Essential Libraries

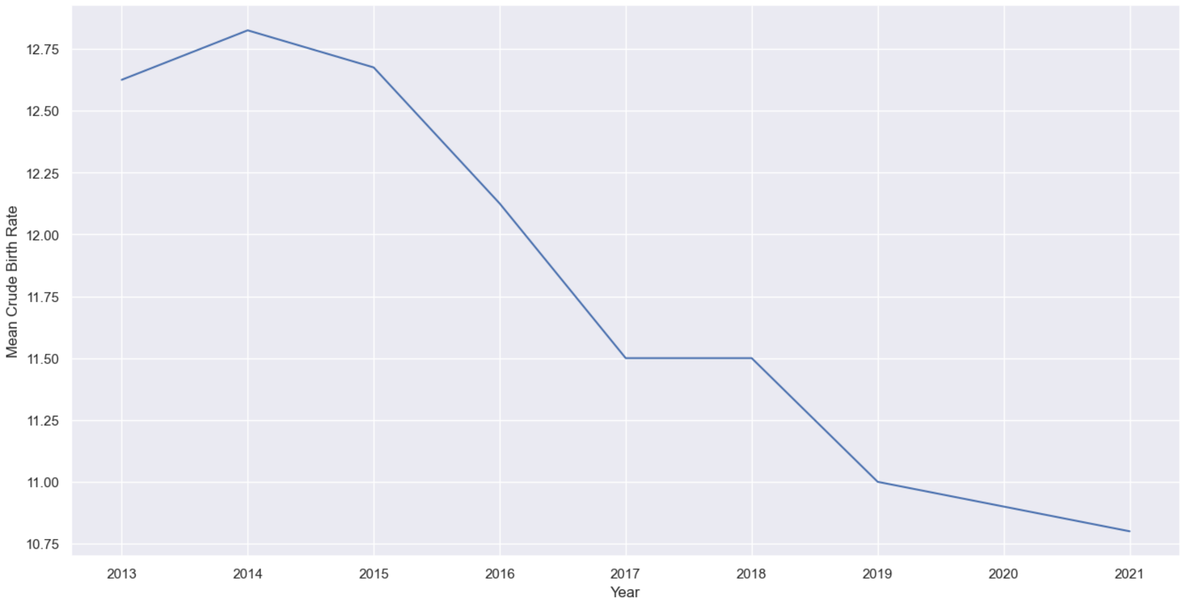
In [21]:
```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt # we only need pyplot
sb.set()
```

In [22]:
```
birth_death_rate_data = pd.read_csv('crude-birth-death-natural-increase-rates-by-e
graduate_salary_data = pd.read_csv('graduate-employment-survey-ntu-nus-sit-smu-sus
```

Singapore Birth Rate over the years

In [23]:
```
birth_rate = pd.DataFrame(birth_death_rate_data[['crude_birth_rate','year']])
birth_rate_above_2012 = birth_rate[birth_rate['year'] >= 2013]

mean_birth_rate_per_year = {}
for x in birth_rate_above_2012['year']:
    mean_birth_rate = birth_rate_above_2012.query('year == ' + str(x))['crude_birt
    mean_birth_rate_per_year[x] = mean_birth_rate

df_mean_birth_rate_per_year = pd.DataFrame(mean_birth_rate_per_year.items(), colum

f = plt.figure(figsize=(16, 8))
sb.lineplot(data = df_mean_birth_rate_per_year, x = 'Year', y = 'Mean Crude Birth 
```

