

# Package ‘lcars’

November 16, 2020

**Title** LCARS Aesthetic for Shiny

**Version** 0.3.6

**Description** Provides Shiny widgets and theme that support a 'Library Computer Access/Retrieval System' (LCARS) aesthetic for Shiny apps.

The package also includes functions for adding a minimal LCARS theme to static 'ggplot2' graphs. More details about LCARS can be found at <<https://en.wikipedia.org/wiki/LCARS>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**URL** <https://github.com/leonawicz/lcars>

**BugReports** <https://github.com/leonawicz/lcars/issues>

**Depends** R (>= 3.5.0)

**Imports** grid, ggplot2, shiny, trekcolors

**Suggests** testthat, knitr, rmarkdown, covr, png, ggrepel, showtext, leaflet, leaflet.extras

**VignetteBuilder** knitr

**Language** en-US

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Matthew Leonawicz [aut, cre] (<<https://orcid.org/0000-0001-9452-2771>>)

**Maintainer** Matthew Leonawicz <[matt\\_leonawicz@esource.com](mailto:matt_leonawicz@esource.com)>

**Repository** CRAN

**Date/Publication** 2020-11-16 08:00:10 UTC

## R topics documented:

inputColumn . . . . .	2
lcars . . . . .	3
lcarsApp . . . . .	3

lcarsBox . . . . .	4
lcarsBracket . . . . .	7
lcarsButton . . . . .	8
lcarsCheckbox . . . . .	9
lcarsdata . . . . .	11
lcarsHeader . . . . .	11
lcarsOutput . . . . .	13
lcarsPage . . . . .	14
lcarsRadio . . . . .	16
lcarsRect . . . . .	18
lcarsSweep . . . . .	20
lcarsToggle . . . . .	22
lcarsWell . . . . .	24
lcars_border . . . . .	25
lcars_elbow . . . . .	27
lcars_pill . . . . .	28
lcars_rect . . . . .	30
renderLcars . . . . .	30
theme_lcars . . . . .	31

<b>Index</b>	<b>33</b>
--------------	-----------

---

inputColumn	<i>LCARS input column</i>
-------------	---------------------------

---

## Description

An LCARS input column is a container for inputs like `lcarsButton` and `lcarsToggle` that can be passed to `lcarsBox`. The inputs occur vertically in the left or right side panel of the box. To fit properly, input widths should be 150px or less, matching the widths of the side panels of the given `lcarsBox` container.

## Usage

```
inputColumn(...)
```

## Arguments

... div contents such as `lcarsButton` elements.

## Value

HTML

## Examples

```
inputColumn()
```

---

lcars	<i>lcars: LCARS aesthetic for Shiny</i>
-------	---

---

## Description

Library Computer Access/Retrieval System (**LCARS**).

## Details

The `lcars` package provides Shiny widgets and Shiny UI theme that support a 'Library Computer Access/Retrieval System' (LCARS) aesthetic for Shiny apps. The package also includes functions for adding a minimal LCARS theme to static `ggplot2` graphs.

---

<code>lcarsApp</code>	<i>Launch LCARS demo apps.</i>
-----------------------	--------------------------------

---

## Description

Currently available apps include: `demo`, `box`, `sweep`, `toggle`, `elements`, `leaflet`.

## Usage

```
lcarsApp(id = "demo")
```

## Arguments

<code>id</code>	character, app id.
-----------------	--------------------

## Examples

```
if (interactive()) {  
  lcarsApp("demo")  
}
```

---

 lcarsBox

*LCARS box*


---

### Description

Create a configurable LCARS box.

### Usage

```
lcarsBox(
  ...,
  title = NULL,
  subtitle = NULL,
  corners = c(1, 4),
  sides = c(1, 3, 4),
  left_inputs = NULL,
  right_inputs = NULL,
  color = "atomic-tangerine",
  side_color = color,
  title_color = color,
  subtitle_color = color,
  title_right = TRUE,
  subtitle_right = TRUE,
  clip = TRUE,
  width_left = 150,
  width_right = 150,
  width = "100%"
)
```

### Arguments

...	box contents.
title	character, box title at top right.
subtitle	character, box subtitle at bottom right.
corners	integer, 1:4, a vector specifying which corner elbows to include: top left, top right, bottom right, bottom left. See details.
sides	integer, 1:4, a vector specifying which sides to include panels: top, right, bottom, left. See details.
left_inputs	optional input column for left side, for example a column of buttons made with <code>inputColumn</code> . See details.
right_inputs	optional input column for right side, for example a column of buttons made with <code>inputColumn</code> . See details.
color	box border colors. See details.
side_color	box border colors. See details.

<code>title_color</code>	text title color.
<code>subtitle_color</code>	text subtitle color.
<code>title_right</code>	logical, right align title.
<code>subtitle_right</code>	logical, right align subtitle.
<code>clip</code>	logical, use empty margin space. See details.
<code>width_left</code>	numeric, the width of the left side panel in pixels. This also adjusts associated corner elbows to match. Defaults to the maximum allowed: 150.
<code>width_right</code>	numeric, the width of the right side panel in pixels. This also adjusts associated corner elbows to match. Defaults to the maximum allowed: 150.
<code>width</code>	a valid CSS unit, the width of the entire box. Fixed pixel width recommended. See details.

