README

The processing done to get the resulting CSV tables has been specified in both the ipython notebook `summary-dataset-generator.ipynb` and the script `summary-dataset-generator.py`.

However, it'll be easier to follow along the reasoning behind the transformations steps by following the ipython notebook. I have listed in steps in brief at the beginning of the `ipynb` file.

The script does the exactly the same thing but makes it runnable on the command line with arguments. Both of them make use of `utils.py` which contains the functions I have used in both the files.

Output ready to view

I have listed the steps below to run both the script and ipython notebook running. However, for some reason, if you want to take a look at the output I have already generated, you look into the files `ready-output/summary-table-generator.html` on a browser. The CSVs that were generated are available in `result` directory. It contains the `summary_table.csv` and `fullDataJoined_table.csv` files.

Steps to run the `summary-dataset-generator.py` script

1. Install docker on your machine.
2. Run the command `docker run hello-world`. It should show a message saying "Hello from Docker" and confirming that your Docker installation is working correctly.
3. Next, download the image I prepared for this challenge to be able to run the script and notebook I have shared in this zip file. It can be accomplished by running `docker pull vipulnj/bird-collision:1.1`.
4. Once we have pulled the docker image, we need to create a container. To do that, run

   ```
   docker container run -d -p 8888:8888 \
   -v /Users/username/Downloads/ta-asgn/:/ta-asgn \
   --name bird_collision_cntnr vipulnj/bird-collision:1.1
   ```

   If you are already running something on port 8888 on your machine, then you change the part after `-p` flag from `8888:8888` to `9000:8888`. Make sure to change to the `username` after the `-v` flag to your computer username. If you change the location of the unzipped `ta-asgn` folder from the Downloads folder to elsewhere, you need to change the location before `:` and after `-v` accordingly.
5. Next, we verify if the docker container is running. To do this, we run `docker container ls`. You should see the container with name you assigned i.e. `bird_collision_cntnr` shows the `STATUS: Up <X> minutes`.
6. Now, to run the script, we need to "enter" into the running container so that we can make use of the exact library versions that I worked with for developing the script. To do this, we run

   ```
   docker exec -it bird_collision_cntnr /bin/bash
   ```

   You should should see something like

   ```
   root@fb6d0460d765:/tmp#
   ```

   The `fb6d0460d765` is the container id and it should be different on your machine. We have now entered the container.
7. Next, we need to navigate to the directory which has been mapped to the location on your local machine and check its contents. For that, we run `cd /ta-asgn && ls` Running it should give something like

   ```
   root@fb6d0460d765:/tmp# cd /ta-asgn/ && ls
   README.md     result
   __pycache__   summary-table-generator-script.py
   data          summary-table-generator.ipynb
   ready-output  utils.py
   ```
8. To generate the summary dataset, we now need to run

   ```
   python summary-table-generator-script.py data result
   ```

   Here, `data` is the directory where the json files reside and `result` is where the generated CSV files are written. $\qquad$ You can specify other directories as input and output directories in the same order. Be sure to place the JSON files in the input directory.
9. You should find the generated CSV files in the `result` directory. The `summary_table.csv` contains the summarized data about every type of bird. The `fullDataJoined_table.csv` is the table just before we apply the group-by operation to get `summary_table.csv`

Steps to run the `summary-dataset-generator.ipynb` file

Runs steps 1 to 6 as mentioned above. Then, follow these steps:

7. Now, instead of running the python script, we can start the jupyter-notebook server. We do so by running the following command:

```
jupyter-notebook --ip=0.0.0.0 --allow-root --no-browser
```

8. This should start a jupyter server showing something like `http://127.0.0.1:8888/?token=f9d827f336eb19f4925f7b797625dbb00759df6c97547e9b`. The token you see could be different than the one listed here. Copy this address and paste it in the browser.
9. This should display the files in the directory. Click on `summary-dataset-generator.ipynb` to open it.
10. From there, you should be able to run each cell until the end to get the CSV files as the output.

You can also follow steps 7 to 10 here to view `interesting-facts.ipynb`

Observations and critique of the datasets

- The `light_levels.json` file does not provide all the dates where collisions happened. This leaves us with many nulls values in `fullDataJoined_table.csv` and therefore null values in `summary_table.csv` for species which had no valid values.
- The `light_levels.json` file also repeats a lot of dates. A same date had multiple values of Light Score. Due to this, one of the values had to be picked before we joined tables. Not doing so, would result in more rows in the resulting table after the left-join than the left-table itself.
- The other two JSON files `chicago_collision_data.json` and `flight_call.json` did not have NULL values in them. Although, `flight_call.json` did not contain information about all Genus-species pairs found in `chicago_collision_data.json`. This was not a major problem since information about this was available in the reasearch paper shared and its supplmentary information PDFs. Searching Google also brought up this information.
- `flight_call.json` incorrectly named columns. Apart from the ones mentioned in the problem statement, I found one more after going through the data that the column `Flight` should be more appropriately named `Collisions` which stands for number of collisions recorded.

Interesting facts found using the data

- Most of the collisions were recorded during the Spring and the Fall seasons. There is a drastic reduction in collision during the Summer and Winter months. This is in line with the common knowledge that most of the bird migration happens during Spring and Fall and therefore the collisions.
- Zonotrichia albicollis is the species that collides the most among the recorded cases. It makes nocturnal flight calls and this leads to the increased number of collisions, which is in agreement with the paper.
- Although the Passerellidae family birds are roughly 21% of the total species in the summary table, they account for 49% of all collisions in Chicago.
- More than two-thirds of the species identified in the dataset make a nocturnal flight call which might have lead to a collision.
- The number of collisions recorded at downtown Chicago (CHI) and McCormick Place (MP) are not very different.
- 96% of the collisions happened for a bird which makes a flight call.
- 76% of the collisions happened for a bird which flies lower than higher.