HW2.ipynb
IPYNB 193.54KB

Pretend you are a faculty looking at this assignment, can you help me write a code review? can you please write a short snippet for each sections and an overall comment. Give your answer short and concise. Please do not include the original code, just your review is good enough.

Overall Comment

The notebook demonstrates a structured approach to data preprocessing and visualization, with clear section divisions. However, there are several issues related to code efficiency, redundancy, error handling, and adherence to best practices. The visualization is well-executed, but the preprocessing could be significantly improved with better use of pandas and more robust validation.

Section Reviews

1. Data Preprocessing

- The method for reading the CSV is error-prone due to `skiprows` and `on_bad_lines='skip'`. Consider using more robust parsing with explicit column names and error handling.
- Non-numeric entry checks are done manually. Use `pd.to_numeric` with `errors='coerce'` more effectively to avoid redundant steps.
- Outlier detection uses IQR but doesn't account for all edge cases. Consider using z-score or more advanced methods for better accuracy.

2. Handling Non-Numeric Entries

- The approach is manual and inefficient. Use vectorized operations to detect and remove non-numeric rows in one go.
- Avoid creating temporary columns like `checkEruptions` and `checkWaiting`. Instead, use boolean indexing directly.

3. Extra Variables

- The extra variables are read separately, which is acceptable. However, merging with the main dataset could be streamlined.

4. Outlier Detection

- The IQR method is applied correctly, but the code is verbose. Use boolean indexing without creating an extra `outliers` column.
- Outlier removal is done via index exclusion, which is fine, but ensure indices are correctly managed after concatenation.

5. Data Visualization

- The violin and box plots are well-crafted and informative.
- However, the code is lengthy and could be simplified using seaborn for combined plots.
- Summary statistics are computed manually; consider using `describe()` for brevity.

6. Code Style and Efficiency

- Avoid redundant operations (e.g., multiple reads of the same CSV).
- Use functions to encapsulate repetitive tasks (e.g., outlier detection, cleaning).
- Prefer vectorized operations over loops for better performance.

Suggestions for Improvement

- Use `pd.read_csv` with `dtype` and `error_bad_lines` for better control.
- Leverage built-in pandas methods (e.g., `describe` , `isna`) for cleaner code.
- Replace manual statistic calculations with `describe()` .
- Consider using seaborn for more concise and advanced visualizations.
- Break down large code blocks into functions for reusability and clarity.