TASK-4:

Write MapReduce codes to perform the tasks using the files you've downloaded on your EMR Instance

Dataset used is: https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-05.csv

1) Get the dataset to with the following command:

```
wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-05.csv
```

2) Create python scripts by using the following command:

```
vim mrtask_a.py
vim mrtask_b.py
vim mrtask_c.py
vim mrtask_d.py
vim mrtask_e.py
vim mrtask_f.py
```

(a) mrtask_a.py

```
# Which vendors have the most trips, and what is the total revenue generated by
that vendor?

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

class MostTripsTotalRevenue(MRJob):

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

    def mapper(self, _, line):
        if not line.startswith('VendorID'):
            data = line.split(',')
            vendor_id = data[0]
            revenue = float(data[16])
            yield vendor_id, revenue

    def reducer(self, key, values):
        yield None, (sum(values), key)

    def final_reducer(self, _, values):
        max_revenue, vendor_id = max(values)
        yield vendor_id, max_revenue

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

(b) mrtask_b.py

```
# Which pickup location generates the most revenue?

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

class MostRevenuePickupLocation(MRJob):

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

    def mapper(self, _, line):
        # Skip the header line
        if not line.startswith('VendorID'):
            fields = line.split(',')
            pickup_location = fields[7]
            revenue = float(fields[16])
            yield pickup_location, revenue

    def reducer(self, pickup_location, revenues):
        yield None, (sum(revenues), pickup_location)

    def final_reducer(self, _, max_revenues):
        max_revenue, pickup_location = max(max_revenues)
        yield pickup_location, max_revenue

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

(c) mrtask_c.py

```
# What are the different payment types used by customers and their count? The
final results should be in a sorted format.
from mrjob.job import MRJob
from mrjob.step import MRStep

class PaymentTypesCount(MRJob):

    def mapper(self, _, line):
        # Skip the header line
        if not line.startswith('VendorID'):
            fields = line.split(',')
            payment_type = fields[9]
            yield payment_type, 1

    def combiner(self, payment_type, counts):
        yield payment_type, sum(counts)

    def reducer(self, payment_type, counts):
        yield payment_type, sum(counts)

    def reducer_sort_results(self, payment_type, counts):
        yield None, (sum(counts), payment_type)

    def reducer_output_result(self, _, sorted_results):
        for count, payment_type in sorted(sorted_results, reverse=True):
            yield payment_type, count

    def steps(self):
        return [
            MRStep(mapper=self.mapper, combiner=self.combiner,
reducer=self.reducer),
            MRStep(reducer=self.reducer_sort_results),
            MRStep(reducer=self.reducer_output_result)
        ]

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

(d) mrtask_d.py

```
# What is the average trip time for different pickup locations?

from mrjob.job import MRJob
from datetime import datetime

class AverageTripTime(MRJob):

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

    def mapper(self, _, line):
        # Skip the header line
        if not line.startswith('VendorID'):
            fields = line.split(',')
            pickup_location = fields[7]
            pickup_datetime = self.parse_datetime(fields[1])
            dropoff_datetime = self.parse_datetime(fields[2])
            trip_time = (dropoff_datetime - pickup_datetime).total_seconds() /
60.0
            yield pickup_location, (trip_time, 1)

    def combiner(self, pickup_location, trip_times):
        total_trip_time = 0
        total_count = 0
        for trip_time, count in trip_times:
            total_trip_time += trip_time
            total_count += count
        yield pickup_location, (total_trip_time, total_count)

    def reducer(self, pickup_location, trip_times):
        total_trip_time = 0
        total_count = 0
        for trip_time, count in trip_times:
            total_trip_time += trip_time
            total_count += count
        average_trip_time = total_trip_time / total_count
        yield pickup_location, average_trip_time

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

(e) mrtask_e.py

```
# Calculate the average tips to revenue ratio of the drivers for different pickup
locations in sorted format.

from mrjob.job import MRJob

class AverageTipsToRevenueRatio(MRJob):

    def mapper(self, _, line):
        # Skip the header line
        if not line.startswith('VendorID'):
            fields = line.split(',')
            pickup_location = fields[7]
            total_revenue = float(fields[16])
            tips = float(fields[13])
            yield pickup_location, (tips, total_revenue)

    def combiner(self, pickup_location, tips_revenues):
        total_tips = 0
        total_revenue = 0
```

```
        for tips, revenue in tips_revenues:
            total_tips += tips
            total_revenue += revenue
        yield pickup_location, (total_tips, total_revenue)

    def reducer(self, pickup_location, tips_revenues):
        total_tips = 0
        total_revenue = 0
        for tips, revenue in tips_revenues:
            total_tips += tips
            total_revenue += revenue
        average_tips_to_revenue_ratio = total_tips / total_revenue
        yield pickup_location, average_tips_to_revenue_ratio

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

(f) mrtask_f.py

