# NM2207

Session 07 Challenges

Before attempting the challenges, you are expected to have watched and coded along with the Lecture videos. A tutorial is meant to practice the skills presented in the video lecture, and show you more applications of it. Tutors will explain the challenges and answer the questions you may have**.**

**The challenges are due to be completed at the end of class each week for full credit which is also attendance. Submitting by midnight of the same day accounts for half the credit.**

# Overview of what we will do today:

* Creating a bar chart
* Importing data

**Part 1**

**Revision of concepts Chapter 2 and 3 of Learn Chartjs**

* Objects
* Arrays
* Canvas

**Objects**

Objects are powerful data structures. They represent an unordered collection of data. In JavaScript, they are stored as key-value pairs within curly braces separated by commas. The key-value pairs can be a variety of data types. They can be integers, strings, arrays, functions, or even objects themselves.

const person = {

  firstName: "tony",

  lastName: "stark",

  age: 35,

};

In this example, the person object has three properties, firstName (string), lastName (string), and age (integer).

**Arrays**

Arrays are the main data structure you use to store one-dimensional data. It is a collection of elements, which are typically of the same data type (although they can be different in JavaScript). Arrays can be looped over, sorted, modified, among other things. They can also be multidimensional, where an array contains other arrays, similar to rows and columns in a data table.

const prime = [2, 3, 5, 7, 11];

const points = [[200,300], [150,100], [100,300]];

In this example, the variable prime is an array of the first 5 prime numbers, in ascending order. The variable points is a multidimensional array containing 3 different coordinates which are also arrays, wherein the first element represents the x-value and the second element represents the y-value of the coordinate.

**Canvas API**

There are two main methods of drawing graphics in HTML – SVG and Canvas. SVG is used more for drawing vectors and shapes (such as circles, boxes, logos) while the HTML Canvas allows for more complex objects to be drawn using JavaScript. You can read more about the difference between the two [here](https://www.tutorialspoint.com/What-is-the-difference-between-SVG-and-HTML5-Canvas).

Chart.js uses the HTML Canvas API to draw its graphics. To draw using Canvas, all you need is a <canvas> element somewhere in your HTML code.

<div id="canvas-container">

    <canvas id="canvas" width="400" height="400"></canvas>

</div>

In your JavaScript file, grab the canvas element from the DOM, get a 2D graphics context from the object, and start drawing.

const canvas = document.getElementById("canvas");

const ctx = canvas.getContext("2d");

ctx.fillStyle = "red";

ctx.fillRect(50,50,50,50);

The code above draws a red rectangle on your HTML canvas element.

When you are using Chart.js, you can simply pass the canvas element to a new Chart, and Chart.js will render its charts based on that.

new Chart("canvas", {

  type: "bar",

  data: {},

  options: {

    maintainAspectRatio: false,

    legend: {

      display: false,

    },

    title: {

      display: true,

      text: "",

    },

  },

});

**Part 2**

This week, we will show you how you can also use data from the tables and figures of published papers for your final project.

**Summary of learnings (Chapter 3 of Learn Chartjs)**

* Creating a bar chart

In this session, you will learn how to create a bar chart. The data has already been provided for you, corresponding to the likelihood of deepfake sharing for older people across various countries.

In your Session 7 challenges folder, open sharing\_deepfakes\_barChart.html.

Right now, you should be presented with an empty graph that looks like this.

Table

Description automatically generated

By the end of this session, you should aim to achieve something like this.

Chart, waterfall chart, box and whisker chart

Description automatically generated

Alright, let’s get started.

1. Add data to bar chart code. Find it in resources/part2.txt

Open part1.txt in the resources folder and copy the three data variables into the top of dfChart.js. Pause and do a quick recap – what are the data structures and variable types of these three different variables?

1. Configuring the dataset object

Next, we want to configure the correct dataset object for ChartJS. This can be found in the dataObj variable.

The data object for a bar chart in ChartJS contains two main properties – labels and datasets.

* labels requires an array of variables that corresponds to your x-axis, which are the various categories of your data. In this case, your labels represent the different countries. Uncomment the labels property and assign it to labelCountry.
* datasets requires an array of objects that correspond to the data for each label. Let’s have a look at what an object in the datasets property might look like. Copy the values below into your code.

{

      label: "Age", // this property is just a semantic naming of your y-axis.

      data: beingOld, // this property contains an array of values that corresponds to your x-axis labels.

      borderWidth: 2, // the border width of your bars

      backgroundColor: "hsla(20,100%,80%,0.8)", // the background color of your bars

      borderColor: "hsla(0,100%,50%,1)", // the border color of your bars

}

* Which do you think is the most important property here? Discuss with the person beside you.
* There are a lot more properties that can go into a dataset object. If you want to find out more, have a look at the Chart.js documentation under “Dataset Properties”. (<https://www.chartjs.org/docs/latest/charts/bar.html>)

Finally, let’s go ahead and add a title to your chart. You can set this through the text field on the title object in your Chart. This text field can either be a string or an array of strings. If you were to use the latter, the title will render on multiple lines.

Currently, the text field is an array with an empty string. Change the array to include a proper title. We’re using an array of 2 strings here so it looks neater.

text: ['Predicting likelihood of deepfake sharing','for Older People']

Save your changes and refresh your html page. You should have a nice-looking bar chart!

**Part 3**

**Summary of learnings**

* Changing the column being read
* Add the correct labels

Now that you understand how data should be provided to Chart.js, try playing around with the charts! Let’s start with changing our y-axis. Previously, we wanted to display the likelihood of deepfake sharing across 8 different countries after controlling for age.

Now, let’s change the prediction to control for gender (male). We already have the beingMale variable, which is an array that contains our data. So all we need to do is change our datasets property.

In your dataset object, replace the label and data properties with the new values.

{

      label: "Male", // this property is just a semantic naming of your y-axis.

      data: beingMale, // this property contains an array of values that corresponds to your x-axis labels.

      borderWidth: 2, // the border width of your bars

      backgroundColor: "hsla(20,100%,80%,0.8)", // the background color of your bars

      borderColor: "hsla(0,100%,50%,1)", // the border color of your bars

    }

Don’t forget to change the title of your bar chart to the appropriate title!

text: ['Predicting likelihood of deepfake sharing','for Males']

Refresh your bar chart. How has the chart and data changed?

**Final Challenge**

Take a look at the original data under resources/sharing\_deepfake\_results.png.

We’ve used the Age and Male variables to construct a bar chart.

Following the steps above, try using one of the other control variables (e.g., education, income) to create **your own bar chart**! Upload the html file to your NM2207 homepage and show your tutors before the end of class.

**Reflection**

* The data is about the demographic and trait features that predict deepfake sharing behavior.
* Based on the results image in the resources folder, what do you understand about age and gender in terms of how it predicts deepfake sharing behavior in different countries? Does this surprise you? Why/why not?
* Based on the results image, what do you understand about income in the same context? Is this surprising? Why/why not?
* Does this give you more ideas for the kind of data story you could make? What are some topics you find interesting, and you’d like to explore through plots? Write a few sentences in the html body of your page.