1_Birth Dataset_jithin

August 6, 2018

```
In [1]: import pandas as pd
        from datetime import datetime

In [2]: df=pd.read_excel("Birth_dataset.xlsx")

In [3]: df.isnull().sum()

Out[3]: year        0
        month       0
        day       480
        gender      0
        births      0
        dtype: int64
```

1) 480 null values in Day Column.

```
In [4]: df.describe()
```

Out[4]:

	year	month	day	births
count	15547.000000	15547.000000	15067.000000	15547.000000
mean	1979.037435	6.515919	17.769894	9762.293561
std	6.728340	3.449632	15.284034	28552.465810
min	1969.000000	1.000000	1.000000	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 [5]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15547 entries, 0 to 15546
Data columns (total 5 columns):
year      15547 non-null int64
month     15547 non-null int64
day       15067 non-null float64
gender    15547 non-null object
births    15547 non-null int64
dtypes: float64(1), int64(3), object(1)
memory usage: 607.4+ KB
```

```
In [6]: #### Categorical Values in Day

In [7]: df.head()
```

Out[7]:

	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

```
In [8]: # df['year'] = df['year'].astype('category')
        # df['month'] = df['month'].astype('category')
        # df['day'] = df['day'].astype('category')
        # df['gender'] = df['gender'].astype('category')

In [9]: df.describe()
```

Out[9]:

	year	month	day	births
count	15547.000000	15547.000000	15067.000000	15547.000000
mean	1979.037435	6.515919	17.769894	9762.293561
std	6.728340	3.449632	15.284034	28552.465810
min	1969.000000	1.000000	1.000000	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 [10]: Columns_list=list(df.columns.values)
         print("Unique Values in each Columns")

         for col in Columns_list:
             print(col,":",df[col].nunique())
```

```
Unique Values in each Columns
year : 40
month : 12
day : 32
gender : 2
births : 3137
```

```
In [11]: pd.pivot_table(df,values='births',index='year',aggfunc='sum',margins=True)
```

Out[11]:

year	births
1969	3600206
1970	3737800
1971	3563548

```
1972     3266235
1973     3146125
1974     3170631
1975     3153556
1976     3176476
1977     3332159
1978     3338300
1979     3499795
1980     3617981
1981     3635515
1982     3685457
1983     3642821
1984     3673568
1985     3765064
1986     3760695
1987     3813216
1988     3913793
1989     4045693
1990     4162917
1991     4115342
1992     4069428
1993     4004523
1994     3956925
1995     3903012
1996     3894874
1997     3884329
1998     3945192
1999     3963465
2000     4063823
2001     4031531
2002     4027376
2003     4096092
2004     4118907
2005     4145619
2006     4273225
2007     4324008
2008     4255156
All    151774378
```

2) Minimum number of babies born in year 1973, 1975

```
In [12]: pd.pivot_table(df,values='births',index='year',aggfunc='sum',margins=True).sort_values(
```

```
Out[12]:          births
year
1973     3146125
1975     3153556
1974     3170631
```

```
1976     3176476
1972     3266235
1977     3332159
1978     3338300
1979     3499795
1971     3563548
1969     3600206
1980     3617981
1981     3635515
1983     3642821
1984     3673568
1982     3685457
1970     3737800
1986     3760695
1985     3765064
1987     3813216
1997     3884329
1996     3894874
1995     3903012
1988     3913793
1998     3945192
1994     3956925
1999     3963465
1993     4004523
2002     4027376
2001     4031531
1989     4045693
2000     4063823
1992     4069428
2003     4096092
1991     4115342
2004     4118907
2005     4145619
1990     4162917
2008     4255156
2006     4273225
2007     4324008
All    151774378
```

3) Maximum number of babies born in 2007

```
In [13]: pd.pivot_table(df,values='births',index='gender',aggfunc='sum',margins=True)

Out[13]:            births
         gender
         F        74035823
         M        77738555
         All     151774378
```

4) Males outnumber females.

```
In [14]: pd.pivot_table(df,values='births',index='year',columns='gender',aggfunc='sum',margins=T
```

Out[14]:

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 [15]: pd.pivot_table(df,values='births',index='year',columns='gender',aggfunc='sum',margins=T
```

Out[15]:

gender	F	M	All
year			
1969	1753634	1846572	3600206
1970	1819164	1918636	3737800
1971	1736774	1826774	3563548
1972	1592347	1673888	3266235
1973	1533102	1613023	3146125
1974	1543005	1627626	3170631
1975	1535546	1618010	3153556
1976	1547613	1628863	3176476
1977	1623363	1708796	3332159
1978	1626324	1711976	3338300
1979	1705837	1793958	3499795
1980	1762459	1855522	3617981
1981	1772037	1863478	3635515
1982	1797239	1888218	3685457
1983	1775299	1867522	3642821
1984	1791802	1881766	3673568
1985	1834774	1930290	3765064
1986	1833708	1926987	3760695
1987	1860111	1953105	3813216
1988	1909210	2004583	3913793
1989	1973712	2071981	4045693
1990	2030966	2131951	4162917
1991	2011601	2103741	4115342
1992	1985118	2084310	4069428
1993	1953456	2051067	4004523
1994	1932234	2024691	3956925
1995	1904871	1998141	3903012
1996	1902664	1992210	3894874
1997	1896928	1987401	3884329
1998	1927106	2018086	3945192
1999	1934510	2028955	3963465
2000	1984255	2079568	4063823
2001	1970770	2060761	4031531

```
2002     1966519   2060857    4027376
2003     1999387   2096705    4096092
2004     2010710   2108197    4118907
2005     2022892   2122727    4145619
2006     2084957   2188268    4273225
2007     2111890   2212118    4324008
2008     2077929   2177227    4255156
All     74035823  77738555  151774378
```

```
In [16]: pd.pivot_table(df,values='births',index='month',columns='gender',aggfunc='mean',margins
```

Out[16]:

gender	F	M	All
month			
1	9242.64	9691.81	9467.23
2	9057.98	9558.44	9307.40
3	9437.58	9919.44	9678.51
4	9130.77	9651.94	9390.75
5	9396.31	9953.59	9674.31
6	9475.93	9965.96	9721.33
7	9957.64	10482.04	10219.84
8	10092.85	10576.01	10334.61
9	10035.70	10544.02	10289.47
10	9709.43	10098.17	9904.40
11	9234.71	9676.87	9455.79
12	9455.89	9897.19	9676.54
All	9521.07	10003.67	9762.29

```
In [17]: pd.pivot_table(df,values='births',index='gender',columns='month',aggfunc='mean',margins
```

Out[17]:

month	1	2	3	4	5	6	7	\
gender								
F	9242.64	9057.98	9437.58	9130.77	9396.31	9475.93	9957.64	
M	9691.81	9558.44	9919.44	9651.94	9953.59	9965.96	10482.04	
All	9467.23	9307.40	9678.51	9390.75	9674.31	9721.33	10219.84	

month	8	9	10	11	12	All
gender						
F	10092.85	10035.70	9709.43	9234.71	9455.89	9521.07
M	10576.01	10544.02	10098.17	9676.87	9897.19	10003.67
All	10334.61	10289.47	9904.40	9455.79	9676.54	9762.29

