# Charting Data Visualization with Python Libraries

#### (20-min Read)

By the end of this document, you will be able to:

1. Use BQL and the pandas library to format a DataFrame for visualization
2. Build and display a time series Line Plot with bqviz
3. Build and display a cross-sectional Scatter Plot with bqviz

# BQuant Supported Libraries Choose Your Charting Weapon

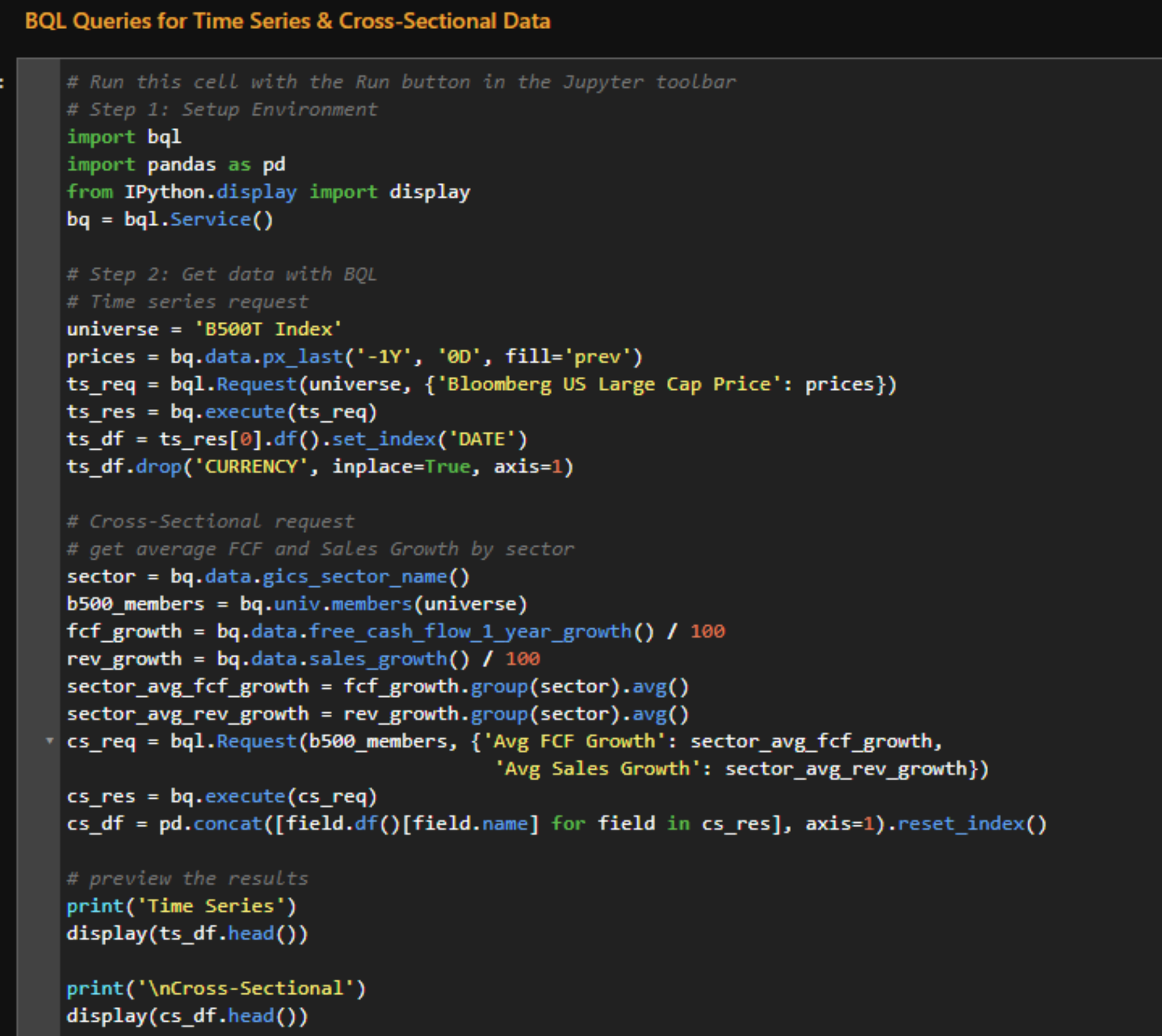
The BQuant environment supports several charting libraries, both open-source and proprietary to Bloomberg. For newcomers to Python data visualization, we cover the Bloomberg-created bqviz library in this tutorial. However, if you have experience in another charting library, feel free to use that in your apps and skip the rest of this tutorial.

