

Vicente De Leon

Big Data Applications

Indiana University

Hadoop Assignment 5

Task 1: Understanding Chicago Crime Data

We are working with two different datasets “data1” and “data2”.

Data1 (Please look at Hadoop_Assignment5.ipynb which will be in the final submission): I did some basic data analysis to understand and see the type of data we are facing here. We have no missing values, and we are dealing with a wide range of victim counts.

```
data1 = pd.read_csv("/content/Violence_Reduction_-_Victim_Demographics_-_Aggregated.csv")
data1.head(10)
```

	TIME_PERIOD	VICTIMIZATION_PRIMARY	AGE	SEX	RACE	NUMBER_OF_VICTIMS
0	3/31/2011	HOMICIDE	30-39	M	BLK	1
1	12/31/2000	HOMICIDE	70-79	M	WHI	2
2	12/31/1992	HOMICIDE	70-79	M	WHI	1
3	12/31/2012	BATTERY	20-29	UNKNOWN	UNKNOWN	1
4	6/30/2003	CRIMINAL SEXUAL ASSAULT	0-19	F	WWH	42
5	9/30/2008	BATTERY	30-39	M	WBH	1
6	12/31/2015	ROBBERY	30-39	F	API	9
7	12/31/2017	ASSAULT	0-19	M	WWH	2
8	3/31/2002	ASSAULT	30-39	M	API	11
9	3/31/2010	BATTERY	0-19	M	BLK	38

data1.dtypes		data1.isnull().sum()		data1.nunique() # https://www.geeksforgeeks.org/python-nunique-method/	
TIME_PERIOD	object	TIME_PERIOD	0	TIME_PERIOD	132
VICTIMIZATION_PRIMARY	object	VICTIMIZATION_PRIMARY	0	VICTIMIZATION_PRIMARY	7
AGE	object	AGE	0	AGE	9
SEX	object	SEX	0	SEX	3
RACE	object	RACE	0	RACE	7
NUMBER_OF_VICTIMS	int64	NUMBER_OF_VICTIMS	0	NUMBER_OF_VICTIMS	420
dtype: object		dtype: int64		dtype: int64	

```
Victimization Primary values:
['HOMICIDE' 'BATTERY' 'CRIMINAL SEXUAL ASSAULT' 'ROBBERY' 'ASSAULT'
 'NON-FATAL' 'HUMAN TRAFFICKING']

Age Range:
['30-39' '70-79' '20-29' '0-19' '40-49' 'UNKNOWN' '50-59' '60-69' '80+']

Unique Sex Value:
['M' 'UNKNOWN' 'F']

Unique Race Values:
['BLK' 'WHI' 'UNKNOWN' 'WWH' 'WBH' 'API' 'I']
```

Data2 (Please look at Hadoop_Assignment5.ipynb which will be in the final submission): this dataset had 5 missing values in column “COMMUNITY_AREA”, which I filled with “UNKOWN” value. Why had I decided to do this? Because this column seems to be an area description where an event happened. Also, I wanted to match the “UNKOWN” values form Data1. Additionally, Data2 provides much more information regarding events, blocks, and location areas.

```
data2 = pd.read_csv("/content/Violence_Reduction_-_Victims_of_Homicides_and_Non-Fatal_Shootings-1.csv")
data2.head(10)
```

	DATE	BLOCK	LOCATION_DESCRIPTION	COMMUNITY_AREA	VICTIMIZATION_PRIMARY	AGE	SEX	RACE	GUNSHOT_INJURY_I
0	3/8/2022 15:27	6000 N KENMORE AVE	APARTMENT	EDGEWATER	HOMICIDE	60-69	M	BLK	NO
1	1/13/2018 1:25	900 W 114TH PL	RESIDENCE	MORGAN PARK	BATTERY	0-19	M	BLK	YES
2	5/16/2008 23:46	9200 S RACINE AVE	ALLEY	WASHINGTON HEIGHTS	HOMICIDE	40-49	M	BLK	YES
3	6/25/2022 0:41	2100 W 36TH ST	STREET	MCKINLEY PARK	HOMICIDE	30-39	M	WWH	YES
4	2/12/2023 17:57	400 E 83RD ST	RESTAURANT	CHATHAM	HOMICIDE	20-29	M	BLK	YES
5	2/10/2023 10:24	7400 S COLFAX AVE	APARTMENT	SOUTH SHORE	HOMICIDE	30-39	M	BLK	YES
6	4/7/2023 12:25	4400 W WEST END AVE	STREET	WEST GARFIELD PARK	HOMICIDE	20-29	M	BLK	YES
7	6/5/2019 13:02	6400 S TALMAN AVE	OTHER (SPECIFY)	CHICAGO LAWN	BATTERY	20-29	M	BLK	YES
8	4/8/2020 14:40	5400 W HIRSCH ST	STREET	AUSTIN	BATTERY	20-29	M	BLK	YES
9	4/8/2020 14:40	5400 W HIRSCH ST	STREET	AUSTIN	BATTERY	20-29	M	BLK	YES

```
data2['COMMUNITY_AREA'] = data2['COMMUNITY_AREA'].fillna('UNKOWN') # changed those missing values to UNKOWN
```

data2.dtypes		data2.isnull().sum()		data2.isnull().sum()	
DATE	object	DATE	0	DATE	0
BLOCK	object	BLOCK	0	BLOCK	0
LOCATION_DESCRIPTION	object	LOCATION_DESCRIPTION	0	LOCATION_DESCRIPTION	0
COMMUNITY_AREA	object	COMMUNITY_AREA	5	COMMUNITY_AREA	0
VICTIMIZATION_PRIMARY	object	VICTIMIZATION_PRIMARY	0	VICTIMIZATION_PRIMARY	0
AGE	object	AGE	0	AGE	0
SEX	object	SEX	0	SEX	0
RACE	object	RACE	0	RACE	0
GUNSHOT_INJURY_I	object	GUNSHOT_INJURY_I	0	GUNSHOT_INJURY_I	0
dtype: object		dtype: int64		dtype: int64	

```
unique_ca = data2["COMMUNITY_AREA"].unique()
print("Unique Community Area values: ")
print(unique_ca)
```

Unique Community Area values:

['EDGEWATER' 'MORGAN PARK' 'WASHINGTON HEIGHTS' 'MCKINLEY PARK' 'CHATHAM'
'SOUTH SHORE' 'WEST GARFIELD PARK' 'CHICAGO LAWN' 'AUSTIN' 'LINCOLN PARK'
'WEST ENGLEWOOD' 'AUBURN GRESHAM' 'ENGLEWOOD' 'DUNNING' 'WEST TOWN'
'DOUGLAS' 'HUMBOLDT PARK' 'BELMONT CRAGIN' 'NEW CITY' 'WASHINGTON PARK'
'LOGAN SQUARE' 'SOUTH CHICAGO' 'GREATER GRAND CROSSING' 'NEAR NORTH SIDE'
'GRAND BOULEVARD' 'SOUTH LAWDALE' 'AVONDALE' 'ROSELAND' 'NEAR WEST SIDE'
'NORTH LAWDALE' 'SOUTH DEERING' 'ROGERS PARK' 'EAST GARFIELD PARK'
'WEST PULLMAN' 'LOWER WEST SIDE' 'NORTH CENTER' 'WOODLAWN' 'HERMOSA'
'ASHBURN' 'ALBANY PARK' 'BURNSIDE' 'LOOP' 'HYDE PARK' 'GARFIELD RIDGE'
'CALUMET HEIGHTS' 'IRVING PARK' 'KENWOOD' 'UPTOWN' 'BRIGHTON PARK'
'AVALON PARK' 'GAGE PARK' 'RIVERDALE' 'UNKNOWN' 'EAST SIDE' 'WEST LAWN'
'FOREST GLEN' 'FULLER PARK' 'BRIDGEPORT' 'PORTAGE PARK' 'PULLMAN'
'NEAR SOUTH SIDE' 'WEST ELSDON' 'JEFFERSON PARK' 'OAKLAND' 'WEST RIDGE'
'ARCHER HEIGHTS' 'BEVERLY' 'ARMOUR SQUARE' 'LAKE VIEW' 'MONTCLARE'
'HEGEWISCH' 'MOUNT GREENWOOD' 'LINCOLN SQUARE' 'NORTH PARK' 'CLEARING'
'OHARE' 'NORWOOD PARK' 'EDISON PARK']

The relationship between both datasets is that they have common columns such as "VICTIMIZATION_PRIMARY", "AGE", "SEX", and "RACE". Another characteristic that I notice from both datasets is that Data1 seems to be a more aggregated in terms of victim demographics for many victimizations over specific time periods. In the other hand, Data 2 is bigger and shows much more details such as dates, locations etc (victims of homicides and non-fatal gunshots).

Data1 Columns:

- TIME_PERIOD: date.
- VICTIMIZATION_PRIMARY: nature of crime.
- AGE: age of victim.
- SEX: gender of victim.
- RACE: race of victim.
- NUMBER_OF_VICTIMS: count of victims.

Data2 Columns:

- DATE: date.
- BLOCK: block where event happened.
- LOCATION_DESCRIPTION: location type.
- COMMUNITY_AREA: area where event occurred.
- VICTIMIZATION_PRIMARY: nature of crime.
- AGE: age of victim.
- SEX: gender of victim.
- RACE: race of victim.
- GUNSHOT_INJURY_I: gunshot or not.

