

GeoLinter: A Linting Framework for Choropleth Maps

Fan Lei, Arlen Fan, Alan M. MacEachren, Ross Maciejewski

Abstract—Visualization linting is a proven effective tool in assisting users to follow established visualization guidelines. Despite its success, visualization linting for choropleth maps, one of the most popular visualizations on the internet, has yet to be investigated. In this paper, we present GeoLinter, a linting framework for choropleth maps that assists in creating accurate and robust maps. Based on a set of design guidelines and metrics drawing upon a collection of best practices from the cartographic literature, GeoLinter detects potentially suboptimal design decisions and provides further recommendations on design improvement with explanations at each step of the design process. We perform a validation study to evaluate the proposed framework's functionality with respect to identifying and fixing errors and apply its results to improve the robustness of GeoLinter. Finally, we demonstrate the effectiveness of the GeoLinter - validated through empirical studies - by applying it to a series of case studies using real-world datasets.

Index Terms—Choropleth Maps, Visualization Linting, Automated Visualization Design, Visualization Recommendation

1 INTRODUCTION

THE rise in visualization popularity has coincided with the rise in easy to use programming languages and software that have opened the design door to novices and experts alike. This has led to data visualizations becoming an extremely popular means of sharing and explaining data. However, the ease of creating visualizations has also enabled the creation of ill-designed visualizations as inexperienced designers who lack knowledge of visualization principles may create graphics that are poorly formed. To help overcome this knowledge gap, recent work in the visualization community has popularized the idea of visualization linting [1], [2], [3] where potential design issues are highlighted in the same manner as coding errors using a linting feature that constantly checks to see if the output follows design guidelines. When a violation of a design guideline occurs, the linting tools suggest corrections and alternatives to potentially improve the design.

Given the frequency of visualization construction errors [2], researchers have begun to develop tools that identify and fix [1] malformed charts. Recent work has developed tools to recommend visualizations based on statistical properties of the data [4] or how the visualization is used [5]. There is also an automated approach for producing various types of visualizations [6], capable of generating graphics without manual definitions. Unfortunately, these tools [1], [2] only cater to simple chart types (e.g. line, bar, pie chart), missing one of the most common visualizations: the choropleth map. Research has shown that upwards of 30% of D3 visualizations on the internet are choropleth maps, making them the most used visualization type in the D3 category [7]. Given the prevalence of choropleth maps, we believe that

the visualization research and practice community can benefit from a framework for linting choropleth maps.

In this paper, we propose GeoLinter, a choropleth map linting framework. GeoLinter borrows concepts from previous linting tools [1], [2] and expands on these concepts to support the design of choropleth maps. We first define a set of best practices and performance measures and then take an interactive approach in detecting and correcting suboptimal designs. Due to the numerous design considerations that go into creating a choropleth map (e.g. the projection, classification method, color selection), GeoLinter divides the design process by first detecting encoding errors and then providing guidance on design choices. We note that GeoLinter is intended to help designers without cartographic experience avoid blunders, rather than to support novel mapping strategies.

Visualization linters are meant to assist in creating guideline-abiding visualizations that adhere to accepted design guidance. While other linters may exist [1], [2], they only cater to simple chart types such as bar and line charts. GeoLinter serves to expand on visualization linting by adding support for choropleth maps. By promoting a solid understanding of cartographic design principles among designers, it helps ensure the creation of visualizations that conform to essential rules. This is important because there are many visualizations that do not conform to guidelines in both the academic and professional community (e.g. VisLies), and visualization software systems often have rulebases for information visualization broadly, but not for choropleth maps or any other thematic map types specifically [1], [2], [3]. As such, our work is of significant relevance to the broad visualization community, with the target audience including visualization practitioners without cartography training.

To demonstrate our approach, we implement the proposed GeoLinter for VegaLite-based specifications and illustrate its effectiveness through a variety of case studies. We carried out a study to gather insights and observations on

- F. Lei, A. Fan, and R. Maciejewski are with the School of Computing, Informatics & Decision Systems Engineering, Arizona State University. E-mail: {flei5,afan5,rmacieje}@asu.edu.
- A.M. MacEachren is with Pennsylvania State University. E-mail: maceachren@psu.edu.

Manuscript received MM DD, 2022; revised MM DD, 2022.

the maps produced by our target audience. Based on the results and feedback from an expert cartographer, we refined GeoLinter to enhance its reliability. Our contributions include:

- GeoLinter, a framework for detecting, annotating, and revising potential design flaws in choropleth maps;
- Guidelines for design parameters in choropleth maps, informed by cartographic literature, and;
- A validation study exploring the functionality of GeoLinter in linting and education.

2 RELATED WORK

In this section, we summarize related works on visualization linting and recommendation tools.

2.1 Visualization Linting

Originally, “linters” were for the automated checking of source code for syntax and stylistic errors. Over the years, researchers saw a need to extend linting capabilities beyond just programming languages. Before its use in visualization, linting tools were developed for data in tabular form. CheckCell [8] and ExcelLint [9] use an outlier and anomaly detection approach to find potential errors in spreadsheet entries. Later, with the onset of large databases, ConTest [10] was proposed for continuous-data-testing, preventing errors in large databases. Due to the importance of data preprocessing in creating robust visualizations, this work set the stage for later research in visualization linting.

McNutt and Kindlmann [3] were one of the first to introduce a visualization linter. Their work defines a set of rules that consider the aesthetics and readability of a visualization and outputs a binary decision with respect to the feasibility of various simple VegaLite [11] chart types such as bar charts and line charts. VisuLint [2] flags visualization errors by drawing a red squiggly line over the location of the error, similar to how Microsoft Word sketches a red line underneath a misspelled word. VizLinter [1] accepts a VegaLite specification as input and translates it into answer set programming (ASP) facts to detect violated rules. It then proceeds to use a fixer engine to correct the rules to output a corrected specification. While our work draws ideas and design considerations from previous linting systems, the focus is on extending linting to choropleth maps, which, to date, remain unsupported in visualization linting.

2.2 Visualization Recommendation

Visualization recommendation systems are another approach to supporting the visualization design process, where visualization recommendation systems attempt to compute (and recommend) a visualization that adheres to best principle design practices. Rule-based recommender systems pick a layout by following a set of rules defined by domain experts. VizDeck [12] is one such system that accomplishes this by suggesting a set of appropriate visualizations given the statistical properties of the data. Voyager [4] takes a similar approach as VizDeck but instead emphasizes browsing, which facilitates exploratory tasks.

Later, hybrid approaches were developed, which combine learning and manually defined rulesets for visualization. For example, Moritz et al. [13] propose Draco, a constraint-based system that learns to assign a score to visualizations by rewarding charts that follow graphical perceptual principles and penalizing rule violations. Dziban [14] expands on Draco by adding chart similarity logic, enabling chart recommendations to remain similar to an anchor chart, making the results more predictable. Finally, recent research has led to the development of machine learning based systems. Qian et al. [15] pioneered the first machine learning based visualization recommender system trained on a large corpus of datasets and visualizations, which supports primitive chart types such as line charts, bar charts, and scatterplots. Li et al. [16] propose KG4Vis, a system that uses a knowledge graph to address the black-box nature of machine learning based recommender systems in an attempt to make the results more explainable. However, these recommendation systems focus on simple chart types (e.g. bar charts, box plots, heatmaps, histograms, line charts and scatter plots), with no support for choropleth maps.

3 CHOROPLETH MAP DESIGN

In this section, we outline all of the components necessary to create a choropleth map. We discuss design considerations, visualization grammar, and finally, our design goals.

3.1 Design Considerations

Choropleth maps are maps that display statistical data for enumeration units (e.g., states, counties) typically using an sequential color scheme to shade each area to reflect the class into which each area’s data value fits [17]. In Slocum et al.’s [18] textbook on Cartography and Geographic Visualization, the authors outline major topics for choropleth maps: data classification, factors for selecting a color scheme, and details of color specification. We condense the map elements into three major design considerations: (1) classification method, (2) symbology, and (3) projection.

(C1) Classification Method: Before applying a data classification method, the designer must choose the number of classes, or unique shades of color, in the map [19]. The choice of classification method is a decades-long debate that ultimately affects the visual impression and reader interpretation of the map [20]. An improper choice of the number of classes or data classification method may fail to portray key trends and patterns that would otherwise be visible. For simple maps, cartographers generally recommend that the number of classes is confined between three (3) to seven (7) [18]. This recommendation is in line with Miller’s observation on the magic number 7 ± 2 , suggesting that there are cognitive limits on our capacity for processing distinct chunks of information, which is a foundational guideline in cartography [21]. Evans [20] recommends that five or fewer classes be used for novice map readers. ColorBrewer [22] supports 11 colors for diverging schemes and 9 colors for sequential schemes, all of which are colorblind safe. Thus, our framework suggests classed maps between 3-11 classes. Anything above 11 classes we consider to be an unclassed map, which maps the range of the data to a continuous

spectrum of colors. Classification methods divide rank-ordered data into a specific number of bins. In choropleth maps, these bins are then mapped to different shades of a selected color scheme. Zhang and Maciejewski [23] identify the problem that data values can be close to a classification boundary and a slight change in the classification method can result in different clusters. Cartographers have studied the performance of different classification strategies under varying use cases and data distributions. Equal intervals, which partitions the overall data range into intervals of equals-sized ranges, are best suited for uniform distributions [20] but can have inconsistent or unpredictable performance [24]. Mean standard deviation is best used when the distribution is unimodal [20]. Quantiles are one of the most versatile classification methods, suited for normal distributions [24]. Our framework uses properties of the data to recommend an appropriate classification.

(C2) Color and Symbology: In a choropleth map, the symbolic features refer to the fill colors, outline width, and text. Each of these features can directly influence the choropleth map's readability [25]. For example, choosing two fill colors that are too similar to each other may make comparison tasks difficult [26], [27]. Also, the choice of outline width for area boundaries can influence the degree to which overall patterns versus individual values are emphasized. A narrow stroke or no stroke for boundaries enhances overall patterns, but completely omitting the stroke might not be a good solution since it could result in the loss of essential information; when neighboring units have the same color hue, the absence of a delineating stroke can cause them to visually merge, hiding the information of individual enumeration units. Bold strokes direct attention to individual enumeration areas. The border frame should also be appropriately selected.