The open-source libraries supported by BQuant include the following.

* [matplotlib](https://matplotlib.org/)
* [plotly](https://plotly.com/python/)
* [seaborn](https://seaborn.pydata.org/)
* [ipyleaflet](https://ipyleaflet.readthedocs.io/en/latest/) for interactive maps

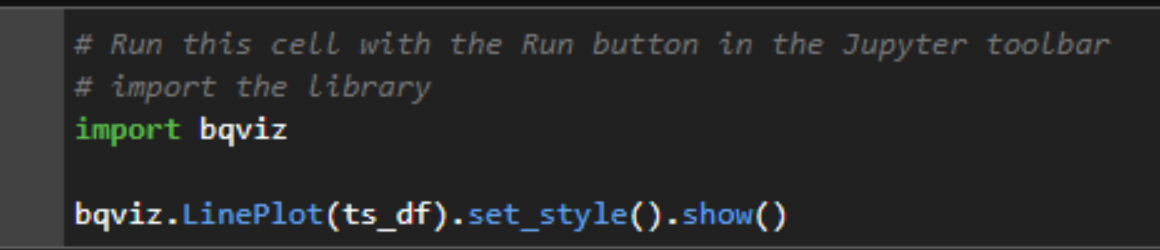
Bloomberg's proprietary charting libraries include the following.

* [bqplot](http://localhost:52923/files/exampleroot/7da582b97da6463aa0bf19221f44b28a/bqplot%20Basics.ipynb)
* [bqviz](http://localhost:52923/files/exampleroot/2f85317198594ff0a9f8bf6cb8898c3d/BQViz%20Basics.ipynb) - a wrapper library for bqplot allowing you to quickly create bqplot charts.



# Initialize a Time Series Line Plot Explore bqviz.LinePlot()

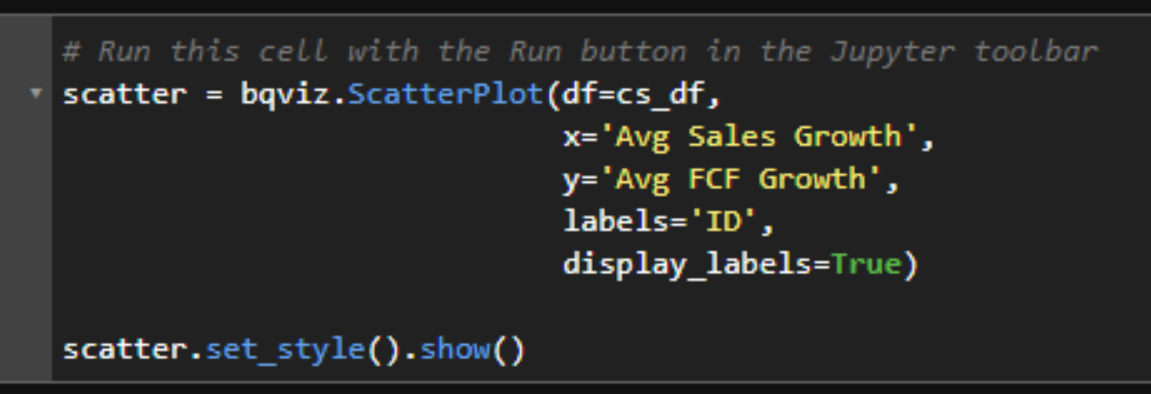
Now that we have some data to work with, we can explore the charting options available to us in bqviz. For many chart objects in the bqviz library, the only required variable is a DataFrame, meaning that we can initiate and display a chart very quickly. The set\_style() method will apply a default styling to the chart, and the show() method will render the chart to the screen.

