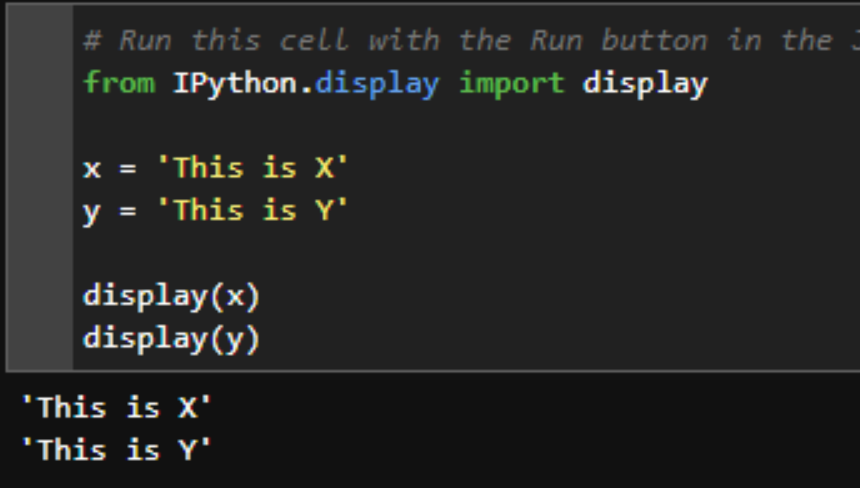
# DataGrids Display Data in an Interactive Table

Access: Launch track

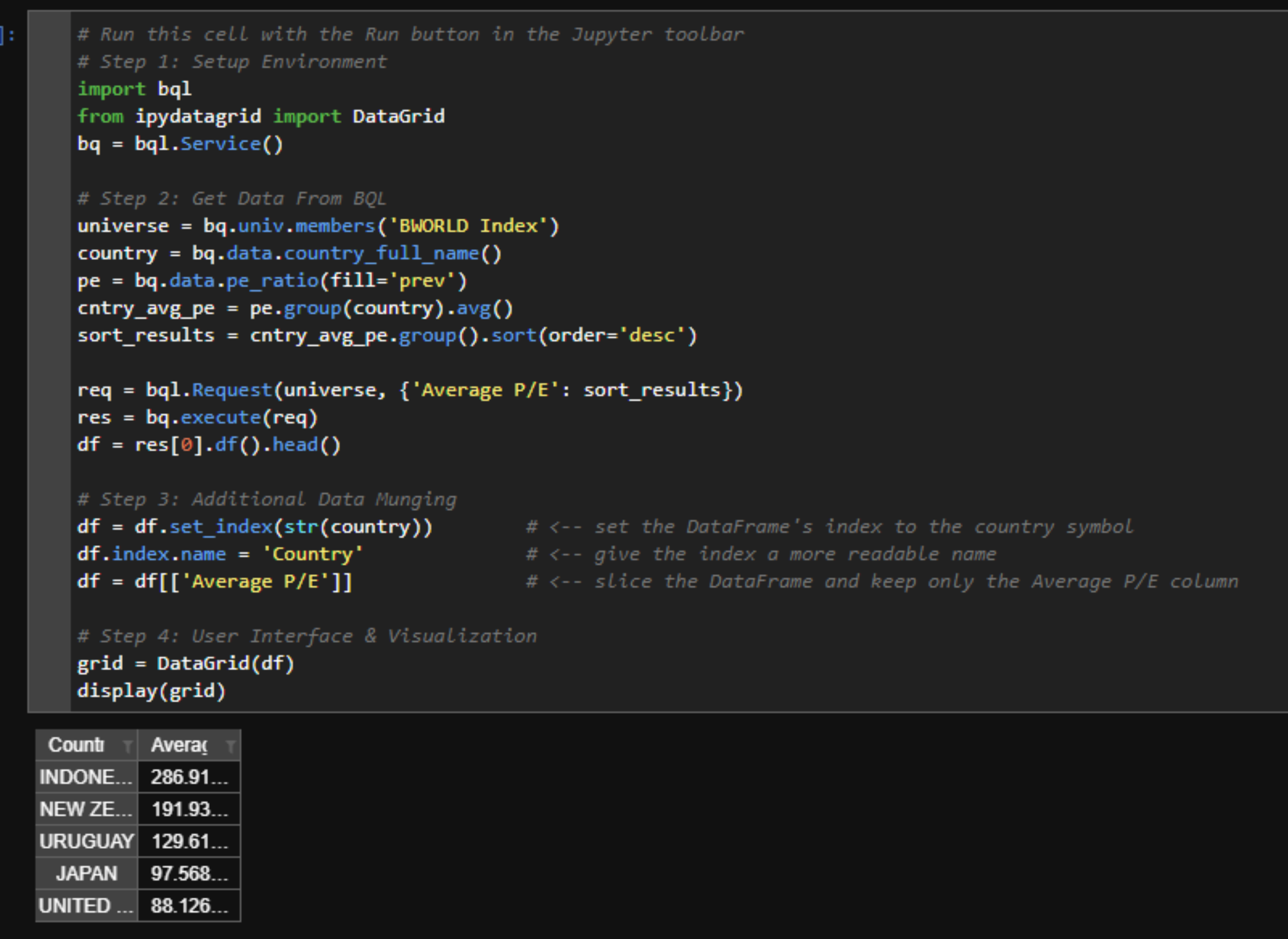
One way we can show two or more items in a Jupyter cell's output is by leveraging the IPython.display library. This library has a function named display() which will tell Jupyter to include its argument in output. We can rewrite the code example above to display both variable

