



Australian Government

Bureau of Meteorology

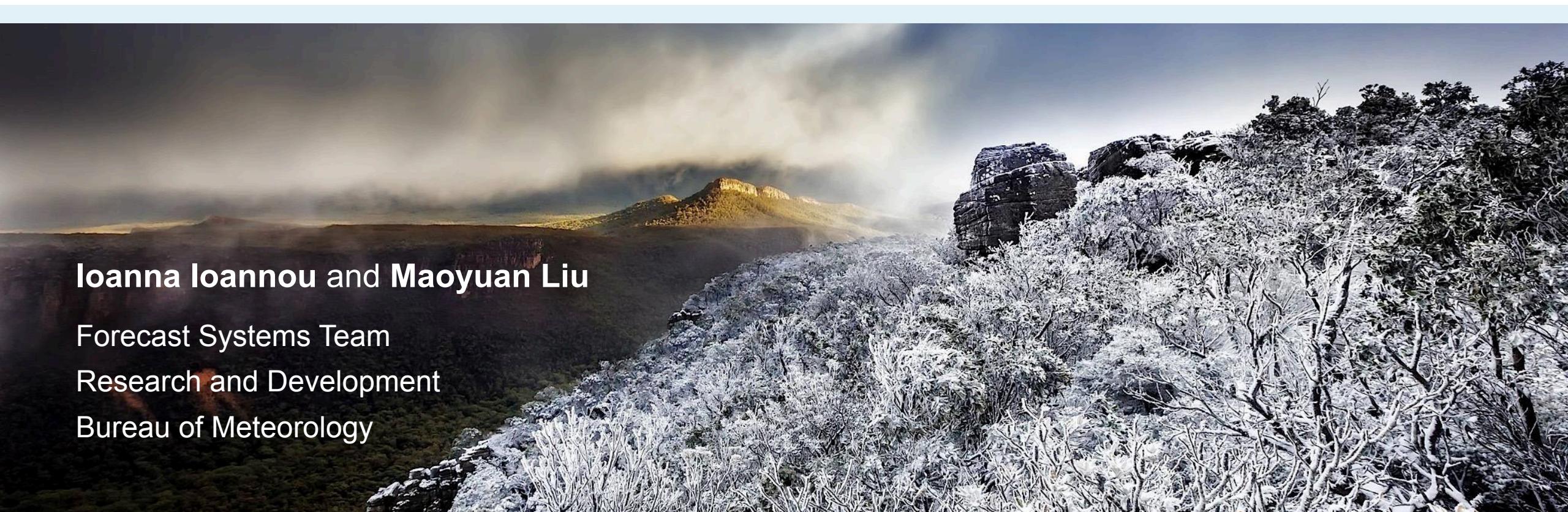
Interactive data displays with Jupyter notebooks