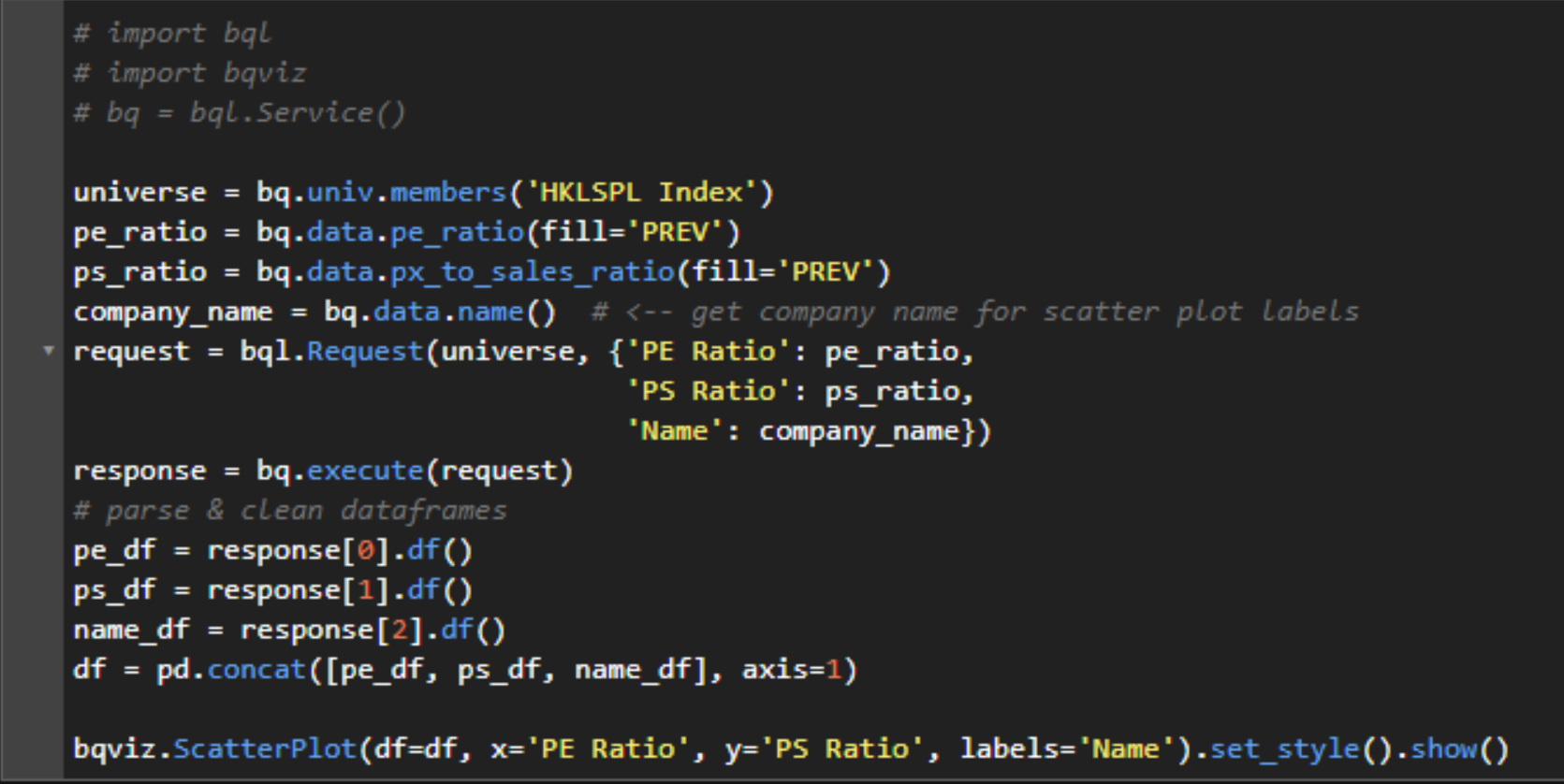
The example below prepares and displays a time series Line Plot for our S&P 500 prices. 

# Initialize a Cross-Sectional Scatter Plot Explore bqviz.ScatterPlot()

Since we made the effort to correctly format our cross-sectional DataFrame, viewing it in a bqviz.ScatterPlot can be done in just a few lines of code. The ScatterPlot requires a few additional arguments.

* x - column name for the independent variable
* y - column name for the dependent variable
* labels - while not required, specifying a column name for marker labels will help users understand your plot
* display\_labels - boolean should be set to True for labels to be displayed





# Launch: Final Project Visualize Data using DataGrids and Scatter Plots

### Project Assignment:

* Use BQL to retrieve average P/E ratios and market caps by sector for members of an equity index
* Use BQL's filter() and group() functions to limit results to market caps above $50 billion and group by sector.
* Deploy a Scatter Plot to visualize the data with market caps along the x-axis and P/E ratios on the y-axis. You may use any BQuant-supported charting library.
* Display the data in table format with a ipydatagrid.DataGrid object

##### Example Output:



