**Data Analysis Task**

**Background**

Discovery (a team at Wikimedia) rely on *event logging* (EL) to track a variety of performance and usage metrics to help them make decisions. Specifically, Discovery is interested in:

* *click through rate*: the proportion of search sessions where the user clicked on one of the results displayed
* *zero results rate*: the proportion of searches that yielded 0 results

and other metrics outside the scope of this task. EL uses JavaScript to asynchronously send messages (events) to the servers when the user has performed specific actions. In this task, you will analyze a subset of their event logs.

**Task**

You must **create a reproducible report**\* answering the following questions:

1. What is their daily overall clickthrough rate? How does it vary between the groups?
2. Which result position do people tend to try first? How does it change day-to-day?
3. What is their daily overall zero results rate? How does it vary between the groups?
4. Let *session length* be approximately the time between the first event and the last event in a session. Choose a variable from the dataset and describe its relationship to session length. Visualize the relationship.
5. Train and Test a simple Classifier to predict if a user is likely to click on one of the search result that appears. Transform the data accordingly. (Bonus points for writing a classifier from scratch) Choose appropriate metrics to measure the performance of your algorithm and visualize the predictions if possible.
6. Summarize your findings in an *executive summary*.
7. **Please share the link to the repo before the end-time, else we’ll consider you opting out of the challenge**

\* Given dependencies and other instructions, **we should be able to re-run your source code with the dataset in the same directory and obtain the same results and figures**. Popular formats for this include Jupyter Notebook (formerly IPython). You are allowed to use Python to complete the tasks above.

Once you've completed the task to your satisfaction, submit the output, along with whatever code you used to produce it, in a github repo. Try to commit and push your code at regular intervals, it helps us understand your approach.

**Data**

The dataset comes from a tracking schema that Wikimedia uses for assessing user satisfaction. Desktop users are randomly sampled to be anonymously tracked by this schema which uses a "I'm alive" pinging system that we can use to estimate how long our users stay on the pages they visit. The dataset contains just a little more than a week of EL data.

|  |  |  |
| --- | --- | --- |
| **Column** | **Value** | **Description** |
| uuid | string | Universally unique identifier (UUID) for backend event handling. |
| timestamp | integer | The date and time (UTC) of the event, formatted as YYYYMMDDhhmmss. |
| session\_id | string | A unique ID identifying individual sessions. |
| group | string | A label ("a" or "b"). |
| action | string | Identifies in which the event was created. See below. |
| checkin | integer | How many seconds the page has been open for. |
| page\_id | string | A unique identifier for correlating page visits and check-ins. |
| n\_results | integer | Number of hits returned to the user. Only shown for searchResultPage events. |
| result\_position | integer | The position of the visited page's link on the search engine results page (SERP). |

The following are possible values for an event's action field:

* searchResultPage: when a new search is performed and the user is shown a SERP.
* visitPage: when the user clicks a link in the results.
* check in: when the user has remained on the page for a pre-specified amount of time.

**Example Session**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| uuid | timestamp | session\_id | group | action | checkin | page\_id | n\_results | result\_position |
| 4f699f344515554a9371fe4ecb5b9ebc | 20160305195246 | 001e61b5477f5efc | b | searchResultPage | NA | 1b341d0ab80eb77e | 7 | NA |
| 759d1dc9966353c2a36846a61125f286 | 20160305195302 | 001e61b5477f5efc | b | visitPage | NA | 5a6a1f75124cbf03 | NA | 1 |
| 77efd5a00a5053c4a713fbe5a48dbac4 | 20160305195312 | 001e61b5477f5efc | b | checkin | 10 | 5a6a1f75124cbf03 | NA | 1 |
| 42420284ad895ec4bcb1f000b949dd5e | 20160305195322 | 001e61b5477f5efc | b | checkin | 20 | 5a6a1f75124cbf03 | NA | 1 |
| 8ffd82c27a355a56882b5860993bd308 | 20160305195332 | 001e61b5477f5efc | b | checkin | 30 | 5a6a1f75124cbf03 | NA | 1 |
| 2988d11968b25b29add3a851bec2fe02 | 20160305195342 | 001e61b5477f5efc | b | checkin | 40 | 5a6a1f75124cbf03 | NA | 1 |

This user's search query returned 7 results, they clicked on the first result, and stayed on the page between 40 and 50 seconds. (The next check-in would have happened at 50s.)

Link to the data: <https://goo.gl/N8RRQb>

Plots: <https://chrisalbon.com/python/matplotlib_grouped_bar_plot.html>

Convert string to time: <https://docs.python.org/2/library/datetime.html#strftime-and-strptime-behavior>

Time Difference: <https://stackoverflow.com/questions/2880713/time-difference-in-seconds-as-a-floating-point>