```
n = 100; (*number of nodos*)
repsp = 10; (*times we use the same probability value p*)
dp = 0.01; (*probability steps' size*)

complexity = Table[0, repsp]; (*list to save K-complexity (1-D representation) for the same p
value*)
complexitymat = Table[0, repsp];  (*list to save K-complexity (adjacency matrix) for the same p
value*)
entropy = Table[0, repsp]; (*list to save entropy values for the same p value*)
seqnet = Table[0, (1/dp) + 1]; (*list to save the K-complexity for each p value (1-D
representation)*)
Matnet = Table[0, (1/dp) + 1]; (*list to save the K-complexity for each p value (2-D
representation)*)
ShannonEntropy = Table[0, (1/dp) + 1];  (*list to save the entropy for each p value*)

Do[
 Do[
  rg = RandomGraph[WattsStrogatzGraphDistribution[n, l, 5]]; (*random graph*)
  mat = Flatten[Normal[AdjacencyMatrix[rg]]]; (*1-D representation of the random graph*)
  If[First[mat] == 0, output = "0" <> ToString[FromDigits[mat]], output =
ToString[FromDigits[mat]]]; (*transform the list into a string*)

  (*measure the complexity using three different methods*)
  complexity[[j]] = StringBDM[output];
  complexitymat[[j]] = BDM[Normal[AdjacencyMatrix[rg]], 4] // N;
  entropy[[j]] = Entropy[output];
  , {j, repsp}];

 (*we perform a trimmed mean discarding the first and fourth quartiles*)
 seqnet[[(1/dp)*(l + dp)]] = TrimmedMean[complexity, {(Length[Select[complexity, # <
Quantile[complexity, 1/4] &]])/Length[complexity], (Length[Select[complexity, # >
Quantile[complexity, 3/4] &]])/Length[complexity]}];
 Matnet[[(1/dp)*(l + dp)]] = TrimmedMean[complexitymat, {(Length[Select[complexitymat, # <
Quantile[complexitymat, 1/4] &]])/Length[complexitymat], (Length[Select[complexitymat, # >
Quantile[complexitymat, 3/4] &]])/Length[complexitymat]}];
 ShannonEntropy[[(1/dp)*(l + dp)]] = TrimmedMean[entropy, {(Length[Select[entropy, # <
Quantile[entropy, 1/4] &]])/Length[entropy], (Length[Select[entropy, # > Quantile[entropy, 3/4]
&]])/Length[entropy]}];
 , {l, 0, 1, dp}]

(*we can plot the results*)
ListLinePlot[{Rescale[seqnet, {0, Max[seqnet]}], Rescale[Matnet, {0, Max[Matnet]}],
Rescale[ShannonEntropy, {0, Max[ShannonEntropy]}]}, DataRange -> {0, 1}, AxesLabel -> Automatic,
PlotRange -> All, PlotLegends -> Placed[{"K-Complexity (1-D representation)", "K-Complexity
(adjacency matrix)", "Entropy"}, {.5, .2}], Frame -> True, GridLines -> Automatic, FrameLabel ->
{"p", "C(G)"}]
```