TDT4225 Assignment 3 MongoDB

Group 45
Group member(s): Simen Refsland

October 19, 2023

Introduction

NB! Largely the same as assignment 2...

In this assignment, I was tasked with cleaning raw data and then inserting structured data from the Geolife dataset. This involved combining and extracting info present in the raw .plt files in such a way that it was compatible with the User, Activity and TrackPoint tables specified. In the next part, several different queries were written to answer questions about the data, sometimes involving a combination of MongoDB queries and manual processing in Python.

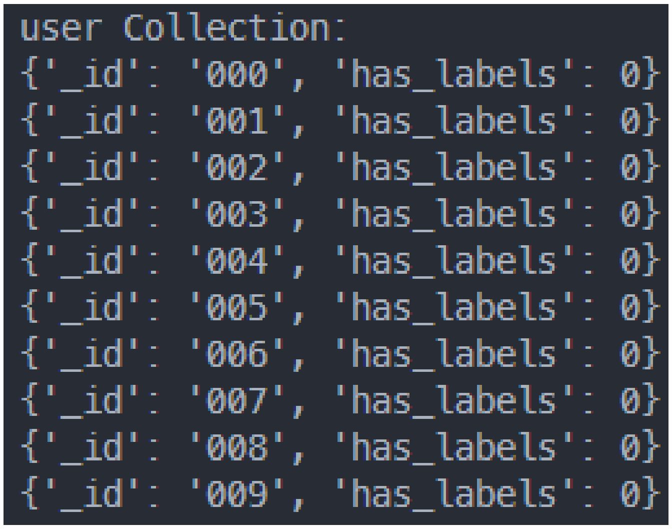
I have not used the virtual machines to run MongoDB, the testing database is local. Since I was the only one in the group, I have worked locally, as opposed to using Git.

Results

Part 1

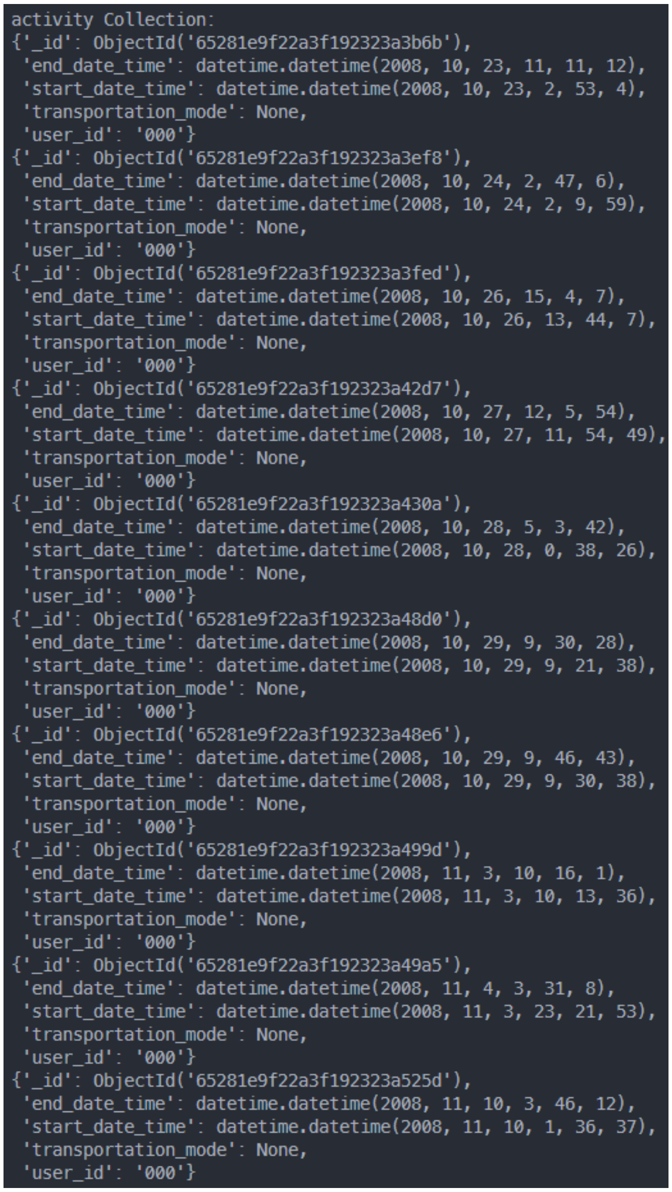
NOTE! When I discard files that are longer than 2506 lines (including header lines), it should be noted that the lines are counted where there is actual text, meaning I don't count the last line which is empty. This *may* slightly affect the number of activities that are inserted. I also assume that we discard activities that have more than 2500 lines, as there would be no need to check if the .plt file exceeds 2500 lines (you could just take the first 2506 lines otherwise). Some transportation modes are also discarded if they don't fit the start and end date of an activity, most often relating to the fact that some users choose to label different parts of a single activity.

The preprocessing and insertion is largely the same as assignment 2, but the datatypes are explicitly defined when they are not a string, as there is no schema specification prior to insertion. The way the data was represented prior to insertion in SQL, was largely the same representation needed to insert it into MongoDB. The modeling decisions are explained in the discussion part.



