



# Colour

## for Data Visualisation

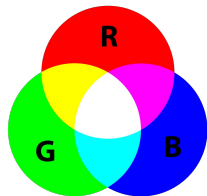
Information Visualisation 2023  
Group 2  
Inge Gsellmann, Michael Hebesberger, Danijela Lazarevic  
03.05.23

Copyright 2023 by the author(s), except as otherwise noted.  
This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence

# The Basics

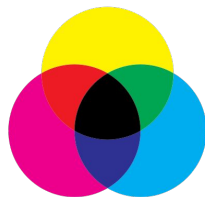
# Colour Models

## RGB



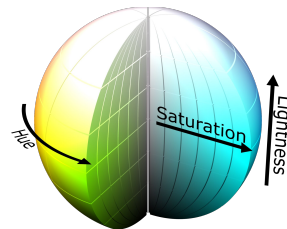
- Red, Green and Blue
- Additive colour model
- Cameras, computer screens

## CMYK



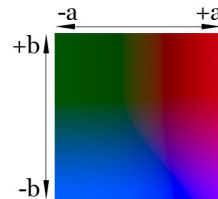
- Cyan, Magenta, Yellow and Key
- Subtractive colour model
- Printing applications

## HSL/HSV



- Hue, Saturation and Lightness/Value
- Based on perception of human vision
- Graphic design, web development

## Lab

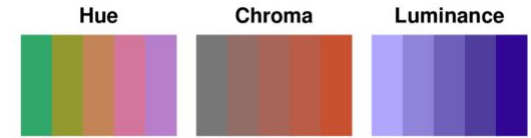


- Lightness, a and b
- Device independent
- Scientific applications, colour correction and matching

All images taken from:  
<https://commons.wikimedia.org>, used under Creative Commons License

# HCL Colour Model

- Also called LCh
- **Hue**, **Chroma** and **Luminance**
- Similar to HSV, but uses chroma instead of saturation to describe the intensity of a colour.
- Describe colours consistent with human perception.
- Choose colours that are perceptually equally different from each other.
- Commonly used in data visualisation, graphic design, and to create aesthetically pleasing colour palettes.



<https://hclwizard.org/color-scheme>, used under  
Creative Commons Attribution 3.0 Unported  
License

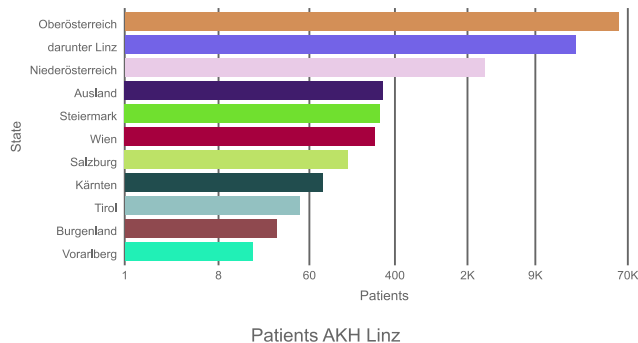
# Colour in Data Visualisation

# Colour in Data Visualisation

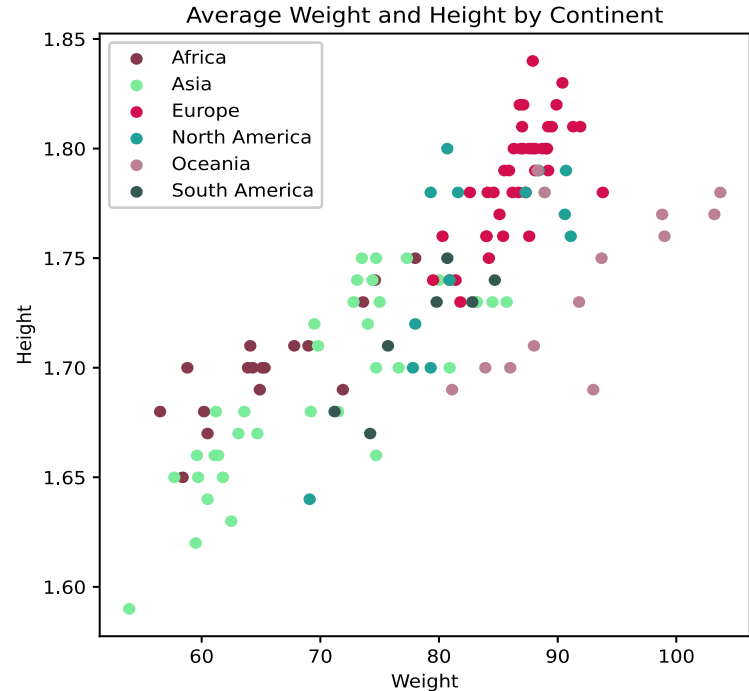
- Categorical colour palettes
- Continuous colour scales
- Diverging colour scales
- Highlighting/De-Emphasising
- Binned colour scales

# Categorical Colour Palettes

- Different hues represent categories.
- No intrinsic order between categories (e.g. countries, gender, companies).
- Good when number of categories is low (hard to read if too many hues).
- Different lightness in hues makes them distinguishable in grayscale.



