In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
sns.set_style('dark')
```

In [2]:
```
df = pd.read_csv("fertility_rate.csv")
df
```

Out[2]:

	Country	1960	1961	1962	1963	1964	1965	1966	1967	1968	...	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
0	_World	4.98	5.00	5.03	5.05	5.06	5.04	4.99	4.97	4.92	...	2.50	2.49	2.47	2.46	2.46	2.44	2.43	2.41	2.40	2.39
1	Afghanistan	7.45	7.45	7.45	7.45	7.45	7.45	7.45	7.45	7.45	...	5.77	5.56	5.36	5.16	4.98	4.80	4.63	4.47	4.32	4.18
2	Albania	6.49	6.40	6.28	6.13	5.96	5.77	5.58	5.39	5.22	...	1.67	1.68	1.69	1.69	1.68	1.66	1.64	1.62	1.60	1.58
3	Algeria	7.52	7.57	7.61	7.65	7.67	7.68	7.68	7.67	7.67	...	2.91	2.95	2.99	3.02	3.04	3.05	3.05	3.02	2.99	2.94
4	Angola	6.71	6.79	6.87	6.95	7.04	7.12	7.19	7.27	7.33	...	6.12	6.04	5.95	5.86	5.77	5.69	5.60	5.52	5.44	5.37
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
182	Venezuela	6.36	6.30	6.23	6.16	6.07	5.98	5.87	5.75	5.61	...	2.44	2.42	2.39	2.37	2.34	2.32	2.29	2.27	2.25	2.23
183	Vietnam	6.35	6.39	6.43	6.45	6.46	6.48	6.49	6.49	6.49	...	1.95	1.96	1.98	2.00	2.01	2.03	2.04	2.05	2.05	2.05
184	Yemen	7.94	7.96	7.99	8.03	8.07	8.11	8.17	8.22	8.28	...	4.55	4.44	4.33	4.21	4.10	3.99	3.89	3.79	3.70	3.61
185	Zambia	7.12	7.17	7.21	7.25	7.27	7.29	7.30	7.32	7.33	...	5.33	5.23	5.13	5.03	4.92	4.81	4.72	4.63	4.56	4.50
186	Zimbabwe	7.16	7.22	7.27	7.31	7.35	7.37	7.39	7.40	7.41	...	4.06	4.06	4.03	3.97	3.90	3.80	3.71	3.62	3.53	3.46

187 rows × 62 columns

In [3]:
```
df.info() # Get the basic information about the dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 187 entries, 0 to 186
Data columns (total 62 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   Country  187 non-null    object 
 1   1960     187 non-null    float64
 2   1961     187 non-null    float64
 3   1962     187 non-null    float64
 4   1963     187 non-null    float64
 5   1964     187 non-null    float64
 6   1965     187 non-null    float64
 7   1966     187 non-null    float64
 8   1967     187 non-null    float64
 9   1968     187 non-null    float64
 10  1969     187 non-null    float64
 11  1970     187 non-null    float64
 12  1971     187 non-null    float64
 13  1972     187 non-null    float64
 14  1973     187 non-null    float64
 15  1974     187 non-null    float64
 16  1975     187 non-null    float64
 17  1976     187 non-null    float64
 18  1977     187 non-null    float64
 19  1978     187 non-null    float64
 20  1979     187 non-null    float64
 21  1980     187 non-null    float64
 22  1981     187 non-null    float64
 23  1982     187 non-null    float64
 24  1983     187 non-null    float64
 25  1984     187 non-null    float64
 26  1985     187 non-null    float64
 27  1986     187 non-null    float64
 28  1987     187 non-null    float64
 29  1988     187 non-null    float64
 30  1989     187 non-null    float64
 31  1990     187 non-null    float64
 32  1991     187 non-null    float64
 33  1992     187 non-null    float64
 34  1993     187 non-null    float64
 35  1994     187 non-null    float64
 36  1995     187 non-null    float64
 37  1996     187 non-null    float64
 38  1997     187 non-null    float64
```

```
 39  1998     187 non-null    float64
 40  1999     187 non-null    float64
 41  2000     187 non-null    float64
 42  2001     187 non-null    float64
 43  2002     187 non-null    float64
 44  2003     187 non-null    float64
 45  2004     187 non-null    float64
 46  2005     187 non-null    float64
 47  2006     187 non-null    float64
 48  2007     187 non-null    float64
 49  2008     187 non-null    float64
 50  2009     187 non-null    float64
 51  2010     187 non-null    float64
 52  2011     187 non-null    float64
 53  2012     187 non-null    float64
 54  2013     187 non-null    float64
 55  2014     187 non-null    float64
 56  2015     187 non-null    float64
 57  2016     187 non-null    float64
 58  2017     187 non-null    float64
 59  2018     187 non-null    float64
 60  2019     187 non-null    float64
 61  2020     187 non-null    float64
dtypes: float64(61), object(1)
memory usage: 90.7+ KB
```

In [4]:

```
df.describe() # Get the summary statistics of the dataset
```

Out[4]:

	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	...	2011	20
count	187.000000	187.000000	187.000000	187.000000	187.000000	187.000000	187.000000	187.000000	187.000000	187.000000	...	187.000000	187.0000
mean	5.533529	5.526578	5.514385	5.499251	5.472888	5.426043	5.375561	5.334545	5.281925	5.229358	...	2.908235	2.8831
std	1.735528	1.747268	1.762098	1.771850	1.784293	1.815374	1.846265	1.852808	1.875046	1.893093	...	1.443078	1.4107
min	1.940000	1.940000	1.790000	1.820000	1.790000	1.740000	1.580000	1.800000	1.830000	1.870000	...	1.200000	1.2700
25%	4.110000	3.930000	4.045000	4.000000	3.970000	3.825000	3.665000	3.555000	3.405000	3.260000	...	1.790000	1.8000
50%	6.250000	6.270000	6.220000	6.170000	6.130000	6.100000	6.080000	6.010000	5.940000	5.850000	...	2.440000	2.4200
75%	6.825000	6.830000	6.835000	6.835000	6.845000	6.805000	6.805000	6.775000	6.750000	6.730000	...	3.905000	3.8450
max	8.190000	8.190000	8.200000	8.200000	8.200000	8.200000	8.200000	8.220000	8.280000	8.330000	...	7.430000	7.3800

8 rows × 61 columns

In [5]:
```
df.isna().sum()# Check the number of missing values in each column
```

Out[5]:
```
Country    0
1960       0
1961       0
1962       0
1963       0
          ..
2016       0
2017       0
2018       0
2019       0
2020       0
Length: 62, dtype: int64
```

In [6]:
```
#creating a copy of the dataframe
df_copy = df
```

In [7]:
```
# Trasposing the dataframe
df=df.T
df
```

Out[7]:

	0	1	2	3	4	5	6	7	8	9	...	177	178	179	180	181
Country	_World	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Arab World	Argentina	Armenia	Aruba	...	United Kingdom	United States	Uruguay	Uzbekistan	Vanuatu
1960	4.98	7.45	6.49	7.52	6.71	4.43	6.98	3.11	4.79	4.82	...	2.69	3.65	2.88	6.26	7.2
1961	5.0	7.45	6.4	7.57	6.79	4.39	7.0	3.1	4.67	4.66	...	2.78	3.62	2.89	6.36	7.12
1962	5.03	7.45	6.28	7.61	6.87	4.34	7.02	3.09	4.52	4.47	...	2.86	3.46	2.88	6.44	7.03
1963	5.05	7.45	6.13	7.65	6.95	4.3	7.04	3.08	4.35	4.27	...	2.88	3.32	2.87	6.49	6.94
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2016	2.44	4.8	1.66	3.05	5.69	2.0	3.32	2.29	1.74	1.87	...	1.79	1.82	1.99	2.46	3.86
2017	2.43	4.63	1.64	3.05	5.6	2.0	3.28	2.28	1.75	1.89	...	1.74	1.77	1.98	2.42	3.82
2018	2.41	4.47	1.62	3.02	5.52	1.99	3.23	2.26	1.76	1.9	...	1.68	1.73	1.97	2.6	3.78
2019	2.4	4.32	1.6	2.99	5.44	1.99	3.19	2.25	1.76	1.9	...	1.63	1.71	1.96	2.79	3.74
2020	2.39	4.18	1.58	2.94	5.37	1.98	3.15	2.23	1.76	1.9	...	1.56	1.64	1.95	2.9	3.71

62 rows × 187 columns

In [8]:

```
# rename the columns
df.rename(columns=df.iloc[0],inplace = True)
```

In [9]:

```
df
```

Out[9]:

	_World	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Arab World	Argentina	Armenia	Aruba	...	United Kingdom	United States	Uruguay	Uzbekistan	Va
Country	_World	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Arab World	Argentina	Armenia	Aruba	...	United Kingdom	United States	Uruguay	Uzbekistan	V
1960	4.98	7.45	6.49	7.52	6.71	4.43	6.98	3.11	4.79	4.82	...	2.69	3.65	2.88	6.26	
1961	5.0	7.45	6.4	7.57	6.79	4.39	7.0	3.1	4.67	4.66	...	2.78	3.62	2.89	6.36	
1962	5.03	7.45	6.28	7.61	6.87	4.34	7.02	3.09	4.52	4.47	...	2.86	3.46	2.88	6.44	
1963	5.05	7.45	6.13	7.65	6.95	4.3	7.04	3.08	4.35	4.27	...	2.88	3.32	2.87	6.49	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2016	2.44	4.8	1.66	3.05	5.69	2.0	3.32	2.29	1.74	1.87	...	1.79	1.82	1.99	2.46	
2017	2.43	4.63	1.64	3.05	5.6	2.0	3.28	2.28	1.75	1.89	...	1.74	1.77	1.98	2.42	
2018	2.41	4.47	1.62	3.02	5.52	1.99	3.23	2.26	1.76	1.9	...	1.68	1.73	1.97	2.6	
2019	2.4	4.32	1.6	2.99	5.44	1.99	3.19	2.25	1.76	1.9	...	1.63	1.71	1.96	2.79	
2020	2.39	4.18	1.58	2.94	5.37	1.98	3.15	2.23	1.76	1.9	...	1.56	1.64	1.95	2.9	

62 rows × 187 columns

In [10]:

```
df=df.drop(df.index[0])
df=df.reset_index()
df.head()
```

Out[10]:

	index	_World	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Arab World	Argentina	Armenia	...	United Kingdom	United States	Uruguay	Uzbekistan	Vanuatu
0	1960	4.98	7.45	6.49	7.52	6.71	4.43	6.98	3.11	4.79	...	2.69	3.65	2.88	6.26	7.2
1	1961	5.0	7.45	6.4	7.57	6.79	4.39	7.0	3.1	4.67	...	2.78	3.62	2.89	6.36	7.12
2	1962	5.03	7.45	6.28	7.61	6.87	4.34	7.02	3.09	4.52	...	2.86	3.46	2.88	6.44	7.03
3	1963	5.05	7.45	6.13	7.65	6.95	4.3	7.04	3.08	4.35	...	2.88	3.32	2.87	6.49	6.94
4	1964	5.06	7.45	5.96	7.67	7.04	4.25	7.05	3.07	4.15	...	2.93	3.19	2.86	6.52	6.84

5 rows × 188 columns

In [11]:

```
df.rename(columns={'index':'Year'}, inplace=True)
df
```

Out[11]:

	Year	_World	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Arab World	Argentina	Armenia	...	United Kingdom	United States	Uruguay	Uzbekistan	Vanuatu
0	1960	4.98	7.45	6.49	7.52	6.71	4.43	6.98	3.11	4.79	...	2.69	3.65	2.88	6.26	7.2
1	1961	5.0	7.45	6.4	7.57	6.79	4.39	7.0	3.1	4.67	...	2.78	3.62	2.89	6.36	7.12
2	1962	5.03	7.45	6.28	7.61	6.87	4.34	7.02	3.09	4.52	...	2.86	3.46	2.88	6.44	7.03
3	1963	5.05	7.45	6.13	7.65	6.95	4.3	7.04	3.08	4.35	...	2.88	3.32	2.87	6.49	6.94
4	1964	5.06	7.45	5.96	7.67	7.04	4.25	7.05	3.07	4.15	...	2.93	3.19	2.86	6.52	6.84
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
56	2016	2.44	4.8	1.66	3.05	5.69	2.0	3.32	2.29	1.74	...	1.79	1.82	1.99	2.46	3.86
57	2017	2.43	4.63	1.64	3.05	5.6	2.0	3.28	2.28	1.75	...	1.74	1.77	1.98	2.42	3.82
58	2018	2.41	4.47	1.62	3.02	5.52	1.99	3.23	2.26	1.76	...	1.68	1.73	1.97	2.6	3.78
59	2019	2.4	4.32	1.6	2.99	5.44	1.99	3.19	2.25	1.76	...	1.63	1.71	1.96	2.79	3.74
60	2020	2.39	4.18	1.58	2.94	5.37	1.98	3.15	2.23	1.76	...	1.56	1.64	1.95	2.9	3.71

61 rows × 188 columns

In [12]:
```
df_copy2 = df
```

Top 15 Countries with the highest birth rates

In [13]:
```
# get the copy
df = df_copy
# Select the most recent year (2020) and birth rate columns
birth_rates_2020 = df.loc[:, ('Country', '2020')]

# Convert the birth rate column to numeric
birth_rates_2020['2020'] = pd.to_numeric(birth_rates_2020['2020'], errors='coerce')

# Sort the countries based on birth rate percentages in descending order
sorted_birth_rates = birth_rates_2020.sort_values('2020', ascending=False)
```

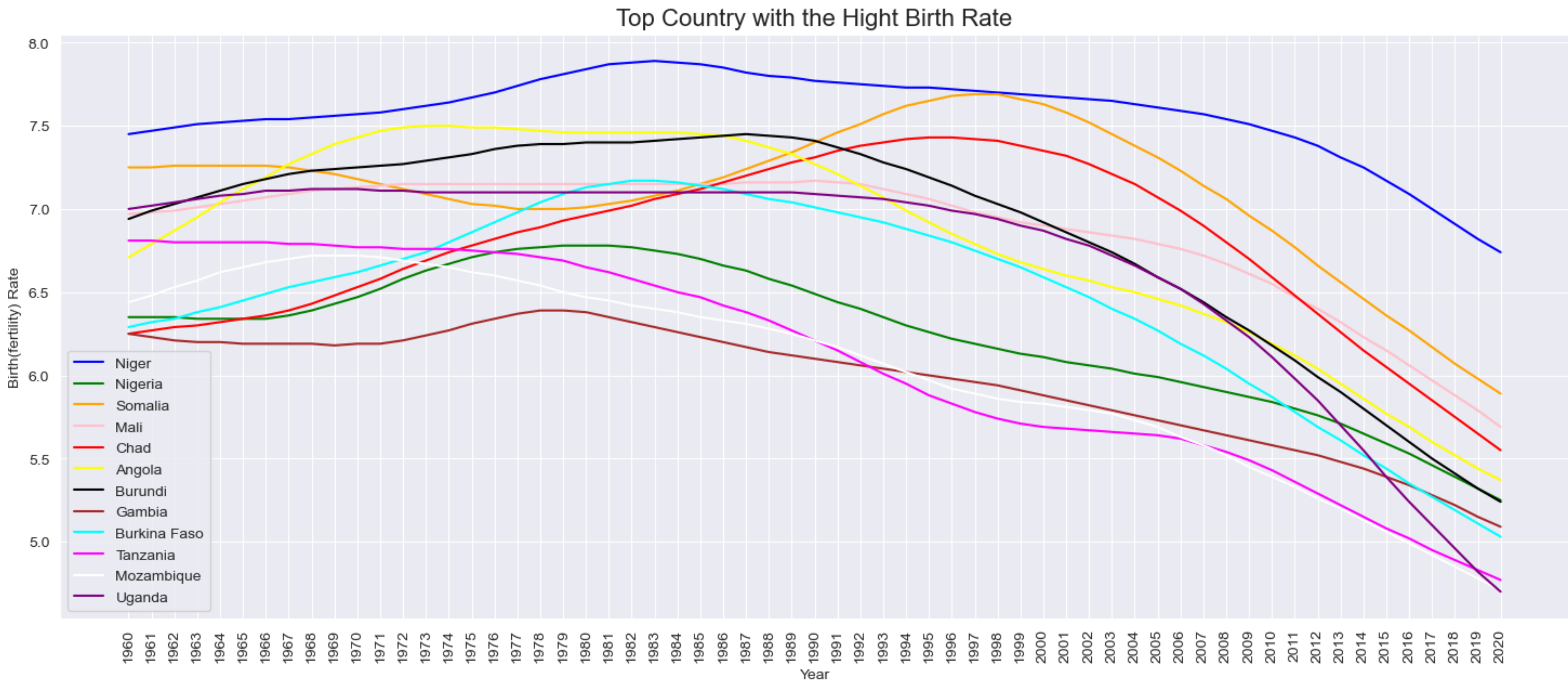
```
# Select the top 15 countries with the highest birth rates
top_15_countries = sorted_birth_rates.head(15)

# Display the top 15 countries and their birth rate percentages
print("Top 15 Countries with the highest birth rates")
print(top_15_countries[['Country', '2020']])
```

```
Top 15 Countries with the highest birth rates
                      Country  2020
124                     Niger  6.74
151                   Somalia  5.89
106                      Mali  5.69
35                       Chad  5.55
4                      Angola  5.37
125                   Nigeria  5.25
29                    Burundi  5.24
64                Gambia, The  5.09
28               Burkina Faso  5.03
165                  Tanzania  4.77
116                Mozambique  4.71
174                    Uganda  4.70
20                      Benin  4.70
34   Central African Republic  4.57
72                     Guinea  4.55
```

In [14]:

```
df = df_copy2
plt.figure(figsize=(18,7))
plt.plot(df.Year,df['Niger'], color = 'blue', label = 'Niger')
plt.plot(df.Year,df['Nigeria'], color = 'green', label = 'Nigeria')
plt.plot(df.Year,df['Somalia'], color = 'orange', label = 'Somalia')
plt.plot(df.Year,df['Mali'], color = 'pink', label = 'Mali')
plt.plot(df.Year,df['Chad'], color = 'red', label = 'Chad')
plt.plot(df.Year,df['Angola'], color = 'yellow', label = 'Angola')
plt.plot(df.Year,df['Burundi'], color = 'black', label = 'Burundi')
plt.plot(df.Year,df['Gambia, The'], color = 'brown', label = 'Gambia')
plt.plot(df.Year,df['Burkina Faso'], color = 'cyan', label = 'Burkina Faso')
plt.plot(df.Year,df['Tanzania'], color = 'magenta', label = 'Tanzania')
plt.plot(df.Year,df['Mozambique'], color = 'white', label = 'Mozambique')
plt.plot(df.Year,df['Uganda'], color = 'purple', label = 'Uganda')
plt.xlabel('Year')
plt.ylabel('Birth(fertility) Rate')
plt.title('Top Country with the Hight Birth Rate', fontsize=16)
plt.xticks(rotation=90)
```

```
plt.legend(loc = 'lower left')
plt.grid(True)
plt.show()
```



Top 15 Countries with the Lowest birth rates

In [15]:
```
df = df_copy
# Select the most recent year (2020) and birth rate columns
birth_rates_2020 = df.loc[:, ('Country', '2020')]

# Convert the birth rate column to numeric
birth_rates_2020['2020'] = pd.to_numeric(birth_rates_2020['2020'], errors='coerce')

# Sort the countries based on birth rate percentages in descending order
sorted_birth_rates = birth_rates_2020.sort_values('2020', ascending=True)
```

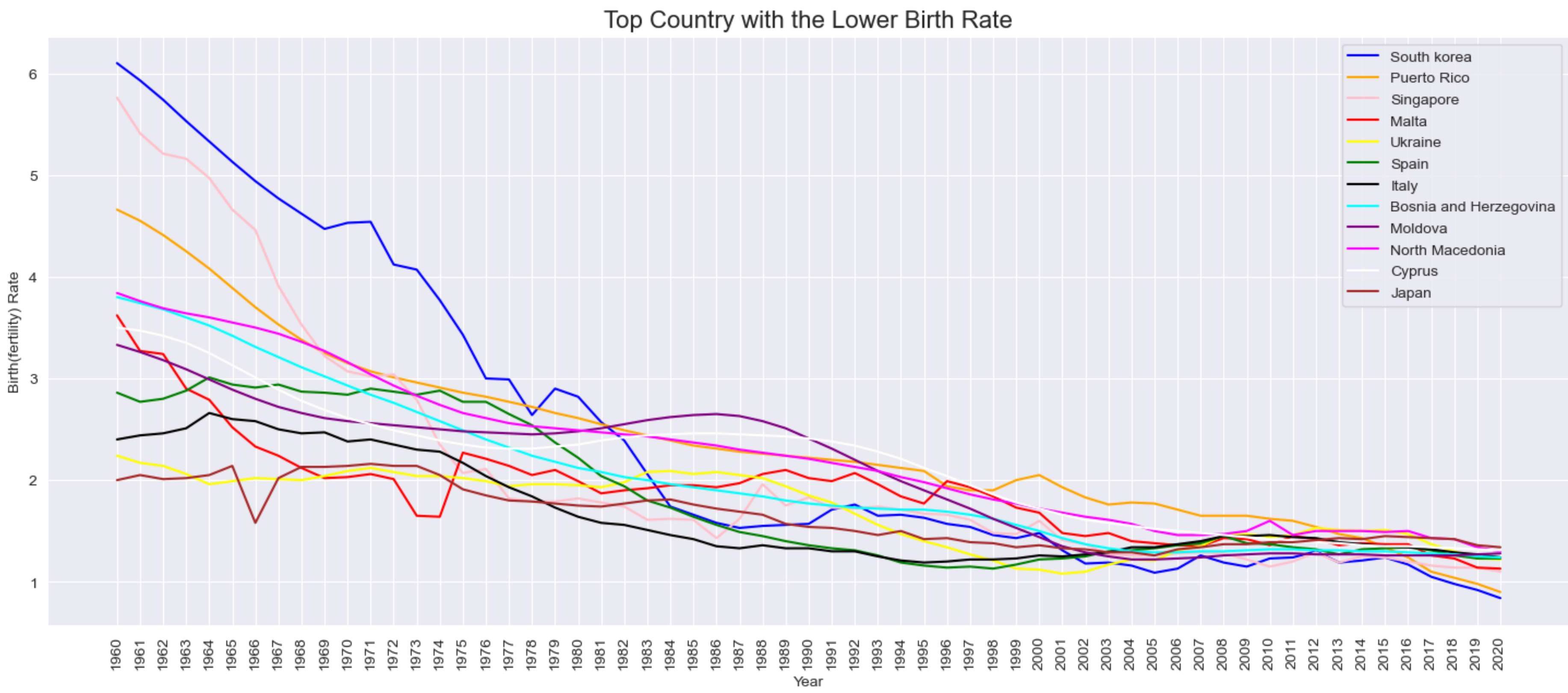
```
# Select the top 15 countries with the lowest birth rates
top_15_countries = sorted_birth_rates.head(15)