Out[23]: <AxesSubplot: xlabel='Year', ylabel='Mean Crude Birth Rate'>



Fresh Graduate Salary over the Years

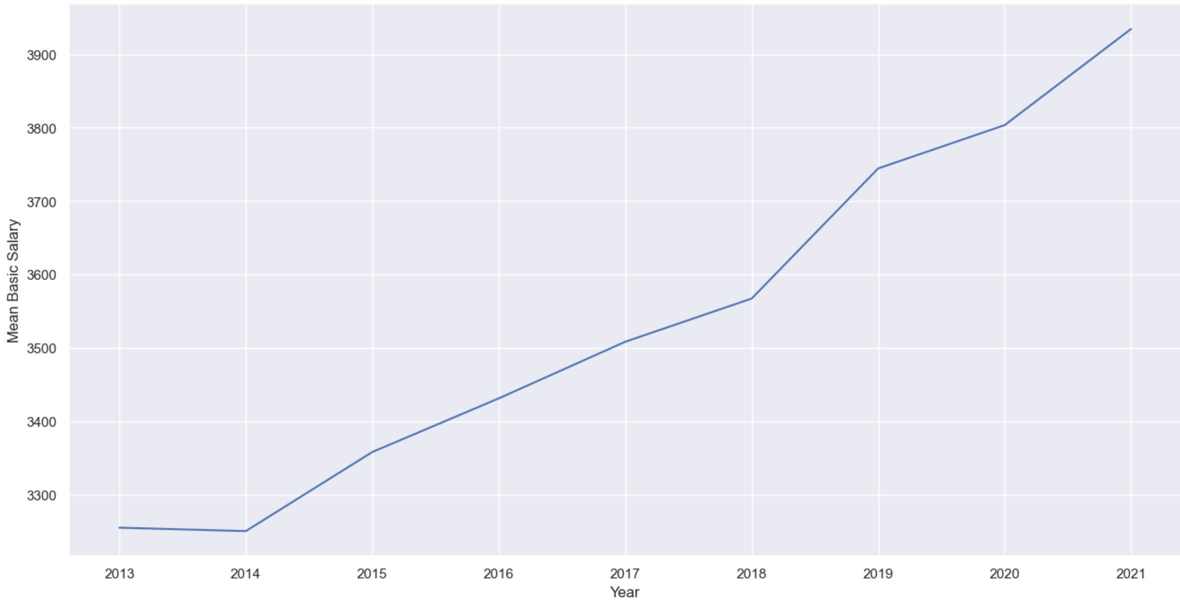
In [24]:
```
basic_salary_rate = pd.DataFrame(graduate_salary_data[['year','basic_monthly_mean'
basic_salary_rate = basic_salary_rate[basic_salary_rate["basic_monthly_mean"].str.

mean_basic_salary_per_year = {}
for x in basic_salary_rate['year']:
    mean_basic_salary_rate = basic_salary_rate.query('year == ' + str(x))['basic_m
    mean_basic_salary_per_year[x] = mean_basic_salary_rate

print(mean_basic_salary_per_year)
```

```
df_mean_basic_salary_per_year = pd.DataFrame(mean_basic_salary_per_year.items(), c

f = plt.figure(figsize=(16, 8))
sb.lineplot(data = df_mean_basic_salary_per_year, x = 'Year', y = 'Mean Basic Sala
```

```
{2013: 3254.883116883117, 2014: 3250.233009708738, 2015: 3358.0091743119265, 2016:
3431.119266055046, 2017: 3508.278260869565, 2018: 3567.198347107438, 2019: 3744.5
2, 2020: 3803.517985611511, 2021: 3934.6716417910447}
```

