Assignment 3: Visualization Using Matplotlib

By Dana DiVincenzo, Pyone Myat Maw, Suhas Nagabhushan, and Chithira Pugazhenthi

Introduction

In this assignment, we were tasked with replicating 2 graphs using Matplotlib. Our goal was to make our graphs look as similar as possible, if not identical, to the original graphs. Links to the original graphs will be at the bottom of the assignment.

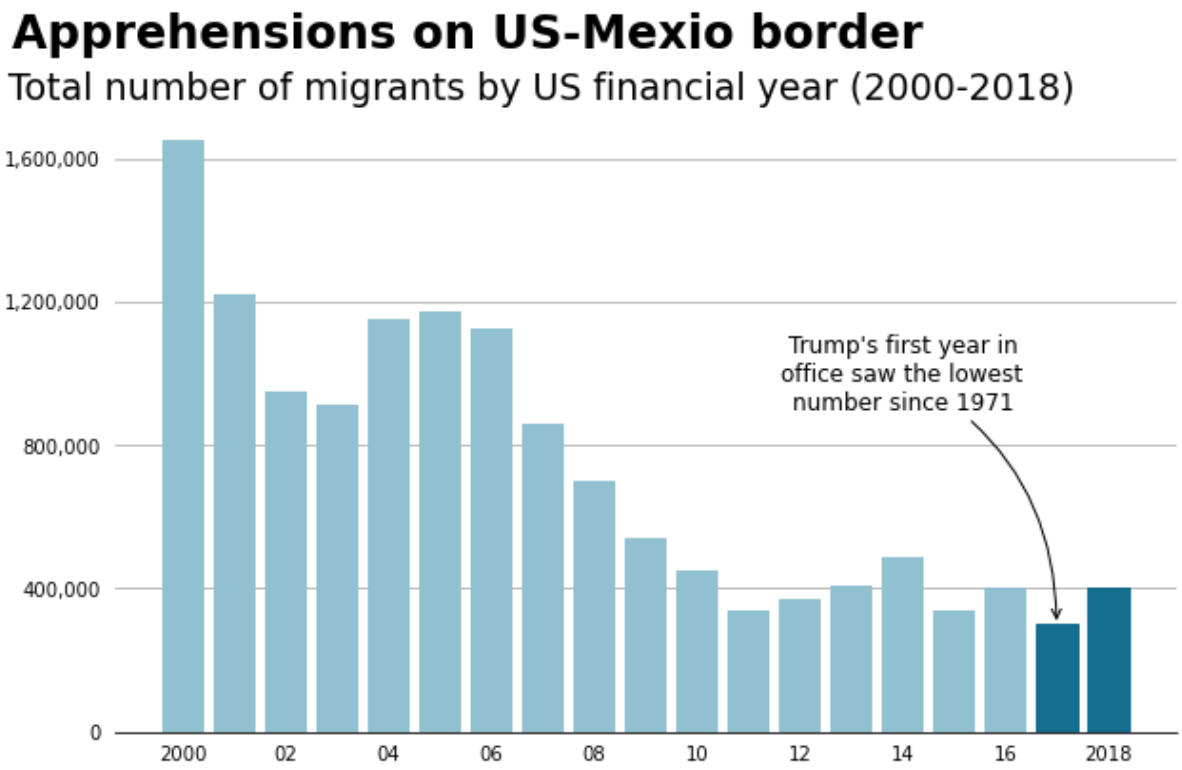
Graph 1

The first graph we were tasked to replicate was a graph showing the number of migrants arrested on the Mexico border from 2000 to 2018. The code to create the graph is below.

