

# Computational Thinking

Project Solution

Wei Ching Lim

## Overview

The objective of our project is to find out the performance of private industries in the United States during 2019 and 2020. In order to do so, we have created a GUI application that displays graphs to the end user.

The final output of our project is a GUI window that displays various graphs when selected by the end user. We categorized our graphs into 3 parts:

1. 2019
  - a. Q1 to Q4 of 2019
  - b. Q1 of 2019
  - c. Q2 of 2019
  - d. Q3 of 2019
  - e. Q4 of 2019
2. 2020
  - a. Q1 to Q4 of 2020
  - b. Q1 of 2020
  - c. Q2 of 2020
  - d. Q3 of 2020
  - e. Q4 of 2020
3. Both 2019 and 2020

Data is visualized into bar charts and line graphs for easier visibility on the end user's side so they can quickly analyze the performance of private industries. This makes it convenient for users to obtain information of high or low performing sectors as well as the trend from 2019 to 2020 in the United States.

## Functions

To visualize the data, we defined a total of 25 functions for our project. The *utilities.py* module contains 11 functions while the *main.py* module contains 13 functions. In our main module, *main.py*, there also contains a class called "gui" for all the widgets that is required in our GUI window. The functions implemented in each module are tabulated as a function dictionary. Table 1 is for the *main.py* module and Table 2 is for the *utilities.py* module.

### **main.py**

The majority of functions are created to display images on our GUI window by calling the relevant functions from *utilities.py*

Input: none

Output: GUI window

```

11
12 class gui:
13     def __init__(self, master):
14         frame = Frame(master, width=750, height=500)
15         frame.place(x=30, y=30)
16
17         # ----- Menu Creation -----
18         #Private Industries in 2019
19         self.mbar = Frame(frame, relief = 'raised', width=750, bd = 2)
20         self.mbar.pack(expand = 0, fill = X, side = TOP)
21
22         # ----- Create industry choice menu -----
23
24         self.fgbutton = Menubutton(self.mbar, text = 'Year 2019')
25         self.fgbutton.pack(side = TOP)
26         self.fgmenu = Menu(self.fgbutton, tearoff=0)
27         self.fgbutton['menu'] = self.fgmenu
28
29         # Populate image choice menu
30         self.fgmenu.add('command', label = 'Q1 to Q4 - 2019', command = self.tot_2019)
31         self.fgmenu.add('command', label = 'Quarter 1 2019', command = self.q1_2019)
32         self.fgmenu.add('command', label = 'Quarter 2 2019', command = self.q2_2019)
33         self.fgmenu.add('command', label = 'Quarter 3 2019', command = self.q3_2019)
34         self.fgmenu.add('command', label = 'Quarter 4 2019', command = self.q4_2019)
35
36         # ----- Create industry choice menu -----
37         #Private Industries in 2020
38         self.fgbutton = Menubutton(self.mbar, text = 'Year 2020')
39         self.fgbutton.pack(side = TOP)
40         self.fgmenu = Menu(self.fgbutton, tearoff=0)
41         self.fgbutton['menu'] = self.fgmenu
42
43         # Populate image choice menu
44         self.fgmenu.add('command', label = 'Q1 to Q4 - 2020', command = self.tot_2020)
45         self.fgmenu.add('command', label = 'Quarter 1 2020', command = self.q1_2020)
46         self.fgmenu.add('command', label = 'Quarter 2 2020', command = self.q2_2020)
47         self.fgmenu.add('command', label = 'Quarter 3 2020', command = self.q3_2020)
48         self.fgmenu.add('command', label = 'Quarter 4 2020', command = self.q4_2020)
49

```

Figure 1.1

This function contains all the widgets to create our GUI window. We included the menus, a listbox, textbox and an image label.

```

82
83
84     def clear(self):
85         self.listbox.delete(0, END)
86         self.lblImage.config(image='')
87

```

Figure 1.2

This function is created to clear the image from the GUI window and empty the text in the 'Log' listbox.

```

87
88     def tot_2019(self):
89         self.lblImage.config(image='')
90         u.Industry2019()
91         try:
92             self.image1 = Image.open('ind2019.png')
93             self.obj1 = ImageTk.PhotoImage(self.image1)
94             self.lblImage = tkinter.Label(root, image= self.obj1)
95             self.lblImage.image = self.obj1
96             self.lblImage.place(x=300, y=10)
97             self.listbox.insert(END, 'Private industries percent change contribution to GDP during 2019.')
98         except:
99             self.listbox.insert(END, 'Error loading graph')
100
101     def q1_2019(self):

```

Figure 1.3

This function will display the graph for the performance of each private industry sector from quarter 1 to quarter 4 in 2019. It calls the function “Industry2019” from the *utilities.py* module. There is another function similar to this but it displays the graph for quarter 1 to quarter 4 in 2020 and will call the function “Industry2020” instead. The output of this function is a line graph.

```

100
101     def q1_2019(self):
102         self.lblImage.config(image='')
103         u.Ind2019_Q1()
104         try:
105             self.image1 = Image.open('ind2019_Q1.png')
106             self.obj1 = ImageTk.PhotoImage(self.image1)
107             self.lblImage = tkinter.Label(root, image= self.obj1)
108             self.lblImage.image = self.obj1
109             self.lblImage.place(x=300, y=10)
110             self.listbox.insert(END, 'Private industies performance during Quarter 1 of 2019.')
111         except:
112             self.listbox.insert(END, 'Error loading graph')
113

```

Figure 1.4

This function will display the graph for the performance of each private industry sector for quarter 1 in 2019. It calls the function “Ind2019” from the *utilities.py* module. In total, we have 8 functions similar to this for a total of 8 quarters from 2019 to 2020. Each function will call the relevant functions in the *utilities.py*. Furthermore, docstrings are indeed to each function defined in the *utilities.py* module to inform users which functions it relates to in the *main.py* module. The output of this function will be a bar chart.

```

210
211     self.listbox.insert(END, 'Error loading graph')
212
213     def both_years(self):
214         self.lblImage.config(image='')
215         u.Industry()
216         try:
217             self.image1 = Image.open('totalIndustry.png')
218             self.obj1 = ImageTk.PhotoImage(self.image1)
219             self.lblImage = tkinter.Label(root, image= self.obj1)
220             self.lblImage.image = self.obj1
221             self.lblImage.place(x=300, y=10)
222             self.listbox.insert(END, 'Private industries percent change contribution to GDP from 2019 to 2020')
223         except:
224             self.listbox.insert(END, 'Error loading graph')
225

```

Figure 1.5

This function will display the graph for the performance of the private industry as a whole. It references the function “Industry” from the *utilities.py* module. The output of this function is a bar chart.

*Table 1 Function dictionary for main.py*

<b>Functions</b>	<b>Description</b>	<b>Output/ Results</b>
def __init__(self, master):	Contains all the widgets. A menu, listbox, textbox and image label is created	Creates GUI window
def clear(self):	To remove image and items in “Log” listbox	Clears image and “Log” listbox
def tot_2019(self):	Displays graph for each private industry sectors percent change to GDP during Q1, Q2, Q3 and Q4 in 2019 as a line graph	Line graph
def q1_2019(self):	Displays a bar chart for the the percent change to GDP of each private industry sector during Q1 in 2019	Bar chart
def q2_2019(self):	Displays a bar chart for the the percent change to GDP of each private industry sector during Q2 in 2019	Bar chart
def q3_2019(self):	Displays a bar chart for the the percent change to GDP of each private industry sector during Q3 in 2019	Bar chart
def q4_2019(self):	Displays a bar chart for the the percent change to GDP of each private industry sector during Q4 in 2019	Bar chart
def tot_2020(self):	Displays graph for each private industry sectors percent change to GDP during Q1, Q2, Q3 and Q4 in 2020 as a line graph	Line graph
def q1_2020(self):	Displays a bar chart for the the percent change to GDP of each private industry sector during Q1 in 2020	Bar chart
def q2_2020(self):	Displays a bar chart for the the percent change to GDP of each private industry sector during Q2 in 2020	Bar chart
def q3_2020(self):	Displays a bar chart for the the percent change to GDP of each private industry sector during Q3 in 2020	Bar chart
def q4_2020(self):	Displays a bar chart for the the percent change to GDP of each private industry sector during Q4 in 2020	Bar chart

def both_years(self):	Displays a bar chart for the the percent change to GDP of the entire private industry (summing every sectors) of Q1 to Q4 from 2019 to 2020	Bar chart
-----------------------	---	-----------

### utilities.py

There are three main types of functions defined to plot the graphs. Each represents the degree of specificity of the graphs:

1. Plots graphs that displays the performance of private industry sectors as a whole according to each quarter (quarter 1 to quarter 4) from 2019 to 2020
2. Plots graphs that displays the performance of each private industry individually from quarter 1 to quarter 4 for year 2019 and 2020 respectively
  - a. Q1 to Q4 in 2019
  - b. Q1 to Q4 in 2020
3. Plots graphs for each private industry sector individually for a specific quarter in a specific year.
  - a. Q1 of 2019 / Q2 of 2019 / Q3 of 2019 / Q4 of 2019
  - b. Q1 of 2020 / Q2 of 2020 / Q3 of 2020 / Q4 of 2020

Input: gdpdata.csv file

Output: When main.py is executed, these functions will plot graphs and save them as a .png file.

```

12 import pandas as pd
13 import matplotlib.pyplot as plt
14
15 def Industry():
16     """
17     This function sums up all the different categories for each quarter of years 2019 to 2020.
18     The results obtained is the total percent change contributed to GDP by all the private industries.
19     A bar chart will be saved as png file.
20     Used in main.py at def both_years(self):
21     """
22     dict = {'Private Industries': str, '2019_Q1': float, '2019_Q2': float, '2019_Q3': float, '2019_Q4': float}
23     df = pd.read_csv('gdpdata.csv', index_col=0, dtype=dict)
24     d = df.dtypes
25     df.astype(d)
26     df.loc['Total'] = df.sum()
27     df2 = df.iloc[-1]
28     df2 = df2.to_dict()
29
30     quarter = list(df2.keys())
31     percent = list(df2.values())
32
33     fig = plt.figure(figsize = (8, 5))
34     plt.bar(quarter, percent, color = 'royalblue')
35     plt.grid(fig, axis='y')
36     plt.xlabel("Quarter")
37     plt.ylabel("Percent Change to GDP")
38     plt.title("Private Industries Percent Change to GDP in Year 2019-2020")
39     plt.savefig('totalIndustry.png', dpi = 100, bbox_inches='tight')
40

```

Figure 2.1

This function will plot the values from a data frame into a graph. The input is data read from *gdpdata.csv* file. The output is saved as an image in a *.png* file.

This function plots a bar chart for the performance of the private industry as a whole for each quarter from 2019 to 2020. An image titled 'totalIndustry.png' will be created and saved.

```

41 def Industry2019():
42     """
43     This function will plot the percent change contributed to GDP for all categories in the private industry.
44     It will plot for all four quarters in 2019.
45     A line chart will be saved as png file.
46     Used in main.py at def tot_2019(self):
47     """
48     dict = {'Private Industries': str, '2019_Q1': float, '2019_Q2': float, '2019_Q3': float, '2019_Q4': float }
49     df3 = pd.read_csv('gdpdata.csv', index_col=0, dtype=dict)
50     df3.reset_index(level=0, inplace=True)
51     d = df3.dtypes
52     df3.astype(d)
53     df2019 = df3.iloc[:, 0:5]
54     df4 = pd.DataFrame(df2019, columns = ['Private Industries', '2019_Q1', '2019_Q2', '2019_Q3', '2019_Q4'])
55     x = list(df4["Private Industries"])
56     yQ1 = list(df4["2019_Q1"])
57     yQ2 = list(df4["2019_Q2"])
58     yQ3 = list(df4["2019_Q3"])
59     yQ4 = list(df4["2019_Q4"])
60
61     fig = plt.figure(figsize = (10,3))
62     plt.plot(x, yQ1, color = 'blue', label="Quarter 1")
63     plt.plot(x, yQ2, color = 'red', label="Quarter 2")
64     plt.plot(x, yQ3, color = 'orange', label="Quarter 3")
65     plt.plot(x, yQ4, color = 'green', label="Quarter 4")
66     plt.legend(loc='upper right')
67     plt.xticks(rotation = 90)
68     plt.grid(fig, 'both')
69     plt.xlabel("Private Industries")
70     plt.ylabel("Percent Change to GDP")
71     plt.title("Private Industries Percent Change to GDP in Quarter 1 to Quarter 4 – Year 2019")
72     plt.savefig('ind2019.png', dpi = 100, bbox_inches='tight')
73

```

Figure 2.2

This function will plot the values from a data frame into a graph. The input is data read from *gdpdata.csv* file. The output is saved as an image in a *.png* file.

This function plots a line graph for the performance of each individual sector in the private from quarter 1 to quarter 4 during 2019. An image titled 'ind2019.png' will be created and saved.

Another function similar to this is created but only for quarter 1 to quarter 4 during 2020.

```

106
107 def Ind2019_Q1():
108     """
109     This function will plot the percent change contributed to GDP
110     for all categories in the private industry only during Quarter 1 of year 2019.
111     A bar chart will be saved as png file.
112     Used in main.py at def q1_2019(self):
113     """
114     dict = {'Private Industries': str, '2019_Q1': float}
115     df6 = pd.read_csv('gdpdata.csv', index_col=0, dtype=dict)
116     df6.reset_index(level=0, inplace=True)
117     d = df6.dtypes
118     df6.astype(d)
119     df2019 = df6.iloc[:, 0:2]
120     df7 = pd.DataFrame(df2019, columns = ['Private Industries', '2019_Q1'])
121     x = list(df7["Private Industries"])
122     y = list(df7["2019_Q1"])
123
124     fig = plt.figure(figsize = (10,3))
125     plt.bar(x, y, color = 'limegreen')
126     plt.grid(axis='y')
127     plt.xticks(rotation = 90)
128     plt.xlabel("Private Industries")
129     plt.ylabel("Percent Change to GDP")
130     plt.title("Private Industries Percent Change to GDP in Quarter 1 – Year 2019")
131     plt.savefig('ind2019_Q1.png', dpi = 100, bbox_inches='tight')
132

```

Figure 2.3

This function will plot the values from a data frame into a graph. The input is data read from *gdpdata.csv* file. The output is saved as an image in a *.png* file.

This function plots a bar chart for the performance of each individual sector in the private for quarter 1 in 2019. An image titled 'ind2019\_Q1.png' will be created and saved.

There are functions with a structure similar to this but it only obtains the columns for quarter 2 in 2019, quarter 3 in 2019 up until quarter 4 in 2020.

*Table 2 Function dictionary for utilities.py*

Functions	Description	Output/ Results
def Industry():	Plots the entire private industries performance from Q1 to Q4 of years 2019 through 2020 as a bar chart	Image saved as: 'totalIndustry.png'
def Industry2019():	Plots the performance of each industry during Q1, Q2, Q3 and Q4 of 2019 as a line graph	Image saved as: 'ind2019.png'
def Industry2020():	Plots the performance of each industry during Q1, Q2, Q3 and Q4 of 2020 as a line graph	Image saved as: 'ind2020.png'
def Ind2019_Q1():	Plots a bar chart for the the percent change to GDP of each private industry sector during Q1 in 2019	Image saved as: 'ind2019_Q1.png'
def Ind2019_Q2():	Plots a bar chart for the the percent change to GDP of each private industry sector during Q2 in 2019	Image saved as: 'ind2019_Q2.png'
def Ind2019_Q3():	Plots a bar chart for the the percent change to GDP of each private industry sector during Q3 in 2019	Image saved as: 'ind2019_Q3.png'
def Ind2019_Q4():	Plots a bar chart for the the percent change to GDP of each private industry sector during Q4 in 2019	Image saved as: 'ind2019_Q4.png'
def Ind2020_Q1():	Plots a bar chart for the the percent change to GDP of each private industry sector during Q1 in 2020	Image saved as: 'ind2020_Q1.png'
def Ind2020_Q2():	Plots a bar chart for the the percent change to GDP of each private industry sector during Q2 in 2020	Image saved as: 'ind2020_Q2.png'
def Ind2020_Q3():	Plots a bar chart for the the percent change to GDP of each private industry sector during Q3 in 2020	Image saved as: 'ind2020_Q3.png'

def Ind2020_Q4():	Plots a bar chart for the the percent change to GDP of each private industry sector during Q4 in 2020	Image saved as: 'ind2020_Q4.png'
-------------------	---	----------------------------------

Final Results

Figures 3 to 8 below shows what our project looks like when executed as well as some examples of what it should display when prompted by the user.



Figure 3 GUI window

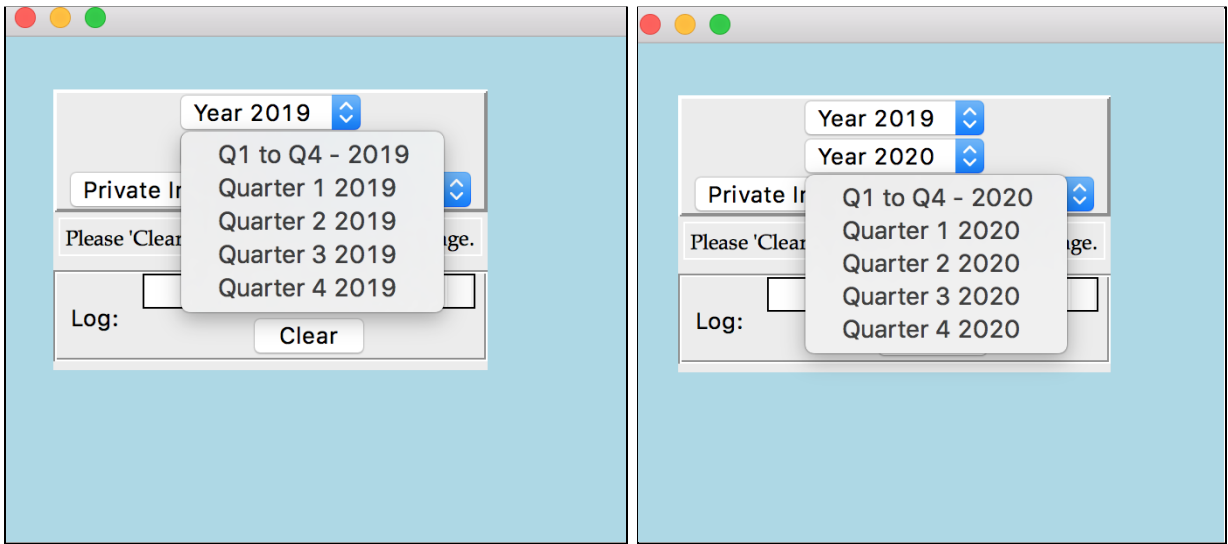


Figure 4 Menu buttons



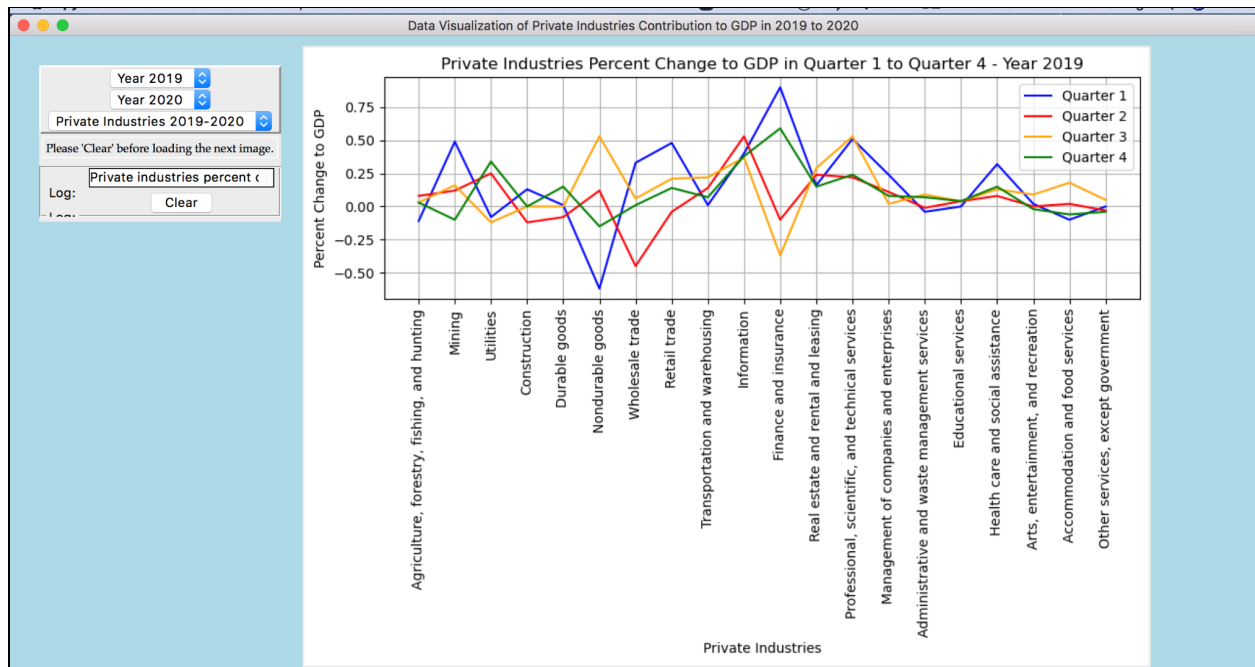


