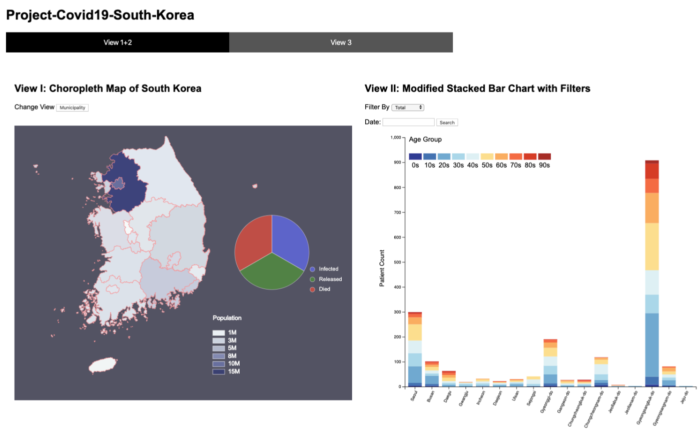
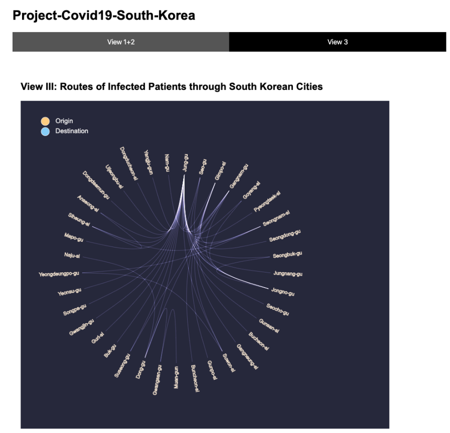
CPSC 436V: M3 Final Project Submission

Rebecca Chen, v6n0b, 18069154
Ben Hwang, r6o0b, 47112157
Danhui Jia, l5y0b, 16119159

OVERVIEW





COVID-19 is an ongoing international public health crisis that has rapidly spread across borders and overseas into countries all over the world. In late February 2020, South Korea was hit with the biggest outbreak of COVID-19 outside of the epicentre. Governance regarding the virus' containment has been a topic of heavy scrutiny by the media and general public. By taking a look at Korean CDC data, we can gain a better understanding of how difficult it was to measure and track the virus in early stages. Users can explore three separate views which visualize the number of infected patients throughout South Korea by location, show a breakdown of patients by age and status across time, and show the routes initial patients took throughout the country in the early stages of the virus. Using our visualization, users will be able to explore details about infected patients and gain a better sense of how this virus has spread so quickly within one country. Our visualization is intended for the general population.

DATA

Korea CDC Dataset
Korea CDC Dataset Detailed Description
Korea Population Dataset
Korea Population Dataset II
Korea Map

**Self-explanatory attribute names are not described. Each variable is labeled with its (data type/cardinality)*

Korea CDC Dataset - table dataset

PatientInfo.csv
Items: each row represents a unique patient
Attributes:

Name	Description/Notes	Data Type	Cardinality
sex	-	Categorical	2 (male or female)
age	Ages are grouped into 10s	Categorical	10 (0s, 10s, 20s, etc.)
confirmed_date, released_date, deceased_date	Date patient is confirmed infected, date patient is deemed recovered, date patient died	Ordinal	~60 days [January 20th - March 20th]
state	Indicates the last known status of the patient	Categorical	3 (recovered, infected, died)

PatientRoute.csv
Items: each row represents the location of a patient at a given time/date
Attributes:

Name	Description/Notes	Data Type	Cardinality
patient_id	Uniquely identifies one patient	Categorical	100,000
province	-	Categorical	8
city	-	Categorical	~30

Time.csv
Items: each row represents aggregate counts for a specific date
Attributes:

Name	Description/Notes	Data Type	Cardinality
date	-	Ordinal	~60 days [January 20th - March 20th]
confirmed, released, deceased	The accumulated number of confirmed infected, released, and deceased patients up to that date	Quantitative	[0-10,000]

TimeProvince.csv
Item: each row represents aggregate counts for a specific date for a specific province
Attributes:

Name	Description/Notes	Data Type	Cardinality
date	-	Ordinal	~60 days [January 20th - March 20th]
province	-	Categorical	17
confirmed, released, deceased	-	Quantitative	[0-100,000]

Korea Population Dataset – table dataset

<u>municipality_population.csv</u>
Items: Municipalities (237 items)
Attributes:

Name	Description/Notes	Data Type	Cardinality
name	-	Categorical	237
pop_2019	-	Quantitative	[9,785-1,500,000]

<u>province_population.csv</u>
Items: Provinces (17 items)
Attributes:

Name	Description/Notes	Data Type	Cardinality
name	-	Categorical	17
pop_2019	-	Quantitative	[346,275-13,653,984]

Korea Map TOPO.JSON – geometry (spatial) dataset

<u>real_provinces. topo.json</u>
Items: Municipality
Positions: arcs to draw a municipality

<u>south_korea_provinces.topo.json</u>
Items: Municipality
Positions: arcs to draw a province

Data Preprocessing Pipeline

<u>View I</u>: Choropleth Map of South Korea with Patient Statistics

- Population for provinces/municipalities is manually scraped from citypopulation.de and renamed province_population.csv / municipality_population.csv
- Pie chart data requires patient count for each status grouped by province/municipality. Preprocessed from PatientInfo.csv and TimeProvince.csv.
  - Municipalities: filter patients in identified municipality, filter by status, count resulting tuples for # of patients by municipality by status. Create data similar format to TimeProvince.csv (province, confirmed, released, deceased)
  - Provinces: taken directly from TimeProvince.csv
  - Final data stored in *patientInfoByMunicipalityAndStatus* and *patientInfoByProvinceAndStatus*

<u>View II:</u> Modified Stacked Bar Chart for Provinces

- Patient data for each province by age group is preprocessed from PatientInfo.csv and includes status and date data for the status and date filters

<u>View III:</u> Innovative View Mapping Routes

- Patient routes data comes directly from PatientRoute.csv. Data was preprocessed to render cities and their links to other cities.
- Final data stored in *patientArray* and *patientRoute.*

GOALS AND TASKS

Project Goals

1. Gain a better understanding of how difficult it is to measure and track the virus.
2. Explore details about infected patients and gain a better sense of how this virus has spread so quickly within one country in order to create an efficient quarantine plan.
3. [Explore] the data in order to [identify] the most infected regions and view the infection growth over time. Identify outliers such as sudden spikes in order to hone in on reasons as to why these spikes happened.
4. See the number of confirmed cases as well as the number of recovered infected, and dead cases by region
5. [Compare] the number of cases between two times, by adjusting a calendar date selection.
6. View the breakdown of cases by the ages of the patients
7. Filter this information by region, and by the different types of confirmed cases (recovered, infected and dead)

View I Task Abstraction

1. Compare the status of patients across regions. Users will be able to see which regions have the most deaths/recoveries in comparison to total patients.
2. Explore potential correlations between different regions (province 1 vs province 2, province 1 vs municipalities within a province 1, municipality 1 vs municipality 2). Users can change between the province and municipality view using a button. E.g. If the two regions with highest % of death/total patients are right next to each other, perhaps a more aggressive strain exists in this region.
3. Locate outlier regions. E.g. Higher % deaths might mean more supplies are needed in the region, higher recovery % might mean that region is doing something good that other regions should copy.

