**HOW COVID-19 HAS IMPACTED THE NUMBER OF NEWBORNS AND ITS CONSEQUENCES**

Logo

Description automatically generated

Submitted to

Dr Bryan Hammer,

Oklahoma State University,

Stillwater, Oklahoma.

October 24, 2021

PROJECT TEAM MEMBERS

Rupom Bhattacherjee - A20221013

Adwoa Boadi-Asamoah - A20198067

Chitra Boorla - A20349295

Srikanth Daruru - A20349204

Kodjo Botchway - A20338464

# **TABLE OF CONTENTS**

[**TABLE OF CONTENTS** 2](#_Toc88827861)

[**Executive Summary** 3](#_Toc88827862)

[**Statement of Scope** 3](#_Toc88827863)

[**Project Goal:** 4](#_Toc88827864)

[**Project Objectives:** 4](#_Toc88827865)

[**Unit of Analysis:** 4](#_Toc88827866)

[**Variables:** 4](#_Toc88827867)

[**Project Schedule** 5](#_Toc88827868)

[**Data Preparation** 6](#_Toc88827869)

[**Data Access** 6](#_Toc88827870)

[**Data Cleaning** 7](#_Toc88827871)

[**Data Transformation** 8](#_Toc88827872)

[**Data Reduction** 8](#_Toc88827873)

[**Data Dictionary** 8](#_Toc88827874)

[**Variable Visualization** 9](#_Toc88827875)

# **Executive Summary**

The changes in the general trends of population growth have been consequential regarding various aspects of life. Be it a surge or a plunge, these changes tend to leave a significant footprint in their wake, which would require targeted mitigations for effective re-stabilization. The Covid-19 pandemic did no less to contribute to this issue. This project seeks to discuss the effect of covid19 on these population changes, specifically, the increase in the number of newborns over a period, which in turn, depending on existing capabilities, will affect the stability of any country as long-term economic growth depends on three factors, i.e., population, participation, and productivity. The potential problem identified is the absence of adequate preparations based on whether there is a baby boom (surge), or baby bust (plunge) in each country. In this project, by identifying the above-related problems, we will try to identify the kind of potential problems given countries will face. This study will benefit the families of newborns and the policymakers to understand the current scenario and prepare for the upcoming challenges.

# **Statement of Scope**

In this project, we aim to determine how covid has impacted population growth in countries worldwide. COVID-19 has had its fair share of impact on various sectors of the world, including but not limited to economic and financial stability, health, and other related development. We are going to understand the population effect on some of these sectors. For example, suppose there is a baby boom in an under-populated country. In that case, it will help in the proper application of underutilized resources and increased economic status since new ventures will be created because of the increase in the availability of labor. On the other hand, in a highly populated country, the baby boom will result in strains in various systems, so they would need to have a proper plan for the effective use of resources and labor.

### **Project Goal:**

● Covid – 19 effects on the number of newborns in different countries

● Effect of increase/decrease in population on different developmental factors

### **Project Objectives:**

● To determine how the total number of covid cases, total deaths, number of patients, availability of hospital beds, and the overall covid19 situation of a country have affected the rate of fertility

● Our sample will be newborn babies and how the surge or plunge in numbers might affect the socio-economic growth factors

### **Unit of Analysis:**

The unit of analysis for our project would be the fertility rate. The fertility rate will help us understand how covid has contributed to the varying rate of pregnancies and its consequences.

### **Variables:**

The following are the variables we are taking into consideration for our project:

1. Fertility Rate (Live Births per Woman)

2. Gross Domestic Product

3. Reproductive choices for women

5. Cost of living

6. Quality of life

7. Covid19 data (e.g., total cases, deaths, etc.)

8. Tweets on public reaction on the effect of covid 19 in birthrate

# **Project Schedule**

The project was set to be worked on and completed over 16 weeks. The overview of the project schedule is provided in the GANTT chart shown below, with all the dates and timelines concerned with the completion of the project along with roles, assignments, and duration. For a better view, please refer to the [online](https://ostatemailokstate-my.sharepoint.com/:x:/g/personal/rupom_bhattacherjee_okstate_edu/Ed5wsC9xFa9BhZvxaPZTPTkBaSmVqxosQ6a7oa-5LNGgww?e=nb1dsq) version of the source file.

