

Assignment 1 - Basic Pandas (Youtube stats)

The file “videos.csv” contain the data with two columns.

1. `video_id` -> id of video
2. `trending_date` -> The date when the video was trending.
3. `title` -> The title of the video.
4. `channel_title` -> The name of the channel that published the video.
5. `category_id` -> id of category to which the video belongs.
6. `publish_time` -> The date and time when the video was published.
7. `Tags` -> Tags associated with the video.
8. `views` -> The total number of views the video has received.
9. `likes` -> The total number of likes the video has received.
10. `dislikes` -> The total number of dislikes the video has received.
11. `comment_count` -> The total number of comments on the video.
12. `thumbnail_link` -> The URL of the video's thumbnail image.
13. `comments_disabled` -> A flag indicating whether comments are disabled for the video.
14. `ratings_disabled` -> A flag indicating whether ratings are disabled for the video.
15. `video_error_or_removed` -> A flag indicating whether the video has been removed or encountered an error.
16. `description` -> The description of the video, typically containing additional details provided by the uploader.

The file “category_id.json” follow this JSON structure.

JSON Structure Overview:

- `kind`: **String** (Indicates the type of the API response),
- `etag`: **String** (An entity tag used for caching purposes)
- `items`: **Array of Objects** (Contains a list of video categories)
Each object in a list ...
 - `kind`: **String** (Indicates the type of the item)
 - `etag`: **String** (An entity tag (ETag) for the item)
 - `id`: **String** (A unique identifier for the video category)
 - `snippet`: **Objects** (Contains detailed info. about the category)
 - `channelId`: **String** (The ID of the channel)
 - `title`: **String** (The name of the video category)
 - `assignable`: **Boolean** (Indicates whether the category can be assigned to a video)

Problem 1:

How many rows are there in the videos.csv after removing duplications?

Hint: In this function, you must load your data into memory before executing any operations. To access videos.csv, use the path /data/videos.csv.

Problem 2:

How many VDO's that contain at least one record (row) of "dislikes" more than "likes"?

Hint: This function receives a Pandas DataFrame in memory. Utilize this DataFrame to answer the question.

Problem 3:

How many VDO that are trending on 22 Jan 2018 with comments more than 10,000 comments?

Hint: This function receives a Pandas DataFrame in memory. Utilize this DataFrame to answer the question. The trending date in the DataFrame is represented in the format 'YY.DD.MM'. For example, January 22, 2018, is represented as '18.22.01'.

Problem 4:

Which date that has the minimum average number of comments per VDO?

Hint: This function receives a Pandas DataFrame in memory. Utilize this DataFrame to answer the question.

Problem 5:

Compare "Sports" and "Comady", how many days that there are more total daily views of VDO in "Sports" category than in "Comady" category?

Hint: This function receives a Pandas DataFrame in memory. Utilize this DataFrame to answer the question. However, you must load the additional data from category_id.json into memory before executing any operations. To access category_id.json, use the path /data/category_id.json.

Example

Process input and output from the dataset using `main.py` for each question

Input	Output
Q1 String	40901 int
Q2 String	122 int
Q3 String	28 int
Q4 String	18.05.02 String
Q5 String	83 int

** Disclaimer: The data used in the example, 'videos.csv', differs from the data used for scoring.