Data1: <https://catalog.data.gov/dataset/violence-reduction-victim-demographics-aggregated>

Data2: <https://catalog.data.gov/dataset/violence-reduction-victims-of-homicides-and-non-fatal-shootings>

We can see both datasets are comma-separated. Due to this, I had to change how the mapper what reading the data. See image below:

```
with open("/content/data1.csv", "r") as f:
    for _ in range(5): # first 5 lines
        print(f.readline().strip())

    print("\n")

with open("/content/data2.csv", "r") as f:
    for _ in range(5): # first 5 lines
        print(f.readline().strip())
```



```
TIME_PERIOD,VICTIMIZATION_PRIMARY,AGE,SEX,RACE,NUMBER_OF_VICTIMS
3/31/2011,HOMICIDE,30-39,M,BLK,1
12/31/2000,HOMICIDE,70-79,M,WHI,2
12/31/1992,HOMICIDE,70-79,M,WHI,1
12/31/2012,BATTERY,20-29,UNKNOWN,UNKNOWN,1

DATE,BLOCK,LOCATION_DESCRIPTION,COMMUNITY_AREA,VICTIMIZATION_PRIMARY,AGE,SEX,RACE,GUNSHOT_INJURY_I
3/8/2022 15:27,6000 N KENMORE AVE,APARTMENT,EDGEWATER,HOMICIDE,60-69,M,BLK,NO
1/13/2018 1:25,900 W 114TH PL,RESIDENCE,MORGAN PARK,BATTERY,0-19,M,BLK,YES
5/16/2008 23:46,9200 S RACINE AVE,ALLEY,WASHINGTON HEIGHTS,HOMICIDE,40-49,M,BLK,YES
6/25/2022 0:41,2100 W 36TH ST,STREET,MCKINLEY PARK,HOMICIDE,30-39,M,WWH,YES
```

We initiate this process by writing the mapper.py and reducer.py code we are going to be using to tackle Task 2. It's important to always include `#!/usr/bin/env python3` on each script and to use `sys` for reading input from `stdin` and writing output to `stdout` just like we did in HW4.

Task2: MapReduce Job Join Operation

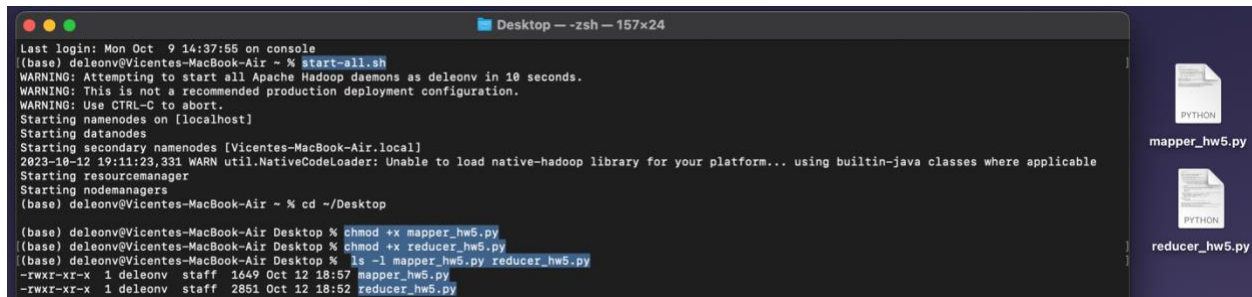
Mapper_hw5.py: This code basically, just like hw4, reads each line from the input data. It uses the VICTIMIZATION_PRIMARY field (column) as the key for both datasets (data1 and data2) and it appends either dataset to indicate which dataset the data row originates from (data1 vs data2) and prints output results (separating key and value with a tab).

```
mapper_hw5.py x
Users > delevon > Desktop > mapper_hw5.py > ...
1  #!/usr/bin/env python3
2
3  import sys
4
5  for line in sys.stdin: # reading line by line
6      line = line.strip() # removing whitespaces from input line
7      cols = line.split(",") # splitting line using comma (data is comma_separated)
8
9      # Some Python logic based on datasets columns Stack Overflow source
10     if len(cols) == 6: # data1 has 6 columns we need VICTIMIZATION_PRIMARY
11         key = cols[1] # VICTIMIZATION_PRIMARY Data1
12         value = "data1.csv," + ",".join(cols) # value from data1 + columns joined into a single string using ","
13     else:
14         key = cols[4] # VICTIMIZATION_PRIMARY Data2
15         value = "data2.csv," + ",".join(cols) # value from data2 + columns joined into a single string using ","
16
17     print(key + "\t" + value)
```

Reducer_hw5.py: This code aggregates data rows based on the key -> VICTIMIZATION_PRIMARY field (column). It basically pairs each row from data 1 with every matching row from data2 based on the same key. Finally, it prints output results.

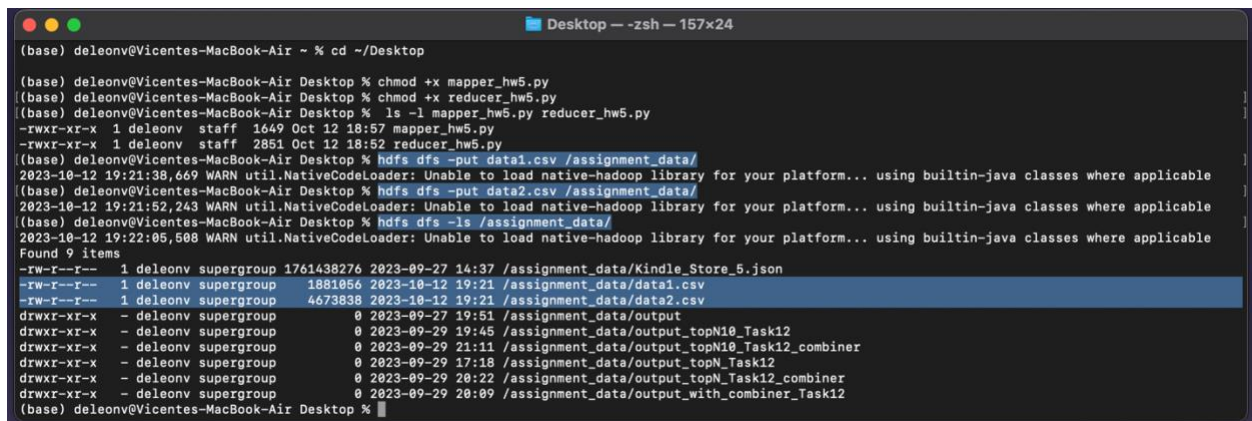
```
reducer_hw5.py x
Users > delevon > Desktop > reducer_hw5.py > ...
1  #!/usr/bin/env python3
2
3  import sys
4
5  current_key = None # current key being processed set to None
6  data1_rows = [] # empty list to store rows from Data1
7  data2_rows = [] # empty list to store rows from Data2
8
9  for line in sys.stdin: # reading line by line
10     line = line.strip() # removing whitespaces from input line
11     key, value = line.split("\t", 1) # splitting lines into columns key and value. 1 will allow values to contain tabs
12
13     # The following code is kind of tricky:
14     # if key is not equal to current key -> it means the program encountered a new key in the input data
15
16     if key != current_key: # checking -> new key from input line vs current processed key
17         if current_key is not None: # checking that it's not the first key encountered
18             for row1 in data1_rows: # iterating through rows from data1 dataset
19                 for row2 in data2_rows: # iterating through rows from data2 dataset
20                     print(row1 + "\t" + row2) # print joined results print(row1 + " " + row2)
21
22         # Let's reset.
23         # We need to reset because it's the only way the code will be able to handle the next key in the input data without problems.
24         # It is important to remember that each key represents a unique group of data records.
25         # When key!= current_key (detecting new key), means we are going into a different group of data record with a new key.
26         # That's why we need to reset, to prepare for new key.
27         current_key = key
28         data1_rows = []
29         data2_rows = []
30
31     dataset, row = value.split(",", 1) # splitting value (separated by "," in mapper.py) into two parts using ("," and 1) as well -> dataset and row
32     # Value format was build with a comma separating the dataset name and the row data.
33     if dataset == "data1.csv": # if dataset equals to dataset 1
34         data1_rows.append(row) # if it does, lets append the row to data1_rows list
35     elif dataset == "data2.csv": # dataset 2
36         data2_rows.append(row) # lets append the row to data2_rows list
37
38     # Repeat process
39     if current_key is not None:
40         for row1 in data1_rows:
41             for row2 in data2_rows:
42                 print(row1 + "\t" + row2) # print joined results print(row1 + " " + row2)
```

Let's start by initiating Hadoop Services and added both Python scripts to make them executable:



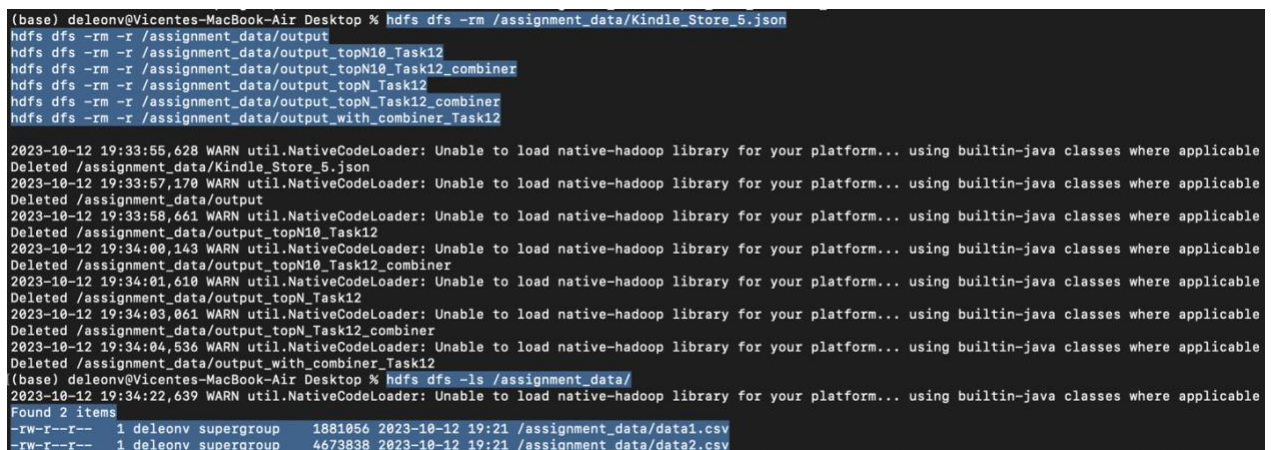
```
(base) deleonv@Vicentes-MacBook-Air ~ % start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as deleonv in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [Vicentes-MacBook-Air.local]
2023-10-12 19:11:23,331 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting resourcemanager
Starting nodemanagers
(base) deleonv@Vicentes-MacBook-Air ~ % cd ~/Desktop
(base) deleonv@Vicentes-MacBook-Air Desktop % chmod +x mapper_hw5.py
(base) deleonv@Vicentes-MacBook-Air Desktop % chmod +x reducer_hw5.py
(base) deleonv@Vicentes-MacBook-Air Desktop % ls -l mapper_hw5.py reducer_hw5.py
-rwxr-xr-x 1 deleonv staff 1649 Oct 12 18:57 mapper_hw5.py
-rwxr-xr-x 1 deleonv staff 2851 Oct 12 18:52 reducer_hw5.py
```

