Lab 5 (5/10)

Submit your team number

Question *Submitted May 10th 2023 at 12:20:29 pm*

Please enter your team number.

12

1. Draw single line chart

In this module, we will build a line chart using d3. A line graph is a graphical representation of information that changes over a period of time.

1. For starters, please download the sales.csv data. This data contains a company's sales, revenue, and expenses per year from 2000 to 2020. We will plot these attributes as a line chart to visualize the growth/decline in the company's performance over the years.

2. Create **lab5.html**. Make sure to import d3.min.js as well.

```
<script src="https://d3js.org/d3.v7.min.js"></script>
```

3. Just as in LE5, create a function for processing the sales.csv dataset by directly iterating over it. All the columns should be read as int/float, which can be done directly by using the + operator. Also, define an SVG container with svgwidth = 500, svgheight = 400, and border = "1px solid rebeccapurple". For this example, we will consider a padding = 50.

Question 1 *Submitted May 10th 2023 at 12:32:16 pm*

Copy your code for the svg container below:

```
// Define SVG parameters
const padding = 50
const width = 500
const height = 400
const borderStyle = "1px solid rebeccapurple"

// Generate SVG
d3.select("body")
    .append("svg")
        .attr("width", width)
        .attr("height", height)
        .style("border", borderStyle)
    .append("g")
        .attr("transform", `translate(${padding}, ${padding})`)
```

Question 2 *Submitted May 10th 2023 at 12:38:28 pm*

1. Next, define xScale and yScale for the line chart using d3.scaleLinear(). We want the year on our x-axis, so for xScale, we use d3.min() and d3.max() for the column "Year" in the data. For y-axis, we will first plot the "Sales" column.

2. Also define the axes below for the chart. Make sure that you translate the x-axis by svgheight-

padding (in y direction) to bring it to the bottom of the SVG container and y-axis by padding (in x-direction).

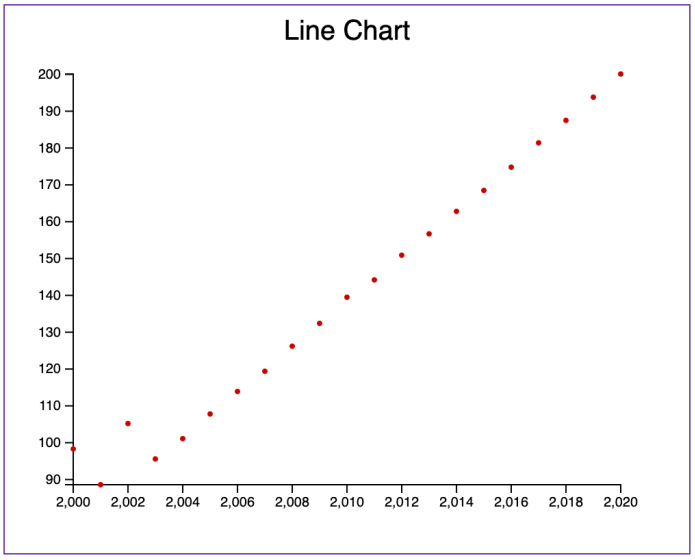
Copy your code for xScale, yScale and axes below:

```
// Create and call x and y scales
xScale = d3.scaleLinear()
    .domain(d3.extent(data, d => d.Year))
    .range([padding, width - padding])
yScale = d3.scaleLinear()
    .domain(d3.extent(data, d=> d.Sales))
    .range([height - padding, padding])
svg.append("g").call(d3.axisBottom(xScale))
    .attr("transform", `translate(0, ${height - padding})`)
svg.append("g").call(d3.axisLeft(yScale))
    .attr("transform", `translate(${padding}, 0)`)
```

Question 3 *Submitted May 10th 2023 at 12:41:42 pm*

We will first build a scatterplot for the datapoint values for "Sales" for each year with color "#CC0000".

