

Python Quant Platform

Web-based Financial Analytics and
Rapid Financial Engineering with Python

Yves Hilpisch



The **Python Quant Platform** offers
Web-based, scalable, collaborative
financial analytics and
rapid financial engineering for
individuals, teams and companies.

<http://quant-platform.com>

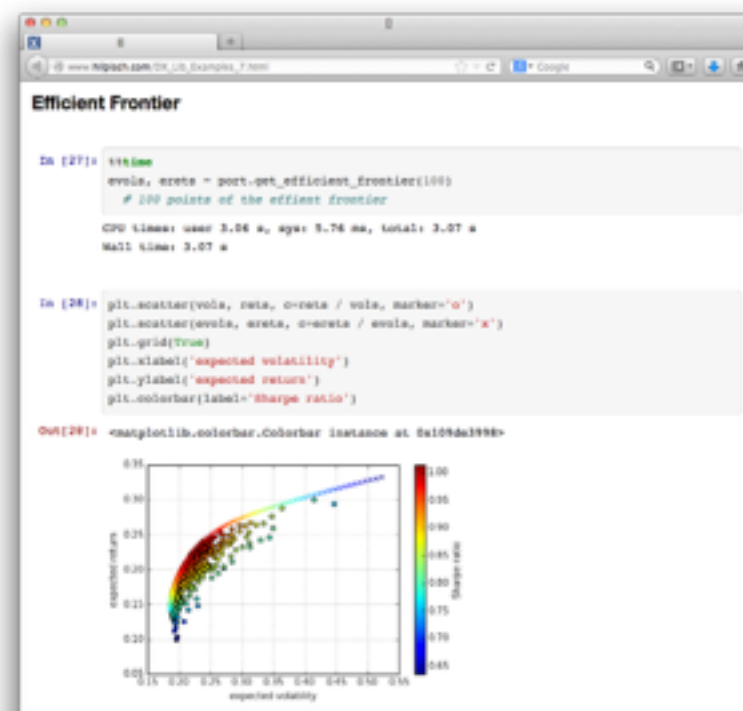
Python Quant Platform – 1

The Analytical Core Components

DEXISION
GUI-based
Financial Engineering



IPython
Interactive, Python-based
Financial Analytics



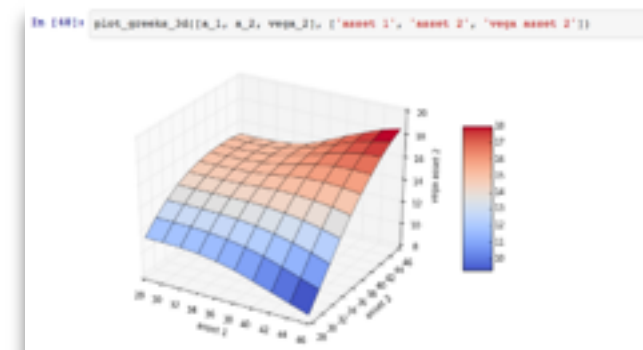
DX Analytics
Python-based
Analytics Library

```
In [1]: # DX Analytics Library
# Python-based Derivative Analytics Library
# valuation classes
# (c) 2014 New Relic

from IPython import Interactive
from IPython Notebook import Interactive

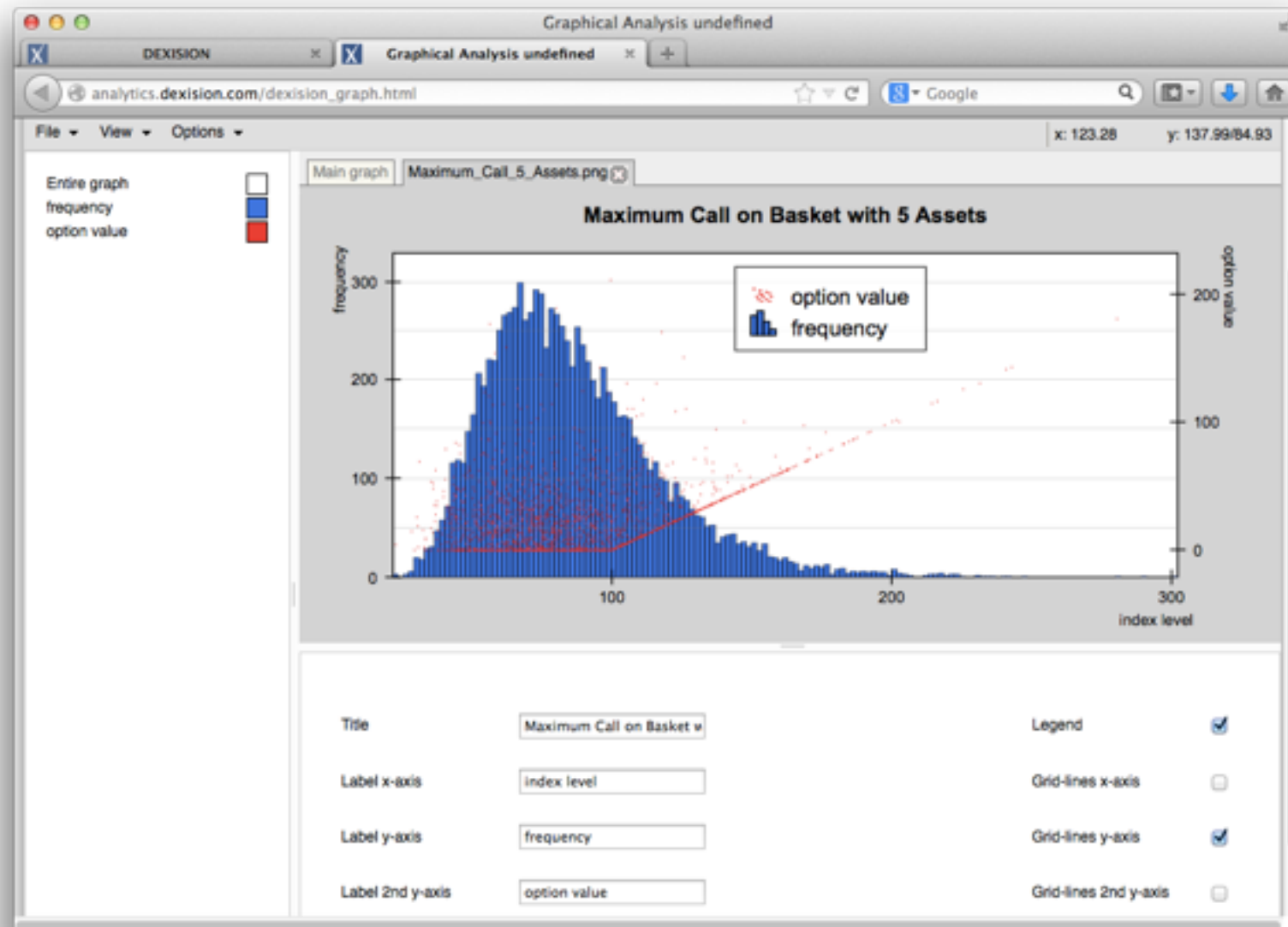
# Basic single factor valuation class
class valuation_class(object):
    """ Basic class for single factor valuation """

    Attributes:
    name: string
    name of the object
    model: object
    instance of model class
    env: object
    instance of market environment
    market environment data for valuation
    payoff: float
    payoff in string
    derivatives: list
    derivatives in python syntax
```



