



# Lab 3: Intro to D3

Jiang, Lei edited this page 3 hours ago · 5 revisions

## Prerequisites

- For tutorials and readings, please see [Lab 3 Pre-lab](#)
- Update your *starter code repository* using one of the following methods:
  - From the GitHub Desktop app click `Sync` on the top right
  - Open a command line prompt. Navigate to the repository directory, for example `cd ~\Development\CS4460-Spring2018\Labs` and run command `git pull`.

## Submission

- For Activity 0, take a screenshot of the page that includes the paragraphs or table generated by d3 and your name.
- For Activity 1, take a screenshot of the page that includes the scatterplot and your name
- Please remember to submit these screenshots before 10:05 AM of lab day (Monday 2/19).

## Activity 0 - Creating a list of data

Start an http server for this lab's directory. You can accomplish this by opening a command line window, navigating to this lab's directory (e.g. `cd ~/development/CS4460-Spring2018/Labs/03_lab/`). From there you will start the http server by executing: `python -m SimpleHTTPServer` (for Python 2) or `python -m http.server` (for Python 3). You will need to start an http server for every lab from here on out. Servers are required to serve local files and run JavaScript.

Similar to the pre lab, you will be creating a list of paragraph elements in your web page. As always, please start by opening your code editor to `03_lab/01_paragraphs/`. In that directory you will see a familiar project structure with `index.html`, `style.css`, and `paragraphs.js` files. `paragraphs.js` is where you will write JavaScript code for this activity. Also notice that there is a CSV data file in the directory: `baseball_hr_leaders_2017.csv`.

The CSV dataset is 2017's top 10 Home Run (HR) leaders from the MLB. See below for a snippet of the dataset:

name	rank	year	homeruns
Giancarlo Stanton	1	2017	57
Aaron Judge	2	2017	48
J.D. Martinez	3	2017	43
Khris Davis	4	2017	41

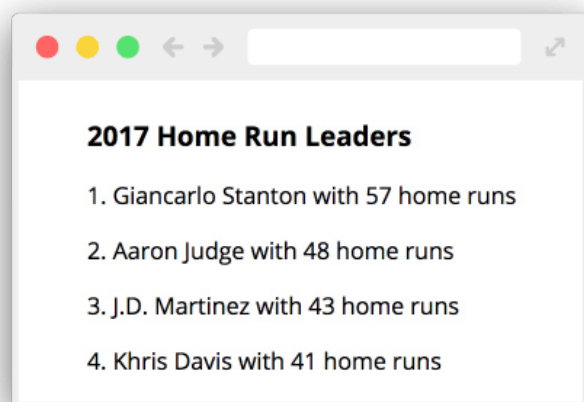
### Pages 7

[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](#)[Lab 5: D3 Selections & Grouping](#)[Lab 6: D3 Enter, Update & Exit](#)[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>

For this activity you will use d3 data binding to create a paragraph ( `<p>` ) for each player in the list. You will also add text content to each paragraph to describe the player's Home Runs for this season and their rank. In the end your web page should look like this:



You will edit the `01_paragraphs/paragraphs.js` file for this activity.

#### 1. Load the CSV file

Using `d3.csv()`, load the dataset at `./baseball_hr_leaders_2017.csv`. Make sure to write a callback function to access the loaded data.

We recommend using `console.log()` on the loaded data to double check everything went according to plan.

#### 2. Create your paragraphs

Append a `p` element to the `#homerun-leaders` div for each player in the dataset. You will need to follow the above examples to achieve this.

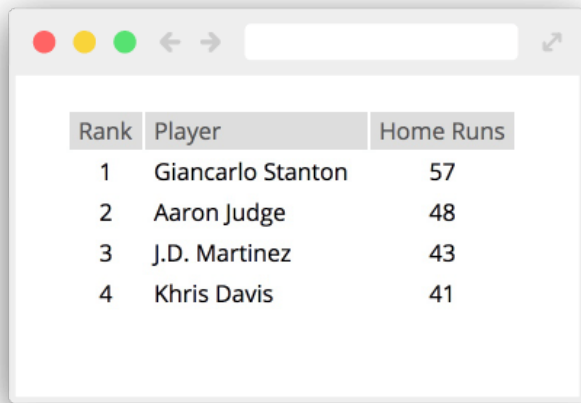
#### 3. Add text content

Add text content using `text()` for your D3 data selection. The text content should include the player's HR rank, their name, and the number of Home Runs they hit this season. Again, you can follow the examples above to add text content.

