Colour in Data Visualisation

A Survey on How Colour Can Be Utilised in Data Visualisation

Inge Gsellmann, Michael Hebesberger, Danijela Lazarevic

706.057 Information Visualisation 3VU SS 2023
Graz University of Technology

15 May 2023

Abstract

Choosing appropriate colours can be critical to the success of a data visualisation. This survey gives an overview of what colour is, how it can be used in data visualisation, and what criteria are important when choosing colours. It then provides an overview of the most common colour libraries in different programming languages, followed by an overview of different online tools which help with colour palette generation, including additional features like colour blindness simulation or calculating pleasant looking yet still distinguishable colour palettes. This survey focuses on free and open-source tools. Tools which have no freely available funtionality were not included. Finally, the authors give their personal conclusions about the examined libraries and tools.

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

Colour is a powerful tool in data visualisation. Humans perceive colour before noticing an object's shape or details [Dabner 2006, page 38]. It has the ability to convey meaning, highlight important information and make data more engaging and visually appealing. However, choosing the right colours for a data visualisation is not always straightforward.

One of the key factors to consider when choosing colours for a data visualisation is accessibility. Colour blindness affects approximately 1 in 12 men and 1 in 200 women globally, meaning that many people may struggle to interpret visualisations which rely solely on colours. It is important to ensure that visualisations are designed in a way which is accessible to all users, including those with colour blindness. This can be achieved by using colour schemes which have high levels of contrast and avoiding the use of colour alone to convey meaning.

Another important factor to consider when choosing colours for data visualisation is cultural significance. Colours can have different meanings and interpretations in different cultures. For example, love is associated with the colour red in many Western cultures, while it is associated with blue in African cultures. It is important to consider the cultural context of the visualisation and choose colours which are appropriate and meaningful in that context.

In addition to these factors, there are also aesthetic considerations when choosing colours for data visualisation. Colours can be used to create visual appeal and engage users with the data [Healey 1996; CBA 2023; Muth 2018].

In order to tackle all those important facettes to colours and choosing the right palettes, there are many different tools which help users. Some focus more on presenting visually pleasing palettes, while others focus more on making colours more accessible for everyone.

Chapter 2

Colour Basics

This chapter will briefly outline the basics needed for talking about colour constructively. Starting with a simple summary of what colours are, this chapter then describes how colours are represented on a computer and what problems arise there.

2.1 Light and Vision

There would be no vision without light, as the human vision is reliant on the presence of light. The light humans can see is a type of electromagnetic radiation, which can be detected by t he human eye. The wavelength of this radiation humans typically are able to see is around between 400 to 700 nanometers, with the shorter waves appearing violet and the longer ones red [Alsaleem et al. 2020]. All other visible colours appear within these two extremes and make up the visible spectrum of light, as can be seen in Figure 2.1.

The human eye has a remarkable ability to distinguish between colours. Scientists estimate, that humans can see up to 10 million different colours. This is due to the presence of specialised cells in the retina, known as cones. When light enters the eye, cones detect the light, translate it into electrical signals, which are sent to the brain, where they are interpreted as colours. There are three types of cones, which are each sensitive to different wavelengths corresponding to the colours of red, green and blue (RGB) [Spring et al. 2023; Mukamal 2017].

2.2 Colour Models

Colour models are used to represent and organise colours in a digital format. They are based on mathematical models and describe colours in a way that humans recognise and visualise the colour through attributes such as hue and brightness. Colour models can be represented in different ways, such as for colour palettes or gradients. They are essential tools for artists, designers, developers and everyone who works with colours on different devices. In data visualisation, colour models can be used to represent different data points or categories, as will be further explained in Chapter 3.

Colour models can be categorised based on their applications: device-oriented, user-oriented, and decice-independent. Device-oriented colour models relate to and are affected by the signal and tools used for displaying, while user-oriented colour models enable users to describe and approximate what they perceive. Device-independent colour models are not affected by device properties and are useful in network transmission where visual data needs to traverse through different hardware devices [Ibraheem et al. 2012]. In the following sections the most common colour models are named and explained.

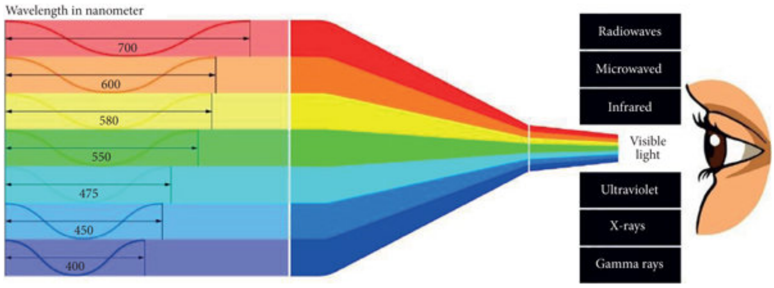


Figure 2.1: Colours humans see based on the wavelength in nanometers. [Image taken from Alsaleem et al. [2020]. Used under the terms of CC BY 4.0.]

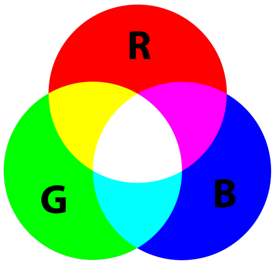


Figure 2.2: Representation of the RGB colour model. [Image taken from Ferlixwangg [2017]. Used under the terms of CC BY-SA 4.0.]

2.2.1 RGB

The RGB (Red Green Blue) colour model is an additive colour model, which means colours are created by adding different colours together. In the RGB colour model colours are created by combining different intensities of red, green, and blue light. The amount of light emitted by each colour is measured on a scale from 0 to 255, with 0 representing no light and 255 representing the maximum amount of light. When all three primary colours are added at full intensity, white light is produced. This can be seen in Figure 2.2.

The RGB colour model is widely used in various industries, including graphic design, web design, photography and video production. It is the standard colour model for creating digital images and is supported by most software applications. It is also commonly used in displays, cameras, and scanners.

Although the RGB colour space is very popular, it might not be the best choice to use in data visualisation, due to the fact that RGB colours are not transferable between devices and the visually perceptive difference between colours is non-linear [Ibraheem et al. 2012; Pedamkar 2023].

2.2.2 CMYK

The CMYK (Cyan, Magenta, Yellow, Key) colour model is a subtractive colour model. In the CMYK colour model, colours are created by subtracting different amounts of cyan, magenta, and yellow from white. The Key colour, also known as black, is added to enhance contrast and provide a true black colour. The model can be seen in Figure 2.3.

The CMYK colour model is widely used in the printing industry, for example for commercial printing, newspapers, magazines, and packaging. It is also used in home and office printers. The CMYK colour model allows for the creation of a wide range of colours, but it is not as vibrant as the RGB colour model, which is used for digital displays. The CMYK colour model has similar disadvantages to the RGB colour model [Ibraheem et al. 2012; Olesen 2023].

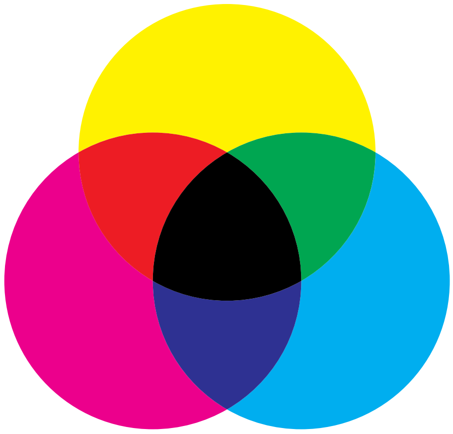


Figure 2.3: Representation of the CMYK colour model. [Image taken from Abdelhamed [2023]. Used under the terms of CC BY-SA 4.0.]

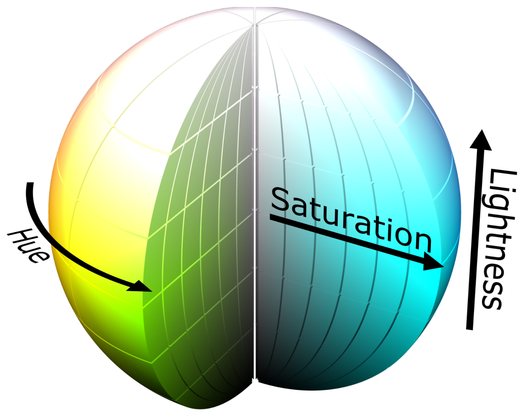


Figure 2.4: Representation of the HSL colour model. [Image taken from SharkD [2009]. Used under the terms of CC BY-SA 3.0.]

