For task 4, Dataset used is: https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-05.csv

MapReduce codes to perform the tasks using the files you've downloaded on

your EMR Instance:

```
[hadoop@ip-172-31-38-229 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow
_tripdata_2017-05.csv
--2023-12-07 11:16:30--  https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata
_2017-05.csv
Resolving nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)... 3
.5.19.132, 3.5.25.155, 16.182.37.185, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|
3.5.19.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 951965526 (908M) [text/csv]
Saving to: 'yellow_tripdata_2017-05.csv'

100%[=====================================>] 951,965,526 25.3MB/s   in 39s

2023-12-07 11:17:10 (23.1 MB/s) - 'yellow_tripdata_2017-05.csv' saved [951965526
/951965526]

[hadoop@ip-172-31-38-229 ~]$ vi mrtask a.py
```

MRJob is installed

```
[hadoop@ip-172-31-38-229 ~]$ pip install --user mrjob
Collecting mrjob
  Using cached https://files.pythonhosted.org/packages/8e/58/fc28ab743aba16e9073
5ad4e29694bd2adaf7b879376ff149306d50c4e90/mrjob-0.7.4-py2.py3-none-any.whl
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib64/python3.7/site-p
ackages (from mrjob)
Installing collected packages: mrjob
Successfully installed mrjob-0.7.4
```

Python script for :

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

```python
# 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()
```

Running the script using the input file and storing output in atext.txt

```
[hadoop@ip-172-31-38-229 ~]$ python mrtask_a.py yellow_tripdata_2017-05.csv > at
ext.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_a.hadoop.20231207.112237.174525
Running step 1 of 2...
Running step 2 of 2...
job output is in /tmp/mrtask_a.hadoop.20231207.112237.174525/output
Streaming final output from /tmp/mrtask_a.hadoop.20231207.112237.174525/output..
.
Removing temp directory /tmp/mrtask_a.hadoop.20231207.112237.174525...
```

Displaying the contents of atext.txt

```
hadoop@ip-172-31-38-229 ~]$ cat atext.txt
2"      92896777.54522054
hadoop@ip-172-31-38-229 ~]$ ^C
hadoop@ip-172-31-38-229 ~]$
```

Which pickup location generates the most revenue?

```python
# 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()
```

Running the script and displaying result

```
hadoop@ip-172-31-38-229 ~]$ python mrtask_b.py yellow_tripdata_2017-05.csv > btext.txt
o configs found; falling back on auto-configuration
o configs specified for inline runner
reating temp directory /tmp/mrtask_b.hadoop.20231207.114342.748107
unning step 1 of 2...
unning step 2 of 2...
ob output is in /tmp/mrtask_b.hadoop.20231207.114342.748107/output
treaming final output from /tmp/mrtask_b.hadoop.20231207.114342.748107/output...
emoving temp directory /tmp/mrtask_b.hadoop.20231207.114342.748107...
hadoop@ip-172-31-38-229 ~]$ cat btext.txt
132"   14040591.220016211
hadoop@ip-172-31-38-229 ~]$
```

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

```python
# 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__':
```

Running the script and displaying the result

```
[hadoop@ip-172-31-34-221 ~]$ python mrtask_c.py yellow_tripdata_2017-05.csv > mr
task_c.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_c.hadoop.20230711.082928.304167
Running step 1 of 3...
Running step 2 of 3...
Running step 3 of 3...
job output is in /tmp/mrtask_c.hadoop.20230711.082928.304167/output
Streaming final output from /tmp/mrtask_c.hadoop.20230711.082928.304167/output..
.
Removing temp directory /tmp/mrtask_c.hadoop.20230711.082928.304167...
[hadoop@ip-172-31-34-221 ~]$ cat mrtask_c.txt
"1"     6780947
"2"     3250362
"3"     55027
"4"     15791
```

What is the average trip time for different pickup locations?

```python
# 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
```

```python
    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()
```