Figure 5 Graph for Q1 to Q4 in 2019

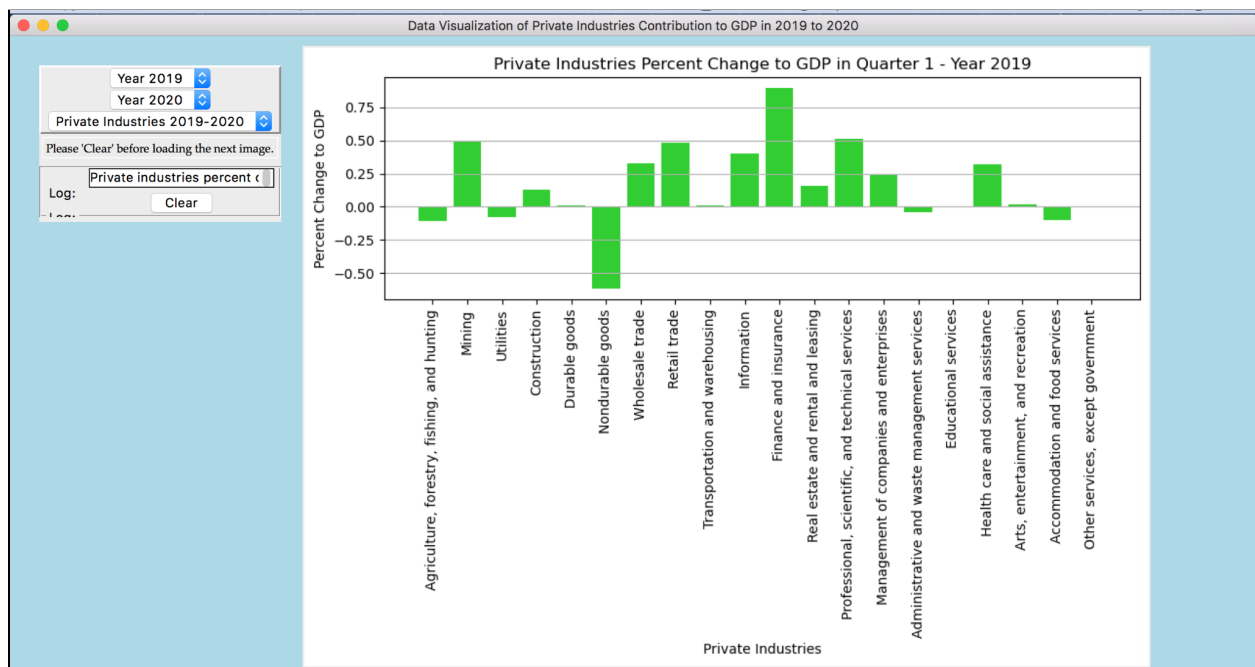


Figure 6 Graph for Q1 in 2019

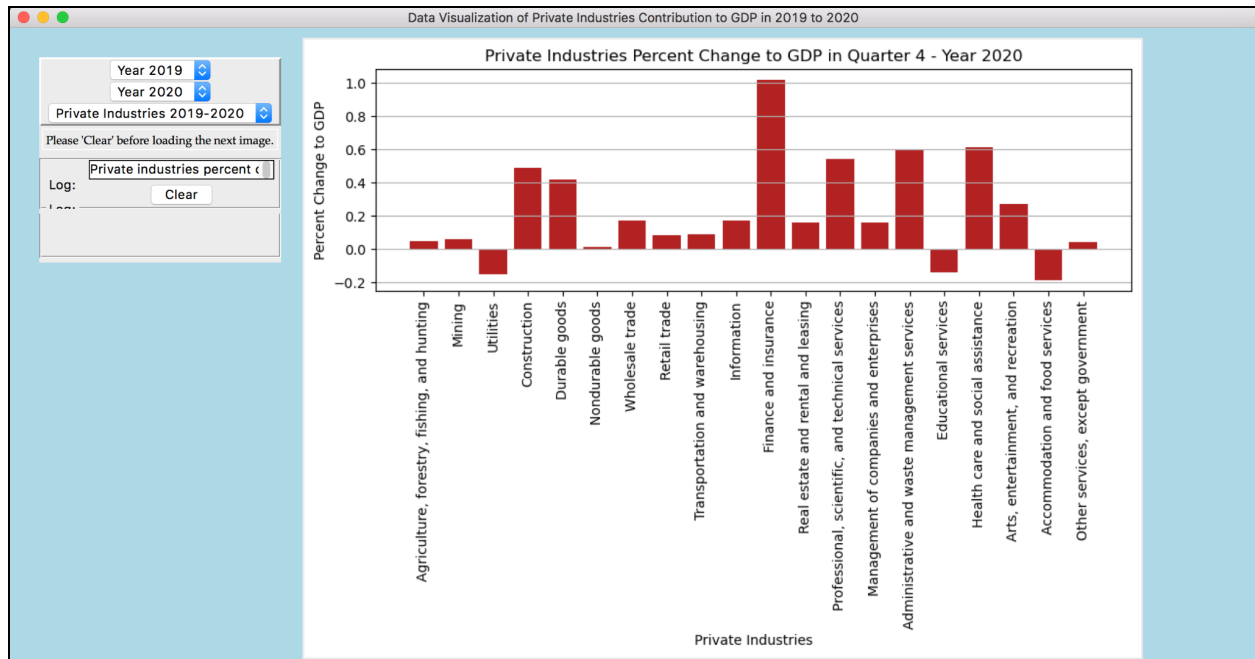


Figure 7 Graph for Q4 in 2020

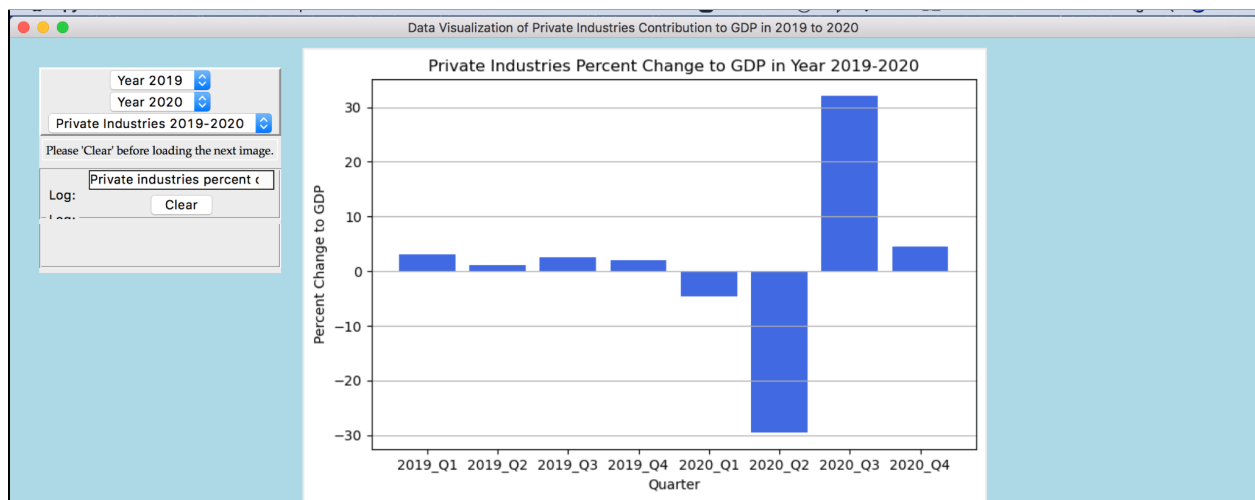


Figure 8 Graph for 2019 - 2020 according to quarters