

pres

July 29, 2014

```
In [1]: %run talktools
```

```
<IPython.core.display.HTML at 0x7fd2c4a55510>
```

1 Plotly collaborative, interactive $T_{\text{E}}X$ and online plotting with

1.1 What is Plotly?

Plotly is an **online** analytics and data visualization tool.

```
In [2]: from IPython.display import IFrame
```

```
IFrame("https://plot.ly/~Dreamshot/407", width="900px", height='500px')
```

```
Out[2]: <IPython.lib.display.IFrame at 0x7fd2c52ad7d0>
```

1.2 Why is Plotly at TUG 2014?

- You can use $T_{\text{E}}X$ symbols to annotate Plotly graphs, (using the [MathJax](#) display engine)

```
In [3]: # from IPython.display import IFrame
```

```
IFrame("https://plot.ly/~PlotBot/3", width="900px", height='500px')
```

```
Out[3]: <IPython.lib.display.IFrame at 0x7fd2c52ad710>
```

1.3 But really ... Why is Plotly at TUG 2014?

... **Philosophy** $T_{\text{E}}X$ provides a system that gives exactly the same results on all computers

- $T_{\text{E}}X$ is **cross-operating system**

$T_{\text{E}}X$ allows anybody to produce high-quality documents efficiently

- $T_{\text{E}}X$ is a **free** and **cross-text editor** platform

Plotly applies the same core principles to graphs

- Plotly is **free**, **cross-operating system** and **cross-scientific computing language**

1.3.1 Collaboration in data-intensive fields sometimes feels like this:

Plotly solves this collaboration (i.e. reproducibility) problem:

- Plotly graphs are **closely connected to their underlying data** (more later)
- Plotly graphs are **stored in the cloud**
- Plotly provides a **common graphing platform** for Python, MATLAB, R, Node.js, Julia and Excel users.

1.3.2 A Plotly graph made in Python

```
In [4]: # from IPython.display import IFrame
```

```
IFrame("https://plot.ly/python/histograms/#Overlaid-Histogram", width="1100px", height='520px')
```

```
Out[4]: <IPython.lib.display.IFrame at 0x7fd2c52b6fd0>
```

1.3.3 The same Plotly graph, now made in MATLAB

```
In [5]: # from IPython.display import IFrame
```

```
IFrame("https://plot.ly/matlab/histograms/#Overlaid-Histogram", width="1100px", height='520px')
```

```
Out[5]: <IPython.lib.display.IFrame at 0x7fd2c4a55210>
```

1.3.4 ... and again the same Plotly graph, now made in R

```
In [6]: # from IPython.display import IFrame
```

```
IFrame("https://plot.ly/r/histograms/#Overlaid-Histogram", width="1100px", height='520px')
```

```
Out[6]: <IPython.lib.display.IFrame at 0x7fd2c52adc50>
```

1.3.5 What if I already have plot-generating code written?

That's OK.

Our libraries come with **figure converters** allowing

- [MATLAB](#),
- [matplotlib](#) and
- [ggplot2](#)

figures to be converted to Plotly figures with one line of code!

```
In [7]: # from IPython.display import IFrame
```

```
IFrame("https://plot.ly/matplotlib/", width="1100px", height='520px')
```

```
Out[7]: <IPython.lib.display.IFrame at 0x7fd2c4a55250>
```

1.3.6 How to make Plotly graph using our web app

- Have some [data](#) to plot
- Go to [plot.ly](#)

1.3.7 Plotly rhymes with collaboration and reproducibility

Plotly allows you to **retrieve** a figure's underlying JSON object!