Out[24]:

```
<AxesSubplot: xlabel='Year', ylabel='Mean Basic Salary'>
```



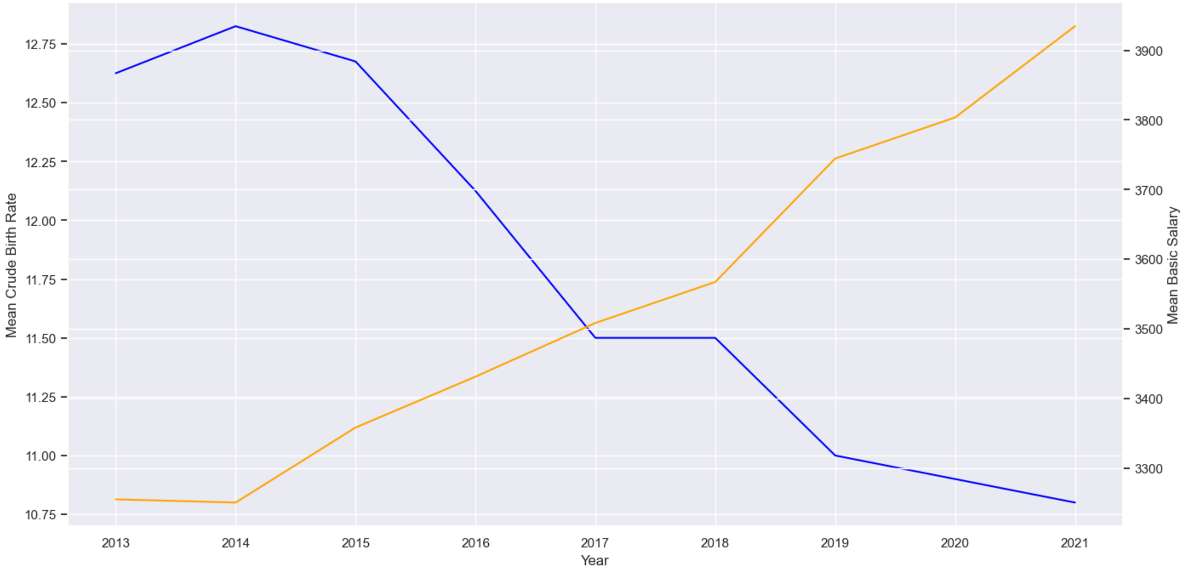
Displaying both Birth Rates and Graduate Salary from 2013 to 2021

In [25]:

```
fig=plt.figure(figsize=(16,8))
ax1 = fig.add_subplot(111)
ax2 = ax1.twinx()
jointDF_birthRate_basicSalary = pd.merge(df_mean_birth_rate_per_year,df_mean_basic_

# changes here
sb.lineplot(x = 'Year', y = 'Mean Crude Birth Rate',data=jointDF_birthRate_basicSa
sb.lineplot(x = 'Year', y = 'Mean Basic Salary' ,data=jointDF_birthRate_basicSalary

# and here
plt.xticks(rotation=60)
```

Out[25]:

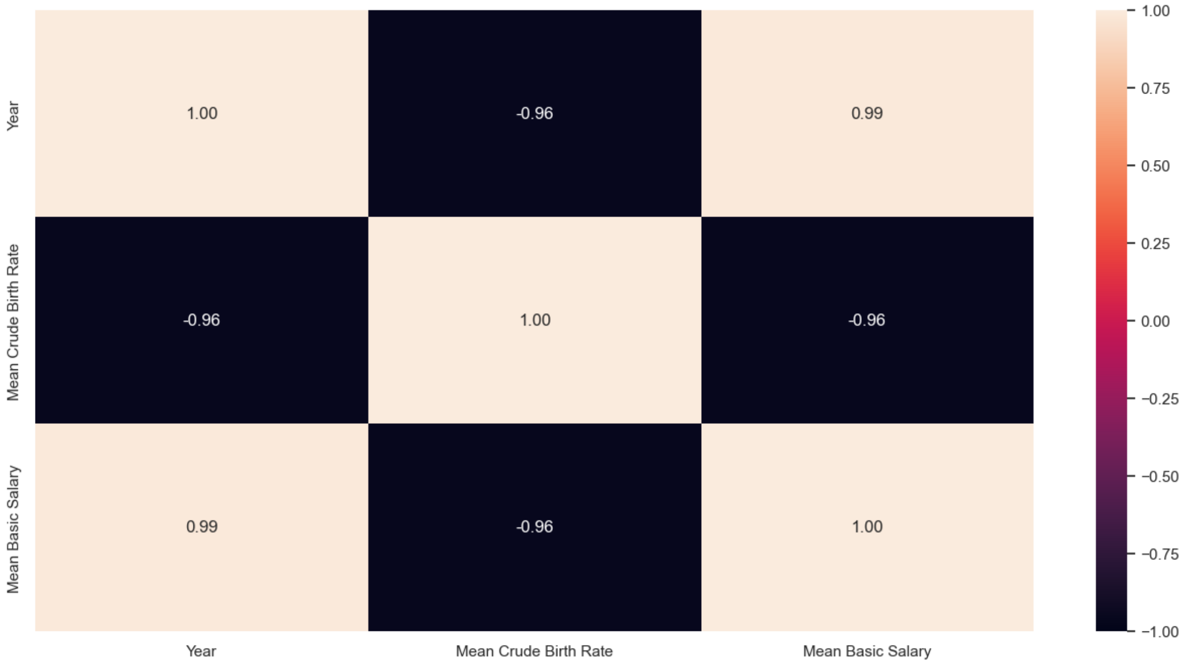
```
(array([2012., 2013., 2014., 2015., 2016., 2017., 2018., 2019., 2020.,
        2021., 2022.]),
 [Text(2012.0, 0, '2012'),
  Text(2013.0, 0, '2013'),
  Text(2014.0, 0, '2014'),
  Text(2015.0, 0, '2015'),
  Text(2016.0, 0, '2016'),
  Text(2017.0, 0, '2017'),
  Text(2018.0, 0, '2018'),
  Text(2019.0, 0, '2019'),
  Text(2020.0, 0, '2020'),
  Text(2021.0, 0, '2021'),
  Text(2022.0, 0, '2022')])
```



Correlation Between Graduate Salary vs Birth Rate

In [26]:
```
fig=plt.figure(figsize=(16,8))
sb.heatmap(jointDF_birthRate_basicSalary.corr(), vmin = -1, vmax = 1, annot = True
```

Out[26]: <AxesSubplot: >



In [27]:
```
jointDF_birthRate_basicSalary.corr()
```

Out[27]:

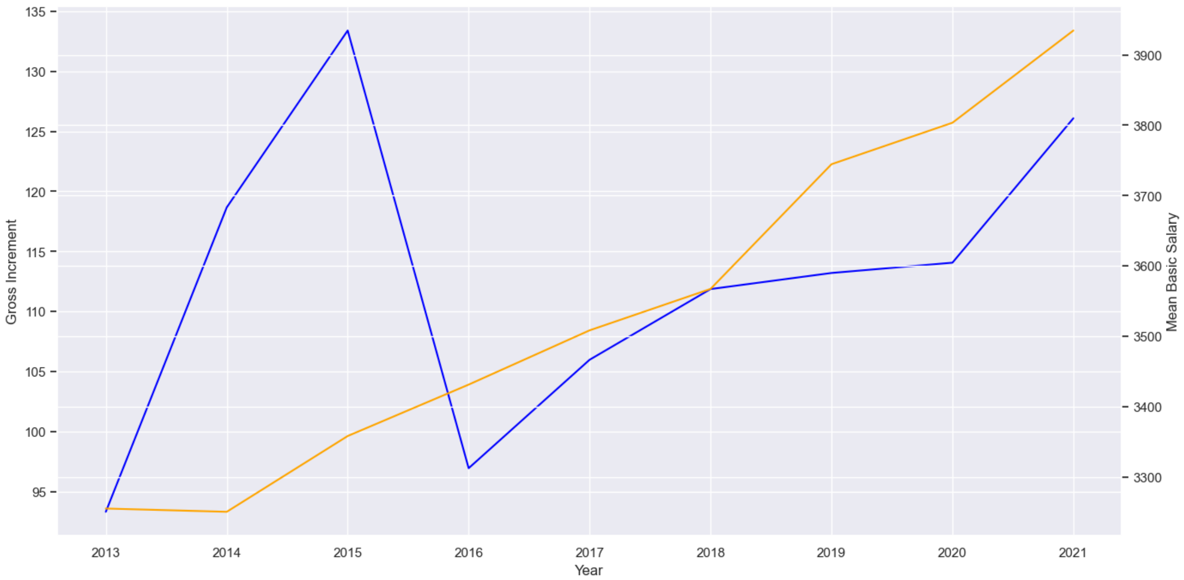
	Year	Mean Crude Birth Rate	Mean Basic Salary
Year	1.000000	-0.962903	0.986401
Mean Crude Birth Rate	-0.962903	1.000000	-0.961479
Mean Basic Salary	0.986401	-0.961479	1.000000

Increment in Graduate Gross Salary

This would mean that there is more overtime pay over the years.

In [28]:
```
gross_salary_rate = pd.DataFrame(graduate_salary_data[['year','gross_monthly_mean'
gross_salary_rate = gross_salary_rate[gross_salary_rate["gross_monthly_mean"].str.
mean_gross_salary_per_year = {}
for x in gross_salary_rate['year']:
     mean_gross_salary_rate = gross_salary_rate.query('year == ' + str(x))['gross_
     mean_gross_salary_per_year[x] =  mean_gross_salary_rate
#print( mean_gross_salary_per_year)

gross_increment_salary_per_year = {}
for x in  mean_gross_salary_per_year:
    gross_increment = ( mean_gross_salary_per_year[x] - mean_basic_salary_per_year
    gross_increment_salary_per_year[x] = gross_increment
#print( gross_increment_salary_per_year)

df_gross_increment_salary_per_year = pd.DataFrame(gross_increment_salary_per_year.

fig=plt.figure(figsize=(16,8))
ax1 = fig.add_subplot(111)
ax2 = ax1.twinx()
jointDF_grossIncrement_meanSalary = pd.merge(df_gross_increment_salary_per_year,df_

# changes here
sb.lineplot(x = 'Year', y = 'Gross Increment',data=jointDF_grossIncrement_meanSala
sb.lineplot(x = 'Year', y = 'Mean Basic Salary' ,data=jointDF_grossIncrement_meanS

# and here
plt.xticks(rotation=60)
```

Out[28]:
```
(array([2012., 2013., 2014., 2015., 2016., 2017., 2018., 2019., 2020.,
        2021., 2022.]),
 [Text(2012.0, 0, '2012'),
  Text(2013.0, 0, '2013'),
  Text(2014.0, 0, '2014'),
  Text(2015.0, 0, '2015'),
  Text(2016.0, 0, '2016'),
  Text(2017.0, 0, '2017'),
  Text(2018.0, 0, '2018'),
  Text(2019.0, 0, '2019'),
  Text(2020.0, 0, '2020'),
  Text(2021.0, 0, '2021'),
  Text(2022.0, 0, '2022')])
```



In [29]:
```
jointDF_grossIncrement_birthRate = pd.merge(df_mean_birth_rate_per_year, df_gross_
jointDF_grossIncrement_birthRate.corr()
```

Out[29]:

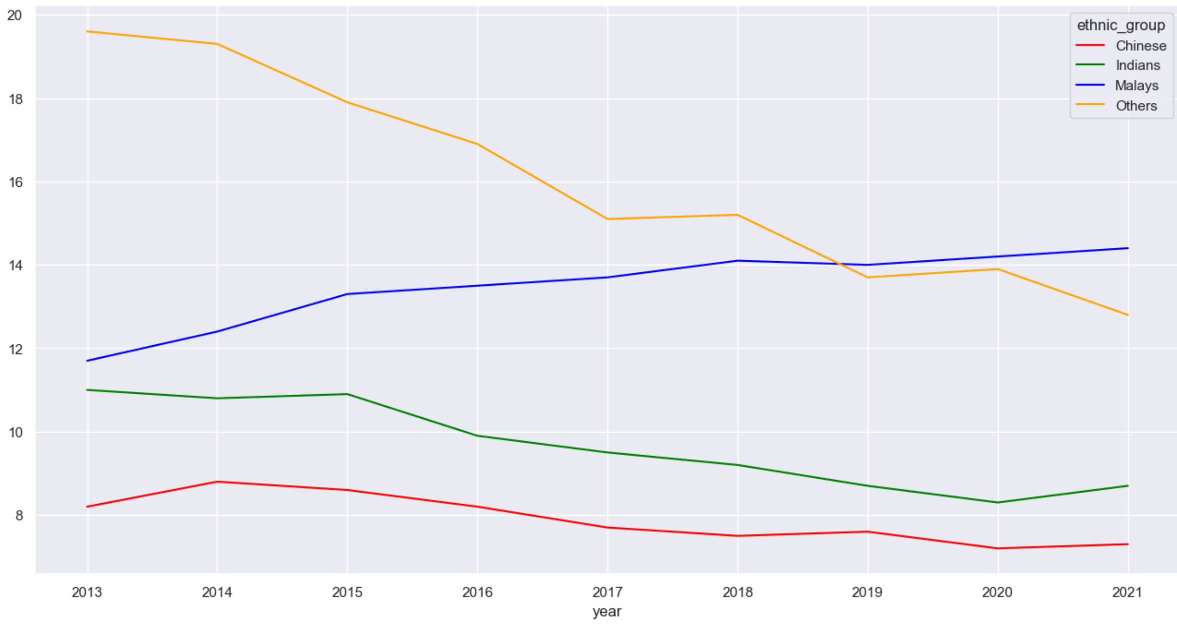
	Year	Mean Crude Birth Rate	Gross Increment
Year	1.000000	-0.962903	0.326838
Mean Crude Birth Rate	-0.962903	1.000000	-0.116502
Gross Increment	0.326838	-0.116502	1.000000

Birth Rate by Ethnic Groups from 2013 to 2021

In [30]:
```
race_birth_rate = pd.DataFrame(birth_death_rate_data[['year', 'ethnic_group', 'cru
race_birth_rate_above_2012 = race_birth_rate[race_birth_rate['year'] >= 2013]

df_race_birth_rate_above_2012 = race_birth_rate_above_2012.pivot(index='year', col
print(df_race_birth_rate_above_2012)
df_race_birth_rate_above_2012.plot(color=['red', 'green', 'blue', 'orange'], figsi
```

```
ethnic_group  Chinese  Indians  Malays  Others
year
2013              8.2     11.0    11.7    19.6
2014              8.8     10.8    12.4    19.3
2015              8.6     10.9    13.3    17.9
2016              8.2      9.9    13.5    16.9
2017              7.7      9.5    13.7    15.1
2018              7.5      9.2    14.1    15.2
2019              7.6      8.7    14.0    13.7
2020              7.2      8.3    14.2    13.9
2021              7.3      8.7    14.4    12.8
```