### Details

This function allows you to customize the inclusion and colors of specific border components of the box. The defaults are closer to standard LCARS style. You can turn on or off specific corner elbows, connecting side panels, control colors of each, as well as title and subtitle inclusion, color and alignment.

### Value

an HTML widget

### Corner elbows

Control which corners of the box display the characteristic LCARS elbow, clockwise from top left. The top and bottom borders are independent of one another. Each work in the same manner. For each, you can have a left elbow (default), a right elbow, or both.

When only one corner is present (on top or on bottom), the bar extends to the other corner and terminates with the characteristic LCARS half pill if the panel border is included (see side panel section below). If the side between the elbow areas is excluded, only the elbows are displayed.

If both elbows are excluded from the top or from the bottom, a simple, straight `IcarsHeader` element is placed above or below the main content area instead, but this can be controlled via `sides`.

### Side panels

Control which sides of the box include an LCARS-styled border, clockwise from top left. Sides connect elbows using straight bars. The top and bottom sides are where title and subtitle text are placed. The title for the top and subtitle for the bottom are included in the bar with standard LCARS right alignment, which can be switched to left. If the top or bottom side panel is excluded, the vertical space remains if `title` or `subtitle` are included, respectively, retaining the text labels; otherwise the space is removed.

By default, left and right sides are 150 pixels wide; top and bottom sides are 30 pixels tall. The top and bottom are fixed, but the widths of the left and right side panels can be adjusted using `width_left` and `width_right`, respectively. They can only be adjusted down to smaller widths. This is to ensure proper scaling for connected corners. The side panels are not meant to accommodate wider inputs and should primarily be used for small buttons and short text.

### Side inputs columns

Input columns are different from left and right sides. The latter refers to whether or not there are vertical connecting bars from elbow to elbow. An input column represents a separate element that is placed in the left or right side panel area above the plain side panel bar itself.

If the side is included and a column of inputs is provided, they combine vertically to form the side panel. Some amount of plain sidebar will pad the bottom beneath any input column, however tall. If the side is excluded, the input column will take up the entire vertical space.

If the side is excluded and no input column is provided, the side panel area is blank. The main content area extends left or right to fill any completely missing left or right side panel. To restrict this, use a black side panel to match the background.

Since the inputs contained in an input column are defined separately and passed to `lcarsBox`, they should be defined to have widths that match the box side panel widths.

### Colors

Box color can be any color given in hex format. Named colors must be LCARS colors. See [lcarsdata](#) for options. By default, all border colors inherit from a single color passed to `color`.

`color` is recycled to length four as needed. `color` actually defines all four corner elbow colors. For corner elbows, use a vector of four colors for the top left, top right, bottom right, and bottom left, respectively.

Similarly for the bars between elbows with `side_colors`, use a vector of four colors for the top, right side, bottom, and left side. This is also recycled to length four. If not provided, it inherits from `color`.

`title_color` and `subtitle_color` are scalar. They inherit from the first color in `color`.

### Margin space

When at least one corner elbow is present on a top or bottom side, that side will include empty margin space to the inside of the elbow. This space is part of the grid area of the side panel. This is why main panel content does not extend into it. You can override this and make use of this space by setting `clip = FALSE`.

Note that this should only be done when both side panels are present so that the main panel content is not directly under or above the elbow near the extreme edge of the box. If you do not want a side panel, you can include it, but set its color to match the background.

### Sizing

There are limitations to the container responsiveness of the LCARS box and sweep. In some cases, using percentage width, e.g., `width = "100%"` will work, but it may respond sluggishly or may not work at all. Fixed pixel width is recommended for `lcarsBox` and `lcarsSweep`. Regardless of responsiveness, these widgets are also not intended to fit very small displays.

### See Also

[lcarsSweep](#)

## Examples

```
## Only run examples in interactive R sessions
if (interactive()) {

  ui <- lcarsPage(
    lcarsBox(
      fluidRow(
        column(3,
          h4("Main panel area"),
          HTML("<p>Some paragraph text and <a href='#'>a link</a>
with LCARS styling.</p>
<p>Use <code>lcarsPage</code>
to apply the LCARS theme and <code>lcarsBox</code>
to draw a characteristic box for framing content.</p>
<p>Many of the <code>lcarsBox</code>
properties are configurable.
See <code>lcars::lcarsApp(\"box\")</code> for a demo</p>")
        ),
        column(9, plotOutput("plot1"))
      ),
      title = "box title",
      left_inputs = inputColumn(lcarsButton("btn1", "A button"))
    )
  )

  server <- function(input, output) {
    output$plot1 <- renderPlot({
      hist(rnorm(500))
    })
  }

  shinyApp(ui, server)
}
```

---

 lcarsBracket

*LCARS bracket element*


---

## Description

A top and bottom bracket pair element used to visually group contents.

## Usage

```
lcarsBracket(
  ...,
  color = "golden-tanoi",
  background_color = "#000000",
  hollow = TRUE,
  width = "100%"
)
```

**Arguments**

...	div contents.
color	bracket color. Any hex color or a named LCARS color.
background_color	background color. Any hex color or a named LCARS color. Applies when hollow = TRUE.
hollow	logical, use a hollow bracket. The cutout section has background_color.
width	a valid CSS unit.

