SUPPLEMENTAL MATERIAL

TimeElide: Visual Analysis of Non-Contiguous Time Series Slices

Michael Oppermann, Luce Liu, and Tamara Munzner

Contents

SUP-1. Datasets

We used multiple datasets from different domains and with varying characteristics during the development of TimeElide, and made them available in the online demo. The datasets are summarized in Table 1 below.

Data source		No. of data points	Interval of data points within a time slice	No. of data points within a time slice	Time slice length	No. of time slices	Interval of time slices	Date range	CSV file size
Occupancy	Lecture building (continuous)	70,922	Fixed: 5 minutes	Dozens to hundreds	Fixed or variable	Dozens to hundreds	Fixed or variable	Apr 22 - Dec 31, 2018	1.6 MB
	Community building (continuous)	93,523	Fixed: 5 minutes	Dozens to hundreds	Fixed or variable	Dozens to hundreds	Fixed or variable	Apr 22, 2018 - Aug 1 , 2019	2.2 MB
	Community building (Fridays, 6pm-12am)	3388	Fixed: 5 minutes	72	Fixed: 6pm-12am	47	Fixed: every Friday	Apr 27, 2018 - Jul 26, 2019	84 KB
Bakery transactions [1]		10,693	Fixed: 15 minutes	Dozens to hundred	Fixed or variable	Dozens to hundreds	Fixed or variable	Oct 30, 2016 - Apr 9, 2017	246 KB
Soccer player in-match events [2]		396	Fixed: 5 minutes	18	Fixed: 90 minutes	22	Variable	Aug 18, 2017 - Mar 31, 2018	12 KB
Bike rides [3]		403	Fixed: 5 minutes	Dozens to hundred	Variable	20	Variable	Jul 18 - Dec 8, 2020	22 KB

Table 1.: Summary of datasets.

Three datasets are publicly available:

(1) Bakery data:
`https://kaggle.com/sulmansarwar/transactions-from-a-bakery`

(2) Soccer data:
`https://doi.org/10.1038/s41597-019-0247-7`

(3) Bike data:
`https://kaggle.com/synapsizeus/strava-bike-rides-from-2020-gpx-files`

SUP-2. Analysis Questions and Tasks

Stakeholder questions related to the analysis of building occupancy during specific time periods motivated the investigation originally. We generalized our approach and included additional datasets and typical analysis questions. A match between these domain-specific questions and abstract tasks (T1-T5) is shown in Table 2.

Domain	Question	Tasks					Required temporal order
		T1. Extrema, avg.	T2. Trends	T3. Patterns	T4. Comparison	T5. Duration	
Occupancy	How many people use the space on Friday evenings?		■				
	When do people usually leave on Friday evenings? Are they coming and going?		■	■			
	Does it happen often that people occupy the space on Friday evenings or are there irregular recurring events?		■	■			■
	Which Friday was the busiest?	■					■
	Are there any noticeable effects of the interventions compared to last week? Are people leaving earlier?				■		■
Bike rides	How did my average speed improve over the last few months?		■				■
	What is my top speed and how often did I get close to it?	■	■	■			■
	How does my speed change during a workout? Is there a significant drop after the first x kilometres? Is my speed getting more consistent?		■	■			■
	Are there any negative outliers?			■			
	When was the longest workout?					■	■
	How does the length of my workouts change over time?					■	■
Bakery transactions	Do we sell more on Saturdays or Sundays?		■		■		■
	How much do we sell throughout the day? Are most sales during the first two hours in the morning?		■	■			
	What is the general sales trend? Are there high fluctuations between weekends? Are there seasonal variations?		■		■		■
	On which weekend did we sell most?	■					■
	Is there a change in our sales this week/month as a result of our recent marketing/promotions?				■		■
Soccer in-match activity	How did my performance change throughout the season?		■				■
	Are there any differences between afternoon and evening matches?			■			■
	What was the activity before and after my injury?			■	■		■

Table 2.: Match between domain-specific questions and abstract tasks.
■ Match, ■ Partial match, □ No match

SUP-3. Manually Specify Desired Slices

TimeElide provides an input form for users to specify slices manually, as shown in Figure 1. Slices can be defined for different days of the week and time periods, but must be shorter than or equal to 24 hours.

According to these slice specifications and the first and last timestamp in the dataset, we generate a set of empty slice arrays that correspond to actual dates. We then iterate through all data points and assign them to these slice arrays based on their timestamps. Some slice arrays may remain empty in case of missing data, which is indicated visually in the visualization.

If users want to analyze slices related to specific dates, in contrast to days of the

week, they need to slice the data in a separate preprocessing step and then choose the option to automatically detect existing slices in TimeElide, as we describe in the following section.

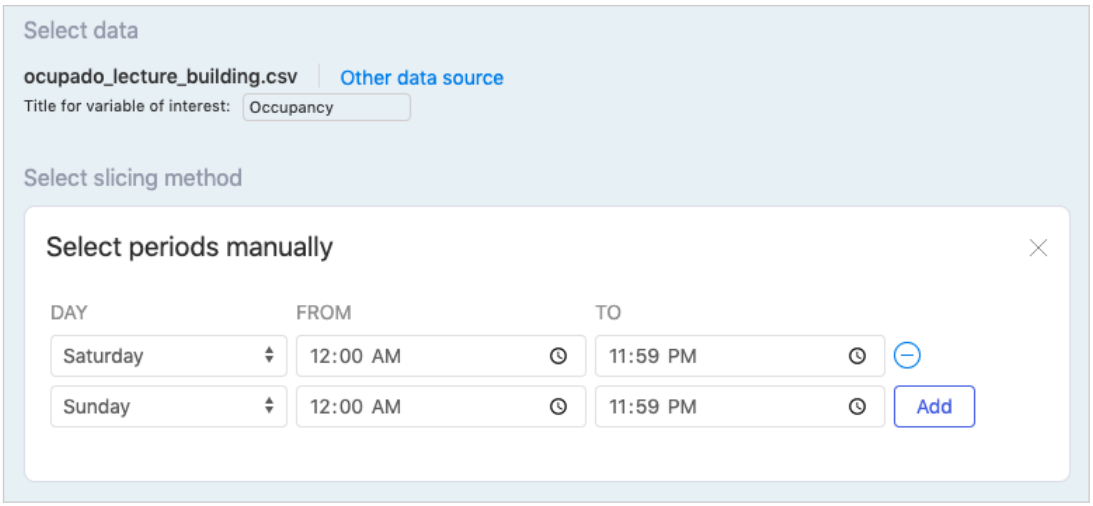


Figure 1. TimeElide manual slicing selection.

SUP-4. Automatically Detect Existing Slices

TimeElide can detect slices automatically to support the following two cases:

- For some datasets, the slices can be inferred from the data rather than being specified by the user. For instance, workouts can occur at different times and days of the week and vary in lengths. Defining manual slices for those cases is not effective.
- Some datasets are generally continuous but users may want to analyze very specific slices that are not supported by TimeElide's manual slicing interface, such as public holidays, and therefore, they remove data outside of these periods in a prior preprocessing step outside of TimeElide.

We implemented a simple mechanism to detect slices based on the differences between timestamps. When iterating through the data, the key question of interest is if a data point is still in the current slice or in a new slice. To determine an appropriate threshold and group the data points into slices, we use the following process:

(1) Sort all timestamps.
(2) Iterate through all data points and determine the differences between consecutive timestamps (in seconds) and save them in a new list.
(3) Remove the top and bottom 1% of this list to minimize the influence of outliers.
(4) Compute the standard deviation of the filtered list and use it as a threshold.
(5) Iterate through the original dataset again and use the threshold to group the data points.

