MapReduce Tasks

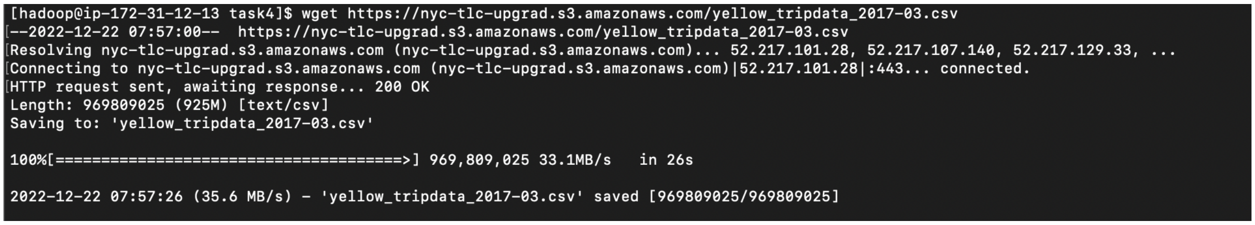
In this task of Mar Reduce programming assignment, we have written code in python using mrjob library.

Before running code on EMR, we need to download csv files in our EMR directory. We have achieved this step using wget command, as follows:

wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-03.csv
wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-04.csv
wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-05.csv
wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-06.csv

NOTE: As we have less space in EMR, owing to this we have runned all our code with only one data file. But same code can also run with more data files

Screenshot:



```
[hadoop@ip-172-31-12-13 task4]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-03.csv
[--2022-12-22 07:57:00--  https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-03.csv
[Resolving nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)... 52.217.101.28, 52.217.107.140, 52.217.129.33, ...
[Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|52.217.101.28|:443... connected.
[HTTP request sent, awaiting response... 200 OK
Length: 969809025 (925M) [text/csv]
Saving to: 'yellow_tripdata_2017-03.csv'

100%[=====================================>] 969,809,025 33.1MB/s   in 26s

2022-12-22 07:57:26 (35.6 MB/s) - 'yellow_tripdata_2017-03.csv' saved [969809025/969809025]
```

Now, let’s look into each task one by one:

Task A:-

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

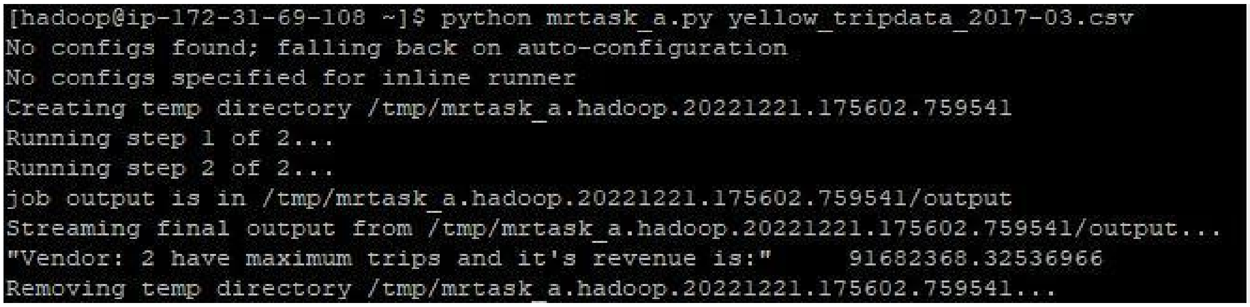
Solution:

- The program has three main parts: the mapper function, the reducer function, and the final reducer function.
- The mapper function processes each line in the input data and extracts the vendor ID and total amount fields. It then emits the vendor ID and total amount as a tuple.
- The reducer function receives a vendor ID and a list of tuples containing the count of trips and the total amount for that vendor.
- It sums the count of trips and total amount for each vendor and emits the vendor ID, total number of trips, and total revenue.

- The final reducer function receives a key (which is an empty string in this case) and a list of tuples containing vendor ID, total number of trips, and total revenue.
- It finds the vendor with the most trips and the total revenue generated by that vendor and emits a string with the vendor ID and the total revenue.
- The program defines a steps function that specifies the sequence of steps to execute in the MapReduce job, including the mapper and reducer functions, and the final reducer function.