Color choice is crucial in creating choropleth maps. ColorBrewer [22] is a tool designed to support appropriate color palette selections for choropleth maps. Brychtová and Çöltekin [26] find that the more distinct colors are in choropleth maps, the faster readers are able to distinguish and complete comparison tasks. This effect was confirmed by a controlled, in-lab study with an eye-tracker [25]. Apart from color discriminability, other perceptual features of choropleth colors must also be taken into account. For example, red regions are perceived as having more area than green ones [28]. Notably, a foundational principle in designing sequential color schemes is that people tend to associate higher values with darker colors [29], [30], [31]. It follows that if the choropleth needs to show logical ordering of values, then rainbow colors must be avoided [32]. Thus, in GeoLinter, the palettes recommended are standard sequential and diverging palettes recommended by ColorBrewer [22].

(C3) Projection: Creating a map requires transforming from the sphere to a 2D plane leading to a variety of distortions. A popular method of quantifying distortion in map projection is the use of Tissot's indicatrix [33]. By projecting a circle from the spherical earth onto a two-dimensional map, the reader can see what kind of distortion occurs. Common map projections often take one of two extremes: (a) equal area – which preserves correct area everywhere but must distort angular relations around points, and (b) conformal –

which preserves angular relations around points but must distort area. Equal area projections are almost always preferable for choropleth maps since it is important to represent the size of places correctly as readers try to understand spatial patterns [34]. Conformal projections were originally developed for situations in which angular and directional relations are important. The most common projection on the web is the Mercator projection, a conformal projection designed to map the entire globe while also drastically distorting both area and shape, thus making it unsuited for data mapping [35]. However, recent internet tools have used other map projections and have even facilitated education about map projections for ordinary audiences [36]. There are situations in which compromise projections are chosen (e.g., when some distortion of areas is acceptable in exchange for pleasing appearance at the global scale or rhetorical effect). At the global scale, however, the best choice is now considered by many to be an equal area projection inspired by the attractive Robinson global compromise projection – the Equal Earth projection, jointly developed by Šavrič *et al.* [37]. In many cases, Šavrič *et al.*'s natural earth projections are also appropriate [38]. For regions smaller than the globe, the best projections for data maps are equal area projections designed specifically to minimize other distortion across the area of interest. For the USA, as an example, the Albers Conic Equal Area projection is appropriate as it minimizes size distortions of the map [39]. For small scale maps much smaller than the scale of the Earth, conformal projections such as Mercator can be used, as long as the distortion of the area is minimal. The specific equal area projection that works best will depend on the extent of the geographic area depicted (from global, through continental and national, to more local) as well as on the specific location on the globe and the shape of the region (e.g., an Albers equal area projection with standard parallels positioned appropriately is good for east-west extending places in mid-latitudes like the USA while a Lambert azimuthal equal-area projection with the center point in the center of the country is good for an elongated north-south region like Argentina). Our framework considers the geographic extent and recommends an appropriate projection for the map with the support of a projection wizard proposed by Šavrič *et al.* [40].

3.2 Grammar Considerations

Along with considerations directly related to choropleth map design, grammar and syntax must also be considered. While our framework is agnostic to grammar or encoding languages, we have chosen to instantiate our work using VegaLite [11], a high-level visualization grammar known for its conciseness and use in other extant linting tools [1], [2]. Within VegaLite, marks and encoding channels must be explicitly declared. For choropleth maps, the options for mark and encoding channel are confined to geographical shape and color, respectively. Position, orientation, and size are also considered, but are less customizable. For example, maps are generally positioned at the center. North is usually pointed up, and size is adjusted so that the map can fit within the screen. Thus, declarative errors for choropleth maps are limited. However, due to the open-ended nature

Name	Description	Action
DATA_URL	<code><data.url></code> must be nonempty.	The data file must be specified.
DATA_FEATURE	<code><data.format.feature></code> must be nonempty.	At least one geographical feature must be specified.
MARK	<code><mark></code> must have the value “geoshape”.	For a choropleth map, the only viable option for <code><mark></code> is “geoshape” in VegaLite.
COLOR_FIELD	<code><encoding.color.field></code> must be nonempty.	For a choropleth map, the color field must not be empty.
COLOR_TYPE	<code><color.type></code> should have the value “quantitative” for a choropleth map or “nominal” for a thematic map that portrays regions of categorical data. Nominal values should use categorical colors.	The <code><color.type></code> must be either quantitative or nominal.
NUM_CLASSES	The number of classes is outside of the recommended range 3-7. Unclassed maps are not allowed.	Display the top performing classification methods for 3-7 classes, sorted by GVF or Moran’s I.
LEGEND_COLOR	The legend contains 2 or more colors that are too similar.	Suggest an array appropriate color palette for the legend.
BORDER_COLOR	The border color is too similar to the legend colors.	Suggest an appropriate color palette for the legend, and reset the border color to black.
BG_COLOR	The background color is too similar to the border or legend colors	Suggest an appropriate color palette for the legend, and set the background color to white.
LOW_GVF	The classification GVF score is low. This is triggered if the current GVF is lower than the average GVF of all possible classification methods across all number of classes.	Display the top performing classification methods for 3-7 classes, sorted by GVF or Moran’s I.
PROJ*	The map projection is not appropriate for the given geographic extent.	Based on the geographic extent from GeoJSON, recommend the most appropriate projection with the support of a projection wizard [40].
DATA_NORM*	The map should portray a normalized value and not an absolute value.	Suggest data normalization techniques as recommended by Slocum et al. [18]
TITLE_LEGEND*	The legend should include the data units being mapped. The title should be present and preferably reference the map theme, location, and time being mapped.	Automatically label the legend using the units from the data normalization stage or a custom-defined label. Generates a title based on the units and enforces a non-empty title.

TABLE 1

The list of rule violations detected and corrected by GeoLinter. Encoding errors are listed above the double horizontal line and soft rules below. Rules marked by an asterisk were added in or refined after conducting a validation study (Section 5).

of the other design considerations (class-interval selection, symbology, and projection), solving these issues is a nontrivial problem. We introduce an engine that checks for required encoding prerequisites that are necessary for the choropleth map. Encoding errors refer to invalid grammar specifications that may lead to visualizations that cannot compile. We identify two major categories of encoding problems: 1) missing the key properties related to the map visualization in the specification, and 2) invalid property types and values for the choropleth map visualization.

3.3 Design Goals

Given these design considerations, it is clear that the design of a choropleth map is a multi-objective problem, and the considerations may vary according to task. There is no absolutely “correct” choropleth map [41]; however, there are multiple best practice design guidelines from the cartography community. We have identified three key design goals for building a linting framework to support choropleth map development. A summary of the guidelines can be found in Table 1.

G1: Detect and indicate encoding errors. Encoding errors prevent the choropleth map from compiling or rendering. The first goal is to detect these errors in the grammar

specification and indicate the location of the error along with an explanation as to why the map failed to render.

G2: Detect and explain suboptimal design parameters (C1, C2, C3). Poor choropleth map design can fail to portray important trends that need to be visualized. Our second goal is to identify any design choices (**C1: Classification**, **C2: Color and symbology**, **C3: Map projection**) that may be problematic. Not all visualization designers have experience in cartography, so we also support explaining domain-specific technical terms in an understandable way. We refer to the design parameters related to **C1**, **C2**, and **C3** as soft rules, and they are summarized in Table 1.

G3: Automatically fix the detected errors (C1, C2, C3). Surfacing the error alone is inadequate. For example, just indicating that the classification method is problematic does not help a novice mapmaker the visualization designer, as they may not even be aware of different classification methods. Our framework provides options to correct the errors by calculating a set of good classification methods and enabling their comparison. Mathematical definitions of our methods and metrics and a list of best practices are explained in-depth in Sections 4.1, 4.2, and 4.3.

G4: Support integration to GIS frameworks. A geographical information system (GIS) is a database containing geo-

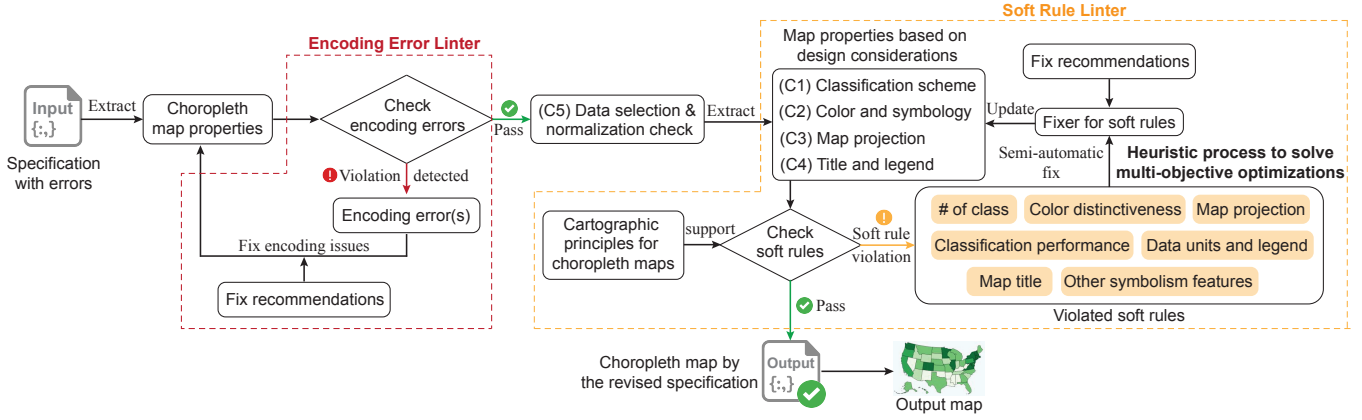


Fig. 1. The end-to-end pipeline of GeoLinter. The soft rule linter is the foundation of our framework, which offers guidance on design parameters based on best practices in cartographic literature.

graphic data with software tools for analyzing and visualizing those data [42]. GIS has become a standard for geoinformatics in both industry and academia. Our framework is designed to support GeoJSON files, an accepted format for defining geographical shapes, along with their non-spatial properties. GeoJSON naturally lends itself for automatic integration into existing GIS frameworks, strengthening the reach of our framework.

4 THE GEOLINTER FRAMEWORK

Based on the design goals, we have developed GeoLinter (Figure 1), a linting framework for choropleth maps. GeoLinter is based on the theory of choropleth map design and lints based on derived guidelines. A series of encoding errors and soft rules form the basis of our linting procedure. Table 1 summarizes all of the rules in GeoLinter and the workflow is documented in Figure 1. Figure 2 shows the interface of the framework.

To create our ruleset, we designed specifications to create a viable choropleth map using visualization grammar. We developed a set of encoding rules to satisfy goal G4. Five properties serve as a prerequisite to making a choropleth map within VegaLite: (1) the dataset must be specified; (2) at least one geographical feature must be specified; (3) the `<mark>` property must be “geoshape”; (4) the `<encoding.color.field>` property must be nonempty, and; (5) the `<color.type>` property must be either “quantitative” or “nominal”. The fifth rule enables designers to visualize both statistical and categorical data. We also highlight that our system supports ordinal data, which possesses an inherent order without a quantifiable distance between data points [43]. Since encoding errors prevent a choropleth map from rendering properly, resolving this step is mandatory. While other design parameters (e.g. color, background) are also required, VegaLite has predefined default values if they are not explicitly specified. Our framework automatically collects the value based on the VegaLite defaults. Here, our hard rules are functions of the grammar syntax and not design. We classify our design guidelines as soft rules and define them over a set of three guidelines to satisfy our design constraints and goals: classification guidelines C1, G2, G3, symbology guidelines C2, G2, G3, and map projection guidelines C3, G2, G3.

We first conceptualized and developed the initial framework of GeoLinter, aiming to help visualization designers prevent common mistakes in choropleth map creation. This initial iteration of the tool was created based on established cartographic guidelines and principles, as detailed in this section. Following the establishment of this initial version of GeoLinter, we initiated an evaluation study (Section 5). The purpose of this study was to gather feedback and to assess how effectively GeoLinter was assisting in reducing common design errors. The evaluation study’s methodology and its key findings are detailed. During the evaluation study, we identified certain limitations in our tool as well as areas where designers were still creating erroneous maps, indicating areas for potential improvement. To gain deeper insights, we not only relied on the feedback from the evaluation study but also incorporated the valuable inputs from our co-author, a cartography expert, and the reviews from the initial draft of this paper. Based on the feedback, we refined GeoLinter, enhancing its interface and updating guidelines, detailed in Section 6.

4.1 Classification Guidelines

Numerous classification methods for choropleth map design exist. We instantiate the most popular methods and define our linting recommendations on these methods based on the cartographic literature.

Equal Intervals - Divides attribute values into ranges of equal size [20]. For a given number of classes k , equal intervals classification divides the classes into intervals having a width $w = \frac{x_0 - x_{n-1}}{k}$, with the first class being $(-\infty, x_{n-1} - (k-1)w]$, and the last class being $(x_{n-1} - w, x_{n-1}]$, where x_0, x_{n-1} are the first and last entries in the dataset in sorted order. Although simple to construct and interpret, the result may include sparse or empty classes under datasets with extreme outliers [44].

Quantiles - Distributes a set of values into classes that have approximately $\lfloor \frac{n}{k} \rfloor$ members [20]. No classes are left empty or contain an unbalanced number of values. While quantiles avoids the problem of sparse classes, the intervals may be significantly different in range size, which can cause difficulties in interpretation [24]. Additionally, readers tend to assume that all places assigned to any given class are similar, but for skewed distributions, it is typical for at

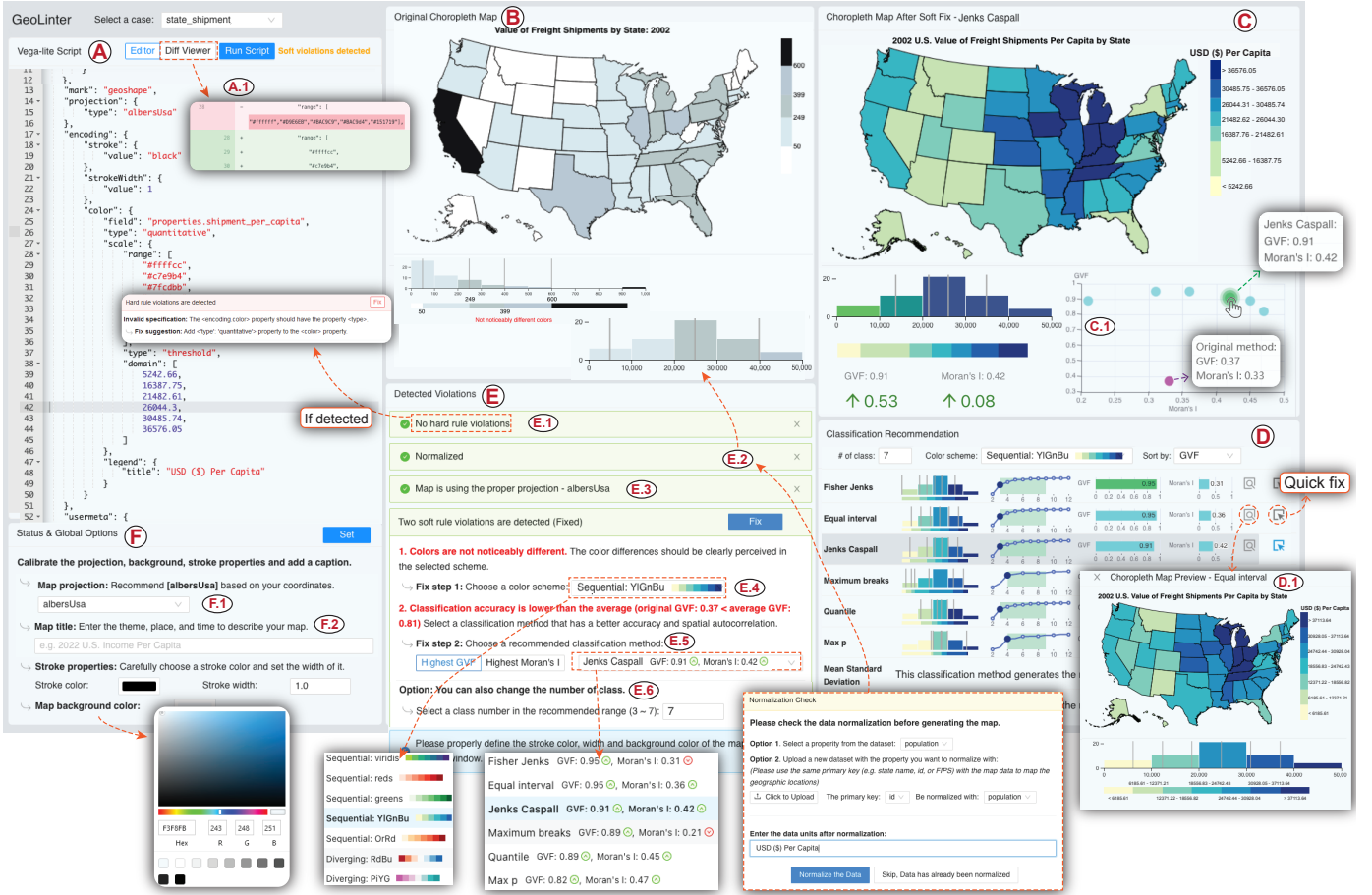


Fig. 2. The GeoLinter Interface: (A) the VegaLite code editor; (B) the original map view; (C) the map after applying soft fixes; (D) classification recommendations based on GVF and Moran's I; (E) detected violations with guides and prompts on map design improvements, and; (F) the status and global options panel. A choropleth map showing the value per capita of freight shipments in the U.S. by state 2002. In the original choropleth map design (B), the data classification accuracy is lower than the average value; the colors between bins are nearly indistinguishable; the map data has not been normalized and the data units are missing. After applying the suggested fixes from GeoLinter, the designer produces (C).

least one of the classes to contain values that are extremely different from one another.

Mean Standard Deviation - Uses the sample mean (\bar{x}) and standard deviation (s) to define class boundaries as distance from the mean [20]. The distance is usually a multiple of the standard deviation. A common setup for this method is to assign $k = 5$, where the classes c are defined as $c_i = \bar{x} + (2 - i)s, i \in \{0, 1, 2, 3, 4\}$. This classifier is best used when the data is normally distributed [20].

Maximum breaks - Considers the difference between sorted values to place breakpoints [45]. It considers how far apart each value is from the next one in the sorted sequence, then places $k - 1$ breaks between the values that are furthest apart from each other. Maximum breaks is an appropriate method to use when the objective is to partition the data points in each class such that they are separated from those in adjacent classes. It performs well when the distribution is not unimodal. This method is deterministic given k , as running this algorithm will always produce the same result under the same dataset. However, because this method only considers the top $k - 1$ differences between sorted values, within-group properties are often ignored [45].

Head tail - This algorithm [46] recursively partitions the data around iterative means until the distributions in each of

the classes are no longer heavy-tailed. This method proves effective with heavy-tailed distributions, such as log-normal and power law distributions [46].

Jenks-Caspall - Often considered a refined version of the original Jenks Natural Breaks method, the Jenks-Caspall classification seeks to minimize deviations around class means using a heuristic process [47]. While the Jenks Natural Breaks method focuses on finding "natural" clusters in data sets, aiming to reduce variance within classes and maximize variance between them, the Jenks-Caspall method introduces a more systematic approach to this objective.

Fisher-Jenks - A dynamic programming approach to minimize the sum of deviations around class medians, loosely based on Jenks-Caspall [47]. Whereas Jenks-Caspall may not provide an optimal classification for k classes, Fisher-Jenks is guaranteed to produce an optimal result.

Max-p - A classifier that uses the max-p [48] algorithm applied to map classification. In short, the max-p-regions algorithm aggregates n areas into a number of homogeneous regions, such that they satisfy a contiguity constraint while maximizing intraregional homogeneity. Since it is a heuristic, there is no guaranteed optimal solution.

Classification Performance Measures To determine the best choice of classifier, we utilize Goodness of Variance Fit

(GVF) and Moran's I, two numerical measures of classification results in choropleth map design correlated with perceptual accuracy [49], [50].

This is important, as relying solely on appearances and preferences when designing maps does not correlate with the accuracy and robustness of the map. Hegarty et al. [51] found that individual preferences, even those of domain experts, are not a good indicator of display effectiveness. They stress the importance of conducting additional objective measures of display efficiency. In a study by Brewer and Pickle [44], quantile classification resulted in the highest accuracy for basic analytical tasks such as data extraction and comparison. Jenks's natural breaks, equal interval performed satisfactorily, while methods such as shared area and box plot were significantly worse. Thus, the poorer performing class schemes were not used in our work, as separate studies have confirmed that they are perceptually less effective.

GVF - Smith's [24] GVF is defined as

$$GVF = 100 - ((SSW/SST) \times 100), \quad (1)$$

where *SST* (Sum of Squares Total) is the sum of deviations squared of individual data values from the overall mean, and *SSW* (Sum of Squares Within) is the sum of squared deviations of class data values from each class mean and then summing it for all classes. GVF is the preferred metric, and GeoLinter will mark the soft rule violation **LOW_GVF** if the GVF is lower than the average GVF of all possible classification methods across all number of classes. Since all values of GVF were calculated, the threshold for GVF can be set at a certain percentile threshold. For the purposes of GeoLinter being a proof-of-concept, we arbitrarily set the threshold as the average GVF.

