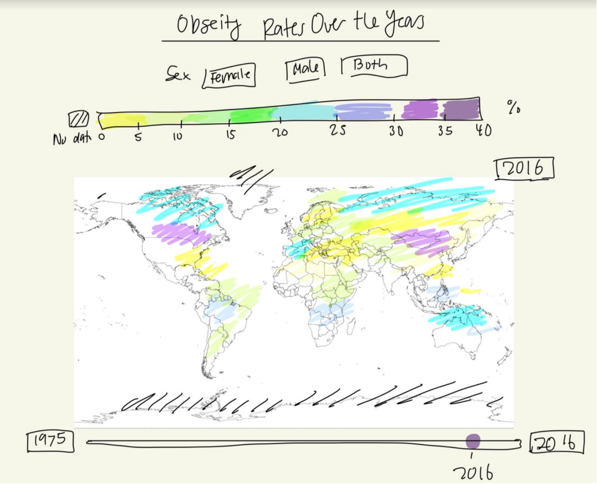
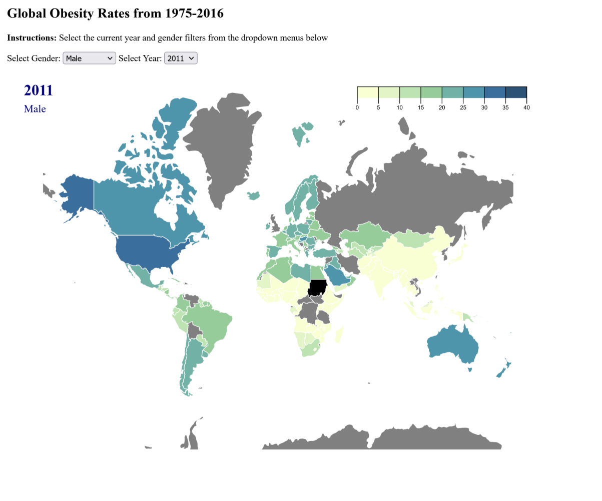
Project 2 Final Report

Names: Reade Plunket (rwp227), Tiffany Pan (tp284), Kai Tong Gao (kg428), Vin Bui (vdb23)

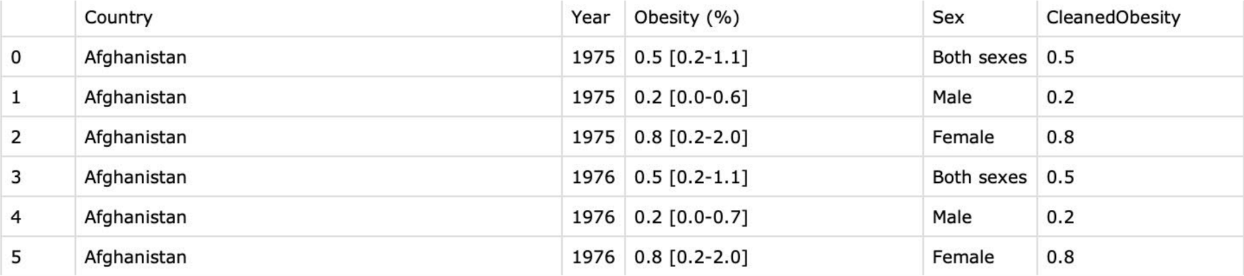
Screenshots

Sketch	Final
 Obseity Rates Over the Years Sex Female Male Both No data 0 5 10 15 20 25 30 35 40 % 2016 1975 2016 2016	Global Obesity Rates from 1975-2016 **Instructions:** Select the current year and gender filters from the dropdown menus below Select Gender: Male Select Year: 2011 2011 Male

Data Description

Report where you got the data. Describe the variables. If you had to reformat the data or filter it in any way, provide enough details that someone could repeat your results. If you combined multiple datasets, specify how you integrated them. Mention any additional data that you used, such as shape files for maps. Editing is important! You are not required to use every part of the dataset. Selectively choosing a subset can improve usability. Describe any criteria you used for data selection.