Figure 2 illustrates the differences between timestamps and the detected threshold for four different datasets that are naturally divided into slices (bike rides, soccer games) or where the data has been sliced in a preprocessing step (occupancy on Fridays 6pm-midnight, opening hours of bakery).

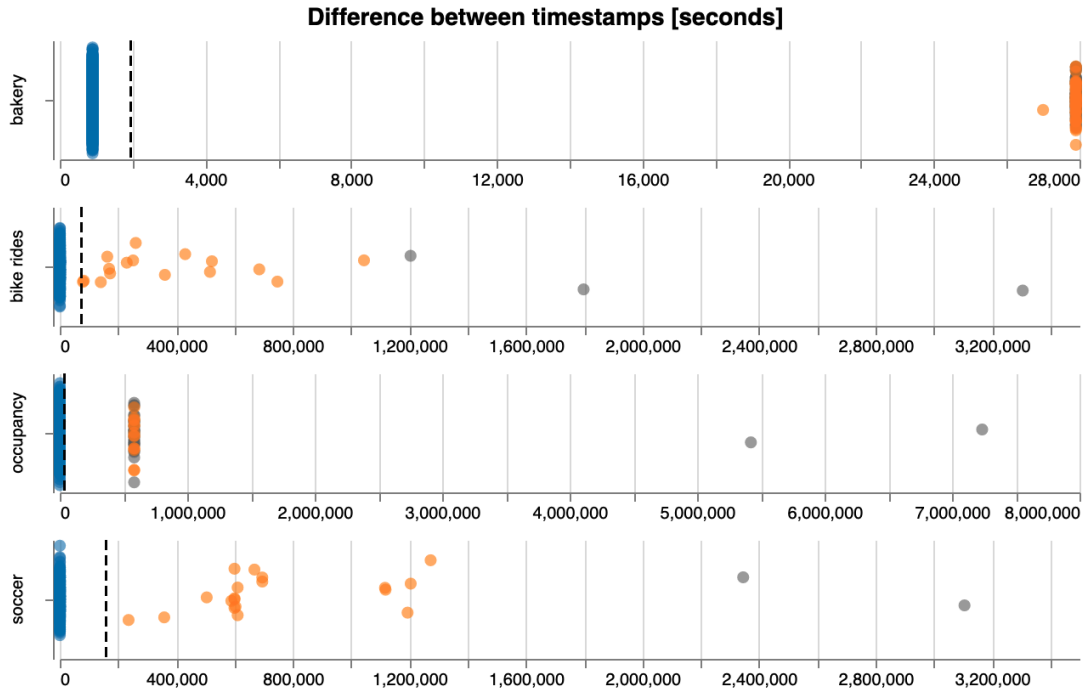


Figure 2. Automatically detect existing slices. Each dot corresponds to the difference between two consecutive timestamps. The black dashed line indicates the detected threshold. The bottom and top 1% of differences (grey dots) are considered outliers and are not used for determining the threshold. If the differences between two data points is smaller than the threshold, the data points are in the same slice (blue), otherwise they are in separate slices (orange).

SUP-5. Usage Scenarios

In this section, we describe two additional usage scenarios that relate to bakery sales transactions and bike rides, to provide further evidence of the utility of TimeElide.

SUP-5.1. Usage Scenario: Bakery Transactions

A bakery manager wants to find out when the store is busiest and when extra help is most needed. They use the number of transactions as a proxy measure because this data has been already recorded over several months.

The manager uploads the dataset to TimeElide and matches the columns of the CSV file with the two required variables *timestamp* and *value*, as shown in Figure 3. They also rename the variable of interest *"# Items sold"* in the sidebar, as illustrated in Figure 4-1 to create more descriptive axis titles.

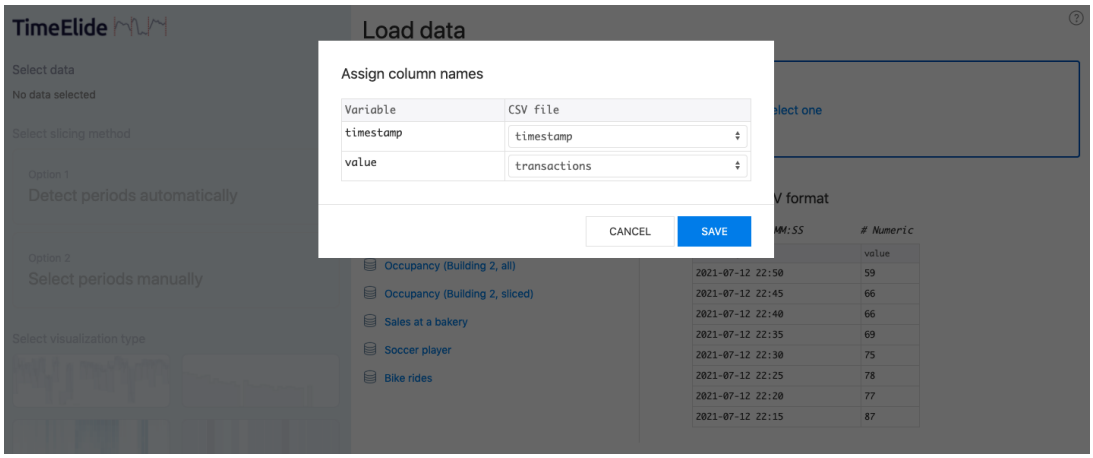


Figure 3. Assign columns of uploaded CSV file to *timestamp* and *value* variables.

The manager chooses to automatically detect slices to see each day of the week as a separate slice (Figure 4-2). TimeElide detects 162 slices and visualizes them as a 2D heatmap by default. The 2D heatmap confirms the manager's prior assumption that most transactions occur between 9am and 5pm (Figure 4-4).

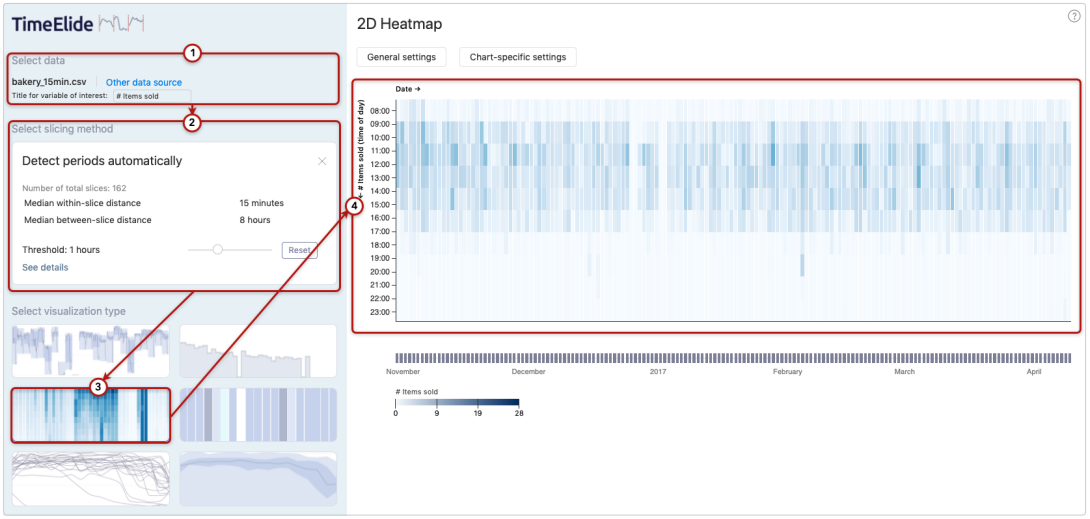


Figure 4. Annotated screenshot of TimeElide: Analyzing the number of transactions at a bakery at different days and times of the week using the 2D heatmap. (1) Rename title of variable of interest to *"# Items sold"*, (2) select automatic slicing method, (3) 2D heatmap gets suggested, and (4) 2D heatmap with absolute within-slice timing is displayed in the main view.

