

## Q3 [15 points] D3 Warmup - Visualizing Wildlife Trafficking by Species

In this question, you will utilize a dataset provided by [TRAFFIC](#), an NGO working to ensure the global trade of wildlife is legal and sustainable. TRAFFIC provides data through their interactive Wildlife Trade Portal, some of which we have already downloaded and pre-processed for you to utilize in Q3. Using species-related data, you will build a bar chart to visualize the most frequently illegally trafficked species between 2015 and 2023. Using D3, you will get firsthand experience with how interactive plots can make data more visually appealing, engaging, and easier to parse.

Read chapters 4-8 of Scott Murray's [Interactive Data Visualization for the Web, 2nd edition](#) (sign in using your GT account, e.g., jdoe3@gatech.edu). This reading provides an important foundation you will need for Homework 2. The question and autograder have been developed and tested for D3 version 5 (v5), while the book covers v4. What you learn from the book is transferable to v5, as v5 introduced few breaking changes. We also suggest briefly reviewing chapters 1-3 for background information on web development.

TRAFFIC International (2025) Wildlife Trade Portal. Available at [www.wildlifetradeportal.org](http://www.wildlifetradeportal.org).

Technology	<ul style="list-style-type: none"><li>D3 Version 5 (included in the lib folder)</li><li>Chrome 97.0 (or newer): the browser for grading your code</li><li>Python HTTP server (for local testing)</li></ul>
Allowed Libraries	D3 library is provided to you in the <b>lib</b> folder. You must <b>NOT</b> use any D3 libraries (d3*.js) other than the ones provided.
Deliverables	<ul style="list-style-type: none"><li><b>Q3.html</b>: Modified file containing all html, javascript, and any css code required to produce the bar plot. Do not include the D3 libraries or q3.csv dataset.</li></ul>

### IMPORTANT NOTES:

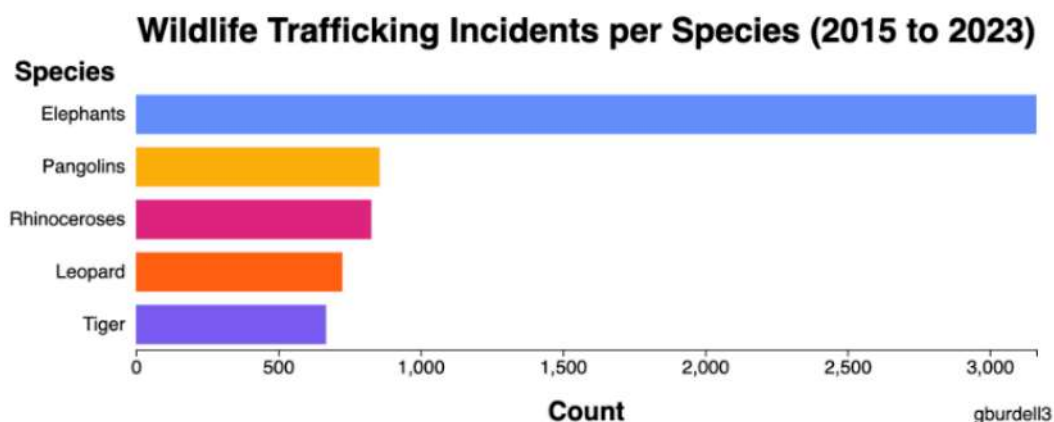
- Setup an HTTP server to run your D3 visualizations as discussed in the D3 lecture (OMS students: [watch lecture video](#). Campus students: see [lecture PDF](#)). The easiest way is to use [http.server](#) for Python 3.x. **Run your local HTTP server in the hw1-skeleton/Q3 folder.**
- We have provided sections of skeleton code and comments to help you complete the implementation. While you do not need to remove them, you need to write additional code to make things work.
- All d3\*.js files are provided in the **lib** folder and referenced using **relative paths** in your html file. For example, since the file "Q3/Q3.html" uses d3, its header contains: `<script type="text/javascript" src="lib/d3/d3.min.js"></script>`. **It is incorrect to use an absolute path such as:** `<script type="text/javascript" src="http://d3js.org/d3.v5.min.js"></script>`. The 3 files that are referenced are:
  - a. lib/d3/d3.min.js
  - b. lib/d3-dsv/d3-dsv.min.js
  - c. lib/d3-fetch/d3-fetch.min.js
- In your html / js code, use a **relative path** to read the dataset file. For example, since Q3 requires reading data from the q3.csv file, the path must be "q3.csv" and **NOT** an absolute path such as "C:/Users/polo/HW1-skeleton/Q3/q3.csv". Absolute paths are specific locations that exist only on your computer, which means your code will **NOT** run on our machines when we grade, and you will lose points. **As file paths are case-sensitive, ensure you correctly provide the relative path.**
- Load the data from q3.csv using D3 fetch methods. We recommend d3.dsv(). Handle any data conversions that might be needed, e.g., strings that need to be converted to integer. See <https://github.com/d3/d3-fetch#dsv>.
- VERY IMPORTANT:** Use the [Margin Convention](#) guide to specify chart dimensions and layout.

Tasks and point breakdown

**Q3.html:** When run in a browser, should display a **horizontal** bar plot with the following specifications:



1. [3.5 points] The bar plot must display one bar for each of the five most trafficked species by count. Each bar's length corresponds to the number of wildlife trafficking incidents involving that species between 2015 and 2023, represented by the 'count' column in our dataset.
2. [1 point] The bars must have the same fixed thickness, and there must be some space between the bars, so they do not overlap.
3. [3 points] The plot must have visible X and Y axes that scale according to the generated bars. That is, the axes are driven by the data that they are representing. **They must not be hard-coded.** The x-axis must be a `<g>` element having the `id: "x_axis"` and the y-axis must be a `<g>` element having the `id: "y_axis"`.
4. [2 points] Set x-axis label to 'Count' and y-axis label to 'Species'. The x-axis label must be a `<text>` element having the `id: "x_axis_label"` and the y-axis label must be a `<text>` element having the `id: "y_axis_label"`.
5. [2 points] Use a linear scale for the X-axis to represent the count (recommended function: `d3.scaleLinear()`). Only display ticks and labels at every 500 interval. The X-axis must be displayed below the plot.
6. [2 points] Use a categorical scale for the Y-axis to represent the species names (recommended function: `d3.scaleBand()`). Order the species names from greatest to least on 'Count' and limit the output to the top 5 species. The Y-axis must be displayed to the left of the plot.
7. [1 point] Set the HTML title tag and display a title for the plot. **Those two titles are independent of each other and need to be set separately.** Set the HTML title tag (i.e., `<title> Wildlife Trafficking Incidents per Species (2015 to 2023)</title>`). Position the title "Wildlife Trafficking Incidents per Species (2015 to 2023)" above the bar plot. The title must be a `<text>` element having the `id: "title"`.
8. [0.25 points] Add your GT username (usually includes a mix of letters and numbers) to the area beneath the bottom-right of the plot. The GT username must be a `<text>` element having the `id: "credit"`
9. [0.25 points] Fill each bar with a unique color. We recommend using a [colorblind-safe palette](#).



**NOTE:** Gradescope will render your plot using Chrome and present you with a Dropbox link to view the screenshot of your plot as the autograder sees it. This visual feedback helps you adjust and identify errors, e.g., a blank plot indicates a serious error. Your design does not need to replicate the solution plot. However, **the autograder requires** the following DOM structure (including using **correct IDs** for elements) and sizing attributes to know how your chart is built.

```

<svg id="svg1"> plot
  | width: 900
  | height: 370
  |
  +-- <g id="container"> containing Q3.a plot elements
    |
    +-- <g id="bars"> containing bars
      |
      +-- <g id="x_axis"> x-axis
        |
        | +-- (x-axis elements)
        |
        +-- <text id="x_axis_label"> x-axis label
          |
          +-- <g id="y_axis"> y-axis
            |
            | +-- (y-axis elements)
            |
            +-- <text id="y_axis_label"> y-axis label
              |
              +-- <text id="credit"> GTUsername
                |
                +-- <text id="title"> chart title

```