Zika virus data analysis using MapReduce

An outbreak of the Zika virus, an infection transmitted mostly by the Aedes species mosquito (Ae. aegypti and Ae. albopictus), has been sweeping across the Americas and the Pacific since mid-2015.

The dataset is a report on zika which contains the fields as

Column-1(report_date) - The report date is the date that the report was published. The date should be specified in standard ISO format (YYYY-MM-DD).

Column-2(location) - A location is specified for each observation following the specific names specified in the country place name database. This may be any place with a 'location_type' as listed below, e.g. city, state, country, etc. It should be specified at up to three hierarchical levels in the following format: [country]-[state/province]-[county/municipality/city], always beginning with the country name. If the data is for a particular city, e.g. Salvador, it should be specified: Brazil-Bahia-Salvador.

Column-3(location_type) - A location code is included indicating: city, district, municipality, county, state, province, or country. If there is need for an additional 'location_type', open an Issue to create a new 'location_type'.

Column-4(data_field) - The data field is a short description of what data is represented in the row and is related to a specific definition defined by the report from which it comes.

Column-5(data_field_code) - This code is defined in the country data guide. It includes a two letter country code (ISO-3166 alpha-2, list), followed by a 4-digit number corresponding to a specific report type and data type.

Column-6(time_period) - Optional. If the data pertains to a specific period of time, for example an epidemiological week, that number should be indicated here and the type of time period in the 'time_period_type', otherwise it should be NA.

Column-7(time_period_type)- Required only if 'time_period' is specified. Types will also be specified in the country data guide. Otherwise should be NA.

Column-8(value) - The observation indicated for the specific 'report_date', 'location', 'data_field' and when appropriate, 'time_period'.

Column-9(unit) - The unit of measurement for the 'data_field'. This should conform to the 'data_field' unit options as described in the country-specific data guide.

Problem Statements:

1) Which place has the most zika suspected cases?
2) On which date the most confirmed(any confirmed) cases are recorded?
3) What are the various types of cases?

1) Which place has the most zika suspected cases?
Answer :
zika_mapper.py

```
#!/usr/bin/python

import sys

for line in sys.stdin:
 splits = line.split(',')
 if len(splits[3]) and ('zika_suspected' in splits[3]):
  if splits[1].startswith('"'):
    splits[1] = splits[1][1:]
  if splits[1].endswith('"'):
    splits[1] = splits[1][:-1]
  if splits[7].startswith('"'):
    splits[7] = splits[7][1:]
  if splits[7].endswith('"'):
    splits[7] = splits[7][:-1]
  if ('NA' in splits[7]):
    splits[7] = 0
  splits[1] = str(splits[1])
  splits[7] = int(splits[7])
  if len(splits[1]) and (splits[7]):
   print'{0},{1}'.format(splits[1],splits[7])
```

zika_reducer.py

```
#!/usr/bin/python

import sys
a_dict = {}
counter = 0

for line in sys.stdin:
 location,value = line.rstrip('\n').split(',')
 location = str(location)
 value = int(value)
 if(counter == 0):
  a_dict[location] = value
  counter = counter + 1
 else:
  nh=[key for key in a_dict]
  if(location in nh):
```

```
  a_dict[location] = a_dict[location] + value
 else:
  a_dict[location] = value

maximum=max(a_dict,key=a_dict.get)
print(maximum,a_dict[maximum])
```

Command

```
cat cdc_zika.csv | sort | ./zika_mapper.py | ./zika_reducer.py
```

2) On which date the most confirmed(any confirmed) cases are recorded?

Answer :

zika_mapper.py

```
#!/usr/bin/python
import sys

for line in sys.stdin:

    line = line.strip()

    words = line.split(",")

    if(words[3].find("confirmed") == -1):
                pass
    else:
        print '%s\t%s' % (words[3],words[7])
```

zika_reducer.py

```
#!/usr/bin/python
from operator import itemgetter
import sys
current_word = None
current_count = 0
word = None

for line in sys.stdin:

    line = line.strip()
```

```
    word, count = line.split('\t', 1)

    try:
        count = int(count)
    except ValueError:

        continue

    if current_word == word:
        current_count =max(count, current_count)
    else:
        if current_word:

            print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word


if current_word == word:
    print '%s\t%s' % (current_word, current_count)
```

Command
cat cdc_zika.csv | sort | ./zika_mapper.py | ./zika_reducer.py

Cloudxlab command
ex: hadoop jar /usr/hdp/2.6.5.0-292/hadoop-mapreduce/hadoopstreaming.jar -input /user/support1161/cdc_zika.csv -output /user/support1161/war_cloudxlab_output -file /home/support1161/zika_mapper.py -file /home/support1161/zika_reducer.py -mapper /home/support1161/zika_mapper.py -reducer /home/support1161/zika_reducer.py

2) What are the various types of cases?
Answer :
zika_mapper.py

```
#!/usr/bin/python
import sys
```

```
for line in sys.stdin:
 splits = line.split(',')
 if splits[3].startswith(""):
    splits[3] = splits[3][1:]
 if splits[3].endswith(""):
    splits[3] = splits[3][:-1]
 print(splits[3],1)
```

zika_reducer.py

```
#!/usr/bin/python
import sys
a_dict = {}

for line in sys.stdin:
 case,value = line.split(',')
 a_dict[case] = value

for a in sorted(a_dict.keys()):
 print a
```

Command

```
cat cdc_zika.csv | sort | ./zika_mapper.py | ./zika_reducer.py
```