View II Task Abstraction

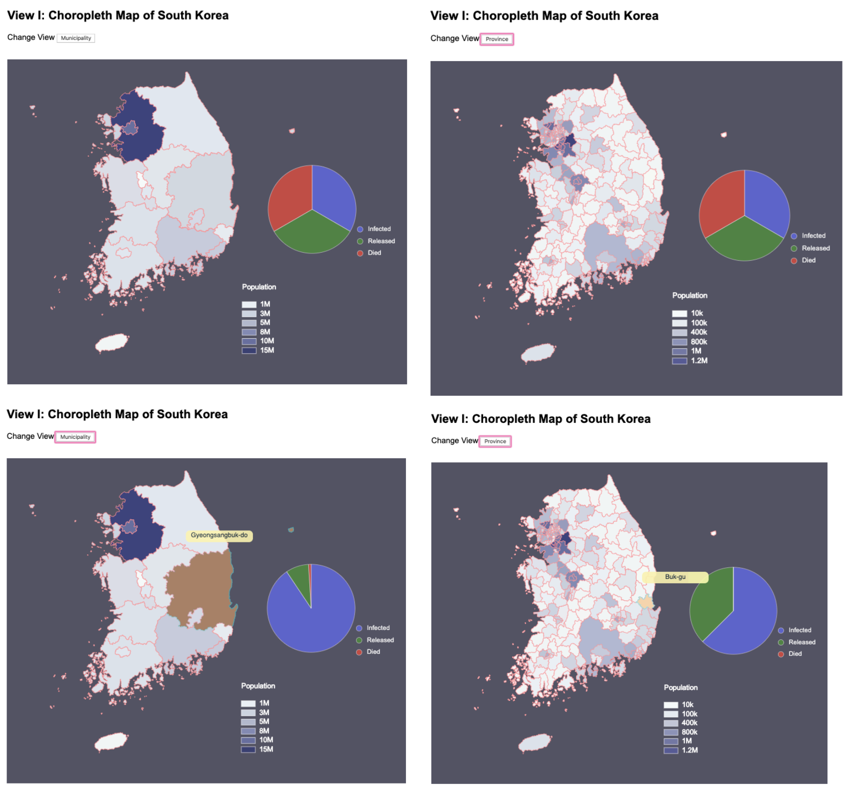
1. Query number of patients by province and a combination of status, and date. Users can use the dropdown and calendar interaction elements to filter by status and date.
2. Present trends of patient count in provinces across time. Users can change the date to view the increase of patients over time by age group.
3. Identify outliers in age groups. E.g. Perhaps the number of patients of a specific age group is proliferating faster than other age groups.

View III Task Abstraction

1. Explore the movement of patients through cities before they were identified as infected.
2. Compare the number of paths going through one location to other locations.

VISUALIZATION

View I – Choropleth Map of South Korea



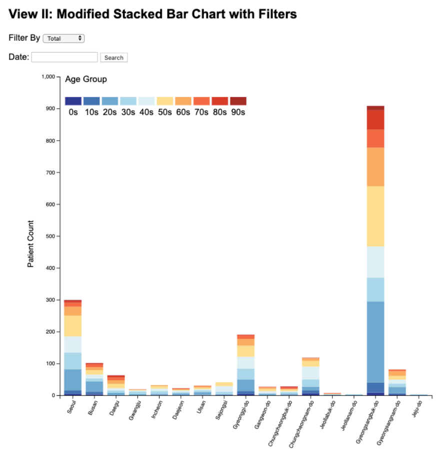
→ As we found a topo.JSON file for both South Korean provinces and municipalities, we decided to represent the categorical province and municipality names with their spatial geometry form. They are represented in a choropleth map view. The spatial relation between provinces/municipalities can possibly provide insights for the spread of the virus.

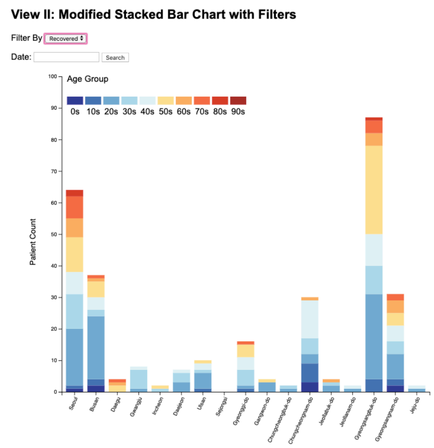
→ We represented the quantitative population of each region using a sequential colour scale.