#### 4. Highlight the Home Run King

Add a bold font styling to the player that has the most Home Runs this season, Giancarlo Stanton. There are a number of different ways to select and style this `p` element based on the rank value, index of the player JS object, or the player's name.

**Challenge 1:** Create a table of the dataset like in the below snippet:



Rank	Player	Home Runs
1	Giancarlo Stanton	57
2	Aaron Judge	48
3	J.D. Martinez	43
4	Khris Davis	41

We have provided the following table HTML to get you started:

```
<table id="homerun-table">
  <thead>
    <tr>
      <td>Rank</td>
      <td>Player</td>
      <td>Home Runs</td>
    </tr>
  </thead>
  <tbody>
    </tbody>
</table>
```

You will need to append a table row `<tr>` for each player to the `<tbody>`. You will then need to add a `<td>` cell element for each column.

Style your table so that readers can differentiate the column headers from the column data content.

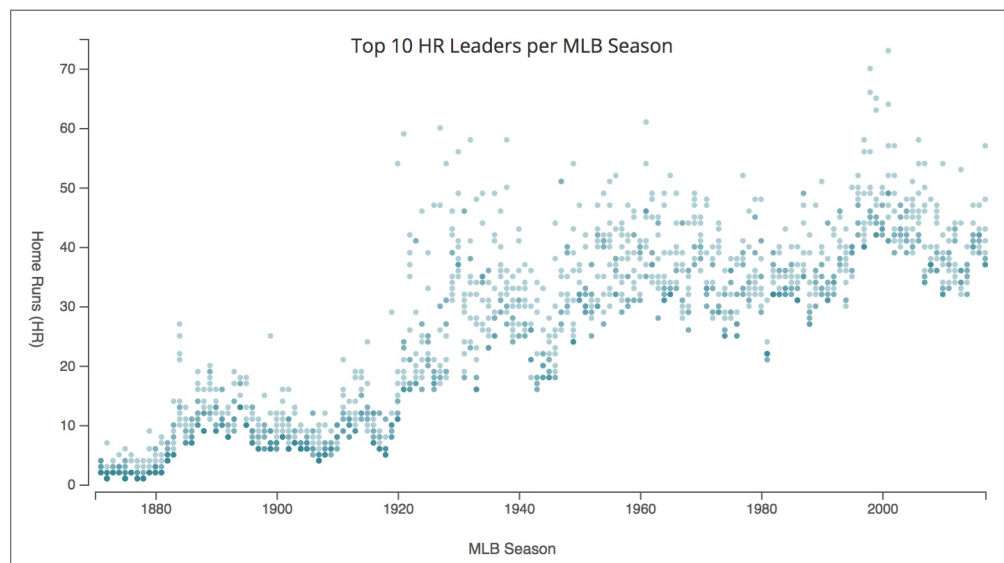
## Activity 1 - Drawing a pixel scatterplot

The starter code for this activity can be found at `03_lab/02_scatterplot/`.

We have learned from our lectures that a pixel chart is a great way to visualize data when there is a lot of data. Pixel charts are also effective when the pixel represents one data case.

In this exercise you will create a pixel scatterplot for the expanded Home Run leaders dataset. The expanded dataset includes all of the historical top 10 Home Run leaders. The dataset spans from 1871 to the current MLB season. In total there are 1,679 data cases in the `baseball_hr_leaders.csv` dataset.

You will create the following scatterplot during this activity:



You will edit the `02_scatterplot/scatterplot.js` file for this activity. We have already added code to create the axes, scales and labels for the scatterplot. You will be calling the following two functions: `scaleYear(year)` and `scaleHomeruns(homeruns)`. Each function returns a scaled pixel value for the input numeric data value.

### 1. Load the CSV file

Load the dataset at `./baseball_hr_leaders.csv` using `d3.csv()` as you did in the previous activity. Again be sure to include all code that requires the loaded dataset within the call-back function.

### 2. Create and center your circles

Create a `circle` element for each data case by: Select all `circle` elements in the svg, then create a data-binding with the dataset, and finally enter/append `circle` elements.

