`fmtplot`: Format Code for Plots in `pdf` and `html` Output

lehmann7

2021-09-07

Contents

Abstract– `fmtplot` is a format code, which produces interactive plots for `html` using `flot` and static plots for `pdf` using `matplotlib`. Plots are generated using a `marky` format code with a class `fmtplot` which implements plot and code generation for both cases. The plots which are generated using `fmtplot` can be referenced using `pandoc-fignos`.

---

1 Prerequirements

In order to the `fmtplot` with `html` output, JavaScript libraries have to be downloaded using.

```
cd data
./get_fmtplotjs
```

The script download the following files:

```
jquery.canvaswrapper.js
jquery.colorhelpers.js
jquery.event.drag.js
jquery.flot.browser.js
jquery.flot.drawSeries.js
jquery.flot.hover.js
jquery.flot.image.js
jquery.flot.js
jquery.flot.legend.js
jquery.flot.navigate.js
jquery.flot.resize.js
jquery.flot.saturated.js
jquery.flot.selection.js
jquery.flot.symbol.js
jquery.flot.touch.js
jquery.flot.touchNavigate.js
jquery.flot.uiConstants.js
jquery.js
jquery.mousewheel.js
```

---

2 Include

Include `fmtplot` as follows.

```
<?
___(file="fmtplot.md")
?>
```

3 Plot Generation

Plots are prepared by calling the class `fmtplot` together with for the classes `fmtplot_flot` for `html` and `fmtplot_mplt` for `pdf`. The format code has the following arguments.

```
<?
pltdat = fmtplot(
    html=fmtplot_flot(aspect=(16, 9)),
    pdf=fmtplot_mplt(figdir="data", figsize=(16, 9),
        figdpi=200, fontsize="11pt")
)
?>
```

`aspect` specifies the aspect of a full-width plot for `html` output. for `pdf` output `figdir` specifies the directory for figure output `build/<figdir>/`, `fisize` is the size of the figure in `cm` and `figdpi` and `fontsize` specify DPI number and the font size in `pt`.

In order to init Fig. 1 for output, the format code

```
fmtplot.setup(figid, data, label, style, color)
```

is called with the corresponding arguments.

```
<?
___(pltdat.setup("plot1", data=data,
    label=label, style=style, color=color))
?>
```

Figure Identifier

`figid` is the identifier of the figure, used for internal referencing and referencing inside the document. For `pdf` documents the `figid` also is used for the image filename `build/<figdir>/<figid>.png`.

Plot Data

Plot `data` is specified in two sequences containing `x` and `y` values of point coordinates.

```
<?
import numpy as np
x1 = np.array(range(10)) + 0
x2 = np.array(range(10)) + 1
```

```
x3 = np.array(range(10)) + 2
x4 = np.array(range(10)) + 3
x5 = np.array(range(10)) + 5
x6 = np.array(range(10)) + 6
y1 = 40*np.array(range(10)) + 4
y2 = 30*np.array(range(10)) + 5
y3 = 20*np.array(range(10)) + 6
y4 = 10*np.array(range(10)) + 7
y5 = 10*np.array(range(10)) + 8
y6 = 10*np.array(range(10)) + 9
data=[
    (x1, y1),
    (x2, y2),
    (x3, y3),
    (x4, y4),
    (x5, y5),
    (x6, y6)
]
?>
```

Plot Labels

For each data sequence `(x, y)` the label is specified in a list. A sequence with the label `None` does not appear in the legend and in the `html` choices.

```
<?
label=[
    "Label for (x1, y1)",
    "Label for (x2, y2)",
    "Label for (x3, y3)",
    "Label for (x4, y4)",
    "Label for (x5, y5)",
    "Label for (x6, y6)"
]
?>
```

Plot Style

For each data sequence `(x, y)` the style is specified in a list. The style for one sequence is a tuple where the first element is the `styleid` identifier string and the other elements are arguments `argN` for the `styleid`s in the order of the identifiers in the string.

- ("<styleid>", <arg1>)
- ("<styleid><styleid>", <arg1>, <arg2>)

```
<?
style=[
    ( "^", 11),
    ( "o", 11),
    ( "s", 11),
    ("DL", 11, 2),
    ("+L", 11, 2),
    ("xB", 11, 0.5)
]
?>
```

The styles can be combined `Lo`, `^B`, `DLB` in order to annotate lines and bars with points. There are as many arguments as chars in the `<styleid>` string. The following markers `^osD+x` can be used for points in `<styleid>`. Ths is a subset of the `matplotlib` markers. which is supported by `flot`. `lines` and `bars` are specified using `B` and `L`. Table 1 summarizes the style specification.

Table 1: List of style identifier strings `<styleid>` and corresponding arguments. The bar width is specified relative to the minimum distance between neighbouring `x` points in the data sequence.

`<styleid>`	shape (symbol)	argumet
`o`	points (circle)	point size in `pt`
`s`	points (square)	
`D`	points (diamond)	
`^`	points (triangle)	
`x`	points (cross)	
`+`	points (plus)	
`L`	lines	line width in `pt`
`B`	bars	relative bar width

Plot Color

For each data sequence `(x, y)` the color for lines and points is specified in a list.

```
<?
```

```
color=[
    "#ff0000",
    "#00ff00",
    "#0000ff",
    "#ff00ff",
    "#00ffff",
    "#000000"
]
?>
```

Plot Output in Document

The format code `show(figid, caption="...")` places Fig. 1 in the document using the settings decribed above. Additionally the argument `caption="..."` is used for setting the figure caption. The figure Fig. 1 is referenced by appending `fig:` to the `figid` using the keyword `@fig:plot1`.

```
<?
___(pltdat.setup("plot1", data, label=label, style=style, color=color))
___(pltdat.show("plot1", """This figure is generated using the format code
fmtplot.plot(...) with the arguments data, label, style, color
described above. The figure caption is set using the argument caption.
"""))
?>
```

---

4 Plot Legend

By default `fmtplot` hides the legend. The legend can be placed outside of the plot in a separate canvas/image or inside the plot. In order to configure the legend the arguments `legpos` and `legcols` of the format code constructor are used. `legcols` specifies the number of columns in the legend and `legpos` specifies the legend position using the following values.

- `None`: do not show legend
- `"out"`: show legend in separate image using `fmtplot.legend(figid)`
- `"nw"`: show legend in upper left corner
- `"ne"`: show legend in upper right corner
- `"sw"`: show legend in lower left corner
- `"se"`: show legend in lower right corner

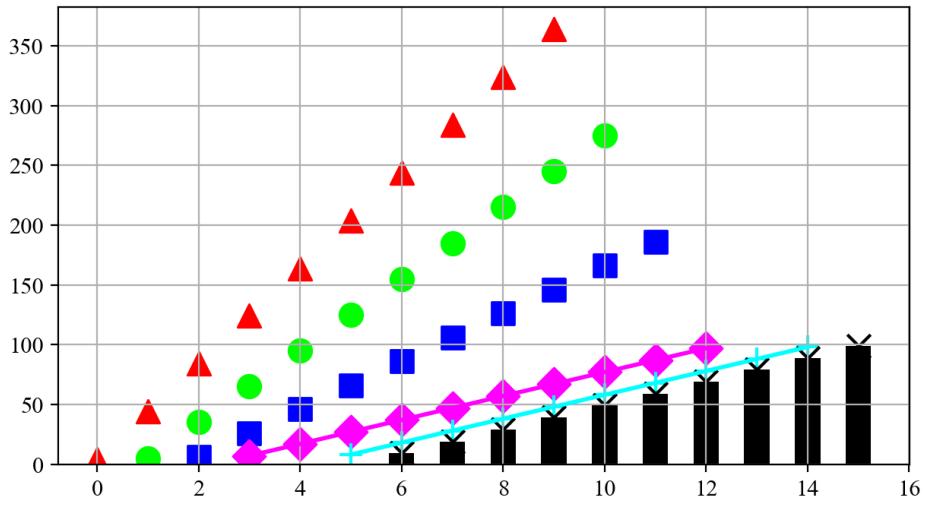

Figure 1: This figure is generated using the format code fmtplot.plot(...) with the arguments data, label, style, color described above. The figure caption is set using the argument caption.

