

Chris Chung cc2299

Max Ma mm2559

## **Final Deliverable Document for Exploring Europe Energy Trade**

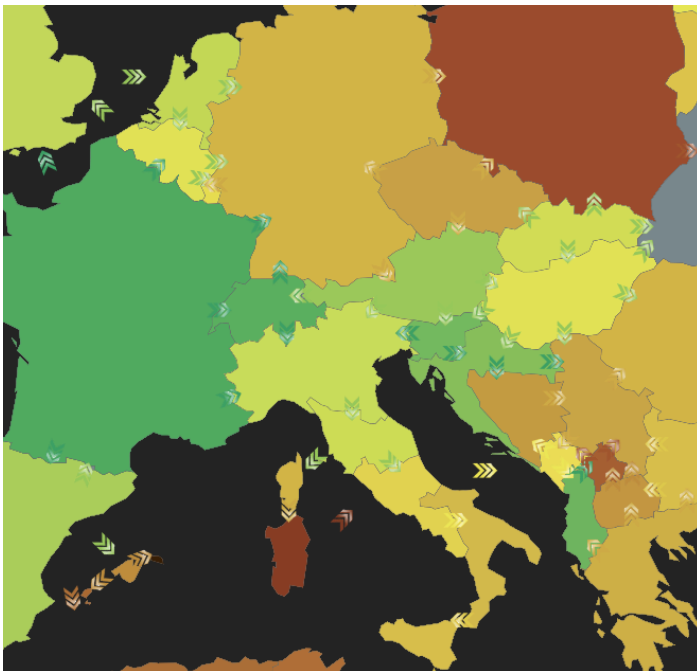
[Link to Render](#)

Our goal for this project is to explore energy trade and energy production specifically in Europe. We were motivated by the fact that we both heard a lot about energy and energy trade in the news recently, but we really didn't know that much about the topic. We wanted to explore the subject of how energy trade works and how events like the Russia-Ukraine war have affected this trade, and we wanted to educate people in our situation with an interactive visualization. We wanted to create an article-style project where users will be introduced to the topic and interact with through scrolling, hovering, and clicking. Users will go from one section/interactive visualization to the next through scrolling, similar to that of interactive scroll-based NY Times articles.

This project is for individuals with little experience or knowledge in European energy who want to learn more about it, including topics like what types of energy are being generated, which countries are trading with whom, and what effects did the Russia-Ukraine war have on these trades. This project can be used in a similar fashion to that of the interactive New York Times articles, where people who want to seek this knowledge after hearing about it may come across it on a google search or link our project as another tool/information source.

One of our first inspirations for this subject matter was the website [electricitymaps.com](https://electricitymaps.com) which was a website that allowed users to zoom onto different countries and hover over them

to see their energy production breakdown. The site was incredibly clean and was one of the very few online energy resources that was easy to use. We wanted to take the informational yet clean aesthetics and turn it into a more storytelling project, as people who are completely new to the field may have still been overwhelmed. Electricitymaps visualizes trade only happening with neighboring countries could be misleading and the complexity of some of the supplemental graphs may be overwhelming.



Source: [electricitymaps.com](http://electricitymaps.com)

The biggest inspiration for us having the scroll-trigger storytelling aspect was Cuthbert Chow's interactive article [What is the best \(undergraduate\) college major in the United States?](#) This project particularly inspired us with how it was able to surprise the user with a lot of data but then break down the data into segments where users could easily understand what was going on. The project was not particularly long or complex, but was short and efficiently conveyed the major points to the user. The creator also wrote an article on exactly how they

created the interactive scrolling visualization in D3, so we used similar scrolling mechanisms with scroll sections and cleaning functions.

We used multiple datasets from multiple sources for our project. The first being EU trade in electricity, which was a dataset that showed how much electricity countries within the EU traded with each other from 2018 to 2023. This was taken from [Eurostat](#). The initial scrubbing was done with the website's internal tools that allow users to select variables and structure, so not much was done in terms of the data manipulation. The second resource we used was <https://www.eia.gov/>, which gave us a massive dataset of the world energy balances for almost every country. This was a dataset that consisted of every energy type a country produced/consumed from 1971 to 2023. In order to use this dataset, we had to first filter out just the production of each energy source, and then filter by which countries we wanted to look at specifically. For just isolating Germany and Russia, this filtering was done in Python. In Javascript, since all the variables were years and the rows were different energy types we restructured it so the energy types were the variables and the years were the rows. This allowed us to finally create the line graphs we use in the project.

