datascience_assignment_10.1

July 14, 2018

0.1 Read the dataset from the below link

https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/06_Stats/US_Baby_Names/US_Ba

0.2 Steps

- Import numpy, pandas
- Read the CSV file from the URL provided using read_csv method of pandas and load to dataframe df
- Show first few records using head method on df

```
In [2]: import numpy as np
        import pandas as pd
```

```
In [3]: # Read the CSV file from the URL provided using read_csv method of pandas and load to
        df = pd.read_csv('https://raw.githubusercontent.com/guipsamora/pandas_exercises/master

        # Show first few records using head method on df
        df.head()
```

```
Out[3]:    Unnamed: 0     Id     Name  Year Gender State  Count
        0       11349  11350     Emma  2004      F    AK     62
        1       11350  11351  Madison  2004      F    AK     48
        2       11351  11352   Hannah  2004      F    AK     46
        3       11352  11353    Grace  2004      F    AK     44
        4       11353  11354    Emily  2004      F    AK     41
```

0.3 1. Delete unnamed columns

0.4 Steps

- Find all the columns whose name starts with Unnamed, case insensitive on axis=1
- Delete the columns by using drop method on dataframe df
- Persist the result by passing inplace=True to the drop method
- Display first few records using head method on df

```
In [4]: # Find all the columns whose name starts with Unnamed, case insensitive on axis=1
        # Delete the columns by using drop method on dataframe df
        # Persist the result by passing inplace=True to the drop method
        df.drop(df.columns[df.columns.str.contains('Unnamed',case = False)], axis = 1, inplace
In [5]: # Display first few records using head method on df
        df.head()

Out[5]:       Id     Name  Year Gender State  Count
        0  11350     Emma  2004      F    AK     62
        1  11351  Madison  2004      F    AK     48
        2  11352   Hannah  2004      F    AK     46
        3  11353    Grace  2004      F    AK     44
        4  11354    Emily  2004      F    AK     41
```

0.5 2. Show the distribution of male and female

0.6 Steps:

- Find all the Male records by using filter on Gender column of dataframe equal to 'M' and store it dataframe male
- Get distribution of Male records (count, mean, standard deviation, min, max, lower quartile, median, upper quartile) by using describe method on Count field on dataframe male
- Find all the female records by using filter on Gender column of dataframe equal to 'F' and store it dataframe female
- Get distribution of Feale records (count, mean, standard deviation, min, max, lower quartile, median, upper quartile) by using describe method on Count field on dataframe female

```
In [9]: # Find all the Male records by using filter on Gender column of dataframe equal to 'M'
        # and store it dataframe male
        male = df[df['Gender'] == 'M']

        # Get distribution of Male records (count, mean, standard deviation, min, max, lower q
        # by using describe method on Count field on dataframe male
        male['Count'].describe()

Out[9]: count    457549.000000
        mean         41.615650
        std         118.074308
        min           5.000000
        25%           7.000000
        50%          12.000000
        75%          29.000000
        max        4167.000000
        Name: Count, dtype: float64

In [10]: # Find all the female records by using filter on Gender column of dataframe equal to
         # and store it dataframe female
         female = df[df['Gender'] == 'F']

         # Get distribution of Feale records (count, mean, standard deviation, min, max, lower
         # median, upper quartile) by using describe method on Count field on dataframe female
         female['Count'].describe()
```

```
Out[10]: count    558846.000000
         mean         29.310925
         std          75.962992
         min           5.000000
         25%           6.000000
         50%          10.000000
         75%          23.000000
         max        3634.000000
         Name: Count, dtype: float64
```

0.7 3. Show the top 5 most preferred names

0.8 Steps:

- Get count of records group by column Name on dataframe df and store it in name_count_df
- Sort in descending order on Count field of name_count_df and take the first 5 records and store in in name_count_sorted5
- Print the top 5 preferred names

```
In [11]: # Get count of records group by column Name on dataframe df and store it in name_coun
         name_count_df = df.groupby('Name').count()

         # Sort in descending order  on Count field of name_count_df and take the first 5 reco
         name_count_sorted5 = name_count_df.sort_values(by='Count',  ascending=False).head(5)

         #Print the top 5 preferred names
         print(name_count_sorted5['Count'])
```

```
Name
Riley     1112
Avery     1080
Jordan    1073
Peyton    1064
Hayden    1049
Name: Count, dtype: int64
```

