

d) What is the average trip time for different pickup locations?

```
from mrjob.job import MRJob
from mrjob.step import MRStep
from datetime import datetime

class VendorTripDetails(MRJob):

    def steps(self):
        return [
            MRStep(mapper=self.mapper, reducer=self.reducer),
            MRStep(reducer=self.get_average_trip_time)
        ]

    def get_dateformat(self, datetime_str):
        formats = ['%Y-%m-%d %H:%M', '%Y-%m-%d %H:%M:%S', '%d-%m-%Y %H:%M:%S', '%d-%m-%Y %H:%M']
        for fmt in formats:
            try:
                return datetime.strptime(datetime_str, fmt)
            except ValueError:
                pass
        raise ValueError('No valid date format available')

    def mapper(self, _, line):
        if not line.startswith('VendorID'):
            data = line.strip().split(',')
            pickup_location = data[7]
            pickup_datetime = self.get_dateformat(data[1])
            drop_datetime = self.get_dateformat(data[2])
            trip_time_diff = drop_datetime - pickup_datetime
            trip_time = trip_time_diff.total_seconds() / 60
            yield pickup_location, (trip_time, 1)

    def reducer(self, key, values):
        total_trip_time = 0
        total_count = 0
        for trip_time, count in values:
            total_trip_time += trip_time
            total_count += count
        yield None, (total_trip_time / total_count, key)

    def get_average_trip_time(self, _, values):
        sorted_values = sorted(values, reverse=True)
        for average_trip_time, pickup_location in sorted_values:
            yield pickup_location, round(average_trip_time, 4)

if __name__ == '__main__':
    VendorTripDetails.run()
```