

# Reduced Networks 03

Investigation in how reduced sampling (selecting a sub-sample of the tagged hens per pen) influences social network descriptors

Version: 3.0.

Author: Bernhard Voelkl

Created in: *Mathematica* 10.2

Last modified: 16/02/2021

## Data Import

```
In[1]:= SetDirectory["M://Chicken Research/ChickenRanging4"]
Out[1]= M:\Chicken Research\ChickenRanging4

In[2]:= data = Get["transitiondata"];
          
```

```
In[3]:= allhens = Get["allhens"];
          
```

```
In[4]:= datelist = Get["datelist"];
          
```

```
In[5]:= columns = {"tag", "hen", "ranger", "pen", "date", "absolut time", "from", "to", "time",
           "WG open", "LH open", "FR open", "WG closed", "LH closed", "FR closed"};
          
```

```
In[6]:= dayofyear = Sort[Table[DateDifference[DateList["1 Jan, 2016"],
           DateList[{datelist[[i]], {"DayName", "Day", "Month", "Year"}}]], {i, 72}]]\[All, 1];
          
```

```
In[7]:= SetDirectory["M://Chicken Research/ChickenSocialNetwork"]
Out[7]= M:\Chicken Research\ChickenSocialNetwork
```

## Functions

```
In[1]:= findFriend[i_] := Module[{hen1, pothens, hen2, friends, shift = 0},
  hen1 = pen[[i]];
  pothens = Cases[pen, {_, _, _, _, _, t_, from_, to_, ___} /;
    t - shift ≥ hen1[[6]] && t - shift < hen1[[6]] + 5 && from == hen1[[7]] && to == hen1[[8]]];
  pothens = DeleteCases[pothens, {x_, ___} /; x == hen1[[1]]];
  pothens = Cases[pothens, {_, _, _, _, _, t_, ___} /; t == pothens[[1, 6]]];
  If[Length[pothens] > 0,
    hen2 = RandomChoice[pothens];
    friends = Transpose[{Table[hen1[[1]], {Length[pothens]}], pothens\[All, 1]}],
    friends = {}
  ];
  friends
]
```

`findFriend[i]` is a function that requires that a matrix `pen` is globally defined, where `pen` gives the temporally sorted transition data for one pen for one entire day. Function input `i` is an integer counting variable (indicating the `i`th line of the matrix `pen`). For each transition of a hen the

function finds all other transitions by other hens within 5 seconds at the same antenna (in the same direction). The hen closest in time is considered the “friend” of the focus animal and a an edge {ID focus, ID friend} is created. In case more than one hen crossed the antenna within the same second an edge is created for each pairing with the focus bird {{ID focus, ID friend1}, {ID focus, ID friend2} ..}. In case no other hen crosses the antenna within 5 seconds, an empty list is returned, otherwise the function returns a list of edges.

```
In[4]:= rank[list_] := Module[{ranklist},
  ranklist = (Ordering@Ordering@list + Reverse@Ordering@Ordering@Reverse@list) / 2;
  ReplaceAll[ranklist,
    Apply[Rule, Transpose[{Union[ranklist], Ordering[Union[ranklist]]}]], {1}]]
]
```

**rank**[*List*] is a function that calculates ranks for elements in a list with tied values getting intermediate ranks

```

In[+]:= reducedsampling[r_] := Module[
  {drops, penhensr, m, amatr, pair, aimatr, penaigraphr, penaimultigraphr, edgelistpenr,
   indivobservationsr, max, k, strdistr, gdr, assr, btwr, prcr, lscr, gcr, measuresr},
  drops = Partition[Sort[RandomSample[Range[n], r]], 1];
  penhensr = Delete[penhens, drops];
  m = Length[penhensr];
  amatr = Table[0, {m}, {m}];
  For[i = 1, i ≤ m, i++,
    For[j = 1, j ≤ m, j++,
      pair = Cases[edgelistpen, {{penhensr[[i]], penhensr[[j]]}, _}]];
      If[Length[pair] > 0,
        amatr[[i, j]] = pair[[1, 2]]];
    ];
  ];
  amatr = amatr + Transpose[amatr];
  indivobservationsr = Delete[indivobservations, drops];
  aimatr = N[Table[If[indivobservationsr[[i]] + indivobservationsr[[j]] > 0,
    amatr[[i, j]] / (indivobservationsr[[i]] + indivobservationsr[[j]] - amatr[[i, j]]), "NA"],
    {i, Length[penhensr]}, {j, Length[penhensr]}]]];
  penaigraphr = WeightedAdjacencyGraph[aimatr //. {"NA" → 0, 0. → 0},
    EdgeWeight, DirectedEdges → True, EdgeStyle → Arrowheads[0]];
  penaimultigraphr = AdjacencyGraph[Round[(aimatr //. {"NA" → 0, 0. → 0}) 10000],
    DirectedEdges → True, EdgeStyle → Arrowheads[0]];
  max = Total[DeleteCases[Flatten[amatr], "NA"]] / 2;
  (*total number of links*)
  k = m (m - 1);
  edgelistpenr = DeleteCases[edgelistpen,
    {{___, x_, ___}, _} /; MemberQ[penhens[[Flatten[drops]]], x]];
  strdistr = Transpose[{Range[k] / k,
    PadLeft[Accumulate[Sort[edgelistpenr[[All, 2]]]], k, 0] / max}];
  gdr = N[GraphDensity[penaimultigraphr]]; (*Graphdensity*)
  assr = GraphAssortativity[penaigraphr, FindGraphCommunities[penaigraphr]];
  btwr = BetweennessCentrality[penaimultigraphr];
  prcr = PageRankCentrality[penaimultigraphr, 0.1];
  lscr = StatusCentrality[penaimultigraphr];
  gcr = FindGraphCommunities[penaigraphr];
  measuresr = {penhensr, strdistr, gdr, btwr, prcr, lscr, gcr, r};
  measuresr
]

```

**reducedsampling[r]** is a function that recalculates network statistics on a reduced network with r randomly drawn individuals removed from the data. The output is a list containing (1) a list of henIDs of the network, the strength distribution, graph density, a list with betweennness centralities, a list with Page rank centrality, a list with status centrality, a list with sub-graph community members, and r. Apart from the input provided, the lists *indivobservations*, *edgelistpen*, *penhens* must be defined before running the function.

