

Package: ovscout2 (via r-universe)

August 16, 2024

Title An R Shiny App for Scouting Volleyball Match Files

Version 0.2.9

Description Functions for scouting volleyball match files.

URL <https://github.com/openvolley/ovscout2>

BugReports <https://github.com/openvolley/ovscout2/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Depends datavolley (> 0.12.999)

Imports assertthat, curl, data.table, dplyr, digest, DT, fs, ggplot2, jpeg, htmltools, httpuv, httr, lubridate, methods, ovddata, ovideo (>= 0.17.2), peranavolley, qrcode, R6, R.utils, rappdirs, rintrojs, rmarkdown, servr (>= 0.15.1), shiny, shinythemes, shinyWidgets, stringr, sys, tidyr, uuid, volleyreport (>= 0.4.4)

Suggests covr, DBI, dbplyr, duckdb, knitr, rstudioapi, tcltk, testthat

Remotes openvolley/datavolley, openvolley/ovdata, openvolley/ovideo, openvolley/peranavolley, openvolley/volleyreport

RoxygenNote 7.2.3

VignetteBuilder knitr

Repository <https://openvolley.r-universe.dev>

RemoteUrl <https://github.com/openvolley/ovscout2>

RemoteRef HEAD

RemoteSha cd1406b3c8435b3edb125322a2f1ab485c6aa053

Contents

dv_create	2
dv_scout_from_console	3
dv_set_lineups	4
ovscout2	5
ov_app_styling	5
ov_default_attack_table	5
ov_default_compound_table	6
ov_default_scouting_table	6
ov_default_setter_calls_table	6
ov_default_shortcuts	7
ov_default_winning_symbols	7
ov_install_lighttpd	7
ov_scouter	8
ov_scouting_options	10

Index	12
--------------	-----------

dv_create	<i>Create a new datavolley match object</i>
-----------	---

Description

Create a new datavolley match object

Usage

```
dv_create(
  match,
  more,
  teams,
  players_h,
  players_v,
  video_file,
  attacks = ov_simplified_attack_table(),
  setter_calls = ov_default_setter_calls_table(),
  winning_symbols = ov_default_winning_symbols(),
  zones_or_cones = "Z",
  regulation = "indoor rally point",
  comments
)
```

Arguments

match	list or single-row data.frame: (optional) with components date (defaults to current date), time (defaults to current time), season, league, phase, home_away, day_number, match_number, regulation, zones_or_cones. zones_or_cones can also be provided directly
-------	--

more	list or single-row data.frame: (optional) with components referees, spectators, receipts, city, arena, scout
teams	data.frame: a 2-row data frame describing the home and visiting teams, with required columns team_id, team and optional columns coach, assistant, shirt_colour. The home team must be in the first row of this data frame
players_h, players_v	data.frame: with required columns number, firstname, lastname, and optional columns player_id, role (character vector with "outside", "opposite", "middle", "libero", "setter"), nickname, special_role (character vector with "L", "C", or NA), foreign (logical, defaults to FALSE)
video_file	string: (optional) path to video file
attacks	data.frame: as returned by <code>ov_simplified_attack_table()</code> or <code>ov_default_attack_table()</code>
setter_calls	data.frame: as returned by <code>ov_default_setter_calls_table()</code>
winning_symbols	data.frame: as returned by <code>ov_default_winning_symbols()</code>
zones_or_cones	string: "Z" or "C". Will be ignored if zones_or_cones is provided in the match parameter
regulation	string: "indoor rally point", "beach rally point", or "indoor sideout". Will be ignored if regulation is provided in the match parameter
comments	character: optional vector of length up to 5, of comments

Value

A datavolley object

Examples

```
x <- dv_create(teams = data.frame(team_id = c("TM1", "TM2"), team = c("Team 1", "Team 2")),
  comments = "Test file",
  players_h = data.frame(firstname = toupper(letters[1:7]), lastname = "Player",
    number = 1:7),
  players_v = data.frame(firstname = letters[10:15], lastname = "VisPlayer",
    number = 10:15))

## enter the team lineups for set 1
x <- dv_set_lineups(x, set_number = 1, lineups = list(6:1, 15:10), setter_positions = c(2, 1))
```

dv_scout_from_console *Enter scout codes from the console*

Description

Probably only useful for testing.

