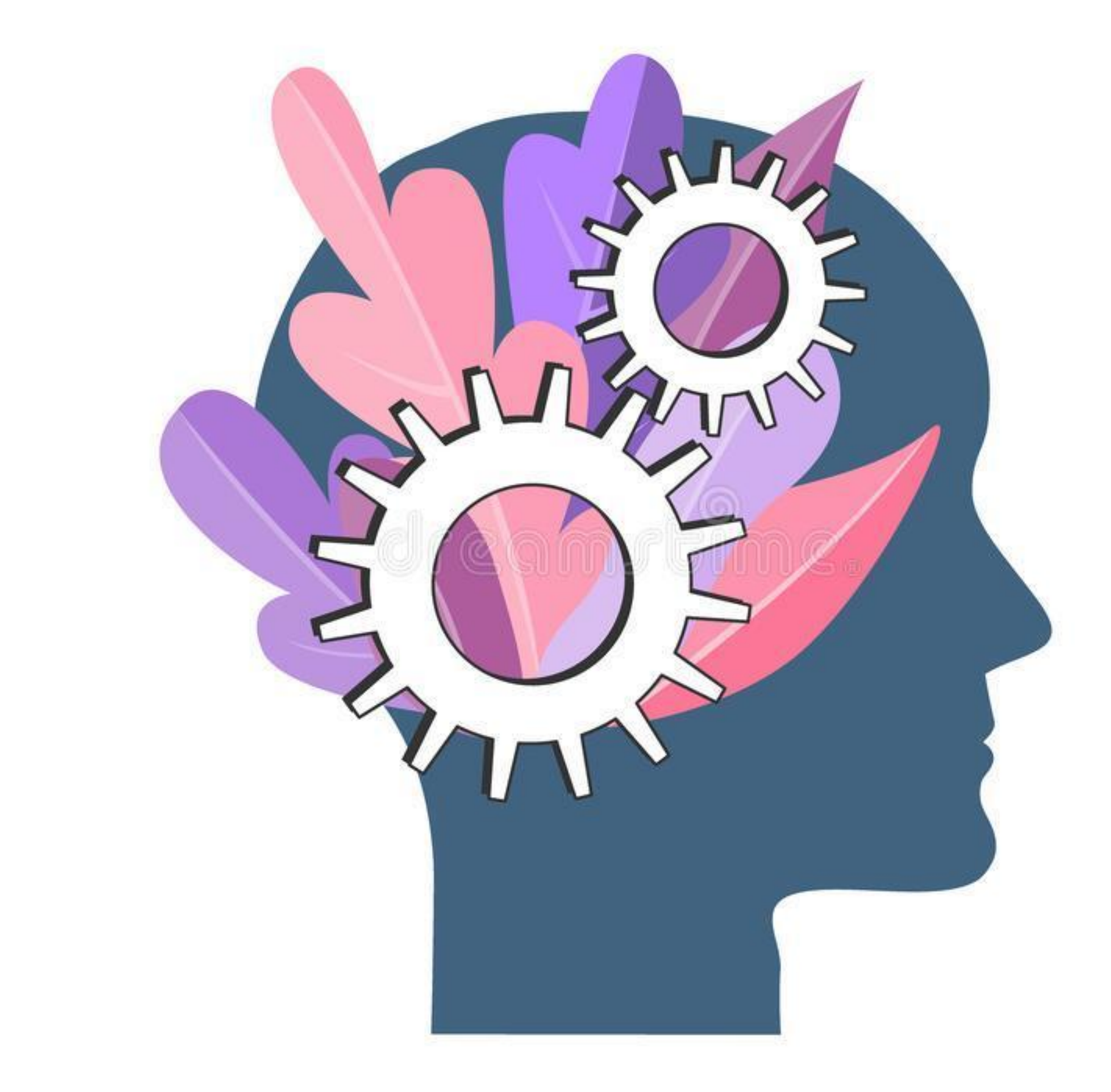
Work Culture Project

Updates

Feb 3 2023



Presenter:

- Mitanshi Vyas (MS Student)

Overview

- Last semester we worked towards collecting the data. This semester we are focusing more towards quantifying the data to figure out useful insights which will help us dig deeper into the working patterns of the workers.

- Analysis of different parameters like workers’ wages, their earnings per day, time taken by a worker to complete a task, EDA of logs table, EDA of workers wages.

- Visualizations to show what a particular worker is looking for in a job.

- Created python scripts containing various functions to calculate and fetch values.

Functions

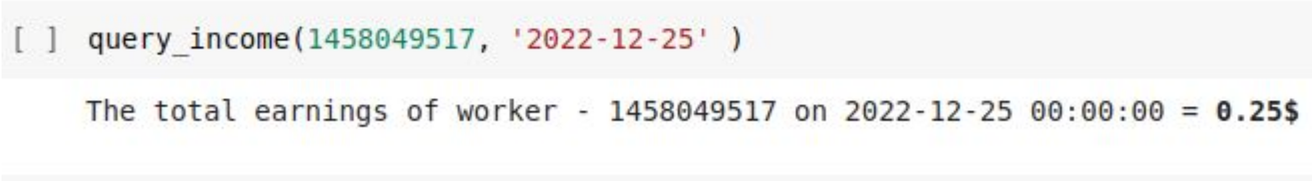
Below are several similar functions that I have created to fetch these insights with their respective description and parameters:

1. **query_income(worker_id, date)** : the function will take the user id and the date for which we want to query a particular worker's income.
2. **avg_scroll(worker_id)** : The function will find the average number of scroll counts. This function doesn't need any parameters.
3. **clicks_per_country(country)** : this function finds the average number of clicks per country. We take the country code of the country as a parameter.
4. **interactions_per_country(country)** : this function will find average number of interactions with tasks by users in that country
5. **interactions_per_task(pool_id)** : this function provides average total interactions per task
6. **start_time(pool_id)** : this function provides the timestamp when the person first visited the task link. The parameter is link to the task.
7. **task_duration(pool_id)** : this function returns the total duration that was taken to complete a task in days.

