

HCC Line Chart: A Custom Chart for RAWGraphs

Amogh Ratna Manandhar

Group G1a

DAT.C406UF Information Visualisation 3VU SS 2026

Graz University of Technology

29 Jun 2026

Abstract

Line charts are one of the most prevalent chart types in data visualisation. This report first surveys the varieties of line charts and the formatting best practices described by Desbarats in his 2023 book *Practical Charts*, covering non-data elements, axis configuration, and the use of inset charts for large value ranges. RAWGraphs is a free and open-source web-based visualisation tool, a platform for creating line charts without programming. Its mechanism for custom chart extensions is described, enabling chart types and formatting options beyond those available in the standard tool.

The HCC Line Chart is a practical implementation of Desbarats's recommendations for line charts, built as a custom chart for RAWGraphs. The chart applies the recommended best practice as default settings and provides extensive customisation options covering artboard size, chart type, line style, axes, gridlines, and legend.

© Copyright 2026 by the author(s), except as otherwise noted.

This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

Contents

Contents	i
List of Figures	iii
1 Introduction	1
2 Line Charts	3
2.1 Varieties of Line Charts	3
2.2 Recommended Formatting	3
2.2.1 Formatting Non-Data Elements	6
2.2.2 Axes and Scales.	6
2.2.3 Inset Charts	7
3 RAWGraphs	9
4 HCC Line Chart	11
4.1 Architecture	11
4.2 Getting Started	11
4.3 Data Dimensions	12
4.4 Customisation Options	12
4.4.1 Artboard	12
4.4.2 Chart	13
4.4.3 Artistic	13
4.4.4 Axes	13
4.4.5 Grid	14
4.4.6 Legend	14
5 Future Work	15
6 Concluding Remarks	17
Bibliography	19

List of Figures

2.1	Standard Line Chart	4
2.2	Multi-Line Chart	4
2.3	Dual-Axis Chart	4
2.4	Step Chart	5
2.5	Index Chart	5
2.6	Bumps Chart	5
2.7	Cluttered Line Chart	6
2.8	Three Types of Axis Scale	7
2.9	Inset Chart	7
4.1	HCC Bumps Chart.	12

Chapter 1

Introduction

Line charts are among the most widely used chart types in data visualisation. They are well suited for communicating trends, changes over time, and comparisons across multiple series. Despite their apparent simplicity, poor formatting choices can obscure the message a chart is intended to convey, while thoughtful design significantly improves readability and comprehension.

This report surveys the varieties of line charts and the formatting best practices described by Desbarats [2023]. It then introduces RAWGraphs [Mauri et al. 2017], a free and open-source web-based tool for creating charts without programming, including support for custom chart extensions. Finally, it presents the HCC Line Chart [Manandhar 2026a], a custom RAWGraphs extension developed by the author of this report. The extension is built with D3.js [Bostock et al. 2011] and Rough.js [Shihn 2023], and implements the Desbarats formatting recommendations as its default settings, while exposing extensive customisation options.

Chapter 2

Line Charts

A line chart displays quantitative values along a continuous axis, connecting data points with straight line segments to emphasise trends and change over time. The primary reference for best practices used throughout this chapter is Desbarats [2023], which covers the process of choosing chart types, best practices to follow, and bad practices to avoid.

2.1 Varieties of Line Charts

Line charts are not a single, monolithic chart type. Several distinct variants exist, each suited to a different kind of data or comparison. Desbarats [2023] describes the following six types of line chart:

- *Standard Line Chart*: The most common form, plotting a single quantitative series against a time or categorical axis (see Figure 2.1).
- *Multi-Line Chart*: Displays two or more series on the same axes, enabling direct comparison of trends (see Figure 2.2).
- *Dual-Axis Chart*: Uses two separate vertical axes to show with two different equivalent scales or units on the same chart (see Figure 2.3). This variant requires care, as misaligned scales can mislead readers.
- *Step Chart*: Connects data points with horizontal and vertical segments rather than diagonal lines, making it suitable for data that changes instantaneously, such as the price of a product (see Figure 2.4).
- *Index Chart*: Rebases all series to a common starting value (typically zero or one hundred percent), enabling comparison of relative change across series which have very different absolute values (see Figure 2.5).
- *Bumps Chart*: Plots ranked positions over time, with each line representing an entity. Bumps charts are well suited to showing how rankings change across discrete time steps (see Figure 2.6).

