List of Commands & Function:

PANDAS IMPORT

import pandas as pd

stats = pd.read\_csv('C:\\Users\\deep.ghosh\\Desktop\\Jupyter Programming Practice\\P4-Demographic-Data.csv')

# Method 2 :: Change Working directory

# First get the Current working Directory

import os

os.getcwd()

**Functions**:

stats.head() #Returns top 5 rows

stats.head(10) #Returns top 10 rows

stats.tail() # Returns bottom 5 rows

stats.info() #Information about the columns - Like str() function in R

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 195 entries, 0 to 194

Data columns (total 5 columns):

Country Name 195 non-null object

Country Code 195 non-null object

Birth rate 195 non-null float64

Internet users 195 non-null float64

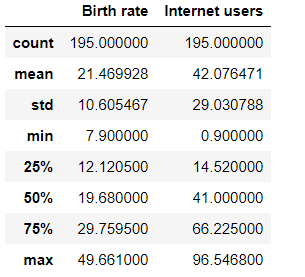
Income Group 195 non-null object

dtypes: float64(2), object(3)

memory usage: 7.7+ KB

#Get Stats on the columns - Like summary() function in R

stats.describe()



len(stats) # Length of STATS 195 will be returned

# See the columns

stats.columns

Index(['Country Name', 'Country Code', 'Birth rate', 'Internet users',

'Income Group'],

dtype='object')

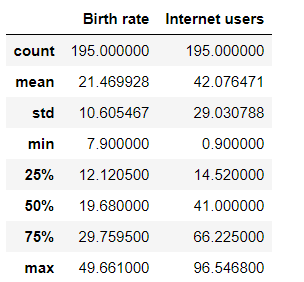
*# Number of Columns*

len(stats.columns)

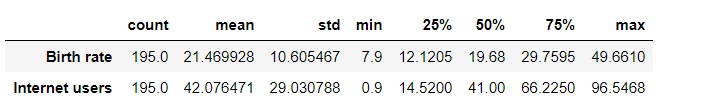
5

# Get the stats on the columns

stats.describe() # Like the summary function in R



stats.describe().transpose()



**# Renaming Columns of Dataframe**

stats.columns

Index(['Country Name', 'Country Code', 'Birth rate', 'Internet users',

'Income Group'],

dtype='object')

**# Removing the GAPS in between two words**

stats.columns = ['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers', 'IncomeGroup']

stats.columns

​

Index(['CountryName', 'CountryCode', 'BirthRate', 'InternetUsers',

'IncomeGroup'],dtype='object')

**After renaming the columns**

stats.head()

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** |
| --- | --- | --- | --- | --- | --- |
| **0** | Aruba | ABW | 10.244 | 78.9 | High income |
| **1** | Afghanistan | AFG | 35.253 | 5.9 | Low income |
| **2** | Angola | AGO | 45.985 | 19.1 | Upper middle income |
| **3** | Albania | ALB | 12.877 | 57.2 | Upper middle income |
| **4** | United Arab Emirates | ARE | 11.044 | 88.0 | High income |

**# Subsetting dataframes in Pandas**

**# 3 Parts**

**# How to Sub Set by Rows**

**# How to Subset by Columns**

**# How to Combine them**

#Printing whole table

stats[:]

*# prints the last 10 Columns*

stats[185:]

Out[50]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** |
| --- | --- | --- | --- | --- | --- |
| **185** | Virgin Islands (U.S.) | VIR | 10.700 | 45.3 | High income |
| **186** | Vietnam | VNM | 15.537 | 43.9 | Lower middle income |
| **187** | Vanuatu | VUT | 26.739 | 11.3 | Lower middle income |
| **188** | West Bank and Gaza | PSE | 30.394 | 46.6 | Lower middle income |
| **189** | Samoa | WSM | 26.172 | 15.3 | Lower middle income |
| **190** | Yemen, Rep. | YEM | 32.947 | 20.0 | Lower middle income |
| **191** | South Africa | ZAF | 20.850 | 46.5 | Upper middle income |
| **192** | Congo, Dem. Rep. | COD | 42.394 | 2.2 | Low income |
| **193** | Zambia | ZMB | 40.471 | 15.4 | Lower middle income |
| **194** | Zimbabwe | ZWE | 35.715 | 18.5 | Low income |

