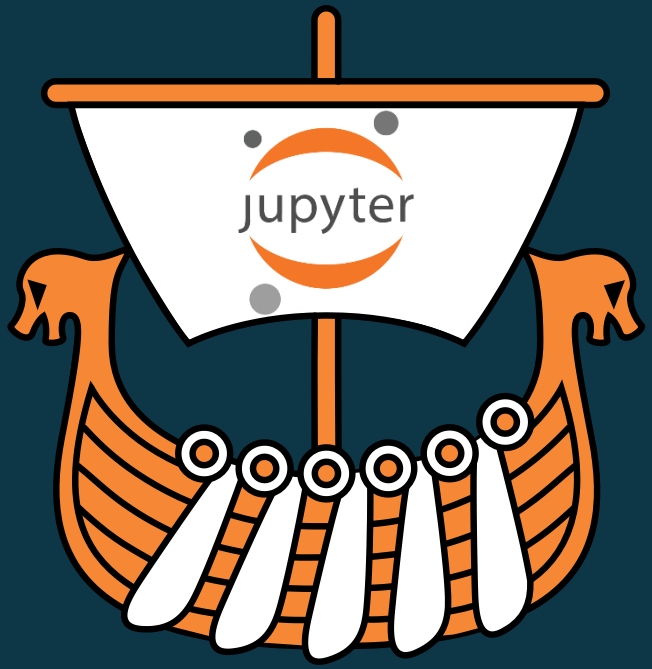


DYALOG

Belfast 2018

Jupyter Notebooks

Adám Brudzewsky

What are notebooks?

A *notebook* combines the functionality of

- a word processor — handles formatted text
- a *shell* or *kernel* — executes statements in a programming language and includes output inline
- a rendering engine — renders HTML in addition to plain text

Example notebook

using Python

global density of metal bands

Creating a world map of

Not secure | ramiro.org/notebook/metal-bands-map/

Apps APL EKGL kdb - Interprocess Co The APL Orchard | ch

Plot the map

We'll use the handy `plot` method available on `GeoDataFrame` objects. To make sure the map shows all countries, including those without data on metal bands, we have to plot these two sets separately. If you like to learn why, check out this notebook on creating choropleth maps using GeoPandas.

In the final code section, we create two separate data frames `known` and `unknown`. The `known` countries will be plotted using a colormap that seemed appropriate and the Jenks classification method, that reduces the variance within classes and maximize the variance between classes. There will be `9` different classes with darker colors indicating higher band ratios.

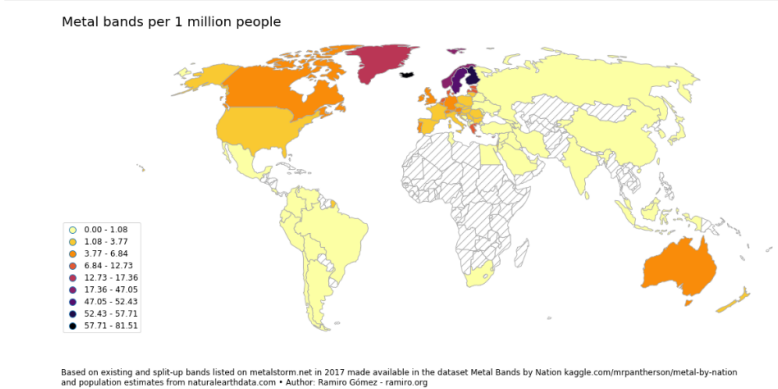
The `unknown` countries will be shown with a white background and a striped pattern. We also add some descriptive text, move the legend to the lower left part of the map and set the legend's size.

```
known = world.dropna(subset=['band_ratio'])
unknown = world[world['band_ratio'].isna()]

ax = known.plot(column='band_ratio', cmap='inferno_r', figsize=(20, 12), scheme='fisher_jenks', k=9, legend=True, edgecolor='#aaaaaa')
unknown.plot(ax=ax, color='#ffffff', hatch='//', edgecolor='#aaaaaa')

ax.set_title('Metal bands per 1 million people', fontdict={'fontsize': 20}, loc='left')
description = '''
Based on existing and split-up bands listed on metalstorm.net in 2017 made available in the dataset Metal Bands by Nation kaggle.com/mrpantherso
and population estimates from naturalearthdata.com • Author: Ramiro Gómez - ramiro.org'''.strip()
ax.annotate(description, xy=(0.07, 0.1), size=12, xycoords='figure fraction')

ax.set_axis_off()
legend = ax.get_legend()
legend.set_bbox_to_anchor((.11, .4))
legend.prop.set_size(12)
```



Based on existing and split-up bands listed on metalstorm.net in 2017 made available in the dataset Metal Bands by Nation kaggle.com/mrpantherson/metal-by-nation and population estimates from naturalearthdata.com • Author: Ramiro Gómez - ramiro.org

Conclusion

The map above and the one posted on reddit six years ago show similar patterns regarding regions with high and low metal band ratios. Moreover, it is obvious that our dataset comprises less countries and, looking at the actual numbers, has a lot less records in total.

