

example

May 14, 2023

```
[1]: # PDFs are autogenerated every time you push
      # go on branch: output-[BRANCH]
      # look at the output directory

      # manually convert ipynb to PDF
      !jupyter nbconvert --to pdf example.ipynb
```

```
[NbConvertApp] Converting notebook example.ipynb to pdf
[NbConvertApp] Support files will be in example_files/
[NbConvertApp] Making directory ./example_files
[NbConvertApp] Writing 26026 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 78115 bytes to example.pdf
```

```
[2]: import platform
      import os
      import sys
      %matplotlib inline

      print(f"Python Interpreter: {sys.executable}")
      print(f"Python Version: {platform.python_version()}")
      print()
      print(f"Working Directory: {os.getcwd()}")
      print()
      print(f"Platform")
      print(f"    System: {platform.system()}")
      print(f"    Release: {platform.release()}")
      print(f"    Version: {platform.version()}")
```

ModuleNotFoundError

Traceback (most recent call last)

Cell In[2], line 4
 2 import os

```

3 import sys
----> 4 get_ipython().run_line_magic('matplotlib', 'inline')
6 print(f"Python Interpreter: {sys.executable}")
7 print(f"Python Version: {platform.python_version()}")

```

File /opt/hostedtoolcache/Python/3.8.16/x64/lib/python3.8/site-packages/IPython/core/interactiveshell.py:2417, in InteractiveShell.run_line_magic(self, magic_name, line, _stack_depth)

```

2415     kwargs['local_ns'] = self.get_local_scope(stack_depth)
2416 with self.builtin_trap:
-> 2417     result = fn(*args, **kwargs)
2419 # The code below prevents the output from being displayed
2420 # when using magics with decorator @output_can_be_silenced
2421 # when the last Python token in the expression is a ';'.
2422 if getattr(fn, magic.MAGIC_OUTPUT_CAN_BE_SILENCED, False):

```

File /opt/hostedtoolcache/Python/3.8.16/x64/lib/python3.8/site-packages/IPython/core/magics/pylab.py:99, in PylabMagics.matplotlib(self, line)

```

97     print("Available matplotlib backends: %s" % backends_list)
98 else:
---> 99     gui, backend =
-> self.shell.enable_matplotlib(args.gui.lower() if isinstance(args.gui, str) else args.gui)
100     self._show_matplotlib_backend(args.gui, backend)

```

File /opt/hostedtoolcache/Python/3.8.16/x64/lib/python3.8/site-packages/IPython/core/interactiveshell.py:3588, in InteractiveShell.enable_matplotlib(self, gui)

```

3567 def enable_matplotlib(self, gui=None):
3568     """Enable interactive matplotlib and inline figure support.
3569
3570     This takes the following steps:
3571     (...)
3572     display figures inline.
3573     """
-> 3588     from matplotlib_inline.backend_inline import configure_inline_support
3590     from IPython.core import pylabtools as pt
3591     gui, backend = pt.find_gui_and_backend(gui, self.pylab_gui_select)

```

File /opt/hostedtoolcache/Python/3.8.16/x64/lib/python3.8/site-packages/matplotlib_inline/_init__.py:1

```

----> 1 from . import backend_inline, config # noqa
2 __version__ = "0.1.6" # noqa

```

File /opt/hostedtoolcache/Python/3.8.16/x64/lib/python3.8/site-packages/matplotlib_inline/backend_inline.py:6

```

1 """A matplotlib backend for publishing figures via display_data"""
3 # Copyright (c) IPython Development Team.
4 # Distributed under the terms of the BSD 3-Clause License.
----> 6 import matplotlib

```

```
7 from matplotlib import colors
8 from matplotlib.backends import backend_agg
```

ModuleNotFoundError: No module named 'matplotlib'

```
[3]: import matplotlib.pyplot as plt
import pandas as pd

# LOAD
url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/
↳titanic.csv"
titanic = pd.read_csv(url)

# PREPARE
survived = titanic[titanic['Survived'] == 1]['Age'].dropna()
not_survived = titanic[titanic['Survived'] == 0]['Age'].dropna()

# VISUALIZE
fig, ax = plt.subplots()
ax.violinplot([survived, not_survived], showmeans=True)
ax.set_xticks([1, 2])
ax.set_xticklabels(['Survived', 'Not Survived'])
ax.set_ylabel('Age')
ax.set_title('Violin Plot using Matplotlib')
plt.show()
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
Cell In[3], line 1
----> 1 import matplotlib.pyplot as plt
      2 import pandas as pd
      4 # LOAD

ModuleNotFoundError: No module named 'matplotlib'
```

```
[4]: import matplotlib.pyplot as plt
import seaborn as sns

# LOAD
titanic = sns.load_dataset("titanic")

# VISUALIZE
sns.violinplot(x="survived", y="age", data=titanic, inner="quartile")
plt.xlabel("Survived")
plt.ylabel("Age")
plt.title("Violin Plot using Seaborn")
```

```
plt.show()
```

```
-----  
ModuleNotFoundError                                Traceback (most recent call last)  
Cell In[4], line 1  
----> 1 import matplotlib.pyplot as plt  
      2 import seaborn as sns  
      4 # LOAD  
  
ModuleNotFoundError: No module named 'matplotlib'
```