# Display the top 30 countries and their birth rate percentages
print("Top 15 Countries with the Lowest birth rates")
print(top_15_countries[['Country', '2020']])
```

```
Top 15 Countries with the Lowest birth rates
                    Country  2020
153             South Korea  0.84
137             Puerto Rico  0.90
147               Singapore  1.10
107                   Malta  1.13
175                 Ukraine  1.22
154                   Spain  1.23
86                    Italy  1.24
23   Bosnia and Herzegovina  1.24
112                 Moldova  1.28
126         North Macedonia  1.30
45                   Cyprus  1.31
88                    Japan  1.34
68                   Greece  1.34
176    United Arab Emirates  1.37
60                  Finland  1.37
```

In [16]:

```
df = df_copy2
plt.figure(figsize=(18,7))
plt.plot(df.Year,df['South Korea'], color = 'blue', label = 'South korea')
plt.plot(df.Year,df['Puerto Rico'], color = 'orange', label = 'Puerto Rico')
plt.plot(df.Year,df['Singapore'], color = 'pink', label = 'Singapore')
plt.plot(df.Year,df['Malta'], color = 'red', label = 'Malta')
plt.plot(df.Year,df['Ukraine'], color = 'yellow', label = 'Ukraine')
plt.plot(df.Year,df['Spain'], color = 'green', label = 'Spain')
plt.plot(df.Year,df['Italy'], color = 'black', label = 'Italy')
plt.plot(df.Year,df['Bosnia and Herzegovina'], color = 'cyan', label = 'Bosnia and Herzegovina' )
plt.plot(df.Year,df['Moldova'], color = 'purple', label = 'Moldova' )
plt.plot(df.Year,df['North Macedonia'], color = 'magenta', label = 'North Macedonia' )
plt.plot(df.Year,df['Cyprus'], color = 'white', label = 'Cyprus' )
plt.plot(df.Year,df['Japan'], color = 'brown', label = 'Japan')
plt.xlabel('Year')
plt.ylabel('Birth(fertility) Rate')
plt.title('Top Country with the Lower Birth Rate', fontsize=16)
plt.xticks(rotation=90)
plt.legend(loc = 'upper right')
```

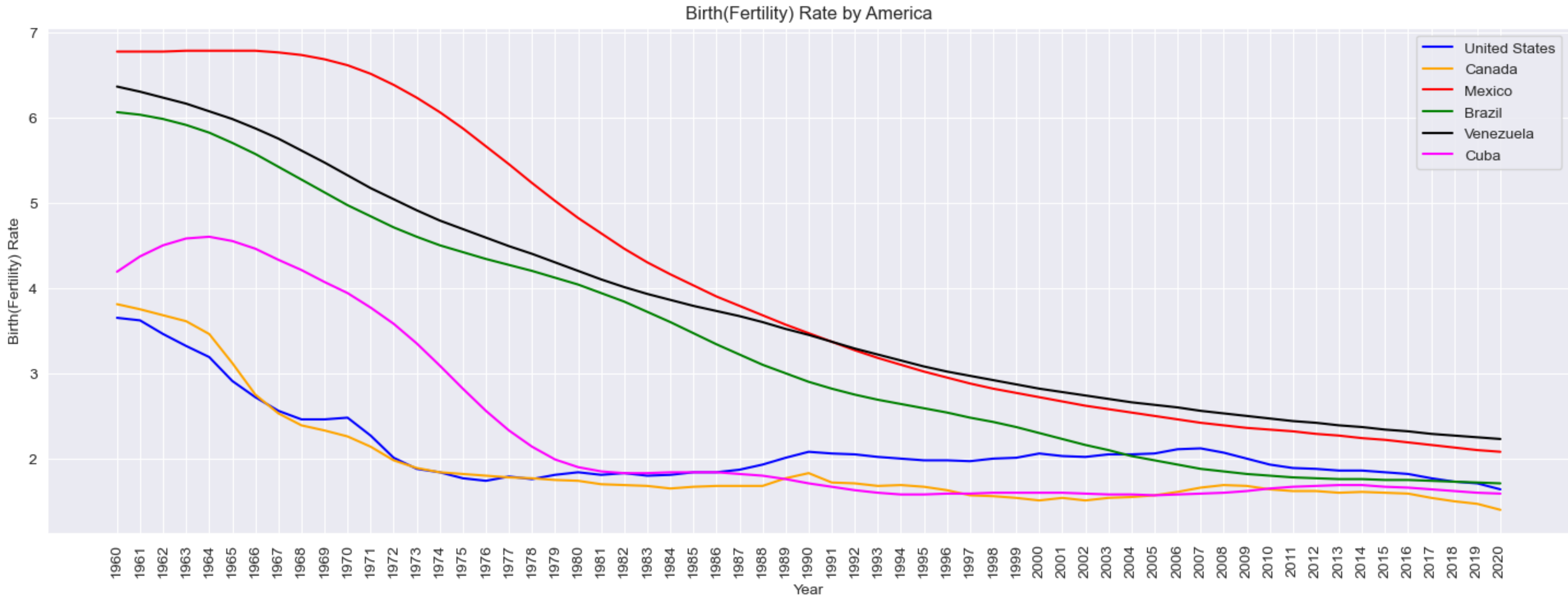
```
plt.grid(True)
plt.show()
```



In [ ]:

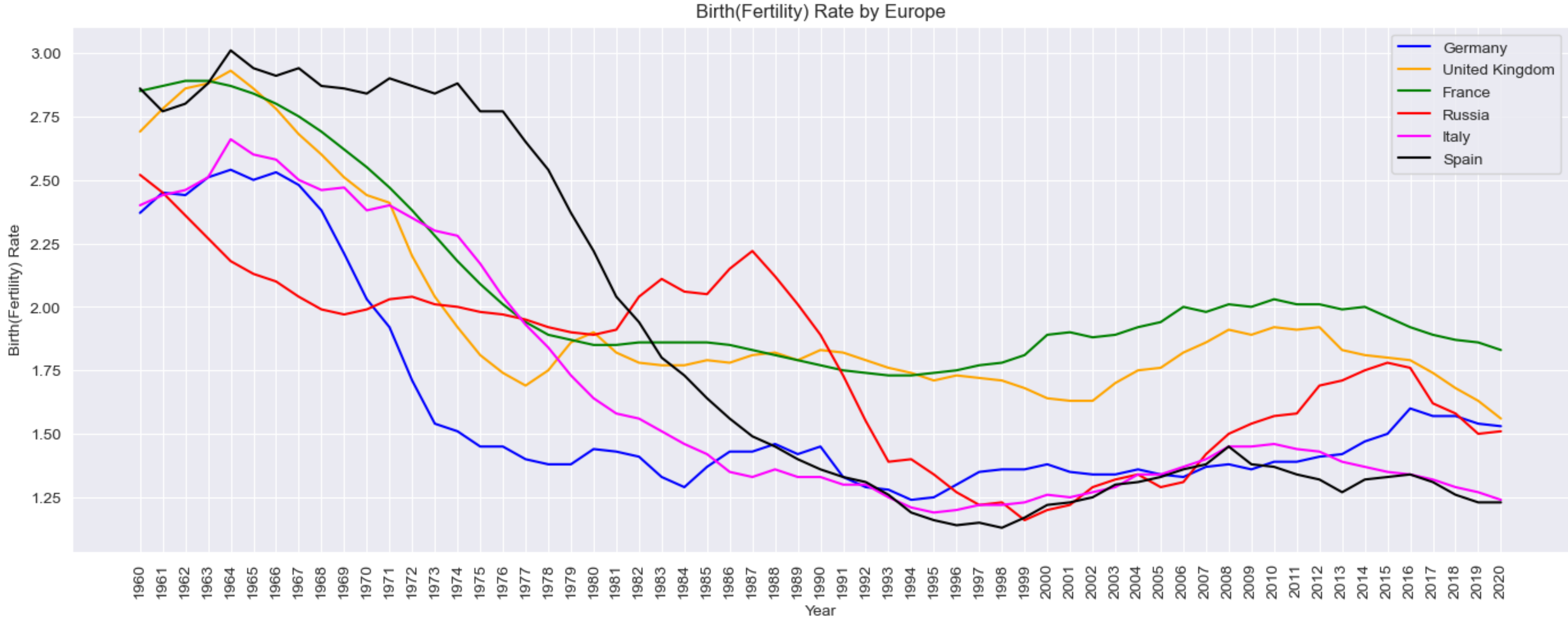
In [17]:
```
plt.figure(figsize=(18,6))
plt.plot(df.Year,df['United States'], color = 'blue', label = 'United States')
plt.plot(df.Year,df['Canada'], color = 'orange', label = 'Canada')
plt.plot(df.Year,df['Mexico'], color = 'red', label = 'Mexico')
plt.plot(df.Year,df['Brazil'], color = 'green', label = 'Brazil')
plt.plot(df.Year,df['Venezuela'], color = 'black', label = 'Venezuela')
plt.plot(df.Year,df['Cuba'], color = 'magenta', label = 'Cuba')
plt.xlabel('Year')
plt.ylabel('Birth(Fertility) Rate')
plt.title('Birth(Fertility) Rate by America')
plt.xticks(rotation=90)
plt.legend(loc = 'upper right')
```

```
plt.grid(True)
plt.show()
```



In [ ]:

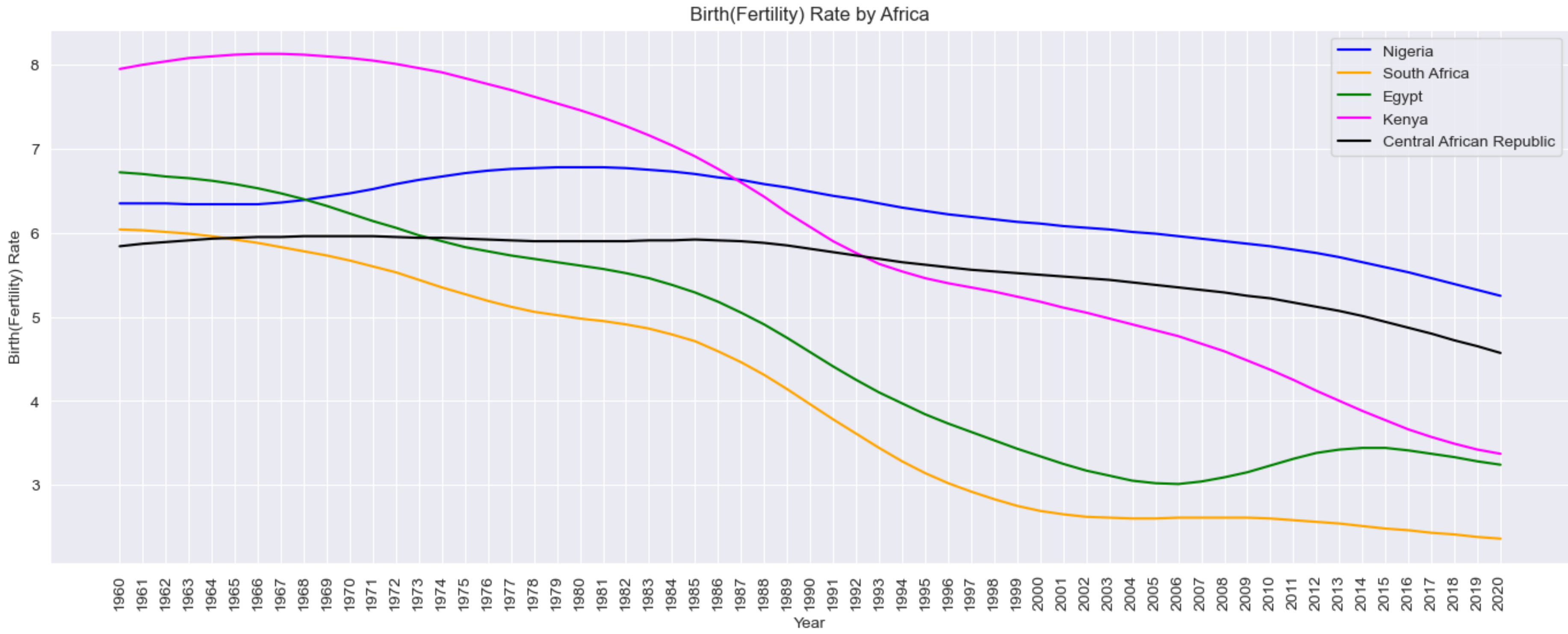
In [18]:
```
plt.figure(figsize=(17,6))
plt.plot(df.Year,df['Germany'], color = 'blue', label = 'Germany')
plt.plot(df.Year,df['United Kingdom'], color = 'orange', label = 'United Kingdom')
plt.plot(df.Year,df['France'], color = 'green', label = 'France')
plt.plot(df.Year,df['Russia'], color = 'red', label = 'Russia')
plt.plot(df.Year,df['Italy'], color = 'magenta', label = 'Italy')
plt.plot(df.Year,df['Spain'], color = 'black', label = 'Spain')
plt.xlabel('Year')
plt.ylabel('Birth(Fertility) Rate')
plt.title('Birth(Fertility) Rate by Europe')
plt.legend(loc = 'upper right')
plt.xticks(rotation=90)
plt.grid(True)
plt.show()
```



In [ ]:

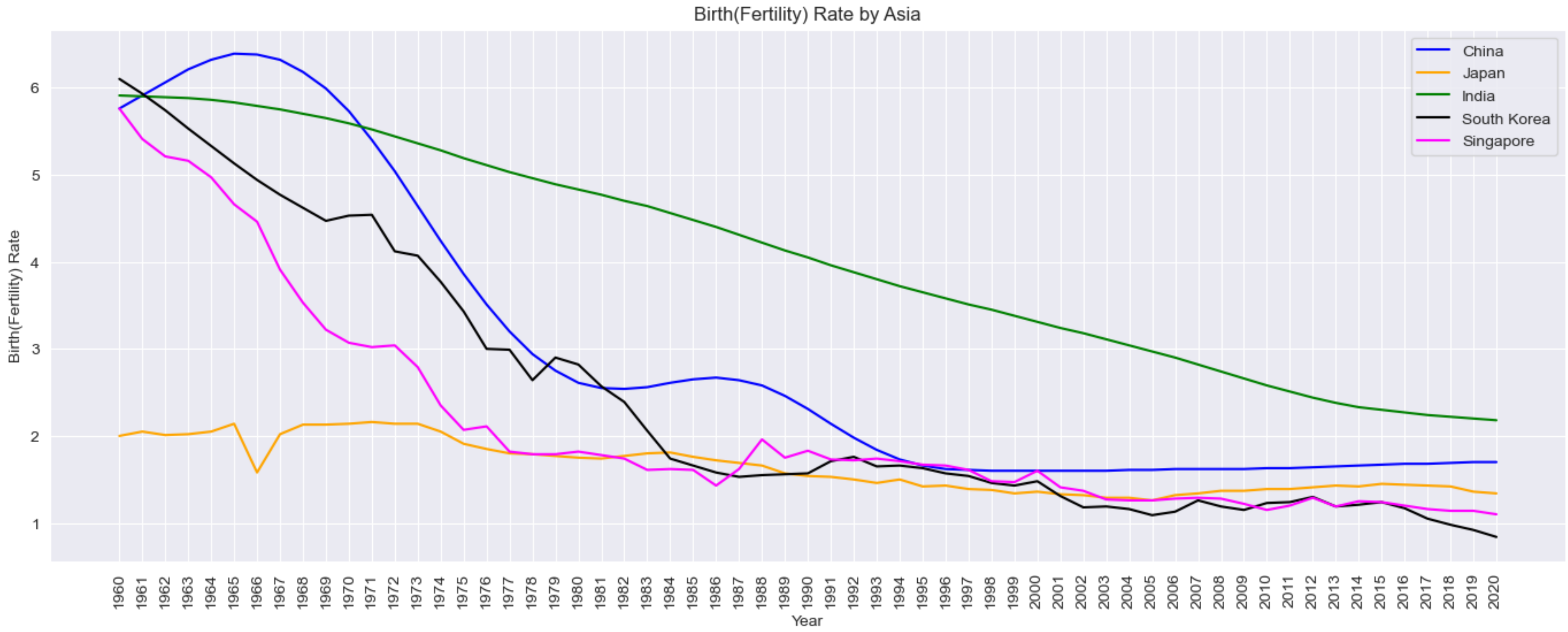
In [19]:

```
plt.figure(figsize=(17,6))
plt.plot(df.Year,df['Nigeria'], color = 'blue', label = 'Nigeria')
plt.plot(df.Year,df['South Africa'], color = 'orange', label = 'South Africa')
plt.plot(df.Year,df['Egypt'], color = 'green', label = 'Egypt')
plt.plot(df.Year,df['Kenya'], color = 'magenta', label = 'Kenya')
plt.plot(df.Year,df['Central African Republic'], color = 'black', label = 'Central African Republic')
plt.xlabel('Year')
plt.ylabel('Birth(Fertility) Rate')
plt.title('Birth(Fertility) Rate by Africa')
plt.legend(loc = 'upper right')
plt.xticks(rotation=90)
plt.grid(True)
plt.show()
```



In [ ]:

In [20]:
```
plt.figure(figsize=(17,6))
plt.plot(df.Year,df['China'], color = 'blue', label = 'China')
plt.plot(df.Year,df['Japan'], color = 'orange', label = 'Japan')
plt.plot(df.Year,df['India'], color = 'green', label = 'India')
plt.plot(df.Year,df['South Korea'], color = 'black', label = 'South Korea')
plt.plot(df.Year,df['Singapore'], color = 'magenta', label = 'Singapore')
plt.xlabel('Year')
plt.ylabel('Birth(Fertility) Rate')
plt.title('Birth(Fertility) Rate by Asia')
plt.legend(loc = 'upper right')
plt.xticks(rotation=90)
plt.grid(True)
plt.show()
```



In [ ]:

In [21]:
```
df['Asia'] = df[['Armenia', 'Fiji', 'Bahrain', 'Cyprus','Bhutan','Brunei Darussalam',
       'Afghanistan', 'Bangladesh', 'India', 'Nepal','Iran','Iraq','Azerbaijan',
       'Pakistan', 'Sri Lanka', 'China', 'Japan','Turkiye','Saudi Arabia','Syrian Arab Republic',
       'Kuwait', 'South Korea', 'Mongolia','Uzbekistan','Yemen',
       'Brunei Darussalam', 'Cambodia', 'Indonesia', 'Laos','Syrian Arab Republic','Jordan','Lebanon',
       'Malaysia', 'Myanmar','Maldives', 'Philippines', 'Singapore', 'Thailand','Georgia',
       'Timor-Leste', 'Vietnam', 'Kazakhstan','United Arab Emirates','Israel','Oman',
       'Kyrgyz Republic', 'Tajikistan', 'Turkmenistan', 'Uzbekistan','Qatar']].mean(axis=1)
```

In [22]:
```
df['Africa']=df[['Algeria',
        'Angola', 'Benin', 'Botswana',
       'Burkina Faso', 'Burundi', 'Cameroon','Cabo Verde',
       'Central African Republic', 'Chad', 'Congo', 'Comoros',
       "Cote d'Ivoire", 'Djibouti', 'Equatorial Guinea', 'Eritrea',
       'Ethiopia','Egypt','Eswatini', 'Gabon', 'Gambia, The', 'Ghana', 'Guinea',
       'Guinea-Bissau', 'Kenya', 'Lesotho', 'Liberia','Libya','Madagascar',
```

