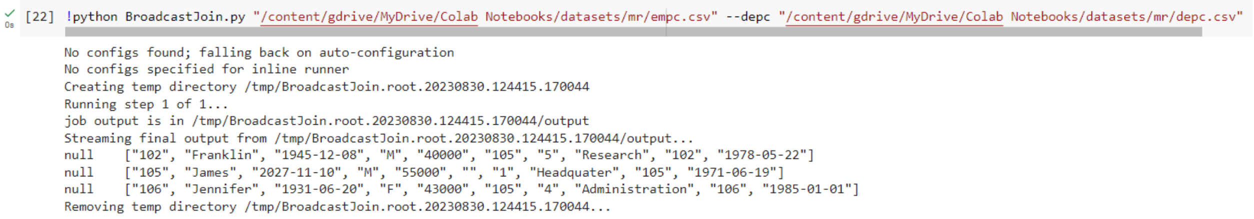
Colab Link :- https://colab.research.google.com/drive/1Jyan09QZrk00AN-lKqATamCRH5HlfbCh?usp=sharing

Q1) Broadcast 2 files depc.csv and emp.csv using algorithm given in the textbook

Code:-

```
%%file BroadcastJoin.py
from mrjob.job import MRJob
from mrjob.job import MRStep
import re

class BroadcastJoin(MRJob):
    def configure_args(self):
       super(BroadcastJoin,self).configure_args()
       self.add_file_arg('--depc',help = 'Path to depc.csv')
    def mapper_init(self):
      with open(self.options.depc,'r') as f:
        self.departement = {}
        for line in f:
          fields = line.strip().split(",")
          dno = fields[0]
          self.departement[dno] = fields[1:]

    def mapper(self, _ , line):
      record = line.split(",")
      eno = record[0]
      dno = record[6]

      if  dno in self.departement and eno == self.departement[dno][1]:
        yield None,record + self.departement[dno]

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

Output:

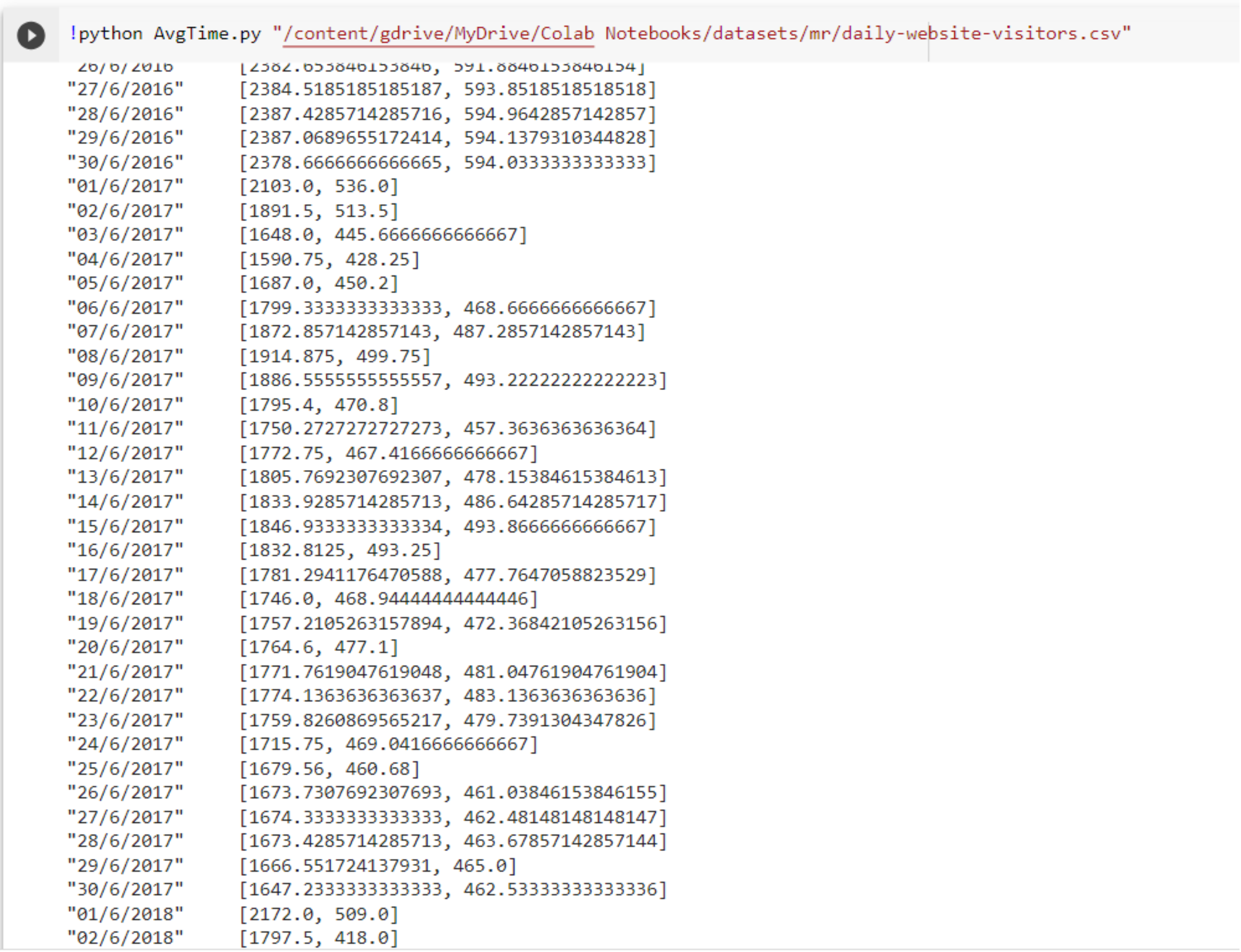