We sourced our first dataset, called “Obesity Among Adults by Country, 1975-2016” from Kaggle, which was provided by Aman Arora. This dataset contains information about the percentage of adults who are obese in each country depending on the year and the adults’ sex. Each row of the data indicates a country’s obesity rate in a particular year for either both sexes, male, or female. A sample of the data can be seen below:



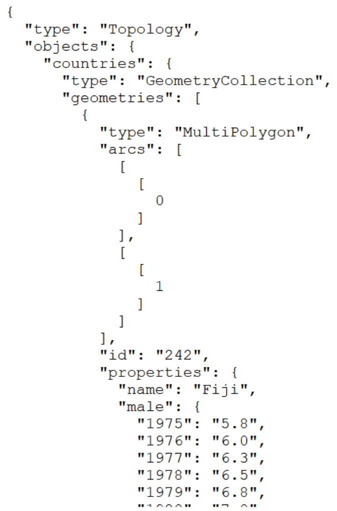
	Country	Year	Obesity (%)	Sex	CleanedObesity
0	Afghanistan	1975	0.5 [0.2-1.1]	Both sexes	0.5
1	Afghanistan	1975	0.2 [0.0-0.6]	Male	0.2
2	Afghanistan	1975	0.8 [0.2-2.0]	Female	0.8
3	Afghanistan	1976	0.5 [0.2-1.1]	Both sexes	0.5
4	Afghanistan	1976	0.2 [0.0-0.7]	Male	0.2
5	Afghanistan	1976	0.8 [0.2-2.0]	Female	0.8

The obesity (%) value was manipulated to create a new variable CleanedObesity which we referenced in our data visualization as the country's obesity value.

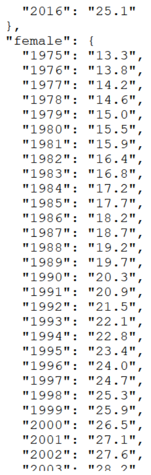
Since we wanted to visualize obesity rates across a multitude of countries, we also needed to have a json dataset that would show the entire globe. Thus, after doing some research on the available topo.jsons that exist for the world, we settled on the one from World Atlas. Within this repository, there were various options depending on what kind of information we wanted to be included. Since we only wanted to show countries, we chose the countries-110m.json that visualized just land rather than one that included both land and water, since we wouldn't need the extra information.

Originally, we had used these two datasets separately and loaded datasets in our code. Our topo.json's purpose was to simply map the geography for the visualization and we created functions that searched for the corresponding obesity value in the Kaggle dataset based on the country's name and the current year and sex selected in our drop down menu. However after creating the first draft of the visualization we realized that because the structure of the Kaggle dataset was so complex, we needed to do math every time we filtered the value to be displayed on the map (i.e. filtering by year and sex). This resulted in our visualization being quite laggy, as values needed to be re-computed over and over again depending on the user's filters, despite the fact that we are essentially re-loading the same information each time. Therefore, we decided it would be better to do some data-cleaning so that our visualization could perform without any lag.