Ioanna Ioannou and Maoyuan Liu

Forecast Systems Team

Research and Development

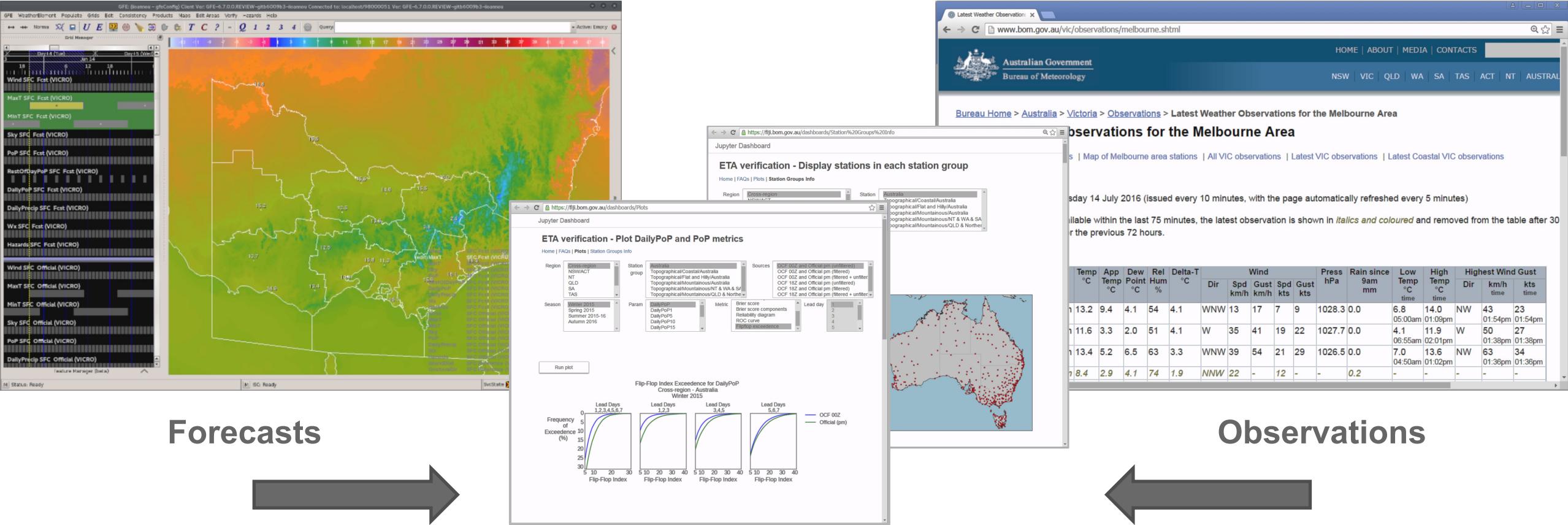
Bureau of Meteorology





Australian Government
Bureau of Meteorology

A bit about us



Forecasts



Interactive verification dashboards

Observations



Why use notebooks?

