# SPARK PROGRAMS

**Bash file:**

export JAVA\_HOME=$(readlink -f $(which javac) | awk 'BEGIN {FS="/bin"} {print $1}') if ! command -v spark-shell --version &> /dev/null

then

export PATH=$(echo $PATH):$(pwd)/bin

fi

Command to run it: **source bash.sh**

**Programs:**

## Write a spark to analyze the given weather report data and to generate a report with cities having maximum temperature for a particular year

import sys if(len(sys.argv)!=3):

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]))) maxi=temp.reduceByKey(lambda a,b:a if a>b else b) maxi.saveAsTextFile(sys.argv[2])

Output:

## Write a spark to analyze the given weather report data and to generate a report with cities having minimum temperature for a particular year

import sys if(len(sys.argv)!=3):

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])

Output:



## Write a spark program to analyze the given Earthquake data and generate statistics with region and magnitude

import sys if(len(sys.argv)!=3):

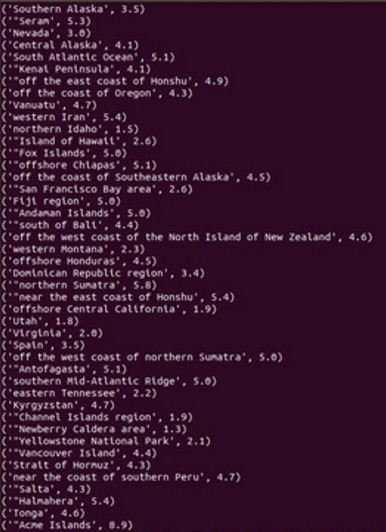
print("Provide Input File and Output Directory") sys.exit(0)

from pyspark import SparkContext sc =SparkContext()

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

temp=f.map(lambda x: (x.split(',')[11],float(x.split(',')[8]))) maxi=temp.reduceByKey(lambda a,b:a if a>b else b) maxi.saveAsTextFile(sys.argv[2])

Output:



## Write a spark program to analyze the given Earthquake data and generate statistics with region and depth

import sys

if(len(sys.argv)!=3):

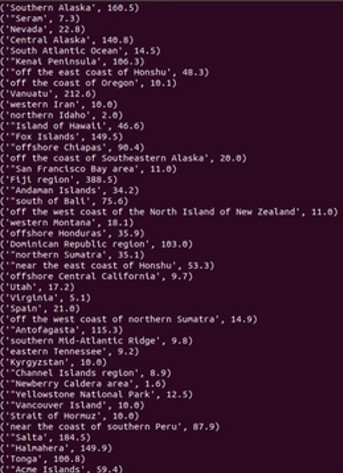
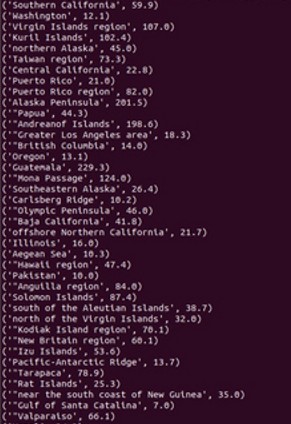
print("Provide Input File and Output Directory") sys.exit(0)

from pyspark import SparkContext sc =SparkContext()

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

temp=f.map(lambda x: (x.split(',')[11],float(x.split(',')[9]))) maxi=temp.reduceByKey(lambda a,b:a if a>b else b) maxi.saveAsTextFile(sys.argv[2])

Output:



## Write a spark program to analyze the given Earthquake data and generate statistics with region and latitude

import sys if(len(sys.argv)!=3):

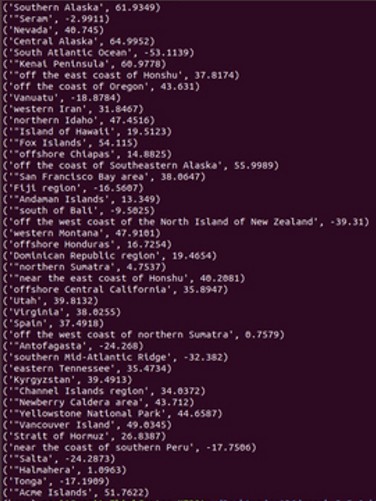
print("Provide Input File and Output Directory") sys.exit(0)

from pyspark import SparkContext sc =SparkContext()