Analysis of Interactions

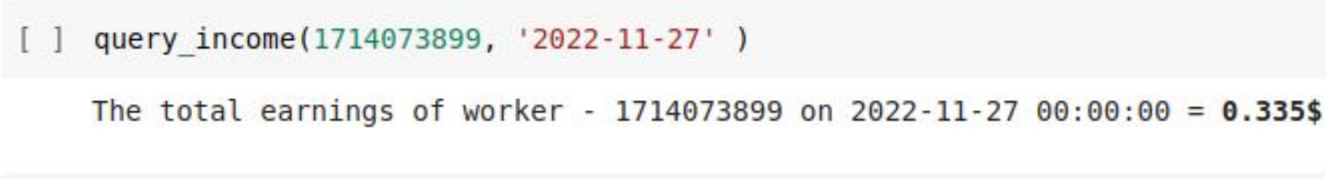
Function Name: query_income(worker_id, date)

Outputs:



```
[ ]  query_income(1458049517, '2022-12-25' )

     The total earnings of worker - 1458049517 on 2022-12-25 00:00:00 = 0.25$
```

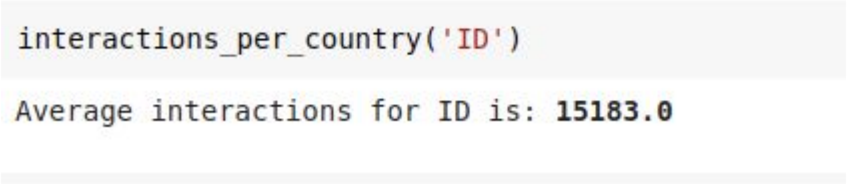


```
[ ]  query_income(1714073899, '2022-11-27' )

     The total earnings of worker - 1714073899 on 2022-11-27 00:00:00 = 0.335$
```

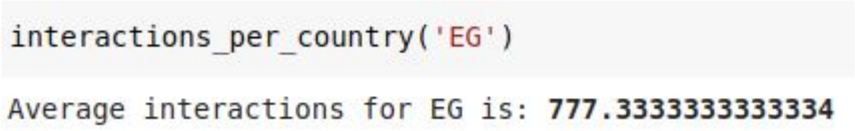
Function Name: interactions_per_country(country)

Outputs:



```
interactions_per_country('ID')

Average interactions for ID is: 15183.0
```

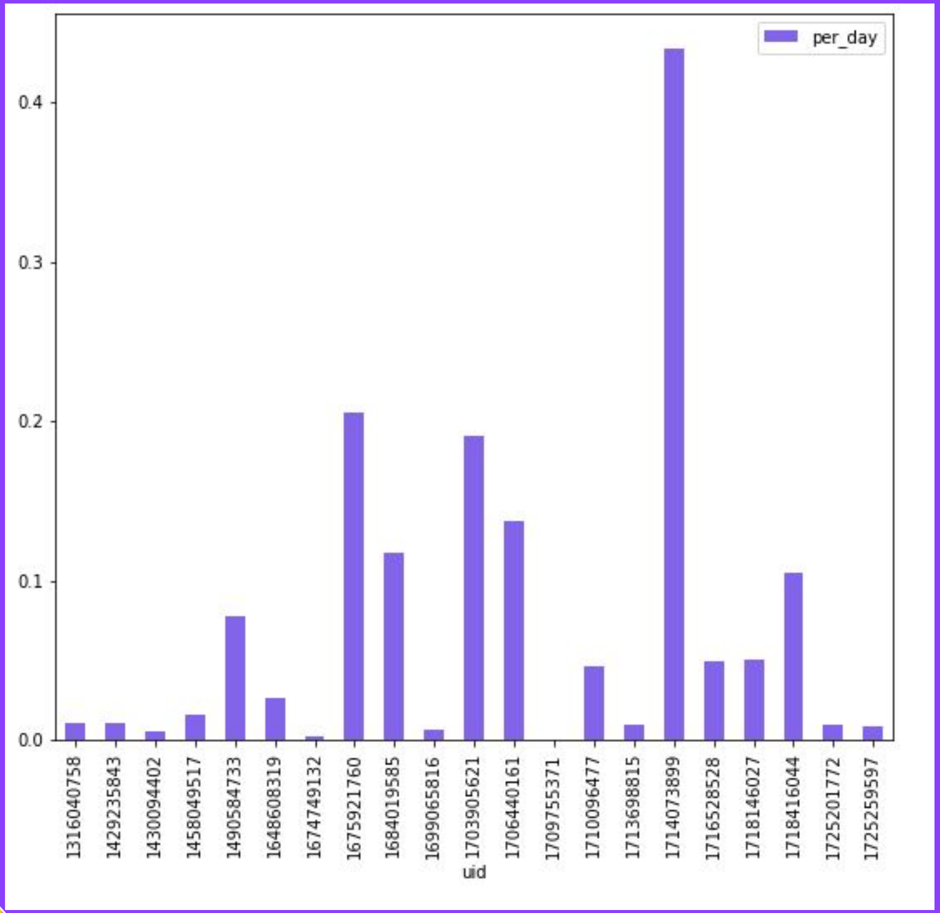


```
interactions_per_country('EG')

Average interactions for EG is: 777.3333333333334
```