```
# How does revenue vary over time? Calculate the average trip revenue per month - analysing it by hour of the day (day vs night) and the day of the week (weekday vs weekend).

from mrjob.job import MRJob
from datetime import datetime

class AverageRevenueOverTime(MRJob):

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

    def mapper(self, _, line):
        # Skip the header line
        if not line.startswith('VendorID'):
            fields = line.split(',')
            revenue = float(fields[16])
            pickup_datetime = self.parse_datetime(fields[1])
            month = pickup_datetime.month
            hour = pickup_datetime.hour
            weekday = pickup_datetime.weekday()
            yield (month, hour, weekday), revenue

    def reducer(self, key, values):
        total_revenue = 0
        num_trips = 0

        for revenue in values:
            total_revenue += revenue
            num_trips += 1

        average_revenue = total_revenue / num_trips

        yield key, average_revenue

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

3) Before running these scripts we need to install mrjob by using the following commands:

```
sudo yum -y install python3-pip
sudo pip3 install mrjob
```

```
Updated:
  python3-pip.noarch 0:20.2.2-1.amzn2.0.5

Complete!
[hadoop@ip-172-31-63-235 ~]$ sudo pip3 install mrjob
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3 install --user` instead.
Collecting mrjob
  Downloading mrjob-0.7.4-py2.py3-none-any.whl (439 kB)
     |████████████████████████████████| 439 kB 21.8 MB/s
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib64/python3.7/site-packages (from mrjob) (5.3.1)
Installing collected packages: mrjob
  WARNING: The scripts mrjob, mrjob-3 and mrjob-3.7 are installed in '/usr/local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed mrjob-0.7.4
[hadoop@ip-172-31-63-235 ~]$
```

Questions:

(a) Which vendors have the most trips, and what is the total revenue generated by that vendor?

```
python mrtask_a.py yellow_tripdata_2017-05.csv > mrtask_a.txt
```

```
[hadoop@ip-172-31-63-235 ~]$ ls
mrtask_a.py  mrtask_a.txt  mrtask_b.py  mrtask_c.py  mrtask_d.py  mrtask_e.py  mrtask_f.py  yellow_tripdata_2017-05.csv
[hadoop@ip-172-31-63-235 ~]$ python mrtask_a.py yellow_tripdata_2017-05.csv > mrtask_a.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_a.hadoop.20240603.233826.200874
Running step 1 of 2...
Running step 2 of 2...
job output is in /tmp/mrtask_a.hadoop.20240603.233826.200874/output
Streaming final output from /tmp/mrtask_a.hadoop.20240603.233826.200874/output...
Removing temp directory /tmp/mrtask_a.hadoop.20240603.233826.200874...
[hadoop@ip-172-31-63-235 ~]$ cat mrtask_a.txt
"2"     92896777.54522054
[hadoop@ip-172-31-63-235 ~]$
```

Vendor “2” ie: VeriFone Inc. has the most trips and revenue generated is 92896777.545

(b) Which pickup location generates the most revenue?

```
python mrtask_b.py yellow_tripdata_2017-05.csv > mrtask_b.txt
```

```
[hadoop@ip-172-31-58-196 ~]$ python mrtask_b.py yellow_tripdata_2017-05.csv > mrtask_b.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_b.hadoop.20240604.000704.853039
Running step 1 of 2...
Running step 2 of 2...
job output is in /tmp/mrtask_b.hadoop.20240604.000704.853039/output
Streaming final output from /tmp/mrtask_b.hadoop.20240604.000704.853039/output...
Removing temp directory /tmp/mrtask_b.hadoop.20240604.000704.853039...
[hadoop@ip-172-31-58-196 ~]$ cat mrtask_b.txt
"132"   14040591.220016211
[hadoop@ip-172-31-58-196 ~]$
```

Pickup location ‘132’ generates the most revenue of 14040591.220016211

(c) What are the different payment types used by customers and their count? The final results should be in a sorted format.

```
python mrtask_c.py yellow_tripdata_2017-05.csv > mrtask_c.txt
```

```
[hadoop@ip-172-31-58-196 ~]$ python mrtask_c.py yellow_tripdata_2017-05.csv > mrtask_c.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_c.hadoop.20240604.001225.060982
Running step 1 of 3...
Running step 2 of 3...
Running step 3 of 3...
job output is in /tmp/mrtask_c.hadoop.20240604.001225.060982/output
Streaming final output from /tmp/mrtask_c.hadoop.20240604.001225.060982/output...
Removing temp directory /tmp/mrtask_c.hadoop.20240604.001225.060982...
[hadoop@ip-172-31-58-196 ~]$ cat mrtask_c.txt
"1"     6780947
"2"     3250362
"3"     55027
"4"     15791
[hadoop@ip-172-31-58-196 ~]$
```