Exa

using

globa
of m

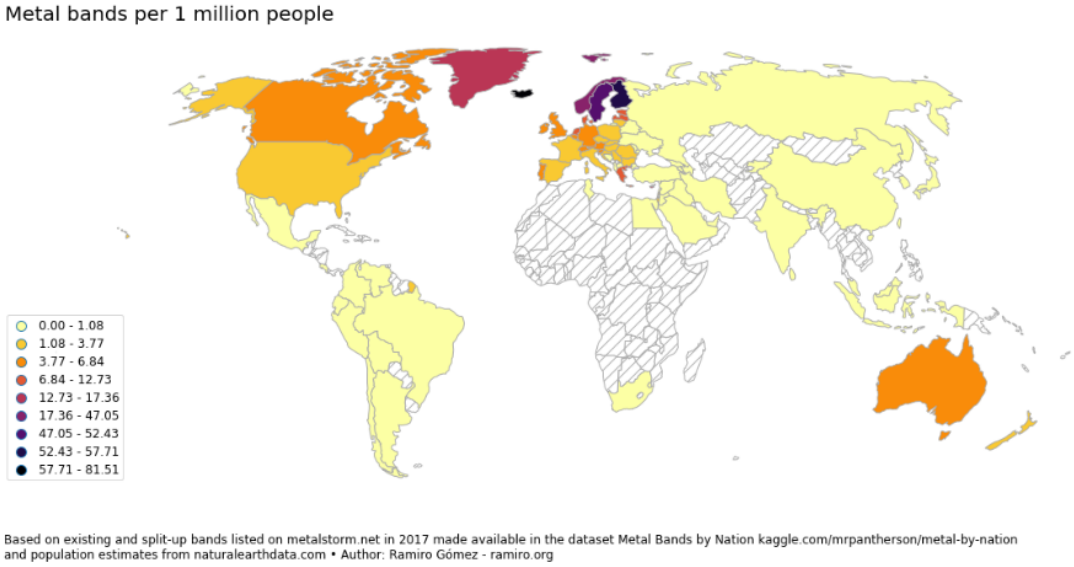
part of the map and set the legend's size.

```
known = world.dropna(subset=['band_ratio'])
unknown = world[world['band_ratio'].isna()]

ax = known.plot(column='band_ratio', cmap='inferno_r', figsize=(20, 12), scheme='fisher_jenks', k=9, legend=True, edgecolor='#aaaaaa')
unknown.plot(ax=ax, color='#ffffff', hatch='//', edgecolor='#aaaaaa')

ax.set_title('Metal bands per 1 million people', fontdict={'fontsize': 20}, loc='left')
description = '''
Based on existing and split-up bands listed on metalstorm.net in 2017 made available in the dataset Metal Bands by Nation kaggle.com/mrpantherso
and population estimates from naturalearthdata.com • Author: Ramiro Gómez - ramiro.org'''.strip()
ax.annotate(description, xy=(0.07, 0.1), size=12, xycoords='figure fraction')

ax.set_axis_off()
legend = ax.get_legend()
legend.set_bbox_to_anchor((.11, .4))
legend.prop.set_size(12)
```



Conclusion

Example notebook

using Dyalog APL

health care expenditure vs GDP per capita

Health Care Spending

localhost:8888/notebooks/Documents/Dyalog/Jupyter%20Notebooks...

File Edit View Insert Cell Kernel Widgets Help Trusted Dyalog APL

Cleaning the Data

Apparently there's no GDP or health expenditure data for Antarctica... Our next step is to filter out any records that have "null" entries anywhere, remove countries with per capita GDP < 25,000 - each time displaying the resulting item count:

In [22]:
```
≠tab←(~∨/tab='null')⌿tab      ⍝ remove countries with null data
≠tab←(tab[;3]≥25000)⌿tab      ⍝ remove counties with per capita GDP<25000
```

Out[22]: 191

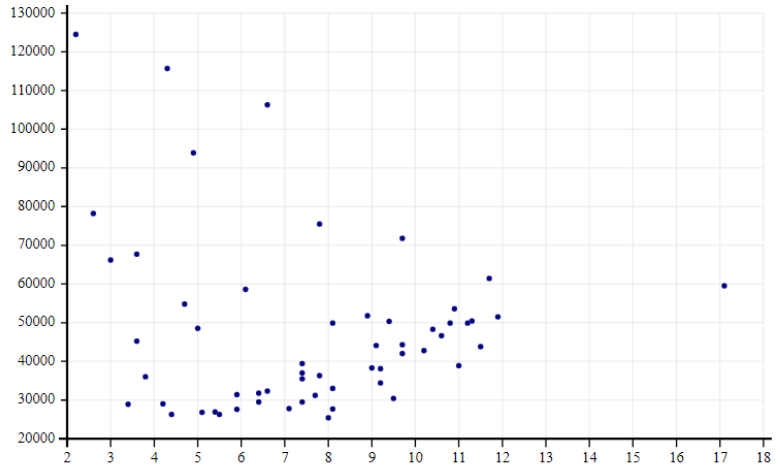
Out[22]: 57

Charting

Finally, we create our scatterplot showing Health Care Expenditure as a percentage of GDP versus per capita GDP:

In [23]:
```
InitCauseway ⍬
sp←⎕NEW Causeway.SharpPlot
sp.SetMarkers ⊂,Causeway.Marker.Bullet
sp.ScatterPlotStyle←Causeway.ScatterPlotStyles.ValueTags
sp.ValueTagStyle←Causeway.ValueTagStyles.Tips
sp.SetValueTags ⊂tab[;1]
sp.XAxisStyle←Causeway.XAxisStyles.GridLines
sp.YAxisStyle←Causeway.YAxisStyles.GridLines
sp.DrawScatterPlot (,⊂tab[;3])(tab[;2])
{⍵}3500⌶sp.RenderSvg Causeway.SvgMode.FixedAspect
```

Out[23]:



