

Package ‘altair’

May 7, 2026

Version 4.2.3

Title Interface to 'Altair'

Description Interface to 'Altair' <<https://altair-viz.github.io>>, which itself is a 'Python' interface to 'Vega-Lite' <<https://vega.github.io/vega-lite/>>. This package uses the 'Reticulate' framework <<https://rstudio.github.io/reticulate/>> to manage the interface between R and 'Python'.

SystemRequirements Python (>= 3.6.0), (Python) Altair (>= 4.2.0),
vega_datasets (>= 0.9.0). To use image functions for MacOS: X11

License MIT + file LICENSE

Encoding UTF-8

ByteCompile true

URL <https://github.com/vegawidget/altair>

BugReports <https://github.com/vegawidget/altair/issues>

Imports reticulate (>= 1.23), htmlwidgets, assertthat, magrittr,
utils, vegawidget (>= 0.4.1), repr

Suggests httr, rprojroot, purrr, readr, knitr, rmarkdown, tibble,
listviewer (>= 2.0.0), testthat, pryr, stringr, tidyr, dplyr,
pkgdown, V8, rsvg, png, fs

RoxygenNote 7.2.3

NeedsCompilation no

Author Ian Lyttle [aut, cre] (ORCID: <<https://orcid.org/0000-0001-9962-4849>>),
Haley Jeppson [aut],
Altair Developers [aut],
Alicia Schep [ctb] (ORCID: <<https://orcid.org/0000-0002-3915-0618>>),
Jake Vanderplas [ctb] (Altair library),
Brian Granger [ctb] (Altair library)

Maintainer Ian Lyttle <ijlyttle@me.com>

Repository CRAN

Date/Publication 2023-09-04 03:50:02 UTC

Contents

alt	2
altair_concatenation	3
altair_version	4
as_chart	5
as_vegaspec.altair.vegalite.v4.api.TopLevelMixin	6
check_altair	6
image	7
import_vega_data	7
install_altair	8
knit_print.altair.vegalite.v4.api.TopLevelMixin	9
renderVegawidget	10
vegawidget	10
vegawidgetOutput	10
vega_embed	11
vw_as_json	11
vw_set_base_url	11
Index	12

alt	<i>Altair object</i>
-----	----------------------

Description

Uses the reticulate framework to access the Altair API.

Usage

```
alt
```

Format

An object of class `python.builtin.module` (inherits from `python.builtin.object`) of length 1.

Details

The Altair Python package is exposed through the `alt` object. You can create and add to chart using its methods and classes, as outlined in the [Altair Python documentation](#).

In this package, use the `$` operator wherever you see the `.` operator used in Python.

See Also

[Altair Python documentation](#), [altair: Field Guide to Python Issues](#)

Examples

```
if (interactive()) {
  vega_data <- import_vega_data()

  plot_basic <-
    alt$Chart(vega_data$cars())$
    encode(
      x = "Miles_per_Gallon:Q",
      y = "Horsepower:Q",
      color = "Origin:N"
    )$
    mark_point()

  plot_basic
}
```

altair_concatenation *Altair plot concatenation*

Description

Altair plots can be concatenated using the following operators: +, |, and &

Usage

```
## S3 method for class 'altair.vegalite.v4.api.TopLevelMixin'
e1 | e2

## S3 method for class 'altair.vegalite.v4.api.TopLevelMixin'
e1 + e2

## S3 method for class 'altair.vegalite.v4.api.TopLevelMixin'
e1 & e2
```

Arguments

e1	Altair chart object
e2	Altair chart object

Value

Compound Altair chart object

Examples

```

if (interactive()){

  # Examples using the beaver1 and beaver2 body temperature data sets
  # Layering Charts
  base <- alt$Chart(beaver1)$encode(
    x = alt$X('time'),
    y = alt$Y('temp', scale = alt$Scale(zero = FALSE))
  )

  scatter_plot <- base$mark_point()
  line_plot <- base$mark_line()

  combined_plot <- scatter_plot + line_plot

  # Horizontal Concatenation
  base2 <- alt$Chart(beaver2)$
  encode(
    x = alt$X("time"),
    y = alt$Y("temp", scale = alt$Scale(zero = FALSE))
  )

  scatter_plot2 <- base2$mark_point()
  line_plot2 <- base2$mark_line()

  combined_plot <-
    (scatter_plot + line_plot)$
    properties(title = "Beaver 1", width = 200)

  combined_plot2 <-
    (scatter_plot2 + line_plot2)$
    properties(title = "Beaver 2", width = 200)

  hconcat_plot <- combined_plot | combined_plot2

  # Vertical Concatenation
  vconcat_plot <- combined_plot & combined_plot2

}

```

altair_version

Installed versions of Altair, Vega, etc.