In [4]:
```
import matplotlib.pyplot as plt
import matplotlib.ticker as tkr
import numpy as np
import pandas as pd
```

In [5]:
```
%%capture
# Here is the data for the height of the bars in the graph
data = [1650000, 1220000, 950000, 910000, 1150000, 1175000,
        1125000, 860000, 700000, 540000, 450000, 340000,
        370000,410000, 490000, 340000, 400000, 300000, 400000]

# Here is the data for the x axis ticks
ticklabel = ['2000', '02', '04', '06', '08', '10', '12', '14', '16', '2018']
tickindex = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

In [6]:
```
fig, ax = plt.subplots()
x = np.arange(19)

# Changing the color of the bars to make it match the graph perfectly, colors
 were found using a color code lookup website
bars = ax.bar(x, data, color = "#91c1d0", zorder = 3)
bars[-2].set_color('#156d90')
bars[-1].set_color('#156d90')
plt.xticks(np.array(tickindex), ticklabel, fontsize = 10, ha = 'center')

# Setting y tick labels to increase by 400 thousand
plt.yticks(np.array(range(0, 1600001, 400000)))

ax.yaxis.set_major_formatter(
        tkr.FuncFormatter(lambda y,  p: format(int(y), ',')))

# Adding a title and subtitle, and changing font, boldness, and alignment to m
atch the graph
plt.suptitle("Apprehensions on US-Mexio border", fontweight = "bold", fontsize
= 24, ha = "left", x = 0.05, y = 1)
plt.title("Total number of migrants by US financial year (2000-2018)", fontsiz
e = 19, ha = "left", x = -.1)

# Removes the left, top, and right spines
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)

# Removes the ticks but keep the tick lables
ax.tick_params(left=False, bottom=False)

# Makes horizontal lines at each y tick label and sets them to go behind the b
ars
plt.grid(axis = "y", zorder = 0)

# Creates floating text within the graph
# Creates a curved arrow which points from the text to the 2017 bar
ax.annotate('Trump\'s first year in\noffice saw the lowest\nnumber since 1971'
, xy=(17, 300000), fontsize = 12, \
            ha = 'center', xytext=(14, 900000), \
            arrowprops = dict(arrowstyle='->', connectionstyle='arc3,rad=-0.3'
))
# Enlarges the graph
fig.set_size_inches(10,6)