Moran's I - When considering spatial correlation, Moran's I [52] is one of the most prominent measures, defined as:

$$I = \frac{N}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j w_{ij} (x_i - \bar{X})(x_j - \bar{X})}{\sum_i (x_i - \bar{X})^2}, \quad (2)$$

where *N* is the number of spatial regions, *i, j* are the indices of the regions, *x* is the variable of interest, \bar{X} is the mean of *x*, *w* is the matrix of spatial weights. The matrix *w* can vary depending on how the weights are chosen and what definition of contiguity is used. In our framework, we use the Queen [53] definition of contiguity-based neighbors, which means that two regions are neighbors if and only if they share at least one point. The other definition of contiguity is Rook [53], which counts two areas as neighbors if they share an edge. For irregular polygons (like most areal units encountered in practice), the differences between these definitions are slight. In order to deal with potential inaccuracies in the polygon file (such as rounding errors), using the queen criterion is recommended in practice [54]. Hence it is also the default for contiguity weights for polygon data in GeoDa [55] and ArcGIS Pro [56]. We note that while useful to visualize the value of Moran's I, it is not a measure of classification performance, and it does not take into account maximizing interclass variances nor minimizing intraclass variances. The purpose of including this measure into our

framework is to introduce an alternative metric that considers the spatial implications of the classification method without the more extreme solution of building spatial considerations into the classification formally.

Number of Classes In our framework, the number of classes is confined to 3-7, conforming to classical cartographic recommendations [50]. Classes fewer than 3 are binary or single color maps, which oversimplify the map, while more than 7 classes may make color comparison and data extraction tasks difficult due to the overwhelming number of unique colors. A map that does not use the recommended range will trigger the soft rule violation **NUM_CLASSES**. The GVF value also generally increases with the number of classes. We calculate the recommended threshold value for the number of classes by starting at 2 classes and incrementing by 1 until the GVF surpasses 0.5. This value is based on finding a knee-point [57], the point at which the cost to increase a parameter is no longer worth the increase in performance.

4.2 Symbology Guidelines

As a continuation of our soft rule linting, we detect and provide guidance on color and symbology, backed by cartographic literature.

Color Theory When choosing a color palette, the main design consideration is to choose between sequential and diverging color palettes. We recommend diverging palettes when there is a meaningful point to contrast with the other values in the data [22]. For example, a map that spans both negatives and positives would likely use the diverging scheme, as zero would be the reference point. All other schemes should use a sequential color palette. Diverging and sequential schemes should be chosen on a per-task basis [58]. For example, if a designer wants to emphasize below and above-average areas in the map, a diverging color scheme would be appropriate, where the average value would be the reference point. Ordinal data can also use sequential or diverging schemes, but categorical data should only use qualitative color scheme. These recommendations are programmed as prompts into GeoLinter. We use various palettes from ColorBrewer [22]. ColorBrewer color palettes are optimized for white backgrounds. Due to the rise in popularity in dark backgrounds in interfaces, CARTO¹ supports colors against non-white backgrounds, using the foundational color schemes from ColorBrewer.

Color Discriminability Color discriminability is primarily used for comparison tasks. Colors that are too similar to each other are not distinguishable by the human eye. ColorBrewer already includes schemes that have been empirically tested to be discriminable on maps. We derive three rules based on color discriminability: (1) **LEGEND_COLOR**, which is violated when two colors from each class are too similar to be distinguishable, (2) **BORDER_COLOR**, which is violated when the border color is too similar to any of the colors in the legend or the background, and (3) **BG_COLOR**, which is violated when the background color is too similar to any of the colors in the legend or the border. Any of these errors are triggered when custom schemes barely meet the discriminability criterion. To determine if

1. carto.com/carto-colors

colors are too similar, we use a model by Stone et al. [59] that determines if two colored glyphs can be differentiated faithfully. We check each pairwise color combination in the legend to determine if this criterion is met.

Symbology Border outline width is another design parameter that affects the interpretability of a map. Overly thick strokes emphasize the individual enumeration areas, while a narrow stroke enhances overall patterns. There are limited recommendations on outline width from the cartographic literature, and GeoLinter simply enables manual adjustment of the border width. Finally, the interface also allows the adjustment of frame width and margins.

4.3 Map Projection Guidelines

The map projections supported by the GeoLinter framework include Albers (a conic equal area projection that minimizes shape distortion for features in the mid-latitudes and extending more east-west than north-south) [60]; Albers USA [61] (a composite USA-centric projection configured for the lower 48 states, Hawaii, and Alaska); Equal Earth [37]; Natural Earth (a compromise pseudocylindrical map projection that is designed to depict the entire earth while balancing area and shape distortion) [38], and; Mercator. Although the most popular projection on the web is the Mercator, it is not appropriate for choropleth maps due to its excessive area distortion. Based on the GeoJSON data, GeoLinter will list appropriate projections and will mark the soft rule violation **PROJ** if the current map projection is not an equal area projection. The best choice of projection for maps of the entire world is the Equal Earth projection due to its equal-area property combined with good handling of other distortion and pleasing appearance for global maps. Thus, it will be our default option. The Albers USA projections is recommended for USA maps, and attempts to minimize other kinds of distortion across the latitude-longitude range containing the USA. It is implemented as an illustration of how GeoLinter can handle maps for specific non-global places.

This guideline underwent a major change (see Table 1) and is further discussed in Section 6.

4.4 Implementation

The frontend was implemented using ReactJS and D3 [61] and consists of six panels, as illustrated in Figure 2. Each panel serves a unique purpose in helping to visualize the different elements and functionalities of GeoLinter.

VegaLite Script: Figure 2A contains the VegaLite specification of the choropleth map. There is an editor and diff viewer panel (A.1) that tracks the changes that are made to the script. The reason for including the code view is for easy embedding into the web. For example, a designer can easily use the GUI interface to customize the map, while the code view automatically updates. The code for the final map is readily available for publishing.

Original Choropleth Map: Figure 2B is a static image of the choropleth map that may or may not have soft rule violations. A histogram view of the original data distribution, class breaks, and colors used is provided.

After Soft Fix: Figure 2C contains the final output of the map and mirrors the specification as defined by the VegaLite Script panel. Whereas the **Original Choropleth Map** view does not change, this view and the script are both updated whenever a fix is applied. A histogram is also shown here to assist in understanding the new class breaks. At the bottom left corner is a table detailing the change of GVF and Moran's I values. At the bottom right is a chart showing the GVF or Moran's I value of all possible classification methods. The original method is shown as a purple dot, and the current method is shown in green, allowing for a quick comparison of performance.

Classification Recommendation: Figure 2D displays all of the classification methods (Section 4.1), given the number of classes, with their corresponding performance metric values. Prior to loading this, the GVF and Moran's I of all methods for all number of classes 3-11 were calculated using the PySAL [53] Python package, a library for geospatial data science. Each method displays a histogram, plots a line chart with the number of classes on the x-axis and GVF on the y-axis, and shows the raw values for GVF and Moran's I. For each classification method used, a green box in the line chart indicates a range representing the recommended number of classes. Separately, we have organized the classification methods themselves in descending order, not based on the number of classes, but according to their respective GVF or Moran's I value. There is also the option to preview the map and apply a quick fix (D.1).

Detected Violations: All detected encoding and soft rule violations are displayed in Figure 2E. We employ step-by-step narrative visual prompts to recommend actions. Any changes in this view will automatically update the VegaLite script and the After Soft Fix View. We note that GeoLinter carries out its rule evaluations independently. That is, each detected violation is evaluated separately. As an illustration, consider the case of low classification performance (**LOW_GVF**). This violation is identified and evaluated independently of any others. While multiple approaches may address this issue - such as implementing a more appropriate classification method or adjusting the number of classes (**NUM_CLASSES**) - each potential solution is regarded as a separate action and does not influence the evaluation of other violations or solutions.

Status & Global Options: Figure 2F contains settings for the map projection, caption or title, border stroke settings, and background color. Any changes in this view will also update the VegaLite script and the After Soft Fix View.

5 EVALUATION

We conducted a validation study to gauge whether potential map designers can improve on flawed map designs. More specifically, the purpose of this study is to assess whether GeoLinter is effective at steering audiences away from cartographic blunders. The feedback from this study will inform the next iteration of GeoLinter (Section 6). In our study, participants used the GeoLinter framework to fix flawed choropleth maps. Quantitative and qualitative measures were captured with respect to number of linting errors corrected and overall satisfaction with the framework.

Task	Correction	Time (s)	δ GVF	δ Moran's I	# Classes	Class Methods	Colors	Projection
1	98.3%	$\mu = 268$, $\sigma = 117$	$\mu = 0.11$, $\sigma = 0.03$	$\mu = -0.01$, $\sigma = 0.04$	$\mu = 5.83$, $\sigma = 0.91$	11 Fisher-Jenks 10 Jenks-Caspall 4 Maximum breaks 4 Quantile 1 Equal Intervals	12 greens 6 reds 6 viridis 3 YiGnBu 2 OrRd 1 RdBu*	9 Equal Earth 5 Albers 4 Natural Earth 1 Albers USA 11 Mercator
2	95.0%	$\mu = 223$, $\sigma = 130$	$\mu = 0.21$, $\sigma = 0.05$	$\mu = -0.11$, $\sigma = 0.02$	$\mu = 5.57$, $\sigma = 1.07$	27 Fisher-Jenks 3 Equal Intervals	15 viridis 12 greens 1 RdBu* 1 reds 1 YiGnBu	13 Equal Earth 5 Natural Earth 3 Albers USA 1 Albers 8 Mercator
3	100%	$\mu = 142$, $\sigma = 68$	$\mu = 0$, $\sigma = 0$	$\mu = 0$, $\sigma = 0$	$\mu = 6$, $\sigma = 0$	30 Jenks-Caspall	30 reds	28 Albers USA 2 Albers

TABLE 2

An overview of the quantitative analysis of the experiment results. An asterisk next to the color scheme indicates a diverging palette. The number of participants choosing the Mercator projection is marked in red, due to strong recommendations against this kind of projection.

5.1 Participants

We recruited 30 participants for the study (24 male, 6 female, $M_{age} = 27.5$, $SD_{age} = 4.21$). Recruitment targeted computer science students who had familiarity with programming, but no formal training in cartographic design. Participants were paid \$10 for the study, which took approximately 30 minutes including training for an effective rate of \$20/hr. 8 of the participants waived the payment. The rationale for recruiting this demographic was to collect initial feedback from audiences with no formal training in cartography. We believe that this demographic is justified for a preliminary evaluation because we are not conducting a full-scale usability study. Instead, we are simply collecting feedback for the sole purpose of improving GeoLinter's next iteration.

5.2 Procedure/Protocol

Our study consists of five phases (as outlined in Figure 3): Pretest; Training; Quiz; Case Studies, and; Post-test and Survey. The study begins with a pre-test questionnaire, gauging the participant's skill level and familiarity with choropleth

maps and visualization grammars using a 5 point Likert scale (1 = Strongly Disagree, 2 = Slightly Disagree, 3 = Neutral, 4 = Slightly Agree, 5 = Strongly Agree). Participants averaged a score of 2.75 when asked "I am familiar with reading choropleth maps." and 1.97 when asked "I am familiar with creating choropleth maps.", indicating that our pool of participants were not particularly confident in designing choropleth maps.

Next, a training session on VegaLite code format and choropleth map design principles and cartography is provided. Finally, each participant was given training on the GeoLinter Framework, walking through the dataset discussed in Case 1 (Section 7.1). At the end of the training session, the participants were given a quiz to verify that they have understood the material. Those that did not score above 80% on the quiz were not considered in the study results. However, all participants who started the study passed. The participants were then asked to use GeoLinter to correct three designs (we refer to these as Tasks 1-3) with the objective: "Design the map that most accurately reflects the underlying data."

In each task, the participants were provided with an initial choropleth design with known flaws. Task 1 contained the flaws **LOW_GVF** and **LEGEND_COLOR**. Task 2 contained the flaws **LOW_GVF** and **BORDER_COLOR**. Task 3 contained the flaw **BG_COLOR**. For all tasks, participants were given a full description of each dataset.

For each of the three tasks, we collected the soft rule correction rate, which was calculated by summing up all of the soft rule violations after the participant was finished with the task. We also collected the task completion time and the change between the presented design and final design with respect to GVF, Moran's I, number of classes, classification method, color palette, and projection method. Results from the tasks are summarized in Table 2. Finally, we conducted an exit survey, consisting of open-ended text responses as well as several repeated questions from the pre-test questionnaire to assess if participants gauged their skills differently after using GeoLinter.

5.3 Results

Results from our design corrections tasks showed a high adoption rate of the GeoLinter suggestions (greater than 95%). In Task 1, 29 out of the 30 participants corrected **LEG-**

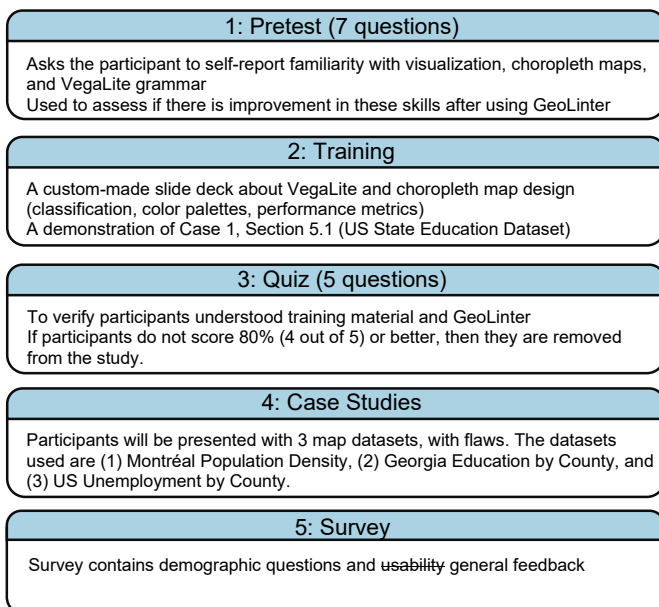


Fig. 3. A summary of our validation study.

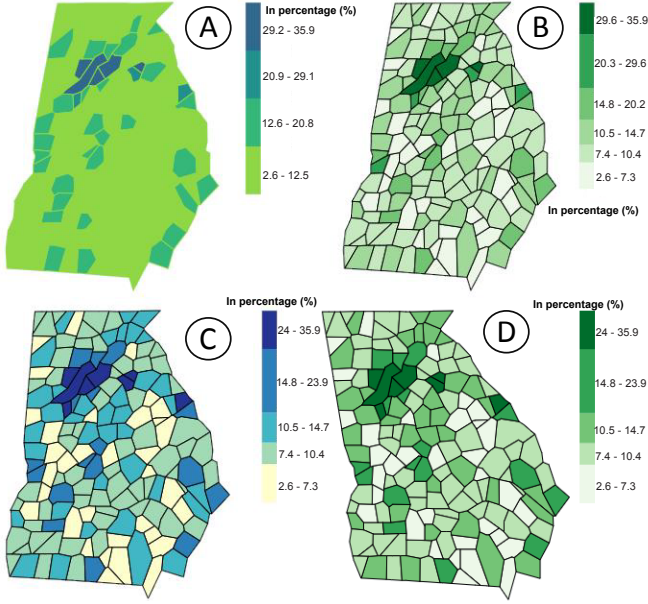


Fig. 4. Exploring various design options to represent the county-level population distribution of the percent of residents with a bachelor's degree in Georgia, USA, 2016 (A) the choropleth map as defined by the original specification. (B) shows the design recommended by GeoLinter, which employs the equal earth projection. (C) shows the design provided by participant 4 during our usability study, which uses a 5-class Fisher-Jenks scheme with a YlGnBu palette under the Natural Earth projection. (D) shows the design provided by participant 17, which uses a 5-class Fisher-Jenks under the Mercator projection.

END_COLOR and all of them corrected **LOW_GVF**. Participants generally favored more classes, with $\mu = 5.83$, $\sigma = 0.91$ for this task. GVF increased by 0.11 and Moran's I decreased by 0.01 on average. When selecting the color palette, 12 participants chose greens, 6 reds, 6 viridis, 6 YlGnBu, 2 OrRd, and 1 RdBu (diverging), indicating that the majority chose a green-related color scheme (the original design our participants were presented with used an unclassified viridis, the default color for VegaLite). The default map projection was Equal Earth, but the largest number of participants changed it to Mercator. None of the participants offered an explanation to this design choice. Most cartographers would agree that changing the projection from Equal Earth to Mercator is an objectively worse choice. That said, the usability study enabled us to identify a key weakness of the current system, the tested instantiation did not have the ability to guide participants toward the best projection for a map depicting a small territory with this shape at this latitude. Of the choices in the system, the best option is the Equal Earth, but that is only because a good choice is not included.

Task 2 uses the Georgia dataset shown in Figure 4. 27 out of the 30 participants corrected **BORDER_COLOR** and all of them corrected **LOW_GVF**. Similar to Task 1, participants generally favored more classes, averaging 5.57 classes. GVF increased by 0.21 and Moran's I decreased by 0.11 on average. When selecting the color palette, 15 participants chose viridis, 12 greens, 1 RdBu (diverging), 1 reds, 1 YlGnBu, indicating that the majority kept the original scheme (the original design our participants were presented with was a viridis scheme). Once again, the default map projection was set at Equal Earth, with 8 participants changing it to

the Mercator projection. Participant 17, who chose to use the Mercator projection said "because that way it looks the closest to the layout of Georgia on a map." We further asked the participant what is Georgia supposed to look like, and the participant mentioned Google Maps, which uses the Mercator projection. Participant 17 lived in Alabama, directly west of Georgia, for 10 years. Participant 18 also offered an explanation as to why the Mercator projection was the best choice, stating that "it distorts the poles ... for a state level map like this, it does not matter." Although partially valid, the explanation fails to capture the relative size distortion caused by the Mercator projection, which makes it generally less preferred by cartographers for choropleth mapping. Figure 4 shows the design originally shown to the participants (A), the design if all of the GeoLinter corrections were adopted (B), and designs chosen from two participants (C-D).

Task 3 is a map of the United States showing the unemployment rate by county. In the initial design presented to participants, the legend was already optimized with a 6-class Jenks-Caspall scheme with a $GVF=0.84$ and Moran's $I=0.68$. The only violation is that the background color is #FC9272, which is the same as one of the fill colors in the legend (**BG_COLOR**). All 30 participants corrected this issue. 28 participants selected the Albers USA projection, and 2 used Albers equal area conic projection.

Task 1 and 2 demonstrate that users tend to blindly follow GeoLinter suggestions, with some injecting their own flawed cartographic knowledge. While this behavior is understandable given the users' unfamiliarity with choropleth map design, the maps they created using GeoLinter's recommendations revealed flaws in the tool's first version. Specifically, the tool's use of compromise projections like Mercator is problematic, as cartographers argue that only equal-area projections are necessary. Additionally, many participants used the equal earth projection, which is not suitable for regional maps like U.S. states. As a result, there is a need to implement a feature for selecting appropriate equal-area projections.

Another issue identified was the lack of titles and units in the legends, which a cartographer expert confirmed as essential elements in all maps. This realization prompted consideration of an additional improvement - implementing checks to ensure that the data is appropriate for choropleth mapping. These suggested improvements are discussed in detail in Section 6.

We also asked each participant *What is your overall impression of this tool?* with a free-form response field. We received positive feedback. 17 out of the 30 participants provided the brief response "good". 8 participants also said that GeoLinter was easy to use, for example "It is very easy to use, even for someone with no previous experience." Another response was "I feel this is a very useful tool to interactively develop choropleth maps. The linting suggestions that continually update as the map is interactively modified makes it very easy for a not very experienced designer to design maps that visually and aesthetically appealing in a nice guided manner." This comment captures the usability of GeoLinter.

We also asked the question *What would you say about this tool's effectiveness as a learning tool?* with a free-form

response field. The response were also positive, for example, one participant said “I could see it can prompt me where I could improve and what was problem with current visualization.” One participant focused on the iterative nature of our pipeline, which was effective at repeatedly correcting errors until there are no more problems: “The interactive nature and continuous updating makes it easy for novices to quickly gain feedback on their design choices when creating such maps. Additionally, the suggestions provided are clear and actionable: the users can understand what is wrong and what corrective steps to take to improve the map.”

In addition to the positive feedback, participants also provided suggestions for improvements to GeoLinter. We summarize these into three categories: metrics, color, and tutorials.

Metrics. In the middle of the study, participant 12 said that deciding the tradeoffs between GVF and Moran’s I was difficult and explained that Moran’s I was hard to grasp. Other participants also voiced their concerns about the lack of in-depth information about our performance measures: “More information about Moran’s I and GVF!” and “Current explanations are good enough, but maybe more that are specific to metrics.” Adding more training or informative tooltips about performance measures can help expand the reach and accessibility of our framework, especially for novices.

Color. Three participants directly stated that there should be more color options. One participant also suggested limiting the options for the outline width: “Make stroke width and color options limited instead of allowing the user set their own.” Incorporating research on color theory and drawing upon more of the research related to the perceptual effects of color (e.g. the further two map symbols are distanced, the more difficult it is to perform comparison tasks [27], larger color distances are beneficial for general choropleth map reading [25], darker glyphs tend to be associated with higher values [30]) on map data could greatly benefit future iterations of GeoLinter. The next extension of GeoLinter will include more help about the implications of different color choices and about which color schemes are essentially equally effective. This will enable the designer to pick the color scheme that best fits into their design goals (e.g., if the map is part of an organization report and that organization has particular hues in their logo and web site).

Tutorials. Six participants mentioned that more information, in general, should be added for each interface element. In the implementation of our interface, the narrative visual prompts and settings disappear after each soft rule is fixed. However, one participant suggested that we should keep the settings even after a rule is fixed: “Improvements to make the UI more intuitive would be useful. Since I mostly just needed to click ‘Fix’, it was confusing when that button disappeared even though I still wanted to make changes.” Another participant preferred if the VegaLite specification panel could be toggled: “The JSON code used to generate the map could be hidden and optionally displayed if user desires to view and edit it.”

6 GEOLINTER V2

Our evaluation study findings (Section 5) and reviews from the initial draft of this paper have been instrumental in identifying areas of improvement for GeoLinter. This led us to transition from our first iteration of the tool to an enhanced second version, which we discuss in detail in this section.

We observed that study participants, despite showing an inclination to adhere to GeoLinter’s suggestions, sometimes opted for the Mercator projection contrary to the tool’s recommendation for an equal-area projection. This trend indicates a possible gap between the tool’s recommendations and user preferences or comprehension, hinting at the need for more comprehensive guidance or education within the tool about optimal projection choices for distinct map types and geographical extents. Furthermore, we identified a need for improvements in areas such as the inclusion of titles and units in the legends. These elements, essential in all maps as confirmed by our cartography expert, were not present in GeoLinter’s initial iteration. We also realized that the importance of appropriate data selection for choropleth mapping had been overlooked. Building upon these insights, we present GeoLinter v2, showcasing enhancements derived from the evaluation study outcomes and a thorough literature review.

6.1 Updated Map Considerations

In Section 3, we presented a review of relevant literature and the identified key considerations in map-making that were directly applicable to GeoLinter. These included guidelines derived from the literature for three key design decisions, those associated with (1) classification method, (2) symbology, and (3) projection. In response to the significant frequency of user errors in the map projection selection, we revisited these guidelines and our implementation strategy for them to ensure more accurate results. Input from reviewers of the initial draft of this paper, a discussion with a cartographer expert on our research team, and a further analysis of Slocum *et al.*’s [18] textbook on thematic map design suggested the inclusion of two additional critical design considerations: (4) title and legend design, and (5) data selection. This paper now presents the enhanced design constraints and goals integrating these important factors.

(C3) Map Projection: Initially, we stated, “For small-scale maps much smaller than the scale of the Earth, conformal projections such as Mercator can be used, as long as the distortion of the area is minimal.” However, following the validation study, we now only exclusively permit equal area projections. Moreover, we utilize Savric *et al.*’s [40] Projection Wizard, which provides more suitable projection recommendations based on the type (e.g., equal-area, conformal, equidistant) and the extent (i.e., the size and shape of the area). For instance, an equal-area projection for regional maps in a square format, such as Wyoming, would employ an Oblique Lambert azimuthal equal-area. For a regional map with an east-west extent, such as Tennessee, an Albers equal-area conic would be utilized. In the case of a region with a north-south extent, like Vermont, a Transverse cylindrical equal-area projection would be adopted.

(C4) Legend Design: As detailed in Slocum et al. [18], legend design can have a significant impact on the effectiveness of a map. The use of either a vertical or horizontal legend will depend on available map space. Numeric values should be placed in a consistent location, either at the bottom or to the right of legend boxes, and boxes should be arranged contiguously. Three methods for specifying class limits are: showing the actual data range in each class, expanding classes to eliminate gaps, or indicating minimum and maximum values and the upper limit for each class. A hyphen can be used to separate numeric values, and including a graphical display with the legend may be useful. The legend and units of the metric being displayed should also be clearly indicated, and should tie in with the title of the map. In GeoLinter, the default legend orientation is vertical, with all of the numerical labels automatically generated, such as in Fig. 2.

(C5) Selecting appropriate data: According to Slocum et al. [18], the choropleth technique is most appropriate for phenomena that are uniformly distributed within each enumeration unit. However, in practice, this is seldom the case, and caution should be exercised in using choropleth maps. One issue to consider when selecting data for choropleth maps is to ensure that raw-total data have been adjusted to account for varying sizes of enumeration units. Four standardization approaches are discussed: 1) dividing an area-based raw total by some other area-based raw total, 2) creating a density measure by dividing a raw total not involving area by either the areas of enumeration units, 3) computing the ratio of two raw totals not involving area, and 4) computing a summary numerical measure (e.g., mean or standard deviation) for each enumeration unit.

Following the revisions to our design considerations, it was necessary to update the corresponding design goals accordingly. The alterations primarily encompass the inclusion and modification of design considerations C3, C4, and C5.

G2: Detect and explain suboptimal design parameters (C1, C2, C3, C4, C5).

G3: Automatically fix the detected errors (C1, C2, C3, C4, C5).

6.2 Updated Framework

Map Projection: The integration of Projection Wizard by Savric et al. [40] helps in two ways: (1) assisting non-expert users in selecting an appropriate projection for their map, ensuring a more accurate representation of the spatial data; (2) offering a fully automatic solution that simplifies the process of choosing a suitable projection, thereby reducing potential errors and misunderstandings. The end result of this is fewer options for the designer, and increased reliability.

Selecting Appropriate Data: To facilitate the process of choosing suitable data for choropleth maps, we integrate data standardization tools into GeoLinter. These tools provide designers with options to standardize their data using the four approaches suggested by Slocum et al. [18]. Through an interactive interface (Figure 2B and E2), designers can input their raw data, select the desired standardization method, and obtain the standardized data with proper

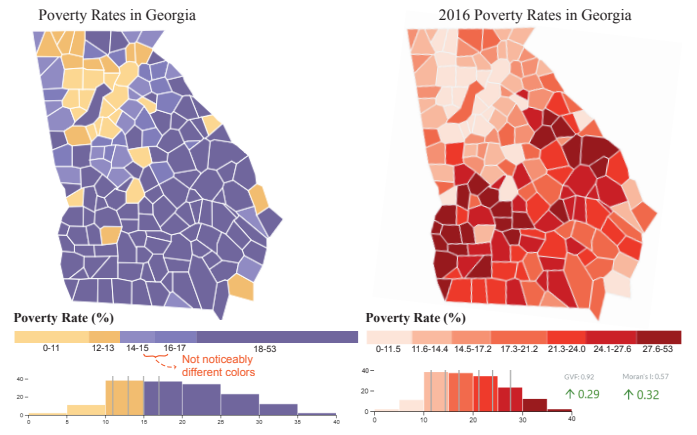


Fig. 5. A published choropleth map showing the poverty rates in percentage in Georgia by county. (Left) The original design where the chosen classification method has a low GVF score and colors that are not easily separable. The bar immediately below it is the legend. The histogram at the bottom suggests that all values above 17 are in one class, and that more than half of all data values fall in this range. (Right) The proposed design based on the GeoLinter suggestions, along with the updated legend and histogram.

units for use in their choropleth map. Maps generated from GeoLinter can only use accepted normalization techniques, further adding to its reliability.

Legend Design: To assist designers in creating effective legends for their choropleth maps, GeoLinter incorporates the following features:

a. Legend orientation: GeoLinter allows designers to choose between vertical and horizontal orientations for their legends, providing a live preview of the changes as they make their selections.

b. Numeric value placement: GeoLinter enables designers to easily define the placement of numeric values in the legend, with options to display them either at the bottom or to the right of the legend boxes.

c. Legend and unit clarity: GeoLinter encourages designers to clearly indicate the legend and units in their map, tying them in with the map's title to ensure a cohesive design. To ensure that the title is descriptive, we employ a template to generate a generic title: [variable] in [region] over [time]. We also allow open-ended titles, such as the ones found in *The Economist*, magazine where authors use a witty joke in lieu of a descriptive title.

7 CASE STUDIES

In this section, we present case studies to demonstrate how our updated framework, GeoLinter v2, supports designers in creating choropleth maps.

7.1 Case 1: Value of Freight Shipments by State 2002

In Figure 2, the designer is creating a choropleth map of US freight shipments in 2002. The Bureau of Transportation Statistics² published this data online, and has published its own version of a choropleth map (as shown in the Original Choropleth Map View in Figure 2B). The first violation

2. bts.gov/archive/publications/freight_in_america/figure_13

is that the data is inappropriate for choropleth mapping (**DATA_NORM**). The map shows absolute, un-normalized values. The designer chooses to normalize the data by dividing the value of freight shipments by the population to yield a derived, per-capita value.

The designer continues to fix design violations as detected by GeoLinter (Figure 2E). The next design error is the choice of colors in the legend. In the original map, the color palette contains two hues that are difficult to distinguish, thus triggering the **LEGEND_COLOR** violation. Also, the GVF score of the original design triggers the **LOW_GVF** violation. For each of these issues, the Detected Violations panel (see Figure 2E) itemizes each soft rule violation along with an explanation on how to address each issue. The first step is to fix the legend fill colors that are too similar to each other, the designer selects a color palette defined by ColorBrewer [22]. The histogram after normalization for this dataset shows that the values range from 0 to 50,000, with no significant skew. In order to address this issue, the designer may choose more unique classes and a color scheme with higher color variance to best show the full extent of the data. In this case, GeoLinter recommends a 7 class Jenks-Caspall classification of the data. This scheme is used in order to capture the nuances of the nonuniform data distribution.

The interface continues to provide guidance on the map projection (Figure 2F.1), which is important in reducing the overall distortion of the map. GeoLinter recommends the Albers Equal Area projection, an equal-area projection that is appropriate given the geographic extent. No major problems with background and border stroke are identified. After following the on-screen prompts (Figure 2 E.2, E.3, E.4), the designer creates a choropleth map using 7 classes with Jenks-Caspall with a YiGnBu color palette on an Albers USA projection. The final design appears in the After Soft Fix View (Figure 2C), and the corresponding code (Figure 2A) is also automatically updated for embedding in websites.