Running the script and displaying the results

```
[hadoop@ip-172-31-38-229 ~]$ vi mrtask_c.py
[hadoop@ip-172-31-38-229 ~]$ python mrtask_c.py yellow_tripdata_2017-05.csv > ctext.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_c.hadoop.20231207.115023.014962
Running step 1 of 3...
Running step 2 of 3...
Running step 3 of 3...
job output is in /tmp/mrtask_c.hadoop.20231207.115023.014962/output
Streaming final output from /tmp/mrtask_c.hadoop.20231207.115023.014962/output...
Removing temp directory /tmp/mrtask_c.hadoop.20231207.115023.014962...
[hadoop@ip-172-31-38-229 ~]$ cat ctext.txt
"1"     6780947
"2"     3250362
"3"     55027
"4"     15791
[hadoop@ip-172-31-38-229 ~]$
```

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

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

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()
~
```

```
[hadoop@ip-172-31-37-41 ~]$ vi mrtask_e.py
[hadoop@ip-172-31-37-41 ~]$ python mrtask_e.py yellow_tripdata_2017-05.csv > etext.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_e.hadoop.20231207.154053.080410
Running step 1 of 1...
job output is in /tmp/mrtask_e.hadoop.20231207.154053.080410/output
Streaming final output from /tmp/mrtask_e.hadoop.20231207.154053.080410/output...
Removing temp directory /tmp/mrtask_e.hadoop.20231207.154053.080410...
[hadoop@ip-172-31-37-41 ~]$ cat etext.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"    0.058159884648828326
"111"   0.077659754354142
"112"   0.10900187462682069
"113"   0.11769752457713911
"114"   0.11571668143567702
"115"   0.1014877461571702
"116"   0.09140144834181262
"117"   0.029865191166550988
"118"   0.03324247058041353
"119"   0.06439098561727422
"12"    0.09033344274559964
```

"250"	0.043282871250452847
"251"	0.0019544811624019544
"252"	0.08558204423759197
"253"	0.06711331976762965
"254"	0.0972480119905075
"255"	0.11601004961517414
"256"	0.1093043993422745
"257"	0.10482648999719887
"258"	0.15786461137400468
"259"	0.12318663572582067
"26"	0.05332139249601685
"260"	0.07205229212930792
"261"	0.10801526749738075
"262"	0.11380613560114887
"263"	0.1142608495023124
"264"	0.11230316290901997
"265"	0.10968911731785466
"27"	0.07044812927929389
"28"	0.10315928607679181
"29"	0.04496955778040933
"3"	0.0352269585440825
"30"	0.431189363995688
"31"	0.09241457441314165
"32"	0.07388668844623049
"33"	0.1204236725267228
"34"	0.11469105842822479
"35"	0.17581828771112398
"36"	0.10412970427360184
"37"	0.103438070866592
"38"	0.03832478224555542
"39"	0.11607313499394789
"4"	0.106212267878830002
"40"	0.11953106379554157
"41"	0.09274584418153507
"42"	0.07096965920546744
"43"	0.10407835112043434
"44"	0.07601977750309023
"45"	0.0983846147461719
"46"	0.12534064929126656
"47"	0.03340712045748017
"48"	0.10582758808855866
"49"	0.09681013151351817
"5"	0.7111597374179 43
"50"	0.10923919835836489
"51"	0.054687879641958155
"52"	0.13055804576734864
"53"	0.09939808622286242
"54"	0.11968717470048443
"55"	0.10775881095465412

