**1. Use a Contributed Module**

**A. Charts Module**

The **Charts** module provides easy chart creation using libraries like Chart.js, Google Charts, and Highcharts.

**Features:**

* Supports various chart types (bar, line, pie, area, etc.)
* Works with Views
* Can use custom data sets (via YAML or JSON)
* Compatible with Layout Builder

**Installation:**

bash

CopyEdit

composer require drupal/charts

drush en charts charts\_library chartjs

Enable any library you prefer: chartjs, google\_charts, highcharts (requires licensing for Highcharts in production)

**Example Usage:**

1. **Create a Chart Block**
   * Go to Structure → Block layout → Place block
   * Select "Charts: Chart Block"
   * Configure type, data source (static, view, or code), chart type, etc.
2. **Using with Views**
   * Create a view
   * Select format: *Chart*
   * Choose chart type and fields

**2. Programmatic Chart Creation**

If you want to **embed charts programmatically**, you can use libraries directly in custom modules or themes.

**A. Example: Chart.js in a Custom Module**

**1. Include Chart.js**

You can add it via a library:

yaml

CopyEdit

# mymodule.libraries.yml

chartjs:

version: 3.x

js:

https://cdn.jsdelivr.net/npm/chart.js: { type: external, minified: true }

dependencies:

- core/jquery

**2. Attach it in your controller or block**

php

CopyEdit

public function build() {

$build = [];

$build['#attached']['library'][] = 'mymodule/chartjs';

$build['chart'] = [

'#markup' => '<canvas id="myChart" width="400" height="200"></canvas>

<script>

const ctx = document.getElementById("myChart").getContext("2d");

new Chart(ctx, {

type: "bar",

data: {

labels: ["Jan", "Feb", "Mar"],

datasets: [{

label: "Sales",

data: [12, 19, 3],

backgroundColor: "rgba(75, 192, 192, 0.2)",

borderColor: "rgba(75, 192, 192, 1)",

borderWidth: 1

}]

},

options: {

scales: {

y: { beginAtZero: true }

}

}

});

</script>',

];

return $build;

}

**3. Use External APIs or Libraries**

You can also fetch chart data from REST APIs or external services like Google Sheets and display them using embedded JavaScript charts (e.g., Chart.js or D3.js).

**Permissions & Caching**

* Ensure anonymous users have permission to view charts (especially if used in Views).
* Use Drupal cache tags and contexts to manage chart data freshness and performance.