

---

# **DDA2003/MDS6 I I 2**

# **Visual Analytics/Data Visualization**

## **Tutorial 7: Assignment 4**

## **High-Dimensional Data Visualization**

Yuxuan LIU

March 20<sup>th</sup>, 2023



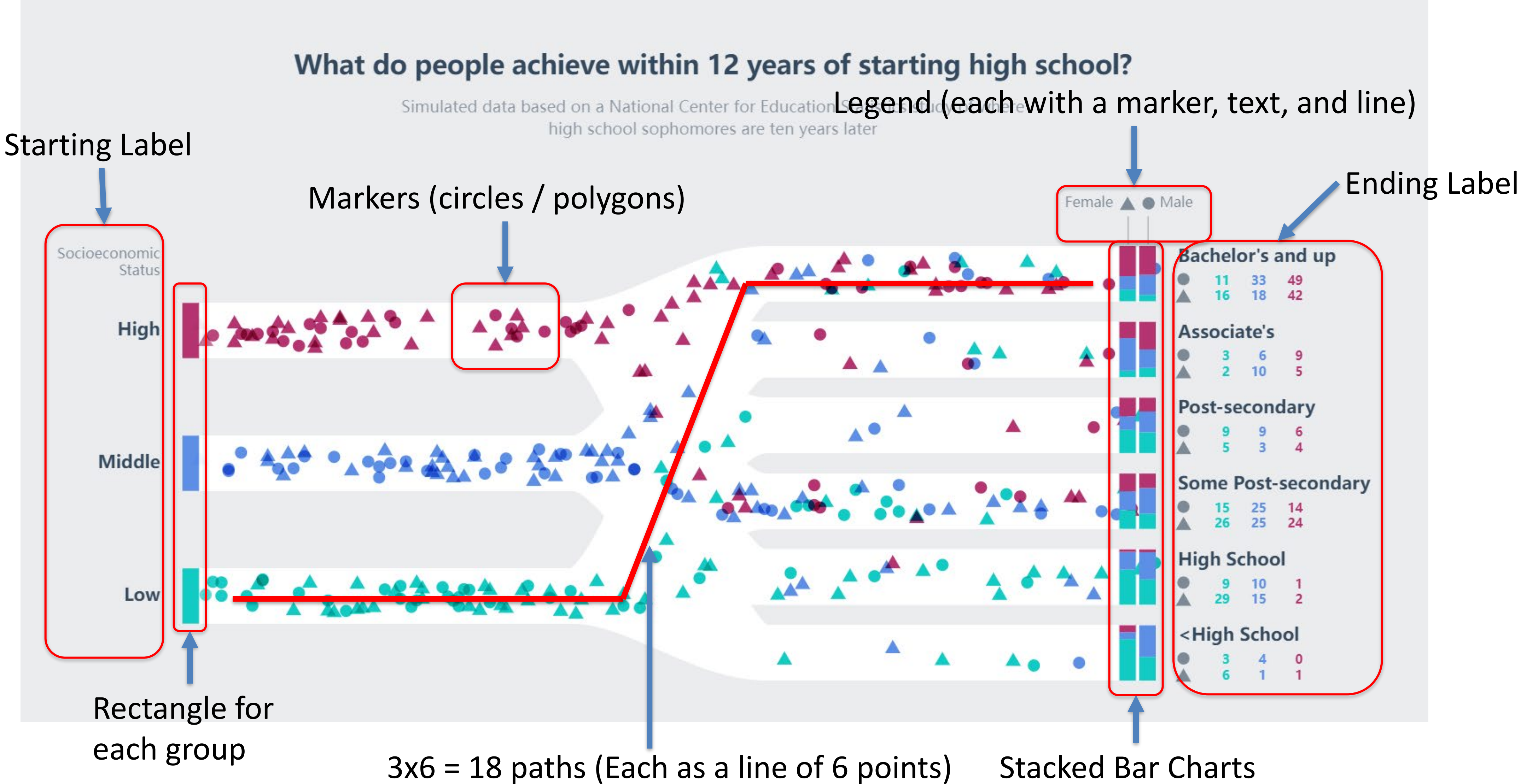


# Overview: 8 steps to do this assignment

---

- **Step-1:** Access data
- **Step-2:** Stack probabilities
- **Step-3:** Create people
- **Step-4:** Visualize paths
- **Step-5:** Visualize people
- **Step-6:** Add color and filter
- **Step-7:** Visualize stacked bar charts
- **Step-8:** Update counting numbers

# Demo Output





# Step-1: Access data

- » **6 (2x3) categories of people**
  - 2 sexes (Male / Female), denoted as **sex**
  - 3 socioeconomic status, denoted as **ses**
- » **Data accessor**
- » a function to fetch data/property given an object like the one on the right.
- » **Data range**
- » map string-type properties like sex and ses to numerical values (0, 1, ...)

```
, {  
  "sex": "male",  
  "ses": "low",  
  "<High School": 10.0,  
  "High School": 26.5,  
  "Some Post-secondary": 35.8,  
  "Post-secondary": 8.7,  
  "Associate's": 6.9,  
  "Bachelor's and up": 12.2  
}, {
```