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

temp=f.map(lambda x: (x.split(',')[11],float(x.split(',')[6]))) maxi=temp.reduceByKey(lambda a,b:a if a>b else b) maxi.saveAsTextFile(sys.argv[2])

Output:



## Write a spark program to analyze the given Earthquake data and generate statistics with region and longitude

import sys if(len(sys.argv)!=3):

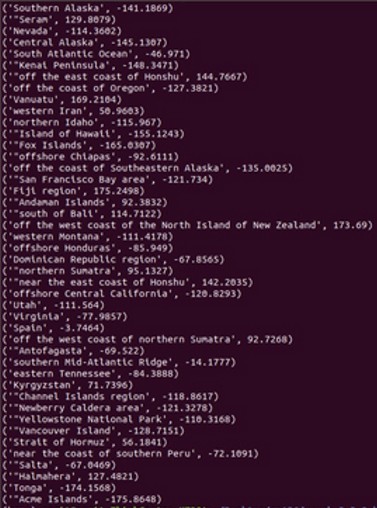
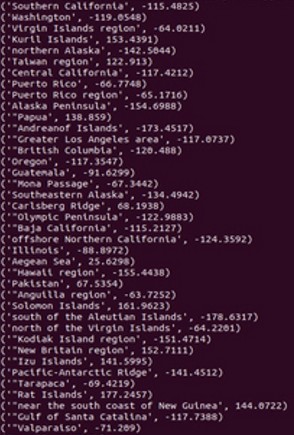
print("Provide Input File and Output Directory") sys.exit(0)

from pyspark import SparkContext sc =SparkContext()

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

temp=f.map(lambda x: (x.split(',')[11],float(x.split(',')[7]))) maxi=temp.reduceByKey(lambda a,b:a if a>b else b) maxi.saveAsTextFile(sys.argv[2])

Output:



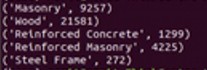
## Write a spark program to analyze the given Insurance data and generate a statistics report with the construction building name and the count of building.

import sys if(len(sys.argv)!=3):

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: (x.split(',')[16],1)) data=temp.countByKey() dd=sc.parallelize(data.items()) dd.saveAsTextFile(sys.argv[2])

Output:

## Write a spark program to analyze the given Insurance data and generate a statistics report with the county name and its frequency.

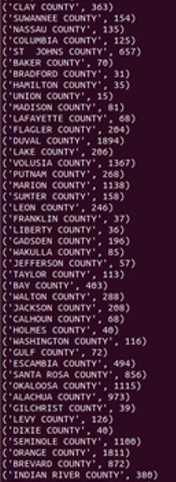
import sys if(len(sys.argv)!=3):

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: (x.split(',')[2],1)) data=temp.countByKey() dd=sc.parallelize(data.items()) dd.saveAsTextFile(sys.argv[2])

Output:



## Write a map-reduce program to analyze the given employee record data and generate a statistics report with the total Sales for female and male employees

import sys if(len(sys.argv)!=3):

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: (x.split('\t')[3],float(x.split('\t')[8]))) total=temp.reduceByKey(lambda a,b : a+b) total.saveAsTextFile(sys.argv[2])

Output:

## Write a map-reduce program to analyze the given sales records over a period and generate data about the country’s total sales, and the total number of the products

import sys if(len(sys.argv)!=3):

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: (x.split(',')[7],1)) data=temp.countByKey() dd=sc.parallelize(data.items()) dd.saveAsTextFile(sys.argv[2])

Output:



## Write a map-reduce program to analyze the given sales records over a period of time and generate data about the country’s total sales and the frequency of the payment mode.

import sys if(len(sys.argv)!=3):

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: (x.split(',')[3],1)) data=temp.countByKey() dd=sc.parallelize(data.items()) dd.saveAsTextFile(sys.argv[2])

Output: