Bringing ipywidgets Support to plotly.py

Jon Mease

About Me

Education

About Me

Education

Employment

About Me

Education

Millersville University

Johns Hopkins University

Employment

APL Johns Hopkins Applied Physics Laboratory

Open Source Contributions

pandas

DASK

plotly

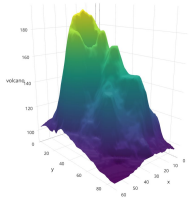
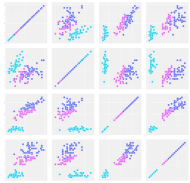
Background

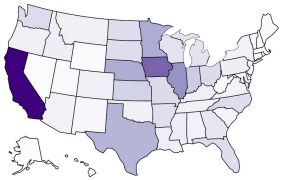
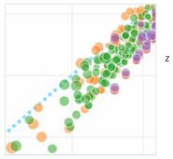
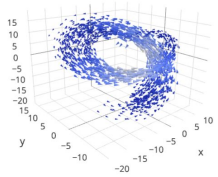
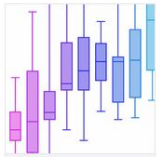
Plotly.js

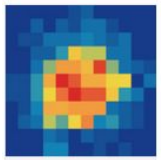
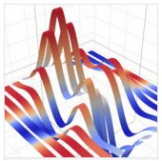
Background

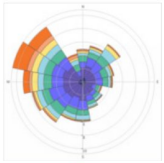
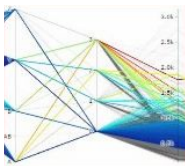
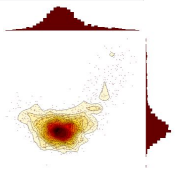
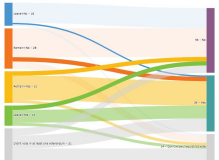
Plotly.js

