Leveraging a Python IDE for Data Science

Xavier Morera

HELPING DEVELOPERS UNDERSTAND SEARCH & BIG DATA

@xmorera www.xaviermorera.com

DATA

Python IDE for Data Science

PyCharm Pro

Spyder

Python Notebooks for Data Science

Special kind of IDE

Web application

Execute Python code

- Other available interpreters
- Scala, R, Shell, SQL, Spark…

Python Notebooks for Data Science

Work with data

- Visualize
- Collaborate

Great for iteration, prototyping , and testing

Python Notebooks for Data Science

Jupyter Notebook

Apache Zeppelin

IPython Kernel-based IDE

Cloudera Data Science Workbench

Scientific Mode in PyCharm

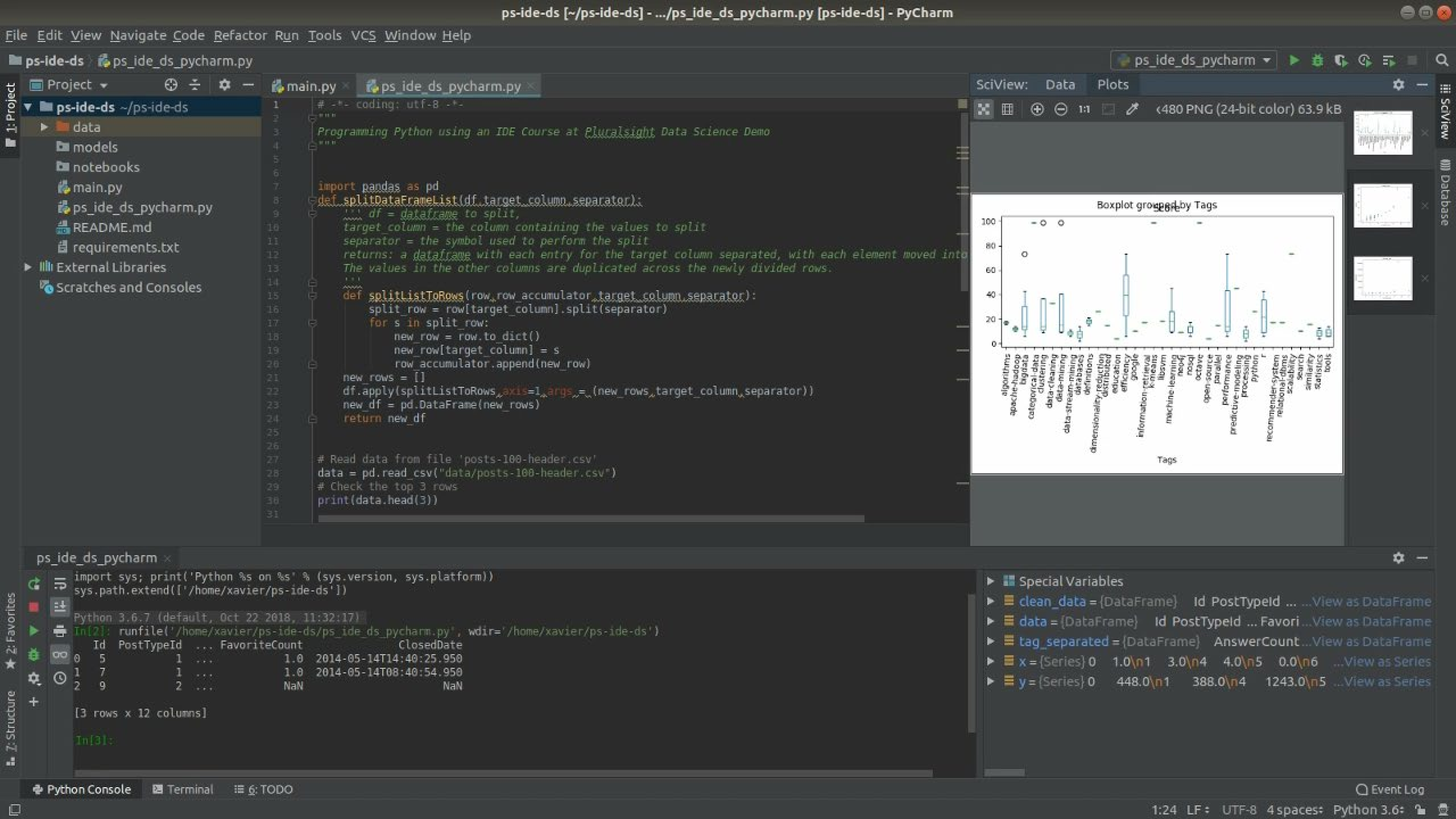
Support for

- Interactive scientific computing
- And data visualization
- Requires numpy and matplotlib