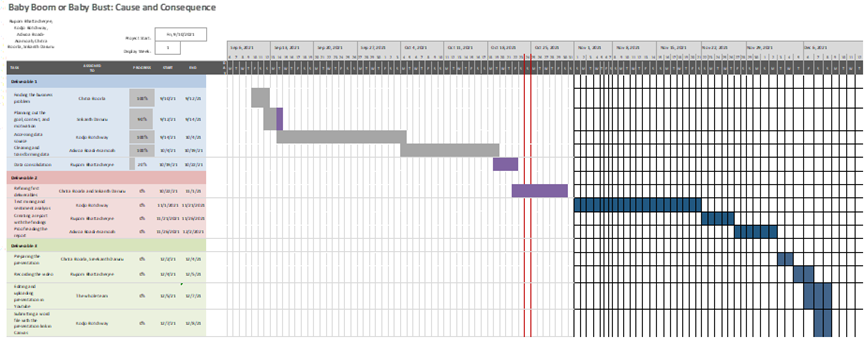


Figure 1: GANTT Chart

# **Data Preparation**

### **Data Access**

The first step in the data preparation process was to identify major sources where the data could be correlated, and some relations could be effectively drawn as such. Most of the data we collected was from population census data mainly wrought from world data banks. Other data was collected from several websites; however, the primary data source was ourworldindata.org, shown in **Figure 1**. The countries that were considered mainly in our analysis included but were not limited to India, the USA, China, France, and Ghana, considering a wide range of nations from different backgrounds. These selections were to grasp different sections of the world in a broader perspective considering the variables and to account for the fact that not all the information was present or available for all the countries.

