YCBS 257 Data at Scale
Team Project

Flight Distance with Map Reduce

Team 3

Nicole Hong
Ahmed Ibrahim
Pranavkumar Pathak

November 17, 2022

Introduction

Capable of Big Data processing

Selecting appropriate fields from a given file

Identifying the last position of flight & computing a distance

Objective

Map Reduce

Extract Flight Data and Summarize Flight Distance between Beijing and Data points, using Map Reduce

JSON to CSV

Reformat JSON Objects & Strings to CSV List

Analysis

Analyze Flight Data in CSV

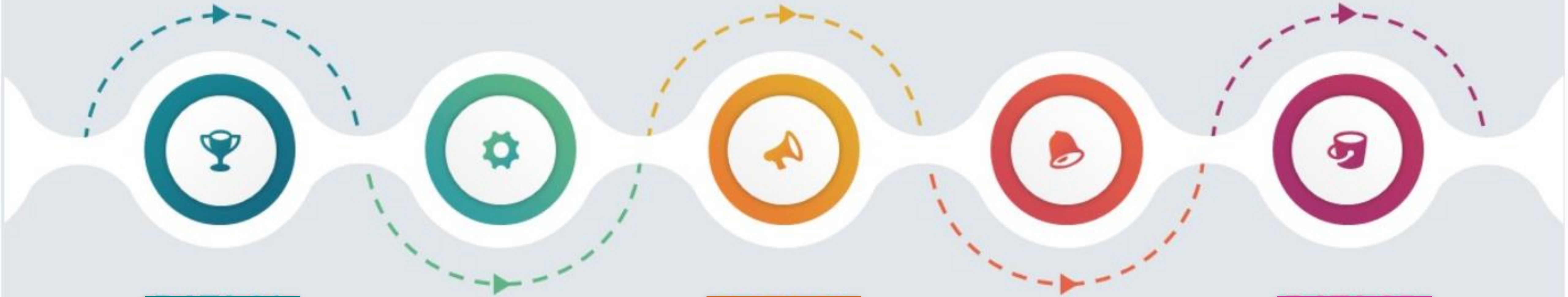
Dataset Description

Data is coming from the following API :

Flight Aware Service Firehose: https://flightaware.com/commercial/firehose/

Important Fields	
pitr	The POSIX epoch time the message was sent by the airplane (seconds since 1 January 1970)
type	The type of message
id	A unique id of that flight.
ident	The flight number (not unique)
dest	The destination airport code
orig	The departure/origin airport code
lat	Latitude
lon	Longitude
Clock	Timestamp

Workflow



Step 1

Verify and Review Input Dataset

Step 2

Build Mapper and Generate Mapper Output File

Step 3

Build Reducer and Generate Reducer Output File

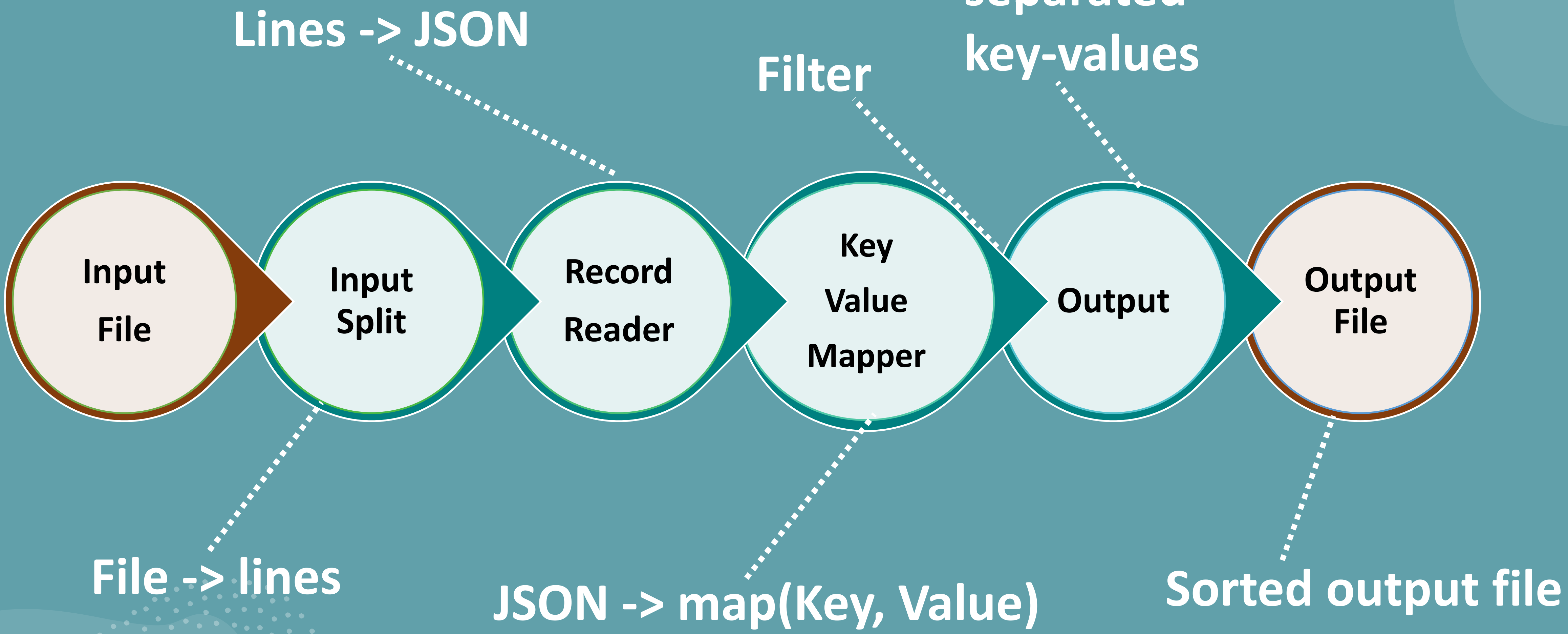
Step 4

Create a CSV file

Step 5

Conduct Data Analysis

Mapper - Workflow



Input File

Input Split

Record Reader

Key Value Mapper

Output

Output File

File -> lines

Lines -> JSON

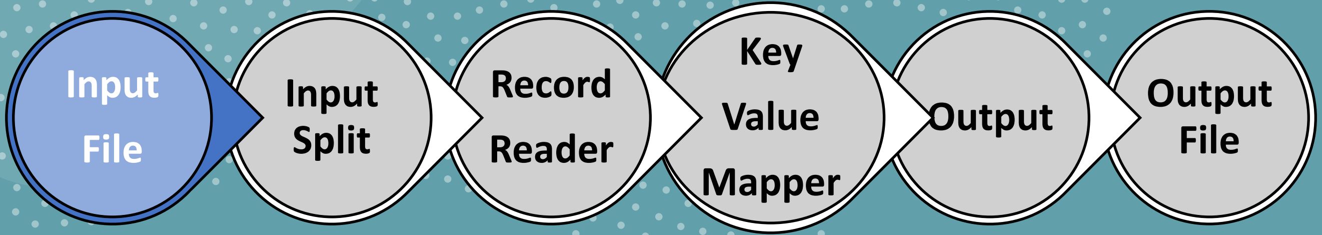
JSON -> map(Key, Value)

