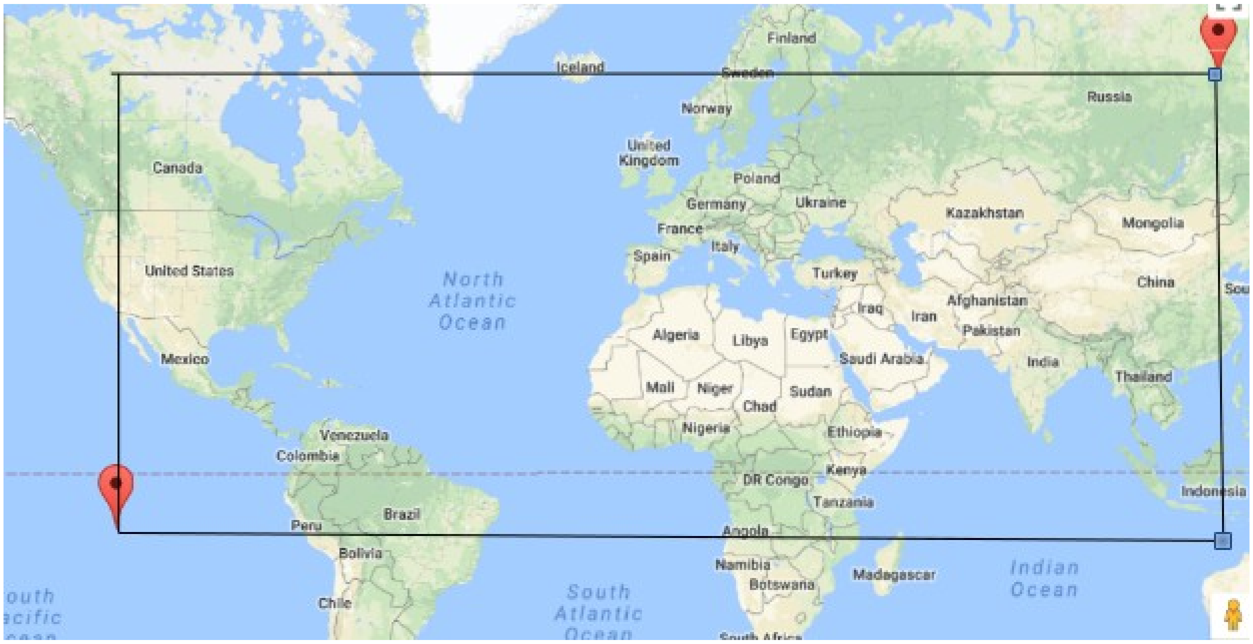
1. November 1st 2013 to november 30th 2013, There are 14388452 rows

2. medallion, hack_license, vendor_id, rate_code, store_and_fwd_flag, pickup_datetime, dropoff_datetime, passenger_count, trip_time_in_secs, trip_distance, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude

3 and 4 medallion, E9A54865CAF737ED003957478C9D8FA1, string
hack_license, 912A2B86F30CDFE246586972A892367E, string
vendor_id, CMT, string
rate_code, 1, int
store_and_fwd_flag, N, string
pickup_datetime, 2013-11-25 15:53:33, string
dropoff_datetime, 2013-11-25 16:00:51, string
passenger_count, 1, int
trip_time_in_secs, 437, int
trip_distance, .60, decimal
pickup_longitude, -73.978104, decimal
pickup_latitude, 40.752968, decimal
dropoff_longitude, -73.985756, decimal
dropoff_latitude, 40.762684, decimal

5. cleaning the data resulted in still unbelievable minimum = -117.47642 and max = 116.984 for longitude and minimum = -12.124776, max = 64.870567 for latitude



6. rate code '3', '0', '210', '7', '1', '2', '4', '9', '8', '6', '5', '10'
store and fwd flag Y', '', 'N'
passenger_count '3', '0', '7', '1', '2', '4', '9', '8', '6', '5', '208'

