Importance Driven Color Assignment

Languénou Eric

Importance Driven Color Assignment

Importance Driven
Color-Group Assignment
in Categorical Visualization

Éric Languenou

LS2N
LABORATOIRE
DES SCIENCES
DU NUMÉRIQUE
DE NANTES

Importance Driven Color Assignment

Importance Driven Color-Group Assignment in Categorical Visualization

by

Éric Languenou

Nantes, June 2022

Affiliation Nantes Université, LS2N, UMR 6004, F-44000 Nantes, France
Email eric.languenou@univ-nantes.fr

Cover: color-group color assignment for polygonal map : 52 groups
Style: A proposition for LS2N report latex style
based on TU Delft Report Style, with modifications
by Daan Zwaneveld,
by E.Languenou

Nantes Université

Summary

Recent covid19 pandemia has made data visualization appearing in everyone's everyday life. Besides, ubiquitous digital technology has facilitated the collect of multi-dimensional numerical data that are analyzed by specialists. Their need to explore and to explain these data to non-specialists is important. With categorical data, a lot of diagrams are constructed on a color-coded paradigm associating colors with data classes. Depending on the number of classes or the geometry of diagrams, the class-color assignment choice can become hard a task ($!n$ permutations for n classes).

The goal of this research is to develop an algorithm aiming at assigning the best color, among a user given color palette, for each class of objects in the case of categorical data while optimizing the ability, for a viewer, to distinguish classes geometrical objects one from another. Using the geometry of the displayed object and an estimation of the class' visual separation ability are means to get a better class-color assignment.

The author presents a novel framework that permits optimizing contrast color of classes objects in the case of categorical visualization.

The method relies on a fitness function separation between palette color distances and the concept of *importance factors* expressing the need to get for a couple of objects classes a high color contrast. We construct an *importance factor* using the chosen kind of visualization and especially the data geometrical properties associated. Two matrices are expressed, a first one named *importance matrix* (independent of the chosen color map) and a second, the *color distance matrix* (independent of the data). As, in this research context, the color map is given by the user, once the energy function is expressed by the inner matrix product of the two matrices, a genetic optimization algorithm searches for the best permutation.

We show applications of the concept to several kinds of categorical visualizations: streamgraphs, line charts, polygonal maps, chord diagrams. For each visualization kind, we show resulting color assignments. Streamgraph diagram and chord diagram contrast driven color-group assignment has not been explored in literature so far.

Nomenclature

Symbols

Symbol	Definition
n	the number of groups in the categorical visualization
M	the importance matrix
$M(l, m)$	the element of importance matrix coding the importance for group i and group j
G	the list of group names
g_i	the name of group at index i
C	the colormap
$(C_1, .., C_n)$	the n colors of the colormap
D	the DE2000 color distance matrix
$D(l, m)$	the DE2000 color distance between colors C_l and C_m
p	a permutation $(p_i, ..., p_n)$
P	the matrix coding the permutation frame change
$A(p_i, ..., p_n)$	the adequacy of permutation $(p_i, ..., p_n)$

Streamgraph

Symbol	Definition
$\gamma_t^{i,j}$	neighborhood descriptor ($\in \{0, 1\}$) between groups i and j for value t
$\delta_t^{i,j}$	elementary importance factor between i and j groups for value t
h_t^i	height of layer i for t value

Polygonal Map

Symbol	Definition
E^i	the edges of the categories i which are frontiers with another category
P_t^i	point at index t along the edges E^i separated by an increment value between successive points
$step$	a user value coding the distance between two points along the edges
nb^i	the number of points along the edges of category i
$dist_i(P_t)$	the distance traveled within the polygon i from the point P_t perpendicularly to the current polygon edge
$\gamma_t^{i,j}$	a neighboring factor equal to 1 if category i and j share a polygon edge for t index
$\delta_t^{i,j}$	the elementary importance for point P_t^i
$\gamma^{i,j}$	neighboring factor equal to 1 if category i and j share a polygon edge
$area_i$	the area of polygon i

Chord Diagram

Symbol	Definition
$F(i,j)$	flux from group i to group j
S^i_{in}	sum of fluxes arriving to group i
S^i_{out}	sum of fluxes starting from group i
O^i	the decreasing ordered list of the group index of incoming flux for group i
nb^i_{in}	the number of incoming flux for group i
$\gamma^i_{q,r}$	neighborhood descriptor that codes the fact that fluxes coming from group p and from group q share a frontier on group i
M^i	the elementary importance matrix for group i
$m^i(q,r)$	the elements of the elementary matrix M^i

Line-graph

Symbol	Definition
nbp	the number of values on x
(x^i_t, y^i_t)	the coordinate of point $t \in [1, nbp]$
$percent_y$	the percentage of the y interval for which two lines are considered as too close one from another
$nbProximity^{i,j}$	the number of t values for which line i and line j are too close
$nbLowAngleCrossing^{i,j}$	the number of t values for which line i and line j are intersecting with a too small angle

Multiple Figures Importance

Symbol	Definition
G^j	the list of group names for figure j
g^j_i	the names at index i in G^j
$G^\cup$	the union set of all G^j
$G^\cap$	the intersection set of all G^j

Contents

1

Introduction

Recent covid19 pandemia has made data visualization appearing in everyone's everyday life. The ubiquitous digital technology has facilitated the collect of multi-dimensional numerical data that are analyzed by specialists. Their need to explore and to explain these data to non-specialists is important. With categorical data, we construct a lot of diagrams on a color-coded paradigm associating colors with data classes. Depending on the number of classes or the geometry of diagrams, the color assignment choice can become hard a task.

The goal of this research is to develop an algorithm aiming at choosing the best color, among a user given color map, for each class of objects in the case of categorical data while optimizing the ability, for a viewer, to distinguish classes geometrical objects one from another. Using the geometry of the displayed object and an estimation of the class' visual separation ability are means to get a better class-color assignment.

We can evaluate the distinguishability of two colors using a color distance which quantify how much a color differs from another color. From simple euclidian RGB distance to "Lab" color space expression, various distances have been proposed in the literature aiming to express the perception of color difference or similarity. The recent developments have come with validity for the whole color cube of colors and the color distance DE2000 [SWD05] proposed by the CIE (Commission internationale de l'éclairage) possesses almost all the mathematical properties of a distance, though being not exactly symmetrical.

Let us recall that, for a diagram showing n classes and n colors, color-class assignment consists in choosing a bijection from the class index set $\{1, ..., n\}$ to the color index set $\{1, ..., n\}$. A permutation of size n can model this and the number of permutations for a set of size n is factorial n ($!n$) which explodes with the increasing of n.

The idea behind the presented framework is to decompose an expression of a good class-color assignment in a geometrical information on one part, and a color information on the other. The author assumes that a contrast driven function can be expressed for two objects, which represents, given the object geometries and their distance, size or interface/border/common edges, the need to set colors with a large color difference in order to distinguish the two objects. We denote this "need" factor the *importance factor*.

A second assumption comprises stating that we can express the distinguishability between two colored objects by the product of this *importance* coefficient and the distance associated with the two colors.

1.1. Research Context

This section presents the context where the need for color-class assignment happen.

1.1.1. The Need to Separate Layers

Several years ago (2018), when the author was developing a software to help music scientists, the need to have a visual separation between groups on categorical data visualizations occurred. Before, we had

developed a first attempt to visualize the personal music library as a music archipelago metaphor[1] [LKG15]. A survey data collection on music listening habits which produced multi-dimensional temporal data was available and gaining comprehension of the general shape of the data was, for the music scientists, mandatory.

The chosen way was to let the user interact with data properties by filtering some values, followed by an evaluation of the filtered data shown in a two-fold visualization, in a fast interactive exploration.

1.1.2. StreamFacet

The research led to a software called "Streamfacet", [LK21] displayed on figure 1.2, based on an *Elastic Facet* paradigm developed by M. Stefaner *et al.* [SM07; SUS08] (see figure 1.1) and Streamgraph diagrams published by Havre *et al.* [Hav+02] in 2002 and popularized by Lee Byron and Martin Wattenberg [BW08] in 2008. A video of the application in use is visible on vimeo[2] (commentary in French).

We composed Streamfacet of two parts, the top part where the encountered values are shown as labeled buttons displayed in columns corresponding to data dimensions. Elastic Facet relates the height of each button to the label number of occurrences in the data. By clicking a button, the associated value becomes a filtering value, and it changes all columns according to the new filtered data.

The bottom part displays a streamgraph for which the grouping dimension (y) and the time dimension (x) are selected within the elastic list top part. While exploring by filtering and displaying streamgraphs, the number of layers varies according to filters and could be high. Even in such cases, the user wants to be able to visually separate layers one from another, even if layers' comparison and diagram understanding is almost impossible. The class-color assignment has therefore to be automatic.

For categorical visualizations where colors do not convey other database information than category, and in order to avoid favoring specific groups, the color map must be neutral in term of brightness and saturation. A widely used solution to optimize layers' visual separation is to choose a color map with a large hue interval as in a rainbow map, even if this guaranties legibility only for a rather small number of layers. A great thickness variation between displayed layers may happen, leading to difficulty in visually separate layers. Another difficulty lies because a layer may have many neighboring layers. In addition, using a rainbow color map usually leads to lesser aesthetical diagrams.

Some sentences extracted from the 2008 article from Byron & Wattenberg [BW08] show that the distinguishability aspect was noticed from the very beginning: "*A third issue is the ability of a reader to distinguish effectively the many layers of a stacked graph (E)*" "*There must be enough local contrast between layers in order to differentiate each layer, a particularly important issue as raised in design issue (E)*".

1.1.3. An automatic Color-class Assignment

The streamgraph display context is not a case where a data scientist or a graphic designer searches for a publication single streamgraph appearance in order to convey information and therefore has a lot of time to explore manually many colors assignment to streamgraph layers. While exploring data through the use of Streamfacet, the user filters data, generates streamgraphs according to dimensions. Consequently, there is a need for an automatic colors-layer assignment. In 2018, the author devised an optimization process (not published then) in order to assign layers to colors in streamgraph visualization. It appears afterward that the concept could be extended to other categorical visualizations.

1.2. Importance, Assignment Optimization and Restrictions on Space Research Dimensions

Searching all together the colors to be used in the diagram and the assignment of colors to the depicted groups leads to a complex optimization problem with constraints. For example, if n is the number of groups, the number of unknown variables will be $3 \times n$, corresponding to RGB color values. To prevent some groups from focusing viewer attention, the palette colors must be "similar" in term of brightness and saturation while being sufficiently different, adding therefore additional constraints.

One way to reduce the problem complexity is to consider the color map as a user given data, limiting then the search for group-color assignment. In addition, the author believes that data scientists have preferences about color map.

[1] vimeo.com/ericlanguenou
[2] www.vimeo.com/ericlanguenou

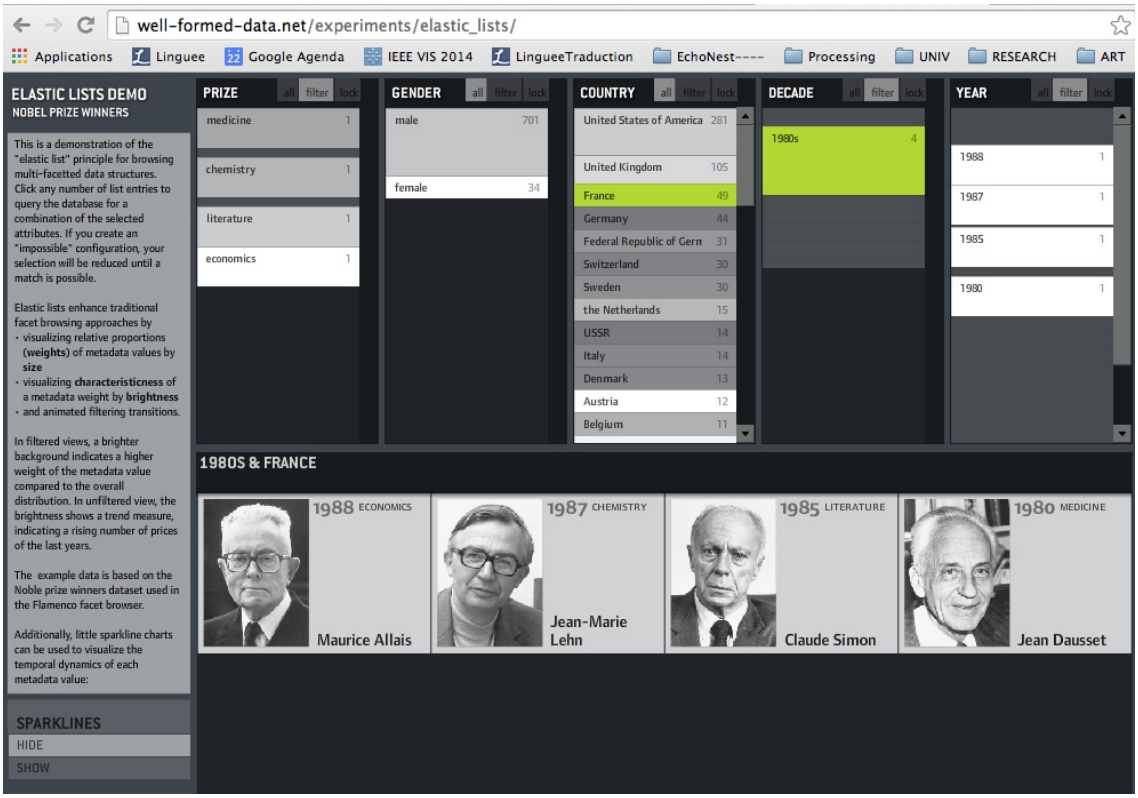


Figure 1.1: Elastic Lists by Moritz Stefaner

As DE2000 color distance is available in a lot of package, transforming diagrams geometrical data into *importance factors* was to be devised together with an evaluation of a good color assignment before choosing an assignment optimization process.

1.3. Problematic

It's useful to list what this presented research is about and what it is not.

This research is not about:

- an attempt to choose the color palette for a given visualization. We let this part to the user;
- an attempt to help the user choose a color assignment which satisfies absolutely the whole set of data visualization constraints;
- a user test analysis to compare the presented method to other visualization color assignments.

This research is about:

- a novel automatic color-class assignment method driven by class color contrast, based on class object geometrical properties and shown data;
- a novel class-color assignment framework in which a lot of categorical visualizations could be integrated;
- the presentation of the framework diagram applications to:
  - streamgraph (new);
  - polygonal map;
  - chord diagram (new);
  - line chart;

The goal is two-folded: first, try to adjust the color contrast in order to distinguish categorical classes with a lot of classes ($n > 15$) and second, give the ability to reduce the color palette hue interval for small number of classes in order to get a more aesthetic figure.

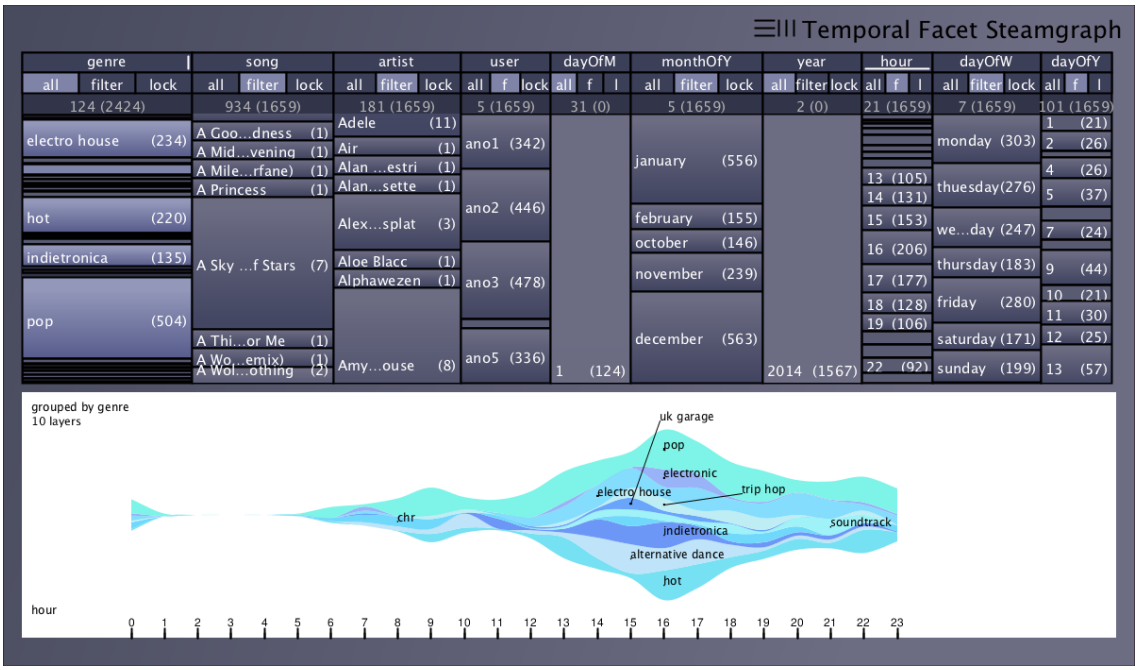


Figure 1.2: StreamFacet using music log

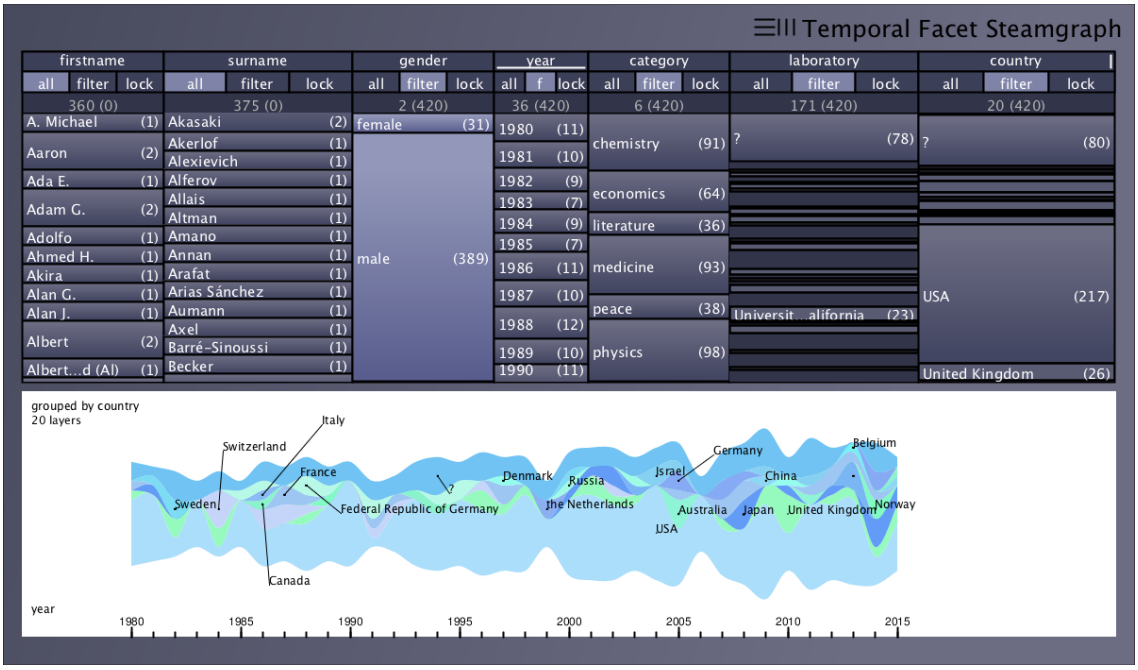


Figure 1.3: StreamFacet using Nobel's prices

1.4. Report Organization

After a short state of the art related to class-color assignment in categorical visualizations in Chapter 2 (page 6), we expose the proposed framework and the corresponding method principles in Chapter 3 (page 10).

Then, the concept of importance matrix, core of our optimization process, is detailed in Chapter 4 (page 13).

After what, in Chapter 5 (page 15), we explain the color-class optimization process and its relation to the QAP problem.

Applications to several categorical kinds of diagram are then listed, with their dedicated importance matrix calculations and results, in Chapter 6 (page 18).

- Streamgraphs in section 6.2 page 19;
- Polygonal maps in section 6.3 page 22;
- Chord diagrams in section 6.4 page 28;
- Scatterplots in section 6.6 page 38;
- Simple neighbor diagrams (single bar chart and pie chart) in section 6.7 page 41;

Then, we detail the extension of the method to class-color assignment with multiple visualizations sharing categories, in the section 6.8 page 42.

Finally, the Chapter 7 page 48 about method implementation follows before a conclusion in Chapter 8 page 50.

2

State of the Art

This chapter presents a state of the art globally limited to published researches about color-class assignment in categorical diagrams, and some related articles.

