HW 8 - Pandas

Due April 1st, 2026 at 11:59pm

Overview

In this homework assignment, you'll practice working with `pandas` DataFrames and import data from the NASA Exoplanet Archive.

Note: If you need help downloading data from the NASA Exoplanet Archive, please reference **Section 6: Bonus Content** in the **[Class 16]** slides on bCourses.

1 Branching with GitHub

Create a branch called `homework8` in your `yourname` repository. Inside that branch, create a folder called **homework8**. Please work on this homework assignment on that branch and in this folder.

Take a screenshot of making this new branch, removing all of the files and adding/committing/pushing it to GitHub.

Name the screenshot: `hw8_branch`. Save it inside your `homework8/` folder in your `homework8` branch.

2 Pandas DataFrame

Learning how to analyze data with the `pandas` package is best done with actual data. To access the data for this problem, pull the latest version of the `main` branch of the `course_assignments` repository.

Steps:

1. Inside your `homework8/` branch, open your `homework8/` folder. Open a Jupyter Notebook and name it `homework8.ipynb`.

2. Import `pandas` at the top of your Jupyter Notebook.

3. Import the `state_data.csv` file as a `pandas` Dataframe with the `pd.read_csv()` function.

Hint: You don't need to copy the data from the `course_assignments` repository, just write a relative path.

4. Print the shape of the data you imported. How many columns are there? How many rows are there?

5. Use the `.head()` and the `.tail()` functions to print the **first five** and **last five rows** of the dataset. Answer the following:
   - Explain what the different columns represent.
   - What is the **first state** in the dataset?
   - What is the **last state** in the dataset?
   - How many entries are there total?
   - Why is there more than 50 states listed?

6. Compute the **median latitude** and **median longitude** of the entire dataset. What are the values?

7. Compute the **absolute value** distance to the **median latitude** for each state. Make a new column in your DataFrame with these values.

8. Compute the **absolute value** distance to the **median longitude** for each state. Make a new column in your DataFrame with these values.

9. Print the shape of your DataFrame with the added columns. How many columns and rows do you have now?

10. For the following print statements, please round your answer to two decimal places:
    - Print the state **closest** to the **median latitude** and by how much.
    - Print the state **farthest** from the **median latitude** and by how much.
    - Print the state **closest** to the **median longitude** and by how much.
    - Print the state **farthest** from the **median longitude** and by how much.

    Hint: You can round to two decimal places by attaching `:.2f` after your variables in f-strings. Example: `print(f"{variable:.2f}")`.

11. Go to the documentation website for Pandas:
    - On the website, look up a `pandas` function we have not used in class or homework.
    - Apply it to your DataFrame.
    - Explain where you found it on the website.
    - Explain what the function does and when it is helpful to use it.

3 Seaborn

Seaborn is a package that provides plotting functions similar to `matplotlib`. However, it also contains a lot of unique datasets.

Steps:

1. In the same Jupyter Notebook as the previous problem (i.e., `homework8.ipynb`), import both `matplotlib` and `seaborn`.
   Hint: To import `seaborn`, call:

   ```
   import seaborn as sns
   ```

2. Go to the `seaborn-data` GitHub repository.

3. Scroll through the different datasets and pick your favorite.

4. Import the dataset using the `sns.load_dataset()` function and store it as a DataFrame in your `homework8.ipynb` Jupyter Notebook.

5. In your dataset, drop all the columns with `NaN` values.

6. Answer the following questions by analyzing your chosen dataset:
   - What is the shape of your dataset?
   - How many rows are there? How many columns are there?
   - Print the **first five** and **last five** rows.
   - List each of the column names and describe what it represents.

7. Create custom subplots:
   - Create a figure with four subplots in a 2x2 grid.
   - Pick two columns. Scatter the data on the top left-hand subplot.
   - Pick two entirely different columns. Scatter the data on the top right-hand subplot.
     Note: If your data does not have enough columns, just switch the x- and y-axes.
   - Pick any column. Create a histogram in the bottom left-hand subplot.
     Hint: Use the `seaborn` function: `sns.histplot()`.
   - Pick a different column. Create a categorical plot (e.g., boxplot) in the bottom right-hand subplot.
     Hint: Use the `seaborn` function: `sns.boxplot()`.
   - Call `plt.show()` to finish the figure.

8. Add information:

- Add a title to each subplot.
- Add an x- and y-label to each subplot. Don't forget units!
- Add a legend to at least one of the subplots.
- Use different colors for your data in each subplot.

9. Explain:
   - Explain what data each of your subplots represents and why you decided on those columns.
   - What did you learn about your dataset from each subplot?

Note: For this problem, you may want to refer to the documentation website for `seaborn` *to understand how the different* `seaborn` *functions work.*

Note: If a ***logarithmic*** *scale makes your data easier to interpret, feel free to use it. Just add a comment.*

4 Final Project Idea: Analyzing Data

If you choose to analyze data for your Final Project, this problem will be of extra help for you. In this problem, we will walk through how to download data from the NASA Exoplanet Archive. This archive holds up-to-date data on exoplanet candidates, confirmed exoplanets and host stars.

For context, if you choose to analyze data for your Final Project you must also perform curve fitting. However, we will cover that in next week's homework.

