

# Package: switchboard (via r-universe)

November 8, 2024

**Version** 0.1

**Date** 2021-10-8

**Title** An Agile Widget Engine for Real-Time, Dynamic Visualizations

**Description** An unsorted collection of visualization widgets rendered in 'Tcl/Tk' <<https://www.tcl.tk/>> to generate agile dashboards for your iterative simulations. Widgets include progress bars, counters, eavesdroppers, injectors, switches, and sliders for dynamic manipulation and visualization of simulation parameters.

**Type** Package

**Depends** R (>= 3.3.2)

**Imports** magrittr

**SystemRequirements** Tcl/Tk toolkit (X11 Quarts for Mac)

**VignetteBuilder** R.rsp

**License** GPL (>= 2)

**URL** <http://lajeunesse.myweb.usf.edu/> <https://github.com/mjlajeunesse/>  
<https://www.youtube.com/c/LajeunesseLab/>

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Suggests** R.rsp

**Config/pak/sysreqs** tcl tk tk-dev libx11-dev

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

**RemoteUrl** <https://github.com/mjlajeunesse/switchboard>

**RemoteRef** HEAD

**RemoteSha** 9fc25eb9e0facba3d8da971dff742bc9701aec0b

## Contents

switchboard-package . . . . .	2
benchmark . . . . .	3
caption . . . . .	4
control_slider . . . . .	5
control_slider_pair . . . . .	6
control_slider_Y . . . . .	7
control_slider_Y_pair . . . . .	8
control_switch . . . . .	10
control_switch_pair . . . . .	11
control_switch_trio . . . . .	12
counter . . . . .	13
counter_tally . . . . .	14
eavesdropper . . . . .	15
eavesdropper_2D . . . . .	17
eavesdropper_X . . . . .	18
injector . . . . .	20
injector_2D . . . . .	22
injector_X . . . . .	24
number . . . . .	25
number_pair . . . . .	26
number_quartet . . . . .	28
number_trio . . . . .	29
progress_benchmark . . . . .	30
progress_ibis . . . . .	31
progress_image . . . . .	33
progress_oyster . . . . .	34
progress_percent . . . . .	36
progress_phylloaxis . . . . .	37
progress_pikachu . . . . .	39
progress_ring . . . . .	40
progress_ring_percent . . . . .	41
showGrid . . . . .	43
switchboard . . . . .	44
switchboard_close . . . . .	45
<b>Index</b>	<b>46</b>

---

switchboard-package	<i>Agile widgets for visualizing and interacting with simulations in R.</i>
---------------------	---

---

## Description

**switchboard** is an experimental package for generating agile dashboards for iterative simulations in R. The switchboard window is rendered in 'Tcl/Tk' <https://www.tcl.tk/> which is the GUI (graphical user interface) toolkit of base R. More information about **switchboard** can be found at <http://lajeunesse.myweb.usf.edu/>.

**Author(s)**

Marc J. Lajeunesse (University of South Florida, Tampa USA)

---

benchmark	<i>Benchmark or signal an event or achievement with a giant checkmark.</i>
-----------	--

---

**Description**

The benchmark widget displays the number of simulation iterations.

**Arguments**

eavesdrop	The variable to track.
benchmark	The numerical value associated with eavesdrop that will trigger a visual benchmark on a widget. For example, if eavesdrop is a variable that ranges from 1 to 100, then benchmark = 50 will flag completion at the 50 value.
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

**Value**

Nothing.

**Usage**

```
benchmark(eavesdrop = NULL, benchmark = NA, label = "",
          size = 1, placeOnGrid = c(1, 1))
```

**See Also**

Other eavesdroppers: [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

Examples

```
## Not run:

for (i in 1:250) {
  switchboard(delay = 0.01) %>%
    benchmark(i, benchmark = 125, label = ">125")
}
switchboard_close()

## End(Not run)
```

---

caption	<i>A caption widget</i>
---------	-------------------------

---

Description

The caption widget displays a small title caption and smaller subtitle caption.

Arguments

eavesdrop	Two strings of the caption widget as c("Title", "subTitle").
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.
extendRow	An integer describing the number of columns a row should extend through. Extends the width of the widget but not it's height. Used to better organize text along a row within caption widget. For example, when size = 1 and extendRow = 2 the caption widget will extend to two columns rather than one, but still have a height of one.

Value

Nothing.

Usage

```
caption(c("", ""), size = 1, placeOnGrid = c(1,1))
```

**Examples**

```
## Not run:

for(i in 1:250) {
  switchboard(delay = 0.01) %>%
    caption(c("A Title", "A small sub title"))
}
switchboard_close()

## End(Not run)
```

---

control_slider	<i>A slider to modify the range of a continuous simulation variable.</i>
----------------	--

---

**Description**

The control\_slider widget displays a movable slider to drag and select a new numeric value of a simulation variable.

**Arguments**

inject	String of the variable name to be modified/injected by the slider widget. For example, inject = "i".
minimum	The minimum value of inject.
maximum	The maximum value of inject.
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

**Value**

Nothing.

**Usage**

```
control_slider(inject = "", minimum = 0, maximum = 100,
  label = "", size = 1, placeOnGrid = c(1, 1))
```

**See Also**

Other injectors: `control_slider_Y_pair()`, `control_slider_Y()`, `control_slider_pair()`, `control_switch_pair()`, `control_switch_trio()`, `control_switch()`, `injector_2D()`, `injector_X()`, `injector()`

**Examples**

```
## Not run:

varToSlide <- 0
for (i in 1:500) {
  switchboard(delay = 0.01) %>%
    control_slider("varToSlide", label = "0 to 100") %>%
    number(varToSlide)
}
switchboard_close()

## End(Not run)
```

---

<code>control_slider_pair</code>	<i>A pair of sliders to modify the range of two continuous simulation variables.</i>
----------------------------------	--

---

**Description**

The `control_slider_pair` widget displays two movable sliders to drag and select new numeric values of two simulation variable.

**Arguments**

<code>inject</code>	A vector of the two strings for each variable name to be modified/injected by the two sliders. For example, <code>inject = c("A", "B")</code> .
<code>minimum</code>	A vector of the two minimum values for each variable in <code>inject</code> .
<code>maximum</code>	A vector of the two maximum values for each variable in <code>inject</code> .
<code>label</code>	A vector of the two small caption/labels for each slider.
<code>size</code>	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
<code>placeOnGrid</code>	A row by column coordinate (e.g., <code>c(row-number, column-number)</code> ) of a grid that designates the position to draw the widget on the switchboard. Use <code>showGrid()</code> to help organize widget placement on dashboard. The default places the first widget in pipe chain to the <code>c(1, 1)</code> position, and all following on the same row stacked to the right.

**Value**

Nothing.

**Usage**

```
control_slider_pair(inject = c("", ""), minimum = c(0, 0),
  maximum = c(100, 100), label = "", size = 1, placeOnGrid = c(1, 1))
```

**See Also**

Other injectors: [control\\_slider\\_Y\\_pair\(\)](#), [control\\_slider\\_Y\(\)](#), [control\\_slider\(\)](#), [control\\_switch\\_pair\(\)](#), [control\\_switch\\_trio\(\)](#), [control\\_switch\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

**Examples**

```
## Not run:

A <- 0
B <- 50
for (i in 1:500) {
  switchboard(delay = 0.01) %>%
    control_slider_pair(inject = c("A", "B"),
      label = c("0 to 100", "0 to 100")) %>%
    number_pair(c(A, B))
}
switchboard_close()

## End(Not run)
```

---

control_slider_Y	<i>A vertical slider to modify the range of a continuous simulation variable.</i>
------------------	---

---

**Description**

The control\_slider\_Y widget displays a vertical slider to drag and select a new numeric value of a simulation variable.

