# Documentation Data Science Recruitment Challenge

Jan Hempel

## Python

Version 3.6.5

Libraries:

- Pandas
- Numpy
- Datetime
- Sys

## Rules and logic

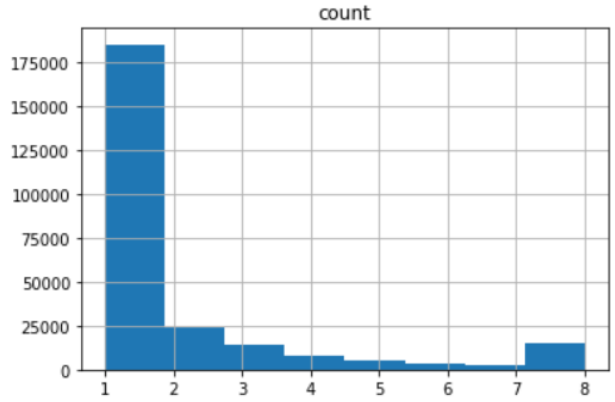
### highly_active

I defined highly_active as having visited the site a certain amount of times. Therefore, I grouped the data by user (uuid) and counted the timestamps. As every definition of what is “highly active” would be rather subjective, I chose to define it on a relative basis. Therefore, I calculated the percentiles of the count of timestamps per user. As one could see, 95% of the users have visited the site less than 8 times during July 2017 and 90% less than 5 times (see visualization). I chose to make the variable rather restrictive and define highly_active = True only if a user belongs to the highest 5% (visited the site at least 8 times during the period).

**Percentiles and visualization**

| Percentile | Number of timestamps |
|---|---|
| 25% | 1 |
| 50% | 1 |
| 75% | 2 |
| 90% | 5 |
| 95% | 8 |
| max | 923 |

The Histogram shows the distribution of the count of timestamps per user. The last bin shows the number of users with 8 or more timestamps (defined as highly_active).




**Further thoughts on this:**

I also thought of considering the date of the last activity in this variable as well. “Highly active” may not only mean that a user has visited a site often but also recently. However, I decided to keep this separated and considered it in my fourth feature (see below).

**Distribution**

Highly active: 14825

Not highly active: 242529

### multiple_days

I defined multiple_days as having visited the site on at least two days during the different period. Therefore, I extracted the day from the timestamp in a first step. To keep the code dynamically usable for future periods, I kept year and month in the “day” variable.

As for highly_active, I grouped the data by user (uuid) and calculated the count of different days to calculate the boolean for multiple_days.

**Distribution**

Visited site on multiple days: 36035

Visited site on only one day: 221319

### weekday_biz

For weekday_biz I first calculated for each timestamp, if it falls into business hours or not. I defined business hours from Monday to Friday, 9am to 5pm. I then grouped the data by user (uuid) and calculated the count of timestamps (count) and the count of timestamps during business hours (sum). I defined weekday_biz to be true if there are more timestamps during business hours than during recreational hours (sum/count > 0.5).

**Distribution**

Visited site preferably during business hours: 77411

Visited site preferably during recreational hours: 179943

## days_since_last_activity

As fourth feature a chose to calculate the number of days since the last visit. The hypothesis behind this feature is the following: The more recently a user has visited the site, the more likely he is to book a flight or hotel. To make the feature dynamically usable in future, I first calculated the latest date in the dataset, without daytime (latest_date). Secondly, I calculated for each user the difference between the latest_date and the last time the respective user visited the site. Then, I only needed to transform this to integer to make it usable in the model.

**Distribution**

Histogram for days_since_last_activity. Each day has its own bin: