HW 7 - Data with Pandas

Due October 31, 2024 at 11:59pm

In this homework assignment, you will explore how to use `pandas` with real-life datasets. You will perform data manipulation, aggregation, and visualization (which is super cool!). If you get stuck, please first reference the pandas documentation or google it! If you are really stuck, then don't hesitate to come to any instructor's office hours, ask questions on Ed or send an email.

1 Analyzing Air Quality Data

In this problem, you will analyze a dataset with pandas containing concentrations of fine particulate matter (PM2.5) across various countries and regions from the World Health Organization. The dataset is stored in a file called `global_air_quality.csv`, which you have been provided.

1.1 Load the Data

First, use `pd.read_csv()` function to load the dataset from the given file named `global_air_quality.csv` into a Pandas DataFrame. Print the DataFrame you have just created.

```
>>> # Your code here
    IndicatorCode
0          SDGPM25
1          SDGPM25
...
```

1.2 Make a New Column

The `FactValueNumeric` column contains the air quality information we want. With the values from the `FactValueNumeric` column, add a new column to your DataFrame called `PM25_Value`. Print the DataFrame containing only the `FactValueNumeric` and `PM25_Value` columns. They should have identical numbers.

```
>>> # Your code here
>>> print(df[['FactValueNumeric', 'PM25_Value']])
```

```
     FactValueNumeric  PM25_Value
0                10.01       10.01
1                10.02       10.02
...                ...         ...
```

1.3 Calculate Average PM2.5 Concentration

The `ParentLocation` column contains the continent location for each country. Group the countries by their `ParentLocation`. Calculate the average PM2.5 concentration for each continent and create a new DataFrame with these averages. Rename the column from `PM25_Value` to `Average_PM25`. Merge the average values back into the original DataFrame. Print the DataFrame with to show columns `Locations`, `Parent Location`,`PM25_Value` and `Average_PM25`.

```
>>>> # Your code here
>>> print(df[['Location', 'ParentLocation', 'PM25_Value',
'Average_PM25']])
                                              Location  \
0                                                Kenya
1                                  Trinidad and Tobago
...                                                ...
             ParentLocation  PM25_Value  Average_PM25
0                    Africa       10.01     29.812282
```

1.4 Analyze Data

Determine which continent (`ParentLocation`) has the highest average air pollution (concentrations of fine particular matter). Does this surprise you? Why or why not? Include these answer as a comment(s) in your code.

1.5 Save the Processed Data

Lastly, save the updated DataFrame with just the columns `Locations`, `Parent Location`,`PM25_Value` and `Average_PM25` as a new CSV file to your system. Please submit this CSV file alongside your Jupyter Notebook submission for this assignment.

2 Planets, planets, planets!

In this problem, you'll be working with a dataset called `planets`, which is part of an extensive data visualization package called `seaborn`. If you're interested in exploring more of `seaborns`'s datasets, you can find a wealth of them here for practice! But for now, we'll just focus on a dataset that's most relevant to the Python DeCal course: exoplanets! Your goal is to recreate two well-known plots in the exoplanet research community.

The `planets` dataset is based on NASA's continuously updated catalog of planets located outside of our solar system. After importing the dataset, you'll see that the columns contain information such as the year each exoplanet was discovered, the discovery method used, and key planetary characteristics like mass, orbital period, and distance from Earth. (Fun fact: Exoplanet research is relatively new, with the first exoplanet discovery occurring in 1992!) If you're curious, feel free to explore NASA's full exoplanet catalog here.

2.1 Create a Scatter plot

First, load the dataset using `sns.load_dataset('planets')` and store it in a Pandas DataFrame. Your task is to create a scatter plot that visualizes the relationship between orbital period (x-axis) and mass (y-axis) of all the exoplanets in the `planets` dataset. Color the data points according to the exoplanet's discovery method. Try using `seaborn` instead of `matplotlib` to generate this plot.

When submitting your scatter plot, be sure to include the following:

- A descriptive title
- Axis labels (with units!)
- A legend detailing the exoplanet color coding

Hint: Consider using a logarithmic scale to improve the data visibility. Logarithmic plots can reveal patterns in data with large value ranges and are very common in astronomy.

2.2 Create a Bar Chart

Next, create a bar chart where each **stacked** bar represents the total number of exoplanets discovered (y-axis) in each year (x-axis), grouped by the discovery method. For example, a bar for 2024 would have sections for radial velocity, mircolensing, imaging, etc. Each discovery method should have its own color. We highly recommend using `seaborn` over `matplotlib` for this bar chart.

Before you submit this plot, make sure it has:

- A descriptive title
- Axis labels (with units!)
- A legend explaining the exoplanet color coding

Hints: It is a good idea to remove missing values (NaNs) in the dataset before plotting. To create a stacked bar plot, you can use Seaborn's plotting functions along with the `hue` parameter to differentiate between discovery methods.