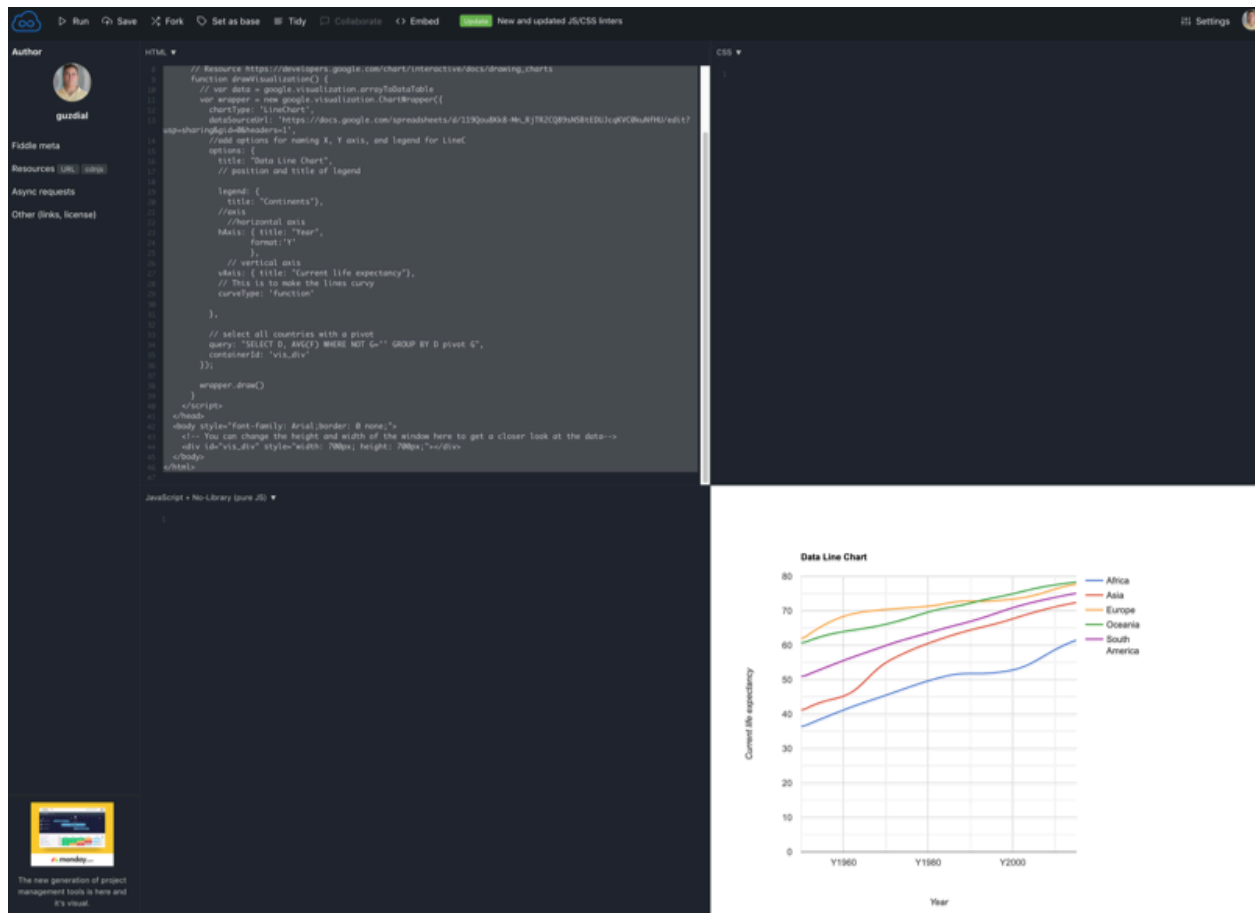


Activity: JavaScript

In this activity, we will be using [JavaScript](#), a standard Web programming language. JavaScript will be inside an HTML page, just like most pages on the Internet. (See [the Wikipedia page on HTML](#).) Our program will read in data from a Google Sheet, and will display the visualization using Google's Chart service ([see explanation and examples here](#)).

First, please take this quick overview of the data set that we will be using -- [click here](#). The data set itself can be found [in this Google Sheet](#). You will not be working with the Google Sheet directly. Instead, you will be modifying a program in JavaScript that will do it for you.