Scientific Mode in PyCharm

Provides

- IPython console
- Documentation tool window
- DataFrame/Array view
- SciView



```
# -*- coding: utf-8 -*-
"""
Programming Python using an IDE Course at Pluralsight Data Science Demo
"""


import pandas as pd
def splitDataFrameList(df,target_column,separator):
    ''' df = dataframe to split,
    target_column = the column containing the values to split
    separator = the symbol used to perform the split
    returns: a dataframe with each entry for the target column separated, with each element moved into
    The values in the other columns are duplicated across the newly divided rows.
    '''
    def splitListToRows(row,row_accumulator,target_column,separator):
        split_row = row[target_column].split(separator)
        for s in split_row:
            new_row = row.to_dict()
            new_row[target_column] = s
            row_accumulator.append(new_row)
    new_rows = []
    df.apply(splitListToRows,axis=1,args = (new_rows,target_column,separator))
    new_df = pd.DataFrame(new_rows)
    return new_df


# Read data from file 'posts-100-header.csv'
data = pd.read_csv("data/posts-100-header.csv")
# Check the top 3 rows
print(data.head(3))
```

```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/home/xavier/ps-ide-ds'])

Python 3.6.7 (default, Oct 22 2018, 11:32:17)
In[2]: runfile('/home/xavier/ps-ide-ds/ps_ide_ds_pycharm.py', wdir='/home/xavier/ps-ide-ds')
   Id  PostTypeId  ... FavoriteCount               ClosedDate
0   5           1  ...           1.0  2014-05-14T14:40:25.950
1   7           1  ...           1.0  2014-05-14T08:40:54.950
2   9           2  ...           NaN                      NaN

[3 rows x 12 columns]

In[3]:
```

```
Special Variables
clean_data = {DataFrame} Id PostTypeId ... ...View as DataFrame
data = {DataFrame} Id PostTypeId ... Favori ...View as DataFrame
tag_separated = {DataFrame} AnswerCount ...View as DataFrame
x = {Series} 0    1.0\n1    3.0\n4    4.0\n5    0.0\n6    ...View as Series
y = {Series} 0    448.0\n1    388.0\n4    1243.0\n5    ...View as Series
```

