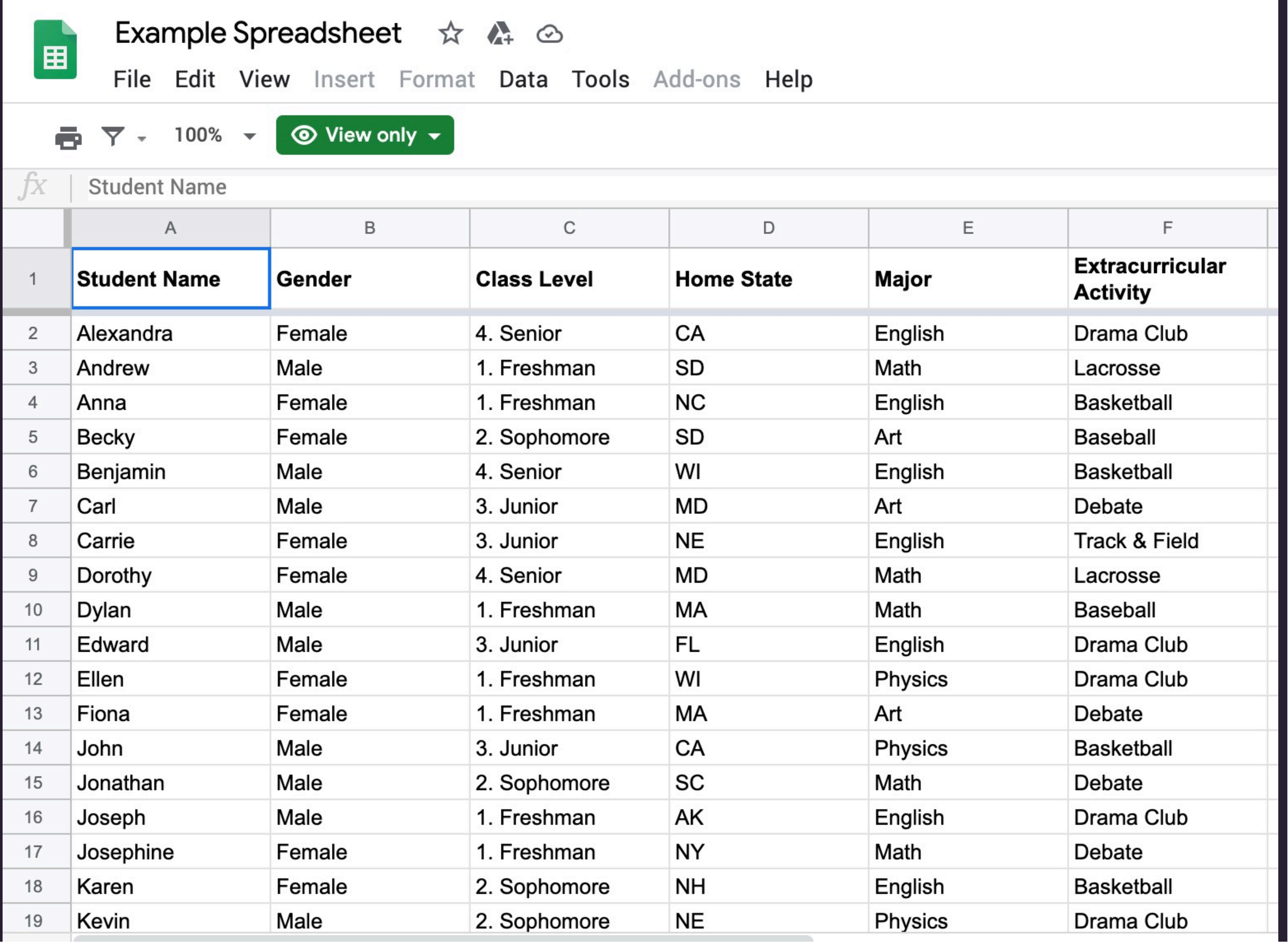
See What I Mean?

Data Visualization in WordPress

Once upon a time...

Sam from the admissions office wants to show a chart of how many students from each class are returning to campus this fall.

Sam gives us... a giant spreadsheet.



