The correlation between happiness levels and birthrate in Singapore

Project proposal

1. Aim, Objectives, and Background

1.1 Introduction

One of the pressing social issues today is the declining birth rate in many countries. This trend has significant long-term implications, particularly for economic stability, labour force availability, and societal sustainability. Low birth rate can lead to an ageing population, which places immense pressure on the country's healthcare, pension system, and economic growth. Like many developed countries, Singapore is also facing this low birth rate crisis. Despite government policies such as parental incentives, housing benefits, and childcare subsidies, the birth rate remains below replacement levels. This presents a critical challenge for policymakers and the public alike. Beyond economic and policy-related factors, an individual’s mental wellness and overall happiness, also play a role in birth rate trends. Research in other regions has suggested a potential link between happiness levels and decisions to have children. To my knowledge, no prior work has been done exploring the correlation between happiness levels and birth rates specifically, in Singapore. This report aims to address this gap by investigating whether happiness levels and their underlying factors influence Singapore's birth rate. By understanding this relationship, the findings could provide insights into how mental wellness might intersect with broader demographic challenges.

1.2 Aim and Objective

1.2.1 Aim

This report project aims to investigate the correlation between happiness levels and birth rates in Singapore. The analysis focuses on uncovering whether happiness levels and their underlying factors have a measurable impact on birth rate trends in the country.

1.2.2 Objectives

The purpose of this project is to explore the following:

- Analyse the **birth rate** trends in Singapore
  - Examine the birth rate data from 1967 to 2023, to understand the trends and patterns over time.
- Identify correlations between birth rate and happiness in Singapore

1.3 Data

1.3.1 Data requirements

This report aims to explore the correlation between birth rate and happiness levels in Singapore. To conduct this analysis effectively, data for both properties are required. Upon further examination of the happiness index, it became clear that the index is not solely influenced by individual emotions. Instead, it is shaped by multiple underlying factors that reflect broader societal conditions. Consequently, this analysis will also consider the six key variables that contribute to the happiness index to provide a more comprehensive perspective.

1.3.2 Choice of publications

To ensure the reliability and accuracy of the data, this report primarily relies on the following authoritative sources:

- **Singapore Department of Statistics (DOS)**: As a government agency, the DOS provides official statistics on various aspects of Singapore's development. The data from DOS is widely recognised for its credibility and serves as the primary source for birth rate data in this report.
- **World Happiness Report (WHR)**: This annual publication, produced by the United Nations Sustainable Development Solutions Network, provides a comprehensive analysis of global happiness levels. Its data is sourced from Gallup World Poll surveys, which are conducted annually in over 150 countries.
- **World Bank Group**: As an international financial institution, the World Bank Group provides open-access datasets on global development indicators, including economic and social metrics. Its data on GDP and life expectancy are particularly relevant to this analysis.
- **Gallup World Poll**: Conducted annually in over 150 countries, the Gallup World Poll serves as the primary data source for the WHR. It provides insights into public opinion on a wide range of topics, including happiness, social support, and freedom to make life choices.

1.3.3 Limitations and Constraints of the Data

*Time Periods Covered by the **World Happiness Report***

The World Happiness Report (WHR), launched by the United Nations (UN) officially started in 2012. Reports from 2005 to 2011, shown in the WHR were taken and referenced from The Gallup World Poll. WHR measures happiness based on six underlying factors to explain the international differences in happiness scores. These six underlying factors are:

```
1. Gross Domestic Product(GDP) per person
2. Healthy life expectancy (HLE)
3. Social support
4. Perceived freedom to make life choices
5. Generosity
6. Perception of corruption
```

Although the WHR did not release a formal report in 2014, data for that year were subsequently included in later editions. This analysis references the latest available data from the WHR 2024. However, certain years and variables present limitations:

- **Missing data for 2012**: he WHR 2012 did not provide comprehensive happiness data for Singapore, including the six key variables. Limited data on metrics such as "Average Cantril Ladder," "Average Life Satisfaction," and "Average Happiness" were available, but these do not align with the WHR's standard six-variable framework. Hence, happiness data for 2012 are excluded from this analysis.

- **Data for 2020**: For WHR 2020, data were provided for the six variables used for measuring happiness, although it was not included in the data provided from 2021 onwards. Hence, this report *will be* using the data provided in 2020.
- **Handling missing data**: The WHR 2023 states: "We base our usual happiness rankings on a three-year average of these life evaluations, since the larger sample sise enables more precise estimates." When data for certain years is missing, the WHR uses data from the previous year and the subsequent two years to calculate an average. This methodology is applied in this report where applicable.

For instance:

- The WHR 2024 does not provide 2020 data in its summary. However, the WHR 2020 includes Singapore's data in Table 2.1, ranking countries by their happiness scores. *This report uses the WHR 2020 data for 2020.*

*Time Periods Covered by the **Birth Rate in Singapore***

The data taken for the birth rate in Singapore were from 1967 to 2023. *This report will be using the data from 2006 to 2023 only.*

Compensating for the different time periods of the data covered

To ensure consistency, this analysis focuses on data from 2006 to 2023, the period during which both WHR and birth rate data are available for Singapore. Limiting the scope to these years enhances the accuracy and reliability of the findings.

1.4 Ethical Considerations

1.4.1 Use of data

The data used in this report is sourced from reputable organisations that permit the use of their datasets under specific conditions:

- **Singapore Department of Statistics (DOS)**: The DOS terms and conditions state the use of data is allowed, 'provided that you credit the source of the Contents, and ensure the datasets and data in the Contents are accurately reproduced'.
- **World Happiness Report (WHR)**: The WHR website explicitly states that all data used in the report is available for download and can be used.
- **Gallup World Poll**: According to the Gallup terms of service, "news articles, reports, and graphs may only be downloaded for your personal use".

The data used in this report complies with these conditions.

1.4.2 Purpose of use of library

- ***Pandas*** library will be used for data manipulation, processing, and analysis, for this project. It provides efficient tools for handling and transforming structured data.
- ***NumPy*** is a numerical computing library that extends Python's capabilities for working with arrays. It facilitates operations like searching, filtering, and applying conditions to array elements.

- ***Matplotlib*** is a versatile plotting library used for creating static, animated, and interactive visualisations. In this project, Matplotlib will be used together with NumPy, to generate meaningful graphs and plots.
- ***Seaborn*** is a high-level interface for creating statistical graphics. It builds on Matplotlib's capabilities and provides additional features for drawing visually appealing and informative plots.

```
# pip install pandas

# MAY NOT NEED THIS PART

#Import libraries and modules
import pandas as pd #to read CSV file
import matplotlib.pyplot as plt #to plot the graph
import numpy as np
import seaborn as sns #to plot the graph
```

2. Data Importing

2.1 Defining scraping and Extraction functions

The data from both the WHR and Singapore's birthrate to be analysed for this project are all in CSV file, and they are all being changed/used as text file instead.

Extracting information for Singapore's *birth rate* By Birth Order

The following function extracts Singapore's total live birth rate annually by birth order. The csv file is obtained from Singapore Department of Statistics.

```
# To import the CSV file, and read it as a Text file
df_BR = open('SG_birthrate_1967-2023.csv', 'r')
df_BR

# To read the file
df_BR = df_BR.readlines()

# Getting Birthrate
birthrate = df_BR[11]
# Tokenise the Birthrate
birthrateData = birthrate.split(",")[1:] #splitting by ',' & removing the 1st column
birthrateData[-1] = birthrateData[-1].strip(" \n") #to remove \n

# Getting Year
year = df_BR[10]
# Tokenise the Year
yearData = year.split(",")[1:] #splitting by ',' & removing the 1st column
yearData[-1] = yearData[-1].strip(" \n") #to remove \n
```

```
# Converting to pandas DataFrame
year_df = pd.DataFrame({'Year': yearData})
birthrate_df = pd.DataFrame({'Birthrate': birthrateData})

# Combine the data into a New dataframe
newDataBR = pd.concat([year_df, birthrate_df], axis=1)

# Setting the header
header = ['Year', 'Birthrate']

# Save the data to a New CSV file
newDataBR.to_csv('new_SG_birthrate.csv', header=header, index=False)
# print(newData)

# Reading the new csv data
newDataBR_csv = pd.read_csv('new_SG_birthrate.csv')
newDataBR_csv
```

```
    Year  Birthrate
0   2023      33541
1   2022      35605
2   2021      38672
3   2020      38590
4   2019      39279
5   2018      39039
6   2017      39615
7   2016      41251
8   2015      42185
9   2014      42232
10  2013      39720
11  2012      42663
12  2011      39654
13  2010      37967
14  2009      39570
15  2008      39826
16  2007      39490
17  2006      38317
18  2005      37492
19  2004      37174
20  2003      37485
21  2002      40760
22  2001      41451
23  2000      46997
24  1999      43336
25  1998      43664
26  1997      47333
27  1996      48577
28  1995      48635
29  1994      49554
30  1993      50225
```

```
31  1992      49402
32  1991      49114
33  1990      51142
34  1989      47669
35  1988      52957
36  1987      43616
37  1986      38379
38  1985      42484
39  1984      41556
40  1983      40585
41  1982      42654
42  1981      42250
43  1980      41217
44  1979      40779
45  1978      39441
46  1977      38364
47  1976      42783
48  1975      39948
49  1974      43268
50  1973      48269
51  1972      49678
52  1971      47088
53  1970      45934
54  1969      44562
55  1968      47241
56  1967      50560
```