**Value**

an HTML widget

**Examples**

```
## Only run examples in interactive R sessions
if (interactive()) {

  ui <- lcarsPage(
    fluidRow(
      column(4,
        h4("Hollow bracket"),
        lcarsBracket(
          lcarsRect("Some text.", text_size = 24, height = 40)
        ),
        h4("Solid bracket"),
        lcarsBracket(
          lcarsRect("Some text.", color = "#000000",
            text_color = "golden-tanai",
            text_size = 24, height = 40),
          hollow = FALSE
        )
      )
    )
  )

  server <- function(input, output) {}

  shinyApp(ui, server)
}
```

---

 lcarsButton

*LCARS button*


---

**Description**

An LCARS wrapper around `shiny::actionButton` with additional color control.



**Usage**

```

IcarsButton(
    inputId,
    label,
    icon = NULL,
    color = "atomic-tangerine",
    hover_color = "red-damask",
    ...
)

```

**Arguments**

<code>inputId</code>	The input slot that will be used to access the value.
<code>label</code>	The contents of the button or link—usually a text label, but you could also use any other HTML, like an image.
<code>icon</code>	An optional icon to appear on the button.
<code>color</code>	button color. Can be any color given in hex format. Named colors must be LCARS colors. See <a href="#">Icarsdata</a> for options.
<code>hover_color</code>	Named colors must be LCARS colors. Other arbitrary colors are not supported for hovering. If <code>hover_color = NULL</code> , the button will darken on hover.
<code>...</code>	Named attributes to be applied to the button.

**Value**

HTML

**See Also**

[Icarsdata](#)

**Examples**

```
IcarsButton("btn", "A button")
```

---

IcarsCheckbox

*LCARS checkbox*

---

**Description**

An LCARS styled toggle button that can be used in place of `checkboxInput`.

## Usage

```
lcarsCheckbox(  
  inputId,  
  label,  
  value = FALSE,  
  color = "atomic-tangerine",  
  background_color = "#000000",  
  label_color = "#FFFFFF",  
  label_right = FALSE,  
  width = NULL  
)
```

## Arguments

<code>inputId</code>	character, the input slot that will be used to access the value.
<code>label</code>	character, display label for the control, or NULL for no label.
<code>value</code>	logical, initial value.
<code>color</code>	Check color. Can be any color given in hex format. Named colors must be LCARS colors. See <a href="#">lcarsdata</a> for options.
<code>background_color</code>	background color, as above.
<code>label_color</code>	label text color, as above.
<code>label_right</code>	logical, set to TRUE to right align the label.
<code>width</code>	a valid CSS unit.

## Value

A checkbox control that can be added to a UI definition

## Examples

```
## Only run examples in interactive R sessions  
if(interactive()){  
  ui <- lcarsPage(  
    lcarsCheckbox("somevalue", "Some value", FALSE),  
    verbatimTextOutput("value")  
  )  
  server <- function(input, output) {  
    output$value <- renderText({ input$somevalue })  
  }  
  shinyApp(ui, server)  
}
```

---

lcarsdata	<i>LCARS colors</i>
-----------	---------------------

---

**Description**

A data frame with 33 rows and 3 columns containing color names and values for each of four palette series.

**Usage**

```
lcarsdata
```

**Format**

A data frame

---

lcarsHeader	<i>LCARS header</i>
-------------	---------------------

---

**Description**

An LCARS header panel.

**Usage**

```
lcarsHeader(
  title = NULL,
  color = "golden-tanoi",
  title_color = color,
  background_color = "#000000",
  title_right = TRUE,
  title_invert = FALSE,
  round = c("both", "right", "left", "none"),
  width = "100%"
)

lcarsh1(
  title = NULL,
  color = "atomic-tangerine",
  title_color = color,
  background_color = "#000000",
  title_align = c("center", "right", "left"),
  title_invert = FALSE,
  width = "auto"
)
```

```
lcarsh2(  
  title = NULL,  
  color = "atomic-tangerine",  
  title_color = color,  
  background_color = "#000000",  
  title_align = c("center", "right", "left"),  
  title_invert = FALSE,  
  width = "auto"  
)  
  
lcarsh3(  
  title = NULL,  
  color = "atomic-tangerine",  
  title_color = color,  
  background_color = "#000000",  
  title_align = c("center", "right", "left"),  
  title_invert = FALSE,  
  width = "auto"  
)  
  
lcarsh4(  
  title = NULL,  
  color = "atomic-tangerine",  
  title_color = color,  
  background_color = "#000000",  
  title_align = c("center", "right", "left"),  
  title_invert = FALSE,  
  width = "auto"  
)  
  
lcarsh5(  
  title = NULL,  
  color = "atomic-tangerine",  
  title_color = color,  
  background_color = "#000000",  
  title_align = c("center", "right", "left"),  
  title_invert = FALSE,  
  width = "auto"  
)  
  
lcarsh6(  
  title = NULL,  
  color = "atomic-tangerine",  
  title_color = color,  
  background_color = "#000000",  
  title_align = c("center", "right", "left"),  
  title_invert = FALSE,  
  width = "auto"
```

)

**Arguments**

title	character, optional title.
color	header color. Any hex color or a named LCARS color.
title_color	text color. Any hex color or a named LCARS color.
background_color	color behind text.
title_right	logical, right align title.
title_invert	logical, invert the color and background color for the title rectangle.
round	character, sides of header to round. The default is to present the header in LCARS full pill style.
width	a valid CSS unit.
title_align	character, for the heading replacers: center, right or left.