- We are a multi-disciplinary team: scientists, programmers, managers
- Excellent facilities for plotting/visualisation
- No need to know anything about web development
- Very popular tool for scientific data analysis and exploration

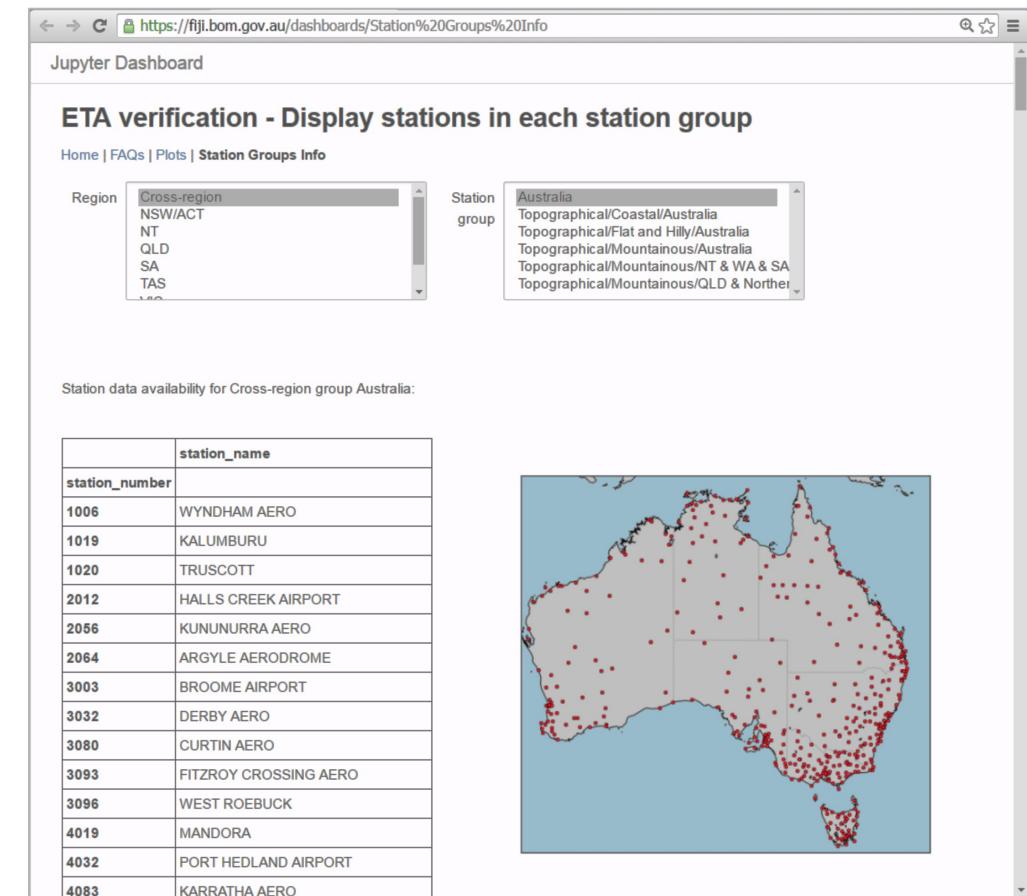
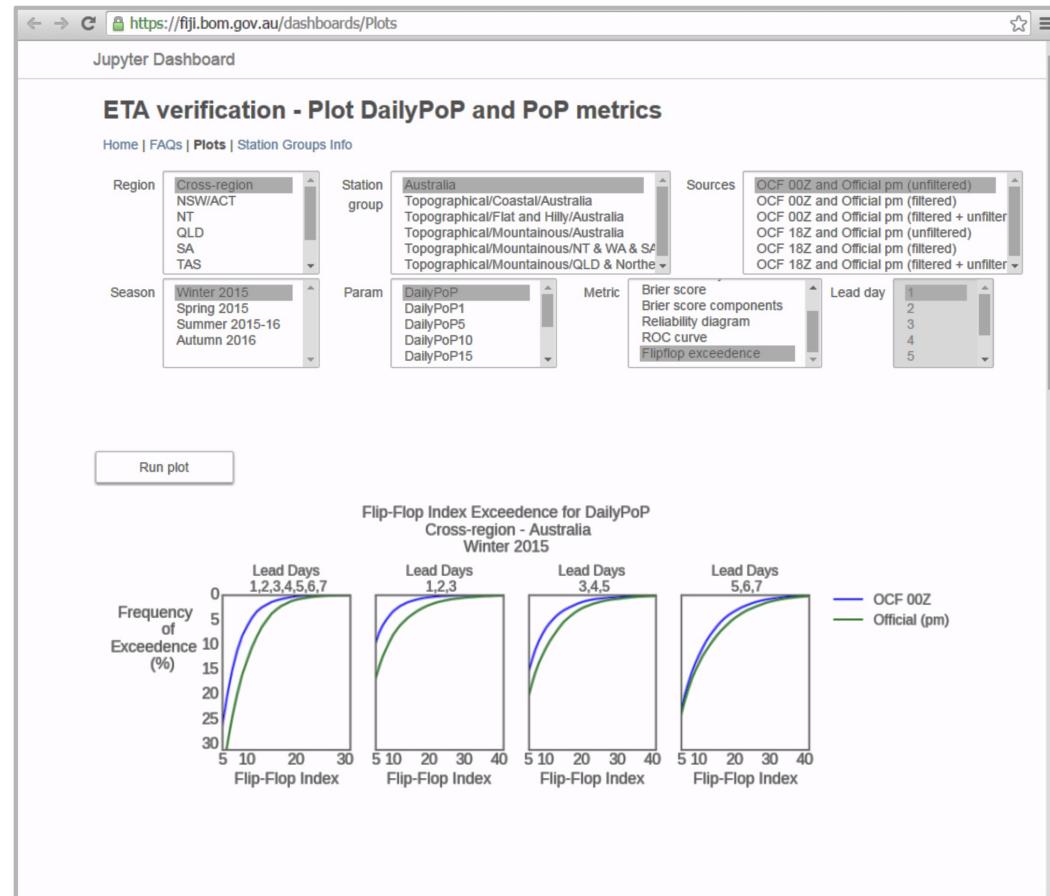