Background

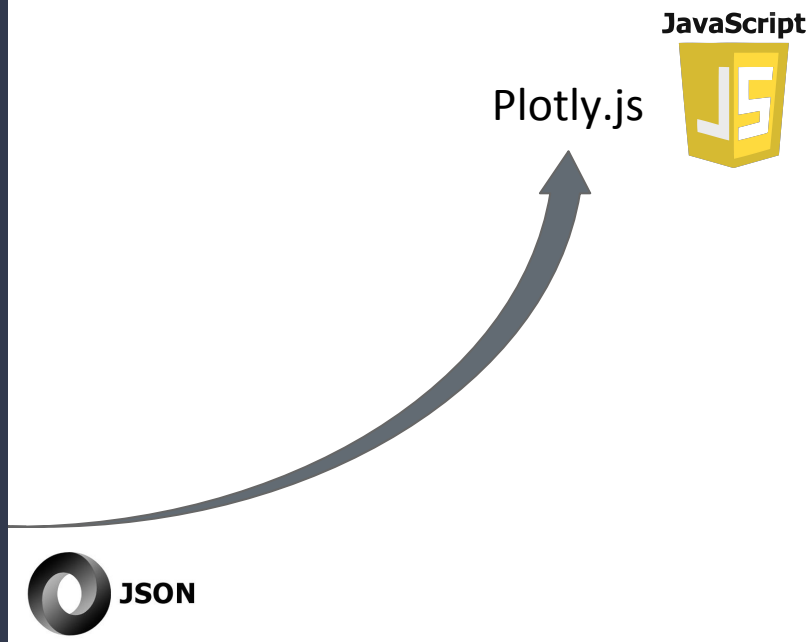






Plotly.js

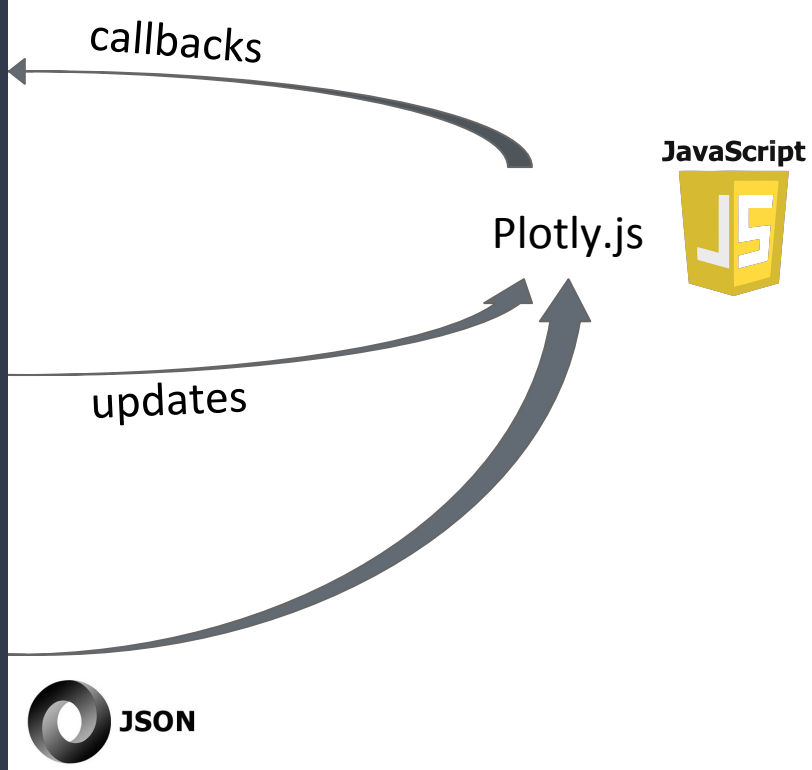
Background

Plotly.js

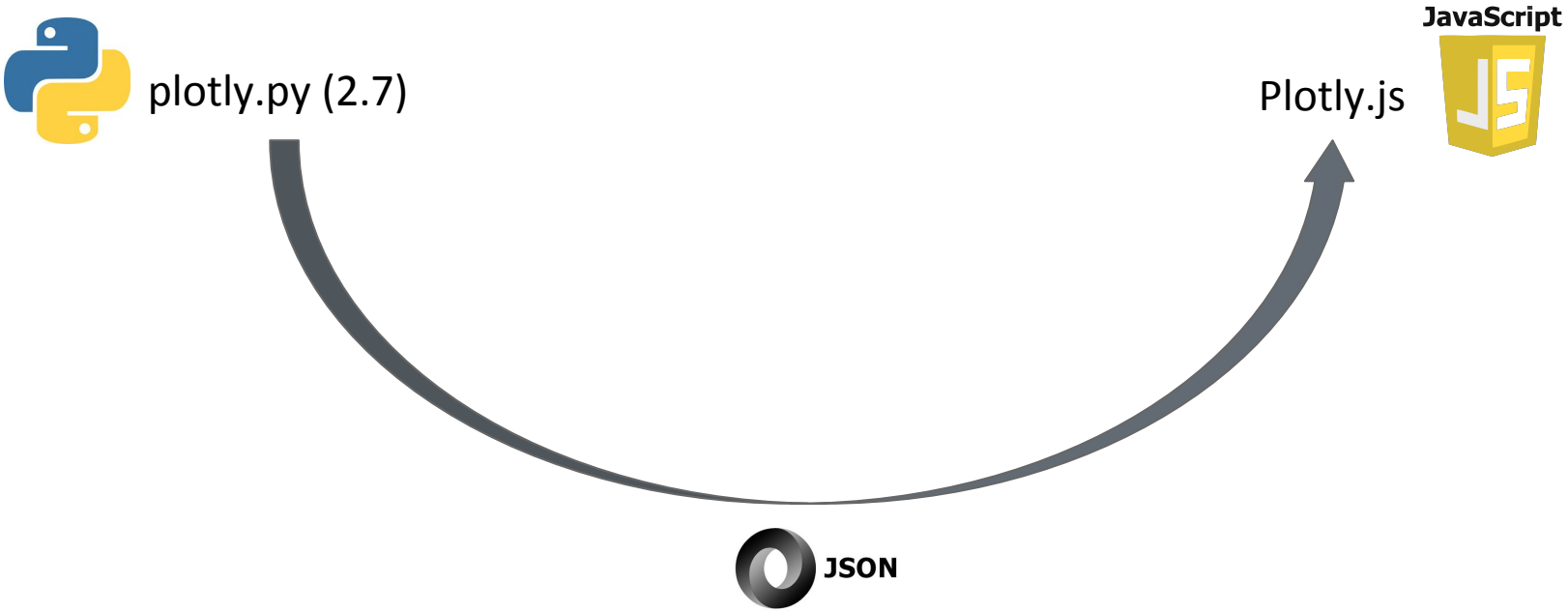
Background



