**DSDV Lab Mid Note**

**1.** **(35pts)** Analyze a visualization. Analyze and evaluate the visualization in Figure 1 by answering the following questions:

a. (5pts) What is the goal of the visualization?

b. (10pts) What is its data? types of data?

c. (10pts) How did the author use marks and channels to encode the data?

d. (10pts) The design is not good, redesign it.

* + 1. A chart with different colored squares

       Description automatically generated
    2. **(STUDENT ANSWER 1** 35/35**)**

**A,** The goal of the visualization is to show the probability of near interaction between patients of COVID-19 and the stewardess

**B,** +, The data is the figures of the probability (%%), which is integer and is shown in terms of color scale.

+, Type of data: attributes

**C,** The author has transmitted the original data (which is integer) into the darkness of color shown in the visualization That means the low figures will be represented by light color ( this case is light blue. Medium figures are shown in white and the highest are shown in the darkest color ( this case is dark pink..).

**D,** The color scale is not good This is how I redesign it. The smallest percentage should be shown in the lightest color white. The bigger the percentage is the ‘more blue’ the color gets. The color continues to be darker and darker until it reaches ‘dark blue' at the end of the scale.

* + 1. **(STUDENT ANSWER 2** 35/35**)**

**a)** This visualization is to show the rate of flight attendants and have close-range contact with a covid patient.

**b)** This is a type of ratio, where the auther get the rate of close contact between a flight attendants and covid patients on a plane.

**c)** In this visualization, the author used squares as a mark on a background of real seats on a plane. For the channels, author used hue and saturation shown below: dark blue - > light blue -> white light -> pink dark pink

**d)** As using 2 different hues, this visualization is a little bit harder to read by using only 1 hue (which I recommend blue as it's cooler and easier to get the change) with a wider range of saturation, this visualization may be better to get the in sight.

1. **(GPT ANSWER)**
   * 1. **a.** **Goal of the Visualization**: The goal of the visualization is to represent the probability of infection when individuals are in close range with a sick person. This is indicated by the color gradient on an airplane seating chart.
     2. **b. Data and Types of Data:** The data consists of the probability percentages of passengers getting infected when they are near a sick person. The types of data are categorical (seat labels and indications for sick person and flight attendant) and quantitative (probability percentages).
     3. **c. Use of Marks and Channels:** The author used color as a channel to encode the data, with different shades representing different probabilities of infection. Darker colors indicate higher probabilities. Marks are used to represent seats, with special icons indicating the seat occupied by a sick person and flight attendants.
     4. **d. Redesign:** To redesign it, I would suggest the following improvements:

* Use clearer labels for rows and columns.
* Make sure that the color gradient is easily distinguishable for all viewers including those with color vision deficiencies.
* Provide clear legends that explain what each mark and color represents.
* Instead of using color alone to represent the probability of infection, consider using a combination of color and size. For example, seats with a higher probability of infection could be represented by larger, darker circles, while seats with a lower probability could be represented by smaller, lighter circles. This would make the visualization more accessible to a wider audience, including those with color vision deficiencies.
* Consider adding interactive elements to the visualization. For example, when a user hovers over a seat, a tooltip could appear showing the exact probability of infection for that seat. This would provide more precise information to the user.
* Finally, ensure that the visualization is responsive so that it can be viewed on devices of different sizes. This could be achieved by using a flexible layout and scalable vector graphics (SVGs).

1. 2. **(35pts)** Design a visualization to show students in a class.
2. Each student has a GPA (10pts), a major (10pts), a group’s ID (10pts), and a seat number.
3. Show all attributes of each student and show links between all members of a group in the classroom arrangement given in Figure 2.
4. A table of numbers and a number

   Description automatically generated with medium confidence
5. A screenshot of a math arrangement

   Description automatically generated
   * 1. **(STUDENT ANSWER 1** 35/35**)**
     2. I will use color, position and interactive visualization for this dataset
     3. +, 6 colors are used to represent 6 group ID

|  |  |
| --- | --- |
| * + 1. **Group ID** | * + 1. **color** |
| * + 1. 1 | * + 1. red |
| * + 1. 2 | * + 1. orange |
| * + 1. 3 | * + 1. green |
| * + 1. 4 | * + 1. blue |
| * + 1. 5 | * + 1. yellow |
| * + 1. 6 | * + 1. purple |

* + 1. +, 3 types of shape are used to represent 3 major

|  |  |
| --- | --- |
| * + 1. **Major** | * + 1. **Shape** |
| * + 1. IT | * + 1. rectangle |
| * + 1. CS | * + 1. triagle |
| * + 1. DS | * + 1. circle |

* + 1. +, The shapes are placed exactly at the seat of student
    2. +, Finally, the size of the shape will represent the GPA of each student. The smallest size stands for 60 and the biggest will be 100
    3. +, Any time the printer touch the shape (rectangle, triangle cude), the original figures will be shower off. They will consist of:
    4. -> Group ID
    5. -> Major (interactive visualization)
    6. -> GPA
    7. **(STUDENT ANSWER 2** 34/35**)**
    8. For showing the student's data in a class, we will use the position in the class for the "Seat" column, which is shown in Figure 2. Therefore, we only need to deal with 3 columns " GPA", "Major" and "GroupID."