Different payment types used by customers with their count:

"1" = Credit Card	6780947
"2" = Cash	3250362
"3" = No Charge	55027
"4" = Dispute	15791

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

```
python mrtask_d.py yellow_tripdata_2017-05.csv > mrtask_d.txt
```

```
[hadoop@ip-172-31-58-196 ~]$ c
"1"     12.681244881244885
"10"    56.4414970815586
"100"   16.301056898951547
"101"   10.8640350877193
"102"   15.480072463768114
"105"   22.727777777777778
"106"   14.046902507889056
"107"   14.914703800523975
"108"   17.747747747747745
"109"   13.047222222222222
"11"    11.318787878787878
"111"   15.347685185185185
"112"   15.342635286583413
"113"   15.655663780443364
"114"   16.265089572862763
"115"   9.849019607843136
"116"   15.213805066967888
"117"   9.020000000000001
"118"   16.645833333333336
"119"   14.638700564971753
"12"    25.1158402521525 2
"120"   12.665972222222223
"121"   16.84583333333333
"122"   11.954761904761904
"123"   12.698253968253967
"124"   35.719855072463766
"125"   16.582152788143137
"126"   32.175581395348836
"127"   17.629476052716164
"128"   15.57929292929293
"129"   14.331578947368426
"13"    20.914534001214356
"130"   37.98320691108302
"131"   11.690000000000003
"132"   46.805229740819875
"133"   17.532022849462365
"134"   16.221703296703296
"135"   15.368173258003768
"136"   9.693303571428572
"137"   14.412591446828932
"138"   41.792668197772832
"139"   18.82820512820513
"14"    18.910311918063318
"140"   14.5447834434852864
"141"   12.8877319434715 24
"142"   14.7246784344186814
"143"   13.878811307745 34
```

```
"144"   17.4224106667367306
"145"   13.9198510297940 44
"146"   15.7799488243430 21
"147"   13.0030516431924 89
"148"   16.9942420695278 64
"149"   13.1344827586206 88
"15"    14.8685897435897 45
"150"   22.0676470588235 32
"151"   13.722869615252 67
"152"   15.0342733188720 25
"153"   12.7767912772585 67
"154"   22.9303030303030 33
"155"   23.8472222222222 18
"156"   17.777777777777 78
"157"   24.965102974828 38
"158"   17.0569933219829 15
"159"   12.9778846153846 16
"16"    11.239308176100 63
"160"   17.3813996316758 76
"161"   16.5013904964861 35
"162"   15.575537437999 3
"163"   17.0072854074029 43
"164"   16.1359139096406 67
"165"   23.1466435185185 16
"166"   16.1281816414201 27
"167"   11.7317094017094 02
"168"   11.9123701355872 49
"169"   12.0082828282828 28
"17"    14.3609867193502 39
"170"   15.232808191847 37
"171"   11.8671794871794 86
"172"   29.295833333333 33
"173"   12.2473985890652 57
"174"   12.0910852713178 32
"175"   11.4327586206896 56
"176"   0.6
"177"   20.9557742782152 23
"178"   8.291562500000001
"179"   15.149970588235 3
"18"    12.6532258064516 12
"180"   27.183506944444446
"181"   15.8496037854170 22
"182"   12.2456
"183"   11.927777777777779
"184"   7.346666666666666
"185"   15.095833333333333
"186"   17.360728051286 6
"187"   1.425
"188"   15.9077920443101 69
"189"   16.6314319475776 14
"19"    10.7683544303797 46
"190"   26.4619853355893 97
"191"   13.4865740740740 74
"192"   13.5868421052631 57
"193"   12.0751834646199 35
"194"   27.3540135033758 46
"195"   22.5711945031712 45
"196"   18.1815644654088 05
"197"   18.0363026819923 43
"198"   13.8061954624781 86
"2"     42.2
"20"    17.1410919540229 86
"200"   11.8553921568627 47
"201"   0.49374999999999997
"202"   15.4358552631578 96
"203"   19.284795321637 43
"204"   0.47999999999999987
"205"   18.7130081300813 02
"206"   11.6488095238095 24
"207"   8.778993536472758
"208"   15.3559071729957 81
"209"   20.529990863407 94
"21"    16.1977987421383 64
"210"   18.4766025641025 67
"211"   17.8059098531735 78
"212"   24.377777777777773
"213"   20.407861635220 13
"214"   8.220833333333333
"215"   56.684661654135 34
"216"   34.080833333333 33
"217"   18.355075901328 27