# Pen 11

## Edgelist

```

In[1]:= data = Cases[data, {_, _, _, _, 11, ___}];

In[2]:= penstring = "11";

In[3]:= SeedRandom[1492]

In[4]:= penhens = Cases[allhens, {_, penstring, ___}] [[All, 3]];

In[5]:= n = Length[penhens];

edgelistpen = {};
For[d = 1, d ≤ Length[dateList], d++,
  pen = Cases[data, {_, _, _, _, _, dateList[[d]], ___}];
  pen = Sort[pen, #1[[6]] < #2[[6]] &];
  If[Length[pen] > 1,
    AppendTo[edgelistpen, ParallelTable[findFriend[i], {i, 1, Length[pen] - 1}]]];
]
];

edgelistpen = DeleteCases[Flatten[edgelistpen, 2], {}];
edgelistpen = Sort /@ edgelistpen;
edgelistpen = Tally[edgelistpen];
edgelistpen = DeleteCases[edgelistpen, {{v1_, v2_}, _} /; v1 == v2];

Save["edgelistpen11red", edgelistpen];

In[6]:= edgelistpen = Get["edgelistpen11red"];

In[7]:= indivobservations = Table[Length[Cases[data, {penhens[[i]], ___}]], {i, Length[penhens]}];

```

## Full network

```

In[1]:= amat = Table[0, {n}, {n}];

In[2]:= For[i = 1, i ≤ n, i++,
  For[j = 1, j ≤ n, j++,
    pair = Cases[edgelistpen, {{penhens[[i]], penhens[[j]]}, ___}];
    If[Length[pair] > 0,
      amat[[i, j]] = pair[[1, 2]];
    ]
  ]
]

In[3]:= amat = amat + Transpose[amat];

In[4]:= aimat = N[Table[If[indivobservations[[i]] + indivobservations[[j]] > 0,
  amat[[i, j]] / (indivobservations[[i]] + indivobservations[[j]] - amat[[i, j]]), "NA"],
{i, Length[penhens]}, {j, Length[penhens]}]];

In[5]:= penaigraph = WeightedAdjacencyGraph[aimat //. {"NA" → 0, 0. → 0},
EdgeWeight, DirectedEdges → True, EdgeStyle → Arrowheads[0]];

In[6]:= penaimultigraph = AdjacencyGraph[Round[(aimat //. {"NA" → 0, 0. → 0}) 10000],
DirectedEdges → True, EdgeStyle → Arrowheads[0]];

```

```

ln[ ]:= max = Total[edgelistpen[[All, 2]]]; (*total number of links*)

ln[ ]:= k = n (n - 1);

ln[ ]:= strdist =
    Transpose[{Range[k] / k, PadLeft[Accumulate[Sort[edgelistpen[[All, 2]]]], k, 0] / max}];

gd = N[GraphDensity[penaimultigraph]];

ln[ ]:= ass = GraphAssortativity[penaigraph, FindGraphCommunities[penaigraph]];

ln[ ]:= btw = BetweennessCentrality[penaimultigraph];

ln[ ]:= prc = PageRankCentrality[penaimultigraph, 0.1];

ln[ ]:= lsc = StatusCentrality[penaimultigraph];

ln[ ]:= gc = FindGraphCommunities[penaigraph];

ln[ ]:= measures = {penhens, strdist, gd, btw, prc, lsc, gc};

```

## Reduced networks

```

redmeasures = Table[reducedsampling[r], {r, 1, n - 30}];

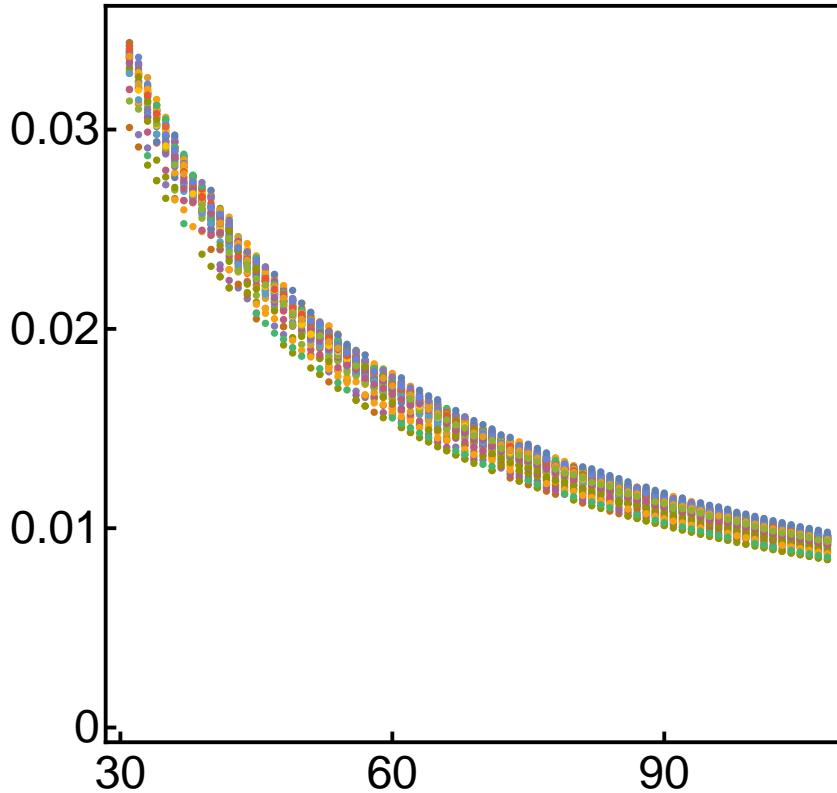
Save["measures", measures];
Save["redmeasures", redmeasures];

ln[ ]:= measures = Get["measures"];

ln[ ]:= redmeasures = Get["redmeasures"];

```

```
In[6]:= pen11centrality = ListPlot[Table[Flatten[
  Reverse[Table[redmeasures[[r, 5, Flatten[Position[redmeasures[[r, 1]], penhens[[ph]]]]], {r, 1, Length[redmeasures]}] //.
    {} -> Null]], {ph, 1, Length[penhens]}]],
  PlotStyle -> PointSize[0.01], AspectRatio -> 1, Frame -> True,
  FrameStyle -> Directive[Thick, Black, 24],
  FrameTicks -> {{{{0, 0.01, 0.02, 0.03}, {}}, {{{0, "30"}, {30, "60"}, {60, "90"}, {}}}}}]
```

