practice_exercises / Prac_Pandas_Questions.ipynb

vaksakalli Add files via upload

1 contributor

241 lines (241 sloc) | 7.73 KB

...

Pandas Exercises

This exercise is concerned with baby names in the US between the years 2000 and 2018. The exercise tasks include exploratory data analysis and some data visualization. With these exercises, you will get to practice with some of the most commonly used Pandas features for hand-on data analytics.

Data source: catalog.data.gov

Inspiration for this exercise: PhantomInsights@github

The data set is available as a CSV file named "baby_names_2000.csv" on GitHub here (https://github.com/vaksakalli/datasets).

Exercise 1: Place this CSV file under the same directory where your Jupyter Notebook file is.

Import Pandas as "pd" and NumPy as "np".

Read the data into a Pandas data frame called `df`.

Exercise 2: How many rows are there in this dataset?

Exercise 3: How many columns does this dataset have?

Exercise 4: What are the names of columns in this dataset? Format the output as a Python list, please.

Hint: Use the `columns` attribute of `df` and cast the result to a Python list.

Exercise 5: Display the first 10 rows.

Exercise 6: Display the last 10 rows.

Exercise 7: Replace the gender "M" with "B" for boy, and "F" with "G" for girl.

Exercise 8: What is the earliest year?

Exercise 9: What is the most recent year?

Exercise 10: How many unique names are there regardless of gender?

Hint: Use the `nunique()` function.

Exercise 11: How many unique names for boys?

Exercise 12: How many unique names for girls?

Exercise 12: How many unique names for girls?

Optional Exercises

Exercise 13: The *gender* variable is categorical with two levels: *B, G*. You will now define a new data frame called `pivot_df` by "spreading" these two values into two columns where the cell values will be the sum of all the counts over all the years. You need to have a unique name in each row. To be clear, your new data frame needs to have exactly three columns: *name, B, G*. This is called a **pivot table**. Once you define your new data frame, display the top 5 rows.

Hint: For this, you will need to use the `pivot_table()` function with the `np.sum` aggregation. In particular, you will need to run the line below.

```
pivot_df = df.pivot_table(index = 'name', columns = 'gender', values = 'c
ount', aggfunc = np.sum)
```

After that, you will need to run the `dropna()` and `reset_index()` functions.

Bonus: Get rid of the columns' name on the top left corner. That is, set the data frame's columns' `name` attribute to "None".

Exercise 14: How many unique names are there that are gender-neutral (that is, names used for both boys and girls)? For more meaningful results, use names where both boy and girl counts are at least 1000.

Hint: Use your new pivot table from above and assign the query result to back to `pivot_df`.

Exercise 15: For the pivot table, define a new column *Total* which is the sum of boy and girl name counts for each name. Set the new column's data type to an integer using the `astype()` function with the `int` option. Then display 10 randomly selected rows.

Exercise 16: What are the top 10 boy names?

Hint: First filter for boys in the `df` data frame (not `pivot_df`). Then use the `groupby()` function on the *names* column followed by the `sum()` function.

Keep in mind that when an aggregation function such as `sum()` is called an a `groupby` object, it is