Lab 6: D3 Enter, Update & Exit (Activities)

Wijaya, Mario edited this page 14 hours ago · 1 revision

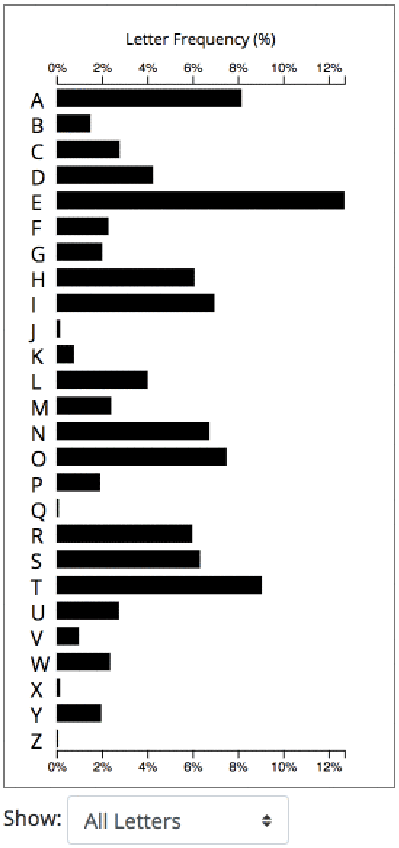
Activity 0 - Drawing a bar chart

> IMPORTANT: Start an http server for this lab's directory. From command line call `python -m SimpleHTTPServer` (for Python 2) or `python -m http.server` (for Python 3).

During today's activities you will be working with the `letters_freq.csv` dataset again. We used this dataset way back in Lab 0. The dataset includes 26 rows. Each row corresponds to a letter in the English alphabet and the frequency that it is used. Here is a snippet of the data table:

letter	frequency
A	0.08167
B	0.01492
C	0.02782
D	0.04253

You will be working toward creating the following interactive bar chart today:



1. How to structure your code for interaction

▸ Pages 13

Lab 0: HTML & CSS
Lab 1: Javascript 101
Lab 2: SVG
Lab 3: Intro to D3 (Pre Lab)
Lab 3: Intro to D3 (Activities)
Lab 4: D3 Chart Types & Scales (Pre Lab)
Lab 4: D3 Chart Types & Scales (Activities)
Lab 5: D3 Selections & Grouping (Pre-Lab)
Lab 5: D3 Selections & Grouping (Activities)
Lab 6: D3 Enter, Update & Exit (Pre-Lab)
Lab 6: D3 Enter, Update & Exit (Activities)
Lab 7: Interaction & Transition 1
Lab 8: Interaction & Transition 2
Lab 9: D3 Layouts
Lab 10: D3 Maps

Clone this wiki locally

https://github.gatech.edu

Similar to the previous week, we have already added structure to your `letter_frequency\main.js` code. There are a number of additions to the code:

- `onCategoryChanged` method - used to handle `change` events from the `select` input widget.
- Layout parameters for configuring the spacing of your bar chart. `barBand` can be used to space out your bars evenly. * `chartG` is a group that has been positioned based on the `padding` . Add your bars to this group with `chartG.selectAll('.bar').data()`
- `d3.csv(fileName, processingMethod, dataCallback)` is included, notice that the `processingMethod` is used as a way to convert the feature of your dataset to integer such as `frequency`
- `updateChart(filterKey)` the method to be called for new data

Take some time to look through the template and read the comments.

2. Create a global data variable

Because you will need to access the loaded dataset in the `updateChart` method, you will need to create a global variable of the loaded dataset. Add the following declaration within the `function(error, dataset) {...}` data callback method:

```
    letters = dataset;
```

3. Create a width scale

Create a linear-scale for the `frequency` data attribute. You will want the output range of this scale to be `[0, chartWidth]` . **Again you might want this to be a global variable.**

4. Create two x-axis

Create two x-axis by appending each `<g>` element of x axis (top and bottom) to `svg` with `ticks` = 6 and use `tickFormat` to format it to percentage form **(We suggest to create a variable that loop through your dataset and format it to percentage).**

5. Create x-label

Create x-label **Letter Frequency(%)** by appending text to your `svg`

Your web page should look like this now:

Activity Result

Activity 1 - Updating with new data

In this final activity you will make your bar chart interactive! You will need to use the **Enter, Update, Exit pattern** to achieve this. You will need to reformat your code in `updateChart` in the following ways:

1. Create an update selection

You will first want to create an update selection of all the bars. Remember to use the key function:

```
var bars = chartG.selectAll('.bar')
    .data(filteredLetters, function(d){
        return d.letter;
    });
```

2. Enter and append all new elements

Use your code from the previous activity to create the bars for each letter. You may need to reformat it as needed:

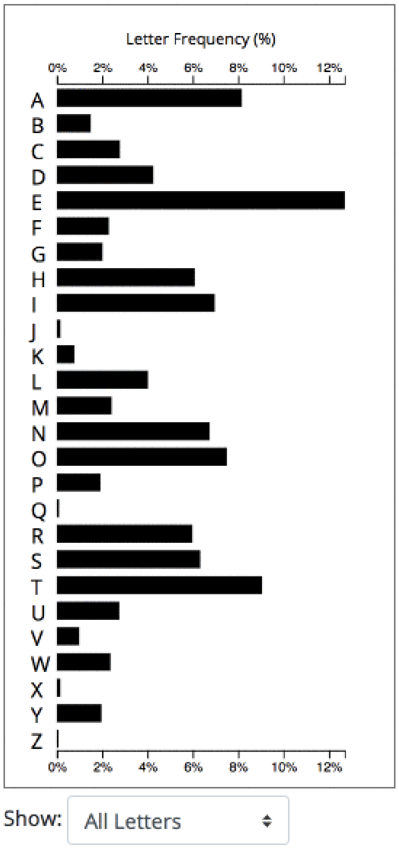
```
var barsEnter = bars.enter()
    .append('g')
    .attr('class', 'bar');

bars.merge(barsEnter)
        .attr('transform', function(d,i){
        return 'translate('+[0, i * barBand + 4]+')';
    });

barsEnter.append('rect')
    .attr('height', barHeight)
    .attr('width', function(d){
        return xScale(d.frequency);
    });

barsEnter.append('text')
    .attr('x', -20)
    .attr('dy', '0.9em')
    .text(function(d){
        return d.letter;
    });
```

3. Exit and remove filtered bars

The final step is to use the `exit` selection to remove the bars that have been filtered out by the data. Because we used the *key function*, the `.bar` elements will be removed if that letter is no longer present in `filteredLetters` :

```
bars.exit().remove();
```

With these steps we should get the following result:



Note: You must be wondering why is "Y" a Vowel? In fact, it is a hybrid! Feel free to read here if you are interested.

Submission

- Please change the text of `<p>` in index.html to your name.
- **Take screenshots of both, the code and the output (browser) of Activity 0 (1 screenshot) and Activity 1 (3 screenshots - All Letters, Only Consonants, Only Vowels) to receive full credit.**
- Please remember to submit the screenshot before 10:05 AM of lab day.

Congratulations, you have now finished Lab 6 and created an interactive bar chart. Next lab we will cover interactions in more depth.

This lab was based on the following material:

- Hanspeter Pfister's CS171 Lab Material (Harvard)
- D3 - Interactive Data Visualization for the Web by Scott Murray
- Enter, Update, Exit by Christian Behrens
- Three Little Circles by Mike Bostock
- Object Constancy by Mike Bostock

Home