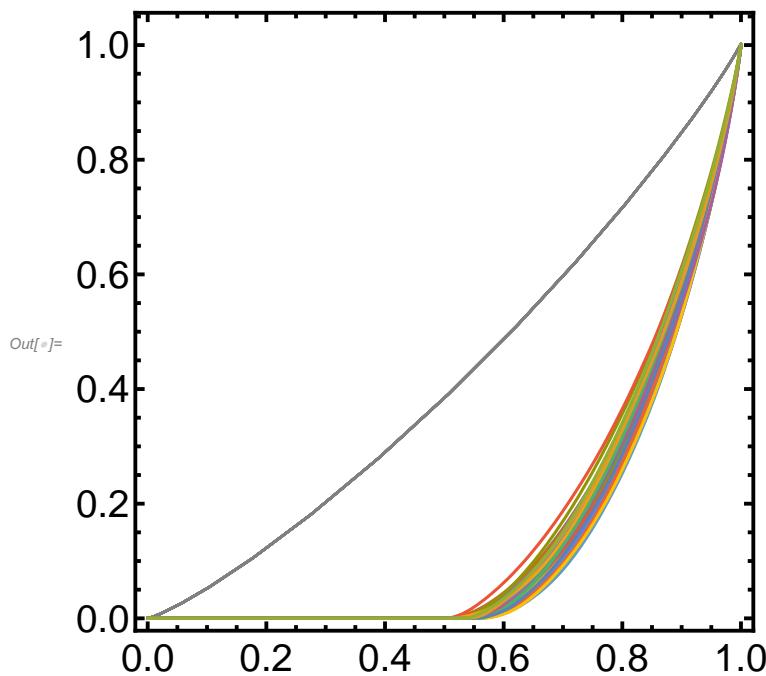


```
In[7]:= Export["pen11centrality.jpg", pen11centrality]
```

```
Out[7]= pen11centrality.jpg
```

```
In[8]:= pen11strengthdistributionred =
  Show[ListPlot[Transpose[{Range[k]/k, PadLeft[Accumulate[
    Sort[Table[Random[PoissonDistribution[N[max/k]]], {k}]]], k, 0]/max}],
  PlotStyle -> Gray, Frame -> True, AspectRatio -> 1, FrameStyle ->
  Directive[Thick, Black, 20], LabelStyle -> Directive[18, Black]],
  ListLinePlot[Table[redmeasures[[i, 2]], {i, 1, Length[redmeasures]}]]];
```

```
In[6]:= pen11strengthdistributionred
```

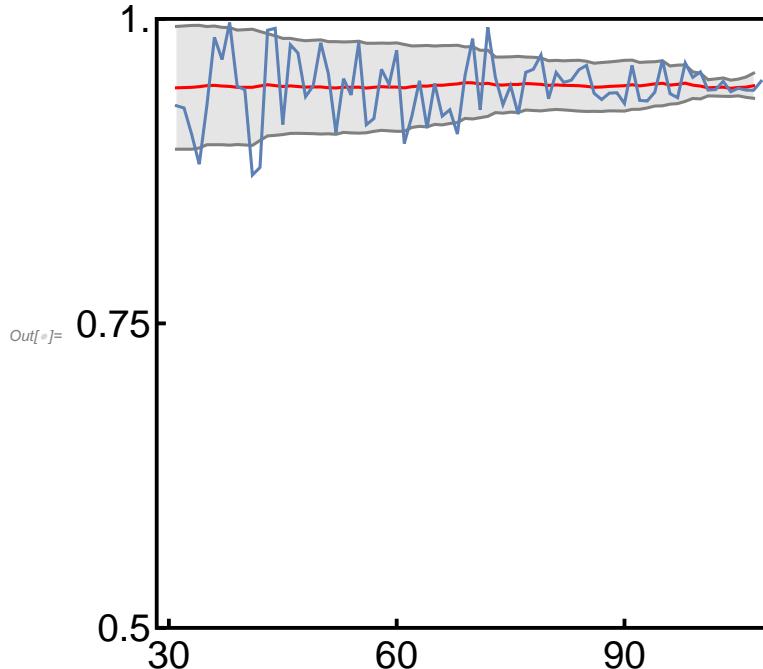


```
In[6]:= Export["pen11strengthdistributionred.jpg", pen11strengthdistributionred]
```

```
Out[6]= pen11strengthdistributionred.jpg
```

```
In[6]:= stdr = Reverse[  
  Table[StandardDeviation[redmeasures[[1;; i, 3]]], {i, 2, Length[redmeasures]}]];  
 mr = Reverse[Table[Mean[redmeasures[[1;; i, 3]]], {i, 2, Length[redmeasures]}]];
```

```
In[6]:= pen11density = Show[ListLinePlot[{mr + 1.96 stdr, mr - 1.96 stdr}, Filling -> {1 -> {2}}, PlotStyle -> Gray, PlotRange -> {All, {0.5, 1}}], ListLinePlot[mr, PlotStyle -> Red], ListLinePlot[Reverse[redmeasures[[All, 3]]]], Frame -> True, AspectRatio -> 1, FrameStyle -> Directive[Thick, Black, 20], LabelStyle -> Directive[18, Black], FrameTicks -> {{{{0.5, 0.75, 1.0}, {}}, {{0, "30"}, {30, "60"}, {60, "90"}, {}}}}]
```



```
In[6]:= Export["pen11density.jpg", pen11density]
```

```
Out[6]= pen11density.jpg
```

## Pen 12

### Edgelist

```
In[1]:= SetDirectory["M://Chicken Research/ChickenRanging4"]
```

```
Out[1]= M:\Chicken Research\ChickenRanging4
```

```
In[2]:= data = Get["transitiondata"];
```

Get: Cannot open transitiondata.

```
In[3]:= SetDirectory["M://Chicken Research/ChickenSocialNetwork"]
```

```
Out[3]= M:\Chicken Research\ChickenSocialNetwork
```

```
data = Cases[data, {_, _, _, 12, ___}];
```

```
penstring = "12";
```

```
In[4]:= SeedRandom[1492]
```

```
In[5]:= penhens = Cases[allhens, {_, penstring, ___}] [[All, 3]];
```

```

In[1]:= n = Length[penhens];

edgelistpen = {};
For[d = 1, d ≤ Length[datelist], d++,
  pen = Cases[data, {_, _, _, _, _, datelist[[d]], ___}];
  pen = Sort[pen, #1[[6]] < #2[[6]] &];
  If[Length[pen] > 1,
    AppendTo[edgelistpen, ParallelTable[findFriend[i], {i, 1, Length[pen] - 1}]]]
  ]
];

edgelistpen = DeleteCases[Flatten[edgelistpen, 2], {}];
edgelistpen = Sort /@ edgelistpen;
edgelistpen = Tally[edgelistpen];
edgelistpen = DeleteCases[edgelistpen, {{v1_, v2_}, _} /; v1 == v2];

Save["edgelistpen12red", edgelistpen];
edgelistpen = Get["edgelistpen12red"];

In[2]:= indivobservations = Table[Length[Cases[data, {penhens[[i]], ___}]], {i, Length[penhens]}];

