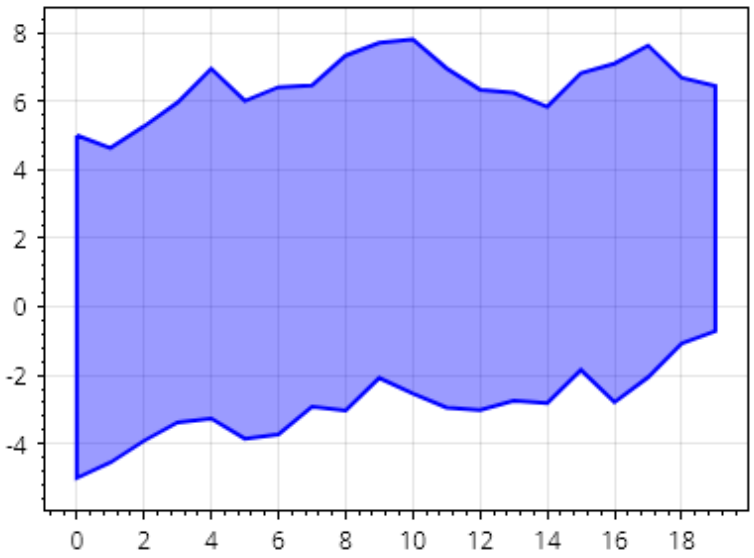
ScottPlot.NET

Stars 5.5k

FillY plot

FillY From Array Data

FillY plots can be created from X, Y1, and Y2 arrays.



FillY.cs

Console WinForms WPF Other

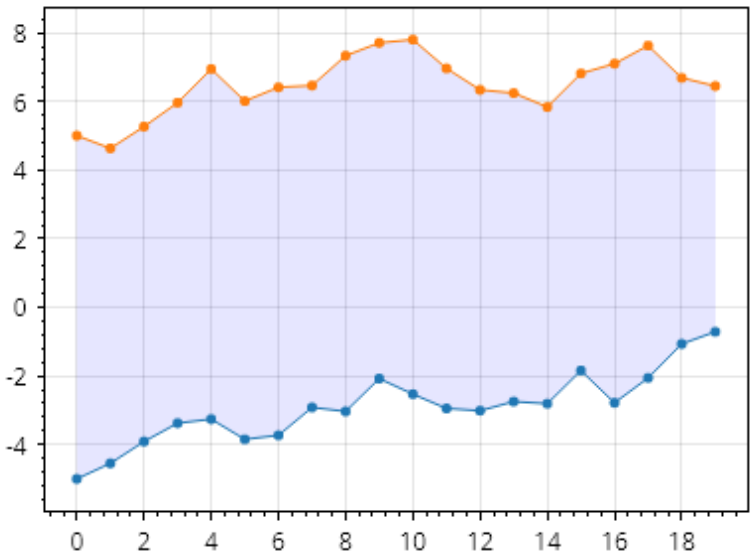
```
ScottPlot.Plot myPlot = new();

RandomDataGenerator dataGen = new(0);

int count = 20;
double[] xs = Generate.Consecutive(count);
double[] ys1 = dataGen.RandomWalk(count, offset: -5);
```

```
double[] ys2 = dataGen.RandomWalk(count, offset: 5);

var fill = myPlot.Add.FillY(xs, ys1, ys2);
fill.FillColor = Colors.Blue.WithAlpha(100);
fill.LineColor = Colors.Blue;
fill.MarkerColor = Colors.Blue;
fill.LineWidth = 2;

myPlot.SavePng("demo.png", 400, 300);
```

Edit on GitHub

FillY From Scatter Plots

FillY plots can be created from two scatter plots that share the same X values.