Source: image created by Michael Hebesberger



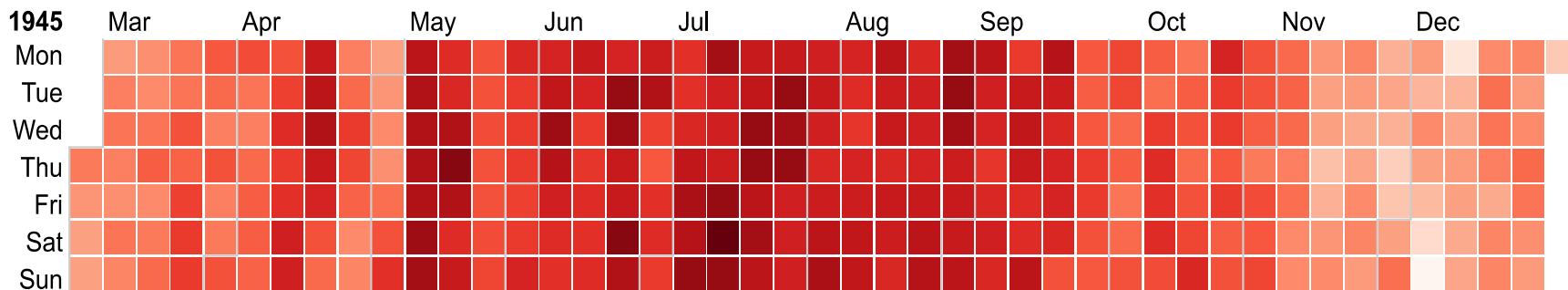
Source: image created by Danijela Lazarevic

Source: <https://blog.datawrapper.de/which-color-scale-to-use-in-data-vis/>

# Continuous Colour Scales

- Gradients (bright to dark)
- Order (visualise values from low to high)

## Mean temperature (°C)



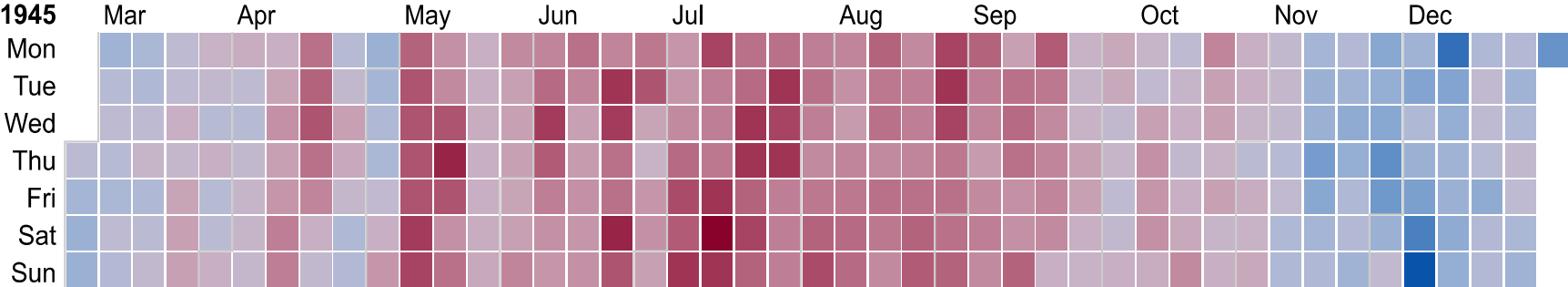
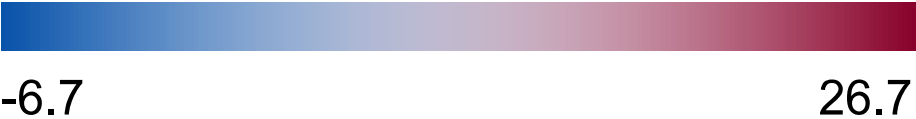
Source: chart created by Danijela Lazarevic using [RawGraphs](#)



# Diverging Colour Scales

- Same as continuous but have bright middle and darker ends.
- Good for depicting negative and positive values.

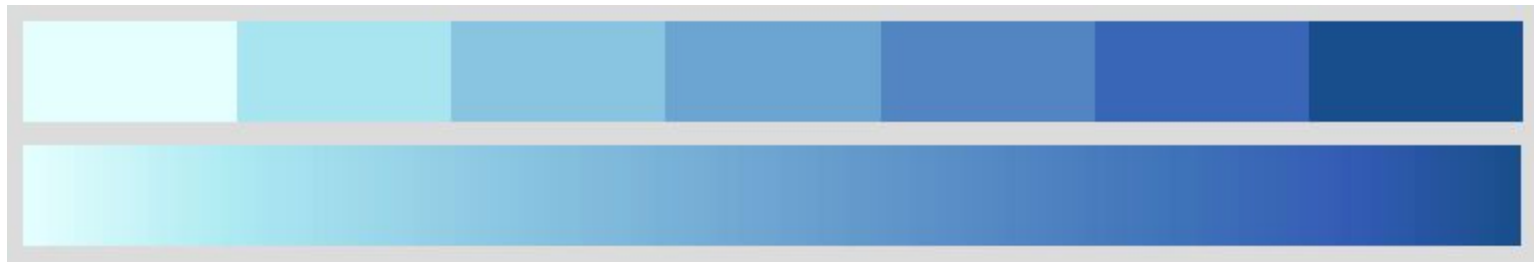
## Mean temperature (°C)