Usage

```
dv_scout_from_console(
  x,
  prompt = "SCOUT> ",
  compound_table = ov_default_compound_table(),
  default_scouting_table = ov_default_scouting_table()
)
```

Arguments

`x` datavolley: a datavolley object as returned by `dv_create()`

`prompt` string: the prompt to show

`compound_table` tibble: the table of default compound codes

`default_scouting_table` tibble: the table of scouting defaults (skill type and evaluation)

Value

A modified version of `x`, with rows added to the `plays2` component

`dv_set_lineups` *Enter the team lineups at the start of a set*

Description

Enter the team lineups at the start of a set

Usage

```
dv_set_lineups(x, set_number, lineups, setter_positions, setters)
```

Arguments

`x` datavolley: a datavolley object

`set_number` integer: set number, 1–3 for beach or 1–5 for indoor

`lineups` list: two-element list with numeric vectors of player numbers. Each lineup is

- for indoor, of length 6, 7, or 8 (first 6 are player jersey numbers in positions 1–6, elements 7 and 8 are optionally the libero jersey numbers)
- for beach, of length 2

`setter_positions` integer: two-element integer vector giving the position on court of the two setters. At least one of `setter_positions` or `setters` must be provided for indoor. Ignored for beach

`setters` integer: two-element integer vector giving the jersey numbers of the two setters. At least one of `setter_positions` or `setters` must be provided for indoor. Ignored for beach

Value

A modified version of x

ovscout2	ovscout2
----------	-----------------

Description

Functions for scouting volleyball match files.

ov_app_styling	<i>Styling to apply to the app</i>
----------------	------------------------------------

Description

Colours can be either hex strings or names, but if using names they must be recognized both by R and as CSS colour names.

Usage

ov_app_styling()

Details

- review_pane_width is expressed as a percentage of the browser window width

Value

A named list of styling parameters

ov_default_attack_table	<i>Default attack combination code table</i>
-------------------------	--

Description

Default attack combination code table

Usage

ov_default_attack_table()

ov_simplified_attack_table()

Value

A tibble

ov_default_compound_table

Default compound skills table

Description

Default compound skills table

Usage

ov_default_compound_table()

Value

A tibble

ov_default_scouting_table

Default scouting (type and evaluation for each skill) table

Description

Default scouting (type and evaluation for each skill) table

Usage

ov_default_scouting_table()

Value

A tibble

ov_default_setter_calls_table

Default setter calls table

Description

Default setter calls table

Usage

ov_default_setter_calls_table()

Value

A tibble

ov_default_shortcuts *Default keyboard shortcuts for ov_scouter*

Description

Default keyboard shortcuts for ov_scouter

Usage

ov_default_shortcuts()

Value

A named list

ov_default_winning_symbols
Default winning symbols table

Description

Default winning symbols table

Usage

ov_default_winning_symbols()

Value

A tibble

ov_install_lighttpd *Install lighttpd*

Description

This is a helper function to install lighttpd. Currently it only works on Windows platforms. The lighttpd bundle will be downloaded from <http://lighttpd.dtech.hu/> and saved to your user appdata directory.

Usage

ov_install_lighttpd(force = FALSE)

Arguments

force logical: force reinstallation if lighttpd already exists

Value

the path to the installed executable

References

<http://lighttpd.dtech.hu/>

Examples

```
## Not run:  
  ov_install_lighttpd()  
  
## End(Not run)
```

ov_scouter

Launch a Shiny app for scouting

Description

Launch a Shiny app for scouting

Usage

```
ov_scouter(  
  dvw,  
  video_file,  
  court_ref,  
  season_dir,  
  auto_save_dir,  
  scoreboard = TRUE,  
  ball_path = FALSE,  
  playlist_display_option = "dv_codes",  
  review_pane = TRUE,  
  playback_rate = 1,  
  scouting_options = ov_scouting_options(),  
  app_styling = ov_app_styling(),  
  shortcuts = ov_default_shortcuts(),  
  scout_name = "",  
  show_courtref = FALSE,  
  dash = FALSE,  
  host,  
  launch_browser = TRUE,
```



```

    prompt_for_files = interactive(),
    ...
)

```

Arguments

dvw	string or datavolley: either the path to a dvw or ovs file or a datavolley object (e.g. as returned by <code>dv_create()</code>). Passing the file name (not the datavolley object) is required if any extra arguments are passed via <code>...</code> . dvw can also be an object as saved by <code>ov_scouter()</code> in ovs format. If dvw is "demo", the app will be started with a demonstration data set
video_file	string: optionally, the path to the video file. If not supplied (or NULL) the video file specified in the dvw file will be used. <code>video_file</code> can also be a URL (including a YouTube URL or video ID)
court_ref	data.frame or string: data.frame with the court reference (as returned by <code>ovideo::ov_shiny_court_ref()</code>) or the path to the rds file containing the output from this
season_dir	string: optional path to a directory with other dvw/ovs files from this season
auto_save_dir	string: optional path to a directory where the dvw will be saved automatically after each rally
scoreboard	logical: if TRUE, show a scoreboard in the top-right of the video pane
ball_path	logical: if TRUE, show the ball path on the court inset diagram. Note that this will slow the app down slightly
playlist_display_option	string: what to show in the plays table? Either "dv_codes" (scouted codes) or "commentary" (a plain-language interpretation of the touches)
review_pane	logical: if TRUE, entry popups will be accompanied by a small video pane that shows a loop of the video of the action in question
playback_rate	numeric: starting playback rate of the video (1.0 is normal speed, higher is faster)
scouting_options	list: a named list with entries as per <code>ov_scouting_options()</code> . See Details, below
app_styling	list: named list of styling options, as returned by <code>ov_app_styling()</code>
shortcuts	list: named list of keyboard shortcuts, as returned by <code>ov_default_shortcuts()</code>
scout_name	string: the name of the scout (your name)
show_courtref	logical: if TRUE, show the court reference lines overlaid on the video
dash	logical: support live MPEG DASH streams? If not specified, will default to TRUE if <code>video_file</code> is a *.mpd stream. Note that DASH support is fragile at best. HLS streams are automatically supported and likely to be more reliable
host	string: the IP address of this machine. Only required if you intend to connect to the app from a different machine (in which case use <code>ov_scouter(..., host = "www.xxx.yyy.zzz", launch_browser = FALSE)</code> , where <code>www.xxx.yyy.zzz</code> is the IP address of this machine, i.e. the machine running the app)