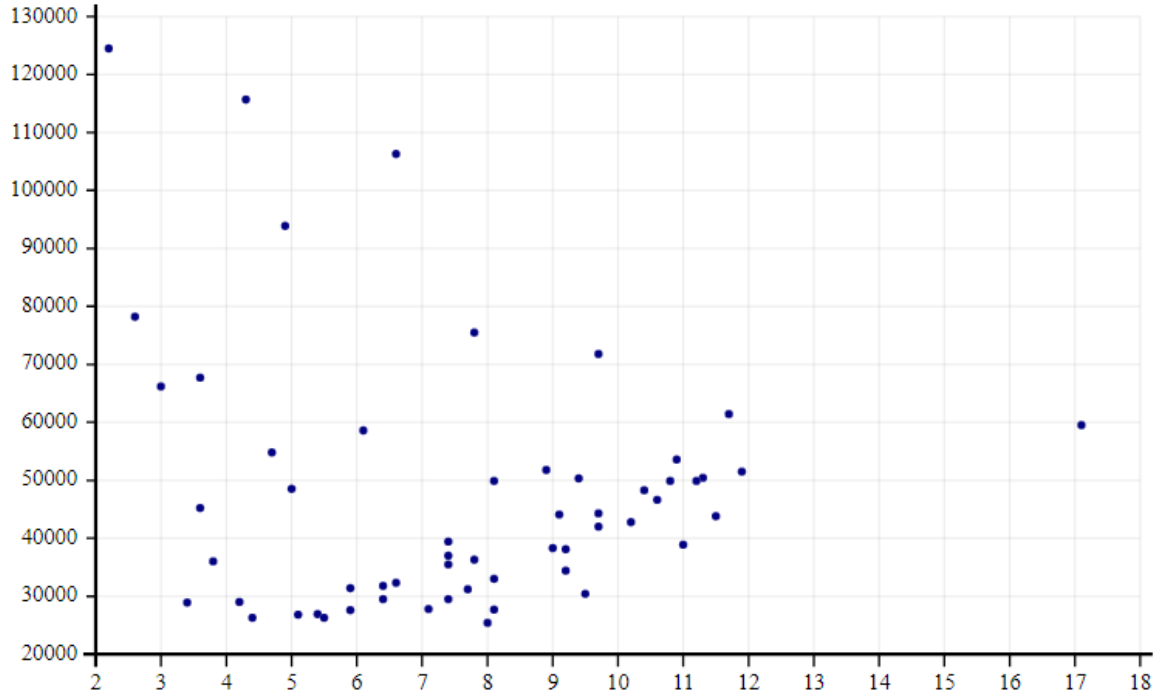
Exa

usin

hea
vs G

In [23]:
```
InitCauseway ⍬
sp←⎕NEW Causeway.SharpPlot
sp.SetMarkers c,Causeway.Marker.Bullet
sp.ScatterPlotStyle←Causeway.ScatterPlotStyles.ValueTags
sp.ValueTagStyle←Causeway.ValueTagStyles.Tips
sp.SetValueTags ctab[;1]
sp.XAxisStyle←Causeway.XAxisStyles.GridLines
sp.YAxisStyle←Causeway.YAxisStyles.GridLines
sp.DrawScatterPlot (,ctab[;3])(tab[;2])
{}3500⌶sp.RenderSvg Causeway.SvgMode.FixedAspect
```

Out[23]:



Notebook benefits

A single document that combines explanations with executable code and its output — an ideal way to provide:

- reproducible research results
- documentation of processes
- instructions
- tutorials and training materials of all shapes and sizes

A digital learning environment for computational thinking

What is *Jupyter* notebook?

First notebook: Mathematica 1.0 in '88

Jupyter notebook is a part of

Project Jupyter, a nonprofit to

> *develop open-source software, standards, and services for interactive computing across dozens of programming languages*

beginning with **Ju**lia, **Pyt**hon, **R**, and now over 70 languages, including Dyalog APL

What is *Jupyter* notebook?

First notebook: Mathematica 1.0 in '88

Jupyter notebook is a part of

Project Jupyter, a nonprofit to

> *develop open-source software, standards, and services for interactive computing across dozens of programming languages*

beginning with **Ju**lia, **Pyt**hon, **R**, and now over 70 languages, including Dyalog APL

Ways to use Jupyter notebooks

On your own PC after installing a Jupyter notebook server

With an online notebook server like cocalc.com

Save notebook with output and use a notebook viewer

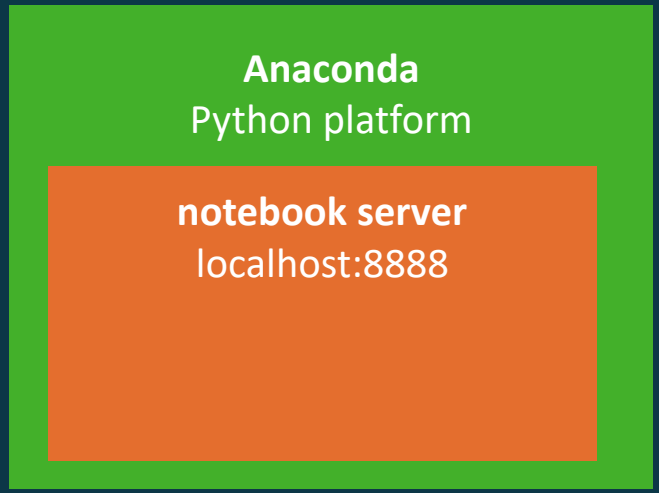
Export to HTML, PDF, LaTeX, etc.