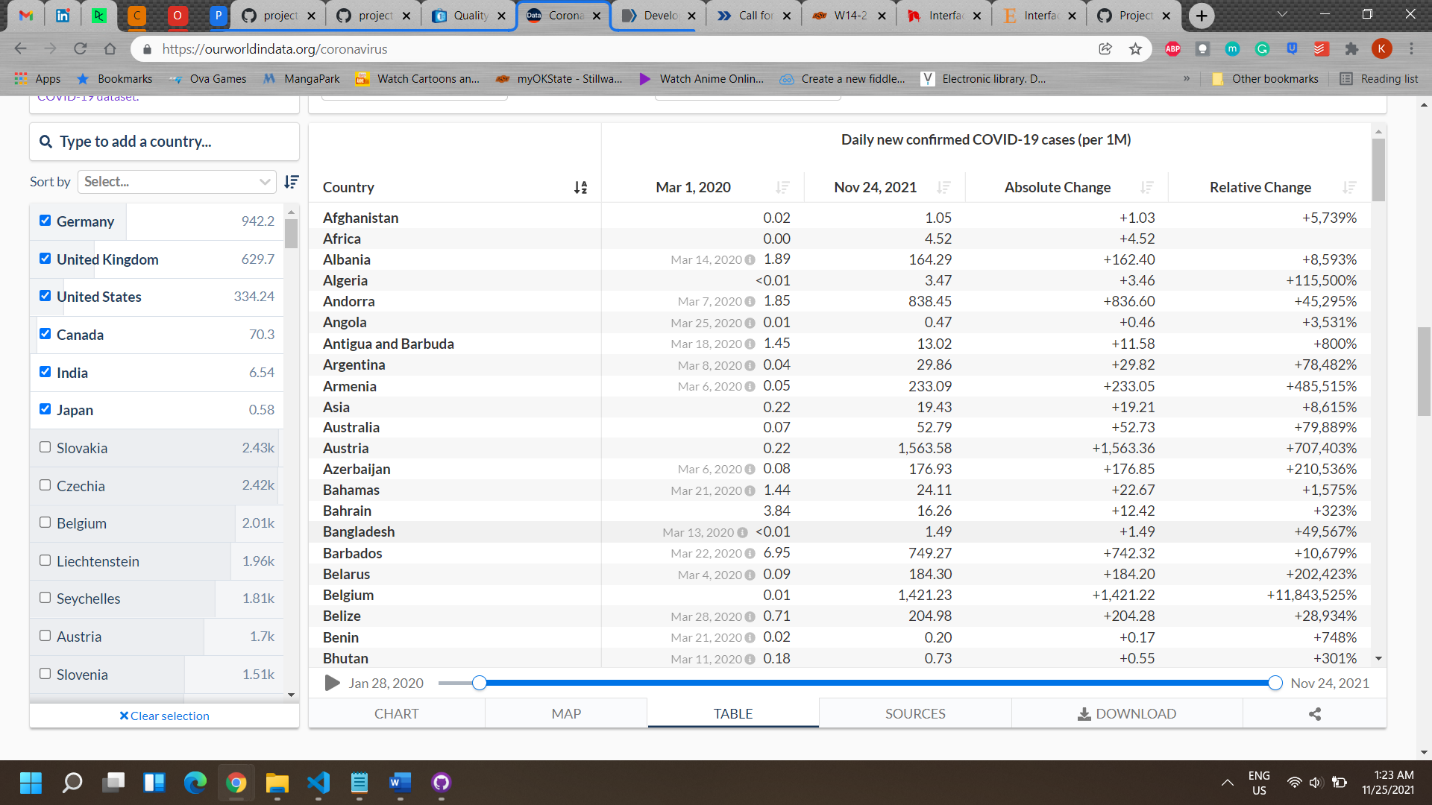


Figure 2: Web Source

The next step involved was to obtain the data. A major volume of the data that we used we primarily did not have to extract from the stated data source through web-scraping. The supporting websites, on the other hand, did some scrapping with the ‘rvest’ function in R Programming to extract the tables and convert them to useable data frames. Consequently, the data still had to be worked on since there were missing and irrelevant data that needed to be cleaned out, which would be further explained in the report. The CSV files for different countries under various variables were downloaded and loaded into a data frame for further analysis. The scraped data was also merged with existing data by the country and then the year to retain the consistency among the different CSV files. The data consisted of the factors contributing to the increase or the decrease in the number of babies born within a particular period. The consequences were described by another data set that investigated the general ramifications for the selected period. The full codes for the scraping and cleaning of the data can be seen in the **Appendix**.

### **Data Cleaning**

The number of countries present in the dataset that we obtained encompassed all the available countries around the world for which the information from the websites could account for. This included aggregated data for regions and sub-regions classified on a continental basis. We had to remove this data and filter out other non-country information. For example, in our *cleaned family planning data for married women.csv* dataset, we had information on the world, region, and subregion aggregations. We had to clean that dataset by removing all the observations for the index subregion. We also had to remove some of the columns we are not using in our work, such as *country type*, *timeframe,* and *FIPS*. Also, in merging some of our data sources, we realized that some of the tabular information contained countries that did not have data in respect to other columns. To clarify, we had almost every detail for the countries, and their Covid-19 breakdowns but not all these countries had data for the reproductive choices for women. The plan was to represent these values by NA values, but on further discussion, we resulted in removing these countries from the data frames. Since the analysis will be conducted as a guideline to the effects of the population changes, the conclusions would still be valid and applicable to every country depending on the variables taken into consideration. We also had the Covid-19 data provided by each day, so we created a column for the cumulative numbers recorded by the month for each country. We wrote a regular expression for obtaining the final cumulative count for the entire year.

### **Data Reduction**

Primarily, for data reduction, the plan is to consider some selected countries from different sections of the world and obtain some correlations on the data we have obtained and use those correlations for analysis with all the other data. We have smaller data sets provided from countries where a population difference would be glaring, including the USA, China, India, and Nigeria.

**Data Consolidation**

The primary datasets we obtained were finally put into one combined file named baby.csv. The other databases used were given similar names describing the information they contained, Baby\_\*\*\*.csv in that light. The codes were written in Python named Baby\_Script.py and R as well and named Data-Combined.R which comprised the file handling, the cleaning, extraction, and reduction write-ups. The section of code used for all of this would be entered in the appendix after significant further work, minimal edits and parsing has been done.

### **Data Dictionary**

We have two major datasets that we used for the report and here are the data dictionaries of the datasets.

*Baby\_Covid.csv*

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type** | **Source** |
| Location | Name of the country | char(30) | https://ourworldindata.org/coronavirus |
| Year | Year for the following data | integer | https://ourworldindata.org/coronavirus |
| Iso\_Code | Country short code | char(30) | https://ourworldindata.org/coronavirus |
| Population | Population of the country | integer | https://ourworldindata.org/coronavirus |
| GDP | GDP of the country | float | https://ourworldindata.org/coronavirus |
| Cases | The number of Covid cases recorded | integer | https://ourworldindata.org/coronavirus |
| Deaths | The number of Covid deaths recorded | integer | https://ourworldindata.org/coronavirus |
| Cases\_  per\_Million | The number of Covid cases recorded per 1 million population | float | https://ourworldindata.org/coronavirus |
| Deaths\_  per\_Million | The number of Covid deaths recorded per 1 million population | float | https://ourworldindata.org/coronavirus |
| Hosp\_Patients\_  per\_Million | Hospital patients recorded per million population | float | https://ourworldindata.org/coronavirus |
| Hospital\_beds\_  per\_thousand | Hospital beds per million population | float | https://ourworldindata.org/coronavirus |
| Icu\_Patients\_  per\_Million | ICU Patients per million population | float | https://ourworldindata.org/coronavirus |

