In [1]:

```
# setup notebook
# notebook formatting
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:90% !important; }</style>"))
```

executed in 9ms, finished 13:09:51 2019-11-21

In [2]:

```
# enable split cells in notebook
# if not installed:  pip install jupyter_contrib_nbextensions; then repeat this cmd
!jupyter nbextension enable splitcell/splitcell

# pretty print all cell's output and not just the last one
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

# imports
import pandas as pd

# using holoviews with bokeh to plot streaming data
import holoviews as hv
hv.extension("bokeh")
from holoviews import opts
from holoviews.streams import Pipe, Buffer

import holoplot
holoplot.patch("streamz")

from bokeh.plotting import output_notebook
output_notebook() # set default; alternative is output_file()

import streamz
import streamz.dataframe
```

executed in 2.03s, finished 13:09:53 2019-11-21

```
Enabling notebook extension splitcell/splitcell...
      - Validating: ok
```

(https://bokeh.org) BokehJS 1.4.0 successfully loaded.

Bokeh with streaming data

image credit: @livingstills via giphy (https://giphy.com/livingstills)

Example 1 - Bokeh & Holoviews with streaming data from Streamz lib

Holoviews docs (quick flyby)

http://holoviews.org/index.html (http://holoviews.org/index.html)

In [3]:

```python
# create new stream of random data

# streamz - create pipelines, and generate data:  https://streamz.readthedocs.io/en/latest/
# integrated with pandas

from streamz.dataframe import Random
df = Random(freq='10ms', interval='100ms')
df.tail(5)
```

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

In [4]:

```python
# create table with rolling window

print("table - rolling summary of `y` frequency:")
df.window('5s').groupby('y').x.sum().round(0)
```

```
table - rolling summary of `y` frequency:
```

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

In [5]:

```python
# display stream and the summaries
print("the raw stream:")
display(df.tail(5))

```

```
the raw stream:
```

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

In [6]:

```
print("graph - rolling summary of `y
display(df.window('5s').groupby('y')
```

executed in 588ms, finished 13:09:54 2019-11-21

```
graph - rolling summary of `y` frequenc
y:
```

In [7]:

```
# stop the stream
df.stop()
```

executed in 3ms, finished 13:09:54 2019-11-21

Example 2 - computations on stream with rolling window

In [8]:

```
simple_sdf = streamz.dataframe.Random(freq='10ms', interval='100ms')
sdf = (simple_sdf-0.5).cumsum()
```