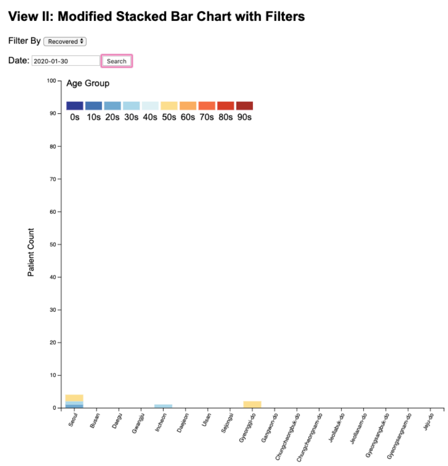
→ We represented the quantitative number of patients of each status using a pie chart that updates based on which province/municipality the user hovers over. This will be able to show the parts-per-whole relationship of each individual status per region. While a pie chart is usually a bad visualization to use, we thought it was ok for in this case since we are simply looking for the relative volume of infected, released, and died patients. We don't necessarily need the exact number. The pie chart has a categorical colour scheme as the patient statuses are categorical.

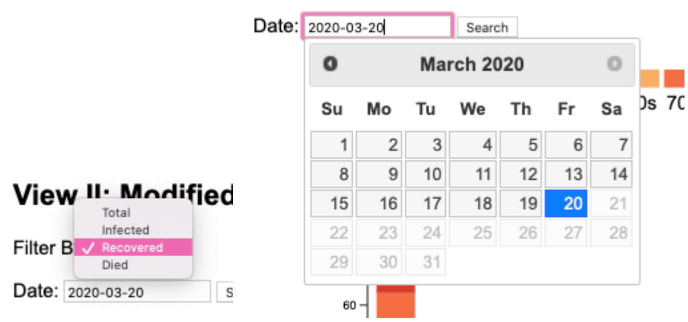
→ We included a tooltip which shoes the province/municipality name that you are hovering over.

View II – Modified Stacked Bar Chart with Filters









→ We represented the categorical provinces using the x-axis of a bar graph. Each bar's horizontal spatial position encodes its province. We decided to use the length encoding for the bars to encode the patient count, and used colour coded stacked bars to show breakdown of the patient count by their age groups.

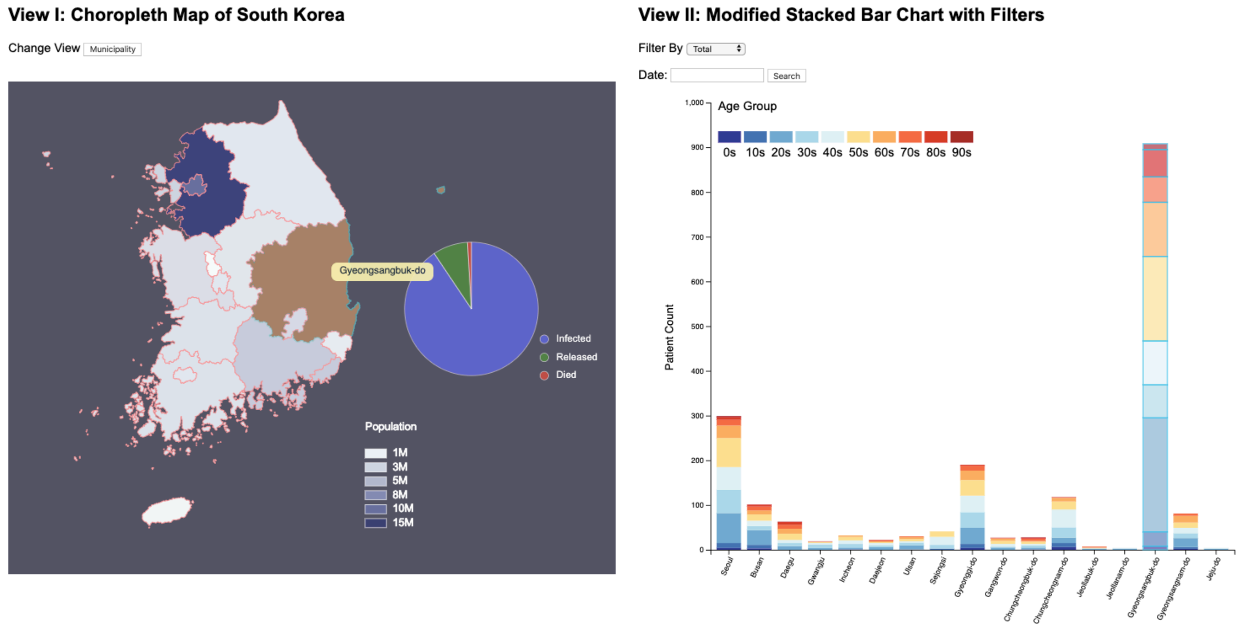
→ While we were cautioned against using a diverging colour scheme for the patient age groups in M2, we decided to keep it, as we wanted to clearly distinguish patients in the >=50 age group from patients in the <50. This is why we chose to diverge the colours between the 40s and 50s age group from blue to orange/red. This is because COVID-19 is a disease that greatly affects patients who are more senior. An example of how this makes our visualization clearer, is when you select Filter by "Died". You can see that the majority of the deaths are from individuals who are >50 years of age. The diverging colour scale we used has 10 buckets, however, the colours we chose are quite easily distinguishable from each other as there is sufficient difference in hue/saturation between them. It is actually quite easy to differ between 50s and 60s even though they are both light shades of orange/yellow, since they are located right next to each other on the stacked bar. If they were not located side-by-side, it would be more difficult to distinguish between the two.