DEXISION

GUI-based rapid financial engineering

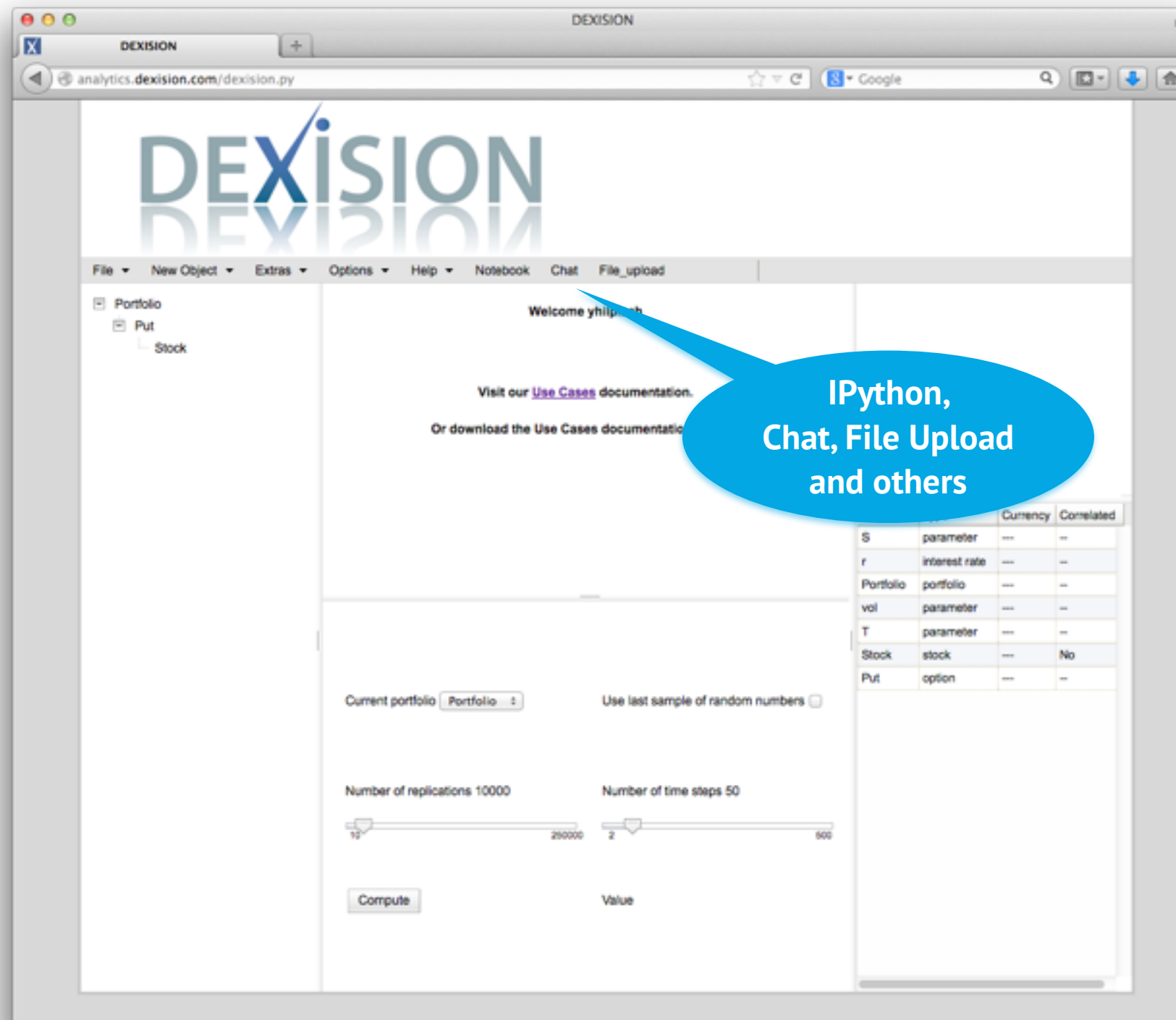


www.hilpisch.com/dexision.mov

www.derivatives-analytics.com

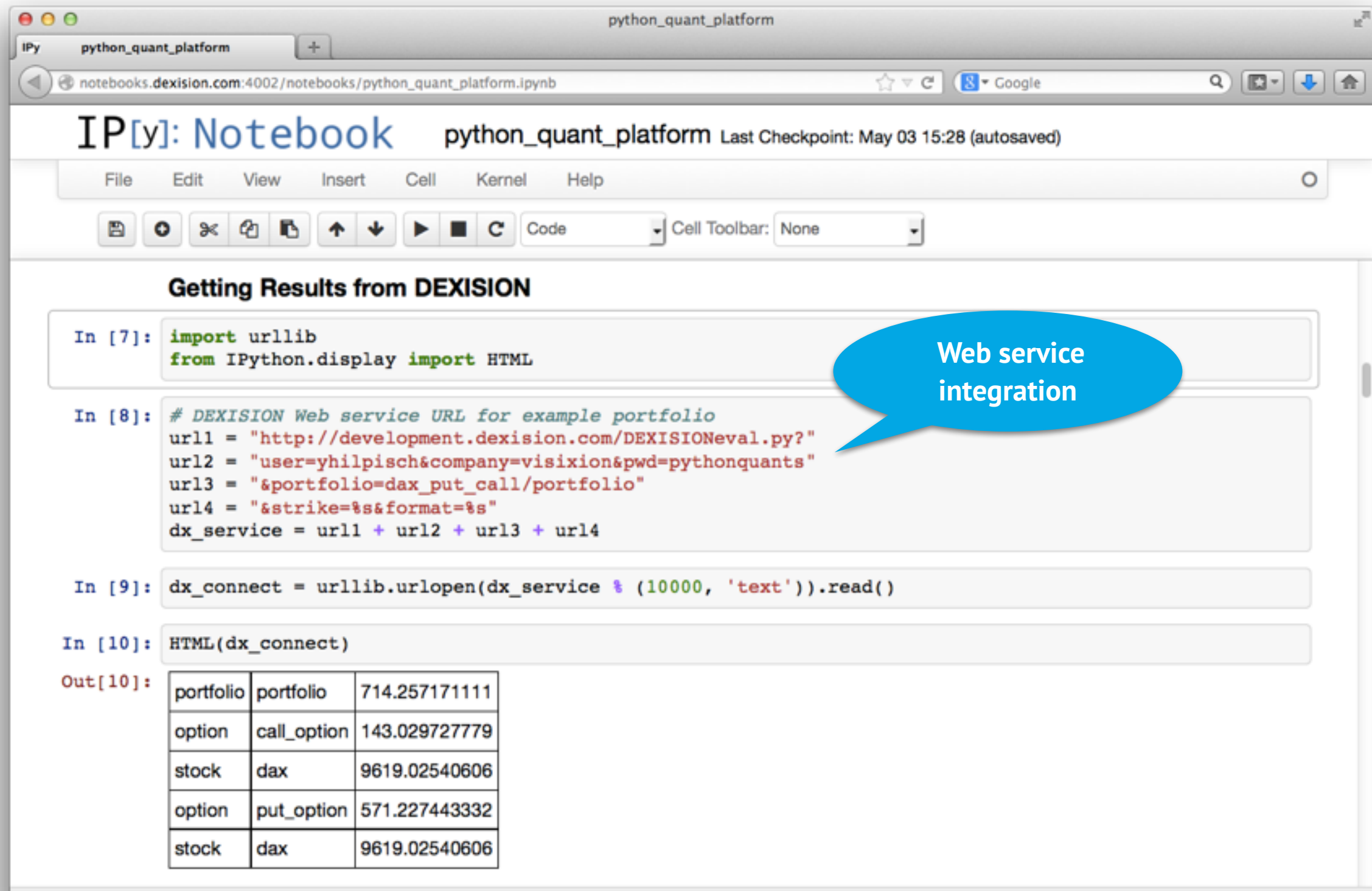
Integrated Technologies

DEXISION is integrated with IPython and other Apps



Integrated Technologies

Getting analytics results in IPython from DEXISION



The screenshot shows an IPython Notebook window titled "python_quant_platform". The browser address bar shows the URL "notebooks.dexision.com:4002/notebooks/python_quant_platform.ipynb". The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with various icons. The notebook content is titled "Getting Results from DEXISION" and contains four input cells with Python code. A blue speech bubble with the text "Web service integration" points to the code in the second input cell. The output of the fourth cell is displayed as a table.

