Jupyter Notebook Assignment
Instructions

Important Note

1. This assignment requires PyCharm Professional. Jupyter Notebook files are not supported by other versions of PyCharm.

2. Before starting this assignment, you should have already played *Tutorial Video: Python Program Design for Jupyter Notebooks.*

3. This is a data cleaning workflow assignment using just plain Python and Jupyter Notebooks. While we are aware that there are many data cleaning features of data preparation and analysis packages, we don't want you to use them for this assignment. Please follow the directions below to avoid grading deductions.

Overview

In this assignment, we will be creating a Jupyter notebook within a PyCharm project. When complete, the notebook will be an example of how to use a Jupyter notebook for a data cleaning workflow. Several Python programs will be imported into the notebook and run. One of the Python programs is provided as a starter file and is complete. Another Python program is provided as a starter file, and it is nearly complete. You will need to create yet another Python program from scratch. Details regarding the starter files and more detailed instructions are provided below.

Tools

You are expected to use the tools that are demonstrated in the tutorial videos: Anaconda, PyCharm, and Jupyter.

Tool Versions

Use the versions of tools that we installed during Week 1 of the course. When working on this assignment, you should use the Anaconda virtual env that we created called *e4_trainor_python_course.*

Starter Files

There is a .ZIP file of starter files for this assignment that you should download from the link provided in the Weekly Schedule. The zip file includes a *data* directory with the following files:

- short_raw_data.csv
- empty_raw_data.csv
- completely_empty_raw_data.csv
- raw_data.csv

The zip file also includes the following program files:

- count_city_name_values.py
- clean_data_coding_errors.py
- data_cleaning_workflow_example.ipynb

Tutorial Sequence

This assignment is organized as a multipart tutorial. Play each of the tutorial videos listed below using links provided in the Weekly Schedule for this assignment:

Tutorial Part 1: Getting Started and Creating Value Counting Programs

Exercise 1 (Regular)

Following the directions provided in the tutorial video, install c*ount_city_name_values.py* (provided as a starter file) into your PyCharm project and fully unit test the program.

Exercise 2 (Regular)

Using the c*ount_city_name_values.py* program as a model, create a program named *count_state_name_values.py* that provides the same functionality for State Name fields. Use the same approach to unit testing that was used for Exercise 1.

Tutorial Part 2: Create Data Cleaning Program

Exercise 3 (Regular)

Following the directions provided in the tutorial video, install *clean_data_coding_errors.py* (provided as a starter file) into your PyCharm project and fully unit test the program.

Note that the code provided only cleans the City Name fields. As a second step, you need to modify this program so that it uses the same approach used to clean City Names to also clean State Names. Please remember to fully unit test this second version of the program.

Please note that we will not be cleaning the Quantity field.

Tutorial Part 3: Create Jupyter Notebook

Exercise 4 (Regular)

Following the directions provided in the tutorial video, install *data_cleaning_workflow_example.ipynb* (provided as a starter file) into your PyCharm project.

Follow the directions in the tutorial video to populate the *Requirements* and *Overview* documentation cells with the following content:

> This notebook requires the Anaconda virtual env that was created during the first week of this course and named *e4_trainor_python_course*.
>
> No notebooks are expected to be run before this one.
> While at least one data analysis notebook is expected to be run after this notebook, it has not yet been developed.

Follow the directions in the tutorial video to populate the *Overview* documentation cell with the following content:

> This sample workflow cleans a raw data .CSV file to produce a cleaned data file.
>
> The .CSV file contains a header row.
>
> Each data row contains three fields separated by commas (City, State, Quantity). We suspect coding errors in the City and State fields. We will not be cleaning the Quantity field.
>
> This workflow has the following parts:
> - Count City Values (with raw data)
> - Count State Values (with raw data)
> - Correct Data Coding Errors
> - Count City Values (with cleaned data)
> - Count State Values (with cleaned data)

Follow the directions in the tutorial video to complete the Jupyter Notebook. Some of the coding will be shown explicitly in the video. You will need to create the rest of the code yourself using the design and style provided in the parts that were shown.

Remember to test the Jupyter Notebook when you are finished. Make sure that programs that are imported into the notebook only run when called. They should not run as a result of being imported. Also, make sure that your notebook runs properly when run from beginning the end. Note that this is testing by inspection where you run the notebook and inspect the output. It is not automated testing.

You should expect the following value counts for City Names after the data have been cleaned:

> Arlington: 55
> Bristol: 48
> Centerville: 52
> Chester: 45
> Clinton: 42
> Dayton: 41
> Dover: 53
> Fairview: 57
> Franklin: 59
> Georgetown: 48
> Greenville: 46
> Lebanon: 48
> Madison: 58
> Milton: 45
> Newport: 49
> Oakland: 52
> Salem: 46
> Springfield: 53
> Washington: 51
> Winchester: 52

You should expect the following value counts for State Names after the data have been cleaned:

California: 95
Florida: 119
Georgia: 97
Illinois: 94
Michigan: 98
New York: 119
North Carolina: 104
Ohio: 94
Pennsylvania: 85
Texas: 95

Exercise 5 (Challenge)
Create one further documentation cell at the bottom of the notebook with the title *Possible Enhancement: Clean Quantity Field.*

In prose, describe a reasonable approach to enhancing this project to include cleaning of the Quantity field. Address the following issues:

- How might we determine a proper upper limit value for Quantity?
- How might we determine a proper lower limit value for Quantity?
- What value might we use (if any) to replace entries over the upper limit?
- What value might we use (if any) to replace entries under the lower limit?
- How would we treat entries that were not numeric (like “Hi, Mom!)?
- Should we consider just dropping records that don’t fall within our standards?

Code Deliverables

You are expected to submit a properly organized PyCharm project that is ready to be tested using Anaconda, PyCharm, and Jupyter. Please refer to the tutorial video for details.

Non-Code Deliverables

Your PyCharm project must also include a data subdirectory that contains all of the data files needed to test your work.

File and Directory Naming

Please name your Python program files as instructed in each tutorial video. Please use the following naming scheme for naming your PyCharm project:

surname_givenname_jupyter_notebook_assignment

If this were my own project, I would name my PyCharm project as follows:

trainor_kevin_jupyter_notebook_assignment

Use a zip utility to create one zip file that contain the PyCharm project directory. The zip file should be named according to the following scheme:

surname_givenname_jupyter_notebook_assignment.zip

If this were my own project, I would name the zip file as follows:

trainor_kevin_jupyter_notebook_assignment.zip

Due By

Please submit this assignment by the date and time shown in the Weekly Schedule.

Last Revised
2025-02-10