

Project 1 Awards, Using GenAI, JS (Part 2)

DSC 106: Data Visualization

Sam Lau

UC San Diego

Announcements

Lab 5 due Friday.

Project 3 coming out soon, checkpoint due on Tuesday.

FAQs:

1. **How do I get the Lab 5 extra credit?** Fix the bug described in Step 5.4, then show us the working website in your video.

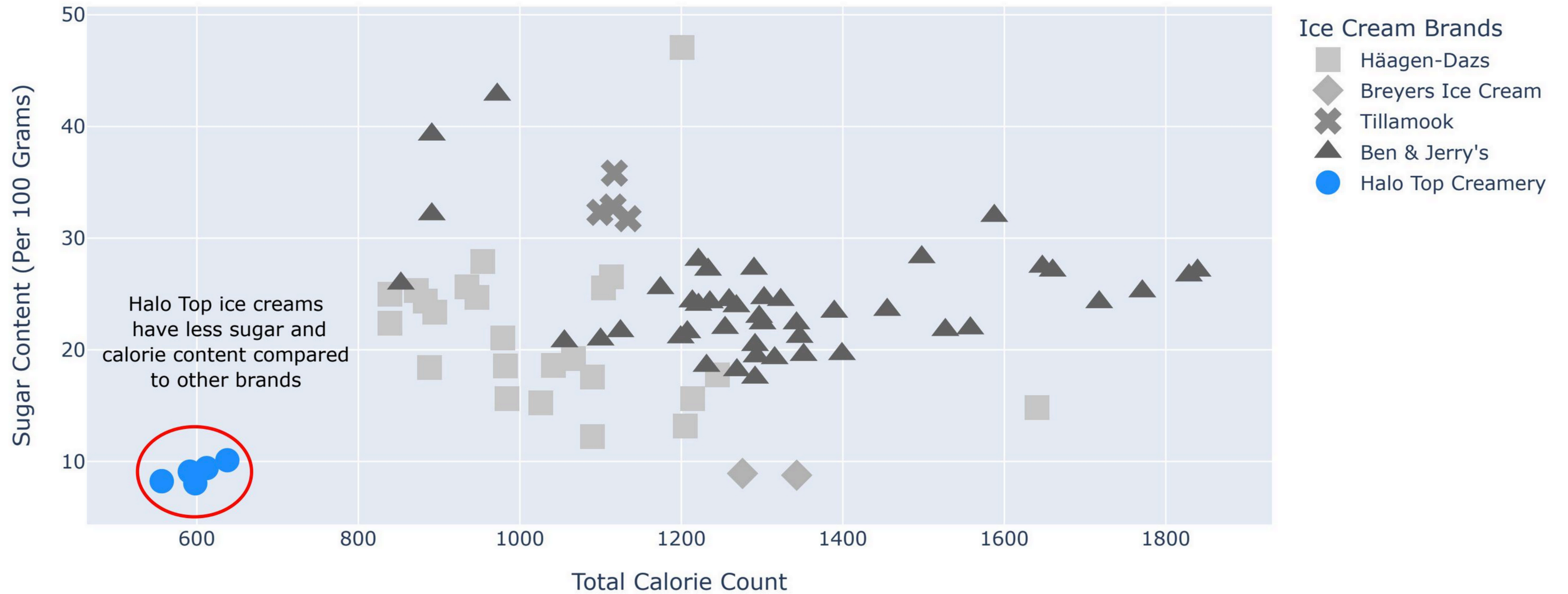
Project 1 Awards

Put this on your resume!

"Received best project award, given to top 5% in a class of 150 students."

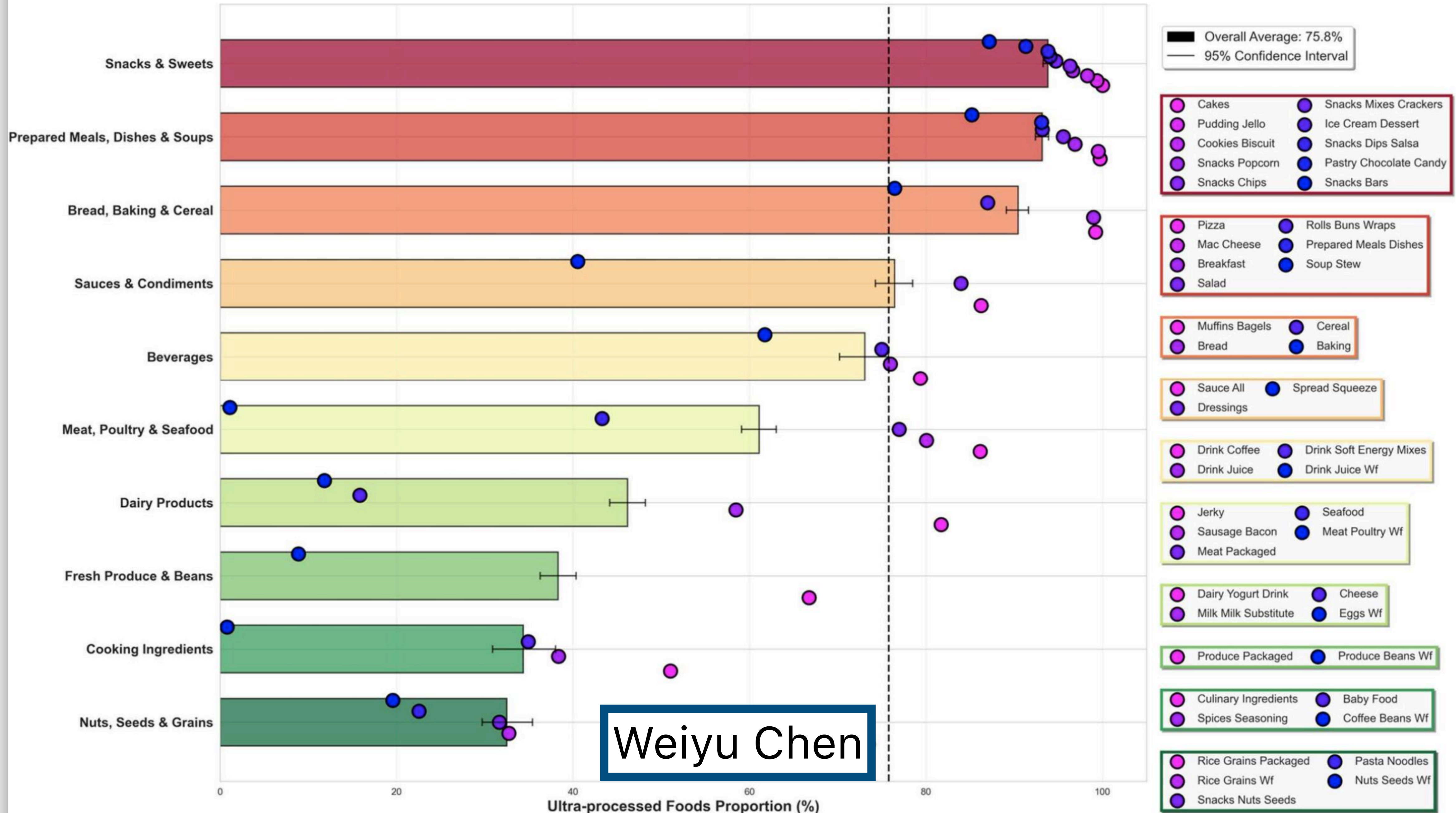
Is Halo Top's Ice Cream as Healthy as Advertised?