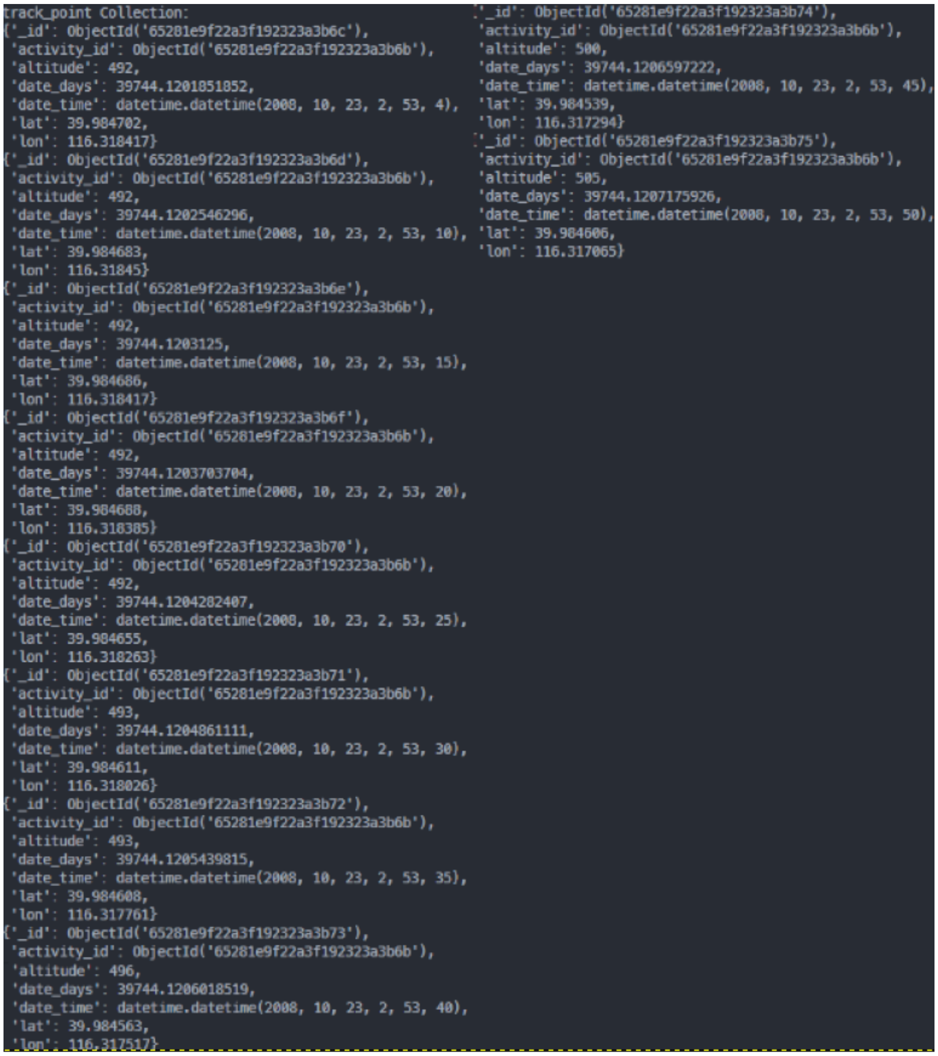
```
user Collection:
{'_id': '000', 'has_labels': 0}
{'_id': '001', 'has_labels': 0}
{'_id': '002', 'has_labels': 0}
{'_id': '003', 'has_labels': 0}
{'_id': '004', 'has_labels': 0}
{'_id': '005', 'has_labels': 0}
{'_id': '006', 'has_labels': 0}
{'_id': '007', 'has_labels': 0}
{'_id': '008', 'has_labels': 0}
{'_id': '009', 'has_labels': 0}
```

Figure 1: First 10 users



```
activity Collection:
{'_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'end_date_time': datetime.datetime(2008, 10, 23, 11, 11, 12),
 'start_date_time': datetime.datetime(2008, 10, 23, 2, 53, 4),
 'transportation_mode': None,
 'user_id': '000'}
{'_id': ObjectId('65281e9f22a3f192323a3ef8'),
 'end_date_time': datetime.datetime(2008, 10, 24, 2, 47, 6),
 'start_date_time': datetime.datetime(2008, 10, 24, 2, 9, 59),
 'transportation_mode': None,
 'user_id': '000'}
{'_id': ObjectId('65281e9f22a3f192323a3fed'),
 'end_date_time': datetime.datetime(2008, 10, 26, 15, 4, 7),
 'start_date_time': datetime.datetime(2008, 10, 26, 13, 44, 7),
 'transportation_mode': None,
 'user_id': '000'}
{'_id': ObjectId('65281e9f22a3f192323a42d7'),
 'end_date_time': datetime.datetime(2008, 10, 27, 12, 5, 54),
 'start_date_time': datetime.datetime(2008, 10, 27, 11, 54, 49),
 'transportation_mode': None,
 'user_id': '000'}
{'_id': ObjectId('65281e9f22a3f192323a430a'),
 'end_date_time': datetime.datetime(2008, 10, 28, 5, 3, 42),
 'start_date_time': datetime.datetime(2008, 10, 28, 0, 38, 26),
 'transportation_mode': None,
 'user_id': '000'}
{'_id': ObjectId('65281e9f22a3f192323a48d0'),
 'end_date_time': datetime.datetime(2008, 10, 29, 9, 30, 28),
 'start_date_time': datetime.datetime(2008, 10, 29, 9, 21, 38),
 'transportation_mode': None,
 'user_id': '000'}
{'_id': ObjectId('65281e9f22a3f192323a48e6'),
 'end_date_time': datetime.datetime(2008, 10, 29, 9, 46, 43),
 'start_date_time': datetime.datetime(2008, 10, 29, 9, 30, 38),
 'transportation_mode': None,
 'user_id': '000'}
{'_id': ObjectId('65281e9f22a3f192323a499d'),
 'end_date_time': datetime.datetime(2008, 11, 3, 10, 16, 1),
 'start_date_time': datetime.datetime(2008, 11, 3, 10, 13, 36),
 'transportation_mode': None,
 'user_id': '000'}
{'_id': ObjectId('65281e9f22a3f192323a49a5'),
 'end_date_time': datetime.datetime(2008, 11, 4, 3, 31, 8),
 'start_date_time': datetime.datetime(2008, 11, 3, 23, 21, 53),
 'transportation_mode': None,
 'user_id': '000'}
{'_id': ObjectId('65281e9f22a3f192323a525d'),
 'end_date_time': datetime.datetime(2008, 11, 10, 3, 46, 12),
 'start_date_time': datetime.datetime(2008, 11, 10, 1, 36, 37),
 'transportation_mode': None,
 'user_id': '000'}
```