**Arguments**

inject	String of the variable name to be modified/injected by the switch widget. For example, inject = "i". The variable should be in boolean form (e.g., 0 or 1, FALSE or TRUE).
minimum	The minimum value of inject.
maximum	The maximum value of inject.
label	A small caption/label for the widget.

size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

**Value**

Nothing.

**Usage**

```
control_slider_Y(inject = "", minimum = 0, maximum = 100,
  label = "", size = 1, placeOnGrid = c(1, 1))
```

**See Also**

Other injectors: [control\\_slider\\_Y\\_pair\(\)](#), [control\\_slider\\_pair\(\)](#), [control\\_slider\(\)](#), [control\\_switch\\_pair\(\)](#), [control\\_switch\\_trio\(\)](#), [control\\_switch\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

**Examples**

```
## Not run:

varToSlide <- 0
for (i in 1:500) {
  switchboard(delay = 0.01) %>%
    control_slider_Y("varToSlide", label = "0 to 100") %>%
    number(varToSlide)
}
switchboard_close()

## End(Not run)
```

---

`control_slider_Y_pair` *A pair of vertical sliders to modify the range of two continuous simulation variables.*

---

**Description**

The `control_slider_Y_pair` widget displays two vertical sliders to drag and select new numeric values of two simulation variable.



**Arguments**

inject	A vector of the two strings for each variable name to be modified/injected by the two sliders. For example, inject = c("A", "B").
minimum	A vector of the two minimum values for each variable in inject.
maximum	A vector of the two maximum values for each variable in inject.
label	A vector of the two small caption/labels for each slider.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

**Value**

Nothing.

**Usage**

```
control_slider_Y_pair(inject = c("", ""), minimum = c(0, 0),
  maximum = c(100, 100), label = "", size = 1, placeOnGrid = c(1, 1))
```

**See Also**

Other injectors: [control\\_slider\\_Y\(\)](#), [control\\_slider\\_pair\(\)](#), [control\\_slider\(\)](#), [control\\_switch\\_pair\(\)](#), [control\\_switch\\_trio\(\)](#), [control\\_switch\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

**Examples**

```
## Not run:
```

```
varToSlideA <- 0
varToSlideB <- 50
for (i in 1:500) {
  switchboard(delay = 0.01) %>%
    control_slider_pair_Y(inject = c("varToSlideA", "varToSlideB"),
      label = c("0 to 100", "0 to 100")) %>%
    number_pair(c(varToSlideA, varToSlideB))
}
```

```
switchboard_close()
```

```
## End(Not run)
```

---

control_switch	<i>A switch to toggle on/off a simulation variable.</i>
----------------	---

---

## Description

The control\_switch widget displays a switch to toggle the state (e.g., on or off, TRUE or FALSE) of a simulation variable. A switch is blue when "on", and gray when "off".

## Arguments

inject	String of the variable name to be modified/injected by the switch widget. For example, inject = "i". The variable should be in boolean form (e.g., 0 or 1, FALSE or TRUE).
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

## Value

Nothing.

## Usage

```
control_switch(inject = "", label = "", size = 1, placeOnGrid = c(1, 1))
```

## See Also

Other injectors: [control\\_slider\\_Y\\_pair\(\)](#), [control\\_slider\\_Y\(\)](#), [control\\_slider\\_pair\(\)](#), [control\\_slider\(\)](#), [control\\_switch\\_pair\(\)](#), [control\\_switch\\_trio\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

## Examples

```
## Not run:

varToSwitch <- 0
for (i in 1:500) {
  switchboard(delay = 0.01) %>%
    control_switch("varToSwitch", label = "0 to 1") %>%
    number(varToSwitch)
```

```

    }
    switchboard_close()

## End(Not run)

```

---

control\_switch\_pair     *A pair of switches to toggle on/off two simulation variables.*

---

## Description

The control\_switch\_pair widget displays two switches to toggle the state (e.g., on or off, TRUE or FALSE) of two simulation variables. A switch is blue when "on", and gray when "off".

## Arguments

inject	A vector of the two strings for each variable name to be modified/injected by the switch widget. For example, inject = c("A", "B"). These two variables should be in boolean form (e.g., 0 or 1, FALSE or TRUE).
label	A vector of two short strings designating labels/captions for each switch.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

## Value

Nothing.

## Usage

```
control_switch_pair(inject = c("", ""), label = c("", ""),
  size = 1, placeOnGrid = c(1, 1))
```

## See Also

Other injectors: [control\\_slider\\_Y\\_pair\(\)](#), [control\\_slider\\_Y\(\)](#), [control\\_slider\\_pair\(\)](#), [control\\_slider\(\)](#), [control\\_switch\\_trio\(\)](#), [control\\_switch\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

## Examples

```
## Not run:

varToSlideA <- 0
varToSlideB <- 0
for (i in 1:500) {
  switchboard(delay = 0.01) %>%
    control_switch_pair(inject = c("varToSlideA", "varToSlideB"),
                        label = c("0 to 1", "0 to 1")) %>%
    number_pair(c(varToSlideA, varToSlideB))
}
switchboard_close()

## End(Not run)
```

---

control\_switch\_trio     *A trio of switches to toggle on/off three simulation variables.*

---

## Description

The control\_switch\_trio widget displays three switches to toggle the state (e.g., on or off, TRUE or FALSE) of three simulation variables. A switch is blue when "on", and gray when "off".

## Arguments

inject	A vector of the three strings for each variable name to be modified/injected by the switch widget. For example, inject = c("A", "B", "C"). These three variables should be in boolean form (e.g., 0 or 1, FALSE or TRUE).
label	A vector of three short strings designating labels/captions for each switch.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

## Value

Nothing.

**Usage**

```
switch(inject = c("", "", ""), label = c("", "", ""),
       size = 1, placeOnGrid = c(1, 1))
```

**See Also**

Other injectors: [control\\_slider\\_Y\\_pair\(\)](#), [control\\_slider\\_Y\(\)](#), [control\\_slider\\_pair\(\)](#), [control\\_slider\(\)](#), [control\\_switch\\_pair\(\)](#), [control\\_switch\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

**Examples**

```
## Not run:

varToSlideA <- 0
varToSlideB <- 0
varToSlideC <- 1
for (i in 1:500) {
  switchboard(delay = 0.01) %>%
    control_switch_trio(inject = c("varToSlideA", "varToSlideB", "varToSlideC"),
                       label = c("0 to 1", "0 to 1", "1 to 0")) %>%
    number_trio(c(varToSlideA, varToSlideB, varToSlideC))
}
switchboard_close()

## End(Not run)
```

---

counter

*A counter to keep track of the number of simulation iterations.*


---

**Description**

The counter widget displays the number of simulation iterations.

**Arguments**

label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., <code>c(row-number, column-number)</code> ) of a grid that designates the position to draw the widget on the switchboard. Use <code>showGrid()</code> to help organize widget placement on dashboard. The default places the first widget in pipe chain to the <code>c(1, 1)</code> position, and all following on the same row stacked to the right.

**Value**

Nothing.

**Usage**

```
counter(label = "", size = 1, placeOnGrid = c(1, 1))
```

**See Also**

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

**Examples**

```
## Not run:

for (i in 1:250) {
  switchboard(delay = 0.01) %>%
    counter(label = "done at 250")
}
switchboard_close()

## End(Not run)
```

---

counter\_tally

*A tally counter to keep track of the number of simulation iterations.*

---

**Description**

The counter\_tally widget displays a tally of simulation iterations.

**Arguments**

label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

**Value**

Nothing.

**Usage**

```
counter_tally(label = "", size = 1, placeOnGrid = c(1, 1))
```

**See Also**

Other eavesdroppers: [benchmark\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

**Examples**

```
## Not run:

  for (i in 1:250) {
    switchboard(delay = 0.01) %>%
      counter_tally(label = "tallying 250")
  }
  switchboard_close()

## End(Not run)
```

---

eavesdropper

*Eavesdrop a continuous variable.*


---

**Description**

The eavesdropper widget displays a horizontal moving window of data (see [eavesdropper\\_X](#) for a vertical window). The X-axis is time-lagged and the widget keeps track of each point until it reaches the end of the plot. This widget is univariate and only eavesdrops one variable; for a bivariate version use [eavesdropper\\_2D](#). The number of data points in the window can be throttled with [delay](#).

**Arguments**

eavesdrop	The variable to track.
minimum	The minimum value of eavesdrop. It defines the plot boundary
maximum	The maximum value of eavesdrop. It defines the plot boundary
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.

placeOnGrid	A row by column coordinate (e.g., <code>c(row-number, column-number)</code> ) of a grid that designates the position to draw the widget on the switchboard. Use <code>showGrid()</code> to help organize widget placement on dashboard. The default places the first widget in pipe chain to the <code>c(1, 1)</code> position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., <code>delay = 0.01</code> results in 0.01 second delay with each iteration).
plotMean	Display a small switchboard-estimated mean and standard deviation whisker plot on a widget.
switch	Display an on/off switch on a widget that controls widget updates. When TRUE it will add the switch in the off-state on the switchboard. The user must activate the switch to start the widget. The simulation proceeds even if the switch in the off-state.

**Value**

Nothing.

**Usage**

```
eavesdropper(eavesdrop, minimum = 1, maximum = 100,
  label = "", size = 1, placeOnGrid = c(1, 1), updates = 100, delay = 0,
  plotMean = FALSE, switch = FALSE)
```

**See Also**

Other moving windows: [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

**Examples**

```
## Not run:

  for (i in 1:400) {
    randomNormal <- rnorm(1, 0, 1)
    switchboard(delay = 0.01) %>%
      eavesdropper(randomNormal, minimum = -5, maximum = 5)
  }
  switchboard_close()

## End(Not run)
```



eavesdropper\_2D

*Eavesdrop two continuous variables on a bivariate plot.***Description**

The eavesdropper\_2D widget displays a bivariate window of data. The plotted data have a half-life and are deleted once their timer (parameter forget) expires. For the univariate version use eavesdropper. The number of data points in the window can be throttled with forget or delay. Options also include automatic regression (plotRegression) and an the sample size of the number of data points currently displayed (plotSampleSize).

**Arguments**

eavesdrop	A vector of two variables to track on a bivariate plot. For example, eavesdrop = c(A, B) would result on A being plotted on X-axis (independent variable) and B on Y-axis (dependent variable).
minimum	A vector of the two minimum values of variables in eavesdrop. For example, eavesdrop = c(A, B) then minimum = c(C, D) are the minimum values of A and B, respectively. They define plot boundaries.
maximum	A vector of the two maximum values of variables in eavesdrop. For example, eavesdrop = c(A, B) then maximum = c(C, D) are the maximum values of A and B, respectively. They define plot boundaries.
label	A vector of two short strings designating labels/captions for each variable used in eavesdrop.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
forget	A time-delay in milliseconds for when displayed points on a widget will be deleted.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).
plotRegression	Display a switchboard-estimated regression line on a widget.
plotSampleSize	Display the number items (N) displayed within a widget.

**switch**                      Display an on/off switch on a widget that controls widget updates. When TRUE it will add the switch in the off-state on the switchboard. The user must activate the switch to start the widget. The simulation proceeds even if the switch in in the off-state.

### Value

Nothing.

### Usage

```
eavesdropper_2D(eavesdrop = c(NULL, NULL), minimum = c(1, 1),
  maximum = c(1, 1), label = c("", ""), size = 1, placeOnGrid = c(1, 1),
  updates = 100, forget = 400, delay = 0, plotRegression = FALSE,
  plotSampleSize = FALSE, switch = FALSE)
```

### See Also

Other moving windows: [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

### Examples

```
## Not run:

  for (i in 1:10000) {
    x <- cos(i/400)
    y <- sin(2 * i/400) / 2
    switchboard(skip = 4) %>%
      eavesdropper_2D(c(x, y), minimum = c(-1, -1), maximum = c(1, 1), forget = 100)
  }
  switchboard_close()

## End(Not run)
```

---

eavesdropper\_X

*Eavesdrop a continuous variable as a conveyor belt.*

---

### Description

The `eavesdropper_X` widget displays a vertical moving window of data (see `eavesdropper` for a horizontal window). The Y-axis is time-lagged and the widget keeps track of each point until it reaches the end of the plot. This widget is univariate and only eavesdrops one variable; for a bivariate version use `eavesdropper_2D`. The number of data points in the window can be throttled with `delay`.

**Arguments**

eavesdrop	The variable to track.
minimum	The minimum value of eavesdrop. It defines the plot boundary
maximum	The maximum value of eavesdrop. It defines the plot boundary
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., <code>c(row-number, column-number)</code> ) of a grid that designates the position to draw the widget on the switchboard. Use <code>showGrid()</code> to help organize widget placement on dashboard. The default places the first widget in pipe chain to the <code>c(1, 1)</code> position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., <code>delay = 0.01</code> results in 0.01 second delay with each iteration).
plotMean	Display a small switchboard-estimated mean and standard deviation whisker plot on a widget.
switch	Display an on/off switch on a widget that controls widget updates. When TRUE it will add the switch in the off-state on the switchboard. The user must activate the switch to start the widget. The simulation proceeds even if the switch in the off-state.

**Value**

Nothing.

**Usage**

```
eavesdropper_X(eavesdrop, minimum = 1, maximum = 100,
  label = "", size = 1, placeOnGrid = c(1, 1), updates = 100, delay = 0,
  plotMean = FALSE, switch = FALSE)
```

**See Also**

Other moving windows: [eavesdropper\\_2D\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

## Examples

```
## Not run:

for (i in 1:400) {
  randomNormal <- rnorm(1, 0, 1)
  switchboard(delay = 0.01) %>%
    eavesdropper_X(randomNormal, minimum = -5, maximum = 5)
}
switchboard_close()

## End(Not run)
```

---

injector

*Inject and eavesdrop a continuous variable.*


---

## Description

The injector widget displays a horizontal moving window of data with a slider that injects/modifies characteristics of the data. (see `injector_X` for a vertical version). The X-axis is time-lagged and the widget keeps track of each point until it reaches the end of the plot. This widget is univariate and only eavesdrops one variable; for a bivariate version use `injector_2D`. The number of data points in the window can be throttled with `delay`.