Source: chart created by Danijela Lazarevic using [RawGraphs](https://rawgraphs.io/)

# Binned Colour Scales

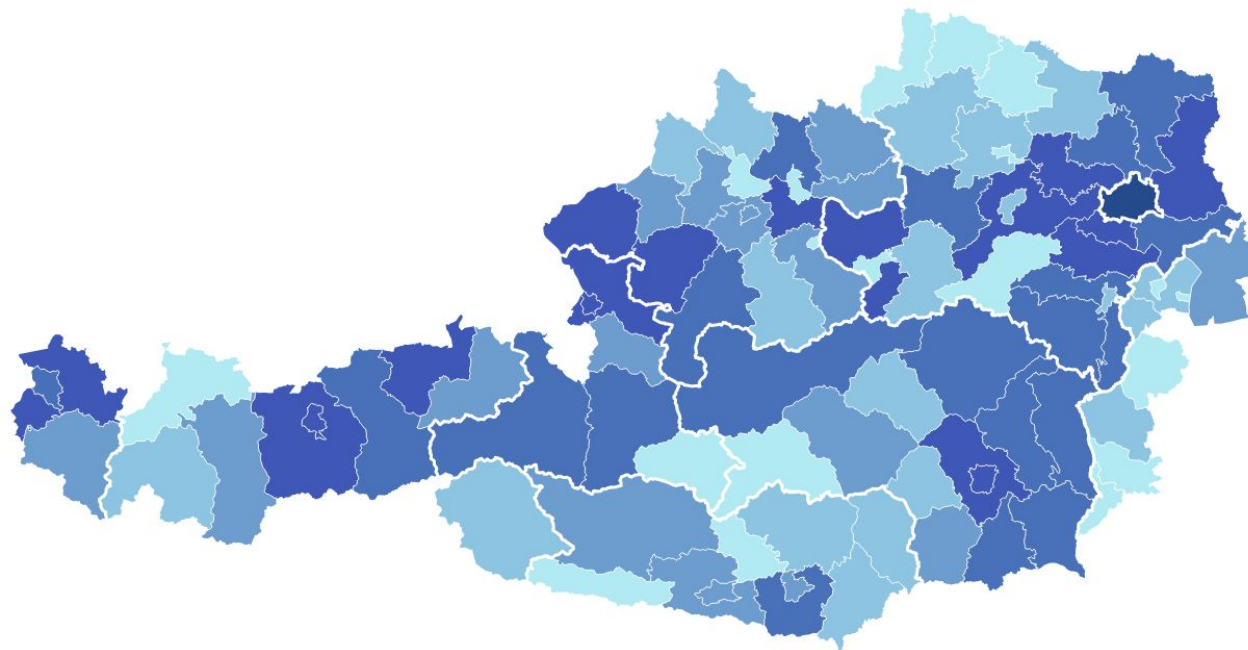
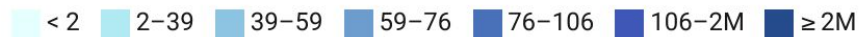
- Continuous data can be binned.
- Data values are split into bins or brackets.
- Increases readability
- Reduces noise by smoothing out outliers.
- Different ways to divide continuous data (linear, quantile, quintile, logarithmic, natural breaks, etc.).



Source: images created by Danijela Lazarevic using [chroma.js](https://github.com/danijela-lazarevic/chroma.js)