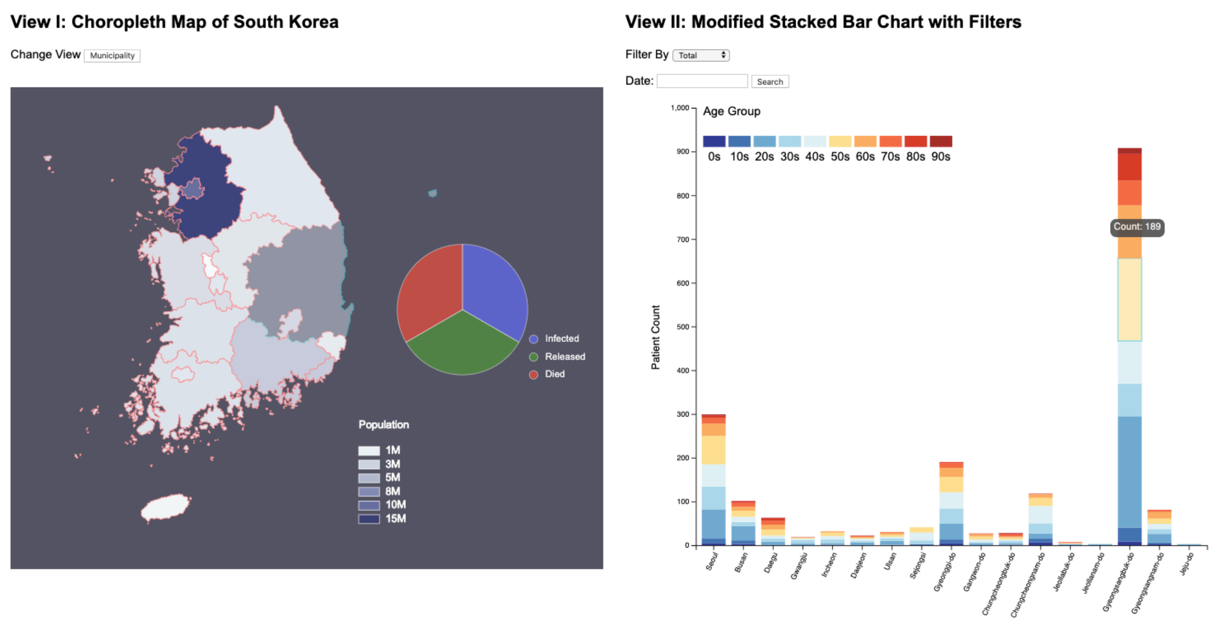
→ We included a tooltip which shows the number of patients in each age group if you hover over that age group's colour in the bar.

→ We included an interaction dropdown to allow users to filter by status [infected, recovered, died, all statuses], so that the patient count only includes patients in one province who are infected, recovered, died, or all at one time.