Legend In Plot

A legend in figures is specified using one of the keywords `nw`, `ne`, `sw` or `se` for `legpos`. In Fig. 2, the legendis placed in the upper right corner using `nw`.

```
<?
legin = fmtplot(
    html=fmtplot_flot(legpos="nw", legcols=2),
    pdf=fmtplot_mplt(legpos="nw", legcols=2)
)
___(legin.setup("plot2", data, label=label, style=style, color=color))
___(legin.show("plot2", """This plot is generated with legend inside
the plot using legpos as one of nw, ne, sw, se and with 2 columns
using legcols=2."""))
?>
```

Separate Legend

A legend in a separate image is specified using the keyword `out`. Fig. 3 has a separate legend given in Fig. 4.

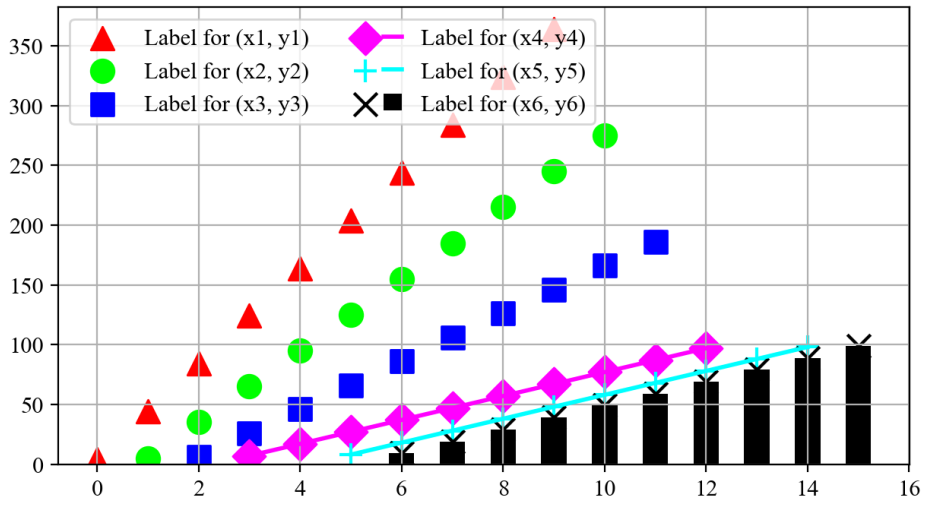

Figure 2: This plot is generated with legend inside the plot using legpos as one of nw, ne, sw, se and with 2 columns using legcols=2.

```
<?
legout = fmtplot(
    html=fmtplot_flot(legpos="out", legcols=2),
    pdf=fmtplot_mplt(legpos="out", legcols=2)
)
___(legout.setup("plot3", data, label=label, style=style, color=color))
___(legout.show("plot3", """This plot is generated with separate legend
using legpos=out and with 2 columns using legcols=2."""))
?>
```

Placement of Separate Legend

In order to place the separate legend for Fig. 3, the `legend(figid, caption="...")` format code is called. The figure Fig. 3 is referenced by appending `fig:` to the `figid` using the keyword `@fig:plot3`. The legend in Fig. 4 is referenced by additionally appending `-legend` using the keyword `@fig:plot3-legend`.

```
<?
___(legout.legend("plot3", caption="""This is the legend for @fig:plot3.
It was placed using the format code fmtplot.legend(figid, caption)."""))
?>
```