Baby.csv

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type** | **Source** |
| Country | Name of the country | char(30) | https://ourworldindata.org/coronavirus |
| Year | Year for the following data | integer | https://ourworldindata.org/coronavirus |
| Code | Country short code | char(30) | https://www.numbeo.com/quality-of-life/rankings\_by\_country.jsp?title=2020 |
| Cost.of.Living.Index | Indexes for the cost of living for the countries | float | https://www.numbeo.com/cost-of-living/rankings\_by\_country.jsp?title=2020&displayColumn=-1 |
| Rent.Index | Indexes for the rent for the countries | float | https://www.numbeo.com/cost-of-living/rankings\_by\_country.jsp?title=2020&displayColumn=-1 |
| Health.Care.Index.x | Indexes for Health Care for the countries | float | https://www.numbeo.com/health-care/rankings\_by\_country.jsp?title=2020 |
| Quality.of.Life.Index | Indexes for Quality of Life for the countries | float | https://www.numbeo.com/quality-of-life/rankings\_by\_country.jsp?title=2020 |
| Annual Births per Country | The annual number of births by the country for that given year | integer | https://ourworldindata.org |
| Family Planning | Family Planning indexes for countries | float | https://ourworldindata.org |
| Live Births per Woman per Country | Live births per woman per country | float | https://ourworldindata.org |

### **Variable Visualization**

Chart

Description automatically generated

The next step we took was to explore the continuous variables of the datasets. In order to effectively utilize the variables, we need to understand the relationships and the interdependencies on each of them with the other values including the target variable. We identified amongst all the given variables, the distribution of the values, obtained the mean and median values to generate comparable statistics across the total of all the data we were collected for the respective countries. We completed this procedure using JMP software. The breakdown of the data was given as:

Cost.of.Living.Index.x

Graphical user interface

Description automatically generated with low confidence

Figure 3: Distribution of the variable Cost of Living Index

The figure above shows the frequency, skewness, and the summary statistics of the variable. The distribution of the Rent.Index variable is right skewed with a mean of 49.215, ignoring the impact of the missing values on the summary.

Chart, line chart, scatter chart

Description automatically generated

Figure 4: Normality plot for the Cost-of-Living Index

This figure above illustrates the normality distribution. This reveals the non-normality of the variable.

Rent.Index

Graphical user interface, application

Description automatically generated

Figure 5: Distribution of the variable Rent Index

The figure above shows the frequency, skewness, and the summary statistics of the variable. The distribution of the Rent.Index variable is right skewed with a mean of 18.089, ignoring the impact of the missing values on the summary.

Chart, scatter chart

Description automatically generated

Figure 6: Normality plot for Rent Index

This figure above illustrates the normality distribution. This reveals the non-normality of the variable.

Health.Care.Index.x

Chart, histogram

Description automatically generated

Figure 7: Distribution of the variable Health Care Index

The figure above shows the frequency, skewness, and the summary statistics of the variable. The distribution of the Health.Care.Index.x variable is left skewed with a mean of 63.414, ignoring the impact of the missing values on the summary.

Chart, line chart

Description automatically generated

Figure 8: Normality plot for Health Care Index

This figure above illustrates the normality distribution. This reveals that the variable is normal.

Health.CareExp..Index

Chart, histogram

Description automatically generated

Figure 9: Distribution of the variable Health Care Expertise

The figure above shows the frequency, skewness, and the summary statistics of the variable. The distribution of the Health.CareExp..Index variable is left skewed with a mean of 114.197, ignoring the impact of the missing values on the summary.

Chart, line chart, scatter chart

Description automatically generated

Figure 10: Normality plot for Health Care Expertise

This figure above illustrates the normality distribution. This reveals that the variable is normal.

Quality.of.Life.Index

Chart, histogram

Description automatically generated

Figure 11: Distribution of the variable Quality of Life Index

The figure above shows the frequency, skewness, and the summary statistics of the variable. The distribution of the Quality.of.Life.Index variable is left skewed with a mean of 134.100, ignoring the impact of the missing values on the summary.

Chart

Description automatically generated

Figure 12: Normality plot for Quality-of-Life Index

This figure above illustrates the normality distribution. This reveals that the variable is normal.