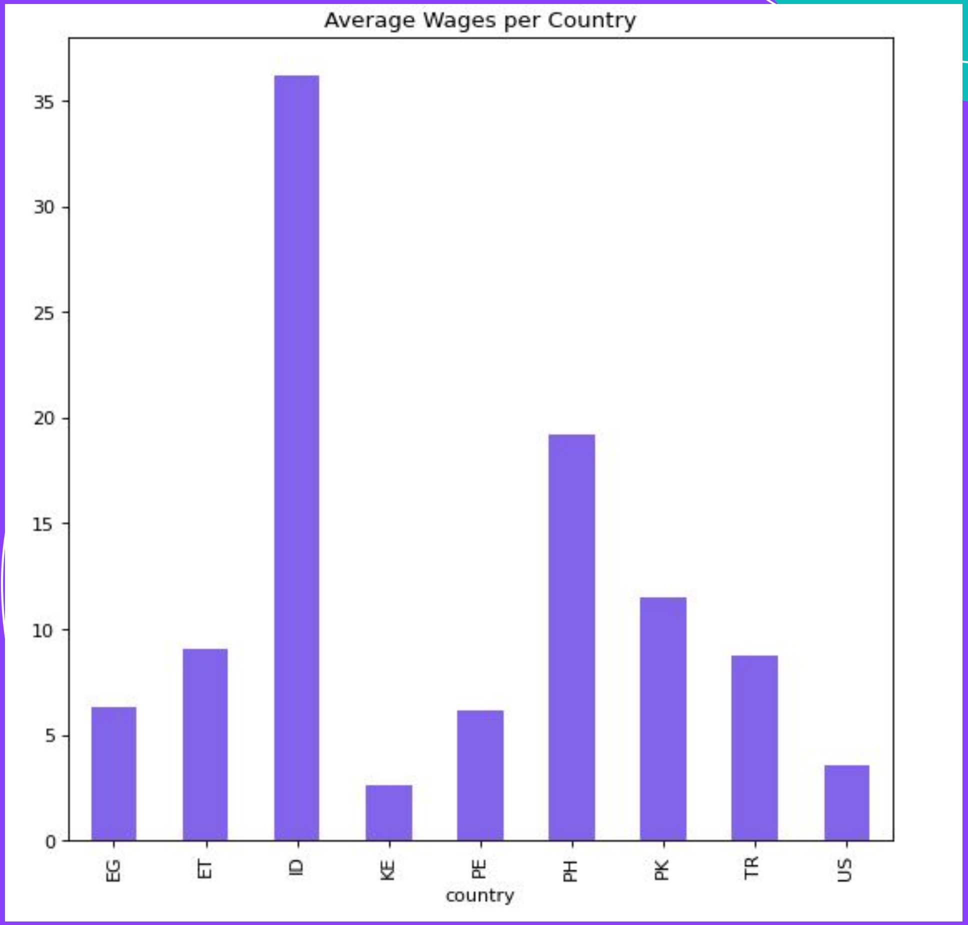
INSIGHTS

Distribution of earnings per day for each user



Worker 1714073899 has a significant higher earnings per day as compared to other workers.

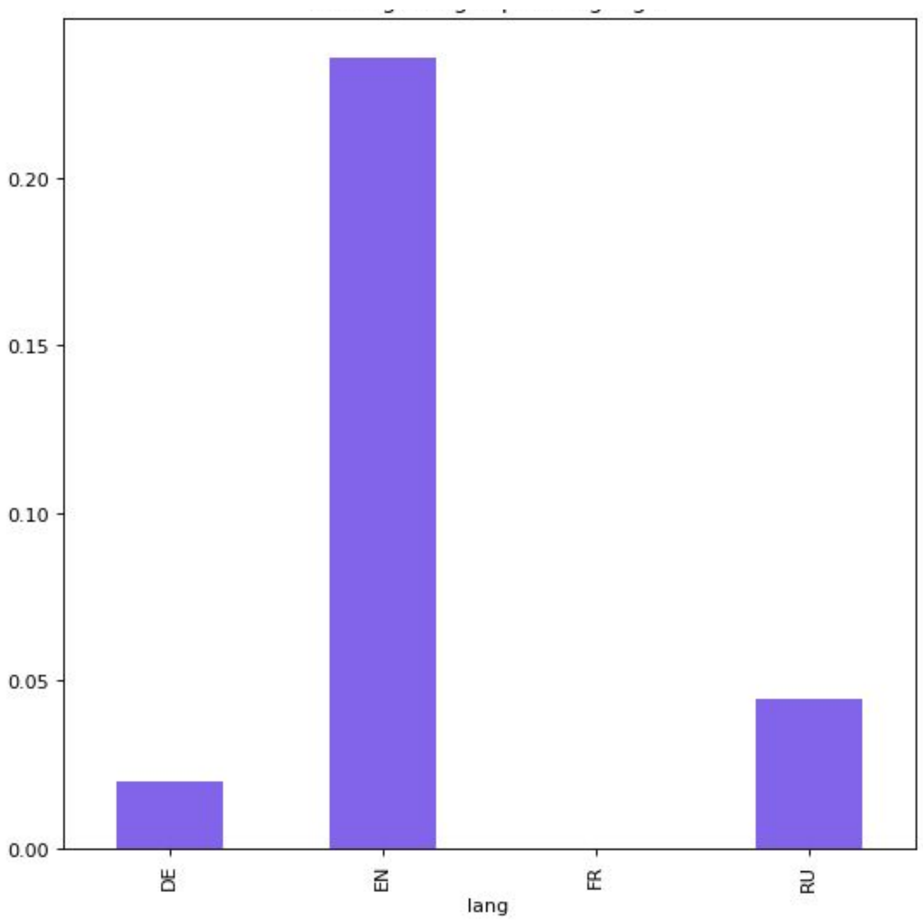
Distribution of workers wages for each country



