Laboratory Work 3 : Using Pandas for Data Analysis

Course: Python Data Processing

Student: Bahar Berra Uyar / KH-222ia.e

Instructor: Svitlana Mykolaivna Kovalenko

Date: 05/10/2024

1. Cover Page of the Report

Laboratory Work 3: Using Pandas for Data Analysis

This report focuses on the usage of the Pandas library in Python to perform data analysis on a dataset containing baby names registered in the United States over multiple years.

2. Topic and Goal of the Lab

Topic:

The primary topic of this laboratory session is **Using Pandas for Data Analysis**, where the objective is to understand and apply the core features of the Pandas library for analyzing data. The chosen dataset for this exercise is the **US Baby Names** dataset, available on Kaggle.

Goal:

The goal of this lab is to familiarize students with the functionalities provided by the Pandas library in Python. By analyzing the US Baby Names dataset

students will perform various data manipulation tasks, such as filtering, aggregating, visualizing, and discovering trends and patterns in the data.

3. Progress of the Work

The laboratory work was carried out using the **US Baby Names** dataset, available from Kaggle. This dataset includes information on baby names from **1880** to **2014**, specifying the gender, year of birth, and count of each name.

The exercises were performed according to the individual task assigned by the instructor. My individual task number was **2**, which involved completing several exercises to explore different functionalities of Pandas. Below is a detailed description of the exercises and the progress made:

3.1 Dataset Description:

The **US Baby Names** dataset is a publicly available dataset on Kaggle that contains data about baby names registered in the United States from 1880 to 2014. It includes columns like:

- **Id**: A unique identifier for each record
- **Name**: Baby name
- **Year**: Year the name was recorded
- **Gender**: Gender of the child (F for female, M for male)
- **Count**: The number of babies given that name in that yea

3.2 Individual Task Assignment:

According to the individual task formula:

```
L = 'B'  # First letter of my name
N = ord(L) % 5 + 1
N
```

```
2
```

I was assigned **Individual Task 2**

Individual task	Exercises

2	3, 4, 5, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19, 20, 22, 23, 24, 27

3.3 Summary of Completed Tasks:

Exercise 3: Get the names of dataset columns

```
import pandas as pd

# Load the US Baby Names dataset
df = pd.read_csv("NationalNames.csv")

#Ex.3
# Get the column names of the dataset
columns = df.columns
print(columns)
```

```
Index(['Id', 'Name', 'Year', 'Gender', 'Count'], dtype='object')
```

Exercise 4: Get general information about data in the dataset

```
#Ex.4
# Get general information (summary statistics) about the data
general_info = df.describe()
print(general_info)
```

```
                 Id          Year         Count
count  1.825433e+06  1.825433e+06  1.825433e+06
mean   9.127170e+05  1.972620e+03  1.846879e+02
std    5.269573e+05  3.352891e+01  1.566711e+03
min    1.000000e+00  1.880000e+03  5.000000e+00
25%    4.563590e+05  1.949000e+03  7.000000e+00
50%    9.127170e+05  1.982000e+03  1.200000e+01
75%    1.369075e+06  2.001000e+03  3.200000e+01
max    1.825433e+06  2.014000e+03  9.968000e+04
```

Exercise 5:. Find the number of unique names in whole dataset

```
#Ex.5
# Count the number of unique names
unique_names_count = df['Name'].nunique()
print(unique_names_count)
```

```
93889
```

Exercise 8: Find the most popular name based on the results of one year (the name for which Count is maximum)

```
#Ex.8
# Find the row with the maximum count in the dataset
most_popular_name_row = df.loc[df['Count'].idxmax()]

# Get the name and year for which the count is maximum
most_popular_name = most_popular_name_row['Name']
most_popular_year = most_popular_name_row['Year']

print(f"The most popular name is '{most_popular_name}' in {most_popular_year}.")
```

```
The most popular name is 'Linda' in 1947.
```

Exercise 9: Count the number of records with Count = minimum.

```
#Ex.9
# Find the number of records with the minimum count
min_count = df['Count'].min()
min_count_records = df[df['Count'] == min_count].shape[0]
print(min_count_records)
```

```
254615
```

Exercise 11: Find the year with the most number of unique names.

```
#Ex.11
# Group the data by 'Year' and count unique names for each year
unique_names_per_year = df.groupby('Year')['Name'].nunique()

# Find the year with the most unique names
year_most_unique_names = unique_names_per_year.idxmax()
most_unique_names_count = unique_names_per_year.max()

result = pd.DataFrame({'Year': [year_most_unique_names], 'Name': [most_unique_names_count]})
result.set_index('Year', inplace=True)

print(result)
```

```
      Name
Year
2008  32488
```