→ We also included a second interaction calendar date picker to allow users to select a specific date, so that the patient count includes all patients in one province on one date. These two interactions stack, so that a user can filter first by status = infected and then date = x.

Interaction between View I and View II

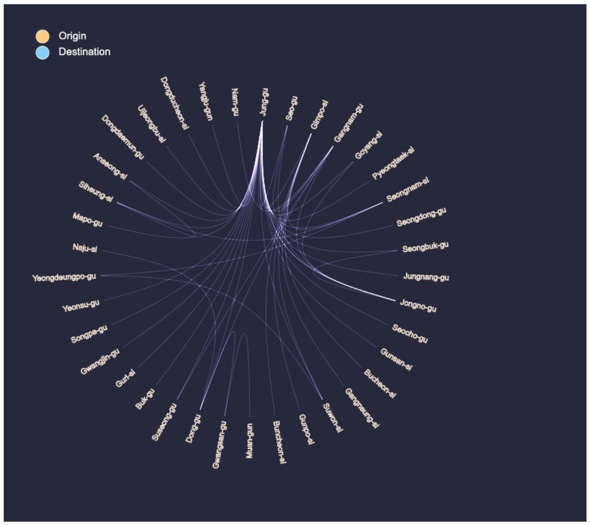


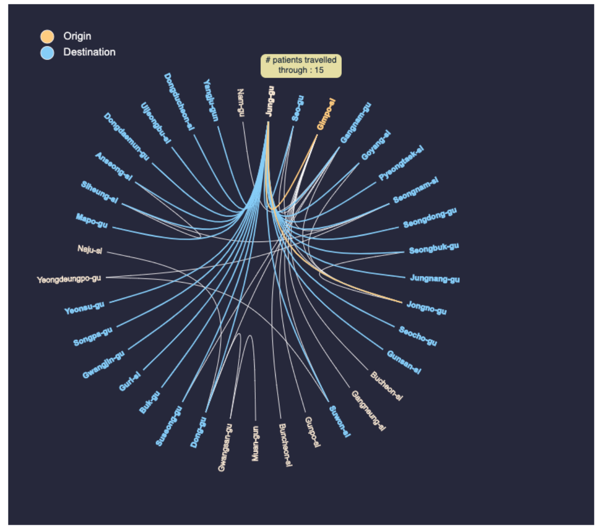


→ When a user hovers over a province in View I, the corresponding bar for that province in View II is highlighted.
→ When a user hovers over a bar in View II, the corresponding province in View I is highlighted.
→ An example of how this could be useful: Filter by "Died" in View II. Notice that most deaths occurred in Daegu. Hover over the bar chart, which highlights Daegu province in the map. Notice that the province with most deaths is not the one with the highest population, nor is it the one with the greatest number of infected patients. Why this is so can be explored with further research into the demographics and specific outbreaks in that region.

View III – Routes of Initial Infected Patients through South Korean Cities

View III: Routes of Infected Patients through South Korean Cities





→ We represented the categorical locations (city level) in a radial axis where the spatial position of each line end point represents the categorical location a COVID-19 patient visited before they were diagnosed. There are roughly 30-40 cities, so the round radial axis fits them all nicely. A single line between two cities represents the movement of one or more infected patients between the two cities.

→ Users are able to hover over a city will highlight all the routes infected patients took to and from that city. The orange highlight indicates a patient that travelled from an ORIGIN city into the SELECTED (hovered) city, and the blue highlight indicates a patient that travelled from the SELECTED (hovered) city to a DESTINATION city. An example of how this is useful is if you hover over Jung-gu municipality. You can see that many patients travelled through this city. You can now go into View I and View II search for Jung-gu municipality in the map and see that it is not even one of the most densely populated provinces/municipalities. Further investigation is needed to pinpoint why so many infected patients travelled through this area (maybe there is a big transportation hub in the area).