launch_browser logical: if TRUE, launch the app in the system's default web browser (passed to `shiny::runApp()`'s `launch.browser` parameter)
 prompt_for_files logical: if `dvw` was not specified, prompt the user to select the `dvw` file
 ... : extra parameters passed to `datavolley::dv_read()` (if `dvw` is a provided as a string) and/or to the shiny server and UI functions

Details

A note on `scouting_options`

If a `*.ovs` file (i.e. a partially-scouted file, that was previously scouted using this app) has been provided in the `dvw` argument, then it will contain the scouting options used during the previous scouting session. Those options will be re-used EXCEPT if `scouting_options` are also provided here. Any scouting options provided here via the `scouting_options` argument will override options saved in the `.ovs` file. Thus, it is recommended that `scouting_options` not be provided here along with a `.ovs` file unless absolutely necessary. If necessary, only the specific, relevant elements of the `scouting_options` list should be provided. Note that `*.dvw` files do not contain saved options, only `.ovs` files that were scouted with this app.

Examples

```
## Not run:
  ov_scouter("demo")

## End(Not run)
```

ov_scouting_options *Scouting options*

Description

Scouting options

Usage

```
ov_scouting_options(
  end_convention = "actual",
  nblockers = TRUE,
  default_nblockers = NA,
  transition_sets = FALSE,
  attacks_by = "codes",
  team_system = "SHM3",
  setter_dump_code = "PP",
  second_ball_attack_code = "P2",
  overpass_attack_code = "PR",
  default_scouting_table = ov_default_scouting_table(),
```

```

    compound_table = ov_default_compound_table(),
    attack_table = ov_simplified_attack_table()
)

```

Arguments

end_convention string: either "actual" or "intended". Is the end coordinate of an attack or serve the actual end location (where the ball contacted the floor or out of bounds area), or the intended one. The actual might differ from the intended if there is a block touch or the ball hit the net. If "actual", and a block touch is recorded, then the end location of the attack will not be used for the dig location (the dig location will be missing)

nblockers logical: scout the number of blockers on each attack?

default_nblockers integer: if nblockers is TRUE, what number of blockers should we default to? If NA, no default

transition_sets logical: scout sets in transition? If FALSE, just the endpoint of each attack (i.e. the dig) and the subsequent counter-attack are scouted

attacks_by string: "codes" (X5, V5, etc) or "tempo" (high, medium, quick)

team_system string: the assumed system that teams are using to assign e.g. passing and hitting responsibilities

- "SHM3" - a setter-hitter-middle rotation, with 3 passers (the libero and two outside hitters)

setter_dump_code string: the attack combination code for a setter dump

second_ball_attack_code string: the attack combination code for a second-ball attack

overpass_attack_code string: the attack combination code for an attack on an overpass

default_scouting_table tibble: the table of scouting defaults (skill type and evaluation)

compound_table tibble: the table of compound codes

attack_table tibble: table of attack codes (X5, V5, etc) as returned by `ov_default_attack_table()` or `ov_simplified_attack_table()`

Value

A named list

Index

`datavolley::dv_read()`, [10](#)
`dv_create`, [2](#)
`dv_create()`, [4](#), [9](#)
`dv_scout_from_console`, [3](#)
`dv_set_lineups`, [4](#)

`ov_app_styling`, [5](#)
`ov_app_styling()`, [9](#)
`ov_default_attack_table`, [5](#)
`ov_default_attack_table()`, [3](#), [11](#)
`ov_default_compound_table`, [6](#)
`ov_default_scouting_table`, [6](#)
`ov_default_setter_calls_table`, [6](#)
`ov_default_setter_calls_table()`, [3](#)
`ov_default_shortcuts`, [7](#)
`ov_default_shortcuts()`, [9](#)
`ov_default_winning_symbols`, [7](#)
`ov_default_winning_symbols()`, [3](#)
`ov_install_lighttpd`, [7](#)
`ov_scouter`, [8](#)
`ov_scouting_options`, [10](#)
`ov_scouting_options()`, [9](#)
`ov_simplified_attack_table`
 (`ov_default_attack_table`), [5](#)
`ov_simplified_attack_table()`, [3](#), [11](#)
`ovideo::ov_shiny_court_ref()`, [9](#)
`ovscout2`, [5](#)

`shiny::runApp()`, [10](#)