2.2 Recommended Formatting

Effective formatting of a line chart involves choices about non-data elements, axis configuration, and the handling of large value ranges. The following subsections summarise the recommendations of Desbarats [2023].

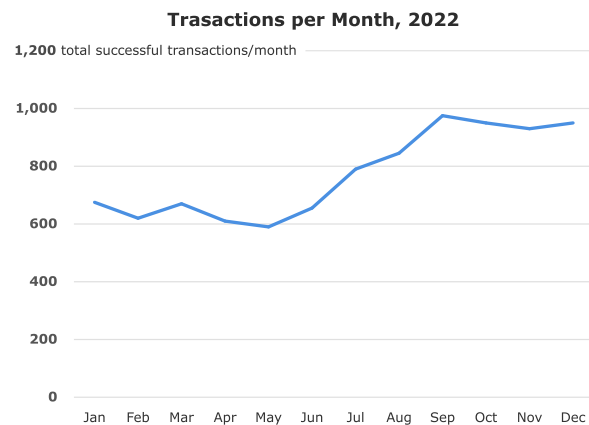


Figure 2.1: Standard Line Chart: A standard line chart showing transactions per months. [Adapted from Desbarats [2023, page 16] by the author.]

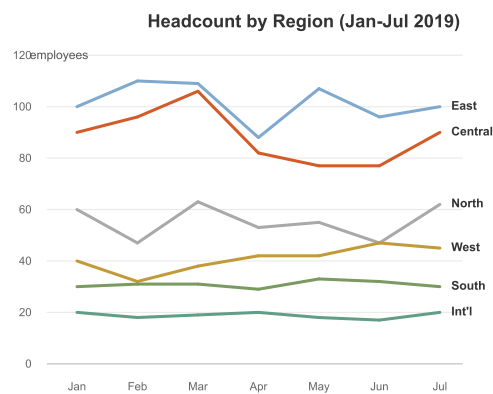


Figure 2.2: Multi-Line Chart: A multi-line chart showing head count by region. [Adapted from Desbarats [2023, page 65] by the author.]

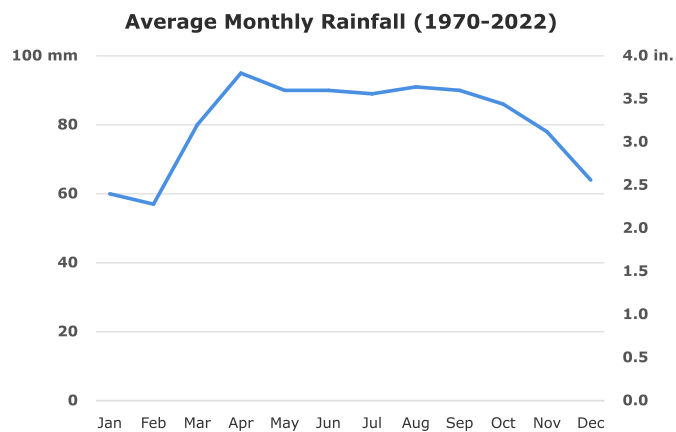


Figure 2.3: Dual-Axis Chart: A Dual-axis chart of average monthly rainfall. [Adapted from Desbarats [2023, page 20] by the author.]

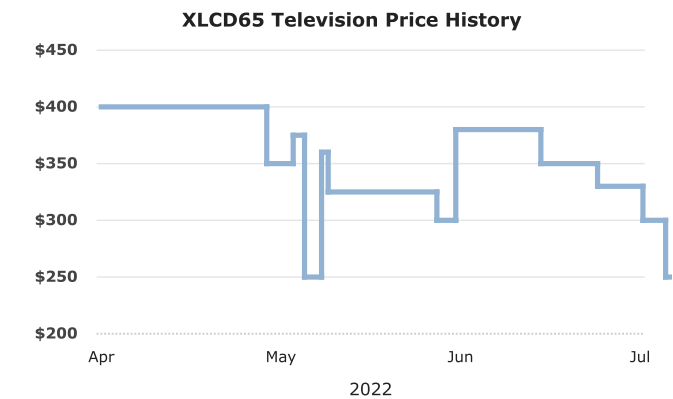


Figure 2.4: Step Chart: A step chart showing the price history of the XLCD65 television. [Adapted from Desbarats [2023, page 98] by the author.]



