**Bokehpivot Interface Tutorial**

**Given by Matt Mowers, 2018-02-07**

**Prep**

* Bring computer and remote into Orion or Scorpio
* Make sure you have access to some of your recent runs (or someone else’s)

**Intro**

* Bokeh
  + Bokeh is a python library for interactive html visualization. It allows us to build charts, maps, and other figures, where we have tools to zoom, pan, hover, save, etc
  + Bokeh also can run a server (similar to R shiny) that allows you to create your own app in python code to interact with an html document.
* Bokehpivot
  + Bokehpivot is a bokeh app I built that runs on the bokeh server used to create pivot charts based on a table of data. This works like excel pivot charts or Tableau.
  + Bokehpivot is fully integrated with ReEDS results with multi-scenario capability
* Uses of bokehpivot on ReEDS team
  + Exploring results of ReEDS runs, one or many scenarios
  + Building html/excel reports to share with stakeholders and teammates
  + Gathering and preprocessing data from one or many scenarios
* Gaps that bokehpivot does not fill
  + Building publication-ready or highly customized figures
    - Instead, Gather data with bokehpivot and output to csv/excel first, and build these figures using excel, R, python, or whatever you prefer

**Tutorial**

* Launch tool from .bat on your desktop on Scorpio or Orion
  + This executable points to the common bokehpivot folder: \\nrelqnap01d\ReEDS\bokehpivot
  + You will see the simple command line call that is made
  + For some reason the first time you do this in a day or so it takes a minute to startup (this is not true on my local machine)
* See terminal window:
  + Prints helpful messages as data is being loaded and manipulated
  + Prints errors
  + To start over (e.g. if stuck), kill this process and relaunch form the .bat file.
* Enter Data Source
  + The path can be
    - A ReEDS run folder that contains a gdxfiles folder
    - A directory that contains ReEDS run folders
    - Multiple of the above separated by | (the pipe symbol)
    - A csv file with a flat (pivotable) table of data
  + Faster path copying
    - Use Shift Right Click to copy paths to files/folders
    - Copy path from the windows explorer address bar
  + At this point, no data has been gathered from the runs, just the names of the folders.
  + Note that you can’t have two runs with the same name
* Build Report
  + Use Filter Scenarios to limit the scenarios as desired
  + Under “Build Report”, select a report, e.g. “standard\_report.py” or “standard\_compare\_report.py”
  + Base Case is required for certain reports, e.g. standard\_compare\_report.py
  + Click “Build Report” or “Build Separate Reports”
    - “Build Report” will build a single html file and single excel file with all the data. The files will be opened automatically
    - “Build Separate Reports” will build separate html files for each section of the report (still only one excel file though, since it can handle way more data without being slow). The files will not be opened automatically
  + Reports are independent processes, so don’t slow you down doing anything else, e.g. creating other reports- they can all be kicked off simultaneously
  + As we continue, bear in mind that report building is completely unaffected by the widgets that will appear beneath
* View Output Files
  + All results from building reports and exporting views will be dumped into your user folder in \\nrelqnap01d\ReEDS\bokehpivot\out
  + Your reports will each be in a directory called “report-” followed by the timestamp
* Filter scenarios
  + Used to limit the gdx data that is fetched
  + Only has impact when a report is built or new result is fetched
  + For now, limit to 3 scenarios to avoid slowness during the demo
* Select a Result
  + Select “Generation (TWh)”
  + Use Filter scenarios to limit the scenarios before selecting the result
  + Selecting a result will fetch data from one or more parameters from the gdx files of the scenarios
    - Generation uses the CONVqmnallyears parameter from Convqn.gdx
    - Other parameters have multiple parameters
  + Result-specific preprocess functions manipulate the data before we can pivot on it
  + Certain columns have additional preprocessing
    - Techs are mapped to a smaller set of categories, which are given specific colors
    - Region columns are joined according to hierarchy so that we can aggregate as desired.
  + Scenarios are combined into one table, with a “scenario” column added to the data
  + After selecting a result and retrieving the table of data, widgets drop down for the core pivot functionality