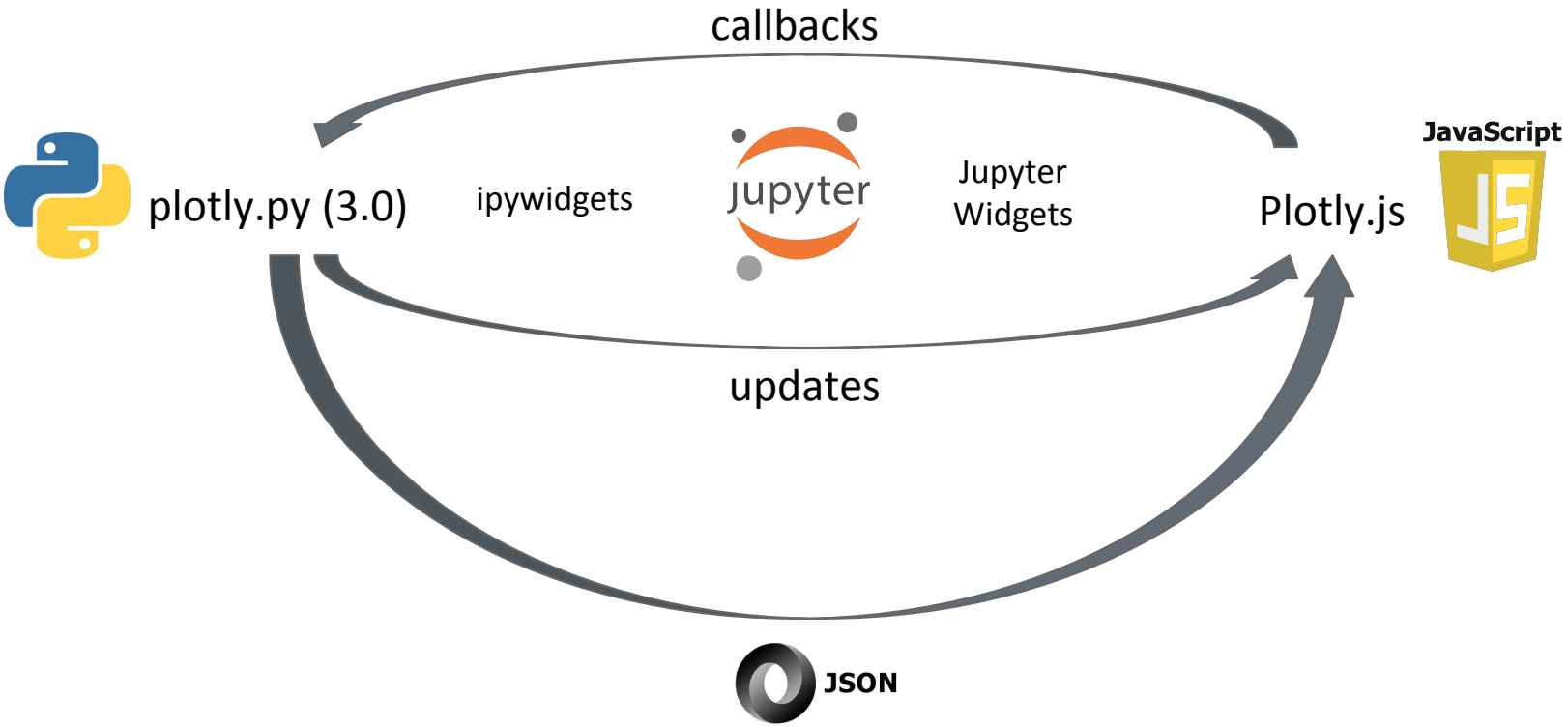
Background



Background



Background


callbacks
plotly.py (3.0)
ipywidgets
jupyter
Jupyter
Widgets
Plotly.js
JavaScript
updates
JSON

