Data Visualisation and Graphics

Rich Park
APL is great for:

- **Data manipulation & pre-processing**
  - Cleaning: $\alpha \neq \omega$, $\alpha \land \omega$, $\alpha \lor \omega$, $\alpha \uparrow \omega$, $\alpha \downarrow \omega$
  - Reformatting: $\phi$, $\phi$, $\phi$, $\Box CSV$, $\Box JSON$

- **Exploration**
  - Summaries: $+/\omega$, $\times/\omega$, $\lceil / \rceil$, $\lfloor / \rfloor$
  - Grouping: $F \Box \omega$
  - Sort/Lookup: $\downarrow \omega$, $\alpha \in \omega$, $\alpha \iota \omega$
Communicating ideas

Antarctic Penguin Culmen Size and Penguin Mass

![Graph of Antarctic Penguin Culmen Size and Penguin Mass](image)

- Adelie
- Chinstrap
- Gentoo
Discovering patterns

*Antarctic Penguin Culmen Size and Penguin Mass*
Coming Up...

- SharpPlot/Vega
- HTMLRenderer/¶WC/Observable
- DUI/Abacus

- A little bit of speculation
Previously...

2013: The Future of SharpPlot and RainPro:
   Nicolas Delcros  dyalog.tv/Dyalog13/?v=Xo3vRQMaPxo

2018: Cross-Platform Charting:
   Brian Becker     dyalog.tv/Webinar/?v=wapJgEXSzvc

2020: APL Cultivations (Orchard chat lessons) 44 and 45
   Nicolas Delcros  apl.wiki/APL_Cultivation
SharpPlot

- Cross platform
- Code examples from C#
- Argument nesting is fiddly
- State machine

```
'InitCauseway' ⎕CY 'sharpplot' ⋄ InitCausewayθ

sp←⎕NEW Causeway.SharpPlot

sp.Set<things>
sp.Draw<thing> ⍵data

sp.RenderSvg Causway.SvgMode.FixedAspect
```
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrawBarChart</td>
<td>Draws a grouped or stacked barchart from an array of arrays of values</td>
</tr>
<tr>
<td>DrawBoxPlot</td>
<td>Construct compound box-whisker plot from a data array and two integer category arrays</td>
</tr>
<tr>
<td>DrawBubbleChart</td>
<td>Scatterplot from 2 arrays of data with markers scaled by 3rd array</td>
</tr>
<tr>
<td>DrawCloudChart</td>
<td>3D Scatter from x,y and multiple z values</td>
</tr>
<tr>
<td>DrawConfidence</td>
<td>Draw a dial or gauge from data and pointer radii</td>
</tr>
<tr>
<td>DrawDialChart</td>
<td>Draw current frame, with any fixed text such as headings and footnotes</td>
</tr>
<tr>
<td>DrawGantttChart</td>
<td>Gantt chart from 3 double arrays of Y-values, X-start and X-end values</td>
</tr>
<tr>
<td>DrawHistogram</td>
<td>Construct a histogram from a floating-point vector</td>
</tr>
</tbody>
</table>
Cross-platform charts in Dyalog APL

The best way to create cross-platform charts in Dyalog APL is to use the InitCauseway function. It will pick the .Net assembly if it’s available, else fall back to pure-APL SharpPlot.

It will initialise the argument namespace (current space if @ is given), by creating a Causeway and a System.Drawing namespace there, so that code can be used interoperably on all platforms supported by Dyalog. The advantage over the previous strategies is that the namespace will be cluttered with fewer names (only Causeway, System and Drawing, rather than the whole list of structures used by Causeway).

However all structures will have to be fully qualified: structures from the Causeway assembly (all in this documentation) will have to be prepended with Causeway, and System.Drawing structures (Color, FontStyle) will have to be prepended with System.Drawing. That way the script will be completely cross-platform.

All you need to do to initialise the namespace once with:

```
'InitCauseway' 'View' Dcy 'sharpplot'
InitCauseway @  a initialise current namespace
```

sharpplot.com/Languages.htm

```
data-33 18 12 10
exp=0 0 0 20
key='UK' 'France' 'Italy' 'Ireland'

sp::NEW Causeway.SharpPlot
sp:Heading='Sample pie chart'
sp.PieChartStyle=Causeway.PieChartStyles.ValueTags
sp.KeyStyle=Causeway.KeyStyles.(CenterAlign+BottomAlign)
sp.SetKeyText(ekey)
sp.SetColors =System.Drawing.Color.(Navy Maroon Teal Green)
sp.SetMargins(42 48 36 18)
sp.DrawPieChart(data exp)
sp.SaveSvg(e'samplepie.svg')
svgs=sp.RenderSvg @
```

Data Visualisation
Vega

- Based on D3
  A JavaScript library for manipulating HTML, SVG and CSS

- Extensible data-driven transformations
  In JavaScript 😊

- Vega-lite
  High level description for basic graphics
Vega

- Define using JSON
  - Hand write in an editor
  - Programmatically generate using JSON
Vega