To run the script type: python mrtask_a.py <input-file-path> > <output-file-path>

Output: Vendor 2 have maximum trips and it’s revenue is: 91682368.759541



```
[hadoop@ip-172-31-69-108 ~]$ python mrtask_a.py yellow_tripdata_2017-03.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_a.hadoop.20221221.175602.759541
Running step 1 of 2...
Running step 2 of 2...
job output is in /tmp/mrtask_a.hadoop.20221221.175602.759541/output
Streaming final output from /tmp/mrtask_a.hadoop.20221221.175602.759541/output...
"Vendor: 2 have maximum trips and it's revenue is:"     91682368.32536966
Removing temp directory /tmp/mrtask_a.hadoop.20221221.175602.759541...
```

Task B:-

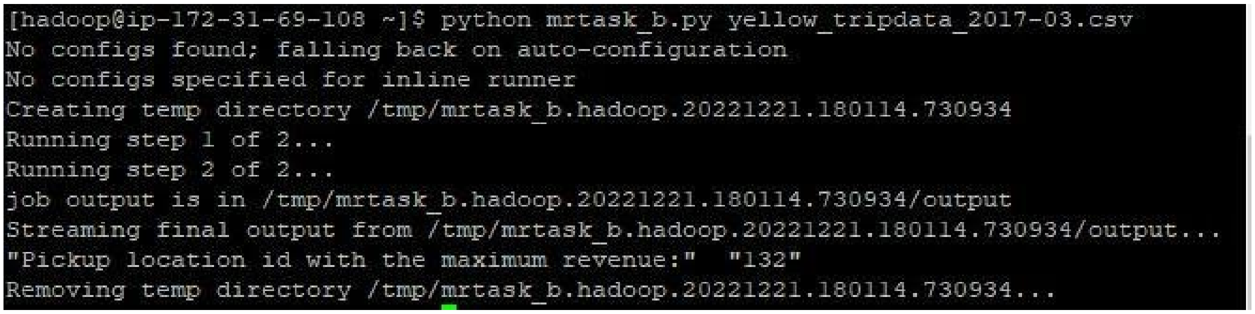
Question: Which pickup location generates the most revenue?

Solution:

- The job consists of two steps.
- In the first step, the mapper function processes each input line, which represents a single taxi trip.
- The mapper extracts the pickup location ID and total amount fields from the input line and emits them as a tuple.
- The reducer function receives a list of total amounts for each pickup location ID and computes the total revenue for that pickup location.
- The reducer then emits the total revenue and pickup location ID as a tuple.
- In the second step, the final reducer function receives a list of tuples containing the total revenue and pickup location ID for each pickup location.
- The final reducer function iterates through the list of tuples and finds the pickup location with the maximum revenue.
- Finally, the final reducer function emits the pickup location ID with the maximum revenue.

To run the script type: python mrtask_b.py <input-file-path> > <output-file-path>

Output: Pickup location id with the maximum revenue: 132



```
[hadoop@ip-172-31-69-108 ~]$ python mrtask_b.py yellow_tripdata_2017-03.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_b.hadoop.20221221.180114.730934
Running step 1 of 2...
Running step 2 of 2...
job output is in /tmp/mrtask_b.hadoop.20221221.180114.730934/output
Streaming final output from /tmp/mrtask_b.hadoop.20221221.180114.730934/output...
"Pickup location id with the maximum revenue:"  "132"
Removing temp directory /tmp/mrtask_b.hadoop.20221221.180114.730934...
```

Task C:-

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

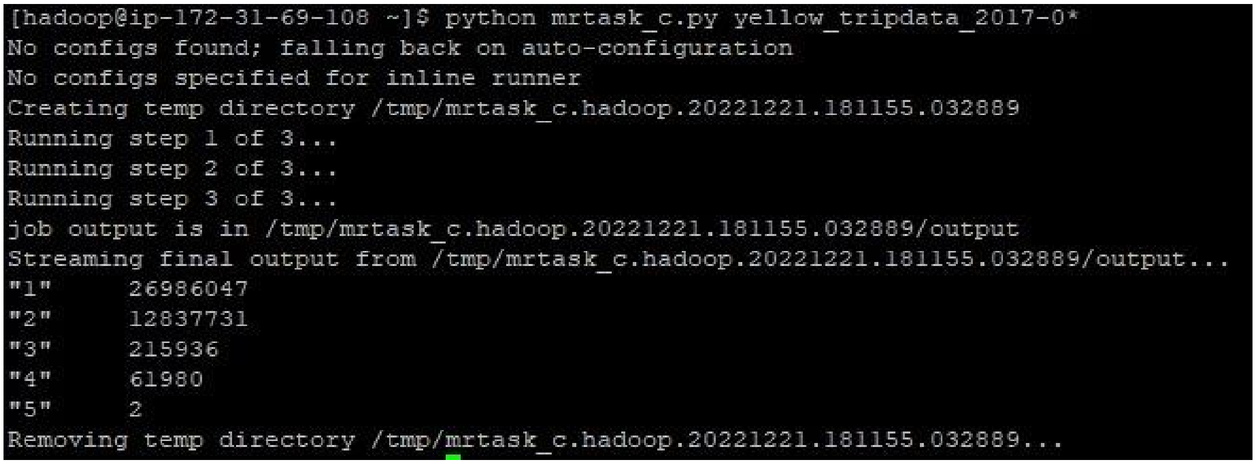
Solution:

- The program is implemented using the MRJob library and consists of three steps:
- The first step is a mapping step that processes each line of the input data. The mapper function extracts the payment type field from each line and emits the payment type as the key and a value of 1.
- The second step is a reducing step that counts the number of occurrences of each payment type. The reducer function sums the counts for each payment type and emits the payment type and total count.
- The third step is another reducing step that sorts the payment types by count in descending order. The reducer function in this step receives a list of payment type and count tuples, sorts them by count in descending order, and emits each payment type and count.

To run the script type: python mrtask_c.py <input-file-path> > <output-file-path>

Output:

1	26986047
2	12837731
3	215936
4	61980
5	2



```
[hadoop@ip-172-31-69-108 ~]$ python mrtask_c.py yellow_tripdata_2017-0*
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_c.hadoop.20221221.181155.032889
Running step 1 of 3...
Running step 2 of 3...
Running step 3 of 3...
job output is in /tmp/mrtask_c.hadoop.20221221.181155.032889/output
Streaming final output from /tmp/mrtask_c.hadoop.20221221.181155.032889/output...
"1"     26986047
"2"     12837731
"3"     215936
"4"     61980
"5"     2
Removing temp directory /tmp/mrtask_c.hadoop.20221221.181155.032889...
```

Task D:-

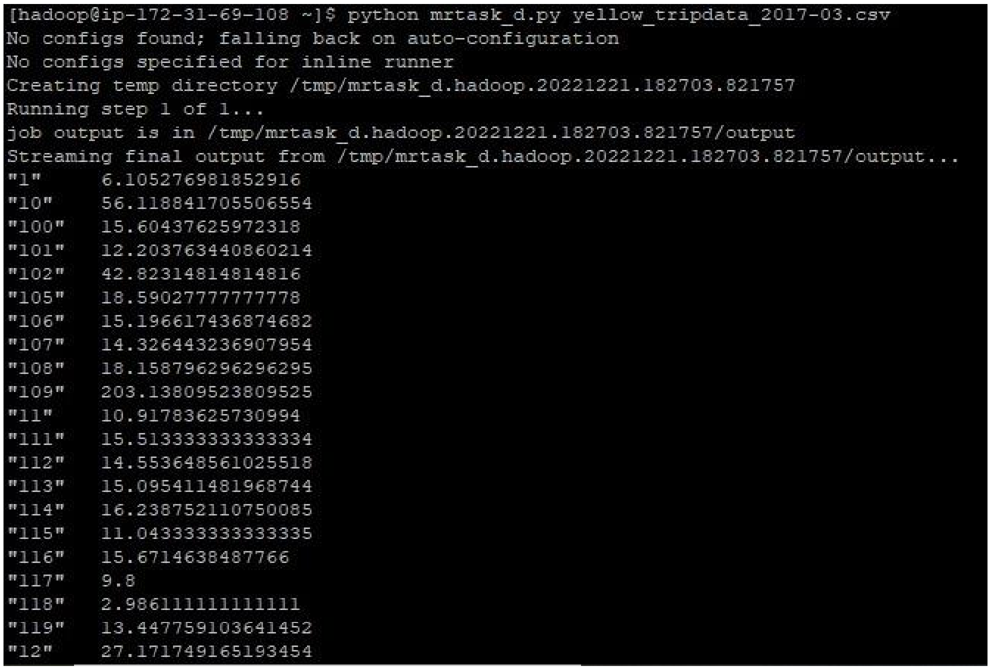
Question: What is the average trip time for different pickup locations?