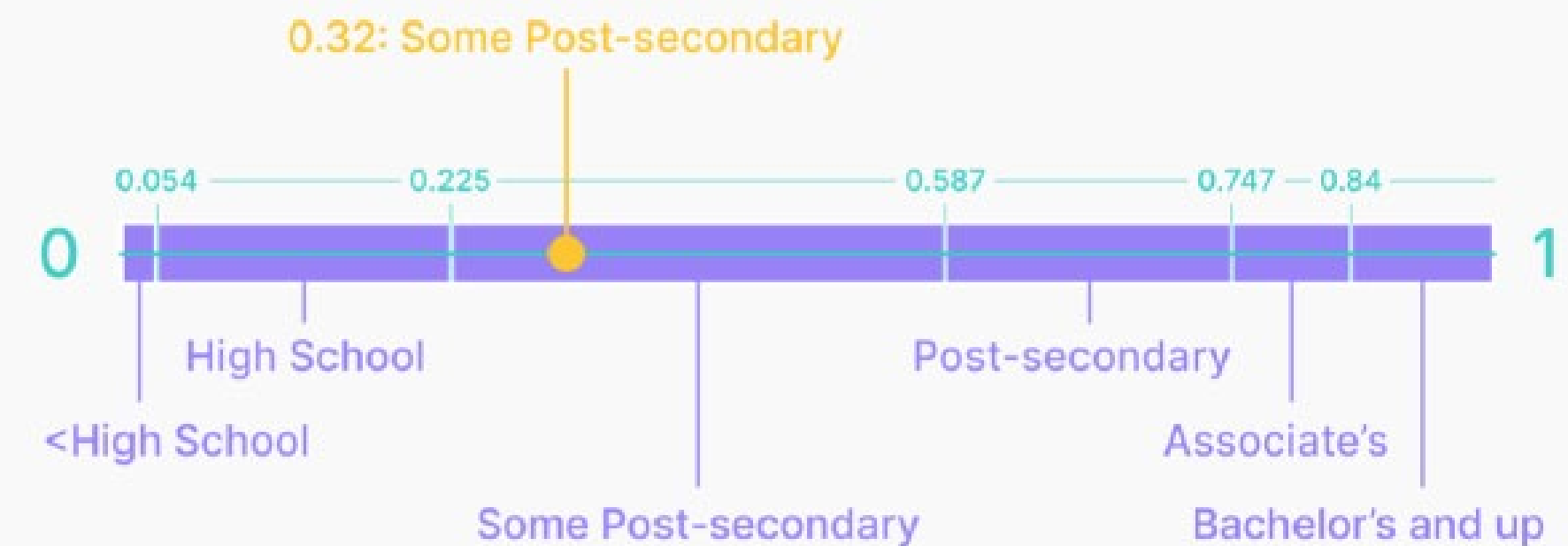




## Step-2: Stack probabilities

- » 6 different levels of highest degree
- » Sum up to 1 but may has rounding error, that you need to handle.
- » **Rounding Error:**
- » Floating-type values are not as accurate as we think. For example:
- » 0.9999...999 can be still considered as 1.0.
- » Stack these probabilities, and we can use random value from 0 to 1 to determine which level to use.

```
, {  
  "sex": "male",  
  "ses": "low",  
  "<High School": 10.0,  
  "High School": 26.5,  
  "Some Post-secondary": 35.8,  
  "Post-secondary": 8.7,  
  "Associate's": 6.9,  
  "Bachelor's and up": 12.2  
}, {
```





# Step-3: Create people

- » TODO: generatePerson() function
- » **statusKey**: used as the key of the 6 categories (2 sub-keys: sex and ses)
- » Already defined in the template
- » `const getStatusKey = ({sex, ses}) => [sex, ses].join("--")`
- » **d3.bisect(stackedProbArray, randomValue)**
- » `edu = d3.bisect([0.1, 0.2, 0.3 0.5, 0.8, 1.0], Math.random())`
- » The total number of people is up to you (10,000 for example).
- » Cannot be too small.

► {sex: 1, ses: 2, education: 4}

[chart.js:56](#)

► {sex: 0, ses: 1, education: 5}

[chart.js:57](#)