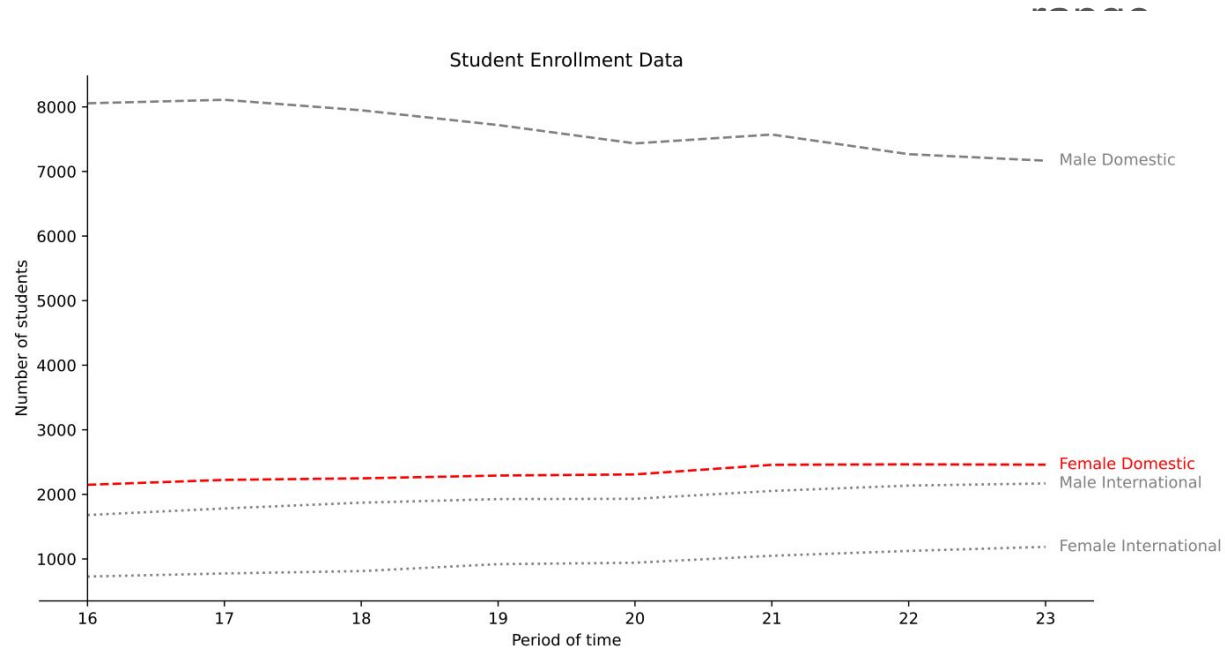
# Binned Colour Scales

## Population Density



# Highlighting/De-Emphasising

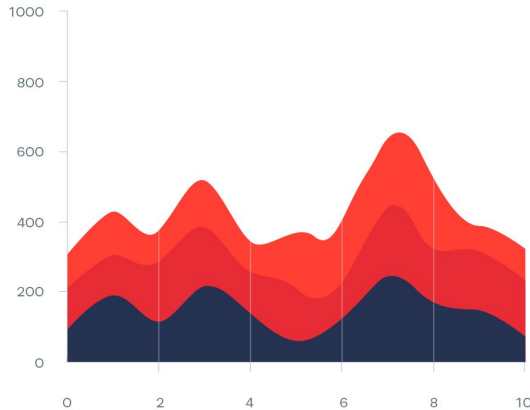
- Highlight a category or that is important.
- De-emphasise categories with “others” or “no data”.
- Differentiate between data by using dashed or dotted lines.



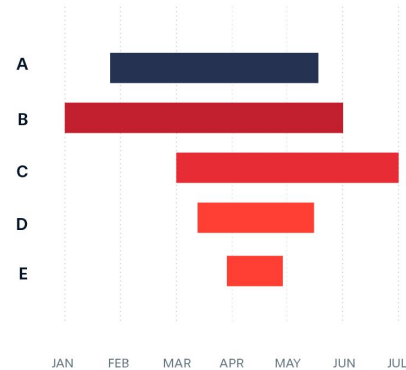
Source: chart created by Danijela Lazarevic

# Touching Colours vs. Distanced Colours

- Directly touching colours are easier to discern as different.
- Large distances make colour comparison difficult.



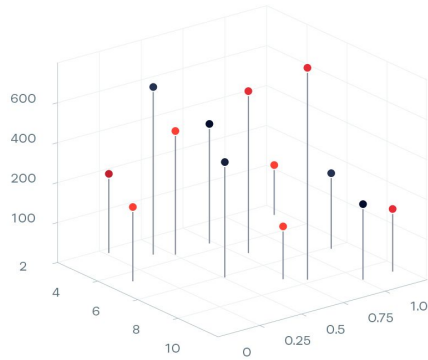
<https://datavizproject.com/data-type/stacked-area-chart/>, used under Creative Commons Attribution 4.0 International License



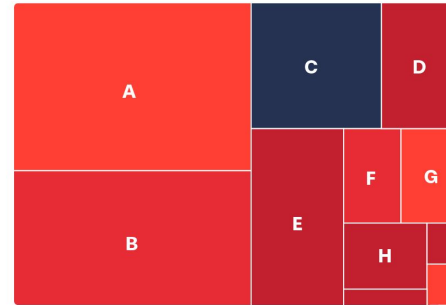
<https://datavizproject.com/data-type/gannt-chart/>, used under Creative Commons Attribution 4.0 International License

# Area Size

- Smaller areas make it harder to identify colours.
- Smaller points need a higher contrast to make them distinguishable.



<https://datavizproject.com/data-type/3d-scatterplot/>,  
used under Creative Commons Attribution 4.0  
International License

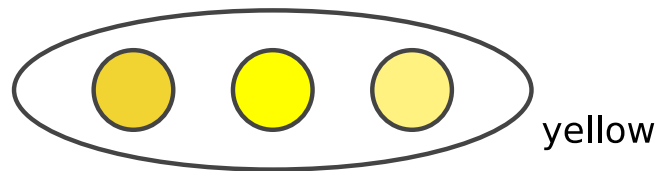


<https://datavizproject.com/data-type/treemap/>, used  
under Creative Commons Attribution 4.0  
International License


# Colour in Culture

# Colour Names

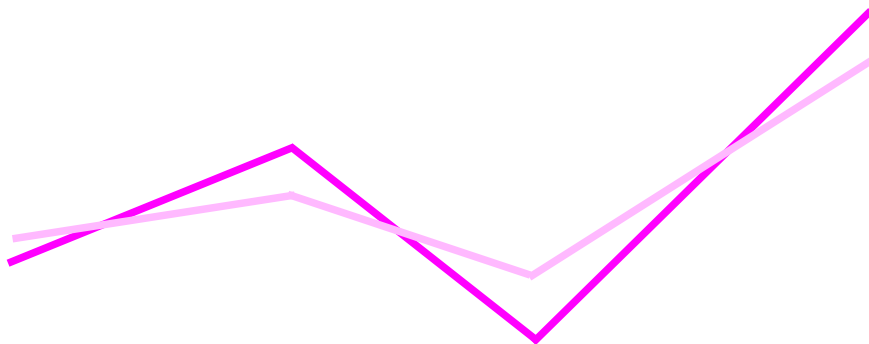
- Names describe hues of colours instead of specific colour values.
- Not all colour names are unambiguous.
- Different languages/cultures assign different names to colours.



aqua? petrol?  
blue? green?  
teal? turquoise?