"218"   17.6841772151898 73
"219"   53.020019157088 13
"22"    22.210856269113 15
"220"   11.6653153153153 17
"221"   19.8119047619047 63
"222"   19.394444444444442
"223"   17.1584813874788 47
"224"   14.4232187776191 41
"225"   15.3939919121894 85
"226"   17.0626620410860 35
"227"   15.5374595469255 68
"228"   18.1189009661835 74
"229"   13.7296343704521 87
"23"    3.605128205128205
"230"   18.167751779266 68
"231"   17.8429057022907 93
"232"   17.807126732249 09
"233"   16.1306186723790 82
"234"   15.7405322191143 75
"235"   21.514437367303 61
```

```
"235"    21.514443736730361
"236"    13.552552898225786
"237"    12.9437287844299447
"238"    13.635731925509624
"239"    13.7666374123862702
"24"     14.0863711798958675
"240"    16.91818181818182
"241"    15.555228758169934
"242"    13.335582822085891
"243"    17.7643825665859554
"244"    17.979727469051.31
"245"    23.330555555555552
"246"    16.1628217064104.72
"247"    16.8767734935736944
"248"    13.558333333333332
"249"    15.335284107159861
"25"     17.171431866408.33
"250"    14.828939393939397
"251"    2.1999999999999997
"252"    22.4311904761904.76
"253"    25.680555555555554
"254"    11.153439153439153
"255"    18.58691486586808
"256"    15.882166210462294
"257"    16.624087893864.01
"258"    26.2963104325699.75
"259"    10.4566666666666669
"26"     11.7172955974842.79
"260"    17.7984913423624.18
"261"    22.7423453298768.98
"262"    13.4534980216107.27
"263"    12.5589729436984.16
"264"    15.6607018783081.33
"265"    10.0978149829738.92
"27"     9.1041666666666668
"28"     33.768
"29"     57.2217741935483.84
"3"      17.919230769230.77
"30"     14.5958333333333.33
"31"     22.172916666666.67
"32"     45.239259259259.26
"33"     19.482581990149.98
"34"     15.7710125448028.65
"35"     12.9039325842696.63
"36"     16.2174234424498.42
"37"     15.677851830387.33
"38"     105.584166666666668
"39"     17.30406.25
"4"      15.9435390056985.56
"40"     15.9235117625192.49
"41"     13.0472987306555.45
```

```
"42"     13.060410811865738
"43"     16.675679763830.65
"44"     9.5041666666666666
"45"     19.5163915500982.02
"46"     29.2574074074074.06
"47"     15.2349514563106.8
"48"     15.3062372956956.65
"49"     14.443083450101.19
"5"      0.3
"50"     15.425370997048.24
"51"     11.9606837606837.6
"52"     18.2170888399412.62
"53"     20.5872395833333.3
"54"     14.7193355119825.68
"55"     47.3187265917603
"56"     23.8063291139240.47
"57"     15.6166666666666.67
"58"     5.3333333333333.3
"59"     19.6166666666666.64
"6"      7.7902777777777.77
"60"     12.9615591397849.44
"61"     15.2518304814962.28
"62"     13.879747675962.818
"63"     65.073039215686.26
"64"     11.291071428571.43
"65"     18.1567927449679.23
"66"     18.577475060199.52
"67"     13.0534313725490.17
"68"     15.8596933872839.83
"69"     19.924176245210.73
"7"      13.9847439971751.42
"70"     26.377022024801.04
"71"     12.4551724137931.02
"72"     17.4637931034482.75
"73"     22.723684210526.315
"74"     12.5256448050367.02
"75"     13.5190759925889.38
"76"     18.7018719806763.27
"77"     16.114341085271.32
"78"     11.4487538940809.96
"79"     15.6507152206594.71
"8"      22.9693146417445.53
"80"     16.7225096679557.16
"81"     10.016666666666.67
"82"     15.1632763628239.48
"83"     16.827437223042.836
"84"     5.575
"85"     15.7237311385459.56
"86"     14.700000000000.01
"87"     22.175754686604.86
"88"     23.095660492322.71
"89"     17.877146332985.75
"9"      153.699275362318.84
"90"     14.5886250229019.05
"91"     23.282697201017.81
"92"     14.8419517102615.7
"93"     38.069672131147.55
"94"     11.6473809523809.51
"95"     18.8484673024523.17
"96"     18.588888888888.85
"97"     16.0816185636206.73
"98"     13.002500000000.01
[hadoop@ip-172-31-58-196 ~]$
```