plt.show()
```



Summary of Graph 1:

The bar graph shows the fluctuations of the number of migrants apprehended at the US southwest (US-Mexico) border for the last 18 years (2000-2018). In 2017, which was the first year of President Trump in office, the migrant's number dropped significantly to 303,916. A little rise can be seen in 2018 reaching to almost 400,000 but it is not as noteworthy as the surges of 2004, 2005 and 2006. After 2006, the number of migrants was mostly likely towards sinking. The highest peak was over 1,600,000 in 2000, and almost a 25% fall in next year, 2001. Overall, the graph represents the plummeting trend in the number of people arrested in the last 18 years.

Shortfalls of Graph 1:

1. No labels on the axes makes it more confusing for the reader to interpret the graph.
2. Bar charts are not the right choice for showing the trend over time, a line chart would be better.
3. Information such as percentages of the fluctuations of apprehended migrants is not shown, and would need to be done by the audience. The information given by this bar graph seems limited.
4. No quantitative values are shown on the bars and only one measurement type is displayed on the graph.
5. The unnecessary information like grids in the graph may not be easier for the audience to read (Chartjunk).

Suggested Improvements to Graph 1:

1. Labelling axes clearly gives the user a better understanding of the graph.
2. Converting the bar graphs into line charts will give a better visualization to understand how the trend of apprehension rate changed over the past 18 years.
3. Comparing two measurement types (e.g: apprehension rate vs the trend of the security level at border) that are correlated would give a better insight.
4. Removing unnecessary information like grids in the chart would provide clearer visualization (Chartjunk).
5. Adding trend line to the graph to depict the future prediction.
6. Adding source of the data.

Tabular summary of Matplotlib functions used to create Graph 1:

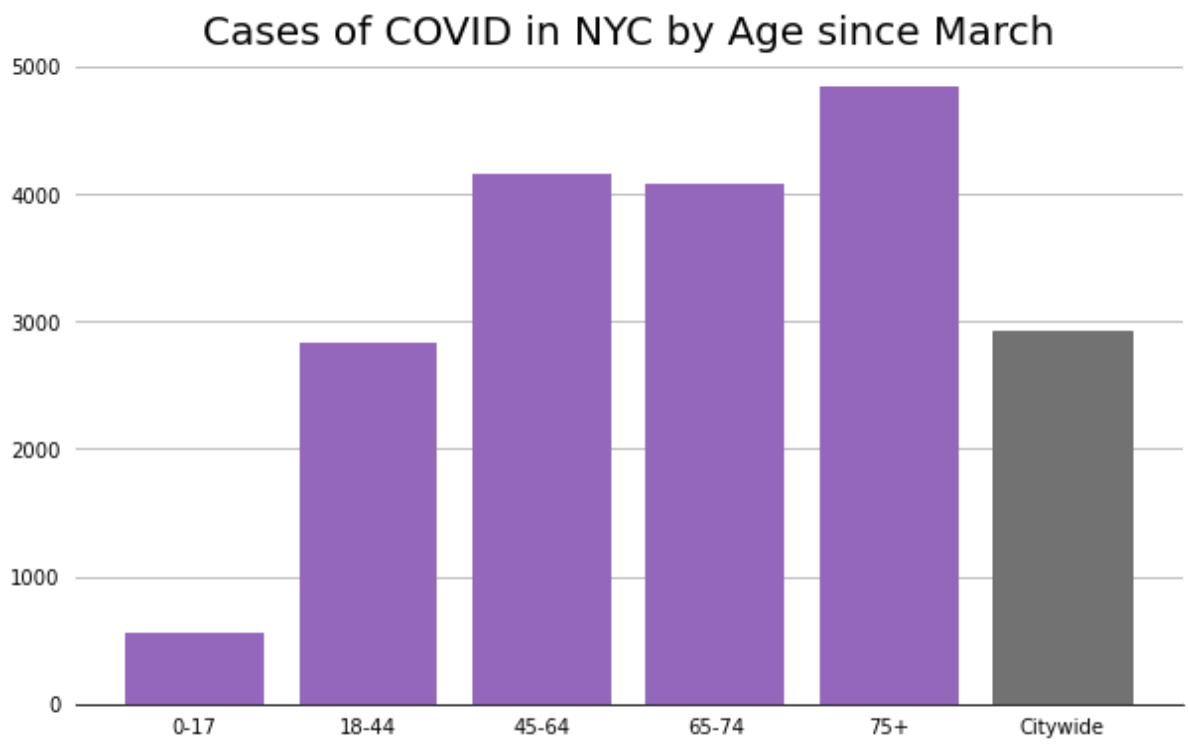
Each function used to create Graph 1 is in the table below and has a brief description of how it was used.

MatplotLib Function:	Description:
ax.bar()	Used to plot the data
bars[].setcolor()	Used to set the color of the bars selected
plt.xticks()	Used to adjust the ticks on the x axis and change size, color, alignment, etc
plt.yticks()	Used to make the y tick labels increase by 400 thousand
plt.suptitle()	Used to create the large title
plt.title()	Used to create the smaller title
ax.spines[].set_visible(False)	Used to remove the top, left, and right spines
ax.tick_params()	Used to remove the ticks but keep the tick lables
plt.grid()	Used to create and format the horizontal lines going across the graph
ax.annotate()	Used to create the text floating in the graph as well as the arrow
fig.set_size_inches()	Used to make the graph larger

Graph 2

For our second graph, we were allowed to replicate any graph of our choosing. We found a graph on nyc.gov which depicted COVID cases in New York City since March, ordered by age group. The code to create this graph is below.

In [7]:
```
# Here is the data for the height of the bars in the graph and the x tick labe
ls
data = {'0-17':551.15, '18-44':2834.21, '45-64':4153.84,
        '65-74':4077.85, '75+':4839.34, 'Citywide':2927.77 }

age_groups = list(data.keys())

covid_cases = list(data.values())

fig, ax = plt.subplots()