Indonesia has the highest average wages for the workers as opposed to Kenya which has the lowest wages for the workers.

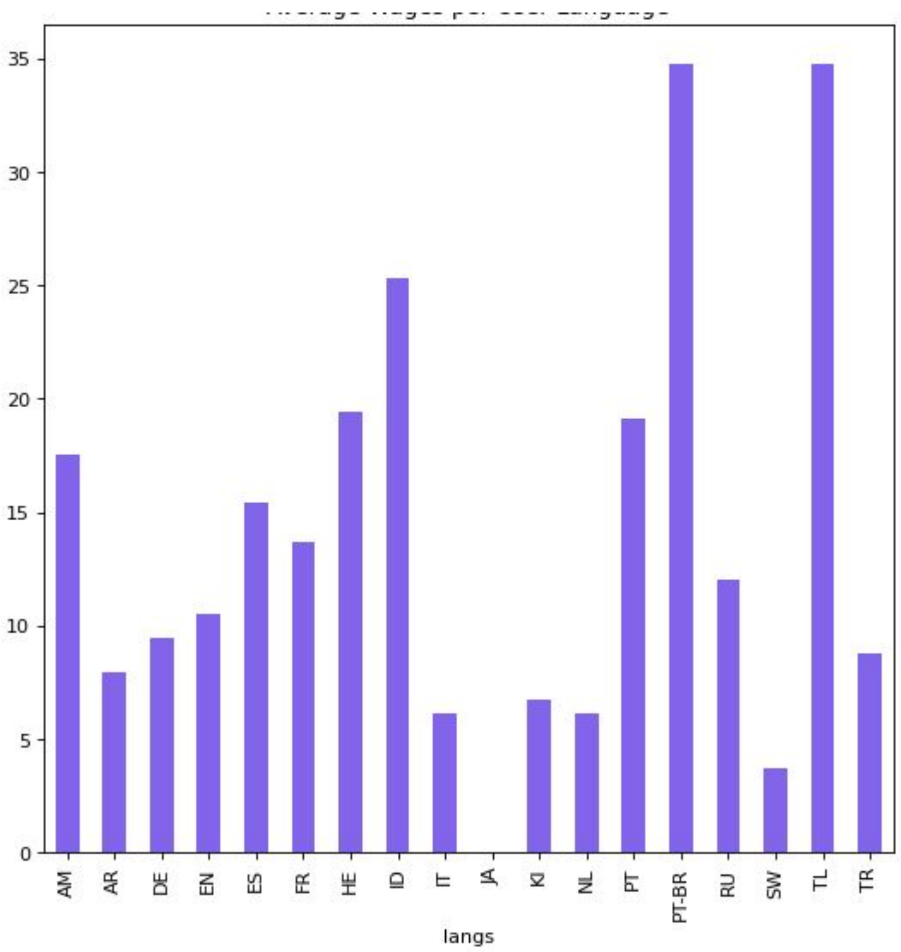
INSIGHTS

Average Wages as per the Language of the Task



English(task language) has the highest average wage for workers, whereas French has the lowest (almost negligible)

Average Wages as per the Language Known by the Worker



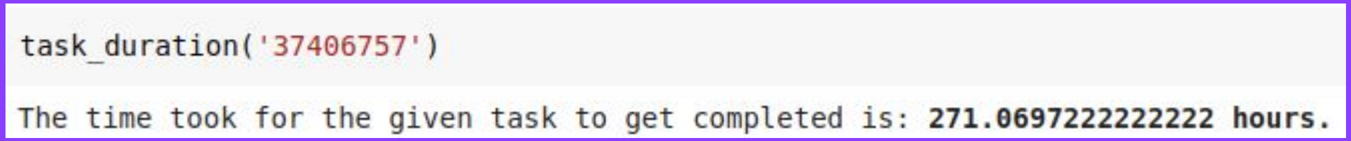
Tagalog(TL) language and Portuguese(Brazil) (PT-BR), both ties for the highest average wages for worker. While Japanese (JA) has the lowest average wages for the workers.