Details

Bringing ipywidgets Support to plotly.py

Jon Mease‡*

Abstract—Plotly.js is a declarative JavaScript data visualization library built on D3 and WebGL that supports a wide range of statistical, scientific, financial, geographic, and 3-dimensional visualizations. Support for creating Plotly.js visualizations from Python is provided by the plotly.py library. Version 3 of plotly.py integrates ipywidgets support, providing a host of benefits to plotly.py users working in the Jupyter notebook. This paper describes the architecture of this new version of plotly.py, and presents examples of several of these benefits.

Index Terms—ipywidgets, plotly, jupyter, visualization

Introduction

The Jupyter Notebook [KRKP+16] has emerged as the dominant interface for exploratory data analysis and visualization in the Python data science ecosystem. The ipywidgets library [GFC] provides a suite of interactive widgets for use in the Jupyter Notebook, and it serves as a foundation for library authors to build on to create their own custom widgets.

This paper describes our work to bring ipywidgets support to plotly.py version 3. Compared to version 2, plotly.py version 3 brings plotly.py users working in the Jupyter Notebook a host of benefits. Figures already displayed in the notebook may now be updated in-place using property assignment syntax. All properties throughout the entire figure hierarchy are now discoverable using tab completion and documented with informative docstrings. Property values are now fully validated by the Python library and helpful error messages are raised on validation failures. Figure transitions may now be animated. Numpy arrays are now transferred between the Python and JavaScript libraries using a binary serialization protocol for improved performance. Finally, Python callbacks may now be registered for execution upon zoom, pan, click, hover, and data selection events.

Plotly.js Overview

Plotly.js is a JavaScript data visualization library based on D3 and WebGL that supports a wide range of statistical, scientific, financial, geographic, and 3-dimensional visualizations [Inc15]. The library was initially developed by Plotly Inc. as a core component of their commercial visualization offerings. The library was open sourced under the MIT license in 2015 [Ploc], and may now be used fully offline without requiring any interaction with Plotly Inc's commercial infrastructure.

```
{"data": [
    {"type": "bar",
     "y": [2, 3, 1],
     "name": "A",},
    {"type": "scatter",
     "y": [3, 1, 2],
     "name": "B",
     "marker": {"size": 12}}
    ],
"layout": {"xaxis":
           {"range": [-1, 3],
            "tickvals": [0, 1, 2]}}}
```

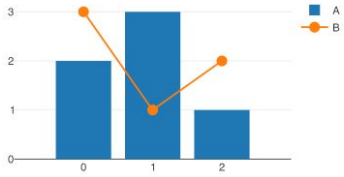


Fig. 1: *JSON specification of a basic Plotly.js figure*

Data model

Plotly figures are fully defined by a declarative JSON specification. Key components of this specification are shown in the example in Figure 1.

The top-level `'data'` property contains an array of the traces present in the figure. The object representing each trace contains a `'type'` property that identifies the trace type (e.g. `'scatter'`, `'bar'`, `'violin'`, `'mesh3d'`, etc.). The remaining properties are used to configure the trace. As of version 1.37.1, Plotly.js supports 32 distinct trace types covering many statistical, scientific, financial, geographic, and 3-dimensional use-cases.