Extracting information from 2024's World Happiness Report

The following function extracts Singapore's Happiness score, variable name **ladder**, over the years from 2006 to 2024.

```
# To import the Excel file
df = pd.read_excel('happiness_data_2024.xls')
df

# Getting Singapore's Data
rows_hap = df.loc[1836:1851, ['Country name', 'year', 'Life Ladder']]

# Rename the columns
rows_hap = rows_hap.rename(columns={'year': 'Year'})
rows_hap = rows_hap.rename(columns={'Life Ladder': 'HappinessScore'})

# Adding in 2020 data - obtained from WHR 2020
hap2020 = pd.DataFrame([{'Country name': 'Singapore', 'Year': '2020',
'HappinessScore': '6.377091'}]) #creates a new dataframe for 2020's
data
rows_hap = pd.concat([rows_hap, hap2020], ignore_index=True)
rows_hap['Year'] = rows_hap['Year'].astype(int) #changing data type to
```

```
string
rows_hap
```

```
   Country name  Year HappinessScore
0     Singapore  2006       6.462703
1     Singapore  2007       6.833755
2     Singapore  2008       6.641957
3     Singapore  2009       6.144677
4     Singapore  2010       6.531402
5     Singapore  2011       6.561042
6     Singapore  2013       6.533207
7     Singapore  2014       7.062365
8     Singapore  2015       6.619525
9     Singapore  2016       6.033481
10    Singapore  2017       6.378438
11    Singapore  2018       6.374564
12    Singapore  2019        6.37836
13    Singapore  2021       6.586717
14    Singapore  2022       6.333046
15    Singapore  2023       6.653942
16    Singapore  2020       6.377091
```

Extracting information for *Gross Domestic Data (GDP) per capita*

The following function extracts Singapore's GDP per capita annually. The xlsx file is obtained from the International Monetary Fund.

- WHR calculates the GDP per capita of a country in terms of purchasing power parity (PPP), which they sourced from the World Bank's Global Economic Prospects.

```
# To import the CSV file, and read it as a Text file
df_gdp = open('world_bank_gdp_2024.csv', 'r')
df_gdp

# To read the file
df_gdp = df_gdp.readlines()

# Getting Singapore's Data
gdp = df_gdp[213]
# Tokenise the GDP
gdpData = gdp.strip(",\n").split(",")[4:] #splitting by ','
gdpData = [value.strip('"') for value in gdpData] #to remove "
# To remove columns 2 and 4
# gdpData_filtered = [value for i, value in enumerate(gdpData) if i
not in [1, 3]]
# print(gdpData_filtered)

# Getting Year
gdpYear = df_gdp[4]
# Tokenise the Year
```

```
gdpYearData = gdpYear.strip(",\n").split(",")[4:] #splitting by ',' & removing the first-4 column
gdpYearData = [value.strip('"') for value in gdpYearData] #to remove "
# print(gdpYearData)

# Converting to pandas DataFrame
gdpYearData_df = pd.DataFrame({'Year': gdpYearData})
gdpData_df = pd.DataFrame({'GDP per capita': gdpData})

# Combine the data into a New dataframe
newDataGDP = pd.concat([gdpYearData_df, gdpData_df], axis=1)

# Setting the header
header = ['Year','GDP per capita (current US$)']

# Save the data to a New CSV file
newDataGDP.to_csv('new_SG_gdp.csv', header=header, index=False)

# # Reading the new csv data
newDataGDP_csv = pd.read_csv('new_SG_gdp.csv')
newDataGDP_csv
```

```
    Year  GDP per capita (current US$)
0   1960                    428.056183
1   1961                    449.148137
2   1962                    472.082740
3   1963                    511.202235
4   1964                    485.530686
..   ...                           ...
59  2019                  66081.719920
60  2020                  61466.803680
61  2021                  79601.412960
62  2022                  88428.702420
63  2023                  84734.255920

[64 rows x 2 columns]
```

Extracting information for *Health Life Expectancy* (HLE)

The following function extracts Singapore's total life expectancy (TLE) at birth. The csv file is obtained from Singapore Department of Statistics (DOS).

- The data provided by DOS includes life expectancy at birth and age 65 years. WHR uses HLE at birth to measure a country's happiness. Hence, only data for life expectancy at birth will be used in this report.

```
# To import the CSV file, and read it as a Text file
df_tle = open('health_life_expectancy_1957-2023.csv', 'r')
df_tle

# To read the file
```

```
df_tle = df_tle.readlines()

# Getting Year
tle_year = df_tle[10]
# Tokenise the Year
tle_yearData = tle_year.split(",")[1:] #splitting by ',' & removing
the 1st column
tle_yearData[-1] = tle_yearData[-1].strip(" \n") #to remove \n
# print(tle_yearData)

# Getting TLE at Birth
tle_birth = df_tle[11]
# Tokenise the TLE at Birth
tle_birthData = tle_birth.split(",")[1:] #splitting by ',' & removing
the 1st column
tle_birthData[-1] = tle_birthData[-1].strip(" \n") #to remove \n
# print(tle_birthData)

# Converting to pandas DataFrame
tle_yearData_df = pd.DataFrame({'Year': tle_yearData})
tle_birth_df = pd.DataFrame({'TLE at Birth': tle_birthData})

# Combine the data into a New dataframe
newDataTLE = pd.concat([tle_yearData_df, tle_birth_df], axis=1)

# Setting the header
header = ['Year','TLE at Birth']

# Save the data to a New CSV file
newDataTLE.to_csv('new_SG_TLE.csv', header=header, index=False)
# print(newData)

# Reading the new csv data
newDataTLE_csv = pd.read_csv('new_SG_TLE.csv')
newDataTLE_csv
```

```
    Year  TLE at Birth
0   2023          83.0
1   2022          83.0
2   2021          83.2
3   2020          83.7
4   2019          83.7
5   2018          83.4
6   2017          83.2
7   2016          83.0
8   2015          82.9
9   2014          82.6
10  2013          82.4
11  2012          82.1
12  2011          81.9
```

```
13  2010          81.7
14  2009          81.4
15  2008          80.9
16  2007          80.6
17  2006          80.3
18  2005          80.1
19  2004          79.6
20  2003          79.1
21  2002          78.6
22  2001          78.3
23  2000          78.0
24  1999          77.6
25  1998          77.3
26  1997          76.9
27  1996          76.6
28  1995          76.3
29  1994          76.2
30  1993          76.1
31  1992          75.9
32  1991          75.6
33  1990          75.3
34  1989          74.9
35  1988          74.7
36  1987          74.5
37  1986          74.2
38  1985          73.9
39  1984          73.3
40  1983          73.0
41  1982          72.6
42  1981          72.5
43  1980          72.1
44  1975          66.8
45  1970          65.8
46  1965          64.5
47  1960          62.9
48  1957          61.1
```

- **Social support, Generosity, Freedom to make life choices**, and **Perception of corruption**: Data are directly sourced from the WHR. The data is derived from Gallup World Poll (GWP) questions with binary responses (0: No, 1: Yes), aggregated into a national average for each factor.
  - Unlike GDP or HLE, which are based on objective national statistics, these variables rely on subjective perspectives. While generally reliable, smaller sample sizes may occasionally cause minor variations, especially in diverse or unequal countries.

Extracting information for *Social support*

The following function extracts Singapore's social support. The excel file is obtained from World Happiness Report

- The GWP question asked for this variable was "If you were in trouble, do you have relatives or friends you can count on to help you whenever you need them, or not?"

```
# Getting Singapore's Data
rows_ss = df.loc[1836:1851, ['Country name', 'year', 'Social
support']]

# Rename the columns
rows_ss = rows_ss.rename(columns={'year': 'Year'})

# Rounding the data to 5 decimals - to match other data
x = round(0.910268962, 5)
x

# Adding in 2020 data - obtained from WHR 2020
ss2020 = pd.DataFrame([{'Country name': 'Singapore', 'Year': '2020',
'Social support': x}]) #creates a new dataframe for 2020's data
rows_ss = pd.concat([rows_ss, ss2020], ignore_index=True)
rows_ss['Year'] = rows_ss['Year'].astype(int) #changing data type to
string
rows_ss

print(rows_ss)
```

```
   Country name  Year  Social support
0     Singapore  2006        0.904329
1     Singapore  2007        0.920632
2     Singapore  2008        0.845259
3     Singapore  2009        0.866255
4     Singapore  2010        0.864162
5     Singapore  2011        0.904474
6     Singapore  2013        0.807911
7     Singapore  2014        0.822033
8     Singapore  2015        0.866437
9     Singapore  2016        0.925128
10    Singapore  2017        0.897350
11    Singapore  2018        0.902841
12    Singapore  2019        0.924918
13    Singapore  2021        0.876409
14    Singapore  2022        0.851947
15    Singapore  2023        0.916326
16    Singapore  2020        0.910270
```

Extracting information for *Perceived freedom to make life choices*

The following function extracts Singapore's perceived freedom to make life choices. The excel file is obtained from WHR.