Exercise 12: Find most popular name of the year with the most number of unique names (that is in 2008)

```
#Ex.12
# Filter the dataset for 2008 and find the most popular name
df_2008 = df[df['Year'] == most_unique_names_year]
most_popular_name_2008 = df_2008[df_2008['Count'] == df_2008['Count'].max()]['Name'].values[0]
print(most_popular_name_2008)
```

```
Jacob
```

Exercise 13: Find the year when the name "Jacob" was the most popular as a female name.

```
#Ex.13
# Find the year when 'Jacob' was most popular as a female name
jacob_female = df[(df['Name'] == 'Jacob') & (df['Gender'] == 'F')]
most_popular_jacob_female_year = jacob_female[jacob_female['Count'] == jacob_female['Count'].max()]
print(most_popular_jacob_female_year)
```

```
             Id   Name  Year Gender  Count
1455556  1455557  Jacob  2004      F    171
```

Exercise 14: Find year, with the most number of gender neutral names (the same male and female names)

```
#Ex.14
# Filter the dataset to find gender-neutral names (names that appear for both male and female)
gender_neutral_names = df.groupby('Name').filter(lambda x: len(x['Gender'].unique()) == 2)

# Group by 'Year' and count the number of unique gender-neutral names per year
gender_neutral_counts_per_year = gender_neutral_names.groupby('Year')['Name'].nunique()

# Find the year with the most gender-neutral names
year_most_gender_neutral = gender_neutral_counts_per_year.idxmax()
most_gender_neutral_count = gender_neutral_counts_per_year.max()

result = pd.DataFrame({'Year': [year_most_gender_neutral], 'Gender_neutral_names': [most_gender_neutral_count]})
result.set_index('Year', inplace=True)

print(result)
```

```
      Gender_neutral_names
Year
2009                  7372
```

Exercise 16: Find the year when the greatest number of children was born

```
#Ex.16
# Find the year with the greatest number of births
year_greatest_births = total_births_per_year.idxmax()
print(year_greatest_births)
```

```
1957
```

Exercise 17: Find the number of girls and boys that were born in each year

```
#Ex.17
# Count births by gender per year
births_by_gender = df.groupby(['Year', 'Gender'])['Count'].sum().unstack(fill_value=0)
print(births_by_gender.head())
```

```
Gender       F       M
Year
1880     90993  110491
1881     91954  100745
1882    107850  113688
1883    112321  104629
1884    129022  114445
```

Exercise 18: Count the number of years when more girls were born than boys

```
#Ex.18
# Count the number of years with more girls born than boys
years_more_girls = (births_by_gender['F'] > births_by_gender['M']).sum()
print(years_more_girls)
```

```
54
```

Exercise 19: Draw the plot of total births per year of boys and girls

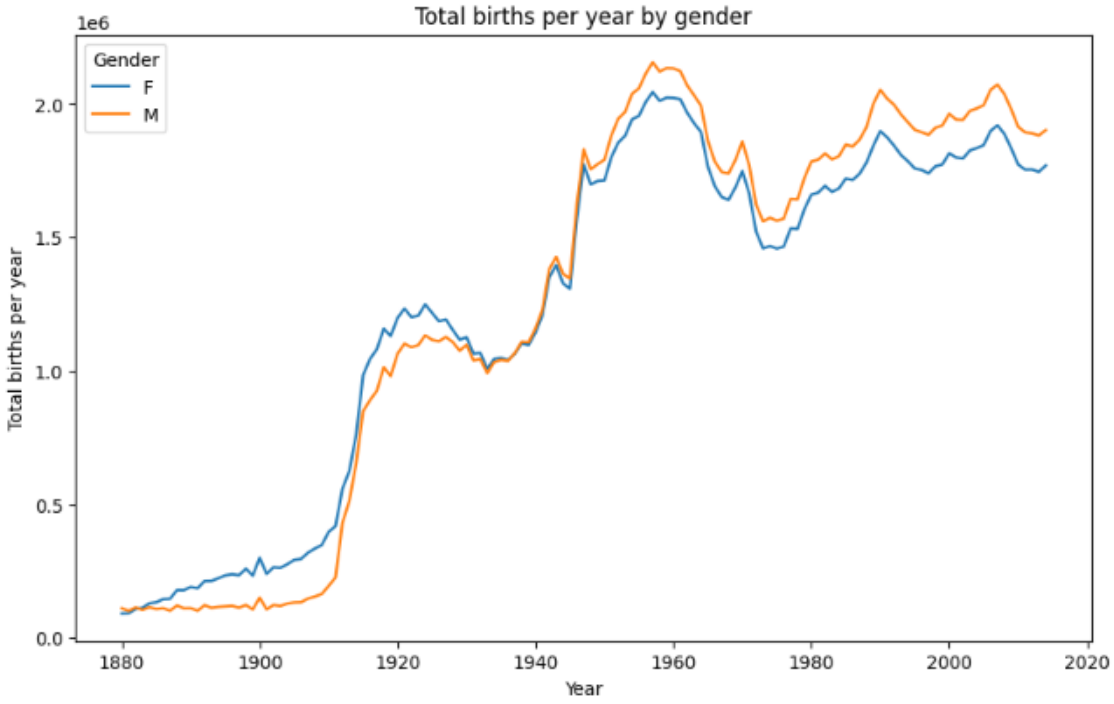
```
#Ex.19
import pandas as pd
import matplotlib.pyplot as plt

# Group by 'Year' and 'Gender', and sum the 'Count' column to get the total number of births per gender per year
total_births_per_year_gender = df.groupby(['Year', 'Gender'])['Count'].sum().unstack()

# Plot the total births per year for each gender
total_births_per_year_gender.plot(kind='line', figsize=(10, 6))

# Set plot labels and title
plt.xlabel('Year')
plt.ylabel('Total births per year')
plt.title('Total births per year by gender')

# Show legend and plot
plt.legend(title='Gender')
plt.show()
```