**Details**

In addition to `lcarsHeader` there are also some LCARS style heading replacers, `lcarsh1` through `lcarsh6`. These default to centered text, whereas `lcarsHeader` is strictly right or left.

**Value**

HTML

**See Also**[lcarsdata](#).**Examples**

```
lcarsHeader("A title")
```

---

IcarsOutput

*Create an LCARS output (client side)*


---

**Description**

UI-side functions for creating dynamic `lcarBox` and `lcarsSweep`.

**Usage**

```
lcarsBoxOutput(outputId)
```

```
lcarsSweepOutput(outputId)
```

**Arguments**

outputId          Output variable name.

**See Also**

[renderLcars](#) for the corresponding server-side function.

**Examples**

```
## Only run examples in interactive R sessions
if (interactive()) {

  ui <- lcarsPage(
    lcarsBoxOutput("box"),
    lcarsSweepOutput("sweep")
  )

  server <- function(input, output) {
    output$box <- renderLcarsBox({
      lcarsBox()
    })
    output$sweep <- renderLcarsSweep({
      lcarsSweep()
    })
  }

  shinyApp(ui, server)
}
```

---

lcarsPage

*LCARS Shiny UI*

---

**Description**

Create a Shiny UI page with an LCARS theme.

**Usage**

```
lcarsPage(
  ...,
  title = NULL,
  force_uppercase = TRUE,
  label_uppercase = TRUE,
  lcars_font_headers = TRUE,
  lcars_font_labels = TRUE,
  lcars_font_text = TRUE
)
```

**Arguments**

...	The contents of the document body.
title	The browser window title (defaults to the host URL of the page).
force_uppercase	logical, see details.
label_uppercase	logical, also make widget labels uppercase globally.
lcars_font_headers	use LCARS-style font family for major heading text. See details.
lcars_font_labels	use LCARS-style font family for LCARS widget labels/titles.
lcars_font_text	use LCARS-style font family for general text (paragraphs, lists, etc.).

**Details**

The LCARS style heavily emphasizes uppercase text. Set `force_uppercase = TRUE` to force this standard via CSS. This does not make everything uppercase; things like input labels are left alone (use `label_uppercase = TRUE`). However, text in general is uppercased.

Set these to `FALSE` if you need control over casing. This allows sensible judgment over how to balance the tension between making something that conforms well to the familiar LCARS aesthetic and making something that communicates information with a lower cognitive load for the user. Similarly, set `lcars_font*` arguments to `FALSE` to use a more readable sans serif font as desired. See examples for recommendations.

**Value**

A UI definition that can be passed to the `shinyUI` function.

**Examples**

```
# Recommended settings
# (results stored in x and not printed due to length)

# for a more standard LCARS style: default settings.
x <- lcarsPage()

# for a more readable style: less uppercase, switch to sans font
x <- lcarsPage(force_uppercase = FALSE, label_uppercase = FALSE,
  lcars_font_labels = FALSE, lcars_font_text = FALSE)
```

---

 lcarsRadio

*LCARS radio buttons*


---

### Description

LCARS-styled radio buttons functions.

### Usage

```
lcarsRadio(
  inputId,
  label,
  choices = NULL,
  selected = NULL,
  inline = FALSE,
  width = NULL,
  choiceNames = NULL,
  choiceValues = NULL,
  label_color = "#FFFFFF",
  choice_color = label_color
)
```

```
lcarsRadioToggle(
  inputId,
  label,
  choices = NULL,
  selected = NULL,
  width = NULL,
  choiceNames = NULL,
  choiceValues = NULL,
  label_color = "atomic-tangerine",
  choice_color = "#000000",
  background_color = label_color,
  checked_color = choice_color,
  checked_background = "pale-canary"
)
```

### Arguments

inputId	character, the input slot that will be used to access the value.
label	character, display label for the control, or NULL for no label.
choices	see shiny::radioButtons for details.
selected	The initially selected value (if not specified then defaults to the first value).
inline	If TRUE, render the choices inline (i.e. horizontally).
width	a valid CSS unit.



choiceNames, choiceValues  
 see `shiny::radioButtons` for details.

label\_color, choice\_color, background\_color, checked\_color, checked\_background,  
 Color for the label, choices text, choices background, checked text and checked background. Can be any color given in hex format. Named colors must be LCARS colors. See [lcarsdata](#) for options.

## Details

`lcarsRadio` is a minimal replacement for `radioButtons` that provides two additional color arguments for consistency with functions like `lcarsCheckbox`. `lcarsRadioToggle` is a more customized toggle style radio buttons wrapper with more color controls.

## Value

A set of radio buttons that can be added to a UI definition.