Set the center point of each circle by data.

- The `cx` position should be set to the `year` data attribute for each circle (use the provided `scaleYear(year)` method).
- The `cy` position should be set to the `homeruns` data attribute (use the provided `scaleHomeruns(homeruns)` method).

Set the radius of each circle to all be `2px`.

### 3. Style your circles

You should now see all of the black circles placed in the chart. You cannot see it because of the black-opaque fill of the circles, but a lot of home run leaders are on top of each other - this is due to players being tied for Home Runs in the same year. To fix this we ask that you style the `fill` and `opacity` of each circle, so that we can see where circles occlude each other.

Hint, we recommend using the `style.css` file to style all circles. In-line styling is better reserved for data-bound styling.

**Challenge 1:** Create a text label for each player.

In this challenge you will need to append nested SVG elements. You will want to append a `<g>` element for each data case. You will want to translate that group based on the `year` and `homeruns` data attributes as you did earlier for the circles.

Then, you will want to append a `<text>` and `<circle>` element to each `<g>`. **Note, you do not need to set the x or y positions of the text or circle because of the transform of the enclosing `<g>` element.** (Remember last weeks section on SVG transform attributes).

When you achieve this you will see a ton of text elements! Try the next challenge to only show one at a time.

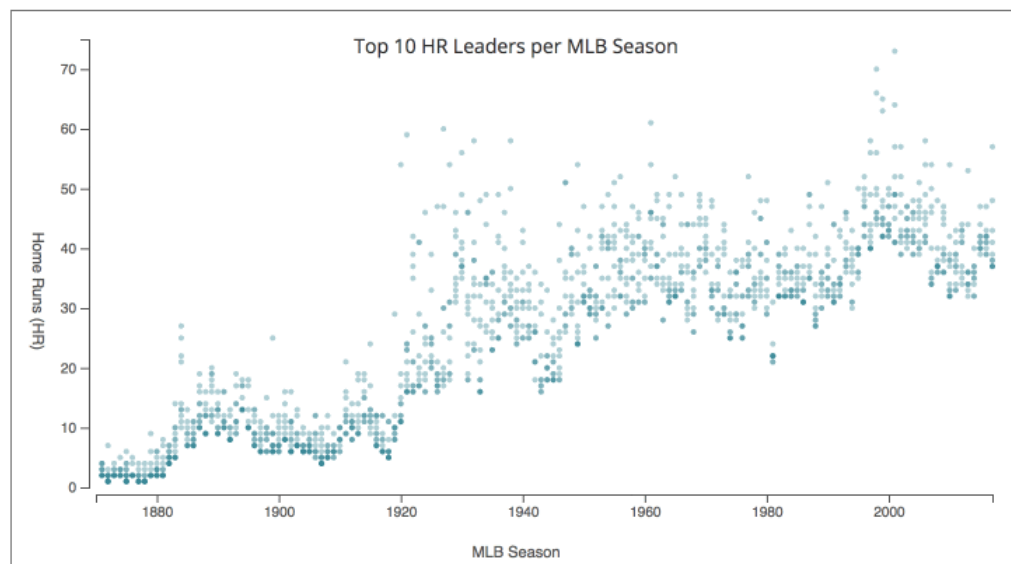
### Challenge 2: On hover show the circle's text label.

To create this effect, you will want to set all text labels to have an 'opacity' of 0 to start. We recommend using the `style.css` file to make a CSS rule.

Next you will create another CSS rule for the text labels, however you will add the `:hover` CSS selector to the group element (e.g. `g:hover text`).

What this does is styles the text elements when the parent `g` element is mouse-hovered. As you may have guessed, within this rule you want to set the text `opacity` to be 1. This makes the text visible on mouse-hover.

See below for how the hover interaction looks:



Congratulations, you have now finished Lab 3 and created a visualization with D3. Next lab we will show how to use D3's axes and scales.

### This lab is based on the following material:

- Hanspeter Pfister's CS171 Lab Material (Harvard)
- [DOM Manipulation and D3 Fundamentals](#) by Alex Lex of U. of Utah
- [Practical applications of a d3js selection](#)
- [D3 - Interactive Data Visualization for the Web](#) by Scott Murray

[Home](#)