Solution:

- The program consists of a single MapReduce step that has a mapper function and a reducer function.
- The mapper function processes each line in the input dataset, which represents a single taxi ride. It extracts the pickup location ID and the pickup and dropoff times for the ride. It then converts the pickup and dropoff times to datetime objects and calculates the trip duration in minutes. Finally, it emits the pickup location ID and trip duration as a tuple.
- The reducer function receives a list of trip durations for each pickup location ID as input. It converts the generator object to a list and calculates the total trip duration and number of trips for the pickup location. It then computes the average trip duration for the pickup location and emits the pickup location ID and average trip duration as output.

To run the script type: python mrtask_d.py <input-file-path> > <output-file-path>

Output:



```
[hadoop@ip-172-31-69-108 ~]$ python mrtask_d.py yellow_tripdata_2017-03.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_d.hadoop.20221221.182703.821757
Running step 1 of 1...
job output is in /tmp/mrtask_d.hadoop.20221221.182703.821757/output
Streaming final output from /tmp/mrtask_d.hadoop.20221221.182703.821757/output...
"1"     6.105276981852916
"10"    56.118841705506554
"100"   15.60437625972318
"101"   12.203763440860214
"102"   42.82314814814816
"105"   18.59027777777778
"106"   15.196617436874682
"107"   14.326443236907954
"108"   18.158796296296295
"109"   203.13809523809525
"11"    10.91783625730994
"111"   15.513333333333334
"112"   14.553648561025518
"113"   15.095411481968744
"114"   16.238752110750085
"115"   11.043333333333335
"116"   15.6714638487766
"117"   9.8
"118"   2.986111111111111
"119"   13.447759103641452
"12"    27.171749165193454
```

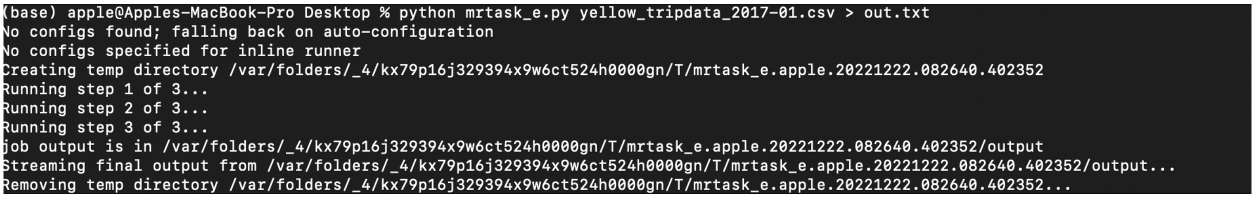
Task E:-

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

Solution:

- The job consists of three steps:
- The first step uses a mapper function (mapper_step1) to process each line in the input dataset and extract the pickup location ID, tips, and revenue values. The mapper function then emits the pickup location ID and the tips and revenue values as a tuple. A reducer function (reducer_step1) is then used to sum the tips and revenue values for each pickup location ID and emit the pickup location ID and the total tips and total revenue values.
- The second step uses a mapper function (mapper_step2) to process the output from the first step and compute the tips to revenue ratio for each pickup location. The mapper function then emits the pickup location ID and the ratio. A reducer function (reducer_step2) is then used to sum the ratios for each pickup location ID and compute the average ratio. The reducer function then emits the pickup location ID and the average ratio.
- The third step uses a reducer function (reducer_step3) to process the output from the second step and sort the pickup location IDs and average ratios by pickup location ID. The reducer function then iterates over the sorted list of ratios and emits the pickup location ID and average ratio for each pickup location.