7.2 Case 2: Poverty Rates in Georgia, USA

The next case features a map created by Novoco ³, a national professional services and consulting organization. It published a map of poverty rates in Georgia (see Figure 5 left). At first, glance, this map is problematic, as most of Georgia is colored #6E639B ■, corresponding to the class that captures the data ranging from 17.1%-52.2%. This could potentially destroy meaningful information and insights that could otherwise be seen. When inspecting the histogram view (Figure 5), the 17.1%-52.2% class appears to contain the majority of the data instances, which means the map can benefit from splitting the class into multiple classes.

In this case, GeoLinter recommends a 7-class quantiles classification method with the sequential reds palette (Figure 5 right). We see that the map is able to portray the data with higher granularity; the areas that were previously colored #6E639B ■ for 17.1%-52.2% are now split up into 4 classes. GeoLinter recommends an Oblique Lambert azimuthal equal-area projection. In this case study, GeoLinter may not explicitly flag soft rule violations, but can still guide the developer to an improved map design.

3. novoco.com/atom/143541

7.3 Case 3: Dark UI: GDP by Country in Europe

Due to the popularity of dark UIs (such as in Google Chrome, Windows, etc.), we show how GeoLinter can also produce maps that work with dark UI backgrounds. We use a published map (Figure 6 (A)) of the GDP of Europe ⁴.

There are five rule violations in this case. The first rule violation is **DATA_NORM**, as the map is showing the raw GDP of Europe. The designer chooses to address the rule violation by choosing to divide the GDP by the population of each country, producing a per capita value. The second rule violation, which is correcting an invalid title and legend with proper units (**TITLE_LEGEND**) can be solved immediately after normalizing the data. The designer chooses to title the map “2010 GDP per Capita by Country in Europe”, denoted by the U.S. dollar in the legend.

The third rule violation continues to improve the legend by selecting a valid color palette (**LEGEND_COLOR**). Since this case study features a dark background, the designer chooses the Tropic diverging color scheme from CARTO ⁵ to separate above-average and below-average values. The fourth issue to fix is to the classification performance (**LOW_GVF**). The designer chooses a 7-class Fisher-Jenks to raise the GVF from 0.79 to 0.91, improving the classification accuracy. Finally, the last issue to address is the projection. The most appropriate equal-area projection for a map of Europe is the “Oblique Lambert azimuthal equal-area,” which is automatically selected. Based on the juxtaposition of the map, it is clear that the original map (Figure 6 (A)) had significant areal distortion. GeoLinter updates the VegaLite specification and presents the fixed map as Figure 6 (E) shows.

8 CONCLUSION

In this paper, we present GeoLinter, a framework for automatically detecting flaws in choropleth maps and recommending fixes for both hard and soft rules. The linter compiles a set of design guidelines and metrics from prior research in cartography to aid in selecting design parameters to enable the creation of choropleth maps that most accurately reflect the underlying data conform to well-established cartographic guidelines. Through a validation study, we identified prevalent cartographic errors often made by inexperienced map designers. These findings prompted us to enhance and refine GeoLinter, aiming to minimize the vulnerability to these specific errors. Although GeoLinter currently supports VegaLite, our framework can easily be extended to other visualization grammars. We believe it will have great use as an add-on in productivity and data visualization software. While most software tools rely on preprogrammed logic to determine recommended views for geographical data, sometimes the default settings can result in suboptimal layouts. Embedding our framework in Microsoft Excel and Tableau can help with creating both accurate and well-designed choropleth maps by presenting options and understandable explanations for critical design parameters.

4. github.com/Nicknyr/Europe_Choropleth

5. carto.com/carto-colors

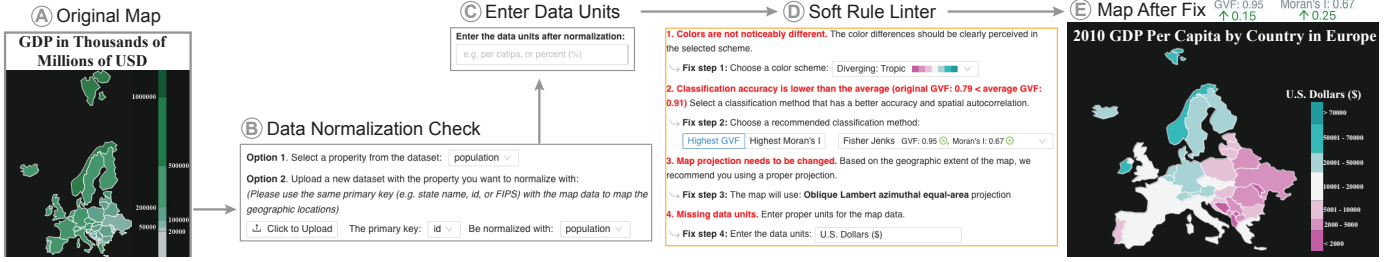


Fig. 6. Case 3, featuring a dark UI map. After setting the original map (A) as the input, GeoLinter provides two options to normalize the data (B) and select the proper data units (C); then the Soft Rule Linter (D) notifies the designer of 3 potential flaws in the original map, specifically 1) not noticeably different colors, 2) low data classification accuracy, and 3) inappropriate map projection. GeoLinter produces (E) to address all errors.

We also believe that the GeoLinter framework could be beneficial for visualization designers who want to learn choropleth map making. A participant from our study directly stated “This tool helped me learn a lot about how to make effective choropleth maps.” Thus, our framework can serve as an educational tool that teaches the basic principles of choropleth mapmaking. Visualization designers can utilize our framework to test out different parameters in their designs.

Limitations and Future Work: We do note that GeoLinter is not without limitations. While GeoLinter embodies a robust system of checks to assist our target audience in avoiding cartographic blunders, one question that goes unanswered is whether a choropleth map the best type of visualization for the task at hand. GeoLinter facilitates users in creating technically accurate maps adhering to established guidelines, however, the tool does not offer assistance in determining the optimal map choice. Thus, a future work or extension could include a function to determine the most fitting map type given the task and dataset.

Another limitation is the framework complexity. Several participants suggested adding tutorials, training, and more information on interface features and performance metrics. Even though some of the terminology and concepts introduced in GeoLinter require extended training, they cannot be eliminated due to its importance in choropleth map design. Future work will focus on developing simpler and more intuitive explanations on the more technical aspects of cartography for future iterations of a map linter, and we will adapt other strategies for building help into visualization tools that target non-experts (e.g. [62]).

Our work and previous visualization linters [1], [2] involve compiling guidelines developed by empirical research into a system. However, this approach may not cater to all audiences at all times. McNutt and Kindlmann [3] argued that linting should consider search or query intent. In Chen et al.’s study on VizLinter [1], several participants rejected some of the rule recommended by their system, influenced by their own preferences or previous experience in visualization, which we also observed in our framework, primarily relating to the choice of projection. A future direction for our work could involve manual specification of intended goals in order to better match audience expectations. We also plan to extend the rule base and functionality of GeoLinter beyond basic choropleth maps. The current version of GeoLinter only contains 4 encoding errors and 6 soft rules, sufficient to address the issues of

most static, univariate choropleth maps. As mentioned in Section 4, each rule is evaluated independently. However, in practice, changing one design parameter affects others as well, and there are multiple solutions for a single violation. Accounting for these interactions should be done in future work in this direction.

Finally, a significant future direction for our work involves conducting comprehensive evaluations of the GeoLinter framework. This would entail submitting the maps produced by GeoLinter to a panel of experts in visualization and cartography for assessment. This approach would not only provide an in-depth understanding of GeoLinter’s performance, but also highlight potential areas for further refinement and enhancement of the system.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grant 1639227 and the U.S. Department of Homeland Security under Award 2017-ST-061-QA0001.

REFERENCES

- [1] Q. Chen, F. Sun, X. Xu, Z. Chen, J. Wang, and N. Cao, “Vizlinter: A linter and fixer framework for data visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 206–216, 2021.
- [2] A. K. Hopkins, M. Correll, and A. Satyanarayan, “Visualint: Sketchy in situ annotations of chart construction errors,” in *Computer Graphics Forum*, vol. 39. Wiley Online Library, 2020, pp. 219–228.
- [3] A. McNutt and G. Kindlmann, “Linting for visualization: Towards a practical automated visualization guidance system,” in *VisGuides: 2nd Workshop on the Creation, Curation, Critique and Conditioning of Principles and Guidelines in Visualization*, 2018.
- [4] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, “Voyager: Exploratory analysis via faceted browsing of visualization recommendations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 649–658, 2015.
- [5] D. Duke, “Modular techniques in information visualization,” in *ACM International Conference Proceeding Series*, vol. 16. Citeseer, 2001, pp. 11–18.
- [6] G. Wills and L. Wilkinson, “Autovis: Automatic visualization,” *Information Visualization*, vol. 9, no. 1, pp. 47–69, 2010.
- [7] L. Battle, P. Duan, Z. Miranda, D. Mukusheva, R. Chang, and M. Stonebraker, “Beagle: Automated extraction and interpretation of visualizations from the web,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–8.
- [8] D. W. Barowy, D. Gochev, and E. D. Berger, “Checkcell: Data debugging for spreadsheets,” *ACM SIGPLAN Notices*, vol. 49, no. 10, pp. 507–523, 2014.
- [9] D. W. Barowy, E. D. Berger, and B. Zorn, “Excelint: Automatically finding spreadsheet formula errors,” *Proceedings of the ACM on Programming Languages*, vol. 2, no. OOPSLA, pp. 1–26, 2018.