Exercise 20: Count number of gender neutral names (same for girls and boys)

```
#Ex.20
# Count the number of gender-neutral names
gender_neutral_name_counts = df.groupby('Name').filter(lambda x: len(x['Gender'].unique()) == 2)
unique_gender_neutral_names_count = gender_neutral_name_counts['Name'].nunique()
print(unique_gender_neutral_names_count)
```

```
10221
```

Exercise 22: Calculate how many years the observation was carried out

```
#Ex.22
# Calculate the number of unique years in the dataset
number_of_years = df['Year'].nunique()
print(f'The observation was carried out for {number_of_years} years.')
```

```
The observation was carried out for 135 years.
```

Exercise 23: Find the most popular gender neutral names (those present each year)

```python
#Ex.23
# First, group by 'Name' and 'Year' and check if the name appears for both 'F' and 'M'
gender_neutral_names = df.groupby(['Name', 'Year'])['Gender'].nunique().unstack()

# Filter names that appear for both genders in all years
popular_gender_neutral_names = gender_neutral_names[gender_neutral_names == 2].dropna(how='any').index

# Display the list of most popular gender-neutral names
popular_gender_neutral_names_list = list(popular_gender_neutral_names)

popular_names_df = pd.DataFrame(popular_gender_neutral_names_list, columns=[0])

print(popular_names_df)
```

```
          0
0   Francis
1     James
2      Jean
3     Jesse
4    Jessie
5      John
6   Johnnie
7    Joseph
8       Lee
9    Leslie
10   Marion
11    Ollie
12   Robert
13   Sidney
14   Tommie
15  William
```

Exercise 24: Find the most popular unpopular names (unpopular name that babies have been called the most times)

```python
#Ex.24
# Find names that were given a small number of times (e.g., minimum number of times)
max_count = df['Count'].max()

# Filter the dataset to get the names that have the minimum count
unpopular_names = df[df['Count'] == max_count]

# Now, find the name that has the maximum count among these "unpopular" names
most_popular_unpopular_name = unpopular_names.loc[unpopular_names['Count'].idxmax()]

print(f"{most_popular_unpopular_name['Name']} is the most popular unpopular name. This name was given to babies {most_popular_unpopular_name['Count']}
```

```
Linda is the most popular unpopular name. This name was given to babies 99680 times.
```

Exercise 27: Find the most popular names each year

```
#Ex.27
# Group by 'Year', find the name with the maximum count for each year
most_popular_names_each_year = df.loc[df.groupby('Year')['Count'].idxmax(), ['Year', 'Name', 'Count']]

# Set the 'Year' as the index for easier readability (optional)
most_popular_names_each_year.set_index('Year', inplace=True)

print(most_popular_names_each_year)
```

```
          Name  Count
Year
1880      John   9655
1881      John   8769
1882      John   9557
1883      John   8894
1884      John   9388
...        ...    ...
2010  Isabella  22883
2011    Sophia  21816
2012    Sophia  22267
2013    Sophia  21147
2014      Emma  20799

[135 rows x 2 columns]
```

4. Link to the Jupyter Notebook

You can access the full Jupyter Notebook containing the data analysis, code, and results on **GitHub**:

- **GitHub Repository**: https://github.com/BaharBerra/Python_lab3

5. Conclusion

In conclusion, this laboratory work provided hands-on experience in using the Pandas library for data analysis. The exercises covered various techniques such as data filtering, aggregation, and visualization. Through the analysis of the US Baby Names dataset, I gained valuable insights into trends and patterns in baby naming practices in the US, such as the rise and fall of certain names over time, and the prevalence of gender-neutral names.

This lab demonstrated how powerful and efficient Pandas is for handling large datasets, making it a valuable tool in the field of data analysis. The skills acquired through this lab will be useful in future projects involving data processing, cleaning, and visualization.

BAHAR BERRA UYAR

National Technical University

«Kharkiv Polytechnic Institute»