(e) Calculate the average tips to revenue ratio of the drivers for different pickup locations in sorted format.

```
python mrtask_e.py yellow_tripdata_2017-05.csv > mrtask_e.txt
```

```
[hadoop@ip-172-31-58-196 ~]$ ls
mrtask_a.py  mrtask_b.txt  mrtask_c.txt  mrtask_d.txt  mrtask_f.py                yellow_tripdata_2017-05.csv.1
mrtask_b.py  mrtask_c.py   mrtask_d.py   mrtask_e.py   yellow_tripdata_2017-05.csv
[hadoop@ip-172-31-58-196 ~]$ python mrtask_e.py yellow_tripdata_2017-05.csv > mrtask_e.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_e.hadoop.20240604.011220.382512
Running step 1 of 1...
job output is in /tmp/mrtask_e.hadoop.20240604.011220.382512/output
Streaming final output from /tmp/mrtask_e.hadoop.20240604.011220.382512/output...
Removing temp directory /tmp/mrtask_e.hadoop.20240604.011220.382512...
[hadoop@ip-172-31-58-196 ~]$
```

```
[hadoop@ip-172-31-58-196 ~]$ cat mrtask_e.txt
"1"     0.11961918264336044
"10"    0.1031314953272488
"100"   0.10054811858030266
"101"   0.1538484771604267
"102"   0.09686867923894887
"105"   0.07175925925925926
"106"   0.11203391656570175
"107"   0.11929564967810695
"108"   0.12120884627098828
"109"   0.23737785016286642
"11"    0.0581598846488283 26
"111"   0.077659754354142
"112"   0.10900187462682069
"113"   0.11769752457713911
"114"   0.11571668143567702
"115"   0.1014877461571702
"116"   0.09140144834181262
"117"   0.0298651911665509 88
"118"   0.03324247058041353
"119"   0.06439098561727422
"12"    0.09033344274559964
"120"   0.14485426649668087
"121"   0.06025386778698323
"122"   0.0352749370090410 45
"123"   0.0750228876958997
"124"   0.09886020334918795
"125"   0.1223467893609015
"126"   0.05493403420192623
"127"   0.08833111068775
"128"   0.10511320192841678
"129"   0.06647065796213625
"13"    0.12390141102889671
"130"   0.1073558396219219
"131"   0.10384807204137783
"132"   0.1025660513460536
"133"   0.09147281347725217
"134"   0.10314438578934823
"135"   0.0785122214655781
"136"   0.032600578121018825
"137"   0.11253077758026406
"138"   0.13048344466631976
"139"   0.0252788454736246 37
```

```
"140"	0.11107044470853594
"141"	0.11186974122790275
"142"	0.1131558975317282
"143"	0.11473162975887714
"144"	0.11052727216740073
"145"	0.09866671202718473
"146"	0.09229805528093973
"147"	0.046714304274392 63
"148"	0.11347366987186253
"149"	0.12259087582337157
"15"	0.12542820971105154
"150"	0.10633375877472878
"151"	0.10794854174655912
"152"	0.084972137978292 44
"153"	0.08409909578769388
"154"	0.07400473993394348
"155"	0.09130948944530835
"156"	0.07496251874062969
"157"	0.11154334990834797
"158"	0.1177428105347338
"159"	0.04880437380957894
"16"	0.1296825635625094
"160"	0.09369565599255213
"161"	0.09989106877197958
"162"	0.11962265897953572
"163"	0.11101476605573193
"164"	0.11138449806107008
"165"	0.07196634804086764
"166"	0.11344656333427505
"167"	0.03560427740254 66
"168"	0.0504102777807923884
"169"	0.11765956298165817
"17"	0.08304806476027186
"170"	0.11737627919224174
"171"	0.04416689447785676
"172"	0.27196083763937995
"173"	0.0662888212150767
"174"	0.09292588232864547
"175"	0.08816570904350345
"176"	0.19982277359326542
"177"	0.09851238280057963
"178"	0.03941897087008663
"179"	0.08647112167836499
"18"	0.07413049841666887
"180"	0.10306335321889983
"181"	0.11383678943060177
"182"	0.04935382990636337
"183"	0.0405831074646546 1
"184"	0.03374233128834356
"185"	0.07860584480628201
"186"	0.10968385132719934
"187"	0.032113037893384 71
"188"	0.07172143661981017
"189"	0.10932535246060386
"19"	0.11032972645756875
"190"	0.10117750519544694
"191"	0.05438239409744 8308
"192"	0.0720589495330306
"193"	0.03922346473669742
"194"	0.12076731965739748
"195"	0.09492841545380133
"196"	0.09427848086047703
"197"	0.10215364731577099
"198"	0.10078643815066789
"2"	0.11989767545323099
"20"	0.0736013986013986
"200"	0.131876061662643 48
"201"	0.1320447135537 96
"202"	0.09899140045991586
"203"	0.04808741627154 3615
"204"	0.0684931506849315
"205"	0.07523246599344224
"206"	0.08833170762908359
"207"	0.0815078125388226
"208"	0.08291166272142231