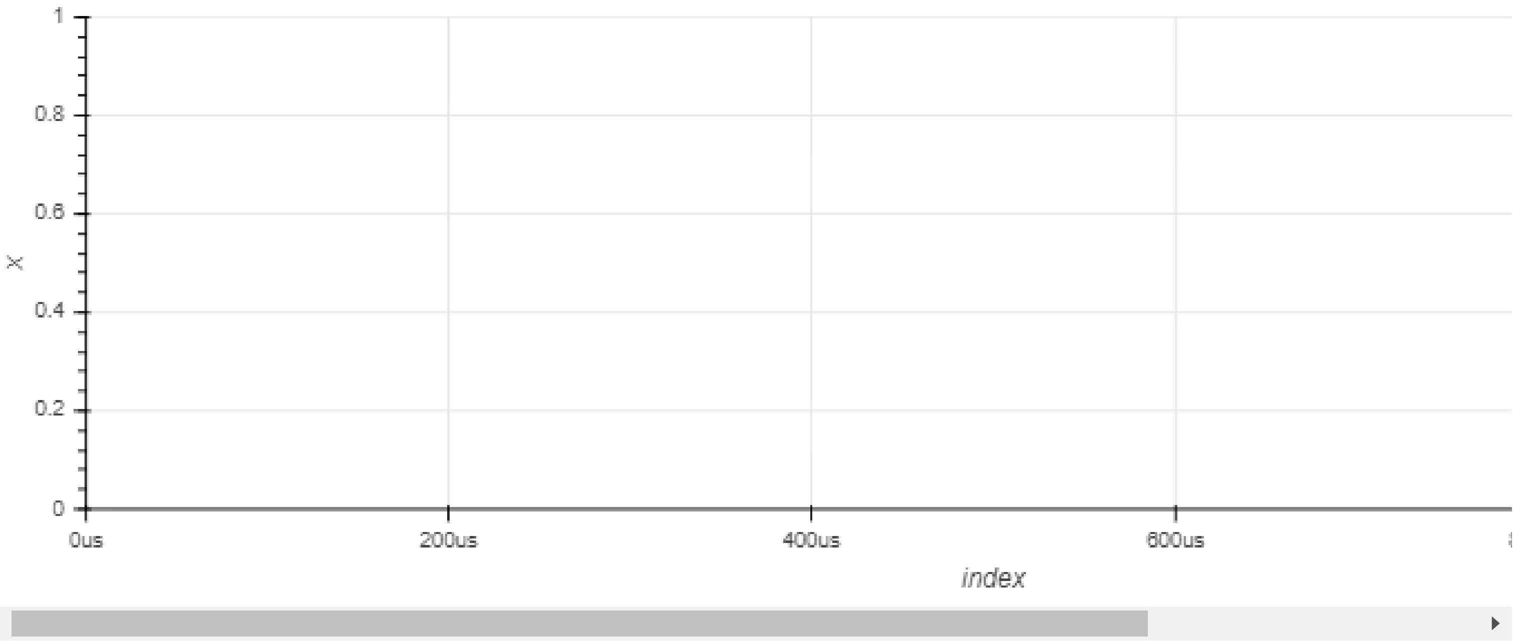
executed in 8ms, finished 13:09:54 2019-11-21

In [9]:

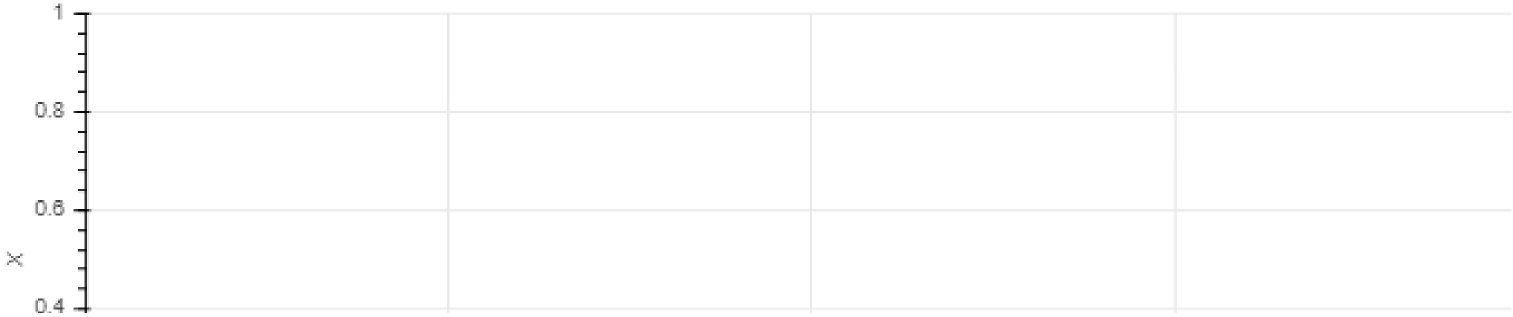
```
raw_stream = hv.DynamicMap(hv.Curve, streams=[Buffer(sdf.x)]).opts(height=300, show_grid=True
smooth_stream = hv.DynamicMap(hv.Curve, streams=[Buffer(sdf.x.rolling('500ms').mean())]).opts

# (raw_dmap.relabel('raw') * smooth_dmap.relabel('smooth')).opts(opts.Curve(width=500, show_g
raw_stream
smooth_stream
(raw_stream.relabel("raw") * smooth_stream.relabel("smooth"))

# stop the stream
# simple_sdf.stop()
# sdf.stop()
```