Figure 2: First 10 activities



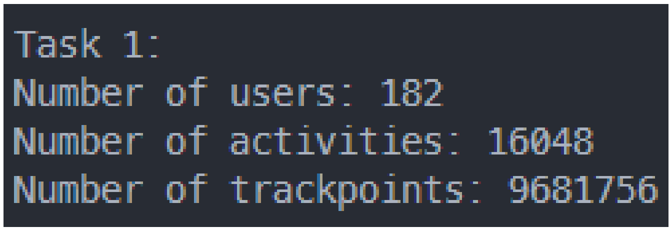
```
track_point Collection:
{'_id': ObjectId('65281e9f22a3f192323a3b6c'),
 'activity_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'altitude': 492,
 'date_days': 39744.1201851852,
 'date_time': datetime.datetime(2008, 10, 23, 2, 53, 4),
 'lat': 39.984702,
 'lon': 116.318417}
{'_id': ObjectId('65281e9f22a3f192323a3b6d'),
 'activity_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'altitude': 492,
 'date_days': 39744.1202546296,
 'date_time': datetime.datetime(2008, 10, 23, 2, 53, 10),
 'lat': 39.984683,
 'lon': 116.31845}
{'_id': ObjectId('65281e9f22a3f192323a3b6e'),
 'activity_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'altitude': 492,
 'date_days': 39744.1203125,
 'date_time': datetime.datetime(2008, 10, 23, 2, 53, 15),
 'lat': 39.984686,
 'lon': 116.318417}
{'_id': ObjectId('65281e9f22a3f192323a3b6f'),
 'activity_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'altitude': 492,
 'date_days': 39744.1203703704,
 'date_time': datetime.datetime(2008, 10, 23, 2, 53, 20),
 'lat': 39.984688,
 'lon': 116.318385}
{'_id': ObjectId('65281e9f22a3f192323a3b70'),
 'activity_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'altitude': 492,
 'date_days': 39744.1204282407,
 'date_time': datetime.datetime(2008, 10, 23, 2, 53, 25),
 'lat': 39.984655,
 'lon': 116.318263}
{'_id': ObjectId('65281e9f22a3f192323a3b71'),
 'activity_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'altitude': 493,
 'date_days': 39744.1204861111,
 'date_time': datetime.datetime(2008, 10, 23, 2, 53, 30),
 'lat': 39.984611,
 'lon': 116.318026}
{'_id': ObjectId('65281e9f22a3f192323a3b72'),
 'activity_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'altitude': 493,
 'date_days': 39744.1205439815,
 'date_time': datetime.datetime(2008, 10, 23, 2, 53, 35),
 'lat': 39.984608,
 'lon': 116.317761}
{'_id': ObjectId('65281e9f22a3f192323a3b73'),
 'activity_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'altitude': 496,
 'date_days': 39744.1206018519,
 'date_time': datetime.datetime(2008, 10, 23, 2, 53, 40),
 'lat': 39.984563,
 'lon': 116.317517}
{'_id': ObjectId('65281e9f22a3f192323a3b74'),
 'activity_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'altitude': 500,
 'date_days': 39744.1206597222,
 'date_time': datetime.datetime(2008, 10, 23, 2, 53, 45),
 'lat': 39.984539,
 'lon': 116.317294}
{'_id': ObjectId('65281e9f22a3f192323a3b75'),
 'activity_id': ObjectId('65281e9f22a3f192323a3b6b'),
 'altitude': 505,
 'date_days': 39744.1207175926,
 'date_time': datetime.datetime(2008, 10, 23, 2, 53, 50),
 'lat': 39.984606,
 'lon': 116.317065}
```

Figure 3: First 10 trackpoints

Part 2

Task 1

Counts the number of documents in user, activity and track_point collections.

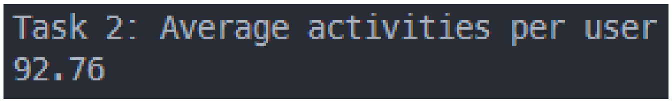


```
Task 1:
Number of users: 182
Number of activities: 16048
Number of trackpoints: 9681756
```

Figure 4: Task 1 result

Task 2

Grouping on user_id and then taking the average.

