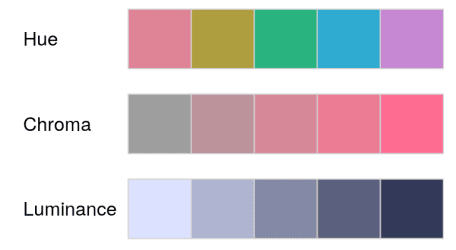
A major update (version 1.4.0) of the R package colorspace has been released to CRAN, enhancing many of the package’s capabilities, e.g., more refined palettes, named palettes, ggplot2 color scales, visualizations for assessing palettes, shiny and Tcl/Tk apps, color vision deficiency emulation, and much more.

**Overview**

The *colorspace* package provides a broad toolbox for selecting individual colors or color palettes, manipulating these colors, and employing them in various kinds of visualizations. Version 1.4.0 has just been released on CRAN, containing many new features and contributions from new co-authors. A new web site presenting and documenting the package has been launched at [http://colorspace.R-Forge.R-project.org/](http://colorspace.r-forge.r-project.org/).

At the core of the package there are various utilities for computing with color spaces (as the name conveys). Thus, the package helps to map various three-dimensional representations of color to each other. A particularly important mapping is the one from the perceptually-based and device-independent color model HCL (Hue-Chroma-Luminance) to standard Red-Green-Blue (sRGB) which is the basis for color specifications in many systems based on the corresponding hex codes (e.g., in HTML but also in R). For completeness further standard color models are included as well in the package: polarLUV() (= HCL), LUV(), polarLAB(), LAB(), XYZ(), RGB(), sRGB(), HLS(), HSV().

The HCL space (= polar coordinates in CIELUV) is particularly useful for specifying individual colors and color palettes as its three axes match those of the human visual system very well: Hue (= type of color, dominant wavelength), chroma (= colorfulness), luminance (= brightness).

[](https://i2.wp.com/eeecon.uibk.ac.at/~zeileis/assets/posts/2019-01-14-colorspace/hcl-properties-1.png?ssl=1)

The *colorspace* package provides three types of palettes based on the HCL model:

* *Qualitative:* Designed for coding categorical information, i.e., where no particular ordering of categories is available and every color should receive the same perceptual weight. Function: qualitative\_hcl().
* *Sequential:* Designed for coding ordered/numeric information, i.e., where colors go from high to low (or vice versa). Function: sequential\_hcl().
* *Diverging:* Designed for coding numeric information around a central neutral value, i.e., where colors diverge from neutral to two extremes. Function: diverging\_hcl().

To aid choice and application of these palettes there are: scales for use with *ggplot2*; shiny (and tcltk) apps for interactive exploration; visualizations of palette properties; accompanying manipulation utilities (like desaturation, lighten/darken, and emulation of color vision deficiencies).

More detailed overviews and examples are provided in the articles:

* [Color Spaces: S4 Classes and Utilities](http://colorspace.r-forge.r-project.org/articles/color_spaces.html)
* [HCL-Based Color Palettes](http://colorspace.r-forge.r-project.org/articles/hcl_palettes.html)
* [HCL-Based Color Scales for *ggplot2*](http://colorspace.r-forge.r-project.org/articles/ggplot2_color_scales.html)
* [Palette Visualization and Assessment](http://colorspace.r-forge.r-project.org/articles/palette_visualization.html)
* [Apps for Choosing Colors and Palettes Interactively](http://colorspace.r-forge.r-project.org/articles/hclwizard.html)
* [Color Vision Deficiency Emulation](http://colorspace.r-forge.r-project.org/articles/color_vision_deficiency.html)
* [Color Manipulation and Utilities](http://colorspace.r-forge.r-project.org/articles/manipulation_utilities.html)
* [Approximating Palettes from Other Packages](http://colorspace.r-forge.r-project.org/articles/approximations.html)

**Installation**

The stable release version of *colorspace* is hosted on the Comprehensive R Archive Network (CRAN) at [https://CRAN.R-project.org/package=colorspace](https://cran.r-project.org/package=colorspace) and can be installed via

install.packages("colorspace")

The development version of *colorspace* is hosted on R-Forge at [https://R-Forge.R-project.org/projects/colorspace/](https://r-forge.r-project.org/projects/colorspace/) in a Subversion (SVN) repository. It can be installed via

install.packages("colorspace", repos = "http://R-Forge.R-project.org")

For Python users a beta re-implementation of the full *colorspace* package in Python 2/Python 3 is also available, see <https://github.com/retostauffer/python-colorspace>.

