Report Programming Project

"Photovoltaic Growth in the U.S."

June 2015

Thomas van der Wardt
10900578

Contents

Purpose and goal

Renewable energy resources, their usage, as well as national and international differences between usage, is a topic that has been visualized in many ways the last decades. Large organizations such governmental research institutes and consultancy firms have spent a lot of time in this field, in attempts to identify trends and growth of renewable energy.

Unfortunately, it seems that growth is mostly visualized in terms of amount of jobs in the renewable energy sector or total revenue of solar panel companies. I want to change this, and visualize the growth of renewable energy per US state based on a number of different parameters:

1. For each US state the total capacity each year;
2. For each US state the annual generated electricity per capita from solar panels for each year;
3. For each US state the average cost per kW solar panel capacity for each year;

By visualizing the growth for each of these parameters, trends can be spotted and market opportunities may arise; which states are lagging behind in terms of capacity or annually generated electricity per capita, and which of these states seem to have access to the cheapest solar panels, and in which states are solar panels most expensive? The visualization will be able to answer such questions, aiding in finding trends in solar panel market of the US.

The visualization will consist of a map of the US for each of the parameters, with an x-axis showing the current year that is being shown and each state filled with a certain color representing the value of the parameter at that year of that state. To be able to make easier comparison between states and to see trends more clearly, for each of the parameters a line graph will be added separately from the US states map visualization. In the line graph, each state is represented by a single line, with the x-axis ticks per year, and the y-axis as the range of values of the selected parameter. As the first and second parameter have very large differences between states, the y-axis will be made flexible; changing the range depending on the maximum and minimum value of the parameter in the selected state.

Data sources and reformatting

Solar panel data is publicly available on 'The Open PV Project' [1], where over 420,000 solar panel installations are stored in a database which can be accessed through their website, and downloaded to CSV files. For this visualization, all data has been downloaded for all states, with for each entry data on: the State the solar panel is in, the zip code of the solar panel installation site, the size (capacity, in kW), the cost (in $) of the solar panel, and the date it was installed. The first entries for installed solar panels date from the 1980's, where only a handful of solar panels were installed in the US, whilst in the last few years over 100,000 new entries were added to the database.

To be able to calculate total capacity per state, total annually generated electricity per capita per state, and the average cost per kW capacity per state, data on population per state is needed. The United States Census Bureau [2] makes annual estimation of state populations, and seems to be the most reliable source for population data for all states.

In order to not make any unnecessary calculations after loading the data in the HTML DOM, all data has been reformatted to CSV files for each state, with one column providing the date, and one column with the value (e.g.

1 The open PV project website: https://openpv.nrel.gov/index

2 The United States Census Bureau (2015), *Annual population estimates*, accessed June 2015, http://www.census.gov/

total capacity up to the year indicated in the date column). The data was reformatted through python, summing up all entries from the initially downloaded file. For the second parameter, the annually generated electricity per state per year, multiple calculations and reformatting steps were required. First, the zip code provided in the initial file was taken to find the average solar radiation of that location (data from National Renewable Energy Laboratory [3]) and use the installed capacity at that location to calculate an estimate for the annual generated electricity. This was done for each solar panel data entry, afterwards storing it in an array dedicated to the state that solar panel was in. Unfortunately, not all data from the Open PV Project is formatted in exactly the same way; some entries suggested capacities of over 1 TW (= 1 million kW), such entries were ignored from further formatting and therefore were not added to the parameters for each state such entries occurred. The values of capacities, annual generated electricity, and cost per kW shown in the visualizations therefore are merely estimation.

Development challenges

During development various challenges were encountered, in this paragraph the major challenges will be discussed as well as how they were overcome.

- *Extreme data points*: as all data had been downloaded from the Open PV database, a fairly high number of entries had 'extreme' values in the capacity column; over 1,000,000 kW for a single entry. After identifying these values, they were checked whether they would be valid values compared to the cost of the data point. If the cost of such an entry was below 100 $/kW it was highly unlikely to be a valid entry and was therefore rejected from further incorporation. Even though the data has been filtered this way, it may very well be possible that either valid entries were not incorporated or non-valid entries were actually incorporated.