Figure 2.5: Index Chart: An index chart showing gross sales and customer satisfaction with reference to January in percentage terms. [Adapted from Desbarats [2023, page 19] by the author.]

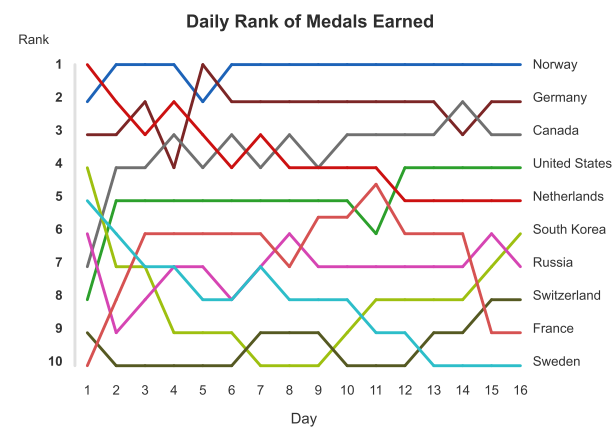


Figure 2.6: Bumps Chart: A bumps chart showing the rank of countries with respect to the number of medals earned. [Adapted from Desbarats [2023, page 93] by the author.]

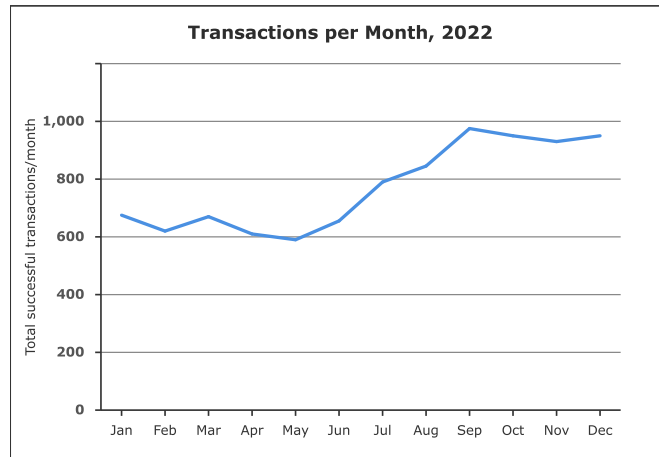


Figure 2.7: A cluttered line chart with non-recommended formatting elements: border, axis lines, tick marks, and heavy gridlines. Compare it to the simple, clean version in Figure 2.1. [Adapted from Desbarats [2023] by the author.]

2.2.1 Formatting Non-Data Elements

Non-data elements are the structural components of a chart that do not directly encode data, such as borders, tick marks, axis lines, and gridlines. Desbarats [2023] recommends minimising these elements, so that the data itself carries the visual weight of the chart:

- *Borders:* Chart borders are not needed and should be omitted.
- *Tick Marks:* Axis tick marks add visual clutter without aiding comprehension and should be removed.
- *Axis Lines:* Axis lines are similarly unnecessary and should be omitted.
- *Gridlines:* Gridlines are useful for helping readers estimate values, but should be styled subtly, so that they do not compete with the data lines.
- *Axis Titles:* Horizontal axis titles are generally easier to read than rotated vertical titles and should be preferred where possible.

Figure 2.7 shows a cluttered line chart, with non-recommended formatting elements. In contrast, compare it to the simple, clean line chart in Figure 2.1, which follows Desbarats' recommendations.

2.2.2 Axes and Scales

Choosing appropriate axis scales is critical to avoiding distortion. Desbarats [2023] provides the following guidance:

- Three types of scale are used on chart axes: categorical, quantitative, and time scales, each with its own appropriate formatting conventions. See Figure 2.8.
- Data should be sorted in a meaningful order, for example chronologically for time scales or by magnitude for categorical comparisons.
- The axis range should extend approximately 10 to 20 % beyond the minimum and maximum data values, so that no data point sits directly on an axis boundary.
- When the axis does not start at zero, the axis is said to be truncated, and this should be clearly indicated to the reader.
- For time scales spanning multiple levels (for example months within years), a multi-level time scale showing both levels is clearer than a single-level scale.