Define using JSON
Hand write in an editor
Programmatically generate using \texttt{⎕JSON}

```
{  
  "config": {  
    "axis": true  
  },  
  "data": {  
    "values": [  
      {"count": 161, "item": "5"},  
      {"count": 166, "item": "3"},  
      {"count": 166, "item": "2"},  
      {"count": 157, "item": "1"},  
      {"count": 171, "item": "6"},  
      {"count": 179, "item": "4"}  
    ],  
    "encoding": {  
      "x": { "field": "item", "type": "nominal" },  
      "y": { "field": "count", "type": "quantitative" }  
    },  
    "height": 300, "mark": "bar", "width": 500  
  }  
vega←{"config":{},"data":{},"encoding":{}}  
vega+⎕JSON{}'{}'  
vega.config+⎕JSON{}'{}'  
vega.config.axis←⊂'true'  
data+⎕JSON{}'{}'  
V2J+{j+⎕JSON{}'{}'}◊j.(item count)←α ω ◊ j  
data.values+⎕JSON{}'{}'  
data.values+V2J/ω  
vega.data←data  
vega.mark←'bar'  
vega.encoding←⎕JSON{}'{}'  
x_enc←{"field": "item", "type": "nominal"}"  
y_enc←{"field": "count", "type": "quantitative"}"  
vega.encoding.x←⎕JSON x_enc  
vega.encoding.y←⎕JSON y_enc  
vega.width←500  
vega.height←300  
(⎕JSON⍠'Compact' 0)vega
```
I guess the interesting bit is how to embed charts in <whatever you're trying to do>
Publishing and embedding

- Print (PDF)
- Raster (PNG, BMP, JPG)
- SVG
- XAML
- WC
- HTMLRenderer
- Browser-based (served)
`plot \{10(\omega)\times02/\omega\}1000`
.NET SharpPlot Viewer
```csharp
'InitCauseway' 'View' ⊕ CY 'sharpplot' ◇ InitCauseway ⊘
sp←NEW Causeway.SharpPlot
sp.DrawLineGraph←{10(iω)×02÷ω}1000
View sp
```
Embed in Windows GUI
Embed in Windows GUI

SharpPlot graph into a Windows Form

RE: SHARPPLOT GRAPH INTO A WINDOWS FORM
by Tomas Gustafsson on Sun Jan 30, 2022 8:27 pm

Beautiful tones, thank you Pierre! I was unaware of the ElementHost control.

In fact I think short samples of both might be valuable for the community? For me personally, WC is the preferred one for now. If you are willingness to spend a few moments on this, I'd be most grateful as might many others too!

RE: SHARPPLOT GRAPH INTO A WINDOWS FORM
by PGilbert on Mon Jan 31, 2022 2:14 am

Here are 2 ways to show the Xaml obtained.

Get a WPF Viewbox from the Xaml representation of the Graph:

CODE: SELECT ALL

```csharp
.NET.sp1.DrawLineGraph(,c10)(t10)

A Render the above Graph as a Xaml text file with 96 dpi
xmlPlot+#.NET.sp1.RenderXaml 96

A Get the WPF Canvas object from the Xaml.
#.NET.USING~'System.Windows.Markup,WPF/PresentationFramework.dll'
#.NET.canvasObj=#.NET.XamlReader.Parse("xmlPlot")

A Get a WPF Viewbox object with the Canvas object as its child.
#.NET.USING~'System.Windows.Controls,WPF/PresentationFramework.dll'
#.NET.viewBox=#.NET.viewBox CHILD(canvasObj)
```

Post: 100
Joined: Mon Sep 19, 2011 6:43 pm

Tomas Gustafsson

Post: 434
Joined: Sun Dec 13, 2009 8:46 pm
Location: Montréal, Québec, Canada

PGilbert
HTMLRenderer

- Cross platform
- Renders SVG
- Use web stack (HTML/CSS/JavaScript)

But I don't want to write JS?
Dyalog User Interface (DUI)
Dyalog User Interface (DUI)

Dyalog MiServer 3 Sample Site
Anyone who can write an APL function should be able to host it on the web.™

MiServer is a free, open-source web server implemented in Dyalog APL. It enables the APL user to build sophisticated websites with little to no knowledge of the web technologies like HTML, JavaScript, etc. We hope that you will enjoy MiServer; join the forums, and contribute via GitHub.

This website serves as guide to MiServer 3 – providing documentation, samples, and advice. At the same time, it showcases many of MiServer’s capabilities. If you are just getting started with MiServer 3, check out the Tutorial tab. Look through the Samples to get an idea of the variety of web content that MiServer can generate.

Explore the tabs for documentation, the broad selection of available controls, and the vast collection of sample pages and mini-apps. See the source of any page by clicking the MS3 logo in the top left corner and access other resources by clicking the Resources link in the top right corner.

HTML video: Specify a list of alternate video sources.
Carlisle Abacus

github.com/the-carlisle-group/Abacus

- Manipulate a Document Object Model (DOM) directly in APL
- Convert APLDOM $\leftrightarrow$ HTML
- Serve in HTMLRenderer
Observable

- Collaborative data story telling
- Easy to insert basic plots
- Compatible with Vega/D3
Data Visualisation

Edward Tufte
  - Keep it simple – less clutter
  - Present in a digestible format

Jacques Bertin
  - *Semiology of Graphics*
  - Mapping of visual properties to data relations

Lee Wilkinson
  - Grammar of Graphics

Inspiration: ggplot2 (R)  D3 (JavaScript)
Present issues, future possibilities

- Read lots of docs
- Learn something that "isn't APL"
- Translate code examples
- Alter your data to fit the framework (easy)
Present issues, future possibilities

- WIBNI we had an APL graphic manipulation framework a la D3/ggplot2?
- Abacus+ ?
- Something else?
APL Media

britishaplassociation.org/webinar-schedule-2022
24th Feb – BAA: Open Session

Dyalog.tv
17th March – Webinar: Link & Sharing Code

(~30th March) – APL Seeds '22