REFLECTION

Describe how your project has developed from the initial proposal

Going into our first project meeting, we separately researched datasets that we could potentially use. Like many other groups, the COVID-19 situation was new and interesting at the time we started (little did we know the topic lost its fun and novelty once the situation got worse around the world). We found that many datasets with worldwide data only had aggregate counts, which we could not do much with. However, we found on Kaggle our dataset specific to South Korea, which had details such as patient ages, routes, and aggregate counts by province/municipality. After choosing this dataset, we got together and white boarded ideas in a project room for 3 hours, which resulted in our M1 sketches. As each deadline came and passed, we revised our visualization goals when easier solutions popped up, or we found a better way to implement something. Through the last week of development, we had daily calls to align our work.

How have your visualization goals changed?

Our visual representations have changed slightly, due to the fact the dataset we were working on was reformatted and updated. For example, going into M2, we found that more data was uploaded which increased the cardinality of dates and patient count. This made our original plan for View II (stacked symbols each representing one patient) difficult. We then replaced the symbols with stacked bars, which scale better and more efficiently encapsulate all the data. Our overarching visualization goals remain the same.

How have your technical goals changed?

Our technical goals have not changed throughout the project. We set out to figure out how to implement our View III as a radial axis with links in between even though we had no template to follow and we ended up implementing it without much problem. We also wanted to keep our code clean and consistent and follow proper update pattern coding despite multiple people working on the base at one time and we had no issues with this as well.

How realistic was your original proposal in terms of what is technically possible in d3?

Our original proposal was realistic in terms of what is technically possible in d3. View I and View II were more or less straightforward and based on the choropleth map and stacked bar graph's we implemented in assignments. We based our innovative view on a D3 visualization called hierarchical edge bundling (https://www.d3-graph-gallery.com/bundle.html), however since our data was not hierarchical, we adjusted it and utilized a radial axis (https://github.com/vasturiano/d3-radial-axis) to create something more similar to a connection map where the vertices are not locations on a map, but locations along the radial axis (https://www.d3-graph-gallery.com/connectionmap.html).

Was there anything you wanted to implement that you ultimately couldn't figure out how to do? If so then what workarounds did you employ, or did you abandon your original idea?

We did have some stretch goals that we eventually did not implement due to time constraints. These include further user interactions with View III such as a popup chart with further information on the cities/patients and linking it with View I and View II. We abandoned these stretch goals.

For View III we also originally envisioned being able to individually highlight one specific patient's route, however due to the way we structured the data, it would have been difficult for us to render the lines so that two patients that went through the same point A and point B would have visible lines (they would overlap). We would have had to change the curve of the line. Because of this, we decided not to allow users to trace an individual patient's path, and instead, we rendered all paths between cities and labeled the number of patients that took that path. We found that this might be more useful/clear to users in achieving our main goal for View III, which is to identify which cities had most patients going through them.

For View I, we initially wanted to implement mini pie charts on the choropleth map of provinces. However, after we changed our View I to include municipalities, we found that there were too many municipalities to rendering an individual pie chart for each. This would have been difficult to look at and not useful in achieving our overarching goal for View I. We then found the workaround of creating a sub-view with a pie chart in View I and linking that to our choropleth map to update based on which province/municipality the user is hovering over.

Lastly, for View I, while this was not originally in our project plan, it would have been helpful to have the provinces outlined in the municipality view. Since we did not plan for this we did not implement it, however if we had extra time it would have been good to have to allow the user to better determine which province belonged to each municipality.

If you were to make the project again from scratch, what would you do differently?

If we were to make the project again from scratch, we would definitely narrow our initial goal for the visualization down to something more specific. Our original goals of "gain a better understanding of how difficult it is to measure and track the virus," and "explore details about infected patients, and gain a better sense of how this virus has spread so quickly" are quite vague, and don't narrow down the visual possibilities for our project. There was a lot of variance in our ideas at the beginning that caused lag in work. While we did match up the data types with the correct visual encodings, we definitely cannot definitively say we chose the best possible data and views to achieve these goals since the goals are quite open to interpretation. A more specific goal, such as "track the **growth** rate of the virus over time" would have narrowed down the possible View II and View III options we had, and allow us to derive some data from the dataset, instead of just transforming the existing data into a form that allowed us to build our views.