```
In [7]: import urllib
        from IPython.display import HTML

In [8]: # DEXISION Web service URL for example portfolio
        url1 = "http://development.dexision.com/DEXISIONeval.py?"
        url2 = "user=yhilpisch&company=visixion&pwd=pythonquants"
        url3 = "&portfolio=dax_put_call/portfolio"
        url4 = "&strike=%s&format=%s"
        dx_service = url1 + url2 + url3 + url4

In [9]: dx_connect = urllib.urlopen(dx_service % (10000, 'text')).read()

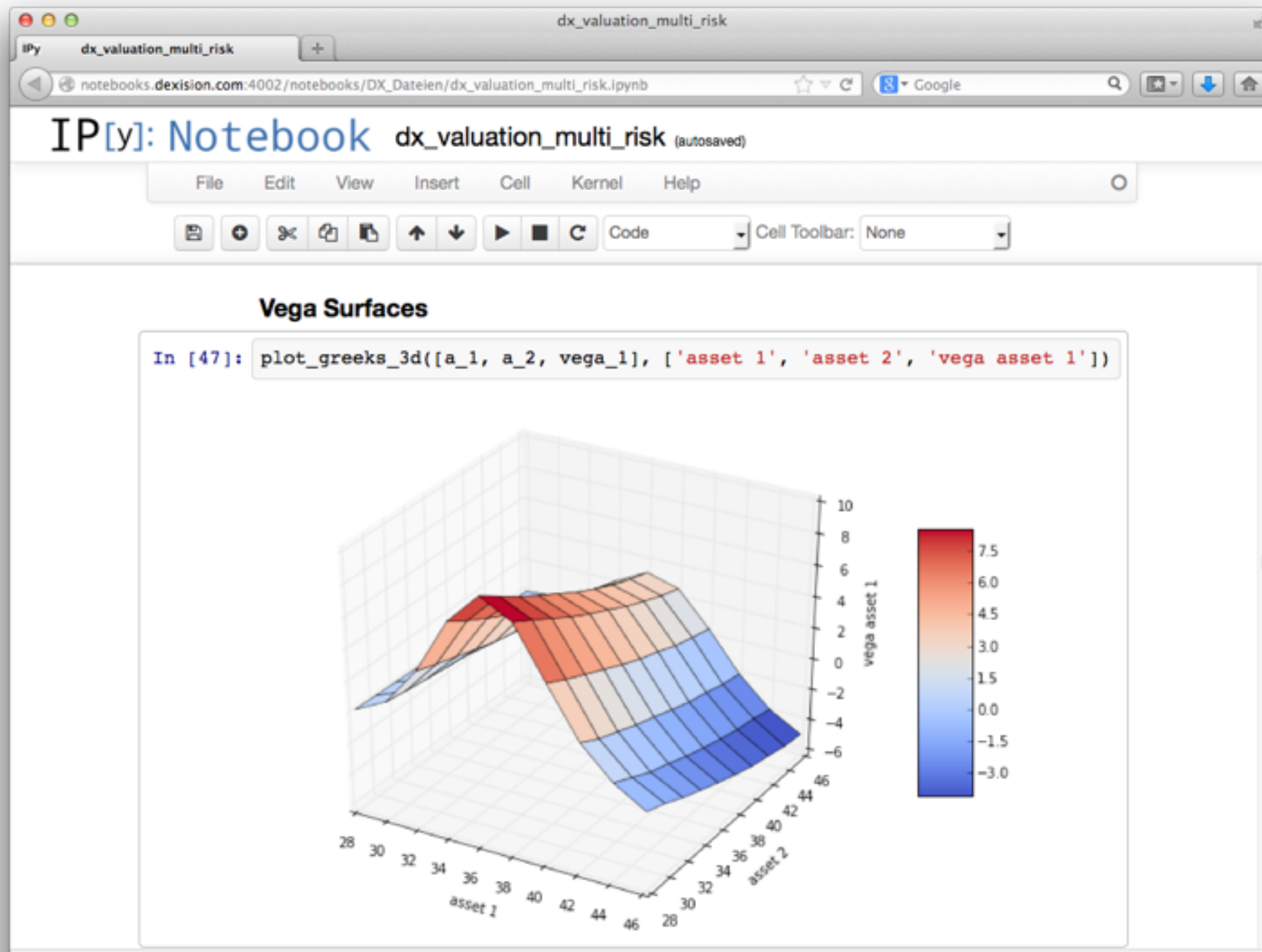
In [10]: HTML(dx_connect)
```

Out[10]:

portfolio	portfolio	714.257171111
option	call_option	143.029727779
stock	dax	9619.02540606
option	put_option	571.227443332
stock	dax	9619.02540606