stats[:10] *# Prints first 10 columns*

Out[53]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** |
| --- | --- | --- | --- | --- | --- |
| **0** | Aruba | ABW | 10.244 | 78.9000 | High income |
| **1** | Afghanistan | AFG | 35.253 | 5.9000 | Low income |
| **2** | Angola | AGO | 45.985 | 19.1000 | Upper middle income |
| **3** | Albania | ALB | 12.877 | 57.2000 | Upper middle income |
| **4** | United Arab Emirates | ARE | 11.044 | 88.0000 | High income |
| **5** | Argentina | ARG | 17.716 | 59.9000 | High income |
| **6** | Armenia | ARM | 13.308 | 41.9000 | Lower middle income |
| **7** | Antigua and Barbuda | ATG | 16.447 | 63.4000 | High income |
| **8** | Australia | AUS | 13.200 | 83.0000 | High income |
| **9** | Austria | AUT | 9.400 | 80.6188 | High income |

*# Exercise*

*# Reverse the Dataframe - Starts from ZAmbia & end to Aruba*

stats[: : **-**1]

Out[55]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** |
| --- | --- | --- | --- | --- | --- |
| **194** | Zimbabwe | ZWE | 35.715 | 18.50000 | Low income |
| **193** | Zambia | ZMB | 40.471 | 15.40000 | Lower middle income |
| **192** | Congo, Dem. Rep. | COD | 42.394 | 2.20000 | Low income |
| **191** | South Africa | ZAF | 20.850 | 46.50000 | Upper middle income |
| **190** | Yemen, Rep. | YEM | 32.947 | 20.00000 | Lower middle income |

*# Quick Exercise*

*# Get only every 20th row*

​

stats[::20]

Out[56]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** |
| --- | --- | --- | --- | --- | --- |
| **0** | Aruba | ABW | 10.244 | 78.9000 | High income |
| **20** | Belarus | BLR | 12.500 | 54.1700 | Upper middle income |
| **40** | Costa Rica | CRI | 15.022 | 45.9600 | Upper middle income |
| **60** | Gabon | GAB | 30.555 | 9.2000 | Upper middle income |
| **80** | India | IND | 20.291 | 15.1000 | Lower middle income |
| **100** | Libya | LBY | 21.425 | 16.5000 | Upper middle income |
| **120** | Mozambique | MOZ | 39.705 | 5.4000 | Low income |
| **140** | Poland | POL | 9.600 | 62.8492 | High income |
| **160** | Suriname | SUR | 18.455 | 37.4000 | Upper middle income |
| **180** | Uruguay | URY | 14.374 | 57.6900 | High income |

# Part 2 . Columns

stats.columns

stats['CountryName']

Out[60]:

0 Aruba

1 Afghanistan

2 Angola

3 Albania

4 United Arab Emirate

**Print Specific Columns**

stats[['CountryName', 'BirthRate']].head()

Out[65]:

|  | **CountryName** | **BirthRate** |
| --- | --- | --- |
| **0** | Aruba | 10.244 |
| **1** | Afghanistan | 35.253 |
| **2** | Angola | 45.985 |
| **3** | Albania | 12.877 |
| **4** | United Arab Emirates | 11.044 |

*# Quick Access - Requires the name to be One Word*

stats.CountryName

Out[72]:

0 Aruba

1 Afghanistan

2 Angola

3 Albania

4 United Arab Emirates

5 Argentina

6 Armenia

7 Antigua and Barbuda

8 Australia

**Returns columns with specific names from 1 to 10**

stats[['CountryName', 'BirthRate', 'InternetUsers']][1:10]

|  | **CountryName** | **BirthRate** | **InternetUsers** |
| --- | --- | --- | --- |
| **1** | Afghanistan | 35.253 | 5.9000 |
| **2** | Angola | 45.985 | 19.1000 |
| **3** | Albania | 12.877 | 57.2000 |
| **4** | United Arab Emirates | 11.044 | 88.0000 |
| **5** | Argentina | 17.716 | 59.9000 |
| **6** | Armenia | 13.308 | 41.9000 |
| **7** | Antigua and Barbuda | 16.447 | 63.4000 |
| **8** | Australia | 13.200 | 83.0000 |
| **9** | Austria | 9.400 | 80.6188 |