Task Duration Calculation

Approach we took:

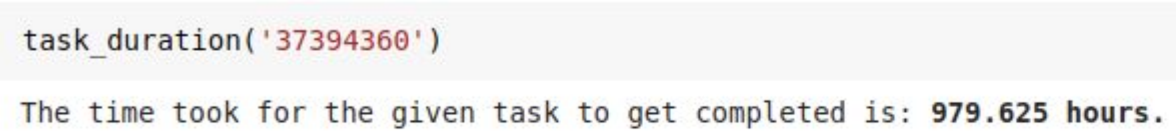
- We referred to the logs_data table to find the timestamps corresponding to each tasks’ url
- Calculated the task duration by taking the first timestamp and the last timestamp in the table corresponding to that tasks’ url

Following are the results we received with the above approach:



```
task_duration('37406757')
```

```
The time took for the given task to get completed is: 271.0697222222222 hours.
```



```
task_duration('37394360')
```

```
The time took for the given task to get completed is: 979.625 hours.
```

ISSUE:

- The results are very large as compared to what we would expect. The issue is because we were taking timestamps from the first time the worker clicked the task to the last time they did so, we are counting all the hours between those time periods
- However, this method is very inefficient and we need to rectify it as it has several loopholes
- What if the worker hasn’t completed the task after viewing it for the first time?
- We also don’t know if the task was even completed in the first place.

Our Future Approach

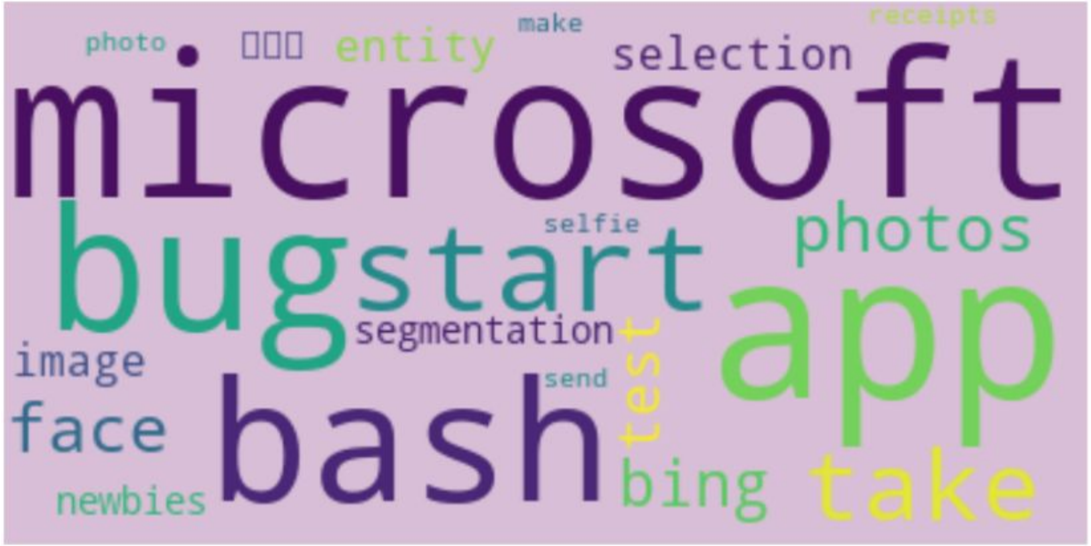
We decided on considering the following measures to improve the accuracy of task duration:

- We plan on integrating income data to check the task receipts and verify whether the task was even completed in the first place or not.
- If the task was completed, then we move forward to find the time taken to complete the task
- We are planning to consider ‘events’ column in the log_data table to make our outputs more precise
- It contains values like ‘TABOPENED’, ‘TABCLOSED’, ‘TABUPDATED’, etc.
- We are planning to integrate these values to find small windows of time for which the worker was active on a task window
- Finally, summing all those durations to find the task duration closest to the actual time taken by the worker to finish the task

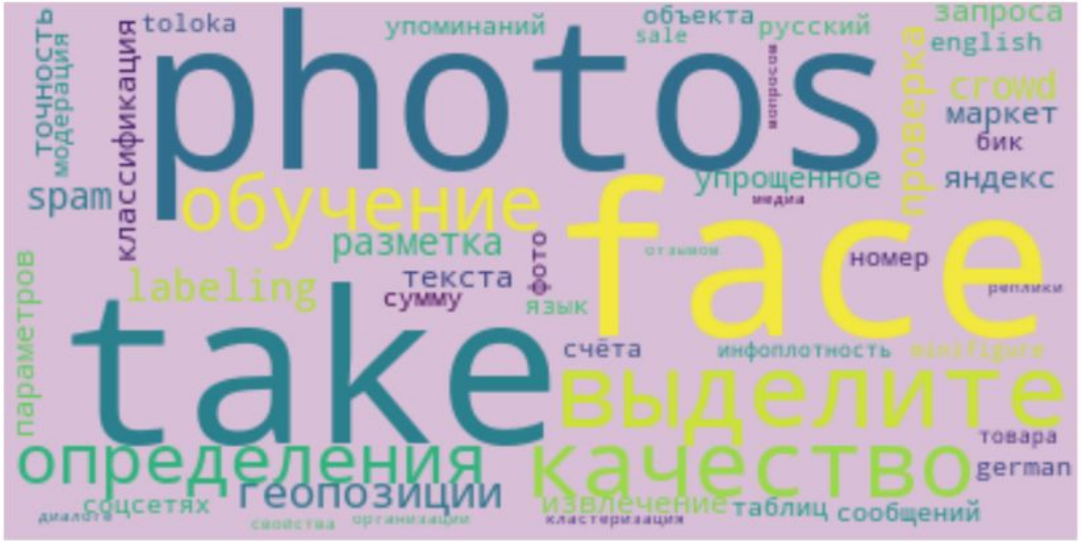
WordCloud

Visual Representation of Words that a particular worker look for the most in Titles of the Projects