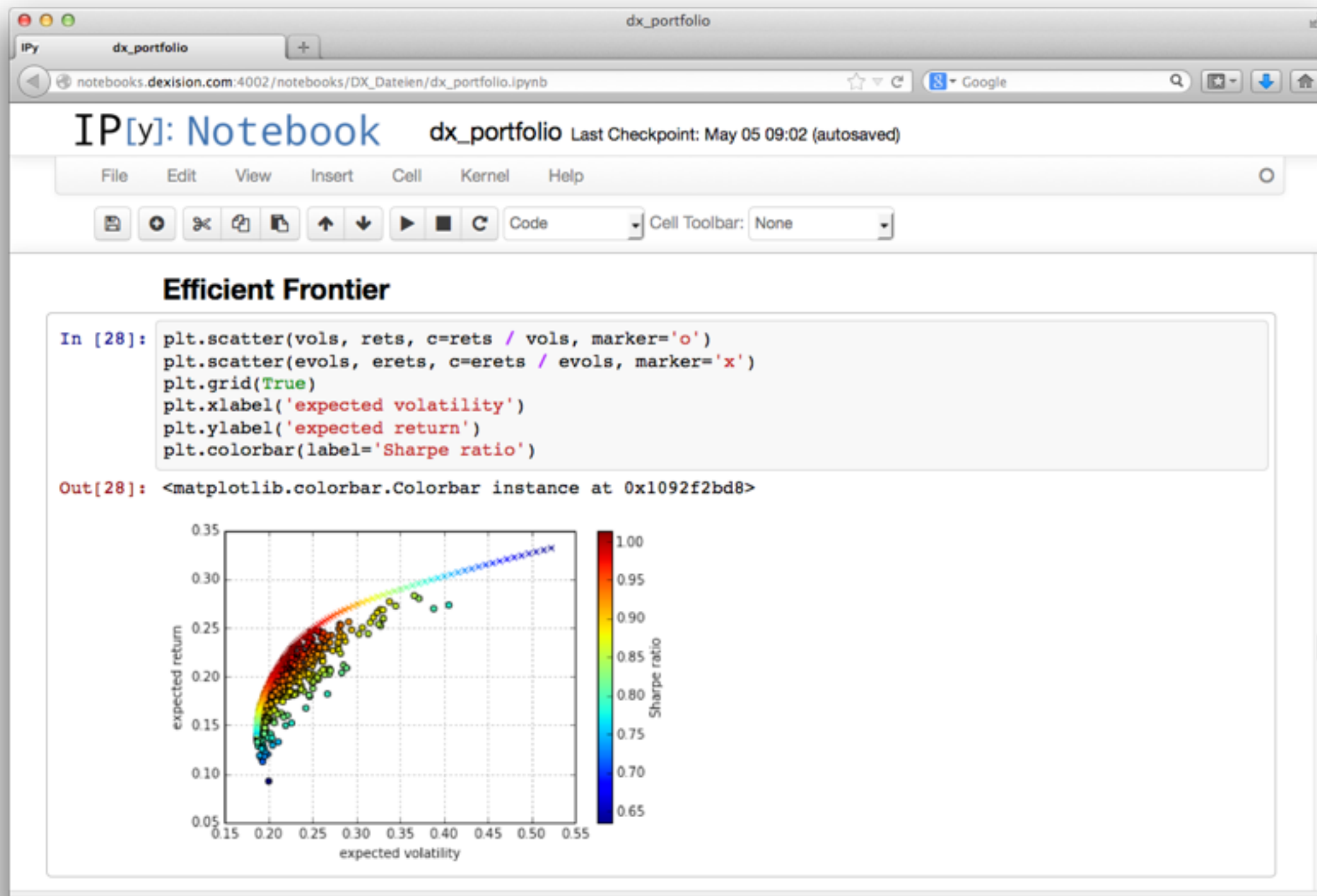
Integrated Technologies

Doing derivatives analytics with DX Analytics



Integrated Technologies

Doing portfolio analytics with DX Analytics



Python Quant Platform – 2

Infrastructure and Applications

Python

Full-Fledged
Python Stack



NumPy, SciPy,
pandas, PyTables
h5py, matplotlib,
IPython, numexpr
Cython LLVM, LLVMpy
Numba, Scikit-learn,
many more

Deployment

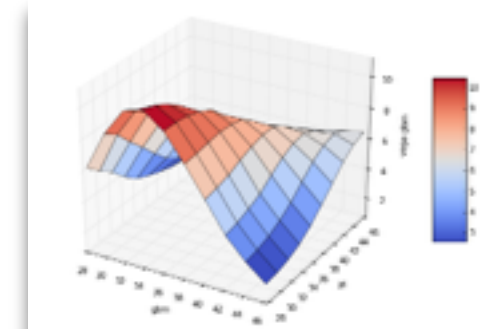
Powerful, Dedicated
Server Infrastructure



32 cores
96 GB RAM
6 TB disk

Applications

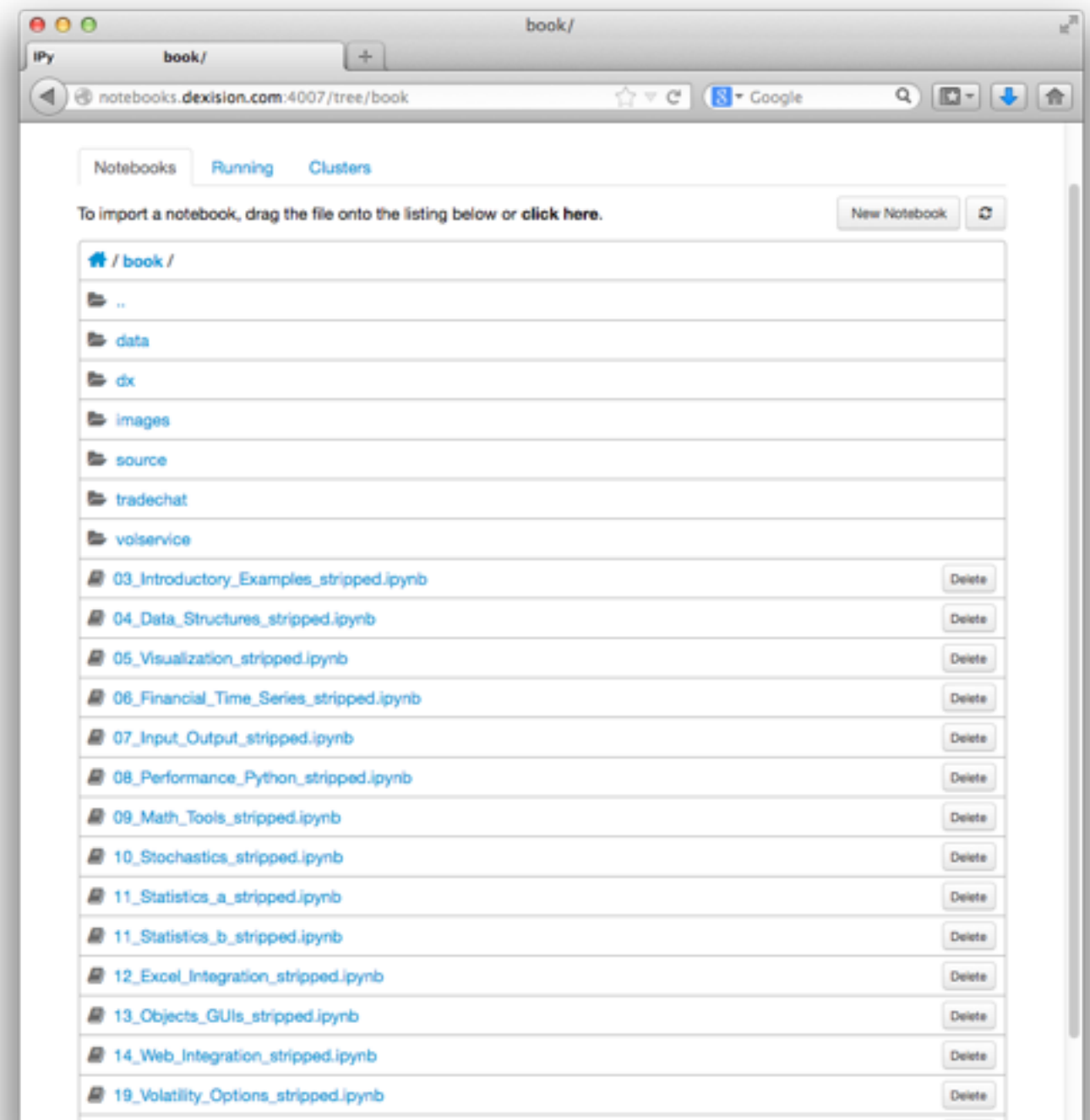
Derivatives, Portfolio,
Trading, Risk, Analysis