They now want to analyze the average number of transactions on different weekdays and if there are any trends or repeating patterns. The manager selects *stepped area chart* (Figure 5) and hovers over the peak bars to learn that they are all Saturdays. In addition, they observe a weekly pattern with more sales taking place between Friday and Sunday compared to the Monday to Thursday period.

Figure 5. Annotated screenshot of TimeElide: Analyzing the average number of transactions at a bakery on different days of the week using the stepped area chart.

The manager changes the slicing method to manually specify desired slices and selects only Fridays, Saturdays, and Sundays (Figure 6-1). The goal is to do a more fine-grained analysis for those days to determine busy hours. The visualization type is changed to 2D heatmap again and the manager adjusts the number of bins to 30, so they can see the average number of transactions in 30-minute intervals (Figure 6-3). They can see that there are often spikes in sales between 10am and 2pm, and decide to hire additional staff for those periods.

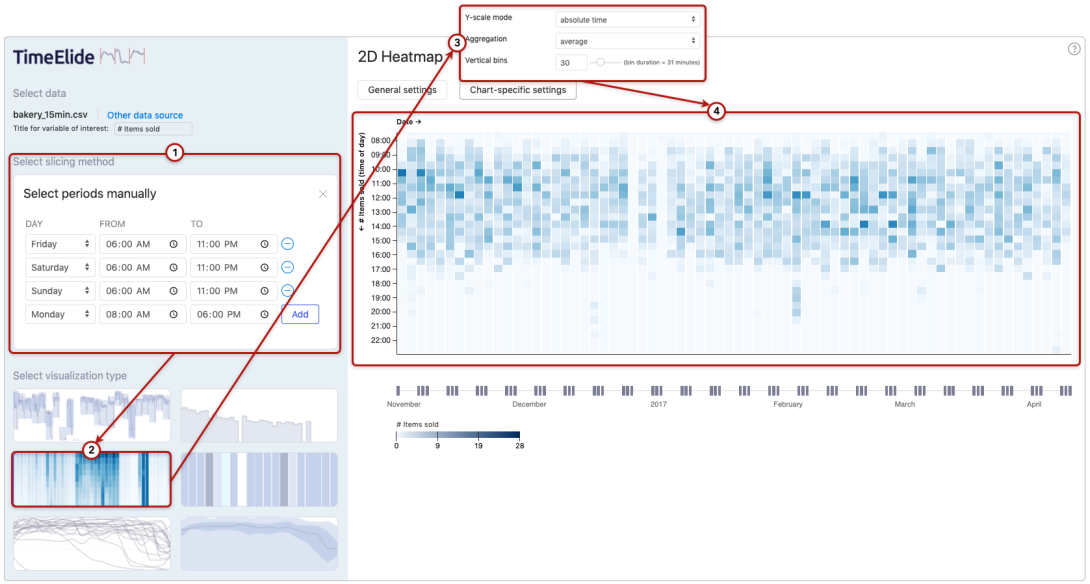


Figure 6. Annotated screenshot of TimeElide: Analyzing the number of transactions at a bakery on Fridays, Saturdays, and Sundays using the 2D heatmap.

SUP-5.2. Usage Scenario: Bike Rides

A hobby athlete tracked their bike rides over the last few months and would like to analyze the speed during all the workout logs in one place. They concatenate the downloaded datasets into one CSV file and upload it to TimeElide to get an overview of all the data.

The user selects the option for automatically detecting existing slices in the dataset. They also select the *2D heatmap* as a visualization type and change the y-scale mode to *absolute duration* and the number of bins to 15 (Figure 7). At a glance the user can see that most of the rides have been around one hour long, with no distinct upwards or downwards trend across the months. It appears that the rides earlier in the year (closer to the left side) have been faster on average.

To confirm, they choose *sparkboxes* in the left sidebar and adjust the settings: normalize slice widths to hold ride length constant, remove raw data layer and replace median with average speed (Figure 8). The result is a visualization with box-plot-bars summarizing each bike ride. There looks to be more variation in average speed in the latter months, but the long-term change seems to be negligible overall.

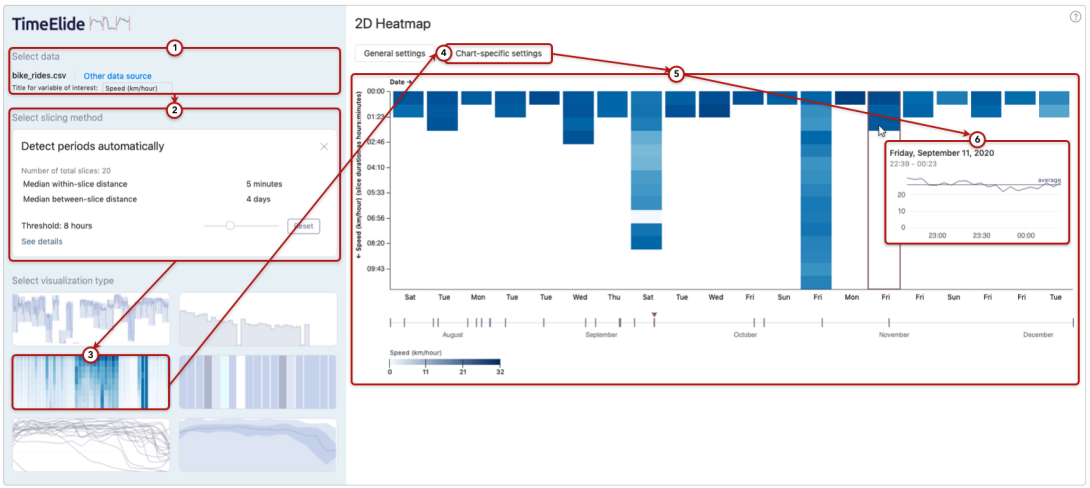


Figure 7. Annotated screenshot of TimeElide: Analyzing the speed during bike rides using the 2D heatmap.

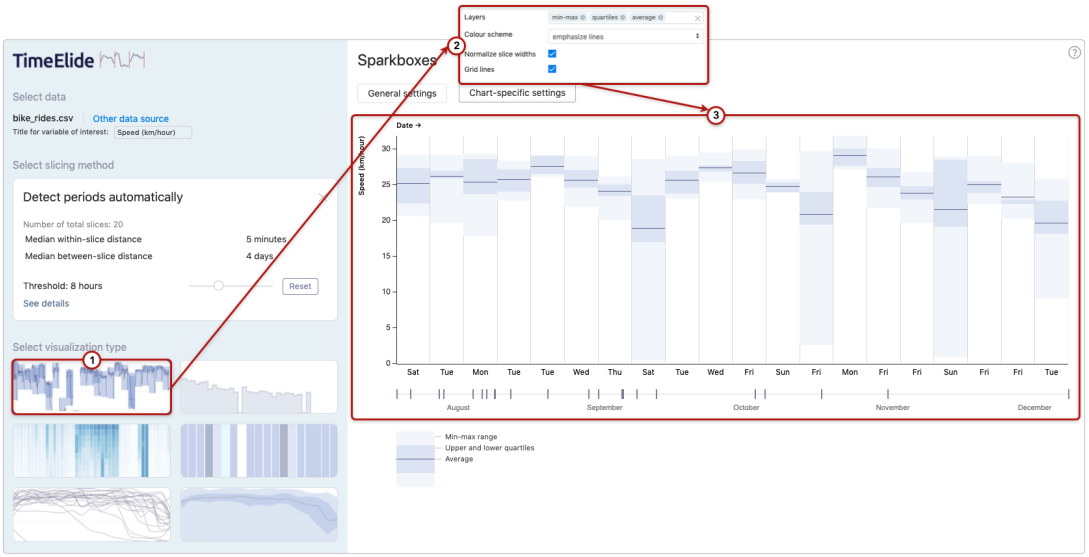


Figure 8. Annotated screenshot of TimeElide: Analyzing the speed during bike rides using box-plot-bars.

