
Introduction: Blazor Charts with Chart.js

BlazorExpress.ChartJS is a **Blazor wrapper around Chart.js**, enabling you to render rich, interactive charts in Blazor applications **without writing JavaScript manually**.

Why use BlazorExpress.ChartJS?

- Native **Blazor component model**
- Strongly-typed configuration (C# instead of JS)
- Works with **Blazor WebAssembly** and **Blazor Server**
- Supports all major **Chart.js chart types**
- Easy data binding and dynamic updates

Blazor Hosting Models (NET 8)

1. Blazor WebAssembly (.NET 8)

Execution: Runs entirely in the browser (via WebAssembly)
Best for:

- Client-side dashboards
- Offline/PWA scenarios
- Public analytics portals

Chart behavior:

- Chart rendering + data processing in the browser
- Data fetched via HTTP APIs

2. Blazor WebApp (.NET 8) – Server

Execution: UI logic runs on the server, UI updates via SignalR
Best for:

- Internal dashboards
- Secure data visualization
- Heavy server-side processing

Chart behavior:

- ChartJS rendering still happens in browser
- Configuration & data come from server

Important: **BlazorExpress.ChartJS works identically in both models**, only the hosting differs.

Installation & Basic Setup

dotnet add package BlazorExpress.ChartJS

Add the Chart.js script in index.html (WASM) or _Host.cshtml (Server):

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

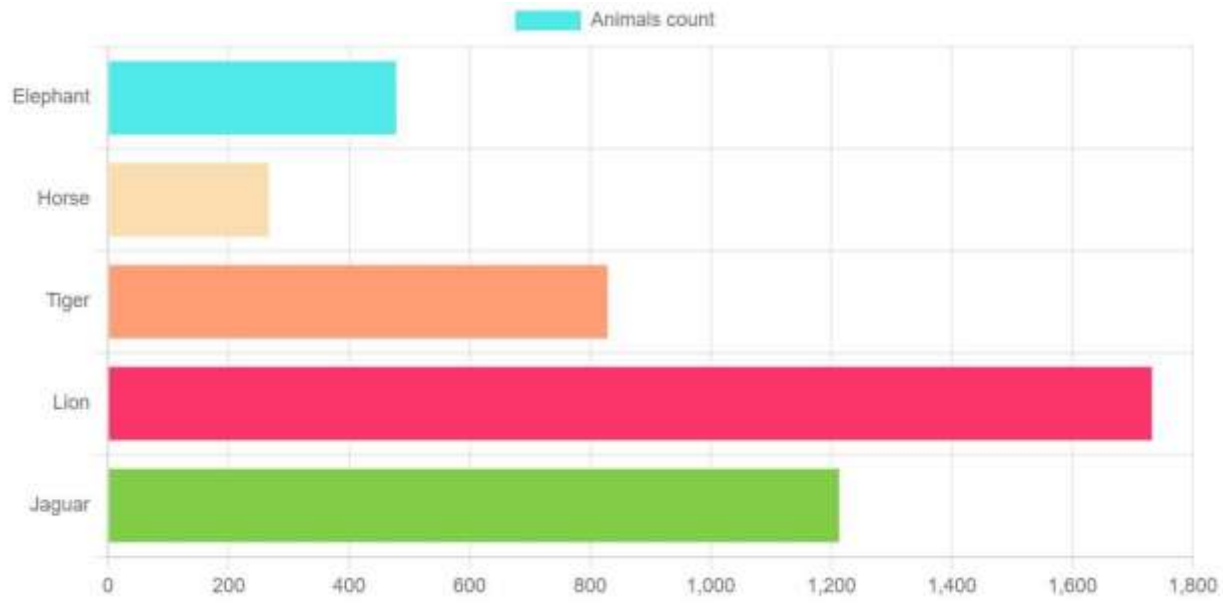
Basic Chart Component Example (Shared)

```
@using BlazorExpress.ChartJS

<Chart Config="@_config" />

@code {
   private ChartConfig _config = new ChartConfig
   {
      Type = ChartType.Bar,
      Data = new ChartData
      {
         Labels = new[] { "Jan", "Feb", "Mar", "Apr" },
         Datasets = new[]
         {
            new ChartDataset
            {
```

```
                Label = "Sales",
                Data = new double[] { 120, 190, 300, 250 },
                BackgroundColor = new[] { "blue" }
            }
        }
    }
};
}
```

Chart Types (Concept + Practical Use)

Bar Chart

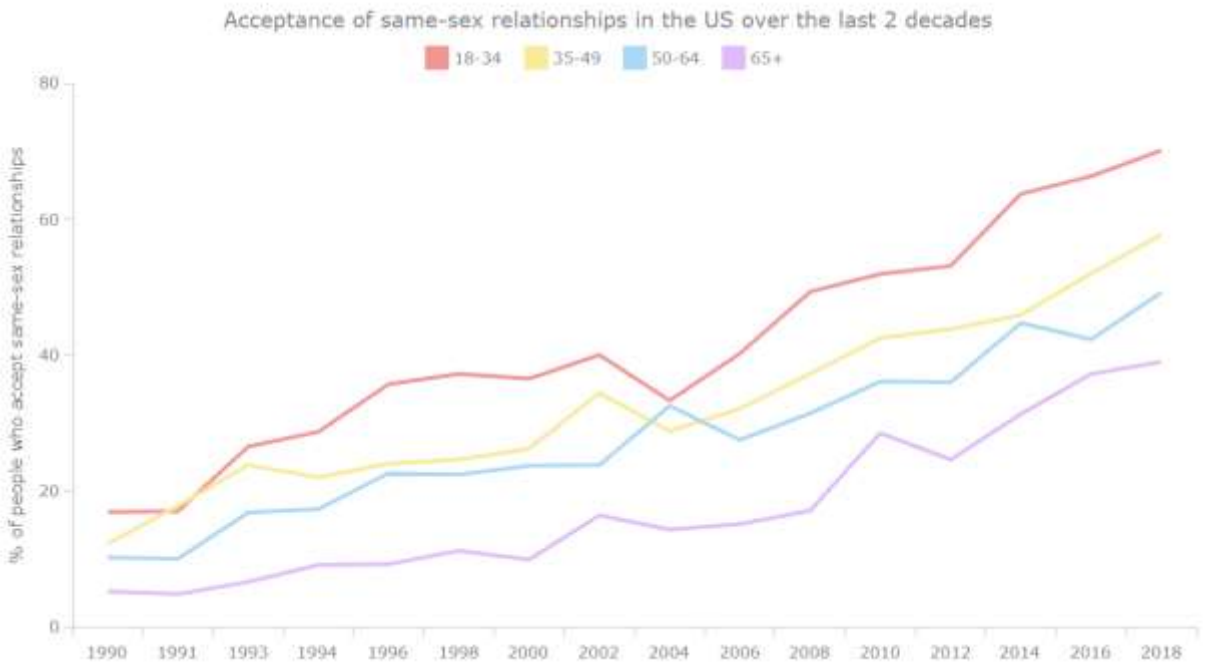


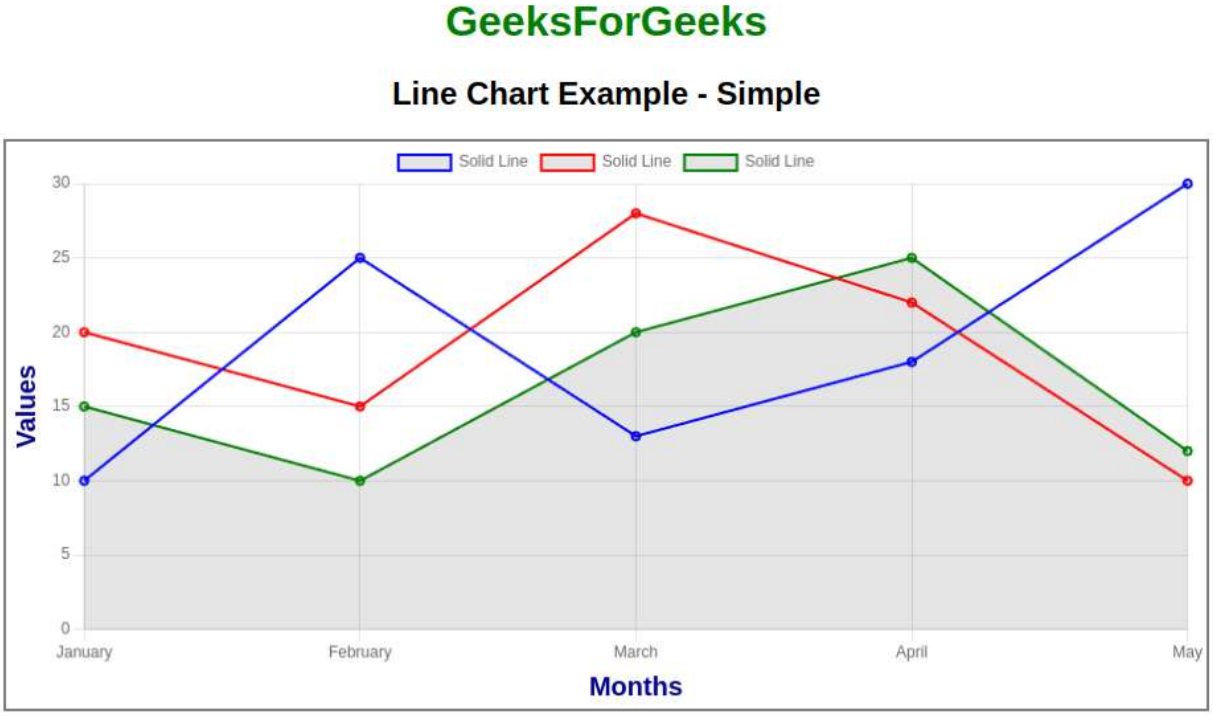
Use cases

- Monthly sales
- Comparison between categories

Type = ChartType.Bar

Line Chart



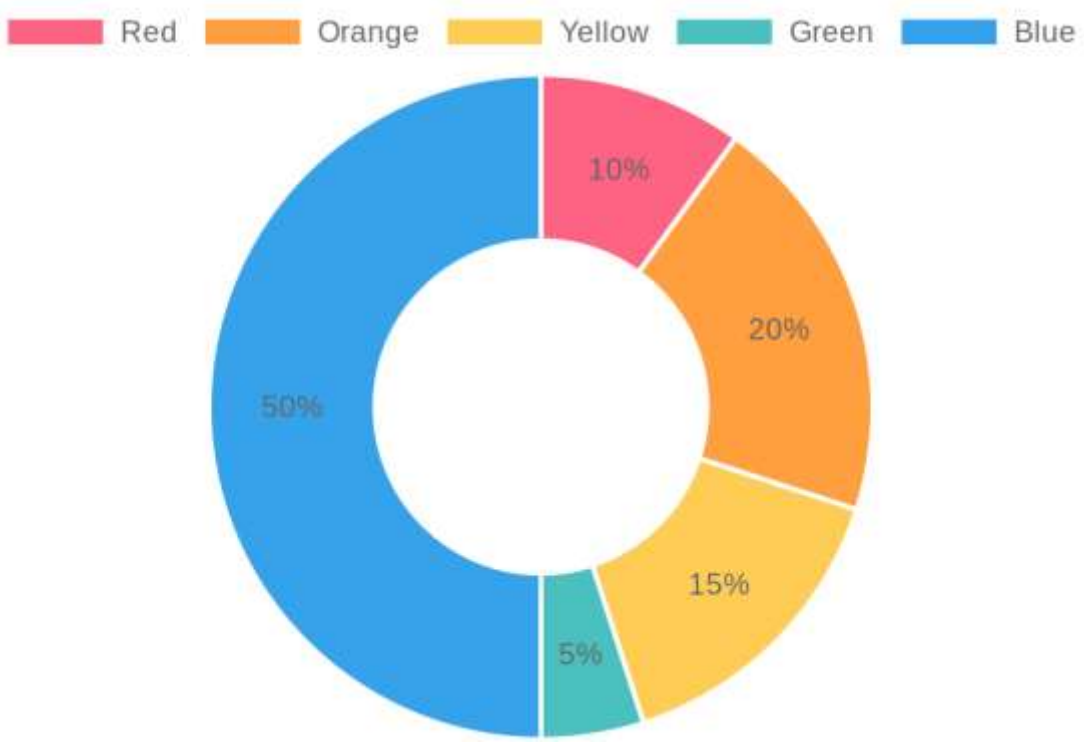


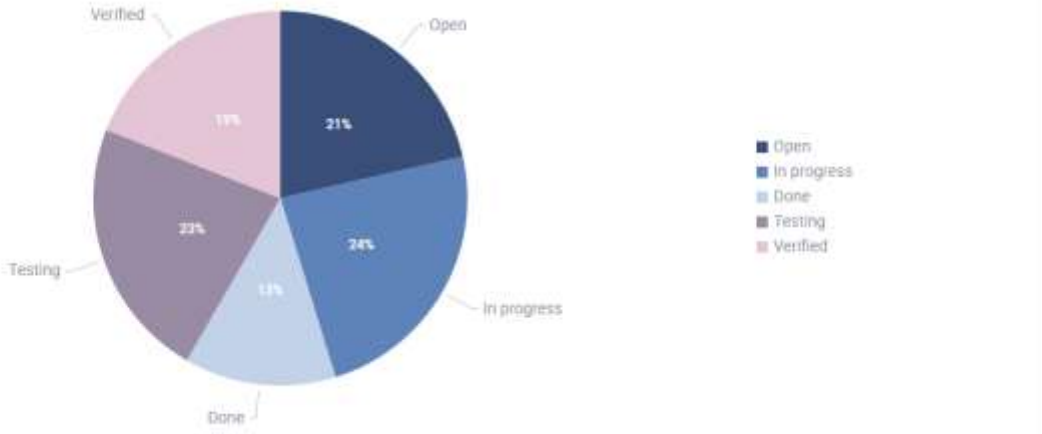
Use cases

- Trends over time
- Performance metrics

Type = ChartType.Line

Pie Chart



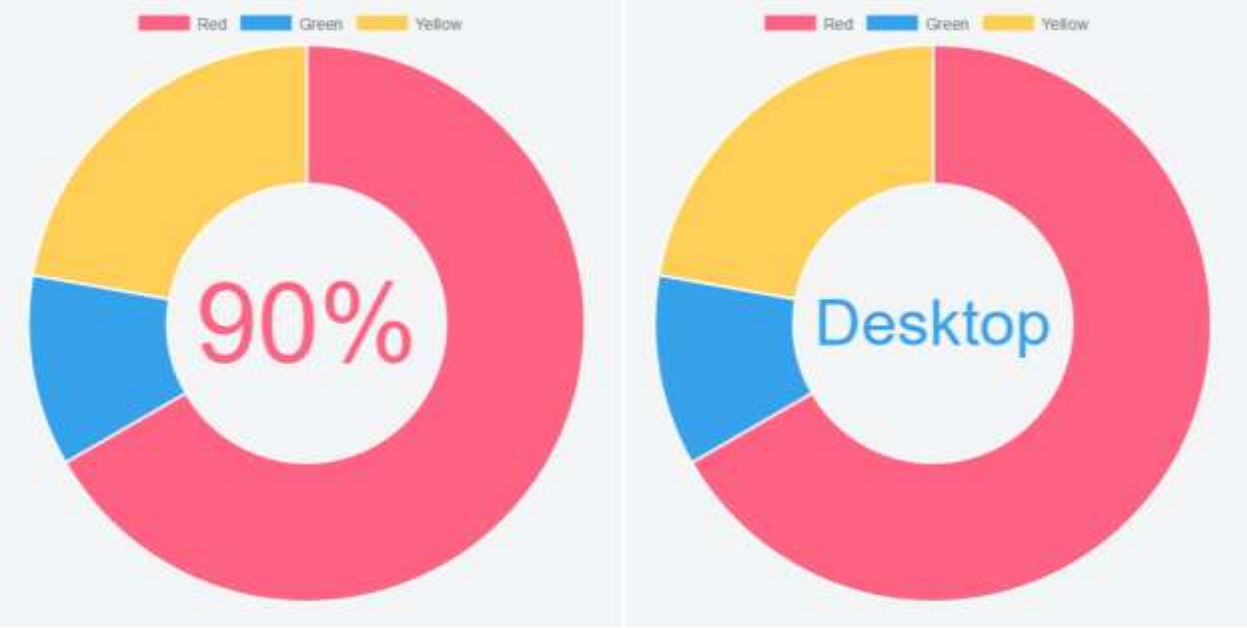


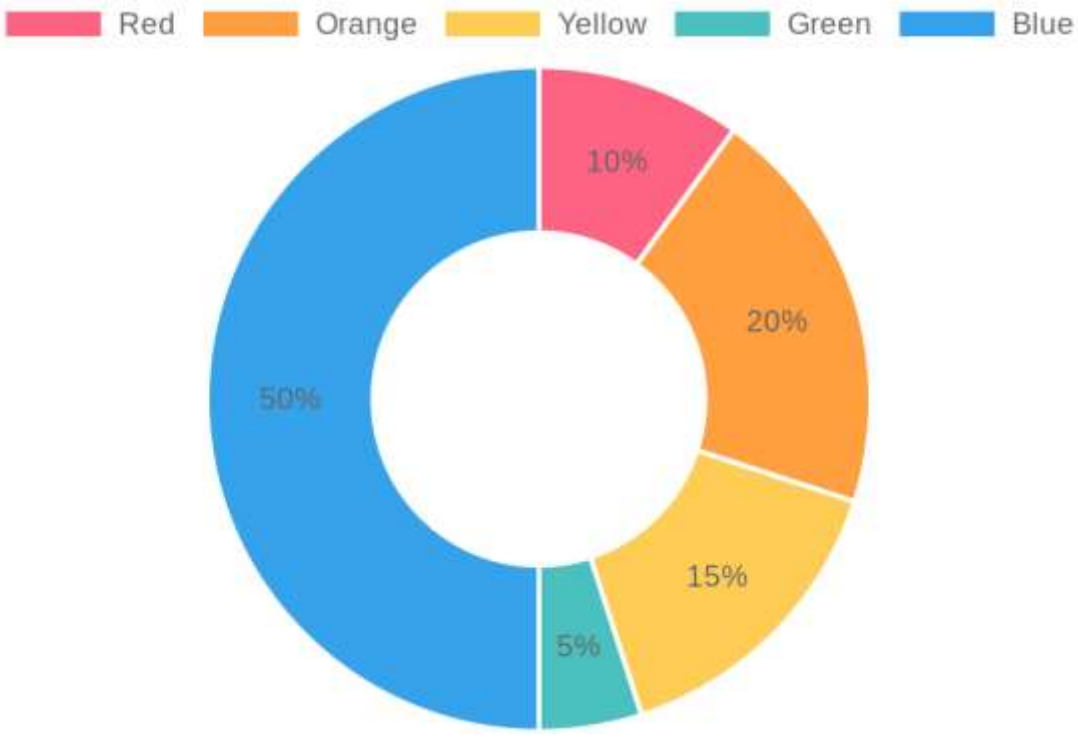
Use cases

- Percentage distribution
- Market share

Type = ChartType.Pie

Doughnut Chart





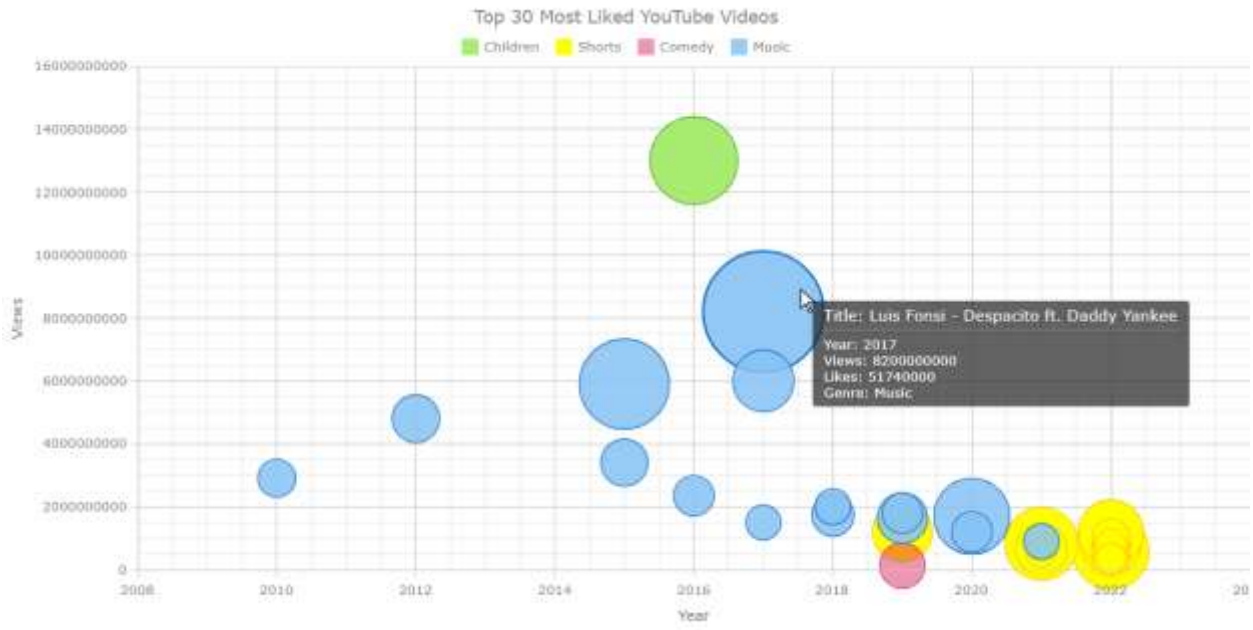
Use cases

- KPI dashboards

- Compact summaries

Type = ChartType.Doughnut

Bubble Chart



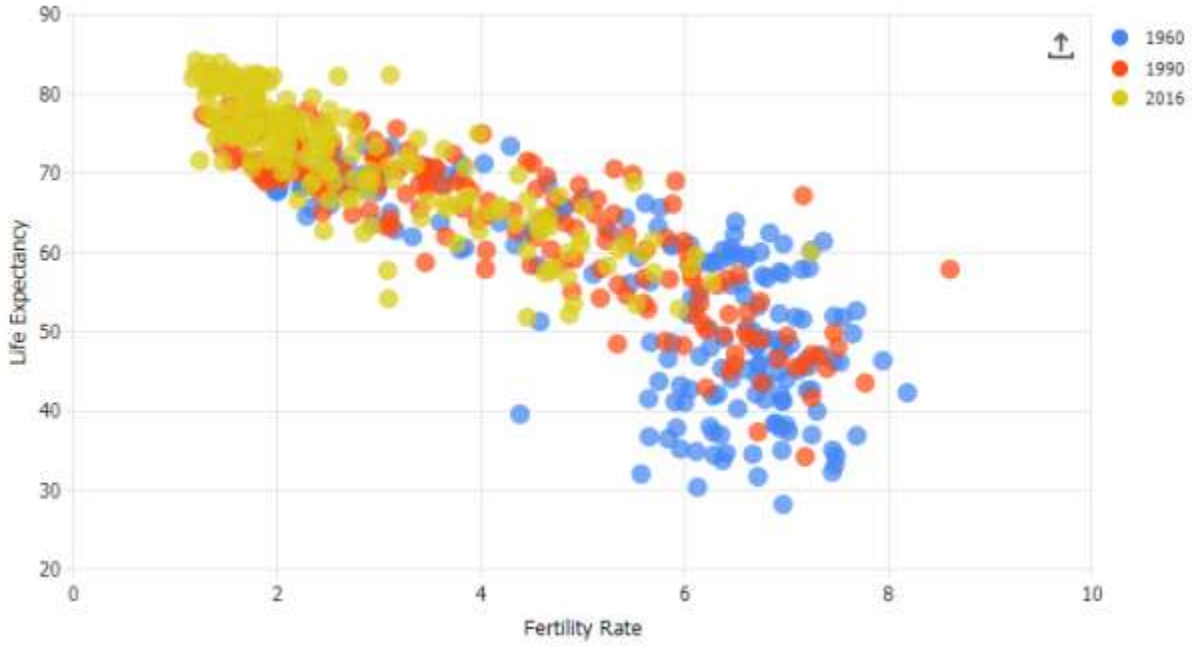
Use cases

- Risk analysis
- Multi-dimension data (X, Y, Size)

Type = ChartType.Bubble

Each data point includes x, y, and r (radius).

