Description

Obtain hex colors from standard LCARS color names.

Usage

\begin{verbatim}
lcars_colors(...)
lcars_2357(...)
lcars_2369(...)
lcars_2375(...)
lcars_2379(...)
\end{verbatim}

Arguments

\begin{verbatim}
... character, LCARS color names.
\end{verbatim}

Details

These functions return the hex colors for LCARS color names. If no argument is provided, the functions return their respective LCARS color set. These functions correspond to LCARS color palettes that contain named colors. Other predefined LCARS color palettes are available but do not have names for each color. You can see all LCARS palettes with \texttt{lcars_pals}.

See Also

\texttt{lcars_pals}

Examples

\begin{verbatim}
lcars_colors()
lcars_2357()
lcars_colors("rust", "danub")
\end{verbatim}
Palettes and palette generating functions based on LCARS colors

Description

Predefined and custom palettes based on LCARS colors.

Usage

lcars_pals()

lcars_pal(palette = "2357", reverse = FALSE)

lcars_colors_pal(palette, reverse = FALSE, ...)

Arguments

palette character, name of a single predefined LCARS palette; or a vector of LCARS or other colors. See details.
reverse logical, reverse color order.
... additional arguments to pass to colorRampPalette.

Details

lcars_pal returns a predefined, qualitative LCARS color palette. lcars_color_pal returns a palette generator based on specific colors. lcars_pals is a function that takes no arguments and returns a list of all predefined LCARS palettes.

Predefined palettes options for palette are "2357", "2369", "2375", "2379", "alt", "first_contact", "nemesis", "nx01", "23c", "29c", "red_alert" and "cardassian".

Custom palettes can also be constructed by passing a vector of colors to palette in lcars_color_pal. This is useful for sequential and divergent palettes. This is essentially a wrapper around colorRampPalette that understands LCARS color names. It accepts any color, allowing you to still use a color like "white" or "#FFFFFF" as the midpoint in a divergent palette for example. A special case is when you provide only a single color, e.g., lcars_color_pal("husk"); this is equivalent to lcars_color_pal(c("white", "husk")).

Value

a palette generating function that takes an argument, n, the number of colors.

Examples

# All predefined LCARS palettes
lcars_pals()

# predefined palette
lcars_pal("2357")

# custom palettes
scale_lcars

Color and fill scale functions for LCARS colors

Description

Scale functions used with ggplot2.

Usage

scale_color_lcars(palette = "2357", discrete = TRUE, reverse = FALSE, ...)

describe_color_lcars1(color = "atomic-tangerine", discrete = TRUE, reverse = FALSE, dark = FALSE, ...)

describe_color_lcars2(low = "atomic-tangerine", high = "near-blue", discrete = TRUE, reverse = FALSE, dark = FALSE, divergent = FALSE, ...)

scale_fill_lcars(palette = "2357", discrete = TRUE, reverse = FALSE, ...)

describe_fill_lcars1(color = "atomic-tangerine", discrete = TRUE, reverse = FALSE, dark = FALSE, ...)

describe_fill_lcars2(low = "atomic-tangerine", high = "near-blue", discrete = TRUE, reverse = FALSE, dark = FALSE, divergent = FALSE, ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>palette</td>
<td>character, name of palette in lcars_pals.</td>
</tr>
<tr>
<td>discrete</td>
<td>logical, discrete or continuous palette.</td>
</tr>
<tr>
<td>reverse</td>
<td>logical, reverse color order.</td>
</tr>
<tr>
<td>...</td>
<td>additional arguments passed to ggplot2: discrete_scale or ggplot2: scale_*_gradientn, for discrete or continuous palettes, respectively.</td>
</tr>
<tr>
<td>color</td>
<td>character, LCARS color name for sequential palette.</td>
</tr>
<tr>
<td>dark</td>
<td>logical, use black instead of white for the base color in sequential palette or midpoint in divergent palette.</td>
</tr>
<tr>
<td>low</td>
<td>character, LCARS color name.</td>
</tr>
<tr>
<td>high</td>
<td>character, LCARS color name.</td>
</tr>
<tr>
<td>divergent</td>
<td>logical, use a divergent palette instead of two-color sequential palette.</td>
</tr>
</tbody>
</table>
Examples

library(ggplot2)
p <- ggplot(diamonds, aes(carat, stat(count), fill = cut)) +
   geom_density(position = "fill")
p + scale_fill_lcars("2357")
p + scale_fill_lcars1("atomic-tangerine", dark = TRUE)
p + scale_fill_lcars2("pale-canary", "danub")

scale_trek

Color and fill scale functions for Star Trek palettes

Description

Scale functions used with ggplot2.

Usage

scale_color_trek(palette = "starfleet", discrete = TRUE,
                 reverse = FALSE, ...)

scale_fill_trek(palette = "starfleet", discrete = TRUE,
                 reverse = FALSE, ...)

Arguments

palette character, name of Star Trek palette. See trek_pal() for list of palette names.
discrete logical, discrete or continuous palette.
reverse logical, reverse color order.
... additional arguments passed to ggplot2::discrete_scale or ggplot2::scale_*_gradientn, for discrete or continuous palettes, respectively.

Details

Most palettes should be used as qualitative palettes. See trekpals to see how many colors are in each predefined palette. Use view_trek_pals() to plot all palettes to see which may work best for your purposes.

Examples

library(ggplot2)
d <- diamonds[diamonds$cut >= "Very Good", ]
ggplot(d, aes(carat, stat(count), fill = cut)) +
   geom_density(position = "fill") +
   scale_fill_trek("starfleet")
Description

A named list of 35 Star Trek color palettes.

Usage

trekpals

Format

A named list.

trek_pal

Description

Return a predefined Star Trek color palette from the trekpals dataset.

Usage

trek_pal(palette, reverse = FALSE)

view_trek_pals(palette)

Arguments

palette character, name of predefined palette. If missing, return all available palette names.
reverse logical, reverse color order.

Details

Many of the palettes are qualitative, and not necessarily evenly spaced in terms of hue, saturation or brightness. This is because many palettes come from logos, symbols, insignia and other simple representations. However, several palettes have specifically been constructed as sequential or divergent if it made sense to do so based on the dominant colors present. Additional special functions exist for the subset of LCARS palettes.

Value

character vector of hex colors or palette names
See Also

lcars_pals, scale_lcars

Examples

trek_pal("lcars_2357")
# leave palette blank to list available names:
trek_pal()

# to view all palettes
view_trek_pals()
Index

*Topic **datasets**
  trekpals, 6

lcars_2357 (lcars_colors), 2
lcars_2369 (lcars_colors), 2
lcars_2375 (lcars_colors), 2
lcars_2379 (lcars_colors), 2
lcars_colors, 2
lcars_colors_pal (lcars_pals), 3
lcars_pal (lcars_pals), 3
lcars_pals, 2, 3, 7

scale_color_lcars (scale_lcars), 4
scale_color_lcars1 (scale_lcars), 4
scale_color_lcars2 (scale_lcars), 4
scale_color_trek (scale_trek), 5
scale_fill_lcars (scale_lcars), 4
scale_fill_lcars1 (scale_lcars), 4
scale_fill_lcars2 (scale_lcars), 4
scale_fill_trek (scale_trek), 5
scale_lcars, 4, 7
scale_trek, 5

trek_pal, 6
trekpals, 6

view_trek_pals (trek_pal), 6