Local notebook server — Python

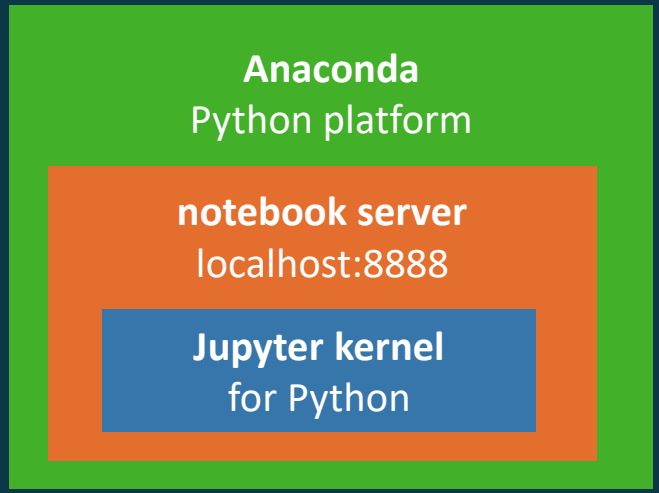
Local notebook server — Python

Anaconda
Python platform

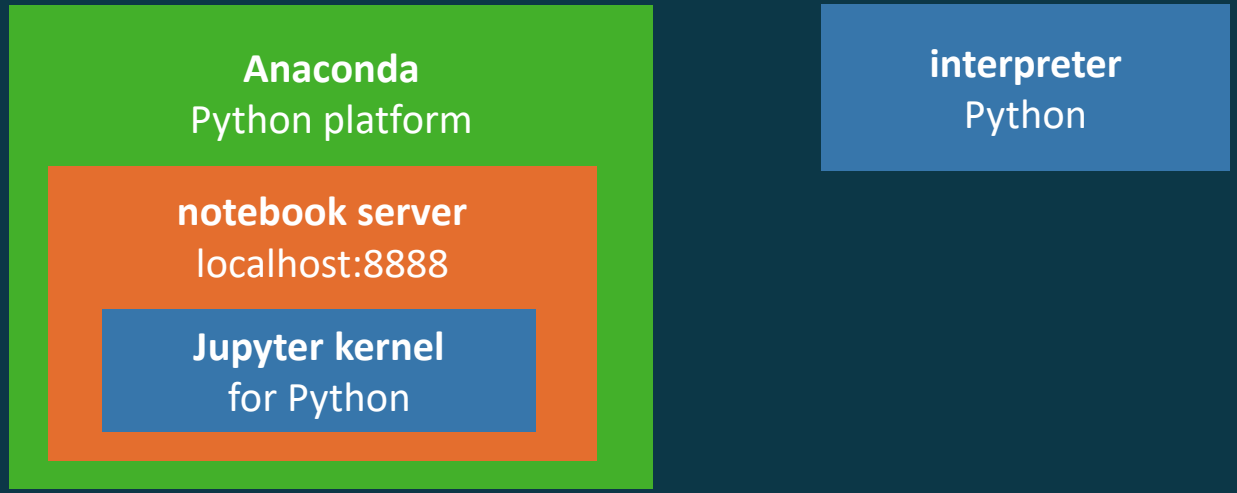
Local notebook server — Python



Local notebook server — Python



Local notebook server — Python



Local notebook server — Python

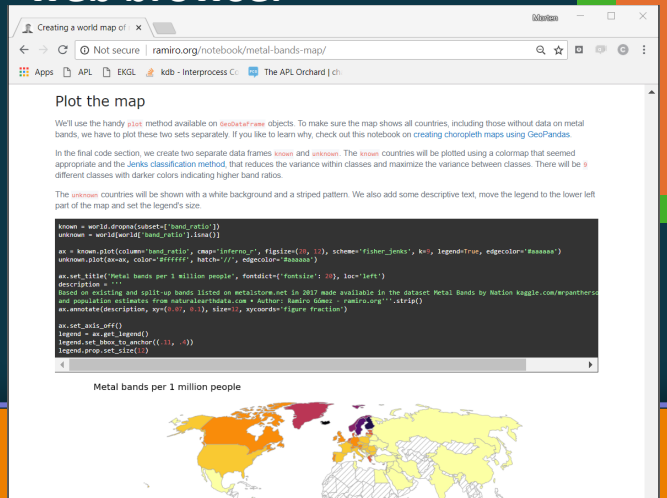
Anaconda
Python platform

notebook server
localhost:8888

Jupyter kernel
for Python

interpreter
Python

web browser



Jupyter Notebooks

Local notebook server — Python

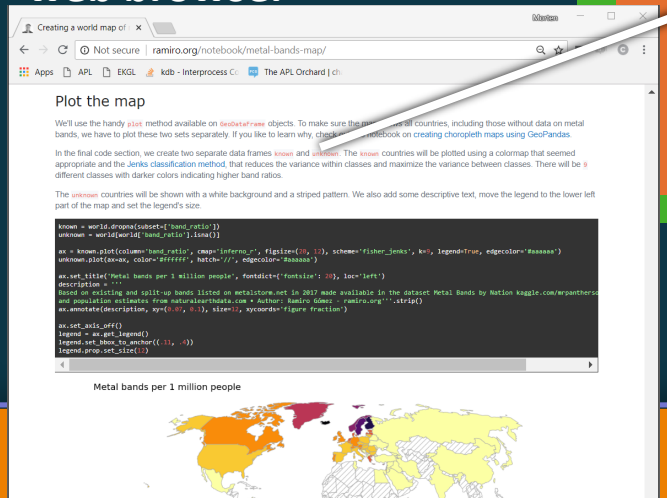
Anaconda
Python platform

notebook server
localhost:8888

Jupyter kernel
for Python

interpreter
Python

web browser



Local notebook server — Python

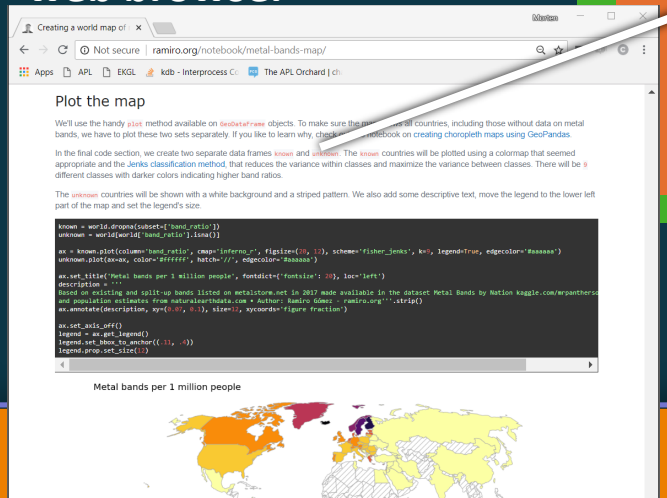
Anaconda
Python platform

notebook server
localhost:8888

Jupyter kernel
for Python

interpreter
Python

web browser



Local notebook server — Python

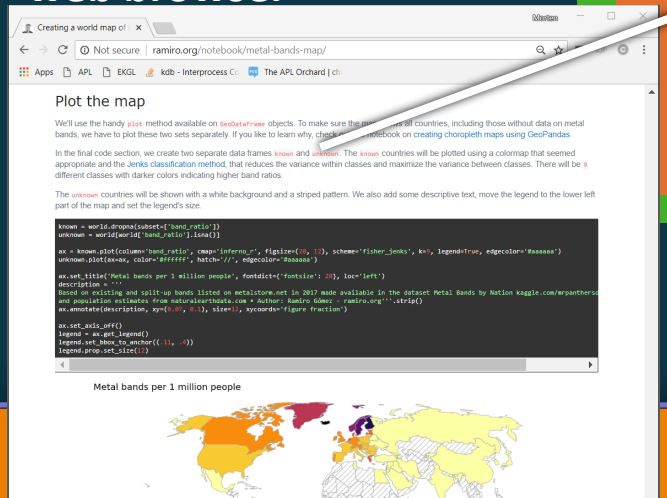
Anaconda
Python platform

notebook server
localhost:8888

Jupyter kernel
for Python

interpreter
Python

web browser



Local notebook server — Python

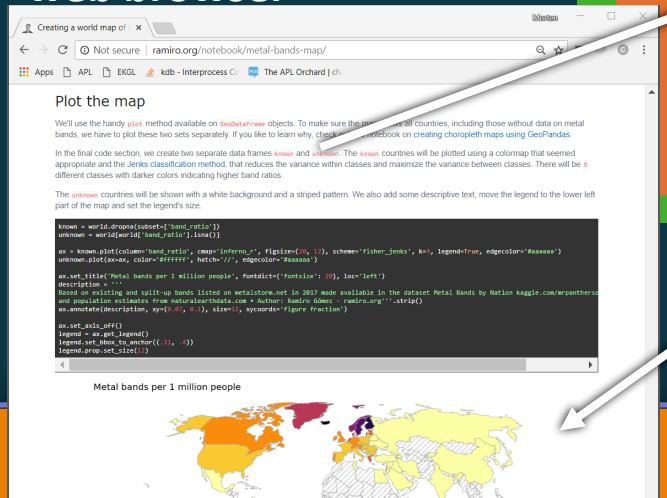
Anaconda
Python platform

notebook server
localhost:8888

Jupyter kernel
for Python

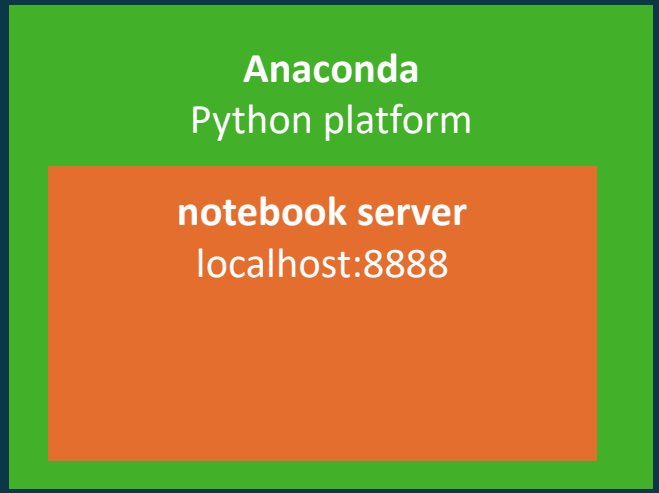
interpreter
Python

web browser

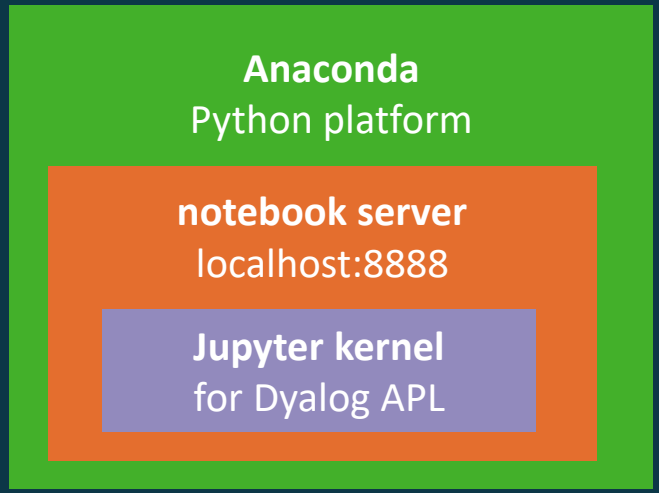


Jupyter Notebooks

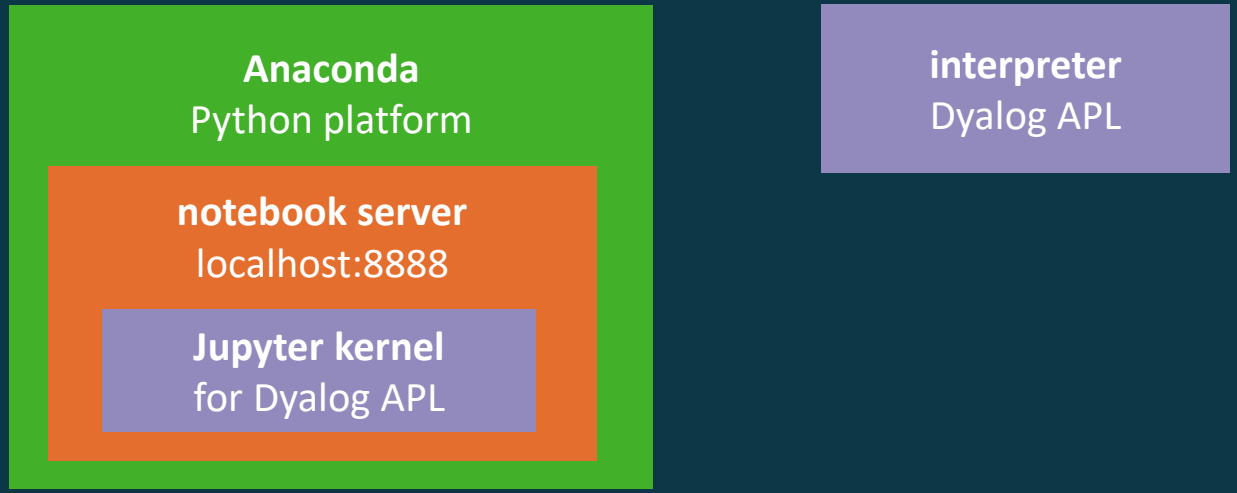
Local notebook server — APL



Local notebook server — APL



Local notebook server — APL



Local notebook server — APL

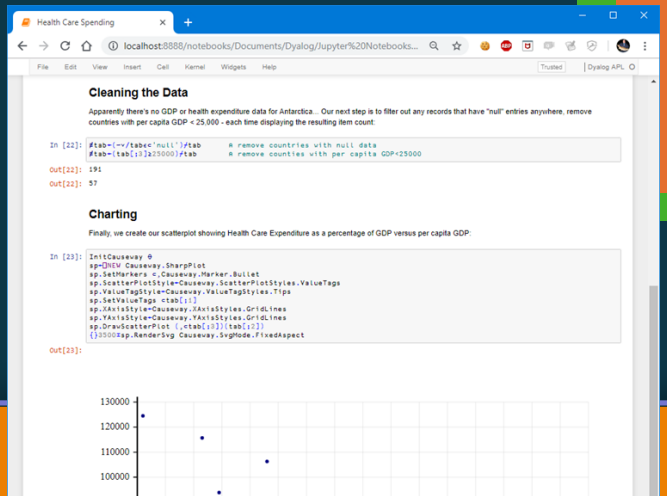
Anaconda
Python platform

notebook server
localhost:8888

Jupyter kernel
for Dyalog APL

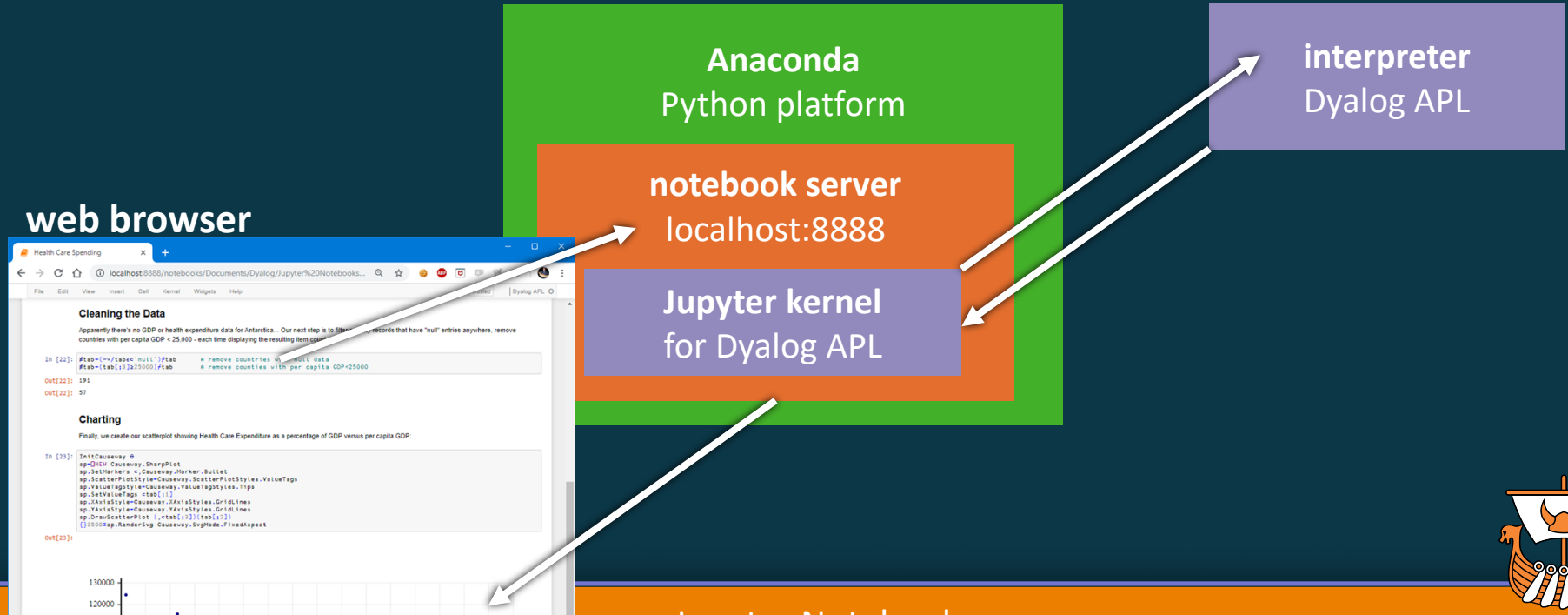
interpreter
Dyalog APL

web browser



Jupyter Notebooks

Local notebook server — APL



Local notebook server — APL

Anaconda
Python platform

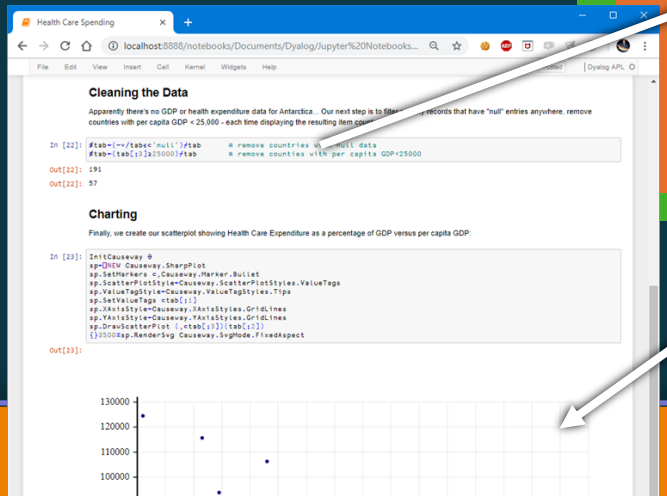
notebook server
localhost:8888

Jupyter kernel
for Dyalog APL

interpreter
Dyalog APL

RIDE protocol