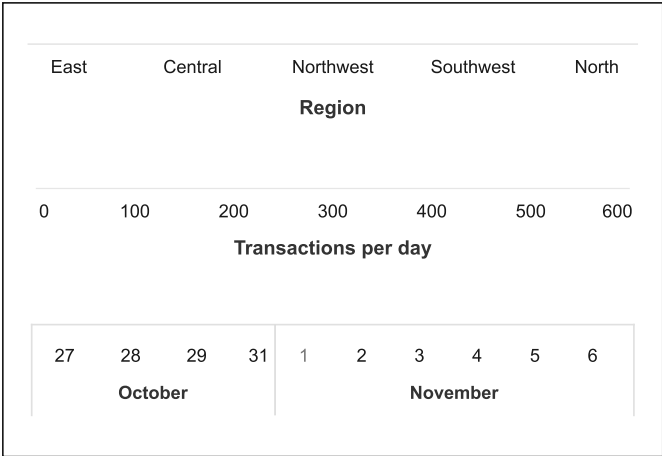


Figure 2.8: The three types of scale used on chart axes: categorical, quantitative, and time. [Adapted from Desbarats [2023, page 7] by the author.]

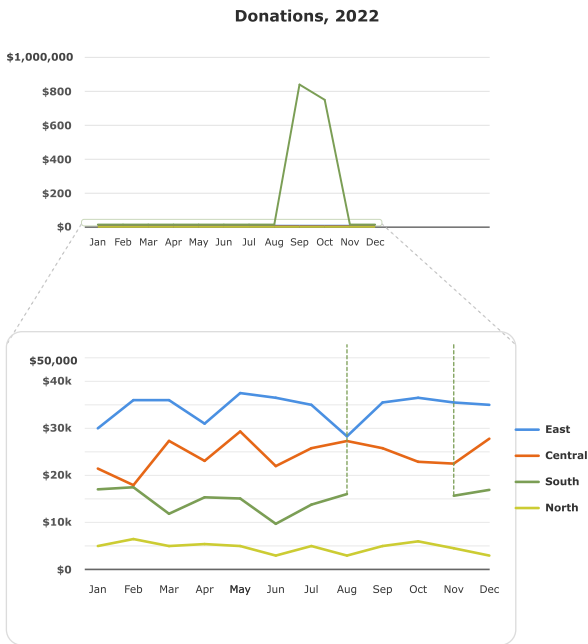


Figure 2.9: An inset chart displays a zoomed in view of any important detail, which otherwise might be overlooked. [Adapted from Desbarats [2023, page 190] by the author.]

2.2.3 Inset Charts

When a dataset contains a large range of values, some trends may appear squished or invisible when displayed on a single shared axis. Desbarats [2023, page 190] recommends using an inset chart in such cases: a small zoomed-in view embedded within the main chart, highlighting a region of interest, as shown in Figure 2.9. This approach avoids distorting the main chart while still communicating fine-grained detail.

Chapter 3

RAWGraphs

RAWGraphs [Mauri et al. 2017] is a free and open-source web-based data visualisation tool designed to make it easy for users without programming skills to create a wide variety of charts. It is available at rawgraphs.io.

RAWGraphs provides a straightforward workflow for creating charts from tabular data:

1. Data is loaded by pasting it directly into the tool, uploading a file, or connecting to an online data source.
2. A chart type is selected from a library of available visualisations or a custom chart can be loaded.
3. The chart is customised using a sidebar of options covering visual properties such as colours, sizes, and labels.
4. The finished chart can be exported in SVG, PNG, or JPG format, or saved as a `.rawgraphs` file for future editing.

RAWGraphs is designed to produce semi-finished charts in a vector graphics format (SVG), which can then be polished and finished in a vector graphics editor such as Adobe Illustrator or Inkscape.

A notable feature of RAWGraphs is its support for custom charts. Custom charts are self-contained pieces of code that can be loaded into the RAWGraphs application on demand, extending its default library of chart types [RAWGraphs 2026]. This mechanism allows developers to implement chart types not included in the standard RAWGraphs release, and to make them available to end users without requiring any modification to the RAWGraphs application itself. Custom charts follow the RAWGraphs chart specification and can expose their own configuration options in the sidebar alongside the built-in ones.

Chapter 4

HCC Line Chart

The HCC Line Chart [Manandhar 2026a] is a custom chart extension for RAWGraphs that implements the line chart formatting recommendations of Desbarats [2023] as its default visual style. It is built using D3.js [Bostock et al. 2011] for chart rendering and Rough.js [Shihn 2023] for the optional hand-drawn artistic effect. The extension is available on GitHub and can be loaded into RAWGraphs as a custom chart. A showcase video is available [Manandhar 2026b].