I am going to take advantage from the "assignment_data" HDFS I created for HW4, and I am going to store both csv files in there. As you can see in the image below, I used the necessary commands to insert files and display HDFS directory content:



```
(base) deleonv@Vicentes-MacBook-Air ~ % cd ~/Desktop
(base) deleonv@Vicentes-MacBook-Air Desktop % chmod +x mapper_hw5.py
(base) deleonv@Vicentes-MacBook-Air Desktop % chmod +x reducer_hw5.py
(base) deleonv@Vicentes-MacBook-Air Desktop % ls -l mapper_hw5.py reducer_hw5.py
-rwxr-xr-x 1 deleonv staff 1649 Oct 12 18:57 mapper_hw5.py
-rwxr-xr-x 1 deleonv staff 2851 Oct 12 18:52 reducer_hw5.py
(base) deleonv@Vicentes-MacBook-Air Desktop % hdfs dfs -put data1.csv /assignment_data/
2023-10-12 19:21:38,669 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
(base) deleonv@Vicentes-MacBook-Air Desktop % hdfs dfs -put data2.csv /assignment_data/
2023-10-12 19:21:52,243 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
(base) deleonv@Vicentes-MacBook-Air Desktop % hdfs dfs -ls /assignment_data/
2023-10-12 19:22:05,508 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 9 items
-rw-r--r-- 1 deleonv supergroup 1761438276 2023-09-27 14:37 /assignment_data/Kindle_Store_5.json
-rw-r--r-- 1 deleonv supergroup 1881056 2023-10-12 19:21 /assignment_data/data1.csv
-rw-r--r-- 1 deleonv supergroup 4673838 2023-10-12 19:21 /assignment_data/data2.csv
drwxr-xr-x - deleonv supergroup 0 2023-09-27 19:51 /assignment_data/output
drwxr-xr-x - deleonv supergroup 0 2023-09-29 19:45 /assignment_data/output_topN10_Task12
drwxr-xr-x - deleonv supergroup 0 2023-09-29 21:11 /assignment_data/output_topN10_Task12_combiner
drwxr-xr-x - deleonv supergroup 0 2023-09-29 17:18 /assignment_data/output_topN_Task12
drwxr-xr-x - deleonv supergroup 0 2023-09-29 20:22 /assignment_data/output_topN_Task12_combiner
drwxr-xr-x - deleonv supergroup 0 2023-09-29 20:09 /assignment_data/output_with_combiner_Task12
(base) deleonv@Vicentes-MacBook-Air Desktop %
```

As you can see, there's a lot of content we don't really need for this assignment. I am going to use the same commands used in HW4, to delete this extra content (hdfs dfs -rm and hdfs dfs -rm -r) as shown below:



```
(base) deleonv@Vicentes-MacBook-Air Desktop % hdfs dfs -rm /assignment_data/Kindle_Store_5.json
hdfs dfs -rm -r /assignment_data/output
hdfs dfs -rm -r /assignment_data/output_topN10_Task12
hdfs dfs -rm -r /assignment_data/output_topN10_Task12_combiner
hdfs dfs -rm -r /assignment_data/output_topN_Task12
hdfs dfs -rm -r /assignment_data/output_topN_Task12_combiner
hdfs dfs -rm -r /assignment_data/output_with_combiner_Task12

2023-10-12 19:33:55,628 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Deleted /assignment_data/Kindle_Store_5.json
2023-10-12 19:33:57,170 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Deleted /assignment_data/output
2023-10-12 19:33:58,661 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Deleted /assignment_data/output_topN10_Task12
2023-10-12 19:34:00,143 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Deleted /assignment_data/output_topN10_Task12_combiner
2023-10-12 19:34:01,610 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Deleted /assignment_data/output_topN_Task12
2023-10-12 19:34:03,061 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Deleted /assignment_data/output_topN_Task12_combiner
2023-10-12 19:34:04,536 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Deleted /assignment_data/output_with_combiner_Task12
(base) deleonv@Vicentes-MacBook-Air Desktop % hdfs dfs -ls /assignment_data/
2023-10-12 19:34:22,639 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 deleonv supergroup 1881056 2023-10-12 19:21 /assignment_data/data1.csv
-rw-r--r-- 1 deleonv supergroup 4673838 2023-10-12 19:21 /assignment_data/data2.csv
```


Just like HW4, this is the code I used to run my MapReduce Job for Task 2. I don't have screenshot from MAC Terminal because my computer went crazy after trying to display all the content from the output results -> part-00000. See image below:

```
40 hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar \
41 -files mapper_hw5.py,reducer_hw5.py \
42 -mapper mapper_hw5.py \
43 -reducer reducer_hw5.py \
44 -input /assignment_data/data1.csv \
45 -input /assignment_data/data2.csv \
46 -output /assignment_data/output_Task2_hw5
```

```

[Restored contents truncated]
3/31/2007, BATTERY, 60-69, M, WH, 1 3/25/2019 14:40, 6000 N FRANCISCO AVE, STREET, LINCOLN SQUARE, BATTERY, 30-39, M, WH, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 1/16/2010 0:21, 1500 N BOONWORTH AVE, SIDEWALK, WEST TOWN, BATTERY, 0-19, M, WH, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 1/10/2010 16:41, 400 E 110TH PL, STREET, ROSELAND, BATTERY, 0-19, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 7/1/2019 15:16, 4000 W VAN BUREN ST, STREET, WEST GARFIELD PARK, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 7/1/2019 0:46, 12000 S STATE ST, SIDEWALK, WEST PULLMAN, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 3/15/2023 11:25, 8000 S HOLLAND RD, PARKING LOT / GARAGE (NON RESIDENTIAL), ROSELAND, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 1/15/2010 5:00, 3700 W CULLOM AVE, APARTMENT, IRVING PARK, BATTERY, 20-29, M, WH, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 6/30/2019 4:12, 4000 W 15TH ST, STREET, NORTH LAWDALE, BATTERY, UNKNOWN, UNKNOWN, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 6/30/2019 3:15, 700 W LOREL AVE, SIDEWALK, AUSTIN, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 6/30/2019 1:30, 4700 W ERIE ST, STREET, AUSTIN, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 6/30/2019 0:49, 12400 S PARNELL AVE, STREET, WEST PULLMAN, BATTERY, 0-19, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 1/24/2010 2:40, 2200 S LAFLIN ST, ALLEY, LOWER WEST SIDE, BATTERY, UNKNOWN, UNKNOWN, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 3/22/2019 13:16, 11300 S MICHIGAN AVE, SIDEWALK, ROSELAND, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 6/30/2019 0:49, 12400 S PARNELL AVE, STREET, WEST PULLMAN, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 6/29/2019 21:04, 12200 S MAY ST, STREET, WEST PULLMAN, BATTERY, 40-49, F, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 3/20/2019 21:09, 7000 S YATES BLVD, ALLEY, SOUTH SHORE, BATTERY, 0-19, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 3/19/2019 20:14, 1000 W ARDYLE ST, SIDEWALK, UPTOWN, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 1/12/2010 18:46, 900 W 32ND ST, SIDEWALK, BRIDGEPORT, BATTERY, 0-19, M, WH, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 1/10/2010 22:45, 3200 S LITUANICA AVE, STREET, BRIDGEPORT, BATTERY, 0-19, M, WH, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 3/19/2019 20:14, 1000 W ARDYLE ST, SIDEWALK, UPTOWN, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 6/29/2019 20:29, 5100 S CALUMET AVE, VACANT LOT / LAND, WASHINGTON PARK, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 3/19/2019 5:11, 2500 W AUGUSTA BLVD, STREET, WEST TOWN, BATTERY, 20-29, M, BLK, YES
3/31/2007, BATTERY, 60-69, M, WH, 1 6/29/2019 9:48, 12200 S EMERALD AVE, STREET, WEST PULLMAN, BATTERY, 30-39, M, BLK, YES

```

The above commands: run command line 50 to view entire content from output (we need to focus on results part-00000), run command line 52 to view results (do not run this your computer will crash just like mine “Your system has run out of application memory”), run command line 54 to export results into csv file (this will also be in final submission).

```

50 type to see entire output content: hdfs dfs -ls /assignment_data/output_Task2_hw5
51
52 type to see part-00000 content: type: hdfs dfs -cat /assignment_data/output_Task2_hw5/part-00000 (DO NOT RUN!)
53
54 type to export data into csv file: hdfs dfs -get /assignment_data/output_Task2_hw5/part-00000 ~/Desktop/output_Task2_hw5.csv

