```
d3.csv("preproccessed_data/year_perMonth_FCA.csv").then(finesData => {
  finesData.forEach(d => {
    d.fines = +d.FINES;
    d.month = d["Month (Name)"];
  });

  const margin = { top: 50, right: 30, bottom: 50, left: 70 },
    width = 1000 - margin.left - margin.right,
    height = 400 - margin.top - margin.bottom;

  const svg2 = d3.select(".placeholder-charts-chart")
    .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform", `translate(${margin.left},${margin.top})`);

  const x = d3.scalePoint()
    .domain(finesData.map(d => d.month))
    .range([0, width])
    .padding(0.5);

  const y = d3.scaleLinear()
    .domain([0, d3.max(finesData, d => d.fines)])
    .nice()
    .range([height, 0]);

  svg2.append("g")
    .attr("transform", `translate(0,${height})`)
    .call(d3.axisBottom(x))
    .selectAll("text")
    .attr("transform", "rotate(-40)")
    .style("text-anchor", "end");

  svg2.append("g")
    .call(d3.axisLeft(y).tickFormat(d3.format(".2s")));

  const line = d3.line()
    .x(d => x(d.month))
    .y(d => y(d.fines));

  // Add the line path
  const path = svg2.append("path")
    .datum(finesData)
    .attr("fill", "none")
    .attr("stroke", "#00A6D8")
    .attr("stroke-width", 3)
    .attr("d", line);

  // Create tooltip
  const tooltip = d3.select("body").append("div")
    .attr("class", "tooltip")
    .style("opacity", 0)
    .style("position", "absolute")
    .style("background", "rgba(0, 0, 0, 0.8)")
    .style("color", "white")
    .style("padding", "10px")
    .style("border-radius", "5px")
    .style("pointer-events", "none")
    .style("font-size", "14px")
    .style("z-index", "10");

  const overlay = svg2.append("rect")
    .attr("width", width)
    .attr("height", height)
    .style("fill", "none")
    .style("pointer-events", "all");

  const hoverLine = svg2.append("line")
    .attr("stroke", "#666")
    .attr("stroke-width", 1)
    .attr("stroke-dasharray", "3,3")
    .style("opacity", 0);

  const hoverCircle = svg2.append("circle")
    .attr("r", 6)
    .attr("fill", "#FF8A00")
    .attr("stroke", "#fff")
    .attr("stroke-width", 2)
    .style("opacity", 0);
```

```
// Add data points with enhanced interactivity
const circles = svg2.selectAll(".data-circle")
  .data(finesData)
  .enter()
  .append("circle")
  .attr("class", "data-circle")
  .attr("cx", d => x(d.month))
  .attr("cy", d => y(d.fines))
  .attr("r", 6)
  .attr("fill", "#FF8A00")
  .attr("stroke", "#fff")
  .attr("stroke-width", 1)
  .style("cursor", "pointer")
  .on("mouseover", function(event, d) {
    // Highlight the circle
    d3.select(this)
      .transition()
      .duration(100)
      .attr("r", 10)
      .attr("stroke-width", 2);

    // Show tooltip
    tooltip.transition()
      .duration(200)
      .style("opacity", .9);

    tooltip.html(`
      <strong>${d.month}</strong><br/>
      Fines: ${d3.format(",.0f")(d.fines)}
    `)
      .style("left", (event.pageX + 10) + "px")
      .style("top", (event.pageY - 28) + "px");
  })
  .on("mouseout", function(d) {
    // Reset circle
    d3.select(this)
      .transition()
      .duration(100)
      .attr("r", 6)
      .attr("stroke-width", 1);

    // Hide tooltip
    tooltip.transition()
      .duration(500)
      .style("opacity", 0);
  })
  .on("click", function(event, d) {
    // Add click animation
    d3.select(this)
      .transition()
      .duration(150)
      .attr("r", 8)
      .transition()
      .duration(150)
      .attr("r", 4);

  });

// Mouse move handler for line hover effect
overlay.on("mousemove", function(event) {
  const [mouseX] = d3.pointer(event);

  const x0 = d3.scaleLinear()
    .domain([0, width])
    .range([0, finesData.length - 1]);

  const i = Math.round(x0(mouseX));
  const d = finesData[i];

  if (d) {
    hoverLine
      .attr("x1", x(d.month))
      .attr("x2", x(d.month))
      .attr("y1", 0)
      .attr("y2", height)
      .style("opacity", 0.7);

    hoverCircle
      .attr("cx", x(d.month))
      .attr("cy", y(d.fines))
      .style("opacity", 1);
  }
})
.on("mouseout", function() {
  hoverLine.style("opacity", 0);
```

```
    hoverCircle.style("opacity", 0);
  });

  // Add animated line drawing
  const totalLength = path.node().getTotalLength();

  path
    .attr("stroke-dasharray", totalLength + " " + totalLength)
    .attr("stroke-dashoffset", totalLength)
    .transition()
    .duration(2000)
    .ease(d3.easeLinear)
    .attr("stroke-dashoffset", 0);

  circles
    .style("opacity", 0)
    .transition()
    .duration(100)
    .delay((d, i) => i * 150)
    .style("opacity", 1);

  svg2.append("text")
    .attr("x", width / 2)
    .attr("y", -20)
    .attr("text-anchor", "middle")
    .style("font-size", "18px")
    .style("font-weight", "bold")
    .style("fill", "#333")
    .text("Monthly Fines Over Time");

  // Add legend
  const legend = svg2.append("g")
    .attr("transform", `translate(${width - 100}, 80)`);

  legend.append("circle")
    .attr("cx", 0)
    .attr("cy", 0)
    .attr("r", 4)
    .attr("fill", "#FF8A00");

  legend.append("text")
    .attr("x", 10)
    .attr("y", 0)
    .attr("dy", "0.35em")
    .style("font-size", "12px")
    .text("Monthly Data");
});
```