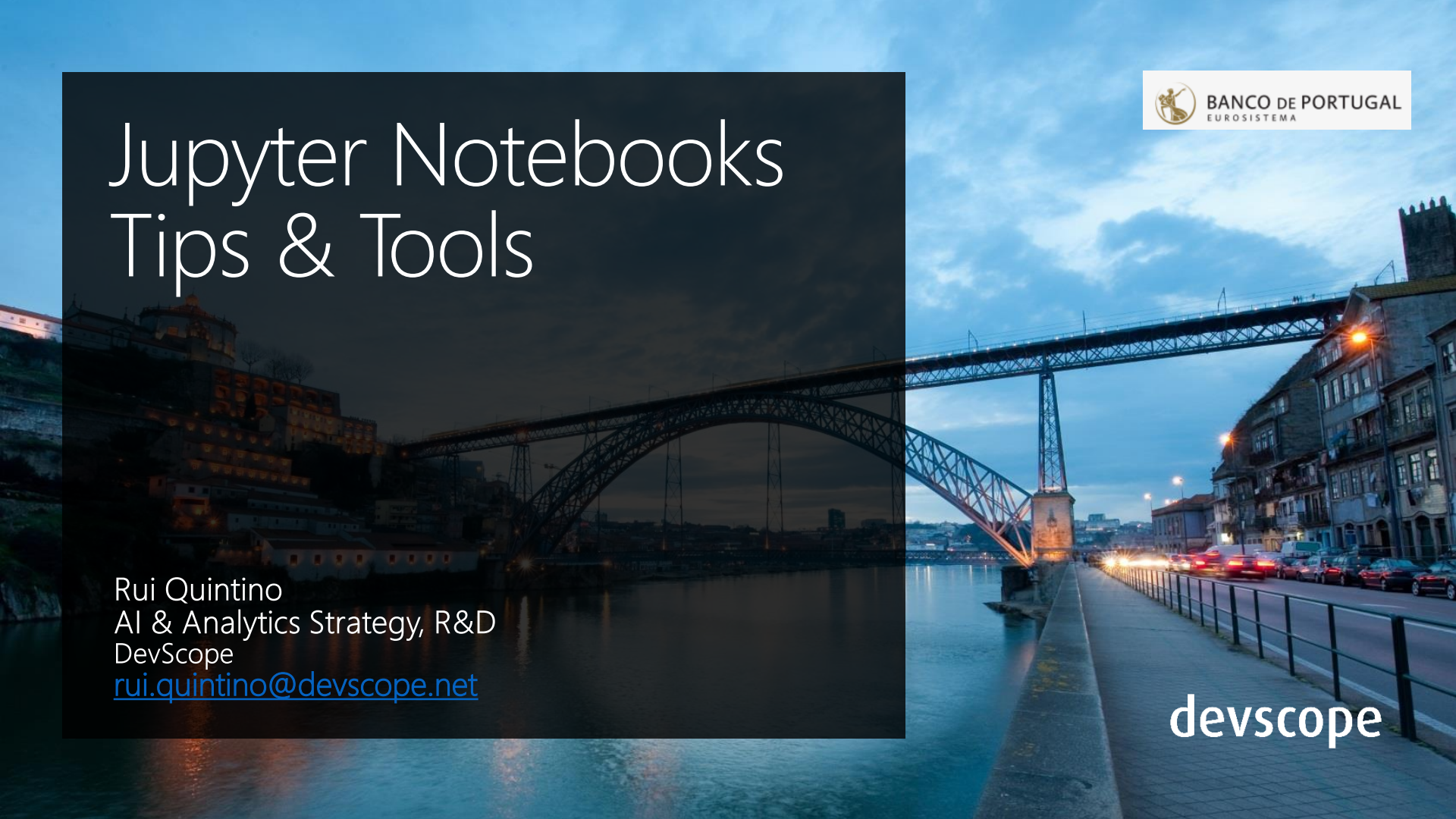


Jupyter Notebooks Tips & Tools

Rui Quintino
AI & Analytics Strategy, R&D
DevScope
rui.quintino@devscope.net

BANCO DE PORTUGAL
EUROSISTEMA

devscope

Contacts & Links

 rui.quintino@devscope.net

 Linkedin.com/ in/rquintino

 Twitter @rquintino

 Medium /devscope-ai

 Github /DevScope/ai-lab

Session Notebooks & Resources

 Github /DevScope/ai-lab

Today's Session

- Jupyter Intro/Basics (What & Why)
- Jupyter Productivity Tips & Tools (How)
- Jupyter Challenges & Hints on Reproducibility
- Q&A


jupyter

1 Jupyter is platform for *interactive* computing across dozens of programming languages.

(though we'll use **python** for this session)

Start at https://jupyter.org/

2 Why Jupyter Notebooks?

- Interactive Programming/Computing
- Literate Programming (Code & Documentation)
- Free and Open Source
- Multi Language
- Multi Platform
- 100% Browser based (awesome for docker/containers)
- Rich Content
- Extensible
- Powerfull Backends
- Huge Community
- (...)

Jupyter Basics

Quit Logout

Files Running Clusters Nbextensions

Select items to perform actions on them.

Upload New

0 / Name Last Modified File size

Filter .* Aa

archive	3 minutes ago	
docker	8 months ago	
src	8 months ago	
jupyter-tips-tools.ipynb	Running seconds ago	2.95 MB
jupyter-tips-tools.py	2 days ago	9.76 kB
titanic-survival.ipynb	9 months ago	2.18 kB
20190328-Python-Porto-Jupyter-Tips-Tricks.pptx	38 minutes ago	13.4 MB
20191216-BPLIM-Workshop-Jupyter-Tips-Tools.pptx	a minute ago	13.6 MB
azure-notebooks-jupyter-tips-tricks.url	9 months ago	120 B
config.yml	9 months ago	182 B
titanic-train.csv	9 months ago	60.3 kB

3 Jupyter/IPython basics

In [25]:
```
a=1
```
executed in 67ms, finished 20:35:08 2019-12-16

In [26]:
```
a+2
```
executed in 69ms, finished 20:35:09 2019-12-16

Out[26]:
```
3
```

In [29]:
```
# History
In[1],In[2]
```
executed in 83ms, finished 20:35:28 2019-12-16

Out[29]:
```
('a=1', 'a+2')
```

In [30]:
```
%history
```
executed in 74ms, finished 20:35:34 2019-12-16

```
a=1
a+2
# History
# In[1]
%history
%%html
<a href="https://jupyter.org/">a link</a>
# Data visualization
import pandas as pd
```