2.0.1. Colormaps Generation

Even if the presented method is only about class-color assignment using a user chosen color map with categorical visualization, we give a rapid review of color usage and color map related articles here-after.

In image understanding, Bertin [Ber83] wrote that color is associative, selective and ordered. Many works in psychology explore color human response [QH87], especially that color precedes size, shape and orientation. From empirical studies in visualization [War88] to crowd sourced color aesthetics experiments [OAH11; OAH14] and distance perception between visual objects [Bor+11; DBH14], human reception to color have been explored.

Some papers provide guidance on color map design [War12; Tuf90; Bre99; BT07; ZHM09; BRT95]) and some methods emphasis on optimizing for color harmony and aesthetics [Wan+08] while others propose an interactive tool [MSK04] to mix, organize colors and explore color combinations. Language and semantics related to colors have also been explored using automated internet searches for term-color association [Lin+13; SS16; HS12b]. Researchers [SP11] have published works on association of colors like color pair preference in term of aesthetic .

Many online tools like *ColorBrewer* [HB03] help to design palettes for specific tasks. Zhou and Hansen have published a survey on color map design in 2016 [ZH16].

2.0.2. Color in Visualization

Tasks that could be performed about visualization have been systematically studied by Adrienko and Adrienko [AA06; BM13] like localization, identification, or else comparison. These tasks have seen usage in color mapping work by Tominski *et al.* [TFS08].

Mittelstadt *et al.* [Mit+15] describe these three tasks:

- Localization is performed when the user wants to see where specific objects are located within the data.
- Identification is performed when the user browses or explores the data and reads values from color encoded objects on the screen.
- Comparison task is used to compare multiple visual encoded objects, and to perceive and understand the relative and absolute differences.

Given a dataset and a kind of visualization diagram, there are a lot of research about defining rules in order to determine which color maps should be used ([BRT95; BH11; BT07; Bre94a; Bre94b; HB03; Mit+15; Rhe00; RT99; RT96; Saj+12].

Bergman *et al.* [BRT95] developed as well a rule-based approach that relies on the varying sensitivity of the human visual system for spatial frequencies as a basic rule for creating color palettes.

Sanyal *et al.* [San+11] explore the visualization of ensemble uncertainty. They apply a discrete color map by combining three sequential color maps from the "ColorBrewer" tool for rendering spaghetti plots.

Hummel *et al.* [Hum+13] design a 2D interactive color map to separate the joint variance parameter space into similar, dissimilar and uncertain regions to visualize and explore an ensemble of 2D convection flows. The color map enables the user to visualize specific values of interest, localization, and identification at the same time.

In practice, an appropriate color mapping scheme is often obtained by a two-step procedure:

1. selecting a good palette using a categorical color palette tool (e.g., ColorBrewer[HB03] or Colorgorical [GLS17])
2. assigning the selected colors to the classes through a trial-and-error process.

Aupetit *et al.* [AS16] published in 2016 a state of the art on visual class separation measures, applied mostly to scatterplots.

Sedlmair *et al.* [Sed+12] developed a taxonomy of factors that influences the human perception of visual class separation, where most factors are derived from the positions of the data points.

Researches are also concerned with factors like class visibility [LSS13], and perceptual distance [Fan+16].

Fang et al.[Fan+16] proposed to maximize the perceptual distance among a set of given colors while incorporating a set of user-defined constraints. They further compared three optimization algorithms solving this problem and found that a Genetic Algorithm (GA) can ease the effect of sticking to the local maximum.

Hurter *et al.* [Hur+10] proposed an optimization method for assigning colors to lines of a metro map that assigns close routes with the most distinguishable colors. Kim *et al.* [Kim+14] proposed a perception-driven color assignment method for assigning colors to un-ordered image segments, where color aesthetics and contrast are incorporated.

Wang *et al.*[Wan+18] developed an approach in class-color assignment to optimize visual class separability in scatterplot diagrams. Based on forms, distinctness and contrast with background, it considers user color map and optimizes through genetic algorithm. They provide a parameter study with numerical measure.

2.0.3. Analysis

Getting the perfect color samples for a diagram and a set of tasks has been widely explored. Many works in the psychology domain explore color human response [QH87], and state that color precedes size, shape and orientation. The color map has to be chosen in relation to the visualization tasks, which have been systematically studied [AA06; BM13].

Some categorical visualization researches use data graphical properties to optimize color assignment in which some are pixel-based [LSS13; Zen+19] and others are segmented-based or else mark-based [Fan+16; Hur+10; Lu+21; Kim+14; Wan+18]. In order to optimize color assignment, researchers estimated various factors like aesthetics, harmony, class visibility, legibility, color distance, semantics, density, overlapping or fidelity to user preferences. Factors are summed using weights to get an energy formula or a fitness function. Among the encountered optimization methods, authors take advantage of simulated annealing, genetic algorithm, stochastic or gradient-based optimization.

We can find another partition among these works. Some are optimizing for both color map and color assignment [Fan+16; Hur+10; Zen+19] and others are based on a user set color map [Lu+21; Wan+18]. Wang *et al.*[Wan+18] present a general view of existing techniques and introduced their approach using *point distinctness*, non-separability, and genetic algorithm to optimize scatterplots diagrams. In 2021[Lu+21], the algorithm was extended to line and bar charts. Another paper [Hur+10] close to our approach is about air-traffic and metro map diagrams optimization. Their method, besides creating a color map, and changing the diagram object's location, computes distances between lines in order to optimize color differences through a result quality estimation. Lee *et al* [LSS13] introduced the concept of *class visibility* and a color optimization algorithm based on their *class visibility* metric. Via color saturation, their pixel-based algorithm tries to increase the saturation difference for small regions. The *class visibility* factor is performed through integrating of *point saliency*, which measures how much a particular point color differs from its surrounding color.

2.0.4. Related Papers

This subsection presents some related papers and corresponding quick summaries. Papers using pixel-based optimization are not listed.

- **Segmented Images Colorization Using Harmony** Catherine Sauvaget, Stephane Manuel, Jean-Noel Vittaut, Jordane Suarez, and VincentBoyer [Sau+10].

  Computer aided colorization process for artists. Uses a harmony factor based on region and color proportion. Based on user interactions. Dynamic programming. Uses Itten's proportion contrast.
- **An automatic generation of metro-like maps to display flight routes for air traffic controllers: structure and color optimization** Christophe Hurter, Mathieu Serrurier, Roland Alonso, Gilles Tabart and Jean-Luc Vinot [Hur+10]

  The paper presents metro lines and air route placement optimization and class-color assignment. The algorithm tries to assign to close routes the most distinguishable colors. Uses DE2000. They discretize lines into points. They proposed a specific distance $dist$ to evaluate the proximity of two lines. A quality factor measures the assignment between colors and lines:

$$qual = \sum_{i,j,i<j}^{N} \Delta(C_i, C_j) \times 2 \times (1 - \frac{1}{1 + e^{-dist(l_i,l_j}})$$

  They have coded the optimization process using simulated annealing. It is to be noted that the used formula could be imported in our importance matrix.
- **Perceptually Driven Visibility Optimization for Categorical Data Visualization** Sungkil Lee, Mike Sipsand and Hans-Peter Seidel, [LSS13].

  The paper introduces the concept of class visibility and a color optimization algorithm based on a *class visibility* metric. Their optimization process tries to increase the contrast for small regions and to reduce it for larger regions via the saturation component (where our optimization process uses color contrast). *Point saliency* measures how much a particular point has a color different from its surrounding. The applied color distance is the CIE1976. *Class visibility* is obtained by integrating over a region which corresponds to the eye fovea. The size of this region can be parametrised and the estimation is based on pixels (where our method is based on object geometry). Color discrimination is calculated by a repulsive force using color distance.
- **Perceptually-based Color Assignment** Kim Hye-Rin, Yoo Min-Joon, Henry Kang and Lee In-Kwon [Kim+14]

  The paper presents a novel method for automatic color assignment based on theories of color perception. Based on color relationships, as well as the spatial configuration of input segments, the optimization focuses on harmony.

  Three factors are considered: harmony, luminance contrast and respect of additional constraints.
- **Categorical Colormap Optimization with Visualization Case Studies** Hui Fang, Simon Walton, E. Delahaye, J. Harris, D.A Storchak and Min Chen, [Fan+16]

  The paper proposes an algorithmic approach for maximizing the perceptual distances among a set of given colors, avoiding local maximum in the optimization process and losing of the original semantic association. The method applies iterative transformations on a predefined colormap. In addition, they can set user constraints, like providing a constant color for a class, setting a color property fixed range or else choosing an adaptive range that could be used for each iteration. They tested three optimization algorithms, Nelder-Mead simplex, simulated annealing and genetic algorithm. Two case studies are presented, London Tube Map, and scatterplots.
- **Optimizing Color Assignment for Perception of Class Separability in Multiclass Scatterplots** Yunhai Wang, Xin Chen, Tong Ge, Chen Bao, Michael Sedlmair, Chi-Wing Fu, Oliver Deussen and Baoquan Chen. [Wan+18]

  The paper presents an approach in class-color assignment to optimize *visual class separability* in scatterplots. Based on forms, *distinctness*, contrast with background, the method considers a user color map and optimization is performed through a genetic algorithm. They provide results of a parameter study with numerical measure.
- **Modeling Color Difference for Visualization Design**. Danielle Albers Szafir, [Sza18]

  The paper presents crowdsourced studies measuring color difference perceptions for three common mark types: points, bars, and lines, aiming at providing quantitative metrics to help people use

color more effectively in visualizations. They present four mark categories: diagonally symmetric marks, elongated marks, asymmetric marks, and area marks. The method emphasis on separated visual angle and uses a CIELAB Euclidian metric. The authors found that perceptible color differences for lines vary inversely with the thickness in case of line graph visualization.

- **Palettailor: Discriminable Colorization for Categorical Data** Kecheng Lu, Mi Feng, Xin Chen, Michael Sedlmair, Oliver Deussen, Dani Lischinski, Zhanglin Cheng and Yunhai Wang [Lu+21]

  An effective approach for color assignment is presented, that is based on a set of given colors. The method optimizes the perception of scatterplots while considering spatial relationships, point density, degree of overlap between point clusters, as well as background color. It is an interactive color assignment system which incorporates top K suggestions, user-defined color subsets, and classes of interest for the optimization. The system focuses on point distinctness (using DE2000) and contrast with the background.

  The paper presents an integrated approach for creating and assigning color palettes to different visualizations, such as multi-class scatterplots, line, and bar charts, with an emphasis on better visual discrimination of classes. The algorithm customized optimization is based on simulated annealing to maximize the combination of three designed color scoring functions:

  – *point distinctness*: same formula as in "Optimizing color assignment for perception of class separability in multiclass scatterplots" [Wan+18]

  $$\alpha(x_i) = \frac{1}{|\Omega_i|} \sum_{x_j \in \omega_i} \Delta\epsilon(C_r, C_s) g(d(x_i, x_j))$$

  – *name difference*: same expression as in "Colorgical" paper [GLS17], based on minimal Hellinger distance
  – *color discrimination*: based on DE2000 color distance, each two colors should have a minimal DE2000 distance of 10. The method considers the background color. Formula calculates the minimum of DE2000 distance of all couples of the color map.
    Point contrast with background:

  $$\beta(x_i) = \frac{1}{|\Omega_i|} \sum_{x_j \in \omega_i} \Delta L(C_r, C_b) ns(x_i))$$

  They conducted a controlled user study for scatterplots and line charts. The method unifies palette creation and color assignment into a single procedure, which may handle colors semantic.

2.0.5. Contribution

Besides the fact that in our application, we consider adjacencies and sizes, factors like *point saliency* [LSS13] and *point distinctness* [Lu+21], applied to scatterplots, bar charts and line charts distances, could, after re-arranging formulas, result in an *importance matrix* expressed to be integrated in our framework. Hurter *et al.* [Hur+10] proposed an algorithm which optimize objects placement and class-color assignment in which the evaluation of inter-lines distance could also, but only for color-class assignment (our framework does not handle objects location modifications), be transformed to fill an *importance matrix*.

The author's contribution is two-folded:

- providing a general class-color assignment framework (with a user given color map) based on a concept of *importance matrix* which codes the need to associate color contrast to class couples;
- showing applications of the concept to several visualization types: streamgraph, polygonal map, chord diagram and line charts. Some of them (streamgraphs, chord diagrams) being never explored in existing literature.

3

Method Principles

This chapter explains the principles of the energy expression necessary in our optimization process to get a class-color assignment optimization. Let us remind that our importance concept relies on an energy function separated in two parts, a color part, independent of the data and a geometrical one, depending on the visualization type and the data.

Let us consider a categorical visualization displaying n categories/groups/classes $G = \{g_1, ..., g_n\}$.

Visualization specialists have color map preferences which leads the author to choose an approach allowing the user to select a color map, $C = \{C_1, ..., C_n\}$ leaving the optimization process trying to compute the best assignment between object classes and color samples to maximize contrast while preserving distinguishability.

The expression of a fitness function (or energy), which separates color distance values (CIE DE2000 [SWD05]) from an evaluation of the visualization type data-based need to get a color contrast, gives the flexibility to adapt this "need" factor calculation to multiple visualization diagrams. In the following, we denote this evaluation factor as "*importance factor*".

The optimization fitness function for a given color-class association (*i.e.* a permutation) is based on two symmetrical matrices:

- D the *color matrix*, made of $n \times n$ elements, that contains $D(i, j)$ the color distance between colormap samples C_i, C_j;
- M which we call *importance matrix*, also made of $n \times n$ elements, that codes $M(i, j)$ the need to assign a high color contrast between classes i and j.

Once the user has chosen the colormap, the color matrix is constant, and for a given visualization of some given data, the importance matrix is constant as well. We then express the fitness function using the two matrices D and M and a given permutation p.

3.1. Energy Evaluation

If the adequacy $A(i \to k, j \to l)$ of a single assignment of two color samples C_i, C_j to two classes k, l is given by the product of the inter-color distance $D(i, j)$ and the classes couple importance value $M(k, l)$, the overall adequacy for a permutation p represented by $(p_1, .., p_n)$ can be expressed by the sum of the individual products, where the color distance is taken from $D(p_i p_j)$ to handle the color permutation:

$$A(p_1, .., p_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} (D(p_i, p_j).M(i, j)) \tag{3.1}$$

Using matrix notation, we express the fitness function $A(p)$, for a given class-color association p, by a simple *matrix inner product*[1] (denoted $< A, B >$) of the two matrices where the color matrix D is permuted using P_p, the square matrix coding the permutation p mapping.

$$A(p) =< P_p.D.P_p^T, M > \tag{3.2}$$

[1] also called Frobenius product

3.2. Color Distance Matrix

A few decades ago, the *commission internationale à l'éclairage* (CIE) elaborated color distance formulas (metrics) to quantify color difference human perception. Since then, the CIE have devised various distances with constant improvements (CIE74, CIE94, etc.). The formulas give, for two "Lab"[2] coded colors, a factor in evaluating the subjective feeling of human perceived difference among colors.

We use, for the color distance matrix D, the most recent color distance DE2000 (or CIEDE2000) formula provided by the CIE [3]. Using the colormap provided by the user, we fill each element (i, j) of the color distance matrix (which is symmetrical, even if the DE2000 formula is not perfectly symmetrical) with DE2000 distance between color samples C_i and C_j. We show a representation of a color distance matrix on figure 3.1.

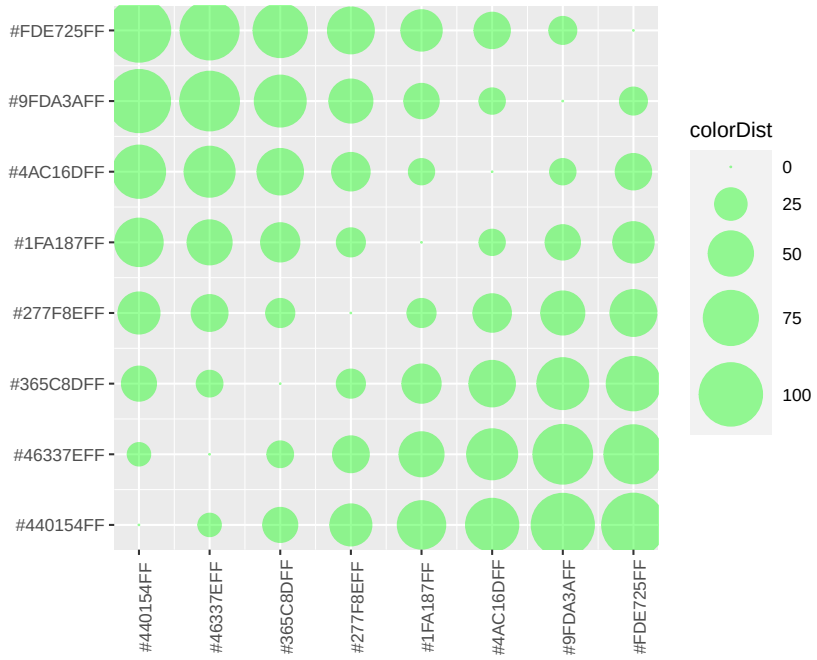


Figure 3.1: Example of Color Distance Matrix

3.3. Importance Matrix

The *importance* matrix M, coding an evaluation of the needed color contrast between two classes, depends on the kind of visualization diagram. The calculations involve graphical object's size, neighborhood, frontier, distance between objects and therefore depend on shown data. We display a representation of an importance matrix on figure 3.2.

We propose several specialized importance matrix evaluation formulas in chapter 6 on page 18 naming streamgraphs, line-charts, chord diagrams and polygonal maps.

The next chapter (chapter 4) introduces the concept and the properties which have to be integrated in the importance factor.

3.4. Energy Estimation Complexity

Thanks to the formulation in two parts (color, geometry) of equation 3.2, the complexity of the energy evaluation is rather low. As the two matrices are symmetrical and the elements for which indexes $i = j$

[2] A specialized color frame
[3] `https://en.wikipedia.org/wiki/Color_difference`

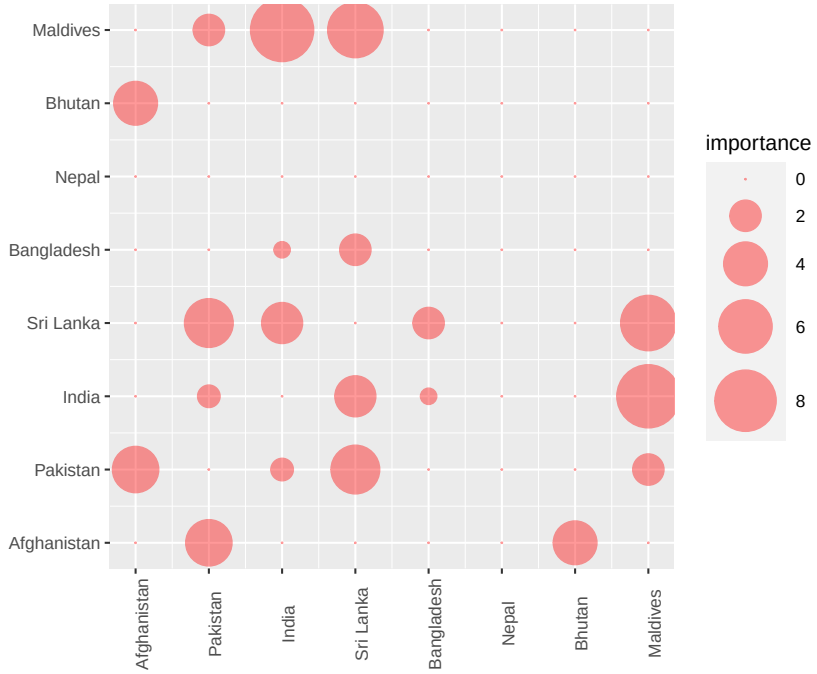


Figure 3.2: Example of Importance Matrix

are null, the complexity of the energy function evaluation, if n is the number of classes, is then reduced to:

- n accesses to the color distance matrix elements corresponding to the current permutation redirection,
- $n(n-1)/2$ multiplications while considering the product of the color distance times the importance factor (but only for nearly half of the symmetrical matrix),
- $(n(n-1)/2)-1$ additions in order to sum the in-between classes' energy contributions.

4

Importance Matrix Computation

The importance factor, as we defined it, is:

- independent of the colormap
- designed for categorical diagrams
- dependent on the graphical properties of the chosen diagram
- dependent on the represented data

In this chapter, we discuss the diagram geometrical properties which have to be handled in the importance factor and we propose diagram specialized importance factor formulas, later in in chapter 6 page 18.

4.1. Visual Properties and Color Separation

Our importance factor is designed to increase the visual discriminability of the classes composing the categorical diagram. Identification capability is let to the user colormap choice, or made possible by labeling the graphical objects. The visual solid angle should be used to express the ability of the viewer to separate one object from another, but for the moment, we calculate the importance factors through object size ratio, therefore independently of the future context of viewing (viewing distance and figure size).

Many graphical objects may compose the class, and they have to be incorporated into the calculus. The eye ability to visually "follow" a continuous graphical object (a line, a layer, a thick arrow, etc.) fosters the visual class separability. The layers of a streamgraph can be either continuous along the time interval, or appearing only on a part of diagram, or else be reduced to a null thickness at a given time and reappearing later.

The object visual separation is critical for adjacent objects to avoid *spreading* (the visual grouping of close objects caused by too much similar colors). In consequence, identifying frontier along adjacent objects is an important aspect.

But we have to reduce the need to assign to two classes two colors, which generate a high color contrast to a single scalar, *i.e.* an element of the importance matrix.

4.1.1. Elementary Importances: Size Matters

Visualizations diagrams are usually composed of several marks (scatterplots, bar-charts) or composed of curves or polygons (stacked-graphs, streamgraphs, polygon maps, chord diagrams). For example, in scatterplots diagram, many discs may represent a class, but in streamgraph visualization, a class' layer may be a single object. In this later case, we decompose the layer into smaller parts to evaluate local discriminability on parts of the class's object.

Let us denote these local evaluations, *elementary importance factors* (or in a shorter way: *elementary importances*).

In case of adjacent class objects visualization, besides the importance of the width and area of the colored region, and when objects discrimination is needed, Newhall (1955) [New55] noticed that the

length of the frontier does not affect the discriminability. It seems that the perpendicular distance from the interface frontier to the other side of the region (a perpendicular thickness) represents a better criterion. The thinner the region, the higher the risk of *spreading* (region visual aggregation) when using similar colors. To the knowledge of the author, there are no published formula, in term of color distance and mutual thickness (or visual angle) that evaluate this risk. Concerning scatterplots, *elementary importances* can involve point density, discs diameter, distances separating discs from one class to discs belonging to another class.

Two kinds of simple empirical tests performed by the author, involving two rectangles filled using two close colors, where:

1. the frontier length is varying (see figure 4.1)
2. the rectangles heights are varying (see figure 4.2)

confirm that the discriminality is affected by height and not affected by the frontier length. It is visible on an electronic version of this report when zooming while being probably too small on a printed version.

Figure 4.1: Varying frontier length

Figure 4.2: Varying rectangles width

The object's visual angle is the main property to be considered, but usually there is no information on the final display size nor knowledge on the distance between the screen (or paper) to the viewer's eye. Some published researches explore discriminability in visualization. Healey and Sawant published a paper [HS12a] about the limits of pixel-based visualizations and visual angles. Maureen Stone explored the spreading effect and distinctness with "tableau" diagrams regarding size [Sto12]. The concept of *Just Noticeable Difference* (JND) [SWD05] has been expressed in order to explore colors which are subjectively considered as the *same*. None of their results apply to our cases.

4.1.2. Compiling Elementary Importances

The case of adjacent objects, as a frontier involves two regions and because the region thickness may vary along the frontier, implies a need to compile or *synthesize* the *elementary importance* factors into a single factor. But *elementary importance* synthesis formulas may depends on diagram type.

4.2. A Matrix to Store Importances

The importance matrix as we define it:

- is symmetrical
- possesses null diagonal values
- could be reduced to a triangular matrix

Depending on the diagram kind, we may associate a null importance value to a classes couple that does not share a frontier. This shortcut is, for example, used in our streamgraph importance calculus.

4.3. Diagram Dependent Formulas

Let us recall that the *importance matrix* codes, independently of the colors, the need for visual separability of object classes using their geometrical properties, *i.e.* shape, neighborhood, area or else distance. In chapter 6, we list considered visual properties and dedicated *importance factor* formulas for steamgraphs, polygonal maps, chord diagrams and line graphs.

5

Color-Group Assignment Optimization

We defined, in previous chapters, the energy (or fitness) expression using an importance matrix and a color distance matrix. This chapter details the energy optimization process that permits finding (or trying to find) the best colors permutation, which gives the highest energy level.

5.1. Permutation Optimization

Various optimization algorithms for colors permutation in the case of categorical visualization have been tested in the literature. Fang *et al.* [Fan+16] experimented Nelder-Mead simplex, simulated annealing and genetic algorithm; they found that genetic algorithm (GA) can alleviate the issue of sticking to the local maxima.

5.2. The QAP Optimization

During the author's search for published existing methods in permutation optimization, it appears to the author that the splitting in two matrices proposed in this report for class-color assignment problem has been published in economics field by Koopmans and Beckmann in 1957 [BK57]. Known as the *Quadratic Assignment Problem* (QAP), it aims at achieving an economic objective of assigning facilities to locations in order to minimize the overall cost. The problem is known as an NP-Complete problem, which can apply to many other optimization problems (backboards, inter-plant transportation, etc.). In 1998, Burkard *et al.* [Bur+98] published a QAP state of the art.

The neos-guide.org [1] formulates the QAP as the following.

Consider the set of numbers $N = \{1, 2, .., n\}$ and the group of bijections between N and itself, defined as $S_n = \{\phi : N \to N\}$. Let F be an $n \times n$ matrix where F_{ij} represents the required flow between facilities i and j. D is an $n \times n$ matrix where D_{ij} is the distance between facilities i and j. We can estimate elementary costs by the product of the distance and the flow between two locations. The goal is then to minimize the overall cost.

A matrix can express a permutation as in a reference frame change [Bur+98]. Each member $\phi \in S_n$ represents a particular arrangement of the n elements so that $(X_\phi D X_\phi^T)_{i,j}$ gives the distance between facilities i and j, when the facilities are permuted according to ϕ. Therefore, individual cost function for transportation between facilities i and j is given by $F_{i,j}(X_\phi D X_\phi^T)_{i,j}$. The sum of each i and j gives the overall cost for a permutation ϕ. Using previous notations, we get the total cost of assigning facilities to locations according to ϕ by the matrix inner product:

$$\langle F, X_\phi D X_\phi^T \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} F_{i,j}(X_\phi D X_\phi^T)_{i,j} \tag{5.1}$$

Even if the QAP energy expression relies on a minimization process, where our class-color assignment formulation expects to maximize the overall color contrast, the two problems are similar. The facilities location distance correspond to the chosen colormap DE2000 color inter-distance and the flow between

[1] `https://neos-guide.org/content/quadratic-assignment-problem`

facilities matches the color contrast importance factor. We can express our class-color assignment problem as a QAP, by inverting the non-null elements of the color distance matrix.

So far, we have not tested QAP corresponding implementation for class-color assignment.

5.3. Algorithms

The ratio between satisfying permutations or quite satisfying permutations and the total number of permutations depends strongly on the data and on the diagram type target.

Sometimes the number of pleasing solutions is high, but for some data and diagram, it can appear that there is not a single satisfying permutation to the problem. For example, it could be easy to construct data for a streamgraph diagram where all layers are neighboring one another with very thin layers. Such data would have no satisfying permutation if the number of layers is high.

But usually, solving this type of problems using genetic algorithm seems to provide the best properties: fast enough, flexible, providing various solutions and being able to be executed on distributed processors.

In the presented framework, we implemented the optimization using genetic algorithm, but ideally, we should test and compare other optimization algorithms.

5.4. Genetic Optimization

In order to get an assignment in a reasonable amount of time, we may use heuristic approaches. In the literature, various methods, simulated annealing, genetic, non-linear gradient, greedy, has been experimented, but when the number of quite satisfying solutions is high enough, genetic approaches provides an advantage as various solutions can be obtained when conducting several optimization processes among which the user may choose.

Genetic optimization (GA) algorithms are inspired by biology, and are based on a representation of the problem's research space using chromosomes. At the beginning, elements (individuals) being generated by random values, compose the population. To calculate the next generation, we evaluate the quality of the individuals using a specialized fitness function. We selected only the best elements to populate the next generation through crossover reproduction between random pairs of these best elements and using random mutations on the chromosomes. The user specifies the size of the population and a number of generations to be calculated. The convergence toward a good solution depends on the energy (fitness) function expression, the chromosomes definition, the size of the population, and some other parameters like the percentage of the population used to transfer their genes, and the percentage of mutations.

With permutation search, the classic genetic optimizer has to be adapted. We code chromosomes using the bijection from initial colors to the object class which may be represented as $(c_1, c_2, ..., c_n)$ where $c_i \in [1, n]$ and $c_i \neq c_j \; \forall i \neq j$.

Crossover reproduction methods must be specialized too. Taking two individuals as parameters, they have to generate valid individuals (permutations). The GA literature has produced many crossover reproduction techniques developed for permutation specialized chromosomes [DW98; US15]. We chose the "*Partially Mapped Crossover*" (PMX) method, among others (see [US15] for a state of the art), in our implementation. We display an example of the PMX method on figure 5.1. Basically, the PMX crossover copies a random chosen interval of the first parent chromosomes, and fills the rest with chromosomes from the second parent.

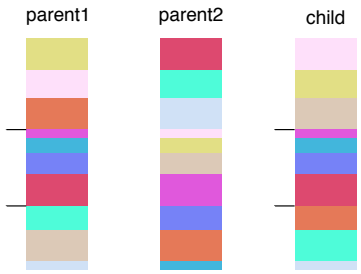


Figure 5.1: Genetic crossover for permutation

Using genetic optimization, we must adapt the mutation process to generate chromosomes which code valid permutations. A swap between two indexes simply performed this.

The author has developed a permutation dedicated genetic package for Java language. And for language R applications, the implemented R6 class is based on the GA package.

5.5. Importance value remapping

Sometimes, the difference between, on one part, between null importance factor and low importance factors and on another part between low and high importance factors, is too small. The optimization process does not converge rapidly enough. A simple linear mapping would not enhance enough the optimization.

We experimented with the use of an importance factor remapping function based on a two-stepped linear function. In order to provide flexibility, the method takes the calculated importance factors interval (zero excepted) as parameters together with a single medium point P located on a percentage of the initial importance interval on x, which is remapped on a percentage of the desired importance resulting interval on y. In the tests, the values 0.6 percent on x and 0.1 percent on y were giving improved results in order to increase high values and clearly separate lowest values from 0 importance value. We display an example of such remapping on figure 5.2.

Moreover, we may set importance factors that are related to classes for which color contrast is unnecessary (0 value), to a penalty value (a negative value). This should increase the convergence but only a few tests has been performed so far. Rigorous tests with running times and results comparison should be run to decide if this option is useful.

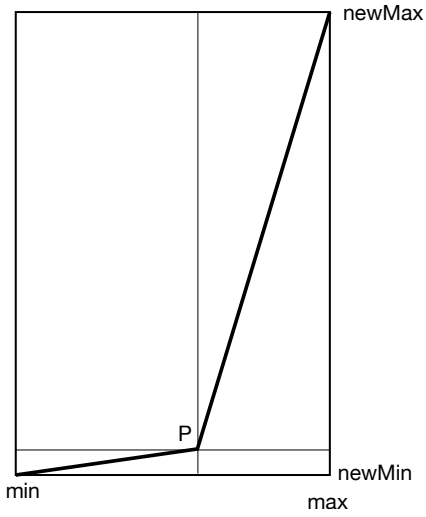


Figure 5.2: Importance value mapping

6

Applications to Categorical Visualizations

6.1. Introduction

This chapter lists several diagrams applications of our class-color assignment method. We organized it as follows. For each categorical diagram, in subsections are explained the diagram graphical rules, the graphical aspects which are handled by the *importance matrix* computation. When it applies, we give *elementary importance* factors formulas too, before providing some explanations on the final *importance factor* calculation (synthesis). Finally, we provide results and show comparison figures with the color assignment corresponding color distances and calculated inter-group importance factors.

The subsections are the following:

- Visualization principles and graphical Rules
- Graphical criteria for color assignment
- Importance factor computation
- Elementary importance factor computation
- Synthesizing to obtain an importance factor
- Results

For each diagram type, three figures are shown:

- using our color assignment optimization;
- using directly colormap index order;
- using random color permutation.;

To visually judge the adequacy of the best color permutation with the calculated inter-class importance factors, the author devised a type of *comparison* figure, dedicated to the color-class assignment problem (see figure 6.7 page 24). This figure presents on the same diagram the importance matrix factors and the permuted color distance matrix. Red discs area is proportional to the importance value and normalized to the maximum importance value encountered within the matrix. Green discs depict the DE2000 color distance normalized to the encountered maximum distance. The more red visible area, the less assignment performance.

We present, for each diagram, specialized comparison figures for:

- diagram with our optimized color assignment,
- diagram using directly the colormap order,
- diagram with a color random order.

The author knows that, in the visualization research domain, it is mandatory to perform user testing, evaluation and comparison with the existing method. Author's lack of time and money explains this absence.

To the author's knowledge, there is no published method to assign color to layers in the *streamgraph* visualization case, nor in *chord diagram* visualization.

The following pages show our framework applications to streamgraphs, polygonal maps, chord diagrams, line charts and some simple neighbor diagrams.

6.2. Streamgraphs

Streamgraph diagrams published by Havre *et al.* [Hav+02] in 2002, were popularized by Lee Byron and Martin Wattenberg in a 2008 New York Times article [BW08]. They are a variation of the stacked area graphs where curves replaces polylines and where layers are organized around a varying central line where stacked-graphs are vertically stacked on the x (time) axis. Since the first scientific publication many improvements have been added, by aggregation [GK15], and notably on the layer order [BH16], about correcting the layer thickness in case of significant slope [Bu+21] and on multi-resolution streamgraphs [Cue+18].

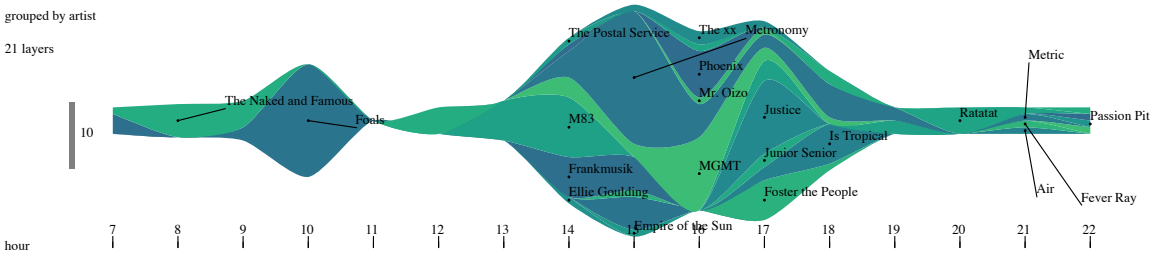


Figure 6.1: Streamgraph with optimized assignment of a partial viridis colormap

6.2.1. Visualization Principles and Graphical Rules

Usually chosen to depict time varying categories, layers height is related to a specific dimension value or a number of occurrences. The layers are stacked on y in constant order along x axis (various orders can be chosen, depending on preferences or data criteria) but layers can present a null thickness for some time intervals. Therefore, layers may share a frontier with many other layers. Layers can be thin on a part of the diagram and thick on another part.

The number of layers, depending on the data, can be huge. Though in this case, legibility is difficult, but in our usage where the streamgraphs are displayed depending on data exploration filters, layers colors must be handled automatically and layer discriminality must be optimized.

6.2.2. Graphical Criteria for Color Assignment

The main visual criterion is concerned with a visual separation between layers. An easy detection of layer borders by the viewer is an important property. As some layer parts may be very thin, it is important that layers' color choice provides enough color contrast to satisfy legibility.

6.2.3. Importance Computation

Consequently, given a layer ordering and depicted data, we must identify the frontier sharing between layers and calculate corresponding importance factors that rely on mutual layers thickness.

6.2.4. Elementary Importance Computation

As layer thickness and frontiers with other layers evolve along the time dimension, evaluating *elementary importance* factors which code x local need to have a large color contrast for two adjacent layers is a first step before compiling, along x interval, a global importance factor.

Let us denote T the set of time samples t used to display the streamgraph composed of n layers for which h_t^i is the height of layer i at time t.

Let us also define $\gamma_t^{i,j}$, a neighborhood descriptor which codes the fact that layers i and j share a frontier on time sample t. This descriptor is a function of data values and of layers chosen order, and return 1 if a frontier exists and 0 otherwise.

If we assume that the thinner the layer, the harder the layer frontier is visible, then a decreasing function would have the desired behavior. We considered that the need to assign high contrasted colors to the layer couple is dependent on the minimum of the two layers thickness (which can be discussed). In absence of knowledge on the human response in such geometrical cases, we can calculate the inter-layer i, j elementary importance factor $\delta_t^{i,j}$, for time t, using the inverse of height:

$$\delta_t^{i,j} = \gamma_t^{i,j} * max(1/h_t^i, 1/h_t^j) \tag{6.1}$$

Then, evaluating for the whole interval T of t values for layers i and j gives an array $\delta^{i,j}$ which has to be reduced to a single factor $M(i, j)$, the importance factor for the couple of layers i, j.

Two figures 6.2 and 6.3 show, for a specialized streamgraph test, a display of elementary importance factors as a red layers. So far, we do not calculate thickness as layer perpendicular size, but we take layer y size, which can give strange results in some parts. This could be enhanced by generating streamgraphs with correct thickness [Bu+21].

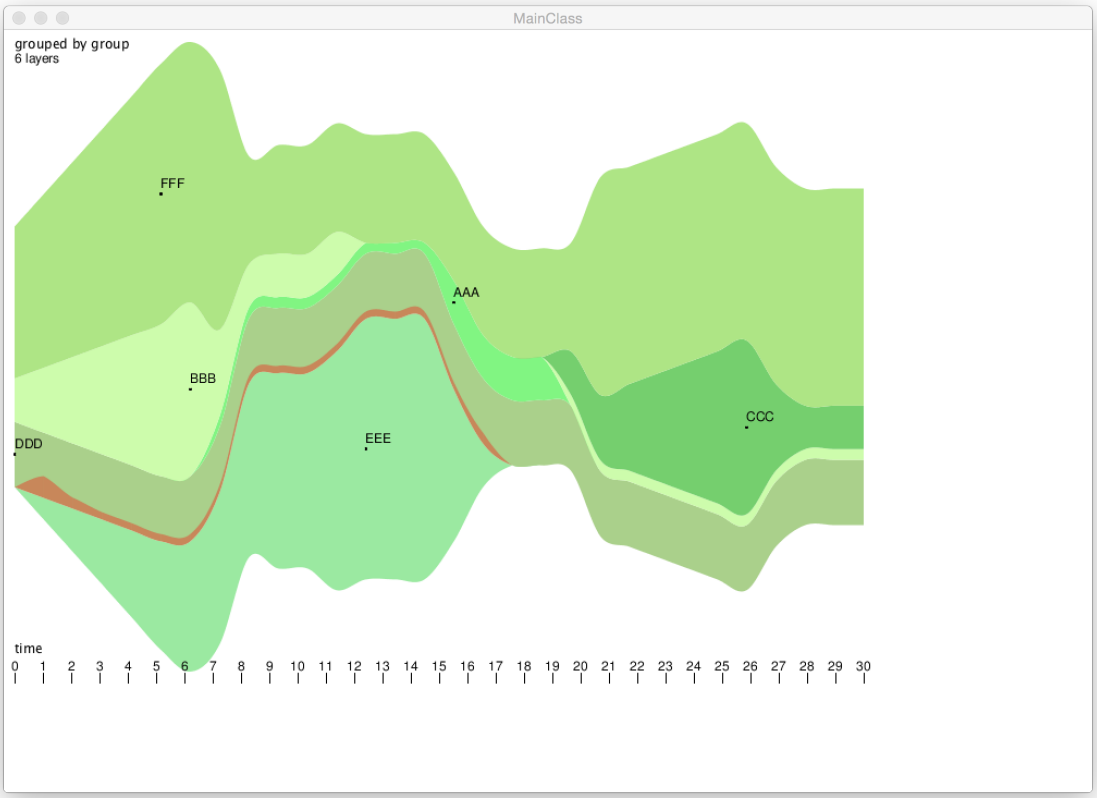


Figure 6.2: Streamgraph elementary importance factors for layers DDD and EEE

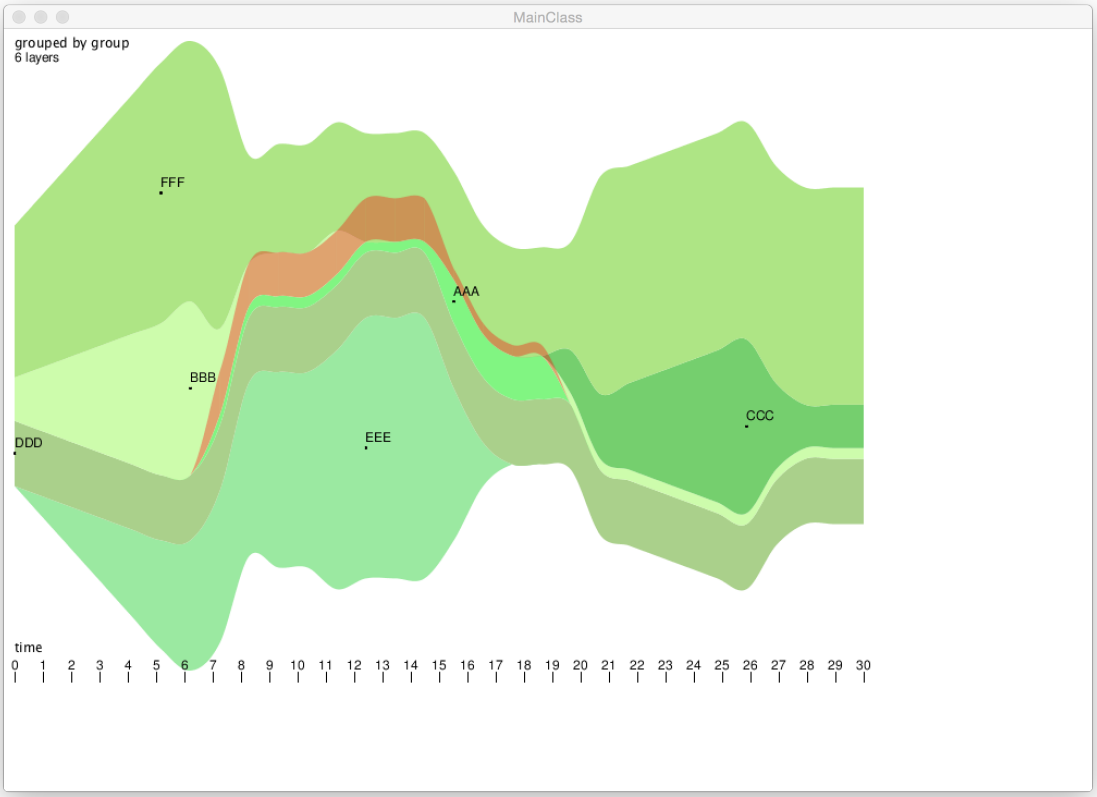


Figure 6.3: Streamgraph elementary importance factors for layers AAA and DDD

6.2.5. Synthesizing to Obtain an Importance Factor

We must, thus, compile elementary importance factor array values in order to get a global inter-layers importance factor, denoted $M(i, j)$. This synthesis process can be performed via several functions:

- maximum value

$$M(i, j) = max_{t \in T}\ \delta_t^{i,j} \tag{6.2}$$

- average value

$$M(i, j) = 1/card(T) \times \sum_{t \in T} \delta_t^{i,j} \tag{6.3}$$

The risk of averaging while synthesizing lies in that compensation would cancel problems. Using maximum formula could sometimes result in stating that all couples of layers share the same importance value. Data experiments seem to show that the synthesis method has to be adapted to the data. Extracting the data properties which make a specific synthesis more appropriate will need further researches.