## Examples

```
## Only run examples in interactive R sessions
if (interactive()) {

  ui <- lcarsPage(
    fluidRow(
      column(6,
        lcarsRadio("dist1", "Distribution type:",
          c("Normal" = "norm",
            "Uniform" = "unif",
            "Log-normal" = "lnorm",
            "Exponential" = "exp"),
          inline = TRUE,
          label_color = "lilac",
          choice_color = "atomic-tangerine"
        ),
        plotOutput("distPlot1")
      ),
      column(6,
        lcarsRadioToggle("dist2", "Distribution type:",
          c("Normal" = "norm",
            "Uniform" = "unif",
            "Log-normal" = "lnorm",
            "Exponential" = "exp"),
          width = "100%"
        ),
        plotOutput("distPlot2")
      )
    )
  )

  server <- function(input, output) {
    output$distPlot1 <- renderPlot({
      dist <- switch(input$dist1,
```

```

        norm = rnorm,
        unif = runif,
        lnorm = rlnorm,
        exp = rexp,
        rnorm)
    hist(dist(500))
  })
  output$distPlot2 <- renderPlot({
    dist <- switch(input$dist2,
      norm = rnorm,
      unif = runif,
      lnorm = rlnorm,
      exp = rexp,
      rnorm)
    hist(dist(500))
  })
}

shinyApp(ui, server)
}

```

---

IcarsRect

*LCARS rectangle element*


---

### Description

A basic rectangle HTML element that conforms to LCARS specifications.

### Usage

```

IcarsRect(
  text = "",
  round = c("none", "both", "left", "right"),
  decorate = c("none", "both", "left", "right"),
  color = "golden-tanoi",
  text_color = "#000000",
  title_color = color,
  text_size = 16,
  title = NULL,
  width = "100%",
  height = 30
)

IcarsPill(
  text = "",
  decorate = c("none", "both", "left", "right"),
  color = "golden-tanoi",
  text_color = "#000000",
  title_color = color,

```

```

    text_size = 16,
    title = NULL,
    width = "100%",
    height = 30
)

IcarsLeftPill(
    text = "",
    decorate = FALSE,
    color = "golden-tanoi",
    text_color = "#000000",
    title_color = color,
    text_size = 16,
    title = NULL,
    width = "100%",
    height = 30
)

IcarsRightPill(
    text = "",
    decorate = FALSE,
    color = "golden-tanoi",
    text_color = "#000000",
    title_color = color,
    text_size = 16,
    title = NULL,
    width = "100%",
    height = 30
)

```

### Arguments

<code>text</code>	character, rectangle text.
<code>round</code>	character, sides of rectangle to round to make an LCARS pill or half pill.
<code>decorate</code>	character, sides of rectangle to decorate with cut pill; applicable if a given side is rounded via <code>round</code> . Logical for <code>IcarsLeftPill</code> and <code>IcarsRightPill</code> .
<code>color</code>	rectangle color. Any hex color or a named LCARS color.
<code>text_color</code>	text color. Any hex color or a named LCARS color.
<code>title_color</code>	title color. Any hex color or a named LCARS color.
<code>text_size</code>	size of text in pixels.
<code>title</code>	optional title text to insert in blank gap in rectangle. Used for header-style rectangles.
<code>width</code>	a valid CSS unit.
<code>height</code>	a valid CSS unit.

**Details**

While text can be made arbitrarily large using `text_size`, the font size of the optional title is fixed at standard header size (`height = 0.5`).

**Value**

a div

**Examples**

```
if (interactive()) {
  ui <- lcarsPage(
    fluidRow(
      column(4,
        h4("Rectangle"),
        lcarsRect("Some text.", text_size = 24, width = 200),
        h4("Pill"),
        lcarsPill("Some text.", text_size = 24, width = 200)
      )
    )
  )
  server <- function(input, output) {}
  shinyApp(ui, server)
}
```

---

 lcarsSweep

*LCARS sweep*


---

**Description**

Create an LCARS sweep; the 'S' or reverse-'S' shape comprised of two LCARS elbows pointing in opposite directions. The sweep is effectively two adjacent LCARS boxes separated by an input column and some specific styling to achieve the sweep display.

**Usage**

```
lcarsSweep(
  column_inputs = NULL,
  left_inputs = NULL,
  right_inputs = NULL,
  title = NULL,
  subtitle = NULL,
  color = "atomic-tangerine",
  reverse = FALSE,
  expand = c(0, 0),
```

```

    column_width = 150,
    left_width = 0.5,
    width = "100%"
  )

```

### Arguments

<code>column_inputs</code>	optional input column for right side, for example a column of buttons made with <code>inputColumn</code> . See details.
<code>left_inputs</code>	content on the left side of the sweep.
<code>right_inputs</code>	content on the right side of the sweep.
<code>title</code>	character, title for box with header.
<code>subtitle</code>	character, subtitle for box with footer.
<code>color</code>	sweep elbow colors. Any hex color or a named LCARS color.
<code>reverse</code>	logical, create a reverse sweep.
<code>expand</code>	integer, length-2 vector, the number of pixels to expand the left and right content containers above or below the implicit border; the top or bottom border where no sweep is present. See example.
<code>column_width</code>	integer, width of the sweep column section in pixels. Must be in pixels, 150 maximum. Smaller is permitted but will not conform as well to LCARS style.
<code>left_width</code>	numeric, number between 0 and 1 giving the proportional width of the left content section. The right section is $1 - \text{left\_width}$ .
<code>width</code>	a valid CSS unit, the width of the entire sweep. Fixed pixel width recommended. See details.

### Details

There are limitations to the container responsiveness of the LCARS box and sweep. In some cases, using percentage width, e.g., `width = "100%"` will work, but it may respond sluggishly or may not work at all. Fixed pixel width is recommended for `lcarsBox` and `lcarsSweep`. Regardless of responsiveness, these widgets are also not intended to fit very small displays.