	A	B	C	D	E	F
1	**Student Name**	**Gender**	**Class Level**	**Home State**	**Major**	**Extracurricular Activity**
2	Alexandra	Female	4. Senior	CA	English	Drama Club
3	Andrew	Male	1. Freshman	SD	Math	Lacrosse
4	Anna	Female	1. Freshman	NC	English	Basketball
5	Becky	Female	2. Sophomore	SD	Art	Baseball
6	Benjamin	Male	4. Senior	WI	English	Basketball
7	Carl	Male	3. Junior	MD	Art	Debate
8	Carrie	Female	3. Junior	NE	English	Track & Field
9	Dorothy	Female	4. Senior	MD	Math	Lacrosse
10	Dylan	Male	1. Freshman	MA	Math	Baseball
11	Edward	Male	3. Junior	FL	English	Drama Club
12	Ellen	Female	1. Freshman	WI	Physics	Drama Club
13	Fiona	Female	1. Freshman	MA	Art	Debate
14	John	Male	3. Junior	CA	Physics	Basketball
15	Jonathan	Male	2. Sophomore	SC	Math	Debate
16	Joseph	Male	1. Freshman	AK	English	Drama Club
17	Josephine	Female	1. Freshman	NY	Math	Debate
18	Karen	Female	2. Sophomore	NH	English	Basketball
19	Kevin	Male	2. Sophomore	NE	Physics	Drama Club

What is our quest?

Create a custom Gutenberg block to display this chart.

Challenge accepted

1. Import the data into WordPress.
2. Process the data.
3. Make an accessible and responsive graph.

Chapter 1

Import the data into WordPress.

Step 1

Have your block store the URL of your Google sheet.

The `edit()` function should render this:

```
<TextControl
  label='Google Sheets URL'
  help='(Must be publicly viewable.)'
  value={ sheetUrl }
  onChange={ onChangeUrl }
/>
```

Step 2

Extract the data.

- This is a **dynamic** block!
- We need PHP to extract and process the data.
- Function called by the render callback.

Call the Google API and get the data.

```
   $get_data = new WP_Http();
   $url = 'https://sheets.googleapis.com/v4/spreadsheets/';
   $url .= $sheet_id;
   $url .= '/values/' . $range;
   $url .= '/?&key=' . $api_key;

   return $get_data->get( $url );
}
```

Chapter 2

Process the data.

Step 1

Use `json_decode()` to convert the data.

```
Array(
  [values] => Array(
    [0] => Array(
      [0] => 1. Freshman
    )

    [1] => Array(
      [0] => 4. Senior
    )
    ...
)
```

Step 2

Remember our problem: We need to count the number of students from each major.

```
$data = array();
foreach ( $data_body['values'] as $d ) {
  if ( array_key_exists( $d[0], $data ) ) {
    // If the value already exists
    $data[ $d[0] ]++;
  } else {
    // Otherwise, create new item
    $data[ $d[0] ] = 1;
  }
}
```

Now we have an array that looks something like this:

```
Array(
  ['1. Freshman'] => '8',
  ['2. Sophomore'] => '8',
  ['3. Junior'] => '12',
  ['4. Senior'] => '8'
)
```

Chapter 3

Make an accessible and responsive graph.

