Let’s make a button

Buttons are HTML elements, and get made using an HTML tag:

<button type="button"> test button </button>

That makes a button, but doesn’t do anything when you click it. For behavior, add Javascript. Tell the button to look for a Javascript function when you click it:

<button type="button" **onclick = “ buttonClicked ( )“** >
test button</button>

Now, make a function in Javascript that will do something:

```
function buttonClicked ( ) {
        //do something
        console.log('here');
}
```

Parsing multi-year data in .csv

Read in the .csv

Filter the array using the relevant properties.

```
filteredArray = dataIn .filter (  function ( d ) {
      return d.year  ==  2016;
});
```

Store as separate variables for later use

Create DOM elements for the data (but don't draw it yet!)

Make a d3 selection

Bind the data to it

Enter( ) DOM elements

Append circles

Add a class

Call a draw function that will be used many times

Making a draw function

Should not append elements here! (Otherwise, you add new circles every time you call it)

Instead, select all of the elements you created in the previous slide using the class that you gave them, re-bind the data, and then draw the circles.

```
function drawPoints ( pointData ) {

        svg.selectAll ('.dataPoints')
              .data ( pointData )
              .attr ('cx', function ( d ) {
                        return +d.x;
             } )…
```

Connect updates to button clicks

Add drawPoints( ) function call to the HTML button click event listener

Hand drawPoints different data, depending on the button click state

```
function buttonClicked ( ) {

        if ( clicked == true ) {
                drawPoints ( data2016 );
                clicked = false;
         }
        else {
                drawPoints ( data2017 ) ;
                clicked = true;
         }

}
```

Set up scales for the data

Use linear scales for x and y. Use the max value from the spreadsheet to determine the top of the domain.

Adjust drawing position

Make the SVG 800 px wide and 500 px tall

Add a group to hold all of the svg elements

Translate the group 100 px over and 100 px down, so that it doesn't hit the edges of the SVG container

```
var svg  = d3.select ( 'svg' )
             .append ( 'g' )
             .attr ( 'transform' , 'translate ( 100,100 )' );
```

Add axes

Draw x and y axes using d3

```
svg .append ("g")
      .attr ( 'transform‘ , 'translate ( 0 , 400 )‘ )
      .call ( d3.axisBottom ( scaleX ) );


svg .append ("g")
      .call ( d3.axisLeft ( scaleY ) );
```

Add axis titles

Draw chart and axis titles

```
svg .append ( 'text‘ )
      .text ( ‘My updating chart‘ )
      .attr ( 'transform','translate (300, -20)‘ )
      .style ( 'text-anchor‘ , 'middle‘ );

 svg .append ( 'text‘ )
      .text ( ‘some x axis value‘ )
      .attr ( 'transform‘ , 'translate ( 260, 440 )‘ );

svg .append ( 'text‘ )
      .text ( ‘some y axis value‘ )
      .attr ( 'transform', 'translate ( -50 , 250 ) rotate ( 270 ) ‘ );
```

Auto-advancing

Sometimes, you don't want to wait for the user to click a button to change the data.

If you have data that you want to autoadvance, you can use a built-in function called setInterval to call the update function

```
window.setInterval ( function ( )
   {
      buttonClicked ( );

}, 1000);
```

See usage in
https://bl.ocks.org/avrasgoldman/1c0c3e883dc03be1d8691bce73c6330f

https://www.nytimes.com/interactive/2017/07/28/climate/more-frequent-extreme-summer-heat.html

Updates with transitions

You can use transitions to help the user follow a change from one screen to the next.

```
function drawPoints ( pointData ) {

        svg.selectAll ('.dataPoints')
              .data ( pointData )
              .transition ( )
             .ease ( d3.easeSin )
             .duration ( 400 )
              .attr ('cx', function ( d ) {
                        return +d.x;
             } )…
```