Family Planning

Chart, histogram

Description automatically generated

Figure 13: Distribution of the variable Family Planning

The figure above shows the frequency, skewness, and the summary statistics of the variable. The distribution of the Family Planning variable is left skewed with a mean of 0.505, ignoring the impact of the missing values on the summary.

Chart, scatter chart

Description automatically generated

Figure 14: Normality plot for Family Planning

This figure above illustrates the normality distribution. This reveals that the variable is normal.

Live Births per Woman per Country

Chart

Description automatically generated with medium confidence

Figure 15: Distribution of the variable Live Births per Woman per Country

The figure above shows the frequency, skewness, and the summary statistics of the variable. The distribution of the Live Births per Woman per Country variable is right skewed with a mean of 2.165, ignoring the impact of the missing values on the summary.

Chart, scatter chart

Description automatically generated

Figure 16: Normality plot for Live Births per Woman per Country

This figure above illustrates the normality distribution. This reveals the non-normality of the variable.

Annual Births per Country (Target Variable)

Graphical user interface, application

Description automatically generated

Figure 17: Distribution of the variable Annual Births per Country

The figure above shows the frequency, skewness, and the summary statistics of the variable. The distribution of the Annual Births per Country variable is right skewed with a mean of 928960.9, ignoring the impact of the missing values on the summary.

Chart

Description automatically generated

Figure 18: Normality plot for Annual Births per Country

This figure above illustrates the normality distribution. This reveals the non-normality of the variable.

### **Data Transformation**

We plan to transform the variables that strongly deviated from the normality assumption. The backing behind the need to transform and make sure the variables conform to these assumptions to perform a regression analysis to determine the effects on the variables on the annual number of births in that given year. The assumptions of linear regression in this study will be based on the normality of error, constancy of variance, linearity between predictors and variable. Since 2020 was massively impacted by Covid as a known phenomenon, it was not exactly intuitive to compare the year 2020 to the other year in terms of the selected variables in that time and hence the removal of the year variable in the analysis in this section. These were identified as the Cost of Living, Rent Index, Live Births per Woman per Country and Annual Births per Country. The tendency of the continuous variables to be linearly related to the target variable. A log transformation would be applied to make these confirm to the assumptions we are going with. We then run a correlation matrix to identify which of the variables has strong correlations with each other.

Text

Description automatically generated with low confidence

Figure 19: Correlation Matrix of Variables

From **Figure 17**, we identified the strong correlations between each and then went further to use VIF values to take out the variables that have high possible affiliations that will distort the predictions of the model. Rent Index and Cost of Living showed a very high correlation factor and between the Health Care Exp Index and the Health Care Index and with that between these pairings we were only going to select just one of them.

### **MULTIPLE REGRESSION**

The building of a multiple regression prediction equation was to help identify the included variables and their order of importance in the said equation. To further elaborate on the ineffectiveness of incorporating both variables with high correlation, the first model was created with all of them present.

Graphical user interface, table

Description automatically generated

Figure 20: Multiple Regression with all factors

Graphical user interface, text, application

Description automatically generated

Figure 21: Summary of fit for Model 1

Graphical user interface, text, application

Description automatically generated

Figure 22: Parameter Estimates of Model 1

The figures above (**Figure 18,19 and 20**) illustrate the significance and order of importance of the respective variables used in the equation. From **Figure 19**, the difference between the RSquare and adjusted RSquare value indicates the presence of variables in the model that are not necessary to the prediction for the effect on the annual births. The significance values (Prob>|t|) show the relevance of the variables to the main target variable. In this model, the only significant variables would be the Family Planning, the Live Births per Woman, and the Quality of Life in this case. However, from the previous correlation analysis we identified some of the parameters that had some affiliation with other parameters. Taking, those out and running another regression model, we came out with Figures **21**, **22** and **23**.

Graphical user interface, application, Word

Description automatically generated

Figure 23: Regression with the selected factors

Graphical user interface, text, application

Description automatically generated

Figure 24: Summary of fit for Model 2

Graphical user interface, text, application

Description automatically generated

Figure 25: Parameter estimates for Model 2

The analytical breakdown of the model with the new parameters states that the model important variable that contributed to the response for the annual births per country in the year 2020 amidst the Covid pandemic was the availability of Family Planning measures to the women and the couples or people on the relationship in general. This was followed by the fertility rate per woman in that country represented by the Live Births per Woman. The presence of the Health Care Index and the Cost-of-Living index, even though, were not proven to be significant, were still included in the model because these were deemed to be important and could not possibly be overlooked in performing the analysis.

Generally, the build of a regression model is very sensitive to the presence of outliers and hence we considered the outliers present in all the predictor variables. These outliers could not be removed as well since India was the predominant outlier in all variables.

### **TEXT MINING AND SENTIMENTAL ANALYSIS**

We also decided to go ahead and perform some sentimental analysis on Tweets regarding the pandemic situation and its effect on birthrate. We scraped all the tweets having pandemic and birthrate as hashtags and then used the regular well-known text mining process generate some information on what people think about the impact of this pandemic situation on birthrate. We haven’t specifically used baby boom or baby bust as hashtags because we wanted to get that information by analyzing the public opinion on birthrate and pandemic. The text data we were going to be used for the analysis were first of all cleaned. The order of the cleaning of the files included:

1. Removing the stop words in the text.
2. Removing the standalone numerical values since some of the numerical values may be affiliated with some text. E.g., Covid-**19**.
3. Removing of all punctuation and extra spaces and formatting options.
4. Changing the words to lower cases so that two words with different capitalizations are not identified as separate words.
5. Removal of the words in the breakdown list that are not necessary to the analysis of the tweets.
6. Stemming/ Lemmatizing the words so their root forms are the ones that are kept.
7. Post-stemming visual analysis to see if any more words need to be removed.
8. Tokenizing the words and converting texts into a document term matrix

After performing the steps, we did the topic analysis, generated a word cloud, assessed different emotions/feelings, generated a classification model, and performed a named entity recognition analysis. The codes are available in script: sentiment\_analysis.ipynb and sentiment.R.

**Topic Analysis:**

Top 10 words for topic #0:

['immigr', 'year', 'coronaviru', 'push', 'popul', 'rise', 'say', 'think', 'econom', 'declin']

Top 10 words for topic #1:

['year', 'ha', 'peopl', '2020', 'japan', 'acceler', 'covid19', 'birth', 'drop', 'declin']

Top 10 words for topic #2:

['becaus', 'boom', 'guarante', 'half', 'born', 'bounc', 'million', 'fewer', '2021', 'babi']

Top 10 words for topic #3:

['educ', 'spiral', 'climat', 'includ', 'global', 'cost', 'health', 'child', 'chang', 'care']

As it can be seen words like coronavirus, population, decline, drop, birth as the top words in the topics indicate that people are talking about the effect of covid 19 as a pandemic on the drop of birthrate. The words like economy, education, cost, health, care indicates the anticipated reasons behind the drop. Figure 26 shows the list of top 15 words from the tweets with their counts. Figure 27 shows the word cloud of the tweets. Similar to the findings from topic analysis, the word cloud does also show that the decline (or drop) in the birthrate is the one people are most talking about.

Table

Description automatically generated

Figure 26: Overall top 15 stemmed terms from the tweets with their frequency

Text

Description automatically generated

Figure 27: Word cloud of most frequent word used in the tweets regarding pandemic and birthrate

Emotions and Feelings:

Surprise and Anticipation

Table

Description automatically generated

Figure 28: Surprise vs Anticipation on the tweets

Chart, bar chart

Description automatically generated

Figure 29: Degree of surpriseness (anticipation-surprise) associated with words from the tweets

Joy and Sadness

Table

Description automatically generated

Figure 30: Joy and sadness associated with the tweets

A picture containing graphical user interface

Description automatically generated

Figure 31: Degree of contentment (joy-sadness) associated with words from the tweets

### **CONCLUSION**

### Appendix

Code for Text Mining and Sentiment Analysis

Setiment\_analysis.ipynb

import selenium

import pandas as pd

from selenium import webdriver

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.common.by import By

from selenium.common.exceptions import TimeoutException

from selenium.webdriver.common.keys import Keys

import time

import nltk

from nltk.stem import PorterStemmer

import matplotlib.pyplot as plt

from nltk import word\_tokenize, sent\_tokenize

from nltk.corpus import stopwords

from nltk.stem import LancasterStemmer, WordNetLemmatizer, PorterStemmer

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.decomposition import LatentDirichletAllocation

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score, plot\_confusion\_matrix

from nltk import word\_tokenize, pos\_tag, ne\_chunk

from nltk.chunk import conlltags2tree, tree2conlltags

url="https://twitter.com/login"