Now, [please click on this link](#). It will take you to JSFiddle, a website that lets you work with JavaScript and HTML. You will see some text code in the upper left hand corner, some control buttons on the top, and a visualization in the lower right hand corner. It should look something like this:

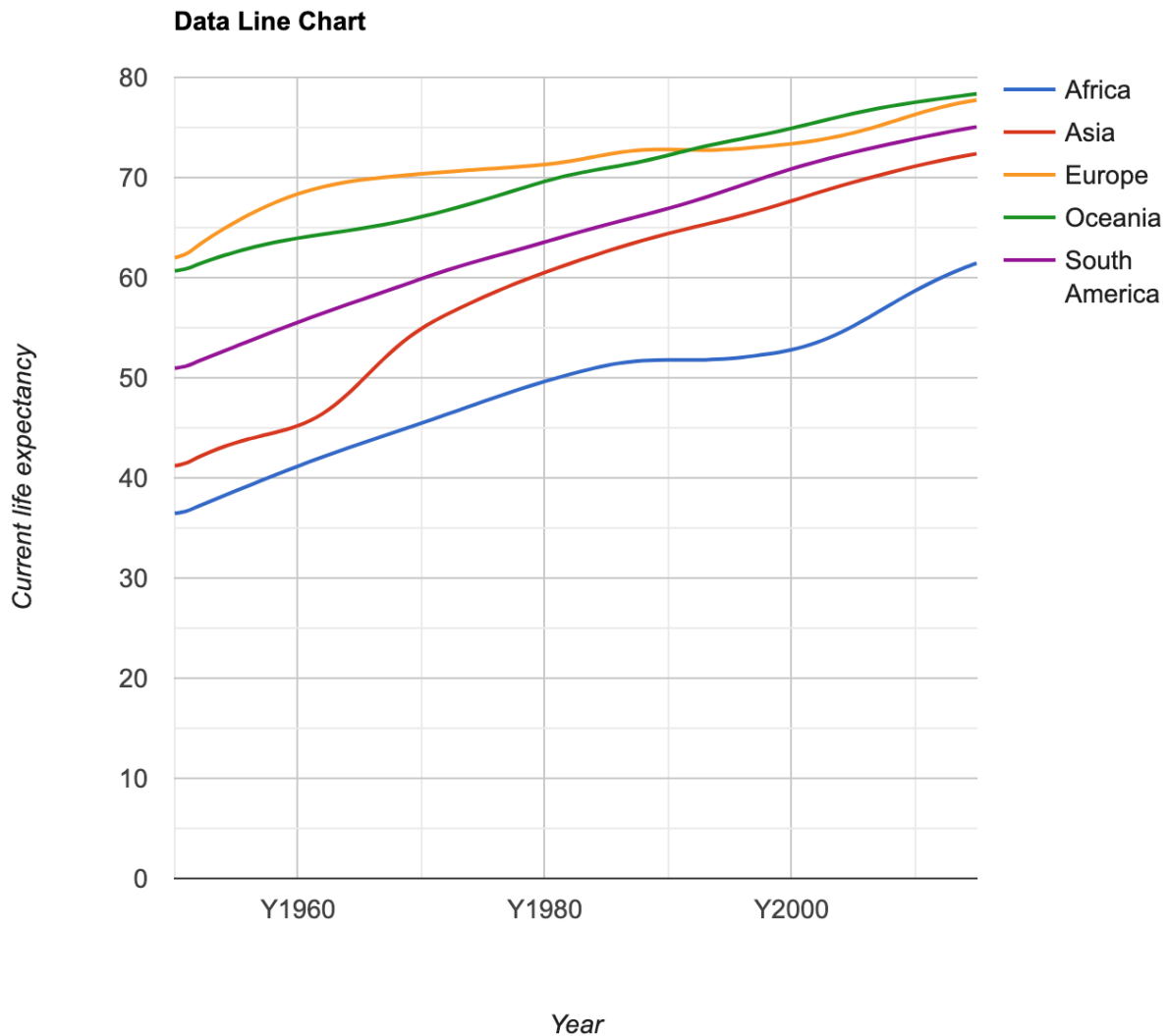


Click the "Run" button to run the JavaScript program again:

▶ Run

You will need to do this after we make changes to the program to get a different visualization.

What this Visualization Shows Us



This visualization shows us how expected life expectancy has increased on all continents, but is different between continents. Life expectancy in Europe has always been well above life expectancy in Africa, for example.

Making Changes

Let's make changes to your visualization, by changing the program.

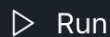
First, click the Fork button along the top row



. This gives you your own copy of the program, different from anyone else's.