Filter

Tab separated key-values

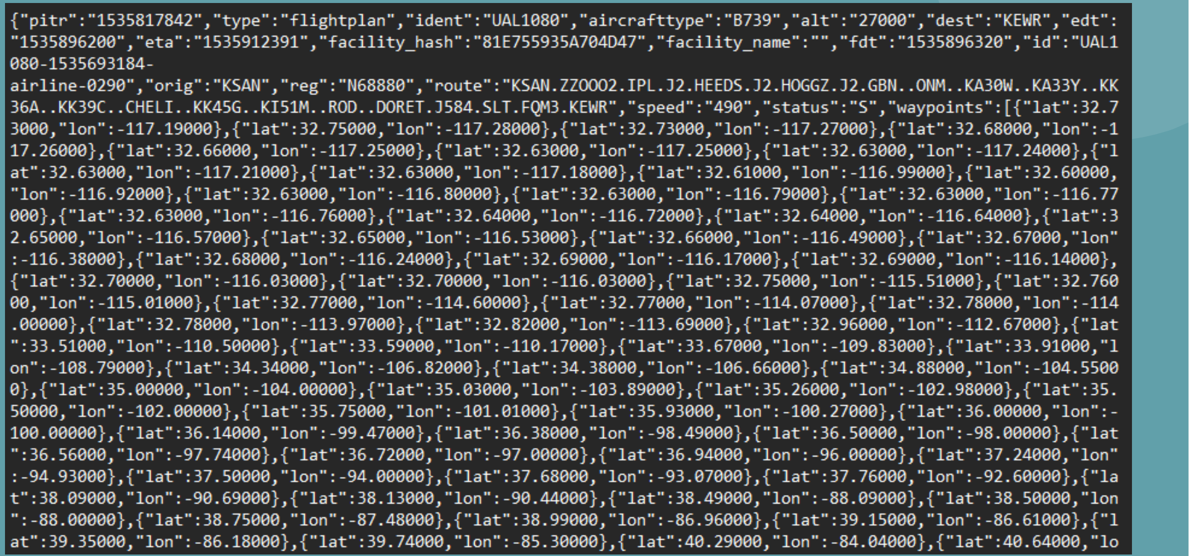
Sorted output file

Input File



Verification

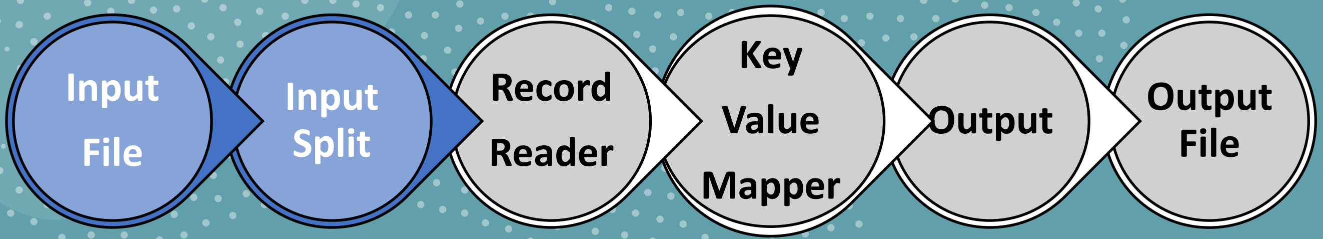
- Is the data structured in a table-like text file? Yes.
- Does every record span only 1 line? Yes.



```
{"pitr":"1535817842","type":"flightplan","ident":"UAL1080","aircrafttype":"B739","alt":"27000","dest":"KEWR","edt":
"1535896200","eta":"1535912391","facility_hash":"81E755935A704D47","facility_name":"","fdt":"1535896320","id":"UAL1
080-1535693184-
airline-0290","orig":"KSAN","reg":"N68880","route":"KSAN.ZZOOO2.IPL.J2.HEEDS.J2.HOGGZ.J2.GBN..ONM..KA30W..KA33Y..KK
36A..KK39C..CHELI..KK45G..KI51M..ROD..DORET.J584.SLT.FQM3.KEWR","speed":"490","status":"S","waypoints":[{"lat":32.7
3000,"lon":-117.19000},{"lat":32.75000,"lon":-117.28000},{"lat":32.73000,"lon":-117.27000},{"lat":32.68000,"lon":-1
17.26000},{"lat":32.66000,"lon":-117.25000},{"lat":32.63000,"lon":-117.25000},{"lat":32.63000,"lon":-117.24000},{"l
at":32.63000,"lon":-117.21000},{"lat":32.63000,"lon":-117.18000},{"lat":32.61000,"lon":-116.99000},{"lat":32.60000,
"lon":-116.92000},{"lat":32.63000,"lon":-116.80000},{"lat":32.63000,"lon":-116.79000},{"lat":32.63000,"lon":-116.77
000},{"lat":32.63000,"lon":-116.76000},{"lat":32.64000,"lon":-116.72000},{"lat":32.64000,"lon":-116.64000},{"lat":3
2.65000,"lon":-116.57000},{"lat":32.65000,"lon":-116.53000},{"lat":32.66000,"lon":-116.49000},{"lat":32.67000,"lon"
:-116.38000},{"lat":32.68000,"lon":-116.24000},{"lat":32.69000,"lon":-116.17000},{"lat":32.69000,"lon":-116.14000},
{"lat":32.70000,"lon":-116.03000},{"lat":32.70000,"lon":-116.03000},{"lat":32.75000,"lon":-115.51000},{"lat":32.760
00,"lon":-115.01000},{"lat":32.77000,"lon":-114.60000},{"lat":32.77000,"lon":-114.07000},{"lat":32.78000,"lon":-114
.00000},{"lat":32.78000,"lon":-113.97000},{"lat":32.82000,"lon":-113.69000},{"lat":32.96000,"lon":-112.67000},{"lat
":33.51000,"lon":-110.50000},{"lat":33.59000,"lon":-110.17000},{"lat":33.67000,"lon":-109.83000},{"lat":33.91000,"l
on":-108.79000},{"lat":34.34000,"lon":-106.82000},{"lat":34.38000,"lon":-106.66000},{"lat":34.88000,"lon":-104.5500
0},{"lat":35.00000,"lon":-104.00000},{"lat":35.03000,"lon":-103.89000},{"lat":35.26000,"lon":-102.98000},{"lat":35.
50000,"lon":-102.00000},{"lat":35.75000,"lon":-101.01000},{"lat":35.93000,"lon":-100.27000},{"lat":36.00000,"lon":-
100.00000},{"lat":36.14000,"lon":-99.47000},{"lat":36.38000,"lon":-98.49000},{"lat":36.50000,"lon":-98.00000},{"lat
":36.56000,"lon":-97.74000},{"lat":36.72000,"lon":-97.00000},{"lat":36.94000,"lon":-96.00000},{"lat":37.24000,"lon"
:-94.93000},{"lat":37.50000,"lon":-94.00000},{"lat":37.68000,"lon":-93.07000},{"lat":37.76000,"lon":-92.60000},{"la
t":38.09000,"lon":-90.69000},{"lat":38.13000,"lon":-90.44000},{"lat":38.49000,"lon":-88.09000},{"lat":38.50000,"lon
":-88.00000},{"lat":38.75000,"lon":-87.48000},{"lat":38.99000,"lon":-86.96000},{"lat":39.15000,"lon":-86.61000},{"l
at":39.35000,"lon":-86.18000},{"lat":39.74000,"lon":-85.30000},{"lat":40.29000,"lon":-84.04000},{"lat":40.64000,"lo
```

- File name: ‘groupassignmentdata.txt’
- File size: 12540 KB
- # rows: 19404

Input Split

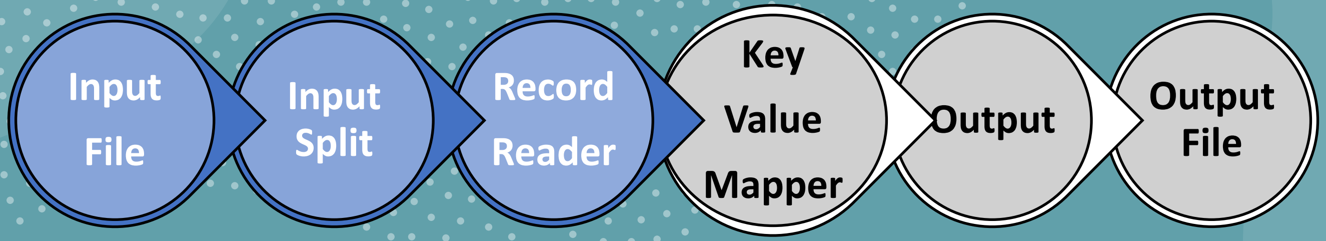


Every line in the text file is sent to the record reader.

```
# Define File I/O for mapper.
finput=open('groupassignmentdata.txt','r')
foutput=open('groupassignmentdata_mapped.txt','w')
```

```
# The 'input splitter' reads every line within the
input file object.
for line in finput:
    # Each line is sent to the 'record reader'.
    recordreader(line)
# Once each line has been read and processed, the
file I/O will close.
finput.close()
foutput.close()
```

Record Reader



Input File

Input Split

Record Reader

Key Value Mapper

Output

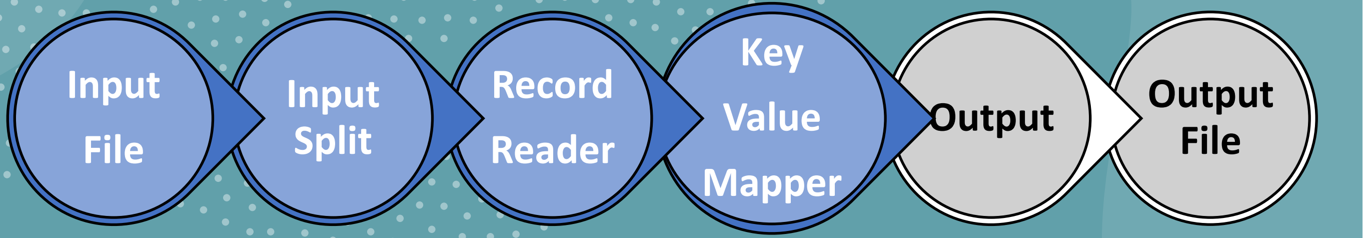
Output File

Every line is parsed and transformed into a JSON object.

