D1_Narender_Q1

August 6, 2018

1 ANALYSING AND VISULAZING THE FOLLOWING DATA SETS USING PANDAS, NUMPY AND MATPLOTLIB

```
In [ ]: # SUMMARY
        """This data consists of (15547, 5) rows and columns,where it is related to no of
        births in every day of an year ofUS hospitalfrom 1969 to 2008.we have 408 NaN's in
        column "day" and we had some outliers in the "day" column so ireplaced it with upper
        quartile values.we can clearly see that from 1988 there is a sudden change in birth
        rate andthere is slight difference in male birth count and female birth count there
        is chance for further analysis so thatwe can clearly know which days there is more
        birthswhether it is weekend or weekday .Here there is a chance of data that are from
        two different populations(it means two differnt countrys)becouse if we see the box
        plot of birth ratethere is a sudden raise in the birth rate practically it cant be"""
```

```
In [ ]: import os
        os.getcwd()
```

2 Birth data of US

```
In [86]: # required librarys
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns
```

```
In [87]: # Reading Data
         BirthD = pd.read_excel("BirthData.xlsx")

         # Anlaysing
         print(BirthD.head(10))
         print(BirthD.tail(10))
```

```
   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
```

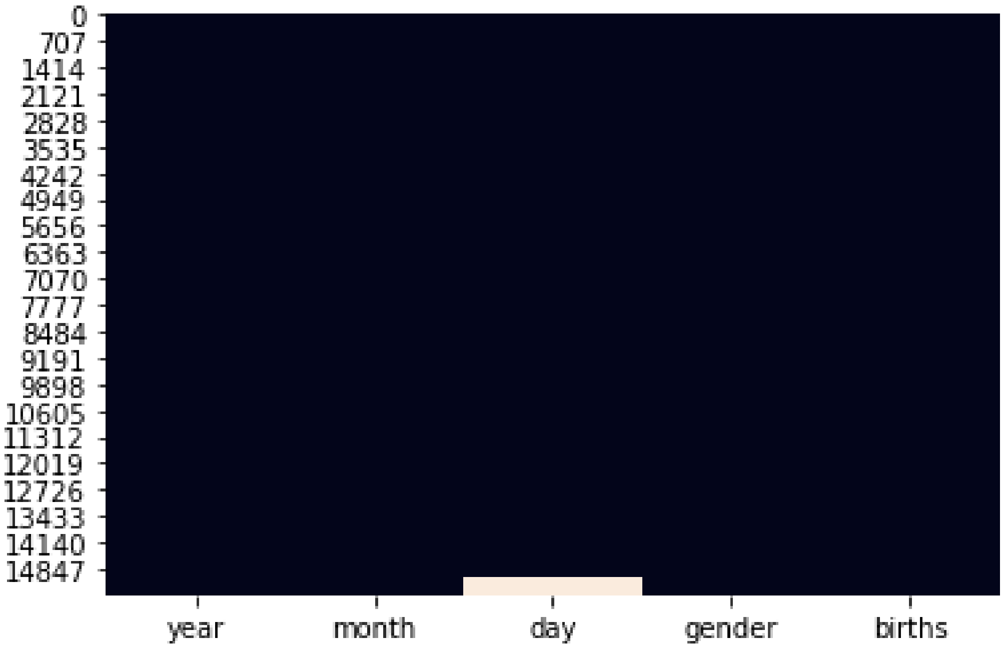
```
4  1969      1  3.0      F    4548
5  1969      1  3.0      M    4994
6  1969      1  4.0      F    4440
7  1969      1  4.0      M    4520
8  1969      1  5.0      F    4192
9  1969      1  5.0      M    4198
       year  month  day gender  births
15537  2008      8  NaN      F  182713
15538  2008      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
```

```
In [88]: print(BirthD.describe())
         print(BirthD.shape)
```