```
       'Malawi', 'Mali', 'Mauritania', 'Mauritius', 'Mozambique','Morocco',
       'Namibia', 'Niger', 'Nigeria', 'Rwanda','Sao Tome and Principe','Senegal',
       'Sierra Leone', 'Somalia', 'South Africa', 'Sudan', 'Sudan',
       'Tanzania', 'Togo','Tunisia', 'Uganda', 'Zambia', 'Zimbabwe',
       ]].mean(axis=1)
```

In [23]:
```
df['North America'] = df[['Aruba','Antigua and Barbuda','Belize','Bahamas','Bahamas',
       'Costa Rica', 'Cuba', 'Dominican Republic', 'El Salvador',
       'Guatemala', 'Haiti', 'Honduras', 'Jamaica', 'Mexico', 'Nicaragua',
       'Panama','Puerto Rico','St. Lucia','St. Vincent and the Grenadines', 'Trinidad and Tobago',
       'United States', 'Grenada','Canada',]].mean(axis=1)
```

In [24]:
```
df['South America'] = df[['Belize', 'Argentina',
       'Bolivia', 'Brazil', 'Chile', 'Colombia', 'Ecuador', 'Guyana',
       'Paraguay', 'Peru', 'Uruguay', 'Venezuela']].mean(axis=1)
```

In [25]:
```
df['Europe'] = df[[ 'Albania','Austria', 'Belgium', 'Bosnia and Herzegovina',
       'Bulgaria', 'Croatia', 'Czechia','Czechia','Slovenia', 'Estonia',
        'Hungary', 'Latvia',
       'Lithuania', 'North Macedonia', 'Montenegro', 'Poland', 'Romania',
       'Slovak Republic', 'Ukraine',
       'Armenia', 'Azerbaijan', 'Belarus', 'Georgia', 'Moldova', 'Russia',
        'Denmark', 'Finland', 'France', 'Germany', 'Greece',
       'Iceland', 'Ireland', 'Italy', 'Malta',
       'Netherlands', 'Norway', 'Portugal', 'Spain', 'Sweden',
       'Switzerland', 'United Kingdom']].mean(axis=1)
```

In [26]:
```
df['Australia/Oceania'] = df[['Australia','Papua New Guinea','New Zealand','Fiji',
                             'Solomon Islands','Vanuatu','New Caledonia','French Polynesia',
                             'Samoa','Guam','Kiribati','Tonga',]].mean(axis=1)
```

In [27]:
```
df = df.astype(float)
df
```

Out[27]:

	Year	_World	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Arab World	Argentina	Armenia	...	Vietnam	Yemen	Zambia	Zimbabwe	Asia	
0	1960.0	4.98	7.45	6.49	7.52	6.71	4.43	6.98	3.11	4.79	...	6.35	7.94	7.12	7.16	6.2172	6.
1	1961.0	5.00	7.45	6.40	7.57	6.79	4.39	7.00	3.10	4.67	...	6.39	7.96	7.17	7.22	6.2170	6.
2	1962.0	5.03	7.45	6.28	7.61	6.87	4.34	7.02	3.09	4.52	...	6.43	7.99	7.21	7.27	6.2112	6.
3	1963.0	5.05	7.45	6.13	7.65	6.95	4.30	7.04	3.08	4.35	...	6.45	8.03	7.25	7.31	6.1994	6.
4	1964.0	5.06	7.45	5.96	7.67	7.04	4.25	7.05	3.07	4.15	...	6.46	8.07	7.27	7.35	6.1734	6.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
56	2016.0	2.44	4.80	1.66	3.05	5.69	2.00	3.32	2.29	1.74	...	2.03	3.99	4.81	3.80	2.4312	4.
57	2017.0	2.43	4.63	1.64	3.05	5.60	2.00	3.28	2.28	1.75	...	2.04	3.89	4.72	3.71	2.3938	4.
58	2018.0	2.41	4.47	1.62	3.02	5.52	1.99	3.23	2.26	1.76	...	2.05	3.79	4.63	3.62	2.3802	4.
59	2019.0	2.40	4.32	1.60	2.99	5.44	1.99	3.19	2.25	1.76	...	2.05	3.70	4.56	3.53	2.3606	4.
60	2020.0	2.39	4.18	1.58	2.94	5.37	1.98	3.15	2.23	1.76	...	2.05	3.61	4.50	3.46	2.3304	4.

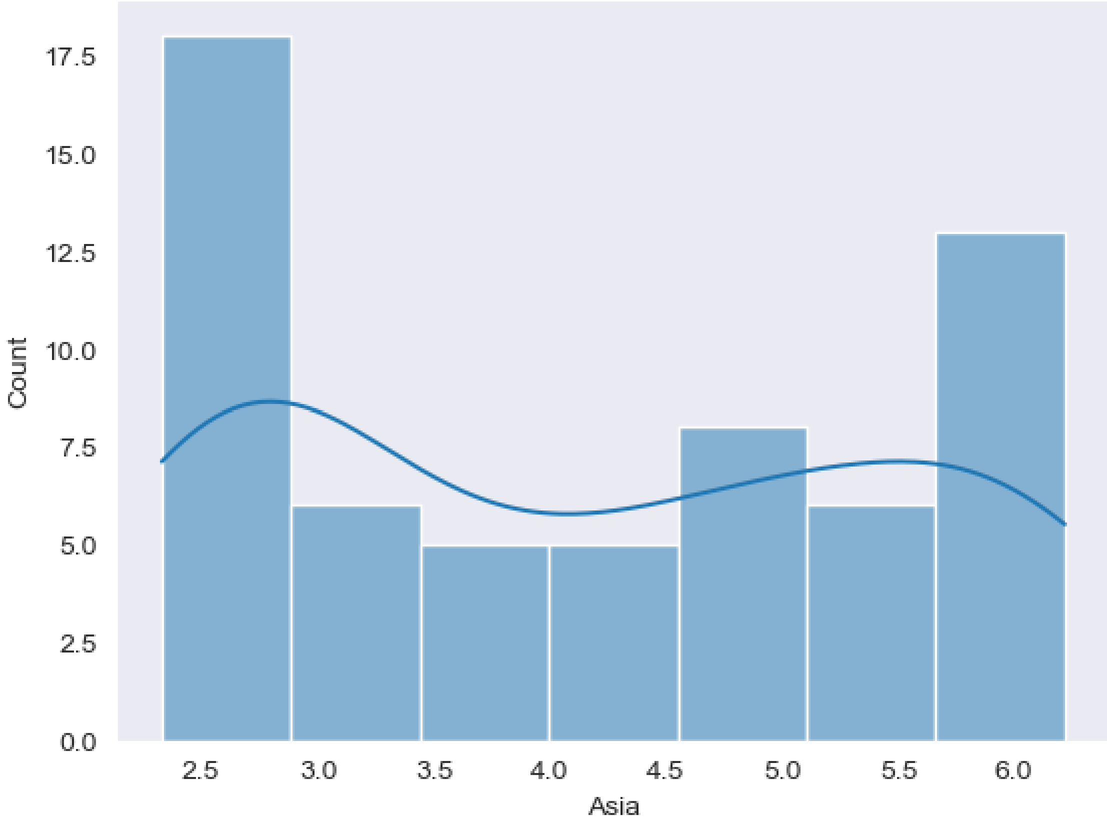
61 rows × 194 columns

In [28]:

```
sns.histplot(df.Asia, kde = True)
```

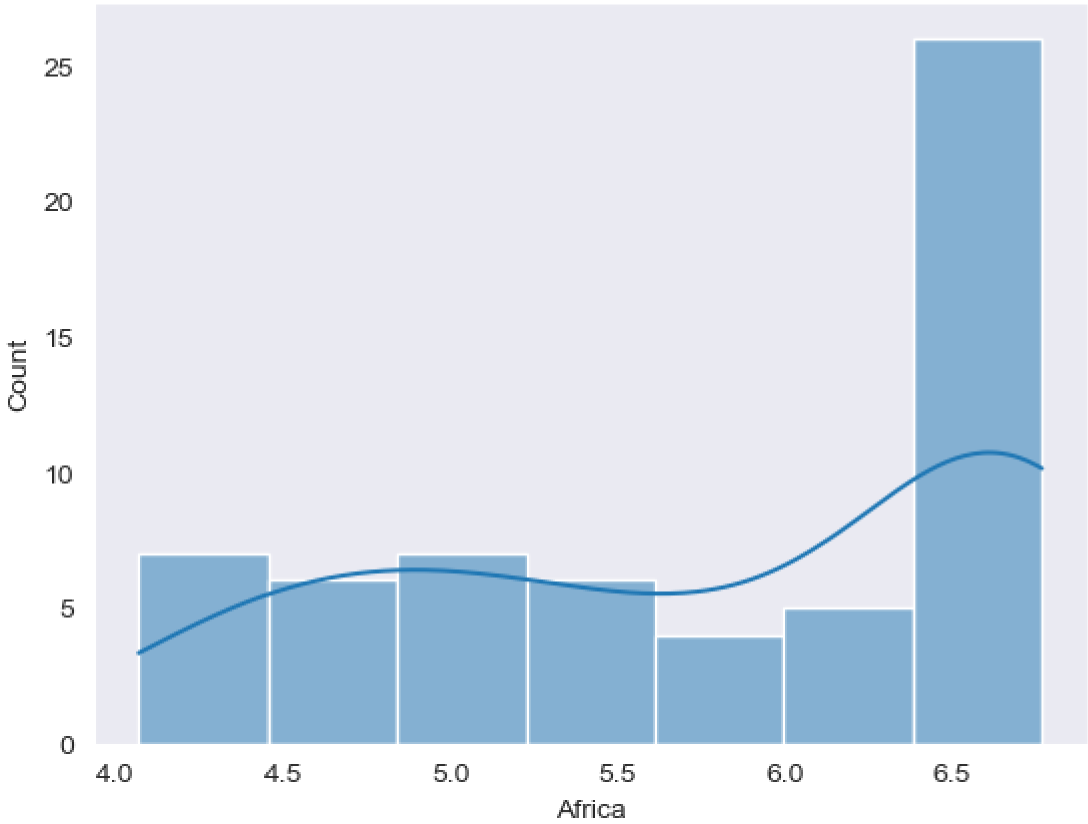
Out[28]:

```
<AxesSubplot:xlabel='Asia', ylabel='Count'>
```



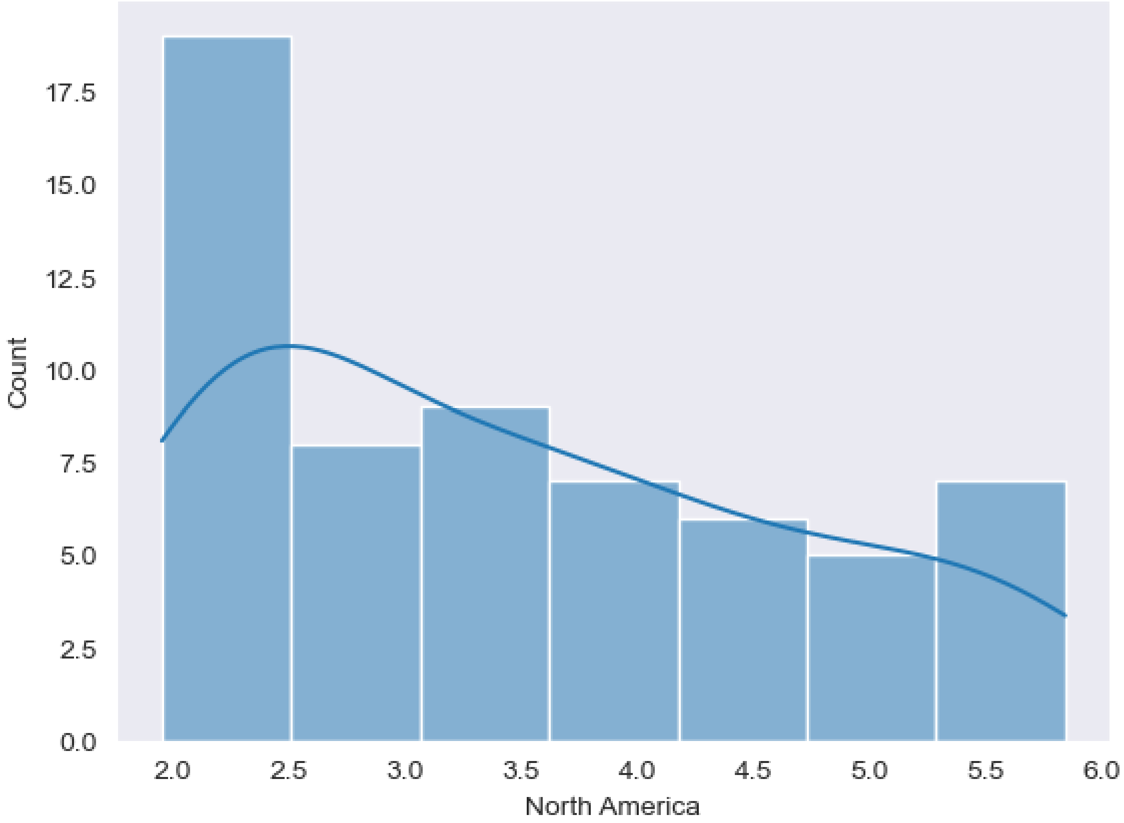
In [29]:
```
sns.histplot(df.Africa,kde = True)
```

Out[29]: <AxesSubplot:xlabel='Africa', ylabel='Count'>



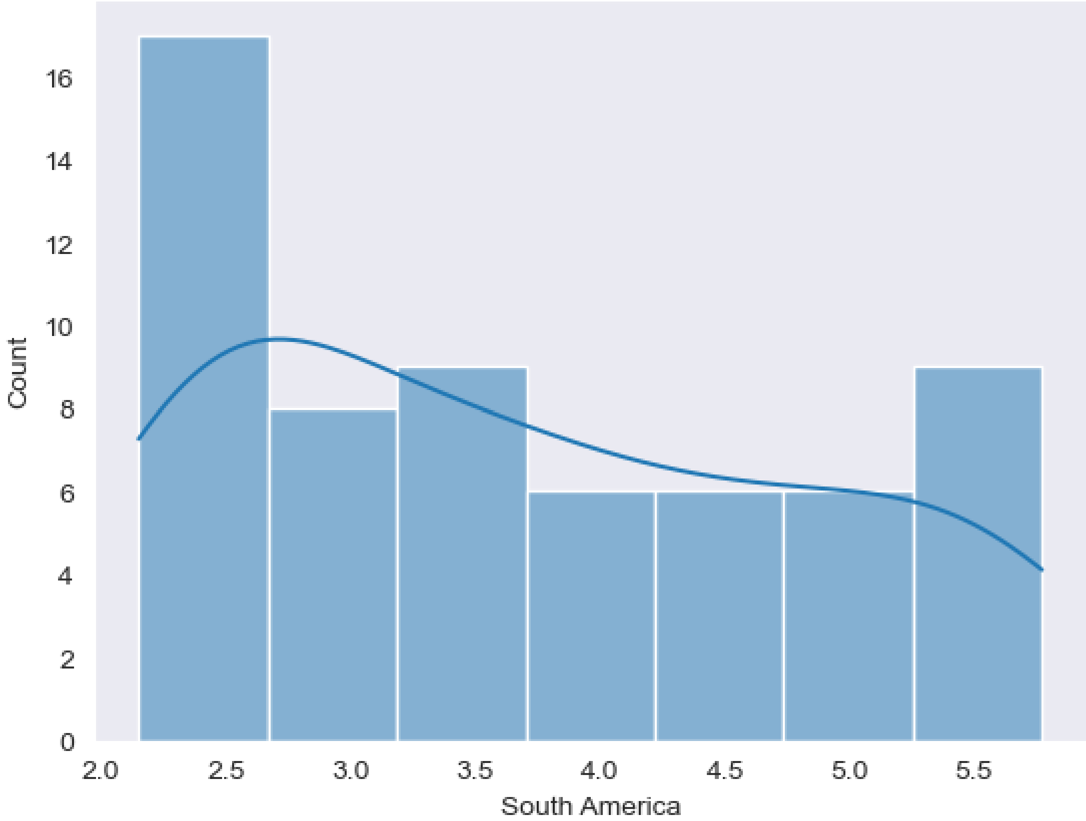
In [30]:
```
sns.histplot(df['North America'],kde = True)
```

Out[30]:
```
<AxesSubplot:xlabel='North America', ylabel='Count'>
```



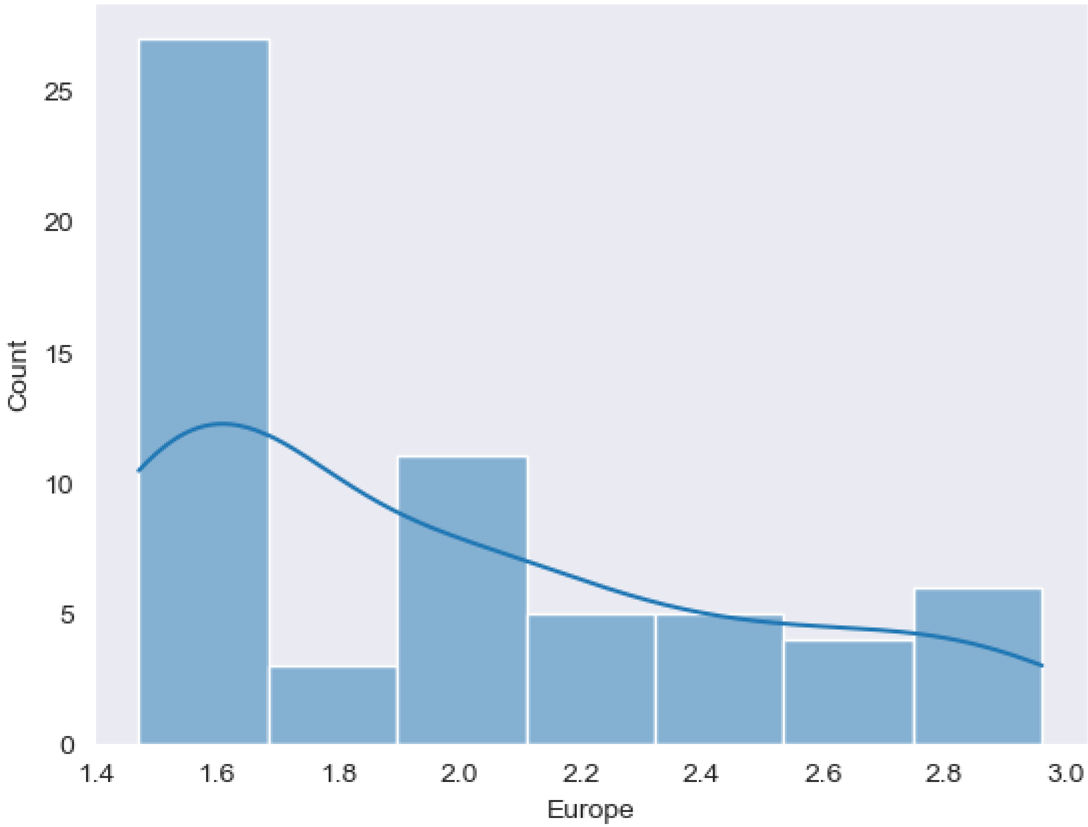
In [31]:
```
sns.histplot(df['South America'], kde = True)
```

Out[31]:
```
<AxesSubplot:xlabel='South America', ylabel='Count'>
```



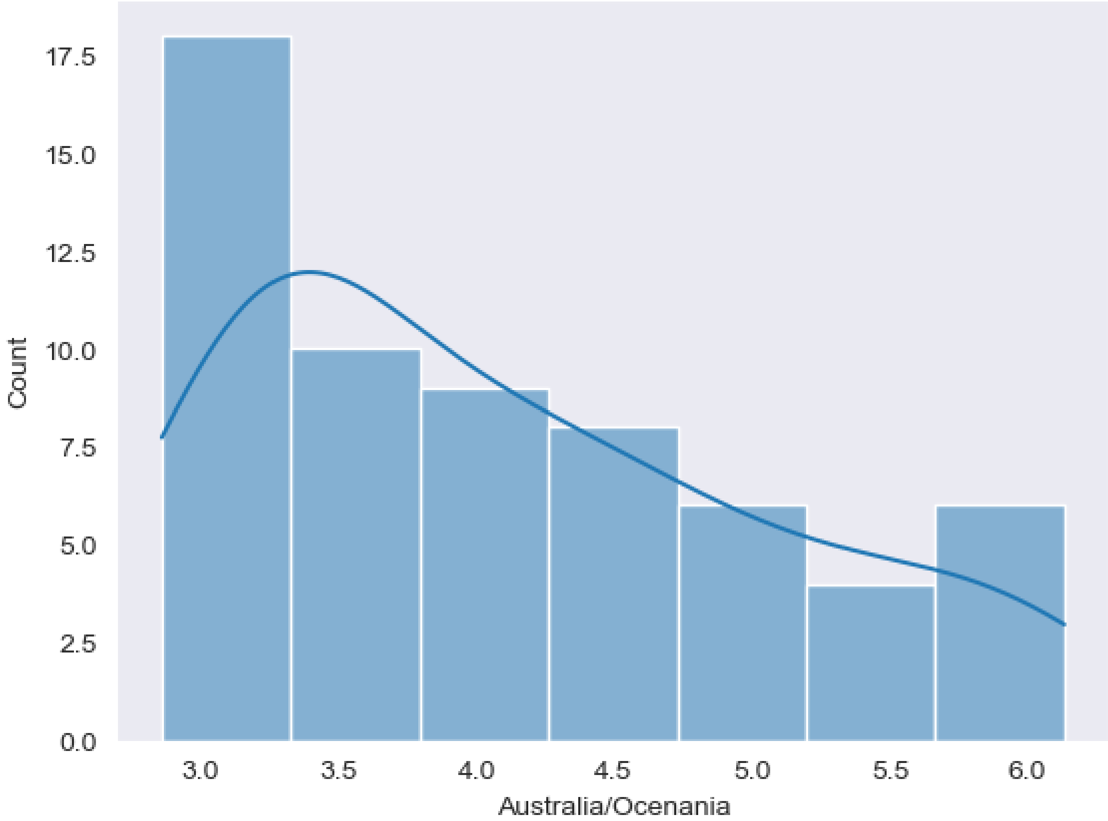
In [32]:
```
sns.histplot(df['Europe'], kde = True)
```

Out[32]:
```
<AxesSubplot:xlabel='Europe', ylabel='Count'>
```

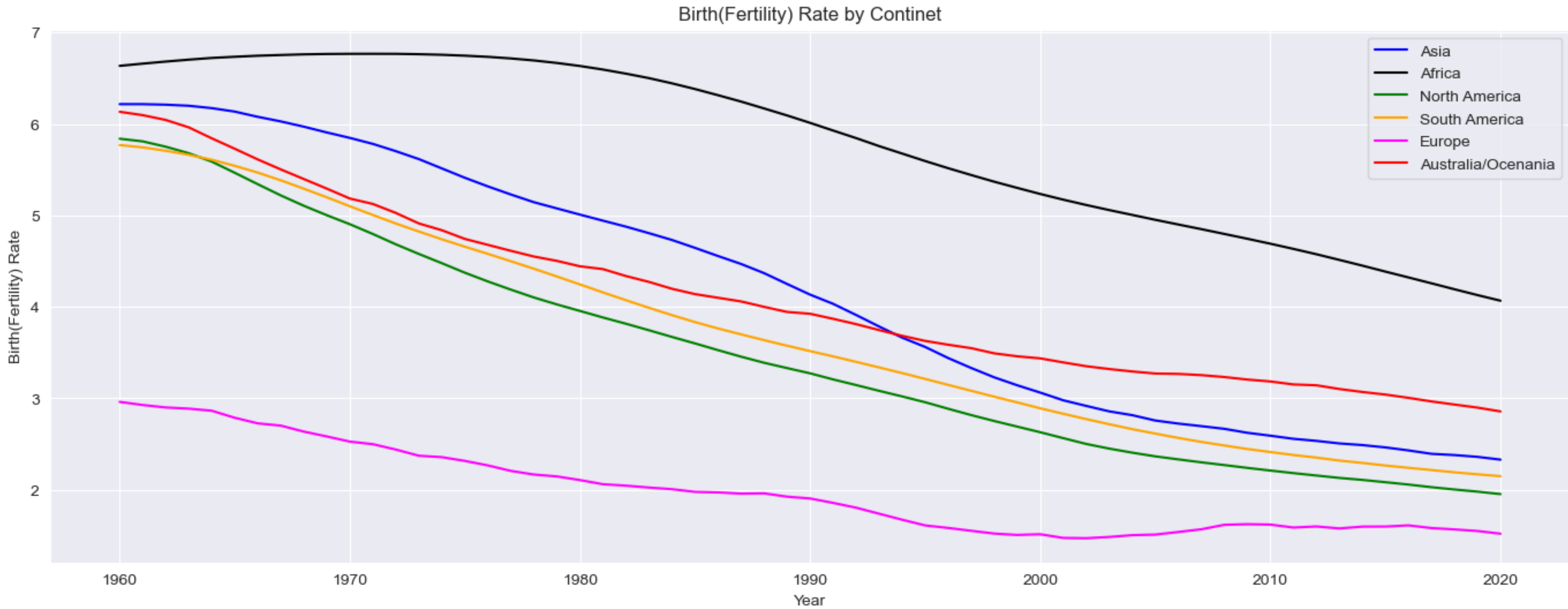


In [33]:
```
sns.histplot(df['Australia/Oceania'], kde = True)
```

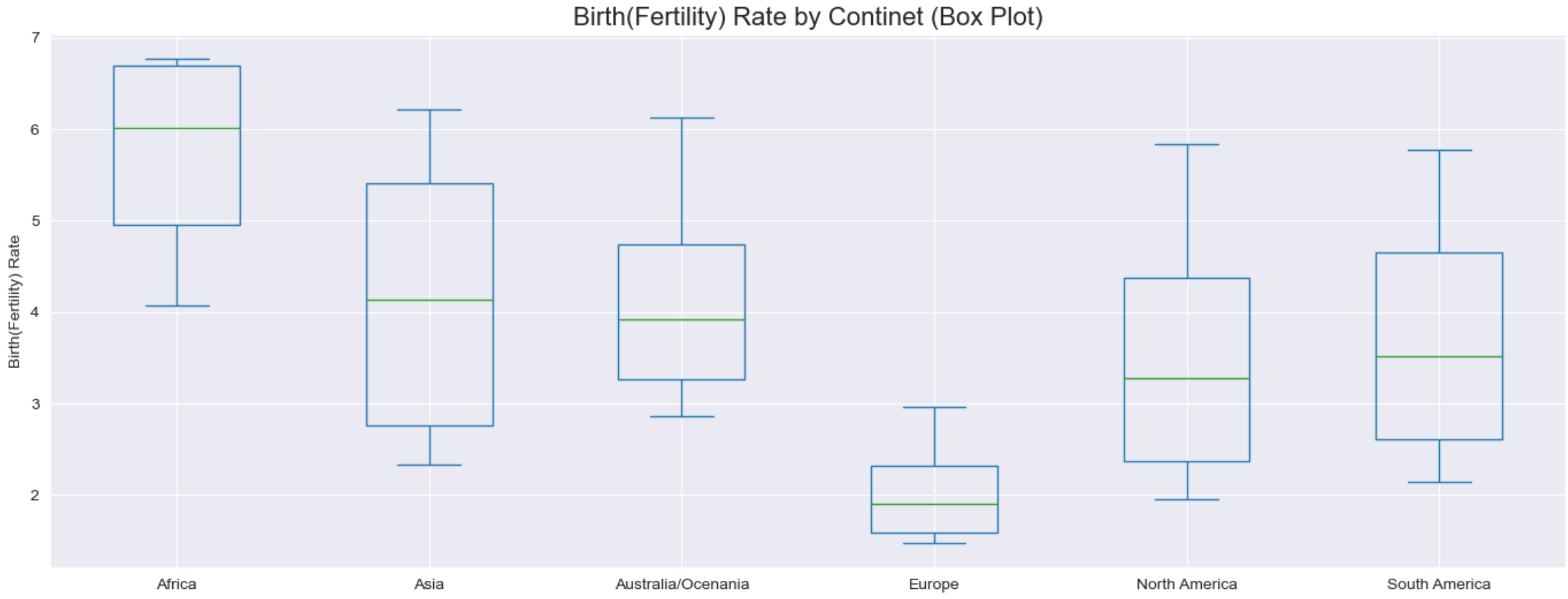
Out[33]:
```
<AxesSubplot:xlabel='Australia/Oceania', ylabel='Count'>
```



In [34]:
```
plt.figure(figsize=(17,6))
plt.plot(df.Year,df['Asia'], color = 'blue', label = 'Asia')
plt.plot(df.Year,df['Africa'], color = 'black', label = 'Africa')
plt.plot(df.Year,df['North America'], color = 'green', label = 'North America')
plt.plot(df.Year,df['South America'], color = 'orange', label = 'South America')
plt.plot(df.Year,df['Europe'], color = 'magenta', label = 'Europe')
plt.plot(df.Year,df['Australia/Ocenania'], color = 'red', label = 'Australia/Ocenania')
plt.xlabel('Year')
plt.ylabel('Birth(Fertility) Rate')
plt.title('Birth(Fertility) Rate by Continet')
plt.legend(loc = 'upper right')
plt.grid(True)
plt.show()
```

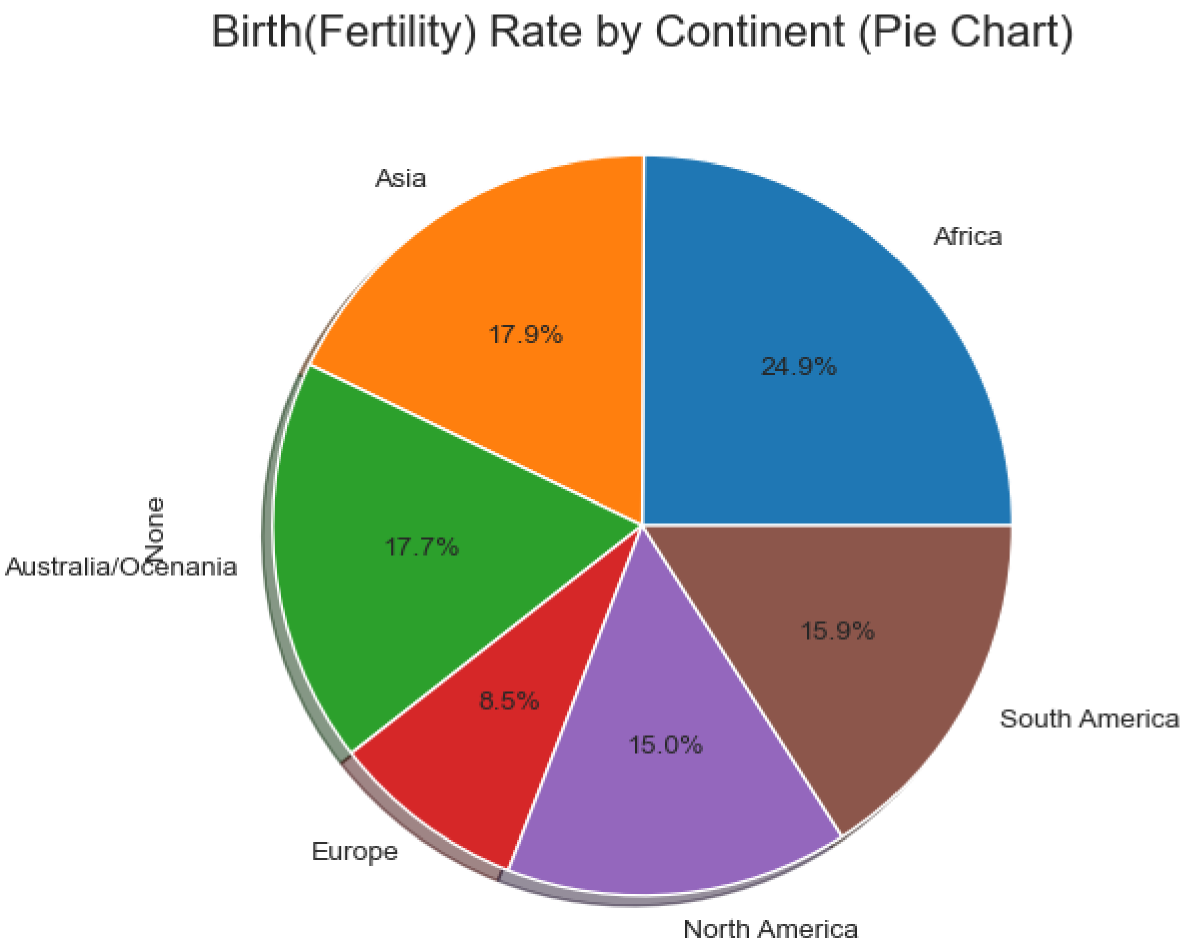


In [35]:
```
df[['Africa','Asia','Australia/Ocenania','Europe','North America','South America']].plot(kind='box',figsize=(17,6))
plt.ylabel('Birth(Fertility) Rate')
plt.title('Birth(Fertility) Rate by Continet (Box Plot)', fontsize=16)
plt.grid(True)
```

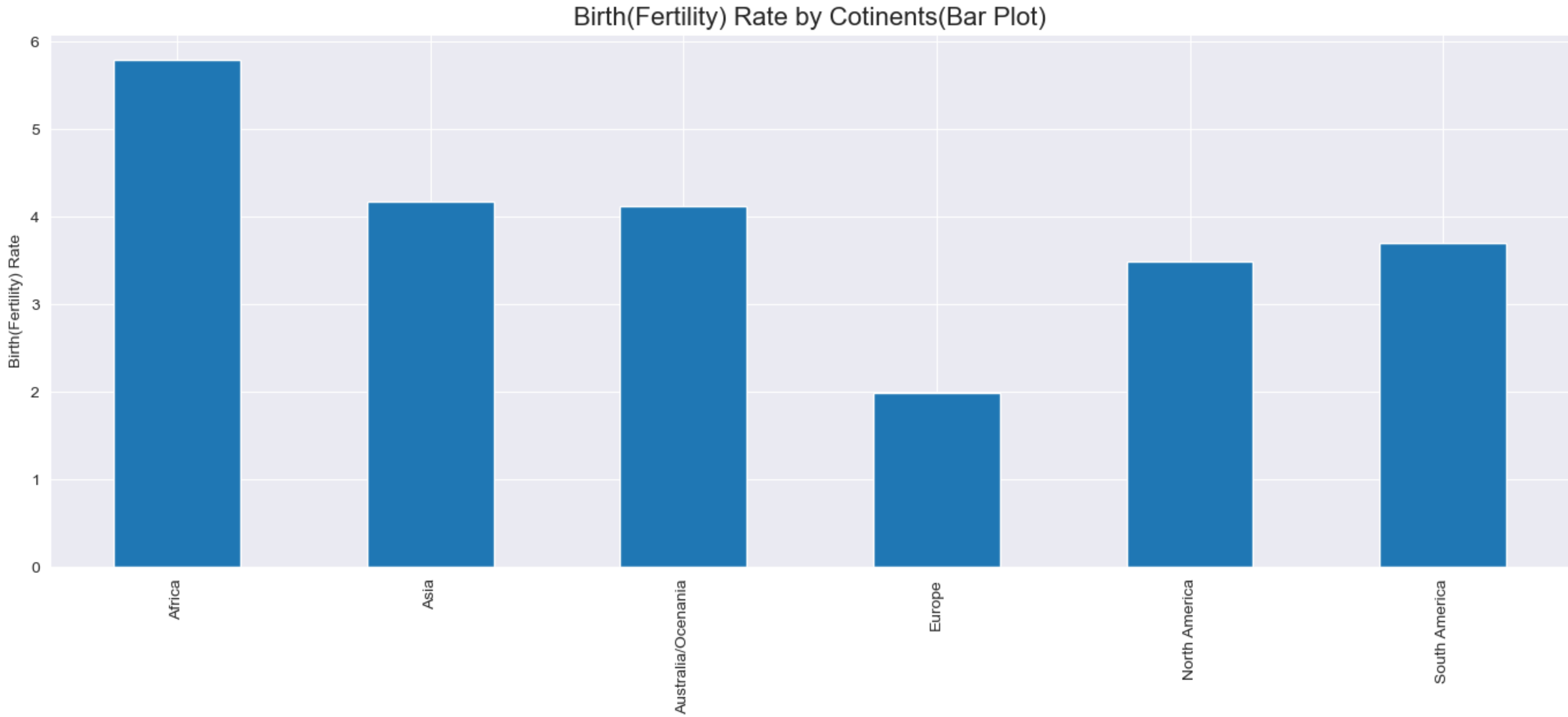


In [36]:

```
c=df[['Africa','Asia','Australia/Ocenania','Europe','North America','South America']].mean()
c.plot(kind='pie', figsize=(7,6),shadow = True, autopct='%1.1f%%')
plt.title('Birth(Fertility) Rate by Continent (Pie Chart)', fontsize=16)
plt.show()
```

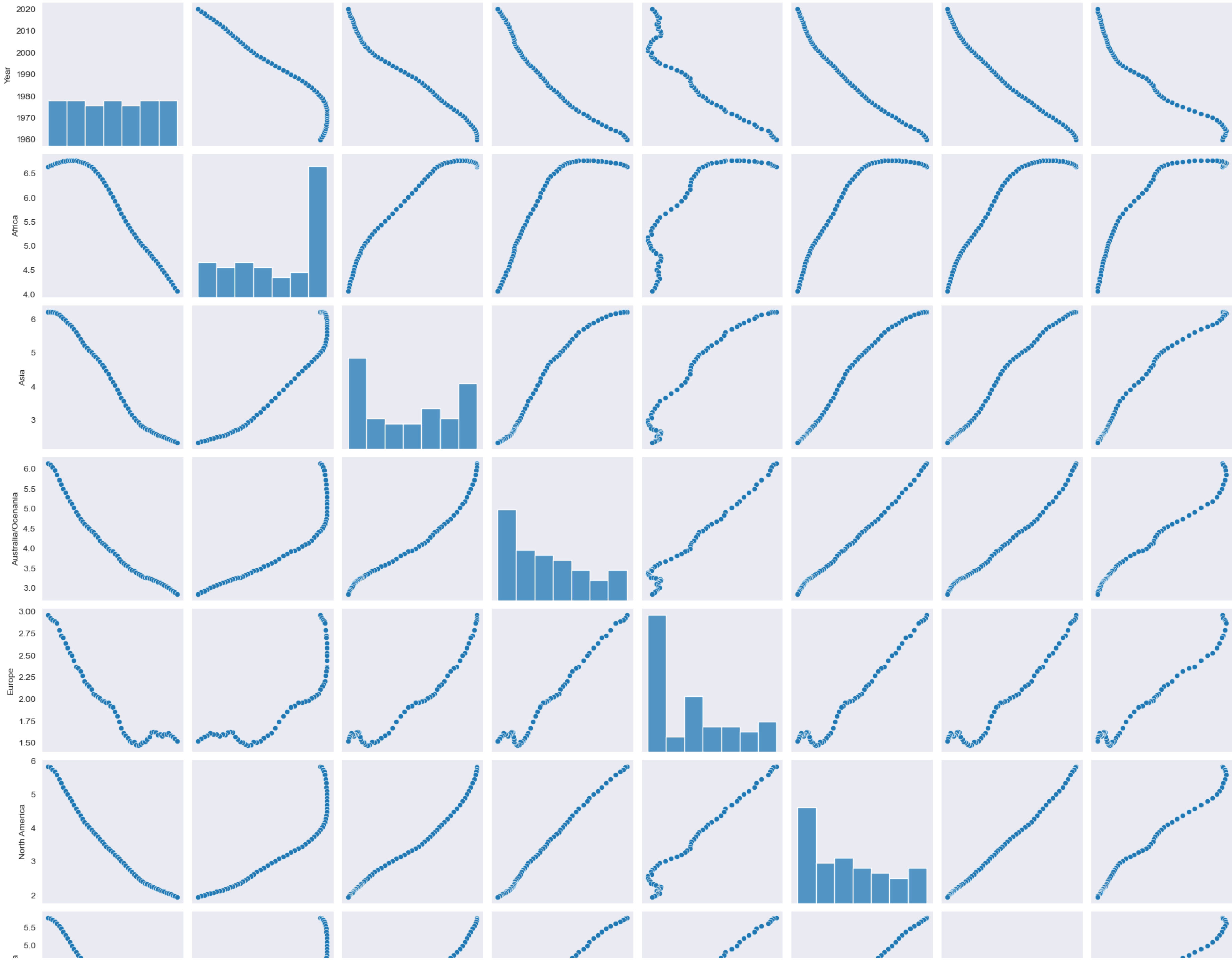


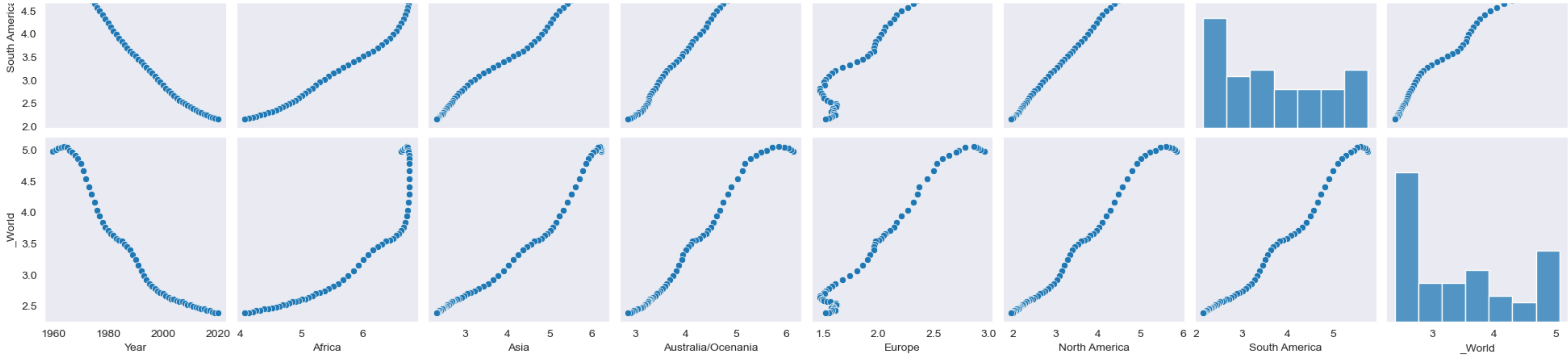
In [37]:
```python
c=df[['Africa','Asia','Australia/Oceania','Europe','North America','South America']].mean()
c.plot(kind='bar',figsize=(17,6))
plt.ylabel('Birth(Fertility) Rate')
plt.title('Birth(Fertility) Rate by Cotinents(Bar Plot)', fontsize=16)
plt.grid(True)
plt.show()
```



In [38]:
```
sns.pairplot(df[['Year','Africa','Asia','Australia/Oceania','Europe',
                 'North America','South America','_World']], kind="scatter")
```

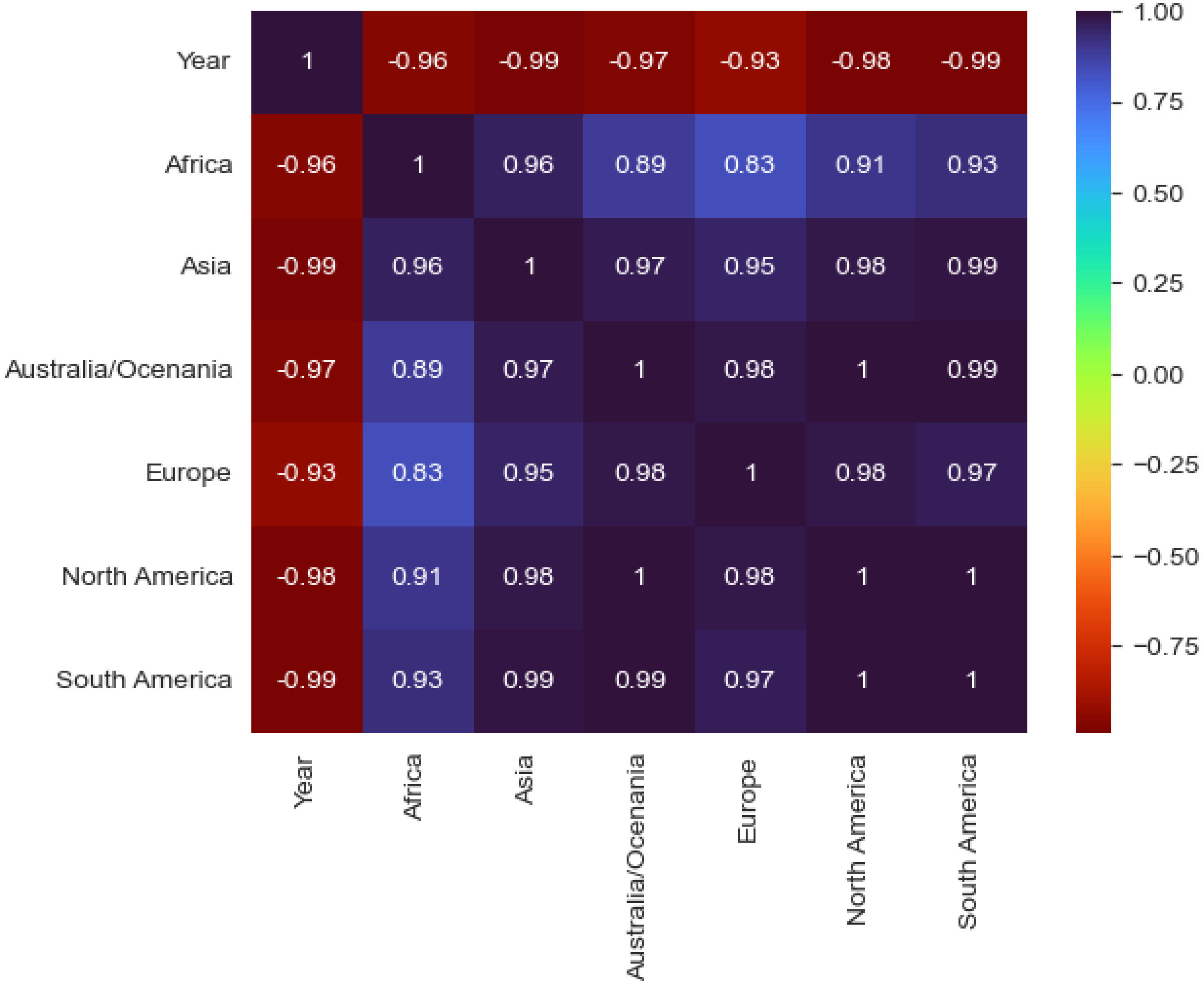
Out[38]:
```
<seaborn.axisgrid.PairGrid at 0x2318c2b0190>
```


Year
Africa
Asia
Australia/Oceania
Europe
North America



In [39]:
```python
df2=df[['Year','Africa','Asia','Australia/Ocenania','Europe','North America','South America']]
sns.heatmap(df2.corr(),annot=True,cmap='turbo_r')
```

Out[39]:
```
<AxesSubplot:>
```



	Year	Africa	Asia	Australia/Ocenania	Europe	North America	South America
Year	1	-0.96	-0.99	-0.97	-0.93	-0.98	-0.99
Africa	-0.96	1	0.96	0.89	0.83	0.91	0.93
Asia	-0.99	0.96	1	0.97	0.95	0.98	0.99
Australia/Ocenania	-0.97	0.89	0.97	1	0.98	1	0.99
Europe	-0.93	0.83	0.95	0.98	1	0.98	0.97
North America	-0.98	0.91	0.98	1	0.98	1	1
South America	-0.99	0.93	0.99	0.99	0.97	1	1

Apply Model For Continents Using (Linear Regression)

In [40]:
```
from sklearn.linear_model import LinearRegression
```

In [41]:
```
def LinerRegression(y, c, color, rs):
    # Create a linear regression model
    X=df['Year']
    X=np.array(X).reshape(-1,1)
```

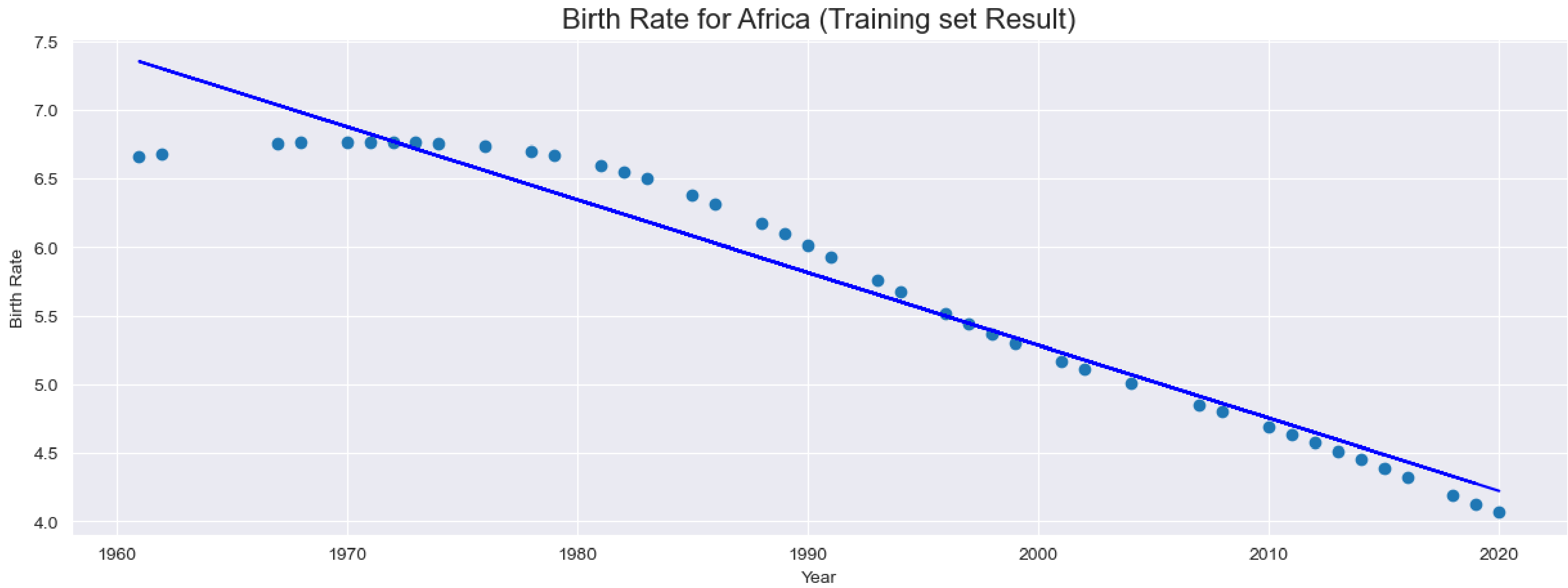
```
    y=np.array(y)
    # Split the data into training and test sets
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2, train_size=0.7, random_state=rs)
    # Train the model on the training set
    model = LinearRegression()
    model.fit(X_train,y_train)
    # Use the trained model to make predictions on the training set
    y_pred = model.predict(X_train)
    plt.figure(figsize=(15,5))
    plt.scatter(X_train, y_train)
    plt.plot(X_train,y_pred, color =color)
    plt.title(f"Birth Rate for {c} (Training set Result)",fontsize=16)
    plt.xlabel('Year')
    plt.ylabel("Birth Rate")
    plt.grid(True)
    plt.show()
    # Print the model's coefficient of determination, intercept,slope etc
    print(f'Coefficient of determination:{model.score(X_train,y_train)} or Accuracy:{round(model.score(X_train,y_train)*100,2)}%
    print('Intercept:', model.intercept_)
    print('slope:', model.coef_)
    print('\n')
    # prediction for 2023,2024 & 2030
    print(f"'brith rate prediction for {c}'")
    x_2023 =np.array([2023]).reshape(-1, 1)
    print('prediction for 2023 birth rate:',model.predict(x_2023))
    x_2024 =np.array([2024]).reshape(-1, 1)
    print('prediction for 2024 birth rate:',model.predict(x_2024))
    x_2030 =np.array([2030]).reshape(-1, 1)
    print('prediction for 2030 birth rate:',model.predict(x_2030))
    x_2080 =np.array([2080]).reshape(-1, 1)
    print('prediction for 2080 birth rate:',model.predict(x_2080))
```

In [ ]:

Linear Regression for Africa

In [42]:

```
LinerRegression(df['Africa'], 'Africa', 'b', 44)
```




```
Coefficient of determination:0.9462904807926936 or Accuracy:94.63%
Intercept: 111.4013551433403
slope: [-0.05305846]


'brith rate prediction for Africa'
prediction for 2023 birth rate: [4.06409719]
prediction for 2024 birth rate: [4.01103874]
prediction for 2030 birth rate: [3.692688]
prediction for 2080 birth rate: [1.03976516]
```

In [ ]:

Linear Regression for Asia

In [43]:

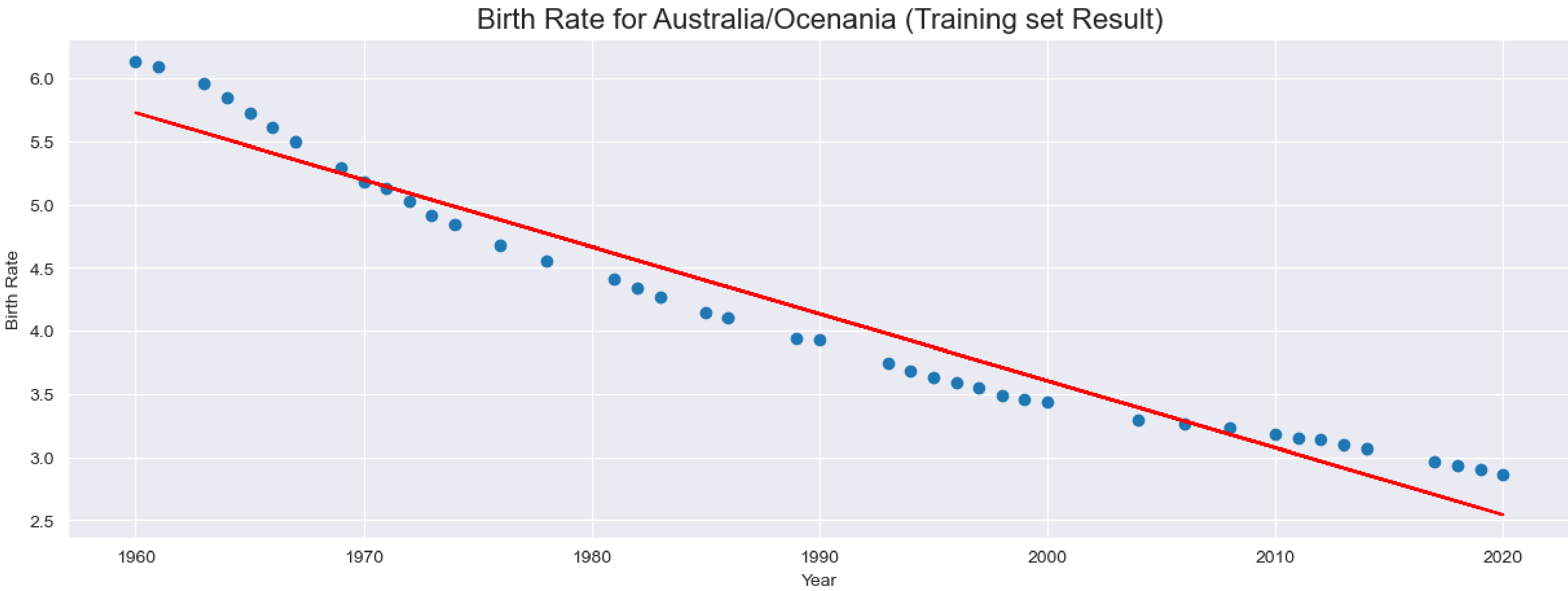
```
LinerRegression(df['Asia'], 'Asia', 'g', 44)
```


```
Coefficient of determination:0.9769387191766975 or Accuracy:97.69%
Intercept: 155.01560764198584
slope: [-0.07579848]


'brith rate prediction for Asia'
prediction for 2023 birth rate: [1.67528976]
prediction for 2024 birth rate: [1.59949128]
prediction for 2030 birth rate: [1.14470043]
prediction for 2080 birth rate: [-2.6452234]
```

In [ ]:

Linear Regression for Australia/Oceania

In [44]:
```
LinerRegression(df['Australia/Oceania'] , 'Australia/Oceania', 'r', 88)
```



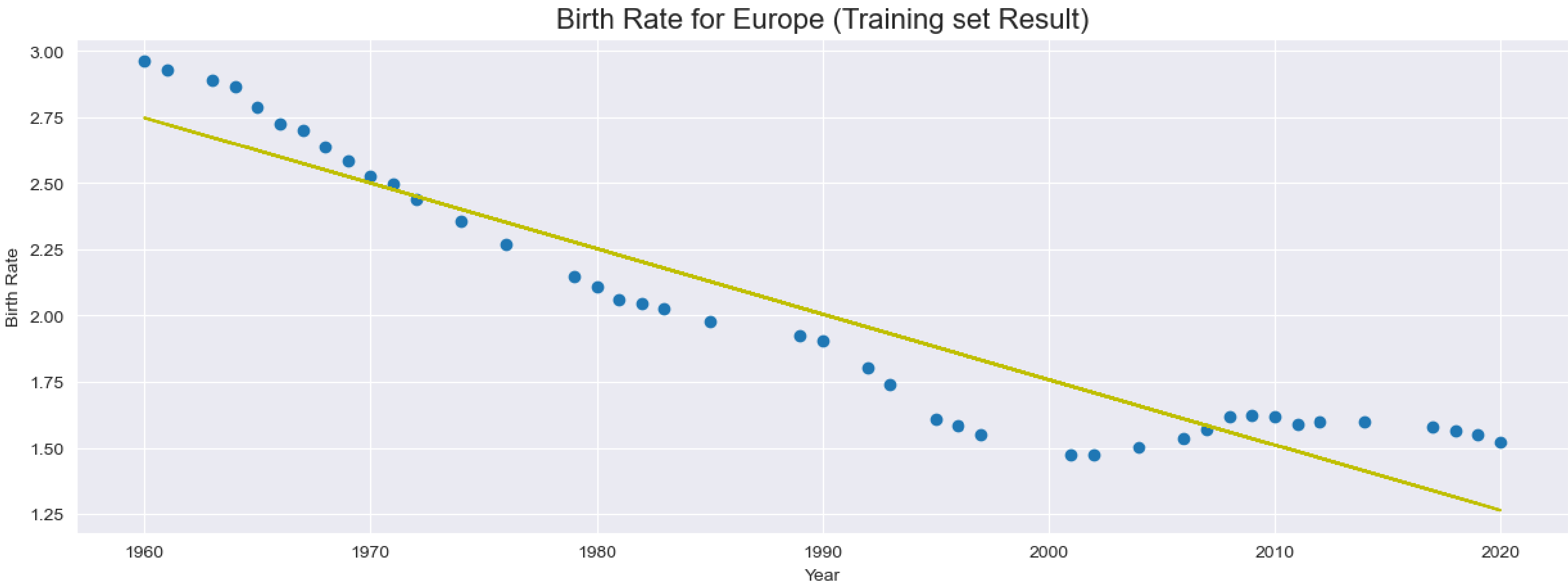

```
Coefficient of determination:0.9500665063850735 or Accuracy:95.01%
Intercept: 109.79038865182868
slope: [-0.05309315]


'brith rate prediction for Australia/Ocenania'
prediction for 2023 birth rate: [2.38294457]
prediction for 2024 birth rate: [2.32985142]
prediction for 2030 birth rate: [2.01129252]
prediction for 2080 birth rate: [-0.64336503]
```

In [ ]:

Linear Regression for Europe

In [45]:

```
LinerRegression(df['Europe'] , 'Europe', 'y', 123)
```



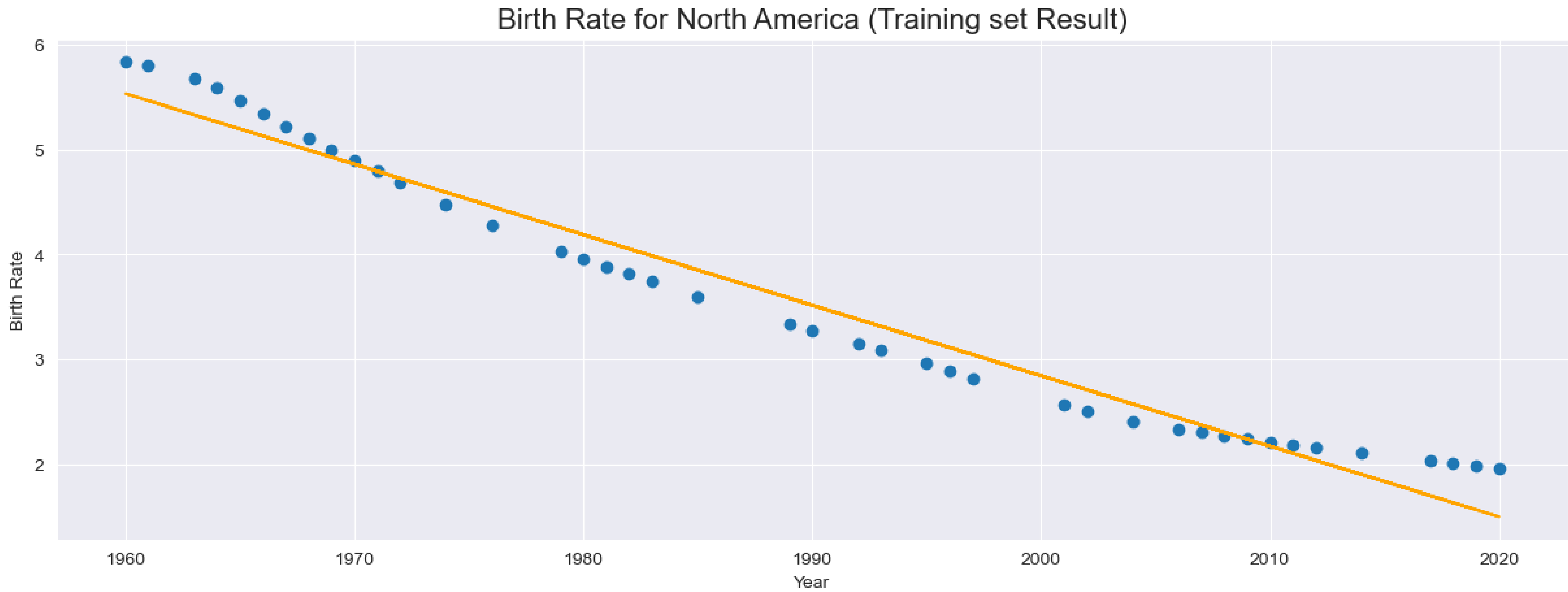

```
Coefficient of determination:0.8818713071238642 or Accuracy:88.19%
Intercept: 51.21345159467995
slope: [-0.0247275]


'brith rate prediction for Europe'
prediction for 2023 birth rate: [1.18971831]
prediction for 2024 birth rate: [1.16499081]
prediction for 2030 birth rate: [1.01662581]
prediction for 2080 birth rate: [-0.21974921]
```

In [ ]:

Linear Regression for North America

In [46]:

```
LinerRegression(df['North America'] , 'North America', 'orange', 123)
```




```
Coefficient of determination:0.9681579481293696 or Accuracy:96.82%
Intercept: 137.48545290781
slope: [-0.06732128]