Description

Returns a named list of version tags for Altair, Vega, Vega-Lite, and Vega-Embed

Usage

```
altair_version()
```

Value

named list of version tags

Examples

```
if (interactive()) {  
  altair_version()  
}
```

as_chart

Create Altair chart from vegaspec

Description

Create Altair chart from vegaspec

Usage

```
as_chart(spec)
```

Arguments

spec An object to be coerced to vegaspec, a Vega/Vega-Lite specification

Value

altair object

Examples

```
if (interactive()) {  
  as_chart(vegawidget::spec_mtcars)  
}
```

```
as_vegaspec.altair.vegalite.v4.api.TopLevelMixin
    Coerce to vegaspec
```

Description

See `vegawidget::as_vegaspec` for details.

Usage

```
## S3 method for class 'altair.vegalite.v4.api.TopLevelMixin'
as_vegaspec(spec, ...)
```

Arguments

<code>spec</code>	An object to be coerced to vegaspec, a Vega/Vega-Lite specification
<code>...</code>	Other arguments (attempt to future-proof)

```
check_altair    Check the Altair installation
```

Description

Provides feedback on any differences between your installed version of Altair and the version this package supports.

Usage

```
check_altair(quiet = FALSE)
```

Arguments

<code>quiet</code>	logical, if TRUE, suppresses message upon successful check
--------------------	--

Details

If the supported Altair version is different from your installed version, this function will act according to where the difference in the version numbers:

- major version leads to an **error**
- minor version leads to a **warning**
- patch version leads to a **message**

If there is no difference:

- `quiet = FALSE`, success message showing version-numbers
- `quiet = TRUE`, no message

To install the supported version into a Python environment called "r-reticulate", use `install_altair()`.

Value

invisible NULL, called for side-effects

See Also

[reticulate::py_config\(\)](#), [install_altair\(\)](#), [altair_version\(\)](#)

Examples

```
## Not run:
# not run because it requires Python
check_altair()

## End(Not run)
```

image	<i>Create or write image</i>
-------	------------------------------

Description

See [vegawidget::image](#) for details.

import_vega_data	<i>Import Vega datasets</i>
------------------	-----------------------------

Description

Lets you access Vega datasets.

Usage

```
import_vega_data()
```

Details

Returns the data object in the Python package [vega-datasets](#). In the documentation for this package, the convention is to assign this object to the name `vega_data`.

Value

An S3 object of class `vega_datasets.core.DataLoader`

See Also

[Vega datasets documentation](#)

Examples

```

if (interactive()) {
  vega_data <- import_vega_data()

  # To list available datasets
  print(vega_data$list_datasets())

  # When accessing a dataset, substitute any "-" in the name with a "_"
  print(head(vega_data$sf_temps()))

  # Metadata are available for each dataset:
  print(vega_data$anscombe$references)
  print(vega_data$anscombe$description)
  print(vega_data$anscombe$url)

  # For local datasets, local path is available
  print(vega_data$sf_temps$filepath)
}

```

install_altair

Install Altair Python package

Description

This function wraps installation functions from [reticulate](#) to install the Python packages **altair** and **vega_datasets**.

Usage

```

install_altair(
  method = c("conda", "virtualenv"),
  envname = "r-reticulate",
  version = getOption("altair.python.version"),
  ...
)

```

Arguments

method	character, indicates to use "conda" or "virtualenv"
envname	character, name of environment into which to install
version	character, version of Altair to install. For general use of this package, this is set automatically, so you should not need to specify this.
...	other arguments sent to <code>reticulate::py_install()</code>

Details

This package uses the [reticulate](#) package to make an interface with the [Altair](#) Python package. To promote consistency in usage of **reticulate** among different R packages, it is **recommended** to use a common Python environment, called "r-reticulate".

Depending on your setup, you can create this environment using `reticulate::conda_create()` or `reticulate::virtualenv_create()`, as described in this [reticulate article](#), or in this package's [Installation article](#).