Spyder

An IDE built for Scientific Python

- IDE for Data Science

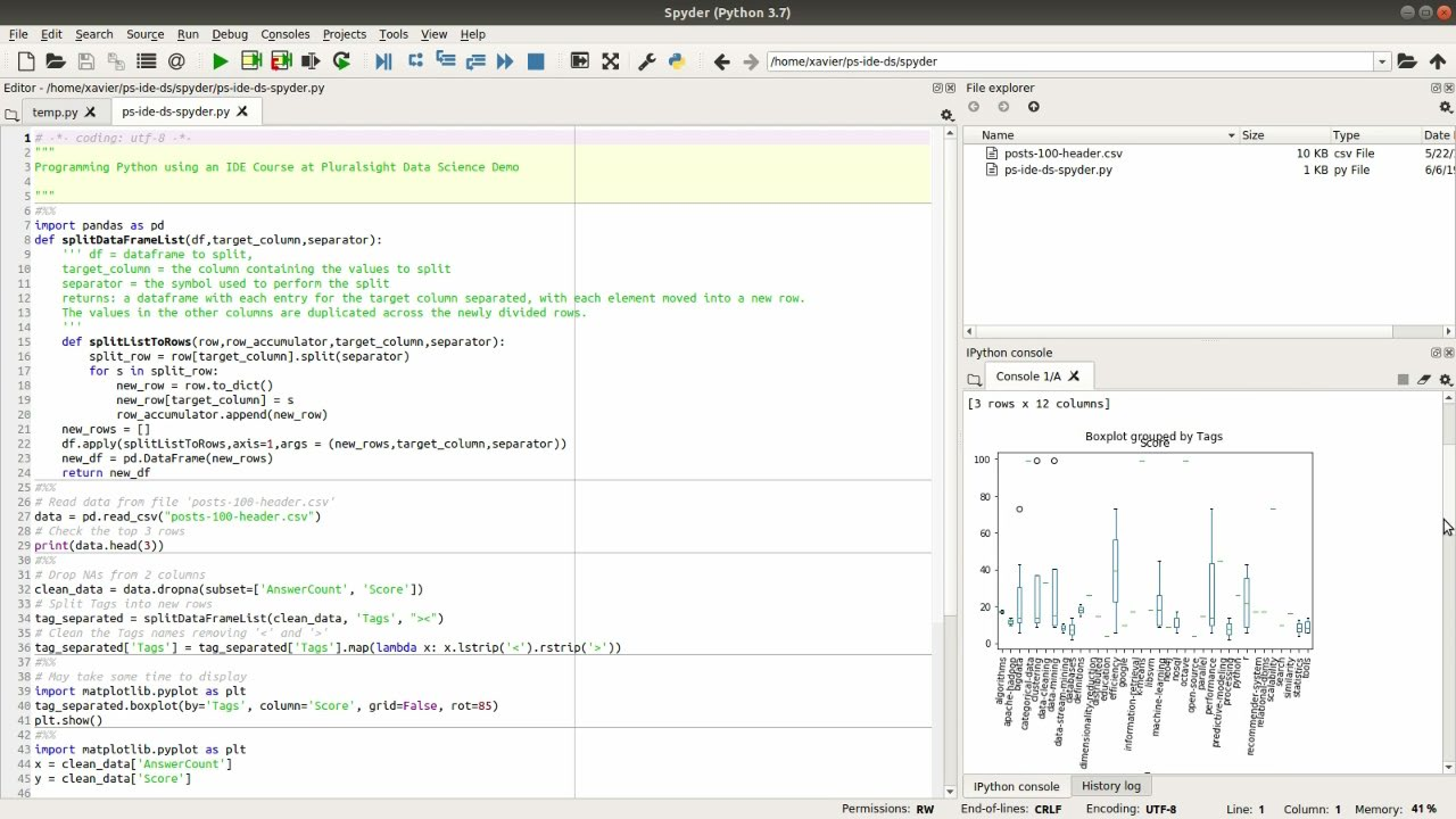
Designed by and for

- Scientists, engineers, and data analysts

Powerful features

IPython console and variable explorer

Extended via Spyder Notebook, Spyder Terminal…



```python
# -*- coding: utf-8 -*-
"""
Programming Python using an IDE Course at Pluralsight Data Science Demo

"""
#%%
import pandas as pd
def splitDataFrameList(df,target_column,separator):
    ''' df = dataframe to split,
    target_column = the column containing the values to split
    separator = the symbol used to perform the split
    returns: a dataframe with each entry for the target column separated, with each element moved into a new row.
    The values in the other columns are duplicated across the newly divided rows.
    '''
    def splitListToRows(row,row_accumulator,target_column,separator):
        split_row = row[target_column].split(separator)
        for s in split_row:
            new_row = row.to_dict()
            new_row[target_column] = s
            row_accumulator.append(new_row)
    new_rows = []
    df.apply(splitListToRows,axis=1,args = (new_rows,target_column,separator))
    new_df = pd.DataFrame(new_rows)
    return new_df
#%%
# Read data from file 'posts-100-header.csv'
data = pd.read_csv("posts-100-header.csv")
# Check the top 3 rows
print(data.head(3))
#%%
# Drop NAs from 2 columns
clean_data = data.dropna(subset=['AnswerCount', 'Score'])
# Split Tags into new rows
tag_separated = splitDataFrameList(clean_data, 'Tags', "><")
# Clean the Tags names removing '<' and '>'
tag_separated['Tags'] = tag_separated['Tags'].map(lambda x: x.lstrip('<').rstrip('>'))
#%%
# May take some time to display
import matplotlib.pyplot as plt
tag_separated.boxplot(by='Tags', column='Score', grid=False, rot=85)
plt.show()
#%%
import matplotlib.pyplot as plt
x = clean_data['AnswerCount']
y = clean_data['Score']