2.2.3 HSL and HSV

The HSL (Hue, Saturation, Lightness) and HSV (Hue, Saturation, Value) colour models are both cylindrical coordinate systems which represent colours in a way that is more intuitive to humans than RGB or CMYK. In the HSL colour model, hue represents the colour's position on the colour wheel, saturation represents the intensity of the colour, and lightness represents the brightness or darkness. In the HSV colour model, hue also represents the colour's position on the colour wheel, saturation represents the amount of grey in the colour, and value represents the brightness. An illustration of HSL can be seen in Figure 2.4.

The HSL and HSV colour models are commonly used in colour picking tools, where users can select colours by adjusting the hue, saturation, and lightness or value values. They are also often used in graphic design, computer grahics, and web development. Similar to RGB and CMYK, these colour models are also not device independent [Ibraheem et al. 2012; Hastings and Rubin 2012].

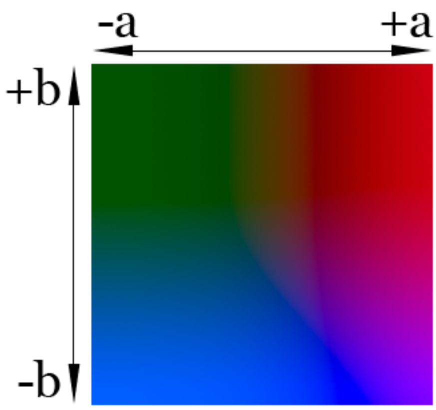


Figure 2.5: Representation of the Lab colour model. [Image taken from Voss [2009]. Used under the terms of CC BY-SA 3.0.]

2.2.4 Lab

The Lab colour model is a colour space which describes all colours visible to the human eye. It was developed to provide a more perceptually uniform colour space than RGB or CMYK. In the Lab colour model, L represents the lightness value ranging from 0 (black) to 100 (white), while a and b represent the colour dimensions ranging from -128 to +127. The a axis represents the red-green axis, with negative values indicating green and positive values indicating red. The b axis represents the blue-yellow axis, with negative values indicating blue and positive values indicating yellow. This can be seen in Figure 2.5.

One of the main advantages of the Lab* colour model is that it is device-independent, meaning that colours will be consistent across different devices and printing processes. This makes it a popular colour space for colour management, for example colour correction and matching. It is also commonly used in scientific research, particularly in colour perception and analysis [Ibraheem et al. 2012; Hastings and Rubin 2012].

2.2.5 HCL

The HCL colour model, also known as the hue-chroma-luminance colour model, is a colour space which was developed as an alternative to the HSL and HSV colour models. Unlike HSL and HSV, which are based on the RGB colour space, HCL is based on the CIELAB colour space.

In the HCL colour model, hue is represented as an angle on the colour wheel, while chroma represents the saturation of the colour and luminance represents the brightness, as seen in Figure 2.6. The advantage for data visualisation of the HCL colour model over other colour models is that it provides a more perceptually uniform colour space, making it easier to create colour palettes while still ensuring consistent colour appearance across different devices.

One of the main applications of the HCL colour model is in data visualisation, particularly for creating colour palettes which are easy to read and differentiate. It can also be used in graphic design for creating colour schemes which are accessible for people with colour vision deficiencies or visually pleasing. The HCL colour model is supported by number of programming libraries and tools, allowing programmers to take advantage of HCL's features. It is also supported by some of the most popular image editing softwares, such as Adobe Illustrator and Inkscape [Sarifuddin and Missaoui 2005].

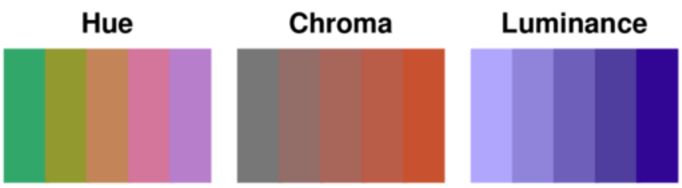


Figure 2.6: Representation of the HCL colour model. [Image taken from Stauffer [2023a]. Used under the terms of CC BY 3.0.]

Chapter 3

Colour in Data Visualisation

Visualising data in a clear and accesible way is an important part of communicating complex information. One of the most powerful tools at a designer or data analyst's disposal is the use of colour. It can evoke emotions, create contrast, and help us differentiate between various elements. This chapter will explore the role of colour in data visualisation and how to effectively use different types of colour schemes to convey information.

3.1 Categorical Colour Palettes

Categorical colour palettes are commonly used to represent different categories of data. This technique assigns different hues to represent each category, allowing viewers to easily differentiate between them. Categorical colour palettes are best suited for depicting data with no inherent order, such as gender, companies, or countries. However, it is important to note that these palettes are most effective when the number of categories is low, typically around 10 categories. When the number of categories is too high, the use of colour can become less effective, and the visualisations can become difficult to read. To ensure readability, it is advisable to choose colours which vary in lightness, enabling visualisations to remain readable in greyscale or for individuals with colour blindness. Categorical colour palettes should only be used when the different colours serve a purpose [Few 2008], as demonstrated in Figure 3.1. In Figure 3.2 the bar chart could be uniformly coloured since the meaning of each bar is written next to it, but the colour aids in differentiating bars with close values (for instance the categories: Ausland, Steiermark, and Wien).

3.2 Continuous Colour Scales

Continuous colour scales are used to depict values which can be ordered, like income, population size, and temperature. These scales use a gradient which transitions from bright to dark, with the bright colour representing low values and the dark colour representing high values, as shown in Figure 3.6. Selecting the appropriate interpolation method for the data is crucial when using continuous colour scales, since it can impact how the viewer perceives the data [Muth 2022d], as discussed in Chapter 3.5. Continuous colour scales are useful when wanting to show outliers, how smooth or abrupt transitions between regions are, or when comparing neighbouring regions to each other [Muth 2022c].

3.3 Diverging Colour Scales

Diverging colour scales share similarities with continuous colour scales, but use a bright middle and darker ends. These scales are typically used to represent data with both negative and positive values. In Figure 3.4, the cold and hot months are more easily distinguishable compared to Figure 3.3, where only one hue is used.

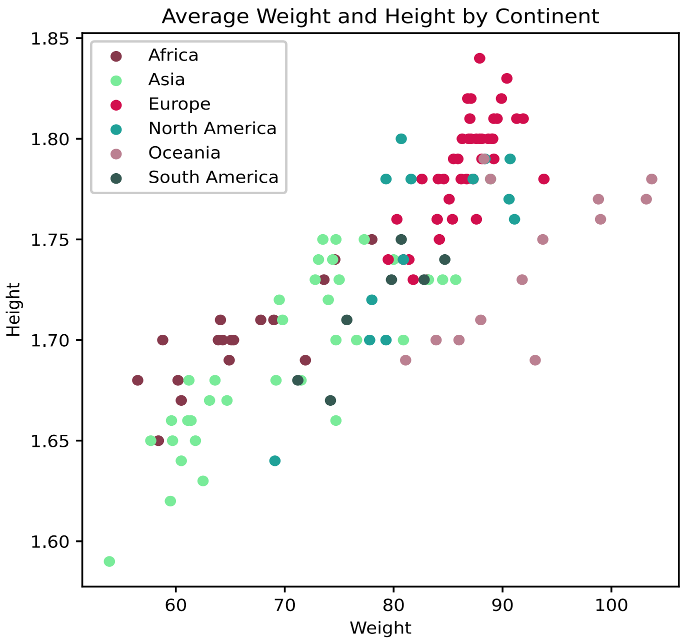


Figure 3.1: Scatter plot depicting the average height and weight by continent. Each dot represents a different country. Colours were chosen using Colorgorical [Gramazio et al. 2017]. [Image created by Danijela Lazarevic using `matplotlib.org`.]

