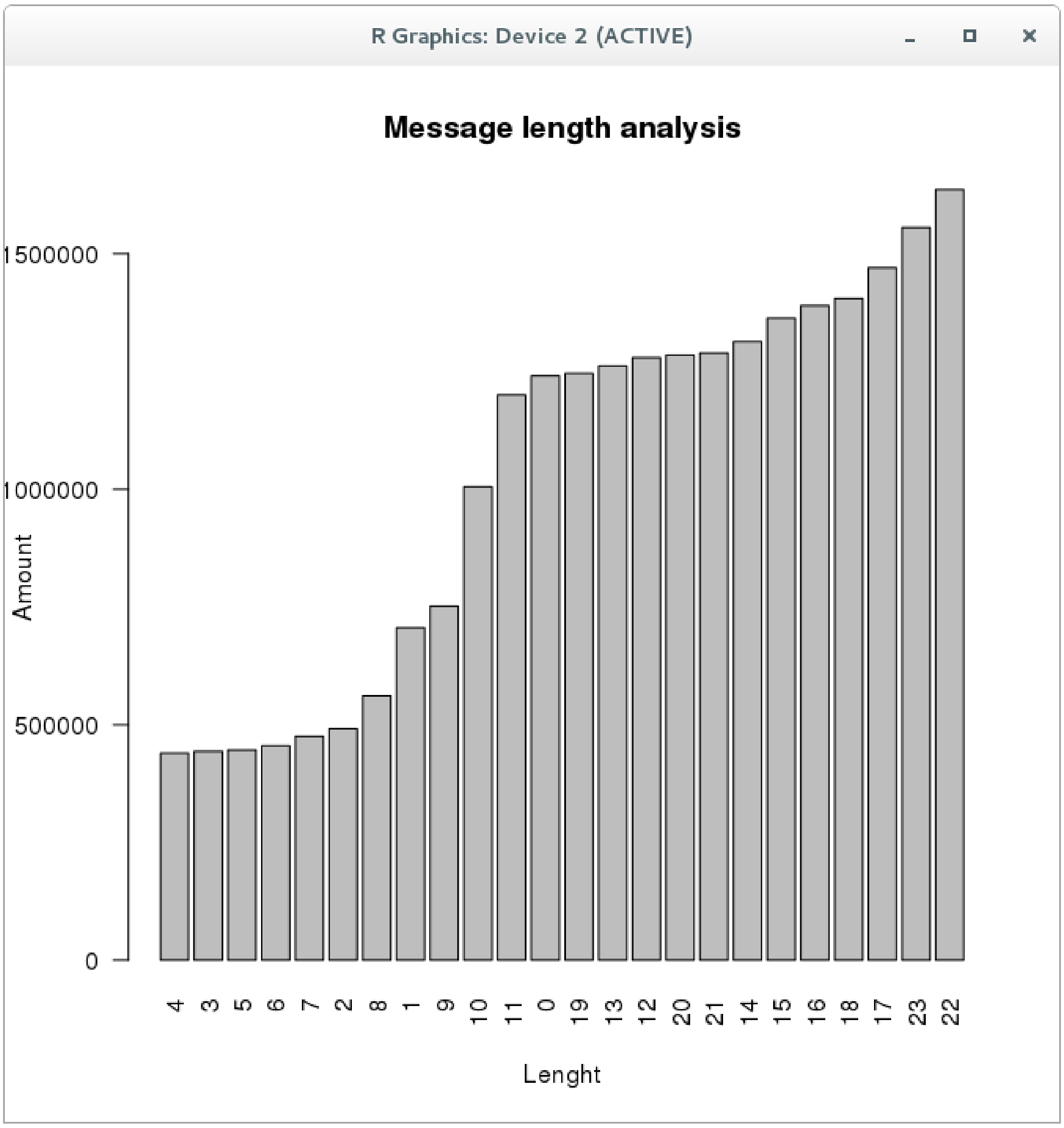
MAPPER:

```
package bdp.hours;
import java.io.IOException;
import java.time.Instant;
import java.time.ZoneId;
import java.time.ZonedDateTime;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.io.LongWritable;

public class HoursMapper extends Mapper<Object, Text, IntWritable, IntWritable>
{
  private final IntWritable one = new IntWritable(1);
  private IntWritable data = new IntWritable();
  String[] fields = new String[4];
  public void map(Object key, Text line, Context context) throws IOException,
InterruptedException
  {
        final int number;
    //Fields contains line as follows.
    //   0         1                  2            3
    //epoch_time;tweetId;tweet(including #hashtags);device
if(line.toString().split(";").length == 4)
{
fields = line.toString().split(";");
}
number = fields[2].length();
if(number <= 140)
{
        try{
   //Obtains an instance of Instant using milliseconds from the epoch field in our string array
                  Instant t = Instant.ofEpochMilli(Long.valueOf(fields[0]).longValue());
//Obtains an instance of ZonedDateTime
//from the instant formed by combining the local date-time and offset.
    ZonedDateTime d = ZonedDateTime.ofInstant(t, ZoneId.of("-3"));
//setting the needed hour to out key
    data.set(d.getHour());
                  context.write(data, one);
                 }
//cathing format exceptions
   catch(NumberFormatException ex){
            ex.printStackTrace();
                 }}}}
```

Mapper extracts the hour from the 2 index via java 8 api for time. After that it sends the K-V pair containing the hour and 1 to reducer. Reducer is the same as in the previous task.


R Graphics: Device 2 (ACTIVE)
Message length analysis
Amount
Lenght