- *Zero-values in data*: as in the year 2000 and even after that some states did not have any solar panels whatsoever, the costs per kW were unknown, but were indicated as '0' in the loaded data. To fix this, a filter function would need to loop through all states and their values and only use the non-zero values for drawing their respective paths in the line graph.

- *Identifying paths as states*: every state was drawn according to its path defined in the us.json file, but this json file did not keep track of the name of the state when drawing it, but only used a specific ID. To be able to identify single states, the ID had to be coupled to a name, but no lookup file was available to do this. To solve this, a separate csv file was created to link an ID to a state abbreviation and name, therefore making it possible to add the state abbreviation as the path ID.

- *Loading all data at once:* As all data was stored in separate file; a separate CSV file for each state, for each parameter, it was necessary to load all files (over 200 files) immediately when opening the website. By directly loading them, it would save time in a later stadium and make it more user-friendly in terms of waiting time. To be able to load all data at once, every file name was given a unique name, based on 1) the parameter it contained data about, and 2) the state abbreviation. By naming each file in the same manner, e.g. 'cost_per_size_AK', it became possible to load all data in a queue using a for loop, looping over all state abbreviations.

Visualization, interactivity, and animation

As the data files consist of data for each state, as well as the population for each separate state, the choice to make a visualization using a geographical map of the US with borders of states, seemed the way to go. As using solely a map and color coding by using bins of values may not successfully deliver the full picture, and is inefficient for users to

3 National Renewable Energy Laboratory(2015), *Solar radiation averages over 30 years, per zipcode*, accessed June 2015, http://www.nrel.gov/gis/solar.html

compare states, each parameter also has a line graph to visualize the growth or change in absolute numbers. The use of color bins to visualize values per state is not effective at presenting the user the actual value of a certain state, and therefore a separate box was added to the geographical map to present the value of the state the mouse is on. As the values of each state change over the years, an x-axis was added with tick marks for each year, as well as a play, pause, and reset button to be able to control the shown data. In the line graphs of the total capacity and the annually generated electricity, the y-axis is kept responsive to the selected state, to be able to identify similar states and to see whether this is a very solar panel intensive or extensive state. If the y-axis was kept fixed, many values would lay very close to each other, making it essentially impossible to identify single states. The cost per kW line graph has a fixed y-axis, as the values per state are rather close to each other, and this immediately gives the user an overview of all other state prices and how the selected state compares to the other states.

The US map view was inspired on the following example:

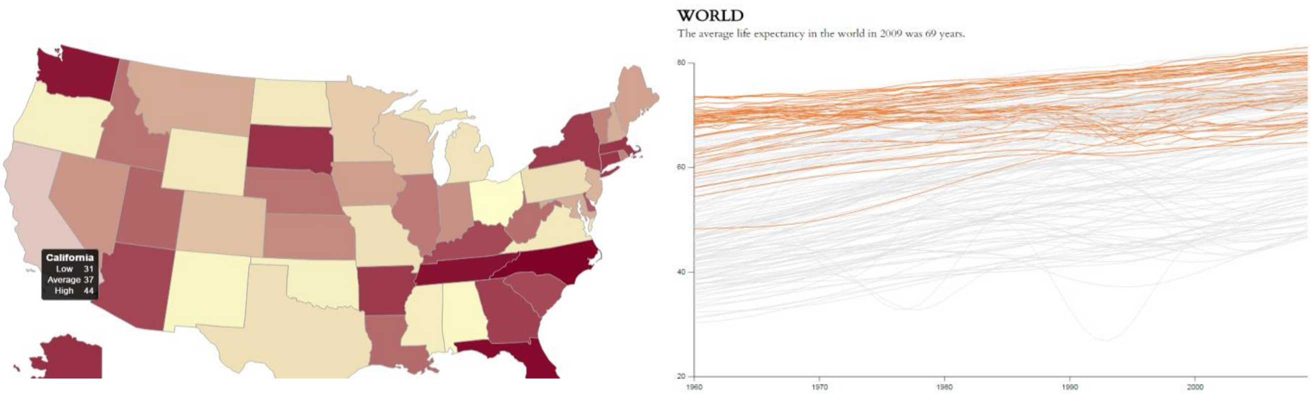

Figure 1 – Left: US map view map, showing the data for each state by hovering over it.. Right: World life expectancy, showing the change in average life expectancy for each country in the world from 1960 – 2010. (Sources: http://bl.ocks.org/NPashaP/a74faf20b492ad377312 and http://projects.flowingdata.com/life-expectancy/)

These visualization were taken as examples because of their simplicity in showing the data, and having no redundancies in their visualizations. Using these examples, I sketched how my visualization should look. See below for these sketches.

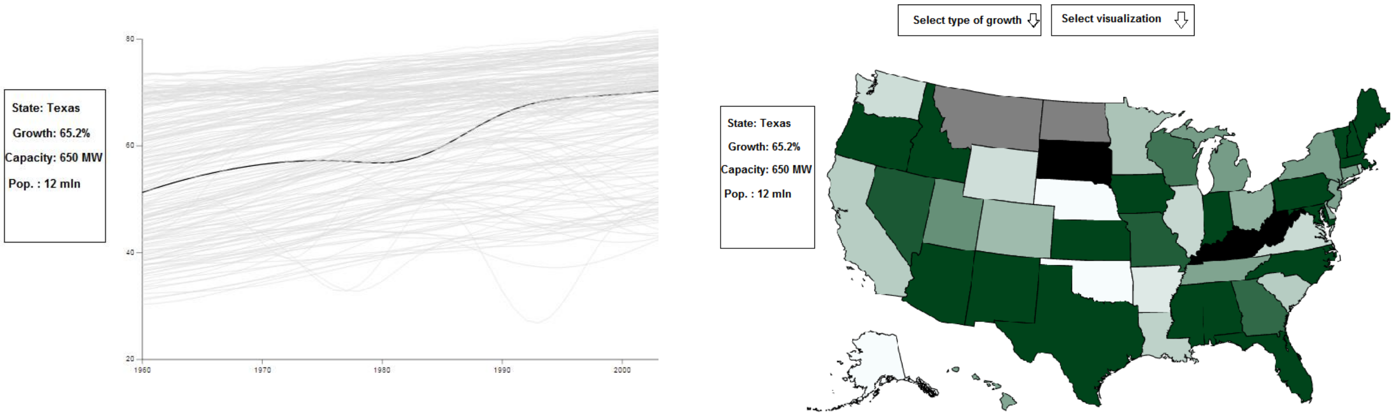

Figure 2 – Left: Line graph view, showing the data for each state by clicking on it. Right: US map view, showing a certain year of values, and coloring each state based on its value, and showing a state's value by hovering over it.

The sketches served as providing a visual aid towards working to my minimal viable product, and after achieving approximately what I sketched, I started to extend the possibilities for each of the visualizations; adding functionalities.

Implementation

The implementation, the step from having CSV files with dates and values for each state to working towards a structured, user-friendly, responsive website to show each of the visualizations with a single click on the mouse. I will describe the implementation of the US map and the line graph separately.

US map

The US map visualization's main component is the drawing of each path, and filling each path based on its value. Path values were taken from a simple US states D3 map, by Mike Bostock, and used stored in a separate JSON file which could then be loaded in JavaScript to draw each state. Each path for a certain state could be tracked by a certain ID, which corresponded to a certain state abbreviation. A separate file is used to track the state abbreviation based on the path ID number. To load the actual data (the solar panel data), a queue was used. By using a queue and calling a function after all data had been successfully loaded, it was ensured that the script would only run if the data had been loaded. As for the visual aspect concerning the values for each state, this was done using a color sequential color scheme provided by ColorBrewer [4]. Bins for each color were decided on the range of values among states; the highest value fitting in the largest bin, and the smallest value in the smallest bin.

Each of the parameters (capacity growth, annually generated electricity, and costs per kW), have been implemented as separate functions, as each of them have slight differences in how the data is named, title names, and legend. This decision was made to make it easier to make subtle changes in a single map, instead of having to work with conditional statements. This way it is also easily achievable to draw a map based on other data, as most functionalities would remain the same; only the data implementation and processing would need to change.

