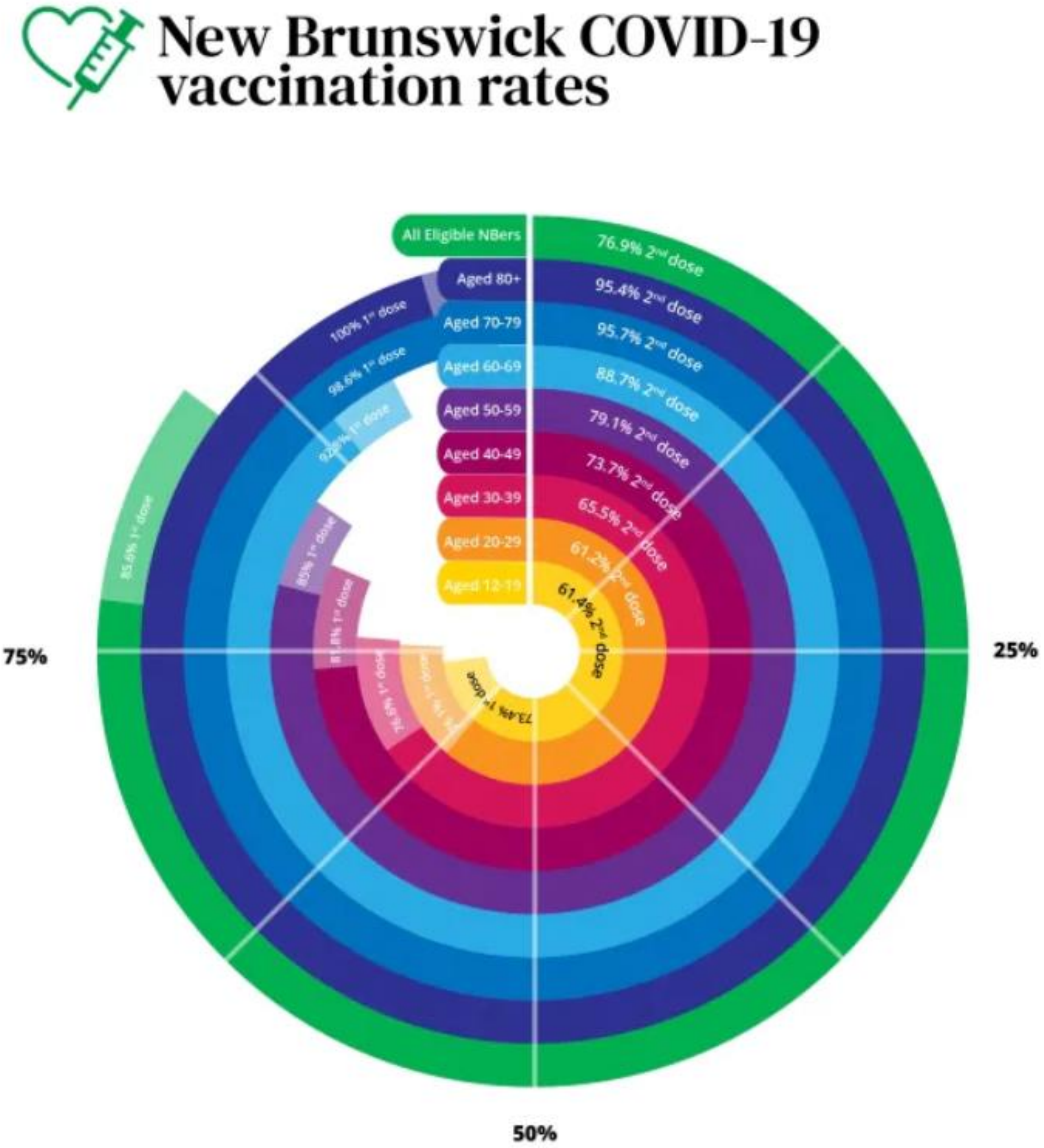
Data Visualization: Assignment 1

Student 1: Sofya Aksenyuk, 150284

Student 2: Uladzimir Ivashka, 150281

Original image



URL: https://www.cbc.ca/amp/1.6173430

Context: New Brunswick COVID-19 vaccination rates.

Story behind: New Brunswick has recorded a big bump in positive COVID-19 cases related to Campbellton and Moncton clusters, with multiple schools and daycares closed on Monday.