Australian Government

Bureau of Meteorology

Why interactive data displays?



Interactive plotting technologies

- Requirements:
 - Browser interactivity
 - Dynamic data selection/manipulation
 - Ease of use - minimize need to learn Javascript/HTML/CSS
 - Python callbacks



Australian Government

Bureau of Meteorology

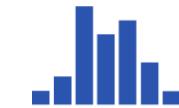
Plotting technologies



matplotlib



Bokeh



plotly

http://localhost:8000/user/root/notebooks/PyCon/1_Plottting_examples.ipynb

How do they compare?

Library	Browser interactivity	Dynamic data selection	Easy of use	Python callbacks
Matplotlib	★	★	★★	
Bokeh	★★	★★	★	ipywidgets
Plotly	★★★	★★★	★★★	



Australian Government

Bureau of Meteorology

ipywidgets

Suite of UI widgets for ipython notebooks

Loading:

Text:

Number: ▾

Speed:

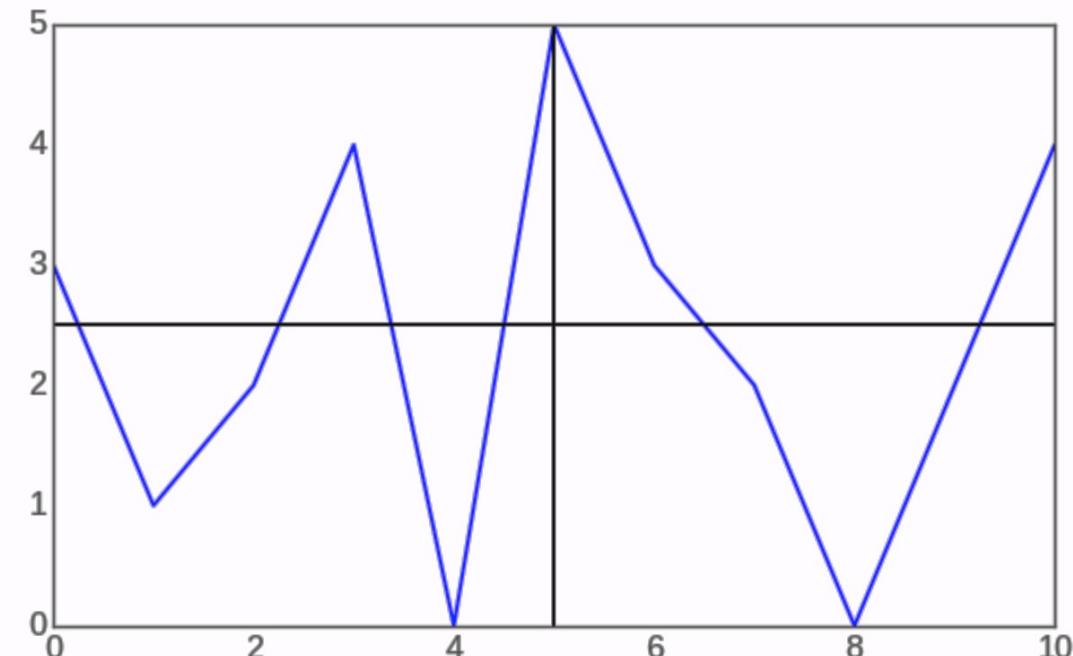
Fruits

- Apples
- Oranges
- Pears

Pizza topping:

- pepperoni
- pineapple
- anchovies

x
y





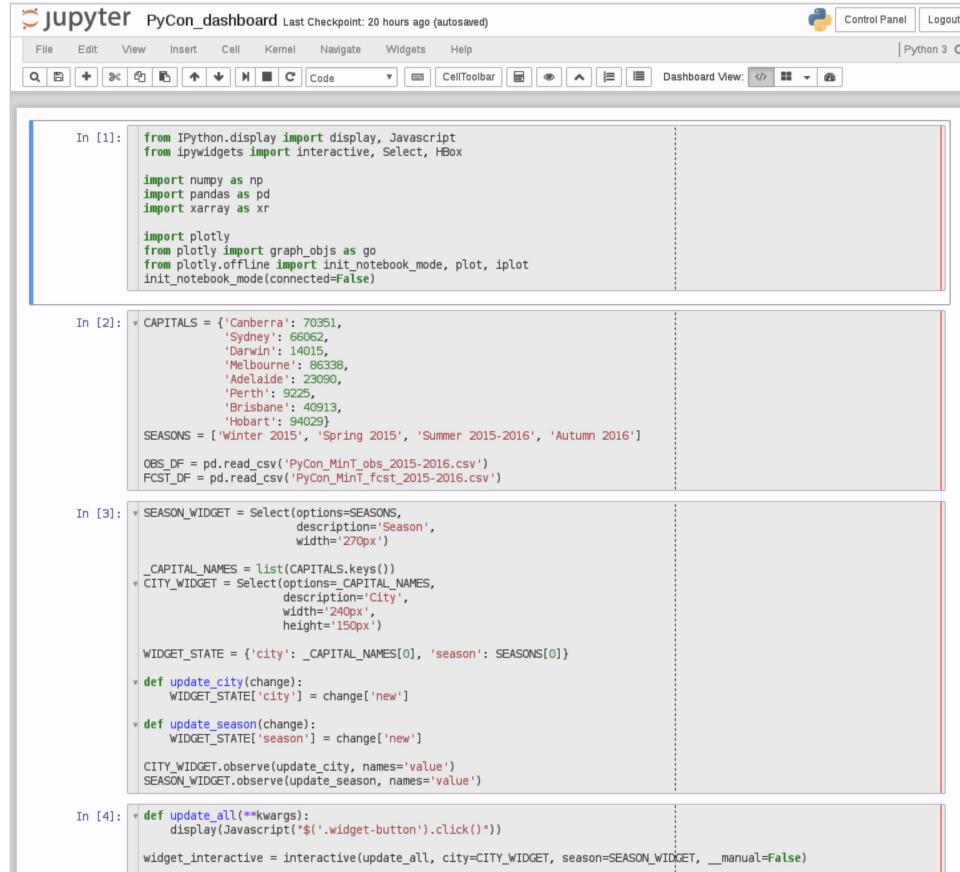
Australian Government

Bureau of Meteorology

SHARING NOTEBOOKS AS DASHBOARDS

With Mao

Sharing Notebooks can be problematic



```
jupyter PyCon_dashboard Last Checkpoint: 20 hours ago (autosaved)
Control Panel Logout Python 3

File Edit View Insert Cell Kernel Navigate Widgets Help
Dashboard View: <> << >> >>

In [1]: from IPython.display import display, Javascript
from ipywidgets import interactive, Select, HBox

import numpy as np
import pandas as pd
import xarray as xr

import plotly
from plotly import graph_objs as go
from plotly.offline import init_notebook_mode, plot, iplot
init_notebook_mode(connected=False)

In [2]: CAPITALS = {'Canberra': 70351,
                 'Sydney': 66062,
                 'Darwin': 14015,
                 'Melbourne': 86338,
                 'Adelaide': 22090,
                 'Perth': 9225,
                 'Brisbane': 40913,
                 'Hobart': 94029}
SEASONS = ['Winter 2015', 'Spring 2015', 'Summer 2015-2016', 'Autumn 2016']

OBS_DF = pd.read_csv('PyCon_MinT_obs_2015-2016.csv')
FCST_DF = pd.read_csv('PyCon_MinT_fcst_2015-2016.csv')

In [3]: SEASON_WIDGET = Select(options=SEASONS,
                           description='Season',
                           width='270px')

_CAPITAL_NAMES = list(CAPITALS.keys())
CITY_WIDGET = Select(options=_CAPITAL_NAMES,
                     description='City',
                     width='240px',
                     height='150px')

WIDGET_STATE = {'city': _CAPITAL_NAMES[0], 'season': SEASONS[0]}

def update_city(change):
    WIDGET_STATE['city'] = change['new']

def update_season(change):
    WIDGET_STATE['season'] = change['new']

CITY_WIDGET.observe(update_city, names='value')
SEASON_WIDGET.observe(update_season, names='value')

In [4]: def update_all(**kwargs):
    display(Javascript("$('.widget-button').click()"))

    widget_interactive = interactive(update_all, city=CITY_WIDGET, season=SEASON_WIDGET, __manual=False)
```

Sharing a file:

- Users may need to execute code cells before the results are visible
- Code dominates the screenspace
- Code may be accidentally modified

Sharing a URL:

- **Code execution kernel is EXPOSED**
- Authentication+SSL is a must
 - Single jupyter notebook instance with shared password (very bad)
 - JupyterHub with system accounts (not as bad)



Australian Government

Bureau of Meteorology

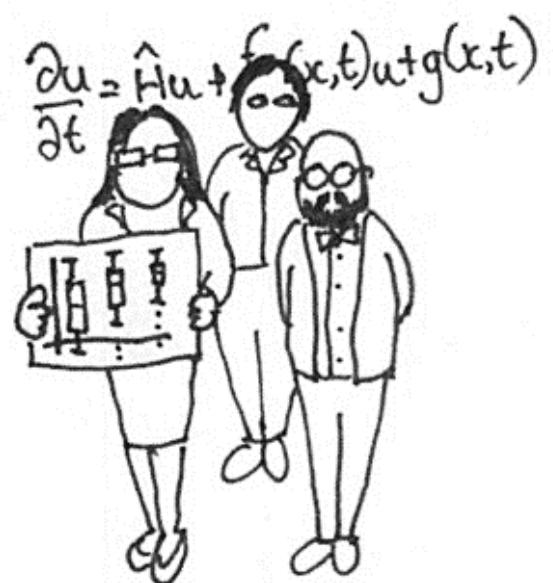
Why dashboards?



