Hands On: Plot the Degree Histogram

In this activity, we are going to plot the degree histogram that we created in the previous activity *Hands-On: Building a Degree Histogram*. Make sure to complete that activity, as well as the previous to that (*Hands-On: Building a Graph*) before starting this one.

Step 1. Import the relevant libraries and functions.

We are using the library XChart. We start by importing all the functions and classes that we need to build our plot.

```
import org.knowm.xchart.{XYChartBuilder, SwingWrapper, BitmapEncoder, CategoryChart, Catego
import org.knowm.xchart.style.markers.SeriesMarkers
import org.knowm.xchart.BitmapEncoder.BitmapFormat
```

Step 2. Define a function to calculate the degree histogram.

Define a function to create a histogram of the degrees. The function is nearly identical to how the histogram was computed in the previous hands on exercise. Remember to only include countries!

Input:

```
def degreeHistogram(net: Graph[PlaceNode, Int]): Array[(Int, Int)] =
  net.degrees.
    filter { case (vid, count) => vid >= 100 }.
    map(t => (t._2,t._1)).
    groupByKey.map(t => (t._1,t._2.size)).
    sortBy(_._1).collect()
```

Output:

```
degreeHistogram: (net: org.apache.spark.graphx.Graph[PlaceNode,Int])Array[(Int, Int)]
```

Step 3. Calculate the probability distribution for the degree histogram.

Get the probability distribution (degree distribution) from the degree histogram by normalizing the node degrees by the total number of nodes, so that the degree probabilities add up to one.

Input:

```
val nn = metrosGraph.vertices.filter{ case (vid, count) => vid >= 100 }.count()
```

Output:

```
nn: Long = 28
```

Input:

```
val metroDegreeDistribution = degreeHistogram(metrosGraph).map({case(d,n) => (d,n.toDouble/
```

Output:

```
metroDegreeDistribution: Array[(Int, Double)] = Array((1,0.6428571428571429), (2,0.14285714
(3,0.07142857142857142), (5,0.07142857142857142), (9,0.03571428571428571), (14,0.0357142857
```

Step 4. Graph the results.

Plot degree distribution and the histogram of vertex degrees.

Input:

```
val xData = metroDegreeDistribution.map(_._1.toDouble)
val yData = metroDegreeDistribution.map(_._2)

val chart = new XYChartBuilder().width(800).height(600).title("Degree Distribution").xAxisT
chart.getStyler.setMarkerSize(6)
val series = chart.addSeries("Degree Distribution", xData, yData)
series.setMarker(SeriesMarkers.CIRCLE)
BitmapEncoder.saveBitmap(chart, "./Degree_Distribution.png", BitmapFormat.PNG)

val chart = new CategoryChartBuilder().width(800).height(600).title("Degree Histogram of No
chart.addSeries("Frequency", xData.map(_.toDouble), yData.map(_.toDouble))
BitmapEncoder.saveBitmap(chart, "./Degree_Histogram.png", BitmapFormat.PNG)
```

Step 5. Display the graphs.

First, exit the Spark's shell by typing *Ctrl+C* in your keyboard. Then run *exit* in the container's shell.

```
exit
```

Once you're in your local terminal shell, run the following commands to copy the plots from from the container to your local system (make sure to be in you *big-data-5/graphx* directory).

Copy degree distribution:

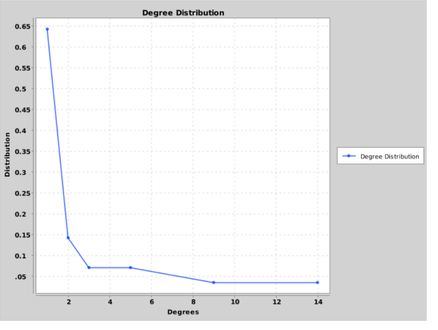
```
docker cp graphx-coursera:Degree_Distribution.png .
```

Copy degree histogram:

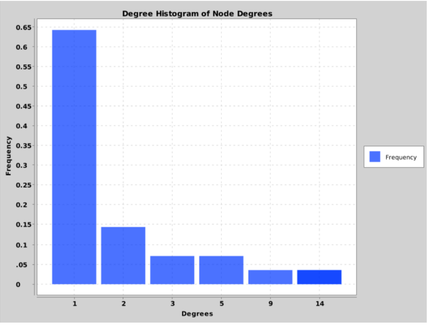
```
docker cp graphx-coursera:Degree_Histogram.png .
```

Now you can display both files locally.

Degree Distribution



Degree Histogram



Mark as completed

Like Dislike Report an issue