 'help("citation")'.
```