FORECASTERS



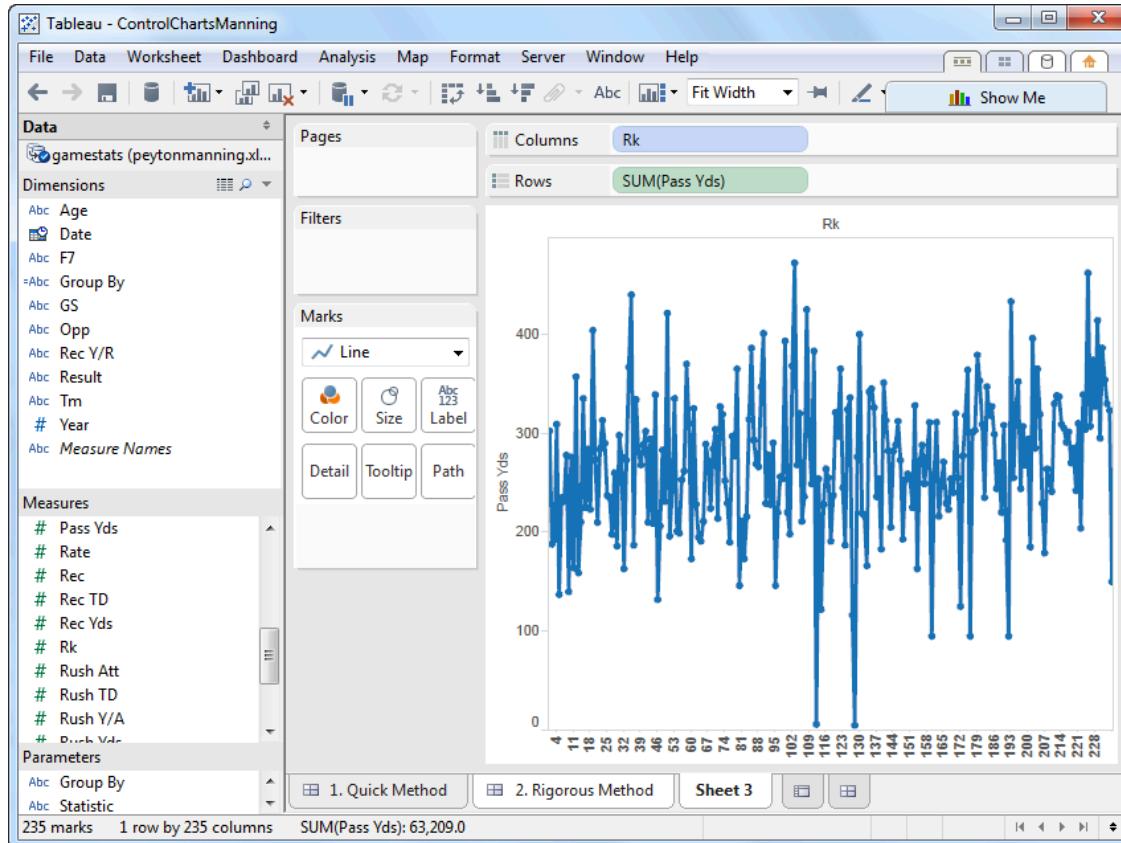
MANAGERS



SCIENTISTS

Data consumers are not necessarily programmers!

Proprietary dashboards



- Adopt the UI and features of the provider
- \$\$\$
- Scripting options? Git?
- Surely there is an alternative