```

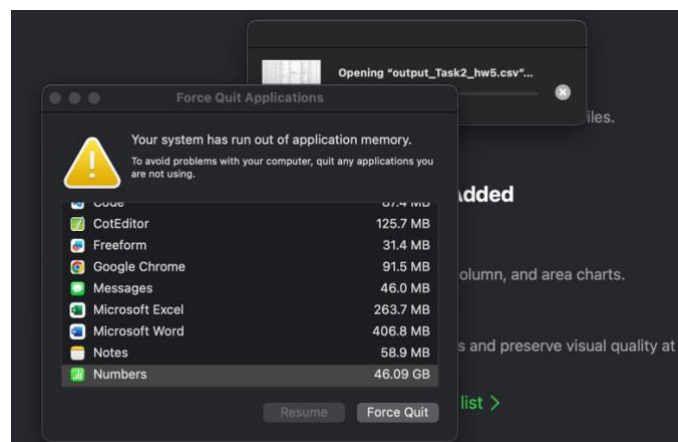
MAC Terminal commands:

```

(base) deleonv@Vicentes-MacBook-Air Desktop % hdfs dfs -ls /assignment_data/output_Task2_hw5
2023-10-12 22:44:36,070 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 deleonv supergroup 0 2023-10-12 22:13 /assignment_data/output_Task2_hw5/_SUCCESS
-rw-r--r-- 1 deleonv supergroup 76769910864 2023-10-12 22:13 /assignment_data/output_Task2_hw5/part-00000
(base) deleonv@Vicentes-MacBook-Air Desktop % hdfs dfs -get /assignment_data/output_Task2_hw5/part-00000 ~/Desktop/output_Task2_hw5.csv
2023-10-12 22:46:04,248 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

```

I tried opening the csv file via Excel and MAC's Numbers, but I had no success due to RAM issues:



I decided to save output results in csv file format because the output has multiple columns. As I wrote earlier, the csv file is an extremely huge file that makes my computer crash due to RAM limitations, I used the following resource to view the 10 first lines from “output_Task2_hw5.csv” file using my MAC Terminal:

Viewing csv file from MAC Terminal command: <https://victorl.in/previewing-csv-files/>

As you can see, the output join results have the following format: <columns data1.csv> <columns data2.csv>. All columns from both datasets are shown in the MAC Terminal CSV output image below:

```

Last login: Thu Oct 12 23:03:53 on ttys001
(base) deleonv@Vicentes-MacBook-Air ~ % head -n 10 ~/Desktop/output_Task2_hw5.csv
9/30/2023,BATTERY,0-19,F,BLK,25 2/10/2021 19:53,3900 W MADISON ST,PARKING LOT / GARAGE (NON RESIDENTIAL),WEST GARFIELD PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 12/29/2011 1:05,1600 N LOTUS AVE,STREET,AUSTIN,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 12/28/2011 18:25,9400 S JUSTINE ST,SIDEWALK,WASHINGTON HEIGHTS,BATTERY,0-19,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/10/2021 18:33,400 N DRAKE AVE,STREET,HUMBOLDT PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 12/28/2011 15:00,11300 S VINCENNES AVE,STREET,MORGAN PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/9/2021 23:15,700 N HAMLIN AVE,ALLEY,HUMBOLDT PARK,BATTERY,30-39,M,WWH,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/8/2021 5:24,4100 W ADAMS ST,SIDEWALK,WEST GARFIELD PARK,BATTERY,30-39,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/8/2021 5:24,4100 W ADAMS ST,SIDEWALK,WEST GARFIELD PARK,BATTERY,40-49,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 6/20/2020 2:26,2700 N PINE GROVE AVE,APARTMENT,LINCOLN PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/7/2021 23:53,7800 S STATE ST,RESIDENCE,GREATER GRAND CROSSING,BATTERY,50-59,M,BLK,YES

```

Data1.csv columns:

	TIME_PERIOD	VICTIMIZATION_PRIMARY	AGE	SEX	RACE	NUMBER_OF_VICTIMS
0	3/31/2011	HOMICIDE	30-39	M	BLK	1
1	12/31/2000	HOMICIDE	70-79	M	WHI	2
2	12/31/1992	HOMICIDE	70-79	M	WHI	1
3	12/31/2012	BATTERY	20-29	UNKNOWN	UNKNOWN	1
4	6/30/2003	CRIMINAL SEXUAL ASSAULT	0-19	F	WWH	42

Data2.csv columns:

	DATE	BLOCK	LOCATION_DESCRIPTION	COMMUNITY_AREA	VICTIMIZATION_PRIMARY	AGE	SEX	RACE	GUNSHOT_INJURY_I
0	3/8/2022 15:27	6000 N KENMORE AVE	APARTMENT	EDGEWATER	HOMICIDE	60-69	M	BLK	NO
1	1/13/2018 1:25	900 W 114TH PL	RESIDENCE	MORGAN PARK	BATTERY	0-19	M	BLK	YES
2	5/16/2008 23:46	9200 S RACINE AVE	ALLEY	WASHINGTON HEIGHTS	HOMICIDE	40-49	M	BLK	YES
3	6/25/2022 0:41	2100 W 36TH ST	STREET	MCKINLEY PARK	HOMICIDE	30-39	M	WWH	YES
4	2/12/2023 17:57	400 E 83RD ST	RESTAURANT	CHATHAM	HOMICIDE	20-29	M	BLK	YES

I can safely say that both mapper_hw5.py and reducer_hw5.py were a success in this Task regarding the MapReduce job. To conclude Task 2, I used the column or field “VICTIMIZATION_PRIMARY” to join two datasets (data1.csv and data2.csv). In the mapper_hw5.py script, each row of the datasets is read, and the script emits a key-value pair where the key is “VICTIMIZATION_PRIMARY” and the value is a combination of the dataset name and the entire row. In the reducer_hw5.py script, it takes the pairs, groups them by the key, and processes rows from both datasets that share the same key. At the end, it then returns results of combination of rows from data1.csv and data2.csv that have the same “VICTIMIZATION_PRIMARY”, which lead to the join operation required for Task 2 completion. The large size of output_Task2_hw5.csv might be due to the high number of matches between the two datasets on “VICTIMIZATION_PRIMARY” column.

Data1.csv, Data2.csv, Assignment 5 Notes text file, Hadoop_Assignment5.ipynb, mapper_hw5.py, reducer_hw5.py, and output_Task2_hw5.csv will be in final submission.

Task 3: New MapReduce Job to Sort Joined Results

The idea behind this task is to extend the join operation from Task 2 and sort the joined results based on the chosen column in ascending or descending order. Again, the output should have the exact same format as Task2 -> <columns dataset1.csv> <columns dataset2.csv>. Also, we already have joined results from Task 2 in which we can try to apply this new sorting idea.

Let's analyze the first 10 rows from Task 2 results:

```
((base) deleonv@Vicentes-MacBook-Air Desktop % head -n 10 ~/Desktop/output_Task2_hw5.csv
9/30/2023,BATTERY,0-19,F,BLK,25 2/10/2021 19:53,3900 W MADISON ST,PARKING LOT / GARAGE (NON RESIDENTIAL),WEST GARFIELD PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 12/29/2011 1:05,1600 N LOTUS AVE,STREET,AUSTIN,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 12/28/2011 18:25,9400 S JUSTINE ST,SIDEWALK,WASHINGTON HEIGHTS,BATTERY,0-19,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/10/2021 18:33,400 N DRAKE AVE,STREET,HUMBOLDT PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 12/28/2011 15:00,11300 S VINCENNES AVE,STREET,MORGAN PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/9/2021 23:15,700 N HAMLIN AVE,ALLEY,HUMBOLDT PARK,BATTERY,30-39,M,WWH,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/8/2021 5:24,4100 W ADAMS ST,SIDEWALK,WEST GARFIELD PARK,BATTERY,30-39,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/8/2021 5:24,4100 W ADAMS ST,SIDEWALK,WEST GARFIELD PARK,BATTERY,40-49,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 6/20/2020 2:26,2700 N PINE GROVE AVE,APARTMENT,LINCOLN PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/7/2021 23:53,7800 S STATE ST,RESIDENCE,GREATER GRAND CROSSING,BATTERY,50-59,M,BLK,YES
```

As we know, our results come from a joined operation of data1.csv + data2.csv based on PRIMARY_VICTIMIZATION column. The first part from (9/30/2023, BATTERY, 0-19, F, BLK, 25) is from data1.csv.

	TIME_PERIOD	VICTIMIZATION_PRIMARY	AGE	SEX	RACE	NUMBER_OF_VICTIMS
0	3/31/2011	HOMICIDE	30-39	M	BLK	1
1	12/31/2000	HOMICIDE	70-79	M	WHI	2
2	12/31/1992	HOMICIDE	70-79	M	WHI	1
3	12/31/2012	BATTERY	20-29	UNKNOWN	UNKNOWN	1
4	6/30/2003	CRIMINAL SEXUAL ASSAULT	0-19	F	WWH	42

The second part from (2/10/2021 19:53, 3900 W MADISON ST, PARKING LOT / GARAGE (NON RESIDENTIAL), WEST GARFIELD PARK, BATTERY, 20-19, M, BLK, YES) comes from data2.csv:

	DATE	BLOCK	LOCATION_DESCRIPTION	COMMUNITY_AREA	VICTIMIZATION_PRIMARY	AGE	SEX	RACE	GUNSHOT_INJURY_I
0	3/8/2022 15:27	6000 N KENMORE AVE	APARTMENT	EDGEWATER	HOMICIDE	60-69	M	BLK	NO
1	1/13/2018 1:25	900 W 114TH PL	RESIDENCE	MORGAN PARK	BATTERY	0-19	M	BLK	YES
2	5/16/2008 23:46	9200 S RACINE AVE	ALLEY	WASHINGTON HEIGHTS	HOMICIDE	40-49	M	BLK	YES
3	6/25/2022 0:41	2100 W 36TH ST	STREET	MCKINLEY PARK	HOMICIDE	30-39	M	WWH	YES
4	2/12/2023 17:57	400 E 83RD ST	RESTAURANT	CHATHAM	HOMICIDE	20-29	M	BLK	YES

So, we must edit and change both the mapper_hw5.py and reducer_hw5.py scripts to use the Task 2 results. I tried the following codes (but had no success):

Experimental mapper_hw5.py script for Task 3:

```
mapper_hw5.py x reducer_hw5.py
Users > deleonv > Desktop > mapper_hw5.py > ...
1  #!/usr/bin/env python3
2
3  import sys
4
5  for line in sys.stdin: # reading line by line
6      line = line.strip() # removing whitespaces from input line
7      cols = line.split("\t") # let's first split the input line by the tab characters so we could separate the two datasets
8
9      sort_key = cols[1].split(",")[0] # extracting DATE column from joined results (data2.csv portion)
10     print(sort_key + "\t" + line) # DATE will be the key and the entire line will be the value
```


For Task 3, I will be using DATE column to sort the joined results based on the date from the part that originates from data2.csv as shown above “2/10/2021 19:53”. This script will print the DATE as key and the entire line as the value.