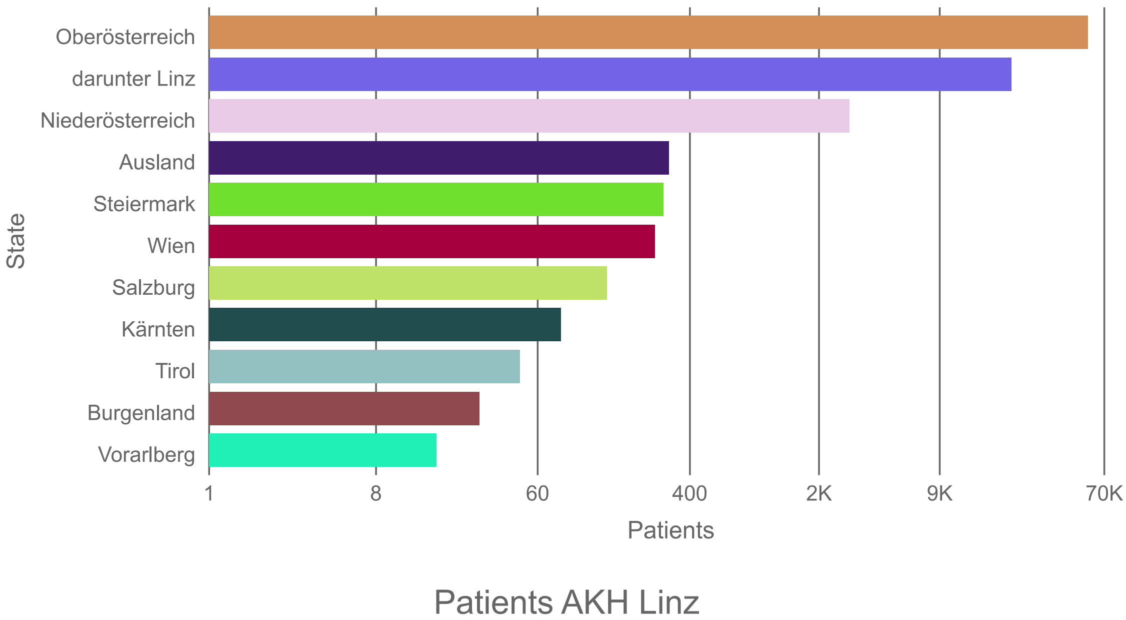


Figure 3.2: Bar chart depicting the origin of patients in LKH Linz using a logarithmic scale. Colours were chosen using Colorgorical [Gramazio et al. 2017]. [Image created by Michael Hebesberger using `chartblocks.io`.]

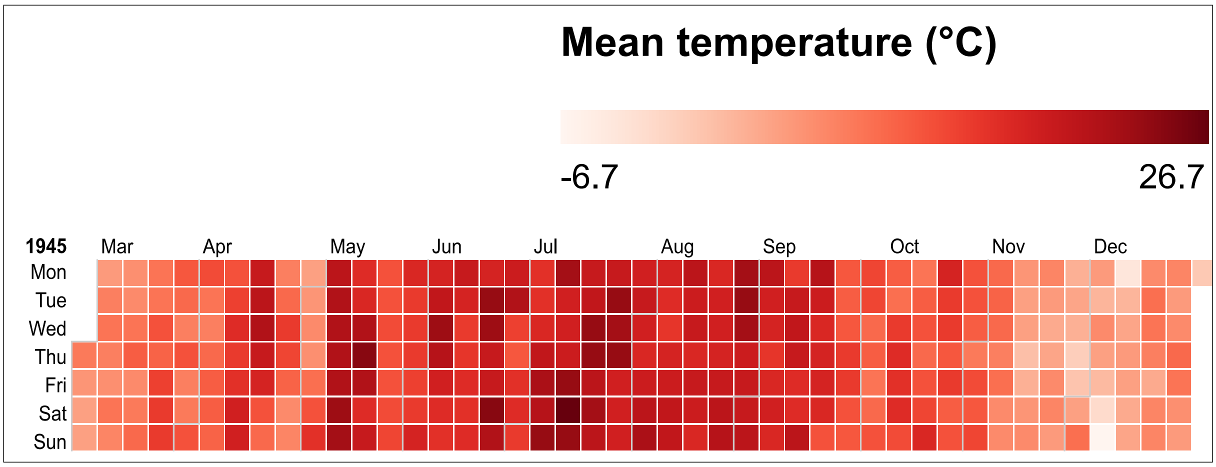


Figure 3.3: Chart showing the mean temperature in Paris 1945 using a continuous colour scale. [Image created by Danijela Lazarevic using `app.rawgraphs.io`.]

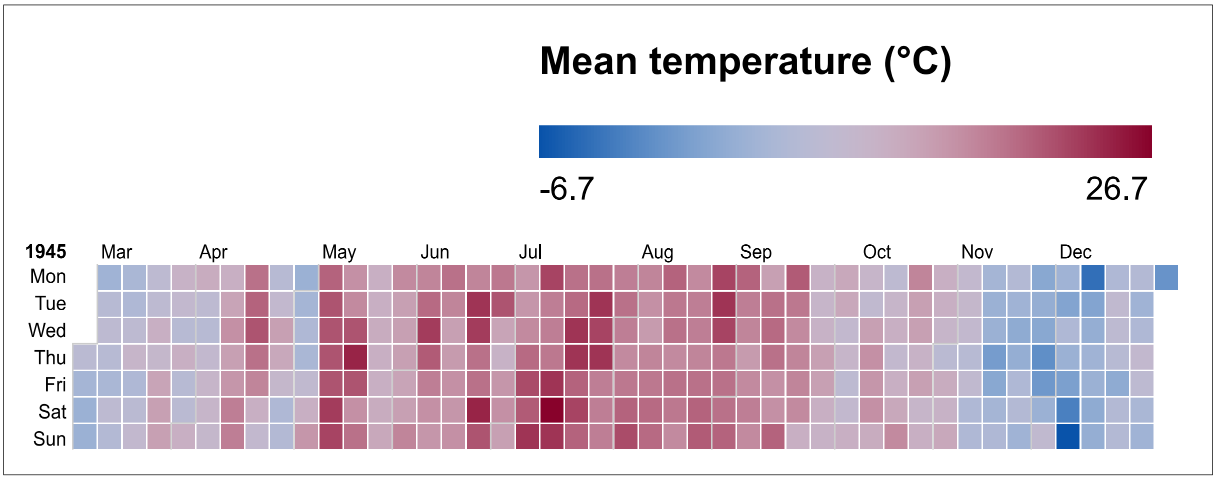


Figure 3.4: Chart showing the mean temperature in Paris 1945 using a diverging colour scale. [Image created by Danijela Lazarevic using `app.rawgraphs.io`.]

3.4 Binned Colour Scales

Binned colour scales make use of a gradient, but instead of a continuous transition, the gradient is split into discrete bins or brackets, as shown in Figure 3.5. These colour scales are useful for representing ordered discrete values, such as grades or clothing sizes. Any continuous data can be binned and thus made discrete. This is useful when the objective is to represent data which is below or above a certain threshold, rather than showing general patterns [Muth 2022c].

3.5 Interpolation

Choosing the right interpolation method for continuous or binned colour scales is important, because it can significantly impact how the data is perceived by the viewer. As evidenced in Figure 3.8b, a poor choice of interpolation can lead to the loss of important information. Therefore, selecting an appropriate interpolation method can enhance the effectivness of visualisations and facilitate the interpretation of data [Muth 2022b].

Three kinds of interpolation are commonly used:

- *Linear Interpolation*: The data is divided into equal bins at equal intervals. This method is used

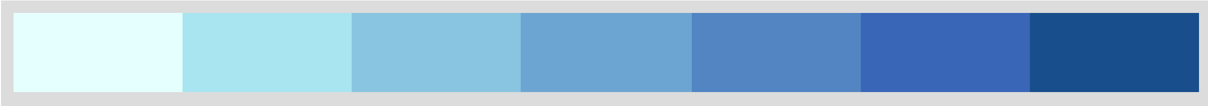


Figure 3.5: Binned colour scale. [Image created by Danijela Lazarevic using `vis4.net`.]

Figure 3.6: Continuous colour scale. [Image created by Danijela Lazarevic using `vis4.net`.]

when the distribution within each bin is assumed to be uniform, as illustrated in Figure 3.8b.

- *Percentiles*: Used when the data is not evenly distributed. Instead of dividing the data into equal intervals, percentiles divide the data based on the distribution of the values themselves, such that each bin contains an equal percentage of the data, as illustrated in Figure 3.7.
- *Natural Breaks*: Used when the data exhibits natural grouping or clustering. Natural breaks divide the data into intervals based on where natural groupings occur in the data, as illustrated in Figure 3.8a.

