

# **ECE 448/528**

## **Application Software Design**

### **Lecture 25. Data Visualization**

#### **Spring 2025**

**Won-Jae Yi, Ph.D.**

**Department of Electrical and Computer Engineering**  
**Illinois Institute of Technology**

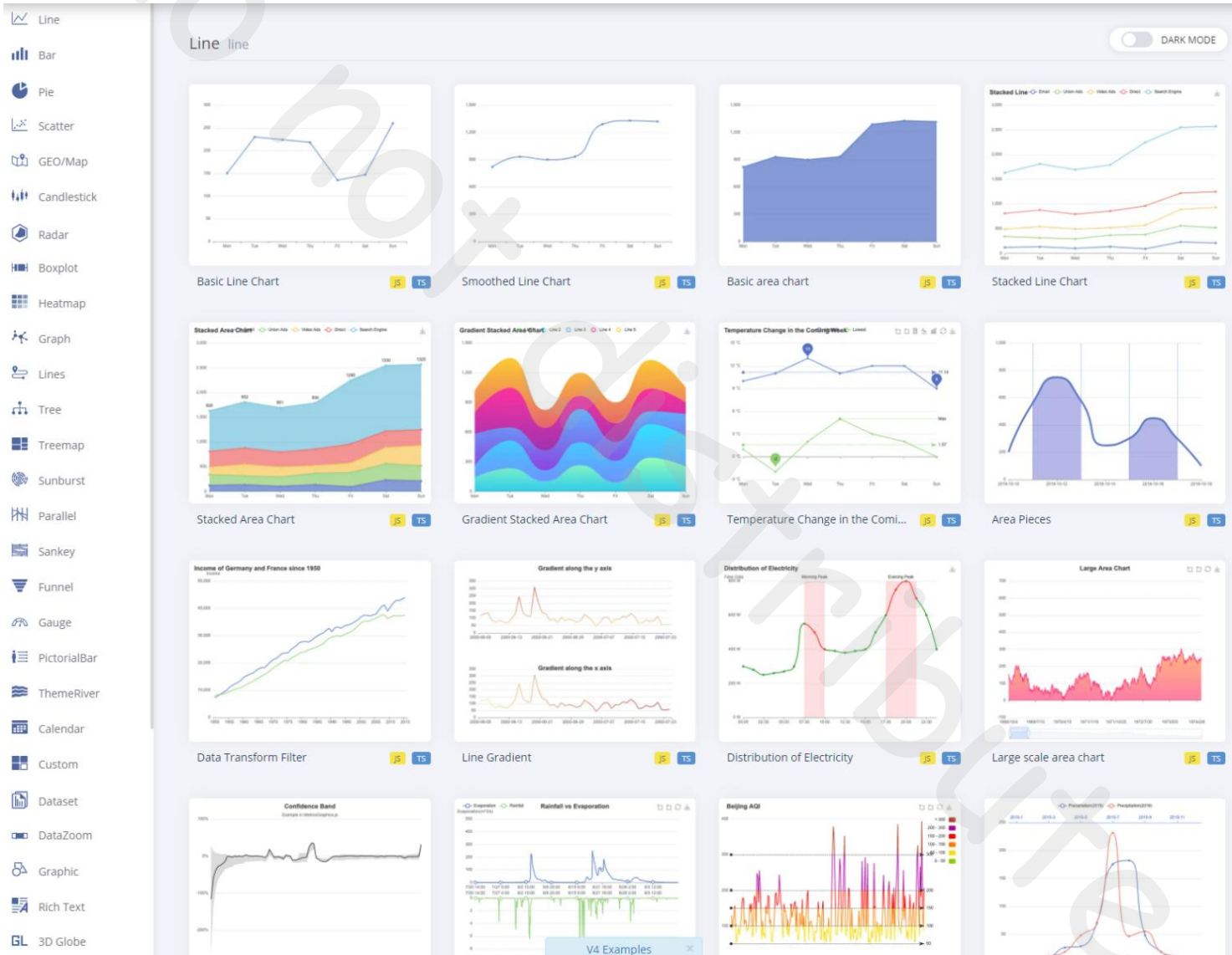
# Data Visualization

- To present data in a way to help humans understand them better
  - Visual communication is preferred because of its ability to carry a huge amount of data.
  - Graphics are intuitive to understand and are not bounded by language barriers and educational background.
- Web applications provide unique advantages for data visualization
  - The ability to visualize data in real-time.
  - The ability to integrate data from multiple sources.
  - The ability to allow human to interact with the visualization.
- Web-based data visualization techniques.
  - Use well established diagrams and charts from a visualization library.
  - Create your own solution from graphic primitives like lines and circles.

# Charts

- Help to reveal internal relations among data in series.
- Many popular choices depending on the type of data, e.g.
  - Bar charts for categorical data.
  - Line charts for continuous data.
  - Pie charts to show portion statistics.
  - X-Y charts to show dependencies.
- We will demonstrate the use of the Apache ECharts library (<https://echarts.apache.org/>)
  - In `public/power.html` under branch `lec25`
  - Pay attention on how to integrate the library with React (and other MVC libraries in general).

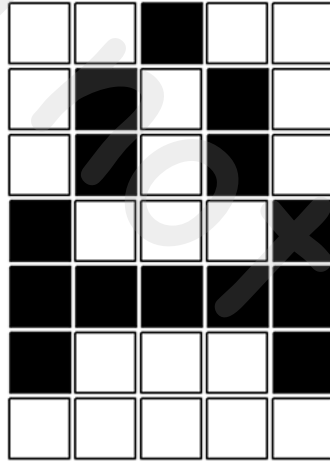
# Apache ECharts



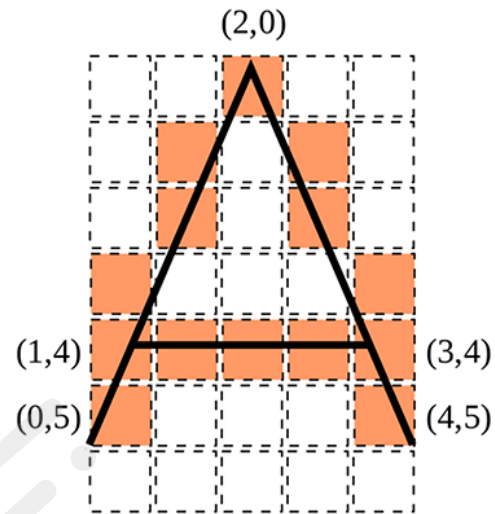
# Web-based Graphics

- Scalable Vector Graphics (SVG)
  - Vector graphics that can be enlarged or shrunk without losing details.
  - HTML tags are introduced for SVG primitives like lines, curves, circles, and paths.
  - Interactive visualization can be easily enabled by associating event handlers with those SVG tags.
  - As part of HTML DOM, SVG can be manipulated directly in the React application.
- HTML5 `<canvas>`
  - Raster graphics that work with pixels.
  - The basic Canvas API allows JavaScript code to draw 2D shapes on the canvas.
  - More advanced and complicated WebGL API allows JavaScript code to access GPUs to create 3D graphics.

# Raster vs Vector



Bitmap-depiction of  
the letter "A"



Vector-depiction of  
the letter "A"