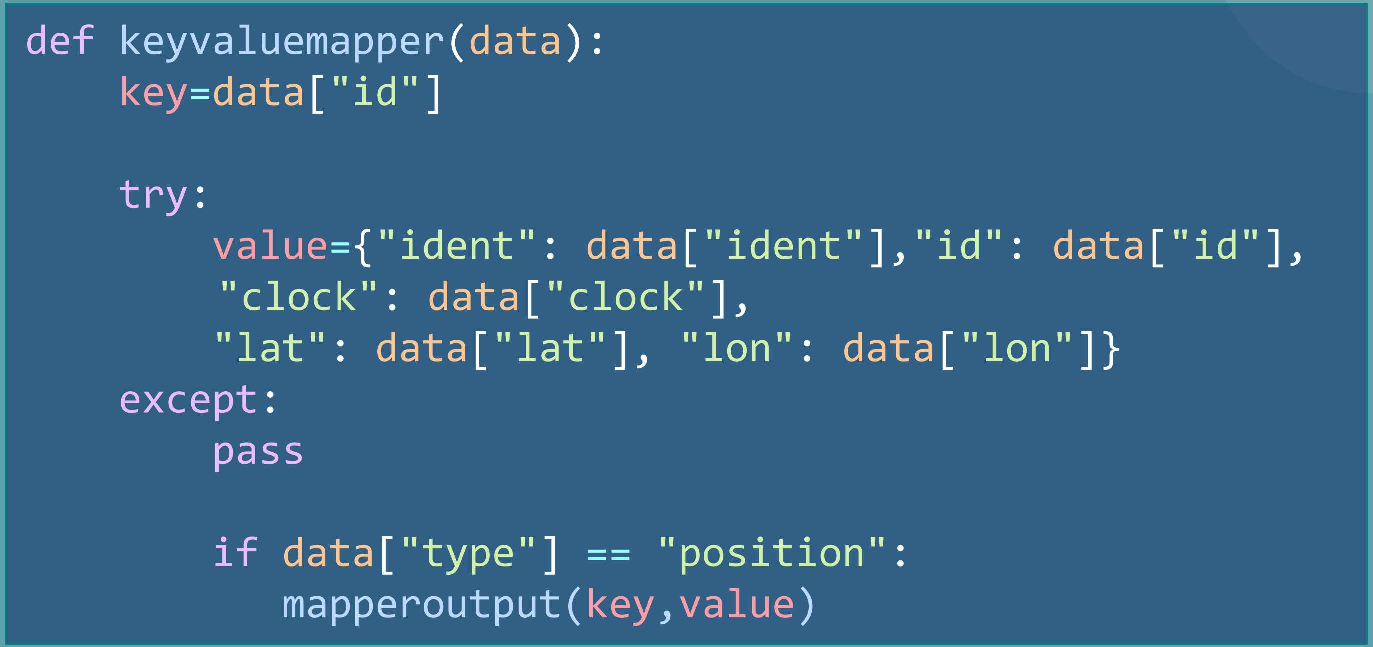
```
# JSON library needed for 'record reader'.
import json
```

```
def recordreader(line):
        # Necessary to convert from  text to JSON.
        data=json.loads(line)
        # The data can be passed to  the key-value mapper.
        keyvaluemapper(data)
```

Key Value Mapper

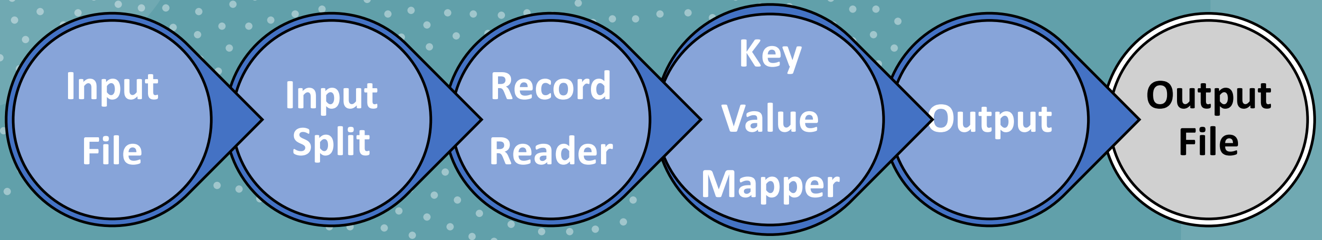


- Key: flight unique id,
- Values: ids, clock, latitude and longitude,
- Filter: Only position records.

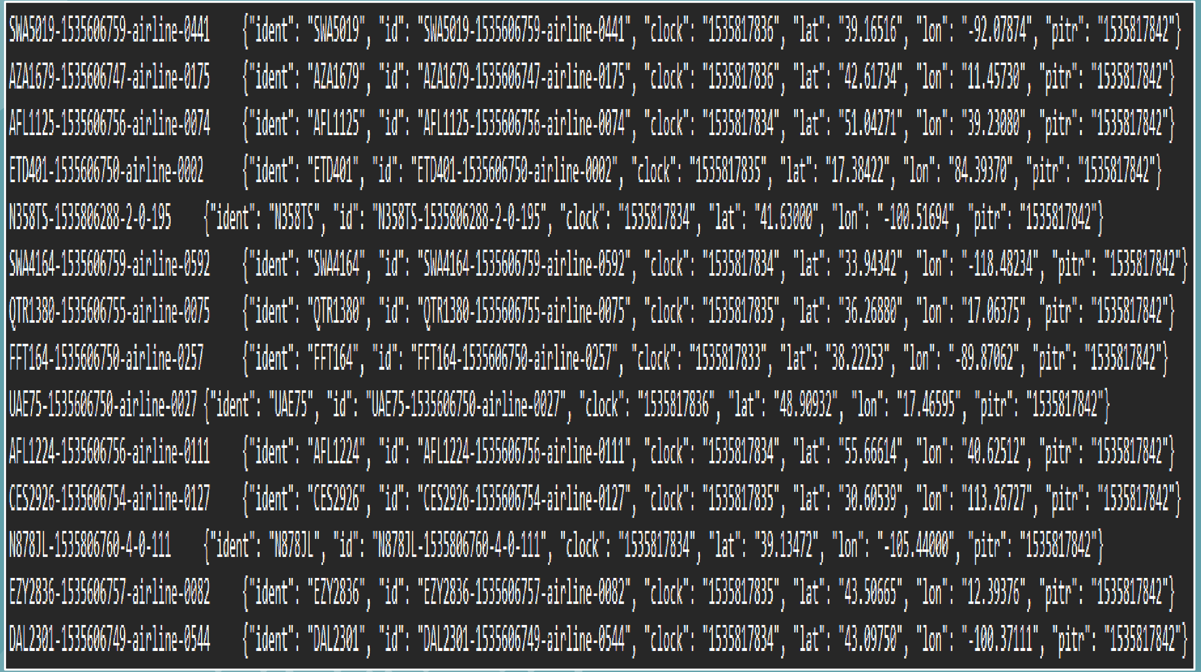


```
def keyvaluemapper(data):
    key=data["id"]

    try:
        value={"ident": data["ident"],"id": data["id"],
        "clock": data["clock"],
        "lat": data["lat"], "lon": data["lon"]}
    except:
        pass

        if data["type"] == "position":
           mapperoutput(key,value)
```