3.6 Emphasis

Emphasising is a technique used to highlight data which is considered particularly important. As illustrated in Figure 3.9, the red colour used to represent female domestic students stands out in contrast to the grey colour used for the remaining categories, allowing the viewer to easily identify and compare the highlighted data with the rest. In addition to highlighting data, it is also possible to de-emphasise certain categories such as "misc.", "others", or "no data", by using a muted or greyed-out colour [Muth 2022d].

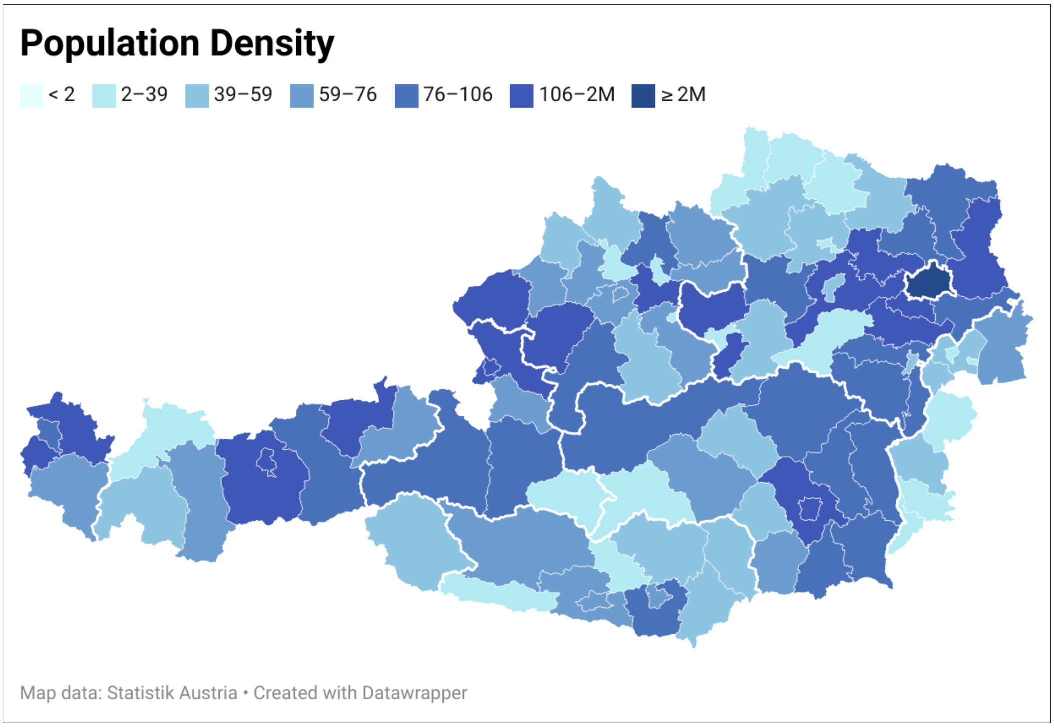
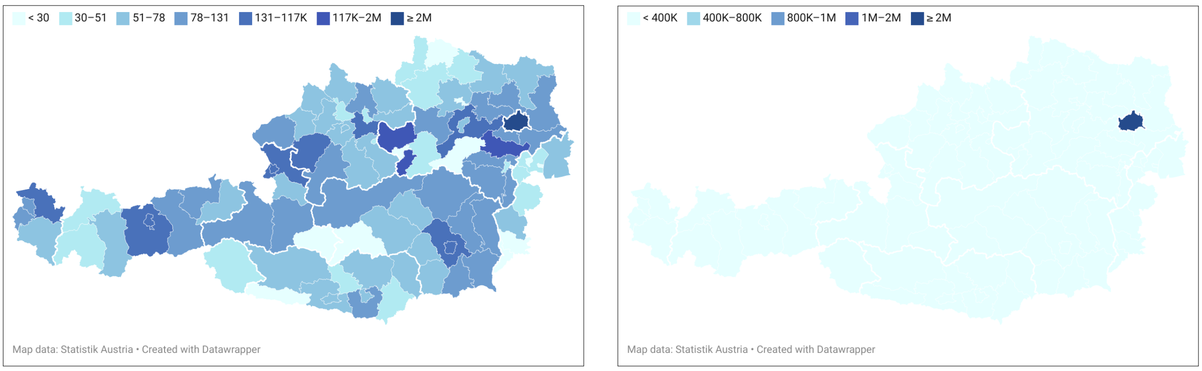


Figure 3.7: Choropleth map depicting population density using a binned colour scale and a percentile interpolation. [Image created by Danijela Lazarevic using `app.datawrapper.de`.]



(a) Natural breaks.

(b) Linear interpolation.

Figure 3.8: Effects of different interpolation methods.

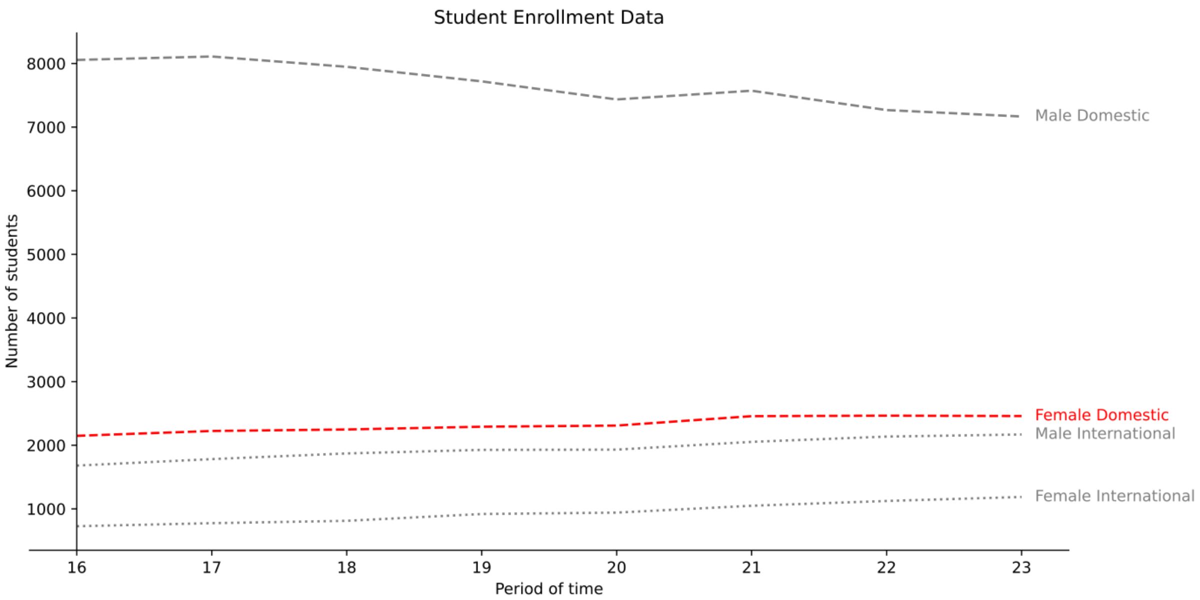


Figure 3.9: Line chart showing enrollment at the Graz University of Technology. One data series (Female Domestic) is highlighted in red. [Image created by Danijela Lazarevic].

Chapter 4

Colour in Culture

Colours serve many purposes in society and always have. Starting from early cave paintings, like those in Figure 4.1, humans have used pigments to create colourful images as a form of self-expression. Over time, humans came to associate colours with concepts, which in turn shaped society. This looks at how colour and culture interact and what aspects of that interaction might be relevant to the field of data visualisation. Failing to consider cultural impact could lead to misleading or poorly readable graphics, while a carefully considered application of colour can synergise with the viewers intuitive understanding.

The topics of this chapter require a chart or visualisation designer to think about their target audience and how it might interact with the graphic. Data visualisation is, after all, a form of communication and knowing one's audience is a vital part of communicating effectively.

4.1 Colour Names

The colour models described in Section 2.2 use numerical values to precisely describe different aspects of a colour. Humans are not that precise and, when talking about colours, will use the hue as a descriptor for a certain colour range. Most people in industrialised cultures utilise only 11 colour words regularly [Gibson and Conway 2017]. In English those would be: black, white, red, green, yellow, blue, brown, orange, pink, purple, and grey.

Evidence suggests that the order in which colour names were developed is the same across cultures [Loreto et al. 2012]. Although the reason for that is unknown, it clearly suggests that colour names are important to language development and do not come about needlessly.

4.1.1 Name Difference

When viewers look at a graph and want to discuss it, they need some way of verbally distinguishing between different elements. Using the colour's name and saying, "Look at the curve of the red line," immediately tells the audience which line to look at, provided it is the only red line.