Derivatives Pricing
Portfolio Management
Quantitative & Algo Trading
Strategy Backtesting
Quantitative Research
Company Valuation
Value-at-Risk
Credit Value Adjustments
Time Series Analysis
Bayesian Statistics
Reporting

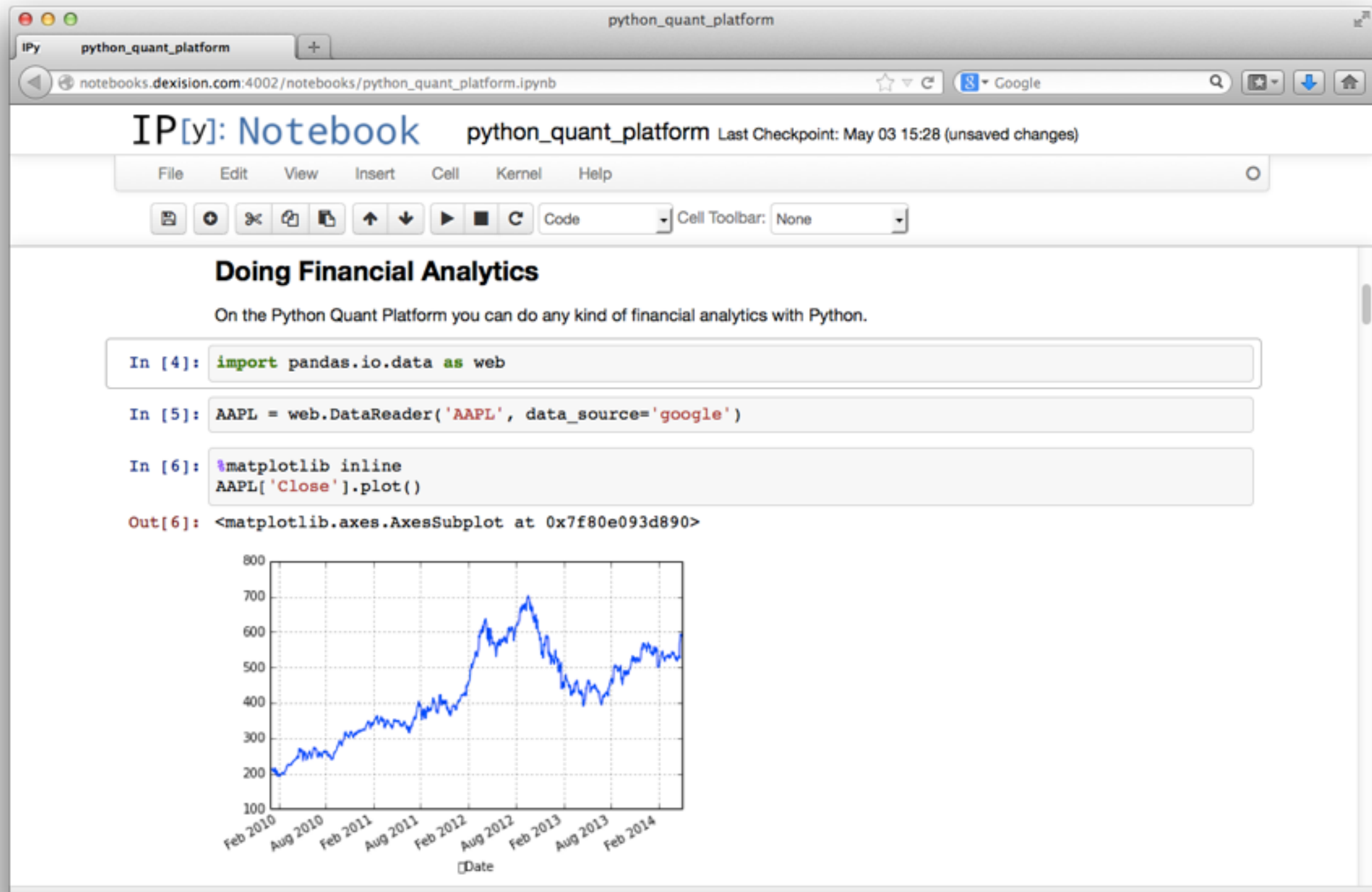
Python for Finance

Book and IPython Notebooks for training, teaching



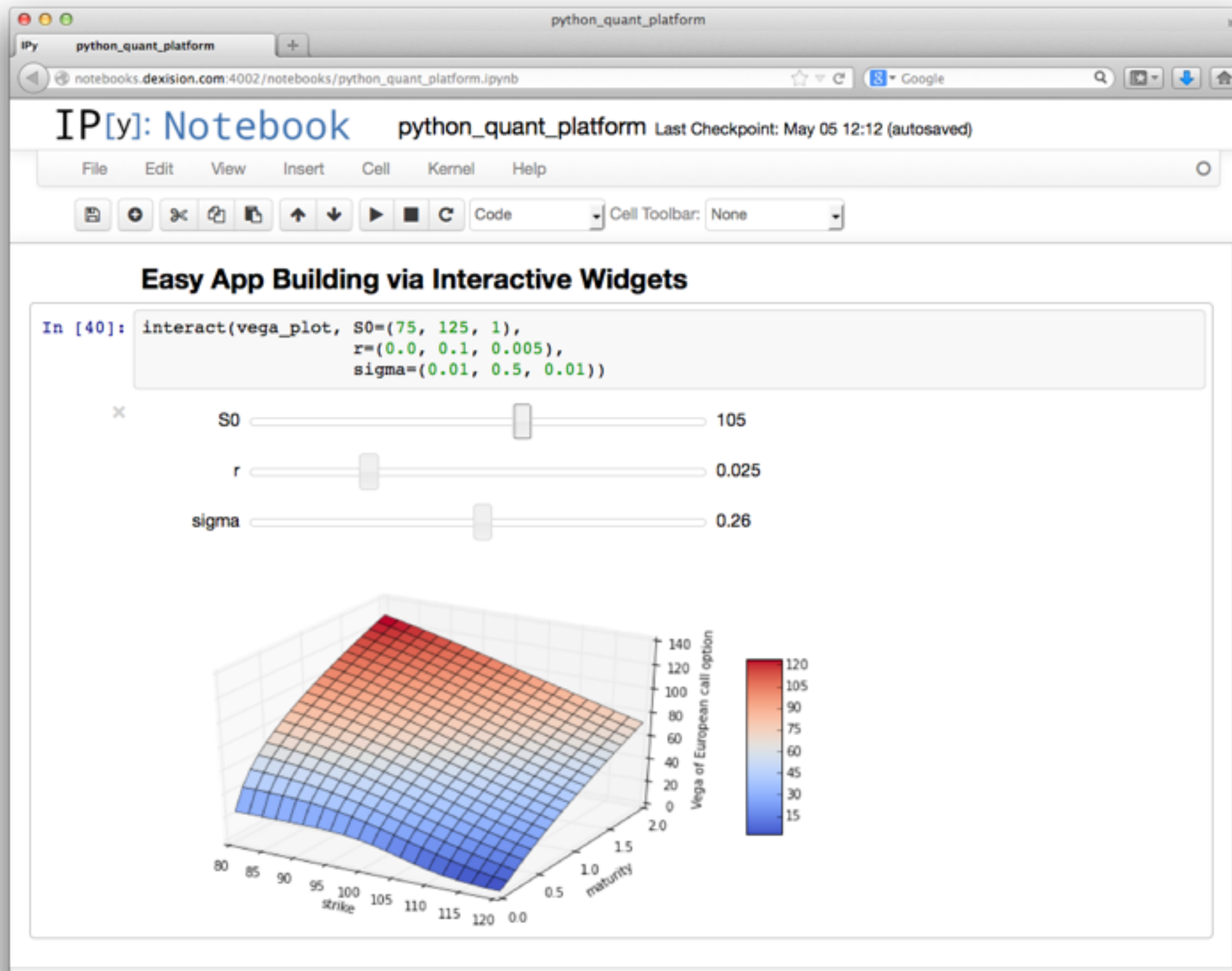
Instant Deployment

Financial analytics with Python in the browser



Easy Application Building

Python & Web apps via interactive IPython widgets



Automated Analytics Workflows

Example Workflow – VSTOXX Volatility Options