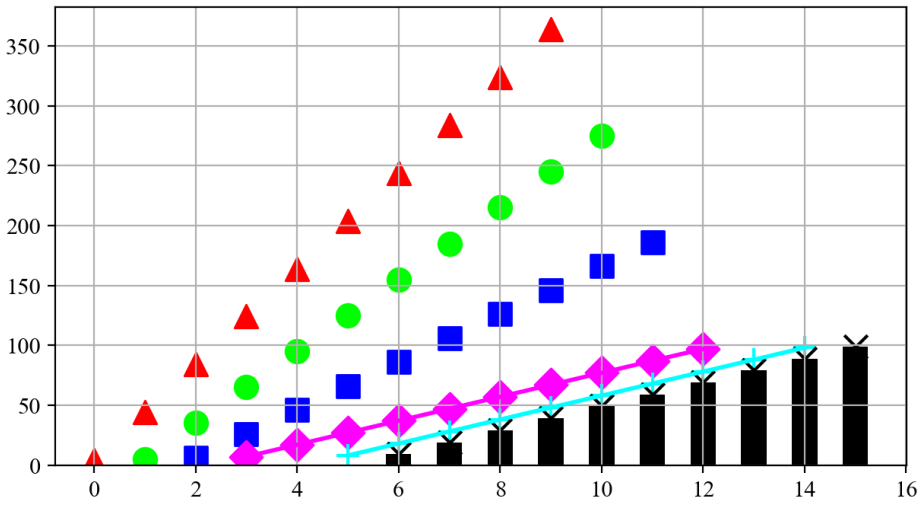

Figure 3: This plot is generated with separate legend using legpos=out and with 2 columns using legcols=2.

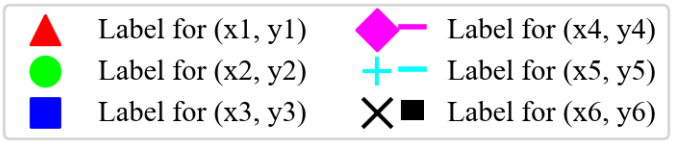

Figure 4: This is the legend for Fig. 3. It was placed using the format code fmtplot.legend(figid, caption).

5 Plot Example

```
<?
ex = fmtplot(
    html=fmtplot_flot(legpos="out", legcols=2),
    pdf=fmtplot_mplt(legpos="out", legcols=2)
)
___(ex.setup(
    "plotex",
    [
        (x1, y1), (x2, y2),
        (x3, y3), (x4, y4)
    ],
    label=[
        "Label for (x1, y1)",
        "Label for (x2, y2)",
        "Label for (x3, y3)",
        None
    ],
    style=[
        ("o", 11),
        ("L", 2),
        ("LD", 1, 5),
        ("Bo", 0.5, 10)
    ],
    color=[
        "#ff0000",
        "#00ff00",
        "#0000ff",
        "#000000",
    ]
))
?>

<?
___(ex.show("plotex", """This plot is generated."""))
?>
```

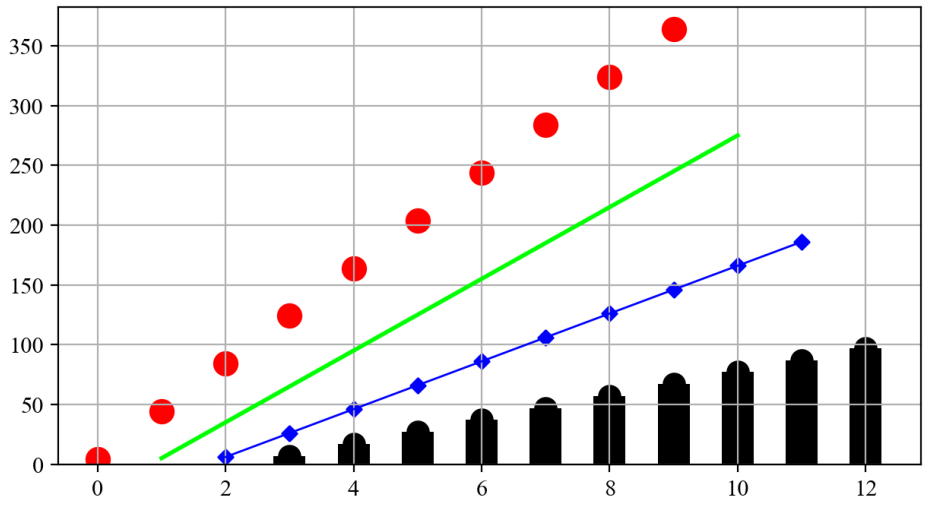

Figure 5: This plot is generated.

```
<?
___(ex.legend("plotex", caption="""This is the legend for @fig:plotex."""))
?>
```