## Arguments

eavesdrop	The variable to track.
minimum	The minimum value of eavesdrop. It defines the plot boundary
maximum	The maximum value of eavesdrop. It defines the plot boundary
inject	String of the variable name to be modified/injected by widgets. For example, if <code>eavesdrop = i</code> then <code>inject = "i"</code> .
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., <code>c(row-number, column-number)</code> ) of a grid that designates the position to draw the widget on the switchboard. Use <code>showGrid()</code> to help organize widget placement on dashboard. The default places the first widget in pipe chain to the <code>c(1, 1)</code> position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.

delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).
plotMean	Display a small switchboard-estimated mean and standard deviation whisker plot on a widget.
switch	Display an on/off switch on a widget that controls widget updates. When TRUE it will add the switch in the off-state on the switchboard. The user must activate the switch to start the widget. The simulation proceeds even if the switch in the off-state.

### Value

Nothing.

### Usage

```
injector(eavesdrop, minimum = 1, maximum = 100, inject = "",
  label = "", size = 1, placeOnGrid = c(1, 1), updates = 100, delay = 0,
  plotMean = FALSE, switch = FALSE)
```

### See Also

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

Other injectors: [control\\_slider\\_Y\\_pair\(\)](#), [control\\_slider\\_Y\(\)](#), [control\\_slider\\_pair\(\)](#), [control\\_slider\(\)](#), [control\\_switch\\_pair\(\)](#), [control\\_switch\\_trio\(\)](#), [control\\_switch\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#)

Other moving windows: [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#)

### Examples

## Not run:

```
populationMean = 0
for (i in 1:4000) {
  randomNormal <- rnorm(1, populationMean, 1)
  switchboard(delay = 0.01) %>%
    injector(randomNormal, inject = "populationMean", minimum = -5, maximum = 5)
}
switchboard_close()
```

## End(Not run)

injector\_2D

*Inject and eavesdrop two continuous variables on a bivariate plot.***Description**

The `injector_2D` widget displays a bivariate window of data with two sliders that injects/modifies characteristics of the data. The plotted data have a half-life and are deleted once their timer (parameter `forget`) expires. For the univariate version use `injector`. The number of data points in the window can be throttled with `forget` or `delay`. Options also include automatic regression (`plotRegression`) and an the sample size of the number of data points currently displayed (`plotSampleSize`).

**Arguments**

<code>eavesdrop</code>	A vector of two variables to track on a bivariate plot. For example, <code>eavesdrop = c(A, B)</code> would result on A being plotted on X-axis (independent variable) and B on Y-axis (dependent variable).
<code>minimum</code>	A vector of the two minimum values of variables in <code>eavesdrop</code> . For example, <code>eavesdrop = c(A, B)</code> then <code>minimum = c(C, D)</code> are the minimum values of A and B, respectively. They define plot boundaries.
<code>maximum</code>	A vector of the two maximum values of variables in <code>eavesdrop</code> . For example, <code>eavesdrop = c(A, B)</code> then <code>maximum = c(C, D)</code> are the maximum values of A and B, respectively. They define plot boundaries.
<code>inject</code>	A vector of the two strings for each variable name to be modified/injected by widgets. For example, if <code>eavesdrop = c(A, B)</code> then <code>inject = c("A", "B")</code> .
<code>label</code>	A vector of the two short strings designating labels/captions for each of in <code>eavesdrop</code> .
<code>size</code>	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
<code>placeOnGrid</code>	A row by column coordinate (e.g., <code>c(row-number, column-number)</code> ) of a grid that designates the position to draw the widget on the switchboard. Use <code>showGrid()</code> to help organize widget placement on dashboard. The default places the first widget in pipe chain to the <code>c(1, 1)</code> position, and all following on the same row stacked to the right.
<code>updates</code>	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
<code>forget</code>	A time-delay in milliseconds for when displayed points on a widget will be deleted.
<code>delay</code>	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., <code>delay = 0.01</code> results in 0.01 second delay with each iteration).
<code>plotRegression</code>	Display a switchboard-estimated regression line on a widget.

`plotSampleSize` Display the number items (N) displayed within a widget.

`switch` Display an on/off switch on a widget that controls widget updates. When TRUE it will add the switch in the off-state on the switchboard. The user must activate the switch to start the widget. The simulation proceeds even if the switch in the off-state.

### Value

Nothing.

### Usage

```
injector_2D(eavesdrop = c(NULL, NULL), minimum = c(1, 1),
  maximum = c(1, 1), inject = c("", "") label = c("", ""), size = 1,
  placeOnGrid = c(1, 1), updates = 100, forget = 400, delay = 0,
  plotRegression = FALSE, plotSampleSize = FALSE, switch = FALSE)
```

### See Also

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

Other injectors: [control\\_slider\\_Y\\_pair\(\)](#), [control\\_slider\\_Y\(\)](#), [control\\_slider\\_pair\(\)](#), [control\\_slider\(\)](#), [control\\_switch\\_pair\(\)](#), [control\\_switch\\_trio\(\)](#), [control\\_switch\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

Other moving windows: [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#)

### Examples

## Not run:

```
populationMean = 0
populationSD = 1
for (i in 1:4000) {
  randomNormal <- rnorm(1, populationMean, populationSD)
  switchboard(delay = 0.01) %>%
    injector_2D(c(populationSD, randomNormal),
      inject = c("populationSD", "populationMean"),
      minimum = c(0.1, - 5),
      maximum = c(3.1, 5),
      label = c("pop mean", "pop SD"),
      forget = 2000)
}
switchboard_close()
```

## End(Not run)

injector\_X

*Inject a continuous variable and display as conveyor belt.***Description**

The injector\_X widget displays a vertical moving window of data with a slider that injects/modifies characteristics of the data. (see injector for a horizontal version). The Y-axis is time-lagged and the widget keeps track of each point until it reaches the end of the plot. This widget is univariate and only eavesdrops one variable; for a bivariate version use injector\_2D. The number of data points in the window can be throttled with delay.

**Arguments**

eavesdrop	The variable to track.
minimum	The minimum value of eavesdrop. It defines the plot boundary.
maximum	The maximum value of eavesdrop. It defines the plot boundary.
inject	String of the variable name to be modified/injected by widgets. For example, if eavesdrop = i then inject = "i".
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).
plotMean	Display a small switchboard-estimated mean and standard deviation whisker plot on a widget.
switch	Display an on/off switch on a widget that controls widget updates. When TRUE it will add the switch in the off-state on the switchboard. The user must activate the switch to start the widget. The simulation proceeds even if the switch in in the off-state.

**Value**

Nothing.



**Usage**

```
injector_X(eavesdrop, minimum = 1, maximum = 100, inject = "",
  label = "", size = 1, placeOnGrid = c(1, 1), updates = 100, delay = 0,
  plotMean = FALSE, switch = FALSE)
```

**See Also**

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

Other injectors: [control\\_slider\\_Y\\_pair\(\)](#), [control\\_slider\\_Y\(\)](#), [control\\_slider\\_pair\(\)](#), [control\\_slider\(\)](#), [control\\_switch\\_pair\(\)](#), [control\\_switch\\_trio\(\)](#), [control\\_switch\(\)](#), [injector\\_2D\(\)](#), [injector\(\)](#)

Other moving windows: [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\(\)](#)

**Examples**

```
## Not run:
```

```
populationMean = 0
for (i in 1:4000) {
  randomNormal <- rnorm(1, populationMean, 1)
  switchboard(delay = 0.01) %>%
    injector_X(randomNormal, inject = "populationMean", minimum = -5, maximum = 5)
}
switchboard_close()
```

```
## End(Not run)
```

---

number

---

*Display the value of a continuous variable in a simulation.*


---

**Description**

The number widget displays the numerical value of a continuous variable.

