Overview and Motivation: *(Provide an overview of the project goals and the motivation for it. Consider that this will be read by people who did not see your project proposal.)*

In this project, we created a node-link diagram showing the Pokémon found in Red and Blue, sorted by type. We also displayed the Pokémon's stats, both through node size and via a tooltip. The idea was motivated by Alex's heavy appreciation for the Pokémon series. Visually representing how the Pokémon are sorted into types seemed like a fun idea, as did the older idea of representing where in the world the Pokémon are found.

Related Work: *(Anything that inspired you, such as a paper, a web site, visualizations we discussed in class, etc.)*

A visualization that was shown in class early in the term, showing how much various ingredients are used together in recipes, provided some level of inspiration for this project. The idea (which was dropped) of sorting Pokémon based on what habitat they are found in may have been inspired by the habitat listing from FireRed and LeafGreen's Pokédex, a feature that was removed from all subsequent games.

Questions: *(What questions are you trying to answer? How did these questions evolve over the course of the project? What new questions did you consider in the course of your analysis?)*

Our original question was, what Pokémon exist in the same habitat? We then changed this from habitat to type, as when we were beginning to organize data collection we found that it would be very difficult to determine all of the habitats that each Pokémon exists in due to the nature of the game and the many versions.
How many Pokémon Generation 1 share the same type? We wanted to answer this question visually using our node/link graph to represent all of the links between Pokémon types.

Data: *(Source, scraping method, cleanup, etc.)*

The data we used is scraped from Bulbapedia, an online encyclopedia about Pokémon. By hand, the three of us each took 50 of the 151 Pokémon and collected the type, base stats, and sprite (in-game graphic). The spreadsheet was exported as a .csv file and we utilized some simple R code to read in the csv and transform it to a .json file, the format we desired for use in our project. This R code is in our Git repository. It produced the json file data.json which is also in our Git repository.

Exploratory Data Analysis: *(What visualizations did you use to initially look at your data? What insights did you gain? How did these insights inform your design?)*

We knew when we started the project that we wanted to use a node-link diagram, and so we didn't use any other visualizations.

Design Evolution: *(What are the different visualizations you considered? Justify the design decisions you made using the perceptual and design principles you learned in the course. Did you deviate from your proposal?)*

The original design we were considering was to visualize the habitats or biomes for each Pokémon, having the links between nodes represent which Pokémon shared the same environments. We shifted course from this primarily because determining the biome would require subjectively deciding which biome each location fell into. There are far too many individual locations in any given Pokémon game to not group them, and the original generation games do not clearly indicate which locations fall into which biome. This added subjectivity to the visual seemed not only like unnecessary added work (for not much return), but we also felt that adding unnecessary subjectivity would weaken the final product. Visualizing the different types and how they connect and cluster would be an unchanged visualization of the in-game data, whereas doing the biomes would mean it's visualizing our interpretation of the in-game data. This did deviate from our original proposal, but it certainly strengthened our final product.

Implementation: *(Describe the intent and functionality of the interactive visualizations you implemented. Provide clear and well-referenced images showing the key design and interaction elements.)*

Our implementation is a node-link graph where each individual node represents one of the 151 Generation 1 Pokémon species. Each node is colored based on the species' type, and if the Pokémon has two types, the second type informs the color of the node's outline stroke color. The size of each node's radius is scaled based on the sum of the following statistics: base HP, base Attack, base Defense, base Speed, and base Special. This makes it so it remains visually simple but there is still a representation of each Pokémon's approximate power level. Each node has a link to any other node that shares one (or both) of its type(s). This leads to the nodes clustering based on shared types in the visual, making it clear how the original game's Pokémon selection spreads across the different types.

As for interactivity in our visual, our main priority was in communicating each node's given information. We did this by making it so that when you hover over any node, a small pop-up appears indicating the node's species, type(s), and each of the 5 stats mentioned above. We had attempted to embed an image of the in-game Pokémon sprite for a graphical representation, but were not able to successfully attain this functionality. Additionally, each node can be clicked and dragged across the graph, with the links having rudimentary physics to get pulled along by a moving node, or move out of the way for the node to get through without them overlapping (and risking hiding any information on our visual). This functionality doesn't necessarily have an explicit tie to the data as for *why* we included it, but it makes the graph far more dynamic and engaging.

We also added a legend for the types to make it clearer which colors indicate which types. We originally planned to make it so that, when hovering over a given type listed in the legend, it would in some way highlight all corresponding nodes of that type. We did not end up having time to implement this, however.

We based our visualization off of this example: https://observablehq.com/@mbostock/hello-cola.