SUP-6. TimeElide Interface

We provide supplementary information about available settings and a set of high-resolution screenshots of TimeElide.

SUP-6.1. Global and Chart-Specific Settings

TimeElide supports both global and chart-specific settings, as shown in Table 3. Global settings remain active when the user changes the chart type while chart-specific settings control only the representation of the currently active chart type. The default chart settings are specified in JSON format and can be changed interactively through the user interface.

			Sparkboxes, sliced line charts, box-plot-bars	Stepped area chart	2D heatmap	Heat stripes	Multi-series line chart	Banded multi-series line chart
Global settings	Show global timeline	boolean						
	Show tooltip	boolean						
	Show missing data	boolean						
	Width	number (*px*)						
	Height	number (*px*)						
Chart-specific settings	Layers	multi-select: *min-max range, iqr, median, average*						
	Colour scheme	select: *emphasize boxes/bands, emphasize lines*						
	Line opacity	number (0-1)						
	Show grid lines	boolean						
	Normalize slice widths	select: *normalized durations, absolute duration*						
	Within-slice timing	select: *normalized duration, absolute duration, abs. time*						
	Within-slice aggregation	select: *average, median, minimum, maximum*						
	Bins	number (*integer*)						

Table 3.: Overview of global and chart-specific settings that are supported in TimeElide. Supported, Not supported

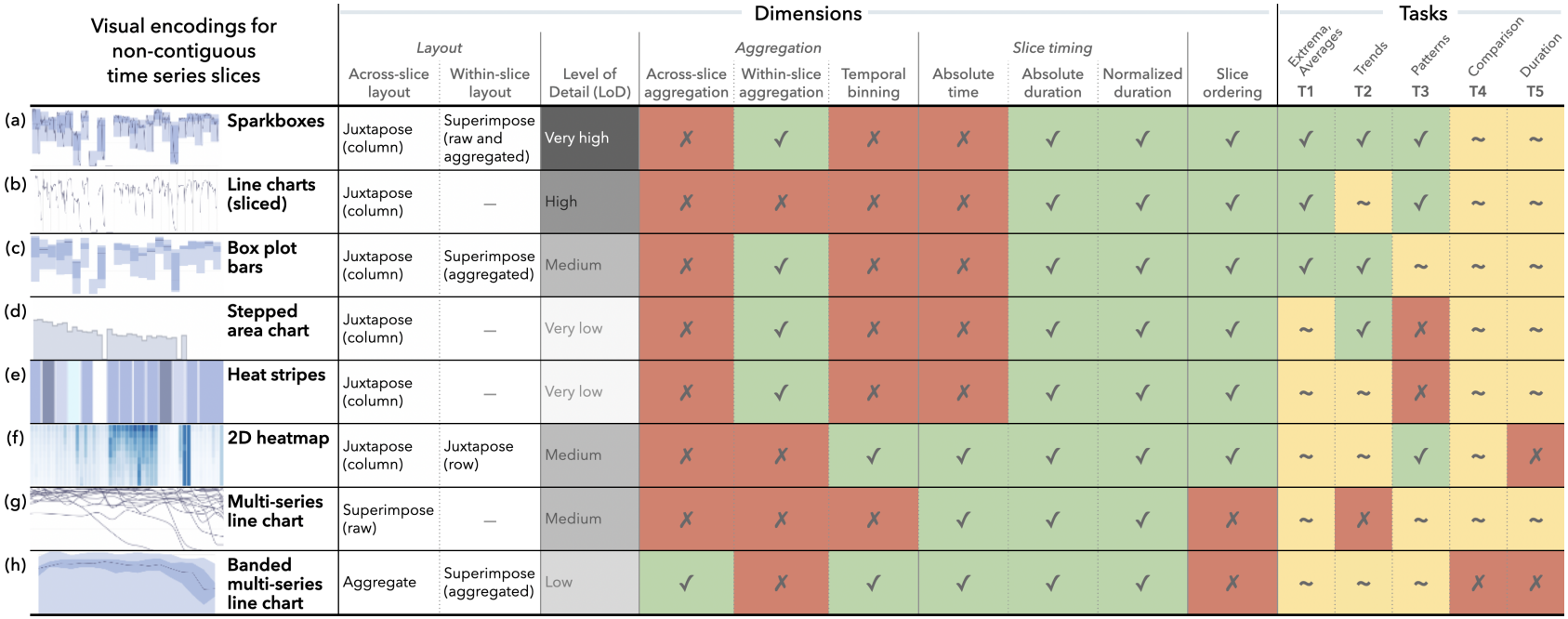
SUP-6.2. Design Space Analysis Rationale

Here, we provide low-level details about the design dimensions and supported tasks. The figure illustrating the visual encoding design space (from the main paper) is included below.

- *The LoD is very high for sparkboxes and only high for line charts.* The aggregated values in sparkboxes are considered as additional information and help to better grasp the gist and overall trends.
- *The LoD is high for line charts (sliced) and only medium for multi-series line charts.* It is easier to compare within-slice patterns in a multi-series line chart but, due to superposition, the information about temporal order of the time slices gets lost.
- *Tasks T1, T2, T3 are only partially supported by the banded multi-series line chart.* It is easy to recognize extrema and averages but it is impossible to link them to individual time slices. The patterns are aggregated across slices and, therefore, analyzing how patterns in specific slices change or deviate is impossible. Due to the across-slice aggregation, it is easy to recognize trends of daily patterns but it is impossible to discern weekly or seasonal trends.
- *Task T4 (comparison) is only partially supported by the presented visual encod-*

ings. Although pairwise comparisons are possible with most visual encodings (Fig 3 a-g), accurate comparisons are challenging when slices are juxtaposed, and in case of superposition, distinguishing the identity of individual lines and their temporal order is difficult.

- *Task T5 (duration) is only partially supported by the presented visual encodings.* Although it is possible to identify slice lengths (durations) in most visual encodings, an accurate analysis is challenging, and especially in case of many slices would require an auxiliary view focused on this specific task.
- *Line charts, stepped area charts, heat stripes, and multi-series line charts do not exhibit a within-slice layout.* In our definition, visual encodings only have a within-slice layout if slices are subdivided into layers or components. The within-slice data points are not aggregated for some of the encodings, such as line charts, but have only a single layer or visual mark per slice.



Visual encodings for non-contiguous time series slices	Dimensions: Layout: Across-slice layout	Layout: Within-slice layout	Level of Detail (LoD)	Aggregation: Across-slice aggregation	Aggregation: Within-slice aggregation	Aggregation: Temporal binning	Slice timing: Absolute time	Slice timing: Absolute duration	Slice timing: Normalized duration	Slice ordering	Tasks: Extrema, Averages T1	Trends T2	Patterns T3	Comparison T4	Duration T5
(a) **Sparkboxes**	Juxtapose (column)	Superimpose (raw and aggregated)	Very high	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓	~	~
(b) **Line charts (sliced)**	Juxtapose (column)	–	High	✗	✗	✗	✗	✓	✓	✓	✓	~	✓	~	~
(c) **Box plot bars**	Juxtapose (column)	Superimpose (aggregated)	Medium	✗	✓	✗	✗	✓	✓	✓	✓	✓	~	~	~
(d) **Stepped area chart**	Juxtapose (column)	–	Very low	✗	✓	✗	✗	✓	✓	✓	~	✓	✗	~	~
(e) **Heat stripes**	Juxtapose (column)	–	Very low	✗	✓	✗	✗	✓	✓	✓	~	~	✗	~	~
(f) **2D heatmap**	Juxtapose (column)	Juxtapose (row)	Medium	✗	✗	✓	✓	✓	✓	✓	~	~	✓	~	✗
(g) **Multi-series line chart**	Superimpose (raw)	–	Medium	✗	✗	✗	✓	✓	✓	✗	~	✗	~	~	~
(h) **Banded multi-series line chart**	Aggregate	Superimpose (aggregated)	Low	✓	✗	✓	✓	✓	✓	✗	~	~	~	✗	✗

Figure 9. Visual encoding design space for non-contiguous time series slices (reproduced from main paper).
Supported, Partially supported, Not supported

SUP-6.3. TimeElide Screenshots

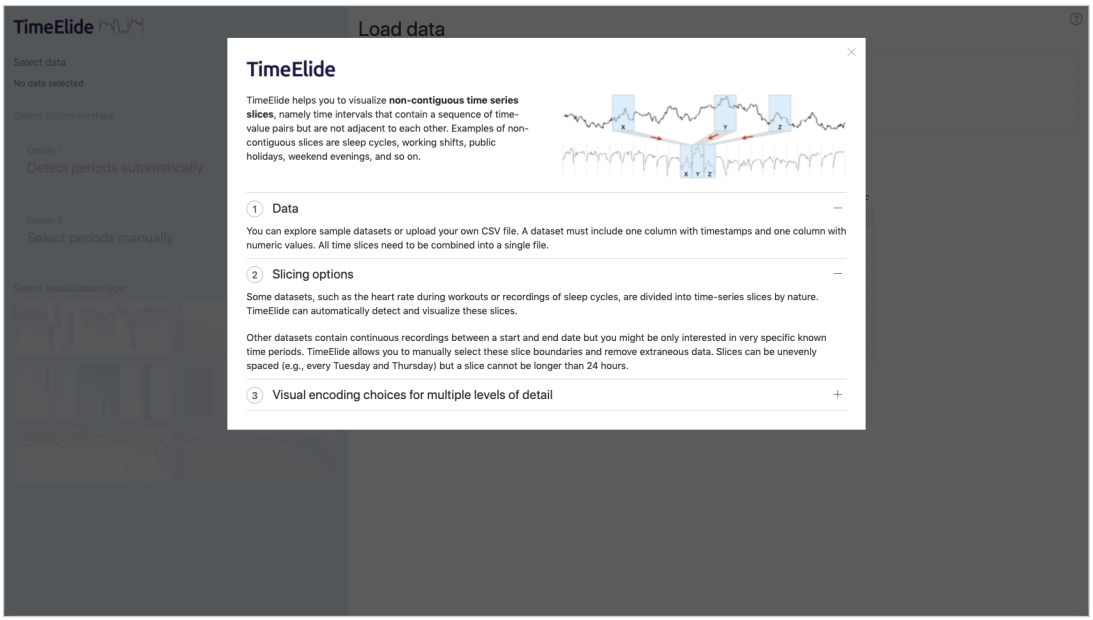


Figure 10. TimeElide welcome modal window: introduce data and slicing options.

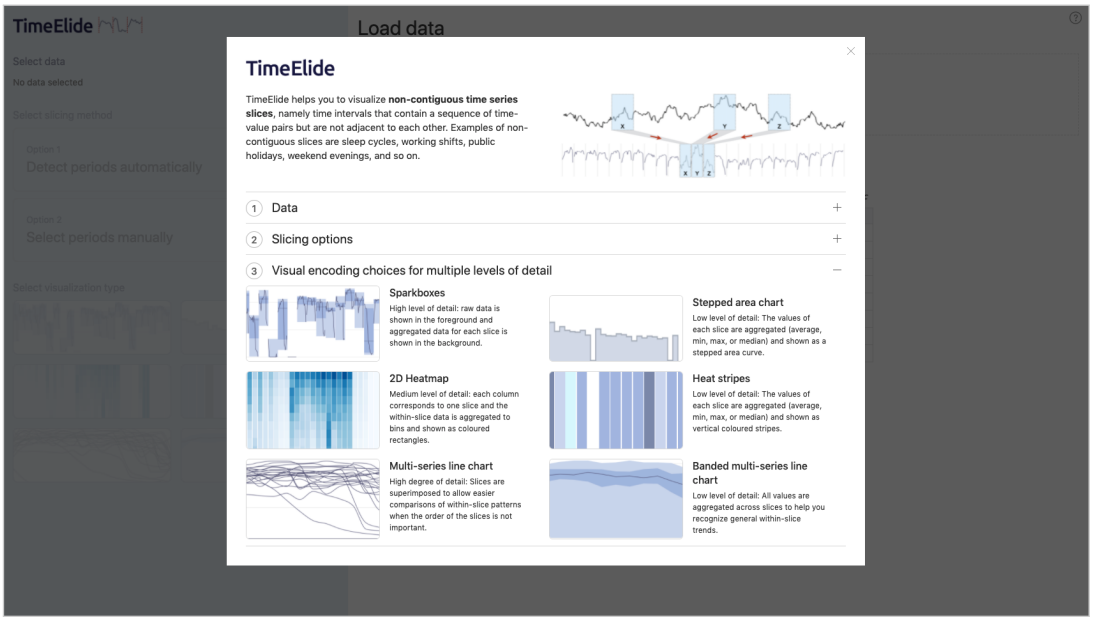


Figure 11. TimeElide welcome modal window: introduce visual encodings.

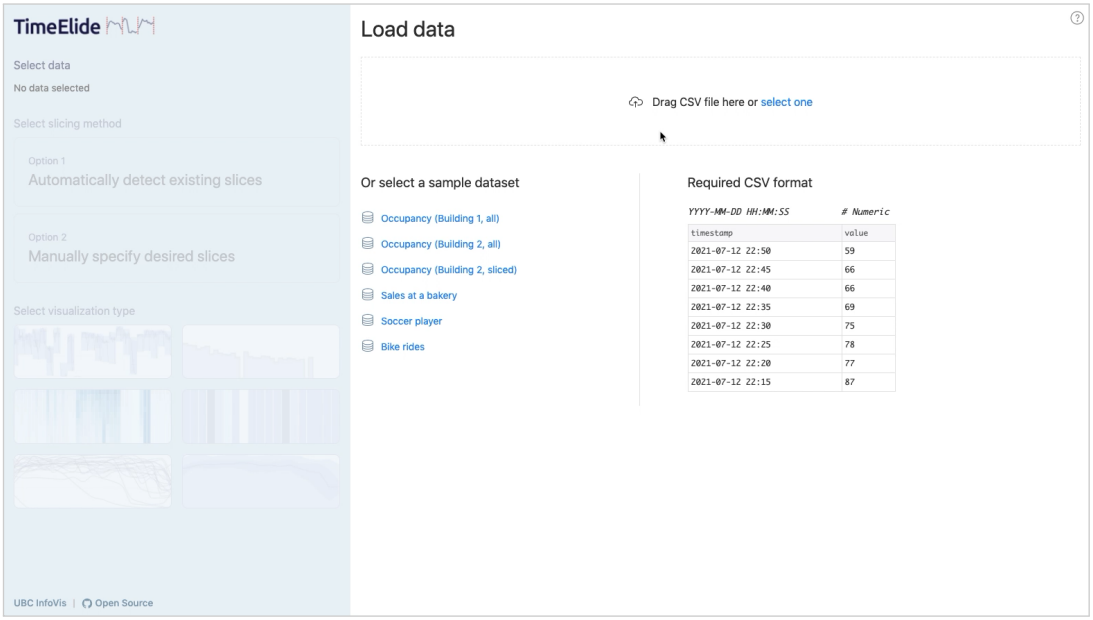


Figure 12. TimeElide data source page.

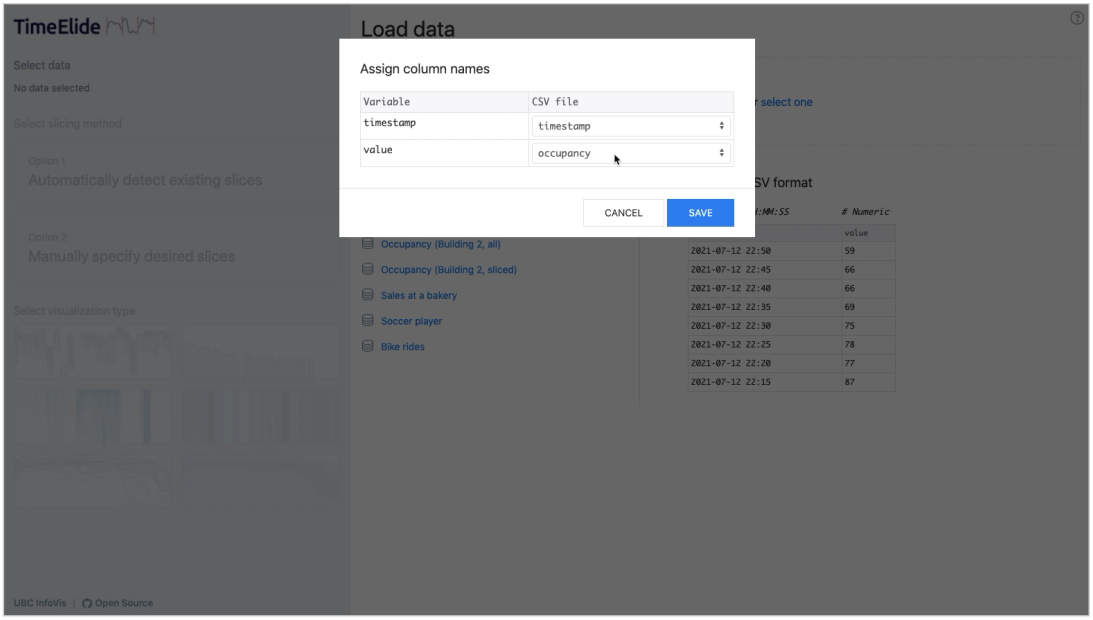


Figure 13. TimeElide data source page: assign columns of uploaded CSV file to *timestamp* and *value* variables.

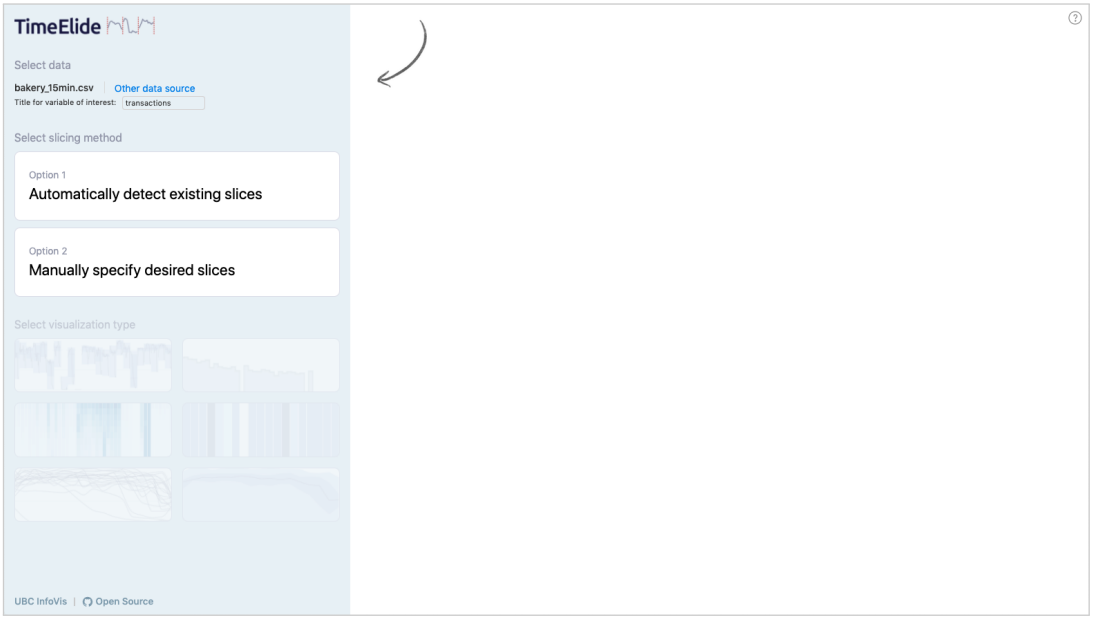


Figure 14. TimeElide slicing options: choose between manual and automatic slicing.

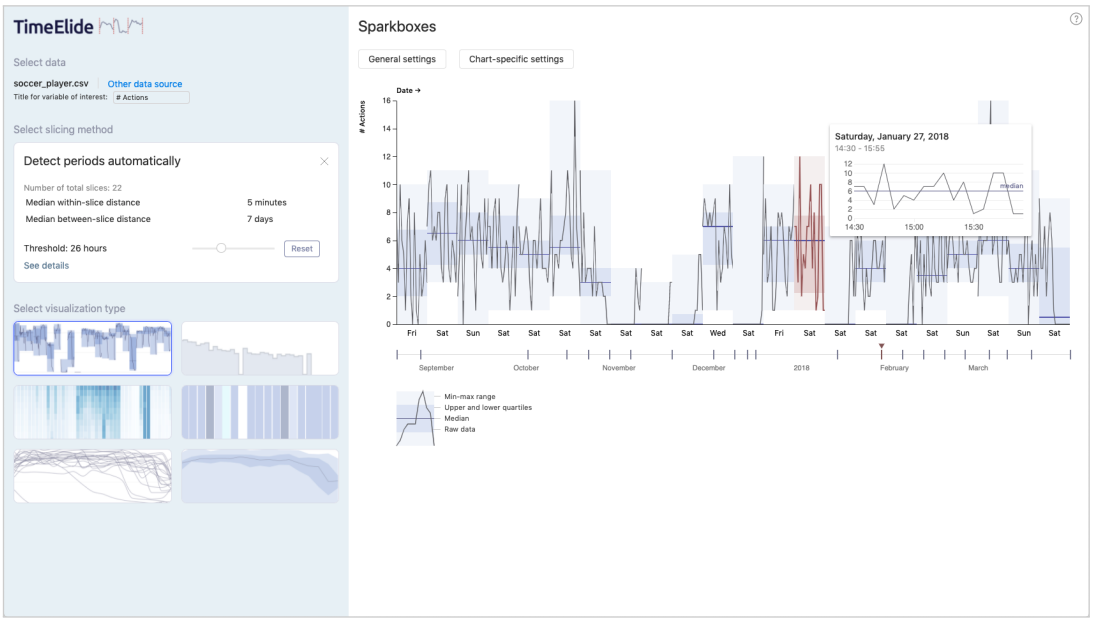


Figure 15. TimeElide sparkboxes and tooltip overlay for one slice: soccer player in-match activity.