```

Name	Size	Type	Date
posts-100-header.csv	10 KB	csv File	5/22/
ps-ide-ds-spyder.py	1 KB	py File	6/6/1

```
[3 rows x 12 columns]
```

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 1 Column: 1 Memory: 41 %

Jupyter Notebook

Open-source web application

- Evolved from IPython

Create and share documents

- Code, equations, visualizations and text

Jupyter Notebook

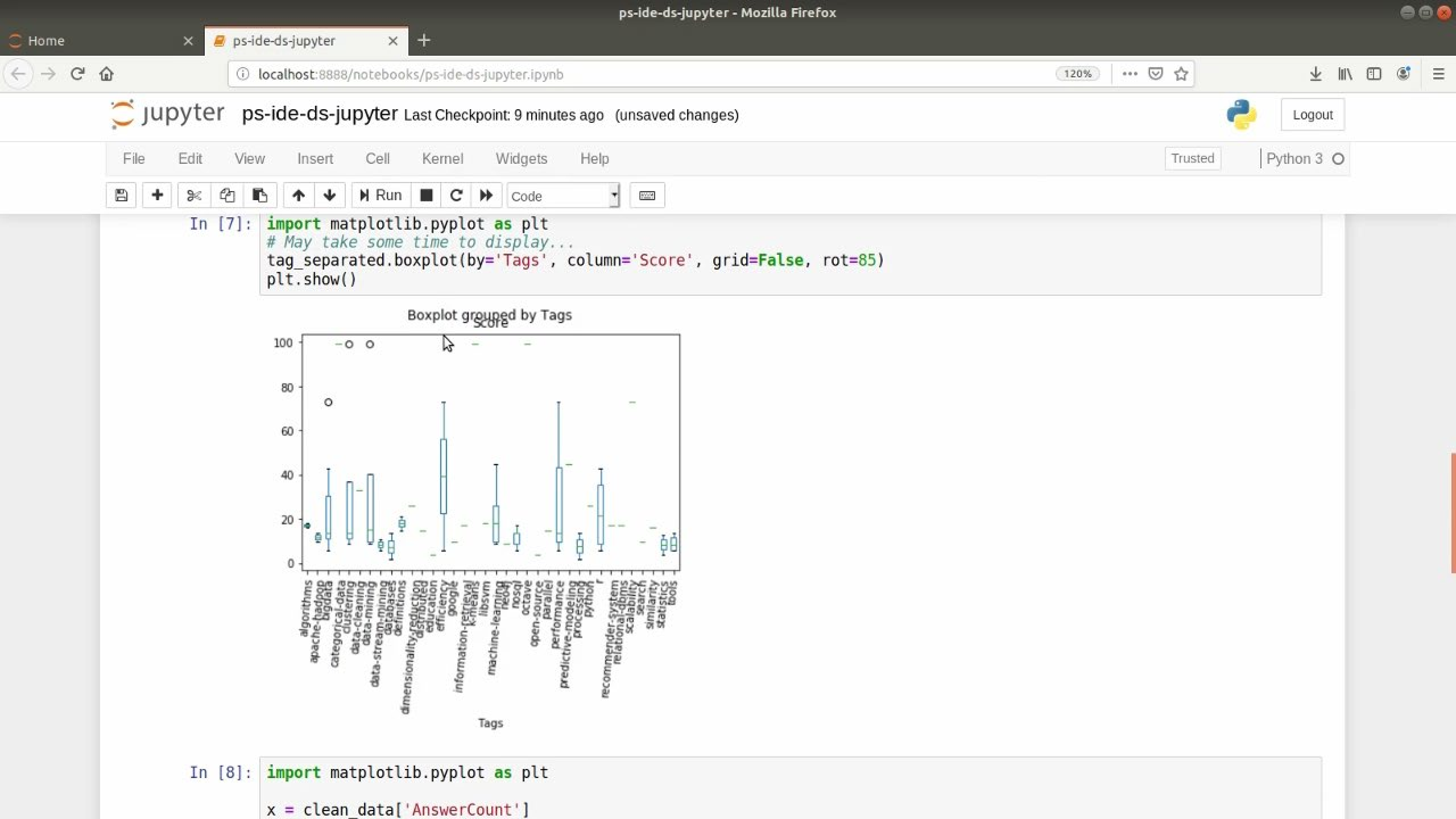
For data cleansing and transformation

Numerical simulation

Statistical modeling

Data visualization

Machine learning



ps-ide-ds-jupyter - Mozilla Firefox
Home
ps-ide-ds-jupyter
localhost:8888/notebooks/ps-ide-ds-jupyter.ipynb
120%
jupyter
ps-ide-ds-jupyter Last Checkpoint: 9 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 ○
Run
Code
In [7]:

```
import matplotlib.pyplot as plt
# May take some time to display...
tag_separated.boxplot(by='Tags', column='Score', grid=False, rot=85)
plt.show()
```

```
import matplotlib.pyplot as plt

x = clean_data['AnswerCount']
```

