```java
package equake;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Mapper.Context;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

/**
 * This class is a Map reduce program to analyze the seismic data and to
 * tabulate the maximum magnitude value based on region
 *
 * @author Arun Kumar P
 *
 */
public class SeismicAnalysis {

    public static void main(String[] args) throws Exception {
        Configuration c = new Configuration();
        String[] files = new GenericOptionsParser(c, args).getRemainingArgs();
        Path input = new Path(files[0]);
        Path output = new Path(files[1]);
        Job j = new Job(c, "eqdata");
        j.setJarByClass(SeismicAnalysis.class);
        j.setMapperClass(SeismicMapper.class);
        j.setReducerClass(SeismicReducer.class);
        j.setOutputKeyClass(Text.class);
        j.setOutputValueClass(FloatWritable.class);
        FileInputFormat.addInputPath(j, input);
        FileOutputFormat.setOutputPath(j, output);
        System.exit(j.waitForCompletion(true) ? 0 : 1);
    }

    /**
     * SeismicMap represents the mapper class.
     * Input Key and Input value - Seismic data from csv
     * Output Key - region (Ex -Northern California)
     * Output Value - magnitude (1.5,1.2,1.9)
     *
     * @author Arun Kumar P
     *
     */
    public static class SeismicMapper extends
            Mapper<LongWritable, Text, Text, FloatWritable> {
        /* (non-Javadoc)
         * @see org.apache.hadoop.mapreduce.Mapper#map(KEYIN, VALUEIN,
         org.apache.hadoop.mapreduce.Mapper.Context)
         */
        public void map(LongWritable key, Text value, Context con)
                throws IOException, InterruptedException {
            String line = value.toString();
            String[] data = line.split(",");
            float magnitude = Float.parseFloat(data[8]);
            String region = data[11];
            Text outputKey = new Text(region.toUpperCase().trim());
            FloatWritable outputValue = new FloatWritable(magnitude);
            con.write(outputKey, outputValue);
        }
    }

    /**
     * SeismicReducer represents the reducer class
     * Input Key    -   region (Ex - Northern California)
     * Input value  -   magnitude (Ex - 1.5,1.2,1.9)
     * Output Key   -   region (Ex - Northern California)
     * Output Value -   maximum magnitude (Ex - 3.3)
     *
     * @author Arun Kumar P
     *
     */
    public static class SeismicReducer extends
            Reducer<Text, FloatWritable, Text, FloatWritable> {
        /* (non-Javadoc)
         * @see org.apache.hadoop.mapreduce.Reducer#reduce(KEYIN, java.lang.Iterable,
         org.apache.hadoop.mapreduce.Reducer.Context)
         */
        public void reduce(Text region, Iterable<FloatWritable> values,
                Context con) throws IOException, InterruptedException {
            ArrayList<Float> mag = new ArrayList<Float>();
            for (FloatWritable value : values) {
                float val = Float.valueOf(value.get());
                mag.add(val);
            }
            float maxMag = Collections.max(mag);
            con.write(region, new FloatWritable(maxMag));
        }
    }

}
```