Figure 16. TimeElide stepped area chart: average soccer player in-match activity.

Figure 17. TimeElide heat stripes: average soccer player in-match activity.

Figure 18. TimeElide 2d heatmap, absolute time: soccer player in-match activity.

Figure 19. TimeElide 2d heatmap, normalized duration: soccer player in-match activity.

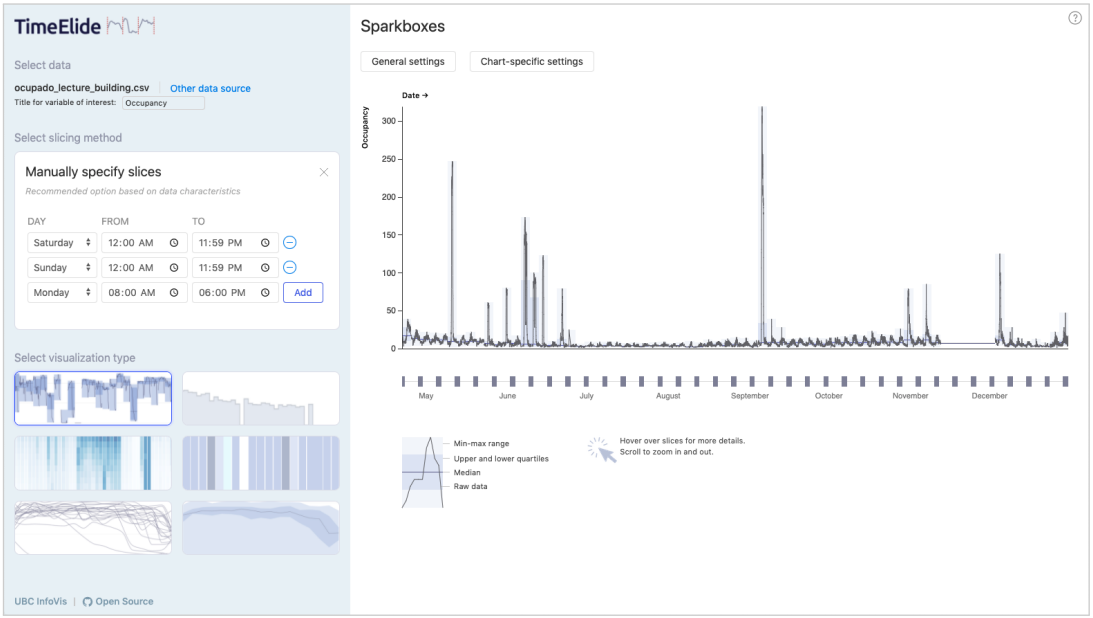


Figure 20. TimeElide sparkboxes: building occupancy on weekends.

Figure 21. TimeElide stepped area chart: average building occupancy on weekends.

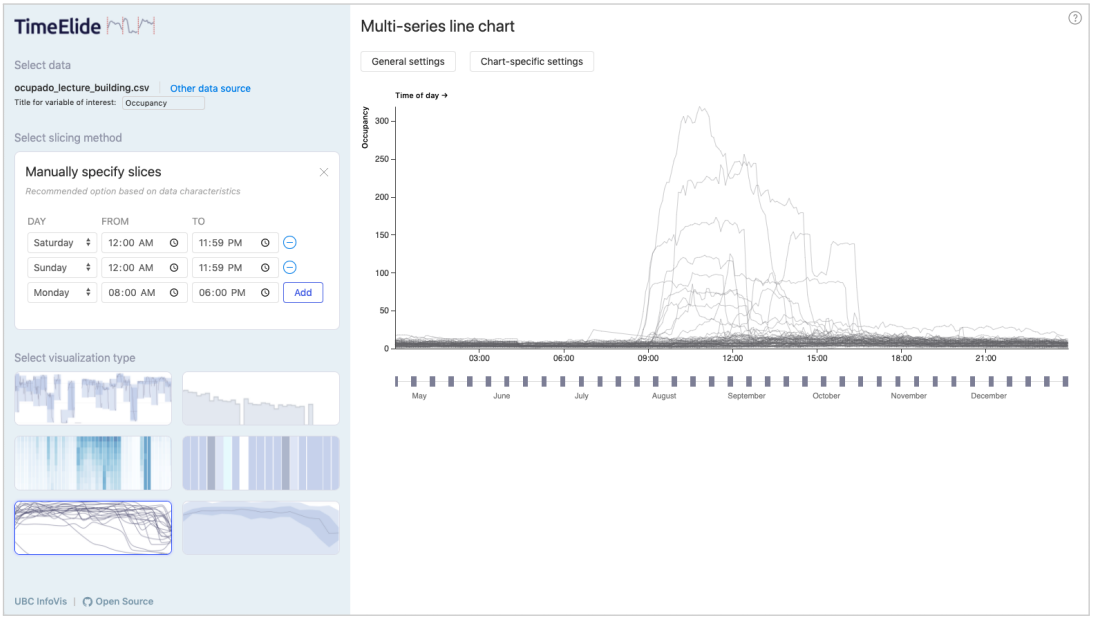


Figure 22. TimeElide multi-series line chart: building occupancy on weekends.

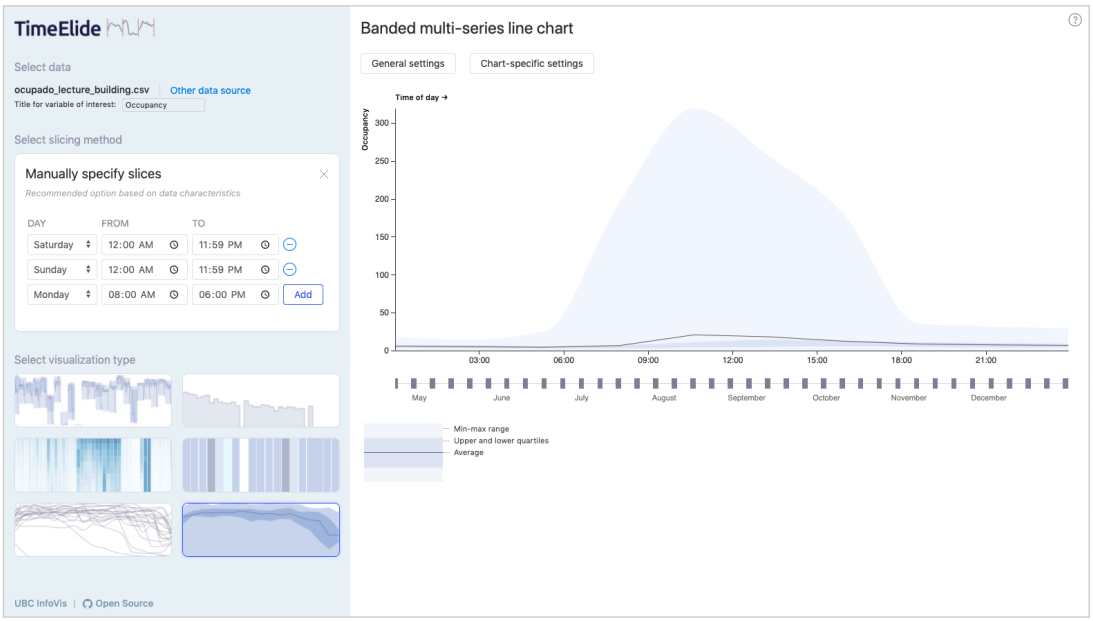


Figure 23. TimeElide banded multi-series line chart: building occupancy on weekends.