Halo Top advertise their ice cream as a healthier alternative, how does it compare with other popular ice cream brands?



Tony Bai

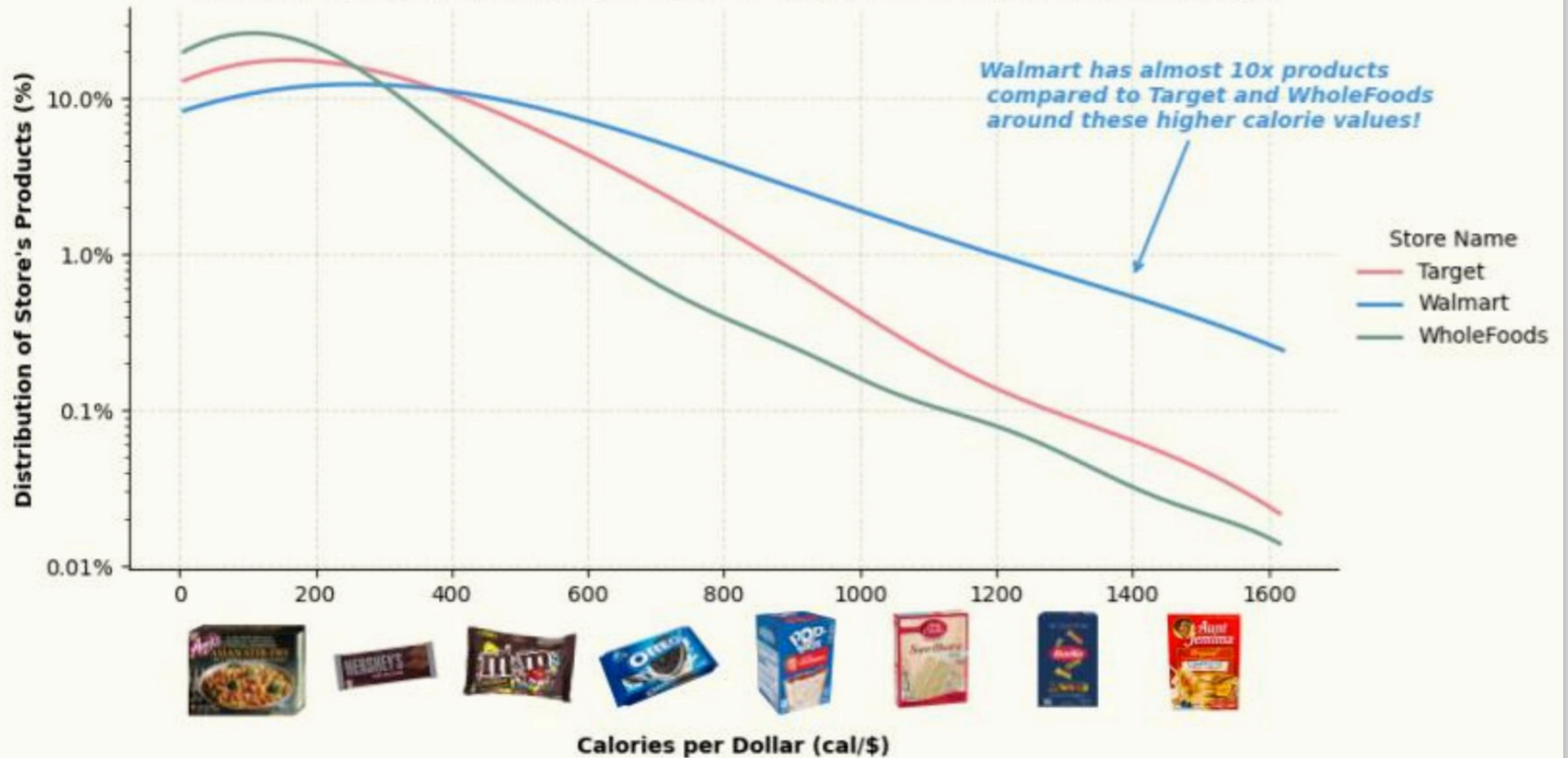
Ultra-processed Food Proportions by Category Groups Gradient from High Processing to Low Processing



Weiyu Chen

Walmart Offers the Most Calories For Your Dollar!

Each curve shows how a store's products are distributed by calorie-per-dollar value.