The plot should look something like this:



Copy paste you code below:

```
// Plot data points
svg.selectAll(".point")
    .data(data).enter().append("circle")
    .attr("class", "point")
    .attr("cx", d => xScale(d.Year))
    .attr("cy", d => yScale(d.Sales))
    .attr("r", 2)
```

```
        .attr("fill", "red")
```

Question 4 *Submitted May 10th 2023 at 12:46:03 pm*

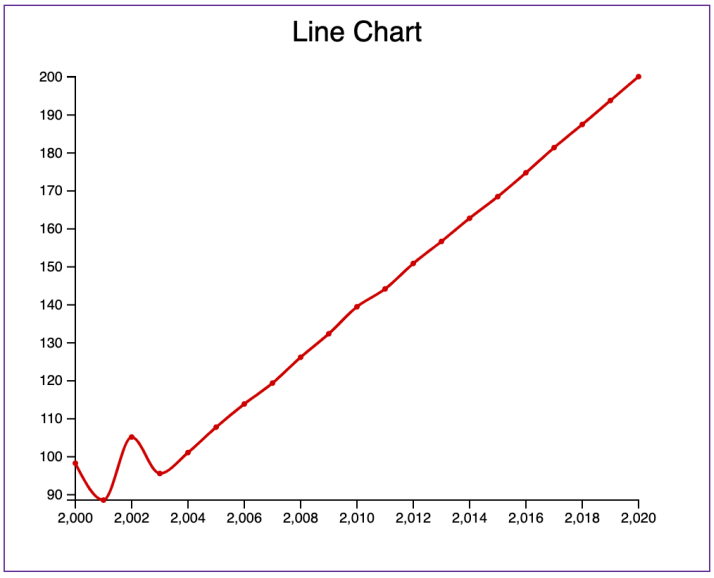
Now, we plot the line joining the scatterplot points. You can use this link as a reference, go to Step 8 to see how we can plot the line chart using d3.line() and SVG path. We will plot Sales with a red color "#CC0000". Copy paste your code below:

```
// Add line chart
svg.append("path")
    .datum(data)
    .attr("d", d3.line()
        .x(d => xScale(d.Year))
        .y(d => yScale(d.Sales))
        .curve(d3.curveMonotoneX)
    )
    .attr("fill", "none")
    .attr("stroke", "#CC0000")
    .attr("stroke-width", 1)
```

Question 5 *Submitted May 10th 2023 at 12:49:16 pm*

Lastly, we can add the plot title using svg.append('text'). Think about how you can correctly align the text if we want it on top of the graph within our padded region (centrally aligned). You can read more about adding plot titles here. Use a new font, which you haven't previously used and set the font size to be 20.

The final chart should look something like this:




```
// Add title
svg.append("text")
    .attr("x", width/2)
```

```
        .attr("y", padding/2)
        .attr("text-anchor", "middle")
        .style("font-size", "20px")
        .style("font-family", "Helvetica")
        .style("font-weight", "bold")
        .text("Line Chart")
```

2. Multiple Line Charts

We will now plot "Revenue" and "Expenses" on the same line chart.

The first thing that we need to do now is to update the domain of yScale to be from the minimum value of all three columns ("Sales", "Revenue" and "Expenses") to the maximum value of these columns. We will use Math.min and Math.max for this.

Question 1 *Submitted May 10th 2023 at 1:12:41 pm*

Copy and paste your code for the updated yScale below:

```
// Update and recall y scale
const allCols = [
    ...d3.extent(data, d => d.Sales),
    ...d3.extent(data, d => d.Revenue),
    ...d3.extent(data, d => d.Expenses)
]
yScale = d3.scaleLinear()
    .domain(d3.extent(allCols))
    .range([height - padding, padding])
d3.select(".yScale").call(d3.axisLeft(yScale))
    .attr("transform", `translate(${padding}, 0)`)
```

Question 2 *Submitted May 10th 2023 at 1:15:30 pm*

Now that we have updated the domain for the y-axis, we will make the scatter plot for "Revenue" in the same way as we did for "Sales".

Next, we'll make the line plot connecting these points in the same way as we did in Slide 1 for "Sales". Set a different color than in Slide 1 for the scatter plot and the line chart, let's say "#00CC00".

Copy paste your code below:

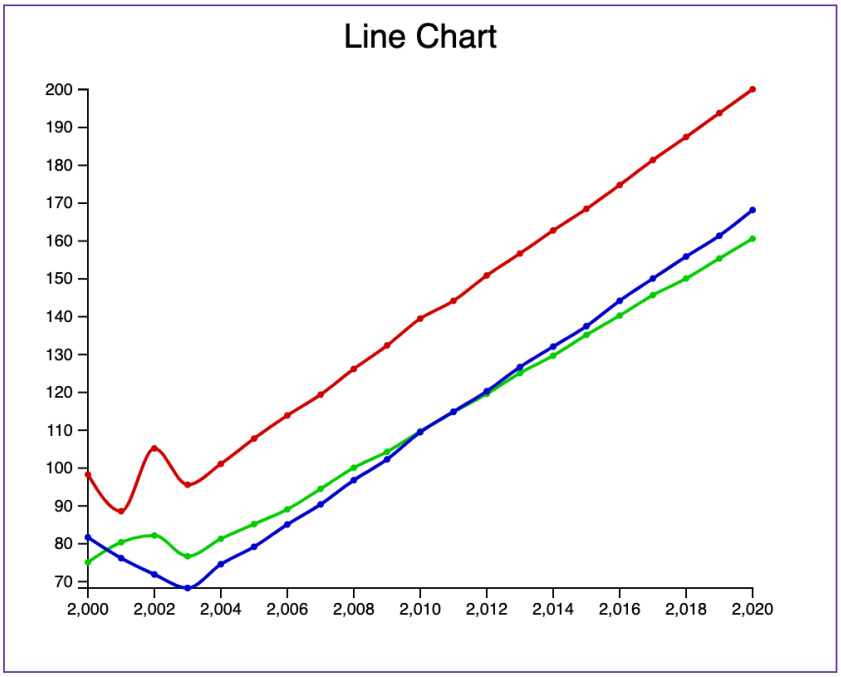
```
// Plot revenue data points
svg.selectAll(".pointSales")
    .data(data).enter().append("circle")
    .attr("class", "point")
    .attr("cx", d => xScale(d.Year))
    .attr("cy", d => yScale(d.Revenue))
    .attr("r", 2)
    .attr("fill", "#00CC00")

// Add revenue line chart
svg.append("path")
```

```
    .datum(data)
    .attr("d", d3.line()
        .x(d => xScale(d.Year))
        .y(d => yScale(d.Revenue))
        .curve(d3.curveMonotoneX)
    )
    .attr("fill", "none")
    .attr("stroke", "#00CC00")
    .attr("stroke-width", 1)
```

Question 3 *Submitted May 10th 2023 at 1:16:25 pm*

Repeat the same thing for "Expenses". Set a different color for "Expenses", let's say "#0000CC".

The final plot should look something like this:



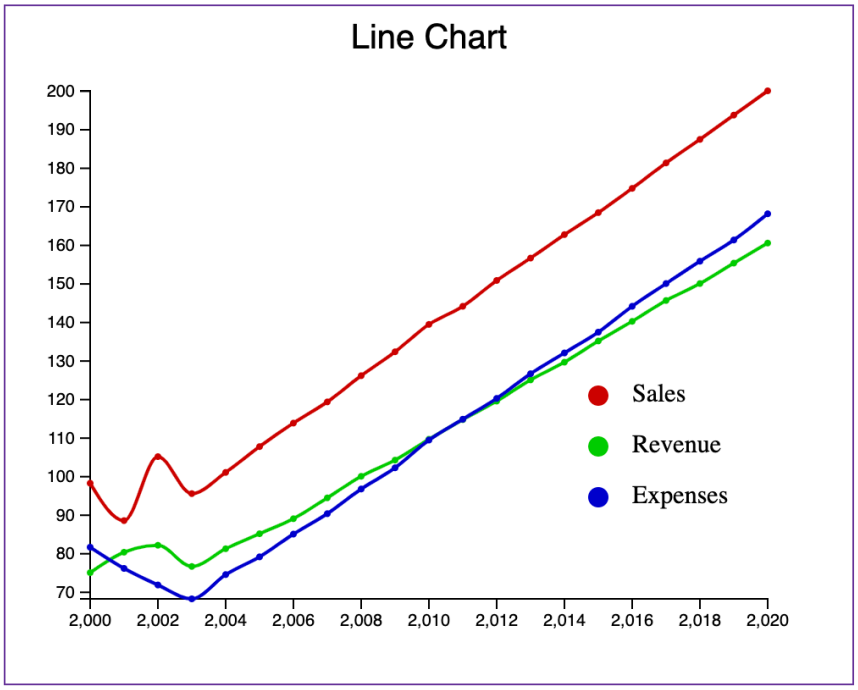
Copy and paste your code below:

```
// Plot expenses data points
svg.selectAll(".pointSales")
    .data(data).enter().append("circle")
    .attr("class", "point")
    .attr("cx", d => xScale(d.Year))
    .attr("cy", d => yScale(d.Expenses))
    .attr("r", 2)
    .attr("fill", "#0000CC")

// Add expenses line chart
svg.append("path")
    .datum(data)
    .attr("d", d3.line()
```

```
            .x(d => xScale(d.Year))
            .y(d => yScale(d.Expenses))
            .curve(d3.curveMonotoneX)
        )
        .attr("fill", "none")
        .attr("stroke", "#0000CC")
        .attr("stroke-width", 1)
```

Question 4 *Submitted May 10th 2023 at 1:54:51 pm*

Lastly, we will add a legend for the graph. We can add the legend manually, and you can get a reference from this link. You can add the legend in the bottom right corner. The final chart should look something like this. Don't worry too much about the exact position of the legend. A slight variation in the position is allowed.



Copy and paste your code below:

```
svg.append("circle").attr("cx",width - 100).attr("cy",height - 150).attr("r", 4).style("fill",
    svg.append("circle").attr("cx",width - 100).attr("cy",height - 130).attr("r", 4).style("fil
    svg.append("circle").attr("cx",width - 100).attr("cy",height - 110).attr("r", 4).style("fil
    svg.append("text").attr("x", width-90).attr("y", height-150).text("Sales").style("font-size
    svg.append("text").attr("x", width-90).attr("y", height-130).text("Revenue").style("font-s
    svg.append("text").attr("x", width-90).attr("y", height-110).text("Expenses").style("font-s
```

3. Changing styles for line charts

We will now change the style of the line chart from solid to a dashed line and try to add some animations.

First, create a new file style.css in the same directory as your lab5.html. Remember to link your html and css files using the following command:

```
<link rel="stylesheet" href="styles.css">
```

Question 1 *Submitted May 10th 2023 at 2:00:36 pm*

We will first add stroke-dasharray in the css file for the particular class (line_sales) that we want to convert from a solid line to a dashed line. You can use this link as reference to learn more about stroke-dasharray. For this exercise, we'll make the sales curve (Red) to dashed by setting the stroke-dasharray = 10. Copy-paste your code that you added to styles.css.

```
#pathSales {
    stroke-dasharray: 10;
}
```

Question 2 *Submitted May 10th 2023 at 2:01:58 pm*

Now, we will try to add some animation to the "Revenue" curve. We first make "Revenue" curve a dashed line by using stroke-dasharray = 10 for it just like the previous question. Then we want the curve to show a linearly moving animation for 10s. You can refer this link for more information about animations and how you can draw cool stuff with it.