► {sex: 1, ses: 0, education: 2}

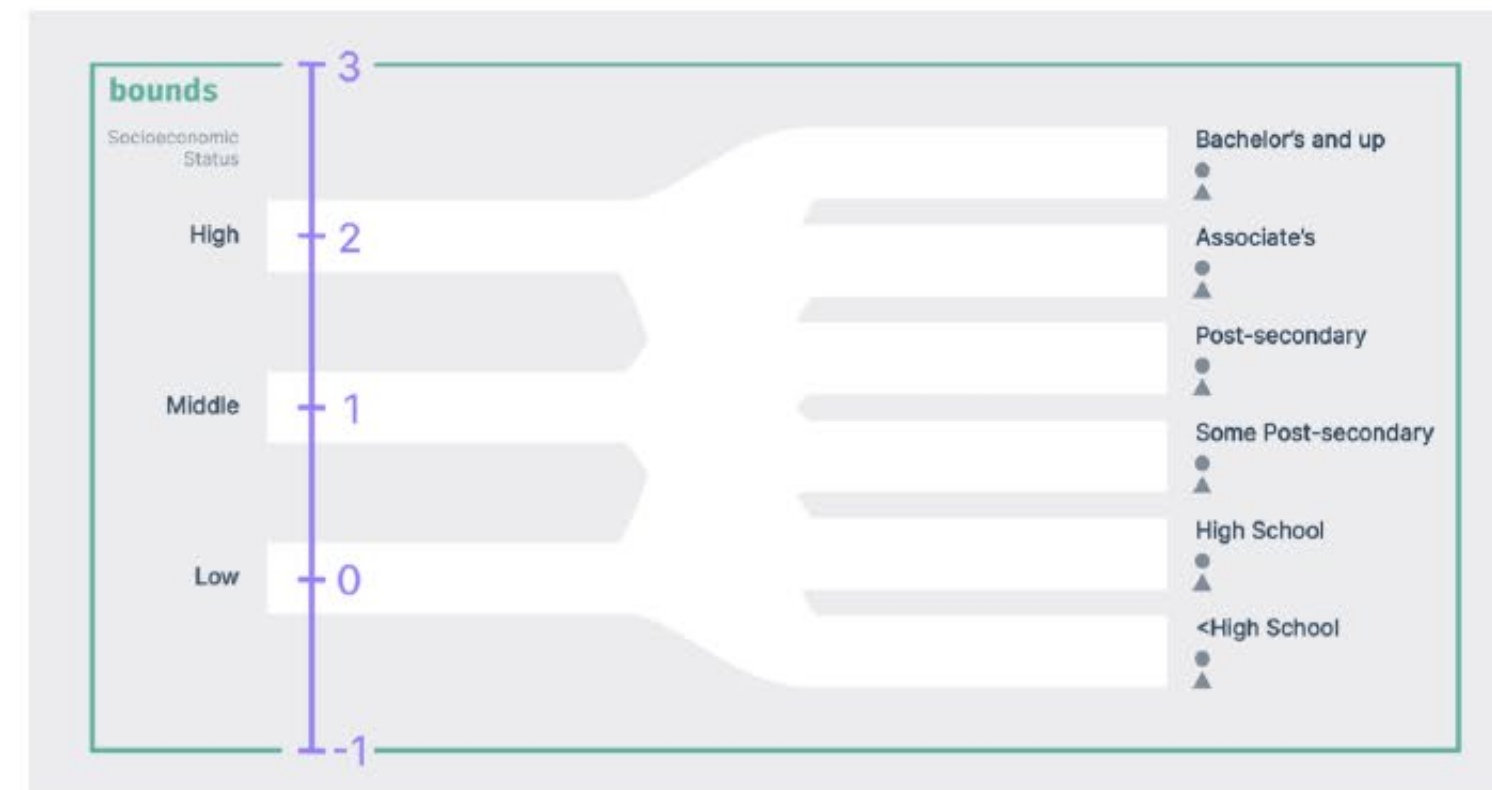
[chart.js:58](#)