Name difference is the concept of how different the names of two colours are to each other. Two shades of green might both be named "green", but a light green and a greenish yellow might well be called "green" and "yellow" respectively, even though the actual colours are perceptually similar [Munroe 2010]. When choosing the colours for data visualisation it is important to keep in mind that people will talk about the visualisation. Choosing colours with a large name difference can simplify that task.

4.1.2 Name Uniqueness

When the basic colour words are no longer sufficient, there is a possibility to either add qualifiers such as "light" or "dark" or to use more uncommon words such as "amaranth" or "heliotrope". However, the latter has the difficulty of not being immediately understandable to everyone and also not being as precise.

Figure 4.1: Ancient cave art using different coloured pigments. [Image taken from Mariano [2005]. Used under the terms of CC BY-SA 3.0.]

In data visualisation, it is therefore preferable to use colours which have immediate and obvious names in the languages which are spoken by the target audience.

4.1.3 Colour Names Across Languages

One important aspect of colour-coded graphs and charts is the names for the used colours be available in the target audience's language. As stated earlier, most languages from industrialised countries have 11 basic colour terms. More generally, languages have between 2 and 12 basic colour terms. To properly achieve this, the colour palette might have to be customised to the language in question. Kim et al. [2019] discuss how colour names vary across languages.

4.2 Colour Associations

There are some associations with colours which can serve as a guide when developing a colour palette. Although colour associations might be a good starting point, there are multiple things one should consider before committing to what might seem like the obvious associations.

The first drawback is that by selecting colours, the designer could encode their own biases into a graph. For instance, red is often seen as bad and green as good, so it is important to pay attention to what implications might be made by using these two colours. Section 4.3.1 explains why especially red and green together are a bad choice in terms of a common form of colour blindness.

Colour choices can also reinforce stereotypes such as choosing blue for masculine and pink for feminine data points. It might be a good idea to stay away from these traditional encodings.

Another common association would be for temperature values with a cool colour like blue representing colder temperatures and a warm colour like yellow or red representing warmer temperatures. There are physiological effects to these colour associations. Warm colours can raise body temperature and blood pressure, while cooler colours decrease both [Dabner 2006, pages 33–34]. This form of colour-based synaesthesia also causes dark objects to seem heavier than light objects [Bartel 2003, page 35].

4.2.1 Intangible Associations

McCandless [2010] gives an overview of culture-specific colour associations for a variety of concepts, such as love being variously represented by red (Western, Japanese, Eastern European), green (Hindu), yellow (Native American), and blue (African). Cultural considerations are important if one wants to reach an international audience, since yellow can on the one hand represent health to Chinese people, while at the same time symbolising illness to Japanese and Hindu people.

4.2.2 Tangible Associations

Some colour associations go deeper than abstract concepts. For instance, political parties often have officially assigned colours, just as some cities assign official colours to subway lines. When designing a visualisation of election results one should use the official party colours, since doing differently will only confuse viewers. Similarly, with any chart regarding subway lines, if there is a "Green Line", it only makes sense to represent it with the colour green in a graph.

Other unofficial associations can be just as strong. For instance when representing bananas and cherries in a graph it would be unintuitive to choose anything other than yellow and red respectively. Choosing semantically resonant colours can significantly improve reading time for the viewer [Lin et al. 2013].

4.3 Accessibility

Two considerations are particularly important in terms of accessibility, namely colour blindness and sufficient contrast.

4.3.1 Colour Blindness

According to CBA [2023], about 8% of men and 0.5% of women worldwide are colour blind in some way, a total of around 300 million people. Colour blindness disproportionately affects male people, as the gene for colour blindness sits on the X chromosome.

There are many different kinds of colour blindness. The most commonly known is red-green colour blindness, meaning that people with this condition will perceive red and green light with the same cone cells of the retina and will therefore see them as the same colour. Three forms of colour blindness are illustrated in Figure 4.2.

In order for colour blind people to be able to properly read a visualisation, the colour palette has to consist of colours which are discernible to them. For red-green colour blindness this means not using red and green as category colours or as the diverging ends of a colour scale. While red-green colour blindness is slowly becoming better known to designers, it is not the only form of colour blindness, so designing accessible charts is not as easy as simply leaving out red or green.

There are multiple strategies to find suitable palettes which work equally well for colour blind people. The first is to convert a potential colour palette to greyscales to check whether the contrast between them is high enough. If it is, that is a good indicator that all colour vision impaired people will still be able to differentiate the chart.

Another possibility is to use colour blindness simulators, which take a colour palette and convert it to how it would look to people with different kinds of colour blindness. Many tools already contain these simulators as a feature, as shown later in Table 6.1.

4.3.2 Contrast

Contrast between colours in a graph or chart is important to distinguish different areas or to reference a legend, where present. As people age, their contrast vision deteriorates [Sorensen 2023], so accessible visualisation should take that into account.

Even without vision impairments, it is important to have colours be perceptually different enough to be reliably told apart by viewers. Differentiating shades of light blue is harder than differentiating for instance blue and yellow. Some tools offer contrast checkers to help with this task, as shown later in Table 6.1.

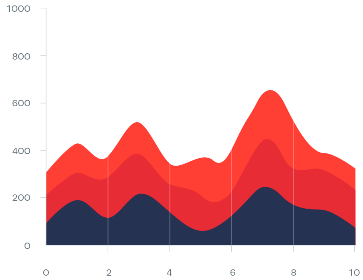
Other factors which greatly affect contrast perception are the size of the coloured area and the distance from other coloured areas. Figure 4.3 shows how, when colours are touching, they are more easily discernible than when there is space between them. The larger the space between two colours is, the

(a) Normal vision. **(b)** Deuteranopia.

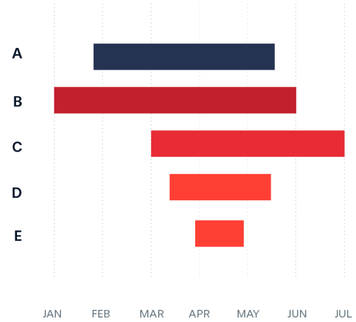
(c) Tritanopia. **(d)** Monochromacy.

Figure 4.2: Three kinds of colour blindness compared to normal vision. [Image (a) taken from Q-lieb-in [2014] and used under the terms of CC BY-SA 4.0. Image (b) taken from Tohaomg [2016a] and used under the terms of CC BY-SA 4.0. Image (c) taken from Tohaomg [2016c] and used under the terms of CC BY-SA 4.0. Image (d) taken from Tohaomg [2016b] and used under the terms of CC BY-SA 4.0.]

more contrast is needed. Similarly, when colours occupy a large area, they are also easier to tell apart than when they only have a very small area [Muth 2022a], as can be seen in Figure 4.4.

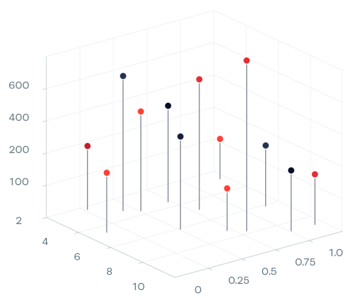


(a) Touching coloured areas.

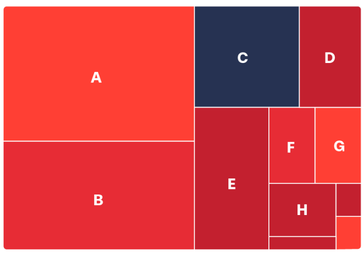


(b) Non-touching coloured areas.

Figure 4.3: Touching colours are easier to differentiate than non-touching colours. [Image (a) taken from ferdio [2023c] and used under the terms of CC BY 4.0. Image (b) taken from ferdio [2023b] and used under the terms of CC BY 4.0.]



(a) Small areas of colour.



(b) Large areas of colour.

Figure 4.4: Small areas of colour are harder to discern than larger areas. [Image (a) taken from ferdio [2023a] and used under the terms of CC BY 4.0. Image (b) taken from ferdio [2023d] and used under the terms of CC BY 4.0.]

Chapter 5

Colour Libraries

Numerous programming libraries and toolkits are available to help developers create visually appealing applications and charts. These libraries provide functions and APIs which make it easy for developers to work with colours, colour models, colour schemes, colour names, and gradients and harmonies. Table 5.1 gives an overview of eight popular colour libraries for their respective programming language. Six of them are then described in more detail.

