MAPPER:

```
package bdp.tweets;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class TweetsCountMapper extends Mapper<Object, Text, Text, IntWritable>
 {
    private IntWritable one = new IntWritable(1);
     private Text data = new Text();
     //an array for our input string
     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
//checking the length of line
if(line.toString().split(";").length == 4)
{
fields = line.toString().split(";");
}
number = fields[2].length(); //length of the tweet
//checking the length of the tweet to avoid non-standard encoding that might cause some
unexpected (i.e. too high) values
if(number <= 140)
{
  //deciding to which group by the value: when it is "1-5" at the end or "6-10"
   if(number%10 <=5 && number%10 >=1)
{
//describing the length
data.set("length " + number/10  + "0-" + number/10  + "5.");
}
else
{
  int a = number/10 +1;
data.set("length " + (number/10)  + "6-" + a + "0.");
}
//sending our resulted pair to reducer for further counting
context.write(data, one);
}}}
```

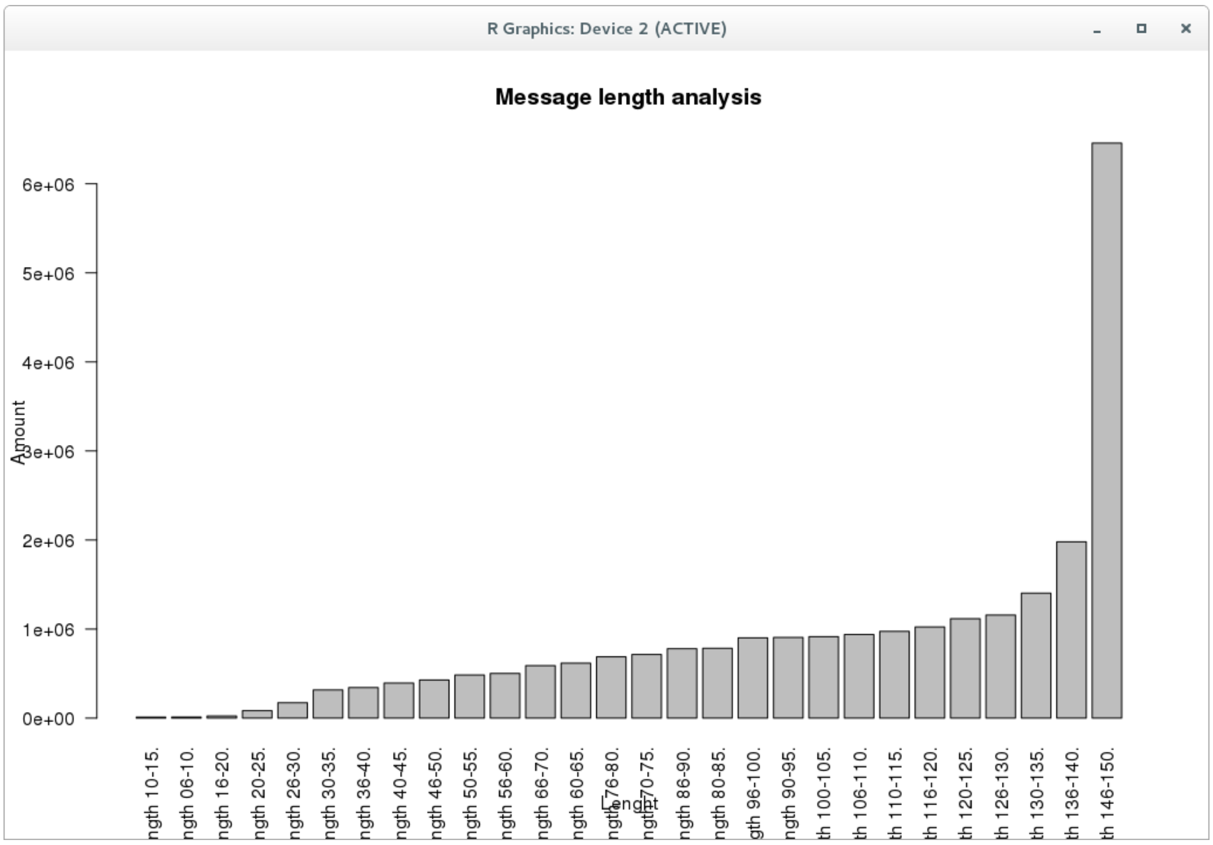
REDUCER:

```
package bdp.tweets;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class TweetsCountReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
   private IntWritable result = new IntWritable();
   public void reduce(Text key, Iterable<IntWritable> values, Context context)
      throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable value : values) {
         //counting the amount of tweets from the group
         sum+=value.get();
      }
      result.set(sum);
      //returning result
      context.write(key,result);
   }
}
```

Mapper is responsible for all the data extraction and parsing the length of the string, which would be passed to the reducer. To do this, the file is read line by line, after that I counted the length of the field with the tweet itself. The length of the tweet is then analysed and associated with the group (which is a text line). After that the K-V pair, containing the name of the group and the 1 is sent to a reducer to count the final amount of data which have the certain length. Reduced returns the K-V pair, containing the name of the cgroup and the summarised amount of data which has the needed length.


R Graphics: Device 2 (ACTIVE)
Message length analysis
Amount
6e+06
5e+06
4e+06
3e+06
2e+06
1e+06
0e+00
Lenght
ngth 10-15.
ngth 06-10.
ngth 16-20.
ngth 20-25.
ngth 26-30.
ngth 30-35.
ngth 36-40.
ngth 40-45.
ngth 46-50.
ngth 50-55.
ngth 56-60.
ngth 66-70.
ngth 60-65.
ngth 76-80.
ngth 70-75.
ngth 86-90.
ngth 80-85.
gth 96-100.
ngth 90-95.
th 100-105.
th 106-110.
th 110-115.
th 116-120.
th 120-125.
th 126-130.
th 130-135.
th 136-140.
th 146-150.