For example, in Python:

```
In [8]: import plotly.plotly as py
```

```
fig = py.get_figure("https://plot.ly/~etpinard/448")
```

```
fig
```

```
Out[8]: {'data': [{'name': u'Pop.',  
  'type': u'scatter',  
  'x': [u'1860',  
    u'1870',  
    u'1880',  
    u'1890',  
    u'1900',  
    u'1910',  
    u'1920',  
    u'1930',  
    u'1940',  
    u'1950',  
    u'1960',  
    u'1970',  
    u'1980',  
    u'1990',  
    u'2000',  
    u'2010',  
    u'2013'],  
  'y': [u'2,874',  
    u'8,293',  
    u'17,577',  
    u'46,385',  
    u'90,426',  
    u'207,214',  
    u'258,288',  
    u'301,815',  
    u'305,394',  
    u'373,628',  
    u'372,676',  
    u'382,619',  
    u'366,383',  
    u'437,319',  
    u'529,121',  
    u'583,776',  
    u'609,456']},  
  {'line': {'color': u'rgb(255, 127, 14)', 'dash': u'solid', 'width': 6},  
  'name': u'Pop. - fit',  
  'opacity': 0.5,  
  'type': u'scatter',  
  'x': [1860,  
    1863.1224489795918,  
    1866.2448979591836,  
    1869.3673469387754,
```

1872.4897959183672,
 1875.6122448979593,
 1878.734693877551,
 1881.857142857143,
 1884.9795918367347,
 1888.1020408163265,
 1891.2244897959183,
 1894.3469387755101,
 1897.469387755102,
 1900.591836734694,
 1903.7142857142858,
 1906.8367346938776,
 1909.9591836734694,
 1913.0816326530612,
 1916.204081632653,
 1919.3265306122448,
 1922.4489795918366,
 1925.5714285714287,
 1928.6938775510205,
 1931.8163265306123,
 1934.938775510204,
 1938.061224489796,
 1941.1836734693877,
 1944.3061224489795,
 1947.4285714285713,
 1950.5510204081634,
 1953.6734693877552,
 1956.795918367347,
 1959.9183673469388,
 1963.0408163265306,
 1966.1632653061224,
 1969.2857142857142,
 1972.408163265306,
 1975.530612244898,
 1978.6530612244899,
 1981.7755102040817,
 1984.8979591836735,
 1988.0204081632653,
 1991.142857142857,
 1994.265306122449,
 1997.3877551020407,
 2000.5102040816328,
 2003.6326530612246,
 2006.7551020408164,
 2009.8775510204082,
 2013],
 'xaxis': u'x',
 'y': [-24905.93916280754,
 -12636.203280905262,
 -366.4673990039155,
 11903.268482898362,
 24173.00436480064,
 36442.74024670292,
 48712.476128605194,

```

60982.21201050747,
73251.94789240882,
85521.6837743111,
97791.41965621337,
110061.15553811472,
122330.891420017,
134600.6273019202,
146870.36318382155,
159140.09906572383,
171409.8349476261,
183679.57082952745,
195949.30671142973,
208219.042593332,
220488.77847523335,
232758.51435713656,
245028.25023903884,
257297.9861209402,
269567.72200284246,
281837.45788474474,
294107.1937666461,
306376.92964854836,
318646.66553045064,
330916.4014123529,
343186.1372942552,
355455.8731761575,
367725.6090580588,
379995.3449399611,
392265.0808218634,
404534.81670376565,
416804.552585667,
429074.2884675702,
441344.0243494725,
453613.76023137383,
465883.4961132761,
478153.2319951784,
490422.96787707973,
502692.703758982,
514962.4396408843,
527232.1755227866,
539501.9114046888,
551771.6472865911,
564041.3831684925,
576311.1190503947],
'yaxis': u'y'}],
'layout': {'annotations': [{'align': u'left',
'arrowcolor': u'#636363',
'arrowhead': 2,
'arrowsize': 1,
'arrowwidth': 2,
'ax': -265,
'ay': -112,
'bgcolor': u'rgba(0,0,0,0)',
'bordercolor': u'',
'borderpad': 1,

