

# Package ‘histoslider’

July 22, 2025

**Title** A Histogram Slider Input for 'Shiny'

**Version** 0.1.1

**Description** A histogram slider input binding for use in 'Shiny'. Currently supports creating histograms from numeric, date, and 'date-time' vectors.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** rlang, htmltools, reactR, shiny

**Suggests** testthat, shinytest2, bslib

**NeedsCompilation** no

**Author** Carson Sievert [cre, aut] (ORCID:  
<<https://orcid.org/0000-0002-4958-2844>>),  
Samuel Hogg [ctb, cph] (Original author of histoslider React component)

**Maintainer** Carson Sievert <cpsievert1@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-11-04 00:30:02 UTC

## Contents

input_histoslider	1
update_histoslider	3
<b>Index</b>	<b>4</b>

---

input\_histoslider      *Create a histogram slider*

---

## Description

Creates a Shiny UI input binding to a histogram with a slider "filter" control.

**Usage**

```
input_histoslider(  
  id,  
  label,  
  values,  
  start = NULL,  
  end = NULL,  
  width = "100%",  
  height = 175,  
  breaks = rlang::missing_arg(),  
  options = list()  
)
```

**Arguments**

id	An input id.
label	A label for the input (can be NULL or a string).
values	a vector of numeric values for which the histogram is desired.
start	A numeric value for the starting handle.
end	A numeric value for the ending handle.
width, height	Any valid CSS unit defining the width/height.
breaks	determines how histogram bins are computed (see <a href="#">hist()</a> for possible values and details).
options	a list of <a href="#">histoslider options</a> .

**Value**

A Shiny UI input element.

**See Also**

[update\\_histoslider](#)

**Examples**

```
if (interactive()) {  
  library(shiny)  
  shinyApp(  
    input_histoslider("x", "Random", rnorm(100)),  
    function(input, output) {  
      observe(print(input$x))  
    }  
  )  
}
```

---

update\_histoslider      *Update a histogram slider*

---

### Description

Change the value of a `input_histoslider()` on the client (must be called inside a currently active user session). See [here](#) for an example.

### Usage

```
update_histoslider(  
  id,  
  label = NULL,  
  values = NULL,  
  start = NULL,  
  end = NULL,  
  breaks = rlang::missing_arg(),  
  options = NULL,  
  session = shiny::getDefaultReactiveDomain()  
)
```

### Arguments

<code>id</code>	An input id.
<code>label</code>	A label for the input (can be NULL or a string).
<code>values</code>	a vector of numeric values for which the histogram is desired.
<code>start</code>	A numeric value for the starting handle.
<code>end</code>	A numeric value for the ending handle.
<code>breaks</code>	determines how histogram bins are computed (see <code>hist()</code> for possible values and details).
<code>options</code>	a list of <a href="#">histosliders options</a> .
<code>session</code>	The shiny user session object.

### Details

Any arguments with NULL values will be ignored; they will not result in any changes to the input object on the client.

### Value

Nothing.

### See Also

[input\\_histoslider](#)

# Index

`hist()`, [2](#), [3](#)

`input_histoslider`, [1](#), [3](#)

`input_histoslider()`, [3](#)

`update_histoslider`, [2](#), [3](#)