Out[30]: <AxesSubplot: xlabel='year'>



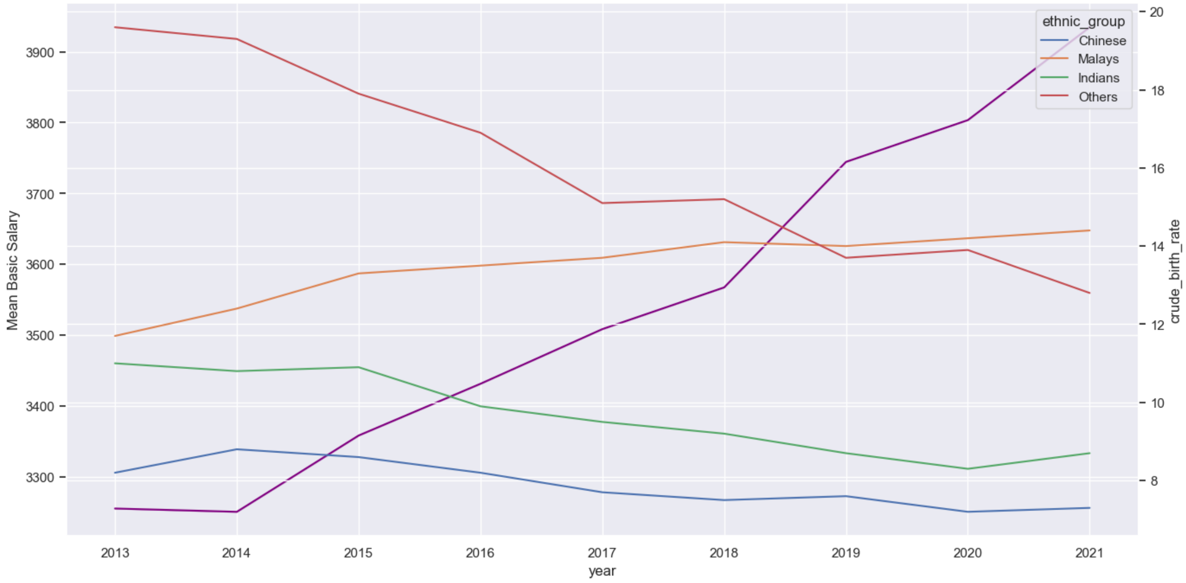
Birth Rate by Ethnic Groups vs Graduate Salary

In [31]:
```
df2_mean_basic_salary_per_year = pd.DataFrame(mean_basic_salary_per_year.items(), 
# changes here
fig=plt.figure(figsize=(16,8))
ax1 = fig.add_subplot(111)
ax2 = ax1.twinx()

jointDF_birthRate_meanSalary = pd.merge(race_birth_rate_above_2012,df2_mean_basic_
```

```
sb.lineplot(x = 'year', y = 'Mean Basic Salary' ,data=jointDF_birthRate_meanSalary
sb.lineplot(x = 'year', y = 'crude_birth_rate',data=jointDF_birthRate_meanSalary,a
```

Out[31]:

```
<AxesSubplot: xlabel='year', ylabel='crude_birth_rate'>
```



Linear Regression of Birth Rate

In [61]:

```
# Import essential models and functions from sklearn
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

basicSalary_train, basicSalary_test, birthRate_train, birthRate_test = train_test_

# Linear Regression using Train Data
linreg = LinearRegression()         # create the linear regression object
linreg.fit(basicSalary_train, birthRate_train)        # train the linear regressio

# Coefficients of the Linear Regression line
print('Intercept of Regression \t: b = ', linreg.intercept_)
print('Coefficients of Regression \t: a = ', linreg.coef_)
print()

# Predict Birth Rate values corresponding to Basic Salary
birthRate_train_pred = linreg.predict(basicSalary_train)

# Check the Goodness of Fit (on Train Data)
print("Goodness of Fit of Model")
print("Explained Variance (R^2) \t:", linreg.score(basicSalary_train, birthRate_tr
print("Mean Squared Error (MSE) \t:", mean_squared_error(birthRate_train, birthRat
print()
```

```
Intercept of Regression         : b =  [ 1491.08109209  1705.67765558 -1193.853524
83  1458.63086798]
Coefficients of Regression      : a =  [[-0.74459418  0.00532433]
 [-0.84931093  0.00485459]
 [ 0.60508087 -0.00370267]
 [-0.71137121 -0.00214303]]

Goodness of Fit of Model
Explained Variance (R^2)        : 0.9492547231225457
Mean Squared Error (MSE)        : 0.06663792001040561
```