Grade levels

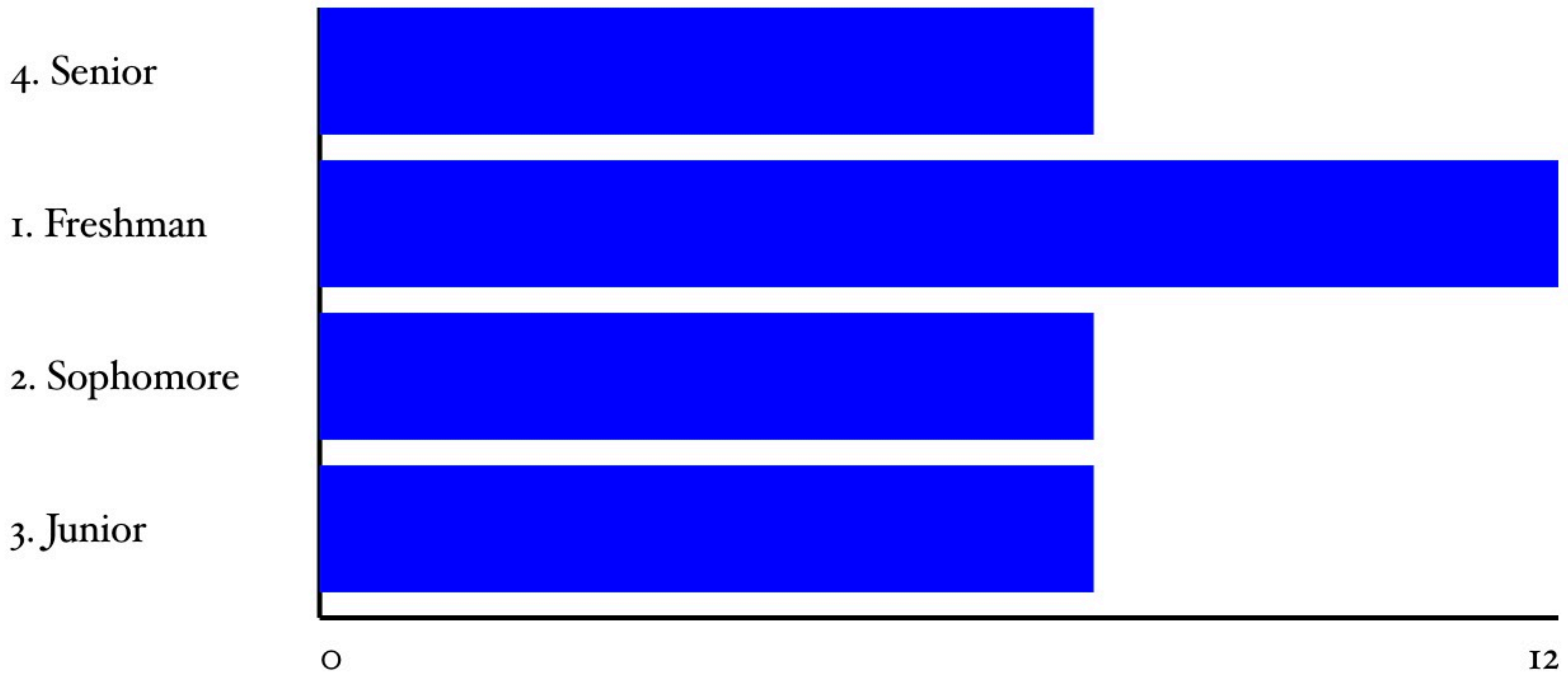


Chart of the grade levels of everyone in our school.

Step 1

Let's set up our SVG.

SVG overview

```
<svg xmlns="http://www.w3.org/2000/svg"
    width="100%" height="SVG_HEIGHT">

  <title>My Chart</title>
  <desc>What my chart is about!</desc>

  <!-- Shapes go here! -->

</svg>
```

SVG height

The SVG needs to account for the height of the sum of the bars in the chart.

```
sizeof($data) * ( BAR_HEIGHT + BAR_GAP )
```

Step 2

Create the X and Y axes.

(Yup. Axes is the plural of "axis". Chop chop.)

Y axis

```
<line
   role="presentation"
   x1="OFFSET%" y1="0"
   x2="OFFSET%" y2="HEIGHT_IN_PX"
   stroke="#000" stroke-width="2" />
```

X axis

```
<line
   role="presentation"
   x1="OFFSET%" y1="HEIGHT_IN_PX"
   x2="100%"    y2="HEIGHT_IN_PX"
   stroke="#000" stroke-width="2" />
```

Step 3

Create the bars!

(a.k.a. the fun part.)

Start a group for all bars

```
<g role="list" aria-label="Bar graph">
  BARS GO HERE.
</g>
```

Single bar creation

- Loop through your array of data.
- Create a group for each bar, containing
    - The bar itself
    - The text label for that bar
    - (optional) Description

```
<g
  role="listitem" aria-label="LABEL, DATA"
  tabindex="0">
    <desc>
      The number of LABEL students
      returning is DATA
    </desc>
    BAR ELEMENT
    LABEL ELEMENT
</g>
```

Bar element

```
<rect
  role="presentation"
  x="OFFSET%"
  y="NUMBER_OF_BARS_SO_FAR * (BAR_HEIGHT + GAP)"
  width="THIS_BARS_WIDTH%"
  height="BAR_HEIGHT"
  fill="#00f" />
```

The bar’s width

The width of the current bar is the value of the bar (how many students in this class level) as a **percentage**.

```
VALUE / MAX_VALUE * 100
```

Bar label

```
<text
  role="presentation"
  x="0"
  y="NUMBER_OF_BARS_SO_FAR * (BAR_HEIGHT + GAP)"
  fill="#000"
  font-size="16">

  LABEL

</text>
```

All together now

```
<g role="list" aria-label="Bar graph">

  <g role="listitem" aria-label="LABEL, DATA" tabindex="0"
    <desc>Optional description for this bar</desc>
    <rect role="presentation" x="OFFSET%" y="NUMBER_OF_BAR
    <text role="presentation" x="0" y="NUMBER_OF_BARS_SO_F
  </g>

  ...
</g>
```