Evaluation: *(What did you learn about the data by using your visualizations? How did you answer your questions? How well does your visualization work, and how could you further improve it?)*

We learned that there are many clusters of Pokémon types. There are many Pokémon that share a single type and many that share a specific type combination. We can see how common a type is based on the amount of links a species of that node has. The more links, the more common the type. The less links, the less common the type.
The final visualization makes it very obvious how prevalent the most common type combinations are by producing large clusters containing the Pokémon with those types. The largest four clusters correspond to types of Water, Normal (including Normal/Flying), Poison (including Grass/Poison), and Ground (including Rock/Ground).

Code/website:
The following is the link to our visualization, which is hosted using Github Pages: https://juniorjedi490.github.io/final/.
The source code is on GitHub at https://github.com/JuniorJedi490/final/tree/gh-pages.

Data:
Our data is available in our Git repository. ssv_data.csv is the original data we collected, and data.json is this data in json format.

Presentation:
The video for our presentation is in our Git repository in the videos folder (csfinal-Large 540p.mov). It is also available on YouTube at this link: https://youtu.be/RBVVKYBsrto.

The following are our talking points that we made for our presentation.

What is it?

- Main contributions
  - node/link graph
  - Links pokemon species that are of the same type
  - Shows this using fill and outline color
  - Key on the side which clarifies which color is associated with each species
  - Size of the nodes based on each species stats total
  - Interactive - when you hover over the nodes it tells you the species, type(s), and stats

    - You can drag the nodes to see how that would affect the rest of the graph and which nodes would be most affected
    - Nodes that have more links can more easily drag other nodes its way, while nodes with fewer links do not affect the others as much
- What is the single most important thing you would like your audience to take away?
    - The most important takeaway for our visualization is that it can be used as a learning tool to help people who are new to pokemon gain a better understanding of the different types pokemon
- What is the best part of the project
    - The best part of our project is that You can see a bunch of clusters of different types of gen 1 pokemon
    - Our visualization also shows the variety of pokemon types and the many type combinations that exist
- What insights did we gain
    - While we were working on this visualization we observed that
    - There are many pokemon that share the same combination of types

Works Cited:

https://observablehq.com/@mbostock/hello-cola
https://www.d3-graph-gallery.com/graph/interactivity_tooltip.html

Appendix:

The contents of this section are notes that we made while working on the project.

- [x] ~~Convert data to json for d3 (Henry)~~
- [x] ~~Make node-link graph from the data based on type (Lauren)~~
- [x] ~~Size of nodes based on total (henry)~~
- [x] ~~Interactivity: when hover on node can see the sprite, type, and stats (Alex)~~
- [x] ~~node color: for dif types (henry)~~
- [x] ~~Key: shows what type each color is (Alex)~~
- [ ] If time: when hover over key of type it highlights the nodes

We are sorting by type instead of location; that makes the data collection a lot simpler
(this is the type in modern games as Red and Blue didn't have Dark, Steel, or Fairy as types)

List stats when you hover over the node
Node size: total
Links: type(s)

Notes on creating tooltips with d3:

```
var Tooltip = d3.select(/* svg */)
    .append("div")
    .attr("class", "tooltip")
    .style() // known options: background-color, border, border-width, border-radius, padding,
opacity

tooltip doesn't truly go away, when nothing it is selected it has 0 opacity, or is hidden with style
"visibility": "hidden"

to use the tooltip, create mouse over, mouse move, and mouse leave functions

d3.select(this) used to select the hovered-over element (in over() and leave())
d3.mouse(this) used to get the cursor position (as an array) for finding where to put the tooltip
(in move()) - alternatively, we can use d3.select(this) or event.pageX()/pageY()
appending "px" used to put the cursor position in pixels

svg.selectAll()
        .on("mouseover", mouse over function)
        .on("mousemove", mouse move function)
        .on("mouseleave", mouse leave function);
```

COLORS:
grass - #52AD18
fire - #C72100
water - #0C66C1
bug - #87950D
normal - #B2B2B2
poison - #6D2470
electric - #E79302
ground - #B18F34
fairy - #E28FE2
fighting - #682714
psychic - #DF366C
rock - #A38A3F
ghost - #454592
ice - #71D5F5
dragon - #6751C9
flying - #5D73D4
steel - #8D8D9F

const fills =
["#52AD18","#C72100","#0C66C1","#87950D","#B2B2B2","#6D2470","#E79302","#B18F34","#

```
E28FE2","#682714","#DF366C","#A38A3F","#454592","#71D5F5","#6751C9","#5D73D4",
“#8D8D9F”];

const outlines =
["#399400","#AE0800","#004DA8","#6E7C00","#999999","#540B57","#CE7A00","#98761B","#C
976C9","#4F0E00","#C61D53","#8A7126","#2C2C79","#58BCDC","#4E38B0","#445ABB",
“#747486”];
```