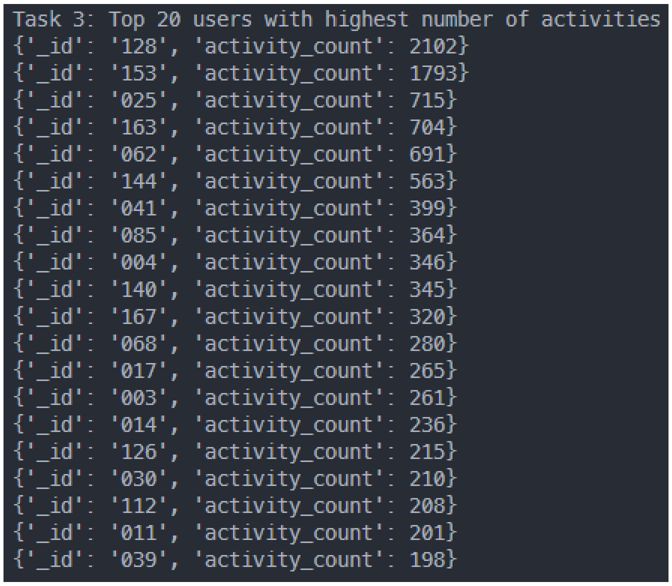


```
Task 2: Average activities per user
92.76
```

Figure 5: Task 2 result

Task 3

Grouping on user_id and sorting by highest activity count and limiting by 20.

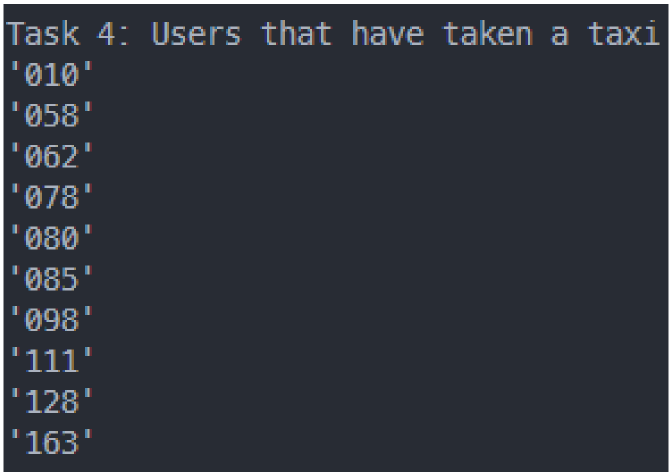


```
Task 3: Top 20 users with highest number of activities
{'_id': '128', 'activity_count': 2102}
{'_id': '153', 'activity_count': 1793}
{'_id': '025', 'activity_count': 715}
{'_id': '163', 'activity_count': 704}
{'_id': '062', 'activity_count': 691}
{'_id': '144', 'activity_count': 563}
{'_id': '041', 'activity_count': 399}
{'_id': '085', 'activity_count': 364}
{'_id': '004', 'activity_count': 346}
{'_id': '140', 'activity_count': 345}
{'_id': '167', 'activity_count': 320}
{'_id': '068', 'activity_count': 280}
{'_id': '017', 'activity_count': 265}
{'_id': '003', 'activity_count': 261}
{'_id': '014', 'activity_count': 236}
{'_id': '126', 'activity_count': 215}
{'_id': '030', 'activity_count': 210}
{'_id': '112', 'activity_count': 208}
{'_id': '011', 'activity_count': 201}
{'_id': '039', 'activity_count': 198}
```

Figure 6: Task 3 result

Task 4

Selecting the distinct user_ids who have transportation_mode at least once.

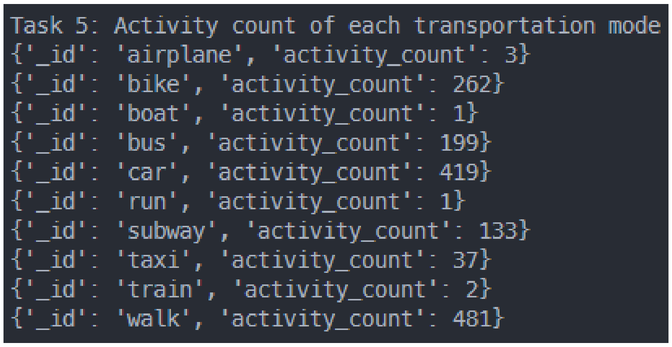


```
Task 4: Users that have taken a taxi
'010'
'058'
'062'
'078'
'080'
'085'
'098'
'111'
'128'
'163'
```

Figure 7: Task 4 result

Task 5

Simple grouping on transportation mode and counting number of activities.

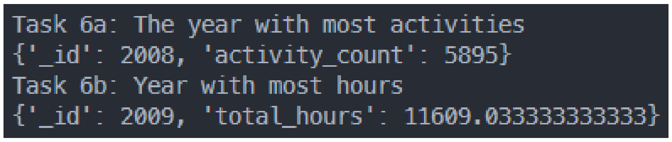


```
Task 5: Activity count of each transportation mode
{'_id': 'airplane', 'activity_count': 3}
{'_id': 'bike', 'activity_count': 262}
{'_id': 'boat', 'activity_count': 1}
{'_id': 'bus', 'activity_count': 199}
{'_id': 'car', 'activity_count': 419}
{'_id': 'run', 'activity_count': 1}
{'_id': 'subway', 'activity_count': 133}
{'_id': 'taxi', 'activity_count': 37}
{'_id': 'train', 'activity_count': 2}
{'_id': 'walk', 'activity_count': 481}
```

Figure 8: Task 5 result

Task 6

2008 is the year with most activities and 2009 is the year with most hours recorded. The first subtask handles edge cases where the activity might end in the next year, this is not done in subtask b because 2009 will be the year with most recorded hours by a wide margin, but an alternative implementation is done in Python, but only differs by about 3 hours, so not significantly.

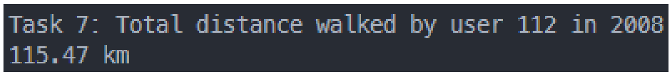


```
Task 6a: The year with most activities
{'_id': 2008, 'activity_count': 5895}
Task 6b: Year with most hours
{'_id': 2009, 'total_hours': 11609.033333333333}
```

Figure 9: Task 6a and 6b result

Task 7

This is calculated by filtering out the start_date_time year, and user id and correct transportation mode. To ensure that we don't measure distances in the next year if there is some overlap, I check that the year of the current trackpoint is 2008.

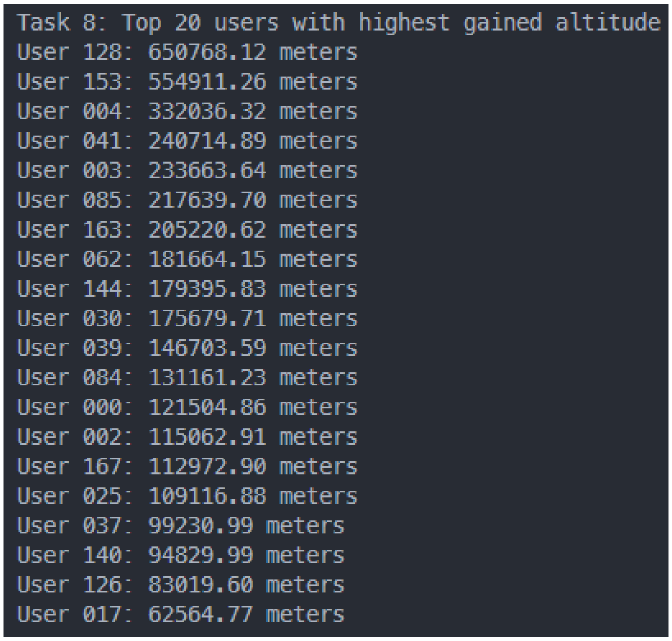


```
Task 7: Total distance walked by user 112 in 2008
115.47 km
```

Figure 10: Task 7 result

Task 8

Filters out trackpoints whose altitude is -777 and adds gained altitude if the current altitude is higher than the last trackpoint.

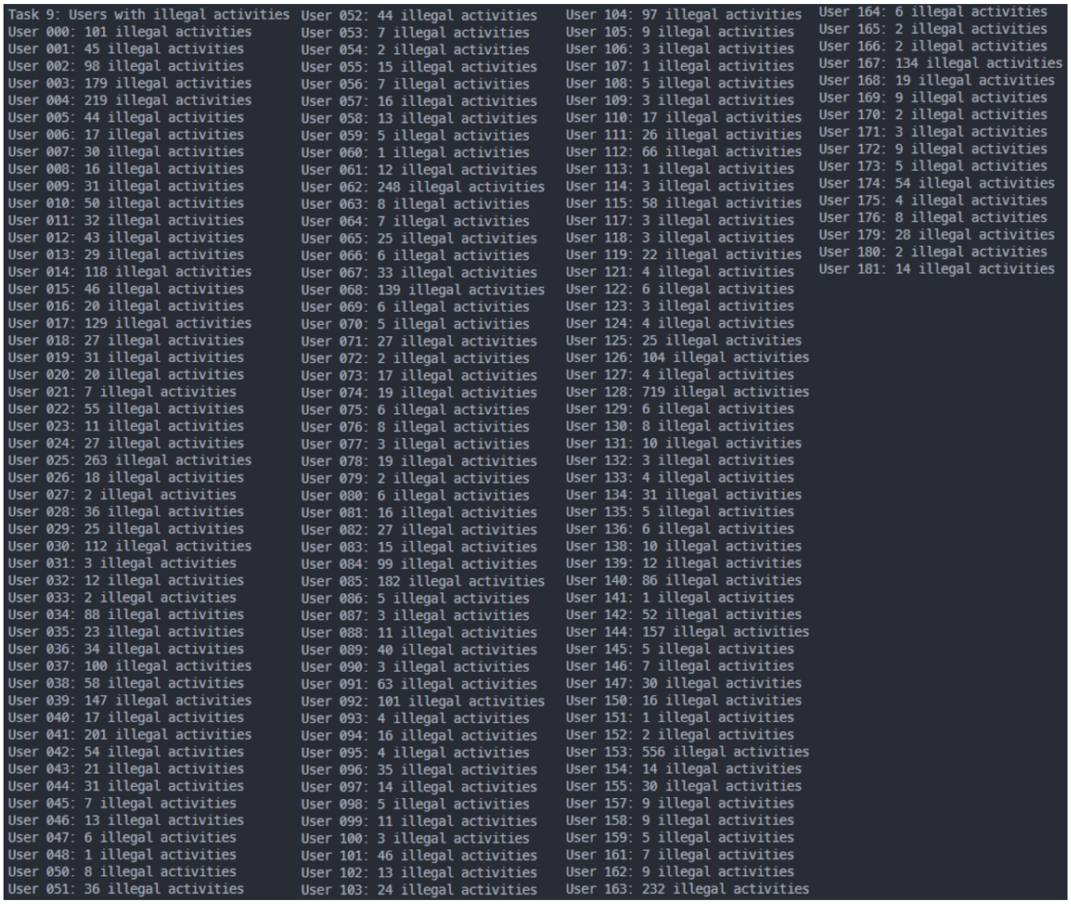


```
Task 8: Top 20 users with highest gained altitude
User 128: 650768.12 meters
User 153: 554911.26 meters
User 004: 332036.32 meters
User 041: 240714.89 meters
User 003: 233663.64 meters
User 085: 217639.70 meters
User 163: 205220.62 meters
User 062: 181664.15 meters
User 144: 179395.83 meters
User 030: 175679.71 meters
User 039: 146703.59 meters
User 084: 131161.23 meters
User 000: 121504.86 meters
User 002: 115062.91 meters
User 167: 112972.90 meters
User 025: 109116.88 meters
User 037: 99230.99 meters
User 140: 94829.99 meters
User 126: 83019.60 meters
User 017: 62564.77 meters
```

Figure 11: Task 8 result

Task 9

Fetches all the activities and counts invalid activities if consecutive timestamps are more than 5 minutes apart. There is no filtering on altitude being -777 here.