```

## Full network

```

In[3]:= amat = Table[0, {n}, {n}];

In[4]:= For[i = 1, i ≤ n, i++,
  For[j = 1, j ≤ n, j++,
    pair = Cases[edgelistpen, {{penhens[[i]], penhens[[j]]}, ___}];
    If[Length[pair] > 0,
      amat[[i, j]] = pair[[1, 2]]
    ]
  ]
];

In[5]:= amat = amat + Transpose[amat];

In[6]:= aimat = N[Table[If[indivobservations[[i]] + indivobservations[[j]] > 0,
  amat[[i, j]] / (indivobservations[[i]] + indivobservations[[j]] - amat[[i, j]]), "NA"],
  {i, Length[penhens]}, {j, Length[penhens]}]];

In[7]:= penaigraph = WeightedAdjacencyGraph[aimat //. {"NA" → 0, 0. → 0},
  EdgeWeight, DirectedEdges → True, EdgeStyle → Arrowheads[0]];

In[8]:= penaimultigraph = AdjacencyGraph[Round[(aimat //. {"NA" → 0, 0. → 0}) 10000],
  DirectedEdges → True, EdgeStyle → Arrowheads[0]];

In[9]:= max = Total[edgelistpen[[All, 2]]]; (*total number of links*)

In[10]:= k = n (n - 1);

In[11]:= strdist =
  Transpose[{Range[k] / k, PadLeft[Accumulate[Sort[edgelistpen[[All, 2]]]], k, 0] / max}];

gd = N[GraphDensity[penaimultigraph]];

In[12]:= ass = GraphAssortativity[penaigraph, FindGraphCommunities[penaigraph]];

In[13]:= btw = BetweennessCentrality[penaimultigraph];

```

```
In[1]:= prc = PageRankCentrality[penaimultigraph, 0.1];
In[2]:= lsc = StatusCentrality[penaimultigraph];
In[3]:= gc = FindGraphCommunities[penaigraph];
In[4]:= measures = {penhens, strdist, gd, btw, prc, lsc, gc};
```

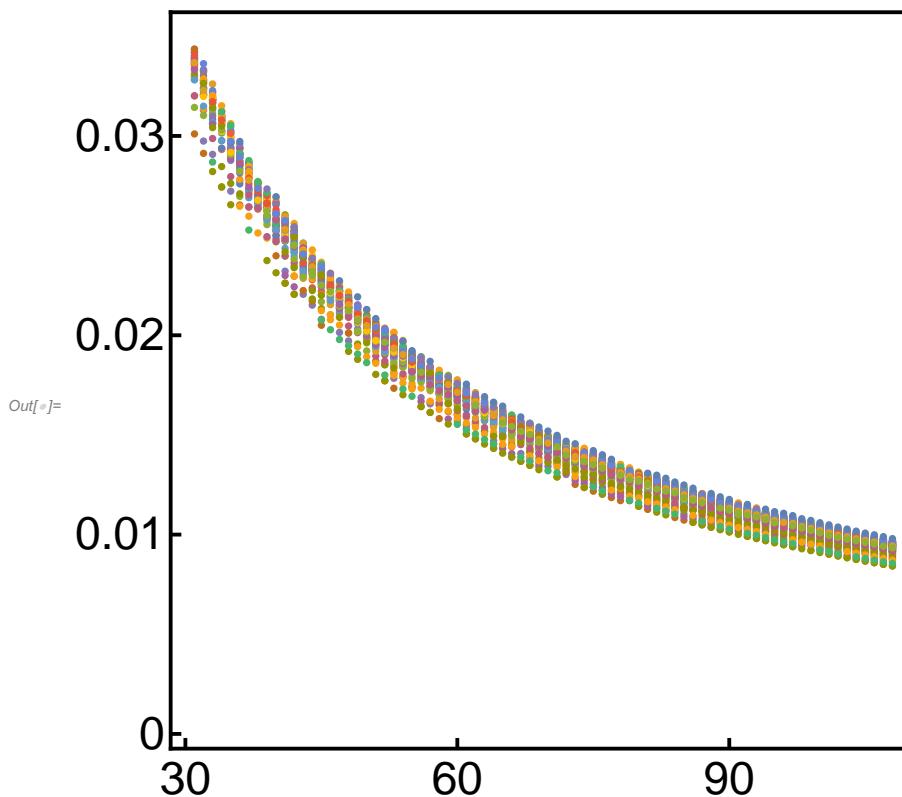
## Reduced networks

```
redmeasures = Table[reducedsampling[r], {r, 1, n - 30}];

Save["measures12", measures];
Save["redmeasures12", redmeasures];

measures = Get["measures12"];
redmeasures = Get["redmeasures12"];

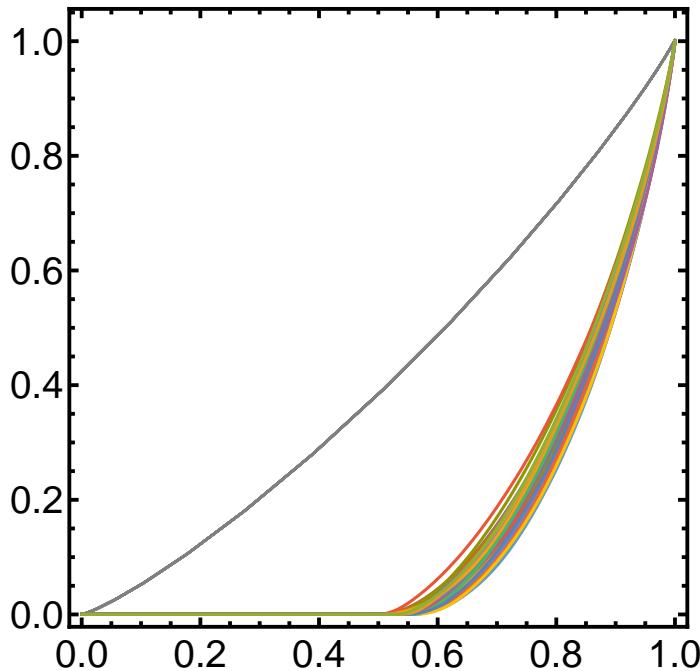
pen12centrality = ListPlot[Table[Flatten[
  Reverse[Table[redmeasures[[r, 5, Flatten[Position[redmeasures[[r, 1]], penhens[[ph]]]]]], {r, 1, Length[redmeasures]}] //.
    {} \[Rule] Null]], {ph, 1, Length[penhens]}]],
  PlotStyle \[Rule] PointSize[0.01], AspectRatio \[Rule] 1, Frame \[Rule] True,
  FrameStyle \[Rule] Directive[Thick, Black, 24],
  FrameTicks \[Rule] {{{0, 0.01, 0.02, 0.03}, {}}, {{0, "30"}, {30, "60"}, {60, "90"}}, {}}}]
```



```
Export["pen12centrality.jpg", pen12centrality]
```

```
pen12strengthdistributionred =
Show[ListPlot[Transpose[{Range[k] / k, PadLeft[Accumulate[
Sort[Table[Random[PoissonDistribution[N[max / k]]], {k}]]], k, 0] / max}],
PlotStyle -> Gray, Frame -> True, AspectRatio -> 1, FrameStyle ->
Directive[Thick, Black, 20], LabelStyle -> Directive[18, Black]],
ListLinePlot[Table[redmeasures[[i, 2]], {i, 1, Length[redmeasures]}]]];
```

pen12strengthdistributionred

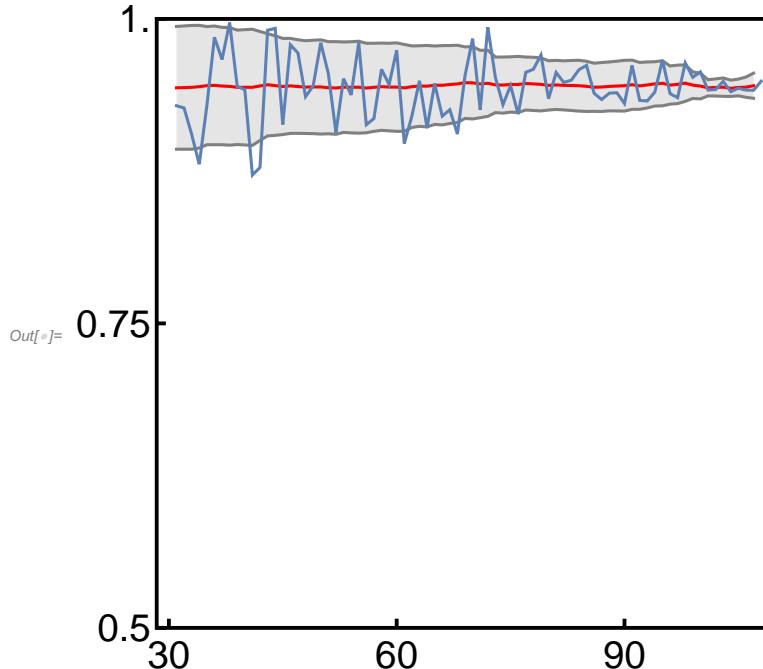


```
Export["pen12strengthdistributionred.jpg", pen12strengthdistributionred]
```

Out[<sup>11</sup>]= pen11strengthdistributionred.jpg

```
In[12]:= stdr = Reverse[
Table[StandardDeviation[redmeasures[[1 ;; i, 3]]], {i, 2, Length[redmeasures]}]];
mr = Reverse[Table[Mean[redmeasures[[1 ;; i, 3]]], {i, 2, Length[redmeasures]}]];
```

```
pen12density = Show[ListLinePlot[{mr + 1.96 stdr, mr - 1.96 stdr}, Filling -> {1 -> {2}}, PlotStyle -> Gray, PlotRange -> {All, {0.5, 1}}], ListLinePlot[mr, PlotStyle -> Red], ListLinePlot[Reverse[redmeasures[[All, 3]]]], Frame -> True, AspectRatio -> 1, FrameStyle -> Directive[Thick, Black, 20], LabelStyle -> Directive[18, Black], FrameTicks -> {{{{0.5, 0.75, 1.0}, {}}, {{0, "30"}, {30, "60"}, {60, "90"}, {}}}}]
```



```
Export["pen12density.jpg", pen12density]
```

---

## Pen 13

### Edgelist

```
In[1]:= SetDirectory["M://Chicken Research/ChickenRanging4"]
Out[1]= M:\Chicken Research\ChickenRanging4

In[2]:= data = Get["transitiondata"];
```

```
In[3]:= SetDirectory["M://Chicken Research/ChickenSocialNetwork"]
Out[3]= M:\Chicken Research\ChickenSocialNetwork

In[4]:= data = Cases[data, {_., _, _, _, 13, ___}];
In[5]:= penstring = "13";
In[6]:= SeedRandom[1492]
In[7]:= penhens = Cases[allhens, {_., penstring, ___}][All, 3];
In[8]:= n = Length[penhens];
```

```

In[1]:= edgelistpen = {};
For[d = 1, d ≤ Length[datelist], d++,
  pen = Cases[data, {_, _, _, _, _, datelist[[d]], ___}];
  pen = Sort[pen, #1[[6]] < #2[[6]] &];
  If[Length[pen] > 1,
    AppendTo[edgelistpen, ParallelTable[findFriend[i], {i, 1, Length[pen] - 1}]]]
  ]
];
In[2]:= edgelistpen = DeleteCases[Flatten[edgelistpen, 2], {}];
edgelistpen = Sort /@ edgelistpen;
edgelistpen = Tally[edgelistpen];
edgelistpen = DeleteCases[edgelistpen, {{v1_, v2_}, ___} /; v1 == v2];

In[3]:= Save["edgelistpen13red", edgelistpen];
In[4]:= edgelistpen = Get["edgelistpen13red"];

In[5]:= indivobservations = Table[Length[Cases[data, {penhens[[i]], ___}]], {i, Length[penhens]}];

```

## Full network

```

In[1]:= amat = Table[0, {n}, {n}];
In[2]:= For[i = 1, i ≤ n, i++,
  For[j = 1, j ≤ n, j++,
    pair = Cases[edgelistpen, {{penhens[[i]], penhens[[j]]}, ___}];
    If[Length[pair] > 0,
      amat[[i, j]] = pair[[1, 2]]
    ]
  ]
];
In[3]:= amat = amat + Transpose[amat];
In[4]:= aimat = N[Table[If[indivobservations[[i]] + indivobservations[[j]] > 0,
  amat[[i, j]] / (indivobservations[[i]] + indivobservations[[j]] - amat[[i, j]]), "NA"],
{i, Length[penhens]}, {j, Length[penhens]}]];
In[5]:= penaigraph = WeightedAdjacencyGraph[aimat //. {"NA" → 0, 0. → 0},
EdgeWeight, DirectedEdges → True, EdgeStyle → Arrowheads[0]];

In[6]:= penaimultigraph = AdjacencyGraph[Round[(aimat //. {"NA" → 0, 0. → 0}) 10000],
DirectedEdges → True, EdgeStyle → Arrowheads[0]];

In[7]:= max = Total[edgelistpen[[All, 2]]]; (*total number of links*)

In[8]:= k = n (n - 1);

In[9]:= strdist =
Transpose[{Range[k] / k, PadLeft[Accumulate[Sort[edgelistpen[[All, 2]]]], k, 0] / max}];

In[10]:= gd = N[GraphDensity[penaimultigraph]];

In[11]:= ass = GraphAssortativity[penaigraph, FindGraphCommunities[penaigraph]];

In[12]:= btw = BetweennessCentrality[penaimultigraph];

In[13]:= prc = PageRankCentrality[penaimultigraph, 0.1];

```

```
In[1]:= lsc = StatusCentrality[penaimultigraph];
In[2]:= gc = FindGraphCommunities[penaigraph];
In[3]:= measures = {penhens, strdist, gd, btw, prc, lsc, gc};
```