0.9 4. What is the median name occurence in the dataset

0.10 Steps:

- Calculate count on dataframe df group by 'Name' column and store in dataframe name_count_df
- Sort on name_count_df by Count field and store in dataframe name_count_df_sorted
- Calculate median on name_count_df
- Display median value

```
In [12]: # Calculate count on dataframe df group by 'Name' column and store in dataframe name_
         name_count_df = df.groupby('Name').count()
```

```
# Sort on name_count_df by Count field and store in dataframe name_count_df_sorted
name_count_df_sorted = name_count_df.sort_values(by='Count',  ascending=True)['Count']

# Calculate median on name_count_df
median_value = name_count_df_sorted.median()

# Display median value
print("Name occurrence median = " + str(median_value))
```

```
Name occurrence median = 8.0
```

0.11 5. Distribution of male and female born count by states

0.12 Steps:

- Get all the Male by filtering column Gender equal to 'M' and store in dataframe male
- Get count group by State column on male dataframe and store in male_count_groupby_state
- Add a new column 'Male' to dataframe male_count_groupby_state populated with values of 'Count' column
- Get all the Feale by filtering column Gender equal to 'F' and store in dataframe female
- Get count group by State column on female dataframe and store in female_count_groupby_state
- Add a new column 'Female' to dataframe female_count_groupby_state populated with values of 'Count' column
- Concanate both male_count_groupby_state, female_count_groupby_state on axis=1 and store in new dataframe male_female_count_groupby_state
- Show the columns "Male", "Female" on datafame male_female_count_groupby_state

```
In [29]: # Get all the Male by filtering column Gender equal to 'M' and store in dataframe mal
         # Get count group by State column on male dataframe and store in male_count_groupby_s
         #Add a new column 'Male' to dataframe male_count_groupby_state populated with values
         male = df[df['Gender'] == 'M']
         male_count_groupby_state = male.groupby('State').count()
         male_count_groupby_state["Male"] = male_count_groupby_state["Count"]

         #Get all the Feale by filtering column Gender equal to 'F' and store in dataframe fem
         # Get count group by State column on female dataframe and store in female_count_group
         # Add a new column 'Female' to dataframe female_count_groupby_state populated with va
         female = df[df['Gender'] == 'F']
         female_count_groupby_state = female.groupby('State').count()
         female_count_groupby_state["Female"] = female_count_groupby_state["Count"]
```

```
# Concanate both male_count_groupby_state, female_count_groupby_state on axis=1 and s
male_female_count_groupby_state = pd.concat([male_count_groupby_state, female_count_g

# Show the columns "Male", "Female" on datafame male_female_count_groupby_state
male_female_count_groupby_state.loc[:,male_female_count_groupby_state.columns.isin(["
```

Out[29]:

```
        Male  Female
State
AK      2587    2404
AL      8419    9878
AR      6475    7171
AZ     10820   14518
CA     31637   45144
CO      9183   11424
CT      5733    6575
DC      3000    3053
DE      2440    2549
FL     20070   25781
GA     15454   19385
HI      3546    3255
IA      6307    7131
ID      4833    4918
IL     16828   21268
IN     10613   13056
KS      6748    7753
KY      7267    8817
LA      9676   10510
MA      8609   10580
MD      9483   11276
ME      2777    2976
MI     13243   16038
MN      9004   10677
MO      9917   11948
MS      6862    7235
MT      2986    2690
NC     13530   17357
ND      2581    2399
NE      5029    5370
NH      2659    2957
NJ     12274   15041
NM      4966    5721
NV      6024    7092
NY     22585   28158
OH     14318   18143
OK      8138    9519
OR      7333    8604
PA     14171   17480
RI      2468    2558
```

```
SC       8195    9465
SD       2908    2838
TN      10588   13063
TX      27791   39760
UT       8233    9515
VA      11997   14759
VT       1618    1398
WA      11049   13329
WI       8940   10549
WV       3733    4305
WY       1904    1456
```