Faults:

- Incorrect chart type choice
- Difficulties in presented data comparison
- Unproportional scaling (caused by data types (first and second dose) combining)
- Need of the chart rotation to read data

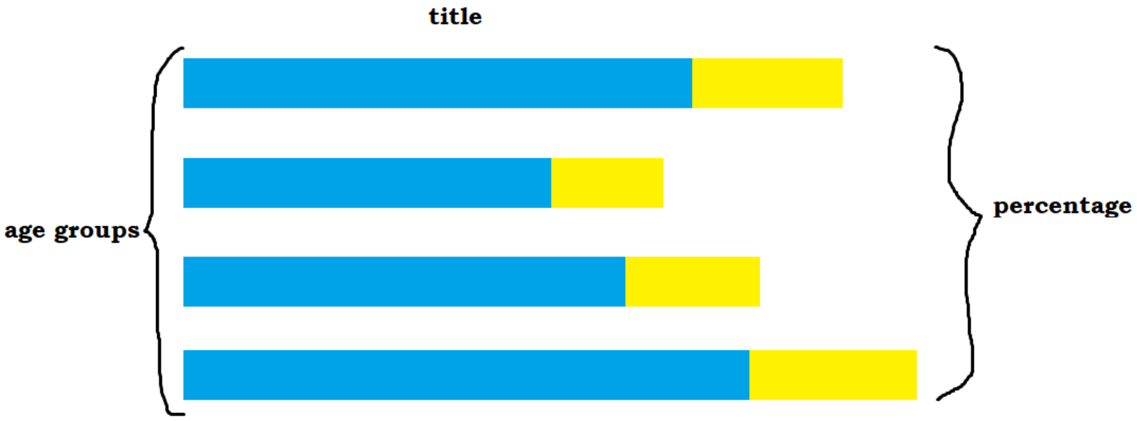
Implications:

- Inaccurate understanding of the data
- Unusefulness of such data visualization
- Incorrect insights

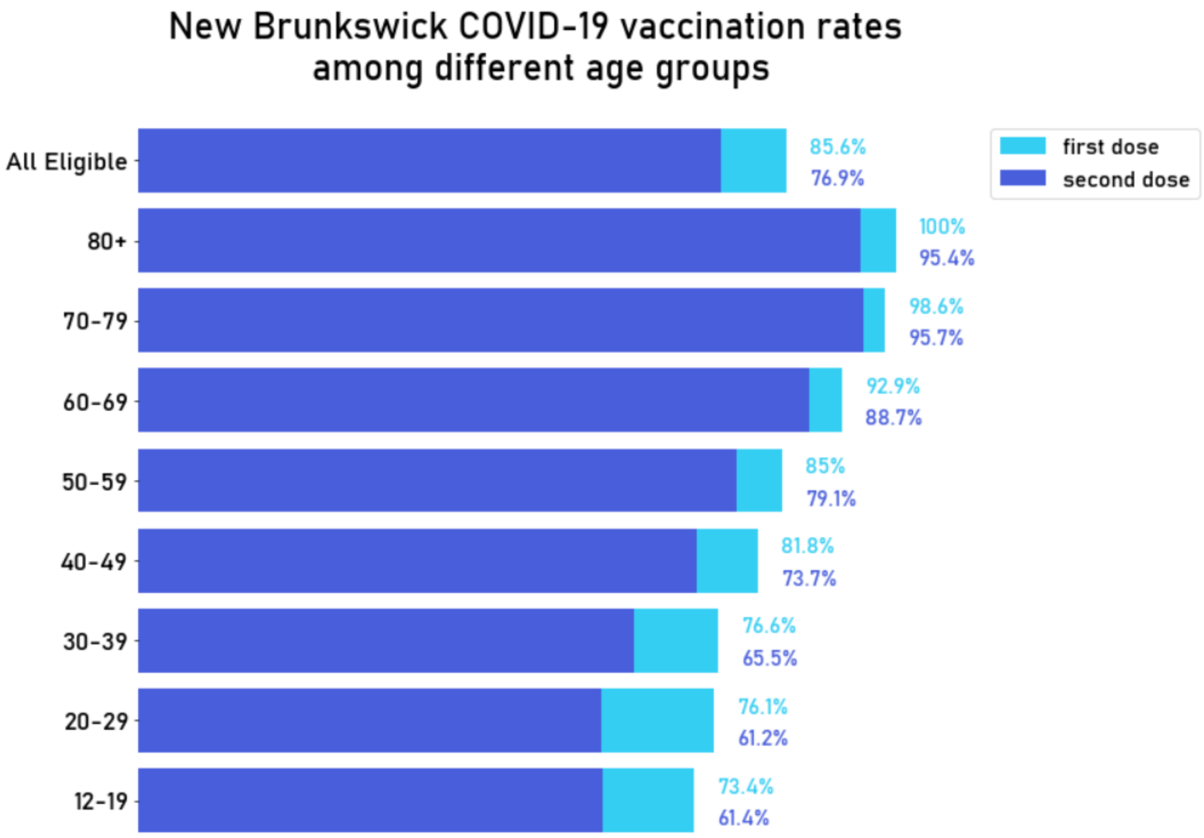
Possible improvements:

- Choosing different chart type (e.g., bar chart)
- Highlighting the group with the highest and lowest values
- Separation of data types (first and second dose)

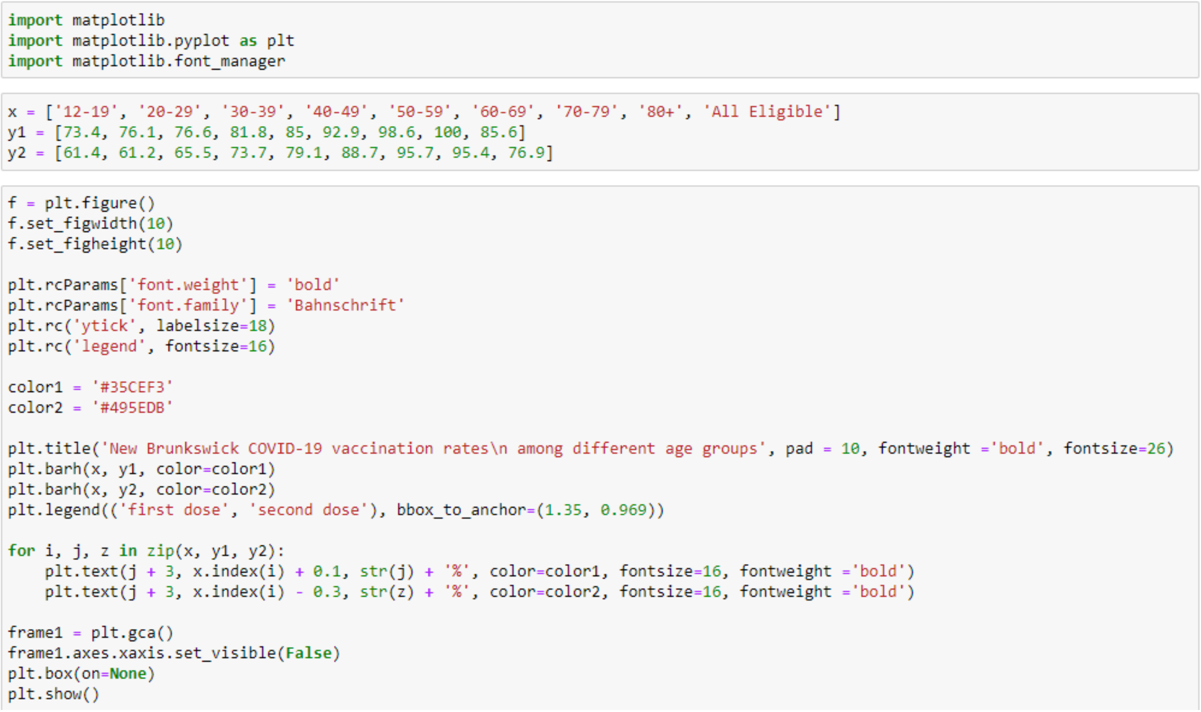
Sketch



Implementation



Implementation code



```
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.font_manager
```

```
x = ['12-19', '20-29', '30-39', '40-49', '50-59', '60-69', '70-79', '80+', 'All Eligible']
y1 = [73.4, 76.1, 76.6, 81.8, 85, 92.9, 98.6, 100, 85.6]
y2 = [61.4, 61.2, 65.5, 73.7, 79.1, 88.7, 95.7, 95.4, 76.9]
```

```
f = plt.figure()
f.set_figwidth(10)
f.set_figheight(10)

plt.rcParams['font.weight'] = 'bold'
plt.rcParams['font.family'] = 'Bahnschrift'
plt.rc('ytick', labelsize=18)
plt.rc('legend', fontsize=16)

color1 = '#35CEF3'
color2 = '#495EDB'

plt.title('New Brunkswick COVID-19 vaccination rates\n among different age groups', pad = 10, fontweight ='bold', fontsize=26)
plt.barh(x, y1, color=color1)
plt.barh(x, y2, color=color2)
plt.legend(('first dose', 'second dose'), bbox_to_anchor=(1.35, 0.969))

for i, j, z in zip(x, y1, y2):
    plt.text(j + 3, x.index(i) + 0.1, str(j) + '%', color=color1, fontsize=16, fontweight ='bold')
    plt.text(j + 3, x.index(i) - 0.3, str(z) + '%', color=color2, fontsize=16, fontweight ='bold')

frame1 = plt.gca()
frame1.axes.xaxis.set_visible(False)
plt.box(on=None)
plt.show()
```