Apache Zeppelin

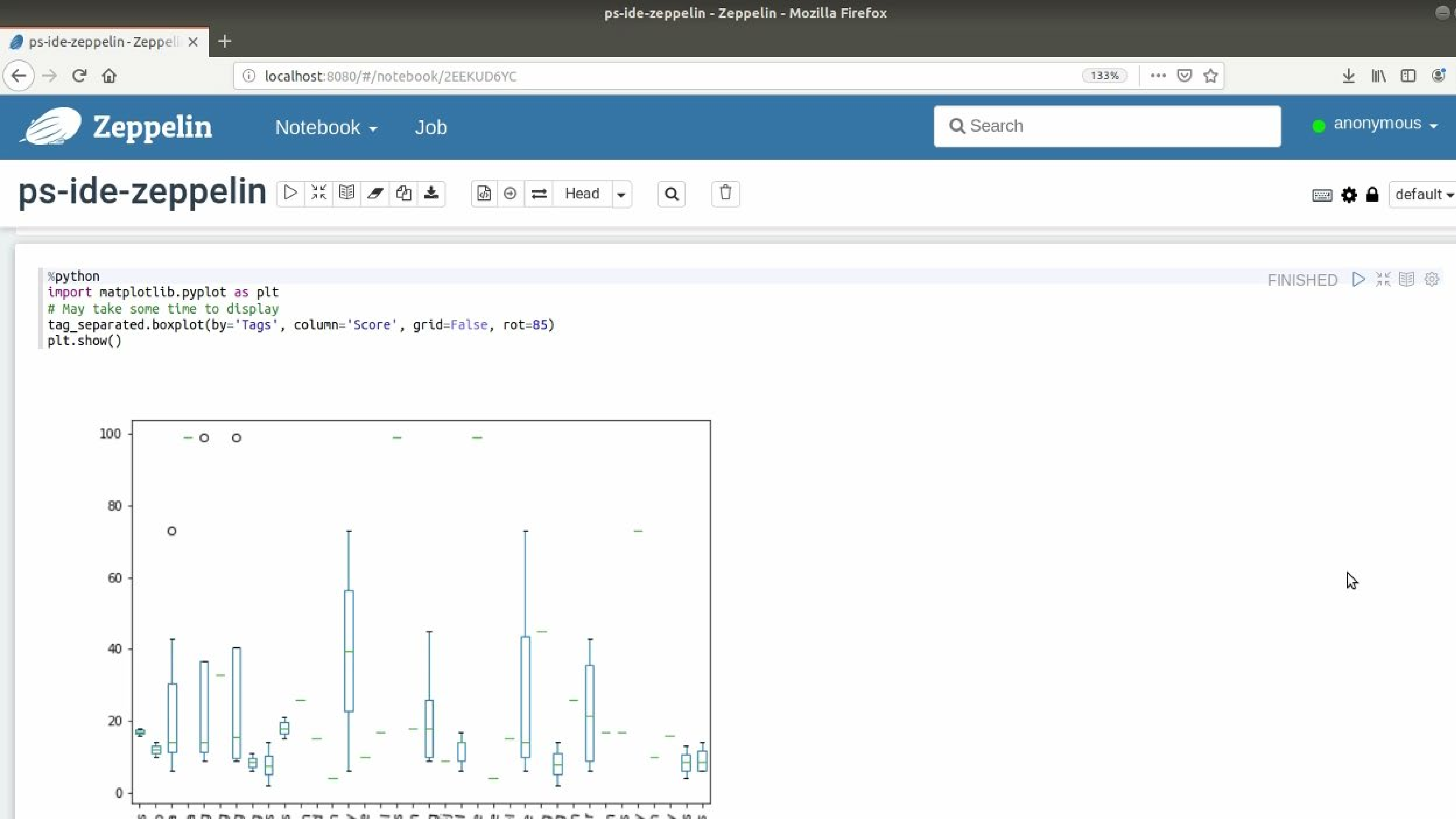
Web-based notebook

Data-drive interactive data analytics

Collaborative documents

- Python
- SQL, Scala, shell commands…

Single user or multi-user


ps-ide-zeppelin - Zeppelin - Mozilla Firefox
ps-ide-zeppelin - Zeppeli
localhost:8080/#/notebook/2EEKUD6YC
133%
Zeppelin
Notebook
Job
Search
anonymous
ps-ide-zeppelin
Head
default
FINISHED
100
80
60
40
20
0

```
%python
import matplotlib.pyplot as plt
# May take some time to display
tag_separated.boxplot(by='Tags', column='Score', grid=False, rot=85)
plt.show()
```