"209"	0.11297948250644907
"21"	0.06386360722984234
"210"	0.07934380776340108
"211"	0.11310320504679515
"212"	0.06409624702307627
"213"	0.11780938978546256
"214"	0.02993836219547989
"215"	0.10353565292295142
"216"	0.11008942973165864
"217"	0.07868790199337758
"218"	0.19235819846724786
"219"	0.1049626881741011
"22"	0.05041866234795492
"220"	0.056836272960188075
"221"	0.108447796031959 78
"222"	0.11278732656864776
"223"	0.09822645175828154
"224"	0.11425442486544808
"225"	0.08099702729092222
"226"	0.08354577861128779
"227"	0.08721017303002598
"228"	0.09978962780363482
"229"	0.11203502738122283
"23"	0.08146106133700898
"230"	0.10662429938743652
"231"	0.11828076056384894
```

```
"232"   0.10577578661446632
"233"   0.11514074932415078
"234"   0.12096822968790891
"235"   0.0557415553911836
"236"   0.11249379678942588
"237"   0.1096331664258742
"238"   0.10834063914173245
"239"   0.11534544071014695
"24"    0.106481262585554 79
"240"   0.10447949050358085
"241"   0.06993302518630319
"242"   0.07890211996979177
"243"   0.09530565669975881
"244"   0.09546944114505117
"245"   0.0
"246"   0.11995315301676546
"247"   0.073359849717676 94
"248"   0.0526239594643503 34
"249"   0.11923576897943587
"25"    0.11052464311894397
"250"   0.043282871250452847
"251"   0.0019544811624019544
"252"   0.085582044237591 97
"253"   0.06711331976762965
"254"   0.0972480119905075
"255"   0.116010049615174 14
"256"   0.10930439934227 45
"257"   0.10482648999719887
"258"   0.157864611374004 68
"259"   0.12318663572582067
"26"    0.053321392496016 85
"260"   0.072052292129307 92
"261"   0.10801526749738075
"262"   0.113806135601148 87
"263"   0.11426084950231 24
"264"   0.11230316290901997
"265"   0.109689117317854 66
"27"    0.070448129279293 89
"28"    0.103159286076791 81
"29"    0.044969557780409 33
"3"     0.03522695854408 25
"30"    0.431189363995688
"31"    0.092414574413141 65
"32"    0.073886688446230 49
"33"    0.12042367252672 28
"34"    0.114691058428224 79
"35"    0.175818287711123 98
"36"    0.104129704273601 84
"37"    0.10343807086659 2
"38"    0.038324782245555 42
"39"    0.116073134993947 89
```

```
"39"    0.116073134993947 89
"4"     0.10621226787883002
"40"    0.11953106379554157
"41"    0.092745844181535 07
"42"    0.070969659205467 44
"43"    0.104078351120434 34
"44"    0.07601977750309023
"45"    0.09838461474617 19
"46"    0.12534064929126656
"47"    0.033407120457480 17
"48"    0.105827588088558 66
"49"    0.096810131513518 17
"5"     0.71115973774179 43
"50"    0.109239198358364 89
"51"    0.054687879641958155
"52"    0.130558045767348 64
"53"    0.09939808622286242
"54"    0.119687174700484 43
"55"    0.107758810954654 12
"56"    0.091876569586566 26
"57"    0.09823938822176 96
"58"    0.023456558453743663
"59"    0.022958827634333566
"6"     0.046688507818970 14
"60"    0.028830566964685292
"61"    0.096145495477512 58
"62"    0.072578876279971 88
"63"    0.074300082930139 12
"64"    0.077779531799905 28
"65"    0.10648853888099 56
"66"    0.114748739602466 52
"67"    0.080161361573226 79
"68"    0.11431798620169 06
"69"    0.038880239956734 77
"7"     0.080553309992971 21
"70"    0.105587196189557 61
"71"    0.057126398156593525
"72"    0.129794823865610 36
"73"    0.066602733157516 19
"74"    0.073093320940942 13
"75"    0.092338544741582 83
"76"    0.156999728236253 33
"77"    0.037589907815713 29
"78"    0.033406620697715 19
"79"    0.117260237357942 65
"8"     0.11288909783738 31
"80"    0.112301995930866 34
"81"    0.13141406281993 53
"82"    0.06402525883541 94
```

```
"83"    0.0606475767711211555
"84"    0.00066415762674341 37
"85"    0.0568739741017691
"86"    0.03836325514349365
"87"    0.12371478220911764
"88"    0.11535120965220538
"89"    0.08902658805206363
"9"     0.07875511622519114
"90"    0.11768850776024835
"91"    0.10308391768201007
"92"    0.056615381413745704
"93"    0.10714443449218 23
"94"    0.038084346952212626
"95"    0.07928530356245143
"96"    0.09975598325818912
"97"    0.09868461512594295
"98"    0.06067388225420083
[hadoop@ip-172-31-58-196 ~]$
```