# Step-4: Draw paths

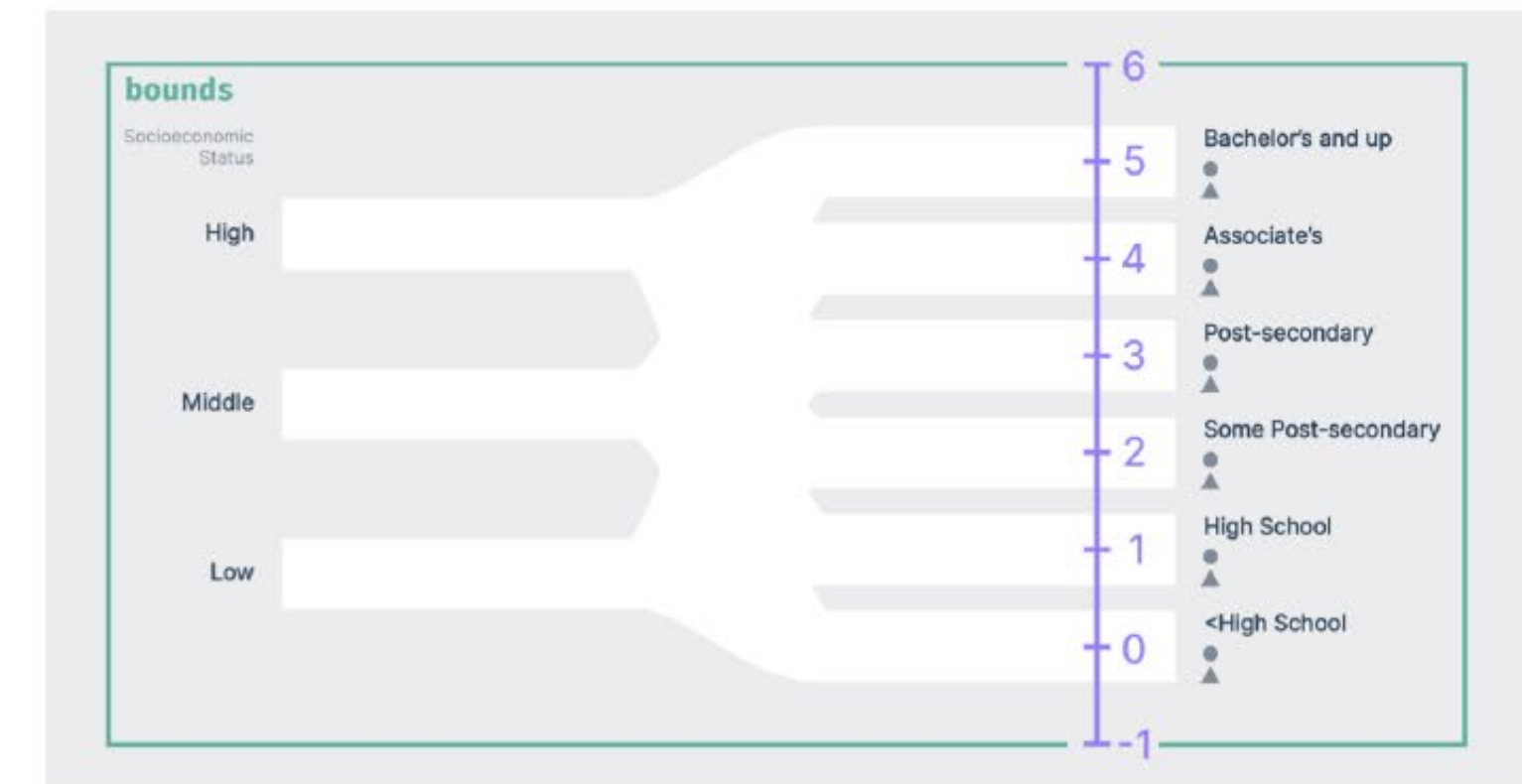


香港中文大學(深圳)  
The Chinese University of Hong Kong, Shenzhen

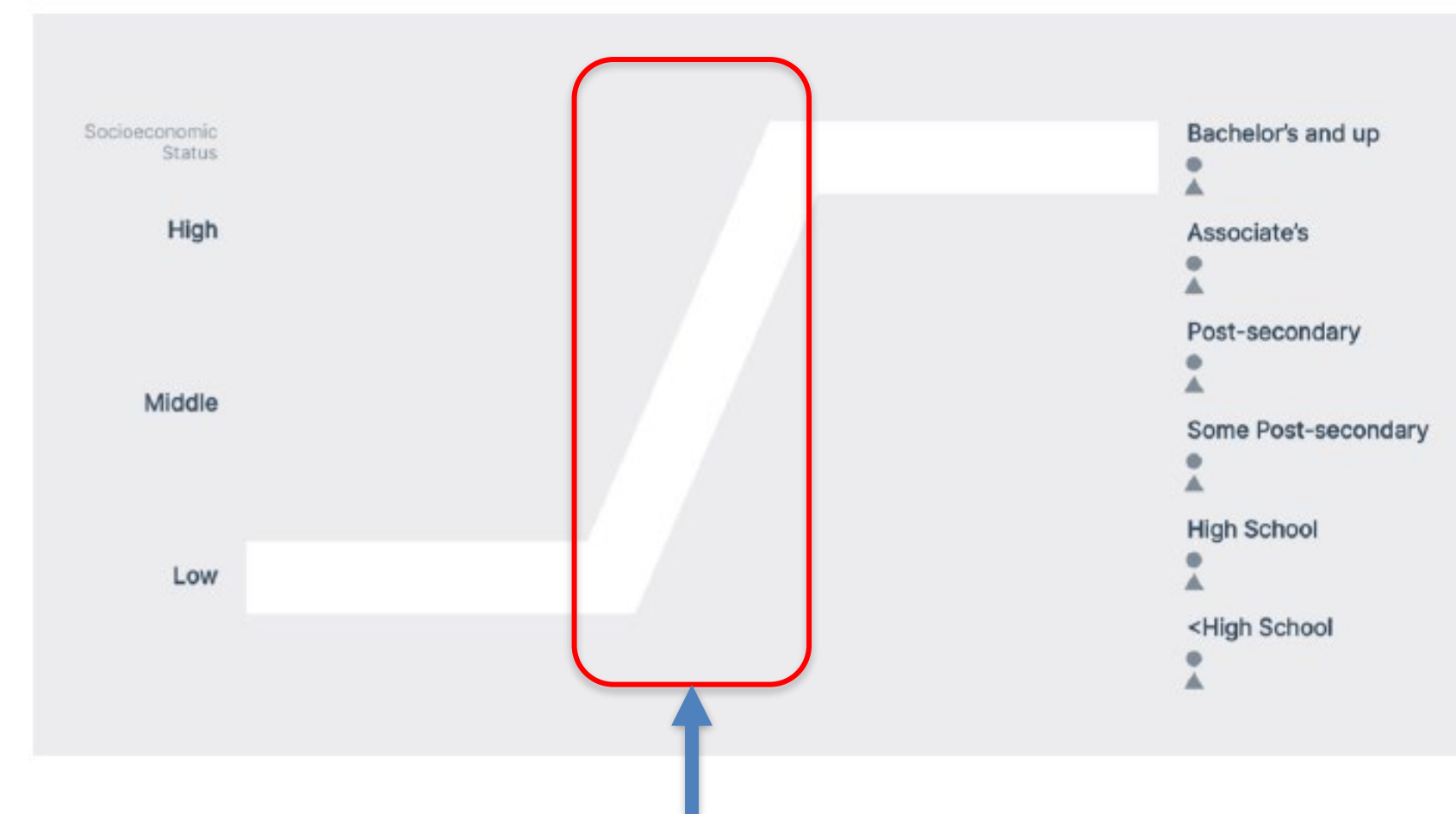
- » Need to define 5 scales
  - xScale (for SVG)
  - startYScale (for ses)
  - endYScale (for education)
  - transitionYScale (for line)
  - colorScale (for sex)