* View data structure of result
  + Click “CSV of full data source” under “Download/Export”. It should open automatically
  + Note that data grabbed from gdx has been preprocessed
    - Scenario column has been added
    - Region columns have been added for encompassing reigons
    - Techs have been mapped to categories
    - Generation has been scaled from MWh to TWh
  + The columns in this file are the columns that you can pivot on
  + Don’t do this right now, but note that we could point to this csv file as “Data Source”, and we’d be able to use the tool, though we would not have all features, e.g. presets and reports
* Play around with core widgets: Chart, X-Axis, Y-Axis, Series, Explode
  + Select chart type and select columns to use for x-axis, y-axis, series, explode, explode group. Questions?
  + Note that X-axis and y-axis are required to see any results
* View Legend
  + Click on “Legend” in widgets section
* View Config Summary
  + Keeps track of the configurations I’m making
* Press Alt to collapse all widget sections
* Select a Preset
  + There are different presets for each result
  + Select “Stacked Bars” if you had selected “Generation” as the result
  + A preset is simply a set of widget configurations that happen together
  + A config summary shown at the bottom of the widgets. See the dropdowns to see that the same configurations were made
  + So we could have made the same configurations one by one to get to the same view
* Modify config off of preset
  + After selecting a preset, we can add further filters and any other configuration
  + Under Explode->Group Exploded Charts By, select interconnect (“in”)
* Zoom/resize browser window
  + CTRL-plus or CTRL-minus to zoom in/out
* Apply Filters
  + Limit tech to Wind and PV with checkboxes
  + Click the Update Filters button for the data/plots to update
  + Note that we have another scenarios filter in the general Filters section. At this point, we’ve already retrieved the data for our result from our chosen scenarios in the “Filter Scenarios” section near the top, so the “Filter Scenarios” section no longer has impact on the current Result. Any further filtering of scenarios must happen in the general “Filters” section.
* Make plot Adjustments
  + Remove axis syncing, set axis bounds, change chart size, bar width, scale axes
* Save view as images:
  + Save figure png using the save icon on each figure
  + Take screenshot
  + windows snipping tool
  + Note that the legend must be saved too
* Download/Export View
  + Click on “All Files of View” under “Download/Export”
  + The files will be downloaded to your user folder in bokehpivot/out, in a directory called “view-” followed by the timestamp
  + View exported view.html, view.csv, report.py, url.txt files
  + View.html can be shared, and doesn’t require bokeh server. The charts will still be interactive in terms of pan, zoom, etc.
* Save/Reload View
  + The full view was saved as an encoded url query string in url.txt
    - If you want to see the configuration, use a url decoder, e.g. https://www.urldecoder.org/
  + Skype your url.txt to someone else
  + Append the url query string (starting with ?) to the end of your URL (after “bokehpivot”)
  + Note that bokeh server must be running bokehpivot for urls to be used
    - Caveat: Jonathan has allowed access to a running bokeh server on Orion/Scorpio from anywhere on the nrel network by using the machine name of Orion/Scorpio
* Add Comparison
  + Under “Comparisons”, select “Difference” as “Operation”, “scenario” as “Operate Across”, and select the base case as “Base”
  + Don’t do this now, but a different example can be seen in the configuration of the “Stacked Bars Gen Frac” preset under “Generation”
* Create Maps
  + Showing a map requires that x-axis is a region. This will be the region boundaries shown in the map
  + Only regions that have gis boundaries defined in bokehpivot\in are allowed: i,n,r,rnew,rto,st. For now select st for state
  + Adjust filters: 2050 for year, Wind and PV for tech
  + Remove interconnect explode group
  + Fool around with plot adjustments if you want to see a clearer chart
  + Change chart type to map.
  + Note that series is another explode dimension for maps
  + Enter “Reds” as “Map Palette 2”
* Download/Export View
  + Same drill as previous export
* Build custom reports
  + In “Build Report” section, select “custom” for “Report” and enter path to exported python file
  + Click “Build Report” and view files
  + Copy python report file, and create multi-section report by adding config from first exported view. Name sections appropriately