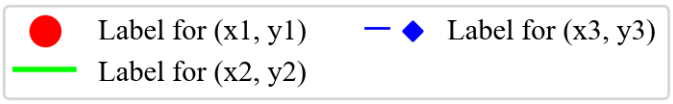

Figure 6: This is the legend for Fig. 5.

---

6 HTML Choice Placement

`html` output supports interactive plots with zooming and paning and enabling and disabling the plot entities using checkboxes. The is feature is demonstrated in Fig. 7. The checkboxes are located below the figure.

```
<?
choice = fmtplot(
    html=fmtplot_flot(legpos="ne", legcols=2),
    pdf=fmtplot_mplt(legpos="ne", legcols=2)
```

```
)
___(choice.setup("plot4", data, label=label, style=style, color=color))
___(choice.show("plot4", """This plot is used together with the format
code fmtplot.choice(figid). For html output, a list of checkboxes is
generated inside a div-tag. For pdf no output is generated."""))
?>
```

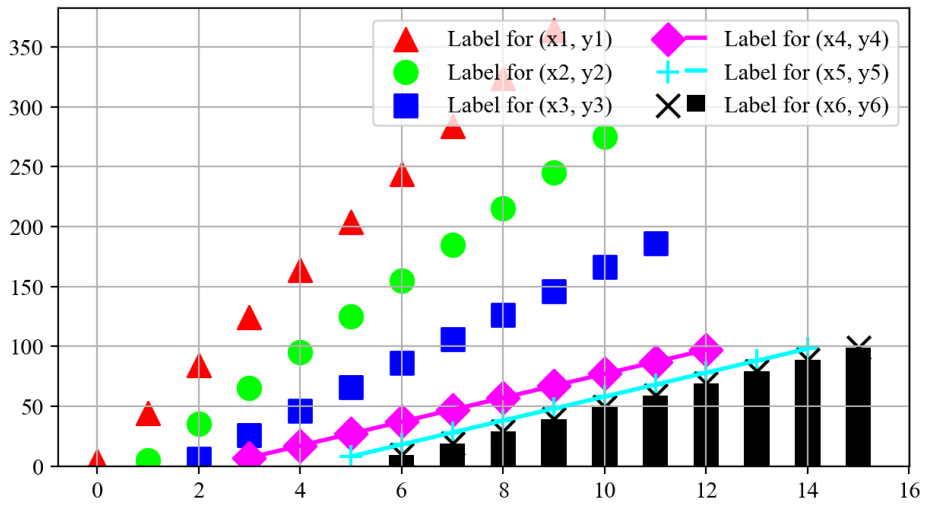

Figure 7: This plot is used together with the format code fmtplot.choice(figid). For html output, a list of checkboxes is generated inside a div-tag. For pdf no output is generated.

In order to place the choice checkboxes for Fig. 7, the `choice(figid)` format code is used. The checkboxes are placed inside a `<div>` tag. For `pdf` output there no choice is displayed.

```
<?
___(choice.choice("plot4"))
?>
```

---

7 Java Script Placement

`html` requires the placement of JavaScript which contains the plot data and setup code for the plots. The JavaScript is inserted into `html` documents

using the the format code `fmtplot.script()`. JavaScripts have to be placed at the end of the document. For `pdf` no output is generated.

```
<?
___(pltdat.script())
___(ex.script())
___(legin.script())
___(legout.script())
___(choice.script())
?>
```

Thanks for reading, please try `fmtplot`.