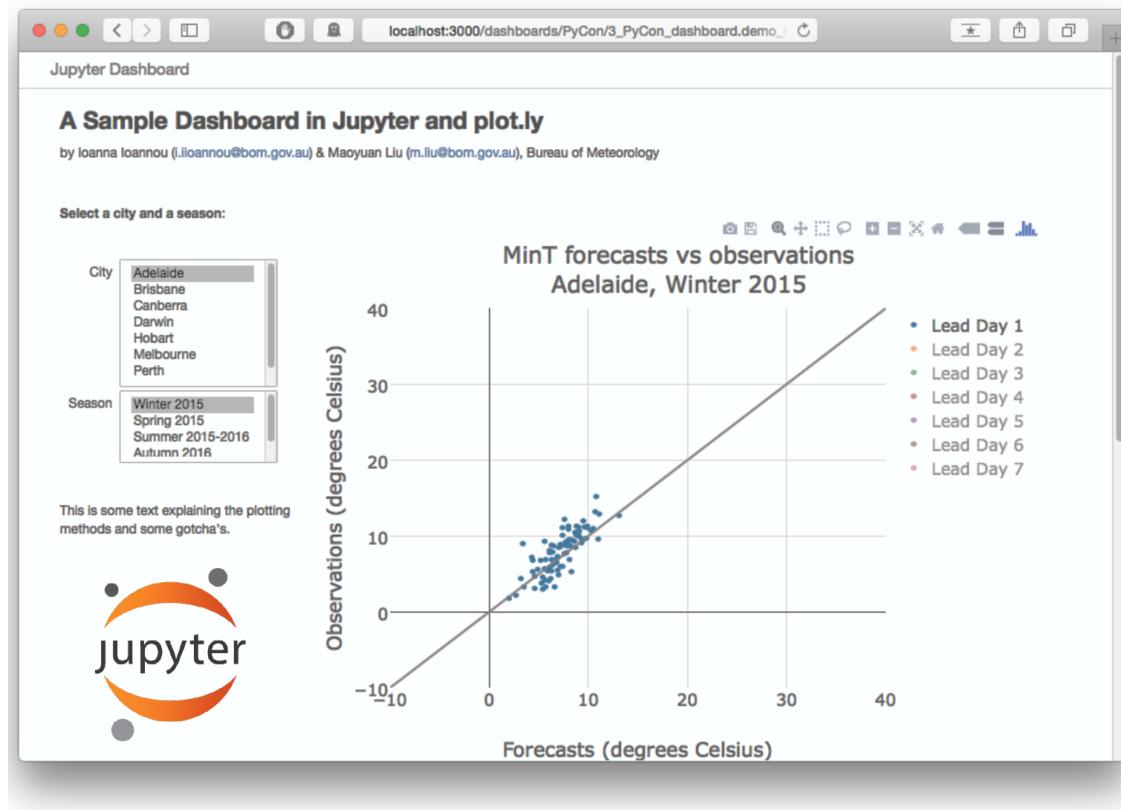
Klipfolio™



GECKOBOARD

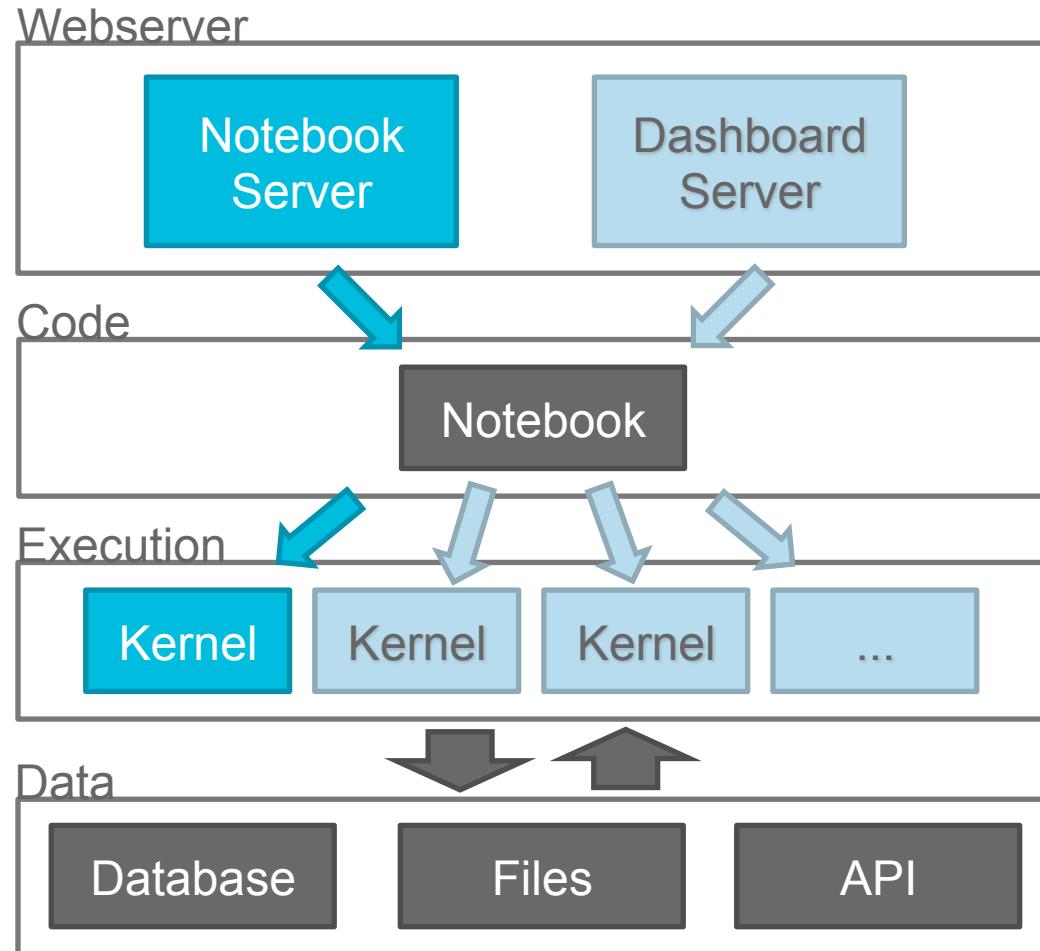


Jupyter Dashboard



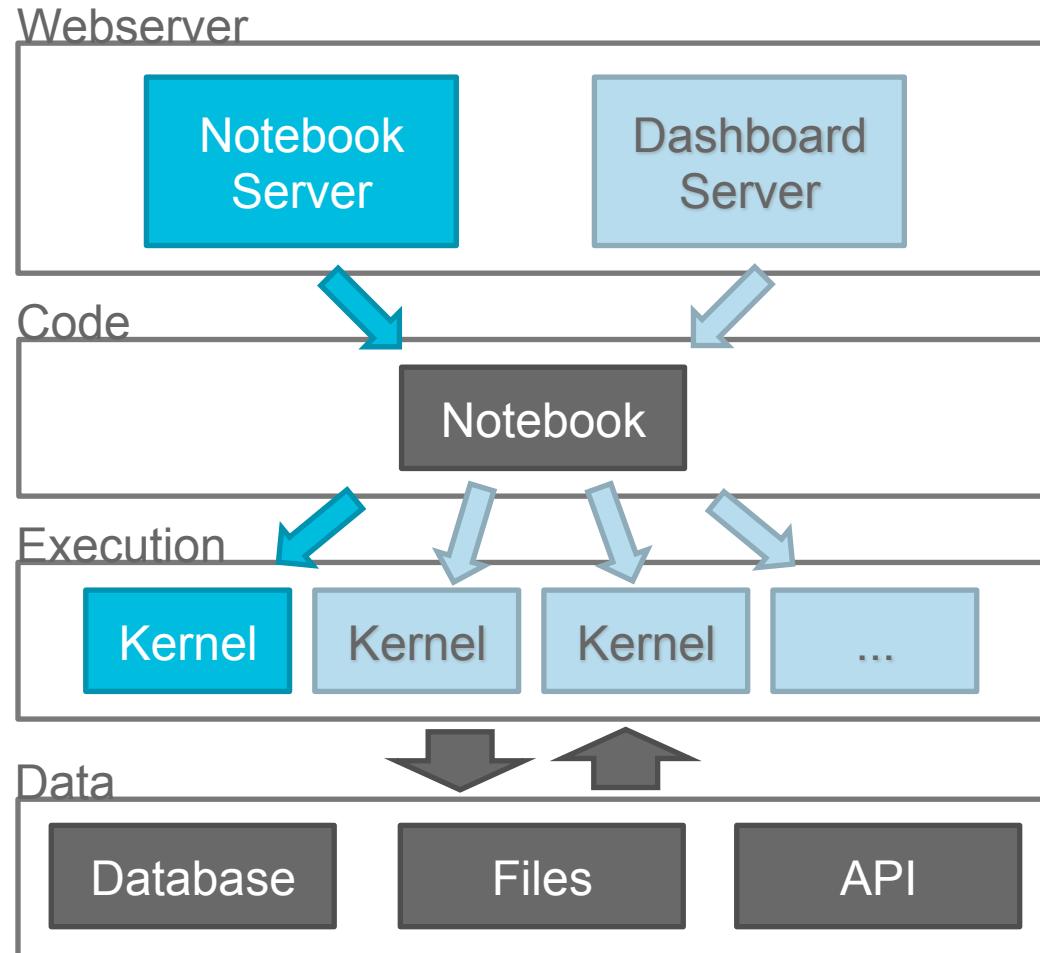
- Serves a notebook file **as is**
 - Executes code cells on load
 - Some control over layout
- Callback functions are in Python – NO JS!
- Does not expose code
- Does not expose kernel
- Open-source:
 - https://github.com/jupyter-incubator/dashboards_server
 - <https://github.com/jupyter-incubator/dashboards>
 - https://github.com/jupyter/kernel_gateway

Jupyter “Stack”



- **Notebook server**
 - For developers
 - Interactive code execution environment
 - Requires authentication
- **Dashboard server**
 - For publication
 - Executes notebooks, but cannot inject code to the kernel
 - Each visitor gets a new kernel, re-executes the notebook

Jupyter “Stack”



- Both access the same notebooks and underlying data
- Fast prototyping to publication

Caveats:

- Limited concurrency if notebooks are memory/CPU hungry
- **NOT** a replacement for traditional websites

Dashboard example

Notebook:

http://localhost:8000/user/root/notebooks/PyCon/2_PyCon_dashboard.demo.ipynb

Dashboard:

http://localhost:3000/dashboards/PyCon/2_PyCon_dashboard.demo

GitHub:

https://github.com/ch41rmn/PyConAU2016_-_Interactive_Data_Displays_With_Jupyter_Notebooks



Australian Government

Bureau of Meteorology

Thank you.