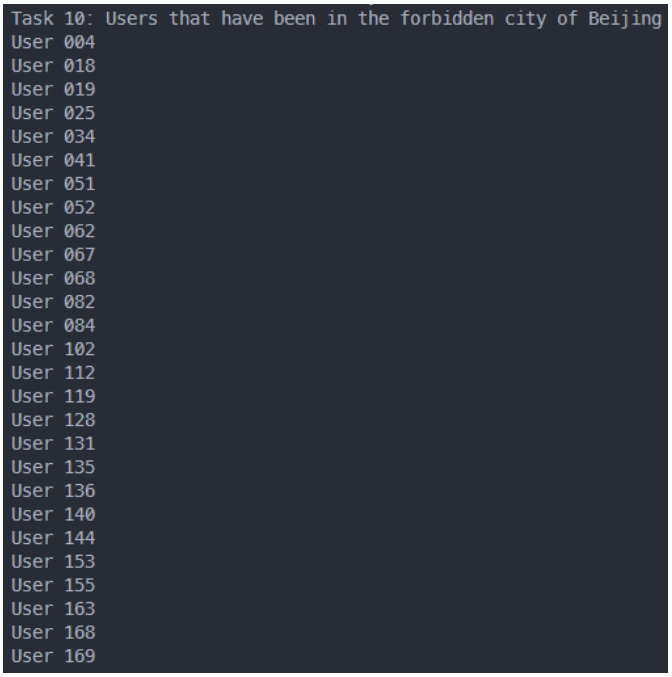


```
Task 9: Users with illegal activities
User 000: 101 illegal activities
User 001: 45 illegal activities
User 002: 98 illegal activities
User 003: 179 illegal activities
User 004: 219 illegal activities
User 005: 44 illegal activities
User 006: 17 illegal activities
User 007: 30 illegal activities
User 008: 16 illegal activities
User 009: 31 illegal activities
User 010: 50 illegal activities
User 011: 32 illegal activities
User 012: 43 illegal activities
User 013: 29 illegal activities
User 014: 118 illegal activities
User 015: 46 illegal activities
User 016: 20 illegal activities
User 017: 129 illegal activities
User 018: 27 illegal activities
User 019: 31 illegal activities
User 020: 20 illegal activities
User 021: 7 illegal activities
User 022: 55 illegal activities
User 023: 11 illegal activities
User 024: 27 illegal activities
User 025: 263 illegal activities
User 026: 18 illegal activities
User 027: 2 illegal activities
User 028: 36 illegal activities
User 029: 25 illegal activities
User 030: 112 illegal activities
User 031: 3 illegal activities
User 032: 12 illegal activities
User 033: 2 illegal activities
User 034: 88 illegal activities
User 035: 23 illegal activities
User 036: 34 illegal activities
User 037: 100 illegal activities
User 038: 58 illegal activities
User 039: 147 illegal activities
User 040: 17 illegal activities
User 041: 201 illegal activities
User 042: 54 illegal activities
User 043: 21 illegal activities
User 044: 31 illegal activities
User 045: 7 illegal activities
User 046: 13 illegal activities
User 047: 6 illegal activities
User 048: 1 illegal activities
User 050: 8 illegal activities
User 051: 36 illegal activities
User 052: 44 illegal activities
User 053: 7 illegal activities
User 054: 2 illegal activities
User 055: 15 illegal activities
User 056: 7 illegal activities
User 057: 16 illegal activities
User 058: 13 illegal activities
User 059: 5 illegal activities
User 060: 1 illegal activities
User 061: 12 illegal activities
User 062: 248 illegal activities
User 063: 8 illegal activities
User 064: 7 illegal activities
User 065: 25 illegal activities
User 066: 6 illegal activities
User 067: 33 illegal activities
User 068: 139 illegal activities
User 069: 6 illegal activities
User 070: 5 illegal activities
User 071: 27 illegal activities
User 072: 2 illegal activities
User 073: 17 illegal activities
User 074: 19 illegal activities
User 075: 6 illegal activities
User 076: 8 illegal activities
User 077: 3 illegal activities
User 078: 19 illegal activities
User 079: 2 illegal activities
User 080: 6 illegal activities
User 081: 16 illegal activities
User 082: 27 illegal activities
User 083: 15 illegal activities
User 084: 99 illegal activities
User 085: 182 illegal activities
User 086: 5 illegal activities
User 087: 3 illegal activities
User 088: 11 illegal activities
User 089: 40 illegal activities
User 090: 3 illegal activities
User 091: 63 illegal activities
User 092: 101 illegal activities
User 093: 4 illegal activities
User 094: 16 illegal activities
User 095: 4 illegal activities
User 096: 35 illegal activities
User 097: 14 illegal activities
User 098: 5 illegal activities
User 099: 11 illegal activities
User 100: 3 illegal activities
User 101: 46 illegal activities
User 102: 13 illegal activities
User 103: 24 illegal activities
User 104: 97 illegal activities
User 105: 9 illegal activities
User 106: 3 illegal activities
User 107: 1 illegal activities
User 108: 5 illegal activities
User 109: 3 illegal activities
User 110: 17 illegal activities
User 111: 26 illegal activities
User 112: 66 illegal activities
User 113: 1 illegal activities
User 114: 3 illegal activities
User 115: 58 illegal activities
User 117: 3 illegal activities
User 118: 3 illegal activities
User 119: 22 illegal activities
User 121: 4 illegal activities
User 122: 6 illegal activities
User 123: 3 illegal activities
User 124: 4 illegal activities
User 125: 25 illegal activities
User 126: 104 illegal activities
User 127: 4 illegal activities
User 128: 719 illegal activities
User 129: 6 illegal activities
User 130: 8 illegal activities
User 131: 10 illegal activities
User 132: 3 illegal activities
User 133: 4 illegal activities
User 134: 31 illegal activities
User 135: 5 illegal activities
User 136: 6 illegal activities
User 138: 10 illegal activities
User 139: 12 illegal activities
User 140: 86 illegal activities
User 141: 1 illegal activities
User 142: 52 illegal activities
User 144: 157 illegal activities
User 145: 5 illegal activities
User 146: 7 illegal activities
User 147: 30 illegal activities
User 150: 16 illegal activities
User 151: 1 illegal activities
User 152: 2 illegal activities
User 153: 556 illegal activities
User 154: 14 illegal activities
User 155: 30 illegal activities
User 157: 9 illegal activities
User 158: 9 illegal activities
User 159: 5 illegal activities
User 161: 7 illegal activities
User 162: 9 illegal activities
User 163: 232 illegal activities
User 164: 6 illegal activities
User 165: 2 illegal activities
User 166: 2 illegal activities
User 167: 134 illegal activities
User 168: 19 illegal activities
User 169: 9 illegal activities
User 170: 2 illegal activities
User 171: 3 illegal activities
User 172: 9 illegal activities
User 173: 5 illegal activities
User 174: 54 illegal activities
User 175: 4 illegal activities
User 176: 8 illegal activities
User 179: 28 illegal activities
User 180: 2 illegal activities
User 181: 14 illegal activities
```

