***EarthQuake.py***

import sys

from pyspark import SparkContext

if(len(sys.argv)!=6):

print("Provide Input File and Output Directory")

sys.exit(0)

sc =SparkContext()

f = sc.textFile(sys.argv[1])

# Region with Magnitude

temp=f.map(lambda x: (x.split(',')[11],float(x.split(',')[8])))

maxi=temp.reduceByKey(lambda a,b:a if a>b else b)

maxi.saveAsTextFile(sys.argv[2])

# Region with Depth

temp=f.map(lambda x: (x.split(',')[11],float(x.split(',')[9])))

maxi=temp.reduceByKey(lambda a,b:a if a>b else b)

maxi.saveAsTextFile(sys.argv[3])

# Region with latitude

temp=f.map(lambda x: (x.split(',')[11],float(x.split(',')[6])))

maxi=temp.reduceByKey(lambda a,b:a if a>b else b)

maxi.saveAsTextFile(sys.argv[4])

# Region with longitude

temp=f.map(lambda x: (x.split(',')[11],float(x.split(',')[7])))

maxi=temp.reduceByKey(lambda a,b:a if a>b else b)

maxi.saveAsTextFile(sys.argv[5])

***insurance.py***

import sys

from pyspark import SparkContext

if(len(sys.argv)!=4):

print("Provide Input File and Output Directory")

sys.exit(0)

sc =SparkContext()

f = sc.textFile(sys.argv[1])

# Total products

temp=f.map(lambda x: (x.split(',')[7],1))

temp=f.map(lambda x: (x.split(',')[16],1)) // sales

data=temp.countByKey()

dd=sc.parallelize(data.items())

dd.saveAsTextFile(sys.argv[2])

# Frequency

temp=f.map(lambda x: (x.split(',')[3],1))

temp=f.map(lambda x: (x.split(',')[2],1)) //sales

data=temp.countByKey()

dd=sc.parallelize(data.items())

dd.saveAsTextFile(sys.argv[3])

***weather.py***

import sys

if(len(sys.argv)!=4):

print("Provide Input File and Output Directory")

sys.exit(0)

from pyspark import SparkContext

sc =SparkContext()

f = sc.textFile(sys.argv[1])

temp=f.map(lambda x: (int(x[15:19]),int(x[87:92])))

mini=temp.reduceByKey(lambda a,b:a if a<b else b)

mini.saveAsTextFile(sys.argv[2])

maxi=temp.reduceByKey(lambda a,b:a if a>b else b)

maxi.saveAsTextFile(sys.argv[3])