Scatter Chart

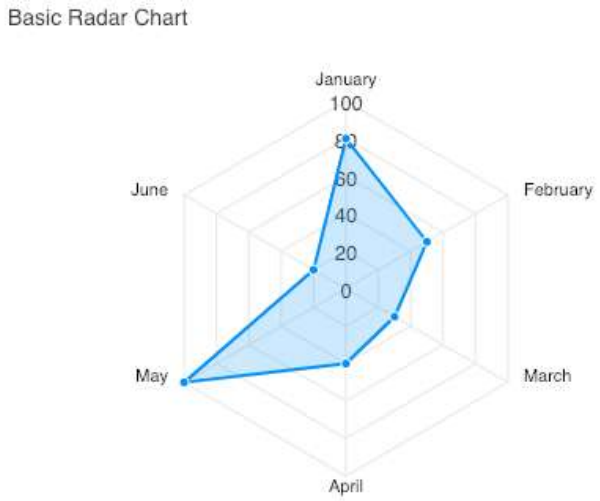


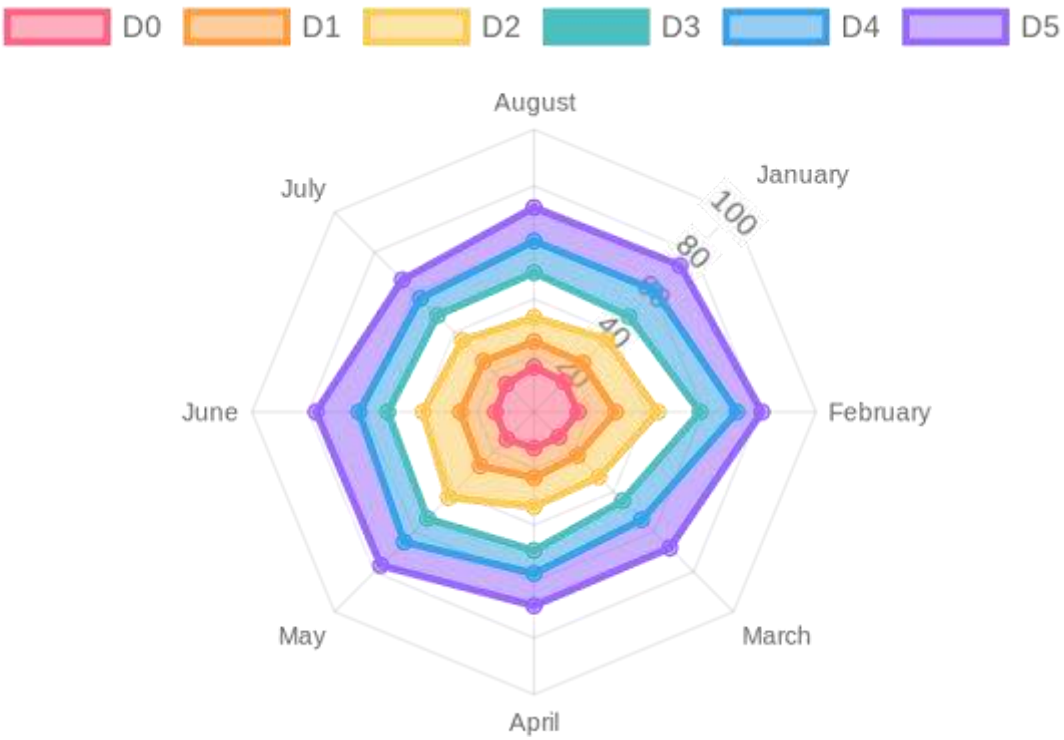
Use cases

- Correlation analysis
- Scientific data

Type = ChartType.Scatter

Radar Chart



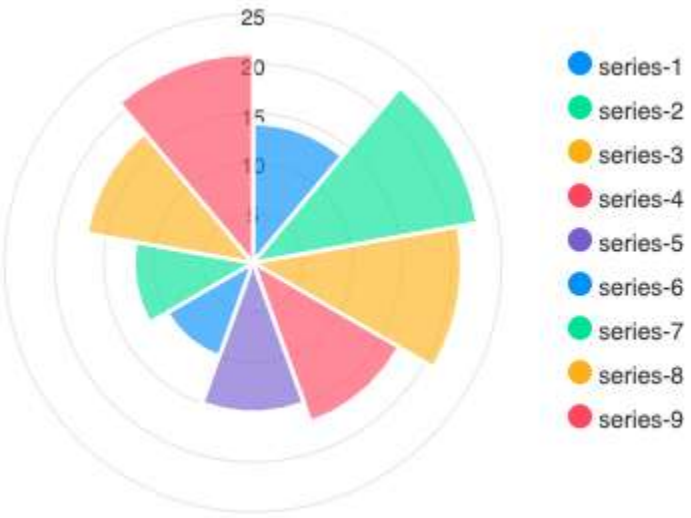


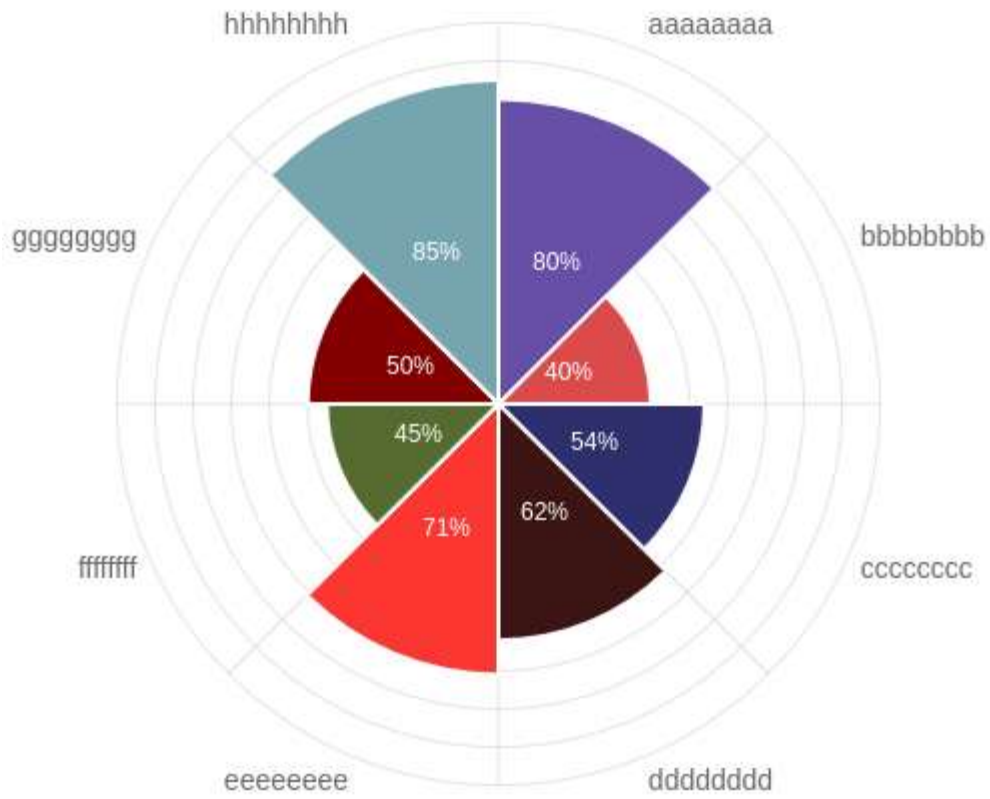
Use cases

- Skill comparison
- Performance evaluation

Type = ChartType.Radar

Polar Area Chart





Use cases

- Weighted categorical data
- Circular data representation

Type = ChartType.PolarArea

Dynamic Data Update Example

```
<button @onclick="UpdateChart">Update</button>


@code {
   void UpdateChart()
   {
      _config.Data.Datasets[0].Data =
         new double[] { 150, 220, 180, 300 };
```

```
    }
}
```

The chart **automatically re-renders** due to Blazor's state change detection.

When to Use Which Chart

Chart Type	Best For
Bar	Comparisons
Line	Trends
Pie/Doughnut	Proportions
Bubble	Multi-dimension
Scatter	Correlation
Radar	Multi-metric comparison
PolarArea	Circular weighted data

Summary

- **BlazorExpress.ChartJS** provides a **clean C# abstraction** over Chart.js
- Works seamlessly with **Blazor WebAssembly & Server (.NET 8)**
- Supports **all major chart types**
- Ideal for **dashboards, analytics, and reporting systems**