VSTOXX Index, Futures, Options



IPython

Data Management for
VSTOXX index, futures & options
– implied volatilities

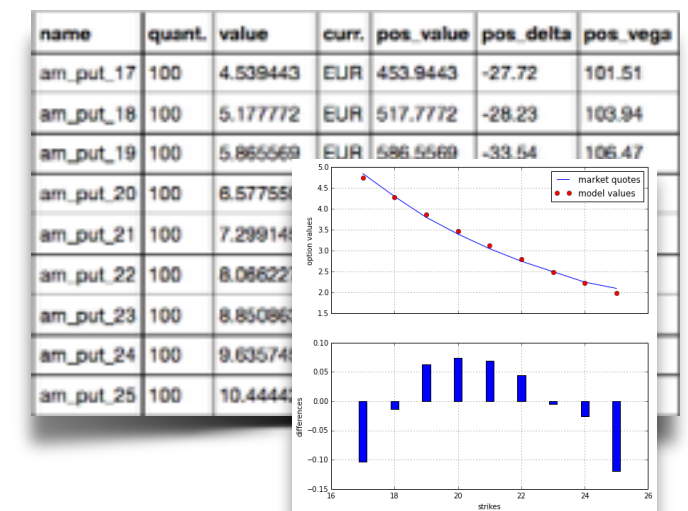
DX Analytics

Modeling and simulation of
VSTOXX index

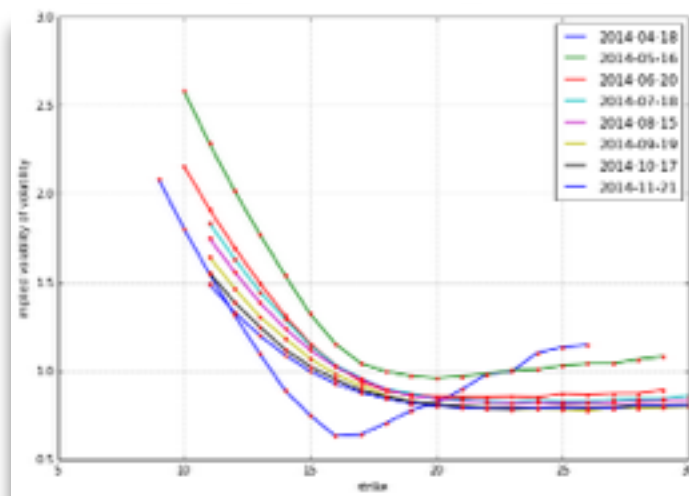
DX Analytics

Modeling of traded
VSTOXX options

OTC Derivatives Portfolio with Greeks



Volatility Surface of VSTOXX Options



DX Analytics + IPython

Calibration of VSTOXX model
to option quotes

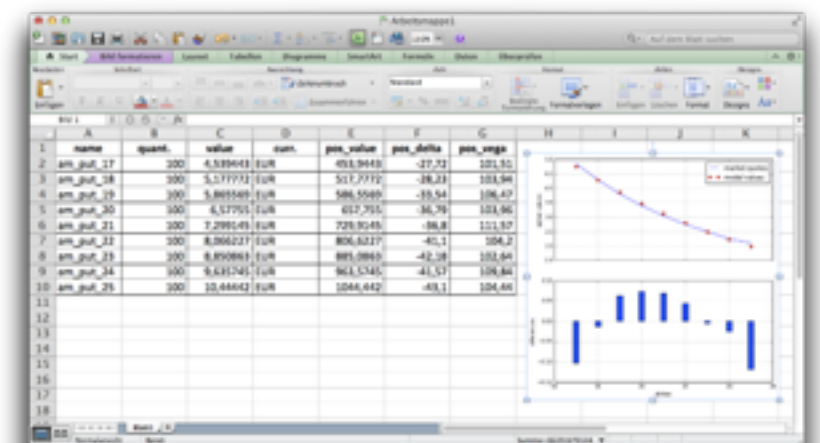
DX Analytics + IPython

Modeling and valuation of
OTC contracts on the VSTOXX
– incl. portfolio statistics and Greeks