Output



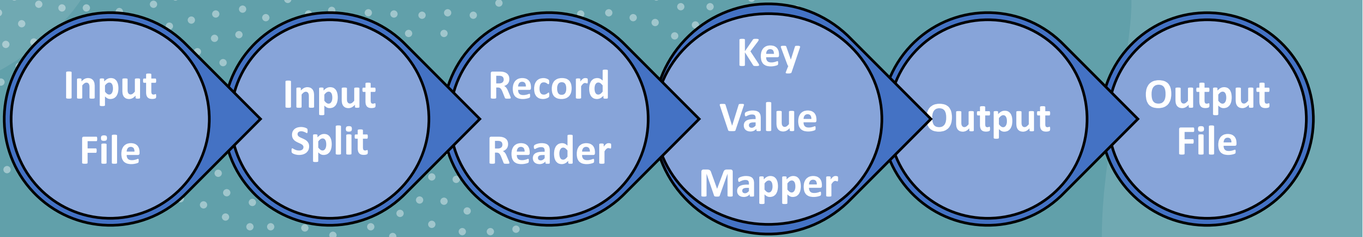
The output block writes to the output file two tab-separated columns being the keys and values.



```
SWA5019-1535606759-airline-0441	{"ident": "SWA5019", "id": "SWA5019-1535606759-airline-0441", "clock": "1535817836", "lat": "39.16516", "lon": "-92.07874", "pitr": "1535817842"}
AZA1679-1535606747-airline-0175	{"ident": "AZA1679", "id": "AZA1679-1535606747-airline-0175", "clock": "1535817836", "lat": "42.61734", "lon": "11.45730", "pitr": "1535817842"}
AFL1125-1535606756-airline-0074	{"ident": "AFL1125", "id": "AFL1125-1535606756-airline-0074", "clock": "1535817834", "lat": "51.04271", "lon": "39.23080", "pitr": "1535817842"}
ETD401-1535606750-airline-0002	{"ident": "ETD401", "id": "ETD401-1535606750-airline-0002", "clock": "1535817835", "lat": "17.38422", "lon": "84.39370", "pitr": "1535817842"}
N358TS-1535806288-2-0-195	{"ident": "N358TS", "id": "N358TS-1535806288-2-0-195", "clock": "1535817834", "lat": "41.63000", "lon": "-100.51694", "pitr": "1535817842"}
SWA4164-1535606759-airline-0592	{"ident": "SWA4164", "id": "SWA4164-1535606759-airline-0592", "clock": "1535817834", "lat": "33.94342", "lon": "-118.48234", "pitr": "1535817842"}
QTR1380-1535606755-airline-0075	{"ident": "QTR1380", "id": "QTR1380-1535606755-airline-0075", "clock": "1535817835", "lat": "36.26880", "lon": "17.06375", "pitr": "1535817842"}
FFT164-1535606750-airline-0257	{"ident": "FFT164", "id": "FFT164-1535606750-airline-0257", "clock": "1535817833", "lat": "38.22253", "lon": "-89.87062", "pitr": "1535817842"}
UAE75-1535606750-airline-0027	{"ident": "UAE75", "id": "UAE75-1535606750-airline-0027", "clock": "1535817836", "lat": "48.90932", "lon": "17.46595", "pitr": "1535817842"}
AFL1224-1535606756-airline-0111	{"ident": "AFL1224", "id": "AFL1224-1535606756-airline-0111", "clock": "1535817834", "lat": "55.66614", "lon": "40.62512", "pitr": "1535817842"}
CES2926-1535606754-airline-0127	{"ident": "CES2926", "id": "CES2926-1535606754-airline-0127", "clock": "1535817835", "lat": "30.60539", "lon": "113.26727", "pitr": "1535817842"}
N878JL-1535806760-4-0-111	{"ident": "N878JL", "id": "N878JL-1535806760-4-0-111", "clock": "1535817834", "lat": "39.13472", "lon": "-105.44000", "pitr": "1535817842"}
EZY2836-1535606757-airline-0082	{"ident": "EZY2836", "id": "EZY2836-1535606757-airline-0082", "clock": "1535817835", "lat": "43.50665", "lon": "12.39376", "pitr": "1535817842"}
DAL2301-1535606749-airline-0544	{"ident": "DAL2301", "id": "DAL2301-1535606749-airline-0544", "clock": "1535817834", "lat": "43.09750", "lon": "-100.37111", "pitr": "1535817842"}
```

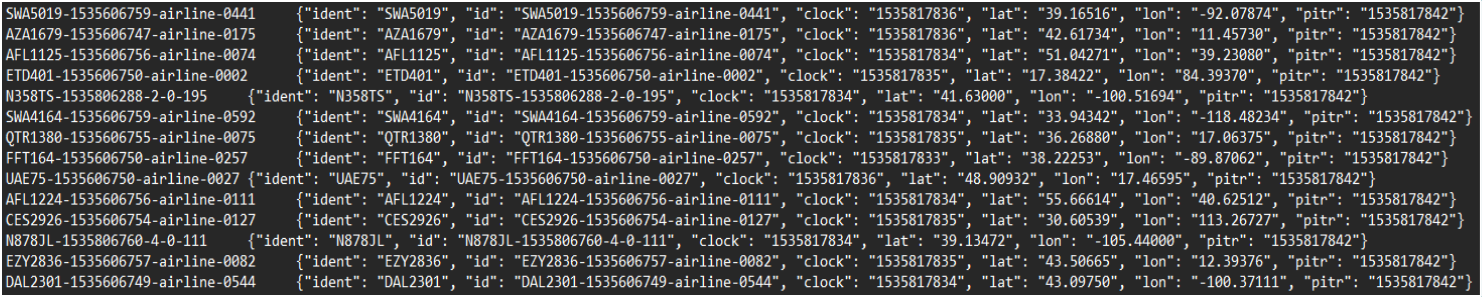
```python
def mapperoutput(key,value):
    dataline = key+"\t"+json.dumps(value)+"\n"
    print(dataline)
    foutput.write(dataline)
```

Output & Sorted File

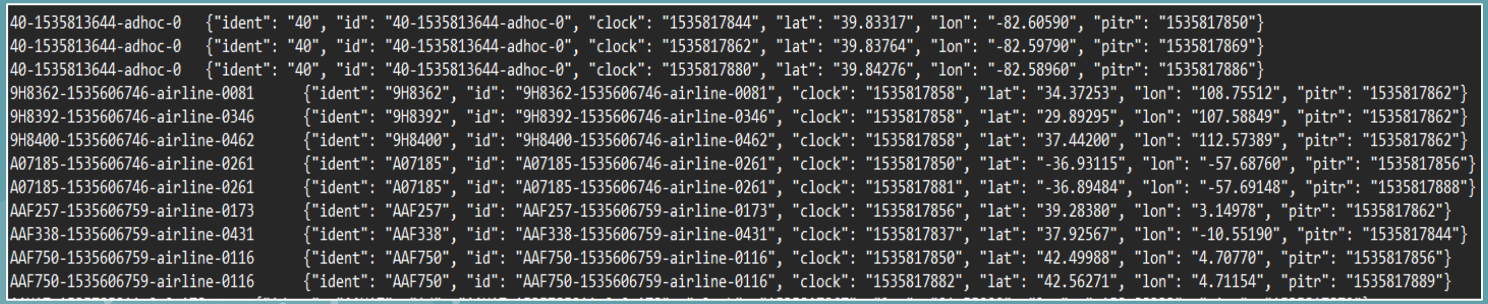


A sort command is used to sort the records in the original output in alphabetic order.

```
! sort < groupassignmentdata_mapped.txt > groupassignmentdata_mapped_sorted.txt
```

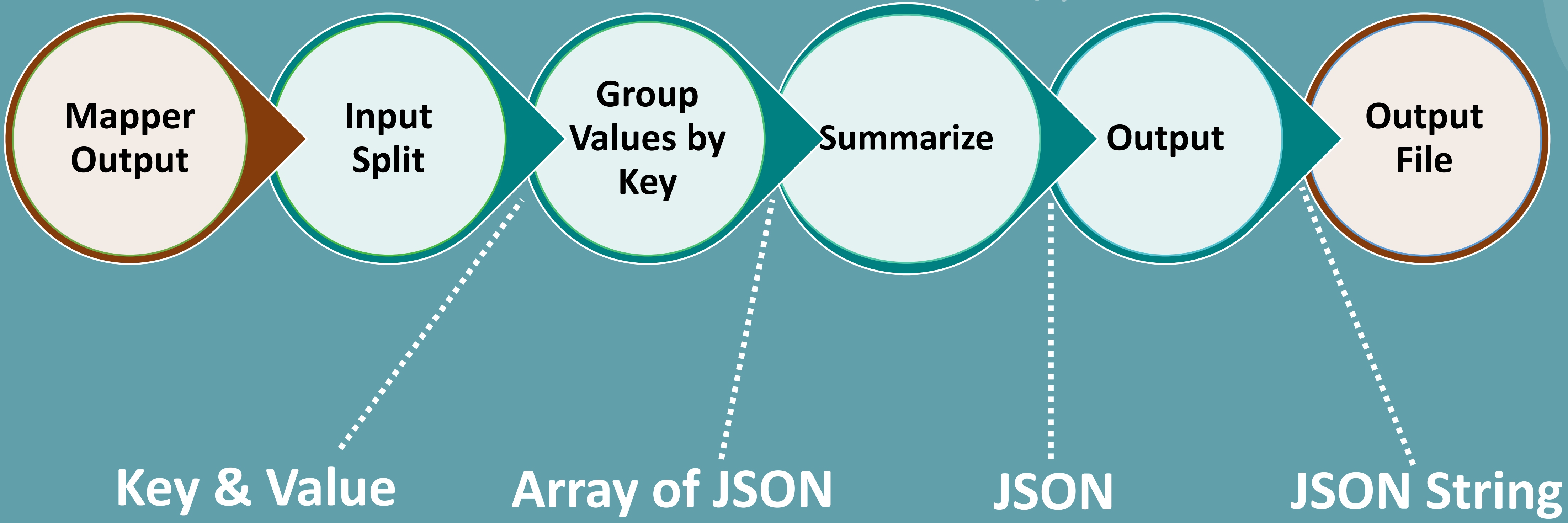


```
SWA5019-1535606759-airline-0441     {"ident": "SWA5019", "id": "SWA5019-1535606759-airline-0441", "clock": "1535817836", "lat": "39.16516", "lon": "-92.07874", "pitr": "1535817842"}
AZA1679-1535606747-airline-0175     {"ident": "AZA1679", "id": "AZA1679-1535606747-airline-0175", "clock": "1535817836", "lat": "42.61734", "lon": "11.45730", "pitr": "1535817842"}
AFL1125-1535606756-airline-0074     {"ident": "AFL1125", "id": "AFL1125-1535606756-airline-0074", "clock": "1535817834", "lat": "51.04271", "lon": "39.23080", "pitr": "1535817842"}
ETD401-1535606750-airline-0002      {"ident": "ETD401", "id": "ETD401-1535606750-airline-0002", "clock": "1535817835", "lat": "17.38422", "lon": "84.39370", "pitr": "1535817842"}
N358TS-1535806288-2-0-195     {"ident": "N358TS", "id": "N358TS-1535806288-2-0-195", "clock": "1535817834", "lat": "41.63000", "lon": "-100.51694", "pitr": "1535817842"}
SWA4164-1535606759-airline-0592     {"ident": "SWA4164", "id": "SWA4164-1535606759-airline-0592", "clock": "1535817834", "lat": "33.94342", "lon": "-118.48234", "pitr": "1535817842"}
QTR1380-1535606755-airline-0075     {"ident": "QTR1380", "id": "QTR1380-1535606755-airline-0075", "clock": "1535817835", "lat": "36.26880", "lon": "17.06375", "pitr": "1535817842"}
FFT164-1535606750-airline-0257      {"ident": "FFT164", "id": "FFT164-1535606750-airline-0257", "clock": "1535817833", "lat": "38.22253", "lon": "-89.87062", "pitr": "1535817842"}
UAE75-1535606750-airline-0027 {"ident": "UAE75", "id": "UAE75-1535606750-airline-0027", "clock": "1535817836", "lat": "48.90932", "lon": "17.46595", "pitr": "1535817842"}
AFL1224-1535606756-airline-0111     {"ident": "AFL1224", "id": "AFL1224-1535606756-airline-0111", "clock": "1535817834", "lat": "55.66614", "lon": "40.62512", "pitr": "1535817842"}
CES2926-1535606754-airline-0127     {"ident": "CES2926", "id": "CES2926-1535606754-airline-0127", "clock": "1535817835", "lat": "30.60539", "lon": "113.26727", "pitr": "1535817842"}
N878JL-1535806760-4-0-111     {"ident": "N878JL", "id": "N878JL-1535806760-4-0-111", "clock": "1535817834", "lat": "39.13472", "lon": "-105.44000", "pitr": "1535817842"}
EZY2836-1535606757-airline-0082     {"ident": "EZY2836", "id": "EZY2836-1535606757-airline-0082", "clock": "1535817835", "lat": "43.50665", "lon": "12.39376", "pitr": "1535817842"}
DAL2301-1535606749-airline-0544     {"ident": "DAL2301", "id": "DAL2301-1535606749-airline-0544", "clock": "1535817834", "lat": "43.09750", "lon": "-100.37111", "pitr": "1535817842"}
```