Figure 12: Task 9 result

Task 10

As stated on Piazza, I have defined a radius (1 km, could be a bit off actual city limits) around the given coordinate where any user within this radius is said to be within the forbidden city limits.

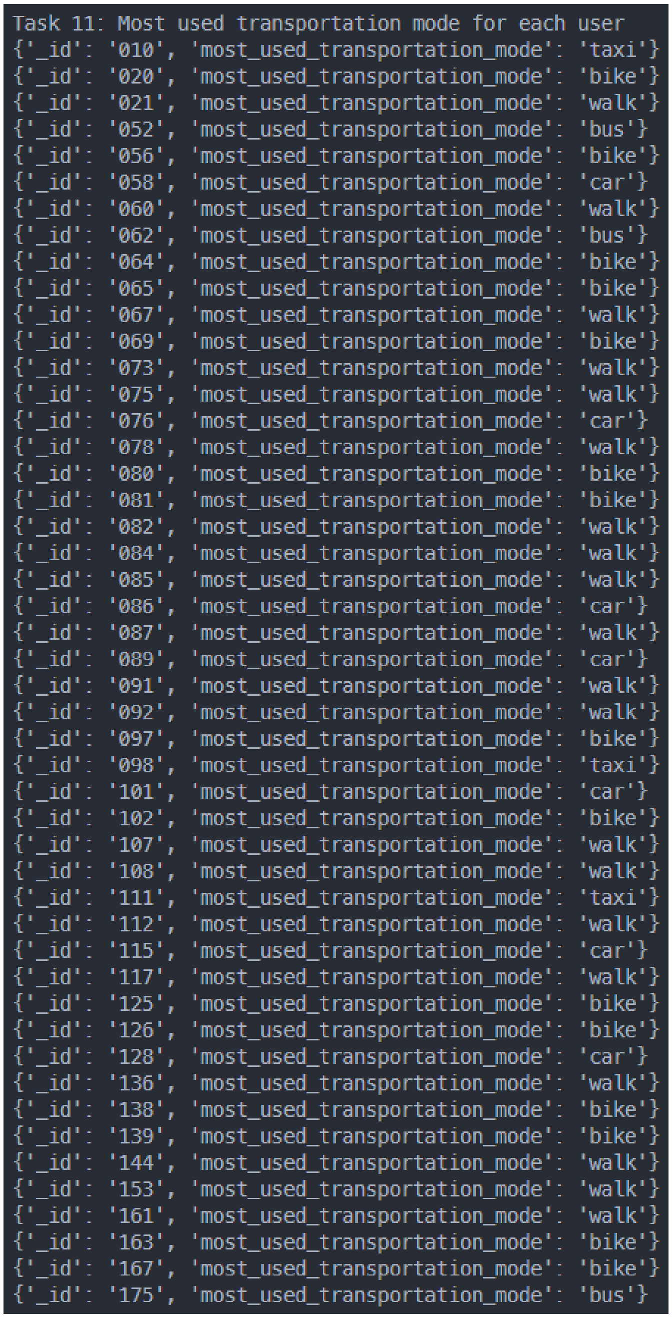


```
Task 10: Users that have been in the forbidden city of Beijing
User 004
User 018
User 019
User 025
User 034
User 041
User 051
User 052
User 062
User 067
User 068
User 082
User 084
User 102
User 112
User 119
User 128
User 131
User 135
User 136
User 140
User 144
User 153
User 155
User 163
User 168
User 169
```