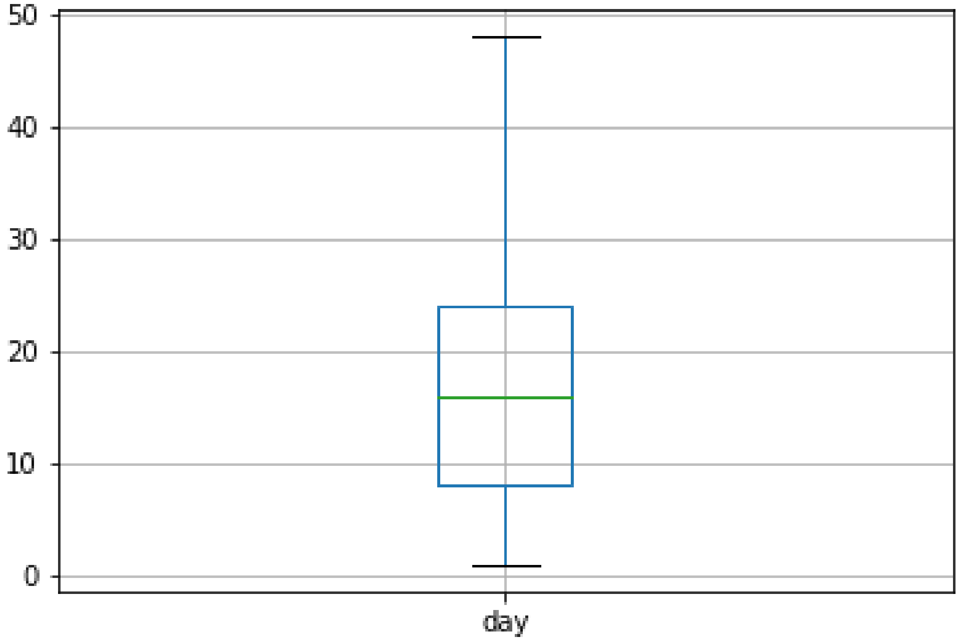
```
               year         month           day         births
count  15547.000000  15547.000000  15067.000000   15547.000000
mean    1979.037435      6.515919     17.769894    9762.293561
std        6.728340      3.449632     15.284034   28552.465810
min     1969.000000      1.000000      1.000000       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
(15547, 5)
```

```
In [89]: # checking for na values count
         print((BirthD.isna().sum()))
         print(sns.heatmap(BirthD.isnull(), cbar=False))
```

```
year        0
month       0
day       480
gender      0
births      0
dtype: int64
AxesSubplot(0.125,0.125;0.775x0.755)
```