### Value

an HTML widget

### See Also

[lcarsBox](#)

### Examples

```

## Only run examples in interactive R sessions
if (interactive()) {

  library(ggplot2)
  d <- data.frame(x = rnorm(500))

```

```

g <- ggplot(d, aes(x)) + theme_lcars_dark()
g1 <- g + geom_histogram(color = "black", fill = "#9999FF", bins = 20) +
  ggtitle("Plot 1")
left <- div(h4("Some text"), p("The fine print.))

ui <- lcarsPage(
  lcarsHeader("LCARS sweep"),
  h4("Change colors and relative widths of content sections"),
  h4("Add title and subtitle, input column padding, and content"),
  lcarsSweep(
    inputColumn(
      lcarsButton("x1", "Button"),
      lcarsRect(color = "hopbush", height = 80)
    ),
    left, plotOutput("plot1", height = 650), # plot taller than sweep box
    title = "Title", subtitle = "Subtitle",
    color = "pale-canary", left_width = 0.3, width = 900,
    expand = c(0, 350) # negative bottom margin added to right side div
  ),
  lcarsSweep( # content from sweep box above extends into sweep box below
    inputColumn(
      lcarsButton("x2", "Button A"),
      lcarsButton("x3", "Button B"),
      lcarsRect(color = "lilac")
    ),
    left, title = "Title 2", subtitle = "Subtitle 2",
    color = "anakiwa", reverse = TRUE, left_width = 0.3, width = 900
  )
)

server <- function(input, output) {
  output$plot1 <- renderPlot(g1)
}

shinyApp(ui, server)
}

```

---

IcarsToggle

*LCARS toggle button*


---

### Description

An LCARS styled toggle button that can be used in place of checkboxInput and lcarsCheckbox.

### Usage

```

IcarsToggle(
  inputId,
  label,
  value = FALSE,

```

```

    pill = FALSE,
    inverse = FALSE,
    true = "Yes",
    false = "No",
    true_color = "dodger-pale",
    false_color = "atomic-tangerine",
    background_color = "#000000",
    border_color = ifelse(inverse, false_color, background_color),
    outer_border = FALSE,
    outer_color = "#000000",
    label_color = "#FFFFFF",
    label_right = FALSE,
    width = NULL
)

```

### Arguments

inputId	character, the input slot that will be used to access the value.
label	character, display label for the control, or NULL for no label.
value	logical, initial value.
pill	logical, use an LCARS pill style with rounded ends instead of the default rounded rectangle.
inverse	logical, invert the color presentation.
true	character, text label for TRUE position.
false	character, text label for FALSE position.
true_color	Color for TRUE position. Can be any color given in hex format. Named colors must be LCARS colors. See <a href="#">lcarsdata</a> for options.
false_color	Color for FALSE position, as above.
background_color	background color, as above.
border_color	border color, as above.
outer_border	logical, use outer border. This makes some adjustments to inner elements if used.
outer_color	outer border color, as above.
label_color	label text color, as above.
label_right	logical, set to TRUE to right align label text.
width	character, use only px units for this widget, e.g. "150px" (the default when NULL). Percentage is the only other unit allowed. It works, but not as well. Fixed widths recommended.

### Value

A toggle button control that can be added to a UI definition.

**Examples**

```
## Only run examples in interactive R sessions
if(interactive()){
  ui <- lcarsPage(
    lcarsToggle("somevalue", "Some value", FALSE),
    verbatimTextOutput("value")
  )
  server <- function(input, output) {
    output$value <- renderText({ input$somevalue })
  }
  shinyApp(ui, server)
}
```

---

 lcarsWell

*LCARS well*


---

**Description**

A simple LCARS well panel wrapper that takes color and background color arguments and understands LCARS color names.

**Usage**

```
lcarsWell(..., color = "atomic-tangerine", background_color = "#000000")
```

**Arguments**

... panel contents.

color border color. Any hex color or a named LCARS color.

background\_color background color. Any hex color or a named LCARS color.

**Value**

HTML

**Examples**

```
lcarsWell()
```



---

lcars_border	<i>LCARS border plot</i>
--------------	--------------------------

---

### Description

Wrap a ggplot object with an LCARS-themed border or only plot the border.

### Usage

```
lcars_border(
  x = NULL,
  width = 10,
  height = 6,
  corners = 1:4,
  length_frac = rep(0.5, 8),
  corner_color = rep("atomic-tangerine", 4),
  ro = width/20,
  ri = height/60,
  side_width = c(1, 2, 1, 5)/5,
  side_n_segments = rep(0, 4),
  side_color = as.list(rep("atomic-tangerine", 4)),
  side_label = rep(NA, 4),
  label_size = 1,
  side_label_adj = list(c(0.5, 0.5), c(-0.2, -0.2), c(0.5, 0.5), c(1.1, -0.2)),
  gap = c(0.02, 0.01),
  bg = "black",
  n = 20
)
```

### Arguments

<code>x</code>	optional inset ggplot object.
<code>width</code>	full plot width in inches.
<code>height</code>	full plot height in inches.
<code>corners</code>	integer, 1:4, a vector specifying which corner elbows to include the LCARS elbow bend: top left, top right, bottom right, bottom left.
<code>length_frac</code>	numeric, the fraction of a side that a corner extends over. See details.
<code>corner_color</code>	vector of corner colors, clockwise from top left. Can be any color given in hex format. Named colors must be LCARS colors. See <a href="#">lcarsdata</a> for options.
<code>ro</code>	vector of corner outer radii, clockwise from top left.
<code>ri</code>	vector of inner outer radii, clockwise from top left.
<code>side_width</code>	width of each side, clockwise from top left.
<code>side_n_segments</code>	for each side clockwise from top left, the number of rectangle segments used to evenly fill the space between corner bends.