We went through several different iterations and designs for this project. We knew that we wanted to start off with a big complicated overview of all the energy/natural gas trade that was happening in the EU as a hook to learn more about the subject. We initially did this by creating these curved connection lines on our projections connecting every major country that traded natural gas and oil with the EU to get this big global view of how much the EU imported. We ran into several issues there, mainly adding hover features became cumbersome and unless the opacity was set very low, it was hard to see some of the smaller trade lines. We eventually

changed this to a version where all electricity trade just within the EU would be visualized. This was also visually complicated, but on a scale where users would know what was going on. Making the stroke width and color of the connection lines the same sacrificed some visual channels, but it became very visible which countries were trading with whom.

Our initial second stage of the project was having the map zoom into Europe and being able to see which countries traded electricity with each other by hovering over a country and having straight connection lines appear that varied in stroke width. This was effective at showing users which countries traded with which, but eventually got more and more buggy the more we tried to implement. We eventually decided that this design direction was also flawed as it didn't really match the story-telling aspect that we wanted. We decided on just focusing on Germany for the second part, so seeing an animated line graph showing energy consumption by energy type over time. This allowed us to better lead into the impact of the Russia-Ukraine war because the line graphs dip a certain point when the war starts. This became more of what drove our design process going forward, and we wanted to continue this process of guiding the user through specific countries and then at the end allow them the freedom to explore different countries.

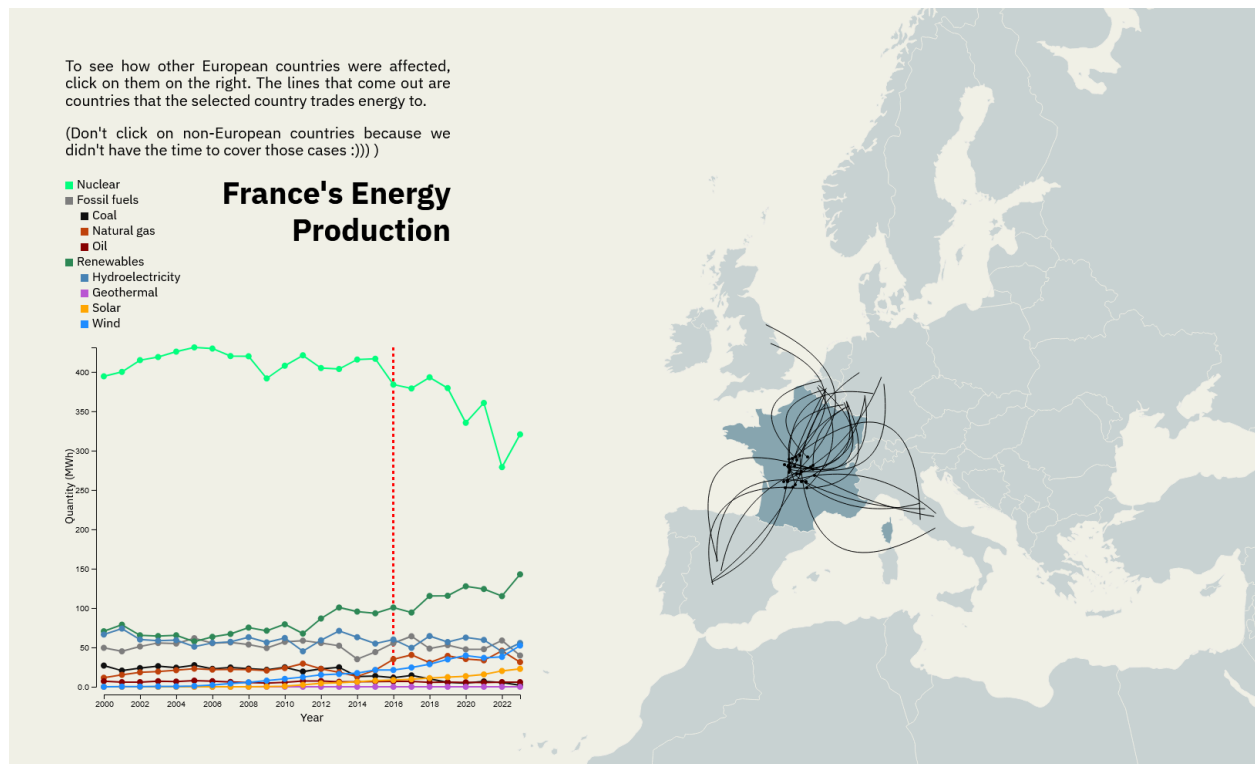
Our final design is a five part scroll-based interactive web visualization. The first is a map of Europe with text talking about EU energy trade. The second phase brings in the electricity trade connections between all EU countries along with text explaining how complicated this whole system really is. The Third section zooms into Germany as a case study and introduces Germany's energy breakdown over the past few years, highlighting a dip in oil and gas in 2016 when the war starts. The fourth section then zooms out to show Russia and its energy

breakdown along with indicators and text as to how its production was affected by the war. The final portion then zooms back out to all of Europe and allows the user to explore the energy breakdown of whichever country they hover over by bringing up a line graph with their energy breakdown.