**Arguments**

eavesdrop	The variable to track.
digits	The number of digits to display in a widget.
label	A small caption/label for the widget.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates with each iteration.

size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

**Value**

Nothing.

**Usage**

```
number(eavesdrop = NULL, digits = 5, updates = 1, label = "",
       size = 1, placeOnGrid = c(1, 1))
```

**See Also**

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [progress\\_benchmark\(\)](#)

**Examples**

```
## Not run:

  for (i in 1:250) {
    switchboard(delay = 0.01) %>%
      number(i * i, label = "i * i")
  }
  switchboard_close()

## End(Not run)
```

---

number\_pair

*Display the value of two continuous variables in a simulation.*

---

**Description**

The number\_pair widget displays the numerical value of two continuous variables.

**Arguments**

eavesdrop	A vector of two variables to track.
digits	The number of digits to display in a widget.
label	A vector of two small captions/labels for the widget.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates with each iteration.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

**Value**

Nothing.

**Usage**

```
number_pair(eavesdrop = c(NULL, NULL), digits = 5,
            updates = 1, label = c("", ""), size = 1, placeOnGrid = c(1, 1))
```

**See Also**

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

**Examples**

```
## Not run:

for (i in 1:250) {
  switchboard(delay = 0.01) %>%
    number_pair(c(i, i/2), label = c("i", "i/2"))
}
switchboard_close()

## End(Not run)
```

---

number_quartet	<i>Display the value of four continuous variables in a simulation.</i>
----------------	--

---

## Description

The number\_quartet widget displays the numerical value of four continuous variables.

## Arguments

eavesdrop	A vector of four variables to track.
digits	The number of digits to display in a widget.
label	A vector of four small captions/labels for the widget.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates with each iteration.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

## Value

Nothing.

## Usage

```
number_quartet(eavesdrop = c(NULL, NULL, NULL, NULL),
  digits = 5, updates = 1, label = c("", "", "", ""), size = 1,
  placeOnGrid = c(1, 1))
```

## See Also

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_trio\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

## Examples

```
## Not run:

for (i in 1:250) {
  switchboard(delay = 0.01) %>%
```

```

        number_quartet(c(i, i/2, i/3, i /4), label = c("i", "i/2", "i/3", "i/4"))
    }
    switchboard_close()

## End(Not run)

```

---

number\_trio

---

*Display the value of three continuous variables in a simulation.*


---

## Description

The number\_trio widget displays the numerical value of three continuous variables.

## Arguments

eavesdrop	A vector of three variables to track.
digits	The number of digits to display in a widget.
label	A vector of three small captions/labels for the widget.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates with each iteration.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.

## Value

Nothing.

## Usage

```

number_trio(eavesdrop = c(NULL, NULL, NULL), digits = 5,
            updates = 1, label = c("", "", ""), size = 1, placeOnGrid = c(1, 1))

```

## See Also

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\(\)](#), [progress\\_benchmark\(\)](#)

## Examples

```
## Not run:

for (i in 1:250) {
  switchboard(delay = 0.01) %>%
    number_trio(c(i, i/2, i/3), label = c("i", "i/2", "i/3"))
}
switchboard_close()

## End(Not run)
```

---

progress_benchmark	<i>Display percent-complete along with an event benchmark.</i>
--------------------	--

---

## Description

The `progress_benchmark` widget displays a progress bar with a percent complete and a triggerable benchmark as a giant checkmark.

## Arguments

eavesdrop	The variable to track with a progress bar.
maximum	The maximum value of eavesdrop that marks the end of what to progress track.
benchmark	The numerical value associated with eavesdrop that will trigger a visual benchmark on a widget. For example, if eavesdrop is a variable that ranges from 1 to 100, then benchmark = 50 will flag completion at the 50 value.
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., <code>c(row-number, column-number)</code> ) of a grid that designates the position to draw the widget on the switchboard. Use <code>showGrid()</code> to help organize widget placement on dashboard. The default places the first widget in pipe chain to the <code>c(1, 1)</code> position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., <code>delay = 0.01</code> results in 0.01 second delay with each iteration).
honest	When TRUE, it updates the widget by the true progression value. The default (FALSE) has a cosmetic modification to the progression value that helps update it in a prettier way.

`closeAtMaximum` Functions like `switchboard_close()` by closing the switchboard window when the eavesdropped value equals maximum. NOTE: if a widget has `closeAtMaximum = TRUE`, then this widget MUST be placed at the end (i.e., last widget) of the pipe chain.

## Value

Nothing.

## Usage

```
progress_benchmark(eavesdrop, maximum = 100, benchmark = NA,
  caption = "", size = 1, placeOnGrid = c(1, 1), updates = 100, delay = 0,
  honest = FALSE, closeAtMaximum = FALSE)
```

## See Also

Other progress bars: [progress\\_ibis\(\)](#), [progress\\_image\(\)](#), [progress\\_oyster\(\)](#), [progress\\_percent\(\)](#), [progress\\_phylloaxis\(\)](#), [progress\\_pikachu\(\)](#), [progress\\_ring\\_percent\(\)](#), [progress\\_ring\(\)](#)

Other eavesdroppers: [benchmark\(\)](#), [counter\\_tally\(\)](#), [counter\(\)](#), [eavesdropper\\_2D\(\)](#), [eavesdropper\\_X\(\)](#), [eavesdropper\(\)](#), [injector\\_2D\(\)](#), [injector\\_X\(\)](#), [injector\(\)](#), [number\\_pair\(\)](#), [number\\_quartet\(\)](#), [number\\_trio\(\)](#), [number\(\)](#)

## Examples

```
## Not run:

  for (i in 1:250) {
    switchboard(delay = 0.01) %>%
      progress_benchmark(i, maximum = 250, benchmark = 125)
  }
  switchboard_close()

## End(Not run)
```

---

progress\_ibis

*Display an ibis as a progress bar.*

---

## Description

The `progress_ibis` widget displays an ibis – a long-legged, long-beak wading bird – as a `progressBar`.

**Arguments**

eavesdrop	The variable to track with a progress bar.
maximum	The maximum value of eavesdrop that marks the end of what to progress track.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).
honest	When TRUE, it updates the widget by the true progression value. The default (FALSE) has a cosmetic modification to the progression value that helps update it in a prettier way.
closeAtMaximum	Functions like switchboard_close() by closing the switchboard window when the eavesdropped value equals maximum. NOTE: if a widget has closeAtMaximum = TRUE, then this widget MUST be placed at the end (i.e., last widget) of the pipe chain.
fill	The direction of how things are animated when displaying progression. The default is horizontal, which tracks progression from left to right (maximum), but vertical can also be used for progress to occur in a bottom to up (maximum) animation.

**Value**

Nothing.

**Usage**

```
progress_ibis(eavesdrop, maximum = 100, size = 1,
  placeOnGrid = c(1, 1), updates = 100, delay = 0, honest = FALSE,
  closeAtMaximum = FALSE, fill = "horizontal")
```

**See Also**

Other progress bars: [progress\\_benchmark\(\)](#), [progress\\_image\(\)](#), [progress\\_oyster\(\)](#), [progress\\_percent\(\)](#), [progress\\_phylloaxis\(\)](#), [progress\\_pikachu\(\)](#), [progress\\_ring\\_percent\(\)](#), [progress\\_ring\(\)](#)



**Examples**

```
## Not run:

for (i in 1:250) {
  switchboard(delay = 0.01) %>%
    progress_ibis(i, maximum = 250)
}
switchboard_close()

## End(Not run)
```

---

progress_image	<i>Display a custom 80 by 80 pixel image as a progress bar.</i>
----------------	---

---

**Description**

The progress\_image widget displays an image as a progressBar.