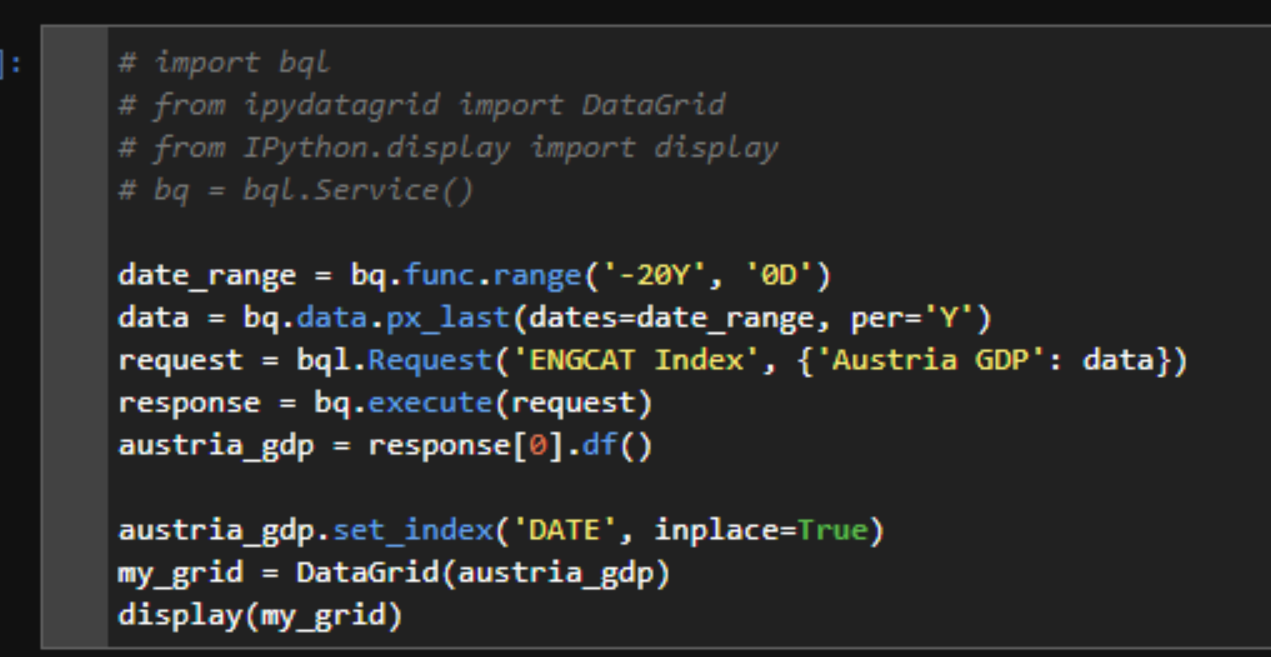


# Initialize a DataGrid Populate and Display an Interactive Table

DataGrids are UI elements that allow you to display data in an interactive table. Once displayed, the user can sort, filter, select, and even change cell values if the developer gives permission. Bloomberg has developed a Python library called ipydatagrid designed specifically for use in BQuant. Simply import ipydatagrid and pass a DataFrame to the DataGrid object to render your first grid.

In the example below, we use BQL to calculate average P/E ratios by country across members of the MSCI ACWI Index. This exercise is from the final "Try It Out" section of the previous [Launch: BQL Concepts](http://localhost:52923/files/2.%20Launch/1%20BQL%20Concepts.ipynb#dg-link-here) training. Run the cell below, and explore the resulting DataGrid. You can adjust column widths & row heights as well as sort & filter each column.





Let's fix the issues highlighted above. The following attributes of the `ipydatagrid.DataGrid` object will help.

* layout - pass a dictionary of CSS items to customize the look of the DataGrid figure. To limit the whitespace, we can reduce the height to 200 pixels.
* renderers - ipydatagrid has a TextRenderer object we can use to customize the number format displayed. We can use this same object to center align the numbers. We'll also use a second TextRenderer object to left align the country name.
* column\_widths - pass a dictionary of column names as keys and pixel integers as values to control the width of individual columns.