Experimental reducer_hw5.py (I initially started trying sorting in descending order) script for Task 3:

```
mapper_hw5.py  reducer_hw5.py x
Users > deleenv > Desktop > reducer_hw5.py > ...
1  #!/usr/bin/env python3
2
3  import sys
4
5  # Descending order:
6  lines = [line.strip() for line in sys.stdin] # reading lines, get ride of whitespaces, and store values in a list
7  lines.sort(reverse = True) # sort list in descending order (DATE column from Task 2 results)
8  for line in lines:
9      _, value = line.split("\t", 1) # tab separation
10     print(value)
```

In this case, the reducer reads the key-value pairs given by the mapper, sorts them in descending order by the DATE key, and then prints the lines.

Finally, I was facing RAM issues regarding Task 3, which changed everything:

```
2023-10-16 19:48:59,456 INFO mapreduce.Job: map 100% reduce 77%
2023-10-16 19:49:11,520 INFO mapreduce.Job: map 100% reduce 78%
2023-10-16 19:49:23,591 INFO mapreduce.Job: map 100% reduce 79%
2023-10-16 19:49:33,713 INFO mapreduce.Job: Task Id : attempt_1697492660248_0002_r_000000_0, Status : FAILED
Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 137
    at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:326)
    at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:539)
    at org.apache.hadoop.streaming.PipeReducer.reduce(PipeReducer.java:128)
    at org.apache.hadoop.mapred.ReduceTask.runOldReducer(ReduceTask.java:445)
    at org.apache.hadoop.mapred.ReduceTask.run(ReduceTask.java:393)
    at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:174)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1762)
    at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:168)

2023-10-16 19:49:35,004 INFO mapreduce.Job: map 100% reduce 0%
2023-10-16 19:49:54,216 INFO mapreduce.Job: map 100% reduce 4%
2023-10-16 19:49:59,745 INFO mapreduce.Job: map 100% reduce 5%
2023-10-16 19:50:05,804 INFO mapreduce.Job: map 100% reduce 7%
```

Since I am running Hadoop locally, I decided to delete both data1.csv and data2.csv from HDFS directory and just have the output results on it:

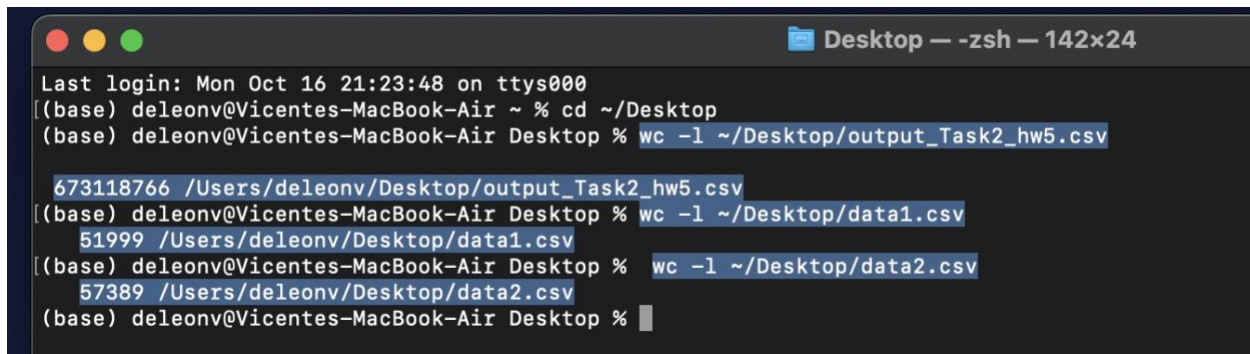
```
(base) deleenv@Vicentes-MacBook-Air Desktop % hdfs dfs -ls /assignment_data/
2023-10-16 20:11:22,579 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
drwxr-xr-x  - deleenv supergroup          0 2023-10-12 22:13 /assignment_data/output_Task2_hw5
(base) deleenv@Vicentes-MacBook-Air Desktop %
```

This is the code used to run the MapReduce Job from Task 3. The stored results will be named “output_Task3_hw”.

```
63  hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar \
64  -files mapper_hw5.py,reducer_hw5.py \
65  -mapper mapper_hw5.py \
66  -reducer reducer_hw5.py \
67  -input /assignment_data/output_Task2_hw5/part-00000 \
68  -output /assignment_data/output_Task3_hw5
```

Even though I tried deleting data1.csv and data2.csv from assignment_data HDFS directory, I was still facing RAM issues. I went ahead using “wc” (word count MAC command) just to see how many lines each csv had:

WC command: [https://www.oreilly.com/library/view/macintosh-terminal-pocket/9781449328962/ch01s01s02.html#:~:text=The%20program%20name%20\(%20wc%20%2C%20the,count%20lines%20and%20not%20words.](https://www.oreilly.com/library/view/macintosh-terminal-pocket/9781449328962/ch01s01s02.html#:~:text=The%20program%20name%20(%20wc%20%2C%20the,count%20lines%20and%20not%20words.)

A terminal window titled "Desktop - zsh - 142x24" showing the execution of the 'wc' command on three CSV files. The first command shows 'output_Task2_hw5.csv' has 673,118,766 lines. The second command shows 'data1.csv' has 51,999 lines. The third command shows 'data2.csv' has 57,389 lines. The terminal text is as follows:

```
Last login: Mon Oct 16 21:23:48 on ttys000
(base) deleonv@Vicentes-MacBook-Air ~ % cd ~/Desktop
(base) deleonv@Vicentes-MacBook-Air Desktop % wc -l ~/Desktop/output_Task2_hw5.csv
673118766 /Users/deleonv/Desktop/output_Task2_hw5.csv
(base) deleonv@Vicentes-MacBook-Air Desktop % wc -l ~/Desktop/data1.csv
51999 /Users/deleonv/Desktop/data1.csv
(base) deleonv@Vicentes-MacBook-Air Desktop % wc -l ~/Desktop/data2.csv
57389 /Users/deleonv/Desktop/data2.csv
(base) deleonv@Vicentes-MacBook-Air Desktop %
```

The above image just shows that “output_Task2_hw5.csv” has 673 million lines. Data1.csv and data2.csv each has approx. 52k and 57k lines respectively. 673 million lines is extremely big for Hadoop MapReduce job in local MAC. Since I am focusing on VICTIMIZATION_PRIMARY column, my script logic for Task 2 produces an output row for every combination of rows from data1 and data 2 that share the same VICTIMIZATION_PRIMARY value.

Cross Join/Cartesian Product: [https://www.listendata.com/2014/06/proc-sql-merging.html#:~:text=Cross%20Join%20%2F%20Cartesian%20product,merged%20table%20\(data%20set\).](https://www.listendata.com/2014/06/proc-sql-merging.html#:~:text=Cross%20Join%20%2F%20Cartesian%20product,merged%20table%20(data%20set).)

(Data1 VICTIMIZATION_PRIMARY rows) x (Data 2 VICTIMIZATION_PRIMARY rows) = output_Task2_hw5 results.

Data1 (x rows) * data2 (y rows) = output (x rows*y rows). It seems my output results from Task 2 suffered from the Cross Join/ Cartesian Product scenario. I think I will need to go back and modify my python scripts to limit the number lines of the “output_Task2_hw5” results. I think this might be a possible solution to Task3. It seems this is standard for situations where multiple rows in one dataset match multiple rows in another dataset on the join key.

1. Cross Join / Cartesian product

The Cartesian product returns a number of rows equal to the product of all rows (observations) in all the tables (data sets) being joined. For example, if the first table has 10 rows and the second table has 10 rows, there will be 100 rows (10 * 10) in the merged table (data set).

Solution to Task 3:

After multiple failed attempts of trying to alter and change my Python scripts for mapper and reducer to tackle Task 3, I managed to find a possible solution just for this task. I had to brute force this solution due to the size of the “output_Task2_hw5” results from Task 2. I had the idea to just work with a subset of the data instead of working with the entire results dataset. To avoid more RAM issues due to the Cartesian product scenario I explained earlier, I had the idea to work with a 2 million rows dataset (just like Kindle_Store.json file data size I used for Homework 4 MapReduce job).

```
61 Task3)
62 My scripts are not working due to the size of the data 673 million lines due to Cartesian Product. I will need to brute for this issue:
63 head -n 2000000 ~/Desktop/output_Task2_hw5.csv > ~/Desktop/smaller_output_Task2_hw5.csv
64
65 type the following to view heavy csv files using MAC Terminal: head -n 10 ~/Desktop/smaller_output_Task2_hw5.csv
```

I added back both datasets to HDFS directory data1.csv and data2.csv as well as the “output_Task2_hw5” subset data called “smaller_output_Task2_hw5” (which will also be in final submission) containing 2 million rows instead of 673 million rows:

```
((base) deleonv@Vicentes-MacBook-Air Desktop % hdfs dfs -ls /assignment_data/
2023-10-17 10:24:34,947 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 3 items
-rw-r--r-- 1 deleonv supergroup 1881056 2023-10-17 10:22 /assignment_data/data1.csv
-rw-r--r-- 1 deleonv supergroup 4616484 2023-10-17 10:23 /assignment_data/data2.csv
-rw-r--r-- 1 deleonv supergroup 227495426 2023-10-17 10:24 /assignment_data/smaller_output_Task2_hw5.csv
(base) deleonv@Vicentes-MacBook-Air Desktop %
```

Also, I decided to create 2 new mapper and reducer scripts to tackle the secondary sorting. This time, the sorting will be in **ascending order**:

Ascending order is by default: <http://www.hadooplessons.info/2017/08/order-by-clause-in-apache-hive.html#:~:text=We%20can%20arrange%20the%20column,order%20is%20the%20default%20order.>

Sorting_mapper.py:

```
Users > deleonv > Desktop > sorting_mapper.py > ...
1  #!/usr/bin/env python3
2
3  import sys
4
5  for line in sys.stdin: # reading line by line
6      part = line.strip().split("\t") # removing white spaces and splitting input line by tab
7      if len(part) == 2: # just making sure the above code was properly split into two main parts
8          # Extract date from Dataset 2 and use it as a key for sorting
9          date = part[1].split(",")[0] # extracting DATE from dataset 2 and use it as key
10         print(date + "\t" + line.strip())
```

The new script prepares data for sorting by extracting a key (DATE from data2.csv part) that Hadoop can use for ascending sorting purposes. This code prints out the extracted DATE (key) alongside the original input line (value).