'brith rate prediction for North America'
prediction for 2023 birth rate: [1.29449994]
prediction for 2024 birth rate: [1.22717865]
prediction for 2030 birth rate: [0.82325096]
prediction for 2080 birth rate: [-2.54281312]
```

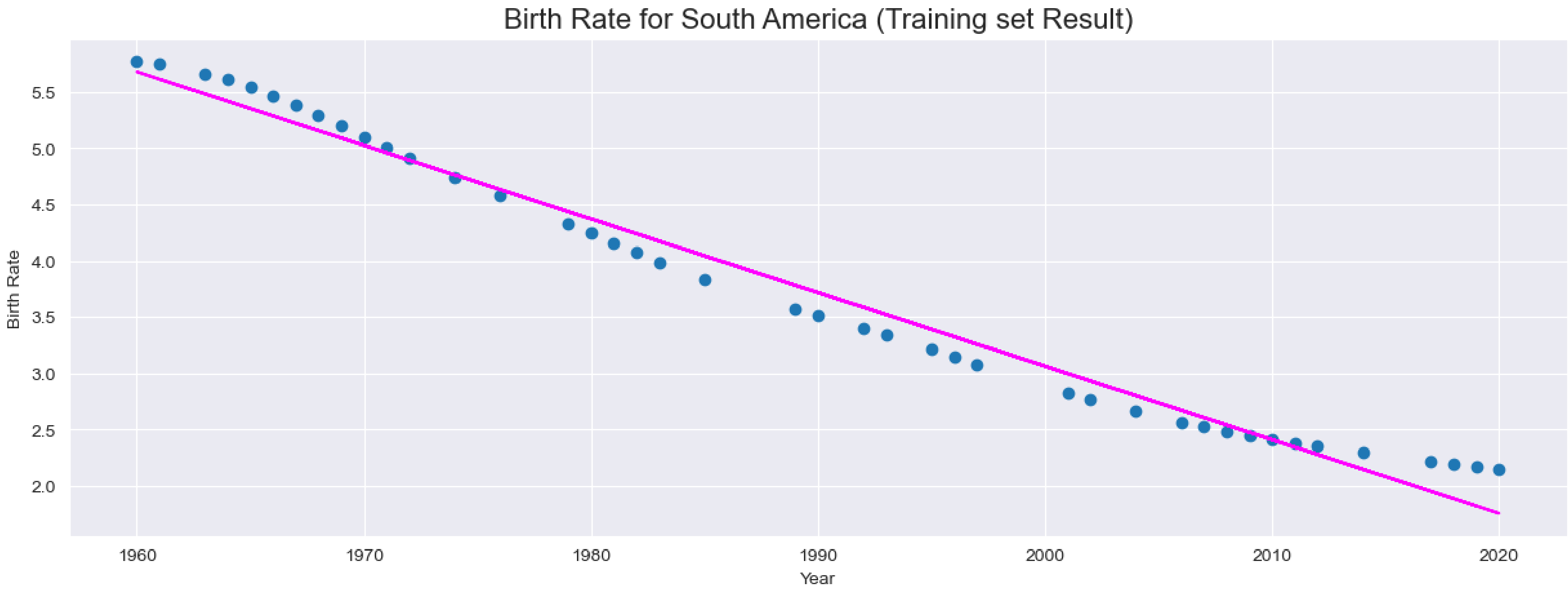
In [ ]:

```

```

Linear Regression for South America

In [47]:

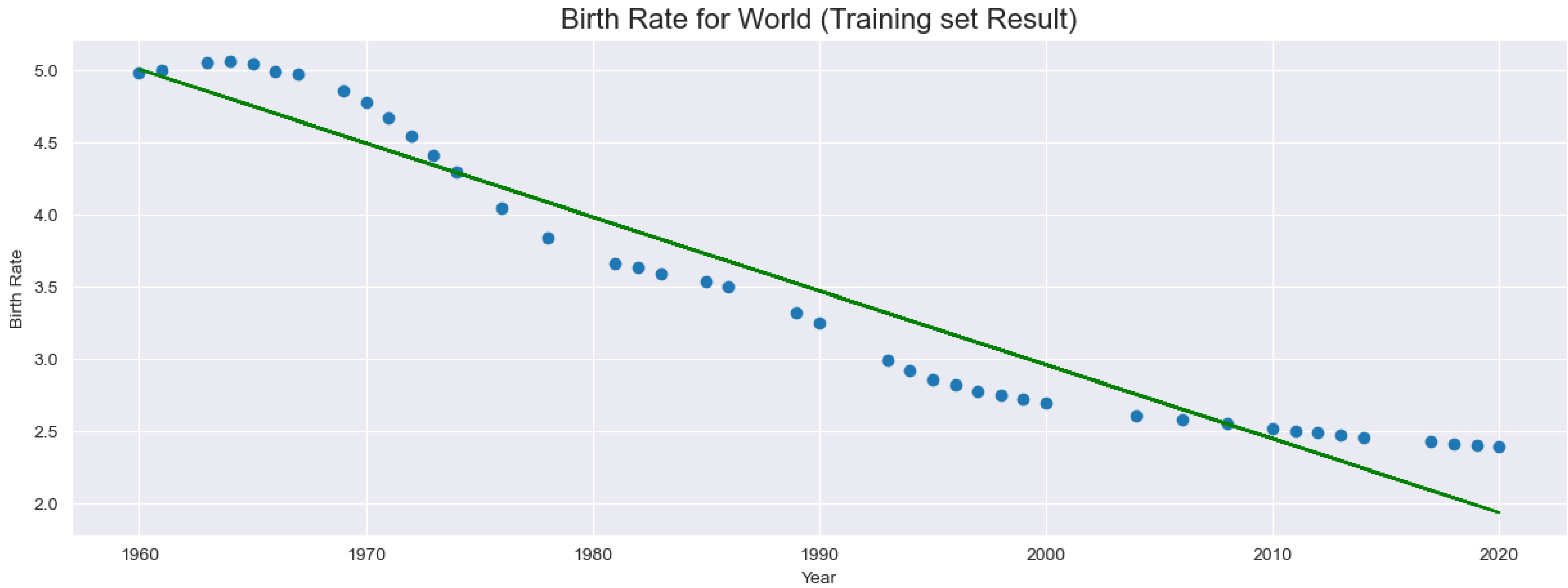
```
LinerRegression(df['South America'] , 'South America', 'magenta', 123)
```




```
Coefficient of determination:0.9813494026864127 or Accuracy:98.13%
Intercept: 133.73470109393025
slope: [-0.06533503]


'brith rate prediction for South America'
prediction for 2023 birth rate: [1.56192662]
prediction for 2024 birth rate: [1.49659158]
prediction for 2030 birth rate: [1.10458138]
prediction for 2080 birth rate: [-2.16217034]
```

World

In [48]:
```
LinerRegression(df['_World'] , 'World', 'g', 88)
```




```
Coefficient of determination:0.9335092282598383 or Accuracy:93.35%
Intercept: 105.28112415218705
slope: [-0.05116096]


'brith rate prediction for World'
prediction for 2023 birth rate: [1.78249802]
prediction for 2024 birth rate: [1.73133706]
prediction for 2030 birth rate: [1.42437129]
prediction for 2080 birth rate: [-1.13367681]
```

In [ ]:

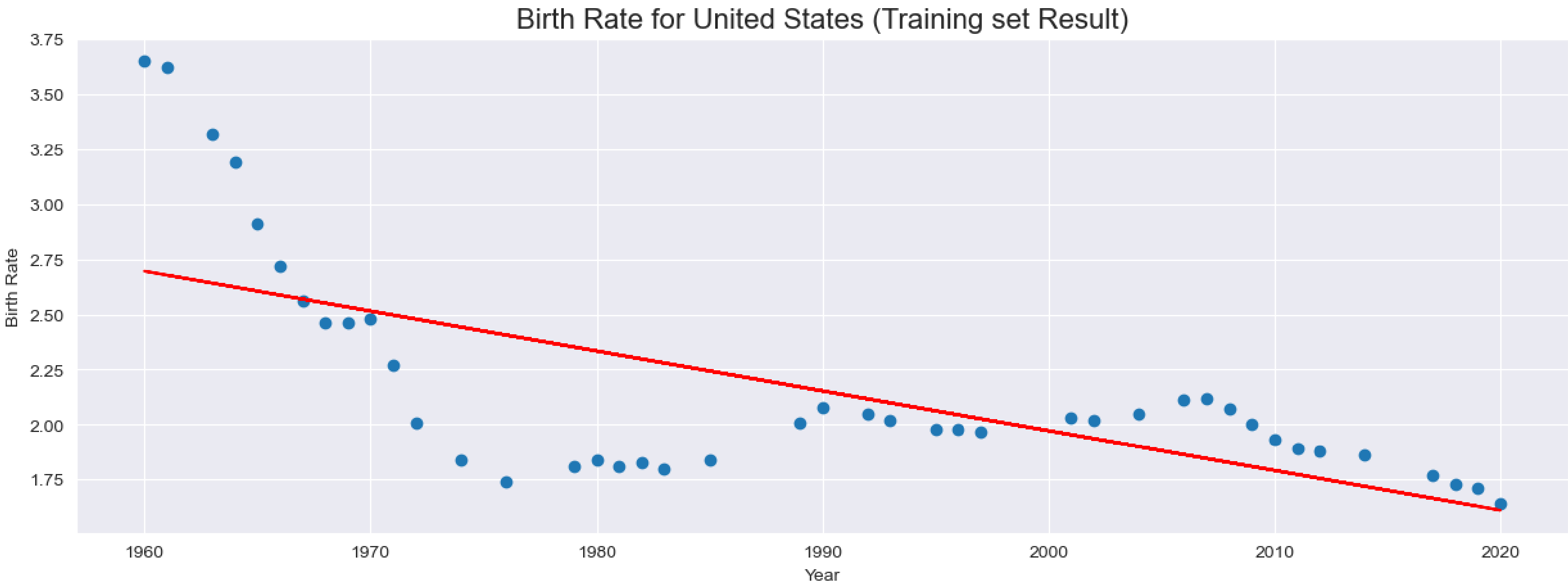
```

```

Apply Model For Some Countries

Unitesd States

In [49]:

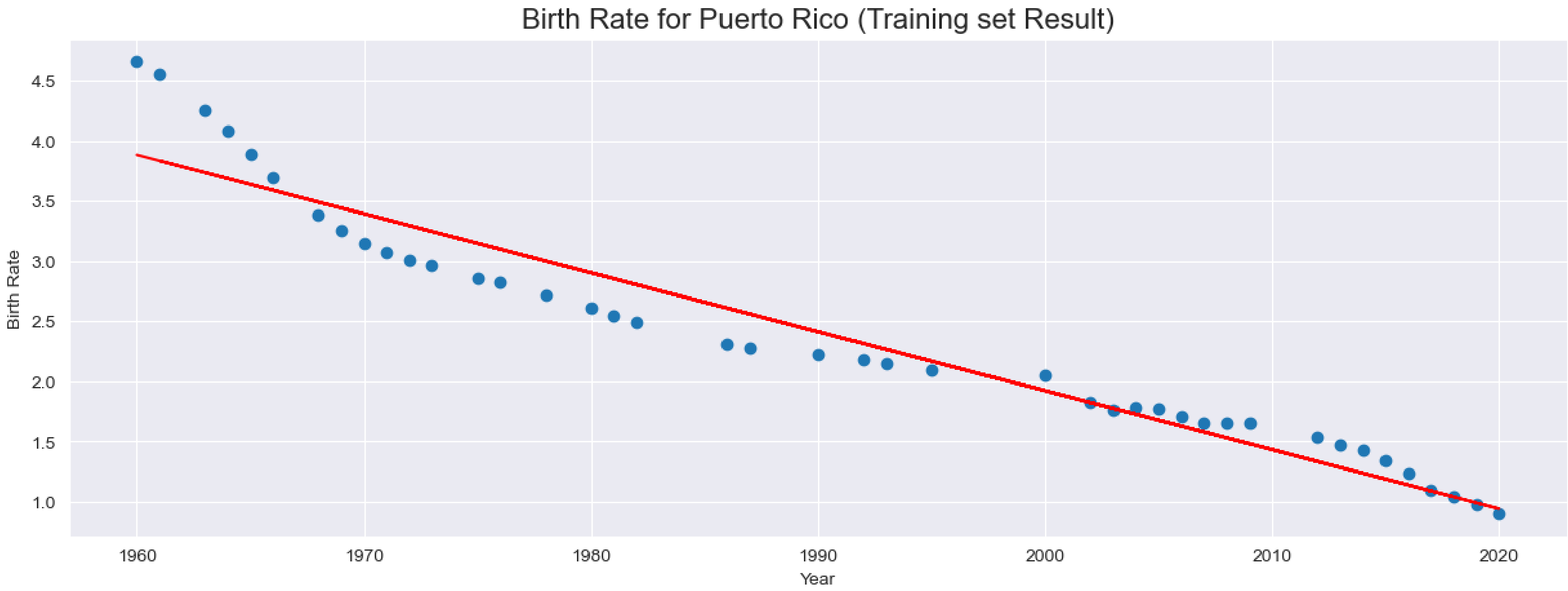
```
LinerRegression(df['United States'], 'United States', 'r', 123)
```




```
Coefficient of determination:0.46242463971106484 or Accuracy:46.24%
Intercept: 38.17030006793257
slope: [-0.01809892]


'brith rate prediction for United States'
prediction for 2023 birth rate: [1.55617928]
prediction for 2024 birth rate: [1.53808035]
prediction for 2030 birth rate: [1.42948682]
prediction for 2080 birth rate: [0.52454068]
```

Puerto Rico

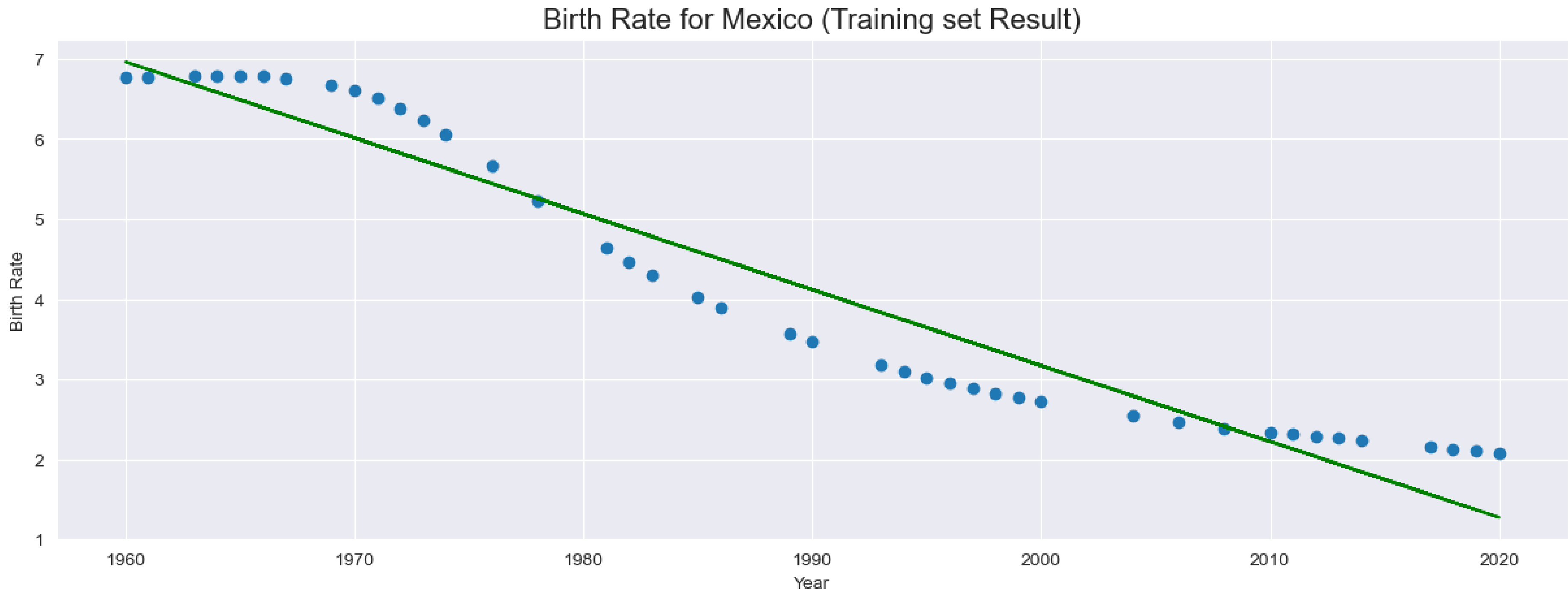
In [50]:
```
LinerRegression(df['Puerto Rico'], 'Puerto Rico', 'r', 124)
```




```
Coefficient of determination:0.9275914393029608 or Accuracy:92.76%
Intercept: 99.96369550953234
slope: [-0.04902046]


'brith rate prediction for Puerto Rico'
prediction for 2023 birth rate: [0.79531332]
prediction for 2024 birth rate: [0.74629287]
prediction for 2030 birth rate: [0.45217013]
prediction for 2080 birth rate: [-1.99885266]
```

Mexico

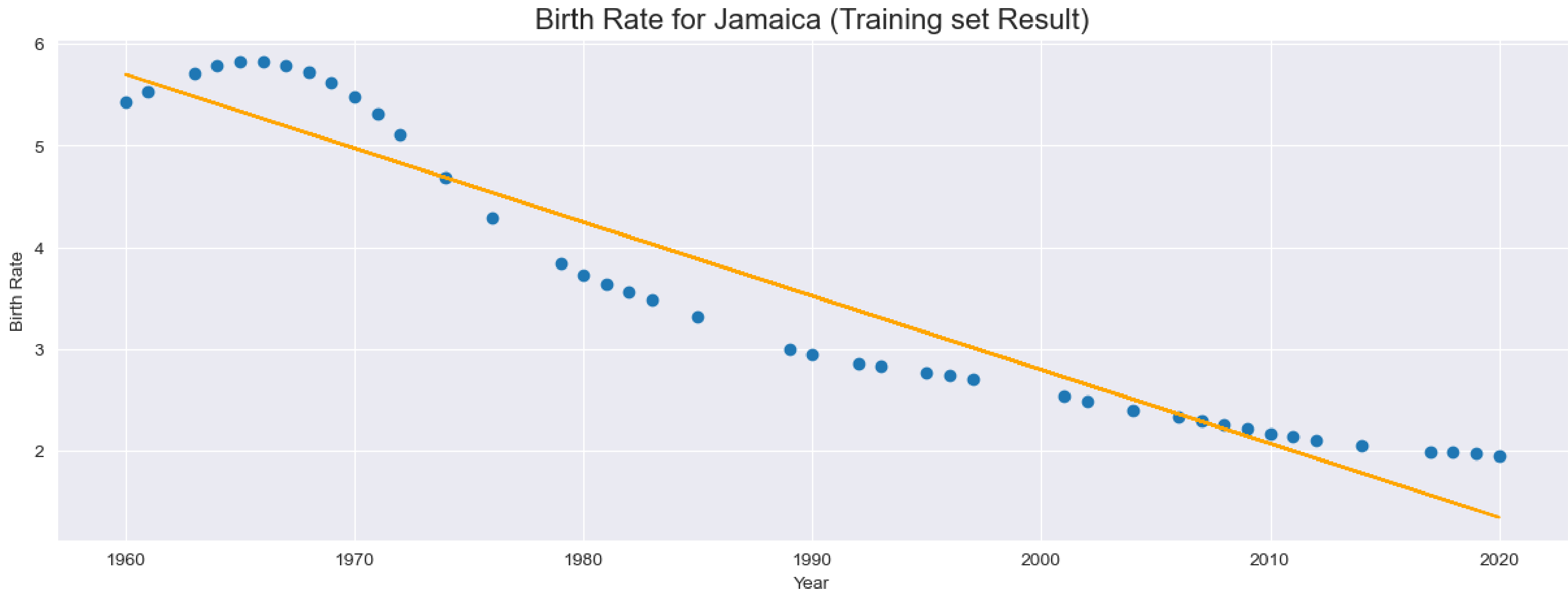
In [51]:
```
LinerRegression(df['Mexico'], 'Mexico', 'g', 88 )
```




```
Coefficient of determination:0.9313254731835299 or Accuracy:93.13%
Intercept: 192.66887168035063
slope: [-0.09474898]