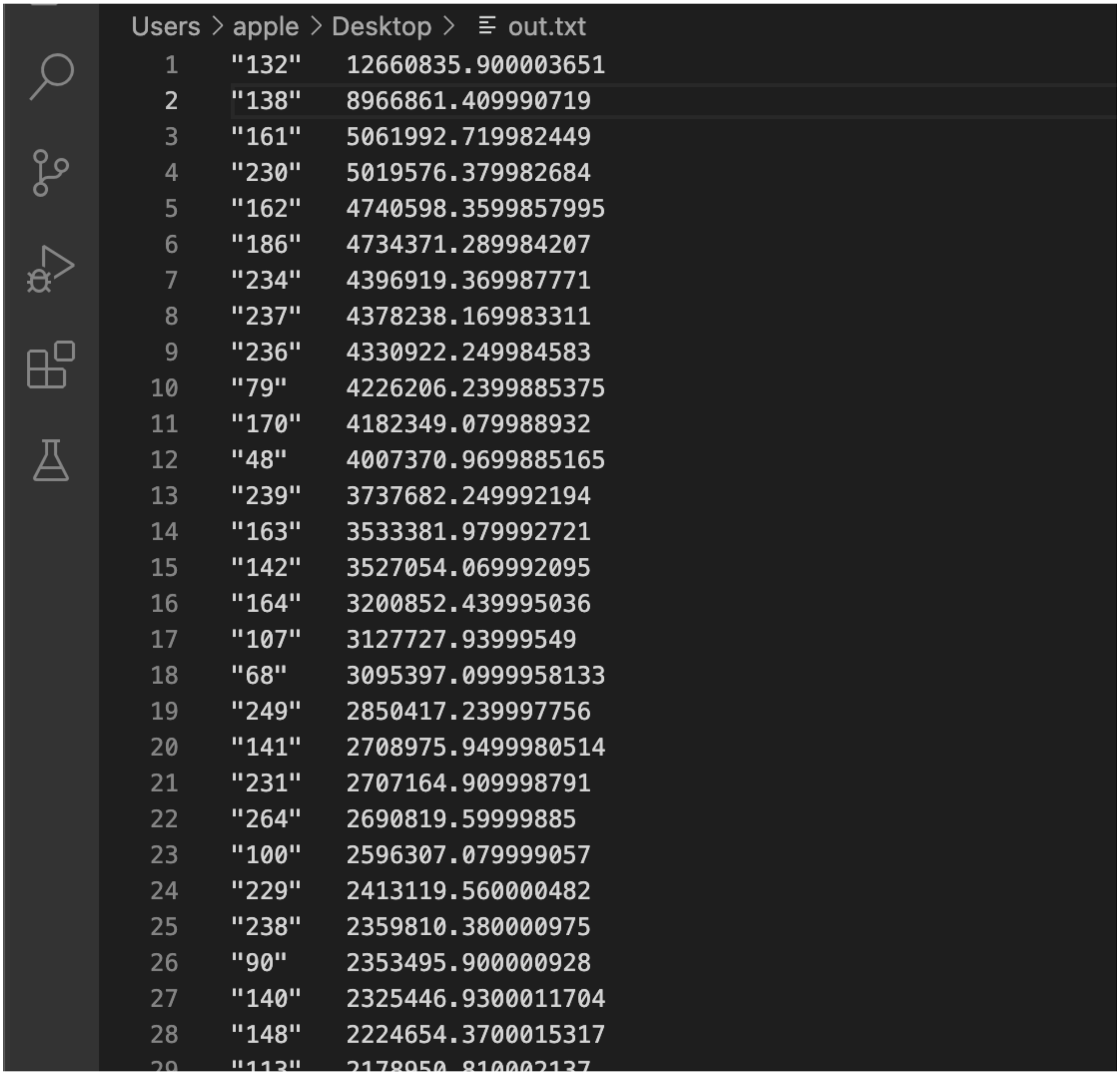
To run the script type: python mrtask_e.py <input-file-path> > <output-file-path>



```
(base) apple@Apples-MacBook-Pro Desktop % python mrtask_e.py yellow_tripdata_2017-01.csv > out.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /var/folders/_4/kx79p16j329394x9w6ct524h0000gn/T/mrtask_e.apple.20221222.082640.402352
Running step 1 of 3...
Running step 2 of 3...
Running step 3 of 3...
job output is in /var/folders/_4/kx79p16j329394x9w6ct524h0000gn/T/mrtask_e.apple.20221222.082640.402352/output
Streaming final output from /var/folders/_4/kx79p16j329394x9w6ct524h0000gn/T/mrtask_e.apple.20221222.082640.402352/output...
Removing temp directory /var/folders/_4/kx79p16j329394x9w6ct524h0000gn/T/mrtask_e.apple.20221222.082640.402352...
```

Output:

Download out.txt on local machine



```
"132"   12660835.900003651
"138"   8966861.409990719
"161"   5061992.719982449
"230"   5019576.379982684
"162"   4740598.3599857995
"186"   4734371.289984207
"234"   4396919.369987771
"237"   4378238.169983311
"236"   4330922.249984583
"79"    4226206.2399885375
"170"   4182349.079988932
"48"    4007370.9699885165
"239"   3737682.249992194
"163"   3533381.979992721
"142"   3527054.069992095
"164"   3200852.439995036
"107"   3127727.93999549
"68"    3095397.0999958133
"249"   2850417.239997756
"141"   2708975.9499980514
"231"   2707164.909998791
"264"   2690819.59999885
"100"   2596307.079999057
"229"   2413119.560000482
"238"   2359810.380000975
"90"    2353495.900000928
"140"   2325446.9300011704
"148"   2224654.3700015317
"113"   2178950.810002137
```

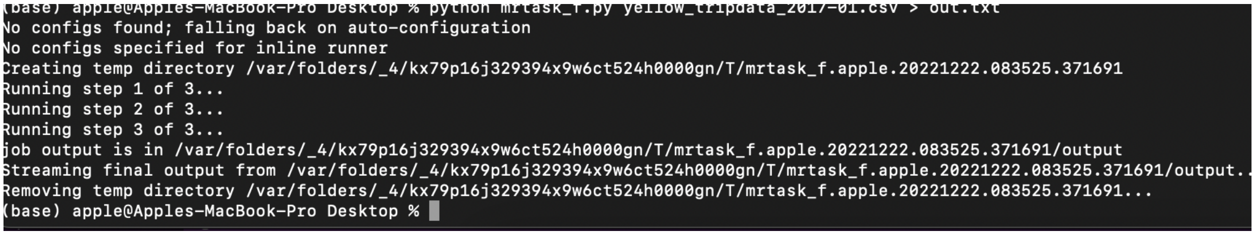
Task F:-

Question: 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).

Solution:

- The job consists of three steps:
- The mapper function in the first step processes each line of the input file, parses the pickup time and total revenue fields, and extracts the month, hour of the day, and day of the week from the pickup time. The function then emits a key that consists of a tuple containing the month, hour of the day, and day of the week, and the total revenue as the value.
- The reducer function in the first step receives a key (a tuple containing the month, hour of the day, and day of the week) and a list of values (the total revenues for that key). The function sums the values to calculate the total revenue for that key, and then divides the total revenue by the number of values to calculate the average revenue. The function then emits the key and average revenue.
- The mapper function in the second step processes the output of the first step, and emits the key and average revenue as the key and value, respectively.
- The reducer function in the second step receives a key (the month, hour of the day, and day of the week) and a list of values (the average revenues for that key). The function sums the values to calculate the total average revenue for that key, and then divides the total average revenue by the number of values to calculate the final average revenue. The function then emits the key and final average revenue.
- The mapper function in the third step processes the output of the second step, and emits the key and final average revenue as the key and value, respectively.
- The reducer function in the third step receives a key (the month, hour of the day, and day of the week) and a list of values (the final average revenues for that key). The function sums the values to calculate the total final average revenue for that key, and then divides the total final average revenue by the number of values to calculate the final average revenue for that key. The function then emits the key and final average revenue.

To run the script type: python mrtask_f.py <input-file-path> > <output-file-path>



```
(base) apple@Apples-MacBook-Pro Desktop % python mrtask_f.py yellow_tripdata_2017-01.csv > out.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /var/folders/_4/kx79p16j329394x9w6ct524h0000gn/T/mrtask_f.apple.20221222.083525.371691
Running step 1 of 3...
Running step 2 of 3...
Running step 3 of 3...
job output is in /var/folders/_4/kx79p16j329394x9w6ct524h0000gn/T/mrtask_f.apple.20221222.083525.371691/output
Streaming final output from /var/folders/_4/kx79p16j329394x9w6ct524h0000gn/T/mrtask_f.apple.20221222.083525.371691/output..
Removing temp directory /var/folders/_4/kx79p16j329394x9w6ct524h0000gn/T/mrtask_f.apple.20221222.083525.371691...
(base) apple@Apples-MacBook-Pro Desktop %
```

Submitted By:

Member 1: Nikhil Dhiman (nikhildhiman3644@gmail.com)
Member 2: Rohan kulkarni (rohan.kulkarni951@gmail.com)
Member 3: Mahesh Kumar Patra (maheshkumarpatra5@gmail.com)