#driver = webdriver.Chrome(executable\_path=r"D:\chromedriver\_win32\\chromedriver.exe")

driver = webdriver.Firefox(executable\_path=r"D:\geckodriver.exe")

driver.get(url)

username=driver.find\_element\_by\_xpath('//\*[@id="layers"]/div[2]/div/div/div/div/div/div[2]/div[2]/div/div/div[2]/div[2]/div[1]/div/div[5]/label/div/div[2]/div/input')

username.send\_keys('rupam\_27'+Keys.ENTER)

userpass=driver.find\_element\_by\_name("password")

userpass.send\_keys('\*\*\*\*\*\*\*'+Keys.ENTER)

search=driver.find\_element\_by\_xpath('//\*[@id="react-root"]/div/div/div[2]/main/div/div/div/div[2]/div/div[2]/div/div/div/div[1]/div/div/div/form/div[1]/div/div/label/div[2]/div/input')

search.send\_keys("pandemic birthrate"+Keys.ENTER)

time.sleep(2)  # Allow 2 seconds for the web page to open

scroll\_pause\_time = 1  # You can set your own pause time. My laptop is a bit slow so I use 1 sec

screen\_height = driver.execute\_script("return window.screen.height;")   # get the screen height of the web

i = 1

n=1

test\_list=[]

#while True:

while n<=100:

    # scroll one screen height each time

    driver.execute\_script("window.scrollTo(0, {screen\_height}\*{i});".format(screen\_height=screen\_height, i=i))

    i += 1

    n+=1

    time.sleep(scroll\_pause\_time)

    # update scroll height each time after scrolled, as the scroll height can change after we scrolled the page

    scroll\_height = driver.execute\_script("return document.body.scrollHeight;")

    # Break the loop when the height we need to scroll to is larger than the total scroll height

    test=driver.find\_elements\_by\_xpath('//div[@class="css-1dbjc4n"]//div[@class="css-1dbjc4n"]//div[@lang="en"]')

    for j in range(len(test)):

        test\_list.append(test[j].text)

    if (screen\_height) \* i > scroll\_height:

        break

text=pd.DataFrame(test\_list,columns=['tweets'])

text.drop\_duplicates(inplace=True,ignore\_index=True)

### Create a Term-Document Matrix

# Remove stop words

stop = stopwords.words('english')

text = pd.DataFrame(text)

textcol = text['tweets']

textcol = textcol.apply(lambda x: " ".join(x for x in x.split() if x not in stop))

# Remove numerical values

patternnum = '\b[0-9]+\b'

textcol = textcol.str.replace(patternnum,'')

# Remove punctuation

patternpunc = '[^\w\s]'

textcol = textcol.str.replace(patternpunc,'')

# Convert to lowercase

textcol = textcol.apply(lambda x: " ".join(x.lower() for x in x.split()))

# Stem the words

porstem = PorterStemmer()

textcol = textcol.apply(lambda x: " ".join([porstem.stem(word) for word in x.split()]))

#remove https

pat\_http='^http?://'

textcol = textcol.str.replace(pat\_http,'')

# Convert data into a document matrix

vectorizer = CountVectorizer()

tokens = pd.DataFrame(vectorizer.fit\_transform(textcol).toarray(), columns=vectorizer.get\_feature\_names())

tokens.columns

print(tokens.columns.tolist())

# LDA

vectorizer = CountVectorizer(max\_df=0.8, min\_df=4, stop\_words='english')

tweet\_values = textcol.values.astype('U') #convert Panda values to unicode

doc\_term\_matrix = vectorizer.fit\_transform(tweet\_values)

doc\_term\_matrix.shape

LDA = LatentDirichletAllocation(n\_components=4, random\_state=35)

LDA.fit(doc\_term\_matrix)

for i,topic in enumerate(LDA.components\_):

    print(f'Top 10 words for topic #{i}:')

    print([vectorizer.get\_feature\_names()[i] for i in topic.argsort()[-10:]])

    print('\n')

text.to\_csv("tweets on pandemic birth rate.csv")

Sentiment.R

library(tidytext)

library(SnowballC)

library(tidyverse)

library(wordcloud2)

library(RColorBrewer)

library(tm)

tweets\_df = read.csv('tweets on pandemic birth rate.csv', header = TRUE)

summary(tweets\_df)

tweets\_data=select(tweets\_df,tweets)

tidy\_dataset=unnest\_tokens(tweets\_data,word,tweets)

head(tidy\_dataset)

counts = count(tidy\_dataset, word)

result1 = arrange(counts, desc(n))

typeof(result1)

slice(result1,1:15)