FillY.cs

Console WinForms WPF Other

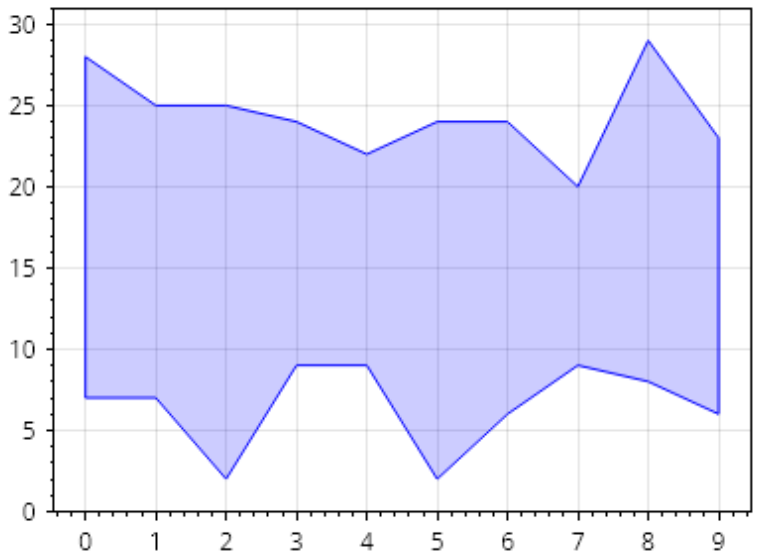
```
ScottPlot.Plot myPlot = new();

RandomDataGenerator dataGen = new(0);

int count = 20;
double[] xs = Generate.Consecutive(count);
```

```
double[] ys1 = dataGen.RandomWalk(count, offset: -5);
double[] ys2 = dataGen.RandomWalk(count, offset: 5);

var scatter1 = myPlot.Add.Scatter(xs, ys1);
var scatter2 = myPlot.Add.Scatter(xs, ys2);

var fill = myPlot.Add.FillY(scatter1, scatter2);
fill.FillColor = Colors.Blue.WithAlpha(.1);
fill.LineWidth = 0;

// push the fill behind the scatter plots
myPlot.MoveToBack(fill);

myPlot.SavePng("demo.png", 400, 300);
```

Edit on GitHub

FillY with Custom Type

FillY plots can be created from data of any type if a conversion function is supplied.



FillY.cs

Console WinForms WPF Other

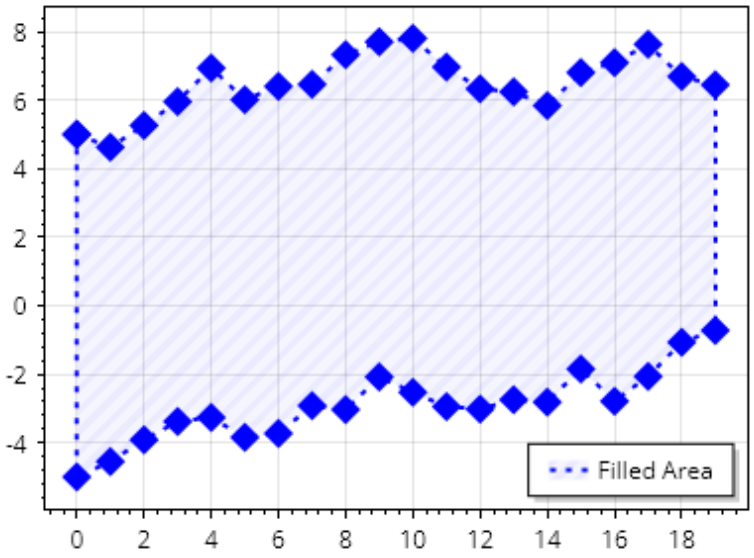
```
ScottPlot.Plot myPlot = new();
```

```
// create source data in a nonstandard data type
List<(int, int, int)> data = new();
Random rand = new(0);
for (int i = 0; i < 10; i++)
{
    int x = i;
    int y1 = rand.Next(0, 10);
    int y2 = rand.Next(20, 30);
    data.Add((x, y1, y2));
}

// create a custom converter for the source data type
static (double, double, double) MyConverter((int, int, int) s) => (s.Item1, s.Item2, s.

// create a filled plot from source data using the custom converter
var fill = myPlot.Add.FillY(data, MyConverter);
fill.FillColor = Colors.Blue.WithAlpha(.2);
fill.LineColor = Colors.Blue;

myPlot.SavePng("demo.png", 400, 300);
```

Edit on GitHub

FillY Plot Styling

FillY plots can be customized using public properties.