5.1 Chroma.js

Chroma.js is a JavaScript library which offers a range of colour manipulation tools, colour scales and colour conversion functions [Aisch 2022]. One of its key features is colour manipulation, allowing users to adjust the brightness, saturation, and hue of colours. Chroma.js also makes it easy to blend and mix colours.

Another important feature of Chroma.js is its colour scales. These scales enable users to map data to colours in a few different ways, such as using linear, logarithmic or quantile scales.

Chroma.js provides functions for converting between various colour formats, such as RGB, CMYK, HSL, LAB, and HCL. The library also includes functions for generating colours, such as random colours or colours which match a specific colour scheme. Some example code for manipulating colour with Chroma.js can be found in Listing 5.1.

The library is compatible with many other popular web frameworks and libraries, including D3.js, React, and Vue.js. Despite its many features, Chroma.js is still very fast and lightweight. With more than 9400 stars on GitHub, it is also one of the most popular colour libraries.

5.2 AC-Colors

AC-Colors is a JavaScript library which offers various colour manipulation and conversion tools [Pillai 2021]. With this library users can adjust brightness, saturation, hue, and other properties of colours using its functions. It also allows users to blend and mix colours.

The library supports several colour formats, such as RGB, HSL, and HEX, and provides functions for converting between them. It also includes functions for generating random colours or colours which match specific colour schemes like monochromatic, analogous, or complementary colours. The code for some simple features can be found in Listing 5.2.

AC-Colors ensures that colours are also accessible to people with vision deficiencies by providing functions for simulating colour blindness and checking the contrast ratios between colours. AC-Colors is designed to be fast and efficient. It is also compatible with various JavaScript frameworks and libraries, including React and Vue.js.

Library	Language	Key Features
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 UIcolour and NScolour.
Sass	CSS	Colour manipulation and mixing. Supports sRGB and HSL.
VIPs	C	Image processing library with colour space conversion and manipulation.

Table 5.1: Overview of different colour programming libraries.

```
// create and manipulate a colour
chroma('pink').darken().saturate(2).hex();

// create a colour scale
chroma.scale(['#fafa6e','#2A4858']).mode('lch').colors(6);

// mixing colours
chroma.mix('red', 'blue', 0.5, 'lch');
```

Listing 5.1: Example code for colour generation with Chroma.js.

```
// create a colour
let blue = new Color({"color":"#0000FF","type":"hex"});

// convert between colour spaces
Color.hexToRgb("#552e3a");

//generate a random colour
Color.random();
```

Listing 5.2: Example code for colour generation with AC-Colors.

```
print(Fore.RED + 'some red text')
print(Back.GREEN + 'and with a green background')
print(Style.DIM + 'and in dim text')
```

Listing 5.3: Example code for colour generation with Colorama.

```
print(cf.italic_coral_on_beige('Hello World'))
color_palette_example = {
    'black': '#000000',
    'white': '#FFFFFF',
}
```

Listing 5.4: Example code for colour generation with Colorful.

5.3 Colorama

Colorama is a Python library which provides an easy way to add colour and styling to text output in the terminal by adding ANSI escape codes to their Python scripts [Hartley and Yaari 2022]. With Colorama, users can change the text colour, background colour and text style of their output. It offers a range of pre-defined colours and styles, as well as the ability to create custom colours. The code for this can be seen in Listing 5.3.

Colorama also provides a simple cross-platform solution for working with coloured text in the terminal. It handles Colorama initialisation for different operating systems, so users do not need to worry about compatibility issues between platforms. Another feature of Colorama is its ability to wrap text in coloured boxes or highlight specific sections of text.

5.4 Colorful

Colorful is a Python library which provides a simple and intuitive interface for working with colours [Furrer 2022]. It allows users to create and manipulate colours using a range of colour models, such as RGB, HSL and HSV. Users can perform a variety of operations on colours, such as adjusting the brightness, saturation and hue. It also provides a range of predefined colours and colour palettes. An example can be found in Listing 5.4.

Colorful offers a range of colour conversion functions, making it easy to convert between different colour models. Another useful feature of Colorful is its ability to generate colour schemes based on user-defined inputs. It offers a range of colour harmonies, such as complementary, analogous and triadic colours, which can be useful for creating visually appealing designs.

5.5 JColor

JColor is a Java library which provides a range of colour manipulation and conversion functions [Nunes 2022]. It also allows users to create, adjust and convert colours using a variety of colour models, such as RGB, HSL and CMYK. Users can perform a range of operations on colours, such as adjusting the

```java
System.out.println(colorize("This text will be yellow on magenta",
  YELLOW_TEXT(), MAGENTA_BACK()));
Color color = Color.RGBtoHSB(red.getRed(), red.getGreen(), red.getBlue(), null);

// Create a gradient
Color startColor = Color.RED;
Color endColor = Color.YELLOW;
GradientPaint gradient = new GradientPaint(0, 0, startColor, 100, 100, endColor);
```

Listing 5.5: Example code for colour generation with JColor.

brightness, contrast and saturation. It also offers a range of predefined colours and colour palettes, as well as the ability to create custom colour palettes. Example code with some features of JColor can be found in Listing 5.5.

Another feature of JColor is that it can generate colour schemes based on user-defined inputs. It offers a range of colour harmonies, such as complementary, analogous and triadic colours.

5.6 SwiftColorGen

SwiftColorGen is a Swift library which simplifies the process of managing and generating colour palettes [del Rio 2017]. The library allows developers to define a set of colours in a centralised location, which can be easily referenced throughout the project. This makes it easier to manage colours and ensure consistency across the project.

One of the key features of SwiftColorGen is its ability to generate Swift code for each colour defined in the palette. This code can be used to quickly and easily access the colours from within the project. The generated code is automatically updated whenever the colour palette is updated, ensuring that the colours remain in sync with the rest of the project. SwiftColorGen also has the ability to generate a colour palette preview, which allows developers to view and test the colours in the palette. Generated code from SwiftColorGen can be found in Listing 5.6. The preview can be generated in a variety of formats, including a PDF file or an Xcode playground.

```
extension UIColor {
  // Color #FFC034 (alpha 255)
  static var goldColor: UIColor {
    return UIColor(named: "Gold") ?? .clear
  }

  // Color #8D00FF (alpha 255)
  static var darkVioletColor: UIColor {
    return UIColor(named: "DarkViolet") ?? .clear
  }

  // Color #00FF00 (alpha 255)
  static var myCoolGreen: UIColor {
    return UIColor(named: "MyCoolGreen") ?? .clear
  }
}
```

Listing 5.6: Example of generated code from SwiftColorGen.

Chapter 6

Colour Tools

Colour palette generators are a powerful resource for designers and developers to choose a colour scheme for a project or charts. These tools allow users to generate colour palettes by selecting one or more base colours, some also support adjusting parameters and testing the generated palettes for issues with colour blindness, differentiation, or naming.

One of the main benefits of colour palette generators is that they save time and effort when choosing a colour scheme. Instead of manually testing different colours and combinations, users can quickly generate a variety of options and choose the one which best fits their needs.

Another advantage of colour palette generator tools is that they provide some guidance for users who are not very experienced in working with colours. By exploring different colour palettes, users can gain a better understanding of how the colours interact and how to create appealing palettes [Stone 2006]. Table 6.1 gives an overview of six colour palette generator tools.

6.1 Colorgorical

Colorgorical is a colour palette generator which offers some additional functionality [Gramazio 2017]. It is very easy to use and creates pleasing colour palettes. The user only needs to choose the number of required colours and can then set some parameters in order to create a palette. The first parameter is the perceptual difference of the colours. Perceptually different colour palettes include colours which are easily distinguishable. The name difference parameter instructs Colorgorical to use colours which have different names. The third of the four main paramters is the pair preference. This favours colours which are aesthetically preferable together. These colours are similar in hue, but have different lightness, usually also cooler colours such as green or blue are used. The last main parameter is the name uniqueness. This favours colours which are named uniquely, like red, and ignores others which are not obviously named.

The user can choose a hue filter on a colour wheel to exclude certain hues from the palette and also choose a brightness range to exclude lighter or darker colours. If wanted, the user can add some starting colours, which are then used to calculate the palette based on those starting colours.

After all parameters have been set to their desired values, a colour palette is generated by clicking the Generate button. After a few seconds, the palette and its individual colours are displayed and their corresponding values from four different colour spaces can be chosen. The available colour spaces are Hex, RGB, Lab, and HCL. The values of the colours are provided in an array format, for programmers, or per colour, to copy and paste them into other tools. The tool also provides three example charts which help the user to immediately assess how the generated palette might look in use.