#removing stop words

data("stop\_words")

tidy\_dataset2 = anti\_join(tidy\_dataset, stop\_words)

counts2 = count(tidy\_dataset2, word)

result2 = arrange(counts2, desc(n))

typeof(result2)

slice(result2,1:15)

#removing numerical values (and blank spaces)

patterndigits = '\\b[0-9]+\\b'

tidy\_dataset2$word=str\_remove\_all(tidy\_dataset2$word, patterndigits )

head(tidy\_dataset2)

counts3 = count(tidy\_dataset2, word)

result3 = arrange(counts3, desc(n))

slice(result3,1:15)

#removing certain words

list\_remove=c("pandemic","birthrate", "https", "tmobilesprint", "â")

tidy\_dataset3 = filter(tidy\_dataset2, !(word %in% list\_remove))

counts4= count(tidy\_dataset3,word)

result4=arrange(counts4,desc(n))

slice(result4,1:15)

#removing new lines

tidy\_dataset3$word=str\_remove\_all(tidy\_dataset3$word, '\r?\n')

counts5= count(tidy\_dataset3,word)

result5=arrange(counts5,desc(n))

slice(result5,1:15)

#removing spacing and tabs

tidy\_dataset3$word = str\_replace\_all(tidy\_dataset4$word, '[ \t]', '')

tidy\_dataset4=filter(tidy\_dataset3, !(word=='') )

counts6= count(tidy\_dataset4,word)

result6=arrange(counts6,desc(n))

slice(result6,1:15)

tidy\_dataset5 = mutate\_at(tidy\_dataset4, "word", funs(wordStem((.), language="en")))

counts5 = count(tidy\_dataset5, word)

arrange(counts5, desc(n)) %>%

  ungroup %>%

  slice(1:15)

install.packages('textdata')

library(textdata)

get\_sentiments('nrc') %>%

  distinct(sentiment)

#JOY and SADNESS

nrc\_joysad = get\_sentiments('nrc') %>%

  filter(sentiment == 'joy' |

           sentiment == 'sadness')

nrow(nrc\_joysad)

newjoin2 = inner\_join(tidy\_dataset5, nrc\_joysad)

counts8 = count(newjoin2, word, sentiment)

spread2 = spread(counts8, sentiment, n, fill = 0)

content\_data = mutate(spread2, contentment = joy - sadness, linenumber = row\_number())

tweet\_joysad = arrange(content\_data, desc(contentment))

#generating plot of top 20

(tweet\_joysad2 = tweet\_joysad %>%

    slice(1:20,107:127))

ggplot(tweet\_joysad2, aes(x=linenumber, y=contentment, fill=word)) +

  coord\_flip() +

  theme\_light(base\_size = 15) +

  labs(

    x='Index Value',

    y='Contentment'

  ) +

  theme(

    legend.position = 'bottom',

    panel.grid = element\_blank(),

    axis.title = element\_text(size = 10),

    axis.text.x = element\_text(size = 10),

    axis.text.y = element\_text(size = 10)

  ) +

  geom\_col()

#surprise and anticipation

nrc\_surprise\_anticipation = get\_sentiments('nrc') %>%

   filter(sentiment == 'surprise' |

        sentiment == 'anticipation')

nrow(nrc\_surprise\_anticipation)

(tweet\_surprise\_anticipation = tidy\_dataset5 %>%

  inner\_join(nrc\_surprise\_anticipation) %>%

  count(word, sentiment) %>%

  spread(sentiment, n, fill = 0) %>%

  mutate(supriseness = surprise - anticipation, linenumber = row\_number()) %>%

  arrange(desc(supriseness)) %>%

  slice(1:20,318:338))

ggplot(tweet\_surprise\_anticipation, aes(x=linenumber, y=supriseness, fill=word)) +

  coord\_flip() +

  theme\_light(base\_size = 15) +

  labs(

    x='Index Value',

    y='supriseness'

  ) +

  theme(

    legend.position = 'bottom',

    panel.grid = element\_blank(),

    axis.title = element\_text(size = 10),

    axis.text.x = element\_text(size = 10),

    axis.text.y = element\_text(size = 10)

  ) +

  geom\_col()

#Wordcloud

df=data.frame(counts5)

set.seed(1234)

wordcloud(words = df$word, freq = df$n, min.freq = 1,

max.words=200, random.order=FALSE, rot.per=0.35,

colors=brewer.pal(8, "Dark2"),scale=c(8,0.5))

set.seed(1234)

wordcloud2(data=df, size=1, color='random-dark')