6.2.6. Results

Figure 6.4 on page 23 depicts the resulting labeled streamgraph color assignment (maximum value synthesis) using a narrow part of the Viridis colormap to show the ability to reduce colormap hue interval while preserving distinguishability. The time series, displaying 21 artists of the music genre "alternative dance" presented throughout the day, comes from a sociological study about music listening usage [LG16].

In order to show the result of our class-color assignment and to compare color assignments, we also show two different color assignments, in figure 6.5 displayed with the basic order colormap and in figure 6.6 with a random color assignment.

To compare results of color assignment, the author chose to depict *comparison figures* dedicated to the class-color assignment problem, with importance matrix and permuted color distance matrix on the same diagram. This allows to verify the assignment of large color distances to large importance factors.

The three figures 6.7, 6.8 and 6.9 on page 24 depict comparisons for the three streamgraphs depicted on page 23: optimized, order and random assignments.

6.2.7. Importance for Similar Graphs

We may perform the same importance computation, based on the same formulas, for stacked-graphs and stacked bar graphs. This is also the case for circular streamgraphs and staked radar charts.

Concerning *sorted streamgraphs*[1] [Usm+20] or *alluvial* diagrams [2], $\gamma_t^{i,j}$, the neighborhood descriptor implementation has to be coded to handle the fact that layers may have various orderings along the time interval, but the formulas remain the same.

6.3. Polygonal Maps

Maps are a kind of visualization widely used to convey information. Categories are often displayed on maps, usually made of polygons of various sizes or based on pixel colors. In this visualization case too, we must choose category color with care. Our framework can handle polygons' discriminability by color assignment optimization, and this section presents an application for polygonal maps.

Figure 6.10: Colombus polygonal maps (Columbus `spData` R package)

6.3.1. Visualization Principles and Graphical Rules

Usually, several polygons associated with categories compose the map's database. Our framework needs polygon's neighborhood information. But on good database file, polygons borders are geometrically correct in term of line sharing, helping to infer polygons neighborhood.

[1] `https://datavizproject.com/data-type/sorted-stream-graph/`
[2] `https://datavizproject.com/data-type/alluvial-diagram/`

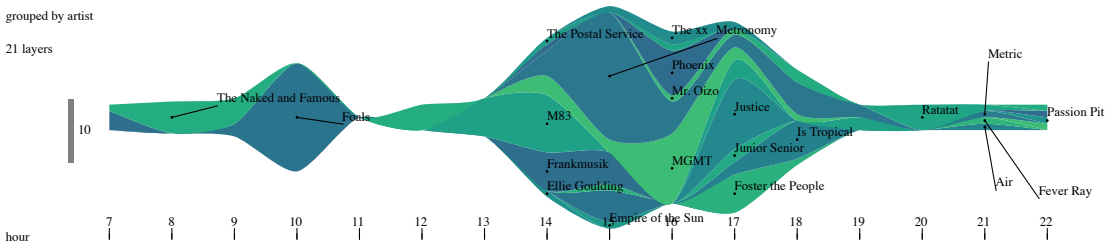


Figure 6.4: Steamgraph with optimized assignment

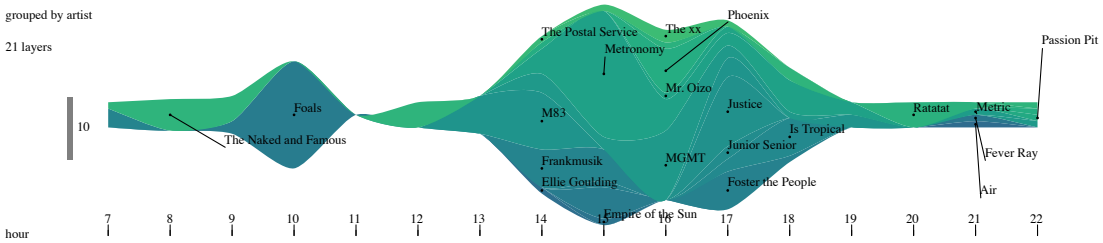


Figure 6.5: Steamgraph with colormap order assignment

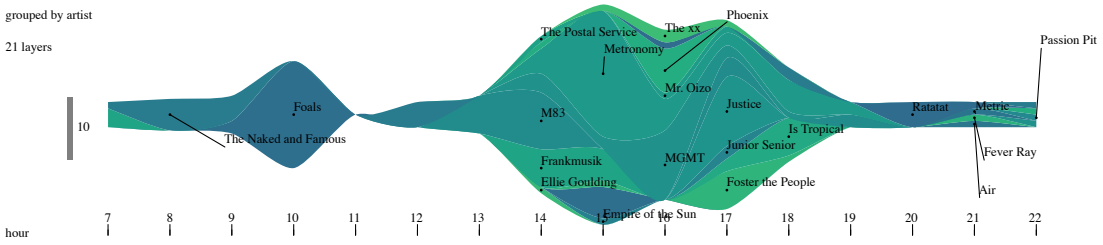


Figure 6.6: Steamgraph with colormap random assignment

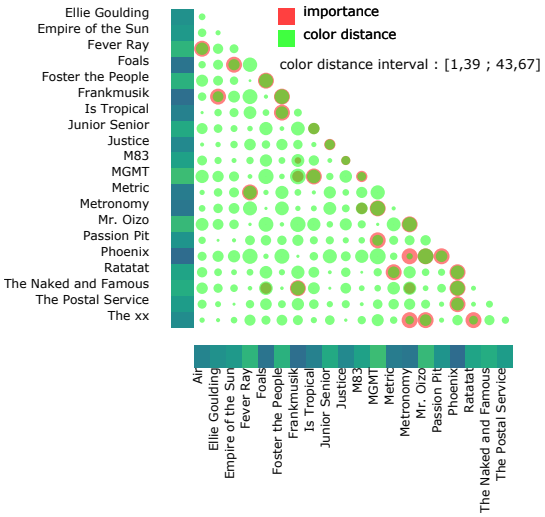


Figure 6.7: Steamgraph matrix comparison for optimized assignment

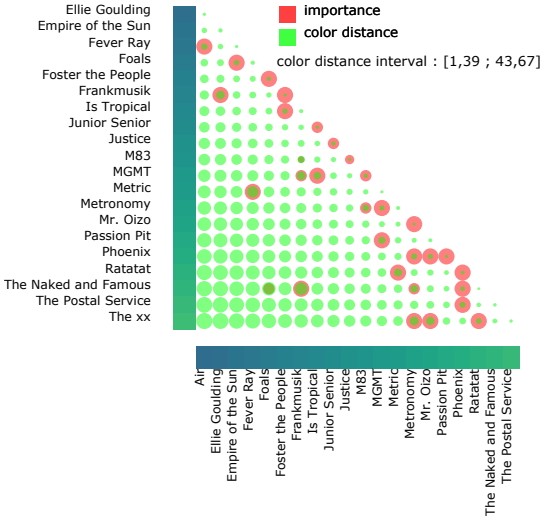


Figure 6.8: Steamgraph matrix comparison for colormap order assignment

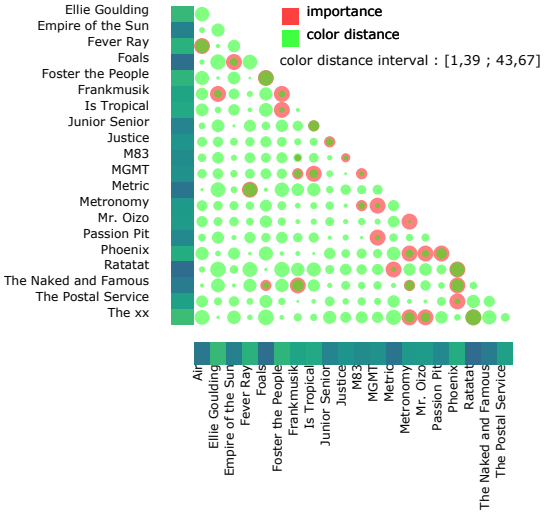


Figure 6.9: Steamgraph matrix comparison for colormap random assignment

6.3.2. Graphical Criteria for Color Assignment

In this kind of visualization too, graphical objects' thickness influences the perception. As stated by Newhall [New55], the length of the border between two polygons does not appear to be a good criterion. The perpendicular distance through the polygon seems to be a more accurate property.

6.3.3. Importance Computation

Several polygons may belong to a category and because polygons may be composed of complicated shapes, we estimate perpendicular distances along the borders which separate categories. As in streamgraph importance computation, we calculate perpendicular distances on the two sides of the border. These perpendicular thicknesses are used to estimate the *elementary importance* factors.

6.3.4. Elementary Importance Computation

We choose a step value, depending on the map scale, used as traveling increment along the frontier between two successive elementary importance factor estimations. On the lines separating category polygons, we generate points using step value, from which we compute perpendicular thicknesses on both sides. Then, we get an *elementary importance* value for each point, by evaluating the maximum of the inverse of the two thicknesses. Therefore, we obtain an array of *elementary importance* factors for each category couple, which has to be synthesized to get a single scalar value, the inter-class importance factor.

Figure 6.11 shows a simple example of perpendicular distances estimation. Four categories compose the map, displayed in pink, light pink, orange and yellow. The four images display the perpendicular distances found for each category, top-left the light pink category, top-right the yellow one, bottom-left, the orange category and bottom-right the pink one.

Let us now consider a map composed of n categories and, for simplification, let us assume that a category i is made of a single polygon i.

We consider E^i the set of edges of the category i which compose the frontiers with another category.

We assume that a point generator is available, that produces 2D points P_t^i, separated by a distance $step$ along the edge of E^i .

If nb^i is the number of points generated along i category, we have $t \in [1, nb^i]$.

Let us consider now a method $dist_j(P_t)$ which calculates the distance traveled within the polygon j from the point P_t perpendicularly to the current polygon edge (i.e. a perpendicular thickness). Calling this method $dist_j(P_t)$ using the neighboring category j on the current edge would give the perpendicular distance traveled within the category j from the point P_t.

As previously for streamgraph diagram, we use the same thickness inverse function to evaluate the distinctness between colors. Therefore, we calculate the elementary importance factor $\delta_t^{i,j}$ as following, using $\gamma_t^{i,j}$ which is equal to 1 if category i and j share a polygon edge on point t, and 0 otherwise:

$$\delta_t^{i,j} = \gamma_t^{i,j} * max(1/dist_i(P_t), 1/dist_j(P_t)) \tag{6.4}$$

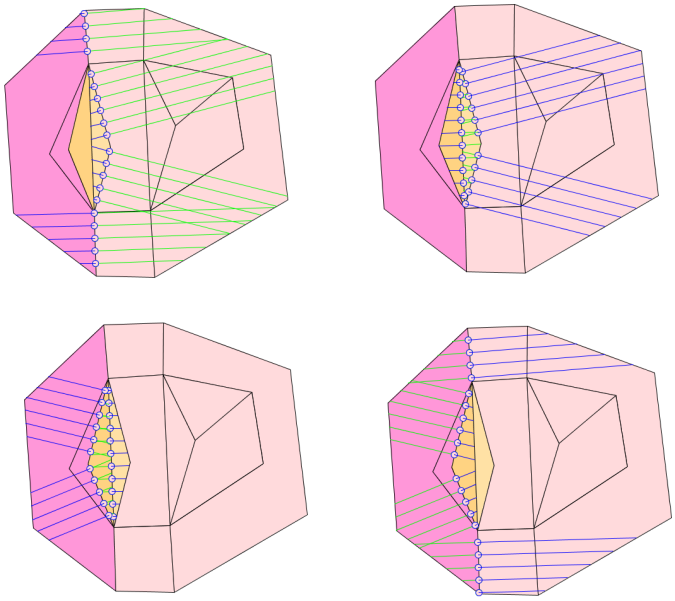


Figure 6.11: Perpendicular distances through categories

6.3.5. Synthesizing to Obtain an Importance Factor

We get the final importance factor for categories i and j *via* compilation of the *elementary importance* factor array. Several approaches have been tested:

Maximum. Simply taking the maximum of the elementary importances array.

$$M(i,j) = max_{t \in [1, nb^i]} \, \delta_t^{i,j} \tag{6.5}$$

Maximum of elementary importance factors excluding a percentage of the biggest values. Experiments shown that getting the maximum value of the elementary importance factors array results in high importance factors for every couple of categories. The sides of polygons which can be very thin cause this behavior. Therefore, cutting a percentage of the highest elementary importance factors within the array provides better results.

Importance by the inverse of area A simpler, yet providing worse results, method comprises directly evaluating the importance factor (without elementary importance factors estimation) by the maximum of the inverse polygon areas:

$$M(i,j) = \gamma^{i,j} \times max(1/area_i, 1/area_j) \tag{6.6}$$

With $\gamma^{i,j}$ equal to 1 if categories i and j share a frontier, and 0 otherwise.

The advantage of this second formula lays because R packages already include polygon area calculation. The greatest inconvenient is that it does not consider the shape of polygons, leading to attribute a small importance value to a thin long polygons despite its low thickness value. Another inconvenient consists in that two polygons that share a small edge but are of huge area result in a small importance factor.

6.3.6. Results

The figure 6.12 displays a 49 categories maps (Columbus `spData` R package) diagram using Viridis categorical colormap. It shows smaller couples of polygons assigned with higher color contrasts. The importance computation is based on inverse polygon areas.

Figure 6.13 on following page depicts the corresponding importance matrix.

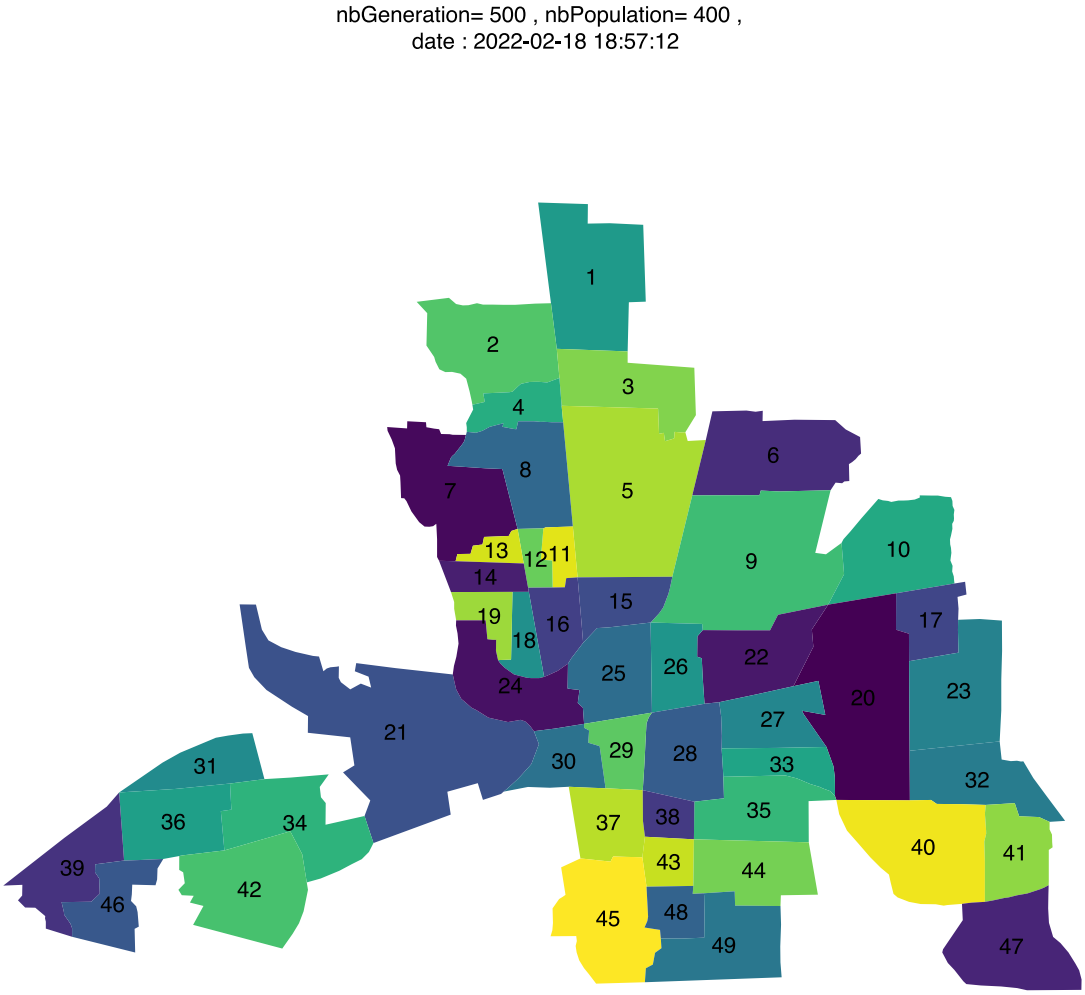


Figure 6.12: Colombus Map

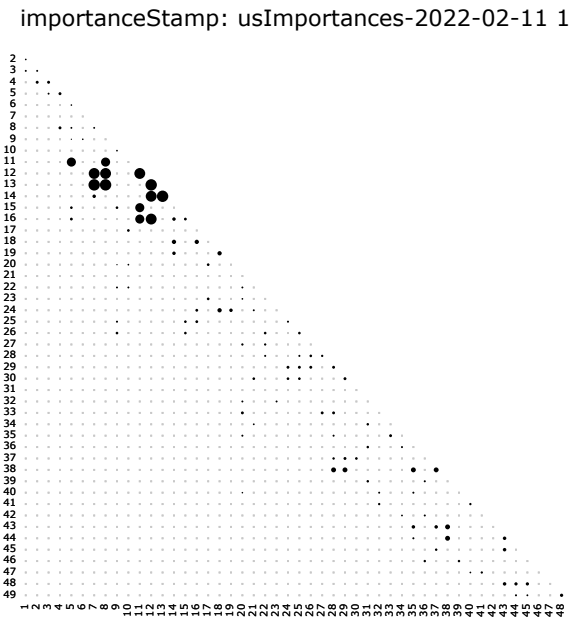


Figure 6.13: Importance Matrix for Colombus District Map

6.4. Chord Diagrams

Martin Krzywinski proposed *chord diagrams*, in a 2007 New York Times article called "*Close-ups of the Genome, Spieces by Spieces by Spieces*" [3]. Chord diagrams depict fluxes (directional type) and weighted relationships between entities (un-directional type). We depict a directional chord diagram in figure 6.18 which is visible on the *data-to-viz* website[4] which shows migration between world regions.