**Arguments**

eavesdrop	The variable to track with a progress bar.
maximum	The maximum value of eavesdrop that marks the end of what to progress track.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).
honest	When TRUE, it updates the widget by the true progression value. The default (FALSE) has a cosmetic modification to the progression value that helps update it in a prettier way.
closeAtMaximum	Functions like switchboard_close() by closing the switchboard window when the eavesdropped value equals maximum. NOTE: if a widget has closeAtMaximum = TRUE, then this widget MUST be placed at the end (i.e., last widget) of the pipe chain.

fill	The direction of how things are animated when displaying progression. The default is horizontal, which tracks progression from left to right (maximum), but vertical can also be used for progress to occur in a bottom to up (maximum) animation.
file	A *.png filename with a transparent background that designates an image to be used as a progress-bar. The total image size should be 80 by 80 pixels, but for best integration into switchboard layout, the actual image must be 75 by 75 pixels placed at the bottom left of the 80 by 80 image. This will leave a 5 pixel whitespace at both the top and right side of the image.

**Value**

Nothing.

**Usage**

```
progress_image(eavesdrop, maximum = 100, size = 1,
  placeOnGrid = c(1, 1), updates = 100, delay = 0, honest = FALSE,
  closeAtMaximum = FALSE, fill = "horizontal", file = "")
```

**See Also**

Other progress bars: [progress\\_benchmark\(\)](#), [progress\\_ibis\(\)](#), [progress\\_oyster\(\)](#), [progress\\_percent\(\)](#), [progress\\_phylloaxis\(\)](#), [progress\\_pikachu\(\)](#), [progress\\_ring\\_percent\(\)](#), [progress\\_ring\(\)](#)

**Examples**

```
## Not run:

  for (i in 1:250) {
    switchboard(delay = 0.01) %>%
      progress_image(i, maximum = 250,
        file = system.file("images", "topography.png", package = "switchboard"))
  }
  switchboard_close()

## End(Not run)
```

---

progress\_oyster

*Display a medieval oyster as a progress bar.*

---

**Description**

The progress\_oyster widget displays the Der naturen bloeme "angry oyster" as a progressBar. Scholars thinks it's more of a fan mussel, but it's still clearly angry at someone, something, or itself.

**Arguments**

eavesdrop	The variable to track with a progress bar.
maximum	The maximum value of eavesdrop that marks the end of what to progress track.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).
honest	When TRUE, it updates the widget by the true progression value. The default (FALSE) has a cosmetic modification to the progression value that helps update it in a prettier way.
closeAtMaximum	Functions like switchboard_close() by closing the switchboard window when the eavesdropped value equals maximum. NOTE: if a widget has closeAtMaximum = TRUE, then this widget MUST be placed at the end (i.e., last widget) of the pipe chain.
fill	The direction of how things are animated when displaying progression. The default is horizontal, which tracks progression from left to right (maximum), but vertical can also be used for progress to occur in a bottom to up (maximum) animation.

**Value**

Nothing.

**Usage**

```
progress_oyster(eavesdrop, maximum = 100, size = 1,
  placeOnGrid = c(1, 1), updates = 100, delay = 0, honest = FALSE,
  closeAtMaximum = FALSE, fill = "horizontal")
```

**See Also**

Other progress bars: [progress\\_benchmark\(\)](#), [progress\\_ibis\(\)](#), [progress\\_image\(\)](#), [progress\\_percent\(\)](#), [progress\\_phylloaxis\(\)](#), [progress\\_pikachu\(\)](#), [progress\\_ring\\_percent\(\)](#), [progress\\_ring\(\)](#)

## Examples

```
## Not run:

for (i in 1:250) {
  switchboard(delay = 0.01) %>%
    progress_oyster(i, maximum = 250)
}
switchboard_close()

## End(Not run)
```

---

progress_percent	<i>Display a percentage.</i>
------------------	------------------------------

---

## Description

The progress\_percent widget displays a text description of a percentage value between 0

## Arguments

eavesdrop	The variable to track with a progress bar.
caption	A small text string describing the percent value context.
maximum	The maximum value of eavesdrop that marks the end of what to progress/track.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).
honest	When TRUE, it updates the widget by the true progression value. The default (FALSE) has a cosmetic modification to the progression value that helps update it in a prettier way.
closeAtMaximum	Functions like switchboard_close() by closing the switchboard window when the eavesdropped value equals maximum. NOTE: if a widget has closeAtMaximum = TRUE, then this widget MUST be placed at the end (i.e., last widget) of the pipe chain.

**Value**

Nothing.

**Usage**

```
progress_percent(eavesdrop, caption = "", maximum = 100,
  size = 1, placeOnGrid = c(1, 1), updates = 100, delay = 0, honest = FALSE,
  closeAtMaximum = FALSE)
```

**See Also**

Other progress bars: [progress\\_benchmark\(\)](#), [progress\\_ibis\(\)](#), [progress\\_image\(\)](#), [progress\\_oyster\(\)](#), [progress\\_phyllotaxis\(\)](#), [progress\\_pikachu\(\)](#), [progress\\_ring\\_percent\(\)](#), [progress\\_ring\(\)](#)

**Examples**

```
## Not run:

  for (i in 1:250) {
    switchboard(delay = 0.01) %>%
      progress_percent(i, maximum = 250)
  }
  switchboard_close()

## End(Not run)
```

---

progress\_phyllotaxis    *Display an organic-like geometric spiral pattern as a progress bar.*

---

**Description**

The progress\_phyllotaxis widget displays the growth of geometric spiral pattern as a progress bar. You know, like the way aloe plants grow.

**Arguments**

eavesdrop	The variable to track with a progress bar.
maximum	The maximum value of eavesdrop that marks the end of what to progress/track.
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.

placeOnGrid	A row by column coordinate (e.g., <code>c(row-number, column-number)</code> ) of a grid that designates the position to draw the widget on the switchboard. Use <code>showGrid()</code> to help organize widget placement on dashboard. The default places the first widget in pipe chain to the <code>c(1, 1)</code> position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., <code>delay = 0.01</code> results in 0.01 second delay with each iteration).
honest	When TRUE, it updates the widget by the true progression value. The default (FALSE) has a cosmetic modification to the progression value that helps update it in a prettier way.
closeAtMaximum	Functions like <code>switchboard_close()</code> by closing the switchboard window when the eavesdropped value equals maximum. NOTE: if a widget has <code>closeAtMaximum = TRUE</code> , then this widget MUST be placed at the end (i.e., last widget) of the pipe chain.

**Value**

Nothing.

**Usage**

```
progress_phyllotaxis(eavesdrop, maximum = 100, caption = "",
  size = 1, placeOnGrid = c(1, 1), updates = 100, delay = 0, honest = FALSE,
  closeAtMaximum = FALSE)
```

**See Also**

Other progress bars: [progress\\_benchmark\(\)](#), [progress\\_ibis\(\)](#), [progress\\_image\(\)](#), [progress\\_oyster\(\)](#), [progress\\_percent\(\)](#), [progress\\_pikachu\(\)](#), [progress\\_ring\\_percent\(\)](#), [progress\\_ring\(\)](#)

**Examples**

```
## Not run:

  for (i in 1:250) {
    switchboard(delay = 0.01) %>%
      progress_phyllotaxis(i, maximum = 250)
  }
  switchboard_close()

## End(Not run)
```

---

progress_pikachu	<i>Display pikachu as a progress bar.</i>
------------------	---

---

## Description

The progress\_pikachu widget displays pikachu as a progressBar. He's the best.