Sorting_reducer.py:

```
Users > delevnv > Desktop > sorting_reducer.py > ...
1  #!/usr/bin/env python3
2
3  import sys
4
5  for line in sys.stdin: # reading line by line
6      # date -> first value, full_line -> remaining line value
7      date, full_line = line.strip().split("\t", 1) # removing white spaces and splitting input line in two parts as before based on tab
8      part = full_line.split("\t") # splitting full_line into parts using tab
9      if len(part) == 2: # just making sure the above code was properly split into two main parts
10         dataset1, dataset2 = part # extracting dataset
11         print(dataset1 + "\t" + dataset2)
```

The purpose of this new reducer is to remove the sorting key (DATE) that was added in the mapper section, restoring the original joined dataset format. It prints out the two datasets separated by tab.

Commands:

```
67 type: cd ~/Desktop
68 type: chmod +x sorting_mapper.py (to make the script executable)
69 type: chmod +x sorting_reducer.py (to make the script executable)
70 type to check if they exist: ls -l sorting_mapper.py sorting_reducer.py
71
72
73 2)
74 hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar \
75 -files sorting_mapper.py,sorting_reducer.py \
76 -mapper sorting_mapper.py \
77 -reducer sorting_reducer.py \
78 -input /assignment_data/smaller_output_Task2_hw5.csv \
79 -output /assignment_data/output_Task3_hw5
80
81 type the following if you have to delete HDFS output: hdfs dfs -rm -r /assignment_data/output_Task3_hw5
82
83 let's display entire content of output directory:
84 type: hdfs dfs -ls /assignment_data/output_Task3_hw5
85
86 type to see part-00000 content: type: hdfs dfs -cat /assignment_data/output_Task3_hw5/part-00000
87 type to export data into csv file: hdfs dfs -get /assignment_data/output_Task3_hw5/part-00000 ~/Desktop/output_Task3_hw5.csv
88 type the following to view heavy csv files using MAC Terminal: head -n 10 ~/Desktop/output_Task3_hw5.csv
89
90
91 type the following to view heavy csv files using MAC Terminal: head -n 10 ~/Desktop/output_Task3_hw5.csv
```

The above command image shows how I made the new mapper and reducer executable within my MAC terminal. The MapReduce job shows how I used “smaller_output_Task2_hw5.csv” and the output results were “output_Task3_hw5”. Finally, I decided to store Task 3 results in csv file format which will be in final submission.

Results:

```
(base) delevnv@Vicentes-MacBook-Air Desktop % hdfs dfs -get /assignment_data/output_Task3_hw5/part-00000 ~/Desktop/output_Task3_hw5.csv
2023-10-17 17:16:37, 847 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
(base) delevnv@Vicentes-MacBook-Air Desktop % head -n 10 ~/Desktop/output_Task3_hw5.csv
9/30/2023, BATTERY, 70-79, M, UNKNOWN, 1, 1/1/2010 0:12, 7800 S WOOD ST, ALLEY, AUBURN GRESHAM, BATTERY, 20-29, M, BLK, YES
9/30/2023, BATTERY, 80+, F, WWH, 3, 1/1/2010 0:12, 7800 S WOOD ST, ALLEY, AUBURN GRESHAM, BATTERY, 20-29, M, BLK, YES
6/30/2023, BATTERY, 40-49, F, WWH, 8, 1/1/2010 0:12, 7800 S WOOD ST, ALLEY, AUBURN GRESHAM, BATTERY, 20-29, M, BLK, YES
9/30/2022, BATTERY, 20-29, M, BLK, 55, 1/1/2010 0:12, 7800 S WOOD ST, ALLEY, AUBURN GRESHAM, BATTERY, 20-29, M, BLK, YES
9/30/2023, BATTERY, 30-39, UNKNOWN, UNKNOWN, 2, 1/1/2010 0:12, 7800 S WOOD ST, ALLEY, AUBURN GRESHAM, BATTERY, 20-29, M, BLK, YES
9/30/2023, BATTERY, 40-49, F, BLK, 24, 1/1/2010 0:12, 7800 S WOOD ST, ALLEY, AUBURN GRESHAM, BATTERY, 20-29, M, BLK, YES
6/30/2023, BATTERY, 60-69, F, WWH, 2, 1/1/2010 0:12, 7800 S WOOD ST, ALLEY, AUBURN GRESHAM, BATTERY, 20-29, M, BLK, YES
9/30/2023, BATTERY, 50-59, M, WWH, 13, 1/1/2010 0:12, 7800 S WOOD ST, ALLEY, AUBURN GRESHAM, BATTERY, 20-29, M, BLK, YES
9/30/2023, BATTERY, 80+, M, WWH, 1, 1/1/2010 0:12, 7800 S WOOD ST, ALLEY, AUBURN GRESHAM, BATTERY, 20-29, M, BLK, YES
9/30/2022, BATTERY, 20-29, M, BLK, 235, 1/1/2010 0:12, 7800 S WOOD ST, ALLEY, AUBURN GRESHAM, BATTERY, 20-29, M, BLK, YES
(base) delevnv@Vicentes-MacBook-Air Desktop %
```

Comparing Results Task 2 vs Task 3:

Smaller_output_Task2_hw5.csv:

```
(base) deleonv@Vicentes-MacBook-Air Desktop % head -n 10 ~/Desktop/smaller_output_Task2_hw5.csv
9/30/2023,BATTERY,0-19,F,BLK,25 2/10/2021 19:53,3900 W MADISON ST,PARKING LOT / GARAGE (NON RESIDENTIAL),WEST GARFIELD PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 12/29/2011 1:05,1600 N LOTUS AVE,STREET,AUSTIN,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 12/28/2011 18:25,9400 S JUSTINE ST,SIDEWALK,WASHINGTON HEIGHTS,BATTERY,0-19,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/10/2021 18:33,400 N DRAKE AVE,STREET,HUMBOLDT PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 12/28/2011 15:00,11300 S VINCENNES AVE,STREET,MORGAN PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/9/2021 23:15,700 N HAMLIN AVE,ALLEY,HUMBOLDT PARK,BATTERY,30-39,M,WWH,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/8/2021 5:24,4100 W ADAMS ST,SIDEWALK,WEST GARFIELD PARK,BATTERY,30-39,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/8/2021 5:24,4100 W ADAMS ST,SIDEWALK,WEST GARFIELD PARK,BATTERY,40-49,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 6/20/2020 2:26,2700 N PINE GROVE AVE,APARTMENT,LINCOLN PARK,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,0-19,F,BLK,25 2/7/2021 23:53,7800 S STATE ST,RESIDENCE,GREATER GRAND CROSSING,BATTERY,50-59,M,BLK,YES
```

Output_Task3_hw5.csv:

```
((base) deleonv@Vicentes-MacBook-Air Desktop % head -n 10 ~/Desktop/output_Task3_hw5.csv
9/30/2023,BATTERY,70-79,M,UNKNOWN,1 1/1/2010 0:12,7800 S WOOD ST,ALLEY,AUBURN GRESHAM,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,80+,F,WWH,3 1/1/2010 0:12,7800 S WOOD ST,ALLEY,AUBURN GRESHAM,BATTERY,20-29,M,BLK,YES
6/30/2023,BATTERY,40-49,F,WHI,8 1/1/2010 0:12,7800 S WOOD ST,ALLEY,AUBURN GRESHAM,BATTERY,20-29,M,BLK,YES
9/30/2022,BATTERY,20-29,M,BLK,55 1/1/2010 0:12,7800 S WOOD ST,ALLEY,AUBURN GRESHAM,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,30-39,UNKNOWN,UNKNOWN,2 1/1/2010 0:12,7800 S WOOD ST,ALLEY,AUBURN GRESHAM,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,40-49,F,BLK,24 1/1/2010 0:12,7800 S WOOD ST,ALLEY,AUBURN GRESHAM,BATTERY,20-29,M,BLK,YES
6/30/2023,BATTERY,60-69,F,WWH,2 1/1/2010 0:12,7800 S WOOD ST,ALLEY,AUBURN GRESHAM,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,50-59,M,WWH,13 1/1/2010 0:12,7800 S WOOD ST,ALLEY,AUBURN GRESHAM,BATTERY,20-29,M,BLK,YES
9/30/2023,BATTERY,80+,M,WHI,1 1/1/2010 0:12,7800 S WOOD ST,ALLEY,AUBURN GRESHAM,BATTERY,20-29,M,BLK,YES
9/30/2022,BATTERY,20-29,M,BLK,235 1/1/2010 0:12,7800 S WOOD ST,ALLEY,AUBURN GRESHAM,BATTERY,20-29,M,BLK,YES
```

You can see that my sorting_mapper.py and sorting_reducer.py logic worked, and the desired output was returned. The above images show the before and after the secondary sorting by DATE (DATE column from data2.csv). I also inspected both csv and both files contain 2 million rows.

```
((base) deleonv@Vicentes-MacBook-Air Desktop % wc -l ~/Desktop/smaller_output_Task2_hw5.csv
2000000 /Users/deleonv/Desktop/smaller_output_Task2_hw5.csv
(base) deleonv@Vicentes-MacBook-Air Desktop % wc -l ~/Desktop/output_Task3_hw5.csv
2000000 /Users/deleonv/Desktop/output_Task3_hw5.csv
```


Task 4: Testing with MRUnit (Python – unittest Module)

MRUnit is a Java library and testing framework designed specifically for testing MapReduce programs. It provides a simple and convenient way to write unit tests for MapReduce programs, ensuring their correctness and reliability in data processing and analytics workflows.