English:  
"the pink line"

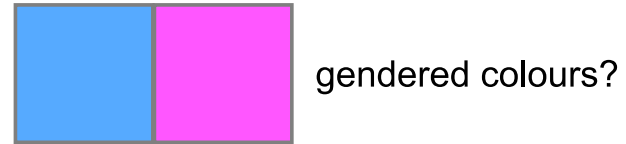
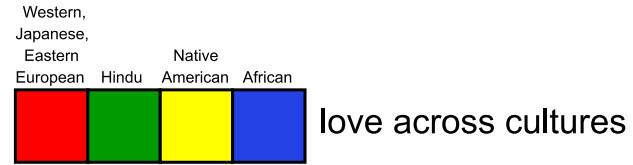
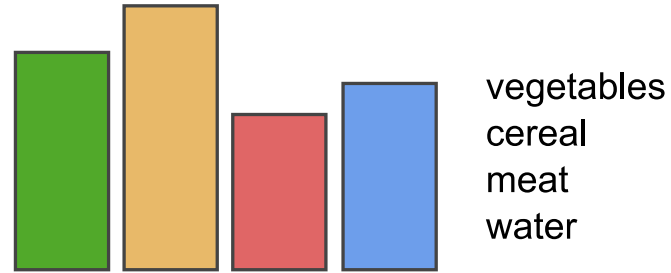


Deutsch:  
"die rosarote Linie"  
"die pinke Linie"



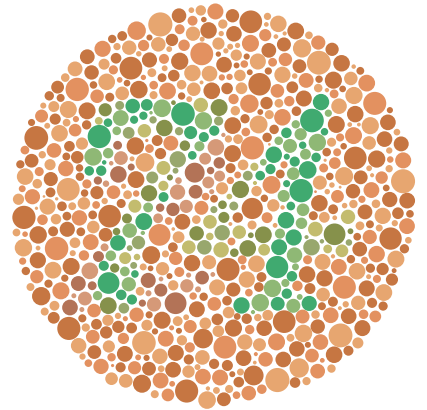
# Colour Associations

- Tangible associations
  - Objects
  - Officially assigned colours
- Conceptual associations
  - Change with culture
  - Change over time

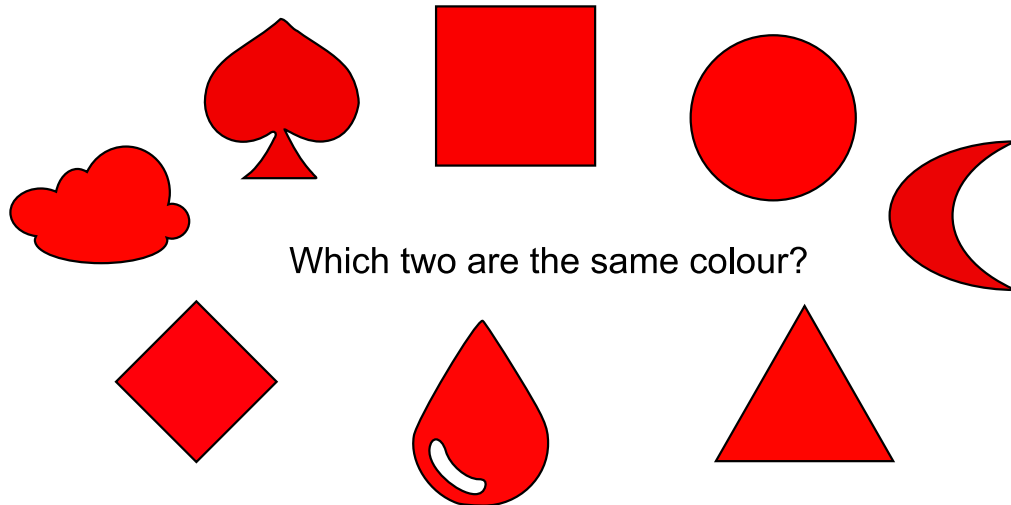


# Accessible Colour

- Different kinds of colour blindness
  - Achromatopsia, protanopia, deuteranopia, tritanopia...
  - Generally genetic but also caused by old age
- Trouble with small contrasts



Source, public domain:  
[https://en.wikipedia.org/wiki/File:Ishihara\\_9.svg](https://en.wikipedia.org/wiki/File:Ishihara_9.svg)



Which two are the same colour?