Cloudera Data Science Workbench

IPython kernel-based IDE

Python development with data

- Also Scala, and R

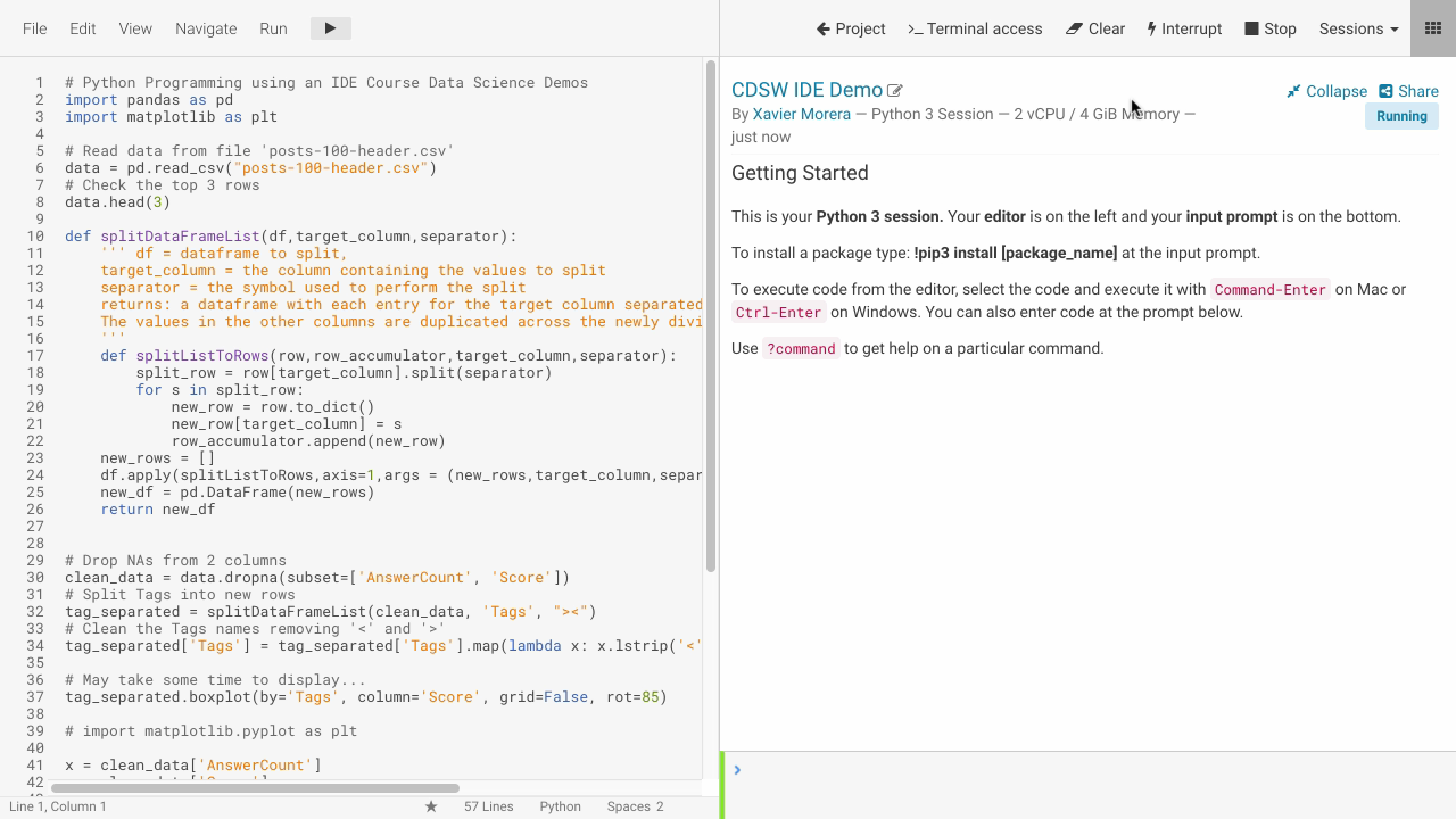
Cloudera Data Science Workbench

Platform for collaborative data science

- At scale
- Self-service
- Leverages Git for collaboration

Machine-learning focused

- Quickly deploy models, with confidence
- Run experiments



```
# Python Programming using an IDE Course Data Science Demos
import pandas as pd
import matplotlib as plt

# Read data from file 'posts-100-header.csv'
data = pd.read_csv("posts-100-header.csv")
# Check the top 3 rows
data.head(3)

def splitDataFrameList(df,target_column,separator):
    ''' df = dataframe to split,
    target_column = the column containing the values to split
    separator = the symbol used to perform the split
    returns: a dataframe with each entry for the target column separated
    The values in the other columns are duplicated across the newly divi
    '''
    def splitListToRows(row,row_accumulator,target_column,separator):
        split_row = row[target_column].split(separator)
        for s in split_row:
            new_row = row.to_dict()
            new_row[target_column] = s
            row_accumulator.append(new_row)
    new_rows = []
    df.apply(splitListToRows,axis=1,args = (new_rows,target_column,separ
    new_df = pd.DataFrame(new_rows)
