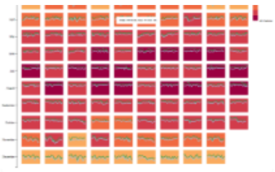

Temperature Daily

	date	max_temperature	min_temperature
1	1997-01-01	25	16
2	1997-01-02	22	18
3	1997-01-03	18	17
4	1997-01-04	20	17
5	1997-01-05	19	15
6	1997-01-06	18	14

You are a data visualization expert. Please complete the following task:

Task: Create a **Matrix View** to visualize the monthly temperature of Hong Kong using the data provided in `temperature_daily.csv`. Focus only on the **last 10 years** of data.

Visualization Requirements:
- **Matrix Structure**:
  - The **x-axis** represents the **year**.
  - The **y-axis** represents the **month**.
  - Each cell corresponds to a specific month of a specific year.

- **Color Encoding**:
  - Use the **background color** of each cell to represent either the **maximum** or the **minimum** temperature of that month.
  - Implement a **mouse click** on any cell to toggle between displaying the maximum and the minimum temperature for all cells.

- **Tooltip**:
  - When hovering over a cell, a tooltip should appear showing the **date** (year and month) and the corresponding **temperature value** (max or min depending on the current mode).

- **In-Cell Mini Line Chart**:
  - Within each cell, embed a **small line chart** that shows the **daily temperature variation** for that month.
  - The x-axis of the mini chart represents the **days of the month**.
  - The y-axis represents the **temperature**.

- **Legend**:
  - Include a legend that explains the mapping between colors and temperature values.

Additional Information:
- The dataset `temperature_daily.csv` will be provided.
- A sample image of the expected output will also be supplied.

Grading Criteria:
- **Correctness (50%)**: All specified requirements must be satisfied.
- **Readability (30%)**: Code should be human-readable – use intuitive variable/function names and include sufficient comments.
- **Maintainabili[illegible]20%)**: Code should be modular and easy to modify.

I created a complete, **interactive Matrix View** as a single HTML file that meets your requirements (last 10 years, year×month grid, background color toggles max/min on click, tooltip, in-cell mini line charts, legend).

Download the HTML visualization

How to run it (locally)

1. Put **both files in the same folder**:
   - `hk_temperature_matrix.html`
   - `temperature_daily.csv`
2. Start a local server in that folder (recommended to avoid browser CSV loading restrictions):

```bash
python -m http.server 8000
```

3. Open in your browser:
   - `http://localhost:8000/hk_temperature_matrix.html`

What's implemented (mapped to your rubric)

- **Matrix structure:** x=Year (last 10 years), y=Month (Jan–Dec), each cell = (year, month)
- **Color encoding:** cell background shows **monthly max** or **monthly min**; click any cell toggles mode globally
- **Tooltip:** hover shows `YYYY-MM` and the current mode's aggregated value
- **Mini line chart in each cell:** daily **max (green)** and daily **min (blue)** temperature lines for that month
- **Legend:** updates with the current mode and color scale

If you want the mini chart to show **only one line** (max *or* min depending on mode) instead of two, tell me and I'll adjust the code (it's a small change).

14:48

no,don't just creat the html, you can give me the code, and i'll run on my VS code

Got it — here's the **full, runnable code** (single HTML file) you can paste into VS Code. It loads `temperature_daily.csv` (must be in the same folder), renders the **year×month matrix for the last 10 years**, supports **click-to-toggle max/min**, **tooltip**, **mini line charts inside each cell**, and an updating **legend**.

> **How to run:** put this file next to `temperature_daily.csv`, then run a local server (recommended):

```bash
python -m http.server 8000
```

Open: `http://localhost:8000/hk_temperature_matrix.html`

```
      for (const [key, days] of groups.entries()) {
        days.sort((a, b) => a.day - b.day);
        const year = +key.slice(0, 4);
        const month = +key.slice(5, 7);
        const { monthlyMax, monthlyMin } = computeMonthlyAggregates(days);
        monthlyMap.set(key, { year, month, days, monthlyMax, monthlyMin });
      }

      drawMatrix();
    }).catch(err => {
      console.error(err);
      d3.select("#chart").append("p")
        .style("color", "crimson")
        .text("Failed to load temperature_daily.csv. Put it next to this HTML and run a local serv
    });
  </script>
</body>
</html>
```

Notes (so you don't get stuck)

- This expects CSV columns named `date`, `max_temperature`, `min_temperature`.
- If your CSV uses different headers, tell me the column names (or paste the first few lines) and I'll adapt the parser.
- If you open the HTML by double-clicking (file://), many browsers block CSV loading — that's why I recommend running a local server.

If you want the **background color scale to be shared between max and min** (so toggling doesn't change legend range), I can modify that too.

14:51

please code using python

已思考 2m 49s ›

Below is a **pure Python** solution using **Bokeh** (no hand-written HTML/JS beyond Bokeh's built-in `CustomJS`). It generates an interactive **Year×Month matrix** for **Hong Kong daily temperature** from `temperature_daily.csv`, **limited to the last 10 years** of the dataset.

✅ Meets your requirements:

- x-axis = **Year**, y-axis = **Month**
- cell background color = **monthly max OR monthly min** (toggle by **clicking any cell**)
- tooltip = `YYYY-MM` + current value (max/min)
- each cell contains **mini line charts** for daily variation (max line + min line)
- legend shows color→temperature mapping; title updates with mode

---

Install dependencies

bash 复制代码

```bash
pip install pandas numpy bokeh matplotlib
```

```
            const data = source.data;
            const fromField = (next === 'max') ? 'monthly_max' : 'monthly_min';

            // Replace the 'value' column so colors + tooltip update
            data['value'] = data[fromField].slice();

            badge.text = `<b>Mode:</b> Monthly ${next === 'max' ? 'Max' : 'Min'} &nbsp; (Click any
            colorbar.title = `Monthly ${next === 'max' ? 'Max' : 'Min'} (°C)`;

            // Clear selection so clicking the same cell again still triggers the callback
            source.selected.indices = [];

            source.change.emit();
            state.change.emit();
        """
    )
    cell_source.selected.js_on_change("indices", toggle_js)

    return column(badge, p)


def main():
    df = load_last_10_years(CSV_PATH)

    layout = build_plot(df)

    output_file(OUTPUT_HTML, title="Hong Kong Monthly Temperature Matrix (Last 10 Years)")
    save(layout)
    print(f"Saved: {OUTPUT_HTML}")
    print("Open it in your browser. (No server needed; it’s a standalone HTML.)")


if __name__ == "__main__":
    main()
```

Notes / quick tweaks

- If you want **only one mini line** per cell (max *or* min depending on mode), say so and I'll adjust the callback to show/hide the corresponding `multi_line`.
- Right now the **color scale is global** across both modes (so the legend stays stable). If you want the legend range to **recompute** separately for max vs min, I can modify the JS to update `mapper.low/high` too.

15:07

there is an error when running the last cell

已思考 24s ›

Yep — the error is coming from **Bokeh version differences**.

In Bokeh **3.x**, `Div` no longer accepts `style=...`; it uses `styles=` (plural). That's why you got:

> `AttributeError: unexpected attribute 'style' to Div`

You also saw a Matplotlib deprecation warning for `cm.get_cmap` — not fatal, but we can clean it up too.