4.1 Architecture

The extension follows the RAWGraphs custom chart specification and is organised into the following modules:

- `metadata.js`: Defines the chart name, identifier (`hcc.linechart`), categories, and description used to register the chart within RAWGraphs.
- `dimensions.js`: Declares the four data dimensions the chart accepts: X Axis (numbers, dates, or strings), Y Axis (numbers only, aggregated by sum by default), an optional Lines dimension for grouping data into separate lines, and an optional Color dimension for colouring the lines.
- `mapping.js`: Transforms the raw input data into the internal format expected by the renderer. Data is grouped first by the Lines dimension and then by the X Axis dimension using `d3.rollups`. The Y Axis and Color values are aggregated using the user-selected aggregation functions.
- `visualOptions.js`: Declares all user-facing customisation options, their types, default values, and the sidebar group they belong to.
- `render.js`: Contains the main rendering function that builds the SVG output using D3.js, with optional Rough.js rendering for the hand-drawn style.

4.2 Getting Started

The HCC Line Chart can be obtained from its GitHub repository [Manandhar 2026a]. Users can either download the prebuilt custom chart implementation `hcc-line-chart.js` file from the GitHub releases page, or follow the readme instructions to build and run it locally or to make further customisations. Once the file is available, it is loaded into RAWGraphs using the custom chart loading interface.

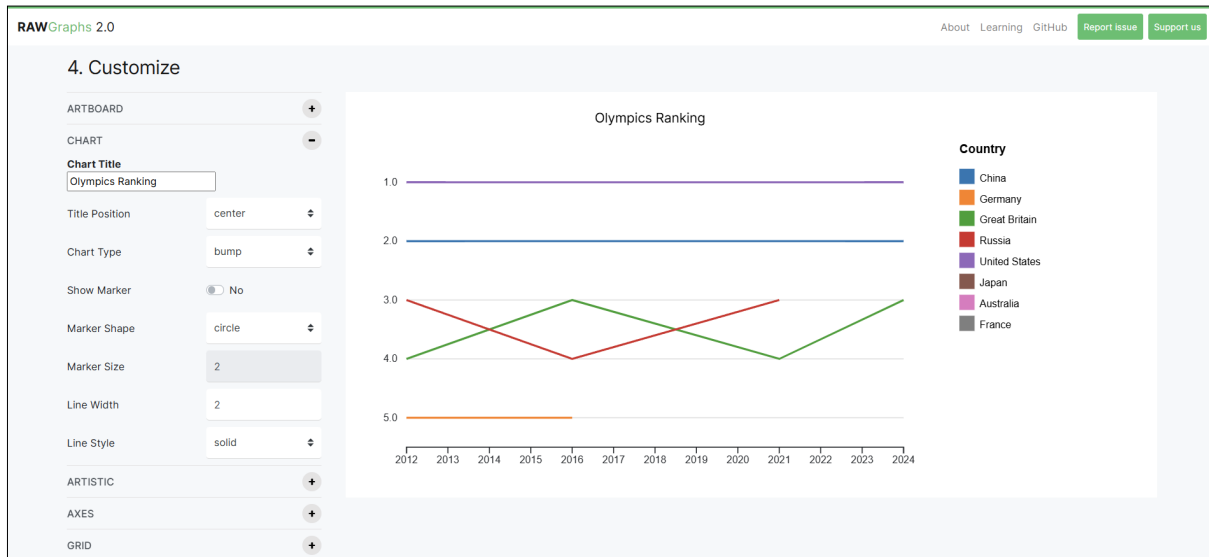


Figure 4.1: Bumps chart created by HCC Line Chart. Some of the Chart options can be seen in the left panel. [Screenshot from RAWGraphs with HCC Line Chart loaded.]

4.3 Data Dimensions

The chart accepts four data dimensions, configured in the RAWGraphs mapping panel:

- **X Axis:** The horizontal dimension. Accepts numbers, dates, and strings. Required.
- **Y Axis:** The vertical quantitative dimension. Accepts numbers only, with aggregation (default: sum). Required.
- **Lines:** An optional dimension used to split the data into separate lines. Accepts numbers, strings, or dates.
- **Color:** An optional dimension used to colour the lines. Accepts numbers, strings, or dates, with aggregation.

