# Package 'Spectran'

January 20, 2025

**Title** Visual and Non-Visual Spectral Analysis of Light

**Version** 1.0.6

**Description** Analyse light spectra for visual and non-visual (often called melanopic) needs, wrapped up in a Shiny App. 'Spectran' allows for the import of spectra in various CSV forms but also provides a wide range of example spectra and even the creation of own spectral power distributions. The goal of the app is to provide easy access and a visual overview of the spectral calculations underlying common parameters used in the field. It is thus ideal for educational purposes or the creation of presentation ready graphs in lighting research and application. 'Spectran' uses equations and action spectra described in CIE S026 (2018) <doi:10.25039/S026.2018>, DIN/TS 5031-100 (2021) <doi:10.31030/3287213>, and ISO/CIE 23539 (2023) <doi:10.25039/IS0.CIE.23539.2023>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Depends** R (>= 2.10)

**LazyData** true

**Imports** chromote, colorSpec, cowplot, dplyr, gghighlight, ggplot2, ggrepel, ggridges, ggtext, grid, gt, htmltools, magrittr, openxlsx, pagedown, patchwork, png, purrr, readr, rlang, scales, shiny, shinyalert, shinydashboard, shinyFeedback, shinyjs, shinyWidgets, spacesXYZ, spsComps, stats, stringr, tibble, tidyr, tidyselect, utils, waiter, webshot2, withr

**Suggests** config, here, magick, pkgload, readxl, rhub, rmarkdown, rsconnect

**URL** https://github.com/LiTGde/Spectran,
https://litgde.github.io/Spectran/,
https://doi.org/10.5281/zenodo.11518043

**BugReports** https://github.com/LiTGde/Spectran/issues

**NeedsCompilation** no

# Contents

---

ColorP *Hex Color pallets for Spectran*

---

#### Description

Contains named vectors of Hex color codes for Spectran's color pallets.

#### Usage

    ColorP

#### Format

ColorP A list containing 5 elements, each a named vector of colors

**Lang** 401 colors along the spectrum that make a nice, yet nonaccurate representation of the visible spectrum. Created from a picture provided by Dieter Lang.

**Lang_bright** same as Lang but with a slightly brightened version of the picture. Strikes a nice balance between "accuracy" and "nice colors". Is the default when starting Spectran()

**Dan_Bruton** 401 colors along the spectrum that were made with the cooltools::wavelength2col() function and are based on Fortran Code by Dan_Bruton. Probably most accurate, but black at both spectral ends, which does not represent the colors themselves but also human sensitivity to them.

**Rainbow** 401 colors along the spectrum that make a vibrant and pleasing, yet nonaccurate representation of the visible spectrum. Created by the grDevices::rainbow() function.

**Color_Rendering** 14 colors that represent the reference colors used to determine the color rendering index.

## Source

[grDevices::rainbow()](grDevices::rainbow())

[cooltools](cooltools)

<https://github.com/LiTGde/Spectran/issues/14>

---

examplespectra *Example spectra used by Spectran*

---

## Description

Contains Measurement and CIE Standard Illuminant Spectra. Additional data to the spectra can be found in the examplespectra_descriptor dataset.

## Usage

```
examplespectra
```

## Format

examplespectra A list of two elements, each containing a table of spectral data

**Measurement** Spectral data measured by a spectroradiometer. The unit of the wavelength (Wellenlaenge) is nm, the unit of the spectral data is W/m^2.

**CIE** Spectral data from the CIE describing Standard Illuminants. The unit of the wavelength (Wellenlaenge) is nm, the unit of the spectral data is W/m^2.

## Source

<https://cie.co.at/data-tables>

---

examplespectra_descriptor

*Details on the example spectra used by Spectran*

---

## Description

Contains additional details on the measurement and CIE Standard Illuminant Spectra collected in examplespectra.