## Arguments

eavesdrop	The variable to track with a progress bar.
maximum	The maximum value of eavesdrop that marks the end of what to progress track.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number , column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1 , 1) position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).
honest	When TRUE, it updates the widget by the true progression value. The default (FALSE) has a cosmetic modification to the progression value that helps update it in a prettier way.
closeAtMaximum	Functions like switchboard_close() by closing the switchboard window when the eavesdropped value equals maximum. NOTE: if a widget has closeAtMaximum = TRUE, then this widget MUST be placed at the end (i.e., last widget) of the pipe chain.
fill	The direction of how things are animated when displaying progression. The default is horizontal, which tracks progression from left to right (maximum), but vertical can also be used for progress to occur in a bottom to up (maximum) animation.

## Value

Nothing.

## Usage

```
progress_pikachu(eavesdrop, maximum = 100, size = 1,
  placeOnGrid = c(1, 1), updates = 100, delay = 0, honest = FALSE,
  closeAtMaximum = FALSE, fill = "horizontal")
```

See Also

Other progress bars: [progress\\_benchmark\(\)](#), [progress\\_ibis\(\)](#), [progress\\_image\(\)](#), [progress\\_oyster\(\)](#), [progress\\_percent\(\)](#), [progress\\_phylloaxis\(\)](#), [progress\\_ring\\_percent\(\)](#), [progress\\_ring\(\)](#)

Examples

```
## Not run:

  for (i in 1:250) {
    switchboard(delay = 0.01) %>%
      progress_pikachu(i, maximum = 250)
  }
  switchboard_close()

## End(Not run)
```

---

progress_ring	<i>Display a progress ring.</i>
---------------	---------------------------------

---

Description

The progress\_ring widget displays an ring/circle progress bar.

Arguments

eavesdrop	The variable to track with a progress bar.
maximum	The maximum value of eavesdrop that marks the end of what to progress track.
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).



honest	When TRUE, it updates the widget by the true progression value. The default (FALSE) has a cosmetic modification to the progression value that helps update it in a prettier way.
closeAtMaximum	Functions like <code>switchboard_close()</code> by closing the switchboard window when the eavesdropped value equals maximum. NOTE: if a widget has <code>closeAtMaximum = TRUE</code> , then this widget MUST be placed at the end (i.e., last widget) of the pipe chain.

### Value

Nothing.

### Usage

```
progress_ring(eavesdrop, maximum = 100, caption = "",
  size = 1, placeOnGrid = c(1, 1), updates = 100, delay = 0, honest = FALSE,
  closeAtMaximum = FALSE)
```

### See Also

Other progress bars: [progress\\_benchmark\(\)](#), [progress\\_ibis\(\)](#), [progress\\_image\(\)](#), [progress\\_oyster\(\)](#), [progress\\_percent\(\)](#), [progress\\_phylloaxis\(\)](#), [progress\\_pikachu\(\)](#), [progress\\_ring\\_percent\(\)](#)

### Examples

```
## Not run:

  for (i in 1:250) {
    switchboard(delay = 0.01) %>%
      progress_ring(i, maximum = 250)
  }
  switchboard_close()

## End(Not run)
```

---

`progress_ring_percent` *Display a progress ring and a percent complete.*

---

### Description

The `progress_ring_percent` widget displays an ring/circle progress bar along with the percent complete.

**Arguments**

eavesdrop	The variable to track with a progress bar.
maximum	The maximum value of eavesdrop that marks the end of what to progress/track.
label	A small caption/label for the widget.
size	A number used to designate the size (magnification) of the widget. The default is set to 1 which is 80 by 80 pixels. For example, setting to 3 will results in a widget 3-times the default size (240 by 240 pixels) and will occupy a grid area of 3 by 3.
placeOnGrid	A row by column coordinate (e.g., c(row-number, column-number)) of a grid that designates the position to draw the widget on the switchboard. Use showGrid() to help organize widget placement on dashboard. The default places the first widget in pipe chain to the c(1, 1) position, and all following on the same row stacked to the right.
updates	The number of times the widget is to be updated (e.g., when it be modified/changed). The default updates the widget 100 times. Increase number for smoother progress bar animation.
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).
honest	When TRUE, it updates the widget by the true progression value. The default (FALSE) has a cosmetic modification to the progression value that helps update it in a prettier way.
closeAtMaximum	Functions like switchboard_close() by closing the switchboard window when the eavesdropped value equals maximum. NOTE: if a widget has closeAtMaximum = TRUE, then this widget MUST be placed at the end (i.e., last widget) of the pipe chain.

**Value**

Nothing.

**Usage**

```
progress_ring_percent(eavesdrop, maximum = 100, caption = "",
  size = 1, placeOnGrid = c(1, 1), updates = 100, delay = 0, honest = FALSE,
  closeAtMaximum = FALSE)
```

**See Also**

Other progress bars: [progress\\_benchmark\(\)](#), [progress\\_ibis\(\)](#), [progress\\_image\(\)](#), [progress\\_oyster\(\)](#), [progress\\_percent\(\)](#), [progress\\_phylloaxis\(\)](#), [progress\\_pikachu\(\)](#), [progress\\_ring\(\)](#)

**Examples**

```
## Not run:

for (i in 1:250) {
  switchboard(delay = 0.01) %>%
```

```

        progress_ring_percent(i, maximum = 250)
    }
    switchboard_close()

## End(Not run)

```

---

showGrid

*Display a layout grid on the switchboard.*


---

### Description

The showGrid widget displays a collection of buttons positioned along a grid to help devise where you would like to organize/place multiple widgets on a switchboard. Buttons contain the row by column coordinates of the grid. NOTE: You can click on any button to copy-to-clipboard the small coordinate script. You can then paste in your switchboard widget placeOnGrid option (e.g., placeOnGrid = c(1, 2)).

### Arguments

nrows	Number of rows to plot on grid.
ncolumns	Number of columns to plot on grid.

### Value

Nothing.

### Usage

```
showGrid(nrows = 4, ncolumns = 4)
```

### See Also

Other dashboard: [switchboard\\_close\(\)](#), [switchboard\(\)](#)

### Examples

```

## Not run:

for (i in 1:250) {
  switchboard(delay = 0.01) %>%
    progress_ring(i, maximum = 250, placeOnGrid = c(1,1)) %>%
    progress_ring(i, maximum = 250, placeOnGrid = c(2,2)) %>%
    progress_ring(i, maximum = 250, placeOnGrid = c(3,3)) %>%
    progress_ring(i, maximum = 250, placeOnGrid = c(4,4)) %>%
    showGrid()
}
switchboard_close()

```

```
## End(Not run)
```

---

switchboard

*Initialize a switchboard window*


---

## Description

The switchboard initializes a switchboard window, then using %>% you can pipe in various widgets into this window to create a dashboard.