DataNitro

Integration of functionality, results
and plots in Excel spreadsheets

Excel Spreadsheet Integration

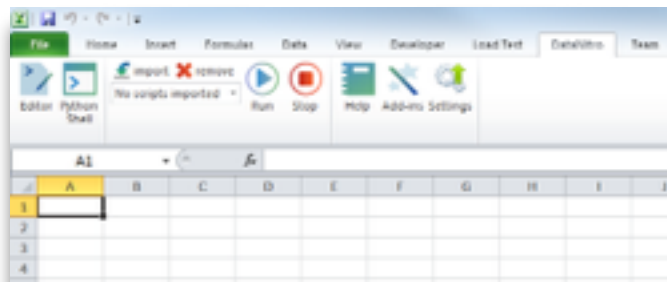


Python Quant Platform – 3

Excel & Data Integration + Collaboration

Excel

Integration via DEXISION,
pandas & xlwings/DataNitro



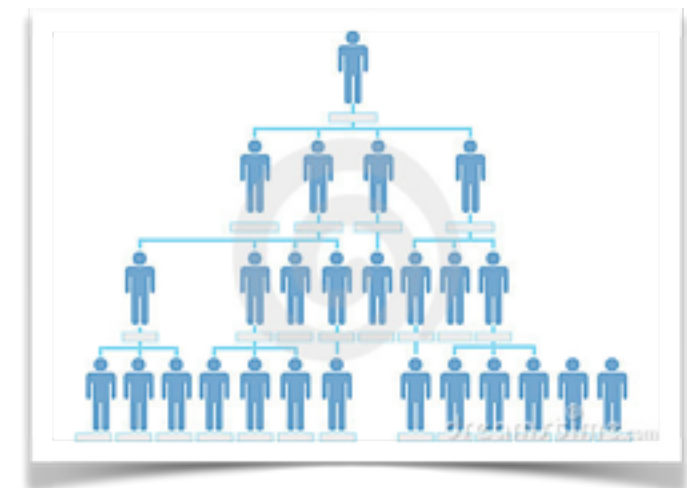
Data

Example: Eikon
Thomson Reuters



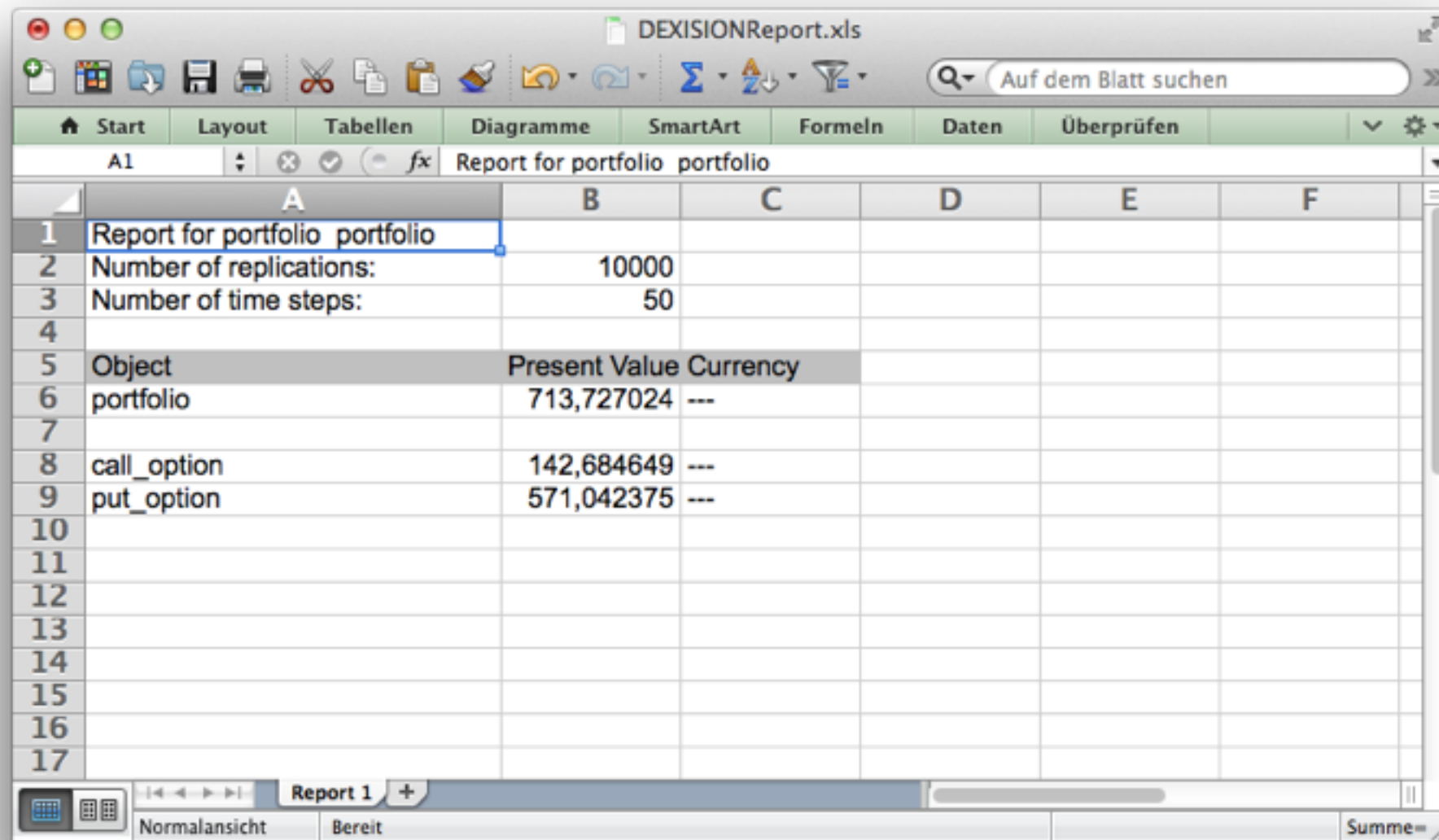
Collaboration

Management of Teams,
Rights and Roles



Excel Integration

Export results directly to Excel or use xlwings/DataNitro



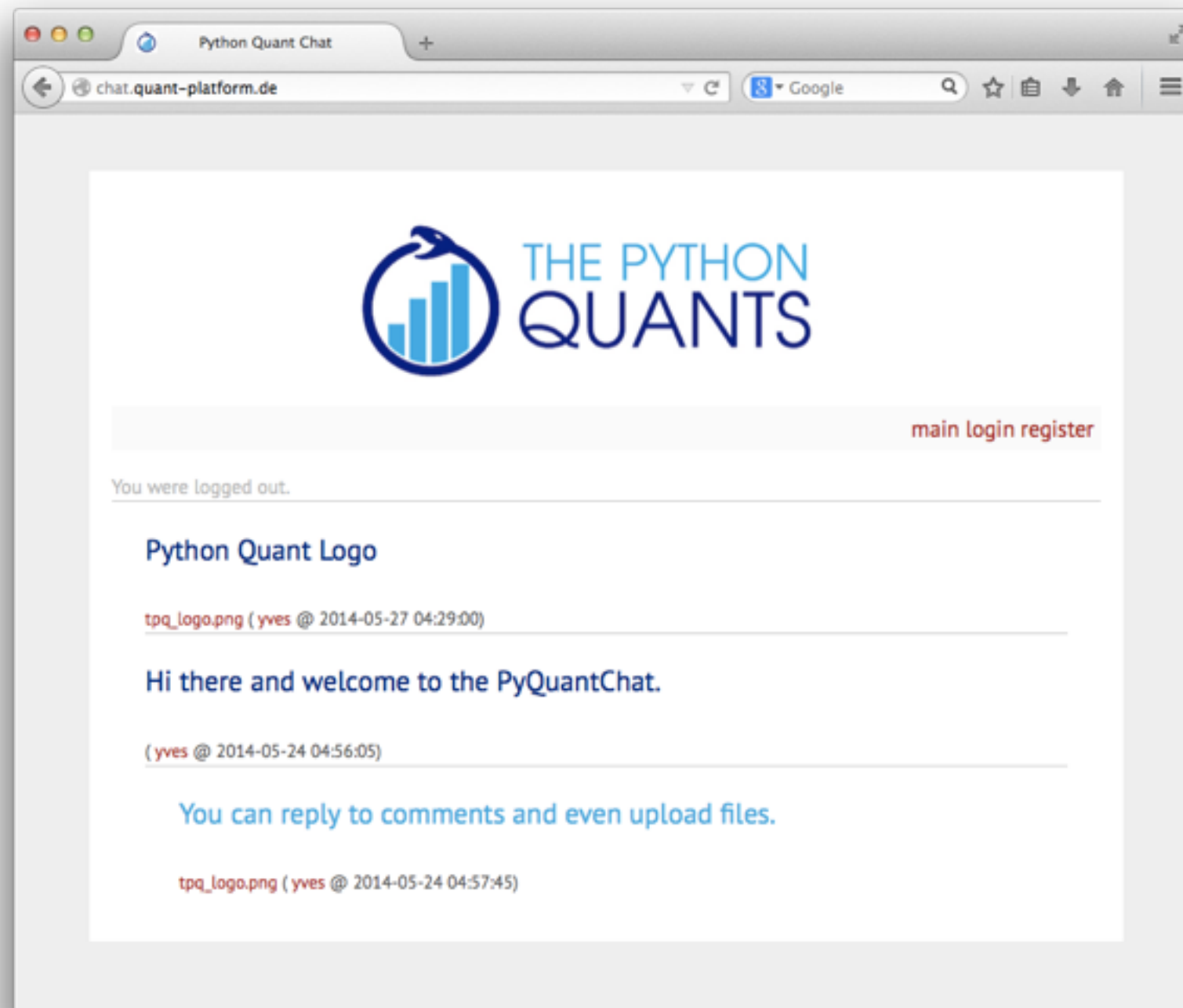
The screenshot shows an Excel spreadsheet titled "DEXISIONReport.xls". The active sheet is "Report for portfolio portfolio". The data is organized as follows:

	A	B	C	D	E	F
1	Report for portfolio portfolio					
2	Number of replications:	10000				
3	Number of time steps:	50				
4						
5	Object	Present Value	Currency			
6	portfolio	713,727024	---			
7						
8	call_option	142,684649	---			
9	put_option	571,042375	---			
10						
11						
12						
13						
14						
15						
16						
17						

The bottom of the window shows the "Report 1" sheet tab and the status bar with "Normalansicht" and "Bereit".

Collaboration

Python Quant Chat to communicate and exchange ideas



Collaboration

Notebook collaboration & sharing based on users and groups

The screenshot displays the Python Quant Platform web interface. The browser address bar shows the URL `analytics.quant-platform.com/nb/portal/login#`. The platform logo, "THE PYTHON QUANTS", is in the top left. The main header reads "PYTHON QUANT PLATFORM".

On the left sidebar, under "Notebooks", there is a "New Notebook" button and a list of existing notebooks: `training`, `Numpy_Case_Study.ipynb`, `Python_for_Finance_1.ipynb`, `Python_for_Finance_2.ipynb`, `Python_for_Finance_3.ipynb`, and `Python_for_Finance_4.ipynb`. Each notebook has a "Delete" button next to it. Below this is a "File upload" section with a "File Uploader" button.