I wrote all my MapReduce job scripts using Python and not Java. It looks like MRUnit is only designed for Java-based MapReduce programs. I will try and use Python's "unittest" framework to tackle Task 4 testing scenario. There are also other unit testing frameworks such as "doctest and PyTest" that make it easy to write and execute this kind of testing. Unittest includes features like test discovery, text fixtures, test suites, and assertion methods to compare expected and actual results. Also, it supports test automation and allows developers to write cases using classes and methods.

- Test methods are named with prefix "test_" to be discovered by the testing framework.

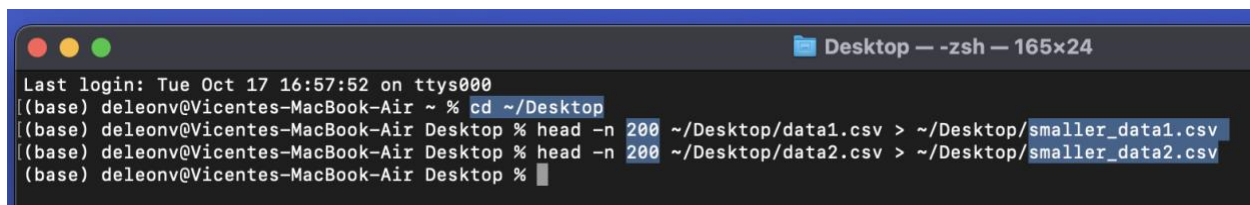
However, after reading I notice that MRUnit will test in a Hadoop like environment (simulating an environment close to the actual Hadoop). In contrast, Python's "unittest" focuses on python script logic instead of simulating a Hadoop environment.

At last, how to do unit testing?

```
1 import unittest
2 import wc_mapper
3
4 class WCMapperTest(unittest.TestCase):
5     def test_mapper(self):
6         input = 'mapper input of wc'
7         expected = ['mapper\t1',
8                     'input\t1',
9                     'of\t1',
10                    'wc\t1']
11         result = []
12         for output in wc_mapper.mapper(input):
13             result.append(output)
14         self.assertEqual(expected, result)
15
16 if __name__ == '__main__':
17     unittest.main()
```

```
class MyTestCase(unittest.TestCase):
    def test_function_name(self):
        result = my_function() # Call the function being tested
        self.assertEqual(result, expected_result)
```

To avoid RAM issues or any other issues regarding data complexity, I will work on Task 4 using a subset of both datasets and the joined results just like I did for Task 3 to ensure if script logic is correct. This is my first-time doing Unit Testing, so I don't want the test cases to fail due to memory issues.



```
Desktop — zsh — 165x24
Last login: Tue Oct 17 16:57:52 on ttys000
(base) deleonv@Vicentes-MacBook-Air ~ % cd ~/Desktop
(base) deleonv@Vicentes-MacBook-Air Desktop % head -n 200 ~/Desktop/data1.csv > ~/Desktop/smaller_data1.csv
(base) deleonv@Vicentes-MacBook-Air Desktop % head -n 200 ~/Desktop/data2.csv > ~/Desktop/smaller_data2.csv
(base) deleonv@Vicentes-MacBook-Air Desktop %
```

Mapper and Reducer within unittesting.py (unittesting.py):

These new functions within the Unit testing program will mimic mapper and reducer MapReduce Jobs, so we can test their logic on 4 simple test cases. Both functions will use the StringIO Module from Python, which is an in-memory file-like object. This object can be used as input or output to the most function that would expect a standard file object. When StringIO object is created it is initialized by passing a string to the constructor. If no string is passed the StringIO will start empty. In both cases, the initial cursor on the file starts at zero. StringIO will help by allowing to treat the string as a file-like object (stdin and stdout).

So, both functions begin with reading the string input. We set up output to StringIO() to capture the output returned by the functions and we store the result into output variable. Anything printed will be captured into this object instead of being printed out. At the end of both functions, we use.getvalue() to the stdout like a string (return entire file content). We just add the original mapper_hw5.py and reducer_hw5.py codes into the functions and it should be good to go.

```
6 def mapper_hw5(input): # mapper_hw5.py based function
7     sys.stdin = StringIO(input) # reading string input
8     output = StringIO() # setting output to StringIO to get mapper_hw5() output
9     sys.stdout = output # storing result into "output"
10
11     # mapper_hw5.py code:
12     for line in sys.stdin:
13         line = line.strip()
14         cols = line.split(",")
15
16         if len(cols) == 6:
17             key = cols[1] # VICTIMIZATION_PRIMARY Data1
18             value = "smaller_data1.csv," + ",".join(cols) # change to smaller data
19         else:
20             key = cols[4] # VICTIMIZATION_PRIMARY Data2
21             value = "smaller_data2.csv," + ",".join(cols) # change to smaller data
22             print(key + "\t" + value)
23
24     return output.getvalue() # retrieve entire content of the file
```

```
27 def reducer_hw5(input): # reducer_hw5.py based function
28     sys.stdin = StringIO(input) # reading string input
29     output = StringIO() # setting output to StringIO to get reducer_hw5() output
30     sys.stdout = output # storing result into "output"
31
32     # reducer_hw5.py code:
33     current_key = None
34     data1_rows = []
35     data2_rows = []
36
37     for line in sys.stdin:
38         line = line.strip()
39         key, value = line.split("\t", 1)
40
41         if key != current_key:
42             if current_key is not None:
43                 for row1 in data1_rows:
44                     for row2 in data2_rows:
45                         print(row1 + "\t" + row2)
46
47                 # Let's reset.
48                 current_key = key
49                 data1_rows = []
50                 data2_rows = []
51
52         dataset, row = value.split(",", 1)
53         if dataset == "smaller_data1.csv": # smaller data1
54             data1_rows.append(row)
55         elif dataset == "smaller_data2.csv": # smaller data2
56             data2_rows.append(row)
57
58     # Repeat process
59     if current_key is not None:
60         for row1 in data1_rows:
61             for row2 in data2_rows:
62                 print(row1 + "\t" + row2)
63
64     return output.getvalue() # retrieve entire content of the file
```

It is important to remember that I am using subset data “smaller_data1.csv” and “smaller_data2.csv” for this task. Now, let’s move into the class and its test cases. We have two test cases for the mapper and two test cases for the reducer. We are going to be using “assertEqual(a, b)” for all of them. This condition checks if a and b are equal (although we are going to be using result vs expected and not a or b). The test cases were straightforward and simple.

Mapper test cases: we are extracting value from VICTIMIZATION_PRIMARY column (second column in smaller_data1.csv and fifth column in smaller_data2.csv) and returning the name of the dataset data1 vs data2 plus the original line as the value (using comma as separator). The expected variable contains the expected output for the input. Finally, self.assertEqual(result, expected) checks if the output of the functions matches the expected result. Also, the mapper_hw5() every line ends with a print statement. Print statements in Python append “\n” at the end by default. Reason why it was included in the expected variable.

```
# Checking for mapper_hw5.py logic using data1.csv
# extract the value from VICTIMIZATION_PRIMARY as the key
# return the name of "dataset1.csv" AND the original line as the value using comma as a separator
def test_mapper_hw5_data1(self): # test case 1
    line_data1 = "6/30/2003,CRIMINAL SEXUAL ASSAULT,0-19,F,WmH,42"
    result = mapper_hw5(line_data1)
    expected = "CRIMINAL SEXUAL ASSAULT\tsmaller_data1.csv,6/30/2003,CRIMINAL SEXUAL ASSAULT,0-19,F,WmH,42\n"
    self.assertEqual(result, expected) # result must match expected

# Checking for mapper_hw5.py logic using data2.csv
# extract the value from VICTIMIZATION_PRIMARY as the key
# return the name of "dataset2.csv" AND the original line as the value using comma as a separator
def test_mapper_hw5_data2(self): # test case 2
    line_data2 = "2/12/2023 17:57,400 E 83RD ST,RESTAURANT,CHATHAM,HOMICIDE,20-29,M,BLK,YES"
    result = mapper_hw5(line_data2)
    expected = "HOMICIDE\tsmaller_data2.csv,2/12/2023 17:57,400 E 83RD ST,RESTAURANT,CHATHAM,HOMICIDE,20-29,M,BLK,YES\n"
    self.assertEqual(result, expected) # result must match expected
```

I personally feel the reducer test cases are much simpler. Test_reducer_hw5() takes on the lines from data1 and data2 (both lines share the common key BATTERY). The mapper_hw5_output variable contains the combined output from running both lines through the mapper_hw5() function and this combined output is then passed to reducer_hw5() (just like Task 2). The expected variable contains the result we are looking for, which is the combination of both input lines separated by tab (“\t”) indication a join operation. The self.assertEqual() just checks and compares the actual result vs expected result.