If the generated palette is not perfectly pleasing, but some colours of it appear as a good starting point, these colours can then also be added as starting colours to generate a new palette based on them [Gramazio et al. 2017].

Name	Categorical	Continuous	Pre-Made Palettes	Colour Blindness	Contrast Checker	Licence
Colorgorical	✓	✗	✗	✗	✓	Open Source
Coolors	✓	✓	✓	✓	✓	Paid
Chroma.js Color Palette Helper	✗	✓	✗	✓	✗	Open Source
Encycolorpedia	✓	✓	✗	✓	✗	Free
HCL Wizard	✓	✓	✓	✓	✗	Open Source
Viz Palette	✓	✓	✗	✓	✗	Open Source

Table 6.1: Overview of different tools for colour palette generation.

6.2 Coolors

Coolors is an online platform which allows users to generate and explore colour palettes and is very easy to use [Bianchi 2022]. It supports both categorical and continuous colour palette generation. Coolors also offers a colour blindness simulator and the option to export colour codes as well as the visual palette in various formats such as hexcode, CSS, and SVG.

The platform offers several paid features. These features include the ability to see palette variations and checking the contrast for a chosen palette. Most of the paid features are useful to designers who require visually pleasing colour schemes for their projects.

6.3 Chroma.js Color Palette Helper

This is a small online tool allowing users to design sequential or diverging colour palettes [Aisch 2019]. Users first choose the type of palette they want, together with the number of colours they require. The results are updated in real time. Users then choose their input colours either by selecting a colour on the LSH colour model or by entering a hexcode.

There are also options to correct the lightness of a palette, which means that the lightness values of each colour are made linear to each other, and to perform Bezier interpolation between colours, which smooth out the values of lightness, saturation and hue according to a Bezier curve. These optimisations are illustrated with small charts.

The tool also features a colour blindness checker, which not only simulates three different kinds of colour blindness, but also calculates the contrast of the colours and passes or fails a palette accordingly. Colour palettes can be exported as RGB hexcodes in a variety of array formats, such as with or without commas and parentheses.

6.4 Encycolorpedia

Encycolorpedia is a free online tool for color selection and matching, not strictly aimed at data visualisation, but more broadly at visual design in general [ECP 2023]. It does also have palette generation, which is why it is included in this comparison.

Users start by entering a colour either by colour code or by typing in a name and then going to a colour's page. This page contains variations of the colour (inverted, darker, lighter, closest web safe colour) as well as related colours. Users are also presented with different categorical colour palettes which include the given colour in different colour spaces (CIELHab, CIELHuv, HSL) as well as continuous colour palettes to the complementary colour, to greyscale, to white and to black, again with a choice of colour spaces.

The tool offers colour blindness simulation for 8 different kinds of colour blindness as well as example charts with that colour. Colour values are also shown for almost every colour space. Unique features in this tool are the link to websites, where one can buy paint of the exact same colour as well as a brief gallery of official flags containing the colour (if applicable) and a link to the closest registered brand colour.

6.5 HCL Wizard

HCL Wizard is a web-based platform which comprises three different tools: a palette generator, a deficiency emulator, and a colour picker [Stauffer 2023b]. The platform uses the HCL colour space, which allows users to create colour palettes which are both aesthetically pleasing and perceptually uniform.

The HCL Wizard palette generator supports the creation of categorical, continuous and, diverging colour palettes. It provides several pre-made palettes which can be customised, but does not allow users to add custom colours or modify individual colours in a categorical colour palette. However, when creating continuous or diverging colour palettes, it is possible to change the colours using HCL colour codes.

The deficiency emulator allows the user to upload an image and check it for five different colour deficiencies. The platform offers a severity level for colour deficiency checks, which can be customised to fit the user's needs. The colour picker tool allows the user to pick a colour from a luminance-chroma plane, a hue-chroma plane, or by entering a hex code. The tool then retrieves the HCL code for that colour.

6.6 Viz Palette

Viz Palette is an online tool which helps users choose the right colour palette for data visualisation [Meeks and Lu 2023]. The platform provides nine different visualisations to preview the chosen colour scheme, as well as colour-blindness checks. It is designed to support easy copying and pasting in and out of JavaScript. Viz Palette recommends three tools for colour palette generation, which can be used to create a colour scheme. The generated palette can then be pasted into Viz Palette for the user to see how the colours appear in different visualisations.

Chapter 7

Concluding Remarks

This survey provided an overview of different libraries and tools to generate colour palettes for data visualisation. The choice of colour palette for data visualisation can greatly improve the resulting chart or can completely ruin the visual impact. To choose colours which are not only visually pleasing, but also useful and accessible is not an easy task, especially if the data analyst is not well versed in graphic design guidelines or is colourblind themselves. There are several tools which aim to simplify this task by automating some of the decisions and calculating colour properties based on proven mathematical formulae.

For libraries, our general recommendation is to use Chroma.js, if JavaScript is a possibility, since it provides a great range of features, like colour interpolations, and also supports many colour spaces. Furthermore, it has a large community, so bugfixes and tutorials are easier to find than with other, lesser known libraries.

Regarding tools, all of them provide good functionality in their specific area. Depending on what kind of visualisation is required (continuous or categorical) and whether starting colours are known or not, we would suggest choosing a tool from the Table 6.1 and reading up on the possible choices in detail in their respective sections.

If the aim is simply a pleasant looking visualisation, using one tool will probably suffice. However, if the intended visualisation is critical to its use case, it would be advisible to create another palette option with a different tool. In this case, it might also be prudent to check up on colour blindness simulators and to read over Chapter 4 to avoid cultural blunders.

Bibliography

Abdelhamed, Youssef [2023]. *File:CMYK color model.svg*. 2023. https://commons.wikimedia.org/wiki/File:CMYK_color_model.svg (cited on page 5).

Aisch, Gregor [2019]. *Chroma.js Color Palette Helper*. 2019. https://vis4.net/palettes (cited on page 28).

Aisch, Gregor [2022]. *Chroma.js*. 03 Feb 2022. https://gka.github.io/chroma.js/ (cited on page 21).

Alsaleem, Riyadh Meteab, Baraa Al-Hilali, and Izz Abboud [2020]. *The main colors (RBG) for human vision*. 2020. https://researchgate.net/figure/The-main-colors-RBG-for-human-vision-9_fig2_341488222 (cited on pages 3–4).

Bartel, Stefanie [2003]. *Farben im Webdesign: Symbolik, Farbpsychologie, Gestaltung*. Springer, 2003. ISBN 3540439242 (cited on page 16).

Bianchi, Fabrizio [2022]. *Coolors*. 2022. https://coolors.co/ (cited on page 28).

CBA [2023]. *About Colour Blindness*. Colour Blind Awareness, 2023. https://colourblindawareness.org/colour-blindness (cited on pages 1, 17).

Dabner, David [2006]. *Das Grafikdesign-Buch: Grundlagen und Übungen*. Reprint. Stiebner, 2006. ISBN 3830713134 (cited on pages 1, 16).

Del Rio, Fernando [2017]. *SwiftColorGen*. 10 Dec 2017. https://github.com/fernandodelrio/SwiftColorGen (cited on page 24).

ECP [2023]. *Encycolorpedia*. https://encycolorpedia.com/ (cited on page 28).

ferdio [2023a]. *3D Scatter Plot*. Data Viz Project. https://datavizproject.com/data-type/3d-scatterplot/ (cited on page 19).

ferdio [2023b]. *Gantt Chart*. Data Viz Project. https://datavizproject.com/data-type/gannt-chart/ (cited on page 19).

ferdio [2023c]. *Stacked Area Chart*. Data Viz Project. https://datavizproject.com/data-type/stacked-area-chart/ (cited on page 19).

ferdio [2023d]. *Treemap*. Data Viz Project. https://datavizproject.com/data-type/treemap/ (cited on page 19).

Ferlixwangg [2017]. *File:The three primary colors of RGB Color Model (Red, Green, Blue).png*. 2017. https://w.wiki/6maR (cited on page 4).

Few, Stephen [2008]. *Practical Rules for Using Color in Charts*. Visual Business Intelligence Newsletter 11 (2008), pages 1–13. ISSN 0021-9525. https://perceptualedge.com/articles/visual_business_intelligence/rules_for_using_color.pdf (cited on page 9).