The main workspace shows an open notebook titled `quanttraining/py4fi/Numpy_Case_Study.ipynb`. The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for saving, undo, redo, and execution. The notebook content is titled "IP[y]: Notebook Numpy_Case_Study (autosaved)".

The notebook content includes a section titled "Using NumPy & pandas for Simulation, Valuation and Risk". The text describes the Black-Scholes-Merton SDE (geometric Brownian motion) for the price process of a stock or index, given by the equation:

$$dS_t = rS_t dt + \sigma S_t dZ_t$$

where S_t is the index level at t , r is the short rate, and σ is the volatility. A discretization for this process is given by the equation:

$$S_t = S_{t-\Delta t} \exp\left(\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}z_t\right)$$

where Z_t is a Brownian motion, z_t is a standard normally distributed random variable.

The notebook also contains a code cell with the following Python code:

```
In [ ]: while 1 > 2:
        S[t] = S[t-1] * np.exp((r - 0.5 * sigma ** 2) * dt +
                                sigma * np.sqrt(dt) * npr.standard_normal(1))
```

Python Quant Platform

Being better and faster
in financial analytics and engineering.

Interactively prototype, collaborate on and share
Python-based analytics workflows and applications
across your organization.

The Python Quants Group

Currently active in Germany, New York and London

**The Python Quants
GmbH**
(Germany)

(GERMANY)

**The Python Quants
LLC**
(New York)

(NEW YORK)

The Python Quants
(London)

(LONDON)

The Python Quants Group has an international team of Python and Finance experts that helps clients to get the most out of Python technology in Quantitative Finance.

The team members have helped banks, hedge funds, asset managers and other financial institutions around the globe with Python-based technologies, projects and trainings. Areas of expertise include interactive financial analytics, derivatives analytics, financial algorithm implementation, risk management, strategy backtesting, Python-Excel integration and unified analytics infrastructures.

The Python Quants GmbH

Dr. Yves J. Hilpisch
pythonquants.com
yves@pythonquants.com
yves.hilpisch
[@dyjh](#)