'brith rate prediction for Mexico'
prediction for 2023 birth rate: [0.99168319]
prediction for 2024 birth rate: [0.89693421]
prediction for 2030 birth rate: [0.32844032]
prediction for 2080 birth rate: [-4.40900873]
```

Jamaica

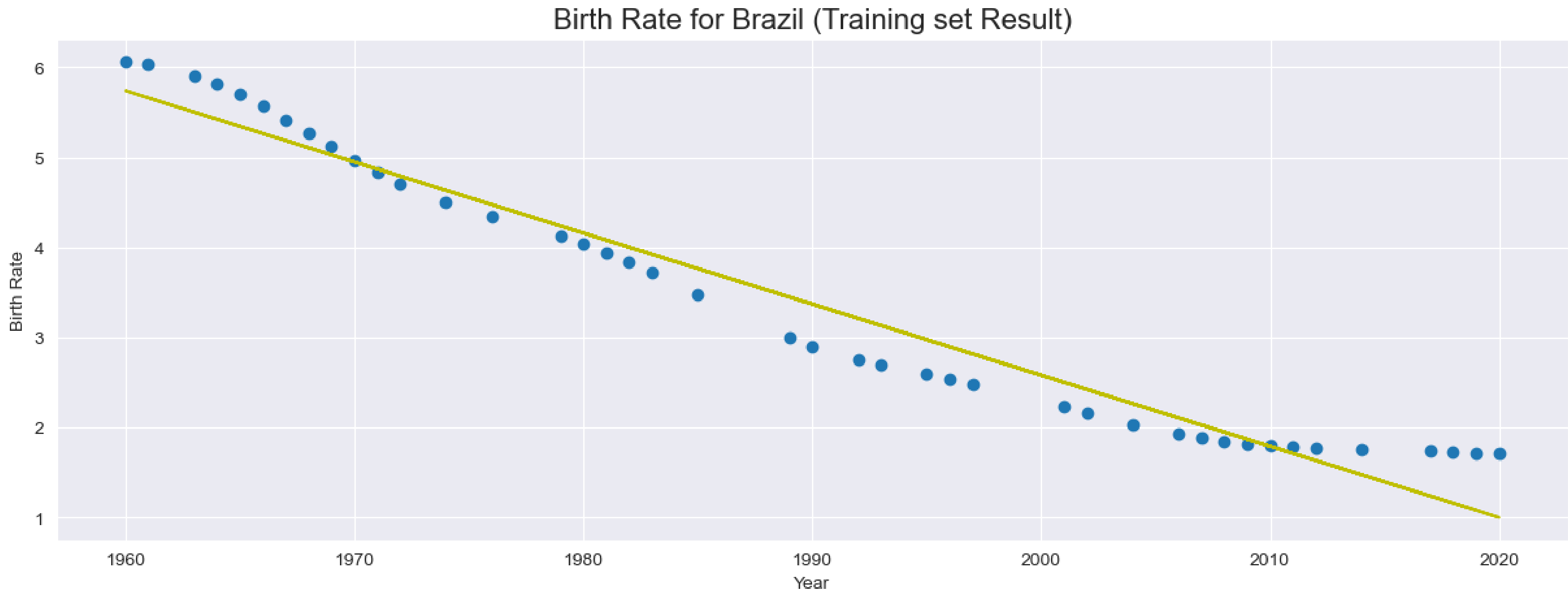
In [52]:
```
LinerRegression(df['Jamaica'], 'Jamaica', 'orange', 123)
```




```
Coefficient of determination:0.9171641283814309 or Accuracy:91.72%
Intercept: 147.94740510464848
slope: [-0.0725762]


'brith rate prediction for Jamaica'
prediction for 2023 birth rate: [1.12575712]
prediction for 2024 birth rate: [1.05318093]
prediction for 2030 birth rate: [0.61772374]
prediction for 2080 birth rate: [-3.01108614]
```

Brazil

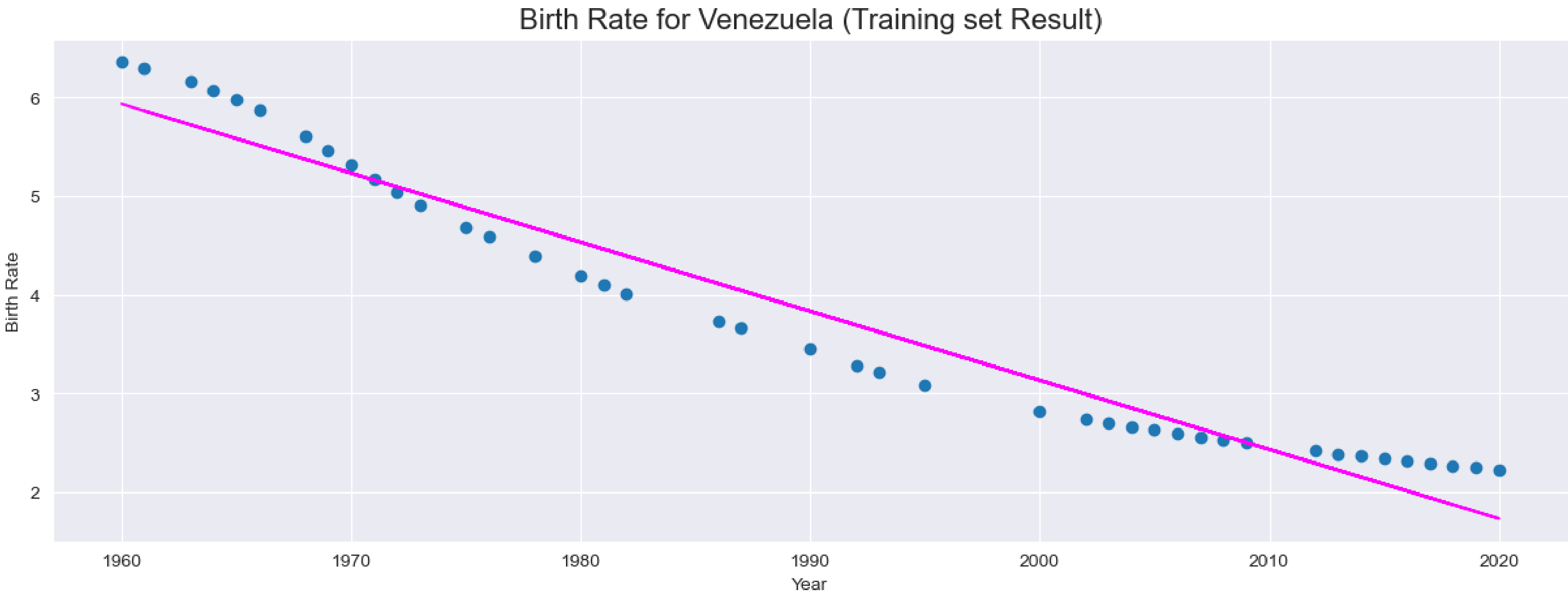
In [53]:
```
LinerRegression(df['Brazil'], 'Brazil', 'y', 123)
```




```
Coefficient of determination:0.9562578942162909 or Accuracy:95.63%
Intercept: 160.5114616504383
slope: [-0.07896526]


'brith rate prediction for Brazil'
prediction for 2023 birth rate: [0.76474606]
prediction for 2024 birth rate: [0.6857808]
prediction for 2030 birth rate: [0.21198926]
prediction for 2080 birth rate: [-3.73627361]
```

Venezuela

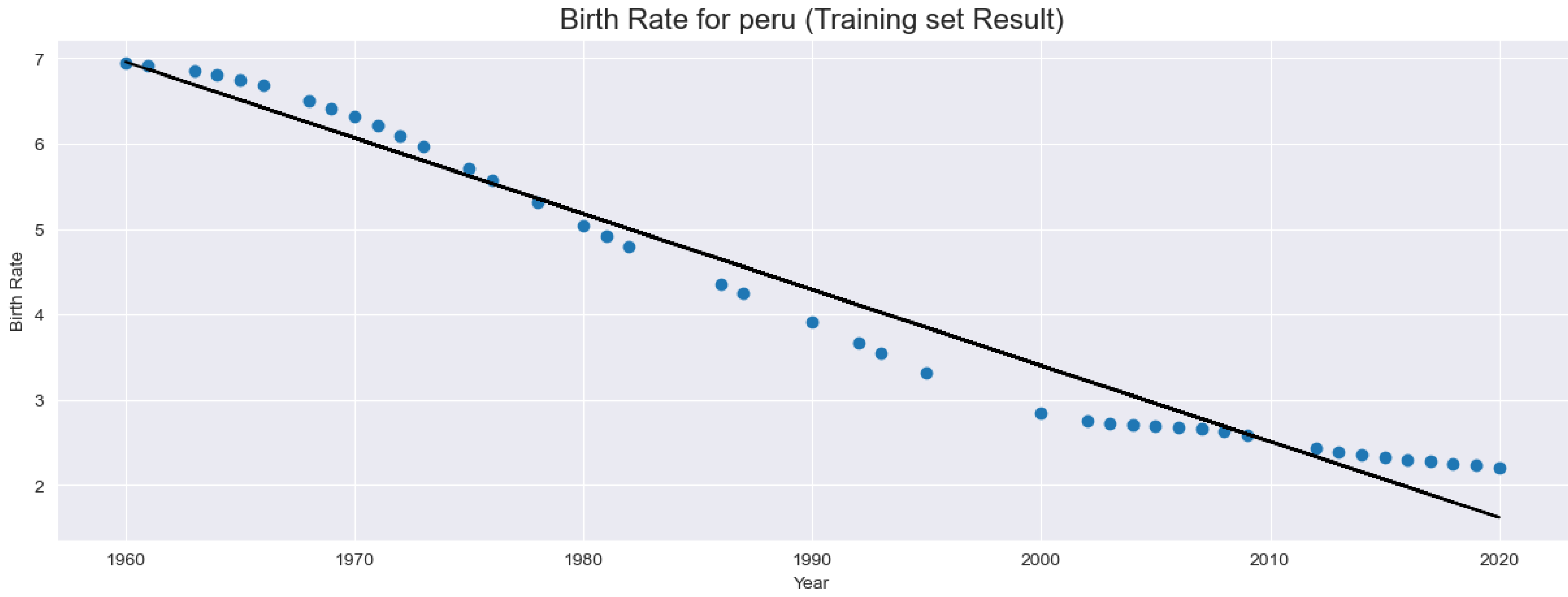
In [54]:
```
LinerRegression(df['Venezuela'], 'Venezuela', 'magenta', 124)
```




```
Coefficient of determination:0.9521524769053126 or Accuracy:95.22%
Intercept: 143.1739156762523
slope: [-0.07001967]


'brith rate prediction for Venezuela'
prediction for 2023 birth rate: [1.52412452]
prediction for 2024 birth rate: [1.45410485]
prediction for 2030 birth rate: [1.03398684]
prediction for 2080 birth rate: [-2.46699663]
```

Peru

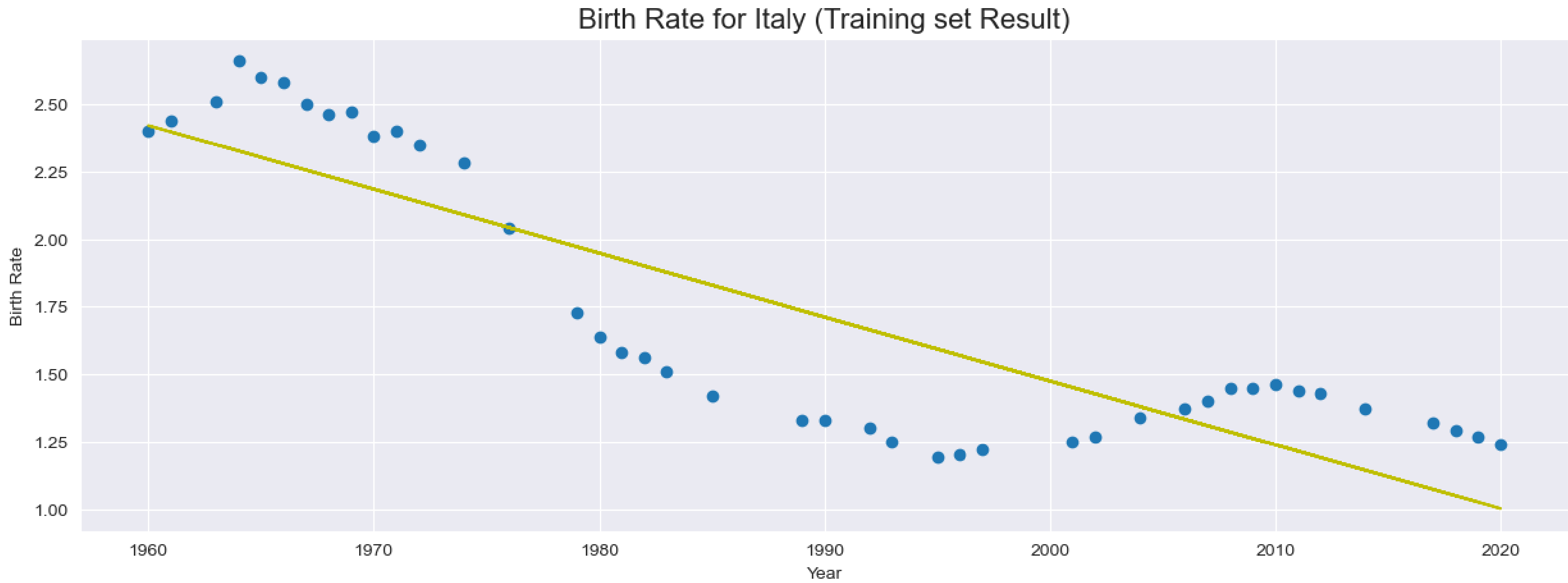
In [55]:
```
LinerRegression(df['Peru'], 'peru', 'black', 124)
```




```
Coefficient of determination:0.970463151800906 or Accuracy:97.05%
Intercept: 181.25822131536316
slope: [-0.08892872]


'brith rate prediction for peru'
prediction for 2023 birth rate: [1.35542899]
prediction for 2024 birth rate: [1.26650028]
prediction for 2030 birth rate: [0.73292798]
prediction for 2080 birth rate: [-3.71350782]
```

Italy

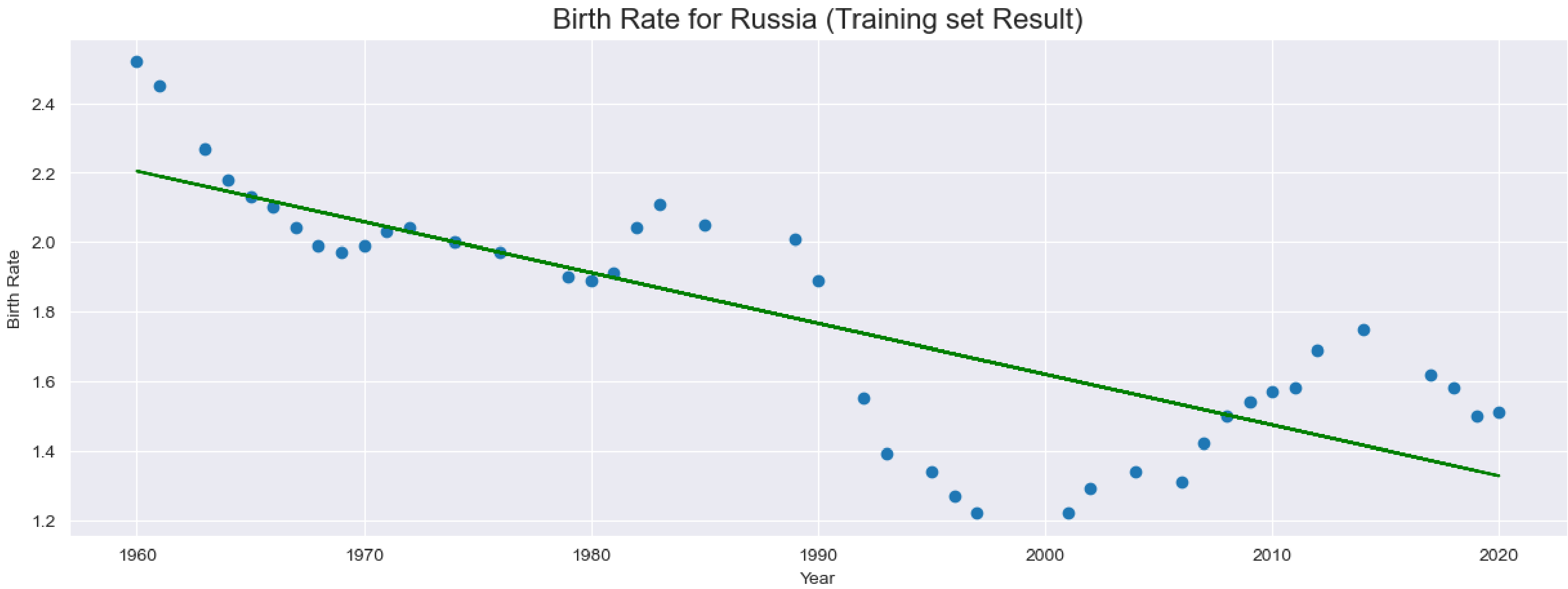
In [56]:
```
LinerRegression(df['Italy'], 'Italy', 'y', 123)
```




```
Coefficient of determination:0.7364060101971547 or Accuracy:73.64%
Intercept: 48.78355332707921
slope: [-0.02365438]


'brith rate prediction for Italy'
prediction for 2023 birth rate: [0.9307327]
prediction for 2024 birth rate: [0.90707832]
prediction for 2030 birth rate: [0.76515201]
prediction for 2080 birth rate: [-0.41756724]
```

Russia

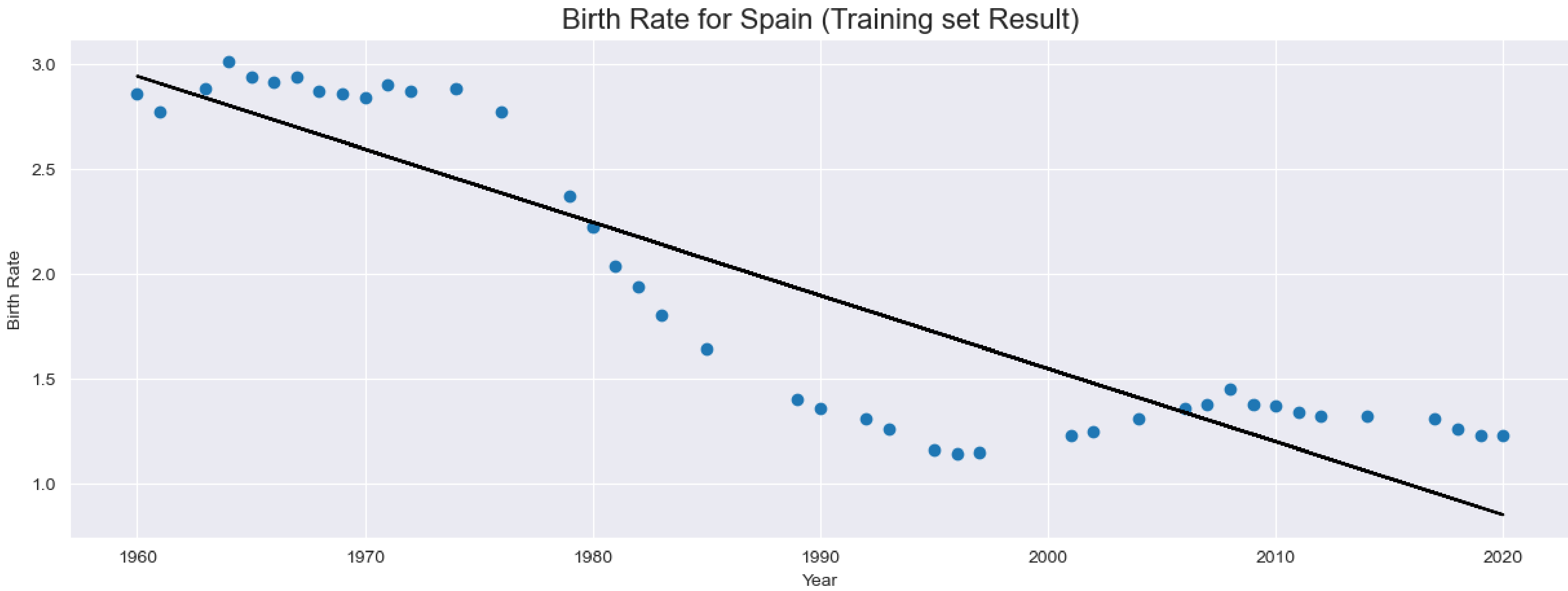
In [57]:
```
LinerRegression(df['Russia'], 'Russia', 'g', 123)
```




```
Coefficient of determination:0.6418997504597603 or Accuracy:64.19%
Intercept: 30.859753695856103
slope: [-0.01461997]