The top-level `'layout'` property is an object with properties that specify characteristics of the figure that are independent of its

* *Corresponding author: jon.mease@jhuapl.edu*
‡ *Johns Hopkins Applied Physics Laboratory*

New in 3.0.0

ipywidgets integration

New in 3.0.0

ipywidgets integration

Imperative API for in-place updates

New in 3.0.0

ipywidgets integration

Imperative API for in-place updates

Great auto-completion

Full inline documentation

New in 3.0.0

ipywidgets integration

Imperative API for in-place updates

Great auto-completion

Full inline documentation

Robust property validation

Helpful error messages

New in 3.0.0

ipywidgets integration

Imperative API for in-place updates

Great auto-completion

Full inline documentation

Robust property validation

Helpful error messages

Python callbacks

New in 3.0.0

ipywidgets integration

Imperative API for in-place updates

Great auto-completion

Full inline documentation

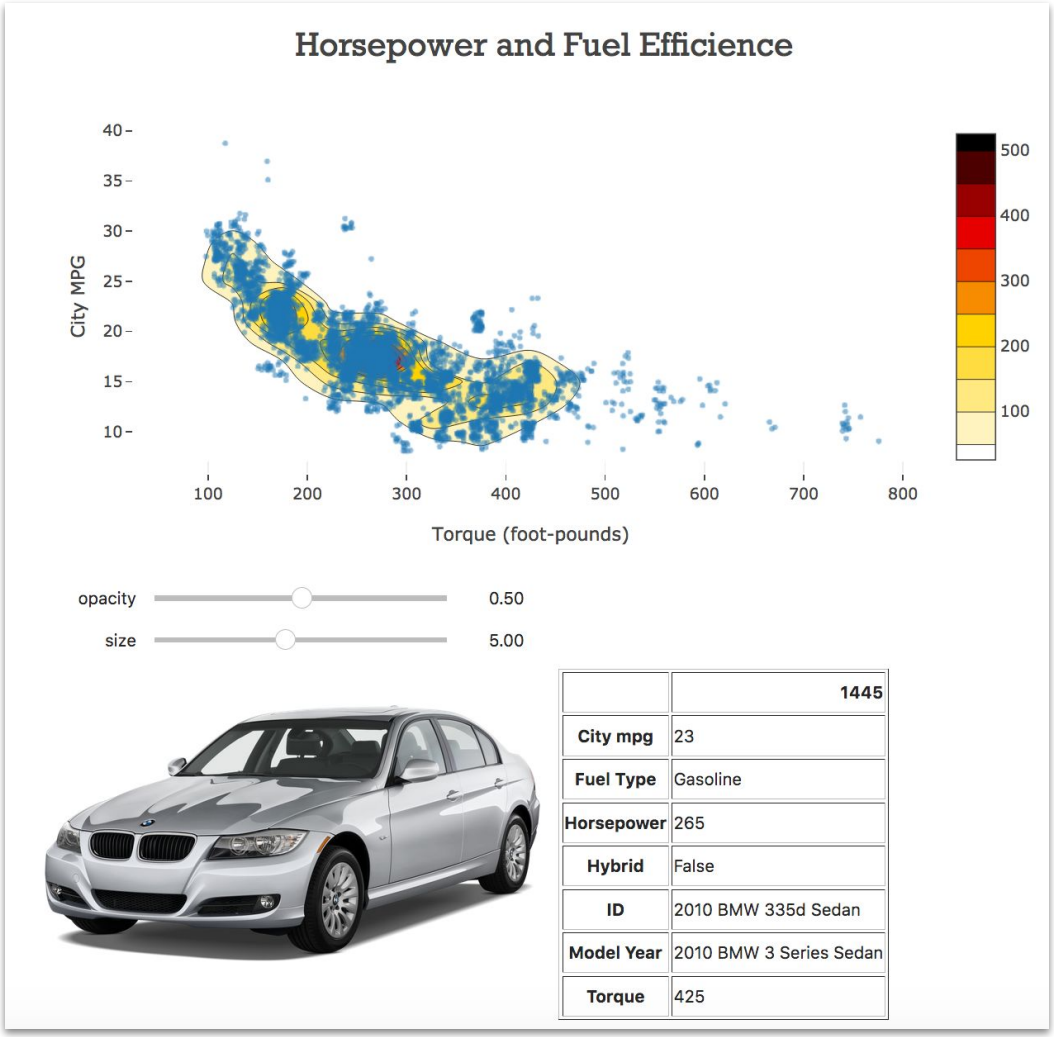
Robust property validation

Helpful error messages

Python callbacks

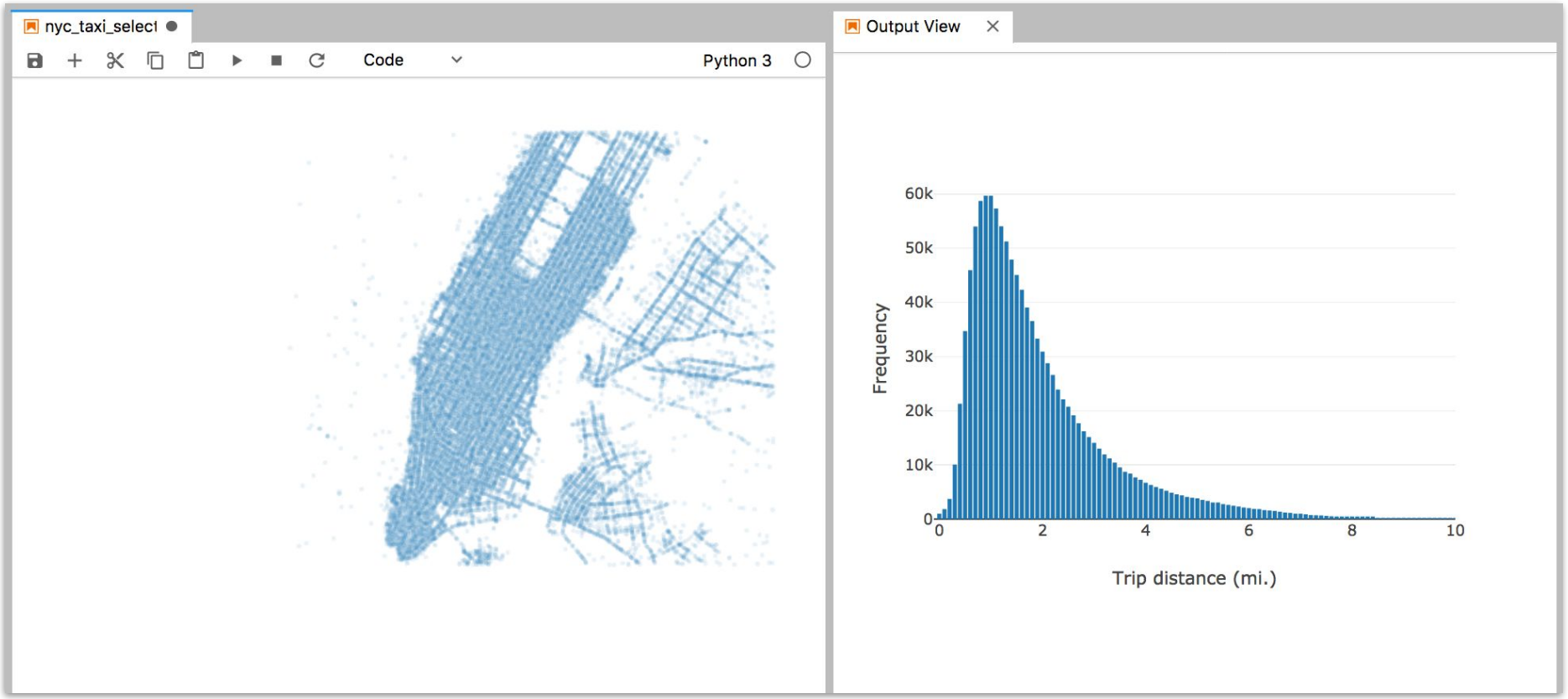
Performance optimizations

Cars Demo



	1445
City mpg	23
Fuel Type	Gasoline
Horsepower	265
Hybrid	False
ID	2010 BMW 335d Sedan
Model Year	2010 BMW 3 Series Sedan
Torque	425

NYC Taxi Demo



Want to Learn More?

Pinned by **@jonmmease** on 

- 3.0 announcement post
- Proceedings
- Example notebooks
- Documentation
- Webinar on July 25th

Other widget DataViz talks:

- bqplot, ipyvolume

Come find me at our table:

Thanks!