WordCloud for worker: 1316040758



WordCloud for worker: 1675921760

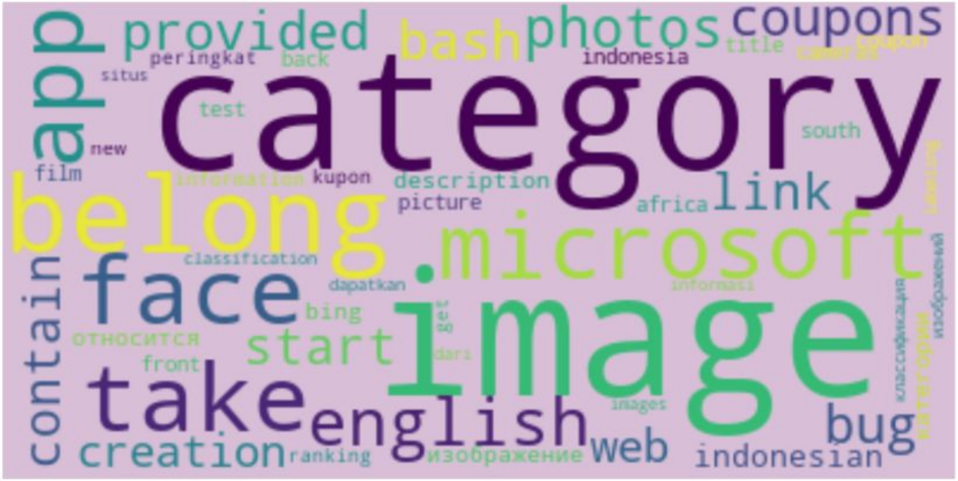


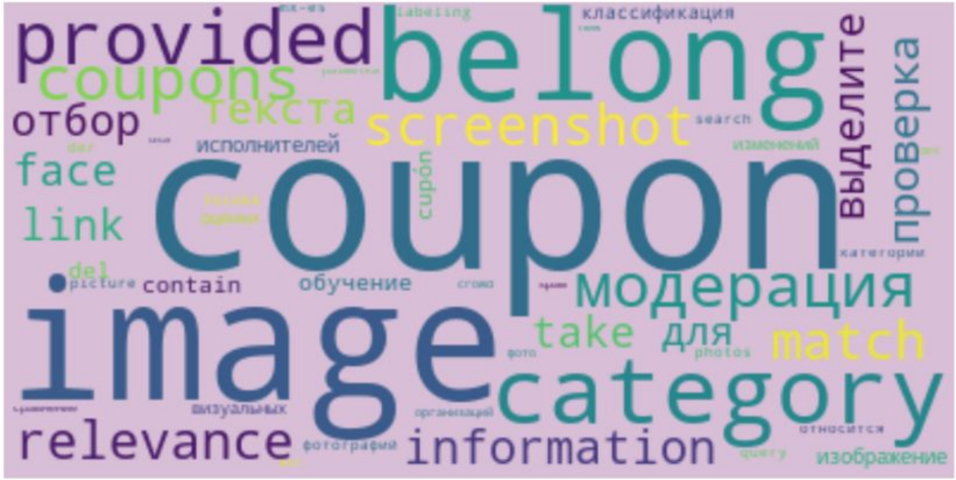
From the above visualization, we can conclude that worker: 1316040758 is more likely to work on projects associated with building apps, debugging, microsoft associated projects. Also projects that require capturing pictures (recurring words like: photos, image, selfie, face, etc.)

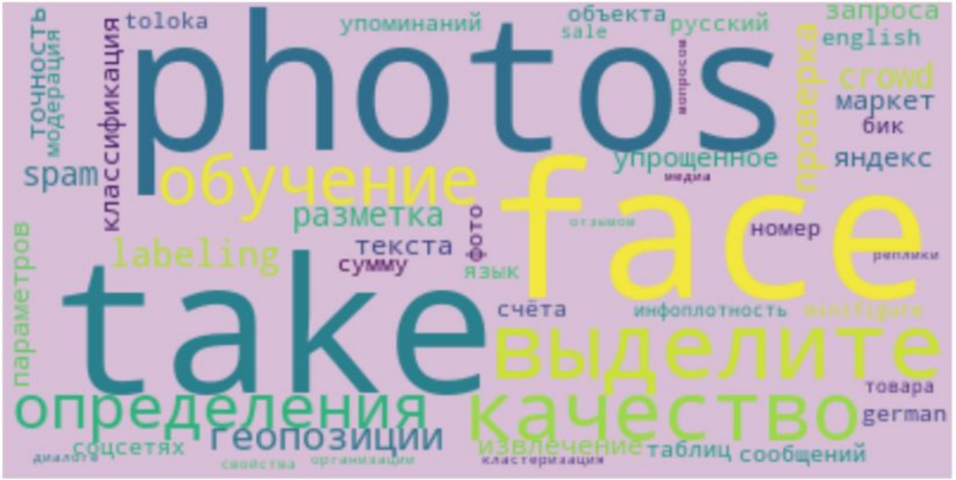
From the above visualization, we can conclude that worker: 1675921760 is more likely to work on projects associated with taking pictures, labeling data. We can also see that they're more likely to take on projects in English and Russian language.