Grade levels

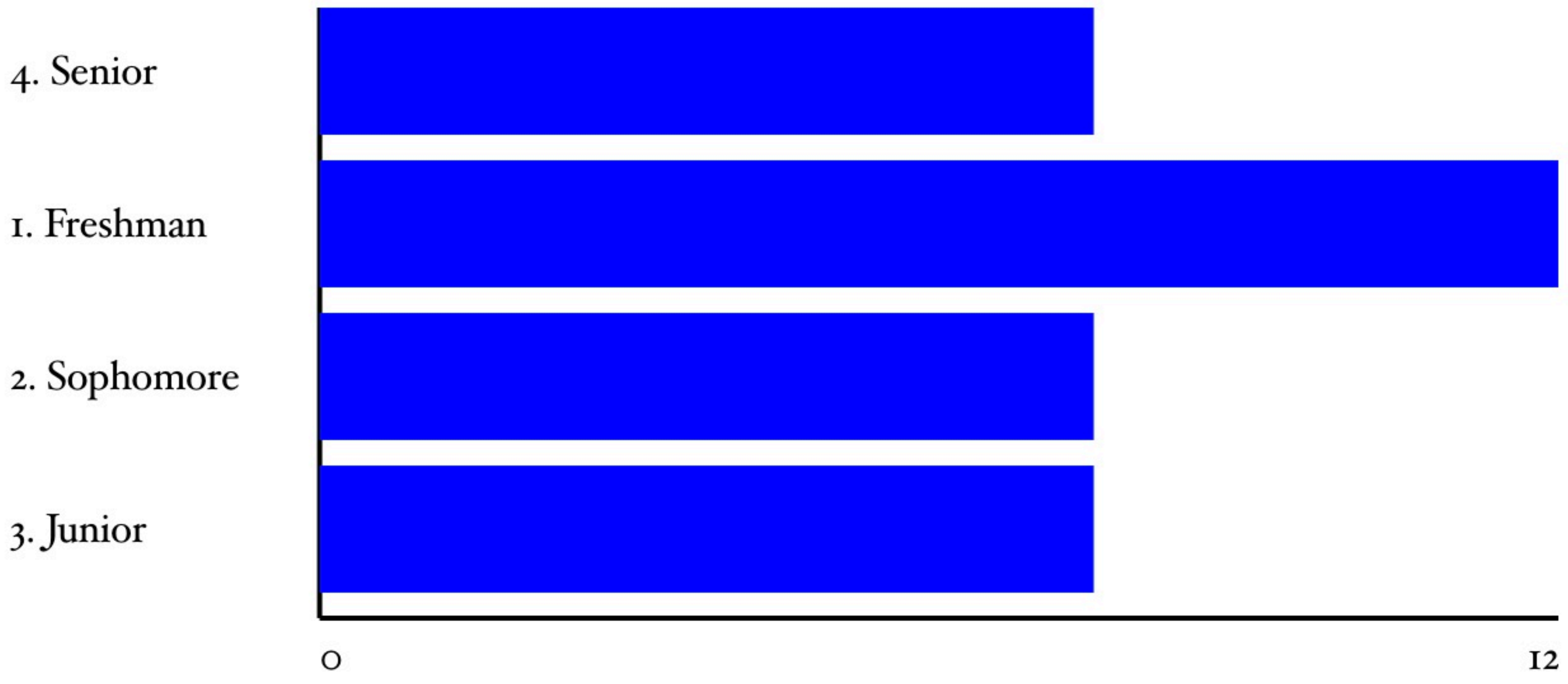


Chart of the grade levels of everyone in our school.

Thank you!!

https://talks.thatdevgirl.com/datavis-lightning/

- Follow me at @jonihalabi
- https://thatdevgirl.com
- https://jhalabi.com

Reference: General

- Besan Block (custom plugin; examples are from here)
- Example Google Sheet (public, view only)
- Longer data visualization talk slides

Reference: SVGs

- SVG Tutorial | W3Schools
- Tips for Creating Accessible SVG | Sitepoint
- Accessible SVGs | CSS-Tricks

Google API (1/2)

* To get this key, go to the [Google APIs Dashboard](https://console.developers.google.com/apis/dashboard). You should have a Google account to access this dashboard. * Inside the dashboard, go to "Select a Project" at the top of the page and click on "New Project". * Give your project a name and click the "Create" button. * From the [Library](https://console.developers.google.com/apis/library) page, search for the "Google Sheets API" and click the blue "Enable" button.

Google API (2/2)

* From the [Credentials](https://console.developers.google.com/apis/credentials) page, click "Create credentials" and select "API key" in the drop-down menu that appears. * A pop-up window with your API key will appear. Copy the key, then click "Restrict Key". * Under the "API restrictions" heading, check "Restrict Key", then select the "Google Sheets API" from the drop down menu. * Click "Save".

Without data you're just anothe
person with an opinion.

-- W. Edwards Deming