**What type of data is the data?**

(1): Quantitative Data
- Maps a continuous domain to a continuous range.
  - Linear: `d3.scaleLinear()`
  - Power/Log: `d3.scalePow()` or `d3.scaleLog()`

(2): Categorical Data
- Maps a discrete domain to continuous or discrete range.
  - Bands (ex: bar chart): `d3.scaleBand()`
  - Points (ex: scatter chart): `d3.scalePoint()`

**What is the domain (input) of the data?**

(1): Quantitative Data
- Data may be fixed, ex: `[0, 100]` for grades
- Data may be pre-processed (ex: minValue, maxValue)
- Data may be found in JavaScript when the data is an array of dictionaries (ex: find min/max of `"score"`):
  - `var min = _.minBy(data, "score")["score"]`;
  - `var max = _.maxBy(data, "score")["score"]`;
  - Domain: `[min, max]`

(2): Categorical Data
- Data may be fixed or pre-processed
- Data may be found in JavaScript when the data is an array of dictionaries (ex: find categories of `"className"`):
  - `var categories = _.map(data, "className")`;
  - `categories = _.uniq(categories)`;
  - Domain: `categories`

**What is the range (output) of the data?**

(1): Horizontal (x-axis)
- `.range([0, width])`

(2): Vertical (y-axis)
- `.range([0, height])`

---

**Create the Scale**

```javascript
var gpaScale = d3.scaleLinear()
  .domain([0, 4])
  .range([0, width]);
```

...what does this scale do?

```javascript
var opponents = _.map(data, "Opponent");
opponents = _.uniq(opponents);
var opponentScale = d3.scale()
  .domain(opponents)
  .range([0, height]);
```

...what does this scale do?

**Using the Scale**

With a scale function variable, you can translate data into a visual encoding:

```javascript
var gpaScale = 3.7; // returns (width * (3.7/4.0))
```

When using `scaleBand`, we have addition information on our scale:

```
bandwidth
0
1
2
```

We can find the width of each band in our scale:

```javascript
someBandScale.bandwidth(); // may return 100
```
Create the Axis

For any scale, we can create an axis on our visualization:

```
var axis = d3.axisTop()
    .scale( scaleName );

svg.append("g")
    .call(axis);
```

Visual Encodings

Example #1
Designed by Tana Wuren for Activity 4, heavily modified for simplicity

Example #2
Designed by Tianyu (Andy) Mao for Activity 4, modified for simplicity