

# Package: volleyreport (via r-universe)

September 11, 2024

**Title** Reports from Volleyball Match Files

**Version** 0.7.19

**Description** Functions for generating reports from volleyball match files.

**URL** <https://volleyreport.openvolley.org>,  
<https://github.com/openvolley/volleyreport>

**BugReports** <https://github.com/openvolley/volleyreport/issues>

**Depends** R (>= 3.6.0)

**Imports** assertthat, cowplot, datavolley (>= 1.5.4), dplyr (>= 1.0.0), fontawesome, ggplot2, ggtext, kableExtra, knitr, methods, ovlytics, ovpaged (>= 0.0.6), pdftools, rmarkdown, showtext, sysfonts, stringr, tidyr, tidyselect, vsoututils, webshot (>= 0.5.2)

**Suggests** shiny, testthat

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Remotes** openvolley/datavolley, openvolley/ovlytics,  
openvolley/ovpaged, openvolley/vsoututils

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

**RemoteUrl** <https://github.com/openvolley/volleyreport>

**RemoteRef** HEAD

**RemoteSha** 83bcf36ef2e248c959b18dbbc6ea0e2ac6201077

## Contents

volleyreport . . . . .	2
vr_attack . . . . .	3
vr_block . . . . .	3
vr_court_plots . . . . .	4
vr_match_summary . . . . .	4
vr_plot_icons . . . . .	7
vr_points . . . . .	8
vr_reception . . . . .	8
vr_serve . . . . .	9
vr_vote . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

volleyreport	<b>volleyreport</b>
--------------	---------------------

---

### Description

Reporting of volleyball statistics

### Author(s)

**Maintainer:** Adrien Ickowicz <ickowicz.adrien@gmail.com>

Authors:

- Ben Raymond
- Tyler Widdison

Other contributors:

- Francesco Oleni [contributor]
- Willem de Wit [contributor]
- Marco Chiapello [contributor]
- Felipe Aparecido Lima [contributor]
- Mark Lebedew [contributor]
- openvolley.org [originator]

### See Also

Useful links:

- <https://volleyreport.openvolley.org>
- <https://github.com/openvolley/volleyreport>
- Report bugs at <https://github.com/openvolley/volleyreport/issues>

---

vr_attack	<i>Generate attack table</i>
-----------	------------------------------

---

**Description**

Generate attack table

**Usage**

```
vr_attack(x, team, by = "player", file_type = "indoor", style = "default")
```

**Arguments**

x	datavolleyplays: the plays component of an object as returned by <code>datavolley::dv_read</code>
team	string: team name
by	string: "player" or "set"
file_type	string: "indoor", "perana_indoor", "beach", "perana_beach"
style	string: see <a href="#">vr_match_summary()</a>

---

vr_block	<i>Generate block table</i>
----------	-----------------------------

---

**Description**

Generate block table

**Usage**

```
vr_block(x, team, by = "player", style = "default")
```

**Arguments**

x	datavolleyplays: the plays component of an object as returned by <code>datavolley::dv_read</code>
team	string: team name
by	string: "player" or "set"
style	string: see <a href="#">vr_match_summary()</a>

---

vr_court_plots	<i>Court plots</i>
----------------	--------------------

---

**Description**

Court plots

**Usage**

```
vr_court_plots(x, ...)
```

**Arguments**

x	datavolley or string: as returned by <code>datavolley::dv_read</code> , or the path to such a file
...	: passed to downstream functions

**Value**

A ggplot object

**Examples**

```
## Not run:  
vr_court_plots(dv_example_file())  
  
## End(Not run)
```

---

vr_match_summary	<i>Generate single-page match summary report</i>
------------------	--

---

**Description**

Generate single-page match summary report

**Usage**

```
vr_match_summary(  
  x,  
  outfile,  
  refx,  
  vote = TRUE,  
  format = "html",  
  icon = NULL,  
  css = vr_css(),
```

```

remove_nonplaying = TRUE,
style = "default",
update_meta = FALSE,
home_players = TRUE,
visiting_players = TRUE,
base_font_size = 11,
court_plots_function = "vr_court_plots",
court_plots_args = list(),
plot_icons,
skill_evaluation_decode = "guess",
single_page_tries = 1L,
shiny_progress = FALSE,
chrome_print_extra_args = NULL,
...
)

vr_css()

```

### Arguments

x	datavolley or string: as returned by <code>datavolley::dv_read</code> , or the path to such a file
outfile	string: path to file to produce (if not specified, will create a file in the temporary directory)
refx	data.frame: some choices of style require a reference data set to calculate e.g. expected SO. This should be from comparison matches (e.g. all matches from the same league), and should be a data.frame of the plays components from those matches. If missing, expected SO and BP will be replaced by reception and serve efficiency
vote	logical: include vote report component? If not explicitly specified, vote might be set to FALSE depending on style
format	string: "pdf" (using latex-based PDF), "paged_pdf" (using pagedown-based PDF), "png", "paged_png", or "html"
icon	string: (optional) filename of icon image to use
css	list: css specifications for some elements, giving (currently fairly limited) control over appearance. See the output of <code>vr_css</code> for an example. Note that some styling does not seem to be applied when exporting to PDF
remove_nonplaying	logical: if TRUE, remove players from the team summaries that did not take to the court
style	string: can be <ul style="list-style-type: none"> <li>"default" - the standard FIVB match report</li> <li>"ov1" - modified version of "default" with score evolution plot, different breakdown by rotation, and other changes</li> </ul>
update_meta	logical: should we update the match metadata before generating the report? Updating the match metadata will recalculate details such as set scores, player