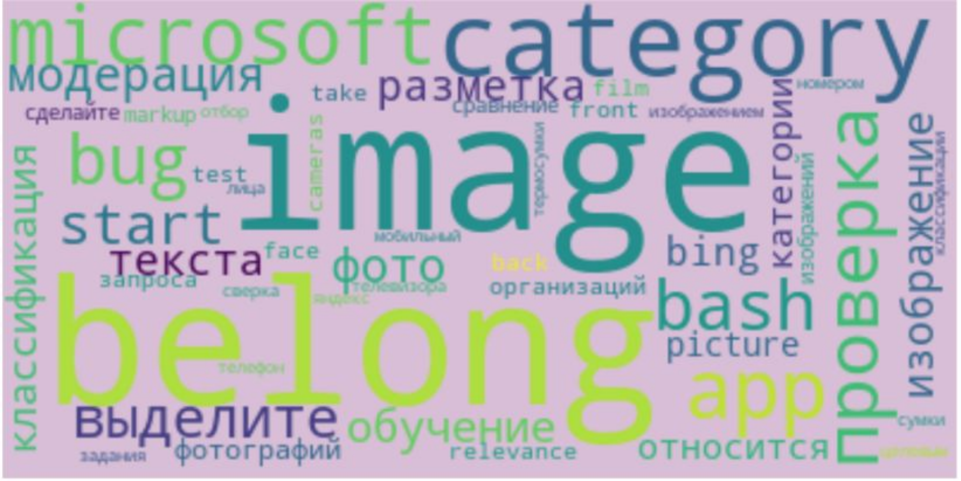
Similarities between WordClouds

- In the following wordclouds, we can see that these users are very likely to select the tasks which includes words such as “Image”, “Category”, “Photo”.
- This implies that these tasks are in high demand and are the top most picks of these workers.









Plans for the Next Week:

- Working on the task_duration function to make it more efficient.
- Building a classification model to help classify as a new task as to be potentially completed by a particular user or not.
- Digging deeper into the log data:
  - ➢ Given pool_id and user_id, extracting interactions
  - ➢ How the workers interact with Toloka Tasks tab
  - ➢ Duration of different time windows they spend on Toloka Task tab

Appendix

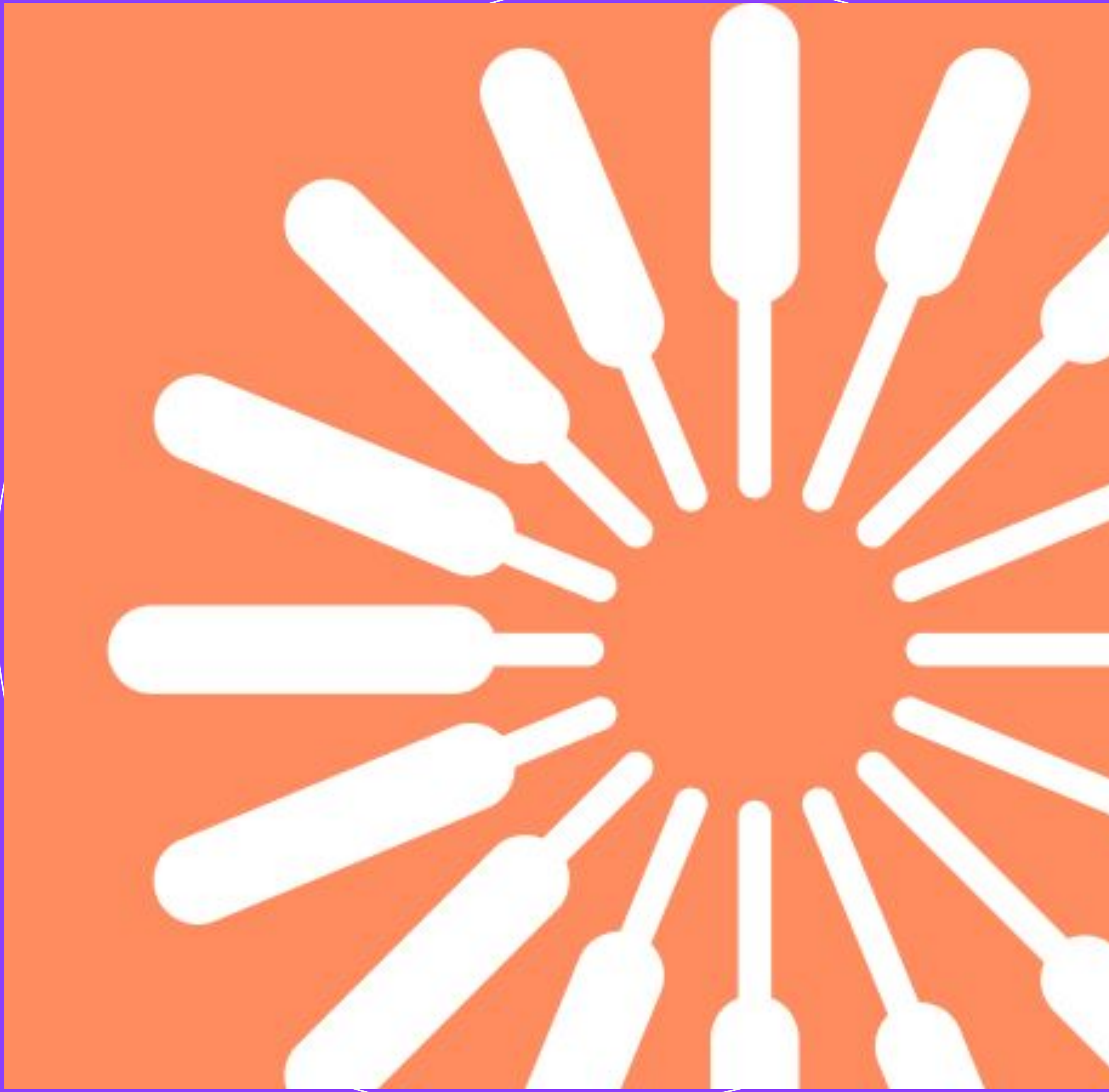
Python Scripts:

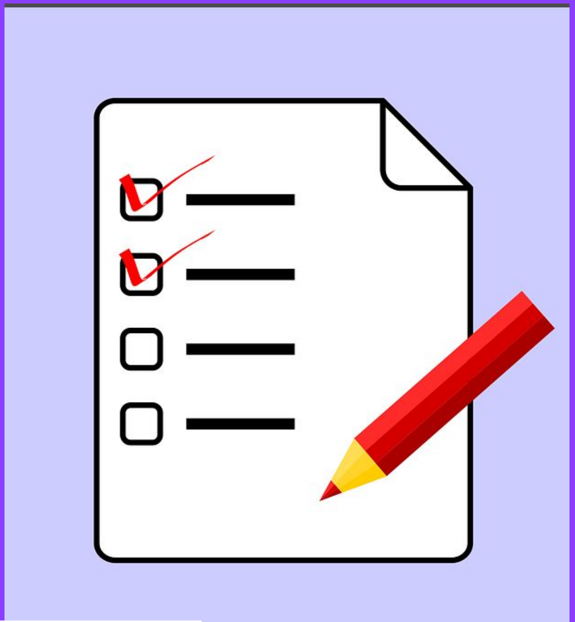
https://colab.research.google.com/drive/1wSgSaN5928oMdlT4Q354SFmlGBfmSR4f?usp=sharing

https://colab.research.google.com/drive/1WJdzT824jEe5V_wbhp_GiVmcPNAdBSlt?usp=sharing

https://colab.research.google.com/drive/1_aBqx09k0W4op00leU9rEUg57TiMzNRT?usp=sharing

Thank You!



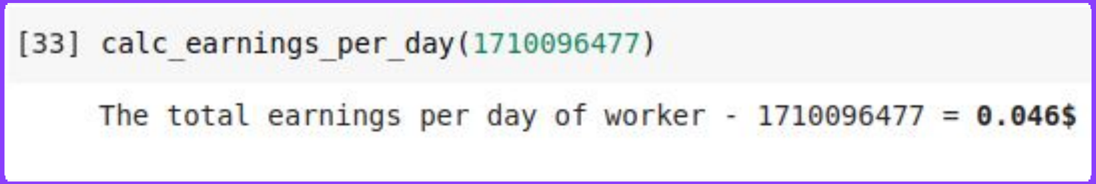


Analysis of Earnings

- Function name: `calc_earnings_per_day(worker_id)`

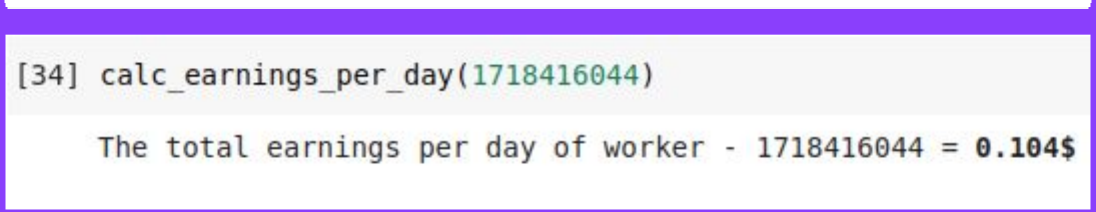
- For getting the total income, income_data was used
- Earnings per day = total earnings / number of days the worker has been on Toloka

Output:



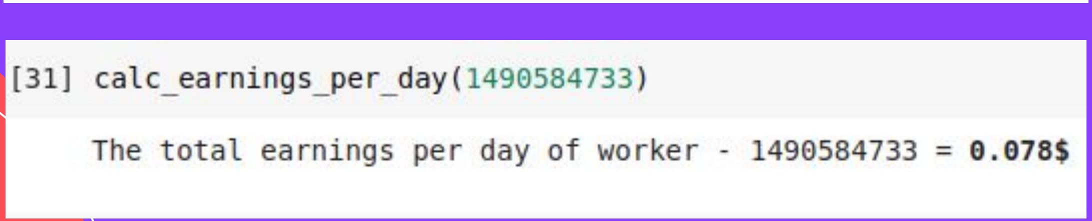
```
[33] calc_earnings_per_day(1710096477)

      The total earnings per day of worker - 1710096477 = 0.046$
```



```
[34] calc_earnings_per_day(1718416044)

      The total earnings per day of worker - 1718416044 = 0.104$
```



```
[31] calc_earnings_per_day(1490584733)

      The total earnings per day of worker - 1490584733 = 0.078$
```