```
In [18]: pd.pivot_table(df,values='births',index='day',columns='month',aggfunc='mean',margins=Fa
```

Out[18]:

month	1	2	3	4	5	6	7	8	\
day									
1.0	4009.22	4661.45	4742.18	4623.82	4651.27	4751.48	5021.98	5068.00	
2.0	4247.40	4743.02	4750.55	4743.82	4616.70	4804.65	5021.45	5008.52	
3.0	4500.90	4761.82	4871.77	4652.92	4570.23	4783.27	4869.42	5065.25	
4.0	4571.35	4760.82	4821.55	4679.52	4577.00	4744.48	4335.32	5087.95	

5.0	4603.62	4728.30	4781.60	4586.82	4687.30	4706.88	4698.82	5108.00
6.0	4668.15	4678.10	4721.88	4642.85	4678.95	4771.85	4984.50	5110.92
7.0	4706.92	4649.65	4722.45	4689.85	4658.95	4681.05	5153.42	5082.68
8.0	4629.65	4668.52	4718.23	4712.02	4623.75	4681.95	5159.80	5153.58
9.0	4537.77	4713.58	4692.08	4649.88	4608.85	4784.27	5075.40	5027.77
10.0	4591.70	4800.85	4785.38	4610.68	4623.18	4831.38	5041.30	5083.38
11.0	4675.15	4815.80	4785.73	4609.95	4652.20	4773.62	4982.23	5104.23
12.0	4700.80	4823.70	4762.42	4590.80	4699.05	4746.25	4936.68	5177.62
13.0	4730.05	4639.40	4652.45	4559.48	4661.65	4671.42	4932.40	5076.30
14.0	4816.20	4862.75	4719.02	4680.12	4670.90	4724.45	5106.00	5096.75
15.0	4733.65	4706.32	4670.02	4702.90	4678.52	4750.48	5140.12	5106.50
16.0	4665.02	4732.02	4703.68	4656.45	4661.45	4858.35	5092.75	5047.80
17.0	4654.65	4792.68	4828.55	4604.40	4626.73	4863.60	5046.52	5074.90
18.0	4707.32	4820.98	4773.50	4633.50	4673.88	4816.27	5000.30	5148.00
19.0	4731.52	4751.60	4748.12	4582.70	4721.75	4768.98	4969.85	5153.40
20.0	4767.52	4751.27	4716.35	4591.27	4806.92	4831.27	5057.55	5141.10
21.0	4790.25	4671.50	4722.88	4656.42	4725.80	4751.18	5087.35	5082.20
22.0	4742.80	4751.52	4667.20	4686.70	4713.50	4795.62	5135.68	5058.85
23.0	4666.75	4757.45	4697.20	4643.73	4721.10	4850.55	5092.27	5007.05
24.0	4653.20	4829.00	4725.05	4581.15	4689.12	4897.52	5036.05	5035.20
25.0	4698.00	4857.02	4773.65	4602.25	4661.92	4890.92	5029.32	5107.68
26.0	4715.90	4790.58	4722.82	4588.50	4657.75	4853.98	4998.70	5167.50
27.0	4747.02	4693.75	4680.85	4571.95	4738.15	4852.32	5070.25	5124.70
28.0	4771.80	4695.30	4694.65	4656.55	4671.65	4862.98	5133.52	5099.77
29.0	4702.30	1934.17	4665.02	4613.25	4704.05	4881.65	5151.90	5062.00
30.0	4644.23	9.80	4663.42	4615.10	4601.18	4981.60	5101.23	5027.85
31.0	4598.27	6.42	4723.25	7.53	4597.05	7.81	5045.27	5048.65
99.0	15.50	13.85	17.03	13.80	15.28	16.78	17.21	17.66

month	9	10	11	12
day				
1.0	4908.32	5167.32	4729.80	4836.50
2.0	4982.00	5103.62	4727.15	4830.30
3.0	5003.92	5067.38	4821.77	4758.50
4.0	5013.40	5005.18	4849.40	4718.73
5.0	4954.60	5025.08	4808.08	4734.68
6.0	4955.00	5048.10	4758.48	4683.05
7.0	4995.45	5024.10	4783.18	4704.32
8.0	5165.48	4989.88	4752.75	4803.80
9.0	5263.40	4945.42	4784.12	4793.82
10.0	5214.50	4975.98	4836.30	4785.32
11.0	5151.95	4854.38	4845.30	4738.50
12.0	5160.82	4893.42	4791.02	4791.30
13.0	5073.88	4865.88	4732.82	4676.68
14.0	5204.27	4934.05	4782.15	4792.10
15.0	5255.35	4919.45	4751.52	4920.80
16.0	5322.88	4828.45	4774.45	4968.10
17.0	5281.58	4818.85	4842.40	4951.60