4.4 Customisation Options

The HCC Line Chart exposes its customisation options through six groups in the RAWGraphs Customize sidebar: Artboard, Chart, Artistic, Axes, Grid, and Legend. The following subsections describe each group. Figure 4.1 shows a Bumps Chart created with HCC Line Chart custom chart. In the left panel, some of the Chart options can be seen.

4.4.1 Artboard

The Artboard group controls the overall dimensions and spacing of the chart:

- Width and height of the chart.
- Margin on each of the four sides independently (top, bottom, left, right).
- Padding on each of the four sides independently (top, bottom, left, right).

4.4.2 Chart

The Chart group controls the main visual properties of the line chart:

- Chart Title: A title string displayed above the chart.
- Title Position: The title can be aligned to the centre, left, or right of the chart.
- Chart Type: Three rendering modes are available. The line mode connects data points with straight segments. The step mode uses `d3.curveStepAfter` to produce horizontal-then-vertical transitions, suited to data that changes instantaneously. The bump mode replaces the y values with computed ranks at each x position, placing rank 1 at the top of the axis.
- Show Marker: A toggle to display a symbol at each data point.
- Marker Shape: When markers are enabled, the symbol shape can be set to circle, square, triangle, diamond, or star, using the corresponding D3 symbol types.
- Marker Size: Controls the size of the data point symbols.
- Line Width: The stroke width of the data lines in pixels (default: 2).
- Line Style: Lines can be drawn as solid, dashed (4,4 dash pattern), or dotted (2,2 dash pattern).

4.4.3 Artistic

The Artistic group provides a hand-drawn visual effect powered by Rough.js [Shihn 2023]:

- Hand-Drawn Style: A toggle that switches the renderer from standard SVG paths to Rough.js sketchy paths for both lines and markers. Disabled by default.
- Roughness: When the hand-drawn style is enabled, this number controls the degree of visual irregularity applied by Rough.js (default: 1).

4.4.4 Axes

The X and Y axes can each be configured independently:

- Show X/Y Axis: A master toggle for each axis. All axis sub-options are disabled when the axis is hidden.
- Show Axis Line: Toggles the domain line of each axis. The y-axis line is hidden by default, consistent with the recommendations of Desbarats [2023].
- Axis Title: An optional text label for each axis. The x-axis title can be positioned either inline at the end of the axis or horizontally centred below it. When a y-axis title is set and a top gridline is present, the gridline is shortened to avoid overlapping the title text.
- Show Ticks: Toggles the tick marks on each axis. Y-axis ticks are hidden by default.
- Show Labels: Toggles the tick label text on each axis.
- Y Axis Start, End, and Step: Optional overrides for the minimum value, maximum value, and tick interval of the y-axis. When left empty, the domain is computed automatically from the data with a 10 % padding above and below.

4.4.5 Grid

The X and Y gridlines can each be configured independently:

- **Show X/Y grid:** Individual toggles for vertical and horizontal gridlines. Horizontal gridlines (y-axis grid) are shown by default; vertical gridlines (x-axis grid) are hidden by default.
- **Grid Line Colour:** Separate colour pickers for the x and y grid lines (default: `\#e0e0e0` for both).
- **Grid Line Type:** Each grid direction can be set to solid, dashed, or dotted independently.
- **Grid Line Width:** A single width value applied to all grid lines (default: 1 px).

4.4.6 Legend

The Legend group controls the chart legend, which is rendered to the right of the chart area:

- **Show legend:** A toggle to display or hide the legend. Hidden by default.
- **Legend width:** The width reserved for the legend panel in pixels (default: 200). When the legend is shown, this width is added to the total SVG width so that the chart area itself is not reduced.

The legend label is taken from the `Color` dimension mapping if one is set, otherwise from the `Lines` dimension, and falls back to a generic label if neither is mapped.