Source: image created by Inge Gsellmann



No colour blindness



Deuteranopia



Tritanopia



Monochromacy

# Tools

# Colorgorical

<http://vrl.cs.brown.edu/color>

- Interactive colour palette generator
- Different scores for generating:
  - Perceptual Distance
  - Name Difference
  - Pair Preference
  - Name Uniqueness
  - Hue filters
  - Lightness range
- Free to use

The screenshot displays the Colorgorical web application interface. On the left is a control panel with the following sections:

- Generate**: A blue button at the top.
- Number of colors**: A slider set to 5.
- Score Importance**: A vertical stack of sliders for:
  - Perceptual Distance
  - Name Difference
  - Pair Preference
  - Name Uniqueness
- Select hue filters**: A circular hue wheel with a central dot and a label "Drag wheel, or add angle:" below it. The wheel has markers at 90°, 180°, and 270°. Below the wheel are input fields for "# to #" and a "+" button.
- Select lightness range**: A horizontal lightness slider with input fields for "25" and "85".

The main area on the right shows the results of the generation process. It features a top navigation bar with tabs for "Color space", "Hex", "RGB", "Lab", and "LCH", and a dropdown for "Array format" set to "No quote". Below this, there are four rows of results, each corresponding to a different set of parameters. Each row includes:

- A list of the generated colors with their RGB values and a "+ start" button.
- A mosaic visualization of the colors.
- A bar chart showing the distribution of colors.
- A scatter plot of the colors.

The four rows of results shown are:

- Colors: `rgb(120,185,143)`, `rgb(207,123,93)`, `rgb(28,241,16)`. RGB values: 120, 185, 143; 207, 123, 93; 28, 241, 16.
- Colors: `rgb(160,232,91)`, `rgb(125,68,0)`, `rgb(133,210,225)`. RGB values: 160, 232, 91; 125, 68, 0; 133, 210, 225.
- Colors: `rgb(50,150,77)`, `rgb(243,106,213)`, `rgb(117,237,13)`. RGB values: 50, 150, 77; 243, 106, 213; 117, 237, 13.
- Colors: `rgb(57,146,131)`, `rgb(168,221,230)`, `rgb(33,77,78)`. RGB values: 57, 146, 131; 168, 221, 230; 33, 77, 78.

# Generating a Colour Palette with Colorgorical

Colorgorical Source

Generate

Number of colors: 1

Score importance

Perceptual Distance

Name Difference

Pair Preference

Name Uniqueness

Select hue filters

90°

180°

270°

Results: Color space Hex RGB Lab LCH Array format No quote Charts Clear all

## Instructions

To generate a palette with  $n$  colors, just enter the number of colors you want and click *Generate*. Bigger palettes will take longer than smaller palettes to make. Results will automatically appear when ready.

For greater detail, please consult our [paper](#) or the [source code](#).

### Score Importance

**Perceptual Distance**  
Increasing *Perceptual Distance* favors palette colors that are more easily discriminable to the human eye. To accurately model human color acuity, this is performed using CIEDE2000 in CIE Lab color space.

**Name Difference**  
Increasing *Name Difference* favors palette colors that share few common names. This is similar to perceptual distance, but can lead to different results in certain areas of color space. This happens when there are many different names for perceptually close colors (e.g., red and pink are perceptually close but named differently). Colorgorical calculates this using Heer and Stone's [Name Difference function](#), which is built on top of the [XKCD color-name survey](#).

**Pair Preference**  
Increasing *Pair Preference* favors palette colors that are, on average, predicted to be more aesthetically preferable together. Typically these colors are similar in hue, have different lightness, and are cooler colors (blues and greens). Pair Preference is based off of [Schloss and Palmer's research on color preference](#).

**Name Uniqueness**

## About

Colorgorical was built by Connor Gramazio with advisement from David Laidlaw and Karen Schloss.

## Documentation

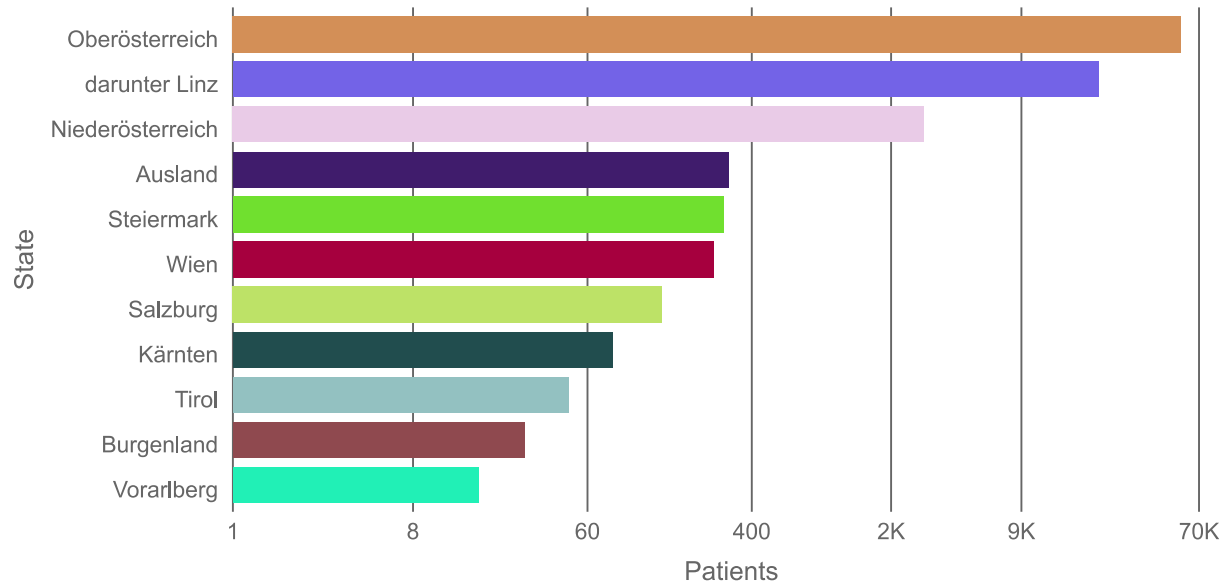
If you'd like to read more about how Colorgorical works, please read our paper [here](#). If you're curious about the implementation, please see the Colorgorical GitHub repository located [here](#).