```
18.0   5246.68  4751.62  4865.30  4936.38
19.0   5261.50  4782.85  4844.38  4962.92
20.0   5248.28  4832.82  4819.38  4877.02
21.0   5271.90  4862.68  4800.82  4816.10
22.0   5316.82  4810.25  4601.90  4661.92
23.0   5320.42  4755.50  4650.65  4466.68
24.0   5284.05  4758.77  4647.12  4126.25
25.0   5240.00  4744.35  4649.80  3844.45
26.0   5250.65  4788.05  4587.25  4383.52
27.0   5190.15  4821.35  4511.55  4850.15
28.0   5168.45  4826.10  4590.77  5044.20
29.0   5218.82  4775.58  4676.73  5120.15
30.0   5224.77  4745.05  4765.48  5172.35
31.0     10.58  4662.80    11.32  4859.20
99.0     21.10    29.32    17.65    25.39
```

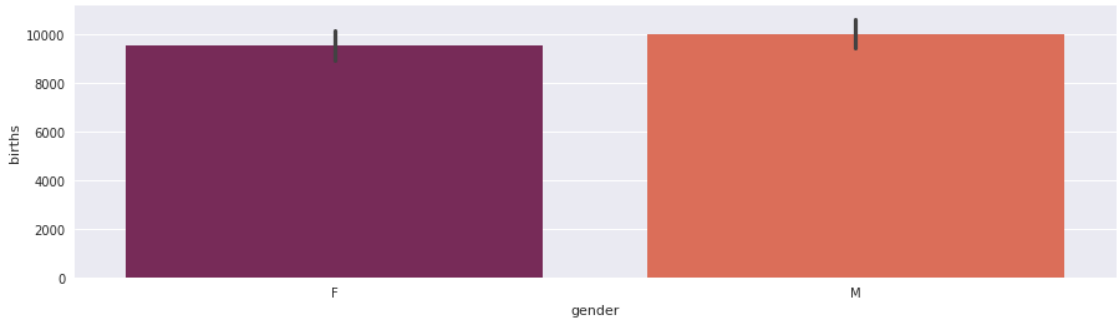
In [19]:
```
# 5316 people have birthday as 16th September (average). (Most)
```

In [20]:
```
import seaborn as sns
from matplotlib import pyplot as plt
```

In [21]:
```
sns.set(style="darkgrid")
```

In [22]:
```
plt.figure(figsize=(15,4))
sns.barplot(x=df['gender'], y=df['births'], palette="rocket")
```

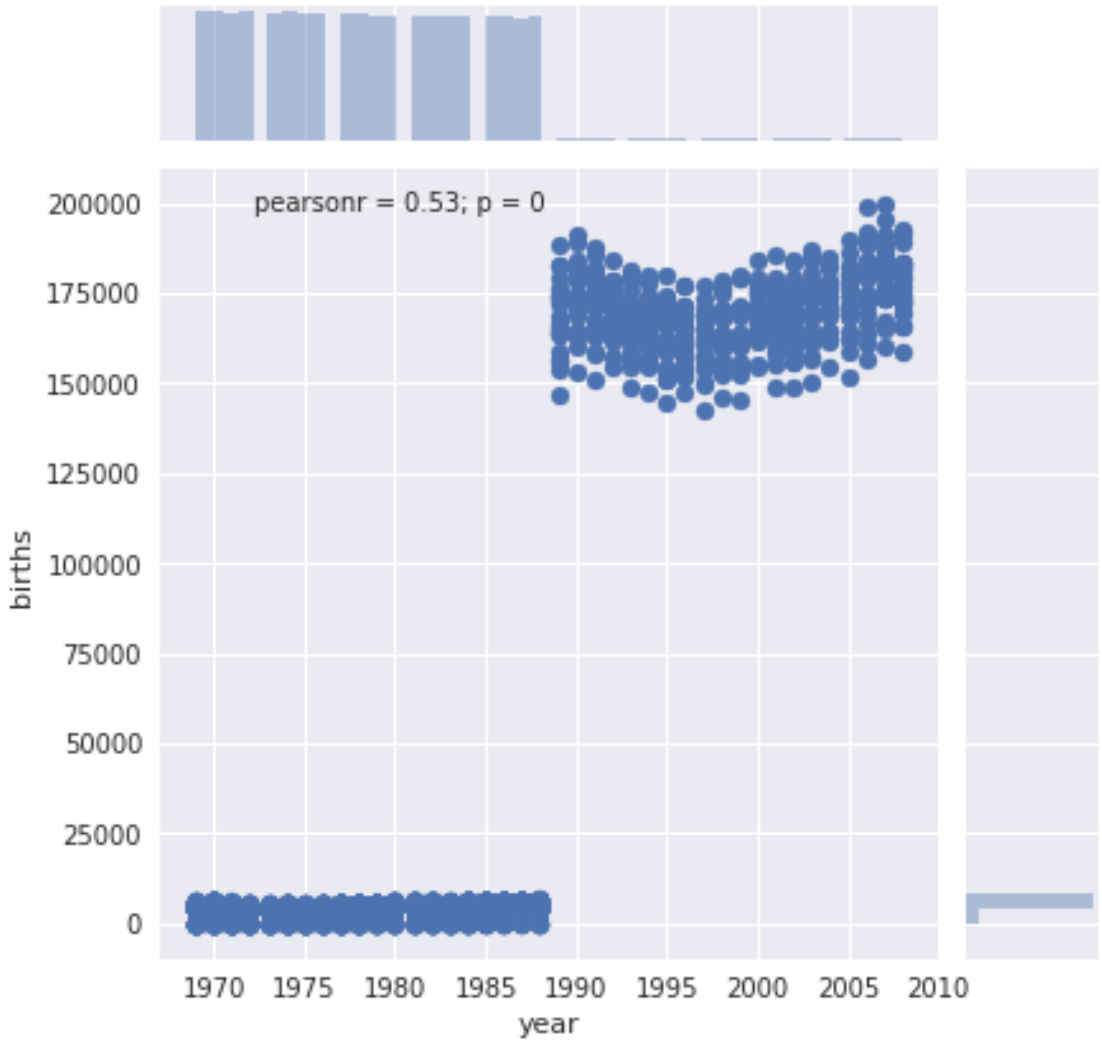
Out[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4895785c88>
```