In [32]:
```
%%html
<a href="https://jupyter.org/">Jupyter Org</a>
```

executed in 69ms, finished 20:36:02 2019-12-16

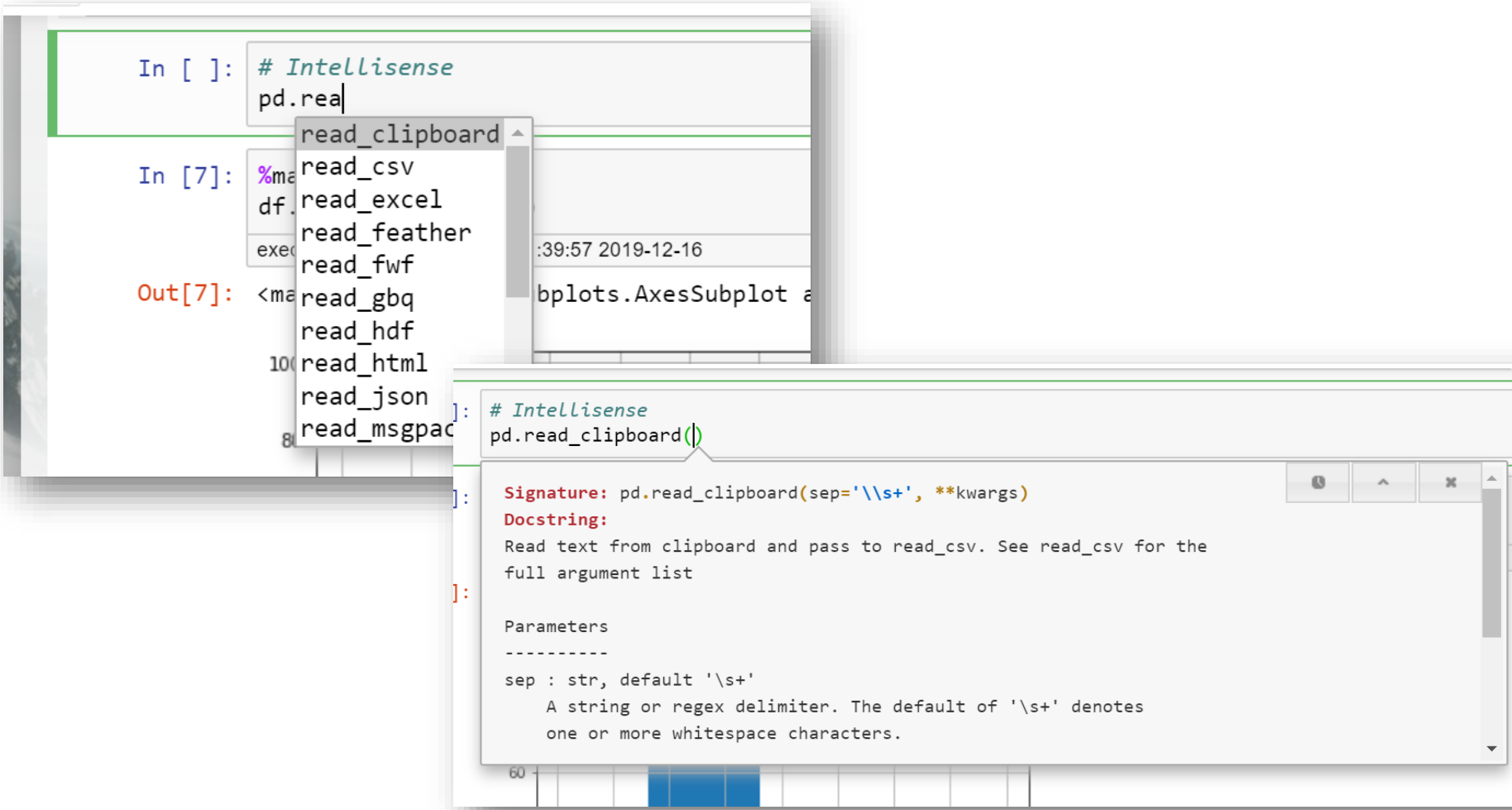
Jupyter Org

In [47]:
```
# Data visualization
import pandas as pd

df=pd.read_csv("titanic-train.csv")
df
```

executed in 181ms, finished 22:06:12 2019-12-16

Out[47]:

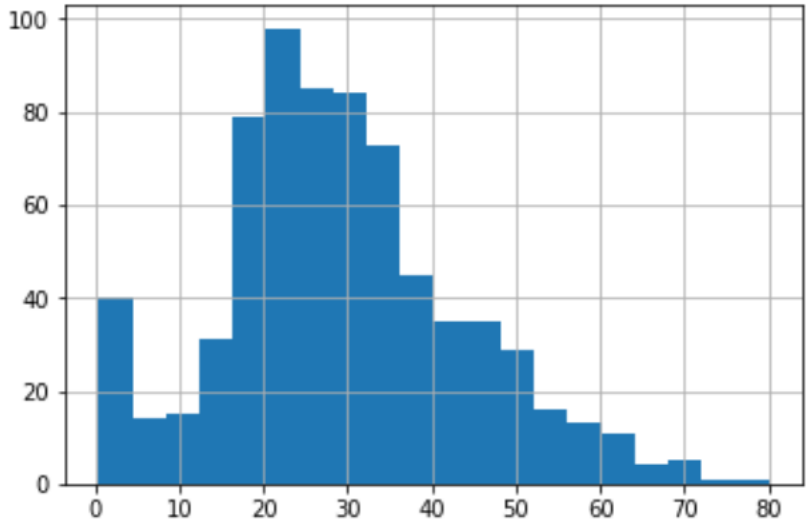
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	


In [ ]:
Intellisense
pd.rea
read_clipboard
read_csv
read_excel
read_feather
read_fwf
read_gbq
read_hdf
read_html
read_json
read_msgpac
In [7]:
%ma
df.
exec
:39:57 2019-12-16
Out[7]:
<ma
bplots.AxesSubplot a
# Intellisense
pd.read_clipboard()
Signature: pd.read_clipboard(sep='\\s+', **kwargs)
Docstring:
Read text from clipboard and pass to read_csv. See read_csv for the
full argument list
Parameters
sep : str, default '\s+'
A string or regex delimiter. The default of '\s+' denotes
one or more whitespace characters.

In [7]:
```
%matplotlib inline
df.Age.hist(bins=20)
```

executed in 407ms, finished 11:39:57 2019-12-16

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x208aced6710>



```
[LaTex](https://www.latex-project.org) equations support:

$$c = \sqrt{a^2 + b/2}$$
```

LaTex equations support:

$$c = \sqrt{a^2 + b/2}$$

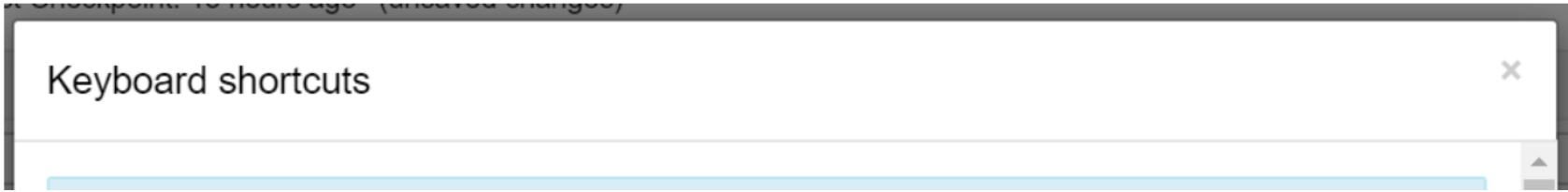
Jupyter Keyboard Shortcuts

4 Learn Keyboard shortcuts!

Some of my favorite:

- command mode shortcuts (for command mode use esc)
- **shift-enter* -> execute & next
- **ctrl-enter** -> execute current
- **d+d** -> delete cell
- **z** -> undo delete
- **b** -> insert below
- **a** -> inser above
- **m** -> switch to mardown
- **i+i** -> interrupt
- f -> find and replace

Command Mode (esc) + h



5 Help Tips

In [17]:
```
# Help tips
?
```

executed in 4ms, finished 17:11:12 2019-12-15

In [45]:
```
# shift+tab 1..4
print()
```

executed in 42ms, finished 22:36:00 2019-12-11

In [19]:
```
??print
```

executed in 8ms, finished 17:11:52 2019-12-15

Magics

6 Magics (they are well, magic!)

In [50]:
```
%lsmagic
```

executed in 123ms, finished 22:21:01 2019-12-16

Out[50]:
```
Available line magics:
%aimport %alias %alias_magic %autoawait %autocall %automagic %autoreload %autosave %bookmark
%cd %clear %cls %colors %conda %config %connect_info %copy %ddir %debug %dhist %dirs %doc
test_mode %echo %ed %edit %env %gui %hist %history %killbgscripts %ldir %less %load %load
_ext %loadpy %logoff %logon %logstart %logstate %logstop %ls %lsmagic %macro %magic %matpl
otlib %mkdir %more %notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %pip
%popd %pprint %precision %prun %psearch %psource %pushd %pwd %pycat %pylab %qtconsole %qui
ckref %recall %rehashx %reload_ext %ren %rep %rerun %reset %reset_selective %rmdir %run %s
ave %sc %set_env %store %sx %system %tb %time %timeit %unalias %unload_ext %who %who_ls
%whos %xdel %xmode

Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html %%javascript %%js %%latex %
%markdown %%perl %%prun %%pypy %%python %%python2 %%python3 %%ruby %%script %%sh %%svg %%s
x %%system %%time %%timeit %%writefile

Automagic is ON, % prefix IS NOT needed for line magics.
```

In [21]:
```
#Shell command
!echo %username%
```

executed in 82ms, finished 17:12:19 2019-12-15

In [22]:
```
# Access Kernel variables
x=1+2

!echo $x
```

executed in 84ms, finished 14:45:35 2019-12-16

```
3
```

In [24]:
```
!echo {x}
```

executed in 72ms, finished 17:12:31 2019-12-15

```
3
```

In [25]:
```
# even python loops & shell scripts
for i in range(10):
    print(i)
    !dir
```

executed in 222ms, finished 17:12:36 2019-12-15

```
0
 Volume in drive C has no label.
 Volume Serial Number is 2EB6-DCC2
```

Jupyter Extensions

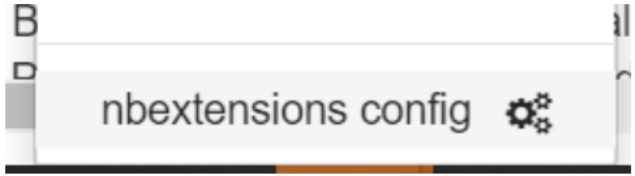
7 Notebook Extensions

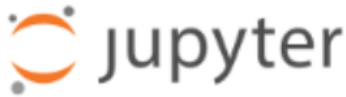
7.1 Install [...]

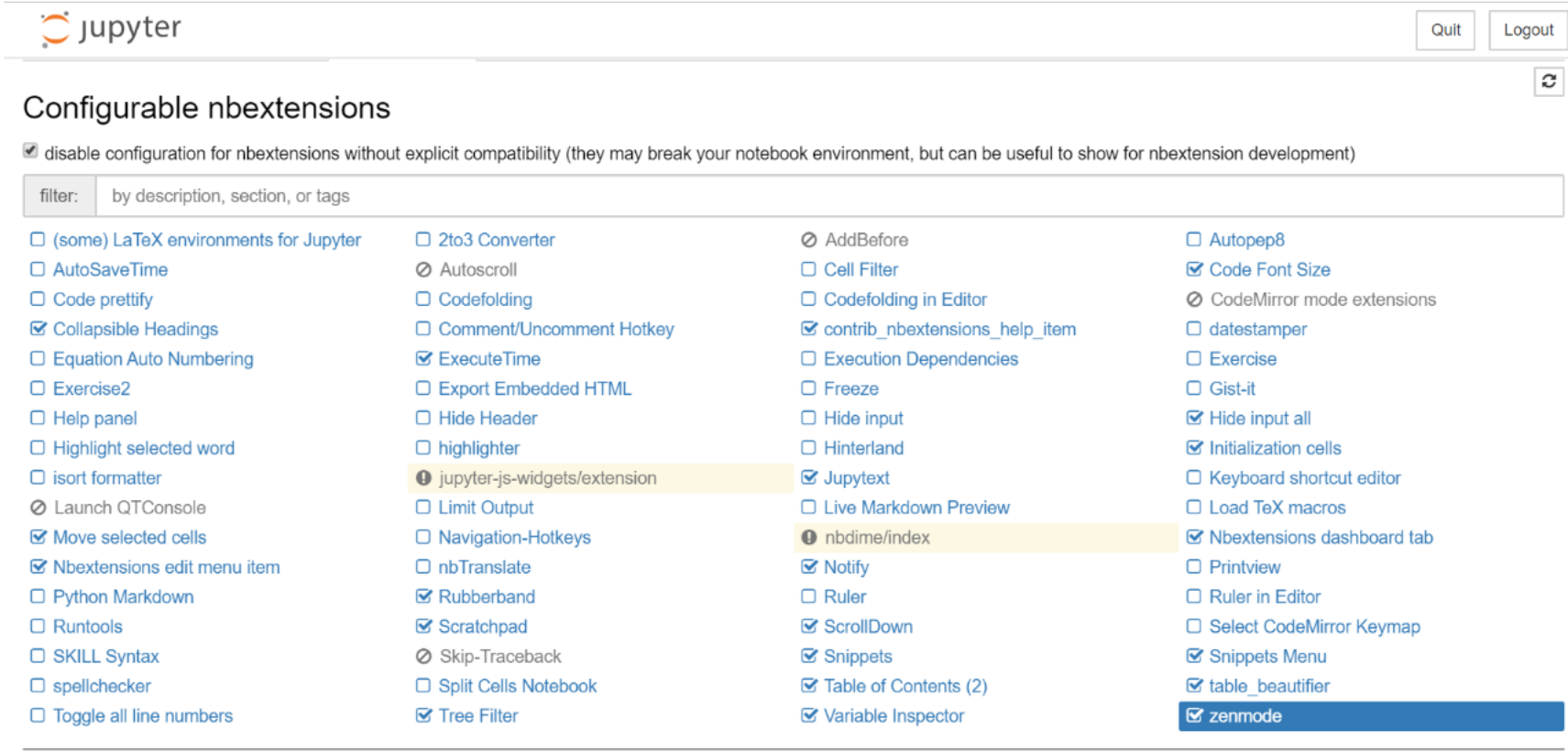
https://github.com/ipython-contrib/jupyter_contrib_nbextensions

- pip install jupyter_contrib_nbextensions
- jupyter contrib nbextension install --user

7.2 Configure/Enable






jupyter
Quit
Logout
Configurable nbextensions
disable configuration for nbextensions without explicit compatibility (they may break your notebook environment, but can be useful to show for nbextension development)
filter:
by description, section, or tags
(some) LaTeX environments for Jupyter
AutoSaveTime
Code prettify
Collapsible Headings
Equation Auto Numbering
Exercise2
Help panel
Highlight selected word
isort formatter
Launch QTConsole
Move selected cells
Nbextensions edit menu item
Python Markdown
Runtools
SKILL Syntax
spellchecker
Toggle all line numbers
2to3 Converter
Autoscroll
Codefolding
Comment/Uncomment Hotkey
ExecuteTime
Export Embedded HTML
Hide Header
highlighter
jupyter-js-widgets/extension
Limit Output
Navigation-Hotkeys
nbTranslate
Rubberband
Scratchpad
Skip-Traceback
Split Cells Notebook
Tree Filter
AddBefore
Cell Filter
Codefolding in Editor
contrib_nbextensions_help_item
Execution Dependencies
Freeze
Hide input
Hinterland
Jupytext
Live Markdown Preview
nbdime/index
Notify
Ruler
ScrollDown
Snippets
Table of Contents (2)
Variable Inspector
Autopep8
Code Font Size
CodeMirror mode extensions
datestamper
Exercise
Gist-it
Hide input all
Initialization cells
Keyboard shortcut editor
Load TeX macros
Nbextensions dashboard tab
Printview
Ruler in Editor
Select CodeMirror Keymap
Snippets Menu
table_beautifier
zenmode

7.3 Script extensions activation (automate, ex: Docker)

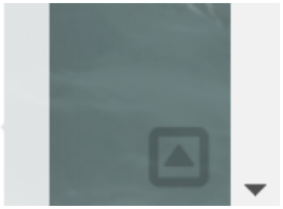
Example, see here binder links here, Jupyter will start with a lot of extensions pre-enabled and ready to go!

- https://github.com/DevScope/ai-lab

```
jupyter nbextension enable toc2/main
jupyter nbextension enable hide_input_all/main
jupyter nbextension enable table_beautifier/main
jupyter nbextension enable execute_time/ExecuteTime
jupyter nbextension enable codefolding/main
jupyter nbextension enable tree-filter/index
jupyter nbextension enable notify/notify
```

7.4 Some Favorite Extensions

7.4.1 Scratchpad (ctrl+b)



7.4.2 Table of Contents

7.4.2.1 ps- map a keyboard shortcut to toggle TOC!

7.4.3 Table Beautifier

In [51]:
```
import pandas as pd

pd.read_csv("titanic-train.csv")
```

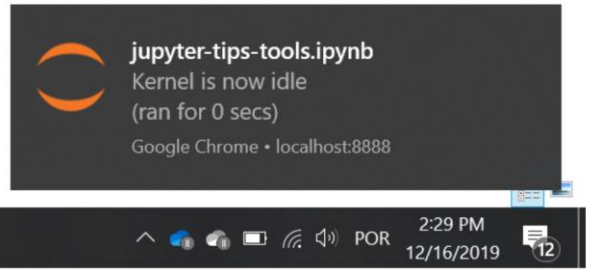
executed in 56ms, finished 22:36:05 2019-12-11

Out[51]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr.	male	35.0	0	0	373450	8.0500	NaN	

7.4.4 Notify

In [40]:
```
%matplotlib inline

import pandas as pd
from sklearn import model_selection
import matplotlib.pyplot as plt
from lightgbm import LGBMClassifier
from sklearn.model_selection import cross_val_score,StratifiedKFold

df=pd.read_csv("./titanic-train.csv")
target="Survived"

# Force categoricals
df[df.select_dtypes(['object']).columns.values]=df.select_dtypes(['object']).apply(pd.Series.a

# Cross validation
results=cross_val_score(LGBMClassifier(),
                X=df.drop(columns=target),
                y=df[target],
                cv=60)

plt.hist(results)
print(np.mean(results))
```



8 Debugging

8.1 Set breakpoints/trace

In [51]:
```
# Use IPython Debugger
# (some improvements over default debugger:
# syntax highlighting,etc)
from IPython.core.debugger import set_trace

def calc(x,y):
    # break here!
    set_trace()

    x=x*3
    return x/y
```

executed in 163ms, finished 22:26:13 2019-12-16

In [*]:
```
calc(8,1)
```

execution queued 22:26:14 2019-12-16

```
> <ipython-input-51-723421edf8e2>(10)calc()
      7     # break here!
      8     set_trace()
      9
---> 10     x=x*3
     11     return x/y

ipdb> x
8

ipdb>
```

8.2 Force debugger

In [*]:
```
%debug print(calc(7,9))
```

execution queued 22:27:13 2019-12-16

```
NOTE: Enter 'c' at the ipdb>  prompt to continue execution.
> <string>(1)<module>()

ipdb> n
> <ipython-input-51-723421edf8e2>(10)calc()
      7     # break here!
      8     set_trace()
      9
---> 10     x=x*3
     11     return x/y

ipdb> a
x = 7
y = 9

ipdb> q
```

8.3 Debug after exception

In [55]:
```
def calc(x,y):
    x=x*3
    return x/y
```
executed in 153ms, finished 22:28:09 2019-12-16

In [56]:
```
calc(8,0)
```
executed in 106ms, finished 22:28:10 2019-12-16

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-56-3983f776a6e8> in <module>
----> 1 calc(8,0)

<ipython-input-55-b8876a6023a3> in calc(x, y)
      1 def calc(x,y):
      2     x=x*3
----> 3     return x/y

ZeroDivisionError: division by zero
```

In [*]:
```
%debug
```
execution queued 22:28:12 2019-12-16

```
ipdb>

> <ipython-input-55-b8876a6023a3>(3)calc()
      1 def calc(x,y):
      2     x=x*3
----> 3     return x/y
```

Package Patterns

9.1 Package pattern & auto reload imports

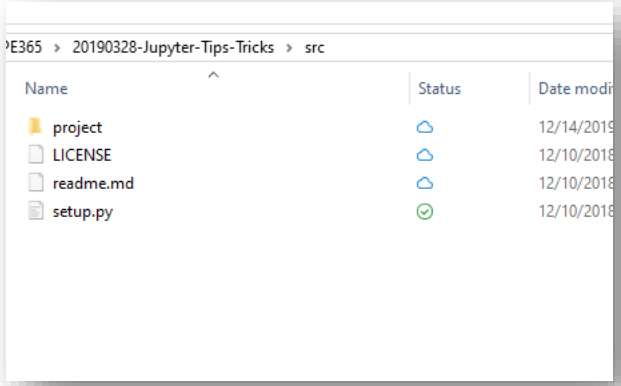
From: **Write less terrible code with Jupyter Notebook by GoDataDriven**

https://blog.godatadriven.com/write-less-terrible-notebook-code

Install folder as editable package (live reference):

tip: install sub-folder instead of root project to avoid full folder copy (pip issue)