side_color	list of color vectors for side segments. Each vector must have the same number of colors as the number of segments for a given side. Can be any color given in hex format. Named colors must be LCARS colors. See <a href="#">lcarsdata</a> for options.
side_label	list of label vectors for side segments. Each vector must have the same number of colors as the number of segments for a given side.
label_size	numeric, global label text size.
side_label_adj	list of four vectors, each giving the adj argument to text for each side.
gap	vector of two values giving the gap fraction from 0 to 1, based on full plot width and height, for the gap between horizontal and vertical segments, respectively.
bg	background color, should be left black for LCARS standard.
n	integer, number of points used to define inner radii quarter circles for corner bends.

### Details

This function draws a plot. It does not return a new ggplot object.

For `length_frac`, a vector of eight values from 0 to 1 is required. Starting from the top side, clockwise around to the left side, they refer to fraction of that side's length over which the relevant corner bend extends. For example, the first value refers to the top left corner bend's rightward horizontal segment. The second value refers to the top right corner bend's leftward horizontal segment. This takes care of the top side. Finally, the last value refers to the downward vertical arm of the top left corner bend.

All arguments that take vectors or lists of length four are in clockwise order from either the top left corner for corner-related arguments or the top side for side-related arguments. Colors may be given as official LCARS color names.

### Value

draws a plot

### Examples

```
lcars_border()

sw <- seq(0.2, 2, length = 4)
lcars_border(width = 5, height = 5, ro = sw, ri = sw / 2, side_width = sw)

len_frac <- c(0.3, 0.5, 0.2, 0.4, 0.3, 0.2, 0.1, 0.3)
n_seg <- c(1, 2, 0, 8)

library(ggplot2)
g <- ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
  geom_point() + facet_wrap(~Species, 2) + theme_lcars_light()
lcars_border(g, corners = 1:3, length_frac = len_frac, side_n_segments = n_seg)
```

---

lcars_elbow	<i>LCARS corner elbow</i>
-------------	---------------------------

---

### Description

Draw a, LCARS elbow polygon. This is a 90-degree rounded corner bend for top left, top right, bottom right and bottom left LCARS corner panels.

### Usage

```
lcars_elbow(
  xmin,
  xmax,
  ymin,
  ymax,
  corner,
  width,
  height,
  ro = width/2,
  ri = height/2,
  n = 20,
  color = "atomic-tangerine",
  draw = TRUE
)
```

### Arguments

xmin	numeric, scalar left x position.
xmax	numeric, scalar right x position.
ymin	numeric, scalar bottom y position.
ymax	numeric, scalar top y position.
corner	integer 1:4 or character: topleft, topright, bottomleft, bottomright. May be abbreviated as tl, tr, br, bl.
width	numeric, the width of the vertical segment of the bend.
height	numeric, the height of the horizontal segment of the bend.
ro	radius of the outer rounded corner.
ri	radius of the inner rounded corner.
n	number of points to define the curve of the inner radial quarter circle. The number of points then used to define the outer curve and extensions of the segments are scaled respectively based on this.
color	ignored if draw = FALSE. Can be any color given in hex format. Named colors must be LCARS colors. See <a href="#">lcarsdata</a> for options.
draw	draw the corner. Return values if FALSE.

**Value**

draws a polygon

**Examples**

```
plot(0:1, 0:1)
lcars_elbow(0.1, 0.9, 0.6, 0.9, "t1", 0.2, 0.05)
```

---

 lcars\_pill

*LCARS pill*


---

**Description**

Wrappers around `lcars_rect` that add rounded edges on one side or two opposing sides to make an LCARS pill.

**Usage**

```
lcars_pill(
  xmin,
  xmax,
  ymin,
  ymax,
  color = "atomic-tangerine",
  direction = c("both", "left", "right"),
  vertical = FALSE,
  gap = "auto",
  n = 50,
  asp = 1,
  gap_color = "#000000"
)
```

```
lcars_half_pill(
  x,
  y,
  r,
  direction,
  color = "atomic-tangerine",
  n = 50,
  asp = 1
)
```

```
lcars_left_pill(x, y, r, color = "atomic-tangerine", n = 50, asp = 1)
```

```
lcars_right_pill(x, y, r, color = "atomic-tangerine", n = 50, asp = 1)
```

```
lcars_bottom_pill(x, y, r, color = "atomic-tangerine", n = 50, asp = 1)
```

```
lcars_top_pill(x, y, r, color = "atomic-tangerine", n = 50, asp = 1)
```

### Arguments

xmin	numeric, scalar left x position.
xmax	numeric, scalar right x position.
ymin	numeric, scalar bottom y position.
ymax	numeric, scalar top y position.
color	pill color. Can be any color given in hex format. Named colors must be LCARS colors. See <a href="#">lcarsdata</a> for options.
direction	integer 1:4 or character: topleft, topright, bottomleft, bottomright. May be abbreviated as tl, tr, br, bl.
vertical	logical, vertical pill.
gap	numeric or "auto", the gap between the pill half circle edge and pill rectangle edge.
n	integer, number of points to define rounded edge.
asp	numeric, aspect ratio. This is useful for preventing distortion of pill half circle for plots with different width and height.
gap_color	the color of gaps if present. This is likely black, but because of the way the pill is drawn, it must be specified to match if the plot background color is not black. Can be any color given in hex format. Named colors must be LCARS colors. See <a href="#">lcarsdata</a> for options.
x	numeric, x position for edge of horizontal half pill or midpoint of vertical half pill.
y	numeric, y position for edge of vertical half pill or midpoint of horizontal half pill.
r	numeric, radius of half pill.

