1. Complex Chart Validation in Cypress

Charts (Bar, Line, Pie, etc.) are usually rendered using **SVG** or **canvas**, so you can't just assert plain text like normal HTML.
With Cypress, you typically validate charts by:

- Checking **data bindings** (values, labels) instead of pixels.
- Validating **DOM structure** of SVG elements (e.g., number of bars, paths, circles).
- Asserting **attributes** like `height`, `width`, `d`, `x`, `y`, `fill`, etc.
- Optionally combining with **API checks** (verify chart data from API, then ensure chart reflects it).

Goal: confirm that **the right data is shown visually**, not do pixel-perfect visual testing.

---

2. Capturing and Validating Graphical Elements

For SVG-based charts:

- Use selectors like `cy.get('svg')`, `cy.get('svg rect')`, `cy.get('svg path')`.
- Validate:
    - **Element count** (e.g. number of bars = number of data points)
    - **Attributes** (e.g. bar heights increase with value, correct colors, etc.)

Example:

```
cy.get('svg rect.bar').should('have.length', 5)
cy.get('svg rect.bar').first().invoke('attr', 'height')
  .then(h => expect(Number(h)).to.be.greaterThan(0))
```

For canvas-based charts, direct DOM is limited, so you often:

- Validate **data source** (API / config passed to chart lib)

- Or use the chart library's **global object** (like `window.myChart.data`) if exposed.

---

3. Using Cypress Commands to Verify SVGs & Canvas Charts

For SVG charts:

- `cy.get()` – grab SVG elements (bars, lines, points).
- `.should()` / `.then()` – assert attributes and counts.
- `.invoke('attr', '...')` – read SVG attributes for validation.

Example:

```
cy.get('svg path.line')
  .should('exist')
  .and($path => {
    const d = $path.attr('d')
    expect(d).to.contain('M') // basic path check
  })
```

For canvas charts:

- Prefer **data-level assertions**:
    - Intercept API: `cy.intercept('/chart-data')` and verify response.
    - Access chart instance if app exposes it:

```
cy.window().its('myChart.data.datasets[0].data')
  .should('deep.equal', [10, 20, 30])
```

Summary:

- **SVG charts** → DOM + attributes-based assertions.
- **Canvas charts** → data + chart-object-based assertions.