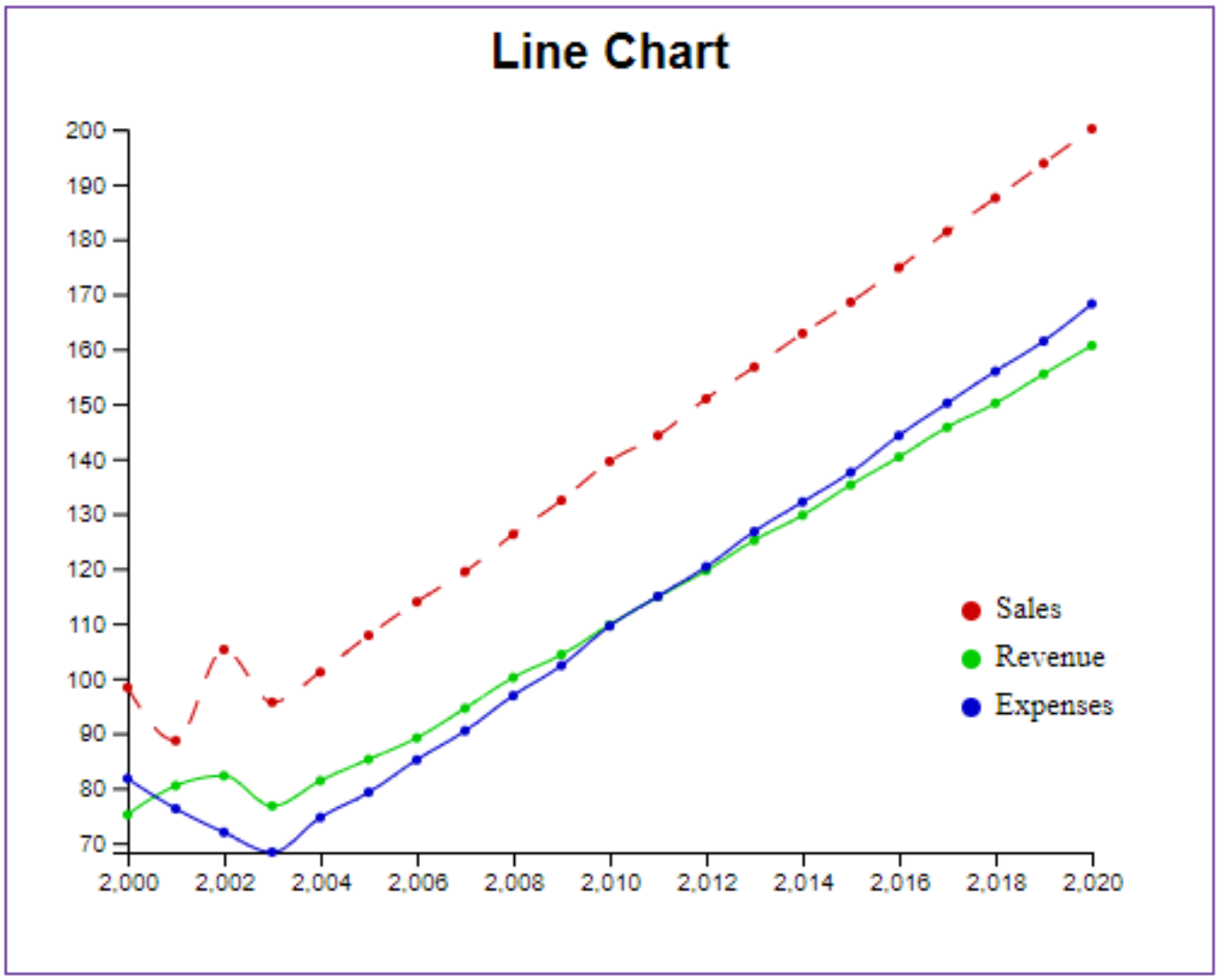
Copy paste your code to add animation to the "Revenue" curve below from the css file.

```
#pathRevenue {
    stroke-dasharray: 1000;
    stroke-dashoffset: 1000;
    animation: dash 5s linear forwards;
}

@keyframes dash {
    to {
        stroke-dashoffset: 0;
    }
}
```

Upload Your Files

Question 1 *Submitted May 10th 2023 at 2:02:47 pm*

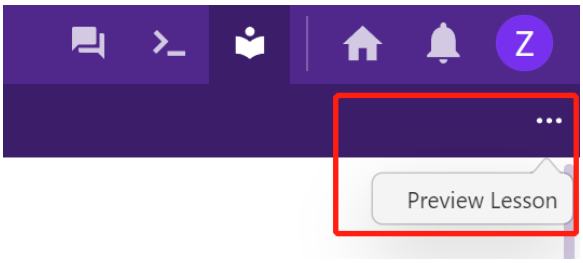
Upload the screenshot of your resulting webpage. You will need to click the "clip" button to upload a file into the Answer box.



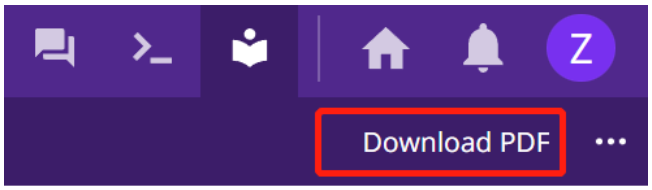
Question 2 *Submitted May 10th 2023 at 2:02:49 pm*

You need to download the PDF of lecture exercise 3 and upload it with other files to the Gradescope. Follow the instructions on how to download PDF file:

1. Click on the ellipsis button and the Preview Lesson.



2. After that, click on the Download PDF button.



- [x] PDF downloaded!
- [ ] Haven't done yet!

Question 3 *Submitted May 10th 2023 at 2:02:52 pm*

Upload the following files to Gradescope. You need to make **a group submission,adding all present members in your team**, so that the present members get the participation credit.

Files to upload:

- lab5.html
- PDF you downloaded as Q2

- [x] Our team uploaded the the files on gradescope!
- [ ] Oops, our team did not upload the files on gradescope!

Feedback

Question *Submitted May 10th 2023 at 2:02:56 pm*

Was the activity today clear? If not, please share how the course can improve it. Your comments will help us design future lab content (and also future students).

y