startYScale



endYScale

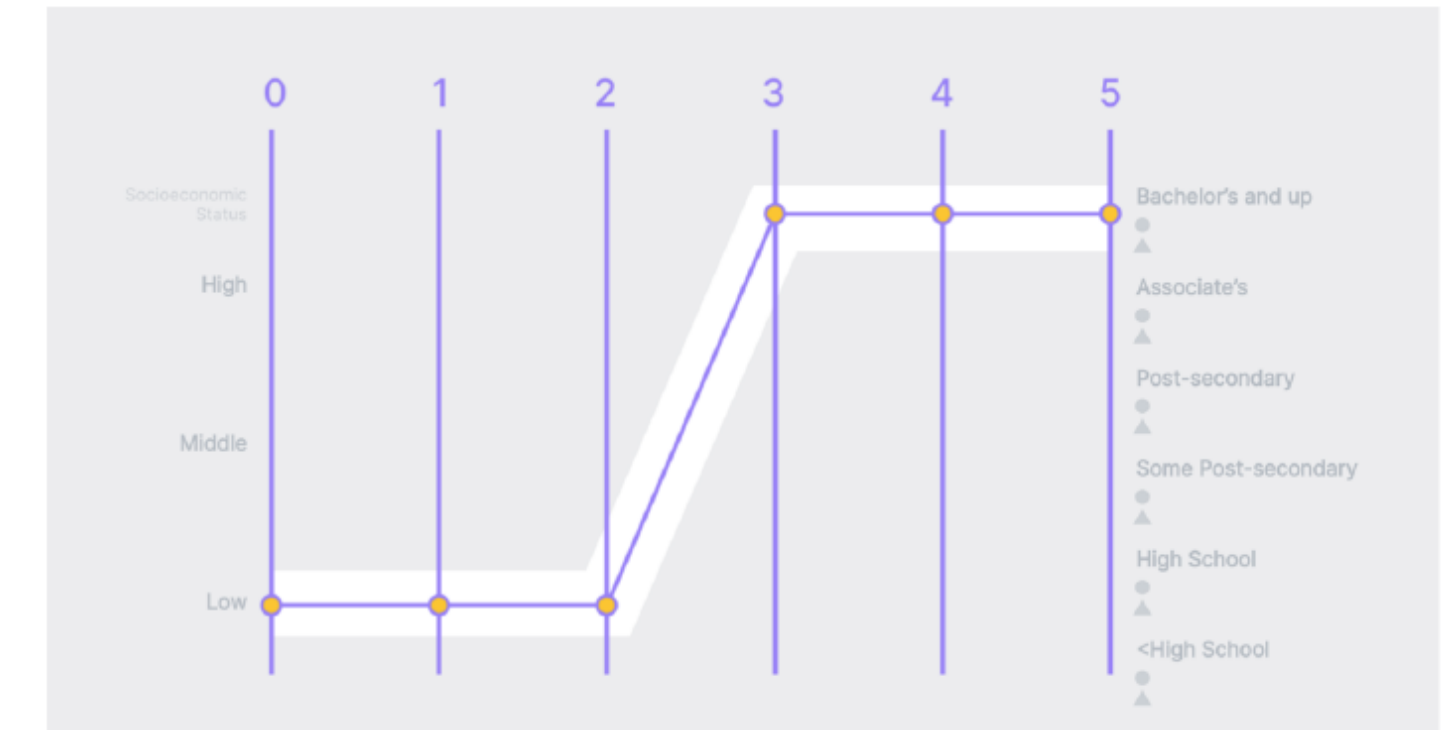


transitionYScale

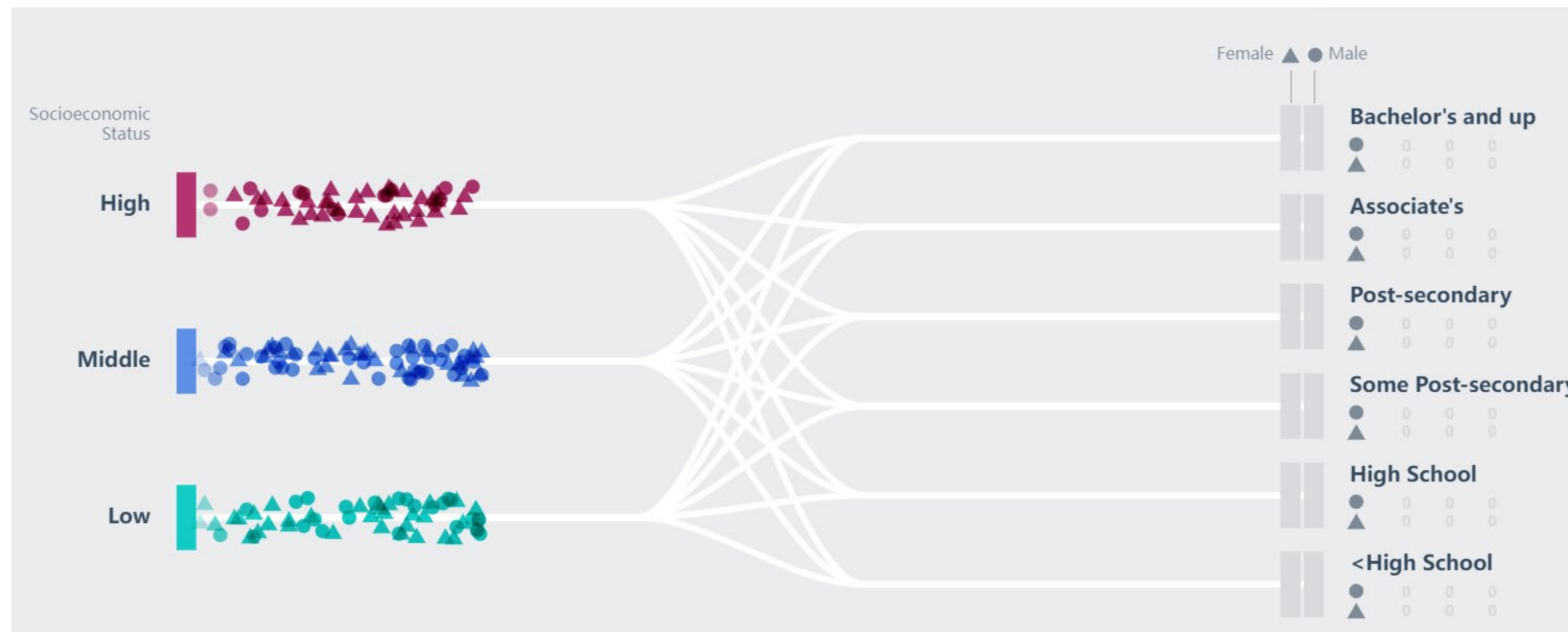


# Step-4: Draw paths

- » 18 (3 ses x 6 edu) paths
- » Each path consists a line element.
- » Each line element has 6 points.



(b) Path parts diagram



18 paths with stroke-width = 5, and you need a larger width

```
▼ 0: Array(6)
  ► 0: (2) [0, 0]
  ► 1: (2) [0, 0]
  ► 2: (2) [0, 0]
  ► 3: (2) [0, 0]
  ► 4: (2) [0, 0]
  ► 5: (2) [0, 0]
  length: 6
```