Figure 13: Task 8 result

Task 11

To account for users that have ties for most used transportation mode, I have just taken the first transportation mode for each user after sorting by user_id and activity count. I noticed that the answers slightly deviate from assignment 2, but this is likely because of the ”selection policy” when there is a tie, which is a result of how entries are ordered differently.



```
Task 11: Most used transportation mode for each user
{'_id': '010', 'most_used_transportation_mode': 'taxi'}
{'_id': '020', 'most_used_transportation_mode': 'bike'}
{'_id': '021', 'most_used_transportation_mode': 'walk'}
{'_id': '052', 'most_used_transportation_mode': 'bus'}
{'_id': '056', 'most_used_transportation_mode': 'bike'}
{'_id': '058', 'most_used_transportation_mode': 'car'}
{'_id': '060', 'most_used_transportation_mode': 'walk'}
{'_id': '062', 'most_used_transportation_mode': 'bus'}
{'_id': '064', 'most_used_transportation_mode': 'bike'}
{'_id': '065', 'most_used_transportation_mode': 'bike'}
{'_id': '067', 'most_used_transportation_mode': 'walk'}
{'_id': '069', 'most_used_transportation_mode': 'bike'}
{'_id': '073', 'most_used_transportation_mode': 'walk'}
{'_id': '075', 'most_used_transportation_mode': 'walk'}
{'_id': '076', 'most_used_transportation_mode': 'car'}
{'_id': '078', 'most_used_transportation_mode': 'walk'}
{'_id': '080', 'most_used_transportation_mode': 'bike'}
{'_id': '081', 'most_used_transportation_mode': 'bike'}
{'_id': '082', 'most_used_transportation_mode': 'walk'}
{'_id': '084', 'most_used_transportation_mode': 'walk'}
{'_id': '085', 'most_used_transportation_mode': 'walk'}
{'_id': '086', 'most_used_transportation_mode': 'car'}
{'_id': '087', 'most_used_transportation_mode': 'walk'}
{'_id': '089', 'most_used_transportation_mode': 'car'}
{'_id': '091', 'most_used_transportation_mode': 'walk'}
{'_id': '092', 'most_used_transportation_mode': 'walk'}
{'_id': '097', 'most_used_transportation_mode': 'bike'}
{'_id': '098', 'most_used_transportation_mode': 'taxi'}
{'_id': '101', 'most_used_transportation_mode': 'car'}
{'_id': '102', 'most_used_transportation_mode': 'bike'}
{'_id': '107', 'most_used_transportation_mode': 'walk'}
{'_id': '108', 'most_used_transportation_mode': 'walk'}
{'_id': '111', 'most_used_transportation_mode': 'taxi'}
{'_id': '112', 'most_used_transportation_mode': 'walk'}
{'_id': '115', 'most_used_transportation_mode': 'car'}
{'_id': '117', 'most_used_transportation_mode': 'walk'}
{'_id': '125', 'most_used_transportation_mode': 'bike'}
{'_id': '126', 'most_used_transportation_mode': 'bike'}
{'_id': '128', 'most_used_transportation_mode': 'car'}
{'_id': '136', 'most_used_transportation_mode': 'walk'}
{'_id': '138', 'most_used_transportation_mode': 'bike'}
{'_id': '139', 'most_used_transportation_mode': 'bike'}
{'_id': '144', 'most_used_transportation_mode': 'walk'}
{'_id': '153', 'most_used_transportation_mode': 'walk'}
{'_id': '161', 'most_used_transportation_mode': 'walk'}
{'_id': '163', 'most_used_transportation_mode': 'bike'}
{'_id': '167', 'most_used_transportation_mode': 'bike'}
{'_id': '175', 'most_used_transportation_mode': 'bus'}
```

Figure 14: Task 11 result

Discussion

I have modeled the data much in the same way as before, where user, activity and trackpoint is its own collection. Looking at the cardinality of the data might suggest that a one-to-many approach is appropriate (as max numbers of activities is ≈ 2000 and only activites with 2500 trackpoints or less are added), where an array of references to the trackpoints may be appropriate. However, if we were to imagine a live production setting, we can easily see that the number of trackpoints and activities would be unbounded, meaning that a one-to-zillions relationship is possible. Looking back, as the user collection only contains the *has_labels* field, and is rarely (if ever) accessed, it would probably be more logical to denormalize this collection into the activity collection. In addition, the data is not expected to change, as the entries can be viewed as "archived" or static, leading to few possibilities for updates. Denormalizing the *user_id* field into trackpoints may also be reasonable, as many of the queries are based on the user, where the only interesting field from activities is the user id.

With this assignment, I decided to utilize manual processing in Python to a greater extent than in the previous assignment. This was to solve tasks which deals with current and previous trackpoints, as it's more intuitive to come up with a solution that works faster than by pure querying, which I learned during the last assignment. This seems to be fairly easy to solve using *$setWindowFields* and the *$shift* operator however, but I only saw this after the fact, and decided to not implement due to time concerns. The answers should be correct nevertheless.

Prior to this, I had very limited experience with MongoDB, especially dealing with more complicated queries than merely finding some data based on some condition. I've therefore learned a lot during this assignment, although I've might not seen the true potential of this type of database due to the strongly relational structure of this data, which in my opinion is better suited for relational databases.

Something that took a while to realize, was that an index was needed on *activity_id* to efficiently use *$lookup* to embed trackpoints into activities, otherwise the time used on lookups were way too long.

Feedback

The assignment was fun and educational, I've found it to have less ambiguity than the last assignment, which was good.