## Usage

```
examplespectra_descriptor
```

## Format

examplespectra_descriptor A list with an element per language. The following description covers elements within each language element, which is a tibble with 11 rows and 8 columns.

**Name** Category of the spectrum (short)

**Beschreibung** Short description of the category of the spectrum

**Identifier** A list of character vectors containing the unique identifiers of each spectrum

**Button_Name** A list of character vectors containing the text to be displayed on the buttons for each spectrum

**Dateinamen** A list of character vectors containing the text used for the download-filenames

**URL** A character vector with URLs to videos giving some context to the measurements or NA, in which case nothing will be displayed

**embargo** A logical vector that indicates whether a special message should be displayed before downloading this spectrum

**download** A list of named character vectors containing a short description of the spectrum.

## Source

https://cie.co.at/data-tables

---

| Specs | *Specifications for visual and nonvisual data wrangling and plotting* |
|---|---|

---

## Description

Contains vectors and tables that are the main basis for the calculations Spectran does and also contain basic naming and coloring conventions.

## Usage

Specs

## Format

Specs A list with 7 elements

**AS_wide** A table of action spectra for the 5 human receptor types and V(lambda) from 1924. The table is in the wide format. The unit of the wavelength (Wellenlaenge) is nm, the unit of the action spectrum is 1 (dimensionless).

**AS_long** A table of action spectra for the 5 human receptor types and V(lambda) from 1924. The table is in the long format. The unit of the wavelength (Wellenlaenge) is nm, the unit of the action spectrum is 1 (dimensionless).

**Efficacy** A named vector of inverse values of efficacy of luminous radiation for the 5 human receptor types and V(lambda) from 1924. The unit is lm/W.

**Alpha** A list. `names` contains the names of the 5 human receptor types. `adjectives` contains these names in their adjective form for every languages included in Spectran. `descriptions` contains the descriptions used for tables and plotting. abb contains the abbreviations (if any) used for the equivalent daylight illuminance for that receptor type.

**Vlambda** A character scalar containing the properly escaped version of V(lambda), used e.g. in Plots

**Alpha.ico** A character scalar containing the properly escaped version of the sign `alpha`, used e.g. in Plots

**Plot** A table of Peak Wavelength (nm) values, Names, Abbreviations, and colors for the 5 human receptor types and V(lambda) from 1924. This table is mainly used for plotting.

## Source

[CIE S026](#)

[DIN/TS 5031-100](#)

---

Spectran | *Unlock the Spectrum: Easy, Educational, and Engaging Analysis of Light Spectra*

---

## Description

Unlock the Spectrum: Easy, Educational, and Engaging Analysis of Light Spectra

## Usage

```
Spectran(
  lang_setting = "English",
  lang_link = FALSE,
  color_palette = "Lang",
  ...
)
```

## Arguments

| | |
|---|---|
| lang_setting | A language for the application. Currently **Deutsch** for German and **English** (default) are implemented. Expects a *character*. |
| lang_link | Only relevant for the App deployed on *Shinyapps.io*. Handles whether a link to the German/English Version of the App is present in the header. Expects a *logical* (default FALSE) |
| color_palette | A color palette for the application. Currently **Lang** (default), **Lang_bright**, **Dan_Bruton**, and **Rainbow** are implemented. Expects a *character*. In terms of color accuracy, the decending order is likely **Dan_Bruton**, **Lang**, **Lang_bright**, and **Rainbow**. However, all of them are wrong in the sense, that monochromatic light can not well be recreated with RGB colors. Look at the documentation for [ColorP](#) for more information about these palettes. |
| ... | Any other settings that get passed to shinyApp |

## Value

Open a viewer with the shiny app

## Examples

```
if(interactive()) {
Spectran()}

#try another language
if(interactive()) {
Spectran(lang_setting = "Deutsch")}

#or try another color palette
if(interactive()) {
Spectran(color_palette = "Dan_Bruton")}
```

# Index