

A Design Framework for Visualization of Embedded Time-Series

Katherine Jackman
jackmankatherinem@gmail.com
University of Saskatchewan
Saskatoon, Canada

Fanny Chevalier
fanny@dgp.toronto.edu
University of Toronto
Toronto, Canada

Carl Gutwin
gutwin@cs.usask.ca
University of Saskatchewan
Saskatoon, Canada

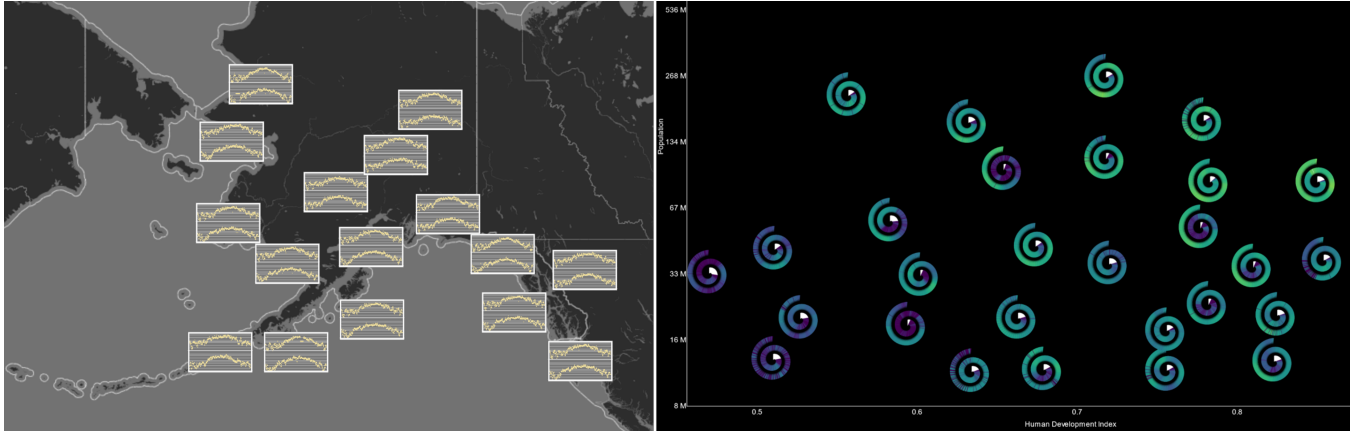


Figure 1: Examples of Embedded Time-Series visualizations. Left: map showing two years of temperature data for 16 locations, using dot plots. Right: scatterplot with spiral colour charts showing two years of Covid-19 cases for each item.

ABSTRACT

Many visualizations use the X and Y dimensions to define a primary spatial layout, but also need to display additional attributes associated with individual data elements — e.g., temperatures over time for different locations on a map. Additional visualizations can be embedded within the primary spatial representation, but little is known about how to design these displays to maintain the goals of both visualizations. We created a design-space framework to set out visual and interaction dimensions for embedded time-series visualizations. The framework helps designers understand different methods for visually embedding a time-series, as well as constraints imposed by the primary spatial representation. We then carried out a crowdsourced study to test the effects of three of the framework’s visual factors: shape (spiral vs. rectangle), size (areas of ~1000, ~2700, ~5200 pixels), and encoding (dot, colour, and dot+colour) on user performance when interpreting embedded time-series in three different primary spatial representations.

CCS CONCEPTS

• Human-centered computing → HCI.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GI '26, June 2026, Waterloo, Canada

© 2026 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/10.1145/1122445.1122456>

KEYWORDS

time-series visualization, embedded visualizations

ACM Reference Format:

Katherine Jackman, Fanny Chevalier, and Carl Gutwin. 2026. A Design Framework for Visualization of Embedded Time-Series. In *The Conference on Graphics Interface*, 18 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

There are many visualization scenarios in which a 2D representation such as a map, a scatterplot, or a network diagram consumes all available spatial coordinates, preventing the use of position for encoding other data dimensions. For example, locations on a map may have associated temperature data over time. When it is important for users to inspect these data as well, designers must find alternative ways to visualize additional temporal and quantitative dimensions. One option is to display temperature data in a separate panel when the user selects a location. However, this requires active user interaction to access the data, partially removes the context provided by the map, limits comparisons and assessments of trends because only a few temperature views can be displayed at once, and reduces the space available for the map itself.

An alternate approach is to embed a visualization of this data *into* the map itself (Figure 1). We call this technique **Embedded Time-Series** (ETS): a method for integrating *time-series data* directly within a visualization substrate that already occupies the full display space (the *base visualization*). By embedding temporal data directly into the base visualization, ETS support investigation of time-series patterns while preserving the surrounding spatial context.

ETS are a subclass of approaches that embed multivariate visualizations into a base visualization, commonly referred to as

nested views [45] or embedded plots [35]). Examples include scatterpies [79], scatterplots within parallel-coordinates plots [80], or adjacency matrices within matrix cells [27]. ETS are embedded plots that focus specifically on time-series data. They are uniquely challenging because they must preserve temporal reasoning—i.e. order (before/after), continuity (trends and cycles), and temporal alignment (synchronous events)—within the spatial and perceptual constraints of the base visualization.

Researchers have explored ways of embedding time-series data within another visualization, using animation (e.g., thematic maps with a slider control [40]), or the third spatial dimension (e.g., colour-coded pillars in a 2.5D perspective view of a map [63, 65]); by adding information to the mark that represents the item (e.g., “growth rings” that show time-series data in concentric circles [8]); or by adding time-series subplots within the base visualization (e.g., Sparklines that show miniature line charts [53]).

Despite these previous explorations, designers lack guidance for embedding time-series data within an existing visualization. We address this gap with an ETS design-space framework that identifies visual constraints imposed by the base visualization, supporting the understanding, comparison, and generation of ETS. The framework is both descriptive and comparative [9] in that it can be used to populate a research agenda for evaluating the effects of different design decisions on perceptibility, usability, and performance.

We evaluated selected dimensions of the ETS framework in a crowdsourced study ($n=398$) to provide designers with initial evidence on how varying ETS designs (shape, encoding, size) affects user performance and perceived effort across several analytical tasks. We found that spiral ETS shapes were slower than rectangles, but did not find differences between the sizes that we tested, and did not see substantial differences between the manipulated encodings.

Our work makes two main contributions: (i) our design-space framework (§3) identifies main visual and interactional dimensions on which ETS representations can vary; (ii) our study of 18 different ETS representations (§4) provides empirical results (§5) that can inform and inspire future explorations of our framework (§6).

2 RELATED WORK

2.1 Visualizing Time-Series Data

In a time-series, one or more variables are recorded at multiple points in time, often recorded as an ordered sequence of time-value pairs. Time-series were one of the earliest data types to be visualized (e.g., Playfair’s 1786 chart of England’s trades [77]). Since then, researchers have explored many methods for visualizing time-series data, summarized below. A comprehensive review can be found in surveys by Aigner et al. [2] and Bach et al. [4, 5].

Timeline-based representations are structured around a main time axis, the “timeline”, with events represented as marks and intervals as lines. In use since the 1700s [76], timelines are still a dominant representation for information such as daily routines [15], medical records [55], or project plans (e.g., Gantt charts). Scalar data has also been mapped along a time axis, for instance using traditional dot plots or line charts, and in *streamgraphs* such as ThemeRiver [41], which are akin to stacked area charts showing magnitude values for several thematic data sources over time. Space-optimization techniques have been proposed as well: *horizon graphs*

divide a standard line-and-area chart into vertical bands that are overlaid on one another with different colour coding [30, 43]. Sun and colleagues extend the idea of the time axis by embedding temporal data onto routes in a map (called “route-zooming”) [62].

Other approaches have been designed to represent periodic data. *Calendar-based* representations cluster temporal data into days, weeks, or months [70], and can be reduced in size to become grid-style tile maps, usually with colour encoding the value [2, 71]. Other representations have been explored to support comparison across similar time periods: stacked rectangles, concentric circles [39, 83], spiral charts [66, 73, 82], and 3D helices [65] all lay out similar time periods at the same part of the geometry.

The constraint of a rigid structure of linear or periodic time can be relaxed when time is important to represent in the context of other data dimensions. For instance, Time Curves [7], connected scatterplots [38], and TimeSplines [54] bend the line that represents time such that similar elements can be placed close together in 2D space, are organized in a scatterplot, or follow a freeform user-defined layout. In these approaches, the oscillation, intersection, and regularity patterns of the bent timeline can reveal complex time- and feature-based relationships between the data items.

All of the techniques above visually represent time in space, but it is also possible to use actual time to move through the time-series dataset using animation (e.g. [13]). This approach may be required when the data to be displayed is large, and can be effective for many tasks [59]; however, seeing only one data value at a time removes the ability to consider and compare values at different time points.

2.2 Visualizations within Visualizations

Embedding visualizations as subplots is a popular method to reveal complex relationships in multivariate data, in a space-efficient manner [25, 35, 45]. Early uses of this approach (Figure 2) include Minard’s maps that used pie charts to show quantities of goods sent to different cities [58, p. 85], and Bertin’s ‘cartogrammes’ which embed diagrams [11, p.118-119] or data-rich glyphs [11, p.394] within geographical maps. Cleveland and Terpenning’s seasonal subseries [21] is another historical example where barplots representing seasonal production patterns are placed within a scatterplot.

The general idea of embedded plots [35] (or nested visualizations [45, 48, 84]) is to upgrade the visual marks that are placed in a map or chart to become self-contained visualizations – that is, subplots with their own axis system. This integration of subplots as marks in a base visualization shows local patterns as visualized by the subplots at a micro level (e.g., deviations from the mean production for a given month in Figure 2-right), while maintaining their global context within the axis system of the base visualization at a macro level (e.g., mean production variations over the year). Visualization idioms building on this approach include scatterplot matrices, which show a scatterplot for each pair in a set of variables [10, 28] and tile maps (also called grid maps), which partition a map into tiles that contain plots [61].

Others have explored the encapsulation of various types of visualizations within a base representation. Embedding traditional plots such as dot plots, bar charts, line charts, and pie charts is now standard practice in cartography and GIS systems [20, 21, 29, p. 164], and also in tabular systems [57]. Multi-variate glyphs can be

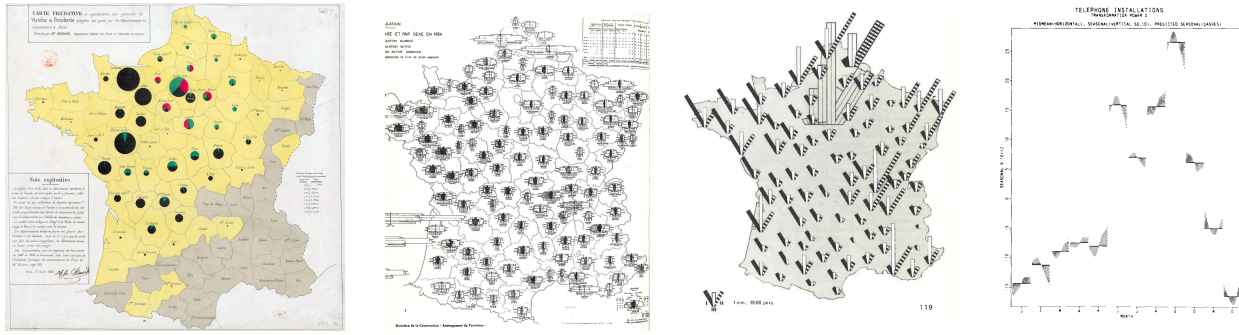


Figure 2: Examples of historical embedded plots: (a) pie charts [58, p. 85], (b) age pyramids [11, p.394], (c) multivariate glyphs [11, p.119] embedded within a map base; (d) subseries barplots embedded within a scatterplot [21].

embedded into a base visualization: e.g., star plots within scatterplots [3] or maps [74], representations of data as petals in a flower placed on the bars of a bar chart [52], or Chernoff faces that represent different data elements as facial features [69]. Other creative embeddings can extend beyond points to lines or areas: for example, representing data on a map’s contour lines [81], on curves [6], or in the lines of line charts [14, p.138].

For time-series data more specifically, embedded subplots include miniature line charts intended to be integrated into text, tables, images, or spreadsheets. To address perceptual issues caused by miniaturization, *Sparklines* simplify the line chart representation by removing axes and legends, resulting in simpler trend lines [64, 67]. A variety of base visualizations have been explored for embedding timeline-based visualizations. For instance, area charts embedded in rectangular blocks of a treemap, to represent stock-market changes [75]: here, perceptual challenges arise, as the rectangular blocks vary in size, making the subplots difficult to compare due to varying aspect ratios. Gruendl et al. proposed an approach that utilizes perspective to represent time-series on a third dimension of parallel-coordinate plots [36]. Also leveraging a third spatial dimension, Tominski et al. showed case-rate information for locations on a map as either stacked towers or helices [65]. In general, any existing time-series visualization can be miniaturized and embedded into other visualizations; Aigner et al.[1] show examples of thumbnail-sized ThemeRiver and TimeWheel charts within a map’s political divisions, and note the tradeoffs that occur in embedded representations (i.e., reduced size of the time-series representations, potential occlusion of the map, finding a reasonable layout, and offloading information such as legends to interaction).

A main concern with embedded visualizations is that there is less space to show the data. The perceptual and analytical limitations of compact visualizations have been considered in several studies: for example, Heer et al. looked at interpretation accuracy with horizon charts of different sizes [43]; Gramazio et al. studied the effects of size on visual search time [34]; and Demiralp et al. investigated “perceptual kernels” that identify perceptual limits for colour, shape, and size [23]. Several researchers have also considered “at-a-glance” interpretation of small visual representations (e.g., [53], [12]). Further, aggregation and placement techniques have been developed to reduce visual complexity and deal with occlusion problems [37, 46, 51, 60].

3 ETS VISUAL DESIGN FRAMEWORK

The ETS framework can guide designers who need to represent time-series data within another existing visualization. The framework’s goal is not to provide instruction on the basics of visualizing time-series data – these foundations are well covered by existing research and surveys (e.g., [1, 4]) – but rather to outline the issues that must be considered when a time-series is visually embedded in another visualization. The framework focuses on the visual design of an ETS, considering issues such as how much space is available, as well as the basic geometry and data-to-visual mapping. These design issues can be constrained by properties of the base visualization, but the base visualization may also provide opportunities that might not exist in a non-embedded context.

3.1 ETS Geometry and Shape

Designers must first decide how a time-series will be represented within the base visualization, including overall geometry, shape, and the mapping of ETS space to the time-series data.

3.1.1 Overall Geometry. There are three main representational forms for ETS visualizations. First, the time-series can be shown as a *2D subplot*. The subplot can use any of the representational techniques for showing time-series data (see overview in section 2): for example, if the data is scalar, the ETS could use a miniature dotplot, radial line chart, ThemeRiver diagram, or horizon chart; if the data involves state snapshots or images, the subplot could be shown as a miniature small-multiples diagram. The use of subplots means that viewers can read the ETS visualizations at different levels: at a macro level, viewers can contextualize the embedded miniatures as marks within the base visualization, while at the micro level, they can inspect the details of a particular ETS representation or compare different ETSs [35].

The choice of which representation to use for 2D subplots is discussed further below, but in general, there are constraints imposed by the amount of space available for the ETS (since different subplot types have different thresholds for visibility) and the visual characteristics of the base visualization. In addition, the amount of data in the time-series may affect whether 2D subplots can be used: since numeric or event-based elements can be represented compactly (e.g., as a single dot), more data can be included in a miniature than for larger elements such as images.

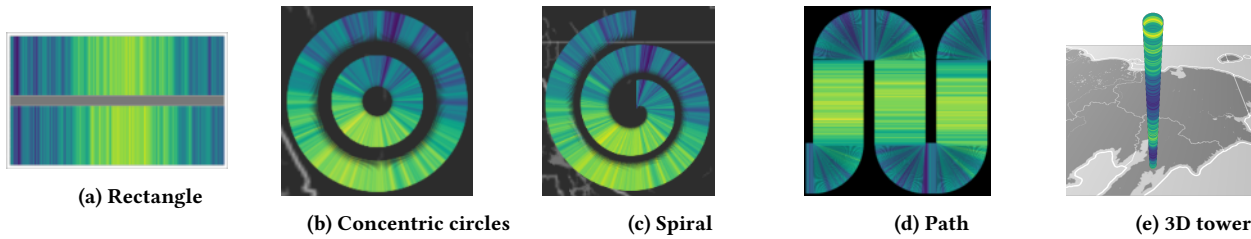


Figure 3: Example ETS geometries and shapes. A-D show 2D subplots; E shows the time-series on a 3D structure.

Second, the time-series can be shown using *pseudo-3D structure*, such as representations that extend upwards as towers on a map, with rings of the tower showing different time values using colour or size (e.g., [63, 65], and Figure 3e). This geometry requires that the base visualization can be viewed in perspective, and may require that the user manipulates the perspective viewpoint to deal with occlusion in the base visualization. Perspective may mean that different parts of the tower are shown at different resolutions (e.g., the bottom of a tower is compressed relative to the top), but control over the viewpoint also provides opportunities to show the time-series in different ways – for example, if the user moves the viewpoint to look directly down on the base visualization, the 3D towers show only one value and occlude very little of the base; if the perspective is from the side, then the time-series towers are given more space and can be shown in higher resolution. 3D structures have similar limitations to 2D subplots in terms of dataset size, but the constraint for 3D involves height rather than area.

Third, the time-series can be shown with *animation*. First, each item that has a time-series can display a single element of the series; control over which item is shown is based on an animation timer or a user control like a slider. This approach reduces the ETS representation to the space needed for a single element, and so can be compact in size – and when individual data elements are large (e.g., images in a time-lapse series), this may be the only reasonable approach. Second, temporal data can be shown as a flow animation, often rendered on top of existing structure in the base visualization (e.g., migration patterns between countries, rendered on connection lines between map locations).

3.1.2 Shape of the ETS. The shape of the ETS (in conjunction with the data mapping) determines how the time-series will appear in the base visualization. Shapes may be limited by the geometry chosen for the ETS, and there are also constraints and opportunities presented by the base visualization; in particular, designers may wish to take advantage of geometry already present in the base.

For 2D subplots, the main shapes seen in previous work are rectangles (miniature dot plots, line charts, timelines, calendars, or small-multiples displays, see Figure 3a), circles (Figure 3b), spirals (Figure 3c), and polygons (radial plots, concentric rings, TimeWheel charts, or spiral charts, or stacked-area shapes (e.g., miniature ThemeRiver charts). If 3D structures are used, there are fewer examples of different shapes – previous work has used stacked towers, helices (i.e., spirals pulled upwards into the third dimension), or multivariate glyphs ‘extruded’ into 3D to show a time axis [31] – although other 2D plot types could be adapted to a 3D presentation (see the rich design space of space-time cube visualizations [7, 32]).

These 2D and 3D options provide a candidate set of shapes (Figure 3). However, the visual elements already used in the base visualization can influence the choice of ETS shape. For example, if data items in a scatterplot use circular marks, this may restrict ETS to using circles or spirals that fit into the existing items – but this also provides an opportunity to integrate ETS into the base visualization without unduly changing the overall design.

Reusing existing geometry depends on how items are already shown: nodes in a graph may be represented as circles, items in a scatterplot by size-encoded marks, locations on a map by stylized map pins, items in a treemap by coloured regions, and values in a bar chart by the bars themselves. These existing shapes can potentially be reused to show time-series data: e.g., Figure 4 shows the series as a coloured border on a graph node (Figure 4a), a spiral chart inside a map pin (Figure 4b), and a colour sequence set into the bars of a barchart (Figure 4c).

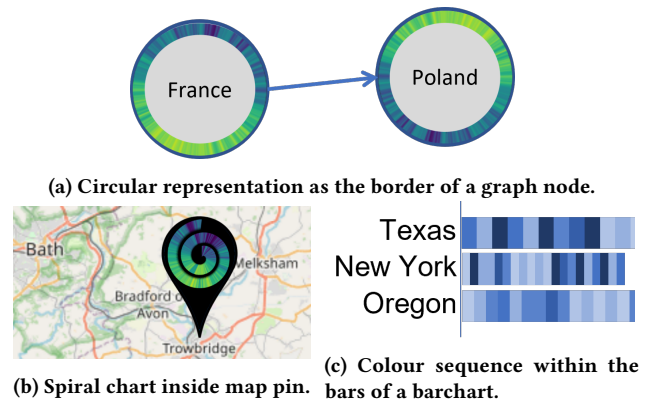


Figure 4: Example ETS that reuse existing base geometry.

Reusing geometry does have limitations. Although the shapes allocated to items in the base visualization may have a fixed size (e.g., map pins), some representations may vary in size (e.g., the bars in a bar chart or scatterplot marks that encode a variable using size). If different ETS have different amounts of space, comparisons can become more difficult, and if the geometry is small, the ETS may be difficult to interpret. If existing geometry is not used, then the ETS designer can decide on the shape, but must consider the effects of adding new visual objects to the base representation. Visual interference may occur if ETSs use a shape that is already in use; in addition, any shapes that are added to the base visualization can cause occlusion (discussed further in subsection 3.2).

Finally, the desire to maintain a harmonious aesthetic presentation in a base visualization may constrain how ETS are represented

– for example, a smooth and rounded design style in the base visualization may preclude using angular shapes for the ETS.

3.1.3 Layout of the time-series. The way in which the time-series is laid out within the space of the ETS is largely governed by the type of time-series visualization being used – for example, dot plots and line charts use a left-to-right or top-to-bottom layout; spiral and circular charts use radial layout; and path-based visualizations lay out the time-series data along a curve or line. However, additional layout issues arise when time-series are embedded. First, designers must recognize that embedded representations will generally be much smaller than a stand-alone chart, which may reduce interpretability. Second, designers can consider whether the ETS should use either a similar or a contrasting layout pattern to that of the base visualization: for example, if the base visualization is read left to right, then it may be advantageous to preserve mapping consistency; but in some cases a contrasting layout may help to visually separate the ETS representations from the base, as long as the visual presentation of the ETS clearly indicates the layout.

3.2 ETS Size

The size of an embedded visualization determines the amount of data that can be represented and the detail of the representation. Increasing size generally increases visibility and interpretation accuracy, but for ETSs, size may be strongly constrained by other elements. A primary limitation on ETS size is the density of information in the base visualization. The base may contain both discrete visual marks as well as continuous background information (e.g., a satellite-image background for a road map). The density of existing discrete elements limits the space available for ETS – whether the designer is adding new visual elements for the ETS, or increasing the size of existing elements to fit ETS inside. Second, if there is continuous background information, its importance for tasks in the base visualization (e.g., to provide context) limits whether ETS can occlude the background. It is also possible that a visualization’s background provides aesthetic value, further limiting ETS size.

For some base visualizations, such as a map with an information-dense and important background, the constraints on ETS size can be severe, and designers may only be able to reuse existing geometry or providing very small ETS representations. Other visualization types have more available space. For example, the ETS space in a scatterplot is limited only by the density of elements (since there is no “background” data). Designers can also adjust the base visualization to make more room for the ETS (e.g., by changing the zoom level or aggregating and filtering some elements).

In other visualizations, such as graphs and trees, the X and Y dimensions are used for layout rather than as continuous data axes, and so the space needed for ETS could be factored in to a graph-layout algorithm. The idea of adapting the base visualization’s layout to accommodate ETS and avoid occlusion can also be applied to views with continuous frames of reference, using distortion-oriented techniques such as fisheye lenses [18].

Restrictions on available space mean that designers need to know how to show time-series data in small spaces. As described earlier, researchers have studied the visibility and interpretation of small visualizations, and designs such as horizon charts [43], Sparklines [53], and “glanceable” representations [12] have been shown to be

effective even at small sizes. Designers may also be able to reduce their ETS space requirements, depending on the needs of the task, by pre-processing to reduce the amount of time-series data in the representation – for example, by aggregating data into larger bins (e.g., months instead of days), or by showing only recent values rather than the full dataset [47].

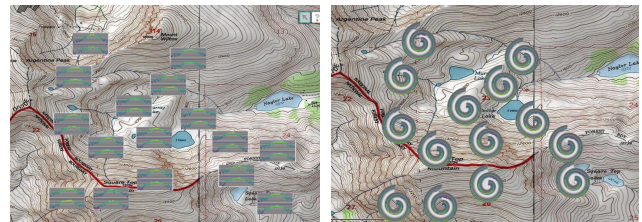
A related issue is that changes to the scale of the base visualization can dramatically affect the space available for ETS representations. If the base allows zooming, then ETS size may need to be dynamically adjusted – and although designers could simply scale the ETS, it may be more appropriate to use a semantic-zoom approach that adapts visual design to the current size.

3.3 ETS Data Encoding

The third visual design decision for embedded time-series is the way in which values are mapped to visual variables. Common encodings for time-series include distance-based coding in which data values are represented by the distance from an axis (e.g., dot plots or line/area charts) and colour encodings in which the data value is mapped to a colour scale (e.g., a false-colour scale such as Viridis [50]). Other visual variables such as size or orientation can also be used, but are less common.

The designer’s choice of encoding, however, must consider the potentially small size of the ETS, and the possible interactions between ETS and base-visualization encodings. Interference between encodings is a primary concern because the visual design of the base visualization may reduce the ability to read an ETS’s encoding. For example, if a map uses false colour to show elevation, then using a similar colour scale for ETS may both reduce visibility and interfere with the base’s visual style. Similarly, if the base visualization has elements represented as lines (e.g., contour lines in a topographic map, as shown in Figure 5a), using a horizontal ETS presentation ETS may become more difficult to read; changing the geometry of the ETS can solve the problem (e.g., Figure 5b).

The size of an ETS can also affect encoding. Because there is limited space for traditional annotations such as axis scales, position-based mappings will not have a reference scale next to the data; adding visual references such as guidelines can provide valuable context. Similarly, small size also implies fewer pixels per data value, making tasks based on visual perception (such as colour differentiation) more difficult. As a result, category colours or colour mappings may need to be further separated in perceptual space.



(a) Interference from base and (b) Using a spiral ETS can reduce ETS using horizontal lines. line interference.

Figure 5: Interference effects between base and ETS.

3.4 ETS Interaction Design

There are several kinds of user interaction that can be added to an ETS. A primary requirement is to allow the user to inspect the ETS and see the data in more detail: for example, hovering over the ETS or clicking on it could provide a larger preview or show the time-series data in a separate larger panel. However, interaction techniques for the ETS must not conflict with those used for the mark itself, or for the base visualization. For example, if clicking on an object in a scatterplot gives detail about the object, then clicking may not be available for interacting with the ETS.

Secondary interaction requirements include the ability to show and hide the ETS representations (to allow the user to focus on the base visualization), local occlusion control (e.g., a user could drag an ETS to see occluded base data), and temporary appearance changes (e.g., temporarily desaturating ETS to reduce colour clutter).

Other techniques can also allow users to work with the time-series data specifically. For example, users could create new ETS representations that are summaries of multiple time series, or could enhance comparisons between ETS (e.g., selecting one ETS could change others to show their values relative to the selected ETS rather than their raw data (Figure 6).

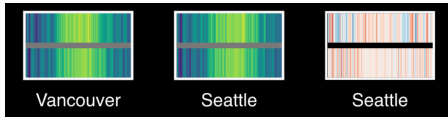


Figure 6: Showing relative differences to a selected time-series. Left: Temperature data for Vancouver; Centre: Original data for Seattle; Right: relative differences for Seattle, as a red-blue warmer/cooler colour chart.

4 EMPIRICAL STUDY OF ETS PERFORMANCE – STUDY METHODS

To provide initial evidence about user performance with a set of ETS visualizations defined by the framework, we carried out a study that asked participants to answer multiple-choice questions about time-series embedded in a base visualization. In the study, each participant worked with one ETS representation across three base visualization / dataset pairs.

4.1 Experimental Factors and Conditions

We compared specific points along three visual dimensions identified by the framework – the shape, size, and data encoding of the ETS (each participant saw one combination of the dimensions).

ETS Shape. We compared two subplot shapes: a rectangle that shows data linearly from left to right akin to a barcode plot, which was chosen because it is a standard for visualizing time-series; and a spiral representation that shows data along the path of the spiral, which was chosen because it has a basic similarity to Rectangle, but with a geometry that allows it to fit in a circle.

ETS Size. ETS size is often limited, so it is important to determine performance at different sizes. We compared three subplot sizes for the study: small, medium, and large (areas of ~1000, ~2600, and ~5100 pixels, determined through informal pilot studies conducted

with members of our research group in order to find sizes that were recognizably different and with the small size still interpretable).

ETS Data Encoding. We compared three encodings: *dot encoding*, where data values are mapped to height within the Rectangle region or to the radial distance on the Spiral path, and a light-orange circle is drawn to represent the value; *colour encoding*, where data values are mapped to a perceptually-uniform colour scale (Viridis), and the area allocated for each data point is coloured according to the mapping; and *dot+colour encoding*, where data values were double-encoded by combining the dot and colour encodings above. All 18 versions of the ETS used in the study are shown in Figure 7.

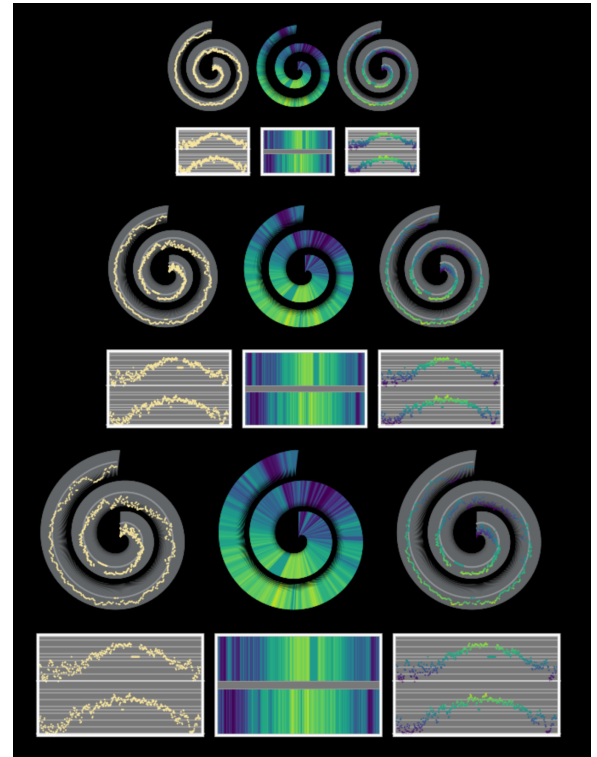


Figure 7: The 18 different ETS representations used in the study, showing the temperature dataset. Encodings from left to right: dot, colour, dot+colour

4.2 Tasks: Base Visualizations, Datasets, and Question Types

To provide a broader look at the performance of the different ETS representations, we tested them with three different base visualizations paired with three different time-series datasets:

- Map + temperature. Daily average temperature data was gathered for 17 locations, and an ETS for each was placed on a geographic map (using the *leaflet* and *react-leaflet* libraries).
- Scatterplot + Covid-19 cases. ETS visualizations showing Covid daily case numbers for 34 countries were placed on a 2D scatterplot (population on the Y axis, Human Development Index score on the X axis).

- **Graph + migration data.** A custom graph was created with countries as nodes and connections between countries as links, to show migration data over time. ETS representations were placed on each link, midway between the start and end nodes.

The three base visualizations and datasets are shown in Figure 8.

In each base visualization, we asked eight questions that covered five types of visual analysis. The ETS representations needed for the question were highlighted in the visualization with coloured circles (see Figure 8), so that participants could focus on the ETS and not have to carry out a lengthy visual search. The question types were adapted from Quadri and Rosen’s taxonomy [56]:

- *Retrieve a value* from the representation or compute a derived value (2 questions, e.g., “What was the approximate average Covid case rate for Uzbekistan in April 2020?”)
- *Compare values* either within one ETS or across multiple ETS (3 questions, e.g., “Which country – Philippines, Yemen, or Nepal – had more cases in June-August of 2021?” or “For Netherlands, in what year did the months August-September have more Covid cases: 2020 or 2021?”)
- *Find an extreme value* within the data (1 question, e.g., “What was the approximate minimum case rate in 2021 for Azerbaijan?”)
- *Determine the range of values* for a time period (1 question, e.g., “What range of case rates were seen in Cambodia in 2021?”)
- *Order items based on values* from the ETS (1 question, e.g., “Rank countries Syria, India, and Vietnam in terms of average case rate, from lowest to highest, for July 2021”).

An example task is shown in Figure 9 (taken from the study screen shown in Figure 8b). This task asks the participant to determine, for the circled ETS, in which of two years were the months January-March colder. To answer the question, the user must find the indicated months for the two bands of the spiral, and compare their colour coding using the provided legend. As shown in the figure, the colours are darker blue (indicating colder temperatures) for the year 2021.

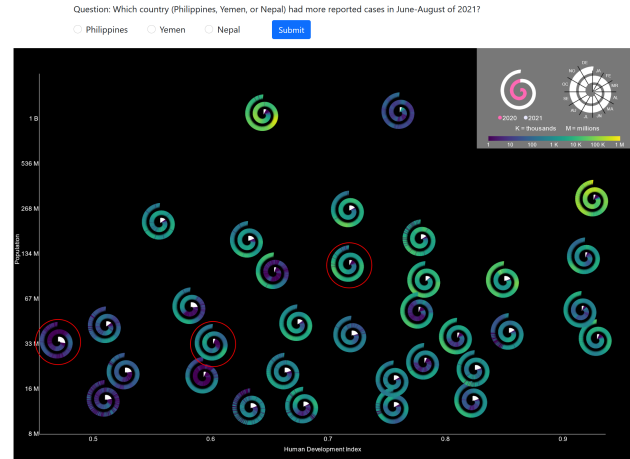
4.3 Study System

A custom web-based application was developed using HTML and the P5 canvas library (www.p5js.org) to display the visualizations, instructions, and questions (Figure 8). All ETS visualizations were generated programmatically, and were drawn over the base visualization. The only interaction with the visualization was hovering over an ETS, to expand its size by 2X. Study data and questionnaires were recorded using the experimental software.

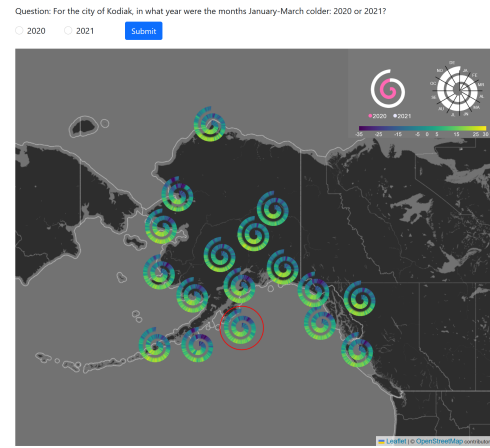
4.4 Procedure

The experiment followed five phases: informed consent, a demographics questionnaire, a colour-vision test, three rounds of study tasks, an effort questionnaire (NASA TLX), and a final debrief. The colour-vision test asked participants to complete an Ishihara test and questionnaire to screen for colour vision deficiencies [44]. Each round of study tasks had five parts:

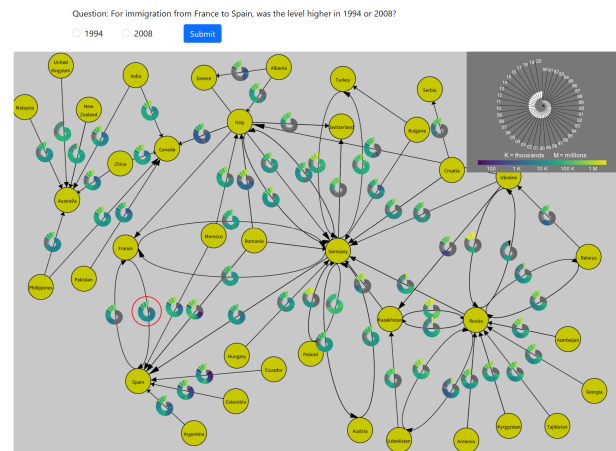
- A page introducing the base visualization and dataset (map / temperature, scatterplot / Covid, or graph / migration), the hover interaction (to enlarge the ETS), and the ETS representation (one combination of the factors described above).



(a) Scatterplot base and Covid dataset.



(b) Map base and temperature dataset.



(c) Graph base and migration dataset.

Figure 8: Study interface, showing base visualizations / datasets and actual questions (larger versions in Appendix).

- A page providing instructions for the tasks: participants were told that they would answer a series of multiple-choice questions

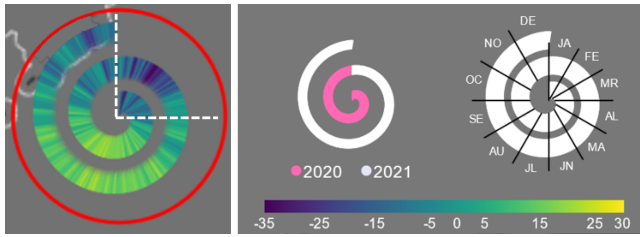


Figure 9: Example task: “in what year were the months January-March colder, 2020 or 2021?”. To answer the question using the ETS shown at left, the user must find the correct regions of the spiral for the indicated months (here annotated with dashed lines) and compare their colour coding using the provided legend (shown at right).

based on the information presented in a visualization, that each question would be presented on a separate page, and that they would be able to read the question before the visualization appeared. Participants were told to answer the questions as quickly and accurately as they could, and that they would need to keep working until the question was answered correctly.

- Three practice tasks using the base visualization with a small number of items; these tasks allowed participants to become familiar with the way the item highlighting and the multiple-choice questions worked.
- A final practice task using the full set of items, for participants to become familiar with the appearance and density of the dataset.
- Eight multiple-choice questions that required analysis of specific ETS visualizations. Question types were presented in the same order for all three base visualizations/datasets.

Each question was presented on a new page (including both practice and test questions). When the participant had finished reading the question, they pressed a ‘Start’ button to show the visualization and the multiple-choice answers (and start the internal timer). To answer a question, participants selected a radio-button item and pressed a ‘Submit’ button. If the answer was incorrect, a popup dialog told the participant to answer again; when the answer was correct, the system moved to the next question.

4.5 Participants and Recruitment

We recruited 398 participants (mean age = 37.7, s.d. = 11.2, 201 men, 191 women, 6 not stated) using the Amazon’s Mechanical Turk (MTurk) crowdsource platform, and gathered data with a custom browser-based experiment tool. Data collected from MTurk has been used previously in several visualization and HCI studies; however, previous research has shown that data quality can vary on crowdsourcing platforms [16, 42]. Therefore, in addition to the exclusions described below, we also restricted participants to be within the United States and to have an acceptance rate (a measure of a worker’s past quality) over 90%.

All participants were paid \$6 for completing the study, which took approximately 30 minutes. Self-reported estimates of monthly visualization usage among participants averaged 30 hours with bar charts, pie charts, and line graphs as the most commonly used.

4.6 Study Design

Our goal was to understand the effects of some of the framework’s main dimensions on user performance with ETS visualizations. The study used a mixed factorial design with five factors:

- **SHAPE** (between-participants): *Spiral* or *Rectangle*
- **SIZE** (between-participants): *Small* (~1000 pixels), *Medium* (~2700 pixels), or *Large* (~5200 pixels)
- **ENCODING** (between-participants): *Dot*, *Colour*, or *Dot+Colour*.
- **BASE VISUALIZATION** and dataset (within-participants): *Map* (temperature), *Scatterplot* (Covid), and *Graph* (migration). The order of base/dataset was counterbalanced using a Latin square.
- **QTYPE**, short for question type (within-participants): *Retrieve*, *Compare*, *Sort*, *Find Extremum*, *Determine Range*, *Compute Derived Value*, and *Sort*.

Our primary dependent variables were **COMPLETIONTIME** (time to correct answer), **HOVERCOUNT** (only included for highlighted items and hovers longer than 300ms), **ERRORS** (number of incorrect answers), and subjective effort ratings (TLX questionnaire).

For each dependent variable, our analysis considered main effects and interactions for the three main factors (SHAPE, SIZE, and ENCODING), and considered interactions between the main factors and the two within-participants factors (BASE and Q-TYPE).

Although 516 participants consented to the study, we excluded 118 participants based on the following criteria: not completing all trials or survey questions (108 people), having a colour-vision deficiency that would limit ability to see the visualizations correctly (6 people), having an overall completion time outside 3 SD from the mean (0 people), or having a mean completion time of less than 4 seconds with zero errors (4 people). We continued to recruit until there were at least 20 participants in each of the 18 conditions (SHAPE × SIZE × ENCODING); after exclusion, there were 398 participants in total.

In summary, each participant worked with one of the 18 ETS representations across all three base visualization / dataset combinations, and completed four practice tasks and eight test tasks with each base visualization / dataset. Each participant therefore carried out $3 \times (4 + 8) = 36$ tasks during the study.

5 RESULTS

We report the results of factorial tests and also report confidence intervals for each of our main analyses as plots [22]. ANOVA and RM-ANOVA are used for factorial analysis, and we report effect sizes using generalized eta-squared η^2 (considering .01 small, .06 medium, and >.14 large [49]). For any follow-up tests involving multiple comparisons, the Holm correction was used. The 95% confidence intervals (CIs) were computed using bootstrapping [26]—for each chart in the following sections, the overall metric was calculated for each condition for each participant and the mean and 95% confidence intervals were computed using bootstrapping. Analyses are organized below by dependent variable. We provide further details on data processing and detailed tables in Appendix A.

5.1 Completion Time

Tasks were timed from when the participant pressed the ‘Start’ button after reading the question, to when they selected the correct

answer and pressed the 'Submit' button. The time for any erroneous answers is included in the measure.

Figure 10 shows a summary of mean completion times per trial, for each independent variable (see also Table 1 in Appendix A). Factor SHAPE had a significant effect on COMPLETIONTIME, with questions involving *Spiral* taking 16.8 seconds CI [15.4, 18.6] on average versus 13.4 seconds CI [12.2, 14.8] for *Rectangle*.

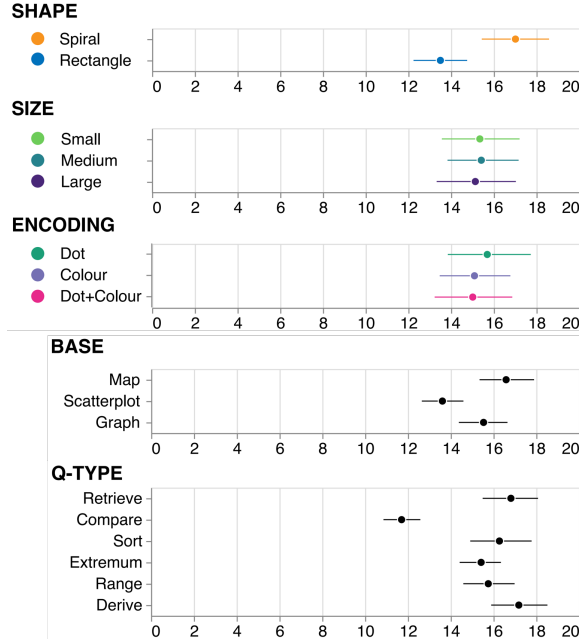


Figure 10: Mean COMPLETIONTIME (in seconds) per trial, for our five independent variables. Error bars are 95% CIs.

There was also a significant interaction between SHAPE and BASE: Although *Spiral* was slower for all base visualizations, the amount of difference between the two shapes varied. The largest difference was in the *Map* tasks, where *Spiral* was on average 5.1 seconds CI [2.7, 7.7] slower than *Rectangle*, versus 2.8 seconds CI [0.8, 4.6] for *Scatterplot* and 2.4 seconds CI [0.1, 4.6] for *Graph*. See details in Figure 19, right in Appendix A.

Our analysis showed no main effects of SIZE or ENCODING on completion time, but the ANOVA did show an interaction between SIZE and BASE. Our interpretation of this interaction is that there is weak evidence that *Large* was faster for the *Graph* tasks (-2.38 sec CI [-5.15, 0.61]), but slower for the *Scatterplot* tasks (1.32 sec CI [-1.38, 4.00]). This finding is inconclusive, though, as confidence intervals cross the zero line. See also Figure 18 in Appendix A.

The ANOVA also showed main effects of factors BASE and Q-TYPE on COMPLETIONTIME. These differences were expected due to the variance in the number of items shown in the base visualization; therefore, we focus primarily on interactions with the visual factors.

5.2 Errors

Errors counted the number of incorrect selections before the participant chose the correct answer. The overall mean for errors was 0.89 errors per trial. Figure 11 shows a summary of mean ERRORS per

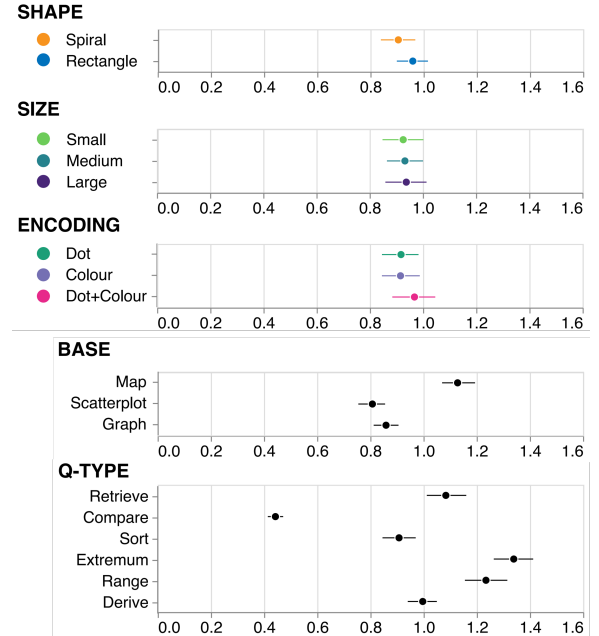


Figure 11: Mean ERRORS for our five independent variables. Error bars show 95% CIs.

trial, for each independent variables; see also Table 2 in Appendix A. No effects were found for any of our main visual factors (SHAPE, SIZE, or ENCODING). There were main effects of both Q-TYPE and BASE, but these differences were again expected due to differences in the number of items and the inherent difficulty of the tasks (e.g., sorting items is more difficult than retrieving a single value). The only interaction with the visual factors was a three-way interaction between SHAPE, ENCODING, and Q-TYPE.

5.3 Hover Count

HOVERCOUNT recorded the number of times a participant hovered over any of the data items highlighted for the task (hovering increased the size of the item) per trial.

Figure 20 shows the mean HOVERCOUNT for each independent variables; see also Table 3 in Appendix A. Factor SHAPE shows a significant effect on HOVERCOUNT, with participants hovering on *Spiral* ETs 1.10 CI [0.90, 1.30] times per trial on average compared to 0.65 CI [0.50, 0.83] times per trial for *Rectangle*. There was also a significant interaction between SHAPE and Q-TYPE: The difference in hover counts between *Spiral* and *Rectangle* was relatively larger for two kinds of questions: computing and comparing derived values (0.80 CI [0.34, 1.25]), and sorting items by a criteria from the time-series (0.82 CI [0.39, 1.25]), whereas CIs for other questions all fall within a [0.19, 0.5] range. See Figure 20 in Appendix A.

5.4 Subjective Effort

After all tasks were complete with a base visualization, participants completed a NASA-TLX survey asking them about their perception of effort. We used the Aligned Rank Transform [78] with the ARTool package in R to enable factorial analysis of our three visual factors (we did not include BASE visualization or Q-TYPE in this analysis).

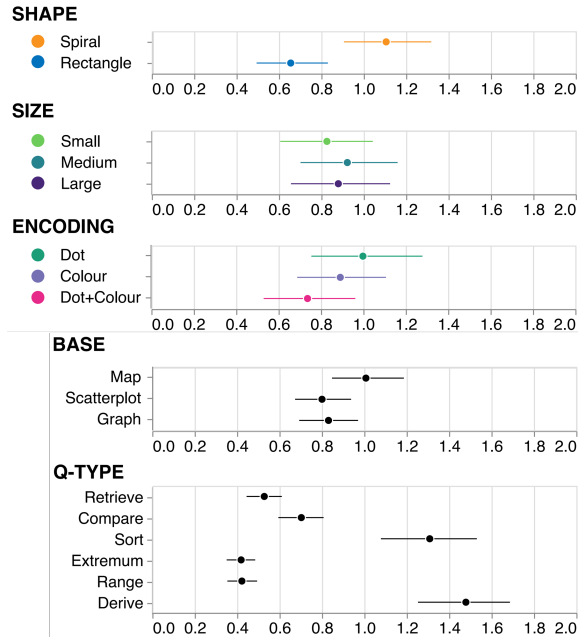


Figure 12: Mean HOVER COUNT for our five independent variables. Error bars show 95% CIs.

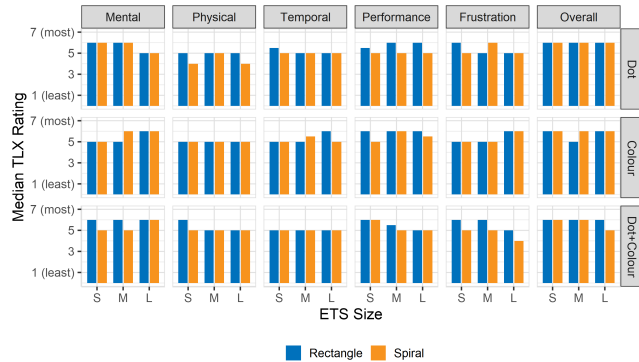


Figure 13: Median ratings for NASA-TLX questions, by SHAPE, SIZE, and ENCODING.

Two participants' answers were removed because they answered the same way on all survey questions.

Participant ratings were similar across all of our main factors, and there were only three significant main effects found: for SHAPE on TLX question 'Physical Effort', and for SHAPE and ENCODING on TLX question 'Subjective Performance'. As can be seen from the median ratings in Figure 13, participants rated *Spiral* as requiring slightly less physical effort than *Rectangle*— but then also rated *Spiral* as having slightly worse performance. Because the differences in the median TLX ratings were uniformly small (one point on the seven-point scale), we do not see strong differences in perceived effort for any of our main factors. The median ratings are all above the midpoints for the response scales, suggesting that our participants did not find the tasks to be easy with any of the configurations of the ETS. See also the factorial results in Table 4 in Appendix A.

6 DISCUSSION

The ETS design framework enabled us to identify specific experimental comparisons for three primary visual factors, and the study provides empirical evidence about how choices in those factors affected interpretation performance. In the following sections we consider the study and framework in a broader context – to explain our results, to discuss how the results and the framework can be used by researchers and practitioners, to explore underlying ideas about reading embedded visual representations, and to outline the limitations of the study and opportunities for future research.

6.1 Interpretation of Results

The study provides initial empirical results about the basic interpretability of ETS visual designs and the relative performance of different choices for ETS shape, size, and encoding:

- ETS visualizations can be highly successful: nearly 400 participants were able to successfully answer a variety of questions using eighteen different ETS representations, across three different base visualizations and datasets.
- Shape matters: our two ETS shapes performed differently, with Rectangle shapes consistently being faster and requiring fewer hovers than Spiral shapes;
- Small size was not a problem: our three sizes performed similarly on all measures (including completion time, errors, hovers, and perceived effort);
- The three encodings also performed similarly, with no major differences between distance-based, colour-based, and combined encodings – across shapes, sizes, and base visualizations.

There are several possible reasons why Spiral was slower than Rectangle. Spiral uses radial geometry, and previous research has suggested that some radial designs may be slower (e.g., [17, 33, 72]). Although other studies have shown advantages for radial layouts (e.g., [12]), it is likely that our non-expert participants were more familiar with Euclidean geometry. Spiral layouts also have a variable scale due to the increasing radius of the curve, and this may have slowed users down, particularly when they needed to compare a range in two different turns of the spiral. Finally, distance-based encodings in spiral layouts require that users judge distances in many directions around the spiral, whereas rectangular layouts only require judgments about distance from the horizontal baseline.

The lack of difference between the sizes was a surprising result. Although a lack of evidence does not prove that there are no size-based differences, our results suggest that any differences are not so large that they are easy to spot in an experimental context. In addition, because we also found no differences in terms of perceived effort, participants were not simply working harder with the small designs. Potential explanations for why we did not find size-based differences include the possibility that the sizes we used were all above a threshold for the tasks – although our pilot testing suggested that the larger sizes did make the tasks easier. It is also possible that the 2X zoom that occurred on hover mitigated any problems with the small representations (although we did not see evidence of increased hovers with the small designs). The lack of evidence for a size difference, however, is potentially good news for ETS designers given the constraints that will be placed on the size

of ETS – our results suggest that users can successfully interpret ETS even when they are small (~1000 pixels in area).

We also found no evidence of differences between the three encodings. This is of interest because the dot-based encodings have a much lower data-to-ink ratio than the colour encoding, but this did not appear to affect performance or perceived effort. There are other encodings to test in future (e.g., the area-based encoding used in horizon charts), but overall, our results again provide cautious good news for designers, who may be forced to use different encodings due to constraints in the base visualization.

6.2 Making use of the Findings

The ETS framework and our empirical study provide both detailed and general insights for designers who need to embed representations of time-series data. At the detailed level, the specific findings of the study can provide initial guidelines: shape can affect people's ability to interpret ETS representations; using spirals could reduce user performance compared to rectangles; smaller sizes can still effectively convey time-series data; both distance-based and colour-based encodings can be successful. Although our study provides information about only a small proportion of the possible designs for ETS, we provide a starting point for designers and a model for others who wish to add further study results.

The framework also sets out constraints and opportunities for designers who want to explore new design possibilities. The examples noted in the framework show that there is a very large opportunity space for generating new kinds of ETS designs – even with 18 variations, our study has only scratched the surface of what designs are likely to succeed (and even in generating examples for the framework, we were consistently surprised at designs that did not conform to our initial expectations). In addition, user interaction techniques with embedded visualizations is an under-explored space: the second part of the framework lists many possibilities for new kinds of interactions and new studies.

The many ways in which ETS design is affected by (and affects) the base visualization is an exciting area for further research. The framework sets out some of these connections, and provides pathways for designers who need to determine appropriate geometry, shape, size, data encoding, and interaction capabilities for an ETS. Along with more general surveys of time-series visualization (e.g., [1, 5]), the framework identifies dimensions that can help researchers and practitioners to describe, compare, and classify ETS designs – and to develop new studies that can add to our empirical knowledge about ETS performance.

In addition to these practical uses, the framework can also serve as an opportunity to expand on ideas of macro/micro readings in information visualization. This idea describes how dense data can provide very specific detail, and, through the emergent visual features that arise when detail is seen at a larger scale, can also provide “big picture” insights. Both macro and micro readings are possible with the same visual marks – that is, “detail cumulates into larger coherent structures” ([68], p.37). The goal of this approach is to increase the amount of information in a visualization, and increase the number of levels at which viewers can read the representation.

Embedded representations increase information density in a visualization – an ETS is an additional dimension represented within

the base. Embedded time-series, however, expand on the conception of macro and micro readings: whereas the original idea considers the aggregation of detail into emergent visual features, ETS provide an orthogonal dimension to the main presentation. If designed well, ETS representations can fit within the macro/micro readings of a base visualization, and also provide new readings at both levels.

A first consideration is how embedding visual structures in the base visualization affects the user's ability to read the base. If ETS are unobtrusive and do not conflict with other marks, users can still read the base visualization for both detail and for overall visual patterns. The idea of reusing existing geometry from the base visualization can be particularly valuable in serving this need – if ETS representations simply add detail to existing visual structures, those structures can still accumulate to show overall insights. An example is a stem-and-leaf plot, which creates a histogram from numbers that are themselves representations of individual items in the histogram. The numbers provide an additional dimension of information, but still allow the overall features of the histogram to be easily read (as Tukey states about these plots, “*If we are going to make a mark, it may as well be a meaningful one.*” ([68], p.46)).

Second, ETS representations themselves provide both micro and macro readings – micro readings of the data within a single ETS, and macro readings that involve multiple ETS. Further research is needed into how time-series visualizations are read at smaller and smaller scales (building on work such as [24, 43] in addition to our study); into ways that representations can still provide useful information even when miniaturized; and into new interaction techniques that can improve inspection of even small ETS. Similarly, some ETS representations can convey a “visual signature” that allows at-a-glance interpretation even at small size, and that can form emergent patterns (and macro readings) when there are multiple ETS in view. Integrating our understanding of how interactions with embedded representations can bridge macro/micro readings has the potential of providing richer insights in dense data spaces (work that has been begun by researchers considering generalized space-time visualizations [5] and perceptual kernels [24]).

Third, some micro/macro readings combine elements of the ETS and the base visualization. For example, an emergent feature of colour-encoded ETS may be to reveal colour patterns that align with geographic regions on a map, perhaps indicating hidden variables. It may also be possible for micro-level inspections of the base visualization to be used in adapting the macro-level composition of ETS representations – e.g., inspecting details in one area of a map could automatically highlight ETSs in other areas that have similar characteristics, or change the ETSs to show relative differences to the area of interest. Although the framework highlights some of the issues that extend between base visualizations and ETS (such as occlusion and interference), more work is needed to explore micro/macro opportunities and issues for visualizations with embedded dimensions, and to document the interplay between a base visualization and embedded elements.

6.3 Limitations and Future Research

There are limitations in the study and the ETS framework, many of which can indicate directions for further work in this area.

First, although our study compared 18 different ETS designs, this is still a small number of the many design possibilities indicated by the framework. In future studies, we plan to consider 3D representations and animations in addition to other types of 2D subplots, and to test additional sizes and encodings. We will carry out an initial comparison of tower-style 3D representations (e.g., Figure 3e) with 2D representations, and will also investigate temporal visualizations such as animated scatterplots, which enable more information to be presented, but introduce additional complexity. Future work should explore the types of analytics tasks that animation supports, including identifying similar evolving trends at several locations, emerging through common-fate perceptual mechanisms [19].

Second, the study focused on micro readings of a small number of ETSs, and our results should be extended to look at combined micro/macro readings that include both the ETS and the base visualization – e.g., assessing overall trends across many ETS, or combining embedded data with information from the base. In addition, although interference between ETS and the base visualization is considered in Section 3.3, further studies are needed to consider how the presence of ETS could change the interpretation of the base visualization (e.g., through occlusion, or from ETS colours altering the perception of colour in the base).

Third, the visualizations used in the study added new ETS representations to the base, but the idea of reusing geometry to show embedded representations should be studied further (e.g., we plan to test our graph/migration example with the time-series data encoded in the graph's links rather than in an added subplot).

Fourth, there were only minimal user interactions built into our study system, and this is a clear area for further study; there are many potential interaction scenarios set out in the framework that can lead to novel techniques and studies (e.g., detailed inspection techniques that help overcome the small size of ETS representations). Fifth, like many visualization studies, our tasks were necessarily artificial, and a clear direction for further work is to explore the use of ETS in real-world usage scenarios.

Fifth, our study used crowd-sourced participants, and although there are advantages to drawing from this population in terms of ecological validity (e.g., a wide variety of ages and experience, as well as a broad range of computing and display hardware), we also plan to carry out additional studies that will investigate different demographics (in particular, experts in visualization and analytics).

Finally, although the ETS framework has gone through several iterations, it has not been formally validated in terms of comprehensiveness, and further explorations in this area can identify ways that the framework can be extended. Because it the first design space to consider embedded time-series representations, it already provides valuable information for designers, but future work will improve its coverage and depth. We plan to investigate whether the framework can be extended to other types of 1D data beyond time-series. There are many visualization systems that could benefit from embedding other kinds of linear datasets (e.g., genomic data, results of sampling processes, or representations of document collections) within an existing visualization.

7 CONCLUSION

Embedded time-series visualizations are representations of a temporal data dimension, inset within a base visualization such as a map, scatterplot, or graph. Much has been written about visualizing time-series on their own, but less is known about how to design embedded versions that preserve the features of the base visualization. To help designers understand the issues involved in creating embedded time-series representations, we created the ETS design framework that sets out constraints and opportunities for embedded visualizations that must work together with the visual and interactional properties of a base visualization. To provide empirical evidence about the performance of ETS designs, we also carried out a crowdsourced study to test the effects of three of the framework's visual factors: shape, size, and encoding. Our results provide initial guidance for designers, and our framework provides insights both into the design of ETS and the ways in which embedded visualizations are (and can be) read and used.

REFERENCES

- [1] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. 2007. Visualizing time-oriented data—A systematic view. *Computers & Graphics* 31, 3 (2007), 401–409. <https://doi.org/10.1016/j.cag.2007.01.030>
- [2] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. 2011. *Visualization of time-oriented data*. Vol. 4. Springer.
- [3] Edgar Anderson. 1957. A semigraphical method for the analysis of complex problems. *Proceedings of the National Academy of Sciences* 43, 10 (1957), 923–927.
- [4] Benjamin Bach, Pierre Dragicevic, Daniel Archambault, Christophe Hurter, and Sheelagh Carpendale. 2014. A review of temporal data visualizations based on space-time cube operations. In *Eurographics conference on visualization*.
- [5] Benjamin Bach, Pierre Dragicevic, Daniel Archambault, Christophe Hurter, and Sheelagh Carpendale. 2017. A descriptive framework for temporal data visualizations based on generalized space-time cubes. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 36–61.
- [6] Benjamin Bach, Charles Perin, Qiuyuan Ren, and Pierre Dragicevic. 2018. Ways of visualizing data on curves. In *TransImage 2018-5th Biennial Transdisciplinary Imaging Conference*. 1–14.
- [7] Benjamin Bach, Conglei Shi, Nicolas Heulot, Tara Madhyastha, Tom Grabowski, and Pierre Dragicevic. 2015. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE transactions on visualization and computer graphics* 22, 1 (2015), 559–568.
- [8] Peter Bak, Florian Mansmann, Halldor Janetzko, and Daniel Keim. 2009. Spatiotemporal analysis of sensor logs using growth ring maps. *IEEE transactions on visualization and computer graphics* 15, 6 (2009), 913–920.
- [9] Michel Beaudouin-Lafon. 2000. Instrumental interaction: an interaction model for designing post-WIMP user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 446–453.
- [10] Richard A Becker, William S Cleveland, and Allan R Wilks. 1987. Dynamic graphics for data analysis. *Statistical science* 2, 4 (1987), 355–383.
- [11] J. Bertin and M. Barbut. 1973. *Sémiologie graphique: les diagrammes, les réseaux, les cartes*. Gauthier Villars. <https://books.google.fr/books?id=F4weAAAAAAAJ>
- [12] Tanja Blascheck, Lonni Besançon, Anastasia Bezerianos, Bongshin Lee, and Petra Isenberger. 2018. Glanceable visualization: Studies of data comparison performance on smartwatches. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 630–640.
- [13] Ilya Boyandin, Enrico Bertini, and Denis Lalanne. 2012. A qualitative study on the exploration of temporal changes in flow maps with animation and small-multiples. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1005–1014.
- [14] Richard Brath. 2020. *Visualizing with Text*. CRC Press.
- [15] Matthew Brehmer, Bongshin Lee, Benjamin Bach, Nathalie Henry Riche, and Tamara Munzner. 2016. Timelines revisited: A design space and considerations for expressive storytelling. *IEEE transactions on visualization and computer graphics* 23, 9 (2016), 2151–2164.
- [16] Erin M Buchanan and John E Scofield. 2018. Methods to detect low quality data and its implication for psychological research. *Behavior research methods* 50, 6 (2018), 2586–2596.
- [17] Michael Burch, Fabian Beck, and Stephan Diehl. 2008. Timeline trees: visualizing sequences of transactions in information hierarchies. In *Proceedings of the working conference on Advanced visual interfaces*. 75–82.
- [18] M Sheelagh T Carpendale and Catherine Montagnese. 2001. A framework for unifying presentation space. In *Proceedings of the 14th annual ACM symposium*

- on User interface software and technology. 61–70.
- [19] Amira Chalbi, Jacob Ritchie, Deokgun Park, Jungu Choi, Nicolas Roussel, Niklas Elmquist, and Fanny Chevalier. 2019. Common fate for animated transitions in visualization. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 386–396.
 - [20] William S Cleveland. 1993. *Visualizing data*. Hobart press.
 - [21] William S Cleveland and Irma J Terpenning. 1982. Graphical methods for seasonal adjustment. *J. Amer. Statist. Assoc.* 77, 377 (1982), 52–62.
 - [22] Geoff Cumming and Sue Finch. 2005. Inference by eye: confidence intervals and how to read pictures of data. *American psychologist* 60, 2 (2005), 170.
 - [23] Çağatay Demiralp, Michael S Bernstein, and Jeffrey Heer. 2014. Learning perceptual kernels for visualization design. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 1933–1942.
 - [24] Çağatay Demiralp, Carlos E Scheidegger, Gordon L Kindlmann, David H Laidlaw, and Jeffrey Heer. 2014. Visual embedding: A model for visualization. *IEEE Computer Graphics and Applications* 34, 1 (2014), 10–15.
 - [25] Dazhen Deng, Weiwei Cui, Xiyu Meng, Mengye Xu, Yu Liao, Haidong Zhang, and Yingcai Wu. 2022. Revisiting the design patterns of composite visualizations. *IEEE Transactions on Visualization and Computer Graphics* 29, 12 (2022), 5406–5421.
 - [26] Thomas J DiCiccio and Bradley Efron. 1996. Bootstrap confidence intervals. *Statistical science* 11, 3 (1996), 189–228.
 - [27] Niklas Elmquist, Thanh-Nghi Do, Howard Goodell, Nathalie Henry, and Jean-Daniel Fekete. 2008. ZAME: Interactive large-scale graph visualization. In *2008 IEEE Pacific visualization symposium*. IEEE, 215–222.
 - [28] Niklas Elmquist, Pierre Dragicevic, and Jean-Daniel Fekete. 2008. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE transactions on Visualization and Computer Graphics* 14, 6 (2008), 1539–1148.
 - [29] ESRI/ArcGIS. [n.d.]. *Create and use maps with pie chart symbols*. Retrieved August 30, 2022 from <https://doc.arcgis.com/en/insights/latest/create/pie-chart-symbols.htm>
 - [30] S Few. 2008. Time on the horizon. *visual business intelligence newsletter, Jun/Jul 2008* (2008). http://perceptualedge.com/articles/visual_business_intelligence/time_on_the_horizon.pdf
 - [31] Clifton Forlines and Kent Wittenburg. 2010. Wakame: sense making of multi-dimensional spatial-temporal data. In *Proceedings of the international conference on advanced visual interfaces*. 33–40.
 - [32] Peter Gatalsky, Natalia Andrienko, and Gennady Andrienko. 2004. Interactive analysis of event data using space-time cube. In *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004*. IEEE, 145–152.
 - [33] Joseph Goldberg and Jonathan Helfman. 2011. Eye tracking for visualization evaluation: Reading values on linear versus radial graphs. *Information visualization* 10, 3 (2011), 182–195.
 - [34] Connor C Gramazio, Karen B Schloss, and David H Laidlaw. 2014. The relation between visualization size, grouping, and user performance. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 1953–1962.
 - [35] Garrett Grolemund and Hadley Wickham. 2015. Visualizing complex data with embedded plots. *Journal of Computational and Graphical Statistics* 24, 1 (2015), 26–43.
 - [36] Henning Gruendl, Patrick Riehmann, Yves Pausch, and Bernd Froehlich. 2016. Time-Series Plots Integrated in Parallel-Coordinates Displays. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 321–330.
 - [37] Diansheng Guo. 2008. Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP). *International Journal of Geographical Information Science* 22, 7 (2008), 801–823.
 - [38] Steve Haroz, Robert Kosara, and Steven L Franconeri. 2015. The connected scatterplot for presenting paired time series. *IEEE transactions on visualization and computer graphics* 22, 9 (2015), 2174–2186.
 - [39] Robert L Harris. 1999. *Information graphics: A comprehensive illustrated reference*. Oxford University Press, USA.
 - [40] Mark Harrower, Sara Fabrikant, and M Dodge. 2008. The role of map animation in geographic visualization. (2008).
 - [41] Susan Havre, Elizabeth Hetzler, Paul Whitney, and Lucy Nowell. 2002. The-meriver: Visualizing thematic changes in large document collections. *IEEE transactions on visualization and computer graphics* 8, 1 (2002), 9–20.
 - [42] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. *Proceedings of the 28th Annual CHI Conference on Human Factors in Computing Systems* (2010), 203–212. <https://doi.org/10.1145/1753326.1753357>
 - [43] Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. 2009. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1303–1312.
 - [44] Shinobu Ishihara. 1987. *Test for colour-blindness*. Kanehara Tokyo, Japan.
 - [45] Waqas Javed and Niklas Elmquist. 2012. Exploring the design space of composite visualization. In *2012 IEEE Pacific visualization symposium*. IEEE, 1–8.
 - [46] Jaemin Jo, Frédéric Vernier, Pierre Dragicevic, and Jean-Daniel Fekete. 2018. A declarative rendering model for multiclass density maps. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 470–480.
 - [47] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. 2008. Visual analytics: Definition, process, and challenges. In *Information visualization*. Springer, 154–175.
 - [48] Yngve Sekse Kristiansen and Stefan Bruckner. 2020. Visception: An interactive visual framework for nested visualization design. *Computers & Graphics* 92 (2020), 13–27.
 - [49] Timothy R Levine and Craig R Hullett. 2002. Eta squared, partial eta squared, and misreporting of effect size in communication research. *Human Communication Research* 28, 4 (2002), 612–625.
 - [50] Yang Liu and Jeffrey Heer. 2018. Somewhere over the rainbow: An empirical assessment of quantitative colormaps. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.
 - [51] Liam McNabb and Robert S Laramée. 2019. Multivariate Maps—A Glyph-Placement Algorithm to Support Multivariate Geospatial Visualization. *Information* 10, 10 (2019), 302.
 - [52] Stefaner Moritz. 2012. OECD Better Life Index. <https://truth-and-beauty.net/projects/oecd-better-life-index>.
 - [53] Ali Neshati, Launa Leboe-McGowan, Jason Leboe-McGowan, Marcos Serrano, Pourang Irani, et al. 2019. G-sparks: Glanceable sparklines on smartwatches. In *45th Conference on Graphics Interface (GI 2019)*. 1–9.
 - [54] Anna Offenwanger, Matthew Brehmer, Fanny Chevalier, and Theophanis Tsandilas. 2023. Timesplines: Sketch-based authoring of flexible and idiosyncratic timelines. *IEEE Transactions on Visualization and Computer Graphics* 30, 1 (2023), 34–44.
 - [55] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. 1996. LifeLines: visualizing personal histories. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 221–227.
 - [56] Ghulam Jilani Quadri and Paul Rosen. 2021. A survey of perception-based visualization studies by task. *IEEE Transactions on Visualization and Computer Graphics* (2021).
 - [57] Ramana Rao and Stuart K Card. 1994. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 318–322.
 - [58] Sandra Rendgen. 2018. *The minard system: the complete statistical graphics of Charles-Joseph Minard*. Chronicle Books.
 - [59] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. 2008. Effectiveness of animation in trend visualization. *IEEE transactions on visualization and computer graphics* 14, 6 (2008), 1325–1332.
 - [60] Christian Rohrdantz, Milos Krstajic, Mennatallah El Assady, and Daniel Keim. 2012. What’s going on? How Twitter and online news can work in synergy to increase situational awareness.
 - [61] A Slingsby. 2018. Tilemaps for summarising multivariate geographical variation. (2018). <https://openaccess.city.ac.uk/id/eprint/20884/>
 - [62] Guodao Sun, Ronghua Liang, Huamin Qu, and Yingcai Wu. 2016. Embedding spatio-temporal information into maps by route-zooming. *IEEE transactions on visualization and computer graphics* 23, 5 (2016), 1506–1519.
 - [63] Sidharth Thakur and Andrew J Hanson. 2010. A 3D visualization of multiple time series on maps. In *2010 14th International Conference Information Visualisation*. IEEE, 336–343.
 - [64] The New York Times. 2021. Coronavirus in the U.S.: Latest Map and Case Count. <https://www.nytimes.com/interactive/2021/us/covid-cases.html>.
 - [65] Christian Tominski, Petra Schulze-Wollgast, and Heidrun Schumann. 2005. 3d information visualization for time dependent data on maps. In *Ninth International Conference on Information Visualisation (IV'05)*. IEEE, 175–181.
 - [66] Christian Tominski and Heidrun Schumann. 2008. Enhanced interactive spiral display. In *SIGRAD 2008. The annual SIGRAD conference special theme: interaction; November 27-28; 2008 Stockholm; Sweden*. Linköping University Electronic Press, 53–56.
 - [67] Edward R Tufte. 2006. *Beautiful evidence*. Graphis Pr.
 - [68] Edward R Tufte, Nora Hillman Goeler, and Richard Benson. 1990. *Envisioning information*. Vol. 126. Graphics press Cheshire, CT.
 - [69] Eugene Turner. 1977. Life in LA. <https://mapdesign.icaci.org/2014/12/mapcarte-353365-life-in-los-angeles-by-eugene-turner-1977/>
 - [70] Jarke J Van Wijk and Edward R Van Selow. 1999. Cluster and calendar based visualization of time series data. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis'99)*. IEEE, 4–9.
 - [71] Fernanda B Viégas, Danah Boyd, David H Nguyen, Jeffrey Potter, and Judith Donath. 2004. Digital artifacts for remembering and storytelling: Posthistory and social network fragments. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*. IEEE, 10–pp.
 - [72] Manuela Waldner, Alexandra Diehl, Denis Gračanin, Rainer Splechtna, Claudio Delrieux, and Krešimir Matković. 2019. A comparison of radial and linear charts for visualizing daily patterns. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 1033–1042.
 - [73] Marc Weber, Marc Alexa, and Wolfgang Müller. 2001. Visualizing time-series on spirals. In *Infovis*, Vol. 1. 7–14.

- [74] Hadley Wickham, Heike Hofmann, Charlotte Wickham, and Dianne Cook. 2012. Glyph-maps for visually exploring temporal patterns in climate data and models. *Environmetrics* 23, 5 (2012), 382–393.
- [75] Infovis Wiki. [n.d.]. *2005-08-24: Integrating graphs into Treemaps*. Retrieved August 30, 2022 from https://infovis-wiki.net/wiki/2005-08-24:_Integrating_graphs_into_Treemaps
- [76] Wikipedia. [n.d.]. *Joseph Priestly's A Chart of Biography*. Retrieved August 30, 2022 from https://en.wikipedia.org/wiki/A_Chart_of_Biography
- [77] Wikipedia. [n.d.]. *William Playfair*. Retrieved August 30, 2022 from en.wikipedia.org/wiki/William_Playfair
- [78] Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 143–146.
- [79] Guangchuan Yu and Maintainer Guangchuan Yu. 2018. Package 'scatterpie'. *R Package version* 1, 2 (2018).
- [80] Xiaoru Yuan, Peihong Guo, He Xiao, Hong Zhou, and Huamin Qu. 2009. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1001–1008.
- [81] Gazi Md Hasnat Zahan, Debajyoti Mondal, and Carl Gutwin. 2021. Contour line stylization to visualize multivariate information. In *Graphics Interface 2021*.
- [82] Di Zhang, Ligu Zhu, Chengcheng Wang, and Lei Zhang. 2016. TimeSpiral, an enhanced interactive visual system for time series data. In *2016 2nd International Conference on Information Management (ICIM)*. IEEE, 127–133.
- [83] Jian Zhao, Fanny Chevalier, and Ravin Balakrishnan. 2011. Kronominer: using multi-foci navigation for the visual exploration of time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1737–1746.
- [84] Qian Zhu, Tao Lu, Shunan Guo, Xiaojuan Ma, and Yalong Yang. 2024. Compositingvis: Exploring interactions for creating composite visualizations in immersive environments. *IEEE Transactions on Visualization and Computer Graphics* 31, 1 (2024), 591–601.

A STUDY DETAILS

For the analysis, values of the dependent measures were capped at maximum values that were informed by informal pilots and our assessments of what was reasonable for the tasks (e.g., considering how long it would take a participant to reasonably answer every choice): maximums were set at 90 seconds per question for completion time, 5 incorrect answers, and 10 hovers (which allows at least 3 hovers per highlighted target). The number of capped values were as follows:

Completion-time values were capped at 90 seconds; 113 values (of 9552) were capped in this way.

Error values were capped at 5 errors, as described above; 51 values (of 9552) were capped in this way.

Hover counts were capped at 10 hovers, as described above; 105 of 9552 values were capped in this fashion.

A.1 Study interfaces

Figure 14 shows a larger version of the scatterplot base visualization and Covid-19 ETS dataset. Figure 15 shows the map base visualization and the daily-temperature dataset. Figure 16 shows the graph base visualization and the human-migration dataset.

A.2 Completion Time

Table 1 shows results of the factorial ANOVA for the model $\text{COMPLETIONTIME} \sim \text{SHAPE} \times \text{SIZE} \times \text{ENCODING} \times \text{BASE} \times \text{Q-TYPE} \times \text{Rand}(\text{Participant})$.

Table 1: Factorial analysis results – COMPLETIONTIME (to correct answer). Only significant interactions are shown.

Factor	DF	F	p	η^2
SHAPE	1,380	10.57	.001	.013
SIZE	2,380	0.10	.90	–
ENCODING	2,380	0.29	.75	–
BASE	2,760	24.73	<.001	.006
Q-TYPE	5,1900	35.95	<.001	.014
SHAPE $\times$ BASE	2,760	5.12	.006	.0015
SIZE $\times$ BASE	4,760	3.60	.006	.002

A.3 Errors

Table 2 shows results of the factorial ANOVA for the model $\text{ERRORS} \sim \text{SHAPE} \times \text{SIZE} \times \text{ENCODING} \times \text{BASE} \times \text{Q-TYPE} \times \text{Rand}(\text{Participant})$.

Table 2: Factorial analysis results – ERRORS (number of incorrect answers). Only significant interactions are shown.

Factor	DF	F	p	η^2
SHAPE	1,380	1.09	.29	–
SIZE	2,380	0.18	.83	–
ENCODING	2,380	0.47	.63	–
BASE	2,760	100.2	<.001	.028
Q-TYPE	5,1900	135.2	<.001	.073
SHAPE $\times$ ENCODING $\times$ Q-TYPE	10,1900	2.49	.006	.003

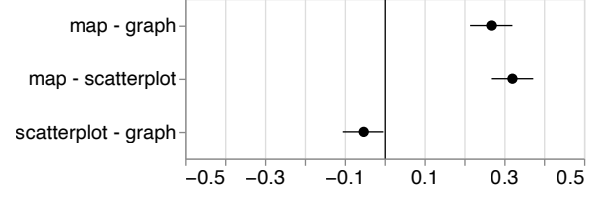


Figure 17: Mean effect of SHAPE on ERRORS. Error bars show 95% CIs.

A.4 Hover Count

Table 3 shows results of the factorial ANOVA for the model $\text{HOVERCOUNT} \sim \text{SHAPE} \times \text{SIZE} \times \text{ENCODING} \times \text{BASE} \times \text{Q-TYPE} \times \text{Rand}(\text{Participant})$.

Table 3: Factorial analysis results – HOVERCOUNT (number of hovers on highlighted items). Only significant interactions are shown.

Factor	DF	F	p	η^2
SHAPE	1,380	11.25	<.001	.016
SIZE	2,380	0.31	.73	–
ENCODING	2,380	1.18	.31	–
BASE	2,760	17.75	<.001	.003
Q-TYPE	5,1900	98.61	<.001	.060
SHAPE $\times$ Q-TYPE	5,1900	9.81	<.001	.006

Table 4 shows results of the factorial ANOVA for TLX scores (transformed using the Aligned Rank Transform) for factors SHAPE, SIZE, and ENCODING.

Table 4: Factorial analysis results – TLX scores.

TLX Scale	SHAPE	SIZE	ENCODING
Mental	$F=.36, p=.55$	$F=.30, p=.74$	$F=1.41, p=.25$
Physical	$F=13.99, p<.001$	$F=.15, p=.86$	$F=.64, p=.53$
Temporal	$F=.53, p=.47$	$F=1.64, p=.20$	$F=.12, p=.89$
Performance	$F=17.04, p<.001$	$F=.14, p=.87$	$F=4.16, p=.016$
Effort	$F=.95, p=.33$	$F=.05, p=.95$	$F=2.05, p=.13$
Frustration	$F=3.00, p=.084$	$F=.077, p=.93$	$F=2.72, p=.067$

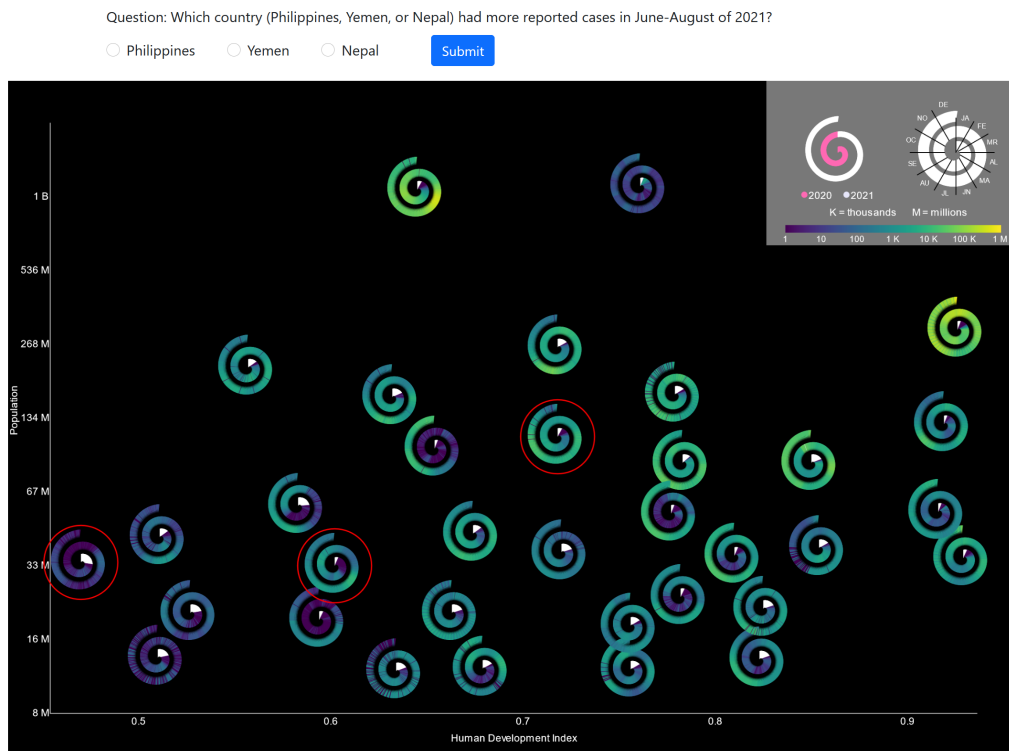


Figure 14: Study interface: scatterplot base and Covid dataset.

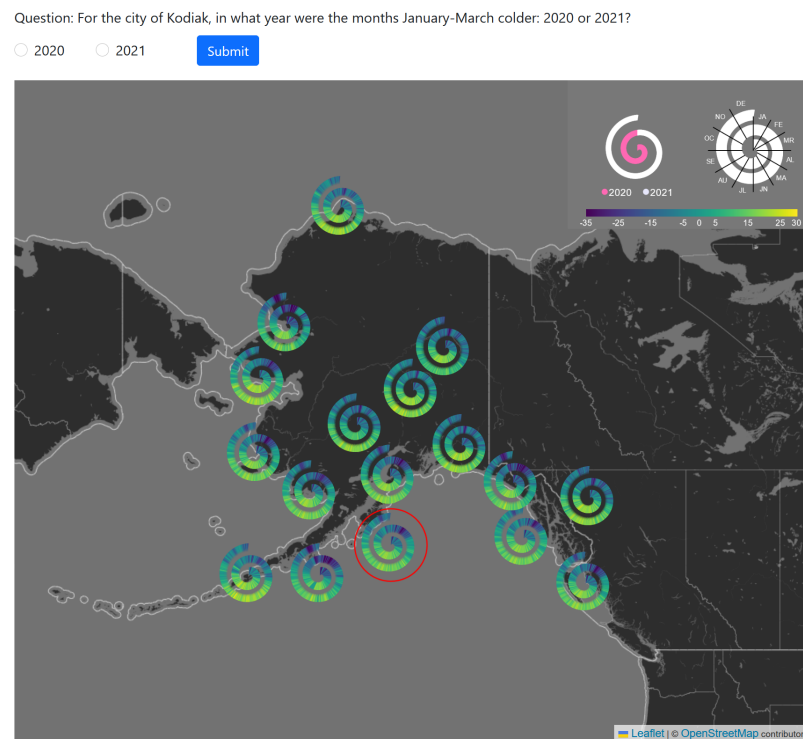


Figure 15: Study interface: map base and temperature dataset.

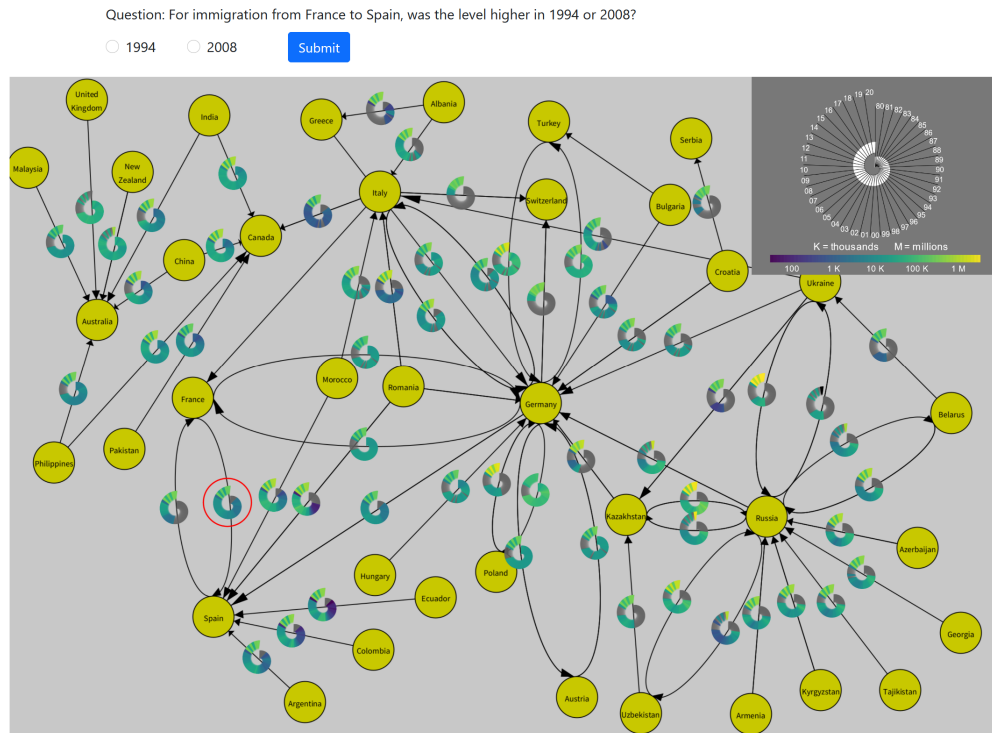
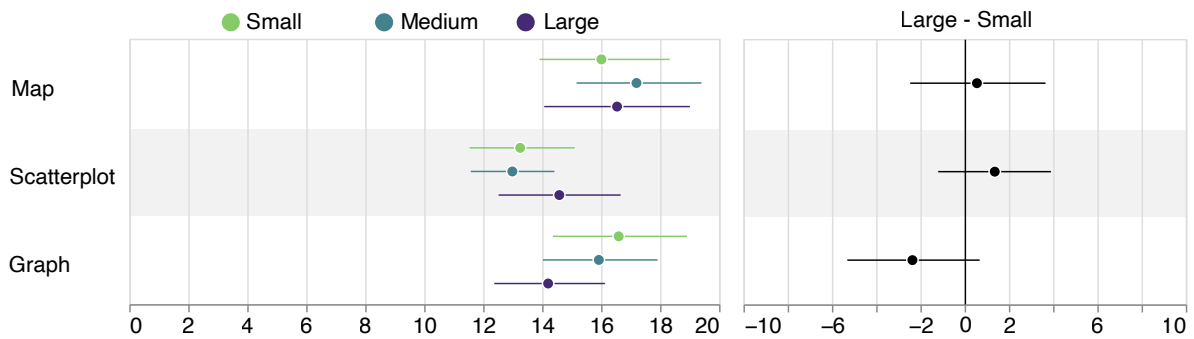
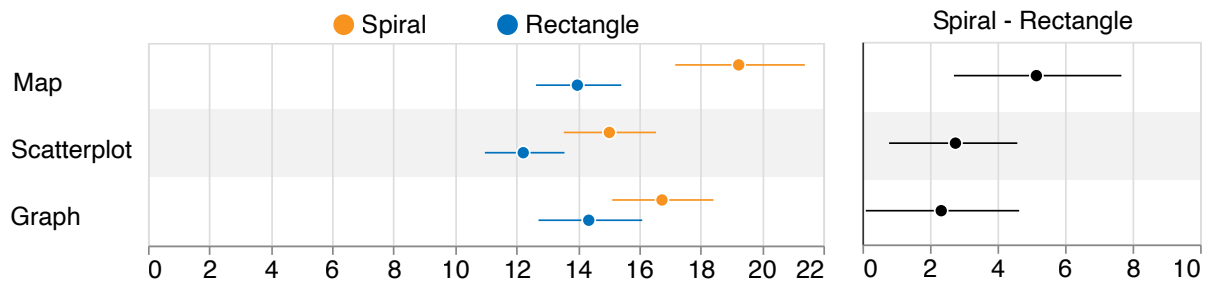


Figure 16: Study interface: graph base and migration dataset.

Figure 18: Left: Mean COMPLETION TIME (sec) by SIZE $\times$ BASE, and Right: effect size. Error bars show 95% CIs.Figure 19: Left: Mean COMPLETION TIME (sec) by SHAPE $\times$ BASE, and Right: effect size. Error bars show 95% CIs.

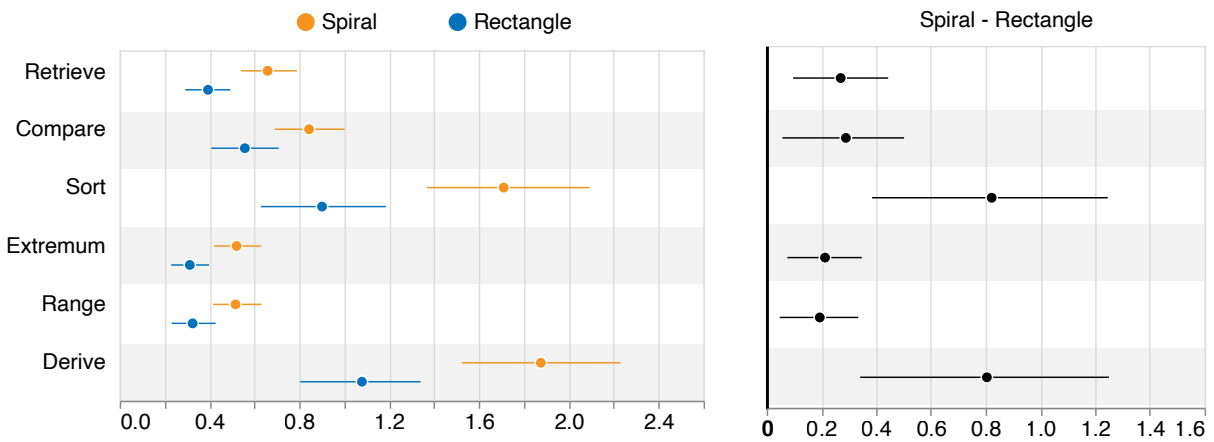


Figure 20: Left: Mean HOVER COUNT by SHAPE and QUESTION TYPE, and Right: effect size. Error bars show 95% CIs.