Line graph

The line graph visualization's main component is the drawing of the line paths, which was achieved by binding the data to a path element, and calling a separate line function to get a x- and y-value for each data entry for each state. Afterwards, an interactive element was added; by clicking on a certain path, the graph is re-drawn and now shows the clicked path as the selected state. In the capacity line graph and the annually generated electricity line graph, this

4 For more color schemes and how information on how to use color schemes, see http://colorbrewer2.org/

means the y-axis changes position based on the largest value in the data of the selected state. In the cost per kW line graph this only means the text above the graph changes to the selected state and the selected path changes from fill-color (from blue to red). To ensure all data has been loaded before drawing the line graph, a timeout is added if the function is being called before all data has been loaded.

Other

As the website does not only consist of the graphs drawn using D3, in this paragraph other used techniques and libraries will be discussed.

Bootstrap, jQuery & JavaScript for Bootstrap

To make a responsive and user-friendly website, a simple bootstrap template was used, including a header bar and an empty body. All bootstrap related styling is stored in a separate CSS file, other styling used for svg placement, text placement, and image placement is stored in a different CSS file.

In order to have the a fully functional bootstrap styling, both jQuery and a bootstrap JavaScript file are included on the website. jQuery is version 1.11.1.

jQuery

A separate piece of jQuery code has been used to change the shown value in the dropdown menu to the clicked item in the dropdown menu. This piece of code has been taken from the example written by Jai, 2014. [5]

D3

For all D3 functionalities to work properly, the D3 v3 JavaScript file was added to the website.

TopoJSON

The TopoJSON v1 JavaScript file was added to the website to be able to use the JSON file with paths for each of the US states successfully.

Queue

Queue v1 JavaScript file was added to the website to be able to load data before executing any functions, ensuring data has been loaded before any other function could be called. This is used in both the US map D3 visualization as well as the line graph visualization.

[5] Jai (2014), *How to display selected item in bootstrap button dropdown title,* answer on question asked on stackoverflow.com, direct link: http://stackoverflow.com/questions/13437446/how-to-display-selected-item-in-bootstrap-button-dropdown-title

Final product

Below is a screenshot of the final product: website using bootstrap styling, with an introduction text and two example visualizations on the right. By clicking the links in the introduction text, an SVG element will be opened below the introduction, and above the references, with the visualization generated through D3.