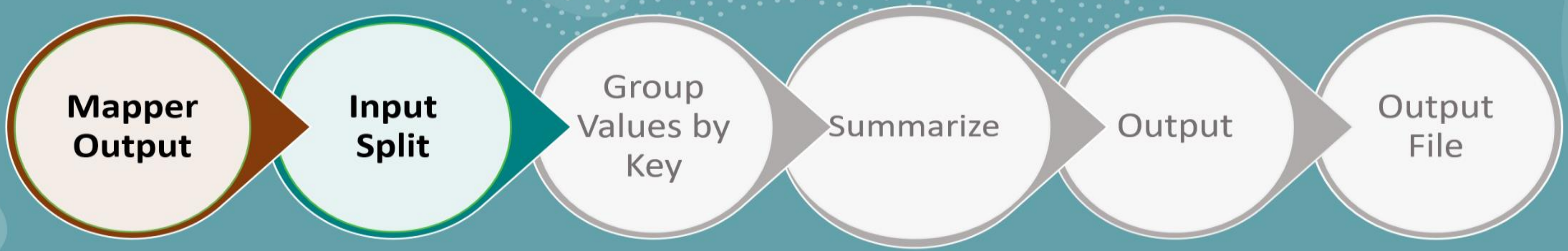


```
40-1535813644-adhoc-0   {"ident": "40", "id": "40-1535813644-adhoc-0", "clock": "1535817844", "lat": "39.83317", "lon": "-82.60590", "pitr": "1535817850"}
40-1535813644-adhoc-0   {"ident": "40", "id": "40-1535813644-adhoc-0", "clock": "1535817862", "lat": "39.83764", "lon": "-82.59790", "pitr": "1535817869"}
40-1535813644-adhoc-0   {"ident": "40", "id": "40-1535813644-adhoc-0", "clock": "1535817880", "lat": "39.84276", "lon": "-82.58960", "pitr": "1535817886"}
9H8362-1535606746-airline-0081      {"ident": "9H8362", "id": "9H8362-1535606746-airline-0081", "clock": "1535817858", "lat": "34.37253", "lon": "108.75512", "pitr": "1535817862"}
9H8392-1535606746-airline-0346      {"ident": "9H8392", "id": "9H8392-1535606746-airline-0346", "clock": "1535817858", "lat": "29.89295", "lon": "107.58849", "pitr": "1535817862"}
9H8400-1535606746-airline-0462      {"ident": "9H8400", "id": "9H8400-1535606746-airline-0462", "clock": "1535817858", "lat": "37.44200", "lon": "112.57389", "pitr": "1535817862"}
A07185-1535606746-airline-0261      {"ident": "A07185", "id": "A07185-1535606746-airline-0261", "clock": "1535817850", "lat": "-36.93115", "lon": "-57.68760", "pitr": "1535817856"}
A07185-1535606746-airline-0261      {"ident": "A07185", "id": "A07185-1535606746-airline-0261", "clock": "1535817881", "lat": "-36.89484", "lon": "-57.69148", "pitr": "1535817888"}
AAF257-1535606759-airline-0173      {"ident": "AAF257", "id": "AAF257-1535606759-airline-0173", "clock": "1535817856", "lat": "39.28380", "lon": "3.14978", "pitr": "1535817862"}
AAF338-1535606759-airline-0431      {"ident": "AAF338", "id": "AAF338-1535606759-airline-0431", "clock": "1535817837", "lat": "37.92567", "lon": "-10.55190", "pitr": "1535817844"}
AAF750-1535606759-airline-0116      {"ident": "AAF750", "id": "AAF750-1535606759-airline-0116", "clock": "1535817850", "lat": "42.49988", "lon": "4.70770", "pitr": "1535817856"}
AAF750-1535606759-airline-0116      {"ident": "AAF750", "id": "AAF750-1535606759-airline-0116", "clock": "1535817882", "lat": "42.56271", "lon": "4.71154", "pitr": "1535817889"}
```

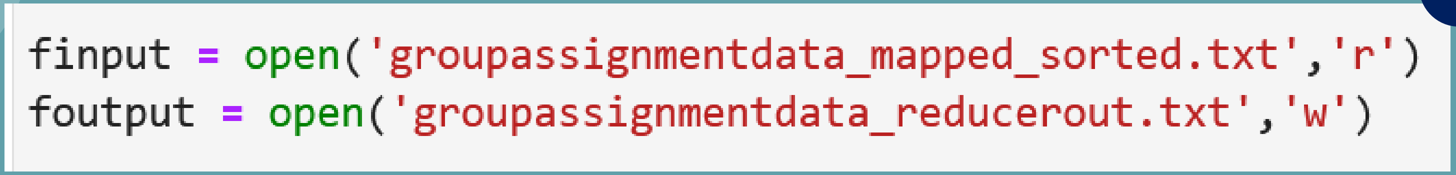
Reducer: General Process



Reducer: Mapper Output & Input Split



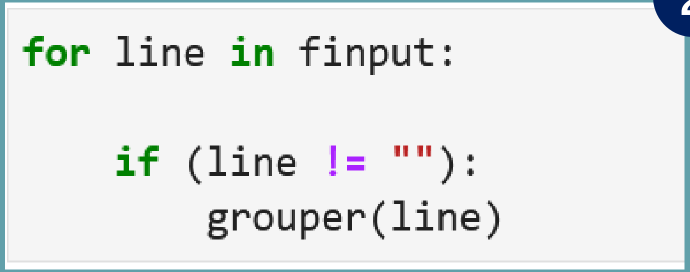
1



```
finput = open('groupassignmentdata_mapped_sorted.txt','r')
foutput = open('groupassignmentdata_reducerout.txt','w')
```

- Set up:
  - finput = Input File from the Sorted Mapper Output File
  - foutput = Output File