1. Easiest change: In line 11, change the word **LineChart** to **ScatterChart**. It should look like this:

```
11      chartType: 'ScatterChart',  
12      dataSourceUrl: 'https://docs.g
```



Press the Run button.

This will generate a scatter chart instead of a line chart.

2. Changing what data we see: Currently, the visualization is showing measured (current) life expectancy. To get the predicted, future life expectancy, we need to access different data in the data set.

We specify which data we want from the data set using the *query* on line #33 in the JavaScript program. Currently, it says:

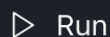
```
SELECT D, AVG(F) WHERE NOT G="" GROUP BY D pivot G
```

This is an industry-standard notation called SQL (often pronounce “sequel” -- see [Wikipedia page here](#)). It says to return the year (column D) and the average of values in column F (current life expectancy) where the continent (G) is not empty, group by year (D), and *pivot* by continent. The “pivot” organizes the data around continents.

To change the visualization to show the future life expectancy instead of currently, simply change the **F** to column **E**, like this:

```
SELECT D, AVG(E) WHERE NOT G="" GROUP BY D pivot G
```

```
32      // select all countries with a pivot  
33      query: "SELECT D, AVG(E) WHERE NOT G='' GROUP BY D pivot G",  
34      containerId: 'vis_div'
```



Press the Run button.

Optional: If you want to fix the vertical axis label to match the data you're seeing, change line #26. From:

```
vAxis: { title: "Current life expectancy"},
```

To:

```
vAxis: { title: "Future life expectancy"}},
```

▶ Run

Press the Run button.

3. Focusing on certain data: One last change we will make is to look at only a single country. Change the query on line #33 so specify the country, which is in column B. For example, changing that line to

```
query: "SELECT D, AVG(E) WHERE B = 'Afghanistan' GROUP BY D pivot G",
```

Allows us to see a graph of just Afghanistan's future life expectancy (column E).

▶ Run

Press the Run button.

Can you show a visualization of the measured life expectancy (column F instead of E) for just the country France?

How about for Cuba?

How the program works

```
1 <html>
2 <head>
3   <script type="text/javascript"
4     src="https://www.gstatic.com/charts/loader.js"></script>
5   <script type="text/javascript">
6     google.charts.load('current'); // Don't need to specify chart libraries!
7     google.charts.setOnLoadCallback(drawVisualization);
8
9     // Resource https://developers.google.com/chart/interactive/docs/drawing_charts
10    function drawVisualization() {
11      var wrapper = new google.visualization.ChartWrapper({
12        chartType: 'LineChart',
13        dataSourceUrl:
14        "https://docs.google.com/spreadsheets/d/119Qou8Kk8-Mn_RjTR2CQ89uNS8tEDUJcQKVC8uWfHU/edit?usp=sharing&id=8dheaders=1",
15        //add options for naming X, Y axis, and legend for LineC
16        options: {
17          title: "Data Line Chart",
18          // position and title of legend
19          legend: {
20            titles: "Continents",
21            //horizontal axis
22            hAxis: { title: "Year",
23                    format: "y"
24                  },
25          // vertical axis
26          vAxis: { title: "Current life expectancy",
27                  // This is to make the lines curvy
28                  curveType: "function"
29                },
30        },
31        // select all countries with a pivot
32        query: "SELECT D, AVG(F) WHERE NOT G=' ' GROUP BY D pivot G",
33        containerId: "vis_div"
34      });
35
36      wrapper.draw()
37    }
38  </script>
39 </head>
40 <body style="font-family: Arial;border: 0 none;">
41   <!-- You can change the height and width of the window here to get a closer look at
42     the data-->
43   <div id="vis_div" style="width: 700px; height: 700px;"></div>
44 </body>
45 </html>
```

This part that starts with <html> is common to all Web pages

The part that starts with <script> is the program that creates the visualization.

This is the part that defines the kind of visualization we are making

Here is where we gather data from our Google Sheet

This section specifies formatting for our visualization

The "query" is an SQL statement that describes what data to take from the sheet. The letters D, F, and G refer to the columns year, current_exp, and Continent from our data set

Now that we have specified the drawing, actually draw it.

Finally, here is some HTML that specifies where the visualization appears

Something else to try

Google Charts supports a wide variety of visualizations. If you are interested, try the example at this JSFiddle ([click here](#)). This presents the current life expectancy data on a map of the world.