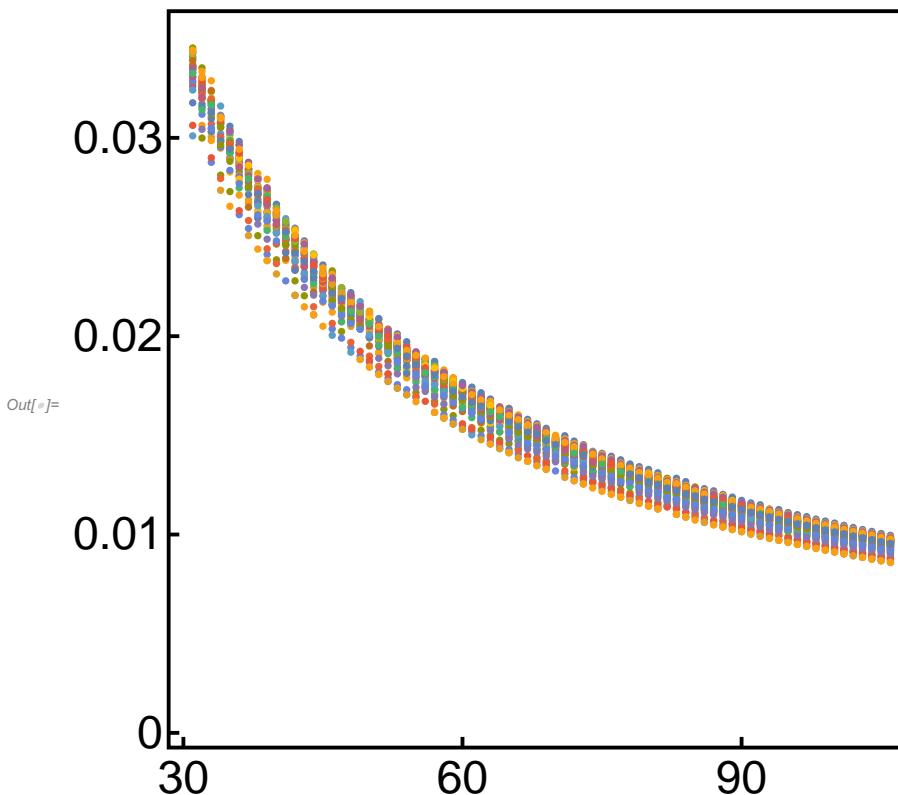
## Reduced networks

```
In[4]:= redmeasures = Table[reducedsampling[r], {r, 1, n - 30}];

In[5]:= Save["measures13", measures];
Save["redmeasures13", redmeasures];

In[6]:= measures = Get["measures13"];
redmeasures = Get["redmeasures13"];

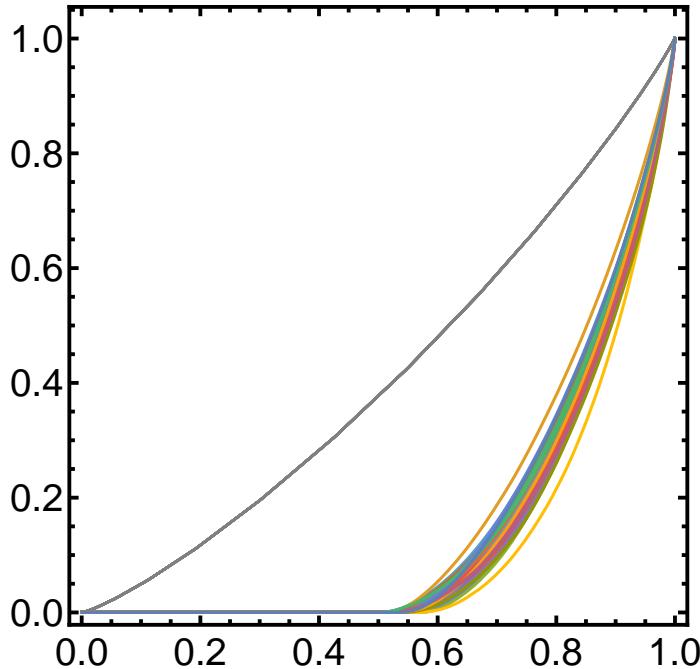
In[7]:= pen13centrality = ListPlot[Table[Flatten[
    Reverse[Table[redmeasures[[r, 5, Flatten[Position[redmeasures[[r, 1]], penhens[[ph]]]]]], {r, 1, Length[redmeasures]}] //.
      {} \[Rule] Null]], {ph, 1, Length[penhens]}]],
  PlotStyle \[Rule] PointSize[0.01], AspectRatio \[Rule] 1, Frame \[Rule] True,
  FrameStyle \[Rule] Directive[Thick, Black, 24],
  FrameTicks \[Rule] {{{{0, 0.01, 0.02, 0.03}, {}}, {{{0, "30"}, {30, "60"}, {60, "90"}}, {}}}}]
```



```
In[8]:= Export["pen13centrality.jpg", pen13centrality]
Out[8]= pen13centrality.jpg
```

```
In[1]:= pen13strengthdistributionred =
  Show[ListPlot[Transpose[{Range[k] / k, PadLeft[Accumulate[
    Sort[Table[Random[PoissonDistribution[N[max / k]]], {k}]]], k, 0] / max}],
  PlotStyle -> Gray, Frame -> True, AspectRatio -> 1, FrameStyle ->
  Directive[Thick, Black, 20], LabelStyle -> Directive[18, Black]],
  ListLinePlot[Table[redmeasures[[i, 2]], {i, 1, Length[redmeasures]}]]];
```

```
In[2]:= pen13strengthdistributionred
```

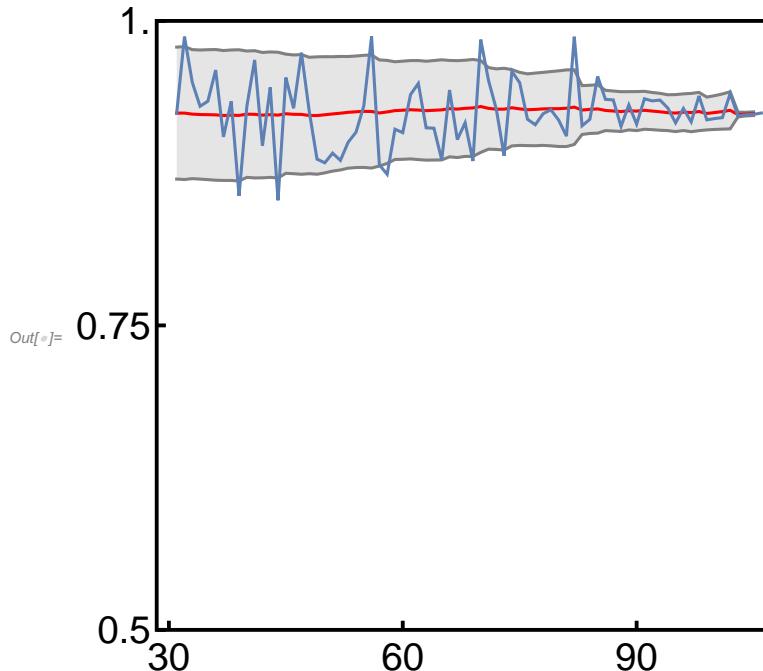


```
Out[2]=
```

```
Out[2]= pen13strengthdistributionred.jpg
```

```
In[3]:= stdr = Reverse[
  Table[StandardDeviation[redmeasures[[1 ;; i, 3]]], {i, 2, Length[redmeasures]}]];
mr = Reverse[Table[Mean[redmeasures[[1 ;; i, 3]]], {i, 2, Length[redmeasures]}]];
```

```
In[6]:= pen13density = Show[ListLinePlot[{mr + 1.96 stdr, mr - 1.96 stdr}, Filling -> {1 -> {2}}, PlotStyle -> Gray, PlotRange -> {All, {0.5, 1}}], ListLinePlot[mr, PlotStyle -> Red], ListLinePlot[Reverse[redmeasures[[All, 3]]]], Frame -> True, AspectRatio -> 1, FrameStyle -> Directive[Thick, Black, 20], LabelStyle -> Directive[18, Black], FrameTicks -> {{{{0.5, 0.75, 1.0}, {}}, {{0, "30"}, {30, "60"}, {60, "90"}, {}}}}]
```



```
In[6]:= Export["pen13density.jpg", pen13density]
```

```
Out[6]= pen13density.jpg
```