## Usage

```
switchboard(
  title = "hidden title",
  delay = 0,
  skip = 0,
  padX = 10,
  padY = 0,
  font = "Helvetica",
  mainColors = c("#007fff", "ffffff", "#666666", "#cccccc")
)
```

## Arguments

title	The title of the switchboard Tcl/Tk window. Switchboards hide the titlebar, but is can be un-hidden by adding a title (e.g., title = "my new title").
delay	Pause each update of the switchboard. Default has no delay, values are in seconds (e.g., delay = 0.01 results in 0.01 second delay with each iteration).
skip	Skips updating the switchboard. The default does not skip updates, values are in number of iterations to skip (e.g., skip = 50 results in updates only occurring after every 50 iterations).
padX	Horizontal padding in pixels between widgets.
padY	Vertical padding in pixels between widgets.
font	The primary font used in all widgets, default is "Helvetica".
mainColors	The color set used in all widgets. It is a vector of four hex colors with each element corresponding to: c("main color", "background", "text color", "alt text color").

## See Also

Other dashboard: [showGrid\(\)](#), [switchboard\\_close\(\)](#)

**Examples**

```
## Not run:

for(i in 1:100)
  switchboard(delay = 0.05) %>%
    progress_ibus(i, maximum = 100, closeAtMaximum = TRUE)

## End(Not run)
```

---

switchboard_close	<i>Closes all switchboard windows.</i>
-------------------	--

---

**Description**

Helps remove all visible/invisible switchboard windows. Place outside of loop to remove windows.

**Usage**

```
switchboard_close()
```

**Value**

Nothing.

**See Also**

Other dashboard: [showGrid\(\)](#), [switchboard\(\)](#)

**Examples**

```
## Not run:

for(i in 1:100)
  switchboard(delay = 0.05) %>% percent(i, maximum = 100, label = "100 loops")
switchboard_close()

## End(Not run)
```

# Index

## \* caption widgets

caption, [4](#)

## \* dashboard

showGrid, [43](#)

switchboard, [44](#)

switchboard\_close, [45](#)

## \* eavesdroppers

benchmark, [3](#)

counter, [13](#)

counter\_tally, [14](#)

eavesdropper, [15](#)

eavesdropper\_2D, [17](#)

eavesdropper\_X, [18](#)

injector, [20](#)

injector\_2D, [22](#)

injector\_X, [24](#)

number, [25](#)

number\_pair, [26](#)

number\_quartet, [28](#)

number\_trio, [29](#)

progress\_benchmark, [30](#)

## \* injectors

control\_slider, [5](#)

control\_slider\_pair, [6](#)

control\_slider\_Y, [7](#)

control\_slider\_Y\_pair, [8](#)

control\_switch, [10](#)

control\_switch\_pair, [11](#)

control\_switch\_trio, [12](#)

injector, [20](#)

injector\_2D, [22](#)

injector\_X, [24](#)

## \* moving windows

eavesdropper, [15](#)

eavesdropper\_2D, [17](#)

eavesdropper\_X, [18](#)

injector, [20](#)

injector\_2D, [22](#)

injector\_X, [24](#)

## \* progress bars

progress\_benchmark, [30](#)

progress\_ibis, [31](#)

progress\_image, [33](#)

progress\_oyster, [34](#)

progress\_percent, [36](#)

progress\_phyllotaxis, [37](#)

progress\_pikachu, [39](#)

progress\_ring, [40](#)

progress\_ring\_percent, [41](#)

benchmark, [3](#), [14–16](#), [18](#), [19](#), [21](#), [23](#), [25–29](#), [31](#)

caption, [4](#)

control\_slider, [5](#), [7–11](#), [13](#), [21](#), [23](#), [25](#)

control\_slider\_pair, [6](#), [6](#), [8–11](#), [13](#), [21](#), [23](#), [25](#)

control\_slider\_Y, [6](#), [7](#), [7](#), [9–11](#), [13](#), [21](#), [23](#), [25](#)

control\_slider\_Y\_pair, [6–8](#), [8](#), [10](#), [11](#), [13](#), [21](#), [23](#), [25](#)

control\_switch, [6–9](#), [10](#), [11](#), [13](#), [21](#), [23](#), [25](#)

control\_switch\_pair, [6–10](#), [11](#), [13](#), [21](#), [23](#), [25](#)

control\_switch\_trio, [6–11](#), [12](#), [21](#), [23](#), [25](#)  
counter, [3](#), [13](#), [15](#), [16](#), [18](#), [19](#), [21](#), [23](#), [25–29](#), [31](#)

counter\_tally, [3](#), [14](#), [14](#), [16](#), [18](#), [19](#), [21](#), [23](#), [25–29](#), [31](#)

eavesdropper, [3](#), [14](#), [15](#), [15](#), [18](#), [19](#), [21](#), [23](#), [25–29](#), [31](#)

eavesdropper\_2D, [3](#), [14–16](#), [17](#), [19](#), [21](#), [23](#), [25–29](#), [31](#)

eavesdropper\_X, [3](#), [14–16](#), [18](#), [18](#), [21](#), [23](#), [25–29](#), [31](#)

injector, [3](#), [6–11](#), [13–16](#), [18](#), [19](#), [20](#), [23](#), [25–29](#), [31](#)

injector\_2D, [3](#), [6–11](#), [13–16](#), [18](#), [19](#), [21](#), [22](#), [25–29](#), [31](#)

injector\_X, [3](#), [6–11](#), [13–16](#), [18](#), [19](#), [21](#), [23](#),  
[24](#), [26–29](#), [31](#)

number, [3](#), [14–16](#), [18](#), [19](#), [21](#), [23](#), [25](#), [25](#),  
[27–29](#), [31](#)

number\_pair, [3](#), [14–16](#), [18](#), [19](#), [21](#), [23](#), [25](#), [26](#),  
[26](#), [28](#), [29](#), [31](#)

number\_quartet, [3](#), [14–16](#), [18](#), [19](#), [21](#), [23](#),  
[25–27](#), [28](#), [29](#), [31](#)

number\_trio, [3](#), [14–16](#), [18](#), [19](#), [21](#), [23](#), [25–28](#),  
[29](#), [31](#)

progress\_benchmark, [3](#), [14–16](#), [18](#), [19](#), [21](#),  
[23](#), [25–29](#), [30](#), [32](#), [34](#), [35](#), [37](#), [38](#),  
[40–42](#)

progress\_ibis, [31](#), [31](#), [34](#), [35](#), [37](#), [38](#), [40–42](#)

progress\_image, [31](#), [32](#), [33](#), [35](#), [37](#), [38](#), [40–42](#)

progress\_oyster, [31](#), [32](#), [34](#), [34](#), [37](#), [38](#),  
[40–42](#)

progress\_percent, [31](#), [32](#), [34](#), [35](#), [36](#), [38](#),  
[40–42](#)

progress\_phyllotaxis, [31](#), [32](#), [34](#), [35](#), [37](#),  
[37](#), [40–42](#)

progress\_pikachu, [31](#), [32](#), [34](#), [35](#), [37](#), [38](#), [39](#),  
[41](#), [42](#)

progress\_ring, [31](#), [32](#), [34](#), [35](#), [37](#), [38](#), [40](#), [40](#),  
[42](#)

progress\_ring\_percent, [31](#), [32](#), [34](#), [35](#), [37](#),  
[38](#), [40](#), [41](#), [41](#)

showGrid, [43](#), [44](#), [45](#)

switchboard, [43](#), [44](#), [45](#)

switchboard-package, [2](#)

switchboard\_close, [43](#), [44](#), [45](#)