web browser



Jupyter Notebooks

Setting up local notebook server

Install Dyalog ☺

Install Dyalog's Jupyter kernel

Install Anaconda

Launch Jupyter notebook server

Setting up local notebook server

Install Dyalog ☺

Install Dyalog's Jupyter kernel

Install Anaconda

Launch Jupyter notebook server

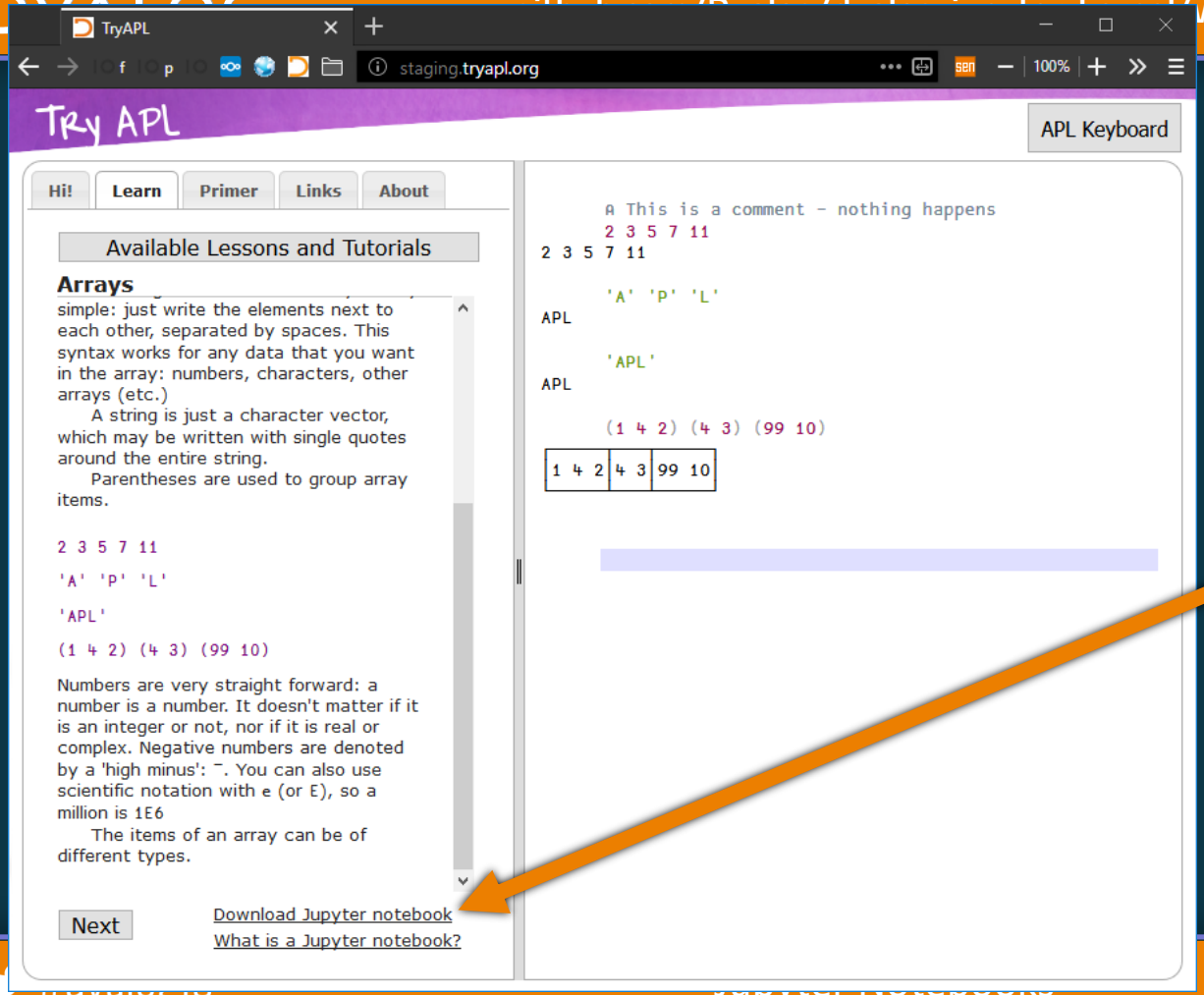
installation instructions

Demo

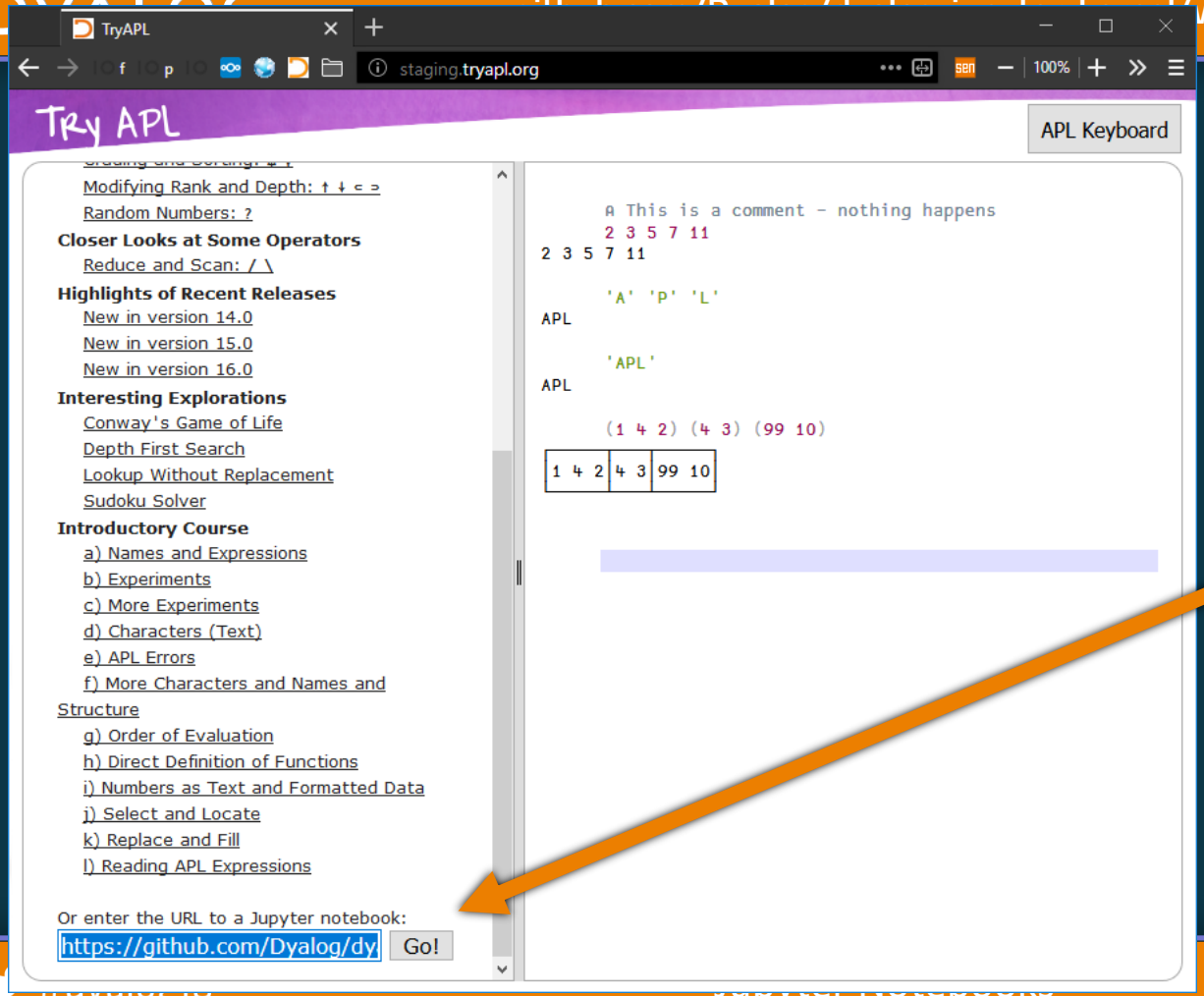
Installing Jupyter
Opening a notebook
Modifying content

Online notebook servers

There are online services for various programming languages



TryAPL's lessons are now Jupyter notebooks



and you can up-load your own Jupyter notebooks as well

Online notebook servers

Benefit: nothing to install

- you may need to sign up for an account

To protect servers, host may place restrictions

- or run in a sandbox with limited connectivity

Notebooks can execute any code

- all code is run on the host server
- same privileges as local execution

Static notebook viewers