```

```

'borderwidth': 1,
'font': {'size': 28},
'opacity': 0.8,
'showarrow': True,
'text': u'$R^2 = 0.960 \\\\ y = -7.33\\times10^6 + 3.93\\times10^3 x$',
'textangle': 0,
'x': 1933.706238213492,
'xanchor': u'auto',
'xref': u'x',
'y': 262636.0070362699,
'yanchor': u'auto',
'yref': u'y'}],
'autosize': True,
'bargap': 0.2,
'bargroupgap': 0,
'barmode': u'group',
'boxmode': u'overlay',
'dragmode': u'zoom',
'font': {'color': u'#444',
'family': u'"Open sans", verdana, arial, sans-serif',
'size': 12},
'height': 474,
'hidesources': False,
'hovermode': u'x',
'legend': {'bgcolor': u'fff',
'bordercolor': u'#444',
'borderwidth': 0,
'font': {'color': u'', 'family': u'', 'size': 0},
'traceorder': u'normal',
'x': 1.02,
'xanchor': u'left',
'y': 1,
'yanchor': u'top'},
'margin': {'autoexpand': True,
'b': 80,
'l': 80,
'pad': 0,
'r': 80,
't': 100},
'paper_bgcolor': u'fff',
'plot_bgcolor': u'fff',
'separators': u'. ,',
'showlegend': True,
'title': u"Portland's Pop",
'titlefont': {'color': u'', 'family': u'', 'size': 0},
'width': 1297,
'xaxis': {'anchor': u'y',
'autorange': True,
'autotick': True,
'domain': [0, 1],
'dtick': 20,
'exponentformat': u'B',
'gridcolor': u'eee',
'gridwidth': 1,

```

```

'linecolor': u'#444',
'linewidth': 1,
'mirror': False,
'nticks': 0,
'overlying': False,
'position': 0,
'range': [1851.5, 2021.5],
'rangemode': u'normal',
'showexponent': u'all',
'showgrid': True,
'showline': False,
'showticklabels': True,
'tick0': 0,
'tickangle': u'auto',
'tickcolor': u'#444',
'tickfont': {'color': u'', 'family': u'', 'size': 0},
'ticklen': 5,
'ticks': u'',
'tickwidth': 1,
'title': u'Census',
'titlefont': {'color': u'', 'family': u'', 'size': 0},
'type': u'linear',
'zeroline': True,
'zerolinecolor': u'#444',
'zerolinewidth': 1},
'yaxis': {'anchor': u'x',
'autorange': True,
'autotick': True,
'domain': [0, 1],
'dtick': 100000,
'exponentformat': u'B',
'gridcolor': u'eee',
'gridwidth': 1,
'linecolor': u'#444',
'linewidth': 1,
'mirror': False,
'nticks': 0,
'overlying': False,
'position': 0,
'range': [-60148.26911629685, 644698.3299534894],
'rangemode': u'normal',
'showexponent': u'all',
'showgrid': True,
'showline': False,
'showticklabels': True,
'tick0': 0,
'tickangle': u'auto',
'tickcolor': u'#444',
'tickfont': {'color': u'', 'family': u'', 'size': 0},
'ticklen': 5,
'ticks': u'',
'tickwidth': 1,
'title': u'Pop.',
'titlefont': {'color': u'', 'family': u'', 'size': 0},

```

```
'type': u'-',
'zeroline': True,
'zerolinecolor': u'#444',
'zerolinewidth': 1}}}
```

1.3.8 Remake this plot, with a few modifications

```
In [9]: import plotly.plotly as py
fig = py.get_figure("https://plot.ly/~etpinard/448") # as in last slide

# Modify the title
fig['layout'].update(title="The Historical Population of Portland, OR")

# Modify the y-axis label
fig['layout']['yaxis'].update(title="Population")

# Plots the data with marker points (not line)
fig['data'][0].update(mode='markers')

# Re-generate plot, get a unique URL
py.plot(fig, filename="tug-conf-ex")
```

Out[9]: u'https://plot.ly/~etpinard/449'

Go to plot's URL (or use `py.iplot()` to embed plot in IPython notebook)

1.4 Moreover, Plotly

- is also a social network (with a twitter-like *feed* of figures and [commenting](#) on each graphs)
- allows users to make *private* figures (kind of like github, see our [plans](#))
- allows users to make *streaming* plots (e.g. a never-ending double pendulum simulation: [graph](#) and [code](#))
- is developing of Open Source Libraries (e.g [Python](#), [MATLAB](#), [R](#))

Thank you.