

Project: Turtles

Project Description

Thanks to Care for Hedland, DBCA and BHP, we have the data collected from the Hedland Turtle Monitoring Program on Cemetary Beach, for both the 2019/20 and 2020/21 seasons. Your job is to read, process, and display this data in a meaningful way.

Background Information

The Hedland Turtle Monitoring Program operates from November to March each year in correlation with the annual Flatback migration to the coastline for the nesting (November to January) and hatchling (December to March) seasons.

The program uses a non-invasive approach to obtain monitoring data through conducting early morning beach patrols assessing turtle tracks and sand patterns. Laid and hatched nest locations are recorded using a GPS and additional turtle and predator activity is documented. This data is collected between 6 and 7am almost every morning between October to March.

The columns in the data and what they represent are as follows:

Date Collected	The date the data was collected.
Nests	Number of nests identified.
Total No. False Crawls	Number of tracks detected that do not lead to a nest (when turtles come on to the beach with the intention of laying eggs, but have been scared off or are not happy with that spot).
Hit Rocks	Number of tracks detected that hit rocks (sometimes turtles will come up at the far ends of the beach, hit the rocks and go back into the water).
Hatched Nests	Number of tracks detected of hatchlings coming out of the nest into the water.
Nest predation	Number of nests that have been disturbed (usually by goannas).

Starter Code

Part One and Part Two of this project both have starter code provided (part_1.py and part_2.py). These files contain several functions, some are completed and some are yet to be completed (by you!). To help you understand what each function is meant to do, they each contain something called a "docstring". A docstring is a comment that describes exactly what the function does, including it's arguments and what it should return.

Let's look at the `read_csv_file` function as an example:

```
def read_csv_file(file_name):  
    '''Reads a csv file and returns the data as a list.  
  
    Args:  
        file_name: a string representing the path and name of a csv file.  
  
    Returns: a list.  
    '''  
    pass
```

The first line in the docstring gives an overview of the function. In this case, this function will read a csv file and return a list.

“Args” are the arguments/parameters that the function accepts. In this case, this function accepts only one argument called `file_name`, which is a string representing the path to a csv file, e.g.

`'data/2020_2021_turtle_data.csv'`.

“Returns” is what the function is meant to return when the function is complete. In this case, it should return a list.

So this docstring is telling us that this function should read a csv file, save the data to a list, and return that list.

Project Requirements

Part One

Your task is to read the `2020_2021_turtle_data.csv` file and output a text summary of this data (see appendix for what the summary should look like).

Starter code has been provided (`part_1.py`) to help you complete this task. If you choose to use this starter code, here are the suggested steps to complete the task:

1. Start by completing the `read_csv_file` function.
2. Complete the `transform_daily_to_monthly` function. Two helper functions have been provided to help you determine the month each row of data was collected in. These are:
 - `convert_mmdyyy_date` - to convert the date string to a datetime object (Python's way of understanding dates).
 - `get_month_name` - get the month name from a datetime object.
3. Complete the `output_nests_per_month_graph` function. You'll need to use the list created from the `transform_daily_to_monthly` function to get the number of nests recorded per month.
4. Complete the `output_monthly_statistics` function.
5. Complete the `output_overall_statistics` function.

Note: you can use the `format_text` function to create the columns in the text output.

Part Two (optional)

Your task is to read both the `2019_2020_turtle_data.csv` and `2020_2021_turtle_data.csv` files, and output a text summary and graphs for this data (see appendix for what the summary and graphs should look like).

Starter code has been provided (`part_2.py`) to help you complete this task. This task is very similar to part one, so you might find it useful to copy/paste some of your code from your completed `part_1.py` to `part_2.py`.

Part Three (optional)

Modify your `part_1.py` to `part_2.py` files to use the raw data files (in `data/raw/`). The data files used in part one and two were cleaned slightly for your convenience, the raw data files have inconsistencies in them, which your code will need to handle.

Submission

Please submit the following:

- A text file containing a link to your project repository.
- Create a screenshots directory in your repository, which will contain screenshots of the output for each part of this project that you have completed.

Appendix

Part One Expected Output

2020/2021 Flatback Turtle Migration at Cemetery Beach

Number of Nests recorded per month (X = 5 nests):

October: 14 XXX
 November: 144 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 December: 170 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 January: 10 XX
 February: 0
 March: 0

Monthly Statistics:

Month	Nests	Hatched Nests	False Crawls	Hit Rocks	Nest Predation
October	14	0	2	0	0
November	144	3	57	30	0
December	170	64	85	3	14
January	10	37	5	1	1
February	0	0	3	0	0
March	0	0	0	0	0

Overall:

Nests	338
Hatched Nests	104
False Crawls	152
Hit Rocks	34
Nest Predation	15

Part Two Expected Output

Overall:

	2019/20	2020/21
Nests	410	338
Hatched Nests	296	104
False Crawls	453	152
Hit Rocks	89	34
Nest Predation	0	15