```
(base) C:\Users'
>pip install -e .
Obtaining file:
Installing collected packages: project
  Found existing installation: project 0.0.2
    Uninstalling project-0.0.2:
      Successfully uninstalled project-0.0.2
  Running setup.py develop for project
Successfully installed project

(base)
>
```



__init__.py ●

src > project > project > __init__.py > test

```python
import os
import sys
from pathlib import Path
from os.path import dirname

__version__ ="0.0.26"

print("Loading project functions & utils...")
print(f"Version:{__version__}")


PROJECT_ROOT_MARKER=".git"
#Alternatives? MLProject project.yml .git

def test():
    print("running workshop v2")

def get_root(folder=None):
    if (not folder):
```

In [58]:
```
%load_ext autoreload
%autoreload 2
```

executed in 162ms, finished 22:30:06 2019-12-16

```
Loading project functions & utils...
Version:0.0.26
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [59]:
```
import project as proj
```

executed in 116ms, finished 22:30:07 2019-12-16

In [60]:
```
proj.get_root()
```

executed in 107ms, finished 22:30:08 2019-12-16

Out[60]:
```
'C:\\Users                                    20190328-Jupyter-Tips-Tricks/'
```

In [63]:
```
proj.test()
```

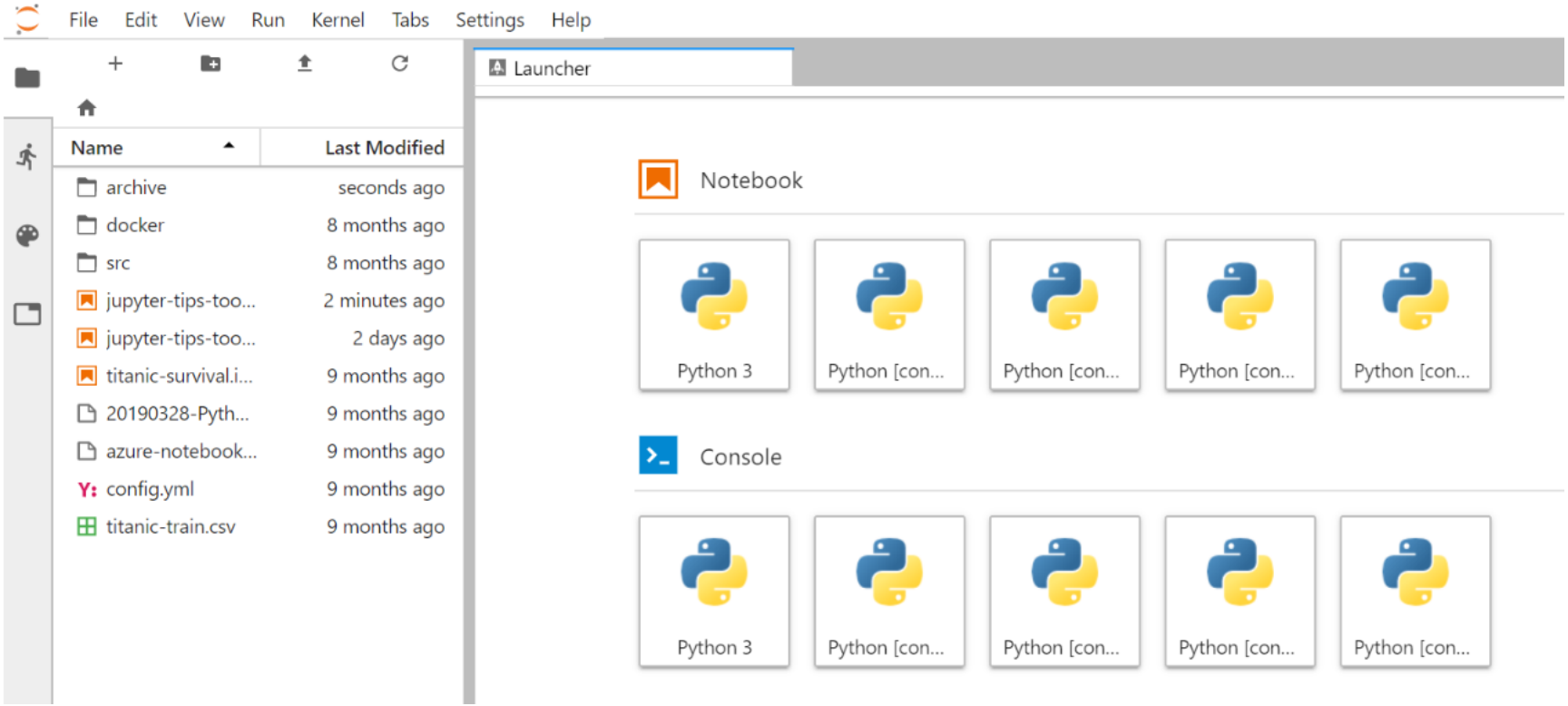
executed in 98ms, finished 22:30:22 2019-12-16

```
running workshop v2!
```