Value

invisible NULL, called for side-effects

See Also

[altair: Installation](#), [reticulate: Using reticulate in an R Package](#), [reticulate: Installing Python Packages](#)

Examples

```
## Not run:
# not run because it requires Python
install_altair()

## End(Not run)
```

```
knit_print.altair.vegalite.v4.api.TopLevelMixin
Knit-print method
```

Description

See `vegawidget::knit_print.vegaspec` for details, particularly on additional packages that may have to be installed.

Usage

```
knit_print.altair.vegalite.v4.api.TopLevelMixin(spec, ..., options = NULL)
```

Arguments

<code>spec</code>	An object to be coerced to <code>vegaspec</code> , a Vega/Vega-Lite specification
<code>...</code>	other arguments
<code>options</code>	list, knitr options

renderVegawidget	<i>Render shiny-output for vegawidget</i>
------------------	---

Description

Deprecated, please use `vegawidget::renderVegawidget`.

Usage

```
renderVegawidget(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

expr	expression that generates a vegawidget. This can be a vegawidget or a vegaspec.
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with <code>quote()</code>)? This is useful if you want to save an expression in a variable.

vegawidget	<i>Create a Vega/Vega-Lite htmlwidget</i>
------------	---

Description

See `vegawidget::vegawidget` for details.

vegawidgetOutput	<i>Shiny-output for vegawidget</i>
------------------	------------------------------------

Description

Deprecated, please use `vegawidget::vegawidgetOutput`.

Usage

```
vegawidgetOutput(outputId, width = "auto", height = "auto")
```

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like "100%", "400px", "auto") or a number, which will be coerced to a string and have "px" appended. For vegawidgets, "auto" is useful because, as of now, the spec determines the size of the widget, then the widget determines the size of the container.

vega_embed	<i>Vega embed options</i>
------------	---------------------------

Description

See `vegawidget::vega_embed` for details.

vw_as_json	<i>Coerce vegaspec to JSON</i>
------------	--------------------------------

Description

Deprecated, please use `vegawidget::vw_as_json`.

Usage

```
vw_as_json(spec, pretty = TRUE)
```

Arguments

<code>spec</code>	An object to be coerced to <code>vegaspec</code> , a Vega/Vega-Lite specification
<code>pretty</code>	<code>logical</code> indicates to use pretty (vs. minified) formatting

Value

`jsonlite::json` object

vw_set_base_url	<i>Set base URL</i>
-----------------	---------------------

Description

See `vegawidget::vw_set_base_url` for details.

Index

- * **datasets**
 - alt, [2](#)
- + .altair.vegalite.v4.api.TopLevelMixin
 - (altair_concatenation), [3](#)
- & .altair.vegalite.v4.api.TopLevelMixin
 - (altair_concatenation), [3](#)

- alt, [2](#)
- altair_concatenation, [3](#)
- altair_version, [4](#)
- altair_version(), [7](#)
- as_chart, [5](#)
- as_vegaspec, [6](#)
- as_vegaspec
 - (as_vegaspec.altair.vegalite.v4.api.TopLevelMixin),
[6](#)
- as_vegaspec.altair.vegalite.v4.api.TopLevelMixin,
[6](#)

- check_altair, [6](#)

- image, [7](#), [7](#)
- import_vega_data, [7](#)
- install_altair, [8](#)
- install_altair(), [6](#), [7](#)

- knit_print.altair.vegalite.v4.api.TopLevelMixin,
[9](#)
- knit_print.vegaspec, [9](#)
- knit_print.vegaspec
 - (knit_print.altair.vegalite.v4.api.TopLevelMixin),
[9](#)

- renderVegawidget, [10](#), [10](#)
- reticulate, [8](#), [9](#)
- reticulate::py_config(), [7](#)
- reticulate::virtualenv_create(), [9](#)

- vega_embed, [11](#), [11](#)
- vegawidget, [10](#), [10](#)
- vegawidgetOutput, [10](#), [10](#)

- vw_as_json, [11](#), [11](#)
- vw_set_base_url, [11](#), [11](#)
- vw_to_bitmap (image), [7](#)
- vw_to_svg (image), [7](#)
- vw_write_png (image), [7](#)
- vw_write_svg (image), [7](#)