6 points in one line have  
the same value





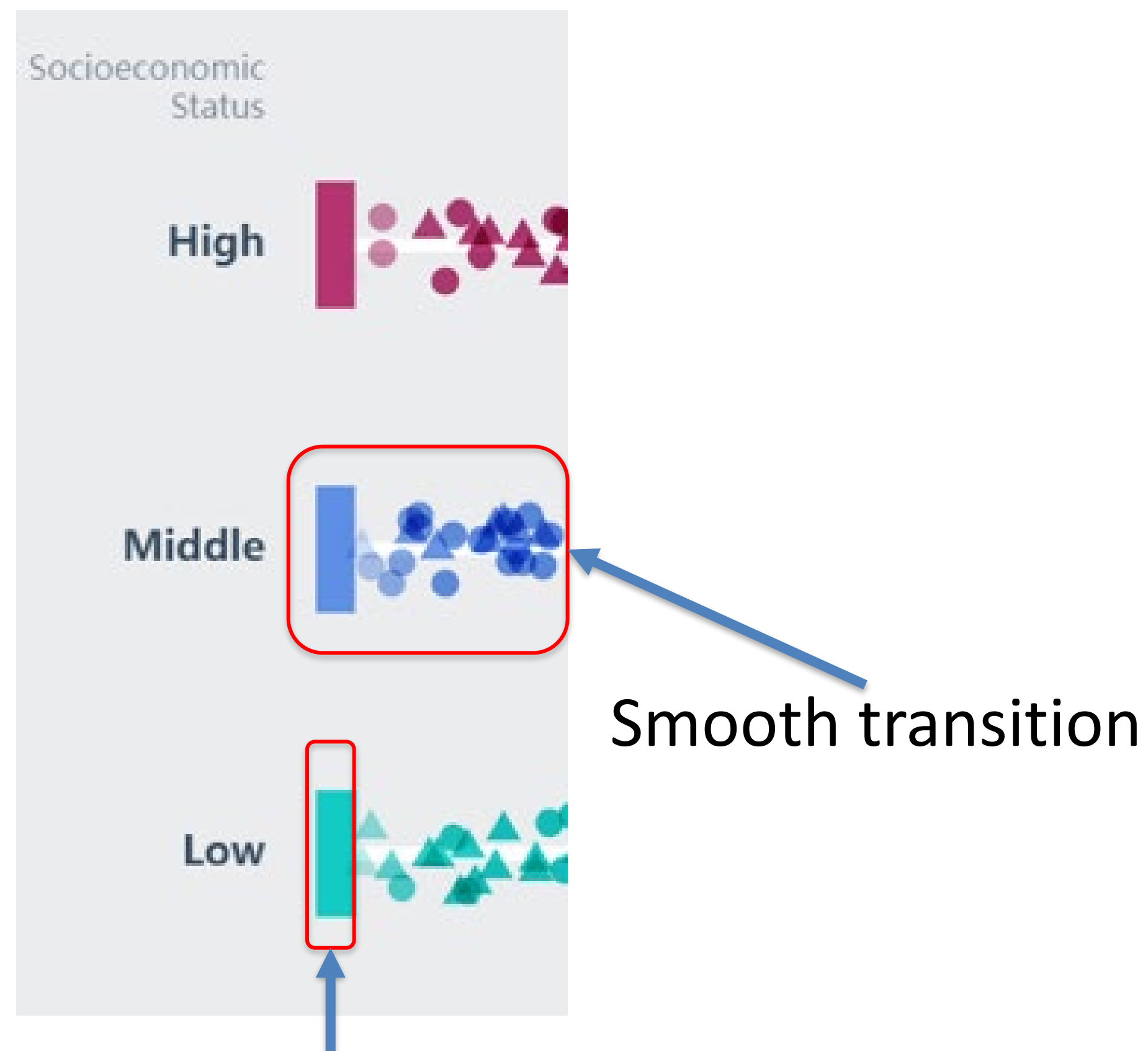
# Step-5: Visualize people

- » Simply draw the markers (circles and polygons)
- » d3.timer
- » `d3.timer(updateMarkers)`
- » Function `updateMarkers(elapsed)`
- » `elapsed` refers to how much time has passed after launching this timer, and the `updateMarkers` (callback) will be periodically triggered to update the position **and opacity** of the markers.
- » Be careful with the join / enter+append. Normal join may not be enough for this task since our marker elements are periodically updating.
- » **[Recommended Reading]**
- » <https://observablehq.com/@thetylerwolf/day-18-join-enter-update-exit>



# Step-5: Visualize people

- » **Note:** The opacity of markers should be changed smoothly from 0 to 1, and the marker should only be visible after the starting rectangles.



Markers invisible within this rectangle

```
250 const markers = d3.selectAll(".marker")
251
252 markers.style("transform", d => {
253   const x = xScale(xProgressAccessor(d))
254   const yStart = startYScale(sesAccessor(d))
255   const yEnd = endYScale(educationAccessor(d))
256   const yChange = yEnd - yStart
257   const yProgress = yTransitionProgressScale(
258     xProgressAccessor(d)
259   )
260   const y = yStart
261     + (yChange * yProgress)
262     + d.yJitter
263   return `translate(${x}px, ${y}px)`
264 })
265 .attr("fill", d => colorScale(sesAccessor(d)))
266 .transition().duration(100)
267   .style("opacity", d => xScale(xProgressAccessor(d)) < 10
268     ? 0
269     : 1
270 )
```

Template chart.js



## Step-6: Add color and filter

- » Remove markers that have finished their journey.
- » A person takes 5 seconds (5000 milliseconds) to cross the chart
- » `const xProgressAccessor = d => (elapsed - d.startTime) / 5000`
- » `xProgressAccessor(data) = 1 =>` means finished
- » (Already given in your template)



## Step-7&8: Visualize Stacked Bar Charts and Count

- » Collect the number people/markers that have finished their journey for each category.
- » Then, draw 2 stacked bar charts (one for Male and another for Female) with these numbers for each education level.
- » The counts and the percentage of stacked bar charts should match to the probabilities in the JSON file.