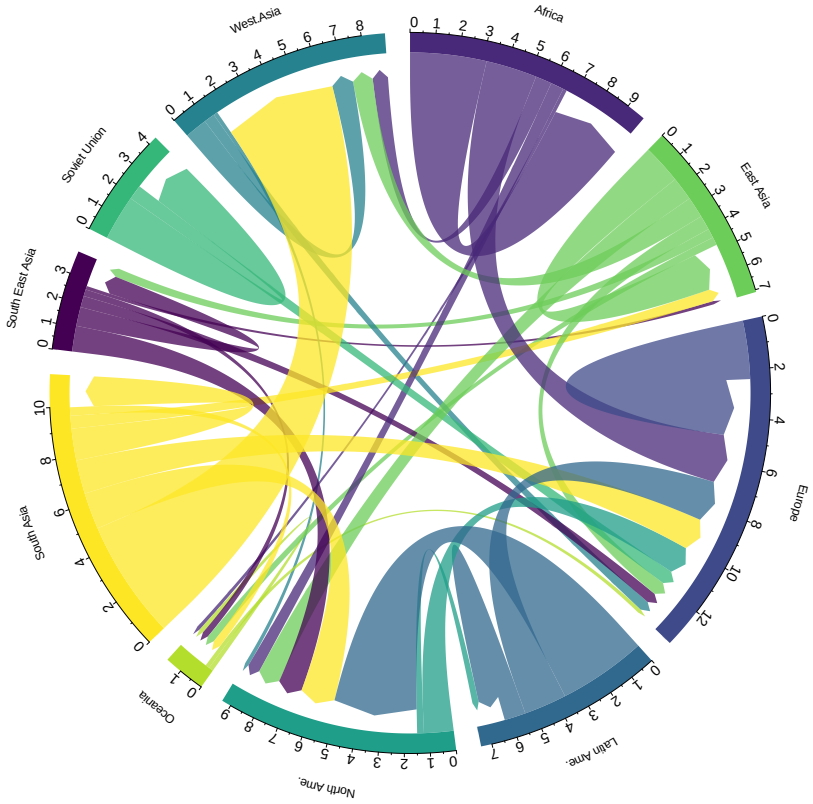


Figure 6.14: Chord diagram with color assignment

6.4.1. Visualization Principles

A chord diagram displays categories around a circle. It shows fluxes as arrows, starting from a group, say i, and arriving on one group, say j, including itself if needed. The width of each arrow depends on the flux $F(i, j)$. On each group, arriving fluxes are sorted in decreasing order.

F, a square matrix containing flux between nb categories may represent handled data with $F(i, j)$ corresponding to the flux from group i towards group j.

Let S^i_{in} be the sum of the flux arriving on the group i and S^i_{out} the sum of the flux starting from the group i.

6.4.2. Visualization Graphical Rules

The scale factor which is used to calculate the arrow's sizes is constant. The following equation, which states that no flux is lost, gives the overall fluxes sum.

$$S = \sum_{i=1}^{n} S^i_{in} = \sum_{i=1}^{n} S^i_{out} \tag{6.7}$$

[3] https://archive.nytimes.com/www.nytimes.com/imagepages/2007/01/22/science/20070123_SCI_ILLO.html
[4] https://www.data-to-viz.com/graph/chord.html

If we neglect the angular inter-group margin, the angular size $Angle^i$ of each group is determined by:

$$Angle^i = \frac{2.\pi.(S^i_{in} + S^i_{out})}{2 * S} \tag{6.8}$$

Chords diagrams depict arrows with the color of the flux starting group. Therefore, all the fluxes starting from a group are filled using the same color. The angular total width of these (grouped) starting arrows is related to S^i_{out} if i is the index of the group.

For each group, we place incoming arrows after outgoing arrows. For a group, incoming fluxes are displayed in decreasing order. Therefore, in each group, arriving fluxes are neighboring with various colors and various thicknesses.

We can omit the factor $(2.\pi)/(2 * S)$ in the following importance factor calculus, as it is present in all arrows' size expression.

6.4.3. Graphical Criteria for Color Assignment

To allow the user to interpret the diagram, in each group, the arrows must be visually distinct one from another. We then focus the importance factor calculation on the arrows neighborhood and their visual width. As a margin separates each category from its two neighboring categories, there is no need to insure separability from groups.

The center of the chord diagram can be visually confusing, and trying to optimize the legibility of the center part of the diagram could also be a goal. But in term of flux understanding, only the size and order of the arrows on each group are essential, arrows parts in the center being kind of "symbolic". In fact, the information lies on the side of the circle.

6.4.4. Importance Computation

Several important aspects have therefore to be computed for a group i,

- what are the groups of the fluxes arriving on group i?
- in which order are they depicted ? fluxes neighborhood?
- what are their visible sizes?
- what is the sum of the group i outgoing fluxes?

6.4.5. Notation

We denote O^i as the decreasing ordered list of the group indexes of incoming fluxes for the group i and nb^i_{in} as the number of incoming flux for the group i.

$O^i = \{x^i_k\}$ with:

- x^i_k is the k-st element of O^i
- $k \in [1, nb^i_{in}]$,
- $x^i_k \in [1, n]$

As O^i is ordered in decreasing order, we have $F(x^i_k, i) \geq F(x^i_{k+1}, i)$ for $k \in [1, nb^i_{in} - 1]$.

In order to ease expression writing, let us call o^i_p the index of group p in O^i.

6.4.6. Flux Neighborhood

Using previous notations, let us define $\gamma^i_{q,r}$, a directional chord diagram neighborhood descriptor which codes the fact that the fluxes coming from the groups p and q share a frontier on the arrival part of group i.

$\gamma^i_{q,r}$ is equal to 1 if the index of p and q are successors in O^i.

$$\gamma^i_{q,r} = \begin{cases} 1 & \text{if } p \in O^i \text{ and } q \in O^i \text{ and } |o^i_q - o^i_r| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6.9}$$

6.4.7. Elementary Importance Matrices

There are n parts on the circle corresponding to the n categories in the chord diagram. Arrows may be neighbors in each of these n parts. Therefore, for each category i, we construct a *local* importance matrix that codes the need to assign color contrast between (arriving and outgoing) arrows on the group i visual part.

We denote these n matrices as *elementary importance matrices* and their calculations involve a frontier descriptor frontier descriptor $\gamma^i_{q,r}$ and the flux values $F(i,j)$. If a category possesses no incoming flux from other groups, its elementary importance matrix is filled with 0 values, as there is no neighboring groups to be considered.

The final chord diagram importance matrix is obtained by setting, for each matrix element, the maximum value of corresponding elements in the *n elementary importance matrices*.

Let M^i be the elementary importance matrix for group i, and $m^i(q,r)$ its elements. M^i is a square matrix containing $n \times n$ elements (same as the chord diagram's final importance matrix).

The importance factor depends, as previously, on the size (*i.e.* thickness) of arrows, and is based on the inverse of arrow size, which is related to the corresponding flux. Considering neighboring information, the importance value for two group's incoming fluxes (including flux from the same group i) is estimated using the maximum of the inverse fluxes.

$$m^i(q,r) = \gamma^i_{q,r} \times \max(1/F(q,i), 1/F(r,i)) \tag{6.10}$$

6.4.8. Final Frontier

A remaining neighboring possibility has to be considered, which corresponds to the frontier for a given group i between the overall outgoing arrows and the first incoming arrow. The index of the first incoming flux is given by x^i_1 which is also the greatest incoming flux for group i.

The size of grouped outgoing fluxes is related to $\sum_{u\in[1,n]} F(i,u)$. Its neighboring flux, *i.e.* the incoming greatest flux, is $F(x^i_1, i)$ (if x^i_1 exists and $x^i_1 \neq i$).

The corresponding *in/out frontier* importance, if x^i_1 exists and $x^i_1 \neq i$, is given by:

$$m^i(i, x^i_1) = max(\frac{1}{\sum_{u\in O^i} F(i,u)}, \frac{1}{F(x^i_1, i)}) \tag{6.11}$$

Finally, the element $m^i(i, x^i_1)$ of the group i elementary importance matrix is updated by getting the maximum of the importance calculated on the frontier by equation 6.11[5] and the previous corresponding $m^i(i, x^i_1)$ about incoming flux (equation 6.10).

6.4.9. Resulting Importance Matrix

To be used by the class-color assignment optimization, we must provide a single importance matrix. Synthesizing the n elementary matrices M^i in a chord diagram's importance matrix M containing $n \times n$ elements, is performed by getting, for each element $M(l,m)$, the maximum of the corresponding elements $m^i(l,m)$ of the n *elementary importance matrices*.

$$M(l,m) = max_{i\in[1,n]}(m^i(l,m)) \tag{6.12}$$

6.4.10. Importance Matrix Data Application

To experiment with the application of our framework on chord diagrams, we implemented the R code given on *data-to-viz*[6] website. A chord diagram is displayed on the site using emigration data[7] coming from a publication by Guy J. Abel (2017) "*Estimates of Global Bilateral Migration Flows by Gender between 1960 and 2015*" [Abe17]

We display the ten *elementary importance matrices* in figures 6.15 and 6.16, for which it is notable that some of these matrices contain only 0 as there are no neighboring arrows for the corresponding groups.

Figure 6.17 shows the resulting synthesized importance matrix.

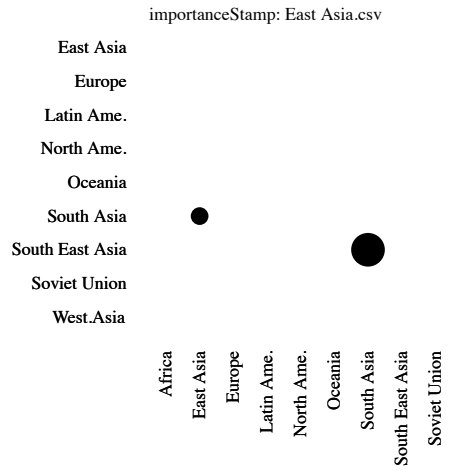
[5] if $x^i_1 = i$ the formula has to be modified as the arriving flux is neighboring the outgoing flux

[6] `https://www.data-to-viz.com/graph/chord.html`

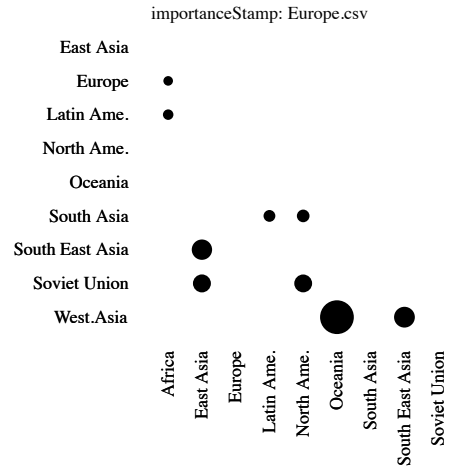
[7] `https://raw.githubusercontent.com/holtzy/data_to_viz/master/Example_dataset/13_AdjacencyDirectedWeighted.csv`

The optimized chord diagram is displayed on figure 6.19. Small fluxes are more visible, for example, on "Oceania" incoming fluxes (at bottom left). Fluxes arriving on “Europe” category, which are greater than fluxes arriving on “Oceania”, are less color contrasted. But the flux from “South east Asia” to “East Asia” is clearly visible on the color optimized diagram, whereas was not enough distinct in the original diagram.

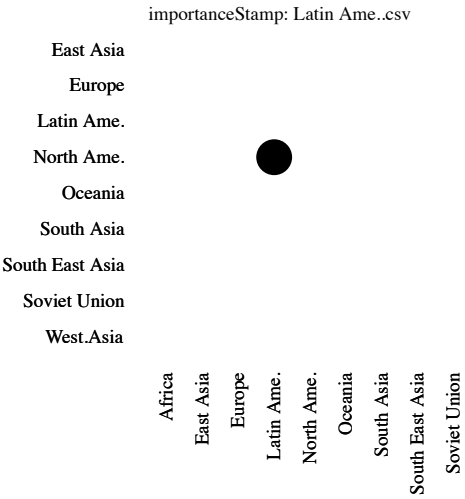
(a) Chord Diagram Africa imp. matrix



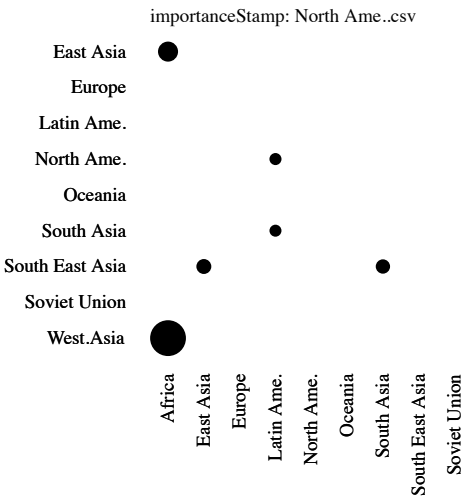
(b) Chord Diagram East Asia imp. matrix



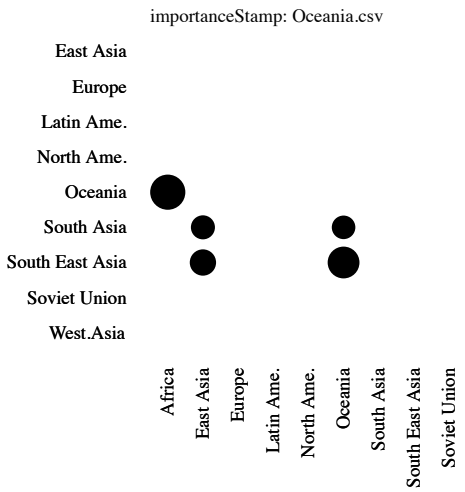
(c) Chord Diagram Europe imp. matrix



(d) Chord Diagram Latin America imp. matrix

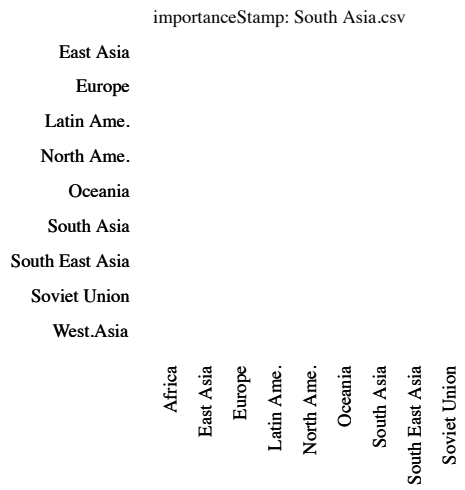


(e) Chord Diagram North America imp. matrix

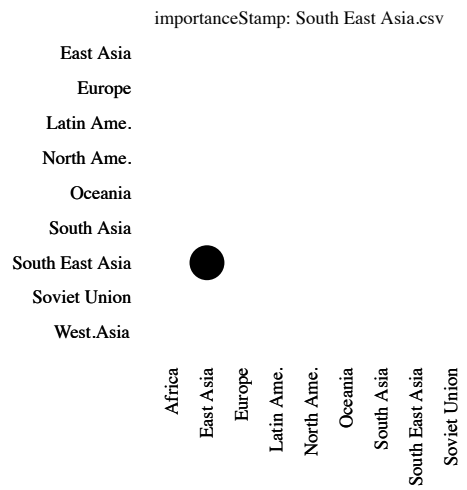


(f) Chord Diagram Oceania imp. matrix

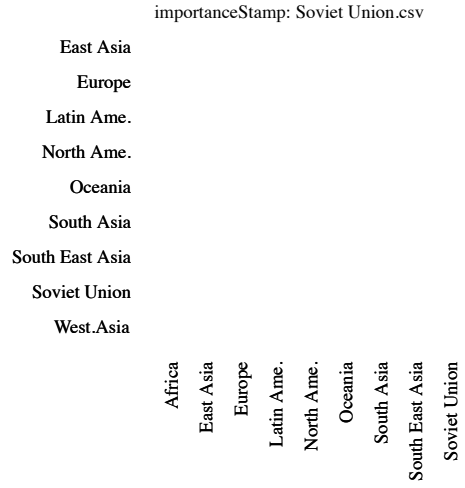
Figure 6.15: Elementary importance matrices of the six first groups



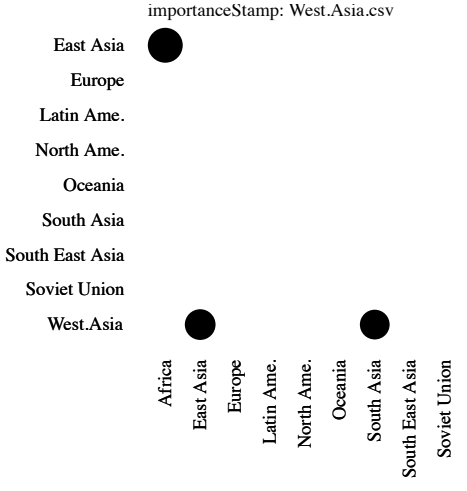
(a) Chord Diagram South Asia imp. matrix



(b) Chord Diagram South East Asia imp. matrix



(c) Chord Diagram Soviet Union imp. matrix



(d) Chord Diagram West Asia imp. matrix

Figure 6.16: Elementary importance matrices of the four remaining groups

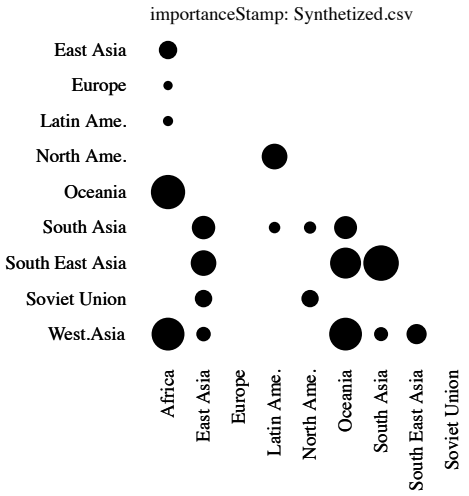


Figure 6.17: Chord Diagram Synthetized Importance Matrix

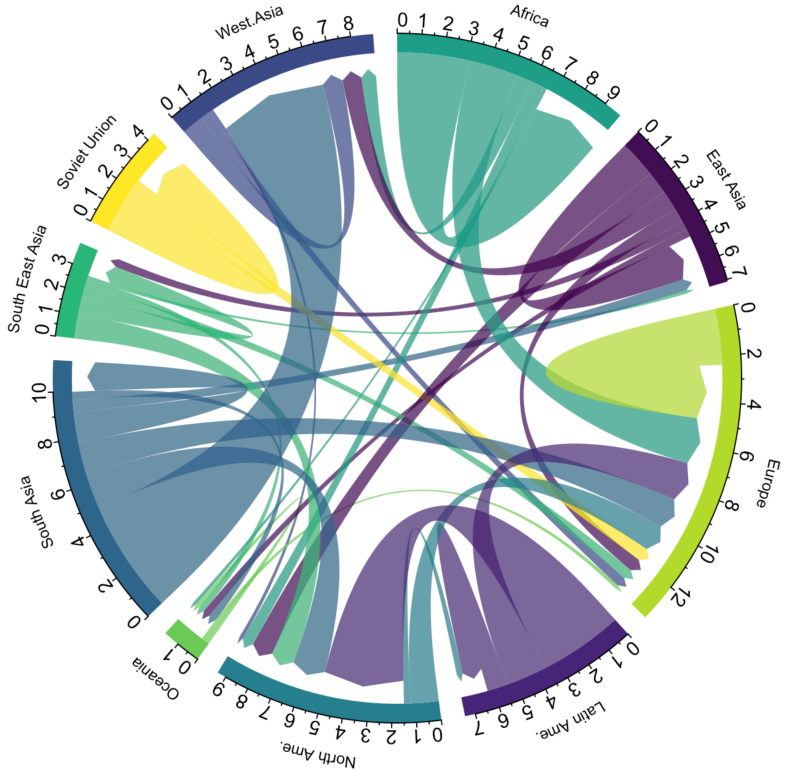


Figure 6.18: Original Chord Diagram from data-to-viz website

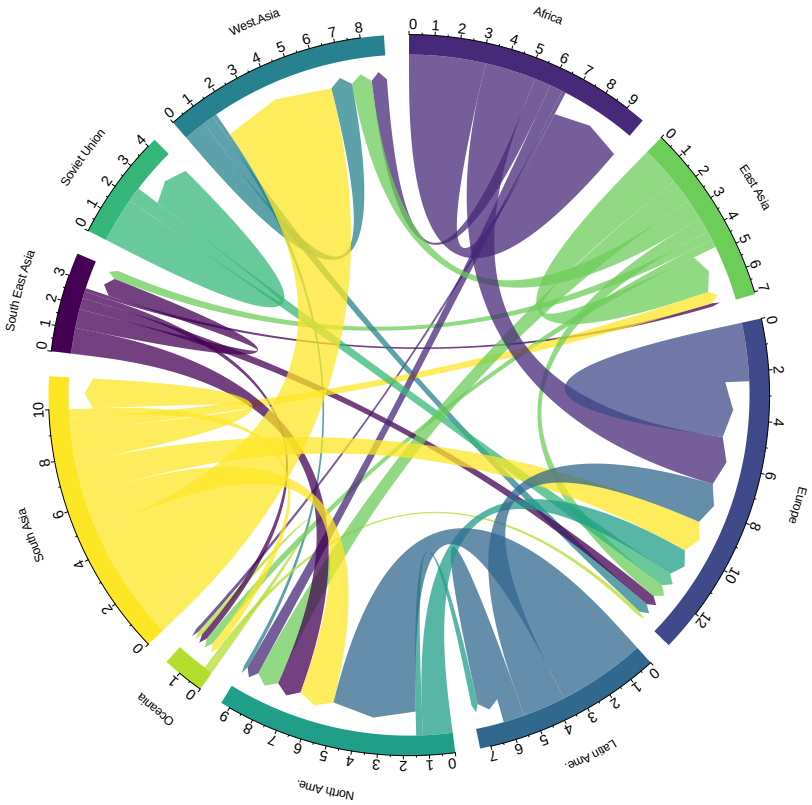


Figure 6.19: Chord diagram with optimized class-color assignment

6.5. Line Charts

Line chart is probably the most used visualization diagram in everyday media, showing time values variation, estimation and comparison. This section treats only line charts which are displayed like function $y = f(x)$ graph (one value y for an x value, per chart). The analysis does not allow (but could be extended to allow) free poly-lines ("metro lines" for example).

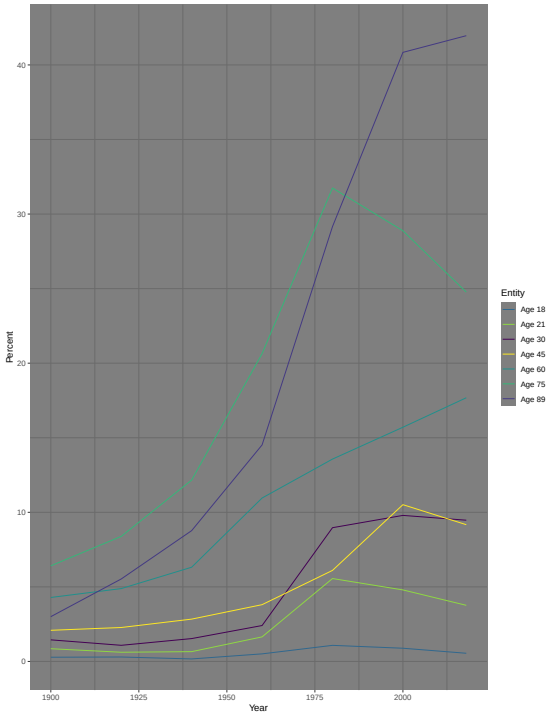


Figure 6.20: Line charts GDP with color assignment optimization

6.5.1. Graphical Criteria for Importance

Let us consider a line chart displaying n curves as segmented lines. Usually, while reading line charts, our eyes follow the various lines, estimating values and comparing one line to the other. Among difficulties which could arise when using close colors are two special cases for which the line eye following process can become hard:

- when lines are close one to another (*Proximity*);
- when two lines are intersecting each other with a low angle crossing (*Low Angle Crossing*).

We incorporate these two cases into the inter-line importance expression.

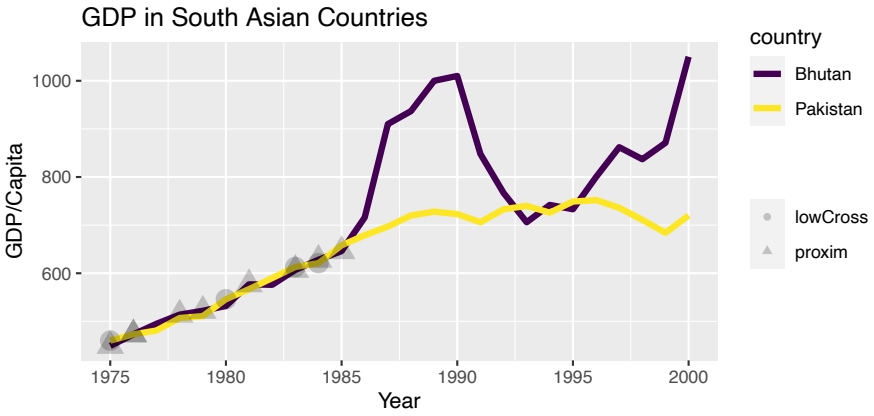


Figure 6.21: Problematic zones detection

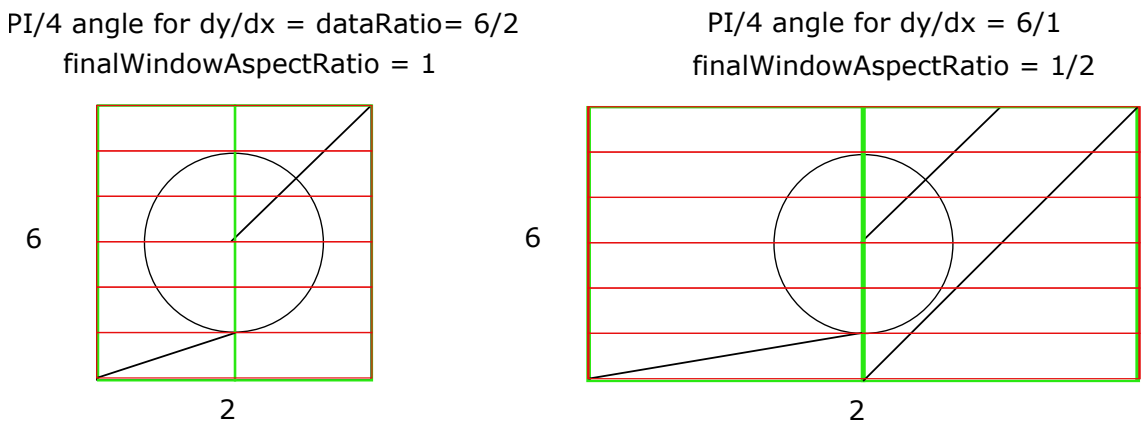


Figure 6.22: Low angle line intersections dependent to image final ratio

6.5.2. Importance Evaluation

We should relate *proximity* factor to the final subtended visual angle, which we could evaluate using the final displayed diagram size and viewed distance. Usually, this information is not available. Alternatively, we evaluate the proximity between lines on y axis, using a percentage of the depicted y interval, depending therefore only on the data. In figure 6.21 *proximity* zones are depicted as triangle marks and *low angle intersections* zones as circles.

On the contrary, we cannot evaluate low angle lines crossings without supplementary diagram information. We can compute a crossing angle as the absolute difference of the two slopes (angles) values. But the visible angle depends on the *final diagram ratio*. As seen in figure 6.22, the lines slope depends on the figure ratio between figure width and figure height and on the x and y displayed intervals.

As a first test, the final importance factor associated with two lines is based on the sum of the number of occurrences of *proximity* zones and *low angle intersections*. Two coefficients to get asymmetrical contributions of *proximity* and *low angle crossing* zones have been introduced without seeing benefits for the tested data so far.

We provide in the following formulas which count the number of "zones" for proximity and for low crossing. As the maximum number of occurrences is, for a diagram, the same for all lines (approximately the number of x samples, because crossing may appear on x samples and between successive x samples), there is no need to normalize it.

Proximity Importance Evaluation

Let us denotes n the number of depicted lines and nbp the number of x values which are given with corresponding y values. The line i is traced using nbp points of coordinates (x_t^i, y_t^i) with $t \in [1, nbp]$.

An absolute y limit value for which two lines would be considered as too close would not be user friendly. We prefer setting $percent_y$, the percentage of y interval which do not depend on data.

Let us denote y_{min} and y_{max} the minimum and maximum y values depicted on the figure
and x_{min} and y_{max} the minimum and maximum values on x axis.

Let us call $nbProximity^{i,j}$, the number of x values for which two lines i and j are too close. This number of occurrences is given by:

$$nbProximity^{i,j} = card\{x_t^i, t \in [1, nbp] \text{ such that } |y_t^i - y_t^j| < percent_y * (y_{max} - y_{min})\} \tag{6.13}$$

Low Angle Crossing Importance Evaluation

Concerning low angles line intersections, we evaluate it through a two-step process.

1. detection of t values where lines are intersecting one another;
2. lines couple crossing angle evaluation for the detected t values.

The user provides $limitAngle$, a value in degrees for which two lines, crossing with an angle less than $limitAngle$, would be difficult to follow if their colors are too close.

Let us consider the intersections between line i and line j. First,we list the t values for which there is a change in y_t^i and y_t^j value order. To perform this detection, $V^{i,j}$, an array of nbp values coding the y order, is computed as following:

$$V_t^{i,j} = \begin{cases} 1 & \text{if } y_t^i > y_t^j \\ -1 & \text{if } y_t^i < y_t^j \\ 0 & \text{otherwise} \end{cases} \tag{6.14}$$

Then detection of patterns like $[-1, 1]$, $[1, -1]$, $[-1, 0, 1]$ and $[1, 0, -1]$ in the $V^{i,j}$ vector gives the set of t values, denoted $SLA^{i,j}$, for which the crossing angle has to be evaluated.

The second step is concerned with the crossing angle evaluation.

Let us define:

- $ratio_{fig} = width/height$ the figure final aspect ratio specified by the user and used to generate the figure;
- $ratio_{data} = \frac{x_{max}-x_{min}}{y_{max}-y_{min}}$ the data ratio.

Let us now calculate the angle with x axis of a single line i for the t value part.

Using $ratio_{fig}$ and $ratio_{data}$, the slope of line i in degrees for the t values provided by vector $SLA^{i,j}$ is given by:

$$visualAngle^i_{t \in SLA^{i,j}} = \frac{180}{\pi} \times (atan((y^i_{t+1} - y^i_t) \times \frac{ratio_{data}}{ratio_{fig}})) \tag{6.15}$$

Let us call $nbLowAngleCrossing^{i,j}$, the number of zones for which the crossing angle is too low, that is given for line i and line j by the following expression:

$$nbLowAngleCrossing^{i,j} = card\{t \in SLA^{i,j} \text{ such that } |visualAngle^i_t - visualAngle^j_t| < limitAngle\} \tag{6.16}$$

where $limitAngle$ is set by the user.

Finally, we express the importance factor for line i and line j, thanks to the two coefficients α and β which are defined to provide flexibility in the importance factor evaluation.

$$M(i, j) = (\alpha \times nbProximity^{i,j}) + (\beta \times nbLowAngleCrossing^{i,j}) \tag{6.17}$$

So far, the few experiments have shown that setting α and β to 0.5 were giving good results, but we should perform thorough testing to confirm this hypothesis.

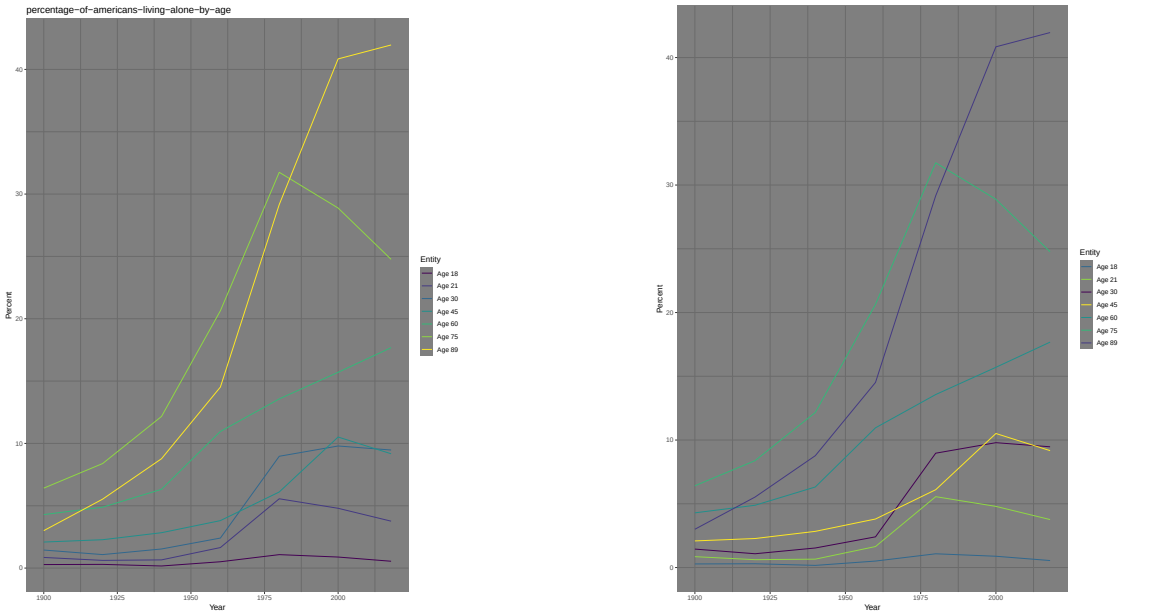
6.5.3. Results

A good test base for line-charts color assignment optimization is provided by a page of the website *Our World in Data*[8]. The treated data are the Percentage of Americans living alone, by age from year 1900 to year 2018. The figure 6.23 shows a image capture from the site. Figures 6.24a and 6.24b display respectively color lines without optimization and with color assignment optimization.

We show comparison figures with the color distance matrix and the importance matrix:

- without color assignment optimization in figure 6.25a
- with a random color assignment in figure 6.25b
- with color assignment optimization in figure 6.26

[8] https://ourworldindata.org/grapher/percentage-of-americans-living-alone-by-age



(a) Line charts, Same data, no optimization

(b) Line charts, Same data, with color assignment optimization

Figure 6.24: Line charts, data from "Our World In Data"

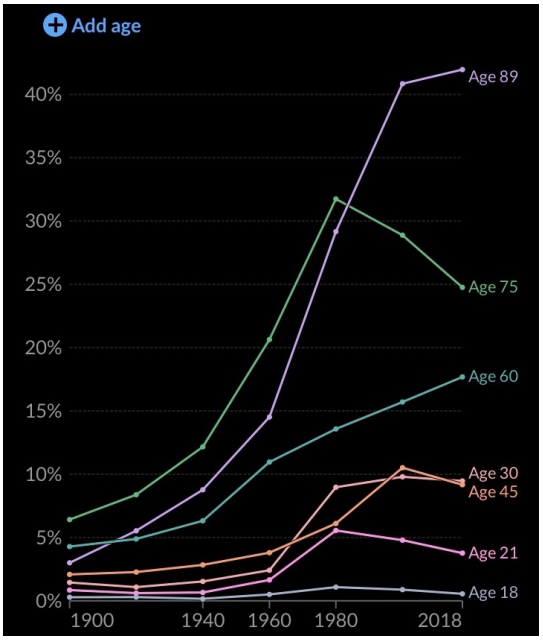


Figure 6.23: Image capture, Line charts, data from "Our World In Data"

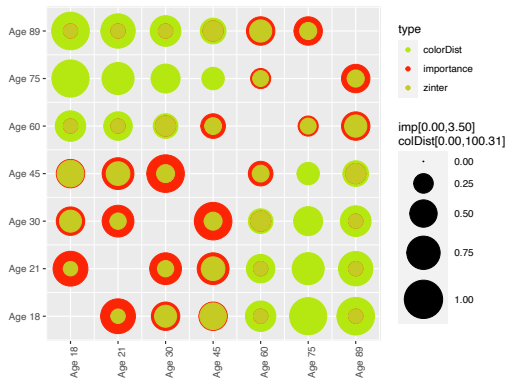
6.6. Scatterplots Diagrams

Scatterplots are 2D diagrams usually displaying records using two properties, sometimes more.

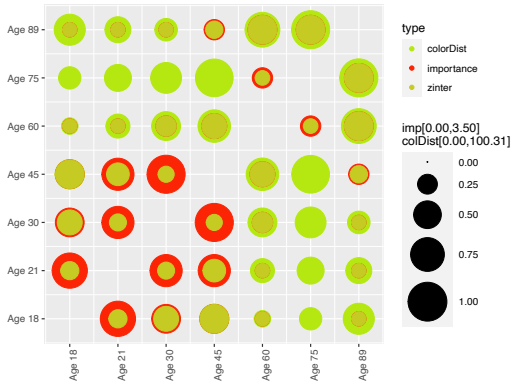
6.6.1. Visualization Principles and Visualization Graphical Rules

In scatterplots diagram, records are shown as points located on the plane according to two chosen properties. With categorical data, point colors show classes as in figure figure[9] 6.27. Sometimes, it integrates the diameter of discs to display a fourth property. Assigning colors to classes can be hard a

[9] `https://en.wikipedia.org/wiki/User:Opabinia_regalis/RfA_data`



(a) Line charts, Results Comparison, no assignment optim



(b) Line charts, Results Comparison, random assignment

Figure 6.25: Line charts, data from "Our World In Data"

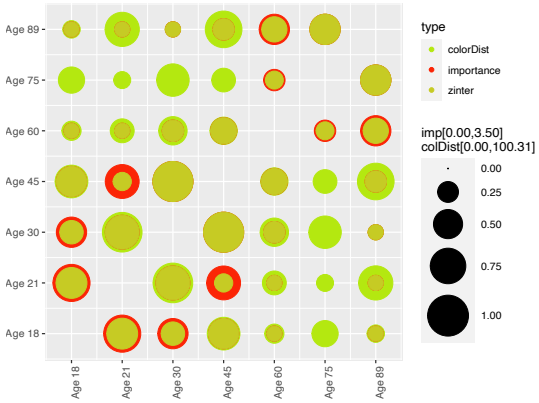


Figure 6.26: Line charts, Results Comparison, with assignment optim

process when discs overlap or when there are regions in the diagram displaying close points belonging to many classes.

6.6.2. Point Distinctness for Class Separation by Palettailor

Here under, we show the formulas for *point distinctness* which are given in the "Palettailor" paper [Lu+21] in order to express the quality of the separation between classes using a palette color permutation.

Let $C = C_1 ... C_m$ the color samples
and $X = x_1 ... x_n$ the points
where data class j is composed of n_j points.
The *point distinctness* is given for point x_i by

$$\alpha(x_i) = \frac{1}{|\Omega_i|} \sum_{x_j \in \Omega_i} \Delta\epsilon(c_i, c_j) g(d(x_i, x_j)) \tag{6.18}$$

with Ω_i the set of the closest point to x_i. The color distance denoted by $\Delta\epsilon(c_i, c_j)$.
$g(d(x_i, x_j))$ a geometric distance allowing a greater weight to close points and a lesser weight to distant points: $g(d) = 1/d$
Ω_i is obtained via an α-shape (based on Delaunay triangulation).
Class j is composed of points $x_1^j, x_2^j, ..., x_{n_j}^j$.