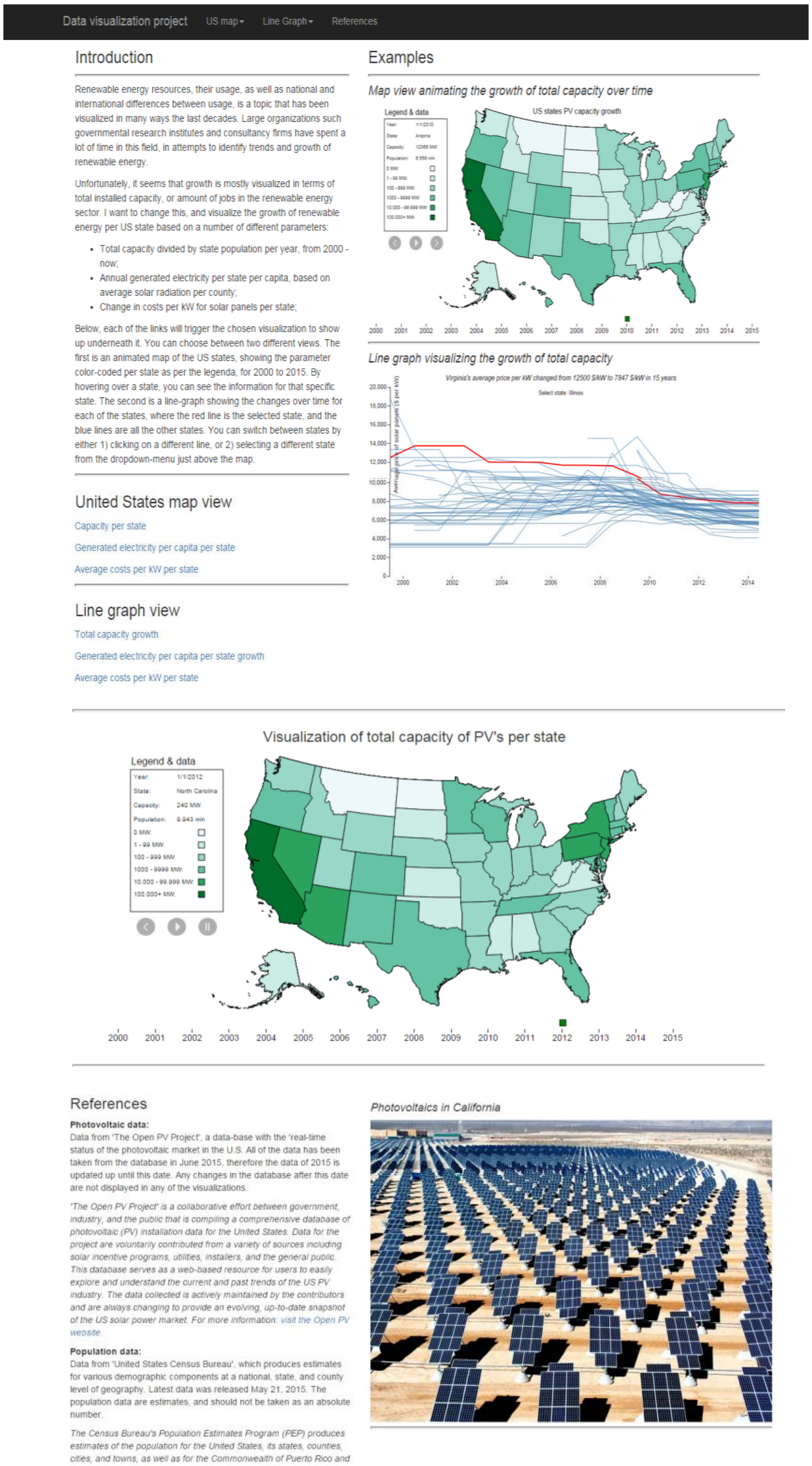

Figure 3 – Screenshot of the webpage with the visualization. First part includes the introduction text, 2 example images of the visualization, and 6 links which trigger their respective script to draw the visualization it names. Next is the visualization, in this case the US map of capacity of solar panels, and after that references, stating which data sources were used to produce the visualization. On the right side a picture of a photovoltaic 'farm' in California is shown.

`

Design process and future usability

The initial design and the final product differ from each other, but only slightly. The major components, the US states map with colors for each state and the line graph illustrating the change/growth of a certain parameter, are successfully implemented in the final version. Additions to the presumed design include:

- X-axis for the US map, showing which year is being shown;
- Play, pause, and reset button to automatically change the colors of each state over time depending on each of their values;
- Interactive line graph; the possibility to click on a state and subsequently redrawing according to this clicked states values (meaning the y-axis will change for capacity parameter and for the annually generated electricity parameter);
- A completely functioning website with links functioning as triggers to execute certain functions;
- Dynamically remove and add HTML elements through D3 (the dropdown menu to select a state);

Overall, the design that was considered to be the minimum viable product in the original design document has been achieved, and several additions have been made to make it more informative and user-friendly.

Possible extensions and improvements

This visualization is rather fine-tuned for the data that was generated from the Open PV project, and therefore has limited use outside this type of data. Nevertheless, the current structure could be rewritten to a format in which it would be possible for the user to upload his/her data, and to show that in either the US-map or the line graph. By doing this, it would be more re-usable and no longer limits the user to the data that was hard-coded by me.

Possible extensions for the visualization include different visualization, for example using bar charts or pie charts to show the distribution of solar panels in the US. Another possible extension is to make a more detailed US map, by using a county-view map instead of a states-view map. The data from the Open PV project makes this not too hard to do, as each entry has a certain zip code which could be linked to a county.