Chapter 5

Future Work

Several directions remain open for extending the HCC Line Chart [Manandhar 2026a]:

- *Highlighted user-defined line*: When a multi-line chart contains many series, it is often useful for readers to identify a single line of primary interest. A future version of the extension could allow the user to designate one series as the focus line and render it with a thicker stroke, while the remaining lines are rendered at a reduced opacity or with a thinner stroke. This technique, recommended by Desbarats [2023] for directing reader attention, would complement the existing line width option.
- *Hand-drawn style for axes and other elements*: The current implementation of the hand-drawn rendering mode, powered by Rough.js [Shihn 2023], applies the sketchy visual effect only to data lines and markers. Extending the effect to axes, tick marks, and gridlines would produce a more visually consistent hand-drawn appearance across all chart elements. This would require replacing the standard D3.js [Bostock et al. 2011] axis and grid SVG paths with equivalent Rough.js rendered paths.
- *Legend position options*: The current implementation renders the legend exclusively to the right of the chart area (see Section 4.4.6). A future version could extend the legend group with a position option allowing the legend to be placed above the chart (top), below the chart (bottom), to the left, or to the right (the current default). Each position would require different layout logic: a top or bottom legend would be laid out horizontally and its height subtracted from the available chart area, while a left or right legend would use a vertical layout with its width reserved on the corresponding side of the chart.
- *Additional chart types*: Three chart types that are currently absent from the extension would broaden its applicability:
 - *Index Chart*: Rebases all series to a common reference value to enable comparison of relative change
 - *Dual-Axis Chart*: Provides two separate vertical axes on the same chart.
 - *Inset Chart*: Embeds a zoomed-in view of a region of interest within the main chart area.

Each of these types are discussed in Chapter 2.

Chapter 6

Concluding Remarks

This report first surveyed the varieties of line charts and the formatting best practices recommended by Desbarats [2023]. It then described RAWGraphs [Mauri et al. 2017], a free and open-source platform for chart creation that supports user-loadable custom chart extensions.

The body of the report describes the HCC Line Chart [Manandhar 2026a], a custom RAWGraphs extension that puts these recommendations into practice. Its defaults suppress unnecessary non-data elements such as the y-axis line and tick marks, apply subtle horizontal gridlines, and extend the y-axis domain by 10 % beyond the data range. The extension offers four line chart types (line, step, and bump charts, as well as multi-line support via the `Lines` dimension), five marker shapes, solid and sketchy rendering via `Rough.js` [Shihn 2023], independent axis and grid configuration, and a configurable legend.

For practitioners who need to create well-formatted line charts quickly and without writing code, the recommended approach is to use RAWGraphs together with the HCC Line Chart extension. The extension’s defaults already embody the key recommendations of Desbarats [2023], reducing the number of manual adjustments required to produce publication-ready output.

Bibliography

- Bostock, Michael, Vadim Ogievetsky and Jeffrey Heer [2011]. *D3: Data-Driven Documents*. IEEE Transactions on Visualization and Computer Graphics 17.12 (Dec 2011), pages 2301–2309. doi:10.1109/TVCG.2011.185. <https://idl.cs.washington.edu/files/2011-D3-InfoVis.pdf> (cited on pages 1, 11, 15).
- Desbarats, Nicholas P. [2023]. *Practical Charts*. Practical Reporting, 15 Nov 2023. 302 pages. ISBN 1738888304 (cited on pages 1, 3–7, 11, 13, 15, 17).
- Manandhar, Amogh Ratna [2026a]. *HCC Line Chart*. 24 Jun 2026. <https://github.com/AmoghR/hcc-line-chart> (cited on pages 1, 11, 15, 17).
- Manandhar, Amogh Ratna [2026b]. *Showcase Video: HCC Line Chart*. 29 Jun 2026. <https://youtu.be/NwEqMH3GxbA> (cited on page 11).
- Mauri, Michele, Tommaso Elli, Giorgio Caviglia, Giorgio Ubaldi and Matteo Azzi [2017]. *RAWGraphs: A Visualisation Platform to Create Open Outputs*. Proc. 12th Biannual Conference of Italian SIGCHI Chapter (CHIItaly 2017) (Cagliari, Italy). ACM, 18 Sep 2017. doi:10.1145/3125571.3125585 (cited on pages 1, 9, 17).
- RAWGraphs [2026]. *RAWGraphs Custom Charts*. 29 Jun 2026. <https://rawgraphs.io/custom-charts> (cited on page 9).
- Shihn, Preet [2023]. *RoughJS: Create Graphics with a Hand-Drawn, Sketchy, Appearance*. 20 Nov 2023. <https://github.com/rough-stuff/rough> (cited on pages 1, 11, 13, 15, 17).