We definitely faced some pretty big tradeoffs when coming to our final design. We initially really wanted to show what kinds of energy (electricity source) were being traded with different countries specifically, but we really couldn't find the data anywhere we looked online. We would have also loved to give the user more ability to explore energy trade by showing prices and how those were affected, but that was too big of a scope for the amount of time we had. The initial feedback we received from the demo day helped us realize our initial prototype was unclear and lacked the story-telling aspect. It also made us investigate our initial scroll trigger functions and throw them out for the sections and clean functions.

Our final project was implemented with just D3 and Javascript. We created one d3 map with a geoConicConformal projection. To implement scroll-events, we created a function that would listen to every time the user scroll and announce which section the user was on. We had 5 sections, and each section ran a draw function and a cleaning function. For example, if the user scrolled down to the section focusing on Germany, the draw function for that section would run and zoom the map into Germany, lower the opacity of the surrounding countries, and animate/make the line graph visible. The cleaning function would hide all other unnecessary elements by setting their opacity to 0. This would work if the user scrolled back up to the previous section, as the draw function would reset the zoom and opacity of the map, and the clean function would then hide all line graphs.

Some screenshots:



Hi, Chris writing here. I apologize for Max's writing - he's not much of a humanities guy.

The most important part of the project was smooth animation, since that's the main 'grab'.

Some things that had to be problem-solved and my perhaps misguided solutions:

- Making curved paths when only two coordinates are provided
  - (Used Quadratic Bezier curve calculations to create a fake, offset midpoint)
- Repeating the trade link animation
  - (Wrapped the transitions into a function)
- STOPPING the repeating trade link animation
  - (kept track of queued transitions and Timeouts, used a global bool to determine whether we should continue repeating)
- Animating the lines for the line graph
  - Used stroke-dashoffset and stroke-dasharray to essentially displace the stroke one length away from its end. Then, bring it back to 0. Credit to [this article](#) for providing guidance
- Animating the circles for the line graph
  - This was a bit more difficult to animate than the line itself. When creating the circles, I added an attribute that kept track of its distance from the last circle, and also the summed distances so far. Then, I used a reversed array of those circles (so that it would be descending in sums) to calculate what fraction of the delay timing it should use.

I'm probably forgetting some of the other, smaller animations, but those were the big ones that I can recall as I'm writing this. Previously, we had a very janky and unwieldy scrolling system that

depended on widths and heights and windows - I just proposed that we scrap that system altogether and start anew with a more robust one. Oh, and transitioning between Country Names vs. countrynames was sometimes confusing, and it probably would've been better to leave it to a single function to decide, but I just manually coded a `removeWhiteSpace` function and used `removeWhiteSpace(str.toLowerCase())` every time. Now that I think about it, why didn't I just set the strings to lowercase in the `removeWhiteSpace` function? Hm.

To be honest, I wasn't really feeling it this semester, but the past few days working on this project have reminded me why I decided to enroll in INFO4310 and also why I enjoyed INFO3300 so much. There's a certain satisfaction to thinking about ways to piece together completely unrelated elements and data points. Professor, I really hope you're able to find some place that will treat your talents as an instructor with more respect than Cornell!

Some things I would've implemented had we started a little sooner, maybe:

- Comparative bar chart that allows you to add countries as colors with the x as energy types and the y being values
- Some visual animations for the Russia-Ukraine war section. Red lines going from Russia into Ukraine would've been pretty simple and effective

Contributions on the next page:



Max Ma mm2559:

- Initial research on topic area. Research in finding and exploring various datasets to use. Found the Electricity trade dataset currently used and the natural gas trade dataset used in prototype.
- Created the prototype EU-Global gas trade connections and the EU electricity trade connections in the prototype. Also prototyped initial scroll listener functions.
- Implemented initial hovering feature of prototype of highlighting a country by hovering over it
- Created line graphs for Germany and Russia along with circles. Created initial tooltip and legend. Implemented Cleaning and drawing animations for line graph.

Chris Chung cc2299:

- Sourced electricity data
- Created trade links
- Created animations
- Added scrolling functionality
- Created the interactive last section