TEAM ASSESSMENT

Describe each team members tasks and responsibilities throughout the project

Updated work breakdown and schedule from M2

Black = original
~~Red~~ = removed
Green = new tasks
~~Strike~~ = completed

Milestone	Target Hours / Finish Hours	Target Deadline/ Finish Date	Ben	Danhui	Rebecca
~~M1- Proposal Discussion and datasets~~	~~15~~ / 10 hours	~~Mar 7~~ / March 10th	~~Sec 3, 4~~	~~Sec 6~~	~~Sec 2, 5, dataset~~
~~M2-1 Implement choropleth map of Korea (v1: static view)~~	~~21~~ / 20 hours	~~Mar 15~~ / March 18th	~~Map render (South Korea Topo), titles, scales, svg set up~~	~~Data Processing (pre-process, fiter, internal data structure), gather additional datasets if needed~~	~~Color label, color legends, render custom symbol~~ ~~create tabs, data processing (manual)~~
~~M2-2 Implement stacked graph with glyphs: different symbols (v2: static view)~~	~~18~~ / 15 hours	~~Mar 22~~ / March 24th	~~Data processing for specific attributes, render axises, axis scales~~	~~Grouping desired attributes, static stacked graph render~~	~~Customized symbol render~~, ~~symbol legends,~~ ~~start writeup for M2~~
~~M2-WIP Discussion of innovative v3~~ ~~Final WIP writeup~~	~~12~~ / 15 hours	~~Mar 24~~ / March 27th	~~Data preprocessing pipelining, render bars by age~~	~~Render bars by age, color render,~~	~~Status update, contribution breakdown, team process, link, screenshot, rationale for design choices and vision changes (if any),~~
~~M3-1.1 Wrap-up of M2~~	10	March 29	~~Data preprocessing pipeline~~	~~Color label, color legends~~	~~Start writeup for M3~~
~~M3-1.2 Interactive links between v1,v2 and 2 UI widgets.~~ ~~Final WIP writeup~~	~~18~~ / 15 hours	~~Mar 29~~ Mar 31	~~Event listener UI1, specific interaction (tooltips)~~	~~Event listener UI2 (temp: radio button input changes y-axis render of v2), bidirectional hovering highlight v1,v2~~	~~Html drop-down menu/button, event listener caller on both v1, v2, data filter according to user input~~

~~M3-2.1 Innovative view data manipulation and processing Understand what data is needed and how to transform current data~~	~~12~~ / 15 hours	~~Mar 31~~ April 2	~~Resources examples that are related and useful for v3: innovative graph rendering, understand data usage~~	~~Resources examples that are related and useful for v3: innovative graph rendering, understand data usage~~	~~Resources examples that are related and useful for v3: innovative graph rendering, understand data usage~~
~~M3-2.2 Implement innovative view 3 (static)~~ ~~Implement view 2~~	~~30~~ / 35 hours	April 5	~~Data process, svg set up, render graph, implement pie chart~~	~~Render graph, color scale, color label~~	~~Render graph, symbol label, text elements, implement pie chart~~
~~M3-(2.3 optional) User interactive transition (interactive)~~	~~(if time permits)~~	~~April 5~~	~~Popup chart content [bar chart]~~	~~Linking v3 to v1,v2, event listener on mouse click~~	~~Tooltip for hovering rendering popup chart~~
~~M3-3 Project refinement Define final implementation Add descriptions~~	~~6~~ / 15 hours	~~April 6~~ April 10	~~Align text usage across multiple views, improvement on color, scale, descriptive title and text (v1)~~	~~Clean up code, comments, logs, improvement on color, scale, descriptive title and text (v2)~~	~~Clean up code, align color usage across multiple views, improvement on color, scale, descriptive title and text (v3).~~
~~M3-Final Project Write up Proof-read Assessment~~ ~~Remaining view implementation~~	~~18~~ / 10 hours	April 10	~~Data, team assessment, reflection, project self-assessment, v1 and v3 cleanup~~	~~Goal and tasks, visualization, reflection, project self-assessment, v1 and v3 cleanup~~	~~Data, goal and tasks, overview, visualization, reflection, team assessment, project self-assessment~~