**Choosing HCL-based color palettes**

The *colorspace* package ships with a wide range of predefined color palettes, specified through suitable trajectories in the HCL (hue-chroma-luminance) color space. A quick overview can be gained easily with hcl\_palettes():

library("colorspace")

hcl\_palettes(plot = TRUE)

[](https://i2.wp.com/eeecon.uibk.ac.at/~zeileis/assets/posts/2019-01-14-colorspace/hcl-palettes-1.png?ssl=1)

Using the names from the plot above and a desired number of colors in the palette a suitable color vector can be easily computed, e.g.,

q4 <- qualitative\_hcl(4, "Dark 3")

q4

## [1] "#E16A86" "#909800" "#00AD9A" "#9183E6"

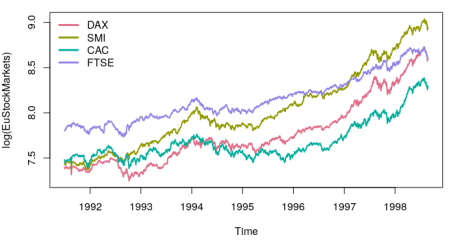
The functions sequential\_hcl(), and diverging\_hcl() work analogously. Additionally, their hue/chroma/luminance parameters can be modified, thus allowing to easily customize each palette. Moreover, the choose\_palette()/hclwizard() app provide convenient user interfaces to do the customization interactively. Finally, even more flexible diverging HCL palettes are provided by divergingx\_hcl().

**Usage with base graphics**

The color vectors returned by the HCL palette functions can usually be passed directly to most base graphics function, typically through the col argument. Here, the q4 vector created above is used in a time series display:

plot(log(EuStockMarkets), plot.type = "single", col = q4, lwd = 2)

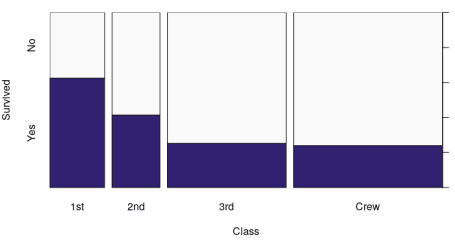
legend("topleft", colnames(EuStockMarkets), col = q4, lwd = 3, bty = "n")

[](https://i1.wp.com/eeecon.uibk.ac.at/~zeileis/assets/posts/2019-01-14-colorspace/eustockmarkets-plot-1.png?ssl=1)

As another example for a sequential palette a spine plot is created below, displaying the proportion of Titanic passengers that survived per class. The Purples 3 palette is used which is quite similar to the **ColorBrewer.org** palette Purples. Here, only two colors are employed, yielding a dark purple and light gray.

ttnc <- margin.table(Titanic, c(1, 4))[, 2:1]

spineplot(ttnc, col = sequential\_hcl(2, "Purples 3"))

[](https://i1.wp.com/eeecon.uibk.ac.at/~zeileis/assets/posts/2019-01-14-colorspace/titanic-plot-1.png?ssl=1)

**Usage with *ggplot2***

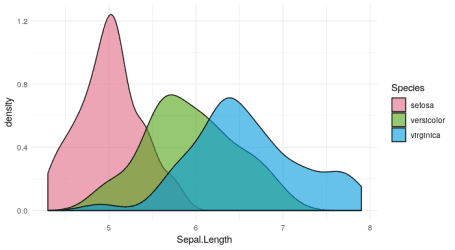
To plug the HCL color palettes into *ggplot2* graphics suitable discrete and/or continuous *gglot2* color scales are provided. The scales are called via the scheme scale\_\_\_(), where is the name of the aesthetic (fill, color, colour), is the type of the variable plotted (discrete or continuous) and sets the type of the color scale used (qualitative, sequential, diverging, divergingx).

To illustrate their usage two simple examples are shown using the qualitative Dark 3 and sequential Purples 3 palettes that were also employed above. First, semi-transparent shaded densities of the sepal length from the iris data are shown, grouped by species.

library("ggplot2")

ggplot(iris, aes(x = Sepal.Length, fill = Species)) + geom\_density(alpha = 0.6) +

scale\_fill\_discrete\_qualitative(palette = "Dark 3")

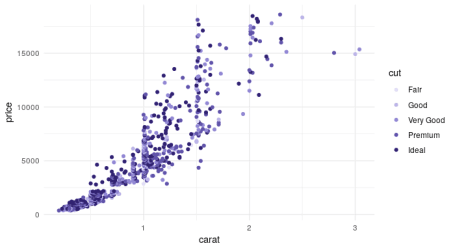
[](https://i2.wp.com/eeecon.uibk.ac.at/~zeileis/assets/posts/2019-01-14-colorspace/iris-ggplot-1.png?ssl=1)

The sequential palette is used to code the cut levels in a scatter of price by carat in the diamonds data (or rather a small subsample thereof). The scale function first generates six colors but then drops the first color because the light gray is too light in this display. (Alternatively, the chroma and luminance parameters could also be tweaked.)

dsamp <- diamonds[1 + 1:1000 \* 50, ]

ggplot(dsamp, aes(carat, price, color = cut)) + geom\_point() +

scale\_color\_discrete\_sequential(palette = "Purples 3", nmax = 6, order = 2:6)

[](https://i1.wp.com/eeecon.uibk.ac.at/~zeileis/assets/posts/2019-01-14-colorspace/diamonds-ggplot-1.png?ssl=1)

**Palette visualization and assessment**

The *colorspace* package also provides a number of functions that aid visualization and assessment of its palettes.

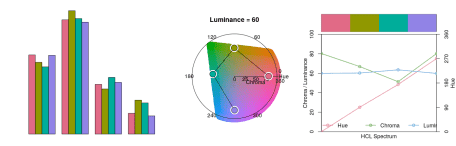
* demoplot() can display a palette (with arbitrary number of colors) in a range of typical and somewhat simplified statistical graphics.
* hclplot() converts the colors of a palette to the corresponding hue/chroma/luminance coordinates and displays them in HCL space with one dimension collapsed. The collapsed dimension is the luminance for qualitative palettes and the hue for sequential/diverging palettes.
* specplot() also converts the colors to hue/chroma/luminance coordinates but draws the resulting spectrum in a line plot.

For the qualitative Dark 3 palette from above the following plots can be obtained.

demoplot(q4, "bar")

hclplot(q4)

specplot(q4, type = "o")

[](https://i1.wp.com/eeecon.uibk.ac.at/~zeileis/assets/posts/2019-01-14-colorspace/allplots-qualitative-1.png?ssl=1)

A bar plot would be another typical application for a qualitative palette (instead of the time series and density plot used above). However, a lighter and less colorful palette might be preferable in this situation (e.g., Pastel 1 or Set 3).

The other two displays show that luminance is (almost) constant in the palette while the hue changes linearly along the color “wheel”. Ideally, chroma would have also been constant to completely balance the colors. However, at this luminance the maximum chroma differs across hues so that the palette is fixed up to use less chroma for the yellow and green elements.

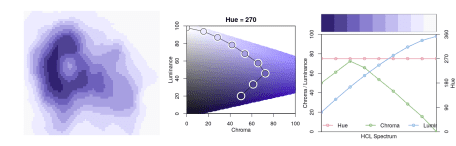
Subsequently, the same assessment is carried out for the sequential Purples 3 palette as employed above.

s9 <- sequential\_hcl(9, "Purples 3")

demoplot(s9, "heatmap")

hclplot(s9)

specplot(s9, type = "o")

[](https://i1.wp.com/eeecon.uibk.ac.at/~zeileis/assets/posts/2019-01-14-colorspace/allplots-sequential-1.png?ssl=1)

Here, a heatmap (based on the well-known Maunga Whau volcano data) is used as a typical application for a sequential palette. The elevation of the volcano is brought out clearly, focusing with the dark colors on the higher elevations.

The other two displays show that hue is constant in the palette while luminance and chroma vary. Luminance increases monotonically from dark to light (as required for a proper sequential palette). Chroma is triangular-shaped which allows to better distinguish the middle colors in the palette (compared to a monotonic chroma trajectory).