Include hyperlink

<https://evan-canfield.shinyapps.io/SE_US_Electricity_Generation/>

# Section 1

From the early part of the last decade, total electricity generation has remained relatively steady in the United States. What has changed is the manner in which it is generated. Renewable energy sources for electricity generation such as solar, wind, and hydroelectric through the use of photovoltaics and turbines technologies, have increased in use accounting for over 16% on a national level. Alternatively, generation from nuclear sources and fossil fuels such as coal, natural gas, and oil continue to dominate the power landscape. In the United States, coal was the major source power, accounting for over half the total of electricity generation at the start of the century, according to the U.S. Energy Information Administration (EIA). Since then, coal use has dropped dramatically to just under 30% of the total generation. Due to advances of fracking, which in turn have unlocked resources from shale deposits all around the country, natural gas has taken its place as the leader in the U.S. at 32%.

In the Southeast region of the United States, the changes in electricity generation between renewable and fossil fuels/nuclear sources have not kept pace with the rest of the country. By the end of 2017, renewable energy sources account for just over 6% of the region’s electricity, which is just a slight increase from 2001. While the region as a whole continues to rely heavily on fossil and nuclear sources, each state within the Southeast tells a different story.

In our Shiny application using 2001 to 2017 data provided by the EIA, we attempt to illustrate how electricity generation has changed over time for each state in the Southeast United States. Specifically, we illustrate changes by renewable and non-renewable energy sources as well as the various resource types. Furthermore, we supplement these visuals by showing carbon over the same time period.

# Section 2

## 2.1

The primary data used for this Shiny application is publicly available data provided by the US Energy Information Administration (EIA) in the form of year EIA-923 reports, or equivalent (see Reference 7.1). Before 2006 the relevant data was reported in reports EIA-906/920 or EIA-906, which are consistent with EIA-923. The reports provide detailed electric power data on electricity generation, fuel consumption, fossil fuel stocks, and receipts at the power plant and prime mover level. This app was only concerned with electricity generation on a yearly basis and was restricted to the Schedule 1 survey data, or equivalent. The reports are available in spreadsheet form from 2001 through 2018, although the 2018 data is not used. See Appendix, Section 1 for more detail.

The most current issue of report EIA-860M was also used (December 2018 as of app development, see Reference 7.2). This report provided generator-specific information for all generators and associated environmental equipment at electric power plants with 1 megawatt or greater of combined nameplate capacity. This report provided context for the EIA-923 information, including energy type, location, and plant name, and other essential information. Location data is comprised of longitude and latitude coordinates, as well as city and state information.

The raw data from EIA-923 reports, or equivalent, and EIA-860 were compiled, merged, and joined initially through a SAS Enterprise Guide process. The output of the SAS process was then further refined in R, until all of the necessary information was contained within a single, tidy dataframe. Each observation within the dataframe was an energy type (see descriptions in Section 2.2) at an operating plant for a given year, with all relevant yearly data such as net electricity generation or net greenhouse gas emissions. If multiple energy types were present at the same location then that location would have as many operations as energy types, with each energy type reporting output data. For example, Plant.ID\_Energy (see [Section 2.2](#section2.2)) values of 10\_COL and 10\_NG indicate coal generation and natural gas generation at Plant ID 10. Both of these observations have the same location data, but different fuel type, generation, and emission data.

Further processing of the dataframe was then performed within the Shiny app, although outside the ui and server. This processing created eight additional dataframes, each specific to a Shiny application component. Creating these specific dataframes was done to reduce processing within the server during operation to the absolute minimum.

See Appendix, Section XYZ, for detailed discussion of the data processing performed in R.

## 2.2

The complete dataset, before further app specific processing, contained the following twelve variables:

#Insert Table

## Section 3

In our design, we utilized a map of the extended Southeast region of the United States as the main visualization which serves as the initial focal point. This map was generated with Leaflet. The states included are Alabama, Arkansas, Florida, Georgia, Kentucky, Louisiana, Mississippi, North and South Carolina, Tennessee, and Virginia. A point was placed on the map for each unique Power.ID\_Energy code (see Section 2.2) for the year selected. Each point represents the electricity generation by a specific energy source type at a power plant. The points are located on the map by their latitude and longitude coordinates. The size of the point is in relation to the effective rate of the plant (see Section 2.2), and the color illustrates the energy source type.

Upon the initial load of the visualization, the map shows data for the most recent year, 2017. The selected year filters all of the data within the application, so that all app components, map and charts, are showing data from the selected year. Located at the top of the page is a slider for selecting the year, and on the right side of the slider action button. When activated the year changes, as indicated by the time line at the top of the page, to the beginning of the data set. This action at first shows data for 2001 and automatically progresses to each following year. This in turn, simultaneously filters the app by the selected year, and animates the regional map and charts to show the yearly changes for each plot.

To supplement the main map visualization five charts were generated using Highcharts and were broken out across four-tab panels. Below is a description of the charts:

* Tab Panel: Total
  + Stacked bar chart: The bar height indicates the total electricity generation for the given year, with the color sections indicating the electricity generated for a specific energy type.
* Tab Panel: By Energy Source
  + Line chart: The line indicates the total electricity generation for the specific energy type for a given year.
* Tab Panel: Renewables
  + Line chart: The line indicates the total electricity generation for either renewable or non-renewable energy for a given year
* Tab Panel: Renewables
  + Line chart: The line indicates total greenhouse emissions per generated electricity (MW) for a given year.
  + Line chart: The line indicates total greenhouse emissions for fossil fuel sources

An additional tab panel, About, contains information on how to use the app.