If you use Colorgorical, please use the following citation:

```
@article{gramazio-2017-ccd,  
  author={Gramazio, Connor C. and Laidlaw, David H. and Schloss, Kare  
  journal={IEEE Transactions on Visualization and Computer Graphics},  
  title={Colorgorical: creating discriminable and preferable color pa  
  year={2017}  
}
```

<https://youtu.be/rONW1sDPUOQ>

# Example Chart



Patients AKH Linz

# Chroma.js Color Palette Helper

<https://www.vis4.net/palettes/#/9|s|00429d.96ffea.ffffe0|ffffe0.ff005e.93003a|1|1>

- Palette Helper based on Chroma.js
- Colour palette definition
- Colour selection
- Check for colourblind-visibility
- Export colours in different coding styles

## Chroma.js Color Palette Helper

This [chroma.js](#)-powered tool is here to help us [mastering multi-hued, multi-stops color scales](#).

1 What kind of palette do you want to create?

Palette type:

Number of colors:

2 Select and arrange input colors

3 Check and configure the resulting palette

correct lightness  bezier interpolation

**⚠ This palette is not colorblind-safe.**

simulate:



Screenshot captured by Inge Gsellmann

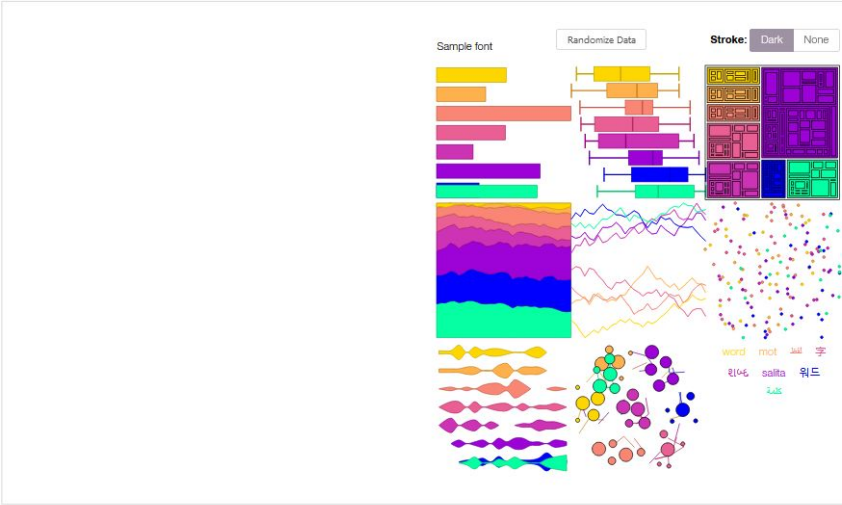


# Viz Palette

<https://projects.susielu.com/viz-palette>

- Interactive colour palette generator.
- Optimized for tweaking, copying, and pasting in and out of JavaScript.
- Examples for easy reviewing of palette.
- Different views to simulate accessibility for colour blindnesses.
- Colour report about names and difficulty of telling them apart.

Color Population: No Color Deficiency - 98% Deuteranomaly - 2.7% Protanomaly - 0.66% Protanopia - 0.59% Deuteranopia - 0.56% Greyscale



The screenshot shows the Viz Palette tool interface. At the top, there's a 'Color Population' section with buttons for different color deficiency types: No Color Deficiency (98%), Deuteranomaly (2.7%), Protanomaly (0.66%), Protanopia (0.59%), Deuteranopia (0.56%), and Greyscale. Below this, there's a 'Sample font' section with a 'Randomize Data' button and a 'Stroke' dropdown menu set to 'Dark'. The main area displays several visualization examples: a horizontal bar chart, a stacked area chart, a line chart, a scatter plot, and a network graph. A 'COLOR REPORT' section is visible at the bottom, listing color names and their corresponding hex codes, along with a note about minimizing name conflicts for categorical palettes.

Sample font Randomize Data Stroke: Dark None

word root 词 字  
리(리) salita 워드  
عنة

### COLOR REPORT

Area link colors difficult to tell apart acc.	#ea5f94	pink	• Minimize name conflicts for categorical palettes
— Lines or small points	#fab775	light orange	
— Medium areas	#fb14e	orange	
— Large areas	#fd70d	gold	
	#06fa2	green	
	#0000ff	indigo	
	#cd34b5	magenta •	
	#9d02d7	magenta •	

Screenshot captured by Michael Hebesberger.