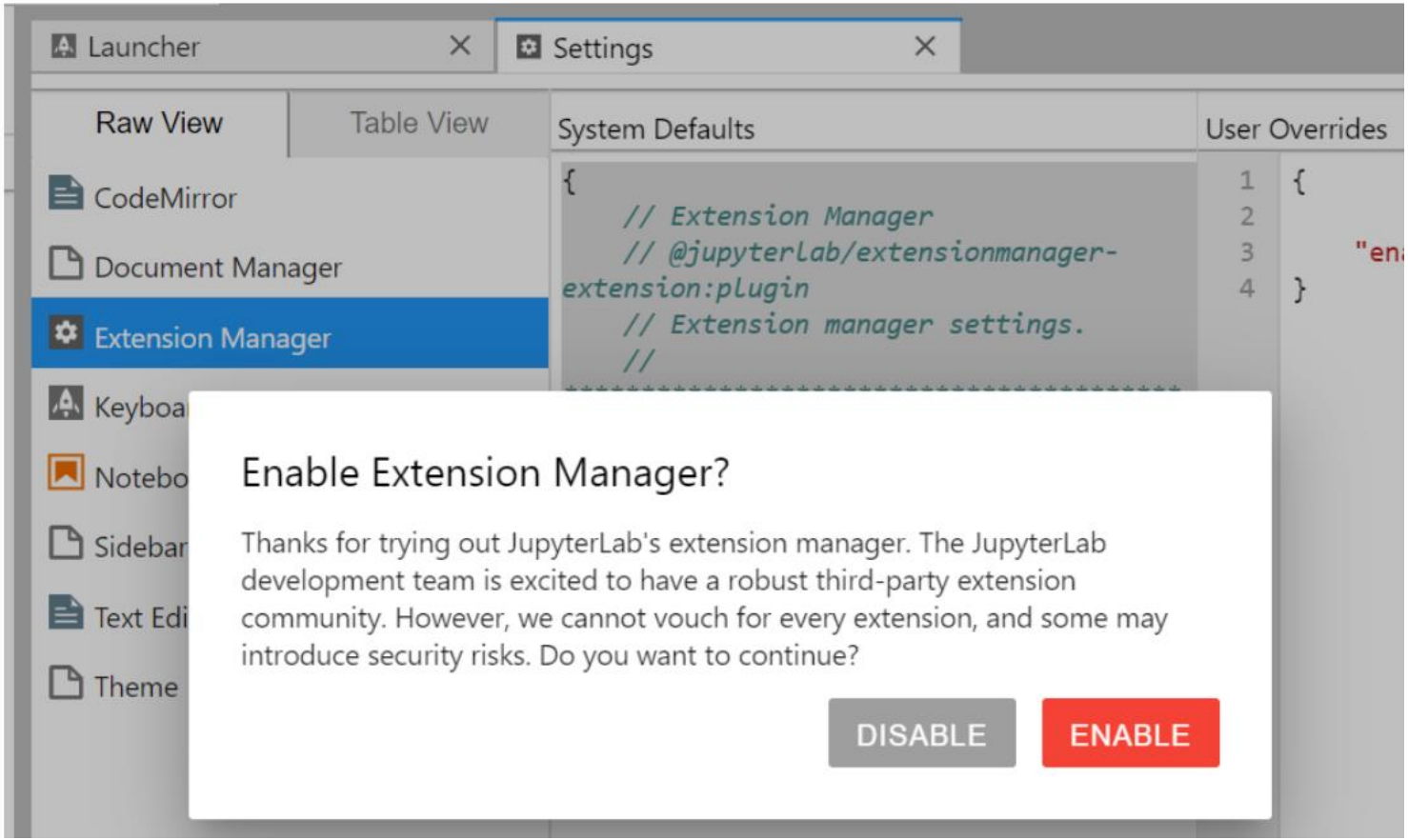
Jupyter Ecosystem

10.1 Jupyter Lab: https://github.com/jupyterlab/jupyterlab

> Future of Jupyter or Jupyter vNext



10.1.1 Tip: Jupyter Lab Extensions - Enable! (not shown by default!)



10.2 Jupyter Hub

https://jupyter.org/hub

> JupyterHub brings the power of notebooks to groups of users.

10.3 MyBinder & repo2docker

https://mybinder.org/

> Turn a Git repo into a collection of interactive notebooks

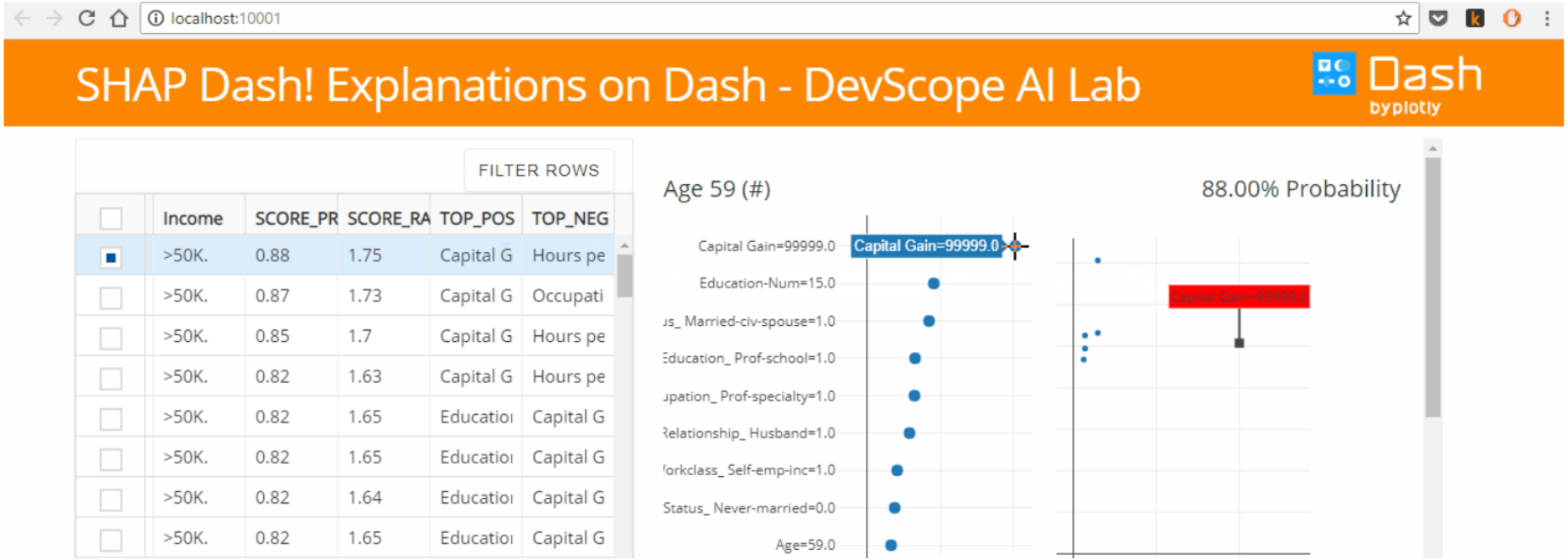
https://repo2docker.readthedocs.io/en/latest/index.html

> jupyter-repo2docker is a tool to build, run, and push Docker images from source code repositories.

MyBinder/Repo2Docker example

SHAP Dash! Explanations on Dash

launch binder



Starting repository: DevScope/ai-lab/master

If a repository takes a long time to launch, it is usually because Binder needs to create the environment for the first time.

Build logs hide

```
Found built image, launching...
Launching server...
```

hub.gke.mybinder.org/user/devscope-ai-lab-yleb4m6s/notebooks/notebooks/shap-dash/shap-dash.ipynb

jupyter shap-dash (unsaved changes)

Visit repo | Copy Binder link

File Edit View Insert Cell Kernel Navigate Widgets Help | Not Trusted | Python 3

Run | Code | Appmode | Notify: Disabled

Contents

In [ ]:
```
%load_ext autoreload
%autoreload 2
```
executed in 113ms, finished 15:48:10 2019-12-16

1 AI Lab tools/utils

In [ ]:
```
import ailab as lab
```
executed in 51ms, finished 15:48:10 2019-12-16

2 References

2.0.1 For more info on SHAP (SHapley Additive exPlanations) see

- https://github.com/slundberg/shap
- http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions
- https://christophm.github.io/interpretable-ml-book/shapley.html

2.0.2 More about Plotly Dash

- https://dash.plot.ly/

Jupyter Challenges

13.1.1 I Don't Like Notebooks - Joel Grus - #JupyterCon 2018

**must watch session for anyone using notebooks, lots of fun too :) **

Slides: https://docs.google.com/presentation/d/1n2RlMdmv1p25Xy5thJUhkKGvjtV-dkAIsUXP-AL4ffl/edit#slide=id.g362da58057_0_1

Session: https://www.youtube.com/watch?v=7jiPeIFXb6U

13.1.2 The First Notebook Wa, Yihui Xie / 2018-09-10

https://yihui.org/en/2018/09/notebook-war/

The two cultures: the R vs Python culture, or data analysis vs software engineering culture

I feel a major difference between the R culture and Python culture is that Python users seem to *create* code more often, whereas R users often *use* code. There seems to be a strong atmosphere of software engineering in the Python world: in the beginning was the custom *class* (with methods). For R users, in the beginning was the *data*.