Like the main visualization, the same color codes used in the map are utilized to differentiate the various energy sources for each of supplemental charts. To focus on a particular data element, a user can interact by selecting/deselecting a type on the chart legend, such as Coal or Renewables.

To drill down and focus on an individual state within the Southeast region, we placed a pull-down menu located at the top right side and above the dashboard. By utilizing the pull-down menu, a user can choose a state and filter the data and visualization. Once selected, the map zooms in to that state while the supplemental visuals show filtered values. Like the regional map, the action button can be used to animate and display the year to year changes of electricity generation for an individual state.

# Section 4

There are two major components to this app, the map visualization utilizing Leaflet, and five charts spread over four separate tab panels. Each component required special consideration for algorithmic design.

##Leaflet Map

The Leaflet map, visualizing all of the power generating location in the Southeast United States, is the only component that is permanently on the screen while the app is in operation. This component is also processing the largest amount of input data in the app, when compared to the charts. Therefore, proper design is essential for the map to run smoothly. The primary method for this is to break up the Leaflet visual into different components. The first component is called outside of the server and the ui and performs the initial render the Leaflet map.

This initial component does not call the points to represent each power plant, though. The points are called by a separate component, inside the server. The function leafletproxy() is used as opposed to leaflet(), nested in an observe() call. This prevents the Leaflet map from being fully rendered after every change. Only the dynamic aspects of the map are affected, and only when an action is observed.

#Insert Code

Additionally, options were set in order to limit how often and when the Leaflet map would refresh. These options further reduced the computational load of the map, without noticeably degrading the operation of the map for the user.

## ##Charts

The primary method for maximizing algorithmic capability for the charts is to maximize data processing to outside of the server and ui before passing the required dataframe into the server. As discussed in Section 2, the raw data was processed into a single tidy dataset before import into the shiny app. From this master dataset, eight unique datasets were developed. Each of these datasets was processed for a specific use in one or two charts. These specific datasets were imported into the server as reactive values. The only processing taking place in the server is filtering via year and/or state. If the full processing of the dataset occurred in the server, this would encompass many additional actions, including a join. As the app is scaled up, those actions, particularly the joins, could be computationally costly.

#Insert Code#

A benefit of having the charts broken out onto separate tabs is that only the charts visible to the user are active. See the screenshot below of the reactlog record of the operation of the app, with Total tab selected. There are two reactive expression expressions active, one grabbing the dataframe to input into the map (observe({ EIA\_923\_SE…}), and the other grabbing the dataframe input into the visible chart on the Total tab (output$net\_gen\_stacked\_bar). The reactive expressions used for input into the charts not visible are not active.

As an additional note, the stacked bar chart was the slowest output within the app, based on the profvis analysis. A select screenshot of the profvis test is found in Appendix, Section 3.

# Section 5

An evaluation is an opportunity for a developer to collect feedback from its end users. This in turn allows the developer to understand the visual’s usefulness, if it meets expectations, and most of all, if it provides invaluable insight.

To evaluate this app we would need to assess several criteria, such as functionality, effectiveness, efficiency, usability, and usefulness. This evaluation can be mainly summarized into major components. First is the interaction and use of the app. This generally covers functionality, usability, and efficiency. Interaction and use of the app could be evaluated through surveying a variety of users, both new and experienced users to Shiny and the subject matter. ~~experienced with Shiny and the subject matter, and not.~~ After the users have engaged with the app, we would ask how comfortable they felt interacting with the site, how intuitive the controls of the site were, and if they would provide ~~and we would evaluate~~ any suggestions on improvements.

More importantly the evaluation would need to assess if the app provided additional value, and if that value is beyond with what is currently available ~~is that value beyond what’s currently available~~ . This would cover effectiveness and usefulness. ~~As currently this data only exists is raw spreadsheets, and that there is not a widely available interactive visualization of the data, it is clear that the app does provide some form of additional value.~~ Since this data exists only in raw spreadsheets and given that there is not a widely available interactive visualization of the data, it is clear that the app does provide some form of additional value

But the question is can the user correctly understand what the app is conveying? To evaluate this, we propose developing a set of scenarios and questions about the subject matter, not too narrow but not too open ended either. We would ask users these questions and asses the responses in their clarity and their accuracy. If all the user can do is recite raw numbers then the app is valid but perhaps not entirely useful. But if the user can create a story with context, trends, relationships, along with appropriate numbers, then the app would be both valid and useful. For example, here are several questions we could ask:

1. Since the mid-2000’s there has been a boom in natural gas production in the Unites States. How may that have affected the use of natural gas to generate electricity since that time?
2. To what extent have renewable energy sources developed in the Southeast United States? Where has the most development been? Least?
3. In the Southeast United States, what was the dominate form of energy in 2001? In 2017? For individual states, how does this compare to the region?

# Section 6

Moving forward we would hope to improve and expand the functionality of the app.

* Filter by energy type, affect both charts and map
* Entire US
* Average Comparison of US, or Region
  + Line across a stacked bar
  + Area chart background of stacked bar
* Select multiple states
* Interplay between Map and chart
* Total emissions projection based on climate change requirements
* Climate policy highlights when hovering on years