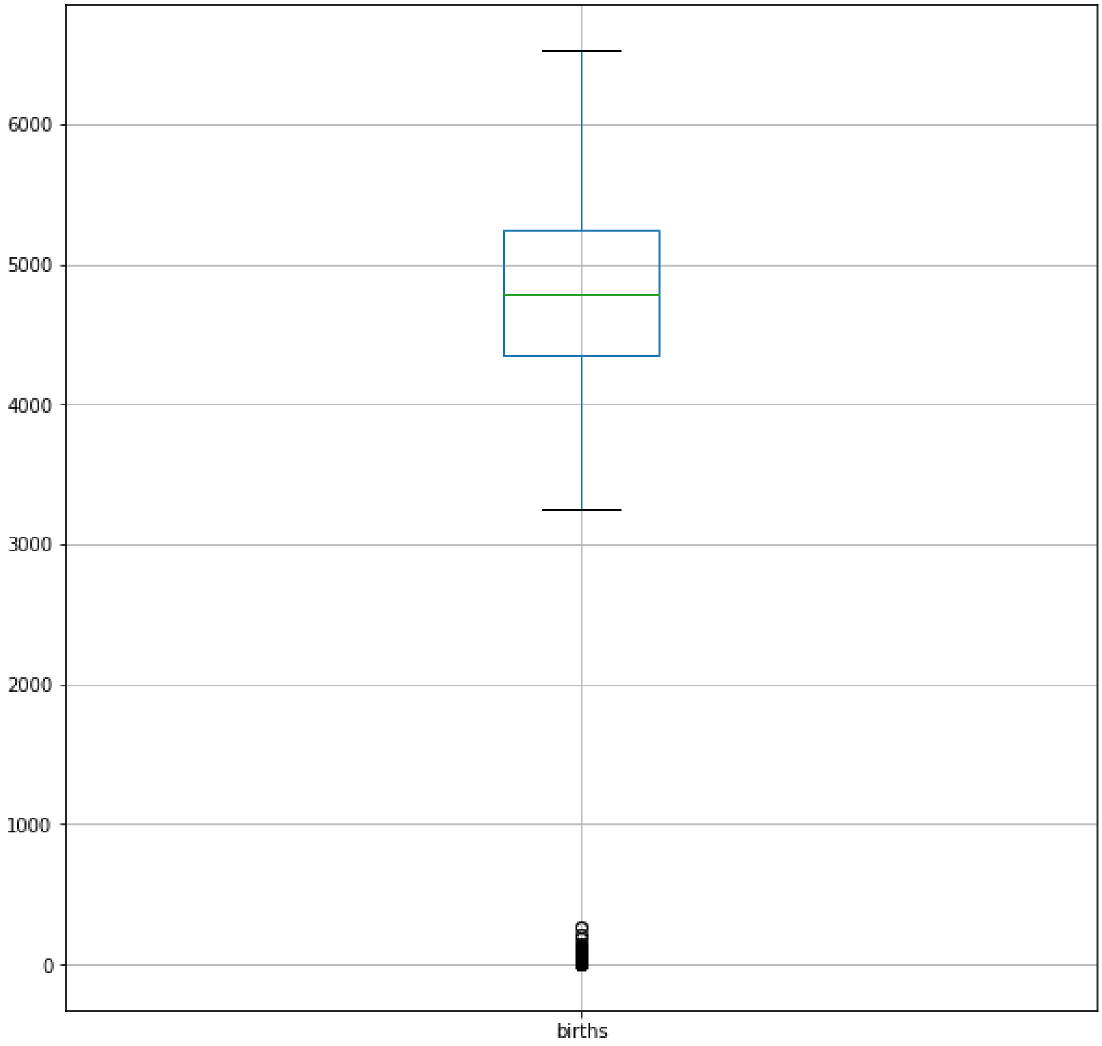


```
In [90]: # handling the na values
         BirthD = BirthD.dropna()

In [97]: # ploting outlier using box plot
         BirthD.boxplot(column='day', return_type='axes')

Out[97]: <matplotlib.axes._subplots.AxesSubplot at 0x7f5164971d68>
```



```
In [146]: BirthD.boxplot(column='births', return_type='axes',figsize=(10,10))
```

```
Out[146]: <matplotlib.axes._subplots.AxesSubplot at 0x7f511169f898>
```



```
In [91]: #handling outliers
         q75, q25 = np.percentile(BirthD.day, [75 ,25])
         iqr = q75-q25
         lwhisk = q75 + (1.5*iqr)
         BirthD["day"] = BirthD["day"].clip(upper=lwhBirthD.describe()isk)
         BirthD.describe()
```

```
In [115]: # changing the data types to apropriate data types
          print(BirthD.info())
          BirthD["day"] =BirthD["day"].astype("int64")
          BirthD["gender"] =BirthD["gender"].astype("category")
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 15067 entries, 0 to 15066
Data columns (total 5 columns):
year      15067 non-null int64
month     15067 non-null int64
day       15067 non-null int16
gender    15067 non-null category
births    15067 non-null int64
dtypes: category(1), int16(1), int64(3)
memory usage: 1.1 MB
None
```