---

## Pen 14

### Edgelist

```
In[1]:= SetDirectory["M://Chicken Research/ChickenRanging4"]
```

```
Out[1]= M:\Chicken Research\ChickenRanging4
```

```
In[2]:= data = Get["transitiondata"];
```

```
In[3]:= SetDirectory["M://Chicken Research/ChickenSocialNetwork"]
```

```
Out[3]= M:\Chicken Research\ChickenSocialNetwork
```

```
In[4]:= data = Cases[data, {_, _, _, 14, ___}];
```

```
In[5]:= penstring = "14";
```

```
In[6]:= SeedRandom[1492]
```

```
In[7]:= penhens = Cases[allhens, {_, penstring, ___}] [[All, 3]];
```

```

In[1]:= n = Length[penhens];

In[2]:= edgelistpen = {};
For[d = 1, d ≤ Length[datelist], d++,
  pen = Cases[data, {_, _, _, _, _, datelist[[d]], ___}];
  pen = Sort[pen, #1[[6]] < #2[[6]] &];
  If[Length[pen] > 1,
    AppendTo[edgelistpen, ParallelTable[findFriend[i], {i, 1, Length[pen] - 1}]]]
  ]
];
In[3]:= edgelistpen = DeleteCases[Flatten[edgelistpen, 2], {}];
edgelistpen = Sort /@ edgelistpen;
edgelistpen = Tally[edgelistpen];
edgelistpen = DeleteCases[edgelistpen, {{v1_, v2_}, _} /; v1 == v2];

In[4]:= Save["edgelistpen14red", edgelistpen];

In[5]:= edgelistpen = Get["edgelistpen14red"];

In[6]:= indivobservations = Table[Length[Cases[data, {penhens[[i]], ___}]], {i, Length[penhens]}];

```

## Full network

```

In[1]:= amat = Table[0, {n}, {n}];

In[2]:= For[i = 1, i ≤ n, i++,
  For[j = 1, j ≤ n, j++,
    pair = Cases[edgelistpen, {{penhens[[i]], penhens[[j]]}, ___}];
    If[Length[pair] > 0,
      amat[[i, j]] = pair[[1, 2]]
    ]
  ]
];
In[3]:= amat = amat + Transpose[amat];

In[4]:= aimat = N[Table[If[indivobservations[[i]] + indivobservations[[j]] > 0,
  amat[[i, j]] / (indivobservations[[i]] + indivobservations[[j]] - amat[[i, j]]), "NA"],
  {i, Length[penhens]}, {j, Length[penhens]}]];

In[5]:= penaigraph = WeightedAdjacencyGraph[aimat // . {"NA" → 0, 0. → 0},
  EdgeWeight, DirectedEdges → True, EdgeStyle → Arrowheads[0]];

In[6]:= penaimultigraph = AdjacencyGraph[Round[(aimat // . {"NA" → 0, 0. → 0}) 10000],
  DirectedEdges → True, EdgeStyle → Arrowheads[0]];

In[7]:= max = Total[edgelistpen[[All, 2]]]; (*total number of links*)

In[8]:= k = n (n - 1);

In[9]:= strdist =
  Transpose[{Range[k] / k, PadLeft[Accumulate[Sort[edgelistpen[[All, 2]]]], k, 0] / max}];

In[10]:= gd = N[GraphDensity[penaimultigraph]];

In[11]:= ass = GraphAssortativity[penaigraph, FindGraphCommunities[penaigraph]];

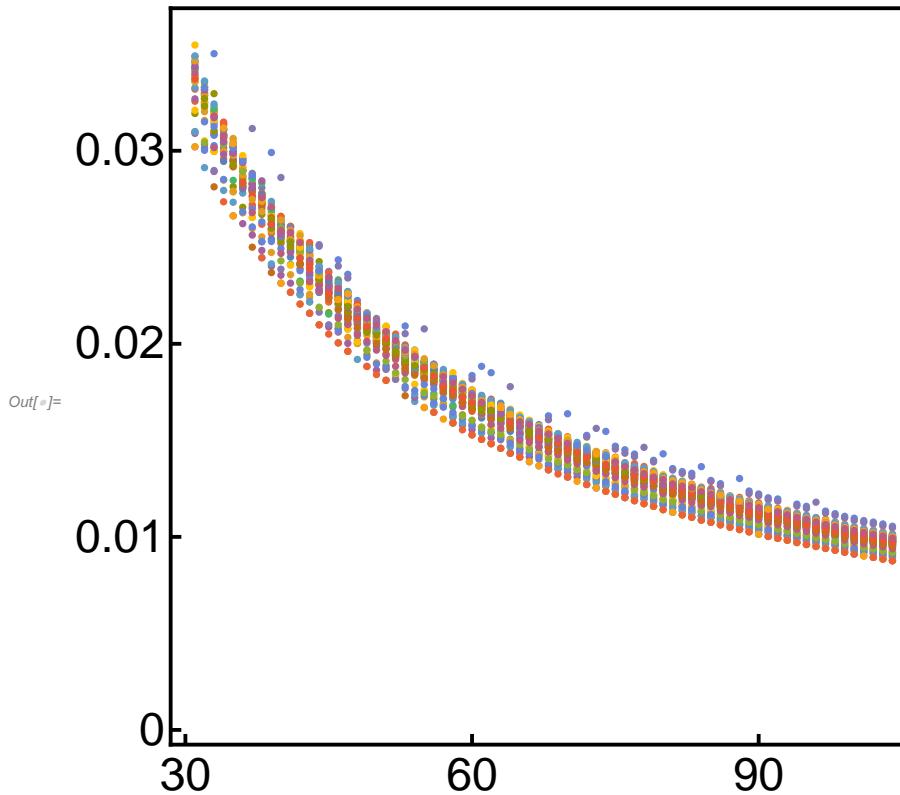
In[12]:= btw = BetweennessCentrality[penaimultigraph];