	starting positions and substitution summaries, and set durations from the scouted play-by-play data
home_players	logical: include a table with individual player statistics for the home team?
visiting_players	logical: include a table with individual player statistics for the visiting team?
base_font_size	numeric: the base font size (the font sizes in different parts of the report are scaled relative to this)
court_plots_function	string or function: a function, or name of a function, that takes a datavolley object and produces a plot object. Supply your own function here to override the court plots that are included in the report for some values of style
court_plots_args	list: named list of arguments to pass to the court plot function
plot_icons	logical or data.frame: some values of style will include plots of various kinds in the report. Currently plot_icons defaults to TRUE for style = "ov1" on a beach match, otherwise FALSE (plot icons generally tend to be visually distracting with indoor, particularly the error icons). Set plot_icons to FALSE for no icons, TRUE to use the icons specified by vr_plot_icons(), or a data.frame as returned by vr_plot_icons() to control the icons that will be used. Note that only (free) fontawesome icons are supported
skill_evaluation_decode	: as for datavolley::dv_read()
single_page_tries	integer: experimental! Ideally we want a single-page report, but until the report is rendered to PDF we don't know for sure whether it will fit on one page. If single_page_tries is greater than 1, we will try re-rendering the report (trying up to this many times). If it does not fit on a single page, the base_font_size will be progressively reduced on each try. Note that this only applies to format "paged_pdf"
shiny_progress	logical: if TRUE, the report generation process will issue shiny::setProgress() calls. The call to vr_match_summary should therefore be wrapped in a shiny::withProgress() scope
chrome_print_extra_args	character: additional parameters to pass as extra_args to pagedown::chrome_print() (only relevant if using a "paged_*" format)
...	: additional parameters passed to the rmarkdown template

## Value

The path to the report file

## Examples

```
## Not run:
f <- vr_match_summary(dv_example_file(), format = "paged_pdf")
if (interactive()) browseURL(f)

## End(Not run)
```

---

vr_plot_icons	<i>Score evolution plot</i>
---------------	-----------------------------

---

**Description**

Score evolution plot

**Usage**

```
vr_plot_icons()

vr_score_evplot(
  x,
  with_summary = FALSE,
  use_icons = FALSE,
  icons,
  home_colour = "darkblue",
  visiting_colour = "darkred",
  low_colour = "#800000",
  mid_colour = "#202020",
  high_colour = "#008000",
  font_size = 12
)
```

**Arguments**

x	datavolley or string: as returned by <code>datavolley::dv_read</code> , or the path to such a file
with_summary	logical: if TRUE, show team summary statistics. For beach, this is by end; for indoor summaries will be by set
use_icons	logical: add icons for ace/error/block kill?. Note that timeout icons are always added
icons	data.frame: a data.frame as returned by <code>vr_plot_icons()</code> . Note that only (free) fontawesome icons are supported
home_colour	string: colour for home team
visiting_colour	string: colour for visiting team
low_colour, mid_colour, high_colour	string: colours for low, mid, and high performance (used for with_summary only)
font_size	scalar: font size

**Value**

A ggplot object

---

vr_points	<i>Generate points table</i>
-----------	------------------------------

---

**Description**

Generate points table

**Usage**

```
vr_points(x, team, by = "player", vote = FALSE, style = "default")
```

**Arguments**

x	datavolleyplays: the plays component of an object as returned by <code>datavolley::dv_read</code>
team	string: team name
by	string: "player" or "set"
vote	logical: if TRUE, include vote detail
style	string: see <a href="#">vr_match_summary()</a>

---

vr_reception	<i>Generate reception table</i>
--------------	---------------------------------

---

**Description**

Generate reception table

**Usage**

```
vr_reception(
  x,
  team,
  by = "player",
  refx,
  style = "default",
  file_type = "indoor"
)
```

**Arguments**

x	datavolleyplays: the plays component of an object as returned by <code>datavolley::dv_read</code>
team	string: team name
by	string: "player" or "set"
refx	data.frame: see <a href="#">vr_match_summary()</a>
style	string: see <a href="#">vr_match_summary()</a>
file_type	string: "indoor", "perana_indoor", "beach", "perana_beach"



---

vr_serve	<i>Generate serve table</i>
----------	-----------------------------

---

**Description**

Generate serve table

**Usage**

```
vr_serve(x, team, by = "player", refx, style = "default")
```

**Arguments**

x	datavolleyplays: the plays component of an object as returned by <code>datavolley::dv_read</code>
team	string: team name
by	string: "player" or "set"
refx	data.frame: see <a href="#">vr_match_summary()</a>
style	string: see <a href="#">vr_match_summary()</a>

---

vr_vote	<i>Calculate vote</i>
---------	-----------------------

---

**Description**

Calculate vote

**Usage**

```
vr_vote(x, team)
```

**Arguments**

x	datavolleyplays: the plays component of an object as returned by <code>datavolley::dv_read</code>
team	string: team name

# Index

`datavolley::dv_read()`, [6](#)  
`pagedown::chrome_print()`, [6](#)  
`volleyreport`, [2](#)  
`volleyreport-package (volleyreport)`, [2](#)  
`vr_attack`, [3](#)  
`vr_block`, [3](#)  
`vr_court_plots`, [4](#)  
`vr_css`, [5](#)  
`vr_css (vr_match_summary)`, [4](#)  
`vr_match_summary`, [4](#)  
`vr_match_summary()`, [3](#), [8](#), [9](#)  
`vr_plot_icons`, [7](#)  
`vr_plot_icons()`, [6](#), [7](#)  
`vr_points`, [8](#)  
`vr_reception`, [8](#)  
`vr_score_evplot (vr_plot_icons)`, [7](#)  
`vr_serve`, [9](#)  
`vr_vote`, [9](#)