  + Enter path to new python file and click “Build Report” and “Build Separate Reports”
  + View results
* Manually modify custom reports
  + Note that, to this point, the “Base Case” selector has not been used. View the new python report file again. Note that, because comparisons were used across scenarios and a scenario was chosen as the base, the base case is hard coded into report file (see adv\_col\_base), which will cause an error when base case is not one of the scenarios. Open standard\_compare\_report.py in bokehpivot\reports\templates\. Note that the different keys are used. “Result” and “preset” allow us to simply use ReEDS presets. The “modify” key is used to allow scenario comparisons.
  + Add 'modify': 'diff' after ‘name’:’Some Name’ in our custom python report, and remove the three elements in the ‘config’ dict that start with ‘adv\_’
  + Now the Base Case from the Build Report section is used.
  + Click “Build Report” again
  + Now build reports with more/different scenarios and different base cases
  + Remember that all widget configuration beneath the “Build Report” section is not necessary, and we can use report files in different sessions.
* Suppress updates/chart render
  + When we have a large amount of data, it is often helpful to suppress updates until many changes are made. To do this, select “Disable” for “Auto Update” under the  “Auto/Manual Update” section.
  + Now make other changes to widgets, and notice that updates are not being made after each change.
  + Trigger an update by clicking “Manual Update”
  + We can also prevent charts from rendering if, for example, we simply want to gather data into csv files. To prevent chart render, select “No” under “Render Plots”
* Change meta
  + Expand the Meta section. There are three types of meta files, which are applied to specific columns in the data:
    - Join: This adds columns to the data. For example, bokehpivot/in/hierarchy.csv is used as a join for adding different regional aggregations.
    - Map: This aggregates data into different categories. For example, bokehpivot/in/tech\_map.csv maps raw techs into combined categories
    - Style: This allows styles, e.g. colors, to be associated with column values.
  + Copy bokehpivot/in/tech\_map.csv to another location, outside of bokehpivot/.
  + Edit the copied file by re-categorizing one or more of the techs
  + Enter the path to the new file in the Meta section, and notice how the
  + Note that meta section changes aren’t used by the Build Report section right now, though this is on the to-do list.
* Get your own repo
  + This is use
  + Wherever you’d like on Orion or Scorpio, clone repo: git clone <https://github.nrel.gov/ReEDS/bokehpivot.git>
  + Copy .bat file on desktop and edit copy to point to your own repo
  + Launch new .bat file
* Add new raw result (using your own repo)
  + Find a parameter in one of ReEDS output gdx files that you want to view
  + Open reeds.py and find a raw parameter (e.g. search for cap\_wind\_nrr). Note that the raw parameters simply have ‘file’, ‘param’, and ‘columns’ keys.
  + Make a copy of one of the raw parameters and paste below.
  + Change the ‘file’, ‘param’, and ‘columns’ to match your parameter
* Add preprocess functions and presets
  + Copy examples from other results that have these settings
  + Note that a result may be an aggregations of other results for speed, e.g. see “Gen by m” vs “Gen by m full” results

**Other Features**

* Existing features not shown
  + Ranges within or between series
  + Other y-axis aggregations
    - Average
    - Weighted Average
    - Weighted Average Ratio
  + Other comparisons
    - Generation fraction
    - year-to-year consecutive changes
  + Bar width manipulations
    - Stacked gen by timeslice with hour-weighted bar widths
    - Duration curves
  + Access to a running bokeh server on Orion/Scorpio from anywhere on the nrel network by using the machine name of Orion/Scorpio
* Features to come
  + Series grouping
  + Separation of core functionality and integration with other systems
  + ReEDS run automatic standard report
    - <https://github.nrel.gov/ReEDS/ReEDS/pull/711>
    - Jonathan and I are may make automatic standard compare reports for batch runs too
  + Others?
* See README for more info at <https://github.nrel.gov/ReEDS/bokehpivot>
* Any questions, please contact me via email or skype
  + I want to know all issues/confusion people are having with the tool!

**Notes**

* We got to custom reports
* Add tutorial as md file to repo?
* Base case: remove options
* Internet explorer issues
* Limit selections/features