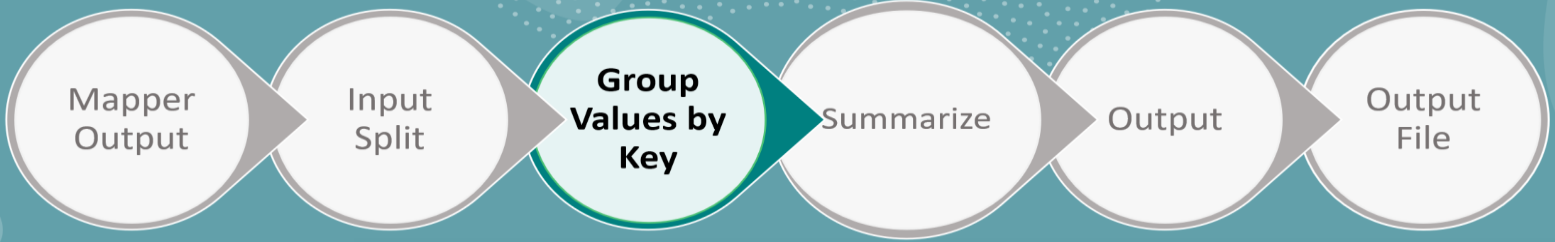
2



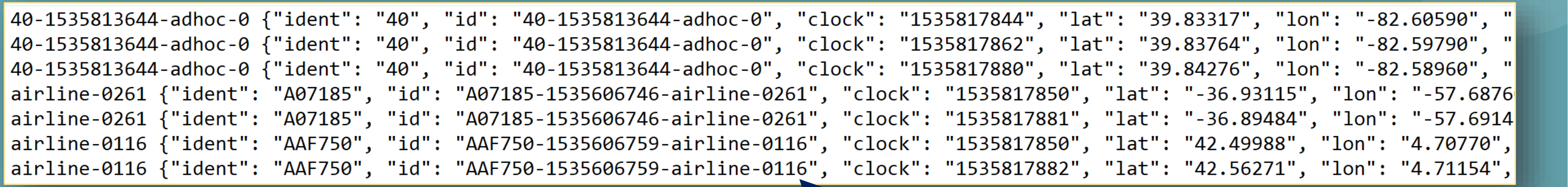
```
for line in finput:

    if (line != ""):
        grouper(line)
```

- Read the input line by line
- Each line sent to the grouper function

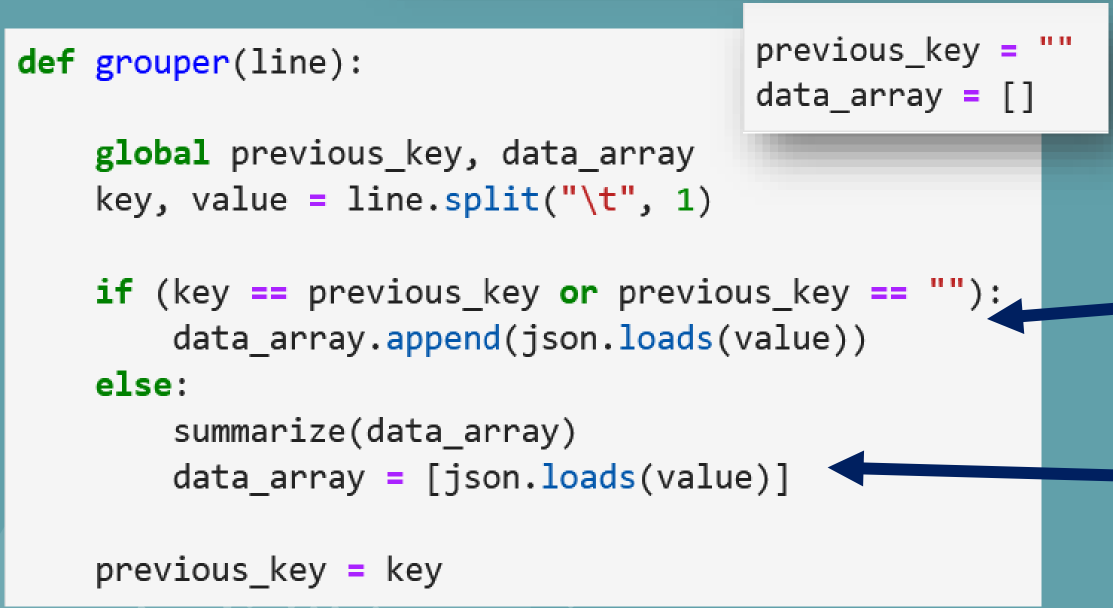
Reducer: Group Values by Key



Sorted Mapper Output File


```
40-1535813644-adhoc-0 {"ident": "40", "id": "40-1535813644-adhoc-0", "clock": "1535817844", "lat": "39.83317", "lon": "-82.60590", "
40-1535813644-adhoc-0 {"ident": "40", "id": "40-1535813644-adhoc-0", "clock": "1535817862", "lat": "39.83764", "lon": "-82.59790", "
40-1535813644-adhoc-0 {"ident": "40", "id": "40-1535813644-adhoc-0", "clock": "1535817880", "lat": "39.84276", "lon": "-82.58960", "
airline-0261 {"ident": "A07185", "id": "A07185-1535606746-airline-0261", "clock": "1535817850", "lat": "-36.93115", "lon": "-57.6876
airline-0261 {"ident": "A07185", "id": "A07185-1535606746-airline-0261", "clock": "1535817881", "lat": "-36.89484", "lon": "-57.6914
airline-0116 {"ident": "AAF750", "id": "AAF750-1535606759-airline-0116", "clock": "1535817850", "lat": "42.49988", "lon": "4.70770",
airline-0116 {"ident": "AAF750", "id": "AAF750-1535606759-airline-0116", "clock": "1535817882", "lat": "42.56271", "lon": "4.71154",
```

- Mapper Output File is sorted by Key-Value, so all keys are grouped next to each other

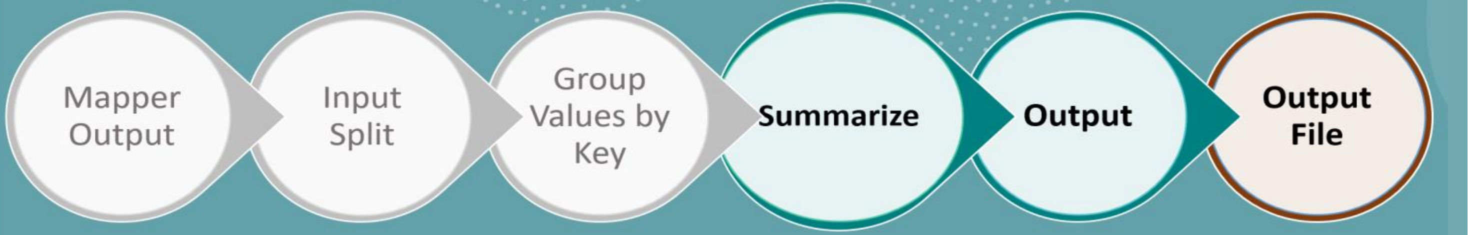

```python
previous_key = ""
data_array = []
```

```python
def grouper(line):

    global previous_key, data_array
    key, value = line.split("\t", 1)

    if (key == previous_key or previous_key == ""):
        data_array.append(json.loads(value))
    else:
        summarize(data_array)
        data_array = [json.loads(value)]

    previous_key = key
```

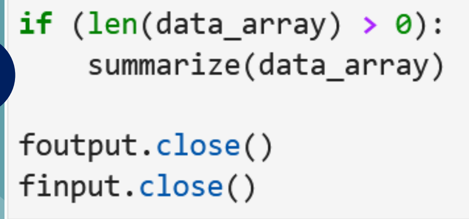
- The reducer keeps all the values of a same key together

- When a new key appears, the collected values are passed for summarization

Reducer: Summarize & Output



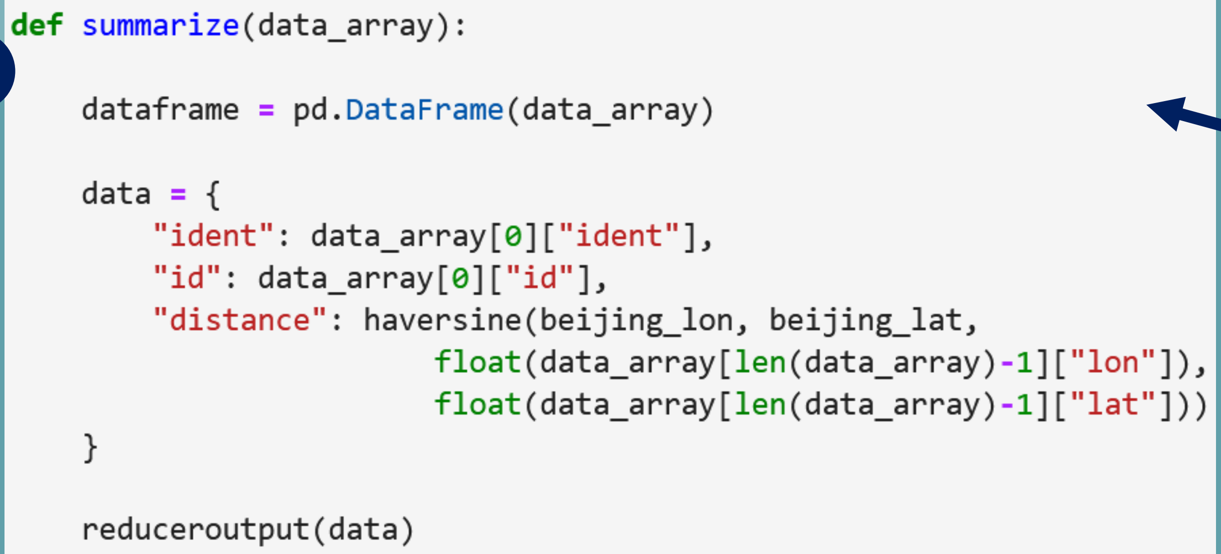
1



```
if (len(data_array) > 0):
    summarize(data_array)

foutput.close()
finput.close()
```

- Each set of data array is passed to ‘summarize’ function

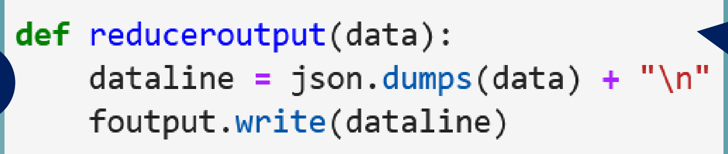
2



```
def summarize(data_array):

    dataframe = pd.DataFrame(data_array)

    data = {
        "ident": data_array[0]["ident"],
        "id": data_array[0]["id"],
        "distance": haversine(beijing_lon, beijing_lat,
                        float(data_array[len(data_array)-1]["lon"]),
                        float(data_array[len(data_array)-1]["lat"]))
    }

    reduceroutput(data)
```

- The summarize function to form data:
  - takes an array of values as input
  - grabs “ident” & “id” data
  - computes ‘distance’ from “lat” & “lon” data via ‘haversine’ function
- Data is then passed to ‘reduceroutput’ function

3



```
def reduceroutput(data):
    dataline = json.dumps(data) + "\n"
    foutput.write(dataline)
```

- A new JSON output object is built with sums and any other calculation as a result of the dataframe
- The function produces the new JSON object as an output.

Summarize: Distance Calculation

Haversine Function:

1. Parameters: latitudes **ϕ1** **ϕ2** & longitudes **λ1** **λ2** of two points
2. Convert latitudes & longitudes in decimal degrees to radians, using ‘*map*’ function
3. Haversine formula:
   - a = Apply sine and cosine
   - c = Square root to ‘a’, multiply 2, and apply arcsine
   - Return Value = apply **r** to ‘c’ to return distance in km

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

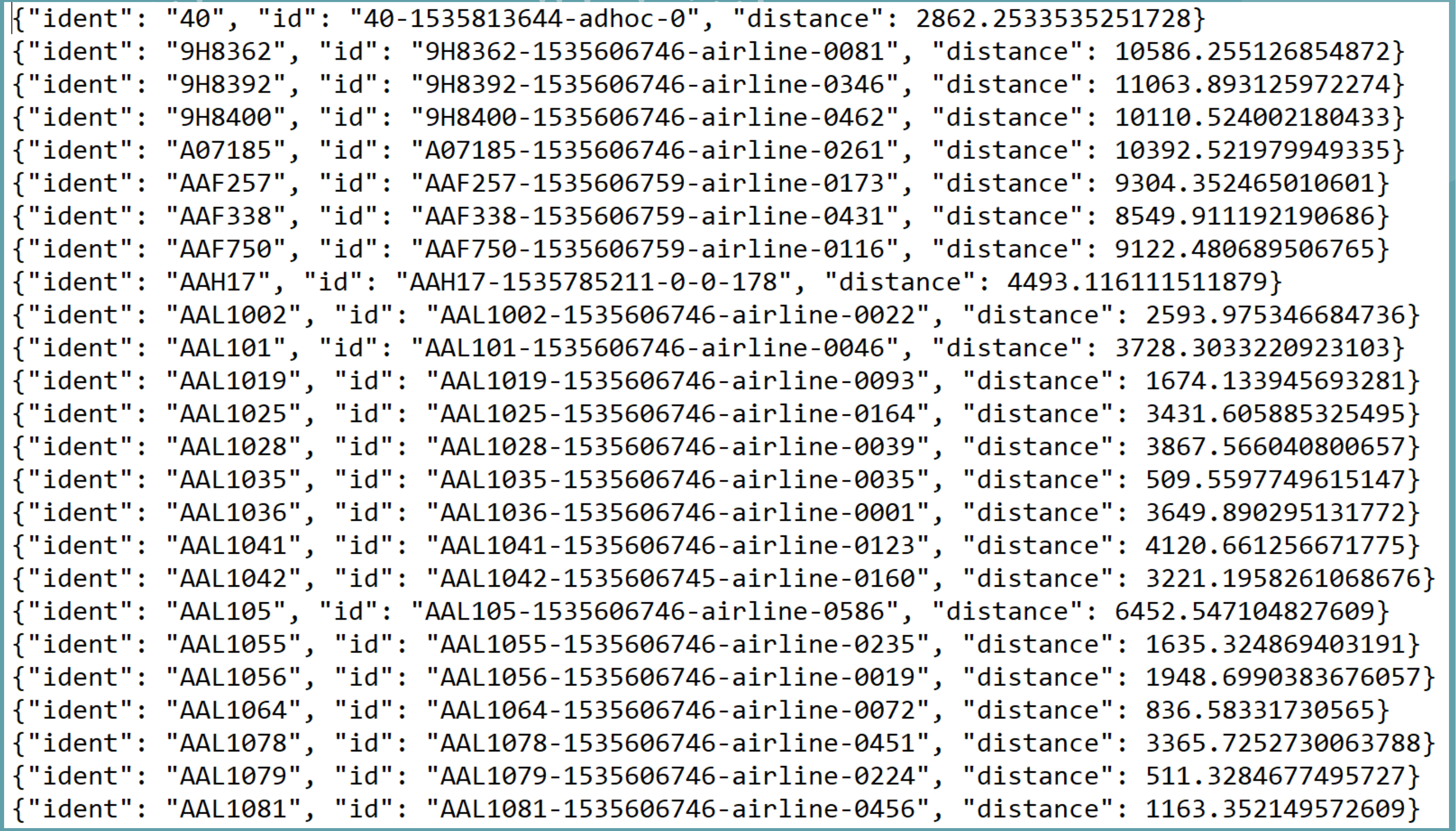
ϕ1	Latitude of Beijing	**ϕ2**	Latitude of Dataset Point
λ1	Longitude of Beijing	**λ2**	Longitude of Dataset Point
r	6,371 - Radius of earth in kilometers		
d	Distance in kilometers		

Reducer: Output File

- The output in a text file with lines of JSON objects

- Each line is flight data relating to the following keys:
  ➔ 'indent'
  ➔ 'id'
  ➔ 'distance'

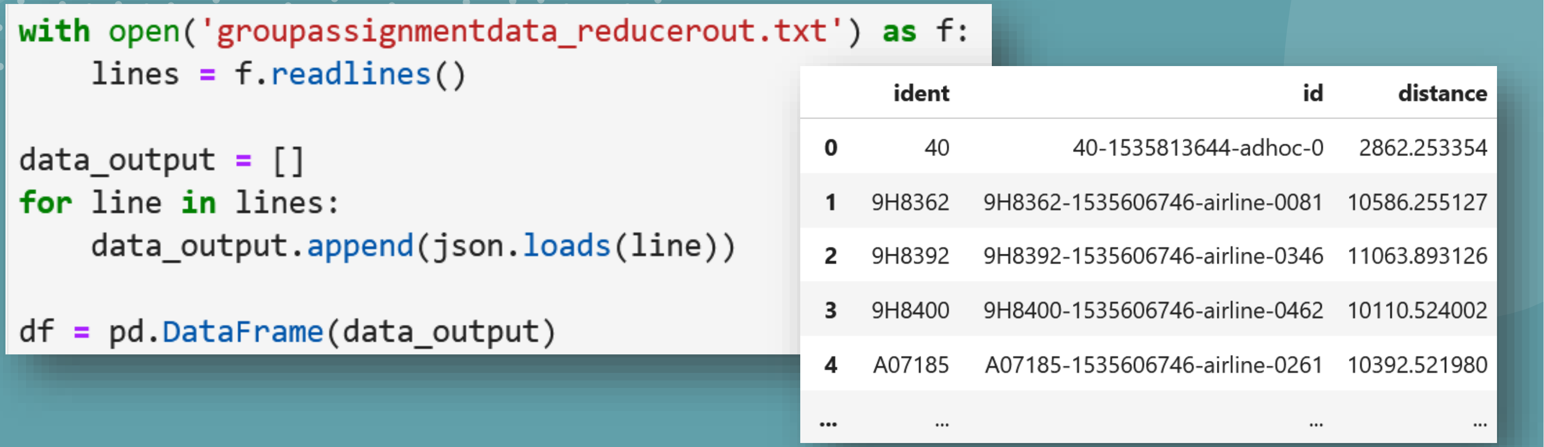
- There are 9,747 lines generated from 'reduceroutput' function



```
{"ident": "40", "id": "40-1535813644-adhoc-0", "distance": 2862.253535251728}
{"ident": "9H8362", "id": "9H8362-1535606746-airline-0081", "distance": 10586.25512685487 2}
{"ident": "9H8392", "id": "9H8392-1535606746-airline-0346", "distance": 11063.89312597227 4}
{"ident": "9H8400", "id": "9H8400-1535606746-airline-0462", "distance": 10110.52400218043 3}
{"ident": "A07185", "id": "A07185-1535606746-airline-0261", "distance": 10392.521979949335}
{"ident": "AAF257", "id": "AAF257-1535606759-airline-0173", "distance": 9304.352465010601}
{"ident": "AAF338", "id": "AAF338-1535606759-airline-0431", "distance": 8549.911192190686}
{"ident": "AAF750", "id": "AAF750-1535606759-airline-0116", "distance": 9122.480689506765}
{"ident": "AAH17", "id": "AAH17-1535785211-0-0-178", "distance": 4493.11611511879}
{"ident": "AAL1002", "id": "AAL1002-1535606746-airline-0022", "distance": 2593.975346684736}
{"ident": "AAL101", "id": "AAL101-1535606746-airline-0046", "distance": 3728.303322092310 3}
{"ident": "AAL1019", "id": "AAL1019-1535606746-airline-0093", "distance": 1674.133945693281}
{"ident": "AAL1025", "id": "AAL1025-1535606746-airline-0164", "distance": 3431.605885325495}
{"ident": "AAL1028", "id": "AAL1028-1535606746-airline-0039", "distance": 3867.566040800657}
{"ident": "AAL1035", "id": "AAL1035-1535606746-airline-0035", "distance": 509.5597749615147}
{"ident": "AAL1036", "id": "AAL1036-1535606746-airline-0001", "distance": 3649.890295131772}
{"ident": "AAL1041", "id": "AAL1041-1535606746-airline-0123", "distance": 4120.661256671775}
{"ident": "AAL1042", "id": "AAL1042-1535606745-airline-0160", "distance": 3221.958261068676}
{"ident": "AAL105", "id": "AAL105-1535606746-airline-0586", "distance": 6452.547104827609}
{"ident": "AAL1055", "id": "AAL1055-1535606746-airline-0235", "distance": 1635.324869403191}
{"ident": "AAL1056", "id": "AAL1056-1535606746-airline-0019", "distance": 1948.6990383676057}
{"ident": "AAL1064", "id": "AAL1064-1535606746-airline-0072", "distance": 836.58331730565}
{"ident": "AAL1078", "id": "AAL1078-1535606746-airline-0451", "distance": 3365.7252730063788}
{"ident": "AAL1079", "id": "AAL1079-1535606746-airline-0224", "distance": 511.3284677495727}
{"ident": "AAL1081", "id": "AAL1081-1535606746-airline-0456", "distance": 1163.352149572609}
```