13.1.3 Jupyter Notebooks Meet the Challenge of Reproducibility

https://thenewstack.io/jupyter-notebooks-challenge-reproducibility/

Interactivity vs. Reproducibility

13.1.4 The Notebook Anti-Pattern by

By Kristina Young, Senior Data Scientist

https://www.kdnuggets.com/2019/11/notebook-anti-pattern.html

(...)

What are notebooks good for?

- Data analysis
- Experimentation
- One time tasks
- Teaching or technical presentations
- Code assessments

What are notebooks bad at?

- Continuous integration (CI)
- Testing
- Version control
- Collaboration
- Reliance on state
- Duplication
- Lack of testing

(...)

13.1.5 Guide for Reproducible Research and Data Science in Jupyter Notebooks

https://github.com/jupyter-guide/jupyter-guide

Guides and Tutorials

- **Parameterize your notebooks**: How to pass in parameters to notebooks
- **Test your notebooks**: How to validate your to notebooks
- **Deploy your notebooks**: How to share your notebooks
- **Typeset equations**
- **Example Notebooks**
- Other sections (to be written)

13.1.6 Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks

https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007007

13.2 Further Notes & Tools

docker & containers

dvc- data version control

https://dvc.org/

> DVC tracks ML models and data sets

unit tests & continous integration

assertions & defensive programming

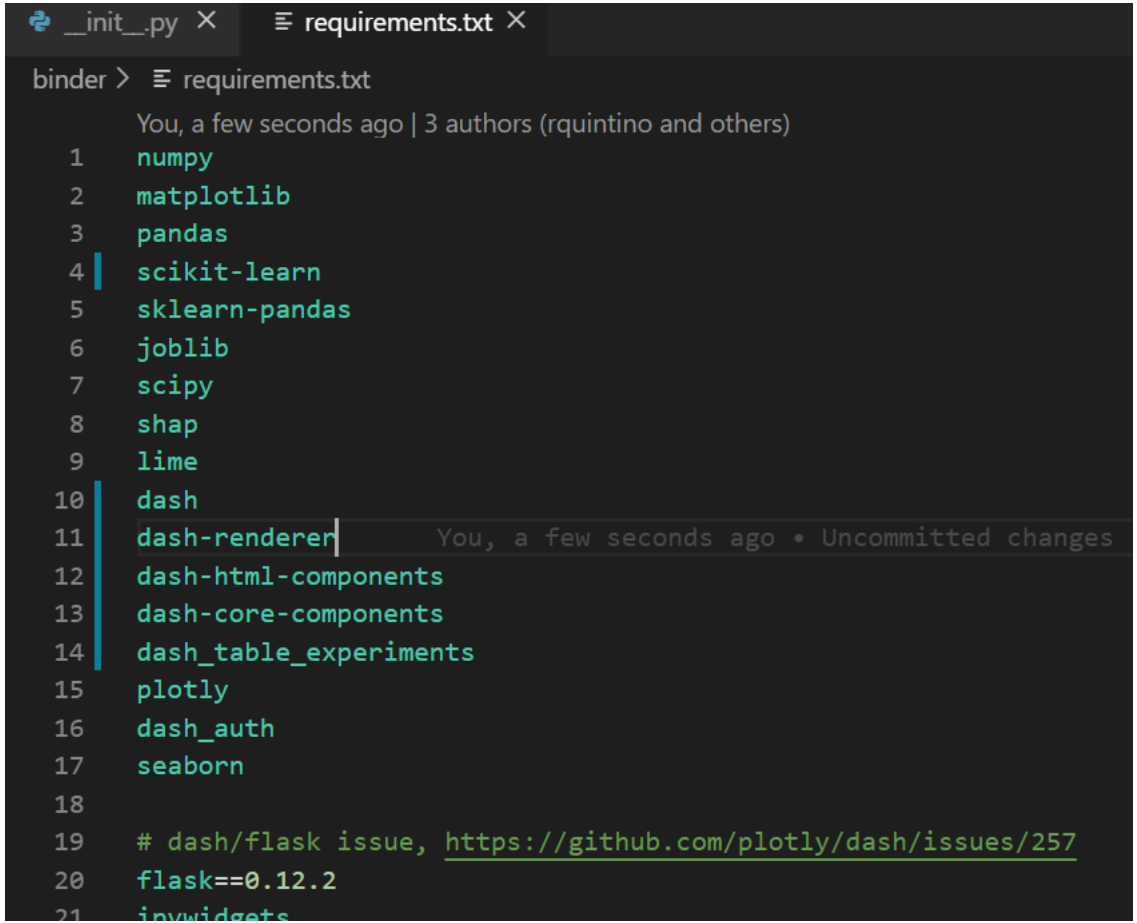
a case study of what not to do

draft/experiment notebooks vs final documents

code, GUIs and bugs

reproducibility case study

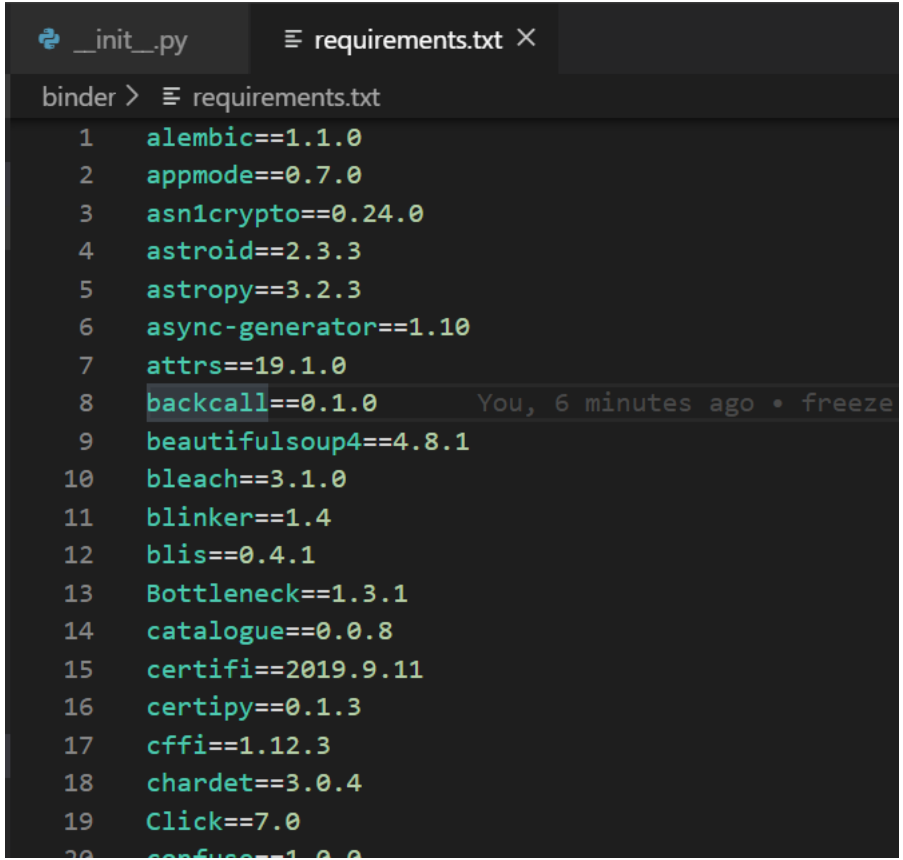
(of what not to do…)