# HCL Wizard

<https://hclwizard.org/>

- Free online open-source tools for creating colour palettes based on HCL.
- Information about choosing right colours for palettes.
- Different online tools:
  - Palette Creator
  - Deficiency Emulator
  - Colour Picker
- Export palettes for easy import in data visualisation tools.

The screenshot displays the HCL Wizard interface. On the left, the 'Base Options' panel includes a 'Type of palette' dropdown set to 'Basic: Diverging', a 'Base color scheme' dropdown set to 'Blue-Red', and an 'Example' dropdown set to 'Map'. Below these are 'Control Options' with checkboxes for 'Reverse', 'Correct colors' (checked), 'Dark mode', and 'Desaturated', and 'Vision' options with radio buttons for 'Normal' (selected), 'Deutan', 'Protan', and 'Tritan'. The 'Color Settings' panel features sliders for Hue 1 (260), Hue 2 (0), Chroma (80), Luminance (30), Luminance (80), Power (1.5), and Number (7), each with a 'SET' button. A 'Return to R' button is at the bottom. The 'Example Plot' shows a map of South Carolina with a diverging color palette. The top navigation bar includes 'Spectrum', 'Color Plane', 'Export', and 'Info'.

<http://hclwizard.org:3000/hclwizard/>, used under Creative Commons Attribution 3.0 Unported License

# Tool Capabilities

Tool Name	Categorical data	Continuous data	Generator	Pre-Made Palettes	Colour Blindness	Contrast Checker	Open-Source
<a href="#">Colorgorical</a>	✓	✗	✓	✗	✗	✓	✓
<a href="#">Chroma.js</a>	✗	✓	✓	✗	✓	✗	✓
<a href="#">Viz Palette</a>	✓	✓	✓	✗	✓	✗	✓
<a href="#">HCL Wizard</a>	✓	✓	✓	✓	✓	✗	✓

# Libraries

# Chroma.js (JavaScript)

<https://github.com/gka/chroma.js/>

- Allows the user to work with colours and colour spaces.
- Colour spaces: RGB, HEX, HSL, HSV, LAB, LCH and CMYK.
- Create and manipulate by mixing, lightening, darkening and saturating colours.
- Support for colour interpolation and scale generation.
- 9.4 k stars on GitHub.

```
// set luminance to 50% for all colors  
chroma('white').luminance(0.5);  
chroma('aquamarine').luminance(0.5);  
chroma('hotpink').luminance(0.5);  
chroma('darkslateblue').luminance(0.5);
```

```
#bcbcbc  
#67ceab  
#ff9dce  
#bcb8d5
```

<https://gka.github.io/chroma.js/>, used  
under BSD 3-Clause license.

# AC Colors (JavaScript)

<https://github.com/vinaypillai/ac-colors>

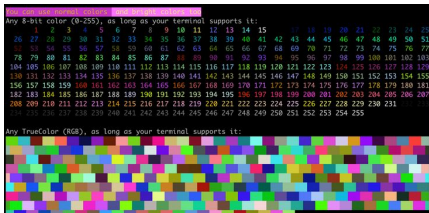
- Convert colours between colour spaces.
- Handle random colour generation and contrast ratio calculation.
- Outputs can be used for Chroma.js
- 272 stars on GitHub



# JColor (Java)

<https://github.com/dialex/JColor>

- Support for a wide range of colour spaces.
- Create and manipulate by mixing, lightening, darkening and saturating colours.
- Support for colour harmony generation, including complementary, split-complementary, and triadic harmonies.
- 437 stars on GitHub

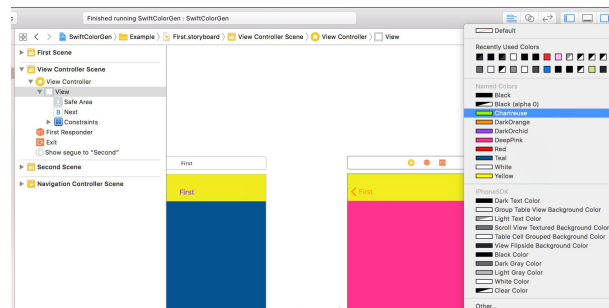


<https://github.com/dialex/JColor>, used under MIT license.

# SwiftColorGen (Swift)

<https://github.com/fernandodelrio/SwiftColorGen>

- Generate colour palettes based images, colours and gradients.
- Support for export to a variety of formats, including UIColor and NSColor.
- 150 stars on GitHub



<https://raw.githubusercontent.com/fernandodelrio/SwiftColorGen/master/Resources/Storyboard0.3.0.png>, used under MIT license.

<b>Library</b>	<b>Programming Language</b>	<b>Key Features</b>
Chroma.js	JS	Wide range of colour spaces; colour manipulation; interpolation.
AC Colors	JS	Colour generation; colour space conversion.
Colorama	Python	ANSI escape code support; cross-platform; easy terminal colouring.
Colorful	Python	Wide range of colour spaces; colour manipulation; harmony support.
JColor	Java	Colour manipulation and conversion library.
SwiftColorGen	Swift	Colour palette generation for iOS and macOS apps; export to UIColor and NSColor.





Questions?