- The GWP question asked for this variable was "Are you satisfied or dissatisfied with your freedom to choose what you do with your life?".

```python
# Getting Singapore's Data
rows_fd = df.loc[1836:1851, ['Country name', 'year', 'Freedom to make
life choices']]

# Rename the columns
rows_fd = rows_fd.rename(columns={'year': 'Year'})

# Rounding the data to 5 decimals - to match other data
z = round(0.926645339, 5)
z

# Adding in 2020 data - obtained from WHR 2020
fd2020 = pd.DataFrame([{'Country name': 'Singapore', 'Year': '2020',
'Freedom to make life choices': z}]) #creates a new dataframe for
2020's data
rows_fd = pd.concat([rows_fd, fd2020], ignore_index=True)
rows_fd['Year'] = rows_fd['Year'].astype(int) #changing data type to
string
rows_fd

print(rows_fd)
```

```
   Country name  Year  Freedom to make life choices
0     Singapore  2006                      0.756874
1     Singapore  2007                      0.866892
2     Singapore  2008                      0.660659
3     Singapore  2009                      0.776382
4     Singapore  2010                      0.846185
5     Singapore  2011                      0.821816
6     Singapore  2013                      0.827103
7     Singapore  2014                      0.834888
8     Singapore  2015                      0.886891
9     Singapore  2016                      0.903736
10    Singapore  2017                      0.926128
11    Singapore  2018                      0.916078
12    Singapore  2019                      0.938042
13    Singapore  2021                      0.878701
14    Singapore  2022                      0.873291
15    Singapore  2023                      0.861233
16    Singapore  2020                      0.926650
```

Extracting information for *Generosity*

The following function extracts Singapore's "generosity". The excel file is obtained from WHR.

- The GWP question asked for this variable was "Have you donated money to a charity in the past month?".

```
# Getting Singapore's Data
rows_gen = df.loc[1836:1851, ['Country name', 'year', 'Generosity']]

# Rename the columns
rows_gen = rows_gen.rename(columns={'year': 'Year'})

# Rounding the data to 5 decimals - to match other data
y = round(0.029879224, 5)
y

# Adding in 2020 data - obtained from WHR 2020
gen2020 = pd.DataFrame([{'Country name': 'Singapore', 'Year': '2020',
'Generosity': y}]) #creates a new dataframe for 2020's data
rows_gen = pd.concat([rows_gen, gen2020], ignore_index=True)
rows_gen['Year'] = rows_gen['Year'].astype(int) #changing data type to
string
rows_gen

print(rows_gen)
```

```
   Country name  Year  Generosity
0     Singapore  2006    0.132058
1     Singapore  2007    0.287093
2     Singapore  2008    0.039528
3     Singapore  2009   -0.081117
4     Singapore  2010   -0.024227
5     Singapore  2011   -0.154827
6     Singapore  2013    0.108811
7     Singapore  2014    0.147937
8     Singapore  2015    0.143529
9     Singapore  2016    0.136512
10    Singapore  2017    0.129031
11    Singapore  2018   -0.072510
12    Singapore  2019    0.020152
13    Singapore  2021    0.060268
14    Singapore  2022    0.088438
15    Singapore  2023         NaN
16    Singapore  2020    0.029880
```

Extracting information for Perception of corruption

The following function extracts Singapore's perception of corruption from 2006-2023. The excel file is obtained from WHR.

- The 2 GWP questions asked for this variable were "Is corruption widespread throughout the government or not", and "Is corruption widespread within businesses or not?".

```
# Getting Singapore's Data
rows_poc = df.loc[1836:1851, ['Country name', 'year', 'Perceptions of
corruption']]

# Rename the columns
rows_poc = rows_poc.rename(columns={'year': 'Year'})

# Rounding the data to 5 decimals - to match other data
w = round(0.109784193, 5)
w

# Adding in 2020 data - obtained from WHR 2020
poc2020 = pd.DataFrame([{'Country name': 'Singapore', 'Year': '2020',
'Perceptions of corruption': w}]) #creates a new dataframe for 2020's
data
rows_poc = pd.concat([rows_poc, poc2020], ignore_index=True)
rows_poc['Year'] = rows_poc['Year'].astype(int) #changing data type to
string
rows_poc

print(rows_poc)
```

```
   Country name  Year  Perceptions of corruption
0     Singapore  2006                        NaN
1     Singapore  2007                   0.063615
2     Singapore  2008                   0.065775
3     Singapore  2009                   0.035198
4     Singapore  2010                   0.060282
5     Singapore  2011                   0.098924
6     Singapore  2013                   0.242398
7     Singapore  2014                   0.132603
8     Singapore  2015                   0.098944
9     Singapore  2016                   0.047311
10    Singapore  2017                   0.161791
11    Singapore  2018                   0.096563
12    Singapore  2019                   0.069620
13    Singapore  2021                   0.144935
14    Singapore  2022                        NaN
15    Singapore  2023                   0.152543
16    Singapore  2020                   0.109780
```

3. Data cleaning and processing

Data for Singapore's birth rate will first be compared with data for Singapore's happiness level accumulated over the years.

Things to note about World Happiness Report (WHR) and Data

- There were no WHR published for the year 2014, so for WHR published in 2015, some of the data were estimated, re-calculated, and re-adjusted based on data from previous years, some based on assumptions, or taking average data based on other reliable sources.

Time ranges provided in both datasets do not fully align

- Data for Singapore's birth rate provides a long-term view of data trends from 1967 until 2023 (latest available).
- Data for Singapore's happiness level was only available starting from 2006 onwards until 2023, excluding 2014.

Since the purpose of this report was to find out about the relationship between happiness levels and birthrate trends, having a complete dataset for analysis would help to avoid gaps and inconsistencies in the data output later on. For missing data, the report will calculate the median values of previous years.

3.1 Data cleaning

3.1.1 Recalculate missing values

This part is done because based on part 2, not all the data was able to be seen. By checking the list, makes it easier and saves time too.

Checking for missing data for *GDP per capita*

```
# Check for missing values
missing_gdp = newDataGDP_csv.isnull().sum()
missing_gdp

Year                          0
GDP per capita (current US$)    0
dtype: int64
```

Checking for missing data and recalculating them for *Generosity*

```
# Check for missing values
missing_generosity = rows_gen.isnull().sum()

# Calculate median of the column (excluding the NaN values)
median_generosity = rows_gen['Generosity'].median()

# Replacing the NaN (the missing value) with the median
rows_gen.fillna({'Generosity': median_generosity}, inplace=True)
rows_gen

   Country name  Year  Generosity
0     Singapore  2006    0.132058
1     Singapore  2007    0.287093
2     Singapore  2008    0.039528
3     Singapore  2009   -0.081117
4     Singapore  2010   -0.024227
```

```
5     Singapore  2011   -0.154827
6     Singapore  2013    0.108811
7     Singapore  2014    0.147937
8     Singapore  2015    0.143529
9     Singapore  2016    0.136512
10    Singapore  2017    0.129031
11    Singapore  2018   -0.072510
12    Singapore  2019    0.020152
13    Singapore  2021    0.060268
14    Singapore  2022    0.088438
15    Singapore  2023    0.074353
16    Singapore  2020    0.029880
```

Checking for missing data and recalculating them for *Perception of corruption*

```
# Check for missing values
missing_poc = rows_poc.isnull().sum()

# To interpolate data - skips the 1st data
rows_poc['Perceptions of corruption'] = rows_poc['Perceptions of 
corruption'].interpolate(method='linear')
# # To fill NaN for 2006's data
rows_poc['Perceptions of corruption'] = rows_poc['Perceptions of 
corruption'].bfill()
rows_poc

   Country name  Year  Perceptions of corruption
0     Singapore  2006                   0.063615
1     Singapore  2007                   0.063615
2     Singapore  2008                   0.065775
3     Singapore  2009                   0.035198
4     Singapore  2010                   0.060282
5     Singapore  2011                   0.098924
6     Singapore  2013                   0.242398
7     Singapore  2014                   0.132603
8     Singapore  2015                   0.098944
9     Singapore  2016                   0.047311
10    Singapore  2017                   0.161791
11    Singapore  2018                   0.096563
12    Singapore  2019                   0.069620
13    Singapore  2021                   0.144935
14    Singapore  2022                   0.148739
15    Singapore  2023                   0.152543
16    Singapore  2020                   0.109780
```

3.1.2 Removing irrelavant datasets

This part of the report removes datasets from previous years. This is so to ensure a complete dataset with consistencies, without any gaps, is obtained.