(f) How does revenue vary over time? Calculate the average trip revenue per month - analysing it by hour of the day (day vs night) and the day of the week (weekday vs weekend).

```
python mrtask_f.py yellow_tripdata_2017-05.csv > mrtask_f.txt
```

```
[hadoop@ip-172-31-58-196 ~]$ python mrtask_f.py yellow_tripdata_2017-05.csv > mrtask_f.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_f.hadoop.20240604.013608.691540
Running step 1 of 1...
job output is in /tmp/mrtask_f.hadoop.20240604.013608.691540/output
Streaming final output from /tmp/mrtask_f.hadoop.20240604.013608.691540/output...
Removing temp directory /tmp/mrtask_f.hadoop.20240604.013608.691540...
[hadoop@ip-172-31-58-196 ~]$ cat mrtask_f.txt
[5, 0, 0]       19.09149097120672
[5, 0, 1]       19.42010595117413
[5, 0, 2]       17.6861589866372 74
[5, 0, 3]       17.25599944664103
[5, 0, 4]       18.099999027301898
[5, 0, 5]       17.105128026116333
[5, 0, 6]       15.658808411947584
[5, 1, 0]       17.87588900705367
[5, 1, 1]       19.429909785928686
[5, 1, 2]       17.373021160869822
[5, 1, 3]       16.436848288253056
[5, 1, 4]       18.217479700823546
[5, 1, 5]       16.275959370214274
[5, 1, 6]       15.1412930674054 17
[5, 10, 0]      16.241741228653577
[5, 10, 1]      16.639627234011154
[5, 10, 2]      27.168365753878387
[5, 10, 3]      16.499299606770162
[5, 10, 4]      16.486803455065 14
[5, 10, 5]      14.029692293163937
[5, 10, 6]      14.908801744148995
[5, 11, 0]      16.680433054406084
[5, 11, 1]      17.194492364612 42
[5, 11, 2]      16.92334503331534
[5, 11, 3]      18.596232180453956
[5, 11, 4]      16.728098769455887
[5, 11, 5]      14.021549507826318
[5, 11, 6]      14.98238805727666
[5, 12, 0]      16.480360929345835
[5, 12, 1]      17.22963705731718
```

```
[5, 12, 2]      17.055630954941162
[5, 12, 3]      17.9757300537716557
[5, 12, 4]      17.69103689287293
[5, 12, 5]      14.4997920550207334
[5, 12, 6]      15.27335388620518
[5, 13, 0]      16.795085865627826
[5, 13, 1]      17.279482350894487
[5, 13, 2]      17.7748277620483346
[5, 13, 3]      18.5956738254752238
[5, 13, 4]      18.2054082897937773
[5, 13, 5]      14.99034880898968
[5, 13, 6]      15.9376012586608894
[5, 14, 0]      17.208584403435885
[5, 14, 1]      17.347854799759338
[5, 14, 2]      18.1109613533840074
[5, 14, 3]      19.0334940816199924
[5, 14, 4]      18.6133879091461944
[5, 14, 5]      15.3365844854614453
[5, 14, 6]      17.1288964891697266
[5, 15, 0]      16.915001820667988
[5, 15, 1]      17.3076824517917745
[5, 15, 2]      17.757653072067977
[5, 15, 3]      18.7507890423560822
[5, 15, 4]      18.4796808458925855
[5, 15, 5]      15.5094643188248422
[5, 15, 6]      17.0565924245776166
[5, 16, 0]      18.261410131236254
[5, 16, 1]      18.8599406481163822
[5, 16, 2]      19.65427089169584
[5, 16, 3]      20.048859995777594
[5, 16, 4]      19.8347954541417054
[5, 16, 5]      15.6357477542431984
[5, 16, 6]      17.324058364775594
[5, 17, 0]      17.5559594925070644
[5, 17, 1]      17.7472419626020084
[5, 17, 2]      18.39607183396174
[5, 17, 3]      18.8877325093480274
[5, 17, 4]      18.4119631373495524
[5, 17, 5]      15.3593093037344264
[5, 17, 6]      17.443964534140774
[5, 18, 0]      16.5002034073240474
[5, 18, 1]      16.4771606313639124
[5, 18, 2]      17.0537790352401084
[5, 18, 3]      17.2216042635706834
[5, 18, 4]      16.837623568682494