In [23]:
```
plt.figure(figsize=(15,4))
sns.jointplot(x='year', y='births',data=df)
```

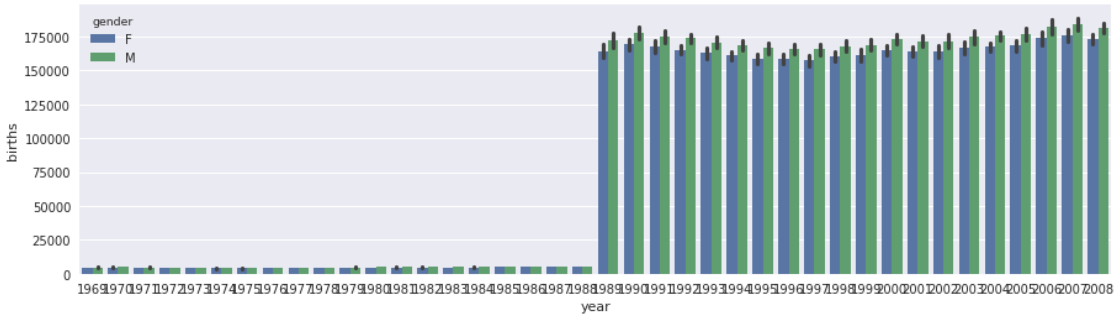
Out[23]:
```
<seaborn.axisgrid.JointGrid at 0x7f489f67eb70>
```

```
<matplotlib.figure.Figure at 0x7f489f67eba8>
```



In [24]:
```
plt.figure(figsize=(15,4))
sns.barplot(x="year", y="births", hue="gender", data=df)
```

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x7f489f751d68>



```
In [42]: df_new=pd.pivot_table(df,values='births',index='year',aggfunc='sum').reset_index().rena
         df_new.columns.values

Out[42]: array(['year', 'births'], dtype=object)

In [43]: plt.figure(figsize=(15,4))
         sns.regplot(x='year',y='births',data=df_new)

Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4872c612b0>
```