- The most important column to visualize is Major in my opinion, so I will deal with it first.

* + 1. - For “Major”, I will use different hues for different major. For example, red for "IT", blue for "CS", and green for "DS" (all of them are in the same level of saturation, but different in hue)
    2. - For "GPA", I will put this column to the second level of priority. I will use the saturation level to distinguish between students with high GPA with students with low GPA.
    3. - Finally, a number in the middle of the box will show the group that this student belongs to. In this case, in order to keep the position number, I will move them to the top of each box.
    4. **Conclusion:** by using 3 channels: position, hue, saturation and a number, I will visualize the given data into a class diagram

\* For the links between members of a group: all members of a group will have the same number in the middle of the box as shown in the first part, which can be easily distinguished.....

3. (30pts) Create an HTML web page with the title “Midterm”. a. Get data from <https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv> (in your code, you can refer to this source as “covid\_global.csv”)

In the dataset,

- “Province/State” and “Country/Region” are used as the key for each row.

* 1. - The date is in US-format

Figure 3 is a sample from the dataset.

A screenshot of a spreadsheet

Description automatically generated

*Figure 3 - A sample from the dataset*

b. (5pts) Write code to draw a horizontal bar chart to show COVID confirmed cases over the world on “04/04/2022”. The chart must

R1. (5pts) have a fixed size (use scale to convert data)

R2. (5pts) have an axis with a title and ticks

R3. (5pts) use Province/State and Country/Region as key/label for a row

R4. (5pts) show only non-zero rows (Use filter function of arrays in javascript)

R5. (5pts) show value in the bar

**Hint:**

- Use rowConverter

- Use parseInt, parseFloat to convert strings to numbers

- Filter function of arrays in javascript.

newDataSet = dataset.filter(d => d[“04/04/2022”] > 0);

<!DOCTYPE *html*>

<html *lang*="en">

<head>

  <meta *charset*="UTF-8">

  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

  <title>Midterm</title>

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

</head>

<body>

  <div *id*="chart"></div>

  <script>

    d3.csv("time\_series\_covid19\_confirmed\_global.csv", rowConverter)

      .then(*data* => {

*data* = *data*.filter(*row* => *row*.cases !== 0);

        drawChart(*data*);

      });

    function rowConverter(*row*) {

      return {

        place: *row*['Province/State'] + ', ' + *row*['Country/Region'],

        cases: +*row*['4/4/22']

      };

    }

    function drawChart(*data*) {

      const svgWidth = 1500;

      const svgHeight = 10000;

      const margin = { top: 10, right: 20, bottom: 50, left: 300 };

      const svg = d3.select("#chart")

        .append("svg")

        .attr("width", svgWidth)

        .attr("height", svgHeight)

      const xScale = d3.scaleLinear()

        .domain([0, d3.max(*data*, *d* => *d*.cases)])

        .range([margin.left, svgWidth - margin.right - 50]);

      const yScale = d3.scaleBand()

        .domain(*data*.map(*d* => *d*.place))

        .range([margin.top, svgHeight - margin.bottom])

        .padding(0.1);

      const colorScale = d3.scaleSequential()

        .domain([0, d3.max(*data*, *d* => *d*.cases)])

        .interpolator(d3.interpolateYlOrRd);

      const xAxis = d3.axisBottom(xScale);

      const yAxis = d3.axisLeft(yScale);

      svg.append("g")

        .attr("transform", `translate(5, ${svgHeight - margin.bottom + 5})`)

        .call(xAxis)

        .append("text")

        .attr("x", svgWidth - margin.right - 50)

        .attr("y", - 30)

        .attr("font-size", "30px")

        .attr("font-weight", "bold")

        .text("Cases")

        .attr("fill", "black");

      svg.append("g")

        .attr("transform", `translate(${margin.left}, 0)`)

        .call(yAxis)

        .append("text")

        .attr("x", - 100)

        .attr("y", 100)

        .attr("font-size", "30px")

        .attr("font-weight", "bold")

        .text("Place")

        .attr("fill", "black");

      svg.selectAll("rect")

        .data(*data*)

        .enter()

        .append("rect")

        .attr("x", margin.left + 5)

        .attr("y", (*d*, *i*) => yScale(*d*.place))

        .attr("width", 0)

        .transition()

        .duration(700)

        .attr("width", (*d*) => xScale(*d*.cases) - margin.left)

        .attr("height", yScale.bandwidth())

        .attr("fill", (*d*) => colorScale(*d*.cases));

      svg.selectAll(".label")

        .data(*data*)

        .enter()

        .append("text")

        .attr("x", (*d*) => xScale(*d*.cases) + 10)

        .attr("y", (*d*, *i*) => yScale(*d*.place) + yScale.bandwidth() / 2 + 5)

        .attr("font-family", "sans-serif")

        .attr("font-size", "11px")

        .attr("fill", "white")

        .text((*d*) => *d*.cases)

        .transition()

        .delay(500)

        .attr("fill", "black");

    }

  </script>

</body>

</html>