FillY.cs

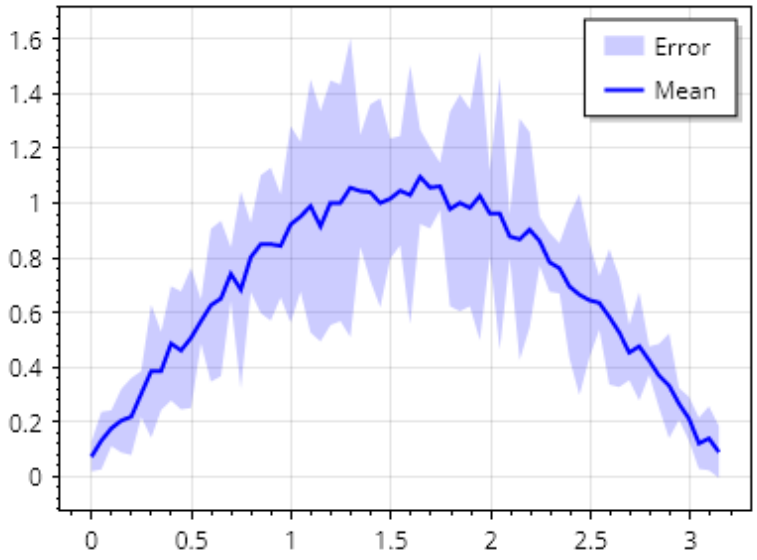
Console WinForms WPF Other

```
ScottPlot.Plot myPlot = new();

int count = 20;
double[] xs = Generate.Consecutive(count);
double[] ys1 = Generate.RandomWalk(count, offset: -5);
double[] ys2 = Generate.RandomWalk(count, offset: 5);

var fill = myPlot.Add.FillY(xs, ys1, ys2);
fill.MarkerShape = MarkerShape.FilledDiamond;
fill.MarkerSize = 15;
fill.MarkerColor = Colors.Blue;
fill.LineColor = Colors.Blue;
fill.LinePattern = LinePattern.Dotted;
fill.LineWidth = 2;
fill.FillColor = Colors.Blue.WithAlpha(.2);
fill.FillHatch = new ScottPlot.Hatches.Striped(ScottPlot.Hatches.StripeDirection.Diagon
fill.FillHatchColor = Colors.Blue.WithAlpha(.4);
fill.LegendText = "Filled Area";

myPlot.ShowLegend();

myPlot.SavePng("demo.png", 400, 300);
```

Edit on GitHub

Filled Error

A line plot with shaded error range may be achieved by layering a FillY beneath a ScatterLine.



FillY.cs

Console WinForms WPF Other

```
ScottPlot.Plot myPlot = new();

// create sample Y values
double[] xs = Generate.Range(0, Math.PI, 0.05);
double[] ys = xs.Select(x => Math.Sin(x) + Generate.RandomNumber(0.1)).ToArray();

// create sample error data
double[] yErr = ys.Select(x => x * Generate.RandomNumber(0.5) + 0.05).ToArray();

// calculate Y ± error
double[] yErrNeg = Enumerable.Range(0, ys.Length).Select(x => ys[x] - yErr[x]).ToArray(
double[] yErrPos = Enumerable.Range(0, ys.Length).Select(x => ys[x] + yErr[x]).ToArray(

// add a shaded area between the error limits
var errFill = myPlot.Add.FillY(xs, yErrNeg, yErrPos);
errFill.LineWidth = 0;
errFill.FillColor = Colors.Blue.WithAlpha(0.2);
errFill.LegendText = "Error";

// add the Y values as a line plot
var meanLine = myPlot.Add.ScatterLine(xs, ys);
meanLine.LineColor = Colors.Blue;
```

```
meanLine.LineWidth = 2;
meanLine.LegendText = "Mean";

// configure the location of the legend
myPlot.Legend.Alignment = Alignment.UpperRight;

myPlot.SavePng("demo.png", 400, 300);
```

Edit on GitHub