- [10] K. Muşlu, Y. Brun, and A. Meliou, "Preventing data errors with continuous testing," in *Proceedings of the 2015 International Symposium on Software Testing and Analysis*, 2015, pp. 373–384.
- [11] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 341–350, 2016.
- [12] D. B. Perry, B. Howe, and C. Aragon, "Vizdeck: Streamlining exploratory visual analytics of scientific data," 2013.
- [13] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer, "Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 438–448, 2018.
- [14] H. Lin, D. Moritz, and J. Heer, "Dziban: Balancing agency & automation in visualization design via anchored recommendations," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–12.
- [15] X. Qian, R. A. Rossi, F. Du, S. Kim, E. Koh, S. Malik, T. Y. Lee, and J. Chan, "ML-based visualization recommendation: Learning to recommend visualizations from data," *arXiv preprint arXiv:2009.12316*, 2020.
- [16] H. Li, Y. Wang, S. Zhang, Y. Song, and H. Qu, "Kg4vis: A knowledge graph-based approach for visualization recommendation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 195–205, 2022.
- [17] B. D. Dent, *Cartography: Thematic map design*. WCB/McGraw-Hill, 1999.
- [18] T. A. Slocum, R. B. McMaster, F. C. Kessler, and H. H. Howard, *Thematic cartography and geovisualization*. CRC Press, 2014.
- [19] R. G. Cromley, "Classed versus unclassified choropleth maps: A question of how many classes," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 32, no. 4, pp. 15–27, 1995.
- [20] I. S. Evans, "The selection of class intervals," *Transactions of the Institute of British Geographers*, pp. 98–124, 1977.
- [21] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological review*, vol. 63, no. 2, p. 81, 1956.
- [22] M. Harrower and C. A. Brewer, "Colorbrewer.org: an online tool for selecting colour schemes for maps," *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003.
- [23] Y. Zhang and R. Maciejewski, "Quantifying the visual impact of classification boundaries in choropleth maps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 371–380, 2016.
- [24] R. M. Smith, "Comparing traditional methods for selecting class intervals on choropleth maps," *The Professional Geographer*, vol. 38, no. 1, pp. 62–67, 1986.
- [25] A. Brychtová and A. Çoltekin, "An empirical user study for measuring the influence of colour distance and font size in map reading using eye tracking," *The Cartographic Journal*, vol. 53, no. 3, pp. 202–212, 2016.
- [26] —, "Discriminating classes of sequential and qualitative colour schemes," *International Journal of Cartography*, vol. 1, no. 1, pp. 62–78, 2015.
- [27] A. Brychtová and A. Çoltekin, "The effect of spatial distance on the discriminability of colors in maps," *Cartography and Geographic Information Science*, vol. 44, no. 3, pp. 229–245, 2017.
- [28] W. S. Cleveland and W. S. Cleveland, "A color-caused optical illusion on a statistical graph," *The American Statistician*, vol. 37, no. 2, pp. 101–105, 1983.
- [29] J. Schiewe, "Empirical studies on the visual perception of spatial patterns in choropleth maps," *KN-Journal of Cartography and Geographic Information*, vol. 69, no. 3, pp. 217–228, 2019.
- [30] M. McGranaghan, "Ordering choropleth map symbols: The effect of background," *The American Cartographer*, vol. 16, no. 4, pp. 279–285, 1989.
- [31] A. of American Geographers, U. C. for Geographic Information Science, M. C. T. Force, and B. of Knowledge Advisory Board, "Geographic information science and technology body of knowledge," Association of American Geographers, 2006.
- [32] I. Golebiowska and A. Coltekin, "Rainbow dash: Intuitiveness, interpretability and memorability of the rainbow color scheme in visualization," *IEEE Transactions on Visualization and Computer Graphics*, 2020.
- [33] P. H. Laskowski, "The traditional and modern look at Tissot's indicatrix," *The American Cartographer*, vol. 16, no. 2, pp. 123–133, 1989.
- [34] L. E. Usery and J. C. Seong, "All equal-area map projections are created equal, but some are more equal than others," *Cartography and Geographic Information Science*, vol. 28, no. 3, pp. 183–194, 2001.
- [35] G. M. Leon, M. Lischka, and A. Breiter, "Mapping the Global South: Equal-Area Projections for Choropleth Maps," *Proceedings - 2020 IEEE Visualization Conference, VIS 2020*, pp. 91–95, 2020.
- [36] F. C. Kessler, S. E. Battersby, M. P. Finn, and K. C. Clarke, "Map projections and the internet," in *Choosing a Map Projection*. Springer, 2017, pp. 117–148.
- [37] B. Šavrič, T. Patterson, and B. Jenny, "The equal earth map projection," *International Journal of Geographical Information Science*, vol. 33, no. 3, pp. 454–465, 2019.
- [38] B. Šavrič, B. Jenny, T. Patterson, D. Petrovič, and L. Hurni, "A polynomial equation for the natural earth projection," *Cartography and Geographic Information Science*, vol. 38, no. 4, pp. 363–372, 2011.
- [39] J. P. Snyder, "Equidistant conic map projections," *Annals of the Association of American Geographers*, vol. 68, no. 3, pp. 373–383, 1978.
- [40] B. Šavrič, B. Jenny, and H. Jenny, "Projection wizard—an online map projection selection tool," *The Cartographic Journal*, vol. 53, no. 2, pp. 177–185, 2016.
- [41] N. Xiao and M. P. Armstrong, "ChoroWare: A software toolkit for choropleth map classification," *Geographical Analysis*, vol. 38, no. 1, pp. 102–121, 2006.
- [42] K.-T. Chang, *Introduction to Geographic Information Systems*. McGraw-Hill Boston, 2008, vol. 4.
- [43] M.-J. Kraak and F. Ormeling, *Cartography: visualization of geospatial data*. CRC Press, 2020.
- [44] C. A. Brewer and L. Pickle, "Evaluation of methods for classifying epidemiological data on choropleth maps in series," *Annals of the Association of American Geographers*, vol. 92, no. 4, pp. 662–681, 2002.
- [45] B. Stern, L. Hurni, S. Wiesmann, and Y. Ysakowski, "Statistics for thematic cartography," *Geographic Information Technology Training Alliance-GITTA*, 2011.
- [46] B. Jiang, "Head/tail breaks: A new classification scheme for data with a heavy-tailed distribution," *The Professional Geographer*, vol. 65, no. 3, pp. 482–494, 2013.
- [47] G. F. Jenks and F. C. Caspall, "Error on choropleth maps: definition, measurement, reduction," *Annals of the Association of American Geographers*, vol. 61, no. 2, pp. 217–244, 1971.
- [48] J. C. Duque, L. Anselin, and S. J. Rey, "The max-p-regions problem," *Journal of Regional Science*, vol. 52, no. 3, pp. 397–419, 2012.
- [49] C. A. Brewer, A. M. MacEachren, L. W. Pickle, and D. Herrmann, "Mapping mortality: Evaluating color schemes for choropleth maps," *Annals of the Association of American Geographers*, vol. 87, no. 3, pp. 411–438, 1997.
- [50] F. Declercq, "Choropleth map accuracy and the number of class intervals," in *Proceedings of the 17th Conference and the 10th General Assembly of the International Cartographic Association*, vol. 1. Institute Cartogràfic de Catalunya Barcelona, 1995, pp. 918–22.
- [51] M. Hegarty, H. S. Smallman, A. T. Stull, and M. S. Canham, "Naïve cartography: How intuitions about display configuration can hurt performance," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 44, no. 3, pp. 171–186, 2009.
- [52] P. A. Moran, "Notes on continuous stochastic phenomena," *Biometrika*, vol. 37, no. 1/2, pp. 17–23, 1950.
- [53] S. J. Rey, L. Anselin, P. Amaral, D. Arribas-Bel, R. X. Cortes, J. D. Gaboardi, W. Kang, E. Knaap, Z. Li, S. Lumnitz et al., "The pysal ecosystem: Philosophy and implementation," *Geographical Analysis*, 2021.
- [54] L. Anselin. (2020) Contiguity-based spatial weights. [Online]. Available: https://geodacenter.github.io/workbook/4a_contig_weights/lab4a.html#spatial-weights---basic-concepts
- [55] L. Anselin, I. Syabri, and Y. Kho, "Geoda: an introduction to spatial data analysis," in *Handbook of applied spatial analysis: Software tools, methods and applications*. Springer, 2009, pp. 73–89.
- [56] ArcGIS Pro 3.1, Esri. (2023) ArcGIS Pro geoprocessing tool reference, spatial autocorrelation (global Moran's I)(spatial statistics). [Online]. Available: <https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-statistics/spatial-autocorrelation.htm#>
- [57] V. Satopaa, J. R. Albrecht, D. E. Irwin, and B. Raghavan, "Finding a 'kneedle' in a haystack: Detecting knee points in system be-

- havior," *2011 31st International Conference on Distributed Computing Systems Workshops*, pp. 166–171, 2011.
- [58] C. Tominski, G. Fuchs, and H. Schumann, "Task-driven color coding," in *2008 12th International Conference Information Visualisation*. IEEE, 2008, pp. 373–380.
 - [59] M. Stone, D. A. Szafir, and V. Setlur, "An engineering model for color difference as a function of size," in *Color and Imaging Conference*, vol. 2014. Society for Imaging Science and Technology, 2014, pp. 253–258.
 - [60] J. P. Snyder, "Map projections used by the us geological survey," US Government Printing Office, Tech. Rep., 1982.
 - [61] M. Bostock, V. Ogievetsky, and J. Heer, "D³ data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.
 - [62] H. Kang, C. Plaisant, and B. Shneiderman, "Helping users get started with visual interfaces: multi-layered interfaces, integrated initial guidance and video demonstrations," in *Proceedings of the 2003 Annual National Conference on Digital Government Research*, 2003, pp. 1–1.

Fan Lei is a Ph.D. student at Arizona State University in the School of Computing and Augmented Intelligence. He received his masters degree in Computer Science from Durham University, UK. His broad research interests include visual analytics, spatio-temporal analysis, information visualization, human-computer interaction, and explainable artificial intelligence.

Arlen Fan is a Ph.D. student at Arizona in the School of Computing and Augmented Intelligence. He received his B.S. in Electrical and Computer Engineering from the University of Rochester in Rochester, New York. His research interests include visualization design focusing on addressing design errors and improving communication.

Alan M. MacEachren is a professor of geography, affiliate professor of information sciences & technology, and director of the GeoVISTA Center (www.GeoVISTA.psu.edu) at Pennsylvania State University. He is PI for the Penn State component of the DHS VACCINE Center of Excellence. His research foci include: geovisual analytics, geovisualization, geocollaboration, spatial cognition, human-centered systems, and user-centered design. He was an associate editor of IEEE Transactions on Visualization and Computer Graphics (TVCG) (2007-2011).

Ross Maciejewski is a professor at Arizona State University in the School of Computing and Augmented Intelligence and Director of the Center for Accelerating Operational Efficiency - a Department of Homeland Security Center of Excellence. His primary research interests are in the areas of geographical visualization and visual analytics focusing on homeland security, public health, dietary analysis, social media, criminal incident reports, and the food-energy-water nexus. He has served on the organizing committees for the IEEE Conference on Visual Analytics Science and Technology and the IEEE/VGTC EuroVis Conference.