# Adding a title and changing font to match the graph
plt.title("Cases of COVID in NYC by Age since March", fontsize = 20)

# Plots the data for the graph
barlist = plt.bar(age_groups, covid_cases, color ='#9467bd', width = .8, zorde
r=2)

# Changing the color of the bars to make it match the graph perfectly, colors
 were found using a color code lookup website
barlist[5].set_color('#727272')

# Makes horizontal lines at each y tick label and sets them to go behind the b
ars
plt.grid(axis = "y", zorder = 0)

# Removes the ticks but keep the tick lables
plt.tick_params(left=False,bottom=False)

# Removes the left, top, and right spines
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)

# Enlarges the graph
fig.set_size_inches(10,6)

plt.show()
```



Summary of Graph 2:

The bar chart gives information about the confirmed number of COVID-19 cases since March 2020 in NYC by different age groups. It is clear from the chart that the age group, 75 years and above, tend to be the most infected with the virus. The age group least likely to contract COVID-19 are of ages 0-17. However, the number of confirmed cases in the 18-44 years is almost 4 times as large as the number of cases in the age group of 0-17. The pattern is soaring for the age groups after 18-44 years. The graph further shows a steady rate of infection for the age groups between 45-64 and 65-74. The last bar in the chart represents the confirmed cases, citywide.

Shortfalls of Graph 2:

1. The two measurement types shown in the graph are not strongly correlated (age groups vs citywide).
2. Quantity value is left out for each bar in the graph.
3. The population size is not mentioned in the graph, it could mislead the audience to wrongly interpret the graph. (the case rate is as per 100,000 people)
4. The unnecessary information like grids in the graph may not be easier for the audience to read (Chartjunk).

Suggested Improvements to Graph 2:

1. Comparing two or more correlated types of measurement, with respect to Covid, would give the users a better understanding of how the impact of the virus has been. (eg. Age groups vs Death rate vs Hospitalization rate)
2. Labelling the test samples (the case rate is as per 100,000 people) would provide a clear picture of the message with the understanding of the image in respect of population.
3. Highlighting critical points of the peak, drop or stagnation in the group would be more helpful for the users to make informed decisions.
4. Removing unnecessary information such as grids and uncorrelated types of measurement (citywide) in the graph would be more appealing for the users to read the trend.

Tabular summary of Matplotlib functions used to create Graph 2:

Each function used to create Graph 2 is in the table below and has a brief description of how it was used.

Matplotlib Function:	Description:
fig.set_size_inches()	Used to make the graph larger
plt.title()	Used to create the title
plt.bar()	Used to plot the data
barlist[].set_color()	Used to set the color of the bars selected
plt.tick_params()	Used to remove the ticks but keeps the tick labels
plt.grid()	Used to create and format the horizontal lines going across the graph
ax.spines[].set_visible(False)	Used to remove the top, left, and right spines

Conclusion:

Overall, our group was successful in recreating the assigned graphs using Matplotlib. Some of the most common adjustments we made to do this was setting titles, removing spines, removing ticks, setting labels, and changing the colors of the bars in our graphs. By completing this assignment, we improved our understanding and capabilities using Matplotlib. We also improved our skills in creating a formal report in Jupyter Notebook that requires both code and multiple types of output to tell the story of the data.

Sources:

1. Graph 1: https://www.bbc.com/news/world-us-canada-44319094 (https://www.bbc.com/news/world-us-canada-44319094)
2. Graph 2: https://www1.nyc.gov/site/doh/covid/covid-19-data.page?fbclid=IwAR2akR2_P5SU2BnEgGJKeNBHjljL0BE3MXGktGYDv3yCh4xIQwz5O9H52XM (https://www1.nyc.gov/site/doh/covid/covid-19-data.page?fbclid=IwAR2akR2_P5SU2BnEgGJKeNBHjljL0BE3MXGktGYDv3yCh4xIQwz5O9H52XM)