7. rate_code

minimum = 0.0

max = 210.0

passenger_count

minimum = 0.0

max = 208.0

trip_time_in_secs 8

minimum = 0.0

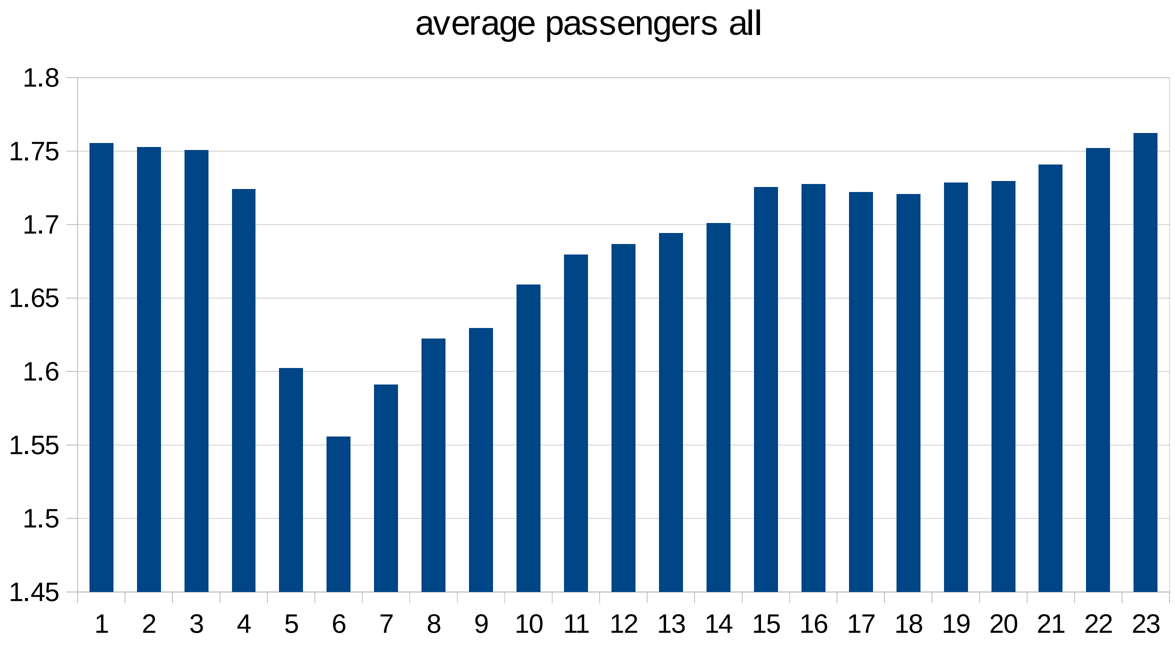
max = 10800.0

trip_distance 9

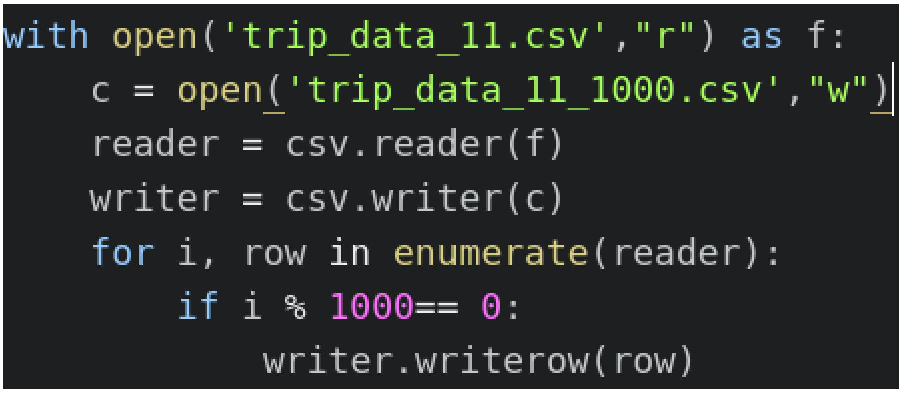
minimum = 0.0

max = 100.0

8.

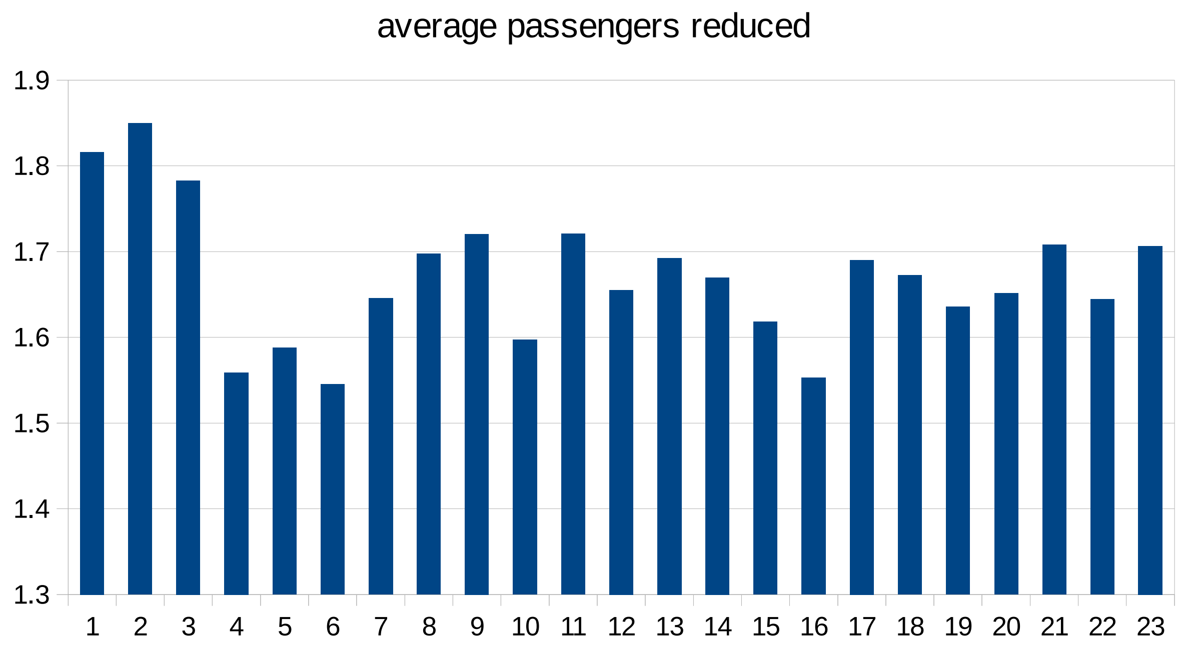


9.



```
with open('trip_data_11.csv',"r") as f:
    c = open('trip_data_11_1000.csv',"w")
    reader = csv.reader(f)
    writer = csv.writer(c)
    for i, row in enumerate(reader):
        if i % 1000== 0:
            writer.writerow(row)
```

10.



The difference between the two graphs are quite a lot, just taking every thousandth row was not a very good way to get a reduced data set. In the reduced set the averages does not show a natural rise and fall during the day.