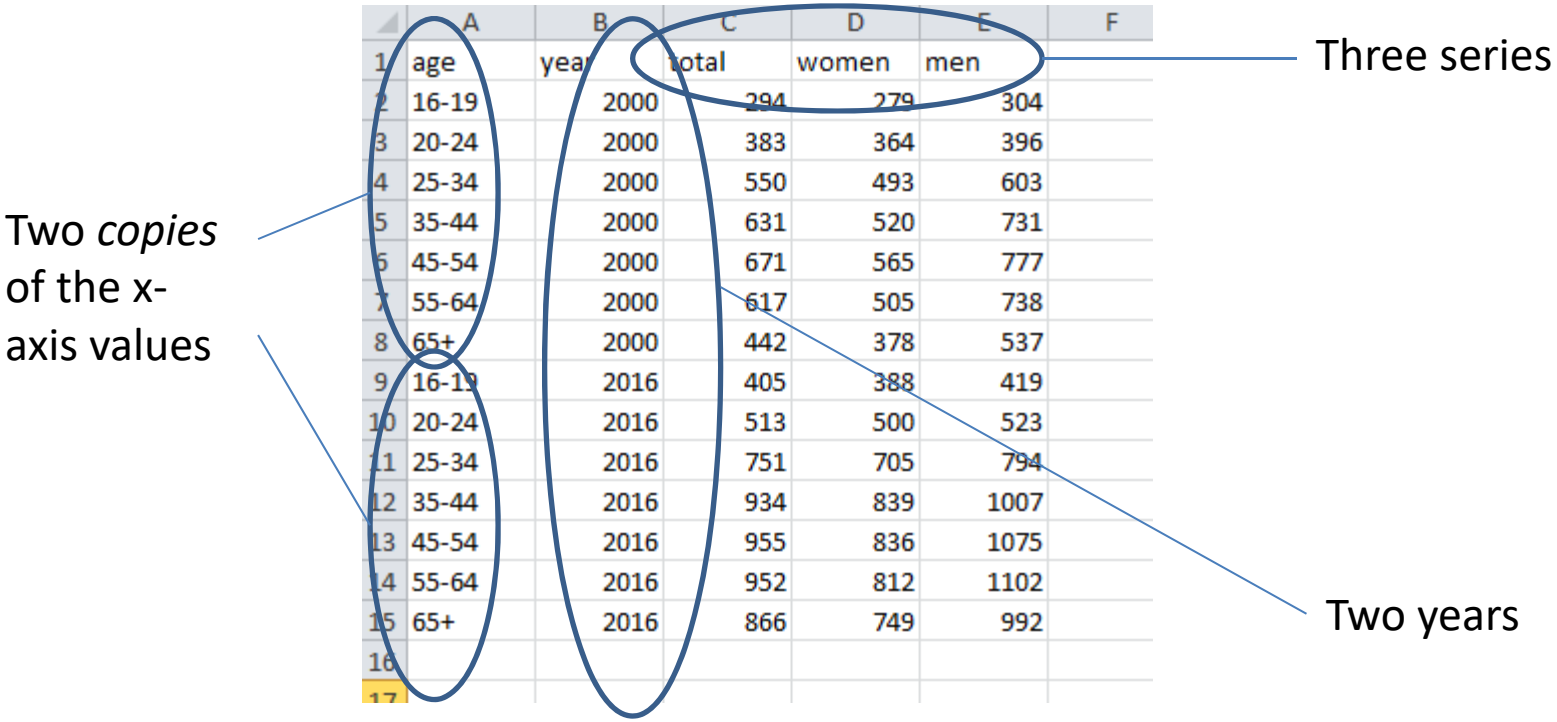
Why do the dots fly in from the origin?

You haven't set coordinates before drawing. If this is not a desired feature, duplicate the positioning code immediately after appending the circle (and before drawing)

CSV file structuring

For this week's homework: Choose one of your final project datasets that has at least two series, and plot a bubble or a bar chart using one series of the data

Notice how I formatted my .csv file: use the same structure in yours!



	A	B	C	D	E	F
1	age	year	total	women	men	
2	16-19	2000	294	279	304	
3	20-24	2000	383	364	396	
4	25-34	2000	550	493	603	
5	35-44	2000	631	520	731	
6	45-54	2000	671	565	777	
7	55-64	2000	617	505	738	
8	65+	2000	442	378	537	
9	16-19	2016	405	388	419	
10	20-24	2016	513	500	523	
11	25-34	2016	751	705	794	
12	35-44	2016	934	839	1007	
13	45-54	2016	955	836	1075	
14	55-64	2016	952	812	1102	
15	65+	2016	866	749	992	
16						