# d3-sankey?



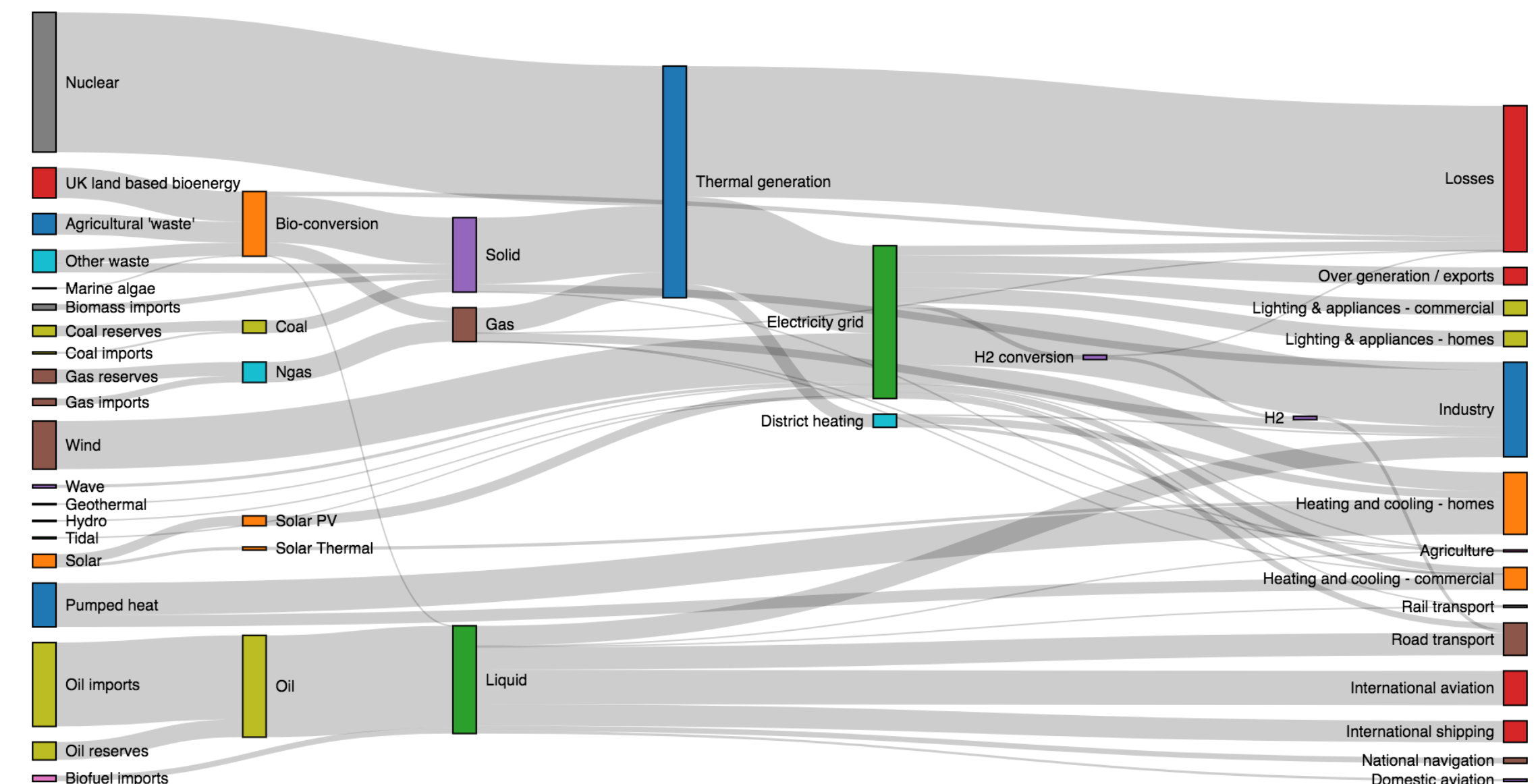
香港中文大學(深圳)  
The Chinese University of Hong Kong, Shenzhen

- » d3-sankey is a sub-library under D3 similar to d3-force and d3-Delaunay.
- » Link of GitHub Repo: <https://github.com/d3/d3-sankey>
- » Sankey diagrams visualize the directed flow between nodes in an acyclic network.
- » For example, this diagram shows a possible scenario of UK energy production and consumption in 2050:

- » d3-sankey can also be used to implement
- » this assignment, and you can have a try
- » to learn how to use it by yourself.

» Both ways are OK

- Traditional D3
- d3-sankey





# Online Resources for Assignment 4

- » **d3-sankey**
- » <https://github.com/d3/d3-sankey>
- » **Join & Enter/Append (Recommended)**
- » <https://observablehq.com/@thetylerwolf/day-18-join-enter-update-exit>
- » And the resources provided in the instruction.



# Q&A for Assignment 4

This assignment is very complicated and time-consuming.  
Please get started as early as possible and prepare enough  
time for you to finish.