__init__.py
requirements.txt
binder > requirements.txt
You, a few seconds ago | 3 authors (rquintino and others)
1 numpy
2 matplotlib
3 pandas
4 scikit-learn
5 sklearn-pandas
6 joblib
7 scipy
8 shap
9 lime
10 dash
11 dash-renderer
You, a few seconds ago • Uncommitted changes
12 dash-html-components
13 dash-core-components
14 dash_table_experiments
15 plotly
16 dash_auth
17 seaborn
18
19 # dash/flask issue, https://github.com/plotly/dash/issues/257
20 flask==0.12.2
21 ipywidgets

In [19]:
```
!pip freeze
```

executed in 2.87s, finished 20:45:36 2019-12-16

```
blis==0.4.1
Bottleneck==1.3.1
catalogue==0.0.8
certifi==2019.9.11
certipy==0.1.3
cffi==1.12.3
chardet==3.0.4
Click==7.0
confuse==1.0.0
cryptography==2.7
cycler==0.10.0
cymem==2.0.3
dash==0.36.0
dash-auth==1.3.2
dash-core-components==0.42.1
dash-html-components==1.0.0a2
dash-renderer==0.17.0
dash-table-experiments==0.6.0
decorator==4.4.0
defusedxml==0.5.0
```


__init__.py
requirements.txt
binder > requirements.txt
1 alembic==1.1.0
2 appmode==0.7.0
3 asn1crypto==0.24.0
4 astroid==2.3.3
5 astropy==3.2.3
6 async-generator==1.10
7 attrs==19.1.0
8 backcall==0.1.0 You, 6 minutes ago • freeze
9 beautifulsoup4==4.8.1
10 bleach==3.1.0
11 blinker==1.4
12 blis==0.4.1
13 Bottleneck==1.3.1
14 catalogue==0.0.8
15 certifi==2019.9.11
16 certipy==0.1.3
17 cffi==1.12.3
18 chardet==3.0.4
19 Click==7.0

11:59 61%

SHAP Dash! Explanations on Dash

launch binder

Dash Classification Eval

launch binder

Dash Search CV Eval

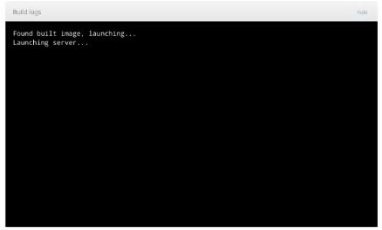
launch binder

11:59 61%

binder

Starting repository: DevScope/ai-lab/master

If a repository takes a long time to launch, it is usually because Binder needs to create the environment for the first time.



jupyter nbviewer

ai-lab / notebooks / shap-dash

```
In [ ]:
%load_ext autoreload
%autoreload 2
```

AI Lab tools/utils

```
In [ ]:
import ailab as lab
```

References

For more info on SHAP (SHapley Additive exPlanations) see

- https://github.com/slundberg/shap
- http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions
- https://christophm.github.io/interpretable-ml-book/shapley.html

More about Plotly Dash

- https://dash.plot.ly/

Read Data

```
In [ ]:
import pandas as pd

feature_names=["Age", "Workclass", "Final Weight", "Educat
               "Occupation", "Relationship", "Race", "Sex"
               "Hours per week", "Country", "Income"]
```

12:00 61%

jupyter shap-dash Visit repo Copy Bin

Menu Not Trusted Python 3

Run Code

Appmode

```
In [ ]:
%load_ext autoreload
%autoreload 2
```

executed in 113ms, finished 15:48:10 2019-12-16

1 AI Lab tools/utils

```
In [ ]:
import ailab as lab
```

executed in 51ms, finished 15:48:10 2019-12-16

2 References

2.0.1 For more info on SHAP (SHapley Additive exPlanations) see

- https://github.com/slundberg/shap
- http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions
- https://christophm.github.io/interpretable-ml-book/shapley.html

2.0.2 More about Plotly Dash

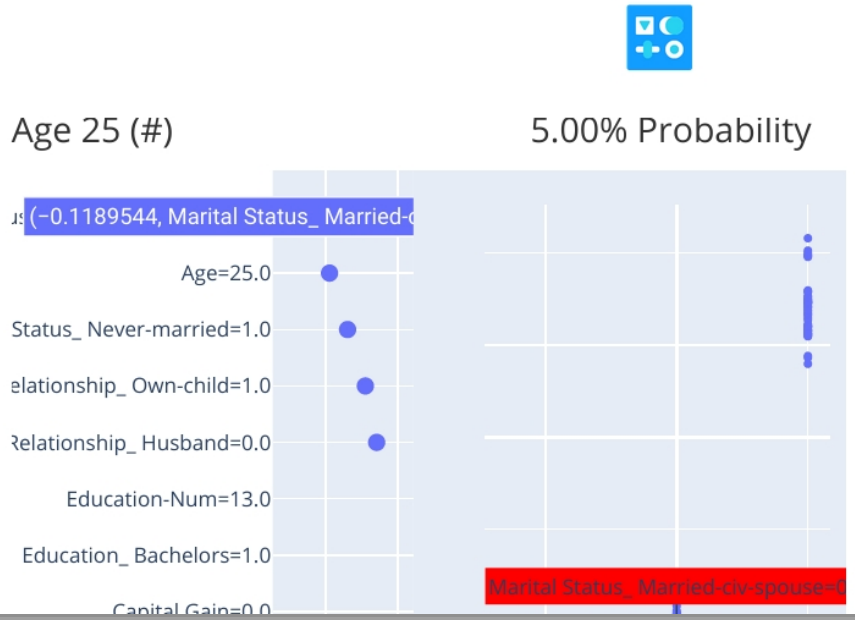
- https://dash.plot.ly/

12:03 60%

SHAP Dash! Explanations on Dash - DevScope AI Lab

FILTER ROWS

	Age	Workclass	Final Weig	Educ
☐	27	Private	212041	Bache
☑	25	Private	196947	Bache
☐	47	Private	165229	12th
☐	35	Self-emp	99146	Assoc
☐	24	Private	148315	HS-gr
☐	34	Private	85632	Some
☐	37	Private	144005	Some
☐	30	State-gov	184901	Bache



Q&A

Other Jupyter Tips?

Let me Know!!! ☺

Thank you!

 rui.quintino@devscope.net

 Linkedin.com/ in/rquintino

 Twitter @rquintino

 Medium /devscope-ai

 Github /DevScope/ai-lab

devscope

Rua Passos Manuel Nº 223 – 4º Andar
4000-385 Porto

Av. Sidónio Pais, Nº 2 – 3º Andar
1050-214 5 Lisboa

T. +351 223 751 350/51
F. +351 223 751 352

info@devscope.net
www.devscope.net