    return new_df


# Drop NAs from 2 columns
clean_data = data.dropna(subset=['AnswerCount', 'Score'])
# Split Tags into new rows
tag_separated = splitDataFrameList(clean_data, 'Tags', "><")
# Clean the Tags names removing '<' and '>'
tag_separated['Tags'] = tag_separated['Tags'].map(lambda x: x.lstrip('<'

# May take some time to display...
tag_separated.boxplot(by='Tags', column='Score', grid=False, rot=85)

# import matplotlib.pyplot as plt

x = clean_data['AnswerCount']
```

CDSW IDE Demo

By Xavier Morera — Python 3 Session — 2 vCPU / 4 GiB Memory — just now

Getting Started

This is your **Python 3 session.** Your **editor** is on the left and your **input prompt** is on the bottom.

To install a package type: **!pip3 install [package_name]** at the input prompt.

To execute code from the editor, select the code and execute it with `Command-Enter` on Mac or `Ctrl-Enter` on Windows. You can also enter code at the prompt below.

Use `?command` to get help on a particular command.

Takeaway

Python IDE for Data Science

- Notebook

Features to enhance working with data

- IPython console
- Execution by cell or paragraph
- Better ways to inspect data
- Embedded visualizations
- Available large amounts of compute & memory

Takeaway

IDE

- PyCharm
- Spyder

Notebook

- Jupyter Notebook
- Apache Zeppelin

Cloudera Data Science Workbench