**# Its working on SUBSETS. First set stats[['CountryName', 'BirthRate', 'InternetUsers']] will give us some outputs**

**# the df1 [1:10] is working on the set of stats[['CountryName', 'BirthRate', 'InternetUsers']]**

**df1 = stats[['CountryName', 'BirthRate', 'InternetUsers']]**

**df1**

**#df1 [1:10]**

|  | **CountryName** | **BirthRate** | **InternetUsers** |
| --- | --- | --- | --- |
| **0** | Aruba | 10.244 | 78.90000 |
| **1** | Afghanistan | 35.253 | 5.90000 |
| **2** | Angola | 45.985 | 19.10000 |
| **3** | Albania | 12.877 | 57.20000 |
| **4** | United Arab Emirates | 11.044 | 88.00000 |
| **5** | Argentina | 17.716 | 59.90000 |
| **6** | Armenia | 13.308 | 41.90000 |
| **7** | Antigua and Barbuda | 16.447 | 63.40000 |

*# Basic Operations with Data Frame*

*# Mathematical Operations*

Multiplication **=** stats.BirthRate **\*** stats.InternetUsers

Multiplication.head()

​

Out[85]:

0 808.2516

1 207.9927

2 878.3135

3 736.5644

4 971.8720

dtype: float64

**# Adding Column to Data Frame**

stats['BirthRate X InternetUsers'] **=** stats.BirthRate **\*** stats.InternetUsers

stats.head()

Out[53]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** | **BirthRate X InternetUsers** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | Aruba | ABW | 10.244 | 78.9 | High income | 808.2516 |
| **1** | Afghanistan | AFG | 35.253 | 5.9 | Low income | 207.9927 |
| **2** | Angola | AGO | 45.985 | 19.1 | Upper middle income | 878.3135 |
| **3** | Albania | ALB | 12.877 | 57.2 | Upper middle income | 736.5644 |
| **4** | United Arab Emirates | ARE | 11.044 | 88.0 | High income | 971.8720 |

**#Removing a Column**

**stats.drop(columns=["BirthRate X InternetUsers"])**

*# Filtering Data Frames*

*# Filtering is all about Rows*

filter **=** stats.InternetUsers **<** 2

filter

Out[93]:

0 False

1 False

2 False

3 False

4 False

5 False

6 False

7 False

8 False

9 False

10 False

stats[filter]

Out[94]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** | **MyMultiplication** | **BirthRate X InternetUsers** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **11** | Burundi | BDI | 44.151 | 1.3 | Low income | 57.3963 | 57.3963 |
| **52** | Eritrea | ERI | 34.800 | 0.9 | Low income | 31.3200 | 31.3200 |
| **55** | Ethiopia | ETH | 32.925 | 1.9 | Low income | 62.5575 | 62.5575 |
| **64** | Guinea | GIN | 37.337 | 1.6 | Low income | 59.7392 | 59.7392 |
| **117** | Myanmar | MMR | 18.119 | 1.6 | Lower middle income | 28.9904 | 28.9904 |
| **127** | Niger | NER | 49.661 | 1.7 | Low income | 84.4237 | 84.4237 |
| **154** | Sierra Leone | SLE | 36.729 | 1.7 | Low income | 62.4393 | 62.4393 |
| **156** | Somalia | SOM | 43.891 | 1.5 | Low income | 65.8365 | 65.8365 |
| **172** | Timor-Leste | TLS | 35.755 | 1.1 | Lower middle income | 39.3305 | 39.3305 |

*# Filtering the country where BirthRate is less than 10*

filterBirthRate **=** stats.BirthRate **<** 10

stats[filterBirthRate]

Out[100]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** | **MyMultiplication** | **BirthRate X InternetUsers** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **9** | Austria | AUT | 9.400 | 80.6188 | High income | 757.81672 | 757.81672 |
| **16** | Bulgaria | BGR | 9.200 | 53.0615 | Upper middle income | 488.16580 | 488.16580 |
| **19** | Bosnia and Herzegovina | BIH | 9.062 | 57.7900 | Upper middle income | 523.69298 | 523.69298 |
| **45** | Germany | DEU | 8.500 | 84.1700 | High income | 715.44500 | 715.44500 |
| **53** | Spain | ESP | 9.100 | 71.6350 | High income | 651.87850 | 651.87850 |
| **68** | Greece | GRC | 8.500 | 59.8663 | High income | 508.86355 | 508.86355 |
| **74** | Hong Kong SAR, China | HKG | 7.900 | 74.2000 | High income | 586.18000 | 586.18000 |
| **76** | Croatia | HRV | 9.400 | 66.7476 | High income | 627.42744 | 627.42744 |
| **78** | Hungary | HUN | 9.200 | 72.6439 | High income | 668.32388 | 668.32388 |
| **86** | Italy | ITA | 8.500 | 58.4593 | High income | 496.90405 | 496.90405 |
| **89** | Japan | JPN | 8.200 | 89.7100 | High income | 735.62200 | 735.62200 |
| **95** | Korea, Rep. | KOR | 8.600 | 84.7700 | High income | 729.02200 | 729.02200 |
| **102** | Liechtenstein | LIE | 9.200 | 93.8000 | High income | 862.96000 | 862.96000 |
| **116** | Malta | MLT | 9.500 | 68.9138 | High income | 654.68110 | 654.68110 |
| **140** | Poland | POL | 9.600 | 62.8492 | High income | 603.35232 | 603.35232 |
| **142** | Portugal | PRT | 7.900 | 62.0956 | High income | 490.55524 | 490.55524 |
| **146** | Romania | ROU | 8.800 | 49.7645 | Upper middle income | 437.92760 | 437.92760 |
| **152** | Singapore | SGP | 9.300 | 81.0000 | High income | 753.30000 | 753.30000 |
| **157** | Serbia | SRB | 9.200 | 51.5000 | Upper middle income | 473.80000 | 473.80000 |

*# Exercise : Need to find countries where BirthRate > 40 & InternetUsers < 2*

filterBirthRate **=** stats.BirthRate **>** 40

filterInternetUsers **=** stats.InternetUsers **<** 2

stats[filterBirthRate **&** filterInternetUsers **<** 2]

Out[111]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** | **MyMultiplication** | **BirthRate X InternetUsers** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Aruba | ABW | 10.244 | 78.90000 | High income | 808.251600 | 808.251600 |
| **1** | Afghanistan | AFG | 35.253 | 5.90000 | Low income | 207.992700 | 207.992700 |
| **2** | Angola | AGO | 45.985 | 19.10000 | Upper middle income | 878.313500 | 878.313500 |
| **3** | Albania | ALB | 12.877 | 57.20000 | Upper middle income | 736.564400 | 736.564400 |

*# How to get unique values in each column*

stats.IncomeGroup.unique()

Out[112]:

array(['High income', 'Low income', 'Upper middle income',

'Lower middle income'], dtype=object)

*# Quick Exercise*

*# Findout everything about MALTA*

stats[stats.CountryName **==** 'Malta']

Out[114]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** | **MyMultiplication** | **BirthRate X InternetUsers** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **116** | Malta | MLT | 9.5 | 68.9138 | High income | 654.6811 | 654.6811 |

*## Accessing individual elements*

*# We need to functions to access individual elements in a Data Set*

*# .at # This is for Labels IMPORTANT : Even integers are treated as LABELs*

*# .iat # for Integer location*

stats.head()

Out[115]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** | **MyMultiplication** | **BirthRate X InternetUsers** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Aruba | ABW | 10.244 | 78.9 | High income | 808.2516 | 808.2516 |
| **1** | Afghanistan | AFG | 35.253 | 5.9 | Low income | 207.9927 | 207.9927 |
| **2** | Angola | AGO | 45.985 | 19.1 | Upper middle income | 878.3135 | 878.3135 |
| **3** | Albania | ALB | 12.877 | 57.2 | Upper middle income | 736.5644 | 736.5644 |
| **4** | United Arab Emirates | ARE | 11.044 | 88.0 | High income | 971.8720 | 971.8720 |

stats.iat[1,4]

Out[116]:

'Low income'

stats.at[0, "InternetUsers"]

Out[117]:

78.9

sub10 **=** stats[::10]

sub10

Out[119]:

|  | **CountryName** | **CountryCode** | **BirthRate** | **InternetUsers** | **IncomeGroup** | **MyMultiplication** | **BirthRate X InternetUsers** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Aruba | ABW | 10.244 | 78.900000 | High income | 808.251600 | 808.251600 |
| **10** | Azerbaijan | AZE | 18.300 | 58.700000 | Upper middle income | 1074.210000 | 1074.210000 |
| **20** | Belarus | BLR | 12.500 | 54.170000 | Upper middle income | 677.125000 | 677.125000 |
| **30** | Canada | CAN | 10.900 | 85.800000 | High income | 935.220000 | 935.220000 |
| **40** | Costa Rica | CRI | 15.022 | 45.960000 | Upper middle income | 690.411120 | 690.411120 |

sub10.iat[10,0]

Out[126]:

'Libya'

In [127]:



sub10.at[10, "CountryName"]

Out[127]:

'Azerbaijan'

**SEABORN**

# Introduction to Seaborn

# It is a powerful visualization Package for Python.

# It is built on top of MATPLOTLIB - PYPLOT

import matplotlib.pyplot as plt

import seaborn as sns

# it is written so that the Graphs get written inside the JUPYTER Notebook

%matplotlib inline

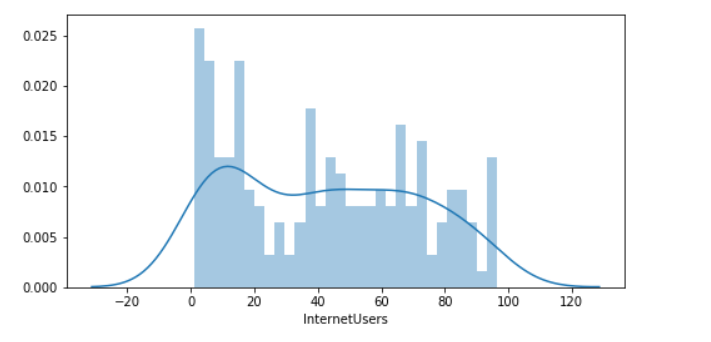
plt.rcParams['figure.figsize'] = 8,4

import warnings

warnings.filterwarnings("ignore")

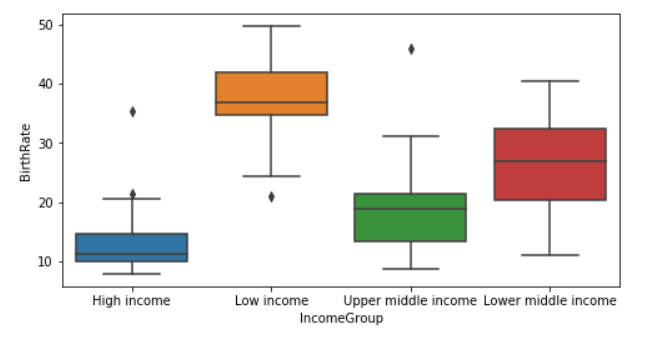
*# Distribution*

vis1 **=** sns.distplot(stats['InternetUsers'], bins **=** 30)



# Boxplots

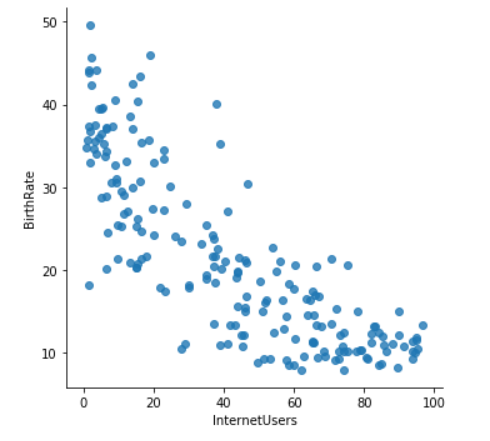
vis2 = sns.boxplot(data=stats, x="IncomeGroup", y="BirthRate")



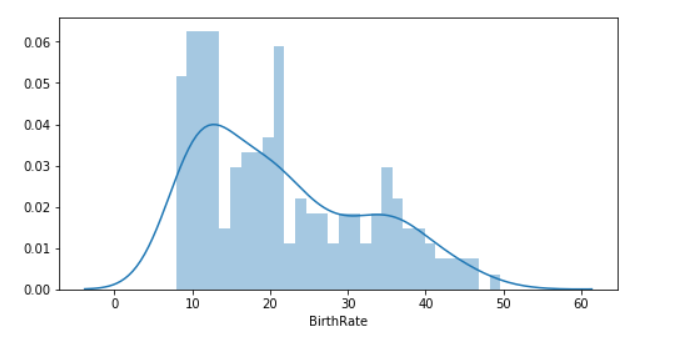
# Seaborn Gallery

# lmplot ~ LM stands for Linear Model Plot

vis3 = sns.lmplot(data = stats, x = 'InternetUsers', y = 'BirthRate', fit\_reg = False)

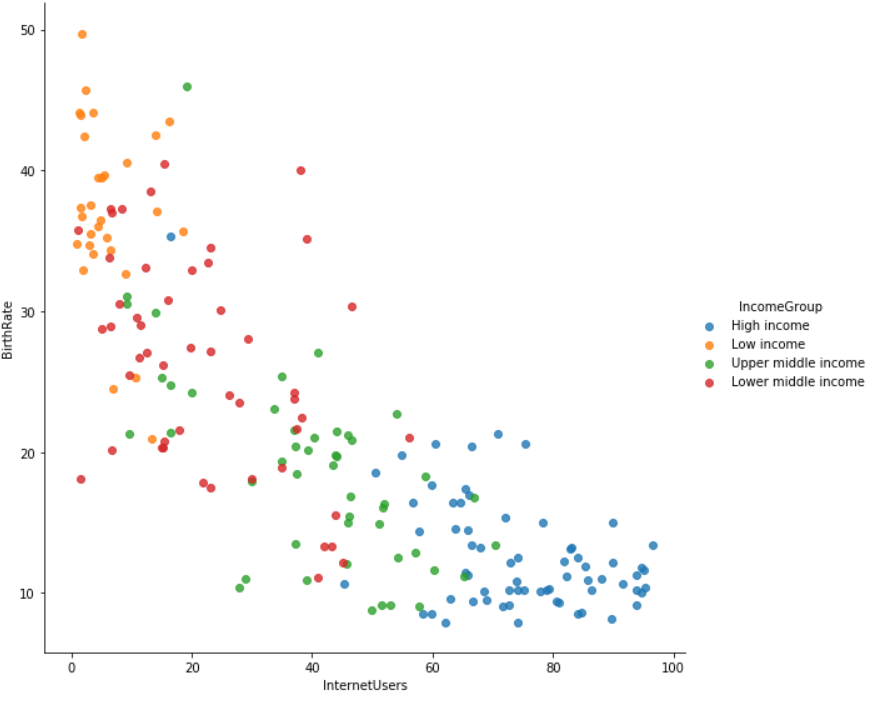


vis4 = sns.distplot(stats['BirthRate'], bins = 30)



# Adding Color on INCOMEGROUP

vis3 = sns.lmplot(data = stats, x = 'InternetUsers', y = 'BirthRate', fit\_reg = False, hue = 'IncomeGroup', size = 8)

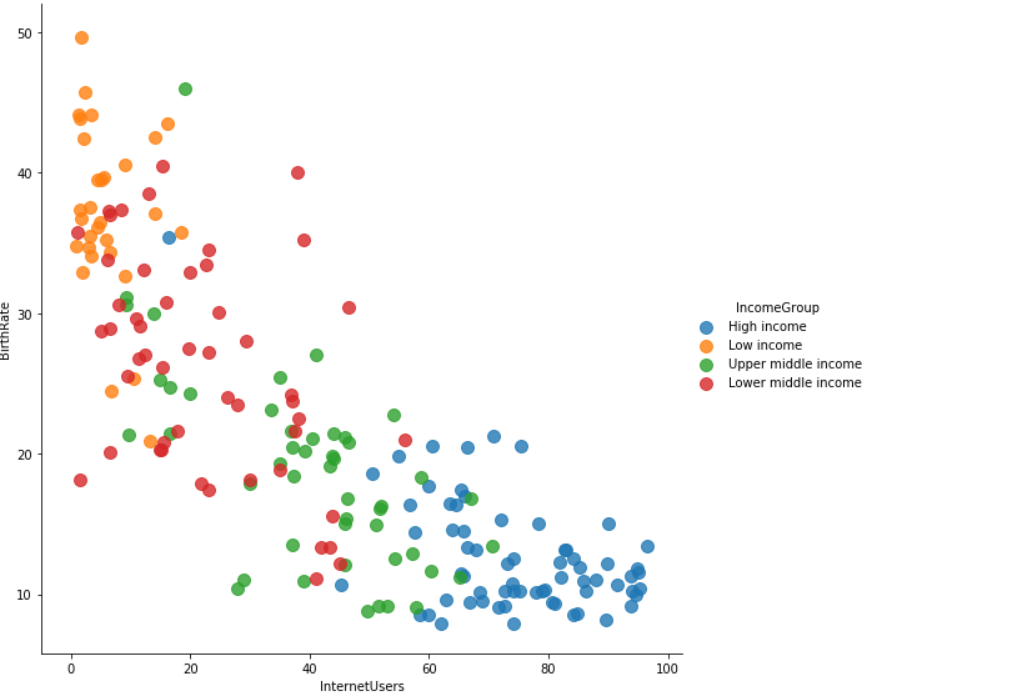


**KEYWORD ARGUMENTS IN PYTHON**

**# Marker Size : How to increase the size of Marker**

vis3 = sns.lmplot(data = stats, x = 'InternetUsers', y = 'BirthRate', fit\_reg = False, hue = 'IncomeGroup', size = 8, \

scatter\_kws= {"s":100})

****