The last test case, test_reducer_hw5_non_matching(), checks the behavior of the reducer_hw5() function when provided with non-matching join results. In this case we have BATTERY vs HOMICIDE as non-matching keys. The expected variable contains empty string, because this join operation does not exist due to the non-matching keys mentioned above. Finally, assertEquals() does its job.

```
def test_reducer_hw5(self): # test case 3 -> testing for join results
    line_data1 = "9/30/2023,BATTERY,0-19,F,BLK,25"
    line_data2 = "4/8/2020 14:40,5400 W HIRSCH ST,STREET,AUSTIN,BATTERY,20-29,M,BLK,YES"

    mapper_hw5_output = mapper_hw5(line_data1) + mapper_hw5(line_data2)
    result = reducer_hw5(mapper_hw5_output)
    expected = "9/30/2023,BATTERY,0-19,F,BLK,25\t4/8/2020 14:40,5400 W HIRSCH ST,STREET,AUSTIN,BATTERY,20-29,M,BLK,YES\n" # this is the result mapper_hw5.py and reducer_hw5.py returns
    self.assertEqual(result, expected) # result must match expected

def test_reducer_hw5_non_matching(self): # test case 4 -> testing for non matching results
    line_data1 = "3/31/2011,HOMICIDE,30-39,M,BLK,1"
    line_data2 = "1/13/2018 1:25,900 W 114TH PL,RESIDENCE,MORGAN PARK,BATTERY,0-19,M,BLK,YES"

    mapper_hw5_output = mapper_hw5(line_data1) + mapper_hw5(line_data2)
    result = reducer_hw5(mapper_hw5_output)
    expected = "" # no output because line_data1 and line_data 2 don't match (it has to PASS)
    self.assertEqual(result, expected) # result must match expected
```

Task 4 Results:

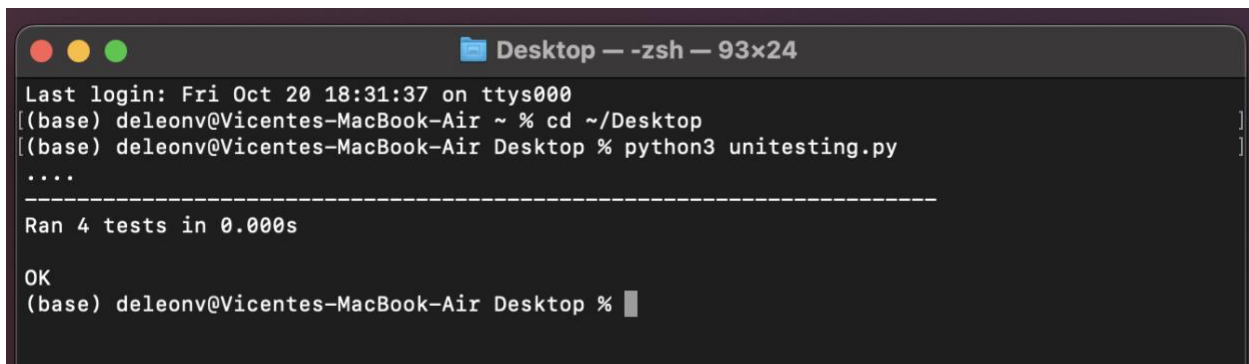
ScriptTestCase(unittest.TestCase):

```
66 class ScriptTestCase(unittest.TestCase):
67
68     # Checking for mapper_hw5.py logic using data1.csv
69     # extract the value from VICTIMIZATION_PRIMARY as the key
70     # return the name of "dataset1.csv" AND the original line as the value using comma as a separator
71     def test_mapper_hw5_data1(self): # test case 1
72         line_data1 = "6/30/2003,CRIMINAL SEXUAL ASSAULT,0-19,F,W,M,42"
73         result = mapper_hw5(line_data1)
74         expected = "CRIMINAL SEXUAL ASSAULT\\tsmaller_data1.csv,6/30/2003,CRIMINAL SEXUAL ASSAULT,0-19,F,W,M,42\\n"
75         self.assertEqual(result, expected) # result must match expected
76
77     # Checking for mapper_hw5.py logic using data2.csv
78     # extract the value from VICTIMIZATION_PRIMARY as the key
79     # return the name of "dataset2.csv" AND the original line as the value using comma as a separator
80     def test_mapper_hw5_data2(self): # test case 2
81         line_data2 = "2/12/2023 17:57,400 E 83RD ST,RESTAURANT,CHATHAM,HOMICIDE,20-29,M,BLK,YES"
82         result = mapper_hw5(line_data2)
83         expected = "HOMICIDE\\tsmaller_data2.csv,2/12/2023 17:57,400 E 83RD ST,RESTAURANT,CHATHAM,HOMICIDE,20-29,M,BLK,YES\\n"
84         self.assertEqual(result, expected) # result must match expected
85
86
87     def test_reducer_hw5(self): # test case 3 -> testing for join results
88         line_data1 = "9/30/2023,BATTERY,0-19,F,BLK,25"
89         line_data2 = "4/8/2020 14:40,5400 W HIRSCH ST,STREET,AUSTIN,BATTERY,20-29,M,BLK,YES"
90
91         mapper_hw5_output = mapper_hw5(line_data1) + mapper_hw5(line_data2)
92         result = reducer_hw5(mapper_hw5_output)
93         expected = "9/30/2023,BATTERY,0-19,F,BLK,25\\t4/8/2020 14:40,5400 W HIRSCH ST,STREET,AUSTIN,BATTERY,20-29,M,BLK,YES\\n" # this is the result mapper_hw5.py and reducer_hw5.py returns
94         self.assertEqual(result, expected) # result must match expected
95
96
97     def test_reducer_hw5_non_matching(self): # test case 4 -> testing for non matching results
98         line_data1 = "3/31/2011,HOMICIDE,30-39,M,BLK,1"
99         line_data2 = "1/13/2018 1:25,900 W 114TH PL,RESIDENCE,MORGAN PARK,BATTERY,0-19,M,BLK,YES"
100
101         mapper_hw5_output = mapper_hw5(line_data1) + mapper_hw5(line_data2)
102         result = reducer_hw5(mapper_hw5_output)
103         expected = "" # no output because line_data1 and line_data 2 don't match (it has to PASS)
104         self.assertEqual(result, expected) # result must match expected
105
106 if __name__ == "__main__":
107     unittest.main()
```

MAC Commands (both commands should give you the right answer):

```
101 After having complete scripts for Task 4 please run (on Desktop):
102 type: cd ~/Desktop
103 type: python3 unittesting.py
104
105 OR
106
107 type: python3 -m unittest unittesting.py
```

Results (All four test cases successfully pass):



```
Desktop — -zsh — 93x24
Last login: Fri Oct 20 18:31:37 on ttys000
[(base) deleonv@Vicentes-MacBook-Air ~ % cd ~/Desktop
[(base) deleonv@Vicentes-MacBook-Air Desktop % python3 unittesting.py
....
-----
Ran 4 tests in 0.000s

OK
(base) deleonv@Vicentes-MacBook-Air Desktop %
```

Sources:

Python MapReduce Jobs: <https://www.pavantestingtools.com/2018/05/how-to-test-python-mapreduce-jobs-in.html>

Python MapReduce Jobs and Testing: <https://blog.zhengdong.me/2012/07/30/streaming-python-unit-testing/>

Sorting Python: <https://www.geeksforgeeks.org/python-list-sort-method/>

MapReduce join tutorial: <https://blog.matthewrathbone.com/2016/02/09/python-tutorial.html>

MapReduce join tutorial: <https://medium.com/@aw.shubh/join-algorithm-using-map-reduce-941f3437b483>

Mapper.py base: <https://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/>

Stack Overflow: <https://stackoverflow.com/questions/25720178/input-split-for-map-function-in-hadoop>

Python Join: [https://www.mygreatlearning.com/blog/join-in-python/#:~:text=In%20Python%2C%20the%20join\(\),specified%20string%20as%20a%20separator.](https://www.mygreatlearning.com/blog/join-in-python/#:~:text=In%20Python%2C%20the%20join(),specified%20string%20as%20a%20separator.)

MapReduce job: https://www.tutorialspoint.com/map_reduce/map_reduce_quick_guide.htm

Key Value pair: <https://techvidvan.com/tutorials/hadoop-mapreduce-key-value-pair/>

Split() Python: <https://stackoverflow.com/questions/21462879/in-line-split-1-what-does-the-1-in-the-square-brackets-indicate-in-py>

Split() Python: <https://www.geeksforgeeks.org/python-string-split/>

Python Append(): <https://www.geeksforgeeks.org/python-list-append-method/>

Python is not None syntax: <https://www.w3docs.com/snippets/python/python-if-x-is-not-none-or-if-not-x-is-none.html>

Hadoop Commands: https://sparkbyexamples.com/apache-hadoop/hadoop-hdfs-dfs-commands-and-starting-hdfs-dfs-services/?expand_article=1

Hadoop Set Up: https://www.tutorialspoint.com/hadoop/hadoop_environment_setup.htm

MapReduce job commands base: <https://hadoop.apache.org/docs/stable/hadoop-streaming/HadoopStreaming.html>

MapReduce job: <https://hadoop.apache.org/docs/r1.2.1/streaming.html>

MRUnit: <https://www.dremio.com/wiki/apache-mrunit/#:~:text=Apache%20MRUnit%20is%20a%20Java,data%20processing%20and%20analytics%20workflows.>

Python Unite Test: <https://www.browserstack.com/guide/unit-testing-python>

StringIO: <https://www.geeksforgeeks.org/stringio-module-in-python/>

Stack Overflow: <https://stackoverflow.com/questions/2654834/capturing-stdout-within-the-same-process-in-python/3113913>

Stack Overflow: <https://stackoverflow.com/questions/1218933/can-i-redirect-the-stdout-into-some-sort-of-string-buffer>

StringIO and getvalue(): <https://wrongsideofmemphis.com/2010/03/01/store-standard-output-on-a-variable-in-python/>

Unittest: <https://blog.zhengdong.me/2012/07/30/streaming-python-unit-testing/>