

# Ways to Display Analytics Results Dynamically in a Web Page

## 1. JavaScript Chart Libraries

- Chart.js (simple, good for dashboards, responsive)

Example:

```
<canvas id="myChart"></canvas>
<script>
const ctx = document.getElementById('myChart');
new Chart(ctx, {
  type: 'bar',
  data: {
    labels: ['A', 'B', 'C'],
    datasets: [{ label: 'Values', data: [10, 20, 30] }]
  }
});
</script>
```

- D3.js (more customizable, but steeper learning curve)
- Plotly.js (good for interactive charts, zooming, tooltips)
- ECharts (used a lot in enterprise dashboards, feature-rich)
- Google Charts (easy integration, supports pie, bar, maps, etc.)

## 2. Embedding BI Tools

If you're already using a BI or analytics tool, many support embedding:

- Power BI Embedded
- Tableau Embedded Analytics
- Looker (Google Cloud)

These can be embedded in <iframe>s or via JS SDKs.

## 3. Server-Side Rendering → Client

If your analytics is computed server-side (Python, R, etc.), you can:

- Generate chart images (e.g., with Matplotlib, Seaborn, ggplot2) and serve them as <img> tags.
- Export results as JSON and render with a client-side library like Chart.js or D3.js.
- Use Flask/Django + Bokeh/Plotly Dash to serve interactive dashboards.

## 4. Real-Time / Streaming Analytics

For dashboards that need to update live:

- Use WebSockets or Server-Sent Events (SSE) to push data updates.
- Pair with a JS charting library that can update data dynamically.
- Example: A stock price chart updating every second with Plotly.