### Value

draws to plot

### Examples

```
op <- par(bg = "black")
plot(0:1, 0:1, asp = 1)
lcars_pill(0.05, 0.45, 0.7, 0.9, "chestnut-rose", "left")
lcars_pill(0.05, 0.45, 0.4, 0.6, "lilac", "both")
lcars_pill(0.05, 0.45, 0.1, 0.3, "orange-peel", "right")
lcars_pill(0.55, 0.65, 0.1, 0.9, "chestnut-rose", "left", vertical = TRUE)
lcars_pill(0.7, 0.8, 0.1, 0.9, "lilac", "both", vertical = TRUE)
lcars_pill(0.85, 0.95, 0.1, 0.9, "orange-peel", "right", vertical = TRUE)
par(op)
```

---

lcars_rect	<i>LCARS rectangle</i>
------------	------------------------

---

**Description**

A simple wrapper around rect.

**Usage**

```
lcars_rect(xmin, xmax, ymin, ymax, color = "atomic-tangerine")
```

**Arguments**

xmin	numeric, left x positions.
xmax	numeric, right x positions.
ymin	numeric, bottom y positions.
ymax	numeric, top y positions.
color	fill and border color. Can be any color given in hex format. Named colors must be LCARS colors. See <a href="#">lcarsdata</a> for options.

**Value**

draws a rectangle

**Examples**

```
plot(0:1, 0:1)
lcars_rect(0.1, 0.9, 0.6, 0.9)
```

---

renderLcars	<i>Create an LCARS outputs (server side)</i>
-------------	--

---

**Description**

Server-side functions for creating dynamic lcarBox and lcarsSweep.

**Usage**

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

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

## Arguments

expr	An expression that returns a Shiny tag object, HTML, or a list of such objects.
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

## See Also

[lcarsOutput](#) for the corresponding UI-side function.

## Examples

```
## Only run examples in interactive R sessions
if (interactive()) {

  ui <- lcarsPage(
    lcarsBoxOutput("box"),
    lcarsSweepOutput("sweep")
  )

  server <- function(input, output) {
    output$box <- renderLcarsBox({
      lcarsBox()
    })
    output$sweep <- renderLcarsSweep({
      lcarsSweep()
    })
  }

  shinyApp(ui, server)
}
```

---

theme\_lcars

*LCARS ggplot themes*

---

## Description

A collection of ggplot2 themes that go well with LCARS styles and colors.

## Usage

```
theme_lcars_light(
  base_size = 11,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)
```

```
theme_lcars_dark(  
  base_size = 11,  
  base_family = "",  
  base_line_size = base_size/22,  
  base_rect_size = base_size/22  
)
```

**Arguments**

`base_size`      base font size.  
`base_family`    base font family.  
`base_line_size` base size for line elements.  
`base_rect_size` base size for rect elements.



# Index

## \* datasets

- lcarsdata, 11
- inputColumn, 2
- lcars, 3
- lcars\_border, 25
- lcars\_bottom\_pill (lcars\_pill), 28
- lcars\_elbow, 27
- lcars\_half\_pill (lcars\_pill), 28
- lcars\_left\_pill (lcars\_pill), 28
- lcars\_pill, 28
- lcars\_rect, 30
- lcars\_right\_pill (lcars\_pill), 28
- lcars\_top\_pill (lcars\_pill), 28
- lcarsApp, 3
- lcarsBox, 4, 21
- lcarsBoxOutput (lcarsOutput), 13
- lcarsBracket, 7
- lcarsButton, 8
- lcarsCheckbox, 9
- lcarsdata, 6, 9, 10, 11, 13, 17, 23, 25–27, 29, 30
- lcarsh1 (lcarsHeader), 11
- lcarsh2 (lcarsHeader), 11
- lcarsh3 (lcarsHeader), 11
- lcarsh4 (lcarsHeader), 11
- lcarsh5 (lcarsHeader), 11
- lcarsh6 (lcarsHeader), 11
- lcarsHeader, 11
- lcarsLeftPill (lcarsRect), 18
- lcarsOutput, 13, 31
- lcarsPage, 14
- lcarsPill (lcarsRect), 18
- lcarsRadio, 16
- lcarsRadioToggle (lcarsRadio), 16
- lcarsRect, 18
- lcarsRightPill (lcarsRect), 18
- lcarsSweep, 6, 20
- lcarsSweepOutput (lcarsOutput), 13

lcarsToggle, 22

lcarsWell, 24

renderLcars, 14, 30

renderLcarsBox (renderLcars), 30

renderLcarsSweep (renderLcars), 30

theme\_lcars, 31

theme\_lcars\_dark (theme\_lcars), 31

theme\_lcars\_light (theme\_lcars), 31