D01_SUSHIL_Q1

August 6, 2018

1 Q1 Birthrate data of United States using numpy and pandas

In [61]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df=pd.read_csv('US_Birthrate_data.csv')    # Read csv file
df
```

Out[61]:

	year	month	day	gender	births
0	1969	1	1.0	F	4046
1	1969	1	1.0	M	4440
2	1969	1	2.0	F	4454
3	1969	1	2.0	M	4548
4	1969	1	3.0	F	4548
5	1969	1	3.0	M	4994
6	1969	1	4.0	F	4440
7	1969	1	4.0	M	4520
8	1969	1	5.0	F	4192
9	1969	1	5.0	M	4198
10	1969	1	6.0	F	4710
11	1969	1	6.0	M	4850
12	1969	1	7.0	F	4646
13	1969	1	7.0	M	5092
14	1969	1	8.0	F	4800
15	1969	1	8.0	M	4934
16	1969	1	9.0	F	4592
17	1969	1	9.0	M	4842
18	1969	1	10.0	F	4852
19	1969	1	10.0	M	5190
20	1969	1	11.0	F	4580
21	1969	1	11.0	M	4598
22	1969	1	12.0	F	4126
23	1969	1	12.0	M	4324
24	1969	1	13.0	F	4758
25	1969	1	13.0	M	5076
26	1969	1	14.0	F	5070

```
27     1969      1  14.0      M    5296
28     1969      1  15.0      F    4798
29     1969      1  15.0      M    5096
...     ...    ...   ...    ...     ...
15517  2007     10   NaN      F  180912
15518  2007     10   NaN      M  189157
15519  2007     11   NaN      F  173513
15520  2007     11   NaN      M  180814
15521  2007     12   NaN      F  173787
15522  2007     12   NaN      M  181426
15523  2008      1   NaN      F  174255
15524  2008      1   NaN      M  182789
15525  2008      2   NaN      F  165669
15526  2008      2   NaN      M  173434
15527  2008      3   NaN      F  172053
15528  2008      3   NaN      M  179129
15529  2008      4   NaN      F  169585
15530  2008      4   NaN      M  177399
15531  2008      5   NaN      F  173141
15532  2008      5   NaN      M  182294
15533  2008      6   NaN      F  169958
15534  2008      6   NaN      M  179267
15535  2008      7   NaN      F  183391
15536  2008      7   NaN      M  192714
15537  2008      8   NaN      F  182713
15538  2008      8   NaN      M  191315
15539  2008      9   NaN      F  179696
15540  2008      9   NaN      M  188964
15541  2008     10   NaN      F  175314
15542  2008     10   NaN      M  183219
15543  2008     11   NaN      F  158939
15544  2008     11   NaN      M  165468
15545  2008     12   NaN      F  173215
15546  2008     12   NaN      M  181235

[15547 rows x 5 columns]
```

2 Maximum Births

```
In [51]: df['births'].max()
```

```
Out[51]: 199622
```

3 Minimum Birth

```
In [52]: df['births'].mean()
```

```
Out[52]: 9762.293561458802

In [53]: #new1=df.fillna(0)
         #new1

In [81]: df.interpolate()
```

Out[81]:

	year	month	day	gender	births
0	1969	1	1.0	F	4046
1	1969	1	1.0	M	4440
2	1969	1	2.0	F	4454
3	1969	1	2.0	M	4548
4	1969	1	3.0	F	4548
5	1969	1	3.0	M	4994
6	1969	1	4.0	F	4440
7	1969	1	4.0	M	4520
8	1969	1	5.0	F	4192
9	1969	1	5.0	M	4198
10	1969	1	6.0	F	4710
11	1969	1	6.0	M	4850
12	1969	1	7.0	F	4646
13	1969	1	7.0	M	5092
14	1969	1	8.0	F	4800
15	1969	1	8.0	M	4934
16	1969	1	9.0	F	4592
17	1969	1	9.0	M	4842
18	1969	1	10.0	F	4852
19	1969	1	10.0	M	5190
20	1969	1	11.0	F	4580
21	1969	1	11.0	M	4598
22	1969	1	12.0	F	4126
23	1969	1	12.0	M	4324
24	1969	1	13.0	F	4758
25	1969	1	13.0	M	5076
26	1969	1	14.0	F	5070
27	1969	1	14.0	M	5296
28	1969	1	15.0	F	4798
29	1969	1	15.0	M	5096
...	...	...	...	...	...
15517	2007	10	31.0	F	180912
15518	2007	10	31.0	M	189157
15519	2007	11	31.0	F	173513
15520	2007	11	31.0	M	180814
15521	2007	12	31.0	F	173787
15522	2007	12	31.0	M	181426
15523	2008	1	31.0	F	174255
15524	2008	1	31.0	M	182789
15525	2008	2	31.0	F	165669

```
15526  2008      2  31.0      M  173434
15527  2008      3  31.0      F  172053
15528  2008      3  31.0      M  179129
15529  2008      4  31.0      F  169585
15530  2008      4  31.0      M  177399
15531  2008      5  31.0      F  173141
15532  2008      5  31.0      M  182294
15533  2008      6  31.0      F  169958
15534  2008      6  31.0      M  179267
15535  2008      7  31.0      F  183391
15536  2008      7  31.0      M  192714
15537  2008      8  31.0      F  182713
15538  2008      8  31.0      M  191315
15539  2008      9  31.0      F  179696
15540  2008      9  31.0      M  188964
15541  2008     10  31.0      F  175314
15542  2008     10  31.0      M  183219
15543  2008     11  31.0      F  158939
15544  2008     11  31.0      M  165468
15545  2008     12  31.0      F  173215
15546  2008     12  31.0      M  181235

[15547 rows x 5 columns]
```

4 For Male gender it says True and for Female False

```
In [105]: [df['gender']=='M']

Out[105]: [0        False
           1         True
           2        False
           3         True
           4        False
           5         True
           6        False
           7         True
           8        False
           9         True
           10       False
           11        True
           12       False
           13        True
           14       False
           15        True
           16       False
           17        True
           18       False
```

```
19        True
20       False
21        True
22       False
23        True
24       False
25        True
26       False
27        True
28       False
29        True
         ...
15517    False
15518     True
15519    False
15520     True
15521    False
15522     True
15523    False
15524     True
15525    False
15526     True
15527    False
15528     True
15529    False
15530     True
15531    False
15532     True
15533    False
15534     True
15535    False
15536     True
15537    False
15538     True
15539    False
15540     True
15541    False
15542     True
15543    False
15544     True
15545    False
15546     True
Name: gender, Length: 15547, dtype: bool]
```

```
In [24]: df['day'].unique()
```

```
Out[24]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10., 11., 12., 13.,
                14., 15., 16., 17., 18., 19., 20., 21., 22., 23., 24., 25., 26.,
                27., 28., 29., 30., 31., 99.,  0.])
```

```
In [31]: df.describe()
```

Out[31]:

	year	month	day	births
count	15547.000000	15547.000000	15547.000000	15547.000000
mean	1979.037435	6.515919	17.221265	9762.293561
std	6.728340	3.449632	15.357008	28552.465810
min	1969.000000	1.000000	0.000000	1.000000
25%	1974.000000	4.000000	8.000000	4358.000000
50%	1979.000000	7.000000	16.000000	4814.000000
75%	1984.000000	10.000000	24.000000	5289.500000
max	2008.000000	12.000000	99.000000	199622.000000

```
In [39]: new9=df['year'].astype('category')
         new10=df['month'].astype('category')
         new11=df['day'].astype('category')
```

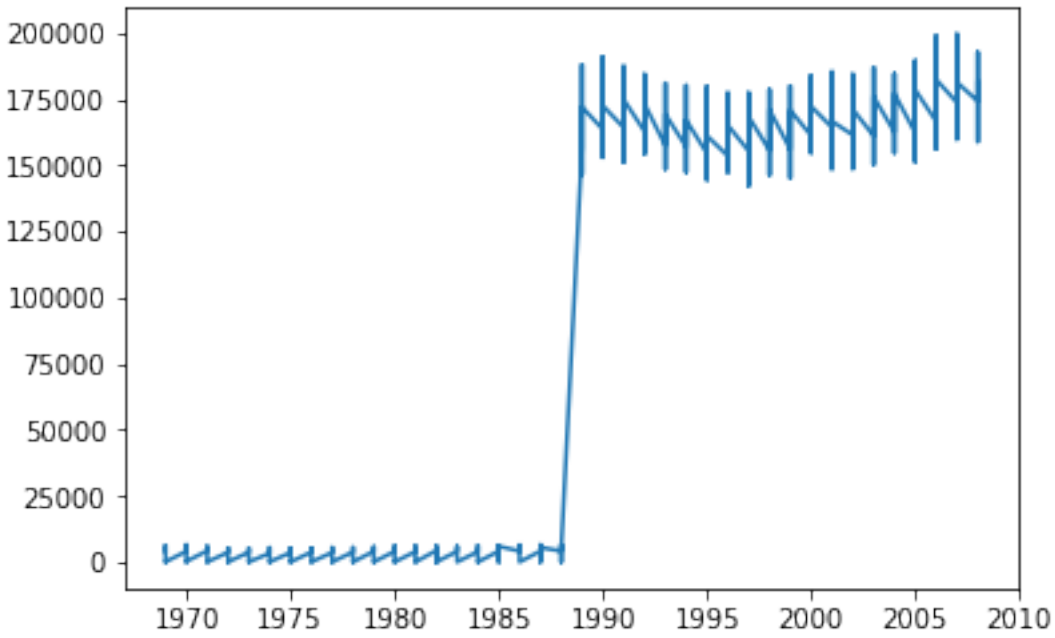
```
In [40]: df.describe()
```

Out[40]:

	births
count	15547.000000
mean	9762.293561
std	28552.465810
min	1.000000
25%	4358.000000
50%	4814.000000
75%	5289.500000
max	199622.000000

```
In [67]: plt.plot(df['year'],df['births'])
```

```
Out[67]: [<matplotlib.lines.Line2D at 0x7f43b0e61898>]
```



```
In [76]: table = pd.pivot_table(df, values='births', index=['day'],columns=['gender'], aggfunc=n
         table
```

Out[76]:

gender	F	M
day		
1.0	1116209	1170645
2.0	1121757	1181411
3.0	1124014	1185073
4.0	1115250	1171338
5.0	1121260	1175691
6.0	1124184	1183889
7.0	1126814	1187267
8.0	1132077	1190299
9.0	1128536	1186519
10.0	1134233	1192964
11.0	1130325	1189236
12.0	1132676	1190280
13.0	1117267	1173629
14.0	1137986	1197565
15.0	1136173	1197253
16.0	1137104	1195352
17.0	1138720	1196738
18.0	1137461	1197488
19.0	1135667	1195516
20.0	1140403	1197228
21.0	1134080	1195483
22.0	1129267	1188444
23.0	1121885	1183289
24.0	1116901	1173599
25.0	1113857	1170118
26.0	1121659	1178549
27.0	1128657	1185431
28.0	1134234	1194396
29.0	1065670	1123608
30.0	1041610	1100178
31.0	654269	687857
99.0	3143	3260

5 Sum of males

```
In [77]: df.loc[df['gender']=='M','births'].sum()
```

Out[77]: 77738555

6 Sum of Females

```
In [82]: df.loc[df['gender']=='F','births'].sum()
```

```
Out[82]: 74035823
```

```
In [86]: df['year'][df.births==df['births']].max()
```

```
Out[86]: 2008
```

7 Prints sum of male and female in respective year

```
In [93]: pd.pivot_table(df, values='births', index=['year'],columns=['gender'],aggfunc=np.sum)
```

```
Out[93]:          births
          gender        F        M
          year
          1969    1753634  1846572
          1970    1819164  1918636
          1971    1736774  1826774
          1972    1592347  1673888
          1973    1533102  1613023
          1974    1543005  1627626
          1975    1535546  1618010
          1976    1547613  1628863
          1977    1623363  1708796
          1978    1626324  1711976
          1979    1705837  1793958
          1980    1762459  1855522
          1981    1772037  1863478
          1982    1797239  1888218
          1983    1775299  1867522
          1984    1791802  1881766
          1985    1834774  1930290
          1986    1833708  1926987
          1987    1860111  1953105
          1988    1909210  2004583
          1989    1973712  2071981
          1990    2030966  2131951
          1991    2011601  2103741
          1992    1985118  2084310
          1993    1953456  2051067
          1994    1932234  2024691
          1995    1904871  1998141
          1996    1902664  1992210
          1997    1896928  1987401
          1998    1927106  2018086
          1999    1934510  2028955
```

```
2000     1984255  2079568
2001     1970770  2060761
2002     1966519  2060857
2003     1999387  2096705
2004     2010710  2108197
2005     2022892  2122727
2006     2084957  2188268
2007     2111890  2212118
2008     2077929  2177227
```

pd.pivot_table(df,values='births',index=['year'],columns=['gender'],margins=True,aggfunc=sum)

8 Print sum of Female and males in respective year in ascending order and use of margin as new column All is added

```
In [108]: pd.pivot_table(df,values='births',index=['year'],columns=['gender'],margins=True,aggfu
```

```
Out[108]: gender         F         M        All
          year
          1973     1533102   1613023    3146125
          1975     1535546   1618010    3153556
          1974     1543005   1627626    3170631
          1976     1547613   1628863    3176476
          1972     1592347   1673888    3266235
          1977     1623363   1708796    3332159
          1978     1626324   1711976    3338300
          1979     1705837   1793958    3499795
          1971     1736774   1826774    3563548
          1969     1753634   1846572    3600206
          1980     1762459   1855522    3617981
          1981     1772037   1863478    3635515
          1983     1775299   1867522    3642821
          1984     1791802   1881766    3673568
          1982     1797239   1888218    3685457
          1970     1819164   1918636    3737800
          1986     1833708   1926987    3760695
          1985     1834774   1930290    3765064
          1987     1860111   1953105    3813216
          1997     1896928   1987401    3884329
          1996     1902664   1992210    3894874
          1995     1904871   1998141    3903012
          1988     1909210   2004583    3913793
          1998     1927106   2018086    3945192
          1994     1932234   2024691    3956925
          1999     1934510   2028955    3963465
          1993     1953456   2051067    4004523
          2002     1966519   2060857    4027376
```

```
2001     1970770   2060761    4031531
1989     1973712   2071981    4045693
2000     1984255   2079568    4063823
1992     1985118   2084310    4069428
2003     1999387   2096705    4096092
1991     2011601   2103741    4115342
2004     2010710   2108197    4118907
2005     2022892   2122727    4145619
1990     2030966   2131951    4162917
2008     2077929   2177227    4255156
2006     2084957   2188268    4273225
2007     2111890   2212118    4324008
All     74035823  77738555  151774378
```

9 Print last 5 values using tail mehod

```
In [110]: pd.pivot_table(df,values='births',index=['year'],columns=['gender'],margins=True,aggfu
```

```
Out[110]: gender         F         M        All
          year
          1990     2030966   2131951    4162917
          2008     2077929   2177227    4255156
          2006     2084957   2188268    4273225
          2007     2111890   2212118    4324008
          All     74035823  77738555  151774378
```

```
In [ ]: # Print in ascending order
```

```
In [112]: pd.pivot_table(df,values='births',index=['year'],columns=['gender'],margins=True,aggfu
```

```
Out[112]: gender         F         M        All
          year
          All     74035823  77738555  151774378
          2007     2111890   2212118    4324008
          2006     2084957   2188268    4273225
          2008     2077929   2177227    4255156
          1990     2030966   2131951    4162917
          2005     2022892   2122727    4145619
          2004     2010710   2108197    4118907
          1991     2011601   2103741    4115342
          2003     1999387   2096705    4096092
          1992     1985118   2084310    4069428
          2000     1984255   2079568    4063823
          1989     1973712   2071981    4045693
          2001     1970770   2060761    4031531
          2002     1966519   2060857    4027376
          1993     1953456   2051067    4004523
          1999     1934510   2028955    3963465
```

```
1994     1932234   2024691    3956925
1998     1927106   2018086    3945192
1988     1909210   2004583    3913793
1995     1904871   1998141    3903012
1996     1902664   1992210    3894874
1997     1896928   1987401    3884329
1987     1860111   1953105    3813216
1985     1834774   1930290    3765064
1986     1833708   1926987    3760695
1970     1819164   1918636    3737800
1982     1797239   1888218    3685457
1984     1791802   1881766    3673568
1983     1775299   1867522    3642821
1981     1772037   1863478    3635515
1980     1762459   1855522    3617981
1969     1753634   1846572    3600206
1971     1736774   1826774    3563548
1979     1705837   1793958    3499795
1978     1626324   1711976    3338300
1977     1623363   1708796    3332159
1972     1592347   1673888    3266235
1976     1547613   1628863    3176476
1974     1543005   1627626    3170631
1975     1535546   1618010    3153556
1973     1533102   1613023    3146125
```

```
In [161]: df.index
```

```
Out[161]: RangeIndex(start=0, stop=42, step=1)
```