Removing data for *Annual Life-Birth*

```
# Removing rows Before 2006
newDataBR_csv = newDataBR_csv[newDataBR_csv['Year'] > 2005]
newDataBR_csv

    Year  Birthrate
0   2023      33541
1   2022      35605
2   2021      38672
3   2020      38590
4   2019      39279
5   2018      39039
6   2017      39615
7   2016      41251
8   2015      42185
9   2014      42232
10  2013      39720
11  2012      42663
12  2011      39654
13  2010      37967
14  2009      39570
15  2008      39826
16  2007      39490
17  2006      38317
```

Removing data for *GDP per capita*

```
# Removing rows Before 2006
newDataGDP_csv = newDataGDP_csv[newDataGDP_csv['Year'] > 2005]
newDataGDP_csv

    Year  GDP per capita (current US$)
46  2006                   33768.45282
47  2007                   39432.88602
48  2008                   40008.57205
49  2009                   38926.80544
50  2010                   47236.68308
51  2011                   53891.45703
52  2012                   55547.55531
53  2013                   56967.42579
54  2014                   57564.80231
55  2015                   55645.60686
56  2016                   56899.91818
57  2017                   61162.09739
58  2018                   66840.63734
59  2019                   66081.71992
60  2020                   61466.80368
61  2021                   79601.41296
62  2022                   88428.70242
63  2023                   84734.25592
```

Removing data for *Health Life Expectancy* (HLE)

```
# Removing rows Before 2006
newDataTLE_csv = newDataTLE_csv[newDataTLE_csv['Year'] > 2005]
newDataTLE_csv

    Year  TLE at Birth
0   2023          83.0
1   2022          83.0
2   2021          83.2
3   2020          83.7
4   2019          83.7
5   2018          83.4
6   2017          83.2
7   2016          83.0
8   2015          82.9
9   2014          82.6
10  2013          82.4
11  2012          82.1
12  2011          81.9
13  2010          81.7
14  2009          81.4
15  2008          80.9
16  2007          80.6
17  2006          80.3
```

3.1.3 Sorting datasets

This part of the report sorts the datasets into decreasing order, based on the year. This is to allow for easier comparison and merging of the data later on.

Sorting data for *Happiness Score*

```
# Sort the data into descending order - to match other data
rows_hap = rows_hap.sort_values(by='Year')
print(rows_hap)

   Country name  Year HappinessScore
0     Singapore  2006       6.462703
1     Singapore  2007       6.833755
2     Singapore  2008       6.641957
3     Singapore  2009       6.144677
4     Singapore  2010       6.531402
5     Singapore  2011       6.561042
6     Singapore  2013       6.533207
7     Singapore  2014       7.062365
8     Singapore  2015       6.619525
9     Singapore  2016       6.033481
10    Singapore  2017       6.378438
11    Singapore  2018       6.374564
12    Singapore  2019        6.37836
```

```
16    Singapore  2020       6.377091
13    Singapore  2021       6.586717
14    Singapore  2022       6.333046
15    Singapore  2023       6.653942
```

Sorting data for *GDP*

```
# Sort the data into descending order - to match other data
newDataGDP_csv = newDataGDP_csv.sort_values(by='Year',
ascending=False)
print(newDataGDP_csv)
```

```
    Year  GDP per capita (current US$)
63  2023                   84734.25592
62  2022                   88428.70242
61  2021                   79601.41296
60  2020                   61466.80368
59  2019                   66081.71992
58  2018                   66840.63734
57  2017                   61162.09739
56  2016                   56899.91818
55  2015                   55645.60686
54  2014                   57564.80231
53  2013                   56967.42579
52  2012                   55547.55531
51  2011                   53891.45703
50  2010                   47236.68308
49  2009                   38926.80544
48  2008                   40008.57205
47  2007                   39432.88602
46  2006                   33768.45282
```

Sorting data for *Social support*

```
# Sort the data into descending order - to match other data
rows_ss = rows_ss.sort_values(by='Year', ascending=False)
print(rows_ss)
```

```
   Country name  Year  Social support
15    Singapore  2023        0.916326
14    Singapore  2022        0.851947
13    Singapore  2021        0.876409
16    Singapore  2020        0.910270
12    Singapore  2019        0.924918
11    Singapore  2018        0.902841
10    Singapore  2017        0.897350
9     Singapore  2016        0.925128
8     Singapore  2015        0.866437
7     Singapore  2014        0.822033
6     Singapore  2013        0.807911
```

```
5     Singapore  2011        0.904474
4     Singapore  2010        0.864162
3     Singapore  2009        0.866255
2     Singapore  2008        0.845259
1     Singapore  2007        0.920632
0     Singapore  2006        0.904329
```

Sorting data for *Perceived freedom to make life choices*

```
# Sorting the data into descending order - to match other data
rows_fd = rows_fd.sort_values(by='Year', ascending=False)
print(rows_fd)

   Country name  Year  Freedom to make life choices
15    Singapore  2023                      0.861233
14    Singapore  2022                      0.873291
13    Singapore  2021                      0.878701
16    Singapore  2020                      0.926650
12    Singapore  2019                      0.938042
11    Singapore  2018                      0.916078
10    Singapore  2017                      0.926128
9     Singapore  2016                      0.903736
8     Singapore  2015                      0.886891
7     Singapore  2014                      0.834888
6     Singapore  2013                      0.827103
5     Singapore  2011                      0.821816
4     Singapore  2010                      0.846185
3     Singapore  2009                      0.776382
2     Singapore  2008                      0.660659
1     Singapore  2007                      0.866892
0     Singapore  2006                      0.756874
```

Sorting data for *Generosity*

```
# Sort the data into descending order - to match other data
rows_gen = rows_gen.sort_values(by='Year', ascending=False)
print(rows_gen)

   Country name  Year  Generosity
15    Singapore  2023    0.074353
14    Singapore  2022    0.088438
13    Singapore  2021    0.060268
16    Singapore  2020    0.029880
12    Singapore  2019    0.020152
11    Singapore  2018   -0.072510
10    Singapore  2017    0.129031
9     Singapore  2016    0.136512
8     Singapore  2015    0.143529
7     Singapore  2014    0.147937
6     Singapore  2013    0.108811
```

```
5     Singapore  2011   -0.154827
4     Singapore  2010   -0.024227
3     Singapore  2009   -0.081117
2     Singapore  2008    0.039528
1     Singapore  2007    0.287093
0     Singapore  2006    0.132058
```

Sorting data for *Perceptions of Corruption*

```
# Sorting the data into descending order - to match other data
rows_poc = rows_poc.sort_values(by='Year', ascending=False)
print(rows_poc)

   Country name  Year  Perceptions of corruption
15    Singapore  2023                   0.152543
14    Singapore  2022                   0.148739
13    Singapore  2021                   0.144935
16    Singapore  2020                   0.109780
12    Singapore  2019                   0.069620
11    Singapore  2018                   0.096563
10    Singapore  2017                   0.161791
9     Singapore  2016                   0.047311
8     Singapore  2015                   0.098944
7     Singapore  2014                   0.132603
6     Singapore  2013                   0.242398
5     Singapore  2011                   0.098924
4     Singapore  2010                   0.060282
3     Singapore  2009                   0.035198
2     Singapore  2008                   0.065775
1     Singapore  2007                   0.063615
0     Singapore  2006                   0.063615
```

3.2 Combine data

Combine ***Birth rate*** with ***Happiness score (ladder)*** into 1 dataset

This part of the report combines the data of both the Birth rate and the ladder only, so as to compare them later on.

```
newDataBR_csv.loc[:, 'Year'] = newDataBR_csv['Year'].astype(int)
#changing data type to string

# Combine the data
combine_BR_hap = pd.merge(newDataBR_csv, rows_hap, on='Year')
combine_BR_hap = combine_BR_hap.drop(columns=['Country name']) # to
remove 'country name''s column
combine_BR_hap

    Year  Birthrate HappinessScore
0   2023      33541       6.653942
```

```
1   2022      35605       6.333046
2   2021      38672       6.586717
3   2020      38590       6.377091
4   2019      39279        6.37836
5   2018      39039       6.374564
6   2017      39615       6.378438
7   2016      41251       6.033481
8   2015      42185       6.619525
9   2014      42232       7.062365
10  2013      39720       6.533207
11  2011      39654       6.561042
12  2010      37967       6.531402
13  2009      39570       6.144677
14  2008      39826       6.641957
15  2007      39490       6.833755
16  2006      38317       6.462703
```

Combine ***Birth rate*** with the all the ***variables***

```
# Combine the data: Birthrate X Gross Domestic Product(GDP) per person
combine_BR_GDP = pd.merge(newDataBR_csv, newDataGDP_csv, on='Year')
combine_BR_GDP

# Combine the data: Birthrate X Healthy life expectancy (HLE)
combine_BR_TLE = pd.merge(newDataBR_csv, newDataTLE_csv, on='Year')
combine_BR_TLE

# Combine the data: Birthrate X Social support
combine_BR_SS = pd.merge(newDataBR_csv, rows_ss, on='Year')
combine_BR_SS = combine_BR_SS.drop(columns=['Country name']) # to
remove 'country name''s column
combine_BR_SS

# Combine the data: Birthrate X Perceived freedom to make life choices
combine_BR_FD = pd.merge(newDataBR_csv, rows_fd, on='Year')
combine_BR_FD = combine_BR_FD.drop(columns=['Country name']) # to
remove 'country name''s column
combine_BR_FD

# Combine the data: Birthrate X Generosity
combine_BR_GEN = pd.merge(newDataBR_csv, rows_gen, on='Year')
combine_BR_GEN = combine_BR_GEN.drop(columns=['Country name']) # to
remove 'country name''s column
combine_BR_GEN

# Combine the data: Birthrate X Perception of corruption
combine_BR_POC = pd.merge(newDataBR_csv, rows_poc, on='Year')
combine_BR_POC = combine_BR_POC.drop(columns=['Country name']) # to
remove 'country name''s column
combine_BR_POC
```

```
    Year  Birthrate  Perceptions of corruption
0   2023      33541                   0.152543
1   2022      35605                   0.148739
2   2021      38672                   0.144935
3   2020      38590                   0.109780
4   2019      39279                   0.069620
5   2018      39039                   0.096563
6   2017      39615                   0.161791
7   2016      41251                   0.047311
8   2015      42185                   0.098944
9   2014      42232                   0.132603
10  2013      39720                   0.242398
11  2011      39654                   0.098924
12  2010      37967                   0.060282
13  2009      39570                   0.035198
14  2008      39826                   0.065775
15  2007      39490                   0.063615
16  2006      38317                   0.063615
```