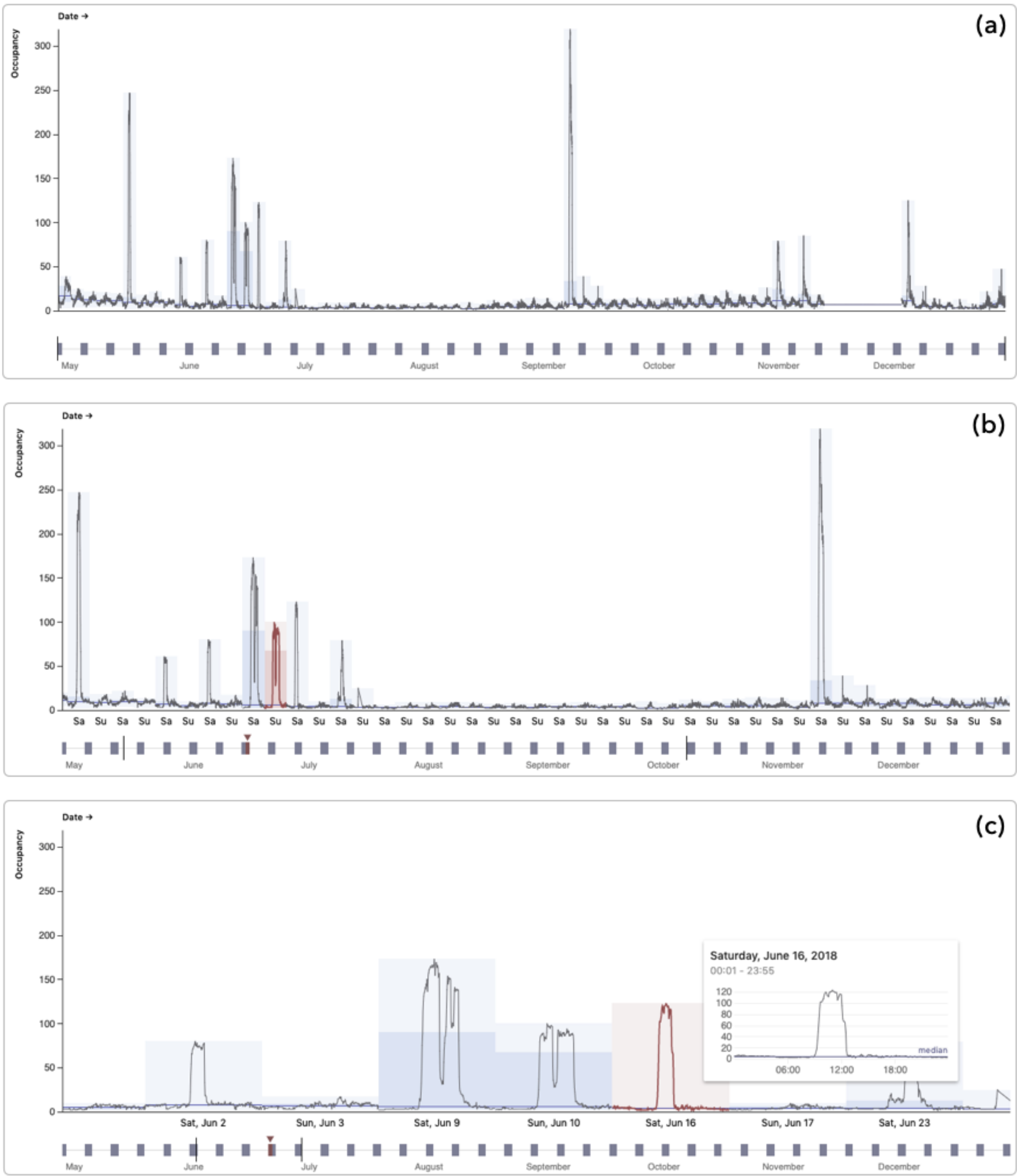


Figure 24. Building occupancy on weekends shown as sparkboxes at three different zoom levels (a) - (c). TimeElide supports interactive panning and zooming for all visual encodings except for the multi-series line chart and the banded multi-series line chart.

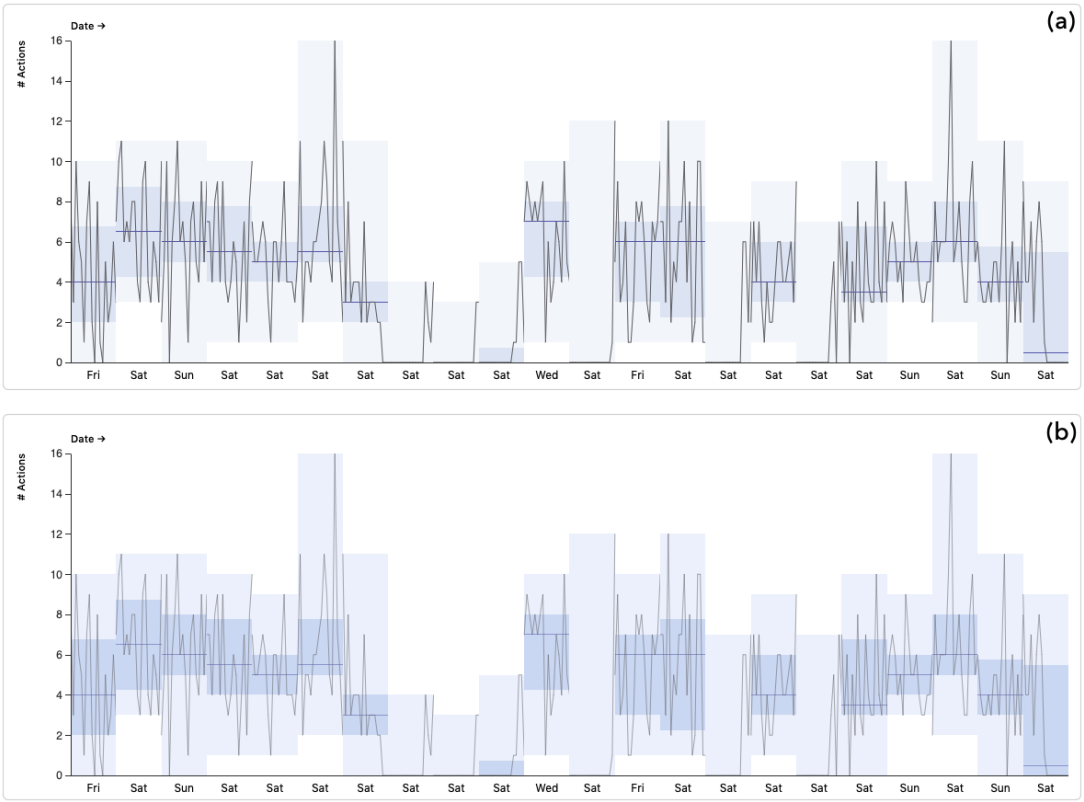


Figure 25. Sparkboxes: different choices for the color palette can shift the focus to selectively emphasize fine or coarse structures in the data in order to support different tasks.