Notebooks are stored as .ipynb files
- .ipynb files are in JSON format
- each code cell may include output from the last execution

You can share an .ipynb file
- anyone with a local notebook server can view it
- … but of course cannot execute anything new

Many online systems have viewers
- GitHub's file previewer
- Project Jupyter's nbviewer.jupyter.org

Exported notebooks

Notebooks can be exported to many standard formats

for example HTML, PDF, and LaTeX

Some formats require 3rd party plug-ins

Exported notebooks are static

that is expressions cannot be re-executed

Demo

Creating a new notebook document
Running our own notebook under TryAPL
Generating rich output

Ways to use notebooks — recap

Installing a Jupyter notebook server on your PC

Use an online notebook server like cocalc.com

Store the notebook with output, then open in a notebook viewer

Export to HTML, PDF, LaTeX, …

Ask questions now!

Wiki github.com/Dyalog/dyalog-jupyter-kernel/wiki
Email notebooks@dyalog.com and tryapl@dyalog.com

Thank you

Technology Partnership (tp.rs): prototype APL kernel
Will Robertson (intern): kernel work and many notebooks
Gil Athoraya (of Optima Systems): syntax colouring

Last day of Dyalog '18… Let's stay in touch!

dyalog.com
    Community
        Chat Room

chat.stackexchange.com/
    rooms/52405

Search: "apl orchard"

DYALOG

Last day of Dyalog '18

dyalog.com

Community

Chat Room

chat.stackexchange.com/rooms/52405

Search: "apl orchard"

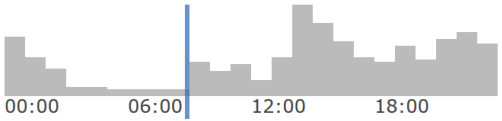
The APL Orchard

Learn and teach, questions about both golfing and general coding. Email support@dyalog.com for write access. Enter)about for chatbot info.

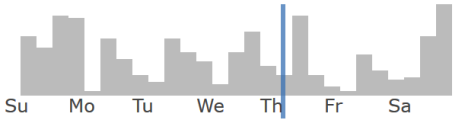
apl | array-manipulation | j | k | tips

first message 2017-01-24

last message 10 seconds ago



43 today | 252 yesterday | 177 per day



1.4k this week | 1.1k last week | 1.3k per week

join this room

view transcript

search for messages containing

86 37580