Create CSV List of All Flights

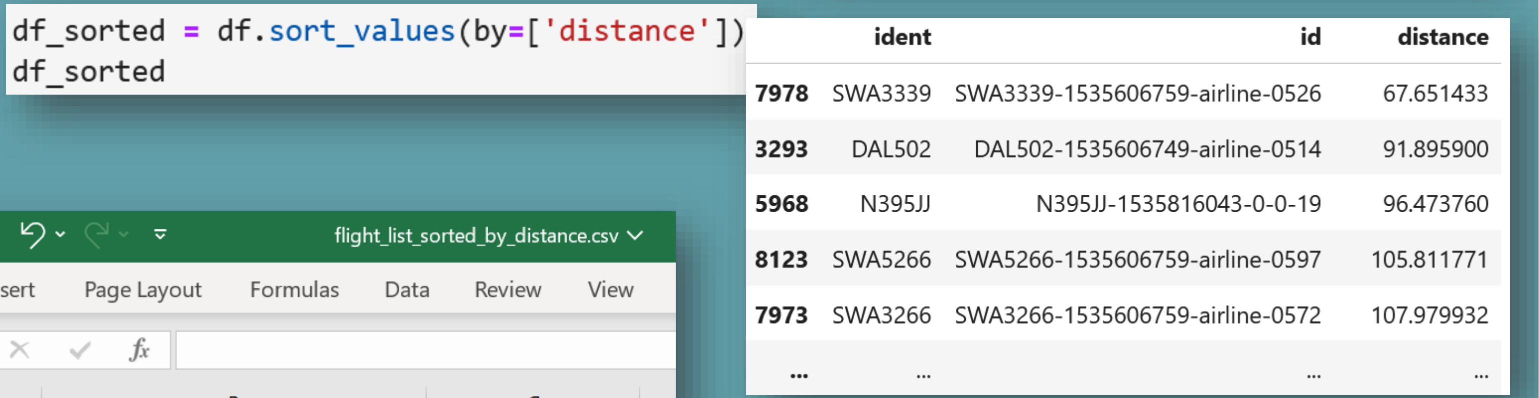
1 Convert JSON objects in text file to Pandas dataframe



```
with open('groupassignmentdata_reducerout.txt') as f:
    lines = f.readlines()

data_output = []
for line in lines:
    data_output.append(json.loads(line))

df = pd.DataFrame(data_output)
```

	ident	id	distance
0	40	40-1535813644-adhoc-0	2862.253354
1	9H8362	9H8362-1535606746-airline-0081	10586.255127
2	9H8392	9H8392-1535606746-airline-0346	11063.893126
3	9H8400	9H8400-1535606746-airline-0462	10110.524002
4	A07185	A07185-1535606746-airline-0261	10392.521980
...	...	...	...

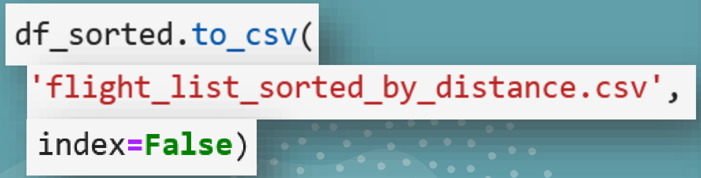
2 Sort dataframe by distance in ascending order



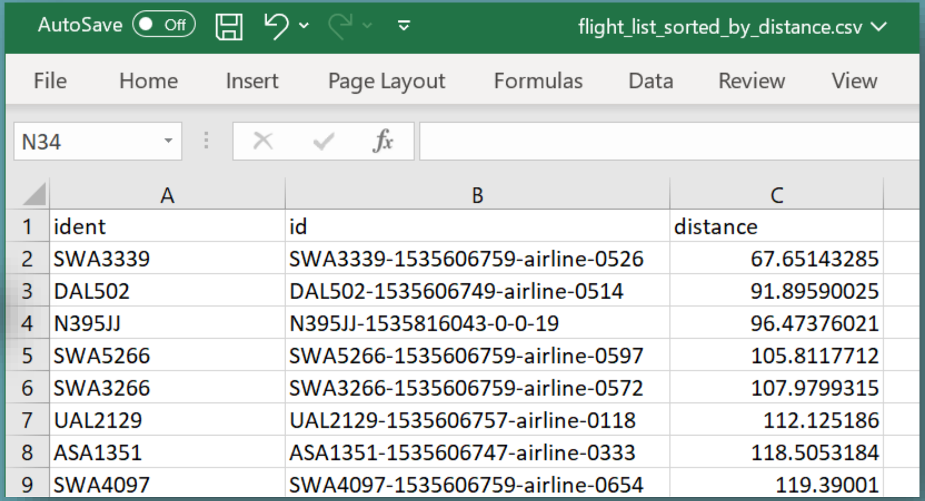
```
df_sorted = df.sort_values(by=['distance'])
df_sorted
```

	ident	id	distance
7978	SWA3339	SWA3339-1535606759-airline-0526	67.651433
3293	DAL502	DAL502-1535606749-airline-0514	91.895900
5968	N395JJ	N395JJ-1535816043-0-0-19	96.473760
8123	SWA5266	SWA5266-1535606759-airline-0597	105.811771
7973	SWA3266	SWA3266-1535606759-airline-0572	107.979932
...	...	...	...

3 Export dataframe to CSV file



```
df_sorted.to_csv(
 'flight_list_sorted_by_distance.csv',
 index=False)
```



	A	B	C
1	ident	id	distance
2	SWA3339	SWA3339-1535606759-airline-0526	67.65143285
3	DAL502	DAL502-1535606749-airline-0514	91.89590025
4	N395JJ	N395JJ-1535816043-0-0-19	96.47376021
5	SWA5266	SWA5266-1535606759-airline-0597	105.8117712
6	SWA3266	SWA3266-1535606759-airline-0572	107.9799315
7	UAL2129	UAL2129-1535606757-airline-0118	112.125186
8	ASA1351	ASA1351-1535606747-airline-0333	118.5053184
9	SWA4097	SWA4097-1535606759-airline-0654	119.39001

Conclusion

- Average distance for 9,747 flights was 8,382 km.
- Flight with the closest distance relative to Beijing was ***China Southern Airline*** (CSN6284). The distance was 20 km.
- Flight with the furthest distance was ***Aerolineas Argentinas Flight*** (ARG1554). The distance was 19,722 km.
- ***American Airlines Group (AAL)*** had the most flights (443 times), and its relative average distance was 10,976 km.
- ***Grand China Air (GDC)*** had the lowest average distance of 71 km, and ***Andes Linease Aereas Flight (ANS)*** had the highest average distance of 19,589 km.

Thank You

Merci