Combine ***Birth rate*** with the all the ***variables*** into 1 dataset

```
# Combining ALL the data
combined_df = combine_BR_GDP.copy() #using Birthrate x GDP's data
combined_df = combined_df.merge(combine_BR_TLE[['Year', 'TLE at
Birth']], on='Year', how='outer') # add TLE's data
combined_df = combined_df.merge(combine_BR_SS[['Year', 'Social
support']], on='Year', how='outer') # add SS's data
combined_df = combined_df.merge(combine_BR_FD[['Year', 'Freedom to
make life choices']], on='Year', how='outer') # add FD's data
combined_df = combined_df.merge(combine_BR_GEN[['Year',
'Generosity']], on='Year', how='outer') # add GEN's data
combined_df = combined_df.merge(combine_BR_POC[['Year', 'Perceptions
of corruption']], on='Year', how='outer') # add POC's data
combined_df
```

```
    Year  Birthrate  GDP per capita (current US$)  TLE at Birth  \
0   2006      38317                   33768.45282          80.3
1   2007      39490                   39432.88602          80.6
2   2008      39826                   40008.57205          80.9
3   2009      39570                   38926.80544          81.4
4   2010      37967                   47236.68308          81.7
5   2011      39654                   53891.45703          81.9
6   2012      42663                   55547.55531          82.1
7   2013      39720                   56967.42579          82.4
8   2014      42232                   57564.80231          82.6
9   2015      42185                   55645.60686          82.9
10  2016      41251                   56899.91818          83.0
11  2017      39615                   61162.09739          83.2
12  2018      39039                   66840.63734          83.4
13  2019      39279                   66081.71992          83.7
```

```
14  2020      38590                   61466.80368          83.7
15  2021      38672                   79601.41296          83.2
16  2022      35605                   88428.70242          83.0
17  2023      33541                   84734.25592          83.0

    Social support  Freedom to make life choices  Generosity  \
0         0.904329                      0.756874    0.132058  
1         0.920632                      0.866892    0.287093  
2         0.845259                      0.660659    0.039528  
3         0.866255                      0.776382   -0.081117  
4         0.864162                      0.846185   -0.024227  
5         0.904474                      0.821816   -0.154827  
6              NaN                           NaN         NaN  
7         0.807911                      0.827103    0.108811  
8         0.822033                      0.834888    0.147937  
9         0.866437                      0.886891    0.143529  
10        0.925128                      0.903736    0.136512  
11        0.897350                      0.926128    0.129031  
12        0.902841                      0.916078   -0.072510  
13        0.924918                      0.938042    0.020152  
14        0.910270                      0.926650    0.029880  
15        0.876409                      0.878701    0.060268  
16        0.851947                      0.873291    0.088438  
17        0.916326                      0.861233    0.074353  

    Perceptions of corruption
0                    0.063615
1                    0.063615
2                    0.065775
3                    0.035198
4                    0.060282
5                    0.098924
6                         NaN
7                    0.242398
8                    0.132603
9                    0.098944
10                   0.047311
11                   0.161791
12                   0.096563
13                   0.069620
14                   0.109780
15                   0.144935
16                   0.148739
17                   0.152543
```

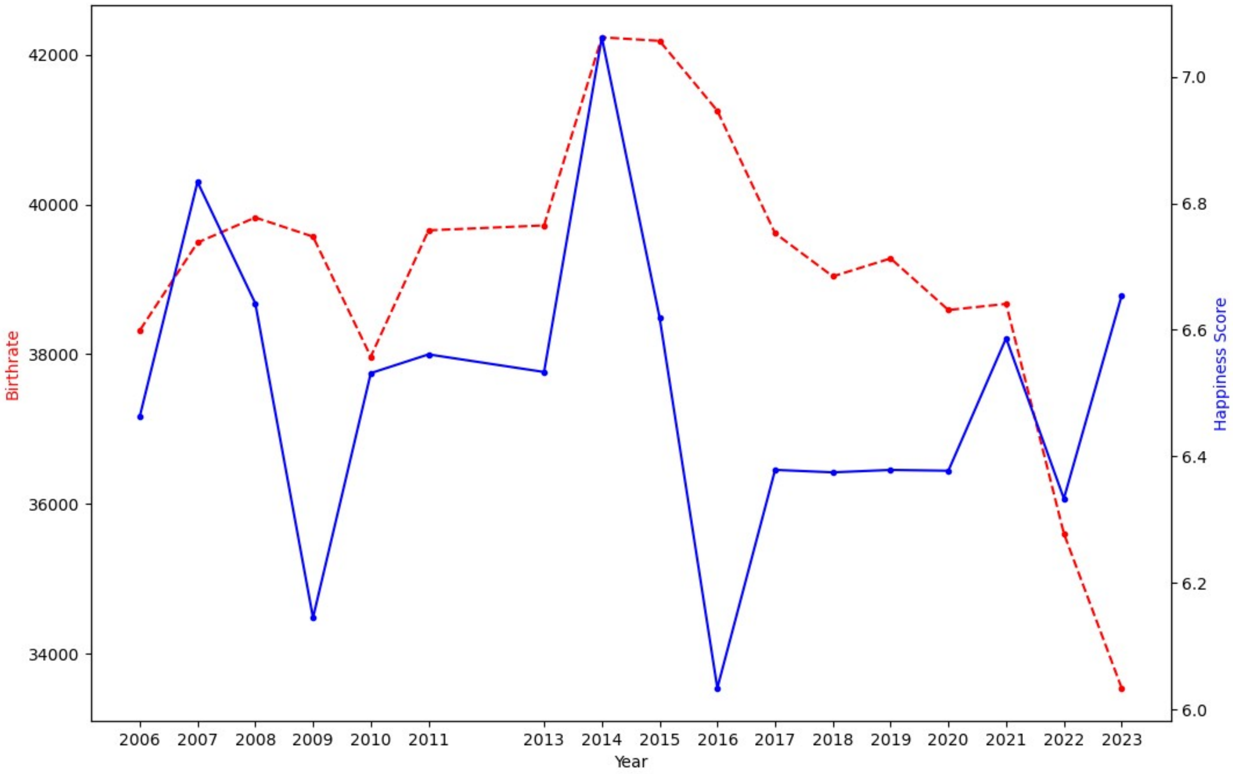
3.3 Comparing data

Comparing ***Birth rate*** with ***Happiness score (ladder)***

This part of the report compares the data on Birth rate with each variable using the '**subplots**' function in *Matplotlib*.

- Birthrate: red-dotted line
- Each variable: blue line

```
# Extract columns from the dataframe
Year = combine_BR_hap['Year']
Birthrate = combine_BR_hap['Birthrate']
HappinessScore = combine_BR_hap['HappinessScore']

# Plotting the Graph
fig, combine_BR_hap1 = plt.subplots(figsize=(12, 8))
combine_BR_hap2 = combine_BR_hap1.twinx()
# Axis setting - data for x & y axis, color of line, line style
combine_BR_hap1.plot(Year, Birthrate, color='red', marker='.', linestyle='--')
combine_BR_hap2.plot(Year, HappinessScore, color='blue', marker='.')

# To show integer values for 'Year'
combine_BR_hap1.set_xticks(Year)

# Graph setting
combine_BR_hap1.set_xlabel("Year") # x-axis title
combine_BR_hap1.set_ylabel("Birthrate", color='red') # LEFT y-axis title color
combine_BR_hap2.set_ylabel("Happiness Score", color='blue') # RIGHT y-axis title color

plt.show()
```



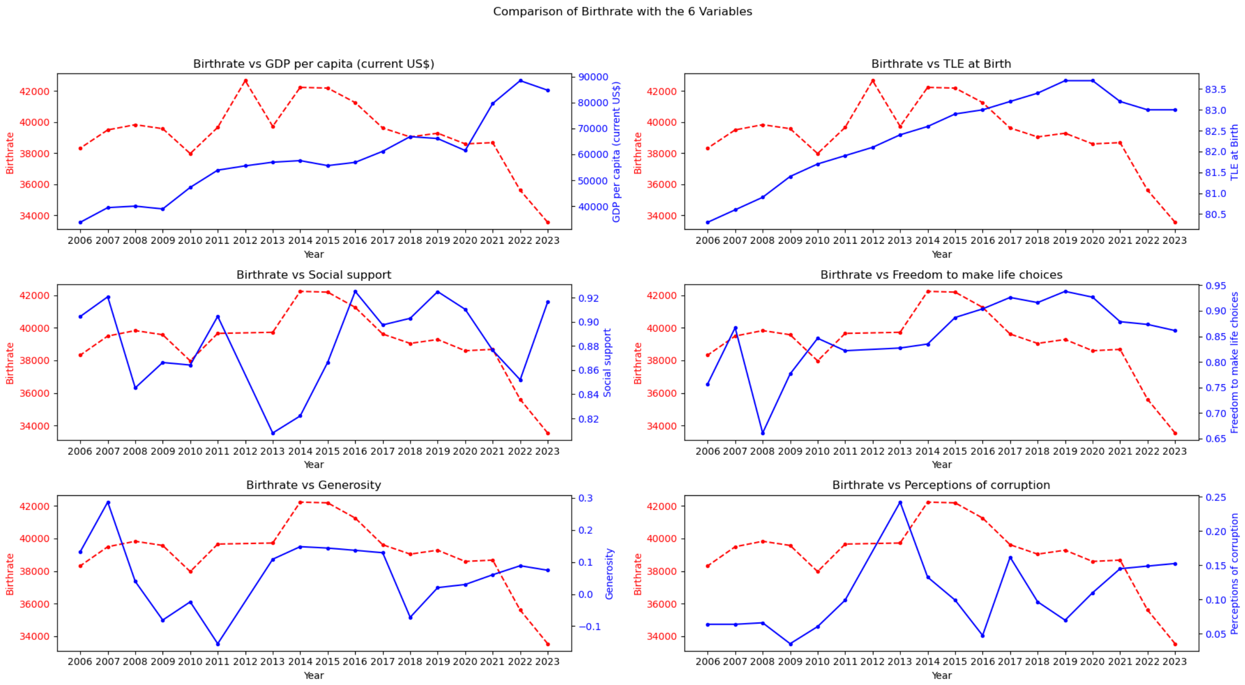
Comparing ***Birth rate*** with each ***individual variables***

This part of the report compares the data on Birth rate with each variable using the '**subplots**' function in *Matplotlib*.

- Birthrate: red-dotted line
- Each variable: blue line

```
# Array to store all the 6 Variables
variables = [
    # (varDF, var_name)
    (combine_BR_GDP, 'GDP per capita (current US$)'),
    (combine_BR_TLE, 'TLE at Birth'),
    (combine_BR_SS, 'Social support'),
    (combine_BR_FD, 'Freedom to make life choices'),
    (combine_BR_GEN, 'Generosity'),
    (combine_BR_POC, 'Perceptions of corruption')
]

# Creating subplots
fig, axes = plt.subplots(ncols=2, nrows=3, figsize=(18, 10))

# 'axes' - is a 2D array. 'flatten' it to convert it to 1D array - 
easier for iteration
axes = axes.flatten()

# Create subplots for each of the variable, while looping through the
```

```
array
for i, (varDF, var_name) in enumerate(variables):
    # Create the Dual y-axis graph
    BR = axes[i]
    BR_var = BR.twinx()

    # Plot birthrate on the LEFT_y-axis
    BR.plot(varDF['Year'], varDF['Birthrate'], color='red',
marker='.', linestyle='--', label='Birthrate')
    # Graph setting
    BR.set_xlabel('Year') # x-axis title
    BR.set_ylabel('Birthrate', color='red') # LEFT_y-axis title_data,
title_color
    BR.tick_params(axis='y', labelcolor='red') # LEFT_y-axis color

    # To show all values for x-axis
    BR.set_xticks(combined_df['Year'])

    # Plot variable on the RIGHT_y-axis
    BR_var.plot(varDF['Year'], varDF[var_name], color='blue',
marker='.', alpha=1, label=var_name)
    # Graph setting
    BR_var.set_ylabel(var_name, color='blue') # RIGHT_y-axis
title_data, title_color
    BR_var.tick_params(axis='y', labelcolor='blue') # RIGHT_y-axis
color

    # Title
    BR.set_title(f'Birthrate vs {var_name}')

# Title
fig.suptitle('Comparison of Birthrate with the 6 Variables')

# Layout settings
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```

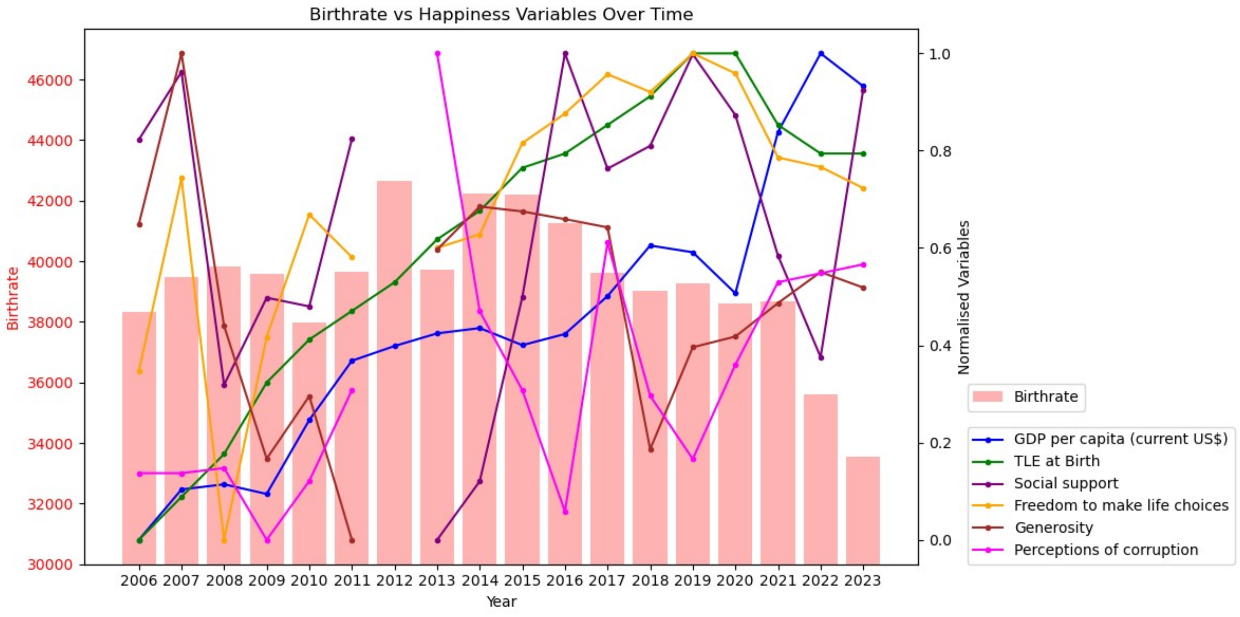


Comparing ***Birth rate*** with each ***individual variables*** in one chart

This part of the report compares the data on Birth rate with each variable in a single chart.

- Birthrate: red bar
- Each variable: colored line

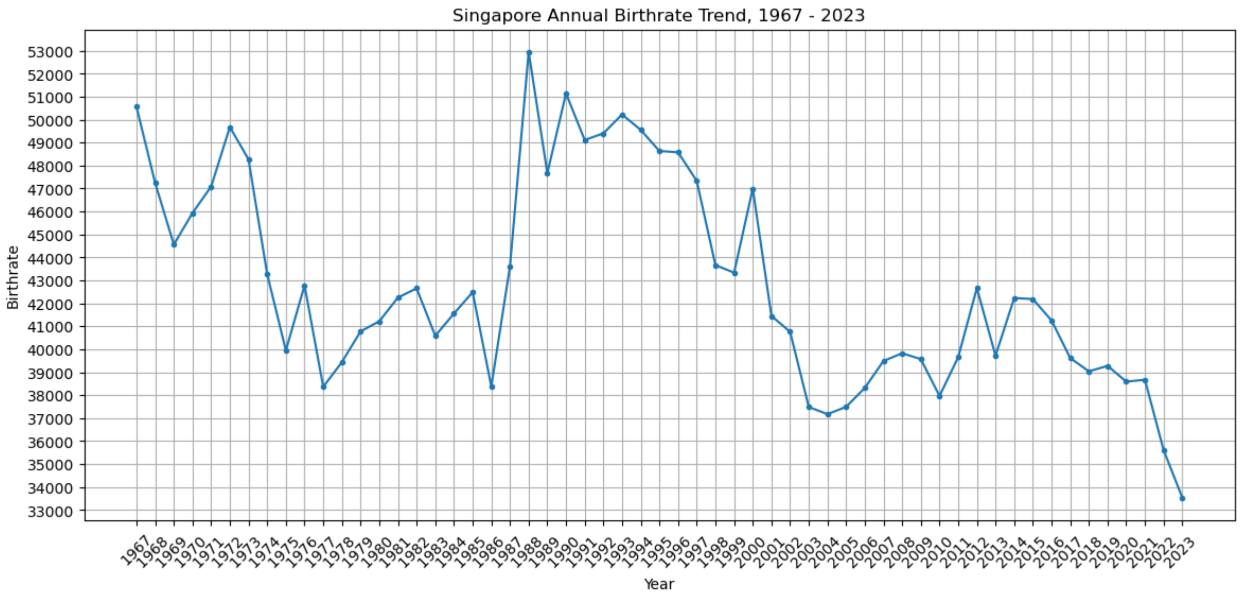
```
# Array to store all the 6 Variables
variable_single = [
    'GDP per capita (current US$)',
    'TLE at Birth',
    'Social support',
    'Freedom to make life choices',
    'Generosity',
    'Perceptions of corruption']

# To normalise the variable to all Having the Same Range
for var in variable_single:
    combined_df[var] = (combined_df[var] - combined_df[var].min()) /
(combined_df[var].max() - combined_df[var].min())

# Create the Dual y-axis graph
fig, BR_single = plt.subplots(figsize=(12, 6))
all_var = BR_single.twinx()

# Plot birthrate on the LEFT y-axis
BR_single.bar(combined_df['Year'], combined_df['Birthrate'],
color='red', alpha=0.3, label='Birthrate')
BR_single.set_xlabel('Year')
BR_single.set_ylabel('Birthrate', color='red')
BR_single.tick_params(axis='y', labelcolor='red')

BR_single.set_ylim(30000, combined_df['Birthrate'].max() + 5000)

# To show all values for x-axis
BR_single.set_xticks(combined_df['Year'])

# Colors for each variable
colors = ['blue', 'green', 'purple', 'orange', 'brown', 'magenta']
for var, color in zip(variable_single, colors):
    all_var.plot(combined_df['Year'], combined_df[var], label=var,
color=color, marker='.')