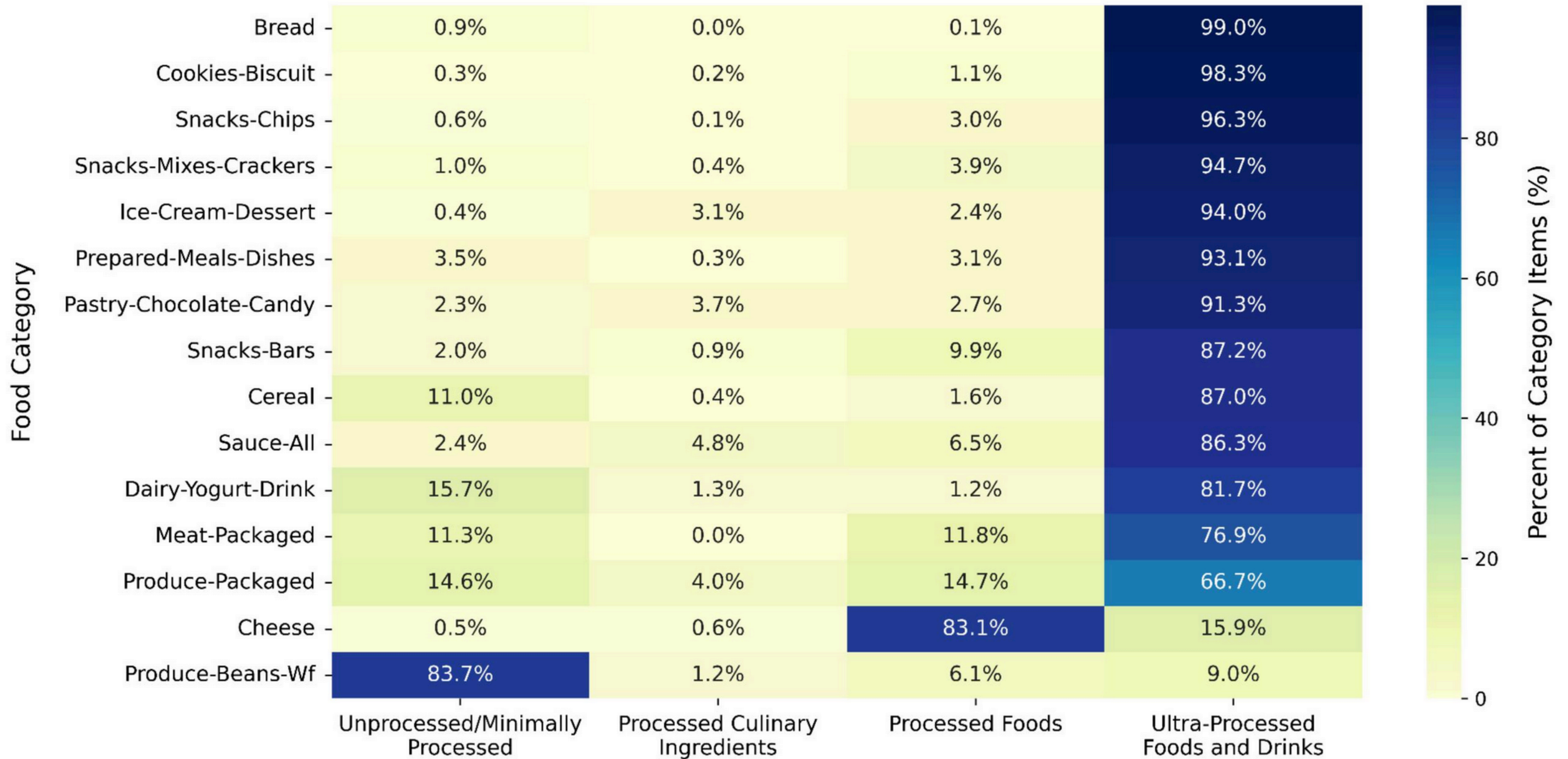


Note: Outliers not within the 1st-99th percentile were removed.
A log scale was applied to highlight smaller groups of high calorie-per-dollar products.

Jacob Doan

Ultra-Processed Foods Dominate the Majority of Grocery Stores Food Categories

Disproportionate Distribution of Food Processing Levels Within Top 15 Food Categories
(Each row represents the percentage of items in that category)

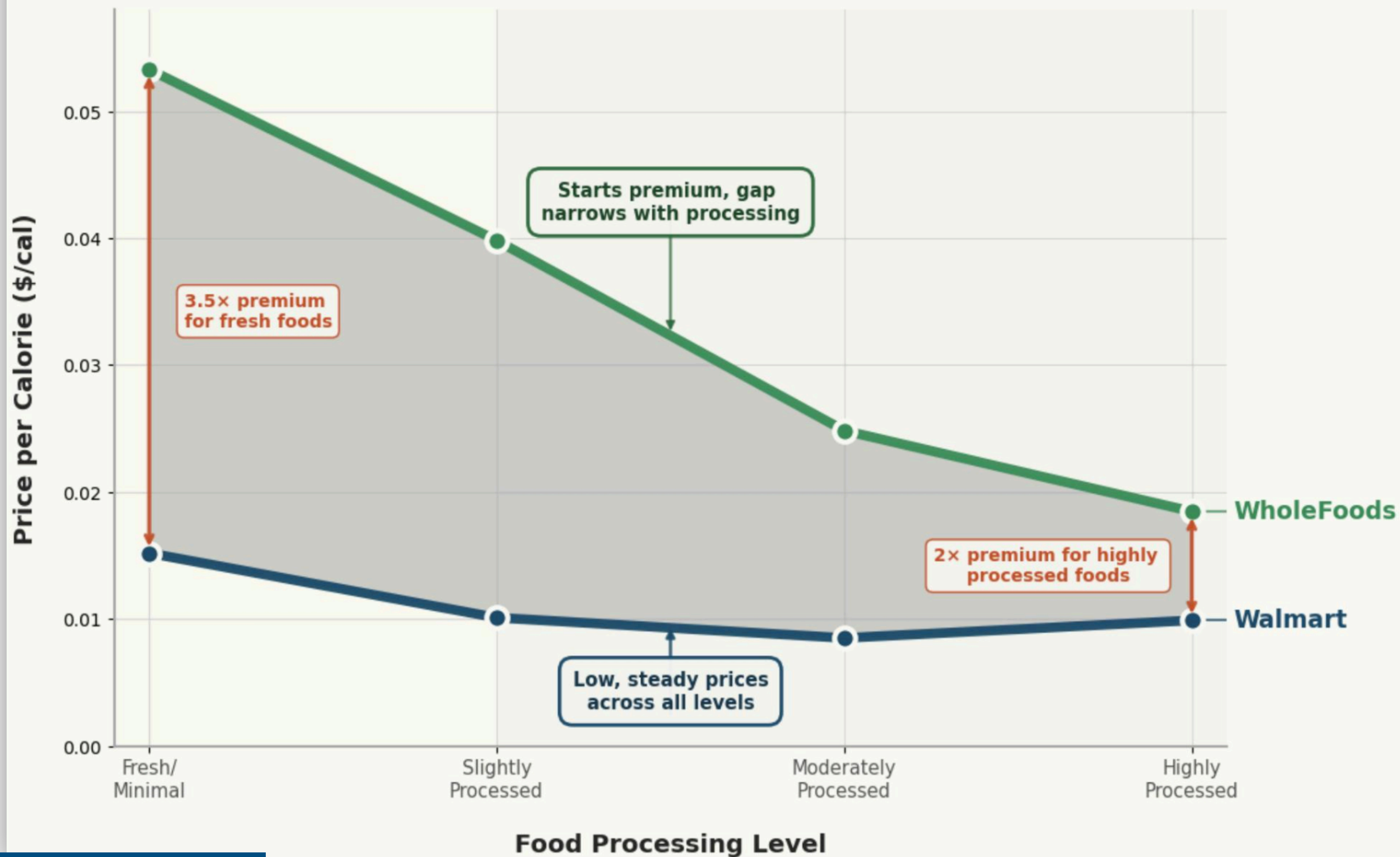


Erika Tong

Food Processing Class From Least to Most Processed

The Fresh Food Premium: Does It Fade or Stay?