```
"55"    47.3187265917603
"56"    23.806329113924047
"57"    15.616666666666667
"58"    5.333333333333333
"59"    19.616666666666664
"6"     7.790277777777777
"60"    12.961559139784944
"61"    15.251830481496228
"62"    13.879747675962818
"63"    65.07303921568626
"64"    11.29107142857143
"65"    18.156792744967923
"66"    18.57747506019952
"67"    13.053431372549017
"68"    15.859693387283983
"69"    19.92417624521073
"7"     13.984743997175142
"70"    26.37702202480104
"71"    12.455172413793102
"72"    17.463793103448275
"73"    22.723684210526315
"74"    12.525644805036702
"75"    13.519075992588938
"76"    18.701871980676327
"77"    16.11434108527132
"78"    11.448753894080996
"79"    15.650715220659471
"8"     22.969314641744553
"80"    16.722509667955716
"81"    10.01666666666667
"82"    15.163276362823948
"83"    16.827437223042836
"84"    5.575
"85"    15.723731138545956
"86"    14.700000000000001
"87"    22.17575468660486
"88"    23.09566049232271
"89"    17.87714633298575
"9"     153.69927536231884
"90"    14.588625022901905
"91"    23.28269720101781
"92"    14.84195171026157
"93"    38.06967213114755
"94"    11.647380952380951
"95"    18.848467302452317
"96"    18.588888888888885
"97"    16.081618563620673
"98"    13.002500000000001
[hadoop@ip-172-31-37-41 ~]$
```

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
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()
```

Run the script and display result

```
[hadoop@ip-172-31-37-41 ~]$ vi mrtask_f.py
[hadoop@ip-172-31-37-41 ~]$ python mrtask_f.py yellow_tripdata_2017-05.csv > ftext.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_f.hadoop.20231207.155043.788897
Running step 1 of 1...
job output is in /tmp/mrtask_f.hadoop.20231207.155043.788897/output
Streaming final output from /tmp/mrtask_f.hadoop.20231207.155043.788897/output...
Removing temp directory /tmp/mrtask_f.hadoop.20231207.155043.788897...
[hadoop@ip-172-31-37-41 ~]$
[hadoop@ip-172-31-37-41 ~]$ cat ftext.txt
[5, 0, 0]       19.09149097120672
[5, 0, 1]       19.42010595117413
[5, 0, 2]       17.686158986637274
[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]       15.141293067405417
[5, 10, 0]      16.241741228653577
[5, 10, 1]      16.639627234011154
[5, 10, 2]      27.168365753878387
[5, 10, 3]      16.499299606770162
[5, 10, 4]      16.48680345506514
[5, 10, 5]      14.029692293163937
[5, 10, 6]      14.908801744148995
[5, 11, 0]      16.680433054406084
[5, 11, 1]      17.19449236461242
[5, 11, 2]      16.92334503331534
[5, 11, 3]      18.596232180453956
```

```
[5, 3, 1]       17.389016241913282
[5, 3, 2]       16.62262594124502
[5, 3, 3]       16.37661024166937
[5, 3, 4]       17.679739001270963
[5, 3, 5]       16.147548793518364
[5, 3, 6]       15.822056911276535
[5, 4, 0]       21.292999725297417
[5, 4, 1]       20.068907006970854
[5, 4, 2]       19.732507288628188
[5, 4, 3]       20.829968819597944
[5, 4, 4]       20.915669272681466
[5, 4, 5]       17.825446810289648
[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]       14.978345859721552
[5, 6, 2]       14.767823020740611
[5, 6, 3]       15.288311117881104
[5, 6, 4]       16.389897123112565
[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]       16.300197449100637
[5, 7, 6]       18.066767322725
[5, 8, 0]       15.534355540424196
[5, 8, 1]       15.310034719953286
[5, 8, 2]       14.991955112543714
[5, 8, 3]       15.095595443793671
[5, 8, 4]       15.144876788509523
[5, 8, 5]       14.67262836614724
[5, 8, 6]       15.685100979728976
[5, 9, 0]       15.79006959000546
[5, 9, 1]       15.809253019981295
[5, 9, 2]       15.622612773623201
[5, 9, 3]       15.764076749004406
[5, 9, 4]       15.912008246957104
[5, 9, 5]       13.814643752241684
[5, 9, 6]       14.779713618189996
[hadoop@ip-172-31-37-41 ~]$
```