# Plot all normalized variables on the RIGHT y-axis
all_var.set_ylabel('Normalised Variables', color='black')
all_var.tick_params(axis='y', labelcolor='black')

# Legend
BR_single.legend(loc='upper left', bbox_to_anchor=(1.05, 0.35))
all_var.legend(loc='upper left', bbox_to_anchor=(1.05, 0.27))

# Title
plt.title('Birthrate vs Happiness Variables Over Time')

# Layout settings
plt.tight_layout()
plt.show()
```



4. Data Analysis

Graph of the *Birthrate* trend in Singapore from 1967 till 2023

```
newDataBR['Birthrate'] = pd.to_numeric(newDataBR['Birthrate'],
errors='coerce')

# Sort into ascending year data - to show latest at the RIGHT of x-
axis
newDataBR = newDataBR.sort_values(by='Year', ascending=True)
newDataBR

# Extract columns from the dataframe
YearBR = newDataBR['Year']
BirthrateBR = newDataBR['Birthrate']

# Plotting the graph
fig, BR_graph = plt.subplots(figsize=(14, 6))
BR_graph.plot(YearBR, BirthrateBR, marker='.')

# Determine y-axis range dynamically
y_min = int(BirthrateBR.min() // 1000 * 1000)  # Round down to nearest
1000
y_max = int(BirthrateBR.max() // 1000 * 1000 + 1000)  # Round up to
nearest 1000
y_ticks = np.arange(y_min, y_max + 1, 1000)  # Interval between each
'tick'

# Set y-axis
BR_graph.set_yticks(y_ticks)
# Set x-axis
plt.xticks(rotation=45)
# Set grid
BR_graph.grid(True)

# Title
BR_graph.set_title('Singapore Annual Birthrate Trend, 1967 - 2023')
BR_graph.set_xlabel('Year')
BR_graph.set_ylabel('Birthrate')