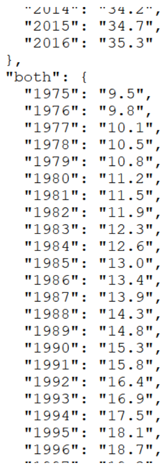
Using the Python script screenshotted in the appendix, we modified our topo.json file to include more parameters in Object.properties for each country object. Along with the country's name, a nested json string that includes each country's obesity rate per year with different gender keys (both, female, male) was added to Object.properties. Then, in our Javascript code, this allowed us to be able to directly access the corresponding obesity values and without needing to perform heavy calculations every time the user filters. A **sample** of the modified topojson file is below for the country named "Fiji":



```
{
  "type": "Topology",
  "objects": {
    "countries": {
      "type": "GeometryCollection",
      "geometries": [
        {
          "type": "MultiPolygon",
          "arcs": [
            [
              [
                0
              ]
            ],
            [
              [
                1
              ]
            ]
          ],
          "id": "242",
          "properties": {
            "name": "Fiji",
            "male": {
              "1975": "5.8",
              "1976": "6.0",
              "1977": "6.3",
              "1978": "6.5",
              "1979": "6.8",
              "1980": "7.0"
```



```
  "2016": "25.1"
},
"female": {
  "1975": "13.3",
  "1976": "13.8",
  "1977": "14.2",
  "1978": "14.6",
  "1979": "15.0",
  "1980": "15.5",
  "1981": "15.9",
  "1982": "16.4",
  "1983": "16.8",
  "1984": "17.2",
  "1985": "17.7",
  "1986": "18.2",
  "1987": "18.7",
  "1988": "19.2",
  "1989": "19.7",
  "1990": "20.3",
  "1991": "20.9",
  "1992": "21.5",
  "1993": "22.1",
  "1994": "22.8",
  "1995": "23.4",
  "1996": "24.0",
  "1997": "24.7",
  "1998": "25.3",
  "1999": "25.9",
  "2000": "26.5",
  "2001": "27.1",
  "2002": "27.6",
  "2003": "28.2"
```



```
  "2014": "34.2",
  "2015": "34.7",
  "2016": "35.3"
},
"both": {
  "1975": "9.5",
  "1976": "9.8",
  "1977": "10.1",
  "1978": "10.5",
  "1979": "10.8",
  "1980": "11.2",
  "1981": "11.5",
  "1982": "11.9",
  "1983": "12.3",
  "1984": "12.6",
  "1985": "13.0",
  "1986": "13.4",
  "1987": "13.9",
  "1988": "14.3",
  "1989": "14.8",
  "1990": "15.3",
  "1991": "15.8",
  "1992": "16.4",
  "1993": "16.9",
  "1994": "17.5",
  "1995": "18.1",
  "1996": "18.7",
```

Design Rationale

A good rule of thumb to follow is "every pixel must be justified." Instead of a 100,000-element breakdown, give us an overview of the design decisions you made and the trade-offs inherent in how you displayed the data. This part ought to include a description of the mapping from data to visual elements. Describe marks and channels you employ such as position, color, or shape. Mention any transformations you performed, such as log scales.

For our visualization we wanted to create a visualization that showed how obesity values changed in the world from the years 1975-2016. We decided to use a topojson file to create a visualization of the world map in which each country would be filled in with a color that mapped to a range where the country's obesity value (specific to the year the data was taken for what subset of the population, female, male or both sexes) fell in.

We decided to use a sequential color scale to color in the different countries because the data is continuous: i.e obesity values range from about 0% to 30% of the population. The color scale would take in the obesity value associated with a specific country and map it to a shade of blue through the **d3's scaleThreshold** which uses a sequential color scale as the range. ScaleThreshold maps each of the colors we chose for our scale to a value in the range of obesity values. We chose this over scaleQuantile because when we tried to use scaleQuantile(), we noted that the majority of the values fell into the same color brackets so the colors on the map looked very similar to one another, which made it very hard to distinguish between the different obesity values.

Mapping an increase in obesity rate to an increase in saturation in fill color would be easily understood to viewers because as the obesity rate increases, the saturation of a color is also increasing as well. To allow viewers to have a better understanding of where the obesity rate "sits" among all the rates, we decided to include a legend that clearly states for which obesity value each of the color ranges starts and ends. That way, viewers who change the years or gender selection in our drop down menu can quickly judge the range each country's obesity rate lies as the fill colors change. Furthermore, to allow for an even more exact understanding of obesity rates among different countries, we decided that there would be a hover feature where a country's name and obesity value would be shown when hovered.

Interactive Elements and their Design Rationale

Give us an outline of the design decisions that went into the interaction affordances you added to your visualization. What process did you use to choose the interactions you developed? How did you make them discoverable, usable, and interesting?

Our visualization has **three** interactions, two of which are **dropdown** menus where the data is filtered by year and sex. The obesity value that is used in the visualization changes depending on the values selected for these two filters. Not only does the visualization interact with the dropdown menus, our visualization also has a **hover** feature in which the country's name and exact obesity value can be seen when the country is hovered over.