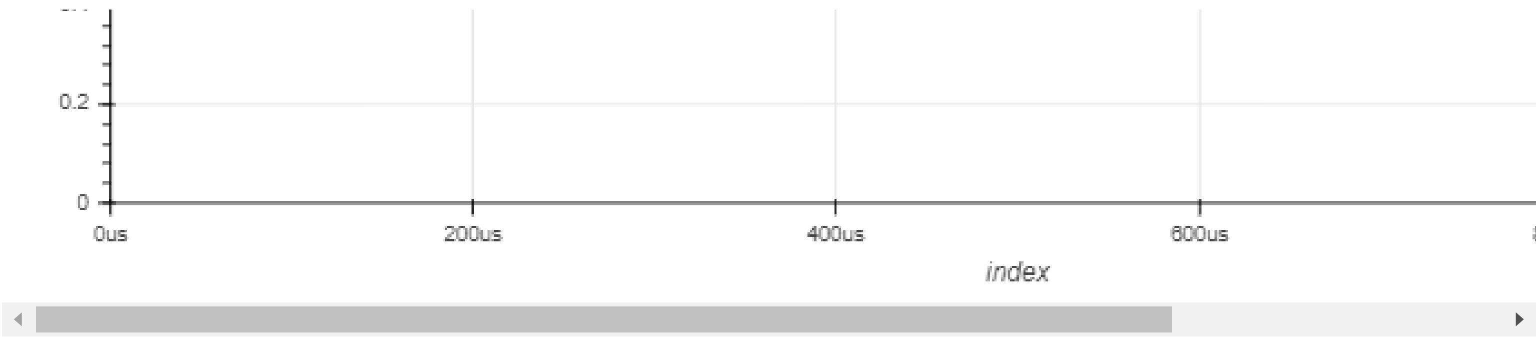
executed in 283ms, finished 13:09:54 2019-11-21

Out[9]:

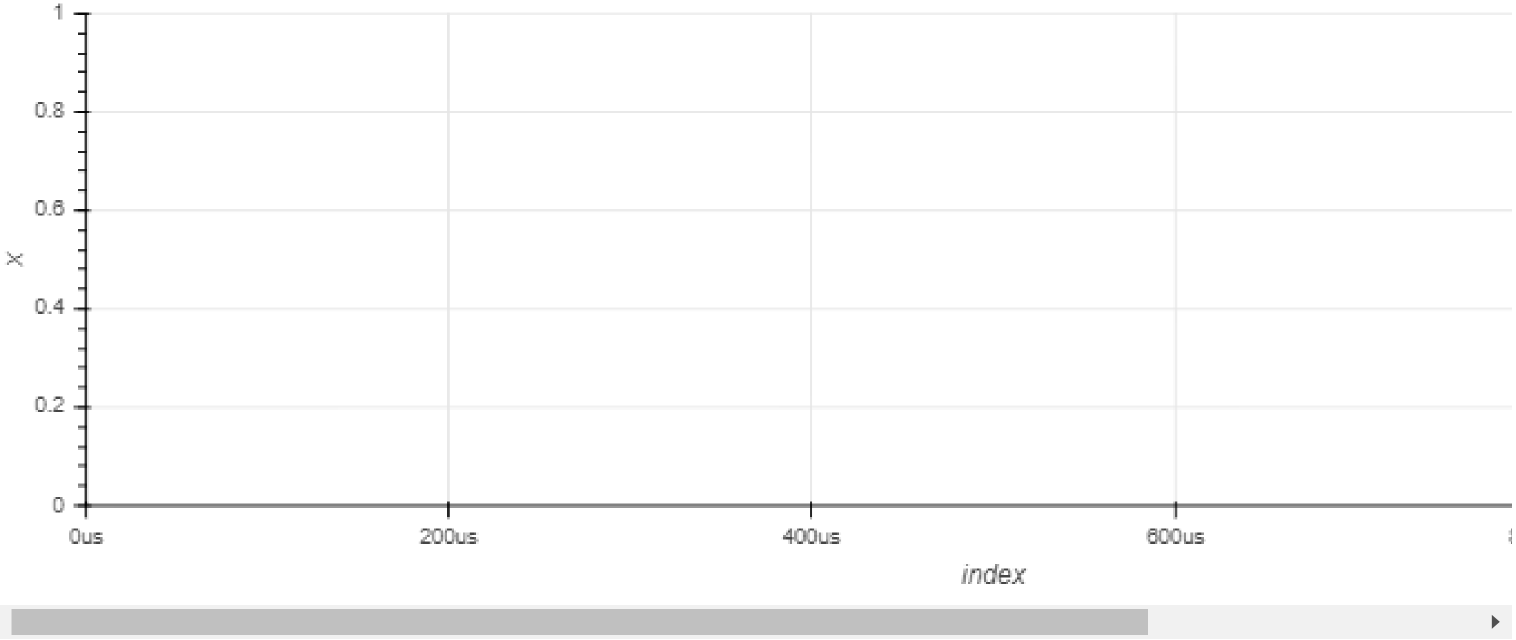


Out[9]:





Out[9]:



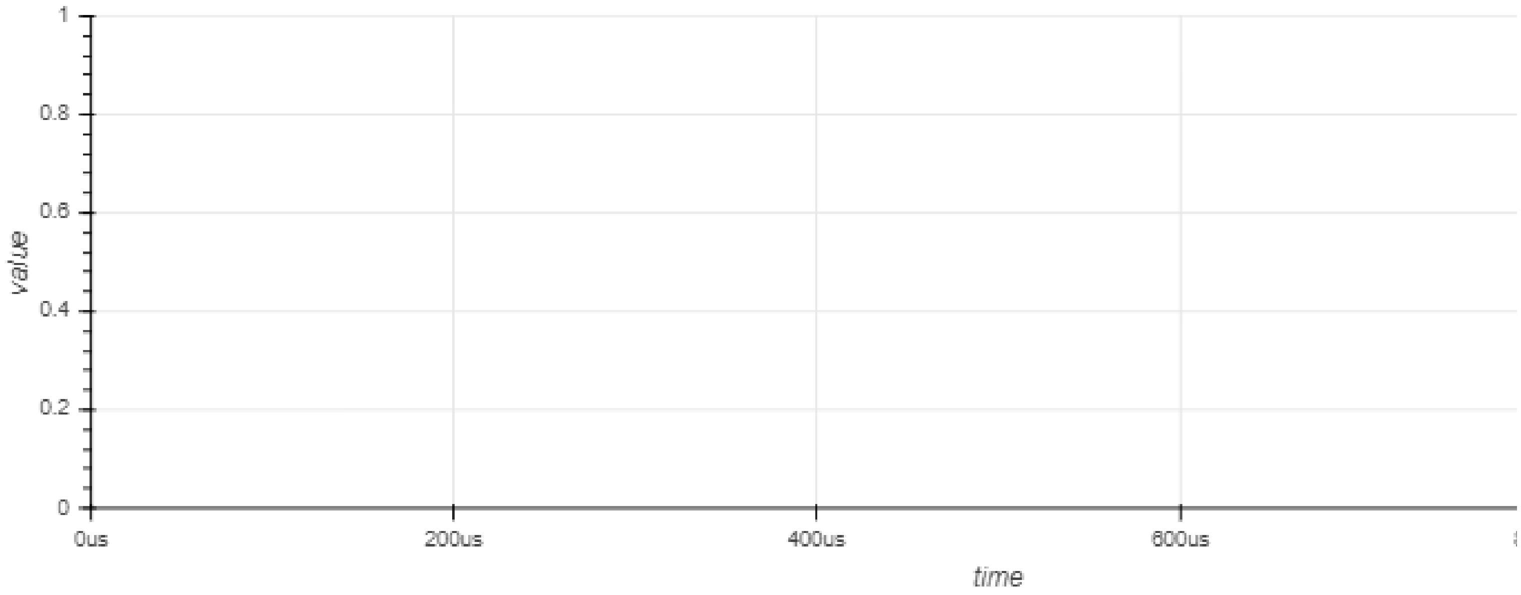
Example 3 - Update multiple cells

In [10]:

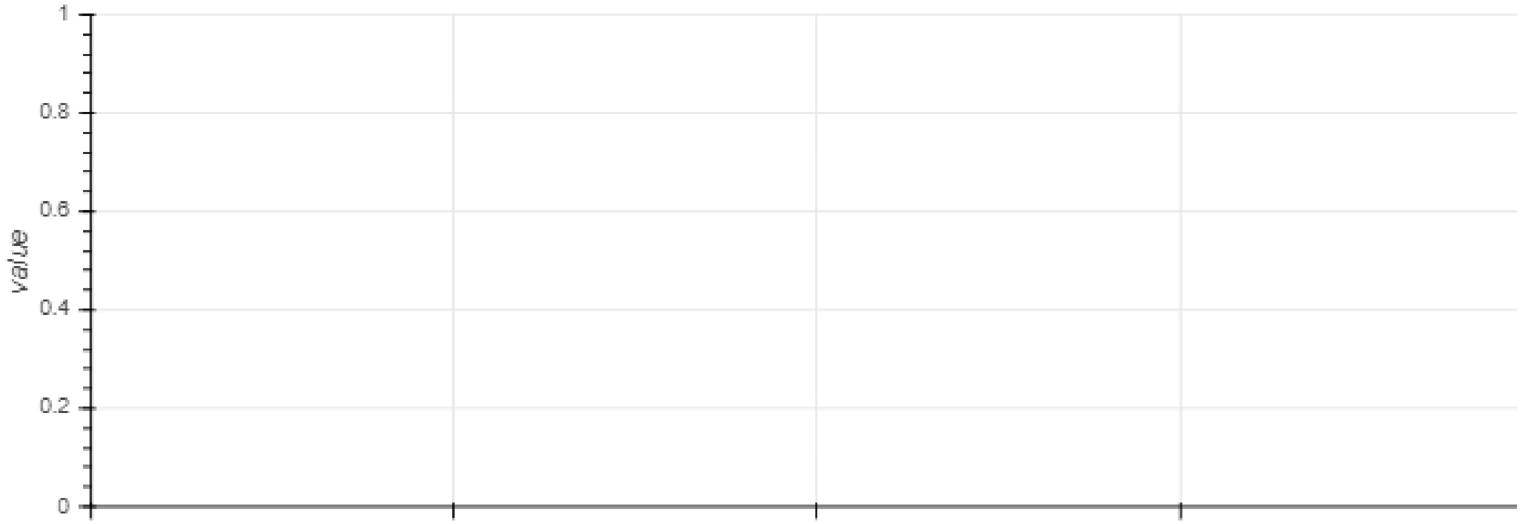
```
# use existing stream for graph
hv.DynamicMap(hv.Scatter, streams=[Buffer(simple_sdf.x)]).redim.label(x='value', index='time'
hv.DynamicMap(hv.Scatter, streams=[Buffer(sdf.x)]).redim.label(x='value', index='time').opts(
dmap = hv.DynamicMap(hv.Dataset, streams=[Buffer(simple_sdf.x, length=500)])
hv.operation.histogram(dmap, dimension='x').opts(fill_alpha=0.75, color=hv.Palette('YlGn'), h
```

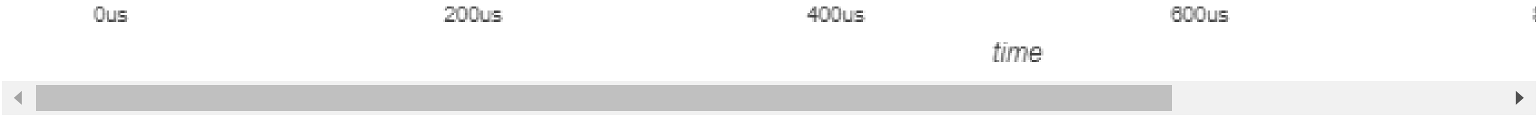
```
executed in 238ms, finished 13:09:54 2019-11-21
```

Out[10]:

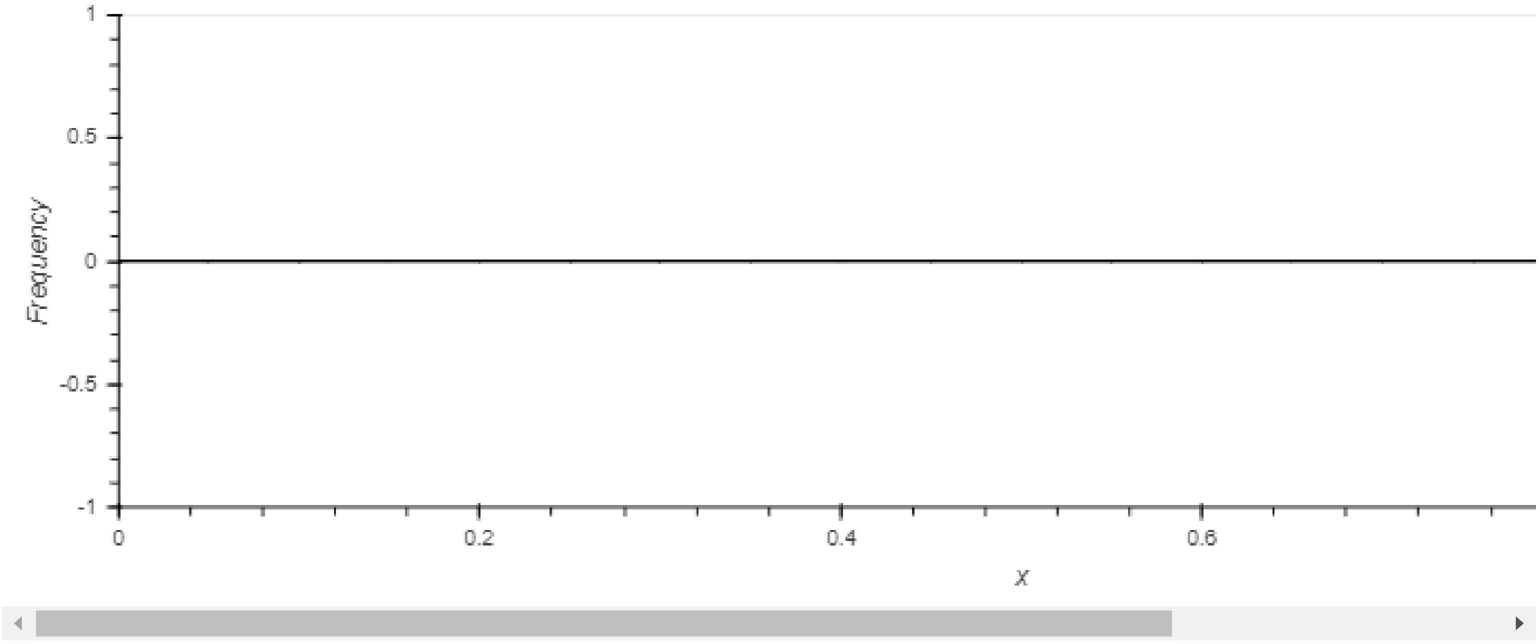


Out[10]:





Out[10]:



Wrap Up

1- Why effective visualizations are important

2- Learned think about the right visual approach

3- Looked at python visualization universe

4- Learned about matplotlib

5- Saw some of Bokeh in action

Acknowledgements

Thanks to **Community Members for feedback & ideas** on this talk

- James Abel
- Stephen McInerney
- Phoebe Polk
- Mark Rice

Thanks to **Data Viz Universe**

- Edward Tufte
- Alberto Cairo
- D3 community

Jupyter Tips

- Mark Roepke (https://www.markroepke.me/posts/2019/06/05/tips-for-slideshows-in-jupyter.html (https://www.markroepke.me/posts/2019/06/05/tips-for-slideshows-in-jupyter.html))

link to Interesting Visualizations & Resources (http://localhost:8888/notebooks/notebooks/04_appendix_inte

In [11]:

```
# stop the stream
# simple_sdf.stop()
# sdf.stop()
```

executed in 3ms, finished 13:09:54 2019-11-21