```

```
In[1]:= prc = PageRankCentrality[penaimultigraph, 0.1];
In[2]:= lsc = StatusCentrality[penaimultigraph];
In[3]:= gc = FindGraphCommunities[penaigraph];
In[4]:= measures = {penhens, strdist, gd, btw, prc, lsc, gc};
```

## Reduced networks

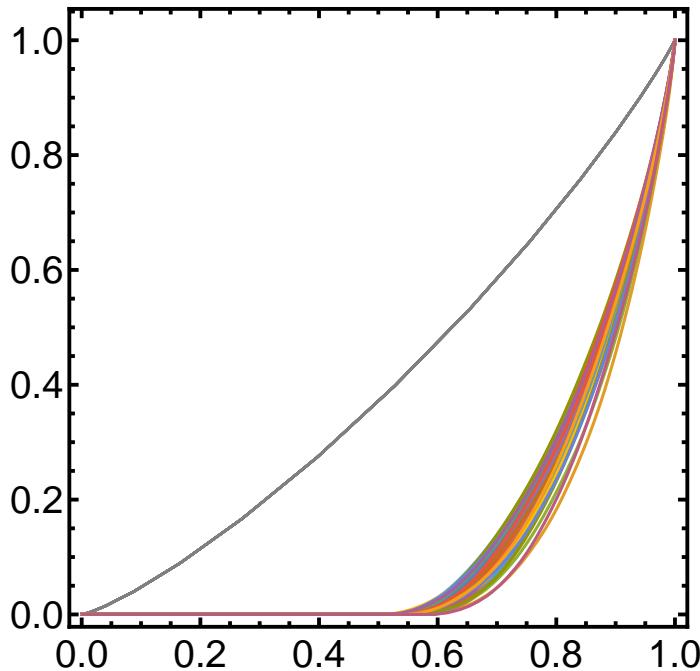
```
In[1]:= redmeasures = Table[reducedsampling[r], {r, 1, n - 30}];
In[2]:= Save["measures14", measures];
Save["redmeasures14", redmeasures];
In[3]:= measures = Get["measures14"];
In[4]:= redmeasures = Get["redmeasures14"];
In[5]:= pen14centrality = ListPlot[Table[Flatten[
Reverse[Table[redmeasures[[r, 5, Flatten[Position[redmeasures[[r, 1]], penhens[[ph]]]]]], {r, 1, Length[redmeasures]}] //.
{} -> Null]], {ph, 1, Length[penhens]}]],
PlotStyle -> PointSize[0.01], AspectRatio -> 1, Frame -> True,
FrameStyle -> Directive[Thick, Black, 24],
FrameTicks -> {{0, 0.01, 0.02, 0.03}, {}, {{0, "30"}, {30, "60"}, {60, "90"}}, {}}}]
```



```
In[6]:= Export["pen14centrality.jpg", pen14centrality]
Out[6]= pen14centrality.jpg
```

```
In[1]:= pen14strengthdistributionred =
  Show[ListPlot[Transpose[{Range[k] / k, PadLeft[Accumulate[
    Sort[Table[Random[PoissonDistribution[N[max / k]]], {k}]]], k, 0] / max}],
  PlotStyle -> Gray, Frame -> True, AspectRatio -> 1, FrameStyle ->
  Directive[Thick, Black, 20], LabelStyle -> Directive[18, Black]],
  ListLinePlot[Table[redmeasures[[i, 2]], {i, 1, Length[redmeasures]}]]];
```

```
In[2]:= pen14strengthdistributionred
```

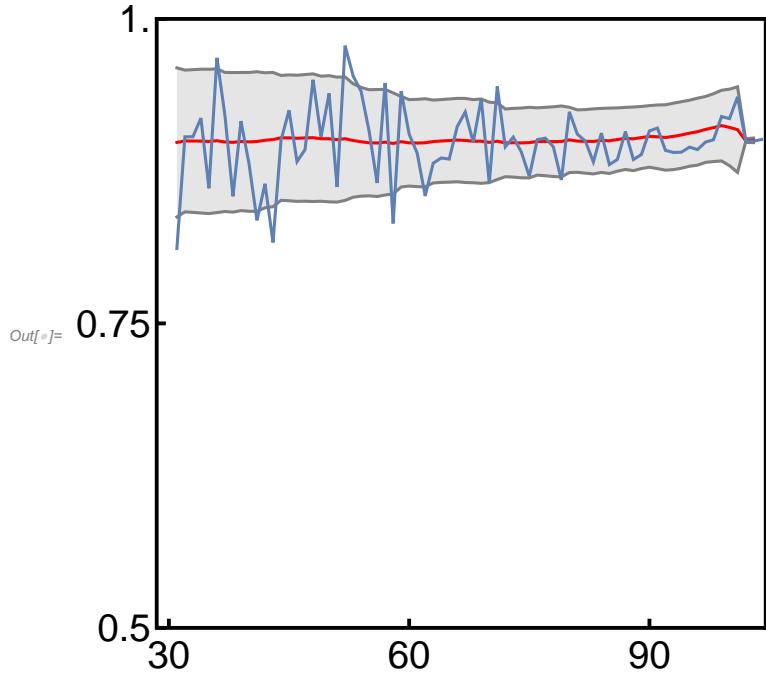


```
In[3]:= Export["pen14strengthdistributionred.jpg", pen14strengthdistributionred]
```

```
Out[3]= pen14strengthdistributionred.jpg
```

```
In[4]:= stdr = Reverse[
  Table[StandardDeviation[redmeasures[[1 ;; i, 3]]], {i, 2, Length[redmeasures]}]];
mr = Reverse[Table[Mean[redmeasures[[1 ;; i, 3]]], {i, 2, Length[redmeasures]}]];
```

```
In[6]:= pen14density = Show[ListLinePlot[{mr + 1.96 stdr, mr - 1.96 stdr}, Filling -> {1 -> {2}}, PlotStyle -> Gray, PlotRange -> {All, {0.5, 1}}], ListLinePlot[mr, PlotStyle -> Red], ListLinePlot[Reverse[redmeasures[[All, 3]]]], Frame -> True, AspectRatio -> 1, FrameStyle -> Directive[Thick, Black, 20], LabelStyle -> Directive[18, Black], FrameTicks -> {{{{0.5, 0.75, 1.0}, {}}, {{0, "30"}, {30, "60"}, {60, "90"}, {}}}}]
```



```
In[7]:= Export["pen14density.jpg", pen14density]
```

Out[7]= pen14density.jpg