4.1 Finding Data

Steps:

1. Navigate to the NASA Exoplanet Archive website.

2. Scroll down to the table called **Work with Data**. Click on **Planetary Systems**.

3. A window that looks like a large spreadsheet will open. Each row is a different exoplanet and each column is a different parameter. In the **Discovery Year** column, write **2019**.

4. At the top of the page, you will see a tab called **Download Table**. Click it. Use the default selected parameters. Click **Download Table**.

5. Rename the file `exoplanet_data.csv`. Save the file in your `yourname` repository, under your `homework8` branch in your `homework8/` folder.

4.2 Importing Data

Steps:

1. Create a new Jupyter Notebook under your `homework8` branch in your `homework8/` folder called `analyze_exoplanet_data.ipynb`.

2. Import `numpy`, `matplotlib` and `pandas` at the top of your Notebook.

3. Import your `exoplanet_data.csv` file as a `pandas` DataFrame. To avoid an error, include this argument: `comment = "#"`

4. Answer the following questions by analyzing your imported dataset:
   - What is the shape of your dataset?
   - How many rows are there? How many columns are there?
   - Print all the column titles.
   - List 5 column titles that you have no idea what they mean.

4.3 Reducing Data

Reducing data means to make your dataset smaller while keeping only the data you want to analyze. As you saw in the last section, you imported nearly **100 columns** of data. While it's great to have more data than necessary, we are going to reduce our dataset by dropping a lot of the columns.

Steps:

1. Go to the NASA Exoplanet Archive Documentation website.

2. Each of the column titles in your dataset are referred to as **Database Column Names** on this website. Find the following **Table Labels** with their associated variable names:
   - Planet Name
   - Host Name
   - Discovery Method
   - Spectral Type
   - Equilibrium Temperature [K]
   - Planet Mass or Mass*sin(i) [Jupiter Mass]

3. In your Jupyter Notebook, for each **Table Label**, leave a comment:
   - Stating the **Table Label**.
   - Stating the **Database Column Name**.
   - Stating the description of the parameter.

- Describe what you think the parameter means in your own words.
  Hint: Feel free to use Google to look up stuff you don't know.

4. Create a smaller dataframe with only the **Database Column Names** from above.

5. Filter your smaller dataset by dropping any rows that contain `NaN` values.

6. Answer the following questions by analyzing your smaller, filtered dataset:
   - What is the shape of your smaller dataset?
   - How many rows are there? How many columns are there?
   - Print the first five and last five rows.
   - Print all the column titles.

4.4 Analyze Data

Steps:

1. Use the `.value_counts()` function to analyze the different discovery methods:

   - What are the different discovery methods?
   - How many are there of each? Which one has the most? Which one has the least?
   - Look up each discovery method. Define/describe it in your own words.

2. Make a scatterplot of the planet mass and equilibrium temperature:
   - Use a `figsize` of (8, 6).
   - On the x-axis, plot **Planet Mass**.
   - On the y-axis, plot **Equilibrium Temperature**.
   - For each discovery method, use a different color.
   - Add a title, axis labels and a legend.
   - Finish the plot with `plt.show()`.

3. Duplicate your previous plot but change both axes to a **logarithmic scale**.
   Hint: Call `np.log()`.

4. Based on your plot, are exoplanets found by the imaging method more likely to be massive or small?

5. Find the coldest planet with a mass larger than **five** Jupiter Masses.

6. Print the entire row for this planet:
    - What is it's name?
    - What index is it at?
    - What is it's equilibrium temperature?
    - What is it's mass (in Jupiter masses)?
    - What type of star is it orbiting? Is it a sun-like star?
      Hint: Is the spectral type of the host star similar to the Sun?

7. Duplicate your **logarithmic** plot:
    - Re-scatter the coldest planet.
    - Make it a different color and marker than the rest of the planets.
    - Make it larger than the rest of the data points.
    - Add a label (with its name) for your legend.

4.5 Check Your Answer

Nature is a highly-respected peer-reviewed scientific journal.

Steps:

1. Check out this Nature paper from 2024.

2. Answer the following:
    - What is the title of this paper?
    - Summarize the abstract to the best of your ability in your own words.
    - Does your answer match the exoplanet?
    - What is special about the planet you found?
    - Do the mass and equilibrium temperature match what the paper reported? If not, what are the differences?

5 Submitting Your Homework

Before submitting your repository, make sure that you have fully run your Jupyter Notebooks. We want to be able to see the outputs when we are grading!

Steps:

1. In your `yourname/` repository, run:

```
git add .
git commit -m "done with hw8"
git push origin homework8
```

2. Take a screenshot of the terminal output.
3. Name the screenshot: `hw8_changes`.
4. Place it inside your `homework8/` folder.
5. Ensure all screenshots are saved correctly and your code runs without errors. Your `homework8/` folder should now contain:

```
homework8/
|--- homework8.ipynb
|--- analyze_exoplanet_data.ipynb
|--- hw8_branch.png
|--- hw8_changes.png
```

6. Go to Gradescope and find the **Homework 8: Pandas** assignment.
7. Select the option to upload a GitHub repository.
8. Make sure to upload your `homework8` branch.
9. Submit your `yourname` repository.

Great job!