```
[22] !python BroadcastJoin.py "/content/gdrive/MyDrive/Colab Notebooks/datasets/mr/empc.csv" --depc "/content/gdrive/MyDrive/Colab Notebooks/datasets/mr/depc.csv"

No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/BroadcastJoin.root.20230830.124415.170044
Running step 1 of 1...
job output is in /tmp/BroadcastJoin.root.20230830.124415.170044/output
Streaming final output from /tmp/BroadcastJoin.root.20230830.124415.170044/output...
null    ["102", "Franklin", "1945-12-08", "M", "40000", "105", "5", "Research", "102", "1978-05-22"]
null    ["105", "James", "2027-11-10", "M", "55000", "", "1", "Headquater", "105", "1971-06-19"]
null    ["106", "Jennifer", "1931-06-20", "F", "43000", "105", "4", "Administration", "106", "1985-01-01"]
Removing temp directory /tmp/BroadcastJoin.root.20230830.124415.170044...
```

Q2)Let you yourself figure out a way a map-reduce based solution to compute moving average of time series data. There is book titled "Data Algorithms" [6]. A copy of the book is placed in shared dataset folder itself. Chapter 6 of this book discusses the computation of Moving average using map-reduce. Refer related section for this purpose. Choose "Example 2: Time Series Data (URL Visits)" as data space.Compute monthly moving average of website visitors (first time and repeat).

```
%%file AvgTime.py
from mrjob.job import MRJob
import re
import shlex

class AvgTime(MRJob):

    def mapper(self, _, line):
        if "Row" not in line:
            record = shlex.shlex(line, posix=True)
            record.whitespace += ','
            record.whitespace_split = True
            record = list(record)

            # Extract relevant fields
            Row_id = record[0]
            Day = record[1]
            Day_Of_Week = record[2]
            Date = re.split('/', record[3])
            Page_Loads = int(record[4].replace(',', ''))
            Unique_Visits = int(record[5].replace(',', ''))
            First_Time_Visits = int(record[6].replace(',', ''))
            Returning_Visits = int(record[7].replace(',', ''))


            if int(Date[1]) < 10:
                date = "0" + Date[1] + "/" + Date[0] + "/" + Date[2]
            else:
                date = Date[1] + "/" + Date[0] + "/" + Date[2]

            # month-year as key and relevant data as value
            yield Date[0] + " " + Date[2], (date, First_Time_Visits,
Returning_Visits)

    def reducer(self, key, values):
        values_list = list(values)
        values_list.sort()

        avg_first_time_sum = 0.0
        avg_returning_sum = 0.0
        count = 0

        for date, first_time, returning in values_list:
            avg_first_time_sum += first_time
            avg_returning_sum += returning
            count += 1

        # Calculate the averages
            avg_first_time = avg_first_time_sum / count
            avg_returning = avg_returning_sum / count

        # Yield the date and calculated averages
            yield date, [avg_first_time, avg_returning]
if __name__ == '__main__':
    AvgTime.run()
```

Output :-



```
!python AvgTime.py "/content/gdrive/MyDrive/Colab Notebooks/datasets/mr/daily-website-visitors.csv"
 "26/6/2016"     [2382.653846153846, 591.8846153846154]
 "27/6/2016"     [2384.5185185185187, 593.8518518518518]
 "28/6/2016"     [2387.4285714285716, 594.9642857142857]
 "29/6/2016"     [2387.0689655172414, 594.1379310344828]
 "30/6/2016"     [2378.6666666666665, 594.0333333333333]
 "01/6/2017"     [2103.0, 536.0]
 "02/6/2017"     [1891.5, 513.5]
 "03/6/2017"     [1648.0, 445.6666666666667]
 "04/6/2017"     [1590.75, 428.25]
 "05/6/2017"     [1687.0, 450.2]
 "06/6/2017"     [1799.3333333333333, 468.6666666666667]
 "07/6/2017"     [1872.857142857143, 487.2857142857143]
 "08/6/2017"     [1914.875, 499.75]
 "09/6/2017"     [1886.5555555555557, 493.2222222222223]
 "10/6/2017"     [1795.4, 470.8]
 "11/6/2017"     [1750.2727272727273, 457.3636363636364]
 "12/6/2017"     [1772.75, 467.4166666666667]
 "13/6/2017"     [1805.7692307692307, 478.1538461538461З]
 "14/6/2017"     [1833.9285714285713, 486.64285714285717]
 "15/6/2017"     [1846.9333333333334, 493.8666666666667]
 "16/6/2017"     [1832.8125, 493.25]
 "17/6/2017"     [1781.2941176470588, 477.7647058823529]
 "18/6/2017"     [1746.0, 468.9444444444446]
 "19/6/2017"     [1757.2105263157894, 472.3684210526315б]
 "20/6/2017"     [1764.6, 477.1]
 "21/6/2017"     [1771.7619047619048, 481.04761904761904]
 "22/6/2017"     [1774.1363636363637, 483.1363636363636]
 "23/6/2017"     [1759.8260869565217, 479.7391304347826]
 "24/6/2017"     [1715.75, 469.0416666666667]
 "25/6/2017"     [1679.56, 460.68]
 "26/6/2017"     [1673.7307692307693, 461.03846153846155]
 "27/6/2017"     [1674.3333333333333, 462.4814814814814 7]
 "28/6/2017"     [1673.4285714285713, 463.6785714285714 4]
 "29/6/2017"     [1666.551724137931, 465.0]
 "30/6/2017"     [1647.2333333333333, 462.5333333333333 6]
 "01/6/2018"     [2172.0, 509.0]
 "02/6/2018"     [1797.5, 418.0]
```