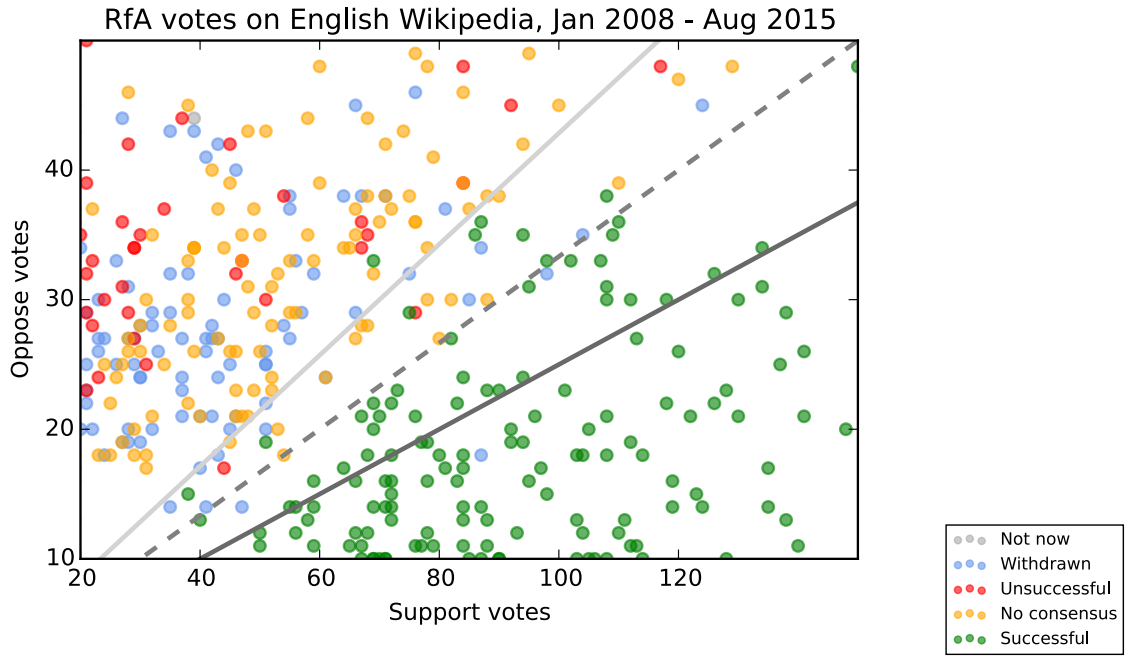


Figure 6.27: Scatterplot example (wikipedia)

The final energy is obtained by summing the point distinctness of the whole points in every classes:

$$E = \sum_{j=1}^{m} \sum_{i=1}^{n_j} \alpha(x_i^j) \tag{6.19}$$

6.6.3. Point Distinctness Energy Developed

The energy equation 6.19 can be rewritten using expression *point distinctness* 6.18:

$$E = \sum_{j=1}^{m} \sum_{i=1}^{n_j} \frac{1}{|\Omega_i|} \sum_{p_k \in \Omega_i} \Delta\epsilon(c_i, c_k) g(d(x_i, p_k)) \tag{6.20}$$

with c_k the color of the p_k point.

Ω_i the set of closest point to x_i can be broken into m sets, each one containing the points of given classes.

Let us call $\Omega_{i,q}$ the subset of Ω_i containing only the points belonging to class q.

$$\Omega_i = \bigcup_{q \in [1,m]} \Omega_{i,q}$$

Therefore, the $\alpha(x_i^j)$ in formula 6.19 can be expressed by:

$$\alpha(x_i^j) = \sum_{q \in [1,m]} \frac{1}{|\Omega_{i,q}|} \sum_{p_k^q \in \Omega_{i,q}} \Delta\epsilon(c_i, c_q) g(d(x_i^j, p_k^q)) \tag{6.21}$$

6.6.4. Importance Formula Based on Point Distinctness

Then, to fit in our importance matrix concept, we re-write the previous energy expression 6.20, while extracting the importance factor.

In order to simplify the re-writing, let us consider the energy part gained by two classes C_1 and C_2. $\Delta\epsilon(C_1, C_2)$ can be factorized leading to the following expression:

$$
\begin{aligned}
E(C_1, C_2) =& \Delta\epsilon(C_1, C_2) \\
&\times \left(\sum_{a=1}^{n_1} \sum_{x_b^2 \in \Omega_a^1} \frac{1}{|\Omega_a^1|} g(d(x_a^1, x_b^2)) \right. \\
&\left. + \sum_{b=1}^{n_2} \sum_{x_a^1 \in \Omega_b^2} \frac{1}{|\Omega_b^2|} g(d(x_b^2, x_a^1)) \right)
\end{aligned} \tag{6.22}
$$

with n_1 (resp. n_2) being the number of points in the the class C_1 (resp. C_2).

Symmetry shows that the two contributions from C_1 to C_2 on one hand and C_2 to C_1 on the other hand must be equal. Taking therefore only one way contribution, we simplify the energy expression 6.22:

$$
E(C_1, C_2) = \Delta\epsilon(C_1, C_2) \times \left(\sum_{a=1}^{n_1} \sum_{x_b^2 \in \Omega_a^1} \frac{1}{|\Omega_a^1|} g(d(x_a^1, x_b^2)) \right)
$$

Finally, the second term of the multiplication gives the importance factor for the classes C_1 and C_2:

$$
M(C_1, C_2) = \sum_{a=1}^{n_1} \sum_{x_b^2 \in \Omega_a^1} \frac{1}{|\Omega_a^1|} g(d(x_a^1, x_b^2))
$$

Therefore, to evaluate $M(C_1, C_2)$, for all the points x_a^1 of class C_1, we construct the influence set Ω_a^1. Then we reduce each set by keeping only the points belonging to class C_2. The weight function is applied and finally the importance factor is calculated by summing the results.

We have not implemented yet the previous importance factor expression for scatterplot diagram in our framework.

6.7. Simple Neighbor Diagrams

Our color-class assignment method may handle some less complicated diagrams like single stacked bar as in figure 6.28 or pie charts (generally discarded by data scientists for the lack of value comparisons but often used in general audience) as in figure 6.29. In these two cases, while classes possess 1 or 2 neighbors, improving class separation and avoiding color spreading through an optimization of color-class assignment can be profitable.

Figure 6.28: Bar chart example (dummy data) with reduced colormap

6.7.1. Visualization Principles and Visualization Graphical Rules

Let us call x_i the values corresponding to the n classes. In a pie chart, the slice angle θ_i occupied by a class i is related to x_i and to the sum of x_i:

$$
\theta_i = 2\pi \times (x_i / \sum_{j=1}^{n} x_j)
$$

In a single bar chart, the height of class i rectangle is related to the total visual height of the chart and to the sum of x_i.

$$
height_i = height_{figure} \times (x_i / \sum_{j=1}^{n} x_j))
$$

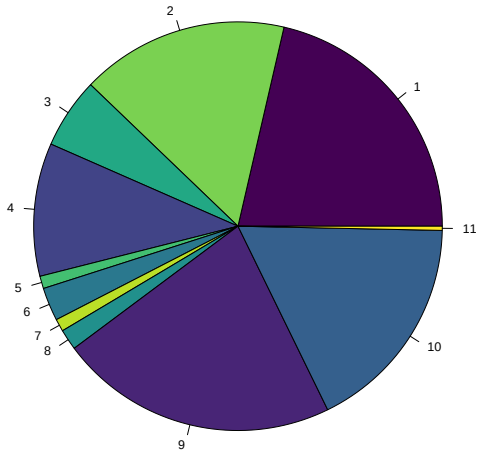


Figure 6.29: Pie chart example (dummy data) using viridis colormap

6.7.2. Graphical Criteria for Color Assignment and Importance Computation

As for previous diagrams, a decreasing function can model the fact that smaller objects need more color contrast with neighbors to be legible. We also apply the size inverse function in these two cases, leading to the following expressions:

$$M(i,j) = \gamma^{i,j} \times max(1/\theta_i, 1/\theta_j) \tag{6.23}$$

$$M(i,j) = \gamma^{i,j} \times max(1/height_i, 1/height_j) \tag{6.24}$$

where $\gamma^{i,j}$ denotes the neighborhood between classes i and j, which is equal to 1 if class objects i and j share a frontier, and 0 otherwise. We can notice that the multiplicative factor $(\sum_{j=1}^{nb} x_j)/2\pi$ which is present in all $theta_i$ values and the corresponding factor appearing in $M(i,j)$ expression (equation 6.23) can be omitted. It won't affect the optimization, as it would only be a final multiplicative factor in the energy. expression.

Same simplification applies for a single bar chart, where $\sum_{j=1}^{n} x_j)/height_{figure}$ may be omitted too (equation 6.24).

Therefore, we can simplify the importance factors expression for both diagrams into the same expression:

$$M(i,j) = \gamma^{i,j} \times max(1/x_i, 1/x_j) \tag{6.25}$$

6.7.3. Results

The Importance matrix is, for these two diagrams, very simple as each class possesses only 1 or 2 neighbors. The optimization process is therefore faster than for more complicated diagrams. We showed two examples on the previous page, figures 6.28 and 6.29.

6.8. Multiple Diagrams Importance Synthesis

When several visualizations of the same type and sharing some groups of data have to be designed, we can construct a global importance matrix. This matrix codes the need for contrast between the objects appearing in the diagrams. The size of the matrix corresponds to the cardinal of the *union* of the categories displayed in the several diagrams.

We *synthesize* the global importance matrix calculated for each visualization.

Figures 6.31 and 6.32 display two streamgraphs which share some category names (music artists).

Keeping the same color for names displayed in the two visualizations seems natural. To achieve this goal, we compute the importance matrices for the two visualizations separately (see figures 6.33a and 6.33b) and we calculate a global *synthesized importance matrix* using the importance values appearing in the two matrices (figure 6.34).

Once the *synthesised importance matrix* is calculated, our color-class assignment optimization affects colors to layers in the two visualizations. The following section explains the formulas involved in the *synthesized importance matrix* calculation.

6.8.1. Global Synthesized Importance Matrix Computation

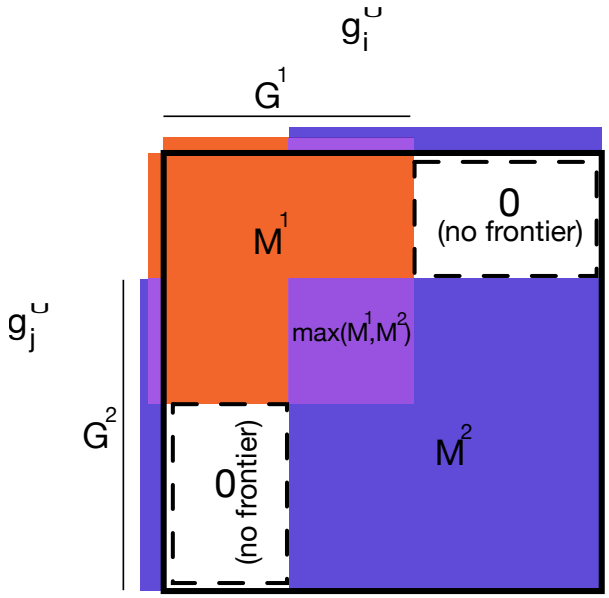


Figure 6.30: Importance matrix synthesis

For seek of simplicity, we give the following explanations for two figures, referred as diagram 1 and diagram 2, but the principles are easily extended

Let us call G^1 the set of group names appearing on diagram 1, and G^2, the set of group names of diagram 2.

Let g_i^1 be the name at index i in set G^1 and g_j^2 the name at index j in G^2.

Let us call $G^\cup$ the union set of G^1 and G^2 and denote $G^\cap$ as the intersection set of G^1 and G^2.

$$G^\cup = G^1 \cup G^2 \tag{6.26}$$

$$G^\cap = G^1 \cap G^2 \tag{6.27}$$

Let us suppose that importance matrices M^1 and M^2 are computed using the formulas and data corresponding to the two diagrams 1 and 2.

We can assume that an access function $M^1(g_i^1, g_j^1)$ is available, that helps to get an element of the matrix M^1 using two names appearing in G^1, instead of using their matrix indexes.

$M^1(g_i^1, g_j^1)$ returns the importance factor for the two class objects which names are g_i^1 and g_j^1. In the same way, importance factor of importance matrix M^2 for the two names g_i^2 and g_j^2 can be obtained by calling $M^2(g_i^2, g_j^2)$.

To compute M, the synthesis of the two importance matrices, for two names $(g_i^\cup, g_j^\cup)$ in $G^\cup \times G^\cup$, the following rules are applied (see figure 6.30):

- use the maximum of M^1 and M^2 if both names belong to $G^\cap$
- use M^1 if one of the names belong to G^1 minus G^2 and the other does not appear in G^2 minus G^1
- use M^2 if one of the names belong to G^2 minus G^1 and the other does not appear in G^1 minus G^2
- otherwise the corresponding importance value is set to 0.

This gives the following formula, for $g_i^{\cup}, g_j^{\cup} \in G^{\cup} \times G^{\cup}$:

$$M^{\cup}(g_i^{\cup}, g_j^{\cup}) = \begin{cases} M^1(g_i^{\cup}, g_j^{\cup}) & \text{if } g_i^{\cup} \in G^1 \setminus G^2 \text{ and } g_j^{\cup} \notin G^2 \setminus G^1 \\ & \text{if } g_j^{\cup} \in G^1 \setminus G^2 \text{ and } g_i^{\cup} \notin G^2 \setminus G^1 \\ M^2(g_i^{\cup}, g_j^{\cup}) & \text{if } g_i^{\cup} \in G^2 \setminus G^1 \text{ and } g_j^{\cup} \notin G^1 \setminus G^2 \\ & \text{if } g_j^{\cup} \in G^2 \setminus G^1 \text{ and } g_i^{\cup} \notin G^1 \setminus G^2 \\ \max(M^1(g_i^{\cup}, g_j^{\cup}), M^2(g_i^{\cup}, g_j^{\cup})) & \text{if } g_i^{\cup} \in G^{\cap} \text{ and } g_j^{\cup} \in G^{\cap} \\ 0 & \text{otherwise} \end{cases} \tag{6.28}$$

The algorithm to fill the *synthesized importance matrix* ($M^{\cup}$) is somehow simpler:

Algorithm 1 Synthesis importance matrix filling

```
//———— g_i and g_i are names of G^∪
//————initialize with 0
for g_i ∈ G^∪ do
  for g_j ∈ G^∪ do
    M^∪(g_i, g_j) ← 0
  end for
end for
//————Fill using M^1
for g_i ∈ G^1 do
  for g_j ∈ G^1 do
    M^∪(g_i, g_j) ← M^1(g_i, g_j)
  end for
end for
//————Fill using M^2
for g_i ∈ G^2 do
  for g_j ∈ G^2 do
    M^∪(g_i, g_j) ← M^2(g_i, g_j)
  end for
end for
//————Fill for G^1 ∩ G^2
for g_i ∈ G^∩ do
  for g_j ∈ G^∩ do
    M^∪(g_i, g_j) ← max(M^1(g_i, g_j), M^2(g_i, g_j))
  end for
end for
```

6.8.2. Results

Figures 6.31 and 6.32 on page 46 depict two streamgraph diagrams containing respectively, 30 and 24 layers (artist names) displaying music usage during the day of several users. We show the corresponding calculated importance matrices in figures 6.33a and 6.33b on page 47 and the *synthetized* matrix is in figure 6.34. This matrix is calculated for the union of layers' name. We display the two streamgraph diagrams with the colors permutation given by our assignment optimization using the *synthesized matrix*.

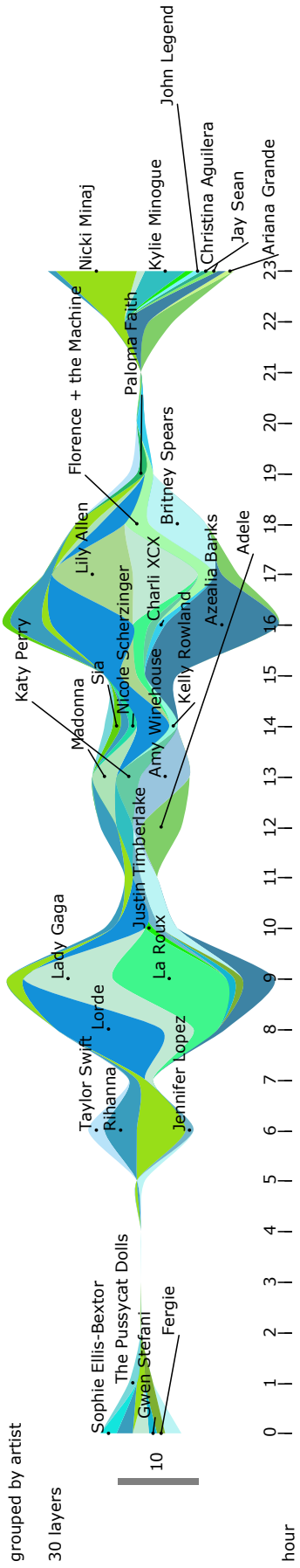


Figure 6.31: Streamgraph 1

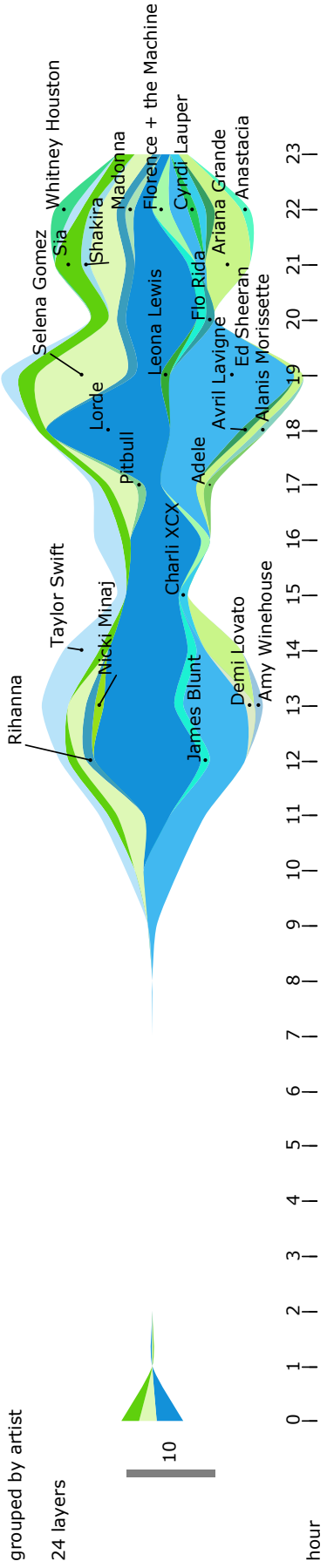
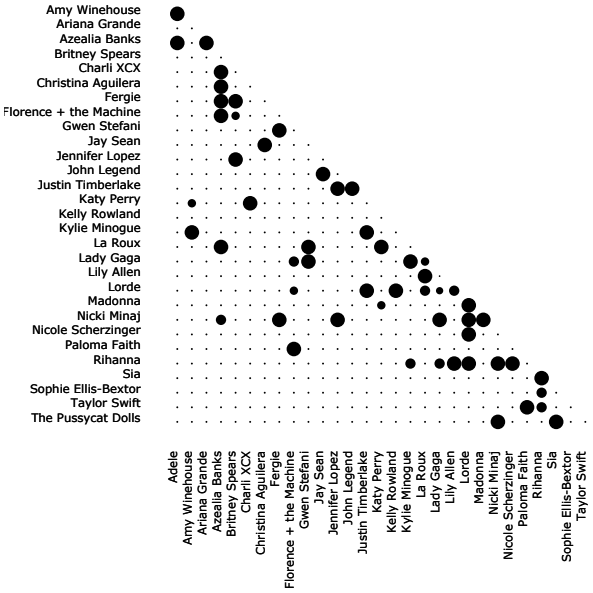
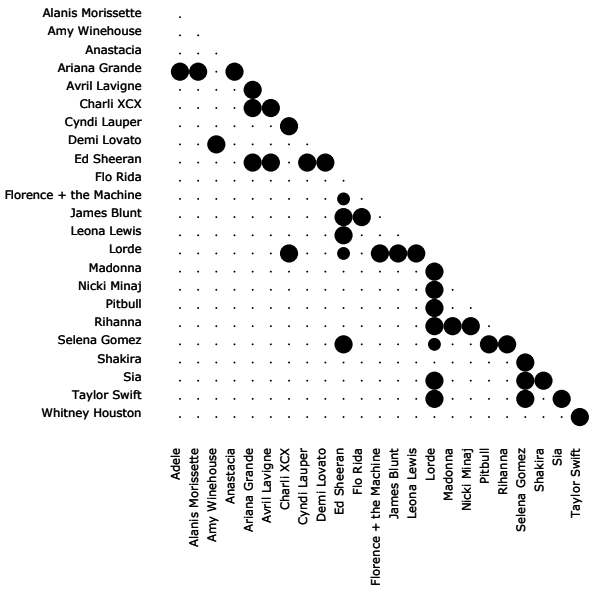


Figure 6.32: Streamgraph 2



(a) First streamgraph imp. matrix



(b) Second streamgraph imp. matrix

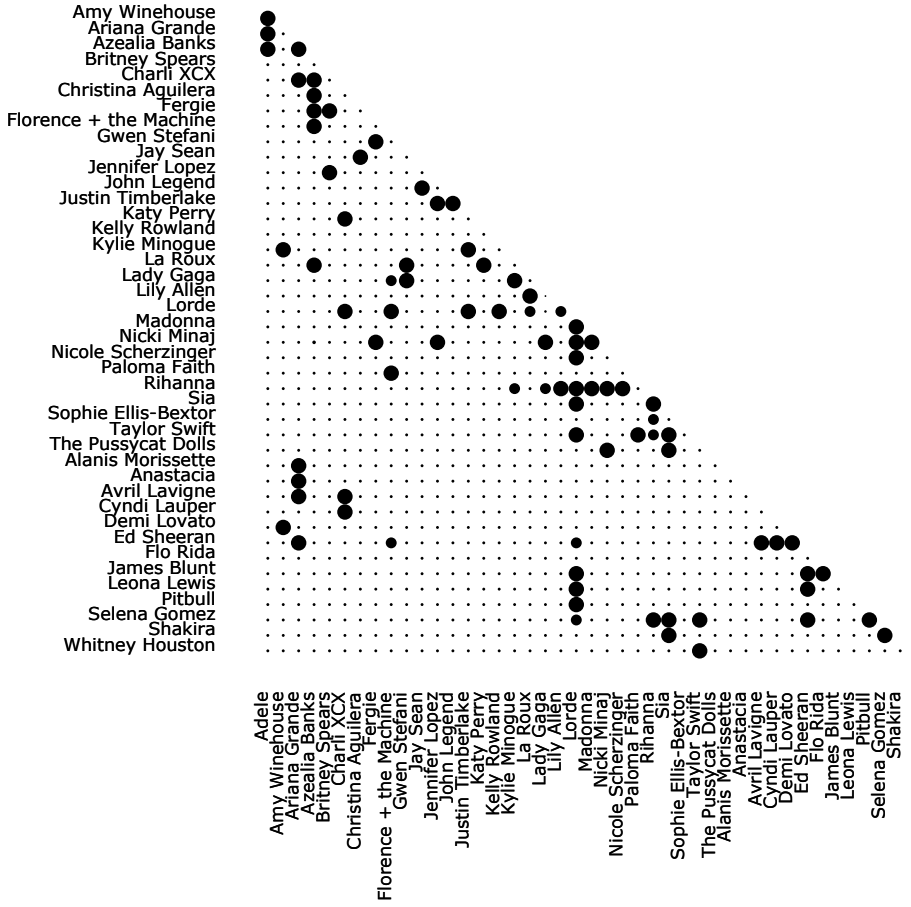


Figure 6.34: Synthetized imp. matrix

7

Implementation

7.1. Java Language and R language

First devised to be used in our *facetStreamgraph* software, we code the importance matrix calculation for streamgraph and genetic color-layer assignment optimization, in java using `processing`[1] framework.

With the potential extensions to other diagrams, we have coded R6 classes which are an implementation of class for language R, in order to provide a package to the community (available soon on github, the author expecting pros, cons, improvements and applications to other categorical diagrams).

The R6 classes are:

- an R6 class for color-class assignment optimization:
  - using `GA` package to estimate the best permutation by genetic optimization and PMX crossover reproduction,
  - which calculate color distance matrix using `Farver` package to compute the DE2000 color-distance matrix
- a base R6 class to handle importance matrix memory allocation, `getters` and `setters`
- inherited R6 classes for:
  - streamgraphs (with natural layers order, but extendable to other layer order)
  - polygonal maps. Coded into language R using the inverse of polygon area (but tested in java using perpendicular distance along the polygon frontier).
  - chord diagrams (limited to directional, so far)
  - line-charts (not poly-lines as in a "metro maps", so far)
  - simple neighbor diagrams (single bar chart and pie chart)

The following list of R packages which have been helping a lot in this research:

- R6
- Farver
- GA
- dplyr
- streamgraph
- viridis
- spdep
- farver
- chorddiag
- ggplot2
- tidyverse
- gtools

The synthetizing of multiple diagrams is coded in java but will be coded in R too. Scatterplot diagram is the next diagram to be included in the R package.

[1] `www.processing.org`

7.2. Optimization runtime

We perform an automatic exhaustive evaluation of the permutations fitness result for visualizations based on a few classes, $n \leq 8$, and for greater values of n, we run the genetic optimization.

The Colombus map (see figure 6.10) which is the most complicated case test (49 classes) has been calculated using 500 generations of 400 individuals, with a running time of 2 minutes on an iMac (Retina 5K, 27 inches, end 2015, 3.3 GHz Intel Core i5 quad core, 8 Go 1867 MHz DDR3) using Rstudio[2] and an R script based on inverse polygons area importance calculation and polygons connection (spdep R package) using package GA (without parallelism).

[2] https://www.rstudio.com/

8

Conclusion

The author has presented a novel framework that permits optimizing contrast color of classes objects for categorical visualization. Our method relies on a separation between palette color distances and the concept of *importance factors* expressing the need to get for a couple of classes a high color contrast. We construct the *importance factors* depending on the chosen kind of visualization and especially on the data geometrical properties associated.

We have presented categorical diagrams applications: streamgraphs, line charts, polygonal maps, chord diagrams.

Neither controlled user tests, nor comparisons with published methods have been performed. The author is aware of this important lack in this research report.

The author hopes that data scientists will embrace this novel framework, will adapt it to other diagrams and will improve the proposed diagram's importance matrix expressions and expand the energy expression.

So far, we have coded the diagrams' *importance matrix* calculation outside the diagram generation code (in java, or in R), but the proposed importance formulas could be directly inserted within the diagram generation code together with an export of the importance matrix.

The developed R6 classes coded in language R will be available soon on a github repository together with use examples of each of the presented diagrams.

The author will add other categorical diagrams like *alluvial* or *Sankey* diagrams to the framework and will implement published *scatterplots* formulas.

In the future, the author thinks that considering the diagram (and presented objects) subtended visual angle, and therefor the future context of the real life diagram viewing (diagram size, viewing distance, electronic device or printed on paper) would be inevitable. Evaluation of the distinguishability and communication from the software to the data scientist about the various problems that could arise with the generated diagram is also perhaps the next step.

References

[AA06] Natalia Andrienko and Gennady Andrienko. *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer Science & Business Media, 2006.

[Abe17] Guy J. Abel. "Estimates of Global Bilateral Migration Flows by Gender between 1960 and 20151". In: *International Migration Review* 52.3 (2017), pp. 809–852.

[AS16] Michaël Aupetit and Michael Sedlmair. "SepMe: 2002 New visual separation measures". In: *2016 IEEE Pacific Visualization Symposium, PacificVis 2016, Taipei, Taiwan, April 19-22, 2016*. Ed. by Chuck Hansen, Ivan Viola, and Xiaoru Yuan. IEEE Computer Society, 2016, pp. 1–8.

[Ber83] Jacques Bertin. *Semiology of Graphics*. University of Wisconsin Press, 1983. ISBN: 0299090604.

[BH11] David Borland and Alan Huber. "Collaboration-Specific Color-Map Design". In: *Computer Graphics and Applications, IEEE* 31 (Sept. 2011), pp. 7 –11.

[BH16] Marco Di Bartolomeo and Yifan Hu. "There is More to Streamgraphs than Movies: Better Aesthetics via Ordering and Lassoing". In: *Comput. Graph. Forum* 35.3 (2016), pp. 341–350. DOI: 10.1111/cgf.12910. URL: https://doi.org/10.1111/cgf.12910.

[BK57] M. Beckman and T.C. Koopmans. "Assignment problems and the location of economic activities". In: *Econometrica* 25 (1957), pp. 53–76.

[BM13] Matthew Brehmer and Tamara Munzner. "A Multi-Level Typology of Abstract Visualization Tasks". In: *IEEE Transactions on Visualization and Computer Graphics* 19 (Dec. 2013), pp. 2376–85.

[Bor+11] Rita Borgo et al. "Evaluating the Impact of Task Demands and Block Resolution on the Effectiveness of Pixel-based Visualization". In: *Visualization and Computer Graphics, IEEE Transactions on* 16 (Jan. 2011), pp. 963 –972.

[Bre94a] Cynthia Brewer. "Color Use Guidelines for Mapping and Visualization". In: vol. 2. London: Pergamon, Dec. 1994, pp. 123–147.

[Bre94b] Cynthia A. Brewer. "Guidelines for use of the perceptual dimensions of color for mapping and visualization". In: *Color Hard Copy and Graphic Arts III*. Ed. by Jan Bares. Vol. 2171. International Society for Optics and Photonics. SPIE, 1994, pp. 54 –63.

[Bre99] Cynthia A Brewer. "Color use guidelines for data representation". In: *Proceedings of the Section on Statistical Graphics, American Statistical Association*. 1999, pp. 55–60.

[BRT95] Lawrence Bergman, Bernice Rogowitz, and Lloyd Treinish. "A rule-based tool for assisting colormap selection". In: *Proceedings Visualization '95*. 1995, pp. 118–125.

[BT07] David Borland and M Taylor. "Rainbow Color Map (Still) Considered Harmful". In: *IEEE computer graphics and applications* 27 (Mar. 2007), pp. 14–7.

[Bu+21] Chuan Bu et al. "SineStream: Improving the Readability of Streamgraphs by Minimizing Sine Illusion Effects". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2021), pp. 1634–1643. DOI: 10.1109/TVCG.2020.3030404.

[Bur+98] R.E. Burkard et al. "The quadratic assignment problem". English. In: *Handbook of Combinatorial Optimization*. Vol. 2. Netherlands: Kluwer Academic Publishers, 1998, pp. 241–337.

[BW08] Lee Byron and Martin Wattenberg. "Stacked Graphs - Geometry & Aesthetics". In: *IEEE Transactions on Visualization and Computer Graphics* 14 (Oct. 2008), pp. 1245–1252.

[Cue+18] Erick Cuenca et al. "MultiStream: A Multiresolution Streamgraph Approach to Explore Hierarchical Time Series". In: *IEEE Transactions on Visualization and Computer Graphics* 24.12 (2018), pp. 3160–3173. DOI: 10.1109/TVCG.2018.2796591.

[DBH14] Cagatay Demiralp, Michael Bernstein, and Jeffrey Heer. "Learning Perceptual Kernels for Visualization Design". In: *IEEE Transactions on Visualization & Computer Graphics* 20 (Nov. 2014).

[DW98] John Dzubera and Darrell Whitley. "Advanced Correlation Analysis of Operators for the Traveling Salesman Problem". In: *Parallel problem solving from nature* III (July 1998).

[Fan+16] Hui Fang et al. "Categorical Colormap Optimization with Visualization Case Studies". In: *IEEE Transactions on Visualization & Computer Graphics* 23 (Jan. 2016), pp. 1–1.

[GK15] Nicolas Greffard and Pascale Kuntz. "Visualizing a Set of Multiple Time Series with an Aggregate Stacked Graph." In: *IV*. Ed. by Ebad Banissi et al. IEEE Computer Society, 2015, pp. 68–74. ISBN: 978-1-4673-7568-9. URL: http://dblp.uni-trier.de/db/conf/iv/iv2015.html#GreffardK15.

[GLS17] Connor Gramazio, David H. Laidlaw, and Karen B. Schloss. "Colorgorical: Creating discriminable and preferable color palettes for information visualization". In: *IEEE Trans. Vis. Comput. Graph.* 23.1 (2017), pp. 521–530.

[Hav+02] S. Havre et al. "Themeriver: Visualizing Thematic Changes in Large Document Collections". In: *Visualization and Computer Graphics, IEEE Transactions on* 8 (Feb. 2002), pp. 9–20. DOI: 10.1109/2945.981848.

[HB03] Mark Harrower and Cynthia Brewer. "ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps". In: *Cartographic Journal The* 40 (June 2003), pp. 27–37.

[HS12a] Christopher G. Healey and Amit P. Sawant. "On the Limits of Resolution and Visual Angle in Visualization". In: *ACM Trans. Appl. Percept.* 9.4 (2012). ISSN: 1544-3558. DOI: 10.1145/2355598.2355603. URL: https://doi.org/10.1145/2355598.2355603.

[HS12b] Jeffrey Heer and Maureen Stone. "Color naming models for color selection, image editing and palette design". In: *Conference on Human Factors in Computing Systems - Proceedings* (May 2012).

[Hum+13] Mathias Hummel et al. "Comparative Visual Analysis of Lagrangian Transport in CFD Ensembles". In: *IEEE transactions on visualization and computer graphics* 19 (Dec. 2013), pp. 2743–52.

[Hur+10] Christophe Hurter et al. "An automatic generation of schematic maps to display flight routes for air traffic controllers: structure and color optimization". In: *Proceedings of the International Conference on Advanced Visual Interfaces, AVI 2010, Roma, Italy, May 26-28, 2010*. Ed. by Giuseppe Santucci. ACM Press, 2010, pp. 233–240.

[Kim+14] Hye-Rin Kim et al. "Perceptually-based Color Assignment". In: *Comput. Graph. Forum* 33.7 (2014), pp. 309–318. DOI: 10.1111/cgf.12499.

[LG16] Philippe Le Guern. *Où va la musique ?* Presses des Mines, 2016. ISBN: 9782356714077.

[Lin+13] Sharon Lin et al. "Selecting Semantically-Resonant Colors for Data Visualization". In: *Computer Graphics Forum* 32 (June 2013).

[LK21] Eric Languenou and Pascale Kuntz. ""StreamByFacet", un nouvel outil interactif d'exploration de données temporelles par une navigation de type facettes et un affichage en streamgraph". In: *21ème édition de la conférence (EGC) Extraction et Gestion des Connaissances 2021*. Montpellier, France, Jan. 2021. URL: https://hal.archives-ouvertes.fr/hal-03346943.

[LKG15] Eric Languenou, Pascale Kuntz, and Nicolas Greffard. "Music archipelago, a facet-like music library comparison tool". In: *Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on*. 2015, pp. 534–535. DOI: 10.1109/RCIS.2015.7128925.

[LSS13] Sungkil Lee, Mike Sips, and Hans-Peter Seidel. "Perceptually Driven Visibility Optimization for Categorical Data Visualization". In: *IEEE transactions on visualization and computer graphics* 19 (Oct. 2013), pp. 1746–57.

[Lu+21] Kecheng Lu et al. "Palettailor: Discriminable Colorization for Categorical Data". In: *IEEE transactions on visualization and computer graphics* 27.2 (2021), 475—484. ISSN: 1077-2626. DOI: 10.1109/tvcg.2020.3030406.

[Mit+15] Sebastian Mittelstadt et al. “ColorCAT: Guided Design of Colormaps for Combined Analysis Tasks”. In: *Eurographics Conference on Visualization (EuroVis) - Short Papers*. Ed. by E. Bertini, J. Kennedy, and E. Puppo. The Eurographics Association, 2015.

[MSK04] Barbara Meier, Anne Spalter, and David Karelitz. “Interactive color palette tools”. In: *IEEE computer graphics and applications* 24 (June 2004), pp. 64–72.

[New55] Sidney M Newhall. “Width and area thresholds of discrimination of two colors.” In: *Journal of General Psychology* 52 (1955), p. 247.

[OAH11] Peter O’Donovan, Aseem Agarwala, and Aaron Hertzmann. “Color compatibility from large datasets”. In: *ACM Trans. Graph.* 30.4 (2011), p. 63.

[OAH14] Peter O’Donovan, Aseem Agarwala, and Aaron Hertzmann. “Collaborative filtering of color aesthetics”. In: *Proceedings of the Workshop on Computational Aesthetics, CAe ’14, Vancouver, British Columbia, Canada, August 8-10, 2014*. Ed. by David Mould. ACM, 2014, pp. 33–40.

[QH87] Philip Quinlan and Glyn Humphreys. “Visual search for targets defined by combinations of color, shape, and size: An examination of the task constraints on feature and conjunction searches”. In: *Perception and psychophysics* 41 (June 1987), pp. 455–72.

[Rhe00] Penny Rheingans. “Task-based Color Scale Design”. In: *Proceedings of SPIE - The International Society for Optical Engineering* (Apr. 2000).

[RT96] Bernice Rogowitz and Lloyd Treinish. “How Not to Lie with Visualization”. In: *Computers in physics* 10 (June 1996).

[RT99] Bernice Rogowitz and Lloyd Treinish. “Data Visualization: The End of the Rainbow”. In: *Spectrum, IEEE* 35 (Jan. 1999), pp. 52 –59.

[Saj+12] Behzad Sajadi et al. “Using Patterns to Encode Color Information for Dichromats”. In: *IEEE transactions on visualization and computer graphics* 19 (Mar. 2012).

[San+11] Jibonananda Sanyal et al. “Noodles: A Tool for Visualization of Numerical Weather Model Ensemble Uncertainty”. In: *IEEE transactions on visualization and computer graphics* 16 (Jan. 2011), pp. 1421–30.

[Sau+10] Catherine Sauvaget et al. “Segmented Images Colorization Using Harmony”. In: *2010 Sixth International Conference on Signal-Image Technology and Internet Based Systems*. 2010, pp. 153–160. DOI: 10.1109/SITIS.2010.35.

[Sed+12] Michael Sedlmair et al. “A Taxonomy of Visual Cluster Separation Factors”. In: *Comput. Graph. Forum* 31.3 (2012), pp. 1335–1344.

[SM07] Moritz Stefaner and Boris Muller. “Elastic Lists for Facet Browsers”. In: *Proceedings of the 18th International Conference on Database and Expert Systems Applications*. DEXA ’07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 217–221. ISBN: 0-7695-2932-1.

[SP11] Karen Schloss and Stephen Palmer. “Aesthetic response to color combinations: Preference, harmony, and similarity”. In: *Attention, perception and psychophysics* 73 (Feb. 2011), pp. 551–71.

[SS16] Vidya Setlur and Maureen C. Stone. “A Linguistic Approach to Categorical Color Assignment for Data Visualization”. In: *IEEE Trans. Vis. Comput. Graph.* 22.1 (2016), pp. 698–707.

[Sto12] Maureen Stone. “In Color Perception, Size Matters”. Undetermined. In: *IEEE Computer Graphics and Applications* 32.2 (2012), pp. 8 –13. DOI: 10.1109/MCG.2012.37.

[SUS08] Moritz Stefaner, Thomas Urban, and Marc Seefelder. “Elastic Lists for Facet Browsing and Resource Analysis in the Enterprise”. In: *Proceedings of the 2008 19th International Conference on Database and Expert Systems Application*. DEXA ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 397–401. ISBN: 978-0-7695-3299-8.

[SWD05] Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. “The CIEDE2000 color-difference formula : Implementation notes, supplementary test data, and mathematical observations”. In: *Color Research and Application* 30 (2005), pp. 21–30.

[Sza18] Danielle Albers Szafir. "Modeling Color Difference for Visualization Design". In: *IEEE transactions on visualization and computer graphics* 24.1 (2018), 392—401. ISSN: 1077-2626. DOI: 10.1109/tvcg.2017.2744359. URL: https://doi.org/10.1109/TVCG.2017.2744359.

[TFS08] Christian Tominski, Georg Fuchs, and H. Schumann. "Task-Driven Color Coding". In: *2008 12th International Conference Information Visualisation*. Aug. 2008, pp. 373 –380. ISBN: 978-0-7695-3268-4.

[Tuf90] Edward R. Tufte. *Envisioning Information*. Cheshire, Conn.: Graphics Press, 1990.

[US15] Dr. Anantkumar Umbarkar and P. Sheth. "CROSSOVER OPERATORS IN GENETIC ALGORITHMS: A REVIEW". In: *ICTACT Journal on Soft Computing (Volume: 6 , Issue: 1)* 6 (Oct. 2015).

[Usm+20] Sheema Usmani et al. "Data-Driven Ranking and Visualization of Products by Competitiveness". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 13640–13641. URL: https://ojs.aaai.org/index.php/AAAI/article/view/7107.

[Wan+08] L. Wang et al. "Color design for illustrative visualization". In: *Visualization and Computer Graphics, IEEE Transactions on* 14.6 (2008), pp. 1739–1754.

[Wan+18] Yunhai Wang et al. "Optimizing Color Assignment for Perception of Class Separability in Multiclass Scatterplots". In: *IEEE Transactions on Visualization and Computer Graphics* PP (Aug. 2018), pp. 1–1.

[War12] Colin Ware. *Information Visualization: Perception for Design*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012. ISBN: 9780123814647.

[War88] Colin Ware. "Color Sequences for Univariate Maps: Theory, Experiments and Principles". In: *Computer Graphics and Applications, IEEE* 8 (Oct. 1988), pp. 41–49.

[Zen+19] Qiong Zeng et al. "Data-Driven Colormap Optimization for 2D Scalar Field Visualization". In: *2019 IEEE Visualization Conference (VIS)*. 2019, pp. 266–270. DOI: 10.1109/VISUAL.2019.8933764.

[ZH16] Liang Zhou and Charles D. Hansen. "A Survey of Colormaps in Visualization". In: *IEEE Trans. Vis. Comput. Graph.* 22.8 (2016), pp. 2051–2069.

[ZHM09] Achim Zeileis, Kurt Hornik, and Paul Murrell. "Escaping RGBland: Selecting colors for statistical graphics". In: *Computational Statistics and Data Analysis* 53 (July 2009), pp. 3259–3270.