Furrer, Timo [2022]. *Colorful*. 04 Dec 2022. https://github.com/timofurrer/colorful (cited on page 23).

Gibson, Ted and Bevil R. Conway [2017]. *The World Has Millions of Colors. Why Do We Only Name a Few? Cognitive Scientists Suggest That We Name the Colors of Things We Want to Talk About*. Edited by Smithsonian Magazine. 2017. https://smithsonianmag.com/science-nature/why-different-languages-name-different-colors-180964945/ (cited on page 15).

Gramazio, Connor C. [2017]. *Colorgorical*. 2017. http://vrl.cs.brown.edu/color (cited on page 27).

Gramazio, Connor C., David H. Laidlaw, and Karen B. Schloss [2017]. *Colorgorical: Creating Discriminable and Preferable Color Palettes for Information Visualization*. IEEE Transactions on Visualization and Computer Graphics 23.1 (2017), pages 521–530. doi:10.1109/TVCG.2016.2598918 (cited on pages 10, 27).

Hartley, Jonathan and Arnon Yaari [2022]. *Colorama*. 25 Oct 2022. https://github.com/tartley/colorama (cited on page 23).

Hastings, G. D. and Alan Rubin [2012]. *Colour Spaces - A Review of Historic and Modern Colour Models*. African Vision and Eye Health 71.3 (Dec 2012), pages 133–143. doi:10.4102/aveh.v71i3.76 (cited on pages 5–6).

Healey, Christopher G. [1996]. *Choosing Effective Colours for Data Visualization*. Proceedings of Seventh Annual IEEE Visualization'96 (San Francisco, California, USA). IEEE, 1996, pages 263–270. doi:10.1109/VISUAL.1996.568118. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=568118 (cited on page 1).

Ibraheem, Noor A., Mokhtar M. Hasan, Rafiqul Z. Khan, and Pramod K. Mishra [2012]. *Understanding Color Models: A Review*. ARPN Journal of science and technology 2.3 (Jan 2012), pages 265–275. ISSN 2225-7217 (cited on pages 3–6).

Kim, Younghoon, Kyle Thayer, Gabriella Silva Gorsky, and Jeffrey Heer [2019]. *Color Names Across Languages: Salient Colors and Term Translation in Multilingual Color Naming Models*. EuroVis (Short Papers) (Porto, Portugal). The Eurographics Association, 2019, pages 31–35. doi:10.2312/evs.20191166. https://diglib.eg.org/bitstream/handle/10.2312/evs20191166/031-035.pdf (cited on page 16).

Lin, Sharon, Julie Fortuna, Chinmay Kulkarni, Maureen Stone, and Jeffrey Heer [2013]. *Selecting Semantically-Resonant Colors for Data Visualization*. Computer Graphics Forum (Proc. EuroVis) 32.3 (2013), pages 401–410. ISSN 0167-7055. doi:10.1111/cgf.12127. http://vis.stanford.edu/papers/semantically-resonant-colors (cited on page 17).

Loreto, Vittorio, Animesh Mukherjee, and Francesca Tria [2012]. *On the Origin of the Hierarchy of Color Names*. Proceedings of the National Academy of Sciences of the United States of America 18 (2012), pages 6819–6824. doi:10.1073/pnas.1113347109 (cited on page 15).

Mariano [2005]. *File:SantaCruz-CuevaManos-P2210651b.jpg*. 2005. https://commons.wikimedia.org/wiki/File:SantaCruz-CuevaManos-P2210651b.jpg (cited on page 16).

McCandless, David [2010]. *Colours in Cultures — Information is Beautiful*. 2010. https://informationisbeautiful.net/visualizations/colours-in-cultures/ (cited on page 16).

Meeks, Elijah and Susie Lu [2023]. *Viz Palette*. https://projects.susielu.com/viz-palette (cited on page 29).

Mukamal, Reena [2017]. *How Humans See In Color*. 08 Jun 2017. https://aao.org/eye-health/tips-prevention/how-humans-see-in-color (cited on page 3).

Munroe, Randall Patrick [2010]. *Color Survey Results*. Edited by xkcd. 2010. https://blog.xkcd.com/2010/05/03/color-survey-results/ (cited on page 15).

Muth, Lisa Charlotte [2018]. *Your Friendly Guide to Colors in Data Visualisation*. 31 Jul 2018. https://blog.datawrapper.de/colorguide/ (cited on page 1).

Muth, Lisa Charlotte [2022a]. *A Detailed Guide to Colors in Data Vis Style Guides - Datawrapper Blog*. Edited by Datawrapper Blog. 2022. https://blog.datawrapper.de/colors-for-data-vis-style-guides/#big-and-small-areas (cited on page 18).

Muth, Lisa Charlotte [2022b]. *How to Choose an Interpolation for Your Color Scale*. Jul 2022. https://blog.datawrapper.de/interpolation-for-color-scales-and-maps/ (cited on page 11).

Muth, Lisa Charlotte [2022c]. *When to Use Classed and When to Use Unclassed Color Scales*. Jul 2022. https://blog.datawrapper.de/classed-vs-unclassed-color-scales/ (cited on pages 9, 11).

Muth, Lisa Charlotte [2022d]. *Which Color Scale to Use When Visualizing Data*. Jul 2022. https://blog.datawrapper.de/which-color-scale-to-use-in-data-vis/ (cited on pages 9, 12).

Nunes, Diogo [2022]. *JColor*. 29 May 2022. https://github.com/dialex/JColor (cited on page 23).

Olesen, Jakob [2023]. *CMYK Color Model: What Is It and How Is It Used?* 2023. https://color-meanings.com/cmyk-color-model/ (cited on page 4).

Pedamkar, Priya [2023]. *RGB Color Model*. https://educba.com/rgb-color-model/ (cited on page 4).

Pillai, Vinay [2021]. *ac-colors*. 06 Jun 2021. https://github.com/vinaypillai/ac-colors (cited on page 21).

Q-lieb-in [2014]. *File:*Вечір на "інтегралі" - річка Південний Буг*.jpg*. 2014. https://w.wiki/6k9j (cited on page 18).

Sarifuddin, M. and Rokia Missaoui [2005]. *A New Perceptually Uniform Color Space With Associated Color Similarity Measure for Content-Based Image and Video Retrieval*. Proc. ACM SIGIR 2005 Workshop on Multimedia Information Retrieval (MMIR 2005) (Salvador, Brazil). Association for Computing Machinery, New York, NY, United States, 2005, pages 1–8. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=206ca4c4bb4a5b6c7b614b8a8f4461c0c6b87710 (cited on page 6).

SharkD [2009]. *File:HSL color solid sphere elliptical.png*. 2009. https://commons.wikimedia.org/wiki/File:HSL_color_solid_sphere_elliptical.png (cited on page 5).

Sorensen, Tyler [2023]. *Contrast Sensitivity and Aging Vision*. 2023. https://rebuildyourvision.com/blog/vision-conditions/presbyopia/contrast-sensitivity-and-aging-vision/ (cited on page 17).

Spring, Kenneth R., Thomas J. Fellers, and Michael W. Davidson [2023]. *Human Vision and Color Perception*. 2023. https://olympus-lifescience.com/en/microscope-resource/primer/lightandcolor/humanvisionintro/ (cited on page 3).

Stauffer, Reto [2023a]. *HCL Color Scheme*. 29 May 2023. https://hclwizard.org/color-scheme/ (cited on page 7).

Stauffer, Reto [2023b]. *HCL Wizard*. 29 May 2023. https://hclwizard.org/ (cited on page 29).

Stone, Maureen [2006]. *Choosing Colors for Data Visualization*. Business Intelligence Network (17 Jan 2006). https://perceptualedge.com/articles/b-eye/choosing_colors.pdf (cited on page 27).

Tohaomg [2016a]. *File:Deuteranopia sight.jpg*. 2016. https://commons.wikimedia.org/wiki/File:Deuteranopia_sight.jpg (cited on page 18).

Tohaomg [2016b]. *File:Monochromacy sight.jpg*. 2016. https://en.wikipedia.org/wiki/File:Monochromacy_sight.jpg (cited on page 18).

Tohaomg [2016c]. *File:Tritanopia sight.jpg*. 2016. https://commons.wikimedia.org/wiki/File:Tritanopia_sight.jpg (cited on page 18).

Voss, Jakob [2009]. *File:Lab color at luminance 50%.png*. 2009. https://w.wiki/6maQ (cited on page 6).