[5, 18, 5]      14.732677980361854
[5, 18, 6]      16.3638792079012954
[5, 19, 0]      16.1170712067446054
[5, 19, 1]      15.9333382793836164
[5, 19, 2]      16.2706201258592784
[5, 19, 3]      16.4723987182292264
```

```
[5, 19, 4]      16.1614230769372264
[5, 19, 5]      14.4867190406634494
[5, 19, 6]      16.2467800241747654
[5, 2, 0]       17.0693025604013544
[5, 2, 1]       17.1029995680332654
[5, 2, 2]       16.4907249005769044
[5, 2, 3]       15.8855274112730014
[5, 2, 4]       17.5101861859099164
[5, 2, 5]       15.7878222921786554
[5, 2, 6]       15.2198288791500244
[5, 20, 0]      16.2783298898202244
[5, 20, 1]      15.922293130064014
[5, 20, 2]      15.9992839471349474
[5, 20, 3]      16.3632360840257584
[5, 20, 4]      16.041364614564294
[5, 20, 5]      15.0082975074762714
[5, 20, 6]      16.5148453257117544
[5, 21, 0]      16.817749214698184
[5, 21, 1]      16.3880158281661434
[5, 21, 2]      16.2314065281509774
[5, 21, 3]      17.1210413825646384
[5, 21, 4]      16.5793831725745274
[5, 21, 5]      15.0301044273078854
[5, 21, 6]      16.834085322670274
[5, 22, 0]      17.3376232680181954
[5, 22, 1]      17.003566938163824
[5, 22, 2]      16.7307500025162064
[5, 22, 3]      17.6998941195175734
[5, 22, 4]      16.729303921215954
[5, 22, 5]      15.1617176035073844
[5, 22, 6]      17.477475480270944
[5, 23, 0]      18.7341667825235274
[5, 23, 1]      17.3640876633350884
[5, 23, 2]      17.2528831344362644
[5, 23, 3]      18.146971367782044
[5, 23, 4]      17.2480780890789624
[5, 23, 5]      15.8460464194004554
[5, 23, 6]      18.864908675805184
[5, 3, 0]       17.0104368334696264
[5, 3, 1]       17.3890162419132824
[5, 3, 2]       16.622625941245024
[5, 3, 3]       16.376610241669374
[5, 3, 4]       17.6797390012709634
[5, 3, 5]       16.1475487935183644
[5, 3, 6]       15.8220569112765354
[5, 4, 0]       21.2929997252974174
[5, 4, 1]       20.0689070069708544
[5, 4, 2]       19.7325072886281884
[5, 4, 3]       20.8299688195979444
[5, 4, 4]       20.9156692726814664
[5, 4, 5]       17.8254468102896484
```

```
[5, 4, 6]       17.38188976038157
[5, 5, 0]       21.593018355659638
[5, 5, 1]       18.81002788103778
[5, 5, 2]       18.894916666663246
[5, 5, 3]       19.845702192891927
[5, 5, 4]       21.227005213121405
[5, 5, 5]       20.992848200309997
[5, 5, 6]       21.125313003037945
[5, 6, 0]       16.36666989228269
[5, 6, 1]       14.9783458597215552
[5, 6, 2]       14.767823020740611
[5, 6, 3]       15.288311117881104
[5, 6, 4]       16.3898971231125565
[5, 6, 5]       19.32208822201787
[5, 6, 6]       19.992739240104342
[5, 7, 0]       15.307352827781946
[5, 7, 1]       14.68863340021126
[5, 7, 2]       14.46050088621336
[5, 7, 3]       14.619068724817286
[5, 7, 4]       14.939835919422332
[5, 7, 5]       16.3001974491006637
[5, 7, 6]       18.066767322725
[5, 8, 0]       15.534355540424196
[5, 8, 1]       15.310034719953286
[5, 8, 2]       14.9919551125437114
[5, 8, 3]       15.0955954437936711
[5, 8, 4]       15.1448767885095223
[5, 8, 5]       14.67262836614724
[5, 8, 6]       15.6851009797289776
[5, 9, 0]       15.79006959000546
[5, 9, 1]       15.8092530199812955
[5, 9, 2]       15.622612773623201
[5, 9, 3]       15.7640767490044406
[5, 9, 4]       15.9120082469571044
[5, 9, 5]       13.8146437522416844
[5, 9, 6]       14.7797136181899966
[hadoop@ip-172-31-58-196 ~]$
```