plt.show()
```



Analysis of Birthrate Trends

- The graph illustrates that Singapore's birthrate has been steadily declining since 1967.
- However, occasional spikes in birthrate are evident in certain years, followed by subsequent declines.
- These fluctuations can be attributed to cultural beliefs and traditions prevalent among Singaporeans:
  - Although Singapore is a multicultural society, approximately 74% of its population is of Chinese ethnicity.
  - According to Chinese folklore, children born in the Year of the Dragon are believed to be wise and charismatic, making this year highly favorable for childbirth. Conversely, the Year of the Tiger is less desirable, as children born in this year are thought to possess wild or stubborn traits.
  - This cultural preference leads to a cyclical rise and fall in birthrates around these sodiac years. This can be seen evidently from the chart. For instance, spikes in birthrate occurred during the Year of the Dragon in 1988 and 2000, while noticeable declines were observed in the Year of the Tiger in 1986 and 1998.

Implemented Policies

- Singapore's changing birthrate trends are also influenced by government policies aimed at addressing population growth and fertility rates over the years.

1960s

- During the 1960s, rapid population growth posed significant socio-economic challenges. Large families often faced financial instability, leading to increased school dropouts and social issues, which in turn negatively impacted public safety and the economy.
- To tackle these issues, the government launched the *National Family Planning Programme*, with the primary goal of reducing the birthrate and achieving zero population growth to stabilise the population size.

1972

- The *National Family Planning Programme* was effective, but between 1970 to 1972, birthrates spiked due to post-war baby boomers and families opting for at least three children.
- To address this, the government introduced the *Two-child Policy* as a population control measure, implementing actions such as:
  - Disincentives for registering a third child in schools
  - Reduction in paid maternity leave from three confinements to two
  - Lower housing priority for larger families
- The main goal of the *Two-child Policy* was similar as the previous one, to stabilise the population at an optimal size.

1987

- The *Two-child Policy* proved too effective, resulting in a fall in birthrate. As a response, the government ended the policy and introduced a new *pro-natalist population policy* with the slogan "Have three, or more if you can afford it".
- Measures taken before to prevent families from having a 3rd child was removed and new incentives were implemented such as:
  - Tax rebates
  - Childcare subsidies
- The main goal of the *pro-natalist population policy* was to encourage parents to have more children and reverse the declining fertility trend.

2001

- Building on the *pro-natalist population policy*, the government introduced the *Baby Bonus Scheme* in April 2001 to further incentivise parenthood.
- The scheme aimed to ease the financial burden of raising children and encourage higher birthrates by offering financial support to parents.
- Despite these efforts and the continuous increase in incentives, Singapore's birthrate continues to decline annually.

4.1 Analysing the 'Birthrate VS Happiness score (ladder)' graph

Key observations:

- From the graph, it shows a negative correlation between birthrate and happiness score over the years. As the happiness score increases, the birthrate shows a declining trend.
- The decreasing trend suggests that higher happiness levels, as measured by the ladder score, do not necessarily lead to higher birthrates.

Comparative Analysis:

- This negative correlation suggests that factors contributing to increased happiness—such as economic stability, healthcare advancements, and personal freedom—do not directly translate to higher birthrates.

Possible causes:

- Rising happiness may correlate with improved quality of life, but this does not directly address the financial and societal pressures associated with raising children.
- Cultural shifts toward prioritising individual goals and career advancement over family planning could play a role.

4.2 Analysing the 'Comparison of Birthrate with the 6 Variables' graph

Key observations:

- Each variable exhibits different levels of correlation with birthrate.
- ***GDP***: showed a negative correlation with birthrate, indicating that as economic development improves, birthrates decline.
- ***Social support***, ***Freedom of choice***, ***perceptions of corruption***, ***generosity***: Exhibited weak correlations with birthrate, suggesting they are *not* primary factors in influencing this trend.

Comparative Analysis:

- Financial factors, such as ***GDP***, likely reflect the rising cost of living and childcare in economically developed nations, which discourages larger families. This trend aligns with urbanization and the preference for smaller, more financially sustainable households.

Possible causes:

- While the weaker variables (e.g., Social Support and Freedom of Choice) do not directly influence birthrate trends, they may *indirectly* affect individual or societal perspectives on starting a family.
- ***Social support*** could alleviate some of the financial burdens of parenting, but its impact appears limited in the observed data.
- In developed countries, people often delay parenthood, as prioritising education, career advancement, and personal goals are more preferred, resulting in smaller family sizes.
- Increased ***Freedom of choice*** can lead individuals to delay or forgo having children prioritising personal aspirations and experiences instead.
- An increase in ***Perceptions of corruption*** might discourage individuals from starting families due to concerns about instability or fairness in societal structures, prompting them to focus on career stability first.

4.3 Analysing the 'Birthrate VS Happiness Variables over time' graph

Key observations:

- Birthrate shows a consistent decline over time, while variables like ***GDP*** and ***Freedom of choice*** exhibit upward trends.
- Despite societal and economic improvements, these trends do not counteract the declining birthrate.

Comparative Analysis:

- The rising trends in variables like Freedom of Choice suggest improved societal conditions, yet these improvements do not address core challenges related to raising a child, such as financial constraints.
- This suggests that external pressures or intrinsic societal values are at play.

Possible causes:

- Even as happiness-related variables improve, factors like increased costs of living and childcare, as well as the challenges of raising children in competitive environments, might overshadow these improvements.
- The rising opportunity cost of childbearing, particularly for working parents, can discourage larger families.

4.4 Other possible reason

- **Financial Constraints**: The cost of raising a child, from education to healthcare, has increased significantly compared to past decade. This financial burden discourages larger family sizes and may delay decisions to have children.
- **Workplace Competition**: As global economies become more competitive, individuals are prioritizing career advancement and professional stability over starting families. The pressure to succeed in the workplace often outweighs the desire to have children, particularly in urbanized and developed regions.
- **Evolving Lifestyles**: Modern lifestyles often emphasise personal freedom and experiences, such as travel, entrepreneurship, hobbies, over traditional family structures. These shifts in priorities reflect changing societal norms, where individual aspirations take precedence over familial responsibilities.
- **Delayed Parenthood**: With increased access to education and career opportunities, many individuals are delaying marriage and parenthood. This postponement reduces the overall fertility window. For highly educated individuals, the idea of giving up years of academic and professional effort to start a family seems impractical.
- **Impact of Current Events**: The global pandemic and resulting economic recession have exacerbated financial insecurity, leading to inflation and heightened stress levels. In this challenging current economy, many people are hesitant to take on additional financial and emotional responsibilities, such as starting a family.
- **Mental Health and Stress**: Economic instability and job losses have significantly increased stress levels, adversely affecting both mental and physical health. Chronic stress can lead to infertility and diminished readiness for parenthood. Poor mental health, compounded by societal pressures, further discourages individuals from considering having children.

4.5 Some ideas for improvement

Despite significant government efforts, Singapore's birthrate policies have yielded limited success. To address the declining birthrate and its societal implications, the following multi-faceted approach is suggested:

1. **Enhancing Financial Incentives**:
   - **Current measures**: The Singapore government has enhanced existing initiatives in 2024, including doubling paternity leave and increasing working mother

  grants. However, these measures are insufficient due to the ongoing inflation, which diminishes the value of financial assistance.
  - **Proposed action**: Instead of solely increasing grants, the government could focus on reducing the financial burden of education. For instance, cutting tuition fees for local university students, regardless of their school, could provide long-term relief for parents and encourage family expansion.
2. **Promoting work-life balance**: A 4-day workweek has been shown to reduce stress, improve focus, and increase productivity in countries where it is implemented. This will be a kill-two-birds-with-one-stone situation, as parents can have more flexibility in working, and the company will not have any losses either. Additionally, implementing a 4-day workweek could allow parents more time to rest and spend with their children. Evidence from other countries shows that reduced workweeks lead to lower stress levels and increased productivity, benefiting both employees and employers.
3. **Addressing Societal Pressures**: In Singapore's highly competitive economy, individuals often prioritize career advancement over family life. To counter this, the government could launch campaigns promoting work-life balance and family-oriented aspirations. Leveraging media and education systems to emphasize the long-term benefits of parenthood for both individuals and society could help reshape societal norms and encourage larger families.
4. **More support for single parents**: With rising divorce rates, there is an increasing number of single-parent households requiring targeted support. The government could introduce targeted support programs to alleviate the financial and emotional burdens faced by single parents and their children, such as:
   - Additional childcare subsidies
   - Housing benefits
   - Community support networks to alleviate the financial and emotional burdens faced by single parents
5. **Implement more events for singles to meet**: In today's fast-paced and competitive work environment, many singles struggle to meet potential partners outside their immediate circles. To address this, the government could:
   - Organize events and initiatives to connect singles
   - Creating online resources/platforms to support matchmaking and social networking

5. Summary

Conclusion

This report explores Singapore's declining birthrate trend and its relationship with happiness and its variables. It highlights the interplay of cultural, financial, and societal factors. Key takeaways include:

- **Long-Term trends**: Singapore's birthrate has shown a steady decline since 1967 despite periodic fluctuations mostly driven by the rising economy, making, and governmental policy changes. This underscores the interplay between cultural traditions and modern societal trends.
- **Economic and Social Influences**:

  - Variables like ***GDP*** negatively correlate with birthrates, reflecting the high cost of living and childcare in developed economies.
  - While happiness-related factors, such as social support and freedom of choice, have improved over time, they show weak direct correlations with birthrate trends. Instead, they indirectly influence decisions through changing lifestyle preferences and priorities.
- **Policy Effectiveness**: Government policies, such as the *Two-Child Policy* and the *Baby Bonus Scheme*, had mixed results. While early measures effectively curbed population growth, later pro-natalist efforts struggled to reverse the fertility decline. This underscores the complexity of influencing birthrates in a modern, urbanized society.
- **Global Context**: Singapore's experience mirrors trends in other developed nations, where economic advancement, urbanization, and evolving social norms contribute to smaller family sizes. However, cultural nuances and localized policies make Singapore's case unique and provide valuable insights for policymakers worldwide.

6. Reference and Resources

6.1 References

Data

- Gallup, Oxford Wellbeing Research Centre, UN Sustainable Development Solutions Network, WHR Editorial Board. (2024, 12 March). *World Happiness Report 2024.* [Online]. Available: [https://worldhappiness.report/ed/2024/#appendices-and-data]
- Singapore Department of Statistics. (2024, 10 May). *Live-Births By Birth Order.* [Online]. Available: [https://tablebuilder.singstat.gov.sg/table/TS/M810081]
- Singapore Department of Statistics. (2024, 6 June). *Life Expectancy By Sex.* [Online]. Available: [https://tablebuilder.singstat.gov.sg/table/TS/M810501#!]
- University of Groningen. (2024, 18 November). *PWT 10.01 Penn World Table version 10.01.* [Online]. Available: [https://www.rug.nl/ggdc/productivity/pwt/?lang=en]
- World Bank Group. (2024, 16 December). *GDP - Singapore.* [Online]. Available: [https://data.worldbank.org/indicator/NY.GDP.PCAP.CD?locations=SG]

Articles

- Amelia Teng, The Straits Times. (2024, 25 November). *More help for aspiring parents as Singapore's total fertility rate hits record low.* [Online]. Available: [https://www.straitstimes.com/singapore/politics/s-pore-s-total-fertility-rate-hit-record-low-in-2022-govt-to-help-aspiring-parents-indranee-rajah]
- Department of Statistics Singapore. (2023). *Understanding Age-Specific Fertility Rate & Total Fertility Rate.* [Online]. Available: [https://www.singstat.gov.sg/-/media/files/visualising_data/infographics/population/total-fertility-rate.ashx]
- Central Provident Fund Board. (2024). *What is baby bonus?* [Online]. Available: [https://www.cpf.gov.sg/service/article/what-is-baby-bonus#:~:text=The%20Baby%20Bonus%20Scheme%20supports,enhanced%20on%201%20August%202004]
- Hale Erin, Aljazeera. (2024, 8 February). *Auspicious but unlucky: The perils of a lunar new year dragon baby boom.* [Online]. Available: [https://www.aljazeera.com/features/2024/2/8/auspicious-but-unlucky-the-perils-of-a-

lunar-new-year-dragon-baby-boom#:~:text=%E2%80%9CPeople%20in%20Taiwan%20kind%20of,by%20superstitious%20parents%20%E2%80%93%20or%20grandparents.]

- Lim Irene, Singapore Infopedia, National Library Board Singapore. (2023). *Singapore's first family planning campaign* [Online]. Available: [https://www.nlb.gov.sg/main/article-detail?cmsuuid=e6a63c59-e56f-40df-83cc-5f695ebd063a]
- Lim Tin Seng, Singapore Infopedia, National Library Board Singapore. (2023). *Two-child policy* [Online]. Available: [https://www.nlb.gov.sg/main/article-detail?cmsuuid=0613c852-aed1-4b29-81fb-faf7de447092#:~:text=Singapore%20Infopedia&text=The%20two%2Dchild%20policy%20was,no%20more%20than%20two%20children]
- National Library Board Singapore. (2015). *"Have three, or more if you can afford it" is annouced.* [Online]. Available: [https://www.nlb.gov.sg/main/article-detail?cmsuuid=1d106f7e-aca1-4c0e-ac7a-d35d0772707d]
- National Population and Talent Division. (2023, September). *Population in Brief 2023.* [Online]. Available: [https://www.population.gov.sg/files/media-centre/publications/population-in-brief-2023.pdf]

6.2 Resources used

Data importing

- Geeks for geeks. (2024, 25 July). *Enumerate() in Python.* [Online]. Available: [https://www.geeksforgeeks.org/enumerate-in-python/]
- Nishali Mudgal, Scaler Topics. (2024, 27 Mar). *Read Excel File in Python Pandas* [Online]. Available: [https://www.scaler.com/topics/pandas/pandas-read-excel/]
- Python Software Foundation. (2024, 27 December). *Introduction to Python 3.1.3. Lists.* [Online]. Available: [https://docs.python.org/3/tutorial/introduction.html#lists]
- Sabareesh Kappagantu, stack overflow. (2013, 3 Jun). *How to read a .xlsx file using the pandas Library in iPython?* [Online]. Available: [https://stackoverflow.com/questions/16888888/how-to-read-a-xlsx-file-using-the-pandas-library-in-ipython]
- W3 Schools. (2024). *Pandas DataFrame median() Method.* [Online]. Available: [https://www.w3schools.com/python/pandas/ref_df_median.asp]

Data plotting

- info river. (n.d.). *Dual Axis Charts 101: Introduction and Best practices.* [Online]. Available: [https://inforiver.com/insights/dual-axis-charts-101-introduction-best-practices/]
- Yan Holtz, Python Gra. (2024, March). *Dual Y axis customization with Matplotlib.* [Online]. Available: [https://python-graph-gallery.com/line-chart-dual-y-axis-with-matplotlib/#:~:text=This%20is%20done%20by%20calling,twinx()%20.&text=As%20can%20be%20seen%20above,goes%20from%204%20to%2020.]
- Ravikiran A S, simplilearn. (2024, 4 July). *A Complete Guide to Data Visualization in Python With Libraries & More* [Online]. Available: [https://www.simplilearn.com/tutorials/python-tutorial/data-visualization-in-python]