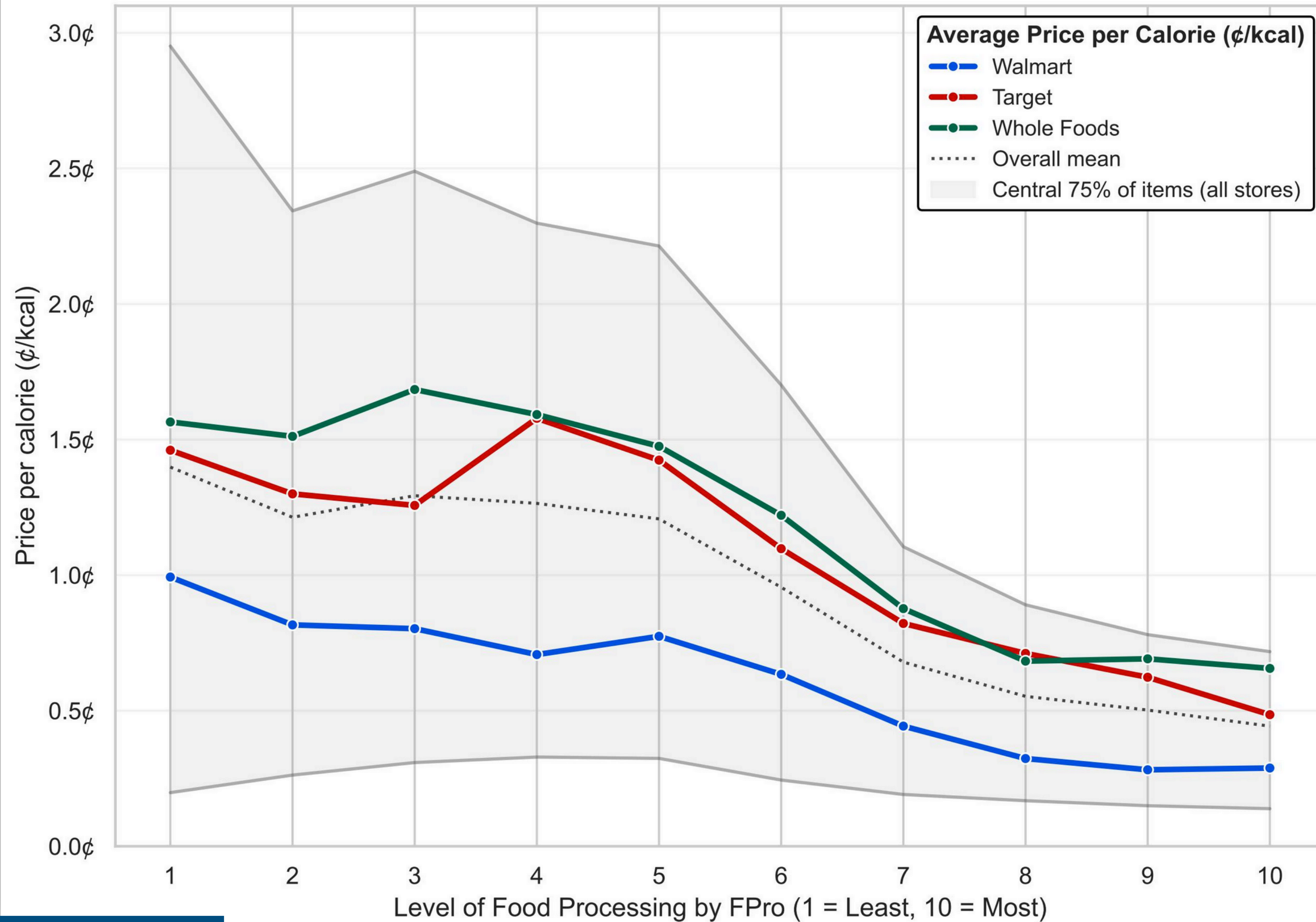
The price gap between premium and budget stores shrinks as food becomes increasingly processed.



Dylan Dsouza

Does More Processing Mean Cheaper Food?

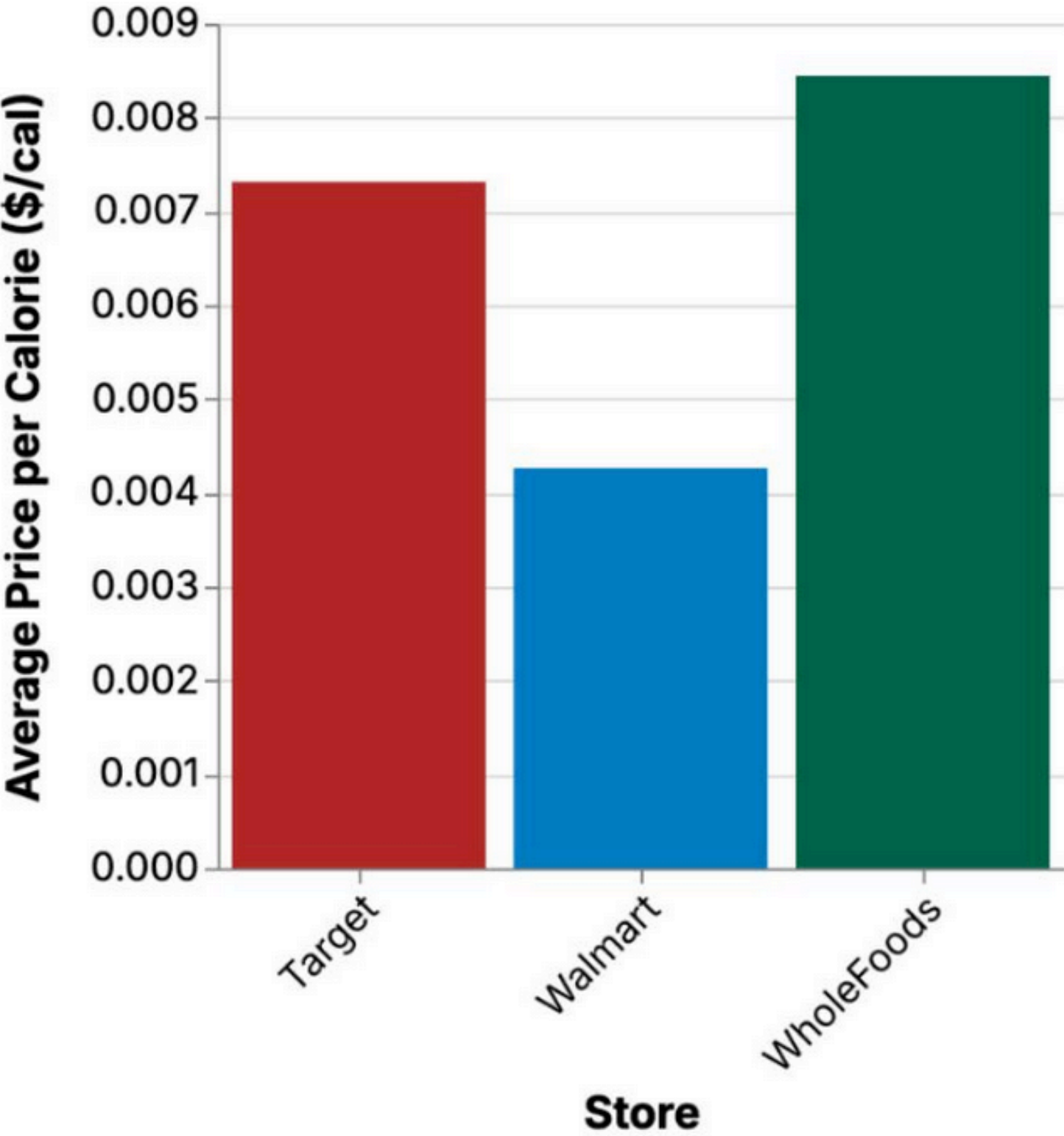
Across Walmart, Target, and Whole Foods, price per calorie falls as foods become more processed.



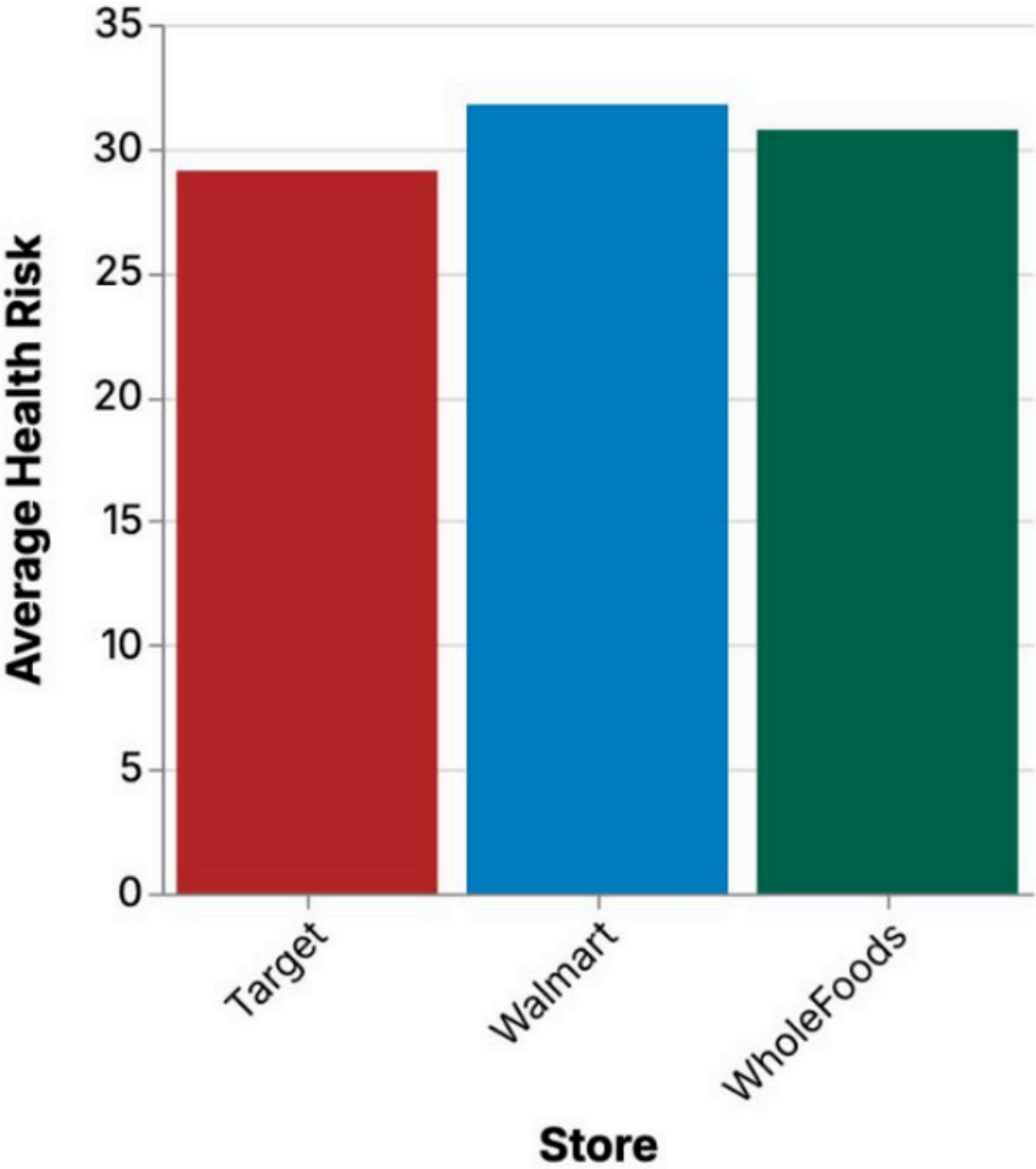
Terence Zhang

Pizza Perplexion: WholeFoods pizzas are expensive and no healthier than their competitors

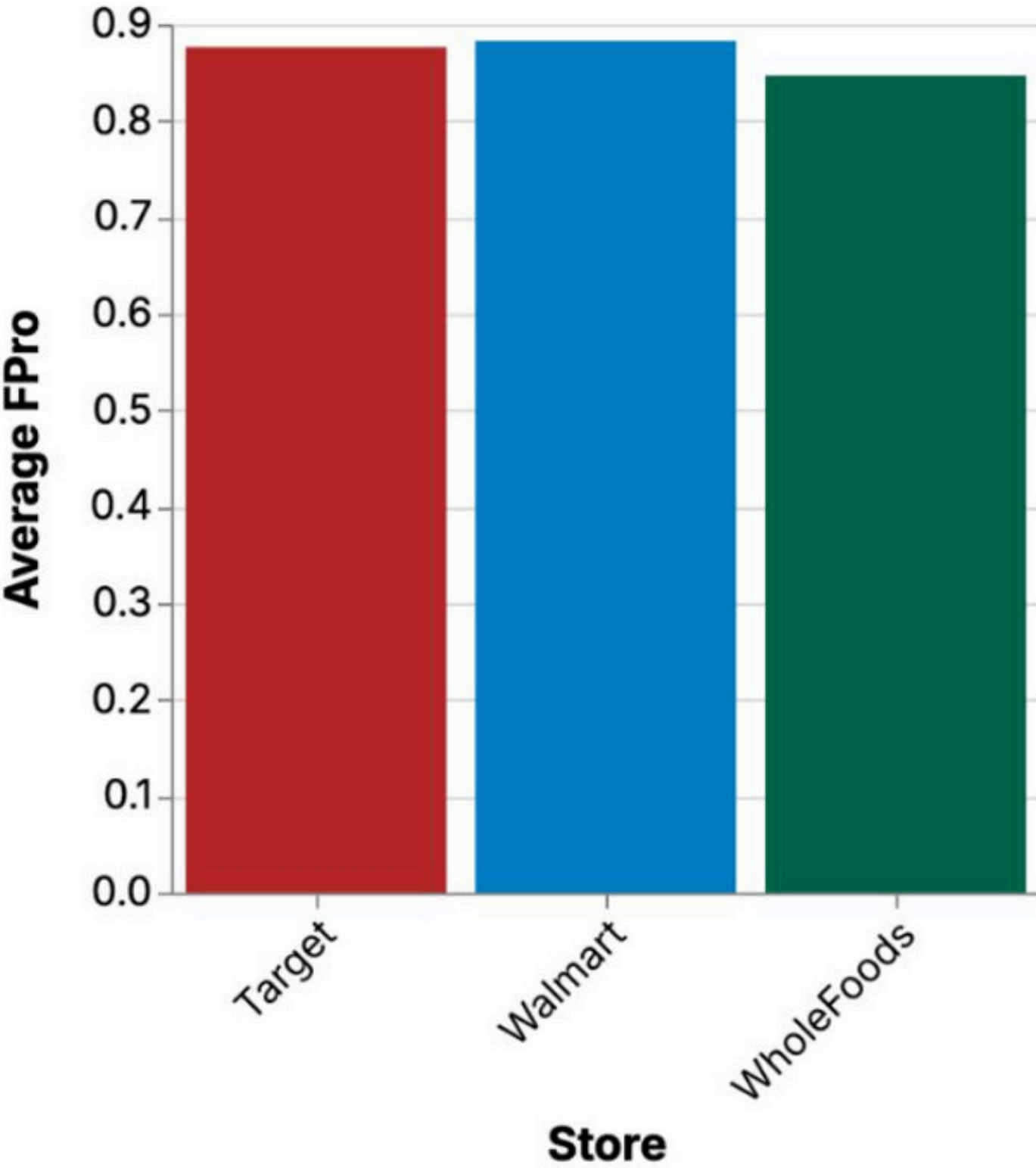
WholeFoods pizzas are on average the most expensive in price per calorie



While pizzas from all three stores have similar average health risk



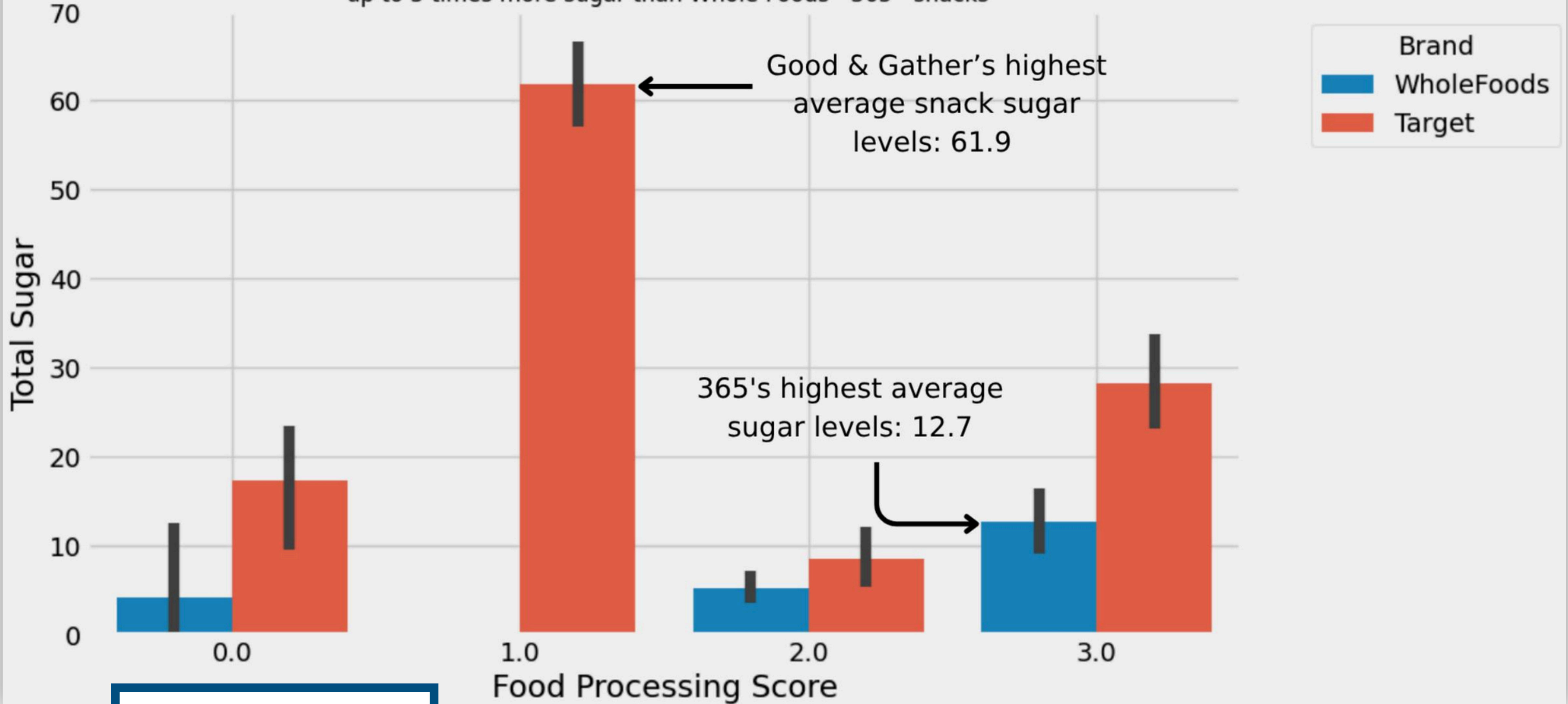
And similar average food processing scores



Mieko Chun

Unhealthy Doesn't Always Mean Highly Processed

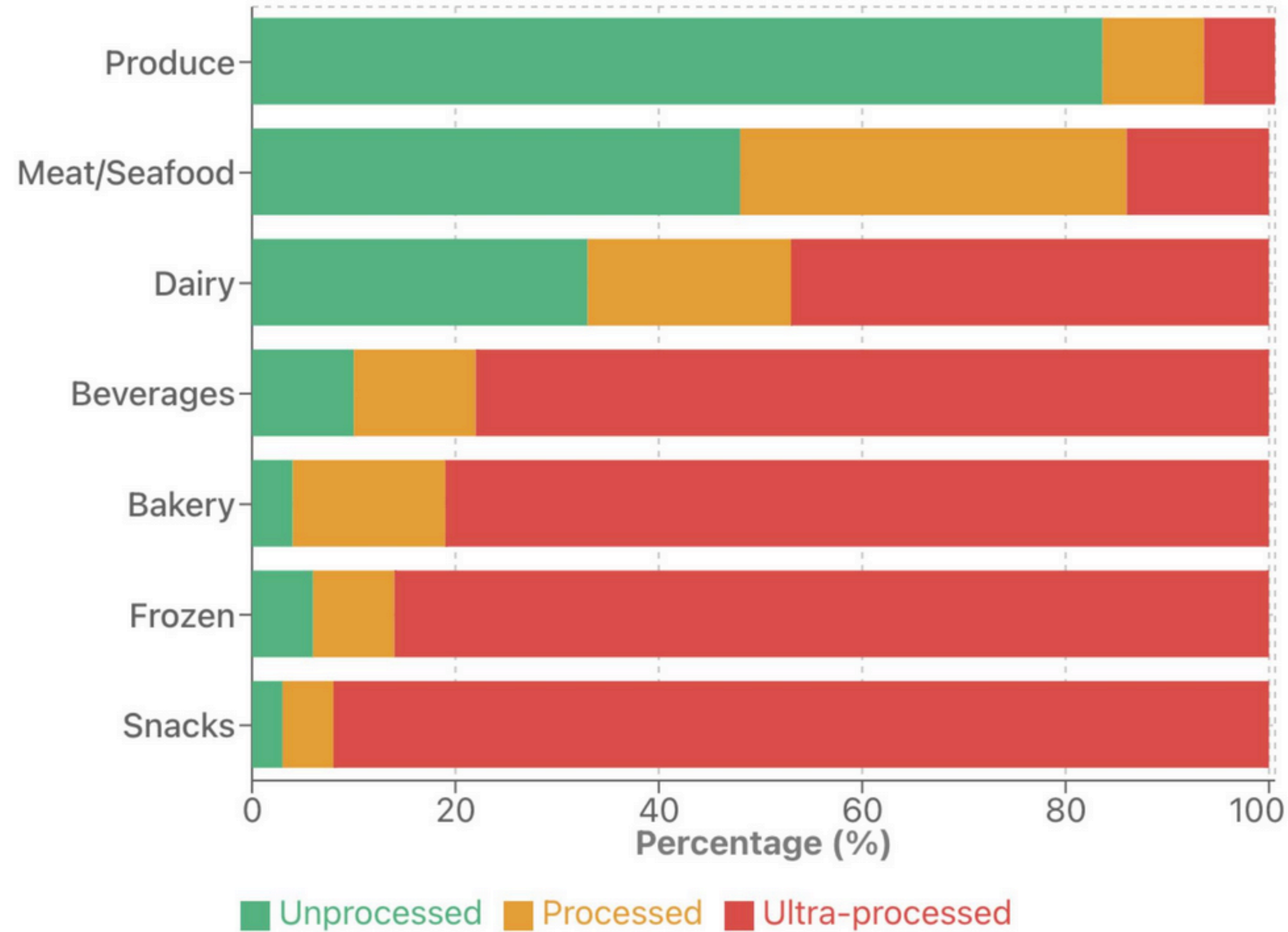
Target's 'Good & Gather' snacks are less processed but can contain up to 5 times more sugar than Whole Foods' '365' snacks



Susana Haing

Watch What You Buy: 67% of Grocery Store Food Is Ultra-Processed

Ultra-processed foods make up 5.6 times more of grocery store inventory than unprocessed foods. 75.8% of products are ultra-processed while the rest are minimally processed.



How to use GenAI in this course

Philosophy



ChatGPT



Whoa, you're not learning "real" programming!

```
print('hello world')
```



Whoa, you're not learning "real" programming!

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

Sometimes details matter (e.g. high-perf computing, kernel drivers). But not always!

What you should know

Visual encodings

Perception, color

How to design a good plot!

JS syntax, functions, objects

The document object model

Event handlers/listeners

D3 core abstractions

What to use GenAI for

Brainstorming plot ideas, generating options to choose from.

Writing simple functions

document API and functions

Event names

Creating initial plot prototypes

You Try: Your favorite website interaction

Go to your favorite website, pick an interactive element, describe event listener and event handler.

tryclassbuzz.com
Code: **interaction**

JS (Part 2)

Async / await

js-lecture/weather01/

(demo)

```
<head>
  <title>Temperature Converter</title>

  <link rel="stylesheet" href="main.css" />
  <script src="main.js" type="module"></script>
```

If main.js code is slow, then page will freeze (!) while waiting for JS to finish

What if the dataset just takes a while to download?