Initially, we decided to implement both the gender and year parameters as dropdowns. We made these discoverable by placing them at the top of the visualization right underneath the title. This draws the viewer's eye to what they are looking at and presents a clean, usable interface for changing how they view the obesity rates on the map. These interactive components are interesting, because they change the visualization depending on what year or gender the user is analyzing. It is also interesting to compare year to year or gender to gender how each country's obesity rate changes. Additionally, the user can even compare gender to year. This allows the user to view over a hundred various combinations of the two to view the rates of obesity throughout the world. We also added labels on the top left corner of the visualization that changed text as the dropdown selection changed so that users received feedback from not only the changing colors of the countries but also the text label so they would be able to easily reference the year and gender selection they had chosen from the drop down menu.

The Story

What does your visualization tell us? What was surprising about it? What insights do you want to convey to the viewer of your visualization?

Our visualization conveys how obesity rates have changed regionally throughout the last few decades. It is really surprising to see just how rapidly these rates have increased in some parts of the world, while others experienced little to no change. For example, in the United States, in 1975 the obesity rate for both sexes was 11.9%, still relatively high compared to the rest of the world. In 2016, that rate increased to 19.0%. In Algeria, the obesity rate in 1975 was 6.9% for both sexes. That rate skyrocketed to 16.9% by 2016. This begs the question, why did the obesity rate double in those years? The middle east has also experienced a tremendous increase in obesity rates. This could be possibly due to the rapid development of those areas in recent decades because of the region's rich oil resources. Meanwhile in Asia, the obesity rates have seen little change. In 1975 the obesity rates for China, India, and Japan had rates of 0.5%, 0.3%, and 1.0% respectively. In 2016, those obesity rates for all of those countries increased on average by under 1%, which wasn't a surprise as some Asian countries such as Japan are often known for having healthy local cuisine and eating habits.

With respect to analyzing gender, we also found some interesting things to share. In 1975, the obesity rate for males was lower than that of females in the United States. The male obesity rate was 10.3% while the women's obesity rate was 13.3%. This pattern remained consistent as well. Although both men and women faced increasing rates of obesity throughout the decades with rates of 35.5% and 37.0% respectively, women rates were higher than men in every single year. This can possibly be attributed to biological factors, difference in lifestyle, or pregnancy-related weight gain.

We also found it surprising how Russia, Iran, the United Kingdom, and South Korea had no data. These are very popular and powerful countries, and we were curious why there was no data for these countries in the dataset. Do the governments in those countries not release such statistics? How would the data for these countries compare in relation to neighboring countries?

Team Contributions

Reade: Read helped research some potential datasets and helped set up the GitHub repo. He also worked on implementing the interactive dropdowns for viewing the obesity rates by gender and year on the map.
Tiffany: Cleaned the obesity values in the Obesity data set to create the CleanedObesity variable; worked on implementing the map of the topojson to interact with the color scale.
Vin: Added new properties in the topojson data using the obesity dataset so that obesity values could be easily assessed through the topojson file only.
Kai: Worked on the color scale, year and gender labels and created a legend showcasing the color scale used for the visualization and implemented a hover feature that showed the exact obesity value of a country when hovered.

```python
import json


def get_country_data(country, obesity_data):
    dict = {}
    male_dict = {}
    female_dict = {}
    both_dict = {}

    for o in obesity_data:
        if o["Country"] == country:
            sex = o["Sex"]
            obesity = o["CleanedObesity"].replace(" ", "")
            year = o["Year"]

            if sex == "Male":
                male_dict[year] = obesity
            elif sex == "Female":
                female_dict[year] = obesity
            elif sex == "Both sexes":
                both_dict[year] = obesity

    dict["male"] = male_dict
    dict["female"] = female_dict
    dict["both"] = both_dict

    return dict


def clean_data():
    obesity_file = open("obesity.json")
    topo_file = open("topo.json")

    obesity_data = json.load(obesity_file)
    topo_data = json.load(topo_file)

    for t in topo_data["objects"]["countries"]["geometries"]:
        country = t["properties"]["name"]
        country_dict = get_country_data(country, obesity_data)
        t["properties"]["male"] = country_dict["male"]
        t["properties"]["female"] = country_dict["female"]
        t["properties"]["both"] = country_dict["both"]

    obesity_file.close()
    topo_file.close()

    json_object = json.dumps(topo_data, indent=2)
    with open("new_data.json", "w") as outfile:
        outfile.write(json_object)


if __name__ == "__main__":
    clean_data()
```