'brith rate prediction for Russia'
prediction for 2023 birth rate: [1.28356306]
prediction for 2024 birth rate: [1.2689431]
prediction for 2030 birth rate: [1.1812233]
prediction for 2080 birth rate: [0.45022502]
```

Spain

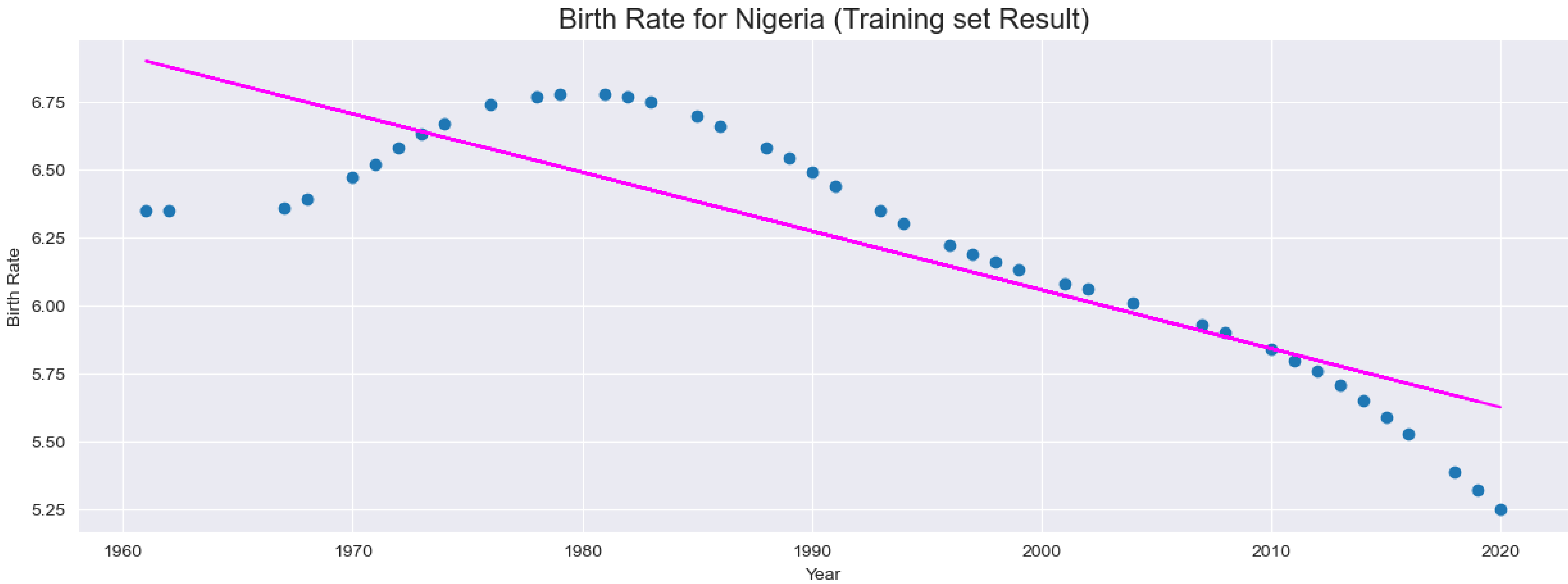
In [58]:
```
LinerRegression(df['Spain'], 'Spain', 'black', 123)
```




```
Coefficient of determination:0.8108439334390722 or Accuracy:81.08%
Intercept: 71.2015424578656
slope: [-0.03482674]


'brith rate prediction for Spain'
prediction for 2023 birth rate: [0.74704833]
prediction for 2024 birth rate: [0.71222159]
prediction for 2030 birth rate: [0.50326115]
prediction for 2080 birth rate: [-1.23807583]
```

Nigeria

In [59]:
```
LinerRegression(df['Nigeria'], 'Nigeria', 'magenta', 44)
```




```
Coefficient of determination:0.7151454877492278 or Accuracy:71.51%
Intercept: 49.203515822473626
slope: [-0.02157276]


'brith rate prediction for Nigeria'
prediction for 2023 birth rate: [5.56181951]
prediction for 2024 birth rate: [5.54024675]
prediction for 2030 birth rate: [5.41081018]
prediction for 2080 birth rate: [4.33217211]
```

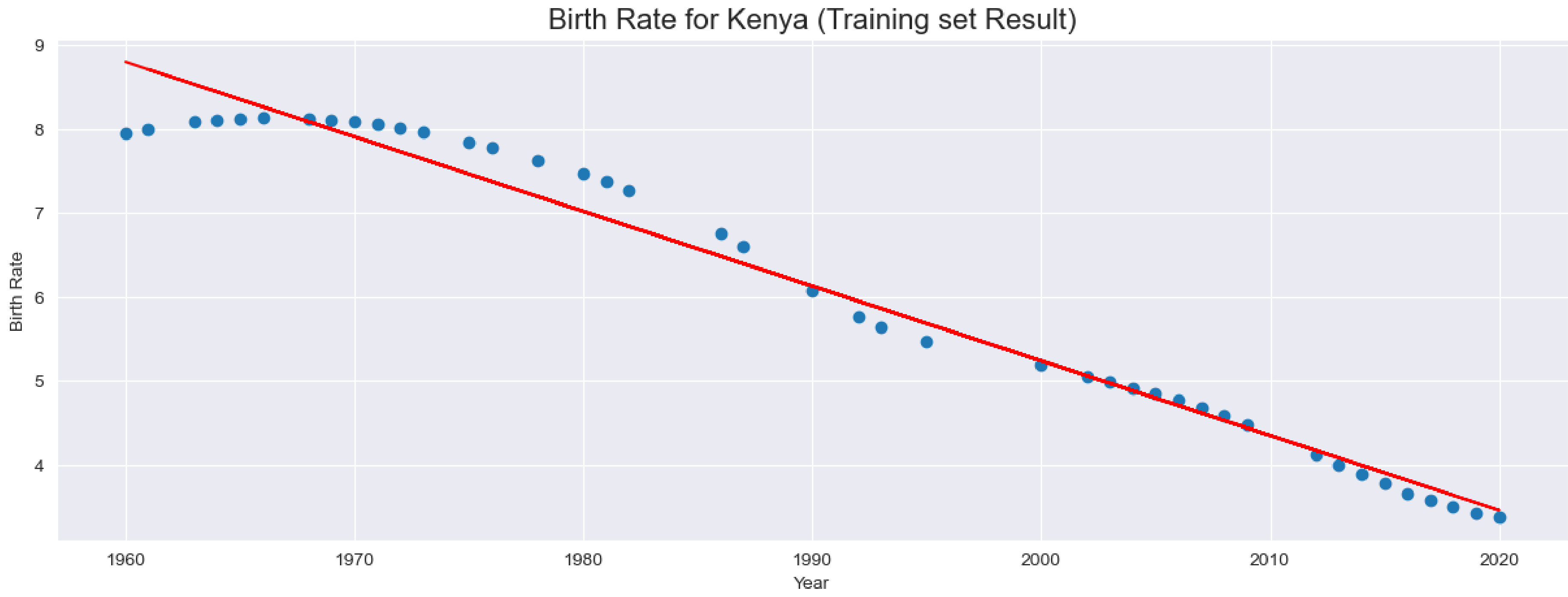
South Africa

In [60]:
```
LinerRegression(df['South Africa'], 'South Africa', 'y', 124)
```


```
Coefficient of determination:0.9602858261594986 or Accuracy:96.03%
Intercept: 148.70374974428293
slope: [-0.0726839]


'brith rate prediction for South Africa'
prediction for 2023 birth rate: [1.66421932]
prediction for 2024 birth rate: [1.59153542]
prediction for 2030 birth rate: [1.15543202]
prediction for 2080 birth rate: [-2.478763]
```

Kenya

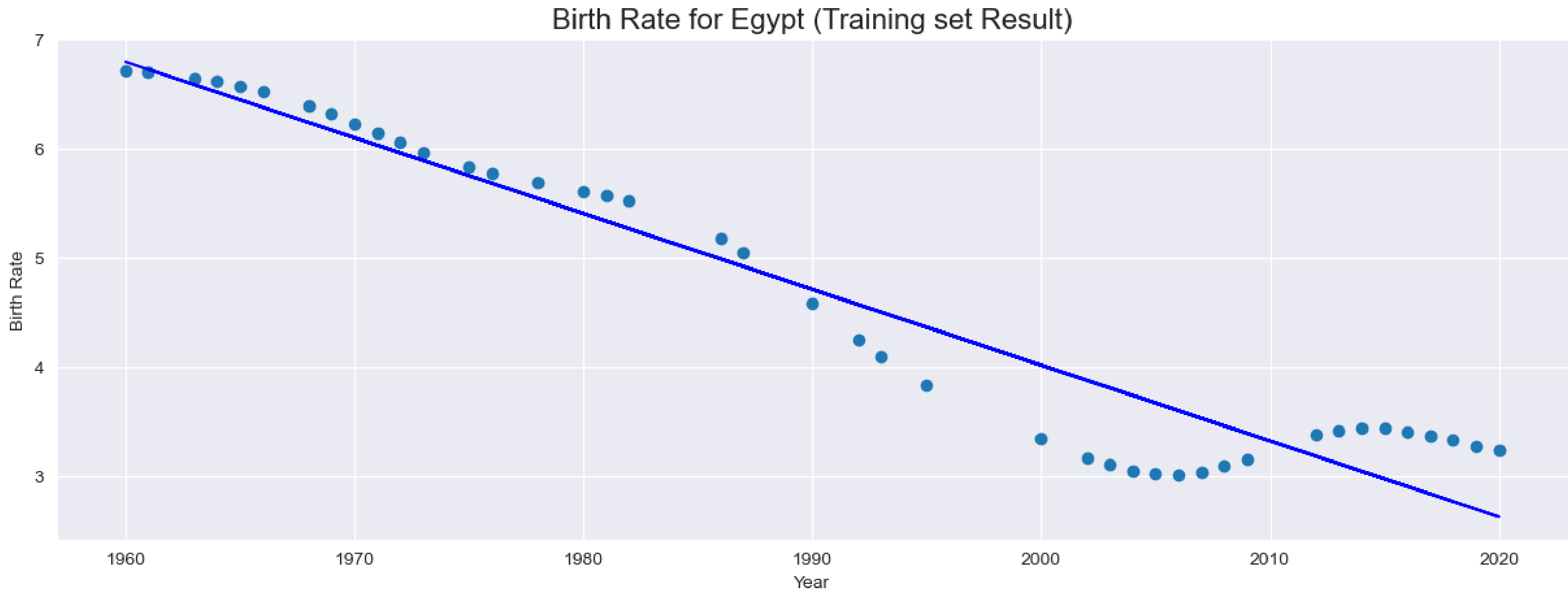
In [61]:
```
LinerRegression(df['Kenya'], 'Kenya', 'r', 124)
```




```
Coefficient of determination:0.9737378632527802 or Accuracy:97.37%
Intercept: 183.2359175886372
slope: [-0.08899992]


'brith rate prediction for Kenya'
prediction for 2023 birth rate: [3.18907393]
prediction for 2024 birth rate: [3.10007401]
prediction for 2030 birth rate: [2.56607447]
prediction for 2080 birth rate: [-1.88392166]
```

Egypt

In [62]:
```
LinerRegression(df['Egypt'], 'Egypt', 'b', 124)
```




```
Coefficient of determination:0.9248272879311439 or Accuracy:92.48%
Intercept: 143.0027557890557
slope: [-0.06949232]


'brith rate prediction for Egypt'
prediction for 2023 birth rate: [2.41978446]
prediction for 2024 birth rate: [2.35029213]
prediction for 2030 birth rate: [1.93333819]
prediction for 2080 birth rate: [-1.54127801]
```

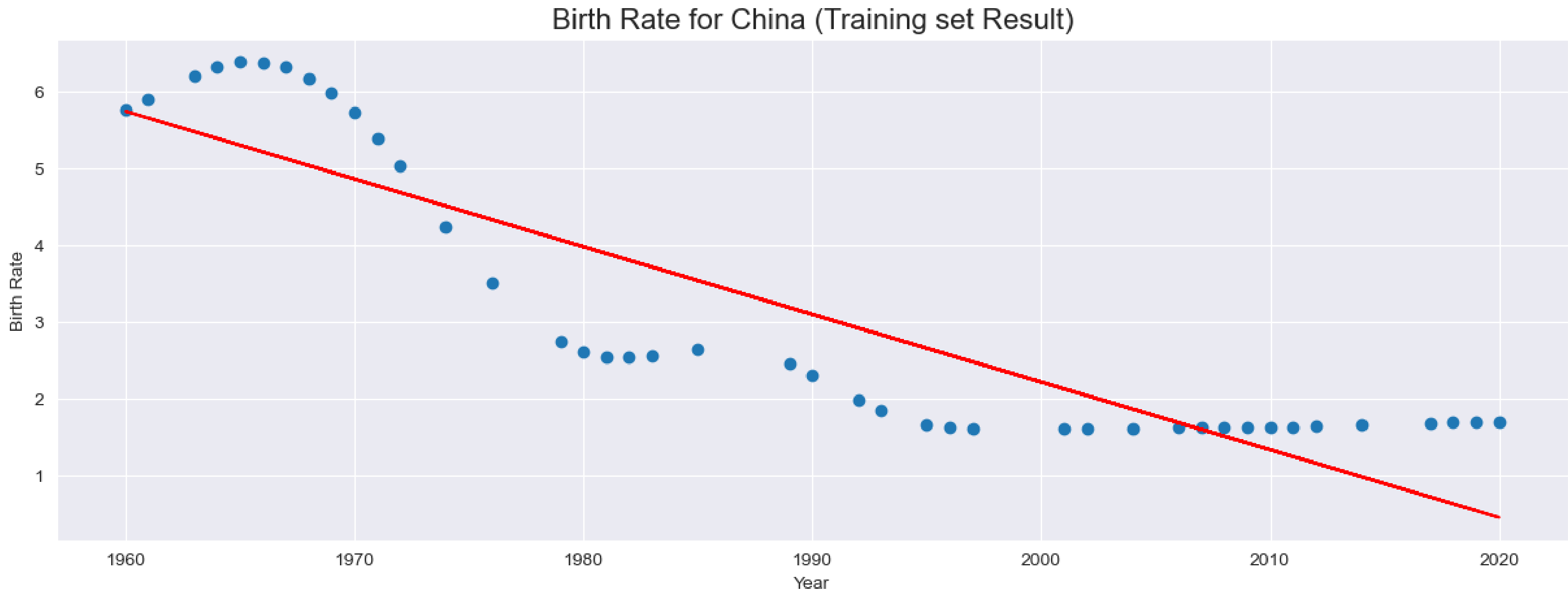
India

In [63]:
```
LinerRegression(df['India'], 'India', 'g', 44)
```


```
Coefficient of determination:0.9949293496865804 or Accuracy:99.49%
Intercept: 144.97117302052789
slope: [-0.07080156]


'brith rate prediction for India'
prediction for 2023 birth rate: [1.73960899]
prediction for 2024 birth rate: [1.66880743]
prediction for 2030 birth rate: [1.24399804]
prediction for 2080 birth rate: [-2.29608016]
```

China

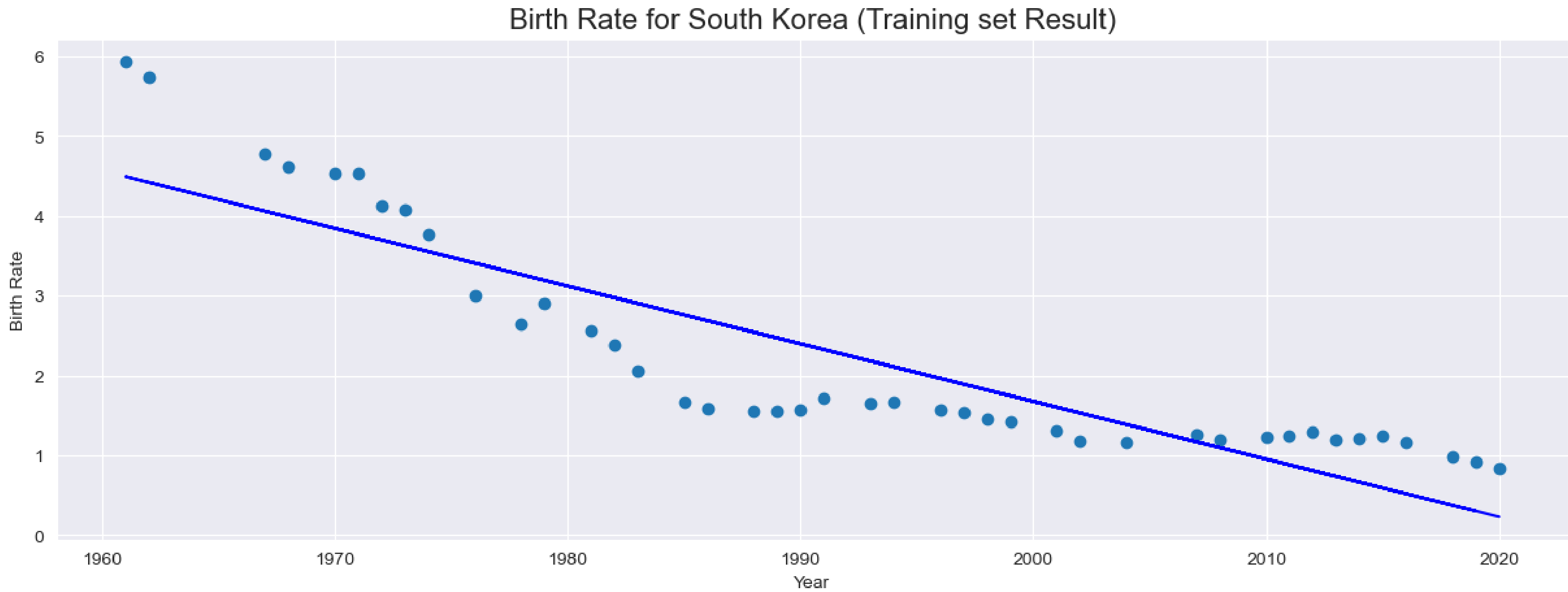
In [64]:
```
LinerRegression(df['China'], 'China', 'r', 123)
```




```
Coefficient of determination:0.7865628028102375 or Accuracy:78.66%
Intercept: 178.74416352343664
slope: [-0.08826413]


'brith rate prediction for China'
prediction for 2023 birth rate: [0.18583185]
prediction for 2024 birth rate: [0.09756772]
prediction for 2030 birth rate: [-0.43201705]
prediction for 2080 birth rate: [-4.84522347]
```

South Korea

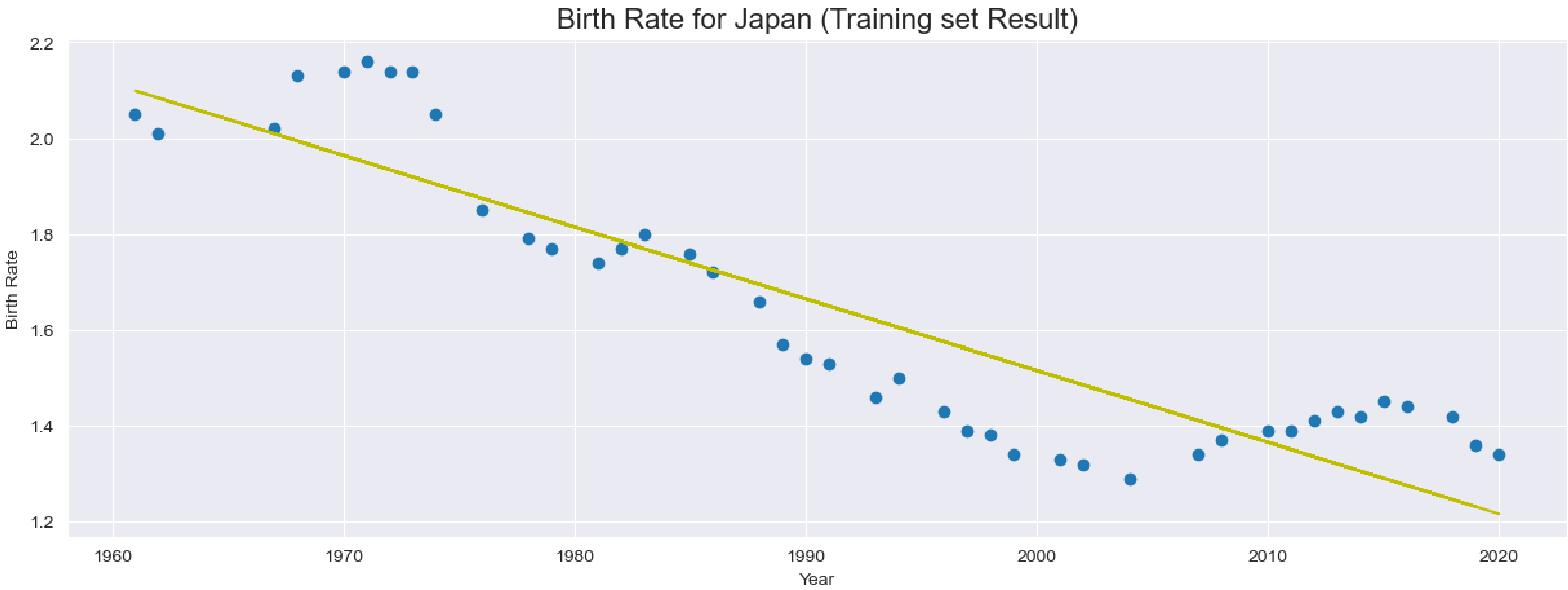
In [65]:
```
LinerRegression(df['South Korea'], 'South Korea', 'b', 44)
```




```
Coefficient of determination:0.7853388999990446 or Accuracy:78.53%
Intercept: 146.16848959146725
slope: [-0.07224632]


'brith rate prediction for South Korea'
prediction for 2023 birth rate: [0.01417868]
prediction for 2024 birth rate: [-0.05806764]
prediction for 2030 birth rate: [-0.49154558]
prediction for 2080 birth rate: [-4.10386171]
```

Japan

In [66]:
```
LinerRegression(df['Japan'], 'Japan', 'y', 44)
```




```
Coefficient of determination:0.8082703086206068 or Accuracy:80.83%
Intercept: 31.450761884930618
slope: [-0.0149677]


'brith rate prediction for Japan'
prediction for 2023 birth rate: [1.17111356]
prediction for 2024 birth rate: [1.15614586]
prediction for 2030 birth rate: [1.06633969]
prediction for 2080 birth rate: [0.31795491]
```