Idea: Allow functions to run in the background (asynchronously) so that page doesn't freeze

```
async function loadWeatherData() {  
  try {  
    const response = await fetch('./weather-data.json');  
    const weatherData = await response.json();  
    return weatherData;  
  } catch (error) {  
    console.error('Error loading weather data:', error);  
  }  
}
```

async = this function uses other async functions

```
async function loadWeatherData() {
  try {
    const response = await fetch('./weather-data.json');
    const weatherData = await response.json();
    return weatherData;
  } catch (error) {
    console.error('Error loading weather data:', error);
  }
}
```

async = this function uses other async functions

await = this function might take a while, so let the browser do other stuff while we wait

```

function loadStory() {
  return getJSON('story.json')
    .then(function (story) {
      addHtmlToPage(story.heading);

      return story.chapterURLs
        .map(getJSON)
        .reduce(function (chain, chapterPromise) {
          return chain
            .then(function () {
              return chapterPromise;
            })
            .then(function (chapter) {
              addHtmlToPage(chapter.html);
            });
        }, Promise.resolve());
    })
    .then(function () {
      addTextToPage('All done');
    })
    .catch(function (err) {
      addTextToPage('Argh, broken: ' + err.message);
    })
    .then(function () {
      document.querySelector('.spinner').style.display = 'none';
    });
}

```

Back in the day, we had to use JS Promises that had a `.then()` and `.catch()` syntax.

async/await is the modern version that makes writing this code a LOT easier

<https://jakearchibald.com/2014/es7-async-functions/>

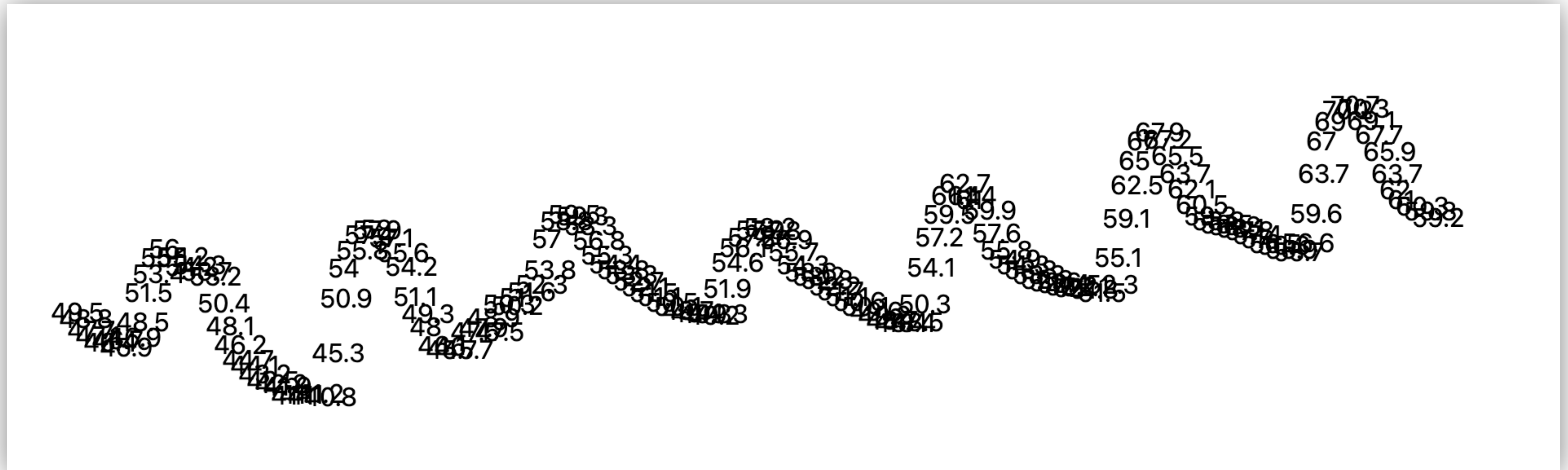
```

async function loadStory() {
  try {
    let story = await getJSON('story.json');
    addHtmlToPage(story.heading);
    for (let chapter of story.chapterURLs.map(getJSON)) {
      addHtmlToPage(await chapter.html);
    }
    addTextToPage('All done');
  } catch (err) {
    addTextToPage('Argh, broken: ' + err.message);
  }
  document.querySelector('.spinner').style.display = 'none';
}

```

Half the code!

Now, let's make our very first data visualization in JS:

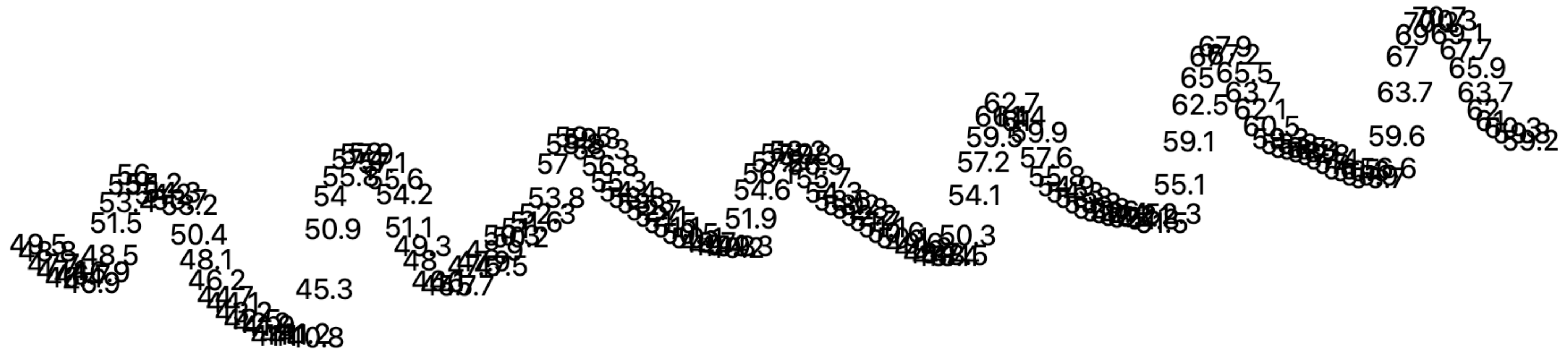


[js-lecture/weather02/](#)

(demo)

[js-lecture/weather03/](#)

(demo)



How would you add an x-axis and y-axis? Gridlines?

tryclassbuzz.com

Code: **axes**